

# Curso avanzado sobre Arduino

Arduino Avanzado



ElCacharreo.com



# Arduino Intermedio: Presente



# Arduino Intermedio: Presente

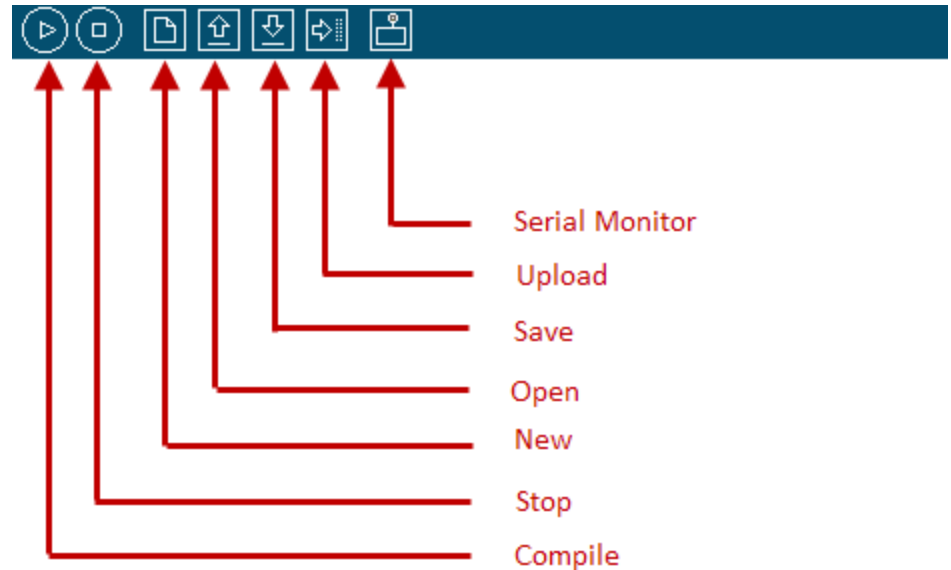
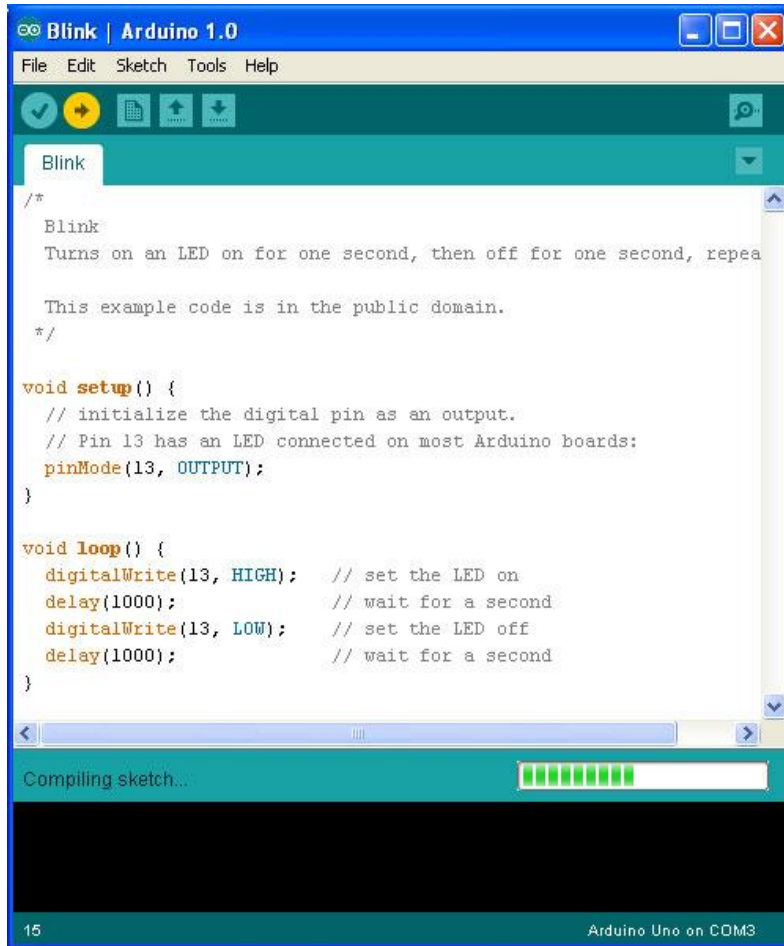


José Antonio Vacas Martínez

blog  
javacasm@elcacharreo.com  
twitter  
linkedin



# Programando Arduino: IDE



# Programando Arduino: Lenguaje

```
void setup()  
{}
```

```
void loop()  
{}
```

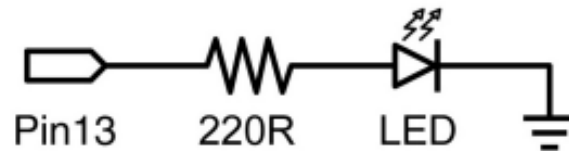
[Guía de referencia de Arduino](#)

[Librerías](#)



# El mundo digital: salidas

## digital output



This is the basic 'hello world' program used to simply turn something on or off. In this example, an LED is connected to pin13, and is blinked every second. The resistor may be omitted on this pin since the Arduino has one built in.

```
int ledPin = 13;                // LED on digital pin 13

void setup()                    // run once
{
  pinMode(ledPin, OUTPUT);      // sets pin 13 as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // turns the LED on
  delay(1000);                  // pauses for 1 second
  digitalWrite(ledPin, LOW);    // turns the LED off
  delay(1000);                  // pauses for 1 second
}
```

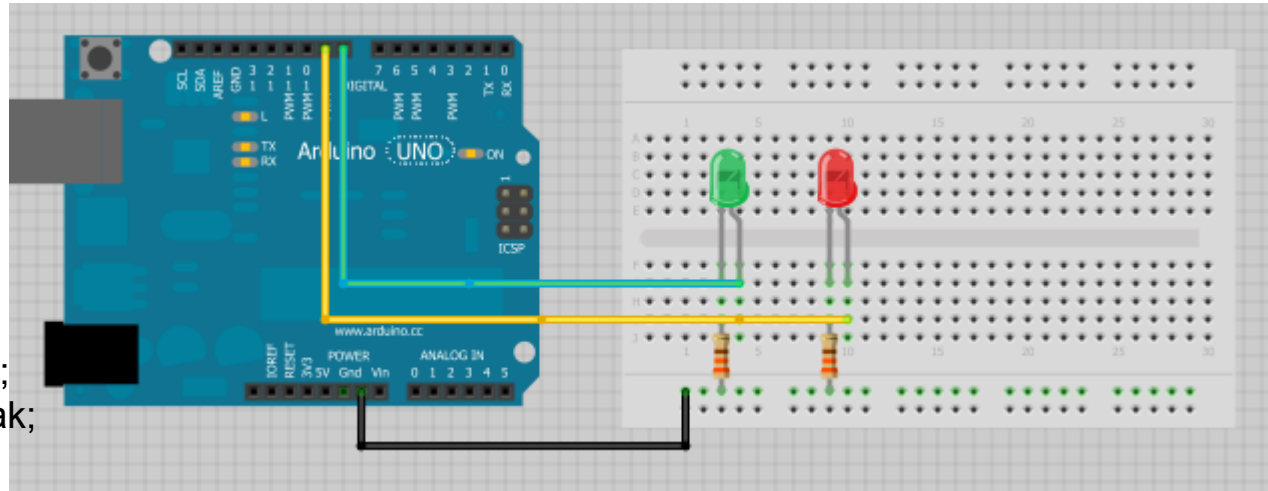


# El mundo digital: Ejemplo semáforo

```
int iPinVerde=2,iPinRojo=3,iPinAmarillo=4;

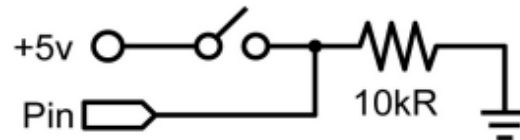
#define iEstadoVerde 1
#define iEstadoAmarillo 2
#define iEstadoRojo 3
void setup() {
  pinMode(iPinVerde,OUTPUT);pinMode(iPinAmarillo,OUTPUT);pinMode(iPinRojo,OUTPUT);}
int iEstad0=iEstadoVerde;
int iPin=iPinVerde;

void loop() {
  delay(1000);
  digitalWrite(iPin,LOW);
  switch(iEstado) {
    case iEstadoVerde:
      iEstado=iEstadoAmarillo;
      iPin=iPinVerde;    break;
    case iEstadoAmarillo:
      iEstado=iEstadoRojo;
      iPin=iPinRojo;    break;
    case iEstadoRojo:
      iEstado=iEstadoVerde;
      iPin=iPinVerde;    break; }
  digitalWrite(iPin,HIGH); } }
```



# El mundo digital: entradas

## digital input



This is the simplest form of input with only two possible states: on or off. This example reads a simple switch or pushbutton connected to pin2. When the switch is closed the input pin will read HIGH and turn on an LED.

```
int ledPin = 13;           // output pin for the LED
int inPin = 2;             // input pin (for a switch)

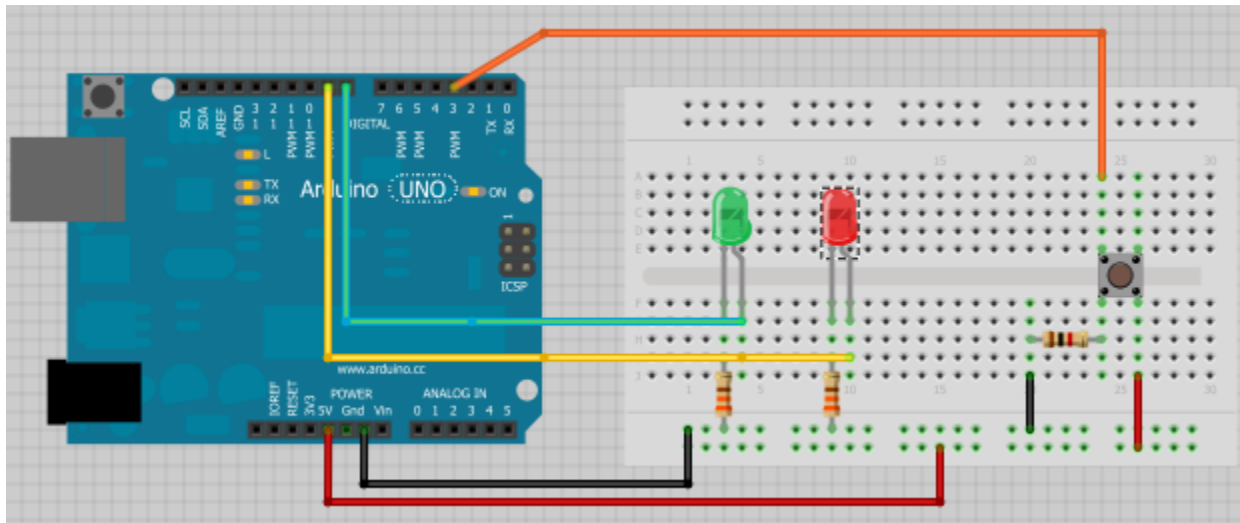
void setup()
{
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inPin, INPUT);   // declare switch as input
}

void loop()
{
  if (digitalRead(inPin) == HIGH) // check if input is HIGH
  {
    digitalWrite(ledPin, HIGH); // turns the LED on
    delay(1000);                // pause for 1 second
    digitalWrite(ledPin, LOW);  // turns the LED off
    delay(1000);                // pause for 1 second
  }
}
```



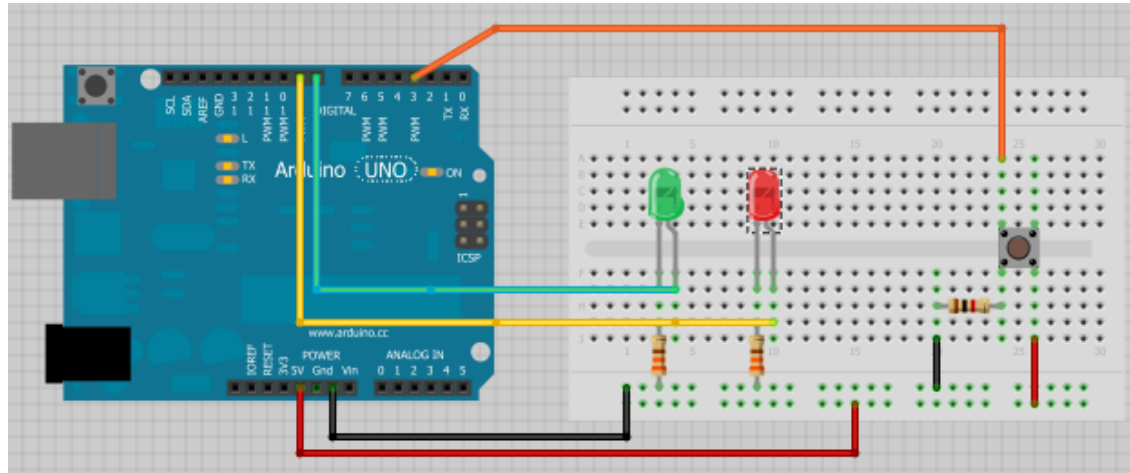


# El mundo digital: Ejemplo semáforo con pulsador



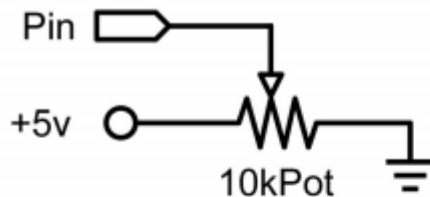
# El mundo digital: Ejemplo semáforo con pulsador con estados

```
int iPinVerde=9,iPinRojo=8 ,iBoton=3;
#define iEstadoVerde 1
#define iEstadoRojo 3
void setup() {
  pinMode(iPinVerde,OUTPUT);pinMode(iPinRojo,OUTPUT); pinMode(iBoton,INPUT);}
int iEstad0=iEstadoVerde;
int iPin=iPinVerde;
void loop() {
  delay(1000);
  if(digitalRead(iBonton)==HIGH)
    iEstado=iEstadoRojo;
  digitalWrite(iPin,LOW);
  switch(iEstado) {
    case iEstadoVerde:
      iEstado=iEstadoRojo;
      iPin=iPinVerde;    break;
    case iEstadoRojo:
      iEstado=iEstadoVerde;
      iPin=iPinVerde;    break; }
  digitalWrite(iPin,HIGH); } }
```



# El mundo analógico: entradas

## potentiometer input



Using a potentiometer and one of the Arduino's analog-to-digital conversion (ADC) pins it is possible to read analog values from 0-1024. The following example uses a potentiometer to control an LED's rate of blinking.

```
int potPin = 0;    // input pin for the potentiometer
int ledPin = 13;   // output pin for the LED

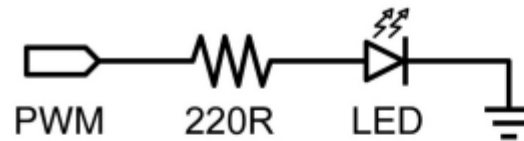
void setup()
{
  pinMode(ledPin, OUTPUT); // declare ledPin as OUTPUT
}

void loop()
{
  digitalWrite(ledPin, HIGH); // turns ledPin on
  delay(analogRead(potPin));  // pause program
  digitalWrite(ledPin, LOW);  // turns ledPin off
  delay(analogRead(potPin));  // pause program
}
```



# El mundo analógico: salidas

## pwm output



Pulsewidth Modulation (PWM) is a way to fake an analog output by pulsing the output. This could be used to dim and brighten an LED or later to control a servo motor. The following example slowly brightens and dims an LED using for loops.

```
int ledPin = 9;    // PWM pin for the LED

void setup(){}     // no setup needed

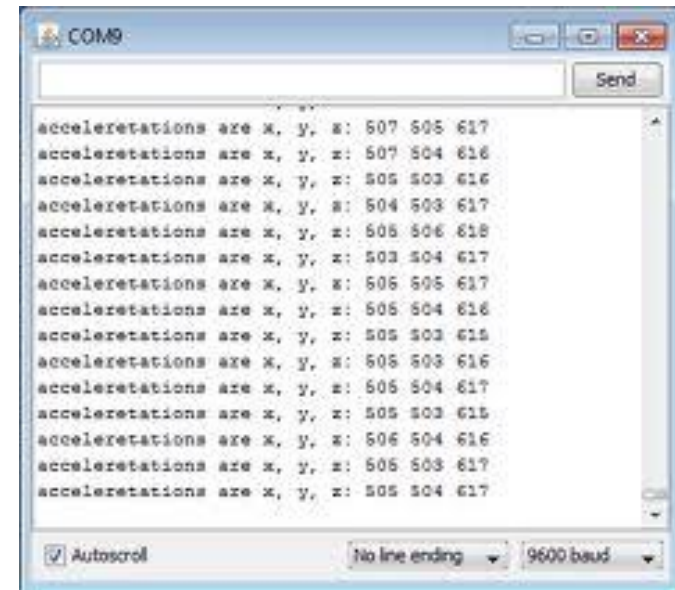
void loop()
{
  for (int i=0; i<=255; i++) // ascending value for i
  {
    analogWrite(ledPin, i); // sets brightness level to i
    delay(100);             // pauses for 100ms
  }
  for (int i=255; i>=0; i--) // descending value for i
  {
    analogWrite(ledPin, i); // sets brightness level to i
    delay(100);             // pauses for 100ms
  }
}
```



# Comunicaciones: introducción

## Comunicando con el pc:

```
void setup() {  
  Serial.begin(9600);}  
  
int i=0;  
void loop() {  
  Serial.print("hola ");  
  Serial.println(i);  
}
```



# Comunicaciones: el puerto serie

## Comandos via serie

### Functions

- begin()
- end()
- available()
- read()
- peek()
- flush()
- print()
- println()
- write()
- SerialEvent()



# Comunicaciones: SPI, I2C, OneWire

Ejemplo [I2C](#)

Ejemplo OneWire del ide

[Ejemplos ....](#)



# Conclusiones

Gracias por vuestra atención

