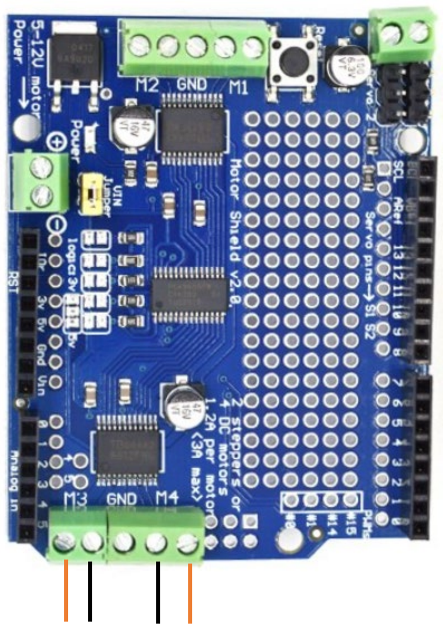
Collision Avoidance Vehicle

Metro M4 with Motors/Ultrasonic Sensor

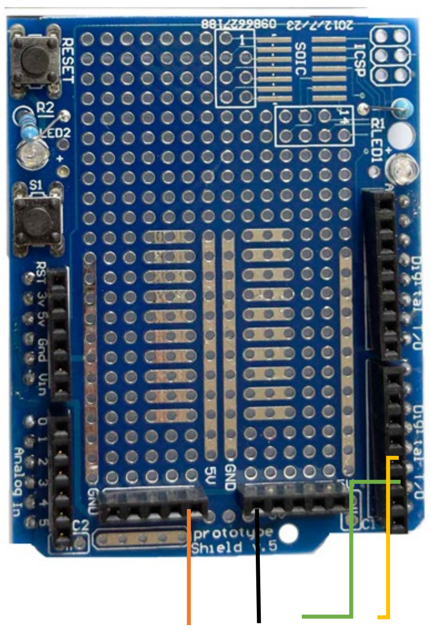
Barry Butler
bbutl58@eq.edu.au

A. Metro M4 Connections

Connect the servo's and ultrasonic sensor to the CPX as shown.

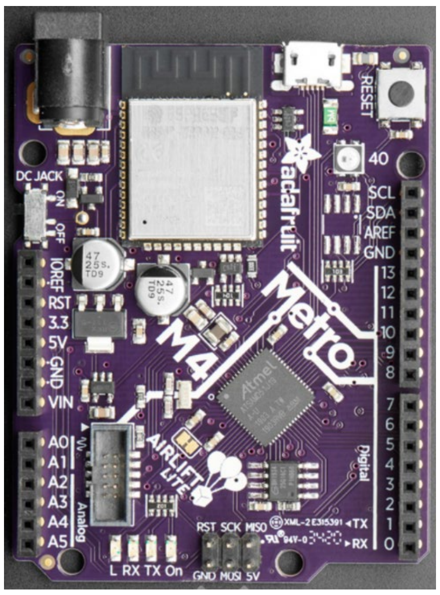


Motors



Ultrasonic Sensor

Metro M4 Connections (with Motor Driver Shield)

A0				
A1			LED	
A2			L (red)	
A3				
A4 (SCL)			SCL	
A5 (SDA)			SDA	
Motor 1			D13	On/off switch
Motor 2			D12	
Motor 3	TT Left		D11	
Motor 4	TT Right		D10	
			D9	
I2C 1			D8	
I2C 2			D7	
I2C 3			D6	
I2C 4			D5	
			D4	
			D3	Ultrasonic echo
			D2	Ultrasonic trigger
			D1	
			D0	

Coolum State High School
Care • Respect • Excellence

ROBOCOAST
Sunshine Coast Robotics
www.robocoast.tech

B. CircuitPython Basic Code

The following code is used for all CircuitPython Code. Type this code into the Mu editor (in **CircuitPython mode**).

```
#IMPORTS------------------------------------------


#CONNECTIONS--------------------------------------


#GLOBAL VARIABLES---------------------------------


#FUNCTIONS----------------------------------------


#PRESS BUTTON TO START----------------------------


#MAIN LOOP------------------------------------------
while True:

   time.sleep(0.1)
```

EACH TIME YOU ENTER A SECTION OF CODE, CLICK THE "SAVE" BUTTON TO TEST IT, AND FIX ANY ERRORS.

- **Select the CIRCUITPY Folder**
- **Always Save Using the Filename – code.py**

C. Import the Required Libraries

We need to import the MotorKit, Neopixel, Digital Input and Ultrasonic Sensor libraries.

```
#IMPORTS------------------------------------------
import time
from adafruit_motorkit import MotorKit
import board
import neopixel
import adafruit_hcsr04
from digitalio import DigitalInOut, Direction, Pull
```

D. Setup All the Connections

Now we need to setup the connections between the motors / sensors and the Metro M4 pins. We have two sensors – a switch and an ultrasonic sensor. We can change the colour of the neopixel (led) to tell us things that are going on when we run the vehicle.

```
#CONNECTIONS-------------------------------------
switch = DigitalInOut(board.D13)
switch.direction = Direction.INPUT
switch.pull = Pull.DOWN

kit = MotorKit()
sonar = adafruit_hcsr04.HCSR04(trigger_pin=board.D2, echo_pin=board.D3)

led = neopixel.NeoPixel(board.NEOPIXEL, 1)
led[0] = (255, 0, 0)
```

E. Global Variables

We need a couple of global variables, to store the vehicle motor state (on or off) and the ultrasonic sensor distance.

```
#GLOBAL VARIABLES------------------------------------
vehicle_on = False
distance = 999
```

F. Create a Function to Change the Speed of Both Motors

Because we are running 2 motors, it simplifies the code by creating a function to control both motors at the same time.

```
#FUNCTIONS---------------------------------------
def run_motors(s1, s2):
    kit.motor3.throttle = s1                        #left motor
    kit.motor4.throttle = s2                        #right motor
    if (s1==0) and (s2==0): led[0] = (255, 0, 0)
    else: led[0] = (0, 255, 0)
```

G. Create Functions to Read all the Sensors

The best way to run the vehicle is to read all the sensors for every iteration of the main loop. The first functions is one to read the ultrasonic sensor.

```
#FUNCTIONS---------------------------------------
. . .
def read_sonar():
    try:
        d = sonar.distance
    except RuntimeError:
        d = 999
    return d
```

The second function calls the first function and also checks if the button switch has been pressed so we can stop the motors if required. Pressing the button toggles the motors on again.

```
def read_all_sensors(show_all):
    global distance, vehicle_on

    if switch.value:
        vehicle_on = not vehicle_on
        if vehicle_on: print('Running')
        else: print('Stopped')

    distance = read_sonar()
    if show_all: print(distance)
```

H. Wait for the Button to be Pressed to turn the Vehicle On

We don't want the motors to start running as soon as we download the code the Metro M4. So write the code to check if the button has been pressed.

```
#PRESS BUTTON TO START----------------------------
run_motors(0,0)               #make sure motors are turned off before starting
print('Waiting to start')
while not switch.value:
    time.sleep(0.1)
vehicle_on = True
```

I. Make the Vehicle Move and Avoid Obstacles

The main loop firstly reads all the sensor values. If the vehicle motors are on, the code then checks the ultrasonic sensor distance. If the distance is less than 10 cm the motors stop, reverse and turn to point in a new direction.

```
#MAIN LOOP------------------------------------------
print('Running')
while True:
    read_all_sensors(True)

    if vehicle_on:
        if distance > 10:
            run_motors(0.6,0.65)
        else:
            run_motors(0,0)
            time.sleep(0.4)
            run_motors(0,0.5)
            time.sleep(1)
            run_motors(0,0)
            time.sleep(0.4)
    else:
        run_motors(0,0)

    time.sleep(0.1)
```