



windsor mold group

EPIC MAKERS' BASE

MICROCONTROLLERS AND PROGRAMMING

Arduino Tutorial

[Abstract](#)

This guide will help you get started with Arduino basics and programming.



University
of Windsor

EPICentre
Entrepreneurship + Practice + Innovation

What is Microcontroller?

A microcontroller (sometimes called an MCU or Microcontroller Unit) is a single Integrated Circuit (IC) that is typically used for a specific application and designed to implement certain tasks. Products and devices that must be automatically controlled in certain situations, like appliances, power tools, automobile engine control systems, and computers are great examples, but microcontrollers reach much further than just these applications.

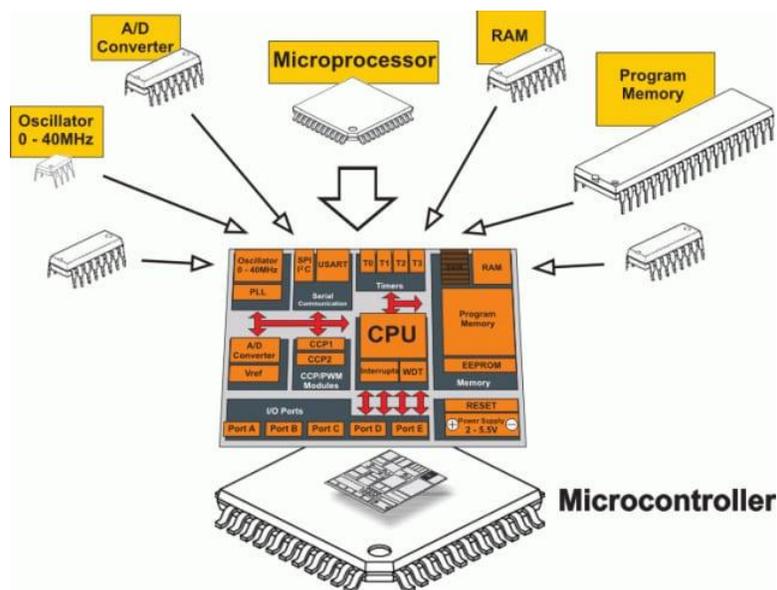
Essentially, a microcontroller gathers input, processes this information, and outputs a certain action based on the information gathered. Microcontrollers usually operate at lower speeds, around the 1MHz to 200 MHz range, and need to be designed to consume less power because they are embedded inside other devices that can have greater power consumptions in other areas.

Types of Microcontrollers:

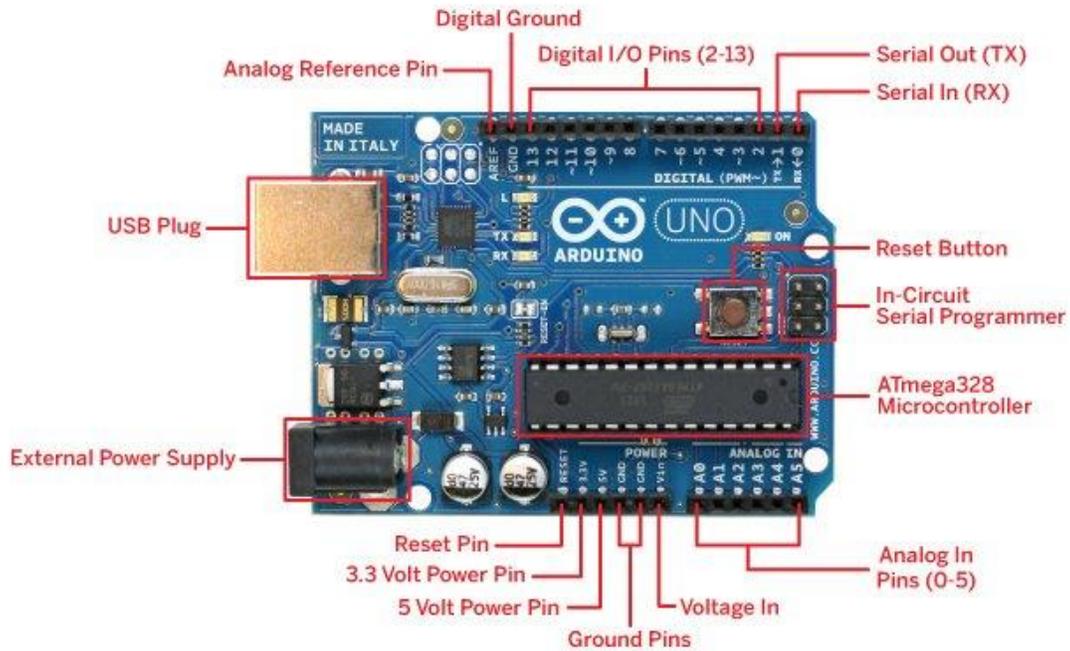
- PIC Microcontroller
- ARM Microcontroller
- 8051 Microcontroller
- AVR Microcontroller
- MSP Microcontroller

Inside a Microcontroller

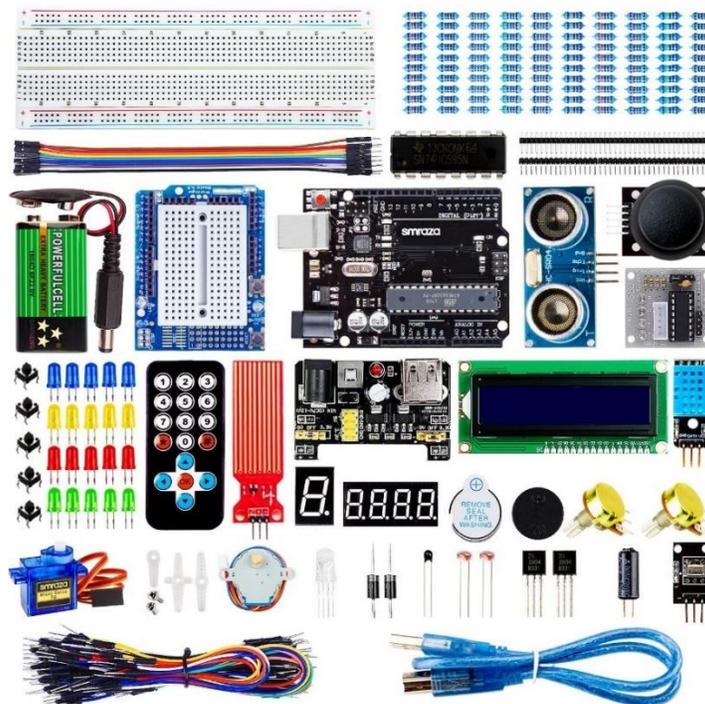
A microcontroller can be seen as a small computer, and this is because of the essential components inside of it; the Central Processing Unit (CPU), the Random-Access Memory (RAM), the Flash Memory, the Serial Bus Interface, the Input/Output Ports (I/O Ports), and in many cases, the Electrical Erasable Programmable Read-Only Memory (EEPROM).



Arduino Microcontroller

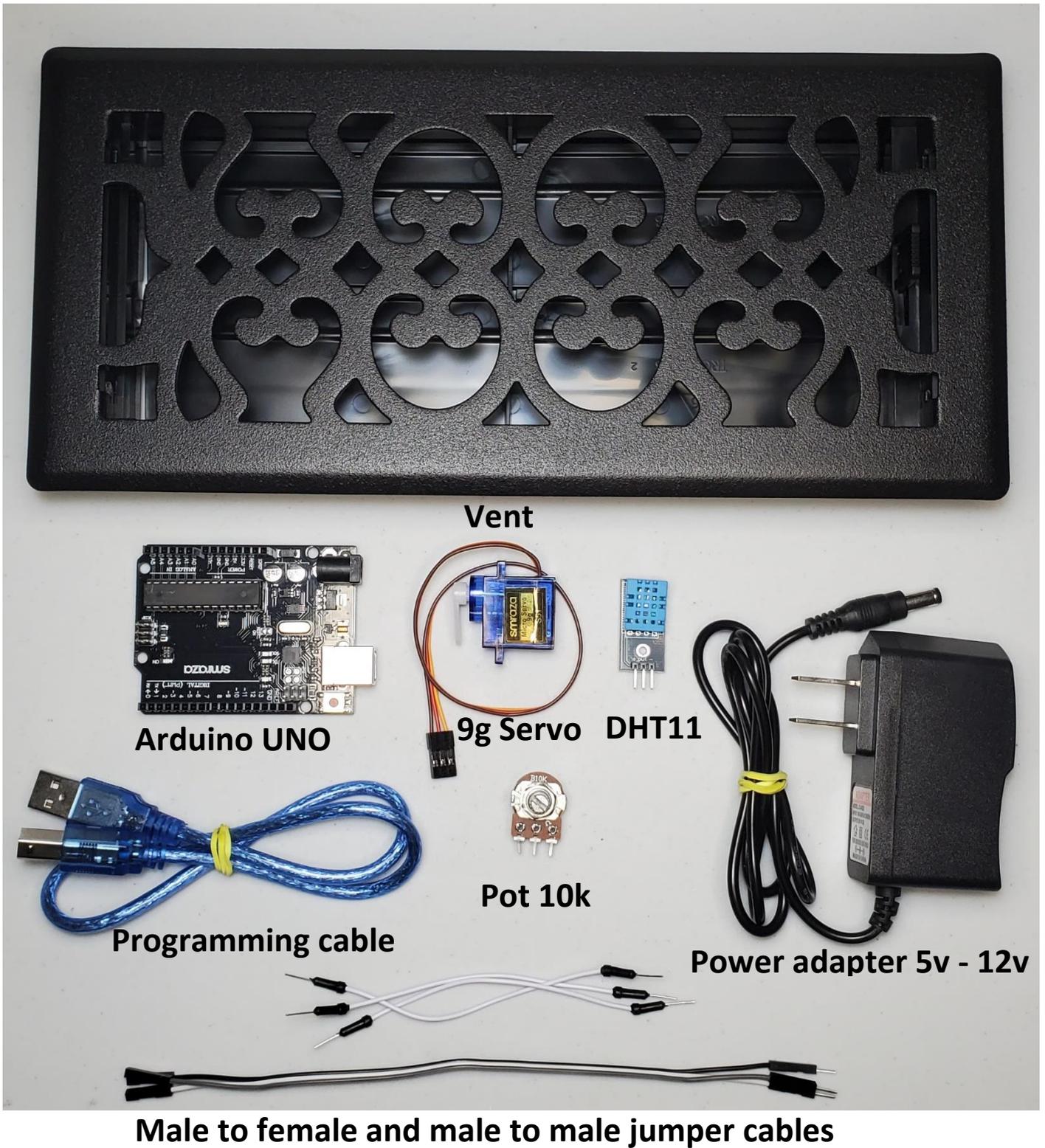


Arduino Uno is a microcontroller board based on the ATmega328P ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.



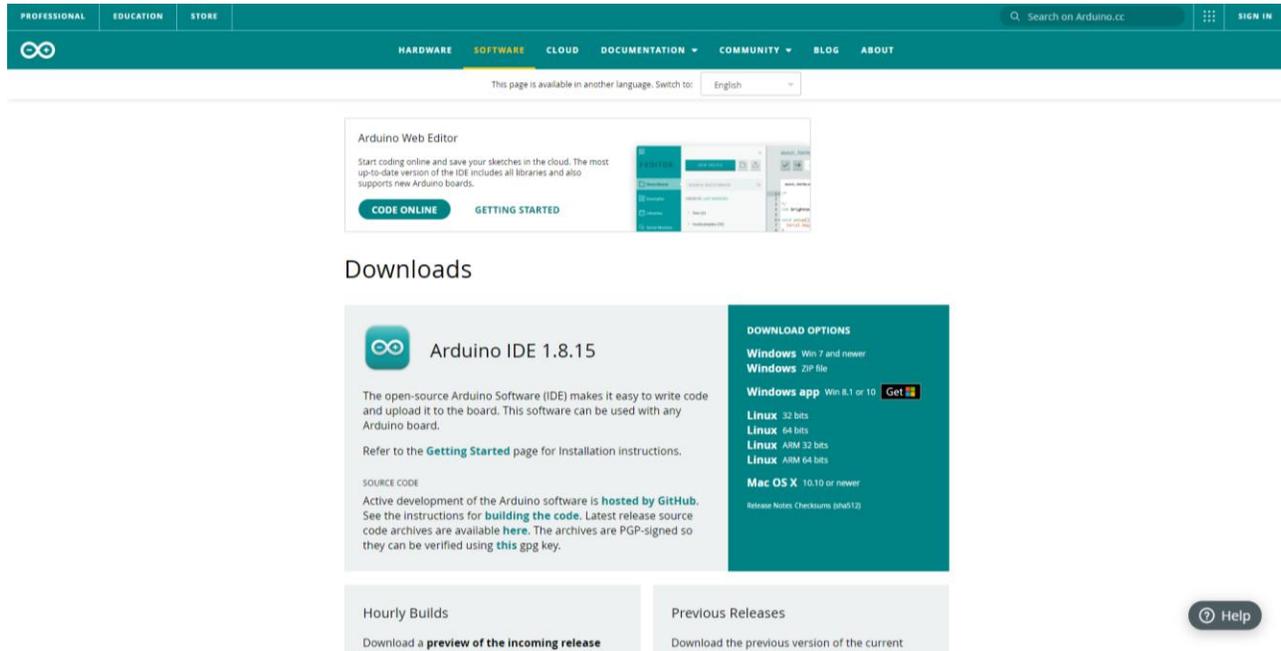
Arduino project kit

DIY Temperature Controlled Vent



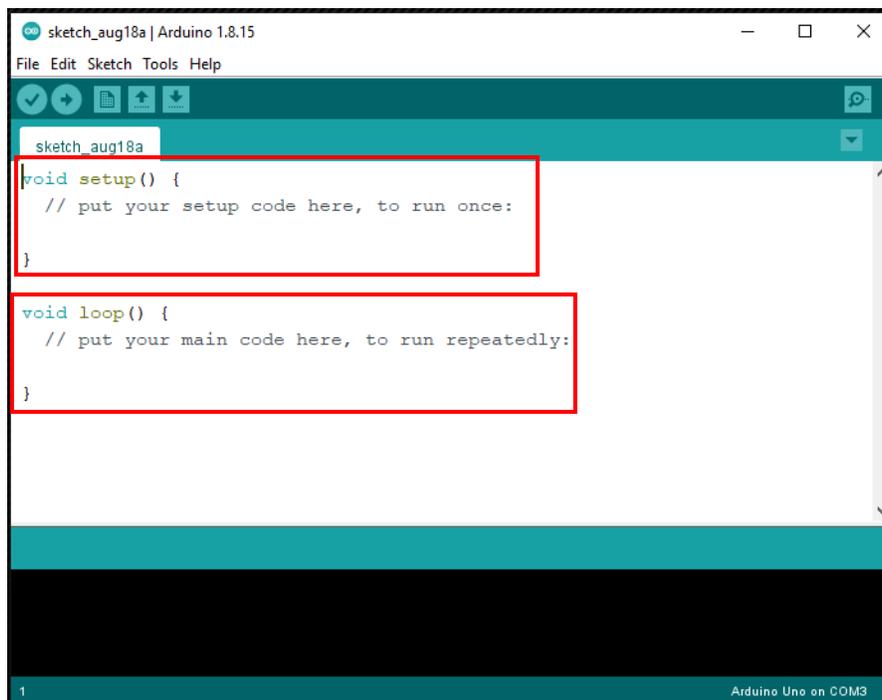
Software and Programming

Download and install Arduino software using <https://www.arduino.cc/en/software>



The screenshot shows the Arduino website's software page. At the top, there is a navigation bar with links for PROFESSIONAL, EDUCATION, STORE, and a search bar. Below the navigation bar, there is a section for "Arduino Web Editor" with a "CODE ONLINE" button and a "GETTING STARTED" button. The main section is titled "Downloads" and features a large card for "Arduino IDE 1.8.15". The card includes a description of the IDE, a "SOURCE CODE" section, and a "DOWNLOAD OPTIONS" section with links for Windows, Linux, and Mac OS X. Below the main card, there are two smaller buttons: "Hourly Builds" and "Previous Releases".

On more details on how to install the software : <https://www.arduino.cc/en/guide/windows>



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_aug18a | Arduino 1.8.15". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for opening, saving, and running. The main editor area shows the following code:

```
sketch_aug18a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

The code is highlighted with red boxes. At the bottom of the IDE, it says "1" and "Arduino Uno on COM3".

Arduino IDE has two main function, Void setup and Void loop.

Coding Syntax and Function

Digital I/O

digitalRead()
digitalWrite()
pinMode()

Analog I/O

analogRead()
analogReference()
analogWrite()

Time

delay()
delayMicroseconds()
micros()
millis()

Math

abs()
constrain()
map()
max()
min()
pow()
sq()
sqrt()

Trigonometry

cos()
sin()
tan()

Communication

Serial
Stream

Constants

HIGH | LOW
INPUT | OUTPUT | INPUT_PULLUP
LED_BUILTIN
true | false
Floating Point Constants
Integer Constants

Data Types

array
bool
boolean
byte
char
double
float
int
long
short
size_t
string
String()
unsigned char
unsigned int
unsigned long
void
word

Control Structure

break
continue
do...while
else

for
goto
if
return
switch...case
while

Further Syntax

#define (define)
#include (include)
/* */ (block comment)
// (single line comment)
; (semicolon)
{ } (curly braces)

Arithmetic Operators

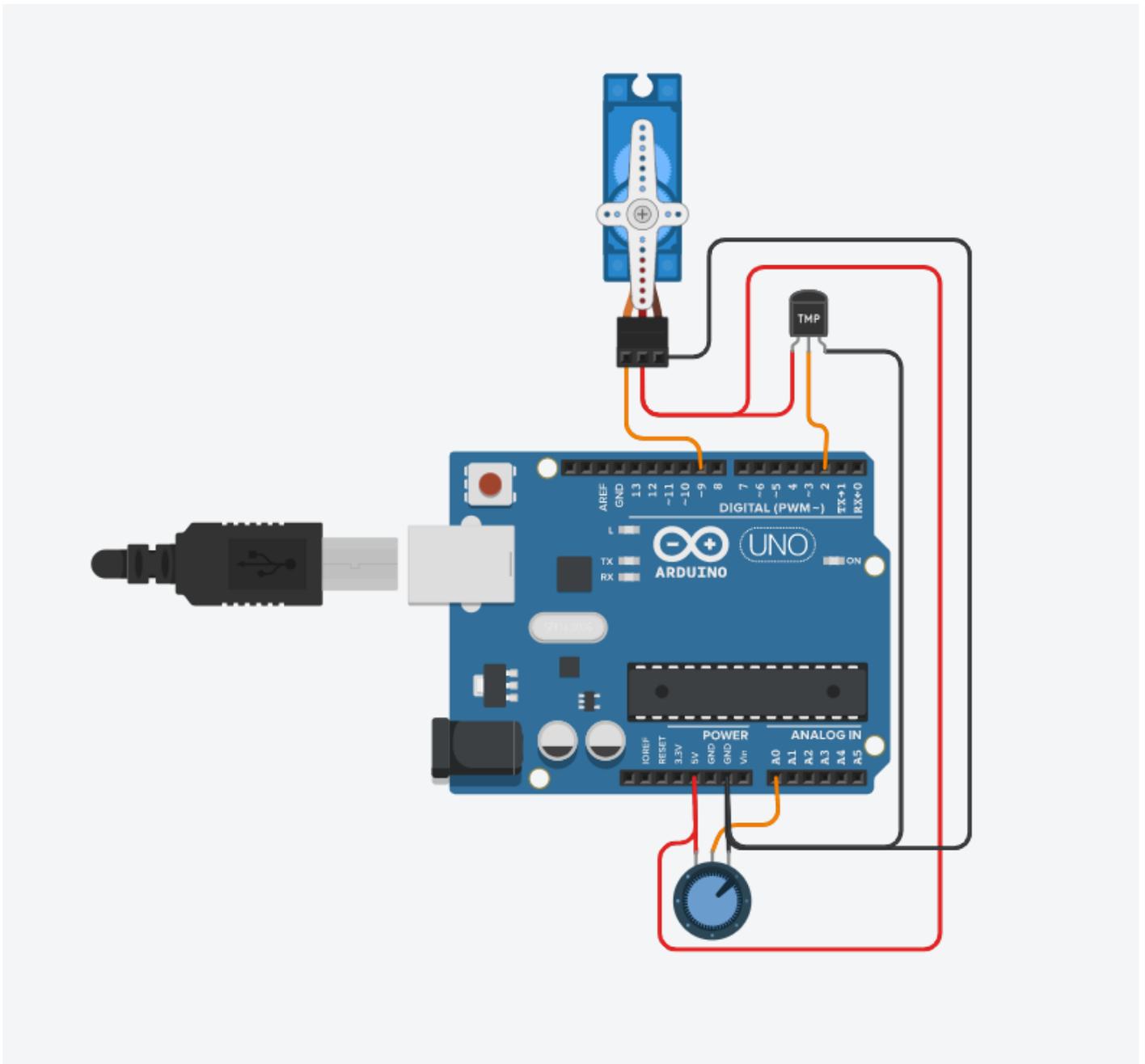
% (remainder)
* (multiplication)
+ (addition)
- (subtraction)
/ (division)
= (assignment operator)

Comparison Operators

!= (not equal to)
< (less than)
<= (less than or equal to)
== (equal to)
> (greater than)
>= (greater than or equal to)

For details on every function visit <https://www.arduino.cc/reference/en/>

Wiring Diagram

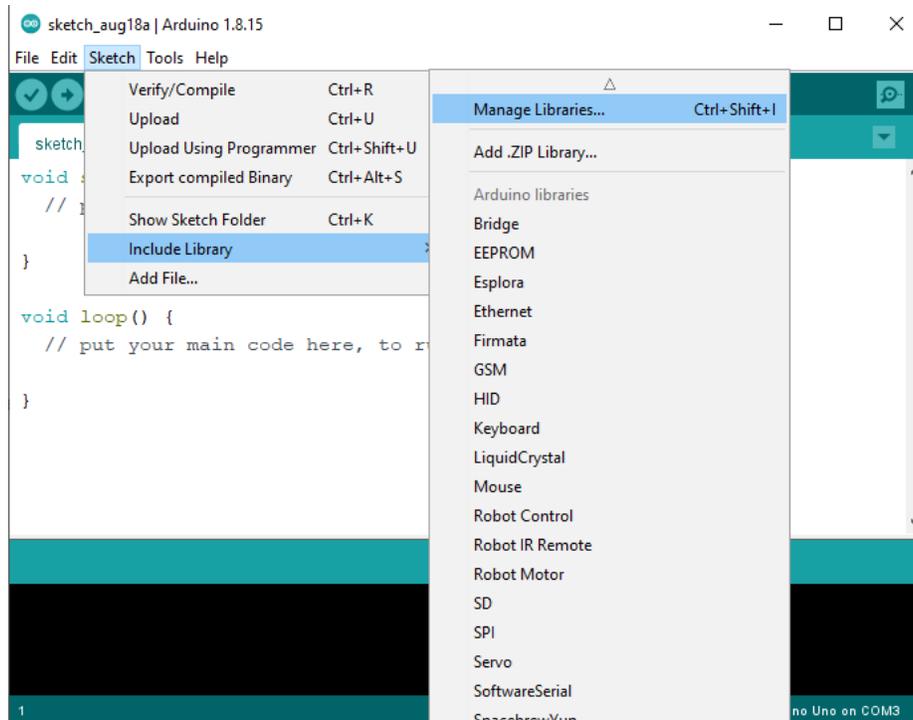


Servo Out → Pin D9
Servo Power → Pin 5v
Servo Gnd → Gnd

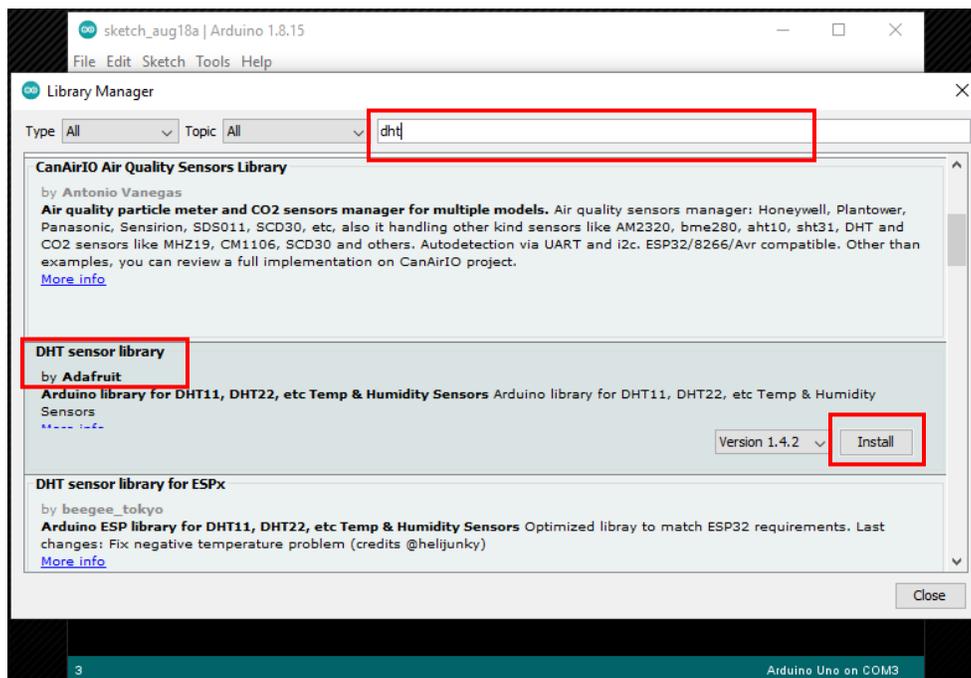
DHT11 Out → Pin D2
DHT11 Power → Pin 5v
DHT11 Gnd → Pin Gnd

Pot Pin 1 → Pin 5v
Pot pin 2 → Pin A0
Pot pin 3 → Pin Gnd

Library Setup



Select Sketch → Include Library → Manage Libraries.



Search DHT in the search bar, find library by Adafruit and click install.

Arduino Code

// REQUIRES the following Arduino libraries:

// - DHT Sensor Library: <https://github.com/adafruit/DHT-sensor-library>

// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit_Sensor

```
#include "DHT.h"
```

```
#include <Servo.h>
```

```
#define DHTPIN 2 // Digital pin connected to the DHT sensor
```

```
// Uncomment whatever type you're using!
```

```
#define DHTTYPE DHT11 // DHT 11
```

```
//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
```

```
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
Servo myservo;
```

```
int val=0;
```

```
int Lastval=0;
```

```
int potpin = 0;
```

```
int pot =0;
```

```
int Atemp;
```

```
int Temp;
```

```
int TempMax;
```

```
int TempMin;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```

Serial.println(F("DHTxx test!"));
myservo.attach(9);
dht.begin();

}

void loop() {
  // Wait a few seconds between measurements.
  delay(500);
  pot = analogRead(potpin);
  Temp = map(pot,0,1023,19,30);
  TempMax = Temp +1 ;
  TempMin = Temp -1 ;

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)

```

```
float hif = dht.computeHeatIndex(f, h);  
// Compute heat index in Celsius (isFahreheit = false)  
float hic = dht.computeHeatIndex(t, h, false);
```

```
Serial.print(F("Humidity: "));  
Serial.print(h);  
Serial.print(F("% Temperature: "));  
Serial.print(t);  
Serial.print(F("°C "));  
Serial.print(f);  
Serial.print(F("°F Heat index: "));  
Serial.print(hic);  
Serial.print(F("°C "));  
Serial.print(hif);  
Serial.println(F("°F"));  
Serial.println(Atemp);  
Serial.println(Temp);  
Serial.println(val);
```

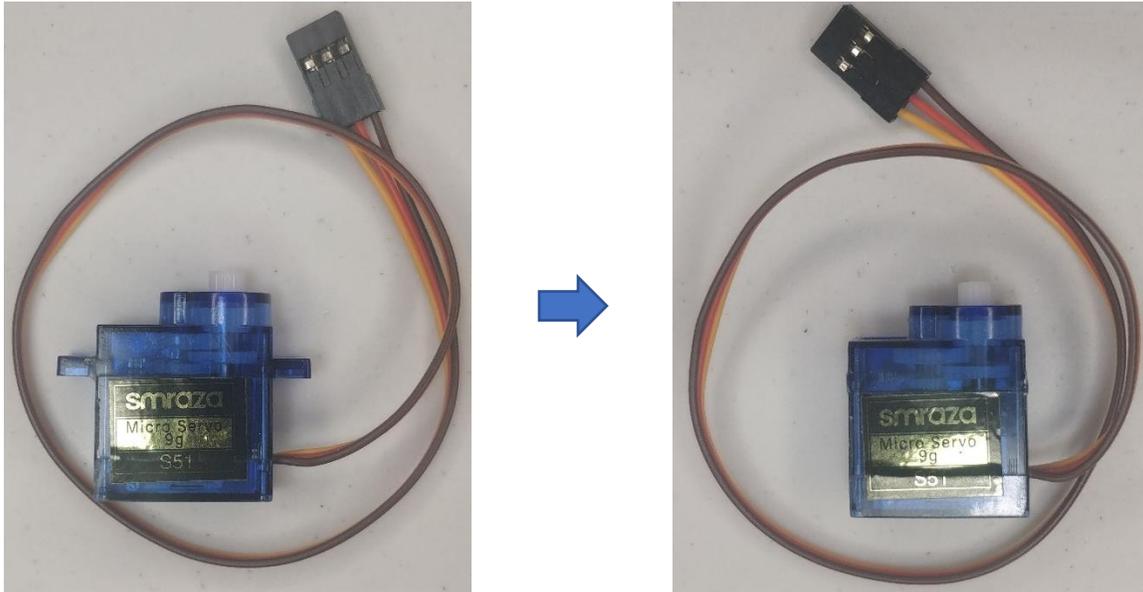
```
Atemp = round(t);
```

```
if ( Atemp == Temp)  
{  
  val =50;  
}
```

```
if( Atemp <= TempMin)  
{  
  val =90;
```

```
}  
  
if ( Atemp >= TempMax)  
{  
  val = 10;  
}  
  
  if (val != Lastval)  
{  
  Lastval = val;  
  myservo.attach(9);  
  delay(100);  
  myservo.write(val);  
  delay (1000);  
  myservo.detach();  
  
}  
  
}
```

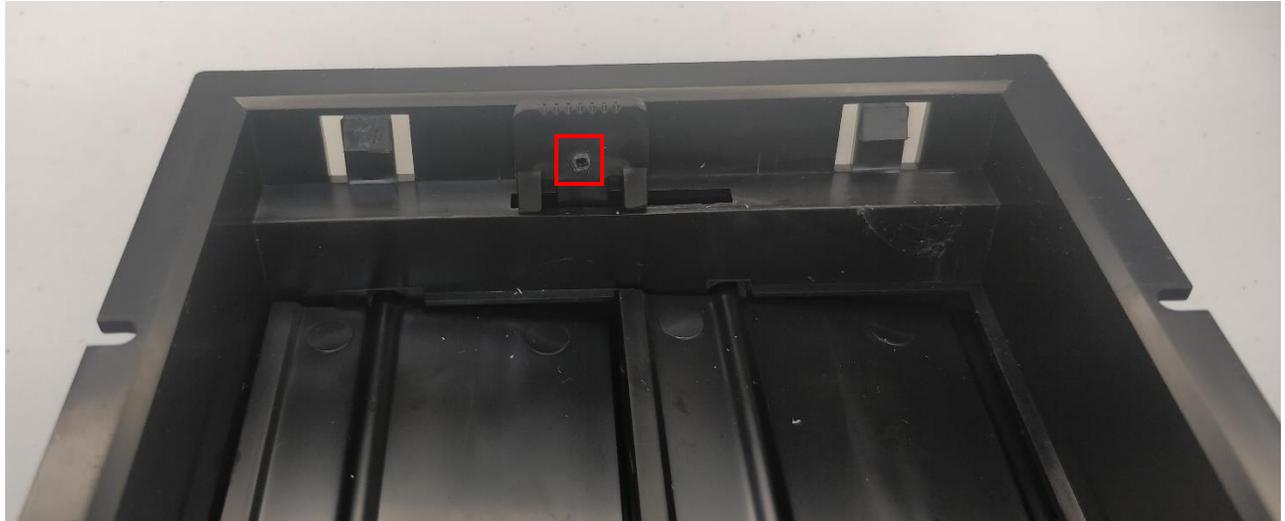
Mechanical Setup



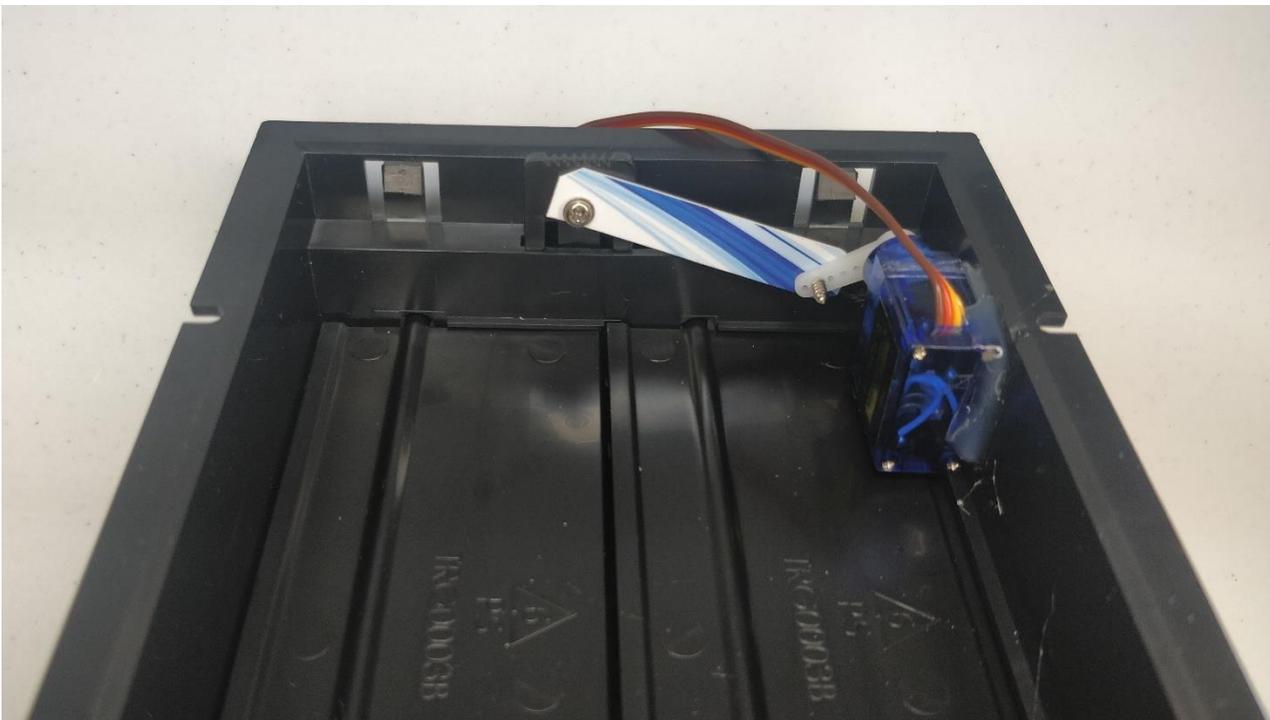
Remove the two support structures from servo using a small hacksaw.



Using an expired credit card or any plastic card cut a link as shown in the image above. The length should be according to the distance between the servo lever and the vent slider. Drill 2 holes such that the screw can freely rotate in the holes. The screws can be found in the servo package.



Drill a hole in center of vent slider with a 1mm drill bit.



After setting the servo arm angle properly, screw the link to the servo arm and stick servo to the vent wall using a hot glue gun. Adjust the link and screw the other side to the vent slider.

Contact Information

Contact us for any information at makerspace@uwindsor.ca

Website: www.epicentreuwindsor.ca

Follow us on social media:

Facebook: @WindsorMoldGroupEPICMakersBase and @epicentre.uwindsor

Twitter: @UofW_EPICentre

Instagram: @epicentre_uwindsor

LinkedIn: epicentre-uwindsor