Kommunikáció - UART

MicLab-08

Név:

Dátum:

Mérőhely:

## 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):

**Fogadás programja**

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

***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.

**2. ábra: Saját képernyőkép a ToolStick Terminalról.**

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

***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 ’0’ 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 ’0’ 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