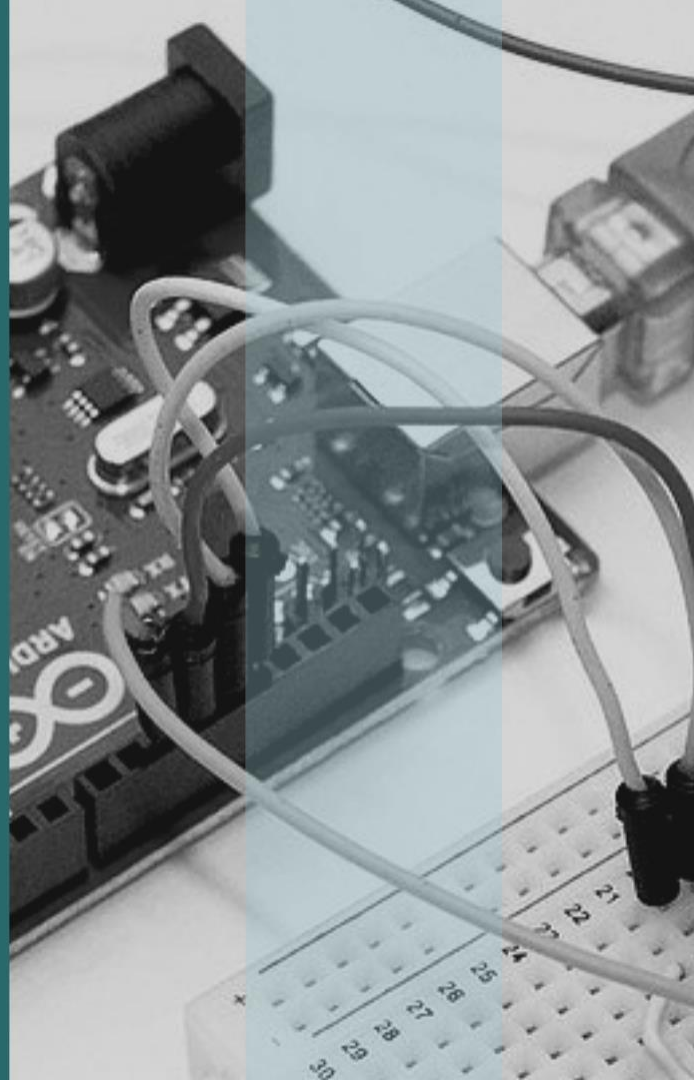


Flisol 2019

How to basics

FUNCIONALIDADES BÁSICAS DE ARDUINO



Contenido

PUNTOS CLAVE

- Entradas y salidas digitales
- Entradas analógicas
- PWM
- Puerto serie
- Conexión y funcionamiento
- Programación Básica.

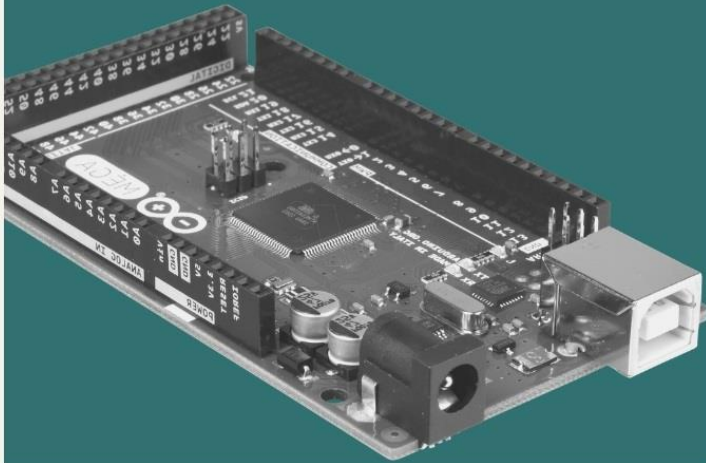


Introducción

QUE ES UN ARDUINO?

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable y una serie de pines, los que permiten establecer conexiones entre el microcontrolador y los diferentes sistemas electrónicos de una manera muy sencilla.

How to basics



¿POR QUÉ USAR ARDUINO?

- Arduino tiene una gran comunidad
- Su entorno de programación es multiplataforma
- Lenguaje de programación de fácil comprensión
- Bajo costo
- Re-usabilidad y versatilidad

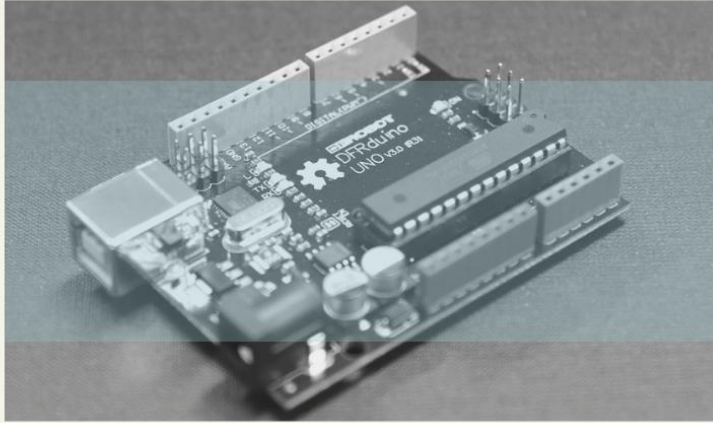
Como funciona un programa?

VOID SETUP()

Es la función principal que se ejecuta una sola vez, cuando se enciende el dispositivo. sirve para configurar y verificar.

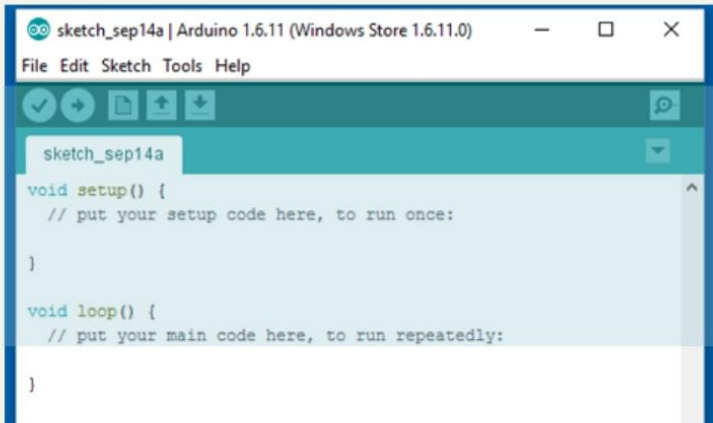
VOID LOOP()

Es la función principal que se repite ciclicamente, en forma de bucle. En ella se ejecuta el código programado, una vez que llega al final del código, vuelve a iniciar desde el principio de la función misma.



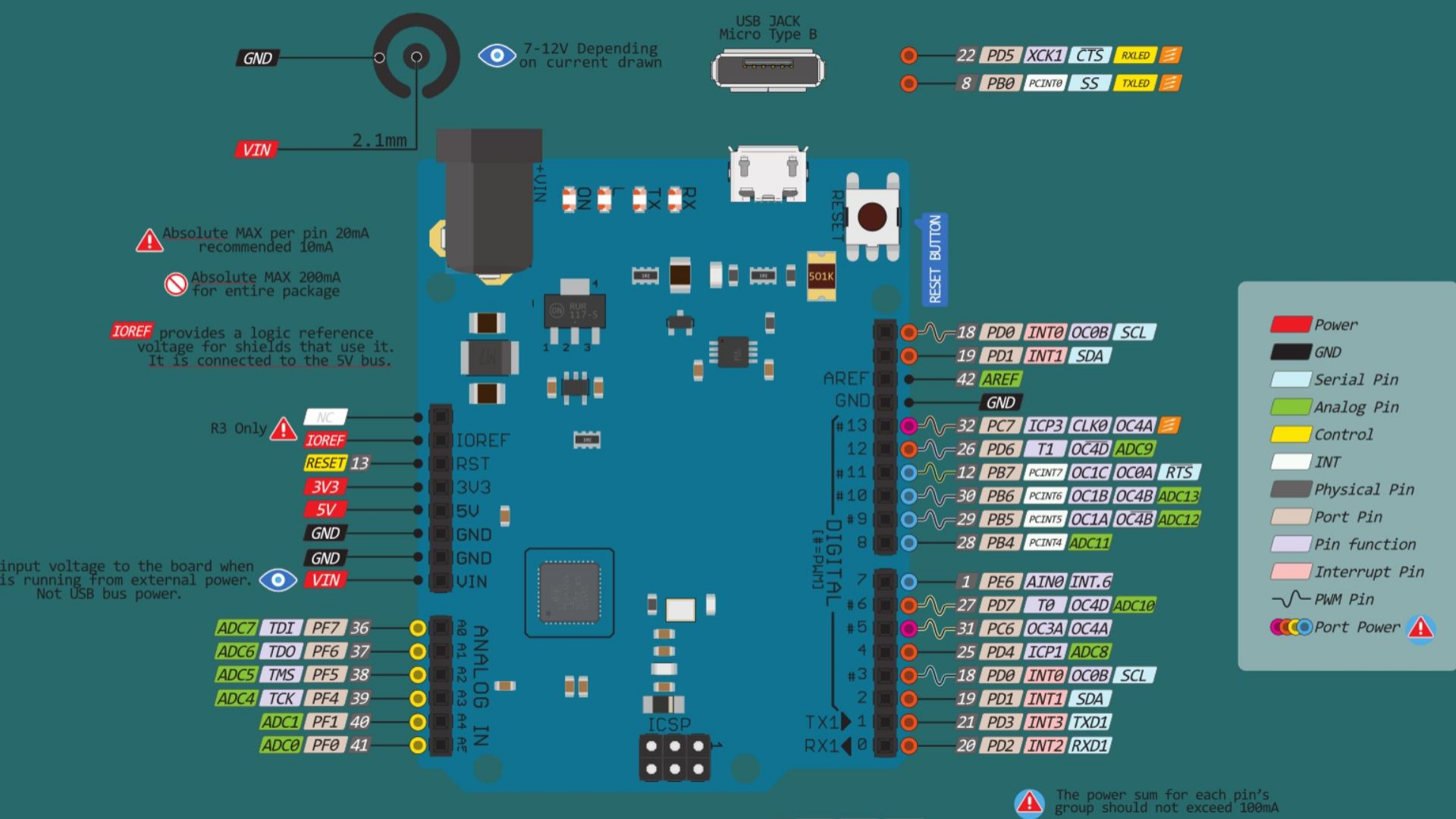
HARDWARE

La placa de Arduino, que es la que va a cumplir con el comportamiento predeterminado por el software desarrollado



SOFTWARE

Instrucciones que se le brindara al dispositivo, para que opere de la manera deseada.



1. Digital

SALIDA/ENTRADA DIGITAL

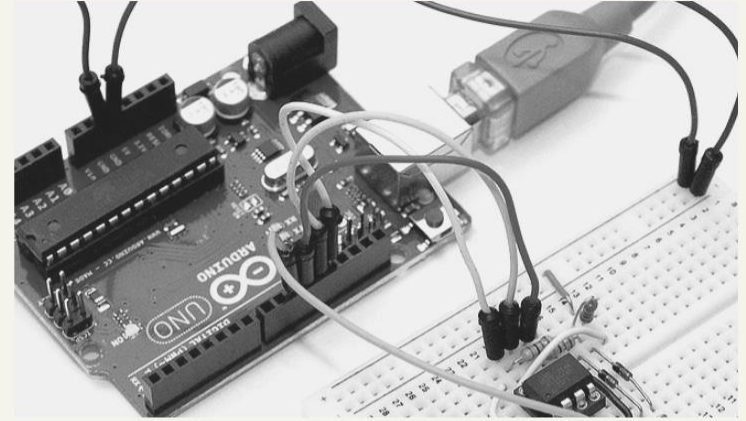
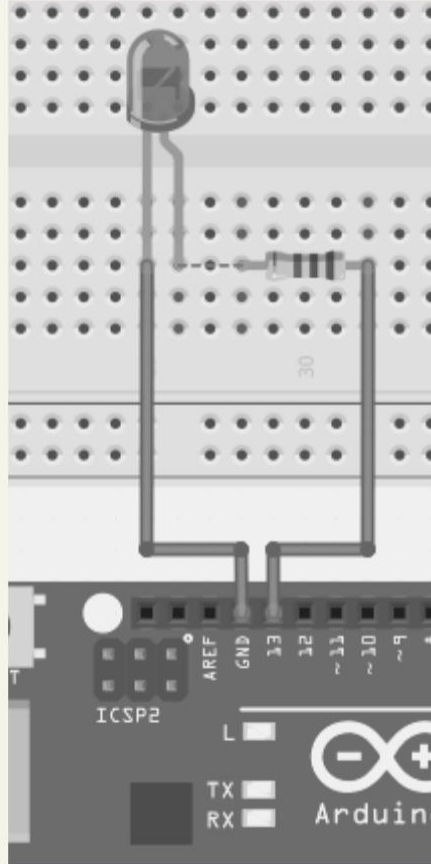
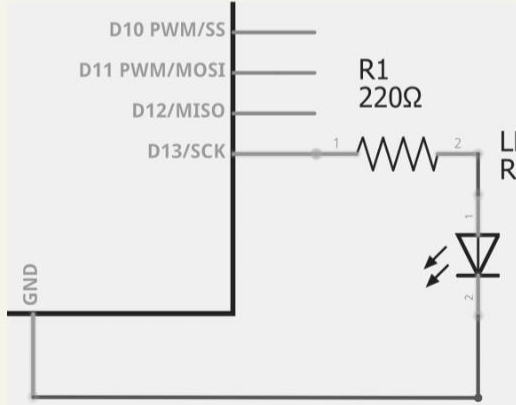
Nos referimos a digital como algún estado que pueda dividir solo en 2 tipos, como Verdadero o falso, alto o bajo, 0 o 1.

1.1 Salida

SALIDA DIGITAL

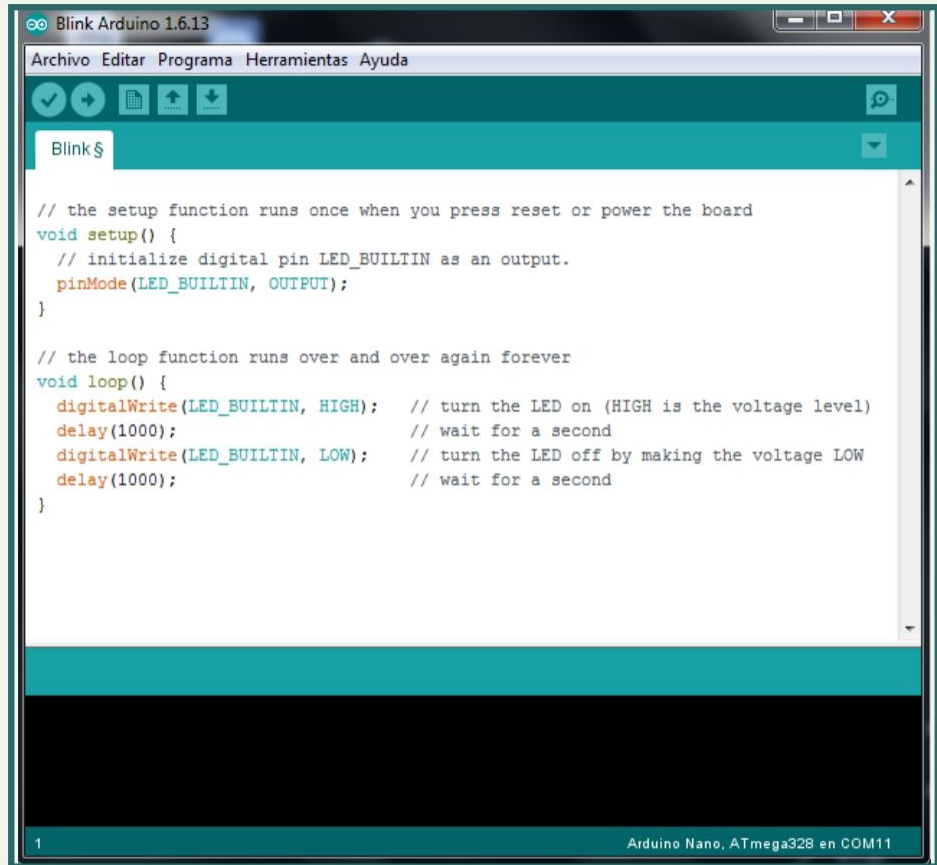
Una salida digital es ampliamente utilizada por ejemplo para encendido de luces, motores, intercambios de información etc. Hoy vamos a utilizarlo con un led para poder visualizar sus efectos.

DIAGRAMA DE CONEXIÓN



Arduino posee en el pin 13, por defecto un led que esta conectado en la placa, por lo que se vera como parpadea junto con el que estamos colocando.

Codificando



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. The toolbar contains icons for opening files, saving, and uploading. The code editor displays the following C++ code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

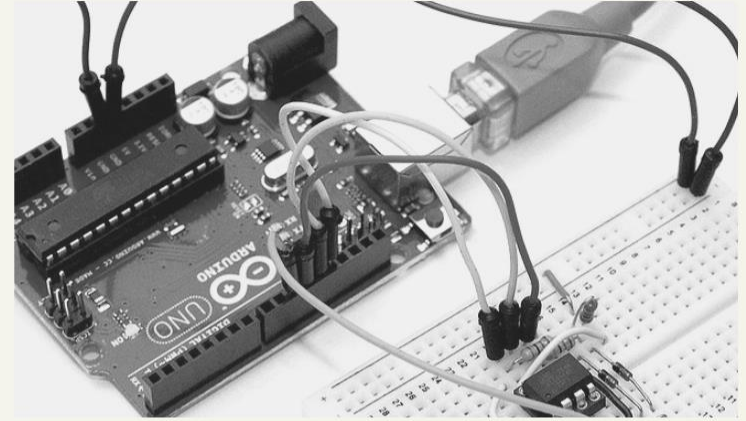
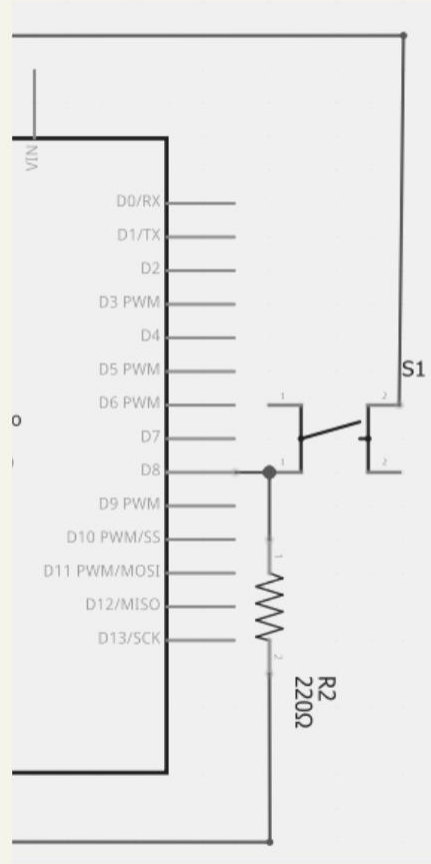
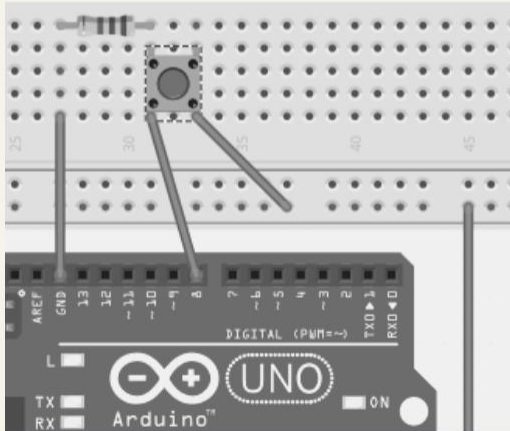
The status bar at the bottom indicates '1' on the left and 'Arduino Nano, ATmega328 en COM11' on the right.

1.2 Entrada

ENTRADA DIGITAL

Otro uso de lo mas útil para todo tipo de sensores, ya que nos permite que el microcontrolador obtenga información del exterior. Hoy la utilizaremos con pulsadores

DIAGRAMA DE CONEXIÓN



Vamos a conectar el pulsador en el pin 8. Es importante recordar que se debe colocar la resistencia a masa que se muestra en el diagrama, para no estar recibiendo estática.

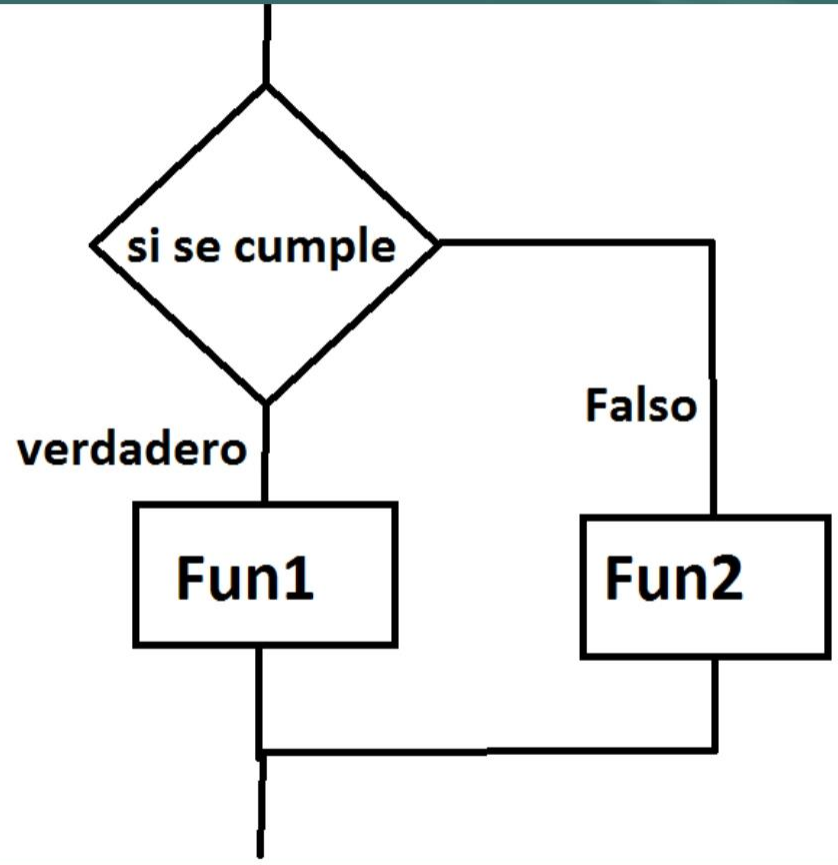
1.3 IF

LASOS DE CONTROL

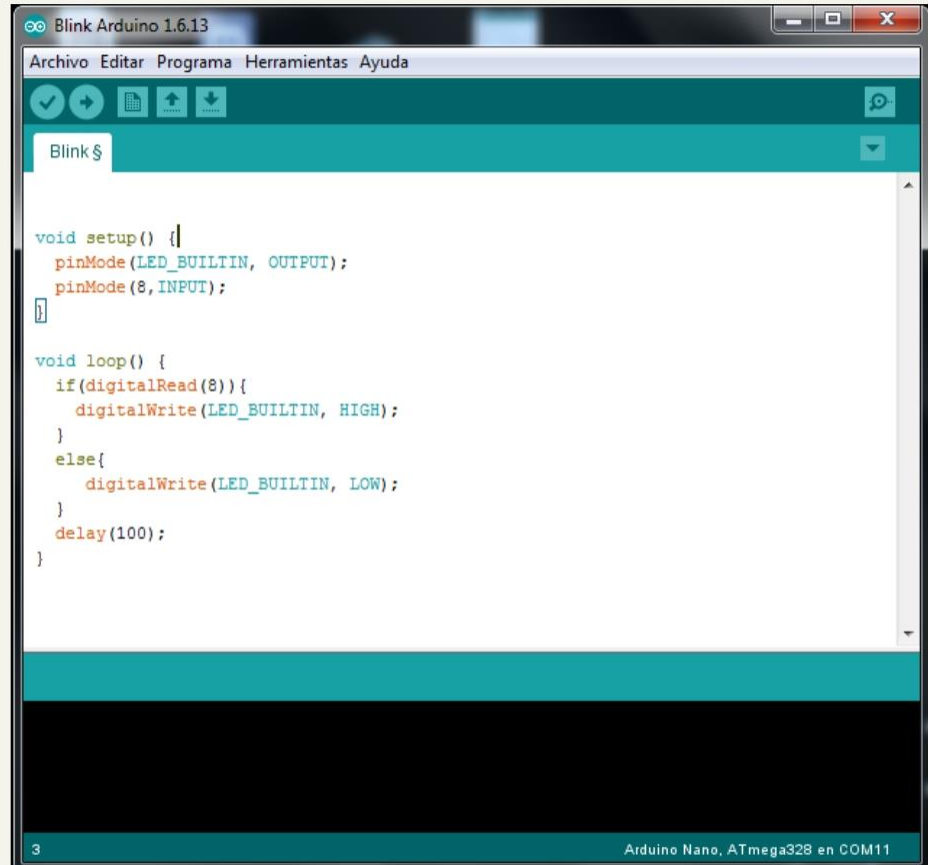
El lazo de control if, es el mas usado en programación. Se interpreta como una pregunta, donde si la respuesta es verdadero se ejecuta lo que esta en la función. De ser falso, puede no hacerse nada, o anexando else {} agregar código para realizar en ese caso. solo se ejecuta uno de los 2.

UTILIZACIÓN DE IF ELSE

```
if(Argumento){  
funcion1();  
}  
else{  
funcion2();  
}
```



Codificando



The screenshot shows the Arduino IDE interface with a custom sketch named "Blink \$". The code is as follows:

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(8, INPUT);  
}  
  
void loop() {  
  if(digitalRead(8)) {  
    digitalWrite(LED_BUILTIN, HIGH);  
  }  
  else {  
    digitalWrite(LED_BUILTIN, LOW);  
  }  
  delay(100);  
}
```

The status bar at the bottom indicates the board is "Arduino Nano, ATmega328 en COM11".

2. Analógico

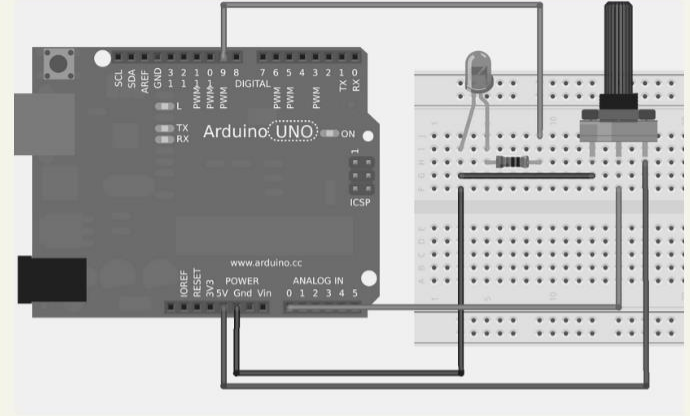
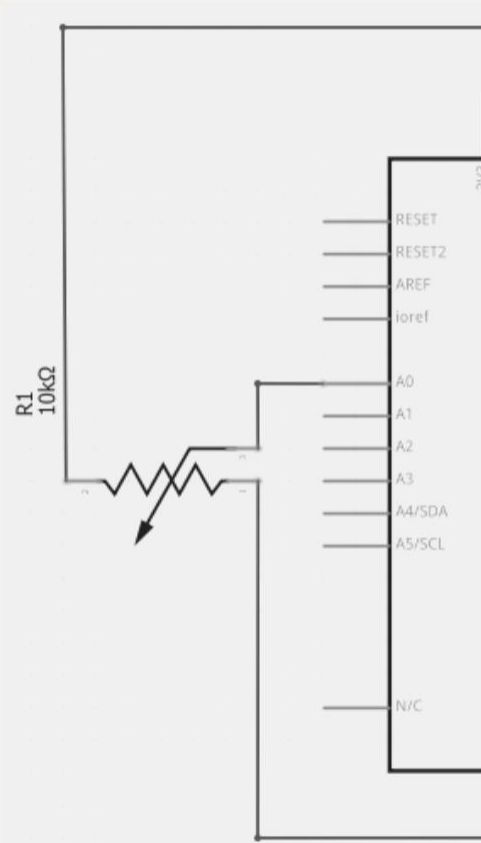
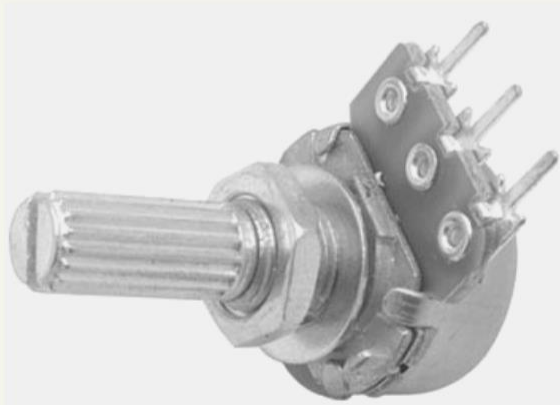
Consiste en interpretar distintos

ENTRADAS ANALÓGICAS

valores de tensión numéricamente,
(en este caso, valores de 0 a 1023).

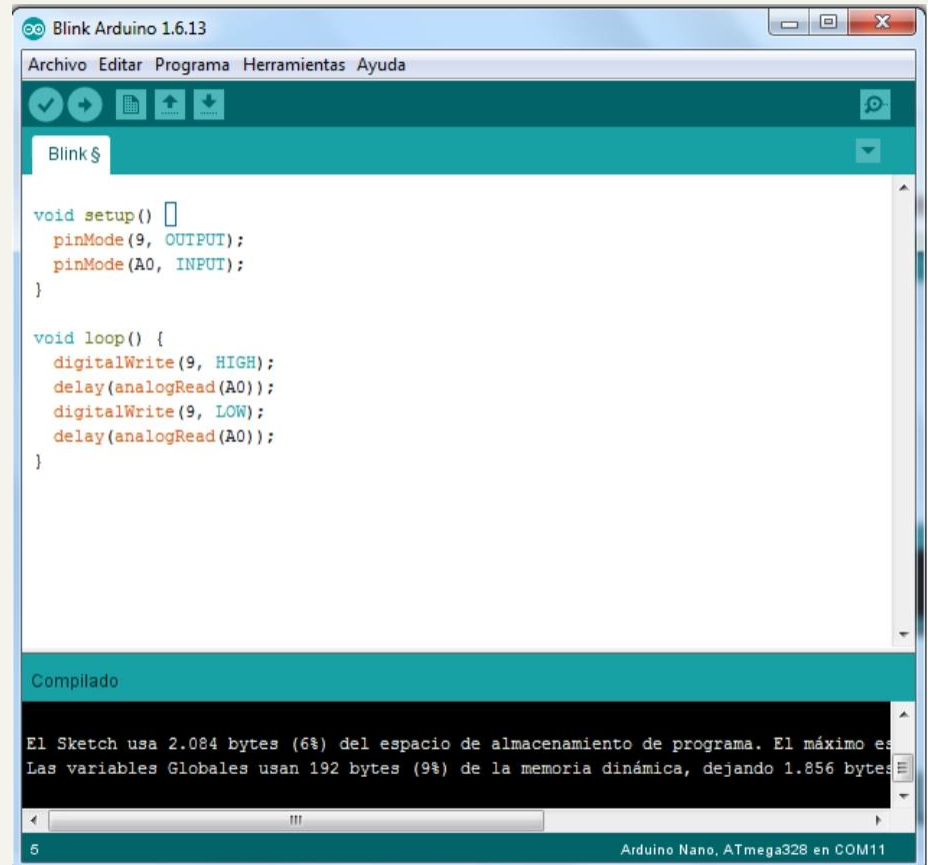
Utilizamos la función `AnalogRead(A)`;
para obtener este valor, donde A
puede ser, por ejemplo A0,
significando que es el pin analógico 0
de Arduino

ANALÓGICO



Conectamos un extremo a 5v, otro a GND, y el del medio al puerto analógico del Arduino.

Codificando



3. PWM

MODULACIÓN DE ANCHO DE PULSO

Es una técnica que consiste en disminuir la cantidad de energía brindada, reduciendo el tiempo ciclo de trabajo. muy utilizada para el control de motores y otros componentes electrónicos.

PWM

sketch_apr24b Arduino 1.6.13

Archivo Editar Programa Herramientas Ayuda



sketch_apr24b \$

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  analogWrite(13, 64);  
}
```

Pulse Width Modulation

0% Duty Cycle – analogWrite(0)



25% Duty Cycle – analogWrite(64)



50% Duty Cycle – analogWrite(127)



75% Duty Cycle – analogWrite(191)



100% Duty Cycle – analogWrite(255)



25% Duty Cycle – analogWrite(64)



50% Duty Cycle – analogWrite(127)



75% Duty Cycle – analogWrite(191)



Esta señal cuadrada es la que enviamos, teniendo un rango de 0 a 255. Equivale a encender y apagar el led rápidamente para nuestro ejemplo.

DIGITALWRITE(A,S)

comando para escribir una salida digital, en el pin A, siendo S HIGH o LOW (alto o bajo)

DIGITALREAD(A)

Realizar una lectura digital del pin A, siendo este un valor alto o bajo.

DELAY(MS)

Tiempo de espera en el código, en milisegundos. delay(1000); equivale a un segundo de espera.

PINMODE(A,M)

Definición de configuración para pines. siendo A el pin, y m el modo de uso, siendo INPUT u OUTPUT

ANALOGREAD(A)

Con esta función podemos obtener el valor de lectura en un rango de 0 a 5v, representado desde 0 a 1023.

ANALOGWRITE(A,P)

Para dar una salida PWN, al pin A, en P usaremos un valor entre 0 y 255.

4 Puerto serie

PUERTO SERIAL

Una comunicación serial, es un intercambio de información se que envía mediante un tren de pulsos.

4 Puerto serie

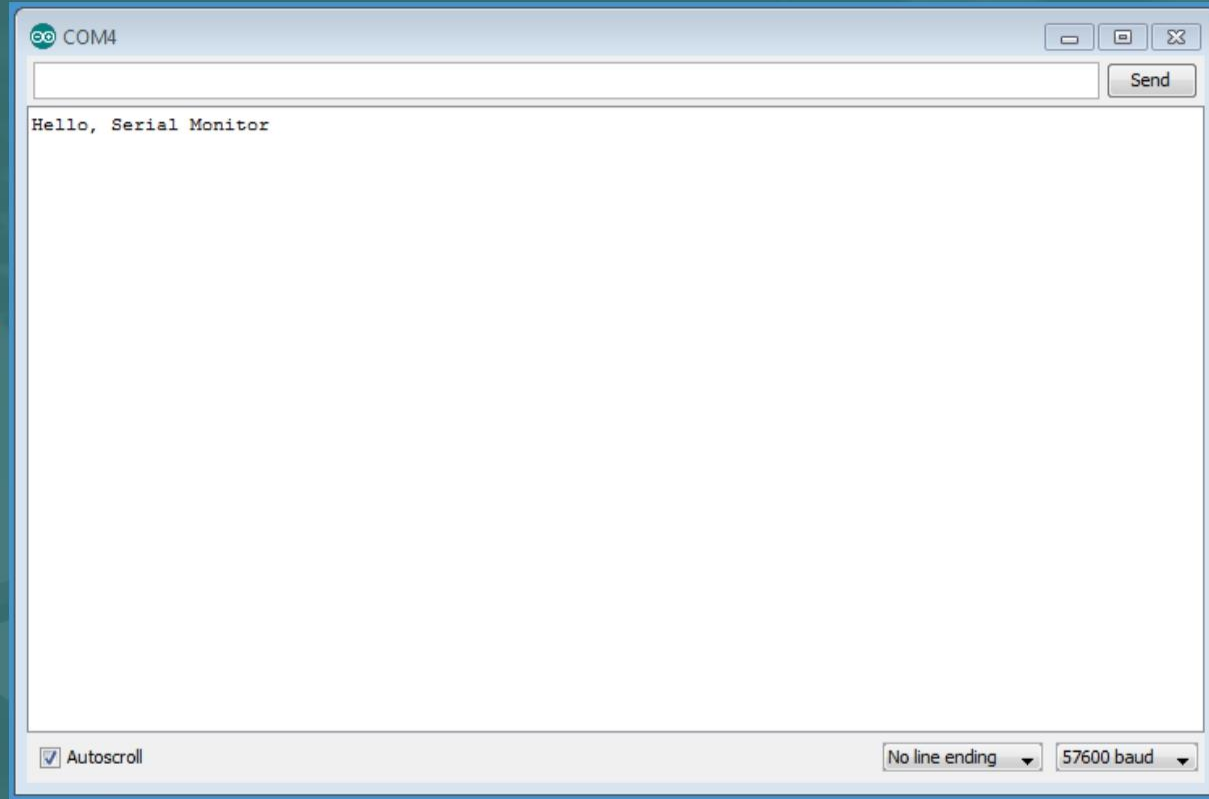
FUNCIONES

`Serial.begin(B)`; Es la función para configurar el puerto de serie de Arduino, donde se especifica la velocidad del puerto en Baudios.

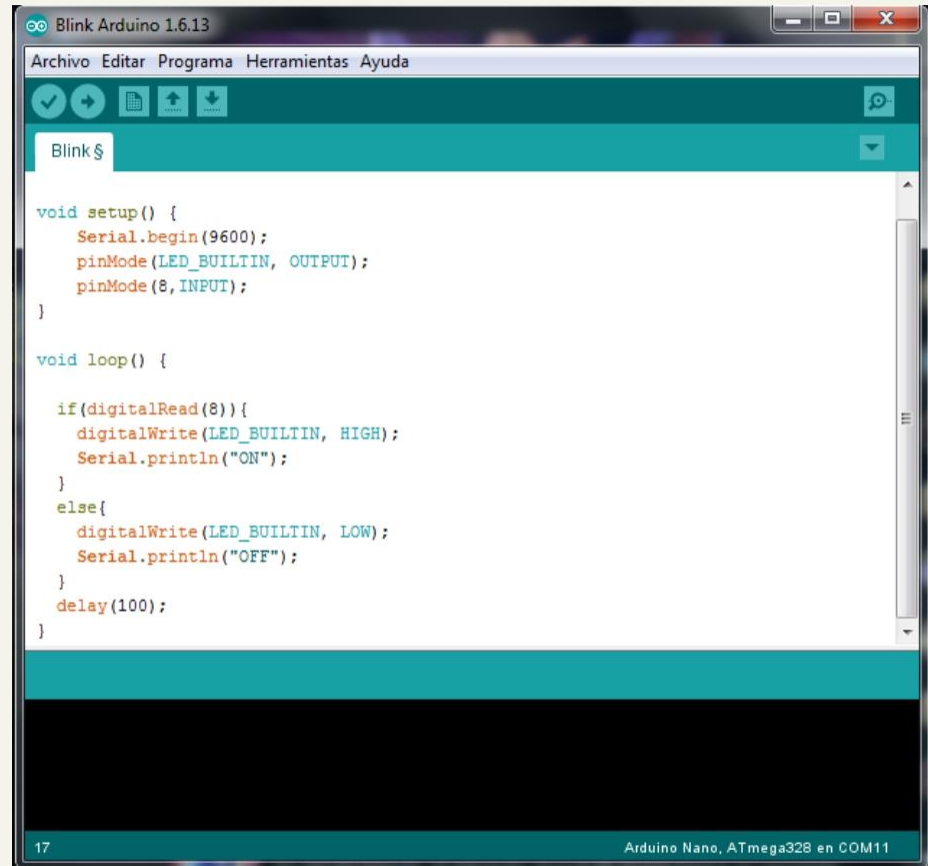
`Serial.println("Texto")`; Se utiliza para enviar datos desde el arduino. Para enviar texto simple, se coloca entre comillas dobles.

MONITOR SERIAL

Incorporado en
el IDE de arduino,
tenemos esta
herramienta que
nos permite
visualizar la
información
entrante



Codificando



The screenshot shows the Arduino IDE interface with the title bar "Blink Arduino 1.6.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for opening, saving, compiling, uploading, and erasing. The file name is "Blink \$". The code in the editor is as follows:

```
void setup() {  
  Serial.begin(9600);  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(8, INPUT);  
}  
  
void loop() {  
  
  if(digitalRead(8)){  
    digitalWrite(LED_BUILTIN, HIGH);  
    Serial.println("ON");  
  }  
  else{  
    digitalWrite(LED_BUILTIN, LOW);  
    Serial.println("OFF");  
  }  
  delay(100);  
}
```

The status bar at the bottom indicates "17" on the left and "Arduino Nano, ATmega328 en COM11" on the right.