## 1.Introducción

a) ¿Cuál es la diferencia entre un pin digital y un pin analógico?

b) ¿Qué hace la función digitalPinToInterrupt?

c) ¿Para qué sirve la función map?

d) ¿Para qué sirve habilitar la conexión serial?

1. La diferencia entre un **pin digital** y un **pin analógico** es la cantidad de estados o valores que pueden representar. El pin digital puede estar en dos estados: HIGH (alto, 5V, 1 en binario) o LOW (bajo, 0V, 0 en binario). El pin analógico puede representar valores dentro de un rango. Como el conversor de análogo a digital es de 10 bits, retorna valores entre 0 y 1023.

El primero se utiliza para encender/apagar leds o leer el estado de un interruptor. El segundo se utiliza para leer señales análogas como de sensores ultrasónicos o de luz.

1. La función **digitalPinToInterrupt(pin)** traduce el pin digital actual al número de interrupción. Es utilizado como el primer parámetro de la función attachInterrupt. Por ejemplo, si hay que conectar el pin 3, se utiliza digitalPinToInterrupt(3).

Esta función permite evitar problemas de compatibilidad cuando el código se ejecuta en una placa diferente ya que los pines específicos con interrupciones y su asignación varía en cada tipo de placa.

1. La función **map(value, fromLow, fromHigh, toLow, toHigh)** permite remapear un rango de valores a otro rango. El valor de fromLow se mapea a toLow, el valor de fromHigh a toHigh y los valores intermedios a sus respectivos valores relativos. Si los rangos son distintos, la función automáticamente escala los valores del input de forma proporcional para que estén dentro del rango del output.

Parámetros:

* value: el número a mapear
* fromLow: el límite inferior del valor actual
* fromHigh: el límite superior del valor actual
* toLow: el límite inferior del valor destino
* toHigh: el límite superior del valor destino

1. Habilitar la **conexión serial** permite establecer una comunicación entre la placa Arduino con una computadora u otro dispositivo. Esto permite la transmisión de datos bit a bit. Se realiza a través del puerto serial UART (Universal Asynchronous Receiver/Transmitter) y los pines utilizados son TX (transmisión) y RX (recepción).

## 2. Uso de Leds

A screenshot of a computer program

Descripción generada automáticamente

1. Cargar el siguiente programa y ejecutarlo.
2. Explicar que realiza el programa.
3. [Opción A] Modificar el programa para que presente la siguiente secuencia de encendido de leds: 000, 001, 011, 111, 110, 100. La misma se debe repetir todo el tiempo en intervalos regulares de 0,2 segundos entre cada valor.
4. d) [Opción B] Modificar el programa para que reproduzca una secuencia ascendente de números codificados en binario desde 0 a 7. Por cada iteración completa de la cuenta, el intervalo entre cada valor se deberá aumentar en 0,5 segundos.
5. El programa primero inicializa las variables de los pines de los leds con sus valores correspondientes. En la función setup(), se establecen los pines como salida mediante la función pinMode(). En la función loop(), se encienden los leds (1 o HIGH), se esperan 1000ms, se apagan los led (0 o LOW) y se esperan 1000ms. El ciclo se repite infinitamente.

En resumen, el programa hace que los leds en los pines 4, 5 y 6 parpadeen cada 1000 milisegundos (1 segundo).

## A screenshot of a computer program Descripción generada automáticamente3.Uso de Joystick

1. Cargar el siguiente programa y ejecutarlo.
2. Explicar que realiza el programa.
3. [Opción A] Modificar el programa de forma que se enciendan los leds respetando el siguiente patrón:
4. [Opción B] Modificar el programa de forma que los leds se enciendan gradualmente desde 000 a 111 indicando la posición del Joystick, tanto del centro hacia arriba como del centro hacia abajo.
5. El programa primero inicializa las variables de los pines del joystick con sus valores correspondientes. En la función setup(), se establece la conexión serial y configura los modos de los distintos pines. En la función loop(), primero se leen los valores que ingresan por los pines VRx (valores eje X), VRy (valores eje Y) y SW (pin conectado al interruptor); luego se mapean los valores de los ejes X e Y en un rango donde podrán moverse (-512 a 512); se imprimen las posiciones de X e Y y el valor del estado de SW en el puerto serial; por último se escribe en el pin del led el estado de SW (enciende o apaga el led dependiendo del valor de SW). Establece un retraso de 10ms para evitar. El ciclo se repite infinitamente.

En resumen, el programa muestra las posiciones mapeadas de los ejes X e Y del joystick y del estado del interruptor acciones o lecturas muy rápidas.

## 4.Uso de Servo

A screenshot of a computer program

Descripción generada automáticamente

1. Cargar el siguiente programa y ejecutarlo.
2. Explicar que realiza el programa.
3. A white sheet with black text

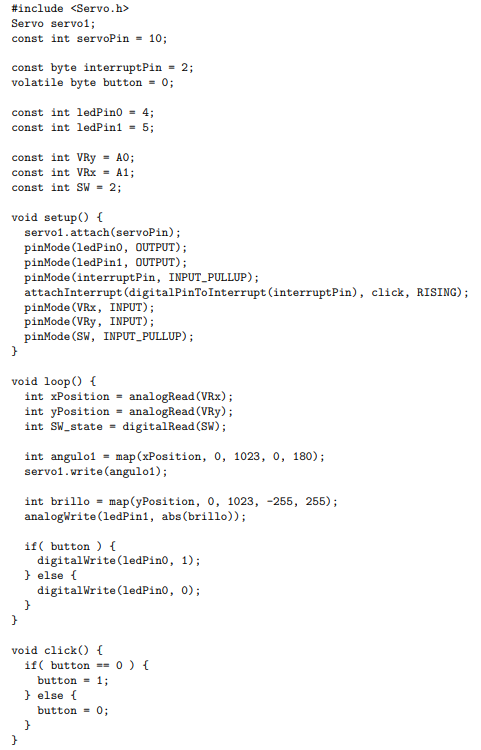
   Descripción generada automáticamente[Opción A] Modificar el programa para que el Servo reproduzca el siguiente patrón de comportamiento una y otra vez, cambiando de posición cada 2 segundos.
4. [Opción B] Modificar el programa para que el movimiento de izquierda a derecha del Joystick reproduzca el mismo comportamiento que el Servo de 0 a 180 grados.
5. El programa primero inicializa la variable del pin del servo con su valor correspondiente y la variable de tipo servo. En la función setup(), se asocia la variable de tipo servo (servo1) al pin del servo (servoPin = 10). En la función loop(), se mueve el servo a cierto ángulo y se espera 1000ms. Se mueve en los ángulos 0º, 30º, 60º, 90º y 180º. El ciclo se repite infinitamente.

En resumen, el programa modifica la posición del servo cada 1000 milisegundos (1 segundo) y se mueve de a 30º.

## 5. Interrupciones

1. Cargar el siguiente programa y ejecutarlo.
2. Explicar que realiza el programa.
3. [Opción A] Modificar el programa para que cada vez que se presiona el botón del Joystick se prendan todos los leds y estos queden prendidos. Cuando se presione nuevamente, estos se deben apagar.
4. [Opción B] Modificar el programa para que cada vez que se presiona el botón del Joystick se intercambie el sentido de rotación del Servo.

Nota: Ver la sección Problema del rebote en un botón para identificar el problema que se da en el código presentado y como solucionarlo.



1. El programa primero inicializa la variable de tipo servo, la variable del botón (interrupción) y las variables de los pines del servo, de la interrupción, de los leds y del joystick con sus valores correspondientes. En la función setup(), se asocia la variable de tipo servo al pin del servo, configura los modos de los distintos pines y asocia una función de interrupción que depende del estado del pin de interrupción. En la función loop(), primero se leen los valores que ingresan por los pines VRx (valores eje X), VRy (valores eje Y) y SW (pin conectado al interruptor). Los valores del eje X se mapean de 0º a 180º y se guardan en la variable angulo1 que luego va a ser utilizado para mover el servo en ese ángulo. Los valores del eje Y se mapean de -255 a 255m y se guardan en la variable brillo que luego va a ser utilizado para encender el led en el pin 1 en ese brillo. Luego, se evalúa una condición que, si se presiona el botón (interrupción), el led en el pin 0 se enciende. En la función click() se evalúa si el botón es presionado.

En resumen, el programa mueve el servo en el ángulo que depende del eje X del joystick, se enciende el led del pin 1 que depende del eje Y del joystick y si se presiona el botón ocurre una interrupción.

## A screenshot of a computer program Descripción generada automáticamente6. Sensor Ultrasónico

1. Cargar el siguiente programa y ejecutarlo.
2. Explicar que realiza el programa.
3. [Opción A] Modificar el programa para que los leds se enciendan desde 000 a 111 según se detecte la distancia a un objeto.
4. [Opción B] Modificar el programa para que todos los leds se enciendan cuando la distancia al objeto este entre 5cm y 10cm del sensor.
5. El programa primero inicializa las variables de los pines del sensor ultrasónico (pingPin: trigger y echoPin: echo) con sus valores correspondientes. En la función setup(), se establece la conexión serial y configura los modos de los distintos pines. En la función loop(), se espera 2ms con la señal de trigger en 0, se coloca en 1 por 10ms y luevo vuelve a 0. Es decir, es la señal de activación para el sensor. Con la función pulseIn se mide el tiempo en que tarda en llegar el eco. Con la función microsecondsToCentimeters convierte el tiempo medido a centímetros. Luego se imprimen en el puerto serial la distancia medida en centímetros. Por último, se establece un retraso de 100ms para realizar la próxima medición.

En resumen, mide la distancia entre el sensor ultrasónico y el objeto, calculando cuánto tarda en ir y volver la señal. Se imprime la distancia en el puerto serial.