# Lab 1: Introduction to Raspberry Pi 3B with MATLAB/Simulink

## Objectives

1. To control Raspberry Pi 3B with algorithms written in MATLAB.
2. To control Raspberry Pi 3B with Simulink models.
3. To control Raspberry Pi 3B wirelessly.

## Software and Hardware

| Software | Hardware |
|---|---|
| 1. MATLAB R2019a<br>2. Simulink R2019a | 1. Host PC<br>2. Raspberry Pi 3B IoT kit<br>3. USB Network Adaptor<br>4. Ethernet cable |

## Introduction

Raspberry Pi hardware is a low-cost and general-purpose computing board, running on a Debian-based Linux operating system. It can be also configured as an embedded system incorporating sensors and actuators. MATLAB and Simulink Support Package (for Raspberry Pi Hardware) provides an interface for accessing Raspberry Pi's computing power and I/O peripherals through the MATLAB command line. Therefore, from the MATLAB interface, you can directly control the board and collect data from sensors connected to the board.

In this course, you will prototype your IoT devices using Raspberry Pi 3B. In summary, the board is comprised of 1.2GHz Broadcom 64bit CPU and 1GB RAM, with in-built WIFI and Bluetooth Low Energy.

For more information, visit the following links:

https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
https://www.youtube.com/watch?v=gbJB3387xUw
https://au.mathworks.com/hardware-support/raspberry-pi-matlab.html

## Prelab work: understanding Raspberry Pi 3B

Visit the URL links in Introduction, therefore identify key hardware components, and state key specifications of Raspberry Pi 3B board, as shown in Figure 1.

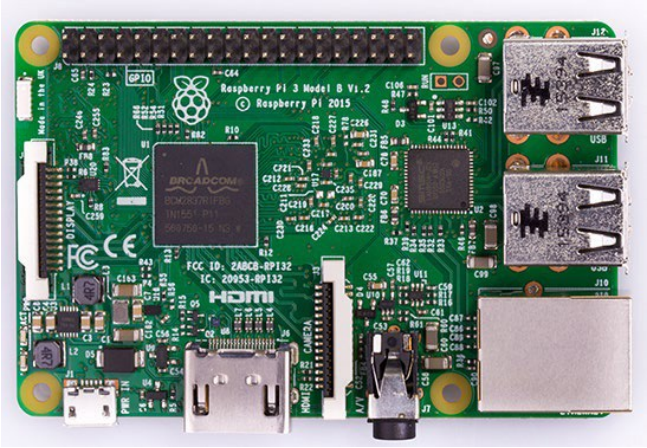


Fig 1: Raspberry Pi 3B Board [1]

# Experiment 1: Raspberry Pi with MATLAB [2]

## A. Setup Board

The latest firmware has been prepared and written to the SD card. In addition, the case for the board and the extension board (Sense HAT) have also been assembled in advance. So to get started on the board, follow the procedures:

1. Insert Firmware SD card to the board, if it has not been already done.
2. Connect the board to host PC with an Ethernet cable and a USB network adaptor.
3. Plug in the power supply.

## B. Establish a Connection

Start MATLAB R2019a. To establish a connection between MATLAB and the board, enter the following command to MATLAB command window

```
myrpi = raspi('169.254.0.2','pi','raspberry');
```

This command requests the MATLAB to connect to a server running on the Raspberry Pi hardware through TCP/IP, and then return a raspi object named as myrpi. Therefore the board can be controlled and programmed through the raspi object in MATLAB.

**Task:**

- Explore the raspi object: myrpi, and recognise the IP address and port number of the board.
- Also Identify basic peripherals of the board such as digital pins.

## C. First Algorithm

The following code will instruct the board to switch on or off led0 as shown in Figure 2.

```
writeLED(myrpi, 'led0', 1); %switch on led0
writeLED(myrpi, 'led0', 0); %switch off led
```

To visualise the location of led0, execute the following command.

```
showLEDs(myrpi);
```

**Task:** write a MATLAB program that can make the LED blink with a period of 1 second for a total of 20 seconds.

Hint: use ***for*** loop with 20 iterations, and ***pause*** function. You can consult MATLAB Help Documents on how to use them.

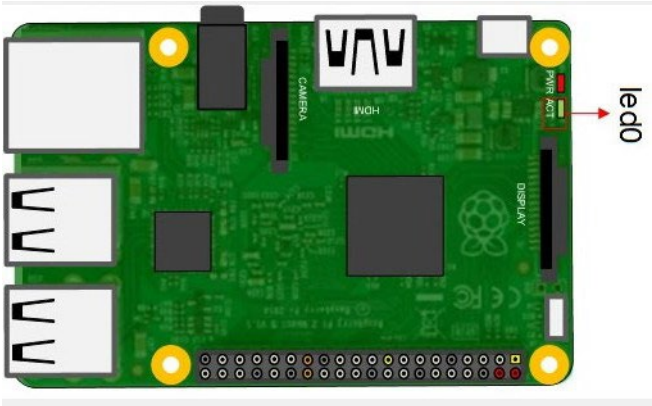



Fig 2: Led0 on the board

## D. Terminal Session

To interact with Linux OS on the board, you can open a terminal session with the board by entering the following command to MATLAB command window.

```
openShell(myrpi)
```

A terminal window will appear and you can execute Linux commands there.

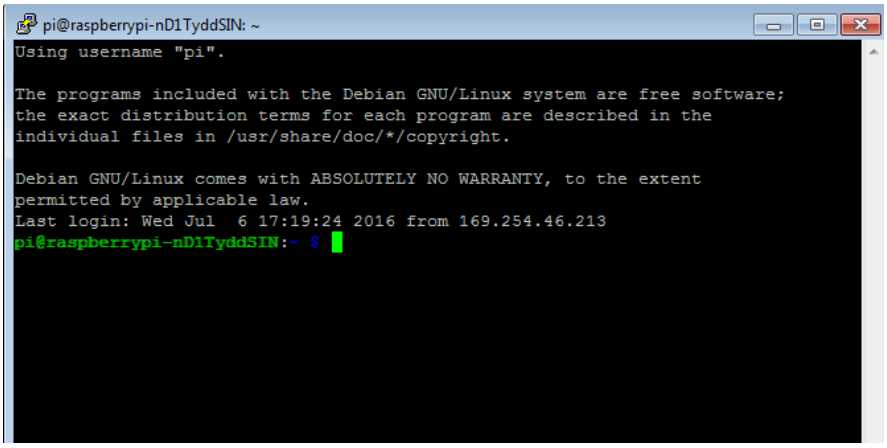



Fig 3: Terminal Session

For example, you can display all the files within the directory /home/pi/ by execute in the terminal.

```
ls -l /home/pi/
```

The output will be returned to the terminal. Alternatively, you can execute Linux commands directly through MATLAB with the following:

```
system(myrpi, 'ls -l /home/pi/')
```

The output will be displayed in the MATLAB Command Window. You are expected to see the same results.

### E. Shutdown the board

There is no a shutdown button on the board so the correct procedure is not to unplug the power cable. Instead, you shall execute the shutdown instruction in either the MATLAB command window or the terminal session.

The following MATLAB command will shut down the system.

```
system(myrpi, 'sudo shutdown -h now');
```

Alternatively, you can enter the Linux command in the terminal session.

```
sudo shutdown -h now
```

**Wait until led0 stops flashing, and then unplug the power cable to complete the shutdown procedures.**

Later you can turn on the board by plugging in the power cable.

## Experiment 2: Raspberry Pi 3B with Simulink [3]

### A. Setup Board

Similarly, to get the board started with Simulink, make sure:

1. Firmware SD card is inserted to the board.
2. Connect the board to host PC with an Ethernet cable and a USB network adaptor.
3. Plug in the power supply.

### B. Configuration and Preparing the First Model

Open the Simulink Model lab1_raspi_demo.slx. The following window will pop up. Explore each block and understand what it does.

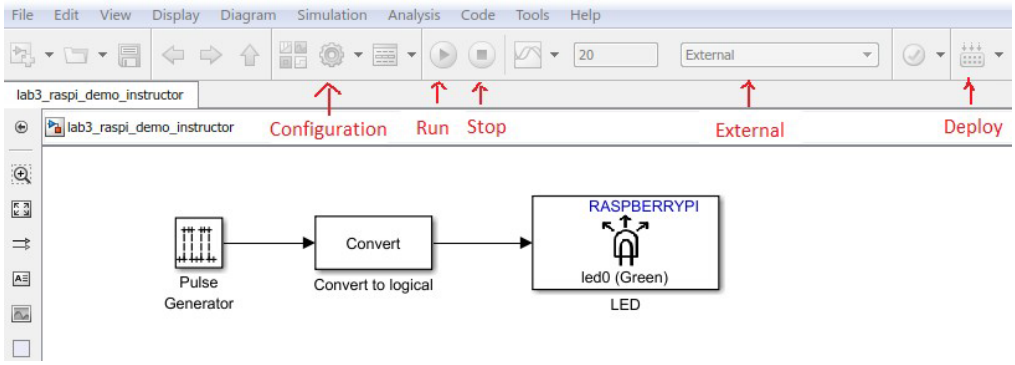



Fig 4: Simulink Model Window

Click **Model Configuration Parameters** Icon at the Icon Bar, leading to a configuration window as shown in Figure 5. Make sure that correct settings are entered as follows.

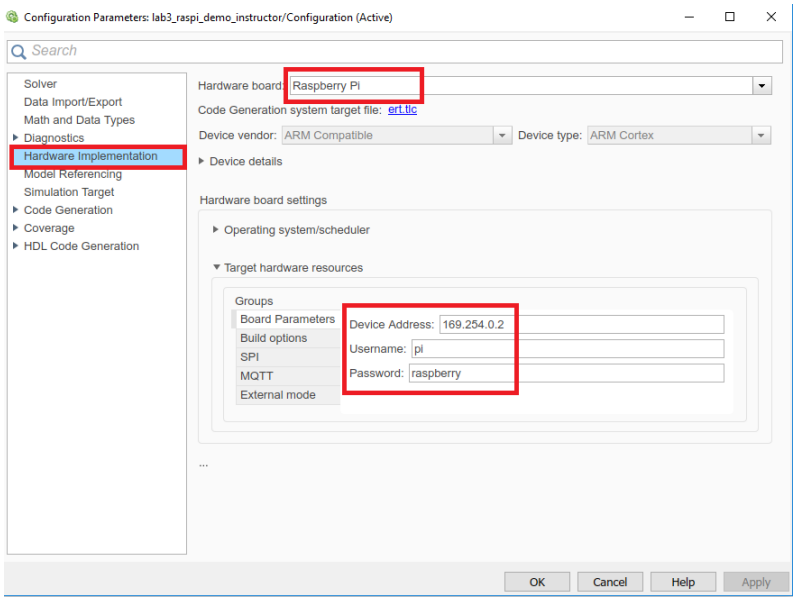



Fig 5: Configuration Parameters

**Task:** You are to modify the Simulink Model so that it can make the LED blink with a period of 1 second for a total of 20 seconds. Open the **Pulse Generator** Block as shown in Figure 6. Assuming the sample time is 0.1 second, what value will you enter to Period and Pulse width?

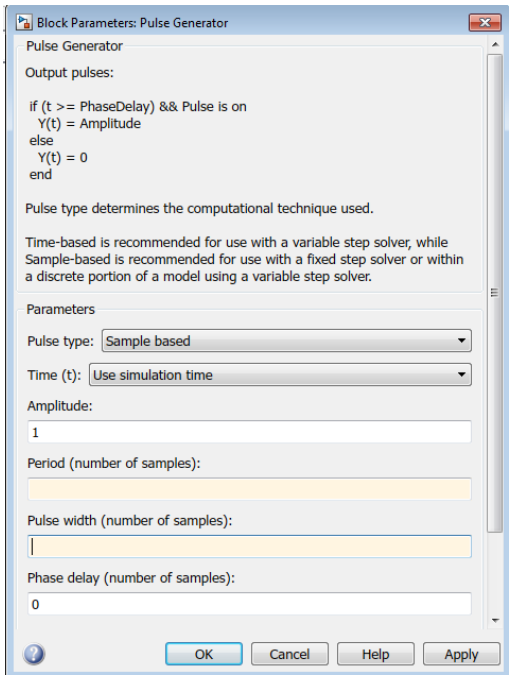



Fig 6: Pulse Generator Block

## C. Run

The model can be run on the board in either **External Mode** or **Deploy Mode**.

- **External Mode:** The block diagram act as a user interface to the board. You can change parameters in the Simulink blocks, which will result in changing parameters in the real-time application.
- **Deploy Mode:** run the model as a standalone application on the board.

The procedures are as follows:

1. First **Run** the Model in the **External** Model (Ref to Figure 4). What do you observe on the LED as well as the Simulink model window? Then **Stop** the model by pressing the Stop Icon.
2. **Deploy** the Model by pressing Deploy Icon (Ref to Figure 4). What do you observe on the LED as well as the Simulink model window?

In External mode, you can stop the Model running on the board by simply pressing the Stop Icon on the Simulink model window. However in Deploy mode, you need to execute the following commands in the MATLAB command window.

```
r = raspberrypi; %%create Simulink Rpi object
isModelRunning(r,'lab3_raspi_demo') %% check if the model is running
stopModel(r,'lab3_raspi_demo') %% stop the model
```

You can also restart the model by the following commands.

```
runModel(r,'lab3_raspi_demo') %% stop the model
```

**Task**: demonstrate to your tutors that you can start and stop the models running on the board in either **External Mode** or **Deploy Mode.**

## Experiment 3: Wireless Configuration [4]

It is inconvenient to have an Ethernet Cable all the time when prototyping smart devices with Raspberry Pi. In this part, you are going to configure wireless connection between the host PC and the board.

1. First keep the cable between the PC and the board. Open a terminal session from MATLAB

   ```
   myrpi = raspi('169.254.0.2','pi','raspberry');
   openShell(myrpi);
   ```

2. In the terminal session, use the following command to scan WiFi networks. From the returned results, can you find the network called ELEC5518_Lab522? Hint: check ESSID.

   ```
   sudo iwlist wlan0 scan
   ```

3. Open the `wpa-supplicant` configuration file in nano with the following command:

   ```
   sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
   ```

4. Go to the bottom of the file and add the following lines (note that ssid and psk of wireless network will be provided in class, the lines below are only displayed as an example).

   ```
   network={
      ssid="ELEC5518_Lab522"
      psk="RaspberryPi3"
        }
   ```

   And save the file and exit by pressing Ctrl+X and then Y followed by Enter.

5. Restart the network interface with the following command

   ```
   wpa_cli -i wlan0 reconfigure
   ```

6. In the terminal session opened at step 1, check if the wireless connection has been established by

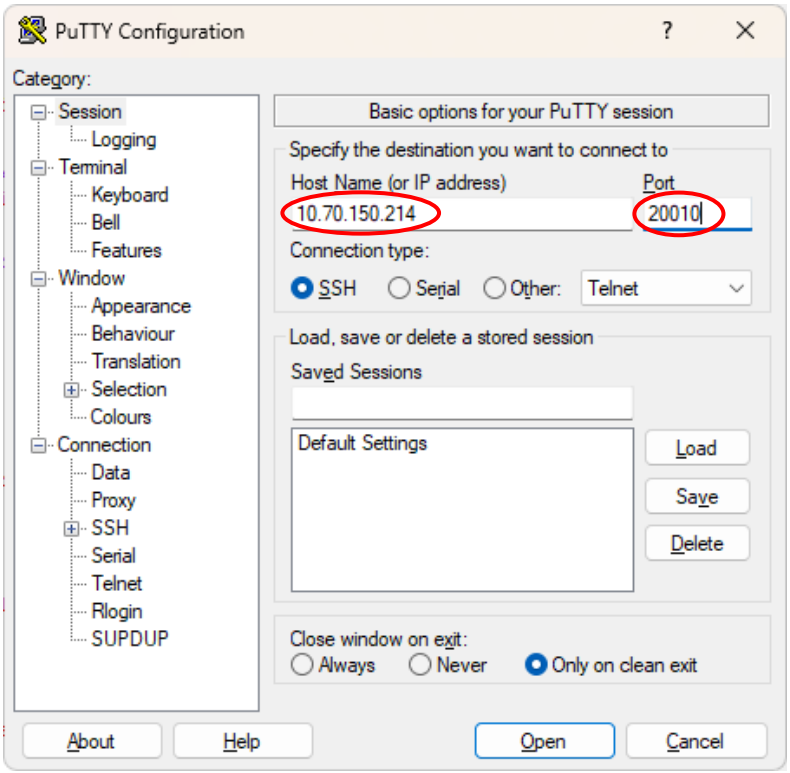
   ```
   ifconfig wlan0
   ```

And you will see similar inet addr info as below:

```
wlan0     Link encap:Ethernet  HWaddr b8:27:eb:06:50:ba
          inet addr:192.168.3.2    Bcast:192.168.3.255  Mask:255.255.255.0
          inet6 addr: fe80::2de9:2983:9ce3:c9c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:102 errors:0 dropped:90 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:35156 (34.3 KiB)  TX bytes:4359 (4.2 KiB)
```

7. **Remove the Ethernet cable, and the board is connected via WiFi only now.** Close the terminal session opened at Step 1.
8. Click the magnifier icon next to the Microsoft Windows icon at your PC desktop as below, and search "Putty".

9. Type in the IP address and Port as below. Please confirm with your tutor for IP address. If your Raspberry Pi Kit is marked as '1', the port should be **20001**, '2' for **20002**…(The The port at below screenshot is for Kit **10**).




10. Click "Open" button, and type *pi* when being asked user name and *raspberry* for password.

**Task**: demonstrate to your tutors that you can execute the following Linux commands after successfully logging in.

```
ls -l /home/pi/
```

And shutdown the device at the end of the experiment by

```
sudo shutdown -h now
```

## Reference

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[2] https://au.mathworks.com/help/supportpkg/raspberrypiio/examples/getting-started-with-matlab-support-package-for-raspberry-pi-hardware.html
[3] https://au.mathworks.com/help/supportpkg/raspberrypi/examples/getting-started-with-raspberry-pi-hardware.html
[4] https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md