CamJam EduKit 2 – Sensors – Introduction

Project Setting up your Raspberry Pi

Objective Write a simple Python program that shows the message “Hello World” on the screen. For this quick activity, you will not be plugging in any of the parts from the CamJam EduKit.

CamJam EduKit 2 Worksheets

CamJam EduKit 2 – Sensors is the second kit in the CamJam EduKit series. You can get the printable activities for free at http://camjam.me/edukit. These kits work with all types of Raspberry Pi computers.

Notes and Assumptions

In these worksheets, we assume you are using the latest version of Raspberry Pi OS and editing your code with *Thonny*, which comes pre-installed with Raspberry Pi OS. The newest version of Python needs a special virtual space, called a ‘venv,’ set up on your Pi. This helps you install and use certain libraries. We will guide you through the simple steps to create this ‘venv’ in this worksheet.

If you are looking at these worksheets directly on your Raspberry Pi, do not copy and paste the code from the worksheets. The spaces at the start of lines might not copy correctly, which can cause the code not to work properly. The spaces at the beginning are important because Python uses them to understand how different pieces of code are connected, like in ‘if’ statements, loops, and functions. Instead, try typing out the code yourself or download it from GitHub using the instructions at the end of this worksheet.

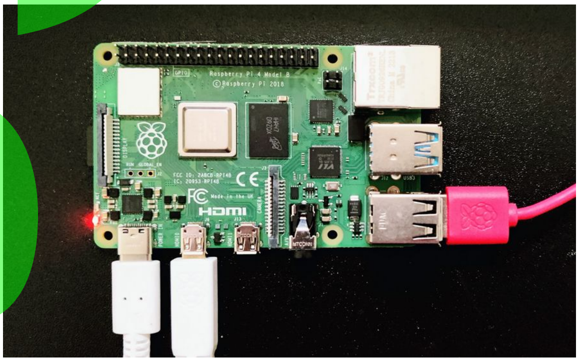
Equipment Required

For this EduKit, you will need:

- Your Raspberry Pi
- A microSD card (or SD card on very old Raspberry Pi’s); 16GB is recommended. It should have the Raspberry Pi OS with Desktop pre-installed. You can find instructions for setting up the Raspberry Pi OS on the Raspberry Pi website (https://www.raspberrypi.org/downloads/).
- A monitor and an HDMI cable to connect it to your Raspberry Pi.
- A keyboard and mouse.
- A power supply for your Raspberry Pi.
- CamJam EduKit 2, which you can buy from The Pi Hut website (https://thepihut.com/edukit).

Setting up your Raspberry Pi

Find your Raspberry Pi.



- Insert the microSD card into the slot (or use the SD card on older Models A and B).
- Connect the HDMI cable from the Raspberry Pi to your monitor.
- Connect your keyboard and mouse to the USB ports. If you're using a Pi Zero, you'll need an adapter for the keyboard and mouse. If your Raspberry Pi has only one USB port, you'll need a USB hub to connect both.
- Plug in the power cable to turn it on.
- Once everything is connected, it should look similar to the picture above, depending on which Raspberry Pi you have.

Updating the Raspberry Pi operating system

It is a good idea to update your Raspberry Pi's software regularly to get the newest fixes and improvements. To do this, your Raspberry Pi needs to be connected to the internet.

Open a terminal by clicking on the icon in the menu bar that looks like this:

Or by choosing Accessories → Terminal from the menu bar at the top, or use the key combination Ctrl + Alt + T (hold down the Ctrl button, the Alt button and press T).

Enter the two commands below, one after the other, making sure to finish each one with a carriage return before starting the next.

```
sudo apt update
sudo apt upgrade
```

The Right Way to Write Python Programs

The people who develop Python have created a way to keep your Python programs separate and safe from each other by using what is known as a *Python Virtual Environment*. Each Virtual Environment, or 'venv' can be changed without affecting the other programs that use Python on your computer. This is especially important when you use 'libraries', which are programs other people have written that make your own code easier to write.

We are going to use this proper way of developing Python so that you are familiar with it, although it might be a little confusing at first. You can make as many of these Virtual Environments as you want on your computer, and can install different versions of the same library in different 'venv's.

It might sound complicated, but it's actually pretty simple to set up and manage.

Let's create your virtual environment you are going to use for this EduKit now. To start, open a terminal window by clicking on the icon on the menu bar that looks like this:

(Or press the keys Ctrl + Alt + T)

Type the following:

```
python -m venv ~/EduKit2venv
```

This command creates a new ‘virtual environment’, using a Python tool called 'venv'. It places this new space inside a folder named 'EduKit2venv' in your main folder. If you just run Python now, it won't use this new space yet. To start using this new environment, you need to turn it on by activating’ it with:

```
source ~/EduKit2venv/bin/activate
```

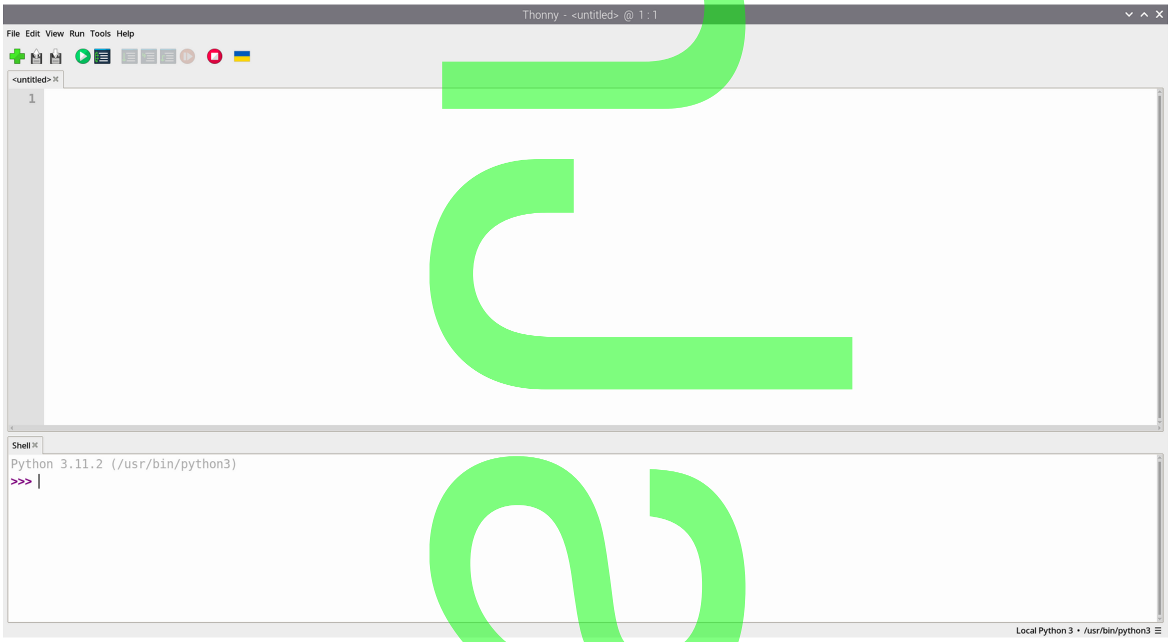
This tells your computer to use the virtual environment ‘EduKit2venv’ you have just created to run your Python code. You will see "(EduKit2venv)" before your username and your Pi's name on the terminal prompt line.

Whenever you want to run your code inside this virtual environment, just remember to activate it. If you do not see the name of the virtual environment at the start of your command line, you’ll need to turn it on first using the source command again:

```
source ~/EduKit2venv/bin/activate
```

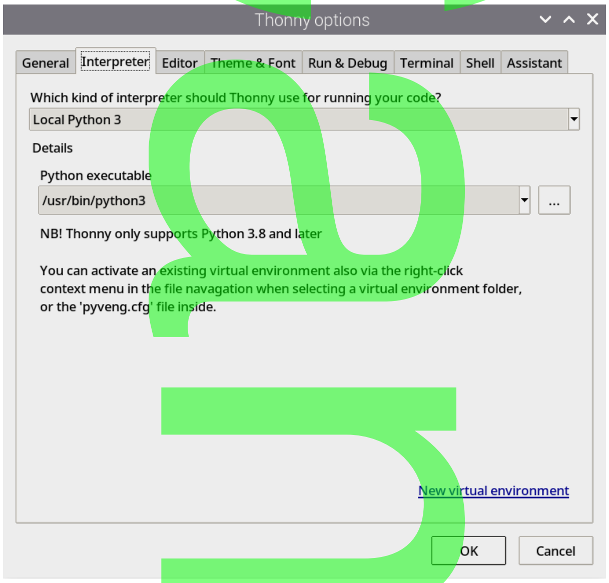
Configuring Thonny to use the Venv

By default, Thonny will use the main Python setup used across the Raspberry Pi, not the virtual environment you created. To switch to the virtual environment, open 'Thonny' from the Menu (under Programming). You should see something like the example below:


Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
Shell
Local Python 3 • /usr/bin/python3

```
Python 3.11.2 (/usr/bin/python3)
>>> |
```

At the bottom right of the Thonny window, you'll see the words 'Local Python 3 - /usr/bin/python'. This shows the current Python setup for the Raspberry Pi. Click on this text with your mouse, then choose 'Configure Interpreter…'. A new window with options will appear.



Which kind of interpreter should Thonny use for running your code?
Local Python 3

Details

Python executable
/usr/bin/python3

NB! Thonny only supports Python 3.8 and later

You can activate an existing virtual environment also via the right-click context menu in the file navagation when selecting a virtual environment folder, or the 'pyveng.cfg' file inside.

Click on the three dots next to the Python Executable box. Then find and select the venv you made earlier, which should be in the folder `/home/pi/EduKit2venv/bin/python`. After that, click OK.

From now on, the code you run in Thonny will use your Python virtual environment.

Getting your Raspberry Pi ready to use the sensors from CamJam EduKit 2

Before you can use some of the sensors in EduKit 2, you need to make some simple setting changes and install some Python libraries.

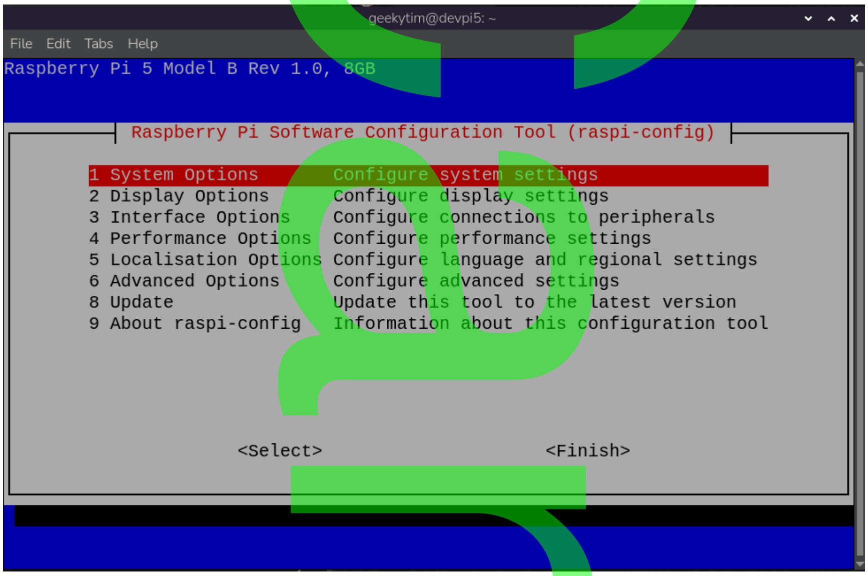
How to Set Up GPIO Zero to use the Thermometer

The thermometer supplied with CamJam EduKit 2 uses a method of communicating called the '1-Wire' protocol. Before you can use it you need to tell your Raspberry Pi to load code that enables it to understand this protocol.

To do this, open a terminal window by pressing the Ctrl + Alt + T key combination. Type:

```
sudo raspi-config
```

This will open the Raspberry Pi configuration tool:



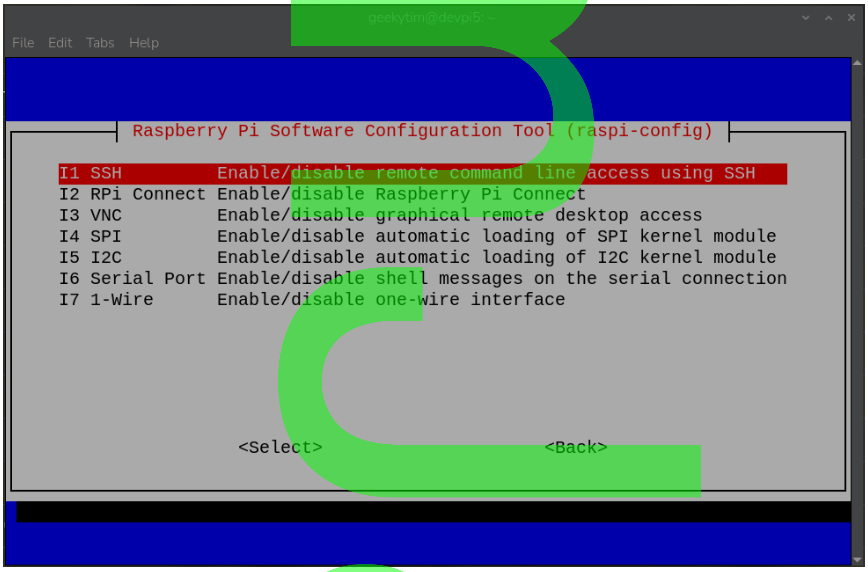
```
Raspberry Pi 5 Model B Rev 1.0, 8GB

        Raspberry Pi Software Configuration Tool (raspi-config)

   1 System Options       Configure system settings
   2 Display Options      Configure display settings
   3 Interface Options    Configure connections to peripherals
   4 Performance Options  Configure performance settings
   5 Localisation Options Configure language and regional settings
   6 Advanced Options     Configure advanced settings
   8 Update               Update this tool to the latest version
   9 About raspi-config   Information about this configuration tool

              <Select>                    <Finish>
```

Using the arrow keys, scroll down to the Interface Options and press Enter to reveal the interface options:



```
        Raspberry Pi Software Configuration Tool (raspi-config)

 I1 SSH         Enable/disable remote command line access using SSH
 I2 RPi Connect Enable/disable Raspberry Pi Connect
 I3 VNC         Enable/disable graphical remote desktop access
 I4 SPI         Enable/disable automatic loading of SPI kernel module
 I5 I2C         Enable/disable automatic loading of I2C kernel module
 I6 Serial Port Enable/disable shell messages on the serial connection
 I7 1-Wire      Enable/disable one-wire interface

              <Select>                    <Back>
```

In the Interface Options, scroll down to 1-Wire and press Enter again. Using the Left and Right arrow keys, highlight 'Yes' and press Enter. The 1-Wire interface will be enabled, so press enter again, and use the right arrow to move to Finish and press enter to exit the configuration tool. You will be asked if you want to reboot, which you should to to complete the configuration change.

You need to add a Python library called *W1ThermSensor*, which is used in worksheet 3. Make sure your Raspberry Pi is connected to the internet. Then, open a terminal and make sure it is using the virtual environment (see above). Then type the following command:

```
python -m pip install w1thermsensor
```

The *w1thermsensor* Python library can now only be used when running your code inside that specific virtual environment.

Writing Code

You're about to write your first small piece of Python code that will simply show “Hello World” on the screen.

Open ‘Thonny’ from the Menu (under Programming) and create a new file by clicking ‘New’ in the File menu or pressing Ctrl+N. Make sure it is using your virtual environment.

Type exactly this code into your editor:

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

Anything after the ‘#’ is just a note for you and doesn’t affect the program.

Save the file and name it *1-helloworld.py*.

Running Code

To start your program, choose the menu option "Run" and then "Run Module," or just press the F5 key. You will see the message "Hello World!" appear in the Python Shell.

Getting the EduKit Code from GitHub

The code made for CamJam EduKits, shown in the worksheets, is also saved on GitHub. Follow these steps to download all the EduKit code.

GitHub Repository

You can find all the EduKit files and collections online at https://github.com/CamJam-EduKit. From there, you can download single files or entire groups of files.

Installing Git

Before you can copy the code from GitHub, make sure you have the Git program installed on your Raspberry Pi. To do this, your Raspberry Pi needs to be connected to the internet. Open the terminal window and start by updating the list of available software with this command:

```
sudo apt update
```

Then install GIT using:

```
sudo apt install git-core
```

Downloading to the Raspberry Pi

You can get the EduKit code and worksheets by following this simple step:

```
cd ~
git clone https://github.com/CamJam-EduKit/EduKit2.git
```

The code will be available in the "code" folder inside the "EduKit2" folder.