

Programozott elektronika középiskolásoknak okosszoba Arduinoval

IV. rész

A program

Most már jöhet a működést (érzékelést és mérést) vezérlő program megírása. Legyen a neve a címben is szereplő „okosszoba” (a fejlesztői környezet az *okosszoba.ino* [9] állományt menti el számítógépünkön).

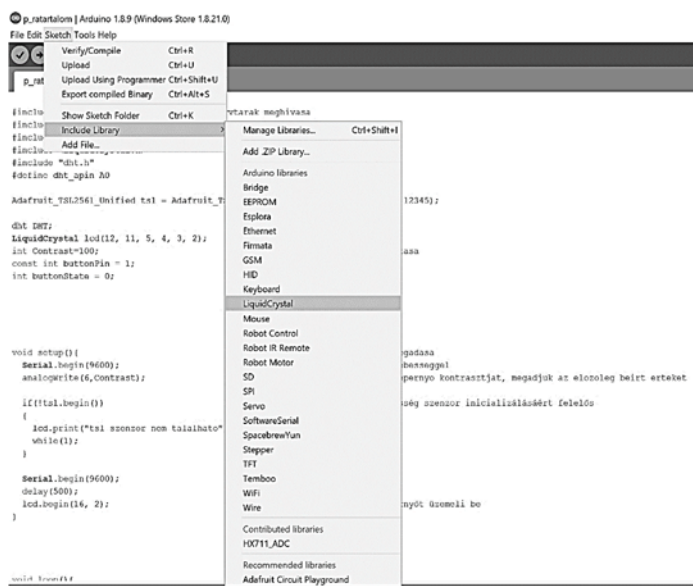
Az Arduinohoz csatlakoztatott eszközök használatához szükség lesz a működésüket elősegítő függvénycsomagokra. Legtöbbször ezek már előre meg vannak írva, ezért érdemes rájuk keresni a világhálón, letölteni a megfelelő állományt, és a fejlesztői környezetben megírt program legelején „meghívni” (betölteni) őket.

Okosszobás alkalmazásunk helyes működéséhez 5 könyvtárra lesz szükségünk: egy az LCD kijelzőhöz („LiquidCrystal.h” [10] - alapértelmezésben tartalmazza már a fejlesztői környezet), kettő a páratartalom és hőmérséklet szenzorhoz (dht.h [11] és Adafruit_Sensor.h [12]), illetve még 2 a fényerősség méréséhez (Wire.h [13] és Adafruit_TSL2561_U.h [14]). Ha sikerült mindezeket beszerezni, az elindított fejlesztői környezetben megnyitunk egy új programot, és hozzáadjuk őket egyenként, néhány egyszerű lépést követve.

Az alapsomagban már meglévő LiquidCrystal.h könyvtár hozzáadásához a „Sketch” fülből kiválasztjuk a „Include Library” opciót, majd a lenyíló listában megkeressük a kívánt könyvtárat.

A többi könyvtár hozzáadásához az „Include Library” lista „Add .ZIP Library” opciójánál megkeressük a letöltött fájlokat, és telepítjük őket.

Az Arduino programozási környezetben megírt programok struktúrája viszonylag egyszerű, több jól beazonosítható rész figyelhető meg, mindegyik jól meghatározott feladatokat lát el.



Első lépésben a könyvtárak hozzáadását, a kezdőfeltételek, az adatcsatornák és a változók beállítását valósítjuk meg.

A könyvtárhozzáadás-rész könnyen beazonosítható az „`#include`” utasítás alapján:

```
#include <LiquidCrystal.h>           // a kijelző könyvtára
#include <Wire.h>                     // az I2C kommunikációs protokoll könyvtára
#include <Adafruit_Sensor.h>         // az Adafruit gyártó szenzorainak meghajtója
#include <Adafruit_TSL2561_U.h>      // a fényérzékelő könyvtára
#include "dht.h"                      // a hőmérséklet és nedvesség érzékelő könyvtára
```

A következő lépésben a kijelző adatcsatornáinak az Arduino digitális bemeneteihez való csatlakozási módjának és a kijelző kezdőkontrasztjának beállítása következik:

```
LiquidCrystal lcd(12,11,5,4,3,2);    // az LCD adatcsatornáinak beállítása
int Contrast=100;                    // az LCD kijelző kontrasztjának beállítása
```

Ezt követi annak a két változónak a definiálás, amelyeken keresztül elérhetjük az érzékelőket („DHT” a hőmérséklet és páratartalom szenzornak, „tsl” a fényérzékelőnek):

```
dht DHT;                              // deklaráljuk a "DHT" változót
#define dht_apin A0;                    // megszabjuk, hogy az A0 bementhez csatlakozik
```

```
Adafruit_TSL2561_Unified tsl = Adafruit_TSL2561_Unified(TSL2561_ADDR_FLOAT, 12345);
//összekapcsoljuk a szenzort a könyvtárral, és deklaráljuk a "tsl" változót
```

Ha mindez megvan, következik a „`setup()`” alprogram, amely az eszközök előkészítését és az alapbeállítások kezdőértékeit fogja megadni. Ez a rész csak egyszer fut le a program működése során.

```
Serial.begin(9600);                    // megnyitja a soros portot 9600 bpm sebességgel
analogWrite(6, Contrast);              // megszabjuk, hogy a 6-os pin felelős a kontraszt beállításáért
```

Az „`if`” döntéshozó utasítás a megvilágítás szenzor beindításáért felelős, és a „`!tsl.begin()`” függvény visszajelzése alapján dönt:

```
if(!tsl.begin())                       // inicializáljuk a fényérzékelő szenzort
{
  lcd.print("tsl szenzor nem található"); // ha nincs beüzemeltetve, kiírjuk azt
  while(1);
}
```

Az ezután következő rész a kijelzőre vonatkozik:

```
lcd.begin(16, 2);                       // inicializáljuk a 2 x 16 karakteres LCS-t
```

A program utolsó szakaszában található a „`loop()`” alprogram, ami tulajdonképpen az Arduino-t működteti, és végrehajtja a kívánt műveleteket.

Először a hőmérséklet és páratartalom mérésére, illetve kijelzésére kerül sor.

```
DHT.read11(dht_apin);                  // megadjuk a hőmérséklet és páratartalom érzékelőt
lcd.setCursor(0, 0);                    // a kiírás helye a 0. sor (felső), 0. karakterénél kezdődik
```

```

lcd.print("e=");           // kiírjuk az "e"-vel jelölt páratartalom karakterét
lcd.print(DHT.humidity);  // kiírjuk az érzékelő által mért páratartalom értékét
lcd.print("%");           // kiírjuk a páratartalom mértékegységét
lcd.setCursor(8, 0);      // a kiírás a 8. karakternél folytatódik
lcd.print("T=");          // kiírjuk a "T"-vel jelölt hőmérséklet karakterét
lcd.print(DHT.temperature); // kiírjuk az érzékelő által mért hőmérséklet értékét
lcd.print("C ");          // kiírjuk a hőmérséklet mértékegységét

```

Amint jól látható, a kiírás helye (kurzor) az „lcd.setCursor” utasítással állítható be. A képernyőre való kiírás az „lcd.print” utasítással történik. A „DHT.humidity” vagy „DHT.temperature” kulcsszavak a szenzorok által mért értékeket jelölik.

Hasonlóan járunk el a fényérzékelő esetében is. Most a kulcsszó „event.light” lesz.

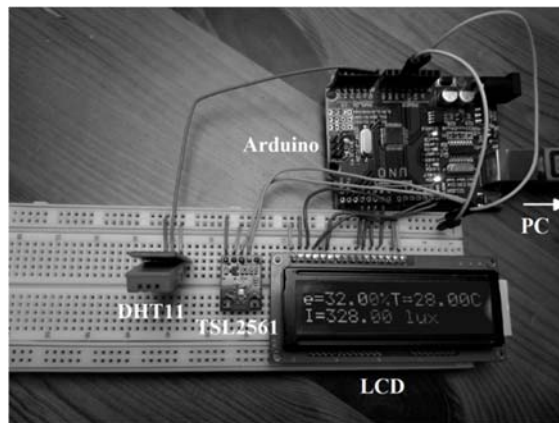
```

sensors_event_t event;    // megadunk egy új eseményt a fényerősség érzékelőnek
tsl.getEvent(&event);     // megadjuk a fényerősség szenzort
lcd.setCursor(0, 1);      // elvisszük a kurzort az alsó sor (1) 0. karakteréhez
lcd.print("I=");          // kiírjuk az "I" -vel jelölt megvilágítás karakterét
lcd.print(event.light);   // kiírjuk az érzékelő által mért megvilágítás értéket
lcd.print(" lux");        // kiírjuk a megvilágítás mértékegységét

```

Az utolsó sorban található „delay” utasítás annyit tesz, hogy 250 ms-onként futtatja az alprogramot, méri újra és újra a hőmérsékletet, páratartalmat és megvilágítást, addig amíg az eszköz be van üzemelve.

A fenti lépések követésével tehát létrehozhatunk egy egészen egyszerű berendezést, amivel megfigyelhetjük környezetünket, betekintést nyerhetünk a programozás és az elektronika világába egy minimálisra leegyszerűsített „okosszoba” létrehozásával, ami bár nem kapcsolja be a fűtést ha hideg van, mégis kedvet hozhat a továbbtanuláshoz, hogy a későbbiekben az is sikerüljön!



Kovács Adél*, Simon Alpár, Tunyagi Arthúr

Magyar Fizika Intézet, Babeş-Bolyai Tudományegyetem, Kolozsvár

* Fizika informatika szakos hallgató, III. év