Pol Naharro

PRACTICA 2 : INTERRUPCIONES PARTE A

Código Practica A:

```
#include <Arduino.h>

struct Button {
  const uint8_t PIN;
  uint32_t numberKeyPresses;
  bool pressed;
};

Button button1 = {18, 0, false};

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

void setup() {
  Serial.begin(115200);
  pinMode(button1.PIN, INPUT_PULLUP);
  attachInterrupt(button1.PIN, isr, FALLING);
}

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!");
  }
}
```

Explicación:

Practica A:

Esta práctica se divide en diferentes bloques el cual cada uno hace su función.

Antes de los bloques se realiza la siguiente instrucción que es para asignar al botón una pata de la placa. Button button1 = {18, 0, false};

El primer bloque es el void IRAM_ATTR isr() que consiste en un contador de las veces que el usuario ha pulsado el botón.

Después de esta el void setup() que con la instruccion Serial.begin(115200); nos da los datos por el monitor y las dos siguientes instrucciones nos conecta el botón y la otra nos ejecuta lo que pasa cuando se activa este.

El loop es un bucle que lo que hace es si el botón se pulsa este, gracias al Serial.printf("Button 1 has been pressed %u times\n", button1.numberKeyPresses); saca por pantalla el mensaje siguiente "Button 1 has been pressed %u times\n" donde %u es el número de la vez que se ha pulsado el botón. Después la instrucción se pone a "false" para que la próxima vez que se pulse vuelva a funcionar.

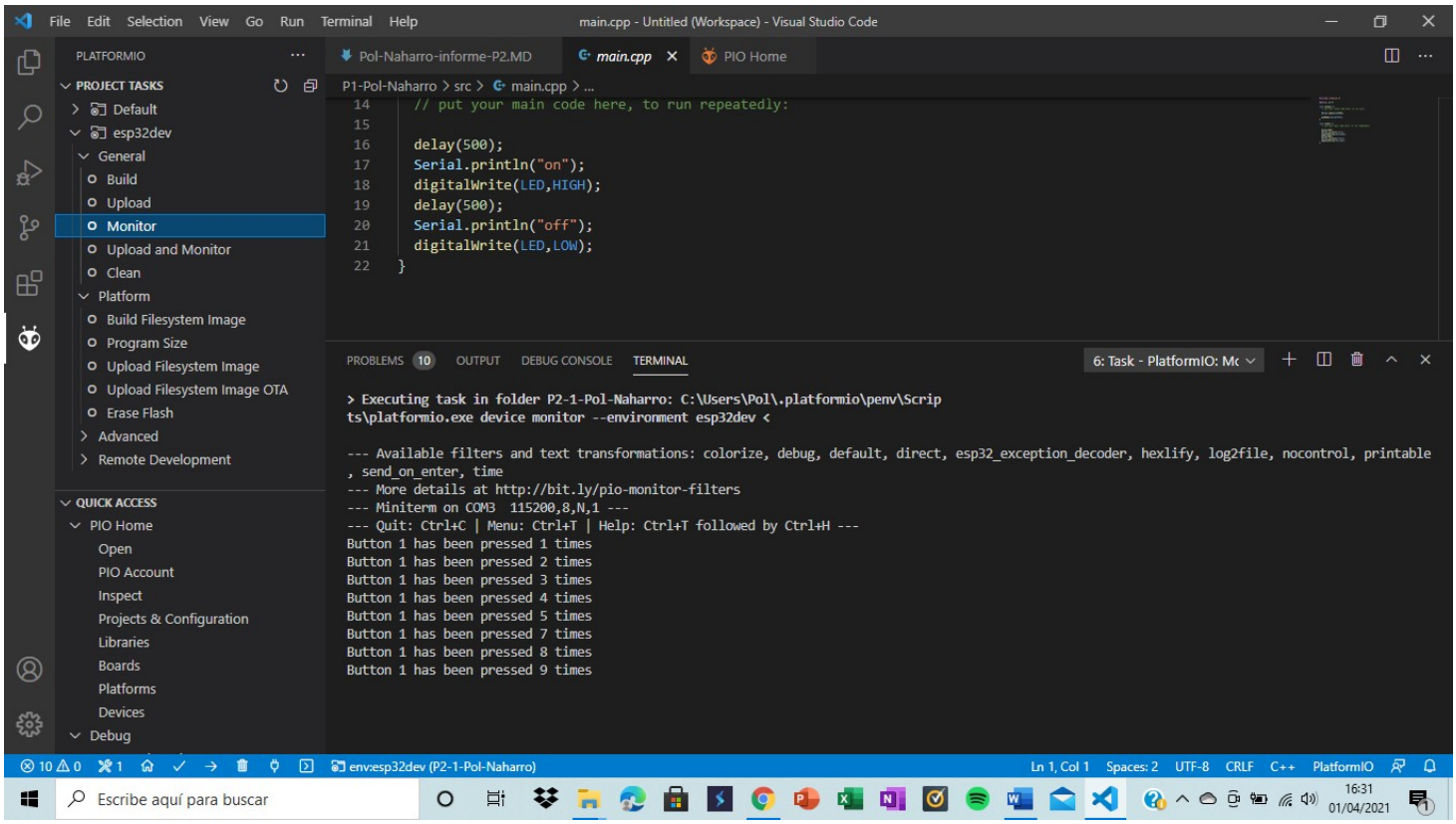
I el último apartado nos sirve para poner un final al código después de un minuto.

Entonces como resumen del programa vemos como el primer bloque es el contador, el segundo, el setup, se encarga de ejecutar e inicializar el botón, el tercero, el loop, lo que hace es enviar un mensaje cuando se ejecuta y el último interrumpe el contador al minuto.

Funcionamiento de la práctica:

Practica A

Comprobación interrupción:


File Edit Selection View Go Run Terminal Help
main.cpp - Untitled (Workspace) - Visual Studio Code
PLATFORMIO
PROJECT TASKS
Default
esp32dev
General
Build
Upload
Monitor
Upload and Monitor
Clean
Platform
Build Filesystem Image
Program Size
Upload Filesystem Image
Upload Filesystem Image OTA
Erase Flash
Advanced
Remote Development
QUICK ACCESS
PIO Home
Open
PIO Account
Inspect
Projects & Configuration
Libraries
Boards
Platforms
Devices
Debug
Pol-Naharro-informe-P2.MD
main.cpp
PIO Home
P1-Pol-Naharro > src > main.cpp > ...
14 // put your main code here, to run repeatedly:
15
16 delay(500);
17 Serial.println("on");
18 digitalWrite(LED,HIGH);
19 delay(500);
20 Serial.println("off");
21 digitalWrite(LED,LOW);
22 }
PROBLEMS 10
OUTPUT
DEBUG CONSOLE
TERMINAL
6: Task - PlatformIO: Mc
> Executing task in folder P2-1-Pol-Naharro: C:\Users\Pol\.platformio\penv\Scripts\platformio.exe device monitor --environment esp32dev <
--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at http://bit.ly/pio-monitor-filters
--- Miniterm on COM3 115200,8,N,1 ---
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
Button 1 has been pressed 1 times
Button 1 has been pressed 2 times
Button 1 has been pressed 3 times
Button 1 has been pressed 4 times
Button 1 has been pressed 5 times
Button 1 has been pressed 7 times
Button 1 has been pressed 8 times
Button 1 has been pressed 9 times
env:esp32dev (P2-1-Pol-Naharro)
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
C++
PlatformIO
Escribe aquí para buscar
16:31
01/04/2021