



Budapesti Műszaki- és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
MIT

Nagyteljesítményű mikrovezérlők tantárgy
[vimim342]

8x8x8 LED Cube

Készítette:
Szikra István
URLJRN

Tartalomjegyzék

1	Összefoglalás.....	3
1.1	Hardver.....	3
1.2	Szoftver.....	3
1.2.1	Mikrokontroller.....	3
1.2.2	PC.....	3
2	Blokkvázlat.....	4
3	Szoftver működés.....	4
3.1	Library.....	4
3.2	Main.....	5
3.3	Animáció.....	5
3.4	Felhasználói eseménykezelő.....	6
3.5	Infra kezelő.....	6
3.6	Parancssor megjelenítő.....	6
3.7	Soros port terminál.....	6
3.8	Parancsértelmező.....	6
4	Prototyping, Standalone Compiler.....	7

1 Összefoglalás

Animációk generálása 8*8*8-as LED kockára mbed modullal (NXP LPC1768 ARM Cortex-M3 mikrokontrollerrel)

1.1 Hardver

- 512 db LED, 8*8*8-as kockába rendezve
- Időosztásos 64*8-as led mátrix meghajtás, 100Hz frissítés *8 szint (szint választó FET váltás)*16 fényesség árnyalat (szint bitminta frissítés) * 64 bites szint meghajtó shift regiszter (820 kbit/s soros sebesség – SPI busz).
- TSOP1736 infra vevő
- Tápellátás: stand-alone, és/vagy csak az mbed USB buszon (RTC)

1.2 Szoftver

A kocka önállóan is használható, viszont egyes funkciók csak PC-ről érhetők el.

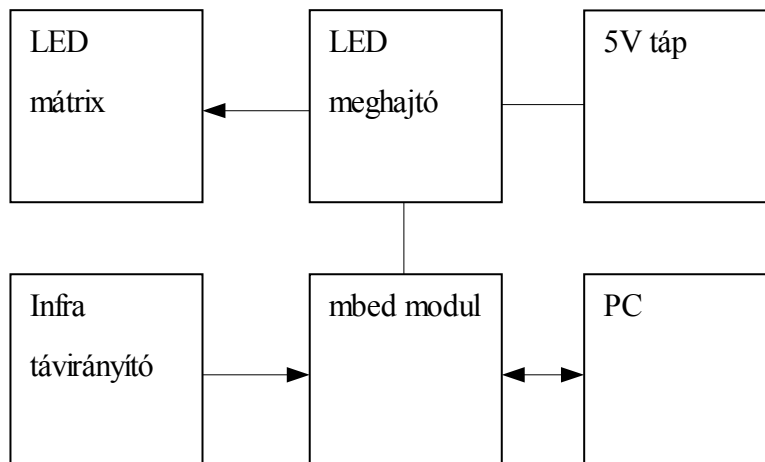
1.2.1 Mikrokontroller

- RTC, idő kijelzés, beállítás
- Soros terminál USB-n keresztül, parancsértelmezővel, időt megjelenítő parancssorral (command prompt)
- Kijelzés: 256 byte-os frame buffer, 4bit/pixel árnyalat
- Rajzolás funkciók
- Animáció választás
- File kezelés: 3D betöltés, lementés, file megtekintés, könyvtár listázás
- Infra távirányítóval futtatható scriptek

1.2.2 PC

Egyszerű soros port terminál. Nincs szükség speciális szoftverre.

2 Blokkvázlat



3 Szoftver működés

3.1 Library

Az mbed online fejlesztőkörnyezete kiterjedt objektum orientált könyvtárral rendelkezik. A programban felhasznált főbb komponenseket mutatom be itt röviden.

Az mbed interfész IC-nek van egy soros portja ami az LPC1768-as ARM mikrokontrollerre van kötve, és amit a PC-n virtuális soros portként el lehet érni. Ehhez Windowson fel kell telepíteni a virtuális soros port drivert¹. A mikrokontrolleren az mbed *Serial* osztály *baud* metódusával lehet beállítani a baud ratet (115200). A *printf*, *putc* metódusokkal lehet formázott szöveget, karaktert küldeni a PC-nek. A *readable()*, *getc()* szolgál a bejövő adatok olvasására.

Időzítőből több osztály is van az mbed könyvtárban: A *Ticker* ütemezésre való, az *attach* metódussal lehet megadni melyik függvény milyen időközönként hívódjon meg. A *Timer* idő mérését szolgálja, melyet a *start()*-tal indítunk és *read_us()* metódussal olvasunk ki. A Real Time Clock kezelésére használt függvények *time*, *set_time*, *mktime*, *localtime*, *strftime*. Az mbed lib 19. revízióban még inicializálni kellett egy *set_time* hívással, amit javítottak 20.-ban, így most már reset után használható a *time* függvény az idő lekérdezésére.

¹ <http://mbed.org/handbook/WindowsSerialConfiguration>

A *LocalFileSystem* osztály az mbed interfész chipen tárolt file rendszerhez fér hozzá. Ez az a tárterület, amit USB-n keresztül elérünk, és amire a bináris állományt letöltjük. A C stdio standard librari-jában megtalálható *fopen*, *fread*, *fwrite*, *fgets*, *fclose*, *opendir*, *readdir*, *closedir* függvények használhatók file kezelésre.

A mikrokontroller programból történő resetelésére egy nem dokumentált függvény használható, aminek hatására a tárolón található az legfrissebb bináris állomány betöltődik, ugyanúgy mintha az mbed modulon megnyomtuk volna a reset gombot.

```
extern "C" void mbed_reset();
```

Az infra távirányító kezeléséhez digitális jelen kell impulzusok idejét mérni, amire az külső megszakítás használó *InterruptIn* osztály alkalmas. Ennek két metódusát (*rise* és *fall*) kell a megfelelő paraméterekkel meghívni, hogy függvényeket rendeljünk a fel és lefutó éleknél bekövetkező eseményekhez. Az idő mérésére a már említett *Timer* osztály való.

A véletlen szám generátort az *srand(time(NULL))* hívással inicializálom, és a *rand()* függvénnyel kérdezem le.

3.2 Main

Induláskor a program inicializálja az általános célú ki-, bemeneteket (státusz és hibajelző ledek, infra vevő), kommunikációs csatornákat, időzítőket (timer, RTC) és lefuttatja az "autoexec.bat" nevű scriptet. A főprogram belső ciklusban frissíti a LED mátrixot, villogtat egy státusz ledet az mbed modulon, és lépteti az animációt ha az engedélyezve van.

3.3 Animáció

A különböző animációk egy függvény pointer tömbben vannak tárolva. A főprogram ezt a tömböt indexelve hívja meg az animációkat (a tömb határait ellenőrizve). Az animációkat lehet futtatni, ebben az esetben a main loop minden ciklusában meghívja a beállított indexű animációt. Lehet léptetni, ekkor csak a beállított számú ciklusban hívódik meg. Jelenleg 17 animáció választható, ebből az egyik véletlenszerű változat animációk között egy részhalmazából választva. A lényegesebbek: pixel shiftelés a frame bufferben körbecsordulással; a kocka palástjának X/Y/Z tengely körüli eltolása egy pixel sorral; véletlen pixel generálás és síkok menti elmozdítása; X,Y,Z irányú feltöltés, ürítés; geometriai minta generálás; óra, perc

kijelzés és palás menti forgatás (saját 4x7-es fontkészlettel); gömb mozgatása szögfüggvényekkel számított koordináták mentén; sík mozgatása és forgatása 3 tengely körül (mátrix szorzással és szögfüggvényekkel).

3.4 Felhasználói eseménykezelő

Az eseménykezelő futtatását egy *Ticker* ütemezi. Az kezelő hívja meg a parancssor frissítését a soros porti terminálra, a soros port bevitel kezelőjét. Szintén itt történik az infra távirányító kezelése. Az érvényes bináris kódot hexadecimális karaktersorozattá alakítja, és mint '.rc' kiterjesztésű filenevű scriptet futtatja.

3.5 Infra kezelő

Az infra megszakítás kezelő dekódolja a távirányító által küldött mintát, és jelzi az eseménykezelőnek, ha érvényes kódot talált.

3.6 Parancssor megjelenítő

Kijelzi az aktuális időt, egy progress indicatort és a begépel, vagy visszahívott parancsot.

3.7 Soros port terminál

A soros portról addig olvas karaktereket, amíg van a bufferben adat. Ezeket egyesével dolgozza fel. Echo nem küld mindenről vissza a PC-nek, erre a parancssor frissítésénél kerül sor. Új-sor karakter fogadásánál meghívja a parancsértelmezőt az aktuális parancssorral. Backspace hatására törli az utolsó karaktert a parancssorból, Tab hatására üres parancssor esetén visszatölti a parancssorba az utoljára végrehajtott parancsot. Minden egyéb karakter bekerül a parancssor végére, amíg az el nem éri a maximális 64 karakter hosszúságot.

3.8 Parancsértelmező

Ez a parancs végrehajtó is egyben. A parancsok listája:

- `dir` kilistázza a fílerendszeren található fileokat.
- `more` kiírja a paraméterben kapott file tartalmát és file méretét
- `time` kiírja az aktuális dátumot és időt, ill. ha van paramétere, akkor beállítja az időt

- `date` kiírja az aktuális dátumot és időt, ill. ha van paramétere, akkor beállítja a dátumot
- `help` megegyezik a "more help.txt" paranccsal
- `info` kiír pár rendszer információt: rendszer órajel, Fordító verziója, fordítás ideje, fordított file neve
- `script` paraméterben megadott file-t beolvassa és soronként végrehajtja a parancsértelmező újbóli meghívásaival
- `load` paraméterben megadott file-t tölti be a 256 byte-os frame bufferbe
- `save` paraméterben megadott file-ra menti el a 256 byte-os frame buffert, ezzel lementhetők szerkesztett ábrák
- `set` pixel beállítás, paraméterek az XYZ koordináta és a fényesség
- `line` 3D vonal rajzolás, paraméterek a kezdő és vég koordináta és a fényesség
- `plane` sík rajzoláshoz használható többszörös vonalrajzoló, paraméterek a vonal kezdő, vég koordinátása, fényessége, vonal eltolási vektora, eltolások száma
- `run` animáció futtatása, paraméterben az animáció száma adható meg, ennek hiányában az aktuálisat indítja el
- `step` a paraméterben megadott számút fog lépni az animáció
- `stop` animáció leállítása, a run-hoz hasonlóan paraméterezhető
- `reset` a mikrokontroller újraindítása

4 Prototyping, Standalone Compiler

Az online fordító az első lépésekben kényelmes, de több ezer soros programok írása egy böngészőablakban már nem az. Az egyik jó leírás mbed-re történő fordításról amit olvastam a *Compiling your own CMSIS Code for the mbed²* ami a *Code Sourcery's ARM EABI toolchain³*-t használja. Az online fordító által generált bináris állományt nem csak mbed modulokon lehet futtatni hanem saját fejlesztésű LPC1768-as mikrokontrollert használó paneleken is: *Prototype to hardware⁴*.

² http://dev.frozenskimo.com/notes/compiling_your_own_cmsis_code_for_the_mbed

³ <http://www.codesourcery.com/sgpp/lite/arm/download.html>

⁴ <http://mbed.org/users/chris/notebook/prototype-to-hardware/>