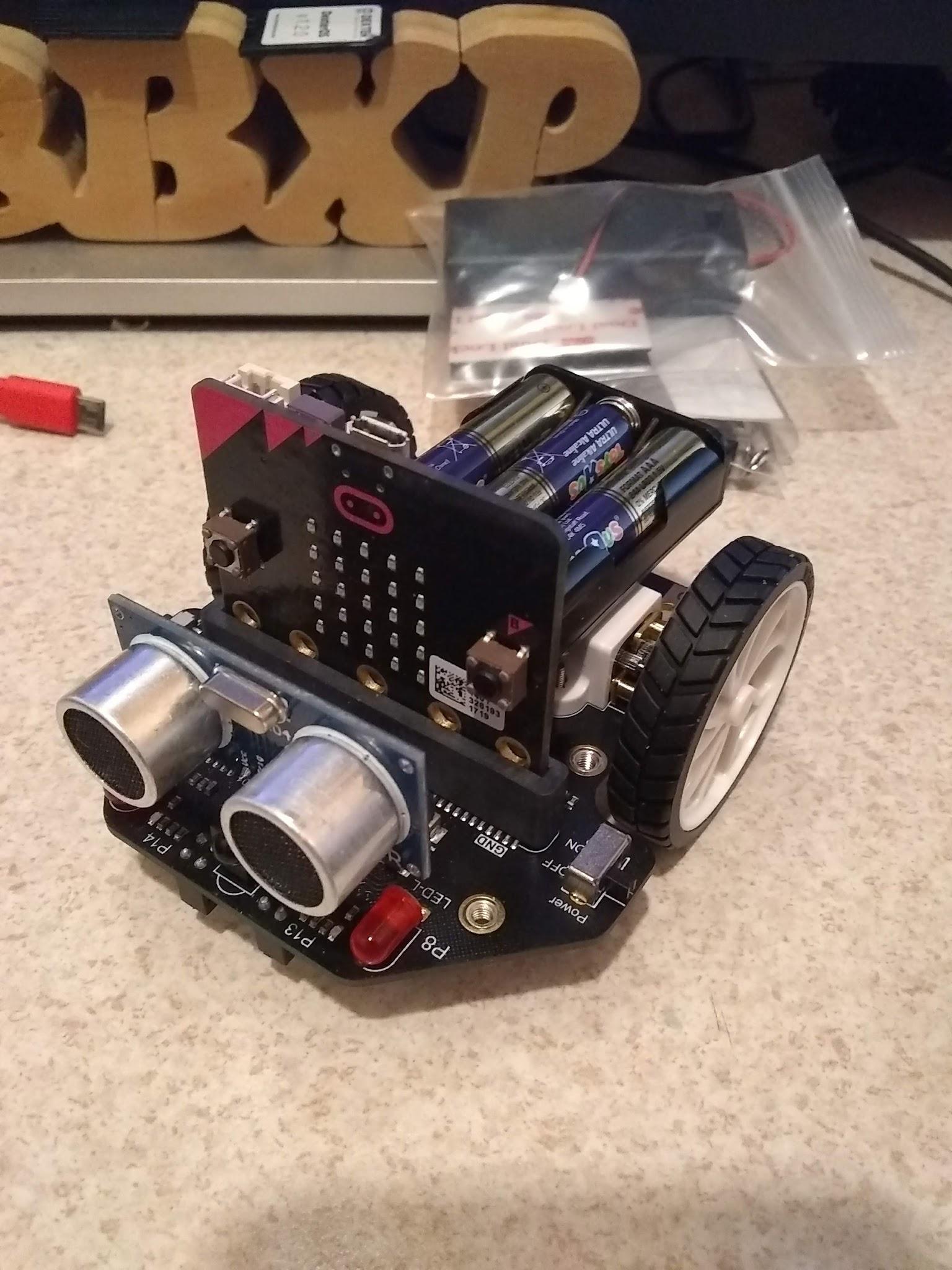
DF Robot micro:Maqueen Robot

For BBC Micro:Bit (programmed with the Arduino IDE)

Sep 10, 2018



Prerequisites:

1. Arduino IDE 1.8.x (recommend 1.8.5) installed and setup for your machine. <https://www.arduino.cc/en/Main/Software>
2. Install the Arduino-nRF5 board core. Follow the instructions here: <https://github.com/sandeepmistry/arduino-nRF5>
3. Install the Adafruit Micro:Bit Library: <https://github.com/adafruit/Adafruit_Microbit> and learn more about it here: <https://learn.adafruit.com/use-micro-bit-with-arduino>
4. Finally install the arduino-BLEPeripheral library (either from the library manager or by following instructions here: <https://github.com/sandeepmistry/arduino-BLEPeripheral> ) \* Note be sure it’s the one from sandeepmistry if you are using the library manager \*
5. SparkFun Mag3110 Compass library. <https://github.com/sparkfun/SparkFun_MAG3110_Breakout_Board_Arduino_Library>
6. NewPing Library (Which I believe is this library) <https://github.com/eliteio/Arduino_New_Ping> ~~and is only needed if you are going to use an ultrasonic.~~

For this project you will need at least one Micro:Bit Controller:

<https://www.dfrobot.com/product-1625.html> or <https://www.dfrobot.com/product-1587.html>

And the Maqueen Robot chassis: <https://www.dfrobot.com/product-1783.html>

The Micro:Bit is setup to use Microsoft:MakeCode or Scratch and Python.

The Sandee P Mistry Board core lets us use this board with the Arduino (C style) language as well.

General Information:

I used the general information that is found either in the Maqueen MakeCode library on the board itself, or from trial and error to come up with something that works (mostly) with the Arduino IDE.

<https://www.dfrobot.com/wiki/index.php/Micro:Maqueen_Robot_Car(V2.0)_SKU:_ROB0148#Ultrasonic_Obstacle-avoiding> and the library here:

<https://github.com/jhlucky/maqueen/blob/master/maqueen.ts>

**Working (or mostly working):**

1. Motors (I2C interface, easier to port than I thought it would be)
2. Line Sensors (P13, P14)
3. Ultrasonic (using newping library) P1, P2 \* Attempted to get this working without the library, but it failed (????) \*
4. 2 LEDs P8, P12
5. Buzzer P0 - \* tone is not yet implimented with the board, analogWrite works but doesn’t sound great, bit bang also works, but doesn’t sound great - this is in the mostly working columb)

### **What’s not working:**

(These things work if you use MakeCode, but so far I’m unable to get them to work with Arduino).

1. NeoPixels (P15 4 pixels) - The Adafruit Neopixel library appears to be the only library that supports the Micro:bit, this isn’t surprising. But I’ve not been able to get it work on P15, it does work on P2 which is what the example provided is using. <https://learn.adafruit.com/micro-bit-lesson-3-neopixels-with-micro-bit/software>
2. IR receiver (P16) - so far I am unable to get this to work at all - \*I need something that doesn’t mess with interrupts or timings, but I think that is a big ask for this device\*
   1. Sep 12 - found a very old way to get this to work using pulseIn, it does work but isn’t a great way to do this, it is also not accurate. But maybe useable until something else comes along better. \* For now I’ll still list this as not working at least not working 100% \*
3. Communication between two Micro:bit controllers - It’s a bit upsetting to me that such a basic thing was neglected in the Arduino IDE (If this does work, and you have supporting code please share - I have not been able to get this to work, or find any code that claims it works) \*\* This is a basic thing in the MakeCode, I’ve not been able to find the library they use or information about how they make it work \*\* <https://learn.adafruit.com/micro-bit-radio-controlled-puppet/code-the-two-micro-bits>

* It should be noted here that I am able to connect a Micro:Bit to a Android tablet and send and receive information using that. But It has to use the Adafruit Bluefruit LE Connect app to do it - <https://play.google.com/store/apps/details?id=com.adafruit.bluefruit.le.connect>

Basic Sketches: (To be duplicated)

1. Drive and turn - (done) I2CMotortest, I2CMotortest2
2. \*RGB Breathing Ambient Light - unable to complete at this time.\*
3. LED Light Flash - (DONE) - blinkdemo
4. Read Ultrasonic Distance (\*DONE\*) - They output to the LED Matrix, I output to the Serial console of the Arduino IDE (Easy to change as needed) newping
   1. Sep 12, new sketch which displays the distance on the LED Matrix
5. \*Read Infrared Key Assignments - unable to complete at this time. \*
   1. Sep 12, 2018 - IR is sort of working using a old method of pulseIn to read the remote values. This works, but is not very good or accurate.
6. \*IR Remote Control - unable to complete at this time\*
   1. Sep 12, 2018 - IR is sort of working using a old method of pulseIn to read the remote values. This works, but is not very good or accurate.
7. Line Tracking - Currently Working on it.
8. Ultrasonic Obstacle Avoiding - (DONE) - avoid
9. Light Operated Sprite (unsure about this one)
10. Wireless Remote Control - \* They are using a Micro:bit Gamepad (see below), I believe I can make this work with the Tablet and Adafruits Bluefruit LE Connect software \* This is a work in progress.

The compass and accelerometer on the Micro:bit also work, as well as the LED Matrix and the two buttons on it. \*The Compass is in the wrong orientation to be used as a true compass, but it maybe able to be used to do relative positions, I will have to test that more.\*

I am in the process of making a library and attempting to duplicate the applications in the DF Robot wiki. It is not even close to being ready. I’ll post a link when it is ready.

DF Robot also provided a Micro:Bit Gamepad for use with this project, the gamepad has 8 buttons total (various P#s), a LED on P16, buzzer on P0, and a vibration motor on P12

Button assignments:

DOWN - P13

LEFT - P14

UP - P8

RIGHT - P15

X - P1

Y - P2

A - P5

B - P11

<https://www.dfrobot.com/product-1711.html>

This was very easy to port over to the Arduino IDE, the buttons are all HIGH until they are pushed then they go LOW.

I may not be able to use this as intended because I am having problems getting two Micro:bit to communicate directly with each other.

A game or something may be coming thou...we will see

# Updates and some more information: 9/11/18

Information that explains BLE usage and terms and sketches:

(Note: This is for the Arduino 101 Curie Board, but the general information should be the same)

<https://www.arduino.cc/en/Tutorial/Genuino101CurieBLELED>

<https://www.arduino.cc/en/Tutorial/Genuino101CurieBLECallbackLED>

<https://www.arduino.cc/en/Reference/CurieBLE>

Really good information for Micro:Bit and Blynk

<http://help.blynk.cc/how-to-connect-different-hardware-with-blynk/arduino/bbc-microbit-arduino-ide>

C++ Code (and library) for use on mbed website with Micro:Bit

<https://github.com/lancaster-university/microbit-dal/>

(Someone smart might be able to use this information and port it over to an Arduino Library (?))

(?) I’m not sure if this is helpful or not: <https://github.com/RedBearLab/nRF51822-Arduino/blob/S130/arduino-1.6.x/hardware/RBL/RBL_nRF51822/libraries/nRF_Examples/examples/Ticker_Task/Ticker_Task.ino>

Other GPIO with Micro:Bit (Which maybe referenced above)

<https://learn.adafruit.com/use-micro-bit-with-arduino/other-gpio>

Python I/O Pins: <https://microbit-micropython.readthedocs.io/pl/latest/pin.html>

mBed uBit.Radio information: (This is part of the Micro:Bit-DAL library above)

<https://lancaster-university.github.io/microbit-docs/ubit/radio/>

Using a nRF24 on an Arduino and a Micro:Bit to send changes - It doesn’t work the way “we” want it too, but it made the light-bulb go off that the Micro:Bit is using the RADIO and not BLE to do peer to peer stuff.

<https://www.youtube.com/watch?v=743c329t1xs>

Code from above video: (Micro:bit) <https://pastebin.com/ZLYRCLhy>

(Arduino) <https://pastebin.com/nCT5D3xB>

# Sep 12, 2018 Update:

1. Basic Library written. Loosely based on the BittyBot2 library that I wrote, thou this works quite a bit differently. This has all the drive functions working, added some sound effects (unfortunately they are blocking, so they should only be used when the motors are off). Added a smile\_bmp from Adafruit for use with the LED Matrix.
2. Found a very old method for reading IR remotes using the pulseIn function. <http://pscmpf.blogspot.com/2010/01/arduino-and-ir-receiver.html> This mostly works, it’s not as good as some of the modern libraries, but does work. One of the biggest issues is the same button may give three or four different codes - however I believe it’s usable. You can see this example is from 2010.
3. Wrote some example sketches based on the current version of the library.

**Still not working:**

1. Peer to Peer communications even though I was able to find some information about this I’ve still not had any luck getting it to work.
2. Neopixels on Pin 15 - this just seems like there is a problem with either board core (how it deals with the LED Matrix seems to be the problem. ) or it maybe a limitation of the Neopixel library. \* I have tried a couple other WS2812 libraries, and none work (most don’t compile) Adafruit's does work and compile on Pin 2 and a couple of other pins.

### Sept 14, 2018 - More Updates.

IR Remote demo created, it works, but is about as you would expect for a IR remote.

\*This has not been added to the library, but plans on to add it later\*. Ambient light sensor code was written, it appeared to work, but I’m not sure that it really is doing what I thought it was. The Micro:bit uses the LED Matrix in reverse bias (mode) to make this work. I tried this with the LEDs on the robot, thinking I could make this turn left and right - I also tried it with the LED Matrix (as I understand how it works), the LED Matrix appeared to work a bit better.