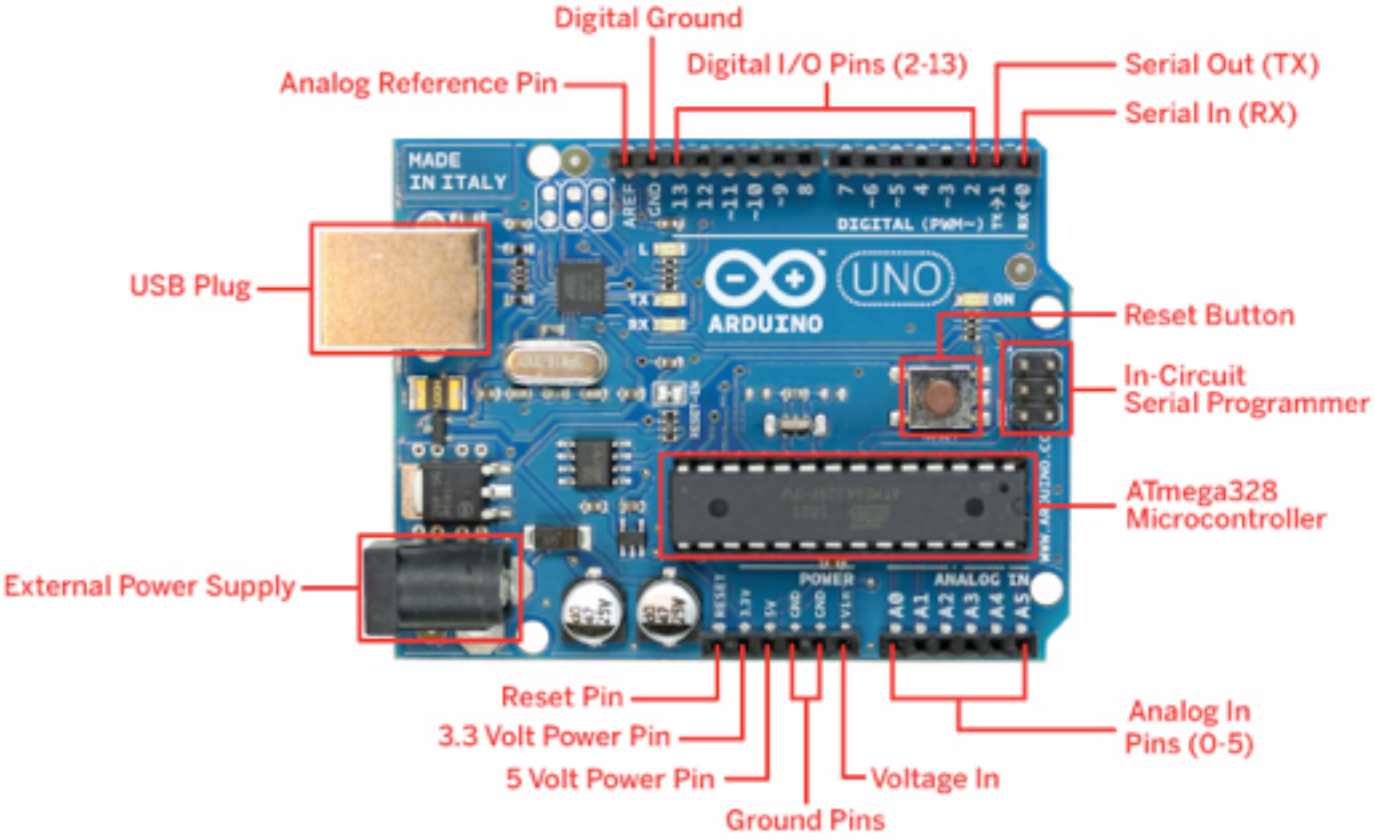
Pins

A *pin* provides an input or output through which the controller can communicate with components.

Digital I/O Pins

- two values: HIGH or LOW
- HIGH: 5 volts sent to the pin
- LOW: pin is at 0 volts



Analog Input Pins

- read analog values (0 - 1023) that represent 0 - 5 volts

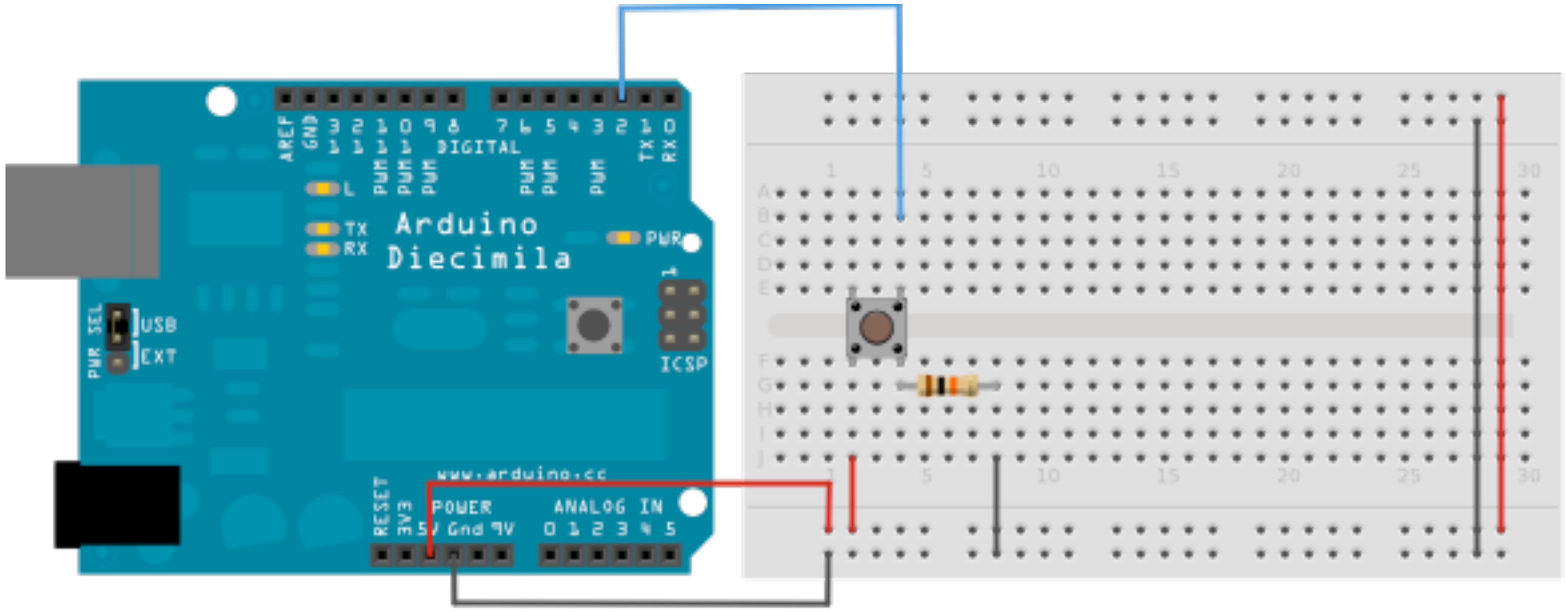
Analog Output (PWM)

- output values between 0 - 255
- uses particular digital pins
- not true analog - uses a technique called *pulse with modulation* to get analog results using a digital means

Digital Input

- we can read digital input from any digital pin as high or low values
- typical digital inputs are switches such as a button

```
digitalRead( pin );
```



Digital Output

- we can write a digital output from any digital pin as high or low values

```
digitalWrite( pin, value );
```

```
int buttonPin = 2;     // the number of the pushbutton pin
int ledPin =  13;      // the number of the LED pin
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop(){
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

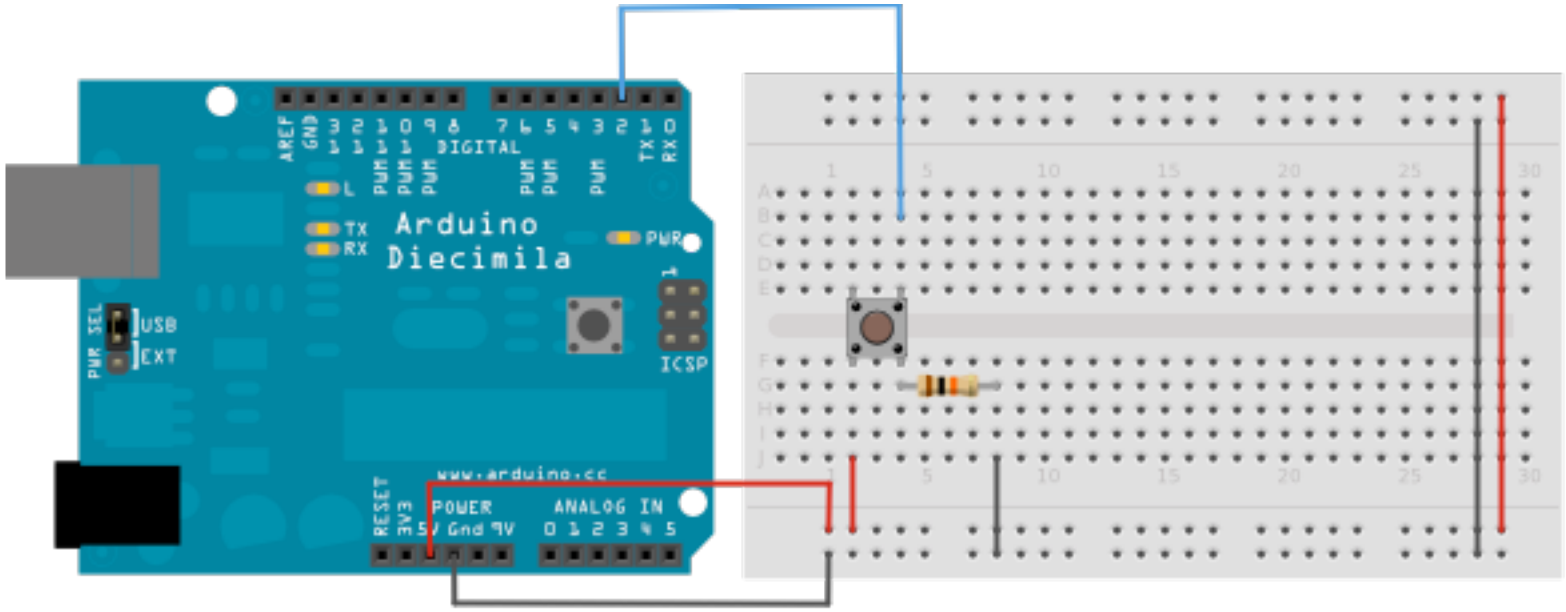
Sending a digital signal to Processing

- Serial data is sent and received as ***bytes***

- If we want to send an on / off state corresponding to HIGH / LOW , we can send a 0 / 1 values or ascii characters such as ‘H’ / ‘L’ with *Serial.write*

```
void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.write(45); // send a byte with the value 45
}
```

```
int buttonPin = 2;
int buttonState = 0;

void setup() {

  pinMode(buttonPin, INPUT);
  Serial.begin(9600);

}


void loop(){

  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH) {
    Serial.write(1);
  }else {
    Serial.write(0);
  }
}
```

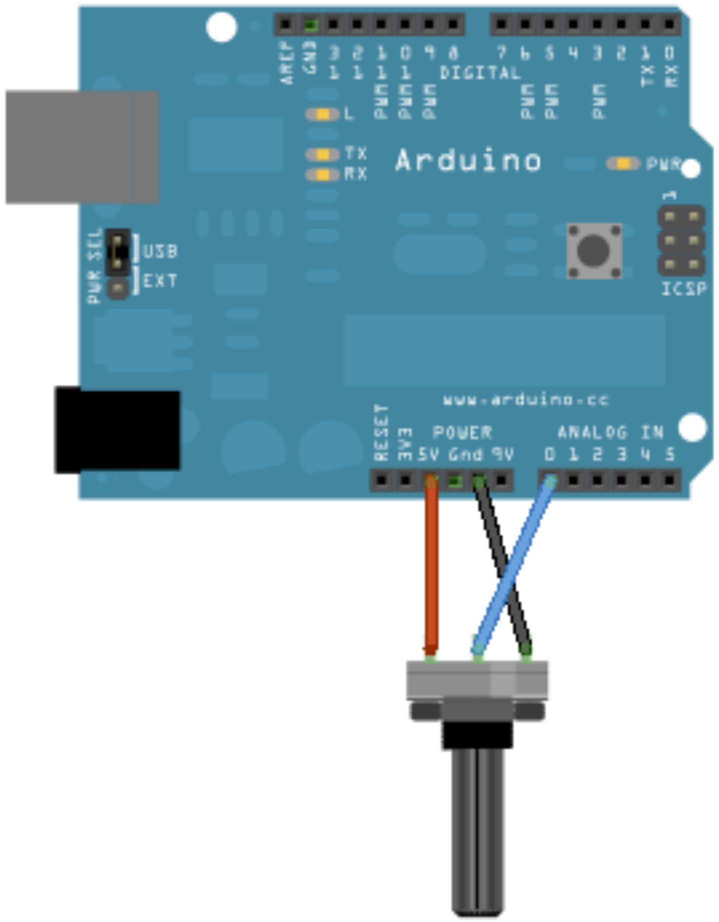


```
import processing.serial.*;

Serial myPort;
int buttonVal = 0;

void setup () {

  size(200, 200);

  println(Serial.list());
  myPort = new Serial(this, Serial.list()[5], 9600);

}


void draw () {

  if(buttonVal == 0){
     background(0);
  }else{
     background(255);
  }

}

void serialEvent(Serial myPort) {
 buttonVal =  myPort.read();
}
```

Analog Input

- we can read analog input from analog input pins
- values range form 0 to 1023

```
analogRead( pin );
```



```
int analogPin = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(analogPin);
  Serial.println(sensorValue);
  delay(1);
}
```

Analog Output

- we can write analog output on pwm pins
- on Arduino Uno pins 3,5,6,9,10,11
- values range from 0 - 255
- *analogWrite* function has nothing to do with the analog pins or the *analogRead* function.

```
analogWrite( pin, value );
```

```
int ledPin = 9;
int analogPin = 0;
int val = 0;

void setup(){
  pinMode(ledPin, OUTPUT);
}

void loop(){

  val = analogRead(analogPin); // read potentiometer
  int mapVal = map(val,0,1023,0,255);
  analogWrite(ledPin, mapVal);

}
```

Send analog data to Processing

- use *println* to send a string of data

```
int analogPin = 0;
int val = 0;

void setup((){
  Serial.begin(9600);
}

void loop(){

  val = analogRead(analogPin); // read potentiometer
  Serial.println(val);

}
```

```
import processing.serial.*;

Serial myPort;
int analogVal = 0;

void setup () {
  size(200, 200);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[5], 9600);
}

void draw () {
  background(0);

}

void serialEvent(Serial myPort) {

  String inString = myPort.readStringUntil('\n');

  if (inString != null) {
    println(inString);
    inString = trim(inString);
    analogVal = int(inString);
 }
}
```