

ELECTRONICS 1

ELECTRONICS FOR INTERACTIVE MEDIA DESIGN
LESSON 2

EMMA PARESCHI

FROM THE LAST TIME

- Basic of electronics: voltage, resistance and current; ohm's law and power law.
- Tools: Multimeter to measure voltage, resistance and continuity.
- Basic of schematic and use of breadboard.

ARDUINO

“Physical Computing is about prototyping with electronics, turning sensors, actuators and microcontrollers into materials for designers and artists.”

“It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller.”

Massimo Banzi, Tinker.it & Arduino Co-Founder

HARDWARE



SOFTWARE



COMMUNITY

[Home](#)
[Buy](#)
[Download](#)
[Products](#)
[Learning](#)
[Forum](#)
[Support](#)
[Blog](#)

[Alerts](#)
[Unread Posts](#)
[Updated Topics](#)

[vulture Forum](#)

Using Arduino

Installation & Troubleshooting

For problems with Arduino itself, NOT your project

Last post: Today at 07:20 am by "mcd", executable file n., by Tupsing

67,313

Posts

15,896

Topics

Project Guidance

Advice on general approaches or feasibility

Last post: Today at 09:30 am by Arduino Arduino 12803 by thummer100

328,263

Posts

45,241

Topics

Programming Questions

Understanding the language, error messages, etc.

Last post: Today at 08:54 am by Problem with EEPROM... by ylls_618

455,139

Posts

55,397

Topics

General Electronics

Resistors, capacitors, breadboards, soldering, etc.

Last post: Today at 09:30 am by Arduino the right chain... by devrock

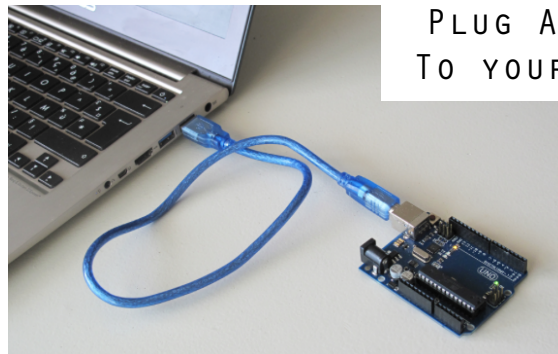
159,694

Posts

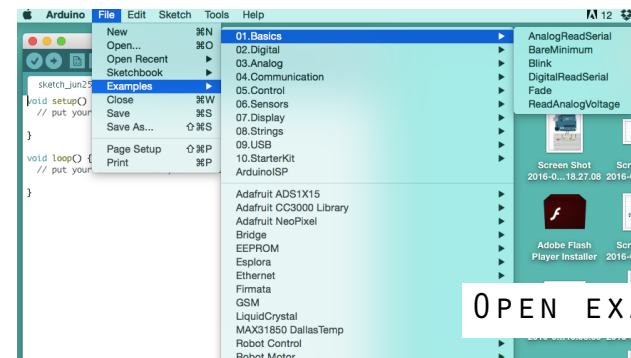
15,984

Topics

BUT...CHECK IF THE ENVIRONMENT WORKS



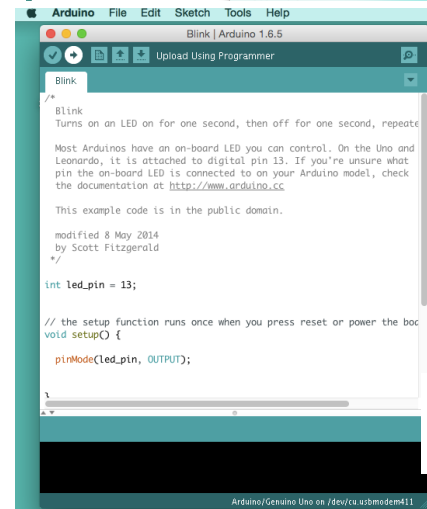
PLUG ARDUINO
TO YOUR LAPTOP



OPEN EXAMPLE BLINK



SELECT THE PORT
AND THE BOARD

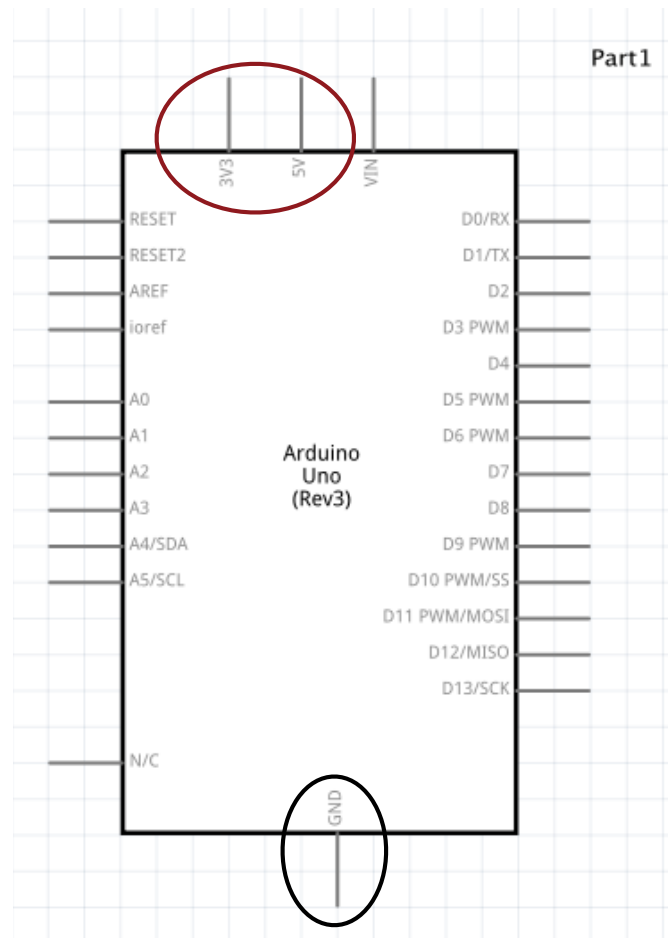


UPLOAD THE CODE
TO

IF IT WORKS, THE LED EMBEDDED ON THE BOARD BLINKS

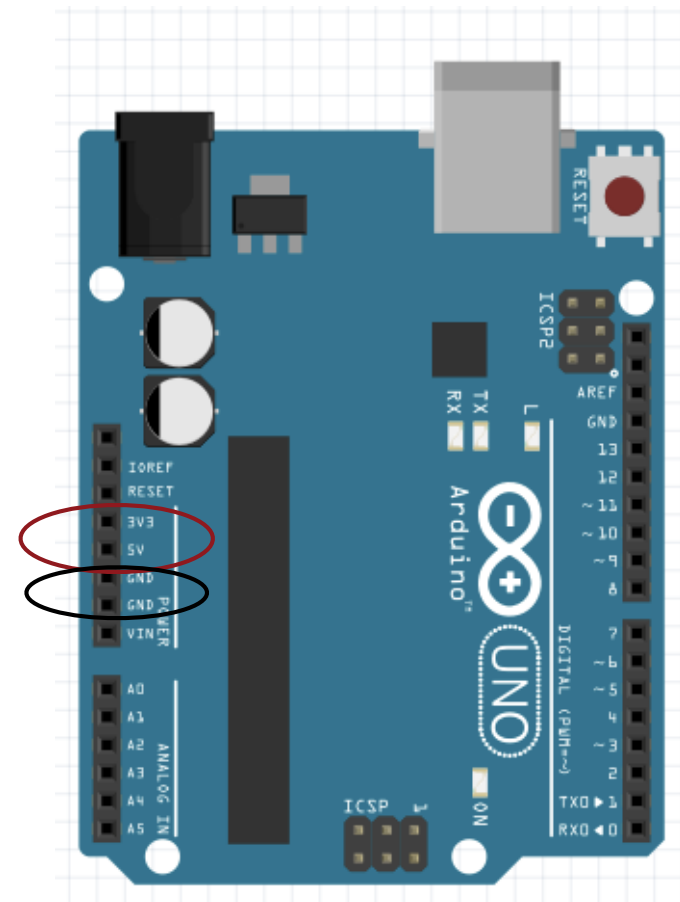
ARDUINO - POWER PINS

POWER PINS

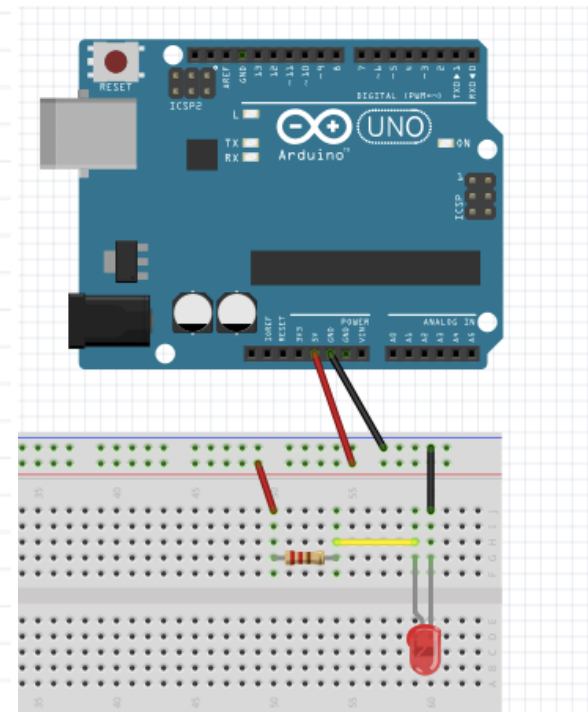
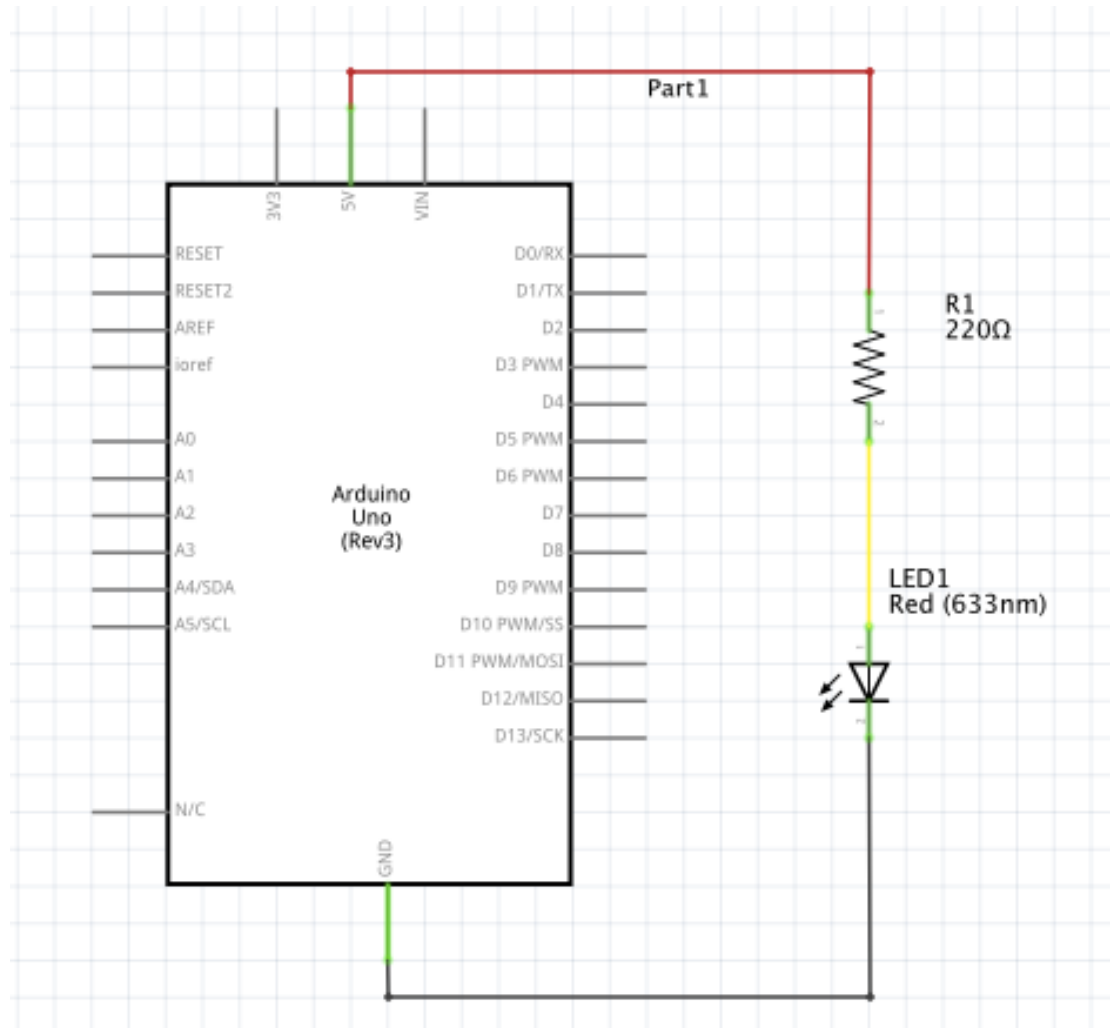


5V / 3.3V:
POWER

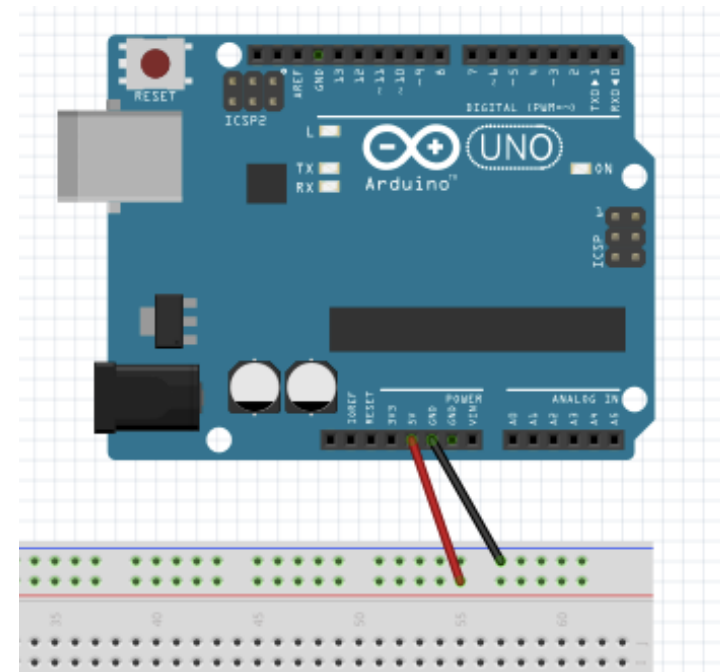
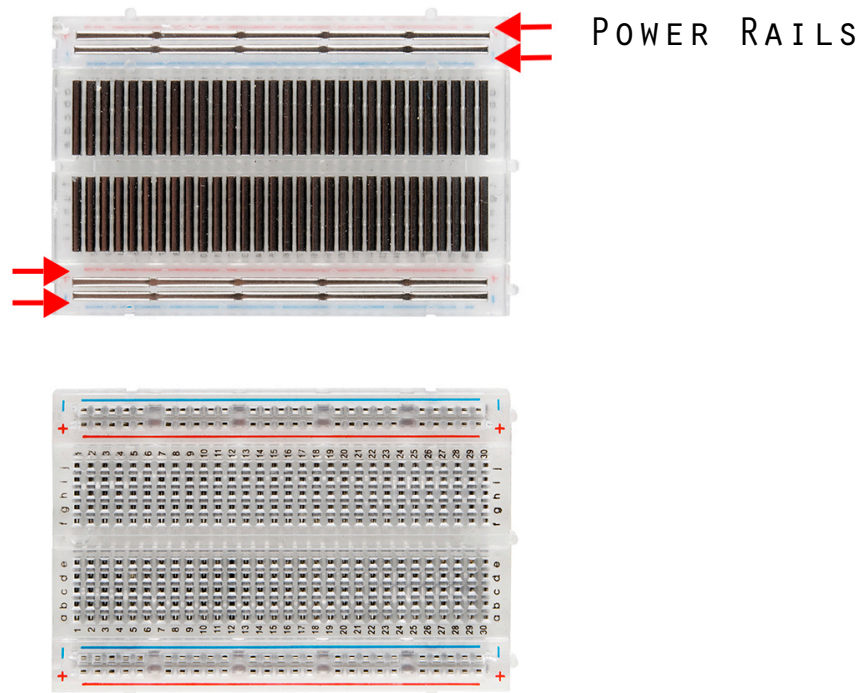
GND:
GROUND PIN



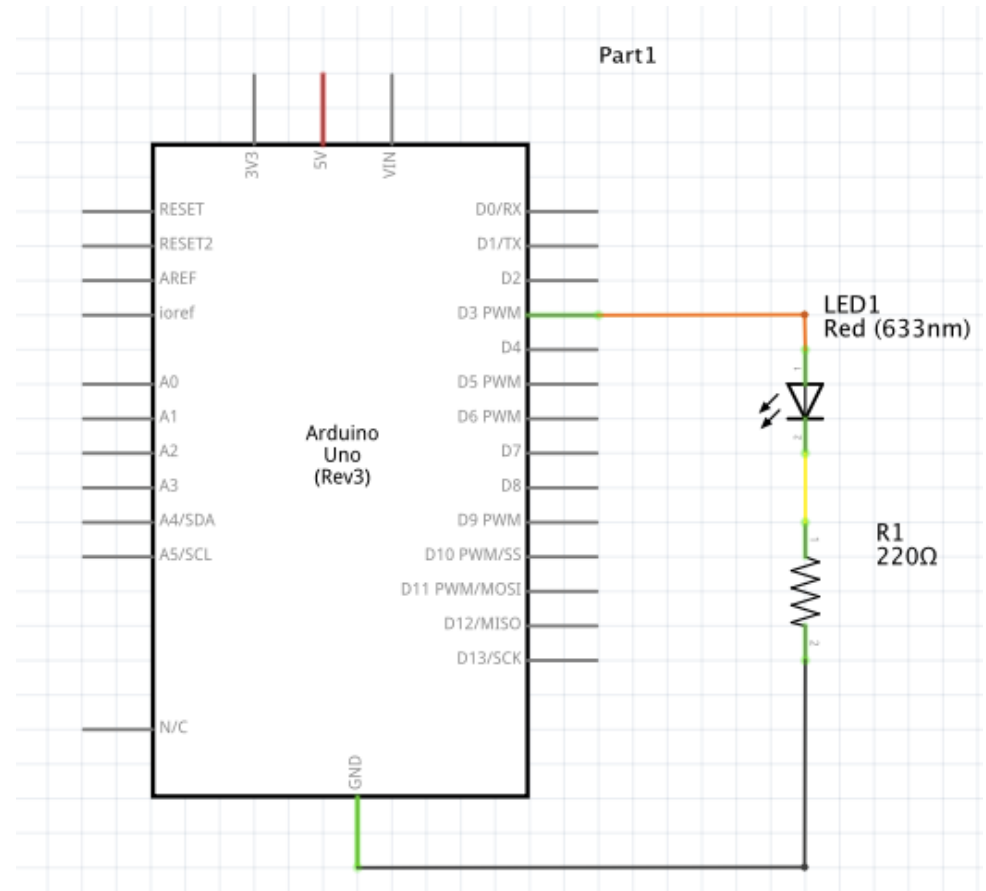
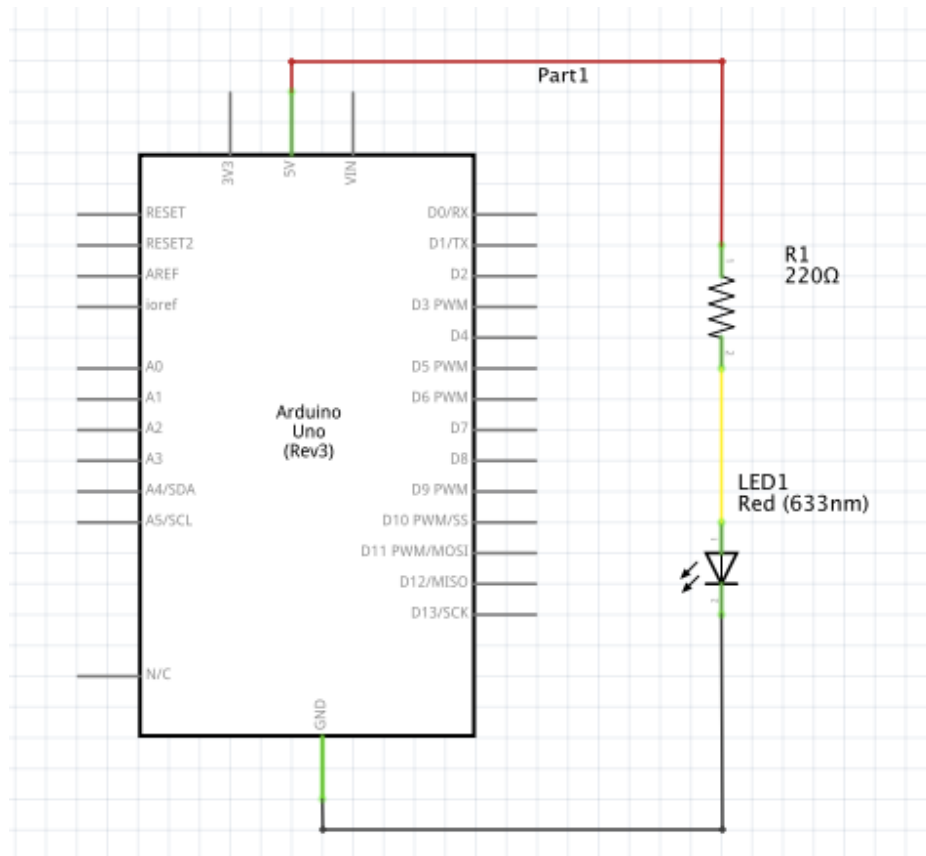
ARDUINO



BREADBOARD - BEST PRACTICE



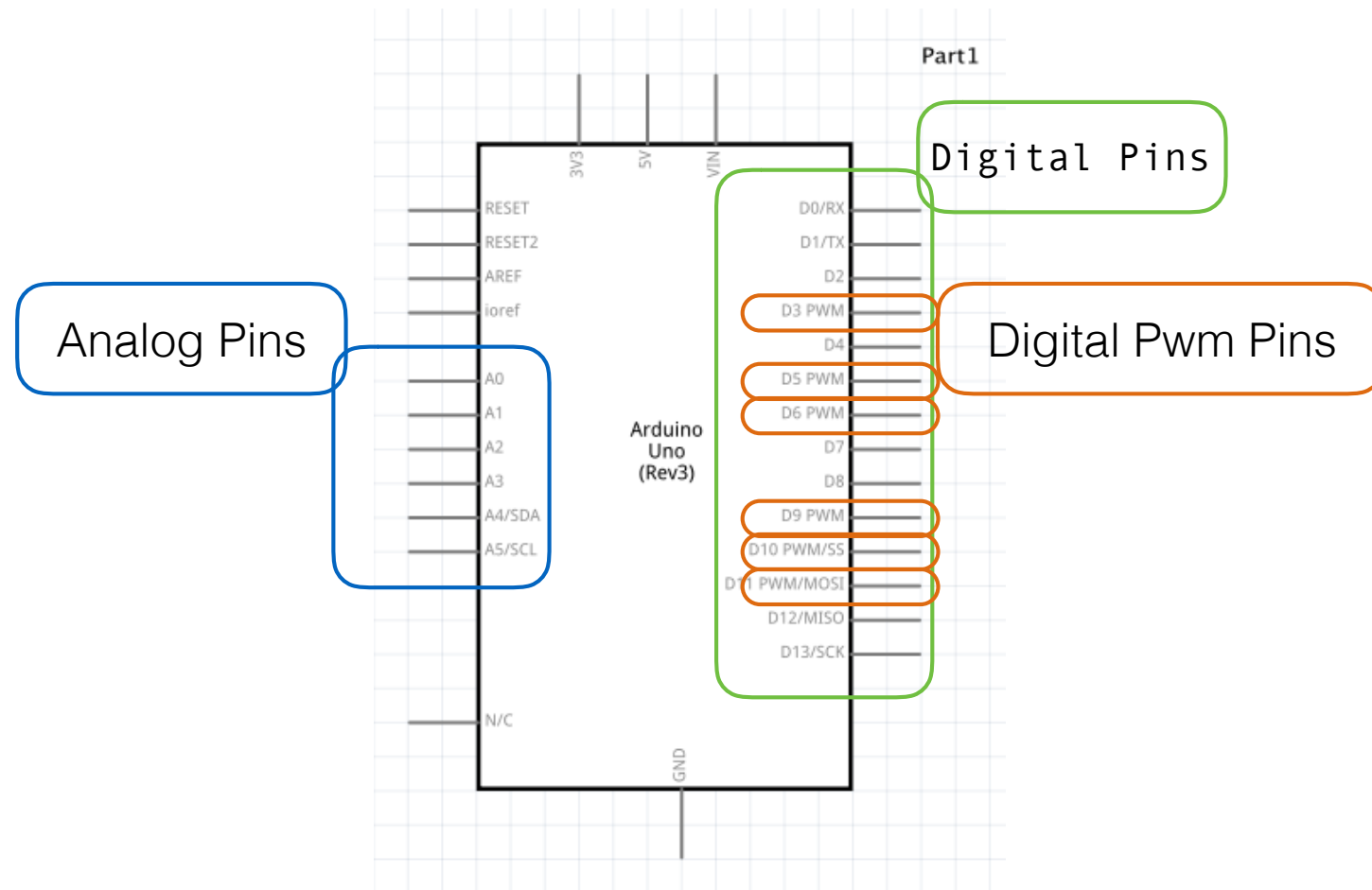
ARDUINO



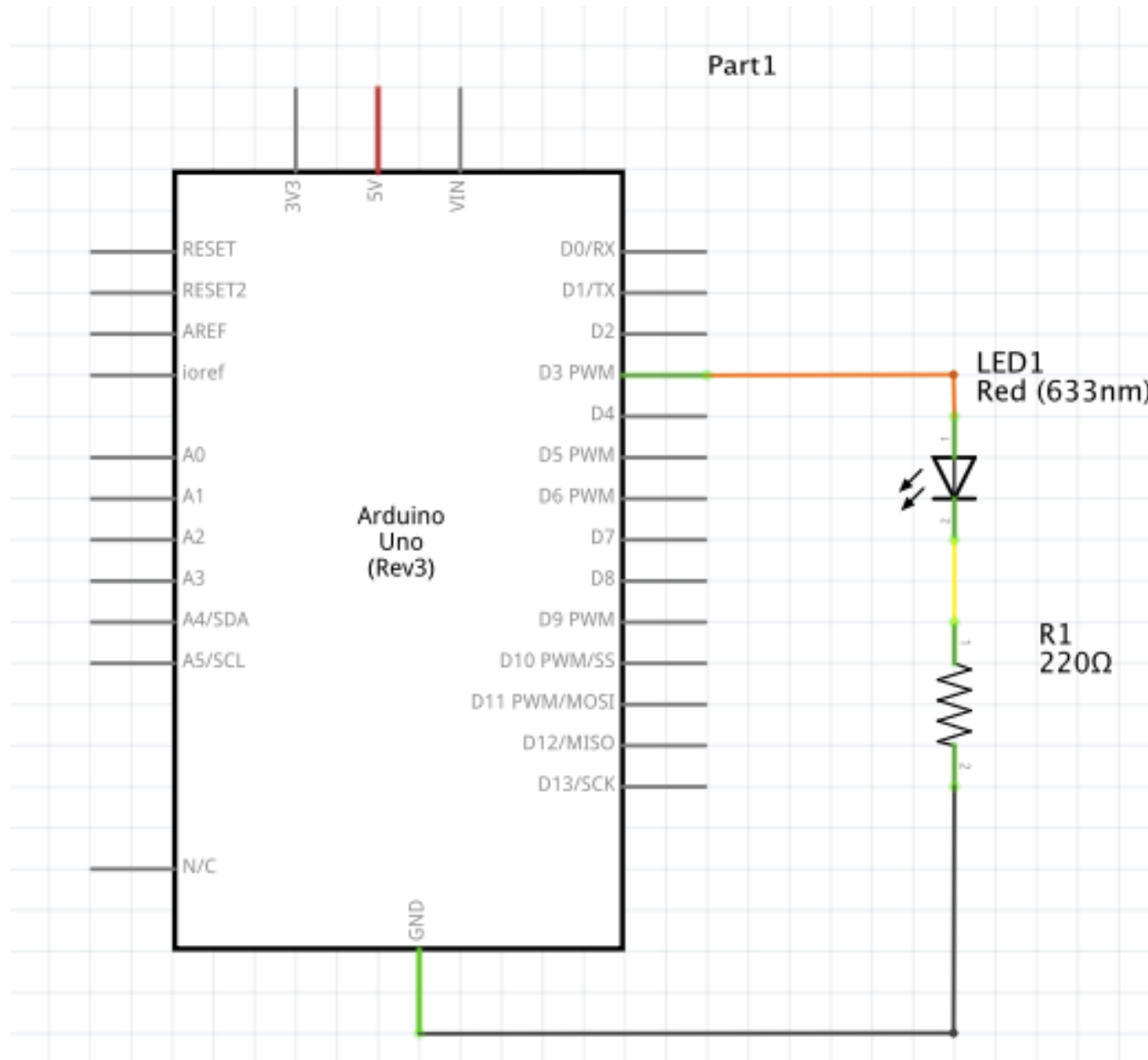
CONNECT THE LED TO PIN D3

INPUT/OUTPUT PIN

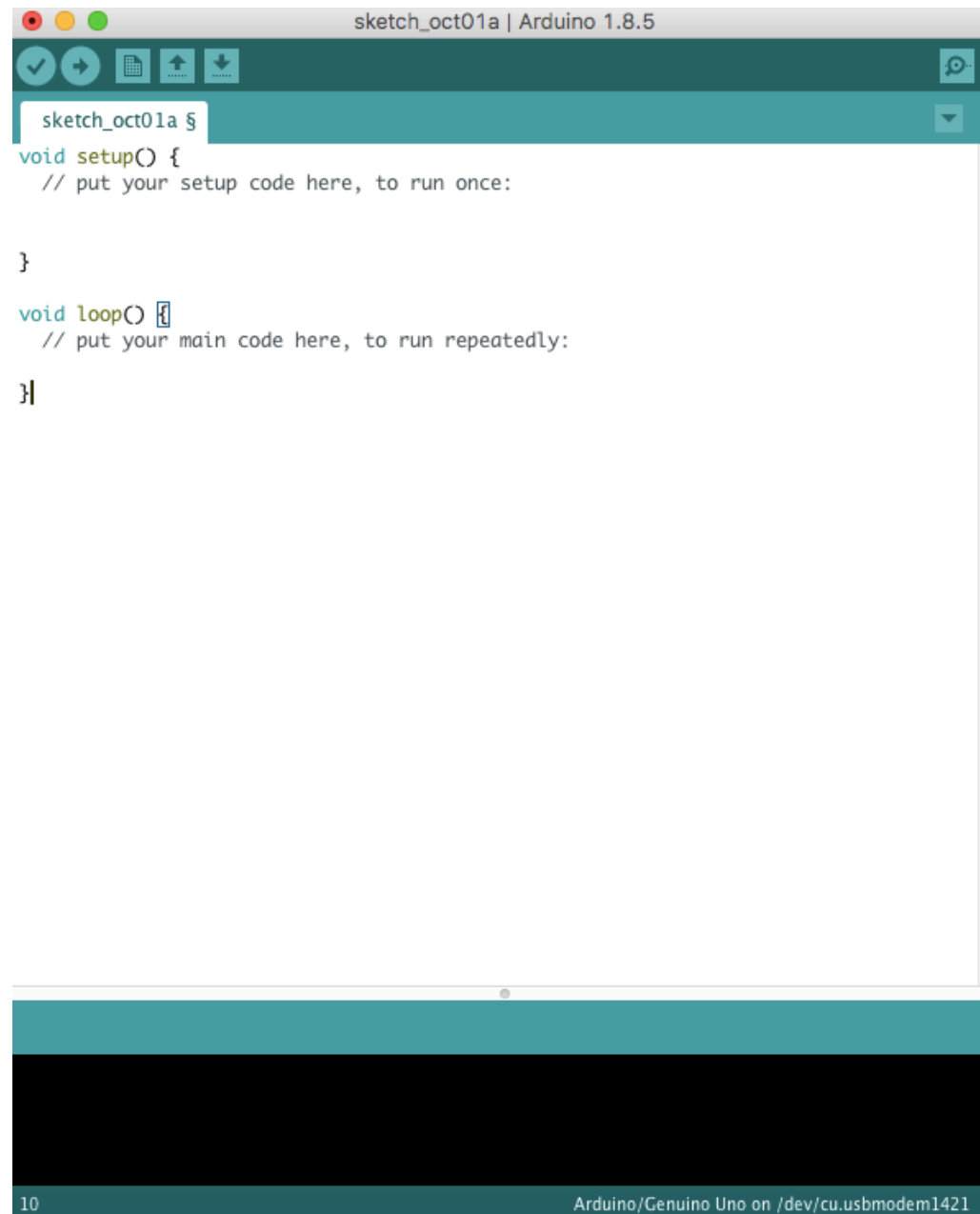
PIN IO (INPUT/OUTPUT)



CONTROL A LED WITH ARDUINO



TIME TO PROGRAM

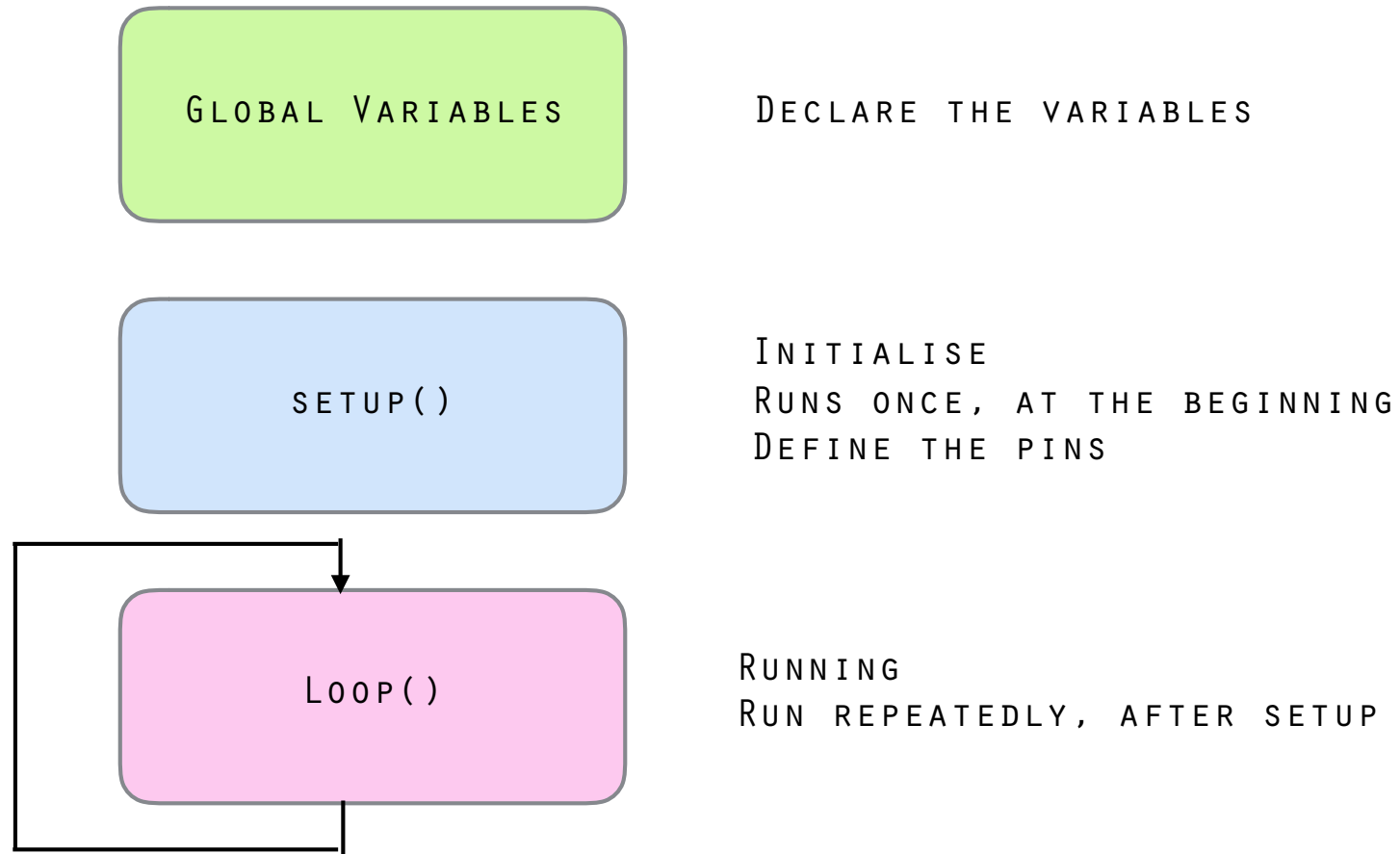


The image shows a screenshot of the Arduino IDE interface. The top window is the sketch editor, titled "sketch_oct01a | Arduino 1.8.5". It contains the following code:

```
sketch_oct01a §  
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Below the sketch editor is a terminal window. The terminal shows the text "10" on the left and "Arduino/Genuino Uno on /dev/cu.usbmodem1421" on the right.

SKETCH



BLINK SKETCH

GLOBAL VARIABLES	<pre>int led_pin = 3; //defin the pin where the Led is connected</pre>
SETUP()	<pre>void setup() { pinMode(led_pin, OUTPUT); //define pin of the Led as an output }</pre>
LOOP()	<pre>void loop() { digitalWrite(led_pin, HIGH); //turn the Led on delay(100); //wait 100millisecond digitalWrite(led_pin, LOW); //turn the Led off delay(100); //wait 100millisecond }</pre>

Done Saving.

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers (but cannot start with a number). They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTH9HXH0

BLINK SKETCH

GLOBAL VARIABLE 'LED_PIN'



The screenshot shows the Arduino IDE interface with a sketch named "_01_blink". The code is as follows:

```
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {
    pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {
    digitalWrite(led_pin, HIGH); //turn the Led on
    delay(100); //wait 100millisecond
    digitalWrite(led_pin, LOW); //turn the Led off
    delay(100); //wait 100millisecond
}
```

A callout box highlights the line `int led_pin = 3; //defin the pin where the Led is connected`. Below the code editor, a status bar indicates "Done Saving." and a message: "The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers (but cannot start with a number). They should also be less than 64 characters long."

At the bottom of the IDE, the hardware and port information is displayed: "20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FT9HXH0".

BLINK SKETCH

SETUP()
DEFINE THE FUNCTION OF THE
PIN
PINMODE(PIN, FUNCTION);

THE PIN IS 'LED_PIN
THE FUNCTION IS OUTPUT



```
_01_blink | Arduino 1.6.9
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {
  pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {
  digitalWrite(led_pin, HIGH); //turn the Led on
  delay(100); //wait 100millisecond
  digitalWrite(led_pin, LOW); //turn the Led off
  delay(100); //wait 100millisecond
}

Done Saving.

The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FT9HXH0
```

BLINK SKETCH

LOOP()

THE LIST OF COMMANDS THAT
ARDUINO RUNS REPEATEDLY.

`DIGITALWRITE(PIN, LEVEL);`
WE CONTROL THE VOLTAGE ON THE
PIN
HIGH: HIGH VOLTAGE (5V)
LOW: LOW VOLTAGE (GROUND)

`DELAY(TIME);`
WE ADD A DELAY IN MILLISECOND



```
_01_blink | Arduino 1.6.9
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {
  pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {
  digitalWrite(led_pin, HIGH); //turn the Led on
  delay(100); //wait 100millisecond
  digitalWrite(led_pin, LOW); //turn the Led off
  delay(100); //wait 100millisecond
}

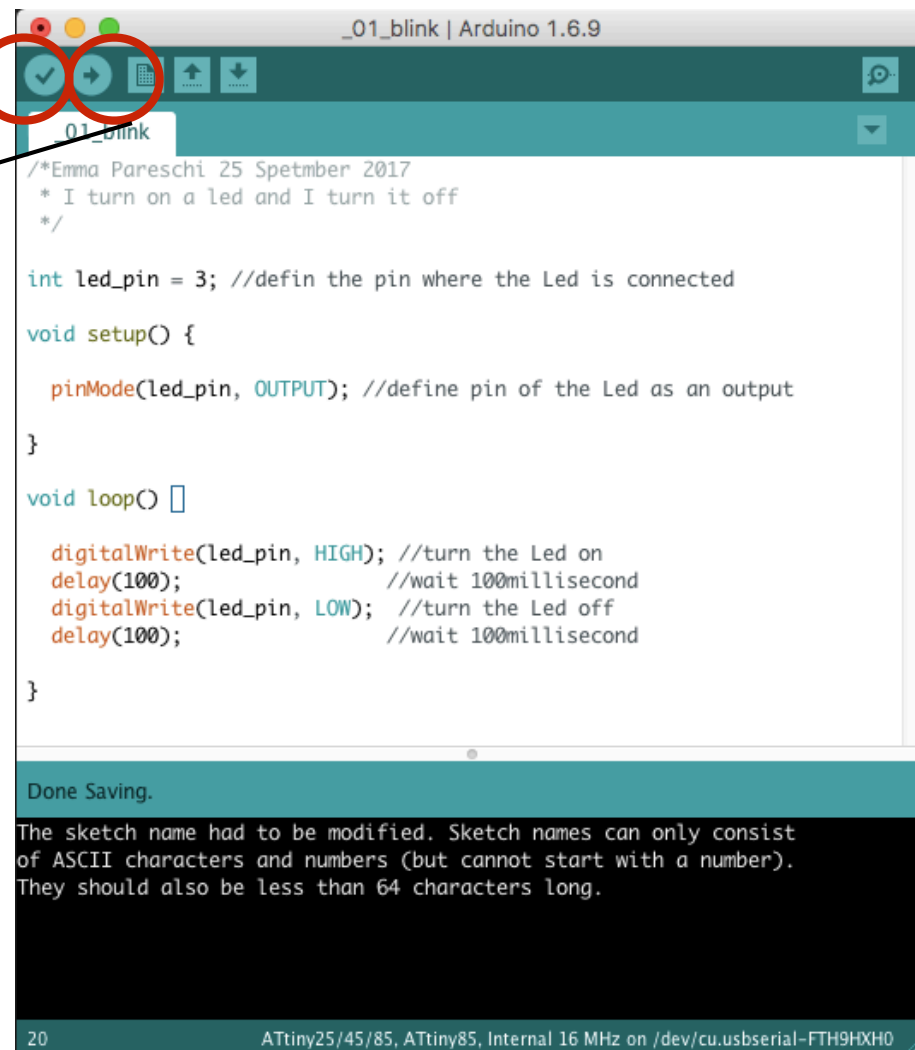
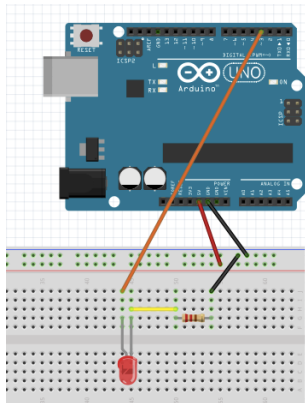
Done Saving.
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTM9HXH0
```


CONNECT, COMPILE AND UPLOAD

1. COMPILE

2. UPLOAD



```
/*Emma Pareschi 25 Spetmber 2017
 * I turn on a led and I turn it off
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {

  pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {

  digitalWrite(led_pin, HIGH); //turn the Led on
  delay(100); //wait 100millisecond
  digitalWrite(led_pin, LOW); //turn the Led off
  delay(100); //wait 100millisecond
}
```

Done Saving.

The sketch name had to be modified. Sketch names can only consist of ASCII characters and numbers (but cannot start with a number). They should also be less than 64 characters long.

20 ATtiny25/45/85, ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTH9HXH0

HOW TO CONTROL THE INTENSITY OF THE LIGHT: ANALOG VS DIGITAL



Analog Signal



Digital Signal

CONTROL THE INTENSITY OF THE LED

`ANALOGWRITE(PIN, NUMBER 0/255);`
WE CONTROL THE VOLTAGE ON THE PIN
255: MAX VOLTAGE (5V)
0: MIN VOLTAGE (GROUND)



The screenshot shows the Arduino IDE interface with a sketch titled "_02_led_analogwrite" in the editor. The sketch is for controlling an LED's intensity using the `analogWrite` function. It includes a comment by Emma Pareschi from September 25, 2017, stating the purpose is to change the LED intensity. The code defines a pin (3) and sets it as an output in the `setup` function. The `loop` function sets the intensity to 255 (max), waits 100ms, sets it to 127 (half), waits 100ms, sets it to 0 (min), and waits 100ms. The status bar at the bottom indicates the sketch is compiled for an Arduino/Genuino Uno on a USB port.

```
_02_led_analogwrite | Arduino 1.6.9
/*Emma Pareschi 25 Spetmber 2017
 * I change the intensity of the Led
 */

int led_pin = 3; //defin the pin where the Led is connected

void setup() {
    pinMode(led_pin, OUTPUT); //define pin of the Led as an output
}

void loop() {
    analogWrite(led_pin, 255); //set max intensity
    delay(100); //wait 100millisecond
    analogWrite(led_pin, 127); //set half intensity
    delay(100); //wait 100millisecond
    analogWrite(led_pin, 0); //set min intensity
    delay(100); //wait 100millisecond
}

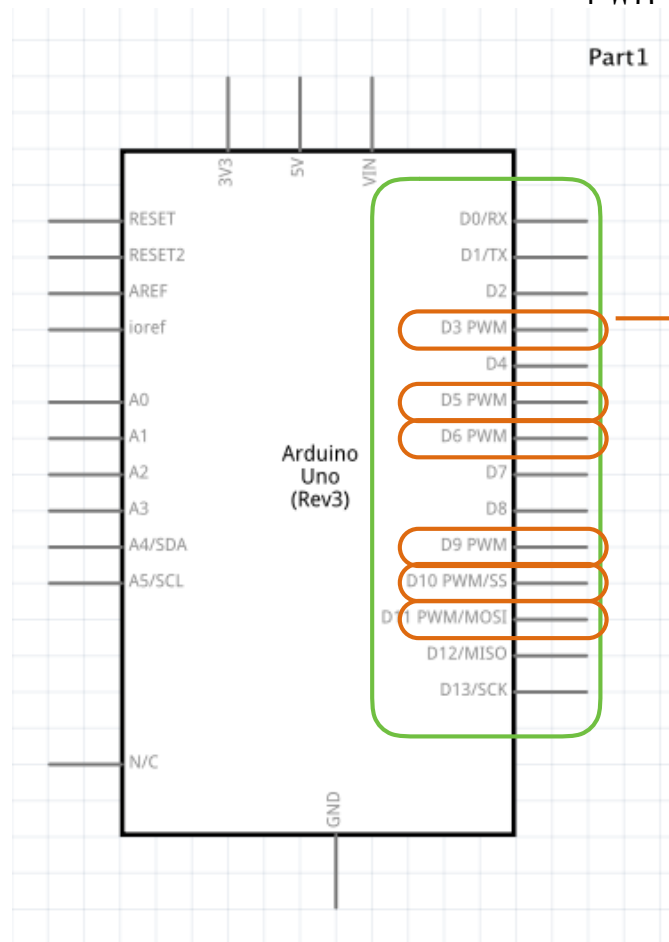
Done compiling.

Sketch uses 1,320 bytes (4%) of program s
Global variables use 11 bytes (0%) of dyn

19 Arduino/Genuino Uno on /dev/cu.usbmodem1421
```

PWM PINS

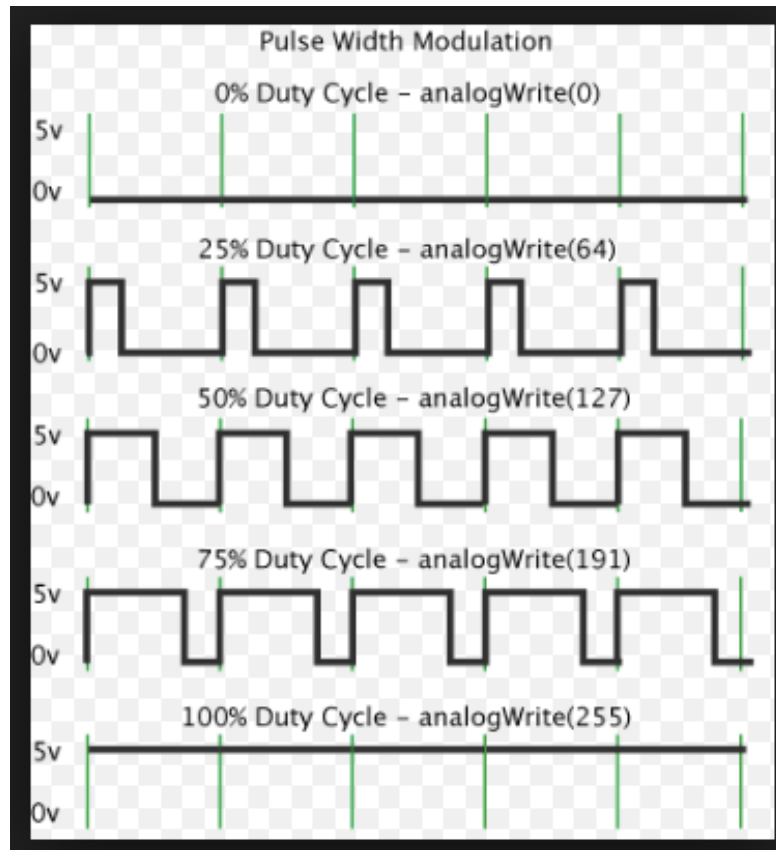
PWM I/O (INPUT/OUTPUT)



ANALOGWRITE() WORKS ONLY
WITH SOME PINS: PWM (~)

PWM

Pulse-width modulation



50% duty cycle



75% duty cycle

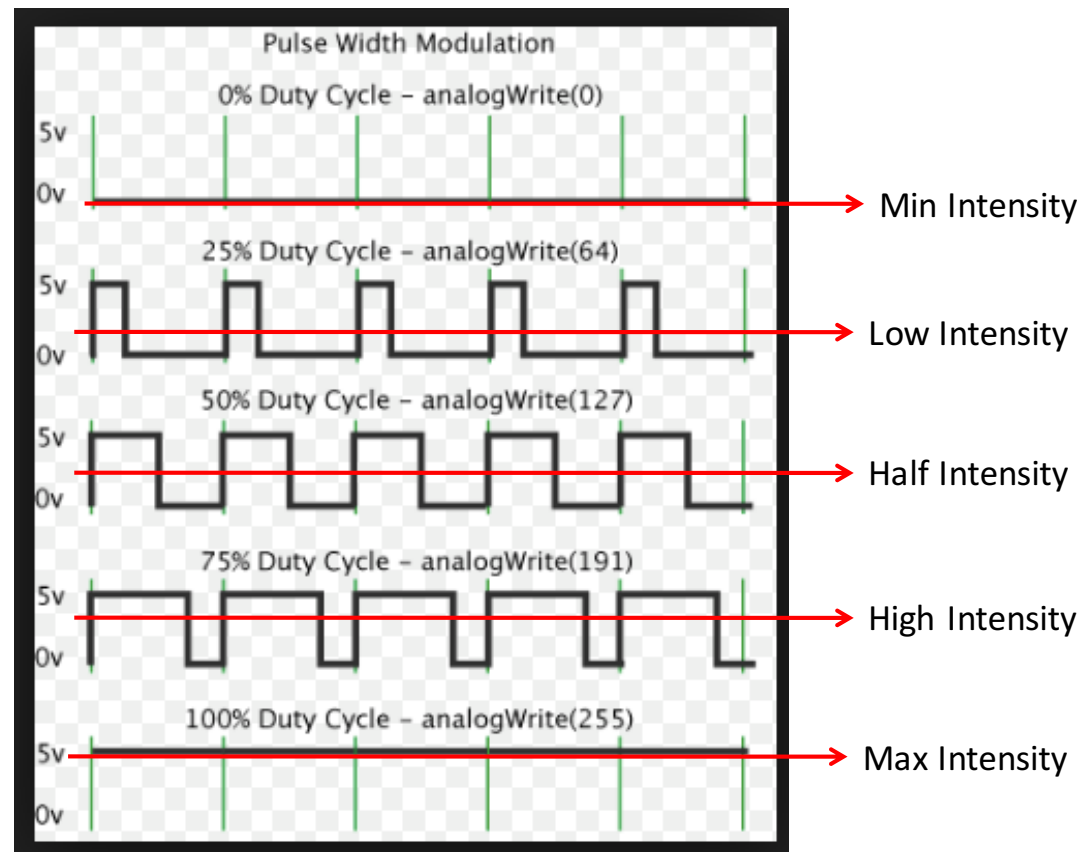


25% duty cycle

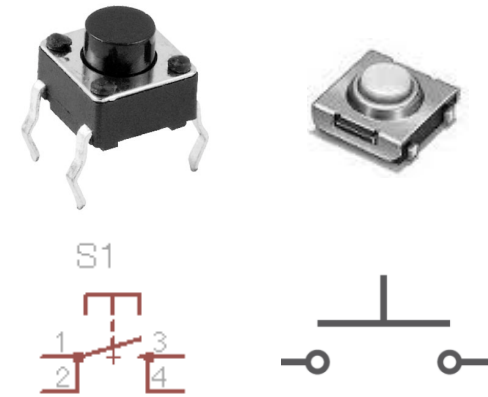
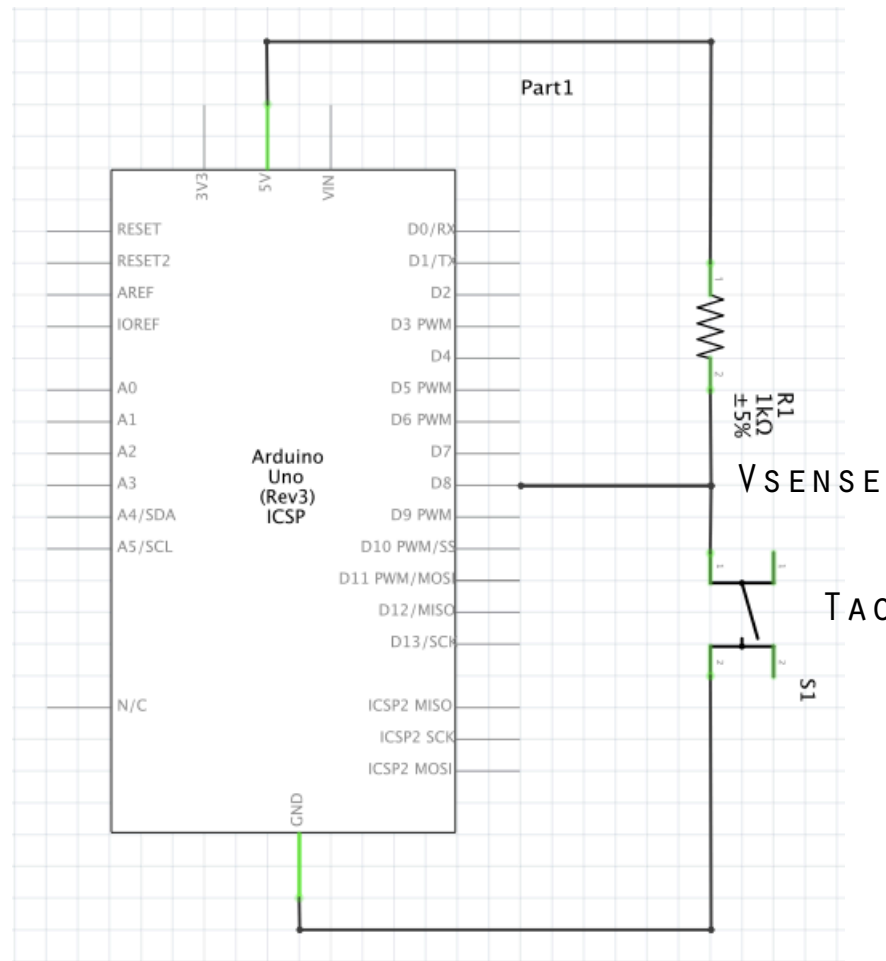


PWM

Pulse-width modulation



HOW TO READ A SENSOR (DIGITAL)



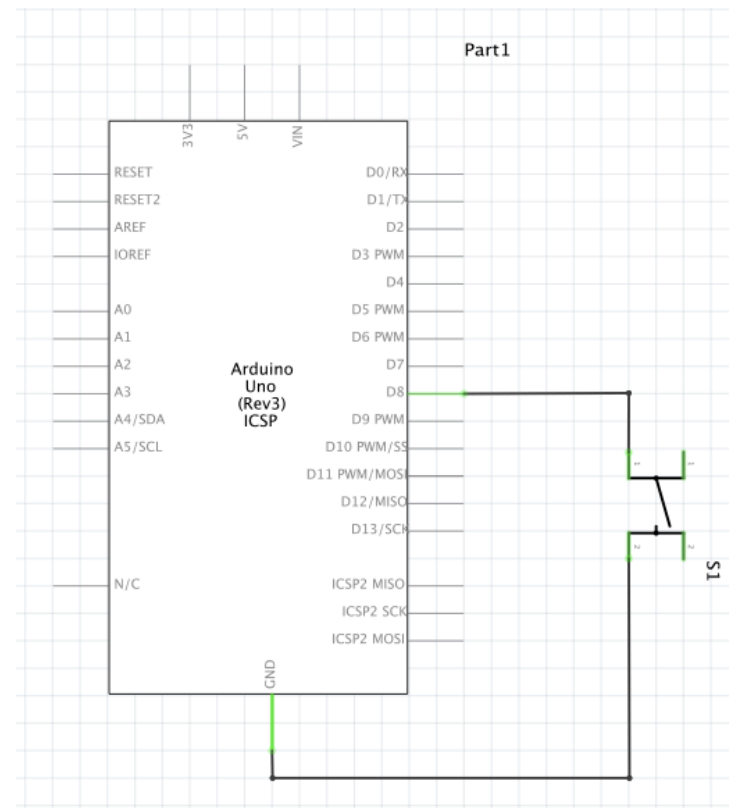
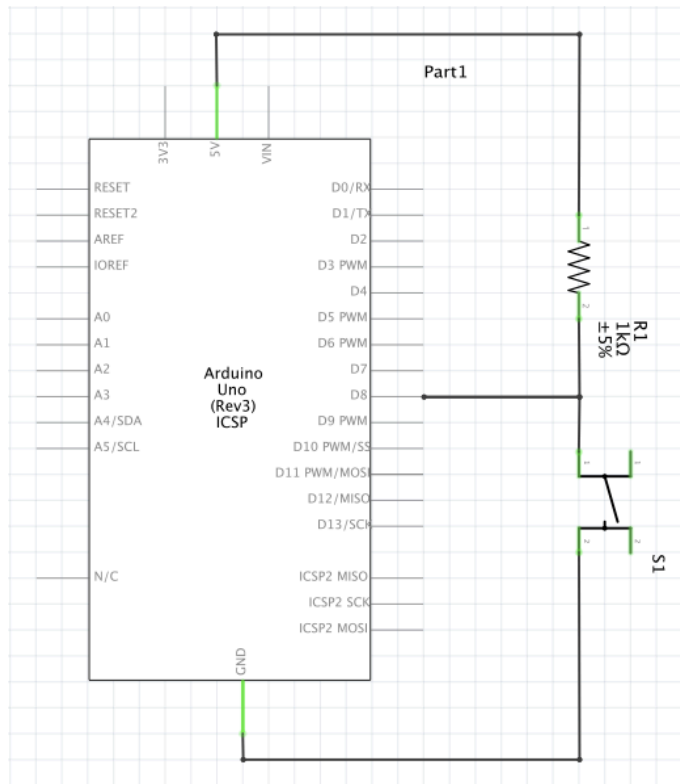
TACT SWITCH

How it works:

If the switch is NOT pushed
 $V_{sense} = 5V$

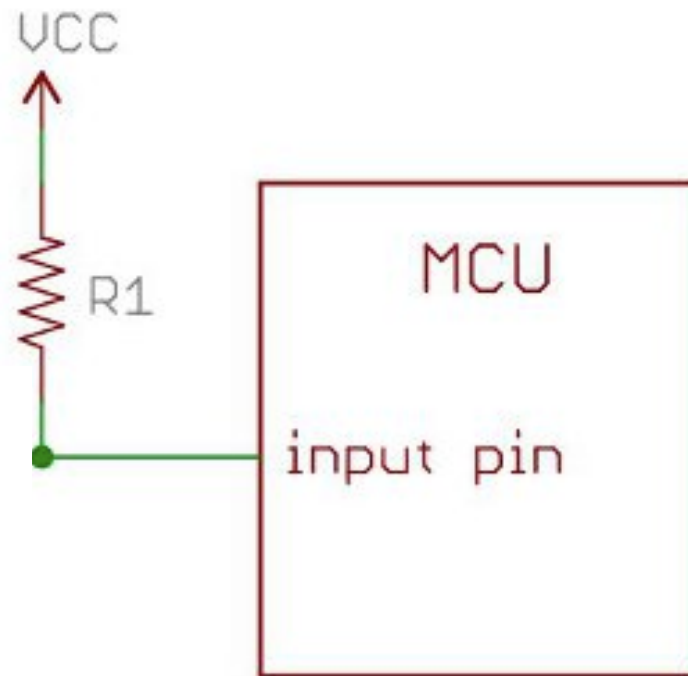
If the switch is pushed
 $V_{sense} = 0$

HOW TO READ A SENSOR (DIGITAL)

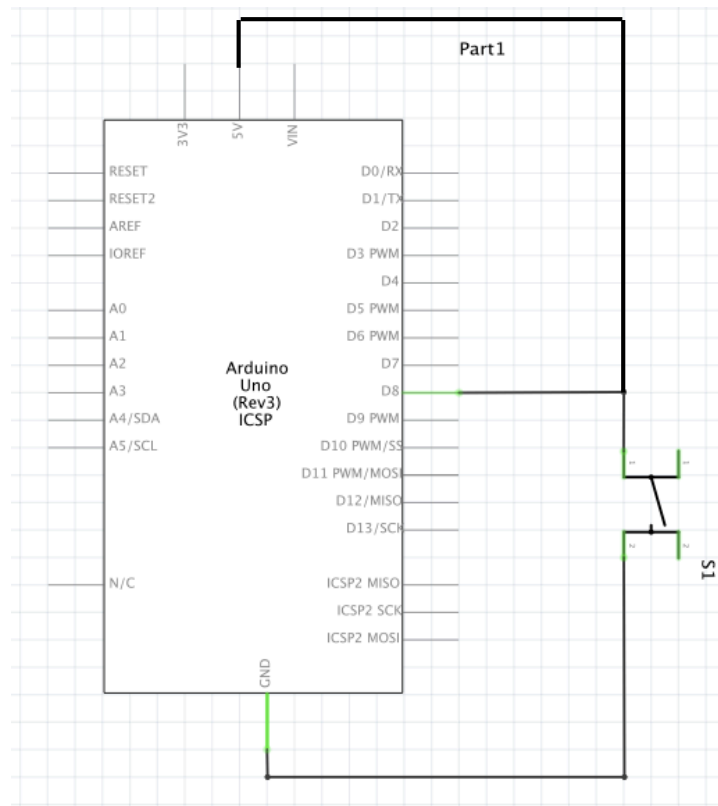


WRONG!!!
(FOR NOW IT'S WRONG)

PULL-UP RESISTOR

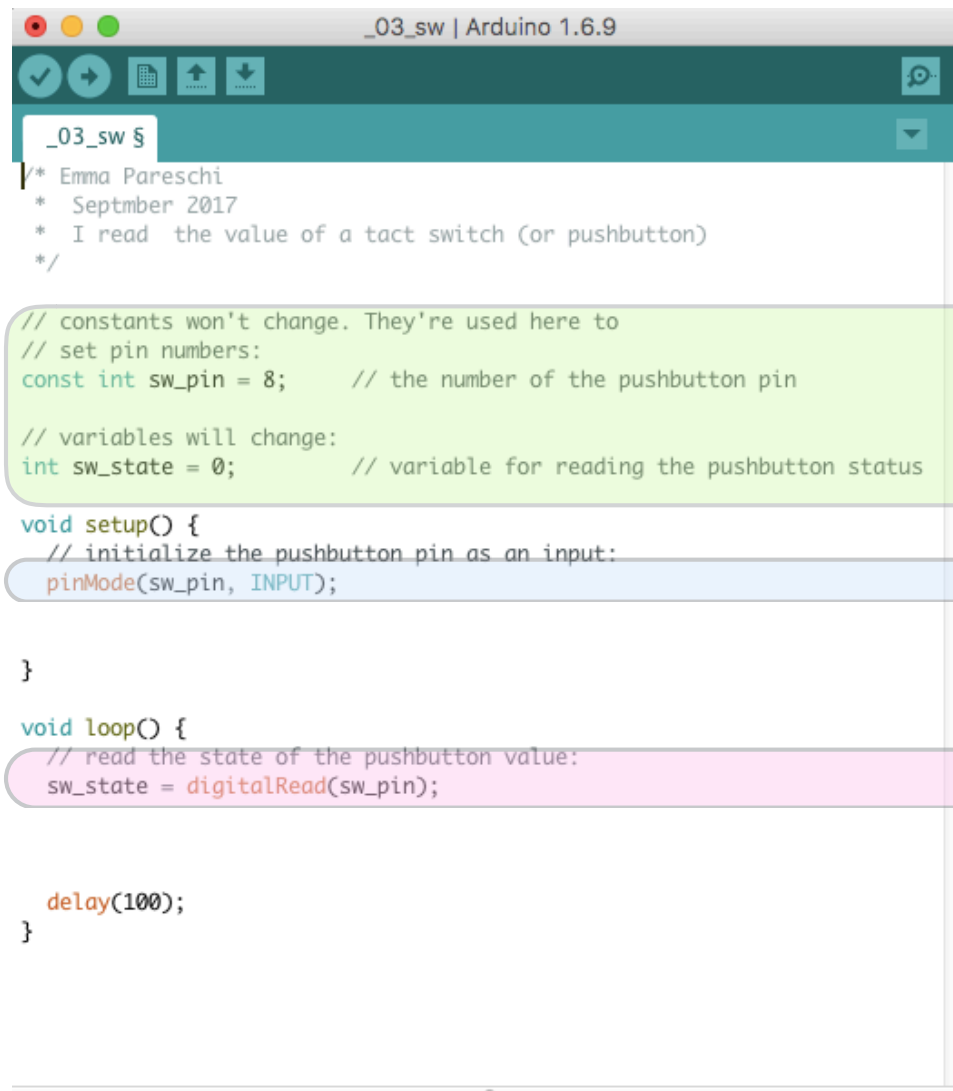


HOW TO READ A SENSOR (DIGITAL)



WRONG!!!
DAMAGE THE PIN!!!

SKETCH - READ A SENSOR (DIGITAL)



The screenshot shows the Arduino IDE interface with a sketch titled "_03_sw | Arduino 1.6.9". The sketch is a C++ program for reading a digital sensor (pushbutton) on an Arduino. The code is as follows:

```
/* Emma Pareschi
 * Septmber 2017
 * I read the value of a tact switch (or pushbutton)
 */

// constants won't change. They're used here to
// set pin numbers:
const int sw_pin = 8;    // the number of the pushbutton pin

// variables will change:
int sw_state = 0;        // variable for reading the pushbutton status

void setup() {
  // initialize the pushbutton pin as an input:
  pinMode(sw_pin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  sw_state = digitalRead(sw_pin);

  delay(100);
}
```


sw_pin: to define the switch pin

sw_state: to save the data

pinMode INPUT

Function to read the value of the pin

SKETCH - READ A SENSOR (DIGITAL) AND PRINT THE VALUES



```
/* Emma Pareschi
 * Septmber 2017
 * I read and print on the computer
 * the value of a tact switch (or pushbutton)
 */

// constants won't change. They're used here to
// set pin numbers:
const int sw_pin = 8;    // the number of the pushbutton pin

// variables will change:
int sw_state = 0;        // variable for reading the pushbutton status

void setup() {
  // initialize the pushbutton pin as an input:
  pinMode(sw_pin, INPUT);
  //open the serial communication with the laptop
  Serial.begin(9600);
}

void loop() {
  // read the state of the pushbutton value:
  sw_state = digitalRead(sw_pin);

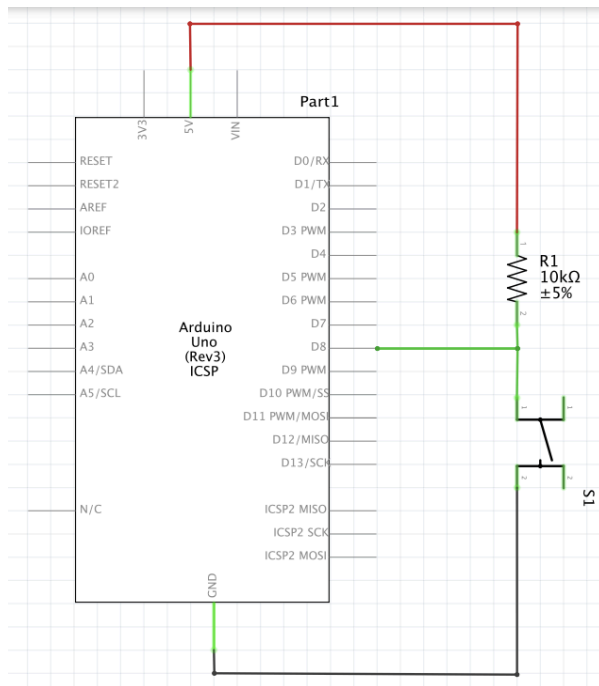
  //print on the serial Monitor
  Serial.print("The value of the switch is: ");
  Serial.println(sw_state);
  //add a delay

  delay(100);
}
```

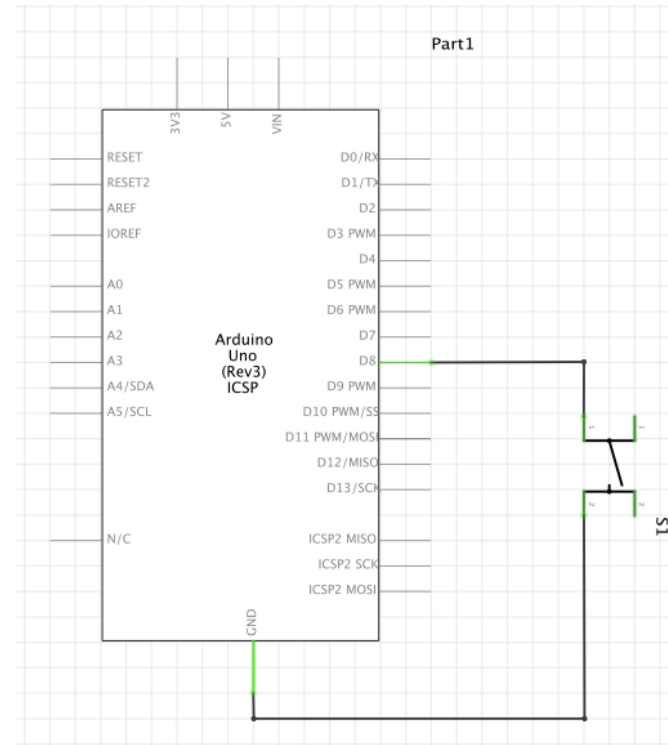
To open communication

To print on serial monitor

PULL-UP RESISTOR

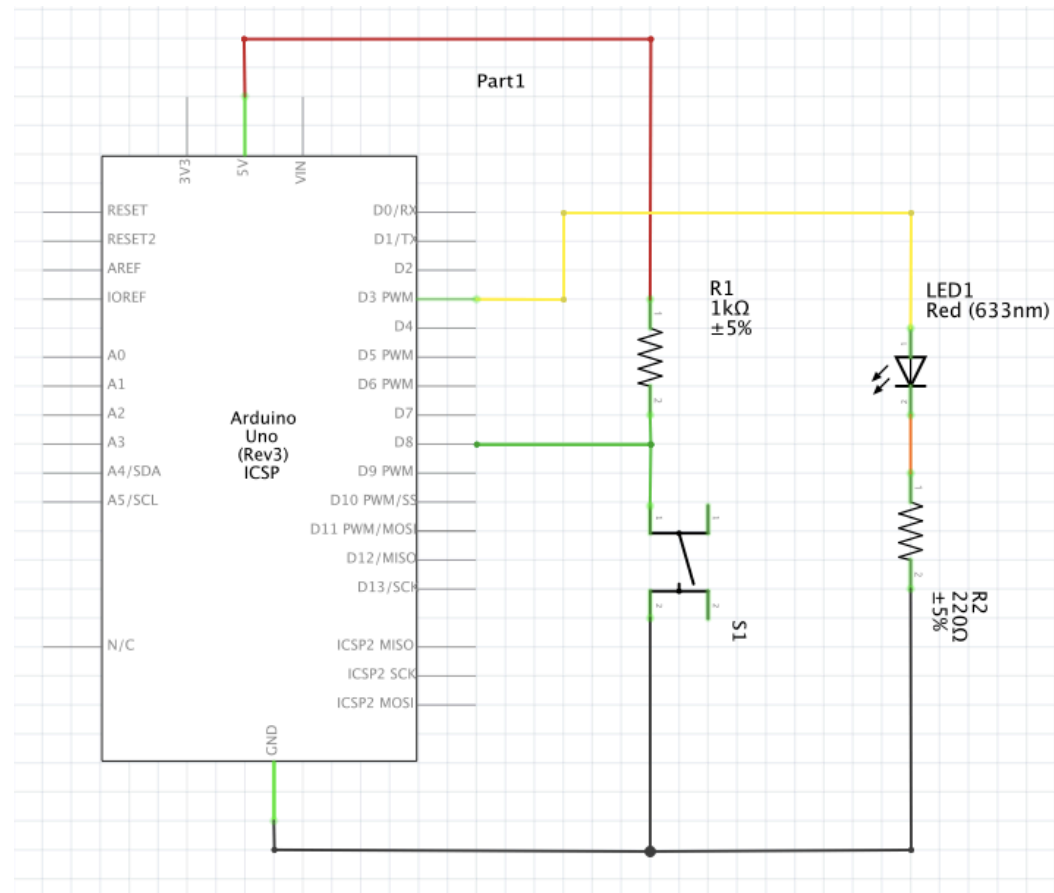


```
PINMODE(3, INPUT);
```



```
pinMode(3, INPUT_PULLUP);
```

TURN ON/OFF THE LED BASED ON THE SWITCH



SKETCH - LED AND SWITCH

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;    // the number of the pushbutton pin
const int ledPin = 13;      // the number of the LED pin

// variables will change:
int buttonState = 0;        // variable for reading the pushbutton state

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Control Structure 'if..else':

```
if (this condition happens)
{
  // action A
}
else
{
  // action B
}
```

IF AND COMPARISON COMPARATOR

```
//all correct|
if (sw_state == LOW){
    Serial.println("CIAO");
}

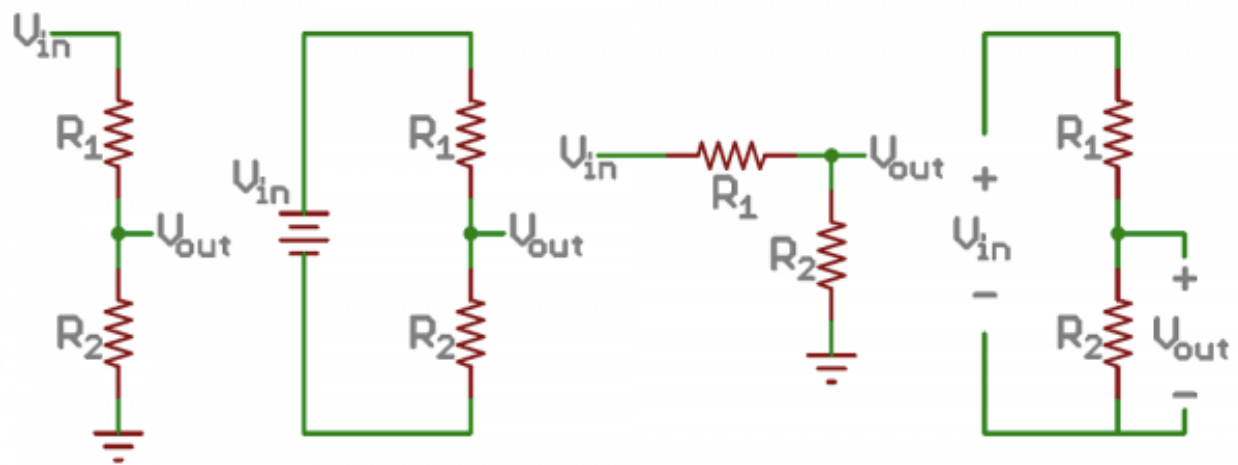
if (sw_state == LOW){ Serial.println("CIAO"); }

if (sw_state == LOW) Serial.println("CIAO");
```

Comparison Operators:

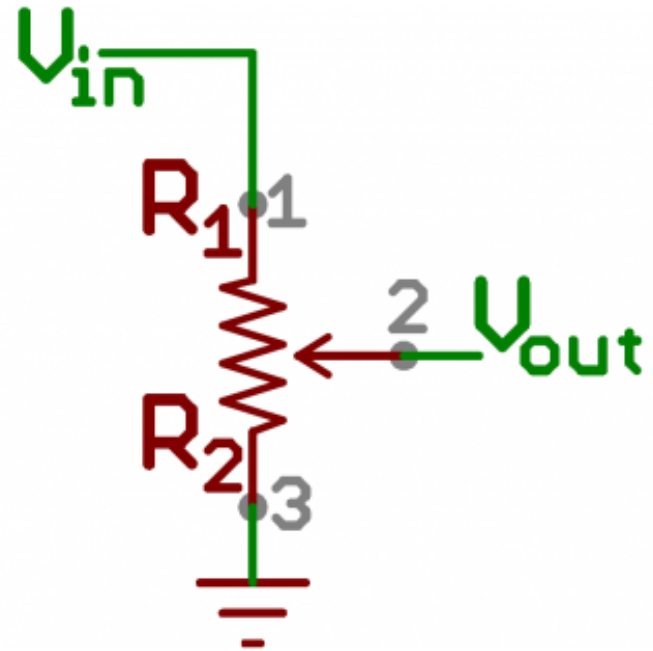
```
x==y (x is equal to y)
x!=y (x is not equal to y)
x<y (x is less than y)
x>y (x is greater than y)
x<=y (x is less than or equal to y)
x>=y (x is greater than or equal to y)
```


VOLTAGE DIVIDER

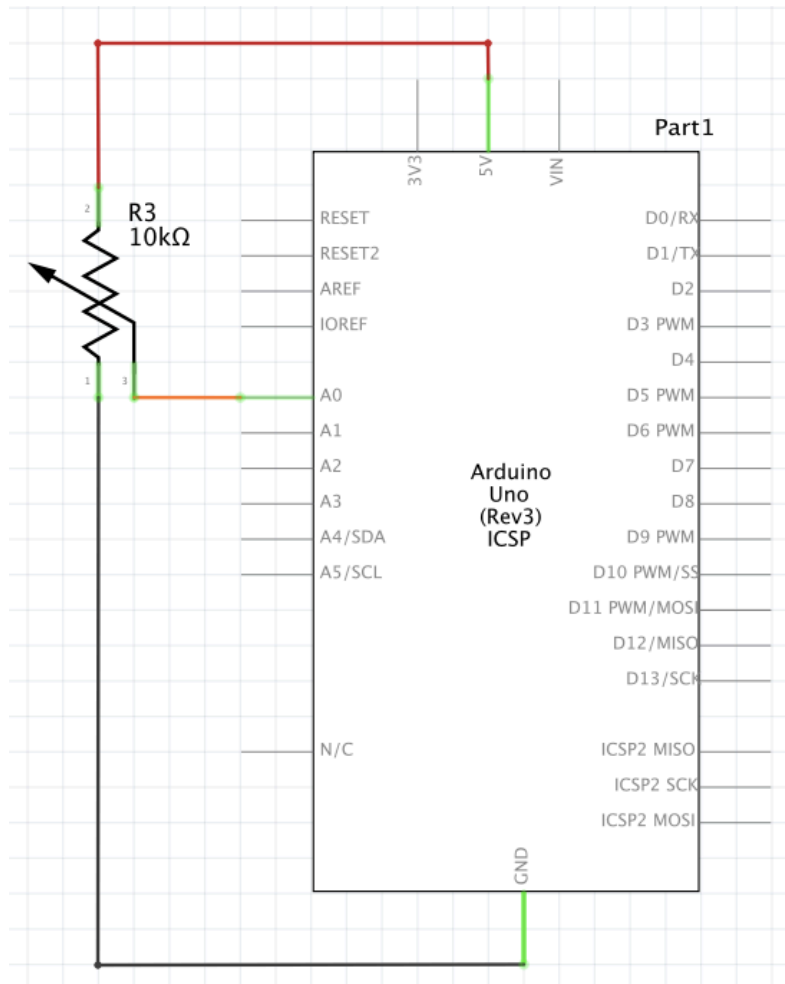


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

RESISTOR DIVIDER OR POTENTIOMETER - APPLICATION

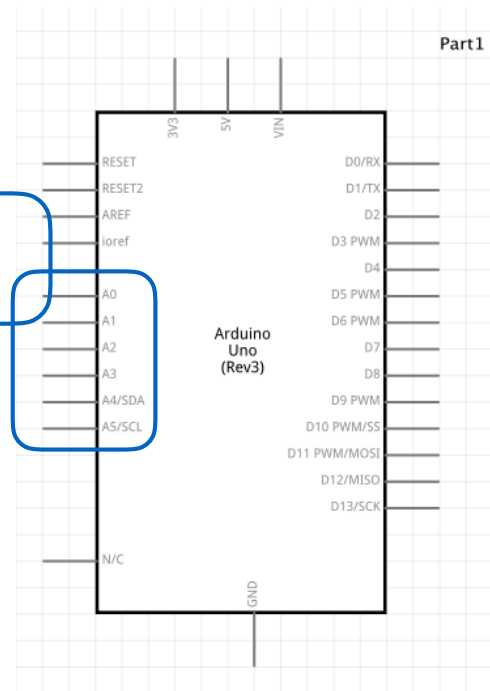


POTENTIOMETER - SCHEMATIC



TO READ A POTENTIOMETER OR ANY SENSOR THAT GENERATE AN ANALOG VOLTAGE => YOU NEED TO USE THE ANALOG PINS.

Analog Pins:
A0...A5



POTENTIOMETER - SKETCH

```
pot_serial
/* Emma Pareschi
 * October 2017
 * I read the output voltage of a potenziometer and
 * I print it on the Serial Monitor
 */

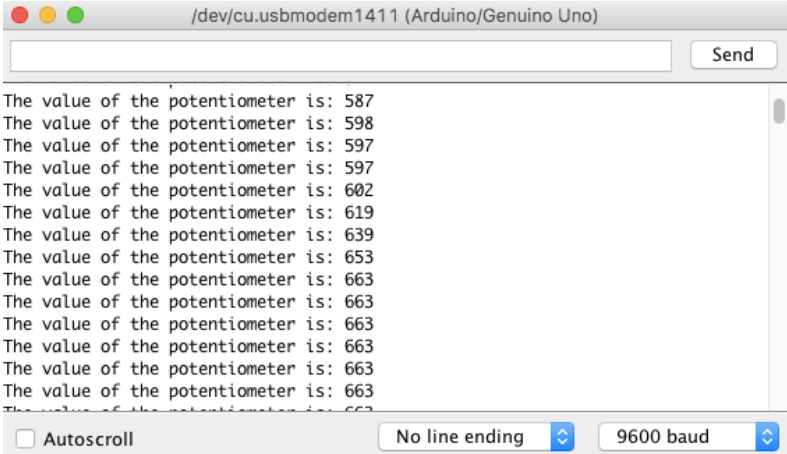
const int pot_pin = A0; //set the variable of the pot pin, it's a constant
int pot_value = 0;      //set the variable to save the status of the pot

void setup() {
  pinMode(pot_pin, INPUT); //define the function of the pin
  Serial.begin(9700);      //open communication
}

void loop() {
  pot_value = analogRead(pot_pin); //read the push button status

  Serial.print("The value of the potentiometer is: ");
  Serial.println(pot_value);

  delay(100);           // wait
}
```



The value of the potentiometer is: 587
The value of the potentiometer is: 598
The value of the potentiometer is: 597
The value of the potentiometer is: 597
The value of the potentiometer is: 602
The value of the potentiometer is: 619
The value of the potentiometer is: 639
The value of the potentiometer is: 653
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663
The value of the potentiometer is: 663

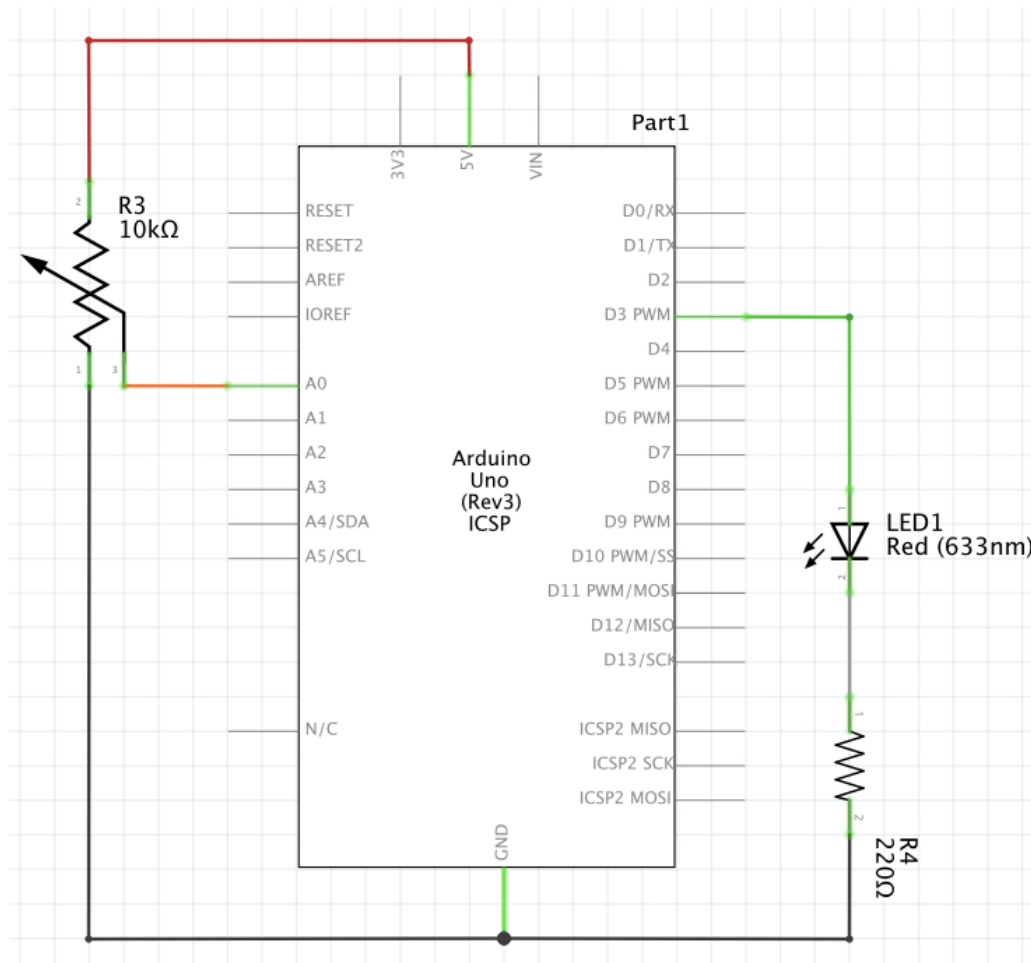
☐ Autoscroll No line ending 9600 baud

Function to read an analog voltage is:

`analogRead(pin);`

It will return a number between 0 and 1023

POTENTIOMETER AND LED - SCHEMATIC



POTENTIOMETER AND LED - SKETCH

```
pot_led_1
/*
 * Emma Pareschi
 * October 2017
 * Analog Input
 * Demonstrates analog input by reading an analog sensor on analog pin 0 and
 * turning on and off a light emitting diode(LED) connected to digital pin 3.
 * The amount of time the LED will be on and off depends on
 * the value obtained by analogRead().
 */

int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;     // select the pin for the LED
int pot_value = 0;   // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT and the potentiometer as INPUT
  pinMode(led_pin, OUTPUT);
  pinMode(pot_pin, INPUT);
}

void loop() {
  // read the value from the sensor:
  pot_value = analogRead(pot_pin);
  // turn the ledPin on
  digitalWrite(led_pin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(pot_value);
  // turn the ledPin off:
  digitalWrite(led_pin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(pot_value);
}
```

POTENTIOMETER AND LED - SKETCH

```
pot_led_2
/*
 * Emma Pareschi
 * October 2017
 */

const int pot_pin = A0;    // select the input pin for the potentiometer
int led_pin = 3;           // select the pin for the LED
int pot_value = 0;         // variable to store the value coming from the sensor

void setup() {
    // declare the ledPin as an OUTPUT and the potentiometer as INPUT
    pinMode(led_pin, OUTPUT);
    pinMode(pot_pin, INPUT);
}

void loop() {
    // read the value from the sensor:
    pot_value = analogRead(pot_pin);

    pot_value = map(pot_value,0,1023,0,255); //Convert from 0-1023 to 0-255

    analogWrite(led_pin,pot_value); //turn the led on depend on the output value
}
```

WEEKLY ASSIGNMENTS

- Study the codes we used in class.
- Ex from the whiteboard in class. See next slides.
- Navigate through <https://create.arduino.cc/projecthub>, select a project, repeat it and document(*) it in your blog. Try to understand the code!
- Read the following pages:
 - <https://learn.sparkfun.com/tutorials/pull-up-resistors>
 - <https://learn.sparkfun.com/tutorials/voltage-dividers>
 - <https://learn.sparkfun.com/tutorials/analog-vs-digital/all>
 - Chapter 11 of “Encyclopedia of Electronic Components Volume 1” (potentiometer)

References:

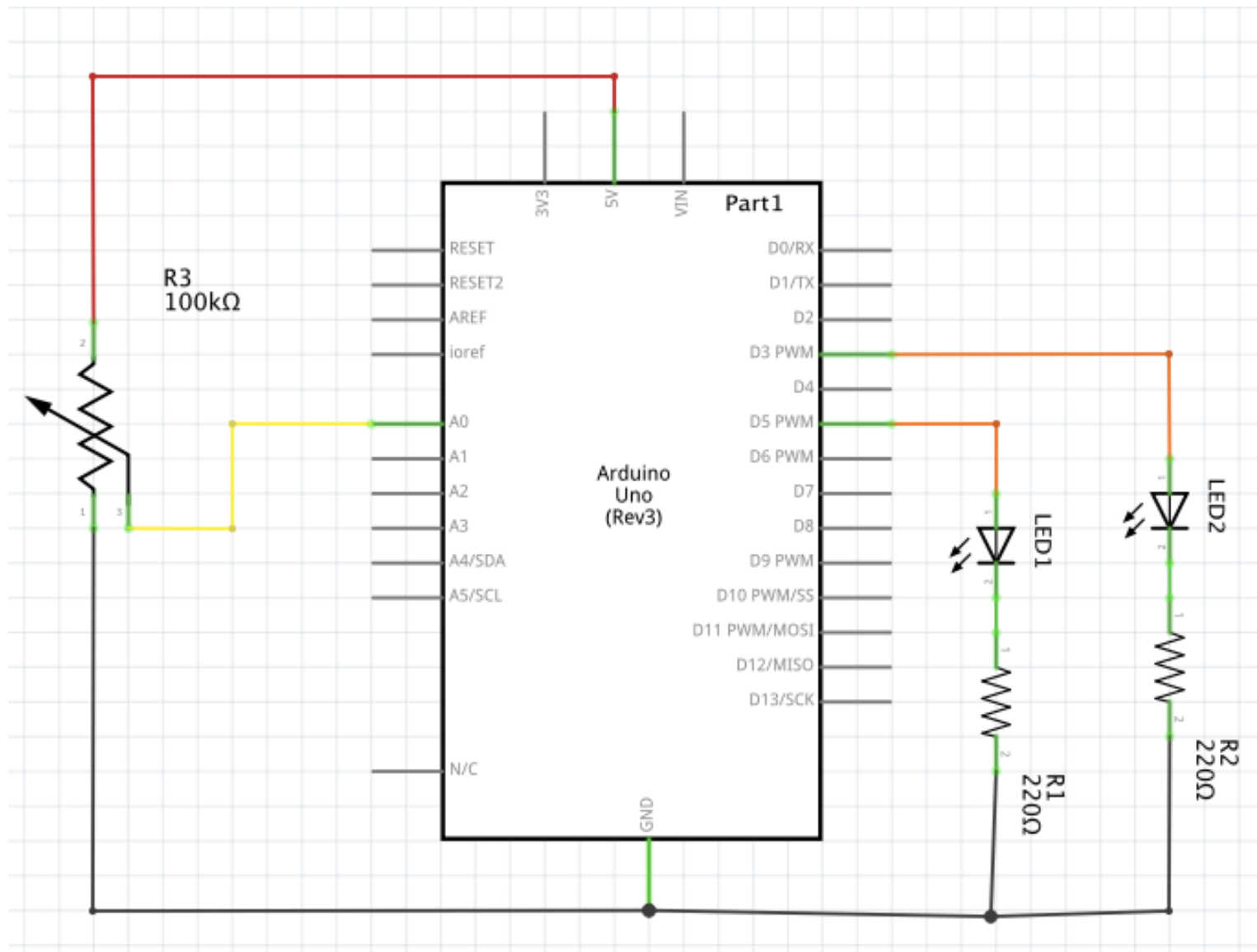
- <https://www.arduino.cc/reference/en/>

(*) documentation: schematic, photo/video of the circuit on breadboard, code (with comments!!!), description on the functionality.

Outcome:

Get familiar Arduino IDE workflow, capability to use the basic function of Arduino language, understand the difference between analog and digital.

WEEKLY ASSIGNMENTS - WHITEBOARD



WEEKLY ASSIGNMENTS - WHITEBOARD

Using the circuit in the previous slide, program the Arduino:

- Ex1: control the intensity of both LEDs (Led1 and Led2) using the potentiometer.
- Ex2: based on the potentiometer rotation, control the brightness of the LEDs in the following way: when the Led1 has maximum intensity, the Led2 has minimum and vice-versa.

LICENCE

EXCEPT WHERE OTHERWISE NOTED, THIS WORK IS LICENSED UNDER:

[HTTPS://CREATIVECOMMONS.ORG/LICENSES/BY/4.0/](https://creativecommons.org/licenses/by/4.0/)

