Maker Mobile Introduction to Arduino

# Summary

In this interactive kit, you will get a chance to learn about micro‐controllers by programming an Arduino compatible board and controlling its various inputs and outputs (LEDs, RGB LEDs, sensors, and potentiometers etc.). In this day and age, open source micro‐controllers like the Arduino have become mainstream and have been adopted by electronic hobbyists around the world. This kit gives participants a good introduction to the world of electronics and electrical engineering.

# Materials

* Arduino compatible board
* Maker Mobile board
* 11 wires
* 1 usb cable
* Computer with arduino installed.

## Introduction

### What is a micro controller?

It is like the brain of the computer. It is used to interpret and execute the instructions we gave it. It takes the inputs and gives desired outputs which will depend on the program we gave it. You take code and you get it to do stuff.

### What is programing?

It is giving a set of instructions to a computer or a microcontroller to get a desired result. Programming is used anytime we want to interact with a computer. When programing it is important that:

* Instructions are precise
* Instructions are in the right order
* Instructions are in a language it understands

We will be using an Arduino. It is a very simple microcontroller. It can only process electrical signals. So when we give it code, that code gets converted into electrical signals.

## Quick tour of the maker mobile Board

This is the components board we use during the workshop. There are 6 main parts to the Maker Mobile board:

* Strip of 10 LEDs
* 5V input
* GND (they are all connected together)
* RGB Led
* Photo sensor
* Potentiometer (trimmer)

## Setting up Arduino

Download the Arduino software from: <https://www.arduino.cc/en/Main/Software> and follow the installation instructions.

The software is known as IDE, which means Integrated Development Environment. Think of it as a text editor that can process the code.

Sometimes, the Arduino compatible boards that we are using require some extra set up. If that is the case visit <https://uottawa-makerspace.myshopify.com/products/a001> or email makerstore@uottawa.ca 

## Activity

There are several challenges and they are listed in increasing difficulty. The goal is to start easy and work your way to the harder challenges.

### Blink

Open the blink example in the Arduino IDE by clicking File -> Examples -> 01. Basics -> Blink

The code is divided into two main sections Setup and Loop. These are functions and they are between curly brackets. Think of the brackets like slices of bread on a sandwich. If you don’t have one on either side, then it won’t work.



#### What does setup mean?

Get things ready!

In the setup section there is the line pinMode(13, OUTPUT);

Think of this line like calling out someone's name to get their attention. Here we are saying: “Hey, Pin 13, you are going to be an output”

#### What does loop mean?

A loop is something that repeats. The function will repeat an infinite number of times because we never tell it when to stop.

In the loop function, we have the line: digitalWrite(13, HIGH);

Digital is like a light switch. You can only be on or off. This is different from an analog switch which is like a light switch with a dimmer. There are many different values between on and off. So when we type in digitalwrite(13, High) we are saying to the arduino, please turn on pin 13.

The next line is: delay(1000);

This means wait for a certain amount of time. That amount is expressed as a number of milliseconds in the parenthesis This means that we are waiting for 1 second.

The next line is digitalWrite(13, LOW);

We are telling pin 13 to turn off.

## Wire it

Wire pin 13 on the arduino compatible board to an LED and then GND on the arduino to GND on the makermobile board.

## Adding the code

Make sure that the board is set to arduino uno and that a COM port is selected. Then hit upload. If it worked it should say done uploading. And light should start blinking

## Add another light

Now let’s add a second light.

In the setup section, copy and paste the pinMode line. Change the number to another pin between 0 ‐ 12. I will pick 12.



What are we doing here? We are letting pin 12 know that it will be an output

In the loop section, copy and paste both the digitalWrite lines and paste them just below. Change the pin number to the same number you put in setup.

Add a new wire that goes from the pin to a second light. Hit upload.

## Challenges

### Challenge 1 and 2

Now that they know how to turn lights on and turn lights off get them to solve the following challenges.

1. Turn on lights in order then turn them off in opposite order
2. Turn on the lights from the outside inwards and turn them off from the inside outwards.

Pass out the challenges page so that way the students can refer to it during the workshop.

SOLUTION

Challenge 1:

 There are two ways to go about solving this challenge. The first way is to copy and paste each line in the right order as shown bellow.

// the setup function runs once when you press reset or power the board

void setup() {

 // initialize digital pins as an output.

 pinMode(13, OUTPUT);

 pinMode(12, OUTPUT);

 pinMode(11, OUTPUT);

 pinMode(10, OUTPUT);

 pinMode(9, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

 digitalWrite(13, HIGH);//Turn on LED attached to pin 13

 delay(1000);// wait for a second

 digitalWrite(12, HIGH);//Turn on LED attached to pin 12

 delay(1000);  // wait for a second

digitalWrite(11, HIGH);//Turn on LED attached to pin 11

 delay(1000);// wait for a second

 digitalWrite(10, HIGH);//Turn on LED attached to pin 10

 delay(1000);  // wait for a second

  digitalWrite(9, HIGH);//Turn on LED attached to pin 9

 delay(1000);                       // wait for a second

 digitalWrite(9, LOW); //Turn off LED attached to pin 9

 delay(1000);                       // wait for a second

 digitalWrite(10, LOW);  //Turn off LED attached to pin 10

 delay(1000);  // wait for a second

 digitalWrite(11, LOW); //Turn off LED attached to pin 11

 delay(1000);                       // wait for a second

 digitalWrite(12, LOW);  //Turn off LED attached to pin 12

 delay(1000);  // wait for a second

 digitalWrite(13, LOW); //Turn off LED attached to pin 13

 delay(1000);                       // wait for a second

}

However, this is bulky and required a lot of copying and pasting. So, we are going to introduce a “for” loop. Just like the void loop ( ), a “for” loop repeats what ever is inside the {}. Unlike the void loop, we can give it a counter, tell it when to stop, and how much to increment each time. For loops have the following syntax:



For loops need 3 “arguments” that tell them what to do. Think of these as rules the loop needs to follow. The first argument is that is called a counter. Usually it looks something like this “int i=0”. What does this mean? With that piece of code, we are creating a variable named “i” that is an integer and we are setting that variable to 0.

*What is a variable?*

*Think of a variable like a box that you can store stuff*

So what would that look like for us? Here is an example of our code



In the first argument of the for loop, we are creating a variable whose name is “i” and that variable can store an integer. That last sentence was a little complicated, so lets break it down. Firtles It will be out counter, or the variable that gets incremented everytime the computer executes a loop

### Challenge 3 and 4: RGB LED

Introduce the concept of RGB LED.

An RGB LED is a type of light that can light up different colours. It does this by mixing red, green, and blue. That is where the name RGB comes from.

If red is on light is red, same with green and blue ‐If more than one colour is on, on you get a new colour.

‐Red + Green = Yellow

‐Red + Blue = Magenta

‐Blue + Green = Cyan

‐Blue + Green + Red = White

Challenge 3: Red ‐ Green ‐ Blue

Challenge 4: Red, purple, blue, turquoise, green, yellow, and white



CHALLENGE 5: PHOTOSENSOR

Coming soon

CHALLENGE 6: POTENTIOMETER

Coming soon