Enabling Educators to Teach Children about Data with Physical Computing

Lorraine Underwood

School of Computing and Communications

Lancaster University

A thesis submitted for the degree of

Doctor of Philosophy

January 2025

Enabling Educators to Teach Children about Data with Physical Computing

Lorraine Underwood, BSc (Hons), PGCE.

School of Computing and Communications, Lancaster University

A thesis submitted for the degree of *Doctor of Philosophy*. December 2024

Abstract

Data has become the foundation of our digital world. We create and consume mass amounts of data in our everyday lives and both this creation and consumption is increasing with the popular growth of topics such as machine learning and artificial intelligence. These growing areas rely on vast amounts of data. How reliable and accurate an artificial intelligence is depends on not only the type of machine learning algorithm it uses, but the data it consumes. Data is used to justify government policy, commercial and environmental decisions that affect the world we live in. Data can have a real impact on our lives. Society needs to have a better understanding of what data is, how it is collected, stored and how it can be analysed and visualised.

Many educators can feel overwhelmed and underprepared trying to add new topics into the already full curriculum. Combining the teaching of data with an existing, engaging and creative topic such as physical computing could be the answer to adding more and fulfilling existing data science education in the curriculum.

This thesis commences with a further explanation of why we need better data science education in schools and how physical computing can support educators. The background section delves further into both areas giving examples of tools and methods. A related work section depicts the BBC micro:bit as a potential tool for data collection in schools. It analyses current research with the micro:bit before the next section details a case study of a project using the micro:bit as a data collection tool in 20 schools. These findings inform the design of a set of tools to support educators to teach data science. The tool is detailed and evaluated in the final sections of this thesis.

The contributions of this these include (1) an understanding of how physical computing can become engaging for primary school aged children in the context of data science learning, (2) a new approach for working with teachers in a technology design process and (3) the design process and set of requirements synthesised from research and user interactions. The ultimate aim is to guide future research toward designing pragmatic, engaging and educationally sound physical computing tools to support educators to teach data science.

Acknowledgements

I would like to extend my deepest gratitude to my supervisors Professor Joe Finney and Dr Elisa Rubegni for their exceptional guidance and support during my PhD journey. You helped me grow and pushed me further than I ever thought I could go.

This thesis is only possible through the collective support, encouragement and love of the many amazing people around me. Thank you to the many members of the Fun Office over the years: Bella, Matt, Hannah and Davina, but especially my dear friends Kobi, Angie and Jan. You have kept me fed, watered and alive and put up with my endless questions. A special thank you to the OG Fun Office member Liz Edwards, this thesis wouldn't be possible without you. Your enthusiasm and energy are contagious. I miss you and hope we will work together again in the future.

Thank you to Bryan Williams for helping me understand data science more. I still owe you some cake. To Professor Richard Harper for equally challenging and supporting me during these final months. Thank you to the Devices Lab for keeping me suitably distracted and sane: Matt, James, Aron and Kier. I'm excited for what the future holds for us all. Thank you to Professor Steve Hodges for his support and guidance. His insightful mentorship and advice have helped me grow in confidence as an academic. Thank you to the Microsoft Team: someone whose boots I could never fill: Dr James Devine; Tom and Catherine for hosting us in Seattle and stopping their dogs from eating me; Peli and Jacqueline and the whole team responsible for micro:bit and MakeCode. None of this would be possible without you.

I want to thank all the children and teachers who took part in all my studies. I am so grateful for your patience, your time and effort, and your feedback on my research. I have been so lucky to work with such kind people. Your support has made a big impact on me.

Speaking of children, I should probably thank my own. David and Sam, the first testers of any lesson or technology I make. You are my heart, my inspiration and my best invention. I am so lucky to have such a great support network full of love and fun which leads me to my husband Phil. My number one fan, my love, my best friend and my rock: I can't wait to overtake you as the best Dr Underwood in our family.

Go raibh maith agat, a chairde.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes but excluding the bibliography. A rough estimate of the word count is: 66,204.

Lorraine Underwood

Summary of contributions

1. **Lorraine Underwood**, Elizabeth Edwards, Elisa Rubegni, Steve Hodges, John Vidler, Joe Finney. 2024. Toolkit for Educators of Data Science: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*
2. **Lorraine Underwood**, Thomas Ball, Steve Hodges, Elisa Rubegni, Peli de Halleux, Joe Finney. 2024. MicroCode: live, portable programming for children via robotics. *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2024). [212]
3. **Lorraine Underwood**, Joe Finney, Elisa Rubegni, Steve Hodges. 2024. Tangible tools for data science education. 2024. *Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research* (New York, NY, USA, 2024). [213]
4. James Devine, Steve Hodges, Thomas Ball, Michał Moskal, Peli de Halleux, Gabriele D'Amone, David Gakure, Joe Finney, **Lorraine Underwood**, Kobi Hartley, Matt Oppenheim, Paul Kos. Plug-and-Play Physical computing and Device Prototyping with Jacdac. 2024. *GetMobile: Mobile Comp. and Comm.* 28, 2 (Jul. 2024), 38–43. [52]
5. Thomas Ball, Joe Finney, Steve Hodges, Elisa Rubegni, **Lorraine Underwood**, Jayne Everson, R Benjamin Shapiro, Colby Tofel-Grehl, Rojin Vishkaie. 2024. Imagining Inclusive Digital Maker Futures with the BBC micro:bit. *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, Jun. 2024), 1032–1034. [10]
6. Kobi Hartley, Elisa Rubegni, **Lorraine Underwood**, Joe Finney, Thomas Ball, Steve Hodges, Peli De Halleux, James Devine, Eric Anderson, Michał Moskal. 2024. Meet MicroCode: a Live and Portable Programming Tool for the BBC micro:bit. *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2024), 355–370. [93]
7. Elizabeth Edwards, John Edward Vidler, **Lorraine Underwood**, Elisa Rubegni, Joe Finney. 2023. Supporting fieldwork for primary education with Computing - micro:bit, clip:bit and game controllers, in Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. 2023, Association for Computing Machinery: Chicago, IL, USA. p. 553–557. [64]

8. **Lorraine Underwood,** Elizabeth Edwards, John Edward Vidler, Elisa Rubegni, Joe Finney. 2023. Introducing Classroom Cloudlet: a mobile, tangible, and transparent approach to Internet of Things education, in Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. Association for Computing Machinery: Chicago, IL, USA. p. 740–744. [211]
9. Mannu Lambrichts, Raf Ramakers, Steve Hodges, James Devine, **Lorraine Underwood**, Joe Finney. 2023. CircuitGIue: A Software Configurable Converter for Interconnecting Multiple Heterogeneous Electronic Components. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2023. 7(2): p. Article 63. [127]
10. James Devine, Michal Moskal, Peli De Halleux, Thomas Ball, Steve Hodges, Gabriele D'Amone, David Gakure, Joe Finney, **Lorraine Underwood**, Kobi Hartley, Paul Kos, Matt Oppenheim. 2022. Plug-and-play Physical computing with Jacdac. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 2022. 6(3): p. Article 110. [53]
11. **Lorraine Underwood**, Karen Smith, Elisa Rubegni, Joe Finney. 2022. Energy in Schools: Empowering Children to Deliver Behavioural Change for Sustainability, in Interaction Design and Children. Association for Computing Machinery: Braga, Portugal. p. 308–314.[210]
12. **Lorraine Underwood** and Joe Finney. 2022. Introducing the Intranet of Things for Education, in 6th FabLearn Europe / MakeEd Conference 2022. Association for Computing Machinery: Copenhagen, Denmark. p. Article 21. [214]

Table of Contents

List of abbreviations and acronyms

Abbreviation	**Meaning**	**Page**
A-Levels	Advanced Levels	14, 20
AI	Artificial Intelligence	7, 13, 87, 93
API	Application Programming Interfaces	56
BCS	British Computer Society	19
CAS	Computing At School	19, 24
CPD	Continuing Professional Development	19, 32
CS	Computer Science	1, 3
EYFS	Early Years Foundation Stage	16
GCSE	General Certificate of Secondary Education	13, 20, 21

GDPR	General Data Protection Regulation	11, 48
GPIO	General Purpose Input Output	39, 40,63
ICT	Information Communication Technology	17, 18, 20, 21, 24
IDE	Integrated Development Environment	19, 36, 44
IoT	Internet of Things	53, 62, 63, 84, 88, 91, 92, 94, 95, 105, 108, 109, 111
IT	Information Technology	5, 17, 19, 20, 55, 93, 104, 117, 118, 188
LED	Light Emitting Diode	34, 38, 41, 45, 46, 62, 67, 68, 69, 78, 88, 89, 90, 114, 132, 134, 136, 139, 141, 143, 144, 145, 172, 173, 186, 187, 198, 212, 214, 216, 221
ML	Machine Learning	9, 13, 79, 84, 87, 88, 92, 93, 95
MMR	Measles, Mumps and Rubella	12
NCCE	National Centre for Computing Education	32, 53, 54, 69, 93, 95
OECD	Organisation for Economic Co-operation and Development	1
PBL	Project-Based Learning	23, 24
PGCE	Post Graduate Certificate in Education	18, 19
QTS	Qualified Teacher Status	18
RAM	Random Access Memory	1, 70
RGB	Red Green Blue	33, 40
STEAM	Science Technology Engineering Art Mathematics	84, 90
STEM	Science Technology Engineering Mathematics	18, 86, 90, 121, 207
T-Levels	Technical Levels	14
UKCCIS	UK Council for Child Internet Safety	54

List of Tables

List of Figures

Chapter 1

Motivation

In an increasingly data-driven world, a core set of data literacy skills are needed to enable individuals and our collective society to make informed decisions. Data literacy is quickly becoming a key skill in everyday life. More and more data are being collected through our personal devices, while online social media interactions and business transactions result in huge volumes of data and metadata that is stored, analysed and traded. As a result, all manner of datasets impacts many aspects of our daily lives. The author firmly believes that people across society need to understand what data science is and take control of their personal data. They need to know when they are sharing it, with whom, and how it is shared. They must have this knowledge to question the ethics and privacy around collecting and sharing data, as well as making informed decisions with and about data.

As the number of jobs that require data literacy grows, the world also needs critical, vigilant data scientists. For all these reasons, there is a growing interest in understanding how data science can be taught in schools [82, 94, 177]. This is further strengthened by the Organisation for Economic Co-operation and Development (OECD) recognising data literacy as a core foundation in their Learning Compass 2030 [158]: *"an evolving learning framework that sets out an aspirational vision for the future of education".*

Computer Science

Computer Science (CS) is the study of computation and information. It is a fast-growing subject that spans theoretical disciplines such as algorithms, applied disciplines such as hardware and software but is often associated with computer programming. The subject has grown quickly due to the fast development of technology. The Apollo computer that went to the moon in 1969 had 4 kilobytes of Random-access memory (RAM). Fifty-five years later the Apple Watch 9 has 64 gigabytes of RAM: 67,108,864 kilobytes. The size and cost of computers have decreased over time with smaller more sophisticated computers becoming more affordable in society.

Computers are used in many aspects of people's lives. Even the idea of what is a "computer" has changed from a stand-alone personal computer with a keyboard and a mouse, to a watch on a wrist or a touch screen to order food in a restaurant. Computer Science is an exciting area to be involved in, and Computer Science education needs to constantly change to keep pace. This can create challenges for educators and schools but also opportunities to involve students in these exciting new areas.

Physical computing

One exciting new area of Computer Science education that has seen a growth in schools in recent years is physical computing. Physical computing is the "*creative design and implementation of interactive, physical objects and systems, which are programmed, tangible media that communicate with their environment through sensors and actuators*" [167]. Students use both hardware and software to create a physical object that can function in the real world. This allows students to be creative with Computing, to realise their ideas into a physical form that can sense and react to the world around it.

Data science

Another area of growth within Computer Science is data science. Again, this growth has been helped by the advances in Computer Science. It has become much easier to collect, store and transfer huge amounts of data. As computers become more prevalent in our lives, the amount of data collected and analysed increases. From our Internet history to our credit card purchases, every action creates data that is stored. As well as human actions, data is collected about our world: weather, nature, animals. Governments, companies and individuals can collect masses of data about our world. This data can have an impact on their decision making.

In order to successfully navigate this data-driven world, it needs to be ensured that future generations understand data. They need to have the knowledge to be able to question data collection and analysis that is done on their lives and on their behalf, and to understand those data-driven decisions. Computer Science education needs to include data science; it should be disseminated from early education in primary schools through to secondary. Society needs to understand what data is, how it is collected, where it is stored to be able to ask: why? Why is this data stored, what is the purpose of it and who: who has access to what data and again: why? Data knowledge cannot be passive. Students need to be empowered to understand and use data to their advantage.

Combining data science and physical computing

The author suggests using physical computing as a tool to support the teaching of data science. Collaborative learning [99, 136], engagement [140], and increasing students' confidence (particularly for girls) [185] are some of the benefits that physical computing has brought to the classroom, not just in Computing. Hodges et al [96] note that physical computing creates "diverse connections" to other subjects.

In the remainder of this chapter, the rising popularity of physical computing is highlighted, as well as its origins and examples of its use by children. The increasing need for data science is explained further and also how it can benefit society, as well as the risk and challenges of both these topics. It is argued that not only is physical computing a powerful tool to teach data science, but the two topics complement each other and provide mutual benefits. The combination can empower students to use physical computing to embed data science into the physical world, to make the abstract solid and create transparency in the data pipeline. Data science gives physical computing tools purpose. Students are creating real tools to be used in the real world for a real purpose. Finally, a set of research questions for this thesis will be defined.

1.1 The rising popularity of physical computing in education

Physical computing has become very popular in the last decade. Cheap, easy to use tools have led to a surge in physical computing lessons in national curricula around the world. Similarly, many of those who are currently underserved by CS education programs have responded positively to a CS education approach built on physical computing hardware [4, 16, 17]. Let's look at its origins in history.

1.1.1 A brief history of physical computing

Constructivism is an established learning theory developed by Piaget which suggests that knowledge is constructed by the student, rather than passively taken in [17]. It builds on the work of Dewey [54] , Piaget [164] and Bruner [30]. Learning is seen as an active process in which knowledge and meaning are constructed in interaction with the new topics. Papert built on this theory and coined the term "constructionism" which combines the learning theory constructivism with actual hands-on construction. His focus was on the art of learning and how students learn to learn by making things. Papert argues that when the learner is consciously engaged in constructing an artefact "*whether it's a sandcastle on the beach or a theory of the universe*": they are building knowledge [162]. Constructionist learning has a long tradition in Computer Science education.

Physical computing is the process of "creatively designing tangible interactive objects or systems using programmable hardware" [7]. It involves the physical construction of a public entity, a physical thing. As Pyzybylla et al. [169] state, physical computing brings together constructionist learning and typical processes of Computer Science education in a creative and practical fashion.

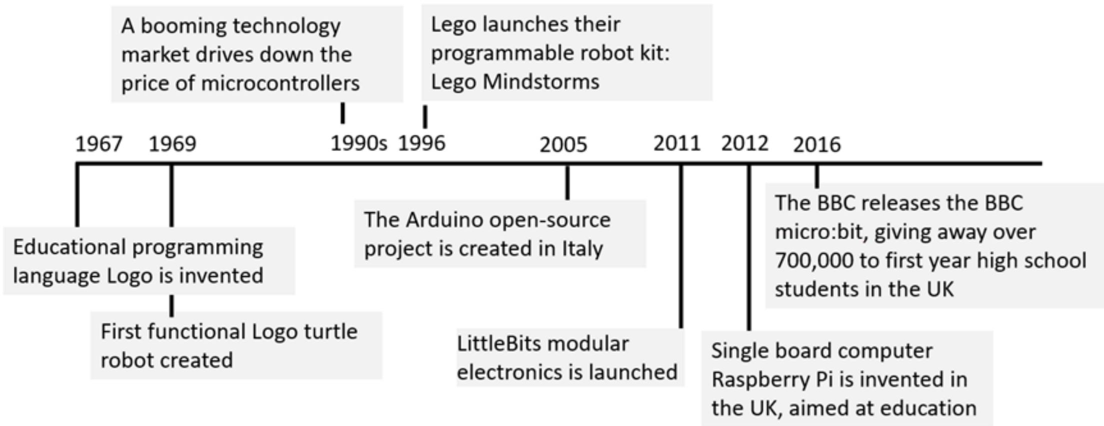


Figure 1 A timeline of a sample of Physical computing devices.

In the 1960s the first uses of physical computing in the classroom were seen through the Logo programming language which allowed students to program an icon on the screen that looked like a turtle to move around and draw shapes. The students learnt how to code by creating variables and loops to make more and more complicated drawings. In 1969 a giant robot turtle was developed that could be tethered to the computer and follow the turtle instructions. The robot turtle had a pen attached to the underside to replicate the shapes on the screen in the real world.

Figure 2 (Left) children using the first Logo Turtle Robot [1]. (Right): Seymour Papert with a smaller turtle robot that just drew a fish[2]

1.1.2 A growing field

Computer Science education followed the growth of Computing and Computing hardware in society. As personal computers (PCs) became cheaper and smaller they appeared in more and more homes and schools followed. Whole computer rooms were created in schools for students to sit with keyboards and mice. Students were taught how to touch type, and how to use software tools such as Microsoft Word and Excel from the Microsoft Office suite. These lessons took place in newly created Information Technology (IT) classes.

However, with the shrinking of both the physical size, cost and complexity of processors in the 1990s there was a change in physical spaces which has affected how IT was taught. Computer rooms started to be replaced with laptop or tablet trollies that could be rolled into a classroom. This made integrating creative Computing topics and other subjects easier. Educators have more space to integrate the learning of IT skills into other subjects when they can easily ask the students to take their tablets out and go to a resource, or create a poster, etc.

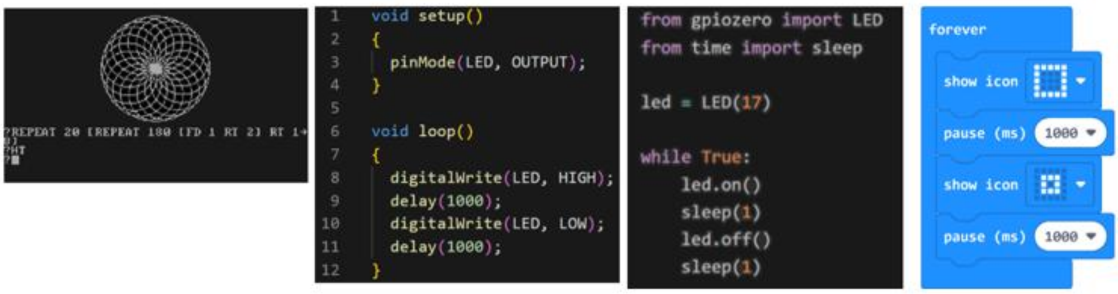
More physical computing devices were developed for education over the years, such as Lego Mindstorms in 1996 and what would become known as Arduino in 2005. These gave educators and students the freedom to be creative in their IT lessons. As Pyzybylla et al. [169] states, physical computing can be the carrier to

[1] http://media.tumblr.com/tumblr_m43bakRVSb1rpx08t.png
[2] https://el.media.mit.edu/logo-foundation/resources/papers/pdf/hoc.pdf

foster creative learning in Computer Science education. In 2012 Raspberry Pi introduced their cheap, accessible computer aimed at education. It encouraged students to build and program electronic circuits using the Python programming language. In 2016 the BBC, alongside 29 other partners, developed the BBC micro:bit: a microcontroller with embedded with sensors that could be programmed with multiple languages like Python, TypeScript and the newly developed block-based language MakeCode.

A brief look at the different programming languages on the different physical computing devices over the years shows how user-friendly programming these devices have become, not only in their physical size but the software used to program them.

Figure 3 shows all the languages next to each other. Logo turtle (a) used key words to create graphics on the screen, these could be then downloaded to the robot. Arduino (b) has its programming environment that needs to be installed on a computer to write C code to then download to the board. Raspberry Pi is a computer in itself (c), users can type and run the code directly onto the device. The BBC micro:bit (d) uses a web-based tool Microsoft MakeCode that is a block-based language. Block-based languages used blocks of code that students drag and drop onto a canvas. The blocks snap together if placed in a valid sequence. This helps the student create programs that will run easily and quickly.



(a) (b) (c) (d)

Figure 3 (a) Logo turtle[3], Logo, 1969 (b) Arduino[4], C, 2005 (c) Raspberry Pi[5], Python, 2012 (d) micro:bit, MakeCode[6],2016.

[3] https://commons.wikimedia.org/w/index.php?curid=79698503
[4] https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
[5] https://projects.raspberrypi.org/en/projects/physical-Computing/4
[6] https://makecode.microbit.org/

1.1.3 Example: Micro:bit Do Your Bit Challenge

Physical computing has a long history of supporting children learning Computing concepts [25]. In recent years there has been a surge in the number of products entering the education market. They are becoming increasingly accessible to educators through their decreasing costs, friendly user interfaces and robustness. This has made them more appealing to schools. The devices can be used across the curriculum in different subjects and across different age groups. Physical computing provides more practical and relevant examples of Computing, allowing students to solve problems that matter to them. Below are some examples of physical computing projects created by children.

Kenna and Margot from the US created a device that uses a sensor to detect how clean water is, it then shines a UV light onto the water to sterilise it [198]. The system is controlled by a BBC micro:bit and powered by solar power, Figure 4 left.

A secondary school student in Thailand, Siwachanat Saengsuwan, created a system using sensors and artificial intelligence to detect when parents parked illegally when dropping their children off at the school [159]. It uses an online Internet of Things service to send messages between the camera and a warning pole. The pole contains a warning light and alarm to let the parents know they are parking illegally, Figure 4 right.

Figure 4 (Left) Checking water for pollution. (Right) Illegal parking detection

1.2 The increasing need for data science education

As data becomes more prevalent in society, and the amount of data people share about themselves continues to grow, studies have shown that society needs to become data-savvy [137]. Franklin et al have shown that most future careers will require some knowledge of data, including statistics and data analytics [79].

In most cases, data is digital and therefore not tangible. People do not see the amount of data collected about them, or the data they create (or use) in any tangible way. For example: a phone provider may state that a customer has used 20MB of data last month, the customer may know that 20MB is a large file, because their work printer doesn't allow files over 20MB but is that a lot of data for one month's use? "A lot of data" is difficult to define. What might be a lot for one person, can be very small for another. If the customer wanted to use less data and save money, what actions could they take? How many emails is 20MB? How many minutes of video is it? How can data relate to physical actions? Data science education is needed to teach us about data, data collection, analysis and data representation in a real-world context.

Benefits of data science

Data science has brought many benefits to the world. The efficiency of collecting and being able to analyse large data sets is one such benefit. This has allowed the identification of problems. For example, the Alan Turing Institute has developed data science projects to gather data to form a national crop modelling framework in the United Kingdom (UK). This is to help predict how UK crops can adapt to climate change over the next 50 years. They pull data together from plant science, hydrology, soil science, insect population dynamics, economics, consumer behaviour and climate models to form an "integrated national crop modelling framework" [103]. Citizen science projects also gather data to develop insights into our world. Citizen science is research conducted with the support of the general public and/or amateur researchers. Examples include the public counting and submitting data on butterflies [22], Asian Hornets [170], plants [155], etc. that they find in their local area. The UK Pollinator Monitoring Scheme asks the public to download an app to count insects that pollinate [217]. As they state: "Pollinating insects play a vital role in our environment, ensuring that many of our crops and wild plants are able to set seed and produce fruit". They ask the public to count pollinating insects and report their numbers in the app. This will help them discover how pollinator populations are changing.

The benefits of citizen science are educating the public about scientific projects, increasing awareness about different topics, creating communities of interest and scientists gaining access to large data sets covering large and variable areas of the country.

Other data science projects use data sets to train machine learning (ML) algorithms to do the data collection more effectively. For example, human volunteers are identifying plants and animals in images gathered by scientists. These labelled images are then used to train machine learning algorithms to identify the plants and animals automatically, without human intervention [205]. In recent years machine learning has become easier to use to train models on custom data sets without the need for specialist computers or servers of computers.

Data challenges

There are many risks and challenges that are associated with data. There have been many cases of data breaches in organisations that store our personal data. In 2018 the world's largest ID database Aadhaar was infiltrated, exposing 1.1 billion Indian citizens' private data such as name, addresses, phone numbers and emails as well as biometric data like fingerprints and iris scans [2]. In 2013 a data breach occurred at Yahoo with over 3 billion accounts accessed [104]. Yahoo didn't admit the breach until 2016 and initially said 1 billion accounts were hacked, it updated this number to the correct 3 billion a year later.

Personal identifiable data such as names, address and in some cases bank details have been accessed by unauthorised and malicious parties. The risks of poor data science management by companies and governments aren't immediately evident, therefore educating society about data should be proactive rather than reactive. It's not just the illegal access of data that people should be aware of, but the ethical decisions made when collecting, analysing and presenting data that effects their lives.

The recent COVID-19 pandemic, the climate change situation and the ongoing claims of election fraud worldwide show us how data can be used to inform and misinform on a mass scale.

Sustainability

As well as the data collected and stored about us, society needs to be made more aware of the data they are consuming; how much data people use, how much data they download and upload to the Internet either browsing web pages or creating and watching videos over streaming services. Data uses energy and energy is not yet a reliable infinite source. The collection, analysis, streaming and storage of all data has an environmental impact which is not visible to many people. Research shows that demand for data centres and network services is increasing, and data traffic is continuing to grow [81]. The advancement of AI services has sped up this demand.

Each time ChatGPT is asked a question, it uses about 2.9 watt-hours of electricity. This is nearly ten times more than the energy needed for a Google search, which consumes about 0.3 watt-hours per query [6]. Society should be aware of the impact of their actions surrounding data.

Data currency

In an ever-increasing digital world, data has become the new currency. Companies offer discounts to new customers once they have signed up and confirmed their email address for marketing emails. The cost of the discount is outweighed by the value of a verified e-mail address. Supermarkets use loyalty cards not just to keep customers coming back to their shop but to gather data on their customers' purchasing behaviour. The supermarkets can gain information such as customer demographics (name, age, gender, postcode) [102] and "individual level purchase" (what was bought, where, when and how much) [125]. All this data is gathered from the loyalty card that is swiped when people shop, it is given away freely and has been shown to have little value to the customer [102]. At what point should people stop and consider the data they are giving away freely? If data is worth something to companies, should it be worth something to its owner?

Data privacy

When giving data to a company or government people need to able to ask questions about the data: what data they are collecting, where is the data stored, who (people and other organisations) has access to this data, will the data become anonymised at any point and if so, how? Finally, when will this data be deleted?

In the UK with General Data Protection Regulation (GDPR), individuals have the right to have personal data erased. This is known as the 'right to be forgotten' [173]. The right is not absolute and only applies in certain circumstances. When dealing with data, society needs to know what questions to ask around data privacy.

The misrepresentation of data

Another reason why more people should be taught data science skills is the misrepresentation of data, accidental or intentional. Data can be skewed in different ways to follow a narrative. There have been many cases when the data has been misrepresented.

The ruling political party in India, Bharatiya Janata Party (BJP) produced an infographic in 2018 that showed two bar charts showing the difference between fuel rates in 2004 and 2018 [24]. At first glance the graph looks like the cost of petrol has decreased as both bar graphs show a smaller bar at the end but a closer look at the numbers show the numbers increase, Figure 5. The data wasn't incorrect; the graph is not representative of the numbers shown on it. It is showing the rate of increase.

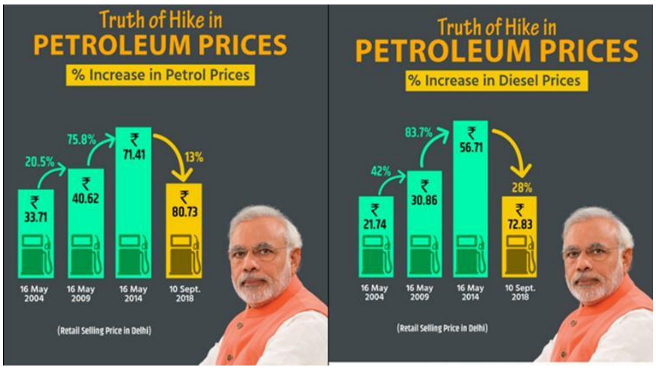


Figure 5 BJP petrol rates infographic 2018.

In the United States in 2020, the state of Georgia produced the graph shown in Figure 6 where the dates along the x axis aren't chronological, and the bars aren't in the same position every day [43].

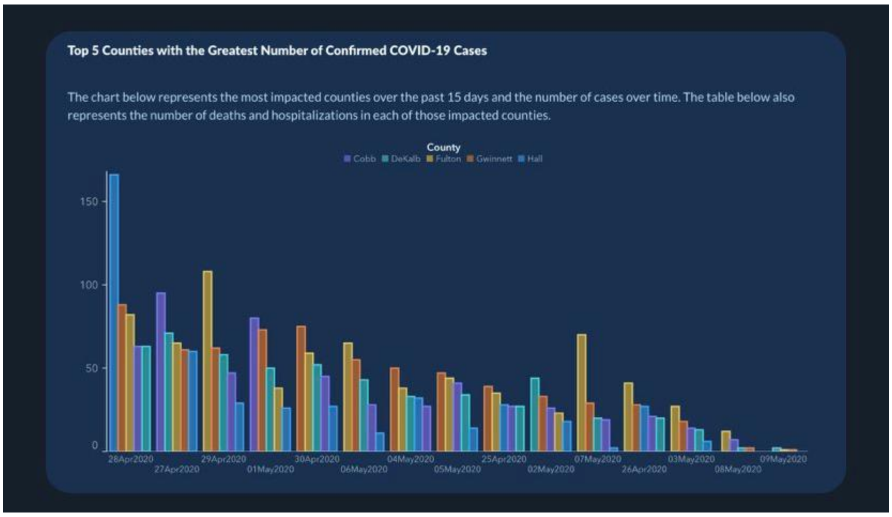


Figure 6 A retracted graph from Georgia, USA 2020

This misrepresentation of data has occurred recently, it has been done by large organisations and by ruling governments. The consequences can be far reaching and long lasting. In 1998 in the UK a then doctor, Andrew Wakefield, published an article in the prestigious medical journal the Lancet claiming a link between the measles, mumps, and rubella (MMR) vaccine and autism. His claim was eventually debunked, and he was struck off the medical register for fraud, but the effects of those claims are still felt today. MMR vaccine rates, and other vaccines such as rotavirus given to babies at 12 weeks old and the pre-school booster given at 3 years old, have all dropped [38]. None of the vaccines reached their target of 95% in 2023-2024. The drop in vaccines rates has led to a rise in the number of cases of measles and mumps [151]. Being able to spot misrepresentation of data is an important skill that can be developed through analysing and representing data on an individual level.

1.2.1 The value of data science skills

Finally, why should students be taught data science skills beyond being aware of their personal data security, their data use, other's use of their data and to spot misrepresentations of data?

Collecting data to analyse gives students an authentic learning experience. They are not looking at made up data from a textbook or data that may be out of date or not relevant to their context. The students can trust the data they have collected, because they collected it. Collecting real data can add value to students' experiences in the classroom [193]. Studies have shown that collecting data about the environment prompts students to ask questions about their local environment, to become involved with it [31]. It can create a connection between them. There have been calls to increase the focus on the application of Computing concepts like data science in the real world [175]. Collecting data in the real world can help increase this focus. Below are examples of data science projects that have contributed to society using data from real life.

H-Unique is a five year, €2.5m project of research to identify people through their hands. The project started as a “forensic identification of individuals from images of their anatomy in child abuse cases” [101]. The project calls on “citizen scientists” to contribute images of their hands. The images are stored in a database and is searchable by researchers looking at the anatomy and variations of the human hand. They use machine learning and artificial intelligence to then identify perpetrators of serious crime [9].

Citizen science is when the public become involved in scientific research. It has grown in popularity in recent years; Theobald et al. identified 388 unique biodiversity-based citizen projects. They estimated that between 1.36 and 2.28 million people volunteered for these projects each year [200].

1.3 A look at Computer Science education

To explain how Computer Science is taught in school examples are used from the UK system. UK primary schools run from ages 4/5 to 11. In the UK children legally have to start school aged 5 in Year 1, however free full-time education starts at age 4 in Reception, and many primary schools will have a Reception class in the school. Some schools will also have a Nursery for children aged 3 to 4 where children attend part time (see the top half of Table 1).

In the UK, most children start secondary school at age 11. There are a small number of middle schools that teach children from ages 11 to 14. Children will start preparing for state exams, GCSEs (General Certificate of Secondary Education) in Year 10 and complete them by the end of Year 11. Children can leave formal education at 16 but must be in some form of education, e.g. in school, traineeship or apprenticeship until they turn 18. If still in school children will prepare for the final set of state exams: the A-Levels (Advanced Levels) or the newly introduced T-Levels (Technical Levels). Table 1 details the ages of children at these different stages.

Child's age	**Year**	**Key Stage**
3 – 4	Nursery	Early years
4 – 5	Reception	Early years
5 – 6	Year 1	Key Stage 1
6 – 7	Year 2	Key Stage 1
7 – 8	Year 3	Key Stage 2
8 – 9	Year 4	Key Stage 2
9 – 10	Year 5	Key Stage 2
10 – 11	Year 6	Key Stage 2
11 – 12	Year 7	Key Stage 3
12 – 13	Year 8	Key Stage 3
13 – 14	Year 9	Key Stage 3
14 – 15	Year 10	Key Stage 4
15 – 16	Year 11	Key Stage 4
16 – 17	Year 12	Key Stage 5
17 – 18	Year 13	Key Stage 5

Table 1 The ages of children in each year of school and their corresponding Key Stage

1.3.1 Cross-curricular connections

Cross-curricular education is often seen as beneficial. Timmerman et al. refer to it as interdisciplinarity [203]. "*Interdisciplinarity turns knowledge into a coherent whole and bring lessons into the real world.*" They also argue that it allows for the study of objects, ideas and events from different angles and can put them into perspective in relation to each other.

In secondary schools, delivering cross-curricular education is a particular challenge as subjects are heavily siloed. Mathematics teachers teach Mathematics in the Mathematics classroom, Geography teachers teach Geography in the Geography classroom and so on. Many of the classrooms are designed and structured to match a specific subject area. There are few opportunities for cross-curriculum collaboration. Most secondary school teachers are specialists in their subject areas, they have a degree in this subject, and they have trained to teach it. Unlike primary school teachers they have no training in other subjects. This makes it difficult for secondary school teachers to teach other subjects besides their own. If teachers wanted to teach together this is again difficult in most secondary schools where the timetable is structured by subject. If the mathematics teacher wants to teach a topic with the help of a Geography teacher, either together or separately, their schedules may not align. Having two teachers teach one class at the same time takes up a lot of resources. Savage comments that this ability to collaborate is seen as an important factor on whether or not a cross-curricular approach is facilitated within a school [178].

In the UK students are tested throughout their education. In primary school students are tested on their English, Mathematics and Science skills through the Standard Assessment Tests (SATs) at the end of Key Stage 1 (aged 7) and the end of Key Stage 2 (aged 11). The Key Stage 2 SATs are more formal and school-level results are published nationally allowing for comparison with other schools.

Secondary school students are tested more regularly both internally and nationally. Most schools will have internal end of year exams for every subject. This regular testing makes it difficult for teachers to teach outside their subject area. They want students to do well in their subject and will not want to take focus away from it. As secondary school students move from Key Stage 3 to 4 to 5, they will drop subjects, keeping the core of Mathematics, English and Science and choosing several options from the remaining “optional” subjects. Teachers in these optional subjects again want to focus on their subject to give students the best experiences of it so they won’t drop it when they have a choice.

Primary schools have more opportunities to explore cross-curricular education. In the UK, many primary schools teach otherwise-siloed subjects thematically. For example, students studying ecological habitats may: learn about the Amazon rainforest in their Geography lessons, be exposed to Spanish and Portuguese words in their language lessons, and paint or sculpt different animals from the Amazon rainforest in art. These themes often run vertically through different year groups in a school; for a set period of time there is an all-encompassing theme across a school that all children are immersed in as they learn.

1.3.2 Primary school Computing

In primary school, from Key Stage 1, schools **must** teach English, Mathematics, Science, Design and Technology, History, Geography, Art and Design, Music, Physical Education (PE), Computing, Ancient and Modern Foreign Languages (at Key Stage 2). There is a separate framework for educating children from birth to 5 years old called the Early Years Foundation Stage (EYFS). In 2021, the strand "Technology" was removed from EYFS, but many schools still follow this guidance for teaching Computing in early years.

The field of Computer Science education incorporates a wide range of topics, from computational thinking to programming skills to data analysis. In many schools children are taught Computing skills often called "digital skills" as soon as they arrive in school, from using a keyboard and mouse to logging into a computer. In the subject of Computing many primary schools will focus on teaching unplugged computational thinking before other technical skills. This can involve paper-based activities. An example activity from Barefoot Computing is "Dance Move Algorithms" [47]. The students follow instructions to dance along to a dance routine; they then create their own sequence of instructions (an algorithm) to create their own dance routine. This teaches students the computational thinking skill approach of algorithms. An extension to the task is to "debug" someone's dance to see what they're doing wrong, another common computational thinking skill that can be taught without computers to younger children. Other computational thinking skills that are taught to children early in education, sometimes without a computer are: logic, evaluation, patterns, decomposition, and abstraction. These are covered in the Barefoot resources that are online and free to educators.

The UK government provides a national curriculum, a set of standards used by schools, so all children learn the same things. The National Centre for Computing Education [81] has developed a curriculum map for teaching Computing in the primary school stages of Key Stages 1 and 2, mapped to the Computing national curriculum.

Physical Setup

Children in primary schools are either taught by the normal classroom teacher or a specialist Computing teacher, the latter being less common. Computer rooms have been replaced with more mobile and flexible setups such as tablets or Chromebooks. These devices will be stored in a central location and borrowed by different classes as and when they need them. There might be a booking system or a timetable for when each class can use them. Devices are typically shared between students in the classroom as well. The students may have individual logins for saving work. Some schools may use a system like Google or Microsoft for students to save their work online, not on a local intranet. Many primary schools, particularly smaller ones, do not have an IT technician on staff. They might have a contract with a company for a technician to come out once a month/term or share a technician with other local schools.

1.3.3 Computing staff

In 2014 the UK Department of Education removed the subject Information Communication Technology (ICT) from the national curriculum and replaced it with Computing. ICT was judged as not being theoretical enough and only covering practical aspects of ICT such as using word processing and spreadsheet software. The new Computing subject covered these practical skills but added coding skills and theoretical aspects. Computing was broken down into three strands: Computer Science, Information Technology and digital literacy. Computer Science is focused more on **computational thinking** skills such as abstraction, algorithms and logic. Information Technology was very much the same as the old ICT subject and digital literacy aims to teach students to be "responsible competent, confident and creative users" of ICT [154].

When this change occurred, ICT had one of the lowest percentages of qualified teachers in England (35%) compared to other core subjects such as Mathematics (74%), Physics (69%), Chemistry (73%) and Biology (88%) [34]. Many of these ICT teachers had backgrounds in business and marketing where the use of office software was common and was sufficient to teach ICT in schools. In a 2017 survey, Secondary school Computing teachers who taught other subjects taught Business studies (23%), Mathematics (16%) and Design and Technology (12%). The change to Computing meant many of the existing ICT teachers lacked the skills needed to teach the new curriculum, particularly Computing skills such as coding.

Recruiting new staff

Recruiting teachers with the relevant qualification to a new subject takes time. In the UK the most common route to train as a teacher is to complete a university-based postgraduate certificate in education (PGCE) which leads to Qualified Teacher Status (QTS). This is an academic year long course (September to June) involving university learning and at least two school placements. Both primary and secondary school teachers take a PGCE. Secondary school teachers must have an undergraduate degree that is relevant to the subject they will be teaching. Some undergraduate degrees also integrate QTS via a Bachelor of Education route. There are bursaries of up to £30,000 for Computing graduates to undertake a PGCE.

In the last 6 years England has only reached its own recruitment target for trainee Computing teachers once. STEM is the grouping of the subjects Science, Technology, Engineering and Mathematics. Compared to other STEM subjects, Physics is the only subject to perform worse. Biology is by far the most popular, regularly surpassing its target for teacher recruitment. The last two years have been particularly low for Computing with recruitment figures under 40%. Table 2 summarises the total teachers recruited in both primary and secondary against their targets then details the targets reached for the STEM subjects Mathematics, Chemistry, Biology, Computing and Physics [67–73].

	17/18	**18/19**	**19/20**	**20/21**	**21/22**	**22/23**	**23/24**
Primary (total)	106%	103%	96%	130%	136%	93%	96%
Secondary (all subjects)	69%	65%	85%	106%	82%	59%	50%
Mathematics	79%	71%	64%	84%	95%	90%	63%
Chemistry	83%	79%	70%	80%	105%	86%	65%
Biology	86%	153%	166%	189%	117%	85%	93%
Computing	**66%**	**73%**	**79%**	**105%**	**69%**	**30%**	**36%**
Physics	68%	47%	43%	45%	22%	17%	17%

Table 2 Percentage of target reached for teachers on PGCE courses.

Keeping staff up to date

Another challenge with Computing staff is the changing nature of the subject. A teacher with a Computing degree older than 5 years may find themselves teaching topics they did not learn at university. Physical computing is not a topic in many university Computer Science degrees: their focus tends to be software though some may offer elective hardware courses. Data science is also not always a part of Computer Science degrees. If it does feature, it tends to be part of a set of elective modules or a specialist degree.

Teacher support

There are organisations in the UK that support Computing teachers. Computing at School (CAS) is a volunteer grassroots network of Computing teachers working together to share resources in primary and secondary Computing. They run CAS hubs locally for teachers to meet in person and develop resources and relationships. When the curriculum changed in 2014 the Department of Education provided funds CAS to support teachers further. The British Computing Society (BCS) is a professional body that represents those working in IT. They accredit IT qualifications in all levels of education, support a global community of IT professionals to promote and further the field of Computing.

Barefoot Computing was setup in 2014 with funding from the Department of Education. It was led by BCS and CAS. The goal of the Barefoot project was to prepare primary school teachers for the new Computing curriculum. After the initial funding, it was taken over by British Telecom (BT) and CAS. They recruited volunteers, typically experienced teachers, to visit schools and deliver continuing professional development (CPD) to groups of teachers. Their website contains lesson plans for teachers of all age groups to support them to teach Computing. Many of the resources are cross-curricular. A similar project was setup for secondary school teachers called Greenfoot [89]. This project was focused more on programming and developed an IDE (integrated development environment) for Java for students to learn how to program. It had some resources, including unplugged ones.

Primary teachers

Primary school teachers can be subject specialists in their roles, but training as a primary school teacher is under one qualification: a PGCE in Primary School

Education. All primary school trainee teachers receive the same training: to become familiar with the National Curriculum for England and explore approaches for teaching every subject.

Some of the trainees may have specialist undergraduate degrees in a subject such as English, Mathematics, etc. To become a primary school trainee, one needs a bachelor's degree in any subject. Therefore, a trainee may not have specific experience or training in a specific subject including Computing.

In summary there is a lack of qualified Computing teachers in schools in England through the shift in subject from ICT and the failure to recruit the required number of new teachers over 6 of the last 7 years. This shortage of teachers can have an adverse effect on the teaching of new topics in the Computing classroom.

1.3.4 Student engagement

A recent report from August 2024 by King's College London shows a worrying decline in the number of students taking a Computing or IT qualification in secondary school [113].

The study covered approximately 80% of state schools and showed the number of hours teaching Computing has declined 28% at Key Stage 3, 59% for Key Stage 4 and 22% for Key Stage 5 between 2010 and 2023. One of the reasons for the decline could be due to the lack of qualified teachers as discussed in the previous section.

The number of students pursuing Computer Science degrees has increased from 14,000 to approximately 21,000 in the last 10 years. Students choosing A-Level Computer Science has seen a four-fold increase over the same period, 18,306 students took the subject in 2023. 57% of schools now offer A-Level Computer Science and 88,000 students took GCSE Computer Science in 2023, with 80% of all schools providing the course.

However there has been an overall decline in the number of students taking a Computing related qualification, particularly girls. In 2023, 21% of the GCSE Computer Science cohort were girls, compared to 43% in 2015 in the previous ICT GCSE. 69% of girls took a Computing related exam at Key Stage 4 in 2013, in 2020 this number has decreased to just 17% and 10% in 2023 [91].

In 2022, 80% of secondary schools taught Key Stage 3 Computing, a decline from 91% in 2010. In 2013, 69% of female students and 72% of male students took a Computing related qualification at Key Stage 4. In the Royal Society Science Education Tracker Report of 2023, it shows only 10% of female students and 33% of male students took a Computing qualification [91]. This is a huge decline, particularly for female students. Female students were more likely to choose a qualification in ICT than Computing. In the King's College report [113], it was found that girls at Key Stage 3 were interested in topics such as "digital media, project work and presentation work, areas more akin to the previous ICT curriculum." GCSE Computer Science is heavily reliant on theory, students learn ***about*** algorithms, programming (including data types) and search algorithms but there is no longer any practical projects or coursework in any of the different exams. There is very little in terms of the topics girls were interested in at Key Stage 3 in GCSE Computing. The subject is simply not appealing to many of them.

The 2023 Science Education Tracker report noted how males are twice as likely to be interested in Computing than females. 61% for males and 32% for females across all school years. There is also a general decline in the percentage of students interested in Computing compared to previous reports. 75% of students in Year 7 in 2019 were interested in Computing while that number has dropped to 68% in 2023. This is in line with an overall drop of all students' interest in all science subjects. The report suggests the younger children have been affected greater by COVID-19 in their early years of education as their overall interest and self-rated ability in all subjects has dropped compared to their older peers.

Computing is often perceived as a difficult subject by students and parents. The 2017 Science Education Tracker report [92] of 4,081 young people aged 14 to 18 attending state-funded schools in England found one of the key reasons to not studying Computer Science quoted by the student was lack of interest. 55% of girls cited lack of interest compared to 38% of boys. When talking about careers that use science (including Computer Science) 90% agree they "require high grades" and 63% agree that they "are difficult to get into", with girls agreeing more than boys. The report itself notes that this underlines "the persistent perceptions of difficulty and a lack of confidence among female students".

The previous reports on students' engagement with Computing are based on secondary school students. There has been no UK report with primary school students on the same scale. Other research in the world around primary school students' attitude to Computer Science show gender stereotypes do exist around girls and Computer Science. Children as young as 6 endorse the stereotype that "girls are less interested than boys in Computer Science and Engineering." [138]. These negative stereotypes that girls are less interested and less able in Computer Science and Engineering contributes to gender disparities in these areas [21, 37, 139].

1.3.5 Teaching techniques in Computer Science education

There are many educational theories that work well in Computer Science. Sentance et al. [186, 187] surveyed 1,417 teachers to find out what techniques or methods teachers reported as successful for teaching Computer Science. Five pedagogical strategies emerged: **unplugged type activities, contextualisation of learning, collaborative learning, developing computational thinking,** and **scaffolding programming tasks.**

Unplugged type activities include programming and problem solving without a computer [16]. These are often done first as part of a progression pathway in Computer Science Education, with younger ages being taught with unplugged activities before moving onto digital programming tasks either with physical computing or coding on computers/tablets. Unplugged type activities suit the younger ages as they tend to be less complicated, require less reading and be more hands on than plugged activities. Coding techniques like iterations and variables can also be taught in an unplugged manner.

Contextualisation of learning is when teachers relate the subject to other aspects of the curriculum or real life. They can add context with both unplugged or plugged tasks: for example, writing an algorithm to make a cup of tea (unplugged) or coding a robot to move a block from point A to point B on a map (plugged). Teaching Computing with other subjects in the curriculum has a lot of potential as it can build on the strengths of the teacher's knowledge in a different subject from Computing.

Collaborative learning includes pupils working in pairs or larger groups. This is quite common in Computing classrooms where resources are shared between students. Teachers take advantage of this by using techniques such as pair programming: pairs of students work collaboratively on a shared project using one workstation [14]. They take turns in writing code and reviewing code, regularly swapping roles. A review of pair programming in undergraduate education suggests the benefits of pair programming include “increased success rates in introductory courses, increased retention in the major, higher quality software, higher student confidence in solutions… improvement in learning outcomes” and “there is some evidence that women, in particular, benefit from pair programming” [28].

Developing computational thinking is a large area of not only Computer Science Education but other subjects in the curriculum. Computational thinking is “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent” [219]. It is a thought process involving logical reasoning to solve problems. In their teachers’ guide to computational thinking, Csizmadia et al [44] state that computational thinking skills “enable pupils to access parts of the Computing subject content. Importantly, they relate to thinking skills and problem solving across the whole curriculum and through life in general.”

In Computer Science students will learn how to program. **Scaffolding programming tasks** involve supporting students in multiple ways: programming a task together as a class, giving the students part of a program and asking them to complete the remainder and giving them a broken program and asking them to debug it. Scaffolding can help prevent students being overwhelmed by a blank page when learning to program. Scaffolding has been known to support students’ learning [202].

Kokotsaki et al. describes another teaching method: **project-based learning (PBL).** It states students learn through active exploration of real-world challenges and problems [121]. It is based on three constructivist principles: learning is context-specific, learners are involved actively in the learning process, and they achieve their goals through social interactions and the sharing of knowledge and understanding. Project-based learning is a particular type of inquiry-based learning where the context of learning is provided through problems within real-world practices [5] that lead to meaningful learning experiences [223]. What makes PBL stand out is the construction of a final product, a 'concrete artefact' [95] which represents pupils' new understandings, knowledge, and attitudes regarding the issue under investigation—often presented using videos, images, sketches, models, and other artefacts [98].

1.3.6 Computer Science education analysis

In the UK and around the world children of all ages are being taught Computing through topics such as computational thinking, algorithms and coding. Computing is seen as an essential part of children's education. In primary school children learn through **cross-curricular** themes while in secondary they learn in individual Computing lessons. Secondary school teachers typically will have a Computing related qualification. There are established pedagogical methods for teaching Computer Science Education, which many teachers use and adapt.

In the UK the number of students choosing Computing is increasing but has not returned to high levels of students that took ICT. The subject suffers from many misconceptions from children, especially girls, that it is difficult, and one needs to be clever to take it.

Primary school teachers have a strong **support network**, experience of teaching **multiple subjects** in a **cross-curricular** environment. While there are few primary school Computing specialists, data shows that there isn't a **staffing** problem of primary school teachers. The **pedagogy** used to teach Computer Science Education is well suited to primary schools. Primary school students are more motivated than their older counterparts and teaching in primary schools gives the opportunity to challenge perceptions of Computing.

Strong support network

Many UK primary school teachers have a strong support network through Barefoot and CAS. They can borrow physical computing equipment from a local CAS hub, attend regular Computing meetings with other Computing teachers and access free and open-sources resources through both the CAS and Barefoot websites. Beyond the UK, as most primary teachers teach every subject it is easier to find support in their own schools. They are not a department of one subject but a team of teachers teaching every subject together.

Multiple subjects

To teach data science using physical computing a teacher must cover multiple subjects together. Primary school teachers are trained in teaching multiple subjects. They teach their class every subject in the same classroom. A primary school classroom can be a Mathematics classroom, a Science classroom and a Geography classroom. Primary school teachers are experienced at using same resources for multiple subjects.

Cross-curricular

As well as teaching multiple subjects, primary school teachers teach them in a cross-curricular environment. They don't teach Geography then stop teaching Geography and teach mathematics. They teach Geography with Mathematics. This is often done under a theme that the entire school follows. South America may be a theme for one term where they learn about the Amazon Jungle in Geography and use Mathematics to calculate the amount of oxygen the trees produce in one day.

While it may seem that secondary school teachers have more specialist Computing knowledge than primary school teachers, this is not always the case. Not every secondary Computing teacher is a specialist in Computing. Primary school teachers have more experience of teaching multiple subjects at once and may have specialism in a subject within the project, e.g. a project is analysing local plant life with graphs. A primary school teacher may be a Geography specialist and have knowledge of plant life. A secondary Computing teacher may have knowledge of neither.

Staffing

It has been seen from Table 2 that in the UK there is no problem in recruiting primary school teachers. Staffing is not an issue. In secondary schools it is seen that Computing teachers may not be specialists in Computing. If a secondary school teacher is struggling to teach Computing, wanting them to teach Computing with physical computing and data science may be too difficult.

Pedagogy

The five common methods for teaching Computer Science: unplugged type activities, contextualisation of learning, collaborative learning, developing computational thinking, and scaffolding programming tasks have a place in both primary and secondary education.

Unplugged activities are more common in primary school as they are done with younger children, particularly those learning to read. The **contextualisation of learning** can happen more easily in primary school as they are covering multiple subjects at once and can go in depth into the context. **Collaborative learning** is easier in primary school as the children are at the same desk/seats for every subject so have a regular partner. Their partner doesn't change every subject so a collaborative relationship can grow and develop over time**. Computational thinking** skills are taught at primary school first and many schools will start teaching these with unplugged activities. Secondary school Computing lessons almost suffer from being in a computer room. There is no physical space for unplugged activities that work well in computational thinking. **Scaffolding programming task** is important at both levels. Primary school teachers are more likely to need scaffolding and use it more often as the majority of the children are beginners. **Project-Based Learning** is more likely to occur in primary schools where multiple subjects are taught at once in themes.

Student engagement

Students' perceptions of subjects are developed throughout their whole life, but they begin at primary school. It is known that some students in secondary schools have a negative perception of Computing and that younger children also perceive Computing as difficult and "not for girls". Primary school is a time where these perceptions can be developed in a more positive light.

1.4 Analysis

In this chapter physical computing and data science are introduced as two tools before a description of Computer Science Education. In the first section the growth of physical computing in education over the years is analysed and two examples of modern-day physical computing projects are given. This analysis concluded that physical computing is a tool that enables students to be creators of technology, not just consumers. The case studies show how it allows them to create real life tools that can make a difference to their environments. Physical computing tools have evolved from complex electronic breadboards to plug and play addons. Their software has also become more accessible over the years, from bespoke software installed on a PC to free open-source websites accessible on many different types of devices. Physical computing has become more accessible to younger audiences and non-specialist educators. It has become cheaper and more widely available. These advances in physical computing have made it more accessible to more diverse audiences, often underserved by traditional Computer Science education programmes. To many students physical computing has transformed their perception of Computer Science and engaged them in the subject.

In section two, the benefits and challenges of data science are introduced. Here examples of how Data science can be used for social good are seen. Advances in this area have made it easier for data to be collected and analysed quickly and in live time-sensitive environments. How powerful decisions are made using data are also seen, and how data science can create change and action in the world with data. Enormous amounts of data are stored and collected on every individual in society, with and without their knowledge. As it has been stated previously, data can be corrupted, manipulated and misused.

An important question remains about how data science can be included in education. There is a risk of placing it in just one topic, e.g. Mathematics, and alienating people not interested in that topic. There is a risk of only teaching the complicated parts of it, e.g. programming data analysis formulae. There is a risk of being too negative: rather than showing the potential of the area, only showing the dangers and the negative examples of bad data science.

Combining physical computing and data science in the classroom could bring many benefits. In many ways the areas are complementary, physical computing reads inputs in from the world (buttons, sensors, etc), processes and outputs it back into the world (movement, sound, displays). Data science involves collecting data from the world, from people, the environment, analysing it then outputting that analysis back into the world.

Combining these two subjects could be mutually beneficial: using physical computing tools to teach data science in the classroom. i.e. give educators and students the opportunity to create and use physical computing tools to collect data and analyse and visualise it. The data could be local, environmental data that means something to the student and their communities. The analysis and visualisations of this data could give students insight into their local environments and prompt them to create action.

In this thesis the aim is to investigate if combining physical computing tools with data science in the classroom can be achievable and beneficial for both educators and students. To successfully teach data science using physical computing educators need **pragmatic** solutions that will **engage students** but that are also **educationally sound**.

1.5 Research questions

This thesis aims to answer the following top-level research question: Can data science learning be made meaningful for primary school aged children in the context of physical computing?

This question is further broken down into three well defined sub-questions:

1. What range of solutions can be generated to support the **pragmatic** teaching of data science in schools?
2. What physical computing tools are needed to create a **meaningful** and **engaging** end-to-end experience around data science?
3. How do these physical computing tools support **educationally sound** teaching of data science?

1.6 Summary

Physical computing and data science are two relatively new and exciting areas of education, particularly Computer Science Education. They both bring advantages and challenges to the classroom. Physical computing is engaging, exciting and has connections to how real-life technology works. Data science is important, relevant to today's problems and contains powerful tools and methods that society needs to take hold of. This chapter has introduced these fields and put forward how both areas are growing and important to teach to today's children.

The next chapter explores physical computing and data science in more detail. It reviews the practical challenges of these topics in classrooms. It is discovered how **pragmatic** solutions are needed to ensure that any new tools that are created can be introduced seamlessly into schools' infrastructure and be usable by educators and children. The chapter highlights how students **engage** with Computer Science education, physical computing and data science education. Finally, it explores the different **educational sound** learning techniques used.

This study aims to give primary school students a pragmatic, engaging and educationally sound experience of Computing in a real-world data science project by extending physical computing. Through this study and subsequent analysis, the overarching research question "Can data science learning become meaningful for primary school aged children in the context of physical computing?" will be answered.

The contributions of this these are (1) an understanding of how physical computing can become engaging for primary school aged children in the context of data science learning, (2) a new approach for working with teachers in a technology design process and (3) the design process and set of requirements synthesised from research and user interactions.

Contribution 1 An understanding of how physical computing can become engaging for primary school aged children in the context of data science learning.

For contribution number 1, in engaging students, section 2.3.2, how physical computing and data have been independently shown to be engaging through real-life, meaningful and purposeful scenarios, is clarified. In the micro:bit, section 3.1, a specific physical computing tool (the BBC micro:bit) is described and in the analysis, section 3.1.5, how it can be engaging for children. In the requirements summary, section 4.4, there is a specific group of requirements under the heading engaging.

All the requirements are evaluated in Aims and requirements analysis, section 6.7, with a further evaluation of engagement, section 6.7.1. Finally, it is conclude with a summary of this contribution in section 7.1.1.

Contribution 2 A new approach for working with teachers in a technology design process.

For contribution number 2, a new technology design process in methodology, section 4.3, is defined and utilised This new process includes the teachers in a co-design process, whilst drawing on their domain expertise. A design process is a systematic approach used to create solutions to problems or fulfil user needs. Our design process starts with participatory design. Participatory design is an established approach that involves stakeholders as decision-makers through collaboration, mutual learning, and empowerment." [80]. Establishing genuine participation with stakeholders can be difficult. They may be involved in the design process but may lack the technical concepts and language needed to express their opinions [192].

In delivering new technologies to the classroom, it was felt that co-design needed to go further. Therefore, in chapter 4, a new methodology is described: participatory education. It starts with participatory design as explained above but then a further two stages are added. Participatory planning checks the technologies compatibility, involves lesson planning and ensures everything matches what the students are currently learning. Participatory delivery continues the joint/staggered approach of co-design. The researcher delivers the technology lessons to the students in collaboration with the teacher. The two equal stakeholders discuss progress throughout and amend the lessons as required. This new methodology helps teach the materials and answer the research questions. In the evaluation chapter, chapter 6, the evaluation of this thesis includes an evaluation of this new methodology, it is put into the context of this project, its delivery is described and evaluated with quotes from the teachers who experienced it.

Participatory education is defined in section 4.3 Methodology. How it was used in the study is described in section 6.1.1 Methodology application. This section puts the methodology in the context of the main study. The methodology is then evaluated in section 6.1.2 Methodology review, with quotes from the teachers who experienced it. Finally, the thesis concludes with a summary of this contribution in section 7.1.2.

Contribution 3: the design process and set of requirements synthesised from research and user interactions.

For contribution number 3, the requirements came from two areas: research and user interactions. Research is derived from the background studies in chapter 2, the micro:bit description in section 3.1, the literature review in 3.2, and the case study Energy in Schools in section 4.2. The second result of the design process are user interactions: the users were consulted in gathering requirements, section 4.1 and again during hardware design in section 5.1. The resulting table of requirements is presented under Aims and requirements in section 4.4.2. They are evaluated in Aims and requirements analysis in section 6.7. Finally, the thesis concludes with a summary of this contribution in section 7.1.3.

Chapter 2

Background

This chapter takes a deeper look at what physical computing and data science are in detail including their places in education. A selection of physical computing devices will be explored as well as an example data science pipeline.

The first section looks at physical computing in detail, how it is covered in the curriculum, the different types of devices found in schools and how the operation of physical computing can be broken down into inputs, outputs, processing, connections and programming.

The second section before analysis looks at data science education, its place in the curriculum and how it is defined under data collection, analysis, visualisation and outputs.

2.1 Physical computing

Physical computing is a combination of hardware, software and materials that can sense the world around them and/or control output. In the world there is physical energy such as light, temperature and movement that can be captured using **inputs**. This energy is **processed** into data. This data can then be **output** back into the world through outputs such as displays, speakers and motors. Physical computing can be broken down into those three stages: input, processing and output. Before each stage is looked at in detail, physical computing in the curriculum is described below.

2.1.1 In the curriculum

In the UK physical computing is mentioned in the Key Stage 2 Computing curriculum [41]: "design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts." And a similar statement at Key Stage 3 [42]: "design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems." There are also many other parts of the curriculum in which physical computing can be used, for example in Key Stage two: "use sequence, selection, and repetition in programs; work with variables and various forms of input and output" and "select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information." As physical computing is a combination of hardware and software it can be used to cover many of the coding goals found in Computing curriculums.

The National Centre for Computing Education (NCCE) specifically mentions using physical computing in their curriculum map [46], it is a unit in Key Stage 2: "Programming A – Selection in physical computing" and Key Stage 3: "Applying programming skills with physical computing".

2.1.2 Physical computing devices

There are many physical computing devices available in education. From Chapter 1 it is known that Arduino was one of the first electronics board made for education. Raspberry Pi then entered the market aiming their boards at education. The NCCE provide teachers with Continuing Professional Development (CPD) on physical computing. The CPD covers these devices, for which they also provide a lending service (Table 3). In this section the focus is on five of these devices plus Arduino and Raspberry Pi to explain physical computing in education further. The next paragraph summarises the seven devices, how they are powered, programmed and their inputs and outputs.

Key Stage 1	Beebots
Key Stage 2	Data loggers Crumble micro:bit
Key Stage 3	Raspberry Pi Pico

Table 3 Physical computing devices in the different Key Stages

Beebots

Figure 7 Beebot

Beebots, Figure 7[7], are robots in the shape and colour of a bee. They are battery powered with standard AA batteries. They have brightly coloured directional buttons on the top of the bee and a small speaker. On the bottom side are wheels and inside is a motor. To program the Beebot, children press the direction buttons then the green Go button. The robot moves in the order the buttons were pressed. With no external device needed to program and no words on the Beebot it is popular for very young children to use, especially those who are pre-literate. A Beebot costs around £81.59 [15]. They are 7cm tall, 13cm long and 10cm wide. Young students can easily pick them up with two hands. Educators tend to use Beebot in small groups of 3 to 5 children, taking turns entering the instructions.

Data loggers

Figure 8 vu+ primary data logger

There are many commercial data loggers in the market. Pictured in Figure 8[8] is the vu+ primary data logger. It is powered by a built-in rechargeable battery. It has a screen that can display live sensor data for its three in-built sensors: temperature, sound and light. The buttons can be used to stop and start recording to its memory. It has two plugs for two further sensors to be attached, such as a temperature probe which is included in the kit. When plugged into a computer, students can see the data live through its software. They can view live numbers, graphs, meters or pictograms. They can also load the saved data. This sensor and its many predecessors are popular for primary schools to use in science experiments. It is sturdy, lightweight and small. This data logger costs £202.79.

[7] https://www.tts-group.co.uk/bee-bot-programmable-floor-robot/1015268.html
[8] https://www.tts-group.co.uk/vu-primary-data-logger/1030063.htm

Crumble

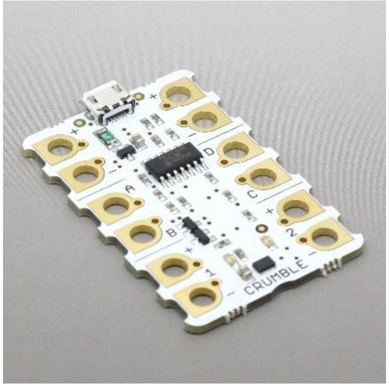


Figure 9 Crumble

Crumble is a microcontroller board, pictured in Figure 9[9]. It is powered over USB and with an AA battery pack. Inputs and outputs can be added to the crumble using crocodile clips. Crumble can be programmed over USB. The programming language for the Crumble is an easy-to-use block language that can be downloaded to a computer. It is very easy to add motors to the Crumble to create programmable robots. A Crumble board cost £12.00, a pair of motors cost £8.22. A starter kit with the board, USB and crocodile cables, battery case and add-ons such as a switch, light sensor, buzzer and RGB LEDs costs £26.40. The Crumble is a flat board measuring 5cmx3cm. Students will generally work in pairs or small teams to code and assemble Crumble projects.

micro:bit

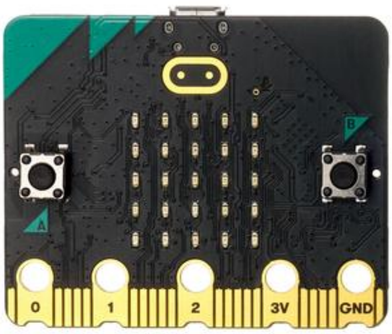


Figure 10 BBC micro:bit

The micro:bit, pictured in Figure 10[10], is a microcontroller. It can be powered over USB and with a battery pack. There are many inputs and outputs embedded into the micro:bit. More inputs and outputs can be added using crocodile clips and by attaching them to the bottom of the micro:bit. It can be programmed using multiple languages from the block-based ones to text-based like JavaScript and Python. The micro:bit can be programmed over USB from a computer or over Bluetooth with a mobile device like a tablet or a mobile phone. With its embedded inputs and outputs, it's very quick and easy to be creative with the micro:bit. A "Go pack" with a micro:bit, battery pack, batteries and USB cable costs £15. A micro:bit on its own costs £12. A micro:bit is a flat board measuring 4cm x 5cm. Students will sometimes work individually with a micro:bit or in pairs or small teams. In 2014 every first-year student in high school in the UK was given a micro:bit [8]. In 2024 a

[9] https://redfernelectronics.co.uk/crumble/
[10] https://microbit.org/new-microbit/

class set of 30 micro:bits were given free to every primary school in the UK that applied [86].

The Microbit Educational Foundation is a not-for-profit organisation that "*aims to inspire every child to create their best digital future*". It was founded in 2016 to continue the micro:bit project internationally.

Raspberry Pi Pico

Figure 11 Raspberry Pi Pico

The Raspberry Pi Pico, pictured in Figure 11[11] is a microcontroller, that can be powered over USB. It is 2cm x 5cm in size. Inputs and outputs can be added to its pins using wires or custom accessories. It can also be programmed over USB using a computer and the text-based language Python. A newer version of the Pico, the PicoW includes Wi-Fi and Bluetooth. Once programmed it can run independently of the computer and control input and output devices.

Raspberry Pi

Raspberry Pi, pictured in Figure 12[12], is a board-level general-purpose device. It is not a microcontroller but a single board computer to which sensors and actuators can be connected directly. It has internal memory to store and run programs. It can run its open operating system and connect to an external monitor. While it's commonly used to run Python programs it can run many different programming languages. The Raspberry Pi is not programmed by another computer. It is a computer. The most recent model is called the Raspberry Pi 5. In 2015 the Raspberry Pi Foundation created the Raspberry Pi Zero, a smaller single board computer, and sold it for $5.

Figure 12 Raspberry Pi

[11] https://www.raspberrypi.com/products/raspberry-pi-pico/
[12] https://www.raspberrypi.org/

Arduino

Figure 13 Arduino

Arduino, pictured in Figure 13[13], is a company that produces different electronic boards, the most common Arduino board is the Arduino Uno. The Uno is a board-level embedded device, it can be powered over USB but there are also battery accessories for it. Inputs and outputs can be added to its pins using wires. It can be programmed by a computer using its IDE in the C programming language. Recently a web block-based environment has been introduced to program Arduino boards. An Arduino Uno 3 board costs £20. There are other versions of Arduino that are smaller and can include Wi-Fi and Bluetooth. As one of the first physical computing devices readily available in schools, Arduino has a lasting popularity especially in Italian schools where it originated.

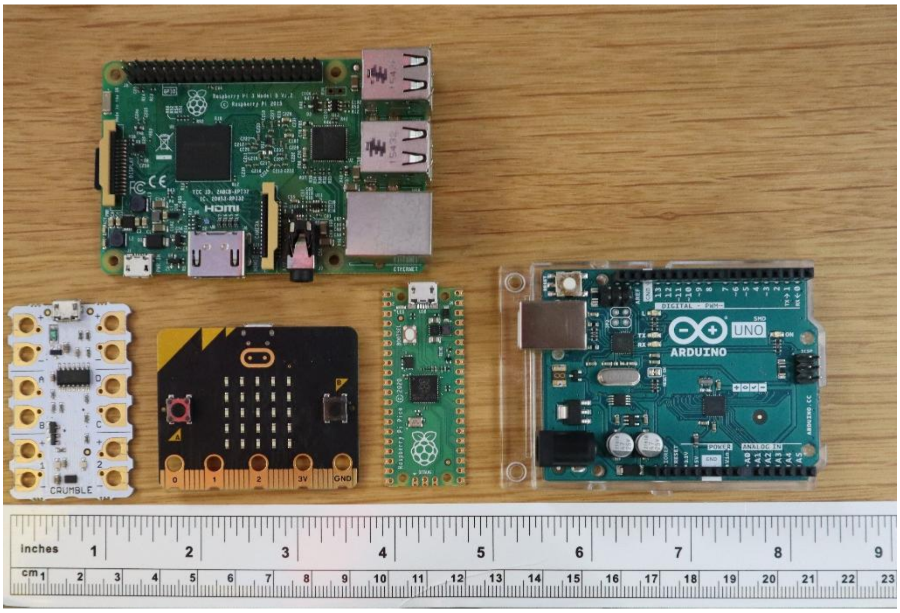


Figure 14 From left to right: Crumble, micro:bit, Raspberry Pi Pico and Arduino Uno. On top: Raspberry Pi

The definition of physical computing devices is that they are *programmable* in some way. Table 4 below summarises how each device is programmed. The microcontrollers are all *mobile* and *battery-operated*. **Transparency** is an interesting feature to analyse. The micro:bit is labelled, it is known what sensors it

[13] https://www.arduino.cc/en/hardware

has on it and the BeeBot also has its arrows clearly marked on its base. However, what it *could* do is transparent and clear but not what it ***is*** doing. Most of the microcontrollers are *lightweight.* The BeeBot is the heaviest but can still be picked up and handled by a small child. The data logger is not quite programmable, students use the buttons to set what sensors they want to record, but they're setting it like an alarm, not programming it. Students cannot be as creative with the data logger compared to the other devices.

Device	**Power**	**Programmable**
Arduino	USB Battery accessories	Block-based, text on PC
Beebot	Batteries	On device
Crumble	Battery pack USB	Block-based on PC
Data logger	Built in battery	Buttons on device
micro:bit	Battery pack USB	Block-based, text, on PC and tablet
Raspberry Pi Pico	Mains USB	Text on PC
Raspberry Pi	Mains USB Battery accessories	Block-based, text on device

Table 4 Comparing power, program and inputs and outputs of common physical computing devices

2.1.3 Device inputs

In the seven devices above inputs are either embedded or can be added to the devices. Inputs are also known as sensors. Examples of sensors are button presses, temperature and light sensors, movement sensors, and sound sensors. These electronic sensors are typically much more sensitive than humans and can also measure minute changes.

Some common sensors found in physical computing projects are buttons, distance sensors such as the passive infrared (PIR) sensor and temperature sensors. These are pictured in the following text and are from the Kitronik micro:bit kit. These are designed to be connected to a microcontroller via a breadboard.

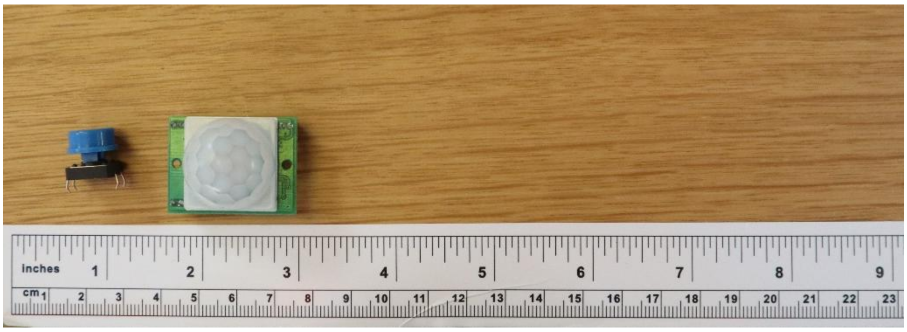


Figure 15 Inputs: a button and a PIR sensor.

The Beebot's and the data logger's inputs are the buttons on it. The data logger also has embedded inputs, its sensors. The crumble, the Raspberry Pi, Raspberry Pi Pico and the Arduino have no embedded sensors, but ones can be added through connections. The micro:bit has buttons, temperature sensor, light sensor, touch sensor, magnetometer, microphone and an accelerometer, all built into the device.

2.1.4 Device outputs

Outputs are also known as actuators. They make some sort of change in the real world and/or communicate information. There are many different categories of actuators, for example:

- visual: a light or LED
- auditory: a speaker or buzzer
- robotics: a motor or several motors that create movement.

Outputs are how physical computing devices interact with the world. These categories are very popular in physical computing education. Many beginner projects start with lighting a single LED on a breadboard. The CamJam EduKit 1[33] is broken down into 8 worksheets. It starts with controlling an LED using code, then using a button. It finishes by controlling an LED and another output: a buzzer. The Kitronik "Inventors Kit for the micro:bit" [13] experiment 1 uses buttons connected to a breadboard to control the micro:bit's built in LEDs.

Figure 16 Common physical computing outputs: a buzzer and different coloured LEDs

2.1.5 Connections between different hardware

Most inputs and outputs must be physically connected to the microcontrollers to work. The microcontrollers (the micro:bit, Arduino, Crumble, Raspberry Pi Pico) and the Raspberry Pi have General Purpose Input Output (GPIO) pins that can be used to connect inputs and outputs to them. There are multiple ways to connect to these pins and different microcontrollers facilitate these connections in different ways.

Headers

Headers are electric connectors consisting of one of more rows of pins. They are often used on PCBs for making connections. There are two types of headers: male (pin) and female (socket). The Arduino Uno has female headers; the Raspberry Pi has male headers. The Raspberry Pi Zero and the Pico come without headers fitted so the user can choose which one to add on by soldering them to the board. A jumper wire is connected to the header, and the other end can be connected to a sensor or an output on a breadboard. A breadboard is a solderless plugboard that is reusable for prototyping in electronics. Multiple components can be added to a breadboard to create a circuit of electronics with a microcontroller controlling them. Figure 17 shows a Raspberry Pi connected to a breadboard which has a button connected to it.

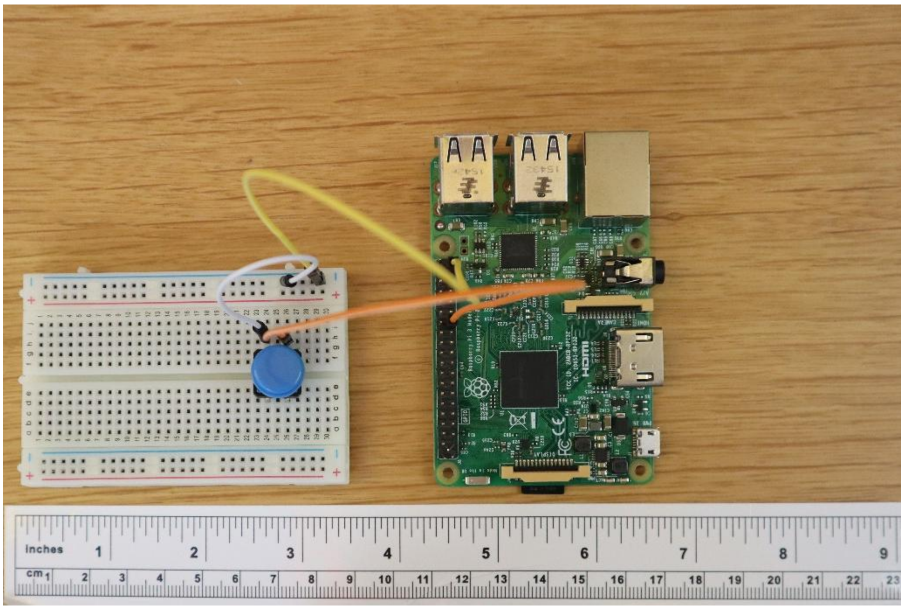


Figure 17 A Raspberry Pi connected to a breadboard

HATS and shields

In 2014 Raspberry Pi coined the term “HAT” (Hardware Attached on Top). The original HAT was an add-on board for the Raspberry Pi B+. It attached onto the pins of the Raspberry Pi and added extra features. Companies use the HAT design to make their own add-on boards for Raspberry Pi. Arduino calls these add on accessories shields. HATS and shields are easy ways to connect extra inputs and outputs to physical computing devices without the need for wires or a breadboard. The disadvantage is many hats/shields cover all of the GPIO pins so other accessories cannot be added.

Crocodile clips

The micro:bit and the Crumble’s GPIOs can be accessed using crocodile clips through holes on the board. As well as GPIO the two boards provide access to ground and power through these holes. Figure 18 shows a micro:bit connected to an arcade button using crocodile clips. The clips are connected to pin 0 and ground on the edge connector of the micro:bit. Crocodile clips are also known as alligator clips and come in different lengths. Some Science classrooms will have crocodile clips for experiments.

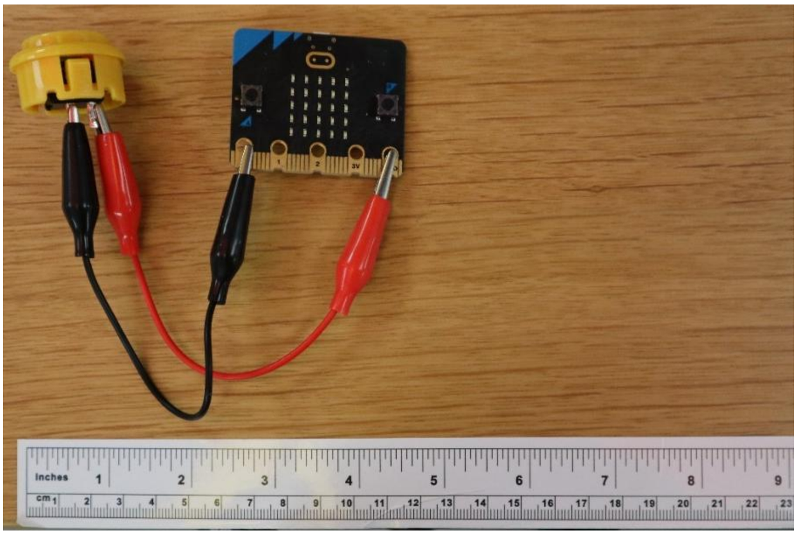


Figure 18 A micro:bit connected to a button using crocodile clips

micro:bit Edge connector

The edge of the micro:bit was designed to have accessories attached on without soldering or any extra wires. These accessories don't cover the entire board like a hat or shield, the LEDs and buttons of the micro:bit are still accessible and visible (see the micro:bit with an OLED screen on the left of Figure 19 below versus a Raspberry Pi with an accessory of RGB LEDs on the right). The micro:bit's buttons and LEDs are still visible. The Raspberry Pi doesn't have embedded buttons or LEDs but now there are no pins left to attach any.

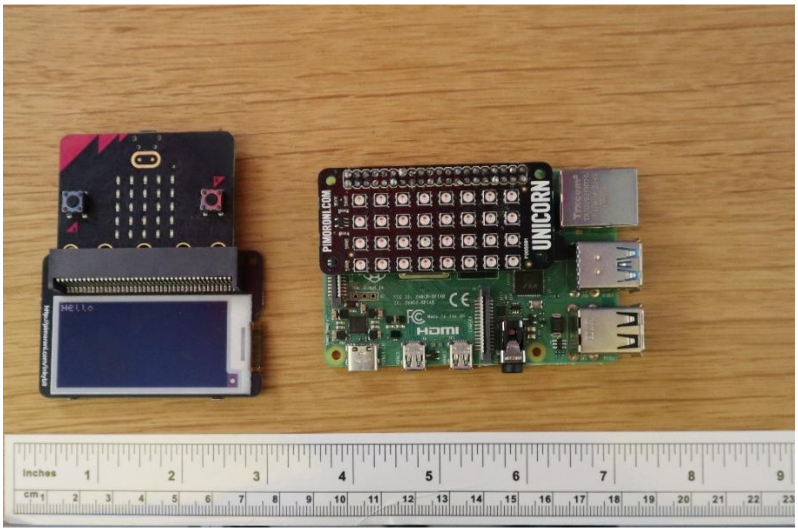


Figure 19 Left: a micro:bit with an accessory attached to its Edge connector. Right: A Raspberry Pi with a hat attached

2.1.6 Controlling inputs and outputs

Connecting these inputs and outputs together is quite difficult. Modern computers are not designed to connect directly to these types of sensors. What is typically used is a microcontroller: a bridging device that can connect sensors and actuators. Microcontrollers are small, simple, low power devices designed to integrate with inputs and outputs. A microcontroller allows us to program what these inputs and outputs should do. Hodges et al. [96] categorised the different types of devices found in education. The seven chosen devices fit broadly across these categories:

Categorization	Example
Packaged electronics; no programming	Data logger
Packaged programmable products (not boards); programmable via PC or phone; often battery-powered	Beebot
Board-level embedded devices; need PC to program but can operate as standalone device; can be battery powered	micro:bit Crumble Arduino Raspberry Pi Pico
Board-level general-purpose devices; often use wired power	Raspberry Pi

Table 5 Categorization of physical computing devices

Packaged programmable products are broken down into two types of products: robot turtles just like the robot turtle from the 1970s and programmable construction sets, like Lego and Vex robotics. The Beebot belongs in the category of robot turtle. The micro:bit, Crumble and Arduino belong in board-level embedded devices. They need a PC to program them but can operate on their own once powered. This category from Hodges et al. is broken down into further subcategories:

- Microcontroller boards with integrated input/ output devices (micro:bit)
- Microcontroller boards with low level input/output (Crumble)
- Microcontroller boards with support for modular input/output (Arduino, Pico)

2.1.7 Programming physical computing

There are multiple ways to program boards and microcontrollers with their connected input and output devices. Some devices allow code to be written in multiple languages. Physical computing programming languages can be placed into these groups:

- Tangible programming languages
- Tile based graphical languages.
- Block based graphical languages.
- Text based languages.

Tangible programming languages are those that involve the physical construction of the program. The children physically snap together pieces of hardware to create a program that does something. Physical computing devices that use tangible programming languages are the Codeapillar [78] , Littlebits[133] and Cubetto [45]. These devices can be used by young children, usually pre-literacy as they contain very few words. They are using physicality to teach logic, patterns and sometimes loops.

Tile based graphical languages include Kodu[120], Scratch jnr [183], and MicroCode [148]. These are digital on-screen languages that again use very few words and are aimed at younger children. They are normally structured to replicate how children learn to read: top to bottom and left to right. MicroCode can be used to code the micro:bit.

Block-based graphical languages are a digital on-screen language such as Scratch [182] or MakeCode [149]. In general, the blocks can be placed anywhere on the screen. Blocks can be placed inside other blocks that will change size. There is no order of where the blocks should go, though a block cannot go inside another block that doesn't "fit", i.e. MakeCode doesn't allow incorrect programs. All these languages do not let the children create programs that will not run. MakeCode can be used to code multiple microcontrollers such as the micro:bit, Lego Mindstorm [129] robots and the Circuit Playground Express [161].

Text based languages are the final step for any Computer Science student, popular languages in education are JavaScript, Python and C. Many secondary schools in the UK will teach Python as their main and sometimes only programming language. There are multiple ways to code text-based languages: using a plain text editor like Notepad++, using large code editors like Visual Studio or Atom that can help support students coding with features like syntax highlighting and autocomplete. These editors can also spot errors in the code and highlight them, just like a word processing editor highlights spelling mistakes. The editors support projects with multiple files and libraries, and they help manage code by supporting services such as GitHub. GitHub is a developer platform where developers can create, store, manage and share their code. As they "push" changes to GitHub it keeps a record of these changes so students can look back at old code, undo changes and restore old code from any point in the past. They can also "pull" the code down to different devices, so they are always working on the latest version of the code. GitHub is normally introduced in university level Computer Science courses but can be used at secondary level as it promotes a lot of good practices in Computing.

Python has its own IDE called IDLE that also supports syntax highlighting and running the programs in the editor. It doesn't have error spotting or GitHub integration like the code editors, but many secondary schools find it adequate for teaching children how to code. Arduino also has its own IDE that schools can install to program the Arduino boards.

Unfortunately, many of these editors cannot be used to program physical computing devices. Students can learn to program using them but need to use different software to program a physical computing device that uses text-based languages like the Raspberry Pi, Arduino and micro:bit. There is a simple Python editor called thonny [201] which does connect directly to the Raspberry Pi Pico. The Raspberry Pi Foundation and the micro:bit Educational Foundation both have web-based Python editors [40, 146]. The Raspberry Pi editor is currently in beta; it is embedded into their learning resources for students to use as they learn but doesn't connect to any of their devices. The micro:bit editor can be connected to the micro:bit device and the code downloaded directly to it.

2.1.8 Physical computing analysis

Physical computing has become easier to teach to younger children. The BeeBot can be coded with limited dexterity and low literacy which is suited to children at the beginning of their school journey who have yet to develop those skills. They can still learn some of the basics of Computing like *logic* and *debugging*. The BeeBot and micro:bit are **transparent**, particularly the Beebot. The arrows on its body make it clear **what** it can do and how to use it. The micro:bit's labels make it clear what it

can do. Many of these boards: the Beebot, micro:bit, Crumble and Pico are very *sturdy*. They can handle the harsh environment of a classroom of children as well as being somewhat waterproof. It's only when accessories are added that the devices become fragile and breakable. The Raspberry Pi in particular is not typically a mobile device; it is normally powered by mains power and plugged into a screen and monitor to code. It is not meant to be picked up or handled by children.

The variety of devices available in schools also makes it easy for students to **progress** in Computing as they age. They can start with no screen devices like the Beebot, move onto block-based devices like the Crumble and micro:bit then onto text-based programming on the micro:bit or Raspberry Pi/Arduino.

The variety of devices and their uses also makes it easier for physical computing to be taught within **multiple subjects** in the curriculum, not just Computing. A popular BeeBot lesson teaches children Geography by putting a compass on the ground and asking the students to head in a particular direction. They can also add maps of their village and navigate from the school to the local playground, learning about navigation and their local area as well as coding the logical steps to complete the task.

Physical computing devices that include outputs allow students to be *creative*. They can create lights, noise and sound that react to something in the real world. The small size of these boards also makes it easier to embed them into larger projects, for example adding the micro:bit to a doll and programming its lights as its face: it smiles when the light sensor detects light and it's sad when the light detector doesn't detect light.

Adding breadboards, LEDs and buttons to a device can make a Computing lesson feel more like an Electronics lesson. This can overwhelm both the student and the teacher as they are now teaching multiple subjects: Electronics, Computing and the third subject the task is based in. For example, the task above is based in Geography learning about maps, compasses and modelling their town . For younger children, like primary aged children, and inexperienced teachers it is better to keep physical computing simple with fewer accessories or embedded inputs and outputs. This avoids the challenges of buying and storing extra components. It also saves time in the classroom: students connecting these components onto a breadboard can cause a lot of problems. They can break the components easily, drop them or lose them. Students can also connect them into the wrong pin on the device or in the wrong hole on the breadboard. Wiring up a component like an LED without a resistor can cause damage and stop working. Debugging problems like these can take up a lot of the students' and the teachers' time. Jin states that "*Working with content from different contexts, e.g., hardware and software, can place a high level of demand on the learner's cognitive system*" [110].

Using crocodile clips with the Crumble or the micro:bit can make it easier to assemble circuits and avoid using a breadboard, however the clips themselves can be difficult for younger children to open and close. They are also not a stable connection, the crocodile clips can move, and the input/output can stop working.

With sensors and outputs embedded into the board, microcontrollers like the micro:bit make it easier for educators to use physical computing in the classroom. They don't *have* to connect anything using breadboards, wires or crocodile clips to demonstrate inputs and outputs as they are already embedded in the board. In the next chapter the micro:bit and its accessories are looked at in closer detail.

2.2 Data science education

Data science is an interdisciplinary field often found within Mathematics, Business and Computing. A data science process or pipeline refers to the "process and tools used to gather raw data from multiple sources, analyse it, and present the results in an understandable format" [100]. An example data science pipeline by Berman et al [4] is detailed in Figure 20 with the stages Acquire, Clean, Use, Publish and Destroy.

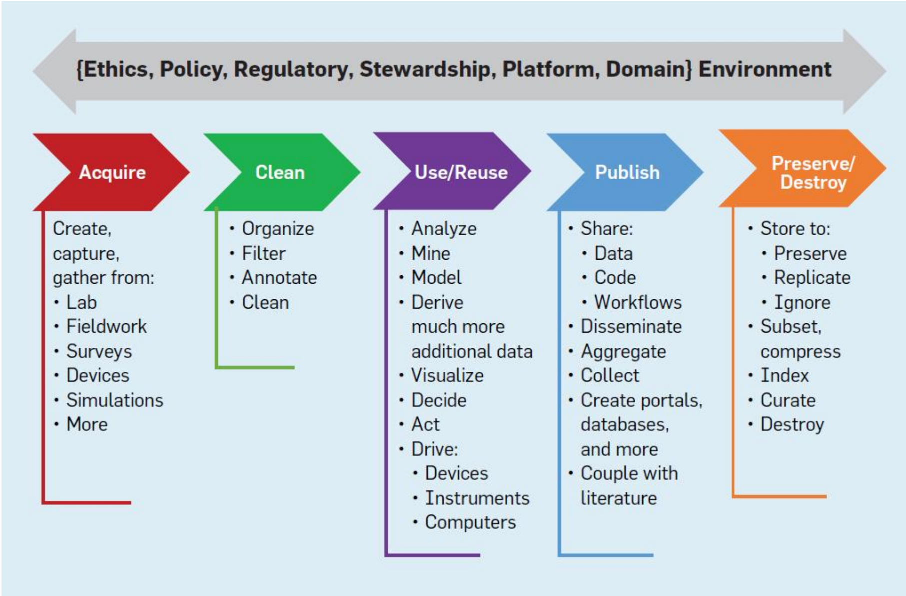


Figure 20 The data life pipeline by Berman et al.

This pipeline describes how data taken from the real world (which may be messy and need interpreting) is processed, analysed and used to answer a real-world question that is then fed back out into the physical environment.

This section looks at each of these stages simplified into collection, cleaning, analysis and output. The final section looks at data science education in the UK national curriculum before analysis.

2.2.1 Data collection

Data collection is the often seen as the first stage of a data science life cycle: "Acquire" in the diagram above. However, before collection can happen an array of decisions need to be made about what data is needed, how it will be collected, when, where, how much and how often. There are many definitions of what data is. Kitchin describes it as "a collection of raw materials that are created by abstracting and capturing the world into representational forms like numbers, images, sounds, and characters" [116].

Data collection can be done in different ways:

- Automatic sensors can collect data like temperature, energy use.
- Manual data collection can occur through in-person surveys.
- Data can be gathered from external sources, such as weather data from a website.

Data can arrive into a system in many ways and in different formats. It isn't always new data: it may be historical data like weather patterns over the last few months in a certain location. Before data collection is done, data scientists must consider what it is they're trying to analyse, what information are they trying to create from the data. This can help ensure data collection is done correctly and accurately. As Kitchin stated, data is a representation of the world. Information is usable output from processing and analysing that data. The data pipeline is a journey that data can take to become information.

Data Privacy

Since the introduction of GDPR in Europe in 2016 schools have a legal responsibility to store their students' data safely and securely. In the UK schools are defined as a public authority, they are required by law to employ or assign a Data Protection Officer. The schools' inspection body Ofsted includes data protection compliance within their inspection criteria.

Using physical computing devices that can store data can create problems for schools trying to adhere to these strict data laws. This is particularly true if a physical computing device is small and mobile and can be carried out of the school with a battery and all its data unencrypted. Schools need to ensure that students personal data is only accessible by authorised personnel.

Ethics and Privacy

When collecting and storing data schools need to consider the ethical implications of their actions. Recording something like students' favourite breakfast cereal may seem harmless but this can be a sensitive area for children who don't eat breakfast or have allergies. Recording personal data like food preferences, height and especially protected characteristics like race or religion can open students up to bullying from their peers if they do not have the same preferences or characteristics as the majority of the class. In the UK, the Equality Act 2010 [75] protects people from discrimination, harassment, and victimization based on certain characteristics like race and religion. Educators need to consider the ethical consequences of collecting, storing and analysing this kind of data from children.

When recording data, it is common for student data (name, class, age) to be recorded too. This is private identifiable information that the school must secure. In many countries it is illegal to share this data with companies or individuals outside of the school without the parents' permission. Schools need to know where in the world the data is being stored and who has access to it. Ensuring the identifiable data is secure places a further burden on the educators.

2.2.2 Data cleaning

Once the data has been collected, it can go through other stages before analysis. One stage is generally known as data processing or data cleaning. There are many reasons why data needs to be "cleaned" before it can be analysed.

Filtering

Users may want to filter data to a specific set: for example, a weather station measures wind, temperature and humidity but the data analysis is only using temperature, so the data collected is filtered to temperature only.

Labelling

As data arrives into a system it might need to be labelled or annotated. A sensor may just be sending a stream of numbers, and the system needs to ensure this data is correctly categorised, e.g. a sound sensor in Classroom 1 on Monday reporting the noise level every hour needs to be stored as something that makes sense. The value it comes into the system as could just be "Sensor1". In Table 6 below a difference between the data in row 1 and 2 cannot be seen.

Sensor	Value
Sensor1	22
Sensor1	22
Sensor3	20

Table 6 Data as it arrives into a system.

The difference between row 1 and row 2 is that they occurred at different times. Row 3 is coming from Sensor3 but it's in Classroom 2. The users must plan what other data needs to be stored alongside the first set of data so they can retrieve it later. Values like date and time and location can be added to the data as it enters the system, see Table 7.

Sensor	Classroom	Date	Time	Value
Sensor1	Classroom1	23/9/2024	09:00	22
Sensor1	Classroom1	23/9/2024	10:00	22
Sensor3	Classroom2	23/9/2024	09:00	20

Table 7 Data after labelling

Data may arrive into a system after the date it was collected, therefore the users need to edit the data or the system to ensure its accuracy. The scenario above is an example, not an absolute truth. Some sensors may record all the necessary data and not need this extra metadata.

Data errors

Collecting data from humans can be prone to errors. Misunderstanding the questions, malicious intent and human error can all lead to the "wrong" data being recorded. Sensors can also malfunction when they are not calibrated or installed incorrectly. For example, a sound sensor next to a window may not be recording the accurate sound level of the room, but the noise outside the room. How this data is "cleaned" can be an ethical dilemma for data scientists. Who decides what is "wrong" data? What is done with this "wrong" data, is it deleted, is it edited? If the data that has been collected is changed, is the analysis still going to be valid? Can decisions be made on data that may be incomplete or skewed? These are all excellent questions for students to consider and answer when dealing with data.

2.2.3 Data analysis

Data analysis is done after the data has been cleaned or processed. This is where the data scientists try to find the answer to the question(s) they are asking. In this example of gathering sound level data, students are asking: what room is the loudest room in the school?

In data analysis multiple sources of data can be analysed, data can be filtered and sorted for analysis. Analysis is not always about mathematical calculations. In the example above, the students could take the sound level at different points in the classrooms, the sound level outside the rooms and another sensor could determine when the classroom doors are open to determine if the noise outside is from the classroom. The students could sort their data sets by sound level, choosing to look at the loudest classroom. They could also use filtering to analyse different classrooms. When they determined the loudest classroom overall, they could look at different dates – which classroom is the loudest on Mondays?

2.2.4 Data outputs

This final stage of a data science pipeline is described as “Publish” where the results and insights from the study are presented. Just like data collection, it can be undertaken in many different ways.

Graphs and charts

Mathematical graphs and charts can be created to describe the information that has been created. Line graphs can be used to compare different data sources of the same type, e.g. temperature in different classrooms. The graphs can be longitudinal over time. Bar graphs can be used to take a snapshot of a moment in time, for example comparing all of the classrooms’ temperature on the hottest day of the year.

Data visualisation

Another form of data visualisation besides charts uses images to represent data. An example could be a thermometer with the temperature scale on it or if students are recording noise levels, an image of a speaker could be used. The louder the sound level, the bigger the image. Digital twins can be used to visualise data. A digital twin is “*a virtual model of a physical object, system, or process that uses data to mimic its real-world counterpart*” [56]. Some digital twins use real-time data to create a live interactive display.

Another form of visualisation is data physicalisation, “*a physical artifact whose geometry or material properties encode data*” [108]. These go beyond 2d visualizations and can create an immersive experience for people to understand data more. Students could use physical computing to create data physicalisations, to make their data physical and tangible.

The output is an important part of the data science pipeline. It is invaluable for students to come to their own conclusions using data. It strengthens the students understanding and their confidence in the subject if they can use their own analysis to make conclusions. The answer is not given to them; it is not implied or a foregone conclusion. It is new information that the students have created by following part of or all of a data science pipeline.

The constructivist learning theory emphasises that individuals learn through building their own knowledge. Collecting and analysing their own data to come to their own conclusions is a form of constructivism. It is empowering for students to construct knowledge, rather than passively taking it in.

2.2.5 In the curriculum

As data science is an inter-disciplinary field it is often covered across the education curriculum. There is no specific data science subject though recently some countries and organisations have developed ones [7, 20]. It is often used as a tool in multiple subjects. As Moon et al state, the "*influence of data science in academia, industry and society is leading to its growing presence in primary and secondary education*" [28].

In the UK the Mathematics primary school curriculum [16] states students "*... should also understand the cycle of collecting, presenting and analysing data*" and "*interpret and present data using bar charts, pictograms and tables*". This is carried into the science curriculum: "*They should also apply their mathematical knowledge to their understanding of science, including collecting, presenting and analysing data*" and "*gathering and recording data to help in answering questions*". Computing also touches on many aspects of data science: "*undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users*" and Geography "*collect, analyse and communicate with a range of data gathered through experiences of fieldwork that deepen their understanding of geographical processes*." A summary of these subjects and requirements are detailed below in Table 8.

Subject	Data science requirement
Maths	• Understand the cycle of collecting, presenting and analysing data. • Interpret and present data using bar charts, pictograms and tables.
Science	• They should also apply their mathematical knowledge to their understanding of science, including collecting, presenting and analysing data. • Gathering and recording data to help in answering questions.
Geography	• Collect, analyse and communicate with a range of data gathered through experiences of fieldwork that deepen their understanding of geographical processes.
Computing	• Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users.

Table 8 Subject and their data science requirement

In the NCCE curriculum map, they have a content taxonomy with ten strands, one of which is "data and information (DI)". In Key Stage 2 a unit name is "data and information – data logging" [48] and the learning objectives are:

- To explain that data gathered over time can be used to answer questions.
- To use a digital device to collect data automatically.
- To explain that a data logger collects 'data points' from sensors over time.
- To recognise how a computer can help us analyse data.
- To identify the data needed to answer questions.
- To use data from sensors to answer questions.

In secondary schools there are similar objectives in the same subjects [15].

Data can also be treated when the Internet of Things (IoT) is taught. The Internet of Things is a network of interconnected devices that automatically gather data and exchange it over the Internet, without human intervention. These devices are sometimes called “smart” devices. In the Key Stage 3 curriculum, in the networks unit, the NCCE include a lesson on Internet services [105]. While the lesson doesn’t specifically mention data it does state a learning objective as: “Explain the term ‘connectivity’ as the capacity for connected devices (‘Internet of Things’) to collect and share information about me with or without my knowledge (including microphones, cameras, and geolocation)”.

In 2020 the UK government released a framework “to equip children and young people for digital life” called “Education for a connected world” [208]. It was developed by the UK Council for Child Internet Safety (UKCCIS) Education Working Group, a group of more than 200 organisations from government, industry, law, academia and charity working together to help keep children safe online. The framework is used by the Department of Education in their guide “Teaching online safety in schools – Guidance supporting schools to teach their pupils how to stay safe online, within new and existing subjects” [197]. Several sections touch on data, for example “managing online information” and “privacy and security”.

2.2.6 Data science education analysis

Some aspects of a data science pipeline are prevalent in multiple subjects across the curriculum in schools, mainly the analysis stage in Mathematics. There are many other aspects that can be applied to other subjects. A project around collecting and analysing data can be easily made **cross-curricular**. In a Chapter 4 case study, students learnt about energy consumption as well as local temperature data leading to discussions around climate change and climate action.

In practical subjects such as Geography and Mathematics there is the opportunity to discuss **ethics** and **privacy** using a data science pipeline. These important topics need to be included in their curriculum. It should not only be about collecting and analysing data for Mathematics but questioning if it is right to collect this data, who it belongs to, where it is being stored and how secure it is. All these questions can be added to a lesson around data collection.

Following a data science pipeline all the way through is a method educators can use to teach students about the different aspects of data science. The structure of the pipeline lends itself to **progression** in education. Younger students could do simple data collection using manual methods or by installing and managing sensors. Older students could do the data cleaning and analysis. Both sets of students could create simple or more complex Mathematical visualisations, in line with their current Mathematics progression. For example, children in Year 2 in the UK learn about tally charts and pictograms, students in Year 4 learn about bar graphs and pie charts, with students in the final upper years of 5 and 6 learning about line graphs. A data science pipeline could be adapted to allow these different ages of students to visualise the data with the method they are learning in their Mathematics lessons.

2.3 Background Analysis

Physical computing is an established and proven way of teaching engaging and creative lessons across multiple subjects, though it is mostly associated with Computing. Data science education is another area seen in multiple subjects in schools that can teach students about real-life scenarios but is often subsumed into Mathematics. Both topics have great potential to be cross-curricular areas that relate to real-world phenomena.

To successfully teach data science using physical computing educators need **pragmatic** solutions that will **engage students** but are also **educationally sound**. This analysis investigates the ways a physical computing solution could achieve this.

2.3.1 Pragmatic solutions

With the lack of qualified Computing staff and technical support, pragmatic solutions are needed that can work in the wide variety of primary schools, big and small, urban and rural, with and without dedicated IT technical staff.

Cost effective.

A subject like Computing has high setup and ongoing maintenance costs that leaves little room for new technologies. Any new piece of technology needs funded staff hours to setup and maintain it. Technology also needs replacing every few years as it ages and becomes unusable. A technician shared across several schools in a large geographical area may visit a school purely for maintenance of the school's essential computer systems and not to install or debug any new equipment or software used in the curriculum.

Utilising an existing resource that schools already have access to would be beneficial for schools dealing with tight budgets. Using open-source tools that don't have subscriptions linked to them can also help schools manage costs. Finally, tools that can be shared amongst students and classes easily can help keep costs down and promote adoption.

Accessibility

There are multiple ways that solutions can be accessible: accessible to use for novices, to install, and maintain and accessible to students.

Due to the low numbers of specialist Computing teachers, any new tools need to be accessible to a range of educators with varying skill levels. Setting up and maintaining new tools and software can require expert knowledge. Even if schools can afford a technician, the technician might not have the skills to manage the new tools or software that need to be installed and setup alongside existing systems.

Many data science tools are professional level tools used by experts in industry. Educators are often not trained in these professional software tools. This is particularly true at primary level where even tools like Microsoft Excel can be unfamiliar to educators. Keeping tools simple and familiar can help support educators.

It is the case that educators do not always have administrative access to school systems. Often, they do not have direct access to the firewall to add websites to it for students to access. They must request access via an external company or a technician. This applies to data coming in and out of the school via websites and Application Programming Interfaces (API). Educators often do not have administrative access to the computers or tablets that the students use. They cannot install new software on these devices. They also cannot install new hardware, as any new sensors or devices that need to be added to a computer needs specialist administrative access.

While there has been a recent surge in the number of plug and play physical computing devices, this wasn't always the case. Many popular physical computing devices like Arduino and Raspberry Pi were not easy to set up in a school environment. Arduino needs specific software installing and access to the USB port for a student to program it. And until very recently it could only be programmed in the C programming language: not one that is regularly taught in schools, especially not primary schools. The IDE used to code an Arduino would need to be installed on the school computers, updated and any new libraries downloaded if needed.

Raspberry Pi is a standalone computer with its own operating system requiring a keyboard, monitor and mouse to function. Its operating system also needs to be downloaded to an SD card for each computer. Any updates need to be done over Ethernet or Wi-Fi, which can be difficult for a school to manage.

Many schools do not permit connecting devices like the Raspberry Pi directly to their Wi-Fi. This often goes against their school policy of controlling students' Internet access. Schools have a duty of care to their students to provide a secure and safe environment; this includes their online environment. Schools will often have custom firewalls to prevent students from accessing illegal or inappropriate content. The firewalls also prevent viruses from infecting the school network. Devices like the Raspberry Pi can often work around these firewalls. It takes extra time and skills to ensure they conform to schools' Internet safety and security policies.

Another way tools can be accessible is by being transparent in what they are doing/can do.

Student accessibility

A very important area of accessibility that needs to be ensured is student access. Devices and software need to be age appropriate. While a mobile phone may seem like the ideal technical solution to many scenarios it is not appropriate (or cost-effective) for young children. Mobile phones can be too small to be shared successfully between students. As young children's literacy is still developing, they can struggle to read small text on a mobile screen, particularly outdoors in daylight. Software tools used every day by adults are not always suitable for young children. Overly complex tools can lead to confusion and disengaged students.

2.3.2 Engaging students

Physical computing has been known to increase engagement, creativity and intrinsic motivation, particularly in underrepresented groups like girls.

Studies agree that physical computing brings many benefits to the classroom, such as collaborative learning [99, 136], engagement [140], increasing students' confidence (particularly girls) [188] and more. Hodges et al. [97] noted that the benefits are not just in Computer Science education, but that physical computing creates "diverse connections" to other subjects.

Using physical computing to teach data science could engage students and increase their confidence in the subject. Many students believe Computing is difficult and that they need high grades to take it. Giving them a strong understanding of the subject at an early age could help build their confidence and change their perceptions. More ways to engage students is to situate the learning in **real-life**, the learning must have **purpose** for the student, and it should be **meaningful**.

Many studies of data science education in schools agree that using a dataset that is real and relevant to the students has a greater impact on them. A recent journal paper analysing 25 data science tools advocates for live and relevant data because it allows novices to "*explore trends and answer questions they are interested in*" and makes for a "*compelling introductory data science learning experience*" [106]. Wolff et al. [222] report that young children "*have the ability to work with complex data sets if they are supported in the right way and if the tasks are grounded in a real-life context*" and support the use of "*real [and] complex urban data sets*" in the classroom. Edwards et al. [63] argue that this sense of relevance can be achieved by using geographically local data. They highlight how "*teaching through local environments can contribute to understanding of global environmental issues and active participation locally and globally.*" In developing a high school-level qualification in data science Farrell et al. [76] also recommend content which emphasises "*real-world applications and societal implications*". As Charlton et al. [35] state "*Authenticity is key to engagement and ownership... Students engage when they see **purpose**, when their learning has meaning.*" In 2018, the Royal Society reported on the extent to which data science exists in the primary and secondary UK curriculum [166]. They identified opportunities for integrating data science in primary schools by "*Using **meaningful data** in Computing when applying technology in the analysis and presentation of data – linked to other areas of the curriculum*" and creating enrichment "*through informative infographics and data visualisation.*"

2.3.3 Educationally sound teaching and learning

Physical computing is an established area of Computer Science education. Many of the teaching strategies discussed in this chapter would work well in a study using physical computing to teach data science. **Contextualisation of learning** is a strong strategy in this topic. For example, using a physical computing device to collect data about bees is situated in Computing and Geography. The students are collecting data about the bees in their local environment, not bees in another country or bees from 10 years ago. This data is live, it's current and it's relevant to them and the environment they live in.

Collaborative learning can happen with physical computing as devices are generally shared. Having students collect data together can help further validate the data. The students can agree on the data as they collect it, if there are any doubts about how to classify an item, the students can discuss it together.

As children go through the data science pipeline their **computational thinking** skills will develop. **Scaffolding programming tasks** can happen when students are preparing the data collection tools. Some of the tools may need special libraries, specific code that may be beyond the understanding of primary school children. Teachers can choose which code to abstract and which code they want the students to view, edit and create.

Progression is another aspect of education that is important to teachers, particularly primary school teachers. If multiple year groups could use the same technology, but differently to suit their learning goals this would not only be cost effective and efficient it would be easy for primary school teachers to support each other.

Teaching physical computing with data science could be done **through project-based learning**. Students could learn through problems within real-world practices by collecting data about their real-world. The meaningful learning experiences will come from their analysis of that data, and the construction of the final product can be their results and/or visualisations of their analysis.

Using physical computing to teach data science is **cross-curricular** as it is combining areas of Computing and Mathematics, but it also has the potential to add more subjects through the topic they are studying, e.g. energy use and Geography.

2.4 Conclusion

This chapter has provided a thorough background of physical computing and data science and their applications in education. It has highlighted the reasons why any innovation in this area needs to be pragmatic and how both areas combined can be engaging and educationally sound for both teachers and students to experience.

Of the physical computing devices looked at in this chapter the author believes that the BBC micro:bit has the most potential to be used as a tool for teaching data science to primary school students. The next chapter will look at the micro:bit in depth and conduct a literature review of the use of micro:bit in research.

Chapter 3

Related Work

In Chapter 2 physical computing devices were reviewed. One device that stood out as a potential data collection device for teaching data science is the BBC micro:bit. The first section of this chapter takes a closer look at the micro:bit. It focuses on the micro:bit hardware, the sensors it has on board, the software, how children code the micro:bit, and the tools they use. Finally, it looks at how the micro:bit delivers the needs detailed at the end of Chapter 2 background analysis: **pragmatic** solutions that will **engage students** but are also **educationally sound**.

The second section in this chapter is a literature review of the micro:bit in research. A thorough review of three main publishing venues ACM, IEEE and Springer is undertaken.

3.1 The micro:bit

Looking at the micro:bit in depth can explore how apt it is as a data science tool. One of its potential functions is as a data collection tool, as discussed in Chapter 2. Here this potential is explored and the existing features of the micro:bit that help with data science, and those which are missing, are discussed.

The micro:bit was first launched in 2016 by the UK national broadcaster (the BBC) and 29 other partners [7]. The initial project provided approximately 800,000 micro:bits to 11/12-year-old UK children in their first year of high school. Since then, the micro:bit has grown into a worldwide product, with 10 million sold in over 60 countries, and an estimated 56 million students who have learned with the micro:bit [34].

3.1.1 micro:bit hardware

The micro:bit has hardware inputs and outputs embedded into it. External hardware such as accessories can also be added onto the micro:bit. This section reviews the embedded hardware before exploring the different types of accessories developed commercially for the micro:bit.

Embedded hardware

The micro:bit has *inputs* and *outputs* embedded into its printed circuit board. Specifically, it has: two physical buttons (1), one touch button in the logo on the front (2), 25 pins (3), a mems microphone (10), a light sensor (5), a temperature sensor (6), 3-axis compass (7) and a 3-axis accelerometer (8). Furthermore, it has the following outputs: 25 red LEDs that act as a screen (9) and a speaker (10). It also has a 2.4GHz radio antenna (11) for sending and receiving messages to other micro:bits and devices. Finally, the micro:bit has a JST battery connector for attaching its 2xAAA battery pack (13) and a micro-USB connector (12) which can power and program the micro:bit.

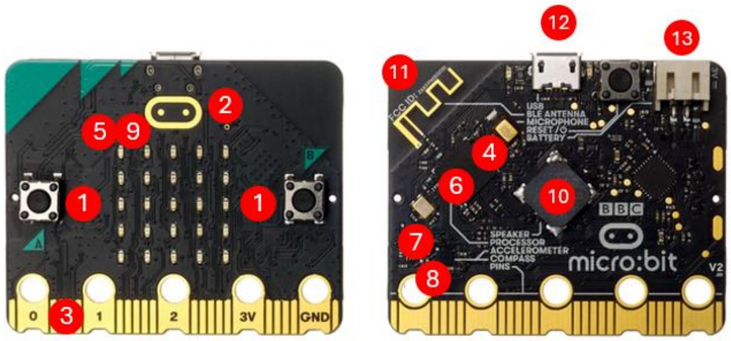


Figure 21 The BBC micro:bit front and back

Hardware accessories

Further inputs and outputs can be added to the micro:bit in multiple ways. As described in 2.2.6, the micro:bit has an Edge Connector which allows it to be slotted into accessories. The micro:bit Educational Foundation keeps track of the different micro:bit accessories available, currently there are 178 accessories [4] divided into 10 categories: cases (6), displays (15), edge connectors and adaptors (4), electronics (60), environmental sensors (23), gaming (10), IoT (5), robotics (41), sound and music (8) and, wearables (6).

Cases keep the micro:bit safe and can also hold the battery close to the micro:bit, not dangling off the side, see Figure 22(a), the MI:pro protector case [150]. Displays add extra outputs to the micro:bit, like 7-segment displays, multicolour LEDS and LCD screens. Figure 22(b) is a 7-segment display connected using crocodile clips [1], the gaming device the NewBit Arcade Shield [117] in Figure 22(c) also has a display. Edge connectors and adaptors break out all the micro:bit GPIO pins into an attached breadboard like the PinBit [165] in Figure 22(d). Electronics is a large category with lots of kits like the Kitronik Inventor's Kit featured in Chapter 2. Environmental sensors add extra sensors like humidity and pressure. The DFRobot's Environment Science Expansion Board [74] includes those two sensors alongside 8 other sensors. A small number of the IoT kits include connectors to the Internet, like the Gravity: IoT Starter Kit [88] for micro:bit but the majority of the kits are smart home kits with multiple sensors plugged into the micro:bit with no internet connection. Another popular category is robotics, there are accessories that control motors and there are full robot cars like the Cutebot [65] Figure 22(e) or robotic arm kits. Sound and music accessories include ones that are extra speakers for the micro:bit or instruments like the pianoboard [226], Figure 22(f). Wearables include sewable thread to attach to the micro:bit pins but mostly contain straps and cases to wear the micro:bit on the wrist, like Wrist:bit [227] in Figure 22(g).

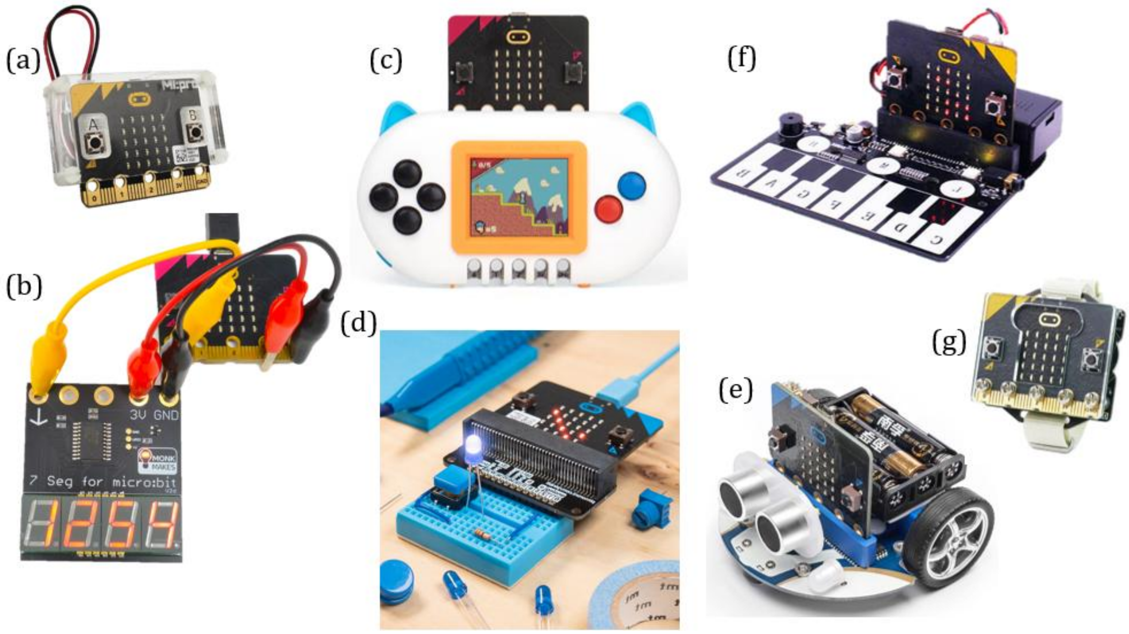


Figure 22 A sample of micro:bit accessories.

Companies can create extensions for their accessories through the MakeCode platform. A set of steps specifies how developers can create and add their own extensions and have it approved to be searchable in the MakeCode extensions webpage. Alternatively, any code repository in GitHub can be added by pasting in its

URL, the extension does not need to be approved to be used. This makes testing accessory code easy to do.

3.1.2 micro:bit software

To use any of the hardware mentioned above, the micro:bit needs to be programmed. This section covers the different ways children can write code onto a micro:bit. The micro:bit can be programmed using multiple programming types such as those listed in 2.2.8: Tile and block based graphical languages and text-based languages. These languages all run on the micro:bit runtime known as CODAL (Component Oriented Device Abstraction Layer). CODAL abstracts each hardware component of the micro:bit as a software component. It is written in C++ and is open source [39].

Tiles

The micro:bit can be coded using a tile-based language called MicroCode. This new language allows children to code the micro:bit away from a computer, tablet or phone. They can place the micro:bit in a mobile battery powered shield, Figure 23, that contains a screen (c), and a d-pad of direction buttons (d) on the left and 2 buttons on the right. Not only does MicroCode make programming the micro:bit portable, offline and not reliant on mains power, but it is also live programming. As soon as a piece of code is edited – a block is changed or added, the program changes on the micro:bit. There is no need to download the code, changes happen immediately.

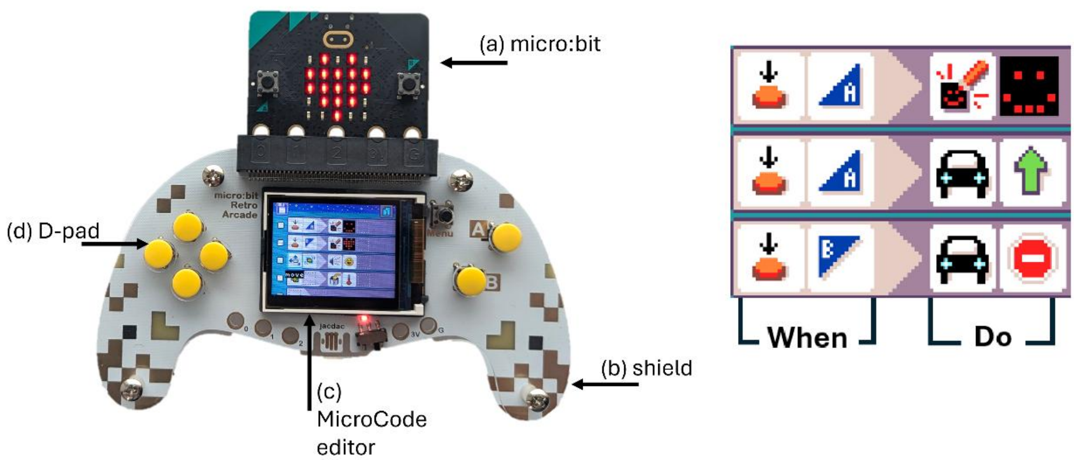


Figure 23 A shield with microcode running on it (left) next to microcode code (right)

Using the buttons on the shield children can add tiles to the screen that codes the micro:bit. Figure 23 right shows some microcode code. The code reads from left to right and top to bottom. When the user moves the cursor over each block alt text displays what the block does. The first line of code is split between **when**:

- pressed down.
- button A

and **do**:

- draw
- a smiley face

The remainder of the code involves a second micro:bit connected to a robot. The next two lines are programming the micro:bit in the shield to act as a remote for the robot micro:bit. The commands will be sent over the micro:bit radio from the shield to the robot.

The shield pictured in Figure 23 has an extra connector at the bottom called Jacdac. Jacdac is a plug-and-play hardware accessory system [53]. Children can add extra inputs and outputs to the shield and then program them using MicroCode. MicroCode is still in beta, it is still being developed, and new blocks are being added to it. As the screen is quite small, only 4 lines of code can be seen at once. The user can scroll up and down the code. Most of the hardware of the micro:bit can be programmed, the temperature sensor, light sensor etc. but not the magnetometer (yet). MicroCode also does not have the full programming functionality that other languages have. You cannot perform Mathematics sums, text editing and logic is limited to simple conditions or increments of 5, e.g. for the temperature the options are “warmer” or “colder”.

Blocks

Scratch is a popular programming tool in primary schools. It uses blocks to program “sprites” on a stage. In the starting program a sprite is a cartoon cat. Students can add and create their own sprites. They can learn how to program using loops, variables and operators with the blocks in Scratch. Students can code the micro:bit using Scratch by adding an extension to it. They need to install a program on their devices to link the micro:bit to Scratch. Once installed they can use the micro:bit to code the sprites and vice versa, for example Figure 24 shows two sets of blocks. The top blocks control the sprite when the A button on the micro:bit is pressed. The bottom blocks control the micro:bit when the sprite is clicked.

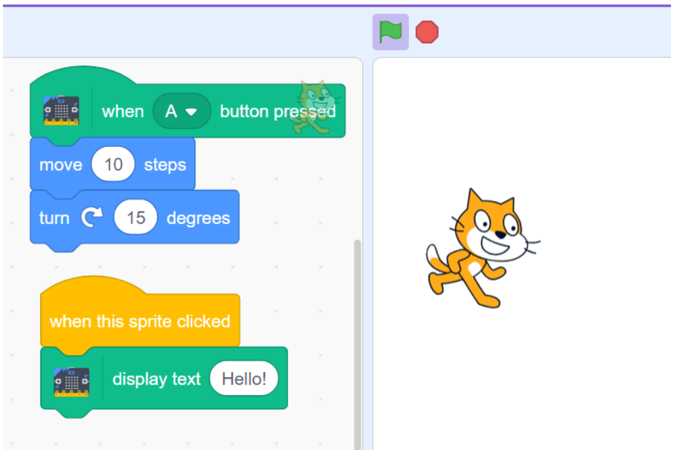


Figure 24 Programming the micro:bit with Scratch.

Another block-based programming language for the micro:bit is Microsoft MakeCode. It is based on the blockly framework from Google [26], as is Scratch. Using the MakeCode website does not require any extra software to be installed. The website is free to use without logging in or subscribing. Once loaded the website caches itself in the computer's memory. Once cached it can run offline. This can be beneficial for schools with unreliable internet access. It is **accessible**.

On the MakeCode website there is a lot of functionality for the students when coding the micro:bit; a simulator of the micro:bit is shown in Figure 25(a). This allows students to see what the micro:bit will do without needing the physical micro:bit itself. Figure 25(b) is the Download button for when students have a physical device to download to. Figure 25(c) allows the students to save the program to their computer or their own GitHub repository. Figure 25(d) is the Undo and Redo buttons, Figure 25(e) is for zooming in and out on the blocks. Figure 25(f) is the menu of the different blocks. These are the default blocks visible to all students. At the bottom of the menu students can add extensions, which creates custom menus with blocks with specialised functionality for additional hardware. Figure 25(g) is the main programming screen, the stage. Figure 25(h) is where students can switch from blocks to JavaScript or Python.

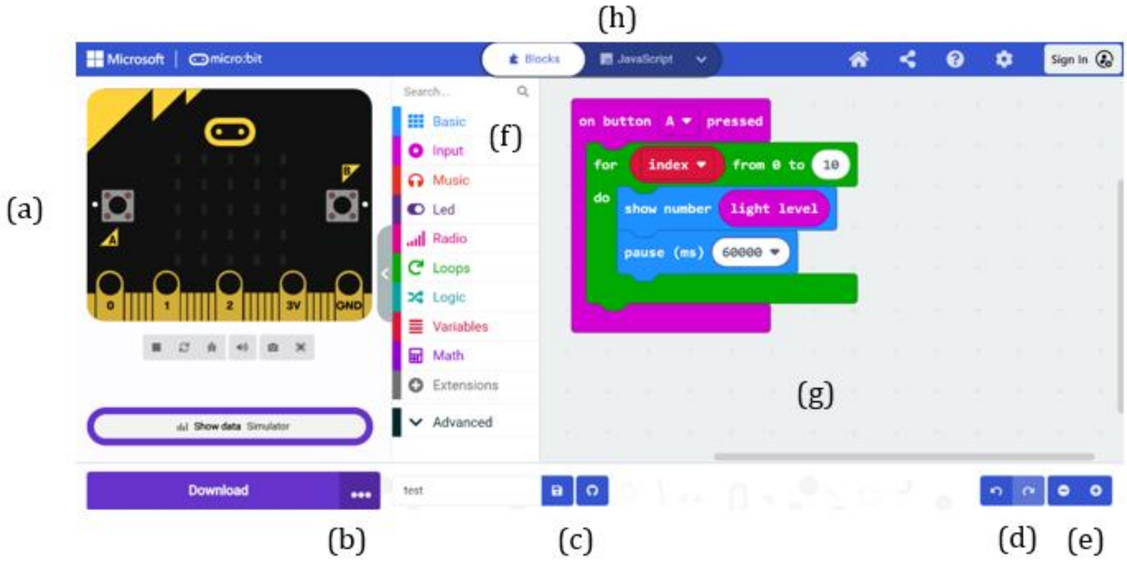


Figure 25 MakeCode website.

Behind the scenes the blocks in MakeCode are TypeScript code. This is compiled into bytecode and downloaded to the micro:bit.

MicroBlocks [147] is another blocks programming language for physical computing. Students can program multiple microcontrollers with MicroBlocks such as micro:bit, Calliope mini and the Adafruit Circuit Playground Express. It is a web-based tool. Once students have paired their device (laptop, tablet) with their micro:bit, MicroBlocks runs their code live on the device just like MicroCode. As soon as a student changes the code, the change happens live on the micro:bit. This is different to how MakeCode works. In the latter changes are not active until the students download their code to the micro:bit. With MakeCode, the code does save to the micro:bit to be used away from the computer once powered. Scratch works differently to both. With Scratch the code isn't live on the micro:bit and it isn't saved when you disconnect the micro:bit from the computer.

The code that runs on the micro:bit using MicroBlocks is also visible to other computers. The micro:bit acts like a memory stick, once plugged in the MicroBlocks code appears. This is not possible with Scratch or MakeCode.

The same MicroBlocks code also works across the different devices. The MicroBlocks code that displays a smiley face on the LEDS of a micro:bit will also work on a Calliope mini, a microcontroller very similar to the micro:bit developed in Germany. It has a 5*5 LED screen too.

MicroBlocks is made of three components: the blocks editor that runs on the computer, a virtual machine that runs on the micro:bit and a communication system that updates the code on the micro:bit as the user edits it. MicroBlocks code is compiled into bytecodes that are executed by the virtual machine. Bytecodes are low-level instructions similar to machine code. They are independent of any particular processor architecture which allows MicroBlocks to support many different microcontrollers.

Text programming

Moving on from blocks, students can code the micro:bit in text-based languages. MakeCode gives students the option to view their block code in JavaScript or Python using a toggle button. Figure 26 shows the toggle and the code from Figure 25 in JavaScript. The students can edit the code in JavaScript or Python and switch back to blocks, and vice versa. This feature was intended to support students' **progression** from blocks to text, but no evaluation has yet been undertaken on the effectiveness of the approach.

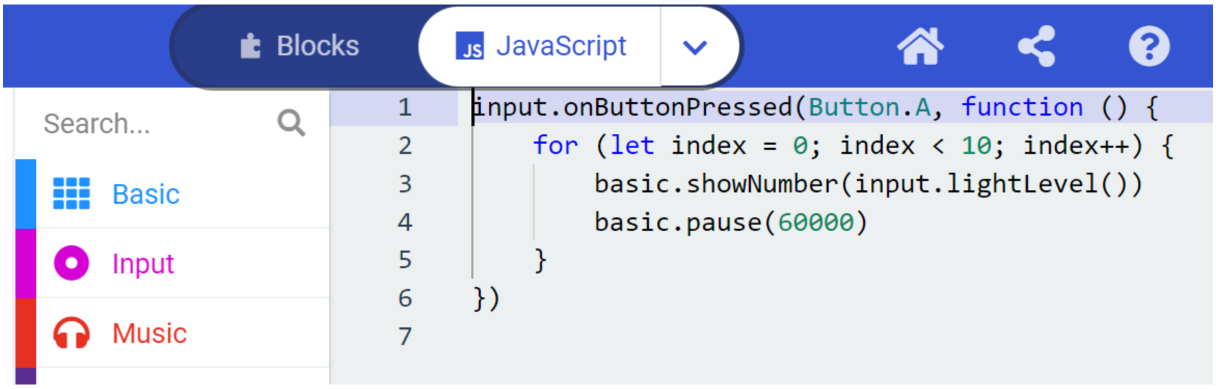


Figure 26 JavaScript code in MakeCode.

There are other ways to program the micro:bit in Python. The Microbit Educational Foundation provide a Python editor [145] similar to MakeCode. It is an online tool, has a micro:bit simulator and can connect directly to a physical micro:bit. Mu is an editor that can be installed on computers and used to code the micro:bit in Python, as is Visual Studio. Only the Visual Studio code editor has a simulator for micro:bit.

Downloading code

Once the code has been written and tested, the students need to download it to the micro:bit. They can plug in a micro:bit to the USB port on the computer and download it by either pairing the micro:bit to the computer once or by dragging and dropping the downloaded file to the micro:bit which appears as a drive.

An alternative way is to use the micro:bit app on a mobile device such as a tablet or phone. To download the code the students need to pair the micro:bit with the device. They must put the micro:bit in Bluetooth mode by pressing A and B buttons together then pressing and releasing the reset button on the back of the micro:bit. If done correctly, the micro:bit display will scroll the text "Pairing mode" and then show a unique graphical representation of the micro:bit's 5-character identifier. The students confirm their 5-character identifier on the app to finish the pairing process. The 5-character identifier is used so students are pairing their micro:bit and not one nearby that's also in pairing mode.

3.1.3 Classroom use

Educators in both primary and secondary schools can use the micro:bit to teach children how to code. Many of the children will be familiar with blocks from using Scratch in primary schools. Educators can start demonstrating how to use MakeCode with the simulator. Once the children are familiar with creating a program, they can then transfer the code to the physical micro:bit. A common first program is displaying "hello world" or their name across the LEDs. It is a very simple but powerful program as many children will not have coded a physical device before.

Educators can then move onto using the buttons to display text across the LEDs and then displaying sensor data. A common lesson is to count the number of times the micro:bit is shaken and use it as a step counter by strapping it to their wrists or legs with the battery pack. The NCCE resources for micro:bit teach students how to code the micro:bit with iteration, variables and conditions in primary school Year 6. Their resources for secondary school students move onto coding the micro:bit using Python (Year 8) and external accessories (Year 9).

3.1.4 Data with the micro:bit

The micro:bit is a microcontroller. It must be programmed by another device such as a PC or an app from a mobile device like a tablet or a phone. Once programmed it can run the program independently, like a computer. The micro:bit's main processor has 128 kilobytes of RAM. In addition it also has 128 kilobytes of persistent storage. Students can store data into this storage that they want to persist between reboots of the device. They can save the data over time, e.g. a temperature reading every hour or when an event happens, e.g. when button A is pressed. An example program of saving data to the micro:bit memory is seen below in Figure 27.

The "on start" block runs as soon as the micro:bit gets power. This block sets up the data storage by initialising the log file with the column "temperature" (a). In a forever loop the data is logged from the temperature sensor, the micro:bit then pauses for 5000 milliseconds (5 seconds), and returns to the start of the loop.

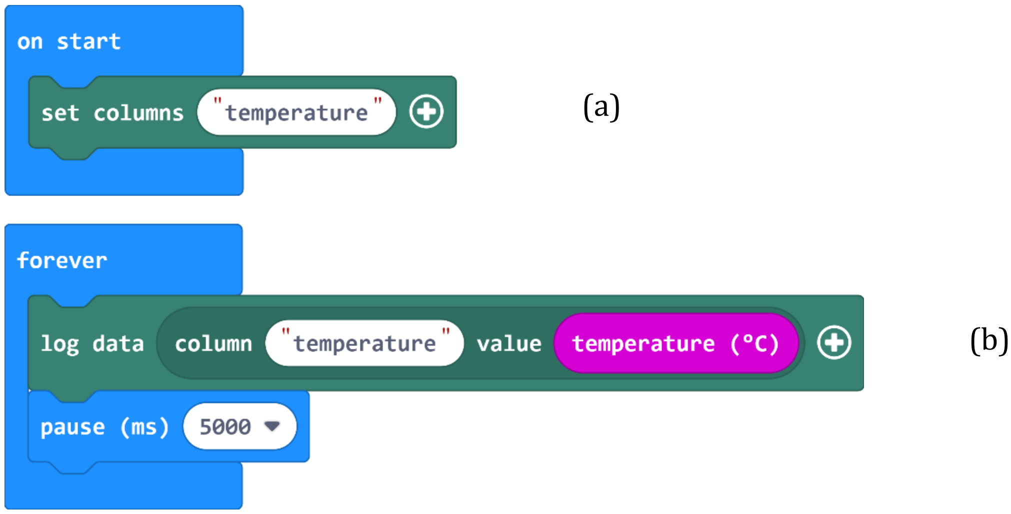


Figure 27 MakeCode blocks to record the temperature to the log file every 5 seconds

When the user plugs the micro:bit into a computer they can open it like an external storage device. Inside is a HTML file called MY_DATA.htm which displays the contents of the log file. Figure 28 shows the log file with the recorded temperature data from the program above at 5 seconds, 10 seconds, 15 seconds, and so on.

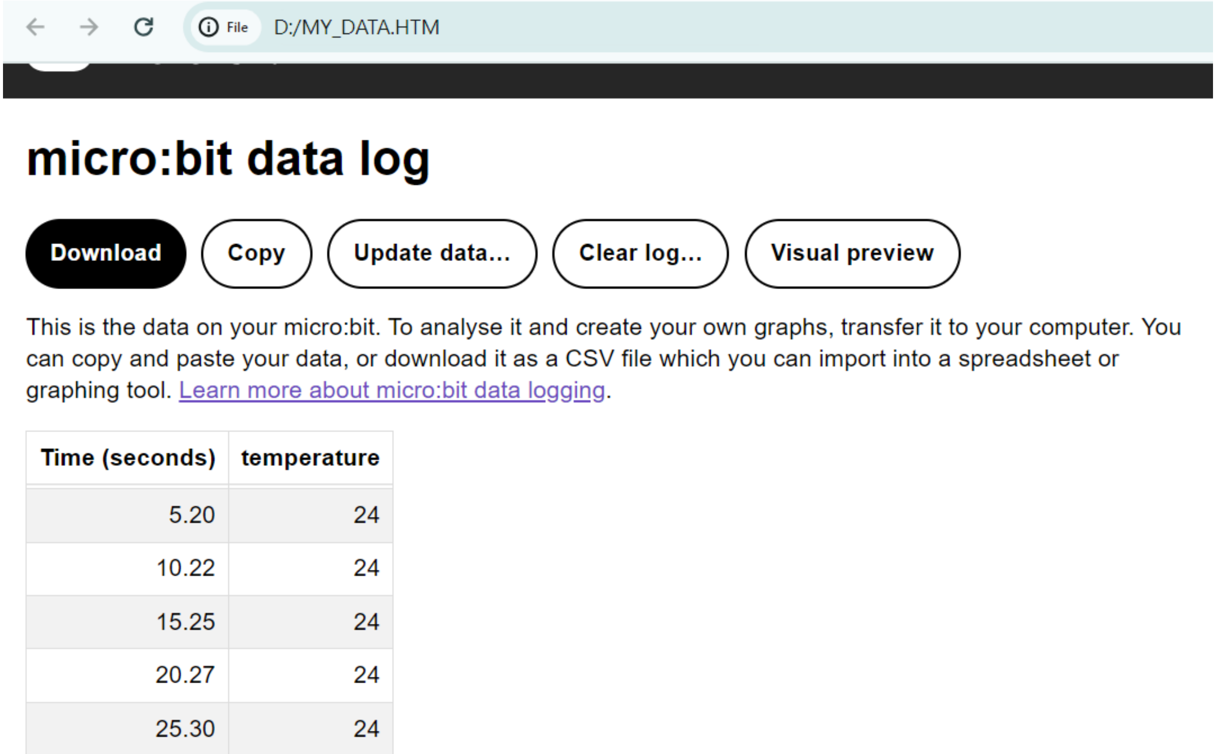


Figure 28 The micro:bit log file

The next time the micro:bit is turned on there will be new rows starting at 5 seconds, 10, 15 seconds and so on. This data will remain on the micro:bit until the user deletes it using the delete block or when they download a new program onto the micro:bit.

3.1.5 Analysis

This section introduced the BBC micro:bit as a physical computing device used in classrooms. It detailed the hardware and software used on the micro:bit as well as accessories that have been developed to add extra functionality to it. How teachers use the micro:bit in classrooms was briefly discussed. How the micro:bit can store and display data was demonstrated.

With its embedded sensors, persistent storage and its screen, the micro:bit has strong potential as a tool for students to use in a data science pipeline, specifically data collection. This section looks at how the micro:bit can fulfil the identified needs highlighted under 2.4 Background Analysis, namely how the micro:bit is cost effective, accessible, engaging and educationally sound.

Cost effective

The micro:bit is cost effective; given the number of sensors embedded. it is relatively cheap to purchase (£12). In the UK, the micro:bit is an existing resource that many schools already have access to. The initial launch of the micro:bit in 2014 provided 800,000 micro:bits for free to children in secondary school. In 2024, another initiative provided a class set of micro:bits for free to primary schools who registered, 675,000 micro:bits were provided with 95% of UK primary schools gaining a class set. The micro:bit has no subscription model for its hardware or software, there is no extra cost associated with using them. All updates to the hardware and software are free. The small size of the micro:bit makes it easy to store and share between classes, meaning schools don’t need to buy multiple sets.

Accessible

The micro:bit is accessible to a range of educators with varying skill levels. There is no setup or installing of hardware or software needed. As it has no software to install there are no updates or issues with the wrong version of the software. The coding platform MakeCode looks and acts very similarly to the Scratch programming environment, that many teachers are already familiar with, meaning they don’t need much extra training to learn how to use the micro:bit. The way the micro:bit can be paired over USB, so the students can press download and not have to drag and drop the program every time they make a change, keeps it simple and accessible for both children and educators. As mentioned previously the micro:bit requires no software to be installed, teachers do not need administrative access to the computers. They also don’t need to install the micro:bit hardware to the school computers or the school’s WiFi or intranet. The micro:bit can be coded and works independently of Internet access.

Student accessible

The micro:bit was developed as a device for children, it is *age appropriate*. It was made to be accessible to children. Microsoft MakeCode was developed for children to easily drag and drop blocks onto a screen. They can zoom in and out of the screen to view the blocks better. They can press undo to undo mistakes. The website is friendly and easy to use.

Engaging

The micro:bit has been known to be engaging. It has been used in several research projects with positive results regarding motivation, creativity, and basic knowledge of coding [180, 184, 185] . Students have found the tool easy to use, enjoyable [87] and useful for problem-solving

(Abonyi-Tóth Andorand Pluhár, 2019)

. "*The introduction of micro:bit into classroom teaching breaks the boring teaching methods of the past. Micro:bit can combine software and hardware products, which not only improves students' understanding of abstract concepts in computational thinking; but also strengthens their problem-solving skills*" [134].

The micro:bit itself has proven to be an engaging tool for teaching students Computing [12, 97, 188], worldwide it has been it has been adopted into multiple national curricula including the UK, Denmark, Croatia and Singapore [142, 143]. Between September 2023 and August 2024 there were 4 million active users across the different micro:bit editors.

As stated in the previous chapter one method to engage students is to make the content relate to **real-life**. With its array of embedded sensors, the micro:bit can be rooted into real-life data collection scenarios. It can gather and store data in students' local environments for them to analyse.

Educationally sound

This **contextualisation of learning** is an **educationally sound** method known to occur in physical computing lessons. The micro:bit supports this method of relating Computing content to other aspects of the curriculum and to real life. By using sensors to collect data about the environment around them, students can learn more about other aspects of the curriculum such as Geography and Science. Collecting real data in real time using the micro:bit helps connect the students to what is happening in real life.

Another method is **computational thinking.** Computational thinking means considering a problem in a way that a computer can help us to solve it. The micro:bit can act as the computer. Students can use it to ask questions and solve problems. They can break down the problems using other computational thinking methods such as *decomposition, algorithms and pattern recognition.* Another educationally sound method is **scaffolding programming tasks** and **collaborative learning.** The MakeCode environment for micro:bit makes it easy for teachers to scaffold. They can share a MakeCode program with the students to give them code blocks to start them on a task. Given the duality of hardware and software of the micro:bit, students can work **collaboratively**. Tasks can be divided between them of building the hardware, connecting accessories and writing the software.

Tasks can also cover multiple subject in a **cross-curricular** way**.** Given the number of environmental sensors on the micro:bit (e.g. temperature, light, sound, motion) it can be used in Science experiments, Geography lessons as well as Computing.

As the micro:bit can be programmed in different ways, using blocks and text based languages, this allows educators to plan for **progression.** Younger children are generally introduced to blocks first and then older children learn text-based programming. Educators can use the micro:bit as a familiar tool to help students progress from blocks to text based programming.

3.1.6 Summary

The micro:bit is a versatile physical computing tool with embedded hardware, easy-to-use software, and is known to be extensible via a range of accessories. Given its ability to record and store data, it has great potential to be used as a data collection tool in classrooms. However, reviewing the available accessories for micro:bit, there are few products that promote the use of the micro:bit as a data collection tool.

This section has described how the micro:bit can meet the requirements detailed in Chapter 2 of being **pragmatic**, **engaging** and **educationally sound**.

The micro:bit is a popular tool used in research. The next section looks at this research, how and why researchers have used the micro:bit, what subjects it's taught in, and the demographers of its users.

3.2 micro:bit literature review

In this section the micro:bit in research literature is explored. A systematic literature review of the use of the BBC micro:bit was completed. A total of 86 papers were analysed. micro:bit has been used as a vehicle to address a growing number of educational challenges with children. However, to the best of our knowledge, there is no systematic review/report bringing together, analysing and summarising all these different uses of these practises.

The goal in this section is to identify how and for what purpose micro:bit is being used in research with children. The purpose of this literature review is to delineate the research space and to identify the opportunities to further explore and investigate the gaps in the current literature. The next section details the methodology used to conduct the literature review.

3.2.1 Methodology

To carry out the literature review the guidance of Kitchenham [115] was followed. A protocol was created to define the search strategy and the inclusion and exclusion criteria. This section will describe the methods used.

Search query

Papers were searched for in the period 2014 — 2023, starting from when micro:bit was first introduced to when the literature survey was started, i.e. September 2023. The initial search was conducted on Google Scholar using the following search query to understand which were the main publication venues: *("child*" OR "kid*" OR ("student*" AND "school")) AND ("microbit" OR "micro:bit" OR "micro bit")*. This informal search yielded 2360 results, and the top 3 publishers were ACM, IEEE and Springer. Therefore, the same query was used in the following ACM, IEEE and Springer online databases and included all papers from their sponsored conferences and affiliated publication venues. Searches were conducted in the full text, capturing publications that mention these terms anywhere in the paper. This strategy was chosen considering that researchers may have not included 'micro:bit' in the title or abstract given that it is a registered trademark for a commercial product.

This initial search resulted in 377 papers (239 results from the ACM database, 104 papers from the Springer database and 34 from IEEE).

Inclusion criteria

The next step of the process was to scan papers by their title to exclude irrelevant topics, e.g. a tiny drill piece is known as a micro bit. Furthermore, studies that did not have a focus on children, education or if the micro:bit was not a significant part of the paper, were excluded. Papers in formal and informal education environments such as summer camps, museums, and community events were included. The number of papers after this step was 308 papers (178 results from the ACM database, 99 papers from the Springer database and 31 from IEEE). Next, further exclusion criteria was applied.

Exclusion criteria

The following criteria were applied to exclude papers from the search results:

- Workshop papers with no details
- Adverts, Books, protocols
- Not in English
- Target population for the activity >18 year olds (e.g. university students)
- Focusing on the technology only, no evaluation with students or teachers
- Duplicate project (if different papers report from the same pool of data it was decided to keep the latest/most comprehensive version)
- Publications prior to 2014 (when micro:bit was launched)

Search strategy

As shown in Figure 29, this search yielded 86 papers. In order to ensure the inclusion and exclusion criteria were being applied correctly, a second researcher screened a sample of 10 papers independently. Their results were compared with the main review. Where there was conflict a third reviewer, the PhD co-supervisor, gave the final decision and the inclusion and exclusion criteria were clarified. The papers were then screened based on the inclusion and exclusion criterion to select papers for inclusion, which yielded a final corpus of 86 publications. Figure 29 highlights the reasons reports were excluded. The top reason for exclusion (136 out of 222 excluded papers) was not using the micro:bit. These papers mentioned micro:bit in their abstract or introduction as a physical computing tool, but their study was not on or using micro:bit .

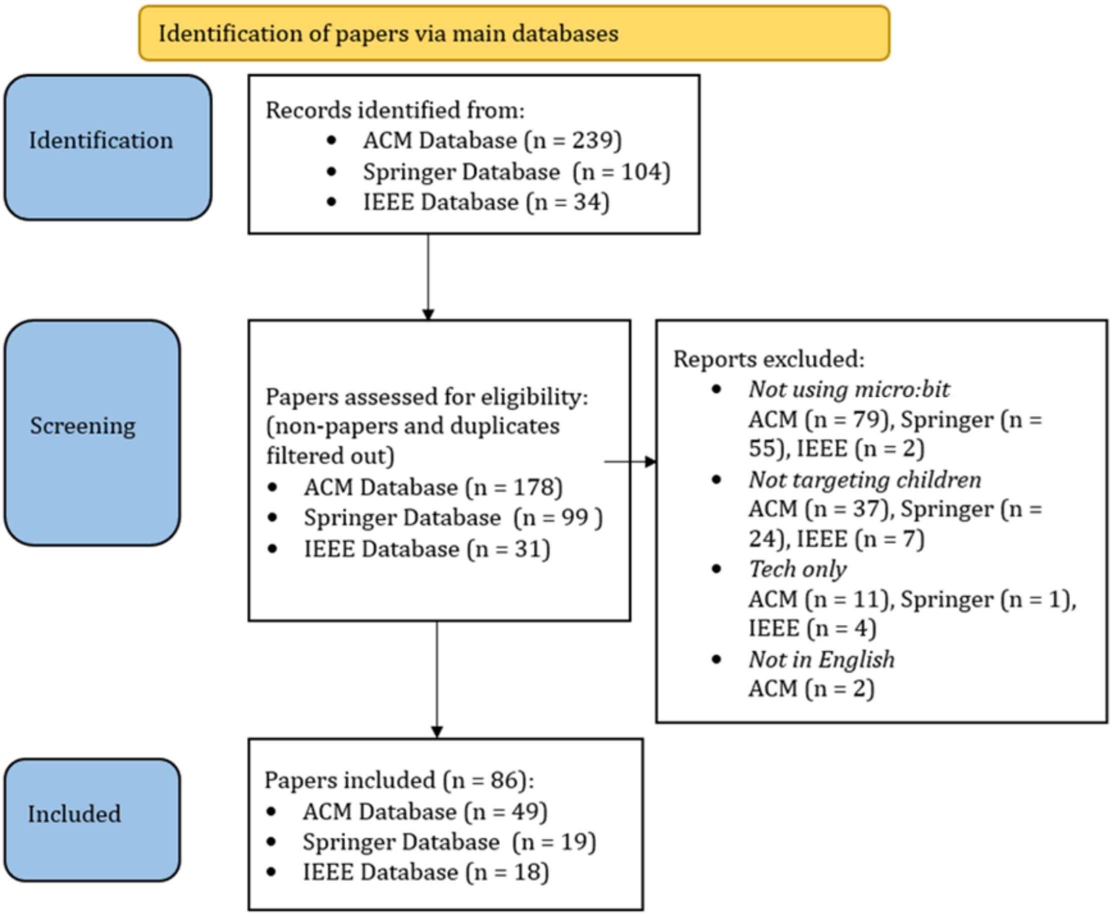


Figure 29 Literature review search strategy

3.2.2 Overview of papers

This section looks at how the micro:bit was used in all of the remaining papers. It looks at rationale for use: why researchers chose the micro:bit, what features of the micro:bit they used in their research and what programming languages they used with the micro:bit. Then the demographics of the users in the study is analysed.

Rationale for use

Looking at why researchers chose the micro:bit; there were 36 unique reasons recorded, with many studies quoting multiple reasons. Figure 30 shows the top 5 rationale for use when choosing the micro:bit in research. Cost was a major reason for choosing the micro:bit. Its onboard sensors were another top reason as well as the opportunity to connect other sensors.

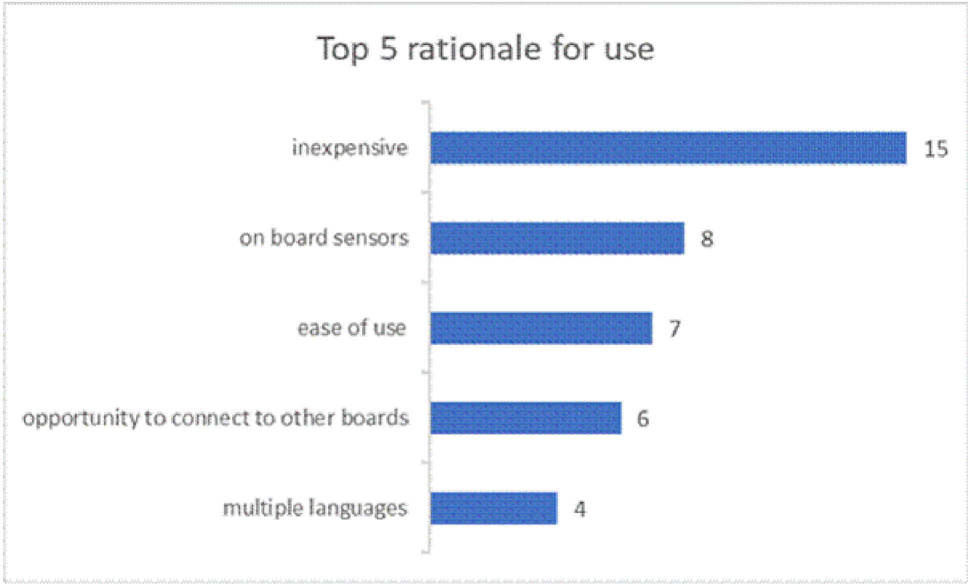


Figure 30 Top 5 rationales for use

The physicality of the micro:bit was also mentioned, it being small (3), portable (2), robust (4) was important to researchers and educators as well as it being tangible (2). Specific functionality was mentioned when explaining why they chose the micro:bit: the on-board sensors as a whole (8), the LED screen (2), Bluetooth (2), radio communication (1) and the movement sensor (1).

Features used

Looking at features of the micro:bit used in the activities, the accelerometer was the most used (28) in the studies that stated any features (67), followed by the screen (27) then buttons (23). 15 studies didn't state what features of the micro:bit were used and 4 studies didn't use any of the features of the micro:bit. Table 15 shows the top 10 features of the micro:bit used in studies.

Features	Count
Accelerometer	28
Screen	27
Buttons	23
Not stated	15
Radio	15
Add on accessories	13
Pins	11
Temperature sensor	9
Add on hardware	7
Light sensor	7

Table 9 Top 10 features used

Coding

Finally, the language the micro:bit was programmed in was analysed. MakeCode was the most popular way to code the micro:bit; 78% (46 out of the 59) of studies which stated the language used MakeCode only, another 4 with MakeCode and another language. Participants did not always program the micro:bits themselves. They were coded by the researchers to be used as a tool in the study, most often the language chosen was not stated in the studies.

How it was coded	Count
MakeCode	46
Not stated	27
Scratch	3
MakeCode & Python	3
Python, ml-machine learning, kittenblock, blockly, touch develop	1 each

Table 10 Programming languages

Demographics of users

This section looks at the main users of the micro:bit during the research activity, who they are and what ages they are? It also looks at the activity itself: how often did the activity take place? What was the frequency and duration of the activity? The answers to these questions can help situate where and how teaching and learning with the micro:bit is taking place and why researchers are choosing to use this tool in education.

Below in Table 11 are the codes used to track the users and a description of each user. For example, if the activity involved people aged 13, the activity would be coded as "Adolescents".

Users	What users were involved in the research?
Children	0 – 11-year-olds (pre and primary school students)
Adolescents	12 – 18-year-olds (secondary school students)
Educators	Adult providing the educational instruction, e.g. teacher, facilitator
Family members	Includes those directly related to the children, e.g. parents, siblings
Researchers	Academic researchers working in this space
Industry	Companies, businesses or products, e.g. micro:bit foundation

Table 11 Codes for user demographics

Below in Table 12 are the two codes used for "where does the activity take place".

Setting	Where did the activity take place?
Online	Technology-mediated engagements, e.g. Zoom interviews
In person	All users are physically present during the research activity

Table 12 Codes for "where does the activity take place?"

To understand how frequent activities were done, three fields were completed for each paper: session length (hours), number of sessions and overall period. For example in the first row of Table 13 below [217] was a 12-week course of one 1-hour lesson. Then [157], [192], [114] and [110] were 1 session lasting 2 hours that was run once.

Papers	Session length (hours)	Number of sessions	Overall period
[215]	1	1	12 weeks
[157] [192] [114] [110]	2	1	Once

Table 13 Sample of activity durations

3.2.3 Results of study

Most of the research involved children (76 of 86) mostly aged under 11 (37 of 86) and 50 studies comprised of children and other participants, see Figure 31. About 20% of the papers (18 of 86) also reported involving educators in the research study. A smaller number included researchers (5 of 86), family members (3 of 86), and industry partners (1 of 86).

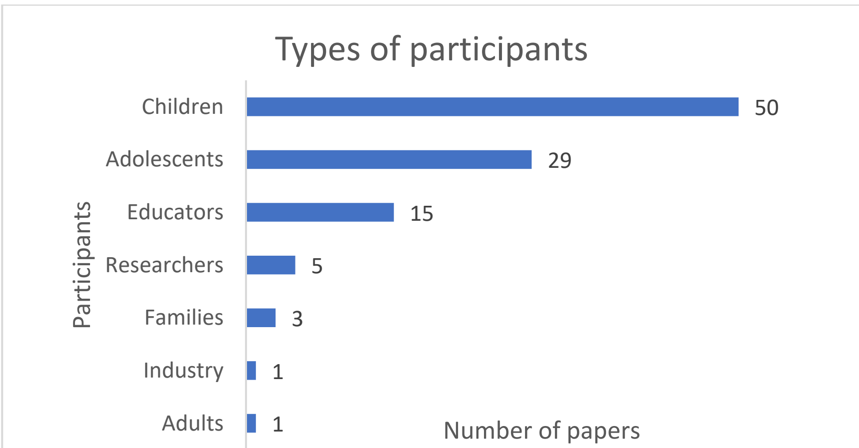


Figure 31 Participants grouped

Two of the papers use the micro:bit with pre-school aged children [59, 160]. These projects used the micro:bit as a tool, as a light up lantern and a musical device. The researchers pre-programmed the micro:bits and placed them inside different devices for the children to experience. The Lanterns researchers Dylan et al. state they chose the micro:bit because it is a "*readily available and affordable product that was already familiar to many UK children*". And because it is a device capable of radio and Bluetooth communications but not Internet: "*Therefore it is appropriate for safely exploring the IoT design space with children*".

Bernad et al. used the micro:bit to develop experimental equipment to give students a greater understanding of how measurement is done in Physics [19]. They did this with children in their last year of primary school. They stated they used Physics experiments the students were familiar with so the development of experimental equipment wouldn't be too difficult for the students. A study by Dias et al. [55] involved teaching **computational thinking** to students from the full primary age range of 6 to 12 years old. The students used physical blocks to "program" the micro:bit and the teacher used a mobile phone to take a photo of the blocks to translate them into MakeCode programs to download to the micro:bit.

Of the 52 studies that stated a length of time for their activity, 2 hours was the most common length of time (13 of 52), followed by 1 hour (11 of 52) then 6 hours, i.e., a day (10 of 52), Figure 32(c). Of the studies that stated the number of times a session was run (56 of 86), the most common repetition was once (23 of 56) then twice (10 of 56) and then four times (9 of 56), Figure 32(b). Finally, the studies that stated the overall period the activity took place (43 of 86) were most run just once (14 of 43), over 1 week (5 of 43) then over 2 weeks (4 of 43), Figure 32(a).

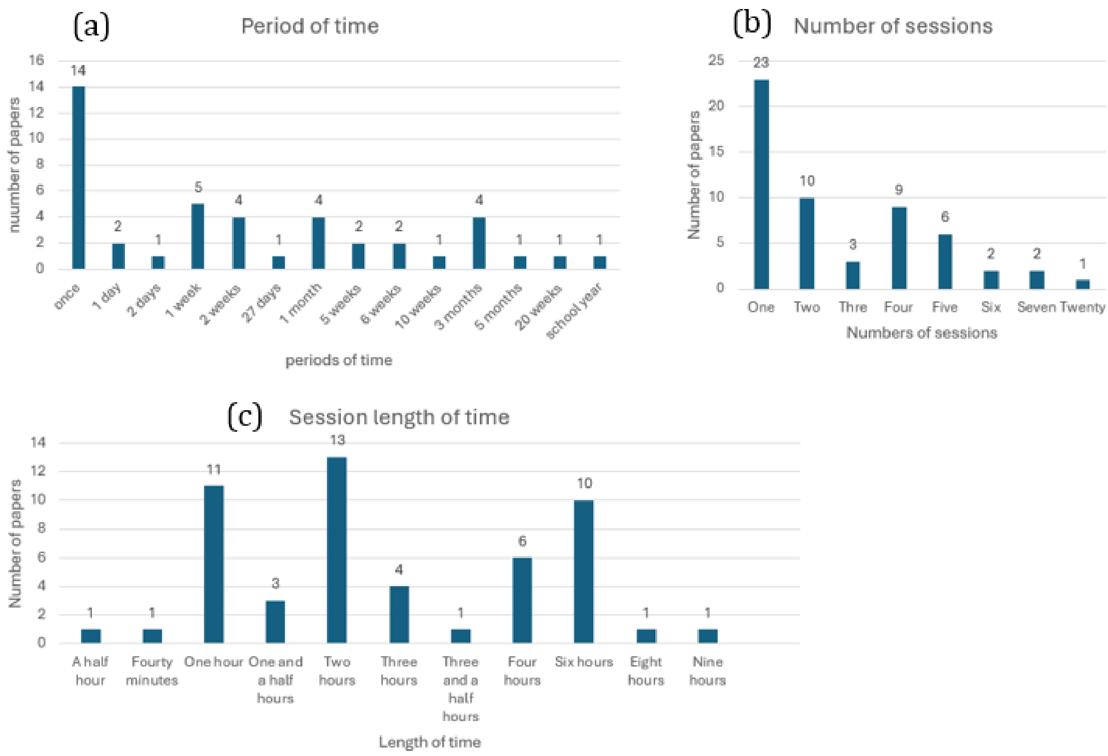


Figure 32 Frequency statistics (a) period of time (b) number of sessions (c) length of time

The largest study was Faherty et al [157]. They ran one-to-two-hour workshops with 4,600 students in over 90 Irish schools. The workshops involved using the micro:bit to engage students in Computing. The workshop was adapted for different age groups. The aim of the project was to collect research surveys from the students and make connections with teachers for further research. There was no further workshop or progression. Jin ran a one-off workshop of hands-on projects modularized into tasks that isolated the software or hardware workloads [110].

Due to the hands-on nature of micro:bit, it is not surprising that most of the research reported took place in person (66 of 86). The context was recorded with 37 of the 66 taking place in school during normal school hours, with another 3 after school hours but still in the school environment. Another 5 studies took place both in person and online. Those that took place online only (11 of 86) were mostly due to COVID-19 restrictions, e.g. [85, 111, 126, 163, 179] and generally the rest focused on the evaluation of micro:bit activities and relied on online surveys or remote interviews.

3.2.4 Summary of papers

This section summarises the papers by the main subject area of the activity that took place in the papers followed by the final section of ethics and privacy before analysis.

The main subject area of the activity was recorded. These were not limited to the academic subjects of the school but the specific subject the authors stated. For example, IoT was a subject area but is not a curriculum subject. Many of the subject areas were grouped under one heading, for example studies involving smart devices, smart objects and smart city were all coded under the same heading: Smart devices [174, 179, 190]. Papers were coded by their main subject, not multiple subjects. For example, Bodon et al. [27] was a study about technology and sport. As the students didn't code any of the tools they used to record their sporting activities the paper wasn't coded under "Coding" but "Sport". Overall there were 31 unique subjects recorded. Table 14 shows the top 10 subjects and their paper count. The most popular subjects the research is based on are around Computing and Computing related topics like Smart devices [173, 178, 189, 216] and IoT [35, 106, 117, 118, 195]. 32 of the papers involved subjects not related to Computing such as S port [108, 123, 222]. This is due to the popular accelerometer and its small size.

Subjects	**Count**
Coding	13
Computational thinking	8
Computer Science	6
machine learning	6
Micro:bit	5
IoT	4
Sports	4
Smart devices	4
STEAM	4
Maker activities	4

Table 14 Top 10 subjects

Coding

King et al. used the micro:bit to see whether children from different socioeconomic backgrounds experienced fun differently when learning to code [114]. They ran a two hour long, playful programming workshop to introduce programming with micro:bits to the primary aged children. They concluded that all the children experienced the workshops as **fun**, with middle income children having the most fun.

Another study ran in a Science centre tested if programming exercises would increase ability beliefs and interest in programming [181]. Their results showed that the young women's **ability beliefs** increased after taking part in the activity, even months later. And their **interest in programming** did not decrease over time, compared to a control group whose interest did.

Cabrera et al. [32] compared students coding the micro:bit using MicroBlocks and MakeCode in one hour-long study. These are both block-based programming tools. The main difference between them is that MicroBlocks uses live coding. When a student makes an edit in MicroBlocks, this happens immediately on the micro:bit. With MakeCode the users must press "Download" if they want to view their changes on the micro:bit device. The study did find students made more errors with MakeCode, they thought the code was live and didn't always press "Download". The authors calls this "expected liveness". The study suggests that *"liveness does affect children's interactions, both in free play and structured activities."* However, overall the MakeCode group had more success completing the two challenges set. The MakeCode students also completed the challenges quicker than the MicroBlocks students.

Krpan et al. did a larger study with 49 6th graders in Croatia (11- to 12-year-olds) [123]. At this age students are already learning to code using Python. Their study looked at how a block-based programming language (MakeCode) could help students with their programming misconceptions. Programming misconceptions occur when students develop incorrect mental models. The students took a Python test before and after the study. The results shows that MakeCode can "*successfully be used as a mediated transfer tool for understanding learned programming concepts*". Krpan et al. reports the reason for using MakeCode was because these students received a micro:bit device on a national rollout in Croatia therefore this study would be **cost effective**.

Computational thinking

Carlborg et al. focussed their research on "Considerations and Technical Pitfalls for Teaching Computational Thinking with BBC micro:bit" [206]. Their **technical pitfalls** were being unable to install an app on school iPads. The educators didn't have the app-store password to install the MakeCode app to code the micro:bits. Unreliable internet connections kept some educators from coding the micro:bit, they did note that MakeCode is cached but that learning resources such as videos are bandwidth heavy and can disrupt learning.

Kwon et al. [221] developed a program integrating **computational thinking** into a technological problem-solving process: hacking a remote-control car with a micro:bit. They used the cars to teach the students about physical computing, sensors and actuators. They then taught the students how to replace the robot's computer with a micro:bit. The students had to program the car with the micro:bit, guided by the teacher. The study results showed that the technological problem-solving process positively affected the children's computational thinking-related competencies. There was also an improvement in their **attitudes towards technology**.

Computer Science

Garcia et al. explored teaching functional programming through fun, engaging and personally relevant activities [84]. Their third unit was around physical computing as they wanted to show students that programming can have **real-world** effects. They used the micro:bit as it was *cost effective* and required *no additional software* to install. Many of the teachers stated they were interested in this unit as they believed it would be more **fun** and **engaging** for students.

Both [157] and [128] were programs to introduce students to Computer Science, in Ireland and Guyana respectively. They both had careers aspects, introducing the roles and careers of a computer scientist, and attending a STEM event to gain further insights into career opportunities. They both ran a micro:bit module, programming it in hands-on sessions with the students. In the Guyanese research an overwhelming majority showed **interest** in the micro:bit module despite finding it more challenging (than a Scratch module).

Machine learning

Machine learning is growing in popularity in society and in education. Six papers used the micro:bit in machine learning and artificial intelligence studies. Project moveSMART [109] used the micro:bit's accelerometer to train a machine learning tool. In three data science lessons students designed their own physical activity monitor using a micro:bit coded using MakeCode, gathered the data then analysed it. The analysis was performed using a custom machine learning tool on a computer that recognised patterns. The researchers developed and trained the machine learning algorithm themselves. They aimed to reinforce the concept of *pattern recognition* in data science and expose students to **real-world** uses of machine learning.

Tamborg et al. also use machine learning and the micro:bit to teach children programming and **computational thinking** skills (PCT) [196]. In their paper, they argue that Mathematics presents opportunities to deal with the challenge of teaching PCT. They state that *"Most national curricula do not have a subject where such topics (programming and computational thinking skills) are apparent to address*." They use the game Rock, Paper, Scissors as an example project. Students take part in 4 tasks. Task 1 the students use flowcharts to describe the game. Task 2 they use the micro:bit and the Scratch programming tool to code the game. Task 3 they use Scratch and a machine-learning environment specifically designed for the task. The students train a model to recognise different hand gestures through the computer webcam. Task 4 the students upload images of hand gestures to Google Teachable Machine on a Raspberry Pi to train a model.

Bilstrup et al [23] used the micro:bit to teach machine learning as a design material. They state the development of machine learning is driven by advances in *"collecting, handling and Computing of large amounts of data".* Their six-hour study with a class of 20 8th grade students (13- to 14-year-olds), and resulting website ml-machine.org guide students through a series of lessons to build their own machine learning model using the micro:bit and its accelerometer. "*ML-Machine builds machine learning models to recognize patterns in the accelerometer data*." They highlight how students need **scaffolds** to support them with the "*difficult part of ML*". The Ml-Machine project was the inspiration for micro:bit CreateAI [141]. CreateAI is a free, web-based tool that supports students to train an ML model and run it on their micro:bit.

Internet of Things

Tamashiro et al. [195] created an IoT kit "Orbit IOT", a micro:bit-based toolkit, to teach both the technical and societal implications of IoT. The kit allows the micro:bit to be connected to the Internet and to control household devices such as lightbulbs and fans. It also includes a website that allows students to see the movement of data in the network. The study included a daylong session with 12- to 13-year-old students to learn how to use the kit and a second day to implement what they had learnt. The website visualised the movement of data which helped support troubleshooting, teaching about networks and created a discussion around **ethics**. The Orbit IoT kit has a lot of potential to teach data science to children.

Charlton et al. also connected the micro:bit to the Internet [35]. Their study created a connection between the micro:bit and the students' phones to send messages over WhatsApp. They stated many reasons for choosing the micro:bit over other devices including having all three of these features: the onboard sensors, the LED screen, and the Bluetooth network. Another reason was being **easy to program**. The size and design of the micro:bit also appealed to the students during the activity with one commenting: "*It's so small but it can do many things*". At the end of the project the students stated how they enjoyed having a sense of **ownership** of the project because the tasks they had set themselves had **meaning** and **purpose** for them.

Sport

Project moveSMART [109] is a web-based platform that aims to combine multiple subjects in elementary schools such as Physical Education and Computer Science. They employ a **project-based learning** approach to expose students to data analysis and machine learning. They note how difficult it is to address the issues due to "*a crowded curriculum required to meet state learning objectives across academic subjects*". Their project aims to solve this problem through an integrative curriculum of using students' physical activity data in Computer Science lessons. The students use an online platform to record their physical data recorded from the micro:bits.

Another project using sport is Kumar et al. [124]. They used a commercial micro:bit accessory called the wear:bit to strap a micro:bit to students' wrists or ankles. The wear:bit provides power and a buzzer to the micro:bit. The students would then move and see their movements painted on a screen in a web program called PaintBall. They aimed to have a lower setup barrier than other activities and technology, where the students can "*simply pick up a pre-programmed wearable before starting their gameplay*." They used the micro:bit as a tool to track movement.

Smart devices

Voštinár et al. [216] created a mock-up of a smart home with multiple sensors in an extra-curricular club for children. One of the children programmed a temperature sensor, if the sensor went over a certain number it would trigger a fan to turn on.

Gennari et al's [174] research used a card-based game with cards for the inputs and outputs of the micro:bit to support students to design, develop and prototype smart objects. The children **collaborated** to play the game and create a micro:bit smart object such as a smart basket-area that uses light and sound to invite passers-by to play and stay healthy. At the start of the project, the researchers showed a **scaffolding programming** example to help the students get started on programming the micro:bit. Using the cards helped **decompose** the project into different parts. The cards were coloured differently depending on their functionality, e.g. Input cards like a button were blue and output cards like an LED were yellow.

Simonofski et al. used the micro:bit radio to create a voting system in a smart city project [190]. The students coded the micro:bits to vote for or against the building of a new mall in the city. The teacher coded the final vote tallying micro:bit. Before the activity students view of a smart city was "a city that contains technology". After using the micro:bits the children mentioned that the smart city must "answer the questions of citizens", "use technology appropriately" or "improve the quality of life of citizens". This demonstrates that the micro:bit helped them connect with technology in a **meaningful** way with **real life** applications for society.

STEAM

STEAM is the combination of Science, Technology, Engineering, Art and Mathematics. Five studies covered some or all the subjects: STEM [83], STEAM [134, 152, 194], and Art, Science and Computing [189].

Lu et al. highlight the effectiveness of **project-based learning** to teach **interdisciplinary subjects** with their micro:bit paper lamp project [194]. They combined Art and Computing using the micro:bit. Nagai et al also combined art, programming and electronics for students to create a light up project with their parents [153]. They state the children had high **motivation**, they did not get bored and were fully engaged. The authors state as the children were able to program the micro:bits alongside the other expressive activities, the children were able to design the electronics in a more **focused** and **meaningful** way.

Maker activities

In an activity connected to COVID-19, Rodrigo asked students "Can technology contribute to helping in the control and prevention of pandemics?" [130]. They used the micro:bit to develop a gadget to help people during the pandemic. The gadget was a necklace that would detect when another person was too close to them. They used the magnetometer and the LED screen on the micro:bit to create the necklaces. The researchers used the creativity spiral proposed by Resnick [172] which includes stages of empathize, define, idealise, correct, prototype and test. The authors state that "teachers must use activities in which students can put coding into practice by solving problems". They also state that teachers can amplify the potential of programming, as it promotes the development of **computational thinking**.

Nikou investigated whether a makerspace program in a primary school motivates and engages children [156]. They ran a 6-week long workshop which included building projects with the micro:bit. A student reported that programming the micro:bit was "**easy** and **fun**". The students also found programming useful as "it helps solve **real-life** problems". They analysed students' engagement and concluded they were **engaged** under the engagement subscales of cognitive, behavioural, emotional, and social.

Ethics and privacy

The final question that is looked at is around ethics and privacy. Given that the micro:bit can store data, was there any discussion around how they stored this data, for how long, did they gain consent in collecting the data, etc? Was there any ethical discussion in general, as part of the activity?

Three codes were used here: Consent, Ethics in content, Both. If consent was acquired either by parents or children or ideally both, this was recorded as Yes. If any ethics were discussed in the activity with the stakeholders, this (Ethics in content) was recorded as yes. If both consent was acquired and ethics was discussed then both was recorded as yes.

Almost half of the papers (42 of 86) discuss ethical issues they have considered. These include, but are not limited to, obtaining consent to participate in the research, risks of participating in the research or using the proposed solutions, privacy & data usage and storage. 30 studies explicitly state consent either from the parents of the student participants or both. 6 studies discuss ethics in the content of the study, but not whether consent was obtained and the final 6 discuss ethics in the content and state that consent was obtained.

Papers that explore the teaching of ethics around micro:bit/physical computing tend to touch on IoT devices. Tamashiro et al. [195] created an IoT kit “Orbit IOT”, a micro:bit-based toolkit, to teach both the technical and societal implications of IoT. In their related work they discuss how the majority of other IoT initiatives cover how the technology works, not its impact on individuals and society.

The kit allows the micro:bit to be connected to the Internet and to control household devices such as lightbulbs and fans. It also includes a website that allows students to see the movement of data in the network. The study included a daylong session with students to learn how to use the kit and a second day to implement what they had learnt by “pranking” their teacher from a different room. The students used webcams to track their teacher during a lunch break and turned on/off lightbulbs and fans to prank them. The pranking activities gave the teachers opportunities to discuss hacking, security and ethics with the students. The website visualising the movement of data helped to support troubleshooting, teaching about networks and helped with the discussion around ethics.

The Orbit IoT kit has a lot of potential to teach data science to children. The study by Tamborg et al. was undertaken with secondary school students but at the lower age range (12 to 13).

Knowles et al.'s paper explores students imagined use of the micro:bit for IoT [119]. They discuss how the children's ideas and imagined uses can inform how to design for their IoT **privacy, security** and **safety**. While parents purchase IoT devices for their children, the adaptability of programmable IoT devices "makes it difficult for parents to anticipate how their child may ultimately engage with the IoT". Therefore, children need better understanding of IoT privacy, security and ethics. In a follow-up paper where the students coded the micro:bit they note how children ages 9–11 have little understanding of what constitutes personal data and "*have very little understanding of their own privacy or that of others*" [118].

Kumar et al. [124] deliberately avoided any kind of machine learning movement tracking with cameras as most of their students came from *"a variety of marginalized backgrounds*" and they did not want to implement a tool that "*can introduce risks around security and surveillance*."

3.2.5 Analysis

This section is a systematic review of micro:bit in the literature. The gaps are revealed within this research. Below is a summary of the literature.

Researchers state the inexpensive **cost** of the micro:bit as the most common reason they chose it, followed by its **accessible onboard sensors** and **ease of use**. Many state they chose the micro:bit as there was no additional software to install [84, 135] except when schools use tablets such as iPads and the only way to program the micro:bit is to install the MakeCode app [206]. When comparing the micro:bit to other solutions Dias et al. [55] state that other available solutions are "*too difficult to understand, overly expensive for school budgets or may require IT infrastructure that is lacking from the majority of public schools.*" From a safety point of view, researchers also chose the micro:bit because it is not connected to the Internet and does not have a camera which offers a safe space for children to learn [55, 124]. These pragmatic reasons led the researchers to choose the micro:bit over other devices or tools.

The **ease of use** of the micro:bit and its embedded accelerometer has made it an effective data collection tool for machine learning. Machine Learning (ML) and Artificial Intelligence (AI) are popular topics now with many governments and institutions discussing how they can be taught in schools and developing resources and educationally sound lessons. The NCCE has developed an online course for teachers "Introduction to Machine Learning and AI". The Danish government have a similar scheme to train teachers [207]. However, Druga et al. state there is a lack of **transparency** in these tools as artificial intelligence and machine learning are typically complex "black boxes" where a lot of the underlying processes have to be abstracted [58]. Bilstrup et al [23] argue that to "*understand the systems students engage with every day, and the political and ethical questions concerning ML, students must engage with technology-close questions related to the construction of these systems*." They also go on to highlight how these systems shouldn't be simplified: "*Educational tools should also expose the more complex parts of ML and the difficult choices should not always be fixed to ease the students in the construction process*".

Studies involving machine learning showed high levels of engagement but are narrow in their application domain. It would be difficult for the educators to adapt the tools generated and apply them to different scenarios, because the tools are scaffolded too much.

In [181] taking part in micro:bit workshop helped young women's **programming ability belief** increase. Increasing young people's programming ability beliefs is important since it is very strongly related to the formation of career interest (e.g. [60–62, 131, 132, 139]. Research has shown that students *believing* that they have what it takes to do well in a career is more important for developing interest than their actual ability, as measured by grades [11, 29, 132]. This is supported by the 2017 Science Education Tracker report's [92] finding discussed in Chapter 2. Students believe Science, including Computer Science, is difficult and this perception of difficulty and lack of confidence is higher amongst female students, who in turn don't choose to study these subjects. If the micro:bit can increase young women's programming ability belief it can increase their engagement in the subject.

Other ways educators and students found the micro:bit engaging is they believe it is **fun** [84, 114] and because it supports **real-life** learning [84, 109, 130, 190].

Through IoT applications and smart cities students were able to create a real-life use for the micro:bit. This connection to the real world created **engagement** and **motivation**. The students saw **meaning** [35, 153] behind the activity. As Charlton et al. [35] state "*Students engage when ... their learning has meaning*."

Much of the research using micro:bit is not situated in education conferences but in human computer interaction and design. Education pedagogy and methods were not often discussed. However, in using the micro:bit, researchers did use a range of methods discussed in Chapter 2. **Computational thinking** was a main subject researchers were using the micro:bit with [55, 77, 89, 134, 173, 206, 221, 226]. They used methods such as **algorithms** [206], **logic** [55, 206], **decomposition** [90], **debugging** [134, 206, 221, 226], **tinkering** [173, 226], **abstraction** [226] and **persevering** [221].

Outside of directly mentioning computational thinking, activities used **pattern recognition** [23, 109], **debugging** [204], **decomposition** and **collaboration** [174] with the micro:bit to support students. There are also several activities that don't teach programming to the children, but use it as a tool [19, 124, 160, 174]. While the majority of the subjects the micro:bit is used in are Computing related, there is some **cross-curricular** learning [19, 109].

Another approach to teaching in Computing and physical computing mentioned previously is **project-based learning**. In the review, just two papers mention this approach [108, 194] but the author believes many of the activities used project-based learning methods. Many of the students learn through problems with real-word practices and construct a final product.

3.2.6 Gap analysis

This section looks at the gaps in the above research around micro:bit and education.

Gap 1: Longitudinal studies and progression

Most studies are one-off studies with students. There is a gap in researchers undertaking longitudinal studies with the micro:bit. While Jin states that physical computing can place a higher cognitive load on students [110] their study was still one lesson that ran once. Research by Sentance et al. [185] on the BBC micro:bit recommends that lessons are run over several weeks: *"Attention should be given to* *the provision of resources that span a period of weeks, have supporting assessment activities, and are linked to the curriculum: this will be preferable to the provision of many short un-sequenced activities."*

While most research with the micro:bit involves one- off studies, teaching with the micro:bit is typically done over several lessons. The micro:bit website and the NCCE website both have micro:bit lesson plans that span a number of lessons over a period of time. The micro:bit website [144] has units of work for 7- to 11-year-olds, 11- to 14-year-olds and 14- to 16-year-olds. Each unit has between 2 and 6 lessons.

When much of the work is completed over one or two lessons, there is very little research on progression with the micro:bit. There is no understanding of how children can progress in their learning using the micro:bit as a tool. Researchers note it is a good tool for **progression** given it can be coded in blocks and text. But there are very few activities where researchers handle progression using either a different activity with the same children or the same activity with children of different ages.

Gap 2: Cross-curricular education

The micro:bit has been applied to **multiple subject** domains, with Computing being the most prominent subject. It has been used as a **cross-curricular** tool to help engage students with other subjects to help fit more teaching into the "*crowded curriculum*". IoT and smart devices are well served in the literature, but semi-automated data collection is not. The students are not given the opportunity to collect the data themselves with the micro:bit as a tool, a tool that they could code as well as use. As stated previously using a dataset that is real and relevant is engaging to students. The micro:bit could be used as a data collection tool for students to collect real data from their local environments.

Gap 3: Data storage and analysis

None of the researchers in this review used the micro:bit's data logger to store data. Where data is mentioned, it is gathered and used straight away to train machine learning models [23, 196]. The data logger on the micro:bit can store multiple sets of data which in turn can be analysed and visualised.

Being able to store data and access it at a different date is an important feature for teachers wanting to teach more about data science.

Gap 4: age range

It is surprising that 50 out of 76 studies that stated the age of the participants were children aged 0 to 11, children in primary schools. The original BBC micro:bit project was aimed just outside that age range (11- to 12-year-olds) for children in the first year of high school. Yet most of the research with micro:bit is aimed at children in primary schools. Researchers use the micro:bit as both a tool and a programmable device with primary aged children. In many of the studies the children didn't code the micro:bit to use it [59, 124, 160], showing its use as a tool as well as a coding device. As much of the research is with primary-aged children this shows the potential of the micro:bit in this age group. The researchers have overcome the complexity of the micro:bit and successfully delivered lessons to this age of children who may have teachers not skilled in Computer Science.

Gap 5: Ethics

There is a worrying lack of ethics mentioned in many of the papers. Given the ability of the micro:bit to store personal data, there are legal and ethical ramifications of using this device in the classroom that everyone (researchers, educators, parents, children) need to consider. If students are being taught to store data using the micro:bit, they must also be taught about data, data privacy, ethics and ownership as well as consent.

3.3 Conclusion

This chapter provided a summary of the micro:bit capabilities and systematically reviewed the use of the micro:bit in research.

In summary, the micro:bit has proven itself to be a good tool to use for data collection in schools. It is **cost effective**, **accessible** to both students and teachers. It is sold worldwide as well as being present in a lot of UK primary and secondary schools. It can be extended with additional hardware through its edge connection. There is no additional setup to add an accessory to the micro:bit, no wires, additional electronics, or physical hardware like screws. The micro:bit can be inserted into (and out of) an accessory in seconds by a child.

The literature review found that the micro:bit has been used in multiple studies around the world in classrooms, museums and after school clubs, with children as young as three, but mainly with primary-aged children. Activities are mostly run once or twice over a handful of hours. Children code the micro:bit mainly use Microsoft MakeCode as well as using it as a tool without coding it. Researchers use the micro:bit as it is cost effective and easy to use. Its embedded sensors and the ability to add extra sensors is appealing to them. Researchers have used the micro:bit to teach mainly Computing topics but also other topics such as sport, Physics, and wellbeing.

There is a current gap in the research with micro:bit to do longitudinal studies over time and over different age groups including multiple subjects and including ethics in the teaching. A full cycle of lessons over several weeks will give better insights into students learning with the micro:bit. Covering multiple subjects will help situate the learning in real life and help teachers fit the learning into a crowded curriculum. The micro:bit data logger has not been utilised in research. Finally, teaching ethics including privacy and security is an important part of every subject in the curriculum; Computing, physical computing and data science should be no different.

Chapter 4

Requirements

This chapter describes the requirements of a system that uses physical computing to support educators to teach children about data science. It uses multiple sources to collate these requirements. The first section gathers requirements from educators in schools, based on how they currently collect and analyse data. The second section is a case study: Energy in Schools. Energy in Schools was a project that supported educators to utilise the micro:bit to collect energy data for analysis in schools. How the project worked in schools is analysed and requirements are created from its evaluation. The final section of this chapter includes a short summary of the requirements gathered from previous chapters of this thesis, and a summary of the design concepts before collating the final list of requirements together. This chapter also looks at the methodology for this thesis – how the author worked with teachers in a novel technology design process.

4.1 Gathering requirements

To understand how data collection and analysis is done in schools a number of educators were interviewed about their current techniques. This was one approach for working with teachers in a technology design process: ensuring they are involved in the requirements stage to gather their insights and to be able to understand any problems they have with teaching data in their classrooms.

Three researchers (RA2, RA6, RA7), three educators (T1, T2, T3), two designers (D1, D2) and two computer scientists (C1, C2) were interviewed as part of an iterative co-design process which continues into Chapter 5.

The first section below looks at how teachers preform data collection in schools and the main issues they face. All the teachers reported doing manual data collection. The second section explores an attempt to use a commercial accessory with the micro:bit for semi-automated data collection.

4.1.1 Manual data collection

In Chapter 6 an evaluation is undertaken through a pilot study and a main study. The teachers from these schools were interviewed about how and why they collect data. All the teachers stated that they undertook some form of manual data logging. In the lessons the students were collecting survey data. They were counting animals, (e.g. birds, bees) or plants that they had learnt about in their lessons. In another set of lessons, the students were counting traffic; the different types of cars on the road outside their school. The data collection is manual, there are no sensors or technology involved. The teachers would print and distribute paper sheets to students. The children would use a tally system to count the data they were collecting, i.e. they would use marks grouped in bunches of 5s to represent numbers I is 1, II is 2, III is 3, IIII is 4 then ~~IIII~~ is five and they would start counting again at I.

The children who took part in these lessons complained how difficult it was using paper sheets outdoors. Below are quotes from a group of Year 5 children from the main study school.

"I find handwriting really hard"

"You had to press it down on your hand. Leaning was difficult."

"And people would rip it."

"We couldn't lean it, it was well wet."

At the end of the data collecting session, the educators would collect all the sheets and add the tallies together. Back in the classroom they would write the totals on a whiteboard to show the students the data to analyse it. This process was long and error prone. In bad weather the sheets would become unusable as they would get wet and often tear. Students would lose the sheets, or they would blow away in the wind if not secured.

It would take a long time for the educator to collate all the data together. The final data, not being digital, couldn't be easily sorted or filtered. The students couldn't see their data, only everyone's data added together. It also wasn't saved for future use.

All the learning had to be done on the same day the data was written on the board as in many cases it would be wiped off the whiteboard at the end of the day.

Many educators are reluctant to use tools such as spreadsheets to display data. They are not trained in using spreadsheets and don't feel confident presenting the data to the class on a spreadsheet or trying to teach their students how to use them. One school was using Chromebooks and the teacher wasn't aware they could still view spreadsheet data on Google Sheets software. They thought because they couldn't install Microsoft Excel on a Chromebook they couldn't use spreadsheets.

4.1.2 Semi-Automatic data collection

Researcher RA2 was involved in a project where children from multiple schools collected biodiversity data from around a local park. They wanted to show all the data together to the different schools, mapped by location.

RA2: *We were trying to make it so the children were able to see all of the data together.*

They had tried using software and Miro boards, but found the children were too young and didn't understand how the software worked. They tried to use Google M aps to map the data with a single login shared between different schools. However, the school they were working directly with had tablets and the mapping only worked on computers or laptops.

RA2: T*he setup is not the same on a tablet.*

RA2 also stated how having the same login for everyone was not secure but was the only way to share the data.

For collecting the data RA2 used the micro:bit as a data collection tool, slotting it into an accessory to gain more buttons. They used Kitronik's game controller to add 6 more buttons to the micro:bit, see Figure 33. The author supported RA2 in coding the game controller.

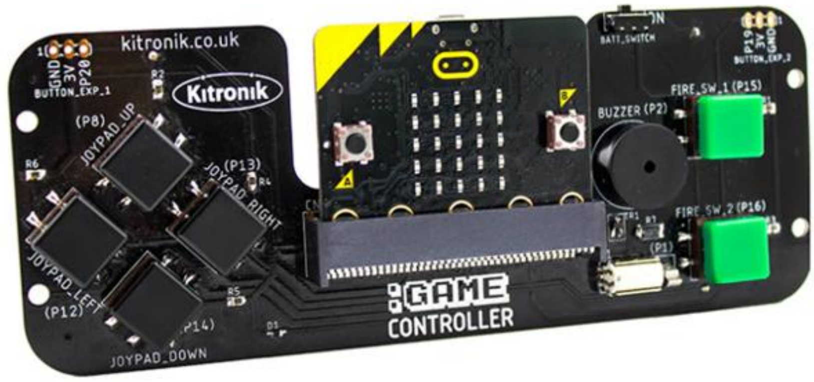


Figure 33 Kitronik game controller[14]

RA2 combined the controller with plants sheets from the nature society, the Woodland Trust. They would label the sheet, for example on Figure 34(a)[15] next to blackberries they would label "UP" to indicate if a child saw some blackberries, they needed to press the UP button on the controller.

Figure 34 Woodland Trust summer spotter sheet

[14] https://kitronik.co.uk/products/5644-game-controller
[15] https://treetoolsforschools.org.uk/activitymenu/?cat=plant_id

There were some problems with using the controller:

RA2: *And the problem with that is that you can't visually see what you're recording*

The image was not close to the button, the students must search the sheet for the item they had found then search the controller for the correct button. They had to hold the sheet and the controller which became difficult after an amount of time. RA2 noted how they (the researchers) wanted more than six buttons. They stated even though children may not be able to handle lots of buttons, the risk of missing items was bigger.

RA2: W*hen you go out and do bird surveys there's no guarantee you're going to see however many birds. You give a kid this and said "you're handling these 4 birds". They could definitely handle that on their own. But the problem is you might not see those 4 birds and that kid over there sees everything and they can't keep a handle of it*

They wanted more buttons than the six available on the controller.

RA2: *We have to have that many buttons so you'd have choices. And you'd also have the flexibility for the birds. And plants, if you were doing plant surveys.*

4.1.3 Summary

This section described the different ways educators and researchers collect and analyse data in classrooms. It highlights some of the issues they have had. Manually collecting data is time consuming for the teachers. Having to collect and collate all the students' data takes time and effort. One might question the accuracy of this data if some of it is lost or damaged due to wet and/or windy weather. If the students know their data wasn't collected because it was lost, they may disengage from the topic. When the teachers don't store the data electronically, there is no way for the students to filter or sort the data. They can only see all the data all at once. Given the temporary nature of this data, there is no opportunity to look at that data longitudinally. Students can't compare data from last week to this week, if last week's data isn't stored anywhere. They can only analyse the data collected from that one scenario. Surveying is an important skill for students of all ages to learn. Amongst other skills it teaches them observation skills, focus, counting, tallying.

As seen above researchers using commercially available technology also struggle to collect, store and analyse data. Different schools use different types of devices, and these are not always compatible with whatever technology is being used, e.g. the iPads unable to view the Google Maps data. Tools developed for data science need to work for these different types of devices.

Storing and sharing the data seems to be the biggest barrier for both manual and semi-automatic data collection techniques. Being able to store then view multiple sets of data from multiple children needs to be possible, easy for a novice to complete and error free. The students should be able to access this data, their data easily and securely. There should be no password sharing between students or schools to support data security.

4.2 Energy in Schools

This section introduces a case study called Energy in Schools run by Lancaster University, including the author. Energy in Schools was a project that supported educators to utilise the micro:bit to collect data for analysis around energy use in primary and secondary schools. This case study is used to further refine the requirements. The three phases of the project, the technology used, the lessons taught and feedback from the stakeholders are covered.

Energy in Schools was a UK government-funded initiative aimed at "*helping UK schools reduce energy usage and costs, whilst educating young people about the importance of energy efficiency*" [66]. It consisted of a consortium of four organisations: Samsung, Centre for Sustainable Energy, MyUtilityGenius and Lancaster University.

Over three phases the project aimed to:

1. Help schools save energy via an energy management IT platform with tools and interfaces for all the stakeholders including pupils, teachers and administrative staff.
2. help schools save money by switching their energy tariff.
3. Provide teachers with materials and lesson plans built around coding the micro:bit to help pupils solve real world problems related to energy and the school environment. This was to improve pupils' and teachers' understanding of climate change, energy and Computing.
4. Support pupils to lead the way in supporting whole-school behaviour change through an energy champions training and support programme.

A core aspect of the project was to involve all the stakeholders within the schools, management, teachers, support staff, students and their parents. For example, installing a temperature sensor in a classroom would show management how hot the room was getting during the day, but they needed the regular users of these rooms: support staff, teachers and pupils to change their behaviour around opening and closing doors and windows which can affect temperature.

This project was a first attempt by researchers including the author to tackle the challenges of teaching an aspect of data science using physical computing in the classroom.

The project was divided into three phases (1) preliminary research, (2) initial deployments to a small number of schools and (3) wider deployments to the whole of the UK. In phase one the initial team worked with a school to design two platforms: an energy platform and the educational platform.

Both platforms were installed and used by three schools in phase two. Training was provided for the different sets of users: site managers, pupils and teachers. Feedback sessions on installation and use of the tools were held to help improve the design of the platforms. In phase three, 20 schools took part in the project, 14 primary schools and 6 secondary schools. Each school was given equipment to run the project, see Table 15.

Item	What was it used for?	Quantity
Samsung Hospitality TV	School dashboard showing energy usage and school league tables	1 per school
Samsung Phone	Used to setup SmartThings Devices Connects micro:bit hub for learning sessions	1 per school
SmartThings hub	Gateway for all sensors and devices	10 per school
Multipurpose (door) sensor	Activity and temperature monitoring	20 per school
Motion sensor	Activity and temperature monitoring	20 per school
Power outlet	Control of outlets as part of IoT learning	5 per school
Colour light bulbs	Control of light bulbs as part of IoT learning	5 per school
micro:bits and batteries	IoT coding sessions: allow pupils to create IoT sensors	30 per school
Aeotec energy monitor	Measures school energy usage	1 per school
Other accessories	TV mount, clip-on lamps, courier costs	1 per school

Table 15 Equipment given to each school in the Energy in Schools phase 3

4.2.1 Energy platform

The Energy platform comprised a portal to help staff manage energy and heat usage, the Energy dashboard (see Figure 35), a tariff switching portal comparing different tariffs at time-of-use, a teacher and student portal which included an energy display screen to support behaviour change by engaging building users with energy use (Figure 37), an energy advice materials, energy champion training (described further in section 4.2.2) and an initial energy audit.

Figure 35 Energy dashboard

The energy dashboard in Figure 35 includes the functionality to view live or historic data and to view the data in more detail over time, comparing different sensors to each other, (Figure 36) .

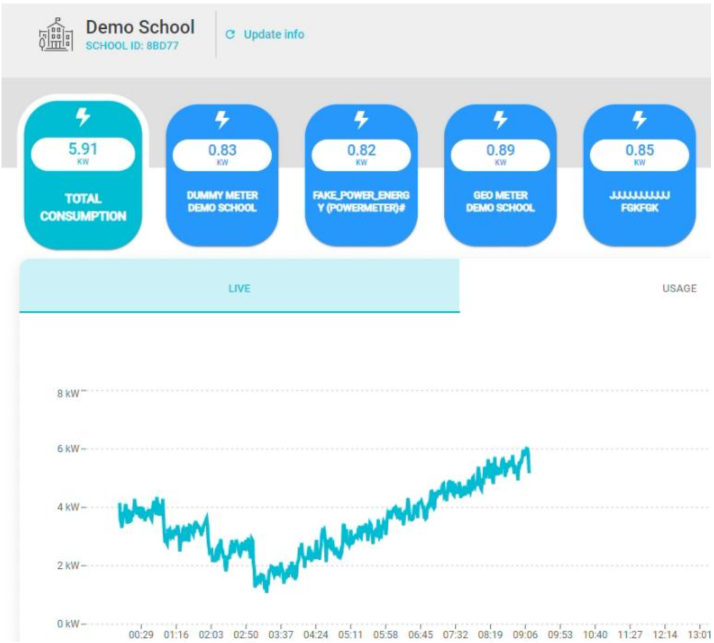


Figure 36 Sensor data in detail

Figure 37 Teacher and student portal

In phase two of the project the sensing and energy monitoring were undertaken using the Samsung SmartThings system. A SmartThings hub was installed in the schools along with energy and gas meters, smart bulbs, door sensors and temperature sensors. All this data was uploaded to the SmartThings cloud and interfaced to the online Energy In Schools server (EIS). In phase three of the project micro:bits were added as sensors and a micro:bit bridge attached to the teacher's computer sent all the micro:bit's data to the Energy in Schools server.

Figure 38 shows all the stakeholders in the project in phase three: students, including energy champions, teachers and the schools' energy manager. Through their classroom lessons and the Energy Champions lessons students (a) deploy Smart Things sensors (b) and the micro:bits (c). These are connected to the Smart Things Hub (d) and the web hub (e) respectively, which in turn upload all their data to the Energy in Schools Server (f). The teachers (g) deliver lesson plans (h) and support the energy champions. As well as collecting data, the students report their findings to the energy manager. The energy manager (i) and the teacher can view the energy manager dashboard and tariff comparison tools (j) to see more in-depth data around the school's energy use including data from electricity and gas meters (k). The whole school, all stakeholders and parents, could see highlights of this data as well as a competitive graph on the large TV screen provided to each school (l).

The TV screen showed highlights of the schools' energy use. Schools were encouraged to place the screen in high footfall areas so all stakeholders could see the data.

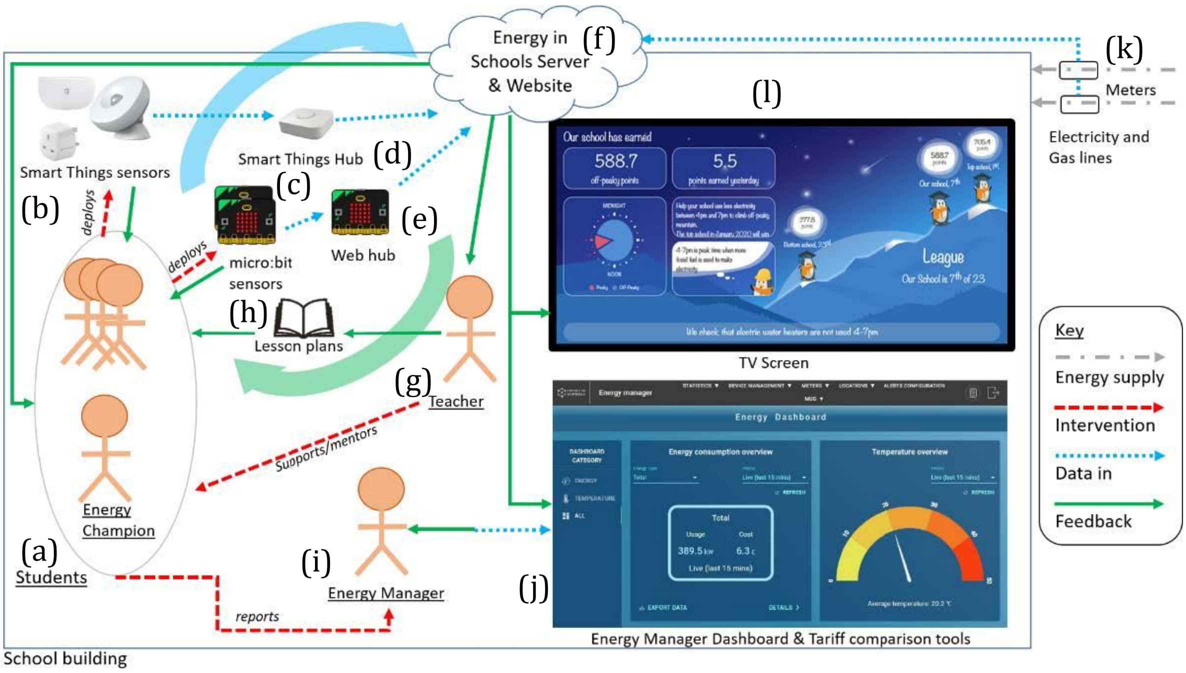


Figure 38 Energy in Schools stakeholders and systems

4.2.2 Educational platform

The Educational platform comprised of teaching and learning resources for Key Stage 2 and Key Stage 3 that were aligned to the curriculum and were supported with a set of micro:bits, an educational environment to teach the basics of coding and to solve real world problems. There were two sets of lessons: energy lessons and IoT and micro:bit, and further activities to extend the lessons and for the energy champions to explore. These activities overlapped with the six energy champion lessons. All the lessons were cross-curricular over Computing, science, Geography and Mathematics. They were for students in primary Key Stage 2, (Year 6) and pupils in secondary (Year 7). Most of the lessons were 60 minutes long. The lessons included a PowerPoint presentation, activity sheets and teacher notes.

Energy lessons

There are 4 energy lessons titled:

1. Where does energy come from?
2. Global warming and climate breakdown.
3. What appliances use the most electricity.
4. Generation mix.

These lessons teach students the theory behind energy: what it is, different sources, how it is created and how it can be measured in kilowatt hours. The lessons cover the pros and cons of the different types of energy, including renewables, fossil fuels and nuclear energy. The students learn what they can do to cut greenhouse gas emissions and reduce climate change. They learn about appliances at school and at home and how to calculate energy costs. The final lesson includes programming the micro:bit to connect to the Internet and download data.

This final lesson starts with a reminder of the different types of energy source and how energy is a mixture of these. It shows the different ways the UK generates energy and what energy it imports and exports. A pie chart of electricity generation mixes on a wet summer day which includes wind, solar, hydro, biomass, imports, gas and nuclear is shown, see Figure 35.

The next part of the lesson codes the micro:bit to display the carbon generation mix for different energy sources for the UK, see Figure 39 for the slide explaining the code. The carbon generation mix is the combination of different energy sources used to generate electricity. The energy sources include fossil fuels such as gas and coal, nuclear, and renewable energy such as solar and wind.

The micro:bit is coded using a custom version of MakeCode built for the Energy in Schools website. The students and teachers had their own logins to access this version of MakeCode. In Figure 39 on the left there are the new menu items: Weather, School Energy, Carbon, Samsung IoT, Share and ISS.

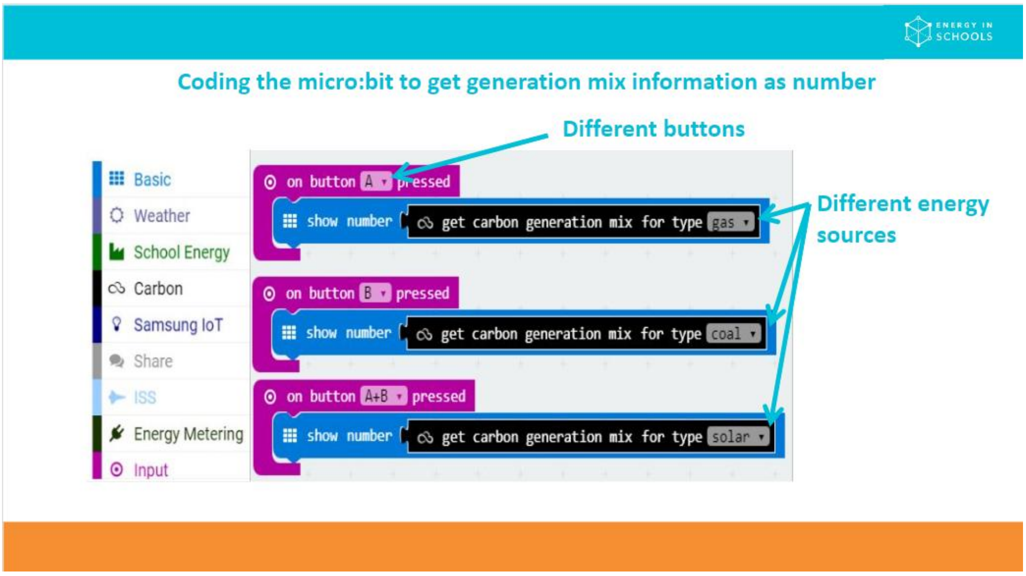


Figure 39 Coding the micro:bit to get generation mix

When the students have completed the programming task, they are asked to create a pie chart with all the data. They then continue programming the micro:bit by creating a conditional statement using the solar data, see Figure 40.

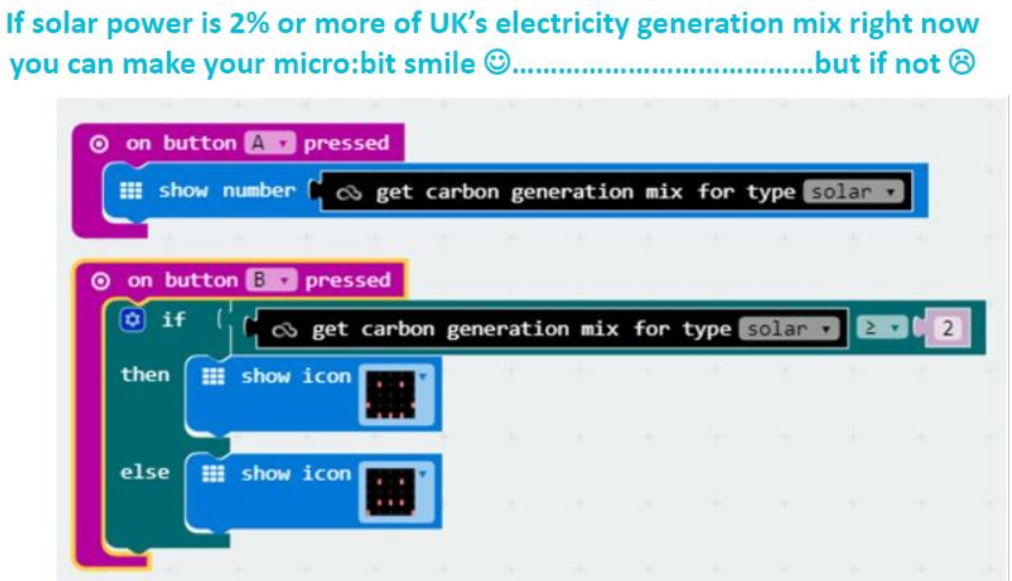


Figure 40 Programming a micro:bit to smile if solar is greater than 2%

The lesson finishes with a description of what "Peak Time" means for electricity bills and how students can cut electricity use at these peak times.

IoT and micro:bit

These are a set of 4 lessons for learning about the Internet of Things and using the Samsung SmartThings sensors and micro:bit to connect to it. The two introductory lessons, Introduction to the IoT and IoT for Good, introduce what the Internet of Things (IoT) is to students. They learn about sensors, actuators and logic. Then they learn how IoT can be used for good, using an example of water pumps in African countries. Smart water pumps help predict if the pump is going to fail and notify a management team, which means they can be fixed in a reduced amount of time [191].

The next lesson is: Introduction to Samsung SmartThings system. This lesson teaches the students how to use the SmartThings system. Students log into the educational portal and view the data for each of the smart sensors, for example they could view how many times the motion sensor went off at the start of a lesson and estimate how many people were in the classroom. For the same day and time they could view the temperature in that room and question if the temperature went up with more people and whether it was too hot. Activity sheets were created to help students compare this data and come to conclusions.

The final lesson is The micro:bit as an Energy Saving IoT Device. This lesson provides several different activities for programming the micro:bit as an Energy Saving IoT device. The different activities include:

- Door magnet sensor which uses a magnet and the magnetometer on the micro:bit to detect when a door is open/closed. Keeping doors closed can help keep heat in the classroom.
- Light sensor to detect when a light is on in a classroom. This includes how to calibrate the micro:bit's light sensor to the room it is in.
- Temperature sensor to detect how warm/cold it is in a location. Again, this includes how to calibrate the sensor. The micro:bit's temperature sensor tends to run hotter than the surrounding temperature because of the electronics on the board.
- Electricity use. This activity attaches the micro:bit next to an electrical cable to detect whether the device is powered on. This is a simple way of telling if an electrical device, like a kettle or a lamp, is on.

- Fossil fuel lightbulb changer. This activity connects the micro:bit to both the Internet (as they did in lesson 4 in the Energy Lessons above) and the Samsung SmartThings bulb. If the carbon index level is high the bulb will turn red, else it will turn a different colour, see Figure 41. This action would be run when the button A on the micro:bit was pressed.

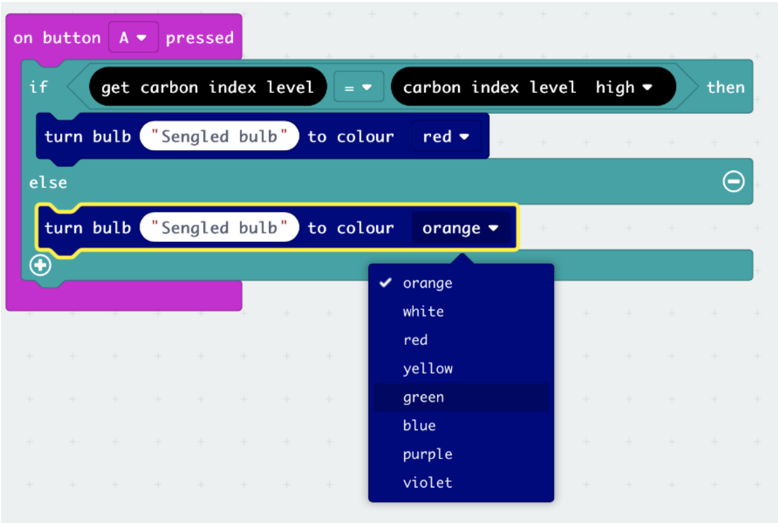


Figure 41 Coding the micro:bit to change a lightbulb colour

Further Activities

Further activities were provided for teachers to use as extension tasks in lessons, extra-curricular clubs and for the energy champions. These activities give more examples of analysing energy data. They use the Energy in Schools website to look at data, analyse it and make conclusions.

One activity uses the Energy in Schools website to compare school energy use by day, week and month, Figure 42. In the lesson the children learn how to calculate electricity use in kWh and pounds and what "Always on" appliances cost. This lesson can be paired with energy competitions to see which class or school can reduce the most energy in a week/month.

Students can use this real life data to analyse the energy use over time for their school to find potential examples of energy being wasted, e.g. being used after hours or over the weekends.

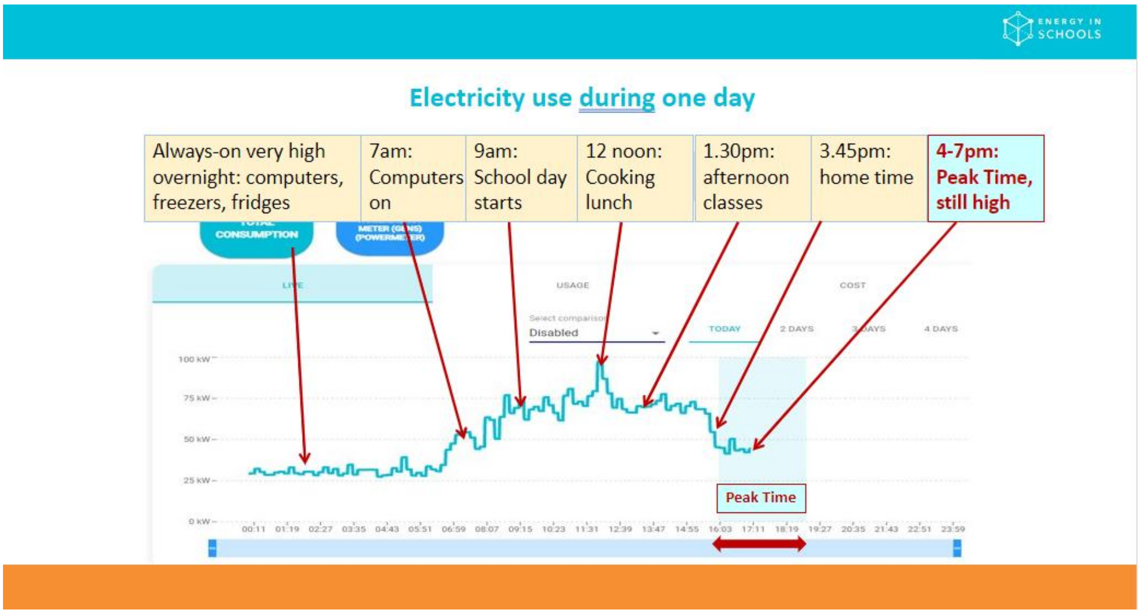


Figure 42 A sample of measuring electricity use during one day

4.2.3 Energy champions

Energy champions are pupils who perform extra-curricular energy activities. They were recruited from the main pupil population and were volunteers. Energy champions take a lead role in:

- Planning and carrying out projects that use the micro:bits, sensors, smart devices and Energy in Schools website to find out more about their school and its energy use.
- Taking practical actions to reduce energy waste and greenhouse gas emissions.
- Getting staff and pupils involved and communicating results – through presentations and displays.

These activities were given to the teachers to help support the energy champions. Some of these activities could also be used as extension activities in class. Not all Energy Champions participated in the taught lessons above therefore the first lesson is an introduction to the Energy in Schools website and how to code the different devices. The second lesson involves students doing their own energy audit so they can identify problem areas in the school.

In the previously mentioned lessons it is not possible to deploy sensors in different classrooms to where the teaching is happening, and the curriculum schedules didn't allow for collection of data over a long period of time. The energy champions, however, could do this type of data collection and report back their findings to school management and the rest of the school.

4.2.4 Evaluation

In this section the aims of the Energy in Schools project are reviewed below. The Energy in Schools project was successful in many of the areas to:

1. Help schools save energy
2. Help schools save money
3. Help pupils solve real world problems related to energy and the school environment.
4. Support pupils to lead the way to support whole-school behaviour change through an energy champions training and support programme.

However, there were some critical technical problems with the project that hindered a lot of these successes.

Save energy

In phase 2 there wasn't any historical data to compare energy to for 2 of the 3 schools. For the school that had historical data, they saved 3754kWh electric and 5616kWh gas over the year. This saved the school £693 on its fuel bills. The school followed the advice of the energy audit conducted by Energy in Schools and installed LED lighting. This school didn't engage in the lessons, but staff did use the platform to spot that the heating was still on in the evenings and the weekends. An engineer was called to rectify the issue, which cut gas waste. In phase three, 8 out of the 14 primary schools reduced their always-on usage.

Save money

In phase 2 two schools used the Energy Platform's tariff switching portal to compare new energy contracts. In the past they used a broker recommended by the local council to choose their energy tariff. Using the tariff switching portal saved both schools money. The first school saved £4,775 per year on gas and electricity contracts. The second school saved £1,614 on changing their gas contract.

In phase three, four of the schools that provided data saved money, with one school saving £5,451 on their electricity bill over the year.

Real world problems

The lessons taught students about energy, energy use and climate change. They learnt about real world problems related to energy and were taught ways in which they could help reduce climate change at home and at school. While the lessons did include studies of real-world problems around the world, like African water pumps, they also related directly to the students' environments. They taught the students how to setup physical sensors to record their environment. This data helped students understand the real world problems their school has in heating different school buildings, providing electricity for computers and lighting. The students were able to calculate how much it cost their school to buy energy and gas for heating, lighting and electricity. They were taught how to use sensors to gather data about energy use in their school. By the end of the lessons the students could view real world data gathered from these sensors in their classrooms to analyse energy use and most importantly energy waste. One teacher noted how coding the micro:bit helped relate them to real life: "*new ways of showing students how coding can be part of real life".*

Energy champions

Energy champions were actively engaged in 12 schools, 2 secondary schools and 10 primary schools with 80-90 students involved in total. In 9 of these schools, energy champions reduced energy waste directly by initiating or increasing switch off activities such as turning off lights, computers and whiteboards, turning down heating controls and checking iPad and laptop chargers. They used the TV screens provided to prompt fellow students and staff to be more energy aware. They took part in competitions with other schools to measure their energy use and gain "peaky points" by bringing down the schools' peak energy consumption.

One school's energy champions monitored electricity use, noting that computers were being powered down at 8pm. The school changed its power down to 4pm, reducing its carbon intense peak time usage.

Another set of champions used multiple temperature and door open/close sensors in a classroom which was too hot to identify the cause. They discovered that heat was entering the room from a heater outside the classroom in the hallway because the door remained open. They encouraged the teacher to close the door to cool the classroom.

Technical problems

In phase two of the project many of the challenges of teaching using physical computing in schools became apparent, particularly practical issues around new hardware deployment.

The Samsung SmartThings devices needed to be setup using an app on a smart device, a phone or a tablet. Before they can be setup, a SmartThings account is needed. From the feedback interviews one technician reported that: “it is not possible to set up a SmartThings account on a tablet, so some staff had to do this on school or personal smart phones.”

These Samsung SmartThings devices had to be constantly connected to the school’s Wi-Fi to work. Schools reported the sensors disconnected on an irregular basis once they had been paired, possibly due to signal issues. Some sensors could not be placed in areas of the school with poor Wi-Fi signal. Some of the secondary schools had multiple buildings across their entire site and found this a challenge.

The teachers had limited knowledge on how to setup these technologies. They were reluctant to use their own personal phones, and many schools don’t have a school mobile phone to use. If they had school tablets, these were often locked down with an admin password to prevent students from installing or downloading malware. In some schools the teachers didn’t have access to the admin password as the tablets were managed by an external technology company. This is why a mobile phone was added to the equipment list in phase three.

The schools’ restricted Internet access prevented many of the devices from sending data out of the school. Even the schools with IT teams on site, did not have control over their firewalls. This was outsourced to their Internet Service Provider (ISP). In some schools it was not possible to start the project on time due to delays in school’s IT providers whitelisting the SmartThings hubs.

In phase three, 20 schools were recruited to the project. The smart sensors for doors, windows and temperature were dropped completely. It was not possible to provide the in-depth technical support for all these devices to all the schools in phase three. More micro:bit sensing activities were added to the lesson plans.

To upload data from the micro:bits to the internet, the micro:bits had to be physically brought to the micro:bit hub that was connected to a teacher's computer. The data was sent over radio from the micro:bits collecting data to the micro:bit hub. The hub would then upload the data to the Energy in Schools server. This upload of data took time and meant that the teacher's computer was out of action during the upload. In depth instructions were needed describing how to upload the data from the students' micro:bits to the hub to the website. It wasn't a simple task to complete.

4.2.5 Analysis

The Energy in Schools project highlights how important **pragmatic** solutions are for schools. Physical computing devices that need setting up through a mobile app can cause problems for schools and teachers. Many schools do not have mobile devices and teachers using their personal devices risk sharing their personal data with students and/or storing student data on their personal device that may be insecure or vulnerable to unauthorised access. For example, setting up a Samsung Smart Things sensor requires installing their app on a mobile device. A teacher may install it on their personal phone. The data recorded by this sensor, whatever data that may be, is now stored on the teacher's personal phone which leaves the school building every day and may be accessed by other people. Energy in Schools overcame this problem by providing each school with a mobile phone but also pairing the SmartThings sensors to the hub before they arrived at the school. The Energy in Schools technical staff found it too difficult to debug device setup and pairing remotely during phase two. They paired the devices for the schools before shipping them to the schools in phase three. On top of this another big change between phase two and three was the addition of manuals to the platform. Manuals and videos on how to setup the SmartThings hub, micro:bit bridge and sensors were created and added to the platforms for school technicians and educators to setup and manage these devices independently. Samsung staff still provided support for the IT staff, but the hope was these manuals would cut down on the number of in-person visits needed.

This project was funded by the UK government so all the equipment was provided, cost wasn't an issue but would be if it was to be repeated in other schools. The technology was also bespoke, it was not accessible to others outside the project. Installing the hardware on the school network was not achievable by staff on site. The teachers in phase two did not use the learning resources to teach the students how to gather data using sensors. Their reasons were a mixture of having no time, no expertise and the sensors not being installed correctly. It was only in phase three when the micro:bits were introduced to the project that the students were able to gather and analyse data from the Internet and the micro:bit sensors.

The Educational Platform that the students and teachers used to view their data was a bespoke website that they were able to use and navigate. Viewing and analysing data in a webpage is simpler than using spreadsheet software like Excel. With websites there is nothing to install, sorting and filtering data is part of the webpage buttons and can be done with a few clicks.

Having a location to view all the data at once was beneficial to the project. Students could see all the data from all of the micro:bits and sensors together. They were able to analyse it and make conclusions from it that benefited their school.

4.2.6 Summary

The Energy in Schools project highlights the practical problems and potential solutions researchers have had trying to collate data from multiple devices. School systems do not easily allow devices to connect to the computers or the internet inside a school. Using the micro:bit instead of smart sensors allowed the project to continue but a bespoke bridging device and website was still needed to analyse the data the students collected with the micro:bits.

4.3 Methodology

In gathering requirements in section 4.1 a participatory design process was followed: involving stakeholders (the teachers) in the design of the new technologies. In the next chapter will look at designing new prototypes and again involve the teachers in developing and testing these. This is a participatory design approach: “the people destined to *use* the system play a critical role in *designing* it.” [182]. In order to continue involving the stakeholders in the next stage of evaluating the new technologies a new design process that extends participatory design was introduced: *participatory education*.

4.3.1 Extending participatory design: participatory education

In research when producing new technologies for teachers to use in the classroom, it is likely that the teachers will not be familiar with these technologies at the start of the process. There can be insufficient time for the teachers to learn how to use the tools, teach with them and integrate them into their curriculum and lessons. There is also a high chance that if the teachers were to try and deliver lessons with new technology, they would need a lot of external support from the researcher or risk the lesson not running to plan. Therefore, the process of participatory design to include *participatory planning* and *participatory delivery* was extended, as seen in Figure 43.

Figure 43 Extending participatory design into planning and delivery

The goal of this new design process is to enable researchers to deliver lessons to students with new technologies in participation with teachers. The researcher will take the lead in delivering the lessons, and the teacher will become a critical partner in planning and delivery. This creates a symbiotic relationship involving the researcher and the teacher. The researcher provides technical skills and context for the new technologies; the teacher provides contextual teaching and subject support. They both learn from each other and from the students as they experience the lessons. This new methodology is described in the following sections, highlighting its strengths compared with the state of the art.

4.3.2 Participatory planning

In the **participatory planning** stage, a consensus needs to be created between the researcher and the teacher. They are both experts in their areas and need to share this knowledge to plan successful lessons. A discussion is needed about the technology that is being introduced, the students and a plan of how the lessons will be delivered. The researcher needs background information about what the students are currently learning in their many different subjects. To engage students with real-life, meaningful and purposeful data and learning it needs to flow with their current learning, otherwise it risks being a onetime only set of lessons. (From 3.2.6 gap analysis it is known that one off lessons are common with the micro:bit and that research calls for longer lessons "that span a period of weeks" [185]). When the lessons are part of what the students are currently learning with their teacher, when the researcher leaves, this learning can continue. This limits the disruption research can cause in the classroom.

4.3.3 Participatory delivery

In the **participatory delivery** stage, the researcher delivers the lessons, but the teachers still play a critical role in this delivery. Teachers bring not only teaching skills to the classroom but curriculum knowledge, student familiarity and can provide invaluable feedback to the researcher(s) around the lessons' pace, content, delivery and the students' engagement. This methodology uses the teacher as an equal partner in the classroom when delivering lessons. It uses the researcher as someone to deliver the lessons but to also bring an outside experience to the students. The researcher's technical skills can be aspiring to students; it can give them a real-life experience of a person in a career in a specific field like STEM.

In the lesson the researcher leads in the delivery of the lesson, the teacher provides live feedback and support. The researcher provides technical knowledge to the students as well as the context such as what a data science life cycle is. Through teaching the students this knowledge is also passed onto the teacher who is observing. The teacher provides teaching experience, curriculum knowledge, student familiarity and feedback on:

1) The pace of the lesson – was it too slow, too fast, were all the students able to follow along.
2) The content of the lesson – was it appropriate for these students, could they understand the content.
3) The lesson delivery – could all the student access the materials delivered in the lesson.
4) Engagement – where the students engaged in the lesson, what engaged them, what didn't.
5) Context – the teacher knows what the students have learnt previously and can provide contextualised learning for the researcher, e.g. if the researcher is mentions South America, the teacher can add that the students recently studied Brazil, which is in South America.

The teachers can also provide subject knowledge to the students, for example knowledge about the eco systems the students are studying and collecting samples from. The students provide feedback to the researcher and the teacher. From this feedback and interactions with the students both the researcher and the teacher can determine student capabilities and their levels of understanding. These interactions between researcher, teacher and student are detailed in Figure 44 below. The three groups: researcher, teacher and students give and receive knowledge during the lessons. They can provide equal contributions.

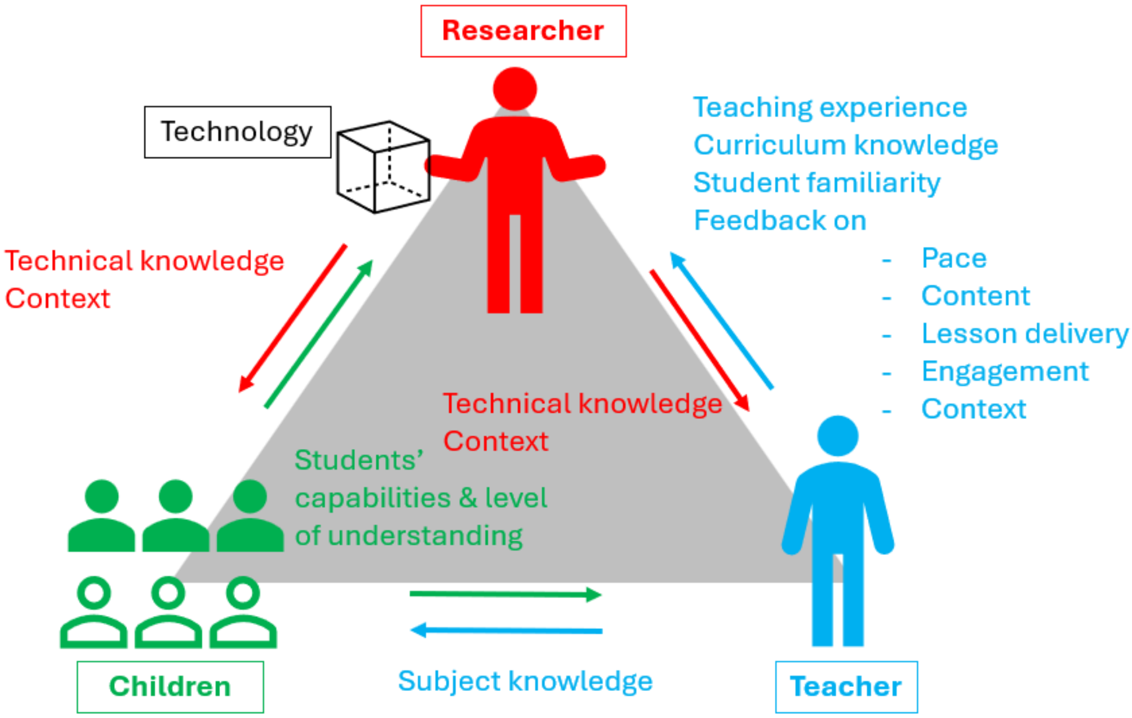


Figure 44 knowledge transfer in a participatory delivery lesson

The teachers' feedback is integrated into the lessons as they happen and to improve the next lesson. After the lesson the teacher and the researcher can meet again and do a more thorough feedback session and discuss the next lesson. This constant feedback is valuable for both parties. The teacher can clarify technical terms and context previously unknown to them. The researcher can gather evaluation from the teacher to improve their lessons and in the longer term, their technology and tools. A researcher with teaching experience does not personally know the students in the class like their teacher does. This relationship between teachers and children is vital for classroom cohesion.

4.4 Requirements summary

Chapters 1-3 discuss the importance of teaching data skills to students; they introduce the fields of physical computing and data science education. Chapter 1 briefly introduces physical computing and the need for data science education in schools. It describes the current methods and tools educators are using to teach Computer Science in schools. It includes information around current teaching techniques, teacher statistics and student engagement. Chapter 1 concludes that physical computing and data science can be mutually beneficial to each other and that primary schools are the ideal place to teach the data science pipeline as its thematic teaching supports cross-curricular education. In Chapter 1 it is asked can data science learning be made meaningful for primary school children in the context of physical computing. These questions are broken down into three solid questions about **pragmatic** teaching, **engaging** end-to-end experiences and **educationally sound** teaching.

Chapter 2 describes physical computing and data science. It summarises that physical computing tools in the classroom are generally mobile, lightweight and sturdy. It summarises ways a physical computing solution to data science can be pragmatic, engaging and educationally sound. **Pragmatic** solutions need to be **cost effective, accessible for educators** to use and install the hardware, software and setup any network connections, **accessible for students** to understand and use and to be age appropriate. To be **engaging** the tools need to reflect **real-life** scenarios, allowing students to collect **meaningful data** that has **purpose** for them in their environments. The tools need to prove to educators they are **educationally sound** by allowing them to teach students **collaborative learning, contextualisation of learning, computational thinking** and also support educators to **progress** students' skills over different age groups, **scaffold programming tasks** for these novice learners and teachers in **cross-curricular** lessons**.** Chapter 2 introduces the micro:bit as a possible physical computing tool for data collection.

Chapter 3 looks more closely at the micro:bit as a tool for data collection, how its hardware and software operate, the different ways it can be coded, how accessories can be added to it and how it can store data in its memory. Chapter 3 then looks at research around the micro:bit, how researchers are using the micro:bit in education. It identifies and summarises the gaps in this research.

In this chapter the first stage of participatory education is used: participatory design. Educators are interviewed on their current practices around data science education. Two solutions are examined – one semi-automatic data collection and the other a larger project called Energy in Schools. The creation of the new methodology participatory education is defined and its three stages: design, planning and delivery.

4.4.1 Concepts definitions

To clarify the requirements of a system to expand physical computing to teach data science each of the design concepts mentioned above are defined. This ultimately goes on to serve as the basis to evaluate the effectiveness of the research.

Pragmatic is dealing with things realistically that is based on practical considerations, rather than theoretically. Under pragmatic there is:

1) Cost effective: this refers to the tools themselves. Cost effective tools are good value for the amount of money paid. As mentioned in 2.3 Background analysis a cost-effective physical computing tool is one that can utilise an existing resource, is open source and can be easily shared between students and classes. Looking at the cost of physical computing tools mentioned in chapter 2 the range is from £12 to £200.
2) Accessibility: this means making something readily usable by everyone, regardless of their abilities. In terms of physical computing tools, they need to be accessible in their hardware, software and networking capabilities. They should be easy to install, use and maintain by novice users, regardless of their background or the technical configuration of their school.
3) User-friendliness: combining the best parts of the physical computing tools studied in Chapter 2, a new user-friendly tool should be: sturdy, transparent, mobile, battery-powered, lightweight, child-led and programmable.

Engaging

Engagement is defined as the students' attention to a particular task or activity. The extent of students' active involvement in the activity is referred to at the moment the activity is taking place. Kearsley states "the fundamental idea underlying engagement theory is that students must be meaningfully engaged in learning activities through interaction with others and worthwhile tasks." [112]. Kearsley's Engagement Theory proposes that students can be engaged in learning activities through interaction and worthwhile tasks and that technology can facilitate this engagement. They emphasise three key components: relating (group work), creating (project-based activities), and donating (authentic, outside-the-classroom focus). This project aims to engage students by covering these three components. It will allow group work, project-based activities and creates authentic, outside the classroom focus by allowing teachers to relate the learning to real-life, meaningful and purposeful projects.

As discussed in Chapter 2 Background, students can be engaged when collecting data if the collection process and the data itself is based in real-life, has meaning and purpose. These three aspects of what is measured are defined below:

1) Real-life: this refers to the physical (non-fictional) world perceived by the students as opposed to imaginary or virtual worlds. Can the students use the tools in a real-life concept? Can they be used to collect and analyse real-life data? Does collecting real-life data engage the students?
2) Meaningful: meaningful data is information that is relevant and useful within a specific context, in this case is the data (and the project as a whole) perceived as being meaningful to the students?
3) Purpose: the reason for which something is done, created or exists. Purpose is discussed in two ways: the purpose of the toolkit and the purpose of the data collected. Do the students and the teacher see purpose in both? Are they more engaged when they see a purpose?

Educationally sound refers to techniques of learning that are well established and beneficial for learning. These techniques of learning include:

1) Collaborative learning is where students work together to achieve a common goal [57]. Collaborative learning is beneficial to students as it promotes social interaction, improves communication skills, creates trust and encourages engagement.
2) Contextualisation of learning is presenting educational material in context to enhance understanding [20]. The context can be in another part of the curriculum the students are currently studying.
3) Computational thinking [219] is defined in section 1.3.5 as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent." It is often presented as a framework of concepts such as pattern recognition, decomposition and algorithms.

4) Scaffolding programming tasks [202] is when the educator presents a programming task that is not complete or has a problem. The students must then read the existing code in order to complete it.
5) Progression is the advancement of students' understanding, skills and knowledge over time [218].
6) Project based learning states that students learn through active exploration of real-world challenges and problems by creating a concrete artefact which represents pupils' new understandings and knowledge [122].
7) Cross-curricular is when different subjects are taught together. Instead of teaching subjects in isolation, educators link them together, allowing students to see how knowledge and skills from various disciplines can be applied to real-world situations and interconnected topics.

4.4.2 Aims and requirements

Aims gathered throughout this thesis include gaps found in the literature. The aim is to use physical computing to teach data science in primary schools (A1) with the micro:bit (A2) as the data collection tool. To support educators to teach students how to collect real life and local data (A3). For educators to be able to cover multiple subjects beyond Computing (A4) with the tools. To store this data for analysis and visualisation (A5) quickly and easily. To utilise a data science pipeline as the guide for their lessons (A6). For the tools to be usable by different age groups, learning different skills, supporting progression (A7). Finally, for educators to teach students about ethics and privacy when collecting and analysing data (A8).

The requirements for a set of tools to use physical computing to teach data science are summarised below.

Identifier	Pragmatic requirements (cost effective)	Derived from
R1	It must be cheap to install and maintain. Failure to minimise the one off and recurrent costs of ownership will reduce adoption.	Section 2.3.1 Section 3.2.5 Section 4.2.5
R2	It should utilise an existing resource and/or use open-source tools to prevent further costs.	Section 2.3.1 Section 3.2.4
R3	Tools need to be shareable between children and classrooms to cut costs.	Section 2.3.1 Section 2.3.3

Table 16 Pragmatic requirements (cost)

Identifier	Pragmatic requirements (accessible)	Derived from
R4	It must be accessible for novice educators to use, setup and maintain.	Section 2.3.1 Section 4.1.3 Section 4.2.5
R5	The tools should be simple and familiar.	Section 2.3.1
R6	The hardware and software need to be installed and used without needing administrative access to school devices.	Section 2.3.1 Section 4.2.5
R7	The tools should be usable with any type of school device, e.g. tablet or laptop.	Section 4.1.3
R8	The tools should be setup without the need for specialist apps on mobile devices.	Section 4.2.4
R9	The tools must be able to work through school firewalls.	Section 2.3.1 Section 4.2.4
R10	The tools should work regardless of WiFi availability and/or signal strength.	Section 2.3.1 Section 4.2.4
R11	Educators should be able to store the data collected digitally to access it again on a later date.	Section 4.1.3
R12	Educators should be able to use a semi-automated survey collection device.	Section 5.1.1
R13	Educators should be able to create their own custom data collection sheets.	Section 5.1.1
R14	Educators and students should be able to upload data from the data collection devices to the data storage quickly and easily.	Section 4.2.4
R15	Educators should be able to display data on their whiteboards to use as a teaching aid.	Section 5.1.2

Table 17 Pragmatic requirements (accessible)

Identifier	Pragmatic requirements (student accessibility)	Derived from
R16	The tools should be age appropriate.	Section 2.3.1 Section 3.2.6
R17	It should be transparent what the tools do/are doing.	Section 2.1.8
R18	Like other physical computing tools made for the classroom, the tools should be sturdy, mobile, lightweight, battery powered and programmable.	Section 2.2.2
R19	The tools shouldn't have small text, screens and buttons like a mobile phone.	Section 2.3.1
R20	The tools' screens should be visible outdoors in daylight.	Section 2.3.1
R21	The tools should work in wet and windy weather.	Section 4.1.3
R22	Students should be able to easily collect data using images and buttons next to each other.	Section 4.1.3
R23	Students should be able to sort and filter the data to analyse it.	Section 4.1.3

Table 18 Pragmatic requirements (student accessibility)

Identifier	Engagement requirements	Derived from
R24	Students should be able to code the data collection tools to help increase their programming ability belief.	Section 3.2.5
R25	It should support students to collect local data to support understanding global environmental issues and active participation locally and globally.	Section 2.3.2
R26	It needs to support students to collect live and relevant data to explore trends and answer questions they are interested in.	Section 2.3.2
R27	The data the students collect should be real data, that's relevant to them.	Section 2.3.2
R28	The data that students collect should have a purpose.	Section 2.3.2 Section 3.2.5
R29	It should be possible to link the data to other areas of the curriculum to create meaning.	Section 2.3.2 Section 3.2.5

Table 19 Engaging requirements

Identifier	Educationally sound	Derived from
R30	The tools should support contextualisation of learning.	Section 1.2.1 Section 2.3.3
R31	The tools should create opportunities for collaborative learning by making tools shareable.	Section 1.3.5 Section 2.3.3
R32	The tools should support computational thinking methods such as debugging, abstraction, logic, etc.	Section 1.3.2 Section 1.3.6 Section 2.3.3
R33	The tools should allow educators to scaffold programming tasks to support novices learning to code.	Section 1.3.6 Section 2.3.3 Section 3.2.5
R34	The tools should support delivery of lessons in the structure of a data science pipeline, to support progression.	Section 2.2.6 Section 2.3.3
R35	The tools should allow educators to plan for progression.	Section 2.3.3 Section 3.2.6
R36	The tools should support cross-curricular teaching to combine multiple subjects.	Section 1.3.6 Section 2.3.3 Section 3.2.6
R37	The tools should support project-based learning.	Section 1.3.6
R38	The tools should create opportunities for teaching about ethics including privacy and consent.	Section 2.2.6 Section 3.2.6

Table 20 Educationally sound requirements

Chapter 5

The Toolkit for Education in Data Science

The Toolkit for Educators in Data Science (TEDS) is a solution for teaching data science using physical computing. It aims to give primary aged students and their educators the tools to run real-life data science studies from data collection, cleaning and analysis through to presentation of information. It is a novel combination of hardware and software that empowers and engages children to experience a data science pipeline.

The toolkit aims to:

- tackle the **pragmatic** challenges of teaching physical computing and data science in schools such as making tools cost effective and accessible to both educators and students.
- support teachers in creating **engaging** lessons around physical computing and data science by giving them the opportunity to collect **real-life**, **meaningful** data that creates **purpose** for the students.
- support teachers to create **educationally sound** curriculum aligned lessons where students can work **collaboratively**, learn **computational thinking** and educators **can contextualise learning** and **scaffold programming tasks**.

The toolkit consists of a data collection device based on the micro:bit, called the clip:bit. It is a digital clipboard with 12 buttons, LEDs and simple number displays. It stores data on the micro:bit. The second piece of hardware in TEDS is the data storage device called the Cloudlet. It stores all of the data from multiple clip:bits and hosts a website for students to view, analyse and visualise their data, Figure 45.

Figure 45 The clip:bit and Cloudlet

5.1 Hardware design

In Chapter 4 requirements were gathered for how teachers could use physical computing to teach data science. Students would need a data collection device for collecting live and relevant data from their local environment. This data needs to be stored on a data storage device. This section describes the design of those two tools. This design was undertaken with educators in an iterative process using participatory design.

5.1.1 Data collection prototyping

To support educators and children in data collection, a prototype electronic clipboard was developed and built from cardboard, Figure 46. The device was made to help students collect data in the field.

Figure 46 A prototype clipboard in cardboard with a 7 segment display and buttons

Different numbers, size and types of buttons were added to the clipboard to assess how the buttons felt to the users. The buttons were added and removed from both sides of the clipboard during feedback sessions. A 7 segment LED was added to the top of the clipboard, again to gather user feedback on its location, its visibility due to its size and visibility in daylight, and its possible use. Other displays such as OLED screens and the LED screen on the micro:bit were also discussed with the users.

Multiple users including educators and design specialists were asked to hold the clip:bit and imagine its use. The informed feedback provided valuable insights that framed an iterative design process with Computer Science experts. A data collection tool, the clip:bit, was designed and produced.

Feedback

While RA2 thought teachers could use the Woodland trust sheets from the previous chapter, and seen in the cardboard prototype in Figure 40, RA6 stated that the teachers should make their own sheets. The sheets could be unique to the environment the students were in, and they could also align to the buttons, making it easier for the students to quickly record what they had identified.

Another piece of feedback on the clip:bit prototype from RA6 was around selecting the item and changing its count value. The initial idea was students would use the buttons next to the items to select it, e.g. blackberries. Then at the top of the board there would be a scrolling button for students to increase the number quickly. A designer from Lancaster Institute of Creative Arts (LICA), D2 suggested a thumb pad for an easier way of scrolling numbers. With their experience of working with children RA6 suggested that speed was not what the hardware needed. Students needed to press the button next to the item and press it again if they saw a second item. This pressing for each item was like a confirmation for the student. Each press would mean they are recording that item. RA6 felt that scrolling up and down the numbers would be too playful and distracting for the students, especially in scenarios where they were only typically seeing a handful of the items like birds. They also stated students wearing gloves in the winter time would find a scrolling button or a thumb button too difficult to use. And younger children would struggle to stop both on the right number, spending too much time recording numbers. Touch buttons and touch screens are also difficult to use with both gloves, outdoors in wet and damp weather and by younger children with limited dexterity.

5.1.2 Data storage, analysis and visualisation

Through interviews with educators surrounding data collection, it was discovered that many of them didn't store the data students collect. As part of TEDS it was decided to develop a data storage device that educators could easily use without administrative access to software, hardware or Wi-Fi. This should be somewhere the data could be stored and accessed by children, where they could easily filter and sort it for analysis and visualisation. This device is called the Classroom Cloudlet, Cloudlet for short.

The Cloudlet itself needed to store the data uploaded to it. It needed some way of multiple students viewing the data from their devices like their school computers, laptops or tablets. Ideally, the data from the Cloudlet could also be shown by teachers on a bigger screen to the class as a teaching aid. As discussed previously, the data would be shown through a web interface. With all these requirements it was decided to use a Raspberry Pi as the cloud storage for visualisation and analysis. The Raspberry Pi can act as a Wi-Fi hotspot for other devices to connect to it. It can host a web interface and a micro:bit can be plugged directly into it through its USB port, either as a bridge or directly from a clip:bit.

A strategic decision was made **not** to put the data online. There were multiple reasons covering this both in terms of privacy and technical accessibility. It was also decided not to create a custom app that students and teachers had to download and install to access the data. The data storage needed to be accessible offline, in-person without administrative access.

Another iterative process was done between a team of computer scientists (the author, RA2, C1 and C2) with both technical and educational experience. The flow of data was looked at, Figure 47(a), between the clip:bits, the Cloudlet, the micro:bits and the students/educators' devices – laptops, tablets, etc. A discussion was had on how data could move between the devices and what was the ideal path. Data needed to move from the multiple micro:bits attached to the clip:bits to the cloud. Technically this can be done in two ways: over the micro:bit radio using another micro:bit as a bridge (just like the Energy in Schools project) or physically plugging the micro:bits directly into the USB port of the cloud using a USB cable. Both options are explored later in the chapter.

Using the outcomes from other projects such as Energy in Schools a discussion was had on the different topics the students could learn on top of data science, e.g. coding, electronics, etc, Figure 47(b).

Finally, a prototype "data centre" was sketched Figure 47(c). The different possible data connections, 5G, WiFi, Bluetooth, etc. that it could have and the physical components of it, e.g. an external flash drive, LEDs, battery were sketched out and discussed.

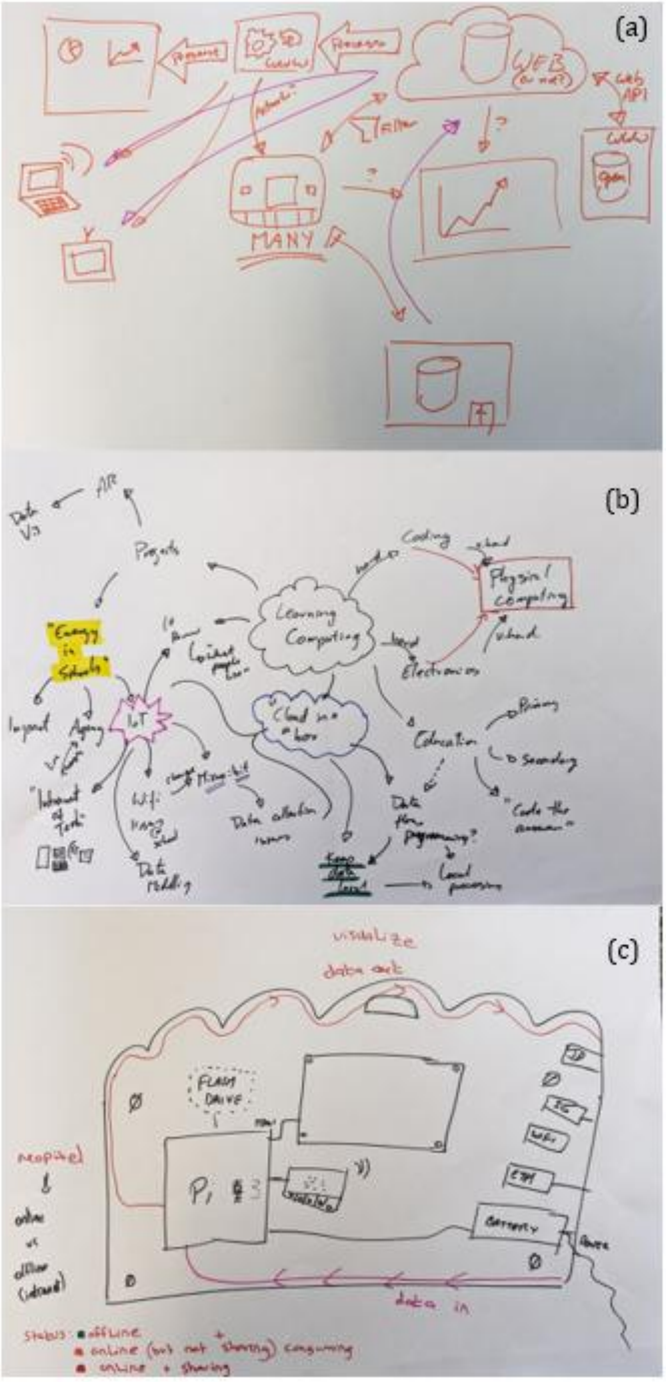


Figure 47 The design phase of the clip:bit and the classroom Cloudlet

5.2 Clip:bit

TEDS supports giving students the full experience of data science by following a data science pipeline. It adapts the data science pipelines described in Section 2.3 Data Science Education. Each of the devices fits into the data science pipeline, Figure 48.

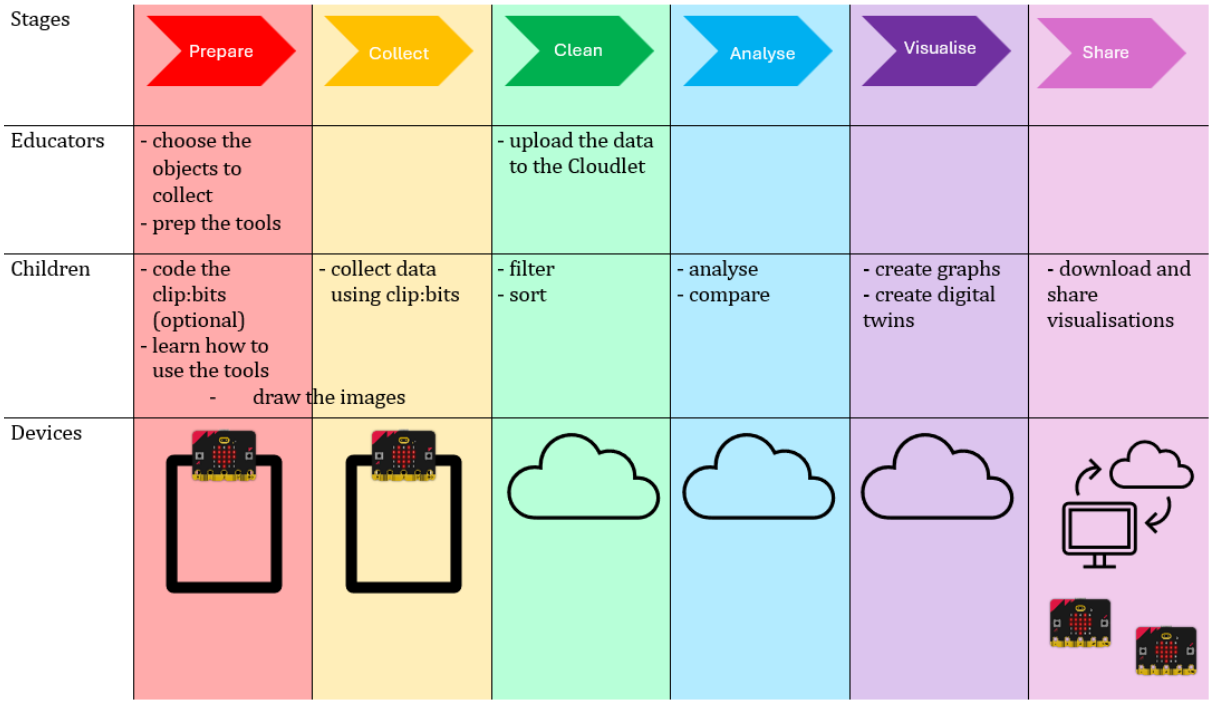


Figure 48 TEDS data science pipeline including devices at each stage

An additional stage included in the TEDS pipeline is "prepare". The tools the students use for data collection must be prepared first. In professional data science lifecycles, the tools that collect the data are removed from the process, they are assumed to be ready to collect. As this toolkit is educational, preparation of these tools in the lifecycle is needed to make the process transparent to educators and children.

The clip:bit is the semi-automated data collection tool, see Figure 49. It is lightweight for young children to carry. It is battery-powered with replaceable batteries and it is mobile and it can be used outdoors. The clip:bit is programmable, it can be setup by the teacher, the student or both by programming the micro:bit or downloading a program to the micro:bit. There is no other software setup needed for the clip:bit. The micro:bit is programmed through MakeCode, the **easy-to-use** block-based website. There are no updates, apps or programs to install on either the clip:bit or the computer programming the clip:bit. The data the students are collecting using the clip:bit can be defined by the teacher with printed images. Or the study can be **child-led**; students can classify the data by drawing sketches whilst outside collecting the data. Outside the students use the images and the clip:bits to identify and count items in their local environment. The counted data is stored in the micro:bit's persistent memory, the data log.

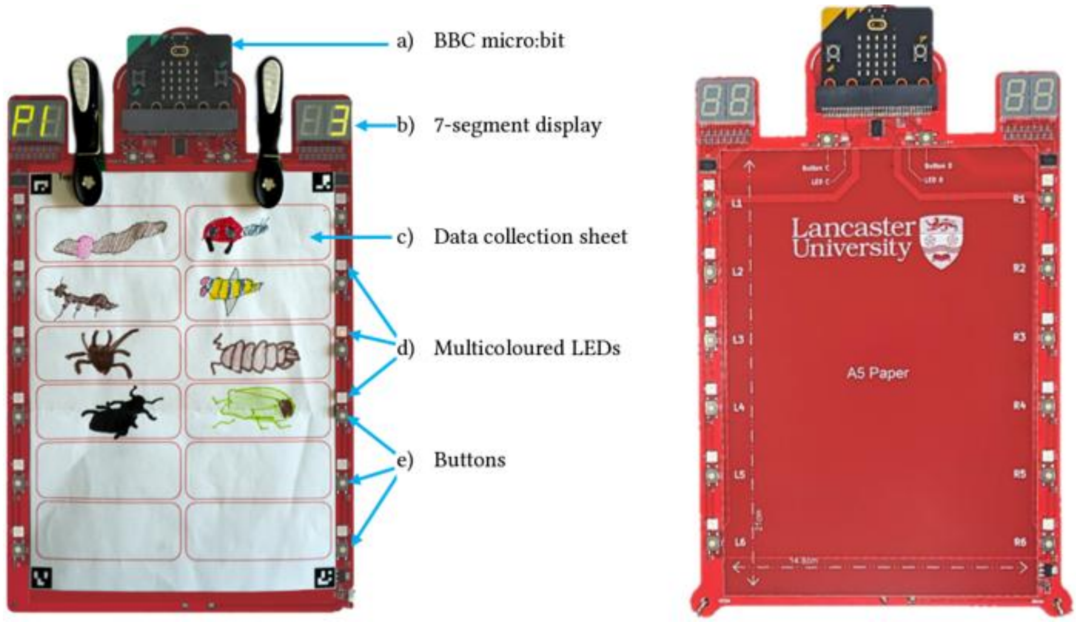


Figure 49 Key components of the clip:bit

The clip:bit, Figure 49, has 14 buttons (e), 6 on each side of the board and two below the micro:bit. For **transparency** the board's components are labelled. The left-hand side buttons are labelled L1, L2, L3, L4, L5 and L6 and the right-hand side buttons are labelled R1, R2, R3, R4, R5 and R6. The final two buttons are labelled C and D, following on from the micro:bit's embedded buttons A and B. The clip:bit has buttons on both the left- and right-hand side of the board so there is no preferred way of holding it, i.e. it is not exclusively designed for right-handed students. In the centre of the clip:bit is space for an A5 (14.8cm by 21cm) piece of paper. This piece of paper will show images of what the students are collecting, it is called the data collection sheet (c). It is designed to line up next to each button. Below each button there is a multicolour LED (d). On the top of the clip:bit are two 7-segment LEDs displays (b).

The clip:bit is controlled by a BBC micro:bit (a) that is plugged in at the top. All aspects of the clip:bit can be coded using a custom extension that is easily loaded into the block-based programming language MakeCode.

5.2.1 Preparation stage

In the preparation stage the educators prepare the tools and the students. Preparation is a vital stage in this lifecycle. It is not just about the practical preparation needed to use technology: assembling devices, putting batteries into them, etc. but understanding how the technology works and becoming familiar with it. It is the first introduction students will have on the toolkit. Preparation of the clip:bit can be done by the educators or the educators and the children together.

5.2.2 What data to collect

Figure 50 A data collection sheet with pre-printed images, in this case different birds to look for.

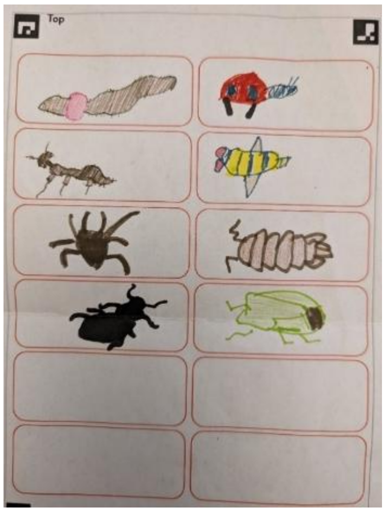


Figure 51: A child's hand-drawn data collection sheet that depicts different insects.

One decision that needs to be made in the preparation stage is what data the children will be collecting. The toolkit supports two options:

1. The teacher provides pre-printed images, as in Figure 50. The detailed images can be useful for identifying the different objects in the wild. These are custom versions of the Wildlife Trust sheets from the previous chapter.
2. Alternatively, the children can lead data classification by drawing images of objects, as in Figure 51. This gives children creative control, and it can help

develop their identification skills as they draw key features of what they are looking for.

Alternatively, the two methods can be combined. A teacher can provide a sheet of objects they are looking for and leave blank boxes for the children to add new objects by sketching them. This new data can be labelled later, back in the classroom, in the “Clean” stage of the pipeline.

The data collection sheet aims to empower the educators and the students. It gives the teachers control over what data they want the students to collect. They can specify exactly what it looks like and add text labels to the sheets for further learning.

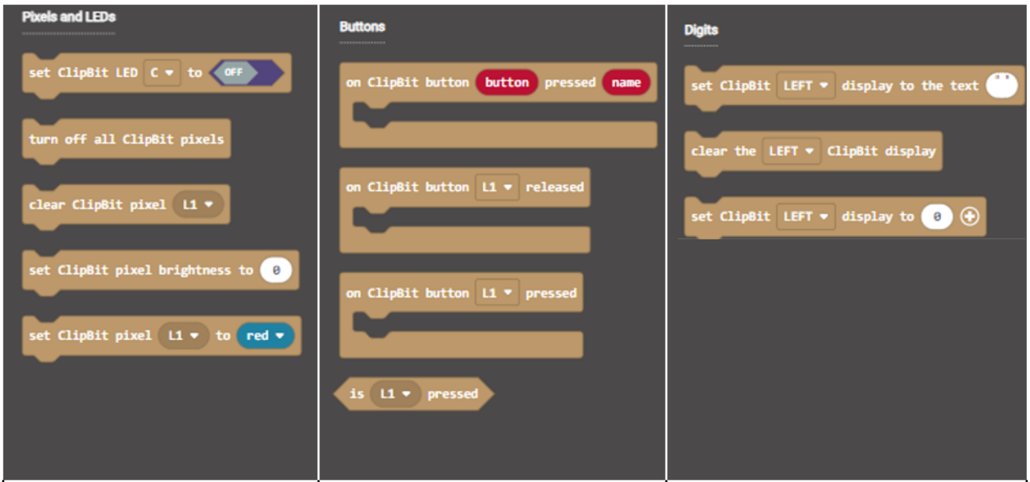
The data collection sheets, either pre-printed, hand-drawn or a combination of both, have four unique QR codes in the corner of each sheet, Figure 50(a). The teacher can take a photograph of the sheets and upload them to the Classroom Cloudlet. The QR codes are used to pinpoint the individual images next to each button. Each image is matched against its button. For example, the top lefthand corner coordinate of each QR code is top left: (0, 0), top right: (1480, 0), bottom left: (0, 2100) and bottom right: (1480,2100). And the image next to the button L1 in the top lefthand corner has the coordinates top left: (183, 51), top right: (706, 51), bottom left: (183,432) and bottom right: (706, 432). Once the software has spotted all four QR codes it skews the image into the rectangular shape it expects it to be. It then takes the known coordinates of the 12 boxes and creates 12 individual images. It makes the white background of these images transparent before it saves them to the cloud. It saves each image against the button it was next to. The QR codes and image recognition software make it quick and easy for images to be uploaded all at once.

5.2.3 Coding the clip:bit

Several custom extensions are available in MakeCode for the clip:bit. These blocks of code allow children and educators to code the clip:bit. The users can:

- Change the colours and brightness of the different LEDs around the board, Figure 52(a).

- Code behaviour for every button and individual buttons on the board including button press and button release events, Figure 52(b).
- Add text and numbers to the 7-segment displays, Figure 52(c).



a) clipbit MakeCode blocks Pixels and LEDs	b) Buttons blocks	c) Digits blocks

Figure 52 MakeCode blocks for the clip:bit

This code can be made available to the educators, they don't need to create it themselves. The same applies to the students. A full program to use the clip:bit as a data collection device, Figure 53, can be made available to the educators and the students as a downloadable hex file. They can download this file directly to the micro:bits attached to the clip:bits. Figure 53 shows a full program for a clip:bit based on the data collection sheet in Figure 51.

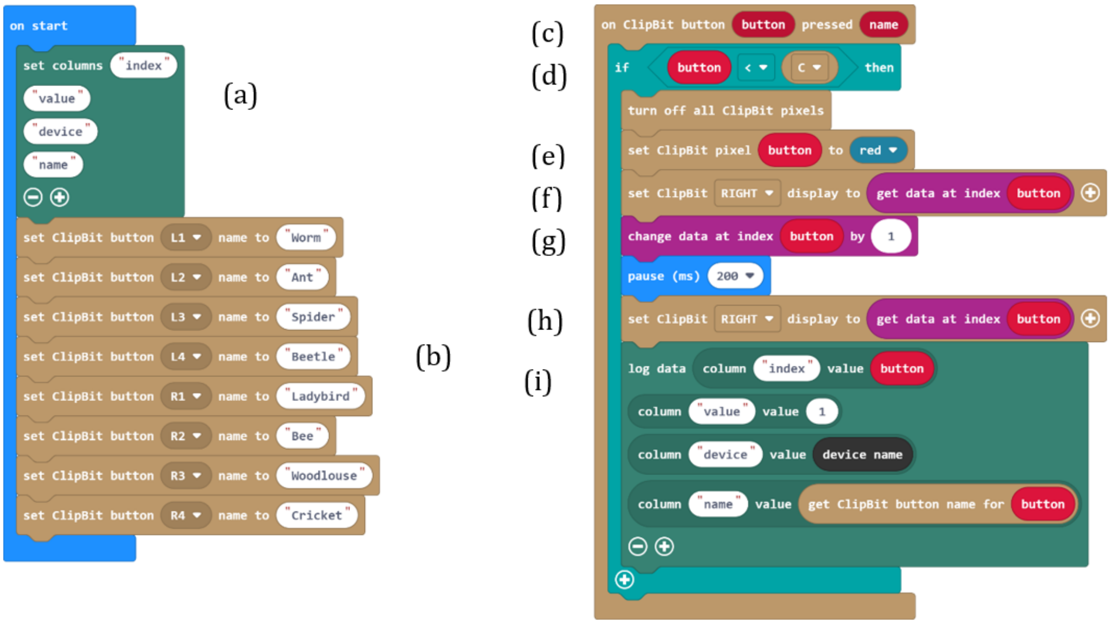


Figure 53 Coding the clip:bit in MakeCode

The program works as follows: (a) when the micro:bit starts, the data log is setup with columns index, value, device, name. Index is the name of the button and value is the count that button has, name is the name of the item being recorded. Device is the unique name of the micro:bit. Next (b) each button is given a name. In this example buttons L1, L2, L3, L4, R1, R2, R3 and R4 are labelled. This can be done very easily by a student.

The rest of the program deals with a button press (c). The program checks the button is not the buttons labelled C or D (d). The clip:bit turns off all the multicoloured LEDS and then turns on the LED of the button that was pressed (e). It gets the value of the button that was pressed and displays it in the right hand 7 segment display (f). If this is the first time this button was pressed, this value will be 0. The clip:bit then changes the value of this button by 1 (g). In some cases, the students may want to change this if they know they're going to count a lot of an item, e.g. they could increase the value by 5 on every button press. Once that value has changed, it displays it again in the right-hand display (h). This way the students sees the increment take place. They see the value of the button change. Finally, the clip:bit saves this change to the micro:bit's data log (i). The total value of the button is not saved, just the number it was changed by. For example, if the value of the button was 7 and is now 8, the number 1 is saved. In the micro:bit log there will be a total of 8 rows for this button. The counting and grouping of these records is done by the Cloudlet later.

5.2.4 Data collection

Once the data collection sheets have been created and the clip:bits have been programmed the students can start collecting data.

It is expected that a clip:bit will be shared between two or three students but can be used individually as well. The students can wear the clip:bits using lanyards attached to holes in the bottom of the board. The clip:bit hangs down with the data collection sheet facing outwards. It can be lifted up to be used, the lanyard doesn't need to be twisted.

Once the students have identified an item they want to count they find it on their data collection sheet looking at the images they have drawn or the ones that have been provided by their teacher. The students press the button next to the image. The LED below the button will light up to indicate which one they pressed. The right hand 7 segment display will display the old value followed by the new value.

Students can continue to look for and count objects outdoors in the environment they are studying. They can walk around as they look for the objects. They may use other devices such as binoculars or magnifying glasses to identify the objects. The clip:bits can be passed between students as they share them.

5.3 Classroom Cloudlet

In the toolkit the data can be uploaded to a custom-made physical representation of a data centre: the Classroom Cloudlet. The Cloudlet looks and acts like a data centre. It collects and stores data for analysis and visualisation. All the data from all the clip:bits can be uploaded and stored here. The images used to collect the data can also be stored here, alongside their corresponding data. The students can connect their devices (tablets, laptops, etc) to the Cloudlet and view their own data as well as their classmates. They can filter and sort the data as well as visualise it using graphs and custom screens.

The Cloudlet is designed to be easily managed by the teacher. It is a standalone device. It does not need installing or setting up. There are no apps or software to install to use the Cloudlet. It is also offline. The Cloudlet creates its own Wifi for students to connect to. It doesn't need to be added to the school network, intranet or Wifi systems. Again, no extra software such as an app or a program needs to be installed on the devices connecting to the Cloudlet to view the data.

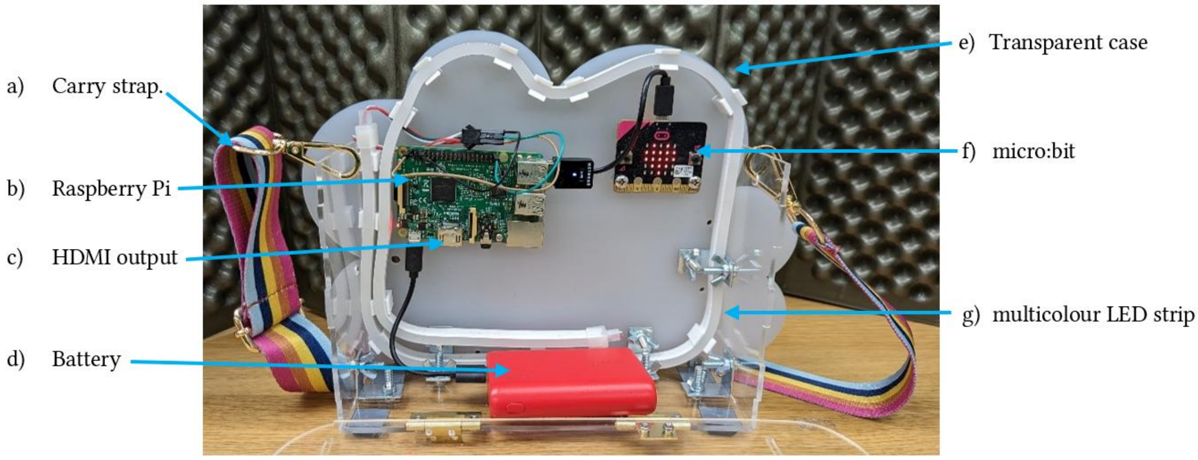


Figure 54 The tangible mobile Cloudlet with multi-coloured LEDs and battery

The Classroom Cloudlet, Figure 54, contains a Raspberry Pi single-board computer (b) connected to a micro:bit (f) and surrounded by a strip of multicolour LEDs (g). The Raspberry Pi has a HDMI video output (c) so an educator can plug it directly into a screen in the classroom to show the data to the class. The Cloudlet can be powered through a mains power supply or with a portable battery (d). The Cloudlet case is transparent (e) so the children can see all these components. The front of the case is hinged so it can be easily opened, and the entire cloud is portable via a strap (a) connected to its sides. While not yet waterproof, the Cloudlet has rubber feet so it can sit stably on the ground or a bench outdoors.

The multicolour LEDS are used to show the status of the Cloudlet. When students view data on the Cloudlet, the LEDS turn green. When a teacher accesses an admin screen, they turn red. When data is being uploaded to the Cloudlet, the lights animate to demonstrate the movement of data from outside the Cloudlet to inside.

The Cloudlet runs a Django webserver on the Raspberry Pi. Django is a Python web framework with a built-in database [199]. To connect to the Cloudlet students can change their devices' WiFi to connect to the Cloudlet's Wifi. Once connected, they then open a web browser on their device and navigate to the Cloudlet's home page. The web address is kept simple http://cloud.local. These webpages are described later in this section.

5.3.1 Data cleaning

Once the students have collected their data using the clip:bits, they can clean it before it enters the Cloudlet. There are multiple ways to upload data to the Cloudlet:

1. Plug the micro:bit from the clip:bit into a computer and download the data log as a csv file. Upload that csv file to the Cloudlet.
2. Plug the micro:bit from the clip:bit directly into the Cloudlet. Upload data from the micro:bit into the Cloudlet.
3. Send the data from the clip:bit micro:bit to the micro:bit connected to the Cloudlet acting as a bridge over radio.

Option 1 allows the educators and the students to edit the data from the micro:bit log, Figure 55 left, before it goes into the Cloudlet. By simply pressing "Download" this saves the log as a csv file to their computer. The educators and the students can then edit this data in spreadsheet software like Excel, Figure 55 right. This is not an easy process for either to complete.

It is deliberately difficult to edit the data from the clip:bit before it goes into the Cloudlet **and** when it's in the Cloudlet. This is to ensure educators and students think carefully about editing data, to consider the ethics behind it.

micro:bit data log

Download | Copy | Update data... | Clear log... | Visual preview

This is the data on your micro:bit. To analyse it and create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. Learn more about micro:bit data logging.

Time (seconds)	section	page	index	value	device	name
10.82	1	1	11	1	pegov	Ladybird
12.28	1	1	11	1	pegov	Ladybird
13.53	1	1	9	1	pegov	Woodlouse
13.95	1	1	9	1	pegov	Woodlouse
15.26	1	1	10	1	pegov	Bee
16.35	1	1	10	1	pegov	Bee
16.87	1	1	10	1	pegov	Bee
17.81	1	1	9	1	pegov	Woodlouse
18.30	1	1	9	1	pegov	Woodlouse
18.67	1	1	9	1	pegov	Woodlouse
19.09	1	1	9	1	pegov	Woodlouse
19.66	1	1	9	1	pegov	Woodlouse
20.19	1	1	9	1	pegov	Woodlouse
20.71	1	1	9	1	pegov	Woodlouse
21.91	1	1	9	1	pegov	Woodlouse

	A	B	C	D	E	F	G
1	Time (seconds)	section	page	index	value	device	name
2	10.82	1	1	11	1	pegov	Ladybird
3	12.28	1	1	11	1	pegov	Ladybird
4	13.53	1	1	9	1	pegov	Woodlouse
5	13.95	1	1	9	1	pegov	Woodlouse
6	15.26	1	1	10	1	pegov	Bee
7	16.35	1	1	10	1	pegov	Bee
8	16.87	1	1	10	1	pegov	Bee
9	17.81	1	1	9	1	pegov	Woodlouse
10	18.30	1	1	9	1	pegov	Woodlouse
11	18.67	1	1	9	1	pegov	Woodlouse
12	19.09	1	1	9	1	pegov	Woodlouse
13	19.66	1	1	9	1	pegov	Woodlouse
14	20.19	1	1	9	1	pegov	Woodlouse
15	20.71	1	1	9	1	pegov	Woodlouse
16	21.91	1	1	9	1	pegov	Woodlouse

Figure 55 (left) micro:bit data log and (right) The downloaded csv file

5.3.2 Data analysis

Once all the data from all the clip:bits has been uploaded to the Cloudlet the students can start to analyse and visualise it. Figure 56 shows three sets of data collected including the data from Figure 55 which is under the team name "Phil and Lorraine". The two other teams are "Jan and Angie" and "Kobi and Hannah".

NAME (All)	COUNT	TEAM (All)	IMAGE
Ant	2	Jan and Angie	
Bee	2	Jan and Angie	
Ladybird	1	Jan and Angie	
Worm	3	Jan and Angie	
Bee	4	Kobi and Hannah	
Worm	3	Kobi and Hannah	
Bee	3	Phil and Lorraine	
Ladybird	2	Phil and Lorraine	
Woodlouse	10	Phil and Lorraine	

Figure 56 Students' view of data in the Cloudlet

Looking closer at the data every team found some bees, which is why there are three different images of bees. Each image belongs to the team that collected it.

The column headings are clickable; once clicked they sort by that heading. The name and team headings also have dropdowns to allow users to filter the data. This sorting and filtering of data is a common way of analysing data on the web. Many websites like food shops, airlines, etc. use similar techniques. Using a webpage to display data can make it easier for children to sort and filter as well as view the numbers and images together.

5.3.3 Data visualisations

Graphs

The first visualisation the students can create using their data is a bar graph, Figure 57. Once the students have selected a team they can select the "Graph" button. The graph appears on the same screen as the data.

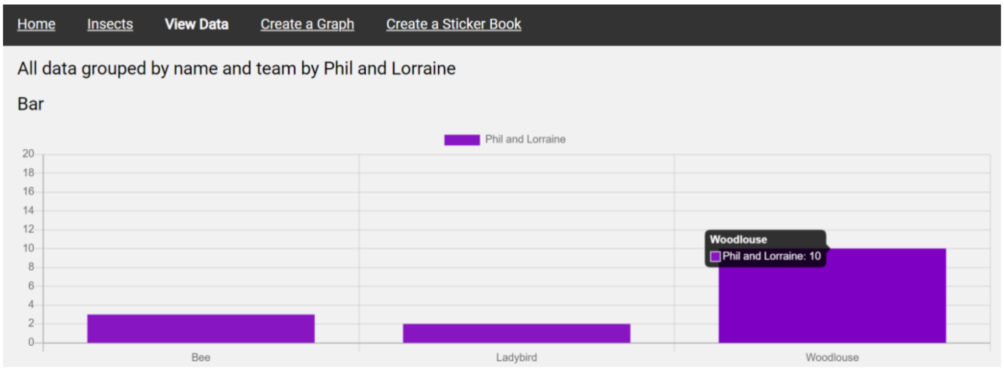


Figure 57 A graph for a team

When scrolled over, the bars display their label and value. The Cloudlet uses chartJS [36] to create these interactive graphs on a webpage. The graphs include vertical and horizontal axes that are labelled. They have minor and major gridlines as well as a key. These are all features of graphs that students learn in the Mathematics curriculum, that can be reinforced here.

Stickerbook

Another visualisation created for students is called a sticker book. This allows students to recreate their data visually as a digital twin. Their images appear as thumbnails down the sides of the window as well the count for each image. The students can add their images to the stage, for example if they found 3 bees, they can add 3 bee images to the stage. Once added to the stage, the children can move it around, but they cannot add more than 3 items.

Looking at the sticker book more closely, Figure 58, the children can filter by their own team or view all teams (b). They can select an item, e.g. Bee from the side menus (c) and then select the image. This adds the bee to the screen. They can use the arrow (d) to move the bee around the scene with their mouse. The students can change the background to use on their Stickerbook (a). Educators can upload their own background image. In the example below is the grass field of the student's local park.

The sticker book allows all of the students' data and images to be added to the sticker book. In the sample data there were nine bees in total but three different images for the three different teams. The first bee belongs to "Phil and Lorraine", they found three bees. After the first three bees, the bee image changes, and the user can see the bee of the next team. This allows every students' image to be added to the visualisation and downloaded (e). Figure 59 is the downloaded Stickerbook of all the images from the data above.

Figure 58 The Stickerbook

Figure 59 A Stickerbook visualisation in the Cloudlet

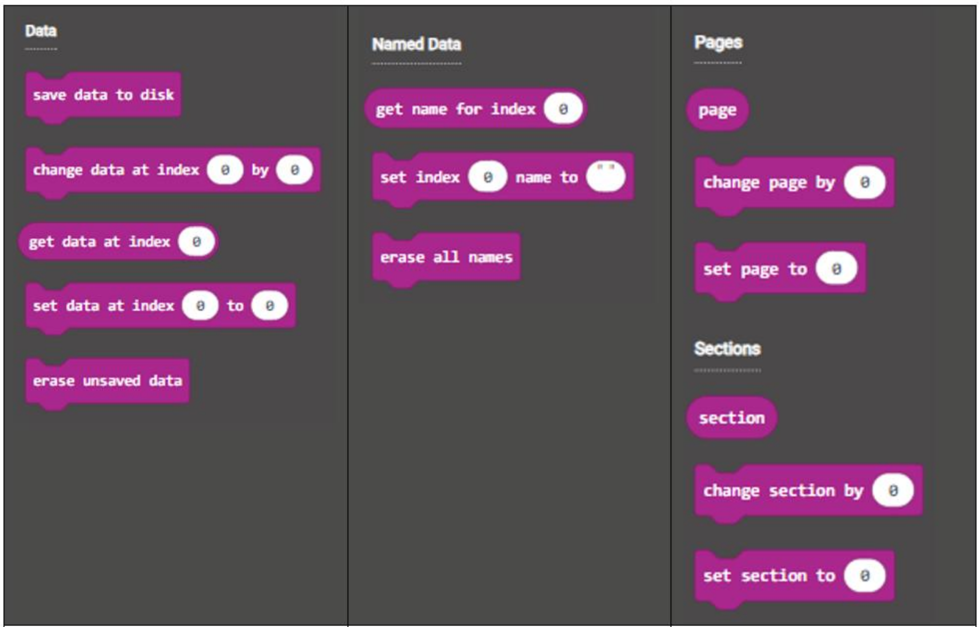
5.4 Advanced features

In the toolkit there are advanced features that were created for unique scenarios. These features are described below.

5.4.1 Clip:bit pages and sections

A separate set of blocks manage multiple pages and sections. From our original interviews with experts who collect data they requested the ability to record multiple pages of data, beyond the 12 objects that can be recorded on one data collection sheet. They also detailed how they recorded data in geographical sections and how important it was to distinguish the sections from each other. The clip:bit blocks allow the user to record a list with the following fields: Page, Section, Name, Index. The blocks allow the user to

- Save, get and erase data, Figure 60(a).
- Get, set and erase the label names of the data, Figure 60(b).
- Set and change the current page and the current section, Figure 60(c).



a) Clip:bit MakeCode data blocks	b) Named Data blocks	c) Pages and section blocks

Figure 60 Clip:bit MakeCode advanced blocks.

This allows for different objects to be counted in different sections. Here's an example of two rows of data stored in the working memory of the micro:bit:

Page	Section	Index	Name	Count
1	1	10	Bee	4
1	2	10	Bee	10

Table 21 Recording different objects in different sections on the clip:bit.

Different objects from different pages:

Page	Section	Index	Name	Count
1	1	10	Bee	4
2	1	10	Wasps	5

Table 22 Recording different objects on different pages on the clip:bit.

There is also code to change the page and section numbers. As well as the clip:bit buttons, students and educators can utilise the buttons on the micro:bit itself, for example in Figure 61 the A and B buttons on the micro:bit change the page number.

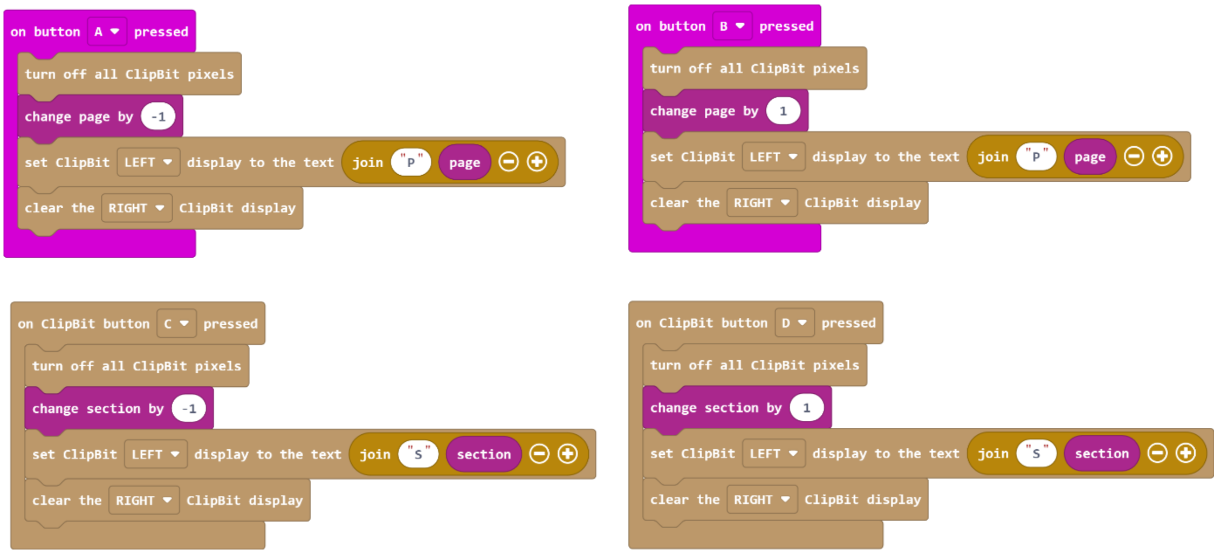


Figure 61 MakeCode blocks to change the page and section numbers.

5.4.2 Preparing the Cloudlet

Like the clip:bit, the Cloudlet does need some preparation. This preparation is ideally done by the educator; therefore it is not part of the students' pipeline.

There are "admin" pages meant for the educators to setup new projects, groups and sessions, Figure 62. A project might be the whole school project to count the number of bees around the school grounds. A project can have multiple groups. A group could be a year group, for example Year 1 students are 5 to 6 years old, and Year 6 students are 10 to 11 years old. Sessions are the moments the groups go out and collect data, educators may name these by the specific class, e.g. "5C front playground" to distinguish them from other classes in the year group or other locations.

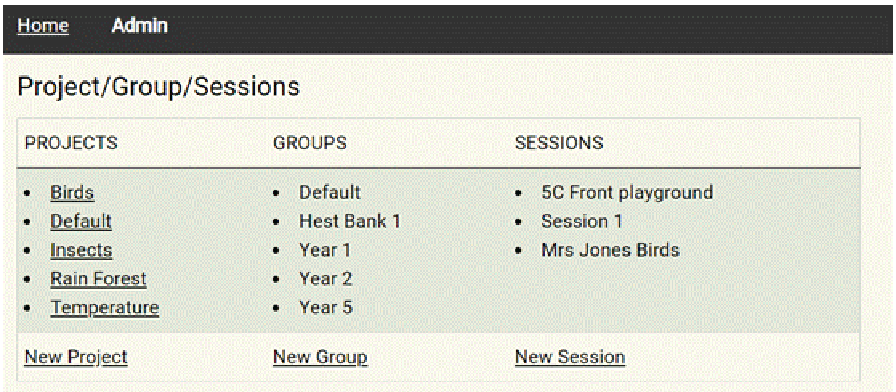


Figure 62 Cloudlet admin page to setup new projects, groups and sessions.

Creating projects, groups and sessions allow the children to view the data at these different levels. They can view all of the data in the "Insects" project for all of the groups, or they can view all of the sessions their group "Year 5" did collecting data. They have the control to view, analyse and visualise the data as they wish.

As well as creating new projects, groups and sessions the educators can upload the data collection sheets to a session if they have been prepared in advance, create student teams and pair the clip:bits to the teams, Figure 63.

PROJECT	GROUP	NAME	DATA		TEAMS	DEVICES	PHOTOS
Insects	Year 5	**5C Front playground**	View Data	Upload Data	0 Add Teams	Pair devices	Add Photos
Rain Forest	Year 5	**Session 1**	View Data	Upload Data	13 Add Teams	Pair devices	Add Photos
Birds	Default	**Mrs Jones Birds**	View Data	Upload Data	12 Add Teams	Pair devices	Add Photos

Figure 63 Cloudlet admin page to upload data, photos and create teams and pair devices.

By creating teams and pairing them with clip:bits this means when the students' data arrives in the Cloudlet it can be viewed and filtered using the children's names, not the micro:bit name. This makes the data more recognisable and personalised for the children.

5.4.3 Storing data

The Cloudlet stores the data in a database. The database entity relationship diagram is described in Figure 64 below.

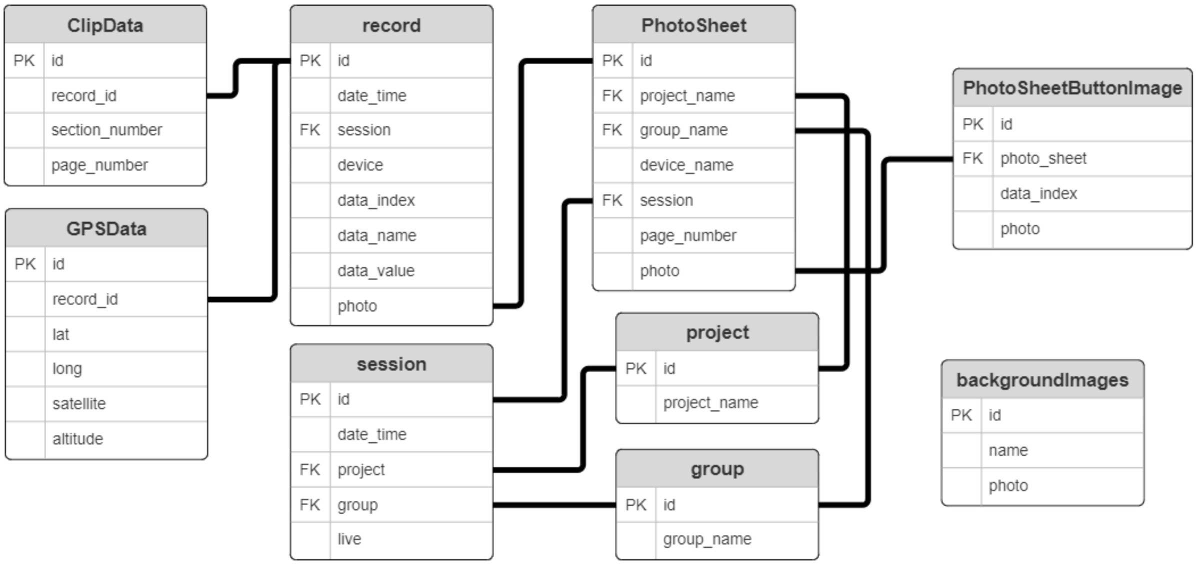


Figure 64 Cloudlet's entity relationship diagram.

Each time a student presses a button on the clip:bit a row of data is recorded, as seen previously in Figure 55: section, page, index, value, device and name. This data is stored in different tables in the database. In the future other sources of data could be uploaded to the Cloudlet, therefore data unique to the clip:bit like section and page number have been separated out from the central record table. Section and page are recorded into ClipData as section_number and page_number. Index, value, device, name are recorded into the main table, record, as data_index, data_value, device, data_name. Every row is given a date and time value in the field *date_time*. This is the date/time it was uploaded to the Cloudlet, not necessarily the date or time the data was recorded. The micro:bit does not have a real time clock therefore it cannot be determined when data was recorded.

Educators also stated they wanted to record information about each session, for example if it was raining when Class 1 went outside it's less likely they will have seen any bees. This information would be added to the session, not the group or the project.

5.5 Conclusion

This chapter introduced TEDS, the Toolkit for Educators of Data Science. TEDS aims to tackle the challenges of teaching data science in primary school using physical computing, specifically the micro:bit. A description of prototypes that were built and tested and different technological solutions is given. TEDS is detailed using a data science pipeline as the structure for its description. How the different tools of TEDS, the clip:bit and the Cloudlet, their hardware and their software, work together is described. The final section describes the advanced features of TEDS such as the database design and data storage before an analysis.

The next chapter is around the evaluation of TEDS. One hundred students were taught using TEDS in a pilot and a main study in two schools with three classes. The requirements from Chapter 4 will be evaluated in Chapter 6.

Chapter 6

Evaluation

This chapter puts the new methodology introduced previously into context and evaluates it. A practical study of TEDS was conducted in two phases. The study took the form of a set of lessons. At first a pilot study tested the methodology, the protocol, and the technology. Next, the aspects tested on the pilot were refined. The pilot study ran through the data science pipeline in two lessons while the main study had a lesson per each stage of the pipeline. The evaluation looks at each stage of the data science pipeline and evaluates it based on the requirements gathered throughout this thesis.

The evaluation finds that educators and students were able to use TEDS to learn about data. The tools were engaging for the students and prompted conversations around Computing as a career. Students enjoyed using the clip:bit and the Cloudlet to collect and analyse their data. Students and educators state that seeing the students' own images and data on a webpage in the Cloudlet gave their data meaning and purpose. The students successfully answered questions around the data they collected in two questionnaires. Educators were able to fulfil curriculum requirements from multiple subjects when using TEDS.

Year 3 teacher: "*Looking at data: I think because it was their data, their information, it's their names that are up there. You know, they're showing each other what they've found out and answering questions about what each other has done. I think it worked really well.*"

6.1 Methodology evaluated

6.1.1 Methodology application

For the evaluation two studies took place – a main study and a pilot study. The two studies focus on the real-world deployment and evaluation of a set of indoor and outdoor lessons using TEDS. The pilot study conducted in July 2023 was with two classes of Year 2 students (ages 6 to 7) in School 1, totalling 48 students. The main study was conducted in the Autumn term, September to December 2023, with two classes of children, one in Year 3 (ages 7 to 8) and the other Year 5 (ages 9 to 10) totalling 52 students. A summary of the students and their ages is in Table 23 below.

	Year 2 (ages 6 – 7)	Year 3 (ages 7 – 8)	Year 5 (ages 9 – 10)
School 1 (pilot)	48		
School 2 (main)		28	24

Table 23 Ages and year groups of children in evaluation schools

The studies took place during the students' normal lessons in school hours. Participation in the study was voluntary. Informed consent was obtained from both the children and their parents or guardians as well as the teachers involved. Ethics approval was gained by the faculty ethics committee [171].

The pilot study was the first time the clip:bit and the Cloudlet were used in an educational setting both indoors and outdoors. The aim of the study was to understand how and if the students could learn how to use the clip:bit in the classroom and outdoors, and could the teachers run their own lessons. Could they be taught what it was, what it could do and how to use it? Could they use it to collect data outdoors? Could they physically handle the clip:bit while collecting data? The study also helped to learn how the Cloudlet could be used in the classroom. Could educators use the Cloudlet to view the data? Could students connect to the Cloudlet to view their data? Could they navigate to their data? Could they understand that was their data and sort/filter it?

In the pilot study the regular class teachers, four teaching assistants, and four researchers (RA1, RA2, RA4 and RA5) were present. The study was split into three parts over two days, held a week apart. Part 1 was learning how to use the clip:bit, part 2 was collecting data using the clip:bit and part 3 was analysing data using the Cloudlet. RA1, RA2, RA4 and RA5 took observational notes during the three parts. RA2 took the clip:bits and uploaded their data to the Cloudlet.

In the main study the regular class teachers, T1 and T2 for Year 3 and T3 for Year 5 were present alongside the author RA1 and another researcher RA2. The study was split into four parts over several weeks. A further description of each lesson and its place in the data science pipeline will follow in Section 6.3.

As discussed in 4.3 a new novel methodology is being followed: participatory education for working with teachers in this technology design process. The following section describes in detail how this methodology was applied.

Participatory planning

For both studies teachers were met with, and it was discussed how the lessons would be primarily delivered by the author. Technology compatibility was checked: if the student devices would connect to the Cloudlet. The Pilot school used iPads while the main school used Chromebooks. Both devices were found to work with the Cloudlet. As part of the planning, it was also discussed what the students were currently learning in the schools. In the Pilot study the students were studying mini beasts, which was continued to be studied in our study. In the main study the Year 3 students were studying birds and Year 5 were studying different ecologies – an international one and a local one. It was agreed the data collection would be used to collect data about birds and a local ecology - the local beach. The lessons plans were created and agreed upon by the teacher and the researcher.

Participatory delivery

The author delivered all the lessons in both studies with the teachers present for support and feedback. Having developed the clip:bit and the Cloudlet the author used their technical knowledge to teach the students how to use them. They taught Year 5 in the main study how to code them. During delivery the author consulted with the teacher to ensure the lessons were going as expected. The teacher provided information about how the students were engaged (4), from their personal knowledge of the students and their behaviour, they advised the author on pace (1), content (2) and delivery (3). The teachers also taught parts of the lesson around subject, e.g. in the main study the Year 3 teacher spoke to the students about birds they had learnt in a previous lesson. This helped students relate what they were currently learning to what they had learnt in the past (5). The Year 5 teacher also took over teaching to remind students about a previous lesson on graphs. Again, the teachers were using previous learning to inform current learning. They were also providing subject knowledge to the students about birds and different ecologies.

At the end of each lesson, the author spoke with the teachers to confirm the lesson had gone to plan and make notes of any changes needed for the following lesson.

6.1.2 Methodology review

This new methodology was observed to work successfully in the two studies. The participatory planning helped situate the learning with the new technologies in the students existing learning. It ensured all the technology worked before the lessons. The time spent planning also helped develop a relationship of trust between the researcher and the teacher. In allowing an outsider to come in and teach their class for a significant amount of time, the teacher is putting a lot of trust in the researcher. By spending time with the teacher, gathering information about the students and the classroom the researcher was able to demonstrate to the teacher their level of commitment to the project, beyond developing the new technology.

The teachers in the main study stated how useful it was to have an expert in the room when teaching about coding:

Y3 “there’s value in having someone else.”

Y5: “Like we were saying: that other person, that’s someone different coming in that is maybe the focus.”

Y3: “and you're enthusiastic and passionate about what you're doing. Someone else delivering that is different than just your own class teacher.”

And how the collaboration worked well:

Y5: “You don’t want to rely on that person coming in, but that person's got that additional knowledge of the area. So, you're working together to do it. You're not passing it over to the person coming in. You're collaborating as the teacher and the adult.”

The constant feedback within and between the lessons ensured equal participation from the teachers. The researcher was able to integrate the teacher’s feedback into the lesson while it was happening as well as the next lesson. This is similar to how teacher trainees are taught – supervised lessons with feedback from their mentor.

6.1.3 Data collection

At the start of every classroom lesson a seating plan was drawn, students were numbered, and their names were temporarily matched with their numbers to ensure only students whose parents and guardians had given consent could be photographed, quoted or have their data recorded. For the data collection lessons which take place outdoors, the students' names were written next to their quotes and when transcribing these notes, the names were matched with the student identifiers. In the following sections quotes from students are labelled with that student identifier, e.g. S1, where the student was identified. The students' drawings and the data they collected using the clip:bits were collected through the Cloudlet. In the final lesson in the main study the students took two questionnaires, their answers to these questions were collected and analysed.

Finally, the students and teachers were interviewed. Nine groups of students were interviewed in groups of between three and five for approximately 20 minutes. Only the students whose parents consented have been included. During the interviews with children, it wasn't possible to identify the students individually, so they are presented anonymously. The three teachers involved in the study were interviewed together for two hours. The audio from these interviews was transcribed.

In summary, the data collected to evaluate TEDS includes:

- Student and teacher quotes from lessons both in the classroom and outdoors in the main and pilot studies.
- Researcher observational notes from lessons both in the classroom and outdoors in the main and pilot studies.
- Data collected from clip:bits in the Year 5 classroom (591 rows).
- Data collected from clip:bits on the beach with Year 3 (1067 rows).
- Data collected from clip:bits on the beach with Year 5 (1775 rows).
- 12 sets of results from questionnaire 1 with Year 3.
- 9 sets of results from questionnaire 2 with Year 3.
- Student quotes from group interviews.

- Teacher quotes from teacher interviews, student lessons and feedback sessions.

6.2 Pilot Study

Before the pilot study, a meeting was held with the teachers of the Year 2 classes to plan the delivery of the lessons as per the participatory plan. The teachers stated the students were ending the term learning about insects and “minibeasts”. The next planned lesson was to go to the local park and count different insects they could find then return to the classroom and draw pictograms of their data. The teachers agreed to run the lesson with the clip:bits and the Cloudlet. They wanted the students to choose any insects from a list of insects they would provide and to draw them on the data collection sheets. Therefore, the data collection sheets were not pre-populated with insect names or images.

The learning objectives of the lesson was for the students to learn the different names of the different insects, what they looked like and to recognise them in real life. The teachers also wanted the students to understand that different insects live in different types of habitats. They wanted students to move around the park and collect data in different habitats.

6.2.1 Training on the clip:bit

At the start of the lesson researchers RA1 (the author) and RA2 introduced the students to the clip:bit and demonstrated how to use it. They explained how the data was stored on the clip:bit and that it would be uploaded to the Cloudlet later. They pressed the buttons on the clip:bit and pointed out how the number in the top right changed. They showed the students how to change the section number by pressing the A button on the clip:bit. At the end of the lesson the students were given some time to press the buttons on their own clip:bits.

The classroom teacher wrote down the names of over 20 different insects found in the local park on the board. The students chose 6 each and drew them on their data collection sheet, sharing one sheet between 2 students. See Figure 65 left for a clip:bit that was filled in by two students.

Before the lesson the teachers stated they wanted students to make notes on their data collection sheet. As the drawn images needed to be photographed and uploaded to the Cloudlet a transparent sheet of paper was put over the data collection sheet for students to make notes without corrupting their images. This can be seen in Figure 65 right below. There was no time to photograph the data collection sheet between the students finishing their drawing and heading out to collect the data.

Figure 65 (Left) Insect pictures drawn by two students. (Right) A clip:bit outdoors with the transparent sheet

Before heading out to the park to collect this data the students were shown how to delete the practice data from their clip:bits. By pressing A and B on the micro:bit together this would delete the data and was also a deliberate action. Students could not accidentally delete their data.

6.2.2 Collecting data

Straight after drawing their images, the two classes went to visit a local park with the clip:bits. The park had an area of long grass, short open grass and a small section of trees. The two classes started together in the area of long grass and started to search for different insects, using their clip:bits to record what they had seen. The students worked in the same pairs as the classroom.

After some time, the students were split into smaller groups and taken to different parts of the park. They were told to change the section number to the area they were going to, everyone started in section 1, the short open grass was section 2, and the trees were section 3. Every group visited at least two areas before the two classes regrouped and headed back to school.

6.2.3 Analysing data

A week after the training and data collection lesson, RA1 and RA2 returned with the Cloudlet. After discussion with the classroom teachers, it was decided to display the data on the classroom screens to discuss it. The school did not have enough working computers for both classes to view the data on their own devices at the same time, and there was insufficient time to do each class separately. The students looked at the data on the main screen and drew pictograms into their Mathematics notebooks.

6.2.4 Pilot Study Analysis

All researchers compared notes and examined the clip:bit data, coming to four practical conclusions:

1. Students need more time training on the clip:bit.

The students needed more in-class clip:bit training time. After drawing the insects, the students only had a few minutes to practice using the clip:bit. The combination of not knowing how to use the clip:bit and being outdoors was too distracting for the students to use the clip:bits correctly. They couldn't hold the clip:bit and something else like a pen or a pair of binoculars. For example, they were too excited when they found a spider and would run over to where the spider was leaving their clip:bit behind.

2. The use of section numbers to identify multiple, different geographic regions within a study was too difficult for the children to manage on their own.

The data from the clip:bits showed invalid data, with either no section number change or section numbers up to 5 when there were only 3 sections. Although the students were told to change the section number when they moved to a new location, and some teachers and assistants helped them, most of the data was invalid. However, where the data was correct, the students were able to recognise that they found different insects in different habitats. When the children were looking at the sticker book a child stated:

"The one with the very very tall grass has the lady birds and bees and on the other side there is just spiders".

When asked why they thought there were more spiders in the wood, the students responded:

"The spiders don't like grassy areas".

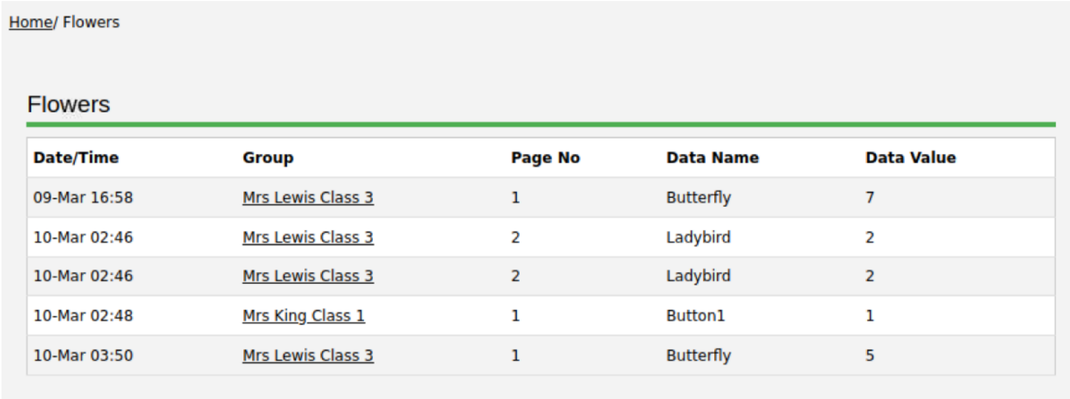
"They like the dark so they can camouflage".

3. The clip:bit battery switch was too fragile and the transparent sheet on top of the data collection sheet did not work.

The battery switch on the side of the clip:bit got caught in children's clothes and snapped off on two occasions. It also broke when in transport. This switch was too fragile for use by children. The transparent sheet made the data collection sheet too difficult to use. Students did not have time to make any notes. It crumpled too easily when the students tried to write on it.

4. The Cloudlet screens were too cluttered.

Teachers commented how the table of data presented by the Cloudlet included unnecessary information making the screen look cluttered and difficult to read for the children, see Figure 66 for the original student screen from the Cloudlet. It was also difficult to read the data from the back of the class.



Home/ Flowers

Flowers

Date/Time	Group	Page No	Data Name	Data Value
09-Mar 16:58	Mrs Lewis Class 3	1	Butterfly	7
10-Mar 02:46	Mrs Lewis Class 3	2	Ladybird	2
10-Mar 02:46	Mrs Lewis Class 3	2	Ladybird	2
10-Mar 02:48	Mrs King Class 1	1	Button1	1
10-Mar 03:50	Mrs Lewis Class 3	1	Butterfly	5

Figure 66 A student screen in the Cloudlet displaying data collected.

These observations were used to inform the main study. For the main study each of the stages was split into a specific one-hour long lesson. The students would have an hour preparing the clip:bit, an hour collecting data, etc. This would provide enough time for them to learn how the tool works before going outside. Section numbers and page numbers were removed from the process as well as the transparent sheets. The Cloudlet screen was also simplified - section or page numbers were removed, also the date and time were removed, and the headings simplified. The clip:bit version 2 was designed with holes to attach a lanyard so the students could let go of the clip:bit to use their hands. Version 2 also moved the battery switch from the side of the clip:bit to the battery pack itself.

6.3 Main Study

The main study took place in School 2; a primary school located in the United Kingdom. 37% of its students are eligible for pupil premium. The pupil premium grant is "*funding to improve educational outcomes for disadvantaged pupils in state-funded schools in England*." Pupils are eligible for pupil premium if they are eligible for Free School Meals or are a current or previously looked-after child [168]. In 2019 1.99 million pupils, 27% of all UK pupils aged 5-16, were eligible to receive pupil premium funding [209].

Seven lessons between two classes of Year 3 and Year 5 were run. Year 3 took part in lessons 1, 3 and 4. Year 5 took part in lessons 1, 2, 3 and 4. The lessons were delivered by researchers RA1 and RA2. The children from Year 3 had met RA2 previously during a different project. The three teachers also knew RA2 from previous projects. Table 17 shows how the lessons align with the data science pipeline.

The prepare stage is over two lessons, one for learning how to use the clip:bit and one for coding the clip:bit. Only Year 5 coded the clip:bit. In the Collect phase both classes headed outdoors for data collection, and clean, analyse and visualise all happen in the final lesson for both classes back in the classroom.

Lifecycle Step			**Lessons**	**Year group**
Prepare			1. Basic clip:bit operation	Year 3 Year 5
Prepare			2. Coding the clip:bit	Year 5
Collect			3. Fieldtrip outdoors	Year 3 Year 5
Clean	Analyse	Visualise	4. Cloudlet	Year 3 Year 5

Table 24 Lesson plans in the lifecycle

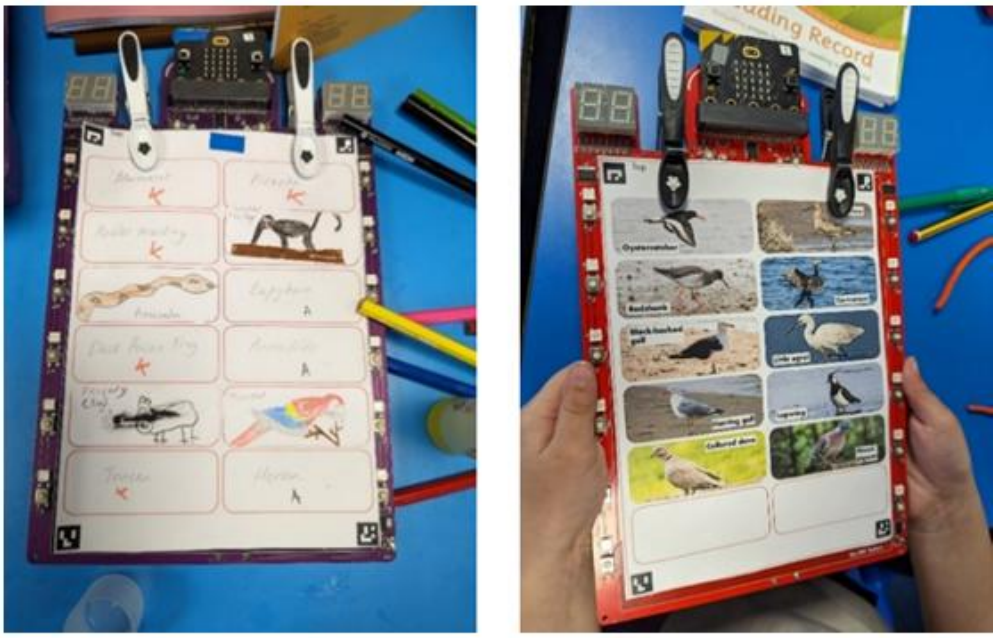
6.3.1 Data preparation: clip:bit operation

Following the pilot study with School 1 it was discovered that younger children needed more time using the clip:bits indoors and away from distractions. For the main study with School 2 the students were given more training on the basic operation of the clip:bit indoors. When School 2 was approached the Year 5 class teacher wanted to use the technology in their current topic of the Amazon rainforest. Given they could not fund a fieldtrip to the Amazon rainforest, it was decided that creating a virtual learning experience via watching a YouTube video of the rainforest and using the clip:bits to count the animals would make a perfect training opportunity. Watching a video meant students sat at their desks and did not have to carry the clip:bit, making it easier to share between students, reducing distraction, and avoiding any weather issues.

For Lesson 1 researchers RA1 and RA2 introduced the Year 5 children to the clip:bit. RA2 delivered a slideshow about their previous personal visit to the Amazon and introduced a list of animals from that habitat. Three different data collection sheets were handed out with the names of the animals, one sheet to each pair of students. The students' first task was to use their library books and iPads to find out what each animal looked like and to draw it onto a sticker. The stickers were subsequently added to the data collection sheet, Figure 67. This allowed the students to work collaboratively when generating images for the data collection sheets. It also allowed the students to work separately so they didn't have to share the sheet and saved time.

In Year 5 there were 12 pairs of students using 12 clip:bits. There were three different data collection sheets, the students either had a grey, blue or brown sticker on their data collection sheet. The three different types of data collection sheets were different to each other, e.g. the blue sheet had the capybara animal, but the brown and grey ones didn't. The grey sheet had a sloth next to button L5; the brown sheet had it next to button R6. This was so the students couldn't copy the group next to them and every group had to pay attention to the screen and the clip:bit to find their animals.

Year 3's topic was local birdlife. The area where the school is located is home to many different types of wading birds which can be spotted year-round. Again, RA1 and RA2 introduced the clip:bit and the different types of birds to the students. They watched a video and used the clip:bit to count the birds. For Year 3 more time was spent on how to use the clip:bit and how to recognise different birds. Year 3 did not draw the birds. The class teacher felt that the students' drawing skills and the similarity between the birds would make it difficult for them to recognise a bird in the video from a hand drawn picture, Figure 67(b).



(a) (b)

Figure 67 (a) Hand drawn student images (b) Teacher provided bird images.

6.3.2 Data preparation Analysis

Children in both Year 3 and Year 5 were able to use the clip:bit successfully and understand its purpose and use. They were able to accurately record data using the clip:bits.

Year 5

For example, in Year 5 the pairs of students all recorded around the same number of Macaws. Figure 68 shows the raw data downloaded from the clip:bits and filtered to Macaws. It should be noted that the Time column shows the time since the device was turned on, which may differ between devices. The mean number of macaws they counted was 3.5, the mode was 3 and the standard deviation was less than 1 at 0.755929. Five pairs of children with the devices posit, vuvov, zizev, vugaz and zivap all recorded 3 macaws. Where the children recorded more, it can be seen from the data that they recorded this extra data at a different time. E.g. the group totat recorded 3 macaws at 791, 794 and 795 seconds but then recorded a fourth at 1236 seconds. The pair of the students with the device zepaz seemed to record two macaws together at 19 and 23 seconds but then recorded three at 622, 626 and 627 seconds. From this data it can be deduced that all the students saw three Macaws at once and were able to use the clip:bit to record this accurately.

Time (seconds)	section	page	index	value	device	name	COLOUR
791.52	1	1	7	1	totat	Macaw	BLUE
794.76	1	1	7	1	totat	Macaw	BLUE
795.39	1	1	7	1	totat	Macaw	BLUE
1236.08	1	1	7	1	totat	Macaw	BLUE
859.57	1	18	7	1	vugaz	Macaw	BLUE
860.51	1	18	7	1	vugaz	Macaw	BLUE
861.35	1	18	7	1	vugaz	Macaw	BLUE
486.74	1	1	7	1	zivap	Macaw	BLUE
487.52	1	1	7	1	zivap	Macaw	BLUE
488.4	1	1	7	1	zivap	Macaw	BLUE
5.05	1	1	7	1	zoveg	Macaw	BLUE
486.74	1	1	7	1	zoveg	Macaw	BLUE
487.52	1	1	7	1	zoveg	Macaw	BLUE
488.4	1	1	7	1	zoveg	Macaw	BLUE
214.52	1	1	2	1	pozit	Macaw	BROWN
217.18	1	1	2	1	pozit	Macaw	BROWN
217.7	1	1	2	1	pozit	Macaw	BROWN
205.35	1	1	2	1	vuvov	Macaw	BROWN
205.82	1	1	2	1	vuvov	Macaw	BROWN
206.3	1	1	2	1	vuvov	Macaw	BROWN
19.09	1	1	2	1	zepaz	Macaw	BROWN
23.43	1	1	2	1	zepaz	Macaw	BROWN
622.47	1	1	2	1	zepaz	Macaw	BROWN
626.43	1	1	2	1	zepaz	Macaw	BROWN
627.27	1	1	2	1	zepaz	Macaw	BROWN
515.28	1	1	2	1	zizev	Macaw	BROWN
515.91	1	1	2	1	zizev	Macaw	BROWN
516.43	1	1	2	1	zizev	Macaw	BROWN

Figure 68 Data from the clip:bits filtered to macaws.

Table 25 below shows all the data the students collected grouped by data collection sheet type. Row 2 shows the group numbers, e.g. groups 1, 2, 4 and 11 had the grey data collection sheet.

	Grey				Brown				Blue			
Animal	**1**	**2**	**4**	**11**	**3**	**7**	**8**	**10**	**5**	**6**	**9**	**12**
Anaconda									3	2	2	2
Aracari	4	1	0	1	1	0	4	0				
Armadillo	6	2	0	2	2	2	8	2	39	3	1	2
Black poison frog									2	2	1	1
Capybara									3	3	4	5
Caiman	6	2	0	2	2	2	8	2				
Capuchin Monkey					2	2	0	2				
Coatis					4	0	91	3				
Giant otter	16	4	0	5	4	10	10	3				
Heron									5	1	1	2
Howler monkey	8	6	0	7					9	6	6	6
Jaguar	6	3	3	3	3	5	7	3				
Lizard	2	2	0	2								
Macaw					3	3	5	3	4	3	3	4
Marmoset	11	1	0	3					2	1	0	0
Milk frog					1	2	21	1				
Peccary					3	1	2	1	9	0	0	0
Piranha	6	1	1	1					2	1	1	2
Sloth	1	1	0	1	1	1	3	1				
Spider monkey	7	3	0	3					3	2	3	4
Toucan	3	1	0	2					9	1	1	2
Vulture					8	8	9	4				

Table 25 Year 5 training data

In the interviews the Year 5 teacher liked the lesson and felt it went well. She felt watching the video gave the students **focus** and linked the topic to technology. She also felt that it highlighted to students how academic subjects are not standalone subjects but overlap like **real life**.

T1: "*I think it worked really well. Actually, watching the video with something that was specifically looking for made them focus on what they were watching more, and it linked that topic to technology for us really… I think it's important from a curriculum point of view and timewise, but also for their future to realise that it isn't, they're not standalone subjects, they do overlap.*"

Year 3

When questioned how they knew which button they pressed, a student in Year 3 answered:

S25: *"Because it lights up next to the picture."*

The students in Year 3 were more confident in using the clip:bit. They had no fear of breaking it by pressing the buttons too fast or too hard. One scene from the bird video did have a lot of birds in the background and several children did press the button a lot and had over 100 in their data logs. In contrast Year 5 were more careful with the device.

S16 and S7 (Year 3): *"I'm going to 99!"*

In the student interviews all the groups of students were able to identify the clip:bit, the micro:bit and explain their uses.

Year 3: *"It's like a clipboard with batteries. It has electricity too, to connect the numbers to that thing on the top. You can take it off, put it into the cloud and the cloud will store it into the computer so you can see how many birds you got".*

Year 5: *"It contains the data that we collect then it sends it to the cloud."*

All the students were able to work together in pairs **collaboratively** to collect the data. While watching the video there was a lot of consultation between the pairs. In Year 3 the students consulted each other to confirm identification independently, they did not ask the teacher for help.

S18: *"Do I press it?"*

S17: *"It's that one!"*

S26: "*It's that [points at picture].*"

S27: "*Is it that one?*"

During the lesson the Year 3 teacher showed the students their Mathematics books to explain how the numbers of birds they were counting can be used in graphs later. They wanted to connect this **cross-curricular** Geography and Computing lesson to a third subject: Mathematics.

T1: "*I kind of interrupted at one point to say: Make this link between what we've just done in Maths, and the graphs and the charts and the collection of information, to what you are doing now*."

After the lesson with Year 3 the teacher reflected that the task was good for **collaboration.**

T1: "*I could see the kids watching and trying it…. They were excited. They were doing it; it surprised me how well they did it together*."

6.3.3 Data preparation: Programming the clip:bit.

The clip:bit can be programmed using MakeCode, however in consultation with the Year 5 teacher, it was decided the full program to record items onto the clip:bit is too advanced for Year 5. Coding the clip:bit is optional in the lifecycle. The toolkit includes all the code needed to collect data using the clip:bit, it is up to the classroom teacher whether to show some or all of this code to the students. At the start of the study, a discussion was had with the Year 5 teacher. She agreed the entire code was too complicated but wanted the students to experience coding the clip:bit in some way. Therefore, in this lesson RA1 and RA2 taught the students how to code some but not all parts of the clip:bit. They **scaffolded** the code for the students, showing them parts of the completed program.

The students were first taught how to code the micro:bit, how to scroll their name across the LEDs and how to code a button. This introduced the concept of an event, and a button press through **logic**. Then the clip:bit program was shared with the students. Figure 69 is an example set of blocks a student created using MakeCode.

They learnt how to change the label of a button (a), how to react to a clip:bit button being pressed (b) and how to change the colour of a clip:bit LED using loops and the micro:bit button (c).

The students enjoyed changing the colour of the clip:bit LEDs. They used **tinkering** to change the code in MakeCode and see what happens on the clip:bit. Some students used loops to create animations on the LEDs.

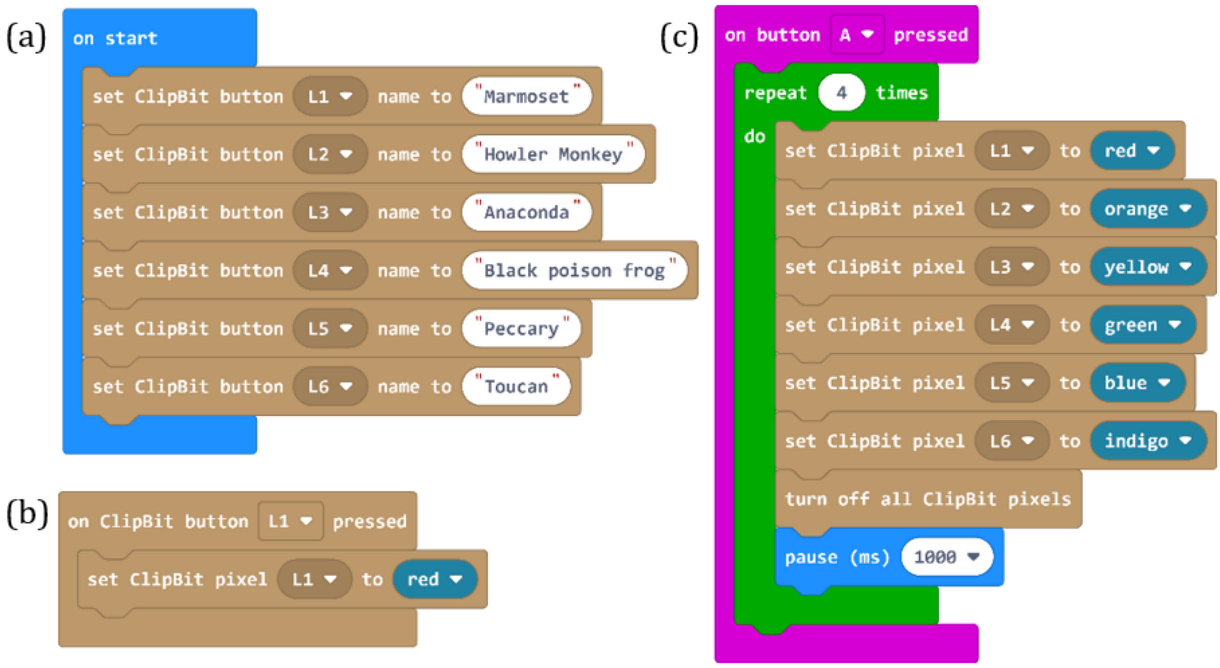


Figure 69 Lesson 2 MakeCode blocks.

6.3.4 Data preparation: programming analysis

The students were successful in coding the clip:bits. All of them entered the text labels and changed the LEDs. Some of them used loops to create animations of the lights. The students were very happy with their clip:bits, especially when they changed the colour of the LEDs. They wanted to demonstrate it to the researchers.

S7 "Look, it's a rainbow!"

S6 "Watch my rainbow."

S1 "Watch my buttons change colour."

Adding the text, e.g. typing Marmoset, Howler Monkey, etc. into the text box is a simple task but the students didn't find it tedious. They were confident because they knew what to do.

S4: *"It is a bit boring, but still fun. I know what to do."*

The Year 5 teacher found the coding of the clip:bit extremely useful for the children's confidence, they felt it gave them a role in the project not just as students learning but as creators of the technology. It also made them think of the use of coding in the **real world**, from a career's perspective. Furthermore, coding the clip:bit gave it a real **purpose**.

T3: "*They were using it for a real purpose: they were going to go on and use that to collect some data later on. And I think some of them were asking about, well, how would we use this in the real world and jobs? So, I think it gave it a real purpose for them*."

The physical nature of the clip:bit also gave the children a sense of **purpose.**

T3: *"...they're suddenly seeing coding on something different, a different way of coding. It's coding for a purpose rather than just making something move on a screen."*

The teacher commented how the order of the lessons made sense: the students needed to see the clip:bit in action before they could code it. This is the opposite to how many Computing lessons with physical computing are run. Students will code an item then use it but given the complexity of the clip:bit they experienced using it before they coded it.

T1: *"I think it was good for them to know what the purpose of the coding was for that activity. First this is what you're going to use it for and then go back and do the code. I think was the right order for them to understand why they were doing it."*

Finally, in this lesson the students were briefly shown their data from the previous lesson on the teacher's screen. Immediately the students recognised their images.

S18: *My sloth! That's mine, look its mine!*

6.3.5 Data collection fieldtrip

For the fieldtrip Year 3 went to a local beach to look for wading birds and Year 5 went to the same beach on a different day to look for beach creatures and plants.

Year 3

As Year 3 were looking for birds they were given a demonstration on how to use binoculars before visiting the beach. Bird images were placed around the school playground and the students practised taking turns using the binoculars and the clip:bit to recognise the birds and count them, Figure 70.

Figure 70 Year 3 students practising using binoculars in the school playground.

On the beach the children split into smaller groups with a member of staff or RA1 or RA2 supervising them. They stayed on the promenade to look for birds through the binoculars, Figure 71. The students worked in the same pairs they had been assigned in the classroom. There were 11 clip:bits between the students.

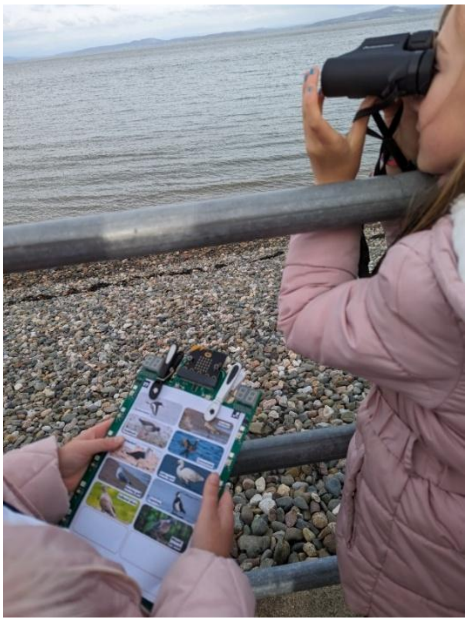


Figure 71 Year 3 students working together to look for birds with a clip:bit on the beach.

Year 5

Between coding the clip:bits and going to the beach, Year 5 had studied beach creatures and plants as a second habitat to compare to the Amazon rainforest. The teacher used the images from these lessons on the clip:bits, see Figure 72.

Figure 72 Year 5 beach data collection sheet.

Year 5 spread across the beach to find the different creatures and plants. As before Year 5 did not have the exact same sheets. There were two different sheets, labelled with blue and brown stickers as seen above in Figure 72.

6.3.6 Data collection fieldtrip analysis

It was found that all of the students were able to use the clip:bit successfully outdoors and recognise and classify different birds, plants and animals. Again, they were able to work **collaboratively** and were **engaged** in the task. Both classes were able to spot and identify several different birds and animals from their sheets. As the students were spread across the beach it is difficult to confirm how accurate their data collection was, because it was not a controlled environment.

Year 3

There were 10 pairs of students collecting data and 1 group of three totalling 23 students. Because Year 3 practiced looking for birds in the playground before the beach, the practice data had to be deleted. From the pilot study, it was found that students would press the delete button. The students in Year 3 were told to only press delete in the playground. However, two groups deleted their final data.

When the students delete all their data, the data is deleted from the log, and a row is added. The row has the name “DELETED” to help identify a deliberate deletion. The two groups that deleted their data are not included in the table Table 26 below.

The top row is the device number; this is used to identify the groups instead of the children’s names. The left column is the name of the bird they were collecting. There were 10 different birds to collect and 2 blank squares next to buttons L6 and R6. From the data below it can be seen that 5 groups pressed L6 and 5 groups pressed R6. Eight of the nine groups pressed a button with no picture next to it.

While the data may not seem consistent, there are several reasons for this. The students were spread across the promenade looking into the distance for the birds. The groups were not going to see or count the same birds as each other as the area was very wide. As it was seen from the pilot study, young students can be distracted when outdoors. They were at times too busy looking for birds or distracted by being outdoors to press the button on the clip:bit. Large groups of birds are difficult to record, not to mention birds that fly away. Students didn’t know whether they were looking at the same bird that flew away and came back.

Animals	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Black-backed gull	3	20	6	2	4	2	17	0	3
Collared dove	3	4	0	0	2	0	0	0	8
Cormorant	0	43	1	6	1	0	2	0	0
Curlew	36	12	10	4	2	15	10	0	1
Herring gull	11	27	12	3	5	64	62	6	1
L6	0	36	0	1	0	1	0	3	4
Lapwing	1	25	0	3	0	0	0	0	0
Little egret	3	17	5	16	1	2	5	0	0
Oystercatcher	125	31	7	2	3	60	214	1	0
R6	0	1	3	0	2	0	2	0	1
Redshank	10	12	0	13	27	3	5	2	3
Wood pigeon	1	2	0	1	0	1	1	2	1

Table 26 Year 3 data from the beach

The students were excited to collect the data on the beach.

S7: S7: "*There's a black-backed gull!*"

RA2: "*How many?*"

S7: *"Let me check…"* (uses the binoculars)

S18 and S22: *"We saw 4 curlews."* (points at the picture of the curlew on the clip:bit)

S18: *"There's an egret!"*

S22: *"Where?!?"*

S18: *"I see it, yeah, press it!"*

S22*: "I've already pressed it."*

The students were confident in using the clip:bit. One student was very comfortable deleting all their data and starting again.

S8: "*I deleted all my numbers, so I'm entering adding them all again… oh no, I wanted to go to 32, but I went to 39.*" (deletes them again)

On the beach the Year 3 teacher was impressed with the students' **focus** after 30 minutes of looking for birds.

"This is somewhere they come to all the time. They either walk past it on the way to school or drive past it, but it's not somewhere they've engaged with. No one has spent the time to engage them. It's great that the birds on the list are not just some random birds you see anywhere, the fact they are local birds that they can recognise it's brilliant. It makes it really ***relevant*** *to them."*

She was also pleased with how the clip:bit helped to **engage** the student in the task:

"Counting gives them an objective, it's something ***meaningful****. It makes them feel important; they get to use this wonderful piece of equipment to do it with. We're not just showing them the birds, again we're* ***engaging*** *them in a task with specialist equipment, that other people don't have. Or that they've not used before.*"

Year 5

Year 5's data was also very mixed. 9 groups of students collected 1775 pieces of data. The adults did not agree how to measure certain items before leaving the classroom. For example, RA1 and RA2 taught the students with blue sheets to measure bladder wrack using their hands as measuring tools. Groups 1-5 had smaller numbers (3, 5, 12 and 12) but the groups with brown sheets measured the bladder wrack by the number of seeds they could see, e.g. groups 6, 7, 8 and 9 recorded 125, 38, 48 and 79 bladder wrack respectively. This happened again for multiple items such as lugworm cast and hornwrack. The substantial difference between the average totals for blue (56.4) and brown (373.25) can be seen in the bottom row of Table 27 below.

	BLUE					BROWN			
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Barnacle	3	8	8	1	3				
Bladderwrack	3	5	12	0	12	125	38	48	79
Channelled wrack						3	42	12	1
Cockle shells	3	3	10	3	2				
Crab	1	8	4	2	4	4	15	14	9
Dogwhelk						4	35	2	1
Egg/knotted wrack						9	90	16	6
Hornwrack	5	4	2	1	2	17	56	8	5
Jellyfish						0	0	0	0
Lugworm cast	1	22	2	1	17	2	92	108	102
Mussel shells	2	5	5	3	5				
Periwinkle	10	25	2	1	21				
Sand mason worm						0	2	3	0
Sea beet						26	50	27	27
Serrated wrack	0	1	1	0	0				
Spartina clump	7	11	7	7	6	51	263	51	1
Thin tellin	0	3	1	1	0				
Whelk egg case	2	2	1	1	0	2	44	0	3
Grand Total	**37**	**97**	**55**	**21**	**72**	**243**	**727**	**289**	**234**
Average	**56.4**					**373.25**			

Table 27 Year 5 data from the beach

In the interviews most of the students stated that they preferred using the clip:bit outdoors, one group in Year 5 stated how they could pay more attention outdoors:

"Because in the classroom you just watched a video. But on the beach, you can find well different stuff."

"You can see it up close. You can see it in person."

"Compared to the video, if you leaned in you could see more stuff."

They compared the activity to another outdoor activity when they had to use pen and paper and they all stated how difficult writing outdoors was on paper, in the wind while trying to lean on something.

The students were very competitive when gathering their data, they raced over to find items they didn't have from their lists.

S7: *"I found Hornwrack!"*

S17: *"Where?"*

S18: *"Where? We need it."*

S17 and her partner didn't respond when S7 said he found a fish (not on the list) but rushed over when he found a crab (on the list). RA2 observed three groups of students collect data and noticed how they didn't trust other groups when they said they had found something. She questioned S17 about this.

RA2: *"If group 2 said they saw something, but you hadn't seen it, would you trust them?"*

S17: *"No. Cause they could be wrong. I need to see it so I can match it."*

RA2: *"What if RA1 said they saw it?"*

S17: *"Yes. Cause they're an adult."*

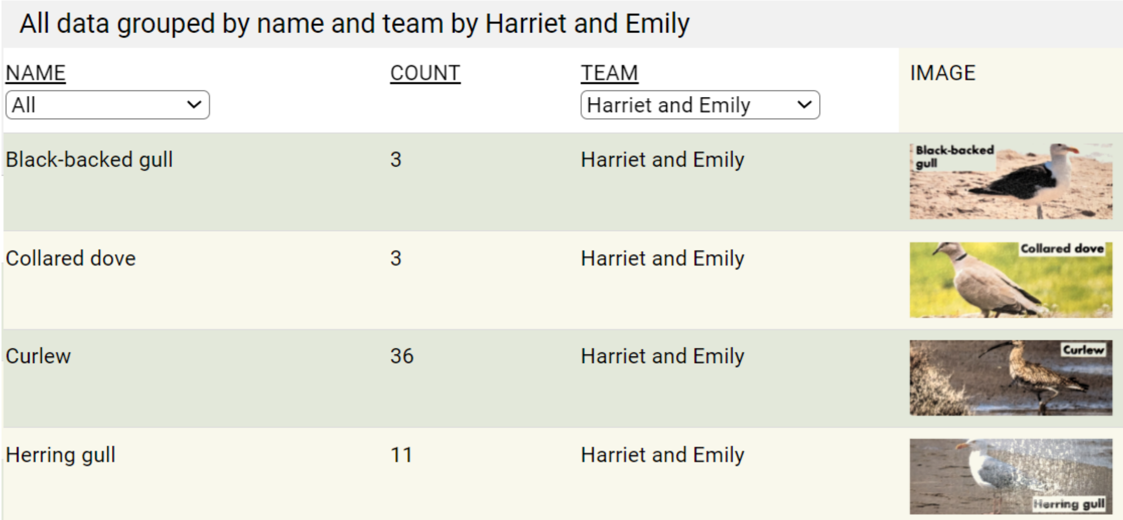
In the teacher interviews, the Year 5 teacher agreed that the clip:bit helps keep the students **engaged**:

"They're staying engaged. I think that makes the difference. It keeps them engaged."

6.3.7 Data analysis and visualisation: Cloudlet

The final lesson for both classes was back in the classroom to analyse and visualise the data they had gathered from the beach. The students were shown how to connect to the Cloudlet and how to navigate to their data. After feedback from the pilot study, the names of the students were added to the data. This way the students would see their own names in the data, see the new layout (Figure 73) filled in with data from Year 3 students. All names have been replaced with pseudonyms.

The students were shown by RA2 how to clean the data, for example the data in Figure 73 is filtered by the team "Harriet and Emily" and sorted by name.



All data grouped by name and team by Harriet and Emily

NAME (All)	COUNT	TEAM (Harriet and Emily)	IMAGE
Black-backed gull	3	Harriet and Emily	Black-backed gull
Collared dove	3	Harriet and Emily	Collared dove
Curlew	36	Harriet and Emily	Curlew
Herring gull	11	Harriet and Emily	Herring gull

Figure 73 Year 3 bird data filtered by team.

The students were given a questionnaire to answer about the data (Figure A1 in Appendix A) then they moved on to creating a bar graph of a data for a specific team. They were encouraged to choose their team name when filtering the data. The students answered a second questionnaire about the graph, Figure A2 in Appendix A. All names have been anonymised in the questionnaires.

The final task of the lesson was to create a visualisation of the data using the "Stickerbook" link.

6.3.8 Data analysis and visualisation analysis

When asked what is stored in the Cloudlet in the group interviews, 33 students answered with a range of correct answers. 45.5% stated that it contained data or information. 15.2% stated that it contained electronics, "wires and stuff" and 9.1% stated that the Cloudlet had a brain. Another 6.1% stated it had lights inside and 6.1% also said it had a micro:bit inside. The final students answered that it had "private stuff", "a memory", and "DNA". Six of the groups were asked a follow up question: what kind of data? 12 students answered:

- Data from the clip:bit.
- Bird data.

- It connects to the clip:bit so that it can tell you what birds you found.
- How many numbers you found.
- Numbers.
- Our data.
- The birds.
- Kind of animals.
- Your drawings.
- How many times you clicked it.
- The stuff from the rainforest.
- From the microbit what we sent to it.

When asked if the Cloudlet is online, most of the students in the groups answered no, 20 out of 25. Looking at the quantitative data gathered from the questionnaires: only ten Year 5 students' parents/guardians completed the consent form for this study. These ten students were mixed into groups of other students therefore their answers could not be gathered separately for the research. All the students' parents/guardians in Year 3 signed the consent form. For Year 3, looking at questions 5 to 12 in the table questionnaire Figure A1 in the students average score was 73%.

In the graph questionnaire, Figure A2, the average score was 84%. The graph questionnaire was similar to the first questionnaire asking the students specific questions about teams and objects that had been found. It also included a question about data that wasn't present, question 4 for Year 3: "Name one bird your team didn't see". In answering the questionnaires Year 5 used more of the features of the website, sorting and filtering, to analyse the data. Most of the students in Year 3 scrolled through the data to find the answers to the questionnaire.

The graph was one of the more popular parts of the Cloudlet. In interviews many Year 3 students stated how it was easier to use, and they liked how it looked. Year 5 also thought the graph was useful:

"On the graph it shows it more clearer."

"On the table you have to filter. On this it's just there."

The Stickerbook is an interactive way to visualise the numbers and images together in the Cloudlet. The students can add the images to a custom background image and move them around, e.g. they can add bird images to the sky or to the beach on a background image.

With the Stickerbook Year 5 were carefully placing the plants animals on the background. They were thinking about where the plants animals lived on the beach and placing them in those locations on the background image of the beach. Year 3 were not as considerate when using the Stickerbook. One set of data had over 100 of a type of bird and many of the Year 3 students tried to add all 100 images to the background. They did not consider the location of where they saw the birds. Several said they just wanted to have fun with the Stickerbook. In the student interviews a group of Year 5 students stated how they wanted to use the Stickerbook to share information with people who didn't know anything about the beach as it is *"easier for them to understand*" and a second group stated

"They'd have more fun with the Stickerbook."

The Year 5 teacher was impressed with how the students remained **focussed** throughout this lesson around data and numbers:

"I think they did really well with it and the fact that they stayed focused for the length of time they did, they wouldn't have stayed focused like that in a normal Maths lesson."

Explaining why they stayed **focused** she points to the **ownership** of the data:

"I think because it was their data, their information, it's their names that are up there."

The realness of it:

*"And again, it was sort of **real things** that they'd seen."*

And the **usefulness** of the technology:

"They're not just an add-on to your computer. You can actually use them for something."

6.4 Systems evaluation

This section looks at TEDS from a technical point of view. It evaluates the battery and storage capabilities of the clip:bit and the Cloudlet.

The clip:bit runs on 2 AA batteries. A test was done to see how long the batteries would last. A clip:bit was given new batteries. One RGB LED and the 7-segment display were turned on and data was recorded to the clip:bit every minute. The clip:bit kept recording for 46.5 hours.

Given the data the students were storing: count, button_name, object_name and device: each clip:bit can store 4,734 rows of data. This was tested using the following program:

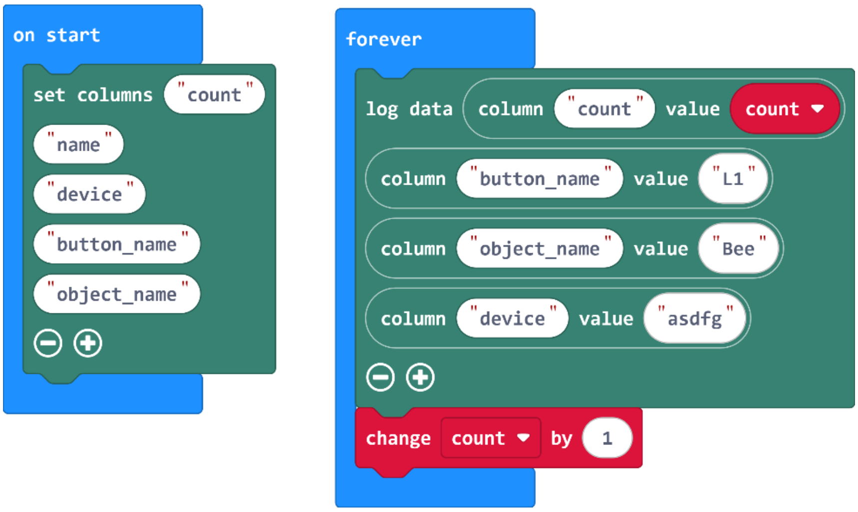


Figure 74 Testing the micro:bit data log.

After a few minutes, the micro:bit was plugged into a computer, and the data log was opened. It said "LOG FULL" on the top of the file and there were 4,734 rows of data. If a student was recording 12 different objects, they could record 394 of each one.

The Cloudlet uses approximately 350millamps of current. With a 10,000millamp hour battery, a Cloudlet will run for up to 28 hours, although this would be affected by Wi-Fi usage, LED usage and CPU load.

Analysing the Cloudlet storage, the textual data of 3710 records take up 1.3MB of space, each row takes up around 350bytes. 12 data collection images take up 1MB of space, approximately 80KB per image. As the textual data is so small it can be discounted when calculating how much data the Cloudlet can store. The Cloudlet has a 16GB SD card of which approximately 5GB is used by the operating system and software, leaving around 8 to 10GB of disk space free. This means the Cloudlet can store 96,000 images, 8000 clip:bit sessions of data.

6.5 Gap Analysis

This section details how the gaps found in research were addressed.

Gap 1: Longitudinal studies and progression

Many of the studies in the related work were completed over one or two short lessons, it was unusual for lessons to occur over a period of time. TEDS supports longitudinal studies. It follows a data science pipeline. The students spent time in each section of the pipeline. The lessons all contained different learning and skills cultivated over this time. Being able to store the data collected from the children also supports much longer studies, e.g. these students could return to the beach in a year's time to count the birds again and compare the different data sets. The evaluation of TEDS did not complete a longitudinal study, but it has the potential to.

TEDS supports progression. Coding the clip:bit is optional. In this study, Year 3 could use the clip:bit without coding it. Year 5 were able to code it then use it. This supports progression of students' coding skills. How you view the data in the Cloudlet can also be filtered by the age of the children. Younger children could use the Stickerbook to place items on the screen and visualise the data. Older children can create bar graphs to visualise it. The teachers can choose which tools the students use in the Cloudlet based on their age and what the curriculum requires them to learn.

Gap 2: Cross-curricular education

In related work many of the studies with micro:bit were in subjects related to Computing or IT only. TEDS supports cross-curricular education. Educators can teach multiple subjects with TEDS. TEDS supports this as it allows educators to choose what data the students collect through the custom data collection sheets. This data can be suited to different topics the students are learning in different subjects. In this study the students in Year 5 were learning about different ecological environments to compare and contrast them, the Amazon rain forest and their local beach. The students in Year 3 were learning about birds in their local area. Both topics are situated in Geography. TEDS supports the teaching of Mathematics through analysing the data in the Cloudlet, sorting, filtering, totalling and graphing the data are all part of the Mathematics curriculum. How TEDS covers the data science parts of the UK primary school curriculum is covered later in this chapter.

Gap 3: Data storage for analysis

This study uses the micro:bit data storage to store data the students are collecting. This allows the students to analyse the data in a more thorough way. The students in Year 5 saw the code of how the data storage was setup, labelled and how a button press saved the data to the micro:bit. Both sets of students understood that the data they had collected was stored on the clip:bits and the Cloudlet. They were able to successfully analyse it, answering questions about the data and scoring high on these tests.

Gap 4: Age range

In Chapter 3 it was discovered how dominant the micro:bit was in primary school education. This gap was not addressed but utilised. TEDS is suited for primary school as it is a cross-curricular tool and students in primary schools learn more often in cross-curricular classrooms. They are not segregated into different subject classrooms and taught one subject at a time like many secondary schools. TEDS is a set of cross-curricular tools than supports multiple subjects to be taught together. TEDS was successfully used in two different year groups in a primary school.

Gap 5: Ethics

In the final interview with students the author questioned them about the ethics of collecting and storing data. Most of the students were conservative with regarding to sharing the information. They were able to describe permission and privacy around data.

R2: Would you be happy if someone else saw this data?

- No, it's private.
- No that's my data.
- You cannot see my data! Without my permission and *other student.*
- No, cause it's mine.
- No one else can see this data. Only R1 and R2.
- I don't trust the internet. If I've uploaded that to Google they could share it to wherever. Even if they delete it, it's out to the public. Because it's my private data, it's not anyone else's data.

Students understood that their data was stored in the Cloudlet, and the Cloudlet was not online, they were happy with this given their mistrust of the Internet.

R2. Is the cloud on the internet?

- No.
- No, we're connected to a cloud.
- Because other people can't see it. Cause it's just stored here.

R2: Is this cloud on the internet?

- No.
- It's private.
- Cause you need to type cloud.local, you need to connect it.

R2: Why is it not on the internet?

- It’s private. If you don’t have it plugged in you can’t see it.

R2. Why?

- So people can’t hack it.
- So that people can’t go onto things that are very very private on there.

R2: What people?

- All: hackers!

6.6 Curriculum analysis

As mentioned in Chapter 2 there are some data science curriculum requirements in multiple subjects in the UK primary national curriculum. Here it is highlighted which subject TEDS specifically supports the teaching of data science through.

Geography: Collect data through experiences of fieldwork.

TEDS supports fieldwork. The clip:bit and the Cloudlet are mobile, lightweight, battery powered devices that can be taken outdoors. The clip:bit is also sturdy and be carried around the neck with a lanyard. In interviews educators have expressed their frustration with having to justify fieldtrips. Given the number of adults to children ratio they must abide by for health and safety, it costs a lot of staff time to take a class on a trip outdoor away from the classroom. They say how using TEDS can help justify taking students out.

“And it's a good way for us as teachers to be justified in taking them out. We want to get them out more. We want to get them out on the Bay and outdoor learning more.”

TEDS also supports different types of fieldwork. The teacher can add any image to the clip:bit for the students to look for. They can add images for multiple fieldtrips around birds, plants or even traffic. It can collect different types of data for different fieldtrips.

Science: Identify and name animals or plants they are looking for and Art: and sketch them

TEDS helps support students learning to identify and name animals or plants they are looking for and sketch them. The students can draw the images of the animals or plants they are looking for onto the data collection sheet. Educators state this helps focus the students.

T1: *"I think the drawing of the animals: it definitely worked. It focussed them in on what they were looking for as they were drawing."*

T2: "*Yeah… drawing it makes them observe it."*

TEDS supports the teacher providing the images themselves. The teachers commented that at times, the images rather than the drawings were more useful as the level of detail in the image was too complicated for the children to draw.

T1: "*But I think with the beach one, with the seaweed, etcetera, I think the actual images* ***were*** *better than what their drawings would have been because it would all have looked the same in their drawings cause it's very hard to draw those distinguishing features."*

Science: Apply their Mathematical knowledge to their understanding of science, including collecting, presenting and analysing data.

TEDS allows students to apply their Mathematical knowledge. In the study, the students had to find the team with the highest number of items. This involved applying their Mathematical knowledge to a set of numbers. Other tasks the teacher could use TEDS for is working out the average number of items or the total number. The Year 3 teacher was very keen for the students to connect their Mathematical knowledge with the learning they were doing with TEDS.

T1: "*I kind of interrupted at one point to say: Make this link between what we've just done in Maths, and the graphs and the charts and the collection of information, to what you are doing now*."

Computing: Combine a variety of software on a range of digital devices to analyse and present data

TEDS supports students using a variety of software on a range of digital devices to analyse and present data. The students use MakeCode to program the micro:bits. The micro:bits clip into the clip:bits which they use to collect their data. The clip:bits transfer their data to the Cloudlet another digital device, to analyse and present their data. The students were able to recognise the different digital devices that make up TEDS and explain their functions in the final interviews. The clip:bit:

- Year 3: *"When you press one of them buttons the buttons count up like 1 2 3."*
- Year 5: *"If you see something on the video you click the button and it counts. You click it however many times and it counts for you."*

The Cloudlet:

- Year 3: *"We went on a trip to the beach and we pressed how many types of birds and how many we saw and it went to the cloud."*
- Year 5: *"It stores all the information. It collects all the data and saves it."*

Mathematics: Answer questions about totalling and comparing categorical data

From the results of the questionnaire in Year 3 it can be stated that the students can successfully analyse data using TEDS. They were able to answer questions about totals of data and comparing data from different birds and different teams.

Mathematics: Interpret and present data using bar charts and tables.

From the results of questionnaire 2 it can be stated that students were able to use bar charts to interpret data.

6.7 Aims and requirements analysis.

This section looks at the aims requirements from Chapter 4 and evaluates how they were (or were not) done.

TEDS support educators to teach data science in primary schools (A1) using the readily available micro:bit (A2). Students can collect real life and local data (A3) covering multiple subjects beyond Computing (A4). Educators can store this data for analysis and visualisation in the future (A5). They can structure their lessons using a data science pipeline as a guide (A6). They can use this toolkit to support progression over different age groups (A7) and cover the important topics of ethics and privacy (A8).

ID	Pragmatic requirements (cost effective)	Evaluation
R1	It must be cheap to install and maintain.	The clip:bit and the Cloudlet are cheap alternatives to other data collection devices. There are no installation or maintenance costs.
R2	Utilise an existing resource and/or use open-source tools.	Many countries have had a national rollout of micro:bits with over 10 million sold worldwide. The platform to program the clip:bit, MakeCode, is open source.
R3	Tools need to be shareable between children and classrooms.	The clip:bit allows up to 12 items to be collected. Pairs of students were able to share a clip:bit, both classifying and searching for 6 items each child. The clip:bit can be stacked to store and share between classrooms. The Cloudlet has a large amount of storage for multiple classes to use and store their data.

Table 28 Pragmatic requirements evaluated (cost effective)

ID	Pragmatic requirements (accessible)	Evaluated
R4	It must be accessible for novice educators to use, setup and maintain.	The code that runs on the clip:bit can be downloaded by the educators, they don't need to know how to write code to use the clip:bit. The Cloudlet is a website that users can navigate and use easily.
R5	The tools should be simple and familiar.	The students state how the clip:bit looks like a clipboard. They also recognised the micro:bit from previous lessons. The tables in the Cloudlet look like tables of data with common features like headings you can sort and filter.
R6	The hardware and software need to be installed and used without needing administrative access to school devices.	Both tools did not need installing. No administrative access was needed onto the school's system to use either the clip:bit or the Cloudlet.
R7	The tools should be usable with any type of school device, e.g. tablet or laptop.	Both tablets and laptops were able to code the clip:bit and view data on the Cloudlet.
R8	The tools should be setup without the need for specialist apps on mobile devices.	No apps were needed to setup or use the tools. The users accessed the Cloudlet using just a web browser.
R9	The tools must be able to work through school firewalls.	To code the clip:bit the students were able to access Microsoft's MakeCode website. The Cloudlet is not online and did not need online access to be setup or run.

R10	The tools should work regardless of WiFi availability and/or signal strength.	The clip:bit is not an online tool. The Cloudlet did not use the school's WiFi, it created its own.
R11	Educators should be able to store the data collected digitally to access it again on a later date.	The clip:bit allows students to download their data as csv files but also for educators it allows them to upload it to the Cloudlet for collation. In this study the students didn't download the csv files but looked at their data in the Cloudlet.
R12	Educators should be able to use a semi-automated survey collection device.	Educators were able to setup the clip:bit for their data collection needs. They were able to add all the animals and plants they wanted the students to find/learn about.
R13	Educators should be able to create their own custom data collection sheets.	Using a template the educators were able to create their own custom data collection sheets with their chosen animals and plants on it.
R14	Educators and students should be able to upload data from the data collection devices to the data storage quickly and easily.	It is possible for educators to upload data from the clip:bits to the Cloudlet. In this study the researchers handled this upload. Some administration is needed first – setting up the sessions, creating the teams, before the data can be uploaded alongside the images from the data collection sheets.

R15	Educators should be able to display data on their whiteboards to use as a teaching aid.	The educators can connect their computers to the Cloudlet to view data on their whiteboards, or they can plug the Cloudlet directly into their computer via a HDMI cable. In this study the former was done in both classrooms.

Table 29 Pragmatic requirements evaluated (accessible)

ID	Pragmatic requirements (student accessibility)	Evaluation
R16	The tools should be age appropriate.	Children from age 7 to 11 were able to use the clip:bit and the Cloudlet. The younger children’s data was less reliable, but they could still do data analysis in the classroom and learn about data.
R17	It should be transparent what the tools do/are doing.	The clip:bit is labelled; however the data collection sheet does obscure the majority of these labels. The students were able to tell what the clip:bit and the Cloudlet were doing when their LEDs lit up.
R18	Like other physical computing tools made for the classroom, the tools should be sturdy, mobile, lightweight, battery powered and programmable.	The clip:bit is sturdy, after the switch was moved to the battery no clip:bits were broken in the field. The clip:bit is mobile, lightweight, battery powered and programmable. The Cloudlet is mobile, and battery powered.
R19	The tools shouldn’t have small text, screens and buttons like a mobile phone.	The only text on these tools is the 7-segment display on the clip:bit. It only displays numbers which students were able to read successfully. The buttons on the clip:bit are small but are far enough apart for students not to mistake them.
R20	The tools’ screens should be visible outdoors in daylight.	The 7-segment display is bright and visible in daylight. The clip:bit had two versions, green light and red light. The red light was easier to see outdoors.

R21	The tools should work in wet and windy weather.	The clip:bit is not waterproof. Teachers can use techniques to make it useable in wet and windy weather. In wet weather the clip:bits were put in transparent plastic bags. The students could still press the buttons and see the sheets. The data collection sheets were held down with pegs which helped stop from blowing away in windy weather.
R22	Students should be able to easily collect data using images and buttons next to each other.	The data collection sheet template aligned the images next to the buttons on the clip:bit board.
R23	Students should be able to sort and filter the data to analyse it.	The Cloudlet allows students to sort their data and filter it. They can view all the data, data for a certain item or data for a certain team. They were successfully able to use these filters to answer questions about their data.

Table 30 Pragmatic requirements evaluated (student accessibility)

ID	Engagement requirements	Derived from
R24	Students should be able to code the data collection tools to help increase their programming ability belief.	The clip:bit can be and was coded by students. The code was scaffolded to show them parts that they can successfully code and change.
R25	Support students to collect local data to support understanding global environmental issues and active participation locally and globally.	Students were able to take the clip:bits into local environments to count environmental data that matters to them. This data collection can help the children actively participate in local and global environmental issues such as national bird watches, etc.
R26	Support students to collect live and relevant data to explore trends and answer questions they are interested in.	TEDS allows children to collect data in multiple sessions. The children can view this data together to explore trends. In this study, the students collected data in multiple sessions, but the data was not related or comparable.
R27	The data the students collect should be real data, that's relevant to them.	The students collected the data using the clip:bit. It is their real live data.
R28	The data that students collect should have a purpose.	The data the students collected had a purpose that the students understood.
R29	It should be possible to link the data to other areas of the curriculum to create meaning.	It is possible to link the data collected by the students to other areas of the curriculum. Students were able to learn about habitats in Geography in these studies.

Table 31 Engaging requirements evaluated (meaningful)

ID	Educationally sound	Evaluation
R30	The tools should support contextualisation of learning/	TEDS helped students to relate what they've learnt with Computing tools to other subjects in the curriculum and real-life.
R31	The tools should create opportunities for collaborative learning by making tools shareable.	Students were able to work together using TEDS. They collected data together and analysed together.
R32	The tools should support computational thinking methods such as debugging, abstraction, logic, etc.	TEDS supported students to learn debugging, abstraction and logic when they were coding the clip:bit.
R33	The tools should allow educators to scaffold programming tasks to support novices learning to code.	The clip:bit code is complicated for this age group. It must be scaffolded by the educator for students to be able to access it.
R34	The tools should support delivery of lessons in the structure of a data science pipeline, to support progression.	TEDS supports the delivery of lessons in a data science pipeline. All year groups completed the whole pipeline.

R35	The tools should allow educators to plan for progression.	TEDS supports progression. Coding is optional. The older group coded the clip:bit while the younger group didn't have to but were still able to use it. Educators could use TEDS with many year groups. To understand data, they could view the data other students have gathered. They could use the Stickerbook for students to visualise the data instead of the graph, if the students are too young to understand graphs.
R36	The tools should support cross-curricular teaching to combine multiple subjects.	As 6.6 demonstrated, TEDS fulfils the requirements of many subjects that cover data science in the curriculum.
R37	The tools should allow project-based learning.	TEDS allows students to explore real world challenges by collecting and analysing data about the real world. Students can create graphs and visualisations as their final artefact.
R38	The tools should create opportunities for teaching about ethics including privacy and consent.	The Cloudlet being a physical device helped create opportunities for educators to discuss ethics around data.

Table 32 Educationally sound requirements evaluated.

6.7.1 Further evaluation

Each of the concepts from 4.4.1 concept definitions are evaluated above in the aims and requirement analysis. Pragmatic, engaging and educationally sound requirements were assessed through interviews and quotes by students and teachers, student questionnaires, researcher observational data and data collected from the device.

For the engagement design concepts the qualitative data is derived from student and teacher quotes and interviews, and the researcher's observational notes are looked at closer, to confirm how the tool engaged the students.

All the observation data was analysed using thematic analysis, it followed closely the Braun and Clarke thematic analysis process.

Step 1: Familiarisation with the data: Two researchers including the author became familiar with the data, reading through all the observation data and the quotes from teachers and students.

Step 2: Generating initial codes. Two researchers generated the initial codes independently on all the data. Each researcher read through all the data independent of the other person and assigned a code to each observation. The coding process was done inductively – the codes were directly derived from the data; they were not pre-determined or shared between the researchers until the end. Before step 3 all the codes were collated and analysed. Similar codes were combined, e.g. private and privacy were combined into one code: privacy. In the first round of analysis only the word "data" was coded by the author, while the second researcher had refined it further into data gathering, data science, data visualisation and data error. The author decided to read through the observations again and refine data using the codes from the second researcher.

Step 3: Searching for themes. Together the two researchers collated and analysed the 43 codes they had generated. They went through each observation and compared codes, agreeing or disagreeing on each other's code for the observation. For example, there was an observation by the author: "The teacher thought that watching 12 buttons while watching the film would be difficult, so she suggested they watch one half each. So one person was responsible for the buttons on the left, the other the right." They added the code "confidence". The second researcher did not and did not agree that the observation could be coded as confidence. They discussed it with a 3rd researcher (as an objective third-party) and the author added more context to the observation: the author had failed to note down that the teacher had said giving each child 6 buttons would set them up for success, they could easily keep track of 6 animals and that would give them confidence in successfully identifying their animals. The 3rd researcher said confidence should be included as a code, but it was also correct that the 2nd researcher had not coded it that way given the author hadn't written down the full quote correctly. Once all codes were agreed on, they were grouped under themes.

Step 4: Reviewing themes. Both reviewers discussed the resulting themes, and the themes were agreed together with the third researcher.

Step 5 defining and naming themes. Each of the 5 themes were defined and named as follows:

1. Engagement and focus. Engagement is defined as the students' attention to a particular task or activity and focus as the ability to concentrate one's attention and mental effort on that task. The following codes were grouped under the theme of engagement and focus: confidence, creativity, engaged, enjoyment, excitement, focus, frustration, happiness, ownership, real world, self-assurance, self-esteem and understanding.

2. Cross-curricular benefits: The term cross-curricular is defined as the process of teaching different subjects together. Cross-curricular benefits allow students to understand how knowledge and skills from various disciplines can be applied to real-world situations and interconnected topics. The following codes were grouped under the theme of cross-curricular benefits: averages, coding, computing, cross-curricular, drawings, internet safety and maths.

3. Purposeful education. The term purpose is defined as the reason for which something is done, created or exists. The following codes were grouped under the theme purposeful education: purpose and real-world.

4. Perceptions and experiences of computing. This term is defined as the students' perception of computing, its application and their potential future career options. The following codes were grouped under this theme: clip:bit, cloud, coding, computing, data, data science and internet safety.

5. Outdoor learning: This is defined as learning that is done away from the normal classroom environment. The following codes were grouped under this theme: outdoors and real-world.

All the quotes and their codes are included in Appendix B.

Mapping Themes

1. Engagement and focus were a key theme observed in the thematic analysis. Engagement is also a core research question. In chapter 4, it is clarified that engagement is being measured in TEDS through the concepts of real-life, meaning and purpose. As well as the practical ways TEDS work – the students can collect real-life data which has meaning, the tools and the data also have a purpose in the data lifecycle, engagement is being measured through observations and teacher and student quotes. The table below highlights specific quotes said by teachers and students that demonstrate how the students were engaged through these concepts.

Who	**Quote**	**Concept**
Year 5	The children really enjoyed it because they really felt like programmers using them and it was doing it for a specific reason rather than just using micro:bit to just put a heart on it	Real-life Purpose Meaning
Year 5	But again, they were using it for a real purpose they were going to go on and use that to collect some data later on. And I think some of them were asking about, well, how would we use this in the real world and jobs. So I think it gave it a real purpose for them.	Real-life Purpose
Year 3	You told them to go specifically look for [data]: use that drop down to find those two. How many oystercatchers had they seen? So they were then using that data to answer questions which in a maths lesson you're just using the printed data on the sheet that's been given to you. That means nothing. They don't have any questions about that.	Meaning Purpose Real-life

Table 33 Quotes from students and teachers around engagement

2. Cross-curricular benefits. Cross curricular is covered in this study as it is listed as a recognised educationally sound teaching technique. Cross-curricular is defined in section 4.4.1, it is listed as a requirement in section 4.4.2 and was evaluated in section 6.7. Discovering it as a theme in the thematic analysis reinforces this evaluation.

3. Purposeful education. Purpose is an aspect under which engagement is measured. Purpose is defined in section 4.4.1, listed as a requirement in section 4.4.2 and evaluated in section 6.7. Discovering it as a theme in the thematic analysis reinforces this evaluation.

4. Perceptions and experiences of computing. This is a new unexpected theme that was discovered through the thematic analysis. It is an interesting outcome that students think about computing as an application and career after using TEDS. This does not map directly to any research question in this thesis but would be an interesting topic for future studies.

5. Outdoor learning. A requirement for TEDS was that it could be used outdoors to collect data (section 4.4.2). The fact outdoor learning has been discovered through thematic analysis validates this assumption and shows that it is even more important than first expected.

New concepts

Through the creation and evaluation of TEDS new concepts were discovered – namely empowerment and ownership. These are described below.

Empowerment

When creating the data collection sheet it was decided with the teachers to have the ability to leave some of the boxes blank. This would allow the students to add their own data if they wanted to during the data collection stage. The students could sketch and label their data, classifying it on the data collection sheet. It would be uploaded with the other data, both the image and the numerical data. It was felt that this freedom would empower the students; it would help them have control over the data they were collecting.

During the evaluation, none of the students used the blank boxes. The Year 5 students did not have blank boxes; the teacher filled the data collection sheet with 12 items. The Year 3 students were too focused on the 10 birds on their sheet they did not have time (or pencils) to draw new birds. Therefore, empowering students by allowing them to collect their own data was not possible during this iteration of TEDS.

The data collection sheet can still be seen as empowering. As stated in 5.2.2 it does empower educators by giving them control of what data the students to collect. The teachers were able to populate both data collection sheets with the images of the items they wanted the students to collect. The students were able to use the sheets and collect the data.

Ownership

When discussing students focus in the data analysis lesson, the teachers spoke about how the students' perception of ownership of the data helped to focus them. Our requirements discussed students collecting relevant real-life data that had meaning but it did not occur to us until the teacher commented that the students would see the data as theirs, that they would own it. As stated by the teacher, the students focus increased because "*it was their data, their information, it's their names that are up there.".* And the students were excited to see their images as quoted by R2: *"When we looked at the data individually in small groups again there was lots of excitement about seeing their own images."* And a year 5 student in lesson 1: "*My sloth! That's mine, look its mine."*

Chapter 7

Conclusion

Following motivation, background, related work, case studies and introducing and evaluating TEDS: this chapter concludes the thesis. This chapter covers lessons learnt throughout this research and concludes with a discussion of future work. It will clarify it contributions and how they are valuable for others, and will answer the overarching research question from Chapter 1: Can data science learning become meaningful for primary school aged children in the context of physical computing? And the sub-research questions:

1. What range of solutions can be generated to support the pragmatic teaching of data science in schools?
2. What physical computing tools are needed to create a meaningful and engaging end-to-end experience around data science?
3. How do these physical computing tools support educationally sound teaching of data science?

7.1 Contributions

The contributions of this thesis are reiterated below and summarised in the subsequent sections.

1. An understanding of how physical computing can become engaging for primary school aged children in the context of data science learning (section 7.1.1).
2. A new approach for working with teachers in a technology design process (section 7.1.2).
3. A design process and set of requirements synthesised from research and user interactions (section 7.1.3).

Finally, the remaining additional contributions are summarised into the user groups for which they are relevant: researchers, technology developers and teachers (section 7.1.4).

7.1.1 How physical computing can become engaging for primary school aged children in the context of data science learning

This thesis showed that physical computing can be used to engage primary school children in data science learning. The approach used was to make the learning experience meaningful, have purpose and to be based in real-life. Physical computing tools were introduced specifically around these requirements. They allowed students and teachers to use physical computing to learn about data science.

The data collection device, the clip:bit, is an easy-to-use tool that students can program themselves and then take outdoors to collect data. Being able to collect real-life data in their local environments was engaging for the students. After collecting the data, it was uploaded to a local data storage device: the Cloudlet. By being able to see and analyse their data, and the data of others, in the Cloudlet students could see purpose in collecting the data, which is engaging. They also saw their own data: their names and images in the Cloudlet, this created a sense of ownership and again increased their engagement in the lessons. These experiences are highlighted through qualitative data in section 6.7.1.

It is also discussed how the teachers were able to align TEDS to the students' current learning in the curriculum, in section 6.3.1. Again, this helped keep the students engaged: the lessons with TEDS were not one-off lessons on a different topic, it was related to what they were learning in their other lessons. Their learning was embedded and aligned with TEDS, not disrupted. The curriculum analysis in section 6.6 demonstrates how TEDS can be aligned to different parts of the curriculum and the gap analysis in section 6.5 highlights how TEDS can be used to cover multiple subjects at once (gap 2 cross curricular education).

It can be therefore concluded that physical computing makes data science engaging to children when the activity is based in real life, when the learning experience has meaning and purpose. It is also particularly engaging when the learning is embedded and aligned with the current curriculum being studied by the students.

7.1.2 A new approach for working with teachers in a technology design process

This thesis introduces a new methodology for working with teachers in a technology design process: participatory education. Participatory education includes participatory design and extends it with participatory planning and delivery. Section 4.3, methodology, describes this new methodology. Section 6.1, methodology evaluated, places it in the context of the study and evaluates it.

Participatory education promotes the co-participation of the teacher, the researcher and the students from design, to planning, to delivery. The researcher leads with their specialist skills in the classroom with support and constant feedback from the teacher. This new methodology was successfully used to deliver TEDS resources. The researcher can deliver the technical content while the teacher manages the students and places the content in context for them. This novel reversing of roles was proven to be successful, as described in section 6.1.2, and could be useful for researchers in similar situations. It not only helped deliver the technical aspects of the technology, but it gave students an insight into technical careers as they could interact with the author, a STEM professional, as a teacher rather than an observer in the classroom, as analysed from teacher quotes in section 6.1.2.

This successful way of involving researchers and emerging technologies in the classroom could encourage more teachers to take part in further research studies. This methodology derisks research interventions from the teachers point of view while exposing students to cutting-edge technology and research.

7.1.3 The set of requirements synthesised from research and user interactions

This thesis collated, created and evaluated a set of requirements from research and user interactions for delivering technology solutions to schools to expand physical computing to teach data science to children. The approach used was to gather extensive background research on physical computing and data science (chapter 2 background) as well the micro:bit (section 3.1 the micro:bit) and a previous case study that used the micro:bit to collect data in schools (section 4.2 Energy in schools). More research was done in the form of a literature review on the micro:bit in education (section 3.2). This revealed gaps in the research that were absorbed into the list of requirements.

- Gap 1: Longitudinal studies and progression (R35).
- Gap 2: Cross-curricular education (R36).
- Gap 3: Data storage and analysis (R11 and R14).
- Gap 4: Age range (not specifically mentioned in the requirements table).
- Gap 5: Ethics (R38).

User interactions helped gather further requirements. These involved hands-on prototyping with designers, computer scientists and educators in sections 4.1, gathering requirements and section 5.1, hardware design.

The aim of the requirements was to answer how the top-level research question can be answered: Can data science learning be made meaningful for primary school aged children in the context of physical computing? The requirements were then broken down into the research sub-questions: pragmatic, engaging and educationally sound.

These requirements gathered can be useful for other researchers developing new technology in computer science education. Researchers and technology developers looking at extending physical computing to teach data science can re-use the requirements table in section 6.7, Aims and Requirements Analysis.

7.1.4 Contributions grouped by user group

Researchers

The Background chapter (chapter 2) is a thorough analysis of physical computing and data science in education. Researchers can use these sections to update their own knowledge in these areas. The same can be said for section 3.1 the micro:bit in the Related work chapter (chapter 3). It gives researchers a solid understanding of the micro:bit hardware, software and its use in the classroom.

The results of the related work section (section 3.2, micro:bit literature review) can also be used by researchers - the summary of papers can help researchers back up their own research and ideas around micro:bit in education. The gaps around micro:bit in education (section 3.2.6, gap analysis) can be filled by other researchers with their own work. Gaps in the literature can help researchers develop new ideas and validate the purpose of their research. The author aims to journalize these sections of the thesis to increase awareness of these contributions in the community.

Technology developers

The clip:bit and the Cloudlet are two pieces of technology evolved from feedback from the end-users: the teachers. This feedback can be valuable for technology developers in their development of tools for education. The design sessions and prototyping before the building of the device are common practice in this area, but the feedback during the evaluation could be novel to many technology developers. Using the teachers' experience of their students to gather feedback during and after every lesson was important to ensure delivery of the next lesson and the project as a whole. Examples of some of the feedback received from the teachers during the pilot study include:

- The teachers told us how younger students cannot use a clip:bit on their own as 12 items is too many for them to manage. They preferred students to share the device, with students each responsible for 6 items. This approach proved effective, as the clip:bit contains 6 items on the left and 6 items on the rights of the device, matching the working positions of two children collaborating side by side.
- If there is a delete button, the students will press it. This made gathering data very difficult and surprised us as researchers. The students were not pressing delete out of bad behaviour or even on accident. The teachers told us it was just because the button is there, and they know how to press it.
- Data such as date/time, teacher name, class name are not needed for data analysis. Keeping the data view screens on the Cloudlet simple was vital for the students to easily see their data.

These insights were gathered from the teachers' perspective and feedback. Even as researchers in child computer interaction and computer science education the research team including the author designed the clip:bit to have as many possible items as it was thought that would be more efficient. It was thought having a delete button would help the students delete their test data, and they wouldn't ever press it after recording their real data. It also assumed all data is important and should be displayed. This was an incorrect assumption.

Technology developers can also learn from the physical layout of the clip:bit, how it's laid out, how students hold it, use it and where the battery is. Students of all ages were able to successfully use and hold the clip:bit to gather data. The developers can learn from the coding blocks developed for the clip:bit, these are open source and available in MakeCode.

The gaps in the research discussed in chapter 3 (section, 3.2.6 gap analysis) could be useful areas for technology developers to focus on. For example, in research the micro:bit is used mostly in primary schools – why is this? Is the device too simple for secondary schools, is it overused in primary schools, so teachers don't want to use it in secondary schools, does it not match the secondary curriculum? Can technology developers help solve some of these problems and increase the use of micro:bit in secondary schools? Can they develop new ways of using the micro:bit in other subjects besides computing?

Educators

This thesis showed how physical computing can be expanded to teach data science, and how it can become meaningful for primary school aged children in the context of data science learning. Educators can be confident that it is possible to use physical computing to teach children about data in cross curricular lessons at primary school. It is possible to teach about data in pragmatic, engaging and educationally sound ways. Educators with little technical knowledge can be reassured that they can teach about data using physical computing, and the lessons can be engaging for the students if that data is based in real-life and has meaning and purpose to the students.

7.2 Research questions

The top-level research question was: Can data science learning be made meaningful for primary school aged children in the context of physical computing?

This thesis has shown that the micro:bit is an ideal device for teaching primary school aged children data science. It has detailed gaps in research with the micro:bit around long-term data studies. This thesis has created a new design process: participatory education to support the design, planning and delivery of new technologies with educators as equal stakeholders. From design sessions with users and case studies of using the micro:bit in schools to collect data a set of requirements were created for developing new physical computing technologies to teach primary school children data science. Finally, an evaluation was run with 100 students to test the new physical computing tools in teaching primary aged children data science. (add more reasons how its meaningful)

The research question is broken down further into sub-questions in the following sections.

7.2.1 Pragmatic teaching of data science

What range of solutions can be generated to support the pragmatic teaching of data science in schools?

The clip:bit and the cloudlet represent two points on the range of solutions for data science education. These concepts have been shown to meet the pragmatic needs of teachers. Further work could be undertaken to further explore this effective coverage of this range, but this author believe that they provide comprehensive support based on feedback from educators and children.

TEDS is **cost effective.** Each clip:bit costs £20 for the bill of materials. This cost does not include the micro:bit, however as stated previously many schools particularly in the UK have access to the micro:bit. This is well within the range of existing physical computing tools found in the classroom: £12 to £200.

The toolkit is **accessible for educators.** The clip:bit and the Cloudlet hardware and software do not need to be installed onto the school system or internet network. No administrative access is needed to set them up or use them. The Cloudlet is not online; it does not need connecting to the school's WiFi or Internet. It doesn't pull any data from unknown websites; therefore it doesn't need any special access to these websites. Both the clip:bit and the Cloudlet work independently of any other software. No software needs to be installed on the school network to use them.

The Cloudlet presents the data through a web page. Therefore, it works with multiple different types of devices like tablets, mobile phones and computers including Chromebook. All that is needed from the school is a device that can connect to a Wi-Fi network and a web browser.

TEDS is next compared to other physical computing devices from Section 2.2 Physical computing devices under the headings: sturdy, transparent, mobile, battery-powered, lightweight, child-led and programmable.

Sturdy

The clip:bit is sturdy, it has no removable parts. Nothing is connected to the clip:bit by wire. The micro:bit is slotted into the edge connector at the top of the clip:bit in a firm grip. Light banging of the clip:bit against a table does not dislodge it. Dropping the clip:bit on the floor from a height can dislodge the micro:bit. However the data is saved at every button pressed, so no data is lost, and the hardware is not broken.

Transparent

Both the clip:bit and the Cloudlet are transparent in many ways. The clip:bit's bare PCB is visible; it is not hidden behind or inside a case. All the electronic components are labelled. Its functionality is transparent at every part of it, the buttons, 7-segment displays and multicoloured LEDS, can be programmed by the students. They know how it works because they made it work.

The Cloudlet is stored in a transparent box. While the Raspberry Pi is not labelled, the multicoloured strip of LEDs around the edge of the Cloudlet box demonstrate its status and activity by changing colour. When the data moves from the clip:bit to the Cloudlet, the LEDs light up to highlight the movement of data. When the teacher enters an administrative page in the Cloudlet, the LEDs glow red to indicate a sensitive area of the tool is being accessed. Currently, viewing data in the Cloudlet has no login requirements. Children do not have to remember a password to access it. While this can be seen as a security problem, if it remains in the classroom and offline only those who can physically access it can see the data.

Mobile

A mobile data collection tool allows children to explore different areas outside the classroom. Data can be collected with the clip:bit from any location, the school playground, the school sport's field, an outdoor activity centre. Being able to travel with the clip:bit gives children and educators more opportunities to choose the data they want to record. The Cloudlet is also mobile as it can be powered by a battery and accessed through a mobile device such as a mobile phone.

Battery powered

The clip:bit is powered by AA batteries in the back of the device. A battery powered data collection device is easier to manage than one that is mains powered. Not only do batteries make the device mobile but it makes it easier to setup and use. The clip:bit does not need to be charged before it is used. Spare batteries can be easily transported as a back-up. The battery case is not sealed into the device; therefore, the batteries can be replaced quickly and easily.

Lightweight

The clip:bit weighs just 240grams with batteries. A lightweight data collection device makes it easier for younger children to carry as they walk and explore different environments. They can bring the clip:bit to the object they are looking at. It gives the children more freedom to move and collect data.

Child led

The clip:bit contains a data collection sheet. This can be pre-printed by the educator or blank. A blank sheet can be given to the children to sketch the objects they are looking for. The children can decide on the data they want to collect in the classroom and whilst in the field by drawing the objects on the data collection sheets. While the clip:bit itself can be programmed, younger children may choose not to program it. Sketching the objects onto the sheet gives children an aspect of control over the data collection. They can decide what objects to look for, what button it goes next to and what it looks like. The goal is to create a sense of **ownership** of the data through these child led activities.

The Stickerbook in the Cloudlet is a free form area for children to visualise their data. It is not a structured or generated graph. They lead in the creation of the digital twin of their data.

Programmable

The clip:bit is programmable. The students can decide how the buttons and the LEDs work. They have control over the device. Programming the micro:bit in the clip:bit requires a web browser. There is no specialist software to learn, install, download or buy to program it.

7.2.2 Meaningful and engaging end-to-end experience

What physical computing tools are needed to create a meaningful and engaging end-to-end experience around data science?

Teachers and children were engaged throughout the lessons spent using TEDS. The teachers were happy that the students were engaged and that TEDS kept them engaged. What helped engage them was the **real-life** application of TEDS. They were collecting data about local birds that they could recognise on their walks to school. It made the task and the data **meaningful** to the students.

The students were also **self-confident in their ability** to use TEDS. The outdoor trip to use the clip:bits was several weeks after they had last seen the clip:bits. However, the students were confident in using the clip:bit. They knew how it worked and were comfortable pressing the buttons to record data. There was no hesitation or uncertainty. They trusted that the clip:bit would store their data and that the Cloudlet also had a copy of their data. After coding the clip:bits and using them outdoors some of the older students discussed with their teacher about careers in Computing. The experience of coding a tool and then using that tool in a real-life situation made the students think beyond the curriculum into real life careers.

7.2.3 Educationally sound teaching

How do these physical computing tools support educationally sound teaching of data science?

It can be argued that TEDS supports teachers to deliver educationally sound lessons. Throughout the process students were able to work **collaboratively**. They worked together when drawing and labelling items in the prepare stage and whilst watching the videos indoors. The clip:bit has 12 buttons, and while teachers don't have to use all 12 buttons, they stated how 6 buttons on each side of the clip:bit was perfect for each individual child to share. Each child could see their image on their side of the board. And the teachers felt that six was the right number of items a child could search for quickly.

Outdoors the students were able to work **collaboratively** to collect the data using the clip:bits. They could use the clip:bits with binoculars and while walking along the beach searching for items. The students listened to each other and worked together when searching for the objects. In the classroom when analysing the data the students were able to successfully complete the questionnaires together.

Using TEDS educators were able to **contextualise learning.** As seen in Section 6.6 Curriculum analysis the educators were able to relate the work to multiple subjects. In this project TEDS helped relate Computing to Geography, Mathematics and A rt. It also helped the students to relate Computing to the real world as a tool and a possible career path.

TEDS supports the teaching of **computational thinking.** When coding the clip:bit the students tinkered with the different parts of it as they learnt how it works, particular the multicoloured LEDs. The students used logic to understand how to the button presses worked. The data in the Cloudlet is an abstraction of the data the students collected. Unnecessary details such as time, date and device name are abstracted away to help students focus on the data. A teacher could point out this data in the downloaded csv file in a lesson on abstraction.

In programming the clip:bit the teachers were also able to **scaffold** the programming. They shared the code with the students and taught them how to write the easier parts of the program. This helped students learn how to program logic and loops. TEDS supports **progression** by giving educators the entire program and supporting them to teach different aspects depending on the students' level of programming skill. TEDS also supports progression as the students don't have to code the clip:bit to use it. They can use the clip:bit as a tool. This was demonstrated in the two classes where Year 5 coded the clip:bit and Year 3 did not, but both were able to use it outdoors to collect data.

TEDS supports **cross-curricular** learning. Using the clip:bit to identify animals in the training video, the teachers said it helped them link the topic (the Amazon) to the technology.

T2: *"I think it worked really well. I think actually watching the video with something that they were specifically looking for made them focus on what they were watching more and it linked that topic to technology for us really."*

When questioned further: RA2: *"You think it's important to link like Geography to technology?"* they responded positively that it helps save time in the curriculum and it helps student realise that in real-life the world isn't about standalone subjects, that they overlap.

T2: "*Yeah. And I think it's important from a curriculum point of view and time wise, but also for their future to realise that it isn't, they're not stand alone subjects, they do overlap."*

7.2.4 Research questions summarised

In summary and to answer the top-level question: Can data science learning be made meaningful for primary school aged children in the context of physical computing, the conclusions of this thesis are reviewed in context.

This thesis confirms that *data science learning becomes meaningful for primary school aged children, provided that physical computing tools are used that are pragmatic, can create meaningful and engaging experiences, and that support educationally sound teaching*.

The tools need to be pragmatic for them to be used successfully in school environments with limited budgets, space, staff experience and strict protocols around installing hardware, software and accessing the Wi-Fi networks. TEDS tackled each of the pragmatic issues discussed and users were able to successfully use it in primary school classrooms.

The tools need be meaningful and engaging for the students. TEDS made it possible for students to collect and analyse data using tools that they programmed and controlled. TEDS connected the learning to real-life and created meaning. The students were engaged in the lessons as the tools were real, the code they wrote had a purpose and the data meant something to them.

Finally, the tools need to support educationally sound teaching TEDS allowed teachers to use current methods of teaching such as collaboration, contextualisation of learning, scaffolding and progression. Teachers were able to teach computational thinking skills with TEDS and cross curricular teaching was possible too.

7.3 Future work

From both the pilot and the main study several lessons were learnt from students and educators using TEDS both in the field and in the classroom. These lessons can be divided between hardware, software and education. There are also aspirational plans to develop TEDS further with new technologies described in the further activities section below.

7.3.1 Hardware improvements

The clip:bit went through two iterations during the research study. Based on feedback from the pilot study using the clip:bit to collect data with children, changes were made to the physical structure of the clip:bit. Holes were added to the bottom to attach a lanyard. This allows the children to wear the clip:bits and have their hands free to use other tools such as binoculars or magnifying glasses. In the first version of the clip:bit the power switch on the side was problematic as it regularly got snagged on clothing and snapped off. This was moved to the back of the clip:bit onto the battery.

Having two 7-segment LEDS was unnecessary by the time the main study was delivered. The left 7-segment LED was to display the section or page number the children were on. Given the difficulties they had switching between different sections or pages, this functionality was dropped from the clip:bit. With no 7-segment LED on the left, it leaves room for a redesign of the clip:bit.

The C and D buttons at the top of the clip:bit was used for changing pages and sections. When this functionality was removed, these buttons did not have any other purpose and therefore can be removed to save space in the next iteration.

The clip:bit is closer to a product than the Cloudlet. The Cloudlet was always a prototype device during this research. It was held together in a custom acrylic box made by the author. A classroom device needs to be sturdy. A separate Wi-Fi dongle was needed to allow multiple devices to connect to the Cloudlet at once.

While the imagined use of the Cloudlet is of a cloud data centre that always remains on, collecting data, it will be turned off, unplugged and moved between classes. The Cloudlet is slow to start. At times the Wi-Fi hotspot was up and running before the website which cost users time when trying to connect. This caused frustration, particularly with young children.

Uploading the data collection sheets to the Cloudlet is slow and laborious. It also leads to some data privacy issues. The teacher may be using their own mobile phone to take photos of the children's data. This needs to be avoided. The Cloudlet could have a camera and a screen so the teacher can use it to upload photos directly.

7.3.2 Software improvements

The phrasing of some of the blocks could be improved, e.g.

A version of the blocks without pages and sections would be helpful for primary educators and children, or a sub menu in the blocks called Advanced that hides these blocks could be created. This would reduce the complexity of the system. Secondary school students could still use these blocks for recording multiple pages and different sections for their data.

Different types of charts could be added to the Cloudlet, such as pictograms for younger children and line graphs for older children. Another improvement would be more control over creating the charts: setting scales, axes titles, etc. so the children can learn more from the process. Viewing multiple sets of data on the same chart over time would also be extremely useful.

Being able to show their data to visitors, parents and other classes was beneficial in the Energy in Schools project. The same could apply to this project. The Cloudlet could have a visualisation screen where teachers add data they want to show to the public.

7.3.3 Education

The teachers in the main study were able to use the clip:bit to complement their lessons. The delivery of the lessons was done by the researcher. For future work, delivery of lessons should be done by the teachers. To support this, learning materials need to be created with presentations, notes and any code files needed for the clip:bit. The lessons plans should be aligned to the curriculum so the teachers can see clearly what goals and objectives these materials cover. The plans can cover multiple year groups and across multiple subjects.

The ability to create different code files for the clip:bit would also be beneficial. At times, the teacher may want the lights on the clip:bit to be different colours, or the button presses for certain items to increase by 2 or 5, etc. The Cloudlet could provide different code files for the clip:bit, but the ability to customise these would be beneficial for educators.

7.3.4 Further activities

The clip:bit and Cloudlet were used in two primary school classes over a period of three months. The students took part in between 5 and 6 lessons. Further research could look at using TEDS in a secondary school at different age groups. The students could code the technology in text-based languages like Python or JavaScript. They could also collect more data using the micro:bits embedded sensors or attach new sensors such as soil moisture sensors or a GPS module to collect location data at the same time.

How to combine multiple data sets into the Cloudlet would need to be explored and developed further.

Other research that could add value is community-based data collection. Community groups collect data, e.g. bird watchers, societies but have problems sharing it with wider audiences. A community owned set of clip:bits and a Cloudlet displayed in a public space would allow the public to collect and analyse data that means something to them.

7.4 Concluding remarks

Working with teachers over many years has motivated the work in this thesis. Close work with schools revealed an enthusiastic response to physical computing (particularly the micro:bit) but also the challenges faced by many educators. These challenges are manageable with collaboration between researchers, educators and children. Creating new physical hardware and software to support teachers in the classroom has been an exciting endeavour for the author.

This thesis has shown that children can be excited and engaged with data science education, given the right tools and motivation.

Appendix A

1. Where is the data stored?	
2. How did it get there?	
3. Who owns the data?	
4. Who can see the data?	

Questions about the data. Use the website to answer these questions.

5. Find the data that **Liz and Lorraine** collected.

What did **Liz and Lorraine** find the most of?

6. How many ***oystercatchers*** did **Liz and Lorraine** see?

7. Find the data that **John and Isobel** collected. Who saw more ***oystercatchers***?

☐	**John and Isobel**	☐	**Liz and Lorraine**

8. What did **John and Isobel** see the fewest of?

9. Why do you think there is a difference between the number of ***oystercatchers*** **john and Isobel** saw compared to the number **Liz and Lorraine** saw? (you could have more than one reason)

10. Who saw the most ***Herring gulls***?

11. Who saw only 1 ***Lapwing***?

12. Who saw 12 ***Curlews***?

13. What is the total number of ***Lapwings*** that everyone saw?

Figure A1 Year 3 Questionnaire

Questions about the graph

Write down the name of your chosen team:

From looking at the graph

1. How many ***Cormorants*** did your team see?

2. What bird did your team see the most of?

3. What was the second highest number of birds you saw?

4. Name one bird your team didn't see.

5. What kind of graph did you use?

Any extra comments about the graph

Figure A2 Year 3 Graph Questionnaire

Appendix B

Quotation Content	Codes
15: I have that one!	clipbit data gathering excitement recognition
14 Toucan! Toucan!	data gathering excitement recognition
15 Is that a capybara? *checks their sheet	clipbit data gathering recognition
21: Can you make me one? (clip:bit) Cause I'd spend all my time watching wildlife videos and counting animals.	clipbit enjoyment
practice in a warm space where it was easy to check student's understanding.	data gathering
Teacher said previously they would research the rainforest animals etc but just to put in their notebook. This gave it a purpose.	data gathering purpose
Generally the combination of book research and tablet research to create the images worked very well	research
The teacher thought that watching 12 buttons while watching the film would be difficult so she suggested they watch one half each. So one person was responsible for the buttons on the left, the other the right.▯	confidence data gathering
Sometimes, if it was a familiar creature I delayed saying the name, waiting for a child to say "is that a toucan?" or "piranha!" etc	data gathering recognition
9) I think getting used to the clip:bits in a space where there are no distractions of being outside/out of school doing fieldwork really helped as an initial step.	clipbit data gathering
The LEDs on the clip:bit were a triumph!	clipbit coding
I said they could program them to turn on a single colour or make a rainbow and there was a ripple of excitement through the cloud at the prospect of making a rainbow.	clipbit excitement
The task was open so it allowed for the range of ability in the class	clipbit cloud progression

It allowed for creativity and experimentation, rather than just following a plan	clipbit cloud creativity
Children LOVED seeing their own images on the cloudlet. Calls of "That's my..." etc.	cloud data visualisation enjoyment ownership
Everyone though the 90-odd sloths was a human error rather than computer or Lorraine deliberately changing the data	data science trust user error
When we looked at the data individually in small groups again there was lots of excitement about seeing group images	cloud data science excitement ownership recognition
There was an interesting situation where children were reluctant to look at other groups individually	cloud data science privacy trust
One of the boys said he didn't want the others to look at his data as it was private.	cloud data science privacy trust
I know you can filter for that but generally the children were scrolling through the whole class list	cloud data science filter
But the children I worked with LOVED the visualiser. They were very excited	cloud data visualisation enjoyment
Children in my group were able to answer questions about their data using the cloudlet	cloud data science
the children in that group understood that we were seeing the same monkeys so they couldn't be added together	cloud data science maths patterns
They haven't sone averages but one child used the word 'estimate' to say they could estimate there were six monkeys because most people had recorded six monkeys.	cloud data science maths patterns
We talked bout the very high numbers. They all knew there were incorrect and hypothesised that children had forgotten to reset micro:bits after the practice	cloud data science maths trust user error

They were happy/relieved that the data could be corrected - though we didn't discuss the ethics of this.	cloud data error data science enjoyment
7 “Look, it’s a rainbow!”	coding excitement
15: Names are not boring at all. I can do it	coding confidence self assurance
18: My sloth! That’s mine, look it’s mine	cloud familiarity happiness recognition
No, that would be wrong. I think it’s 6 cause there’s more 6s (	cloud data science
More people said 6	cloud data science patterns
I said 6, I’m right	cloud self assurance
Q: Do we trust student A and student b?None of the children wanted to answer that question	cloud data science trust
StudentE said 2, student B said 1. Do you believe E more than B?Answer: Yes, cause she’s here (in front of him)	cloud data science trust
No, she’s wrong. There are more 1s	cloud data science user error
I think they were smashing the buttons	clipbit cloud data science user error
I think they didn’t delete their original data.	clipbit cloud data science user error
Student added animals to parts of the background that they felt they belonged, e.g. the monkey in the tree, the frog in the water. They enjoyed doing that	cloud data visualisation enjoyment
One student laughed at seeing their frog in the grass	cloud data visualisation enjoyment place based context

Another said the photos should be the size of the animal. The tree frog shouldn’t be the same size as the tiger.	cloud data visualisation object scale
They loved seeing their photos, browsed the data for their data and their photos. Pointed out their photos.	cloud data science data visualisation enjoyment familiarity
Being able to code the micro:bit gave a lot of them lots of self-esteem	coding self assurance self esteem
The teacher felt the students were “coding for a real purpose”.	coding purpose self assurance
The students don’t know averages but the teacher is keen to use the data when it comes to learning in maths early next year	averages maths
1 “Watch my buttons change colour.”	clipbit coding confidence
10 – completed the code then went back to the rainbow code, set themselves challenges.	challenge clipbit coding confidence
Student didn’t enjoy the Stickerbook because it was messy, the pictures had a hand within the image (so they weren’t cut out like stickers. They weren’t happy with the way they looked when placed.	cloud data visualisation patterns
It was good to see the lights lighting on the Cloudlet, helping to reinforce the connection between the chrome book and data store	cloud data visualisation lights
The children seemed to find it easy to find the data and answer questions in the simplified format	confidence data science
Most of the children I spoke to (though that was only a handful) preferred the photos in the Stickerbook for remembering and identifying (for using)	cloud drawings photos
I think it worked really well. I think actually watching the video with something that was specifically looking for made them focus on what they were watching more and it linked that topic to technology for us really.	clipbit cross curricular focus
Y5: Yeah. And I think it's important from a curriculum point of view and time wise, but also for their future to realise that it isn't, they're not stand alone subjects, they do overlap.	cross curricular real world

Y5: They've drawns things about rainforest before, but that way of actually having those images in front and actually tallying them up from watching a video. I've never done that sort of activity with them before. But I certainly think that having that something to focus on when you were watching it made them look closer.	clipbit drawings focus
And I maybe if they started off with an even simpler task of counting them, I mean they, they, they it still worked, they still understood what they're doing	clipbit data error data gathering
it was really good actually on the other side for observational skills because they were looking not just for one bird, but there was two or three within a picture	clipbit data gathering observation
I don't think so, because they won't experience anything else like that. It's really good for them to see that and you know, start to look at the idea of data and that's that same point within that lesson	cross curricular data gathering frustration
make this link between what we've just done in that Red rose maths and the graphs and the charts and the collection of information to what you are doing now.	cross curricular data gathering maths
Now, they're not used to or frequency tables, but they're not used to real data ones or if they are its favourite crisps and favourite colours.	cloud data gathering data table
I think I think with that the bird video in particular, I think there was, it was intentional that there were things that were a bit messy because I knew that the next week they'd be going outside	data gathering real world
the children really enjoyed it because they really felt like programmers using them and it was doing it for a specific reason rather than we use micro:bit we just put a heart on it	coding purpose real world
But again, they were using it for a real purpose they were gonna go on and and use that to collect some data later on so. And I think some of them were asking about, well, how how would we use this in the real world and jobs and why we to go with it. So I think it it gave it a real purpose for them.	coding purpose real world
Y5. And they they felt a sense of achievement when their code worked.	achievement coding confidence
No, I think I think it was good for them to know what the purpose of the coding was for that activity. 1st of this is what you're going to use it for and then go back and do the code and I think was the right order for them to understand why they were doing it.	coding purpose understanding
So I think the physical pressing and using it for some of them was quite difficult. But again that's cohort specific. I think some of them did find it quite fiddly, quite tricky,	clipbit physical

They were engaged. We were out there for a good while and they were engaged throughout and they were really excited to see the birds	data gathering engaged excitement
Massively. Anything interactive isn't it. Just like you with your own children, when you go to places like museums, for example, they are far more engaged in. Yeah. If you've got a clipboard you got something to go look for, a few questions. Oh they're in their element	clipbit data gathering engaged
Y5: They're in their element and they feel important	clipbit confidence data gathering self assurance
It's like you're saying. They're staying engaged. I think that makes the difference. It keeps them engaged.	clipbit data gathering engaged
Y3_TA: It's a purpose, it's.	purpose
Y5: Because they've counted more than me. So I need to go and find some more	clipbit competition data gathering
And I haven't seen it yet. I need to see it now. But you're giving them a reason to do it. You're giving them a purpose, and we try and do that all the time, don't we? We give try to give everything a purpose. That's their Purpose. That's the reason they were there	clipbit data gathering purpose
Yeah. And again, it's that then going back with with parents, grandparents and passing that information on cause they're the teachers and they know more.	clipbit generation
Y5: I think the way you look at things where you had to sort of do it on a scale of like the hands and stuff Just didn't quite work right.	clipbit data error data gathering
And maybe you come back and look at that as a progression lesson further on, but do some ratio proportions, spatial awareness things? Yeah. First.	clipbit data gathering progression
I do think it looking like a cloud: that helps	cloud physical
And it lit up every time, cause you could hear them going like, look, watch, watch and that would like when they would press a button.	cloud lights
And then to the idea of that, because it it makes my mind hurt really. Like where? Where does all this go?	cloud data storage
But yeah, the idea that it is somewhere. Yeah, stored. And it can be safe	cloud data storage safe

So cause you talked about, you know, where is the data? Who's it for? Who can see it, who can access it. And they did ask quite a few questions. Sean was quite into asking questions. You know: who else can get it? Where is it? What's it doing? You know.	cloud data storage privacy security
Yeah, which is something that we can take on when we're doing other things. And you know remind them when we do computing and other things which Involves data or storing things or Internet safety and things like that. Your phones, your pictures where they go when you upload them onto Facebook. Where are they?	computing data storage internet safety progression
Who can see them if you delete them. Are they gone forever?	cloud data life
I've done online safety, but I've had the questions the other way then or the comments, well think back to that cloud. It's not necessarily safe. People can get on it, just cause it's not on your device anymore. It could still be out there. So they've obviously have made those links.	cloud data storage internet safety
But I do think it's a hard concept for them to understand, isn't it, that these things are out there somewhere.	cloud understanding
I think that was one of their favourite lessons when they were there.	cloud enjoyment
So they were then using that data to answer questions which in a math's lesson you just using the printed data on the sheet that's been given to you. That means nothing. They don't have any questions about.	challenge engaged maths ownership understanding
I think they did really well with it and the fact that they stayed focused for the length of time they did, they wouldn't have stayed focused like that in a normal mass lesson.	cloud focus maths
Looking at data but I think because it was their data, their information, it's their names that are up there.	data ownership

References

[1] 7-segment for micro:bit: *https://monkmakes.com/mb_7_seg.html*. Accessed: 2024-11-11.

[2] Aadhaar breach report: Reactions on freedom and privacy: 2018. *https://www.csoonline.com/article/567915/aadhaar-breach-report-reactions-on-freedom-and-privacy.html*. Accessed: 2024-04-25.

[3] Abonyi-Tóth Andor and Pluhár, Z. 2019. Wandering Micro:bits in the Public Education of Hungary. *Informatics in Schools. New Ideas in School Informatics* (Cham, 2019), 189–199.

[4] Accessories micro:bit: *https://microbit.org/buy/accessories/*. Accessed: 2024-11-01.

[5] Al-Balushi, S.M. and Al-Aamri, S.S. 2014. The effect of environmental science projects on students' environmental knowledge and science attitudes. *International Research in Geographical and Environmental Education*. 23, 3 (2014), 213–227. https://doi.org/10.1080/10382046.2014.927167.

[6] Aljbour, J. et al. 2024. Powering Intelligence: Analyzing Artificial Intelligence and Data Center Energy Consumption. *EPRI White Paper no. 3002028905*. (2024).

[7] Austin, J. et al. 2020. The BBC micro:bit- from the UK to the world. *Communications of the ACM*. 63, 3 (Feb. 2020), 62–69. https://doi.org/10.1145/3368856.

[8] Austin, J. et al. 2020. The BBC micro:bit- from the UK to the world. *Communications of the ACM*. 63, 3 (Feb. 2020), 62–69. https://doi.org/10.1145/3368856.

[9] Baisa, N.L. et al. 2021. Hand-Based Person Identification using Global and Part-Aware Deep Feature Representation Learning. *2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. (2021), 1–6.

[10] Ball, T. et al. 2024. Imagining Inclusive Digital Maker Futures with the BBC micro:bit. *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, Jun. 2024), 1032–1034.

[11] Bandura, A. 1997. *Self-efficacy: The exercise of control*. Freeman.

[12] BBC micro:bit celebrates huge impact in first year: 2017. *https://www.bbc.co.uk/mediacentre/latestnews/2017/microbit-first-year*. Accessed: 2024-01-19.

[13] BBC micro:bit with Kitronik's Inventor Kit: *https://kitronik.co.uk/collections/coding/products/5618-bbc-microbit-with-inventors-kit-and-accessories*. Accessed: 2025-01-06.

[14] Beck, K. and Andres, C. 2004. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional.

[15] Bee-Bot Programmable Floor robot: *https://www.tts-group.co.uk/bee-bot-programmable-floor-robot/1015268.html*. Accessed: 2024-08-15.

[16] Bell, T. et al. 2009. Computer science unplugged: school students doing real computing without computers. *New Zealand Journal of Applied Computing and Information Technology*. 13, 1 (2009), 20–29.

[17] Ben-Ari, M. 1998. Constructivism in computer science education. *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education* (New York, NY, USA, 1998), 257–261.

[18] Berman, F. et al. 2018. Realizing the Potential of Data Science. *Commun. ACM*. 61, 4 (Mar. 2018), 67–72. https://doi.org/10.1145/3188721.

[19] Bernad, P. et al. 2021. Development of Measurement Systems with the BBC Micro:bit. *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*. (2021), 853–858. https://doi.org/10.23919/MIPRO52101.2021.9596834.

[20] Berns, R.G. and Erickson, P.M. 2001. Contextual Teaching and Learning: Preparing Students for the New Economy. The Highlight Zone: Research@ Work No. 5. (2001).

[21] Bian, L. et al. 2017. Gender stereotypes about intellectual ability emerge early and influence children's interests. *Science*. 355, 6323 (2017), 389–391.

[22] Big Butterfly Count: *https://bigbutterflycount.butterfly-conservation.org/*. Accessed: 2024-12-09.

[23] Bilstrup, K.-E.K. et al. 2022. Opportunities and Challenges of Teaching Machine Learning as a Design Material with the micro:bit. *Adjunct Proceedings of the 2022 Nordic Human-Computer Interaction Conference* (New York, NY, USA, 2022).

[24] BJP tells petrol price hike "truth" on Twitter with a graph, Congress "fixes" it for readers: *https://www.businesstoday.in/latest/economy-politics/story/bjp-tells-petrol-price-hike-truth-on-twitter-with-graph-congress-fixes-it-110700-2018-09-11*. Accessed: 2025-01-06.

[25] Blikstein, P. 2013. Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. *Proceedings of the 12th International Conference on Interaction Design and Children* (New York, NY, USA, 2013), 173–182.

[26] Blockly | Google for Developers: *https://developers.google.com/blockly*. Accessed: 2025-01-06.

[27] Bodon, H. et al. 2022. Youth Experiences with Authentically Embedded Computer Science in Sport - Proceedings of the 21st Annual ACM Interaction Design and Children Conference. *Proceedings of the 21st Annual ACM Interaction Design and Children Conference*. (2022), 504–509. https://doi.org/10.1145/3501712.3535297.

[28] Brian Hanks Sue Fitzgerald, R.M.L.M. and Zander, C. 2011. Pair programming in education: a literature review. *Computer Science Education*. 21, 2 (2011), 135–173. https://doi.org/10.1080/08993408.2011.579808.

[29] Brown, S.D. et al. 2008. Social cognitive predictors of college students' academic performance and persistence: A meta-analytic path analysis. *Journal of Vocational Behavior*. 72, 3 (2008), 298–308.

[30] Bruner, J. 1974. *Toward a theory of instruction*. Harvard university press.

[31] Buxton, C.A. 2010. Social Problem Solving Through Science: An Approach to Critical, Place-Based, Science Teaching and Learning. *Equity & Excellence in Education*. 43, 1 (2010), 120–135. https://doi.org/10.1080/10665680903408932.

[32] Cabrera, L. et al. 2019. Programs in the Palm of Your Hand: How Live Programming Shapes Children's Interactions with Physical Computing Devices. *Proceedings of the 18th ACM International Conference on Interaction Design and Children* (New York, NY, USA, 2019), 227–236.

[33] CamJam EduKit: .

[34] Celebrating 10 million micro:bits around the world: *https://microbit.org/news/2024-12-09/celebrating-10-million-microbits-around-the-world/*. Accessed: 2024-12-16.

[35] Charlton, P. and Poslad, S. 2016. A Sharable Wearable Maker Community IoT Application. *2016 12th International Conference on Intelligent Environments (IE)* (2016), 16–23.

[36] ChartJS: Open source HTML5 Charts for your website: *https://www.chartjs.org/*. Accessed: 2025-01-06.

[37] Cheryan, S. et al. 2015. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in psychology*. 6, (2015), 49.

[38] Childhood Vaccination Coverage Statistics, England, 2023-24: 2024. *https://digital.nhs.uk/data-and-information/publications/statistical/nhs-immunisation-statistics/england-2023-24*. Accessed: 2024-12-15.

[39] CODAL target for the micro:bit: *https://github.com/lancaster-university/codal-microbit-v2*. Accessed: 2025-01-06.

[40] code editor: *https://editor.raspberrypi.org/en/*. Accessed: 2024-12-16.

[41] Computing programmes of study: key stages 1 and 2: 2013. *https://assets.publishing.service.gov.uk/media/5a7c576be5274a1b00423213/PRIMARY_national_curriculum_-_Computing.pdf*. Accessed: 2025-01-06.

[42] Computing programmes of study: key stages 3 and 4: 2013. *https://assets.publishing.service.gov.uk/media/5a7cb981ed915d682236228d/SECONDARY_national_curriculum_-_Computing.pdf*. Accessed: 2025-01-06.

[43] COVID-19 Data Misrepresented by Georgia Health Department: *https://climate.law.columbia.edu/content/covid-19-data-misrepresented-georgia-health-department*. Accessed: 2025-01-06.

[44] Csizmadia, A. et al. 2015. *Computational thinking - a guide for teachers*. Computing at School.

[45] Cubetto: *https://www.primotoys.com/*. Accessed: 2024-09-26.

[46] Curriculum teaching resources: *https://teachcomputing.org/curriculum*. Accessed: 2025-01-06.

[47] Dance Move Algorithms: *https://www.barefootcomputing.org/resources/dance-move-algorithms*. Accessed: 2024-07-26.

[48] Data and Information - Data logging: *https://teachcomputing.org/curriculum/key-stage-2/data-and-information-data-logging*. Accessed: 2025-01-06.

[49] Data Science for Everyone: 2022. *https://www.datascience4everyone.org/*. Accessed: 2024-03-01.

[50] Department for Education (DfE) 2013. *National curriculum in England: secondary curriculum*.

[51] Department for Education (DoE) 2013. *National curriculum in England: primary curriculum*.

[52] Devine, J. et al. 2024. Plug-and-Play Physical Computing and Device Prototyping with Jacdac. *GetMobile: Mobile Comp. and Comm.* 28, 2 (Jul. 2024), 38–43. https://doi.org/10.1145/3686138.3686149.

[53] Devine, J. et al. 2022. Plug-and-Play Physical Computing with Jacdac. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3 (Sep. 2022). https://doi.org/10.1145/3550317.

[54] Dewey, J. 1986. Experience and education. *The educational forum* (1986), 241–252.

[55] Dias, R. et al. 2019. Teaching Computational Reasoning without a Computer. *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society* (Oct. 2019), 6795–6800.

[56] Digital Twin: *https://www.sw.siemens.com/en-US/technology/digital-twin/*. Accessed: 2024-09-24.

[57] Dillenbourg, P. 1999. *Collaborative learning: Cognitive and computational approaches. advances in learning and instruction series.* ERIC.

[58] Druga, S. and Ko, A.J. 2021. How do children's perceptions of machine intelligence change when training and coding smart programs? *Proceedings of the 20th Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2021), 49–61.

[59] Dylan, T. et al. 2021. Lanterns: Configuring a Digital Resource to Inspire Preschool Children's Free Play Outdoors - Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. (2021). https://doi.org/10.1145/3411764.3445745.

[60] Eccles, J.S. 1987. Gender roles and women's achievement-related decisions. *Psychology of women Quarterly*. 11, 2 (1987), 135–172.

[61] Eccles, J.S. 1994. Understanding women's educational and occupational choices: Applying the Eccles et al. model of achievement-related choices. *Psychology of women quarterly*. 18, 4 (1994), 585–609.

[62] Eccles, J.S. and Wigfield, A. 2020. From expectancy-value theory to situated expectancy-value theory: A developmental, social cognitive, and sociocultural perspective on motivation. *Contemporary educational psychology*. 61, (2020), 101859.

[63] Edwards, E. et al. 2021. How Can Digital Education Contribute To A Pedagogy For Environmental Care? - Proceedings of the 20th Annual ACM Interaction Design and Children Conference. *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*. (2021), 421–426. https://doi.org/10.1145/3459990.3465190.

[64] Edwards, E. et al. 2023. Supporting Fieldwork for Primary Education with Computing - Micro:Bit, Clip:Bit and Game Controllers - Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*. (2023), 553–557. https://doi.org/10.1145/3585088.3593897.

[65] ELECFREAKS microbit Robot Smart Cutebot Kit, DIY Programmable Robot Car Kit & STEM Educational Robot: *https://shop.elecfreaks.com/products/elecfreaks-micro-bit-smart-cutebot-kit-without-micro-bit-board*. Accessed: 2024-11-11.

[66] Energy in schools: *https://energyinschools.co.uk/*. Accessed: 2024-07-24.

[67] England's Initial Teacher Training Census 2017/2018: 2017. *https://www.gov.uk/government/statistics/initial-teacher-training-trainee-number-census-2017-to-2018*. Accessed: 2024-02-20.

[68] England's Initial Teacher Training Census 2018/2019: 2018. *https://www.gov.uk/government/statistics/initial-teacher-training-trainee-number-census-2018-to-2019*. Accessed: 2024-02-20.

[69] England’s Initial Teacher Training Census 2019/2020: 2019. *https://www.gov.uk/government/statistics/initial-teacher-training-trainee-number-census-2019-to-2020*. Accessed: 2024-02-20.

[70] England’s Initial Teacher Training Census 2020/2021: 2020. *https://explore-education-statistics.service.gov.uk/find-statistics/initial-teacher-training-census/2020-21*. Accessed: 2024-02-20.

[71] England’s Initial Teacher Training Census 2021/2022: 2021. *https://explore-education-statistics.service.gov.uk/find-statistics/initial-teacher-training-census/2021-22*. Accessed: 2024-02-20.

[72] England’s Initial Teacher Training Census 2022/2023: 2022. *https://explore-education-statistics.service.gov.uk/find-statistics/initial-teacher-training-census/2022-23*. Accessed: 2024-02-20.

[73] England’s Initial Teacher Training Census 2023/2024: 2023. *https://explore-education-statistics.service.gov.uk/find-statistics/initial-teacher-training-census/2023-24*. Accessed: 2024-02-20.

[74] Environment Science Expansion Board V2.0 for micro:bit: *https://www.dfrobot.com/product-2194.html*. Accessed: 2024-11-11.

[75] Equality Act 2010: 2010. *https://www.legislation.gov.uk/ukpga/2010/15/contents*. Accessed: 2024-09-25.

[76] Farrell, K. et al. 2023. A Flexible School and College Level Qualification in Data Science. *Proceedings of the 18th WiPSCE Conference on Primary and Secondary Computing Education Research* (New York, NY, USA, 2023).

[77] Fey, J. et al. 2022. Anywear Academy: A Larp-Based Camp to Inspire Computational Interest in Middle School Girls - Proceedings of the 2022 ACM Designing Interactive Systems Conference. *Proceedings of the 2022 ACM Designing Interactive Systems Conference*. (2022), 1192–1208. https://doi.org/10.1145/3532106.3533532.

[78] Fisher-Price Think & Learn Code-a-pillar: *https://service.mattel.com/us/productDetail.aspx?prodno=DKT39&siteid=27*. Accessed: 2024-09-26.

[79] Franklin, C. and Bargagliotti, A. 2020. Introducing GAISE II: A Guideline for Precollege Statistics and Data Science Education. *Harvard Data Science Review*. 2, 4 (Oct. 2020).

[80] Frauenberger, C. et al. 2015. In pursuit of rigour and accountability in participatory design. *International journal of human-computer studies*. 74, (2015), 93–106.

[81] Freitag, C. et al. 2021. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns*. 2, 9 (2021), 100340. https://doi.org/https://doi.org/10.1016/j.patter.2021.100340.

[82] Fry, K. and Makar, K. 2021. How could we teach data science in primary school? *Teaching Statistics*. 43, (2021), S173–S181.

[83] Fung, W.L. 2023. Designing STEM Education in Small Class Teaching Environments: The Hong Kong Experience. *The Asia-Pacific Education Researcher*. 32, 2 (Nov. 2023). https://doi.org/10.1007/s40299-022-00643-8.

[84] Garcia, D. et al. 2023. BJC Sparks: A New Functional-First Middle School CS Curriculum - Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1. *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. (2023), 451–457. https://doi.org/10.1145/3545945.3569842.

[85] Garcia-Ruiz, M.A. et al. 2021. Experiences in Developing and Testing BBC Micro: bit Games in a K-12 Coding Club during the COVID-19 Pandemic. *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)*. (2021), 161–164. https://doi.org/10.1109/ICISFall51598.2021.9627364.

[86] Get started with BBC micro:bit – the next gen: *https://www.bbc.co.uk/teach/microbit/articles/zv4b8p3*. Accessed: 2025-01-06.

[87] Gibson, S.D. and Bradley, P. 2017. A study of Northern Ireland Key Stage 2 pupils' perceptions of using the BBC Micro:bit in STEM education. (2017).

[88] Gravity: IoT Starter Kit for micro:bit: *https://www.dfrobot.com/product-1816.html*. Accessed: 2024-11-11.

[89] Greenfoot: *https://www.greenfoot.org/*. Accessed: 2025-01-10.

[90] Hachmann, R. 2022. The Cyber Weapon: Decomposing Puzzles in Unplugged Computational Thinking Practices with Computational Objects. *KI - Künstliche Intelligenz*. 36, 1 (Nov. 2022). https://doi.org/10.1007/s13218-022-00756-8.

[91] Hamlyn, B. et al. 2023. *Science Education Tracker Research Report: Young people's views on science education*.

[92] Hamlyn, R. et al. 2017. *Science Education Tracker Research Report: Young people's views on science education*.

[93] Hartley, K. et al. 2024. Meet MicroCode: a Live and Portable Programming Tool for the BBC micro:bit. *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2024), 355–370.

[94] Heinemann, B. et al. 2018. Drafting a Data Science Curriculum for Secondary Schools. *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (New York, NY, USA, 2018).

[95] Helle, L. et al. 2006. Project-Based Learning in Post-Secondary Education – Theory, Practice and Rubber Sling Shots. *Higher Education*. 51, (Sep. 2006), 287–314. https://doi.org/10.1007/s10734-004-6386-5.

[96] Hodges, S. et al. 2020. Physical Computing: A Key Element of Modern Computer Science Education. *Computer*. 53, 4 (2020), 20–30. https://doi.org/10.1109/MC.2019.2935058.

[97] Hodges, S. et al. 2020. Physical Computing: A Key Element of Modern Computer Science Education. *Computer*. 53, 4 (Apr. 2020), 20–30. https://doi.org/10.1109/MC.2019.2935058.

[98] Holubova, R. 2008. *Effective Teaching Methods–Project-based Learning in Physics*.

[99] Horn, M.S. et al. 2012. Tangible interaction and learning: the case for a hybrid approach. *Personal and Ubiquitous Computing*. 16, 4 (2012), 379–389. https://doi.org/10.1007/s00779-011-0404-2.

[100] How to Use the Data Science Pipeline for Data Analysis: 2024. *https://www.domo.com/glossary/what-is-the-data-science-pipeline*. Accessed: 2025-01-10.

[101] H-Unique: *https://www.lancaster.ac.uk/scc/research/h-unique/*. Accessed: 2024-04-30.

[102] Hupfeld, A. and Speed, C. 2017. Getting Something for Nothing? A User-Centric Perspective on Loyalty Card Schemes. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), 4443–4453.

[103] Impact of climate change on agriculture: *https://www.turing.ac.uk/research/research-projects/impact-climate-change-agriculture*. Accessed: 2024-08-23.

[104] Inside the Russian hack of Yahoo: How they did it: 2017. *https://www.csoonline.com/article/560623/inside-the-russian-hack-of-yahoo-how-they-did-it.html*. Accessed: 2024-04-25.

[105] Internet services : *https://teachcomputing.org/curriculum/key-stage-3/networks-from-semaphores-to-the-internet/internet-services*. Accessed: 2025-01-06.

[106] Israel-Fishelson, R. et al. 2023. Preparing K-12 Students to Meet Their Data: Analyzing the Tools and Environments Used in Introductory Data Science

Contexts. *Proceedings of the 2023 Symposium on Learning, Design and Technology* (New York, NY, USA, 2023), 29–42.

[107] Jamaludin, A.L.T.G. 2023. Psychophysiological methods to study the triggers of interest: a Singapore case study. *Current Psychology*. 42, 32 (Nov. 2023). https://doi.org/10.1007/s12144-022-03936-9.

[108] Jansen, Y. et al. 2015. Opportunities and Challenges for Data Physicalization. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), 3227–3236.

[109] Jeong, H. et al. 2022. Integrating Computer Science and Physical Education in Elementary Schools with Data Science Learning Modules Using Wearable Microcontrollers. *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)* (2022), 710–715.

[110] Jin, K.H. 2019. Modular Tasks Design for Teaching Young Students on Physical Computing Platforms - Proceedings of the 20th Annual SIG Conference on Information Technology Education. *Proceedings of the 20th Annual SIG Conference on Information Technology Education*. (2019), 106–111. https://doi.org/10.1145/3349266.3351416.

[111] Johnson, M.J. et al. 2023. Chronicles of Exploration: Examining the Materiality of Computational Artifacts - Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*. (2023), 29–47. https://doi.org/10.1145/3568813.3600132.

[112] Kearsley, G. and Shneiderman, B. 1998. Engagement Theory: A Framework for Technology-Based Teaching and Learning. *Educational Technology*. 38, 5 (1998), 20–23.

[113] Kemp, P. et al. 2024. *The future of computing education: Considerations for policy, curriculum and practice*.

[114] King, G.T.M. 2023. Socioeconomic background influences children’s attitudes and learning in creative programming workshop. *Education and Information*

Technologies. 28, 6 (Nov. 2023). https://doi.org/10.1007/s10639-022-11467-w.

[115] Kitchenham, B. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University*. 33, 2004 (2004), 1–26.

[116] Kitchin, R. 2014. *The Data Revolution: Big Data, Open Data, Data Infrastructures & Their Consequences*.

[117] KittenBot Newbit Arcade Shield: *https://www.kittenbot.cc/products/newbit-arcade-shield*. Accessed: 2024-11-11.

[118] Knowles, B. et al. 2019. A Scenario-Based Methodology for Exploring Risks: Children and Programmable IoT - Proceedings of the 2019 on Designing Interactive Systems Conference. *Proceedings of the 2019 on Designing Interactive Systems Conference*. (2019), 751–761. https://doi.org/10.1145/3322276.3322315.

[119] Knowles, B. et al. 2018. What children's imagined uses of the BBC micro:bit tells us about designing for their IoT privacy, security and safety. *Living in the Internet of Things: Cybersecurity of the IoT - 2018*. (2018), 1–6. https://doi.org/10.1049/cp.2018.0015.

[120] Kodu: *https://www.kodugamelab.com/*. Accessed: 2025-01-06.

[121] Kokotsaki, D. et al. 2016. Project-based learning: A review of the literature. *Improving schools*. 19, 3 (2016), 267–277.

[122] Kokotsaki, D. et al. 2016. Project-based learning: A review of the literature. *Improving schools*. 19, 3 (2016), 267–277.

[123] Krpan, Ž.Ž.M. 2023. Mediated transfer: impact on programming misconceptions. *Journal of Computers in Education*. 10, 1 (Nov. 2023). https://doi.org/10.1007/s40692-022-00225-z.

[124] Kumar, V. et al. 2023. PaintBall – Coding Sports Into Art for Cross-Interest Computational Connections - Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. *Proceedings of the 22nd Annual*

ACM Interaction Design and Children Conference. (2023), 713–715. https://doi.org/10.1145/3585088.3594492.

[125] Kumar, V. and Shah, D. 2004. Building and sustaining profitable customer loyalty for the 21st century. *Journal of Retailing*. 80, 4 (2004), 317–329. https://doi.org/https://doi.org/10.1016/j.jretai.2004.10.007.

[126] Kvaššayová, N. et al. 2021. The BBC micro:bit in Slovakia. *2021 International Conference on Information and Digital Technologies (IDT)*. (2021), 359–365. https://doi.org/10.1109/IDT52577.2021.9497573.

[127] Lambrichts, M. et al. 2023. CircuitGIue: A Software Configurable Converter for Interconnecting Multiple Heterogeneous Electronic Components. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2 (Jun. 2023). https://doi.org/10.1145/3596265.

[128] Layne, A. et al. 2020. Cultivating Positive ICT Perceptions: an application of the MST-tree model to the 'Guyanese Girls Code' Initiative. *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. (2020), 934–940. https://doi.org/10.1109/CSCI51800.2020.00174.

[129] Lego Mindstorms - Invent a Robot: *https://www.lego.com/en-gb/themes/mindstorms*. Accessed: 2025-01-06.

[130] Lemonica Rosa, R. 2021. Social Necklace Project: A Student-Made Gadget to Help People during the COVID19 Pandemic - FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education. *FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education*. (2021). https://doi.org/10.1145/3466725.3466737.

[131] Lent, R.W. et al. 1994. Toward a unifying social cognitive theory of career and academic interest, choice, and performance. *Journal of vocational behavior*. 45, 1 (1994), 79–122.

[132] Lent, R.W. and Brown, S.D. 2019. Social cognitive career theory at 25: Empirical status of the interest, choice, and performance models. *Journal of*

Vocational Behavior. 115, (2019), 103316. https://doi.org/https://doi.org/10.1016/j.jvb.2019.06.004.

[133] Littebits: *https://classroom.littlebits.com/welcome*. Accessed: 2024-09-26.

[134] Liu, J. et al. 2022. A Study of Micro:bit Teaching Design Based on the Development of Computational Thinking of Secondary School Students. *2022 4th International Conference on Computer Science and Technologies in Education (CSTE)* (2022), 198–201.

[135] Liu, J. et al. 2022. A Study of Micro:bit Teaching Design Based on the Development of Computational Thinking of Secondary School Students. *2022 4th International Conference on Computer Science and Technologies in Education (CSTE)* (2022), 198–201.

[136] Marshall, P. 2007. Do tangible interfaces enhance learning? *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (New York, NY, USA, 2007), 163–170.

[137] Martinez, W. and LaLonde, D. 2020. Data Science for Everyone Starts in Kindergarten: Strategies and Initiatives From the American Statistical Association. *Harvard Data Science Review*. 2, 3 (Sep. 2020).

[138] Master, A. et al. 2021. Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the National Academy of Sciences*. 118, 48 (2021), e2100030118. https://doi.org/10.1073/pnas.2100030118.

[139] Master, A.H. and Meltzoff, A.N. 2020. Cultural stereotypes and sense of belonging contribute to gender gaps in STEM. *Grantee Submission*. 12, 1 (2020), 152–198.

[140] Merkouris, A. et al. 2017. Teaching Programming in Secondary Education Through Embodied Computing Platforms: Robotics and Wearables. *ACM Trans. Comput. Educ.* 17, 2 (May 2017). https://doi.org/10.1145/3025013.

[141] micro:bit and AI: *https://microbit.org/ai/*. Accessed: 2024-12-16.

[142] Microbit Educational Foundation report on IRIM, the Institute for Youth Development and Innovation in Croatia: *https://microbit.org/impact/foundation-reports/croatia/irim-croatia/*. Accessed: 2024-01-19.

[143] Microbit Educational Foundation report on Ultra:Bit, a national micro:bit programme in Denmark: 2024. *https://microbit.org/impact/foundation-reports/denmark/ultrabit/*. Accessed: 2024-01-19.

[144] microbit lessons: *https://microbit.org/teach/lessons/*. Accessed: 2024-11-02.

[145] micro:bit Python Editor: *https://Python.microbit.org/v/3*. Accessed: 2025-01-06.

[146] micro:bit Python Editor: *https://python.microbit.org/v/3*. Accessed: 2024-12-16.

[147] MicroBlocks: *https://microblocks.fun/*. Accessed: 2025-01-06.

[148] MicroCode: *https://microsoft.github.io/microcode/*. Accessed: 2025-01-06.

[149] Microsoft MakeCode for micro:bit: *https://makecode.microbit.org/*. Accessed: 2025-01-06.

[150] MI:pro Protector Case for the BBC micro:bit: *https://kitronik.co.uk/products/mipro-protector-case-for-the-bbc-microbit*. Accessed: 2024-11-11.

[151] MMR catch-up campaign significantly boosts uptake across country: 2024. *https://www.england.nhs.uk/2024/04/mmr-catch-up-campaign-significantly-boosts-uptake-across-country/*. Accessed: 2024-12-15.

[152] Moon, P.F. et al. 2023. The Tools Being Used to Introduce Youth to Data Science. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2023), 150–159.

[153] Nagai, T. et al. 2023. PICA-PICA: Exploring a Customisable Smart STEAM Educational Approach via a Smooth Combination of Programming, Engineering and Art. *2023 IEEE Global Engineering Education Conference*

(EDUCON). (2023), 1–6. https://doi.org/10.1109/EDUCON54358.2023.10125184.

[154] National curriculum in England: computing programmes of study: *https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study*. Accessed: 2024-09-26.

[155] National Plant Monitoring Scheme: *https://www.npms.org.uk/*. Accessed: 2024-08-23.

[156] Nikou, S.A. 2023. Student motivation and engagement in maker activities under the lens of the Activity Theory: a case study in a primary school. *Journal of Computers in Education*. (Nov. 2023). https://doi.org/10.1007/s40692-023-00258-y.

[157] Nolan, K. et al. 2019. CSinc: An Inclusive K-12 Outreach Model - Proceedings of the 19th Koli Calling International Conference on Computing Education Research. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*. (2019). https://doi.org/10.1145/3364510.3366156.

[158] OECD 2019. *An OECD Learning Framework 2030*. Springer.

[159] On street illegal parking detection and warning using AI cam: 2023. *https://www.hackster.io/siwachanatSW/on-street-illegal-parking-detection-and-warning-using-ai-cam-63f536?f=1*. Accessed: 2024-07-31.

[160] Ouyang, L. et al. 2021. Designing Tangible Interactions with Children for Pre-School Music Education - Proceedings of the 32nd Australian Conference on Human-Computer Interaction. *Proceedings of the 32nd Australian Conference on Human-Computer Interaction*. (2021), 687–691. https://doi.org/10.1145/3441000.3441023.

[161] Overview Circuit Playground Express: *https://learn.adafruit.com/adafruit-circuit-playground-express/overview*. Accessed: 2025-01-06.

[162] Papert, S. and Harel, I. 1991. Situating constructionism. *constructionism*. 36, 2 (1991), 1–11.

[163] Ph.D., W.P.K. 2023. Bringing computational thinking to technology education classrooms: Hacking car activity for middle schools in the republic of Korea. *International Journal of Technology and Design Education*. 33, 3 (Nov. 2023). https://doi.org/10.1007/s10798-022-09750-5.

[164] Piaget, J. 2005. *The psychology of intelligence*. Routledge.

[165] pin:bit: *https://shop.pimoroni.com/products/pin-bit*. Accessed: 2024-11-11.

[166] Pittard, V. 2018. The integration of data science in the primary and secondary curriculum. *Final Report: To the Royal Society Advisory Committee on Mathematics Education*. (2018).

[167] Przybylla, M. and Romeike, R. 2018. Empowering learners with tools in CS education: Physical computing in secondary schools. *it - Information Technology*. 60, 2 (2018), 91–101. https://doi.org/https://doi.org/10.1515/itit-2017-0032.

[168] Pupil premium: overview: *https://www.gov.uk/government/publications/pupil-premium/pupil-premium*. Accessed: 2024-01-22.

[169] Pyzybylla, M. and Romeike, R. 2014. Physical Computing and its Scope – Towards a Constructionist Computer Science Curriculum with Physical Computing. *Informatics in Education*. 13, 2 (2014), 241–254.

[170] Report Asian hornet sightings: *https://www.ceh.ac.uk/our-science/citizen-science/asian-hornet*. Accessed: 2024-08-23.

[171] Research Ethics: *https://www.lancaster.ac.uk/sci-tech/research/ethics/*. Accessed: 2025-01-10.

[172] Resnick, M. 2017. *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. Mit Press.

[173] Right to erasure : *https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/individual-rights/individual-rights/right-to-erasure/#:~:text=Under%20Article%2017%20of%20the,be%20created%20in%20the%20future.* Accessed: 2024-07-31.

[174] Roumelioti, R.G.M.M.R. 2021. Reflection and awareness in the design process: children ideating, programming and prototyping smart objects. *Multimedia Tools and Applications*. 80, 26 (Nov. 2021). https://doi.org/10.1007/s11042-020-09927-x.

[175] Royal Society 2017. After the reboot: Computing education in UK schools. *Policy Report*. (2017).

[176] Royal Society 2012. *Shut down or restart? The way forward for computing in UK schools*. Royal Society.

[177] Rubin, A. 2020. Learning to Reason with Data: How Did We Get Here and What Do We Know? *Journal of the Learning Sciences*. 29, 1 (2020), 154–164. https://doi.org/10.1080/10508406.2019.1705665.

[178] Savage, J. 2012. Moving beyond subject boundaries: Four case studies of cross-curricular pedagogy in secondary schools. *International Journal of Educational Research*. 55, (2012), 79–88. https://doi.org/https://doi.org/10.1016/j.ijer.2012.07.004.

[179] Schaper, M.-M. and Ruiz Garcia, A. 2023. Three Recommendations to Engage At-Risk Students in Critical Reflection on Intelligent Technologies through Remote Learning - Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. (2023). https://doi.org/10.1145/3544549.3573862.

[180] Schmidt, A. 2016. Increasing Computer Literacy with the BBC micro:bit. *IEEE Pervasive Computing*. 15, 2 (Apr. 2016), 5–7. https://doi.org/10.1109/MPRV.2016.23.

[181] Schöttelndreier, U.T.B.K.S. 2023. “Programming Is Not That Hard!” When a Science Center Visit Increases Young Women’s Programming Ability Beliefs. *Journal for STEM Education Research*. 6, 2 (Nov. 2023). https://doi.org/10.1007/s41979-023-00094-w.

[182] Schuler, Douglas. 1993. Participatory Design : Principles and Practices. Taylor & Francis Group.

[183] Scratch - imagine, program, share: *https://scratch.mit.edu/*. Accessed: 2025-01-06.

[184] ScratchJr: *https://www.scratchjr.org/*. Accessed: 2025-01-06.

[185] Sentance, S. et al. 2017. “Creating Cool Stuff”: Pupils’ Experience of the BBC Micro:Bit - Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. (2017), 531–536. https://doi.org/10.1145/3017680.3017749.

[186] Sentance, S. et al. 2017. Teaching with physical computing devices: the BBC micro:bit initiative. *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*. Association for Computing Machinery.

[187] Sentance, S. and Csizmadia, A. 2017. Computing in the curriculum: Challenges and strategies from a teacher’s perspective. *Education and Information Technologies*. 22, 2 (2017), 469–495. https://doi.org/10.1007/s10639-016-9482-0.

[188] Sentance, S. and Csizmadia, A. 2015. Teachers’ perspectives on successful strategies for teaching Computing in school. (Sep. 2015).

[189] Seraj, M. et al. 2020. Impacts of Creating Smart Everyday Objects on Young Female Students’ Programming Skills and Attitudes. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education* (New York, NY, USA, 2020), 1234–1240.

[190] Shapiro, L.F.MorenoR.B. 2020. Teacher and student enactments of a transdisciplinary art-science-computing unit. *Instructional Science*. 48, 5 (Nov. 2020). https://doi.org/10.1007/s11251-020-09518-1.

[191] Simonofski, A. et al. 2019. Engaging Children in the Smart City: A Participatory Design Workshop. *Proceedings of the 1st ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence*. (2019), 1–4. https://doi.org/10.1145/3340435.3342719.

[192] Simonsen, J. and Robertson, T. 2013. *Routledge international handbook of participatory design*. Routledge New York.

[193] 'Smart Handpumps' to bring a reliable water service to rural Africa: 2015. *https://www.ox.ac.uk/news/2015-07-02-%E2%80%98smart-handpumps%E2%80%99-bring-reliable-water-service-rural-africa*. Accessed: 2024-12-16.

[194] Soro, A. et al. 2020. Coding on Country - Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. (2020), 1–8. https://doi.org/10.1145/3334480.3382992.

[195] Stamatina Anastopoulou Mike Sharples, S.A.C.C.C.O. and Wright, M. 2012. Creating Personal Meaning through Technology-Supported Science Inquiry Learning across Formal and Informal Settings. *International Journal of Science Education*. 34, 2 (2012), 251–273. https://doi.org/10.1080/09500693.2011.569958.

[196] Syu, S.-Y.L.-C.L.-Y. 2022. Project-based learning oriented STEAM: the case of micro–bit paper-cutting lamp. *International Journal of Technology and Design Education*. 32, 5 (Nov. 2022). https://doi.org/10.1007/s10798-021-09714-1.

[197] Tamashiro, M.A. et al. 2023. Teaching Technical and Societal Aspects of IoT - A Case Study Using the Orbit IoT Kit - Proceedings of the 2023 ACM Designing Interactive Systems Conference. *Proceedings of the 2023 ACM Designing*

Interactive Systems Conference. (2023), 1236–1247. https://doi.org/10.1145/3563657.3596092.

[198] Tamborg, A.L. et al. 2022. Programming and Computational Thinking in Mathematics Education. *KI - Künstliche Intelligenz*. 36, 1 (2022), 73–81. https://doi.org/10.1007/s13218-021-00753-3.

[199] Teaching online safety in schools: 2023. *https://www.gov.uk/government/publications/teaching-online-safety-in-schools/teaching-online-safety-in-schools*. Accessed: 2024-07-24.

[200] Tech designed to inspire access to clean water for all: *https://microbit.org/news/2021-10-11/tech-designed-to-inspire-access-to-clean-water-for-all/ A*. Accessed: 2024-07-31.

[201] The web framework for perfectionists with deadlines: *https://www.djangoproject.com/*. Accessed: 2025-01-06.

[202] Theobald, E.J. et al. 2015. Global change and local solutions: Tapping the unrealized potential of citizen science for biodiversity research. *Biological Conservation*. 181, (2015), 236–244.

[203] thonny: *https://thonny.org/*. Accessed: 2024-12-16.

[204] Tikva, C. and Tambouris, E. 2023. The effect of scaffolding programming games and attitudes towards programming on the development of Computational Thinking. *Education and Information Technologies*. 28, 6 (2023), 6845–6867. https://doi.org/10.1007/s10639-022-11465-y.

[205] Timmerman, V. 2017. Cross-curricular teaching: How does it work. *URL: http://www. ciep. fr/sites/default/files/atoms/files/crosscut-cross-curricular-teaching-how-does-it-work-synthesis. pdf (Last accessed: 12.03. 2019)*. (2017).

[206] Tseng, T. et al. 2021. PlushPal: Storytelling with Interactive Plush Toys and Machine Learning - Proceedings of the 20th Annual ACM Interaction Design and Children Conference. *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*. (2021), 236–245. https://doi.org/10.1145/3459990.3460694.

[207] Tuia, D. et al. 2022. Perspectives in machine learning for wildlife conservation. *Nature Communications*. 13, 1 (2022), 792. https://doi.org/10.1038/s41467-022-27980-y.

[208] Tyrén, M. et al. 2018. Considerations and Technical Pitfalls for Teaching Computational Thinking with BBC Micro:Bit - Proceedings of the Conference on Creativity and Making in Education. *Proceedings of the Conference on Creativity and Making in Education*. (2018), 81–86. https://doi.org/10.1145/3213818.3213829.

[209] Udvikling af kunstig intelligens og maskinlæring i lærerfaglig teknologiforståelse (Development of artificial intelligence and machine learning in teacher technology understanding): *https://tekforstaa.dk/arbejdspakke/udvikling-af-kunstig-intelligens-og-maskinlaering-i-laererfaglig-teknologiforstaaelse/*. Accessed: 2024-11-04.

[210] UK Council for Internet Safety 2020. *Education for a Connected World*.

[211] UK Department of Education 2023. *The pupil premium (England)*.

[212] Underwood, L. et al. 2022. Energy in Schools: Empowering Children to Deliver Behavioural Change for Sustainability. *Proceedings of the 21st Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2022), 308–314.

[213] Underwood, L. et al. 2023. Introducing Classroom Cloudlet: A Mobile, Tangible, and Transparent Approach to Internet of Things Education. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference* (New York, NY, USA, 2023), 740–744.

[214] Underwood, L. et al. 2024. MicroCode: live, portable programming for children via robotics. *Adjunct Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2024).

[215] Underwood, L. et al. 2024. Tangible tools for data science education. *Proceedings of the 19th WiPSCE Conference on Primary and Secondary Computing Education Research* (New York, NY, USA, 2024).

[216] Underwood, L. and Finney, J. 2022. Introducing the Intranet of Things for Education. *6th FabLearn Europe / MakeEd Conference 2022* (New York, NY, USA, 2022).

[217] Venn-Wycherley, M. et al. 2020. Design Studios for K-12 Computing Education. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. (2020), 1227–1233. https://doi.org/10.1145/3328778.3366923.

[218] Voštinár, P. and Knežník, J. 2020. Using BBC micro:bit in primary and secondary schools for creating simple smart home. *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. (2020), 648–652. https://doi.org/10.23919/MIPRO48935.2020.9245314.

[219] Welcome to the UK Pollinator Monitoring Scheme (PoMS): *https://ukpoms.org.uk/*. Accessed: 2024-08-23.

[220] Wilson, M. 2009. Measuring progressions: Assessment structures underlying a learning progression. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*. 46, 6 (2009), 716–730.

[221] Wing, J.M. 2006. Computational thinking. *Communications of the ACM*. 49, 3 (Mar. 2006), 33–35. https://doi.org/10.1145/1118178.1118215.

[222] Wolff, A. et al. 2016. Urban Data in the primary classroom: bringing data literacy to the UK curriculum. *The Journal of Community Informatics*. 12, (Jan. 2016). https://doi.org/10.15353/joci.v12i3.3278.

[223] Woongbin Park and Hyuksoo Kwon 2023. Bringing computational thinking to technology education classrooms: Hacking car activity for middle schools in the republic of Korea. *International Journal of Technology and Design Education*. 33, 3 (Nov. 2023). https://doi.org/10.1007/s10798-022-09750-5.

[224] Worsley, M. 2022. PE++: Exploring Opportunities for Connecting Computer Science and Physical Education in Elementary School - Proceedings of the 21st Annual ACM Interaction Design and Children Conference. *Proceedings of*

the 21st Annual ACM Interaction Design and Children Conference. (2022), 590–595. https://doi.org/10.1145/3501712.3535293.

[225] Wurdinger, S. et al. 2007. A qualitative study using project-based learning in a mainstream middle school. *Improving Schools*. 10, 2 (2007), 150–161. https://doi.org/10.1177/1365480207078048.

[226] Yahboom Micro:bit piano expansion board: *https://category.yahboom.net/products/pianoboard*. Accessed: 2024-11-11.

[227] Yahboom Wrist:bit wearable watch kit based: *https://category.yahboom.net/products/wristbit*. Accessed: 2024-11-11.

[228] Yu, J. et al. 2023. Exploring Computational Thinking with Physical Play through Design - Proceedings of the 22nd Annual ACM Interaction Design and Children Conference. *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*. (2023), 124–136. https://doi.org/10.1145/3585088.3589365.