Interrupt

MICLAB-02

Név: Pilter Zsófia, Vad Avar

Dátum: 2024. 09. 16. Mérőhely: 1 (bal és jobb)

Bevezetés

Az interrupt használatának megismerése.

Ajánlott irodalom

http://www.inf.u-szeged.hu/noise/Education/MicLab/

Jegyzőkönyv készítése

A jegyzőkönyvek az órán végzett munka dokumentálására szolgálnak. A letölthető minta jegyzőkönyvet kell kiegészíteni a megfelelő információkkal: név, dátum, mérőhely (pl. 3. jobb), a feladatokhoz tartozó esetleges kifejtendő válaszokkal, valamint a kódok lényeges részével.

A jegyzőkönyveket a CooSpace-en kell feltölteni, külön pdf formátumban csatolni kell a jegyzőkönyvet (a fájl neve a következő mintát kövesse: NagyJ.KissB.03.pdf), egy külön zip fájlban pedig a kódokat (*.c, *.cwg). Amennyiben probléma merül fel a beadás során, az anyagokat az oktató e-mail címére kell elküldeni, levél tárgya legyen pl. MicLab 02.

1. feladat – LED vezérlése interrupttal I

Írjon egy programot, mely a LEDo-át 10 Hz frekvenciával villogtatja, ha a BTNo nyomógomb le van nyomva. A vezérlési feladatot a Timer 2 interrupt rutin lássa el.

A program részekre bontott forráskódja (Config, Main.c, Interrupts.c, ha van):

Config:

```
<property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"</pre>
value="Enabled"/>
   value="Start"/>
   cproperty object="TIMER16_2" propertyId="timer16.control.timerrunningstate"
value="Timer is Running"/>
   cproperty object="TIMER16_2"
propertyId="timer16.initandreloadvalue.targetoverflowfrequency" value="20"/>
   cproperty object="TIMER16 2"
propertyId="timer16.initandreloadvalue.timerreloadvalue" value="52776"/>
   cproperty object="TIMER16_2"
propertyId="timer16.reloadhighbyte.reloadhighbyte" value="206"/>
   value="40"/>
   <property object="TIMER16 3" propertyId="ABPeripheral.included" value="true"/>
   cproperty object="TIMER_SETUP_0" propertyId="ABPeripheral.included"
value="true"/>
   cproperty object="WDT_0" propertyId="ABPeripheral.included" value="true"/>
   <property object="WDT_0" propertyId="wdt.watchdogcontrol.wdtinitialvalue"</pre>
value="5"/>
   cproperty object="WDT_0" propertyId="wdt.watchdogcontrol.wdtperiodactual"
value="6.554 s"/>
 </mode>
  <modeTransition>
   cproperty object="RESET → DefaultMode"
propertyId="modeTransition.source" value="RESET"/>
   cproperty object="RESET → DefaultMode"
propertyId="modeTransition.target" value="DefaultMode"/>
 </modeTransition>
</device:XMLDevice>
Interrupts.c
#include <SI EFM8BB1 Register Enums.h>
                                                 // SFR declarations
#define ONBOARD LED P1 B4
#define ONBOARD BTN P0 B2
// This routine changes the state of the LED whenever Timer2 overflows.
SI INTERRUPT (TIMER2 ISR, TIMER2 IRQn)
{
  TMR2CN0\_TF2H = 0;
  if (!ONBOARD_BTN)
        ONBOARD_LED = !ONBOARD_LED;
  }
  else
  {
        ONBOARD LED = 1;
  }
}
Main:
main.c-ben semmi érdemleges nem volt.
Az elkészült programot be kell mutatni!
```

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- <u>Igen</u>
- Nem

2. feladat – LED állapotának billentése nyomógombbal

Írjon egy programot, mely a LEDo-át átbillenti az aktuális állapotának az ellenkezőjére, ha megnyomjuk a BTNo-át. A nyomógomb lenyomásakor a véletlenszerű, ún. "pergése" okozta többszöri detektálás elkerüléséhez pergésmentesíteni is kell a nyomógombot.

A megvalósítás részletei:

- A nyomógombot a Timer 2 használatával pergésmentesítse, mely 10 ms-os periódusidővel generál egy interruptot és léptet egy változót (extern uint8_t legyen), amivel az időt mérhetünk.
- A nyomógomb állapotát a főprogramban ellenőrizze. Amennyiben a gomb 50 ms-ig lenyomott állapotban van, úgy minősítse lenyomottnak és billentse át a LEDo állapotát az ellenkezőjére!

Tippek a pergésmentesítés algoritmusához: Ha a gombot lenyomtuk, akkor a program várakozzon 50 ms-ig (tartsa fel a főprogramot a megfelelő C vezérlési szerkezettel), billentse át a LEDo-át, majd ismét várakozzon, egészen addig, amíg a gomb lenyomott állapotban van (ezzel el lehet kerülni a LEDo folyamatos billegtetését, ha a gomb sokáig van lenyomva). Ezután ismét várjon 50 ms-ot, hogy a gomb felengedése is pergésmentesítve legyen.

A program részekre bontott forráskódja (Config, Main.c, Interrupts.c, ha van):

```
Config:
```

```
<?xml version="1.0" encoding="ASCII"?>
<device:XMLDevice xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"</pre>
xmlns:device="http://www.silabs.com/ss/hwconfig/document/device.ecore"
name="EFM8BB10F8G-A-QSOP24" partId="mcu.8051.efm8.bb1.efm8bb10f8g-a-qsop24"
version="4.0.0" contextId="%DEFAULT%">
  <mode name="DefaultMode">
    cproperty object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    cproperty object="INTERRUPT_0" propertyId="ABPeripheral.included"
value="true"/>
    cproperty object="INTERRUPT_0"
propertyId="interrupt.interruptenable.enableallinterrupts" value="Enabled"/>
    cproperty object="INTERRUPT 0"
propertyId="interrupt.interruptenable.enabletimer2interrupt" value="Enabled"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"</pre>
value="Enabled"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    cproperty object="TIMER16_2" propertyId="timer16.control.runcontrol"
value="Start"/>
    cproperty object="TIMER16_2" propertyId="timer16.control.timerrunningstate"
value="Timer is Running"/>
    cproperty object="TIMER16 2"
propertyId="timer16.initandreloadvalue.targetoverflowfrequency" value="100"/>
    cproperty object="TIMER16_2"
propertyId="timer16.initandreloadvalue.timerreloadvalue" value="62984"/>
    cproperty object="TIMER16_2"
propertyId="timer16.reloadhighbyte.reloadhighbyte" value="246"/>
```

```
<property object="TIMER16_2" propertyId="timer16.reloadlowbyte.reloadlowbyte"</pre>
value="8"/>
   cproperty object="TIMER16_3" propertyId="ABPeripheral.included" value="true"/>
   cproperty object="TIMER_SETUP_0" propertyId="ABPeripheral.included"
value="true"/>
   cproperty object="WDT_0" propertyId="ABPeripheral.included" value="true"/>
   cproperty object="WDT_0" propertyId="wdt.watchdogcontrol.wdtenable"
value="Disable"/>
   <property object="WDT_0" propertyId="wdt.watchdogcontrol.wdtperiodactual"</pre>
value="6.554 s"/>
 </mode>
 <modeTransition>
   cproperty object="RESET → DefaultMode"
propertyId="modeTransition.source" value="RESET"/>
   cproperty object="RESET → DefaultMode"
propertyId="modeTransition.target" value="DefaultMode"/>
 </modeTransition>
</device:XMLDevice>
Interrupts.c
#include <SI EFM8BB1 Register Enums.h>
#define ONBOARD LED P1 B4
#define ONBOARD BTN P0 B2
//-----
// TIMER2 ISR
//-----
// TIMER2 ISR Content goes here. Remember to clear flag bits:
// TMR2CN0::TF2H (Timer # High Byte Overflow Flag)
// TMR2CN0::TF2L (Timer # Low Byte Overflow Flag)
//-----
extern uint8 t tim;
SI INTERRUPT (TIMER2 ISR, TIMER2 IRQn)
{
     if (!ONBOARD BTN)
     {
          tim++;
     }
     else
     {
          tim = 0;
     }
}
Main.c:
#include <SI EFM8BB1 Register Enums.h>
                                               // SFR declarations
#include "InitDevice.h"
#define ONBOARD LED P1 B4
#define ONBOARD_BTN P0_B2
// $[Generated Includes]
// [Generated Includes]$
```

```
// SiLabs_Startup() Routine
// This function is called immediately after reset, before the initialization
// code is run in SILABS_STARTUP.A51 (which runs before main() ). This is a
// useful place to disable the watchdog timer, which is enable by default
// and may trigger before main() in some instances.
//----
void SiLabs_Startup (void)
 // $[SiLabs Startup]
 // [SiLabs Startup]$
}
uint8_t tim = 0;
// main() Routine
// -----
int main (void)
  // Call hardware initialization routine
 enter_DefaultMode_from_RESET();
 ONBOARD_LED = 1;
 while (1)
  {
       if (tim >= 5)
             while (!ONBOARD_BTN);
             ONBOARD_LED = !ONBOARD_LED;
   // $[Generated Run-time code]
   // [Generated Run-time code]$
}
```

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- <u>Igen</u>
- Nem

Szorgalmi feladat – LED vezérlése interrupttal II

Írjon egy programot, mely a LEDo-ás LED-et 5 Hz, 10 Hz és 50 Hz frekvenciával villogtatja. A vezérlési feladatot a Timer o interrupt rutin lássa el, a villogási frekvenciák között a BTNo-ás nyomógomb segítségével lehessen váltani. A nyomógombot pergés mentesíteni is szükséges.

Segítség: Mivel a Timero 8 bites, ezért a kívánt frekvencia elérésére önmagában nem képes, egy timer interrupt rutinban megvalósított számláló bevezetésével érhető el a kívánt frekvencia.

A program részekre bontott forráskódja (Config, Main.c, Interrupts.c, ha van):

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

Megjegyzések