The Engineering Innovation Center is a large academic maker space with plenty of tools and equipment. In order to use these items you must have the proper training. This online popup class will teach the basic fundamentals of this tool or piece of equipment.

Microcontroller I – will demonstrate the basic use of an Arduino Uno microcontroller along with example programs. The key is the right tool for the right job. This class will discuss how to identify and use these tools.
What is a microcontroller?

- A **microcontroller** is a small computer (SoC) on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.

- Often **small and low cost**. The components are chosen to minimize size and to be as inexpensive as possible.

Application of Microcontroller in Day to Day Life Devices:

- Light sensing & controlling devices
- Temperature sensing and controlling devices
- Fire detection & safety devices
- Industrial instrumentation devices
- Process control devices
Microcontrollers

- Teensy
- MSP430 TI Launchpad
- Adafruit Circuit Playground
- ATtiny 85
Arduino Examples
Shields for Arduino

Shield List
We will be using the Arduino Uno
TO GET STARTED DOWNLOAD THE ARDUINO SOFTWARE (IDE) & INSTALL

https://www.arduino.cc/en/Main/Software
Structure

- setup() { // put your setup code here, to run once:}
- loop() { // put your main code here, to run repeatedly:}

Control Structures
- if..else
- for
- switch case
- while / do while
- break

Further Syntax
- ; (Semicolon)
- {} (Curly Braces)
- // (Single Line Comment)
- /**/ (Multi-Line Comment)
- #define / #include

Other
- Arithmetic Operators =, +, -, /, *, %
- Comparison Operators ==, !>, <=
- Boolean Operators &&, ||, |, !
- Pointer Access Operators *, &
- Bitwise Operators &|, <<, ~
- Compound Operators ++, *+, |=
Variables

- Constants: HIGH, LOW, integer or floating point constants
- Data Types: void, boolean, unsigned char, double
- Conversion: char(), int(), byte()
- Variable Scope & Qualifiers: static, volatile, const
- Utilities: sizeof(), PROGMEM
Language Reference Part 3

Functions

- Digital I/O : pinMode()
- Analog I/O : analogRead()
- Advanced I/O : tone()
- Time : delay()
- Math : sqrt()
- Communication : serial

Other

- Characters
- Random Numbers
- Bits and Bytes
- External Interrupts
- Keyboard/Mouse
MAKE SURE BOARD IS SET TO ARDUINO/GENUINO UNO AND ALSO PORT IS SELECTED FOR THE USB OF THE BOARD
Arduino Uno Program #1

Using Push Button Switch demo

- Push button has pull-up resistor to 5V. Pin 4 will read values from button and Pin 8 will power LED.

- 10k Ohm resistor (Brown Black Orange), 680 Ohm (Blue Gray Brown)
Arduino Sketch / Code

```cpp
const int switchPin = 4; // Switch connected to PIN 4
const int ledPin = 8;   // LED connected to PIN 8
int switchState = 0;    // Variable for reading Switch status

void setup() {
    Serial.begin(9600); // Serial data transmission 9600 bits per second.
    pinMode(ledPin, OUTPUT); // LED PIN is an output
    pinMode(switchPin, INPUT); // Switch PIN is input with PULLUP
}

void loop() {
    switchState = digitalRead(switchPin); // Reads the status of the switch.
    if(switchState == LOW) { // If the switch is pressed.
        digitalWrite(ledPin, HIGH); // LED ON
        Serial.print("Led was turn on. Switch State: ");
        Serial.println(switchState, BIN); // Prints Binary value of switchState.

        delay(3000); // 3 second delay
        switchState = digitalRead(switchPin); // Reads the status of the switch.
        digitalWrite(ledPin, LOW); // LED OFF
        Serial.print("Led was turn off. Switch State: ");
        Serial.println(switchState, BIN); // Prints Binary value of switchState.
    }
}
```
Arduino Uno Program #2

Sound sensor

- 390 Ohm resistors, Sound Sensor
- Connect pin G to ground, + to 3.3V, A0 to analog pin 0

More Info On Project
Calibrating Sensor

```cpp
const int soundSensor = A0;
int soundLevel = 0;
int highThreshold = 0;
int lowThreshold = 0;

void setup() {
    Serial.begin(9600);
    pinMode(soundSensor, INPUT);
    highThreshold = analogRead(soundSensor);
    lowThreshold = analogRead(soundSensor);
}

void loop() {
    soundLevel = analogRead(soundSensor);
    Serial.println(soundLevel); // Comment this line after seeing Serial Plotter (Tools)
    /*if(soundLevel > highThreshold) { // Uncomment after seeing Serial Plotter (Tools)
        highThreshold = soundLevel;
    } else if (soundLevel < lowThreshold) {
        lowThreshold = soundLevel;
    } else {
        Serial.print("High Threshold:");
        Serial.println(highThreshold);
        Serial.print("Low Threshold:");
        Serial.println(lowThreshold);
        Serial.println();
    }*/
```
void outThreshold(int cPin){
    switcher = false;
    if(intensity[cPin] < 10){
        intensity[cPin] = 255;
        delay(50);
        currentPin = currentPin + 1;
    }

    if(currentPin == numberOfPins){
        currentPin = 0;
    }
}

void inThreshold(){
    switcher = false;
    fadeCounter++;
    if(fadeCounter == fadeDelay){
        fadeCounter = 0;
        for(int i=0; i < numberOfPins; i++){
            analogWrite(LED Pins[i], intensity[i]);
        }
    }
    for(int i=0; i < numberOfPins; i++){
        if(intensity[i] < 0){
            intensity[i] = 0;
        }
    }
    inThreshold();
    switcher = true;
}
}

// Set Pin Numbers:
const int soundSensorPin = A0;
const int LEDPins[3] = {3, 5, 6};

// Variables That Will Change:
int soundReading = 0;
int highThreshold = 430;
int lowThreshold = 420;
int intensity[3] = {0, 0, 0};
inumberOfPins = 3;
in currentPin = 0;
int fadeCounter = 0;
in fadeDelay = 50;
boolean switcher = true;

void setup(){
    pinMode(soundSensorPin, INPUT);
    for(int i=0; i < numberOfPins; i++){
        pinMode(LED Pins[i], OUTPUT);
    }
}

void loop(){
    soundReading = analogRead(soundSensorPin);
    if(soundReading > highThreshold || soundReading < lowThreshold){
        if(switcher){
            outThreshold(currentPin);
        } else {
            if(switcher){
                inThreshold();
            } else {
                inThreshold();
            }
        }
    }
}
Arduino Project #3

Measure your reaction time

- Materials: Breadboard, LEDs (2), Push Buttons (3), Arduino, USB A/B, 100 Ohm Resistor (2), 10 kOhm Resistor (3), Wires (12)

- More Info