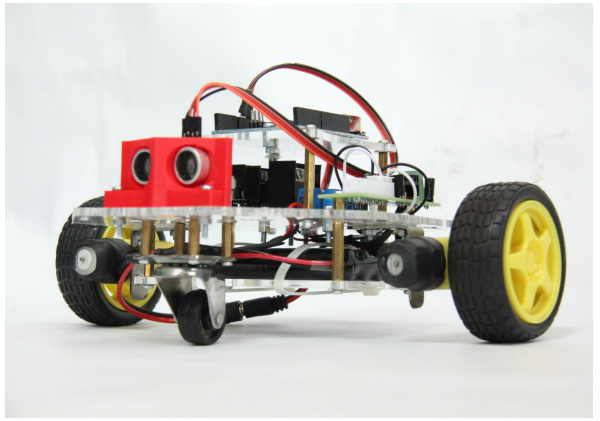
LearnOBots
Think Explore Make

Programming Robots with Python

Shamyl Bin Mansoor

Co-founder & CTO LearnOBots



PyCon Pakistan 2017

- Is an EdTech company developing educational technology tools
- Promotes Science Technology Engineering Arts & Mathematics (STEAM) in Pakistani Schools
- Developed a Technology based school curriculum and educational kits for learning to Code, Make Robots, Electronics, Astronomy etc etc
- Kids projects have been featured by Hackaday / MIT

Me?

- Teach CS at NUST
- Make “things” at LearnOBots
- Love making / playing with Technology / Hardware / Gadgets
  - Was part of the recent CERN Hackathon
  - Work has been featured on Hackaday
  - Hate publishing research papers, but still have a few

LearnOBots
Think Explore Make

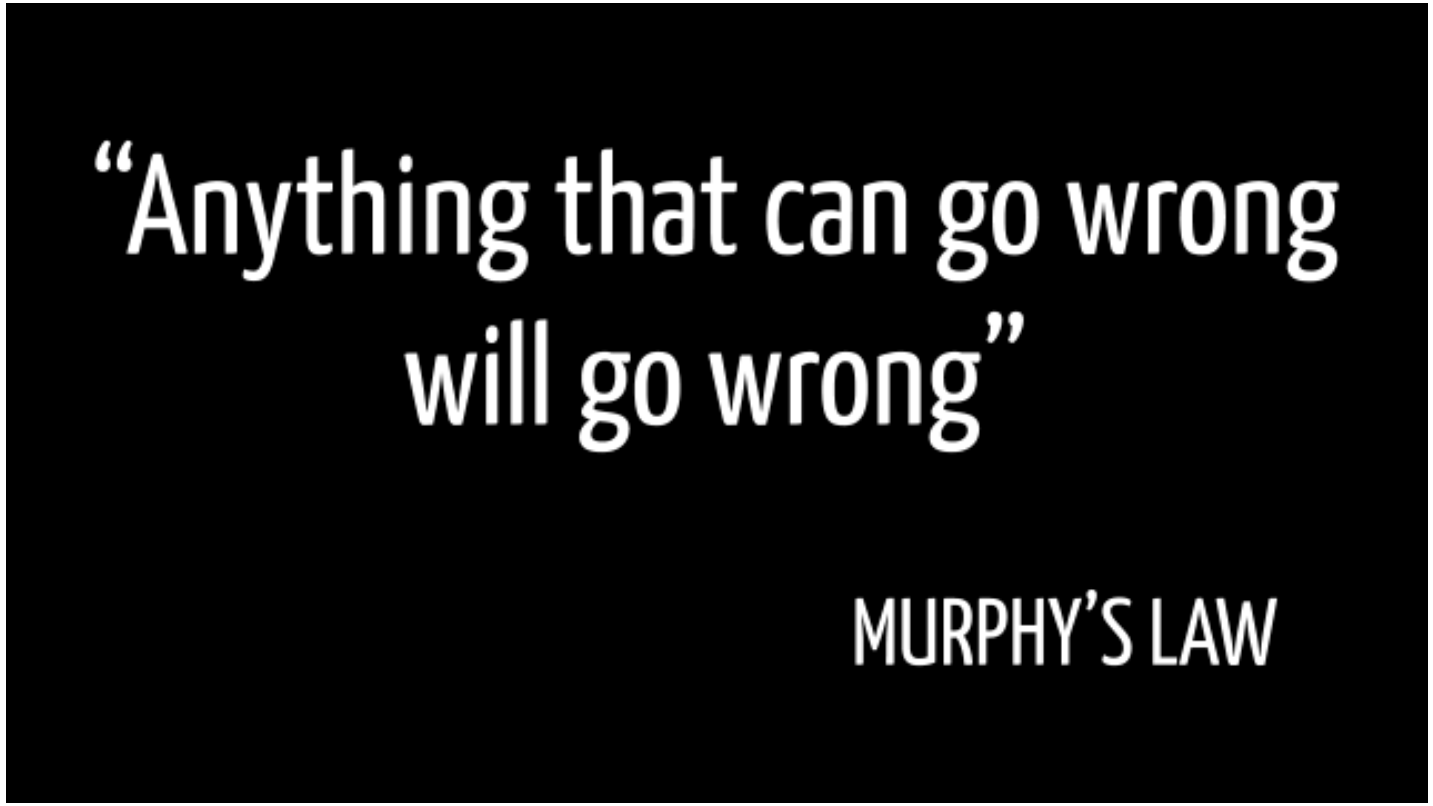
Outline

- Introduction
  - Python for Hardware
  - MicroPython
  - Motivation
- Programming Robots with Python
  - Python on the Raspberry PI
  - Accessing GPIOs on the PI
  - Sensors and Actuators interfacing
  - Programming Robot Movement
- Conclusion

PyCon Pakistan 2017

Disclaimer

- Hardware prototype never works when you show it to someone!



LearnOBots
Think Explore Make

Python for Hardware

- Programming hardware was hard!
  - 8051, C / Assembly
  - High Learning Curve
- Suitable for Hardware?
  - Easy to use
  - Packages available
  - Raspberry PI
- What about Real time requirements?
  - Trade off
  - Use a dedicated Microcontroller for Real time requirements
- Who uses python with hardware?

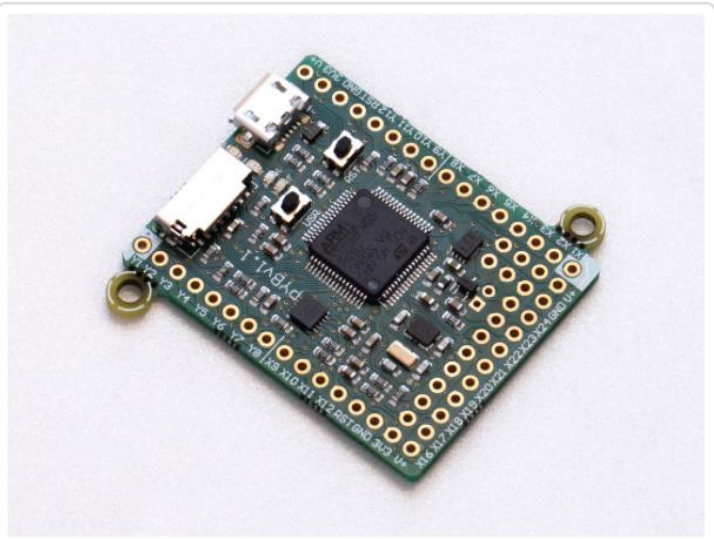
PyCon Pakistan 2017

LearnOBots
Think Explore Make

MicroPython

MicroPython

- Lean and efficient implementation of Python3
- Small subset of Python Standard Library
- Optimized to run on microcontrollers



MicroPython pyboard v1.1

PyCon Pakistan 2017

LearnOBots
Think Explore Make

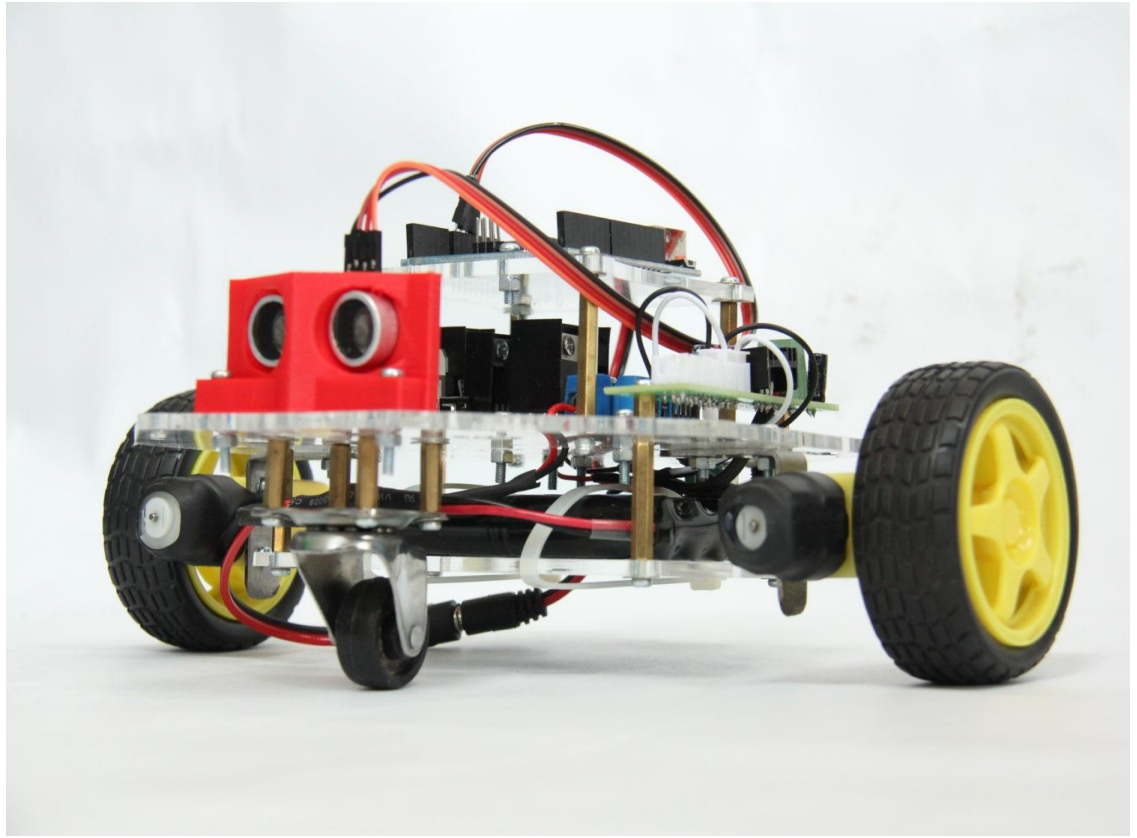
Why Robots with Python?

- Python is easy!
- Robots are Fun!
- Lots of Libraries!

LearnOBots
Think Explore Make

python™

+



=

Fun

PyCon Pakistan 2017

Robots: What do we need?

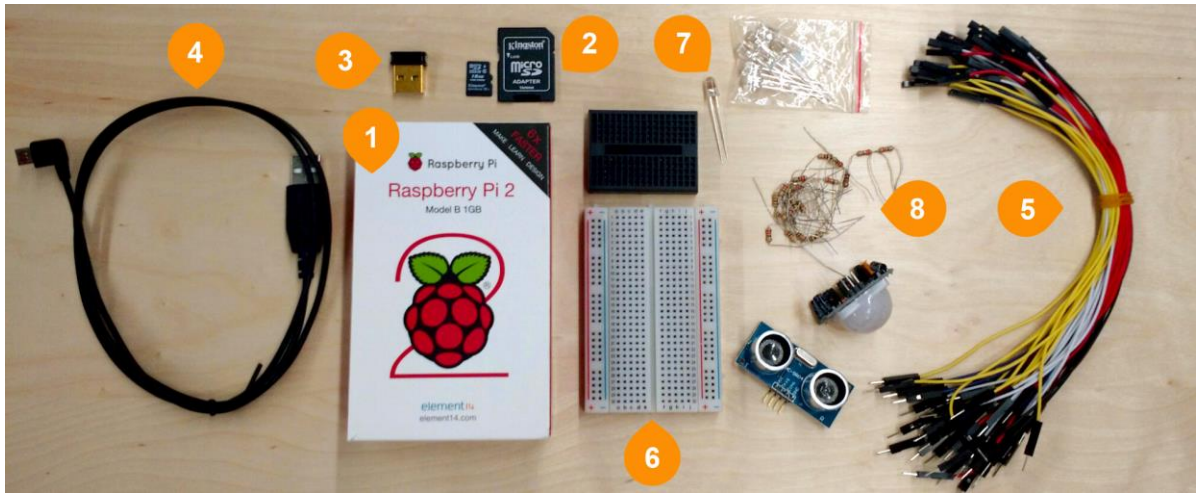
- A brain (Processor / Controller)
  - Raspberry PI or an Arduino
- Some Actuators
  - Motors (DC or Servo)
- Some Sensors
  - Sonar / Depth / Vision

What we will Use?

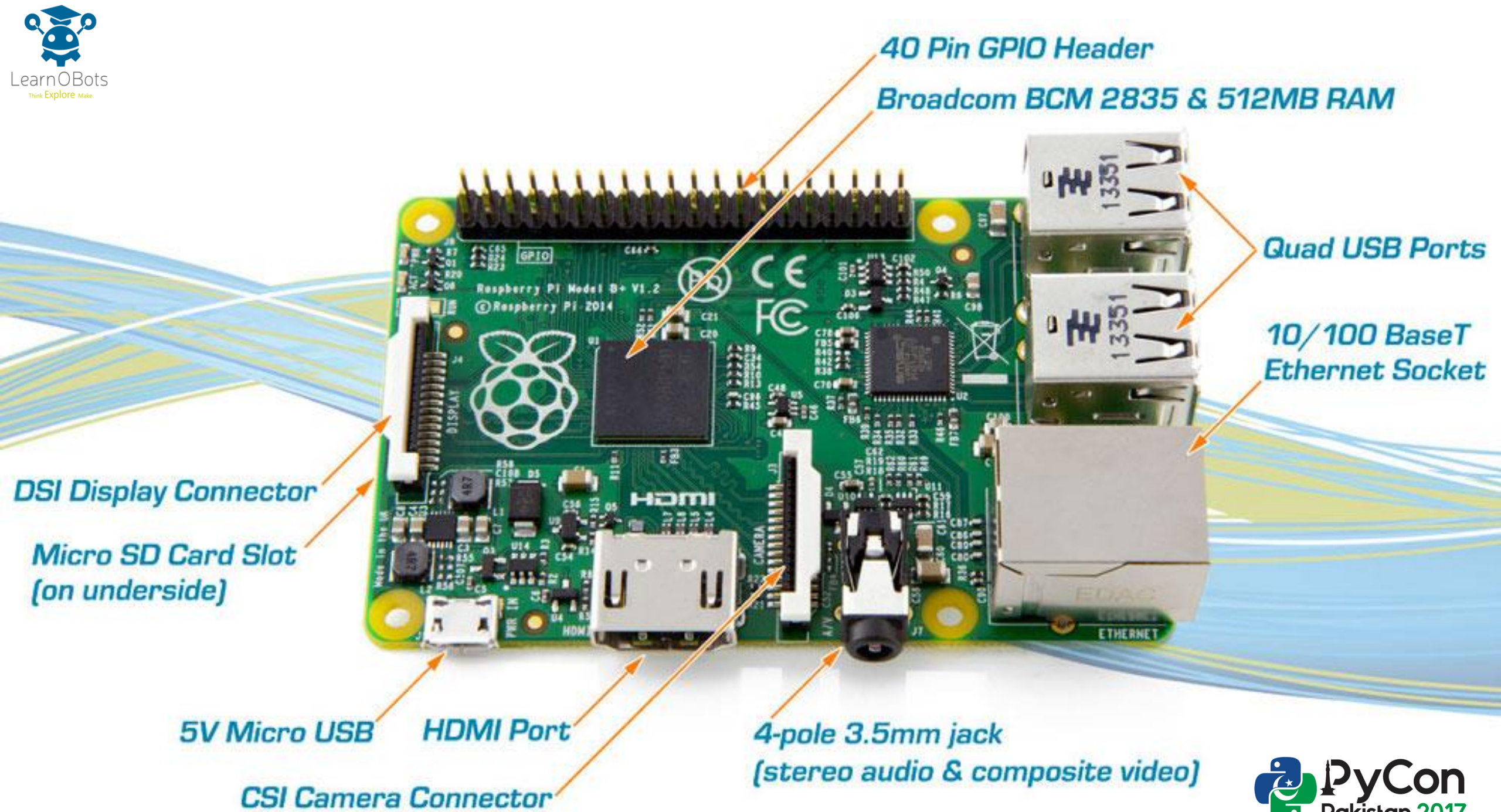
- Raspberry PI since we like Python
- Motors (DC, two of them)
- Sonar Sensor (For sensing obstacles)

Getting started with Raspberry PI

- Buy one! (instock.pk or ewall.com.pk)
- Download Raspbian Image (A Debian based OS for the Raspberry PI)
  - www.raspberrypi.org
- Copy Image to an SD Card
- Plug in Raspberry PI in a TV, connect keyboard and mouse and you're good to go!



LearnOBots
Think Explore Make

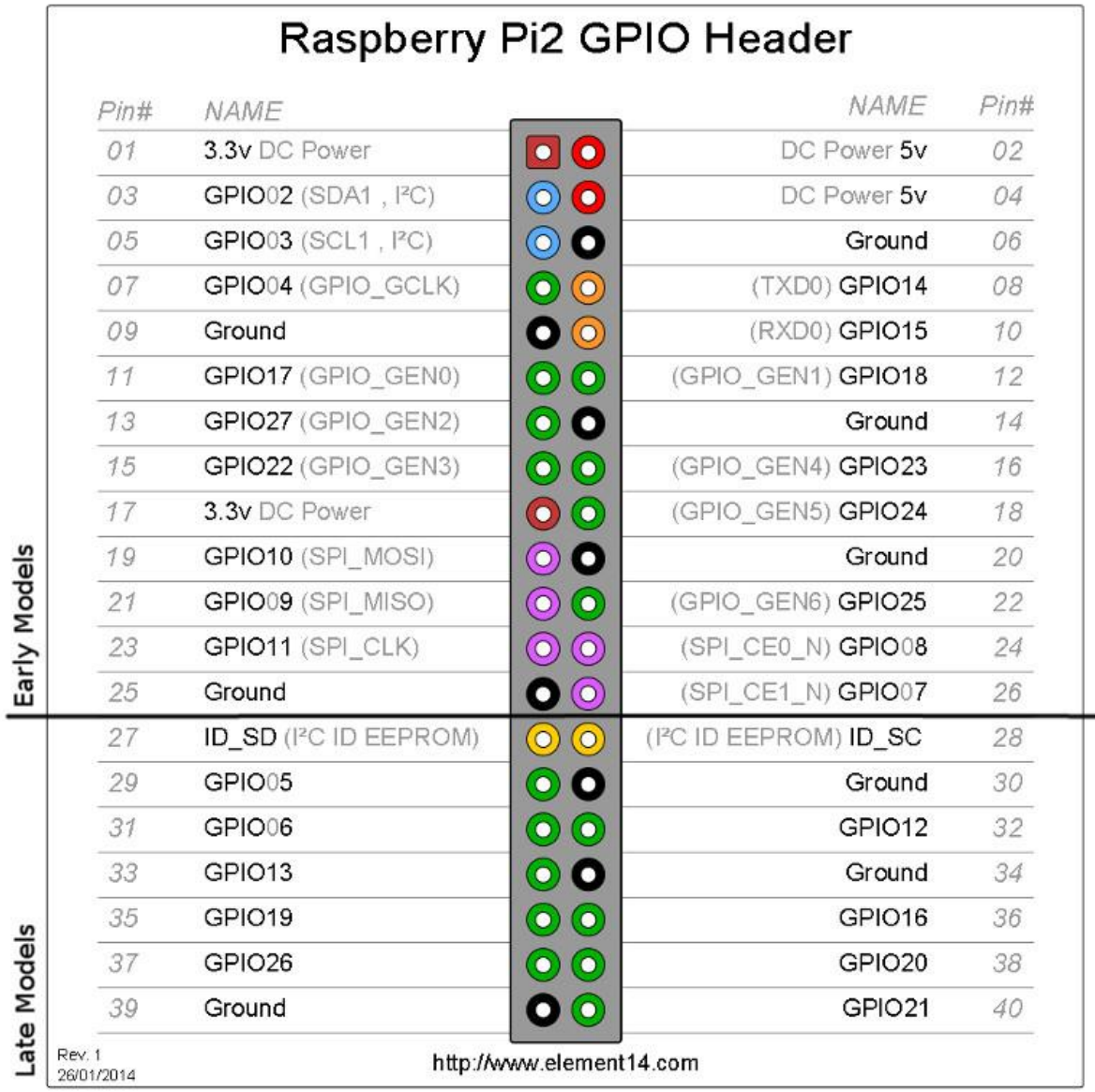


PyCon Pakistan 2017

LearnOBots
Think Explore Make

Accessing GPIO Pins

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)
GPIO.setup(24, GPIO.OUT)
```



PyCon Pakistan 2017

LearnOBots
Think Explore Make

Python on Raspberry PI

- IDLE and Python Come pre-installed on Raspbian
- We will use our favourite editor to write python scripts

PyCon Pakistan 2017

LearnOBots
Think Explore Make

Configuring VNC

- Install VNC viewer from https://www.realvnc.com/ on your MAC or Windows
- Raspbian already comes pre-installed with VNC
- You can connect to your PI over the same network or over the internet
  - Need to create an account with VNC
- Follow https://www.realvnc.com/en/connect/docs/raspberry-pi.html#raspberry-pi-connect-cloud to connect to Raspberry PI

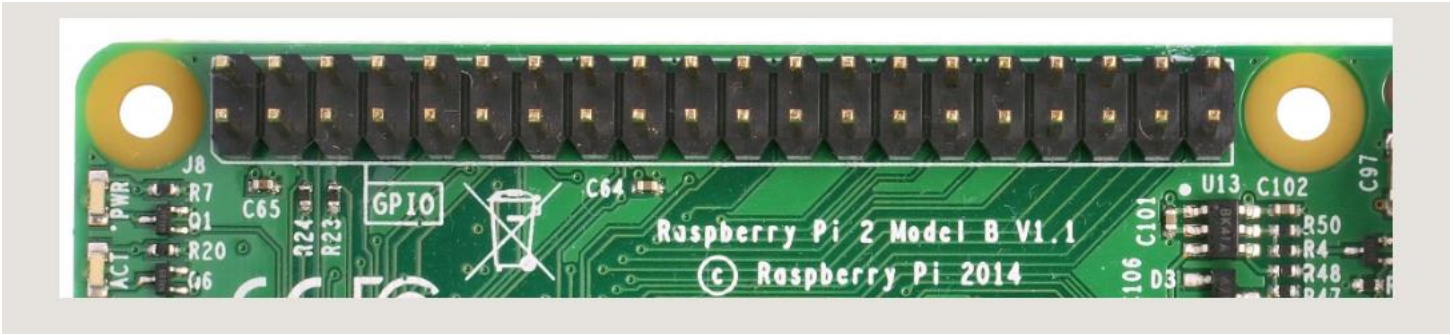
LearnOBots
Think Explore Make

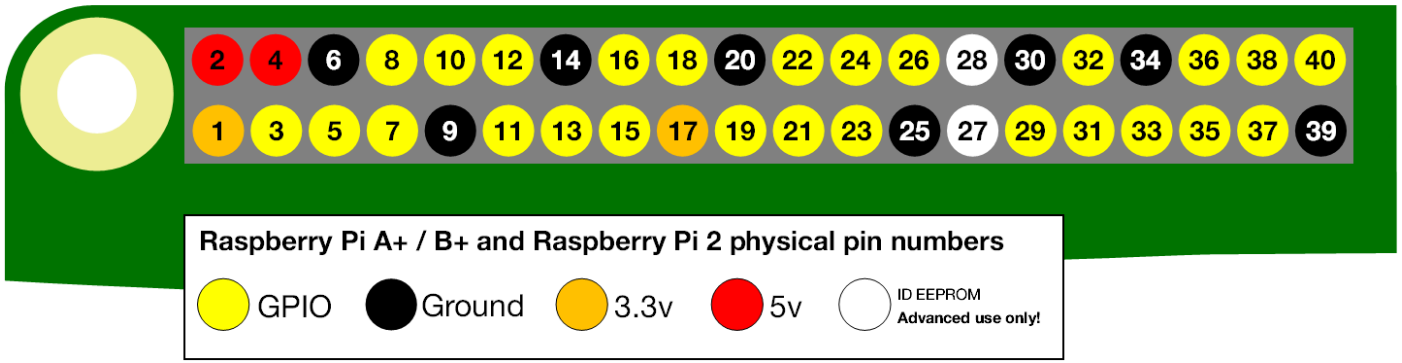
Task 1: Hello World

- Blinking an LED
  - Connect an LED to IO Port of Raspberry PI
  - Make it Blink
  - We will use pin 11 (GPIO 17)
  - And pin 9 (Gnd)

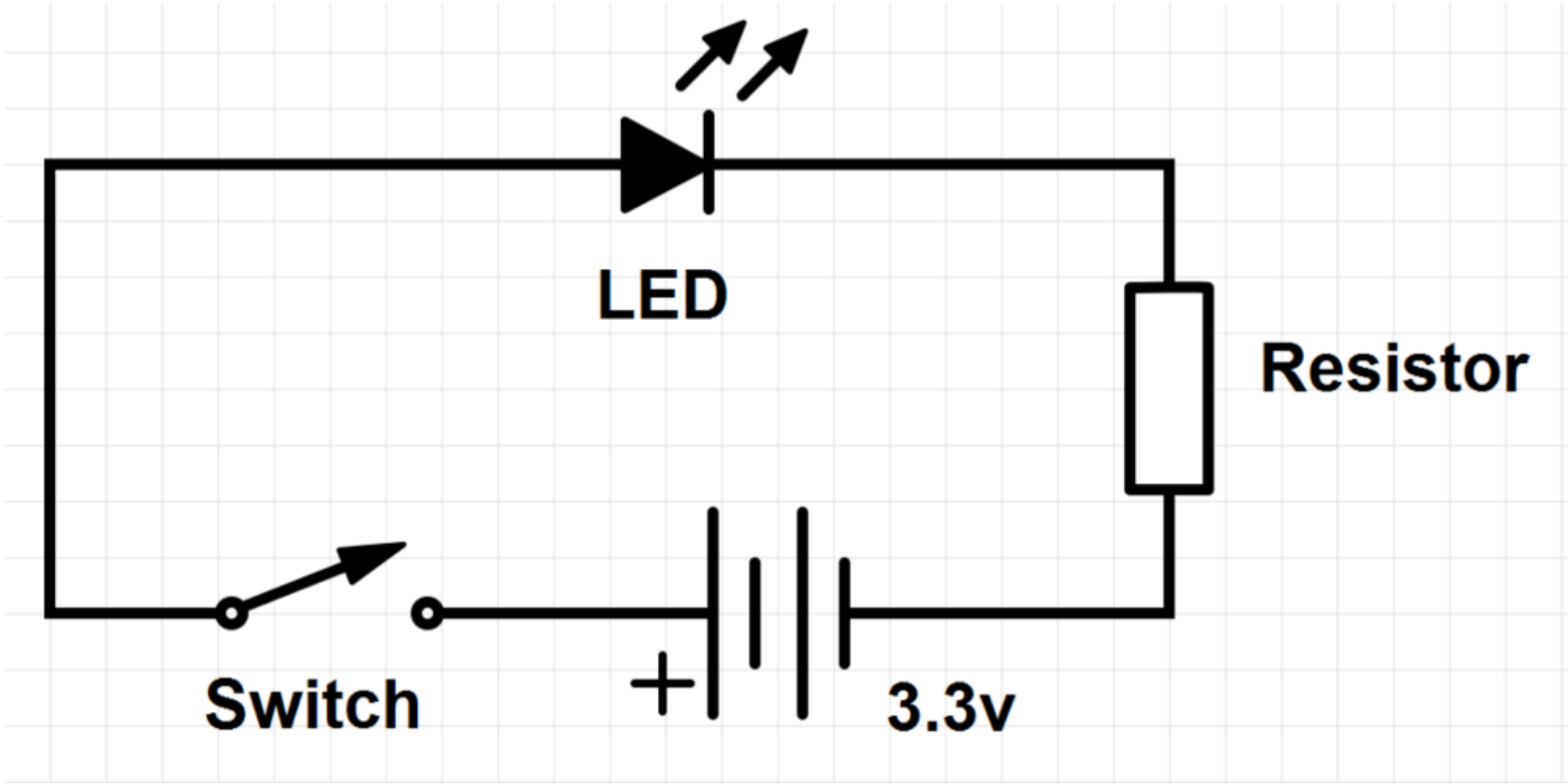
PyCon Pakistan 2017

GPIO Pins


GPIO
Raspberry Pi 2 Model B V1.1
© Raspberry Pi 2014


2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
Raspberry Pi A+ / B+ and Raspberry Pi 2 physical pin numbers
GPIO
Ground
3.3v
5v
ID EEPROM
Advanced use only!

LED Circuit



What's in the Code?

```
#Blinking an LED
import RPi.GPIO as GPIO
import time

LedPin = 11    # pin11

def setup():
  GPIO.setmode(GPIO.BOARD)       # Numbers GPIOs by physical location
  GPIO.setup(LedPin, GPIO.OUT)   # Set LedPin's mode is output
  GPIO.output(LedPin, GPIO.HIGH) # Set LedPin high(+3.3V) to turn on led
```

BLINK : HELLO WORLD!

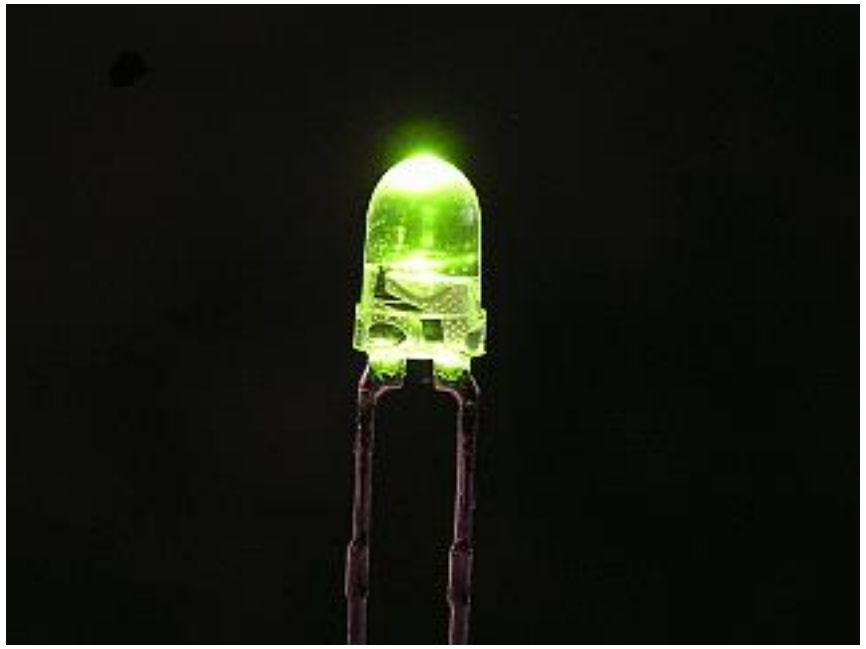
```
def blink():
  while True:
    GPIO.output(LedPin, GPIO.HIGH)  # led on
    time.sleep(0.2)
    GPIO.output(LedPin, GPIO.LOW) # led off
    time.sleep(0.2)

def destroy():
  GPIO.output(LedPin, GPIO.LOW)    # led off
  GPIO.cleanup()                   # Release resource
```

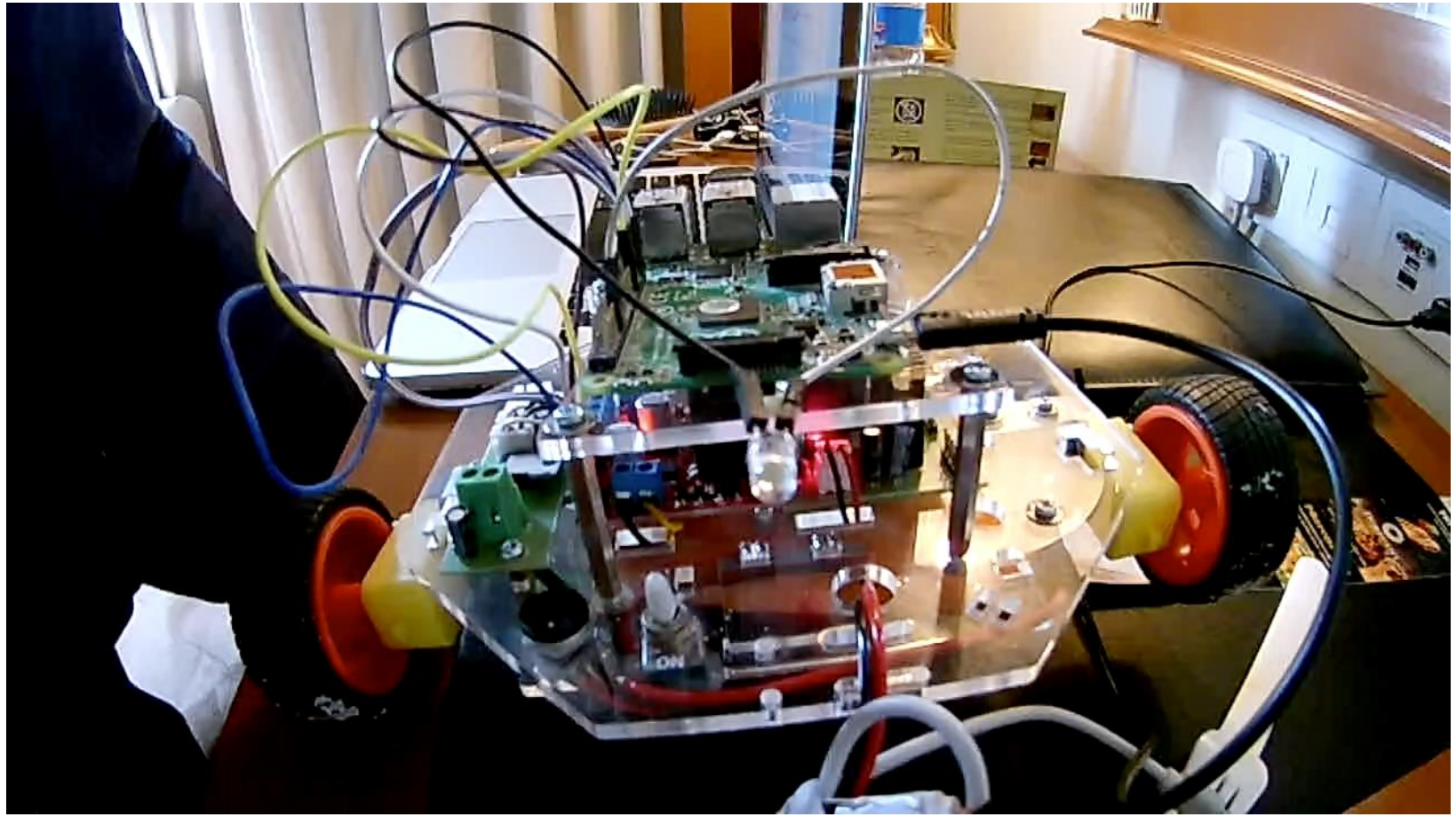
Main Function

```
if __name__ == '__main__':     # Program start from here
  setup()
  try:
    blink()
  except KeyboardInterrupt:  # When 'Ctrl+C' is pressed,
    destroy()
```

Output!

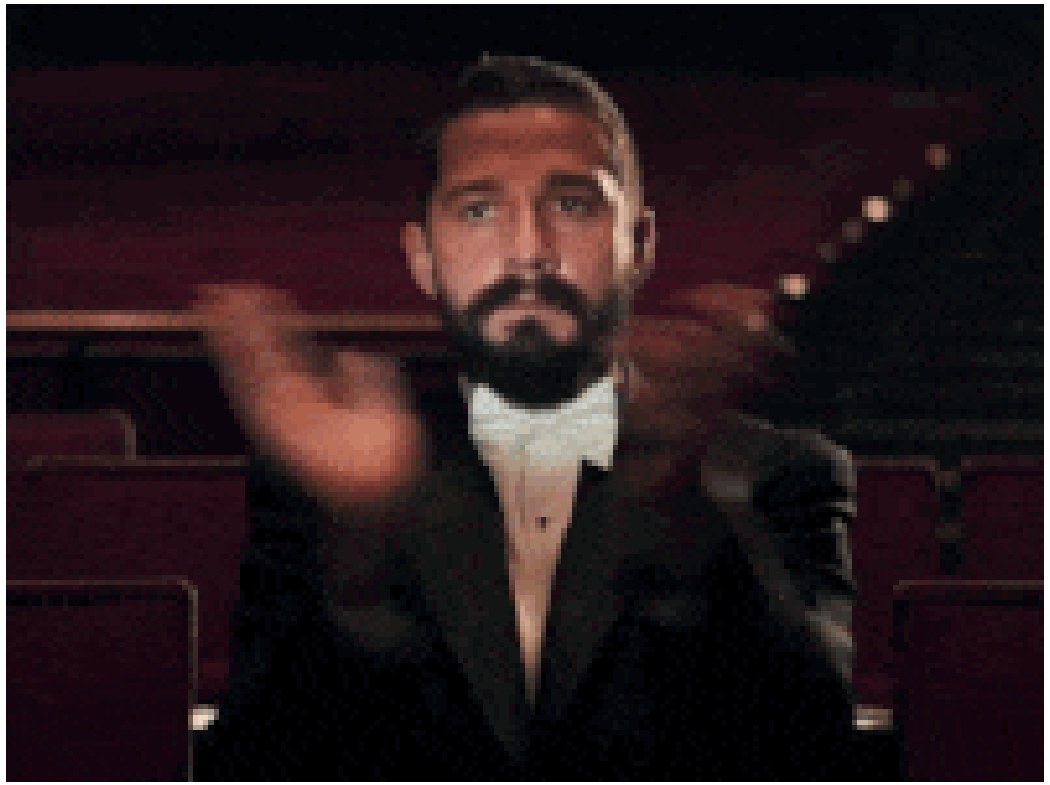


LED Blinking: Hellooo World!



Congratulations!

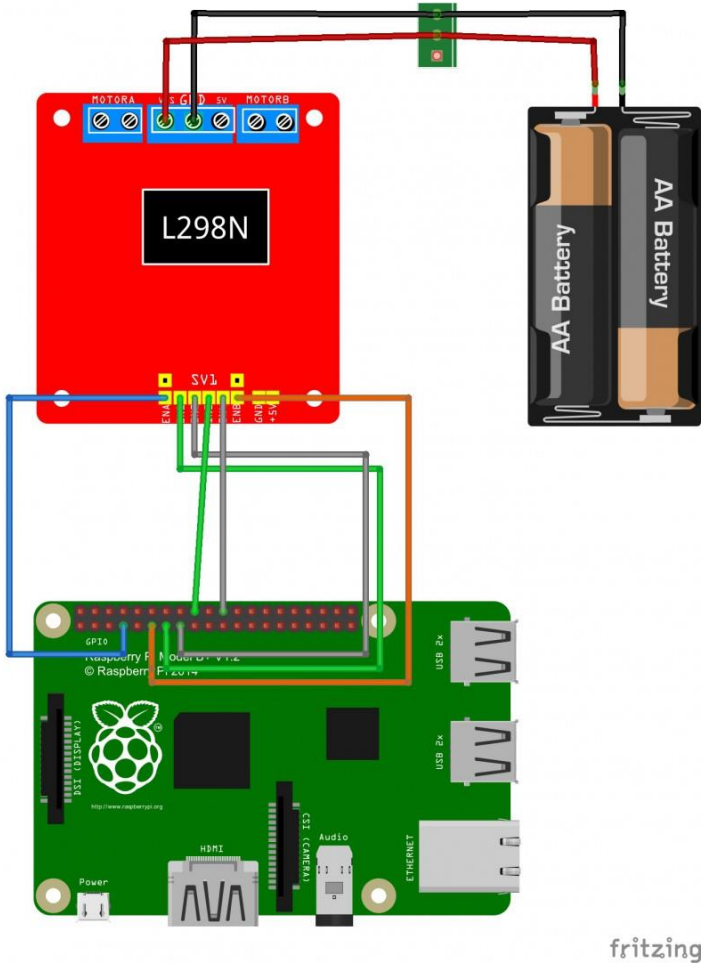
You are a Python Hardware Expert Now!



LearnOBots
Think Explore Make

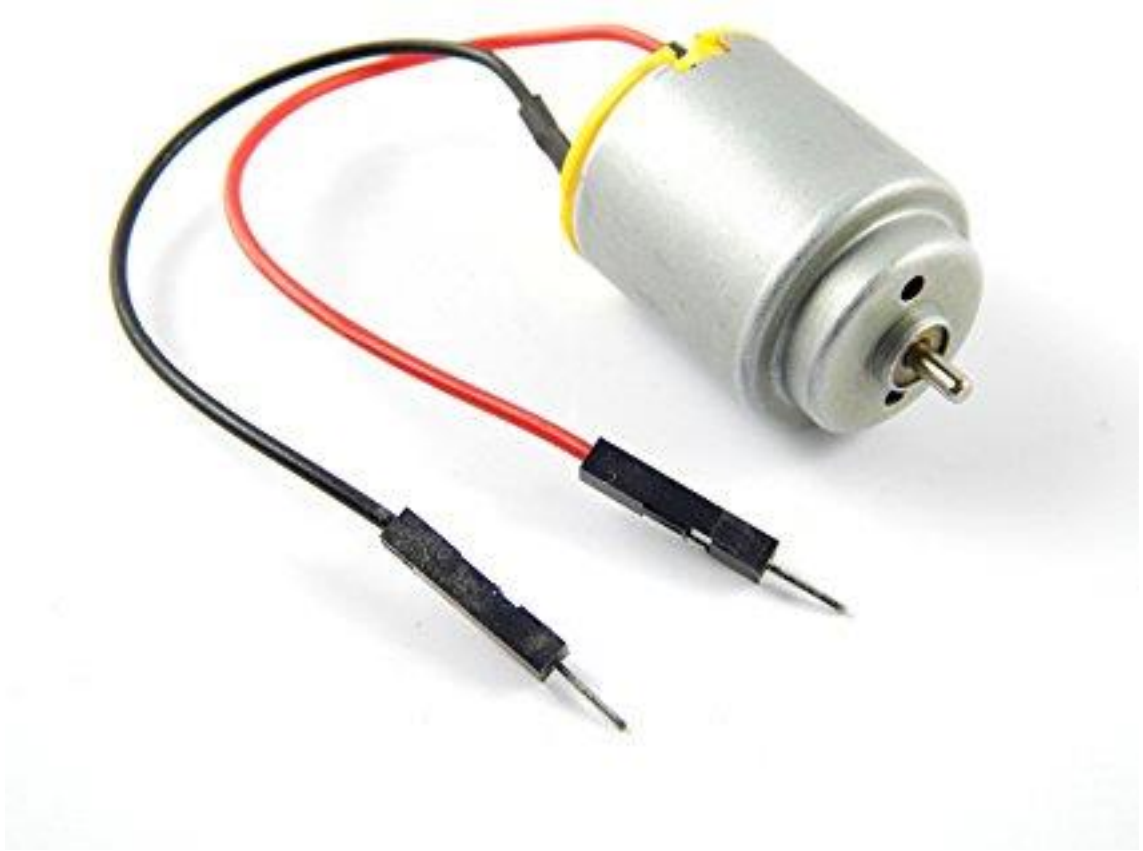
Task 2: Interfacing Motors

- Using CodiBot we will be using an L298 Hbridge to control the 2 motors for robot movement
- Moving Codi Forwards
- Moving Codi Backwards
- Turning Codi using Differential Drive



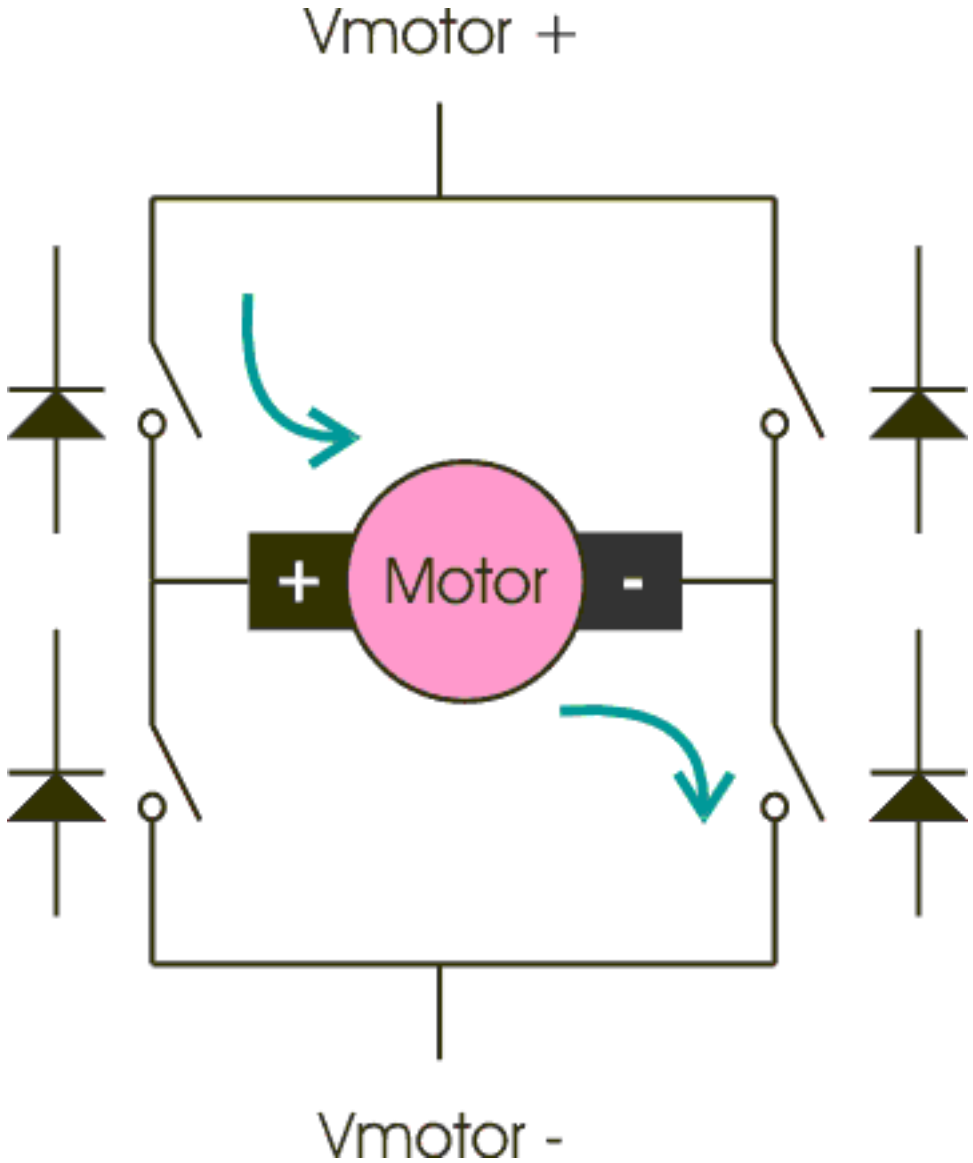
PyCon Pakistan 2017

Have you ever tried this?

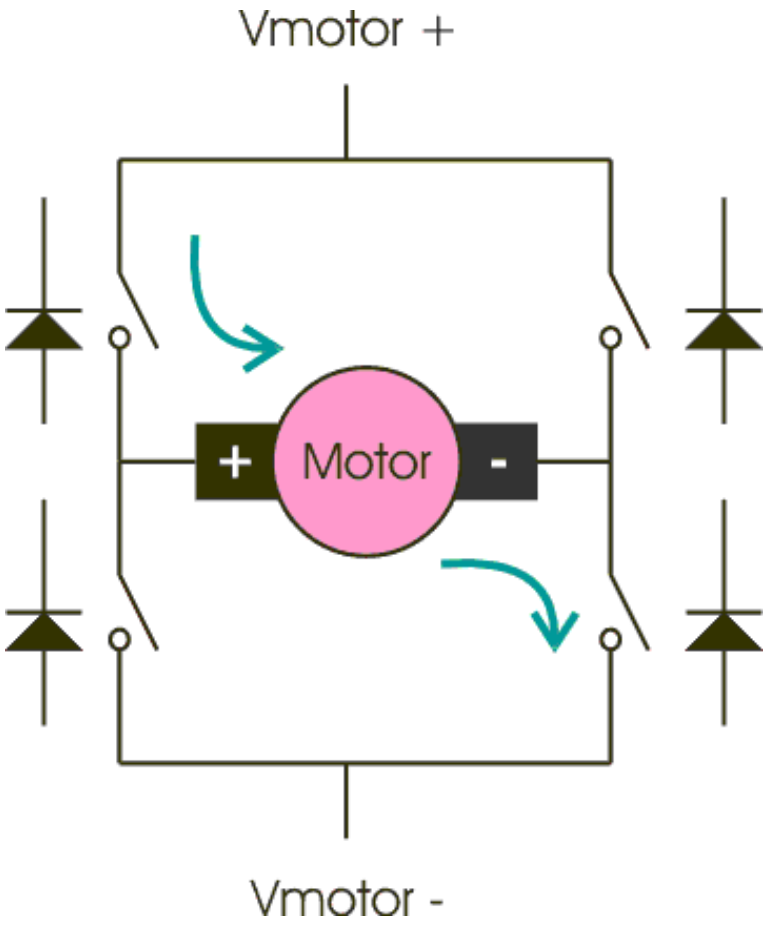


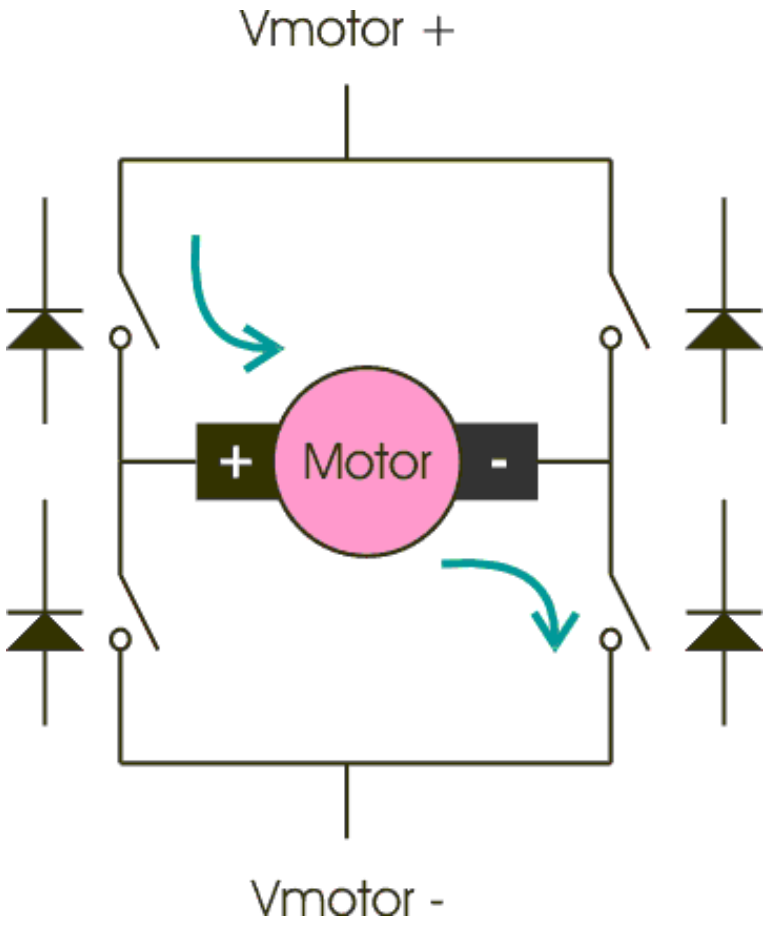
H-Bridge

- Changing Directions



Differential Drive





```
1  # Import required libraries
2  import sys
3  import time
4  import RPi.GPIO as GPIO
5
6  GPIO.cleanup()
7
8  # Define GPIO signals to use
9  # Physical pins 29,31,33,35,37,40
10 RightMotor = 29
11 RM1a = 31
12 RM1b = 33
13
14 LeftMotor = 40
15 LM1a = 35
16 LM1b = 37
17
```

```
20  def setup():
21      GPIO.setmode(GPIO.BOARD)
22      GPIO.setup(RightMotor, GPIO.OUT)
23      GPIO.setup(LeftMotor, GPIO.OUT)
24
25      GPIO.setup(RM1a, GPIO.OUT)
26      GPIO.setup(RM1b, GPIO.OUT)
27      GPIO.setup(LM1a, GPIO.OUT)
28      GPIO.setup(LM1b, GPIO.OUT)
29
30
```

```
31  def forward(x):
32      #H-Bridge Pin Settings
33      GPIO.output(RM1a, GPIO.HIGH)
34      GPIO.output(RM1b, GPIO.LOW)
35      GPIO.output(LM1a, GPIO.HIGH)
36      GPIO.output(LM1b, GPIO.LOW)
37
38      #Turning Motors ON
39      GPIO.output(RightMotor, GPIO.HIGH)
40      GPIO.output(LeftMotor, GPIO.HIGH)
41      print ("Moving Forward")
42      time.sleep(x)
43      GPIO.output(RightMotor, GPIO.LOW)
44      GPIO.output(LeftMotor, GPIO.LOW)
45
```

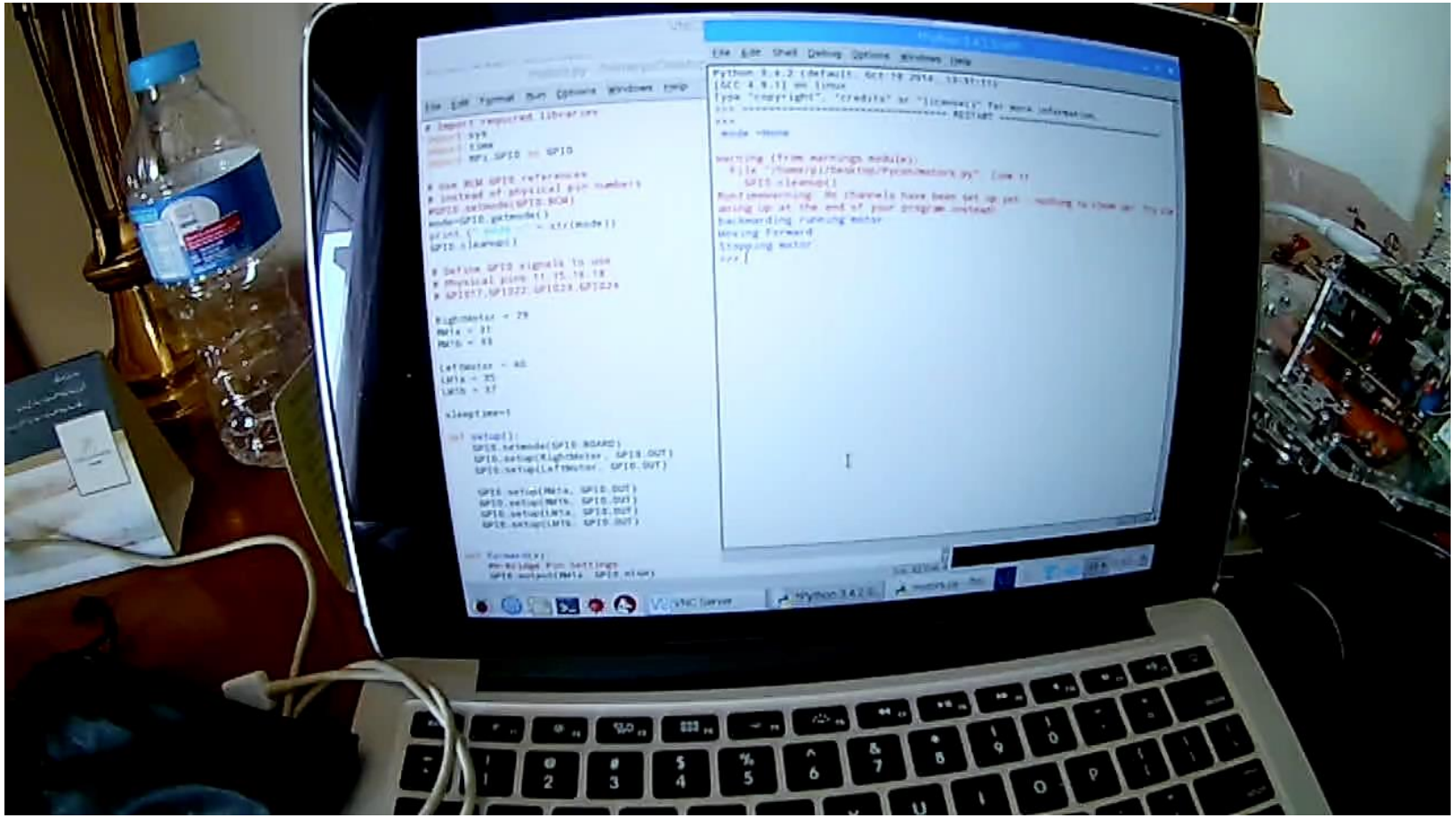
```
46  def reverse(x):
47      #H-Bridge Pin Settings
48      GPIO.output(RM1a, GPIO.LOW)
49      GPIO.output(RM1b, GPIO.HIGH)
50      GPIO.output(LM1a, GPIO.LOW)
51      GPIO.output(LM1b, GPIO.HIGH)
52
53      #Turning Motors On
54      GPIO.output(RightMotor, GPIO.HIGH)
55      GPIO.output(LeftMotor, GPIO.HIGH)
56
57      print ("backwarding running motor")
58      time.sleep(x)
59
60      GPIO.output(RightMotor, GPIO.LOW)
61      GPIO.output(LeftMotor, GPIO.LOW)
62
```

```
64  def dance(x):
65      #H-Bridge Pin Settings
66      GPIO.output(RM1a, GPIO.LOW)
67      GPIO.output(RM1b, GPIO.HIGH)
68      GPIO.output(LM1a, GPIO.HIGH)
69      GPIO.output(LM1b, GPIO.LOW)
70
71      #Turning Motors On
72      GPIO.output(RightMotor, GPIO.HIGH)
73      GPIO.output(LeftMotor, GPIO.HIGH)
74
75      print ("Dancing!")
76      time.sleep(x)
77
78      GPIO.output(RightMotor, GPIO.LOW)
79      GPIO.output(LeftMotor, GPIO.LOW)
80
81  def destroy():
82      print ("Stopping motor")
83      GPIO.cleanup()
84
```

Motor Connections H-Bridge

HIGH	LOW	HIGH	LOW	HIGH	LOW
LOW	HIGH	LOW	HIGH	HIGH	LOW
HIGH	LOW	LOW	HIGH	HIGH	LOW
LOW	HIGH	HIGH	LOW	HIGH	LOW
CODI will move FORWARDS	CODI will move BACKWARDS	CODI will move LEFT	CODI will move RIGHT	CODI won't move	CODI won't move

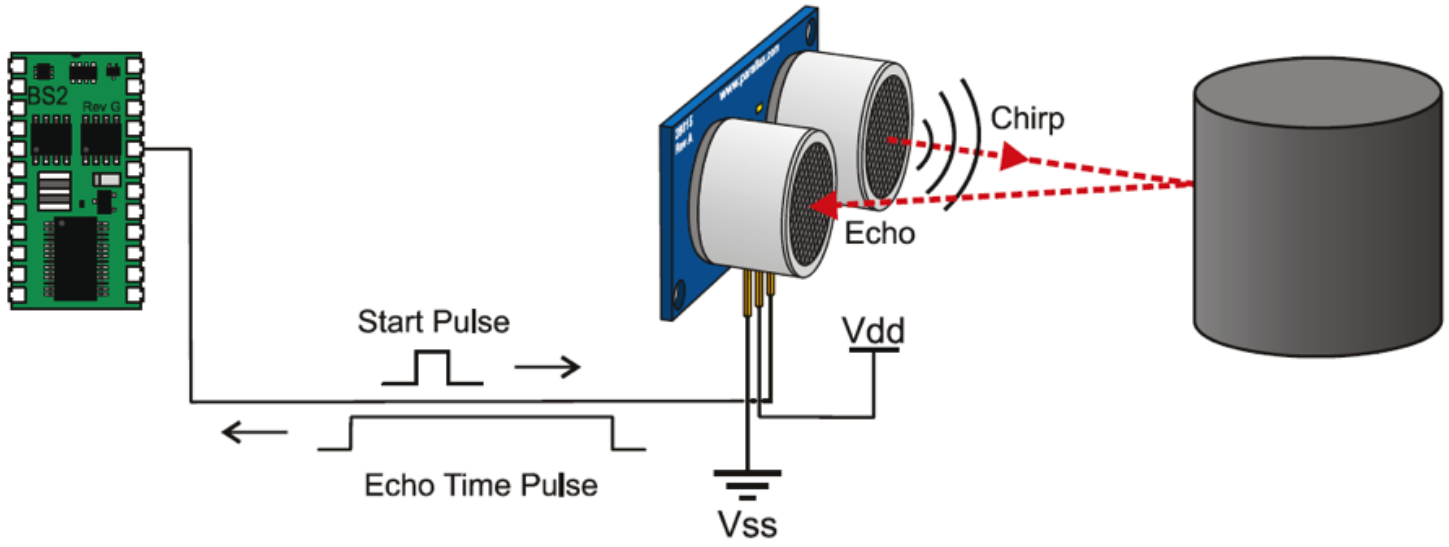
Motors Movement

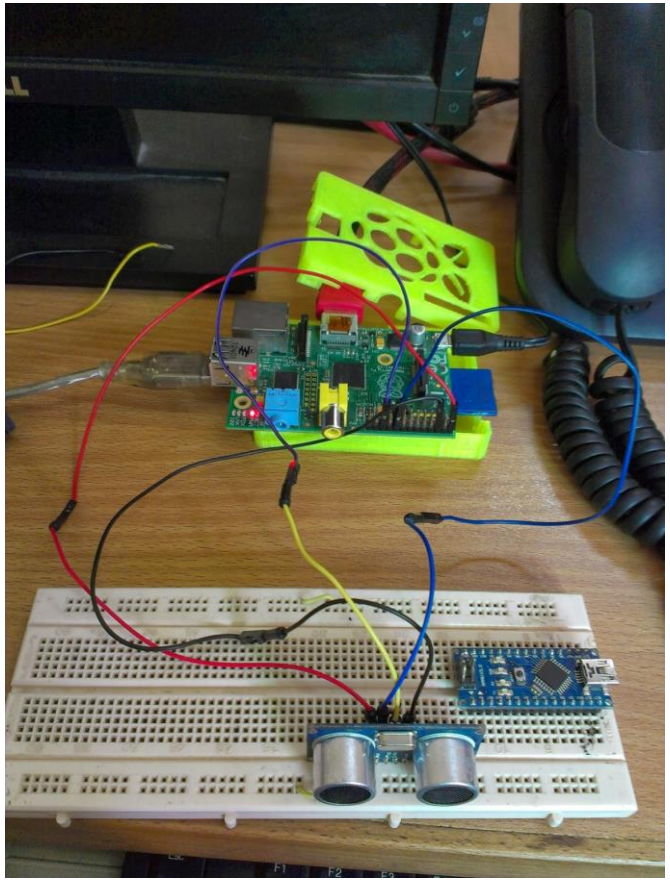


LearnOBots
Think Explore Make

Task 3: Sensing the world

- Interfacing a Sonar Sensor with Raspberry PI
- Using sonar to detect obstacles





PyCon Pakistan 2017

```
1  #Sonar interface with the Raspberry PI
2
3  #Import Python libraries
4  import time
5  import RPi.GPIO as GPIO
6
7
8  GPIO.setmode(GPIO.BOARD)
9
10 GPIO_TRIGGER = 29    #GPIO_24
11 GPIO_ECHO = 40        #GPIO_25
12
13 #Set Pins as output and input
14 GPIO.setup(GPIO_TRIGGER,GPIO.OUT) # Trigger
15 GPIO.setup(GPIO_ECHO,GPIO.IN) # Echo
16
17 #Set Trigger low
18 GPIO.output(GPIO_TRIGGER, False)
19
20 #Allow module to settle
21 time.sleep(0.5)
22
```

```
23  def sonar():
24      #Send 10us pulse to trigger
25      GPIO.output(GPIO_TRIGGER, True)
26      time.sleep(0.00001)
27      GPIO.output(GPIO_TRIGGER, False)
28
29      while GPIO.input(GPIO_ECHO)==0:
30          start = time.time()
31
32      while GPIO.input(GPIO_ECHO)==1:
33          stop = time.time()
34
35      #Calculate pulse length
36      elapsed = stop-start
37
38      #Distance pulse traveled in that time is time
39      #multiplied by the speed of sound (cm/s)
40      distance = elapsed * 34000
41
42      #That was the total distance so half it for reaching the object
43      distance = distance /2
44
45      return distance
```

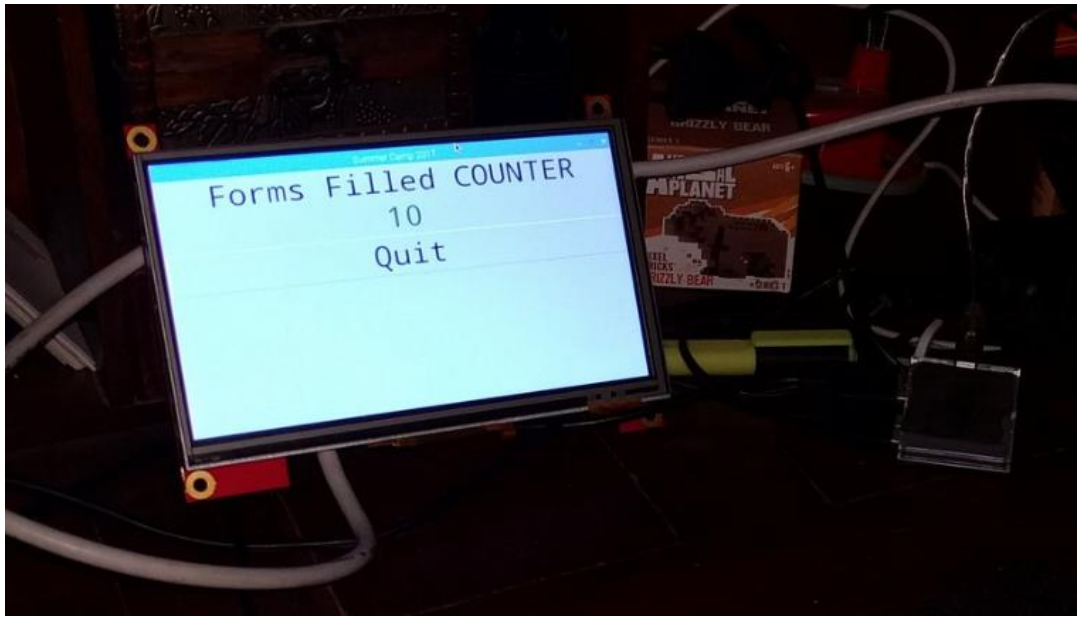
```
47  while True:
48      time.sleep(0.3)
49
50      distance = sonar()
51      print (distance)
```

LearnOBots
Think Explore Make

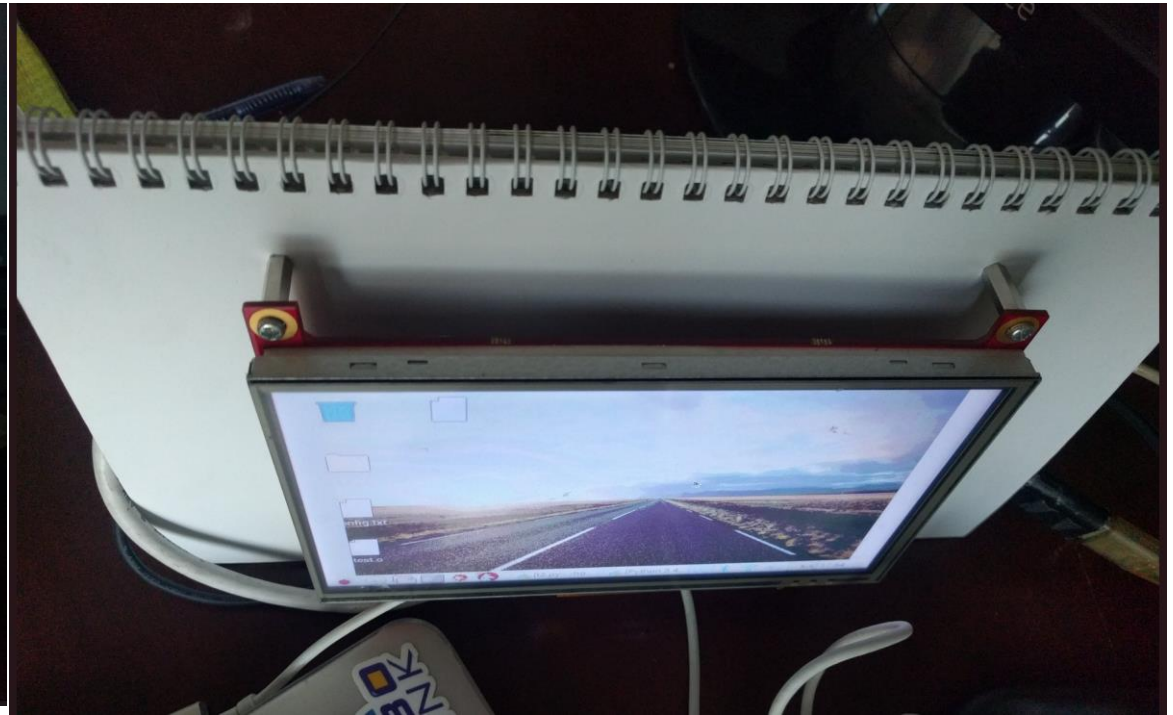
Task 4: Obstacle Avoidance Robot

- Using our learning from the previous 3 tasks we can now program an obstacle avoidance robot.
- Robot moves randomly and turns as soon as it encounters an obstacle
- But let's leave that task for you ☺

PyCon Pakistan 2017


Forms Filled COUNTER
10
Quit

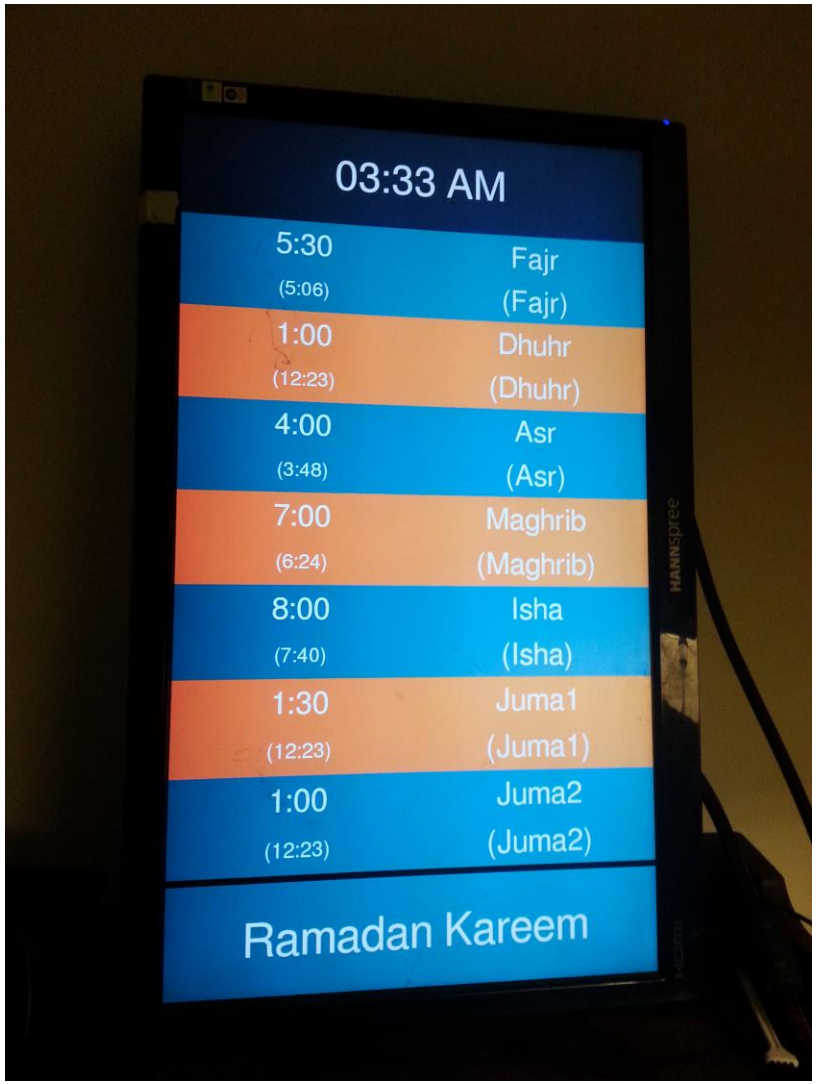

test.o

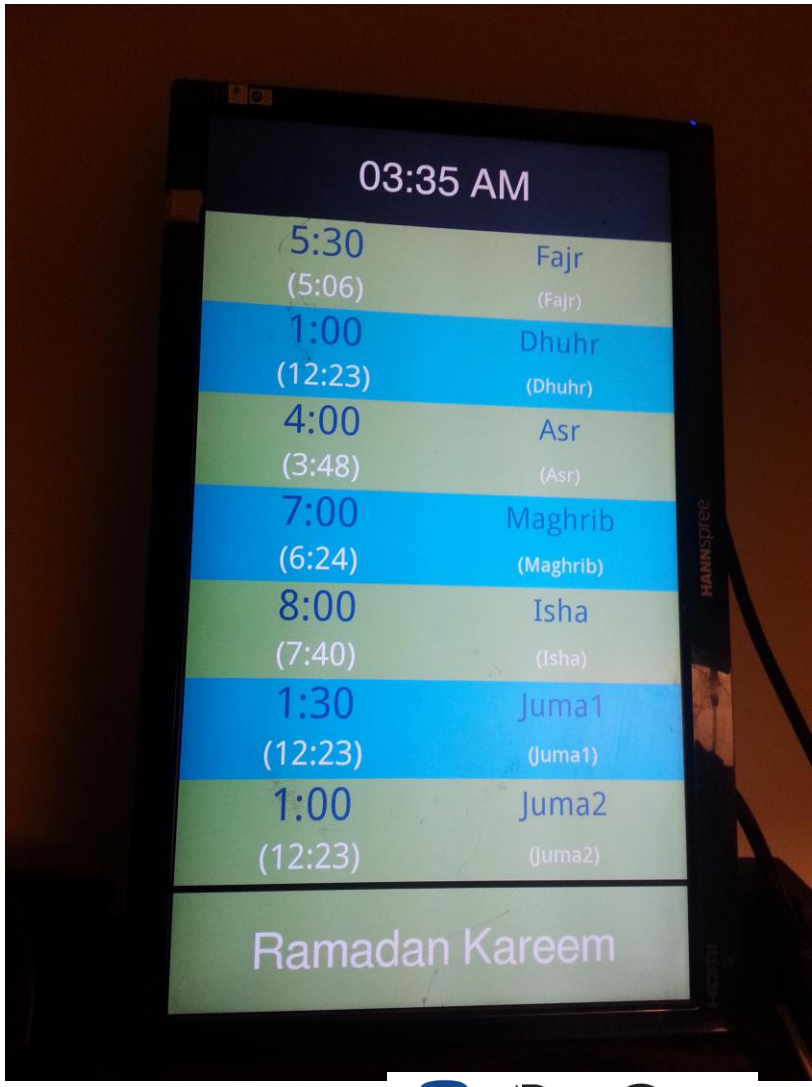


LearnOBots
Think Explore Make

Conclusion

- Python is Easy
- Lots of library support
- Raspberry PI is a natural tool to develop applications that use Python / Hardware and a GUI





PyCon Pakistan 2017

LearnOBots
Think Explore Make

Thank You

Find Today's Code on

https://github.com/shamyl/Pycon

- t/@shamylmansoor
- shamyl@learnobots.com
- fb.com/learnobots

PyCon Pakistan 2017