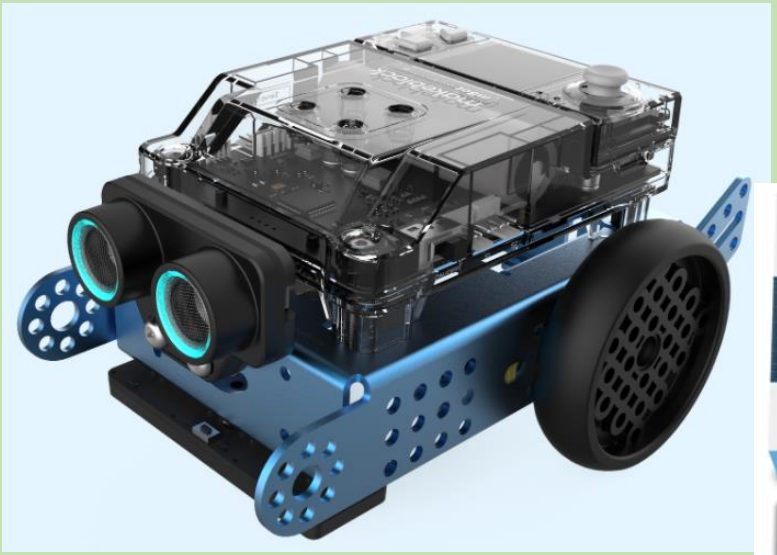
Digital Technology

MakeBlock mBot2 / CyberPi

Block Coding



Version 1.0
Feb 2022

Barry Butler
bbutl58@eq.edu.au

Content and Challenges

Section	Content
A	The mBot2 Vehicle
B	Introduction and Setup
C	Our First Program – Hello
D	Push the Buttons
E	Run the Motors
F	Avoid or Seek
G	Detect and Follow a Line
H	SumoBot
I	Connect Servos, Sensors and Motors
Appendix 1	CyberPi Extras

Documentation

https://www.yuque.com/makeblock-help-center-en/mblock-5

CyberPi Blocks https://www.yuque.com/makeblock-help-center-en/cyberpi
(including Pocket Shield, mBot2 Shield and mBuild Modules)

Firmware Update

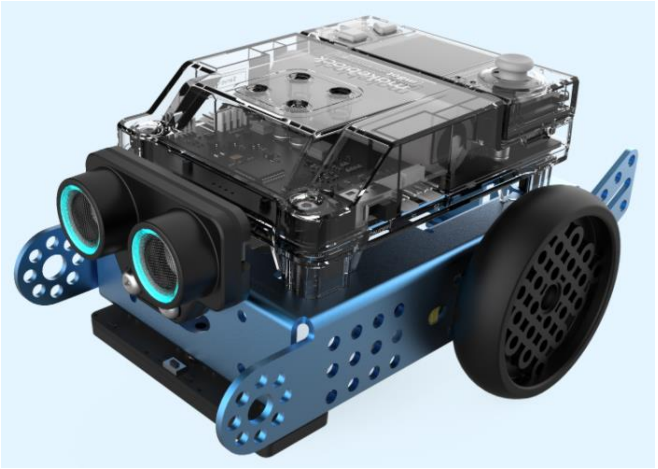
To update the CyberPi firmware:

1. Open the online ide at https://ide.mblock.cc/#/
2. Click on **Devices** and add the CyberPi device to the list, if it is not there already
3. Click **Connect** and connect the CyberPi (download and install the device driver, if asked)
4. Click on **Settings** and select Firmware Update

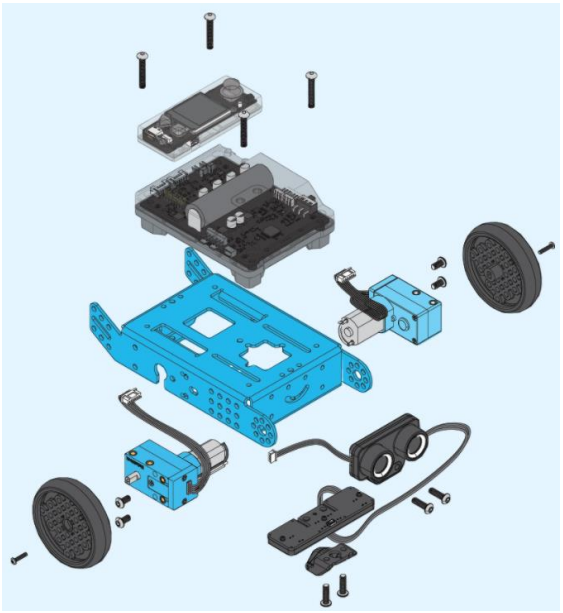
A. The mBot2 Vehicle

Documentation

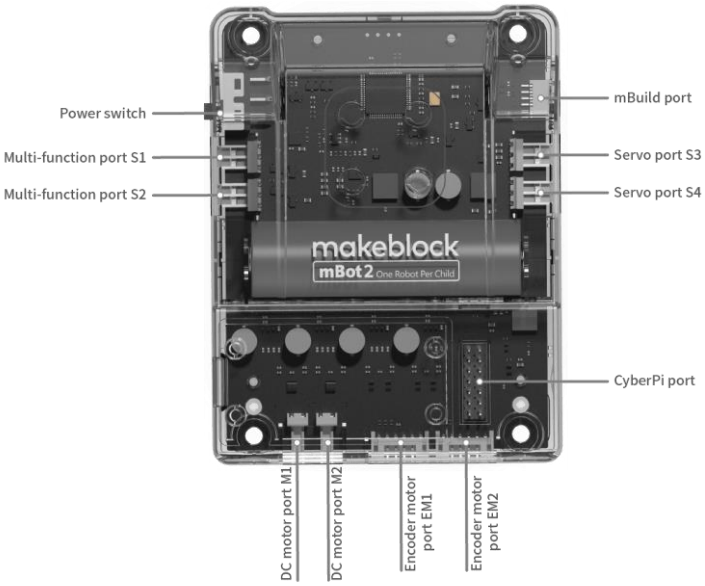
MBot2 Introduction
https://education.makeblock.com/help/cyberpi-series/cyberpi-series-cyberpi-series-packages-and-extensions/mbot2-introduction/
Operational Guide
https://education.makeblock.com/help/cyberpi-series/cyberpi-series-cyberpi-series-packages-and-extensions/mbot2-operational-guide/
Python Reference
https://www.yuque.com/makeblock-help-center-en/mcode/cyberpi-api-shields#9eo89
mBuild Modules (Ultrasonic Sensor 2, Quad RGB Sensor)
https://www.yuque.com/makeblock-help-center-en/mcode/cyberpi-api-mbuild
or
https://education.makeblock.com/help/mblock-python/mblock-python-editor-python-api-documentation-for-devices/mblock-python-editor-python-api-documentation-for-cyberpi/mblock-python-editor-apis-for-mbuild-modules/



The Build



The Connections
(Ultrasonic into the mBuild port, motors to EM1/EM2)



The Power Switch must be turned on before the you can upload code

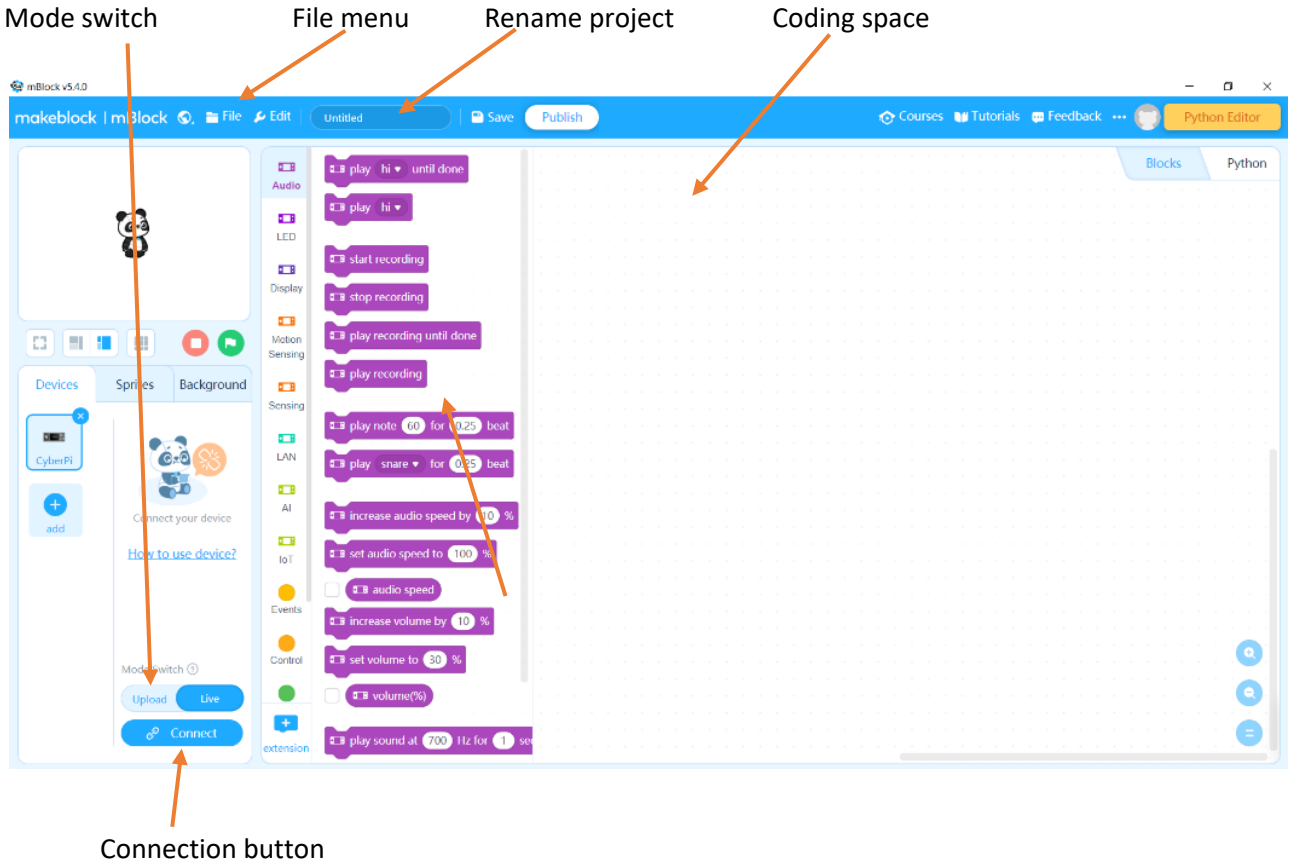
B. Introduction and Setup

Download and Install the Software

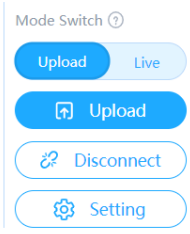
Download and install the mBlock Windows or Mac software from https://mblock.makeblock.com/en-us/download/

(The PC software seems to be more stable than the web version located at https://python.mblock.cc/)

1. Run the MakeBlock software



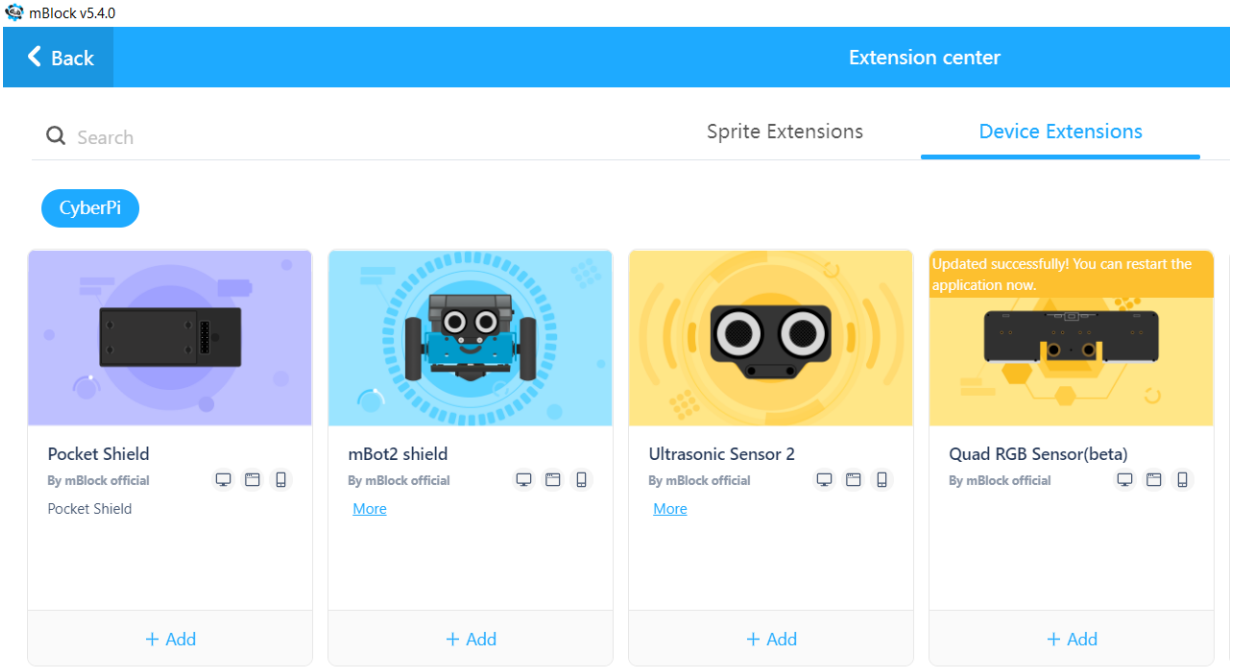
When you **Connect** and change to **Upload** mode the Upload button appears, to upload code to the mBot2.



Click the Setting button to carry out a firmware update and set up a wi-fi connection.

2. **INSTALL AND UPDATE BLOCK MODULES**

If the mBot2, Ultrasonic Sensor 2 or Quad RGB Sensor pictures have a plus or update symbol on them, click on this symbol to add or update these blocks.



Click the Add buttons to add these block sets to your software.

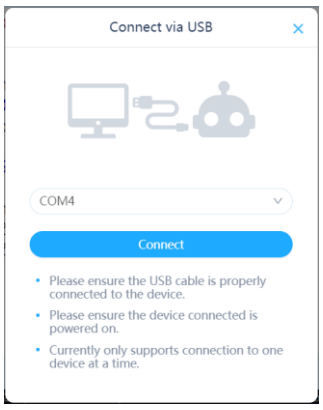
3. **TURN ON THE MBOT2 USING THE SWITCH ON THE SIDE**

The lights on both the ultrasonic sensor and the line follower sensor should turn on. If they are don't, the wiring is incorrect or unplugged, and needs to be fixed.

4. Select **Upload** mode.

5. Plug the mBot2 into a USB port and click the **Connect** button 

.



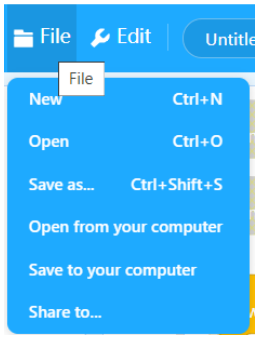
Find Your Port

You can easily find your device by first leaving your mBot **unplugged**. Click *Connect* and look at the list of USB ports.

Close the connect window, then **plug in** your mBot2. Click *Connect* again and look for the port that has just been added.

Select your USB port from the list and click **Connect**.

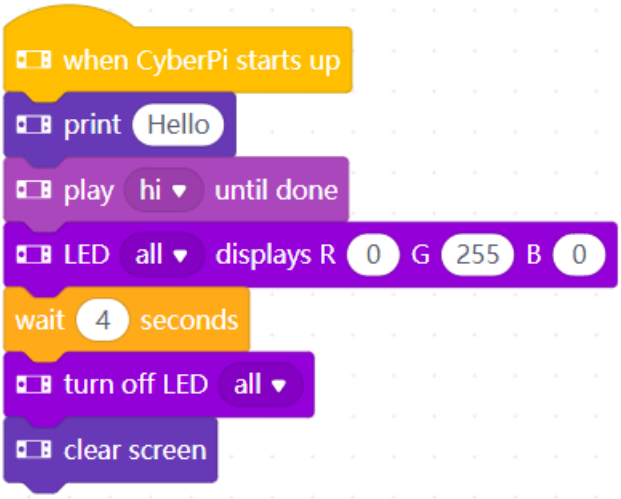
6. Click on the **File** menu and select **New Project**.



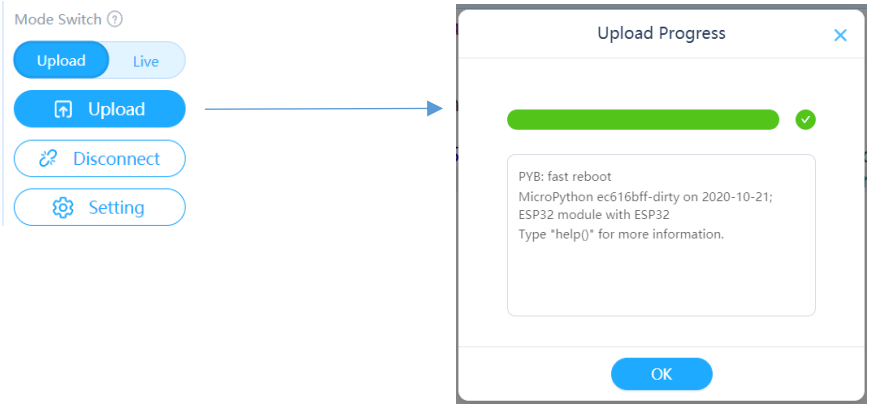
7. Start coding

C. Our First Program – Hello

Our first program will write ‘hello’ on the console, say it on the audio speaker and turn all LED’s to green for 2 seconds.



Click the **Upload** button to send your code to the mBot2.



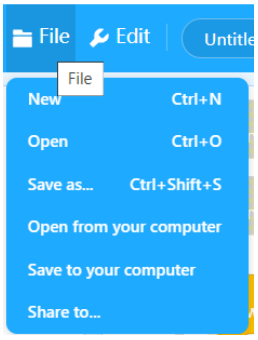
The code will start executing immediately it is uploaded.

Unsuccessful Upload

If the upload is unsuccessful check three things:

1. The **mBot2 is turned on** (the power switch on the left side).
2. The cable is plugged in and a **connection established** (see section B4).

Save the Project and Upload to the CyberPi



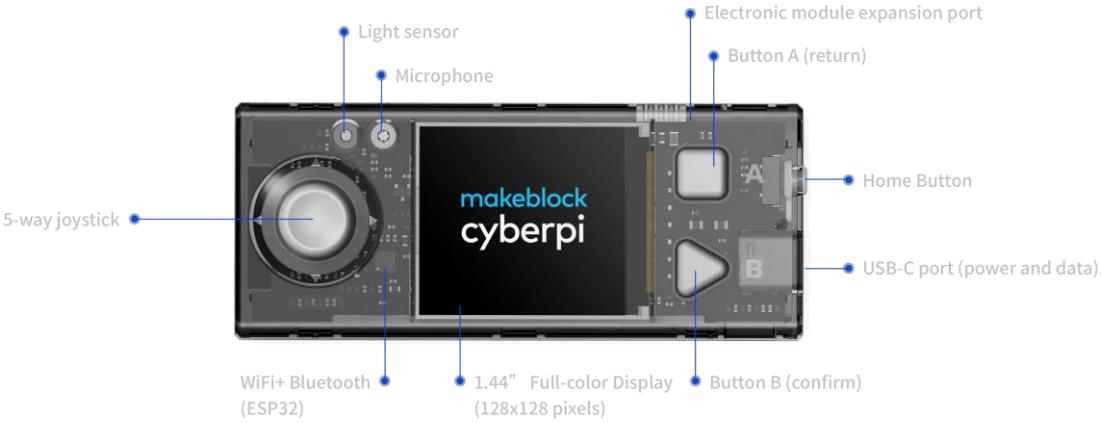
Save the project to your computer by clicking on the **File** menu and choosing **Save to your computer**.

It is a good idea to create a folder to contain all your projects.

Make sure you type in a descriptive name for your file.

D. Buttons

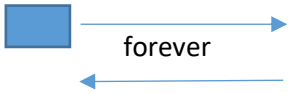
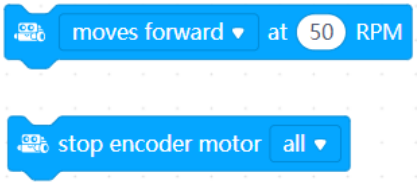
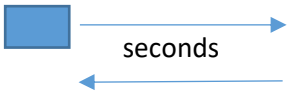
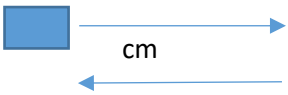
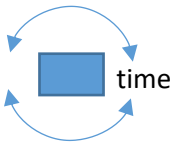
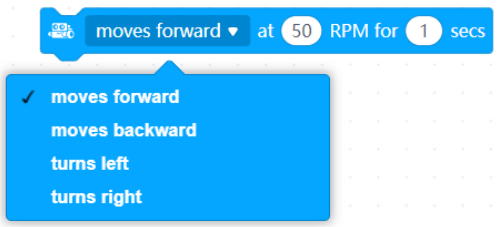
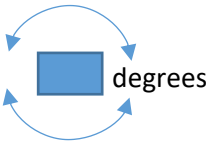
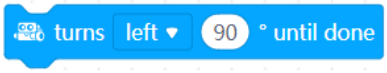
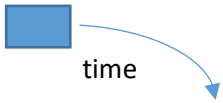
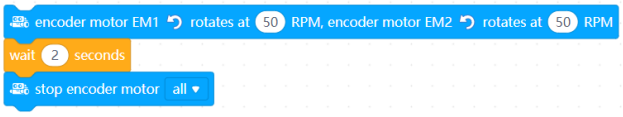
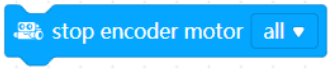
The mBot2 is controlled by a module called **cyberpi**. This has a joystick, a home button and two push buttons (A and B). We can use the joystick and buttons in our code. It also has a light sensor and microphone that we can use.



Instead of the code running automatically when it is uploaded, let's turn on the display when we press button A.

E. Run the Motors

There are a number of ways we may want to move the mBot2. The blocks we need are in the mBot2 Chassis group.

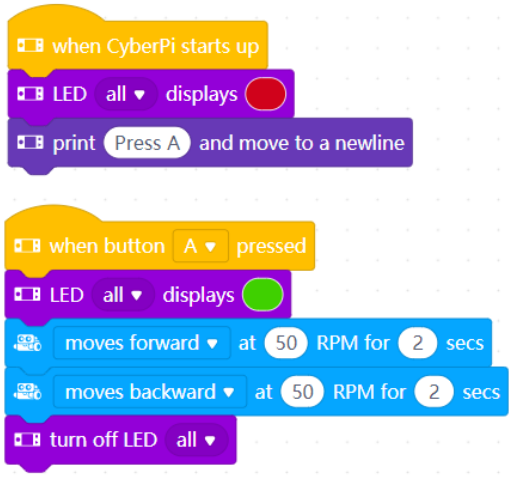
Movement		Commands
Forward or backward forever. (Should only be used when the ultrasonic sensor or colour sensors are used to control when the motors should stop)		
Forward or backward for a length of time		
Forward or backward for a fixed distance		
Turn on the spot for a length of time (wheels turning in different directions)		
Turn for a number of degrees of heading		
Gradual turn for a length of time (wheels turning in the same direction or one wheel stopped)		
Stop motors		

Code Templates

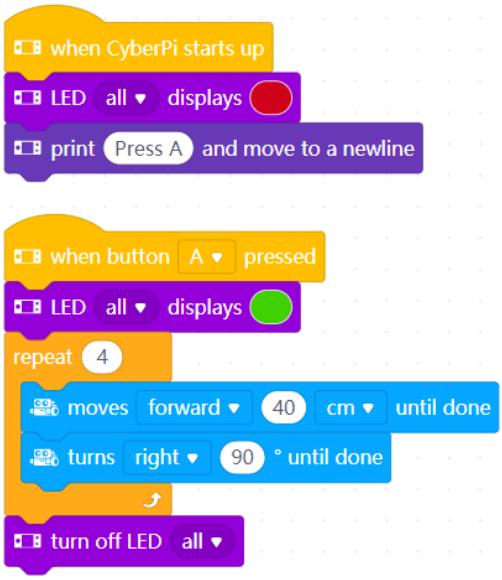
There are two basic code templates we use when running motors. In both cases, we use button A to turn on the mBot2 to start the actions.

Separating code into sections makes it much easier to understand the code and make changes to it. Later, we will add more sections as we require them.

1. **Single Actions**. Use this when the mBot2 actions should only occur once.



If we have actions that are repeated, we can use a **for loop**. For example, to move in a square:

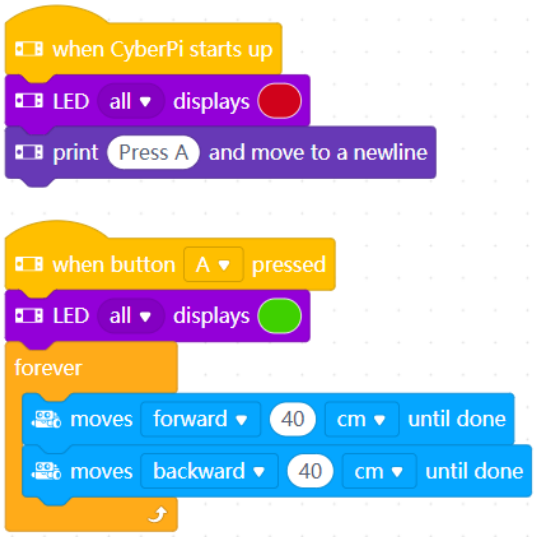


CHALLENGES

1. Place one or more large objects on the floor. Navigate the mBot2 through and/or around them.

2. One of the RoboRAVE competitions is AMAZE-ing. It consists of a series of boards that make up a maze. You do not know the shape of the maze until the competition. The person who keeps the robot on the boards and has the fastest time wins.

2. **Forever Actions**. This code has a ***while True*** loop that repeats the actions forever – or until you press the **home button** next to the USB connection.



This code is mainly used in conjunction with the joystick and buttons, or the ultrasonic and line follower sensors, where the mBot2 will respond to changes in sensor values.

CHALLENGES

3. Place two small objects on the floor at least 1m apart. Drive around these multiple times in a figure of 8. When you turn use the led's to indicate your turns.

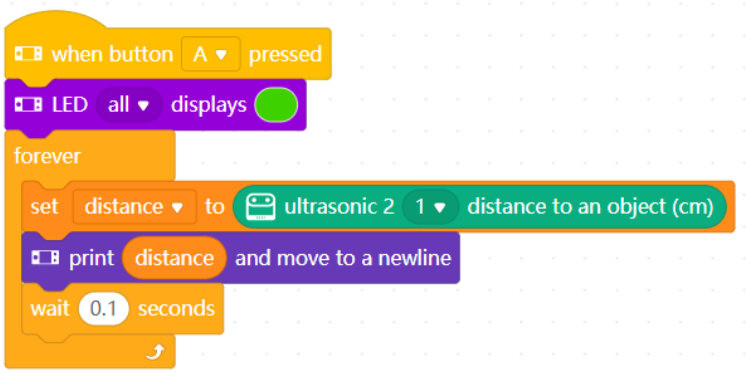
4. Place a large object on the floor and turn around the object 3 times in a large, smooth circle.

F. Avoid or Seek

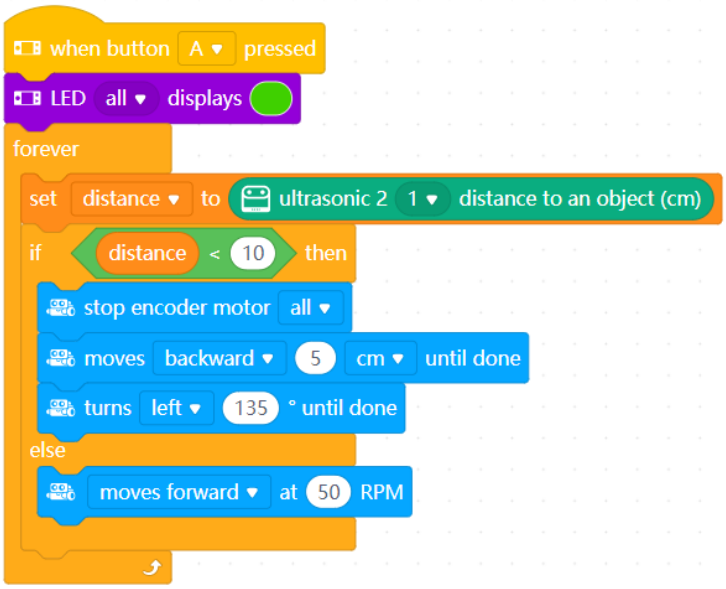
The **Ultrasonic Sensor** is used to measure the distance between the mBot2 and anything in front of it (up to about 200cm). It can be used to avoid obstacles or seek out an object and move toward it.

The minimum distance detected is about 4cm. Smaller distances give a reading of 300.

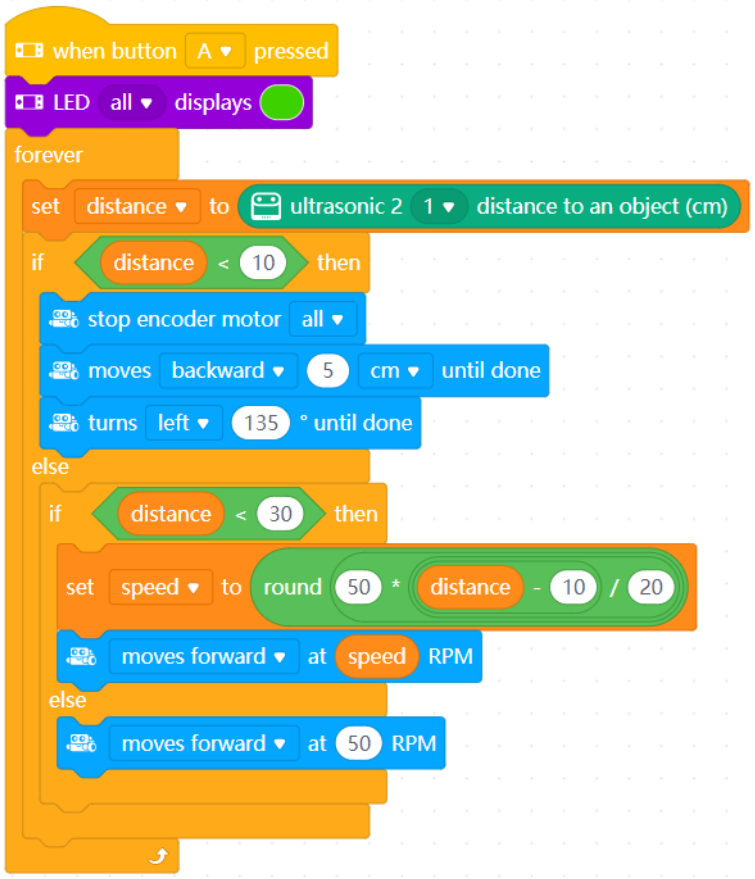
Test your Ultrasonic Sensor with this code. Putting all the sensor reading code into a function unclutters the main loop.



Obstacle Avoidance

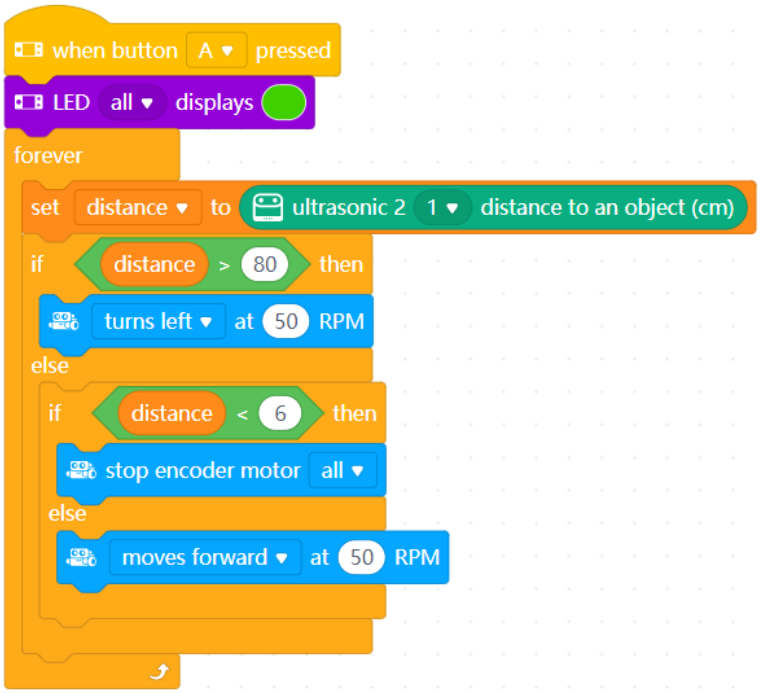


Slow Down when Close to a Collision



Seek Objects and Move Toward Them

Rotate to detect an object closer than 80cm, then move toward the object.



CHALLENGES

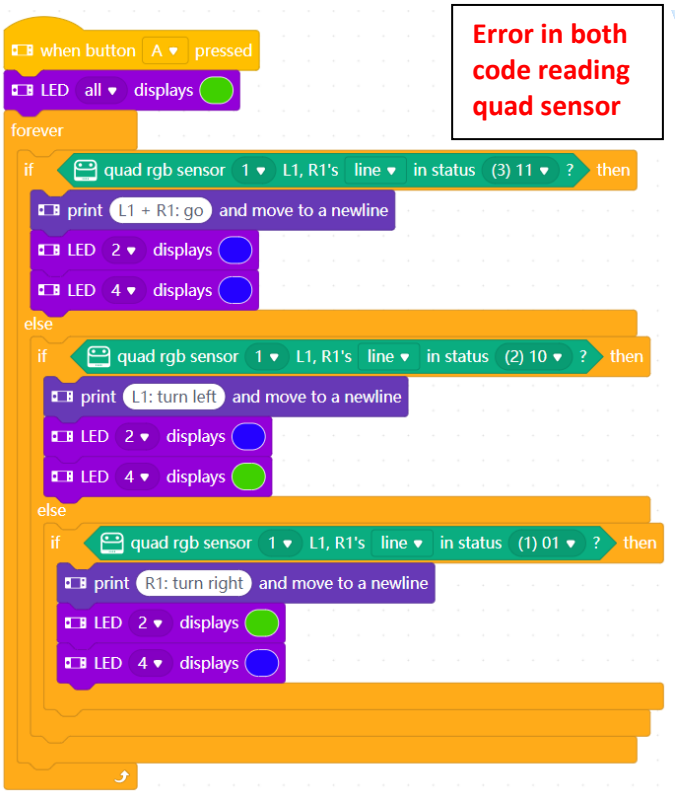
5. Place 4 objects at the corners of a square. Find one of them and stop before you hit it. Turn and find the next object, until you have found all four.

6. Find your way autonomously through a simple maze (sides are 10cm high)

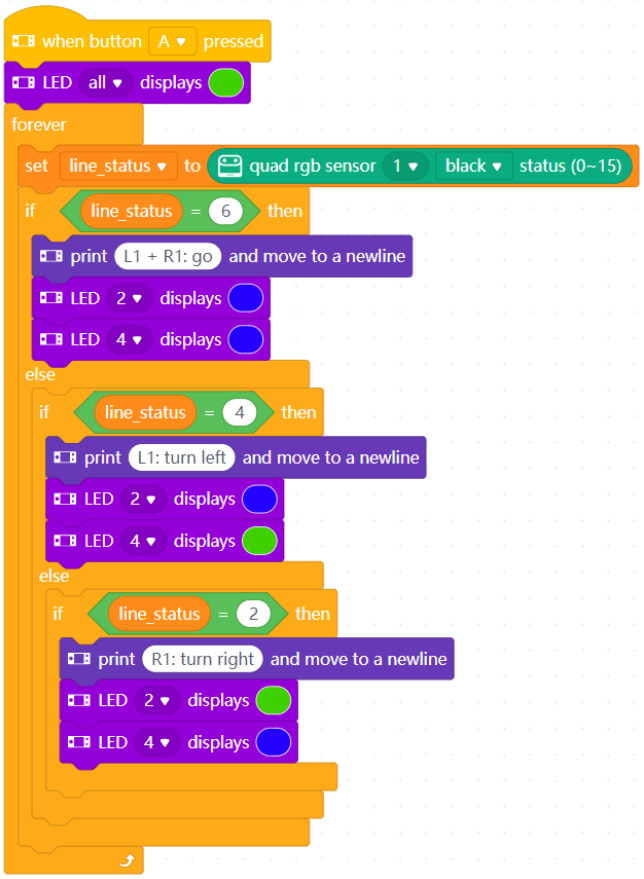
G. Detect and Follow a Line

The **Quad RGB Sensor** (color sensor) enables us to detect and follow lines, and detect colours and respond to the colours in different ways.

Test the Sensor using either of these, by passing the mBot2 over a black line on a white background and checking the displayed message and lights.



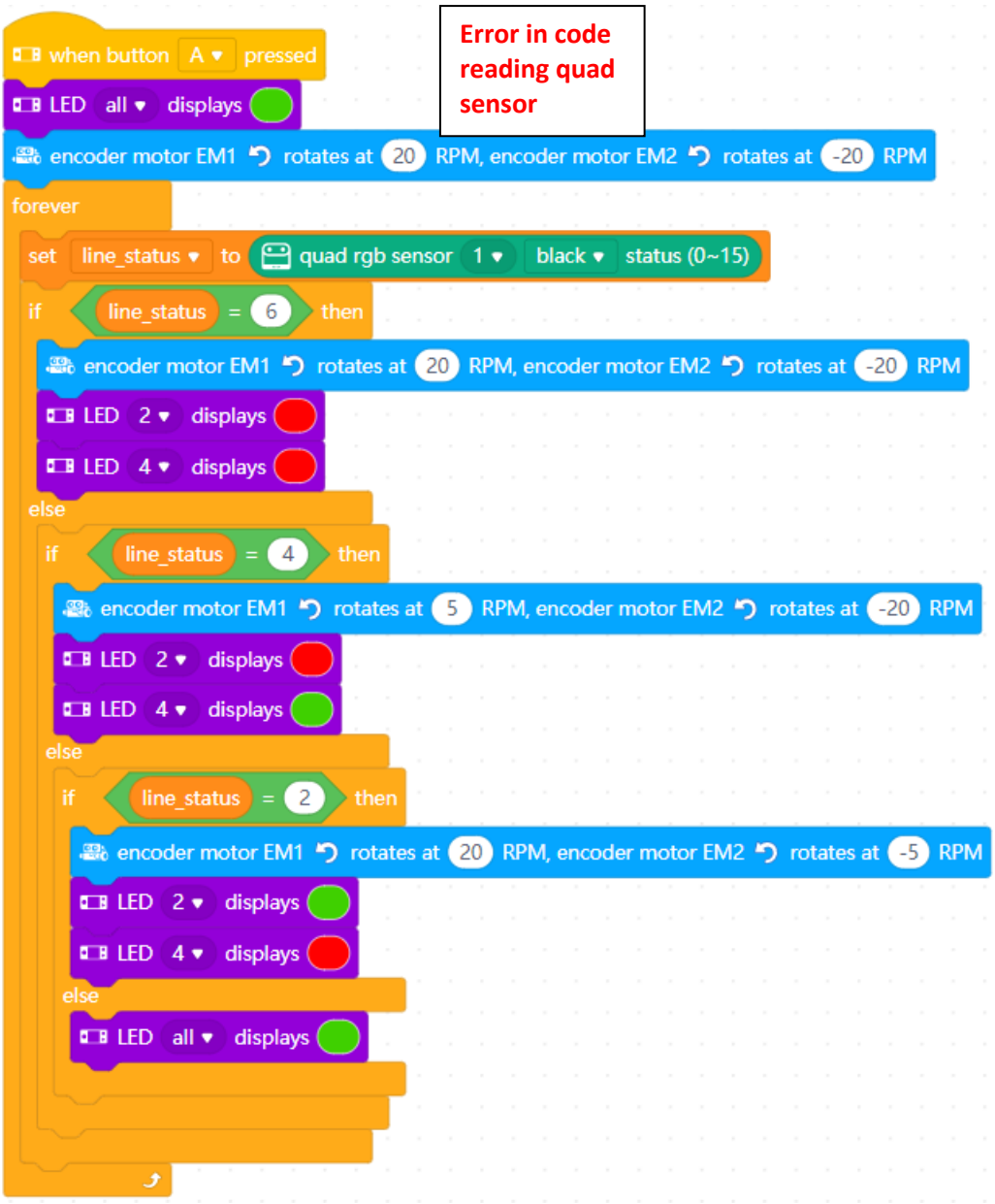
Error in both code reading quad sensor



We can use the color sensor values to test whether the color sensor is on or off a black line.

- On a line will give a low reflectance value or off a line will give a high value.
- Assume for a start that if the reflected light value is less than 50% if we are on or near a black line.
- Place the mBot2 on the middle of the black line
- If both sensors L1 and R1 are on black – go straight ahead
- If only sensor L1 is on black – turn to the left
- If only sensor R1 is on black – turn to the right

First, test the code below without the motors driving. Then take off the comment # and try with the motors running.



To follow the line faster, you might need the change:

- The power to the left and right wheels
- The difference in power between the left and right wheels
- How you interpret the percentage color sensor values
- Use the L2 and R2 sensors as well

CHALLENGES

7. Oval Race. Follow an oval line from start to finish. Time the run. The robot that does the quickest time wins.

8. RoboRAVE Line Follower Race. Be the fastest robot to get from home to the box.

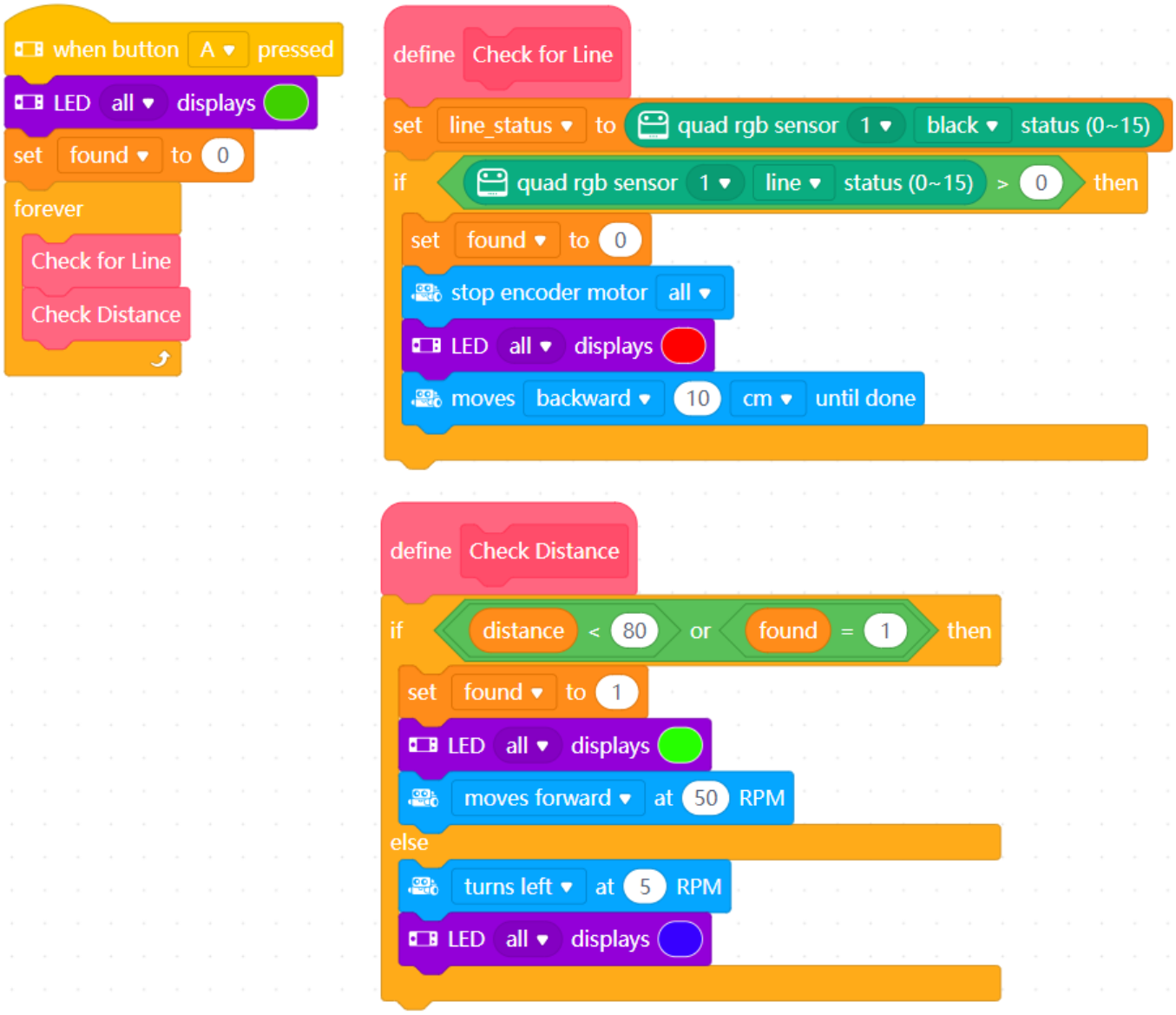
H. SumoBot

SumoBots use the ultrasonic sensor to seek and destroy another robot vehicle in the Sumo ring, while using the color sensor to sense the white border and avoid falling off the edge.

H1. Basic Sumo Code

The basic actions of a SumoBot are:

- A three second wait before doing anything
- Move forward from the edge 20cm
- Rotate until the ultrasonic sensor locates the other vehicle (less than 80cm away)
- Drive full speed toward the other vehicle
- If the white edge is detected (high reflectance value) then stop, back up and rotate to locate the other vehicle



H2. Enhancements

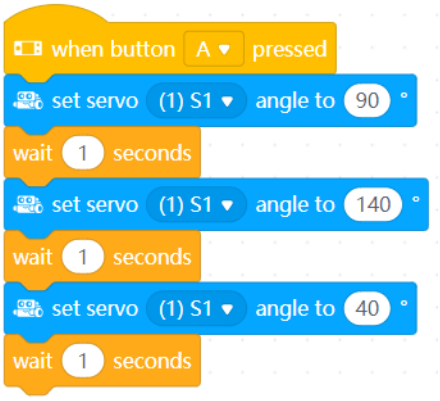
- Only scan left and right up to 90 degrees the first time
- Stop every 10 degrees when scanning to make sure scan detects vehicle (moving too fast doesn't work)
- Use movement sensor to detect a collision or the bot lifted off the ground (pitch or roll) and respond to that (see Appendix 1)
- If motion is stopped for x seconds, use a series of rapid wheel movements (e.g. back and forth) to try and get free
- Use a different strategy:
    - Follow white line around the outside (use L2 or R2)
    - Drive to a random place
    - Drive forward until white line and turn and randomly go somewhere else until white line
- Use more than one ultrasonic sensor at different angles

I. Connect Servos, Sensors and Motors

These blocks are found in the mBot2 Extension Port group.

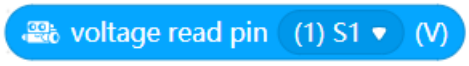
Servos

Up to 4 servos can be plugged in the servo ports on the right-hand side (S3 and S4), or the general IO ports on the left (S1 and S2).

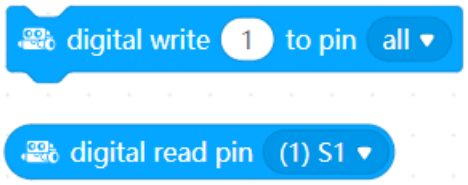


Read Analog Sensors

Read analog sensors (such as potentiometers or soil moisture sensors) using ports S1 and S2



Read and Write Digital Sensors



Run DC motors

Additional motors can be run from the M1 and M2 ports.

```
cpi.mbot2.motor_set(power, port)          #power is -100 to 100
cpi.mbot2.motor_stop(port)


cpi.mbot2.motor_drive(power1, power2)     #set the power to M1 and M2
```

Appendix 1 CyberPi Extras

Ultrasonic, slider (potentiometer) and multi-touch

```
import cyberpi as cpi
import time

while True:
   distance = cpi.ultrasonic2.get(index=1)
   pot = cpi.slider.get()
   touch = cpi.multi_touch.is_touch(ch = 1)    #1-8 or ch = "any"
   print(distance, pot, touch)
   time.sleep(0.1)
```

Light sensor

```
light = cpi.get_bri()
```

Sound sensor

```
volume = cpi.get_loudness(mode = "maximum")
```

Audio Commands

```
cpi.audio.play_tone(freq, t)
cpi.audio.add_vol(val)                #-100 – 100
```

Accelerometer/Gyro Commands

```
forward = cpi.is_tiltforward()
backward = cpi.is_tiltback()
left = cpi.is_tiltleft()
right = cpi.is_tiltright()
```

```
cpi.is_shake()
cpi.get_shakeval()      #0-100

cpi.get_pitch()         #pitch angle
cpi.get_roll()          #roll angle

cpi.get_yaw()           #yaw angle
cpi.reset_yaw()
```