UNIVERSIDAD DE LIMA

FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

ASIGNATURA:

INTERNET DE LAS COSAS IoT

GUÍA DE LABORATORIO

PRÁCTICA DE LABORATORIO N.º 1

SEÑALES DIGITALES Y ANALÓGICAS

2023 - 2

PRÁCTICA DE LABORATORIO N.º 1
INTRODUCCIÓN A LA PLATAFORMA ARDUINO

1. OBJETIVOS

- Reconocer el entorno de desarrollo de la plataforma Arduino IDE.
- Conocer la sintaxis de programación en ESP32 así como la declaración de las variables y las principales sentencias usadas.
- Manejar entradas y salidas digitales y analógicas del microcontrolador mediante la utilización de placas de prototipos.
- Usar el PWM del ESP32 para el encendido de un LED

2. FUNDAMENTO TEÓRICO

2.1 Software Arduino IDE y funciones clave

El Software Arduino IDE es un entorno de programación de código abierto y gratuito diseñado especialmente para programar y cargar un código en las placas Arduino. Entre las principales funciones del Arduino tenemos:

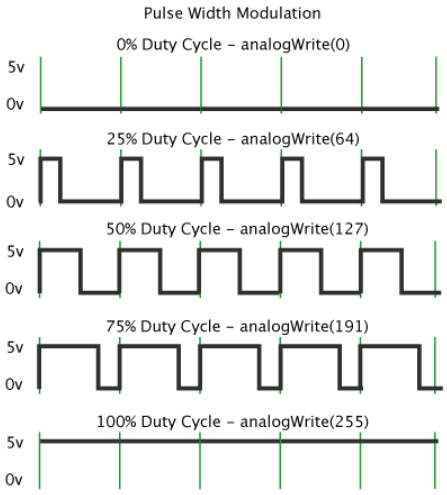
1. **Void setup ( ):** El propósito de la presente función es realizar la configuración inicial del programa, como por ejemplo establecer la dirección de los pines, y otras variables necesarias para el funcionamiento del programa. Todo lo que se encuentre dentro del Void setup ( ) se ejecutará solo una vez al inicio del programa.
2. **Void loop ( ):** Después de que la sea ejecutada la función void setup ( ), se ejecutará la función void loop ( ), la cual hará que el programa entre en un ciclo infinito, repitiendo constantemente el código contenido en dicha función. En este apartado es donde se realiza la lógica del programa, se toman decisiones y se realizan las acciones repetitivas.
3. **PinMode** ( )**:** La función pinMode( ) se utiliza para configurar los pines entradas (INPUT) o salidas (OUTPUT). Cuando un pin se establece como entrada, puede leer el estado de una señal eléctrica, mientras que, al configurarse como salida, puede enviar una señal eléctrica.
4. **DigitalWrite** ( )**:** Se utiliza para escribir un valor digital en un pin configurado como salida. Cuando un pin se configura como salida, puede tener dos estados:

   HIGH (1 o 5V) o LOW (0V).

5. **DigitalRead** ( ): Se utiliza para poder leer el valor de una entrada digital, estas entradas siempre están asociadas a los sensores, puede tener dos estados:

   HIGH (1 o 5V) o LOW (0V).

6. **AnalogRead(pin):** Lee el valor de un determinado pin definido como entrada analógica con una resolución de 10 bits. Esta instrucción sólo funciona en los pines (0-5). El rango de valor que podemos leer oscila de 0 a 1023 en una resolución de 10 bits o de 0 a 4069 en una resolución de 12 bits.
7. **map ( ) :** Permite hacer equivalencias entre diferentes rangos. Imagine que al variar un potenciómetro desea que se vea reflejado en la intensidad de brillo de un led, como la entrada analógica es 1024 niveles y las salidas analógicas varía en un rango de 256 niveles, tendríamos que hacer una equivalencia para que el 0 de entrada equivalga al 0 de salida, y el registro 1023 equivalga al registro de salida 255.
8. **Delay** ( )**:** Se utiliza para crear pausas en el programa y configurar el tiempo de ejecución de ciertas acciones durante un tiempo específico en milisegundos (ms).
9. **Serial.begin**( )**:** Se utiliza para inicial la comunicación serial en el Arduino. La comunicación serial es una forma de enviar y recibir datos byte por byte y permite monitorear el funcionamiento de un programa.
10. **PWM** El procedimiento para generar una señal analógica es el llamado PWM. Señal PWM (Pulse-width modulation) señal de modulación por ancho de pulso. Esta instrucción sirve para escribir un pseudo-valor a uno de los pines de Arduino marcados como PWM. El valor que se puede enviar a estos pines de salida analógica puede darse en forma de variable o constante, pero siempre con un margen de 0-255. Si enviamos el valor 0 genera una salida de 0 voltios en el pin especificado; un valor de 255 genera una salida de 5 voltios de salida en el pin especificado. Para valores de entre 0 y 255, el pin saca tensiones entre 0 y 5 voltios - cuanto mayor sea el valor, más a menudo estará a HIGH (5 voltios). Teniendo en cuenta el concepto de señal PWM. Si una señal tiene un periodo de 10 ms y sus pulsos son de ancho (PW) 2ms, dicha señal tiene un ciclo de trabajo (duty cycle) de 20% (20% on y 80% off). La Figura muestra tres señales PWM con diferentes "duty cycles".

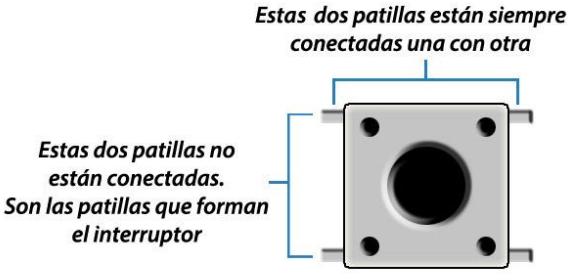


Las señales PWM son comúnmente usadas para el control de velocidad de motores DC (si se disminuye el ciclo de trabajo sobre la señal de control del circuito de potencia que actúa sobre el motor, el motor se mueve más lentamente), ajustar la intensidad de brillo de un LED, etc. En el caso del ESP32 se pueden usar un total de 16 pines digitales con la función PWM, quiere decir que tenemos un total de 15 canales. Parte de la configuración completa del PWM es definir la frecuencia de trabajo, podemos usar una frecuencia de 5000Hz para controlar el brillo de un led. La arquitectura del ESP32 también permite trabajar con resoluciones de 1 a 16 bits para el control de la salida, procedemos a usar la configuración de resolución de 8 bits. Con la configuración adecuada en el setup, procedemos a definir el pin PWM con la función `ledcAttachPin(GPIO, canal_definido)`, finalmente realizamos la activación de la función PWM con la sentencia `ledcWrite(canal, frecuencia)`.

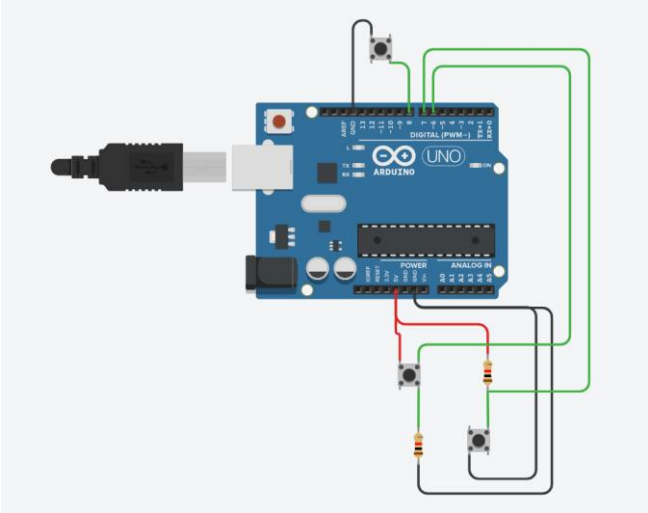
2.2 Placa de prototipos

En este apartado analizaremos los componentes que usamos en la placa de prototipos del laboratorio fabricados para la asignatura

1. **Pulsadores:** Los pulsadores son elementos que dejarán pasar la corriente eléctrica siempre y cuando se mantengan pulsados, cuando dejan de estar pulsados no dejará pasar la corriente, las conexiones internas son como se muestran en la figura:



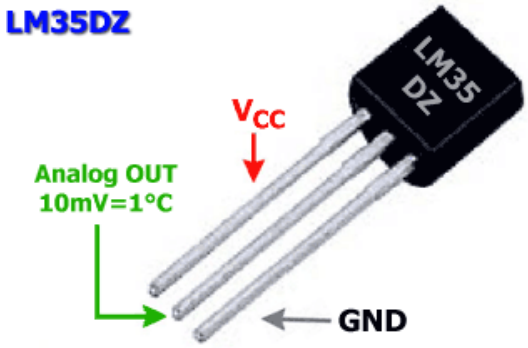
En la siguiente imagen se aprecia el cableado de los 3 tipos de configuraciones de botones: pull_up sin resistencia, pull_up y pull_down con resistencia. La imagen se ve un Arduino ya que este microcontrolador si acepta el INPUT_PULLUP en todas sus entradas digitales



2. Sensor de temperatura LM35

El sensor LM35 mide la temperatura de forma lineal generando un voltaje proporcional al valor de la temperatura y viene calibrado en grados Celsius.

Es un sensor que presenta únicamente 3 pines (VCC, GND y Data), por ello su conexión es muy sencilla



Para leer la temperatura de estos sensores, solo necesitas leer el voltaje de salida del sensor con un pin analógico de la placa Arduino y emplear la función analogRead () en tu sketch y conseguirás lecturas de temperatura con dos puntos decimales.

3. **Potenciómetro** Tenemos 3 terminales A, B y C. Si conectáramos los terminales A y C al circuito sería una resistencia Fija del valor igual al máximo de la resistencia que podría tener. Ahora bien, si conectamos los terminales A y B el valor de la resistencia dependería de la posición donde estuviera el terminal B, este potenciómetro se puede variar su resistencia conforme se rote la perilla, es por ello que se le llama potenciómetro rotatorio a la Figura

4. **Fotorresistor** Un fotorresistor, o LDR (light-dependent resistor) es un dispositivo cuya resistencia varía en función de la luz recibida. Podemos usar esta variación para medir, a través de las entradas analógicas, una estimación del nivel de luz. Por tanto, un fotorresistor disminuye su resistencia a medida que aumenta la luz sobre él. Los valores típicos son de 1 Mohm en total oscuridad, a 50-100 Ohm bajo luz brillante. La variación de la resistencia es relativamente lenta, de 20 a 100 ms en función del modelo. Esta lentitud hace que no sea posible registrar variaciones rápidas, como las producidas en fuentes de luz artificiales alimentadas por corriente alterna. Este comportamiento puede ser beneficioso, ya que dota al sensor de una gran estabilidad. Finalmente, los fotoresistores no resultan adecuados para

proporcionar una medición de la iluminancia, es decir, para servir como luxómetro Esto es debido a su baja precisión, su fuerte dependencia con la temperatura y, especialmente, a que su distribución espectral no resulta adecuada para la medición de iluminancia.