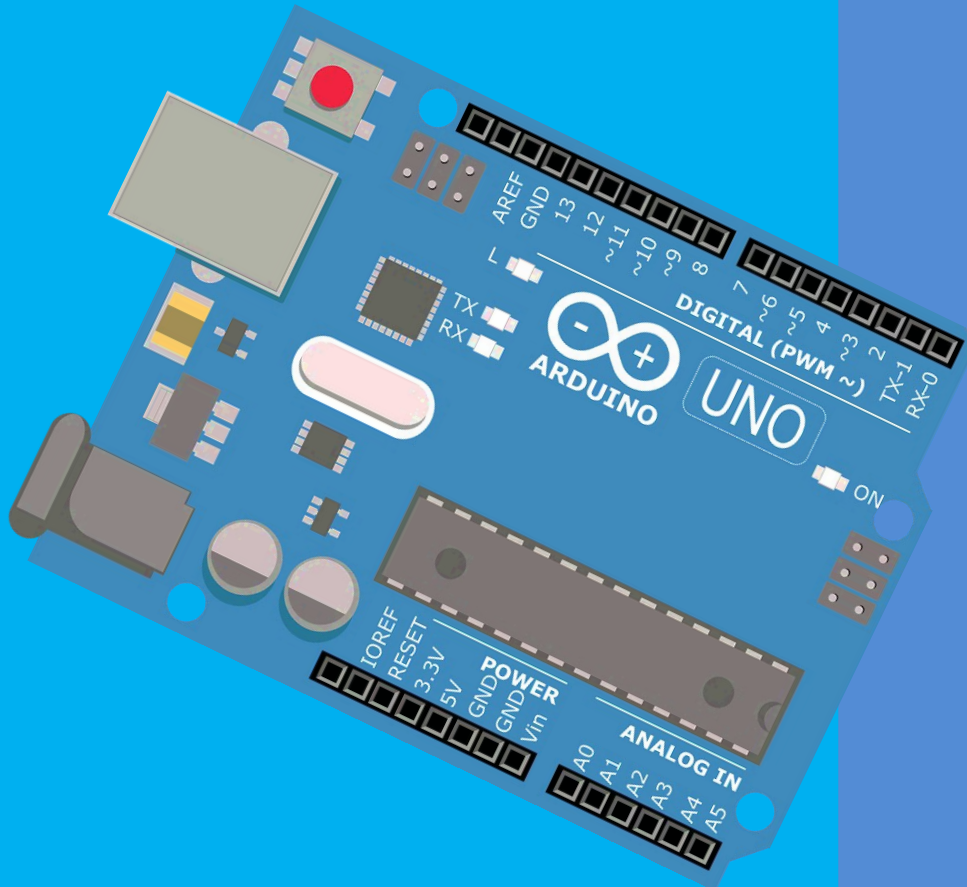


ARDUINO NIVEL 1

¿Cuántas luces puedo conectar al Arduino?

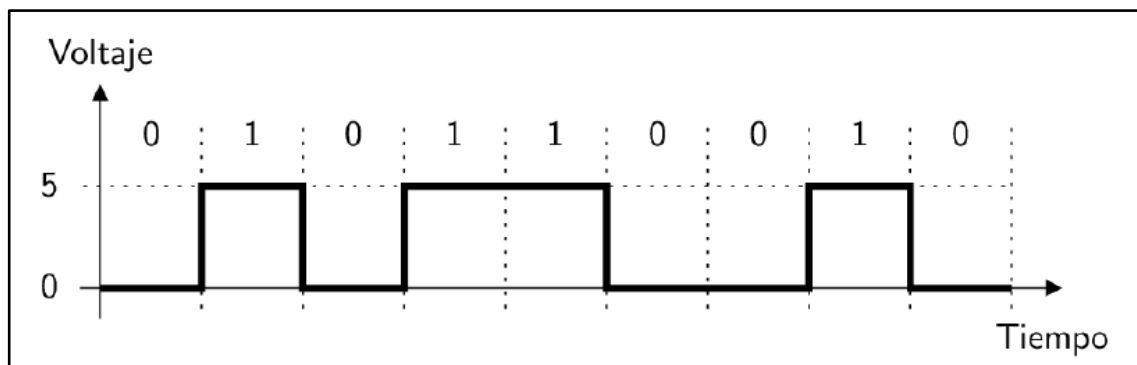
Instructor: Konrad Peschka



¿Cuántas luces puedo conectar al Arduino?

Una de las funciones más interesantes (si no la más) de Arduino y en general de todos los autómatas es su capacidad de interacción con el mundo físico. Podemos, por ejemplo, realizar mediciones de tensión, obtener lecturas de gran variedad de sensores, encender dispositivos o controlar motores y actuadores. Esta interacción se lleva a cabo en gran parte mediante el uso de las entradas y salidas tanto digitales como analógicas.

Una señal digital es una variación de voltaje entre 0 Volt. a 5 Volt. sin pasar por los valores intermedios. Por lo tanto, una señal digital dispone solo de dos estados. Al valor inferior de tensión 0 Volt. le asociamos un valor lógico LOW o '0', mientras que al valor superior 5 Volt. le asociamos HIGH o '1' lógico.

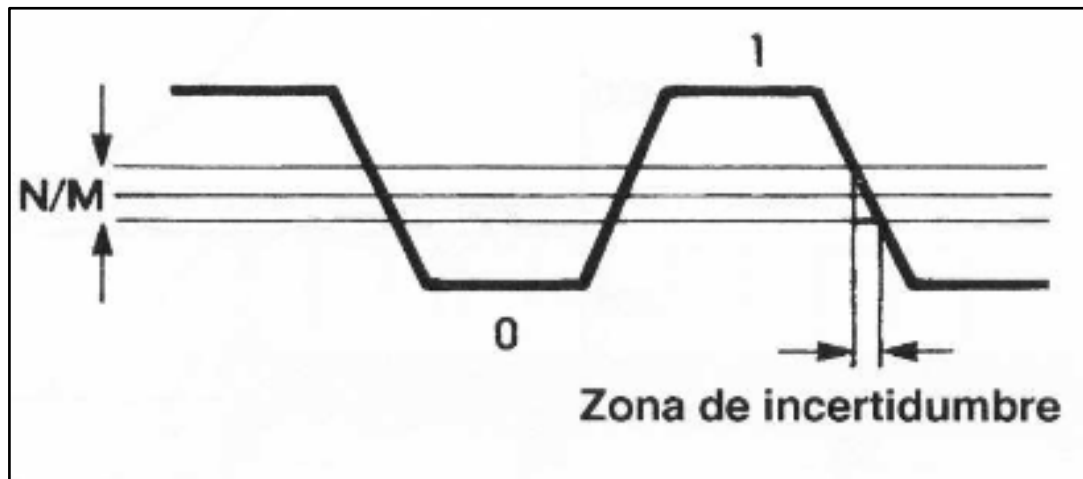


Valores de voltaje para HIGH, LOW

Zona de incertidumbre

La zona de incertidumbre es la zona o región de valores que deben evitarse ya que no garantiza el funcionamiento adecuado del equipo, el rango de incertidumbre es la región donde no se asegura el funcionamiento adecuado del circuito lógico el fabricante especifica que una entrada baja varíe de 0 a 0.8V y un alta varía de 2 a 5V. La región que está comprendida entre 0.8 y 2V se le denomina región prohibida o de incertidumbre y cualquier entrada en este rango daría resultados impredecibles.

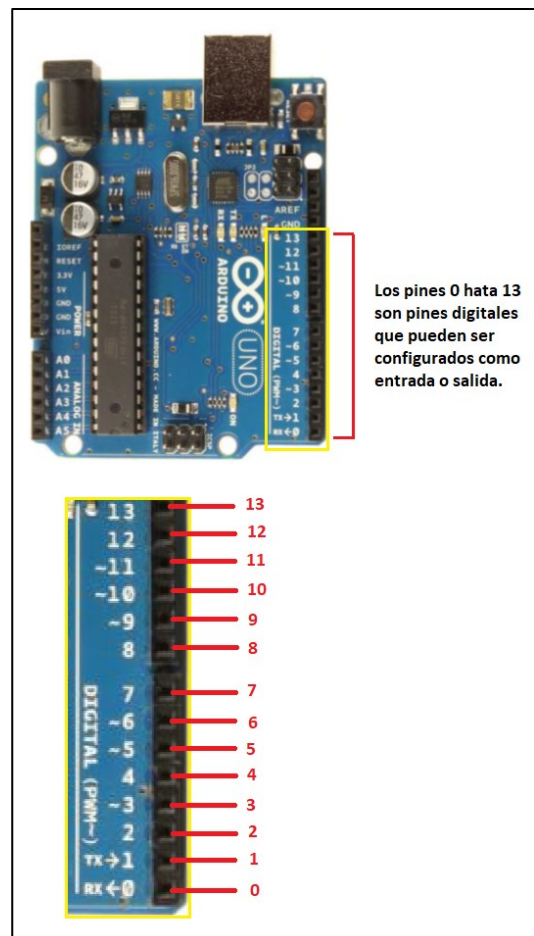
Curso de Arduino nivel 1



Zona de incertidumbre

Puertos de entrada-salida de la placa Arduino

La placa Arduino dispone de 14 pines-hembra de entradas o salidas (según lo que convenga) digitales, numeradas desde la 0 hasta la 13. Adicionalmente dispone de 6 puertos de entrada- salida analógica que según sea el caso se pueden usar como entrada y salida digital, para disponer de un total de 20 puertos de entrada y salida



Puertos digitales

Es importante recordar que no podemos sobrepasar los límites de voltaje permitidos, es decir, si se aplica un voltaje mayor a 5 voltios en una entrada digital, la placa de Arduino se quemará. Lo mismo aplica para voltajes negativos, por lo que debemos asegurar que el voltaje aplicado a una entrada digital se encuentre entre 0 y 5 voltios.

Curso de Arduino nivel 1



Puertos analógicos Arduino

Puertos analógicos usados como puertos digitales

Los pines analógicos pueden ser utilizados de forma idéntica a los pines digitales, usando el alias A0 (para entrada analógica 0), A1, etc. Por ejemplo, el código sería este para establecer un pin analógico 0 a una salida, y para poner en ALTO:

- `pinMode(A0, OUTPUT);`
- `digitalWrite(A0, HIGH);`

digitalRead() funciona en todos los pines. Se redondea el valor analógico recibido. Si `analogRead(A0)` es mayor que o igual a 512, `digitalRead(A0)` será 1, de lo contrario 0.

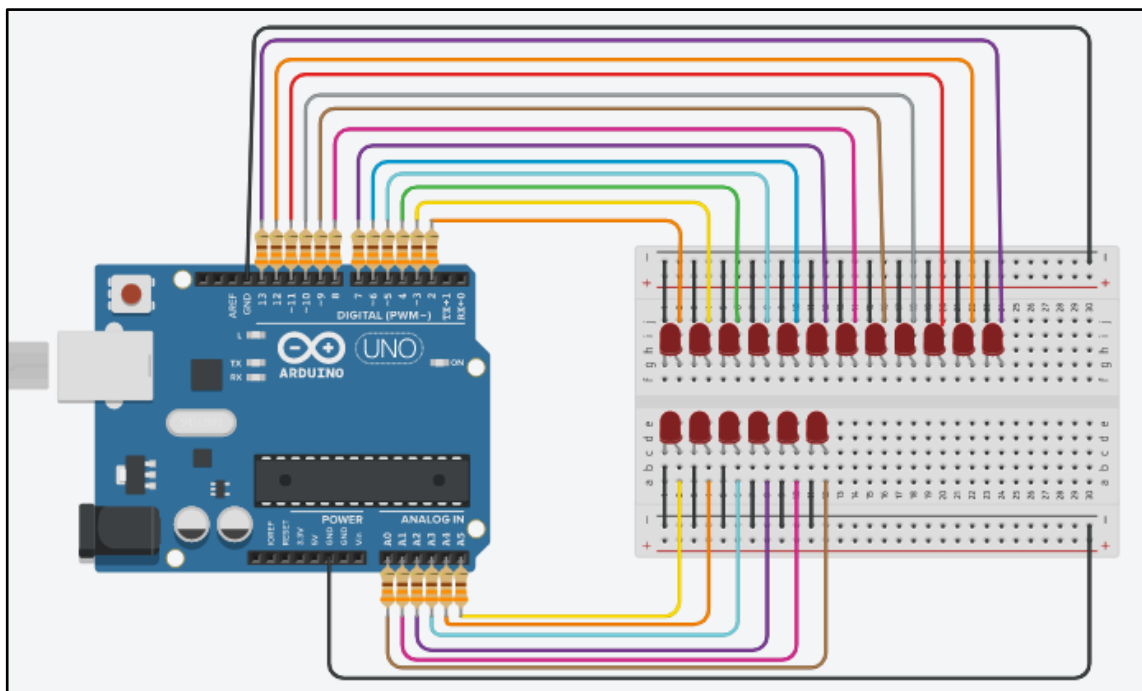
digitalWrite() funciona en todos los pines, que permitían el parámetro de 0 o 1. `digitalWrite(A0,0)` es el mismo que `analogWrite(A0,0)` y `digitalWrite(A0,1)` es el mismo que `analogWrite(A0,255)`

Curso de Arduino nivel 1

Circuito Propuesto

El circuito propuesto consta de un Arduino y 18 Leds que encenderán en secuencia, ocupando de esta manera todos los pines del Arduino, para ello usaremos:

- 1 Arduino uno
- 18 Diodos Leds color rojo
- 18 Resistencia 330Ω
- 1 Protoboard



Circuito: Uso de todos los pines

Curso de Arduino nivel 1

Diagrama de flujo del circuito

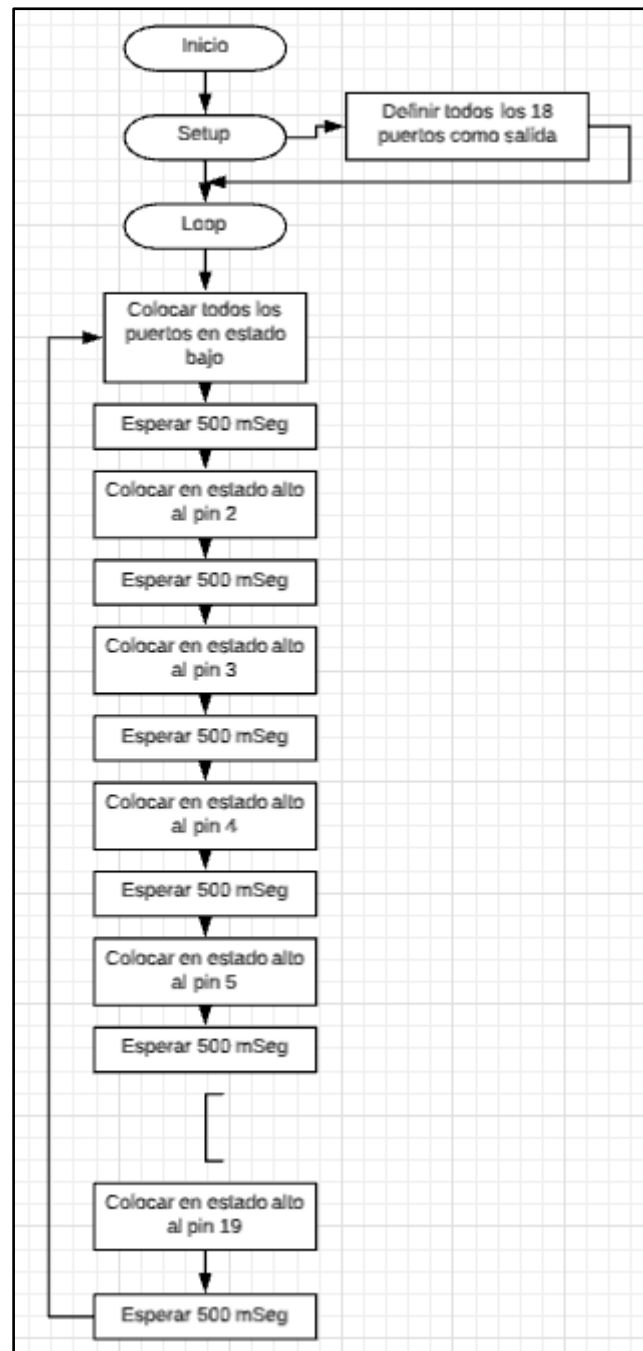


Diagrama de Uso de todos los pines

Curso de Arduino nivel 1

Solución de la Programación

Programa: Uso de todos los pines	
//Ejemplo usando entradas analógicas de Arduino como digitales	
void setup() {	
pinMode(2, OUTPUT);	
pinMode(3, OUTPUT);	//Definir todos los puertos como salida
pinMode(4, OUTPUT);	
pinMode(5, OUTPUT);	
pinMode(6, OUTPUT);	
pinMode(7, OUTPUT);	
pinMode(8, OUTPUT);	
pinMode(9, OUTPUT);	
pinMode(10, OUTPUT);	
pinMode(11, OUTPUT);	
pinMode(12, OUTPUT);	
pinMode(13, OUTPUT);	
pinMode(14, OUTPUT);	//los puertos analógicos son representados
pinMode(15, OUTPUT);	// con el número de puerto del 14-19
pinMode(16, OUTPUT);	
pinMode(17, OUTPUT);	
pinMode(18, OUTPUT);	
pinMode(19, OUTPUT);	
}	
void loop() {	
digitalWrite(2, LOW);	
digitalWrite(3, LOW);	
digitalWrite(4, LOW);	
digitalWrite(5, LOW);	
digitalWrite(6, LOW);	//Colocar todos los puertos en estado bajo
digitalWrite(7, LOW);	
digitalWrite(8, LOW);	
digitalWrite(9, LOW);	
digitalWrite(10, LOW);	
digitalWrite(11, LOW);	
digitalWrite(12, LOW);	
digitalWrite(13, LOW);	
digitalWrite(14, LOW);	
digitalWrite(15, LOW);	
digitalWrite(16, LOW);	
digitalWrite(17, LOW);	
digitalWrite(18, LOW);	
digitalWrite(19, LOW);	
delay (500);	// Esperar 0.5 Seg
digitalWrite(2, HIGH);	// Colocar el pin 2 en estado alto
delay (500);	// Esperar 0.5 Seg
digitalWrite(3, HIGH);	// Colocar el pin 3 en estado alto
delay (500);	// Esperar 0.5 Seg

Curso de Arduino nivel 1

```
digitalWrite(4, HIGH); // Colocar el pin 4 en estado alto
delay(500); // Esperar 0.5 Seg
digitalWrite(5, HIGH); // Colocar el pin 5 en estado alto
delay (500); // Esperar 0.5 Seg
digitalWrite(6, HIGH); // Colocar el pin 6 en estado alto
delay (500);
digitalWrite(7, HIGH);
delay (500);
digitalWrite(8, HIGH);
delay (500);
digitalWrite(9, HIGH);
delay (500);
digitalWrite(10, HIGH);
delay (500);
digitalWrite(11, HIGH);
delay (500);
digitalWrite(12, HIGH);
delay (500);
digitalWrite(13, HIGH);
delay (500);
digitalWrite(14, HIGH);
delay (500);
digitalWrite(15, HIGH);
delay (500);
digitalWrite(16, HIGH);
delay (500);
digitalWrite(17, HIGH);
delay (500);
digitalWrite(18, HIGH);
delay(500);
digitalWrite(19, HIGH); // Colocar el pin 19 en estado alto
delay(500); // Esperar 0.5 Seg
} //Cierre de llaves del Loop
```