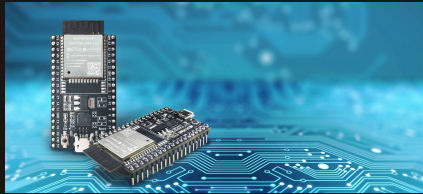


Sensores

Introducción al Internet de las Cosas



Marco Teran
Universidad Sergio Arboleda

2023

Contenido

- 1 Introducción al IoT y los sensores
- 2 Sistemas embebidos y sensores
- 3 Sistemas embebidos y sensores
- 4 Tipos de sensores para el IoT
- 5 Procesamiento de los datos de sensor
- 6 Métricas de desempeño de los sensores
- 7 Desafíos en la implementación de sensores para el IoT
- 8 Laboratorio de sensores
- 9 Installing DHT Sensor Library
- 10 Funciones en Arduino
- 11 Interrupciones

Introducción al IoT y los sensores

Introducción al IoT y los sensores

- El IoT es un sistema de dispositivos interconectados que recopilan y comparten datos a través de Internet.
- Los sensores son dispositivos que se utilizan para medir y detectar cambios en el entorno físico, como la temperatura, la humedad, la presión, el movimiento, la luz y el sonido.
- Los sensores son esenciales para el IoT, ya que permiten a los dispositivos recopilar datos del mundo físico y transmitirlos a través de la red.

Sistemas embebidos y sensores

Sistemas embebidos y sensores

- Los sistemas embebidos son sistemas informáticos integrados en dispositivos físicos, como automóviles, electrodomésticos, cámaras y dispositivos médicos.
- Los sensores se utilizan comúnmente en los sistemas embebidos para medir variables como la temperatura, la humedad, la presión, el movimiento y la luz.
- Los sistemas embebidos a menudo requieren sensores altamente precisos y confiables para garantizar un funcionamiento seguro y efectivo.

Sistemas embebidos y sensores

- El procesamiento en sistemas embebidos simples consiste en leer entradas, computar salidas, escribir salidas y esperar hasta el próximo ciclo de procesamiento o evento de activación.
- En sistemas embebidos, las entradas generalmente consisten en comandos ingresados por un usuario, comandos recibidos de otras fuentes como un servidor de red, y mediciones de sensores.

Sistemas embebidos y sensores

Sensores

Los sensores son componentes eléctricos o electrónicos sensibles a alguna propiedad de su entorno y producen una salida correspondiente a la propiedad medida.

- Los sensores pasivos miden aspectos del entorno como temperatura o intensidad de luz respondiendo directamente al parámetro medido sin perturbar el entorno.
- Los sensores activos generan algún tipo de estímulo que aplican al entorno y miden la respuesta.

Sensores

- Los sensores inteligentes integran la función de medición y conversión en una sola unidad, lo que simplifica el diseño de hardware.
- Los circuitos básicos interfazan los tipos comunes de sensores con los procesadores embebidos.
- Algunos tipos de sensores pueden operar en ambos modos, pasivo y activo.
- Ejemplos de sensores activos incluyen el sensor de distancia ultrasónico, radar y sensores LIDAR.
- La elección de un sensor inteligente en aplicaciones específicas dependerá de factores como el costo del sensor, el volumen de producción anticipado y la presión del tiempo de mercado.

Tipos de sensores para el IoT

Tipos de sensores para el IoT

- Los sensores utilizados en el IoT incluyen sensores de temperatura, humedad, presión, movimiento, luz, sonido, posición y gas.
- Los sensores de temperatura miden la temperatura ambiental o de un objeto en particular.
- Los sensores de humedad miden la cantidad de humedad en el aire o en un objeto.
- Los sensores de presión miden la presión atmosférica o la presión en un objeto.
- Los sensores de movimiento detectan cambios en la posición, velocidad o aceleración de un objeto.

Sensores de luz

Sensores de luz

- Los sensores de luz en sistemas embebidos pueden variar desde fotoresistores simples hasta complejos arreglos de sensores de múltiples bandas utilizados en dispositivos como cámaras de video, microscopios y telescopios astronómicos.
- Los fotodiodos y fototransistores son dispositivos semiconductores que convierten la luz en corriente eléctrica, son más sensibles y consistentes que los fotoresistores.
- Los sensores de video contienen una matriz bidimensional de elementos sensibles a la luz, con filtrado que limita la entrada de cada elemento a un rango de frecuencia específico dentro del espectro de luz.

Sensores de temperatura

Sensores de temperatura

- Un termistor es un elemento resistivo que cambia la resistencia con los cambios de temperatura y se puede medir su resistencia a través de un circuito divisor de voltaje.
- Un termopar es un dispositivo de detección de temperatura que se conecta a dos metales diferentes en un solo punto para medir la temperatura en ese lugar.
- Los termistores son más adecuados para rangos de temperatura de -50 a $250\text{ }^{\circ}\text{C}$, mientras que los termopares soportan un rango de temperatura más amplio de aproximadamente -200 a $1,250\text{ }^{\circ}\text{C}$.

Sensores de presión

Sensores de presión

- Los sensores de presión miden la presión de líquidos y gases, ya sea de forma absoluta o relativa a algún punto de referencia como la presión atmosférica.
- Los sensores de presión diferencial miden la diferencia de presión entre dos ubicaciones, y se usan en aplicaciones como medir la caída de presión a través de un filtro de fluido.
- Los sensores de presión de bajo costo suelen ser contruidos con un material piezorresistivo que cambia en resistencia en respuesta a la presión medida, y el sistema integrado mide la resistencia del sensor para determinar la lectura de presión.

Sensores de humedad

Sensores de humedad

- Los sensores de humedad miden la presión parcial del vapor de agua en el ambiente y la expresan en un porcentaje relativo a la presión de saturación.
- Son útiles en aplicaciones de monitoreo y control ambiental para mantener la humedad dentro de límites deseados para equipos electrónicos o confort humano, y a menudo se combinan con sensores de temperatura.
- Los sensores de humedad de bajo costo se construyen con elementos sensibles que varían su capacidad según la humedad y miden la capacitancia del elemento para calcular la humedad relativa, y algunos son "smart sensors" con capacidad de procesamiento y comunicación digital.

Sensores de flujo

Sensores de flujo

- Los sensores de flujo miden la cantidad de gas o líquido que pasa por una región activa del sensor, generalmente mediante la colocación de algún tipo de restricción en el fluido y midiendo el efecto de la restricción.
- Algunos diseños de sensores de flujo utilizan tecnologías de medición que no requieren restricción de flujo, como ultrasonido y láser. Estos sensores pueden tener salida analógica o interfaz digital.
- Los sensores de flujo se usan en una amplia variedad de aplicaciones, desde monitorear el consumo de combustible en automóviles y aviones hasta medir el flujo de anestésicos en aplicaciones médicas, así como en medidores de agua y gas en hogares y edificios comerciales.

Sensores de fuerza

Sensores de fuerza

- Un sensor de fuerza mide la cantidad de fuerza aplicada a un objeto, como una báscula de baño.
- Los sensores de fuerza comúnmente utilizan resistores de fuerza, piezoeléctricos o fluidos/gases bajo presión para medir la fuerza aplicada.
- Los sensores de fuerza se utilizan en una amplia variedad de aplicaciones, desde medir el peso de una persona en una báscula hasta controlar la fuerza en maquinaria y robots industriales.

Sensores ultrasónicos

Sensores ultrasónicos

- Un sensor ultrasónico genera una onda de sonido de alta frecuencia que se transmite a una región de medición y escucha los ecos de los objetos en el rango.
- El tiempo entre la transmisión y recepción de la señal, multiplicado por la velocidad del sonido en el medio de medición, representa la distancia de ida y vuelta del transmisor al objeto y de vuelta al receptor.
- Los sensores ultrasónicos simples utilizan dos señales digitales para controlar la operación del sensor y leer la salida de medición. El pin de disparo es una señal de entrada del sensor que inicia un ciclo de medición en respuesta a un borde de pulso ascendente. El pin Echo es una salida del sensor que se activa cuando se transmite el pulso y vuelve a desactivarse cuando se recibe el eco.

Sensores de audio

Sensores de audio

- Los sensores de audio reciben señales sonoras en el rango de audición humana y producen una salida eléctrica en respuesta a la señal recibida. Un micrófono estándar es un ejemplo de sensor de audio.
- Los dispositivos de asistente inteligente escuchan continuamente el sonido en su entorno y, cuando se detecta una secuencia de sílabas de activación (como "Alexa" o "Hey Google"), el dispositivo graba los sonidos siguientes e intenta interpretar el comando proporcionado como una secuencia de palabras.
- En aplicaciones más simples, un sensor de audio puede simplemente monitorear la intensidad del sonido en su entorno y producir una salida cuando el nivel de sonido supera un umbral. Las aplicaciones más sofisticadas del monitoreo de audio incluyen la detección de retardo de

Sensores magneticos

Sensores magneticos

- Un sensor de campo magnético, o magnetómetro, detecta la presencia de un campo magnético en la cercanía del sensor. Los sensores simples responden solo a la intensidad del campo.
- Los sensores más sofisticados miden los componentes vectoriales tridimensionales del campo magnético.
- Un ejemplo de uso es el magnetómetro que detecta el campo magnético terrestre, la cual puede ser calibrada para obtener una orientación más precisa basada en la ubicación en la que se encuentra. También se utilizan sensores magnéticos en sistemas de seguridad para detectar la apertura de puertas mediante la presencia de un imán que genera un campo magnético local.

Sensores químicos

Sensores químicos

- Los sensores químicos miden atributos de componentes químicos en la cercanía del sensor, siendo sensibles a la presencia de un elemento o compuesto particular en el gas o líquido que lo rodea.
- Ejemplos comunes de sensores químicos incluyen detectores de monóxido de carbono y radón en hogares, y los motores de gasolina modernos contienen sensores de oxígeno en el sistema de escape.
- Sensores de bajo costo y un solo chip están disponibles para medir la calidad del aire, reportando los niveles de dióxido de carbono y compuestos orgánicos volátiles en el aire circundante, y la nanotecnología está permitiendo la producción de una variedad de dispositivos de detección química.

Sensores ionizantes

Sensores ionizantes

- La radiación ionizante se compone de partículas electromagnéticas capaces de ionizar átomos y moléculas, y es perjudicial para el tejido vivo a niveles bajos.
- Tanto la radiación natural como la generada por el hombre se puede medir con sensores de radiación ionizante que miden los cambios en un material sensible debido a la radiación incidente.
- Los sensores de radiación ionizante pueden informar eventos individuales o acumular la dosis de radiación en un período de tiempo.

Sensores RADAR

Sensores RADAR

- RADAR es un acrónimo para Radio Detection and Ranging, que se refiere a un sistema que transmite señales de radio y recibe eco de los objetos encontrados por los pulsos.
- Los sensores de radar de un solo chip están disponibles para la detección de vehículos cercanos en aplicaciones como el control de crucero adaptativo en automóviles.
- Los sensores de radar automotriz miden la distancia y la velocidad relativa del vehículo por delante del vehículo anfitrión y permiten mantener la distancia adecuada y responder a eventos como frenadas repentinas del vehículo delantero.

Sensores LIDAR

Sensores LIDAR

- LIDAR es un acrónimo de Light Detection and Ranging, que utiliza luz láser o infrarroja para detectar objetos en la proximidad del sensor, en lugar de señales de radiofrecuencia como el radar.
- El sensor LIDAR emite un pulso de luz enfocado y mide la distancia a un obstáculo en la dirección del haz mediante la detección del tiempo de recepción de la reflexión del objeto. Repite este proceso miles de veces para crear un mapa tridimensional de la zona.
- Aunque los componentes ópticos necesarios para los sensores LIDAR de precisión son costosos, su precio ha disminuido en los últimos años gracias al desarrollo de la tecnología de vehículos autónomos.

Sensores de vídeo e infrarrojos

Sensores de vídeo e infrarrojos

- Cámaras de video digital color tradicionales producen una secuencia de imágenes compuestas de píxeles que contienen tres intensidades de color (rojo, verde y azul).
- Sistemas integrados pueden utilizar cámaras de video para capturar escenas para consumo humano, como en una aplicación de videoportero.
- Algunas áreas de aplicación requieren el procesamiento de flujos de datos de video por máquinas, como en un sistema de seguridad que monitorea una zona y detecta movimiento comparando imágenes secuenciales.

Sensores de vídeo e infrarrojos

- Vehículos autónomos necesitan un sistema de procesamiento de video más sofisticado que el simple diferencial de imágenes, integrando varios tipos de datos de sensores incluyendo cámaras de video.
- Cámaras infrarrojas son similares a las de video, pero sensibles a longitudes de onda de luz más largas que el rojo visible, respondiendo al calor radiado en el infrarrojo, útiles en aplicaciones como monitoreo de temperaturas en circuitos eléctricos.
- El procesamiento de los datos de video en vehículos autónomos requiere algoritmos de inteligencia artificial sofisticados y procesamiento de alta velocidad y en tiempo real.

Sensores inerciales

Sensores inerciales

- Un sensor inercial detecta cambios en el movimiento del cuerpo al que está unido.
- Para caracterizar completamente la aceleración y la rotación de un objeto en el espacio tridimensional se requieren tres acelerómetros y tres sensores de velocidad de rotación.
- Los sensores inerciales se utilizan en aeronaves, naves espaciales y barcos para rastrear con precisión el movimiento en presencia de perturbaciones y deben tener errores de medición lo suficientemente pequeños para no degradar la medición de posición a un grado inaceptable.

Sensores GNSS

Sensores GNSS

- Los receptores del Sistema de Posicionamiento Global (GPS) recopilan señales de satélites en órbita alrededor de la Tierra para calcular la posición del receptor y proporcionar la hora actual.
- Varios sistemas de navegación por satélite constelaciones están en funcionamiento en todo el mundo, incluyendo Galileo, Beidou y Glonass.
- Los receptores modernos de navegación por satélite son capaces de recibir señales de una o varias constelaciones simultáneamente, lo que aumenta la precisión y la velocidad de la medición de la posición.

Sensores GNSS

Sensores GNSS

- Los receptores GNSS son comunes en aeronaves, automóviles, dispositivos militares, teléfonos inteligentes y otros equipos que operan en exteriores.
- La integración de sensores inerciales con un receptor GNSS crea un sistema GPS/INS, que se utiliza ampliamente en la navegación aérea, naval y militar.
- El siguiente tema tratará sobre las tecnologías de comunicación más útiles para conectar sensores a procesadores integrados.

Procesamiento de los datos de sensor

Procesamiento de los datos de sensor

- Los errores de medición en los sensores pueden tener diversas causas.
- En algunas aplicaciones, no es necesario corregir el error de medición, pero en otras sí es crítico eliminarlo tanto como sea posible.
- Los sensores de precisión suelen ser más costosos, pero pueden incluir técnicas de diseño que mejoran la calidad de la medición.

Procesamiento de los datos de sensor

- En algunos casos, es necesario realizar una calibración adicional para mejorar la precisión de la medición.
- Para reducir el ruido aleatorio en las mediciones de los sensores, se pueden tomar varias lecturas y promediarlas.
- En situaciones con señales que varían rápidamente, puede ser útil construir un filtro digital para procesar la señal de entrada.

Métricas de desempeño de los sensores

Métricas de desempeño de los sensores

Métricas de desempeño

Las métricas de desempeño de los sensores son medidas que se utilizan para evaluar la calidad y precisión de los datos que se recopilan.

- La resolución se refiere a la capacidad de un sensor para detectar cambios pequeños en una variable medida.
- La precisión se refiere a la medida en que los datos del sensor son comparables con un valor de referencia.
- La linealidad se refiere a la relación entre la entrada del sensor y la salida del sensor.
- La estabilidad se refiere a la capacidad de un sensor para mantener su precisión a lo largo del tiempo.

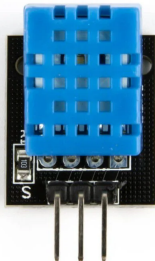
Desafíos en la implementación de sensores para el IoT

Sistemas embebidos y sensores

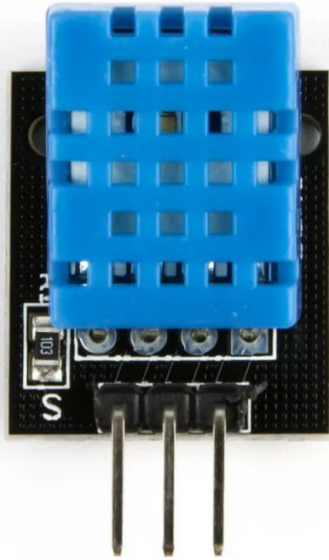
- Seleccionar los sensores adecuados para una tarea específica puede ser un desafío, ya que se requiere una comprensión detallada de los parámetros de rendimiento del sensor y las condiciones del entorno.
- La calibración y la integración de los sensores en el sistema embebido también pueden presentar desafíos, ya que es importante asegurarse de que los sensores sean precisos y funcionen correctamente.
- Además, los desafíos en la transmisión de datos pueden incluir problemas de seguridad y privacidad de los datos transmitidos.
- Otros desafíos incluyen la conservación de energía de los sensores, el procesamiento de datos en tiempo real y la compatibilidad con diferentes plataformas y protocolos de red.

Laboratorio de sensores

Introducción al DHT11

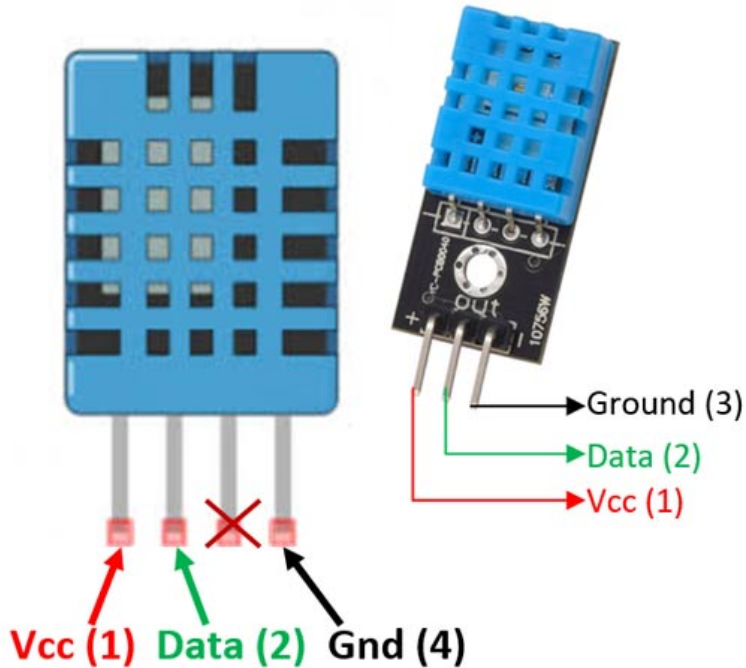


- El DHT11 es un sensor de humedad y temperatura de bajo costo.
- Es fácil de usar y se puede conectar directamente a una placa Arduino o Raspberry Pi.
- Proporciona mediciones precisas de la humedad relativa y la temperatura ambiente.



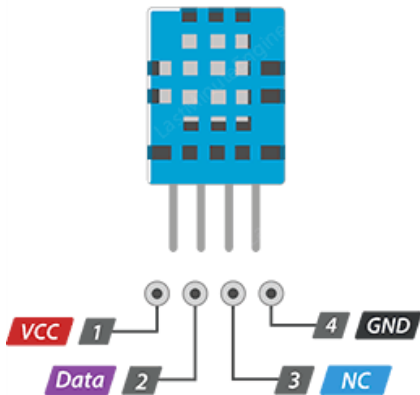
Importancia del DHT11 en el IoT

- El DHT11 es un sensor esencial en aplicaciones de IoT relacionadas con el monitoreo ambiental y control de clima.
- Puede utilizarse en combinación con otros sensores para mejorar la precisión de los datos recopilados.
- La facilidad de uso y bajo costo lo hacen ideal para proyectos de IoT de bajo presupuesto.

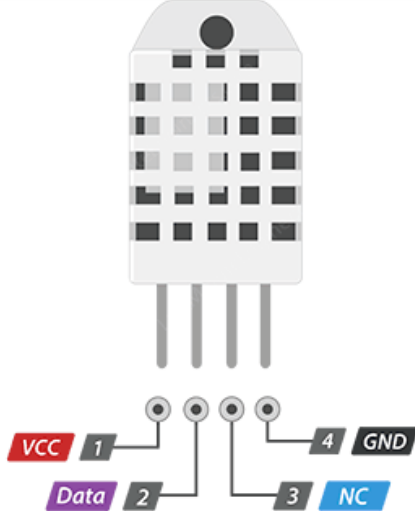


Principales características del DHT11

- Rango de medición de humedad: 20% a 90% HR con una precisión de $\pm 5\%$.
- Rango de medición de temperatura: 0 a 50 grados Celsius con una precisión de ± 2 grados.
- Voltaje de funcionamiento: 3.3V a 5V.
- Salida digital con una resolución de 1 grado Celsius y 1% de humedad relativa.
- Tiempo de respuesta rápido: 2 segundos para la humedad y 5 segundos para la temperatura.
- Bajo consumo de energía: menos de 1mA en promedio.

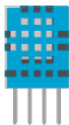


DHT11 Pinout

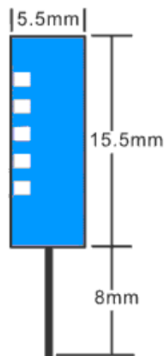
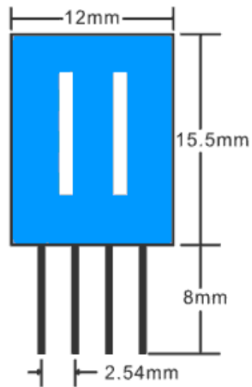
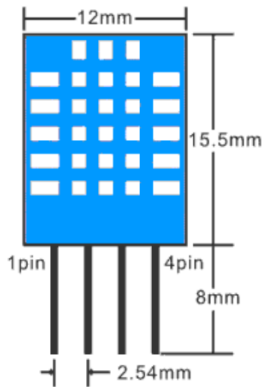
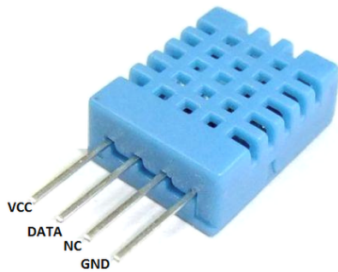


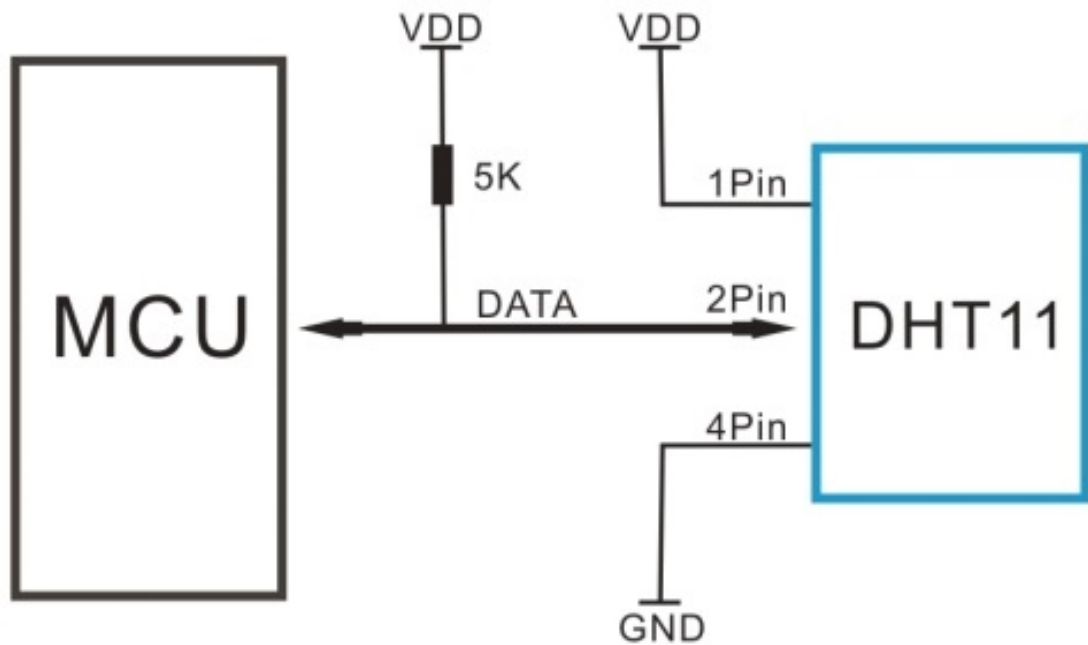
DHT22 Pinout

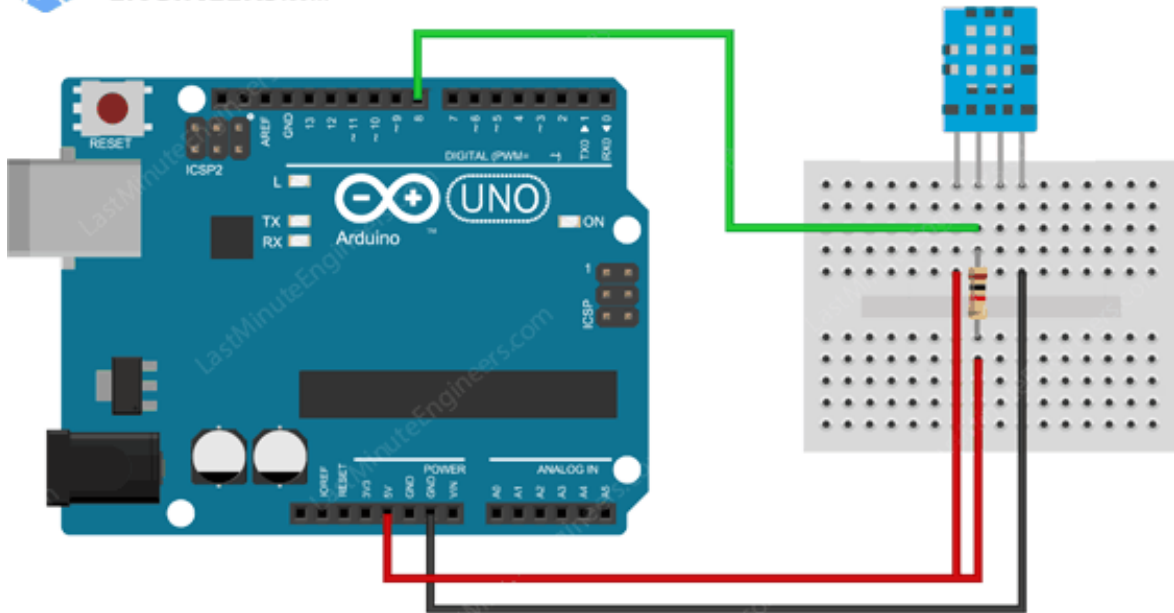


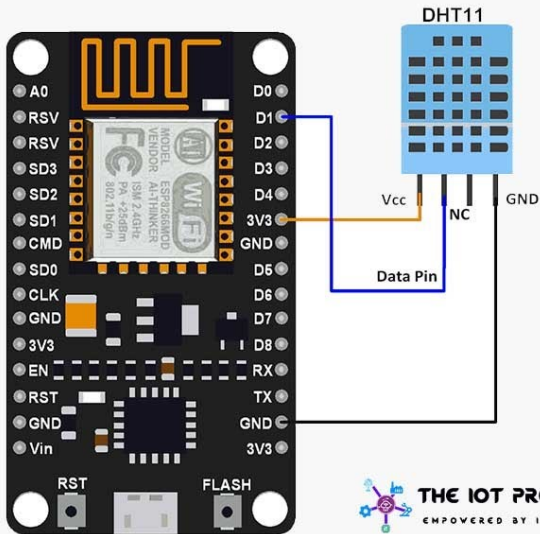
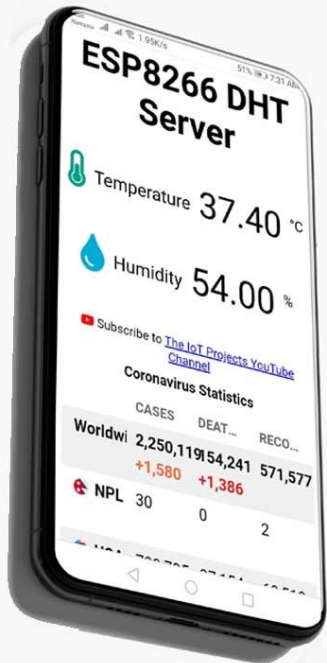


	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Humidity Range	20-80% / 5%	0-100% / 2-5%
Temperature Range	0-50°C / $\pm 2^{\circ}\text{C}$	-40 to 80°C / $\pm 0.5^{\circ}\text{C}$
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Body size	15.5mm x 12mm x 5.5mm	15.1mm x 25mm x 7.7mm
Advantage	Ultra low cost	More Accurate





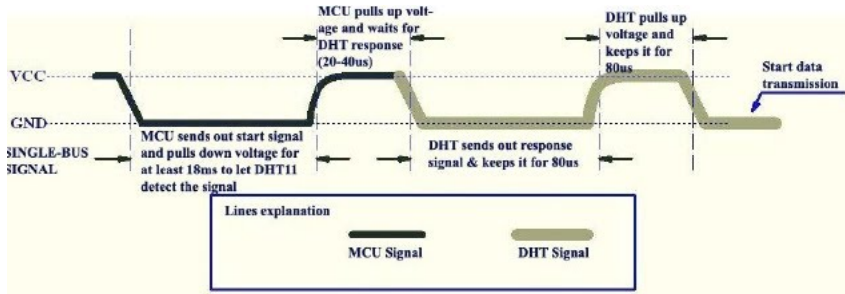




THE IOT PROJECTS
EMPOWERED BY INNOVATION

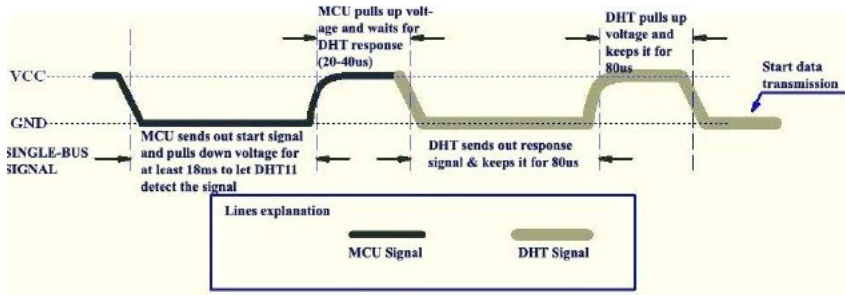
DHT11: protocolo OneWire

- Para comunicarnos con el sensor debemos de implementar en nuestra MCU el protocolo que el fabricante nos proporciona el datasheet es el siguiente:
- La MCU se comunicará con el DHT y éste le enviará 5 bytes de datos donde los 2 primeros corresponderán a la Humedad Relativa, los 2 siguientes a la Temperatura y el último byte para el Checksum.



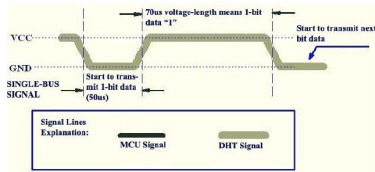
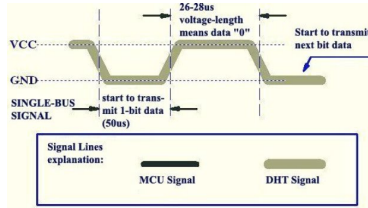
DHT11: protocolo OneWire

- Primero de todo MCU mandará una secuencia de START al DHT y después el mismo esperará a que el DHT responda. Después el DHT mandará una señal al MCU para indicar de que está preparado para inicial la trasmisión bit a bit.



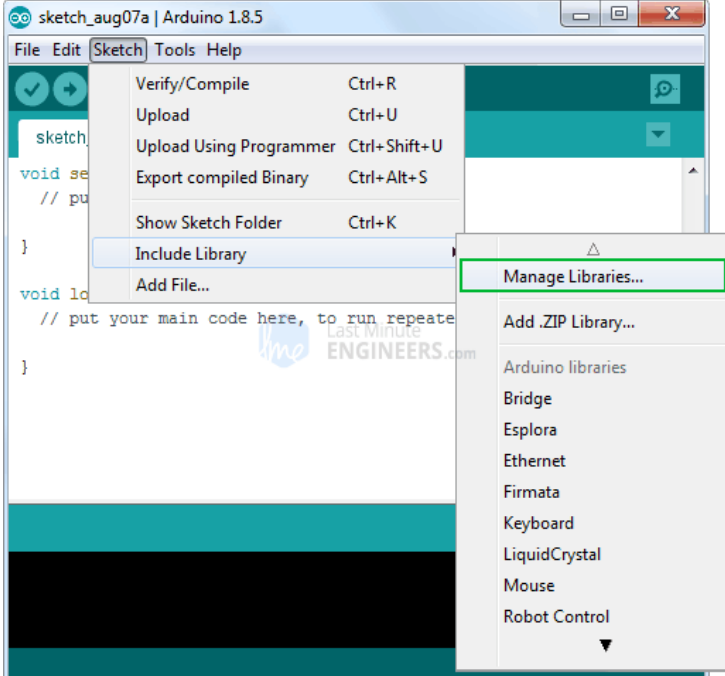
DHT11: protocolo OneWire

- Siempre que inicia la transmisión de un bit el sensor, mandará un pulso de bajo voltaje de 50us y después enviará un pulso de alto voltaje que según la duración del mismo el pulso significará que es un "0" o un "1".





Installing DHT Sensor Library





Type Topic

DHT sensor library by Adafruit

Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors

[More info](#)

Version 1....

Install

DHT sensor library for ESPx by beegee_tokyo

Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized library to match ESP32 requirements. Last changes: Use correct field separator in keywords.txt.

[More info](#)

DHTlib by Rob Tillaart Version 0.1.28 **INSTALLED**

Optimized Library for DHT Temperature & Humidity Sensor on AVR only.

[More info](#)



Type Topic

Adafruit LSM303DLHC by **Adafruit**

Unified sensor driver for Adafruit's LSM303 Breakout (Accelerometer + Magnetometer) Unified sensor driver for Adafruit's LSM303 Breakout (Accelerometer + Magnetometer)

[More info](#)

Adafruit TSL2561 by **Adafruit**

Unified sensor driver for Adafruit's TSL2561 breakouts Unified sensor driver for Adafruit's TSL2561 breakouts

[More info](#)

Adafruit Unified Sensor by **Adafruit**

Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used by many Adafruit sensor libraries.

[More info](#)

Version 1.... ▾

Install

Funciones en Arduino

Funciones en Arduino

Definición

Las funciones son bloques de código reutilizable.

- Ayudan a dividir un programa en tareas más pequeñas y manejables.
- Ayudan a mantener un código más ordenado y legible.
- Son esenciales para la programación estructurada en Arduino.
- Se pueden crear funciones personalizadas o utilizar las funciones predefinidas de la librería de Arduino.

Características de las funciones en Arduino

- Deben tener un nombre descriptivo.
- Pueden tener parámetros que se utilizan para pasar información a la función.
- Pueden devolver valores utilizando la palabra clave "return".
- Pueden ser llamadas desde cualquier parte del código.
- Se definen fuera de la función "void setup" y antes de la función "void loop".

Aplicaciones de las funciones en Arduino

- Lectura de sensores: se pueden crear funciones para leer diferentes tipos de sensores como temperatura, humedad, luz, etc.
- Control de actuadores: se pueden crear funciones para controlar motores, luces, relés, etc.
- Realización de cálculos complejos: se pueden crear funciones para realizar cálculos matemáticos o conversiones de unidades.
- Comunicación con otros dispositivos: se pueden crear funciones para enviar y recibir datos a través de diferentes protocolos como I2C, SPI, Bluetooth, etc.

Tipos de funciones en Arduino

- 1 Funciones de retorno de valor
- 2 Funciones sin retorno de valor
- 3 Funciones con parámetros
- 4 Funciones con valores predeterminados

Funciones de retorno de valor en Arduino

Definición

Las funciones de retorno de valor en Arduino son aquellas que devuelven un valor específico después de que se ejecuta el bloque de código. Estas funciones se definen con el tipo de dato que se va a devolver y deben utilizar la palabra clave "return" para devolver el valor.

- Las funciones de retorno de valor pueden recibir parámetros para utilizar en el bloque de código.
- El tipo de dato que se devuelve puede ser cualquiera que se pueda manejar en Arduino (enteros, flotantes, caracteres, etc.).
- El valor que se devuelve debe ser compatible con el tipo de dato que se define en la definición de la función.

Funciones de retorno de valor en Arduino

```
int sumar(int a, int b) {  
    int resultado = a + b;  
    return resultado;  
}  
  
int resultadoSuma = sumar(2, 3);
```

Funciones sin retorno de valor en Arduino

Definición

Las funciones sin retorno de valor en Arduino son aquellas que no devuelven ningún valor específico después de que se ejecuta el bloque de código. Estas funciones se definen con el tipo de dato "void" y no utilizan la palabra clave "return".

- Las funciones sin retorno de valor también pueden recibir parámetros para utilizar en el bloque de código.
- Estas funciones suelen utilizarse para realizar una tarea específica en el programa sin necesidad de devolver un valor.
- Al no devolver un valor, estas funciones no pueden ser utilizadas en una expresión o asignación de variable.

Funciones sin retorno de valor en Arduino

```
void prenderLed(int pin) {  
    digitalWrite(pin, HIGH);  
}  
  
prenderLed(13);
```

Funciones en Arduino: Funciones en bloque

Definición

Las funciones en bloque son un tipo de función que permite agrupar varias instrucciones juntas para realizar una tarea específica.

- Pueden tener parámetros que se utilizan para pasar información a la función.
- No devuelven valores utilizando la palabra clave "return".
- Pueden ser llamadas desde cualquier parte del código.

Funciones en Arduino: Funciones en bloque

```
void encenderLed(int pinLed) {  
    digitalWrite(pinLed, HIGH);  
    delay(1000);  
    digitalWrite(pinLed, LOW);  
    delay(1000);  
}
```

Funciones en Arduino: Funciones que devuelven valores

Definición

Las funciones que devuelven valores son un tipo de función que permite realizar cálculos y devolver un valor como resultado de la función.

- Tienen un tipo de dato de retorno que indica el tipo de valor que devuelve la función.
- Pueden tener parámetros que se utilizan para pasar información a la función.
- Devuelven valores utilizando la palabra clave "return".
- Pueden ser llamadas desde cualquier parte del código.

Funciones en Arduino: Funciones que devuelven valores

```
int sumar(int num1, int num2) {  
    int resultado = num1 + num2;  
    return resultado;  
}
```

Funciones con parámetros por referencia en Arduino

Definición

Las funciones con parámetros por referencia en Arduino son aquellas que utilizan la dirección de memoria de una variable existente en el programa como parámetro de entrada. Esto permite que los cambios realizados en la función afecten a la variable original fuera de la función.

- Para indicar que un parámetro es por referencia, se utiliza el símbolo "&" antes del nombre del parámetro en la definición de la función.
- Estas funciones permiten reducir el uso de memoria en el programa al evitar la creación de nuevas variables en la función.
- Es importante tener en cuenta que al utilizar parámetros por referencia, se deben tomar precauciones para evitar cambios no deseados en la variable original.

Funciones con parámetros por referencia en Arduino

```
void incrementar(int &x) {  
    x++;  
}  
  
int numero = 5;  
incrementar(numero);
```

Funciones en Arduino: Funciones con parámetros por referencia

Definición

Las funciones con parámetros por referencia son un tipo de función que permite modificar los valores de una variable pasada como parámetro dentro de la función.

- Los parámetros se definen con el símbolo "&" al final del tipo de dato.
- Los parámetros se pasan sin el símbolo "&" al llamar la función.
- No devuelven valores utilizando la palabra clave "return".
- Pueden ser llamadas desde cualquier parte del código.

Funciones en Arduino: Funciones con parámetros por referencia

```
void duplicar(int& num) {  
    num = num * 2;  
}
```

Funciones en Arduino: Funciones con parámetros por valor

Definición

Las funciones con parámetros por valor son un tipo de función que permite utilizar el valor de una variable pasada como parámetro dentro de la función, sin modificar la variable original.

Ejemplo

Resumen

En este ejemplo se utilizan tres funciones diferentes:

- Una función para encender y apagar un LED.
- Una función para leer el valor del sensor LDR utilizando una función de retorno de valor.
- Una función para leer los valores de temperatura y humedad del sensor DHT11 utilizando parámetros con valores predeterminados.

Cada función tiene valores predeterminados para algunos de sus parámetros, lo que permite su uso sin necesidad de proporcionar valores para todos los parámetros. Además, se utilizan funciones de retorno de valor (en el caso de la función para leer el sensor LDR) y funciones void (en el caso de las otras dos funciones). Los valores de los sensores y el estado del LED se imprimen por comunicación serial para su posterior análisis o visualización.

Ejemplos de código para Arduino

```
// Ejemplo de función para encender y apagar un LED
void parpadearLed(int pin = 13, int tiempoEncendido = 500, int tiempoApagado =
    500) {
    digitalWrite(pin, HIGH);
    delay(tiempoEncendido);
    digitalWrite(pin, LOW);
    delay(tiempoApagado);
}

// Ejemplo de función para leer el sensor LDR
int leerLDR(int pin) {
    int lectura = analogRead(pin);
    return lectura;
}
```

Ejemplos de código para Arduino

```
// Ejemplo de función para leer el sensor DHT11
void leerDHT11(int pin = 2, int tiempoEspera = 2000) {
    DHT dht(pin, DHT11);
    delay(tiempoEspera);
    float temperatura = dht.readTemperature();
    float humedad = dht.readHumidity();
    Serial.print("Temperatura: ");
    Serial.print(temperatura);
    Serial.print(" grados Celsius, Humedad: ");
    Serial.print(humedad);
    Serial.println(" %");
}

void setup() {
    pinMode(13, OUTPUT); // configurar el pin 13 como salida
    Serial.begin(115200); // iniciar comunicación serial
}
```


Ejemplos de código para Arduino

```
void loop() {  
  // Ejecutar la función para encender y apagar el LED  
  parpadearLed();  
  
  // Leer el valor del sensor LDR y mostrarlo por serial  
  int valorLDR = leerLDR(A0);  
  Serial.print("Valor LDR: ");  
  Serial.println(valorLDR);  
  
  // Leer los valores del sensor DHT11 y mostrarlos por serial  
  leerDHT11();  
}
```

Lectura de un sensor LDR mediante una función

```
int leerLDR(int pinLDR) {  
    int valorLDR = analogRead(pinLDR);  
    return valorLDR;  
}  
  
int pinSensorLDR = A0;  
int valorSensorLDR = leerLDR(pinSensorLDR);  
  
void setup() {  
    // Configuración del pin del sensor LDR  
    pinMode(pinSensorLDR, INPUT);  
}  
  
void loop() {  
    // Realizar acciones con el valor del sensor LDR  
}
```

Interrupciones

Interrupciones

- A menudo en un proyecto necesitas el ESP32 para ejecutar tu programa normal, mientras que continuamente necesitas que se monitorize para algún tipo de evento. Una solución ampliamente adoptada es el uso de una interrupción.
- El ESP32 ofrece hasta 32 ranuras de interrupción para cada núcleo. Cada interrupción tiene un cierto nivel de prioridad y se puede clasificar en dos tipos.
 - **Interrupciones de hardware** – Estas ocurren en respuesta a un evento externo. Por ejemplo, una interrupción de GPIO (cuando se pulsa una tecla) o una interrupción de toque o pulsación (cuando se detecta el pulsar)
 - **Interrupciones de software** – Estas ocurren en respuesta a una instrucción de software. Por ejemplo, una simple interrupción de temporizador o una interrupción de temporizador de vigilancia (cuando el temporizador se agota)

```
attachInterrupt (GPIOPin, ISR, Mode);
```

Esta función toma tres parámetros:

- **GPIOPin:** Establece la clavija GPIO como una clavija de interrupción, que le dice al ESP32 qué clavija debe monitorear.
- **ISR:** Es el nombre de la función que se llamará cada vez que se dispare la interrupción.
- **Mode:** Define cuándo se debe disparar la interrupción. Cinco constantes están predefinidas como valores válidos:

LOW	Los disparadores interrumpen cuando el pin está LOW
HIGH	Los disparadores interrumpen cuando el pin es HIGH
CHANGE	Los disparadores interrumpen cuando el pin cambia de valor, de HIGH a LOW o LOW a HIGH
FALLING	Los disparadores interrumpen cuando el pin va de HIGH a LOW
RISING	Los disparadores interrumpen cuando el pin va de LOW a HIGH

¡Muchas gracias por su atención!

¿Preguntas?



Contacto: Marco Teran
webpage: marcoteran.github.io/
e-mail: marco.teran@usa.edu.co

