The University of the West Indies, St. Augustine
INFO 2603 Platform Technologies 1
2018/2019 Semester 1
Lab 8 : 21-22, November 2018

Networking & Virtualisation using Raspberry Pi Robots

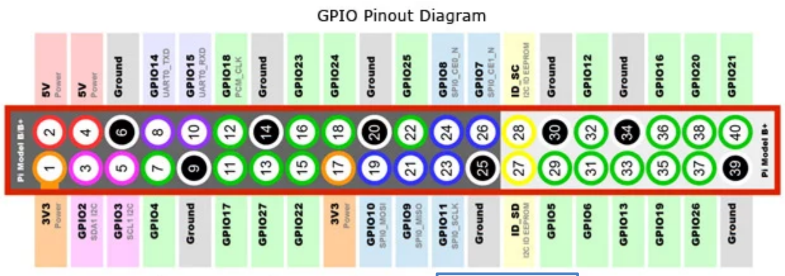
Learning Objectives

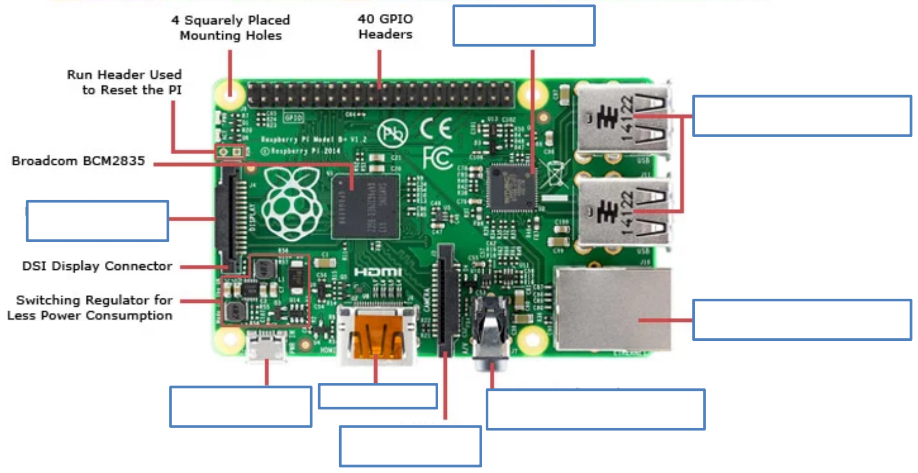
- Understand the basic components of the Raspberry Pi microcomputer
- Learn about the GoPiGo Robot Car's components
- Be able to connect to the GoPiGo Robot via a computer
- Get familiar with the Virtual Network Computer (VNC) interface for communicating with the GoPiGo
- Learn how to control the GoPiGo Robot using the Python programming interface

Activity 1

Identify the basic components of a Raspberry Pi

- Ports - HDMI, USB, Power/Mini USB, Ethernet
- CPU
- WiFi Card
- Micro SD Card
- GPIO - General Purpose I/O - allows for communication with a vast array of both input and output devices. Also allows for interfacing with breadboards. [A breadboard is an electrical component that provides a universal interface for a variety of I/O devices, e.g., cameras, distance sensors, motors, actuators, LEDs]



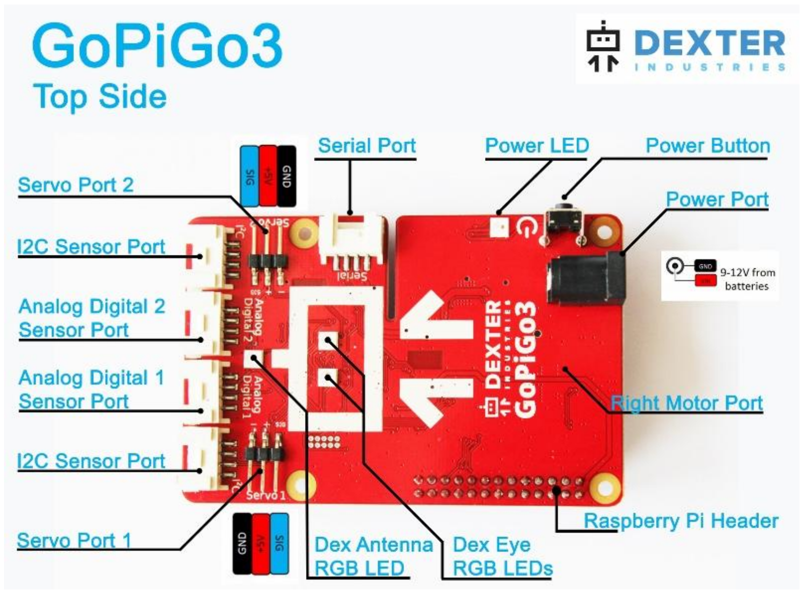

4 Squarely Placed Mounting Holes
40 GPIO Headers
Run Header Used to Reset the PI
Broadcom BCM2835
DSI Display Connector
Switching Regulator for Less Power Consumption

- What is the Broadcom BCM2835?
- What is it used for on the Raspberry Pi?

Activity 2

Identify the basic components of the robot car

- GoPiGo Robot Car's components
- GoPiGo3 Board
- Battery Pack
- Wheels
- Chasis
- Motors
- Encoders



Activity 3

Setup the Connection to the Robot

1. Observe the the port where the battery pack wire interfaces with the red board. This red board is the GoPiGo board that interacts with the green board, the Raspberry Pi, so that we can control the GoPiGo robot.

2. To turn on your robot, press the black button near the previously mentioned port.

3. After pressing the button, you should notice that an LED has begun flashing. Once this LED becomes a solid forest-green colour, give your robot about 30 seconds to start up. Just like any other computer, the GoPiGo needs to be properly initialised before use.

4. Once 30 seconds has passed, you can **ping** your robot. To ping your robot, open the **command prompt** and type **`ping <robot name>`**. Pinging is used to tell whether your computer can communicate with another computer over a network.

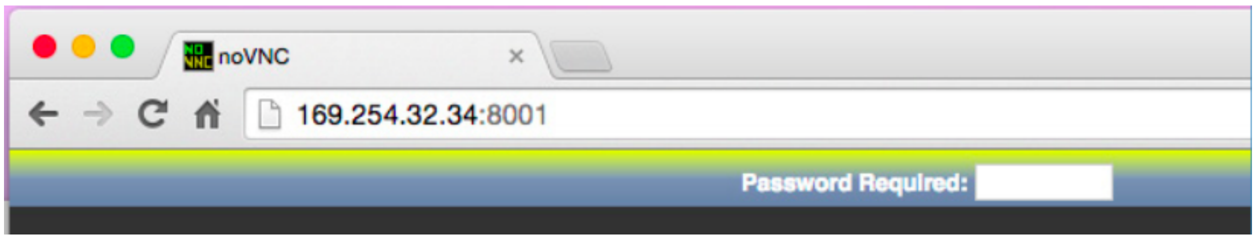
5. Try pinging the robot of your neighbouring group. What happens?

6. If you can ping the robot, then you can communicate with it over the network. Open up Google Chrome and type your robot's name in the address bar, followed by a forward slash. eg. **`bob/`**

7. From here you will see a screen with two options. Click the left option - to launch VNC.

- What does VNC stand for?
- What is its purpose?
- Explain how it works in terms of clients and servers.

8. Type in the password **`robots1234`** to access the robot's computer.



9. Once inside, you will see the GUI of a fully-fledged operating system.
   - What is the name of this operating system?
   - Which operating system is it based off of?
   - Is it open-source?
   - Explain whether virtualisation is taking place here.

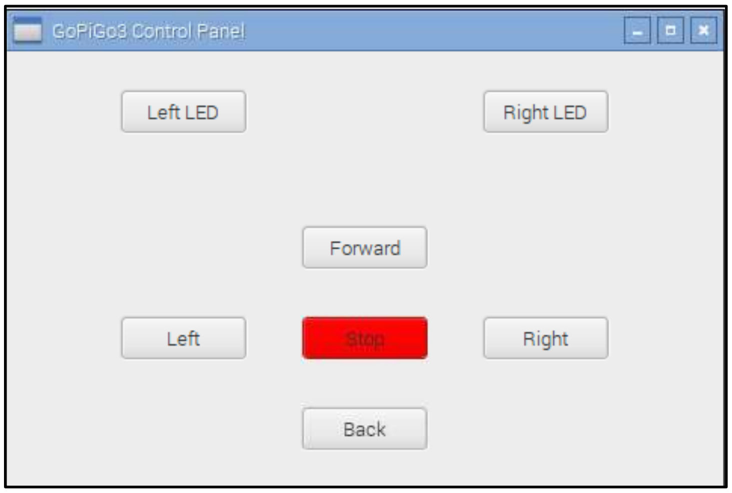
10. Explore the robot computer's desktop using the client machine:
    - What is the IP address of the robot's computer?
    - What is the IP address of the desktop machine that you are using to remote into the robot's computer?
    - Are they the same? Explain why or why not.

11. What happens if you try to connect to your neighbour's robot using steps 6,7 and 8? Did it work? Why or why not? Explain carefully using networking concepts.

Activity 4

Control the Robot using the Control Panel application

1. On the desktop, you will see a control panel application. Open this control panel application.



2. Click the buttons for the LEDs on the robot to turn them on/off. Try moving the robot forward and back, right and left.

Activity 5

Getting Started with IDLE

IDLE is Python's Integrated Development and Learning Environment.

IDLE has 2 windows:

- The Shell Window is an interactive interpreter. It allows you to type code and run it within the same window, showing the output of your code.
- The Text Editor allows code to be saved in files and edited. These files are called python files and end in `.py`

Simple Code - Hello World!

1. Launch the IDLE Shell Window
2. Type the code:

   ```
   print("Hello World!")
   ```
3. You should see the output: Hello World!
4. Try printing your name. Type the code:

   ```
   print("My name is Bob")
   ```
5. What do you see?
6. Now, type the following code: `print hey, this is fun`
7. What do you notice? Let's fix that error. Type the code: `print("hey, this is fun”)`

Activity 6

Control the Robot using Python Commands

You need to include the following import statement at the top of your program:

```
from UWIGoPiGo import *
```

The following functions can be used to control the movement of the GoPiGo robot.

Table 1. API for GoPiGo3

Function	Return Value	Description	Example Code
init_sensor()	None	Initialises the distance sensor	init_sensor()

Function	Return Value	Description	Example Code
measure()	distance	Measures the distance from the robot to the nearest object in front of the distance sensor	distance = measure()
init_servo()	None	Initialises the servo motor	init_servo()
rotate_servo(position)	None	Rotates servo to a specific position	rotate_servo(90)
reset_servo()	None	Resets servo to default position	reset_servo()
forward()	None	Moves the robot forward	forward()

backward()
reverse()
	None	Moves the robot backward	backward()

stop()	None	Stops the robot from moving	stop()
go_forward(distance)	None	Moves the robot forward a specified distance in cm	go_forward(5)
go_backward(distance)	None	Moves the robot backward a specified distance in cm	go_backward(5)
rotate_right(angle)	None	Turns the robot RIGHT (clockwise) by a specified angle	rotate_right(180)
rotate_left(angle)	None	Turns the robot LEFT (anti-clockwise) by a specified angle	rotate_left(180)

Function	Return Value	Description	Example Code
get_speed()	speed	Gets the speed at which the robot moves	speed = get_speed()
set_speed(speed)	None	Changes the current speed of the robot	set_speed(75)

Move Robot Programmatically - Forward and then backward

```
from UWIGoPiGo import *
import time

forward( )
time.sleep(2)
stop( )

backward( )
time.sleep(2)
stop( )
```

Control LEDs (Light Emitter Diodes) Programmatically

```
from UWIGoPiGo import *

GPG = get_robot()

#For non-RGB leds the API call
#GPG.set_led(LED_ID,intensity)
#Non-RBG LED_ID: GPG.LED_BLINKER_LEFT, GPG.LED_BLINKER_RIGHT

GPG.set_led(GPG.LED_BLINKER_LEFT,255) #0 off, 255 0n
GPG.set_led(GPG.LED_BLINKER_LEFT,0) #0 off, 255 0n

#For RGB leds (i.e., GPG.the API call
#GPG.set_led(LED_ID,red intensity,green intensity, blue intensity)
#Non-RBG LED_ID: GPG.LED_EYE_LEFT, GPG.LED_EYE_RIGHT, GPG.LED_WIFI

GPG.set_led(GPG.LED_EYE_LEFT,0,0,255) #0 off, 255 0n
GPG.set_led(GPG.LED_EYE_LEFT,0,0,0)
```

References and Resources
https://www.raspbian.org/
https://www.raspberrypi.org/documentation/remote-access/vnc/
https://searchnetworking.techtarget.com/definition/virtual-network-computing