

PRACTICA 2 : INTERRUPCIONES

Objetivo: realizar dos tipos de interrupciones. Para eso se dividirá la práctica en dos partes

Practica A: interrupción por GPIO, la cual se controla a través de un botón analógico

Práctica B: Interrupción por temporizador, usando un contador del microprocesador.

A: Interrupción por botón

- Header

En la cabecera se declara la tupla Button, esta estructura contiene 3 parámetros: el número del pin al cual este está conectado, el número de veces que se pulsó y el estado. Declaramos nuestro botón al pin 18 y el contador lo definimos a 0.

```
struct Button {  
    const uint8_t PIN;  
    uint32_t numberKeyPresses;  
    bool pressed;  
};  
  
Button button1 = {18, 0, false};
```

- ISR

A continuación se escribe la funcion ISR que se ejecutará cuando la interrupción ocurra. En este caso incrementa en uno el numero de pulsaciones del botón y cambia su estado a "pulsado".

```
void IRAM_ATTR isr() {  
    button1.numberKeyPresses += 1;  
    button1.pressed = true;  
}
```

- SETUP

El Setup se empieza con el begin del puerto serie para mostrar resultados por pantalla. Luego se declara el pin deseado para la funcionalidad del botón. Y por último se utiliza la funcion *attachInterrupt* para que ejecute el ISR cuando detecte una caída (pulsación en la cual el pin va de HIGH a LOW) en el pin en el cual tenemos el boton conectado.

```
void setup() {  
  
    Serial.begin(9600);  
    pinMode(button1.PIN, INPUT_PULLUP);
```

```
attachInterrupt(button1.PIN, isr, FALLING);  
  
}
```

- LOOP

Aquí se regula la pulsación del boton. Cuando este esté pulsado se imprime por pantalla el número de veces que ha sido pulsado y se configura a un estado de "no pulsado".

Finalmente se hace que el boton se desactive tras 1 minuto de inactividad.

```
void loop() {  
  
  if (button1.pressed) {  
    Serial.printf("Button 1 has been pressed %u times\n", button1.numberKeyPresses);  
    button1.pressed = false;  
  }  
  
  //Detach Interrupt after 1 Minute  
  static uint32_t lastMillis = 0;  
  if (millis() - lastMillis > 60000) {  
    lastMillis = millis();  
    detachInterrupt(button1.PIN);  
    Serial.println("Interrupt Detached!");  
  }  
}
```