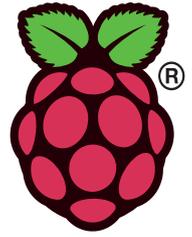


CLASSROOM CHALLENGE



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 at: <http://piface.openlx.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 snaps/holders
- Breadboard, LEDs, buzzer, resistors, jump wires



TASK 1 – SETTING UP THE BASIC HARDWARE (10 minutes)

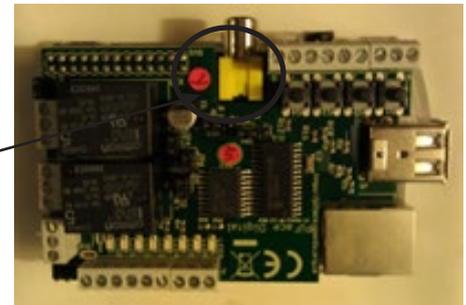
Set up the Raspberry Pi as shown below. You will also need an **interface board** to connect to your Pi to control components such as motors as they require a higher current (and often a higher voltage) to operate. This provides protection to the Raspberry Pi's components by using a device called a **relay** to turn the motor on or off.



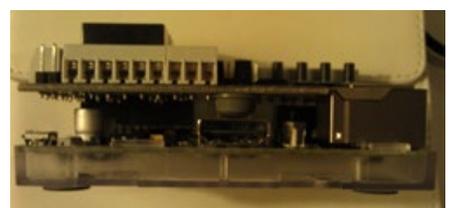
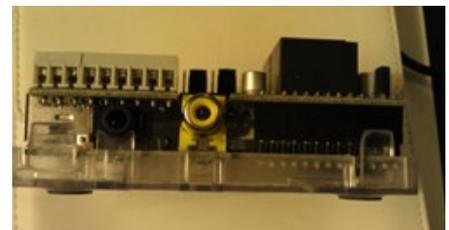
This image shows the Raspberry Pi's **GPIO (General Purpose Input Output)** connector. The Raspberry Pi operates at 5v DC and it is possible to connect components that do not draw a large current e.g. **LEDs (Light Emitting Diodes)** without using an interface board. However, you still need to ensure that you know the voltage required so that you do not cause damage to the Raspberry Pi.



In this image, a **Piface** interface board has been connected to the Raspberry Pi. This allows you to connect a wide range of hardware components but also protects the Raspberry Pi. The operating system installed onto the **SD card** is slightly different from the standard **Raspbian Wheezy Debian Linux** image. It contains code to allow the Piface to operate with Scratch and Python to write control programs.

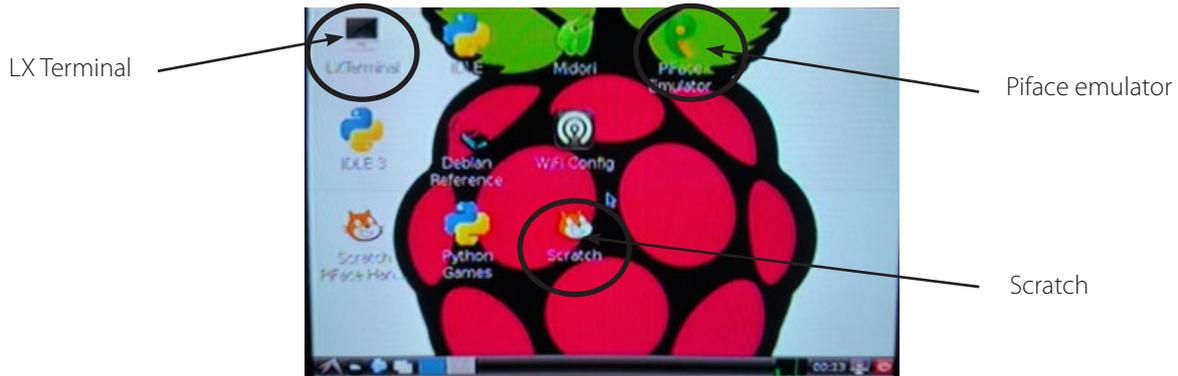


The Piface connects to the Raspberry Pi's GPIO connector. If you are doing this, make sure that you are very gentle and that the rubber foot sits on the Pi to keep the board level.



Below are steps for setting up the Raspberry Pi ready to use. You will need to repeat these steps every time you want to use the Pi with the Piface:

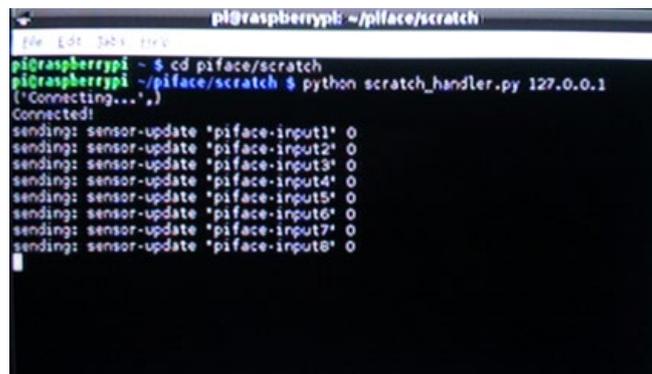
1. Connect up a USB hub with a keyboard and mouse and plug in the power supply. Boot up the Pi and login using the username **pi** and password **raspberrypi** unless a separate user account has been set up in which case your teacher will tell you what to type in.
2. Enter **startx** at the **raspberrypi\$** prompt.



3. Click on the **Scratch** icon on the desktop. **Do NOT click on the Scratch Piface Handler link!**
4. Click and hold the **Shift** key on the keyboard and click on the **Share** menu item in Scratch and select **Host Mesh**. You should see a pop up showing the IP address 127.0.0.1. Click OK.
5. Minimise Scratch and click on the **LX terminal** link on the desktop.
6. Type the following at the prompt:

```
cd piface/scratch
python scratch_handler.py 127.0.0.1
```

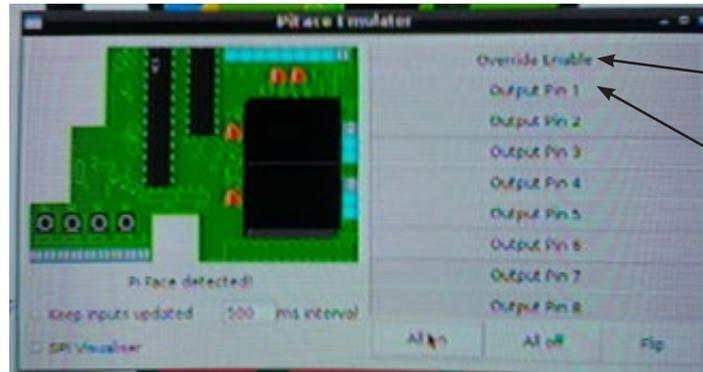
You should then see some output showing that the Piface is now connected.



Note: You will need to repeat step 4–6 each time you close Scratch or the LX terminal window. If you find that a program appears not to be connecting to the Piface, run these steps just in case the connection has been lost.

USING THE PIFACE EMULATOR

The Piface version of the Raspbian operating system also has an icon on the desktop to the **Piface Emulator**. This allows you to check the connection with the device.



Click override enable.
Click on output with Piface connected.



LED now indicates output is on.

TASK 2 – CONNECTING SIMPLE COMPONENTS - CONTROLLING AN LED (10 minutes)

The Piface has 8 LEDs connected to its output ports. These can easily be programmed to turn on and off. Output ports 1 and 2 are also connected to the two relays.

Firstly connect Scratch to the Piface as detailed in Task 1.

1. Click on **Variables** then **Make a variable**.



2. Name the variable **piface-output1**.
3. Repeat this until you have created variables for piface-outputs 2 to 8.



4. Create the program below. Each output can only have a value of **0 (off)** or **1 (on)**. Think about what the program will do before you run it. The **space bar** being pressed acts as input to begin the program.



5. Change the program so that it turns the LEDs on and then off in sequence so that only one LED is lit at any moment.
6. Write another program that turns on the odd number outputs (1,3,5,7) and then turns them off and turns the even number LEDs on (2,4,6,8). This sequence should repeat 3 times.

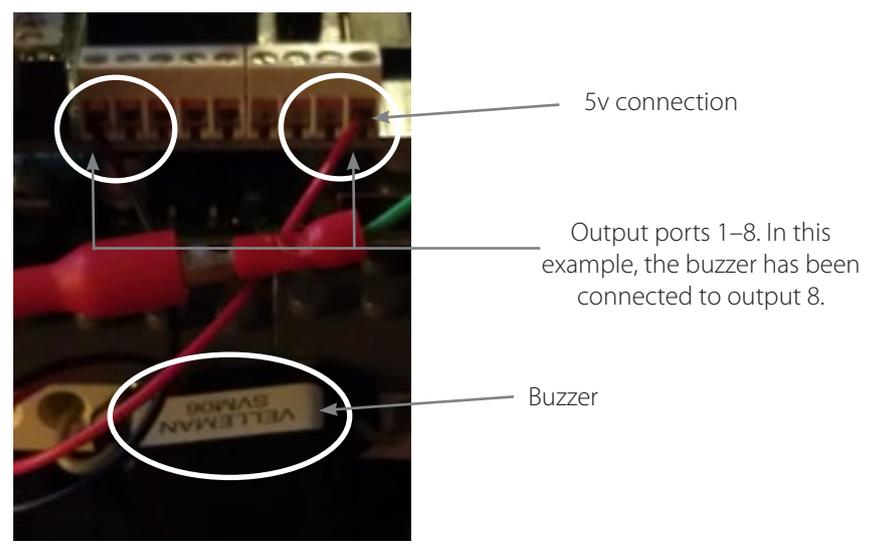
The Piface also has 4 push button switches that can be used for input to trigger an output. The following program uses a switch input.

7. Change the program so that **piface-input1** switches on **piface-output1** and piface-inputs 2, 3 and 4 switch on the outputs with the same number.

TASK 3 – ADDING A BUZZER (10 minutes)

The Raspberry Pi or Piface are not supplied with any sound output devices however, it is possible to connect a **piezo buzzer** like the one shown below.

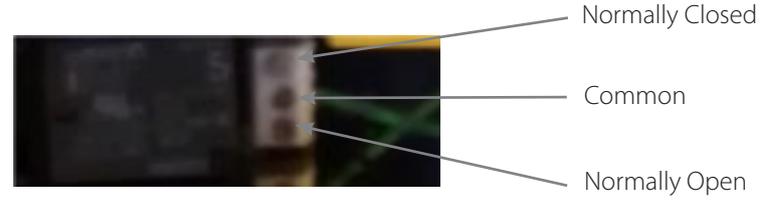
1. Connect the buzzer as shown below. The rightmost port is the 5v output and you also need to connect the negative wire of the buzzer to another output port.



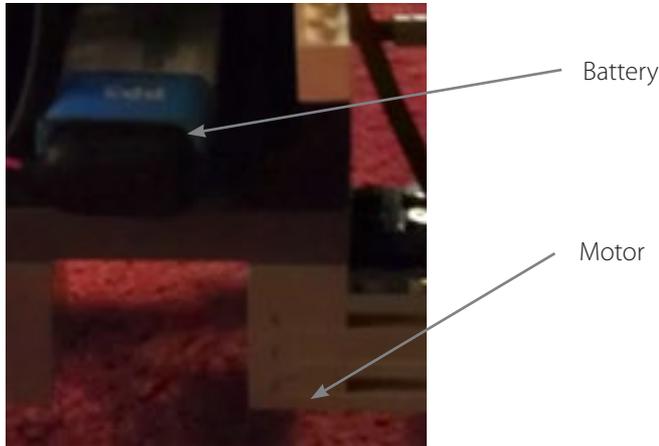
TASK 4 – CONTROLLING MOTORS (10 minutes)

Another very useful output device is a motor. This allows us to create devices that move. To begin with, you will connect a DC motor to an output port on the Piface. However, electric motors often draw more current than other devices and may operate at a higher voltage that would damage the Raspberry Pi. To provide protection, the Piface has two relays installed. A relay is an electrical switch that allows a low powered circuit to turn on a higher powered one. However, the two circuits are isolated. Relays are also used to switch on a car starter motor (a high powered circuit) when the key is turned in the ignition (a low powered circuit).

Each relay has three connecting pins. The centre pin is **common**. You should always connect to this port. The top pin of the three is called **Normally Closed** you should connect to this pin when you want the relay to break the circuit when it is activated. The bottom pin is **Normally Open** which will cause the relay to close the circuit when activated. Since we wish to turn a motor on when the relay is switched, we need to connect one wire to the common and the other to the **Normally Open** port.

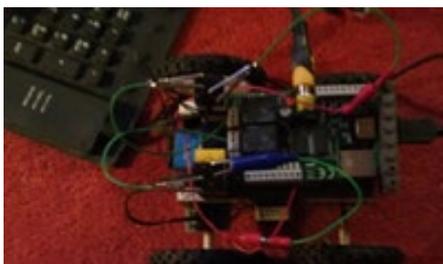
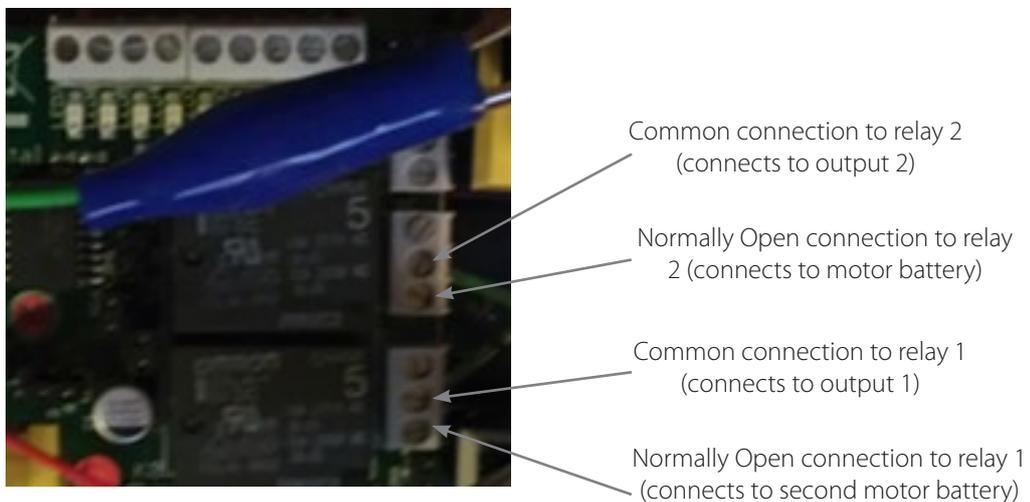


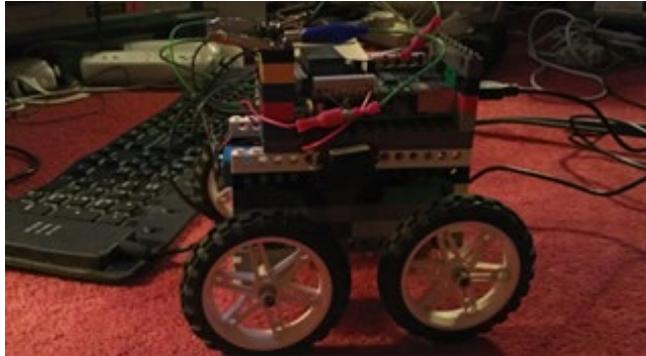
The motor illustrated comes from a Lego kit. Its operating voltage is 9v. This is being supplied by a 9v battery connected in the circuit.



TASK 5 – BUILDING A ROBOT (30 minutes)

Using a construction kit or individual components, you are now going to create a robotic buggy controlled by the Raspberry Pi with a fitted Piface. Your buggy will need four wheels with two being driven by motors each connected to a relay on the Piface as shown here:

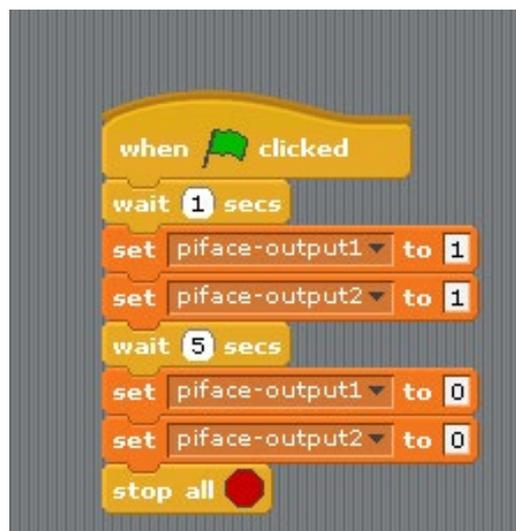




The finished robot buggy.

TASK 6 – PROGRAMMING THE ROBOT (30 minutes)

1. Connect the robotic buggy to the Piface relays as outlined above.
2. Check that Scratch is connected to the Raspberry Pi as in Task 1.
3. Create a program to move the buggy by turning the motors on and off.
4. The above program moves the buggy in a straight line. Turning one motor off will cause the buggy to turn. Write a program to move the buggy so that it moves and turns to effectively “draw” a square and return to its original starting position.



TASK 7 – INVESTIGATING HOW ROBOTS ARE USED (30 minutes)

Through your investigation, you have learned that the robotic buggy is controlled by a program that switches the relay driving the motor on and off as required. This can also be controlled by **sensors** as input devices. Robotic output devices are called **actuators**. They may be motors but could also be **pneumatic** or **hydraulic** devices.

Robots are used in a number of different industries e.g. car manufacture, nuclear power plants, military applications, space exploration and even medicine. There are advantages and disadvantages associated with their use. It is important that robots are reliable as they are often used in safety-critical environments or in remote situations where maintenance and repair would be very costly e.g. space.

In this task, you will **investigate and research** the use of robots. You should focus on:

- The uses of robots giving specific examples
- Why it is important that they are reliable
- What input and output devices are used in the examples that you have given
- Try to find information about how they are programmed

Create a **presentation** of your findings.

Here are some useful websites:

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

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

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