

LAB 3.1: INPUT/OUTPUTS ON ARDUINO UNO BOARD

Onur Kilincceker (MSKU, Computer Engineering)

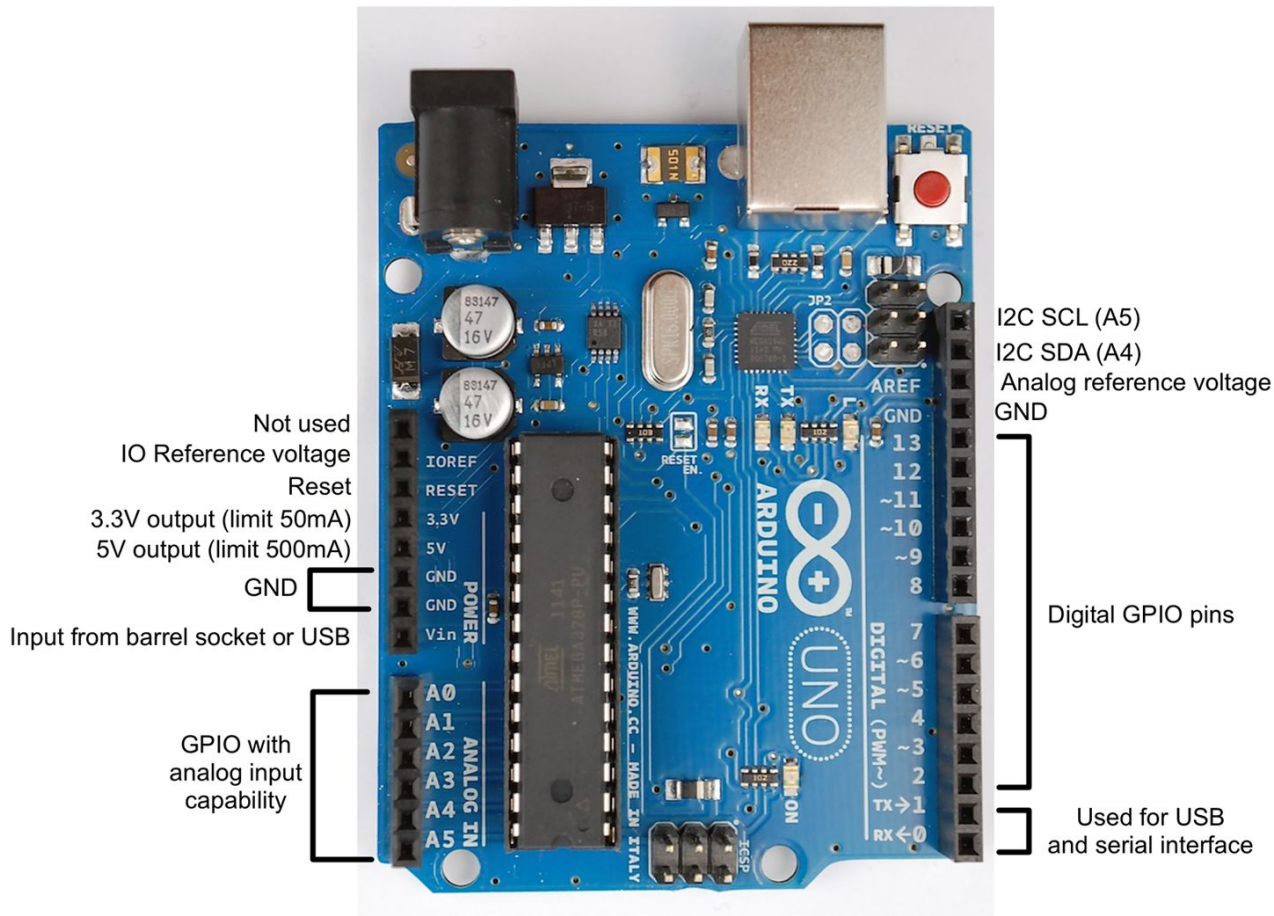
CREDITS

- <https://www.arduino.cc/>
- <https://www.simulide.com/p/home.html>
- <http://simonmonk.org/>

INPUT/OUTPUTS

- The Arduino is about physical computing, and that means attaching electronics to the Arduino board. So you need to understand how to use the various options for your connection pins.
- Outputs can be digital, which just means switched between being at 0V or at 5V, or analog, which allows you to set the voltage to any voltage between 0V and 5V
- Likewise, inputs can either be digital (for example, determining whether a button is pressed or not) or analog (such as from a light sensor)

ARDUINO UNO PINOUTS

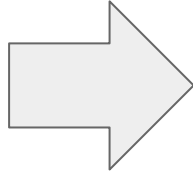


DEVELOPMENT



Arduino IDE

Programming



Simulation

SimulIDE

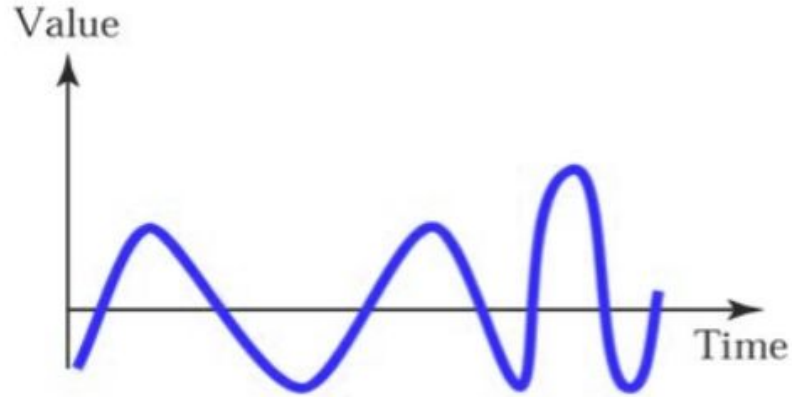
Real Time Electronic Circuit Simulator. With PIC, AVR and Arduino simulation.



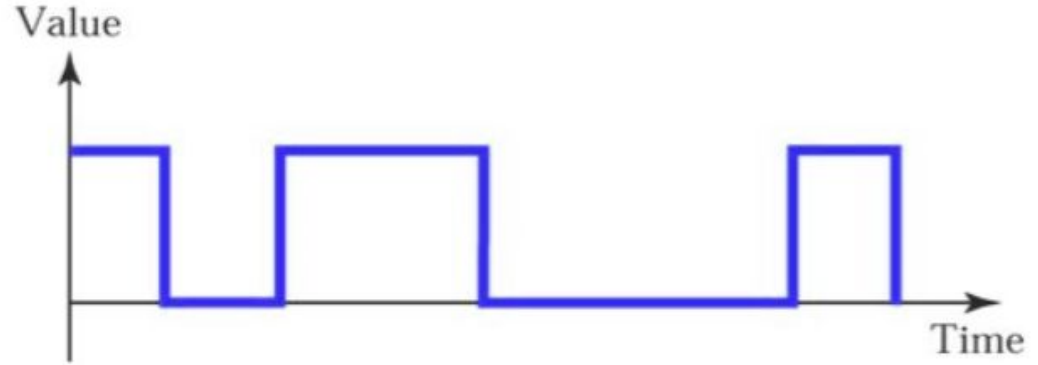
Realization



ANALOG VS DIGITAL SIGNALS



Analog signal



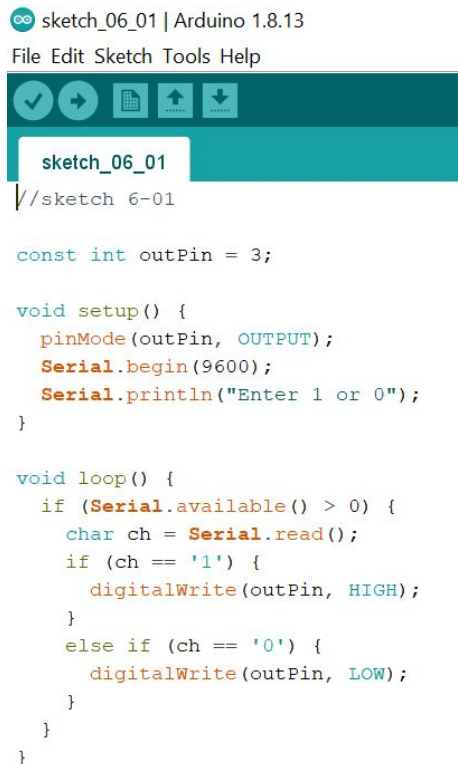
Digital signal

DIGITAL OUTPUTS

- In earlier week, you have made use of the LED attached to digital pin 13 of the Arduino board.
- Let's experiment with one of the other pins on the Arduino. You will use digital pin 4, and to see what is going on, you will fix some wire to your multimeter leads and attach them to your Arduino.

EXAMPLE: //SKETCH 6-01 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES

At the top of the sketch, you can see the command `pinMode` . You should use this command for every pin that you are using in a project so that Arduino can configure the electronics connected to that pin to be either an input or an output, as in the following example:
`pinMode(outPin, OUTPUT);`



```
sketch_06_01 | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_06_01
//sketch 6-01

const int outPin = 3;

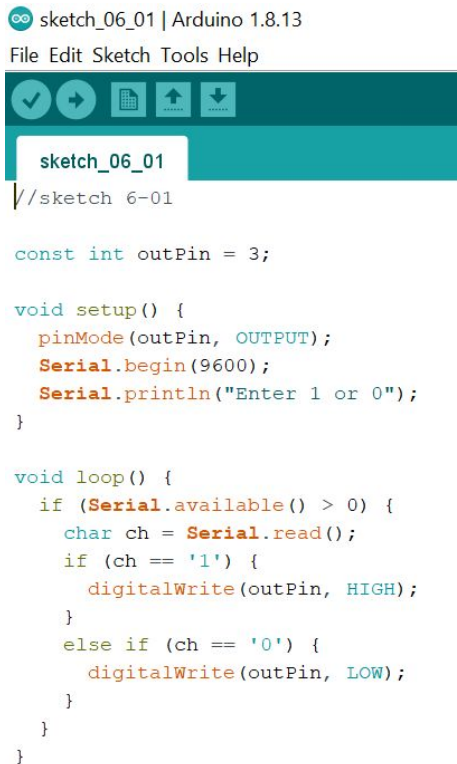
void setup() {
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter 1 or 0");
}

void loop() {
  if (Serial.available() > 0) {
    char ch = Serial.read();
    if (ch == '1') {
      digitalWrite(outPin, HIGH);
    }
    else if (ch == '0') {
      digitalWrite(outPin, LOW);
    }
  }
}
```

As you might have guessed, `pinMode` is a built-in function. Its first argument is the pin number in question (an `int`), and the second argument is the mode, which must be either `INPUT`, `INPUT_PULLUP` or `OUTPUT` . Note that the mode name must be all uppercase.

EXAMPLE: //SKETCH 6-01 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES

This loop waits for a command of either 1 or 0 to come from the Serial Monitor on your computer. If it's a 1, then pin 3 will be turned on; otherwise, it will be turned off.



```
sketch_06_01 | Arduino 1.8.13
File Edit Sketch Tools Help

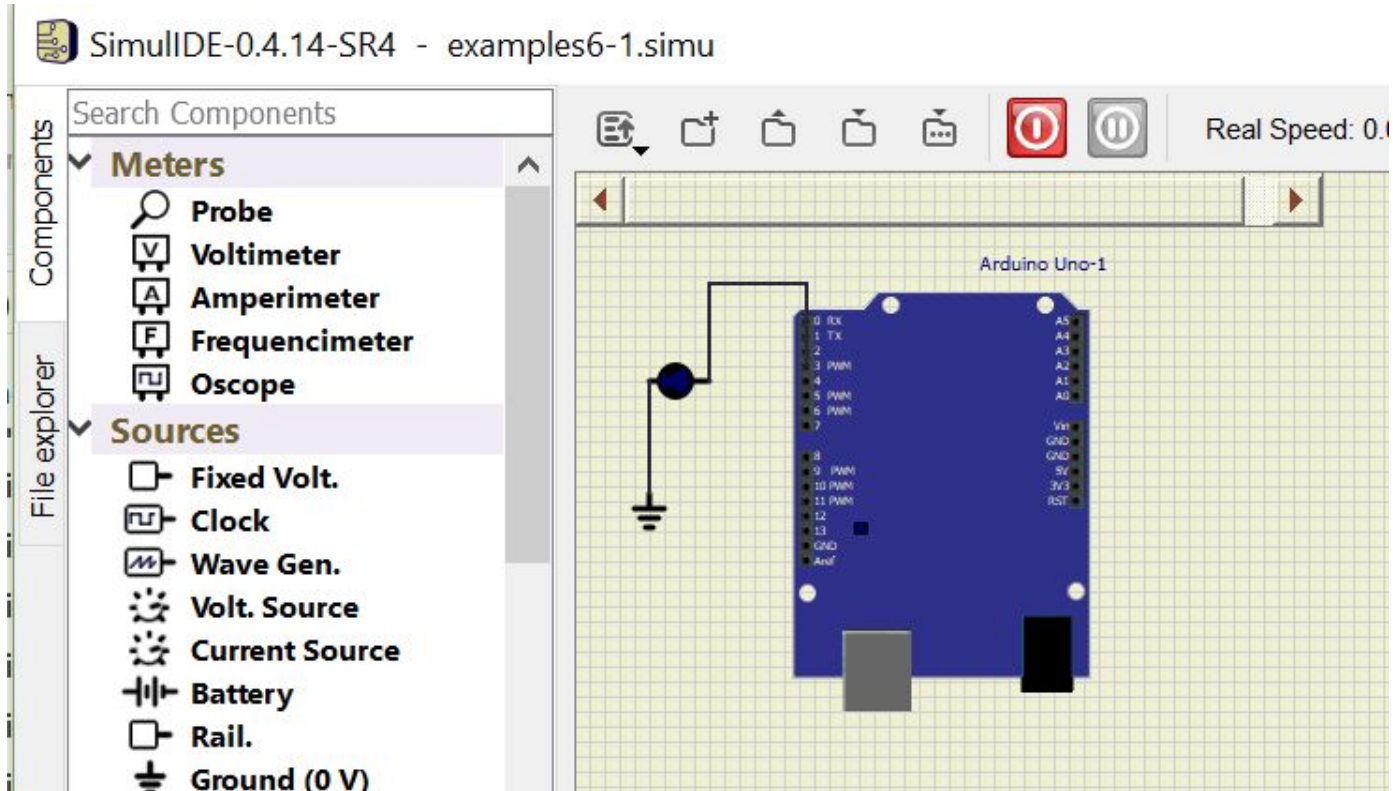
sketch_06_01
//sketch 6-01

const int outPin = 3;

void setup() {
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter 1 or 0");
}

void loop() {
  if (Serial.available() > 0) {
    char ch = Serial.read();
    if (ch == '1') {
      digitalWrite(outPin, HIGH);
    }
    else if (ch == '0') {
      digitalWrite(outPin, LOW);
    }
  }
}
```

EXAMPLE: //SKETCH 6-01 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



DIGITAL INPUTS

- The most common use of digital inputs is to detect when a switch has been closed. A digital input can either be on or off. If the voltage at the input is less than 2.5V (halfway to 5V), it will be 0 (off), and if it is above 2.5V, it will be 1 (on).

EXAMPLE: //SKETCH 6-02 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



```
sketch_06_02 | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_06_02 $
//sketch 06-02

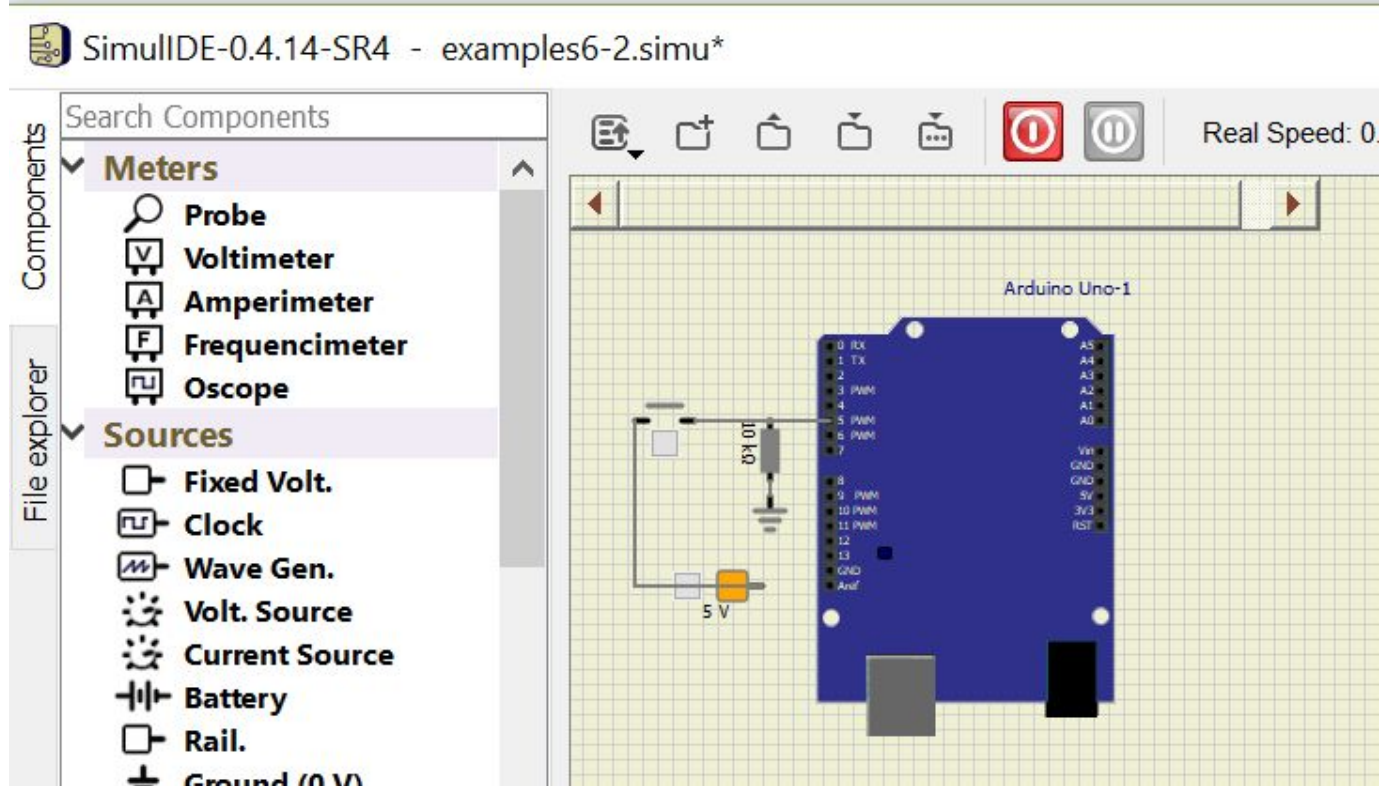
const int inputPin = 5;

void setup() {
  pinMode(inputPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  int reading = digitalRead(inputPin);
  Serial.println(reading);
  delay(1000);
}
```

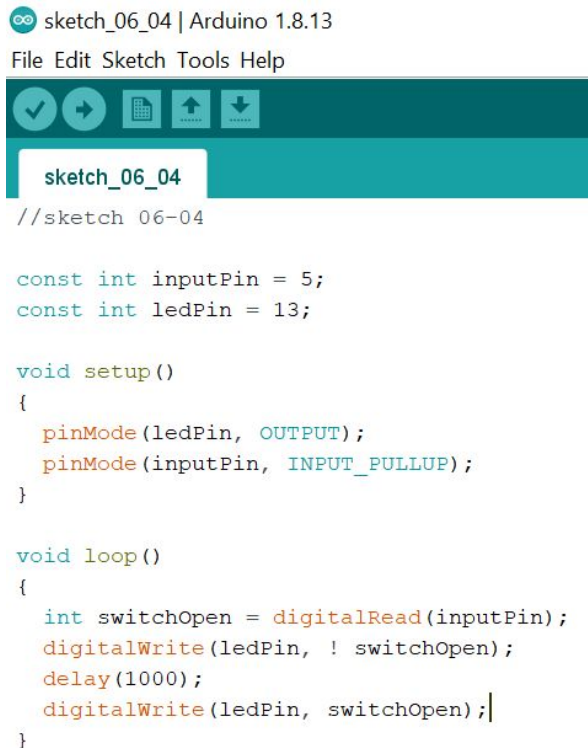
As with using an output, you need to tell the Arduino in the setup function that you are going to use a pin as an input. You get the value of a digital input using the digitalRead function. This returns 0 or 1.

EXAMPLE: //SKETCH 6-02 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



EXAMPLE: //SKETCH 6-04 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES

Looking at the loop function of sketch 6-04, the function reads the digital input and assigns its value to a variable switchOpen . This is a 0 if the button is pressed and a 1 if it isn't (remember that the pin is pulled up to 1 when the button is not pressed).



```
sketch_06_04 | Arduino 1.8.13
File Edit Sketch Tools Help

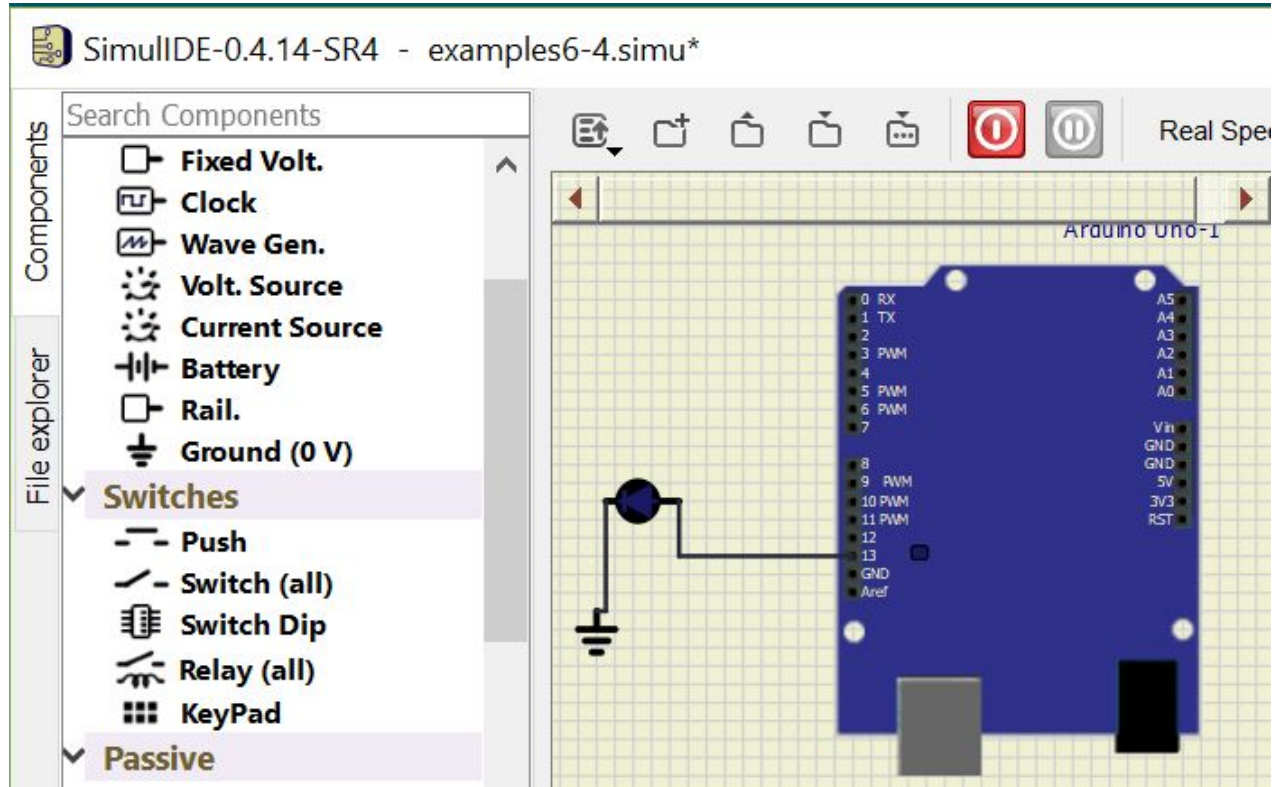
sketch_06_04
//sketch 06-04

const int inputPin = 5;
const int ledPin = 13;

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

void loop()
{
  int switchOpen = digitalRead(inputPin);
  digitalWrite(ledPin, ! switchOpen);
  delay(1000);
  digitalWrite(ledPin, switchOpen);
}
```

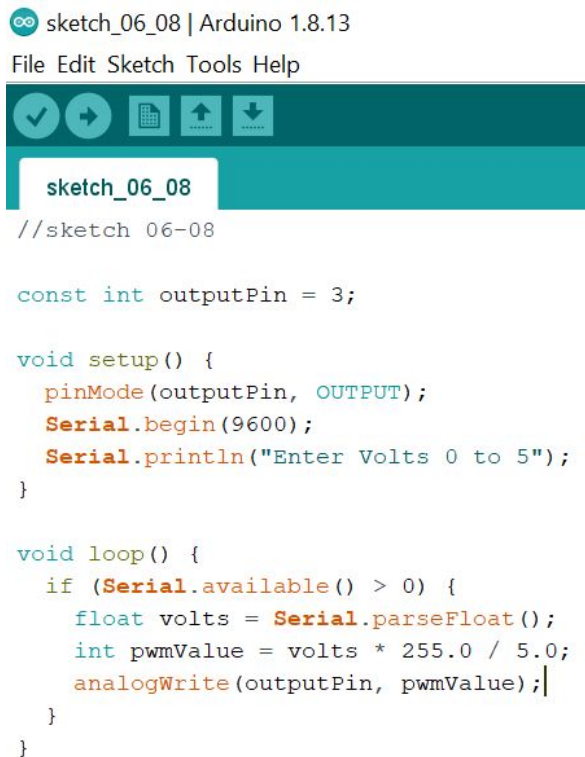
EXAMPLE: //SKETCH 6-04 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



ANALOG OUTPUTS

- A few of the digital pins—namely digital pins 3, 5, 6, 9, 10, and 11—can provide variable output other than just 5V or nothing. These are the pins on the board with a ~ or “PWM” next to them. PWM stands for Pulse Width Modulation, which refers to the means of controlling the amount of power at the output. It does so by rapidly turning the output on and off.

EXAMPLE: //SKETCH 6-08 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



```
sketch_06_08 | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_06_08
//sketch 06-08

const int outputPin = 3;

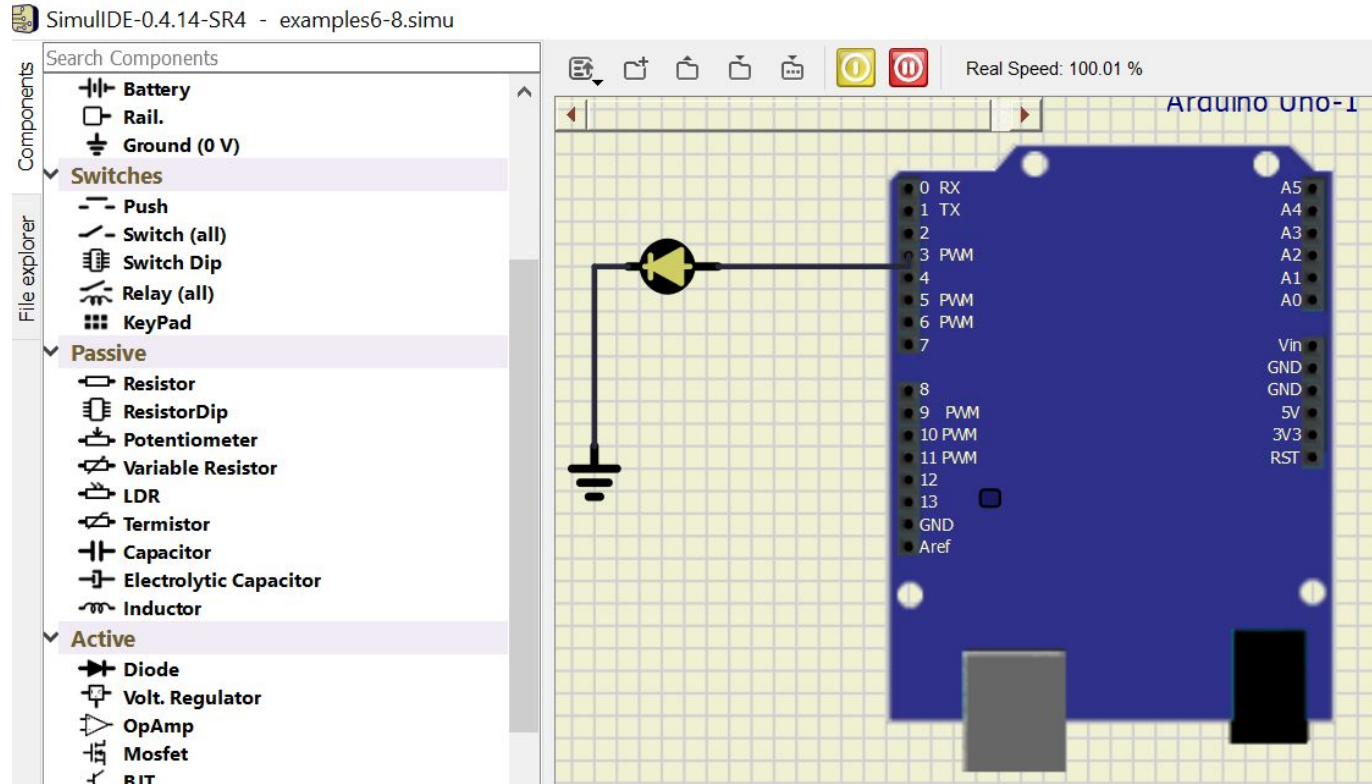
void setup() {
  pinMode(outputPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Enter Volts 0 to 5");
}

void loop() {
  if (Serial.available() > 0) {
    float volts = Serial.parseFloat();
    int pwmValue = volts * 255.0 / 5.0;
    analogWrite(outputPin, pwmValue);
  }
}
```

The program determines the value of PWM output between 0 and 255 by multiplying the desired voltage (0 to 5) by 255/5. (Readers may wish to refer to Wikipedia for a fuller description of PWM.)

You can set the value of the output by using the function `analogWrite`, which requires an output value between 0 and 255, where 0 is off and 255 is full power. This is actually a great way to control the brightness of an LED.

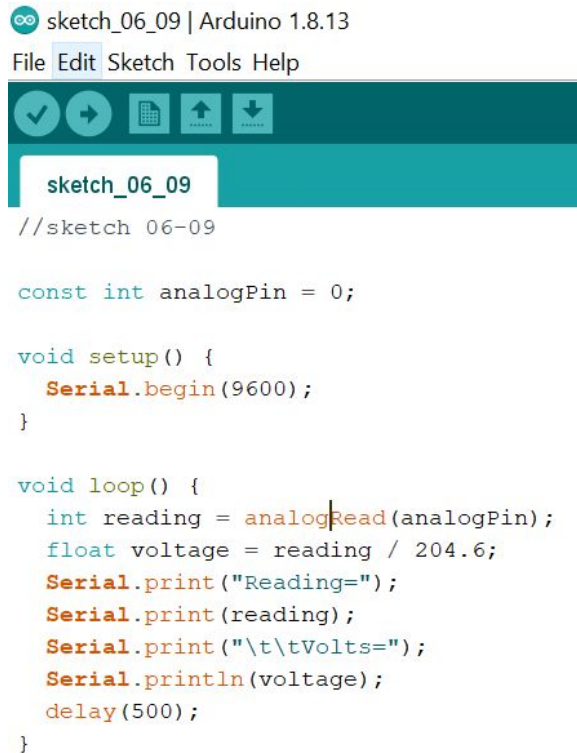
EXAMPLE: //SKETCH 6-08 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



ANALOG INPUT

- Digital inputs just give you an on/off answer as to what is happening at a particular pin on the Arduino board. Analog inputs, however, give you a value between 0 and 1023 depending on the voltage at the analog input pin.

EXAMPLE: //SKETCH 6-09 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



```
sketch_06_09 | Arduino 1.8.13
File Edit Sketch Tools Help

sketch_06_09
//sketch 06-09

const int analogPin = 0;

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

void loop() {
  int reading = analogRead(analogPin);
  float voltage = reading / 204.6;
  Serial.print("Reading=");
  Serial.print(reading);
  Serial.print("\t\tVolts=");
  Serial.println(voltage);
  delay(500);
}
```

When you run this sketch, you will notice that the readings change quite a bit. As with the digital inputs, this is because the input is floating.

Take one end of the wire and put it into a GND socket so that A0 is connected to GND.

Your readings should now stay at 0. Move the end of the lead that was in GND and put it into 5V and you should get a reading of around 1023, which is the maximum reading.

So, if you were to connect A0 to the 3.3V socket on the Arduino board, the Arduino voltmeter should tell you that you have about 3.3V.

EXAMPLE: //SKETCH 6-09 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES



```
//sketch 06-09

const int analogPin = 0;

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

void loop() {
  int reading = analogRead(analogPin);
  float voltage = reading / 204.6;
  Serial.print("Reading=");
  Serial.print(reading);
  Serial.print("\t\tVolts=");
  Serial.println(voltage);
  delay(500);
}
```

The `analogRead()` command converts the input voltage range, 0 to 5 volts, to a digital value between 0 and 1023. This is done by a circuit inside the microcontroller called an *analog-to-digital converter* or *ADC*.

EXAMPLE: //SKETCH 6-09 → SIMON MONK - PROGRAMMING ARDUINO _ GETTING STARTED WITH SKETCHES

