

# Práctica 2 Parte 2:

---

## Código:

---

```
#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);
    }
}
```

## Funcionamiento:

---

En el caso de este código, lo que hace es hacer una interrupción cada "x" tiempo, de manera que esta no será hecha manualmente sino que el programa lo hará por nosotros.

Podemos ver en la parte del loop que el código está hecho para que la interrupción sea cada segundo, y que cuando esta ocurra nos mostrará por pantalla un mensaje diciéndonos que ha sido interrumpido y el número total de interrupciones que han ocurrido mientras el programa está funcionando.

La terminal nos muestra lo siguiente:

□