

CamJam EduKit Robotics - Introduction

Project Setting up your Raspberry Pi

Description Set up your Raspberry Pi and run your first python program to print “Hello World” to the screen. You will not be connecting any of the contents of the CamJam EduKit Robotics kit to the Raspberry Pi for this short exercise.

The CamJam EduKit Robots Worksheets

The CamJam EduKit Robotics is the third kit in the CamJam EduKit series. You do not have to have used the first two kits to be able to use this kit, although depending on your knowledge of the Raspberry Pi, its GPIO (General Purpose Input/Output) pins and electronics, you may find it worth reading through the previous worksheets to understand some of the more basic concepts. You can download the worksheets, for free, at <http://camjam.me/edukit>.

The EduKits are compatible with all flavours of the Raspberry Pi – Models A, B, A+, B+ and Pi2.

Equipment Required

For this worksheet you will require:

- A Raspberry Pi
- An SD card to fit your version of the Pi (8GB recommended)
- Monitor & cable to connect to the HDMI or composite output of your Pi
- A keyboard and mouse
- A Raspberry Pi power supply

Setting up your Raspberry Pi

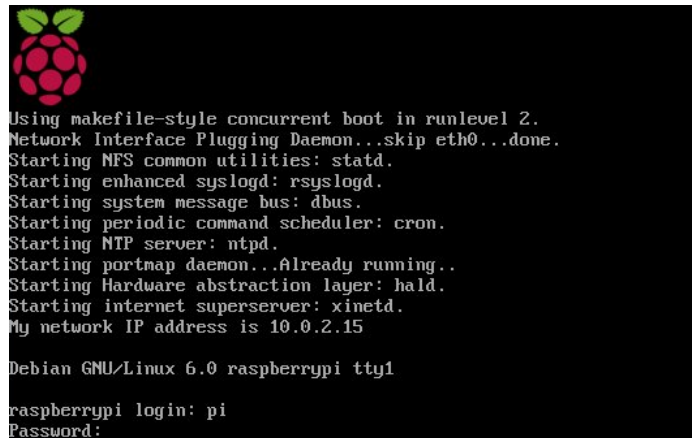
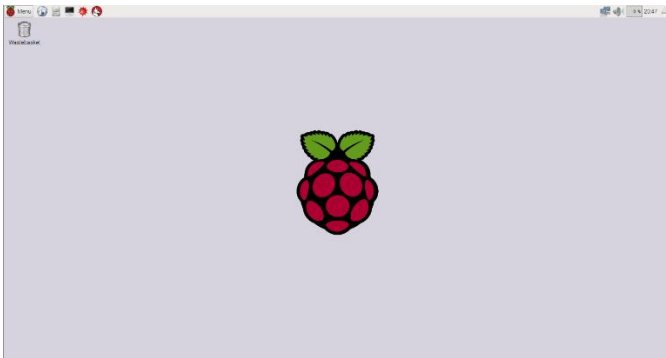
Find your Raspberry Pi.

- Plug in the SD card (or Micro SD on the A+/B+/Pi2) with the Raspbian operating system on it.
- Plug the HDMI/video cable into the Pi and the monitor.
- Plug the keyboard and mouse into the USB ports. You may need a USB hub if you are using a Model A or A+.
- Plug in the power cable.

When all wired up it should look like the one on the right. The A+, B+ and Pi2 will look slightly different.



After the Raspberry Pi has finished starting up, you will see either the Graphical User Interface (as in the image below left) or the ‘terminal’ (as in the image below right). Robots don’t need the GUI running as they are not often connected to monitors when they are driving around, therefore it is suggested that you run your Pi without the GUI. However, at this stage, it doesn’t matter too much and you may prefer to use IDLE (the Python Integrated Development Environment – or editor) within the desktop environment to edit your code.



It is good to keep your Raspberry Pi's operating system up to date with the latest fixes and improvements. You can only do this if your Raspberry Pi is connected to the internet. It may take some time (perhaps up to an hour), so you should only do this when you have time. Type in each of the two commands below, one after the other, leaving each command to complete before starting the next.

```
sudo apt-get update  
sudo apt-get upgrade
```

Identifying your Version of Raspbian

These worksheets have been written to be used with the Raspbian operating system, as supplied by the Raspberry Pi Foundation from their website: <https://www.raspberrypi.org/downloads/raspbian/>. There are two 'versions' of Raspbian available at present – called **Wheezy** and **Jessie**.

Jessie is the latest version, so the code has been written on a Raspberry Pi using that version of the operating system. You can identify which version of the OS you currently have using the command:

```
lsb_release -c
```

in a terminal window. This will return either Jessie (new) or Wheezy (old).

The Raspberry Pi comes with two versions of Python:

- Python 2.7, run using the `python` command
- Python 3.2, run using the `python3` command

When the Raspberry Pi was first released, some of the important Python libraries were only available for Python 2.7. However, almost every library, and all the ones used in these worksheets, are available for Python 3.2. It has been decided that all code for this EduKit will be developed for Python 3.2.

All code will work on all versions of the operating system, but there is one difference that you need to be aware of:

- **Jessie** is able to access the GPIO pins in Python as a standard user. Therefore, to run Python code that includes GPIO, you only need to type the following into a terminal window:
`python3 yourcode.py`
- **Wheezy** requires Python to be run as the privileged 'Super User' if the code accesses the GPIO pins. Therefore, to run Python code that includes GPIO you only need to type the following into a terminal window:
`sudo python3 yourcode.py`

Writing Code

You are now going to create your first small piece of Python code that will simply print “Hello World” to the screen.

First, you are going to create a directory where the code for the EduKit worksheets will be stored. If you are in GUI mode, open a Terminal window by choosing XTerminal from the Menu (under Accessories).

Type in the following commands, pressing the ‘return’ key at the end of each line.

`cd ~` - Changes to your home directory.

`mkdir EduKitRobotics` - Makes a new directory called ‘EduKitRobotics’.

`cd EduKitRobotics` - Changes to the ‘EduKitRobotics’ directory.



If you are in Terminal mode, or if you prefer to use ‘nano’ as your editor, then type the following into the terminal:

`nano 1-helloworld.py` - Opens the ‘nano’ editor with the name ‘1-helloworld.py’



If you prefer to use IDLE as your editor, open ‘Python 3 (IDLE)’ from the Menu (under Programming), and create a file using the IDLE menu item ‘New file’ in the File menu (or use Ctrl+N).

Type in the following code exactly as seen into your preferred editor:

```
#Print Hello World!
print("Hello World!")
```

Everything on the same line after a ‘#’ is a comment and will be ignored by Python.



To save the code in nano, type “Ctrl + x” then “y” then “enter”.

“Ctrl + x” tells nano that you want to exit. It will ask you whether you want to save the file, to which you answer ‘y’ for ‘yes’.

It will then prompt for the name of the file, which was set when you opened the file. Just press the “enter” key to take the default.



If you’re using IDLE, save the file in the EduKitRobotics directory.

Running the Code

Running the code will depend on whether you are running in Terminal or GUI mode.



Make sure you are in the EduKitRobotics directory using the following command:

```
cd ~/EduKitRobotics
```

To run this code type:

```
python3 1-helloworld.py
```

You will see it print “Hello World!” to the terminal window.



In IDLE, select the menu option Run > Run Module, or press F5.
You will see “Hello World!” printed to the Python Shell.

Conventions used in these Worksheets

Throughout these worksheets it is assumed that you are using Raspbian (Jessie), and you are editing your code with ‘nano’, the terminal window text editor. If you use the same, then you can follow the instructions exactly as they appear. If not, then remember the following

- If you have Raspbian (Wheezy), then you need to add the word `sudo` before issuing the `python3` command to run your code from the terminal window. You will NOT be able to run your code from within the IDLE editor.
- If you are using Raspbian (Jessie) and prefer to use the IDLE editor to edit your code, then follow the example in the section above.

Notes

If you are viewing these worksheets on your Raspberry Pi, you should not copy and paste any of the code from the worksheets as the spacing will not necessarily be pasted and the code will not always work. Instead, type it in.

The indentation at the beginning of Python code is important – it is Python’s way of recognising how code should be grouped in ‘conditions’, ‘loops’ and ‘functions’.