Kommunikáció - UART

MICLAB-08

Név: Pilter Zsófia, Vad Avar

Dátum: 2024.11.11.

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 – Karakter küldése és fogadása polling módban

Kösse össze a két mikrovezérlő kit UART RX és TX vonalait, valamint a GND-t (ügyeljen arra, hogy kimenetet csak bemenettel kössön össze). A baud rate legyen 115200 bit/s (a SYSCLK legyen 24,5 MHz, a Timer 1 Clock Source beállítása a Use SYSCLK legyen). A páros egyik tagja készítsen egy olyan programot, ami a BTN0 nyomógomb lenyomására elküld egy „L” karaktert UART-on polling módban. A páros másik tagja pedig folyamatosan fogadja az UART-on beérkező adatot és ha a „L” karakter érkezik be, akkor kapcsolja be az EFM8 kiten lévő LED0-át.

Küldés programja

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="CLOCK_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="CLOCK_0" propertyId="clock.clockselect.clocksourcedivider"
value="SYSCLK / 1"/>
    <property object="CLOCK_0" propertyId="clock.clockselect.sysclk" value="24.500
MHz"/>
    <property object="CROSSBAR0" propertyId="xbar0.uart0.data" value="Enabled"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="P0.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
```

```
    <property object="P0.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1highbyte.timer1highbyte" value="150"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.targetoverflowfrequ
ency" value="230400"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.timerreloadvalue"
value="150"/>
    <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.timer01control.timer1runcontrol" value="Start"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.clocksource"
value="Use SYSCLK"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.mode"
value="Mode 2, 8-bit Counter/Timer with Auto-Reload"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerrunningstate" value="Timer is Running"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerswitch1:runcontrol" value="Start"/>
    <property object="UART_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="UART_0" propertyId="uart.serialportcontrol.enablereceive"
value="Enabled"/>
    <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/myProject_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
```

```c
//-----------------------------------------------------------------------------
#include <SI_EFM8BB1_Register_Enums.h>                  // SFR declarations
#include "InitDevice.h"

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

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

  while (1)
  {
      if (!ONBOARD_BTN)
        {
          SBUF0 = 'L';

          while (!SCON0_TI);
          SCON0_TI = 0;
         }

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

Fogadás programja

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="CLOCK_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="CLOCK_0" propertyId="clock.clockselect.clocksourcedivider"
value="SYSCLK / 1"/>
    <property object="CLOCK_0" propertyId="clock.clockselect.sysclk" value="24.500
MHz"/>
    <property object="CROSSBAR0" propertyId="xbar0.uart0.data" value="Enabled"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="P0.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
    <property object="P0.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1highbyte.timer1highbyte" value="150"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.targetoverflowfrequ
ency" value="230400"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.timerreloadvalue"
value="150"/>
    <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.timer01control.timer1runcontrol" value="Start"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.clocksource"
value="Use SYSCLK"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.mode"
value="Mode 2, 8-bit Counter/Timer with Auto-Reload"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerrunningstate" value="Timer is Running"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerswitch1:runcontrol" value="Start"/>
    <property object="UART_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="UART_0" propertyId="uart.serialportcontrol.enablereceive"
value="Enabled"/>
    <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/feladat_10_01_receive_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"

#define ONBOARD_LED P1_B4
#define LED_ON 0U
#define LED_OFF 1U
// $[Generated Includes]
// [Generated Includes]$

//-----------------------------------------------------------------------------
// 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 sent = 0;

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

  while (1)
  {

      while(!SCON0_RI);
      SCON0_RI = 0;

      sent = SBUF0;

      if (sent == 'L')
        {
          ONBOARD_LED = LED_ON;
```

```
        }

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

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

2. feladat –4 jegyű szám küldése PC-re UART-on

A mikrovezérlővel az USB kábelen keresztül UART üzeneteket lehet küldeni a PC-nek (fogadni is lehet PC-től). Ehhez egy segédprogramot kell használni a PC-n, aminek a leírása a feladat végén található.

A SYSCLK legyen a maximális 24,5 MHz. A baud rate legyen közelítőleg 115200 bit/s.

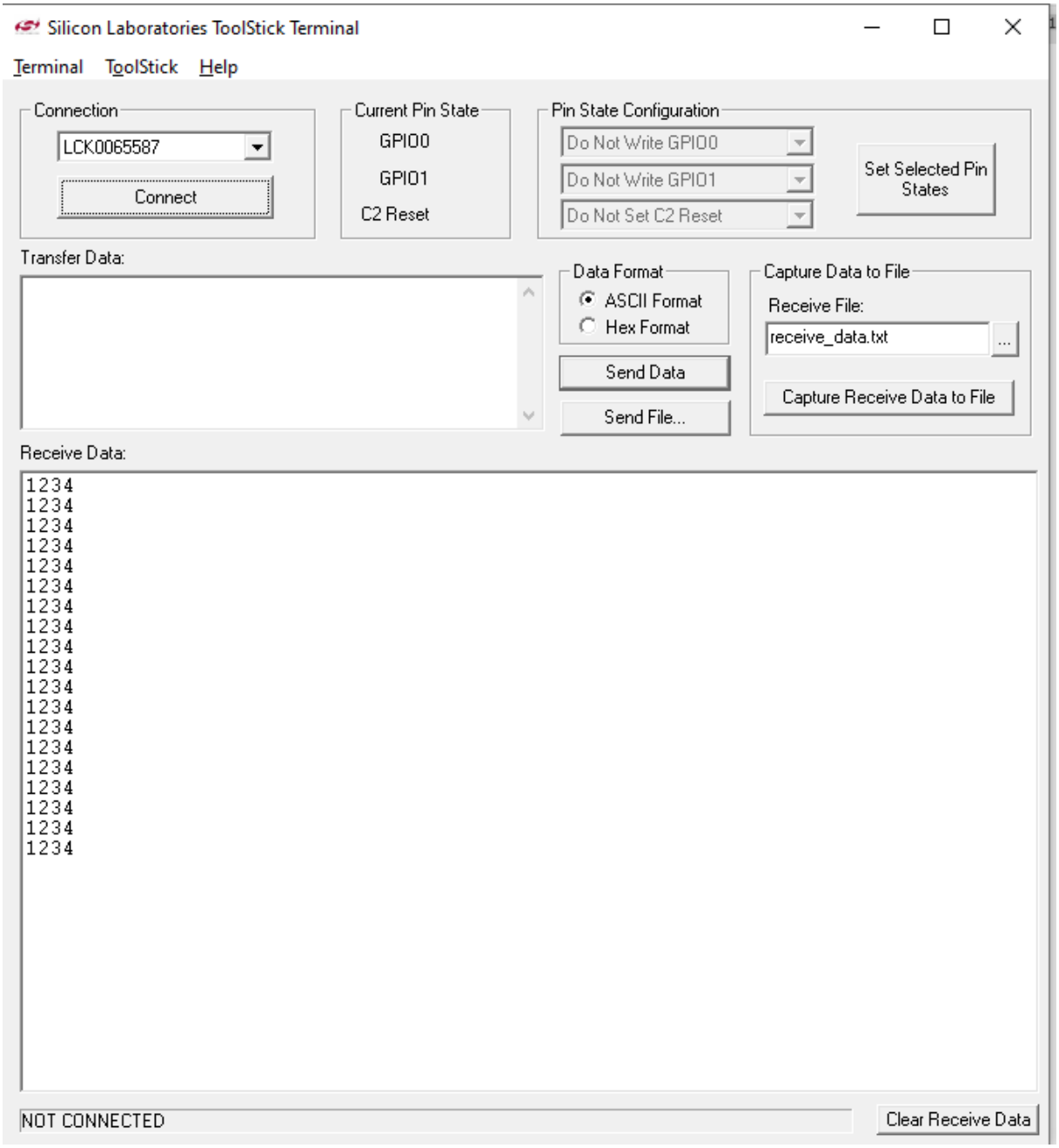
Az UART-ot polling módban használja a kódban, amihez írjon egy saját függvényt. Az „sprintf” függvény segítségével konvertálja át az 1234 négyjegyű számot stringbe. A kapott stringet és egy ’\n’ karaktert egy megfelelő méretű tömbben tároljon el.

Ezután a karakter tömb elemeit egyesével ki lehet küldeni az UART-on. Másodpercenként egyszeri ismétléssel küldje a négyjegyű számot.

Tipp: a sorvégi ’\0’ karaktert figyelje az tömb indexelésnél, azt már nem kell kiküldeni.

A bejövő adatokat a PC-n a ToolStick Terminal programmal lehet megjeleníteni. A programot nagyon egyszerű használni: csak a „Connect” gombra kell kattintanunk az előlapon. Fontos, hogy csak akkor tudunk kapcsolódni, ha nem debug módban fut a kódunk, azaz a feltöltés után állítsuk le a debugolást, ekkor újraindul a kódunk, de már nem debug módban. A mikrovezérlőtől fogadott üzenet a „Receive Data:” ablakban jelenik meg.

Készítsen képernyőképet a ToolStick Terminalról működés közben.



1. ábra: Saját képernyőkép a ToolStick Terminalról.

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="CLOCK_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="CLOCK_0" propertyId="clock.clockselect.clocksourcedivider"
value="SYSCLK / 1"/>
```

```
    <property object="CLOCK_0" propertyId="clock.clockselect.sysclk" value="24.500
MHz"/>
    <property object="CROSSBAR0" propertyId="xbar0.uart0.data" value="Enabled"/>
    <property object="DefaultMode" propertyId="mode.diagramLocation" value="100,
100"/>
    <property object="P0.4" propertyId="ports.settings.iomode" value="Digital
Push-Pull Output"/>
    <property object="P0.4" propertyId="ports.settings.outputmode" value="Push-
pull"/>
    <property object="PBCFG_0" propertyId="pbcfg.settings.enablecrossbar"
value="Enabled"/>
    <property object="TIMER01_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="TIMER01_0"
propertyId="timer01.timer0mode2:8bitcountertimerwithautoreload.targetoverflowfrequ
ency" value="10"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1highbyte.timer1highbyte" value="150"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.targetoverflowfrequ
ency" value="230400"/>
    <property object="TIMER01_0"
propertyId="timer01.timer1mode2:8bitcountertimerwithautoreload.timerreloadvalue"
value="150"/>
    <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="1000"/>
    <property object="TIMER16_2"
propertyId="timer16.initandreloadvalue.timerreloadvalue" value="63494"/>
    <property object="TIMER16_2"
propertyId="timer16.reloadhighbyte.reloadhighbyte" value="248"/>
    <property object="TIMER16_2" propertyId="timer16.reloadlowbyte.reloadlowbyte"
value="6"/>
    <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.timer01control.timer1runcontrol" value="Start"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.clocksource"
value="Use SYSCLK"/>
    <property object="TIMER_SETUP_0" propertyId="timer_setup.timer1.mode"
value="Mode 2, 8-bit Counter/Timer with Auto-Reload"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerrunningstate" value="Timer is Running"/>
    <property object="TIMER_SETUP_0"
propertyId="timer_setup.timer1.timerswitch1:runcontrol" value="Start"/>
    <property object="UART_0" propertyId="ABPeripheral.included" value="true"/>
    <property object="UART_0" propertyId="uart.serialportcontrol.enablereceive"
value="Enabled"/>
    <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/feladat_10_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"
#include  <stdio.h>

#define ONBOARD_BTN P0_B2
// $[Generated Includes]
// [Generated Includes]$

uint16_t number = 1234u;
uint16_t count = 0;
char buffer[6];

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

void
UART0_Send_String (char *str)
{
  uint8_t inc = 0;
  while (buffer[inc] != '\0')
    {
      SBUF0 = buffer[inc];
      while (!SCON0_TI);
      SCON0_TI = 0;
```

```
      inc++;
    }
}

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

  SCON0_TI = 1;
  while (1)
  {
      while (!TMR2CN0_TF2H);
      TMR2CN0_TF2H = 0;
      count++;

      if (count > 1000)
      {
          count = 0;
          sprintf (buffer, "%d\n", number);
          UART0_Send_String (buffer);
      }

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

Az elkészült programot be kell mutatni!

A gyakorlatvezető ellenőrizte:

- Igen
- Nem

A program működött:

- Igen
- Nem

Szorgalmi feladat – Karakter küldése és fogadása interrupt módban

Készítsen egy programot, ami UART-on elküld egy 'L' karaktert, ha a BTN0 nyomógomb le van nyomva, ha nincs lenyomva, akkor a 'o' karaktert küldje. A küldés mellett fogadja is a program az UART-on beérkező adatot és az 'L' karakter beérkezése esetén kapcsolja be a LED0 LED-et, a 'o' karakter beérkezése esetén pedig kapcsolja le a LED-et. A küldés és a fogadás is interrupt módban történjen. A páros mindkét tagjánál ugyanaz a kód fusson.

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