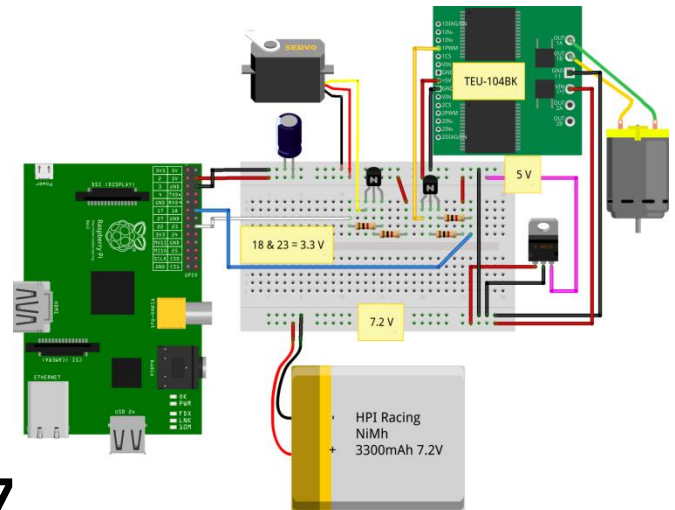
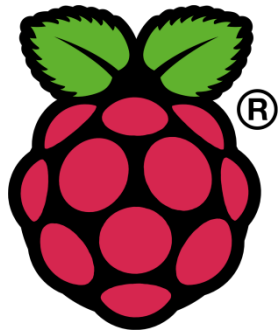


# Arduino BOE kit and Raspibot Board

## Class 5: RaspiRobot Board Motor Controls



August 11, 2017  
Don Wilcher

# RaspiRobot Board Motor Controls

## Agenda:

- Building a small mobile robot with RaspiRobot board
- What is a LED Matrix?
- The Adafruit LED Backpack
- Attaching the LED Matrix to the RaspiRobot Board
- Configuring I2C port for the Raspberry Pi
- Controlling a LED Matrix with RaspiRobot Board

# Building a small mobile robot with RaspiRobot board



**Project notes taken from Simon Monk's Github site**

## raspirobotboard3

---

Python library and design files for the RasPi Robot Board V3

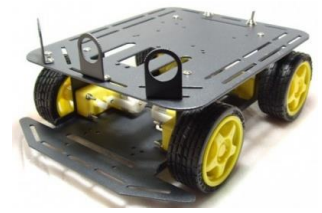
If you were looking for the library for the version 2 board go to <https://github.com/simonmonk/raspirobotboard2>

Rusty Gereard has produced a Java library for the RRB3 - <https://github.com/R-Gerard/RRB4J>

### Source:

<https://github.com/simonmonk/raspirobotboard3>

# Building a small mobile robot with RaspiRobot board...



## Installing the Python Libraries

To install the library, issue the following commands

```
$ cd ~  
$ git clone https://github.com/simonmonk/raspirobotboard3.git  
$ cd raspirobotboard3/python  
$ sudo python setup.py install
```

Attach the RRB3 to your Raspberry Pi. You do not need to attach batteries, motors or anything else to the RRB3 just yet. For now you can just power it through the Pi's normal USB power connector.

Lets run some tests from the Python Console now that everything is installed. We can experiment with the RRB3, even without any motors

Open a Python console (Python2 not 3) by typing the following into a Terminal window:

```
$ sudo python
```

# Building a small mobile robot with RaspiRobot board...



Then, within the python console, type the following, one line at a time:

```
from rrb3 import *  
rr = RRB3(9, 6)  
rr.set_led1(1)  
rr.set_led1(0)  
rr.set_led2(1)  
rr.set_led2(0)  
rr.sw1_closed()
```

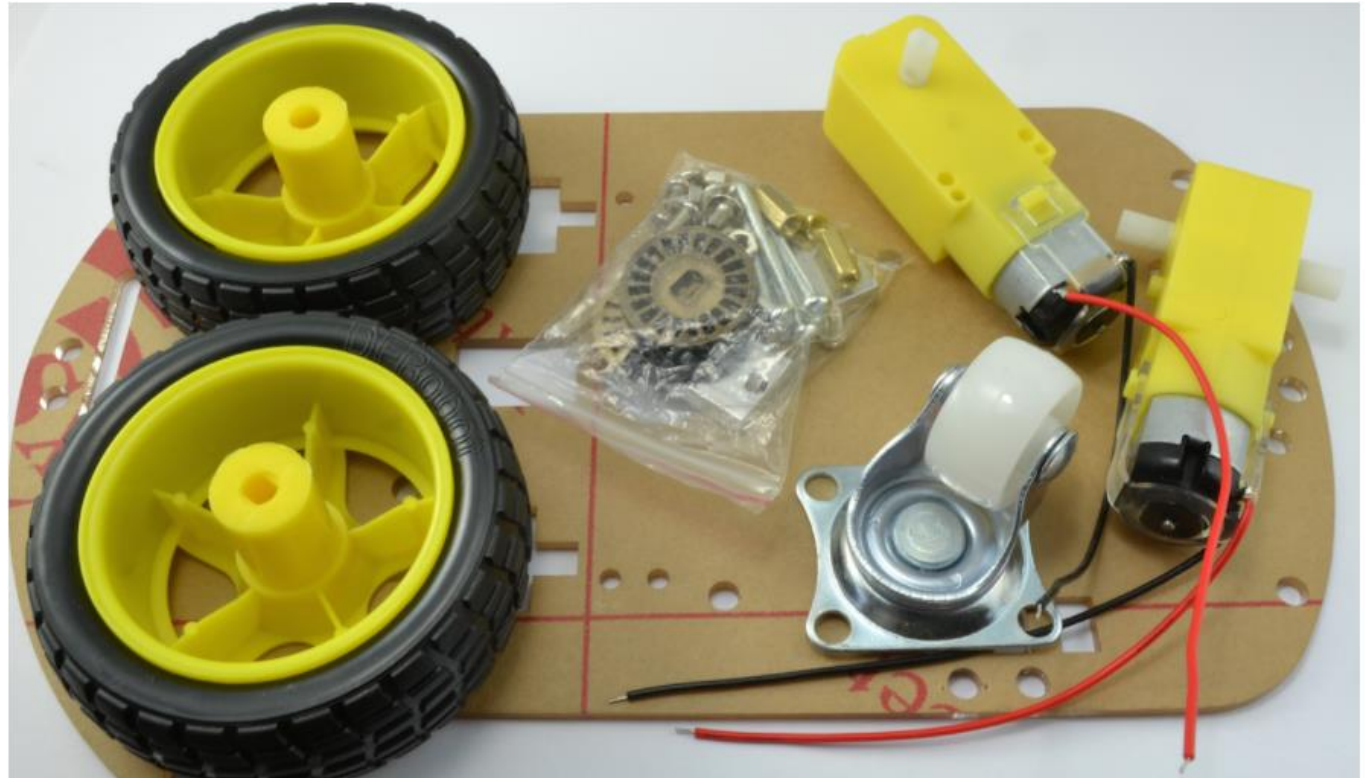
The last step should display the answer "False" because no switch is attached.

If you prefer, you can use True and False in place of 1 and 0 in the examples above.

# Building a small mobile robot with RaspiRobot board...



## Mobile robot kit



## Mobile robot kit Instructions:

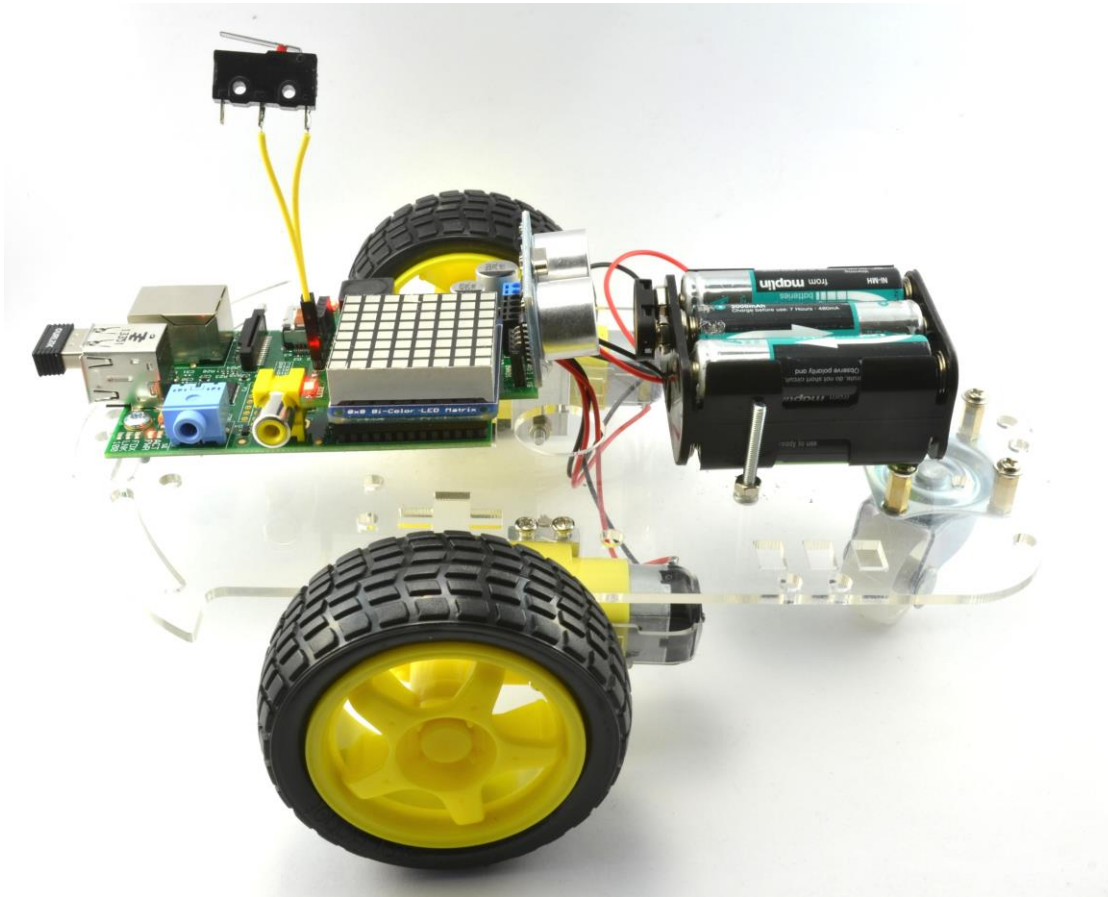
<https://github.com/simonmonk/raspirobotboard2>

# Building a small mobile robot with RaspiRobot board...



Mobile robot kit

Assembled kit



# Building a small mobile robot with RaspiRobot board...



## API Reference

---

### General

---

The library implements a class called RRB3. This is only available for Python 2 and any Python programs that you write that use the library must be run as a super user.

To import the library and create an instance of the class, put this at the top of your Python program.

```
from rrb3 import *  
rr = RRB3(9, 6)
```

The first parameter '9' is the battery voltage (6 x 1.5V AA batteries). The second parameter ('6') is the motor voltage (6V for most low cost robot chassis motors). It is important to set these values correctly, as the library will manage the voltage supplied to the motors, to prevent them burning out or running too fast.

The rest is pretty straightforward, there are just a load of useful methods on the class that you can use.



# Building a small mobile robot with RaspiRobot board...



## LEDs

---

There are two LEDs built-in to the RaspiRobotBoard, called LED1 and LED2. Both of these can be turned on and off using the following methods:

To turn LED1 on just do:

```
rr.set_led1(1)
```

To turn it off again do:

```
rr.set_led1(0)
```

To control LED2 just do the same thing but using set\_led2.

# Building a small mobile robot with RaspiRobot board...



## Switch Inputs

The `sw1_closed()` and `sw2_closed()` functions return true if the contacts for that switch are closed. By default, the switches are open. You can test out closing the switch by shorting the two contacts with a screwdriver.

The following test program will show you the state of each of the switch contacts.

```
from rrb3 import *

rr = RRB3()

while True:
    print("SW1=" + str(rr.sw1_closed()) + " SW2=" + str(rr.sw2_closed()))
    raw_input("check again")
```

# Building a small mobile robot with RaspiRobot board...



## Open Collector Outputs

---

The RRB3 has two high-power open collector outputs. These can each source up to 2A and so are suitable for driving loads at the battery voltage, such as high power LEDs, IR senders, alarm bells, relays etc.

To turn the Open Collector OC1 output on just do:

```
rr.set_oc1(1)
```

To turn it off again do:

```
rr.set_oc1(0)
```

To control OC2, substitute set\_oc2 in place of set\_oc1 in the examples above

# Building a small mobile robot with RaspiRobot board...



## Motor (High Level Interface)

There are two levels of command for controlling the motors. There is a high level interface that assumes that the motors are connected to wheels on a rover. These commands are forward, reverse, left, right and stop.

```
rr.forward()
```

... will start both motors running in the same direction to move the robot rover forwards. They will continue in this direction until another command is issued.

If you want to move forward for a certain amount of time, you can specify a number of seconds as an optional first argument. If you supply a second parameter between 0 and 1 this will control the speed of the motor. This is set to 0.5 as a default. If you want the motors to run indefinitely, but also want to control the speed, then use 0 as the first parameter.

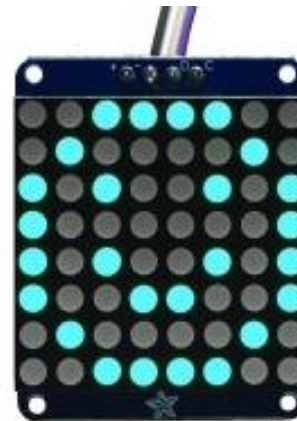
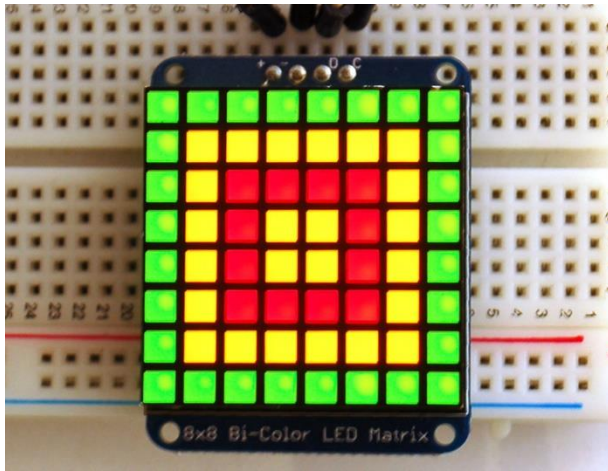
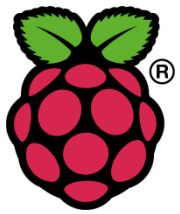
Some examples:

```
rr.forward()      # forward half speed indefinitely
rr.forward(5)     # forward for 5 seconds at half speed
rr.forward(5, 1)  # forward for 5 seconds at full speed
```

The commands left, right and reverse all work in the same way.

The stop command stops all the motors.

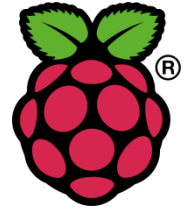
# What is a LED Matrix?



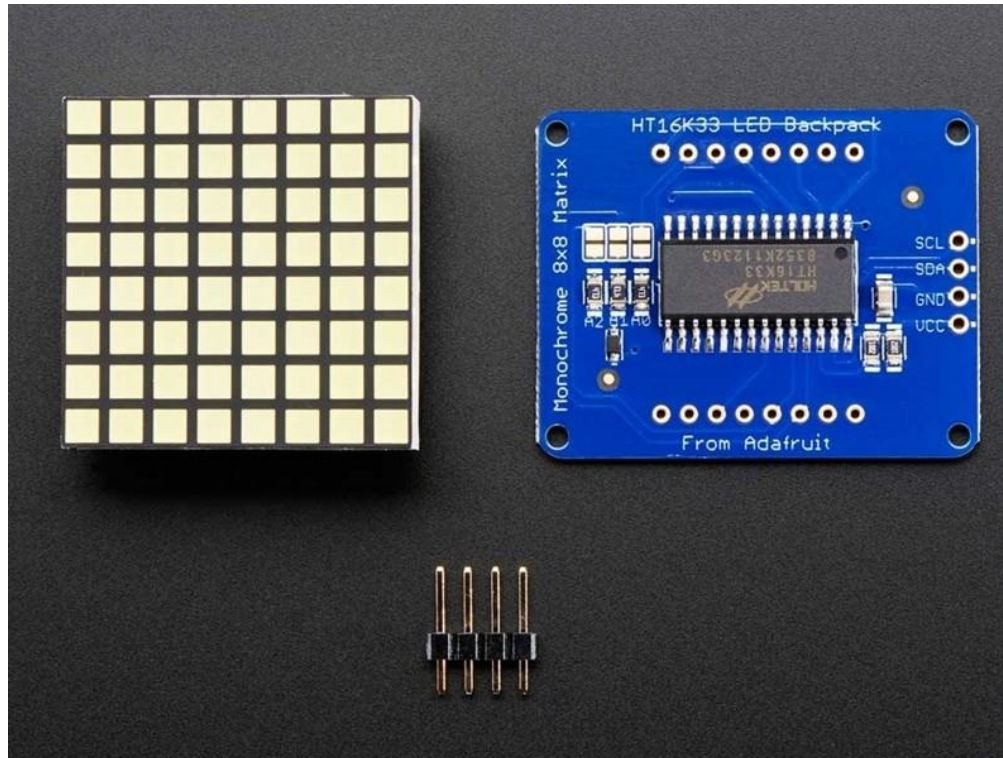
Presented by:

# Question 1

**What is a LED Matrix?**



# What is a LED Backpack?



**Resource :**

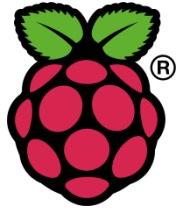
<https://learn.adafruit.com/adafruit-led-backpack/overview>

# Question 2

**What advantage does the LED Backpack provide for the LED Matrix?**



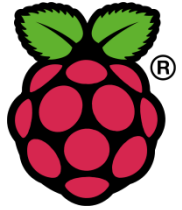
# Attaching the LED Matrix to the RaspiRobot Board?



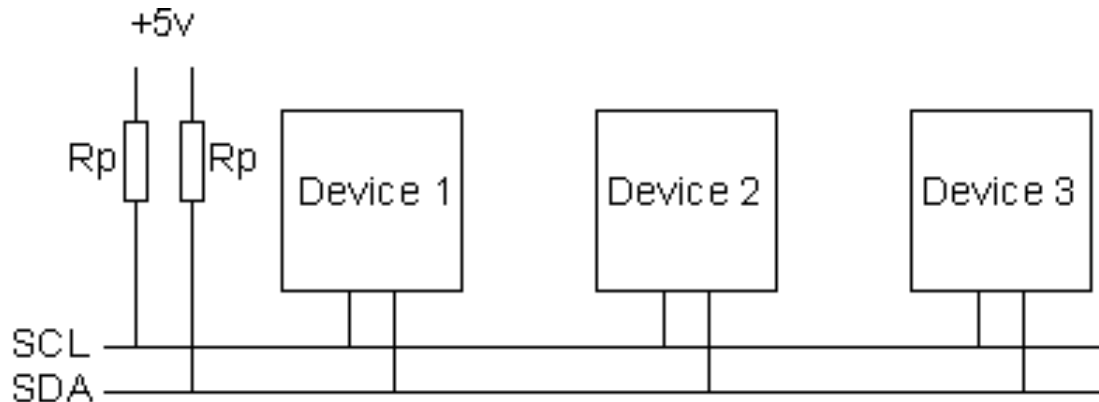
# Question 3

**What connector on the RaspiRobot Board does the LED Matrix mate to?**

# Configuring I2C on the Raspberry Pi



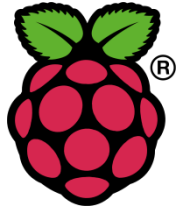
I2C (**Inter-Integrated Circuit**) is a very commonly used standard designed to allow one chip to talk to another. So, since the Raspberry Pi can talk I2C we can connect it to a variety of I2C capable chips and modules



**Configuring I2C tutorial resource:**

<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>

# Configuring I2C on the Raspberry Pi...



## Adafruit Python LED Backpack

Python library for controlling LED backpack displays such as 8x8 matrices, bar graphs, and 7/14-segment displays on a Raspberry Pi or BeagleBone Black.

Designed specifically to work with the Adafruit LED backpack displays ---->  
<https://learn.adafruit.com/adafruit-led-backpack/overview>

For all platforms (Raspberry Pi and Beaglebone Black) make sure your system is able to compile Python extensions. On Raspbian or Beaglebone Black's Debian/Ubuntu image you can ensure your system is ready by executing:

```
sudo apt-get update
sudo apt-get install build-essential python-dev
```

You will also need to make sure the python-smbus and python-imaging library is installed by executing:

```
sudo apt-get install python-smbus python-imaging
```

Install the library by downloading with the download link on the right, unzipping the archive, navigating inside the library's directory and executing:

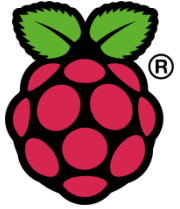
```
sudo python setup.py install
```

See example of usage in the examples folder.

[https://github.com/adafruit/Adafruit\\_Python\\_LED\\_Backpack](https://github.com/adafruit/Adafruit_Python_LED_Backpack)

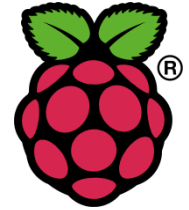
Presented by:

# Question 4



**What does I2C stand for?**

# Controlling the LED Matrix with RaspiRobot Board



Open an editor window using:

```
nano scrolling_clock.py
```

Then paste in the following text and save the file using CTRL-X and then Y and then ENTER.

```
import time
from datetime import datetime
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
from Adafruit_LED_Backpack import Matrix8x8

display = Matrix8x8.Matrix8x8()
display.begin()

font = ImageFont.truetype("/usr/share/fonts/truetype/freefont/FreeSansBold.ttf", 9)

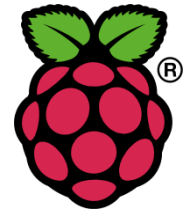
im = Image.new("1", (8, 8), "black")
draw = ImageDraw.Draw(im)
width, ignore = font.getsize("88 : 88 : 88")

def format_time():
    d = datetime.now()
    return "{:%H : %M : %S}".format(d)

message = format_time()
x = 8
while True:
    x = x - 1
    if x < -(width + 20):
        x = 8
    message = format_time()
```

## Scrolling Clock code

# Controlling the LED Matrix with RaspiRobot Board



## Scrolling Clock code...

```
draw.rectangle((0, 0, 7, 7), outline=0, fill=0)
draw.text(x, -1, message, 1, font=font)
display.set_image(im)
display.write_display()
time.sleep(0.1)
```

You can now run the program using the command:

```
sudo python scrolling_clock.py
```

The time should now slowly scroll across the display.

### Resource :

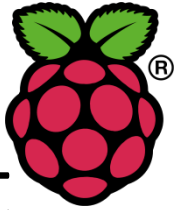
<https://learn.adafruit.com/raspirobot-battery-powered-raspberry-pidisplays>

# Question 5

**What Python library is used to establish the timing for the LED Matrix Scrolling Clock?**



# Additional Resources



Simon Monk's github site for RaspiRobot  
documentation

<https://github.com/simonmonk/raspirobotboard2>

<https://github.com/simonmonk/raspirobotboard3>

Adafruit: RaspiRobot board product

<https://www.adafruit.com/product/1940>

Adafruit: LED Backpack overview

<https://learn.adafruit.com/adafruit-led-backpack/overview>

Adafruit: Raspberry Pi LED Display tutorial

<https://learn.adafruit.com/raspberry-pi-led-matrix-display/overview>