

Raspberry Pi Nodebots

by the Mozilla Campus Club Birmingham (mozccbham.org)

This set of lessons will show you some of the basics on Node.js and Raspberry Pi for robotics.

Setup: Checking everything works

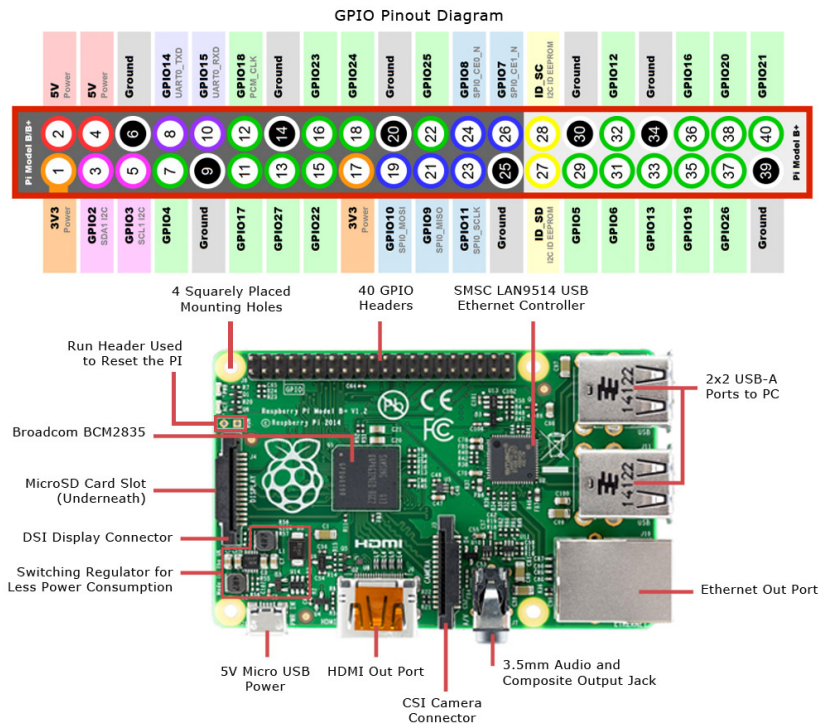
Collect a screen, keyboard, mouse, VGA adapter, Pi, USB power cable and set the pi up as shown below. Turn the pi on and check that it boots properly and that everything works.



Example 1: Switching on an LED with the GPIO pins

First, we need to connect an LED to the Pi. Make sure the Pi is off by removing the power cable and then connect the LED to the Pi as shown on the board. You will need the resistor in your pack.

You might find the pin diagram below useful.



Now connect the micro-usb power cable to the Pi again and the Pi should power on.

The starting code for the example is on the desktop in the nodebots folder. Open this folder and open example 1. In this folder, you will find a file called index.js. Double-click the file and it will open so you can edit it. The text will be the same as in the image below.

```
var wpi = require("wiringpi-node")
var sleep = require('sleep')

// Set up pins for output
wpi.wiringPiSetupGpio()

var ledPin = 3 // GPIO pin 3, Physical Pin 5
wpi.pinMode(ledPin, wpi.OUTPUT)

// Turn LED off
wpi.digitalWrite(ledPin, wpi.LOW)

// See if you can change the blinking pattern
while(true) {
  wpi.digitalWrite(ledPin, wpi.LOW)
  sleep.sleep(1);
  wpi.digitalWrite(ledPin, wpi.HIGH)
  sleep.sleep(1);
}
```

To run the script, we use the node runtime environment. To run the program, open a terminal and type

```
cd Desktop/nodebots/example1
```

to change directory into the correct folder. You can now type
`ls`

and you will see the files that are in the current folder. To start the node program type
`sudo node index.js`

and you can then stop it later by pressing `ctrl+c`. If everything has been set up correctly, the LED you have connected will be blinking. If it isn't, try swapping the LED direction.

Challenge

See if you can modify the code to change the blinking pattern.

Example 2: Running a webserver on the Raspberry Pi

In this example, we are going to run a website server on the raspberry pi and then access the website from another machine on the network.

For this example, we need to open the nodebots folder on the desktop, then open `example2` and double-click on the `index.js` file to edit.

```
var express = require("express")
var app = express()

app.get('/', function(req, res) {
  res.send("Hello World")
})

app.get('/hello', function(req, res) {
  res.send("Hello there, how are you?")
})

// add a goodbye route here

app.listen(3000)
console.log("Listening on port 3000!")
```

Change into the directory on the terminal by typing
`cd ../example2`
and run the web server in this folder.

This script will run a webserver. This example contains two routes, / and /hello.

If you open the web browser you can browse to the website running on the Pi.

In the address bar type
`localhost:3000/`
You should see "Hello World". Now try the second route
`localhost:3000/hello`
You should now see "Hello there, how are you?".

To access the webserver from another computer (or smartphone) on the same network you will need the address of your machine. In the top right corner of the screen, hover over the network icon and it will tell you the address. It will be something like 10.10.10.2. Make sure your webserver is running and then go to another machine, open the browser and type
`10.10.10.2:3000/`
You should see "Hello World".

Challenge

See if you can add a third /goodbye route.

Example 3: Controlling a Pi robot from a web page

In this example, we are going to run a web server on the raspberry pi that can control the robot motors.

For this example, we need to open the nodebots folder on the desktop, then open example3 and double-click on the index.js file to edit.

This script will run a webserver and has routes to make the robot move forwards, left and stop.

At this point you should raise your robot's wheels off the table.

Now with the program running, open the browser and go to

localhost:3000/forwards

To stop the robot, go to

localhost:3000/stop

You should be able to see these links as bookmarks in the browser.

Challenge

See if you can complete the /backwards route and add the /right route.

```
var wpi = require("wiringpi-node")
var express = require("express")
var app = express()

// Set up pins for output
wpi.wiringPiSetupGpio()

var leftPin1 = 7
var leftPin2 = 8
var rightPin1 = 9
var rightPin2 = 10

wpi.pinMode(leftPin1, wpi.OUTPUT)
wpi.pinMode(leftPin2, wpi.OUTPUT)
wpi.pinMode(rightPin1, wpi.OUTPUT)
wpi.pinMode(rightPin2, wpi.OUTPUT)

// Turn all motors off
wpi.digitalWrite(leftPin1, wpi.LOW)
wpi.digitalWrite(leftPin2, wpi.LOW)
wpi.digitalWrite(rightPin1, wpi.LOW)
wpi.digitalWrite(rightPin2, wpi.LOW)

app.get('/forwards', function(req, res) {
  wpi.digitalWrite(leftPin1, wpi.LOW)
  wpi.digitalWrite(leftPin2, wpi.HIGH)
  wpi.digitalWrite(rightPin1, wpi.LOW)
  wpi.digitalWrite(rightPin2, wpi.HIGH)
})

app.get('/backwards', function(req, res) {
  // Add code here to move the robot backwards
})

app.get('/left', function(req, res) {
  wpi.digitalWrite(leftPin1, wpi.HIGH)
  wpi.digitalWrite(leftPin2, wpi.LOW)
  wpi.digitalWrite(rightPin1, wpi.LOW)
  wpi.digitalWrite(rightPin2, wpi.HIGH)
})

// add route to move robot right

app.get('/stop', function(req, res) {
  wpi.digitalWrite(leftPin1, wpi.LOW)
  wpi.digitalWrite(leftPin2, wpi.LOW)
  wpi.digitalWrite(rightPin1, wpi.LOW)
  wpi.digitalWrite(rightPin2, wpi.LOW)
})

app.listen(3000)
console.log("Listening on port 3000!")
```