APLICACIONES DE IoT







UNIDAD I.ADQUISICIÓN Y PROCESAMIENTO DE DATOS

El alumno manipulará sensores y actuadores conectados a hardware abierto para procesar y almacenar datos.



1.2 SENSORES DIGITALES

DSM ENERO 2023 UTM

UNIDAD I





AGENDA

- Introducción / Conceptos
- Sensores
- Conexiones en Arduino
 - Digital input
- Sensores digitales
- Medir la variable del sensor digital en Arduino
- Conclusión





INTRODUCCIÓN

- Una señal digital es un tipo de señal generada por algún tipo de fenómeno electromagnético en que cada signo que codifica el contenido de la misma puede ser analizado en término de algunas magnitudes que representan valores discretos, en lugar de valores dentro de un cierto rango.
- Por ejemplo, el interruptor de la luz sólo puede tomar dos valores o estados: abierto o cerrado, o la misma lámpara: encendida o apagada.



INTRODUCCIÓN...

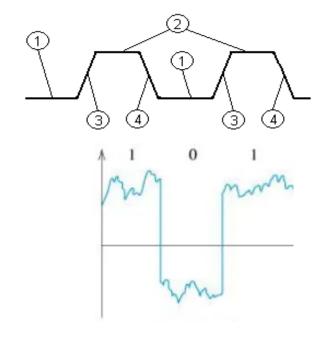
- Los sistemas digitales, como por ejemplo un microcontrolador, usan la lógica de dos estados representados por dos niveles de tensión eléctrica, uno alto, H y otro bajo, L (de High y Low, respectivamente, en inglés).
- Por abstracción, dichos estados se sustituyen por ceros y unos, lo que facilita la aplicación de la lógica y la aritmética binaria.
- 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.





FLANCO DE SUBIDA Y BAJADA

- Además de los niveles, en una señal digital están las transiciones de alto a bajo y de bajo a alto, denominadas flanco de bajada y de subida, respectivamente.
- En una señal digital, se denomina flanco a la transición del nivel bajo al alto (flanco de subida) o del nivel alto al bajo (flanco de bajada).



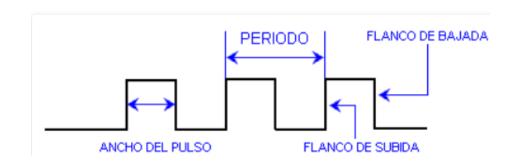


FLANCOS VS PULSOS

- La señal básica es una onda cuadrada (pulsos) y las representaciones se realizan en el dominio del tiempo. Sus parámetros son:
 - Altura de pulso (nivel eléctrico)
 - Duración (ancho de pulso)
 - > Frecuencia de repetición (velocidad pulsos por segundo)



Periodo, ancho de pulso, flancos







TIPOS DE SEÑALES DIGITALES

- Discreta: puede tomar un conjunto de valores
- Binaria: Encendido (1) Apagado (0)





FUNCIONES DE CÓDIGO ARDUINO

- En arduino para tratar las entradas y salidas digitales usamos las siguientes funciones:
 - pinMode() configura en el pin especificado si se va a comportar como una entrada o una salida. http://arduino.cc/en/Reference/PinMode
 - digitalWrite() Escribe un valor HIGH o LOW en el pin digital especificado. Si el pin está configurado como OUTPUT pone el voltaje correspondiente en el pin seleccionado. Si el pin está configurado como INPUT habilita o deshabilita la resistencia interna del correspondiente pin. http://arduino.cc/en/Reference/DigitalWrite
 - b digitalRead() lee el valor del pin correspondiente como HIGH o LOW. http://arduino.cc/en/Reference/DigitalRead

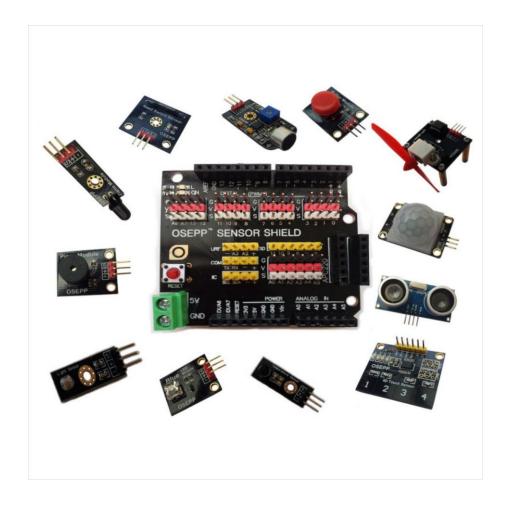


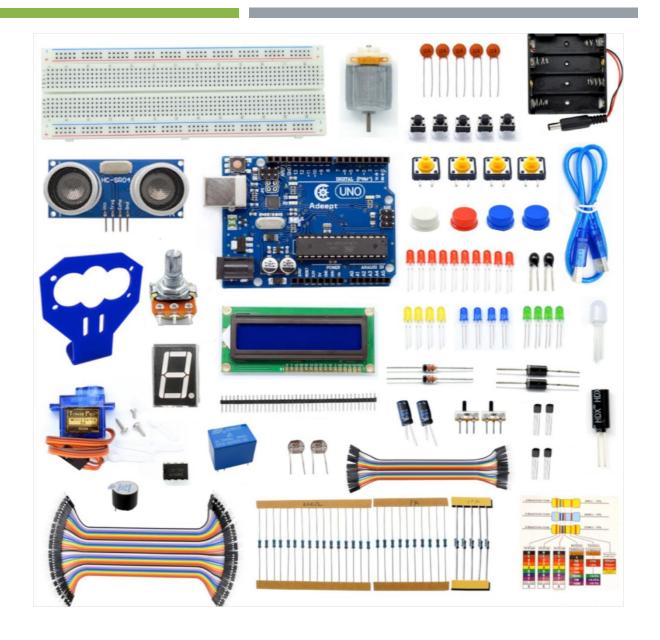
SENSORES

- Definición.- Aparato que detecta magnitudes físicas y químicas.
- Los sensores nos permiten relacionar el mundo externo "real y analógico" con el mundo "virtual y digital" de la computadora.
- De esa manera diversas características y sucesos del espacio exterior a la máquina como son temperatura, movimiento, presencia, luz, etc. - pueden intervenir en procesos numéricos.



SENSORES..





CONEXIONES EN ARDUINO

Existen 2 tipos de conexiones (pines) en Arduino:

- I. Conexiones de entrada
- 2. Conexiones de salida





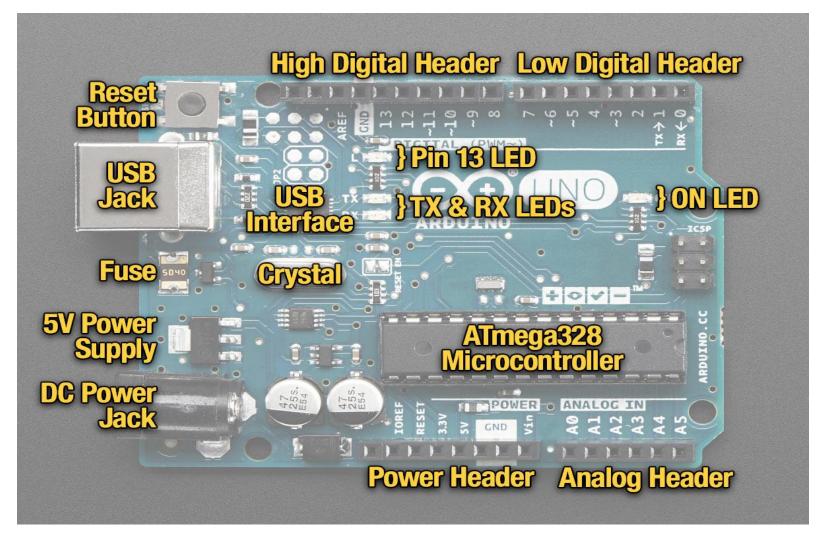
CONEXIONES EN ARDUINO...

- Conexiones de entrada. A través de sensores conectados en los pines de entrada, Arduino recibe datos del exterior (entorno).
- Conexiones de salida. Dependiendo del proyecto en el que esté trabajando, y en función de las instrucciones que le hayamos dado programando el microcontrolador, Arduino puede conectarse con diversos actuadores (relés, pantallas, motores,...), y sistemas lógicos (otras placas, computadoras,...) para provocar la respuesta que necesitamos.
- Ver https://www.arduino.cc/en/Tutorial/Foundations





CONEXIONES EN ARDUINO...



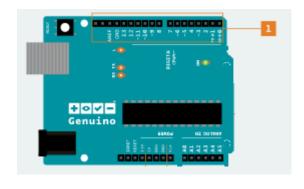
DIGITAL INPUT

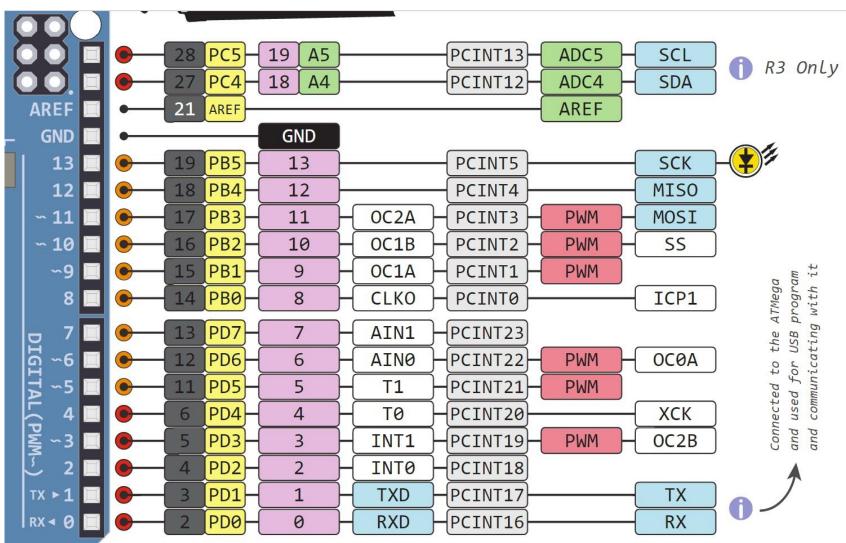
- En Arduino los pins pueden ser configurados como "inputs" o como "outputs"
- El Arduino Uno tiene 14 pins I/O digitales que operan a 5V, 6 de estos tienen salida (PWM, Pulse-Width Modulation). Cada pin puede dar y recibir un máximo de 40mA. Algunos pins tienen características especiales:
 - Serial: 0 (RX) y I (TX). Estos pins se usan para la comunicación serie. Están conectados al chip ATmega 16u2 para la conversión USB a TTL Serial data.
 - PWM, pins: 3,6,5,9,10,11. Proporcionan una salida PWM de 8 bits a 490Hz.
 - SPI, pins: 10(SS), 11(MOSI), 12(MISO), 13(SCK). Estos pins soportan la comunicación SPI (Bus SPI).
 - External Interrupts, pins : 2 y 3. Estos pins se pueden configurar para llamar una función del programa cuando reciben una interrupción externa.
 - LED: 13. Hay un LED conectado al pin 13. Cuando el pin tiene un valor HIGH, el LED se enciende. Cuando el pin tiene un valor LOW, el led se apaga.
 - https://www.arduino.cc/en/Tutorial/Foundations/DigitalPins



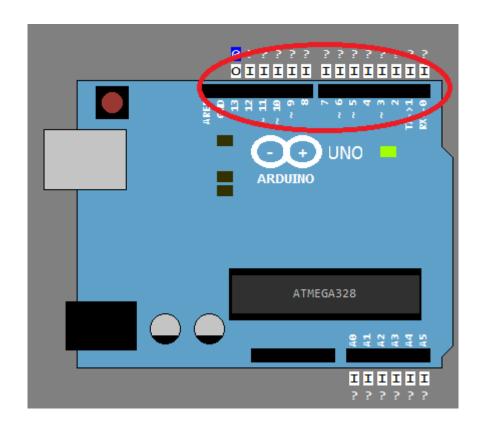


DIGITAL INPUT...





DIGITAL INPUT..





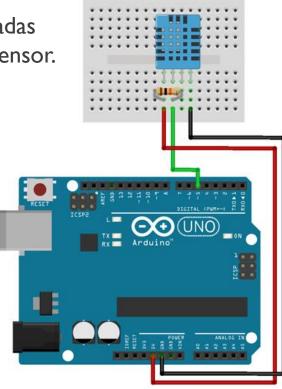
SENSORES DIGITALES

- Vimos que los sensores digitales, nos pueden dar una señal digital simple con dos estados o una salida en bus digital.
- En caso que el sensor use comunicación por bus, deberemos usar algunos de los buses que implementa Arduino o usar hardware adicional entre el Arduino y el bus.
- Arduino dispone de buses serie I2C y SPI para comunicarse con dispositivos sin necesidad de HW adicional
- Un bus (o canal) es un sistema digital que transfiere datos entre los componentes de una dispositivo electrónico o entre varios. Está formado por cables o pistas en un circuito impreso, dispositivos como resistencias y condensadores además de circuitos integrados.



SENSORES DIGITALES..

 Estas salidas en bus de datos se pueden conectar a las entradas digitales de Arduino para poder medir la(s) variable(s) del sensor.





MEDIR LA VARIABLE DEL SENSOR EN ARDUINO

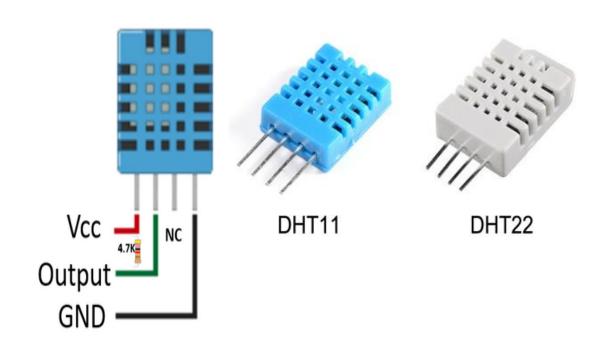
- Para medir la variable del sensor digital en el código Arduino, se pueden seguir 2 caminos:
 - I. Código Arduino
 - 2. Código Arduino con librerías del sensor



EJEMPLO: DHT11, DHT22

Características

- Fuente de alimentación de 3.3V a 6V.
- Consumo de corriente de 2.5mA.
- Salida Señal digital.
- Medición de la temperatura entre -40 y 125°C, con una precisión de 0,5°C a 25°C.
- Resolución de la medición de temperatura: 8-bit, 0,1°C
- Medición de la humedad entre 0 y 100%, con una precisión del 2-5% para temperaturas entre 0 y 50°C.
- Resolución de la medición de la temperatura: 8-bit, 0,1%
- Frecuencia de muestreo de 2 muestras/s: 2Hz.
- Conexión por resistencia con un valor entre 4,7K y 10K.

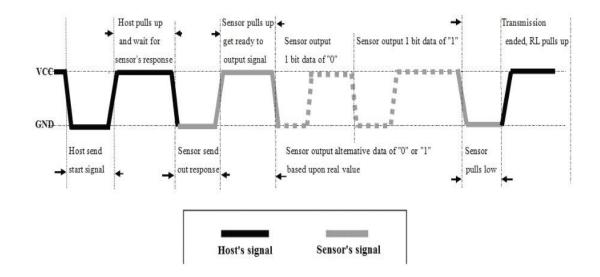




EJEMPLO..

- Ventaja, No habrá que generar código en Arduino IDE, para tratar directamente la señal digital a grados o porcentaje de humedad relativa.
- La trama de 40-bits que transmite, la precisión es más elevada. Incluso incluye unos bits de paridad para detectar fallos en la señal. Eso con una señal analógica no se tiene, a parte de que la analógica es muy sensible a las variaciones de voltaje.
- https://cdnshop.adafruit.com/datasheets/Digital+humidity+and+temperat ure+sensor+AM2302.pdf

See below figure for overall communication process, the interval of whole process must beyond 2 seconds.







EJEMPLO..

```
#include "DHT.h"
     // Ejemplo sencillo de uso para el DHT22
     const int DHTPin = 7;
    DHT dht(DHTPin, DHTTYPE);
     void setup() {
        Serial.begin(9600);
10
11
        Serial.println("Test DHT22");
12
13
        dht.begin();
14
15
     void loop() {
16
        // Tiempo de espera entre tomas de mediciones de 2 segundos.
17
        delay(2000);
18
19
20
        // Lee temperatura y humedad durante unos 250ms
        float h = dht.readHumidity();
21
        float t = dht.readTemperature();
23
        if (isnan(h) || isnan(t)) {
24
           Serial.println("Fallo en la lectura");
25
26
           return;
27
28
29
        Serial.print("Humedad relativa: ");
30
        Serial.print(h);
31
        Serial.print(" %\t");
32
        Serial.print("Temperatura: ");
33
        Serial.print(t);
34
        Serial.print(" *C ");
35
36 }
```

VIDEO

- https://youtu.be/u7hgbGgdVcE
- ¿Utiliza sensores digitales, si/no y por qué?



CONCLUSIÓN

- A la hora de elegir un sensor, debemos leer detenidamente las características y elegir uno que sea compatible con nuestro sistema (intensidad y voltaje) y que sea sencillo de usar o nos faciliten una librería potente.
- Otras características que debemos tener en cuenta: La resolución de un sensor es el menor cambio en la magnitud de entrada que se aprecia en la magnitud de salida. Sin embargo, la precisión es el máximo error esperado en la medida.



REFERENCIAS

- 1. Sensores en entradas y salidas digitales. https://aprendiendoarduino.wordpress.com/2017/10/22/entradas-y-salidas-digitales-arduino/
- 2. Digital pins tutorial. https://www.arduino.cc/en/Tutorial/Foundations/DigitalPins
- 3. Blog Luis Llamas. https://www.luisllamas.es/arduino-dht11-dht22/
- 4. Circuito.io https://www.circuito.io/blog/arduino-uno-pinout/
- 5. DHT22 Datasheet. https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf



PREGUNTASY COMENTARIOS



