

# How To: Electronics 1

Joint event ASME & Createathon, 4/5/22

TL;DR: Learn the basics of using arduino, LEDs, and servos.

Pptx:

[https://docs.google.com/presentation/d/1633uU\\_I0tyR3AmTB4Oc\\_qdrymQwPv0bhpo2FxxpQUZk/edit?usp=sharing](https://docs.google.com/presentation/d/1633uU_I0tyR3AmTB4Oc_qdrymQwPv0bhpo2FxxpQUZk/edit?usp=sharing)

## Setup and super-basics

What is Arduino?

- Small microprocessor board very popular for prototyping.
- Allows your computer to interface via serial with basic electronics
- Comes in a variety of sizes and configurations depending on need
- Most common is the Uno or the Leonardo, good for general use.

Installing the software

- Arduino is programmed in C, but don't worry, the syntax is fairly easy to infer, and the examples they provide are excellent.
- Two main sections of the code, "setup()" and "loop()".
  - Setup runs once, on arduino power up
  - Loop runs as long as the board is on, repeatedly.
- Download the IDE from <https://www.arduino.cc/en/software>

## Getting Started

Ex1: on-board blink

- Using the blink example, we'll demonstrate how to upload a sketch to the arduino and see it working
- COM port, check working, see upload process

/\*

Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED\_BUILTIN is set to the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:

<https://www.arduino.cc/en/Main/Products>

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

by Colby Newman

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

\*/

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

void setup() {

  // initialize digital pin LED\_BUILTIN as an output.

  pinMode(LED\_BUILTIN, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

  digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

  delay(1000); // wait for a second

  digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

  delay(1000); // wait for a second

}

Ex2: Serial read with simple button

- Use this to demonstrate serial, as well as basic using of pins

/\*

DigitalReadSerial

Reads a digital input on pin 2, prints the result to the Serial Monitor

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/DigitalReadSerial>

\*/

// digital pin 2 has a pushbutton attached to it. Give it a name:

int pushButton = 8;

// the setup routine runs once when you press reset:

void setup() {

```

// initialize serial communication at 9600 bits per second:
Serial.begin(9600);
// make the pushbutton's pin an input:
pinMode(5, OUTPUT); // MOD
pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(5, HIGH); // MOD
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1); // delay in between reads for stability
}

```

### Ex3: LED external

On a circuit this time, very simple  
 Demonstrate how to set up LED with resistor

/\*

Blink

Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED\_BUILTIN is set to the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at:

<https://www.arduino.cc/en/Main/Products>

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

by Colby Newman

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink>

\*/

```

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(11, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(11, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}

```

Ex4: LED tied to button

If button then LED

/\*

DigitalReadSerial

Reads a digital input on pin 2, prints the result to the Serial Monitor

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/DigitalReadSerial>  
\*/

```

// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 8;

```

```

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(5, OUTPUT); // MOD
  pinMode(11, OUTPUT); // MOD
  pinMode(pushButton, INPUT);
}

```

```

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(5, HIGH); // MOD

```

```

// read the input pin:
int buttonState = digitalRead(pushButton);
// print out the state of the button:
Serial.println(buttonState);
if(buttonState==0) {
    digitalWrite(11, LOW);
} else {
    digitalWrite(11, HIGH);
}
delay(1);    // delay in between reads for stability
}

```

## Ex5: LED control over serial

Show dimming / brighten via serial

```

const int ledPin = 11;    // the pin that the LED is attached to

void setup() {
    // initialize the serial communication:
    Serial.begin(9600);
    // initialize the ledPin as an output:
    pinMode(ledPin, OUTPUT);
}

void loop() {
    int brightness;

    // check if data has been sent from the computer:
    if (Serial.available()) {
        // read the most recent byte (which will be from 0 to 255):
        brightness = Serial.parseInt();
        // set the brightness of the LED:
        analogWrite(ledPin, brightness);
        Serial.println(brightness);
    }
}

```

## Ex6: Servo wiring and control

If we get this far?

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

```
Servo myservo; // create servo object to control a servo
```

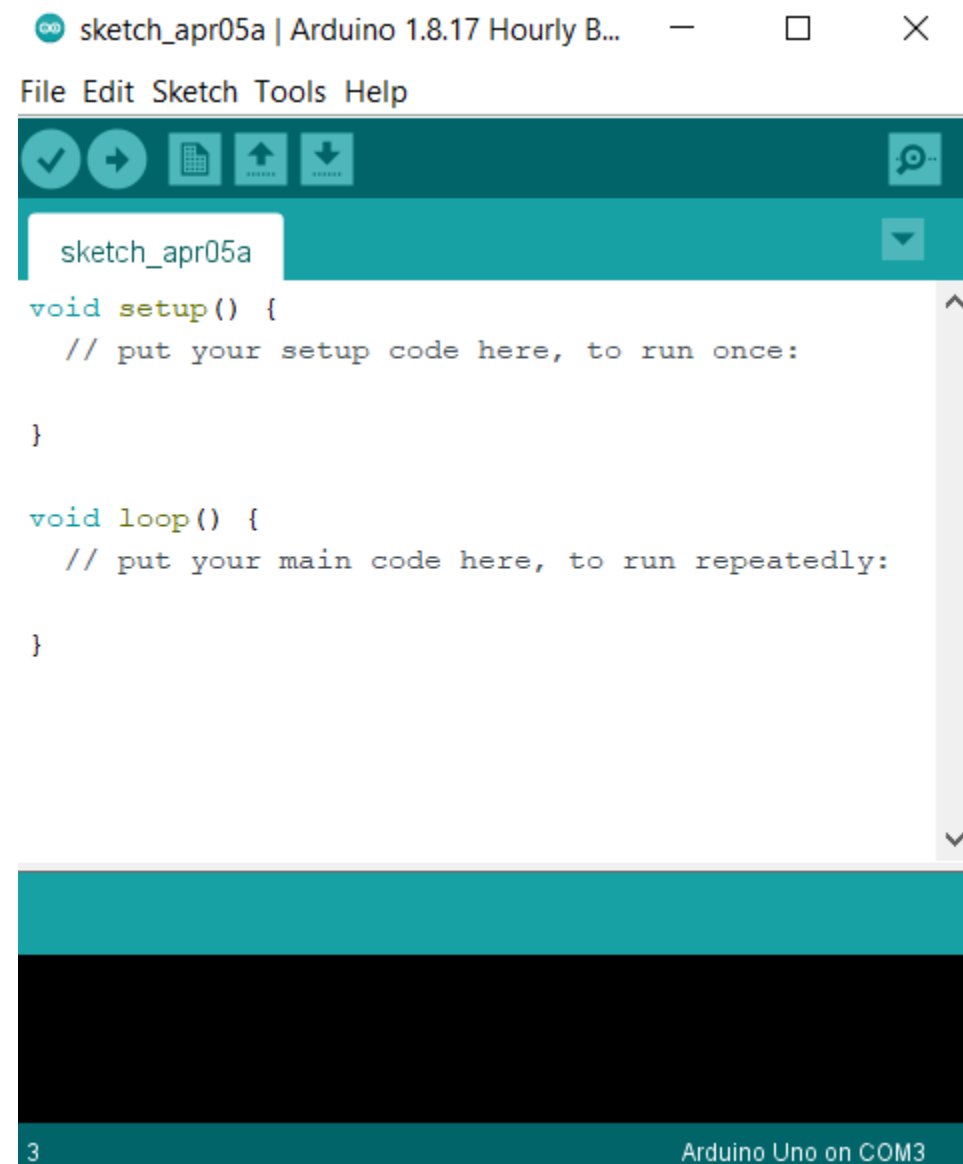
```
// twelve servo objects can be created on most boards
```

```
int pos = 0; // variable to store the servo position
```

```
void setup() {  
  myservo.attach(11); // attaches the servo on pin 9 to the servo object  
}
```

```
void loop() {  
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees  
    myservo.write(pos); // tell servo to go to position in variable 'pos'  
    delay(15); // waits 15ms for the servo to reach the position  
  }  
}
```

During Pres Transfer



## Blink

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

Done compiling.

Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.

```
// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 8;
```

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(5, OUTPUT); // MOD
  pinMode(pushButton, INPUT);
}
```

```
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(5, HIGH); // MOD
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1);           // delay in between reads for stability
}
```



```

*/

// digital pin 2 has a pushbutton attached to it. Give it a name:
int pushButton = 8;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(5, OUTPUT); // MOD
  pinMode(11, OUTPUT); // MOD
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(5, HIGH); // MOD
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  if(buttonState==0) {
    digitalWrite(11, LOW);
  } else {
    digitalWrite(11, HIGH);
  }
  delay(1); // delay in between reads for stability
}

```

```

const int ledPin = 11;      // the pin that the LED is attached to

void setup() {
  // initialize the serial communication:
  Serial.begin(9600);
  // initialize the ledPin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  int brightness;

  // check if data has been sent from the computer:
  if (Serial.available()) {
    // read the most recent byte (which will be from 0 to 255):
    brightness = Serial.parseInt();
    // set the brightness of the LED:
    analogWrite(ledPin, brightness);
    Serial.println(brightness);
  }
}

```

---

```

#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(11); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}

```