Electronics 1

Electronics for Interactive Media Design

Les 2

Emma Pareschi

Voltage - RESISTOR - Current

A resistor is one of the most fundamental components in electronics. Its purpose is to impede a flow of current and impose a voltage reduction.

Two wires or conductors attached at opposite ends or sides of a relatively poor electrical conductor.

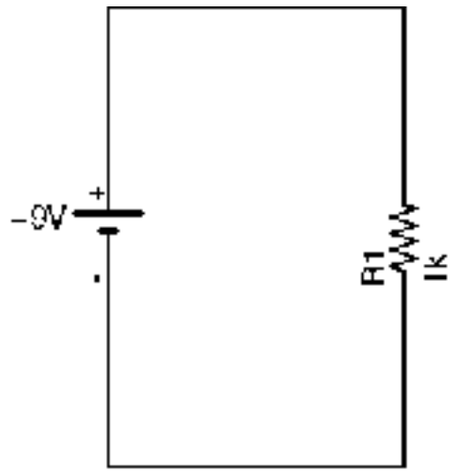
The resistance of resistor is measured in ohms, universally represented by the Greek omega symbol, Ω.

Ω	kΩ	MΩ
Ohms	Kilohms	Megohms
1	0.001	0.000001
10	0.01	0.00001
100	0.1	0.0001
1,000	1	0.001
10,000	10	0.01
100,000	100	0.1
1,000,000	1,000	1

Ohm's Law

Ohm's Law defines the relation between
voltage, current and resistor

V = I * R = IR



You know the Voltage and the Resistor =>
=> calculate the current you consume: I = V / R

You know the Voltage and the Current =>
=> calculate the resistor you will use: R = V / I

Electric Power

In physics: power is defined as the rate at which energy is transferred (or transformed). Unit of measure WATT (W).

ENERGY is basically the ability of something to move something else. Forms of energy: mechanical, electrical, chemical, electromagnetic, thermal, and many others.

TRANSFORMATION: energy can never be created or destroyed, only transferred to another form. A lot of what we're doing in electronics is converting different forms of energy to and from electric energy.

Energy type converted	Converted by
Mechanical	Electric Motor
Electromagnetic	LED
Heat	Resistor
Chemical	Battery
Wind	Windmill

Electric power is measured by combining both how much electric energy is transferred, and how fast that transfer happens.

Electric Power for Resistors

how much electric energy is transferred: Voltage
how fast that transfer happens: Current

Unit of measure: Watt (W)

$$P = V * I = VI$$

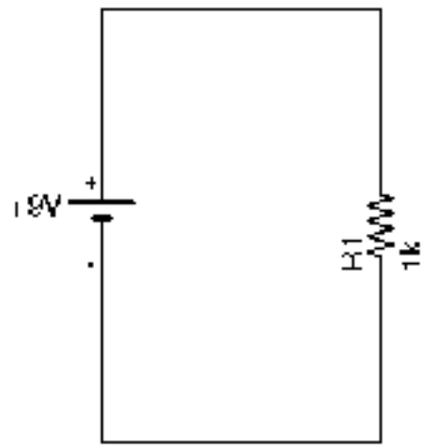
Electric Power for Resistors

$P = VI$
or
$P = RI * I = RI^2$
or
$P = V * V/R = V^2/R$

The most common power rating is 1/4W = 0.25W

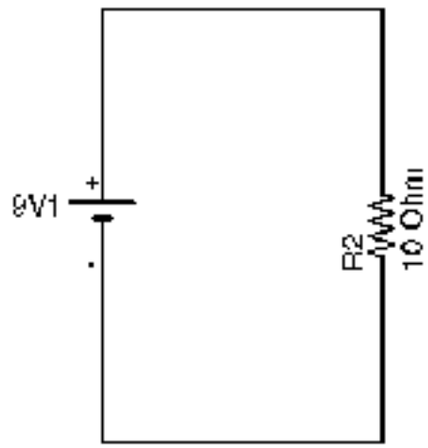
In case of resistors electric energy is converted to heat.

Power examples



V = 9 V
R = 1K Ohm
I = 0.009 A

P = 0.081W

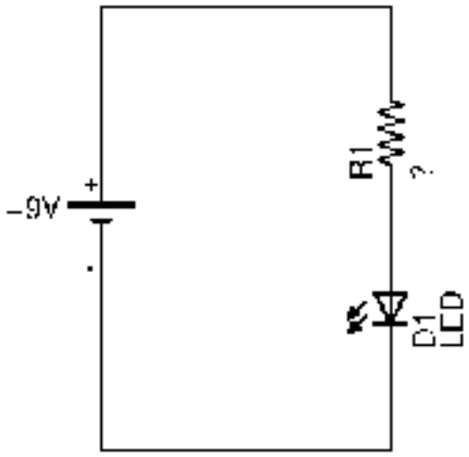


V = 9 V
R = 10 Ohm
I = 0.9 A

P = 810W

Circuit LED

How to use Ohm's law in the designing of my circuit?



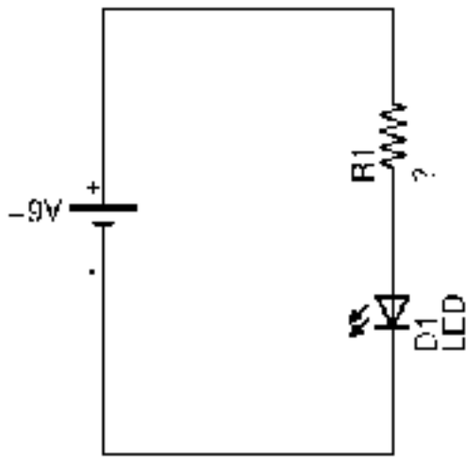
Specifications from Datasheet:
- wavelength of emitted light
- luminous intensity
- maximum forward voltage and current
- maximum reverse voltage and current,
- working values for voltage and current

Productgegevens

LED Colour:	Red	**Viewing Angle:**	68°
LED Mounting:	Through Hole	**Lens Shape:**	Round
Bulb Size:	T-1 3/4 (5mm)	**Packaging:**	Each
Forward Current If:	25mA	**Product Range:**	-
Forward Voltage:	1.7V	**Automotive Qualification Standard:**	-
Wavelength Typ:	660nm	**MSL:**	-
Luminous Intensity:	32mcd		

Circuit LED

How to calculate the resistor



Specifications:

- working values for voltage and current
  - Vf = 1.7V
  - If = 25mA

$R = (V_{CC} - V_F) / I$

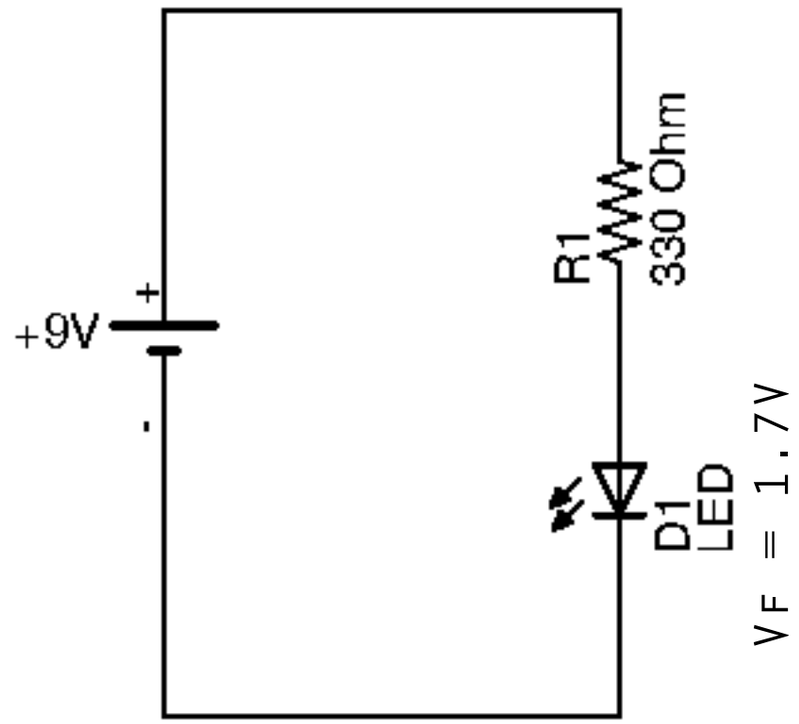
R = (9 - 1.7)/0.025 = 292 Ohm ~ 300 Ohm

Circuit LED

Typical forward voltages for various colours are, typical current 20mA:

- Infrared LED: 1.6V to 2V
- Red LED: 1.6V to 2.1V
- Orange LED: 1.9V to 2.1V
- Amber LED: 2V to 2.1V
- Yellow LED: 2V to 2.4V
- Green LED: 2.4V to 3.4V
- Blue LED: 3.2V to 3.4V
- Ultraviolet LED: 3.3V to 3.7V
- White LED: 3.2V to 3.6V

Circuit LED Analisys



Vcc = 9V
R = 330 Ohm
Vf = 1.7V

I = (Vcc - Vf) / R = 22mA

Power of the resistor:
P_R = (Vcc - Vf)*I = 160mW

Power of the LED:
P_{LED} = Vf * I = 37mW

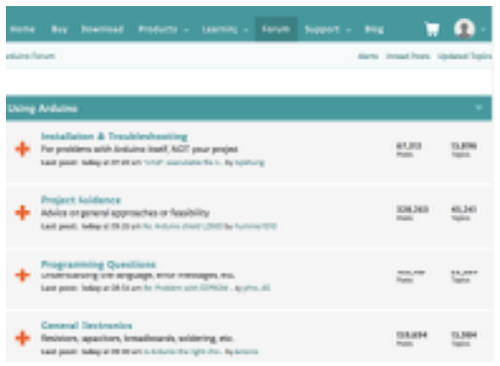
Arduino

> ``***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

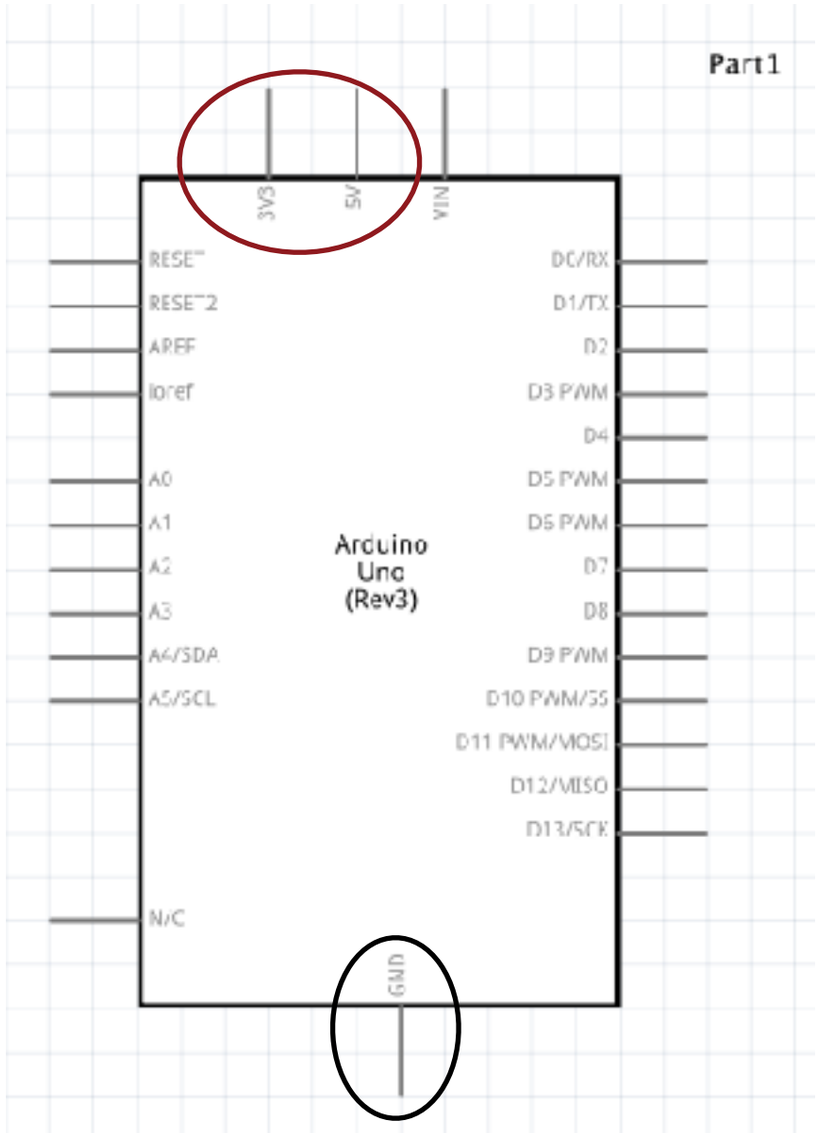
Software

Community



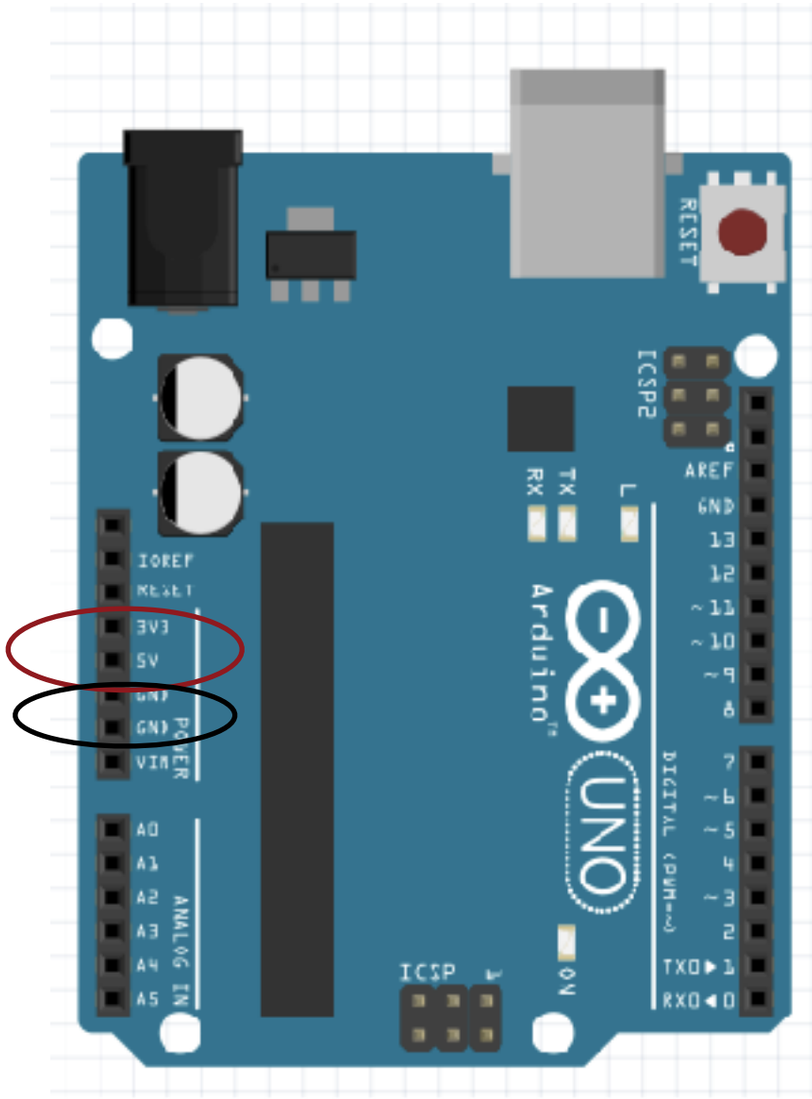
Arduino - Power Pins

Power Pins



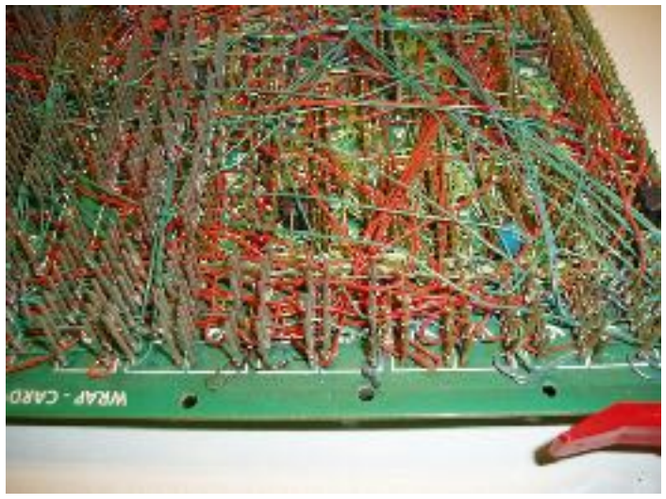
5V / 3.3V:
Power

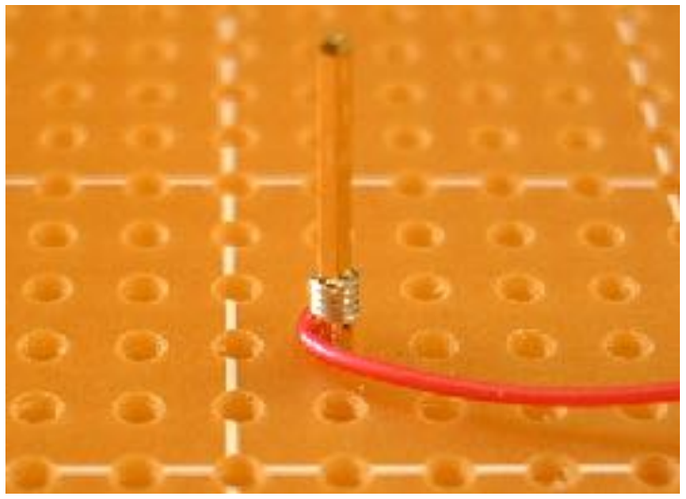
GND:
Ground Pin



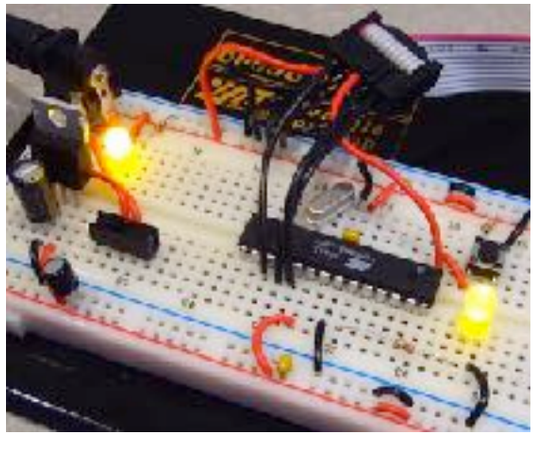
BreadBoard

Wire - Wrap

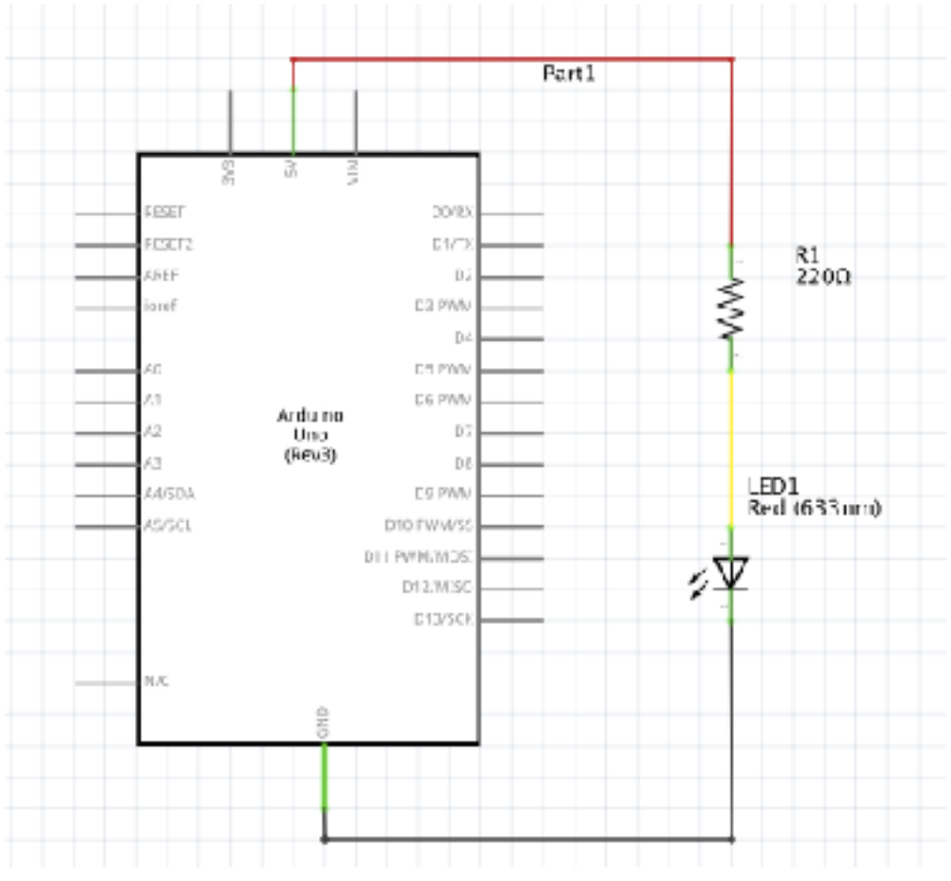


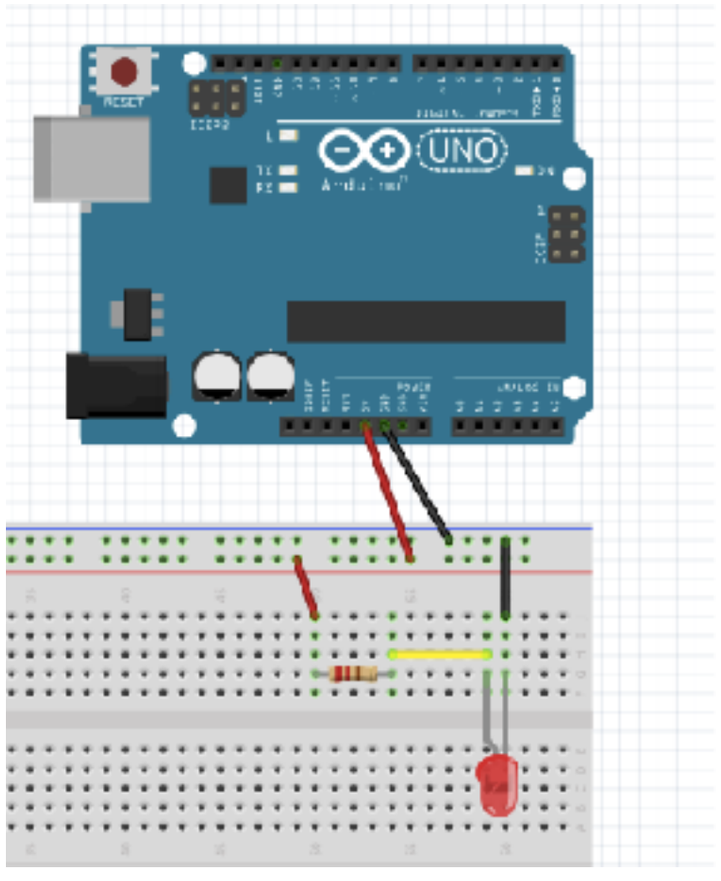


Breadboard

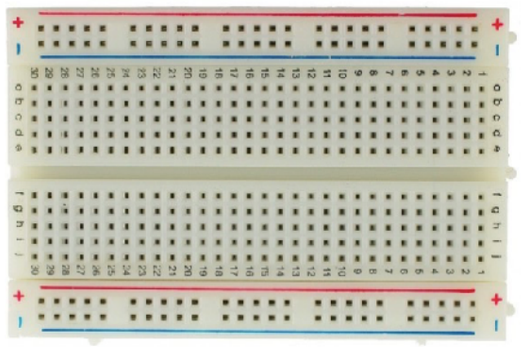


Arduino

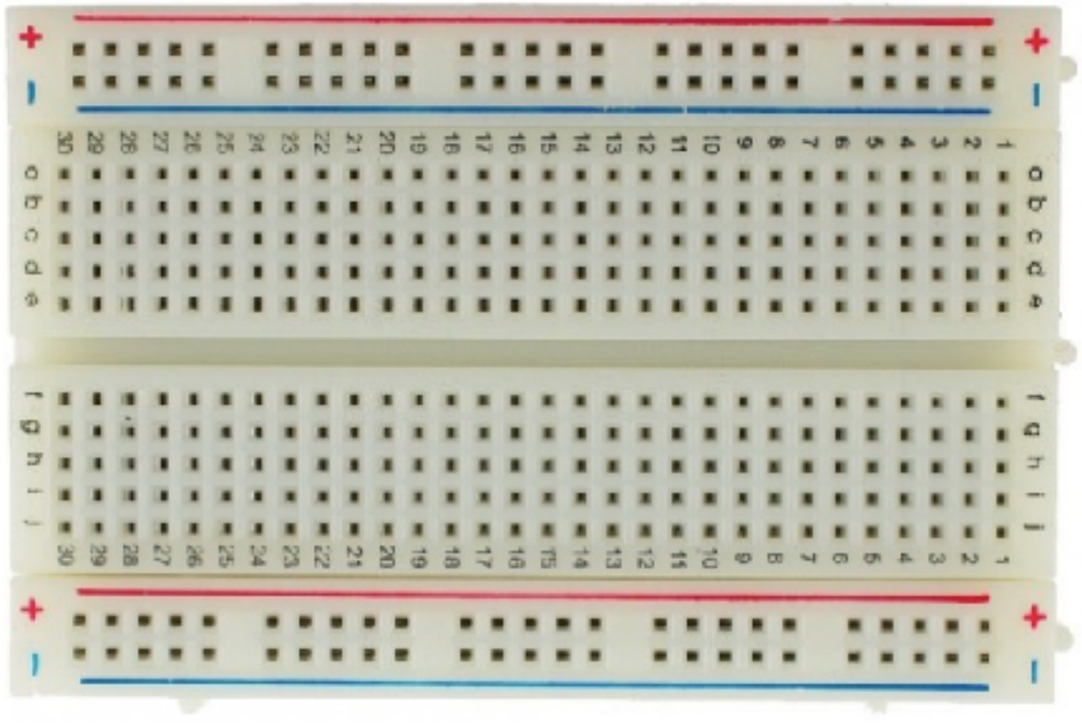

Part1
3V3
5V
VIN
RESET
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
NC
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
Arduino
Uno
(Rev3)
R1
220Ω
LED1
Red (633nm)

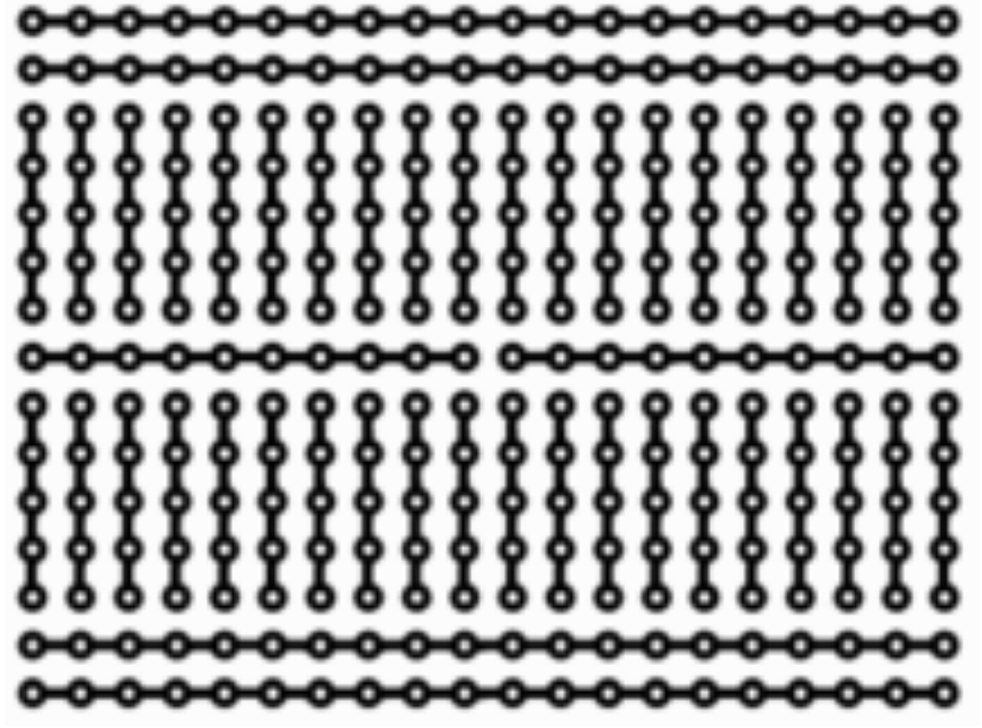

RESET
UNO

BreadBoard

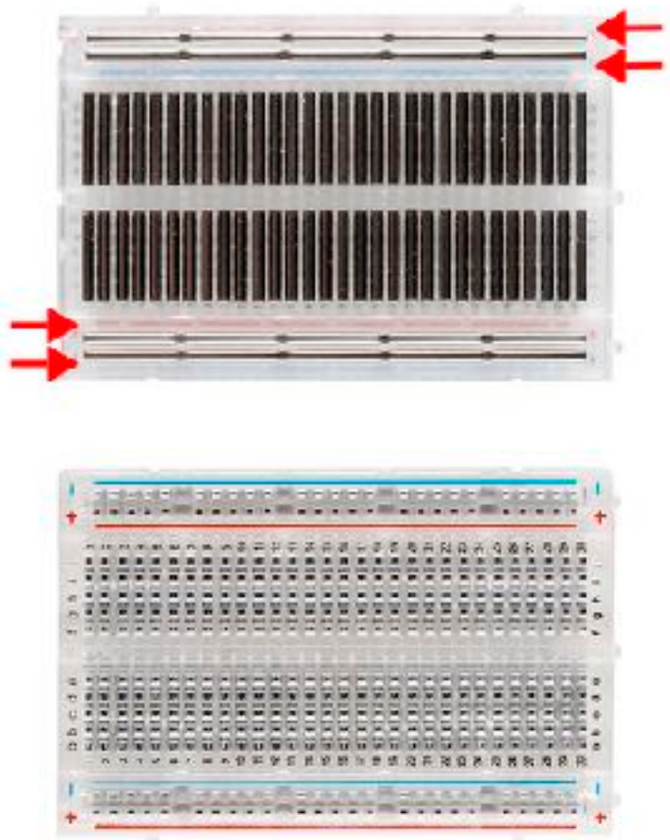


It is a physical support for making temporary circuits and prototyping, and they require absolutely no soldering.

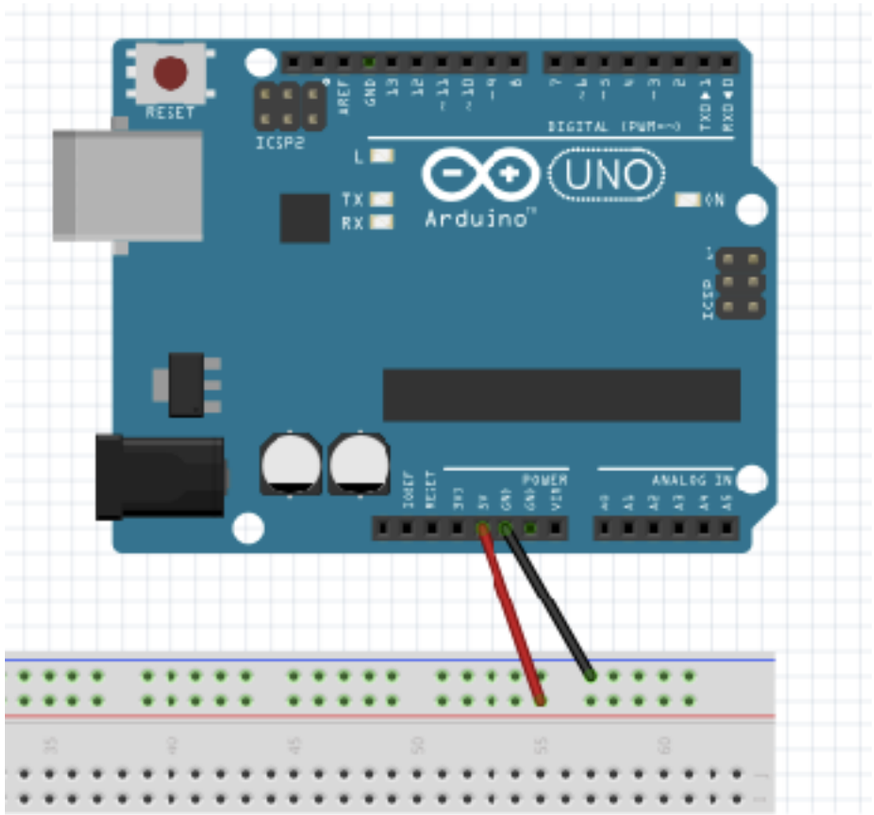




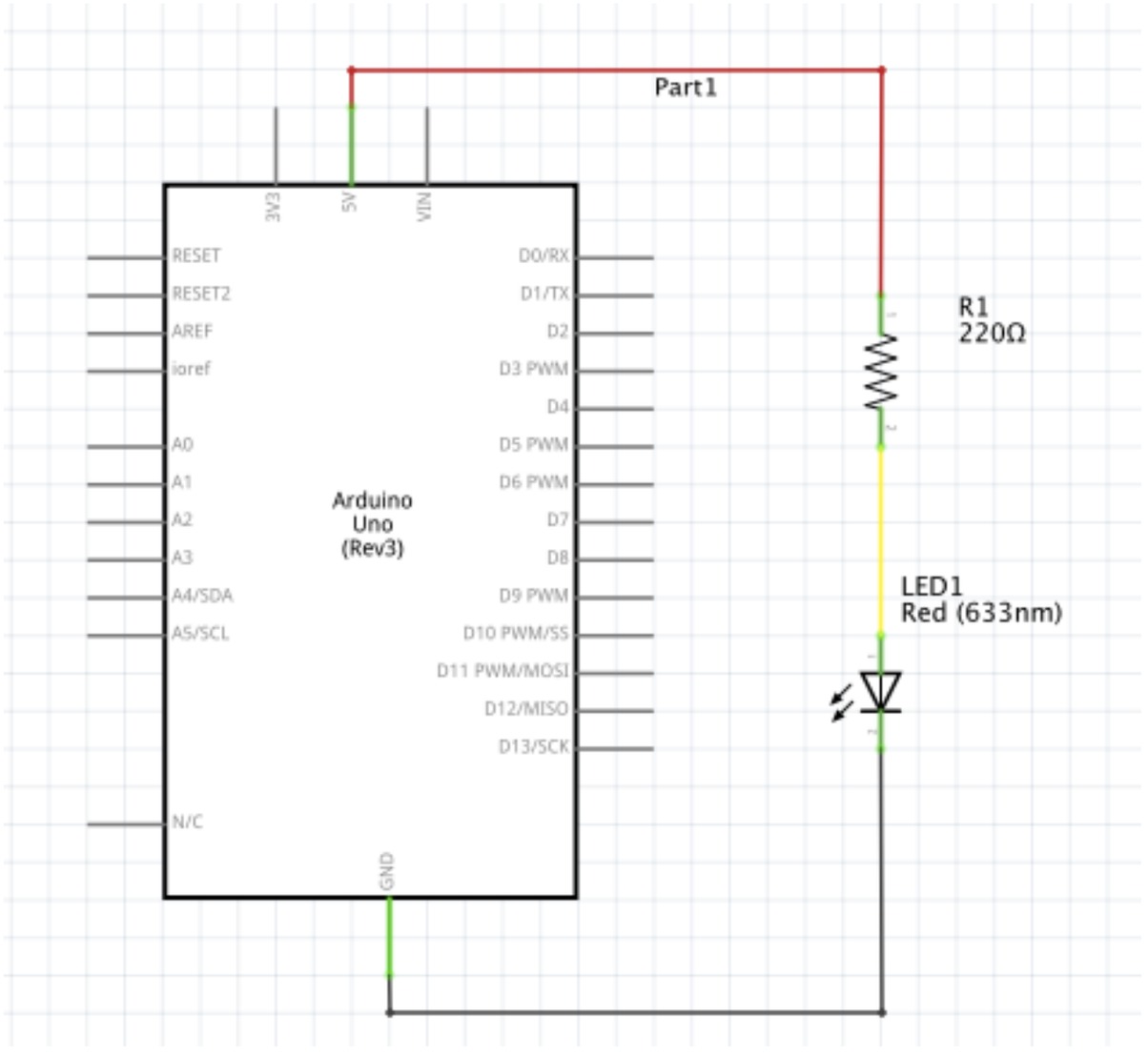
BreadBoard - best practice

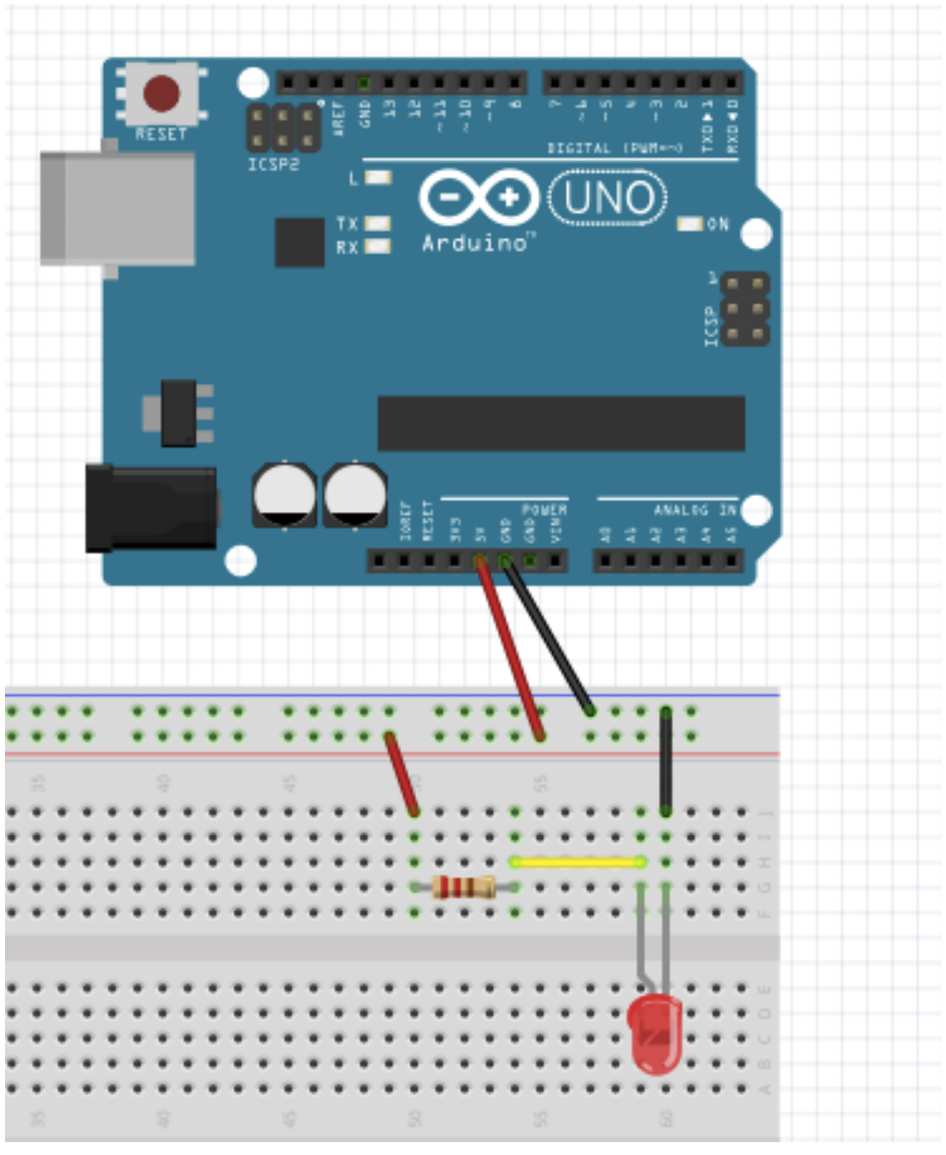


Power Rails

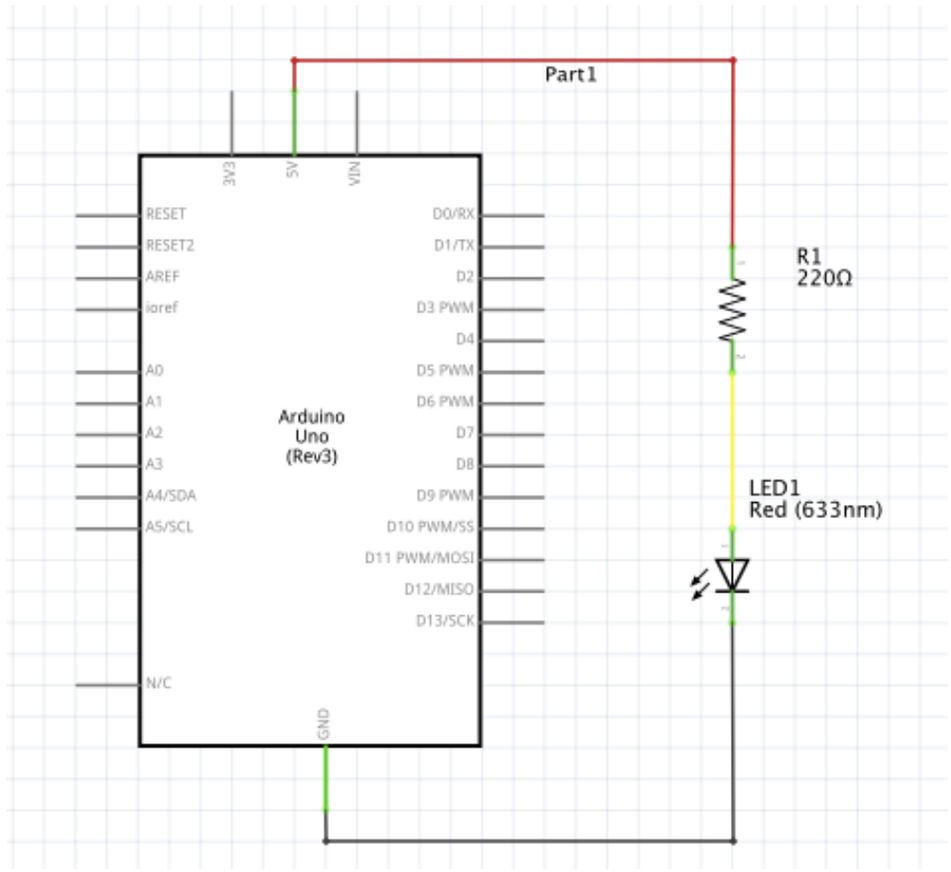


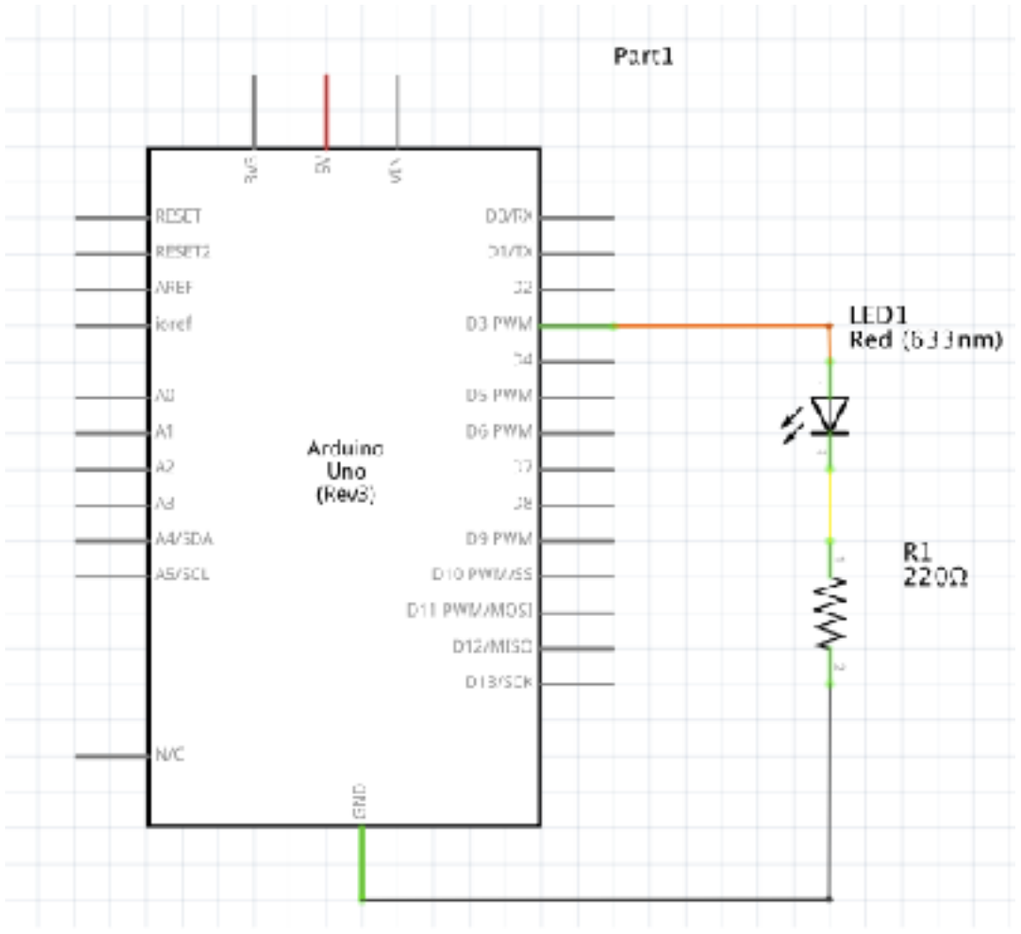
Arduino


Part1
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
R1
220Ω
LED1
Red (633nm)


RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
DIGITAL (PWM~)
POWER
ANALOG IN

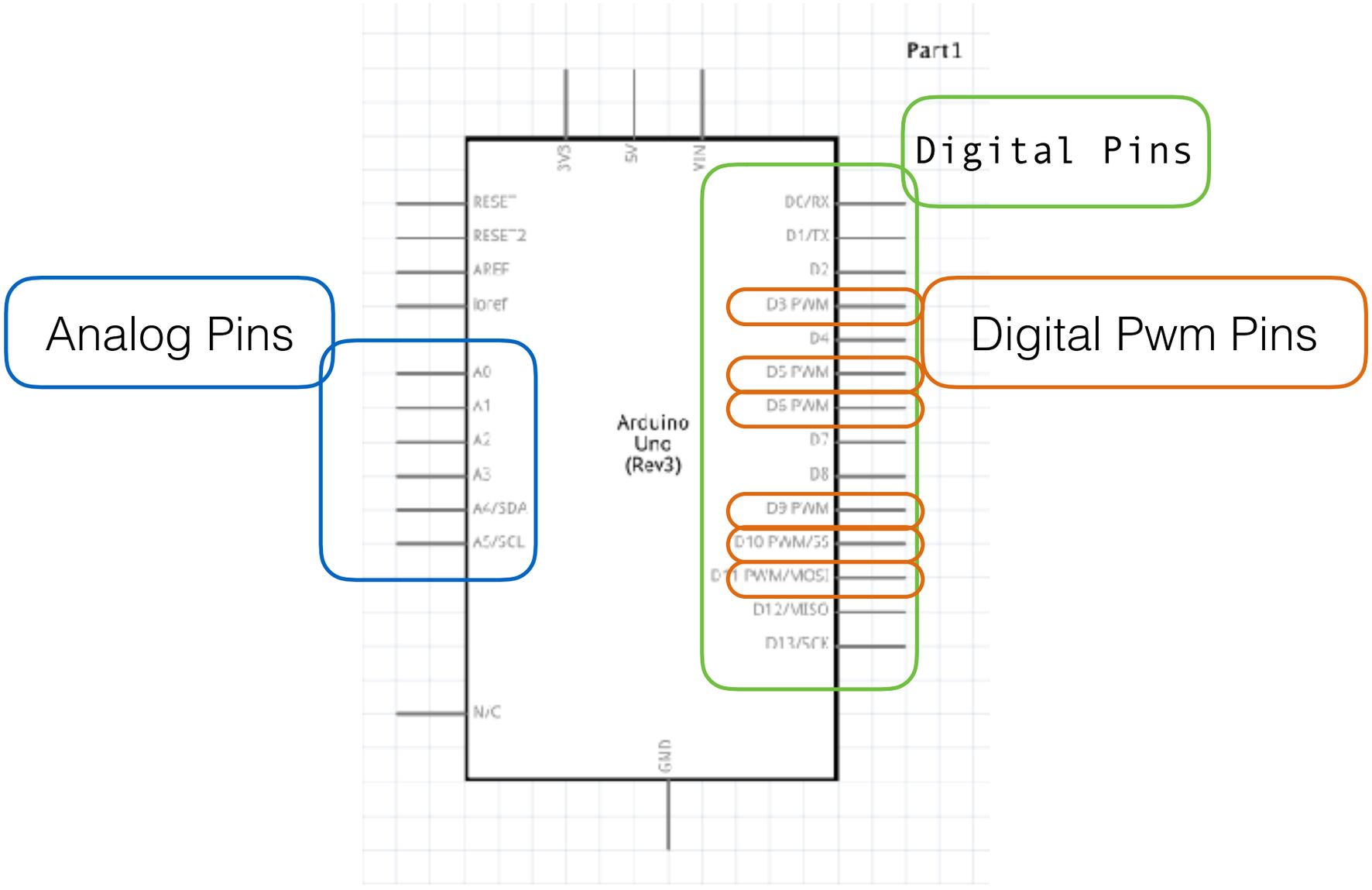
Arduino


Part1
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
Arduino
Uno
(Rev3)
R1
220Ω
LED1
Red (633nm)

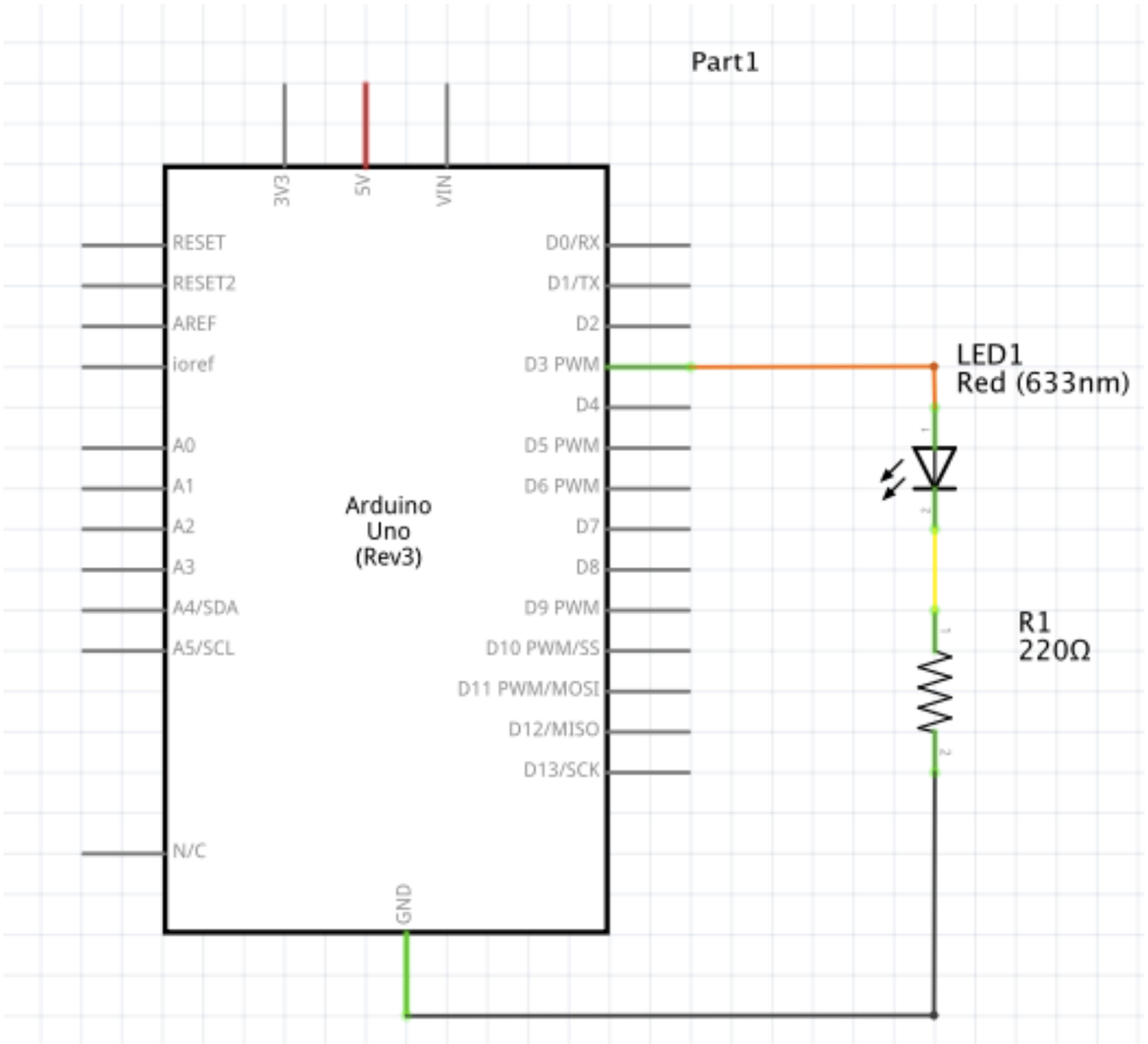

Part1
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
D3 PWM
D5 PWM
D6 PWM
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
GND
Arduino
Uno
(Rev3)
LED1
Red (633nm)
R1
220Ω

Input/Output Pin

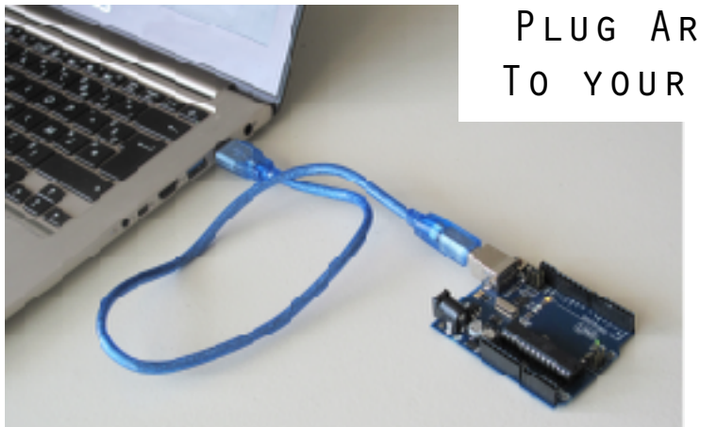
Pin IO (Input/Output)



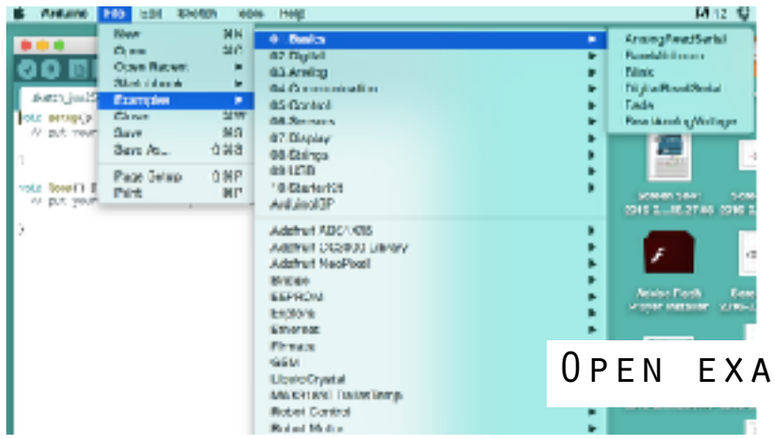
Control a Led with Arduino



But…check if the environment works



Plug Arduino
To your laptop



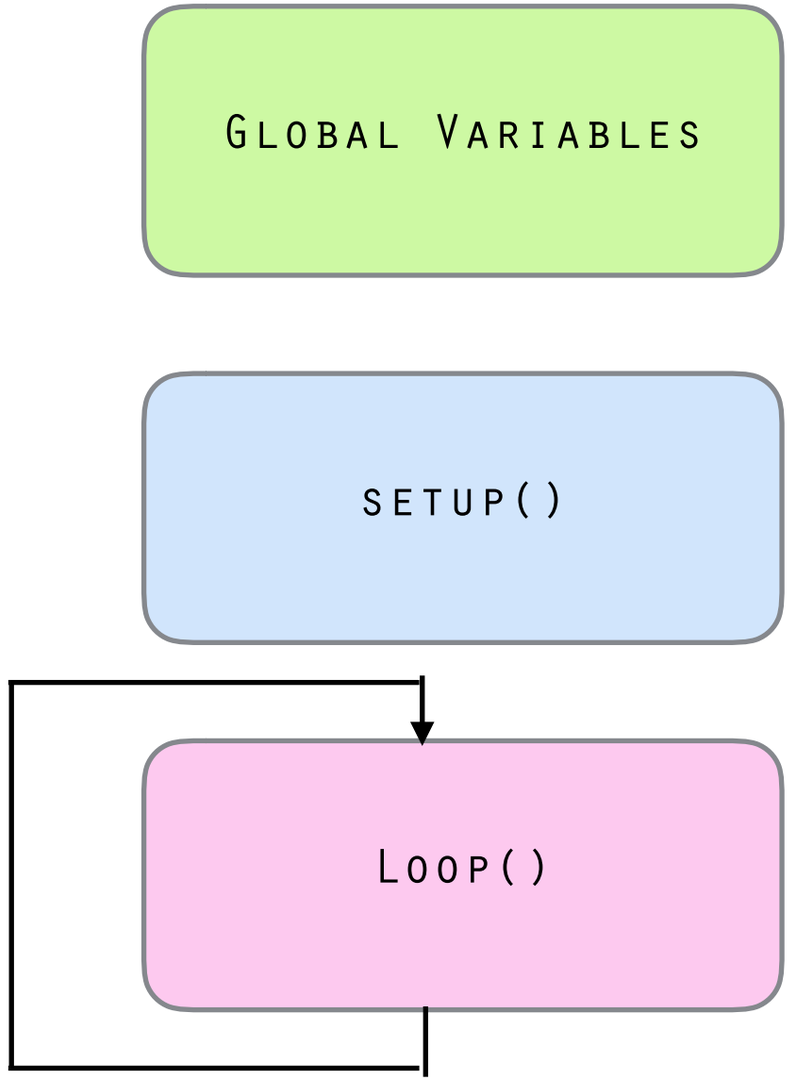
Open example Blink

Select the port
And the Board

Upload the code
To

If it works, the led embedded on the board blinks

SKETCH



Declare the variables

Initialise
Runs once, at the beginning
Define the pins

Running
Run repeatedly, after setup

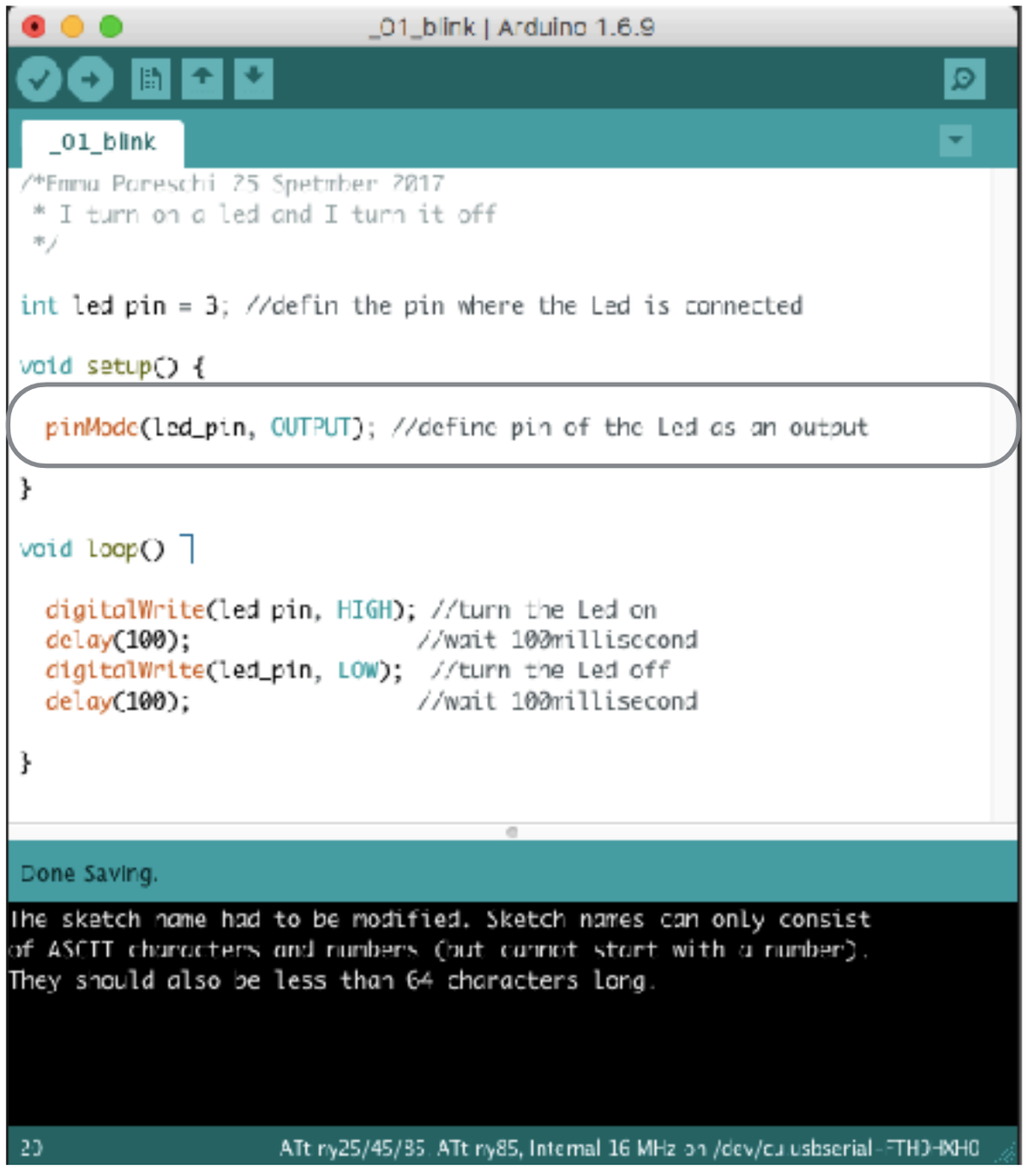
Blink Sketch

Global Variable ‘led_pin’

Blink Sketch

setup()
Define the function of the pin
pinMode(pin, function);

The pin is 'led_pin
The function is Output



Blink Sketch

loop()

The list of commands that arduino runs repeatedly.

digitalWrite(pin, level);

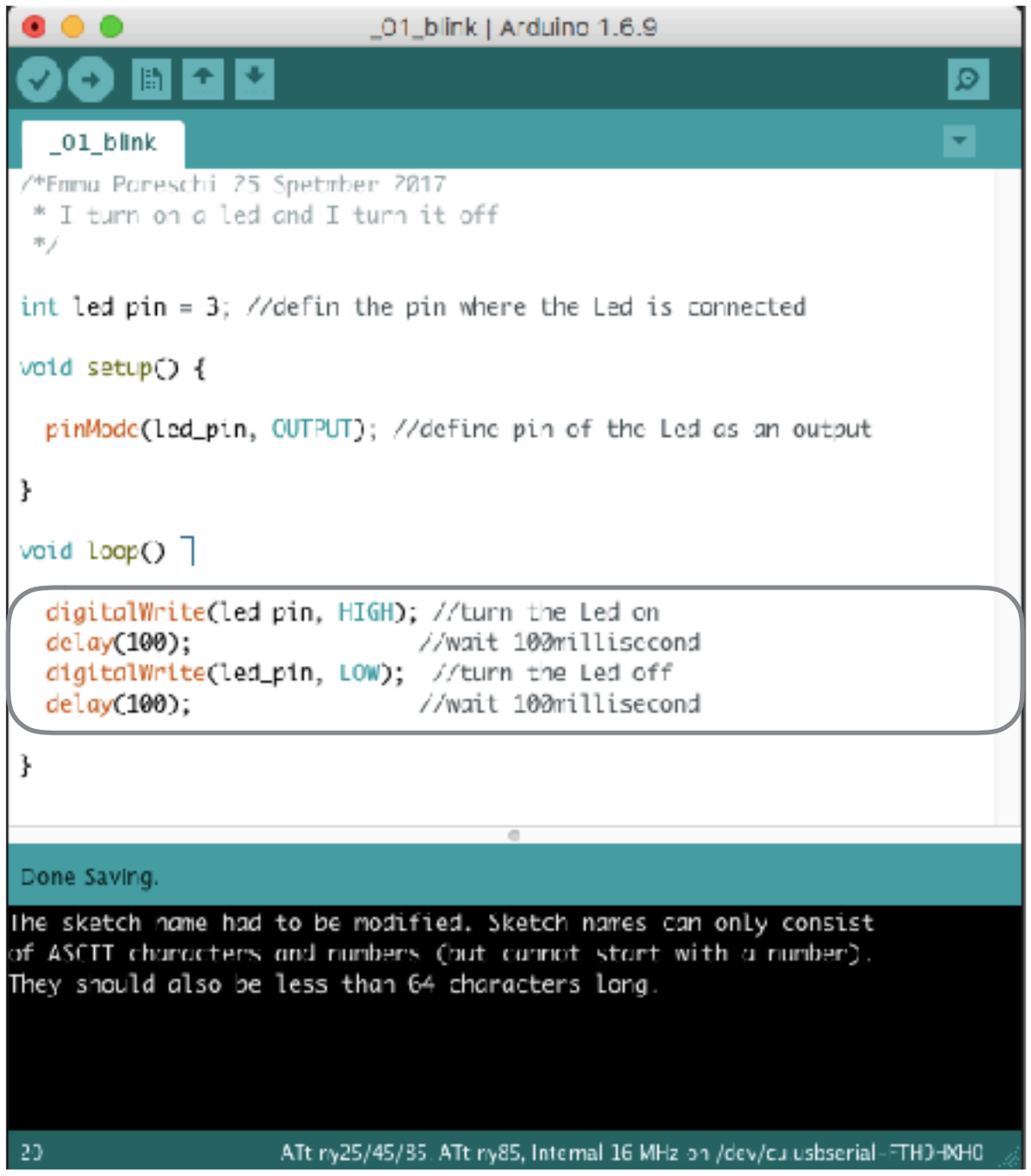
We control the voltage on the pin

HIGH: high voltage (5V)

LOW: low voltage (ground)

delay(time);

We add a delay in millisecond



Blink Sketch

Global variables

setup()

Loop()

_01_blink | Arduino 1.6.9

_01_blink

```
/*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 100milliscond
  digitalWrite(led_pin, LOW);  //turn the Led off
  delay(100);                  //wait 100millisecond

}
```

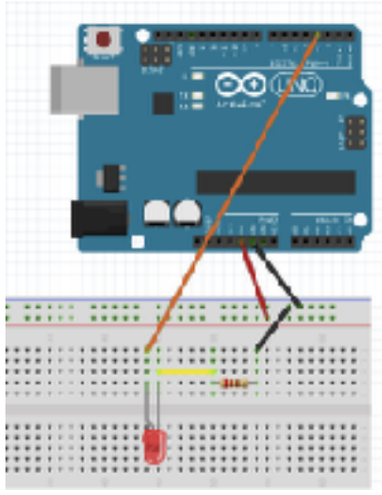
Done Saving.

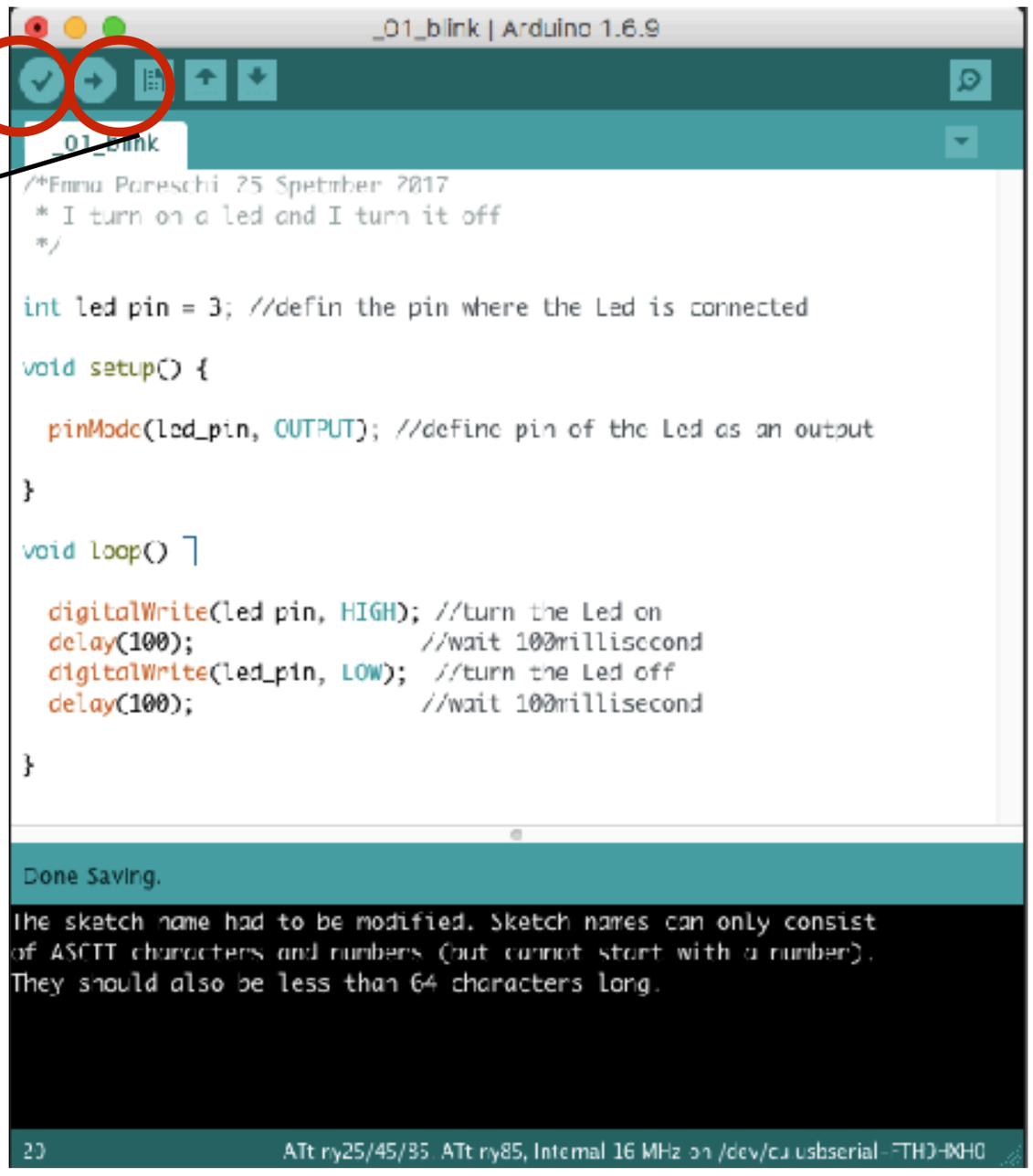
```
The sketch name had to be modified. Sketch names can only consist
of ASCII characters and numbers (but cannot start with a number).
They should also be less than 64 characters long.
```

23 ATtiny25/45/85 ATtiny85, Internal 16 MHz on /dev/cu.usbserial-FTHJ4XH0

Connect, Compile and Upload

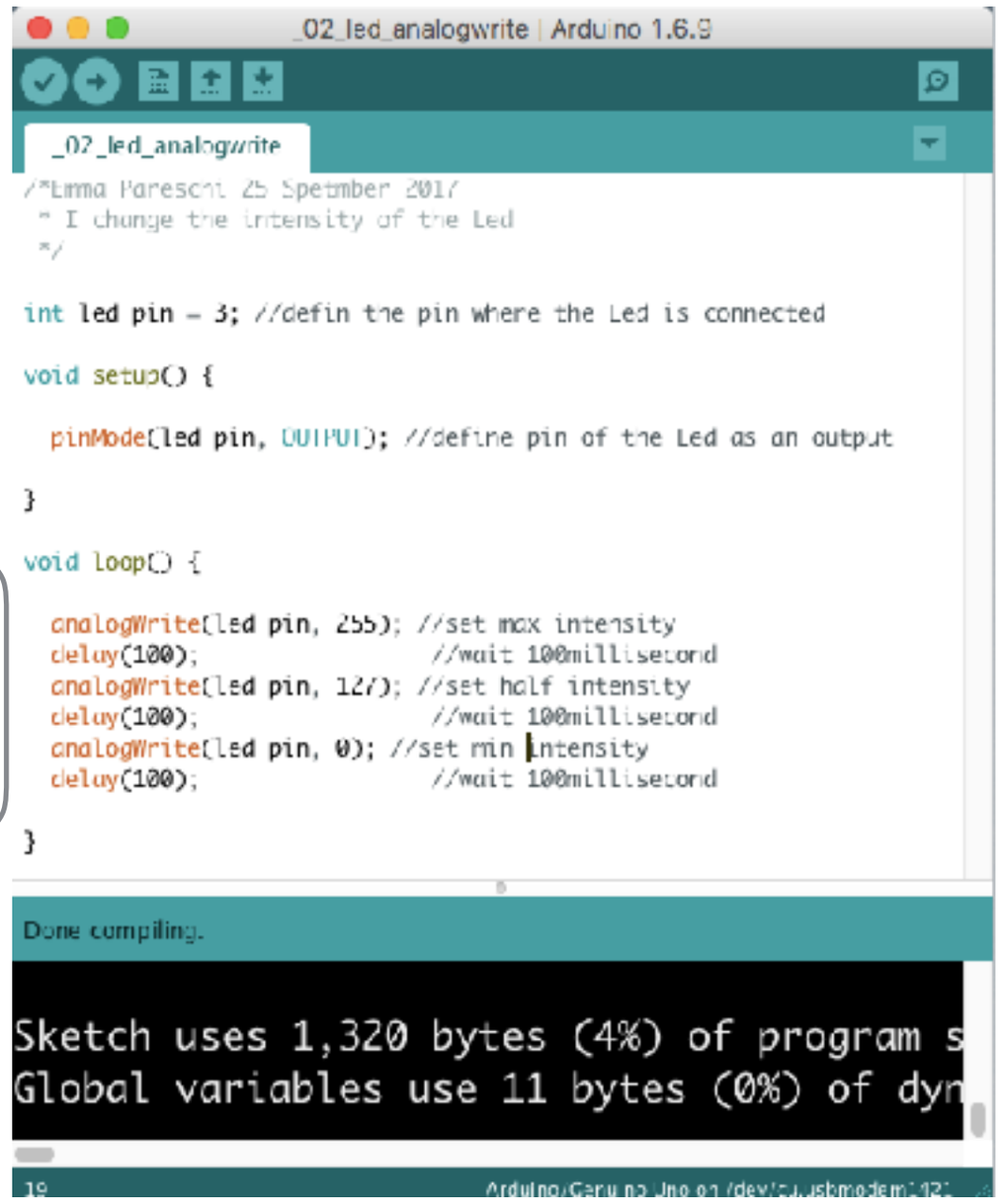
1. Compile
2. Upload





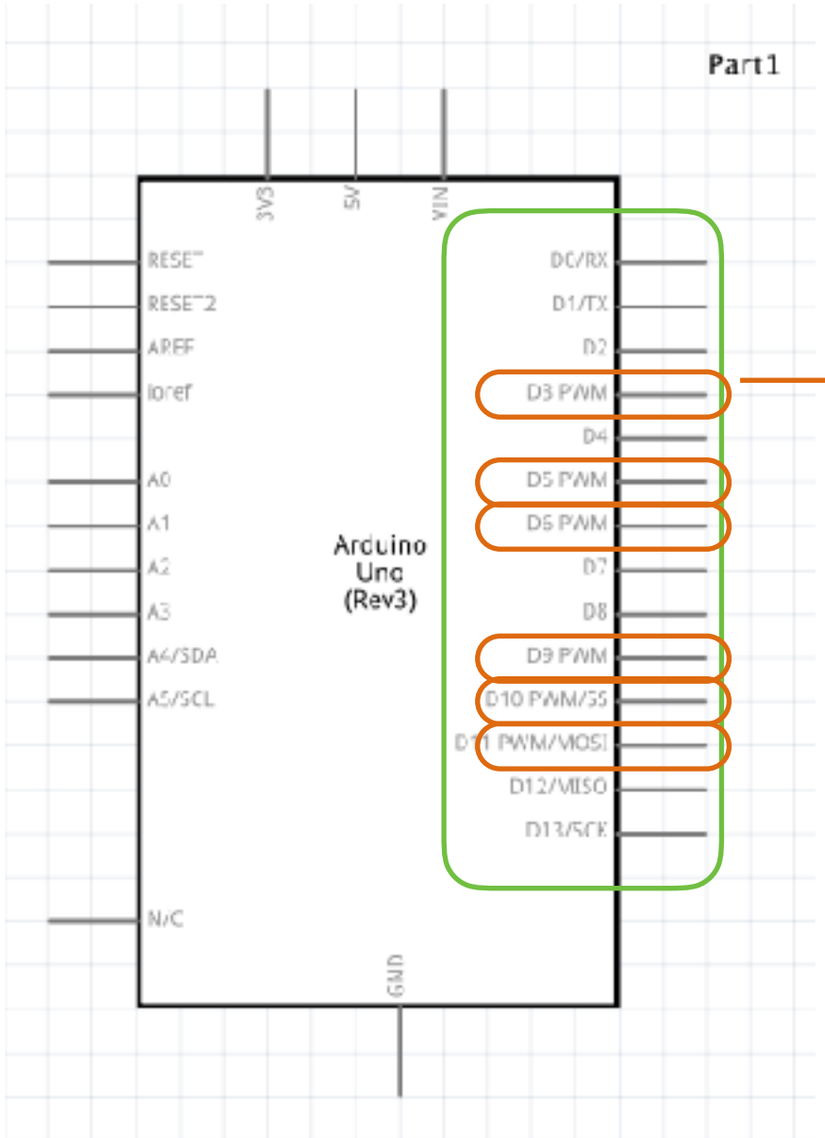
Control the Intensity of the Led

analogWrite(pin, number 0/255);
We control the Voltage on the Pin
255: Max voltage (5V)
0: Min voltage (ground)



PWM Pins

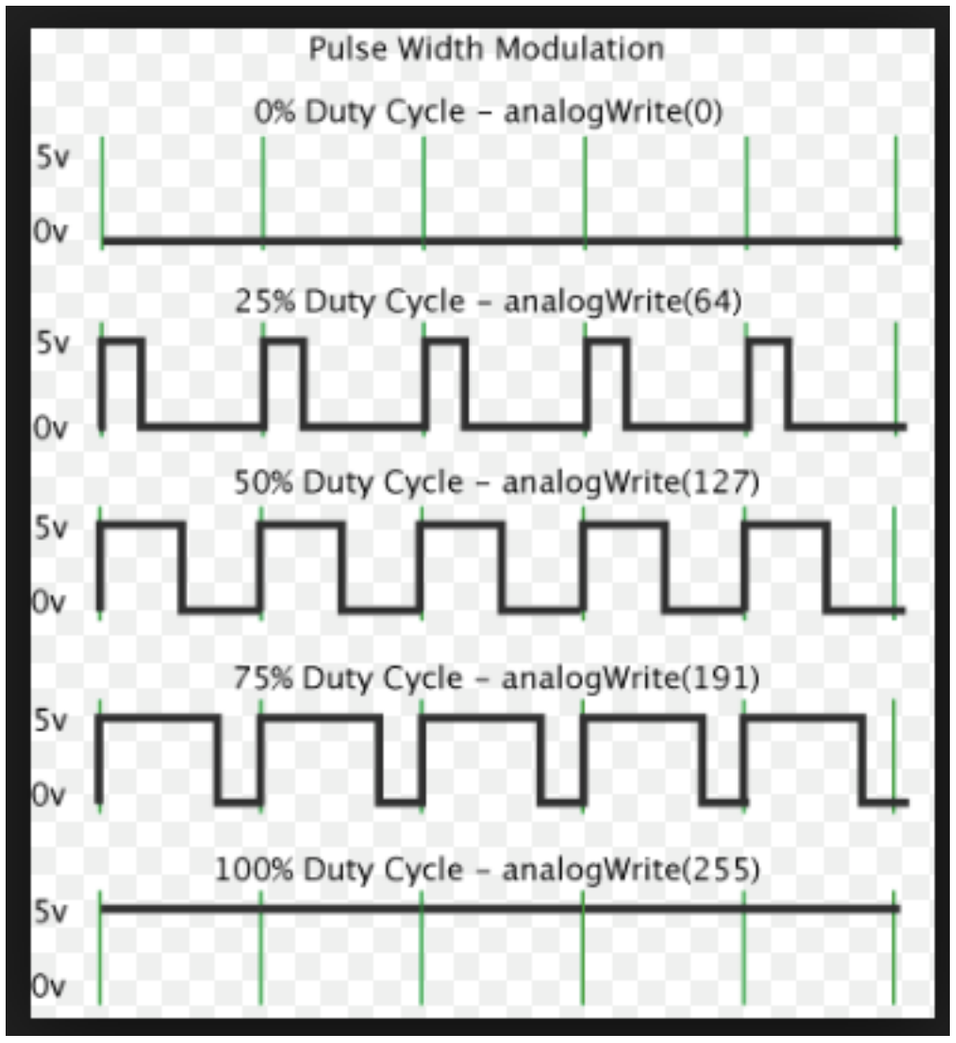
PWM I/O (Input/Output)

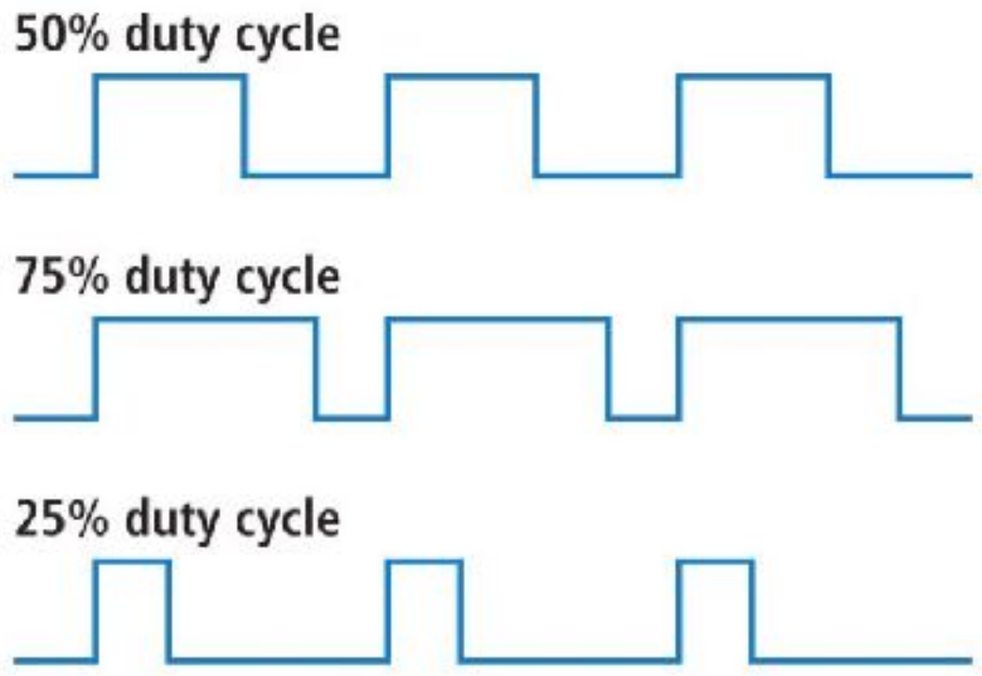


AnalogWrite() works only with some pins: PWM (~)

PWM

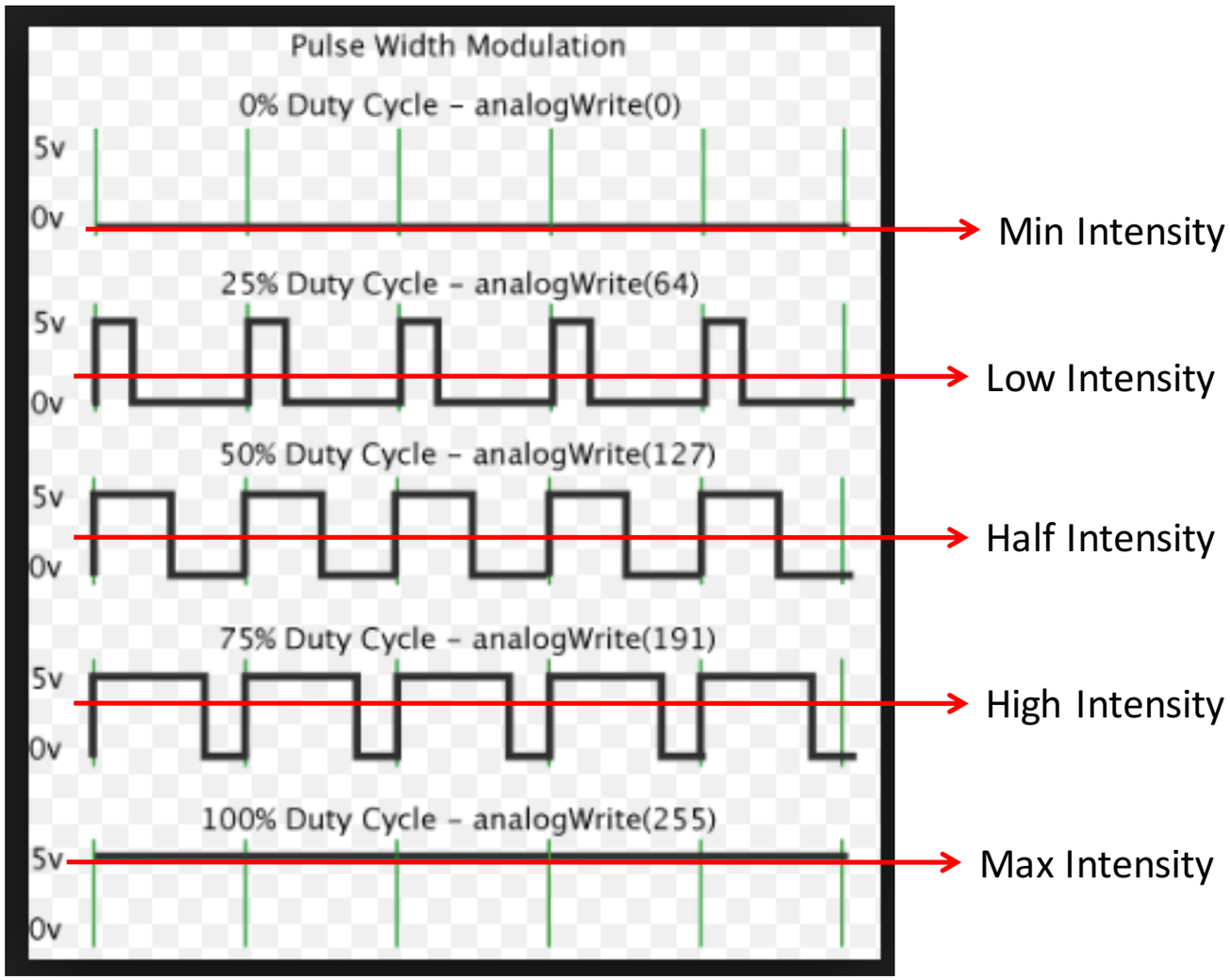
Pulse-width modulation



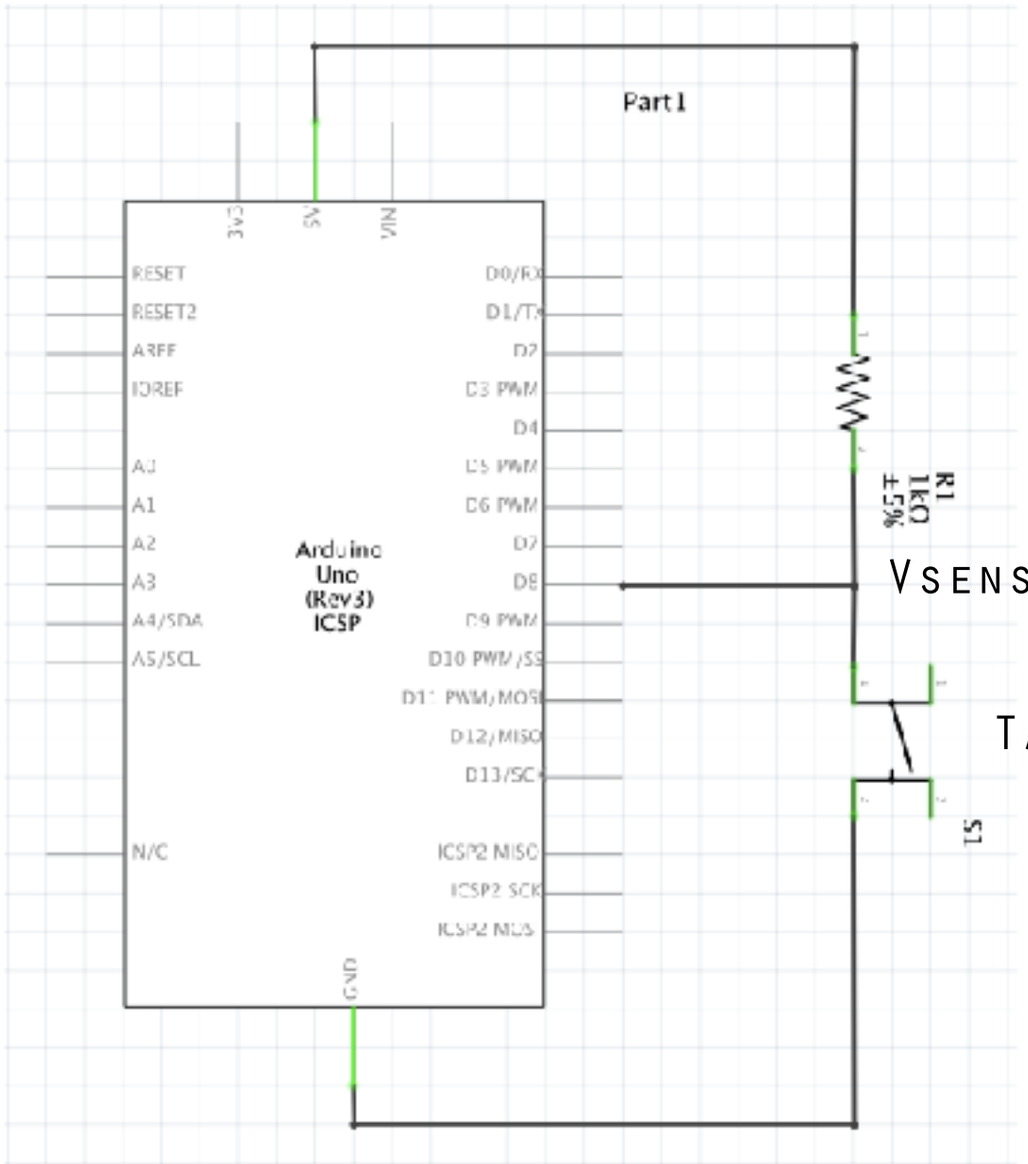


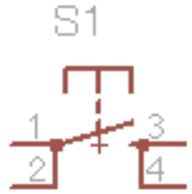
PWM

Pulse-width modulation



How to read a sensor (digital)



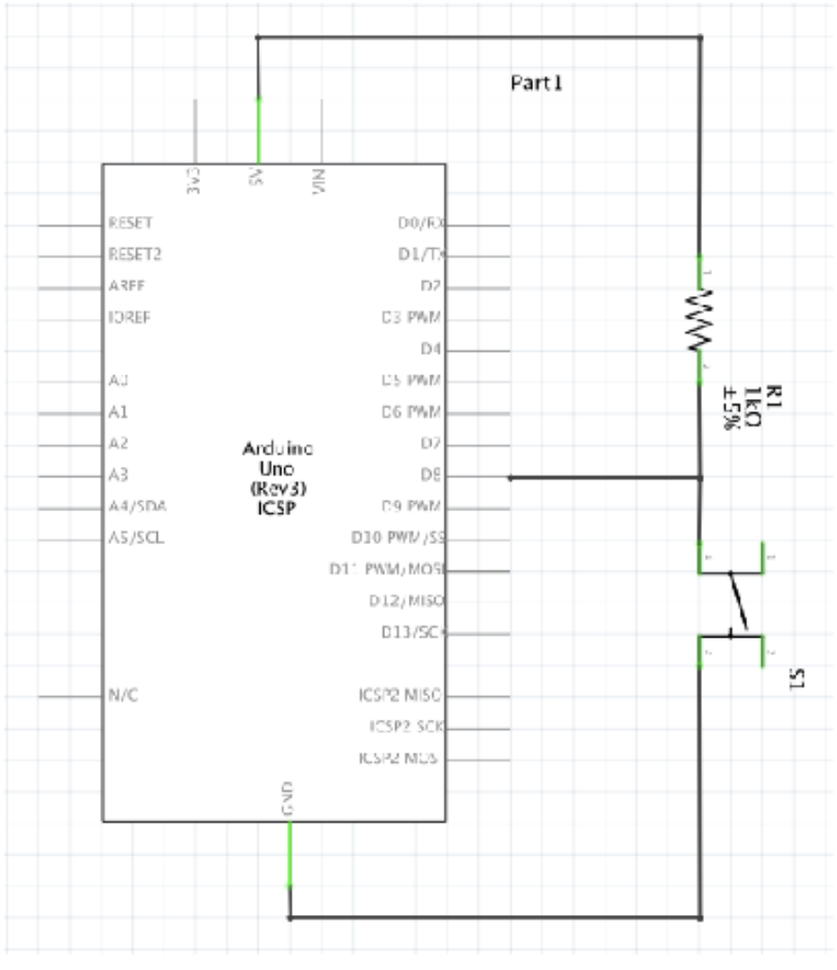


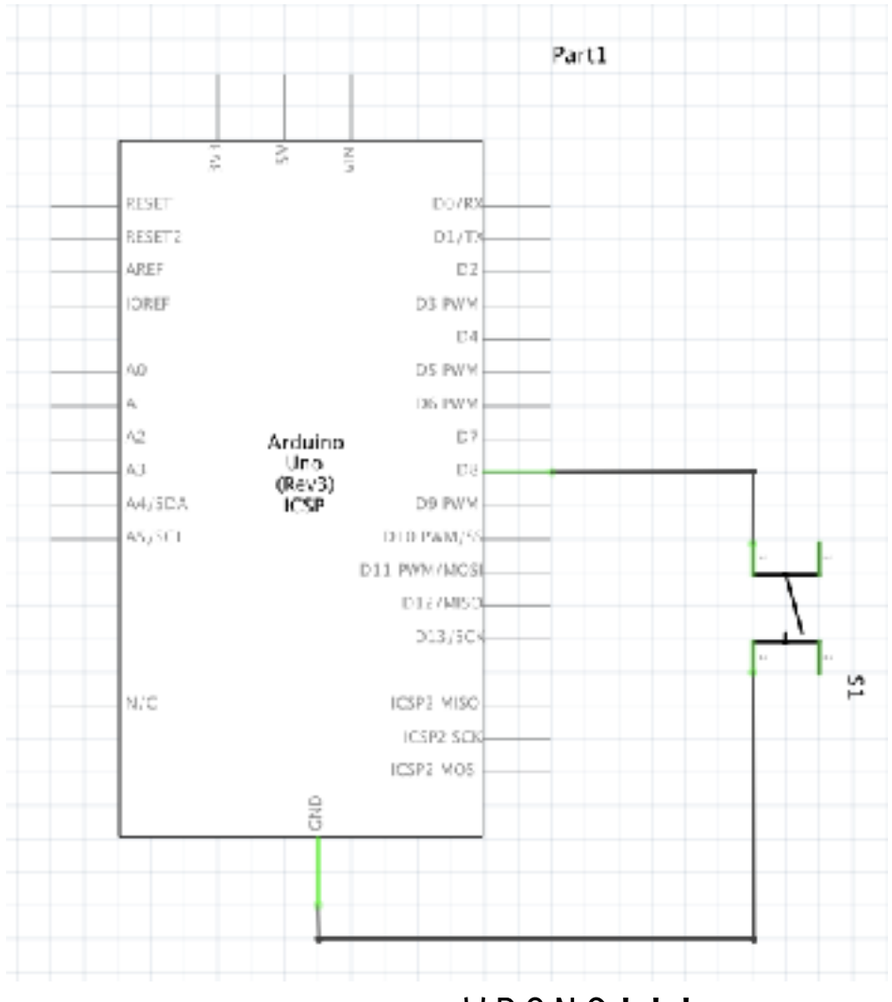
Tact Switch

How it works:
If the switch is NOT pushed
 Vsense = 5V

If the switch is pushed
 Vsense = 0

How to read a sensor (digital)





WRONG!!!

(FOR NOW IT'S WRONG)

Sketch - read a sensor (digital)

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 to print the values

_03_sw | Arduino 1.6.9

_03_sw

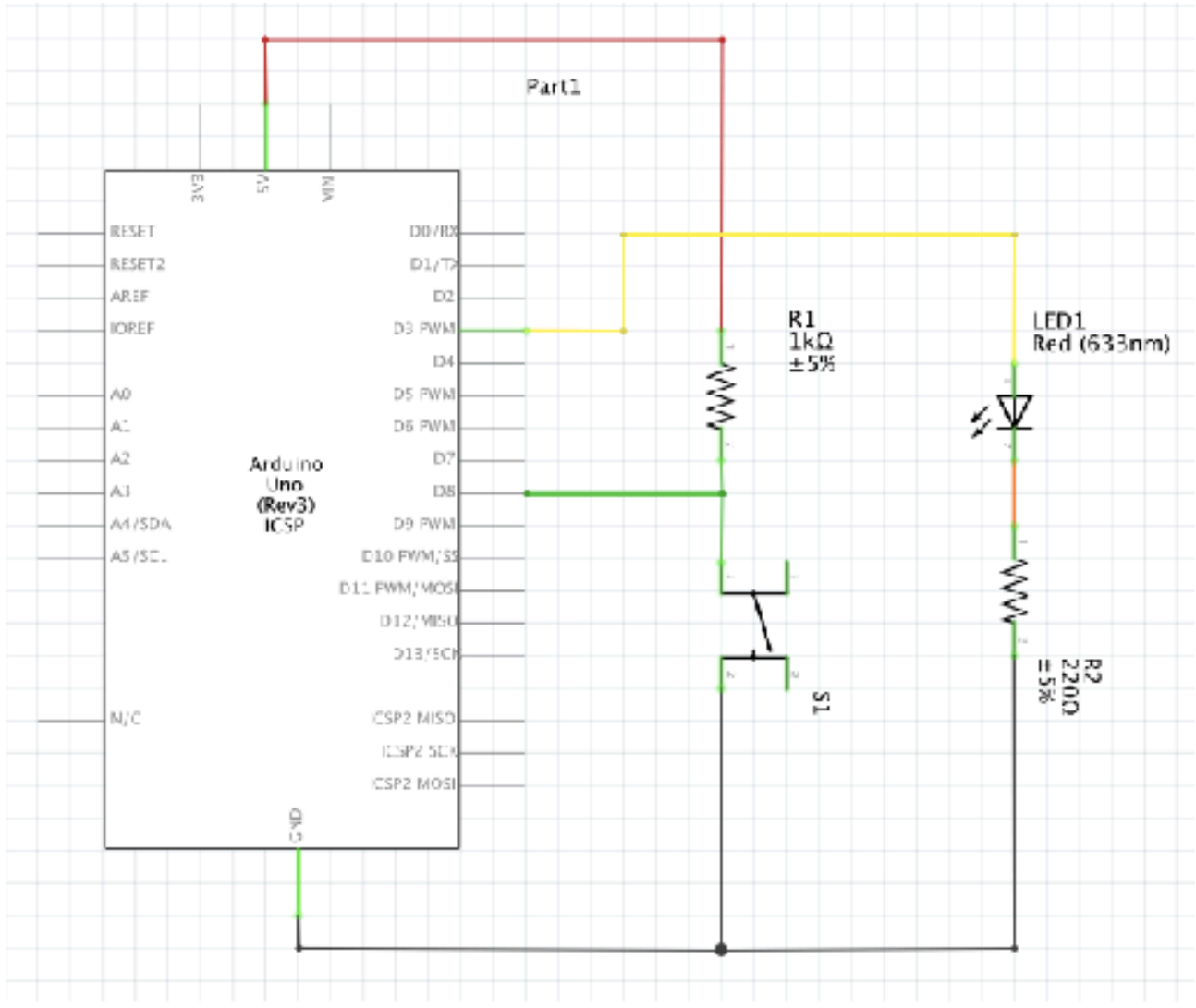
```
/* Emma Pareschi
 *  September 2017
 *  I read and print on the computer
 *  the value of a tact switch (or pushbutton)
 */

// 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(9700);
}

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

Turn on/off the Led based on the switch

..next time..



Sketch - Led and Switch

..NEXT TIME..

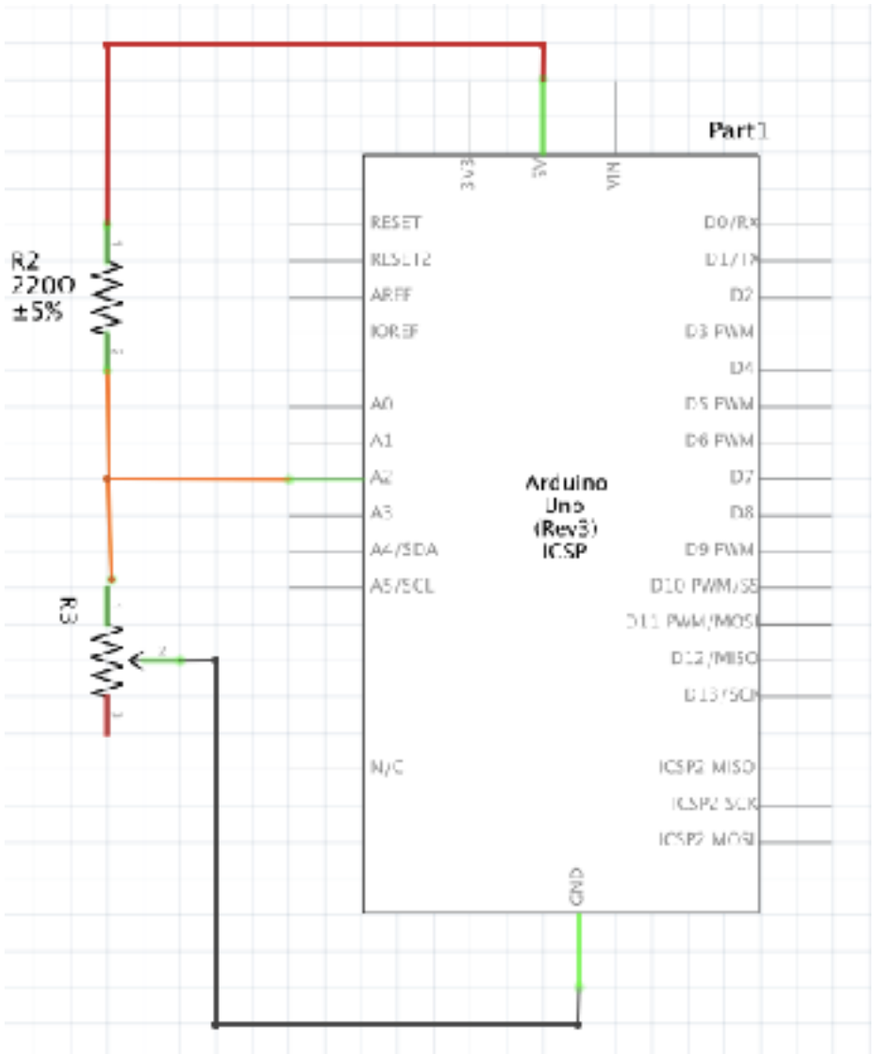
```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin

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

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
}
```

How to read a sensor (Analog)

..next time..



Assignments

1) Design and program a circuit with the following specifications:
   - at least two LEDs
   - use the functions digitalWrite and analogWrite

2) Design and program a circuit with the following specifications:
   - at least one tact switch
   - use the functions digitalRead, Serial.print and Serial.println

   More information about
   Serial.print: https://www.arduino.cc/en/Serial/Print
   Serial.println: https://www.arduino.cc/en/Serial/Println

3) Document the two projects in your blog with two posts, the documentation includes:
   - a short description of the functionality of the circuit
   - schematic of the circuit
   - sketch (arduino code)
   - if it works: upload a video of the circuit that works
   - if it doesn't work: explain what you did to debug hardware and software

Note: if you take inspiration or you reproduce a circuit/code done by someone else, you have to acknowledge the source in your post.

OnLine Calculators

http://www.ohmslawcalculator.com/ohms-law-calculator

http://www.ohmslawcalculator.com/led-resistor-calculator

http://www.ohmslawcalculator.com/voltage-divider-calculator

Sources and Licence

Ohm's Law:
https://learn.sparkfun.com/tutorials/voltage-current-resistance-and-ohms-law

Electric Power:
https://learn.sparkfun.com/tutorials/electric-power

Resistors (pg.75)
Encyclopedia of Electronic Components Volume 1, 1st Edition,
Charles Platt

Leds (pg. 205)
Encyclopedia of Electronic Components Volume 2, 1st Edition,
Charles Platt.

Arduino commands:
https://www.arduino.cc/en/Reference/HomePage