

A survey of the benefits and issues arising from the deployment of physical artefacts in computer science teaching

E. Currie^{1,*} and C. James-Reynolds²

¹Department of Computer Science, Middlesex University, London, England.

²Department of Computer Science, Middlesex University, London, England.

Abstract

This paper describes the introduction of the use of physical artefacts in the teaching of the curriculum in the Department of Computer Science at Middlesex University. The rationale for the change is discussed, together with a description of the various technologies and the areas in which they were deployed. We conclude with a discussion of the outcomes of the work and the conclusions reached, prime amongst which are that the policy has been successful in motivating and engaging students, with a resultant improvement in student progression. In addition to their value in the taught part of the curriculum, these technologies have enabled students to become involved in real-world projects, interacting with external organizations and producing products of value in diverse areas such as the arts and assistive technologies.

Keywords: physical computing, microcontroller, robotics.

Received on 18 January 2017, accepted on 06 June 2017, published on 09 June 2017

Copyright © 2017 E. Currie and C. James-Reynolds, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.20-6-2017.152745

1. Introduction

The study of Computer Science has always involved a balance of abstract concepts and practical work. Students sometimes find the former difficult to grasp in isolation and in the curriculum, they are not always closely integrated with the latter. Practical computer programming courses have often involved exercises that are not related to real-world problems and are often considered by students to be rather dull. This, in turn, has tended to make programming and other problem solving tasks seem to students like a necessary evil, rather than something exciting and engaging. This can affect the amount of effort that students are willing to put into their study, which together with the incremental nature of programming, tends to result in the ‘falling behind and staying behind’ reported by [1][2]. This effect has been noted in programming courses at all levels and in all countries and cultures worldwide. Those who succeed in

programming and problem solving tend to be those who immerse themselves in the subject. There is evidence that the use of physical artefacts can promote the necessary level of engagement and motivation for successful study in computer science. In the rest of this paper, we describe some examples of the deployment of physical technologies within the Information Technology and Computer Science undergraduate and postgraduate programmes at Middlesex University and discuss some of the outcomes.

2. Related work

Much of the existing literature focuses on encouraging the uptake of STEM-based subjects and use of physical computing in schools [3]. In the UK there has been a dramatic shift in the school curriculum and a new focus on computer programming; however this is not yet reflected in the experiences of those who are entering undergraduate

*Corresponding author. Email: e.currie@mdx.ac.uk

study. Blikstein [4] has noted that some platforms such as Arduino expose children to too much detail, at too low a level of abstraction. Kato [5] has attempted to address this by the development of visual interfaces. This does not seem to be the case at undergraduate level, where Okita's [6] work suggests that students who learn via low transparency text based programming languages not only did equally well in assessment as those students who learned in high transparency visual coding environments, but additionally were better placed to solve new problems with unfamiliar materials.

Herger and Bodarky [7] identified, in workshops with schools, that it is important to manage time and resources effectively in order to complete planned exercises and this is equally important at undergraduate level.

Cambron [8] explored the use of Arduino in a first-year robot-based project as a first experience of electronics and processors and found it "invaluable for retention purposes". Rubio et al [9] also found value in increased retention and engagement and the number of students who learned effectively. They also identified that the mean grades and number of high performance students did not change significantly.

3. Context

In 2013, the undergraduate computing programmes at Middlesex University underwent a revalidation. For many years, students had struggled with the computer programming and problem-solving strand of the BSc Computer Science (CS) programme, while the BSc Information Technology (IT) programme was very much management-oriented, with minimal computer programming content. Many CS students, after struggling with the programming content in the first year of their course, would transfer to the IT programme to avoid further study of programming. We were unhappy with this role for the IT programme, and with the poor progression rate of students on the CS programme, and it was decided that the revalidated programmes should both have a strong programming and problem-solving core thread. Furthermore, they should embrace the use of physical artefacts to motivate and engage students in project-based learning. Through this, students could be exposed to concrete implementations of theoretical concepts to reinforce their understanding. We now describe some of the technologies used and the areas in which they were deployed.

4. Arduino microcontrollers

The use of Physical Computing in the IT curriculum was introduced 5 years ago as a trial with final year students and fully integrated since the revalidation of the programme. First year undergraduates take a module that introduces concepts such as smart homes, embedded systems, sensors and automation, personal online presence, and simple

machine learning. Six weeks are dedicated to Physical Computing workshops using Arduino. Although most of the coursework is completed over a period of three weeks, it was considered to be important to give the students time to try out ideas and to encourage a sense of "playing" with the technology. The two-year trial had highlighted some of the difficulties students had with mapping schematics to breadboards and identification of components such as resistors. Students were also frustrated with the practical difficulties of rewiring breadboards at the start of each taught session.



Figure 1 Arduino kit.

A kit (Figure 1) was therefore developed as part of an undergraduate project [10], that allows the use of a wide range of sensors; however, these were prewired and accessed by simply patching them in using 3.5mm jack to jack patch cables. Students were originally given a specific group challenge, but are now presented with a set of criteria to which their projects must conform; the actual project is negotiated with the tutor. As the students concurrently study a Java programming module, the challenge is not just about writing code from scratch, but the process of reuse and modification of existing code, for incorporation into the system. Through this, students gain insights into the techniques of real-world development of software projects. In practice over the last two years, approximately 40% of students have chosen to move beyond the kit to using breadboards directly and have had the technical expertise to feel comfortable with this. There have been a range of different reasons for this, but greater flexibility in developing their products seems to have been paramount among these. Students post their work on Social Media, which is an important part of maintaining a portfolio and personal profile, but also serves to allow others to critique their work and for employers, family and friends to view their work in an easily accessible form.

Arduino microcontrollers have also been deployed in the first year of the CS programme. Here, they are used for group-based projects over a period of weeks, running in parallel with the students' other studies. These projects are used to give the students practical experience of otherwise rather abstract concepts such as finite state machines, set

theory, functional programming and propositional logic. The students learn Racket, a multi-paradigm dialect of Lisp, and they are able to control the Arduino directly with this language by running the Arduino Service Interface Protocol (ASIP), which was developed for this purpose and is available at [11]. Typical projects have included a three-way traffic light system and games such as noughts and crosses and battleships. CS students also learn about assembly programming with the Arduino, through use of the Atmel Studio simulator.

In a third year Multimedia Engineering module, students use the Arduino to develop a multimedia experience. This is criteria-based and must include the use of sensors and the control of media. The assessment criteria include "meaningful interaction" and a "fun" component as well as an overview of the processes involved. As in the first year, plenty of time is given to explore ideas. Those students who engage with the work produce a wide range of interesting artefacts. It is permitted to use existing code, but this has to be modified or adapted to give a new type of interaction, with any code written by others being clearly referenced. Examples of projects include 'light chimes' – sensing the position of a swinging torch to generate music, interactive T-shirts that respond to proximity of others and smart home automation.

Students on all the computing science programmes take an individual project module in their final year of study, with a strong emphasis on practical problem-solving projects. Physical computing projects are popular; however they often present challenges. For example, a project proposal might appear to be appropriate, yet be easy to implement by making relatively small changes to existing code examples available online. It is also important to ensure that students have some additional prototyping and workshop skills, such as the ability to solder and to use drawing software and laser cutters for making front panels and enclosures for the physical artefacts they produce, which enables their work to be presented in a professional manner. There is evidence that undertaking such projects also benefits students in terms of employability. Students have reported back that discussing and/or demonstrating their project work has helped them to impress potential employers at the interview stage. Several students have had their project work published, for example [12] which is valuable when they apply to study at postgraduate level.

The use of physical computing artefacts can also allow students to engage with ongoing live research projects in a very practical sense. For example, this has been achieved with aspects of a departmental research project investigating the development of Smart Home systems. A number of MSc students have been engaged to develop a so-called 'smart cooker' that provides feedback information to a home hub that can turn the cooker off if it detects that the user has fallen asleep or a pot is boiling over.

Connection of sophisticated interfaces such as electroencephalogram (EEG) sensors has allowed students to explore Brain-Computer Interfaces, leading to the development of games designed to encourage users to develop awareness of their EEG signals and the ability to

control them. It is hoped that the development of serious games of this nature will assist in training those who may have to rely on such interfaces to exercise control over their mobility or environment as a result of physical disability. Students tend to be motivated by this type of project, because they see them as simultaneously feasible to attempt, but also cutting edge in the application of computing to real world problems.

Physical computing projects also enable students to become involved with real-world projects and organizations in other ways. As early as 2011, Middlesex Design Engineering students, working with the designer Moritz Waldemeyer, used physical computing technologies to contribute to the making of Elli Goulding's video for the song 'Lights'. In the same year, computing and arts students collaborated to produce visual displays including stop-frame photography and animation, for Roger Daltrey's production of the Who's opera 'Tommy'. This was performed at the Royal Albert Hall, followed by tours of the UK, USA and Canada. Many exciting physical computing projects exist at the interface between the arts and technology. Indeed, the success of such projects at Middlesex led to the development of a new postgraduate programme in 2012, MSc Creative Technologies; a collaboration between the School of Arts and the School of Science and Technology. Although not a large programme, this course has attracted some extremely talented students who have produced some excellent work. Examples of projects completed by students on the programme include a MIDI interface for a Theremin, pollution monitoring smartclothing and interactive art installations. Physical computing projects can also be of direct value to people with disabilities. In 2015-16, three students worked on a suite of applications to assist Down's syndrome sufferers to navigate routes, automatically construct an electronic diary of their movements and to remain safe. This project was inspired by the Department's involvement in the European-funded research project Personalized Smart Environments to increase Inclusion of people with Down's syndrome (POSEIDON) [13], which began in 2013. Such assistive technology has enormous potential to improve people's lives.

5. Robots

A bespoke robotic platform (MIRTO; (Middlesex Robotic platform)) was developed for use with the first year CS students [14]. This comprises a set of HUB-ee wheels [15], an Arduino Teensy and Raspberry Pi computer running Linux. The robot is equipped with quadrature encoding, wireless card, bump sensors and infrared detectors for use in line-following algorithms and similar. The Teensy interfaces to these components and the Raspberry Pi is connected to the Teensy via its serial port. The ASIP protocol is used to enable the robot to be controlled by Racket programs loaded onto the Raspberry Pi.

The robots are used in projects that reinforce a number of CS concepts. For example, the functional programming concept of higher order functions is used to map Arduino

pin-setting functions across a number of pins and the concepts of functions as first class objects and side effects are demonstrated through lists of functions employed in causing the robot to explore an unknown area. Another example is the use of Racket Contracts to specify required robot behavior [16]. Through the use of Linux, students also gain familiarity with use of a command line interface, which most are unfamiliar with, having grown up using only graphical user interfaces.

Student projects using this technology have included PID line-following algorithms using the IR sensors and controlling robots remotely through web pages, the Twitter API and email servers. A number of students go well beyond the taught material and one team competed successfully in the Eurobot national championships, coming 4th out of 17 teams.

6. Logic circuits

The CS students also build simple combinatorial and sequential logic circuits using components such as logic gates, adders, clock sources and breadboards. This reinforces their knowledge of a number of concepts, including propositional logic. The latter is an example of the holistic approach to the curriculum, as students see the same concept in their study of fundamental underpinnings and in their programming workshops.

IT students cover topics such as multiplexing, which allows multiple signal sources to share a communication channel. This is a fundamental concept in understanding computer connectivity and is often taught using diagrams. However, by using practical circuits with a microcontroller and multiplexer chip and working with actual code to control communication, students gain a far better understanding of the process, enabling them to see it in action and also to understand the limitations of this approach.

Two student projects have used this approach. The first was a laser harp that allowed 12 notes to be played, where only 8 would have been possible without multiplexing. The second was an interface for a simulation of an analogue synthesizer that allowed the use of real knobs and switches to be manipulated, rather than using a mouse on a screen.

7. Discussion and conclusions

Some of the benefits of engaging students with a physical computing approach are

- Motivation through hands on experience
- A chance to experience a whole lifecycle from concept to prototype.
- Opportunities to engage with family, friends, potential employers and contribute to online communities
- Understanding testing strategies and designing tests.
- Opportunities to be creative with open-ended assignments

- Engaging with current debate about sensors/data /internet of things

Motivating students and encouraging an exploratory outlook in their learning is important in terms of retention and also for their induction into university culture. Working with physical manifestations challenges students perceptions of computing and require them to work in new ways, with success in the set tasks giving them confidence to tackle new problems. Research [17] indicates that collaborative learning encourages critical thinking skills, together with developing working relationships, friendships and a sense of responsibility for work. Physical computing lends itself to group work and in the first year this allows students to develop a stronger cohort identity. Critical thinking is often aligned to the higher level processes in Bloom's Taxonomy [18] such as Analysis, Synthesis and Evaluation. Engagement with these processes requires real challenges that have sufficient complexity to ensure that success is only possible through decision-making and strategies that involve critical thinking.

One of the key areas explored through physical computing is the need to develop test strategies to isolate and clearly identify the nature of any problems. In such systems, errors may lie in hardware (incorrect wiring or choice of components), software (incorrectly written code that appears to work, but which may not be carrying out the intended task) or the communication between devices (failure to correctly specify the communication port). Students need to develop analytical skills to identify where the problems lie, testing strategies to identify the nature of the problem and creative solutions in order to get their projects working. Often errors in hardware will be mistaken for errors in the software. Students also need to explain their thinking to the rest of their group, and defend their proposed approach against others who may be offering competing solutions. They need to listen to others and sometimes recognize that another's approach is superior to their own. These are invaluable skills for their future careers in the workplace. In some coursework, assessment is by video and students are expected to present an explanation of their project and its code, which also develops valuable transferable skills.

For first year students, assessment sets a threshold for progression, but does not count towards their final classification. For the final year students, it is important that assessment enables grades assigned to accurately reflect each student's achievement. The Arduino community actively encourages code reuse, but this must be properly cited and documented so that a student's individual contribution can be evaluated.

Another issue with physical computing is the need to engage with the work over a period of time. Students often become adept at managing deadlines with a "just in time" approach. The physical computing tasks require more time than students might expect when they have less experience of this type of work. However, emphasizing the exploratory nature of the projects and providing the necessary time in the labs helps to overcome this.

Developing kits, storing them and ensuring that damaged or

missing items are replaced is also time consuming. Breadboards are not ideal, as often circuits are unreliable and temporary in nature. On the other hand, there is a loss of flexibility in using self-contained kits. Introducing soldering makes it difficult to reuse components and requires specialist lab provision, but is the better option for final year projects where students may wish to make the artifact more robust and demonstrate it without the risk of failure.

Use of a Physical Computing approach has been beneficial in Computer Science and IT in motivating students and helping them engage with the subject area. It does not offer an easy option for the tutors, as they have to manage more equipment and ensure that students get through initial thresholds such as a working circuit and code, help students develop test strategies and negotiate the deliverables. For final year students, it is important that tutors are continually aware of student progress and that the “rules” for code and hardware reuse are clear.

In this paper we have presented examples from our own subject area, computer science. However, in many respects, similar approaches can be applied in other subject areas. In business studies, a real-life case study, in collaboration with a company, might require the application of a range of analytical strategies to a problem, in order to find an optimal solution. For example, understanding the nature of a business, its market and goals, will have considerable implications for how it might wish to invest profits in order to expand. Students approaching such tasks will need to analyze, develop creative solutions and apply appropriate evaluative tools to justify their decisions. The decisions they

make will have to be legally acceptable and fit into the ethos of the business, or justify any change management required, factoring this into a solution.

Furthermore, although this paper focuses mainly on the area of Higher Education, similar approaches can work well in such activities as management training situations. Activities involving physical artefacts that are inclusive and require a range of skills, such as the “Marshmallow Challenge” [19] develop teamwork and critical thinking, although in many cases the critical thinking comes in the final moments of the exercise when the task is failed. At this point, groups start discussing the assumptions they have made and realise that evaluation should occur as a cyclical process throughout a project and not just at the end. Although the marshmallow challenge appears deceptively simple, it is an excellent vehicle for demonstrating the problems with conducting projects in teams under time pressure.

Physical computing is not a panacea for teaching programming and other aspects of computer science, and it is important that tasks relate to the skills and knowledge being taught. However, we have found that it does help students with understanding some of the more abstract aspects of programming and other computing concepts. It also plays a major role in student motivation; we often find that students do not want to leave at the end of their lab sessions. Progression rates have improved since the introduction of physical computing and many students have engaged in external activities to show off their work. These include open days, National Science Week events and other external projects as described above.

References

- [1] AHADI, A., LISTER, R., TEAGUE, D. (2014) Falling behind early and staying behind when learning to program. In *Proceedings of the 25th Psychology of Programming Conference, PPIG*, Brighton, 25-27 June 2014, 77-88
- [2] ROBINS, A. (2010) Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education* **20**, 37–71. doi:10.1080/08993401003612167
- [3] BUECHLEY, L., EISENBERG, M., CATCHEN, J., CROCKETT, A. (2008) The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Florence, Italy, ACM, pp. 423–432.
- [4] BLIKSTEIN, P. (2013) Gears of our childhood: constructionist toolkits, robotics, and physical computing, past and future. In *Proceedings of the 12th International Conference on Interaction Design and Children*. New York, New York, USA, June 24 - 27, 2013, ACM, pp. 173–182.
- [5] KATO, Y. (2010) Splish: A Visual Programming Environment for Arduino to Accelerate Physical Computing Experiences. In *8th International Conference on Creating, Connecting and Collaborating through Computing (C5)*, La Jolla, California, 25-28 January 2010, IEEE, pp. 3–10. doi:10.1109/C5.2010.20
- [6] OKITA, S.Y. (2014) The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms: Relative merits of transparency in learning adaptability. *British Journal of Educational Technology* **45**, 844–862. doi:10.1111/bjet.12101
- [7] HERGER, L.M., BODARKY, M. (2015) Engaging students with open source technologies and Arduino. In *Integrated STEM Education Conference (ISEC)*, Princeton University, New Jersey, 2015 IEEE pp. 27–32.
- [8] CAMBRON, M.E. (2014) Using the Arduino in Freshmen Design. In *Enhancing the First Year of Engineering Education*, College Station, TX; August 7 - 8, 2014, pp T4A -1–3
- [9] RUBIO, M.A., ROMERO-ZALIZ, R., MANOSO, C., de MADRID, A.P. (2014) Enhancing an introductory programming course with physical computing modules. In *Frontiers in Education Conference (FIE)*, Madrid, Spain, Oct 22-25, 2014 IEEE, pp. 1–8.
- [10] CLARKSON, R. (2014) A Self Contained Arduino Toolbox, Middlesex University Undergraduate Project submitted April 24 2014
- [11] Racket Asip Client Library. Available at <https://github.com/fraimondi/racket-asip>. (Last accessed 29/03/2017)

- [12] HERNANDEZ MENGESHA, L., JAMES-REYNOLDS, C. (2016) Interactive evolutionary generative art. In *AI-2016 Thirty-sixth SGAI International Conference on Artificial Intelligence*, Cambridge, UK, 13-15 Dec 2016. 377-382
- [13] AUGUSTO, J. C., KRAMER, D., ALEGRE, U., COVACI, A., SANTOKHEE, A. (2017) *The user-centred intelligent environments development process as a guide to co-create smart technology for people with special needs*. Universal Access in the Information Society (2017) Springer Berlin Heidelberg 1-16
- [14] The Middlesex Robotic platfOrm (MIRTO). Available at <https://github.com/fraimondi/myrtle>. (Last accessed 29/03/2017).
- [15] Creative Robotics: HUB-ee. Available at <http://www.creative-robotics.com>. (Last accessed 29/03/2017).
- [16] BOENDER, J., CURRIE, E., LOOMES, M., PRIMIERO, G., RAIMONDI, F. (2015) Teaching Functional Patterns through Robotic Applications. In *Proceedings TFPIE 2015: the Fourth International Workshop on Trends in Functional Programming in Education*, Utrecht University, Netherlands, November 2015, 20-32.
- [17] ANURADHA A. (1995) Collaborative Learning Enhances Critical Thinking. *Journal of Technology Education* Volume 7, Number 1 Fall 1995, 22-30
- [18] BLOOM, B. S. (1956) *Taxonomy of educational objectives, handbook 1: Cognitive domain*. New York: (Longmans Green).
- [19] WUJEC, T. The Marshmallow Challenge. <http://www.tomwujec.com/design-projects/marshmallow-challenge/> (Last accessed 11/01/2017)