

```

# Codi 1ra part

#include <Arduino.h>

#define BOOT

struct Button {
  const uint8_t PIN;
  uint32_t numberKeyPresses;
  bool pressed;
};
Button button1 = {BOOT};
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!");
  }
}

```

Funcionament del programa

En el codi de la pràctica, primer caldrà assignar un pin per a la sortida on connectarem el cable pel polsador, en aquest cas amb la línia de codi 'Button button1 = {BOOT};'

Amb aquest polsador provocarem interrupcions, les quals seran efectuades en prémer el polsador. En fer-ho, i tenint el programa monitoritzat, es mostrarà per pantalla el missatge "Button 1 has been pressed %u times\n", on %u és el nombre de vegades que s'ha polsat el polsador, i cada vegada que li donem sumará 1.

Quan el polsador no es faci servir durant més d'un minut, el programa mostrarà per pantalla el missatge "Interrupt Detached!", i per tornar a posar en funcionament el programa caldrà fer un RESET.

Així es mostra per pantalla:

![image] (<https://user-images.githubusercontent.com/101355262/171416969-b64e2602-d6cd-4941-8680-7934a88d63ac.png>)

Codi 2a part

#include <Arduino.h>

```
volatile int interruptCounter;
int totalInterruptCounter;
hw_timer_t * timer = NULL;
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
void IRAM_ATTR onTimer() {
  portENTER_CRITICAL_ISR(&timerMux);
  interruptCounter++;
  portEXIT_CRITICAL_ISR(&timerMux);
}
void setup() {
  Serial.begin(115200);
  timer = timerBegin(0, 80, true);
  timerAttachInterrupt(timer, &onTimer, true);
  timerAlarmWrite(timer, 1000000, true);
  timerAlarmEnable(timer);
}
void loop() {
  if (interruptCounter > 0) {
    portENTER_CRITICAL(&timerMux);
    interruptCounter--;
    portEXIT_CRITICAL(&timerMux);
    totalInterruptCounter++;
    Serial.print("An interrupt as occurred. Total number: ");
    Serial.println(totalInterruptCounter);
  }
}
```

Funcionament del programa

La funció d'aquest programa es que cada un cert temps es produeixi una interrupció de manera automàtica sense necessitat de premer cap botó.

Primer declararem les diferents variables necessàries. Seguidament en el void setup declarem el timer, el qual determina el temps que passa entre cada interrupció.

En el loop hi ha un sumatori de manera que es conten les interrupcions que s'han produït i les mostri per pantalla amb el missatge següent: "An interrupt as occurred. Total number: //aquí ens dirà el total d'interrupcions que portem."

El programa es mostra per pantalla de la següent manera:

![image] (<https://user-images.githubusercontent.com/101355262/171418790-f0273152-399c-46b0-bc26-ff6dc38ff8c0.png>)