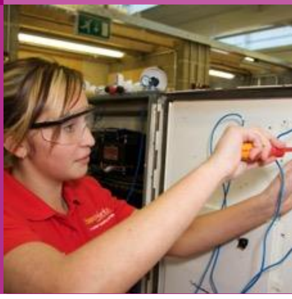
PROCAT
PROSPECTS COLLEGE OF ADVANCED TECHNOLOGY

CASE STUDY

Using Raspberry Pi and Python to develop programming skills

Sector area: Electronics and Automation



Commissioned and funded by
The Education & Training Foundation

Using Raspberry Pi and Python to develop programming skills

1: CONTACT DETAILS

Name: Silvia Tinena Coris

Role: Electronics and Automation Course Lead

Industry/sector area: Electronics and Automation

Email: silvia.tinena-coris@procat.ac.uk

2: WHAT DID YOU SET OUT TO DEVELOP AND WHY?

I wanted to motivate students to engage with and develop their understanding of programming which can for some appear initially theoretical and daunting. After my first session with students I had asked them what they felt about programming. They replied that it was difficult, 'for geeks', and was not very cool at all!

I knew then that I needed to change my approach and to support the delivery of the HNC unit 58, microprocessor skills, I introduced the **Raspberry Pi**, which costs £30, and **Python**, a free open source, as the programming language.

Commissioned and funded by The Education & Training Foundation

3: THE PROCESS - WHAT DID YOU ACTUALLY DO?

I completed some research about how other teachers have been using Raspberry Pi and Python to get some inspiration. There are loads of books and websites dedicated to doing projects with the Raspberry Pi and Python and both companies run useful communities of practice where up-to-date ideas are shared.

I then explained the basics of programming to my students and we started doing some easy programmes, which were upgraded and personalised depending on the individual. This way, I was able to ensure that each student was fully engaged in what they were doing and had 'ownership' over what they developed – it was theirs.

After completing programming on the screen, it was time to move on to some embedded programming. For that, we used the **BerryClip General Purpose Input/Output (GPIO) board**), which is an attachment to the GPIO pins. This provides a physical interface between the Raspberry Pi and the outside world, and can be programmed in Python as well (See the PowerPoint in Section 6 below titled 'using the berryclip GPIO board'). At first the learners were asked to do simple things such as making a light-emitting diode (LED) blink. They were shown the basics and were asked to change the LED that was blinking, the speed, create a sequence… then they came up with more ideas, such as doing traffic lights, a roulette, a counter, a thermometer… As usual, they all helped each other and tested their programmes.

For their third assignment, they created a stripboard that included as many LEDs/buzzers/push buttons/7-segment displays as they wanted, and then they programmed it. Some students created a 'piano', with different keys, different sounds and different colours; others did a 'lock', with a numeric keyboard; another group did a moisture sensor; and one created a reaction game… all of the students were fully engaged and were learning by doing.

Commissioned and funded by
The Education & Training Foundation

I have saved all of these examples and all of the students' input and will use this to introduce these topics to future classes and to share my practice with other staff.

4: WHAT DIFFERENCE HAS THIS MADE?

The fact that the students were given freedom to do their own programmes, according to their interests, was motivating and allowed each student to work at their own pace using peer support where necessary. The process encouraged:

- experiential learning – learning through doing, rather than seeing or believing;
- problem-solving – students had to think, analyse the data and use each other to iron out glitches;
- collaboration – students tested each other's programmes.

Overall the process made students realise that programming can be fun! They all enjoyed the unit and each of them decided to incorporate a microcontroller in their final HNC project. They were also so motivated by Raspberry Pi that they each purchased the app of their own accord.

From a tutor's perspective using these applications enabled me to differentiate pace and content. I recognised the importance of allowing choice to enhance motivation and engagement and was able to stand back and learn from my own students. For example, when teaching the 'dictionary' type in Python, I asked the group to create a shopping list to understand the concept (See the 'introduction to Python' PowerPoint in Section 6 below). Not only did they enjoy it, but they also invented a new way of playing: giving the other person 10 chances to guess as many products from their shopping list as possible. Then, they suggested changing the shopping list for things that they had in their car, in their bedroom… and the lesson took off in all directions! I captured their ideas and am using them to support my work with future students.

5: LESSONS LEARNT

The fact that students were given freedom to develop their own programmes and to customise the messages to be displayed on the screen was a great idea, as they felt that they were part of the teaching, learning by doing rather than sitting as a passive element being done to.

In addition, new ideas for new programmes kept appearing as they were working. Sometimes inspiration struck me and I felt excited telling them about a possible upgrade for a programme; other times, it was the students who felt inspired and wanted to share a great idea with me and with each other. For example, when working with the 'security word' programme, we had the idea of doing something similar to a login system – ask the user for a login and a password. As a result all students ended up understanding how these systems work!

I can say that this unit is great for encouraging students to engage with each other and work collaboratively, so I would definitely encourage it to be done at the beginning of the year. Maybe next time I will not do so much pseudo-coding and throw the students in a bit more at the deep end, so they learn as they go right from the start! Experience has shown that students do not need to know as much theory as I initially thought and that they learn better from doing rather than from being told.

6: RESOURCES PRODUCED

- An introduction to Raspberry Pi (PPT)
- An introduction to Python (PPT)
- Using the BerryClip GPIO board (PPT)

Commissioned and funded by
The Education & Training Foundation