

PRÀCTICA 2: INTERRUPCIONS

Introducció

L'objectiu d'aquesta pràctica és comprendre el funcionament de les interrupcions en un microcontrolador ESP32. Per a això, es realitzaran diversos experiments utilitzant interrupcions per GPIO i per temporitzador.

Fonaments Teòrics

Una interrupció és un mecanisme que permet a un sistema atendre esdeveniments externs sense necessitat d'estar monitoritzant-los constantment (polling). Quan es produeix una interrupció, el microcontrolador executa una rutina de servei d'interrupció (ISR), després de la qual torna a l'execució normal del programa.

Tipus d'interrupcions

1. **Interrupcions per esdeveniment hardware (GPIO):** Quan un pin canvia d'estat, es pot configurar perquè generi una interrupció.
2. **Interrupcions per temporitzador (Timer):** Un comptador intern del microcontrolador genera una interrupció després d'un temps determinat.

Pràctica A: Interrupció per GPIO

Descripció

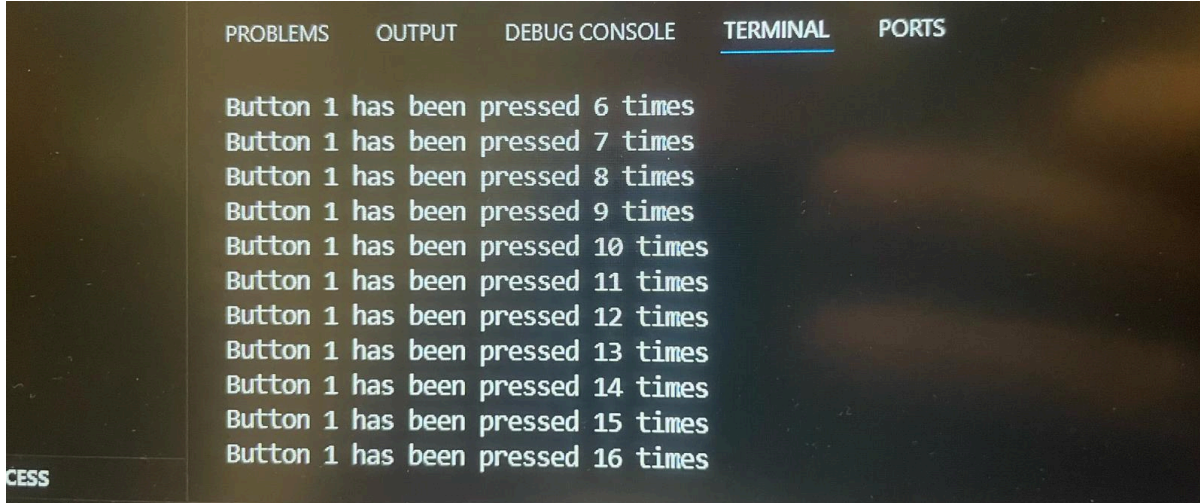
En aquesta part de la pràctica, s'utilitza un botó connectat al pin GPIO10 per generar una interrupció quan és premut. La interrupció incrementa un comptador de pulsacions i mostra el resultat per monitor sèrie.

Funcionament del codi

- Es defineix una estructura **Button** que conté el número de pin, el comptador de pulsacions i un estat de premut.
- La funció **isr()** s'executa quan es detecta la interrupció i incrementa el comptador.
- A **setup()**, es configura el pin com a entrada amb **INPUT_PULLUP** i s'associa la interrupció amb **attachInterrupt()**.
- A **loop()**, es comprova si el botó ha estat premut i es mostra per monitor sèrie.
- Passats 60 segons, es desactiva la interrupció amb **detachInterrupt()**.

Resultats obtinguts

Quan es prem el botó, es mostra al monitor sèrie un missatge amb el nombre de vegades que s'ha premut. Després de 60 segons, la interrupció es desactiva automàticament.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Button 1 has been pressed 6 times
Button 1 has been pressed 7 times
Button 1 has been pressed 8 times
Button 1 has been pressed 9 times
Button 1 has been pressed 10 times
Button 1 has been pressed 11 times
Button 1 has been pressed 12 times
Button 1 has been pressed 13 times
Button 1 has been pressed 14 times
Button 1 has been pressed 15 times
Button 1 has been pressed 16 times
```

Pràctica B: Interrupció per Timer

Descripció

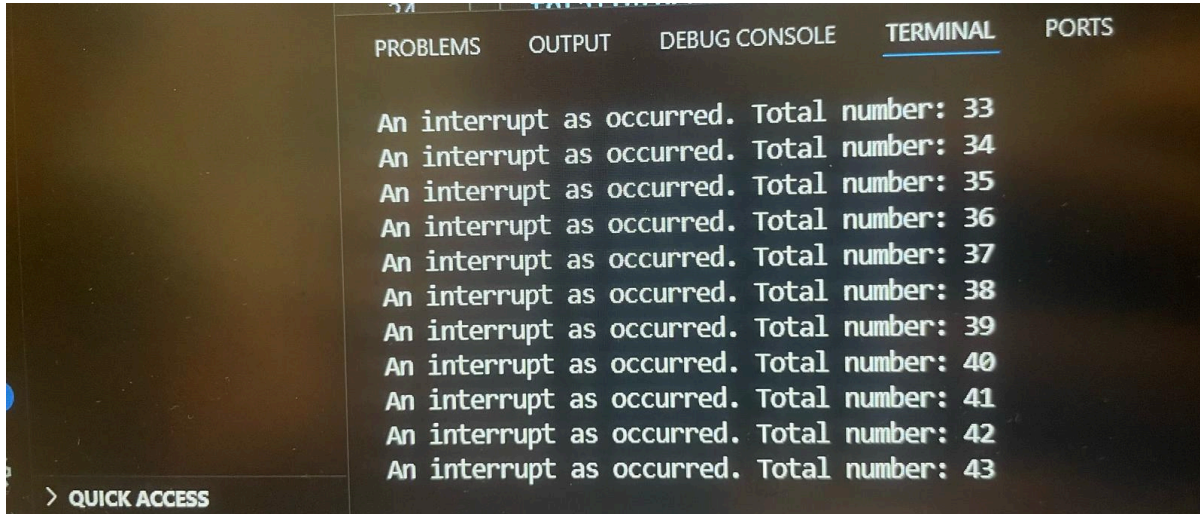
En aquesta part, es configura un temporitzador que genera una interrupció cada segon. Aquesta interrupció incrementa un comptador i mostra el nombre total d'interrupcions generades.

Funcionament del codi

- Es declara una variable ***interruptCounter*** per comptar les interrupcions.
- A ***setup()***, es configura un temporitzador amb ***timerBegin()***, es vincula la interrupció amb ***timerAttachInterrupt()*** i s'activa amb ***timerAlarmEnable()***.
- Quan el temporitzador arriba al temps definit, s'executa la funció ***onTimer()***, que incrementa ***interruptCounter***.
- A ***loop()***, si hi ha interrupcions pendents, es decrementa ***interruptCounter*** i es mostra el nombre total d'interrupcions.

Resultats obtinguts

Cada segon es mostra al monitor sèrie un missatge indicant el nombre total d'interrupcions generades pel temporitzador.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
An interrupt as occurred. Total number: 33
An interrupt as occurred. Total number: 34
An interrupt as occurred. Total number: 35
An interrupt as occurred. Total number: 36
An interrupt as occurred. Total number: 37
An interrupt as occurred. Total number: 38
An interrupt as occurred. Total number: 39
An interrupt as occurred. Total number: 40
An interrupt as occurred. Total number: 41
An interrupt as occurred. Total number: 42
An interrupt as occurred. Total number: 43
> QUICK ACCESS
```

Exercici Extra: Control de la freqüència de parpelleig d'un LED RGB

Descripció

Es controla el parpelleig d'un LED NeoPixel utilitzant dos botons. Un botó augmenta la freqüència de parpelleig, mentre que l'altre la disminueix. L'estat del LED i la freqüència actual es mostren pel monitor sèrie.

Funcionament del codi

- Es defineixen botons per augmentar i disminuir la freqüència de parpelleig.
- Una ISR detecta quan es prem un botó i ajusta el temps de parpelleig.
- El LED alterna entre encès i apagat amb l'interval establert.
- Quan es modifica la freqüència, es mostra el nou valor per monitor sèrie.

Resultats obtinguts

Quan es prem un botó, la freqüència de parpelleig del LED s'ajusta en conseqüència i es mostra un missatge amb el nou interval de temps al monitor sèrie.

Conclusions

Aquesta pràctica ha permès comprendre el funcionament de les interrupcions en un ESP32 mitjançant GPIO i temporitzadors. Les interrupcions són una eina molt útil per gestionar esdeveniments sense necessitat d'usar bucles de verificació constants, millorant l'eficiència del codi.

A més, hem observat que les interrupcions permeten una resposta immediata a esdeveniments externs, fent que el microcontrolador pugui realitzar altres tasques sense haver d'estar constantment verificant l'estat d'un pin o d'un temporitzador. Això suposa un gran avantatge en aplicacions de temps real, on una resposta ràpida és essencial.

Finalment, l'exercici extra ha demostrat com es poden combinar interrupcions per millorar la interacció amb l'usuari, ajustant la freqüència de parpelleig d'un LED en temps real. Aquest tipus de mecanismes són essencials en interfícies interactives i en dispositius de control automatitzat.