
Taller Robótica 1

Agenda

Introducción a Arduino

- Qué es Arduino
- Tipos de Arduino
- Entradas/salidas
- Tipos de sensores
- Motores
- Módulos de comunicación



Tinkercad

- Simulación de circuitos
- Limitaciones (comunicación entre arduinos)



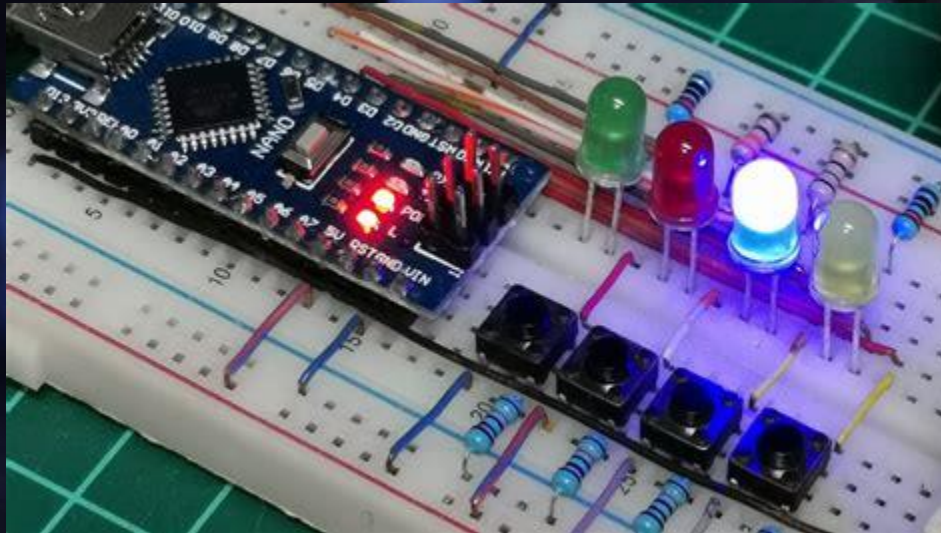
Otros simuladores

- Proteus 8

¿Qué es Arduino?

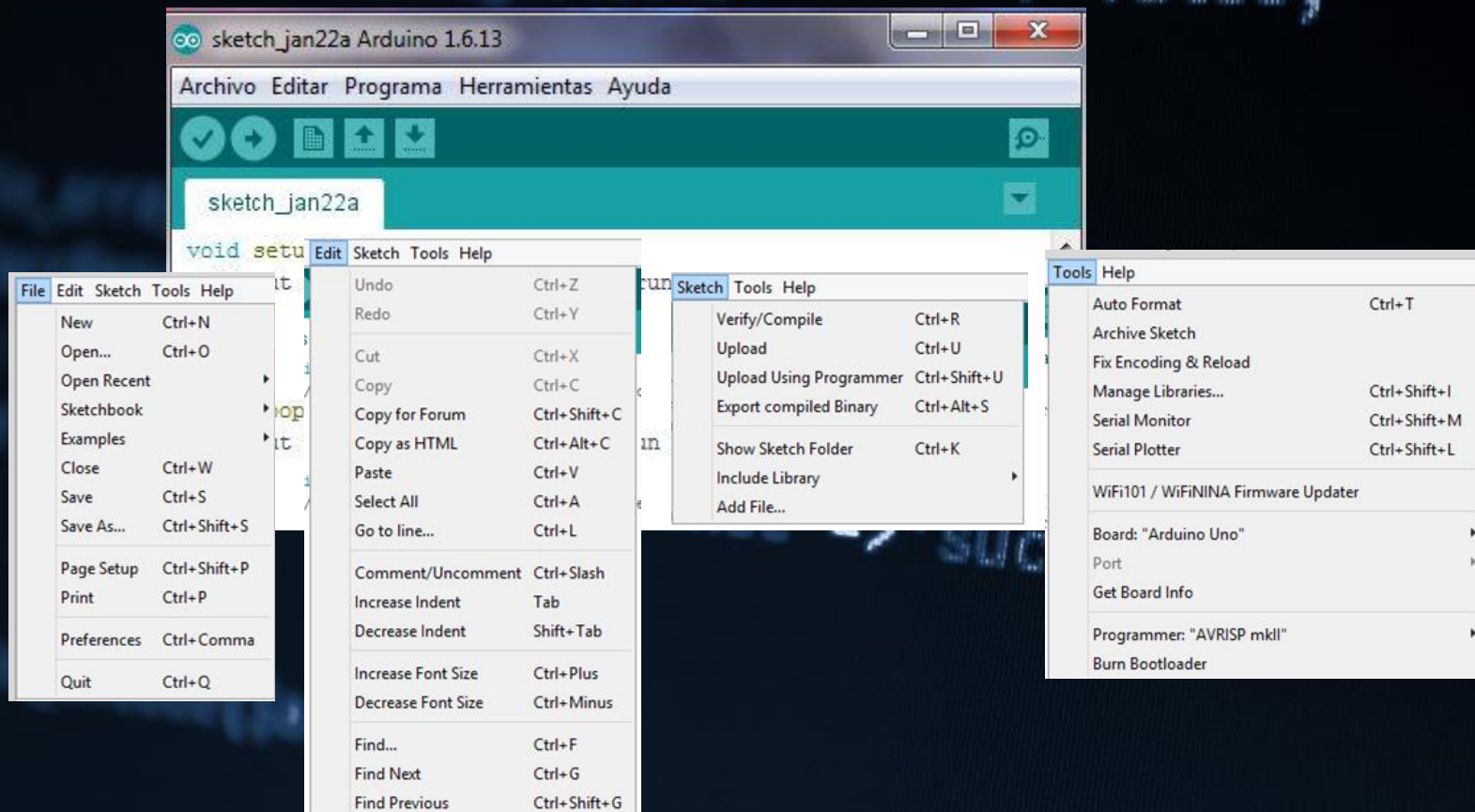
Arduino es una compañía de desarrollo de software y hardware libres creada en el año 2005 el Instituto de Diseño Interactivo de Ivrea (Italia). Su principal producto, el arduino, es una plataforma de electrónica "open-source" o de código abierto cuyos principios son contar con software y hardware fáciles de usar que tiene todos los elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador ATMEGA. Es decir, es una placa impresa con los componentes necesarios para que funcione el microcontrolador y su comunicación con un ordenador a través de la comunicación serial.

Las razones de su éxito global son su bajo coste, la facilidad y accesibilidad para aprender a usar la herramienta y el sencillo desarrollo de su software que además permite crear proyectos que se pueden comercializar.



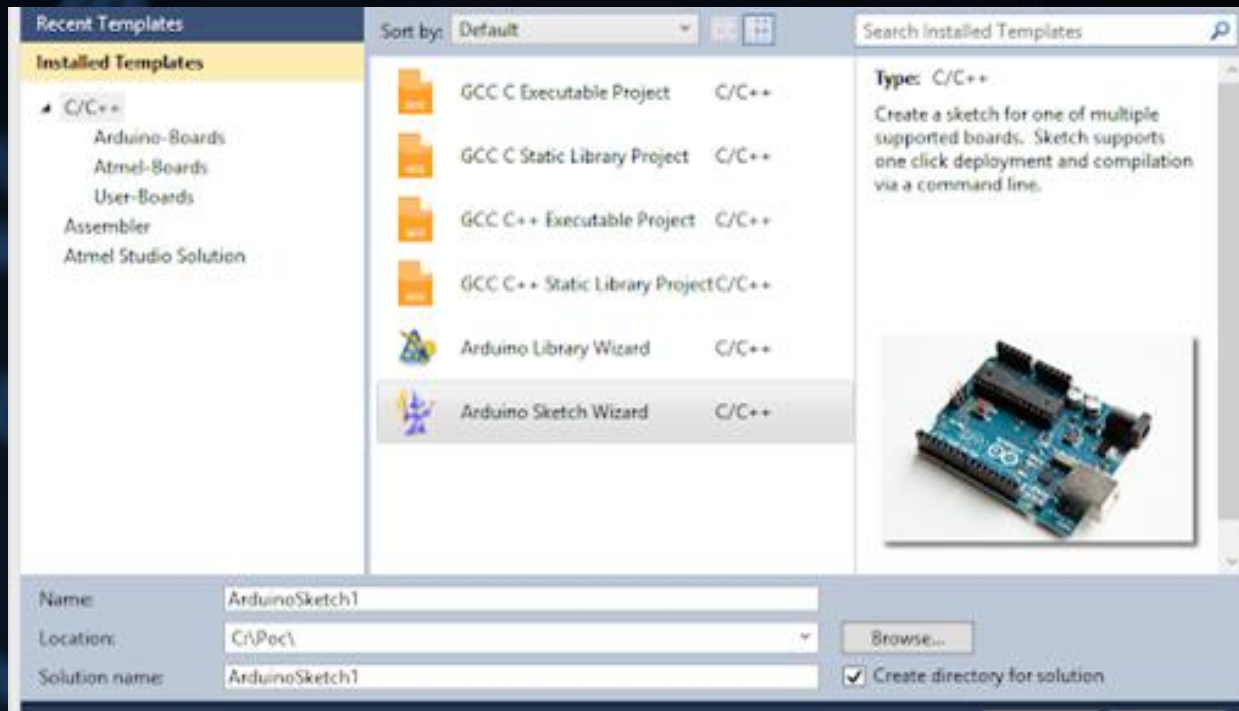
¿Cómo programar un Arduino?

El lenguaje de programación más utilizado para la programación en arduino es el propio de Arduino, basado en C++. Para crear, verificar y cargar un Sketh en Arduino se utiliza el IDE de Arduino, el IDE es el entorno de programación de Arduino.




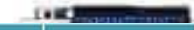




¿Cómo programar un Arduino?

Existen otras alternativas a la hora de programar nuestro Arduino si queremos cargar un Sketch escrito en otro lenguaje. Una de las herramientas más utilizadas es el ATMELE Studio, por medio del AVR Duda.



Tipos de Arduino

Modelo	Microcontrolador	Voltaje de entrada	Voltaje del sistema	Frecuencia de Reloj	Digital I/O	Entradas Analógicas	PWM	UART	Memoria Flash	Interfaz de Programación
Arduino Due	AT91SAM3X8E	5-12V	3.3V	84MHz	54	12	12	4	512Kb	Nativa USB
Arduino Leonardo	ATmega32U4	7-12V	5V	16MHz	20	12	7	1	32Kb	Nativa USB
Arduino Uno - R3	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	USB via ATmega16U2
Arduino Pro 3.3V/8MHz	ATmega328	3.35 -12V	3.3V	8MHz	14	6	6	1	32Kb	Cabecera compatible con FTDI
Arduino Pro 5V/16MHz	ATmega328	5 - 12V	5V	16MHz	14	6	6	1	32Kb	Cabecera compatible con FTDI
Ethernet	ATmega328	7-12V	5V	16MHz	14	6	6	1	32Kb	Cabecera compatible con FTDI
Arduino Mega 2560 R3	ATmega2560	7-12V	5V	16MHz	54	16	14	4	256Kb	USB via ATmega16U2
Arduino Mini 05	ATmega328	7-9V	5V	16MHz	14	6	8	1	32Kb	Cabecera Serial
Arduino Pro Mini 3.3V/8MHz	ATmega328	3.35 -12V	3.3V	8MHz	14	6	6	1	32Kb	Cabecera compatible con FTDI
Arduino Pro Mini 5V/16MHz	ATmega328	5 - 12V	5V	16MHz	14	6	6	1	32Kb	Cabecera compatible con FTDI
Arduino Fio	ATmega328P	3.35 -12V	3.3V	8MHz	14	8	6	1	32Kb	Cabecera compatible con FTDI o Inalámbrica via XBee ¹














Tabla extraída del sitio web: <https://www.zonamaker.com/arduino/intro-arduino/conociendo-arduino-introduccion>

Entradas y salidas Digitales

El arduino por tratarse de un sistema digital, usa la lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto, H y otro bajo. Si el nivel alto se representa por 1 y el bajo por 0, se habla de lógica positiva y en caso contrario de lógica negativa.

Función utilizada para leer el valor de un pin:

digitalRead(pin);

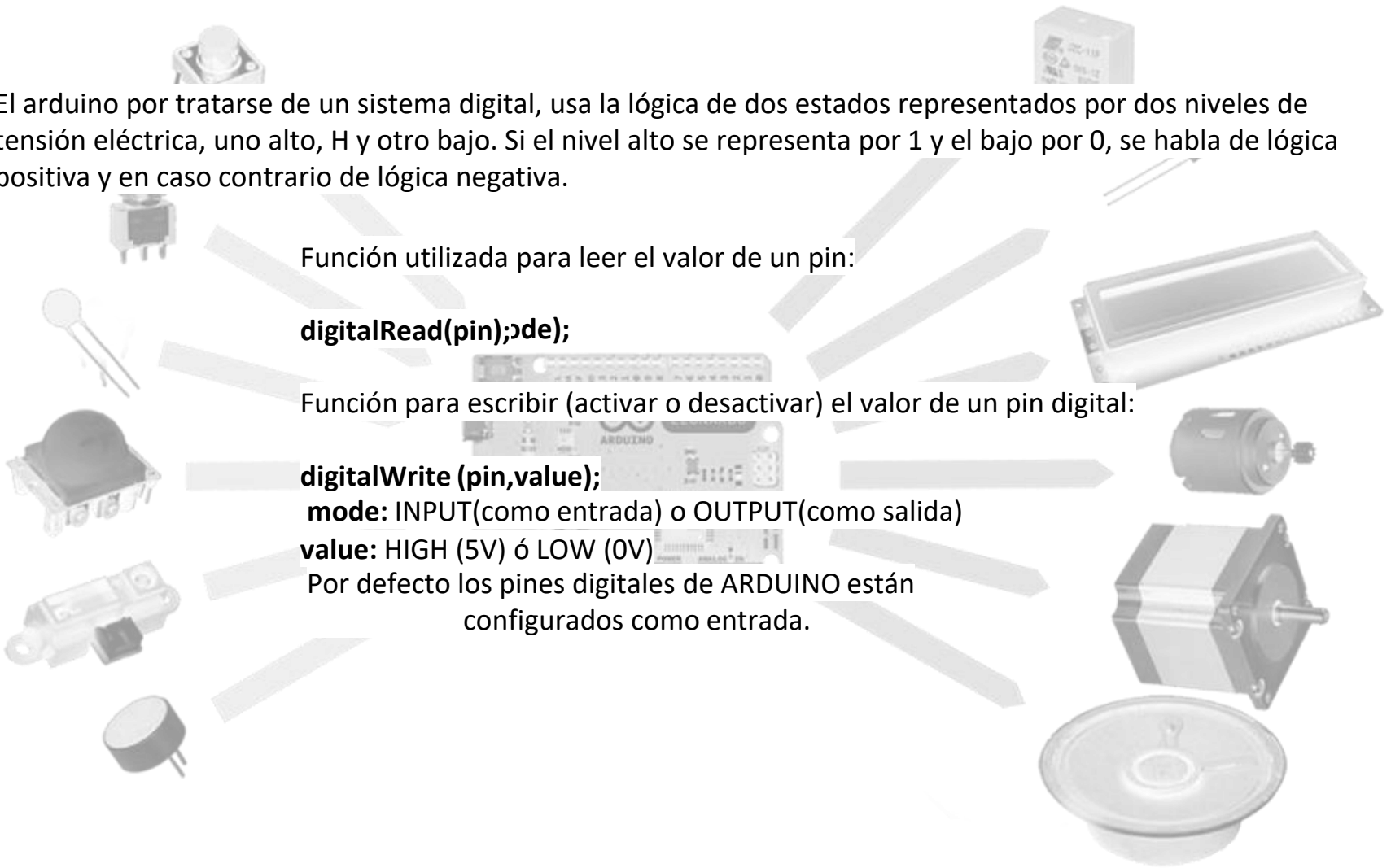
Función para escribir (activar o desactivar) el valor de un pin digital:

digitalWrite (pin,value);

mode: INPUT(como entrada) o OUTPUT(como salida)

value: HIGH (5V) ó LOW (0V)

Por defecto los pines digitales de ARDUINO están configurados como entrada.



Entradas y salidas Analógicas

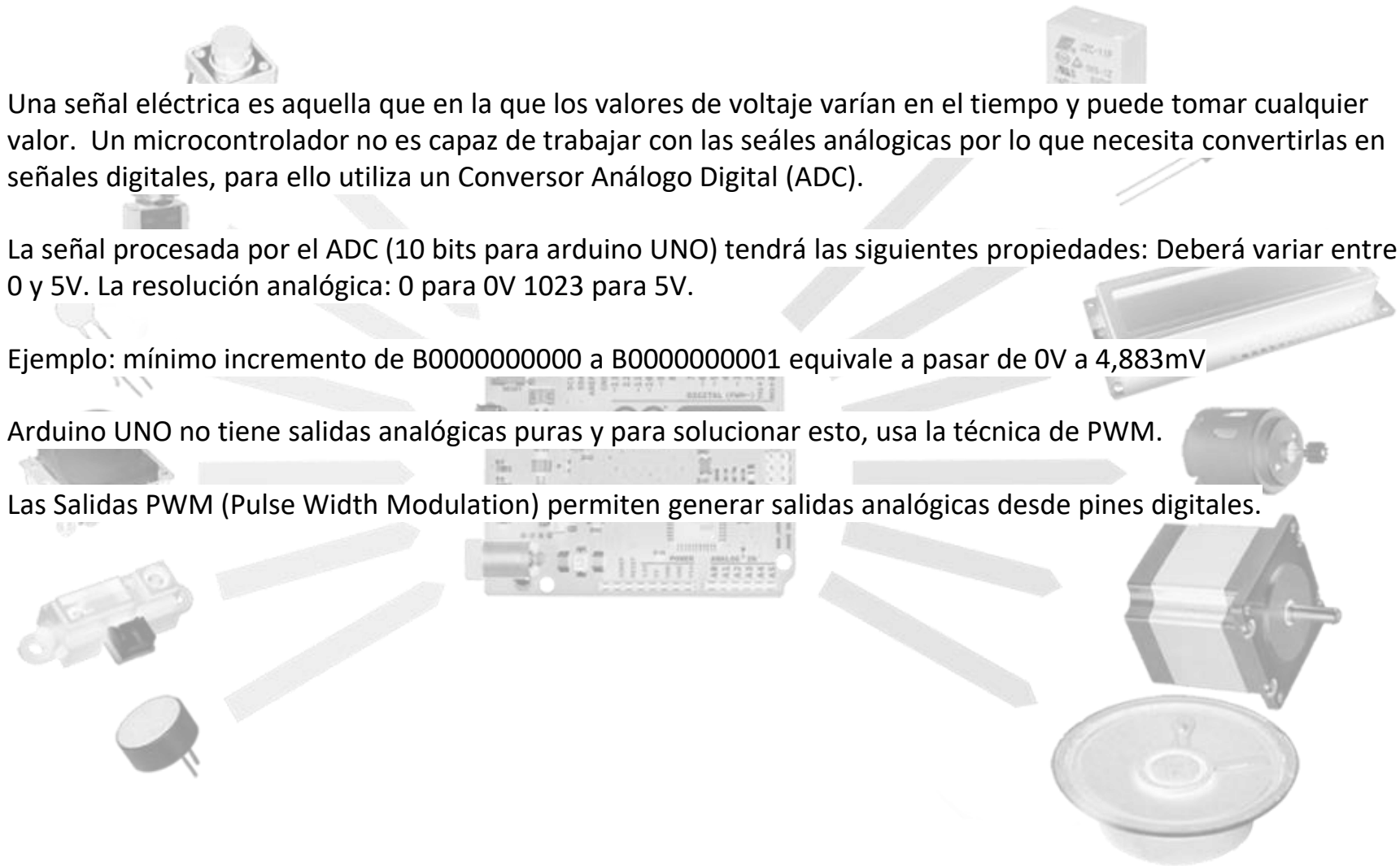
Una señal eléctrica es aquella que en la que los valores de voltaje varían en el tiempo y puede tomar cualquier valor. Un microcontrolador no es capaz de trabajar con las seales analógicas por lo que necesita convertirlas en señales digitales, para ello utiliza un Conversor Análogo Digital (ADC).

La señal procesada por el ADC (10 bits para arduino UNO) tendrá las siguientes propiedades: Deberá variar entre 0 y 5V. La resolución analógica: 0 para 0V 1023 para 5V.

Ejemplo: mínimo incremento de B0000000000 a B0000000001 equivale a pasar de 0V a 4,883mV

Arduino UNO no tiene salidas analógicas puras y para solucionar esto, usa la técnica de PWM.

Las Salidas PWM (Pulse Width Modulation) permiten generar salidas analógicas desde pines digitales.



Entradas y salidas Analógicas

analogRead(pin);

Lee valor desde un pin analógico, con una resolución de 10bits.

Los pines analógicos no necesitan ser inicializados en la función **setup()**.

analogWrite (pin, value);

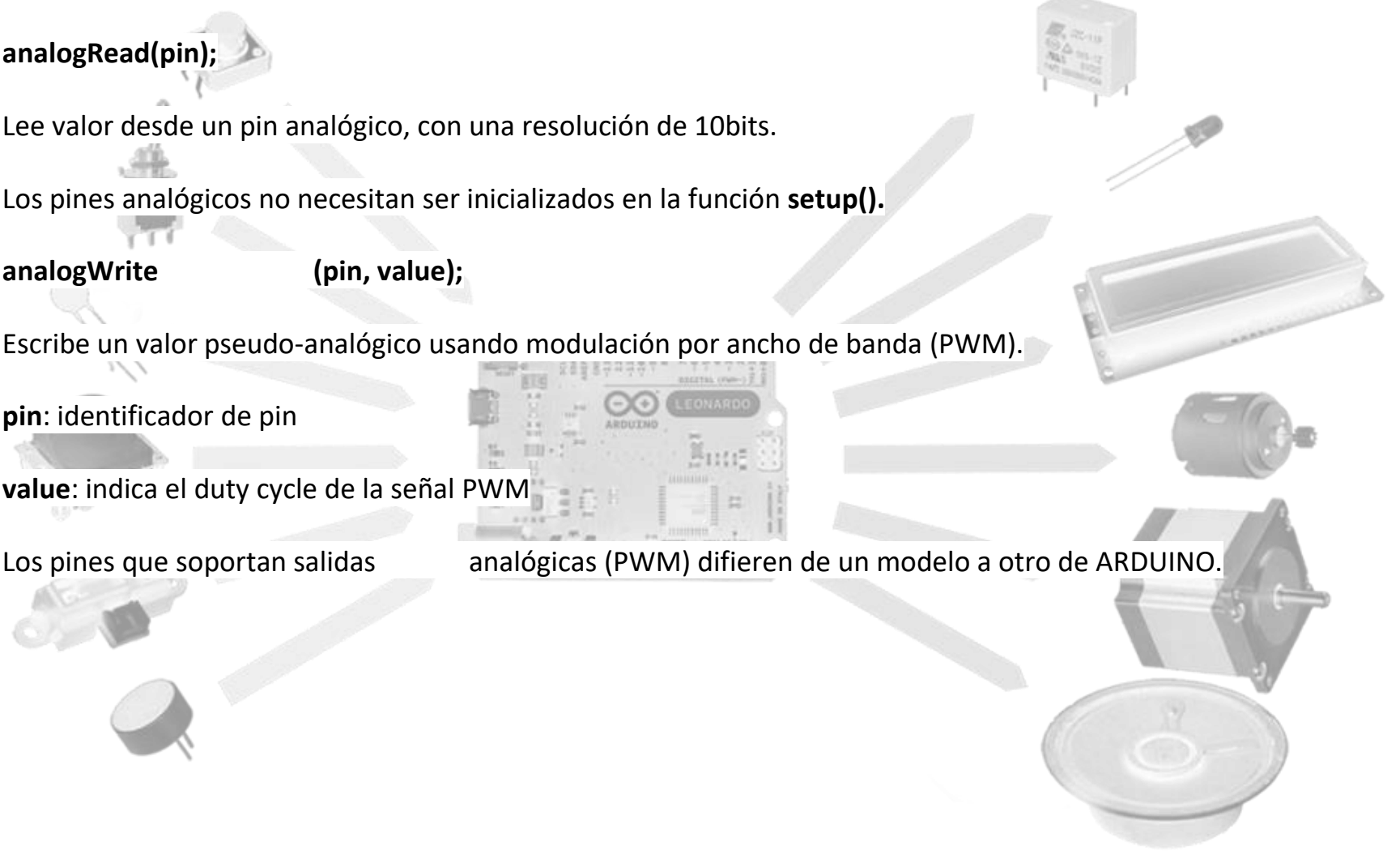
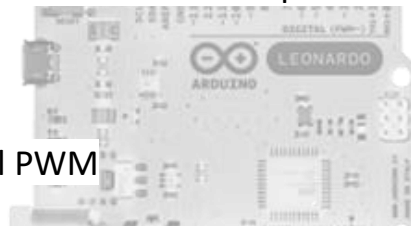
Escribe un valor pseudo-analógico usando modulación por ancho de banda (PWM).

pin: identificador de pin

value: indica el duty cycle de la señal PWM

Los pines que soportan salidas

analógicas (PWM) difieren de un modelo a otro de ARDUINO.



Entradas y salidas Analógicas

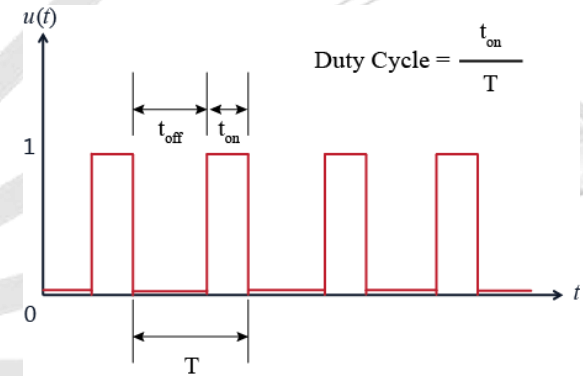
analogWrite (pin, value)

Escribe un valor pseudo-analógico usando modulación por ancho de banda (PWM).

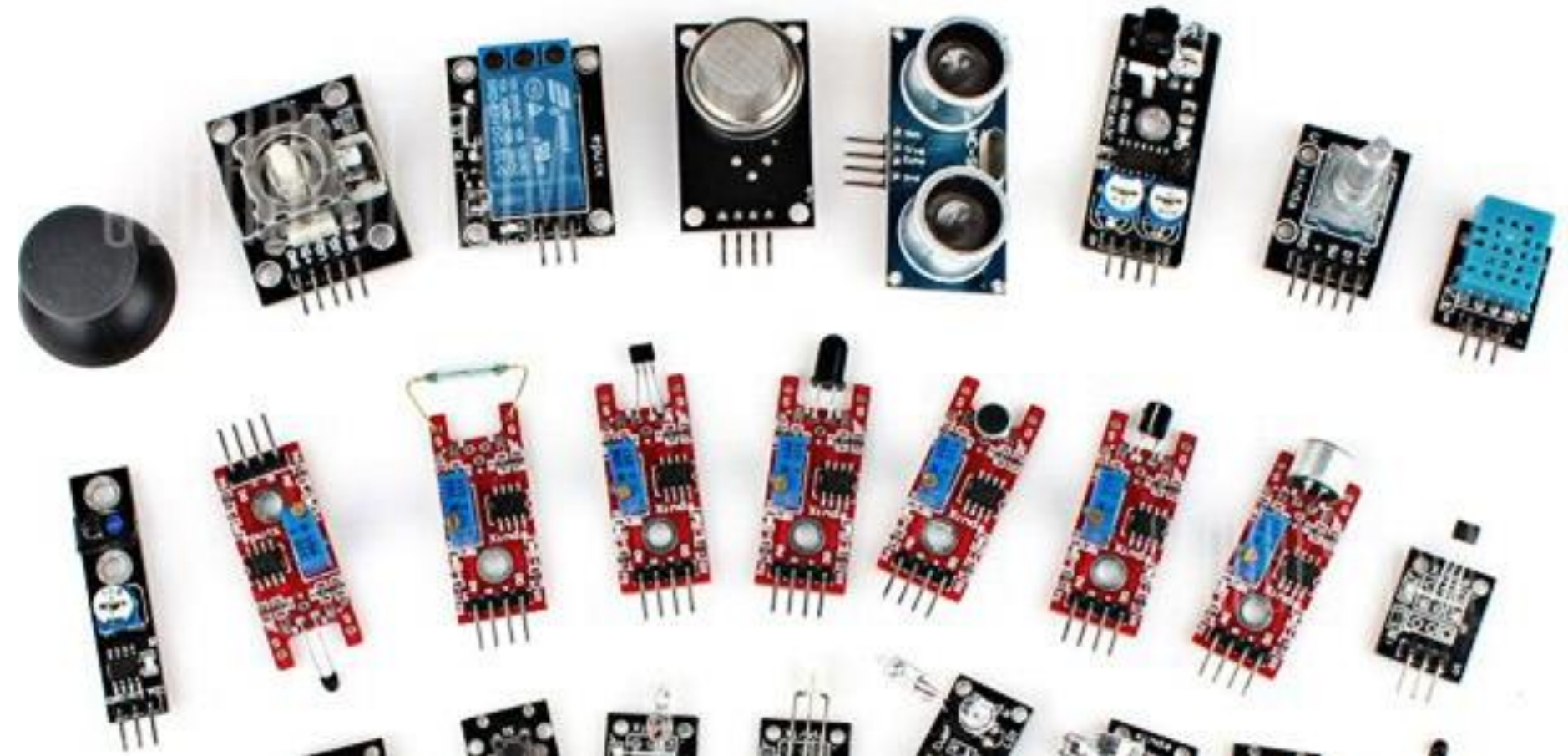
pin: identificador de pin

value: indica el duty cycle de la señal PWM

Los pines que soportan salidas analógicas (PWM) difieren de un modelo a otro de ARDUINO.

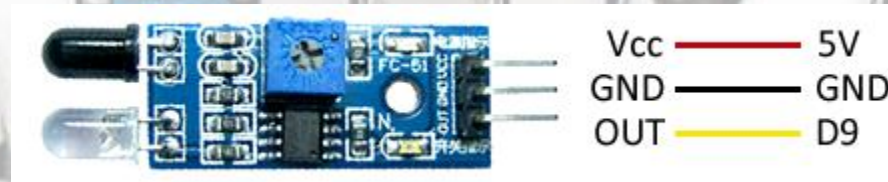


Tipos de Sensores



Tipos de Sensores

Sensor IR



Estos sensores son típicamente implementados en robots móviles con el fin de detectar obstáculos mediante la reflexión que produce en la luz. Está compuesto por un led emisor de luz infra roja y un fotodiodo que generalmente vienen integrados en una placa junto con un comparador LM393, lo cual permite comparar con un valor umbral la distancia al objeto, obteniendo como resultado una señal digital.

Tipos de Sensores

Sensor de ultrasonido



Sabiendo que la velocidad del sonido es 343 m/s en condiciones de temperatura 20 °C, 50% de humedad, presión atmosférica a nivel del mar. En base a la siguiente ecuación podemos concluir que el sonido tarda 29,2 microsegundos en recorrer un centímetro. frecuencia no audibles por el ser humano, que al rebotar en el objeto es reflejado hacia el “parlante” que actúa como receptor. De esta manera podemos obtener el tiempo que se tardó a señal en ir y volver. Ese tiempo lo relacionamos con la velocidad del sonido para así estimar la distancia hasta el objeto.

En lo que concierne al código, deberemos poner un pin digital como salida en el pin Trigger y otro pin digital como entrada para leer el pin Echo

Motores



Ilustraciones extraídas del sitio: <https://www.luisllamas.es/tipos-motores-rotativos-proyectos-arduino/>

Motores

Motor paso a paso

Estos motores son los más implementados en robótica en diversas aplicaciones como por ejemplo CNC. Su principal característica le da el nombre, la posibilidad de controlar su giro por pasos permite que tengan una precisión entono a los 1.8° (200 pasos por vuelta) y 3.75° (96 pasos por vuelta).

Para conseguir el accionamiento de estos motores necesitaremos si o si un driver o controlador para coordinar la energización de las bobinas.



Motores

Motor de corriente continua

Estos motores son los más sencillos de implementar y los podemos encontrar en una enorme cantidad de proyectos en los que no se necesita precisión de giro. A diferencia de los motores paso a paso, estos motores pueden ser accionados directamente por un pin digital de Arduino (no recomendable).

Incluso se puede utilizar el mismo driver de un motor PAP para el accionamiento de dos motores DC.

Para invertir el sentido de giro del motor basta con invertir la polaridad de la conexión del motor.



Motores

Servomotores

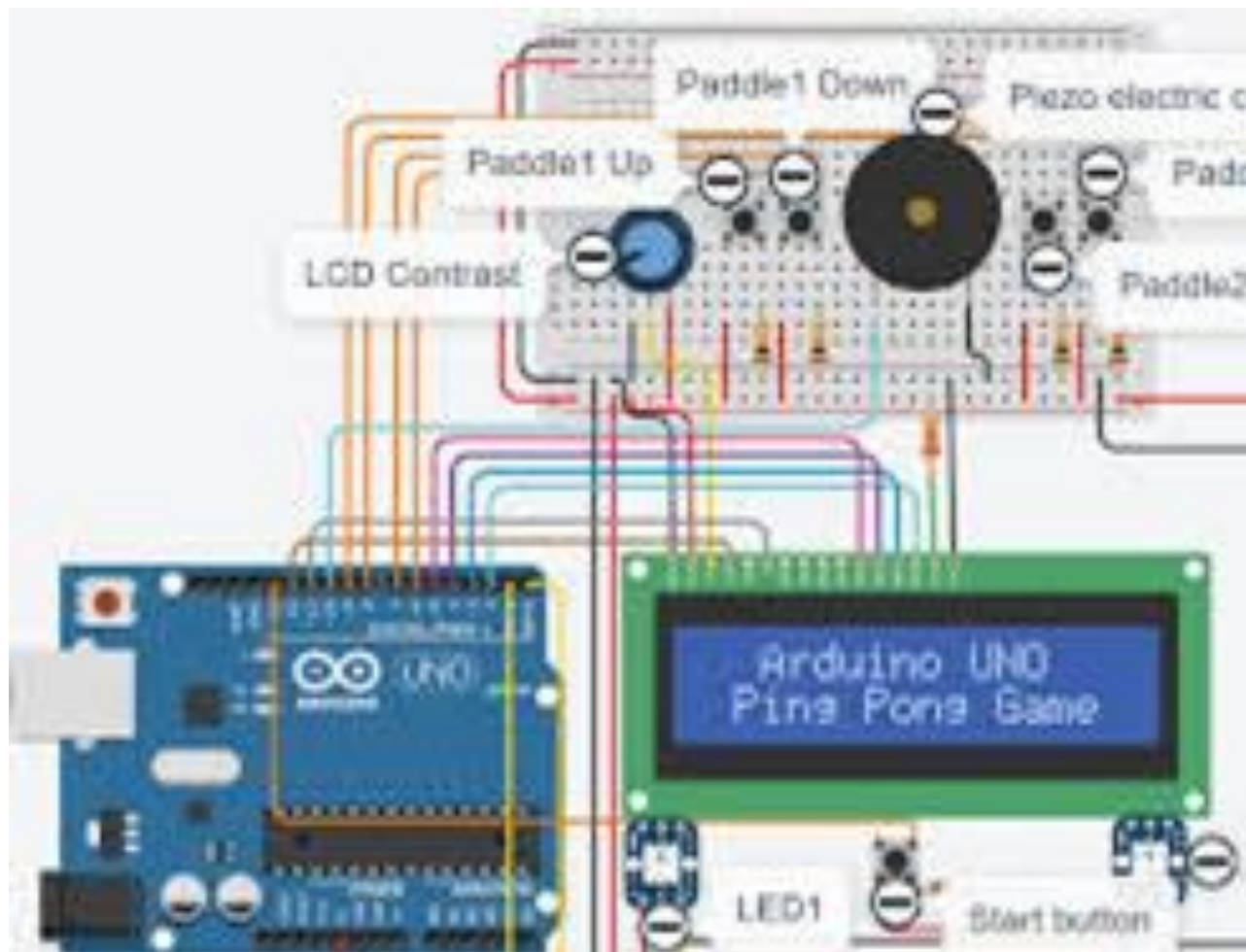
Junto con los motores PAP los servomotores tienen una infinidad de aplicaciones en lo que refiere al mundo de la robótica. Internamente están compuestos por un motor DC acoplado a un reductor y acompañado por un controlador (generalmente potenciómetros) para posicionar al eje en la posición deseada.

Son sencillos de implementar, generalmente responden a un tren de pulsos PWM. Los servos típicos no dan una vuelta completa, su rango es de 0 a 180°. Se caracterizan por tener un alto par y bajas velocidades



Simuladores

Tinkercad



Módulos de comunicación

Comunicación puerto serie



Módulos de comunicación

Comunicación puerto serie

Casi la totalidad de las placas arduino cuentan con al menos una unidad UART (universally asynchronous receiver/transmitter) esta unidad se encarga de realizar la conversión de datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada. En el caso del arduino UNO se cuenta con un conector USB conectado físicamente a los pines tx (pin 1) y rx (pin 0).

El IDE de arduino nos permite enviar y recibir información del Arduino mediante el monitor serial, podemos imprimir en pantalla en tiempo real el valor de las variables o asignarles un nuevo valor fácilmente.

Módulos de comunicación

Comunicación RF

El módulo RF de 433 MHz se popularizó por su bajo costo y facilidad de acceso e implementación. Estos son transmisores/receptores que operan a 433MHz, aunque también existen módulos similares a 315MHz. Ambas frecuencias pertenecen a bandas libres, por lo que su uso es gratuito.

La implementación de estos módulos equivaldría comunicar dos arduinos con conexión de tx a rx y de rx a tx pero sin necesidad de implementar cables.

Desgraciadamente la comunicación entre se da únicamente entre emisor y receptor siendo esta unidireccional.



Módulos de comunicación

Comunicación Bluetooth

Los módulos HC-05 son módulos que podemos usar para conectar arduino a bluetooth, ya sea desde una pc, tablet, teléfono celular o desde un arduino a otro con la configuración maestro esclavo.

Su rango de alcance va de 5 a 10 metros de distancia.

La implementación de estos módulos equivaldría comunicar dos arduinos con conexión de tx a rx y de rx a tx pero sin necesidad de implementar cables como comentamos anteriormente.

Incertidumbre: Que tan fiables pueden ser tratándose de un robót móvil.

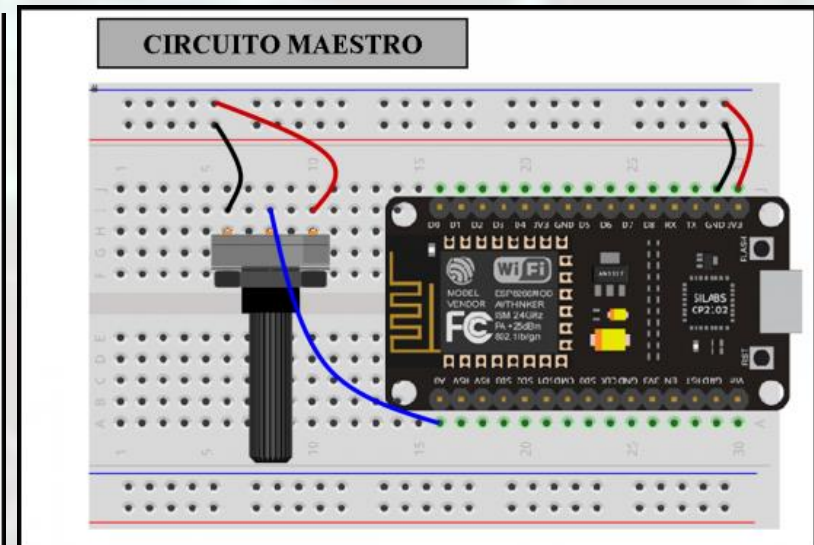
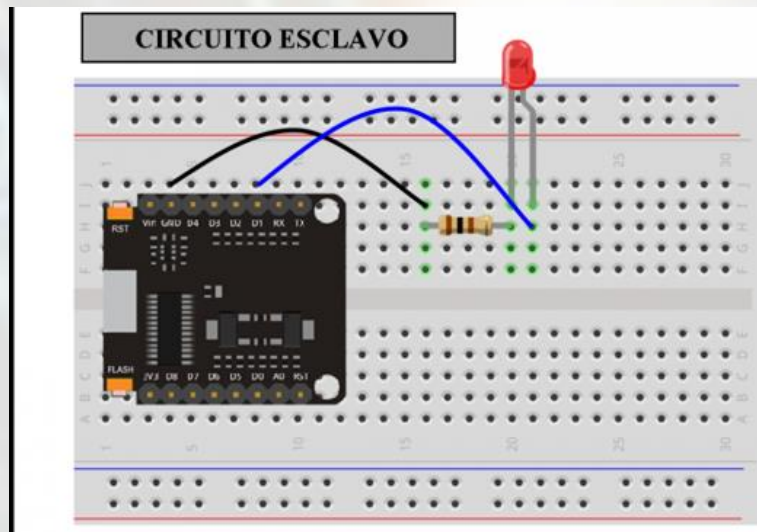


Módulos de comunicación

Comunicación Wifi

La comunicación vía wifi a tomado un papel fundamental en aplicaciones referidas al IOT. Uno de los más populares placas de desarrollo es sin lugar a dudas el NodeMCU. Este es un dispositivo capaz de programarse también en el entorno Arduino, cuenta con entradas y salidas analógicas y digitales tal cual un arduino con la diferencia que se puede conectar a una red wifi o en configuración maestro esclavo.

Existen también los ESP 01 y ESP 8266. Estos son más compactos pero cuentan con pines limitados y son más complejos a la hora de cargar un sketch.



Instaladores

Ide Arduino

<https://www.arduino.cc/en/software>

Tinkercad

<https://www.tinkercad.com/things/flHpOFbJ1CN-arduino-simulator-and>

Documentación

Pines digitales arduino: <http://arduino.cc/en/Tutorial/DigitalPins>

Entradas/salidas: <https://aprendiendoarduino.wordpress.com/2016/12/18/entradas-y-salidas-arduino-2/>

Tipos de Arduino: <https://www.zonamaker.com/arduino/intro-arduino/conociendo-arduino-introduccion>

Sensor de ultrasonido: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>

Sensor IR: <https://www.luisllamas.es/detectar-obstaculos-con-sensor-infrarrojo-y-arduino/>

Puerto serie: <https://www.luisllamas.es/arduino-puerto-serie/>

Comunicación bluetooth: <https://www.luisllamas.es/conectar-arduino-por-bluetooth-con-los-modulos-hc-05-o-hc-06/>

Comunicación Wifi: <http://agrportfolioeducativo.blogspot.com/2020/03/nodemcu-04espnow-comunicacion.html>

Ejemplos

Links...

Muchas gracias!

