PCA – PWM out

MICLAB

Név: Pilter Zsófia, Vad Avar

Dátum: 2024.11.18.

Mérőhely: 1 jobb és bal

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 03.

1. feladat – PCA konfigurálása, LED vezérlése PWM módban

A Port I/O Mapping ablakban engedélyezze a PCA0_CEX0 kimenetet. Pin-skipp segítségével tolja el a PCA0_CEX0-át a P1.4-be. Konfigurálja be a PCA perifériát 8 bites módban. Az órajelét a Timer 0 overflow adja. A Cannel 0 legyen Edge-aliagned PWM módban.

Készítsen egy programot, ami a PCA segítségével 1 másodperces periódusidővel villogtatja az EFM8BB1LCK panelen lévő LED0-át 25%-os kitöltési tényezővel. Ehhez először a Timer 0-át konfigurálja be, majd a programban állítsa be PCA capture/compare regiszterét a megfelelő értékre.

Tipp: az 1 s-os periódusidőt el lehet érni a megfelelő beállításokkal a konfigurátorban.

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"
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">
    <property object="CROSSBAR0" propertyId="xbar0.pca0.cex" value="CEX0"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="P0.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.3" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.4" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.5" propertyId="ports.settings.skip" value="Skipped"/>
```

```
    <property object="P0.6" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.7" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.3" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
    <property object="P1.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="PCACH_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannel.capturecomparehighbyte" value="65"/>
    <property object="PCACH_0" propertyId="pcach.pcachannel.capturecomparelowbyte"
value="65"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelcomparatorfunction"
value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelpulsewidthmodulationmo
den" value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcacontrol.channelcapturecomparemode" value="Predefined 8~11-bit
pulse modulator"/>
    <property object="PCACH_1" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequency" value="256.000
Hz"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequencyintegervalue"
value="256"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockperiod" value="3.906 mS"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.selectpcacountertimerpulse"
value="Timer 0 overflow"/>
    <property object="PCA_0" propertyId="pca.pcacountertimerruncontrol.pcarun"
value="Start"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER01_0"
propertyId="timer01.timer0highbyte.timer0highbyte" value="7"/>
    <property object="TIMER01_0"
propertyId="timer01.timer0mode2:8bitcountertimerwithautoreload.targetoverflowfrequ
ency" value="256"/>
    <property object="TIMER01_0"
propertyId="timer01.timer0mode2:8bitcountertimerwithautoreload.timerreloadvalue"
value="7"/>
    <property object="TIMER16_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER16_3" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER_SETUP_0" propertyId="ABPeripheral.included"
value="true"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.clockcontrol0.timer01prescale" value="SYSCLK / 48"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer0.mode"
value="Mode 2, 8-bit Counter/Timer with Auto-Reload"/>
```

```
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer0.timerrunningstate" value="Timer is Running"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer0.timerswitch1:runcontrol" value="Start"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer01control.timer0runcontrol" value="Start"/>
    <property object="WDT_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="WDT_0" propertyId="wdt.watchdogcontrol.wdtenable"
value="Disable"/>
  </mode>
  <modeTransition>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.source" value="RESET"/>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.target" value="DefaultMode"/>
  </modeTransition>
</device:XMLDevice>
```

Main.c:

```
//=========================================================
// src/feladat11-01_main.c: generated by Hardware Configurator
//
// This file will be updated when saving a document.
// leave the sections inside the "$[...]" comment tags alone
// or they will be overwritten!!
//=========================================================

//-------------------------------------------------------------------------
// Includes
//-------------------------------------------------------------------------
#include <SI_EFM8BB1_Register_Enums.h>                  // SFR declarations
#include "InitDevice.h"
// $[Generated Includes]
// [Generated Includes]$

//-------------------------------------------------------------------------
// 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)
{
  // $[SiLabs Startup]
  // [SiLabs Startup]$
}

//-------------------------------------------------------------------------
// main() Routine
// ------------------------------------------------------------------------
int main (void)
{
  // Call hardware initialization routine
  enter_DefaultMode_from_RESET();

  while (1)
  {
```

```
    // $[Generated Run-time code]
    // [Generated Run-time code]$
  }
}
```

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

2. feladat – LED fényerejének változtatása nyomógombbal

A kiegészítő panelen lévő SW1 nyomógombbal 0%-os kitöltési tényezőtől 75%-os kitöltési tényezőig 4 lépésben növelje a LED0 fényerejét gombnyomásra. Az utolsó utáni lépés után ismét az első következzen. A PWM frekvenciája legyen 100 Hz feletti. Amig az SW1 le van nyomva, addig folyamatosan léptesse a fényerőt egy timer interruptban. A 4 lépés megtételéhez szükséges idő legyen 1 másodperc.

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"
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">
    <property object="CROSSBAR0" propertyId="xbar0.pca0.cex" value="CEX0"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="INTERRUPT_0" propertyId="ABPeripheral.included"
value="true"/>
    <property object="INTERRUPT_0"
propertyId="interrupt.interruptenable.enableallinterrupts" value="Enabled"/>
    <property object="INTERRUPT_0"
propertyId="interrupt.interruptenable.enabletimer2interrupt" value="Enabled"/>
    <property object="P0.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.3" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.4" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.5" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.6" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.7" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.3" propertyId="ports.settings.skip" value="Skipped"/>
```

```
    <property object="P1.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
    <property object="P1.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="PCACH_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannel.capturecomparehighbyte" value="1"/>
    <property object="PCACH_0" propertyId="pcach.pcachannel.capturecomparelowbyte"
value="1"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelcomparatorfunction"
value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelpulsewidthmodulationmo
den" value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcacontrol.channelcapturecomparemode" value="Predefined 8~11-bit
pulse modulator"/>
    <property object="PCACH_1" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequency" value="255.208
kHz"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequencyintegervalue"
value="255208.333333"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockperiod" value="3.918 uS"/>
    <property object="PCA_0" propertyId="pca.pcacountertimerruncontrol.pcarun"
value="Start"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER16_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER16_2" propertyId="timer16.control.runcontrol"
value="Start"/>
    <property object="TIMER16_2" propertyId="timer16.control.timerrunningstate"
value="Timer is Running"/>
    <property object="TIMER16_2"
propertyId="timer16.initandreloadvalue.targetoverflowfrequency" value="4"/>
    <property object="TIMER16_2"
propertyId="timer16.initandreloadvalue.timerreloadvalue" value="1734"/>
    <property object="TIMER16_2"
propertyId="timer16.reloadhighbyte.reloadhighbyte" value="6"/>
    <property object="TIMER16_2" propertyId="timer16.reloadlowbyte.reloadlowbyte"
value="198"/>
    <property object="TIMER16_3" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER_SETUP_0" propertyId="ABPeripheral.included"
value="true"/>
    <property object="WDT_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="WDT_0" propertyId="wdt.watchdogcontrol.wdtenable"
value="Disable"/>
  </mode>
  <modeTransition>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.source" value="RESET"/>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.target" value="DefaultMode"/>
```

```
  </modeTransition>
</device:XMLDevice>
```

Main.c:

```
//=========================================================
// src/feladat11-02_main.c: generated by Hardware Configurator
//
// This file will be updated when saving a document.
// leave the sections inside the "$[...]" comment tags alone
// or they will be overwritten!!
//=========================================================

//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------
#include <SI_EFM8BB1_Register_Enums.h>                  // SFR declarations
#include "InitDevice.h"
// $[Generated Includes]
// [Generated Includes]$

#define SW1_BTN P0_B1


//-----------------------------------------------------------------------------
// 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)
{
  // $[SiLabs Startup]
  // [SiLabs Startup]$
}

//-----------------------------------------------------------------------------
// main() Routine
// ----------------------------------------------------------------------------
int main (void)
{
  // Call hardware initialization routine
  enter_DefaultMode_from_RESET();

  while (1)
  {
    // $[Generated Run-time code]
    // [Generated Run-time code]$

  }
}
```

Interrupts.c:

```
//=========================================================
// src/Interrupts.c: generated by Hardware Configurator
```

```
//
// This file will be regenerated when saving a document.
// leave the sections inside the "$[...]" comment tags alone
// or they will be overwritten!
//=========================================================

// USER INCLUDES
#include <SI_EFM8BB1_Register_Enums.h>
#define SW1_BTN P0_B1
#define STATE1 64u
#define STATE2 128u
#define STATE3 191u
#define STATE4 1u

//-----------------------------------------------------------------------------
// 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)
//
//-----------------------------------------------------------------------------

uint8_t pca_register;
SI_INTERRUPT (TIMER2_ISR, TIMER2_IRQn)
{
  if (!SW1_BTN)
    {
      pca_register = PCA0CPH0;
      switch (pca_register)
        {
        case STATE4:
          PCA0CPH0 = STATE1;
          break;
        case STATE1:
          PCA0CPH0 = STATE2;
          break;
        case STATE2:
          PCA0CPH0 = STATE3;
          break;
        case STATE3:
          PCA0CPH0 = STATE4;
          break;
        }
    }
}
```

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

3. feladat – LED fényerejének állítása potenciométerrel

Mérje folyamatosan az ADC-vel a kiegészítő panelen lévő potenciométert interrupt módban és a mért értékkel arányosan változtassa a LED0 kitöltési tényezőjét 0 ás ~100% között. A PWM frekvenciája legyen 100 Hz feletti. Az ADC kód legyen 8 bites és legyen balra igazítva. a Voltage Reference legyen 3,3 V. Elég a felső 8 bitet kiolvasni (ADC0H regiszter).

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"
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">
    <property object="ADC_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="ADC_0"
propertyId="adc.accumulatorconfiguration.accumulatorshiftandjustify" value="Left
justified"/>
    <property object="ADC_0" propertyId="adc.configuration.gaincontrol" value="1x
gain"/>
    <property object="ADC_0" propertyId="adc.configuration.sarclockdivider"
value="0"/>
    <property object="ADC_0" propertyId="adc.control.enableadc" value="Enabled"/>
    <property object="ADC_0" propertyId="adc.control.startofconversion"
value="Timer 2 overflow"/>
    <property object="ADC_0"
propertyId="adc.multiplexerselection.positiveinputselection" value="ADC0.15
(P1.7)"/>
    <property object="ADC_0" propertyId="adc.view.view" value="Advanced"/>
    <property object="CROSSBAR0" propertyId="xbar0.pca0.cex" value="CEX0"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="INTERRUPT_0" propertyId="ABPeripheral.included"
value="true"/>
    <property object="INTERRUPT_0"
propertyId="interrupt.extendedinterruptenable1.enableadc0conversioncompleteinterru
pt" value="Enabled"/>
    <property object="INTERRUPT_0"
propertyId="interrupt.interruptenable.enableallinterrupts" value="Enabled"/>
    <property object="P0.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.3" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.4" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.5" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.6" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P0.7" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.0" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.1" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.2" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="P1.3" propertyId="ports.settings.skip" value="Skipped"/>
```

```
    <property object="P1.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
    <property object="P1.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="P1.7" propertyId="ports.settings.inputmode" value="Analog"/>
    <property object="P1.7" propertyId="ports.settings.iomode" value="Analog
I/O"/>
    <property object="P1.7" propertyId="ports.settings.skip" value="Skipped"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="PCACH_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelcomparatorfunction"
value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcachannelcapturecomparemode.enablechannelpulsewidthmodulationmo
den" value="Enabled"/>
    <property object="PCACH_0"
propertyId="pcach.pcacontrol.channelcapturecomparemode" value="Predefined 8~11-bit
pulse modulator"/>
    <property object="PCACH_1" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCACH_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequency" value="255.208
kHz"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockfrequencyintegervalue"
value="255208.333333"/>
    <property object="PCA_0"
propertyId="pca.pcacountertimerconfiguration.pcaclockperiod" value="3.918 uS"/>
    <property object="PCA_0" propertyId="pca.pcacountertimerruncontrol.pcarun"
value="Start"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER16_2" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER16_2" propertyId="timer16.control.runcontrol"
value="Start"/>
    <property object="TIMER16_2" propertyId="timer16.control.timerrunningstate"
value="Timer is Running"/>
    <property object="TIMER16_2"
propertyId="timer16.initandreloadvalue.targetoverflowfrequency" value="4"/>
    <property object="TIMER16_2"
propertyId="timer16.initandreloadvalue.timerreloadvalue" value="1734"/>
    <property object="TIMER16_2"
propertyId="timer16.reloadhighbyte.reloadhighbyte" value="6"/>
    <property object="TIMER16_2" propertyId="timer16.reloadlowbyte.reloadlowbyte"
value="198"/>
    <property object="TIMER16_3" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER_SETUP_0" propertyId="ABPeripheral.included"
value="true"/>
    <property object="VREF_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="VREF_0" propertyId="vref.hidden.voltagereferenceselect"
value="VDD pin"/>
    <property object="VREF_0"
propertyId="vref.voltagereferencecontrol.selectvoltagereference"
value="Unregulated VDD"/>
    <property object="WDT_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="WDT_0" propertyId="wdt.watchdogcontrol.wdtenable"
value="Disable"/>
```

```
  </mode>
  <modeTransition>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.source" value="RESET"/>
    <property object="RESET &#x2192; DefaultMode"
propertyId="modeTransition.target" value="DefaultMode"/>
  </modeTransition>
</device:XMLDevice>
```

Main.c:

```
//=========================================================
// src/feladat11-03_main.c: generated by Hardware Configurator
//
// This file will be updated when saving a document.
// leave the sections inside the "$[...]" comment tags alone
// or they will be overwritten!!
//=========================================================

//-------------------------------------------------------------------------------
// Includes
//-------------------------------------------------------------------------------
#include <SI_EFM8BB1_Register_Enums.h>                  // SFR declarations
#include "InitDevice.h"
// $[Generated Includes]
// [Generated Includes]$

//-------------------------------------------------------------------------------
// 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)
{
  // $[SiLabs Startup]
  // [SiLabs Startup]$
}

//-------------------------------------------------------------------------------
// main() Routine
// ------------------------------------------------------------------------------
int main (void)
{
  // Call hardware initialization routine
  enter_DefaultMode_from_RESET();

  while (1)
  {
    // $[Generated Run-time code]
    // [Generated Run-time code]$
  }
}
```

Interrupts.c:

```
//=========================================================
// src/Interrupts.c: generated by Hardware Configurator
//
```

```
// This file will be regenerated when saving a document.
// leave the sections inside the "$[...]" comment tags alone
// or they will be overwritten!
//=========================================================

// USER INCLUDES
#include <SI_EFM8BB1_Register_Enums.h>

//-----------------------------------------------------------------------------
// ADC0EOC_ISR
//-----------------------------------------------------------------------------
//
// ADC0EOC ISR Content goes here. Remember to clear flag bits:
// ADC0CN0::ADINT (Conversion Complete Interrupt Flag)
//
//-----------------------------------------------------------------------------
SI_INTERRUPT (ADC0EOC_ISR, ADC0EOC_IRQn)
  {
   PCA0CPH0 = ADC0H;
  }
```

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

Megjegyzések

A második feladatban a config elvileg helyes, mégis azt tapasztaltuk, hogy nem azt tapasztaltuk, amit akartunk.