Bay Coding Club

Robotics

Lesson Plan

By: Paul Richards 1/21/2021

Legos and robotics

What could be better than adding electronics to legos? This will combine electronics with lego devices and vehicles.

Legos has a variety of motors:

Like power functions:



And NXT:



But the remote control features are limited to infra-red:



And the mindstorm brick:



This course will leverage the large number of lego projects currently available online with the new user-friendly microcontroller: micro-bit.

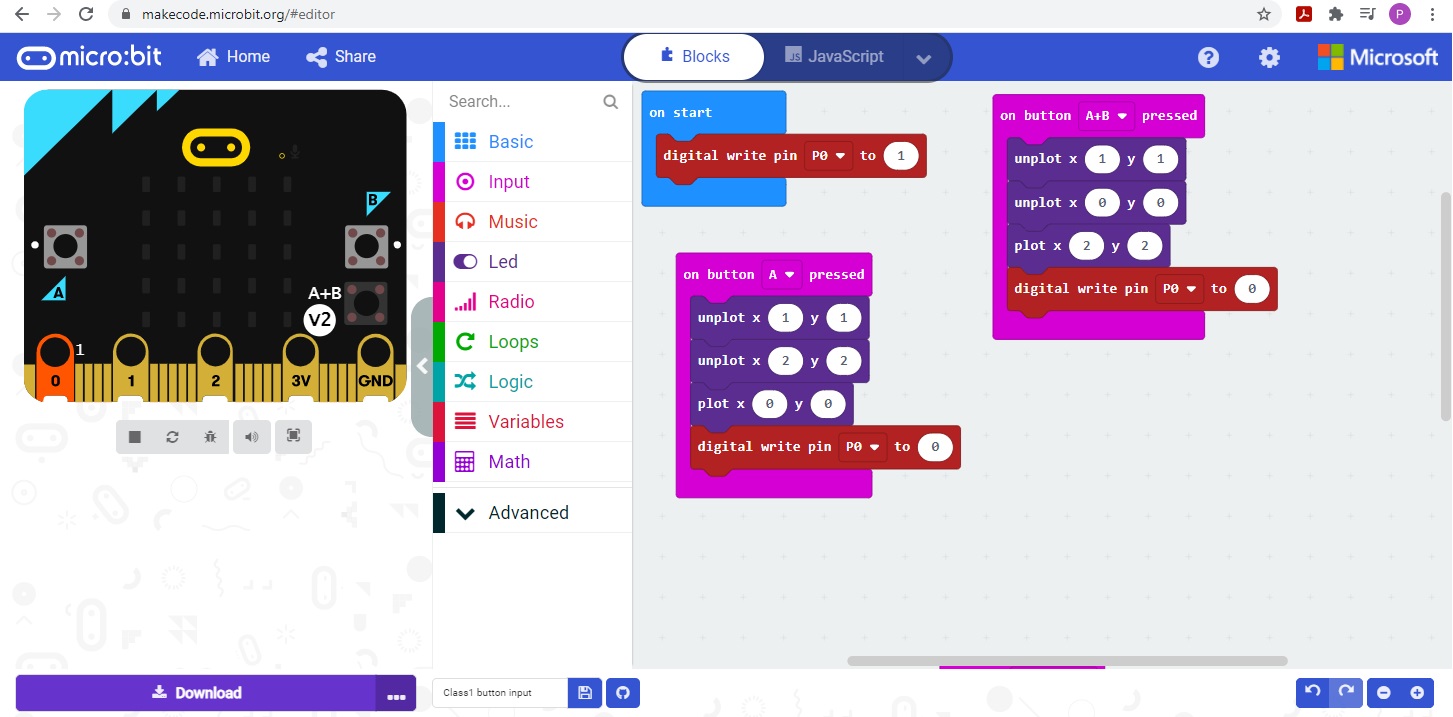
Micro-bit was developed in the UK with young students in mind. Using an intuitive blockly development environment and a rich set of examples, students can easily build working robots.

Our first project will introduce the microbit platform. Next, we will build a mecanum vehicle (which can drive sideways), and then we will build a spider and walking biped. Finally we will add infra-red receive and transmit to our models and explore adding our own custom components.

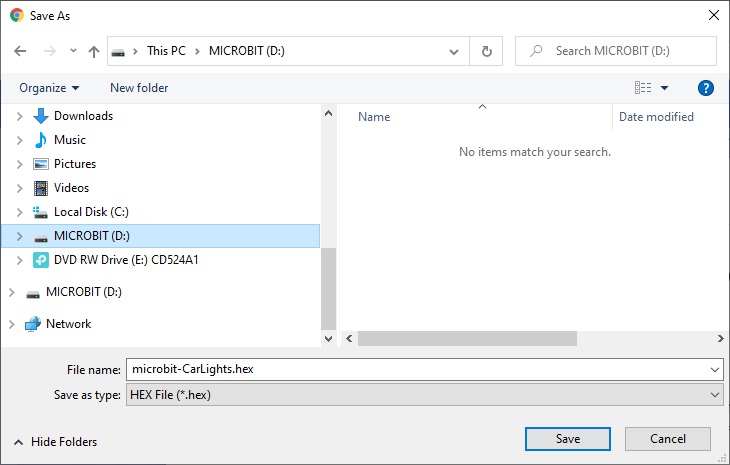
Project 1: Microbit Development Environment

We start with the micro-bit development environment, because it gives students a foundation they can build on for all subsequent projects. It also gives a chance to make sure all parts are on order for the subsequent projects.

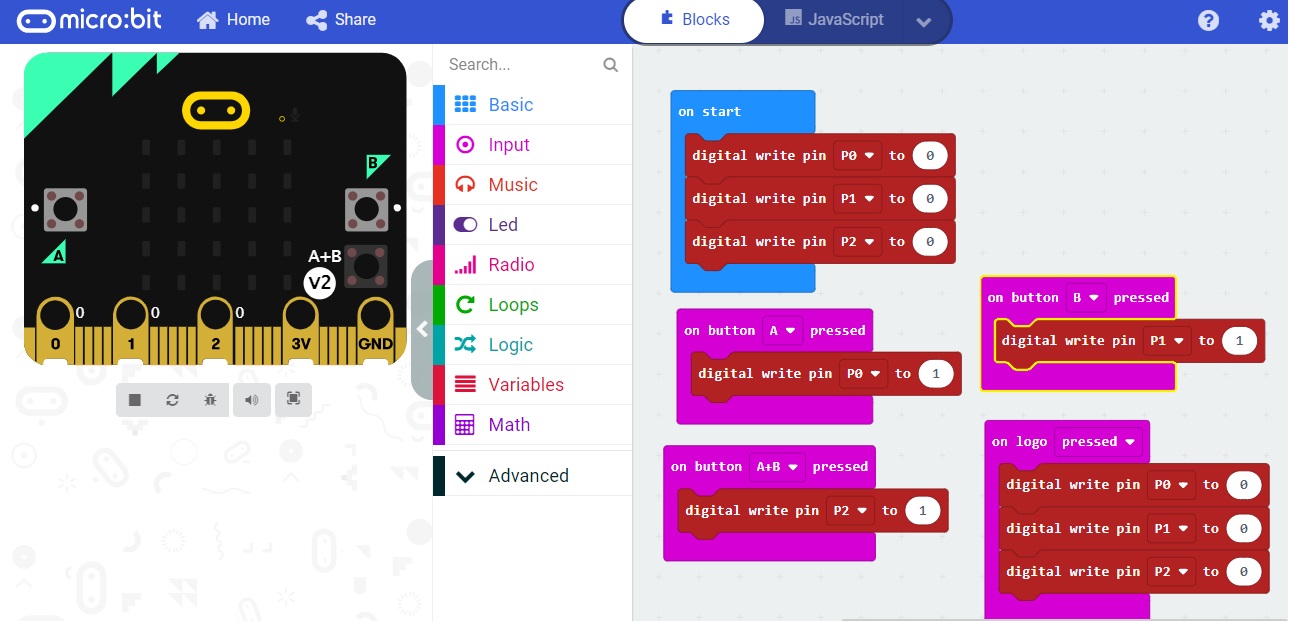
Class 1   
  
 Microbit code development

Microbit devices are programmed using a web-browser online at site: <https://makecode.microbit.org/>Your project will be saved online. You can code a blockly project by **connecting blocks in the Blocks Tab:**  


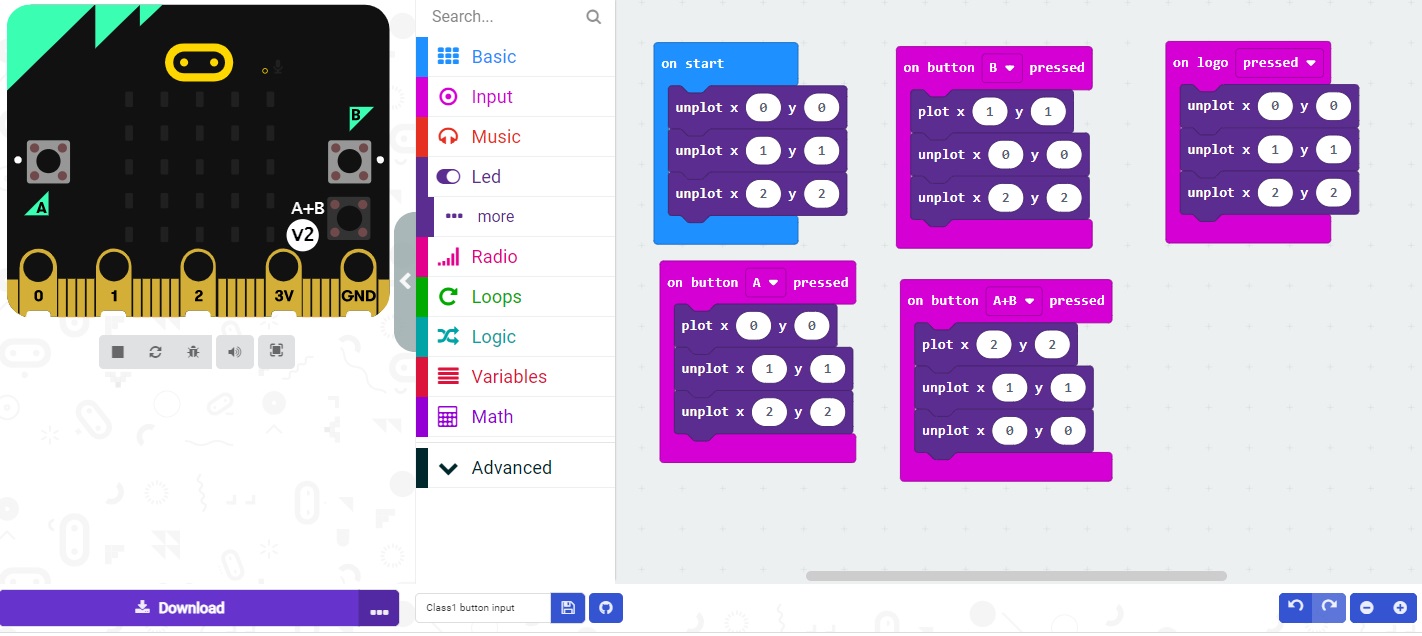
### and Downloading code to the microbit

You download software to the microbit device by connecting the microbit via usb to your personal computer then press the download button on the web-browser and then select the micro-bit drive  
  
The microbit device will start to blink. When it finishes blinking, the download is complete.

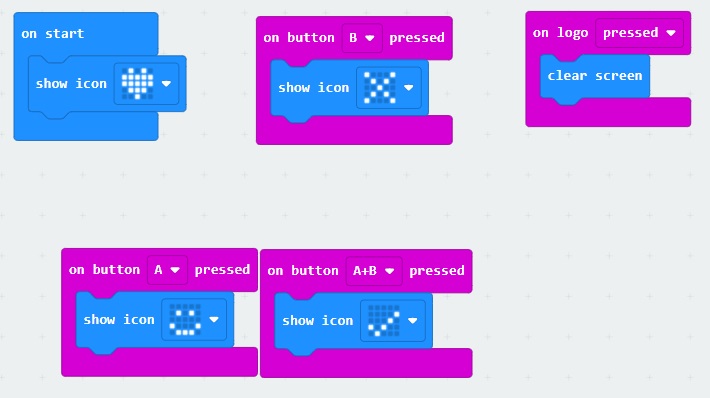
# Class 2 Virtual Environment, Outputs

You can also test your code by exercising it in the virtual/simulated microbit  
Create this code:  
  
Then press buttons on the left and see the result

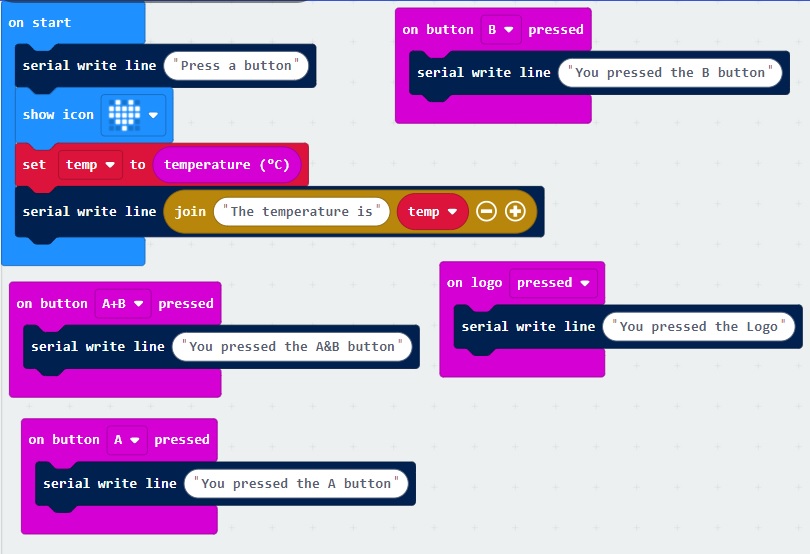
# Class 3 LEDS

The microbit comes with an array of leds that you can use to draw simple characters  
or display messages one character at a time  
You can also test your code by exercising it in the virtual/simulated microbit  
Create this code:  
  
Then press buttons on the left to see the result

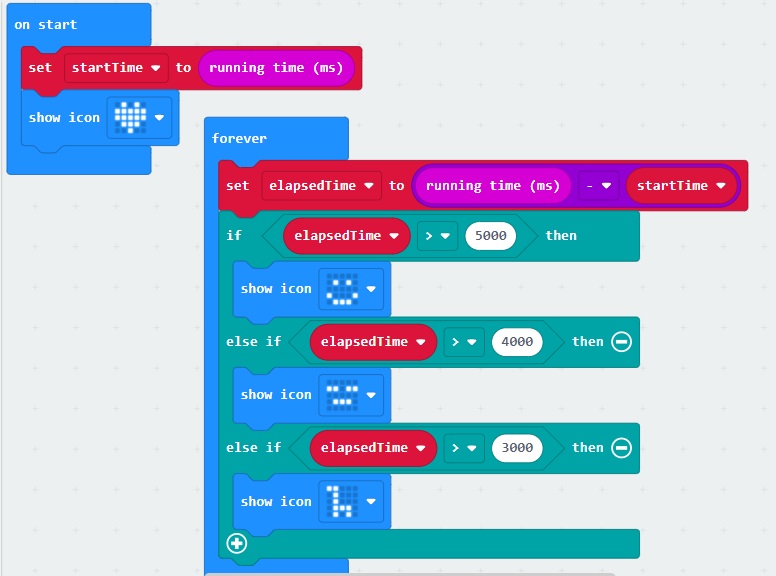
# Class 4 ICONS

The microbit comes with an array of leds that you can use to draw simple characters  
or display messages one character at a time  
You can also test your code by exercising it in the virtual/simulated microbit  
Create this code:  
  
Then press buttons on the left and see the result

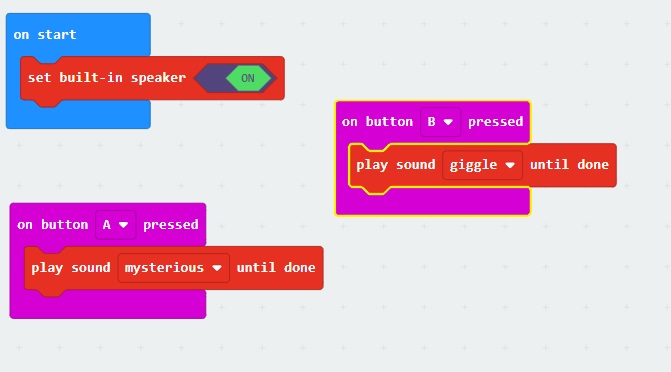
# Class 5 Serial Output

The microbit lets you output text strings to a console. This is useful for debugging programs.  
Create this code:  
  
Press the  button  
Then press buttons on the left to see the result

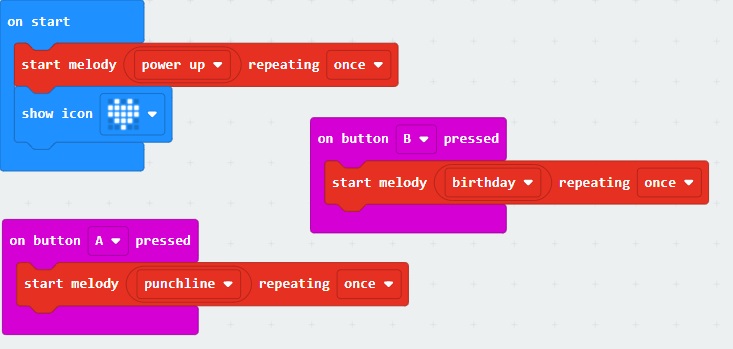
# Class 6 If Statements

If statements are found in every programming language  
Can you guess what this code does?  
  
Create this code and watch the simulator to see the actual behavior

# Class 7 Sounds

Sounds are a good-way to give feedback to the user and keep them engaged.  
Here is some example sound code:  
  
Change the values around to try various songs

# Class 8 Melodies

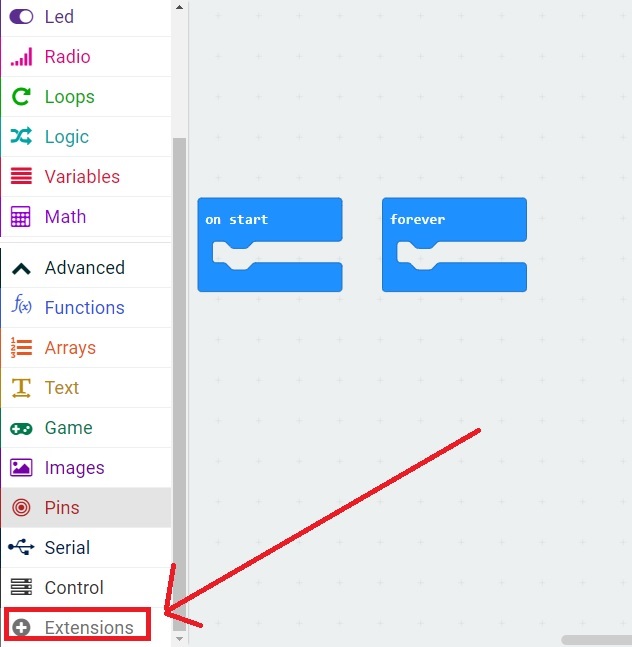
Sounds are another good-way to give feedback.  
Here is some example melody code:  
  
Change the values around to try various melodies

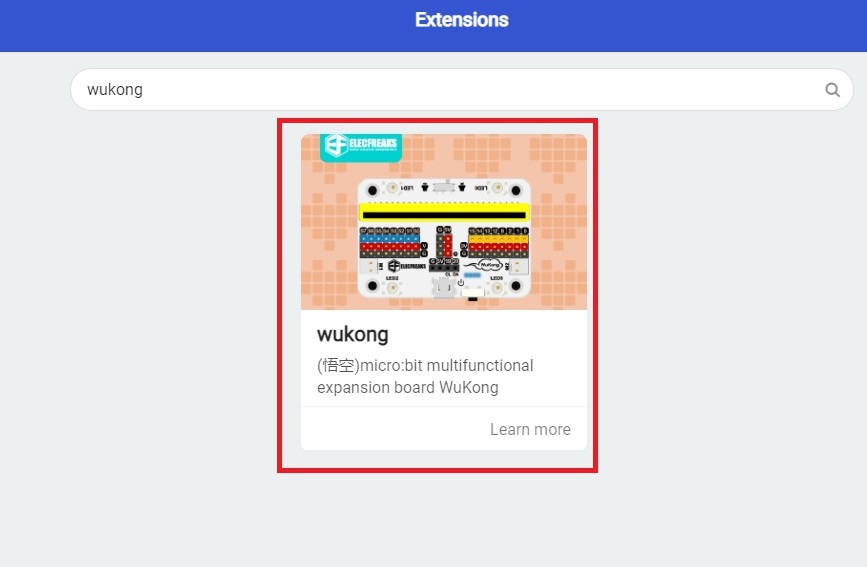
Project 2: Mecanum Vehicle and Joystick


Class 9  
 The assembly of this vehicle is described in the document: <https://paulware.github.io/BCCRobotics/mecanumAssembly.pdf>

# Class 10 Wukong Breakout Board

The wukong breakout board provides power and allows the micro-bit controller to control the motors  
  
It is discussed here: [wiki](http://www.elecfreaks.com/learn-cn/microbitExtensionModule/wukong.html)We will be writing code to interface with this board. But first you need to load the extension.  
To load the extension in micro-bit IDE, Open and Advanced, and click Extensions:  


Then in the search box enter this name: wukong  
  


We will be writing code to exercise this device.

# Class 11 Micro-bit radio

## Each micro-bit comes with its own radio. Two micro-bits can pass text back and forth when set to the same group. Here is a routine which responds to sends the name of the button pressed to the other radio. It will also print out any radio message on the console.

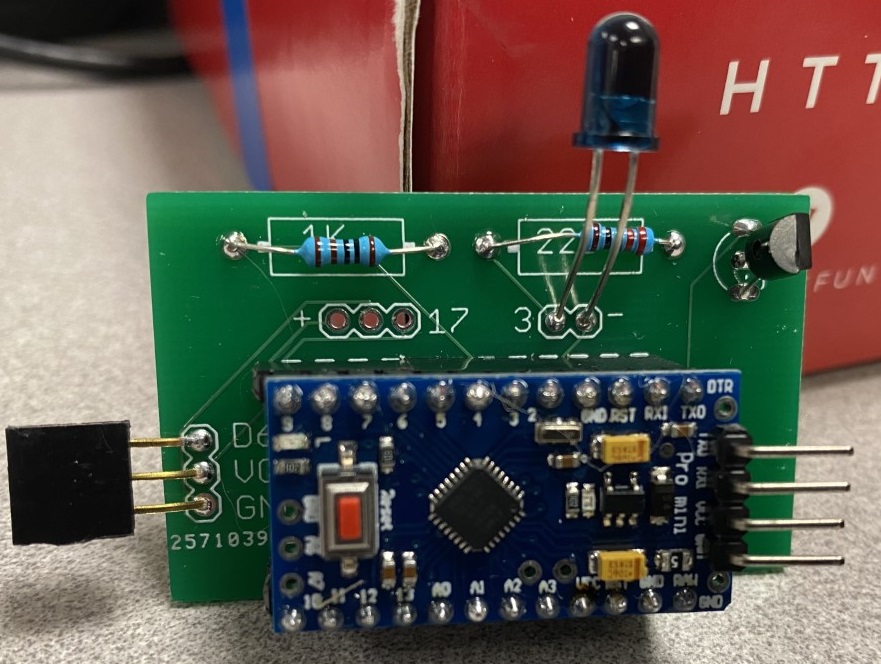
# Class 12 Joystick

The joystick X and Y direction are read via the analog inputs.  
This code will read the X and Y of the joystick and send an associated radio message to indicate direction  

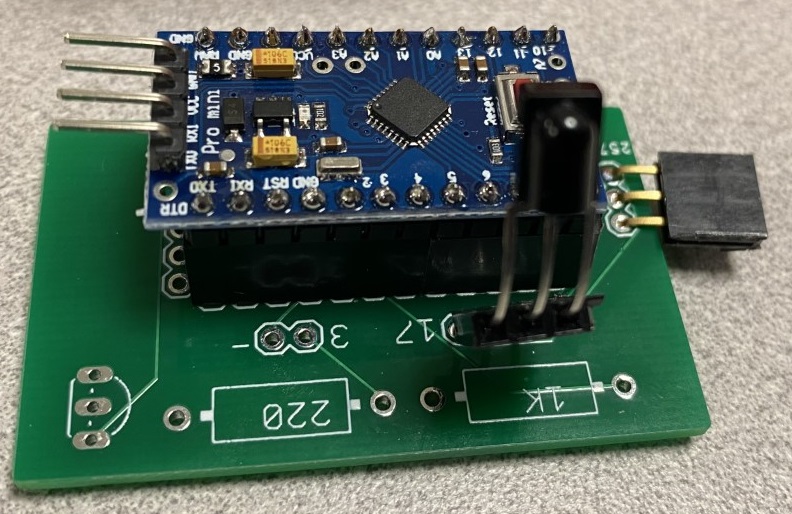

# Class 13 Joystick Buttons

This code will read the joystick buttons and send an associated radio message to indicate when to fire  


# Class 14 nfra-red transmit

An infra-red transmit is accomplished by using a printed-circuit board which looks like:  
  
The connecting wire is similar to a servo 3-wire connector (gnd-vcc-data)  
This is connected to a digital output on the micro-bit. When the output transitions to 1 (energized),  
the infra-red component transmits its pulse

# Class 15 Infra-red Receive

An infra-red receive is accomplished by using a printed-circuit board which looks like:  
  
The connecting wire is similar to a servo 3-wire connector (gnd-vcc-data)  
This is connected to a digital output on the micro-bit. When the ir receiver detects a pulse, it sets the  
digital output to a 1

Class 16

Joystick/Car

The joystick and car are integrated together by using the radio to pass commands to the car:

Mecanum Car  
This javascript can be pasted directly into the makecode ide javascript tab:

input.onButtonPressed(Button.A, function () {

wuKong.mecanumRun(wuKong.RunList.rear, 56)

strip.showRainbow(1, 360)

})

radio.onReceivedString(function (receivedString) {

if (receivedString == "F") {

wuKong.mecanumRun(wuKong.RunList.Front, 80)

} else if (receivedString == "B") {

wuKong.mecanumRun(wuKong.RunList.rear, 80)

} else if (receivedString == "L") {

wuKong.mecanumRun(wuKong.RunList.left, 80)

} else if (receivedString == "R") {

wuKong.mecanumRun(wuKong.RunList.right, 80)

} else if (receivedString == "S") {

wuKong.mecanumRun(wuKong.RunList.stop, 0)

}

})

input.onButtonPressed(Button.B, function () {

wuKong.mecanumStop()

strip.showColor(neopixel.colors(NeoPixelColors.Black))

})

let strip: neopixel.Strip = null

radio.setGroup(1)

wuKong.mecanumWheel(

wuKong.ServoList.S1,

wuKong.ServoList.S2,

wuKong.ServoList.S3,

wuKong.ServoList.S4

)

wuKong.setLightMode(wuKong.LightMode.BREATH)

strip = neopixel.create(DigitalPin.P16, 4, NeoPixelMode.RGB)

basic.forever(function () {

})

Joystick

radio.setGroup(1)

pins.setPull(DigitalPin.P13, PinPullMode.PullNone)

pins.setPull(DigitalPin.P14, PinPullMode.PullNone)

pins.setPull(DigitalPin.P15, PinPullMode.PullNone)

pins.setPull(DigitalPin.P16, PinPullMode.PullNone)

basic.forever(function () {

if (pins.digitalReadPin(DigitalPin.P15) == 0) {

radio.sendString("Open")

} else if (pins.digitalReadPin(DigitalPin.P13) == 0) {

radio.sendString("Close")

} else if (pins.digitalReadPin(DigitalPin.P16) == 0) {

radio.sendString("LEDL")

} else if (pins.digitalReadPin(DigitalPin.P14) == 0) {

radio.sendString("LEDR")

} else {

if (pins.analogReadPin(AnalogPin.P2) > 550 && (pins.analogReadPin(AnalogPin.P1) > 400 && pins.analogReadPin(AnalogPin.P1) < 600)) {

radio.sendString("F")

} else if (pins.analogReadPin(AnalogPin.P2) < 450 && (pins.analogReadPin(AnalogPin.P1) > 400 && pins.analogReadPin(AnalogPin.P1) < 600)) {

radio.sendString("B")

} else if (pins.analogReadPin(AnalogPin.P1) < 450 && (pins.analogReadPin(AnalogPin.P2) > 400 && pins.analogReadPin(AnalogPin.P2) < 600)) {

radio.sendString("L")

} else if (pins.analogReadPin(AnalogPin.P1) > 550 && (pins.analogReadPin(AnalogPin.P2) > 400 && pins.analogReadPin(AnalogPin.P2) < 600)) {

radio.sendString("R")

} else {

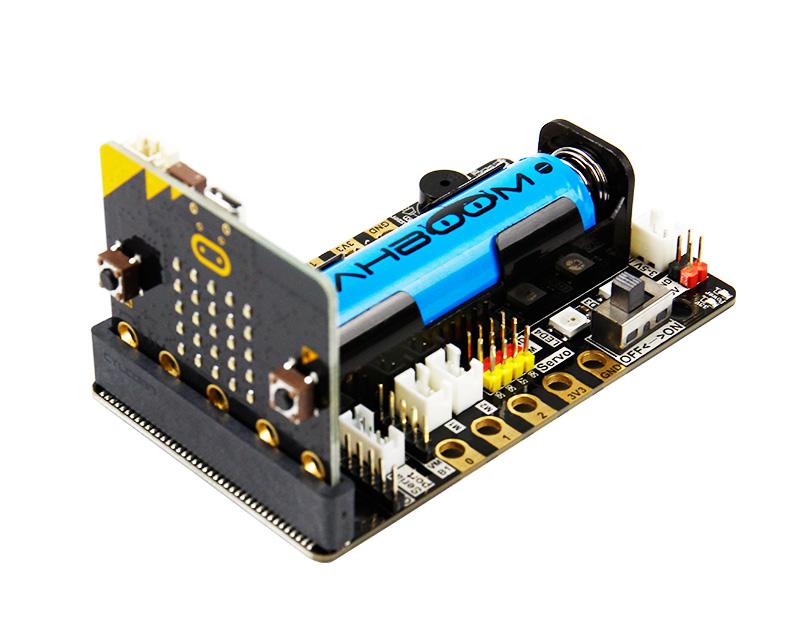
radio.sendString("S")

}

}

})

## Spider Project

### Extension Interface: <https://github.com/lzty634158/SuperBit>

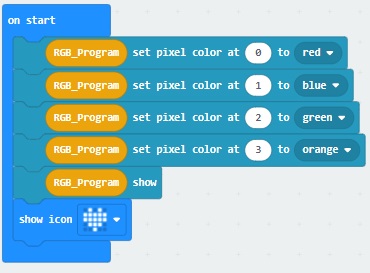
## [video](https://www.youtube.com/watch?v=a6OLxZTac7w)

Class 17 Assembly

Spider assembly instructions are found here: <https://paulware.github.io/BCCRobotics/spiderAssembly.html>

Class 18 Super Bit Lights

# Super Bit LED

This code will set the 4 LEDs to different values:  
  
As an exercise: set different sounds on different button presses

Class 19 Super Bit Music

## Project Biped

  
Class 20: Assembly

Assembly instructions are located here: <https://paulware.github.io/BCCRobotics/bipedAssembly.html>

Class 21: Joystick Control

This javascript will control the joystick:

radio.setGroup(1)

pins.setPull(DigitalPin.P13, PinPullMode.PullNone)

pins.setPull(DigitalPin.P14, PinPullMode.PullNone)

pins.setPull(DigitalPin.P15, PinPullMode.PullNone)

pins.setPull(DigitalPin.P16, PinPullMode.PullNone)

basic.forever(function () {

if (pins.digitalReadPin(DigitalPin.P15) == 0) {

radio.sendString("Open")

} else if (pins.digitalReadPin(DigitalPin.P13) == 0) {

radio.sendString("Close")

} else if (pins.digitalReadPin(DigitalPin.P16) == 0) {

radio.sendString("LEDL")

} else if (pins.digitalReadPin(DigitalPin.P14) == 0) {

radio.sendString("LEDR")

} else {

if (pins.analogReadPin(AnalogPin.P2) > 550 && (pins.analogReadPin(AnalogPin.P1) > 400 && pins.analogReadPin(AnalogPin.P1) < 600)) {

radio.sendString("F")

} else if (pins.analogReadPin(AnalogPin.P2) < 450 && (pins.analogReadPin(AnalogPin.P1) > 400 && pins.analogReadPin(AnalogPin.P1) < 600)) {

radio.sendString("B")

} else if (pins.analogReadPin(AnalogPin.P1) < 450 && (pins.analogReadPin(AnalogPin.P2) > 400 && pins.analogReadPin(AnalogPin.P2) < 600)) {

radio.sendString("L")

} else if (pins.analogReadPin(AnalogPin.P1) > 550 && (pins.analogReadPin(AnalogPin.P2) > 400 && pins.analogReadPin(AnalogPin.P2) < 600)) {

radio.sendString("R")

} else {

radio.sendString("S")

}

}

})

Class 22: Javascript Code for the biped

This code will respond to joystick commands:

input.onButtonPressed(Button.A, function () {

wuKong.mecanumRun(wuKong.RunList.rear, 56)

strip.showRainbow(1, 360)

})

radio.onReceivedString(function (receivedString) {

if (receivedString == "F") {

wuKong.mecanumRun(wuKong.RunList.Front, 80)

} else if (receivedString == "B") {

wuKong.mecanumRun(wuKong.RunList.rear, 80)

} else if (receivedString == "L") {

wuKong.mecanumRun(wuKong.RunList.left, 80)

} else if (receivedString == "R") {

wuKong.mecanumRun(wuKong.RunList.right, 80)

} else if (receivedString == "S") {

wuKong.mecanumRun(wuKong.RunList.stop, 0)

}

})

input.onButtonPressed(Button.B, function () {

wuKong.mecanumStop()

strip.showColor(neopixel.colors(NeoPixelColors.Black))

})

let strip: neopixel.Strip = null

radio.setGroup(1)

wuKong.mecanumWheel(

wuKong.ServoList.S1,

wuKong.ServoList.S2,

wuKong.ServoList.S3,

wuKong.ServoList.S4

)

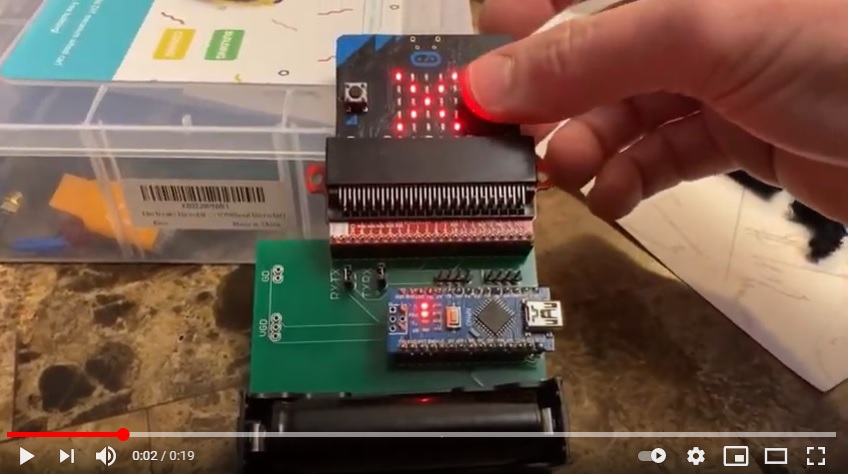
wuKong.setLightMode(wuKong.LightMode.BREATH)

strip = neopixel.create(DigitalPin.P16, 4, NeoPixelMode.RGB)

basic.forever(function () {

})

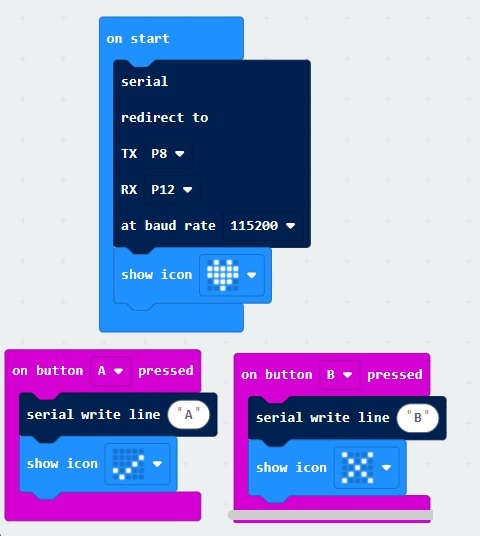
Class 23: Crossover to Arduino



# Its possible to send commands from a micro-bit device to an arduino using a printed circuit board.

To send a command from a micro-bit to an arduino, connect the M8 (tx) pin on the micro-bit to the  
D0 (rx) pin on the arduino nano.

## Micro-bit code

This code will output the character 'A' when the A button is pressed and 'B' when the B button is pressed  


### Javascript microbit code

input.onButtonPressed(Button.A, function () {

serial.writeLine("A")

basic.showIcon(IconNames.Yes)

})

input.onButtonPressed(Button.B, function () {

serial.writeLine("B")

basic.showIcon(IconNames.No)

})

serial.redirect(

SerialPin.P8,

SerialPin.P12,

BaudRate.BaudRate115200

)

basic.showIcon(IconNames.Heart)

basic.forever(function () {

})

Class 24: Arduino Code to respond to characters

This code will respond to a character 'A' by turning off the little led connected to D13.  
When a character 'B' is received it will turn on the little led.

unsigned long ledTimeout = 0;

void setup() {

pinMode (13,OUTPUT);

digitalWrite (13,1);

Serial.begin (115200);

}

void loop() {

char ch;

if (Serial.available()) {

ch = Serial.read();

if (ch == 'A') {

digitalWrite (13,0);

} else if (ch == 'B') {

digitalWrite (13,1);

}

}

}