A planned collision of three cultures to revalue science - learners, teachers and makers

Abstract

This paper reports on a series of facilitated software and electronics experiences for a non-specialist audience, collectively known as Shrimping. These experiences are supported by community activities, collectivised resources and written material intended to catalyse and guide an audience of young people and adults in a deprived area to explore and selectively adopt practices characteristic of “Maker Culture”.

We present evidence that the established pathways to Science, Technology, Engineering and Mathematics (STEM) approaches and knowledge are not appropriate for many in our target audience.

To address these shortcomings, the project introduces a canonical design for a prototyping microcontroller board called the Shrimp (ShrimpingIt 2012), drawing upon pre-existing practice in the maker community (Icecreamterror 2009, Boak 2010, Oomlout 2012). We describe how we use the Shrimp as a foundation for accessible interaction design activities. These offer an alternative pathway to engagement and revaluation of STEM which is particularly suitable for those in the deprived community of Morecambe where we live and work.

Central to the programme are concepts such as *authenticity* from constructivist learning and *positive deviance* from studies of regeneration. We steer our activities to recontextualise science for the individual, emphasising congruence with the participants’ identity, interests, economic and social context, and community self-sufficiency.

Shrimping aims to stimulate and promote individual contribution, local innovation and collaborative transaction within the community as a means of overcoming real-world technology access and engineering challenges.

Introduction

In the developed world, Science, Technology, Engineering and Mathematics (STEM) educators face a demographic anomaly. The new generation intimately value and recognise the importance of STEM endeavours (Schreiner and Sjøberg 2007). They are fascinated by the science in the media, and the technology in their pocket (Jenkins and Pell 2006, IET 2012a). They are confident of their personal ability to perform well in science, despite recognising that it may be more challenging than other subjects (Schreiner and Sjøberg 2007). However, the majority of students, even those who do well in STEM subjects, drop them as soon in their school career as they possibly can, and would never consider them as the foundation of a career (Schreiner and Sjøberg 2007).

What’s more, the trend is worsening. Japan, a high technology consumerist society (Kharaz and Gertz 2012) leads the table, with students displaying the least interest of all the 40 countries studied by the Relevance of Science Education (ROSE) project (Schreiner and Sjøberg 2007).

In their long-term study, Schreiner and Sjøberg found that in “modern societies neither scientists nor engineers are heroes or attractive role models for the young generation” (Schreiner and Sjøberg 2005) and that in “wealthy countries, nearly no girls want to work with technology, and even boys are ambivalent” (Schreiner and Sjøberg 2010).

A recent Geology examination paper in the UK asks students to “explain why opencast coal mining is cheaper than underground mining” (OCR 2011). On the surface, this is a simple enough question. The beneficial economics of mining shallow deposits which don’t require intense, long-term investment for unknown gains are self-evident.

A similar question should be presented to STEM educators. Where should you encourage science learners to ‘start digging’ in order to get the most for their efforts, taking account of their *personal interest, engagement and satisfaction*, as well as their accumulation of *scientific practice and knowledge*.

Our contention is mainstream science educators are asking students to start digging in the wrong place, and using the wrong tools.

Braund and Reiss (2006) seek to explain why pupils’ “attitudes to school science decline progressively across the age range of secondary schooling” and why “declining numbers of students are choosing to study science at higher levels and as a career”. They observe that school science courses...

> “have traditionally been constructed from a scientist‟s viewpoint with the concepts being developed in a way that is seen to be sensible by a scientist. Typically this means that pre-eminence is given to scientific concepts [but] students...want teachers to show them why the concepts are important.” (Braund and Reiss 2006)

Teaching science in a way which is “decontextualised” (IET 2012a) relies on student’s expectation of deferred gratification - that one day the effort they are investing will pay dividends in some life context which is relevant to them. Few students seem to have this confidence. This causes those who are not already inherently interested in science subjects to disengage.

To counter this trend we agree with Braund and Reiss’ (2006) suggestion to make “context the driving force” of learning. Encountering science in a meaningful, personally motivating context is arguably the missing piece in traditional education.

We believe that the DIY culture of *Makers* (Silver 2009, Kutsnetzov et al 2010, Wang et al. 2011) can provide an environment for students to engage in technology activities which are inherently interesting and motivating, but where you only have to scratch the surface to encounter a rich seam of STEM techniques and knowledge.

Shrimping activities revolve around a central innovation; a canonical design for an incredibly low-cost microcontroller board, which can be built by participants within minutes, but can be demonstrated to act as the foundation of many devices. We believe that exploring this design space will engage and empower those who feel disenfranchised in the face of conventional science education.

In this paper we summarise the evidence which backs up our position, and outline the rationale for an alternative programme of community activities built around *Maker culture*, tuned for those who are ill-served by the current pathways to STEM learning. We summarise the design brief which we set ourselves, the strategies and assets which we have brought together to offer an alternative pathway. Through this work we hope to broaden the appeal of science.

Student Culture

When students articulate their disinterest and disappointment with STEM subjects, they express a longing for exactly those aspects which are strengths and defining characteristics of Maker culture.

Schreiner and Sjoberg (2005) believe this to be an issue of self-identity, finding that students “reject the subject...when mathematics is no longer compulsory...because they do not want to "belong" to the mathematics culture, or to carry the identity of a mathematician”.

In a survey exploring barriers to STEM uptake they found that “young people, especially girls, although they appreciate the technology, would rather like to have an identity that conveys late-modern values. Such values might be self-realisation, creativity and innovation, working with people and helping others” (Schreiner and Sjoberg 2007).

These sentiments are strongly concordant with a recent, reflective account of participation in the Maker movement...

> “Makers find resonance with materials and people. They play with their palm fronds, circuit pathways, and code while keeping a theme in mind to find a strong overlap and resonance between the goal and the materials. They make their creations ultimately for humans first and efficiency after, which is why they watch how people resonate with their creations, discussing recently catalyzing events contagiously and sharing immediately, even/especially with kinks still in the design.” (Silver 2009)

The value of *information technology* as a route to introducing STEM subjects is underlined by the results student surveys, with the Institution of Engineering and Technology observing that students “do not see ICT as being naturally linked to science. The imagery of ICT is very different from that of science, i.e. fast changing, contemporary, young and ‘sexy’”, and suggesting that science teaching “could well borrow some of this imagery in order to re-brand itself” (IET 2012a)

Electronics seems to resonate particularly well, with UK ‘Key Stage 3’ pupils seeing electronics as “more enjoyable than D&T in general” and both boys and girls “link[ing] enjoyment of the subject to the '*making*' techniques involved, learning new things and the *authenticity* of the tasks.” (Lunn 2012), our italics.

There is a clear link here with the constructivist concept of *authenticity* (Maor 1999) (Cey 2001), through which “students can see how [a learning task] relates to something of significance in their world...help[ing] to maintain enthusiasm” (Lister and Leaney 2003). As you will see later, real-world deployment, led by the students themselves is central to *Shrimping* projects.

An explanation for electronics and ICT bucking the trend is suggested by Ogawa and Shimode’s study of Japanese students. They found that girls who described science as a low priority subject for them, “show distinct interest to the mechanism of how mobile phones can send and receive messages”, identifying these as “scientific topics relevant to their daily ways of life”, and suggesting that “ ‘mobile phone system’ can serve as a ‘trigger’ or ‘breakthrough’” to engage girls in science learning. (Ogawa and Shimode 2004)

Teacher Culture

Unfortunately, schools in general pursue just the opposite approach. In their report “Studying STEM, What are the Barriers?”, the Institute of Engineering and Technology shares qualitative studies where secondary school pupils describe themselves as “relatively passive recipients in the knowledge transmission process with less and less time devoted to practical work”, the curriculum being described as “irrelevant to life” with subject matter which is “boring and over-prescriptive” (IET2012a)

Of course, there are some examples of successfully integrating physical computing and prototyping in schools, with positive evidence of success. Slovene and Slovakian schools seem to have been engaged on this course for some time (Kocijancic, 2001, Balogh 2010).

One UK ‘Electronics in Schools’ initiative seems to have been inspiring. In the words of one teacher; “it captures a lot of children. It’s that thing where they actually get something and it works. You can see it in their faces when they've soldered together a circuit, particularly the children that thought they weren't capable of doing it” (DigitalDandT 2012).

However, even where students are invited to engage with science within a design context, they can be prevented from developing their own skills of science investigation by a focus on formal outcomes. Lunn et al (2012) warn that teachers seeking a “working product” from a design activity are likely to...

- constrain students’ scope for making design decisions
- devise tasks which are...guaranteed to work, but...lack authenticity
- exercise a high degree of control over pupils' work to minimise problems arising, and when they do arise to identify and solve them for pupils
- insist on pupils working alone, so that their work can be assessed

> As a consequence, pupils “experience disengagement and boredom, find it difficult, and learn little of value” (Lunn et al 2012)

As contributors outside the school system, we have the advantage of being beyond the reach of targets and assessments. We can assist learners to undertake real world experiments which are destined to fail, but which give them valuable experience. Collaboration and peer learning can be actively encouraged. Wild ideas can be pursued to their logical conclusion.

An effective summary of our approach is the formula recommended by the IET to improve the uptake of post-16 STEM subjects ...

> dynamic, futuristic-orientated outside-school-science approaches that bring attention to and highlight the positive values, aspirations and opportunities available in the world of science, technology, engineering and mathematics. (IET 2012a)

Maker Culture

Members of the Maker community are exposed to issues of materials science, algebra, logic and myriad other subjects every day as part of their inventive leisure activities (Wang and Kaye 2011). Questions such as “What is the capacitance of the human body?” (Capsense

2012), “How can we predict the trajectory of a paper ball?” (Wired 2012), “How much power does an LED use?” (Hari 2010), “What’s the maximum diameter of a Honey Bee?” (Hydronics 2012) and “What are the constituent gases of human flatulence?” (Randofo 2009) are purely practical aspects of a self-actualising project which by definition suits the engineer’s tastes and interests.

Are Makers interested in STEM subjects for their own sake? Sometimes. However that’s a long way from the focus of their efforts. They are interested in solving a real-world problem which they have set themselves. Along the way, they are compelled to engage with science.

Makers are engaged in the practice of “science as a way of knowing”, and *making* embodies an “interaction between science and society”; aspects which Fensham characterises as “casualties of the unfinished debate about scientific literacy” which took place when developing the UK National Curriculum, a development which unfortunately influenced education policy around the world. He mourns “the establishment of a new subject area called Technology” and suggests that its knock on effects “destroyed its role as the bridge to more relevant content for school science” (Fensham 2004).

From a pedagogical point of view, we build upon concepts of constructivist learning (Nwana 1997) (Boyle, 2000) (Denis and Hubert 2001) (Cey, 2001) (Camarata et al 2003) (Lister and Leaney 2003), (Barak 2005), which emphasise the learner’s exploratory, active synthesis of knowledge.

Shrimping activities take this further, and more literally, towards Steeg and Ling’s (2006) concept of “constructi*onism*” (as opposed to constructi*vist* learning). In their model, actually building artifacts is a key method of synthesising knowledge...

> “The core argument of constructionism is that people learn best when they are making something (be it a sandcastle on the beach or a theory in physics) because of the powerful interaction between thinking and action during construction. Learning is most powerful when the construction environment is rich and there is ample opportunity to view the success of one’s construction efforts (feedback).”
>
> (Steeg and Ling 2006)

The touchpoints of Maker culture project strongly onto the debate of science as a top-down teaching institution. Community members gain reputation through demonstrating “skill”, and demonstrating “sharing norms” with a key form of participation being “learning through teaching” (Wang and Kaye 2011).

With aspects of “resistance” “repurposing and challenging authority” and “community participation” central to the culture (Wang and Kaye 2011) it’s no surprise that makers have developed their own learning resources and approaches independently of formal institutions. We turn to a canonical example; the Arduino platform.

Arduino: Designing for Designers

Massimo Banzi is a member of the team which created the Arduino microcontroller board as a tool for prototyping new ‘products’ and learning interaction design. He justifies the design choices made by the team as an explicit response to some of the issues with formal teaching methods described above.

> "I started following a subconscious instinct to teach electronics the same way I was taught in school. Later on I realised that it simply wasn't working as well as I would like, and started to remember sitting in a class, bored like hell, listening to all that theory being thrown at me without any practical application for it." (Banzi 2009)

Mellis et al (2007) summarise their project as follows...

> "Arduino allows users to create working electronic prototypes, either stand-alone objects or devices tethered to a computer. It can read from a wide range of sensors, control a broad spectrum of output devices, and communicate with software running on a computer or talking over a network... [We are] disseminating prototyping techniques within design educations as a way of communicating the values of new interactive devices...[encouraging them] to go beyond the screen, trying to give meaning to human-machine interactivity through actually designing the machine itself" (Mellis et al 2007)

It is important to acknowledge the huge impact which Arduino has had - a project which designed the *experience* of doing electronics, with a focus on comprehensibility, shareability, experimentation and immediate results.

Projects such as Oomlout's ARDX kit (ARDX 2012) take an Arduino board as a starting point, and advance the ideas further, packaging a series of components and guides to lead novices through simple experiments which introduce the programming and circuitry required to sense or control the world using specific components.

There is a huge potential market for relatively affluent IT-literate hobbyists to play with hardware like Arduino in their free time. For them, the ARDX kit is well worth the £60 plus postage. They are confident of the things they may achieve with the kit, have all the ICT tools they need to take full advantage and are happy to accept a speculative cost in case it doesn't suit them.

Given available funding, free public Arduino workshops such as our Howduino events (Howduino 2012) sometimes take place, offering guidance and support through experiments with Arduino. However, many of the same assumptions are built-in to these public events, delegates tend to be self-selecting as affluent middle-class professionals and their families, and many are already familiar with Arduino. At the end of the day, the ARDX kits are returned to the organisers, with the expectation that interested people will go home, buy the kits themselves and use their own laptops to continue experimentation.

The early pages of Oomlout's ARDX guide detail how to download and install the Arduino Integrated Development Environment (IDE) followed by "a quick little primer targeted at people who have a little bit of programming experience" (ARDX 2012). It is excellently written, but nevertheless, the assumptions are clear - users are expected to have their own computer with internet access, to have purchased an ARDX kit, have administrative access to install new software, and already have the rudimentary grasp of math, logic and procedural programming needed to comprehend the monolithic programs used to control an Arduino.

Shrimping: Designing for Access

Microcontroller hacking offers a very rich way into the maker community, and in turn to engage with STEM, because of the diversity of projects, application domains and scientific subjects which it opens up. However, as with high school science, we are concerned that the pathways on offer to get into creating your own interactive projects using the Arduino microcontroller board are implicitly targeting those who already ‘get it’. There are some clear obstacles for young people and adults in deprived areas which we would like to eliminate.

Shrimping shares many common objectives with the Arduino project, but it is also important to note the contrasts. Like Arduino, the project is “about more than hardware and software” with a fundamental aim being to “encourage a community to form around the project” (Mellis et al 2007). However, our design work specifically targets locals in disadvantaged areas, who we aim to reach through workshopping directly, or facilitating workshops within schools, so the nature of community building is quite different.

We have a similar focus on cost as a design constraint, since if “a hardware tool is not cheap, people are hesitant to purchase it, slowing distribution and keeping it inaccessible to many people. Further, if the board is expensive, people will not use many of them, meaning they may have to disassemble one work to build the next.” (Mellis et al 2007). However, *Shrimping* is a volunteer project with no funding aiming to work with large numbers of children and adults on low incomes. Even the cost of a £15 board is prohibitive in these circumstances.

Our work is more closely modelled upon the extra-curricular activities of Yousuf (2008), Ruiz-del-Solar (2010), Sipitakiat and Blikstein (2010) and Millner and Baafi (2011) who use electronics and robotics in disadvantaged areas to catalyse “engaging science technology engineering and math activities” (Millner and Baafi, 2011) and who design tools which are “low-cost, multi-purpose, easy to assemble, and easy to be used with scrap electronics and inexpensive off-the-shelf sensors and construction materials” (Sipitakiat and Blikstein 2010).

Like these projects, in order to reach, engage and empower our target audience to we need to drastically change the model of access, compared to existing mainstream electronics kits and workshopping approaches. In particular we need to...

- provide laptops and desktops for all participants to browse maker community resources and write code, guaranteeing that those without ICT access are on an even footing
- radically reduce the cost of microcontroller boards and project components, so participants can take experiments home and deploy them in real life for pocket money
- incrementally introduce the constituents of monolithic software and hardware, to avoid alienating participants with incomprehensible complexity beyond their control
- reduce constraints on design direction to the minimum to ensure personal relevance

We believe it is possible to achieve all of these things, as part of a volunteer-only project, and with no outside funding, building on examples already out there in the open source and maker communities, and we have begun to prove it in practice.

Programme: ICT Resources

The *Shrimping* community (ShrimpingIt 2012) is centred in Morecambe, UK, with web resources hosted at http://shrimping.it and with the @ShrimpingIt twitter account for social media presence.

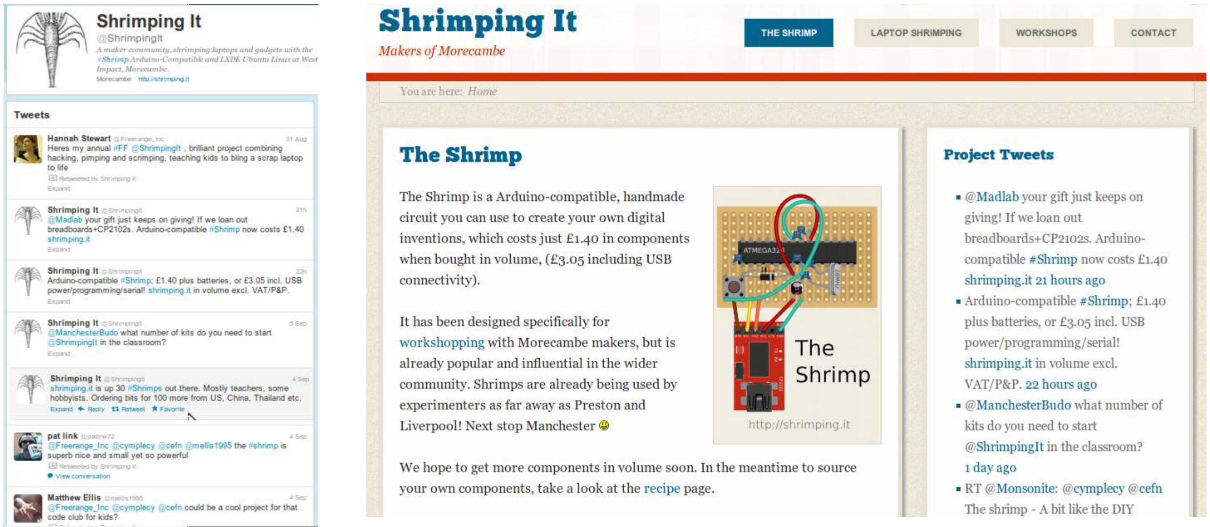


Figure 1 : Web and Social Media resources in support of the Shrimping Project

As a means of providing entry-level computing tools for our participants, we invite donations of laptops or all-in-ones from local individuals and institutions who are upgrading their hardware, typically to meet the needs of increasingly resource-hungry releases of Microsoft Windows.

When we receive hardware, it is first securely wiped. We then install a lightweight version of linux, based on Ubuntu and the lightweight windowing environment LXDE, which brings these older machines back to life. The community has so far donated more than thirty machines, of various ages and states of repair.

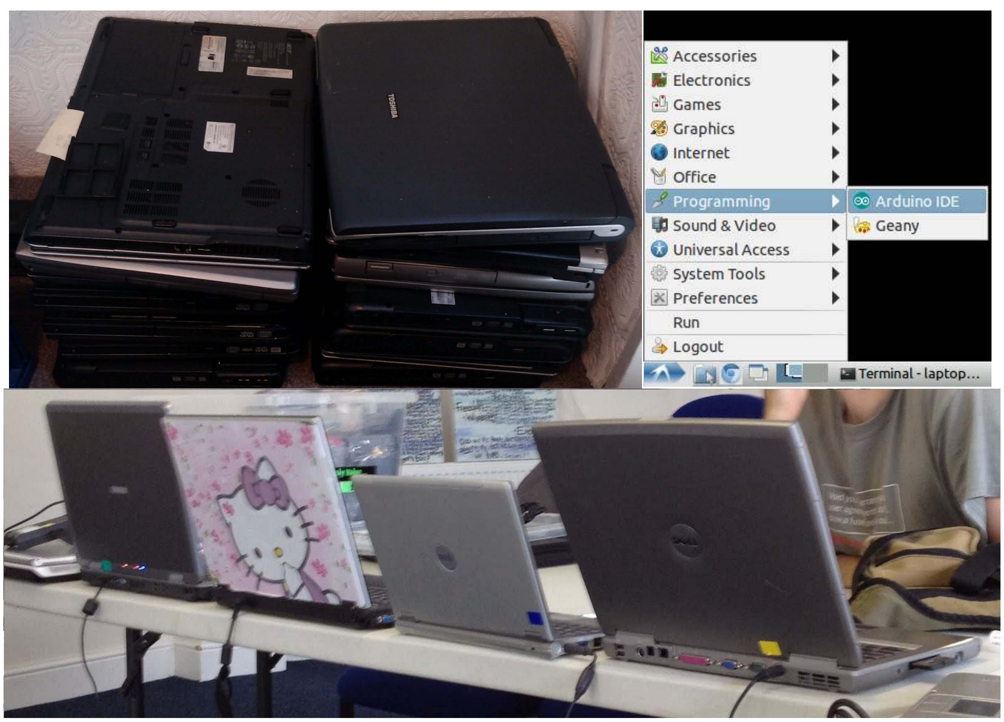


Figure 2. Collected laptops (top left) are rebuilt with an open source OS (top right) and made available as a studio resource at workshops (bottom)

Donated machines are made available to participants during workshops, providing the Chromium browser and the Arduino IDE for microcontroller programming as a bare minimum for learners to be able to explore Shrimp prototyping. Other studio software used for designing interactive behaviours include the Geany text editor for Python programming, and the Processing editor for Multimedia Java programming.

So far, ten machines have been brought into studio use. Participants in *Shrimping* workshops are eligible for a long-term loan of a studio laptop which they can take home to continue their browsing and prototyping. At the present time, we limit this offer to the unwaged or under-16s.

Programme: The #Shrimp Microcontroller Board

As an alternative to the use of Arduino boards, we employ our own design for an Arduino-compatible circuit known as the #Shrimp which costs less than one-tenth of the price.

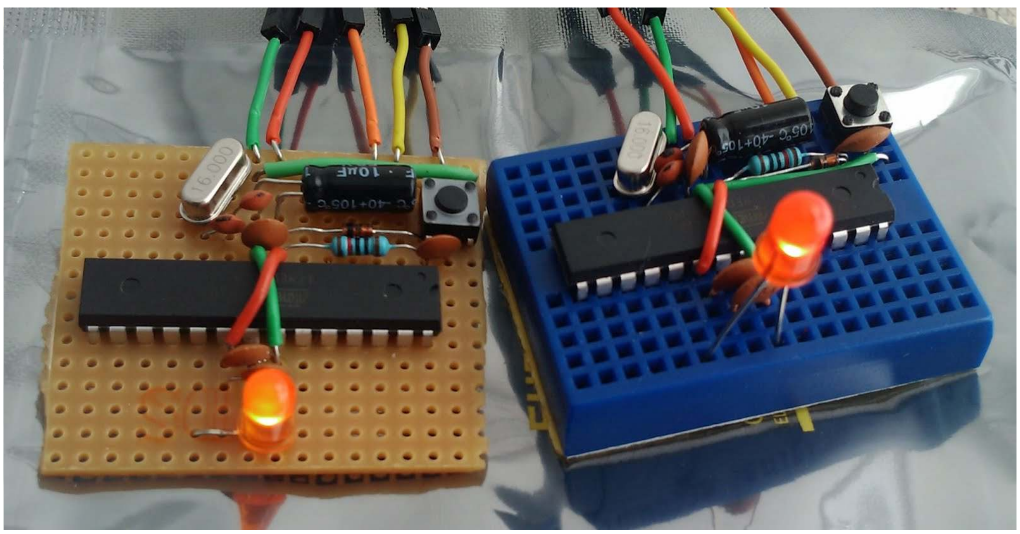


Figure 3. The #Shrimp circuit - a low-cost Arduino-compatible prototyping kit, shown here in breadboard and stripboard versions.

Figure 3 shows how easy it is for experimenters to transition from a #Shrimp prototype completed on a solderless breadboard (shown in blue) to a finished, soldered circuit on copper stripboard (shown in brown). Familiarising people with breadboard prototyping and embedding the implicit knowledge about how to easily downstream your work is central to the circuit design.

An Arduino is essentially a thin wrapper of protective electronics around the ATMEGA-328P Microcontroller and a USB (to UART) chip. Many hobbyists have developed their own homebrew circuits for using the microcontroller directly (Icecreamterror 2009, Boak 2010, Oomlout 2012). We have drawn on these experiments and innovations, to find a new compromise design which suits extremely low-cost prototyping, without compromising compatibility with the Arduino IDE or basic stability when prototyping. To translate pin numberings to the conventions used within the community, we can use the reference “pinmapping” from arduino.cc shown in Figure 4, making it possible to follow the instructions for any Arduino project using a #Shrimp.

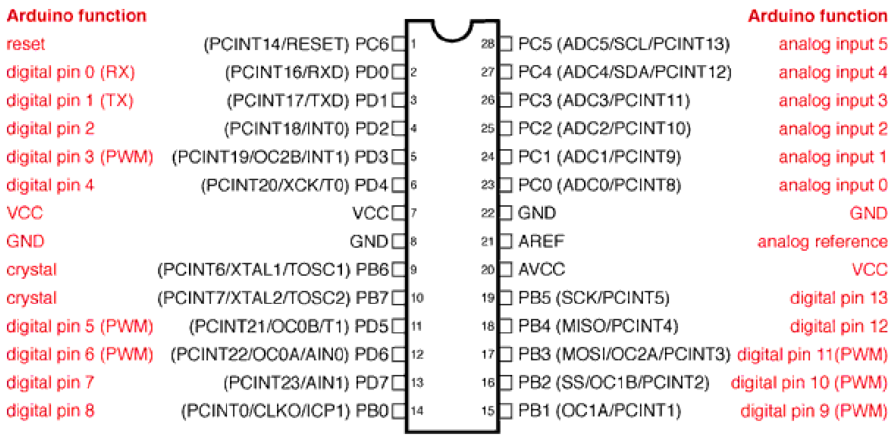


Figure 4: The Arduino to ATMEGA328 Pinmapping published at http://arduino.cc

Whilst an Arduino costs around £15 in the UK, the 328P microcontroller on its own can be sourced for around £3. The #Shrimp uses a still cheaper, compatible chip which is available in large volumes for around £1.15. We have added the bare minimum of supporting components to create a stable framework for workshop prototyping. The bill of materials for a #Shrimp assuming a large-volume purchase is as below, leading to a cost per prototyping board of less than one-tenth of the retail cost for a typical Arduino...

ATMEGA328-PU	£1.15
2x22pF Capacitor, 4x100nF Capacitor, 1x10µF Capacitor, 1x10KΩ Resistor, 1x1N4148 Diode	Each £0.01 x9 £0.09
16MHz Quartz Crystal	£0.07
6mm Tactile Switch	£0.04
5mm Red LED	£0.02
Various Colored Wire	£0.03
Total	**£1.40**

Table 1. Bill of materials for the #Shrimp low cost
Arduino-compatible prototyping board

Unlike the Arduino, the #Shrimp circuit does not include its own USB to UART converter chip, required to communicate to the microcontroller when it is being programmed, or when the device is being used as a slave device attached to a laptop. Hobbyists working with microcontrollers often use a module based on the same FTDI chipset as the Arduino, costing around £10.

To address this issue, we have sourced an alternative, standalone USB to UART adapter from China based on the CP2102 chipset, which is shown as the red module in Figure 5. These modules are available for a price of £1.65 in volume, and provide a means of powering, programming and sending Serial communications to a #Shrimp when it is not running from batteries. Combined together, the CP2102 and #Shrimp is essentially equivalent to a complete Arduino board for most prototyping purposes, at a combined cost of just over £3.

As part of our accessible technology workshops, CP2102 modules are loaned to participants for a returnable deposit of £2. This means they can be used during prototyping and programming, then returned if necessary to save on cost. On the same model, a supply of breadboards, (donated by a local electronics supplier), are loaned out for £3 each. So far, none have been returned, which we take as a very good sign!

Programme: Shrimping Workshops

To help with dissemination of this radical access strategy for microcontroller programming, we worked with experts at the Arduino forum (Arduino Forum Thread 2012) to condense their expertise into standardised circuit layouts. The resulting prototyping sequence for a #Shrimp project is outlined below. The diagrams shown in Fig 4 are used as references for constructing the circuits during workshops.

The minimal ‘Endangered Shrimp’ and ‘Protected Shrimp’ circuits are intended for breadboard prototyping. ‘Endangered’ is a simple circuit without protection, suitable for digital projects, whilst ‘Protected’ is is able to resist temporary surges and losses of voltage when the power supply experiences changes in load and electrostatic discharges such as those gathered from walking across a synthetic carpet.

The ‘Vampire Shrimp’ and ‘Camel Shrimp’ are soldered onto stripboard, and intended for deployed projects - the first drawing its power (and sometimes communications) from a host laptop, whilst the second has a self-contained power supply of its own.

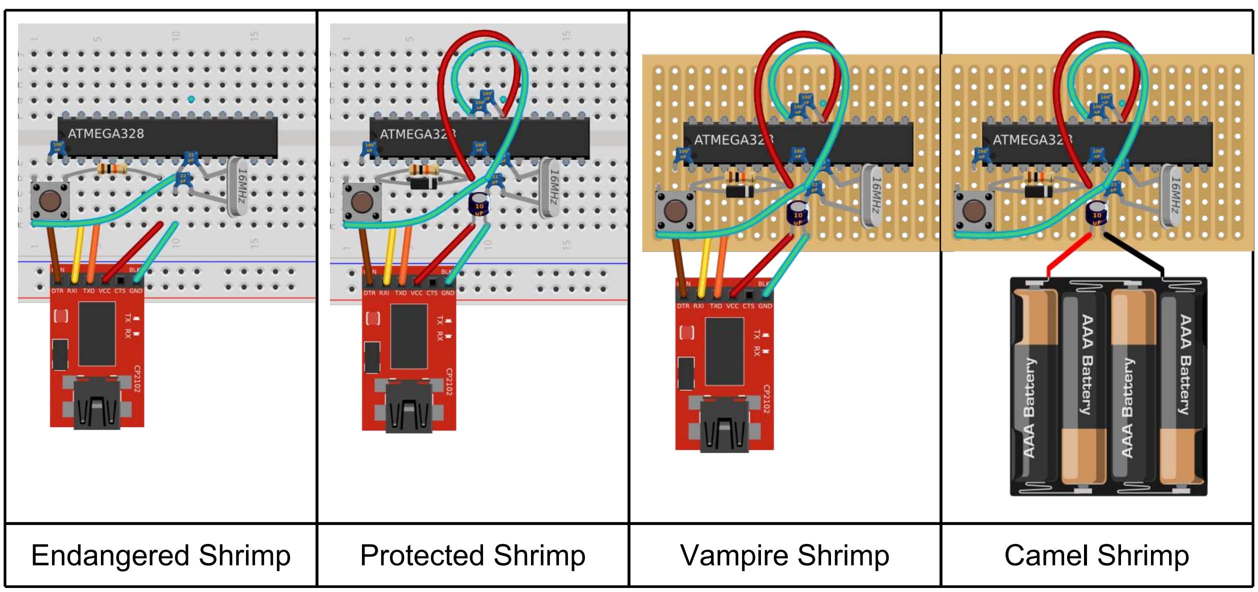


Figure 5. Stages of progression for a prototype based on the #Shrimp microcontroller circuit

Each of these circuits can be constructed within less than a minute with practice. Workshop attendees of course, take a good deal longer to complete their circuit, learning along the way about the principles of breadboards, the orientation of components and other subjects.

We have now completed #Shrimp-building introductions at three workshop venues across the North of England, alongside a campaign of presentations and online dissemination about the potential for the #Shrimp design to be used by hobbyists and as part of STEM teaching. This has led to a series of unstructured interviews and ongoing collaborations with local ICT and D&T educators from Liverpool and Manchester in the south to Carlisle in the north.

A total of approximately 30 #Shrimps have been distributed to interested experimenters, leaving us with no kits remaining after most presentations workshops, as we failed to anticipate demand.

Even more have had hands on experience with building or programming the devices. Our greatest engagement has come from teaching professionals, with active collaborations underway with more than 4 different classrooms, and one teacher asserting ‘I want kits for all my classes, yesterday’. Supply relationships are currently being negotiated to help all enthusiasts get hold of sufficient kits at cost. In particular we have now ordered components for an additional 100 units to test new classroom materials in a Manchester school before making them available more widely.

Figure 6. Various workshops and presentations in
Morecambe and around the North of England

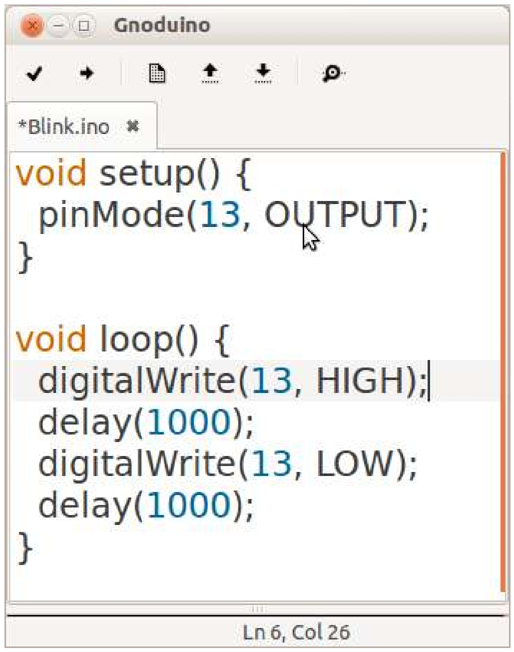
Our community engagement has ranged from Open Source enthusiasts (Oggcamp 2012) through the upcycling Mendr community (Mendrs 2012), ICT teaching groups (Raspberry Jam 2012) and technology publishers (Hoile 2012) as we tested the design concept and supporting materials.

Having road-tested the concept with enthusiasts and educators we have now begun a series of workshops in our local community alongside the Patchworks project (Patchworks 2012), which aims to improve technology access for homeless and disadvantaged adults within the West End of Morecambe.

Programme: Programming Introduction

To help users in their first encounter with programming, a form of interactive shell programming is employed, which avoids the need for understanding monolithic code blocks all at once.

To illustrate this, we can look at the classic ‘Blink sketch’, sample code which is distributed as a simple introductory example for new Arduino users. It causes the green LED attached to Arduino’s pin 13 to toggle between HIGH (light on) and LOW (light off) once a second.



Listing 1. The Blink.ino example sketch distributed with the Arduino IDE

To a computer scientist, this code provides an incredibly simple example, and makes perfect sense. By convention, any steps defined in a function called `setup()` are executed only once, whilst the steps in a function called `loop()` are repeated over and over again.

The single step in `setup()` configures pin 13 to be an output. The four steps in `loop()` turn on the LED, wait for a second, turn it off, wait for a second and then repeat in sequence forever. However, to a novice, a very large number of concepts must be simultaneously understood in order to relate the code to the behaviour in anything more than a superficial way. Although you can read this code somewhat intuitively, to write code for themselves, experimenters need to have total command of every programming detail they use.

Even the first word `void`, and the question ‘why is it there’ can open up a large and complex topic about function definitions, return values and pointers. A shortlist of concepts which are forced upon the reader from just this code listing is roughly as follows; *data types*, *the void keyword*, *functional programming with side-effects*, *function definitions*, *C statement syntax*, *code blocks*, *sequenced arguments*.

Perhaps worse, the only way that a novice programmer can verify their comprehension of any of these aspects is to upload a complete code listing to a #Shrimp, which encapsulates all of the concepts simultaneously, and see if it does what they expect. Frequently this means they are limited to copying whole programs written by others, and learning by modifying small parts. This makes it extremely difficult to create bespoke behaviour for their own project designs, as noone else may have shared code for a sufficiently similar behaviour.

Our teaching approach for novice programmers is very different, and is based on the use of a python library and firmware (pyfirmata (PyFirmata 2012) and StandardFirmata (Firmata 2012)) which allows the microcontroller to be remote-controlled from a laptop, one step at a time.

Such an approach is mirrored in Richards et al's (2012) Senseboard project. They were attempting to resolve the following issues which had arisen when teaching programming...

- Almost all conventional languages require students to passively learn a large amount of material before they can begin writing and understanding even the simplest of programs;
- Students with no background in programming were frustrated by the pedantic syntax and cryptic naming conventions used in written programming languages

(Richards et al 2012)

They claim that their success in teaching programming "rested substantially on this notion of building from one thing to the next, from immediate, direct manipulation to alternative forms of interaction, from concrete activities to programming concepts". (Richards et al 2012)

Participants are told we have put software on the #Shrimp which allows us to remote control it. We demonstrate this by issuing single, self-contained commands from the laptop in the python language, using an interactive interpreter known as the python shell (van Rossum 2012).

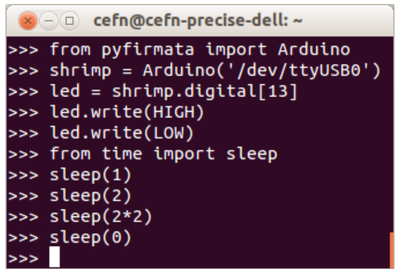
In this way we interactively introduce and demonstrate the use of individual concepts, in more or less this sequence, *values, types, expressions, names, variables, steps, functions* and *loops*, which finally provides enough knowledge to author the Blink behaviour. After encountering each statement in turn and in isolation, participants are able to confirm their understanding by typing individual steps and observing immediate effects.

An example session from one of our workshops is illustrated by screenshots and descriptions below. Of course, it is hard to recreate the interactive presentation here using only screenshots. The reader should note that each line typed is individually explained during the class and when executed they often have immediate and visible consequences. Each line preceded by >>> is a single command, sometimes followed by responses from the computer (lines without the >>>). The sequence is demonstrated at the front of the class using a digital projector, with participants using their own computer, terminal and #Shrimp to demonstrate each of the principles, and experiment with their own variations on the commands we provide.

<table>
<tr>
<td>


Here we introduce values, expressions and types.

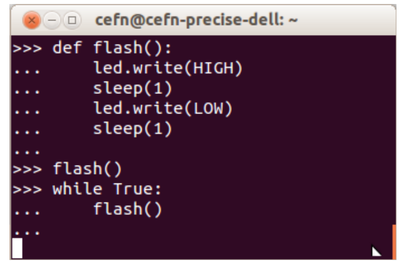
Initially the computer just parrots the text and number values `'Shrimp'` and `1` provided, showing that it recognises them. Then we introduce expressions by showing it can do arithmetic with number values.

We show how one or more values can be stored with a human readable name for our convenience. We demonstrate retrieving a named value.
</td>
<td>


The Arduino IDE tells us our #Shrimp is connected with the name `'/dev/ttyUSB0'`.

In the first line, we load the Arduino remote control functionality from `pyfirmata`. Connecting with `Arduino('/dev/ttyUSB0')` we then store the connection for later use with the name `shrimp`.

We get hold of digital pin 13 storing it with the name `led`. We light the LED on pin 13 with `led.write(HIGH)`, whilst `led.write(LOW)` extinguishes it.

We load in some time functionality and demonstrate the `sleep()` command which causes the computer to wait for the specified time in seconds before the cursor reappears. This effect is self-evident during a presentation.
</td>
<td>


Here a more complex structure is introduced - a function definition - combining principles already proven by the participants themselves.

We execute this function once, showing that the light blinks once.

Finally a while loop is introduced, triggering identical behaviour to the earlier Blink sketch, where the light blinks over and over again forever..

However, unlike the Blink sketch, every constituent of the program has been demonstrated individually.

During an interactive session, as participants ask questions it's often possible to run lines of code which correspond to their questions, giving them immediate and concrete answers.
</td>
</tr>
</table>

Table 2. A typical workshopping sequence to teach programming concepts using the #Shrimp

With a final flourish we reveal that all of the individual lines of code we have interactively typed and observed can be put into a file so they can be executed in sequence with a single click - a program, which creates a behaviour that we want. The listing, `blink.py` looks as follows.

```
from pyfirmata import Arduino
from time import sleep

shrimp=Arduino('/dev/ttyUSB0')
led=shrimp.digital[13]

def flash():
   led.write(1)
   sleep(1)
   led.write(0)
   sleep(1)

while True:
   flash()
```

Listing 2. The PyFirmata code equivalent to the Blink.ino demonstration sketch which was shown in Listing 1.

We can run this stored program by typing `python blink.py` which demonstrates the same behaviour.

This is our final step before introducing participants to the full Arduino IDE and the example Blink sketch. Blink.ino can be introduced as a way of expressing the same behaviour as Blink.py, but using the language C and the Arduino IDE to put code onto the microcontroller directly.

This sequence also allows us to discuss the advantages of having the #Shrimp able to run the behaviour on it's own, without a laptop attached, and the fact that it is a self-contained computer in its own right. We see this as a radically inverted model of disclosure from the standard model adopted as part of Arduino learning.

Programme: Experiments, Kits and Laptop Shrimping

Once participants have been guided through building the circuit, and have command of basic programming concepts, they should be in a position to approach creating a functioning project based around their #Shrimp.

For those who are not yet ready to embark on a totally freestyle design of their own, the workshop format provides at least three semi-formal routes to getting practice with the designing, prototyping and downstreaming process.

Shrimp Experiments

As mentioned earlier, the ARDX experimenters' guide offers curated code listings and explanations for various physical computing experiments; ways of sensing and controlling the world. This is complemented by the 'Learning' resources on the http://arduino.cc wiki, which provide additional experiments, similarly well-supported by code, diagrams, and explanations. Lastly, the wider Arduino community provides more of a wild-west internet resource, with individuals documenting their own projects which you can replicate for yourself. Many contributors have adopted the very high standards demonstrated by the Arduino team when documenting projects on their own blogs, making this an extremely rich resource.

Figure 7. The Input and Output box (IO Box) used during workshops

To support physical computing experiments, we make available an inputs and outputs box, (IO Box) - a collection of both mainstream components like LEDs, switches and distance sensors, and more unusual ones like smoke, pressure or albedo sensors.

To take an example from a workshop, someone wishing to experiment with Persistence of Vision (POV) grabbed a tilt switch from the IO Box, and created a display from 8 LEDs on a breadboard to prove the principle, which was able to write letters into the air. From this point they soldered together their #Shrimp and decided how they wanted to incorporate the POV circuits into their deployed project.

Shrimp Side Dishes - Project Kits

Many electronics kits lead to a project with a single function. By contrast, our design is rather a *starting point* for your choice of project, and you can repurpose for a series of multiple different projects if you wish. To help people get started, we provide example projects as so-called ‘side dishes’ for your #Shrimp.

For example, to replicate the behaviour of the ‘Simon’ memory game, which cost around £30 in 1981 you simply need a #Shrimp, four buttons, four LEDs, four resistors and a piezo buzzer.

Figure 8 shows the original Simon game design. The buttons light in sequence, playing corresponding musical notes. You replicate the sequence, (and the tune), by copying the correct sequence of button pushes. Fortunately, the patent protecting the MB Games design has long expired, leaving us free to experiment in this domain.

Figure 8. The MB Games ‘Simon’ Game. Original (top left) Arcade button equivalent (top right)
A build for homemade backlit ‘Simon’ buttons (lower 6 frames)

In Figure 8 you can see high-quality backlit arcade buttons which have been purchased for a high-profile demonstrator of the #Shrimp ‘Simon’ clone. These cost £2 each and will be embedded within a laser-cut project box as a project for a Liverpool school. However, in the same figure you can see another strategy being pursued, which could minimise the cost of a functioning project for #Shrimp enthusiasts.

A collaborator in the maker community has worked with us to develop an alternative strategy for creating illuminated buttons, using simple £0.04 tactile switches and £0.02 LEDs attached to stripboard, and using ping-pong balls as diffusers(Cook 2012). As a result of this development a kit for a 'Simon' clone may be provided for less than £1 on top of the cost of the #Shrimp itself.

The same kit of parts as the Simon (with longer wires) can be used to create a team buzzer project, suitable for a quiz game, revealing who pushed the buzzer first by issuing different sounds, and illuminating the button belonging to the fastest finger.

As the workshops proceed, delegates' innovations are able to be documented and turned into additional 'side dishes' for future delegates. An example is the POV display mentioned earlier, which needs nothing more than 8 resistors, 8 LEDs, a tilt-switch and batteries. A more advanced experiment currently being explored could lead to a POV display attached to a rotating turbine powered by the wind, a particularly suitable project for testing on the exposed Morecambe seafront.

To support specific requests from teachers incorporating the #Shrimp into their curriculum we are now preparing a series of kits and supporting documentation for projects such as a steady hand game and an internet-connected sensor board.

In an explicit attempt to build on the positive youth engagement with mobile technology discussed earlier by Ogawa and Shimode (2004), longer term developments include a design for a #Shrimp-based charging dock for Android phones, to be undertaken during a workshop at OSHCamp (2012).

Laptop Shrimping

The final model of experimentation is a synthesis of the two separate streams of our community project. On the one hand we are bringing donated laptops back to life, and on the other creating USB-powered microcontroller projects. This creates the potential for really creative upcycling of laptops, an activity described as *Laptop Shrimping*.

Our recovered laptops are generally at a disadvantage compared to modern machines. Although some donations are relatively new (the newest is around one year old), the oldest machine so far donated is a 14 year old laptop which runs at 233MHz! Machines with such specifications are never going to compete with a modern machine from a performance point of view. However, their age and low value have some advantages, it makes it possible to undertake radical modifications of the chassis, and take risks with the software and hardware build which we would avoid with a modern machine.

One Laptop Shrimping example has been documented in some detail, including a serialisation of the build process for O'Reilly to demonstrate Arduino programming (Hoile 2012). We started with a donated 11 year old Compaq Presario 700, a laptop with serious CPU performance and memory limitations. However, it is able to decode MP3s, play music and send serial communications over to a #Shrimp. We used this to construct a music playing laptop which drives a 64 element full color graphic equaliser attached to the rear of the screen. The cost of components for this modification is approximately £30, transforming a machine which was otherwise destined for disposal into something which attracts a great deal of interest.

Figure 9. Shrimped Laptop as commissioned by O'Reilly for their Safari Blog

Conclusion

The Shrimping project is a condensation of multiple perspectives and insights from the student, maker and teacher communities, made concrete through a designed technology access programme, intended for a disadvantaged community in the West End of Morecambe.

The programme combines the provision of ICT equipment alongside planned prototyping activities, based on the #Shrimp, a novel, low-cost Arduino-compatible microcontroller circuit. Cheap, accessible 'side dishes' - beginners projects to go with the #Shrimp - are under active development, designed to motivate engagement and entertain participants. Through these innovations, we aim to give confidence to participants to create their own software and hardware designs as a portal to a broader engagement with STEM subjects.

After broad experimentation and dissemination within the maker community, we have now begun a series of fortnightly technology access workshops in our local community centre. Through this community activity, Individuals are already emerging who demonstrate 'positive deviance' (Marsh et al 2004) defying the disadvantaged label applied to the area as a whole.

We reject the ecological fallacy (Vaughan and Geddes 2009) which assumes that all those in a disadvantaged area suffer from the same difficulties, and expect that the example of a few experimental community members can act as a point of crystallization for others to approach *Maker Culture* as a touchpoint for self-determined, fulfilling and eventually, economically significant innovation.

The impact of the approach is not limited to our area. The work was initially focused on a deprived area of the UK seaside town of Morecambe. However, further exploration has uncovered systemic issues in STEM schooling, which could broaden the impact of the strategy and design, and a number of collaborators are actively downstreaming our work into the UK education system.

References

Alers, S., & Hu, J. (2009). AdMoVeo : A Robotic Platform. Proceedings of Design and Semantics of Form and Movement (pp. 410–421).

Alonso, F., & Manrique, D. (2010). How Blended Learning Reduces Underachievement in Higher Education: An Experience in Teaching Computer Sciences. IEEE Transactions on Education, 54(3), 471–478.
Arduino Forum Thread (2012) Workshopping £1.40 Arduino-Compatible (£3.05 including USB!) http://arduino.cc/forum/index.php?topic=111523.0
ARDX (2012) ARDX Experimenter's Guide for Arduino. Retrieved from http://ardx.org/src//guide/2/ARDX-EG-OOML-WEB.pdf
Balogh, R. (Ed.). (2010). Robotics in Education. Reviewed Slovak Professional Magazine for Scientific and Engineering Issues. AT&P Journal.
Barak, M. (2005). From order to disorder: the role of computer-based electronics projects on fostering of higher-order cognitive skills. Computers & Education, 45(2), 231–243.
Banzi, M. (2009). *Getting Started With Arduino*. Make:Books, O'Reilly Media.
Beug, A. (2012). Teaching Introductory Programming Concepts: A comparison of Scratch and Arduino. California Polytechnic State University.
Blank, D., Kay, J., & Marshall, J. (2012). Calico: a multi-programming-language, multi-context framework designed for computer science education. Proceedings of the 43rd ACM technical symposium on Computer Science Education.
Blikstein, P., & Sipitakiat, A. (2011). QWERTY and the art of designing microcontrollers for children. Proceedings of the 10th International Conference on Interaction Design and Children - IDC '11, 234–237.
Boak, K. (2010) Sustainable Suburbia DIY Arduinos. Retrieved from http://sustburbia.blogspot.co.uk/2010/08/diy-arduinos.html
Booysen, T., Rieger, M., & Ferrein, A. (2011). Towards inexpensive robots for science & technology teaching and education in Africa. AFRICON, 2011, (September), 13–15.
Boyle, T. (2000). Constructivism: A suitable pedagogy for information and computing sciences. Proceedings of the 1st Annual Conference of the LTSN Centre for Information and Computer Sciences, 2–5.
Braund, M., & Reiss, M. (2006). Towards a more authentic science curriculum: The contribution of out of school learning. International Journal of Science Education, 1–32.
Buechley, L. (2010). LilyPad Arduino: rethinking the materials and cultures of educational technology. Proceedings of the 9th International Conference of the Learning Sciences, 2, 127–128.
Camarata, K., Gross, M., & Do, E. (2003). A physical computing studio: Exploring computational artifacts and environments. International Journal of Architectural Computing, 1(2), 169–190.
Capsense (2012) Native Capacitive Sensors without additional Hardware http://arduino.cc/playground/Code/CapacitiveSensor
Cey, T. (2001). Moving towards Constructivist Classrooms.
Chiasson, S., & Gutwin, C. (2005). Design principles for children's technology. interfaces.
Chu, R., Lu, D., & Sathiakumar, S. (2008). Project-based lab teaching for power electronics and drives. IEEE Transactions on Education, 51(1), 108–113.
Conradi, B., Serényi, B., Kranz, M., & Hussmann, H. (n.d.). SourceBinder: Community-based Visual and Physical Prototyping. Workshop on Open Design Spaces (in Conjunction with DIS 2010). Aarhus, Denmark, August.
Cope, B., & Kalantzis, M. (Eds.). (2000). Multiliteracies: Literacy learning and the design of social futures. Routledge.
Cook, Mike (2012) Personal correspondence with the author
Corso, J., & Ramirez-Fernandez, F. (2007). Sensors as an alternative way for teaching Embedded Systems and Microelectronics. International Conference on Engineering Education
Date, M., Patkar, S., Patil, M., Narendra, N., Shelke, S., Kamath, A., & Ghosh, D. (2012). e-Prayog: A new paradigm for electronics laboratories. 2012 IEEE International Conference on Technology Enhanced Education (ICTEE), 1–10.
Denis, B., & Hubert, S. (2001). Collaborative learning in an educational robotics environment. Computers in Human Behavior, 17(5-6), 465–480.
DigitalDandT (2012) Why teach electronics in schools? http://www.digitaldandt.org/index.php/electronics
Domokos, J., & Harris, J. (2011). Morecambe, the seaside town that cuts brought to a standstill. Guardian.co.uk.
Earley, R. (2011). Upcycling textiles: adding value through design. KEA: Towards Sustainability in the Fashion and Textiles Industry.
Edwards, J.-A. (2005). Exploratory talk in peer groups: Exploring the zone of proximal development. Fourth Congress of the European Society for Research in Mathematics Education (CERME 4).
Engineering and Technology: Skills & Demand in Industry, Annual Survey. (2012).
Ethnographic Action Research: A user's handbook developed to innovate and research ICT applications for poverty eradication. (2003).
Fensham, P. (2004). Engagement with science: An international issue that goes beyond knowledge. Proceedings of the Science and Mathematics Education Conference 1–10.
Fensham, P. (2006). Student interest in science: the problem, possible solutions, and constraints. Proceedings of the ACER Research Conference: Boosting Science Learning - What will it take?, 70–73.
Ferrein, A., & Marais, S. (2011). RoboCup Junior: A vehicle for S&T education in Africa? AFRICON, 2011, (September), 13–15.
Firmata (2012) A generic protocol for communicating with microcontrollers from software on a host computer http://firmata.org/

Fritzing (2010) Fritzing: Open-source software for documenting prototypes , learning interactive electronics and PCB production. .
Gray, P., & Feldman, J. (2004). Playing in the Zone of Proximal Development: Qualities of Self-Directed Age Mixing between Adolescents and Young Children at a Democratic School. American Journal of Education, 110(FEBRUARY), 108–145.
Grudin, J. (2012). A Moving Target — The Evolution of Human-Computer Interaction. In A. Sears & J. A. Jacko (Eds.), The human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications (pp. 1–40). Lawrence Erlbaum.
Haupt, M., Perscheid, M., & Hirschfeld, R. (2010). PhidgetLab: crossing the border from virtual to real-world objects. Proceedings of the 15th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, 73–77.
Hari (2010) Arduino 56x8 scrolling LED matrix http://g33k.blogspot.co.uk/2010/02/arduino-56x8-scrolling-led-matrix.html
Hogan, D. M., & Tudge, J. R. H. (1999). Implications of Vygotsky ' s Theory for Peer Learning. In A. M. O'Donnell & A. King (Eds.), Cognitive Perspectives on Peer Learning. Lawrence Erlbaum.
Hoile, C (2012) Time Traveling with Old Laptops and Arduino Compatibles. O'Reilly's Safari Online http://tinyurl.com/bmud5tb
Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. K. (2009). Comparing the use of tangible and graphical programming languages for informal science education. Proceedings of the 27th international conference on Human factors in computing systems - CHI '09, 975.
Howduino (2012) A free, one or two day workshop about connecting the internet to the real world, breathing life into inanimate objects and creating new ways to interact with things http://howduino.com
Hydronics (2012) Honey Bee Counter Instructable. http://www.instructables.com/id/Honey-Bee-Counter/
Institution of Engineering and Technology. (2010). Education and Skills Policy Statement.
Institution of Engineering and Technology. (2012a) Studying Stem : what are the barriers ?
Institution of Engineering and Technology. (2012b). Engineering and Technology Skills demand 2012 Issues and Actions.
IcecreamTerror (2009) Stripboard Arduino Instructable http://www.instructables.com/id/Stripboard-Arduino/
Jamieson, P. (2011). Arduino for Teaching Embedded Systems. Are Computer Scientists and Engineering Educators Missing the Boat? International Conference on Frontiers in Education: Computer Science and Computer Engineering.
Jenkins, E., & Pell, R. (2006). The Relevance of Science Education Project (ROSE) in England: a summary of findings.
Jeon, S.-H., Kim, Y.-G., & Koh, J. (2011). Individual, social, and organizational contexts for active knowledge sharing in communities of practice. Expert Systems with Applications, 38(10), 12423–12431.
Karna-Lin, E. (2006). Can robots teach? Preliminary results on educational robotics in special education. Advanced Learning …, 0–2.
Knörig, A., Wettach, R., & Cohen, J. (2009). Fritzing: a tool for advancing electronic prototyping for designers. Proceedings of the 3rd International Conference on Tangible, Embedded and Embodied Interaction
Kocijancic, S. (2001). Mechatronics as a challenge for teaching technology in secondary education. Frontiers in Education Conference 1–4.
Kuznetsov, S., & Paulos, E. (2010). Rise of the expert amateur: DIY projects, communities, and cultures. Proceedings of the 6th Nordic Conference on Human-Computer Interaction
Lau, W., Ngai, G., Chan, S., & Cheung, J. (2009). Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students. ACM SIGCSE Bulletin, 504–508.
Lister, R., & Leaney, J. (2003). Bad theory versus bad teachers: Toward a pragmatic synthesis of constructivism and objectivism. International Conference of the Higher Educational Research and Development Society of Australasia
Lombardi, M. (2007). Authentic learning for the 21st century: An overview. Educause learning initiative.
Lunn, S., Barlex, D., Murphy, P., McCormick, R., Revill, P., & Turnbull, P. (2012) Dispelling myths about electronics in schools.
Maor, D. (1999). A teacher professional development program on using a constructivist multimedia learning environment. Learning Environments Research, 307–330.
Mapping Deprivation. Townsend Centre for International Poverty Research (2000).
Marghitu, D., Fuller, M., Brahim, T., & Banu, E. (2009). Auburn University Robotics and Computer Literacy K-12 Engineering Camps: A Success Story. American Society For Engineering Education - Southeastern Section Annual Conference
Marghitu, D., & Zylla-Jones, E. (2007). Multidimensional Computer Literacy Program for Typical and Special Needs Children: An Auburn University Case Study. Proceedings of Society for Information Technology & Teacher Education International Conference
Marsh, D. R., Schroeder, D. G., Dearden, K. a, Sternin, J., & Sternin, M. (2004). The power of positive deviance. BMJ (Clinical research ed.), 329(7475), 1177–9.
Mazzone, E., Xu, D., & Read, J. (2007). Design in evaluation: reflections on designing for children's technology. Proceedings of the 21st British HCI Group, 2 (September).

Mellis, D. A., Igoe, T., Cuartielles, D., & Banzi, M. (n.d.). Arduino : An Open Electronics Prototyping Platform. alt.chi section of the CHI 2007 conference in San Jose (CA). (pp. 1–11).
Mendrs (2012) First Mending Research Symposium 29 June – 2 July 2012 http://mendrs.net
Millner, A., & Baafi, E. (2011). Modkit: blending and extending approachable platforms for creating computer programs and interactive objects. Proceedings of the 10th International Conference on Interaction Design and Children, 2–5.
Mills, K. (2006). Discovering design possibilities through a pedagogy of multiliteracies. Journal of Learning Design, 1(3), 61–72.
Morecambe Resort Action Plan. (2002).
Nourbakhsh, I. (2009). Robot Diaries: Creative technology fluency for middle school girls. Robotics & Automation Magazine, IEEE, (March), 16–18.
Nov, O. (2007). What motivates wikipedians? Communications of the ACM, 50(11).
Nwana, H. S. (1997). The computer science education crisis: fact or illusion? Interacting with Computers, 9(1), 27–45.
OCR (2011) Oxford, Cambridge and RSA Examinations Board Geology GCE paper retrieved from http://www.ocr.org.uk/.../ocr_66394_pp_11_jun_gce_uf794.pdf
Oggcamp (2012) A Free Culture Unconference August 18-19 2012 http://oggcamp.org/
OSHCamp (2012) Open Source Hardware Camp. A weekend of open source hardware talks and workshops. http://lanyrd.com/2012/oshcamp/
Oomlout (2012) Breadboard Arduino Compatible Kit (BBAC) http://tinyurl.com/br97nmm
http://oomlout.co.uk/breadboard-arduino-compatible-kit-bbac-p-211.html
Patchworks (2012) A research project to explore the health and communication needs of homeless people in Lancaster and Morecambe http://www.catalystproject.org.uk/content/patchworks
PyFirmata (2012) Python interface for the Firmata Protocol https://bitbucket.org/tino/pyfirmata/src
van Rossum (1999) Python and the Interactive Shell 'IPython' http://www.python.org/doc/essays/ppt/acm-ws/sld029.htm
Randall, C., & Beaumont, J. (2010). e-Society, Social Trends 41.
Raspberry Jam (2012) A global network of events for enthusiasts of the Raspberry Pi computer http://raspberryjam.org.uk/preston/
Regeneration in Historic Coastal Towns. (2007).
Revolution Education. (2003). Picaxe-18 Electronic Game Project.
Richards, M., Petre, M., & Bandara, A. (2012). Starting with Ubicomp: using the senseboard to introduce computing. Proceedings of the 43rd ACM technical symposium on Computer Science Education
Ruiz-del-Solar, J. (2010). Robotics-Centered Outreach Activities: An Integrated Approach. Education, IEEE Transactions on, 53(1), 38–45.
Schmidtbauer, M., & Johnson, S. (2012). Squishy circuits as a tangible interface. Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems
Schreiner, C., & Sjøberg, S. (2007). Science education and youth's identity construction - two incompatible projects? In D. Corrigon, J. Dillon, & R. Gunstone (Eds.), The Re-emergence of Values in the Science Curriculum (pp. 1–16).
Shrimping.it Community Website and @ShrimpingIt Twitter Feed (2012). http://shrimping.it
Silver, J. (2009). Awakening to maker methodology: the metamorphosis of a curious caterpillar. Proceedings of the 8th International Conference on Interaction Design and Children, 3–6.
Sipitakiat, A., & Blikstein, P. (2010). Think Globally , Build Locally : a Technological Platform for Programmable Bricks for Education. Fourth International Conference on Tangible, Embedded and Embodied Interfaces (pp. 231–232).
Randofo (2009) The Twittering Office Chair [Instructable] http://www.instructables.com/id/The-Twittering-Office-Chair/
Sjøberg, S., & Schreiner, C. (2010). The ROSE project: An overview and key findings. Oslo: University of Oslo, (March), 1–31.
Steeg, T., & Ling, M. (2006). Who's In Control (of the teaching of computer control)? In E. Norman, D. Spendlove, & G. Owen-Jackson (Eds.), DATA International Research Conference (pp. 135–142).
Tanrikulu, E., & Schaefer, B. C. (2011). The users who touched the ceiling of scratch. Procedia - Social and Behavioral Sciences, 28, 764–769.
Taylor, R. A., Taylor, A. R., Jackson, D. J., Jackson, A. D., Member, S., Society, I. C., Society, I., et al. (2012). A Custom-PCB Design for Microcontroller Education. American Society for Engineering Education Annual Conference.
Timcenko, O. (2007). Example of Using Narratives in Teaching Programming: Roles of Variables. Psychology of Programming Interest Group.
Unsworth, L. (2001). Teaching multiliteracies across the curriculum. Open University Press.
Vaughan, L., & Geddes, I. (2009). Urban form and deprivation: a contemporary proxy for Charles Booth's analysis of poverty. Radical Statistics, (99), 46–73.
Vogel, D. (2012). Form+Code in Design, Art, and Architecture, by Casey Reas, Chandler McWilliams, and LUST. Journal of Mathematics and the Arts, (August), 1–3.
Wang, T., & Kaye, J. (2011). Inventive leisure practices: understanding hacking communities as sites of sharing and innovation. PProceedings of the 2011 annual conference extended abstracts on Human factors in computing systems, 1–10.

Wang, W., Zhuang, Y., & Yun, W. (2003). Innovative control education using a low cost intelligent robot platform. Robotica, 21(3), 283–288.
Wired (2012) Smart Bin moved to catch thrown waste. http://www.wired.co.uk/news/archive/2012-07/24/smart-bin
Yingying, S., Liyan, G., & Zuyao, Z. (2010). Researches and development of interactive educational toys for children. 2010 International Conference on Artificial Intelligence and Education (ICAIE), 344–346.
Yousuf, M. A., Rosales, S. F., Vara, R. E. G., & De, V. (2008). Educational Robots for Economically Challenged Communities. International Conference on Engineering Education (pp. 1–9).