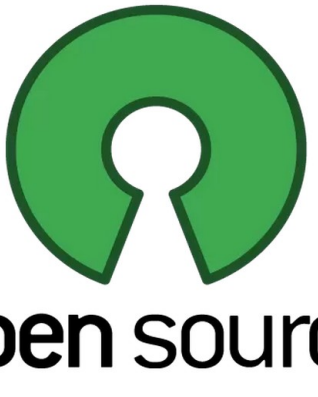


Software libre, hardware libre

Alberto Labarga – Experimental Serendipity S.L.

Laboratorio de Fabricación Digital, Mutilva, 7 de Febrero de 2014

http://www.apptivismo.org/laboratorio-fabricacion-digital/

http://arduino.cc/

Arduino Uno

Arduino Leonardo

Arduino GSM Shield

Arduino Robot

Arduino Esplora

Arduino Wireless SD Shield

Arduino Due

Arduino Yún

Arduino Ethernet Shield

Arduino Mega ADK

Arduino Ethernet

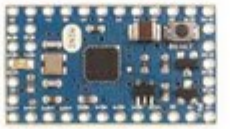
Arduino Motor Shield

Arduino Tre

Arduino Micro

Arduino WiFi Shield

Arduino Mega 2560

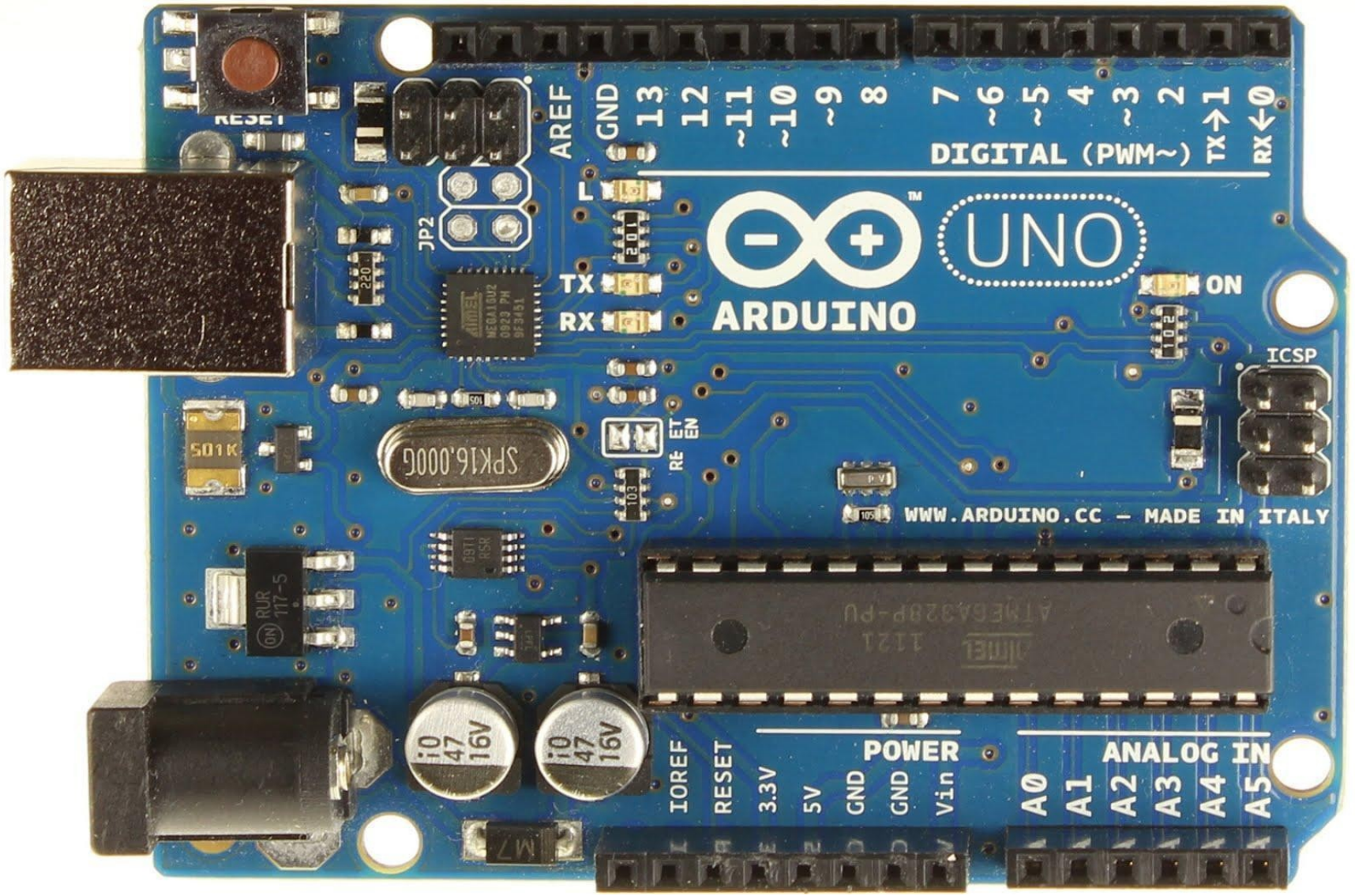


Arduino Mini

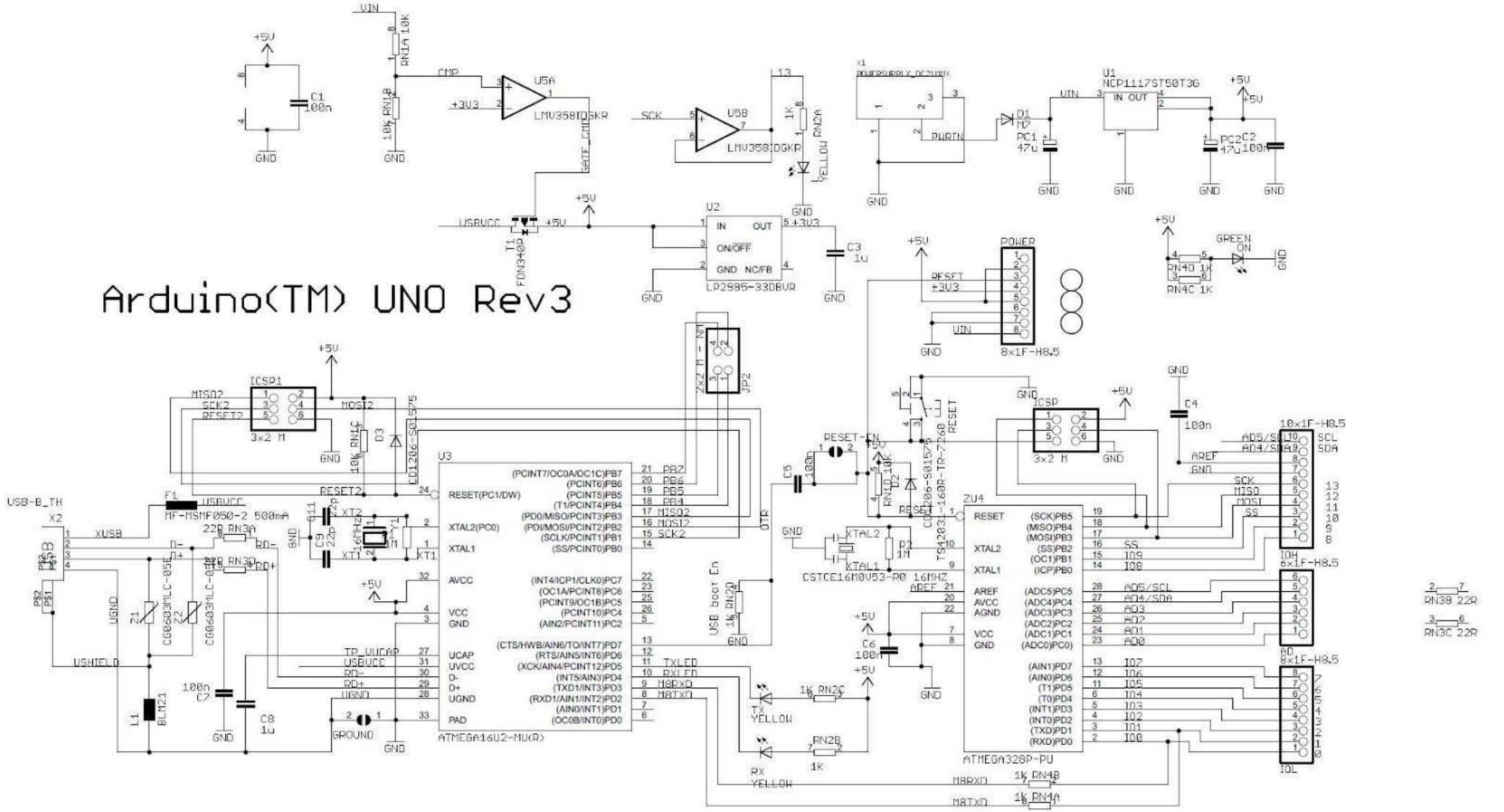
Arduino Wireless Proto Shield

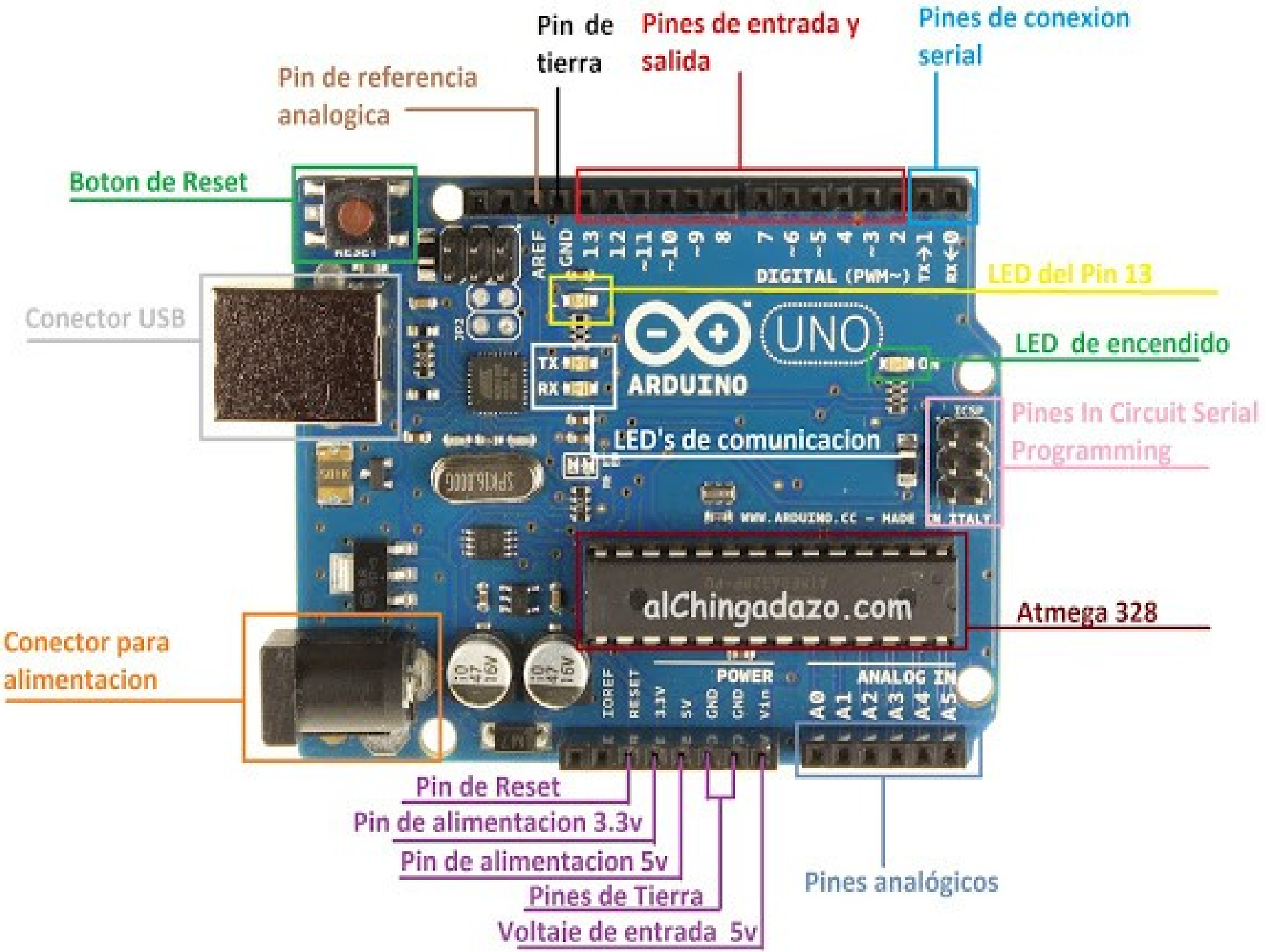
http://arduino.cc/en/Main/Products

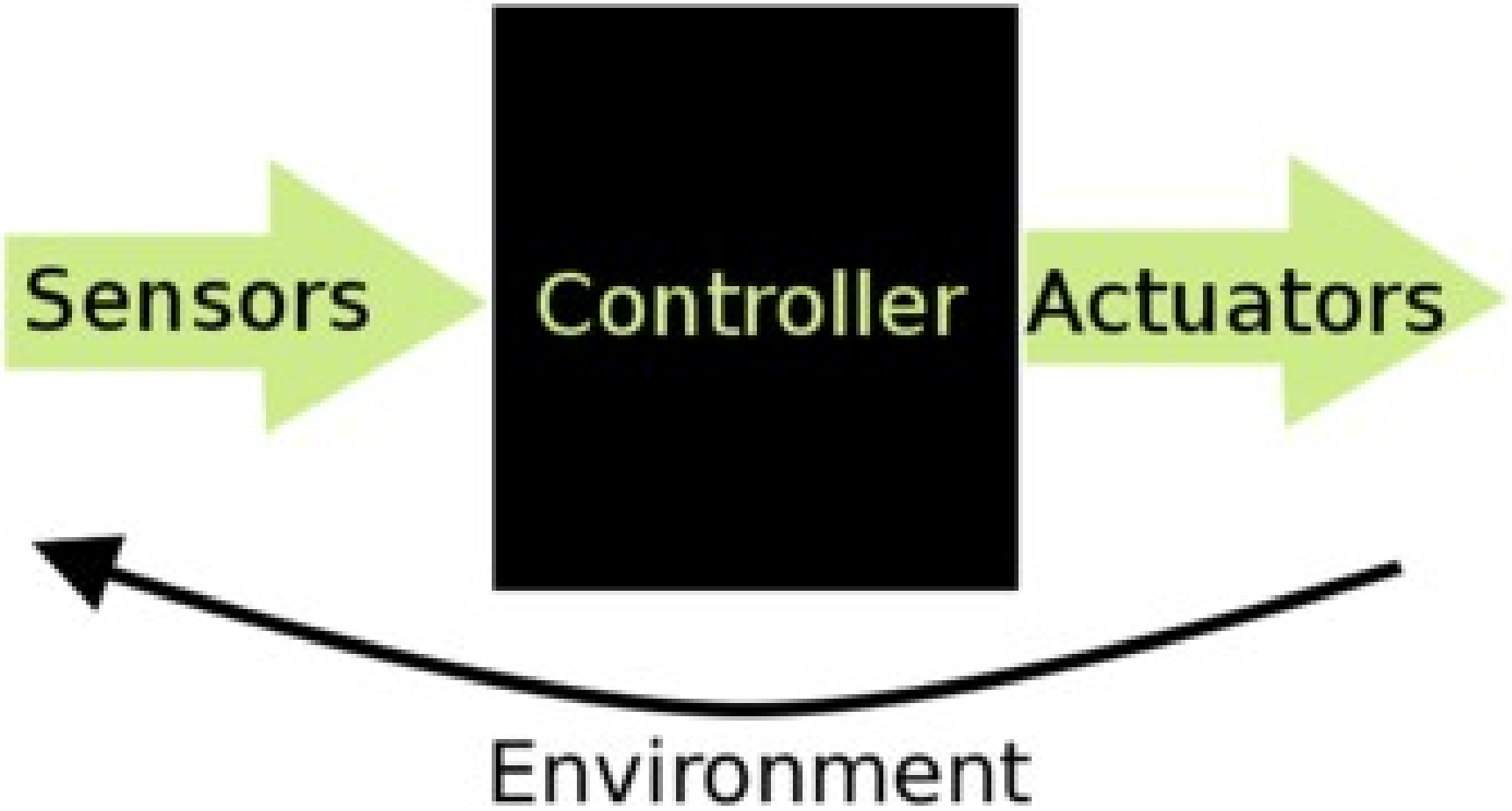
Arduino UNO


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
JP2
TX
RX
ARDUINO
UNO
ON
ICSP
WWW.ARDUINO.CC – MADE IN ITALY
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Hardware

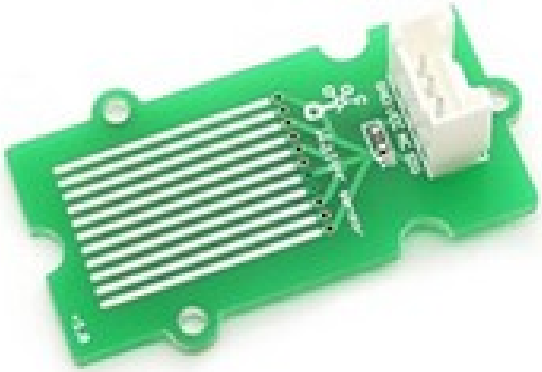

Arduino(TM) UNO Rev3
ATMEGA16U2-MU(R)
ATMEGA328P-PU
LMV358IDGKR
LP2985-330DBVR
NCP1117ST50T3G
TS12031-160R-TR-260
CSTCE16M0V53-R0 16MHZ
USB-B_TH
ICSP1
ICSP
POWER
8x1F-H8.5
10x1F-H8.5
6x1F-H8.5
3x2 M
2x2 M - NM
JP2
RESET-EN
USB boot En
GREEN ON
RN38 22R
RN3C 22R


Pin de tierra
Pines de entrada y salida
Pines de conexion serial
Pin de referencia analogica
Boton de Reset
Conector USB
LED del Pin 13
LED de encendido
Pines In Circuit Serial Programming
LED's de comunicacion
Atmega 328
Conector para alimentacion
Pin de Reset
Pin de alimentacion 3.3v
Pin de alimentacion 5v
Pines de Tierra
Voltaie de entrada 5v
Pines analógicos
ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN
alChingadazo.com


Sensors
Controller
Actuators
Environment

Sound

Sharp IR Sensor



Humidity

Photocell (light sensor)

Rotation sensor
(potentiometer)

PIR

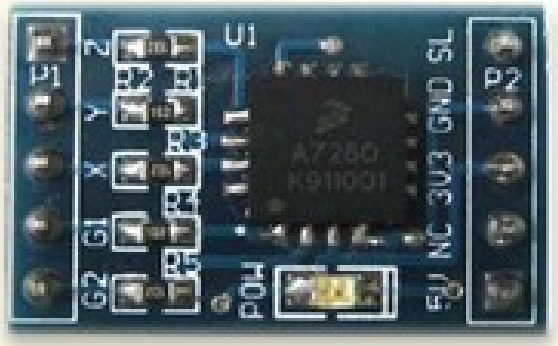
Temperature

Polar Heart Rate Sensor

AC Current Sensor

Co

CO2 Sensor

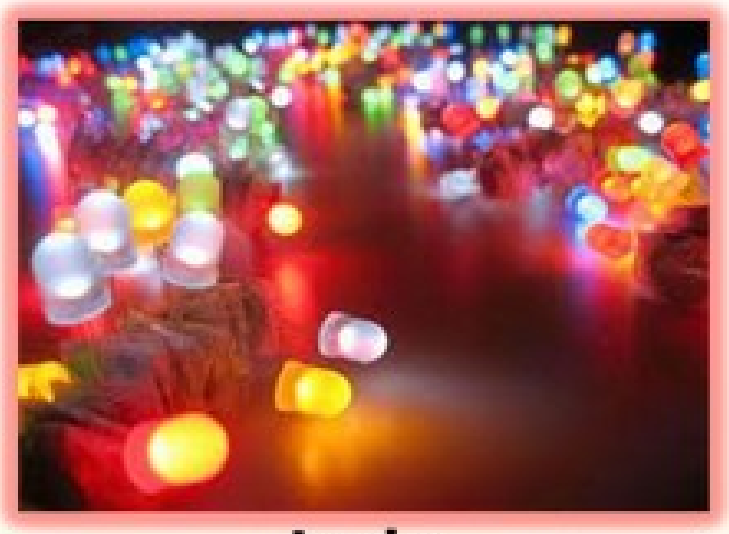


Tri-Axis Accelerometer

Pressure

Motors

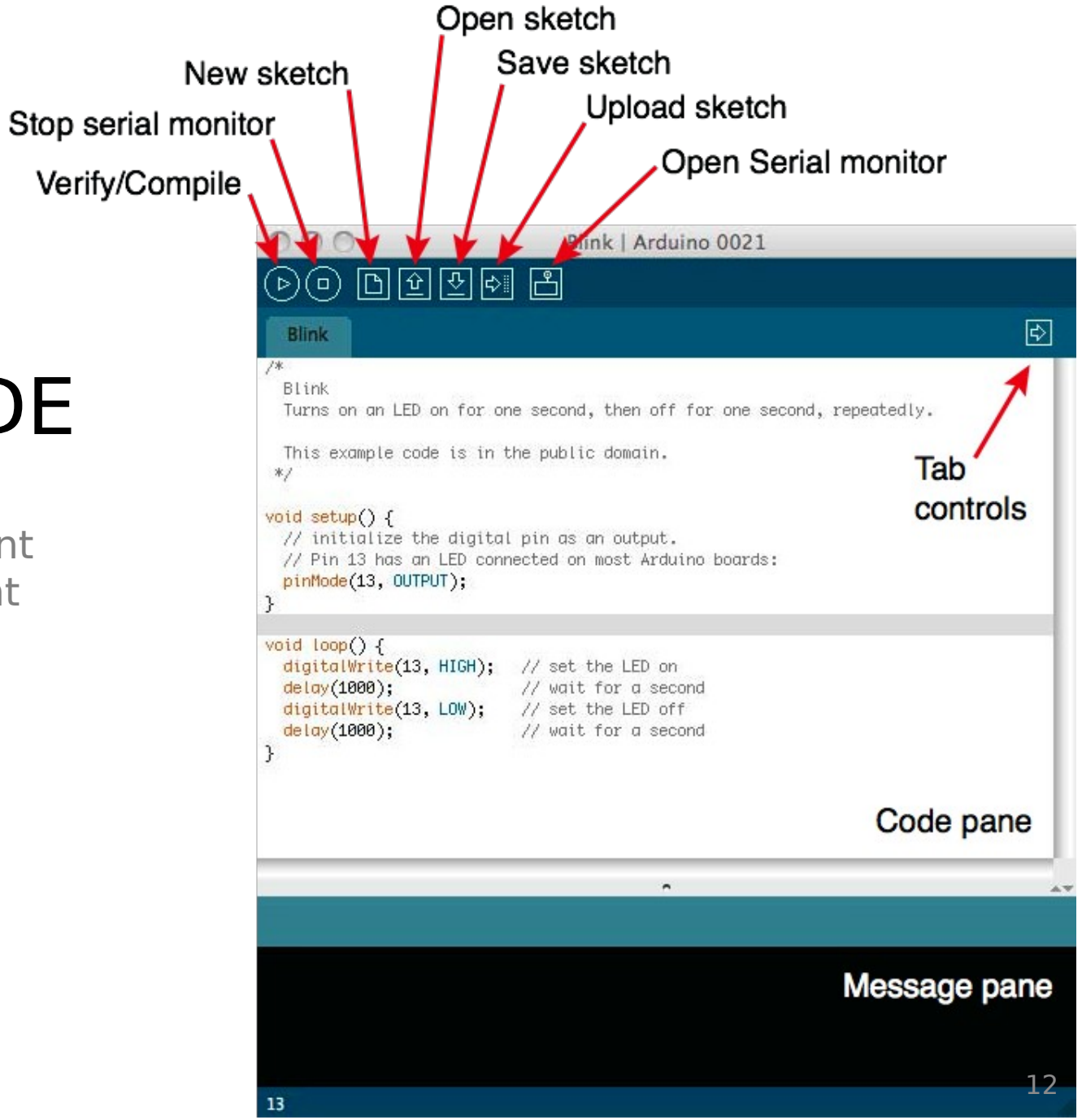
Piezo Buzzer



Leds

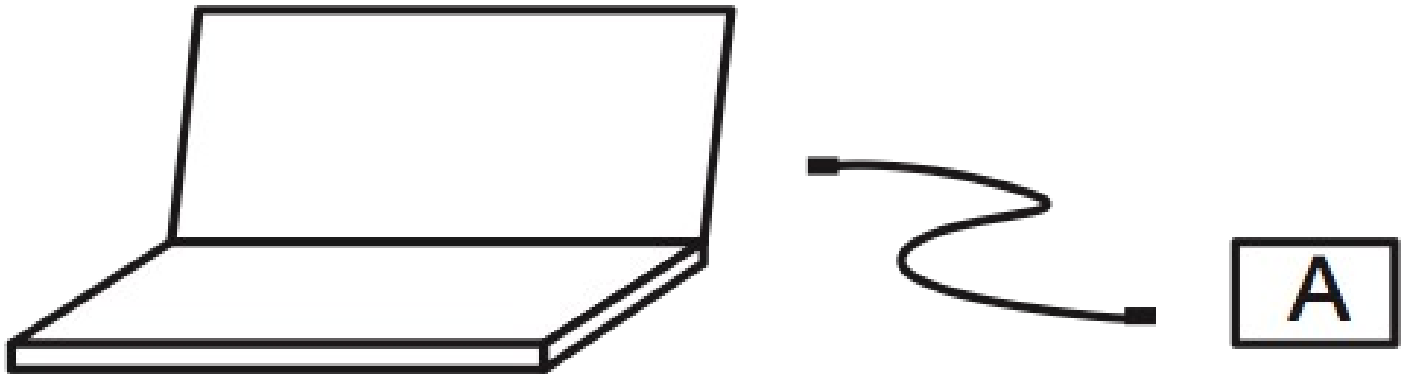
Arduino IDE

IDE = Integrated Development Environment

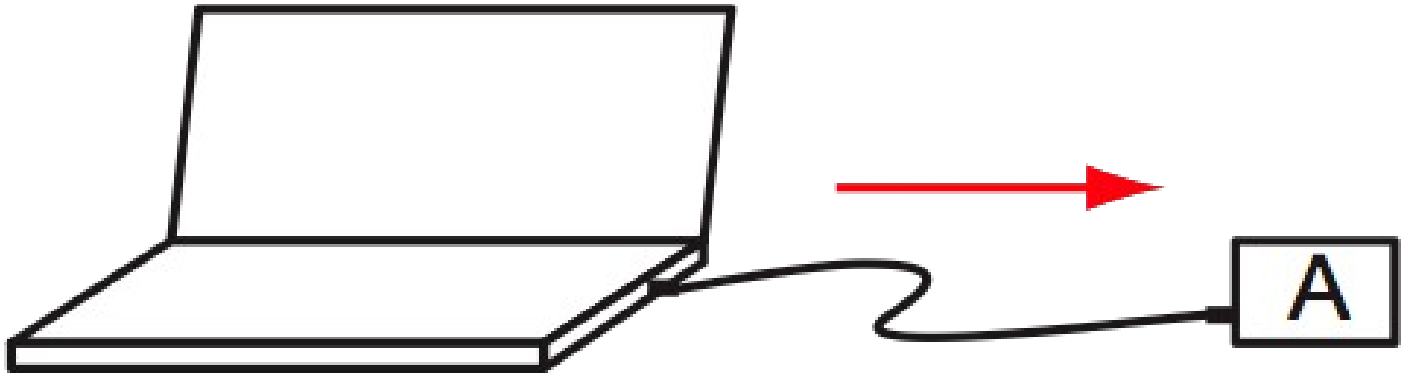


http://arduino.cc/es/main/software

Write sketch on PC

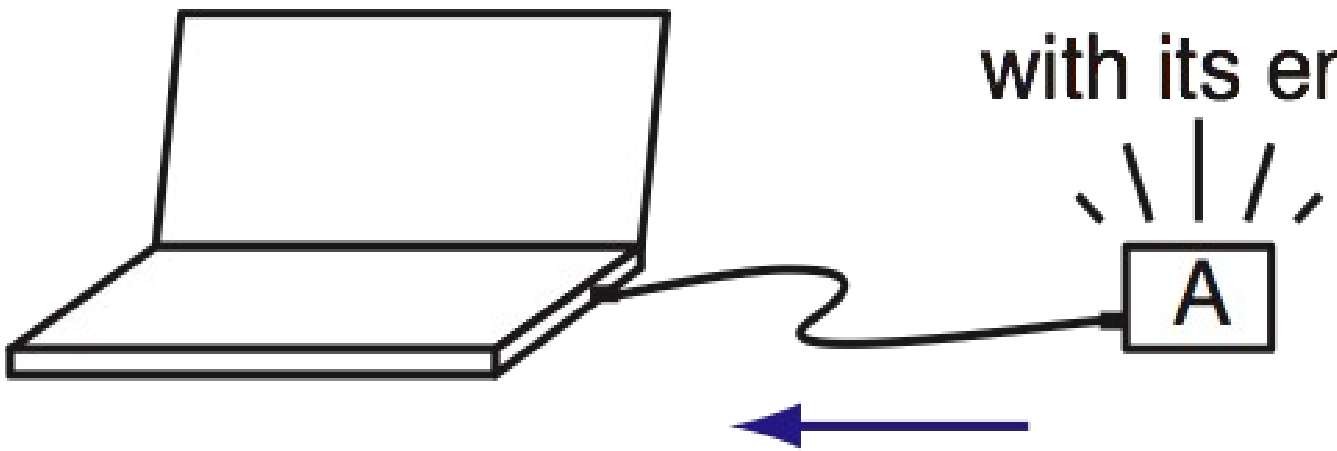


Download sketch to Arduino



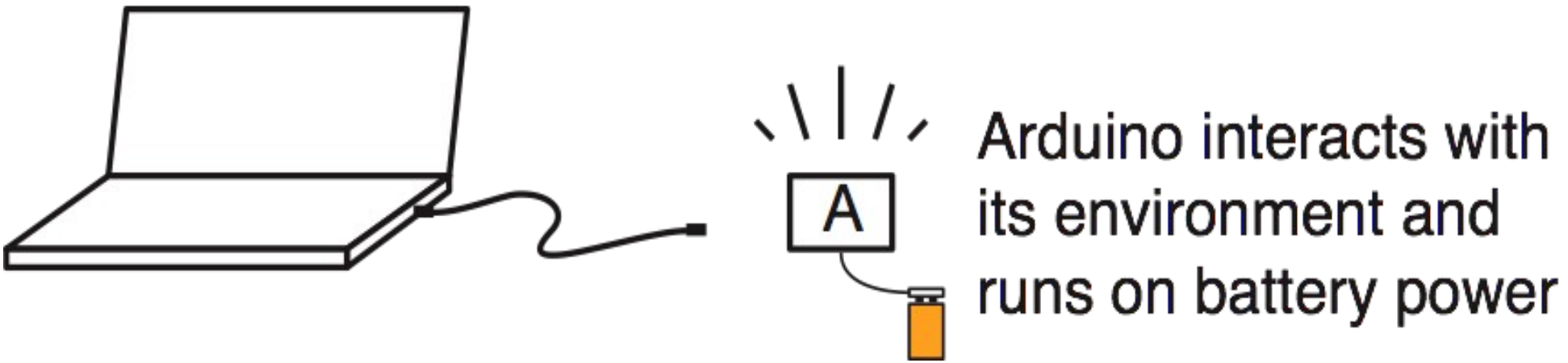
Run sketch on Arduino
and send data back to PC

Arduino interacts
with its environment



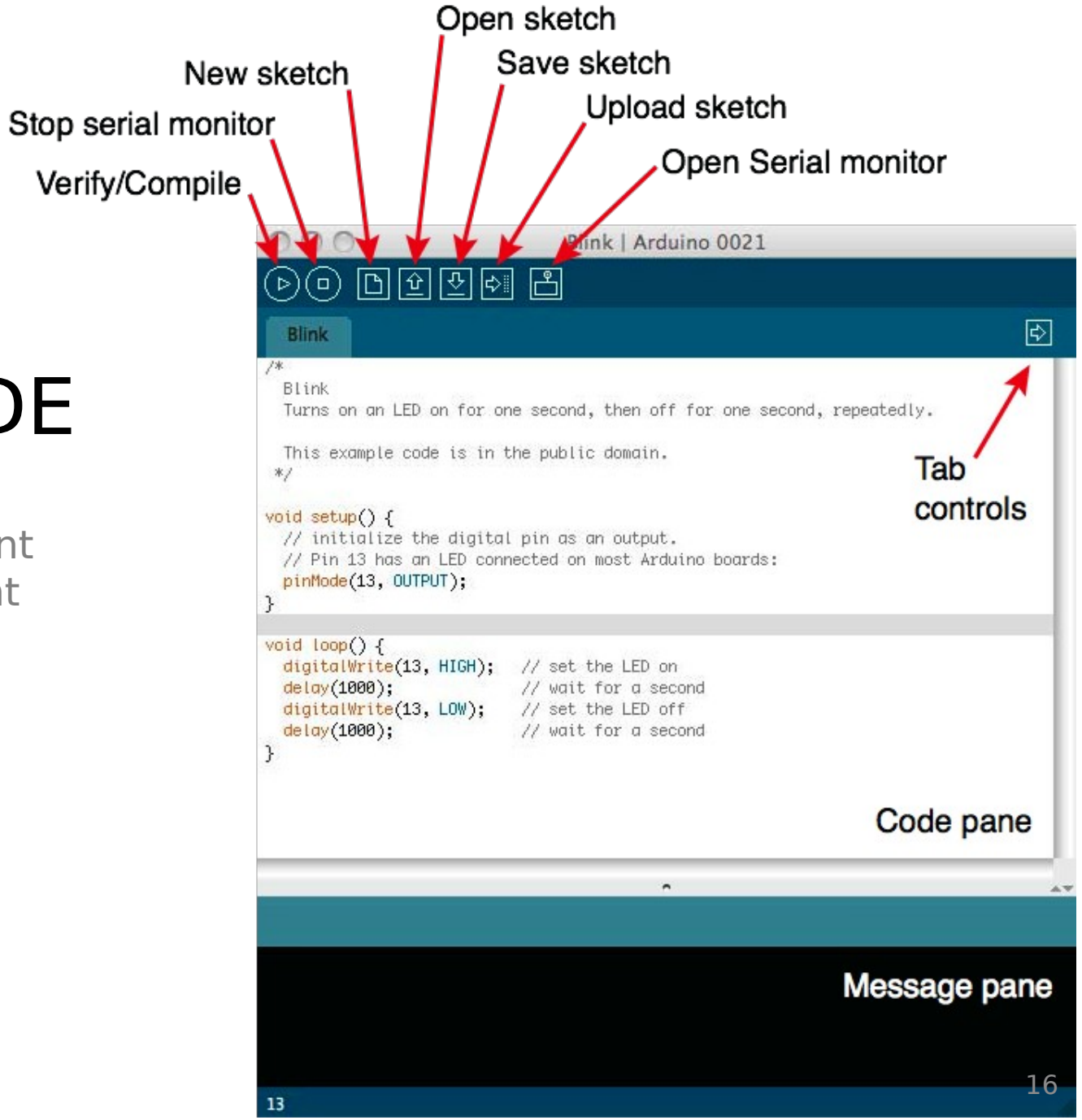
Serial communication
back to host

Run Arduino in stand alone mode



Arduino IDE

IDE = Integrated Development Environment



http://arduino.cc/es/main/software

Esqueleto básico

```
void setup()
{
//Se ejecuta al encender
}
void loop()
{
//Se ejecuta tras setup()
//Su ejecución se repite indefinidamente mientras
la
//placa esté alimentada
}
```

Programando el Arduino

// (single line comment)
It is often useful to write notes to yourself as you go along about what each line of code does. To do this type two back slashes and everything until the end of the line will be ignored by your program.

/* */ (multi line comment)
If you have a lot to say you can span several lines as a comment. Everything between these two symbols will be ignored in your program.

{ } (curly brackets)
Used to define when a block of code starts and ends (used in functions as well as loops)

; (semicolon)
Each line of code must be ended with a semicolon (a missing semicolon is often the reason for a programme refusing to compile)

Variables

Variables

A program is nothing more than instructions to move numbers around in an intelligent way. Variables are used to do the moving

int (integer)
The main workhorse, stores a number in 2 bytes (16 bits). Has no decimal places and will store a value between -32,768 and 32,768.

long (long)
Used when an integer is not large enough. Takes 4 bytes (32 bits) of RAM and has a range between -2,147,483,648 and 2,147,483,648.

boolean (boolean)
A simple True or False variable. Useful because it only uses one bit of RAM.

float (float)
Used for floating point math (decimals). Takes 4 bytes (32 bits) of RAM and has a range between -3.4028235E+38 and 3.4028235E+38.

char (character)
Stores one character using the ASCII code (ie 'A' = 65). Uses one byte (8 bits) of RAM. The Arduino handles strings as an array of char's

Operadores matemáticos

Maths Operators

Operators used for manipulating numbers. (they work like simple maths)

- **=** (assignment) makes something equal to something else (eg. x = 10 * 2 (x now equals 20))
- **%** (modulo) gives the remainder when one number is divided by another (ex. 12 % 10 (gives 2))
- **+** (addition)
- **-** (subtraction)
- ***** (multiplication)
- **/** (division)

Comparison Operators

Operators used for logical comparison

- **==** (equal to) (eg. 12 == 10 is FALSE or 12 == 12 is TRUE)
- **!=** (not equal to) (eg. 12 != 10 is TRUE or 12 != 12 is FALSE)
- **<** (less than) (eg. 12 < 10 is FALSE or 12 < 12 is FALSE or 12 < 14 is TRUE)
- **>** (greater than) (eg. 12 > 10 is TRUE or 12 > 12 is FALSE or 12 > 14 is FALSE)

Podemos crear funciones

```
int* mi_funcion(char * param1, char * param2)
{
//Código de nuestra función
}
```

Estructuras de control

Control Structure

Programs are reliant on controlling what runs next, here are the basic control elements (there are many more online)

```
if(condition){ }
else if( condition ){ }
else { }
```

This will execute the code between the curly brackets if the condition is true, and if not it will test the `else if` condition if that is also false the `else` code will execute.

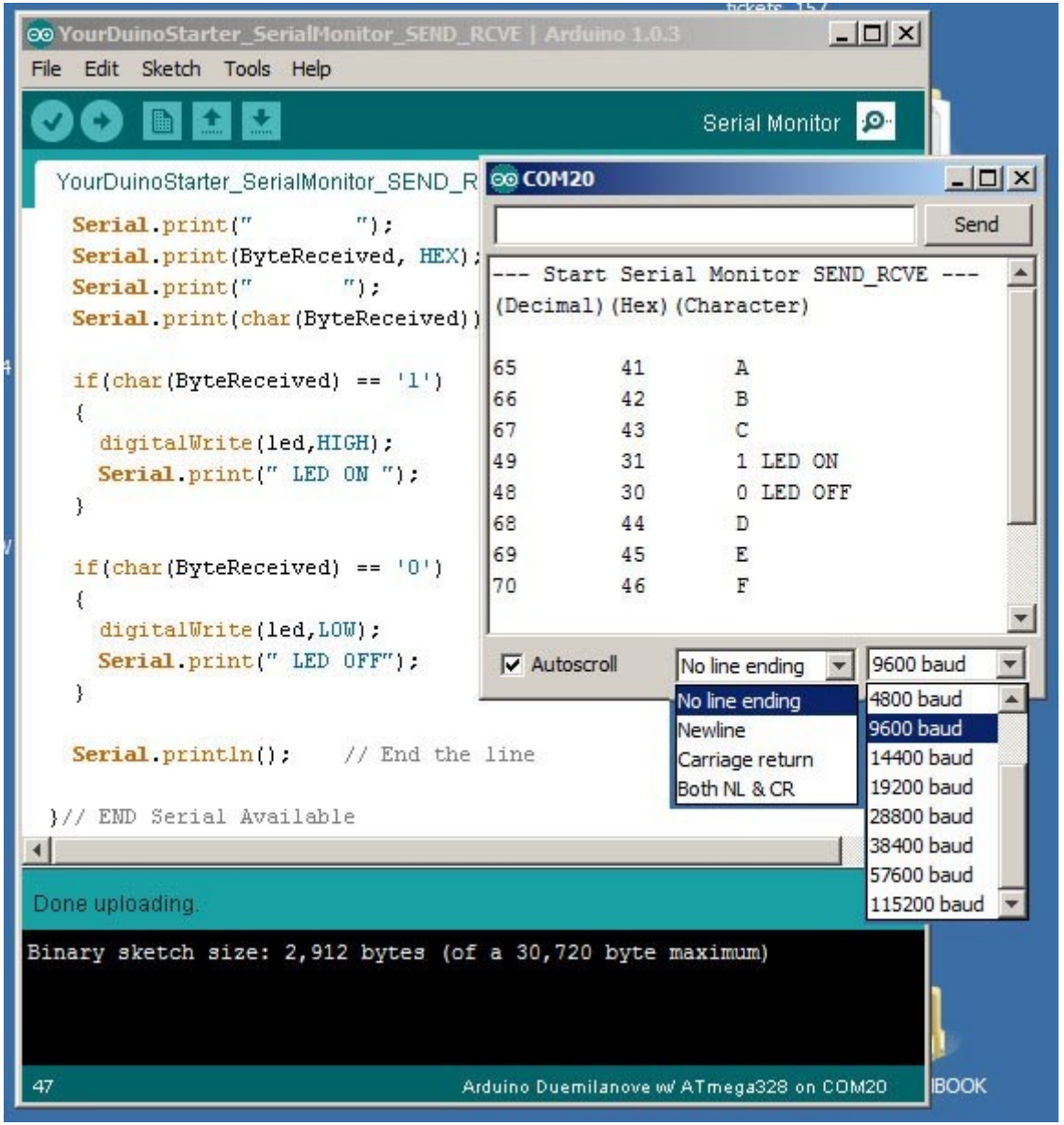
```
for(int i = 0; i <
#repeats; i++){   }
```

Used when you would like to repeat a chunk of code a number of times (can count up i++ or down i-- or use any variable)

Comunicación Serie

- Inicialización
  - Serial.begin(speed)
  - Habitualmente se hace en setup()
- Leer
  - Serial.read()
- Escribir
  - Serial.print(val)
  - Serial.println(val)
  - Donde val puede ser un número, una cadena o una variable

Comunicación Serie



Estructura

- setup() *(inicialización)*
- loop() *(bucle)*

Estructuras de control

- if *(comparador si-entonces)*
- if...else *(comparador si...sino)*
- for *(bucle con contador)*
- switch case *(comparador múltiple)*
- while *(bucle por comparación booleana)*
- do... while *(bucle por comparación booleana)*
- break *(salida de bloque de código)*
- continue *(continuación en bloque de código)*
- return *(devuelve valor a programa)*

Sintaxis

- ; (punto y coma)
- {} (llaves)
- // (comentarios en una línea)
- /* */ (comentarios en múltiples líneas)

Variables

Constantes

- HIGH | LOW
- INPUT | OUTPUT
- true | false
- Constantes Numéricas

Tipos de Datos

- boolean *(booleano)*
- char *(carácter)*
- byte
- int *(entero)*
- unsigned int *(entero sin signo)*
- long *(entero 32b)*
- unsigned long *(entero 32b sin signo)*
- float *(en coma flotante)*
- double *(en coma flotante de 32b)*
- string *(cadena de caracteres)*
- array *(cadena)*
- void *(vacío)*

Funciones

E/S Digitales

- pinMode()
- digitalWrite()
- digitalRead()

E/S Analógicas

- analogRead()
- analogWrite() - *PWM (modulación por ancho de pulso)*

E/S Avanzadas

- tone()
- noTone()
- shiftOut()
- pulseIn()

Tiempo

- millis()
- micros()
- delay()
- delayMicroseconds()

http://arduino.cc/es/Reference/HomePage

Entradas y salidas

Digital

`pinMode(pin, mode);`

Used to set a pins mode, `pin` is the pin number you would like to address (0-19 (analog 0-5 are 14-19). the mode can either be `INPUT` or `OUTPUT`.

`digitalWrite(pin, value);`

Once a pin is set as an `OUTPUT`, it can be set either `HIGH` (pulled to +5 volts) or `LOW` (pulled to ground).

`int digitalRead(pin);`

Once a pin is set as an INPUT you can use this to return whether it is HIGH (pulled to +5 volts) or LOW (pulled to ground).

Analog

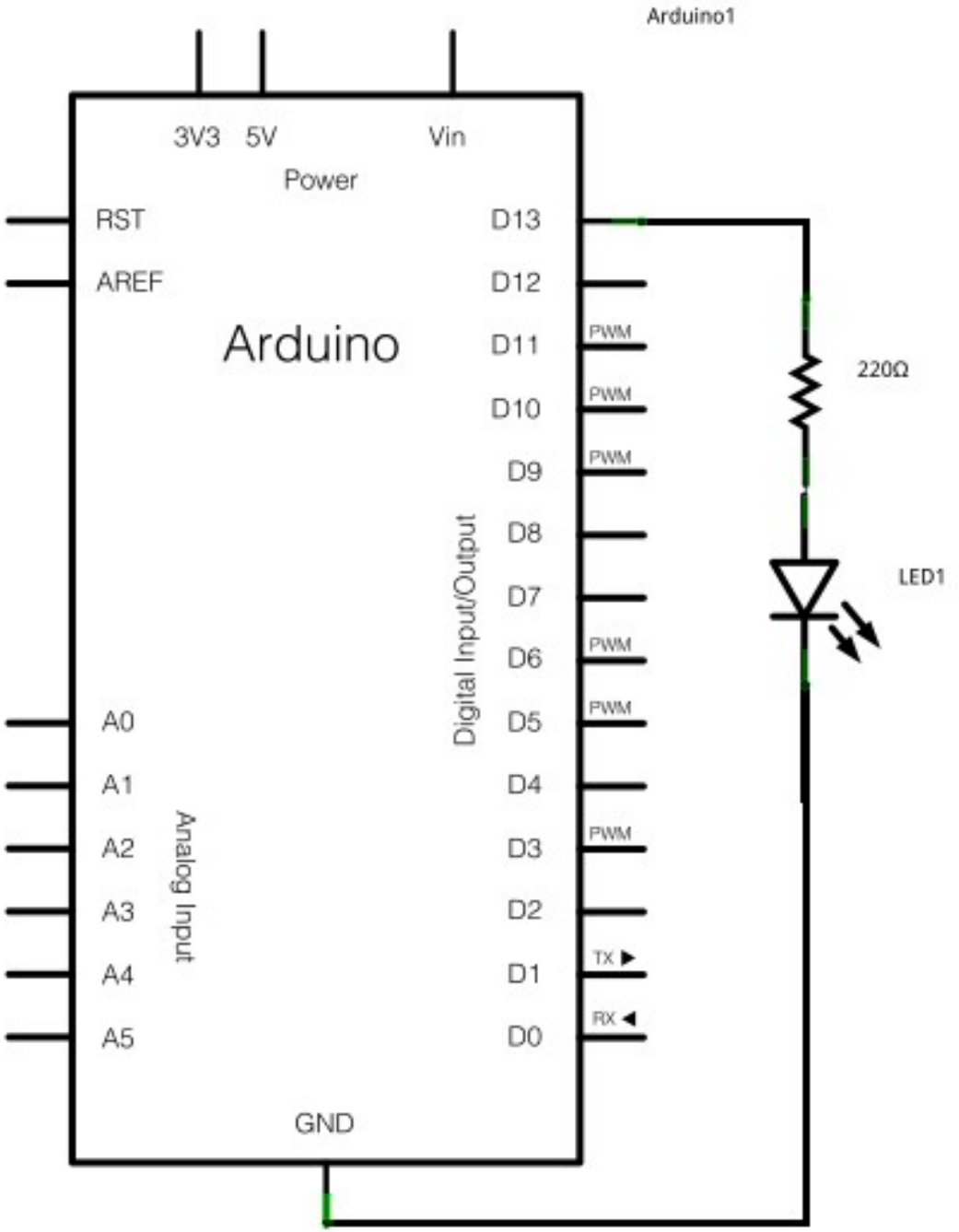
The Arduino is a digital machine but it has the ability to operate in the analog realm (through tricks). Here's how to deal with things that aren't digital.

`int analogWrite(pin, value);`

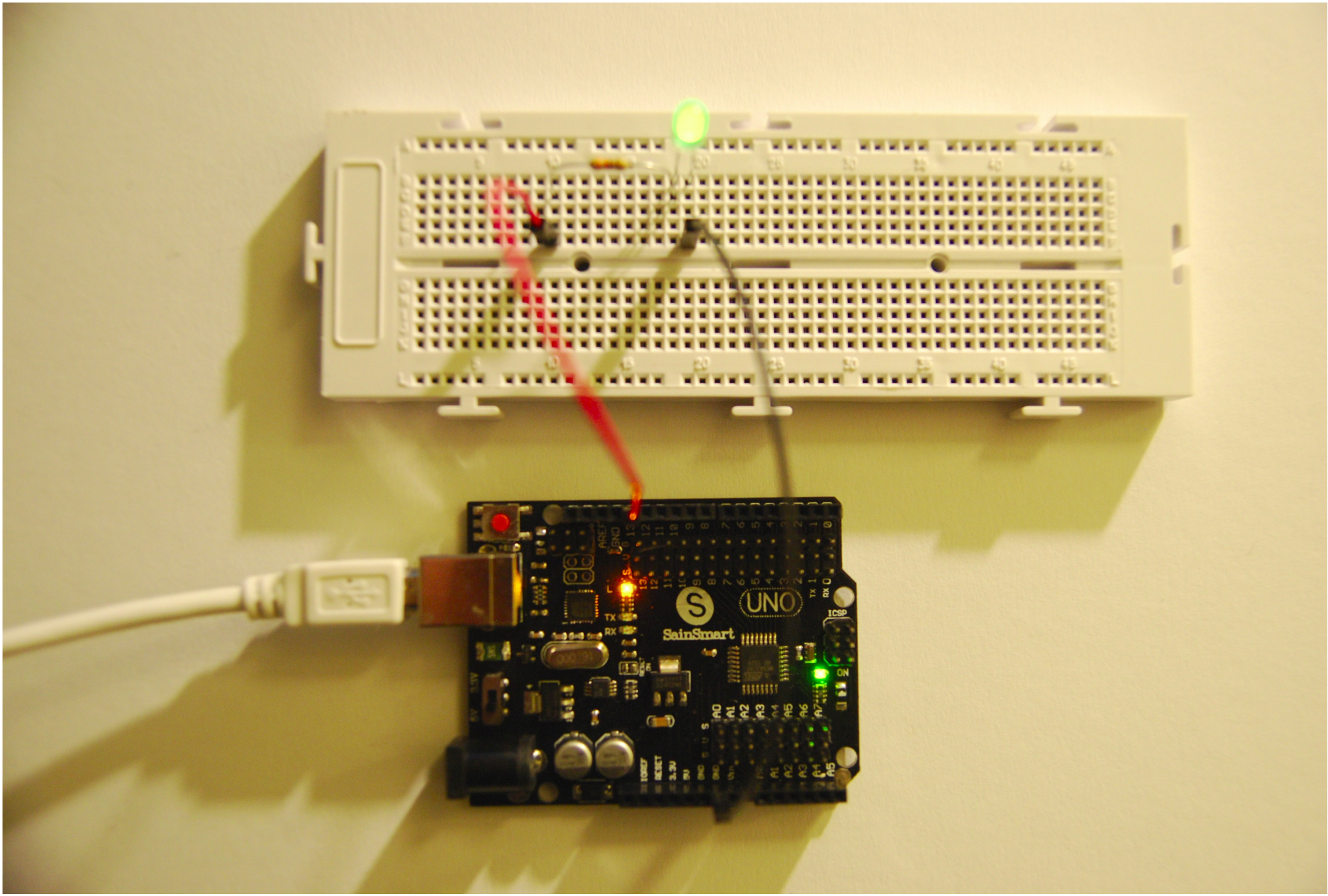
Some of the Arduino's pins support pulse width modulation (3, 5, 6, 9, 10, 11). This turns the pin on and off very quickly making it act like an analog output. The value is any number between 0 (0% duty cycle ~0v) and 255 (100% duty cycle ~5 volts).

`int analogRead(pin);`

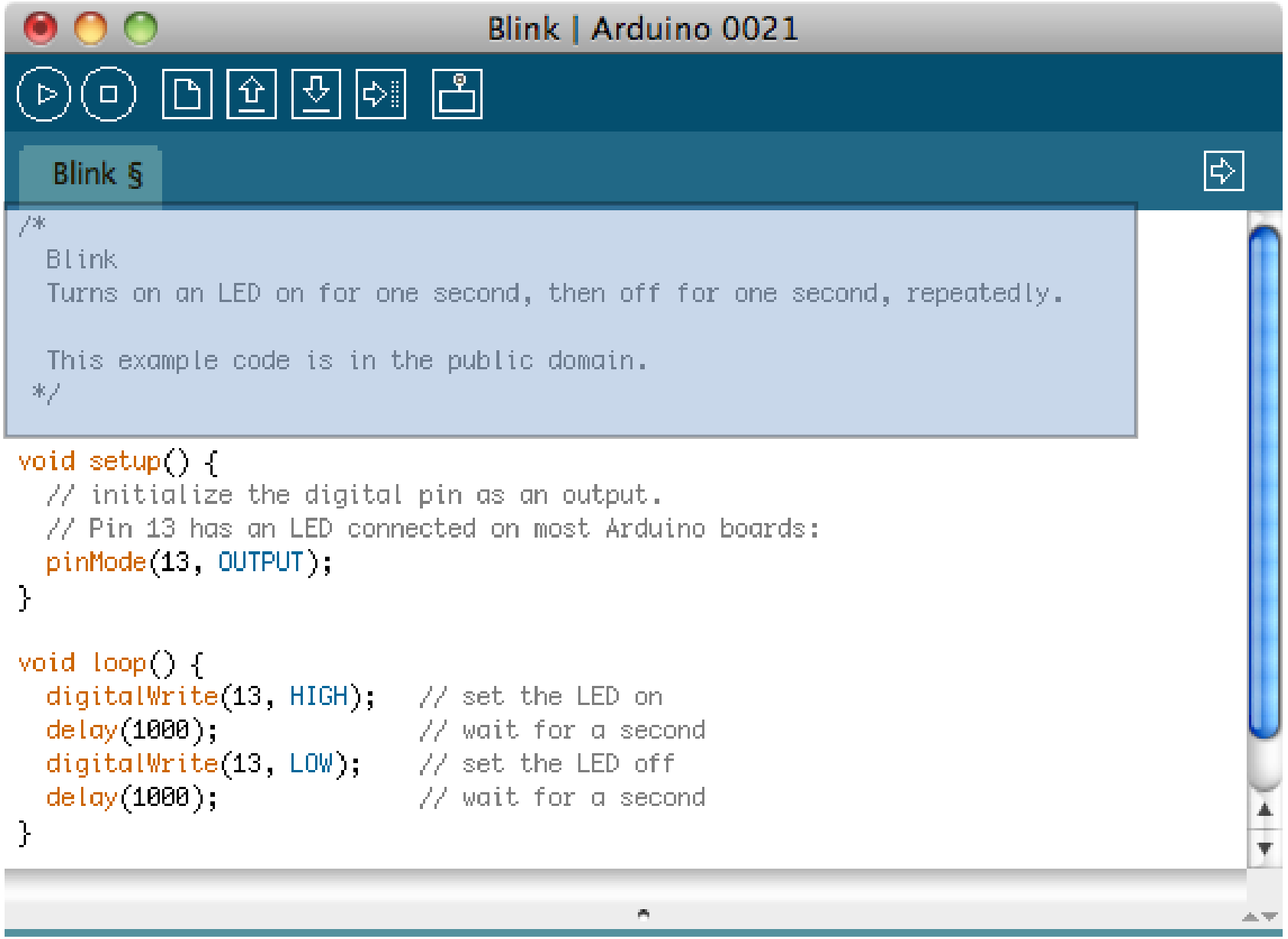
When the analog input pins are set to input you can read their voltage. A value between 0 (for 0 volts) and 1024 (for 5 volts) will be returned



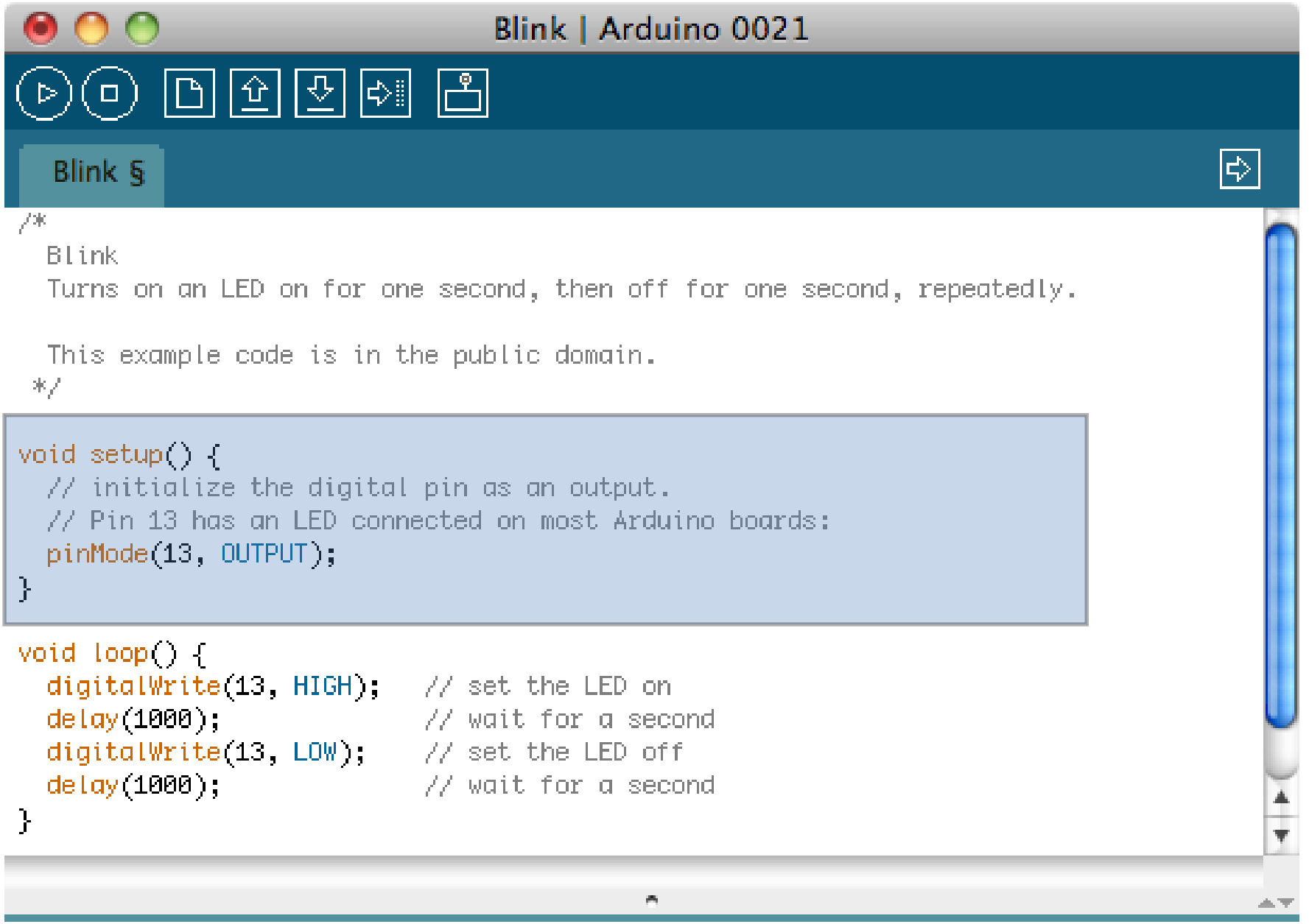
http://arduino.cc/en/Tutorial/Blink


UNO
SainSmart
ICSP
ON

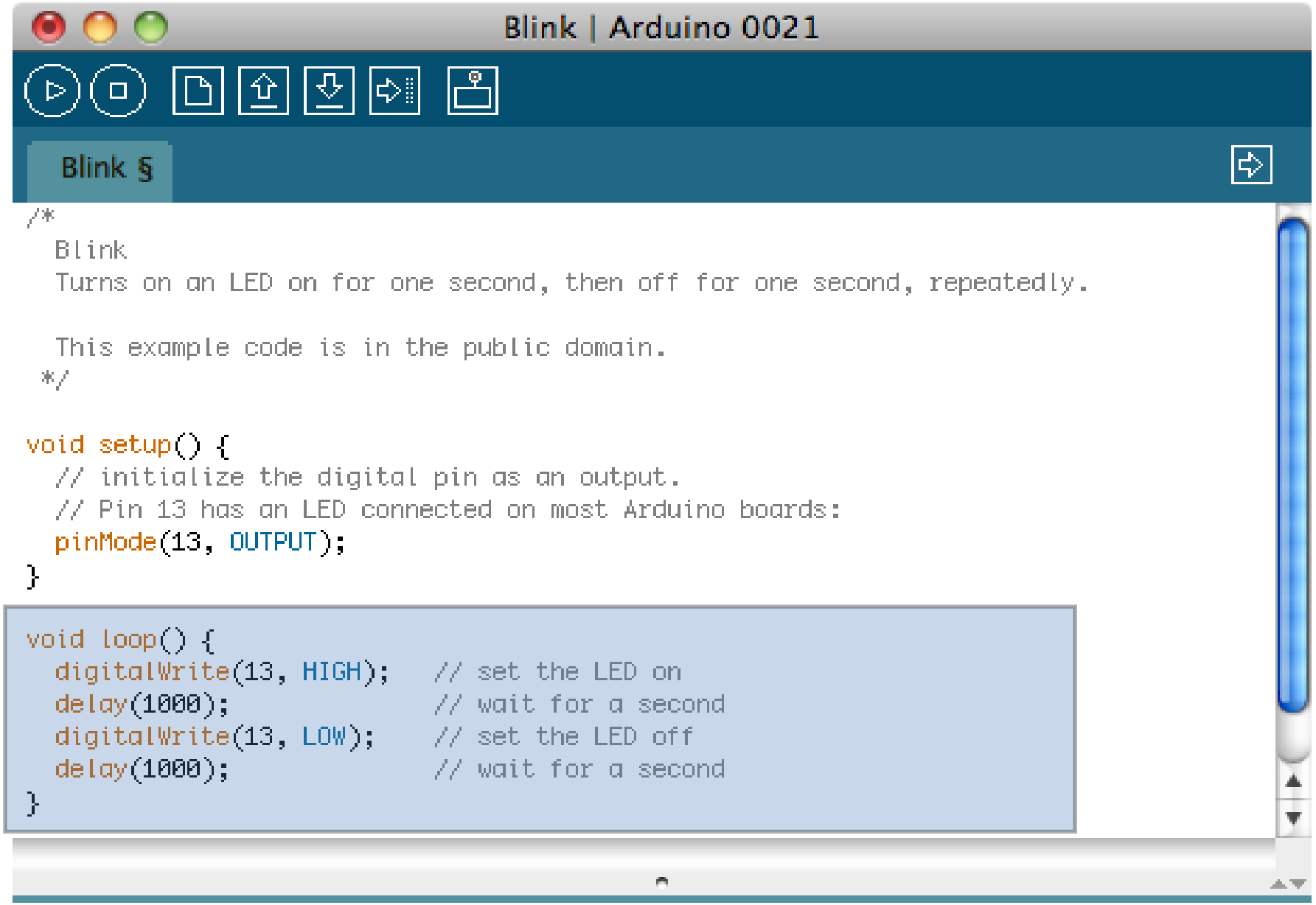
cabecera


Blink | Arduino 0021
Blink §
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}

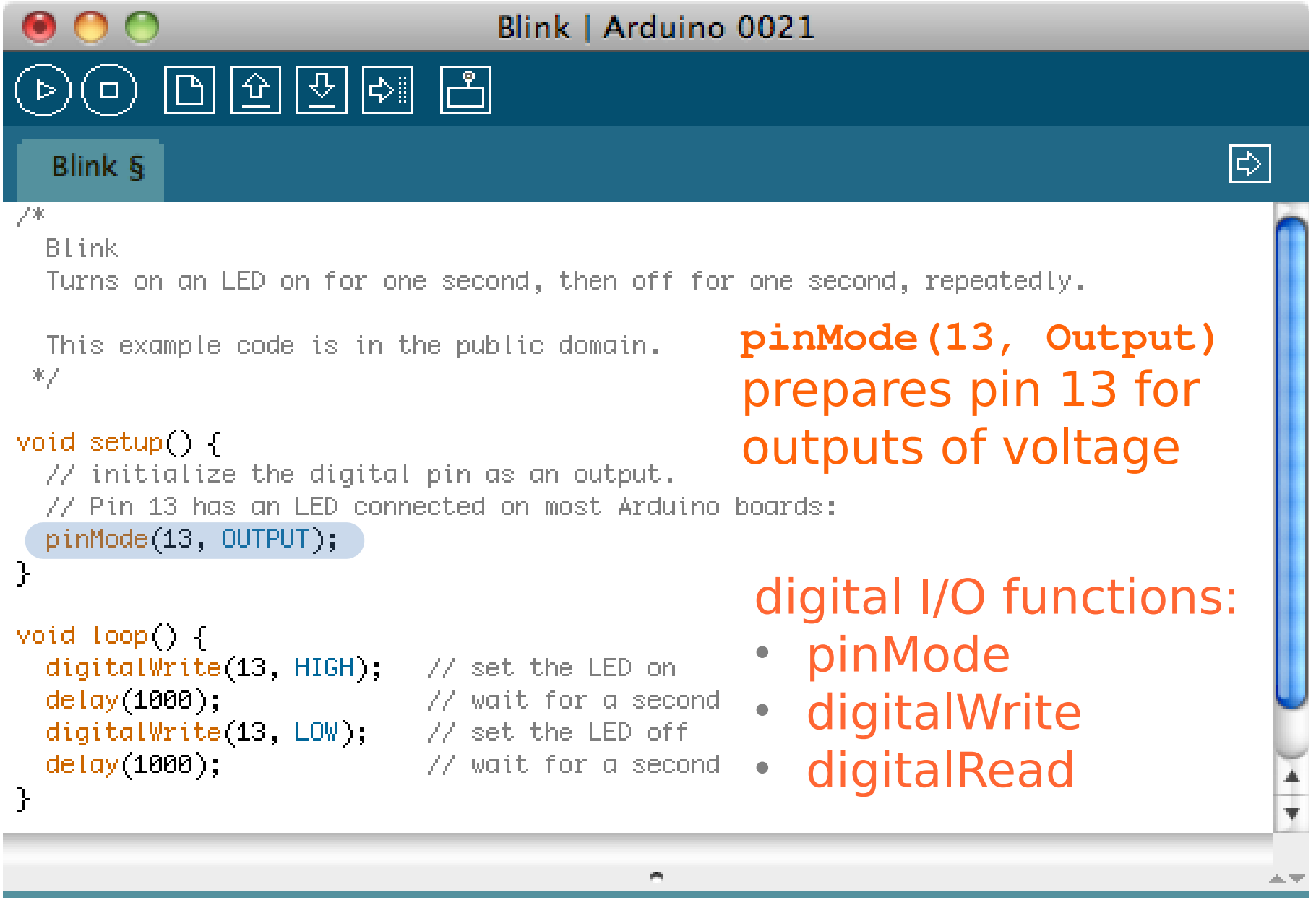
code structure: setup function

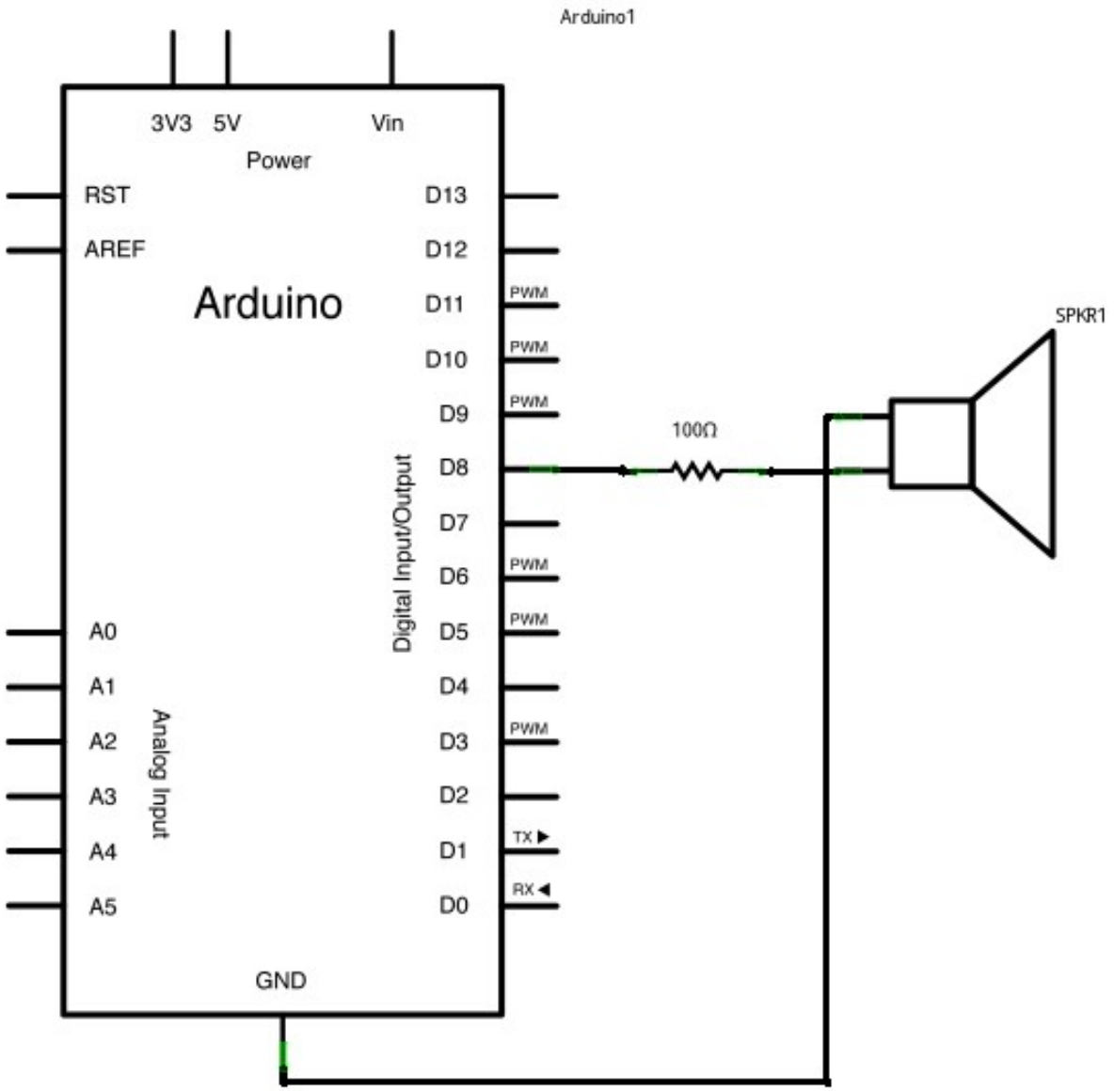


code structure: loop function

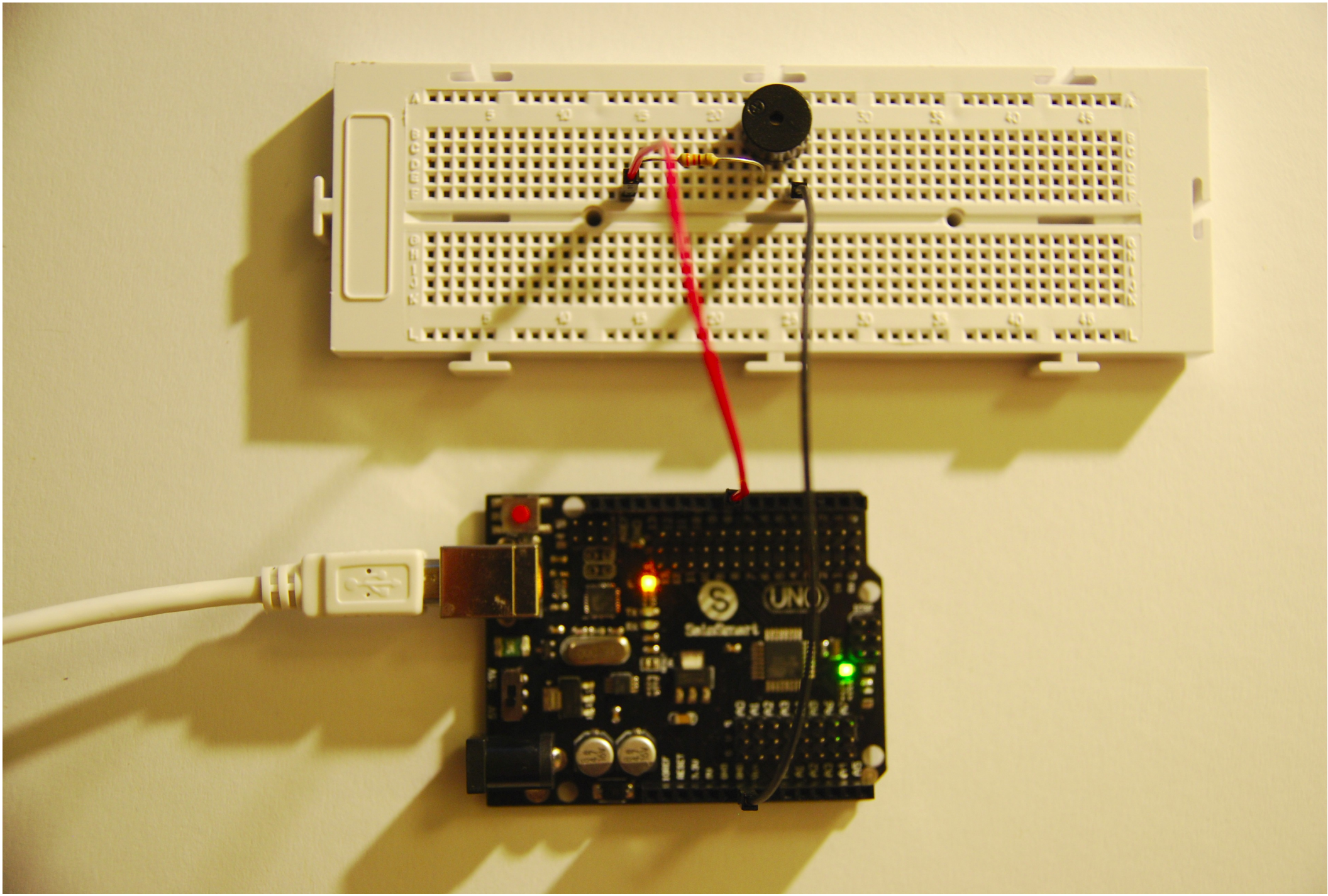


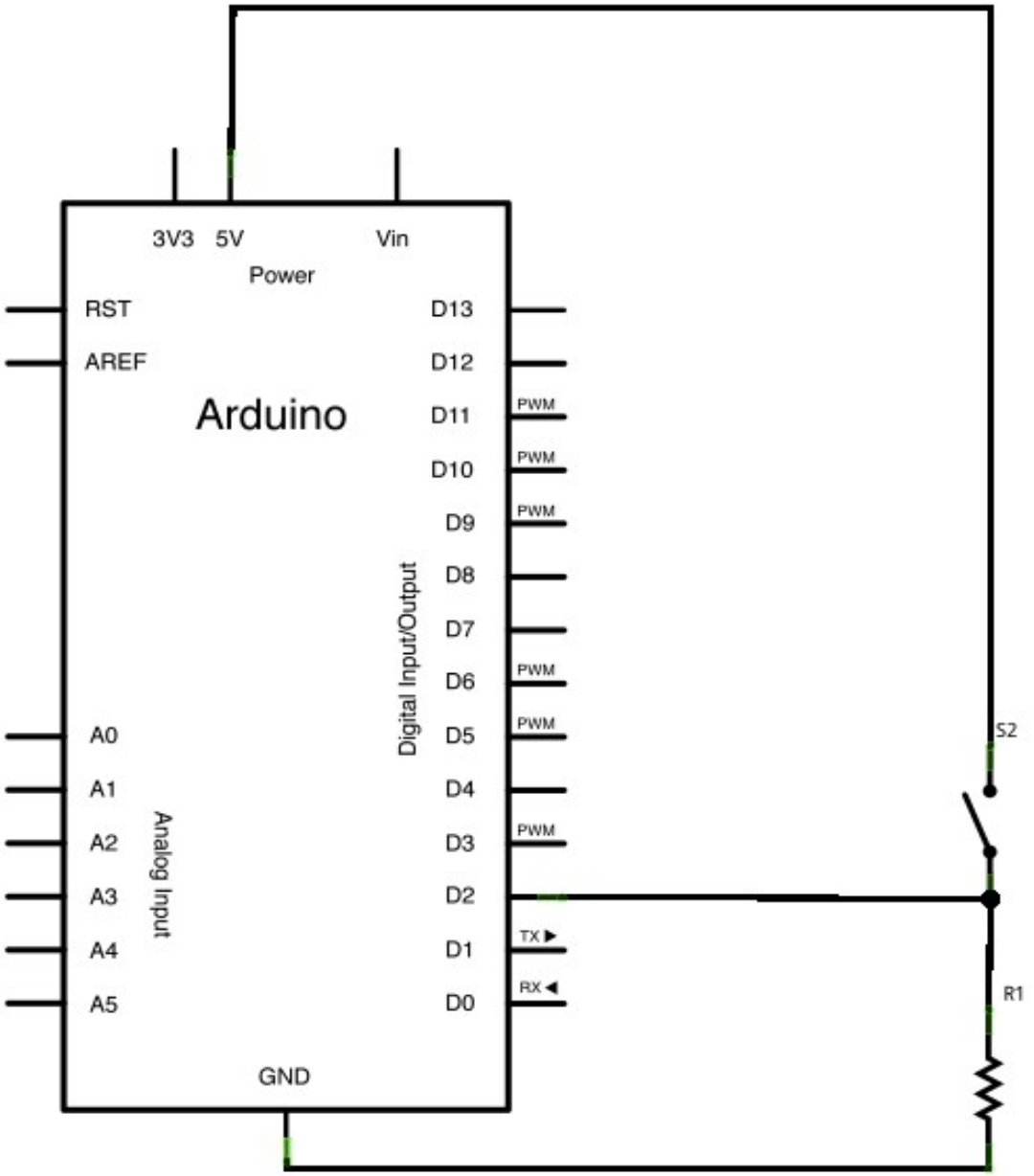
código


Blink | Arduino 0021
Blink §
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected on most Arduino boards:
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED on
delay(1000); // wait for a second
digitalWrite(13, LOW); // set the LED off
delay(1000); // wait for a second
}
pinMode(13, Output)
prepares pin 13 for outputs of voltage
digital I/O functions:
• pinMode
• digitalWrite
• digitalRead



http://arduino.cc/en/Tutorial/Tone

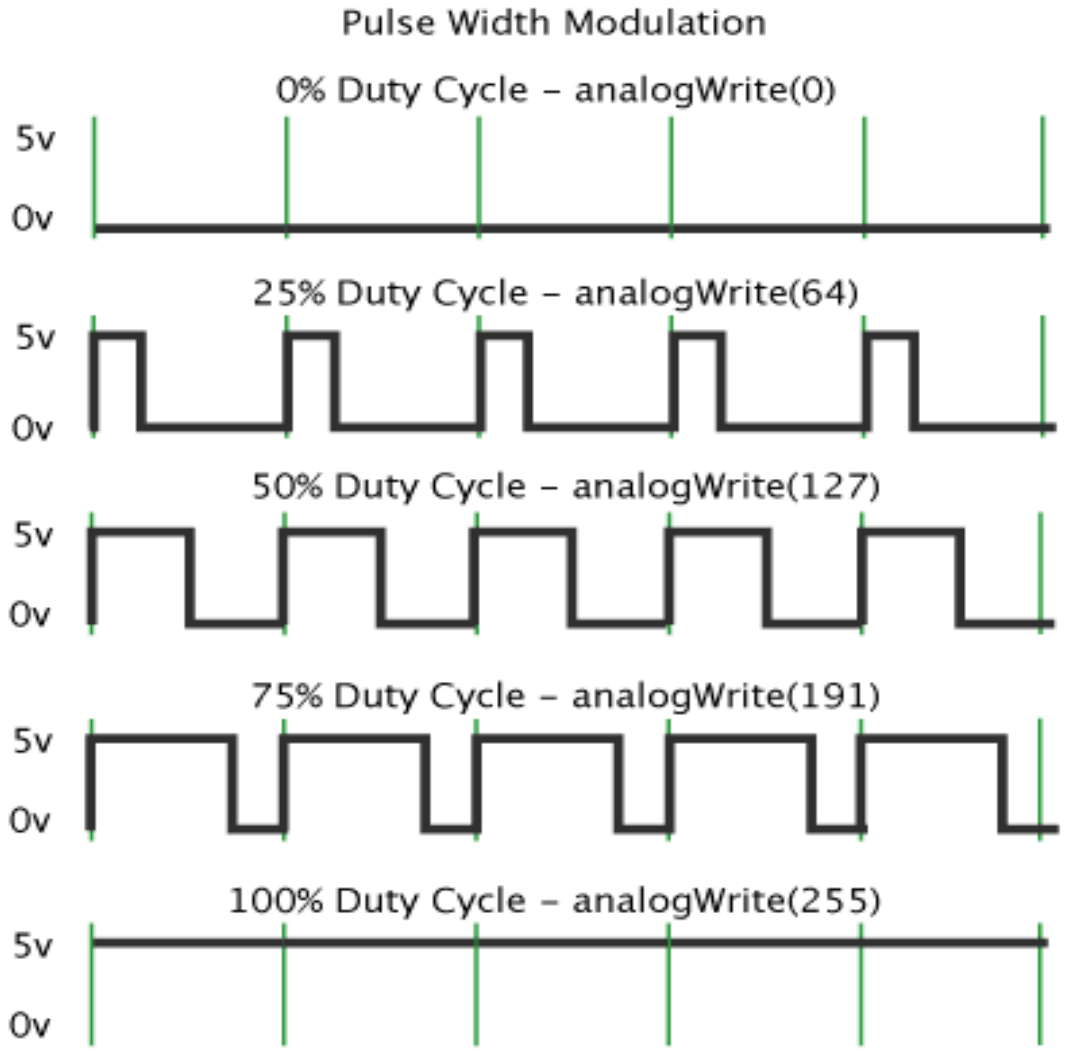




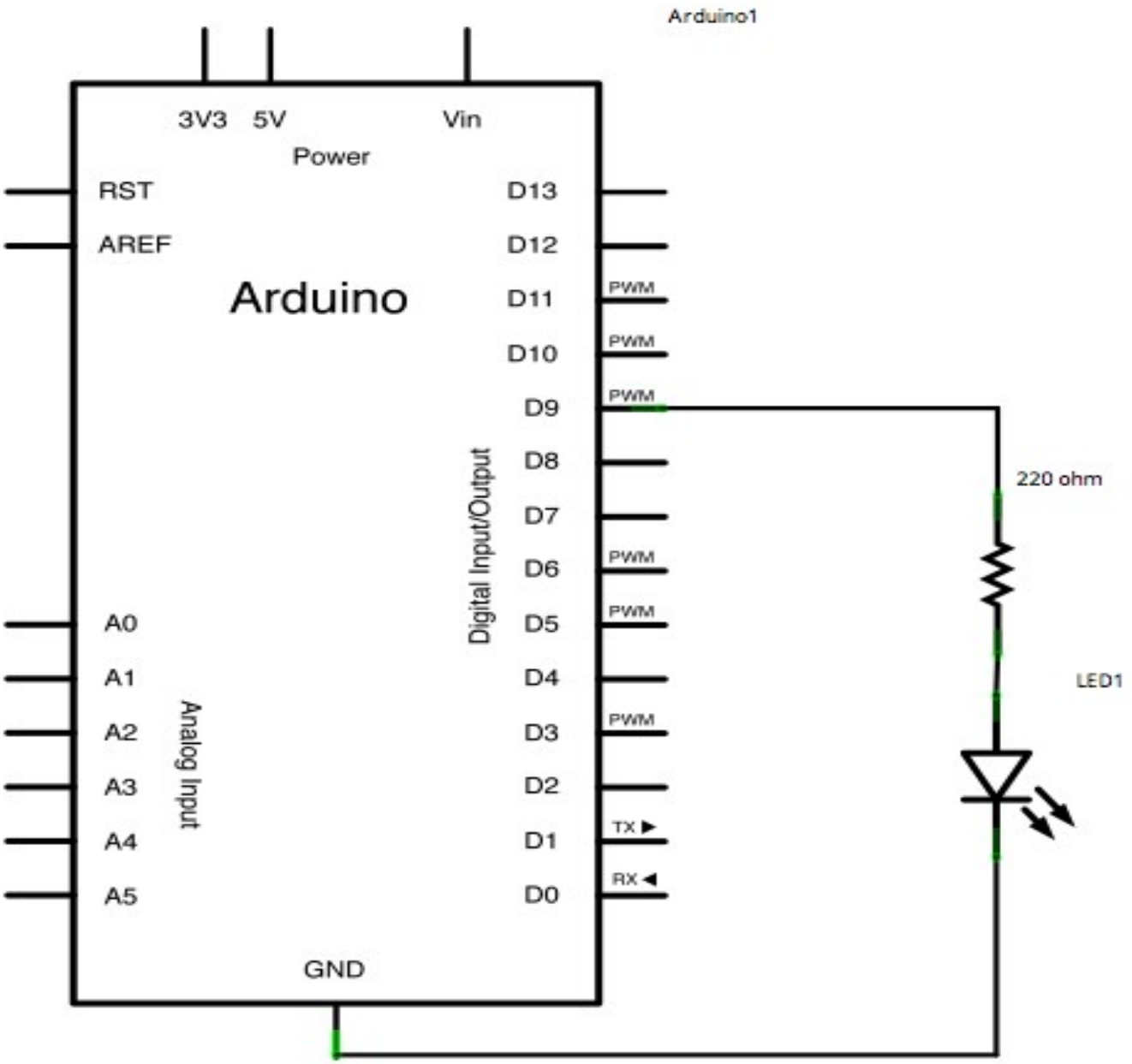
http://arduino.cc/en/Tutorial/Button
http://arduino.cc/en/Tutorial/ButtonStateChange

UNO
SainSmart
ICSP
ON
TX
RX
AREF
GND
RESET
3.3V
5V
IOREF

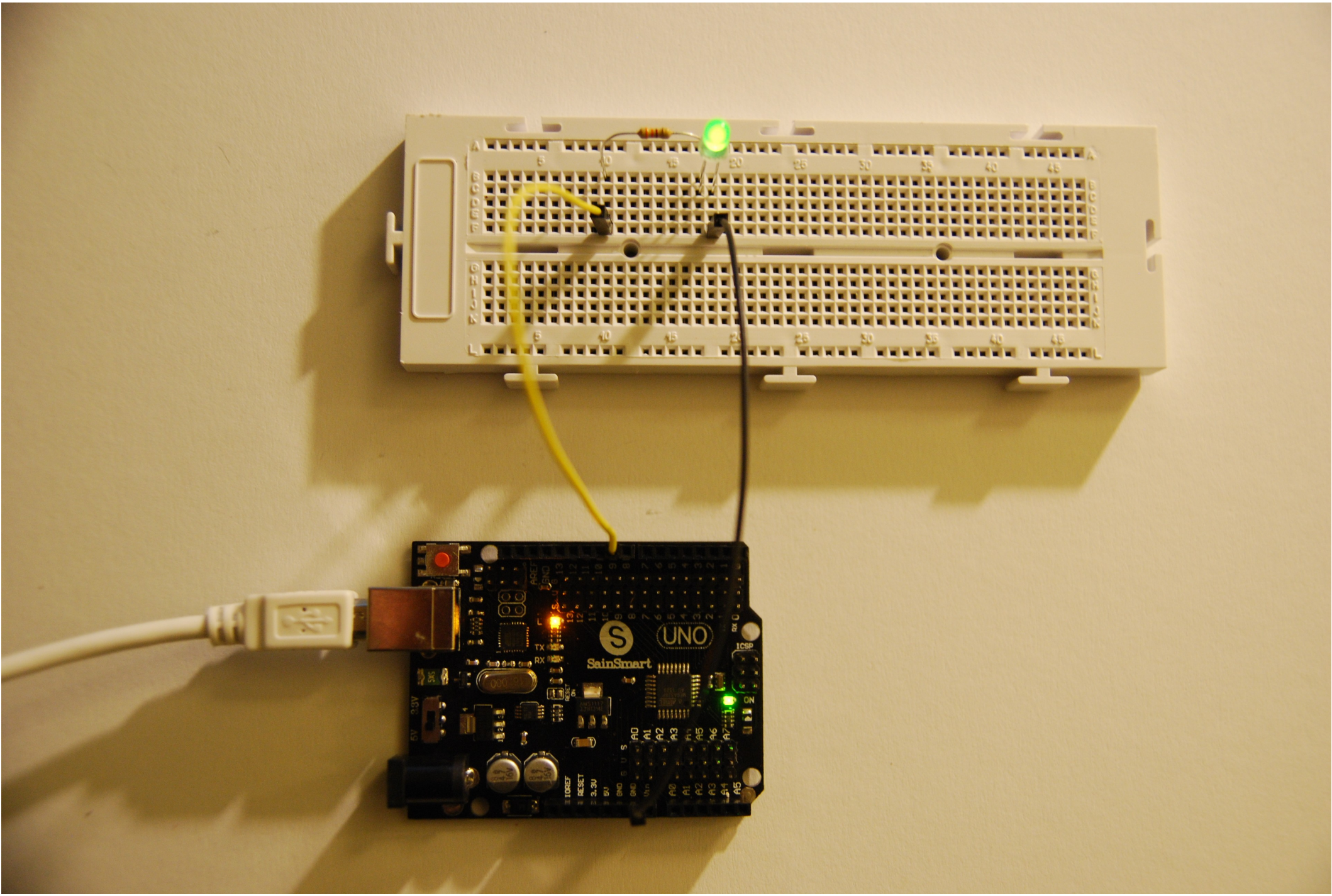
PWM

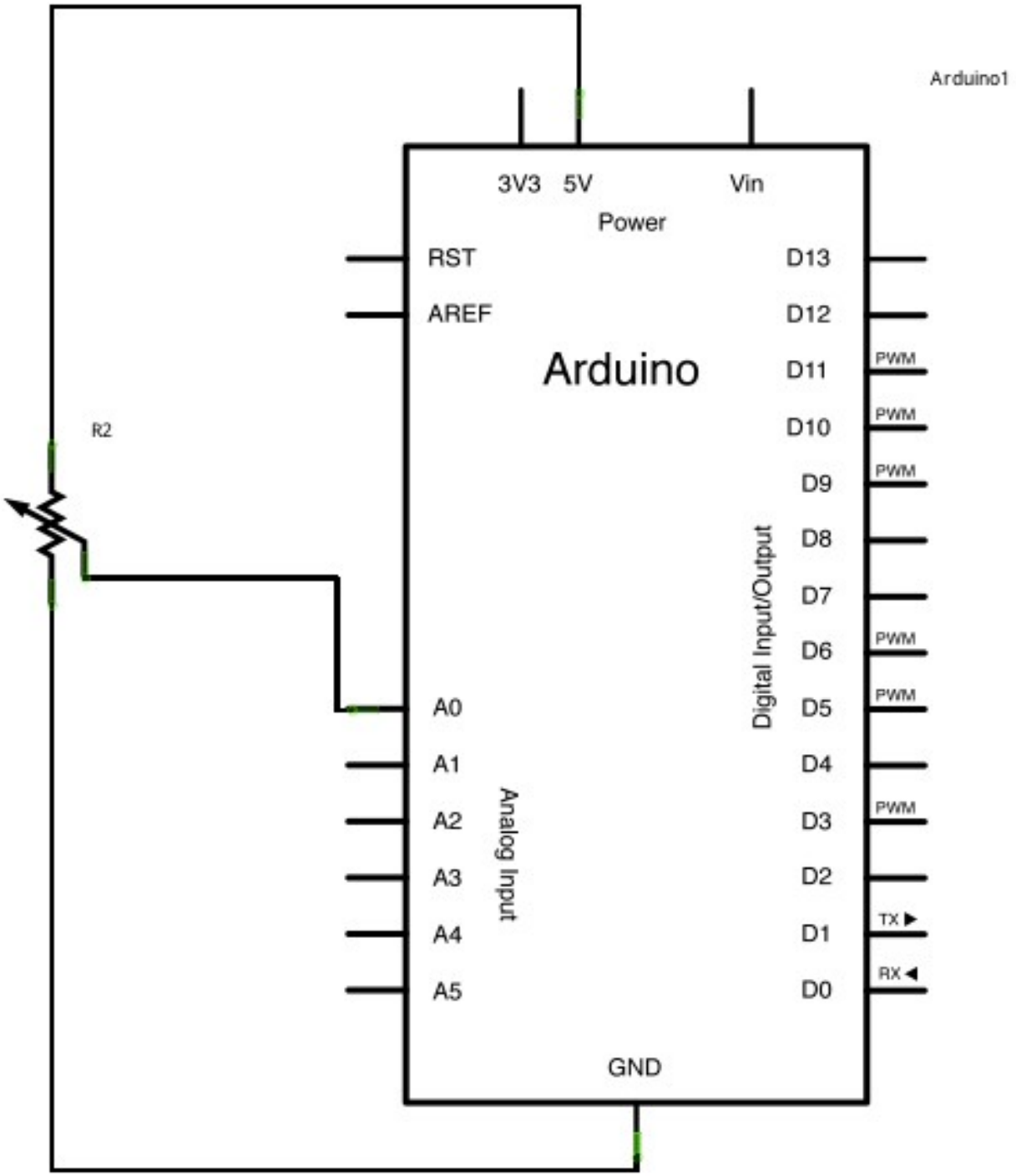


http://es.wikipedia.org/wiki/ModulaciC3%B3n_por_ancho_de_pulsos

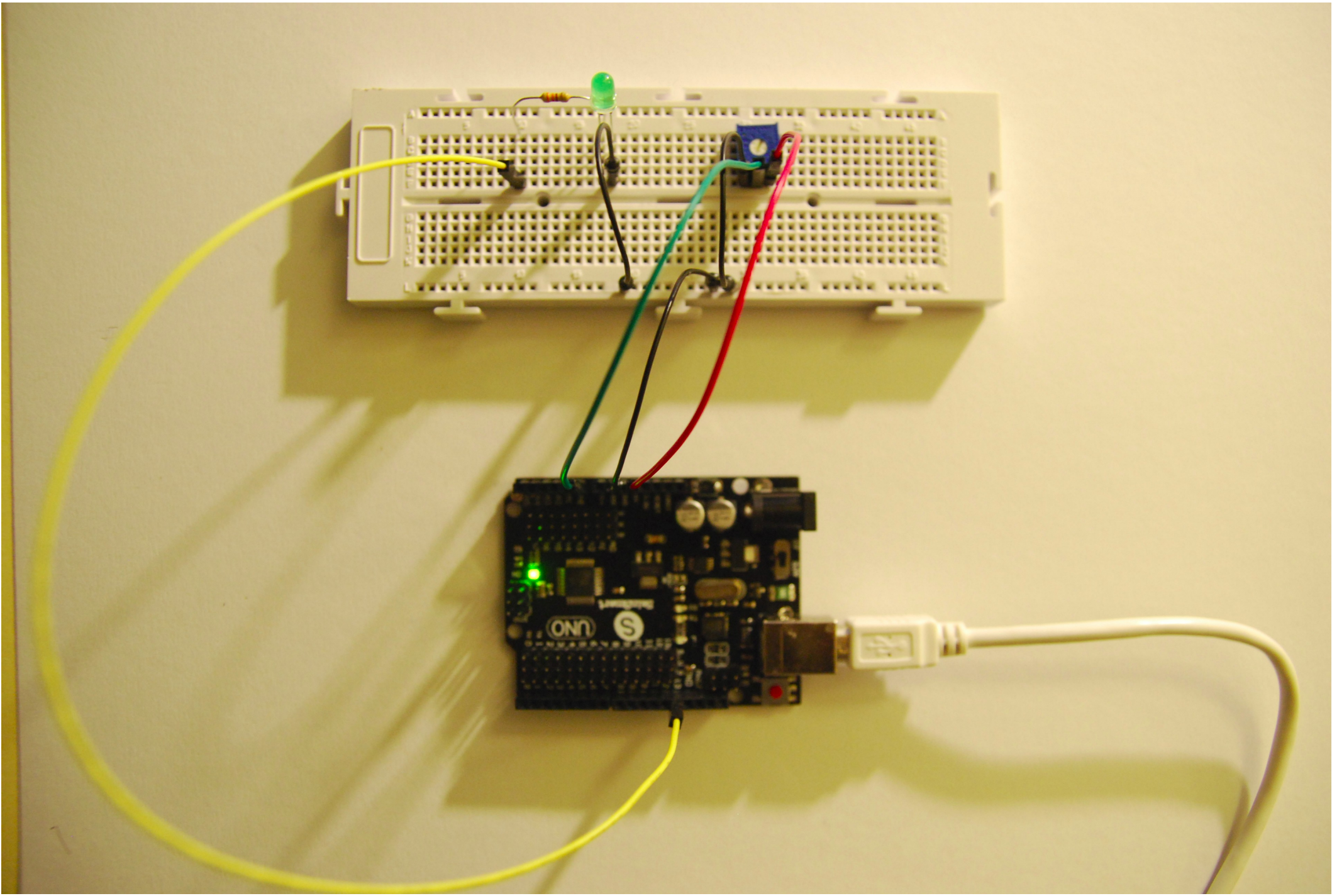


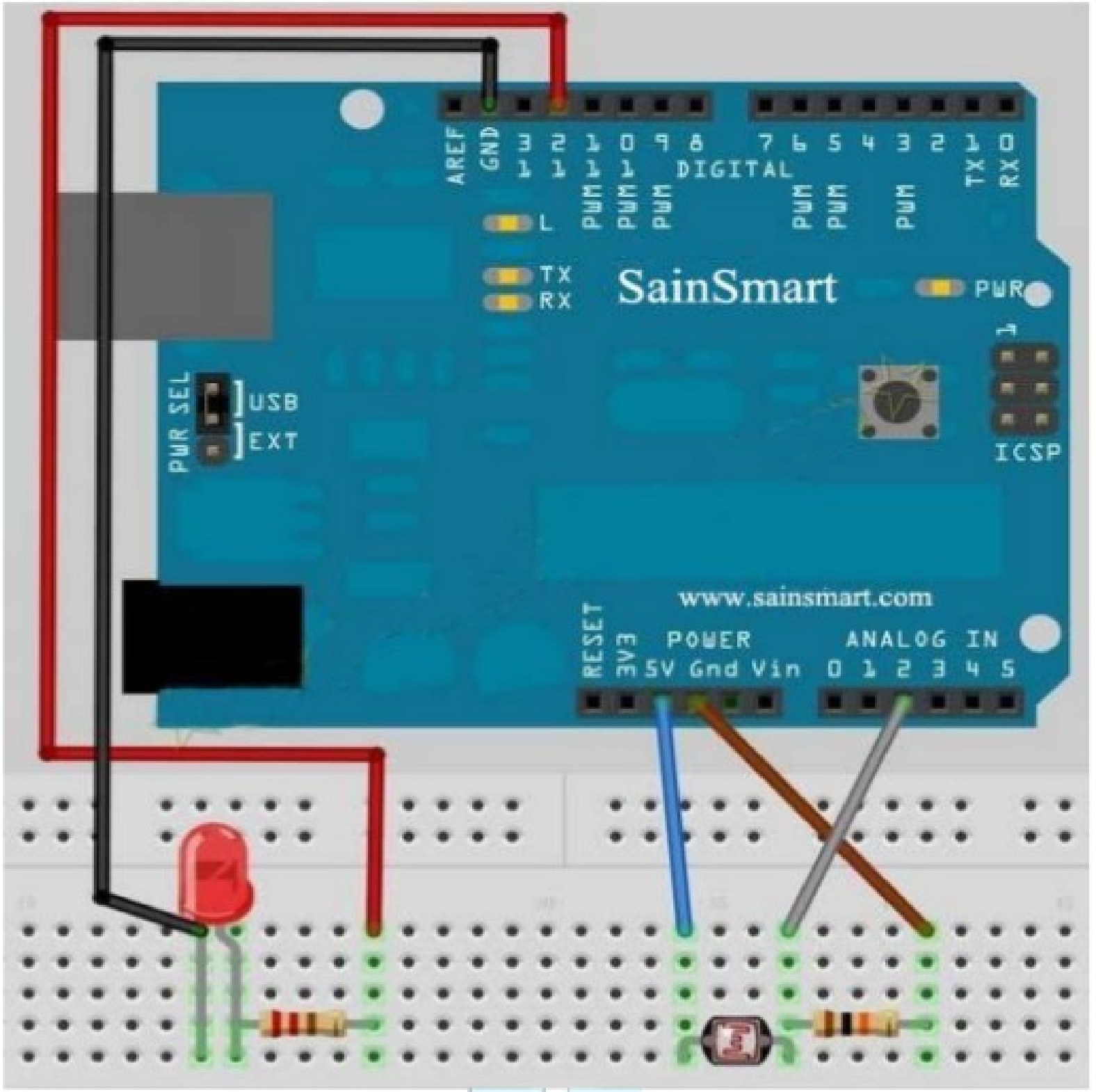
http://arduino.cc/en/Tutorial/Fading

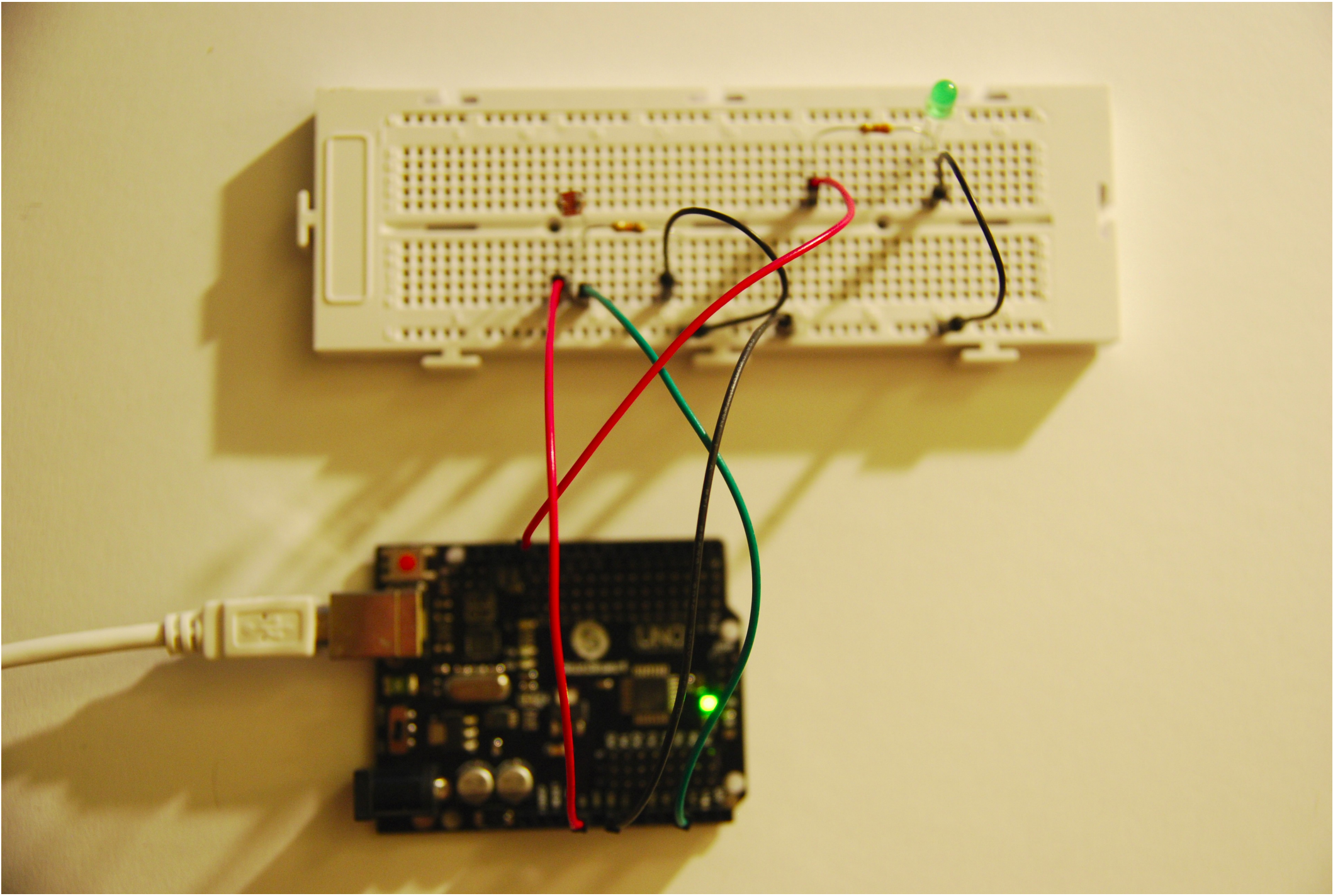

SainSmart
UNO

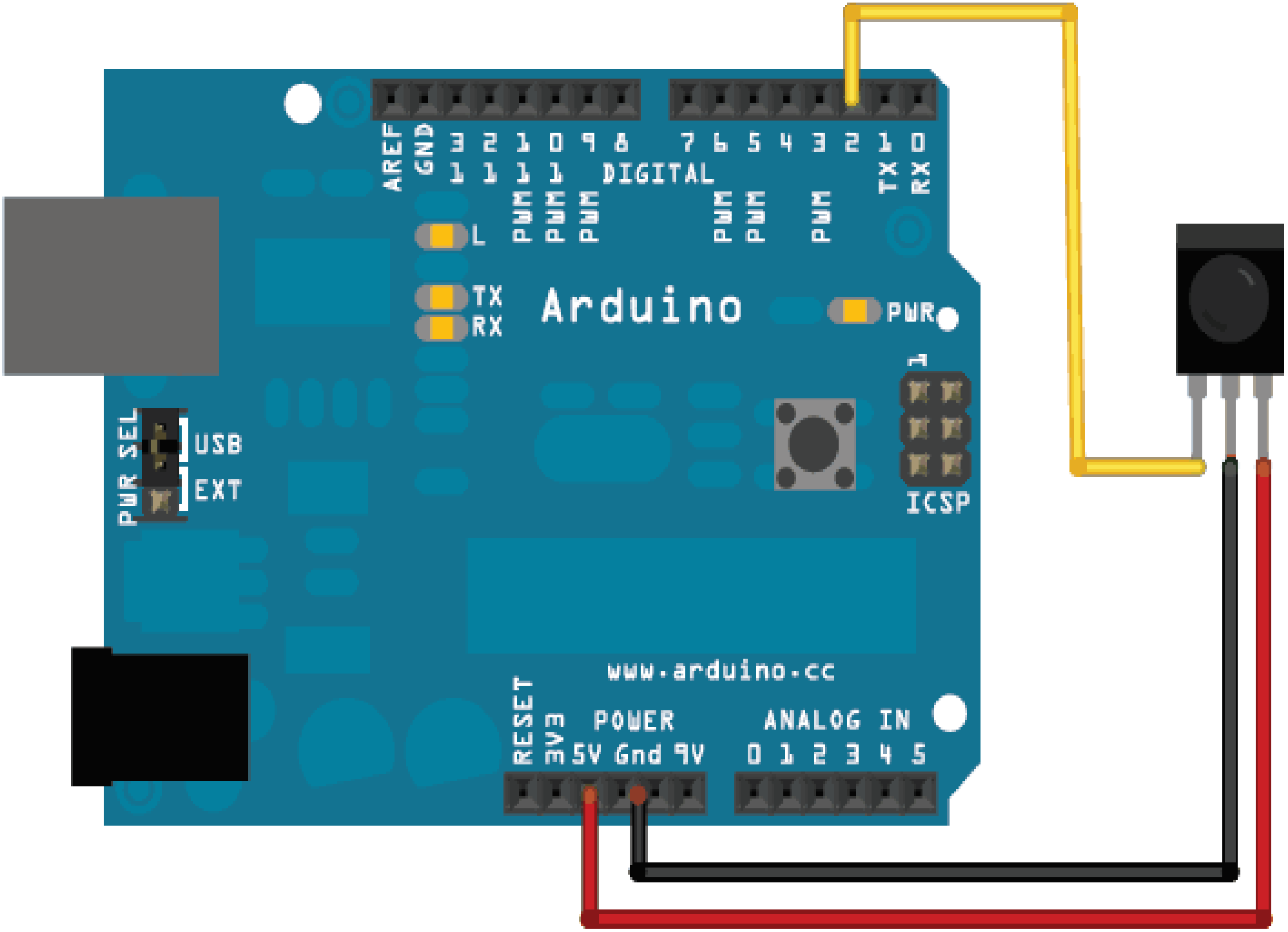


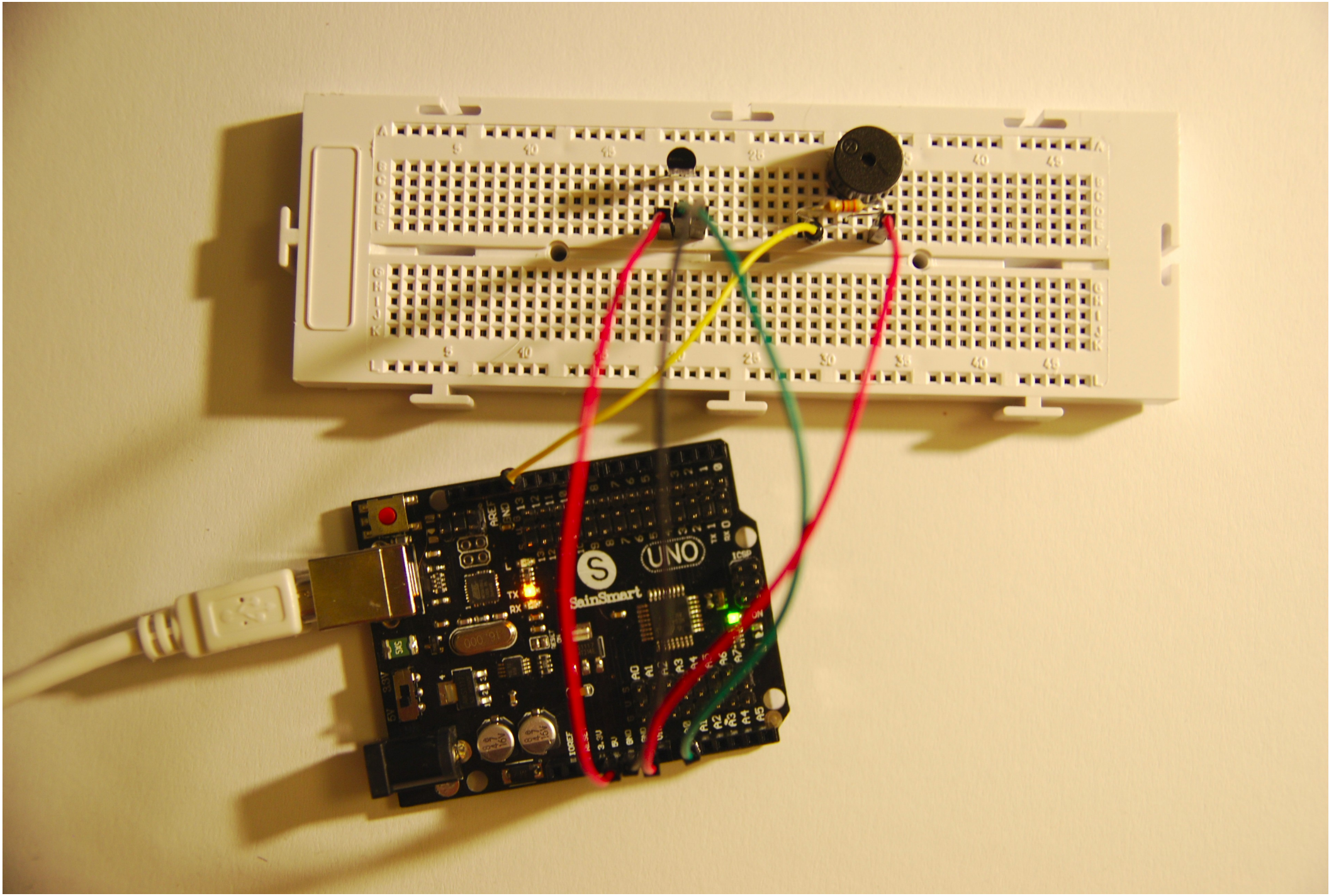
http://arduino.cc/en/Tutorial/AnalogInOutSerial

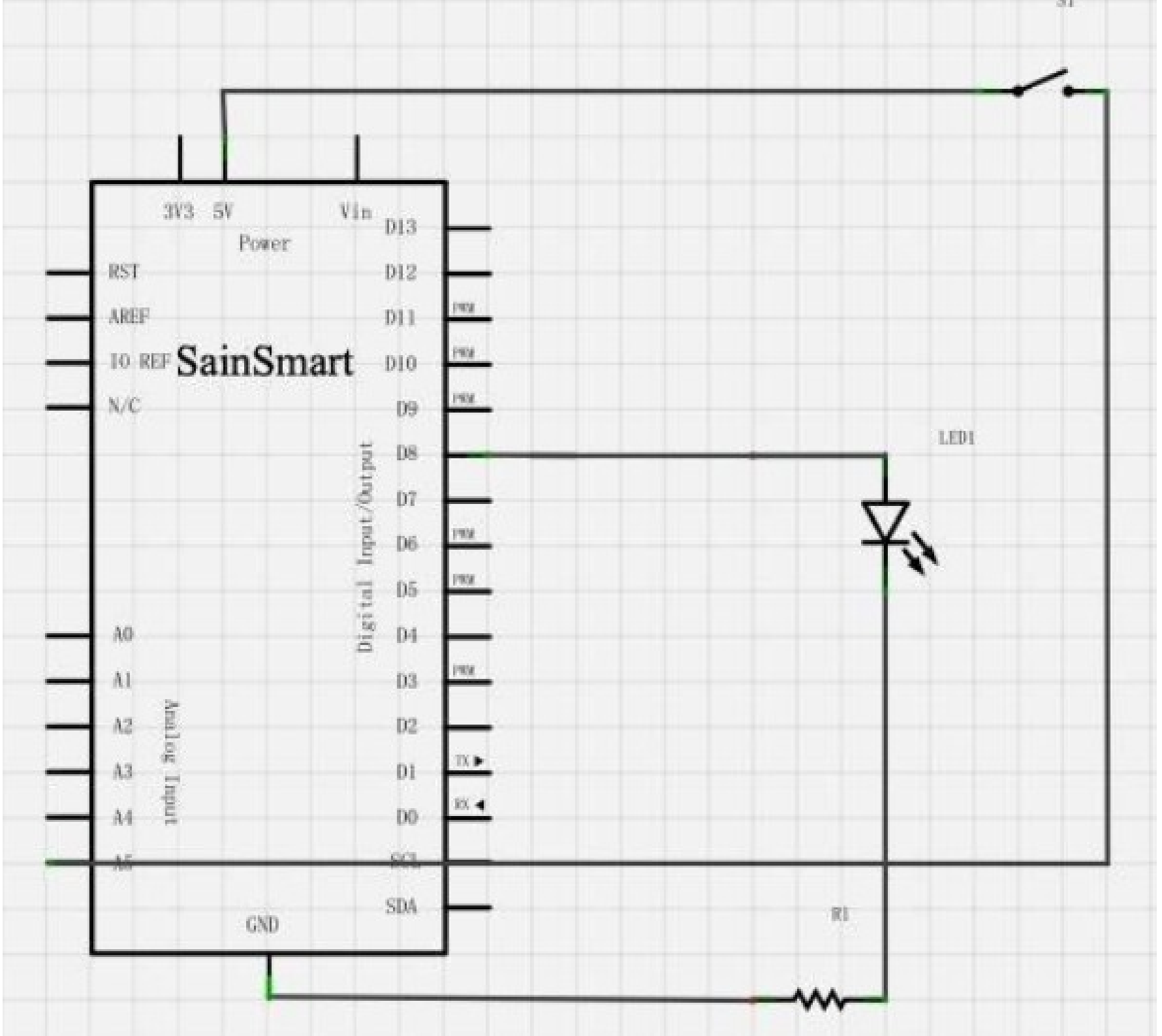


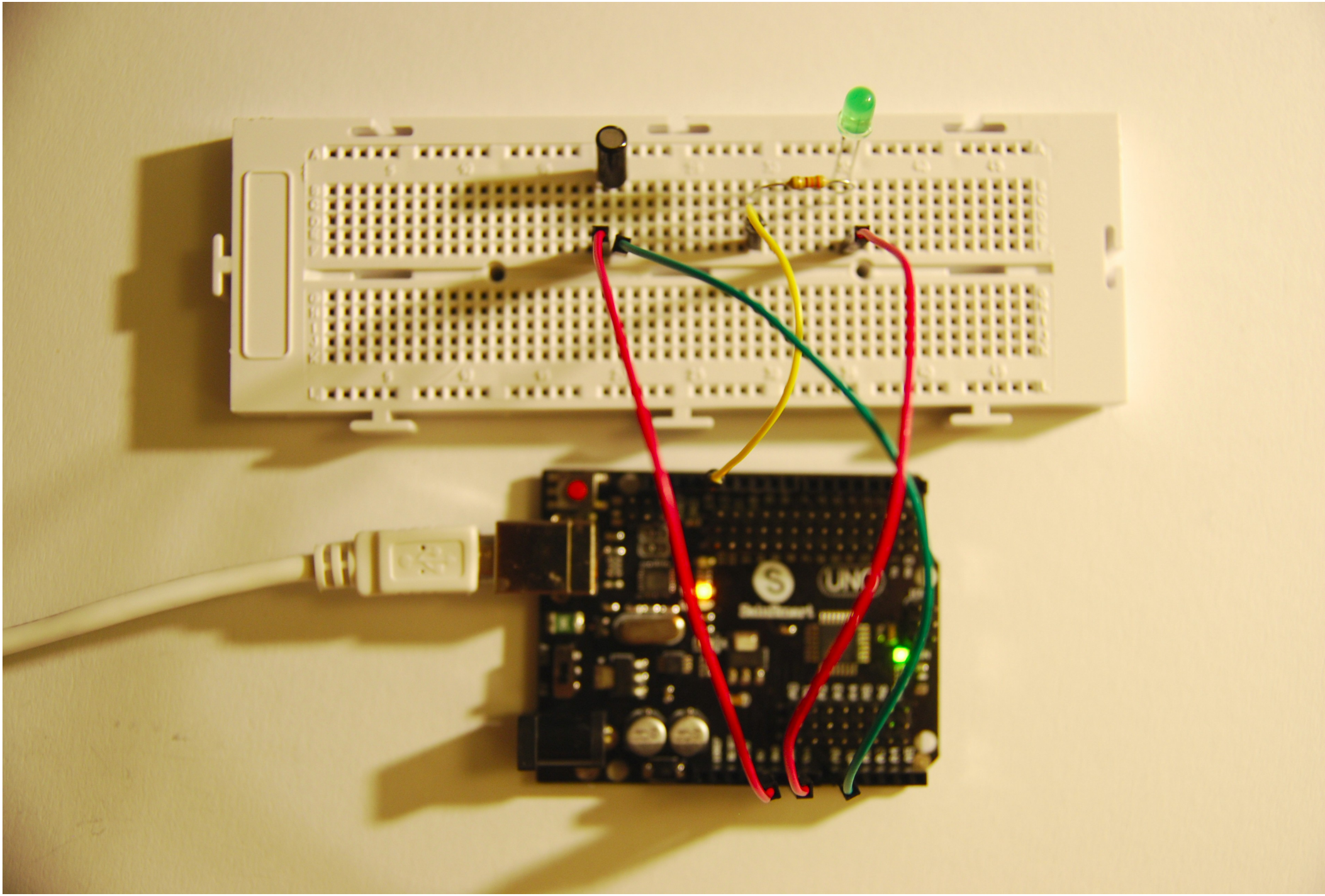

AREF
GND
13 12 11 10 9 8 7 6 5 4 3 2 1 0
DIGITAL
PWM
PWM
PWM
PWM
PWM
PWM
TX
RX
L
TX
RX
SainSmart
PWR
PWR SEL
USB
EXT
ICSP
www.sainsmart.com
RESET
3V3
5V
Gnd
Vin
POWER
ANALOG IN
0 1 2 3 4 5




AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
DIGITAL
PWM
TX
RX
L
TX
RX
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
3V3
5V
Gnd
9V
POWER
ANALOG IN
0
1
2
3
4
5


UNO
SainSmart


3V3
5V
Vin
Power
RST
AREF
IO REF
N/C
SainSmart
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
SCL
SDA
PWM
TX
RX
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
LED1
R1



Referencias

- Arduino web site
  - http://arduino.cc/en/Guide/Environment
  - http://arduino.cc/en/Tutorial/HomePage
- Adafruit tutorial #1 and 2
  - http://www.ladyada.net/learn/arduino/lesson2.html
- Leah Buechley’s Introduction to Arduino
  - http://web.media.mit.edu/~leah/LilyPad/03_arduino_intro.html