

In this Assignment, you will be introduced to the other half of electronics, the code, while initially intimidating with a little bit of curiosity and reading it really is just an instruction guide to making electronics move and function. There are two different ways to produce working code, one is through a text editor or IDE. (Integrated Development

Environment) This is most likely what you think of when you picture coding, however, there is another type that is just as powerful. Visual Coding, using blocks or “nodes” to control your electronics, both do the same thing so it is up to you to decide which you’re more comfortable with.

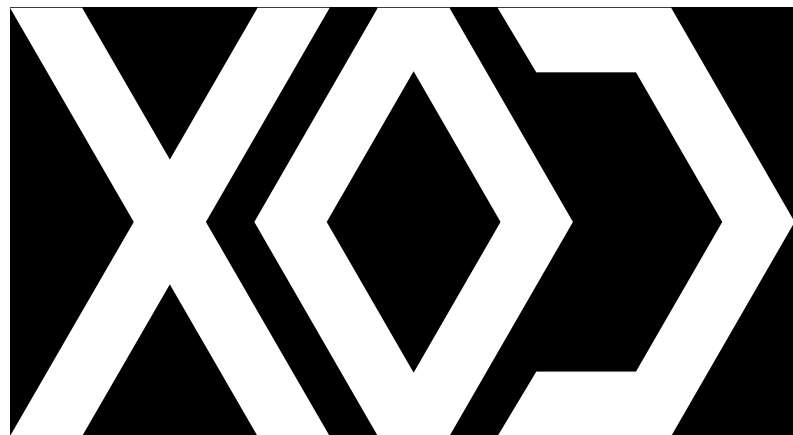
Within this assignment, we will introduce you to both Arduino IDE and XOD. If you have made it this far in the module then you have at least opened Arduino IDE and read through some instructions. Arduino IDE is the backbone of what makes Arduino an open-platform microcontroller. It allows ultimate flexibility in how your Arduino acts. While initially intimidating understanding what code makes what happen is the easy part. There are plenty of experts out there to help and plenty of forums that have more than likely already answered the question you are asking.

XOD on the other hand is more accustomed to those of us who are new to coding and may have used other visual coding platforms before such as Grasshopper, Blockly Code, Scratch Code, etc. These allow us to better understand just how code is linked together to perform functions. XOD is no different than Arduino IDE its only difference is its appearance, in XOD you place a node or block down onto the canvas and it will show you its required inputs and outputs. This makes it much easier to string together a series of nodes to move a motor.

To start here are some links on the “Basics” of Arduino Coding. Read up on how the IDE functions initially to better understand the concepts of working with Arduino then make your decision on whether to work with the IDE or XOD. Lastly, as a warning, XOD is relatively new in the Arduino Coding Community, therefore many of the GROVE Sensors found in the Lab are not useable within XOD. You’re more than welcome to search for them but understand that you may have to revert back to the IDE for specific functions.



<https://www.arduino.cc/en/software>



<https://xod.io/downloads/?skip=1>

In this assignment, we will be using the Arduino IDE to create a sketch(code) that allows us to interface with electronics. Code is the instruction set that tells electronic components what and how to do what we expect. In Arduino IDE we will be using a text interface to generate an instruction set that will alter a numerical value using

a potentiometer. To start we will need to understand some of the basic rules that come with C++ coding.

<https://www.javatpoint.com/arduino-coding-basics>

A4a- Altering Data

(Code Download Link)

Links Needed

[Grove Sensors Wiki Page](https://wiki.seeedstudio.com/Grove-System/)
<https://wiki.seeedstudio.com/Grove-System/>

To complete this assignment you will need to select a few sensors to work with. Grab a Visible Light Sensor and an Ultrasonic Ranger and one potentiometer (either sliding or rotational). The goal of this circuit is to have the Arduino record all of the sensor's values and then alter their strength by the value of the potentiometer.

Sensor Links
<https://wiki.seeedstudio.com/Grove-Light-Sensor/>

<https://wiki.seeedstudio.com/Grove-Ultrasonic-Ranger/>

<https://wiki.seeedstudio.com/Grove-Rotary-Angle-Sensor/>

<https://wiki.seeedstudio.com/Grove-Slide-Potentiometer/>

```
M5A3 $
1 ////////////////////////////////////////////////////
2 #include "Ultrasonic.h"
3 Ultrasonic ultrasonic(3);
4 #define ROTARY_ANGLE_SENSOR A1
5 #define ADC_REF 5
6 #define GROVE_VCC 5
7 #define FULL_ANGLE 300
8 int light_sensor = A0;
9 ////////////////////////////////////////////////////
10
11 void setup()
12 {
13   Serial.begin(9600);
14   // Set PinMode of Potentiometer to Input
15 }
16
17 ////////////////////////////////////////////////////
18
19 void loop()
20 {
21   //
22   //
23   //
24   //
25   //
26   //
27   //
28   //
29   //
30   //
31   float Light = raw_light/degrees;
32   float Measure = RangeInInches/degrees;
33
34   Serial.println(Light);
35   delay(250);
36   Serial.println(Measure);
37   delay(250);
38 }
39
```

Instructions:
Go to each of those links and look for the example code section. Look under the Void Setup and Void Loop functions. Read to understand how it is drawing in the information from the sensors. Figure out what code you will need from each function to make the sensor work. After the code has been copied over to your IDE alter your variables to match that which is shown below.

value = raw_light

(Proceed to M5A5)

In this assignment, we will be using the XOD IDE to create a sketch(code) that allows us to interface with electronics. Code is the instruction set that tells electronic components what and how to do what we expect. In XOD IDE we will be using a node interface to generate an instruction set that will alter a numerical value using

a potentiometer. To start we will need to understand some of the basic rules that come with C++ coding.

First, Complete the Tutorial within XOD IDE then begin coding your Altering Data assignment.

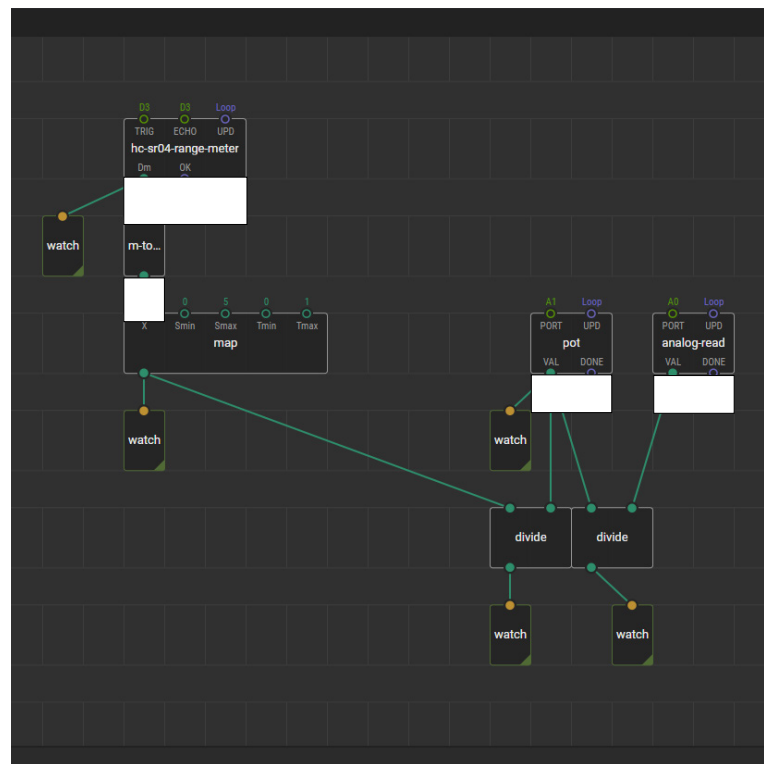
A4b- Altering Data

(Code Download Link)

To complete this assignment you will need to select a few sensors to work with. Grab a Visible Light Sensor and an Ultrasonic Ranger and one potentiometer (either sliding or rotational). The goal of this circuit is to have the Arduino record all of the sensor's values and then alter their strength by the value of the potentiometer.

(If you receive an error about cloud usage capacity, create an account using the orange person icon in the top right, then sign in to resolve this error)

(Proceed to M5A5)



Instructions:

First, complete the tutorial prompted when you open XOD IDE. This will show you how to navigate and use simple nodes. Then fill in the blanks on the code above using the hardware names below. As you fill it out use the HELP menu to understand what the node is doing. Be sure to mirror where you plugged in your hardware to the ports mentioned in the inputs!

- Node Bank
- Potentiometer
- Analog Read
- hc-sr04-range-meter
- mtoin