Mérési jegyzőkönyv a
Beágyazott operációs rendszerek mérés

című laboratóriumi gyakorlatról

A mérés helyszíne:

A mérés időpontja:

A mérést végezték:

A mérést vezető oktató neve:

A jegyzőkönyvet tartalmazó fájl neve:

Felhasznált eszközök:

Eszköz megnevezése	Az eszköz típusa	Azonosító száma
AVR Atmega128	mikrovezérlő	

Az elvégzett mérési feladatok

3.1. Regiszterek elérése

Írja meg a „Mérés laboratórium 2” során assemblyben kiadott „futófény” feladatot C nyelven az **1_IO_registers.c** váz felhasználásával! (A LED-soron „vándoroljon” egy LED folyamatosan valamelyik irányba, szemmel még követhető sebességgel. Ha a fény „kilépett” a LED-sor egyik végén, „jöjjön vissza” az indulási oldalon. Ne legyen olyan állapot, amikor egyik LED sem ég!)
Késleltetésre használja az **<util/delay.h>** által biztosított függvényeket!
Ha ezzel elkészült, egészítse ki a feladatot úgy, hogy az INT gomb lenyomására álljon meg, majd felengedésére induljon el ismét a futófény!

```
int main()
{
   PORTC = 1;
   DDRC = 0xff;
   int i= 0;
   while(1)
   {
      for(i=0; i<50; i++)_delay_ms(20);
      if(PORTC == 0x80)
         PORTC = 1;
      else
         PORTC <<= 1;
   }
   return 0;
}
```

3.2. Megszakítások

„LED kapcsolgatás”: írjon egy olyan programot, amely az INT gomb lenyomására kigyújt egy
LED-et, majd ismételt lenyomására a LED elalszik. A feladatot megszakítás segítségével készítse el
felhasználva a **2_ISR.c fájlt**!

```
int gomb;


ISR(INT4_vect)
{
   if(gomb)
   {
      PORTC = 1;
      gomb=0;
   }
   else
   {
      PORTC = 0;
      gomb=1;
   }
}

int main()
{
   gomb = 0;
   EICRB = 2;
   EIMSK = 0x10;
   sei();
   DDRC = 0xff;


   return 0;
}
```

3.3. Stdio műveletek

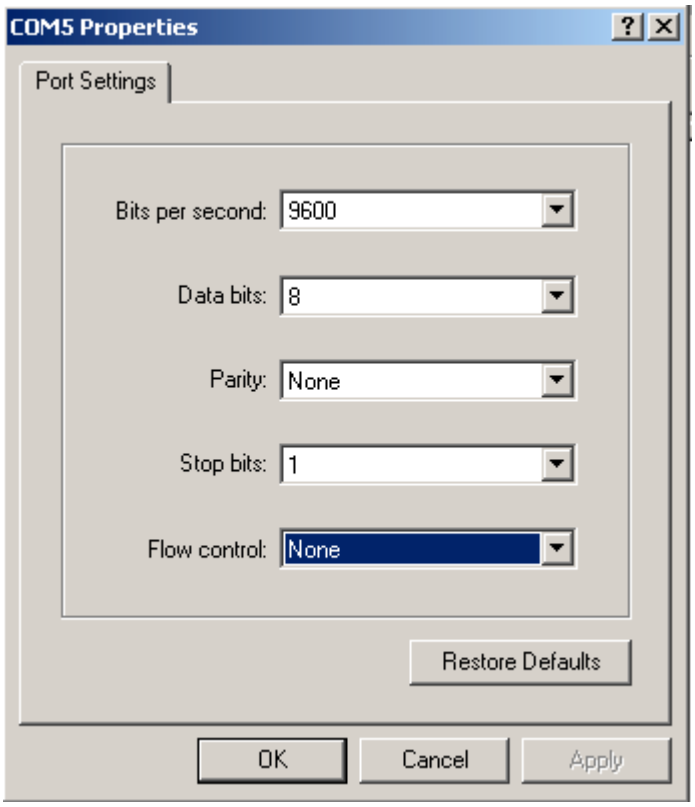
„Írógép”: írjon programot, amely a számítógépen futó soros porti terminál ablakába gépelt karaktereket megjeleníti a mérýpanel LCD képernyýjén! A megvalósításhoz használja a C standard I/O függvényeit, és az LCD valamint a soros port kezeléséhez adott API-t **(<board/lcd.h>**, **<board/serial.h>**) továbbá a **3_stdio.c** vázat! (A PC-n futó terminál beállításai: jelváltási sebesség 9600 baud, adatbitek száma 8, paritás nincs, stopbitek száma 1 és áramlásszabályozás szintén nincs.)

Először leteszteltük a kijelző működését printf-el

```
int main()
{
   LCD_init();
   serial_init();
   stdin = &serial_stdin;
   stdout = &LCD_stdout;
printf("Proba");
   return 0;
}
```

Miután ezt kiírta az LCD, a Hyperterminal program segítségével megoldottuk a feladatot.

Hyperterminal beállítások:



A Kód:

```
 int main()
{
   LCD_init();
   serial_init();
   stdin = &serial_stdin;
   stdout = &LCD_stdout;
   char c;
   while(1)
   {
      c = getchar();
      putchar(c);
   }
   return 0;
}
```

Az "Írógép" működött.

3.4. µ*C/OS – egy taszk*

A **4a_ucos_task.c** fájl segítségével írjon egy µC/OS alkalmazást! Az alkalmazás tartalmazzon egy single shot szervezésõ taszkot, amely kiírja az LCD képernyýre az operációs rendszer verzióját!

```
/*********************************************************************************
 * Taszkok                                                       *
 *********************************************************************************/

/* Az OS verziójának kiírása az LCD panelra. */

void Task1(void *data)    {

{
   LCD_init();
   serial_init();
   stdin = &serial_stdin;
   stdout = &LCD_stdout;
         printf("%d",OS_VERSION);
   return 0;
}}
```

Az LCD kijelzőn meg is jelent a 252

3.5. µ*C/OS – időkezelés*

A **4b_time.c** fájl segítségével írjon a már meglévő inicializáló taszkhoz még két további taszkot! Az egyik a másodpercek múlását jelenítse meg az LCD panelen, a másik a már ismert „futófény”-t valósítsa meg! A feladatokat az OS időkezelő szolgáltatásainak igénybevételével (**OSTimeDly()** és/vagy **OSTimeDlyHMSM()**) készítse el!Használhatnánk-e az időkezelő függvények helyett az elsý feladatban alkalmazott várakozó hurkokat (igen/nem, miért)?

- először mindhárom task vermeinek méretét beállítottuk

```
#define TASK1_STK_SIZE  128
#define TASK2_STK_SIZE  128
#define TASK3_STK_SIZE  128
```

- beállítottuk a Prioritásokat

```
#define TASK1_PRIO  10
#define TASK2_PRIO  11
#define TASK3_PRIO  12
```

- inicializáltuk a 2. és a 3. taskot:

```
OSTaskCreate(Task1, NULL, &Task1Stk[TASK1_STK_SIZE - 1], TASK1_PRIO);
```

A másodperc számláló task:

```
void Task2(void *data)    {
int i=0;
   LCD_init();
   serial_init();
   stdin = &serial_stdin;
   stdout = &LCD_stdout;
         while(1)  {
         OSTimeDlyHMSM(0,0,1,0);
         printf("%d\r",i++);
         }    return 0;
```

→ Tökéletesen működött

Futófény task:

```
void Task3(void *data) {
   PORTC = 1;
   DDRC = 0xff;
   int i= 0;
   while(1)   {
      for(i=0; i<50; i++)_delay_ms(20);
      if(PORTC == 0x80)
         PORTC = 1;
      else
         PORTC <<= 1;              }    return 0;}
```

→ Ez is működött