UCR | Facultad de Educación

Protea

Programa de Tecnologías Educativas Avanzadas

Bach. Pablo Sanabria Campos

Agenda

- Arduino.
- Entradas y salidas.
- IDE Arduino (instalación y configuración).
- Programación.
- Ejemplos.
- Proyecto.

Arduino

Es una placa de hardware libre que se usa para facilitar la creación de proyectos de electrónica de manera multidisciplinar. Se han fabricado distintos modelos y marcas genéricas.

Entradas y salidas

Entradas: las entradas sirven para lectura de sensores, interruptores o cualquier señal proveniente del exterior de la placa.

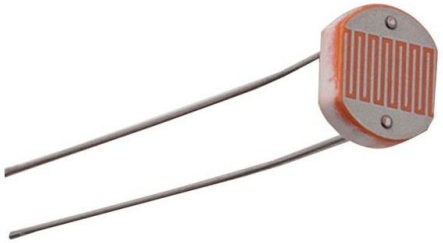
Salidas: las salidas pueden generar tanto señales digitales como analógicas. Se usan para realizar acciones.

Otros pines: también existen pines de reset, ground (0 V), 5 V, 3.3 V y comunicaciones.

Analogía con el ser humano.

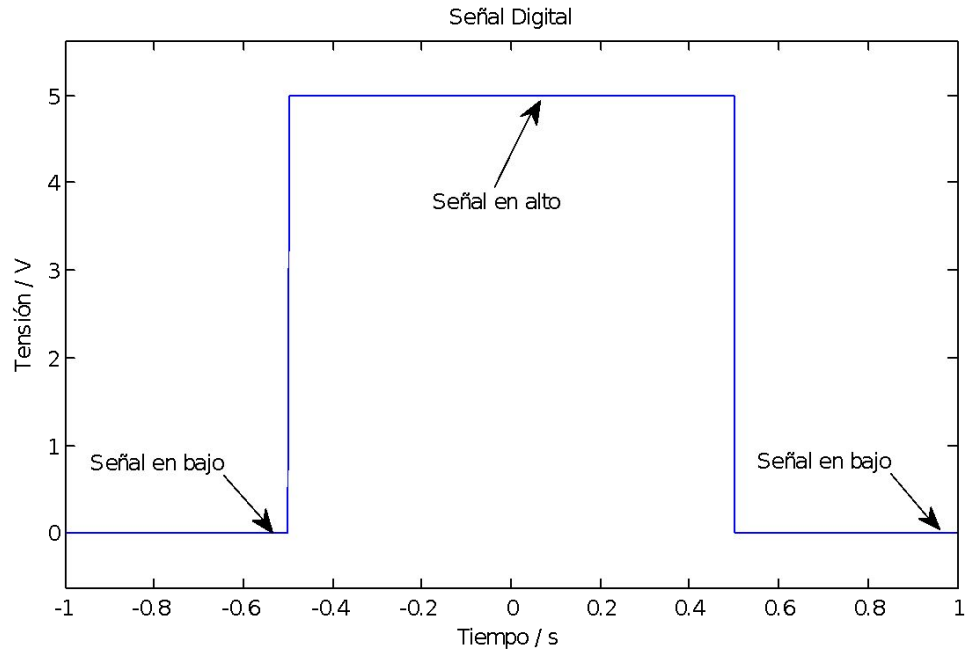
- **Entradas:** el ser humano cuenta con sensores o sentidos que indican temperatura, sonido, sabor, olor y color, entre otros. Estos datos los **recibe** el cerebro (Arduino).
- **Salidas:** el ser humano realiza acciones como el movimiento de músculos o la generación de sonidos. El cerebro **envía** datos a distintas partes del cuerpo.
- En el Arduino, una entrada equivale a un medidor de tensión. Una salida equivale a una fuente de energía.

¿Cuál es un dispositivo de entrada o de salida?

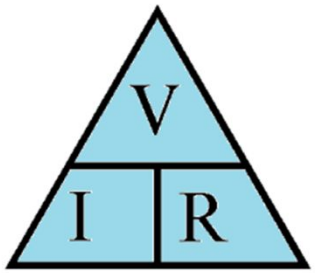


Entrada o Salida Digitales.

- Una entrada digital indica si se está recibiendo una señal en ALTO (5 V) o en BAJO (0 V).
- Una salida digital envía una señal en ALTO (5 V) o (0 V).



Ejemplo de entrada digital.

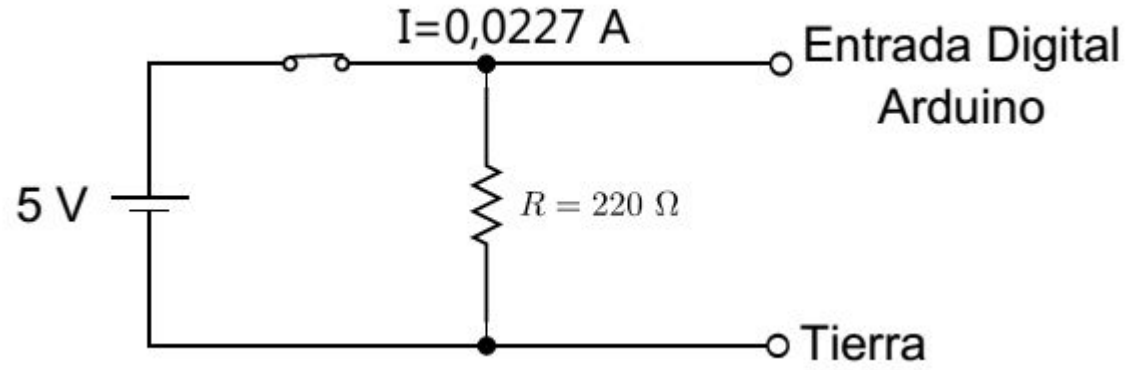


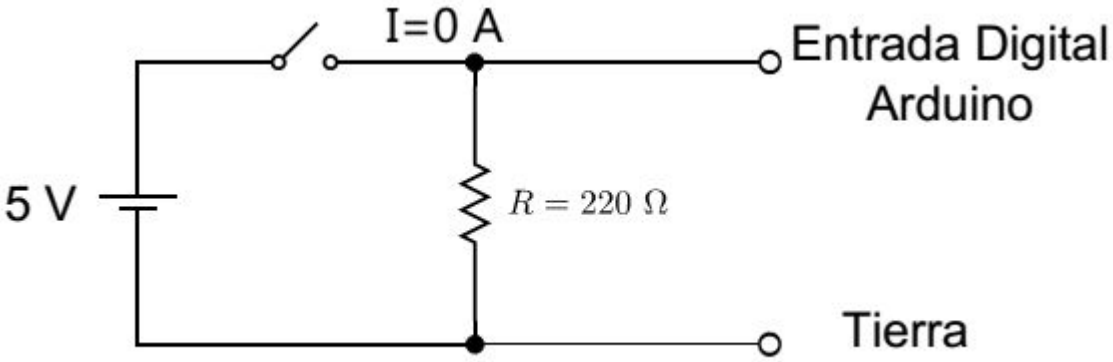
Triangulo Ley de Ohm

V = I x R

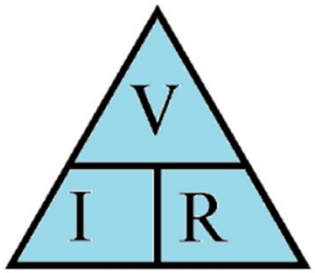
I = V / R

R = V / I





Ejemplo de salida digital.

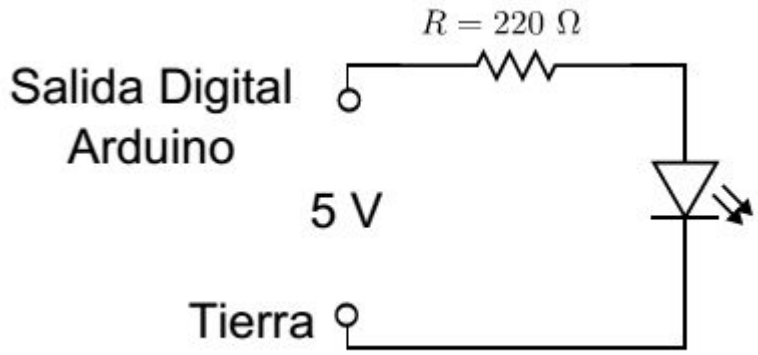


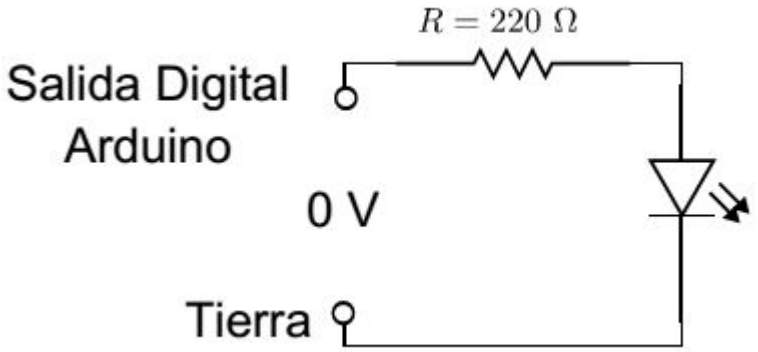
Triangulo Ley de Ohm

V = I x R

I = V / R

R = V / I





Entradas Analógicas.

- El Arduino mide señales que no son constantes. Los valores que mide varían entre 0-5 V. Luego el Arduino realiza una conversión en números de 0 a 1023. Se utiliza la regla de 3.

$$\frac{\text{Valor real o medido}}{5} = \frac{\text{Valor en el monitor}}{1023}$$

Ejemplo #1.

- En un resistor el Arduino mide una tensión de 2,4 V. ¿Cuál es el valor que se muestra en el monitor?

$$\frac{\text{Valor real o medido}}{5} = \frac{\text{Valor en el monitor}}{1023}$$

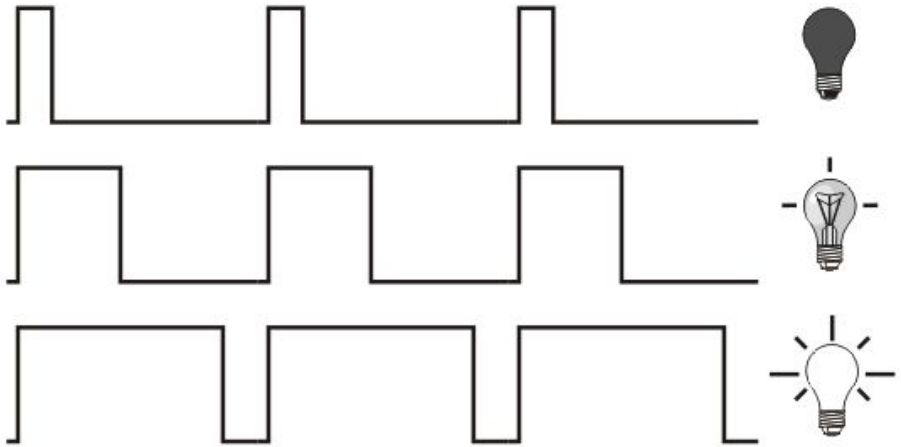
Ejemplo #2.

- En el monitor se muestra un valor de 700. ¿Cuál es el valor real o medido en el Arduino, en Volts?

$$\frac{\text{Valor real o medido}}{5} = \frac{\text{Valor en el monitor}}{1023}$$

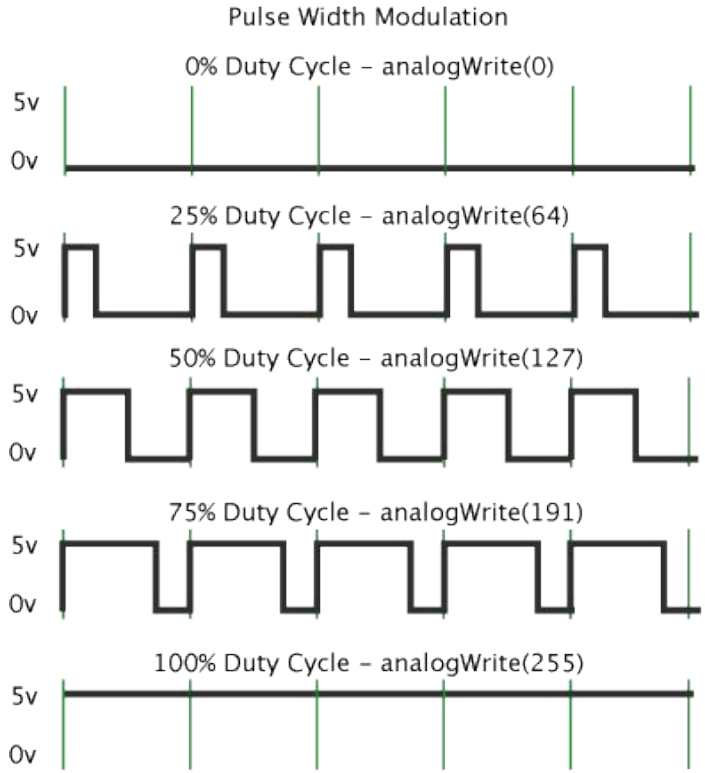
Salidas Analógicas (PWM).

- El Arduino genera una señal con diferentes anchos de pulso D. El usuario puede escoger entre valores de 0 V y 5 V.



Salidas Analógicas (PWM).

- Si D=0%, la tensión media es 0 V. El usuario debe enviar un valor de 0. Si D=100%, la tensión media es 5 V y el usuario debe enviar 255.



Salidas Analógicas.

- Se utiliza la regla de 3. El valor deseado por el usuario es la tensión de salida medio entregada a la carga (LED, servomotor).

$$\frac{\text{Valor deseado por el usuario}}{5} = \frac{\text{Valor en el programa}}{255}$$

Ejemplo #1.

- Se desea enviar una señal de 3,5 V a un LED RGB. ¿Cuál valor se debe enviar en la programación?

$$\frac{\text{Valor deseado por el usuario}}{5} = \frac{\text{Valor en el programa}}{255}$$

Ejemplo #2.

- Un usuario envía una 230 a una salida tipo PWM. ¿Cuál es el valor de salida deseado por el usuario, en Volts?

$$\frac{\text{Valor deseado por el usuario}}{5} = \frac{\text{Valor en el programa}}{255}$$

IDE Arduino

- Ingresar a la página oficial de Arduino.
- Dirigirse a Descarga (Download).
- Escoger el Sistema Operativo.
- Instalar.
- Conectar el Arduino. En la pestaña herramientas indicar: Placa: **nombre de placa. Puerto** al que está conectado.

Programación

- Incluir librerías si es necesario.
- Definir variables, constantes, y nombres para facilitar la programación. Se definen el nombre y número del puerto analógico.
- Definir si las terminales digitales son entrada o salida. Se activa el monitor serial.
- Definir las terminales analógicas o utilizar.
- Ejecutar el programa indefinidamente.

Programación

- Librerías. Son un conjunto de funciones extra para el uso en un programa o sketch. Para incluir un librería se utiliza el comando “#include<nombre librería>” .

```
#include <Servo.h>
.
.
.
```

Definir constantes, variables u objetos

- Se definen números variables de tipo entero con int. Se definen números decimales con float o double, entre otros. Ejemplos:

```
int variableEntera;
int  nombreConstante=13;
float  numeroDecimal;
.
.
.
```

Nota: Después de cada línea de código se termina con el símbolo ;

Definir constantes, variables u objetos.

- Para definir un puerto analógico se utiliza:

```
int sensorPin = A0;
int nombreEntrada=A1;
int analogOutPin = 9;
int nombreSalida = 8;
.
.
.
```

Definir terminales y monitor serial.

- Se debe escribir la función void setup (){}.Se definen las entradas y salidas digitales. Se utiliza el comando pinMode(número de pin, OUPUT o INPUT) para las Entradas/salidas digitales.

```
void setup () {
pinMode(13, OUTPUT);
pinMode(12, INPUT);
}
.
.
.
```

Definir terminales y monitor serial.

- Se inicializa el puerto de comunicación serial Arduino-PC. Se activa el monitor serial con:

```
void setup () {
Serial.begin(9600);
}
.
.
.
```

Definir el lazo indefinidamente.

- Este lazo se repite indefinidamente. Es la parte importante de la programación. Aquí se indica lo que se quiere lograr con el proyecto.

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

Definir el lazo indefinidamente.

- Se pueden enviar señales analógicas al pin de salida con analogWrite(pin de salida,valor). Se pueden leer datos de sensores mediante analogRead(pin de entrada).

```
int analogOutPin = 9;
int analogInPin = A0;

void loop () {
analog Write(analogOutPin, valor entre 0-255);
analogRead(analogInPin);
.
.
.
}
```

Definir el lazo indefinidamente.

- Se pueden enviar señales analógicas al pin de salida con analogWrite(pin de salida,valor). Se pueden leer datos de sensores mediante analogRead(pin de entrada).

```
int analogOutPin = 9;
int analogInPin = A0;

void loop () {
analog Write(analogOutPin, valor entre 0-255);
analogRead(analogInPin);
.
.
.
}
```

Definir el lazo indefinidamente.

- Se pueden enviar señales digitales al pin de salida con digitalWrite(pin de salida, HIGH o LOW). Se pueden leer datos de sensores mediante digitalRead(pin de entrada).

```
void setup () {
pinMode(13, OUTPUT);
pinMode(12, INPUT);
}

void loop () {
digitalWrite(13, HIGH);
digitalWrite(13, LOW);
digitalRead(12);

}
```

Definir el lazo indefinidamente.

- Mediante el comando delay(), se espera “n” cantidad de milisegundos y luego realiza la siguiente acción. 1000 milisegundos= 1 segundo.

```
void loop () {
delay(1000);
.
.
.
}
```

Definir el lazo indefinidamente.

- Mediante el comando Serial.println() se muestra un valor de entrada o salida y se muestra en el monitor serial del IDE de Arduino.

```
int analogInPin = A0;
int valorEntrada;

void loop () {
valorEntrada=analogRead(analogInPin);
Serial.println(valorEntrada);
.
.
.
}
```

Encender y apagar un LED cada segundo.

- Se deben definir las instrucciones CLARAMENTE.

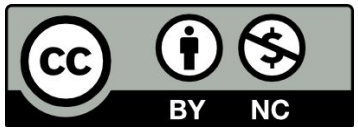
1. No se ocupan librerías.
2. Se necesita un nombre para el pin. ¿Cuál número de pin se va a utilizar?
3. ¿El pin es de entrada o salida?
4. El LED debe encenderse, 1 segundo después se apaga. Se mantiene apagado un segundo y se repite la secuencia indefinidamente.

Proyecto

- Realizar un contador de 0 a 9. Se deben poner resistores de 450 Ω. Utilizar el NTE3059. Buscar la hoja de datos en Google.

Importante.

- Para proyectos más complejos es necesario conocer los comandos de control.
- If (if-else) (Si, Si no)
- For (para)
- Switch (interruptor)
- While (mientras)


BY
NC