

TERRAPIN WORKS

INTRO TO ARDUINO

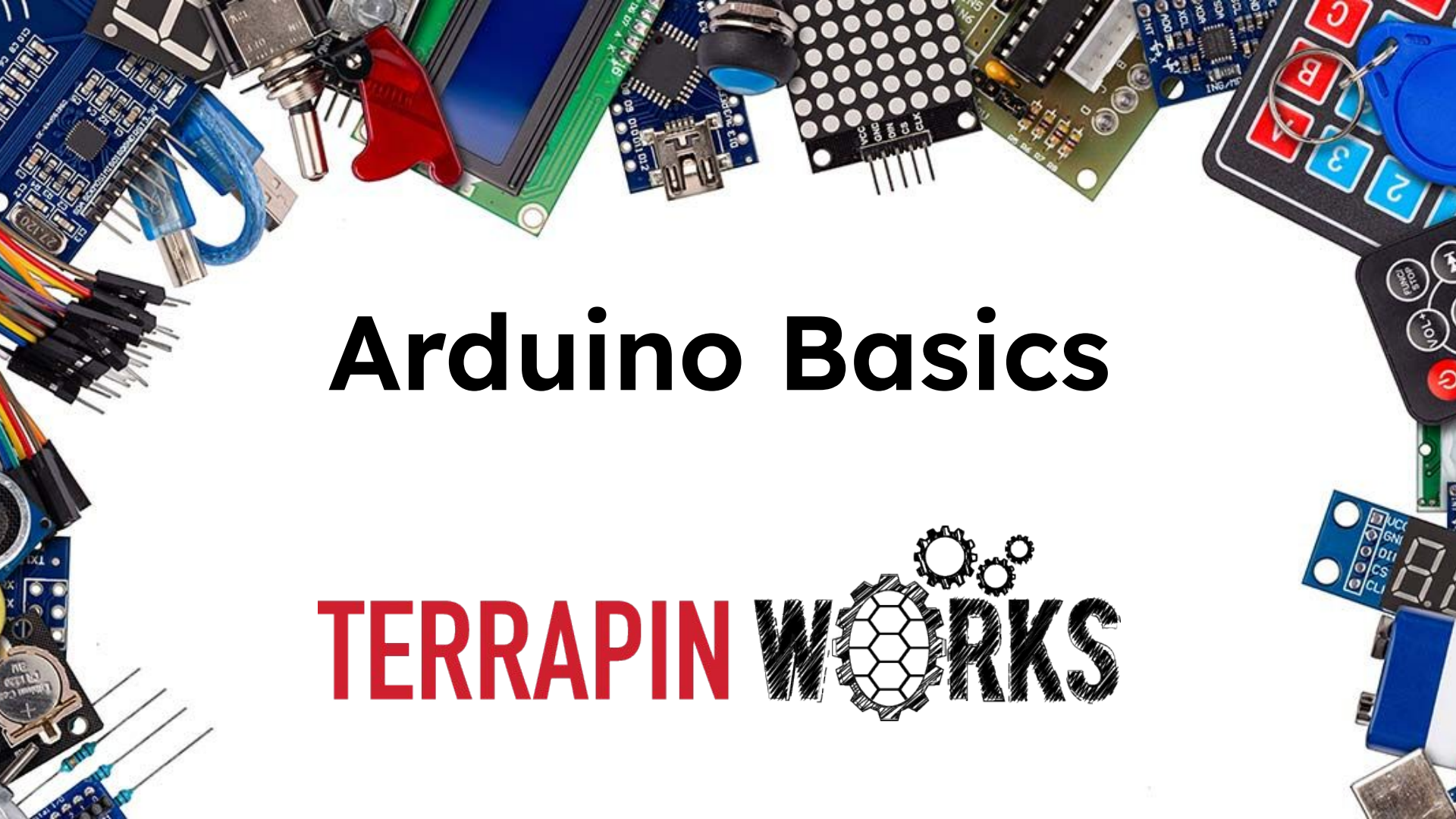


<https://go.umd.edu/TWSP25>

Students please sign in for
the TW Workshop!

While you are waiting,
please go to **tinkercad.com**
and make an account!



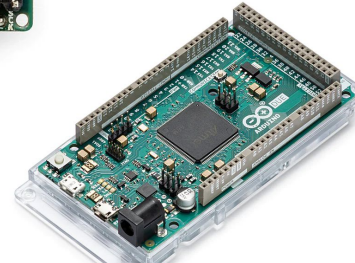
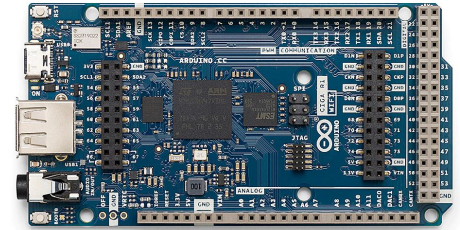
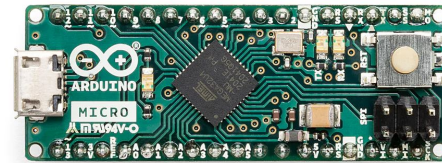
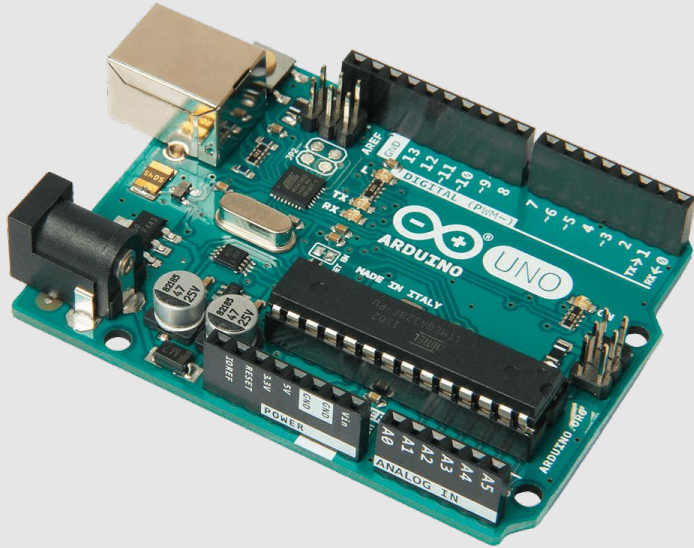


Arduino Basics

TERRAPIN  **WORKS**

What is it?

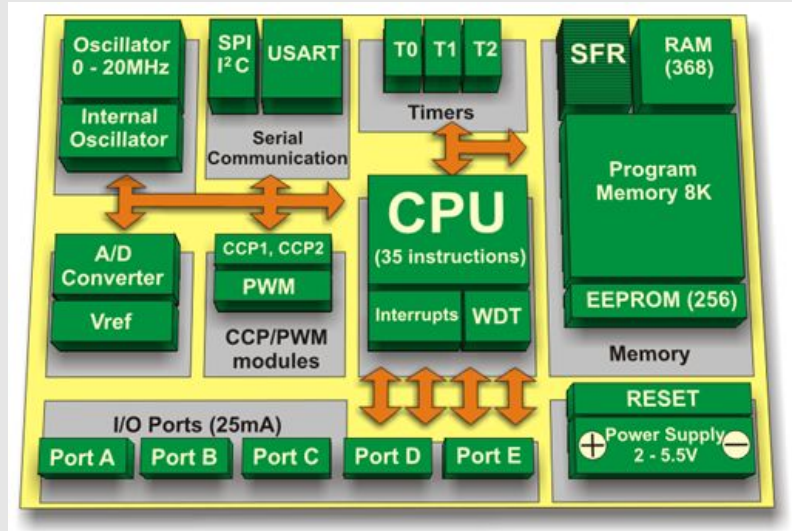
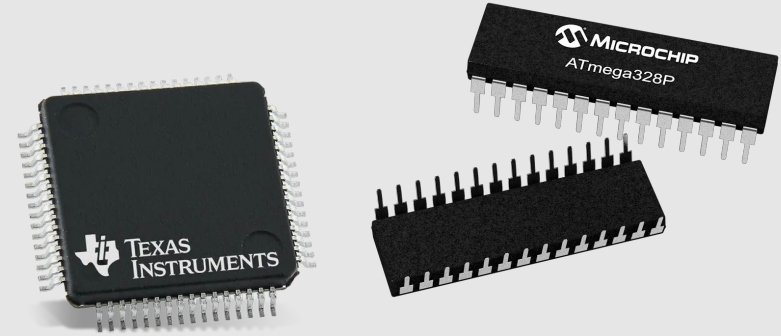
It's a family of user-friendly **microcontrollers** that can automate electronics projects!



The Microcontroller

A compact **Integrated Circuit (IC)** designed to control operations in an embedded system (circuit board). It's made of three major parts:

- Processor (CPU)
- Memory
- Input and Output (I/O) ports/devices



Why use one?

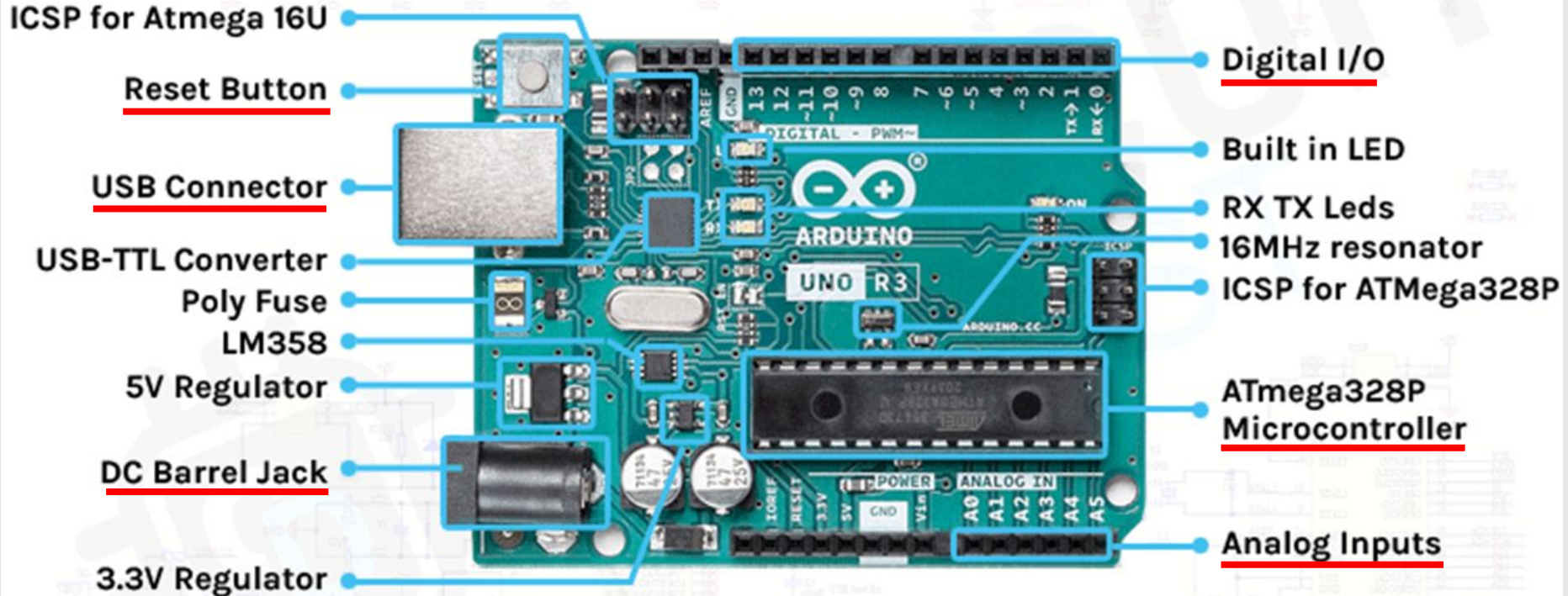
**Cheap &
Compact**

**Easy to
program**

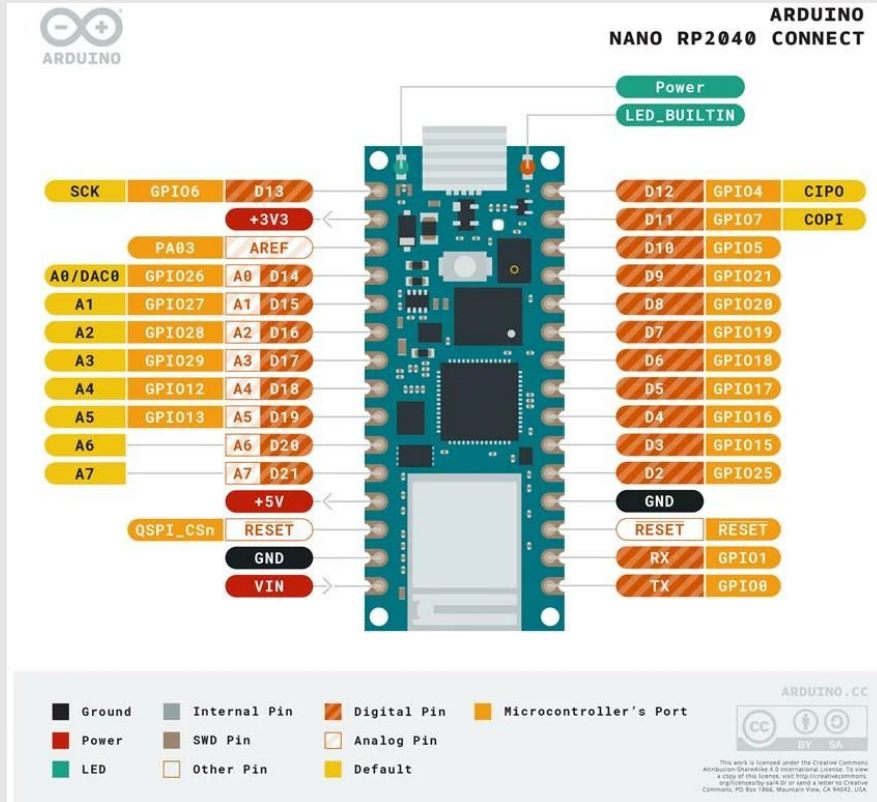
**Automates
tasks**



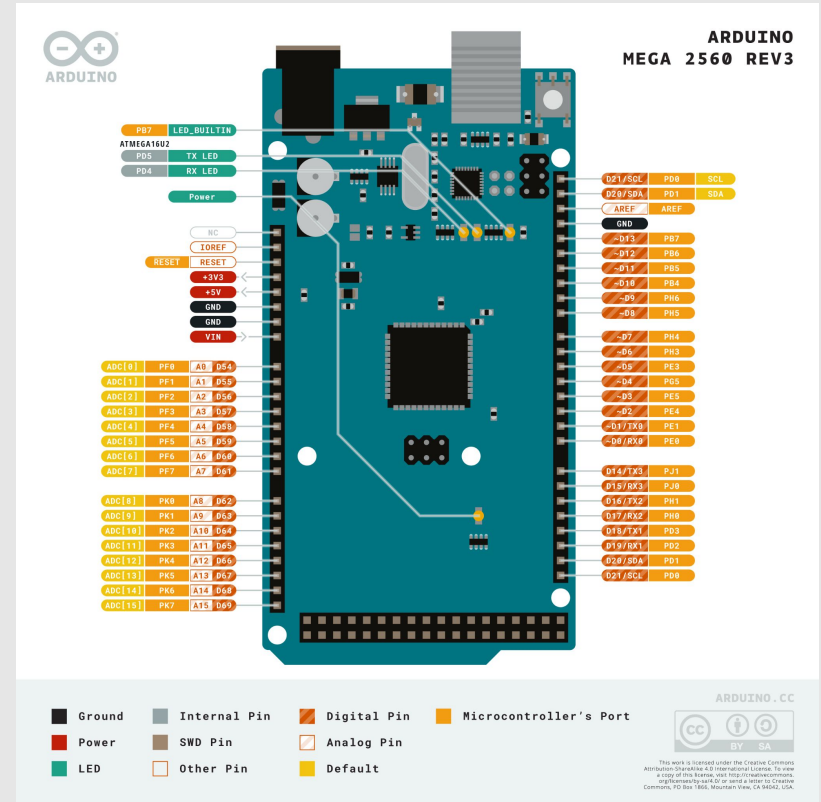
The Arduino Uno



Arduino Nano



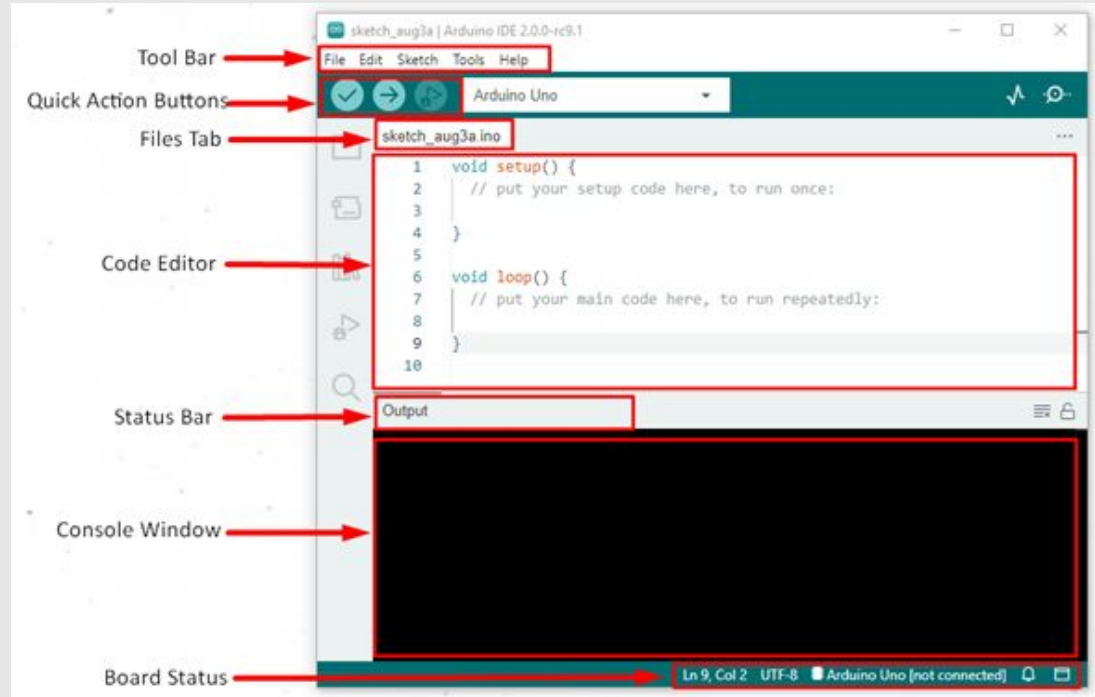
Arduino Mega



The Arduino IDE

(Integrated Development Environment)

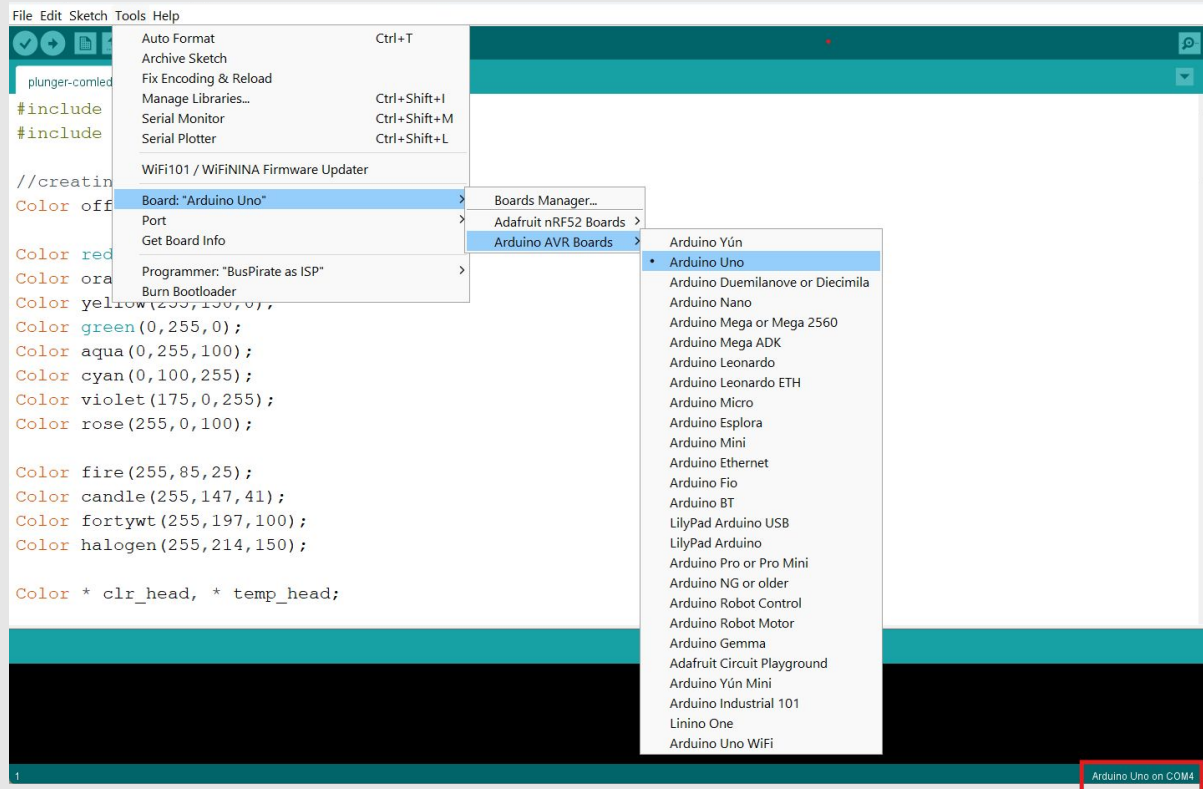
- Program Arduinos by **writing code** on a computer and **uploading** to the microcontroller via USB
- Uses a version of C++ programming language



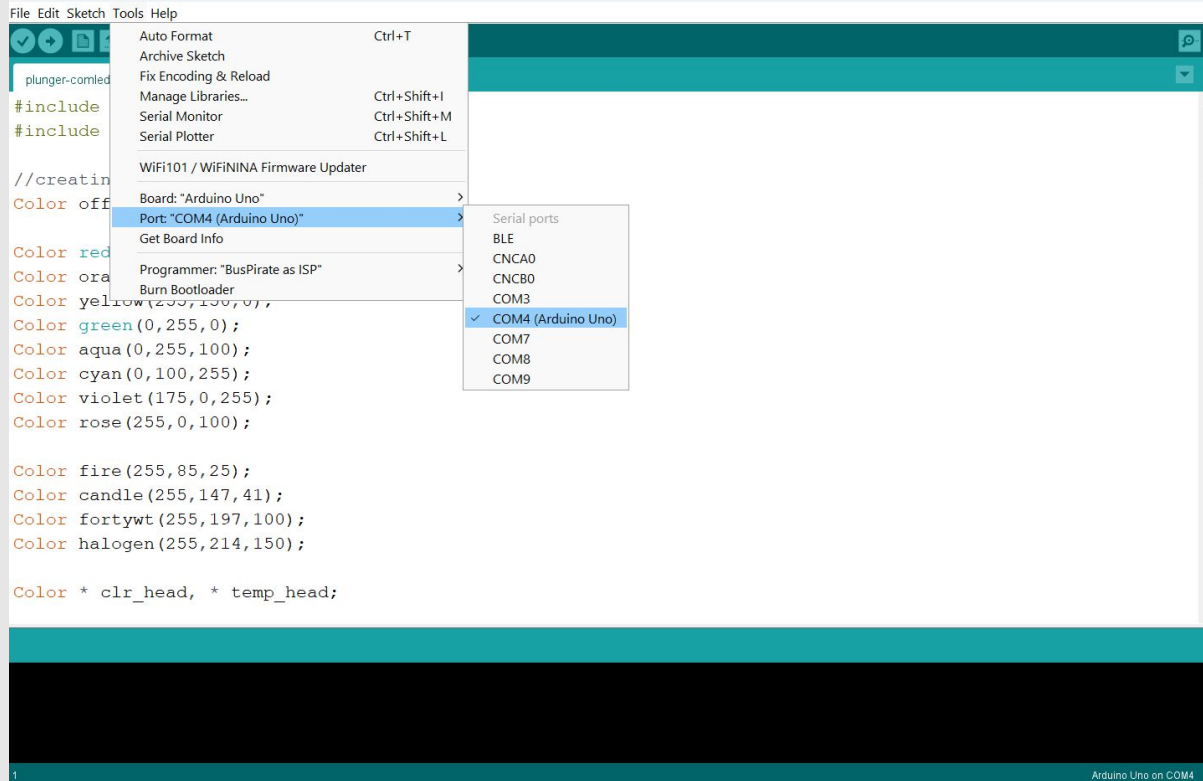
Connecting to the Arduino

Program Arduinos via a **USB** serial connection and the Arduino IDE



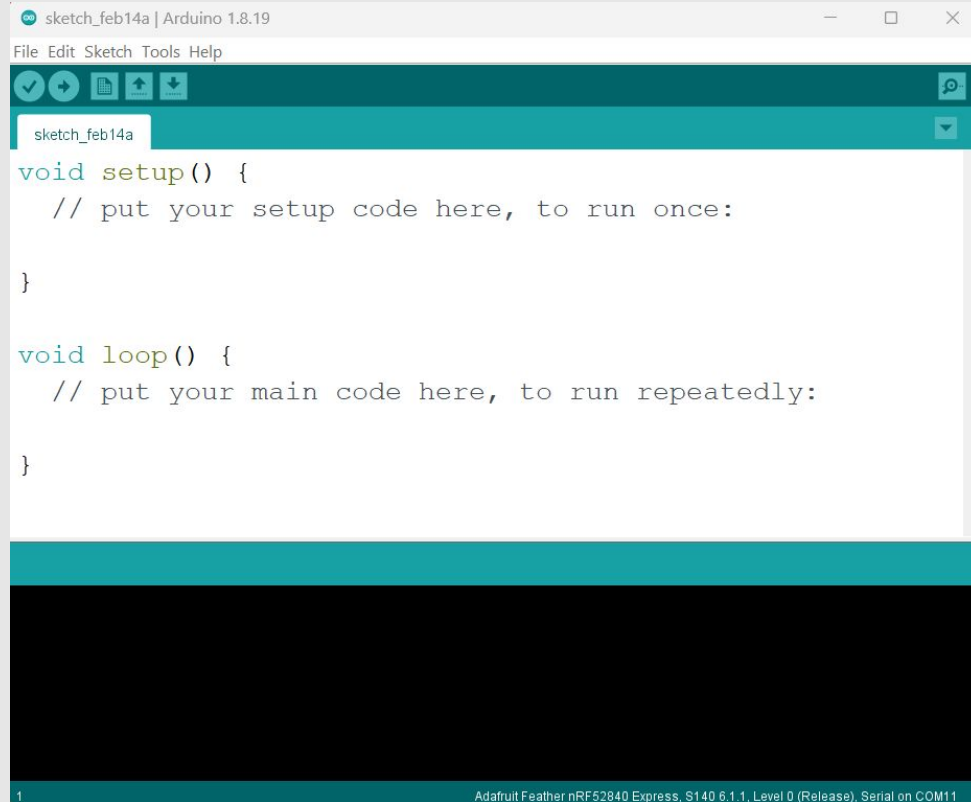


Tools > Board > Arduino AVR > Arduino Uno



Tools > Port > COMx (Arduino Uno)

Programming with Arduino



```
sketch_feb14a | Arduino 1.8.19
File Edit Sketch Tools Help
[Icons]
sketch_feb14a
void setup() {
  // put your setup code here, to run once:
}

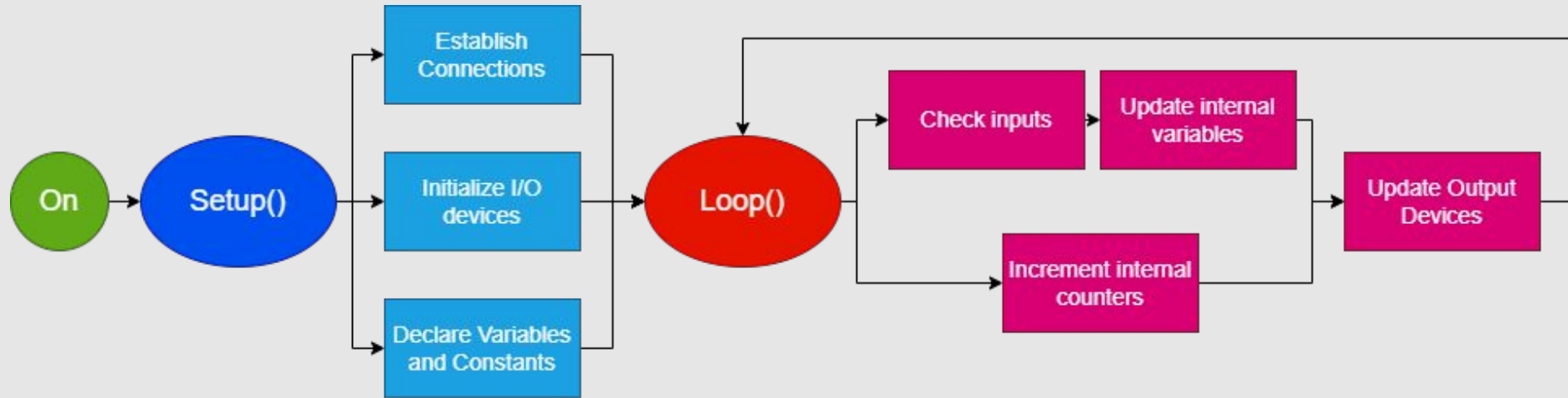
void loop() {
  // put your main code here, to run repeatedly:
}

1
Adafruit Feather nRF52840 Express, S140 6.1.1, Level 0 (Release), Serial on COM11
```

Setup() function runs once on power-up

Loop() function runs right after **Setup()** over and over until power-off or an error

Program Workflow



Getting Started

On your own

Visit [the Arduino website](#) to install the Arduino IDE. Select your operating system and follow installation instructions.

...OR code online with [Arduino Cloud](#)!

Right Now

We don't have that many Arduinos :(

So...

Visit [Tinkercad](#) to simulate Arduino circuits online!

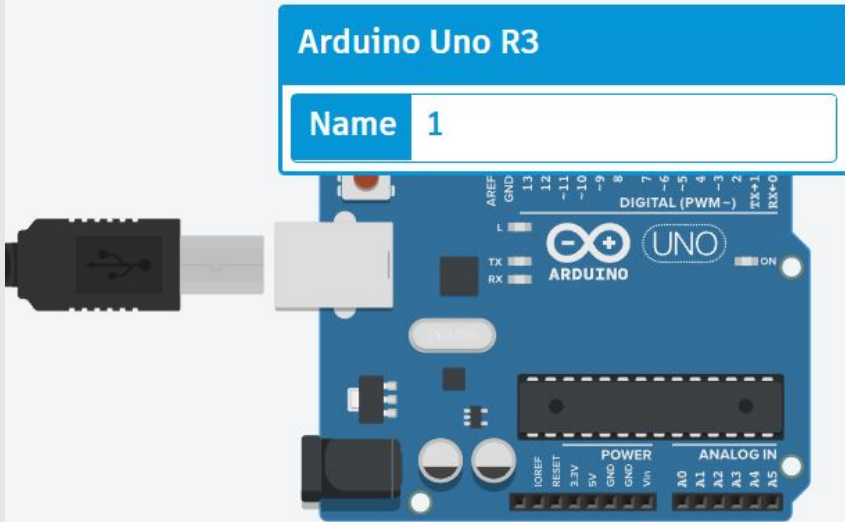


Programming Reference

Check out the arduino reference page to learn about all the functionalities provided by arduino.

<https://www.arduino.cc/reference/>

Project 1 - Blink



Text



1 (A

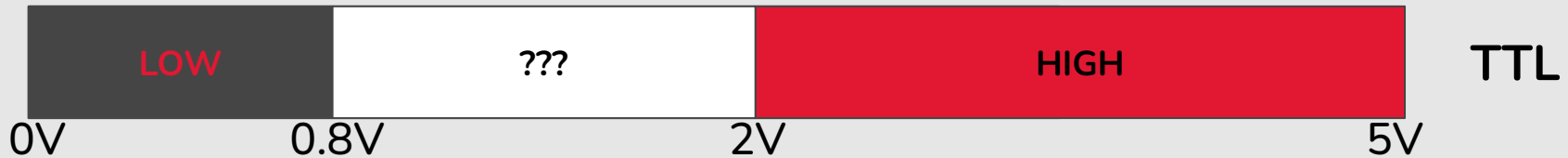
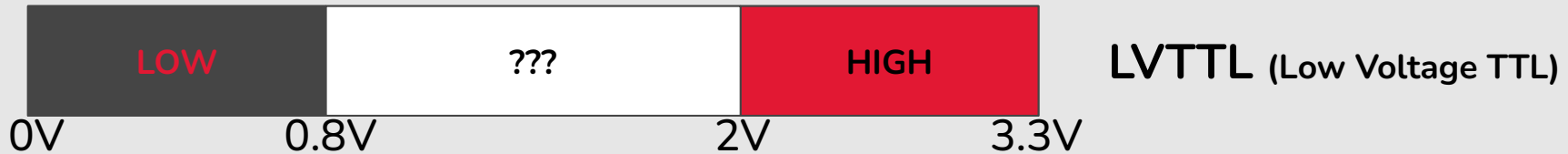
```
1 // C++ code
2 //
3 void setup()
4 {
5     pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10    digitalWrite(LED_BUILTIN, HIGH);
11    delay(1000); // Wait for 1000 millisecond(s)
12    digitalWrite(LED_BUILTIN, LOW);
13    delay(1000); // Wait for 1000 millisecond(s)
14 }
```

No need to write anything - the code should already be there!

Digital Signals

TTL Levels set the voltage ranges that devices agree correspond to digital **HIGH** or **LOW**.

Devices may use **different** TTL standards!



Digital Input/Output

digitalRead(pin)

pin = the digital pin # to read from

Returns: **HIGH** or **LOW** voltage seen at **pin**

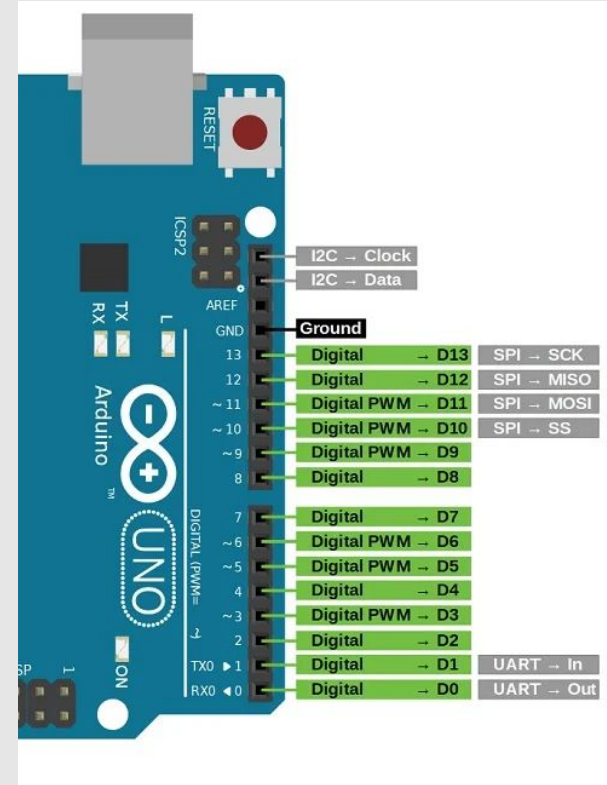
ex. `val = digitalRead(7);`

digitalWrite(pin,value)

pin = the digital pin # to write to

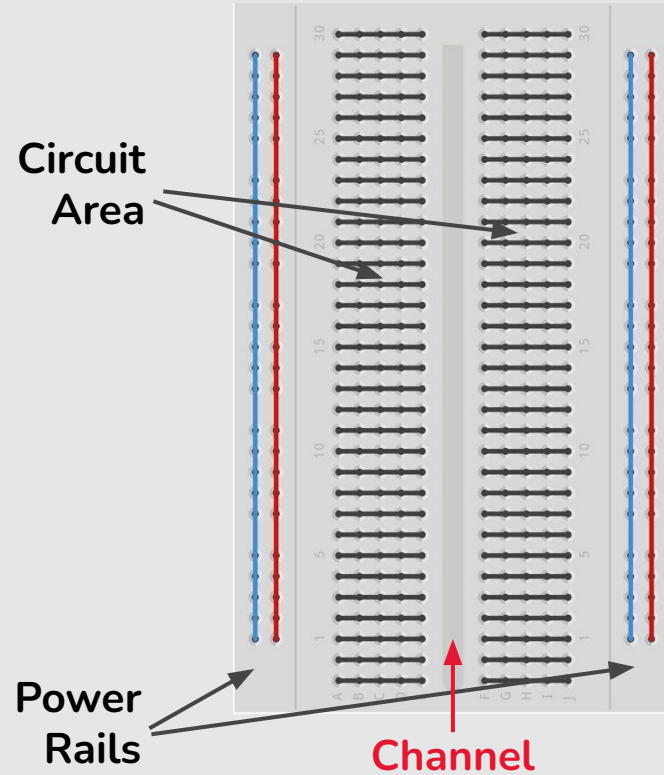
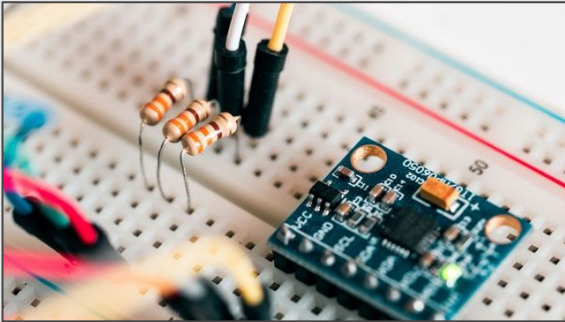
value = digital value to write to **pin**
(must be either **HIGH** or **LOW**)

ex. `digitalWrite(13, HIGH);`



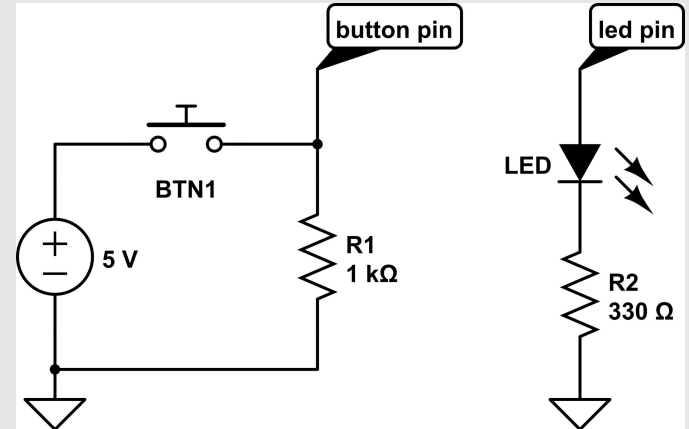
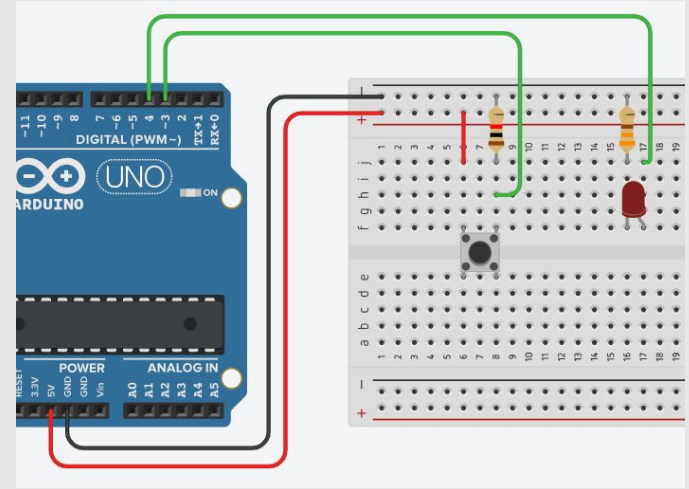
The Breadboard

- **Pre-connects** groups of headers that accept wires and leads
- Allows prototyping **without soldering!**



Project 2 - LED Button

```
1 int buttonPin = 3;
2 int ledPin = 4;
3
4 void setup() {
5     pinMode(buttonPin, INPUT);
6     pinMode(ledPin, OUTPUT);
7 }
8
9 void loop() {
10     bool buttonStatus;
11     if(digitalRead(buttonPin) == HIGH)
12         buttonStatus = 1;
13     else
14         buttonStatus = 0;
15     digitalWrite(ledPin, buttonStatus);
16     delay(10);
17 }
```





Code



Start Simulation



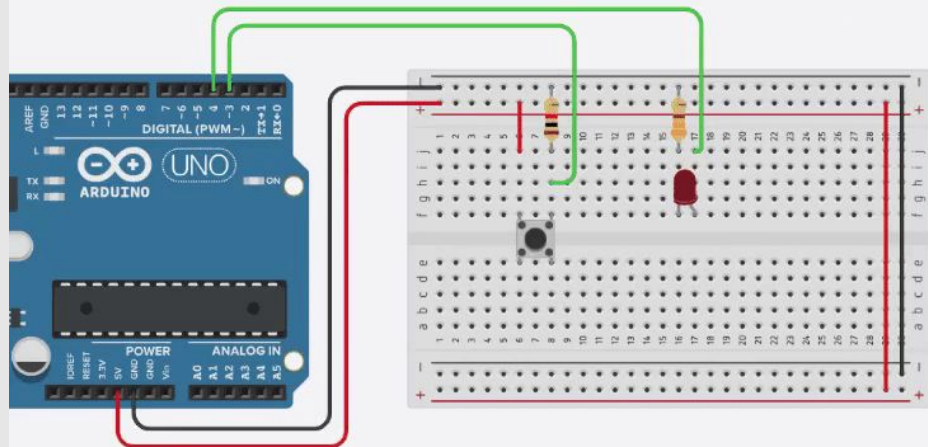
Send To



Text



1 (Arduino Uno R3)



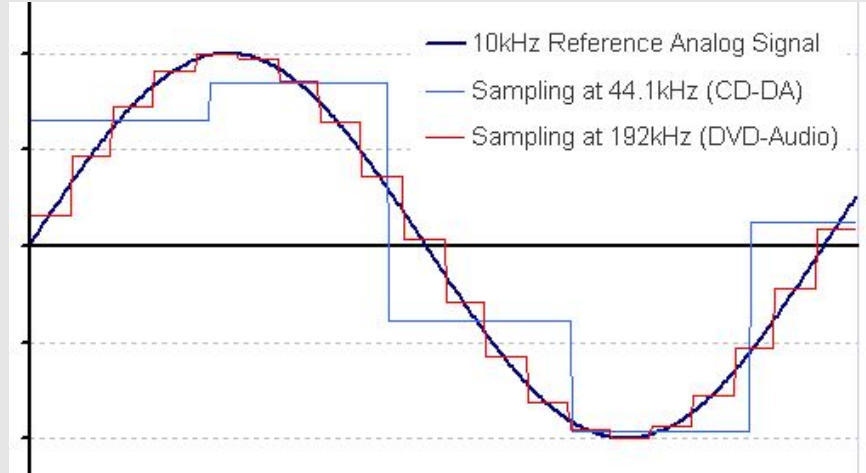
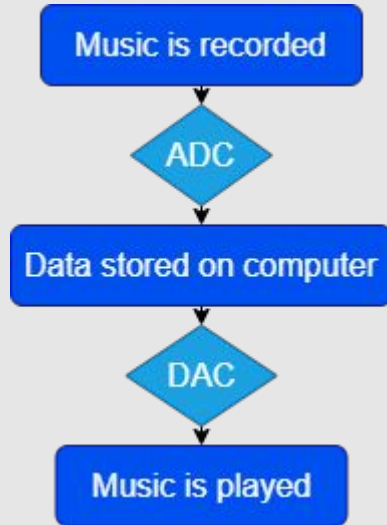
```
1 int buttonPin = 3;
2 int ledPin = 4;
3
4 void setup() {
5   pinMode(buttonPin, INPUT);
6   pinMode(ledPin, OUTPUT);
7 }
8
9 void loop() {
10  bool buttonStatus;
11  if(digitalRead(buttonPin) == HIGH)
12    buttonStatus = 1;
13  else
14    buttonStatus = 0;
15  digitalWrite(ledPin, buttonStatus);
16  delay(10);
17 }
18
19
20
21
```



Serial Monitor

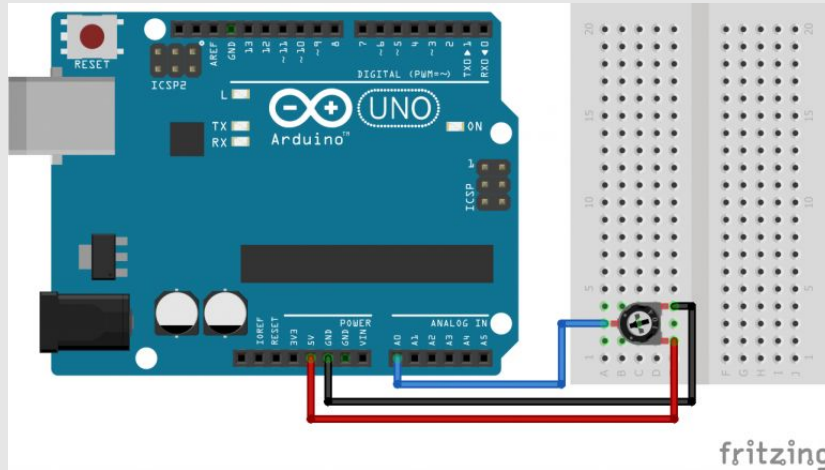
Analog Signals

- Continuous and organic - “real life”
- Turned into digital data with an **Analog-Digital Converter (ADC)**



The ADC

- It converts voltage measurements into a 10-bit digital value between 0 and 1023
- The Arduino Uno has six ADCs at pins A0-A5
- The Arduino Uno does **not** have any DACs

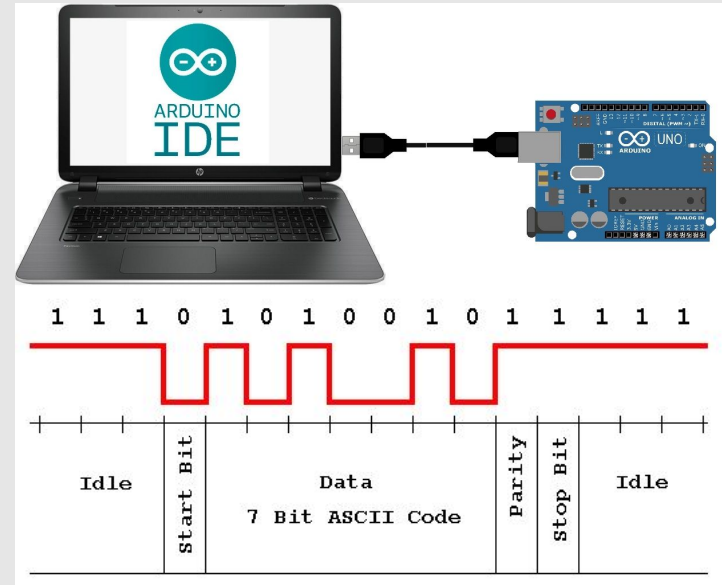


```
void setup() {  
    int ADCvalue = 0;  
    // initialize serial communication at 9600 baud rate  
    Serial.begin(9600);  
}  
  
void loop() {  
    // Read the analog value from pin A0  
    int ADCvalue = analogRead(A0);  
    // print the value at serial monitor  
    Serial.println(ADCvalue);  
    delay(100);      // delay in between reads for stability  
}
```

Serial Communication

- Arduino uses the Serial/UART communication protocol
- The Arduino's Serial Library lets the arduino “talk” to a computer through the Arduino IDE serial monitor!
- We use commands to control what the Arduino sends

Serial refers to data transferred one bit at a time



Serial Commands

Serial.begin(speed)

speed = bitrate of usb communication (use 9600)

Serial.print(string)

string = text or variable to write to serial monitor

Serial.println(string)

string = text or variable to write to serial monitor **WITH** a new line

 Code

 Start Simulation

Send To

Text



Start / stop simulation

no R3) ▾

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   Serial.print("Hi there!");  
7   delay(500);  
8 }
```

 Serial Monitor

Send

Clear



 Code

 Start Simulation

Send To

Text



Start / stop simulation

no R3) ▾

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4  
5 void loop() {  
6   Serial.println("Hi there!");  
7   delay(500);  
8 }
```

 Serial Monitor

Send

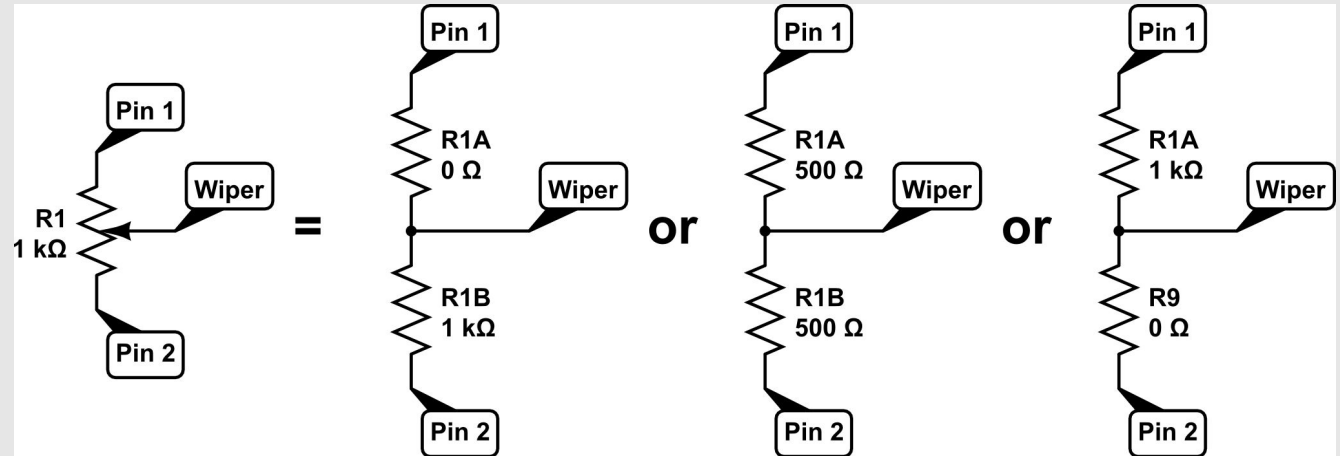
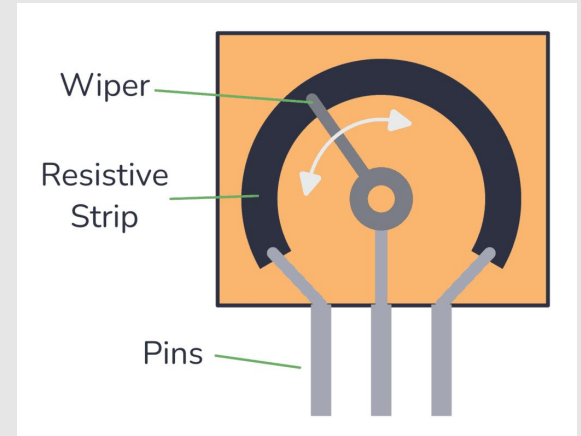
Clear



Potentiometers

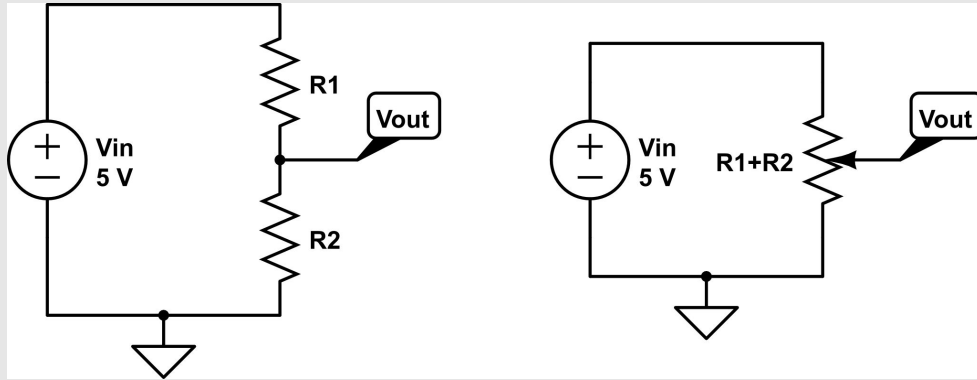
These **variable resistor dividers** can change the center pin (wiper) voltage as you turn the knob!

A pot's written **value** is the resistance between the outer pins.



Dividing Voltages

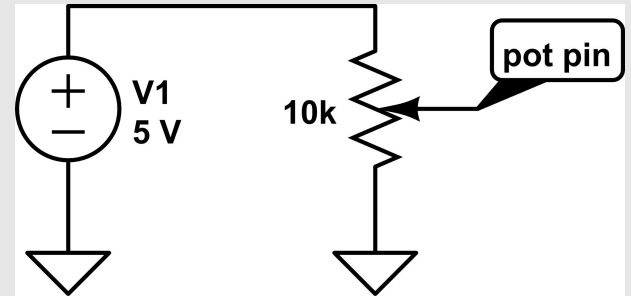
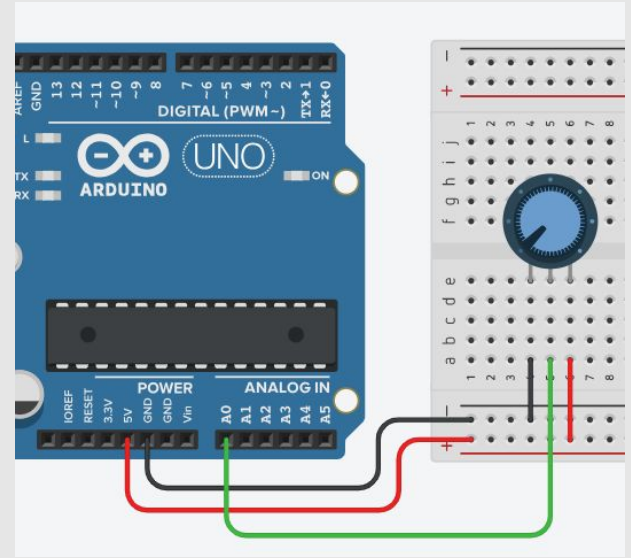
A main use of pots is in **voltage dividers**. These use the ratio between two resistances to get a fraction of a supplied voltage.



$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Project 3 - Serial Pot

```
1 int potPin = A0;
2
3 void setup() {
4     Serial.begin(9600);
5     pinMode(potPin, INPUT);
6 }
7
8 void loop() {
9     int potVal = analogRead(potPin);
10    Serial.print("Pot Value: ");
11    Serial.println(potVal);
12    delay(10);
13 }
```





Code



Start Simulation



Send To



Start / stop simulation

no R3)

Text

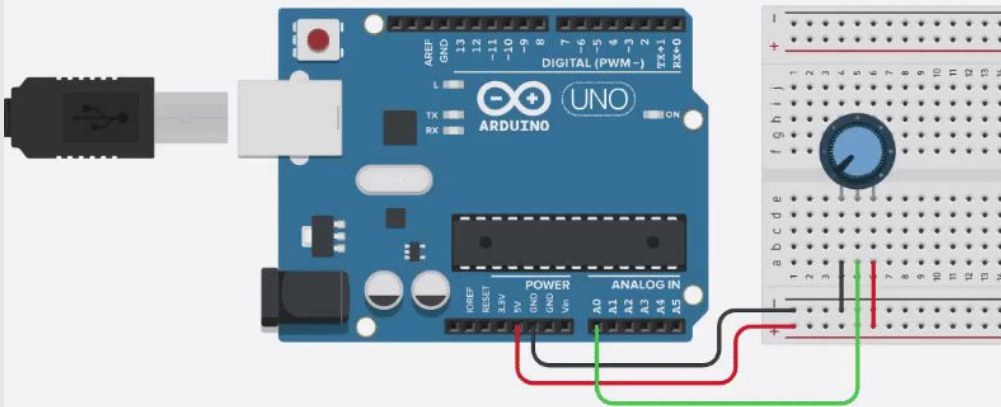
```
1 int potPin = A0;
2
3 void setup() {
4   Serial.begin(9600);
5   pinMode(potPin, INPUT);
6 }
7
8 void loop() {
9   int potVal = analogRead(potPin);
10  Serial.print("Pot Value: ");
11  Serial.println(potVal);
12  delay(10);
13 }
```



Serial Monitor

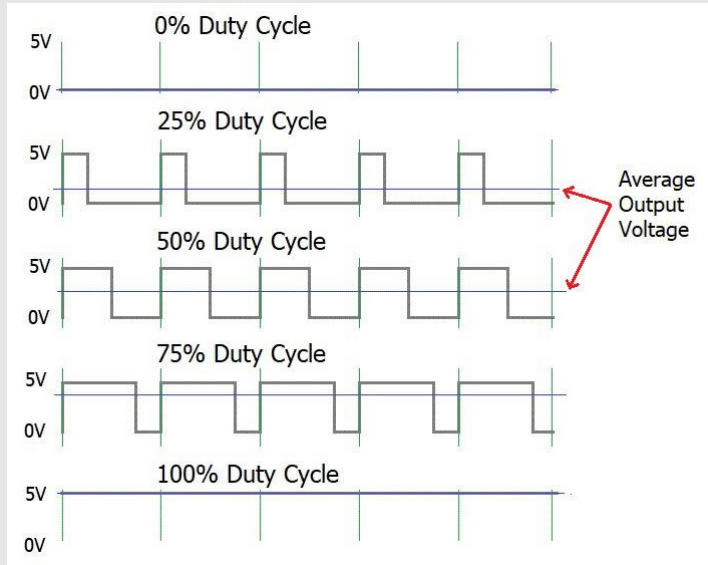
Send

Clear



Pulse Width Modulation (PWM)

PWM allows digital pins to *effectively* output voltages between **HIGH** and **LOW** by rapidly toggling the pin on and off at a set **duty cycle**.

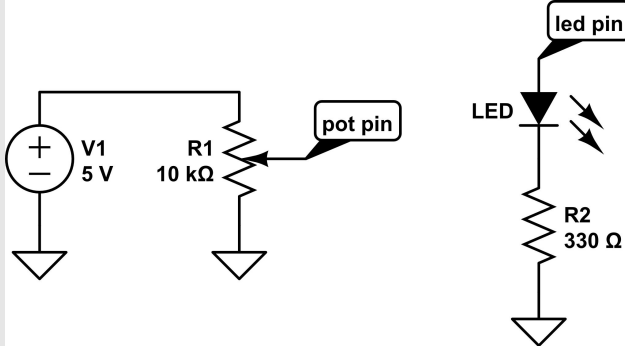
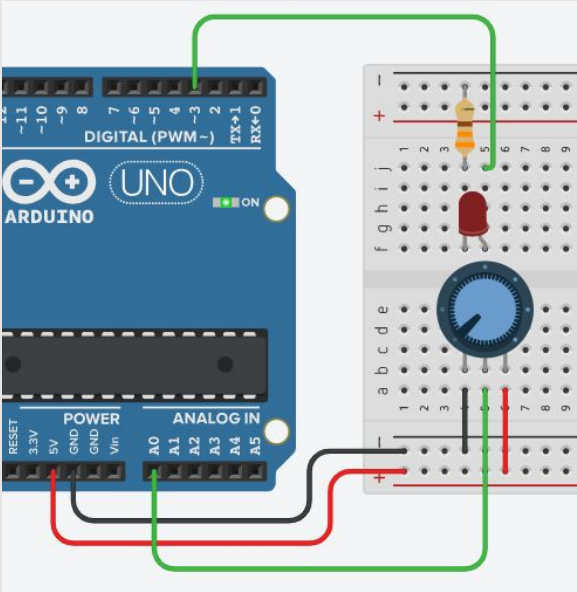


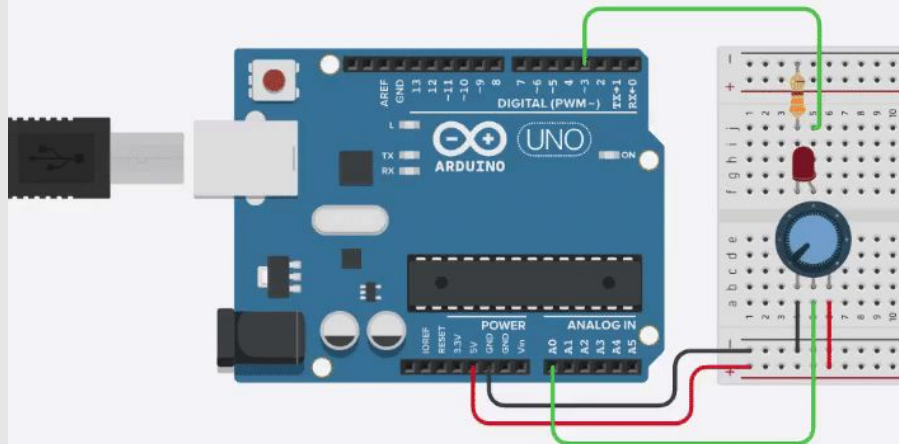
Duty Cycle is the fraction of time for which the pin is on.

$$D = \frac{T_{on}}{T_{total}}$$

$$V_{eq} = DV_{max}$$


```
1 int potPin = A0;
2 int ledPin = 3;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(potPin, INPUT);
7     pinMode(ledPin, OUTPUT);
8 }
9
10 void loop() {
11     // getting pot value
12     int potVal = analogRead(potPin);
13     Serial.print("Pot Value: ");
14     Serial.println(potVal);
15
16     // converting to LED value
17     int ledVal = map(potVal, 0, 1023, 0, 255);
18     analogWrite(ledPin, ledVal);
19     delay(10);
20 }
```





Text



Start / stop simulation

no R3)

```
1 int potPin = A0;
2 int ledPin = 3;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(potPin, INPUT);
7     pinMode(ledPin, OUTPUT);
8 }
9
10 void loop() {
11     // getting pot value
12     int potVal = analogRead(potPin);
13     Serial.print("Pot Value: ");
14     Serial.println(potVal);
15
16     // converting to LED value
17     int ledVal = map(potVal, 0, 1023, 0, 255);
18     analogWrite(ledPin, ledVal);
19     delay(10);
20 }
```

Serial Monitor

Send

Clear



Keep Exploring!

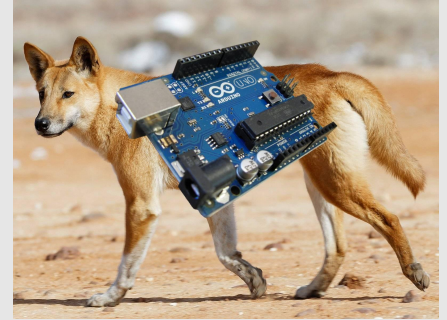
Try other projects such as...

- I2C/SPI Protocols
- ZigBee
- Motor control
- SD Card
- Ethernet
- GSM
- LCD Displays

Slides: <https://tinyurl.com/23k4jkx6>



Follow us on
Instagram!



Please give us your feedback!

<https://tinyurl.com/6eayw8r8>