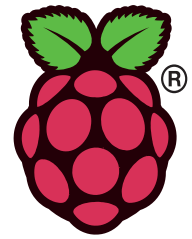


CLASSROOM CHALLENGE: (ADVANCED) INSTRUCTIONS AND ANSWERS FOR TEACHERS

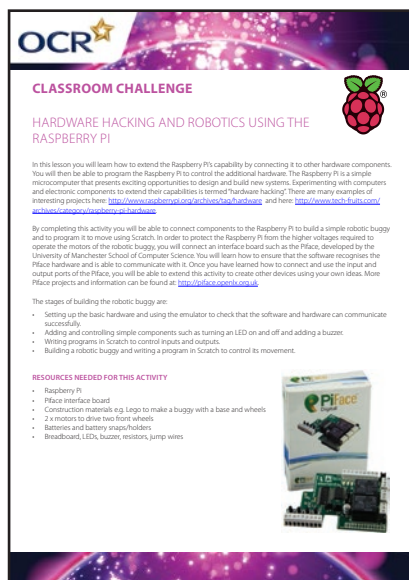


HARDWARE HACKING AND ROBOTICS USING THE RASPBERRY PI

These instructions should accompany the OCR resource 'Hardware Hacking and Robotics using the Raspberry Pi'.

The term "hardware hacking" is used to describe the process of adding and modifying components to extend capabilities. There are a large number of innovative projects online that provide inspiration, some examples can be found on the following sites: <http://www.raspberrypi.org/archives/tag/hardware> and <http://www.tech-fruits.com/archives/category/raspberry-pi-hardware>.

"Hardware Hacking" is a sector term and is not to be confused with computer hacking which is a criminal offence.



OCR

CLASSROOM CHALLENGE

HARDWARE HACKING AND ROBOTICS USING THE RASPBERRY PI

In this lesson you will learn how to extend the Raspberry Pi's capability by connecting it to other hardware components. You will then be able to program the Raspberry Pi to control the additional hardware. The Raspberry Pi is a simple microcomputer that presents exciting opportunities to design and build new systems. Experimenting with computers and electronic components to extend their capabilities is termed "hardware hacking". There are many examples of interesting projects here: <http://www.raspberrypi.org/archives/tag/hardware> and here: <http://www.tech-fruits.com/archives/category/raspberry-pi-hardware>.

By completing this activity you will be able to connect components to the Raspberry Pi to build a simple robotic buggy and to program it to move using Scratch. In order to protect the Raspberry Pi from the higher voltages required to operate the motors of the robotic buggy, you will connect an interface board such as the Piface, developed by the University of Manchester School of Computer Science. You will learn how to ensure that the software recognises the Piface hardware and is able to communicate with it. Once you have learned how to connect and use the input and output ports of the Piface, you will be able to extend this activity to create other devices using your own ideas. More Piface projects and information can be found on <http://www.piface.org.uk>.

The stages of building the robotic buggy are:

- Setting up the basic hardware and using the emulator to check that the software and hardware can communicate successfully.
- Adding and controlling simple components such as turning an LED on and off and adding a buzzer.
- Writing programs in Scratch to control inputs and outputs.
- Building a robotic buggy and writing a program in Scratch to control its movement.

RESOURCES NEEDED FOR THIS ACTIVITY

- Raspberry Pi
- Piface interface board
- Construction materials, e.g. Lego to make a buggy with a base and wheels.
- 2 x motors to drive two front wheels
- Batteries and battery snap holders
- Breadboard, LEDs, buzzer, resistors, jump wires

Expected approximate duration:

Task 1 - 10 minutes

Task 2 - 10 minutes

Task 3 - 10 minutes

Task 4 - 10 minutes

Task 5 - 30 minutes

Task 6 - 30 minutes

Task 7 - 30 minutes

This activity allows learners to use the Raspberry Pi to connect a variety of components to control system input and output. The Raspberry Pi operates at 5v and it is imperative that components are not connected directly to it that will draw significant power from it. To protect the Raspberry Pi's circuits, an interface board should be used. In the examples shown, the Piface developed by University of Manchester School of Computer Science is used although other similar products are available. The Piface is simple to connect components to and the specialised version of the Raspberry Pi operating system communicates with the board and the Scratch programming IDE. Further Piface documentation can be located at: <http://piface.openlx.org.uk>. The purpose of this activity is to build a robotic buggy which is controlled by the Raspberry Pi and programmed using Scratch.

It is recommended that this is trialled before being introduced to learners.

Resources used in illustrations are:

- Raspberry Pi Model B
- Piface interface board
- Raspbian Debian Wheezy Pifaced Operating System installed on a 4GB SD Card. This can be downloaded from: <http://pi.cs.man.ac.uk/download/>
- USB keyboard, mouse, powered USB hub
- Micro USB Power supply
- Breadboard
- 1 x 6v piezo buzzer
 - This example uses Lego RCX 9v motors. More information can be found at: <http://www.philohome.com/motors/motorcomp.htm>
- 4 x buggy wheels
- 2 x 9v PP9 batteries with battery snap type connectors
- Jump wires
- Small flat screwdriver

Information about the Piface interface can be found at: <http://pi.cs.man.ac.uk/interface.htm>

The Piface has 8 input ports, 8 outputs ports including 2 connected to relays. It connects to the GPIO (General Purpose Input Output interface) on the Raspberry Pi. Make sure that the Raspberry Pi is not powered up when you connect the Piface to it.

TASK 1 – BASIC HARDWARE SETUP

This activity covers basic equipment setup. It is advisable that the teacher connects the Raspberry Pi to the Piface interface board since this requires only gentle pressure and the GPIO pins may be easily damaged or bent. Once connected, the Piface can remain in situ for the rest of the activity.

Once the hardware is connected, the next phase is to ensure that the operating system recognises that the Piface is connected. During the course of the activities, the steps in Task 1 may need to be repeated if it is found that a program is not operating.

TASK 2 – CONTROLLING AN LED

This task makes use of the Piface's own LEDs. Learners will recognise that an output or input can have one of two values: 1 (on) or 0 (off). Setting the value in the code will turn the port on or off or recognise an input value in order to produce an output response to a condition. This is a key concept for all the following programming tasks.

TASK 3 – ADDING A BUZZER and TASK 4 – CONTROLLING A MOTOR

These two tasks involve the use of output devices. A piezo buzzer (which can be bought online for approximately £2) will normally operate around 6v so can be connected to an output port without the use of a relay. The motor however, should be used with a relay. Allow learners to experiment with the **Normally Open** and **Normally Closed** connections so that they understand that they can either write code to turn the motor on when required or to turn the motor off respectively.

TASK 5 – BUILDING A ROBOT and TASK 6 – PROGRAMMING A ROBOT

These two tasks guide learners through building a simple buggy robot. They should be provided with either construction kit components or individual components consisting of motors, gears (to reduce the motor speed), axles, battery packs and materials for building the chassis. It is imperative that the motor is supplied with power of an appropriate voltage and that connections with the Piface are made using the relay ports to protect the board.

TASK 7 – ROBOTICS INVESTIGATION AND RESEARCH

This final activity provides learners with background information on robotics. This will assist in developing skills of research and enquiry that will support practical investigation as well as developing elements such as algorithm development and understanding computer systems. They will be able to evaluate the positive and negative impacts of the use of robotics and consider the importance of reliability.

Learners should be encouraged to visit the following sites:

<http://en.wikipedia.org/wiki/Robotics>

<http://www.nasa.gov/audience/foreducators/robotics/home/index.html>

<http://spectrum.ieee.org/robotics/>