



Arduino Inicial

Ing. Juan C. Abdala

Clase 2



Objetivo de esta clase

- Aprender más sobre los puertos de arduino.
- Aprender qué es y cómo usar una interrupción en Arduino.
- Aprender a usar el ADC en arduino.
- Aprender que es un divisor resistivo y para que se usa.



Los puertos de Arduino.

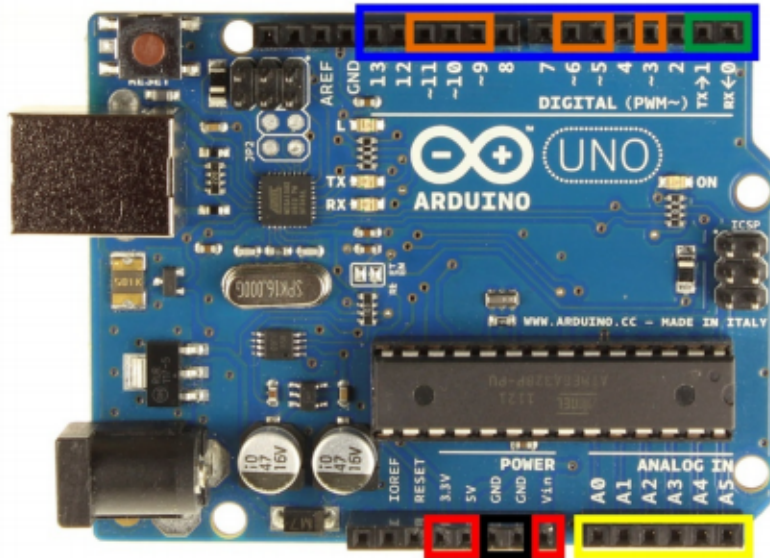


Diagrama de pines de un microcontrolador con colores y funciones:

- GND
- ALIMENTACIÓN
- ENTRADAS ANALÓGICAS
- E/S DIGITALES
- PWM
- PUERTO SERIE



Configurando salida

```
//Configuracion  
pinMode(pin, OUTPUT);
```

```
//Ejecución  
digitalWrite(pin, HIGH/LOW);
```



Configurando Entrada

```
//Configuracion  
pinMode(pin, INPUT);
```

```
//Lectura
```

```
var = digitalRead(pin);    //Lee un puerto, devuelve alto o bajo.
```



Como lo uso

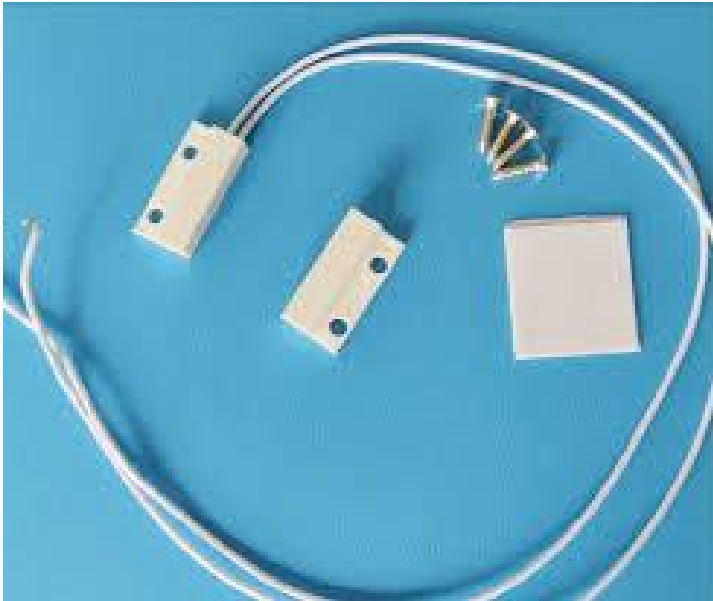
```
var = digitalRead(pin);
```

```
//Consulto si var es alto o bajo  
if (var == 0)  
{  
    Serial.println("Es Bajo");  
}
```

```
//Consulto si var es alto o bajo  
if (var == 1)  
{  
    Serial.println("Es Alto");  
}
```

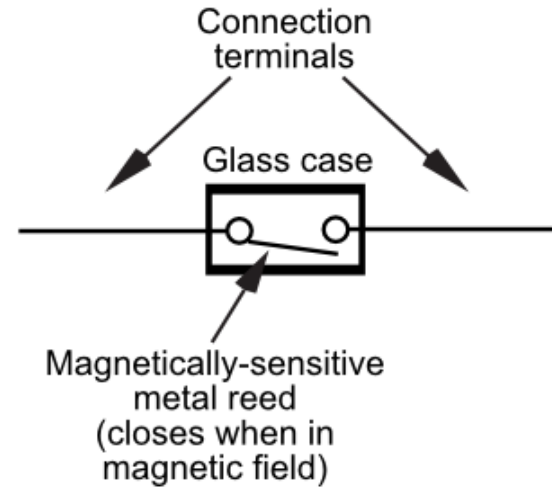
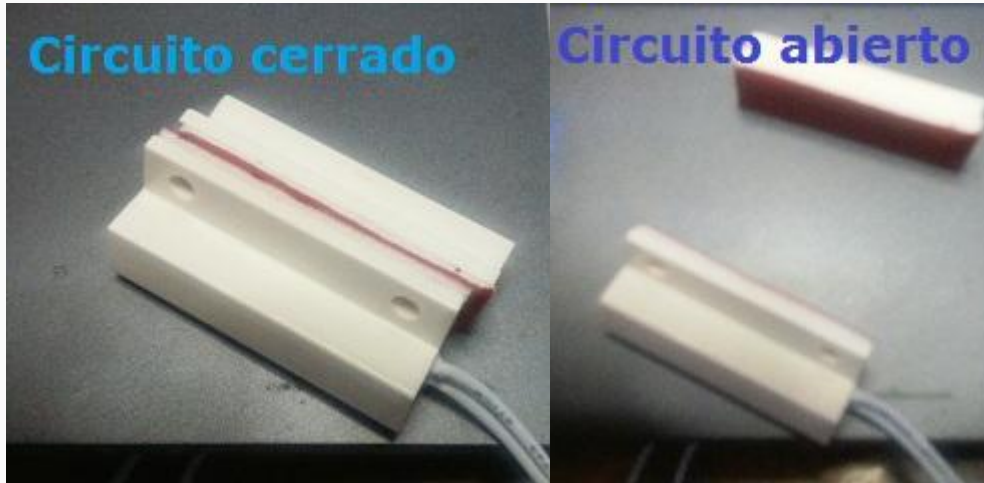


Sensor Magnetico

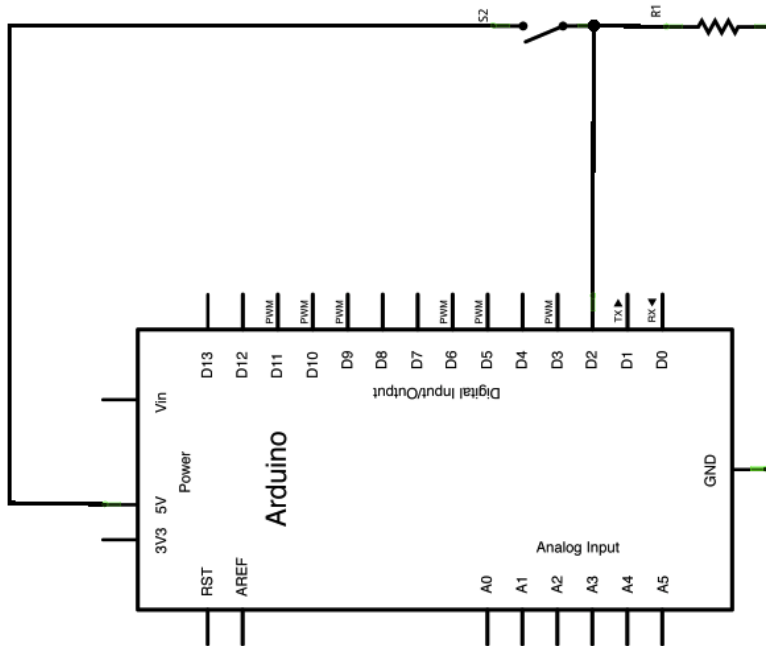


El sensor cierra el circuito cuando se encuentran juntas las 2 partes.

Sensor Magnetico



Sensor Magnetico



“La corriente es vaga”.
Cuando la llave está abierta va a ser un bajo. (pulldown). Cuando se cierre va a ser un alto”.

Leyendo un puerto

```
int val = 0; // variable donde guardo lo que leo del
```

```
int puerto = 2
```

```
void setup()
```

```
{  
  Serial.begin(9600);  
  pinMode(puerto, INPUT);  
}
```

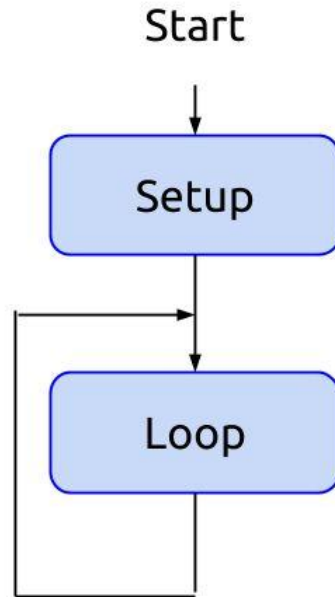
```
void loop()
```

```
{  
  // lee el valor de entrada  
  val = digitalRead(puerto);  
  
  // si la puerta esta abierta  
  if(val == 0) {Serial.println("Puerta Abierta"); }  
  
  //Si la puerta esta cerrada  
  else {Serial.println("Puerta Cerrada"); }  
  
  delay(1000);  
}
```

Interrupciones



Ciclo de Ejecución de Normal Arduino



Una interrupción:

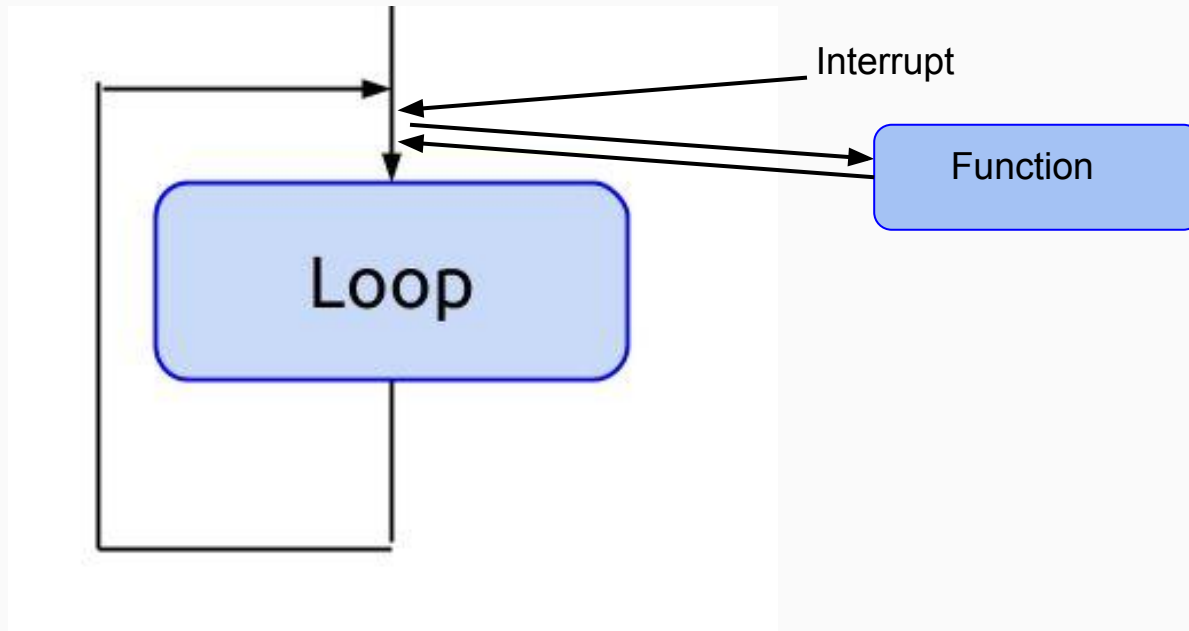
Es una señal recibida por el procesador de una computadora, que le indica que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico.

Interrupciones

Puede ocurrir por:

- Un evento hardware, previamente definido.
- Un evento programado, o Timer
- Una llamada por software.

Ciclo de Ejecución de Arduino con interrup.



Interrupciones en Arduino Uno.

Solo por pin 2 y 3 se llaman 0 y 1 respectivamente.

Libreria integrada **AttachInterrupt**.

Interrupciones en Arduino

Para definir una interrupción necesitamos tres cosas:

1. Un pin de Arduino que recibirá la señal de disparo.
2. Una condición de disparo.
3. Una función que se ejecutará, cuando se dispara la interrupción (Llamada call back function).

Interrupciones en arduino.

En cuanto a la condición de disparo, los modos pueden ser:

- LOW, el pin es Bajo.
- CHANGE, cuando pasa de Alto a Bajo o viceversa.
- RISING, Cuando pasa de Bajo a Alto.
- FALLING, Cuando pasa de Alto a Bajo.

Interrupciones en arduino

Función:

attachInterrupt(numero_de_interrucion[0/1], Función, modo);

Nuestro Ejemplo

- Vamos a realizar una interrupción externa.
- Con un sensor magnetico.
- Basado en un esquemático.

Ejemplo

```
void puerta()
{
    delay(100);
    int val = digitalRead(2); // lee el valor de entrada
    // si la puerta esta abierta
    if(val == 0) {
        Serial.println("Puerta Abierta"); }
    //Si la puerta esta cerrada
    else{
        Serial.println("Puerta Cerrada");
    }
    delay(100);
}
```

```
void setup()
{
    Serial.begin(115200);
    pinMode(2, INPUT);
    attachInterrupt(0, puerta,CHANGE);
}

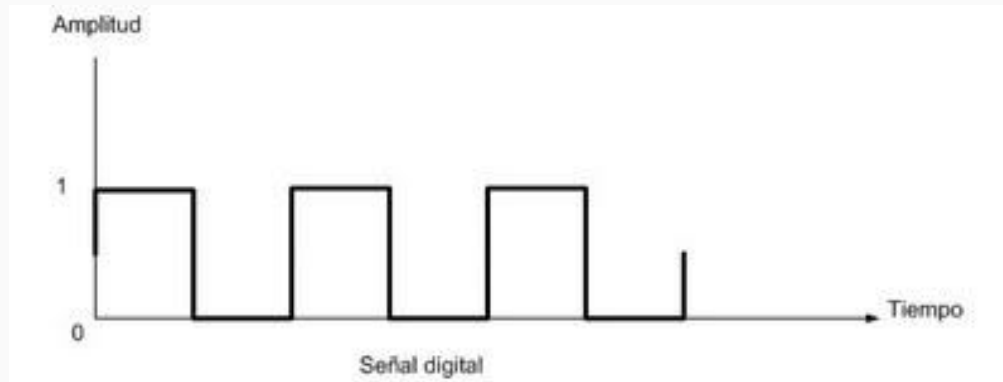
void loop()
{
    Serial.println("No pasa nada");
    delay(1000);
}
```

Conversor Analógico-Digital



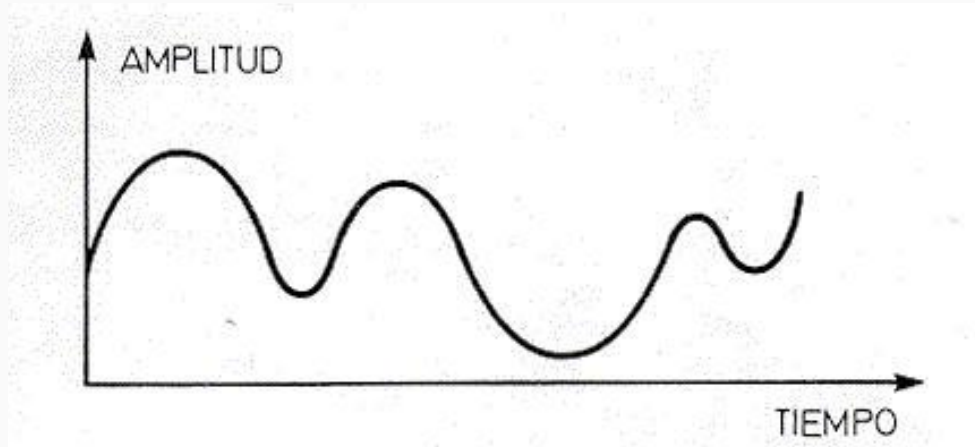
Señal Digital

Usan la lógica de dos estados representados por dos niveles de **Tensión (DC)** eléctrica, uno alto, H y otro bajo, L (de High y Low).



Señal Analógica

Una señal analógica es continua, y puede tomar infinitos valores de **Tensión (DC)**.



Tipo de sensores.

Analogicos: Estos reciben alimentacion y devuelven valores en mV(mili Volt) que deben ser traducidos.

Digitales: Nos devuelven el valor procesado solo debemos tomarlo.

Conversor A/D

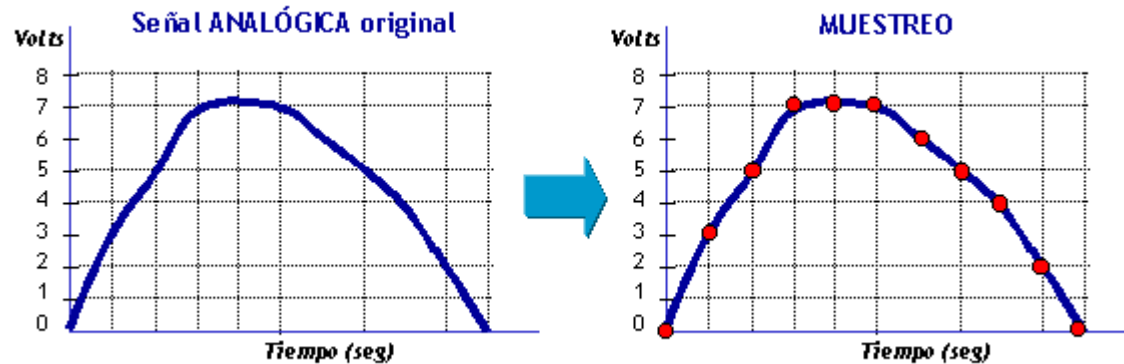
- Muchas veces voy a necesitar transformar una señal analógica.
- Arduino posee, 5 entradas.
- Se denotan en la placa con la letra “A”, ej A1

Conversor A/D

Para realizar esa tarea, el conversor ADC (Analog-to-Digital Converter - Conversor Analógico Digital) tiene que efectuar los siguientes procesos:

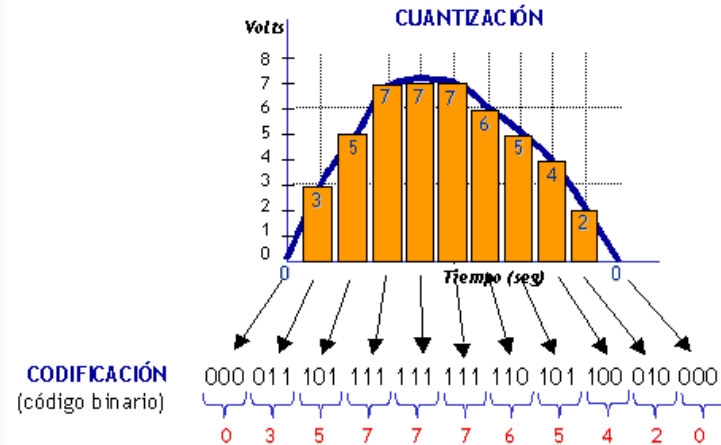
- 1.- Muestreo de la señal analógica.
- 2.- Cuantización de la propia señal
- 3.- Codificación del resultado de la cuantización, en código binario.

Muestreo



Consiste en tomar valores discretos de tensión o voltaje a intervalos regulares en diferentes puntos de la onda senoidal.

Cuantización y Codificación



Cuantización de la señal eléctrica analógica para su conversión en señal digital.

Codificación permite asignarle valores numéricos binarios equivalentes a los valores de tensiones que conforman la señal eléctrica analógica original.



Conversor A/D en Arduino

- El modulo ADC lee voltages entre 0 y 5 volts.
- Posee una resolución de 10 bits (1024 valores).
- Posee una resolución de $5/1024 = 4.8$ mv
- La funcion que lee el pin analogico entrega un valor decimal entre 0 y 1023.
- Debo aplicar la funcion de conversion para obtener voltage.
$$\text{voltage} = \text{dato_sensado} * (\text{voltage_operación}/\text{resolución})$$



Conversor A/D en Arduino

- OJO: Con el modo normal de funcionamiento podemos colocar un valor mayor al soportado por el micro, nuestro caso 5v!!!
- Función:

```
int valor_sensor = analogRead("Ax");
```

```
float voltaje = valor_sensor * (5.0 / 1024.0);
```


Divisor Resistivo o de Tension

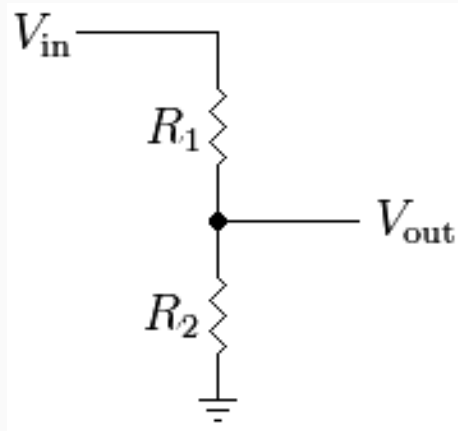
Divisor de Tension.

- Un divisor de tensión es una configuración de circuito eléctrico que reparte la tensión de una fuente entre una o más impedancias conectadas en serie.
- La impedancia (Z) es la medida de oposición que presenta un circuito a una corriente cuando se aplica una tensión.
- Cuando un circuito es alimentado con corriente continua (CC), su impedancia es igual a la resistencia



Divisor de Tension

Se define como:



$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in}$$

Divisor de Tension - Usos

- Adaptación de Voltages superiores al permitido por el ADC. (Permite medir baterías).
- Para el caso de usar 2 resistencias iguales $V_o = (V_i/2)$.
- Polarización de Transistores (No nos interesa).



Practica conversor A/D

- Ahora armemos el circuito.
- Conectemoslo a un pin analogico.
- Imprimamos y llamemos a la función loop cada un segundo.

Descansemos 15 min

15:00



Practico

Vamos al Practico 2:

Material: <https://github.com/jcabdala/ArduinolnicialUNC>



Consultas:



abdalajc@gmail.com



@toniabdala

Google Group: arduino-unc-septiembre@googlegroups.com

