# INTRODUCING MICROCONTROLLER BASED COMPUTER PLATFORMS INTO ENGINEERING COMPUTER EDUCATION

Jian Jiang
Southampton Solent University
United Kingdom

**Abstract**

Computer skills have become essentials for most of us living and working in this modern world. For engineering students, an understanding of computer architecture, interfacing, resource management and networking is fundamental. The current module used in our university to teach students this knowledge is facing some problems, including poor student engagement, lack of practical experience and weak linkage between content. One reason for this is that modern computers are highly integrated and are well designed to prevent non-specialists from making changes to the system. Students rarely have the opportunity to look into a real computer to understand how the hardware links together. A potential solution is introducing microcontroller platforms, such as Raspberry Pi, Arduino and Phidgets, into the module. These beginner-friendly computer platforms can give students the chance to explore and actually take control of the system. Although they are less powerful, they have a similar structure to PC's. Knowledge and skills gained from the microcontroller platforms can be easily transferred to other more complicated modern computer systems. These microcontroller platforms are also cost effective and benefit from huge open sources. They have been very popular with students as well as professionals in recent years. By introducing the microcontroller platform into our module, we hope to help students' learning and improve their engagement. Building, using and working on these microcontroller platforms, students can get a better understanding of how computer systems work and a better linkage among content taught in class.

## Background and current practice

Computer skills have become essentials for most of us living and working in this modern world. For engineering students, an understanding of computer architecture, interfacing, resource management and networking is fundamental. Before coming to University, our students have gained most of their knowledge of using a computer (e.g., a PC), such as basic calculation, documentation and internet communication, through their early school education and self-learning. However, few of them actually looked into the computer to understand how the hardware links together to make the whole system work, due to the fact that modern computers are highly integrated and are well designed to prevent non-specialists from making changes to the system.

For students studying audio engineering and media technology, understanding the basic fundamental principles of computer architecture and knowledge of how software and hardware interacts with each other are especially important. They underpin other knowledge and skills that will be introduced to students in their second and third years, which students will benefit from throughout their career.

In order to help students to understand how computers work behind the screen, Southampton Solent University has this first year unit, computing fundamentals, which aims to introduce students to the basics of computer architecture, interfacing, resource management, networking, peripheral and software coding, and provide students with a foundation in computer literacy that will underpin units across Levels 5 and 6.

Currently this unit is delivered between weekly lectures and tutorial sessions. Lectures explore the fundamentals of computing theory and architecture. Tutorial sessions focus on weekly group activities that require students to apply principles covered in the lectures to given problems. The teaching schedule can be roughly divided into three sections. The first 5 weeks focus on knowledge of computing theory, such as the concepts of binary system, sampling rate and bit depth. During the second 5 weeks, the lectures focus on computer structure, and tutorials focus on software development. There is one in-class test at the end of each of these two sections. Then during the final 7 weeks students will be introduced to selected topics related to modern computer science and they will be given time to finish their final project, which requires them to produce a combined hardware and software solution for a given problem, and present it as a group.

Although these all sound well designed, we are experiencing very poor student engagement. The attendance rate for some tutorial sessions is only about 20%. Students seem to lack interest and enthusiasm for the content.

Feedback from students highlighted that:

- Students think some of the tasks designed in tutorials are not very interesting, such as using Excel table to demonstrate binary theory. They can see those tasks are to help them understand the theory. But they just don't feel exited doing it.
- They feel the unit is too theoretical and they can't see how the knowledge links to practical problems / projects. Some of the students can remember the theory very well, but still don't have a clue of what to do in the final project.
- They view the links between the three teaching sections as quite weak. The theory taught in the first section doesn't seem relevant to the software development and hardware interface introduced in the second section. Most students are getting even more confused when they start their final project as they can't see how the knowledge they learnt can be applied to the given problem.

To motivate students to learn this module, materials they are more excited about need to be included in it, especially for tutorial tasks, which can be much more flexible than theory taught in lectures. Students also need to be given more chance to use the knowledge in practical scenarios and link teaching sections with a golden thread.

## Potential Solution

After discussing with colleagues teaching second and third computer units, we found something that may help: microcontroller platforms. Microcontroller platforms, such as Raspberry Pi, Arduino and Phidgets, are basically small computers but less powerful. They have a similar structure to PC's and can conduct simple tasks. From feedback, students showed great interest in the microcontroller platform, Raspberry Pi in particular. It has been a talking point for students for several years. There are competitions between universities to use them for different interesting tasks, varying from making your TV a smart TV to using it to build a touch screen game console. They are beginner friendly, cost effective and benefit from huge open sources. Many online tutorials, instructions and developer forums can give users great support.

Some of the microcontroller platforms, such as Raspberry Pi and Arduino, are actually designed for computer education purposes. They have been adopted in teaching practices at school levels (Roman, 2015) as well as at universities (Darr, Stombaugh, Shearer, & Gates, 2007; Hamrita, 2005; Land, 2015; Leeb, 2015). For example, MIT (Leeb, 2015) actually introduced students to the microcontroller as one of the most useful tools/skills that will help them to solve practical problem and even to get greater success in their career.

More often, microcontrollers were used to assist laboratory sessions of engineering classes (Ibrahim, 2007; Bolanakis, Glavas, & Evangelakis, 2007; Jones & Joordens, 2003; Milliken & Cregan, 2012). For example, Jones & Joordens used microcontrollers as a solution to the problem that distance-education students are lacking laboratory practical experiences.  This is very similar to our case in which students are also lacking practical experience. After five years' (1998-2003) implementation, they concluded, "The solution proved to be extremely successful and very well accepted by all students" (Jones & Joordens, 2003, p.455).

To use the microcontroller, students will need to build up the system by themselves. Since they have a very similar structure to the more comprehensive versions of computers, all the principles students learned in the class can be applied to these microcontrollers and all the skills they learned could be transferred to other computer systems. As a result students gain direct and practical experience of building a computer and taking control of the system, which is just what we are looking for.

By introducing the microcontroller platforms into our unit from the beginning, microcomputers can actually act as a golden thread going through the unit. Students will need theory and hardware knowledge when they start to build the system up. They need the knowledge of software development when they program code to control the system. They can also use the system in their final projects and show how they link their knowledge together.

Raspberry Pi was introduced to students taking our unit this year in week 11 as a suggestion for their final project. Many students showed great interest already. More than half of students have decided to use Raspberry Pi in their final project. If we can actually use them in the unit (not only a suggestion), it surely can increase students' engagement, and hopefully stimulate their deep learning. Therefore, the teaching team decided to introduce the microcontroller based computer platforms into our computing fundamental unit to:

- Give students direct and practical experience of building a computer system.
- Help students to better link the content taught in the class.
- Improve students' engagement.

## Plan

The plan is to introduce Raspberry Pi (one of the microcontrollers designed for computer science education) to our computing fundamental unit.

In the first 5 weeks, when lectures are focusing on the background theory of computing, Raspberry Pi will be introduced to students during the tutorial sessions. Students can use them to practice the theory. In addition, a variety of interesting projects will be demonstrated to students and let them know what they can do with Raspberry Pi. Students can potentially use some of the ideas in their final project. Besides generating great interest from students, this should give them a clear view of what they can expect in the final exam.
In the second 5 weeks, when the unit is focusing on hardware interface and software development, small tasks will be given to students to make changes to their own Raspberry Pi and learn how the hardware actually links with each other. During the tutorial, we can also demonstrate how the software interacts with hardware. Students can practice their (Python) coding skills on the Raspberry Pi system and actually see the results of software and hardware interaction.

In the last 7 weeks, when the lectures are trying to give students more ideas about what they can do with computer systems (not limited to the microcontrollers), tutorials are focusing on helping them with their projects and helping them to recall and link the knowledge they learnt during the first two sections. Students can use Raspberry pi in their final project. But they don't have to. If they can leave Raspberry Pi behind and go explore other computer systems they are more than welcome.

If the feedback is good and students want to go further in this direction of their study, they will be introduced to the *Raspberry Pi Challenge,* which acts as an interest group in many universities. Students can establish their interest group in Solent and network with people interested in using Raspberry Pi from other universities.

## Conclusion

In conclusion, a microcontroller platform, Raspberry Pi, is planned to be introduced into a computing fundamental unit. From the literature research and our current student feedback, it has great potential to solve the problems met during current teaching practice. The plan is for the next academic year. However, as a preliminary test, students have been encouraged to use Raspberry Pi in their final project this year. Although it was introduced at a fairly late stage of the unit, students have shown great interest. As a result, more than half of students decided to use it in their final projects. By taking advantage of the huge online tutorials/resources, surprisingly, they have managed to establish the system by themselves and proposed some very interesting ideas, such as building up a motion control CCTV camera and building a drum recording device. From the feedback obtained at this stage, most of the students feel the Raspberry Pi helped their studies. On many occasions, they have to go back to check their lecture notes in order to achieve some features they want to achieve on the device. In addition, they do feel they better understand the whole concepts of this unit and feel less scared of opening a real computer.

From next academic year students will be guided step by step to build the micro-computer system by themselves throughout the unit. They can practice the knowledge taught in class on Raspberry Pi. This will give them very direct and practical experience of the computer system. The Raspberry Pi will also link the knowledge they learnt together, and they can use it in their final project. This will give students a much clearer view of the unit and help them to understand the topic better. Finally, taking advantage of so many interesting existing projects, we hope to increase students' interest, improve engagement and help them to think / learn beyond the unit.

## References

Bolanakis, D. E., Glavas, E., & Evangelakis, G.A. (2007). An integrated microcontroller-based tutoring system for computer architecture laboratory course. *International Journal of Engineering Education, 23*(4), 785-797.

Darr, M. J., Stombaugh, T. S., Shearer, S. A., & Gates, R. S. (2007). A new course to teach microcontroller and embedded networking to biosystems and agricultural engineers, *International Journal of Engineering Education, 23*(4), 716-722.

Hamrita, T. K. (2005). Robotics, microcontroller and embedded systems education initiatives: An interdisciplinary approach. *International Journal of Engineering Education, 21*(4), 730-738.

Ibrahim, D. (2007). Low-cost microcontroller-based hardware for introducing digital filter fundamentals to students. *International Journal of Engineering Education, 23*(5), 1000-1010.

Jones, J. T., & Joordens, M. (2003). Distance learning for laboratory practical work in microcontrollers. *International Journal of Engineering Education, 19*(3), 455-459.

Land, B. R. (2015). Microcontrollers as components in electronic design. Retrieved from http://people.ece.cornell.edu/land/courses/ece4760

Leeb, S. (2015). Microcomputer project laboratory - Spring 2015. Retrieved from http://web.mit.edu/6.115/www/page/home.html
Milliken, J., & Cregan, M. (2012). Embedded C programming: A practical course introducing programmable microprocessors. *European Journal of Engineering Education, 37*(6), 557-574.
Roman, S. (2015). Teaching literature with the Raspberry Pi. Retrieved from http://www.edutopia.org/discussion/teaching-literature-raspberry-pi

## Author Details

Jian Jiang
james.jiang@solent.ac.uk