

## LESSON GUIDE MIX 4

Encourages

- Critical thinking
- STEM interest / skills
- Learning by discovery
- Problem solving



**STEM**  
learning kit using  
**electronic**  
building blocks

Microduino  
brings our class  
projects to life.



**140+**  
PIECES

MicroDuino Learning  
Kit includes everything  
needed for a fun  
learning experience  
in STEM.

**idea** lab  
**MICRODUINO**

Need more inspiration? Find other  
inventions at [microduinoinc.com](http://microduinoinc.com)

 **MICRODUINO**  
[MICRODUINOINC.COM](http://MICRODUINOINC.COM)

## Table Of Contents

Scratch Introduction	3-25
Hardware Introduction	26-46
Rotating Motor Project	47-68
Motion Sensor Motor Project	69-88
Emoji Robot Project	89-108
Single Axis Steadicam Project	109-127
Airbag Project	128-147
Wired RC car Project	148-167
Safety Car Project	168-187
Delivery Truck Project	188-208
Obstacle Course Project	209-229
Edge Avoidance Car Project	230-250
Line Finding Car 1 Project	251-271
Line Finding Car 2 Project	272-294



Your students can use Scratch to code their own interactive stories, animations, and games. In the process, they learn to think creatively, reason systematically, and work collaboratively — essential skills for everyone in today's society.

With Scratch, you can program your own interactive stories, games, and animations — and share your creations with others in the online community.

Scratch helps young people learn to think creatively, reason systematically, and work collaboratively — essential skills for life in the 21st century.

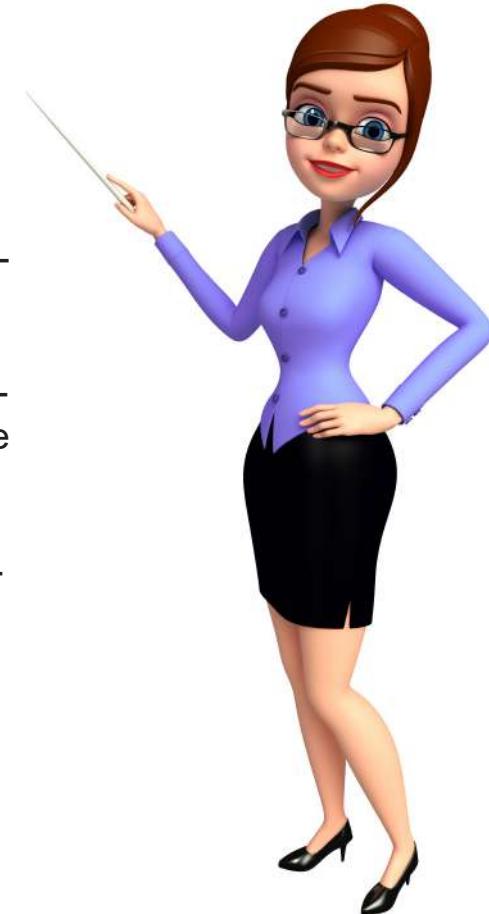
# Introduction

The Microduino MIX guide will walk you through an introduction in the use of Arduino's Scratch editor and then a number of projects. With each project the challenge level increases so as to expand your knowledge of coding and code thinking.

The process of coding also involves how to organize in your thinking the process and sequence of events that will result in the correct operation of your project.

In our projects we use the Arduino 'Scratch' editor to construct and execute code to drive the hardware of projects. Scratch has become one of, if not the, most popular coding tool today.

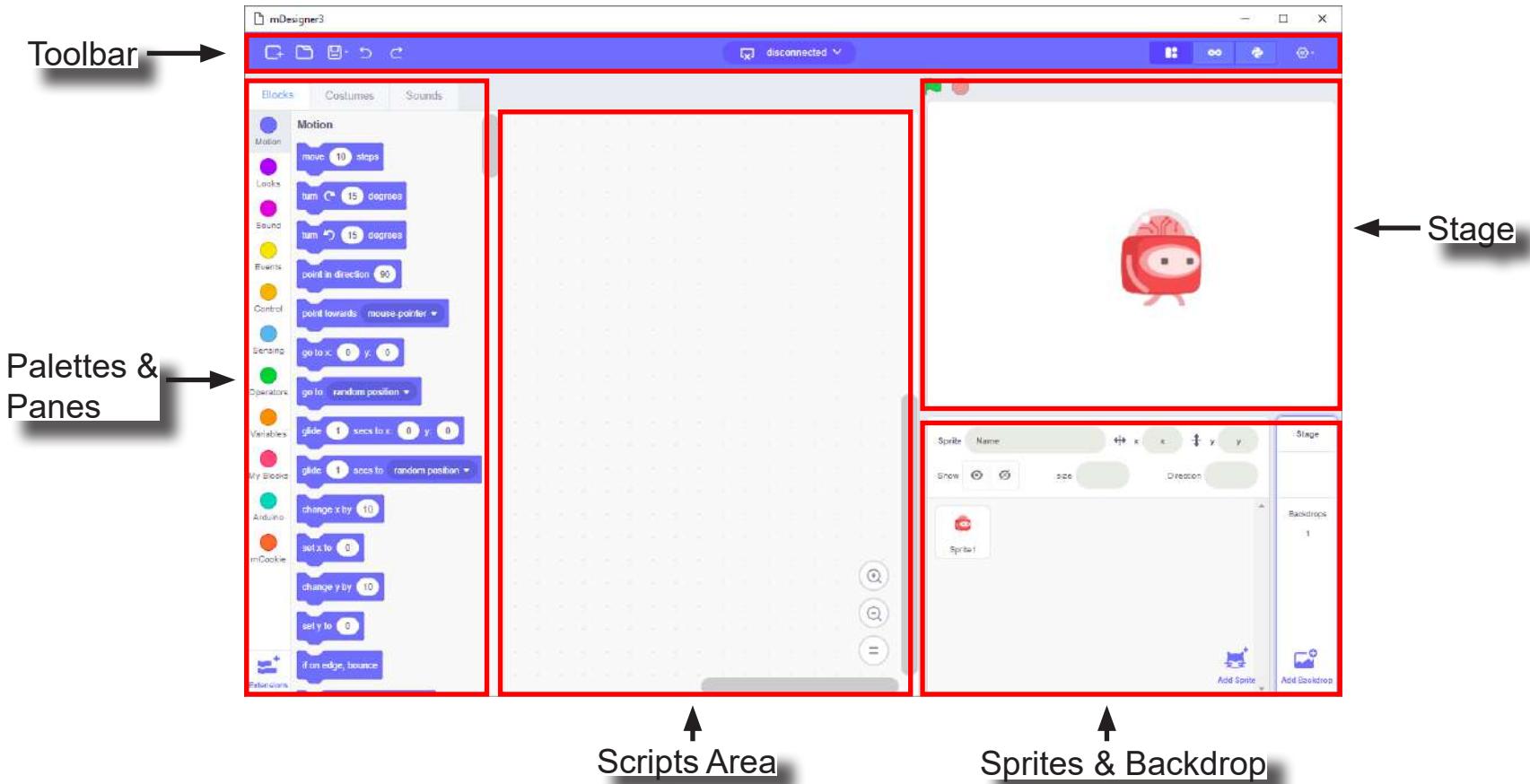
As you read the introduction of Scratch keep in mind that the basic areas of the scratch editor uses many similar terms used in the parts of a stage play. It refers to stages, scripts, costumes, backdrops and so on. This is meant to help us create a mental image of what we doing. Beyond this we get into the more technological terms of technology.



# Installation

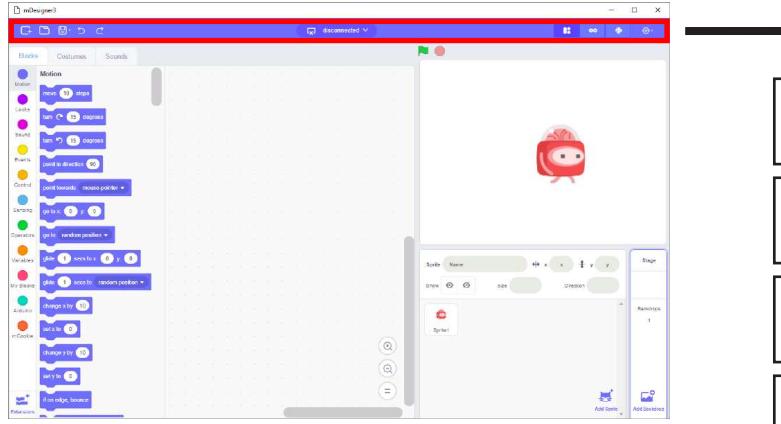


# The Interface



# Toolbar

The **Tool Bar** in Scratch provides many useful functions and configuration options. Below is a quick overview of each menu item. Some terms and phrases will be explained later.



A screenshot of the Scratch application interface. On the left is the script editor with categories like Motion, Sound, and Control. In the center is the stage with a cat sprite. On the right is the settings menu, which includes sections for Kit, Flash firmware, Language, Check update, Issue report, and About. The toolbar at the top has several icons: a plus sign for New Project, a folder for Open Project, a save icon for Save / Save As, an arrow pointing left for Undo, an arrow pointing right for Redo, a disconnected serial port icon, and three mode selection icons (Scratch, Arduino, Text Coding).

**New Project** - creates a new blank project.

**Open Project** - open a saved project.

**Save / Save As** - save / save as the current project.

**Undo** - undo the most recent change in the project.

**Redo** - redo the most recent undo in the project.

**Serial Port** - select serial port for usage or programming of MIX hardware.

**Mode Selection** - select different control modes for Scratch:

- **Scratch / Live mode** - control MIX hardware in real time using Scratch blocks. MIX hardware requires a constant connection to a PC in order to function.
- **Arduino Mode** - compiles and uploads codes onto MIX hardware. Once a program is uploaded, MIX hardware can run independently without a PC connection.
- **Text Coding Mode** - text based programming. Currently under development.

# Stage

Scratch can be thought as a school play. The Stage is where the viewer sees the play. Sprites can be thought as actors or characters in a play. Backdrop is the background. Together they make the Stage. Below is a quick overview. Some terms and phrases will be explained later.

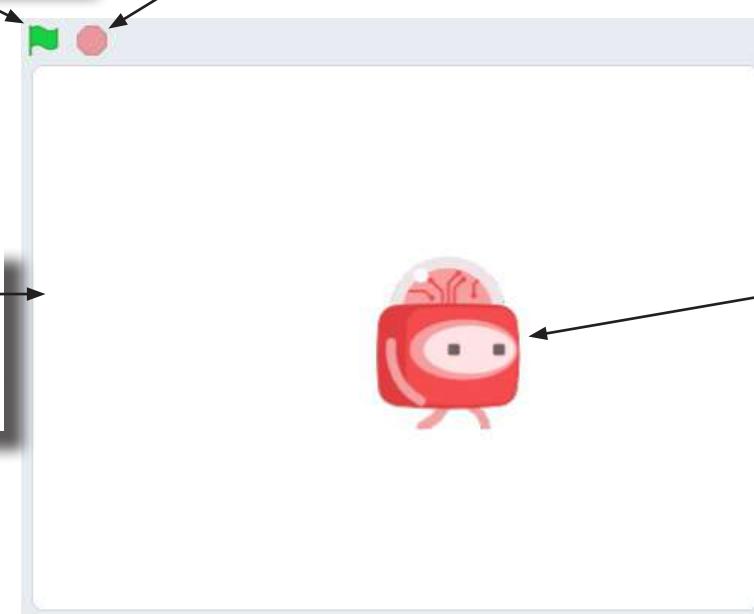
## Green Flag:

When clicked, all “When Green Flag Click” blocks activates.



## Stop Sign:

When clicked, stops all Scripts that are running.



## Backdrop:

Is the background of the stage.  
There can only be one Backdrop.  
It can be change and you can also design your own.

## Sprite:

Sprites can be thought as actors or characters. There can be multiple Sprites on the Stage. You can also design your own.

The Stage includes **Sprite(s)** and a **Backdrop**.

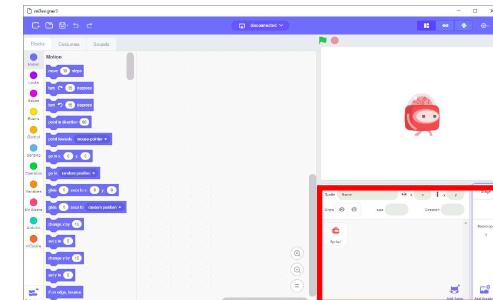
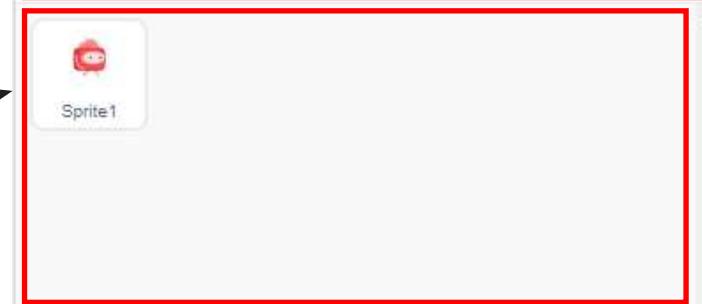
# Sprites & Backdrop

This section allows you to select, edit and create Sprites and Backdrops. Selecting a Sprite or Backdrop is done by clicking on it from this section. Below is a quick overview. Some terms and phrases will be explained later.

When a sprite is selected. Edit the sprite name, sprite position on the stage, visibility, size of the sprite and the angle of the sprite.



Sprites in your project are located here. There can be multiple sprites in a project. New projects automatically have one Sprite included. Its name is Corey.



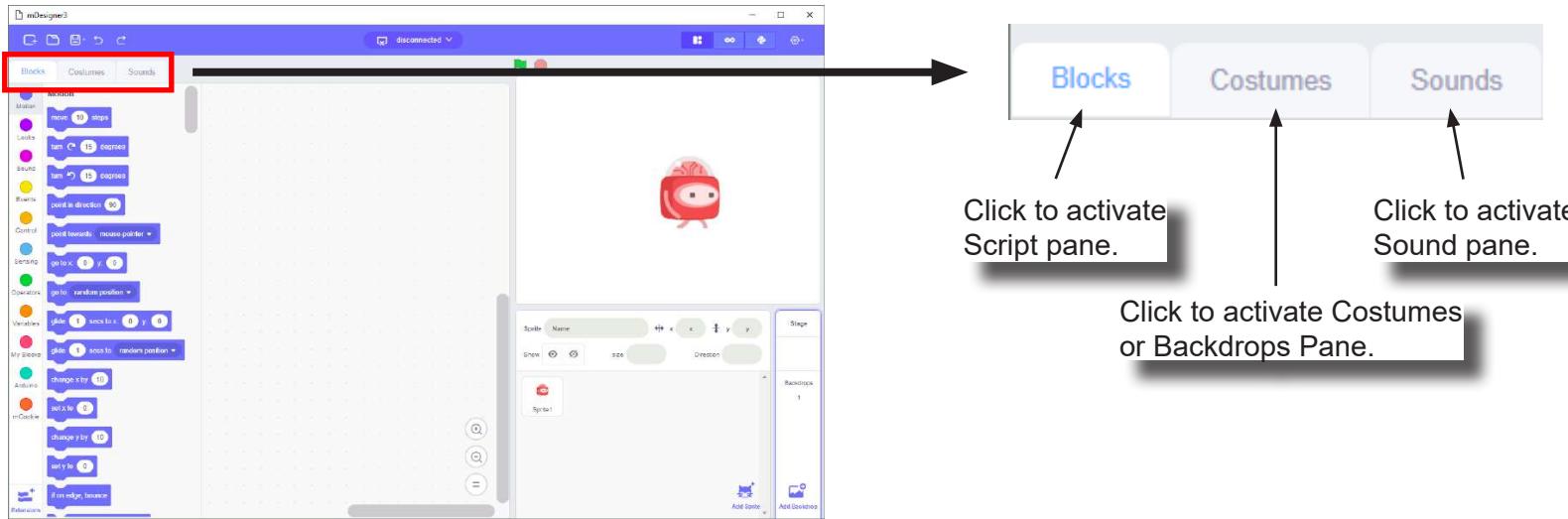
Current backdrop.

Add a new backdrop from the backdrop library.

Add a new sprite from the sprite library.

# Palettes & Panes

This section of the interface contains the Sprite Pane, Costume Pane / Backdrop Pane, and Sound Pane. You can switch between the panes by clicking on the appropriate tabs.

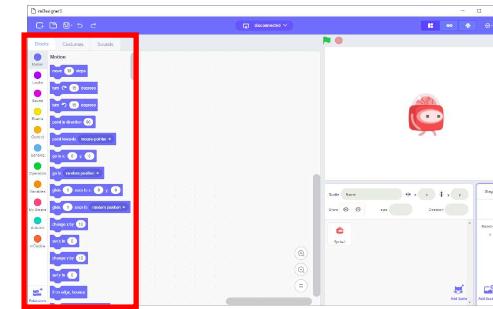
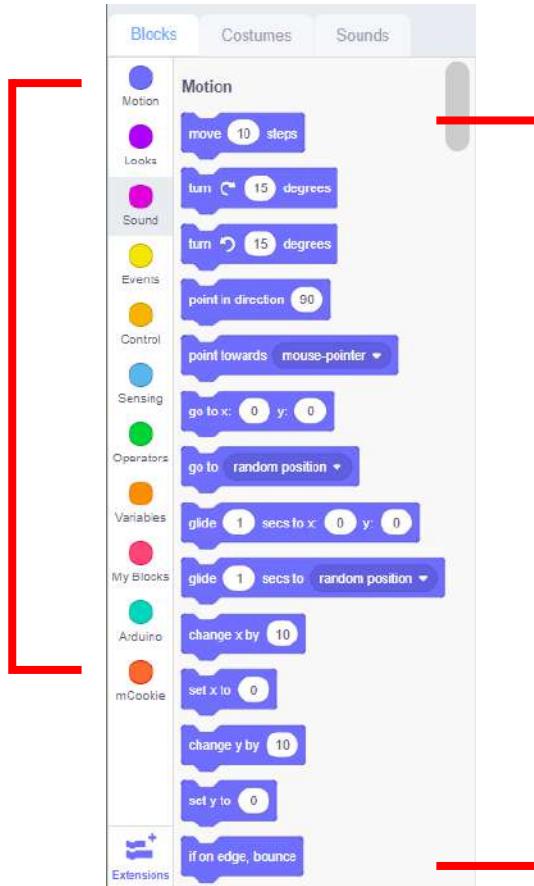


# Script Pane

The Script Pane contains the tools you need to make your Sprite or Backdrop do things. A Script can be thought of as the instructions that your Sprite or Backdrop follows, similar to a school plays' script which instructs the actor what to do. Some terms and phrases will be explained later.

## Block Categories:

Blocks are sorted to different categories for quicker access. Each category has its own color and the Blocks contained within are the same color. For example, "Motion" Blocks are all purple.

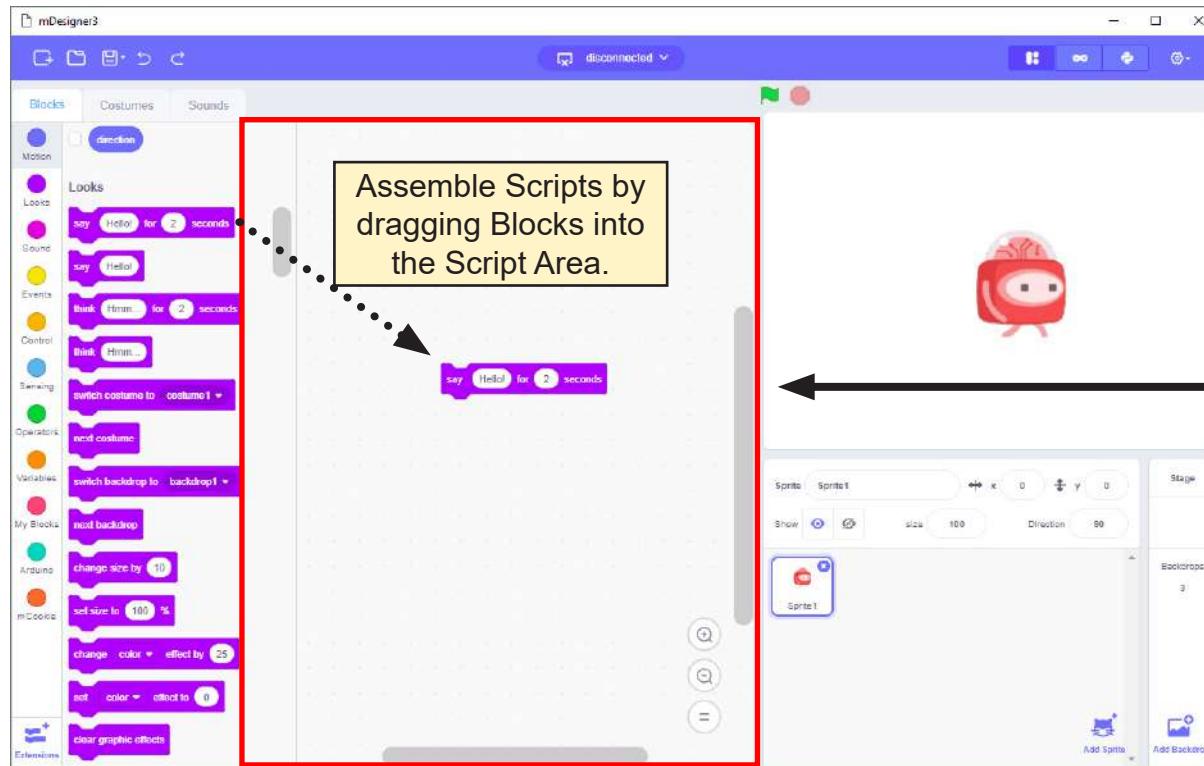


## Blocks:

Blocks are instructions for your Sprite or Backdrop to follow. There are many Blocks each with their own type. Blocks are categorized into different categories.

## Scripts Area

The Scripts Area is one of the most important parts of the Scratch interface. It allows you to create and assemble the Script for your currently selected Sprite or Backdrop. You can drag Blocks from the Scripts Pane into the Scripts Area to assemble your Script. Each Sprite or Backdrop has its own personal Scripts Area. A Script can be thought of as the instructions that your Sprite or Backdrop follows.

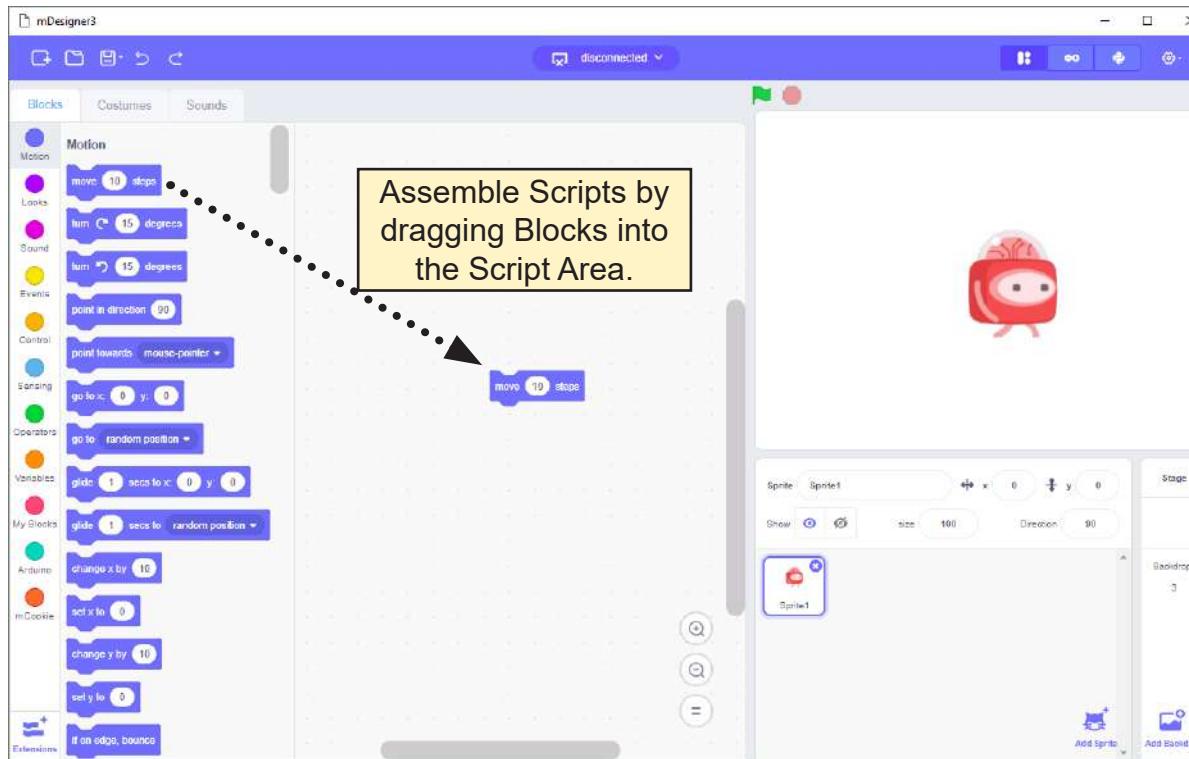


**Scripts Area:**  
Assemble your Script for your  
currently selected Sprite or  
Backdrop.

# Exercise: “First Steps”

Let's create your very first project! Let make Corey take its very first steps.

Assemble the Script below by dragging the Blocks you need from the Sprite Pane into the Scripts Area. The “Block Locator” sidebar is there to aid you in finding the blocks you need you need quickly. Use it for your convenience!



## Block Locator

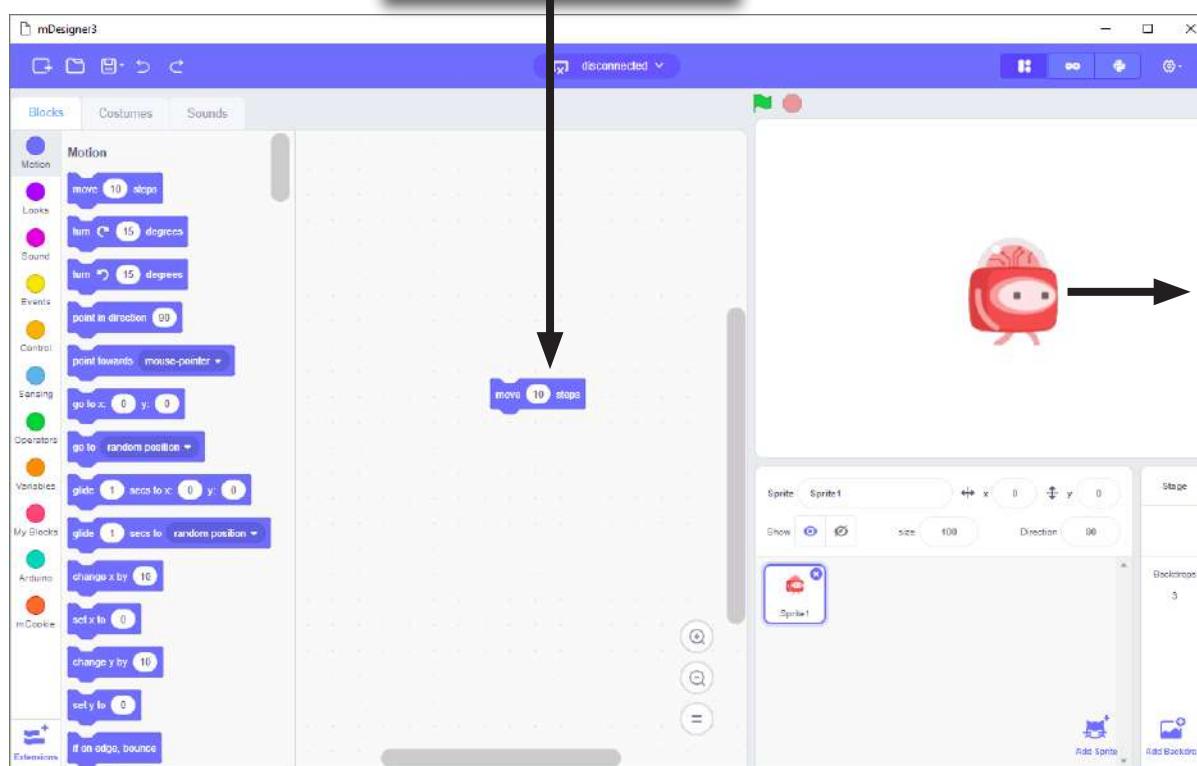
Motion

move 10 steps

## Exercise: “First Steps”

Now click on the Block you just added to the Scripts Area. What happens to Corey on the stage? It moved 10 steps to the right!

Click on any colored part  
of the block to activate it.



# Types of Blocks

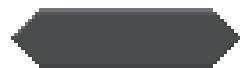
There are different types of Blocks which have different shapes. Each shape has a different purpose and is intended to aid in assembling Scripts.



Hat blocks are the blocks that start every script. They are shaped with a rounded top and a bump at the bottom — this is so you can only place blocks below them.



Stack blocks are the blocks that perform the main commands. They are shaped with a notch at the top and a bump on the bottom — this is so blocks can be placed above and below them.



Boolean blocks are the conditions — they are either true or false. It's like asking your friend: "Does  $2 + 2 = 4$ ?", and they would either tell you "Yes" or "No". Boolean blocks have a hexagonal shape.



Reporter blocks are the values. Shaped with rounded edges, Reporter blocks can hold numbers and strings. It is like asking a friend, for example, "What is  $2 + 2$ ?", and they would answer "4". It is not just equations however, it can report a variable, for example, "What is your age?". They may answer: "15".



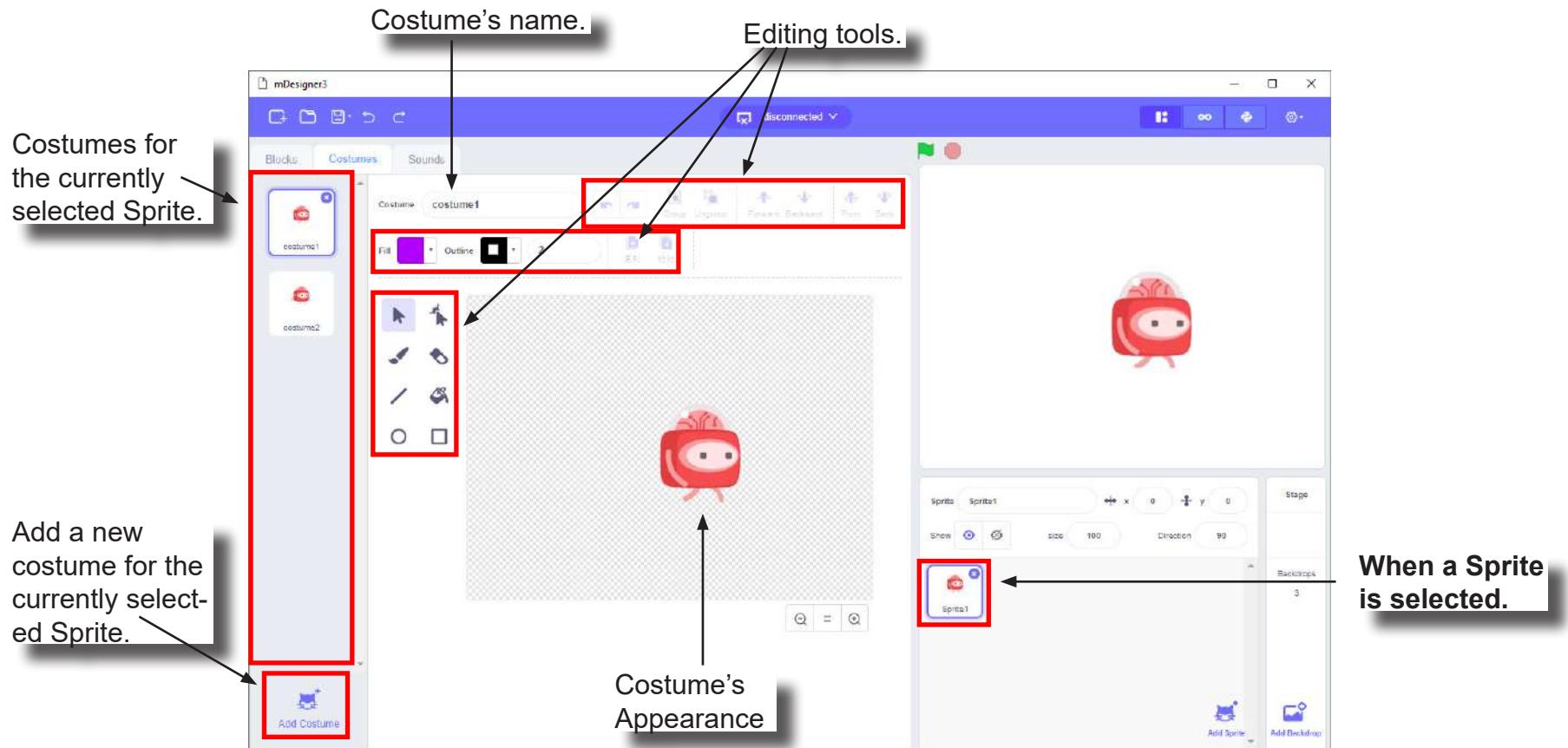
C blocks are blocks that take the shape of "C's". Also known as "Wrap blocks", these blocks loop the blocks within the Cs or check if a condition is true.



Cap blocks are the blocks that end scripts. They are shaped with a notch at the top and a flat bottom — this is so you cannot place any blocks below them.

## Costume Pane

When a Sprite is currently selected, the middle tab becomes the Costume Pane. This section allows you to design Costumes for your currently selected Sprite. Costumes can be thought of as a different image / appearance for your Sprite. It is useful to use Costumes when creating animations for your Sprite or if you want your Sprite to change its appearance.



## Exercise: “Second Steps”

How can we animate Corey to look like it is walking?  
We change Corey’s costume to look like it is taking steps.

Assemble the Script below:



### Block Locator

Motion

move (10) steps

Looks

switch costume to [costume1]

Control

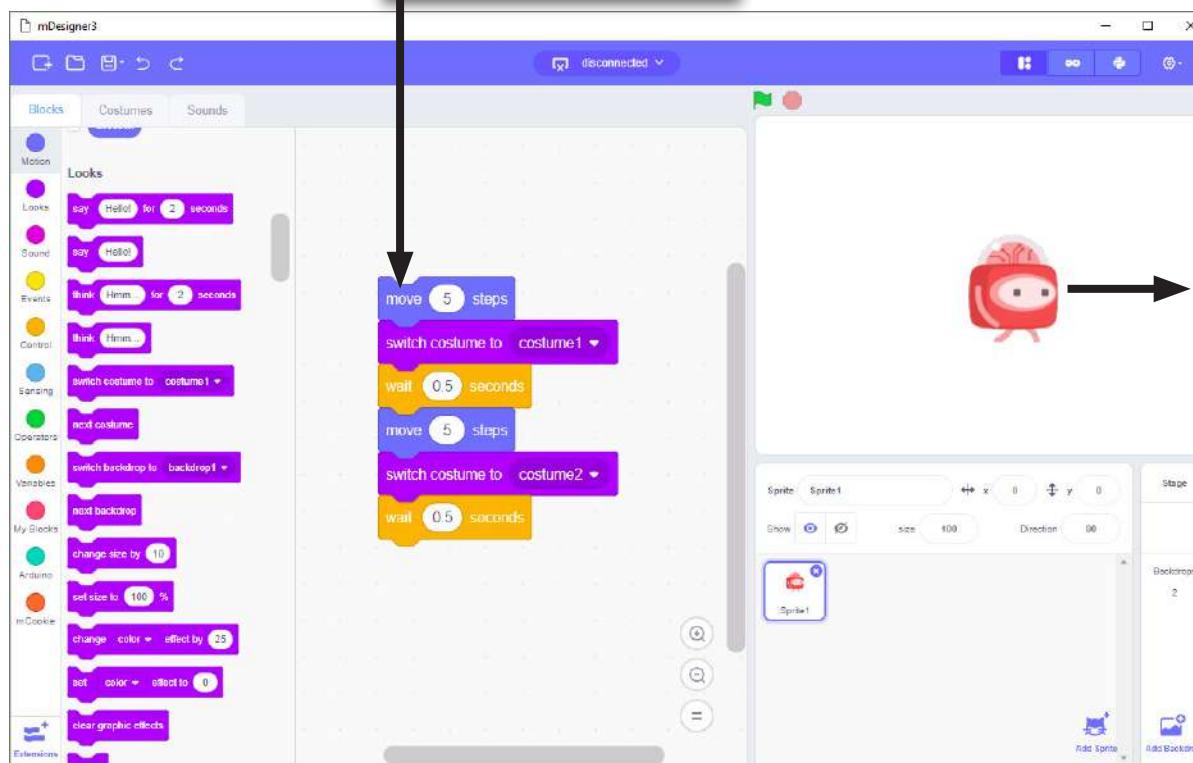
wait (1) secs

## Exercise: “Second Steps”

Click on any color portion of the Script. What does Corey do?

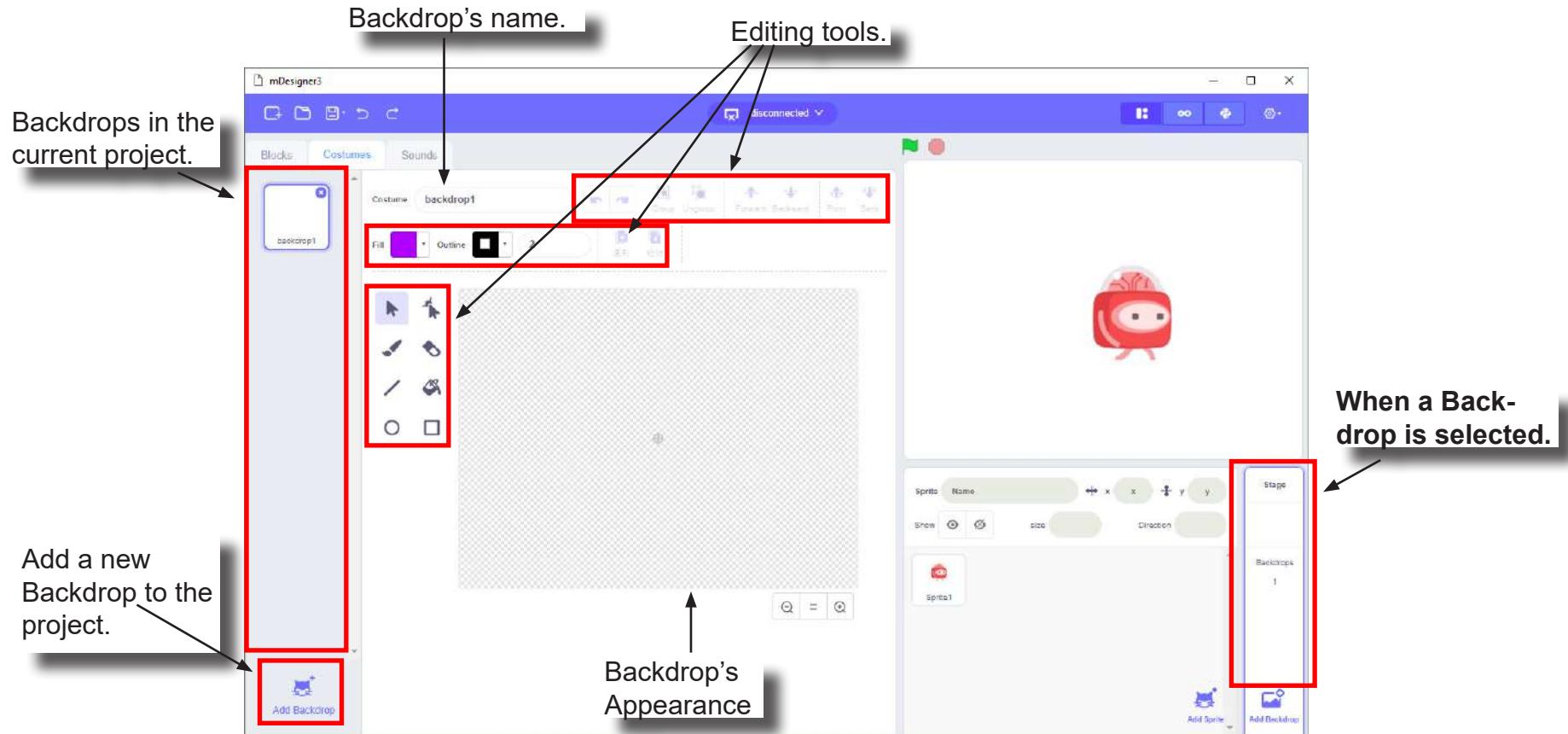
Corey walks with an animation! Click repeatedly to make Corey perform the animation over and over again.

Click on any colored part  
of the block to activate it.



## Backdrop Pane

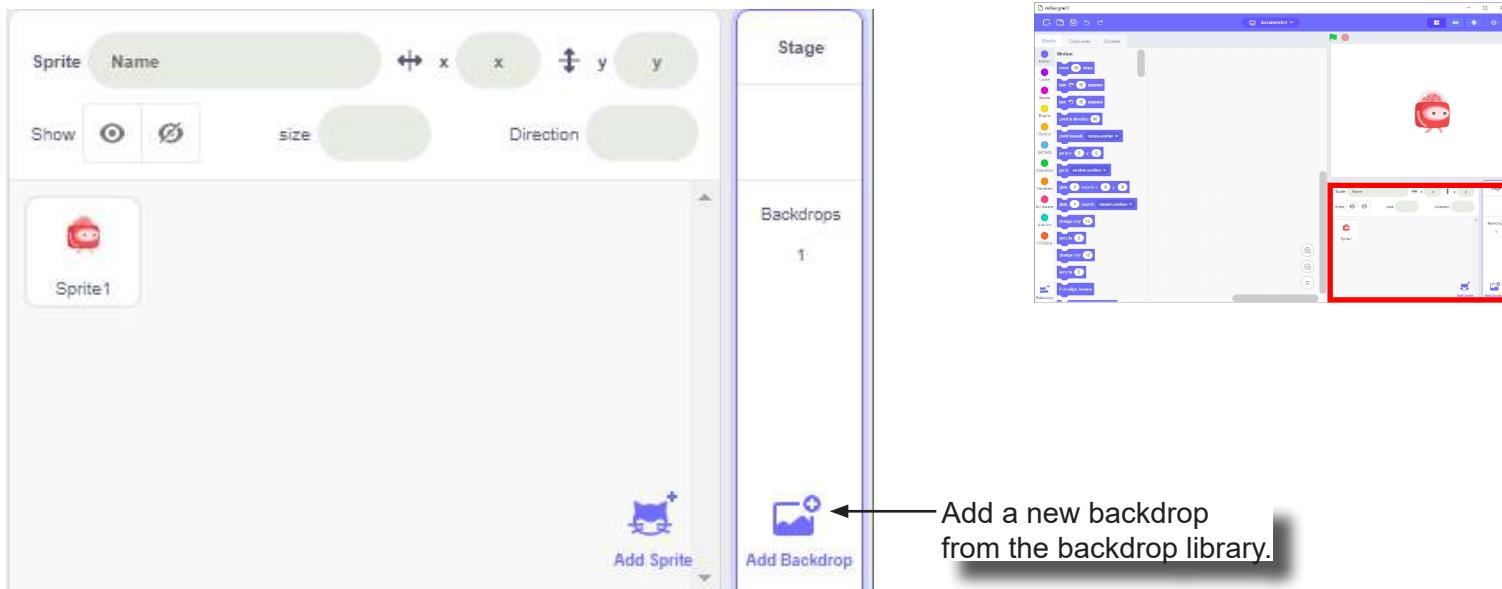
When Backdrop is currently selected, the middle tab becomes the Backdrops Pane. This section allows you to design Backdrops. It is useful to have multiple Backdrops if you want to change the background for your project.



# Creating a Backdrop

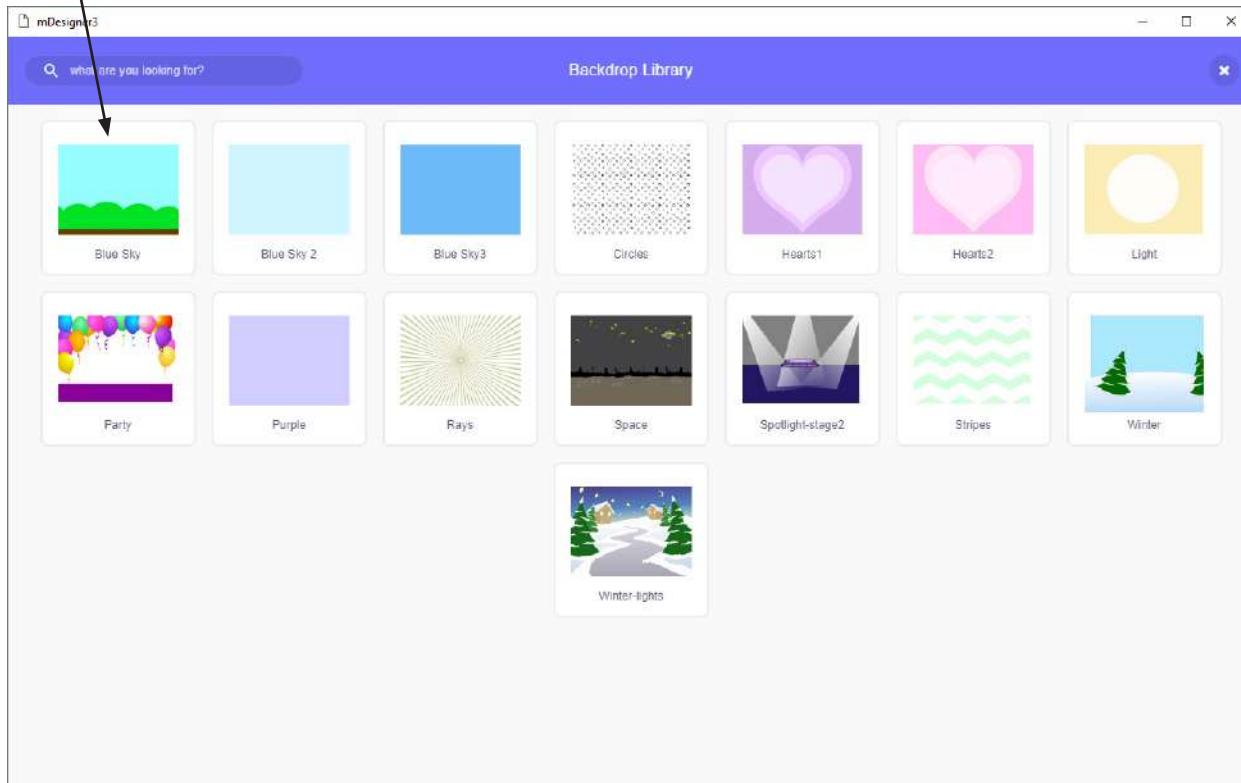
How can we make Corey walk in a park?

We can change the Backdrop to a scenery of a park.  
But first we need to create a Backdrop of a park.

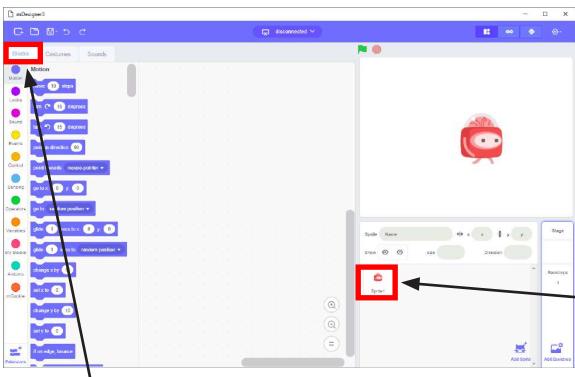


# Creating a Backdrop

Select a backdrop for the park.



## Exercise: “A Walk in the Park”



2. Click to switch to Switch Pane.

Don't forget to switch back to Corey by selecting Corey and then switching to the Scripts Pane.

Once you have created your new Backdrop, assemble the blocks below:

1. Click to select Corey.

```
forever
  move (5) steps
  switch costume to [costume1 v]
  wait (0.5) seconds
  move (5) steps
  switch costume to [costume2 v]
  wait (0.5) seconds
switch backdrop to [backdrop1 v]
switch backdrop to [Blue Sky v]
```

A Scratch script consisting of a 'forever' loop. Inside the loop, there are two 'move (5) steps' blocks, each preceded by a 'switch costume to [costume1 v]' block and followed by a 'wait (0.5) seconds' block. After the loop, there are two 'switch backdrop to' blocks: one for 'backdrop1' and one for 'Blue Sky'. A black arrow points from the text 'Set to your newly Created Backdrop.' to the 'Blue Sky' backdrop switch block.

## Block Locator

### Motion

move (10) steps

### Looks

switch costume to [costume1 v]

switch backdrop to [backdrop1 v]

### Control

wait (1) secs

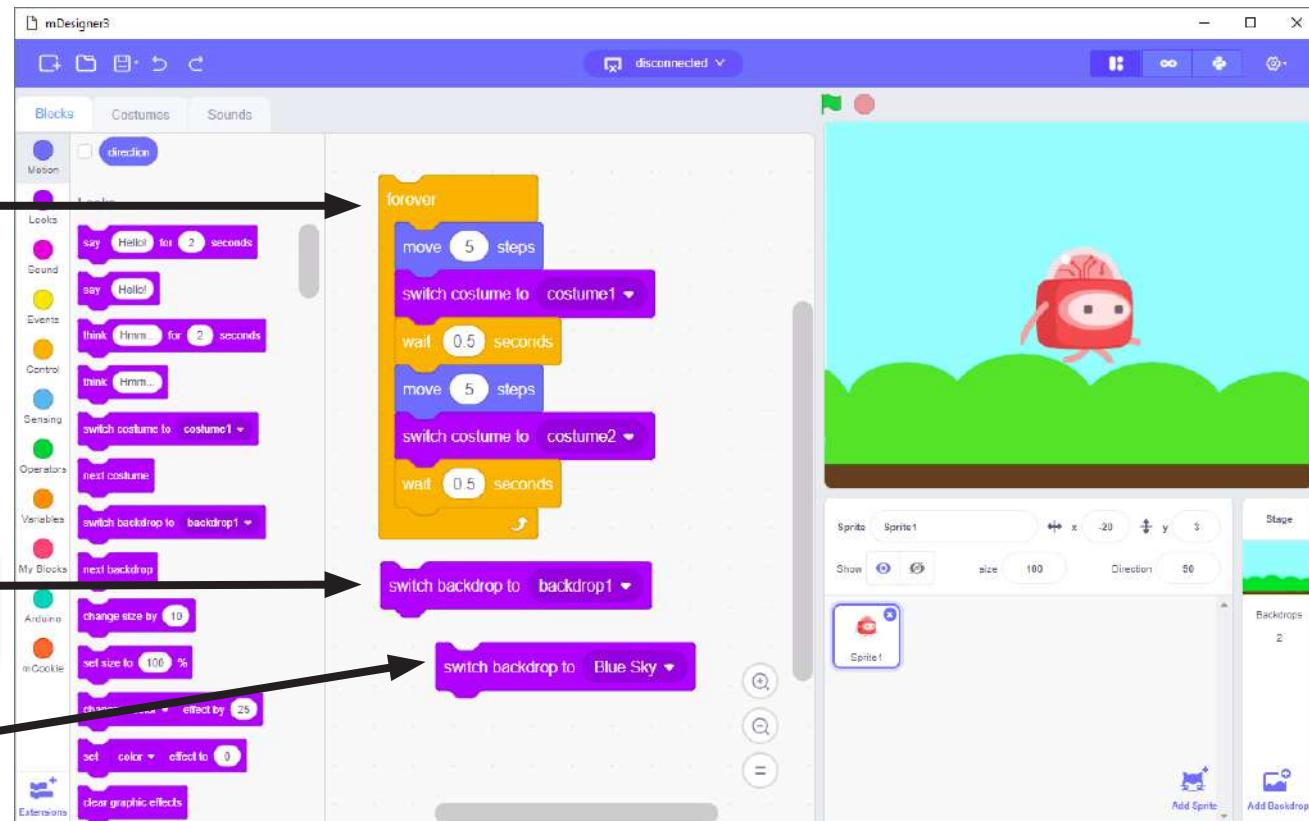
forever

## Exercise: “A Walk in the Park”

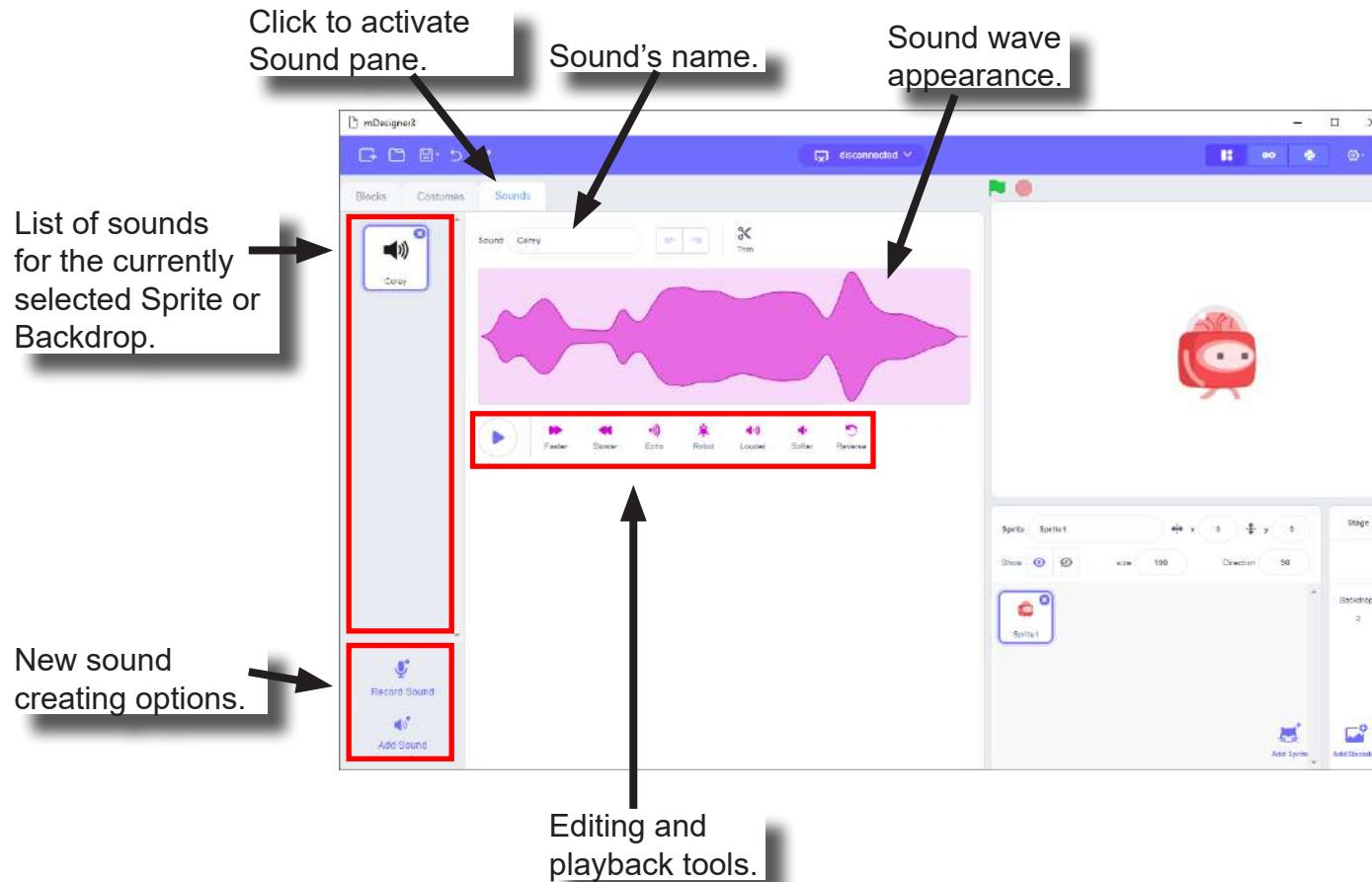
Click and Corey will start walking forever.

Click to change the Backdrop to the default white one.

Click to change the Backdrop to your custom one.



## Sound Pane



# Further Learning

This chapter is intended to give a basic understanding of how to use Scratch when creating projects in the MIX kits. There are a vast amount of topics to learn about Scratch which are outside the scope of this product. Many resources can be found online to learn more about Scratch!

## Links

- <https://scratch.mit.edu/>



# Hardware Mix 4

Microduino Educational Kits  
inspire STEM/STEAM

- Critical thinking
- STEM concepts
- Learning through discovery
- Technology instruction
- Problem solving
- Introduce coding concepts using hardware for easy visualization



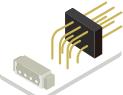
# MIX 4 Kit Contents

1x  mCenter+

1x  OLED

1x  Motor Controller

2x  IO Splitter

1x  Servo Connector

1x  Remote Control

2x  Wheel

1x  USB Cable

## Sensors

1x  IR Receiver

1x  Joystick

4x  Line Finder

1x  Motion

2x  Potentiometer

1x  Touch Button

24x  Hub Connector Cables

## Trinkets

1x  Buzzer

1x  Color LED

2x  Motor

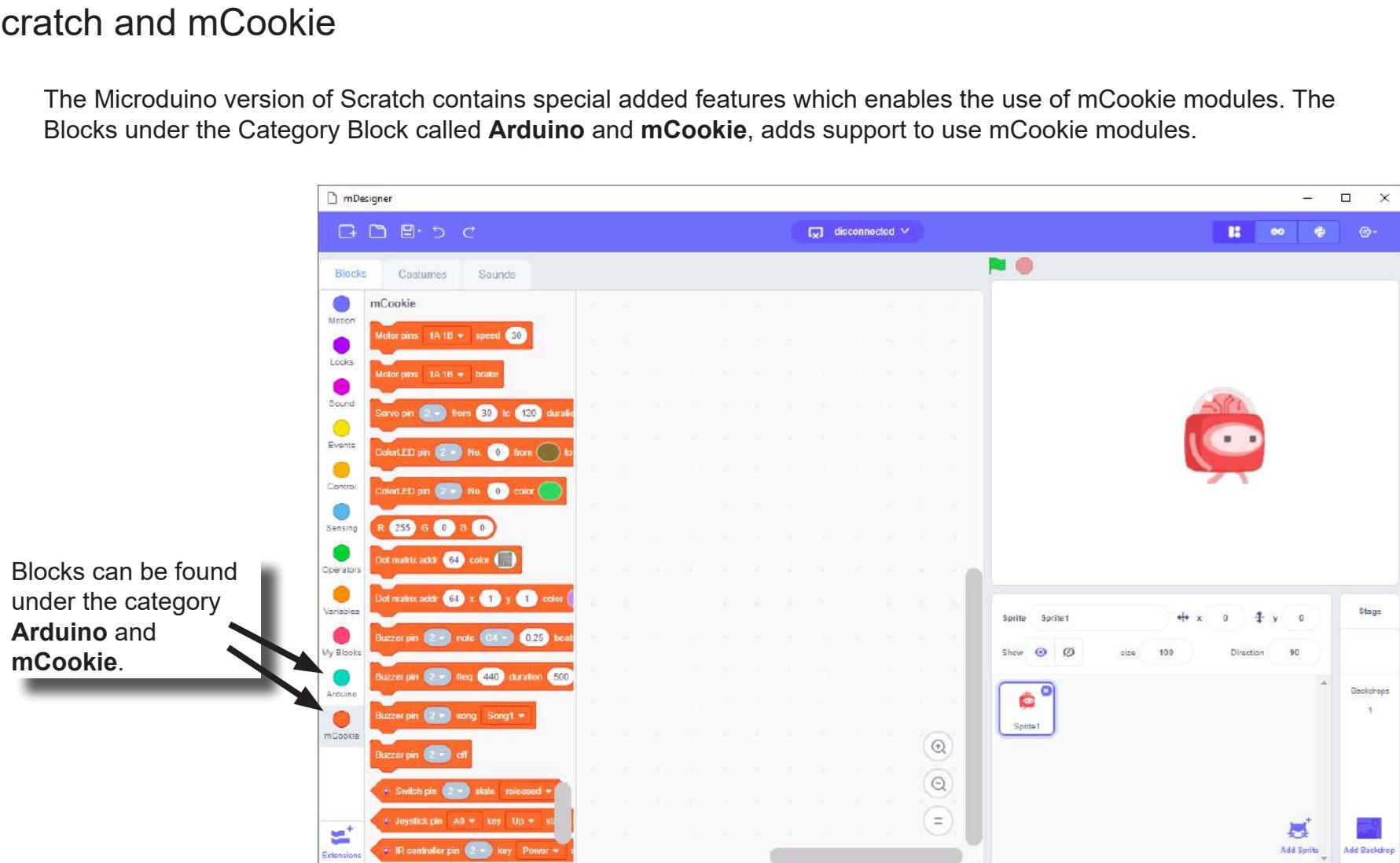
2x  Servo



Building Blocks

# Scratch and mCookie

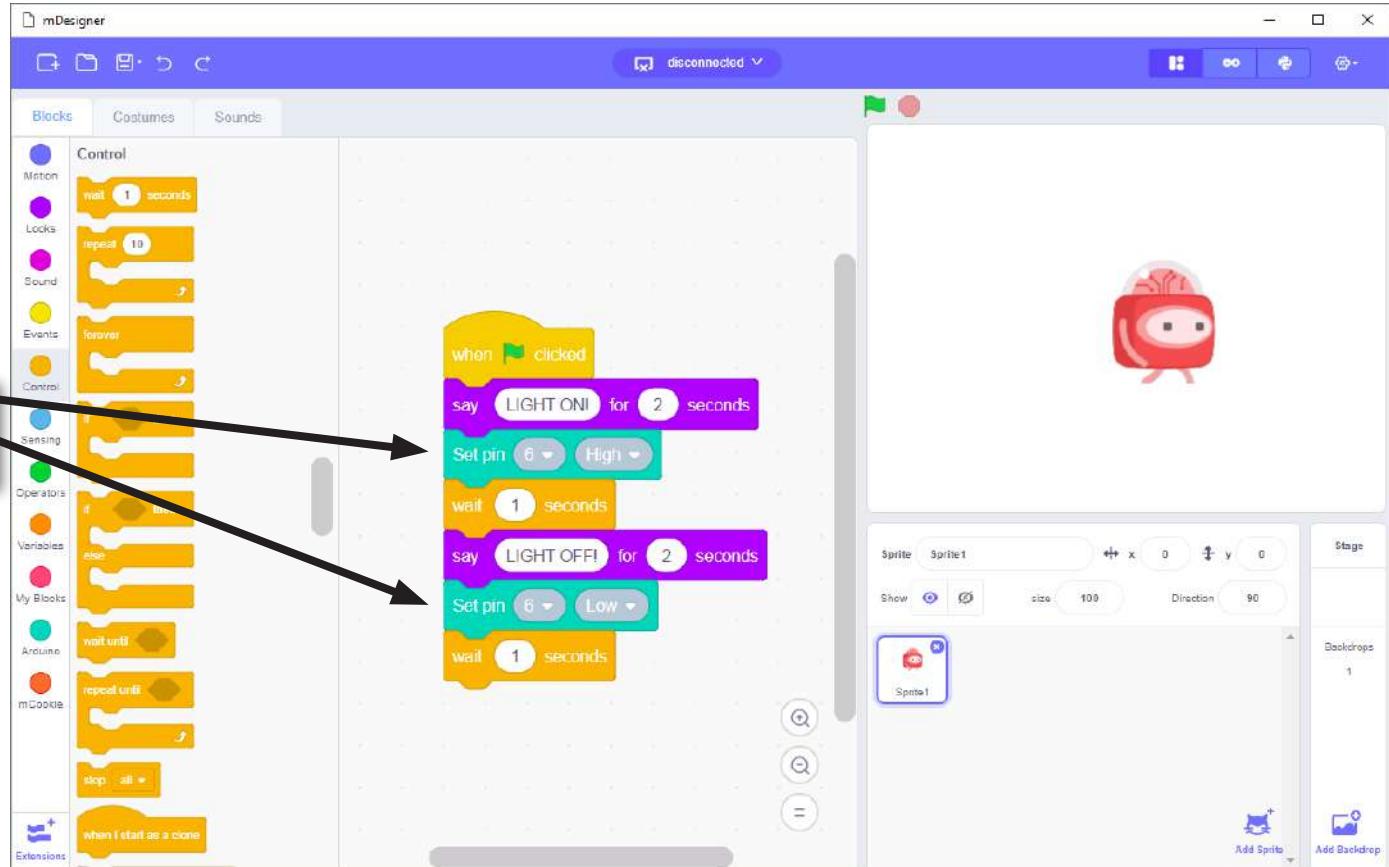
The Microduino version of Scratch contains special added features which enables the use of mCookie modules. The Blocks under the Category Block called **Arduino** and **mCookie**, adds support to use mCookie modules.



# Scratch and mCookie

You can use the Blocks under the **Arduino** and **mCookie** category in your Scripts as you would use any other Block.

**Arduino and  
mCookie Blocks.**

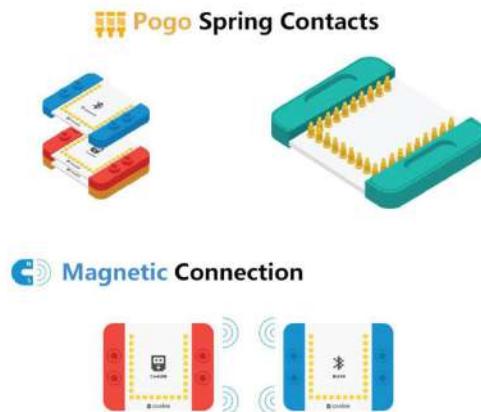


## mCookie Series

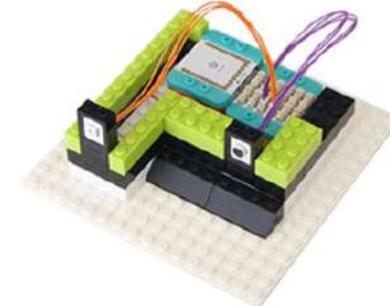
Microduino's mCookie series is designed for ease of use. The series is modular, stackable, magnetized, easily connectable and building blocks compatible. It makes assembling a project quick and easy.



Modular



Stackable and Magnetized

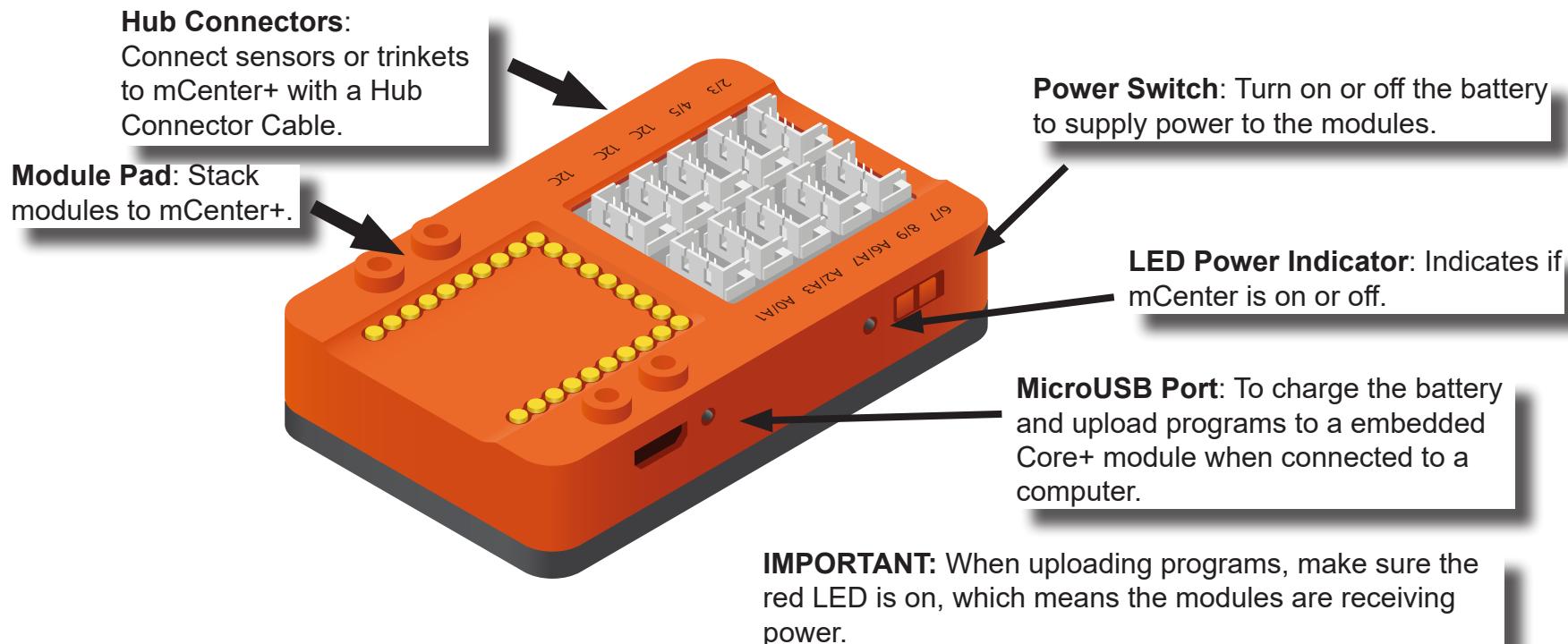


Building Block Compatible

# mCenter+

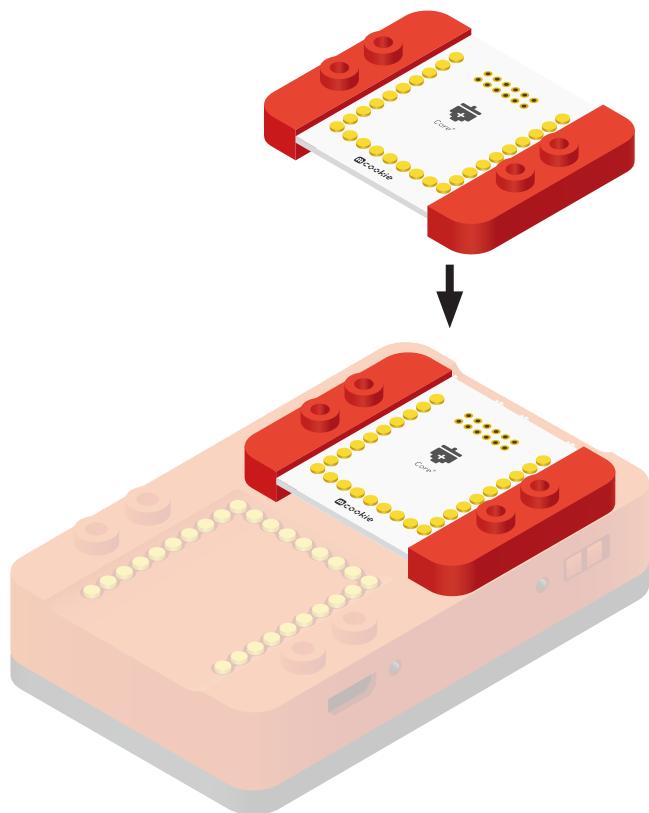
Is a five-in-one module which contains:

- **Li-ion Battery**: battery pack to power your projects which can be recharged via the USB port.
- **mCookie Core+**: a tiny central processing unit (CPU), which is the brains of your projects.
- **USB Programmer**: programs the Core+ module and acts as a serial communication bridge between your computer and Core+ module.
- **Bluetooth Upload**: similar to the USB programmer, but works wirelessly over Bluetooth communication.
- **Hub**: connects various modules such as sensors, trinkets, and other modules to your projects. These modules can then be accessed by the Core+ module.



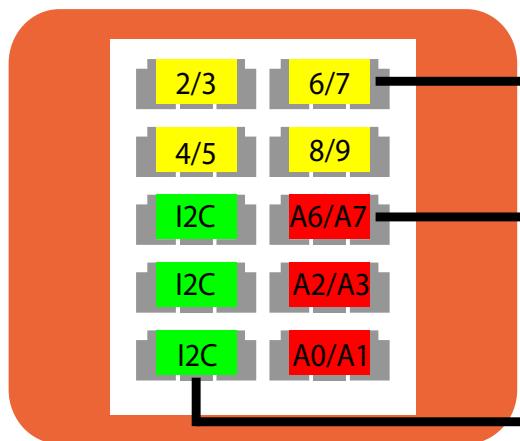
## Embedded Core+ Module

mCenter+ contains an embedded Core+ module. The Core+ module is the brains of your projects. It is a tiny central processing unit (CPU) that you can program or control. It is capable of controlling or sensing information from other modules. It is programmable when connected to a computer.

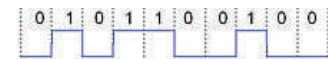


## mCenter+'s Hub

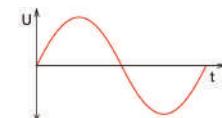
mCenter+ incorporates a Hub. The Hub connects various modules such as sensors, trinkets, and other modules to a project. These modules can then be accessed by the embedded Core+ module. The Hub has 10 connection header ports. Each connection header has support for different sets of features. Some headers may support a feature, while another may not.



Digital signals produce only a off (LOW) or on (HIGH) state. There is no “middle” state between the two.



Analog signals are continuous and feature many “middle” points. They look like smooth curves.



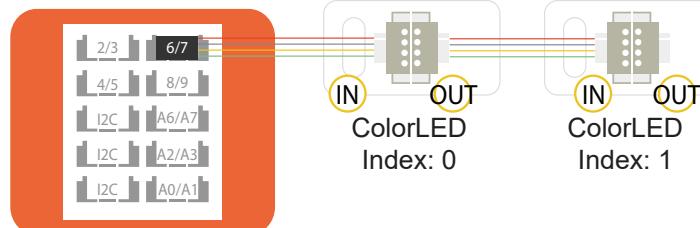
IIC is a type of serial communication interface. It is more advanced than digital and analog signals. Under IIC, messages are exchanged and understood between communicating modules. Modules such as the OLED screen, temperature & humidity sensor, real time clock (RTC) module and other modules use this method of communication.

# Color LED



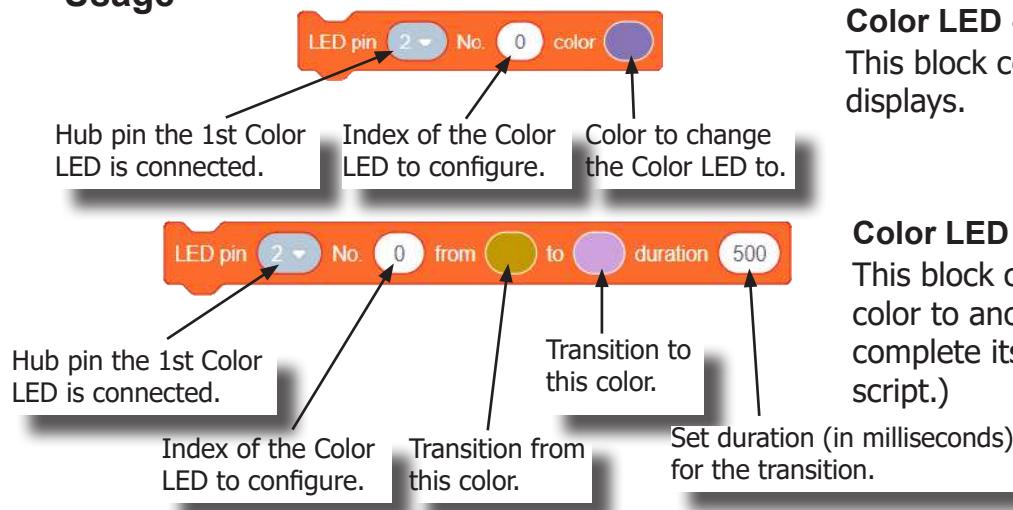
The Color LED is a LED composed of 3 smaller LEDs which emit red, green, and blue. Each color's brightness value can be configured individually. This can produce a wide variety of colors. It is controlled over a special communication protocol which is different from the basic "Digital Output" (used by Single-Color LED).

## Indexing

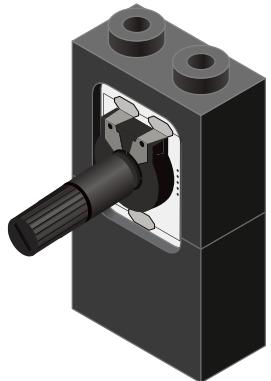


- The Color LED connected directly to the Hub is index 0.
- The Color LED connected next in the chain is index 1.
- And so on.

## Usage



# Potentiometer



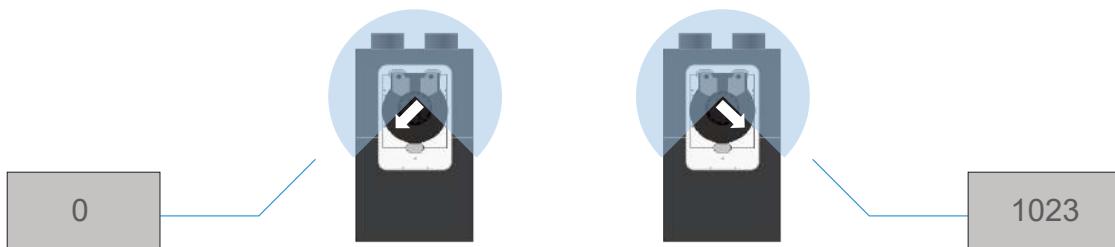
A Potentiometer is a input module which can be turned or rotated right or left. Based on the position of the knob, it produces a different signal value for the Core module to read. It must be connected to an analog port (ports denoted with "A").

## Analog Read block

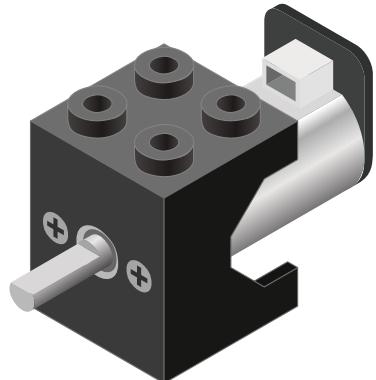
Analog read pin A0 ▾

Pin the Potentiometer  
is connected to.

This block returns the value from the Potentiometer. A value between 0 and 1023 is returned. If the potentiometer is turned all the way to the left, it produces a 0 value. If it is turned all the way to the right, it produces a 1023 value. In the middle it produces a value around 512.



# Motor



**Motor - Set Speed block**



**Motor - Brake block**



A Motor is a electrical machine which converts electrical energy to mechanical energy. It is capable of rotating clockwise and counter clockwise. It is controlled by a Motor Controller. A Motor Controller can control up to 2 Motors. A Wheel can be attached to the Motor.



**Motor Controller**

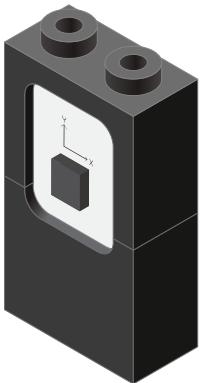


**Wheel**

This block sets the speed of the motor. Positive values turns the Motor in one direction. While negative values turns the Motor in the opposite direction. A value of 0 stops the Motor. Values between -255 and 255 are accepted. 255 is maximum clockwise speed. -255 is maximum counter-clockwise speed.

This block sets the speed to 0 to stop the Motor.

# Motion



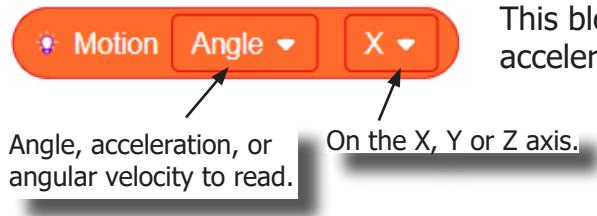
The Motion module is a triple axis accelerometer and gyroscope.

It is capable of measuring on the X, Y & Z axis the:

- Angle
- Acceleration
- Angular velocity

It uses the "I2C" communication protocol and must be connected to an "I2C" port on mCenter+.

## Motion - Read block

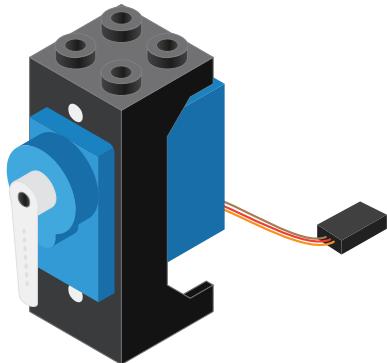


Angle, acceleration, or  
angular velocity to read.

On the X, Y or Z axis.

This block reads data from the Motion module. It is capable of retrieving the angle, acceleration and angular velocity on the X, Y or Z axis.

# Servo



A Servo is an actuator which consist of a motor and other electronic components which allows for precise angle adjustments of the attached horn. The servo included in the kit has an adjustable angle range between 0 and 180 degrees.

## Servo - Set Angle block



Pin the Servo is connected to.

Angle to set Servo to.

Sets the angle of the Servo. Quickly moves (as fast as possible) to the desired angle.

## Servo - Move From-To Angle block



Pin the Servo is connected to.

Starting angle of Servo.

Target angle for Servo.

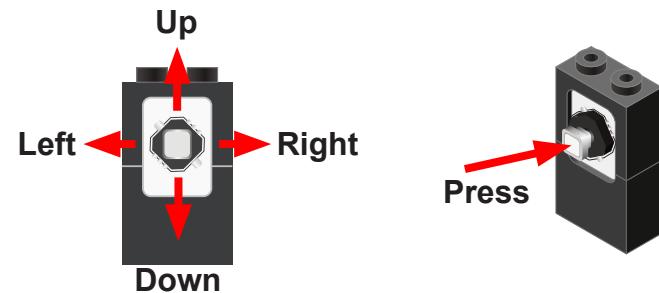
Sets the angle from the starting angle to the target angle. This movement is performed in the specified duration time.

# Joystick



A Joystick is an input module (sensor) which can sense if the stick is in the **up**, **down**, **left** or **right** positions.

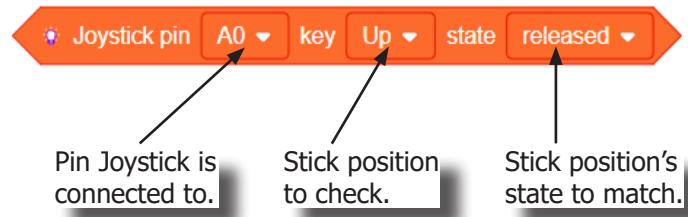
In addition, it can sense if it being **pressed** (pressing directly on stick).



## Usage

The Joystick must be connected to an "Analog Read" compatible pin. These are the pins prefixed with an "A" (pins A0 to A7).

## Joystick State block



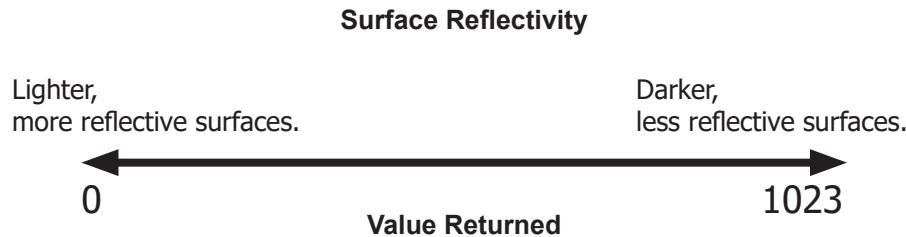
On the defined Joystick pin, if the stick's position (**up**, **down**, **left**, **right** or **center**) matches the state (released, pressed or pressing), then this block returns TRUE.

Otherwise, this block returns FALSE.

# Line Finder



The Line Finder module is able to detect reflectivity or gray level of a surface. In general, darker or less reflective surfaces produce a higher value. While a lighter or more reflective surface produces a lower value. **It must be connected to a pin with an "A" prefix (A0/A1, A2/A3, A6/A7).**



## Analog Read block

Analog read pin A0 ▾

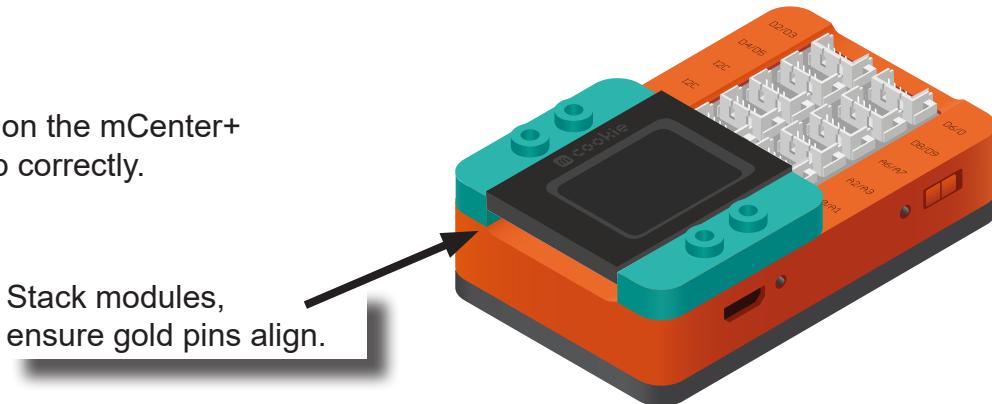
This block returns the value that the Line Finder detects. A value between 0 and 1023 is returned.

Pin that Line Finder  
is connected to.

# Hardware Connectivity

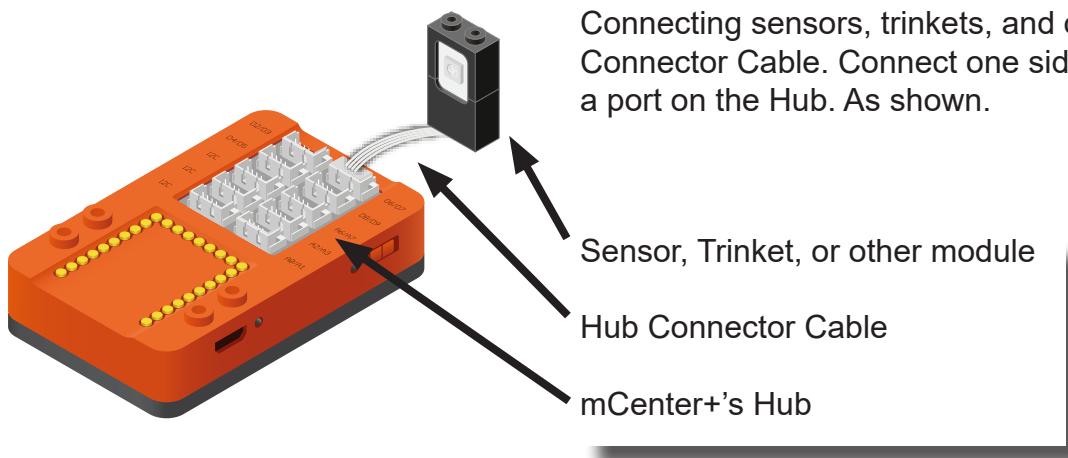
## Stacking Modules

Stacking module is done simply by stacking them on the mCenter+ or upon themselves as shown. Ensure pins line up correctly.



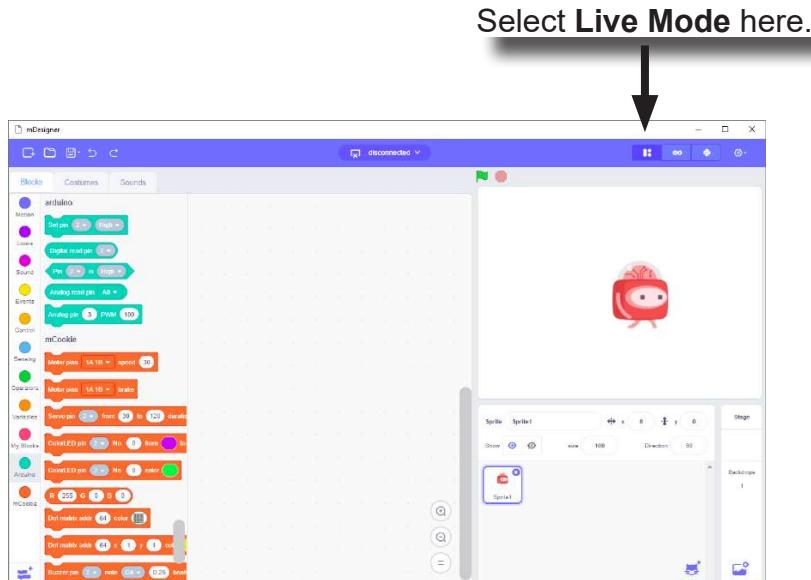
## Connecting Modules to the Hub

Connecting sensors, trinkets, and other modules to the Hub is done with the Hub Connector Cable. Connect one side of the cable to the module and the other to a port on the Hub. As shown.

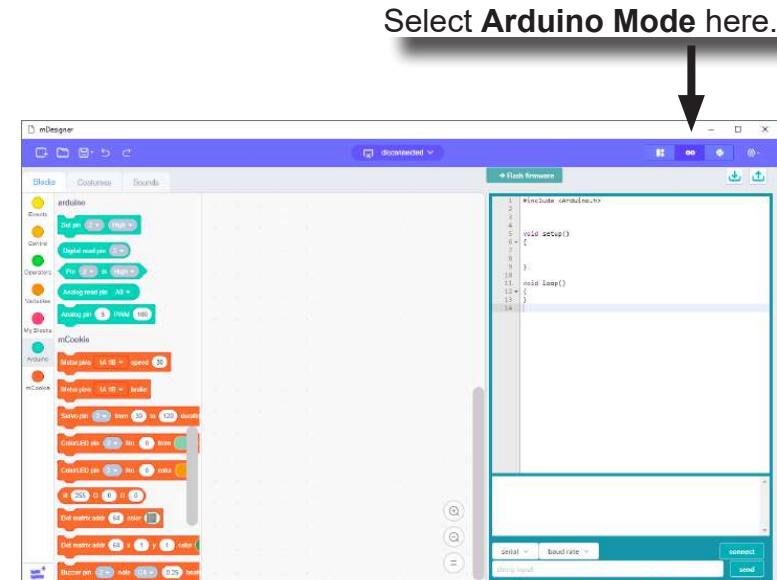


# Setup for Scratch

Using the mCookie modules with Scratch can be accomplished in two ways: **Live mode** and **Arduino mode**. The projects in this MIX Kit will use Scratch in Live mode. Arduino mode is an advanced mode and will not be covered in this Kit.



Live Mode

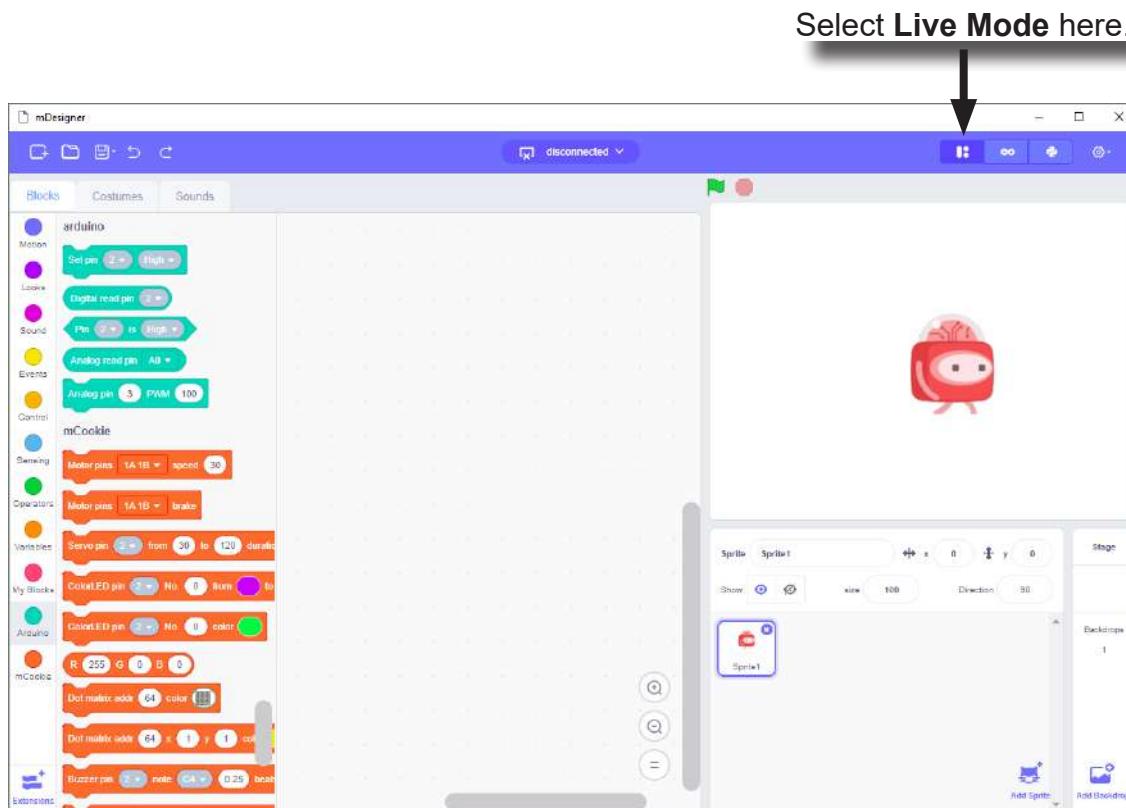


Arduino Mode

## Live Mode

Live Mode is the default mode when using Scratch. It allows you to use Scratch as normal, but with the ability to use modules included in the MIX Kit. You must upload a special firmware onto the Core module to use this mode.

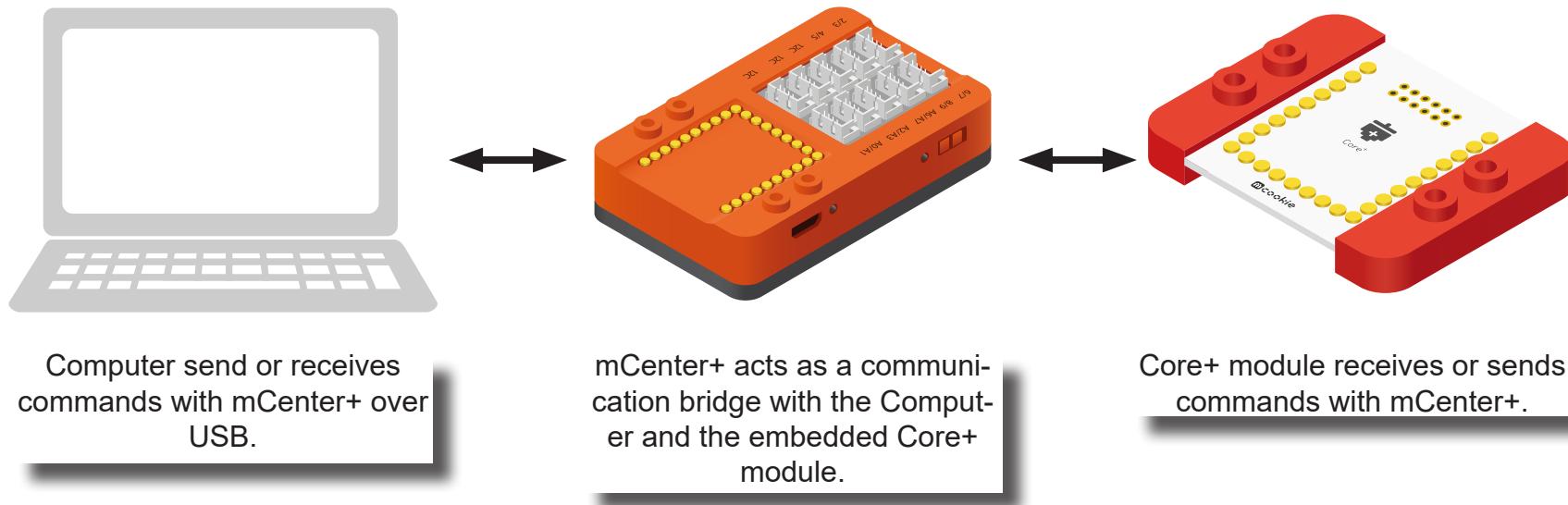
In this mode, to control the modules, you will need to have your project connected to your computer sending commands in real time. If you disconnected the project from your computer, your project will no longer be receiving commands.



## Live Mode

A **firmware** is a program that runs on the Core module. Live Mode requires a special firmware to be upload onto the Core module. This special firmware allows the Core module to perform commands sent from your computer in real time.

Uploading this special firmware only needs to be done once, unless it is overwritten with another firmware (such as when using Arduino mode).



# Live Mode

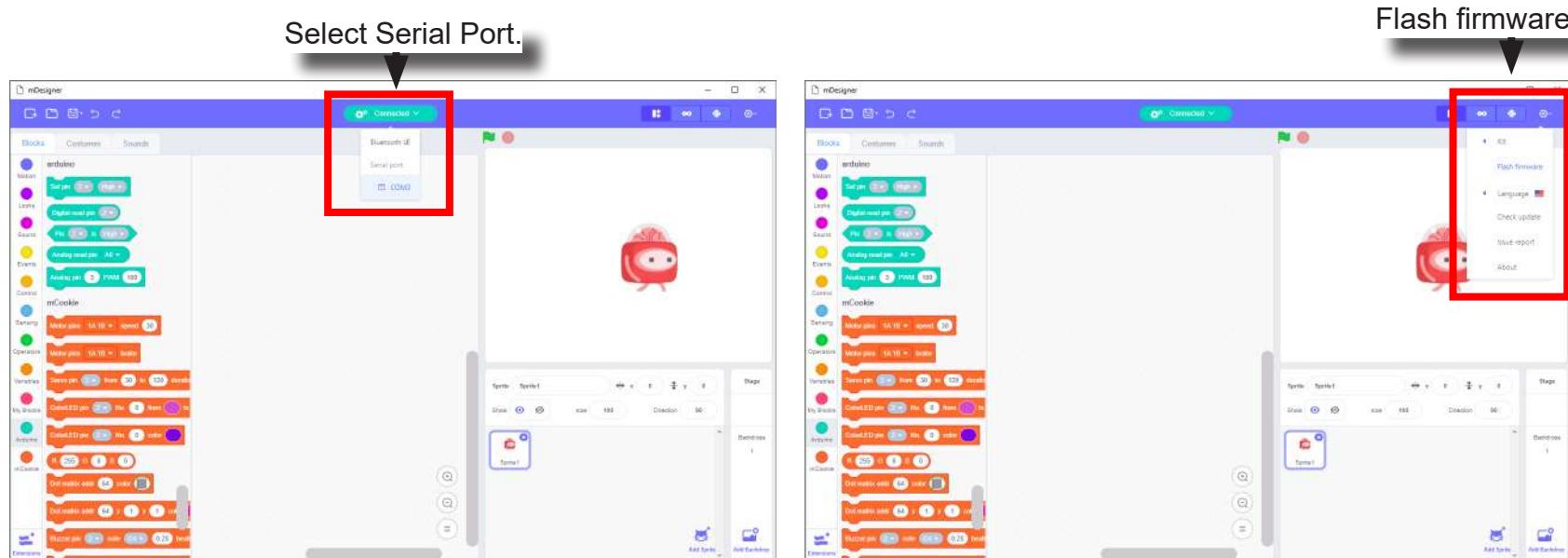
## Uploading the Special Firmware

### In Hardware

- Ensure drivers are install.
- Connect mCenter+ to your computer using a MicroUSB cable.
- A LED near the power switch should be on. If not, turn on mCenter+ with the power switch.

### In Scratch:

- Select the port mCenter+ is connected to.
- Select it again to disconnect. It should show “disconnected” before attempting to upload the firmware.
- Under **Settings** select **Flash firmware** and select **mCookie** to upload the special firmware. Uploading will start.
- Select the port mCenter+ is connected to again to begin using Scratch with MIX hardware.

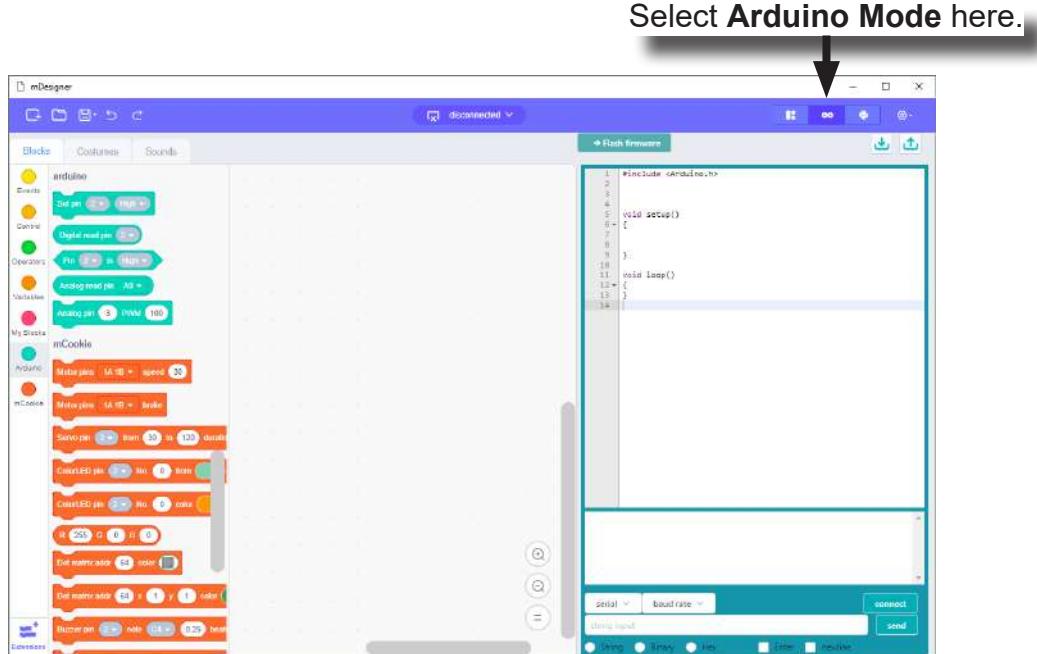


## Arduino Mode

Arduino Mode is a special mode which can be enabled. It acts as a drag and drop editor for programming. The Stage is disabled and many Block Categories cannot be used.

The main advantage of using Arduino Mode is that code is generated, compiled, and uploaded. This means that you can disconnect your project from your computer and it can still work. Whereas, during Live Mode, you need to have your project connected to your computer to instruct your project what to do.

Arduino Mode can be accessed by clicking on the **Arduino Mode** button.



mCenter+ runs its own program without needing to be connected to a Computer.



#### WHAT YOU WILL NEED



MCENTER +



COLOR LED



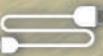
MOTOR DRIVER



N20 REDUCTION  
MOTOR



POTENTIOMETER



USB CABLE



SENSOR CABLE  
x2

## MIX4A01 Rotating Motor

Control a motor using a  
potentiometer.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



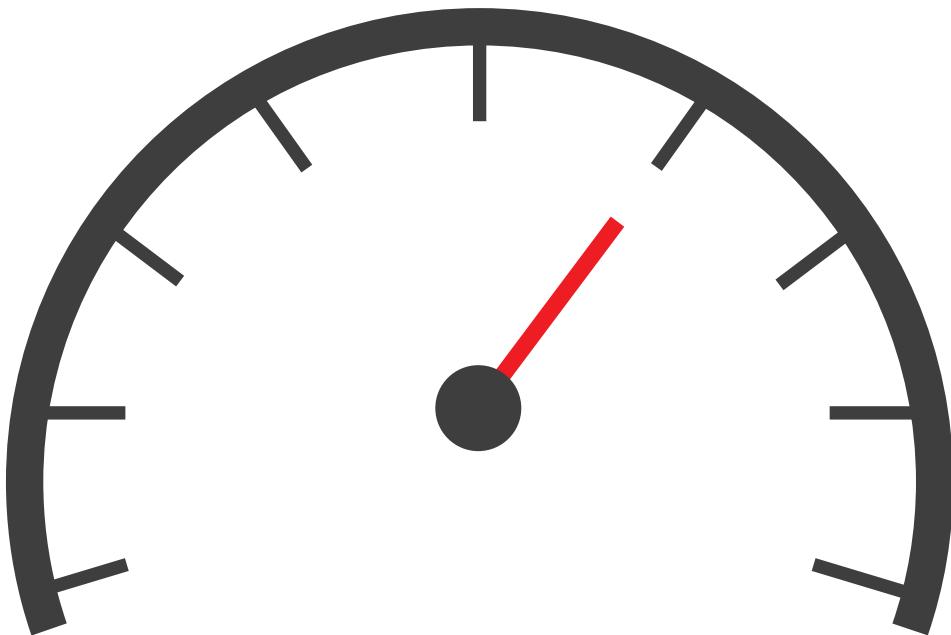
ASSEMBLY DIFFICULTY:



**MICRODUINO**

## About Revolutions Per Minutes (RPM)

Revolutions per minute (abbreviated rpm, RPM, rev/min, r/min) is a measure of the frequency of rotation, specifically the number of rotations around a fixed axis in one minute. It is used as a measure of rotational speed of a mechanical component.



# Review of Modules



mCenter+

Is a five-in-one module which contains:

- **Li-ion Battery:** battery pack to power your projects which can be recharged via the USB port.
- **mCookie Core+:** a tiny central processing unit (CPU), which is the brains of your projects.
- **USB Programmer:** programs the Core+ module and acts as a serial communication bridge between your computer and Core+ module.
- **Bluetooth Upload:** similar to the USB programmer, but works wirelessly over Bluetooth communication.
- **Hub:** connects various modules such as sensors, trinkets, and other modules to your projects. These modules can then be accessed by the Core+ module.

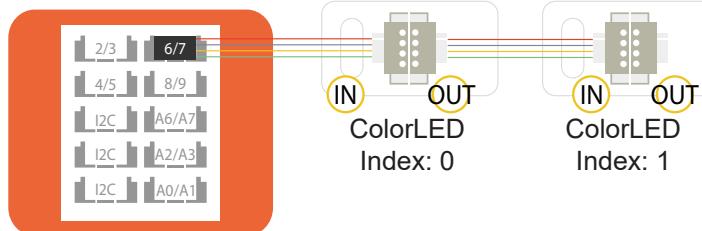
# Review of Modules

## Color LED



The Color LED is a LED composed of 3 smaller LEDs which emit red, green, and blue. Each color's brightness value can be configured individually. This can produce a wide variety of colors. It is controlled over a special communication protocol which is different from the basic "Digital Output" (used by Single-Color LED).

### Indexing



- The Color LED connected directly to the Hub is index 0.
- The Color LED connected next in the chain is index 1.
- And so on.

## Usage

Hub pin the 1st Color LED is connected.

Index of the Color LED to configure.

Color to change the Color LED to.

Hub pin the 1st Color LED is connected.

Index of the Color LED to configure.

Transition from this color.

Set duration (in milliseconds) for the transition.

LED pin 2  
No. 0  
color blue

LED pin 2  
No. 0  
from yellow  
to purple  
duration 500

### Color LED - Color Select block

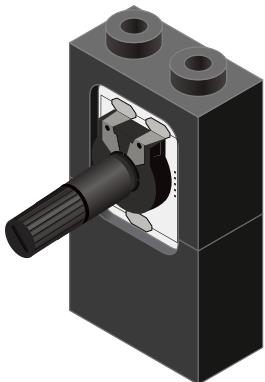
This block configures what color a particular Color LED displays.

### Color LED - Color Transition block

This block causes a particular Color LED to transition from one color to another color under a time span. (Note: This block must complete its transition before proceeding to the next block in the script.)

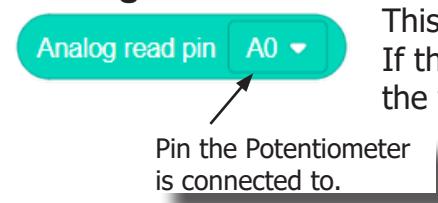
# Review of Modules

## Potentiometer

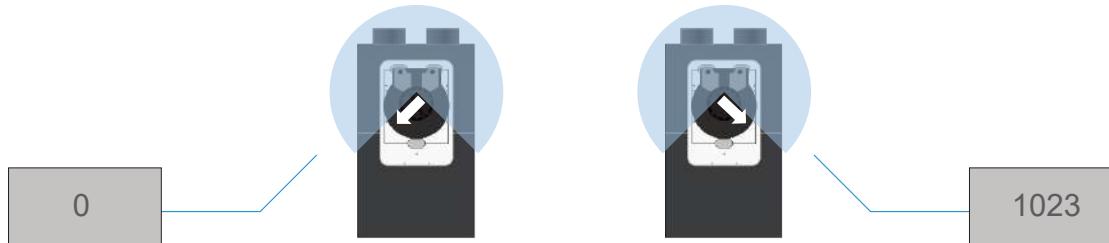


A Potentiometer is an input module which can be turned or rotated right or left. Based on the position of the knob, it produces a different signal value for the Core module to read. It must be connected to an analog port (ports denoted with "A").

## Analog Read block

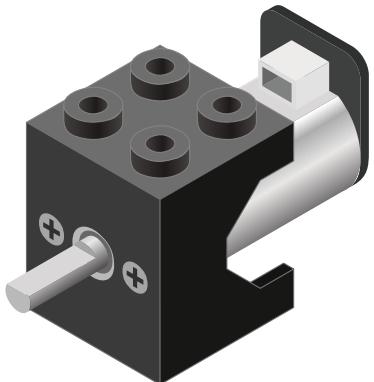


This block returns the value from the Potentiometer. A value between 0 and 1023 is returned. If the potentiometer is turned all the way to the left, it produces a 0 value. If it is turned all the way to the right, it produces a 1023 value. In the middle it produces a value around 512.



# Introduction of New Modules

## Motor



**Motor - Set Speed block**



A Motor is a electrical machine which converts electrical energy to mechanical energy. It is capable of rotating clockwise and counter clockwise. It is controlled by a Motor Controller. A Motor Controller can control up to 2 Motors. A Wheel can be attached to the Motor.

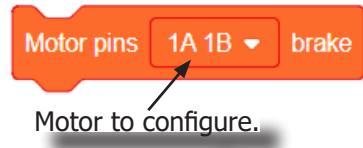


**Motor Controller**



**Wheel**

**Motor - Brake block**



This block sets the speed to 0 to stop the Motor.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

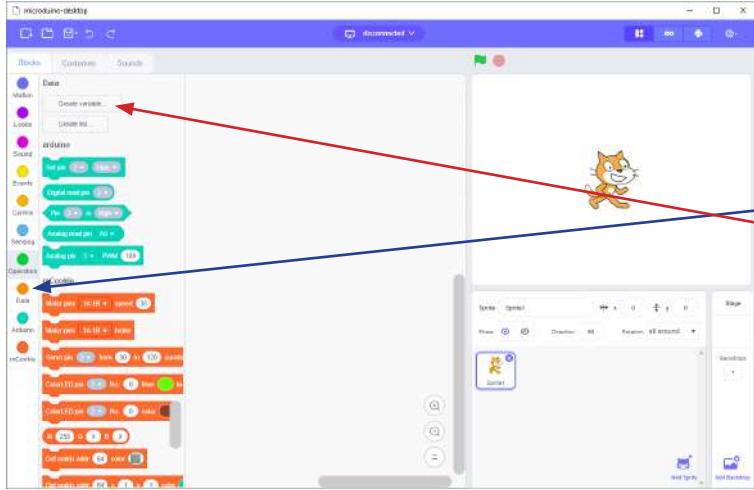
## Greater Than Block



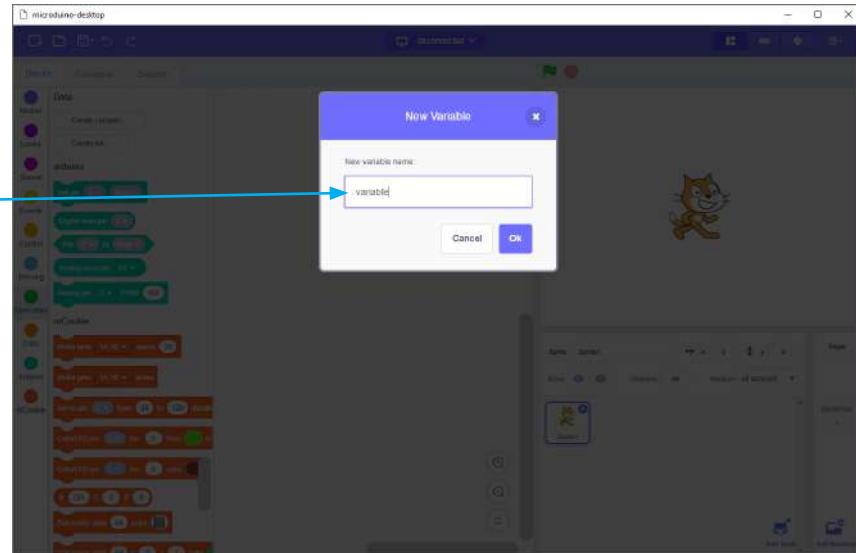
This block returns TRUE if the left input is greater than the right input.

# Review of Concepts

## Creating a Variable in Scratch



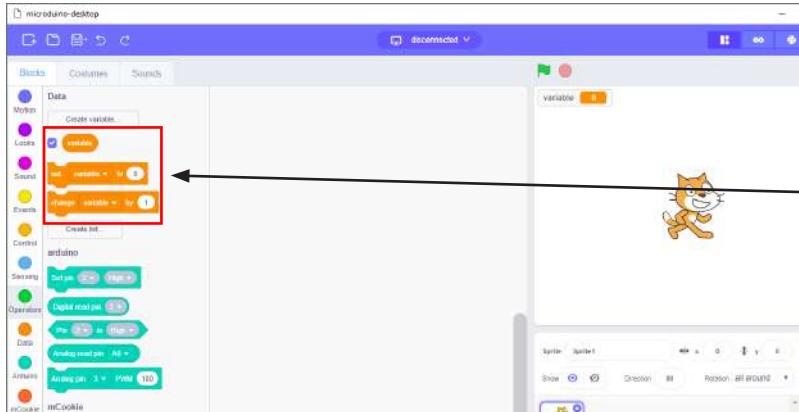
Click on the "Variable" category.  
Then click on the "Create variable..." button.



Name your variable.

# Review of Concepts

## Creating a Variable in Scratch (continued)



Several new blocks will appear which allows access and modification of the new variable.

### Variable Value block

**variable** This block returns the current value of the variable.

### Variable Set block

**set [variable] to [value]** This block sets the value of the variable. It will overwrite the current value.

### Variable Change block

**change [variable] by [value]** This block changes the variable value by adding the input value (negative values or blocks are accepted).  
input value

## The Problem

How can we create a motor controller with a revolutions indicator?

# Project Worksheet

Complete the worksheet below to your best abilities.

1. A solution.

Use a Potentiometer to control the speed of the Motor. Use a Color LED to indicate the revolutions per minute.

2. List all the parts needed for the solution. What is the purpose of each part?

3. What is a simple explanation of the logic for the solution?

4. Create a flow chart of the solution.

5. How do you assemble and connect the modules for the solution?

6. Configure blocks to relate to the flow chart.

# Project Worksheet - Answers

2. List all the parts needed for the solution. What is the purpose of each part?

1x  mCenter+ - control the project, connect sensors and trinkets to project, and supply power.

1x  USB Cable - connect mCenter+ to a computer for charging and programming the project.

2x  Hub Connector Cable - connect sensors and trinkets to the mCenter+.

1x  Motor Controller - controls and drives the attached motors.

1x  Motor - motor to control the speed of.

1x  Potentiometer - used as an input to control the speed of the motor.

1x  Color LED - indicate the revolutions per minute of the motor.

## Project Worksheet - Answers

3. What is a simple explanation of the logic for the solution?

Use a Potentiometer as an input to adjust the speed of the Motor.

Use a Color LED to indicate the current revolutions per minute.

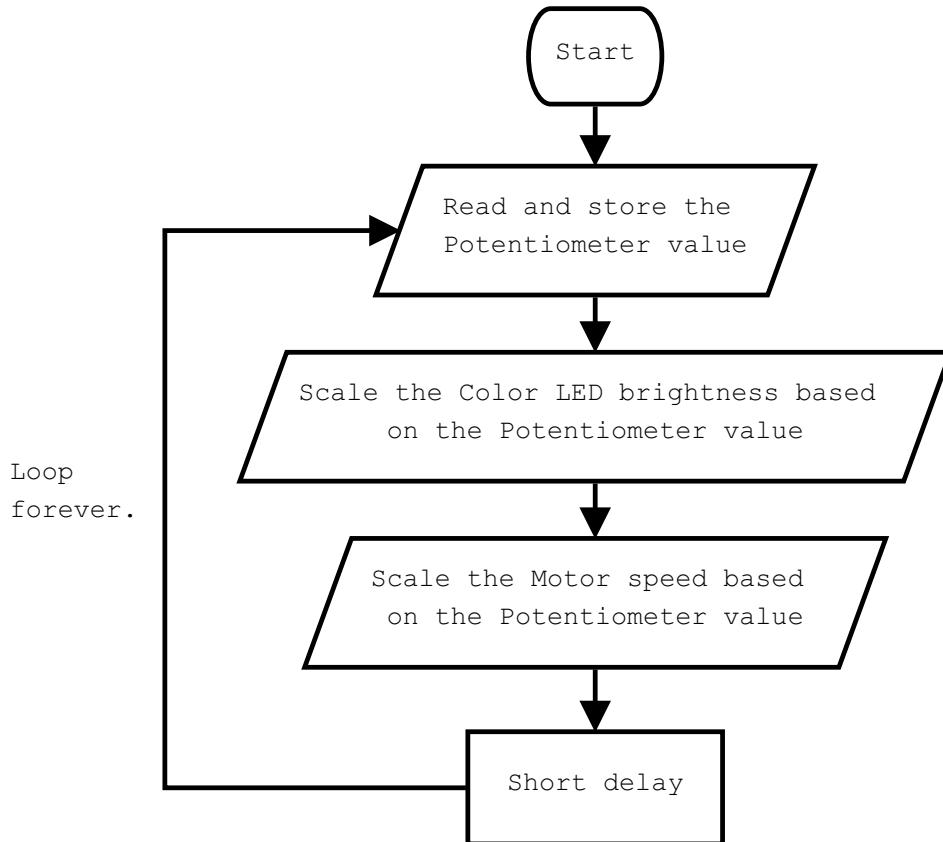
Read in the value of the Potentiometer (between 0 and 1023).

Map the Motor's speed based on the Potentiometer value (accepts values between 0 and 255).

Map the Color LED's brightness level based on the Potentiometer value (accepts values between 0 and 255).

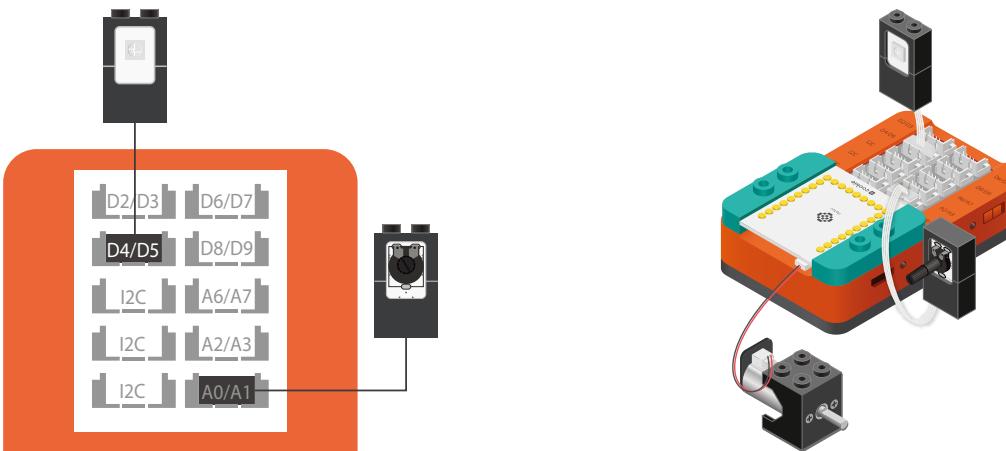
# Project Worksheet - Answers

4. Create a flow chart of the solution.



# Project Worksheet - Answers

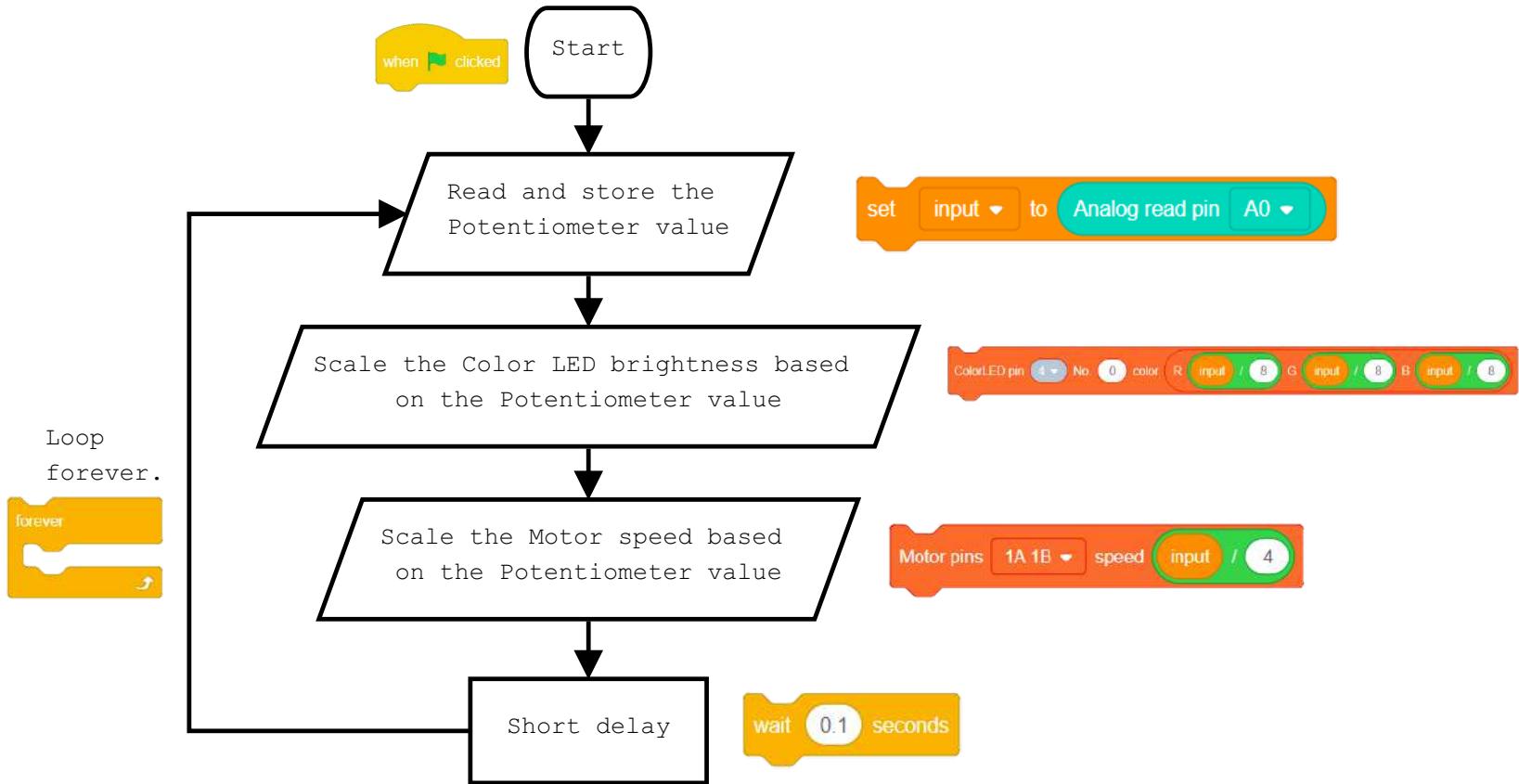
5. How do you assemble and connect the modules for the solution?



1. Stack the Motor Controller onto the mCenter+.
2. Connect a Motor to connector “1A1B” on the Motor Controller.
3. Connect a Color LED (IN port) to pin 4/5.
4. Connect a Potentiometer to pin A0/A1.
5. Insert one end of the USB cable into mCenter+ and connect the other to a computer.

# Project Worksheet - Answers

6. Configure blocks to relate to the flow chart.



# The Script

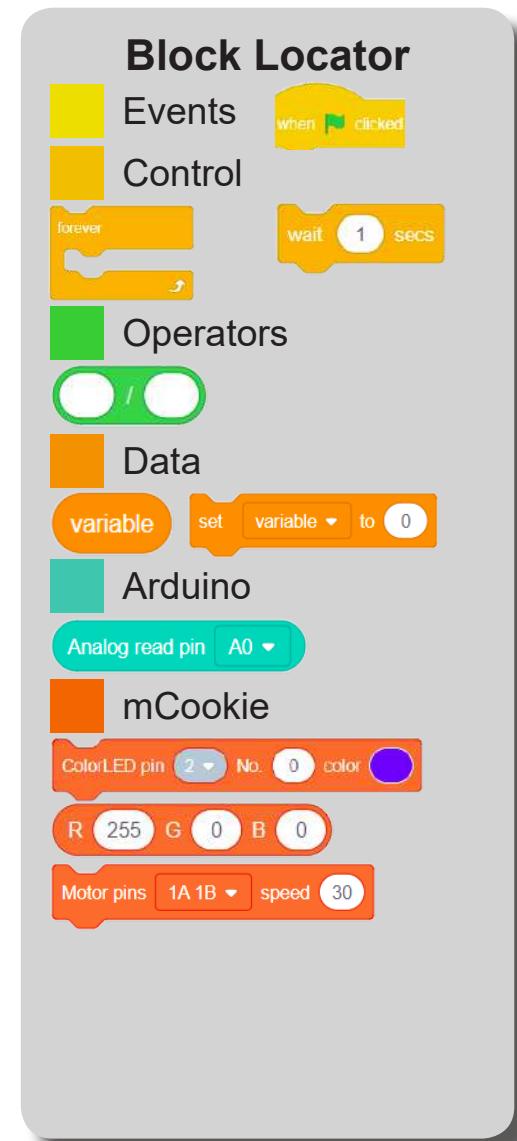
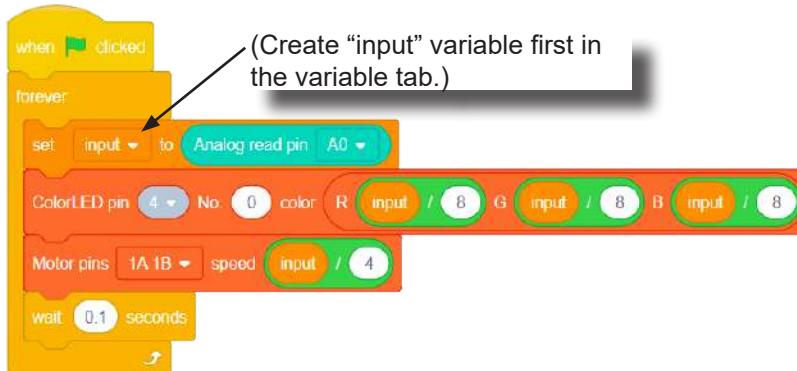
Click Green Flag to activate script.

Read and store the value  
of the Potentiometer (pin A0).

Indicated the RPM with  
the Color LED (pin 4).

Set Motor (1) speed based  
on Potentiometer value.

Short delay before looping.  
Loop.

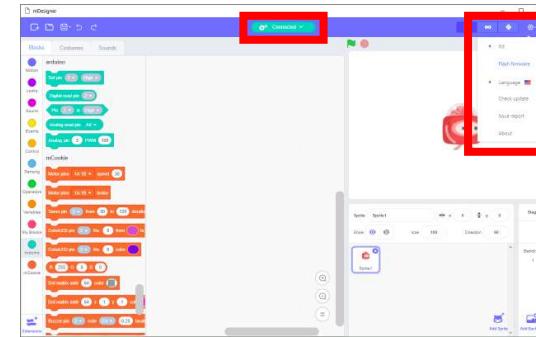


# Testing the Program

## Important:

- Ensure that mCenter+ is on. Flip the switch on mCenter+ to turn it on. An LED on mCenter+ will light up to indicate power is being supplied.
- Ensure you have selected the Serial Port and flashed the special firmware (mCookie) first before testing (this only needs to be done once unless the firmware is overwritten).
- Ensure you have selected the Serial Port and connected before testing.

## Select Serial Port



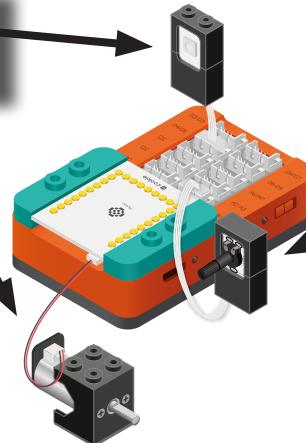
Flash firmware

## Testing:

Click the green flag to activate the script.

```
when green flag clicked
  forever
    set [input v] to (analog read pin [A0 v])
    ColorLED pin [No. 0 color R input / B G input / B B input / B]
    Motor pins [1A 1B speed input / 4]
    wait [0.1 seconds]
```

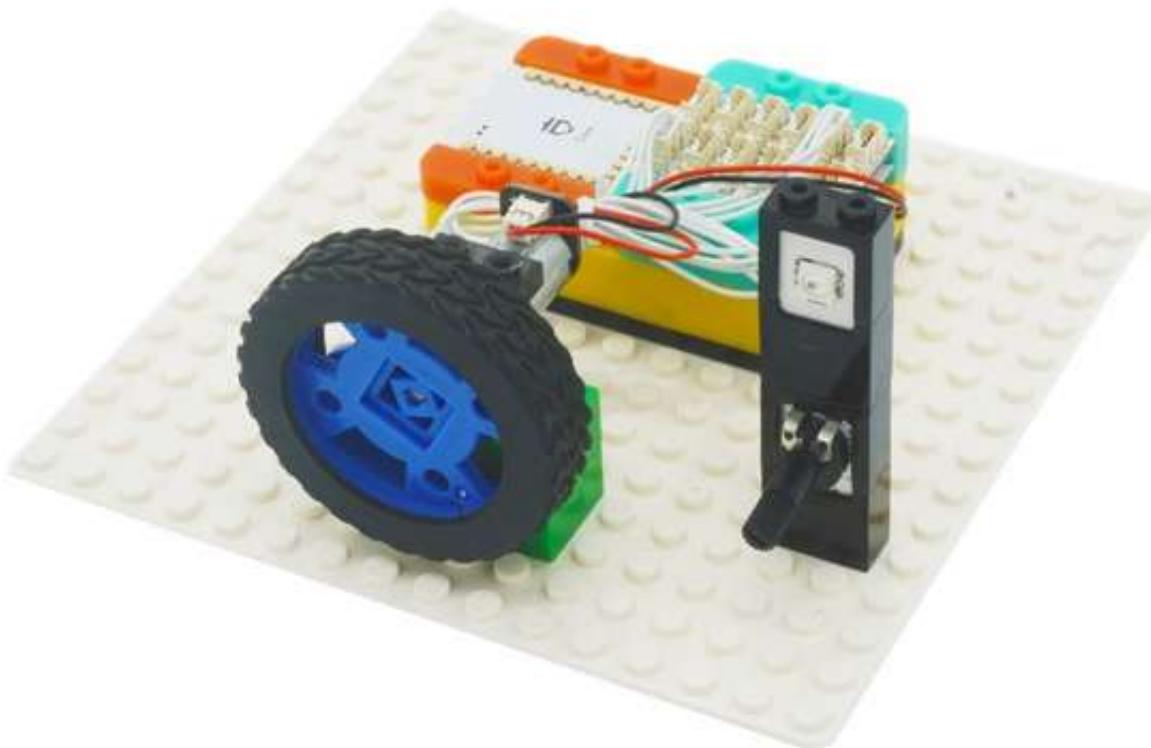
The Motor and Color LED will react to the position of the Potentiometer.



1. Rotate the Potentiometer to adjust the speed of the Motor.

## Create a Structure

Create a structure for the project using building blocks, paper, or other crafting materials.



## Project Challenge

Scale the Potentiometer input so that the values between 0 and 511 causes the Motor to rotate backwards (-255~0).

Scale the Potentiometer input so that the values between 512 and 1023 causes the Motor to rotate forwards (0~255).

When the Motor is rotating backwards, make the Color LED red.

When the Motor is rotating forwards, make the Color LED green.

## Review



Motor Controller - controls and drives attached motors. Can control up to 2 motors.



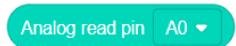
Motor - converts electrical energy into mechanical energy in the form of torque.



Motor - Set Speed block - set the speed of a motor.



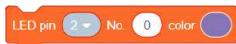
Potentiometer - input module with an adjustable knob.



Analog Read block - reads the value from the potentiometer. Returns a value between 0 and 1023.



Color LED - configurable to display a wide range of colors.



Color LED - Color Select block - configures what color a particular Color LED displays.

## Expanding the Project

Write down a way of expanding this project. Create it on your own.

---

---

---

---

---

---

---

---

---

---

## Research

Research more about revolutions per minute (RPM).

List 5 contexts where this term is commonly used.

---

---

---

---

---

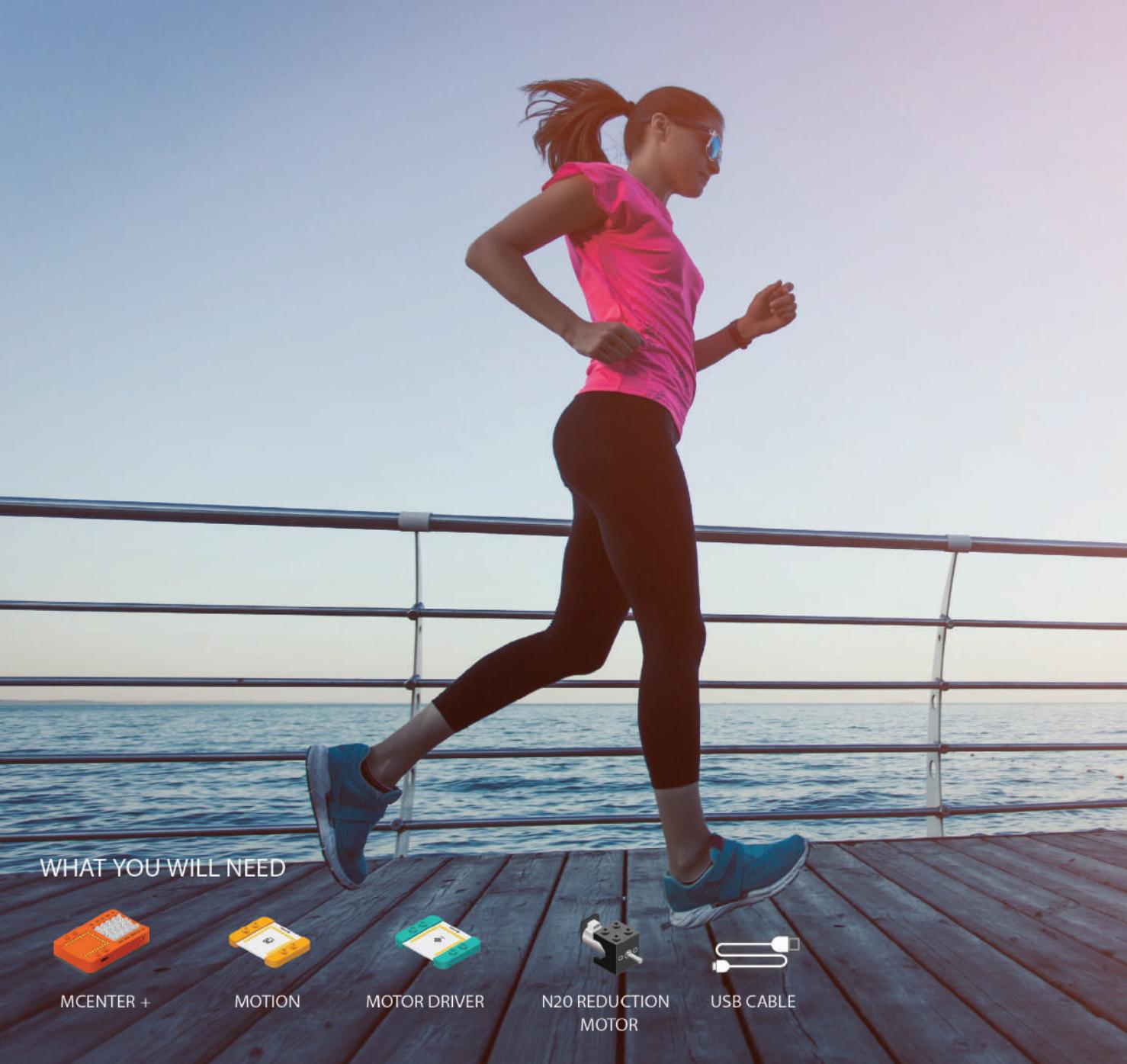
---

---

---

---

---



#### WHAT YOU WILL NEED



MCENTER +



MOTION



MOTOR DRIVER



N20 REDUCTION  
MOTOR



USB CABLE

MIX4A02

# Motion Sensor Motor

Control the speed and direction of the motor using a motion sensor.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



ASSEMBLY DIFFICULTY:

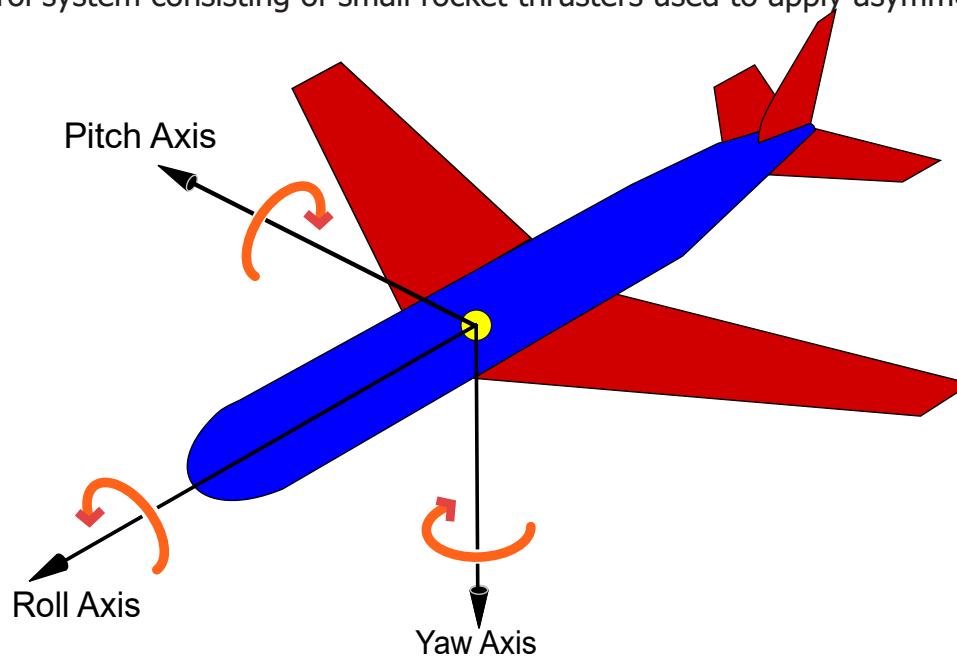


**MICRODUINO**

## About Roll, Pitch and Yaw

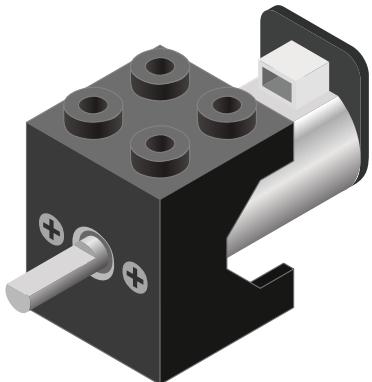
An aircraft in flight is free to rotate in three dimensions: pitch, nose up or down about an axis running from wing to wing; yaw, nose left or right about an axis running up and down; and roll, rotation about an axis running from nose to tail. The axes are alternatively designated as lateral, vertical, and longitudinal. These axes move with the vehicle and rotate relative to the Earth along with the craft. These definitions were analogously applied to spacecraft when the first manned spacecraft were designed in the late 1950s.

These rotations are produced by torques (or moments) about the principal axes. On an aircraft, these are intentionally produced by means of moving control surfaces, which vary the distribution of the net aerodynamic force about the vehicle's center of mass. Elevators (moving flaps on the horizontal tail) produce pitch, a rudder on the vertical tail produces yaw, and ailerons (flaps on the wings that move in opposing directions) produce roll. On a spacecraft, the moments are usually produced by a reaction control system consisting of small rocket thrusters used to apply asymmetrical thrust on the vehicle.



# Review of Modules

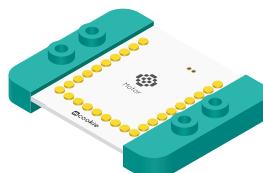
## Motor



**Motor - Set Speed block**



A Motor is a electrical machine which converts electrical energy to mechanical energy. It is capable of rotating clockwise and counter clockwise. It is controlled by a Motor Controller. A Motor Controller can control up to 2 Motors. A Wheel can be attached to the Motor.

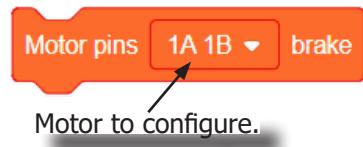


**Motor Controller**



**Wheel**

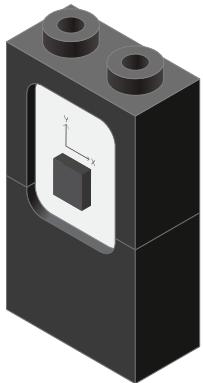
**Motor - Brake block**



This block sets the speed to 0 to stop the Motor.

# Introduction of New Modules

## Motion



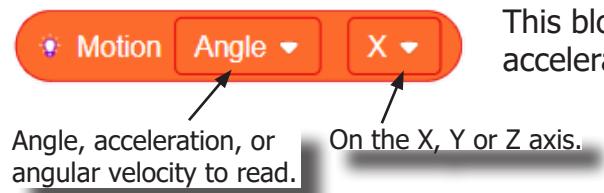
The Motion module is a triple axis accelerometer and gyroscope.

It is capable of measuring on the X, Y & Z axis the:

- Angle
- Acceleration
- Angular velocity

It uses the "I2C" communication protocol and must be connected to an "I2C" port on mCenter+.

### Motion - Read block



This block reads data from the Motion module. It is capable of retrieving the angle, acceleration and angular velocity on the X, Y or Z axis.

Angle, acceleration, or  
angular velocity to read.

On the X, Y or Z axis.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

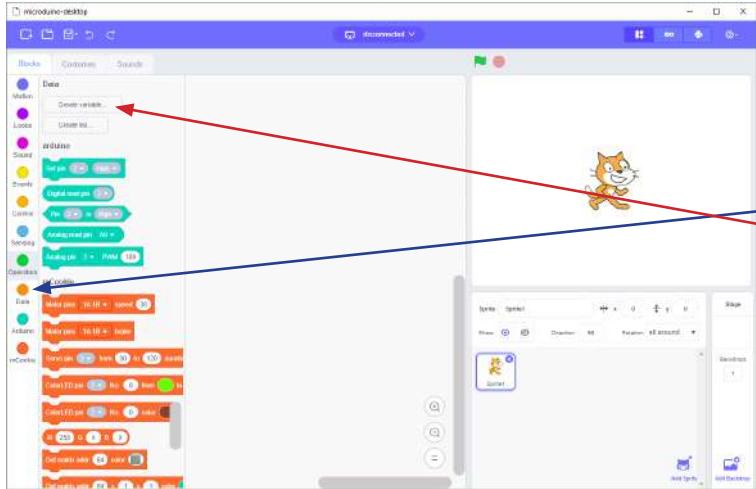
## Greater Than Block



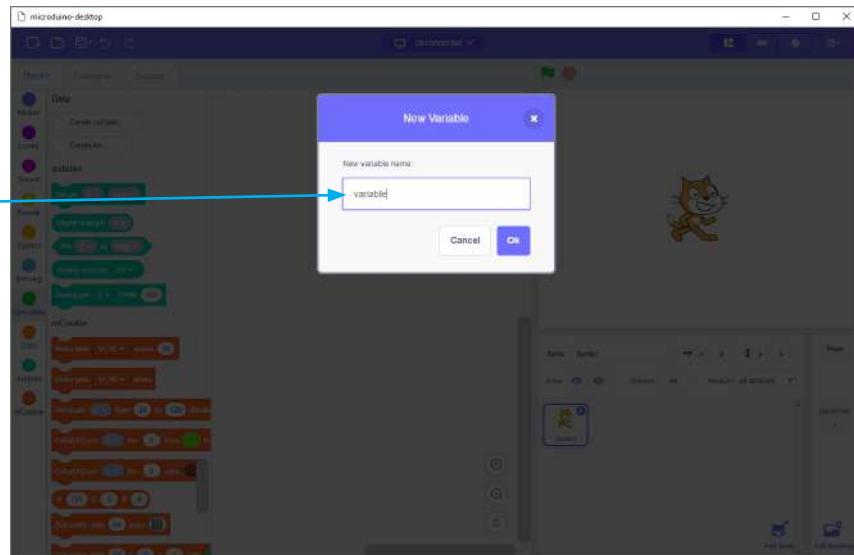
This block returns TRUE if the left input is greater than the right input.

# Review of Concepts

## Creating a Variable in Scratch



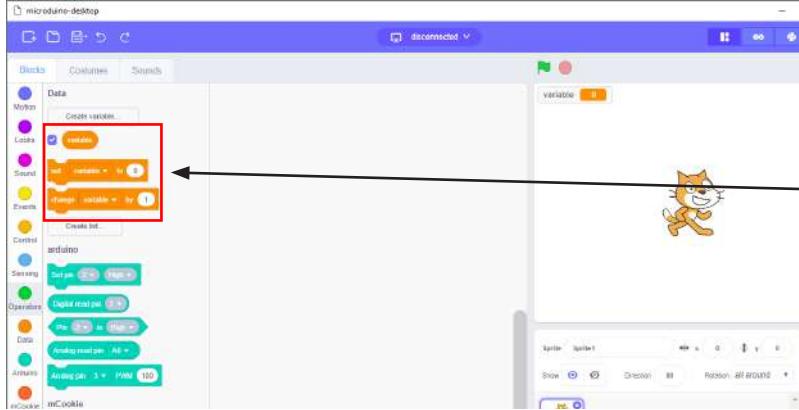
Click on the “Variable” category.  
Then click on the “Create variable...” button.



Name your variable.

# Review of Concepts

## Creating a Variable in Scratch (continued)



Several new blocks will appear which allows access and modification of the new variable.

### Variable Value block

**variable** This block returns the current value of the variable.

### Variable Set block

**set [variable v] to [0]** This block sets the value of the variable. It will overwrite the current value.

### Variable Change block

**change [variable v] by [1]** This block changes the variable value by adding the input value (negative values or blocks are accepted).

input value

## The Problem

How can we control a motor using a motion module?

# Project Worksheet

Complete the worksheet below to your best abilities.

1. A solution.

Use a Motion module to detect the pitch angle. Change the speed and direction of the Motor based on the pitch angle.

2. List all the parts needed for the solution. What is the purpose of each part?

3. What is a simple explanation of the logic for the solution?

4. Create a flow chart of the solution.

5. How do you assemble and connect the modules for the solution?

6. Configure blocks to relate to the flow chart.

## Project Worksheet - Answers

2. List all the parts needed for the solution. What is the purpose of each part?

1x  mCenter+ - control the project, connect sensors and trinkets to project, and supply power.

1x  USB Cable - connect mCenter+ to a computer for charging and programming the project.

1x  Hub Connector Cable - connect sensors and trinkets to the mCenter+.

1x  Motor Controller - controls and drives the attached motors.

1x  Motor - motor to control the speed of.

1x  Motion - detect motion.

## Project Worksheet - Answers

3. What is a simple explanation of the logic for the solution?

Use a Motion module to detect the pitch.

Read and store the "X" angle (pitch).

Constrain the value between -90 and 90.

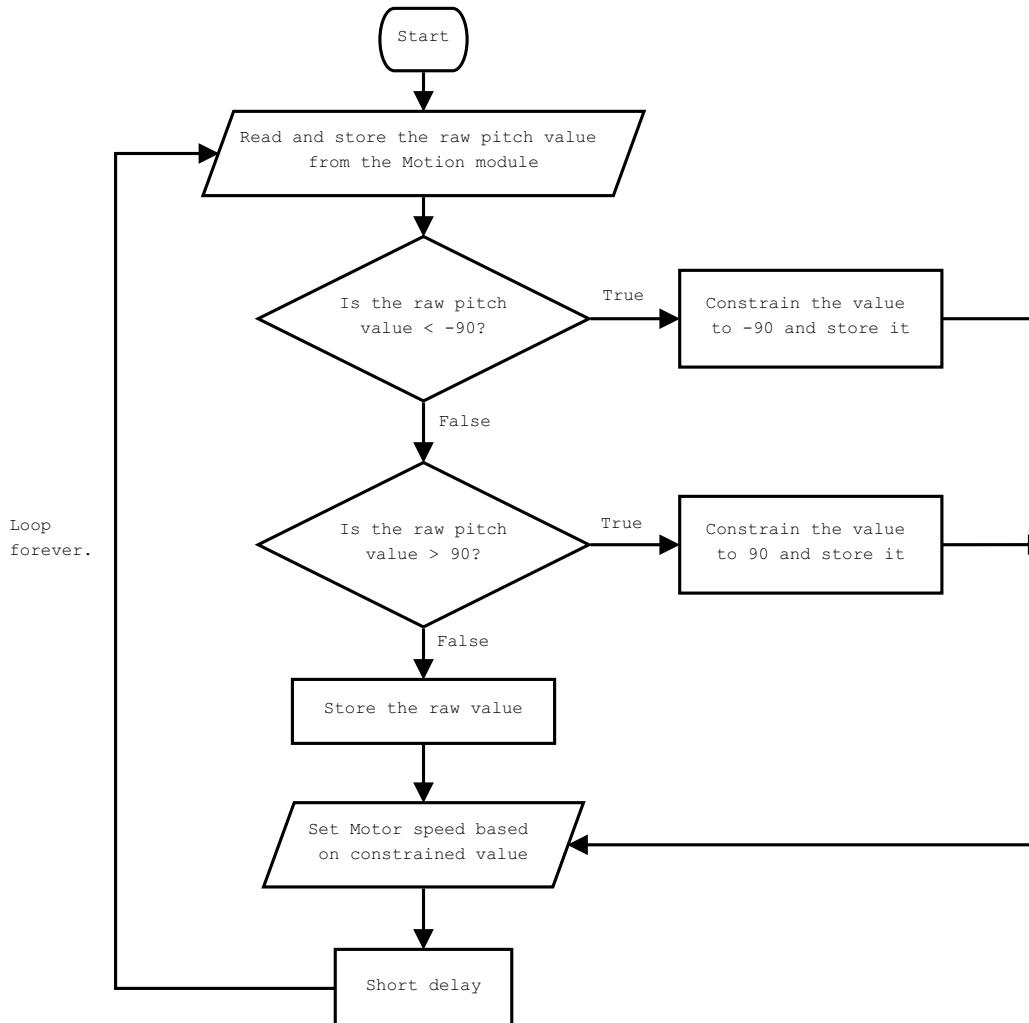
Multiple the value by 2.8 and use that to set the speed of the motor:

- $-90 * 2.8 = -252$
- $90 * 2.8 = 252$

Produces a value between -252 and 252 for speed of the motor.

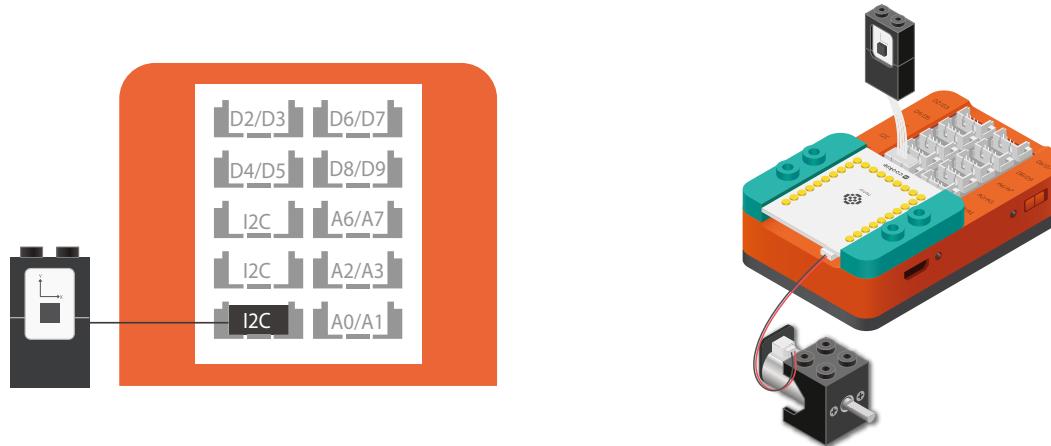
# Project Worksheet - Answers

## 4. Create a flow chart of the solution.



# Project Worksheet - Answers

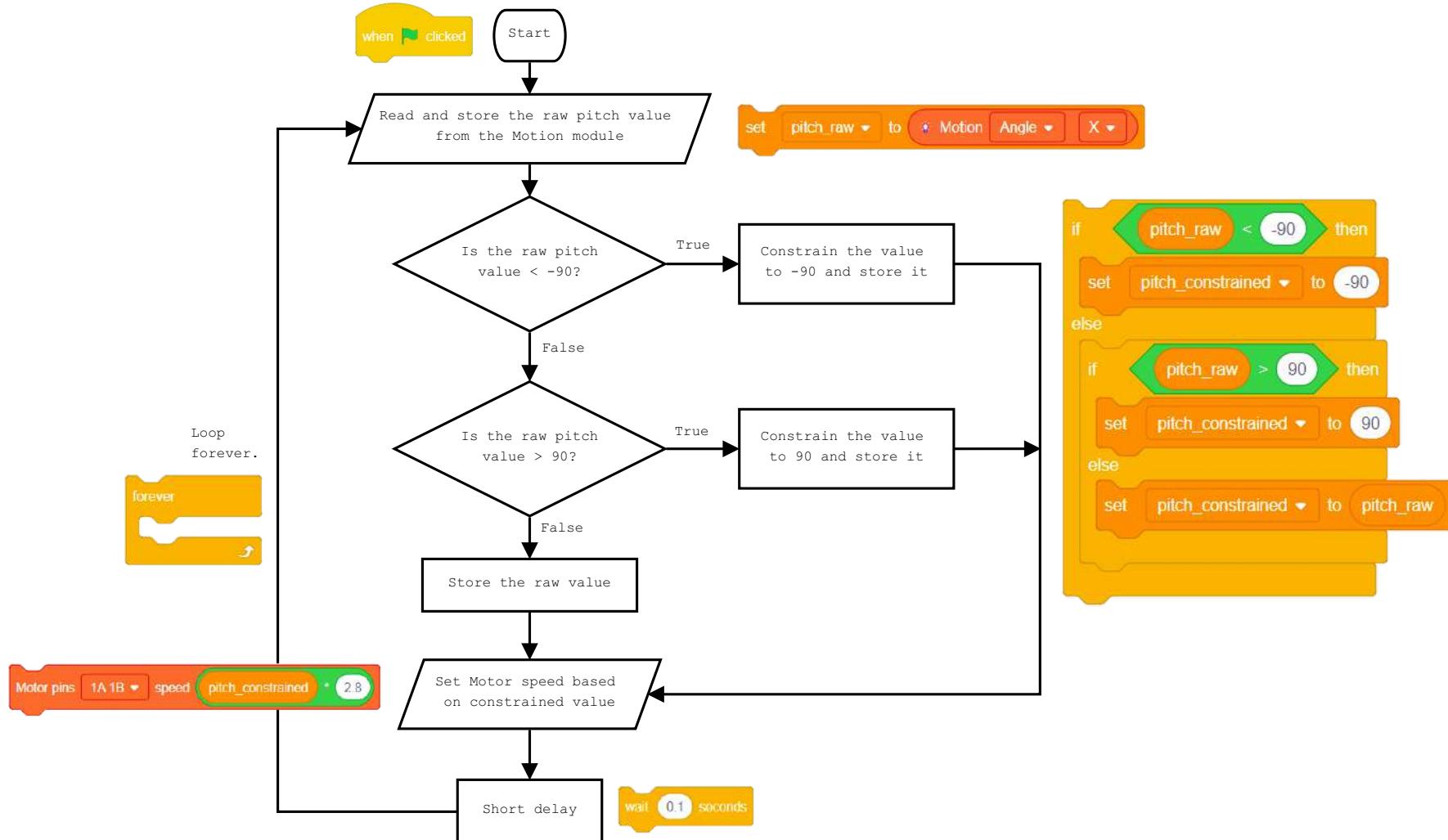
5. How do you assemble and connect the modules for the solution?



1. Stack the Motor Controller onto the mCenter+.
2. Connect a Motor to connector “1A1B” on the Motor Controller.
3. Connect the Motion module to a “I2C” pin.
4. Insert one end of the USB cable into mCenter+ and connect the other to a computer.

# Project Worksheet - Answers

## 6. Configure blocks to relate to the flow chart.



# The Script

Click Green Flag to activate script.

Read and store the “X” angle from the Motion module.

Check if the pitch (raw) value is < -90.

Constrain it to -90.

Check if the pitch (raw) value is > 90.

Constrain it to 90.

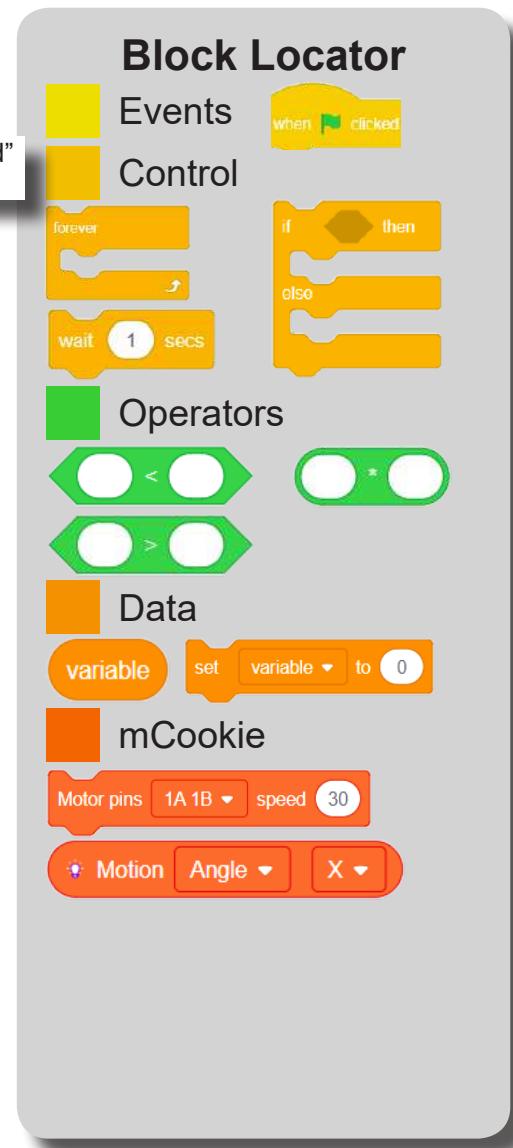
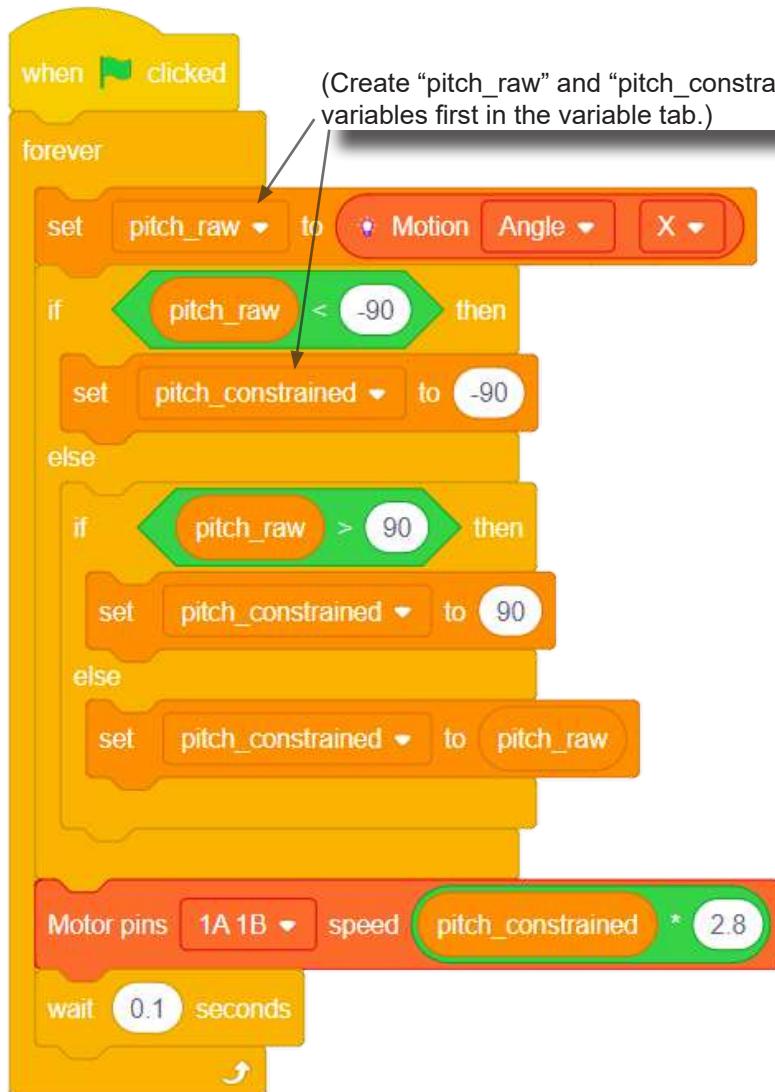
If between -90 and 90.  
Then accept the raw value.

Set the Motor (1) speed based on the constrained “X” angle value.

Short delay before looping.

Loop.

(Create “pitch\_raw” and “pitch\_constrained” variables first in the variable tab.)

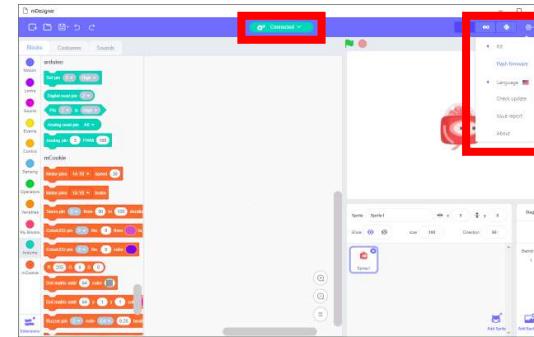


# Testing the Program

## Important:

- Ensure that mCenter+ is on. Flip the switch on mCenter+ to turn it on. An LED on mCenter+ will light up to indicate power is being supplied.
- Ensure you have selected the Serial Port and flashed the special firmware (mCookie) first before testing (this only needs to be done once unless the firmware is overwritten).
- Ensure you have selected the Serial Port and connected before testing.

## Select Serial Port



Flash firmware

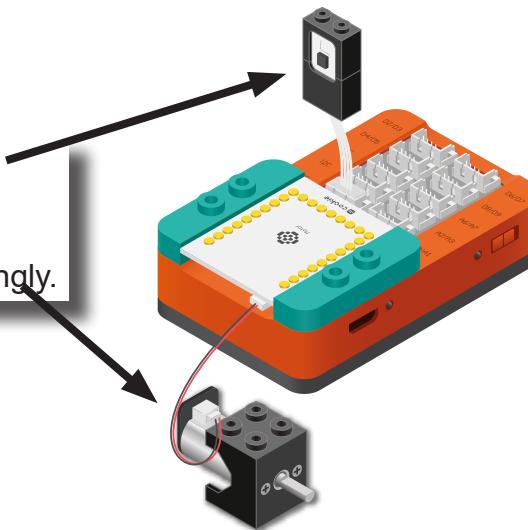
## Testing:

Click the green flag to activate the script.



1. Rotate the Motion module along the "X" axis (pitch).

2. The Motor will turn accordingly.



## Create a Structure

Create a structure for the project using building blocks, paper, or other crafting materials.



## Project Challenge

Constrain the “X” angle to -20 and 20.  
How does the motor react to this new constrained angle?

## Review



Motor Controller - controls and drives attached motors. Can control up to 2 motors.



Motor - converts electrical energy into mechanical energy in the form of torque.



Motor - Set Speed block - set the speed of a motor.



Motion - module which detects motion such as roll, yaw and pitch.



Motion - Read block - returns various motion readings from the Motion module.

## Expanding the Project

Write down a way of expanding this project. Create it on your own.

---

---

---

---

---

---

---

---

---

---

---

## Research

Research more about roll, pitch and yaw.  
Why is this information important to aircrafts?

---

---

---

---

---

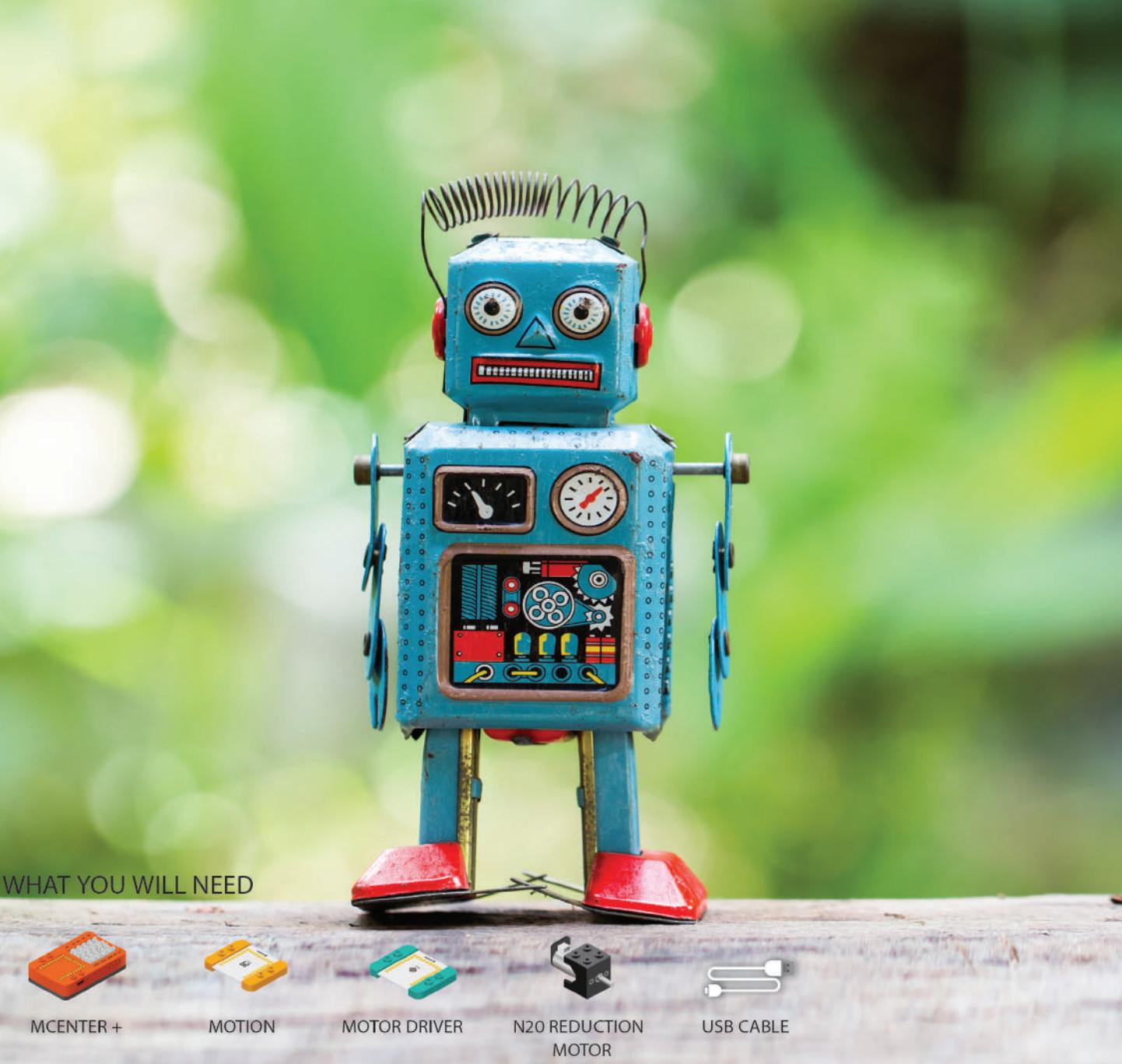
---

---

---

---

---



## WHAT YOU WILL NEED



MCENTER +



MOTION



MOTOR DRIVER



N20 REDUCTION  
MOTOR



USB CABLE

MIX4A03

# Emoji Robot

Control the robot's eye expressions with a motion module.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



ASSEMBLY DIFFICULTY:

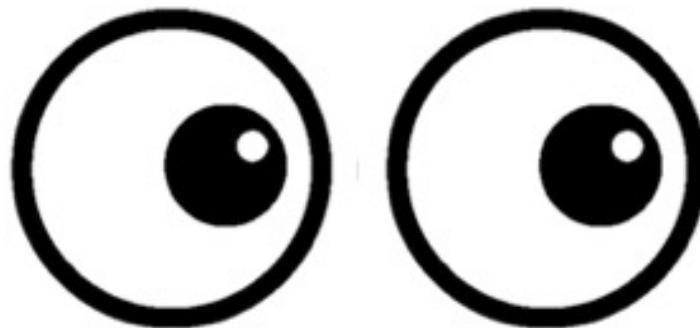


**MICRODUINO**

## About Vision Stabilization

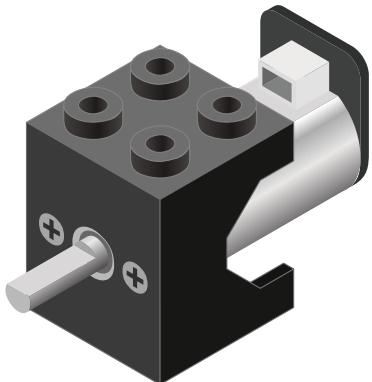
The vestibulo-ocular reflex (VOR) is a reflex, where activation of the vestibular system causes eye movement. This reflex functions to stabilize images on the retinas during head movement by producing eye movements in the direction opposite to head movement, thus preserving the image on the center of the visual field(s).

For example, when the head moves to the right, the eyes move to the left, and vice versa. Since slight head movement is present all the time, the VOR is very important for stabilizing vision: patients whose VOR is impaired find it difficult to read using print, because they cannot stabilize the eyes during small head tremors, and also because damage to the VOR can cause vestibular nystagmus.

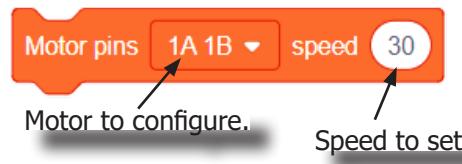


# Review of Modules

## Motor



**Motor - Set Speed block**



Motor to configure.

Speed to set.

A Motor is a electrical machine which converts electrical energy to mechanical energy. It is capable of rotating clockwise and counter clockwise. It is controlled by a Motor Controller. A Motor Controller can control up to 2 Motors. A Wheel can be attached to the Motor.

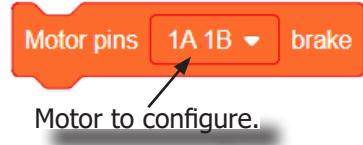


**Motor Controller**



**Wheel**

**Motor - Brake block**

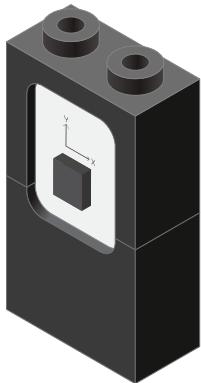


Motor to configure.

This block sets the speed to 0 to stop the Motor.

# Review of Modules

## Motion



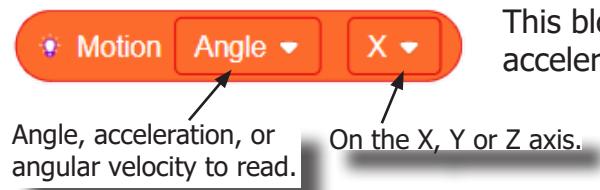
The Motion sensor is a triple axis accelerometer and gyroscope.

It is capable of measuring on the X, Y & Z axis the:

- Angle
- Acceleration
- Angular velocity

It uses the "I2C" communication protocol and must be connected to an "I2C" port on mCenter+.

### Motion - Read block



This block reads data from the Motion module. It is capable of retrieving the angle, acceleration and angular velocity on the X, Y or Z axis.

Angle, acceleration, or  
angular velocity to read.

On the X, Y or Z axis.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

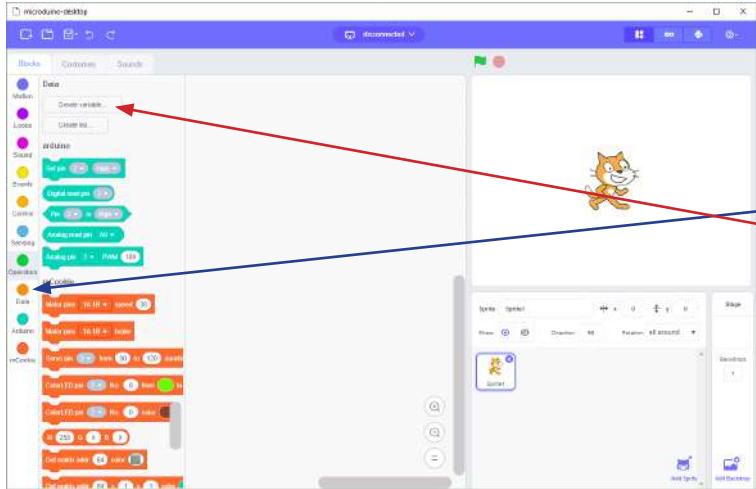
## Greater Than Block



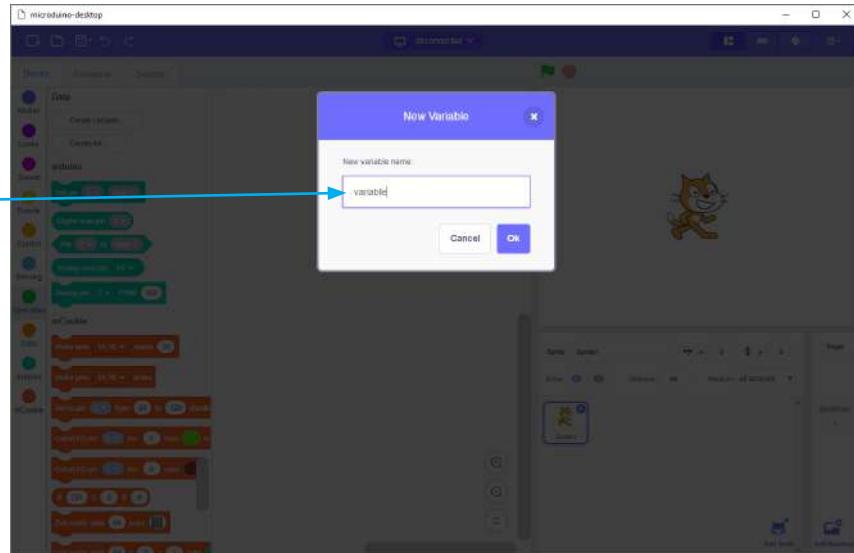
This block returns TRUE if the left input is greater than the right input.

# Review of Concepts

## Creating a Variable in Scratch

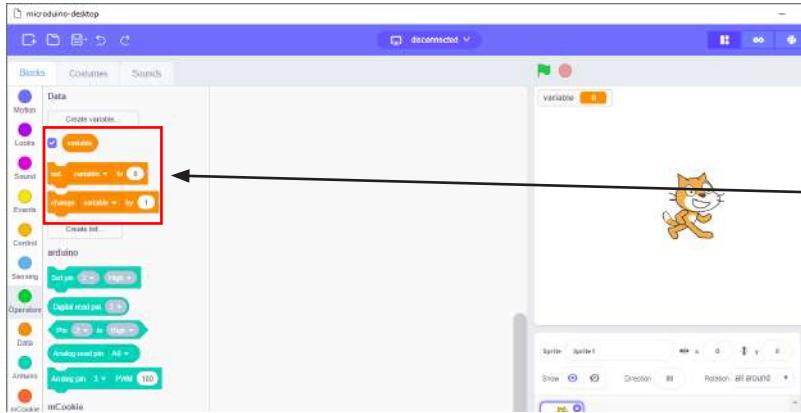


Click on the "Variable" category.  
Then click on the "Create variable..." button.



# Review of Concepts

## Creating a Variable in Scratch (continued)



Several new blocks will appear which allows access and modification of the new variable.

### Variable Value block

**variable** This block returns the current value of the variable.

### Variable Set block

**set [variable v] to [0]** This block sets the value of the variable. It will overwrite the current value.

### Variable Change block

**change [variable v] by [1]** This block changes the variable value by adding the input value (negative values or blocks are accepted).  
input value

## The Problem

How can we create a motion sensitive Emoticon robot?

# Project Worksheet

Complete the worksheet below to your best abilities.

1. A solution.

Use the Motion module to detect the “X” angle (pitch). Set the eyes (2 Motors) spin speed based on the pitch value.

2. List all the parts needed for the solution. What is the purpose of each part?

3. What is a simple explanation of the logic for the solution?

4. Create a flow chart of the solution.

5. How do you assemble and connect the modules for the solution?

6. Configure blocks to relate to the flow chart.

# Project Worksheet - Answers

2. List all the parts needed for the solution. What is the purpose of each part?

1x  mCenter+ - control the project, connect sensors and trinkets to project, and supply power.

1x  USB Cable - connect mCenter+ to a computer for charging and programming the project.

1x  Hub Connector Cable - connect sensors and trinkets to the mCenter+.

1x  Motor Controller - controls and drives the attached motors.

2x  Motor - motor to simulate the robot's eyes.

2x  Wheel - attach to motor.

1x  Motion - detect motion.

## Project Worksheet - Answers

3. What is a simple explanation of the logic for the solution?

Use a Motion module to detect the pitch.

Use 2 Motors (with Wheels) to simulate the robot's eyes.

Read and store the "X" angle (pitch).

Constrain the value between -20 and 20.

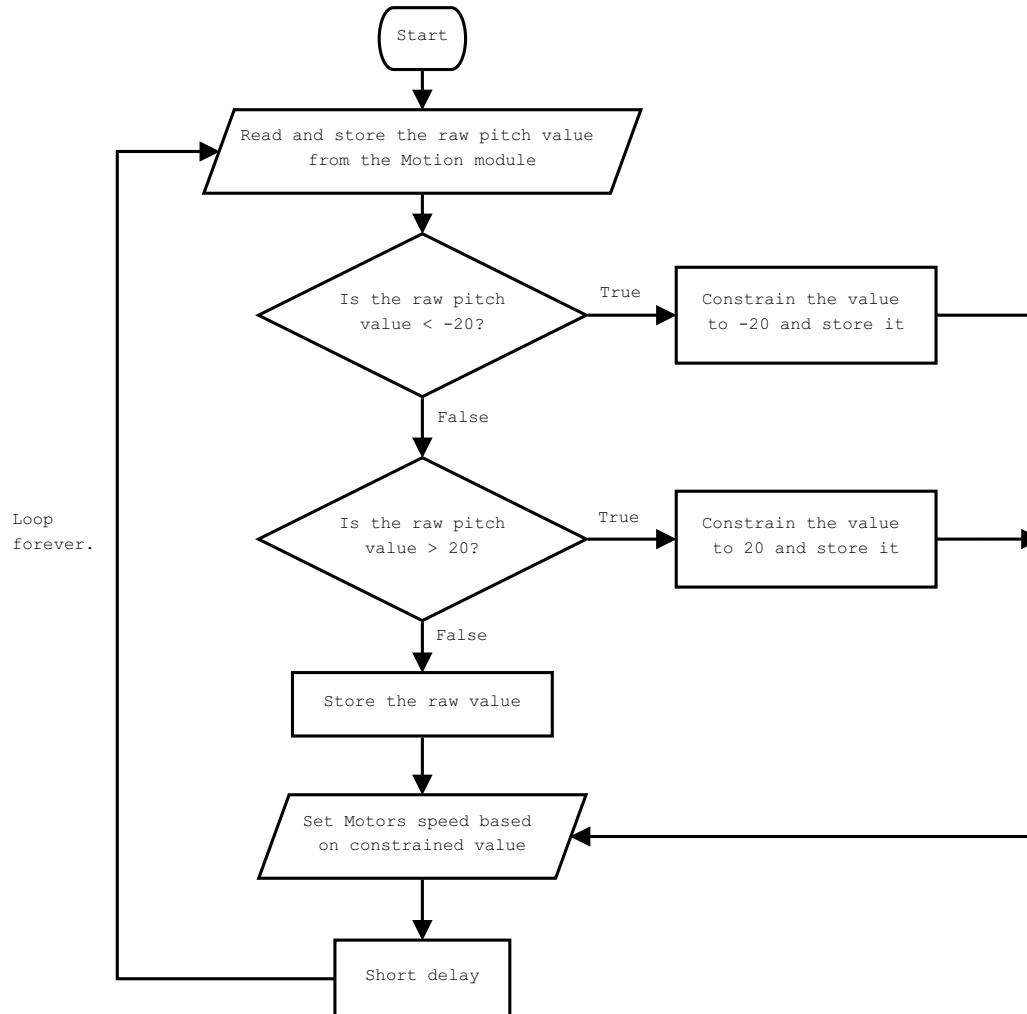
Multiple the value by 5 and use that to set the speed of the motors:

- $-20 * 5 = -100$
- $20 * 5 = 100$

Produces a value between -100 and 100 for speed of the motors.

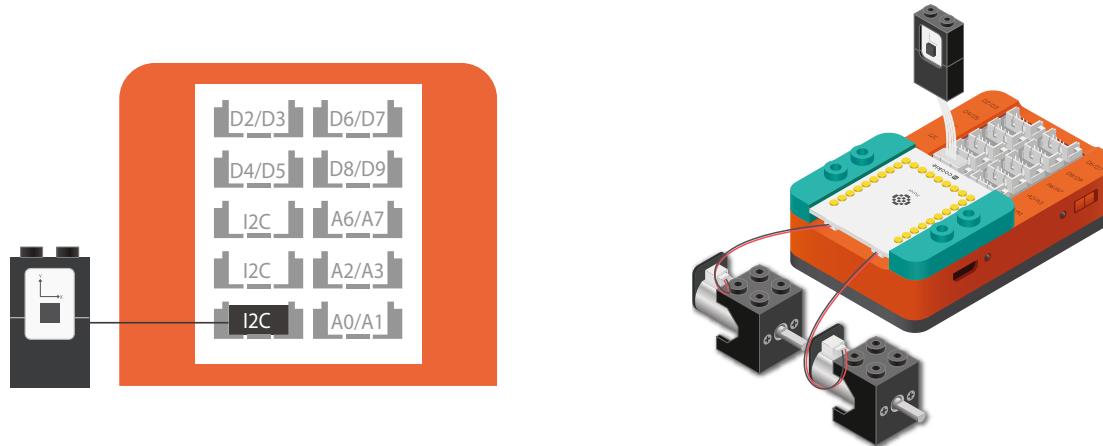
# Project Worksheet - Answers

## 4. Create a flow chart of the solution.



# Project Worksheet - Answers

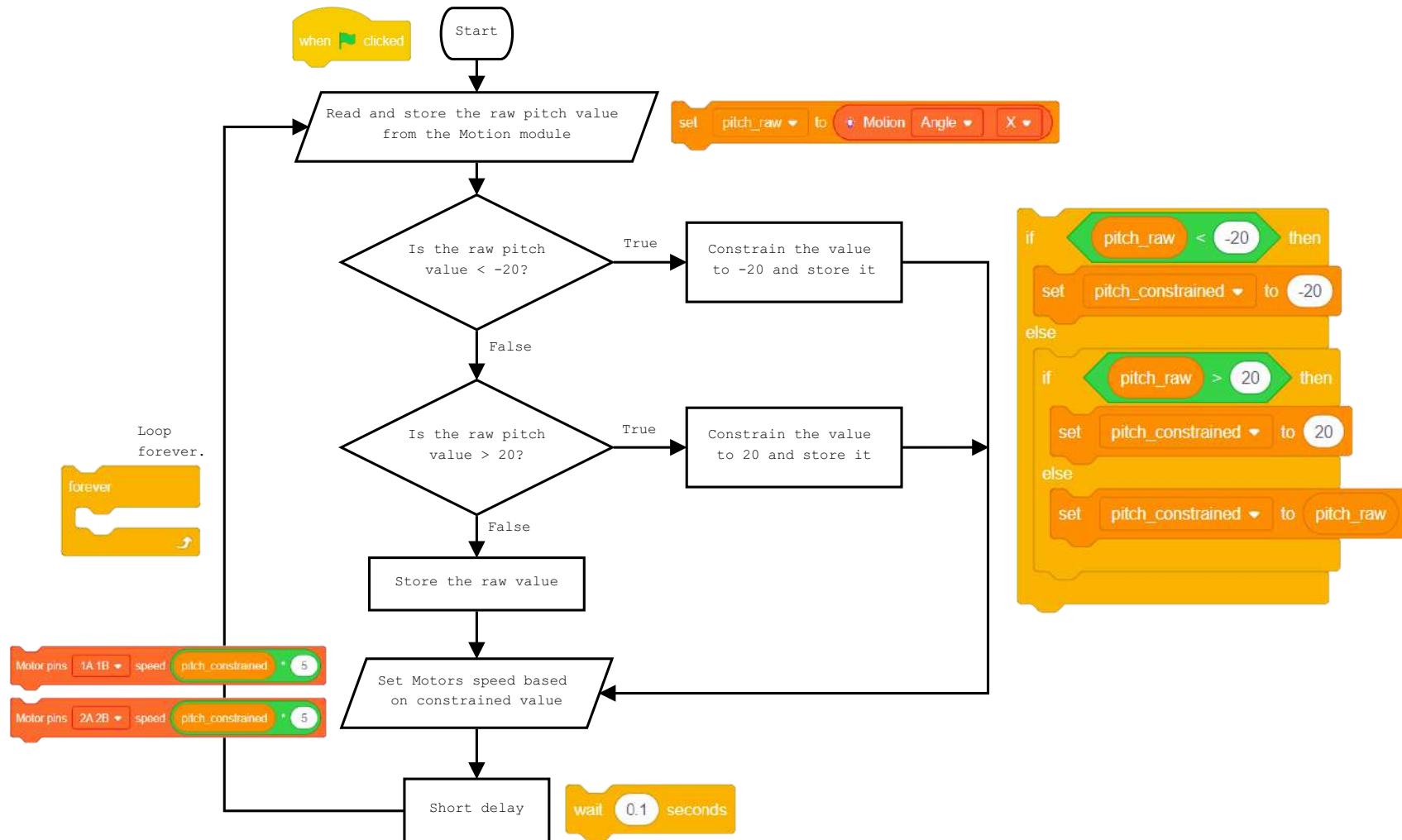
5. How do you assemble and connect the modules for the solution?



1. Stack the Motor Controller onto mCenter+.
2. Connect a Motor to connector “1A1B” on the Motor Controller.
3. Connect a Motor to connector “2A2B” on the Motor Controller.
4. Connect the Motion module to a “I2C” pin.
5. Insert one end of the USB cable into mCenter+ and connect the other to a computer.

# Project Worksheet - Answers

## 6. Configure blocks to relate to the flow chart.



# The Script

Click Green Flag to activate script.

Read and store the "X" angle from the Motion module.

Check if the pitch (raw) value is < -20.

Constrain it to -20.

Check if the pitch (raw) value is > 20.

Constrain it to 20.

If between -20 and 20.  
Then accept the raw value.

Set the Motor (1) speed based on the constrained "X" angle value.

Set the Motor (2) speed based on the constrained "X" angle value.

Short delay before looping.

Loop.

```
when green flag clicked
  forever
    set [pitch_raw v] to [Motion Angle X]
    if [pitch_raw < -20] then
      set [pitch_constrained v] to [-20]
    else
      if [pitch_raw > 20] then
        set [pitch_constrained v] to [20]
      else
        set [pitch_constrained v] to [pitch_raw]
    end
    motor pins 1A 1B speed [pitch_constrained * 5]
    motor pins 2A 2B speed [pitch_constrained * 5]
    wait [0.1 seconds]
```

## Block Locator

Events



Control



Operators



Data



mCookie

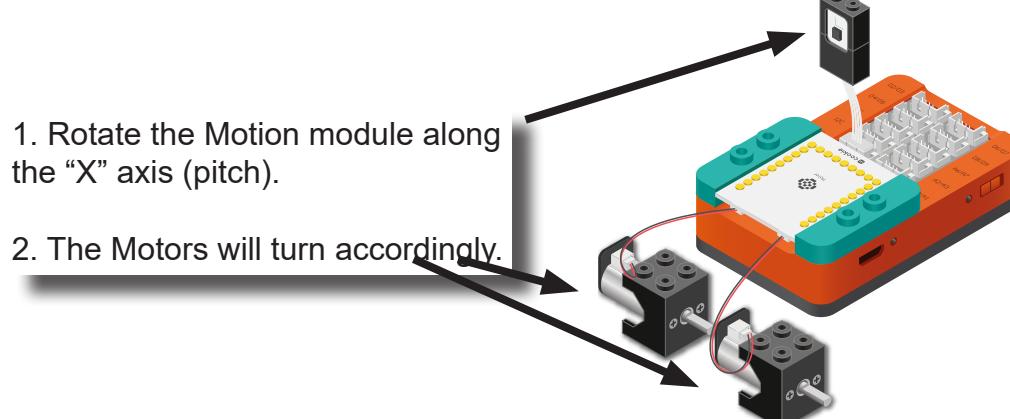
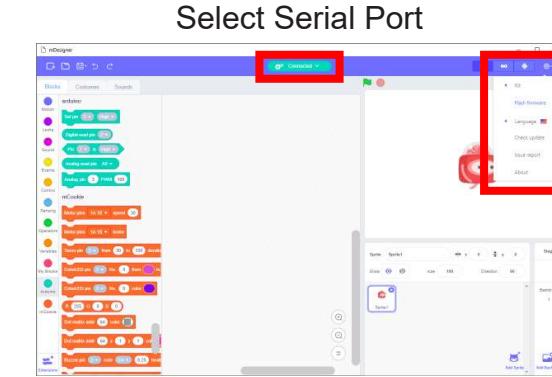
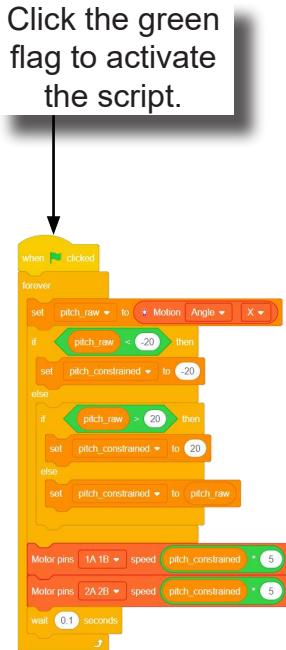


# Testing the Program

## Important:

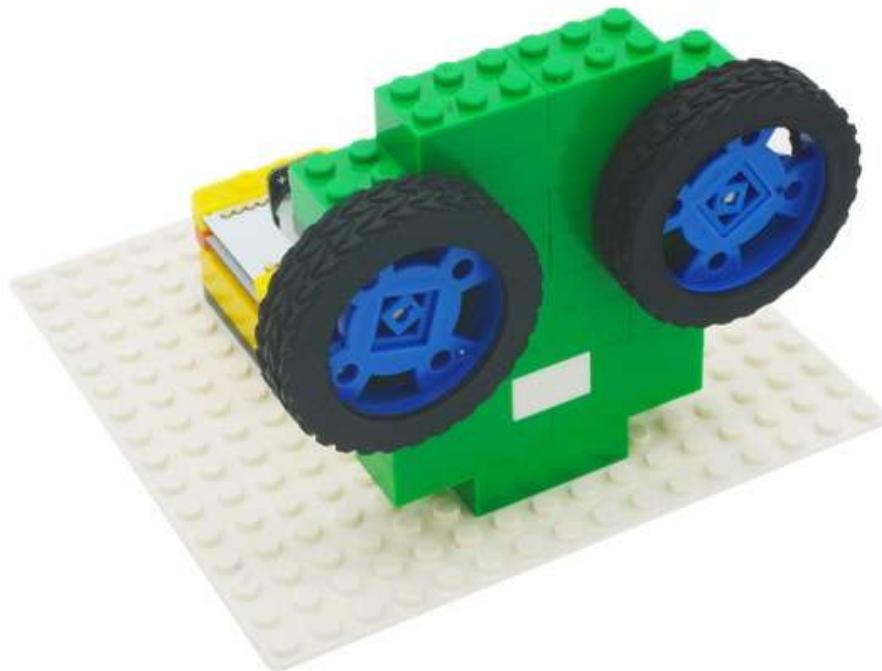
- Ensure that mCenter+ is on. Flip the switch on mCenter+ to turn it on. An LED on mCenter+ will light up to indicate power is being supplied.
- Ensure you have selected the Serial Port and flashed the special firmware (mCookie) first before testing (this only needs to be done once unless the firmware is overwritten).
- Ensure you have selected the Serial Port and connected before testing.

## Testing:



## Create a Structure

Create a structure for the project using building blocks, paper, or other crafting materials.



## Project Challenge

Increase the maximum speed in which the Motors can spin.

Hint: Change the multiplication factor when setting the Motor speeds.

Change the way the motors react to the pitch value to your liking.

(Example, eyes rotating in opposing directions.)

## Review



Motor Controller - controls and drives attached motors. Can control up to 2 motors.



Motor - converts electrical energy into mechanical energy in the form of torque.



Wheel - attaches to the Motor.



Motor - Set Speed block - set the speed of a motor.



Motion - module which detects motion such as roll, yaw and pitch.



Motion - Read block - returns various motion readings from the Motion module.

## Expanding the Project

Write down a way of expanding this project. Create it on your own.

---

---

---

---

---

---

---

---

---

---

---

## Research

Research other eye reflexes such as the “Optokinetic response”.

Why are these reflexes important and why do they happen without your direct control?

---

---

---

---

---

---

---

---

---

---

---



## WHAT YOU WILL NEED



MCENTER +



MOTION



SERVO



SERVO  
CONNECTOR



USB CABLE



SENSOR CABLE

MIX4A04

# Single Axis Steadicam

Use a servo to stabilize  
a platform.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



ASSEMBLY DIFFICULTY:

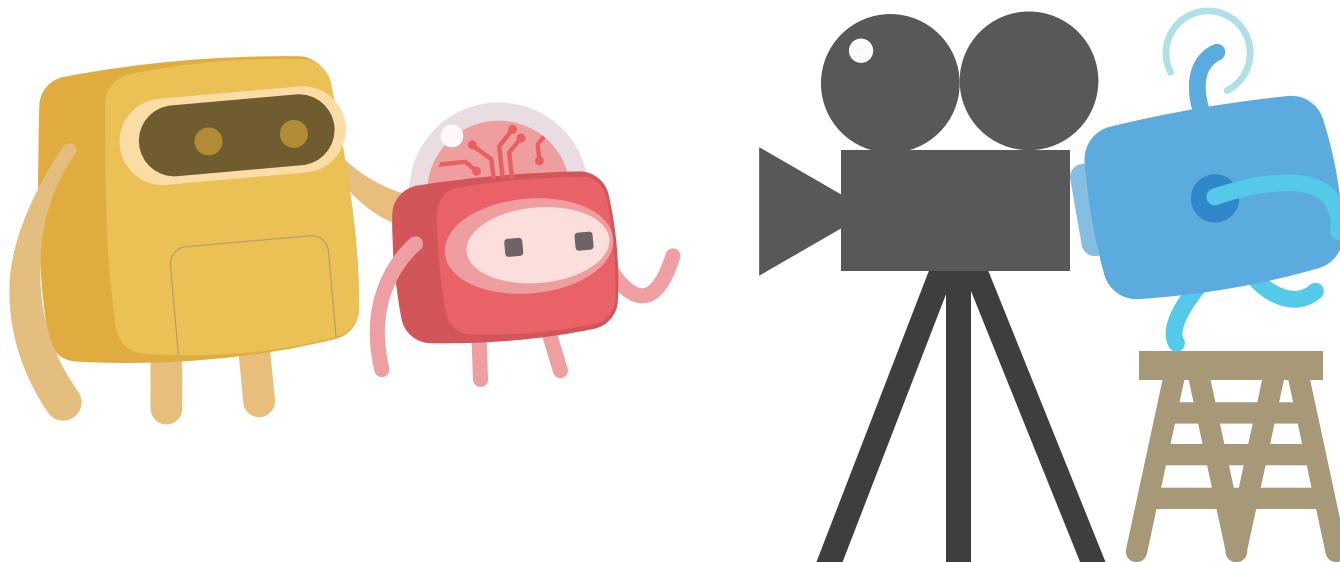


**MICRODUINO**

## About Camera Stabilizers

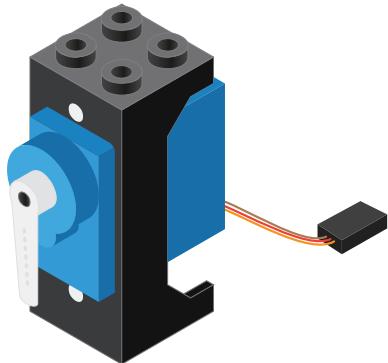
A camera stabilizer, or camera-stabilizing mount, is a device designed to hold a camera in a manner that prevents or compensates for unwanted camera movement, such as "camera shake".

For small hand-held cameras, a harness or contoured frame steadies the camera against the photographer's body. In some models, the camera mount is on an arm that protrudes in front of the photographer; beneath the camera is a handle grip. Another variation positions the camera atop a fulcrum braced against the photographer's chest or abdomen.



# Review of Modules

## Servo



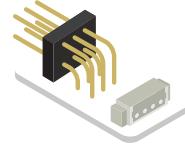
A Servo is an actuator which consist of a motor and other electronic components which allows for precise angle adjustments of the attached horn. The servo included in the kit has an adjustable angle range between 0 and 180 degrees.

## Assembly

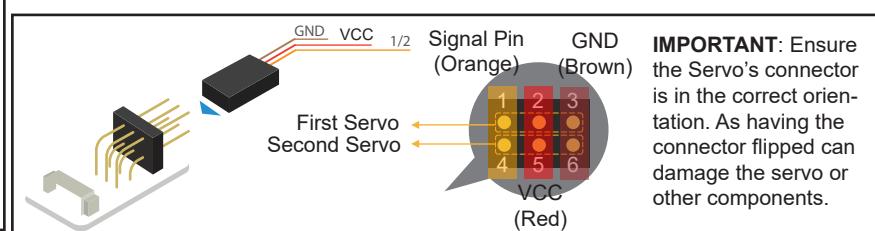
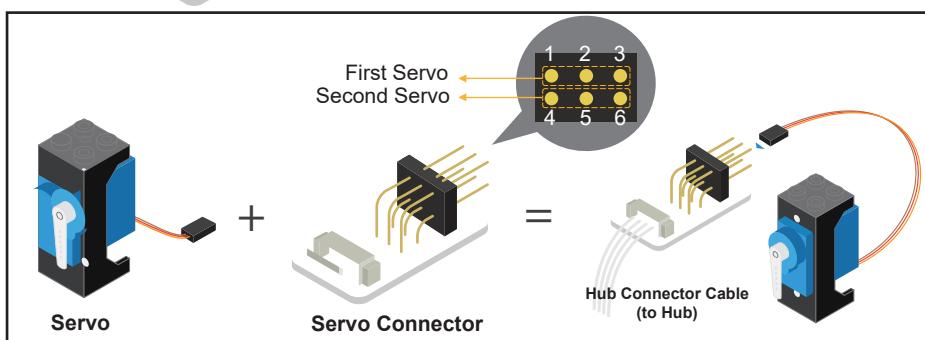
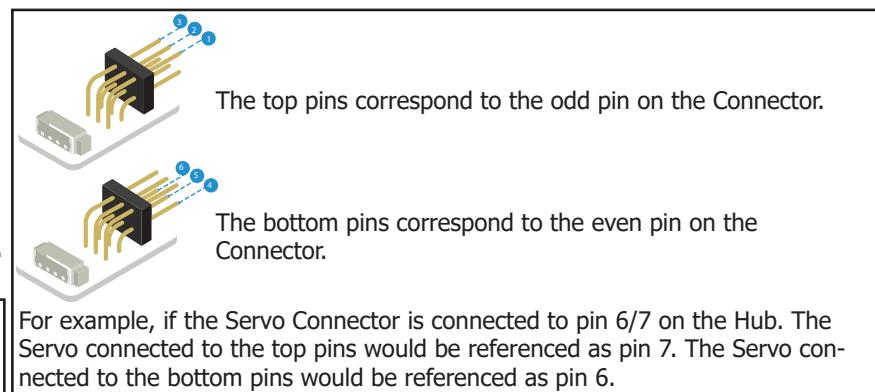
Install the Servo into the Servo Block Connector and fasten with the included screws.



## Connectivity

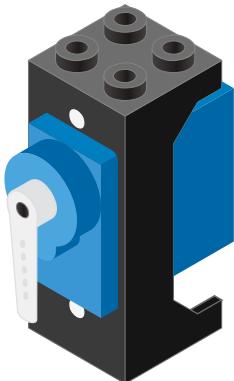


A Servo Connector is required in order to connect the Servo to the Hub. The Servo Connector can support up 2 Servos at a time.



# Review of Modules

## Servo (continued)



### Usage

The angle, between 0 and 180 degrees, is adjustable using blocks in Scratch. Calibrating your attachment (the horn) to your desired 0 angle can be done by first setting the servo to 0 degrees, then attaching the horn as desired.

#### Servo - Set Angle block



Pin the Servo is connected to.

Angle to set Servo to.

Sets the horn angle of the Servo. Quickly moves (as fast as possible) to the desired angle.

#### Servo - Move From-To Angle block



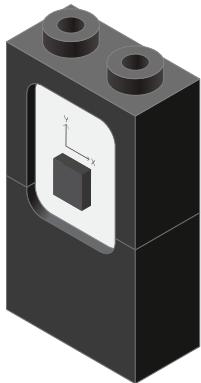
Pin the Servo is connected to.

Starting angle of Servo.

Sets the horn angle from the starting angle to the target angle. This movement is performed in the specified duration time.

# Review of Modules

## Motion



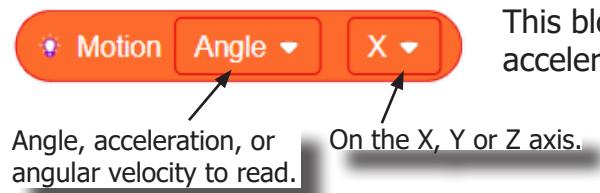
The Motion sensor is a triple axis accelerometer and gyroscope.

It is capable of measuring on the X, Y & Z axis the:

- Angle
- Acceleration
- Angular velocity

It uses the "I2C" communication protocol and must be connected to an "I2C" port on mCenter+.

### Motion - Read block



This block reads data from the Motion module. It is capable of retrieving the angle, acceleration and angular velocity on the X, Y or Z axis.

Angle, acceleration, or  
angular velocity to read.

On the X, Y or Z axis.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

## Greater Than Block



This block returns TRUE if the left input is greater than the right input.

## The Problem

How can we create a one-axis camera stabilizer?

## Project Worksheet

Complete the worksheet below to your best abilities.

1. A solution.

Use the Motion module to detect the current “Y” angle. Stabilize the project using a Servo compensating for the “Y” angle value.

2. List all the parts needed for the solution. What is the purpose of each part?

3. What is a simple explanation of the logic for the solution?

4. Create a flow chart of the solution.

5. How do you assemble and connect the modules for the solution?

6. Configure blocks to relate to the flow chart.

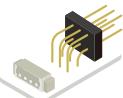
## Project Worksheet - Answers

2. List all the parts needed for the solution. What is the purpose of each part?

1x  mCenter+ - control the project, connect sensors and trinkets to project, and supply power.

1x  USB Cable - connect mCenter+ to a computer for charging and programming the project.

2x  Hub Connector Cable - connect sensors and trinkets to the mCenter+.

1x  Servo Connector - connect a Servo to mCenter+.

1x  Servo - to steady the camera.

1x  Motion - detect the un-stabilized angle.

## Project Worksheet - Answers

3. What is a simple explanation of the logic for the solution?

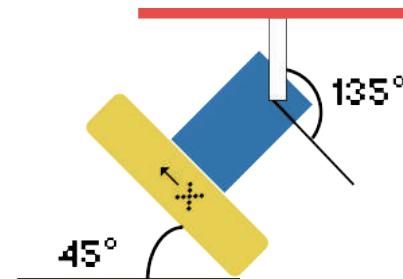
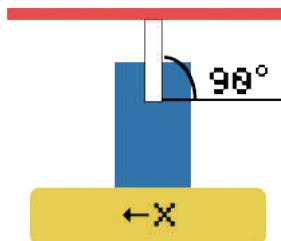
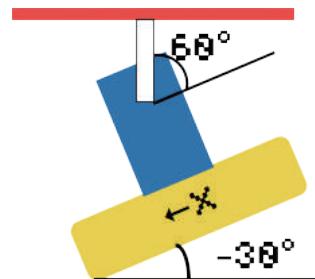
Use a Motion module to detect the "Y" angle. The Motion module must lay horizontal.

Use a Servo to compensate for the "Y" angle for stabilization.

Read "Y" angle.

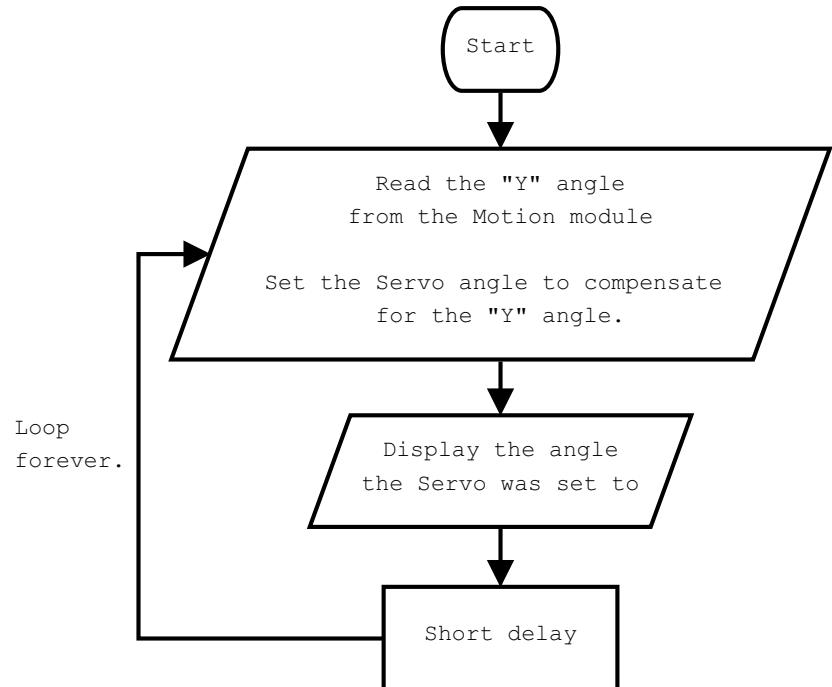
Set the Servo to compensate for the "Y" angle by adding 90.

Display the angle the Servo is set to.



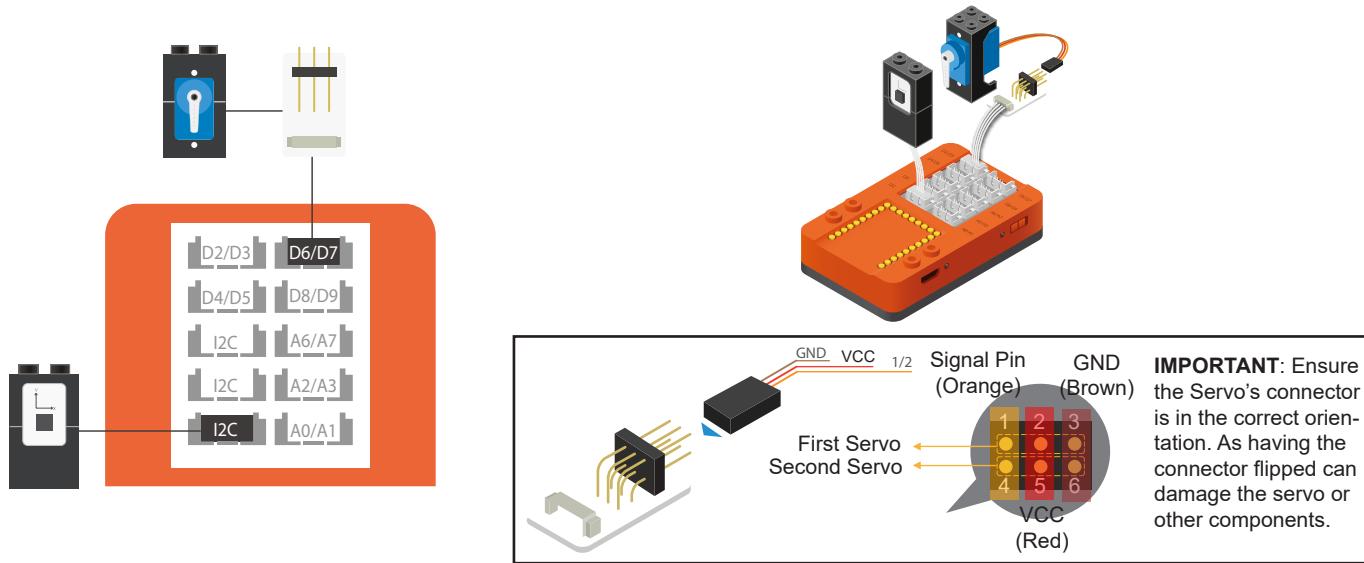
# Project Worksheet - Answers

4. Create a flow chart of the solution.



# Project Worksheet - Answers

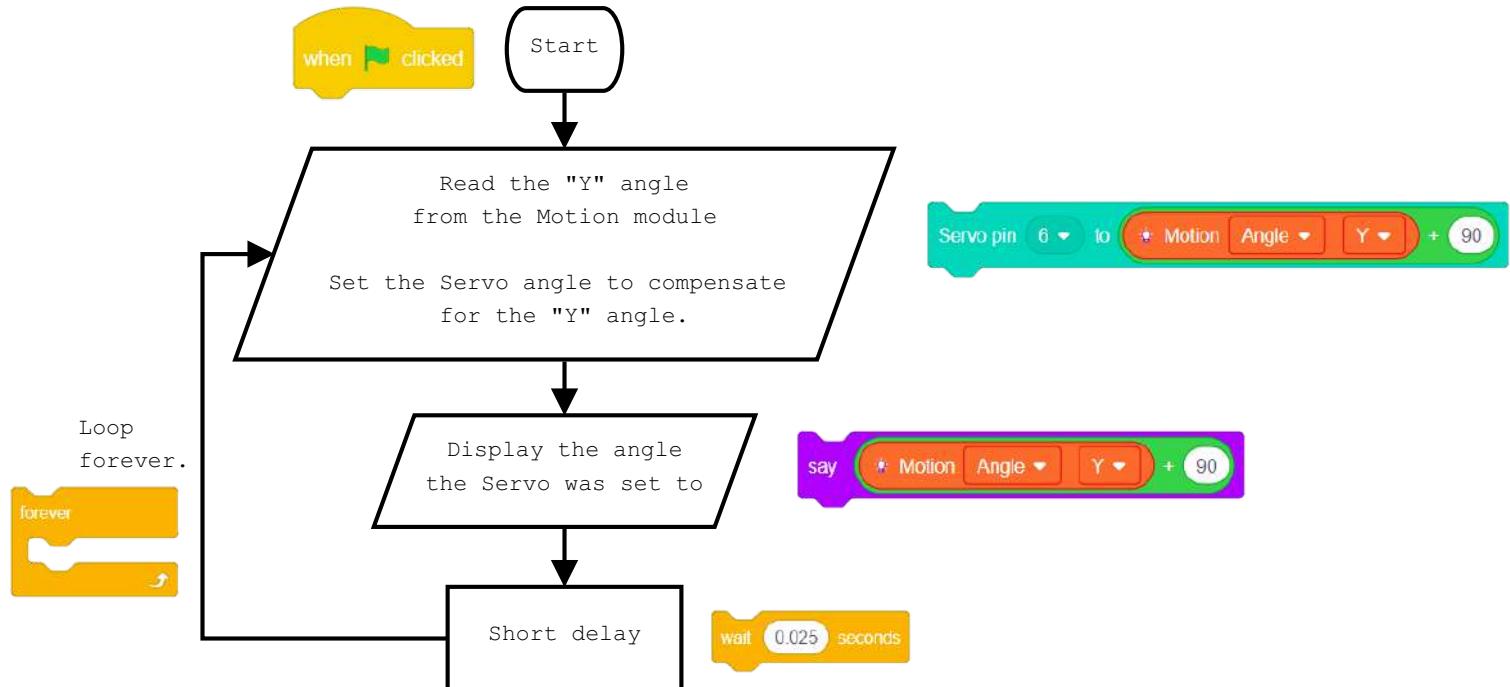
5. How do you assemble and connect the modules for the solution?



1. Connect a Servo Connector to pin 6/7 using a Hub Connector Cable.
2. Connect two Servos to the Servo Connector. Ensure the connector is in the correct orientation.
3. Connect the Motion module to a "I2C" pin.
4. Insert one end of the USB cable into mCenter+ and connect the other to a computer.

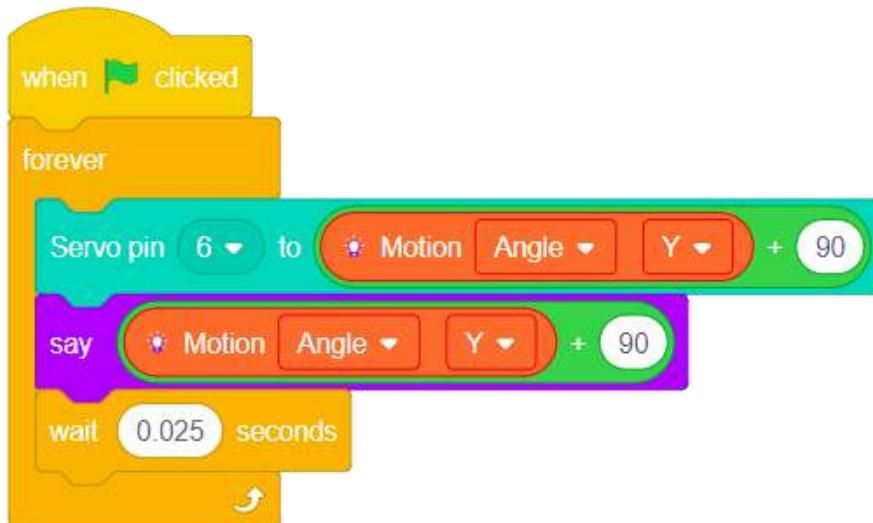
# Project Worksheet - Answers

6. Configure blocks to relate to the flow chart.



# The Script

Click Green Flag to activate script.



Read "Y" angle and compensate with Servo (pin 6).

Display the Servo angle on the stage.

Short delay before looping.

Loop.

## Block Locator

Looks	say Hello!
Events	when green flag clicked
Control	forever wait 1 secs
Operators	+ -
Arduino	Servo pin 2 to 90 ***
mCookie	Motion Angle X

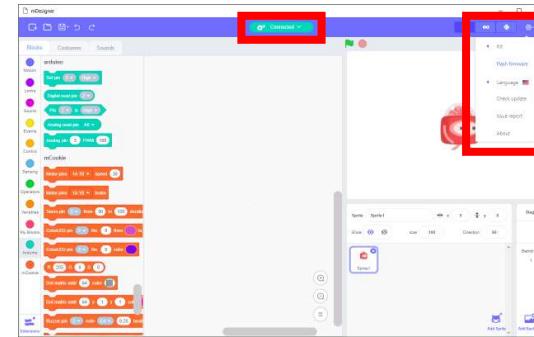
\*\*\* Enable this block under settings (gear icon on top right). Then select Kit > mCookie Extension.

# Testing the Program

## Important:

- Ensure that mCenter+ is on. Flip the switch on mCenter+ to turn it on. An LED on mCenter+ will light up to indicate power is being supplied.
- Ensure you have selected the Serial Port and flashed the special firmware (mCookie) first before testing (this only needs to be done once unless the firmware is overwritten).
- Ensure you have selected the Serial Port and connected before testing.

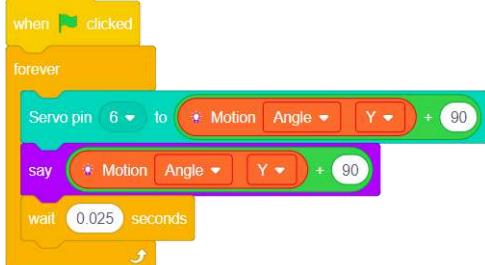
## Select Serial Port



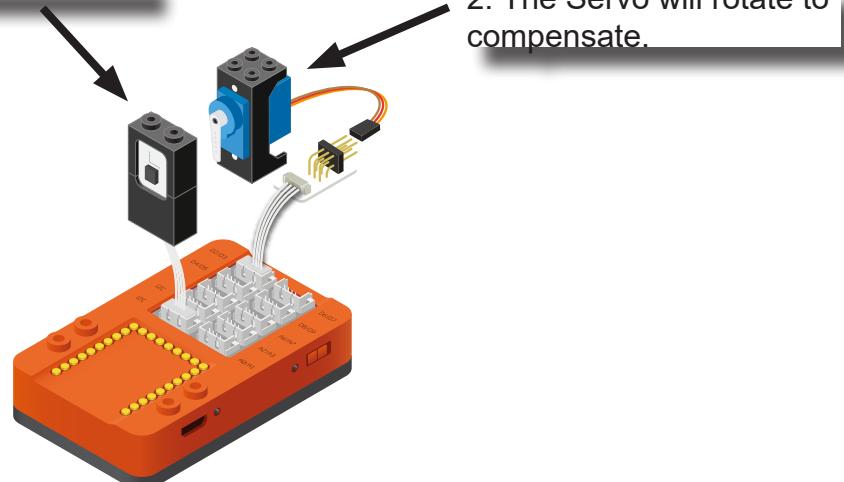
Flash firmware

## Testing:

Click the green flag to activate the script.



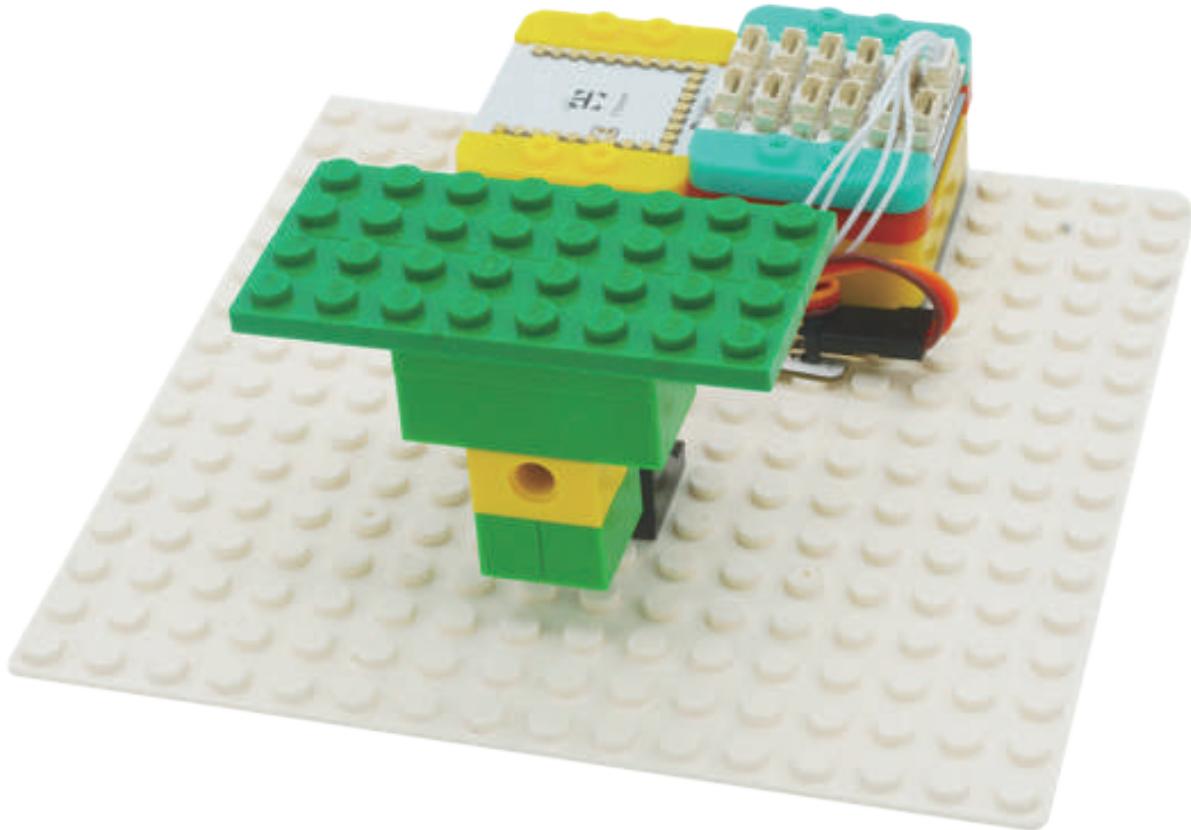
1. Rotate the Motion module.



**Note:** If the program is working correctly and you wish to use the program without being connected to your computer. Switch to Arduino mode (top right), then "Flash Program" (this overrides the special firmware, you will need to reflash to use Live mode).

## Create a Structure

Create a structure for the project using building blocks, paper, or other crafting materials.



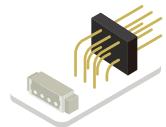
## Project Challenge

Add a Color LED to the project.  
Use it to indicate when the camera stabilizer is at an extreme angle.

## Review



Servo - an actuator with precise angle control.



Servo Connector - an adapter to connect a Servo to the Hub on mCenter+.



Servo - Set Angle block - Sets the angle of the Servo. Quickly moves (as fast as possible) to the desired angle.



Motion - module which detects motion such as roll, paw and yaw.



Motion - Read block - returns various motion readings from the Motion module.

## Expanding the Project

Write down a way of expanding this project. Create it on your own.

---

---

---

---

---

---

---

---

---

---

---

## Research

Research other things self-stabilize.

What are some very important life saving applications of this principle?

---

---

---

---

---

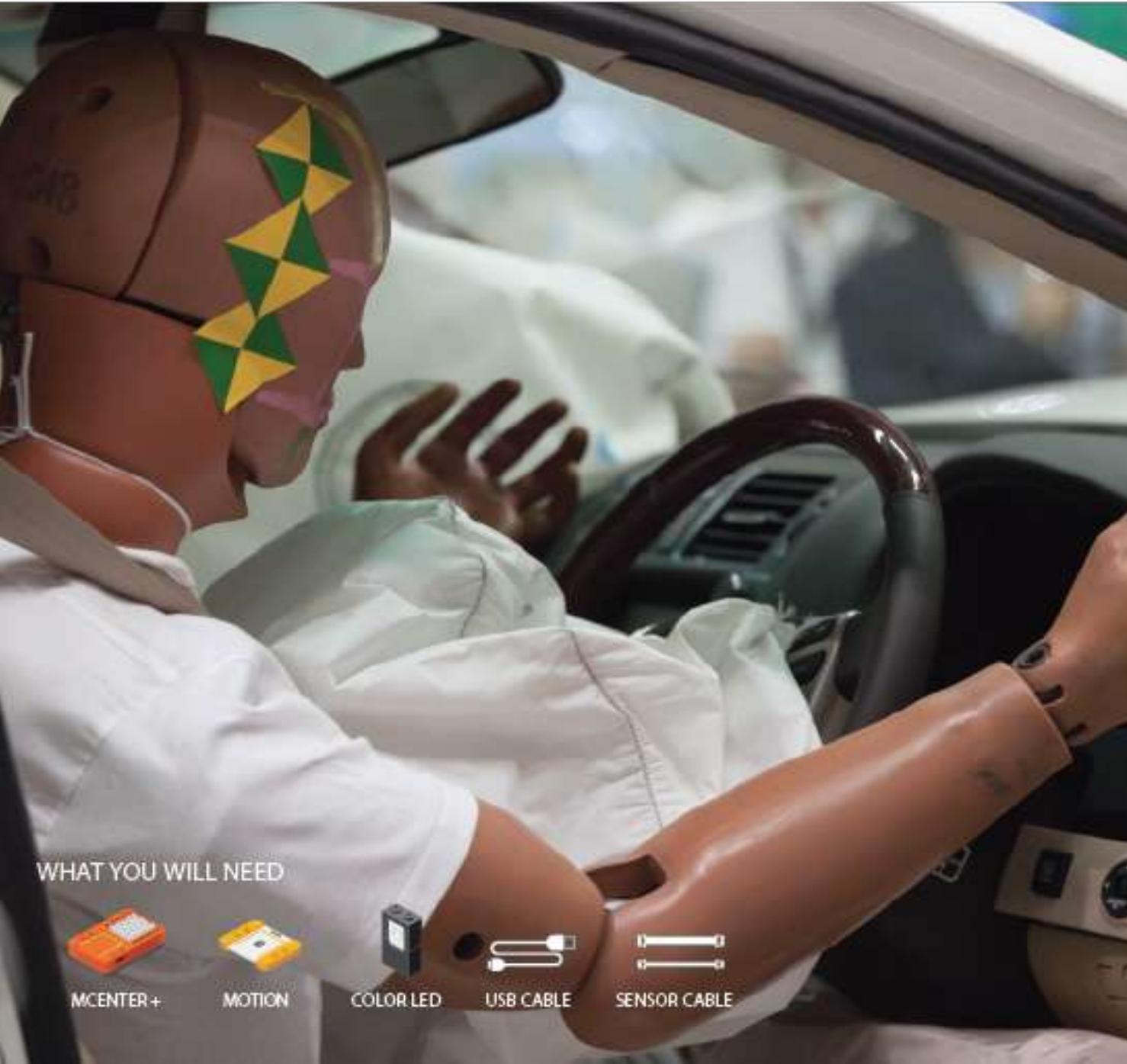
---

---

---

---

---



#### WHAT YOU WILL NEED



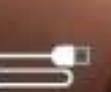
MCENTER+



MOTION



COLOR LED



USB CABLE



SENSOR CABLE

## MIX4A06 Airbag

Create an airbag which triggers when a large acceleration is sensed.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



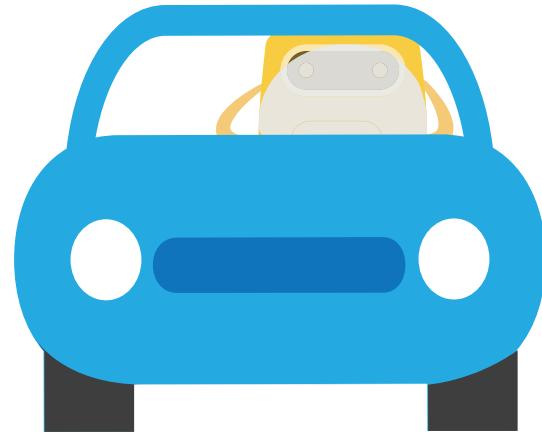
ASSEMBLY DIFFICULTY:



**MICRODUINO**

## About Airbags

An airbag is a type of vehicle safety device and is an occupant restraint system. The airbag module is designed to inflate extremely rapidly, then quickly deflate during a collision or impact with a surface or a rapid sudden deceleration. It consists of the airbag cushion, a flexible fabric bag, inflation module and impact sensor. The purpose of the airbag is to provide the occupants a soft cushioning and restraint during a crash event to prevent or reduce any impact or impact-caused injuries between the flailing occupant and the interior of the vehicle. The airbag also provides an energy absorbing surface between the vehicle's occupant and a steering wheel, instrumental panel, A-B-C- structural body frame pillars, headliner and windshield/windscreen.



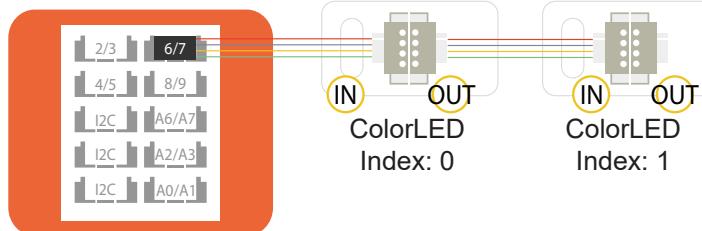
# Review of Modules

## Color LED



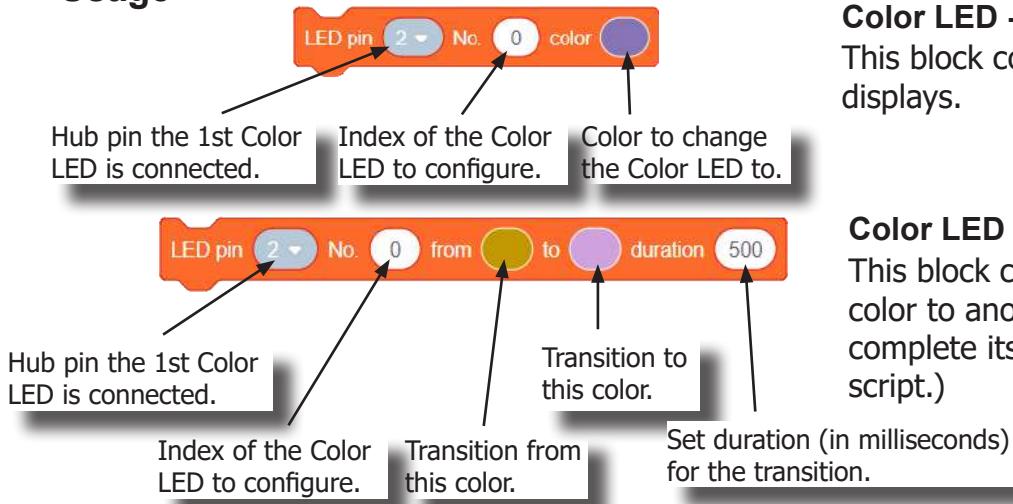
The Color LED is a LED composed of 3 smaller LEDs which emit red, green, and blue. Each color's brightness value can be configured individually. This can produce a wide variety of colors. It is controlled over a special communication protocol which is different from the basic "Digital Output" (used by Single-Color LED).

### Indexing



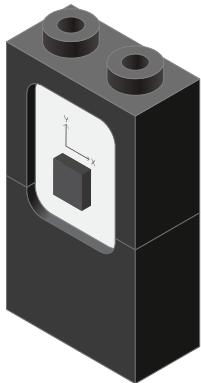
- The Color LED connected directly to the Hub is index 0.
- The Color LED connected next in the chain is index 1.
- And so on.

### Usage



# Review of Modules

## Motion



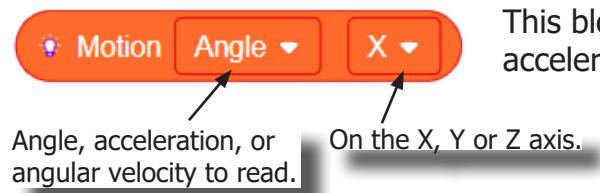
The Motion sensor is a triple axis accelerometer and gyroscope.

It is capable of measuring on the X, Y & Z axis the:

- Angle
- Acceleration
- Angular velocity

It uses the "I2C" communication protocol and must be connected to an "I2C" port on mCenter+.

### Motion - Read block



This block reads data from the Motion module. It is capable of retrieving the angle, acceleration and angular velocity on the X, Y or Z axis.

Angle, acceleration, or  
angular velocity to read.

On the X, Y or Z axis.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

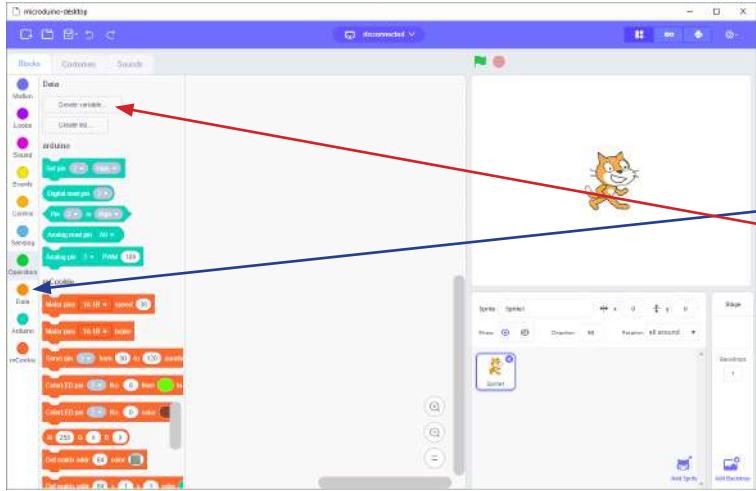
## Greater Than Block



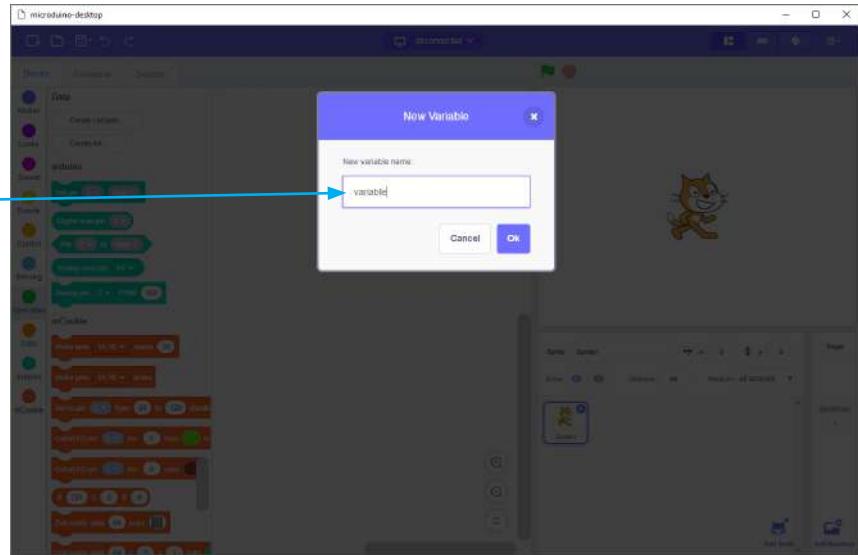
This block returns TRUE if the left input is greater than the right input.

# Review of Concepts

## Creating a Variable in Scratch



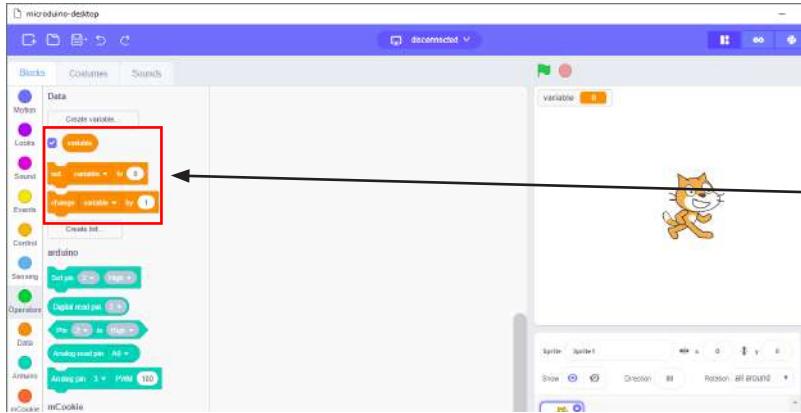
Click on the "Variable" category.  
Then click on the "Create variable..." button.



Name your variable.

# Review of Concepts

## Creating a Variable in Scratch (continued)



Several new blocks will appear which allows access and modification of the new variable.

### Variable Value block

**variable** This block returns the current value of the variable.

### Variable Set block

**set [variable] to [value]** This block sets the value of the variable. It will overwrite the current value.

### Variable Change block

**change [variable] by [value]** This block changes the variable value by adding the input value (negative values or blocks are accepted).  
input value

## The Problem

How can we detect large deceleration forces and indicated when detected?

# Project Worksheet

Complete the worksheet below to your best abilities.

1. A solution.

Use the Motion module to detect a large deceleration. Use a Color LED to indicate a large deceleration has occurred.

2. List all the parts needed for the solution. What is the purpose of each part?

3. What is a simple explanation of the logic for the solution?

4. Create a flow chart of the solution.

5. How do you assemble and connect the modules for the solution?

6. Configure blocks to relate to the flow chart.

## Project Worksheet - Answers

2. List all the parts needed for the solution. What is the purpose of each part?

1x  mCenter+ - control the project, connect sensors and trinkets to project, and supply power.

1x  USB Cable - connect mCenter+ to a computer for charging and programming the project.

2x  Hub Connector Cable - connect sensors and trinkets to the mCenter+.

1x  Motion - to detect large acceleration forces.

1x  Color LED - indicate when a large acceleration is detected.

## Project Worksheet - Answers

3. What is a simple explanation of the logic for the solution?

Large deceleration occurs when a sudden stop happens, such as when avoiding a car accident.

Use the Motion module to detect a large negative deceleration (sudden stop).

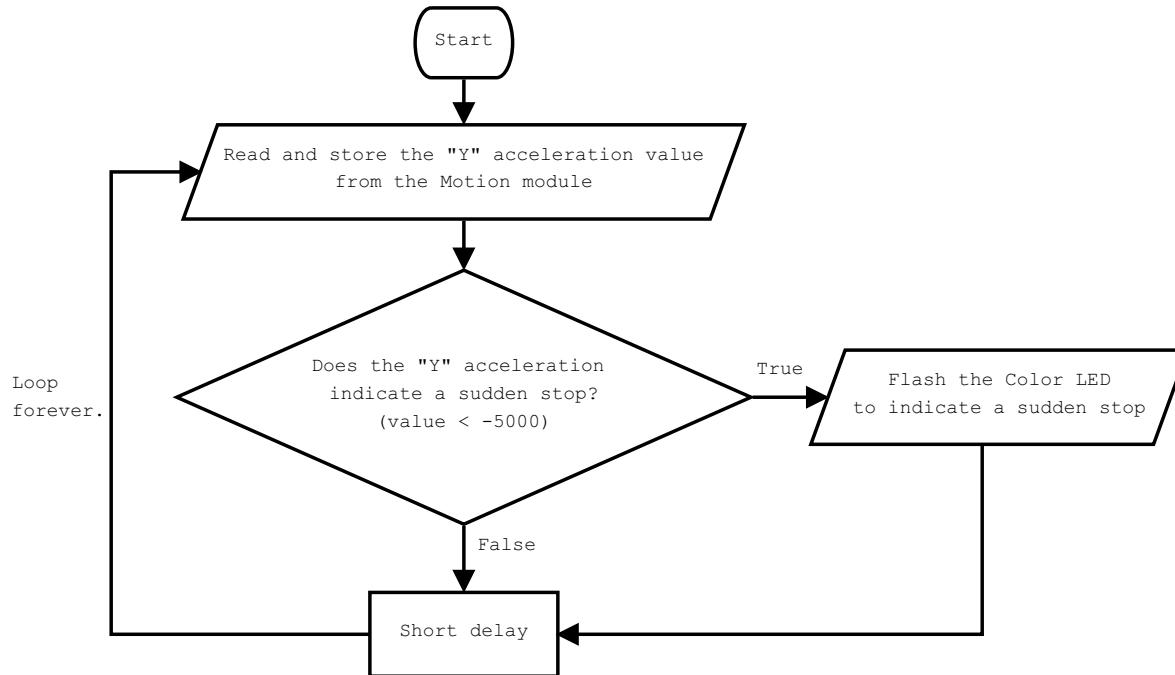
Use the Color LED to indicate when such force has been detected.

Read the "Y" acceleration. Check if it is a large negative value, which occurs during a sudden stop.

If there is a large value, then flash the Color LED to indicate it has occurred.

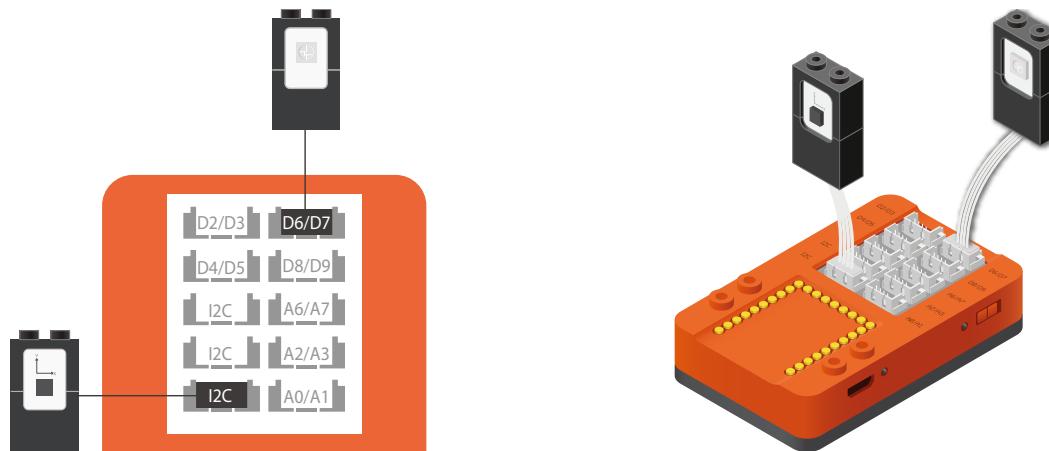
# Project Worksheet - Answers

4. Create a flow chart of the solution.



## Project Worksheet - Answers

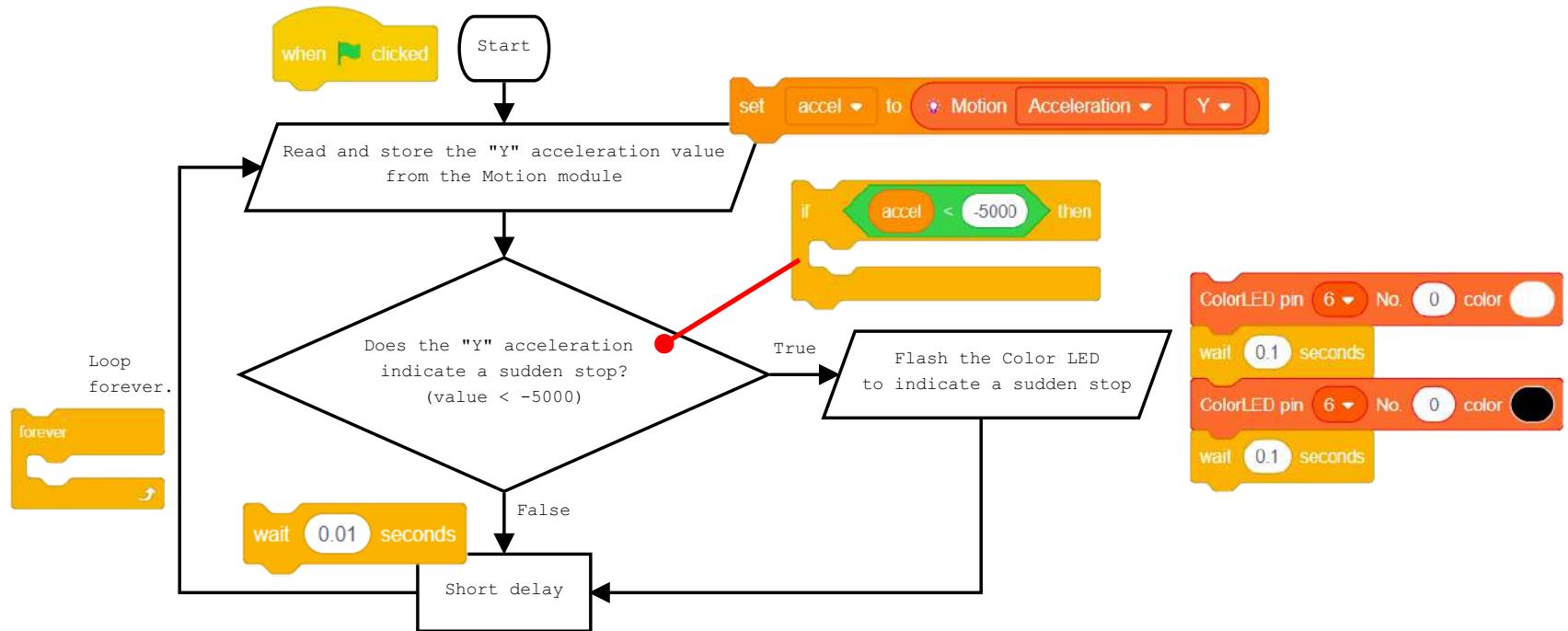
5. How do you assemble and connect the modules for the solution?



1. Connect the Motion module to a "I2C" pin.
2. Connect a Color LED (IN port) to pin 6/7.
3. Insert one end of the USB cable into mCenter+ and connect the other to a computer.

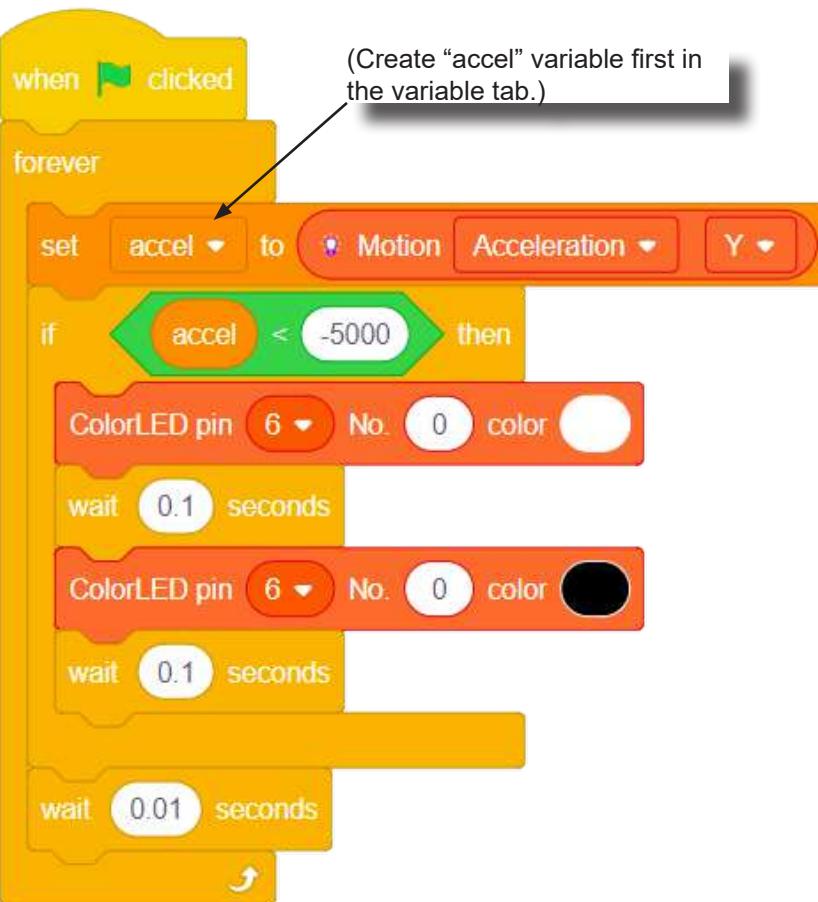
# Project Worksheet - Answers

6. Configure blocks to relate to the flow chart.



# The Script

Click Green Flag to activate script.



Read and store the "Y" acceleration from the Motion module.

Check if there is a large negative acceleration (sudden stop).

If there is, flash Color LED (pin 6).

Short delay before looping.

Loop.

## Block Locator

Events

Control

Operators

Data

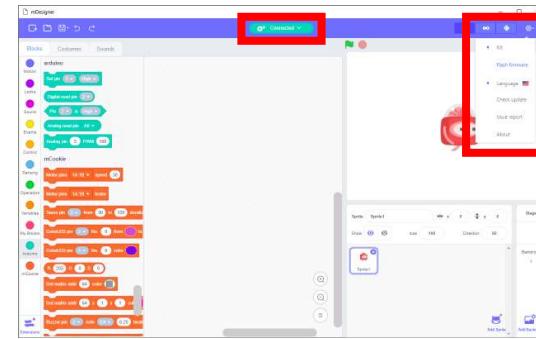
mCookie

# Testing the Program

## Important:

- Ensure that mCenter+ is on. Flip the switch on mCenter+ to turn it on. An LED on mCenter+ will light up to indicate power is being supplied.
- Ensure you have selected the Serial Port and flashed the special firmware (mCookie) first before testing (this only needs to be done once unless the firmware is overwritten).
- Ensure you have selected the Serial Port and connected before testing.

## Select Serial Port



Flash firmware

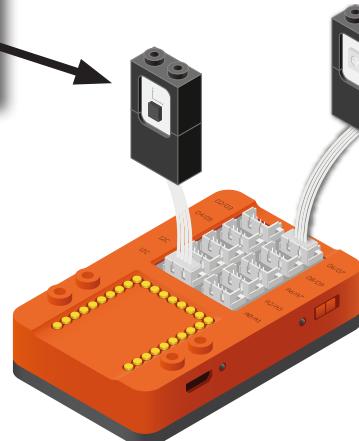
## Testing:

Click the green flag to activate the script.



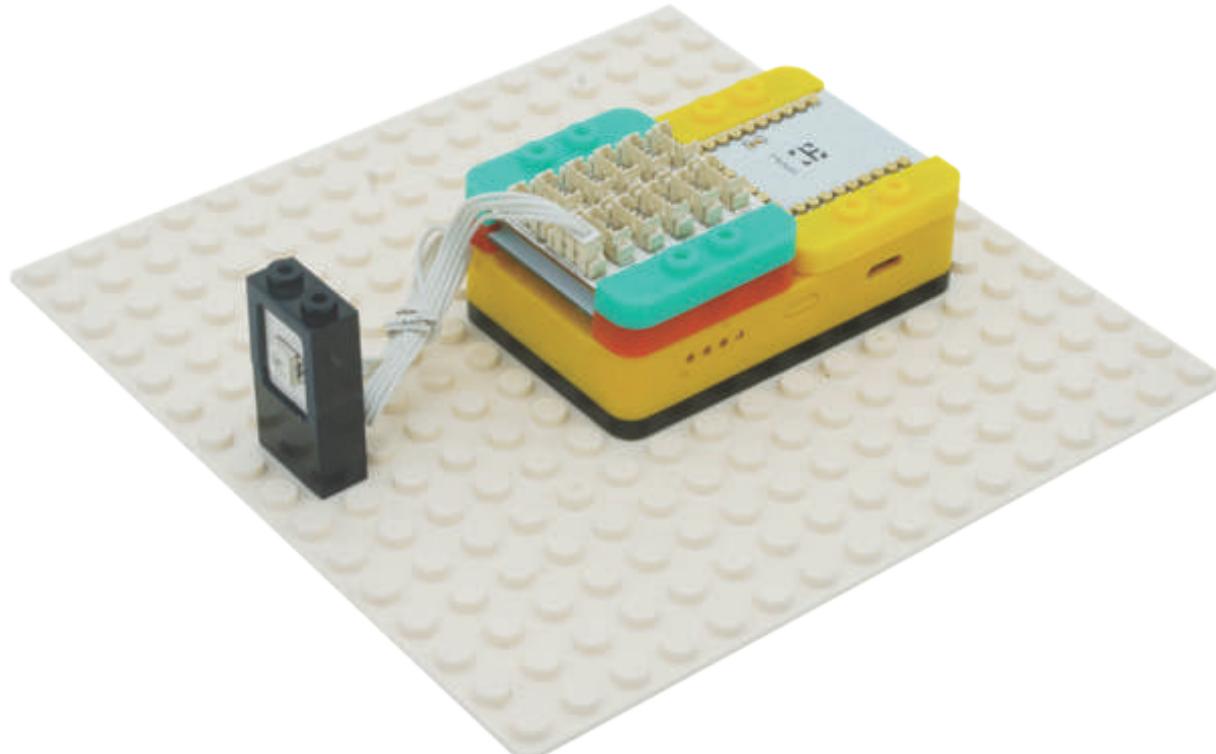
1. Create a large negative acceleration force (sudden stop).

2. When a large negative acceleration occurs, the Color LED will flash.



## Create a Structure

Create a structure for the project using building blocks, paper, or other crafting materials.



## Project Challenge

Create your own lighting effects when a sudden stop occurs.

How would you detect a large forward acceleration?

Modify the script to indicate when a large forward acceleration occurs.

## Review



Motion - module which detects motion such as roll, paw and yaw.



Motion - Read block - returns various motion readings from the Motion module.



Color LED - configurable to display a wide range of colors.



Color LED - Color Select block - configures what color a particular Color LED displays.

## Expanding the Project

Write down a way of expanding this project. Create it on your own.

---

---

---

---

---

---

---

---

---

---

## Research

Research more about the history of airbags.

Besides being in cars, what other places are airbags used?

---

---

---

---

---

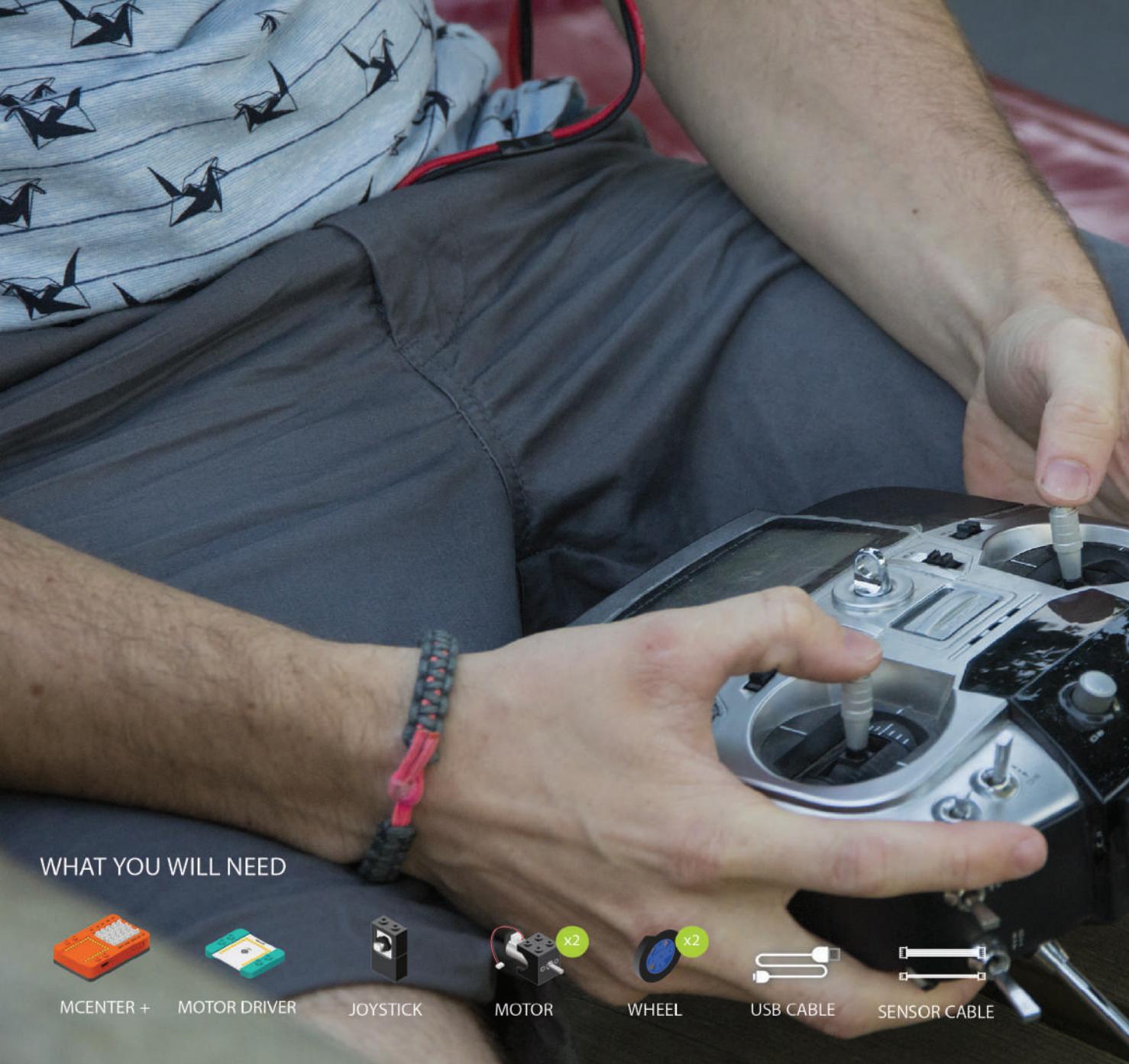
---

---

---

---

---



## WHAT YOU WILL NEED



MCENTER +



MOTOR DRIVER



JOYSTICK



MOTOR x2



WHEEL x2



USB CABLE



SENSOR CABLE

## MIX4B06 Wired RC Car

Use a joystick to control the car forward, backward, left and right.

PROJECT TIME: 90 minutes

PROJECT DIFFICULTY:



ASSEMBLY DIFFICULTY:



**MICRODUINO**

## About RC Cars

Radio controlled (or R/C) cars are battery/gas-powered model cars or trucks that can be controlled from a distance using a specialized transmitter or remote. The term "R/C" has been used to mean both "remote controlled" and "radio controlled", where "remote controlled" includes vehicles that are connected to their controller by a wire, but common use of "R/C" today usually refers to vehicles controlled by a radio-frequency link. This section focuses on radio-controlled vehicles only.



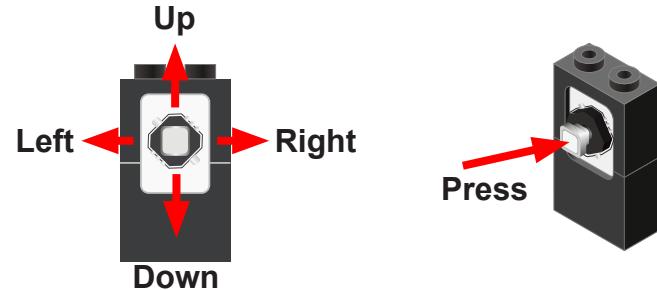
# Review of Modules

## Joystick



A Joystick is an input module (sensor) which can sense if the stick is in the **up, down, left or right** positions.

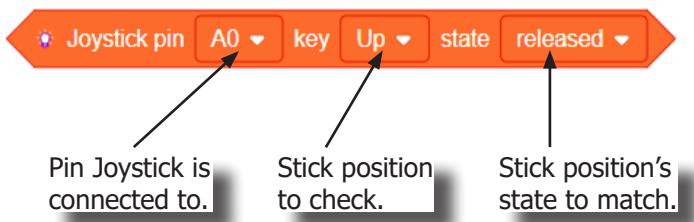
In addition, it can sense if it is being **pressed** (pressing directly on stick).



## Usage

The Joystick must be connected to an "Analog Read" compatible pin. These are the pins prefixed with an "A" (pins A0 to A7).

## Joystick State block

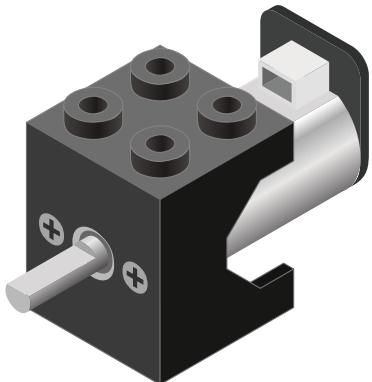


On the defined Joystick pin, if the stick's position (**up, down, left, right** or **center**) matches the state (released, pressed or pressing), then this block returns TRUE.

Otherwise, this block returns FALSE.

# Review of Modules

## Motor



**Motor - Set Speed block**



Motor to configure.

Speed to set.

A Motor is a electrical machine which converts electrical energy to mechanical energy. It is capable of rotating clockwise and counter clockwise. It is controlled by a Motor Controller. A Motor Controller can control up to 2 Motors. A Wheel can be attached to the Motor.



**Motor Controller**



**Wheel**

**Motor - Brake block**



Motor to configure.

This block sets the speed to 0 to stop the Motor.

# Review of Blocks

## Addition Block



This block returns the sum of the left input and right input.

## Subtraction Block



This block returns the value of subtracting the left input by the right input.

## Multiplication Block



This block returns the product of the left input and right input.

## Division Block



This block returns the value of dividing the left input by the right input.

## Less Than Block



This block returns TRUE if the left input is less than the right input.

## Equal Block



This block returns TRUE if the left input is equal to the right input.

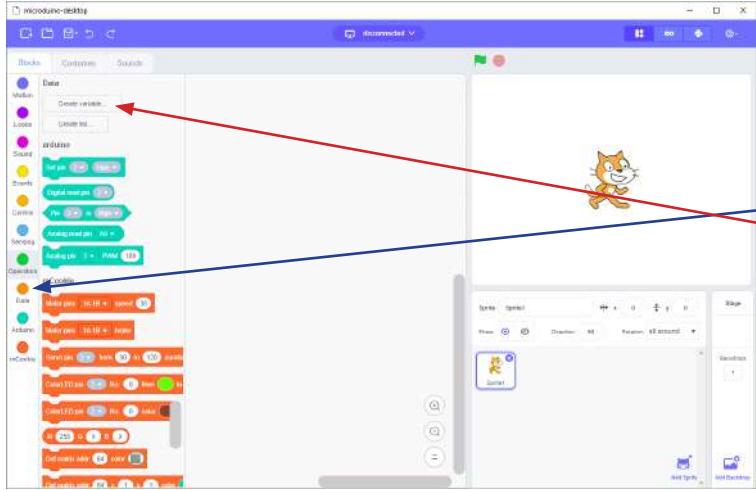
## Greater Than Block



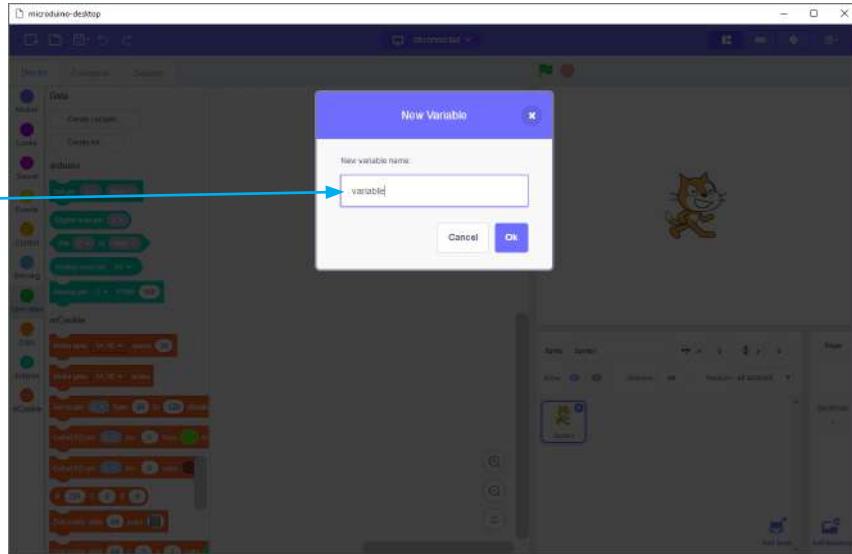
This block returns TRUE if the left input is greater than the right input.

# Review of Concepts

## Creating a Variable in Scratch

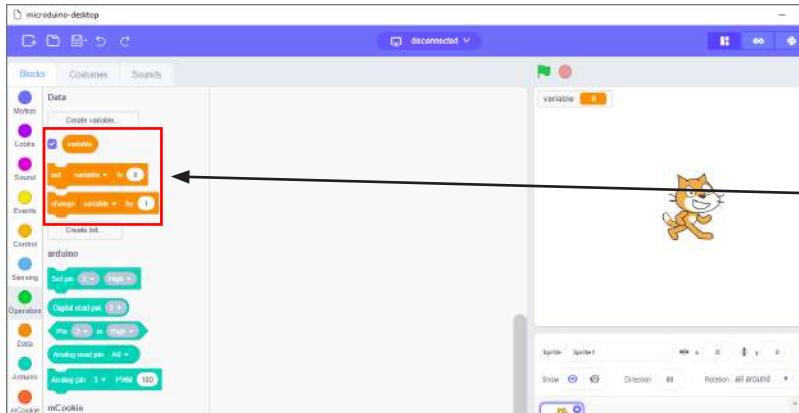


Click on the “Variable” category.  
Then click on the “Create variable...” button.



# Review of Concepts

## Creating a Variable in Scratch (continued)



Several new blocks will appear which allows access and modification of the new variable.

### Variable Value block

**variable** This block returns the current value of the variable.

### Variable Set block

**set [variable v] to [0]** This block sets the value of the variable. It will overwrite the current value.

### Variable Change block

**change [variable v] by [1]** This block changes the variable value by adding the input value (negative values or blocks are accepted).  
input value

## The Problem

How can we create a RC car controllable with a joystick module?