



Cseh Róbert

**60 nap alatt
Arduino**

A tanfolyam

Idézetek hallgatóktól akik a tanfolyamon elején a felmérésben kérdéseket tettek fel

„Végre megértettem a tranzisztor működését! A Wikipedia túl egyszerű, a Sulinet fizika szekciója túl bonyolultan írta le....”

D. Péter - Kecskemét

„Hírlevelet kapok, amik alapján a honlapot böngésem, és sóvárgok a tudás után.”

O. Vince - Taksony

A kezdet: „Teljesen kezdő vagyok a témában. Tudtok segíteni?”

K. Tibor - Toronto

„...izgalmas lehetőségek, hiszen az elektronika az analóg világból átfordult digitálisba. Mindent mikrocsipekkel csinálnak (gsm-ek, órák, mosógépek és riasztók, járműelektronika stb.)”

D. Péter - Kiskunhalas

„Eldöntendő kérdés, hogy PIC, vagy AVR. A megépítése egy áramkörnek talán már nem gond, a programozásra még nem állt rá az agyam. Talán majd most.”

Mi jut eszedbe, ha azt hallod: TavIR?

„Egyszemélyes projekt, ami eddig kitartóbb a magyar internet történetében, mint az eddigi hasonló projektek a látókörömben!”

Ha AVR programozásról tudni szeretnék valamit ez az első oldal.

AVR programozás, fejlesztések, új gondolkodás – nem csak elmélet.

Tartalomjegyzék

1.	Történelem	9
1.1.	<i>Vágjunk is bele</i>	9
1.2.	<i>A tanfolyam elején a segítségedet kérném még</i>	10
2.	Hát én immár kit válasszak?	10
2.1.	<i>Arduino családtagok</i>	10
2.2.	<i>Mit válasszak?</i>	13
3.	Ökölszabály	13
3.1.	<i>Mi is az a LED?</i>	13
3.2.	<i>Tranzisztorok</i>	15
4.	A kérdőív hibája	16
5.	A mikrokontroller	17
5.1.	<i>"Hát én immár kit válasszak?"</i>	17
5.2.	<i>Az a bizonyos első</i>	18
5.3.	<i>Ez összerakható vagy szétszedhető?</i>	19
5.4.	<i>És ez működik?</i>	19
6.	Arduino	20
6.1.	<i>Hardware-telepítés</i>	20
6.2.	<i>A puding és a próba</i>	20
6.3.	<i>És ez hogyan működik?</i>	21
7.	A TavIR-AVR fórumról	21
8.	Alaputasítások	22
9.	Futófény	25
9.1.	<i>LED villogó</i>	25
9.2.	<i>Futófény</i>	26
9.3.	<i>Upgrade / feljavítási tippek</i>	28
10.	Nyomógomb	28
10.1.	<i>Külvilági jelek</i>	28
10.2.	<i>A gyakorlat</i>	29
11.	Nyomógomb és a prell	30
11.1.	<i>Való világ</i>	30
11.2.	<i>A gyakorlat</i>	30
11.3.	<i>Nyomógombok és a valóság</i>	31
12.	Változók	33
12.1.	<i>Változók általában</i>	33
12.2.	<i>A változatosság gyönyörködtet</i>	33
12.3.	<i>Túlcsordulás</i>	35
13.	Megjegyzések, változók	35
13.1.	<i>Aritmetika</i>	36
13.2.	<i>Megjegyzések</i>	37
14.	Analóg kimenet (PWM)	38
14.1.	<i>A háttér</i>	38
14.2.	<i>Szoftveres megoldás</i>	38
14.3.	<i>Hardver-megoldás</i>	39
15.	Soros kommunikáció	40
15.1.	<i>A történelem homálya</i>	40
15.2.	<i>Sorosporti programozás</i>	41
16.	Nemfelejtő memória	43
16.1.	<i>EEPROM</i>	44
17.	Időzítés	46
18.	LCD kezelés I.	49

18.1.	<i>Arduino és az LCD</i>	49
18.2.	<i>Kicsit a bekötés háttéréről</i>	51
18.3.	<i>Használat</i>	51
18.4.	<i>Hogyan működik?</i>	52
19.	LCD kezelése II.	55
19.1.	<i>"Árvízűtűró tükrőfűrógép"</i>	55
19.2.	<i>Saját karakterek</i>	56
19.3.	<i>Használat</i>	56
19.4.	<i>Hogyan definiálunk mégis?</i>	60
19.5.	<i>A kiírás az LCD-re</i>	61
19.6.	<i>Sebességbajnok kijelző</i>	61
20.	LCD kezelés III.	62
20.1.	<i>Kijelző belső felépítés</i>	63
20.2.	<i>A kijelző-sebesség</i>	63
20.3.	<i>Sebességmámor</i>	64
20.4.	<i>Építsünk órát!</i>	66
21.	Ciklikus hangok	66
21.1.	<i>Hangkeltés</i>	67
21.2.	<i>Hangkezelés</i>	68
21.3.	<i>Ciklikus feladatok</i>	69
22.	Eljárások és a hangok	72
22.1.	<i>Szervezzük univerzálissá!</i>	73
22.2.	<i>beep()</i> eljárás.....	77
22.3.	<i>Hang sorrend</i>	80
22.4.	<i>Ébresztőóra</i>	81
23.	Óratippek	81
23.1.	<i>Eltelt idő kiszámítása</i>	81
23.2.	<i>Telik az idő</i>	82
23.3.	<i>Ébresztés</i>	83
24.	Analóg világ	84
24.1.	<i>De mi is az az analóg jel?</i>	84
24.2.	<i>Gyakorlatban</i>	84
24.3.	<i>A második kísérlet</i>	85
24.4.	<i>Jelszintmérés</i>	86
25.	Analóg varázslat	87
25.1.	<i>Az analog jelek haszna</i>	87
26.	Megszakítás	90
26.1.	<i>Bréking! Rendkívüli esemény!</i>	90
26.2.	<i>Interrupt</i>	91
26.3.	<i>Programsorok, utasítások, háttér</i>	93
26.4.	<i>Bug vagy Feature? Hiba vagy „jó ez nekem”?</i>	94
27.	Megszakítások korlátai	96
27.1.	<i>A korlátok</i>	96
27.2.	<i>Összefoglalásul</i>	97
28.	I²C/TWI busz alapjai	98
28.1.	<i>De hogyan is működik?</i>	98
28.2.	<i>A protokoll felépítése</i>	99
28.3.	<i>I²CScanner</i>	100
29.	IIC hőmérő	101
29.1.	<i>Hőmérő</i>	101
29.2.	<i>Hőmérő konfigurálás</i>	103
30.	I²C EEPROM	105
30.1.	<i>Hőmérő? Már megin'?</i>	105

30.2.	<i>EEPROM</i>	106
31.	I²C és az EEPROM II.	109
31.1.	<i>Függvény? Eljárás?</i>	109
31.2.	<i>Logikai műveletek</i>	112
31.3.	<i>Vissza az EEPROM-hoz</i>	113
32.	LCD ébresztőóra	114
32.1.	<i>Gombok</i>	114
32.2.	<i>Az Arduino kivezetései lassan elfogynak</i>	114
32.3.	<i>Ébresztés</i>	115
33.	I²C és az óra	119
33.1.	<i>Az óra probléma</i>	119
33.2.	<i>Az IIC óra: PCF8563T</i>	119
33.3.	<i>Chip működése</i>	122
34.	Portbővítés	127
34.1.	<i>Oh, azok a lábak</i>	128
34.2.	<i>Gyakorlati trükkök</i>	130
34.3.	<i>Búcsú az I²C/TWI busztól</i>	131
35.	Bootloader	131
35.1.	<i>Szűz chip</i>	131
35.2.	<i>Gyakorlat</i>	131
35.3.	<i>Bootloader</i>	132
35.4.	<i>Hogyan működik?</i>	134
36.	Shift regiszter	135
36.1.	<i>Számoljunk</i>	135
36.2.	<i>A múlt öröksége</i>	135
36.3.	<i>De hogyan működik?</i>	135
36.4.	<i>A shift regiszter</i>	136
37.	Shift regiszter II.	139
38.	A shift regiszter és hétszegmenses kijelző	142
38.1.	<i>A kijelző</i>	142
38.2.	<i>De miért jó a hétszegmenses kijelző?</i>	145
38.3.	<i>MAX72xx</i>	145
39.	A shift regiszter és az LCD	146
39.1.	<i>Alapmegoldás</i>	146
39.2.	<i>Egyszerű kezelés</i>	148
40.	Az optimalizálás	148
40.1.	<i>Arduino</i>	148
41.	Kisebb nem lehetne?	150
41.1.	<i>Villám-trükk</i>	150
41.2.	<i>Chipfüggés - de nem számít</i>	150
41.3.	<i>Végső menedék - avagy minden mindegy</i>	151
41.4.	<i>Tanácsok</i>	152
Köszönöm		153

1. Történelem

Köszöntelek a *TavIR: 60 nap alatt Arduino* tanfolyam első leckéjében.

Ha nem probléma, akkor a leírások, leckék és tippek tegeződve, esetleg harmadik személyben íródnak... Mostantól minden átlag első-második-harmadik napon egy rövid leckét, tippet vagy gondolatébresztő feladatsort fogsz kapni. A feladatsorok megoldásai nem feltételei a következő leckéknek! A legtöbb esetben valamely későbbi lecke a megoldás részletes bemutatását is tartalmazni fogja.

1.1. Vágjunk is bele...



Valamikor egy messzi-messzi galaxisban volt egy elektronikai gyártó-kereskedő cég, aki a Nippon Calculating Machine Corporation Ltd. nevet viselte 1945 és 1967 közt - majd 1967-ben vette fel a Busicom nevet. A múlt század 1969. évében az Intelhez fordultak - akik a memóriachipek területén már nevet szereztek ekkor -, hogy részükre programozott logikájú chipeket gyártsanak. A Busicom számológépekhez, így a szükséges 21 chipes kollekciót egyetlen tokba sikerült elhelyezni - hiszen ekkor még a gépek csak kapuáramkörök segítségével végezték a műveleteket. A Busicom több céget is megkeresett a feladattal, de csak az Intel vállalta el; majd itt a felkérésnek megfelelően négyféle áramkör készült el - különböző tudással. A gyártás egyszerűsítése végett végül ténylegesen nem négyféle, hanem egyetlen áramköri lap készült - amely a különböző folyamatok sorrendjeit tartalmazta. A szerződés annak módja és rendje szerint lezajlott és az elkészült chipek valamint és dokumentációk átadásra kerültek. Majd másfél év telt el és az Intel megkereste a Busicom-ot, hogy a chip fejlesztési eredményeinek egyes részeit használnák más célra is, így megvásárolnák. A Busicom-nak ez csak egy "sajtpapír" volt, hiszen ők csak felhasználták az eredményeket - nem kutattak-fejlesztettek.

Így az Intel birtokába került egy olyan technológia, mely:

- egy chiprészletet tartalmaz, ahol folyamatok hajthatóak végre,
- egy másik chip/chiprészlet, ahol ezen folyamatok sorrendisége letárolható,
- ennek a letároló résznek a tartalma megváltoztatható, felhasználó által elkészíthető.

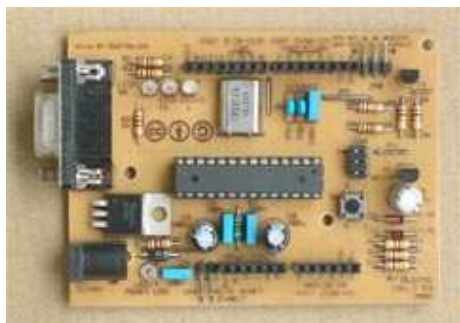
El is jutottunk így processzorig (folyamat-/utasítás-végrehajtó egység) és a program-memóriáig (tárolt, módosítható utasítássorrend).

A tudás (és immár a szabadalom) birtokában az Intel 1971. november 15-én piacra dobta az i4004 processzorát, mely 4 bites processzorként 2300 db tranzisztort tartalmazott és 10 um csíkszélességű volt. Ezt követte a 3300 tranzisztoros 8008 típusjelű 8 bites processzor 1972 áprilisában. Az Intel 4004-es teljesítménye mára elég törpének tűnik, kb. 0.25 MIPS (0.25 millió utasítás/másodperc).

Előrevetítésként: az Arduino lapka számítási kapacitása 16 MIPS, azaz ennek 64-szerese!

A tudomány és történelem lecke ennyi volt mára – pihenésül és bevezetésül. A következő lecke néhány elektronikai ökölszabályt mutat be - melyeket már a mikrokontrollerek alkalmazásával összefüggésben is használhatunk...

2.1.2. Arduino/S



Az Arduino/S-nek soros porti felülete van csak, így régi soros porti csatlakozó (RS-232) kell hozzá, valamint hálózati 9 V (min. 500 mA) adapter. **Fontos!** Csak a MAX232 chippel szereltek működnek az USB-Soros átalakítókkal, a tranzisztoros illesztést tartalmazók nem! A központi chip eredetileg ATmega8, később ATmega168 (újabbban az ATmega328 chippel is működtethető). A hardware csatlakozás megjelenésekor a későbbi kváziszabványos felületet alakította ki.

"Vén kecske is megnyalja a sót."

2.1.3. Arduino Diecimila/Duemilanove



USB kommunikációs felületet már beépítve tartalmazza ez a lapka - ez volt az újítás kulcsa. Az FT232BM illetve később FT232RL illesztőchip minden operációs rendszerhez kapcsolható, hiszen a driver-támogatás sokféle rendszerhez rendelkezésre áll (Win98..Win7x64(Win8), Linux, MacOSX, Android, WinCE,...). Az Arduino szoftverkörnyezet azonban csak MacOSX, Linux, Windows alá érhető el:) ...

A Diecimila és a Duemilanove közt egyetlen különbség, hogy a Duemilanove esetén az USB-külső táp közti átkapcsolás automatikus, míg a Diecimilla esetén kézzel kell ezt megtenni. Általában ATmega168 vagy újabban ATmega328 chippel szerelik, de visszafelé az ATmega8 áramkörrel is kompatibilis. A kialakítás/elrendezés az Arduino/S rendszerrel 1:1 csereszabatos (kivéve soros porti csatlakozót, mely helyén az USB-B került beépítésre).

"Stabil, kiforrott középkorú úr, párját keresi. Az ifjonti hév helyett a tapasztalat dominál."

2.1.4. Arduino Mega / Arduino Mega2560



Az Arduino hardware nagytestvérenek is nevezik. Nagyobb tudású chip került bele, de feláldozták a kezelhetőség oltárán az Arduino- koncepciót. A csatlakozók méretileg és elrendezésileg csereszabatosak ugyan az Arduino/S kváziszabvánnyal, de a funkciók nem kerültek a ténylegesen Arduino-szabványos helyükre. A nagyobb memória miatt került a piacra, a rosszabb szoftveres támogatás, a korábbi kódok átportolása lassan halad - így az elterjedésének sok minden útjába áll. A központi chip SMD, 0.5 mm lábostással szerelt, mely nem cserélhető, ATmega1280 vagy ATmega2560 típusú IC. Az USB illesztés a Duemilanove esetén már debütált megoldás lett.



"Nem mind arany, mi fénylik."

2.1.5. Arduino UNO

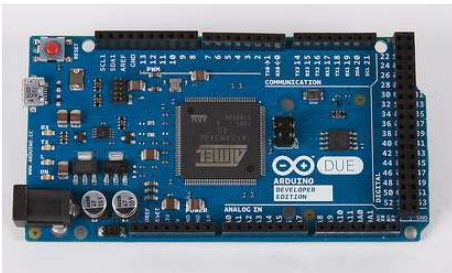
Az UNO lapka 2010-ben jelent meg.

A Duemilanove-hez képest az újdonsága az volt, hogy az USB illesztő chip szabadon átprogramozható, így a soros-USB átjáró helyett MIDI, joystick vagy

billentyűzetként is bejelentkezhet a PC-re. (Ezek a funkciók nem az Arduino szoftver-komponensei!) Az áramkör alapvetően ATmega328 chippel szerelt, visszafele az ATmega168 chippel kompatibilis. Az UNO újdonsága volt, hogy a chip betöltő-programja is megváltozott: kisebb és funkciószegényebb lett (adatellenőrzések, EEPROM programozás hiányzik). Az UNO driverrel időnként vannak problémák, a használt LUFA belső rendszer nem mindig van a helyzet magaslatán - de folyamatosan javul!

"Fiatalos hévvel durrbele..."

2.1.6. "Arduino Due"

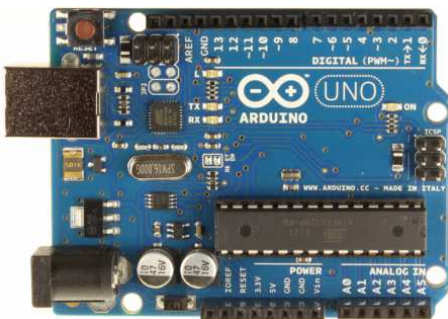


A cím nem véletlen idézőjeles. Olyan az eszköz, mint az izzó vasdarab: beszélnek róla, fényképen látható - de senki nem fogta még a kezében:).

Az Arduino lassan 2 éve publikálta, hogy ARM-ra is kiterjesztik a szoftvert. A SAM3 chip csak 3.3 V feszültségről jár és a bemenetei sem szeretik az 5 V-ot... Így műszakilag elég érdekes lesz a fedpanelek illesztése...

A probléma kezeléséhez felmerült, hogy a tápfeszültség-csatlakozón legyen kivezetve a rendszer-tápfeszültség mellett az I/O buszok feszültségjele is. Sőt, hamar hozzányúlunk az áramkörhöz, akkor az ADC jelek mellett az I²C/TWI busz is jelenjen meg külön kivezetve. Hely csak a panel tetején a digitális portok végén volt. Ez a kváziszabvány lábkiosztás módosítását jelentette. Az átalakítás óta hívják ezt az elrendezést R3-nak.

"Mint a Colombo felesége..."



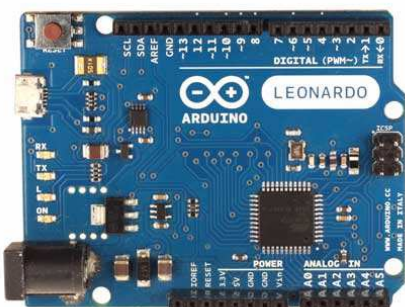
2.1.7. Arduino UNO R3

A lapkán a bővített csatlakozósor mellett az USB illesztőchip az ATmega16U2-ra bővült. A panelek driver-támogatása már közel 100%-os (MacOSX alatt nem tökéletes). De az automata tápfeszültség választás támogatása megmaradt. Az Arduino a fejlesztések során az UNO áramkört preferálja.

"A fejlődés nem áll meg!"

2.1.8. Leonardo

A központi ATmega32U4 chip beépített USB vezérlőre épülő áramköri kialakítás teljesen új fejlesztés. Kicsit új, így a gyermek fejlődésének még vannak bizonytalanságai. Az Arduino 1.0 óta támogatott. A Leonardo lehet billentyűzet, eger is - az Arduino 1.0.1 már beépítve a minták közt tartalmazza a programokat.



"Fejlődőképes :)."

28.3. I²CScanner

Mielőtt nekifognánk az egyes eszközök részletes taglalásának, egy segédprogramot meg kell ismernünk. Ez nem más, mint az I²Cscanner nevű kis alkalmazás. Ennek segítségével felderíthetjük a buszt, hogy mely címeken/azonosítókon vannak eszközök.

```

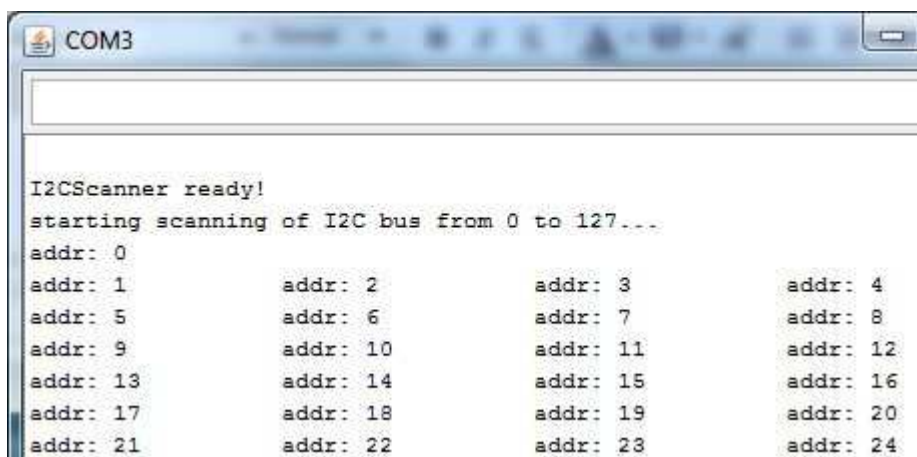
/* I2Cscan EXT1
I2CScanner minta C-Arduino hibrid
60 nap alatt arduino tanfolyam
(c) TavIR http://avr.tavir.hu
Forras: 2009, Tod E. Kurt, http://todbot.com/blog/
*/

#include "Wire.h"
extern "C" {
#include "utility/twi.h"
}

//A busz scannelesenek fuggvenye: adott cimem van e valami?
void scanI2CBus(byte from_addr, byte to_addr,
                void(*callback)(byte address, byte result) )
{
  byte rc;
  byte data = 0;
  for( byte addr = from_addr; addr

```

A programban számos utasítás ismeretlen lehet még. Nem is csoda, hiszen az Arduino és a C nyelv keveredik benne. Ennek megértése és magyarázata jóval meghaladja ezen tanfolyam kereteit - egyelőre fogadjuk el, hogy ez így jó. És használjuk egészséggel. Az eredmény pedig a terminálképernyőre visszaírt lista, hogy hol kapunk eszközt a buszon. (*Haladók részére*: az ún. pointer vagy mutató (*p) a kettővel későbbi leckében megjelenik, ahogyan az I²C eljárás-utasítások is.)



```

COM3
I2CScanner ready!
starting scanning of I2C bus from 0 to 127...
addr: 0
addr: 1          addr: 2          addr: 3          addr: 4
addr: 5          addr: 6          addr: 7          addr: 8
addr: 9          addr: 10         addr: 11         addr: 12
addr: 13         addr: 14         addr: 15         addr: 16
addr: 17         addr: 18         addr: 19         addr: 20
addr: 21         addr: 22         addr: 23         addr: 24

```

A mai napra még egy olvasni való: a Philips IIC ismertetője - magyar nyelven. A következő leckében már bele is ugrunk a kommunikáció megismerésébe és az első alkalmazásunk a hőmérő lesz (LM75, TCN75 chippel)...

Ha valami kérdés merült volna fel ezzel vagy a korábbi bármely leckével, a [Fórum](#) mindig nyitva áll! Kérdezz bátran! Senki sem a Tudás Szent Gráljával a fejében született!

Hasznos link a tananyaghoz:

- [Philips IIC ismertető](#).

-25 esetén az egyes lépések:

1. +25 : 0001.1001
2. negált: 1110.0110
3. szám +1: 1110.0111

És ez a logika adja az eredményt. Azaz hőfok kiszámítása ez alapján megírható. Mondjuk házi feladatként...

Így, ha készen kapnád, ez lenne a sültgalamb - bár így nem repül:



Még nem próbáltam, ha teszteled - kérnék egy beszámolót!

Recept: Rozmaringos sült galamb puliszkával és zsályamártással

30.1.2. Hőfokugrálás

A hőmérő mérése a chipen belül, egy kalibrálatlan diódán a feszültségesésen át kerül mérésre. A szilícium félvezetők esetén ez kb. 2 mV/fok változást jelent. Az eszközben belül ezt analóg feszültség szintként megmérjük. Némi számolás és kész is a hőmérő:).

Ha a chipünk gyanúsán viselkedik, melegszik vagy forrósodik - az nagyfrekvenciás gerjedésre utal. Ekkor a Vcc és a GND kivezetése közé 100 nF kerámiakondenzátort kell raknunk, - és megoldódik a probléma.

30.1.3. Pontosság

Vessünk egy pillantást a chip adatlapra:

Legrosszabb esetben a valós és a mért hőfok közt +/- 1.5 fok van és a reprodukciós mérés hibája is 1 fok. A hőmérő így nem igazán alkalmas precíziós mérésre - hiába tud 0.0625 fok pontosan valamit! Emlékeztet a klasszikus mérési szabályra: *"Fő a pontosság: Mérd mikrométerrel! Jelöld krétával! Vágj baltával!"* Ilyen pontatlanság termosztátvezérlést ne bízunk rá, ám közelítő mérést egyszerűen végezhetünk vele...

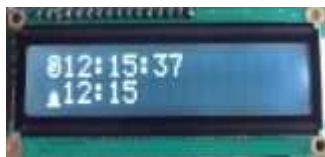
De elkalandoztunk az eredeti témától, gyérünk vissza a nemfelejtő memóriához!

30.2. EEPROM

A nemfelejtő memóriák szinte minden projektben jelentős szerepet töltenek be - kalibrálási adatokat, mérési eredményeket tárolhatunk a chipben. Az AVR belső EEPROM memóriáját nagyon hamar kinőhetjük, így szükséges valami egyéb megoldást találni - ami nagyobb kapacitású, mint a belső; könnyen kezelhető és egyszerű felépítésű. Így a keresgélésben el is jutunk nagyon hamar a külső I²C-s EEPROM-hoz.

A 24LC256 típusjelű memóriára esett a választásom a kísérletek során, mivel ez már jelentős tárkapacitást képvisel - minimális költség mellett. Az I²C flash memóriák tárkapacitása kbit-ben van

32. LCD ébresztőóra



Az LCD kijelzővel megépített óra az utóbbi időben, mintha egy kicsit a háttérbe szorult volna. Pedig sokminden erre a projektre épül(ne)!

Eddig a programunknak mit is kell(ene) tudnia?

- fix indulási időpont,
- fix ébresztési időpont,
- opcionálisan nyomógommbal beállíthatóság,
- ébresztő hanggal,
- opcionálisan kikapcsolt háttérvilágítás,
- másodperces kijelzésű óra-perc-másodperc,
- opcionális dátumkijelzés.

A program kezd egyre nagyobbra nőni :) ! Eddig valami a bevezetőben látható képet kellene látni, ha [2x16 karakteres fekete-fehér kijelzőnk](#) lenne...

32.1. Gombok

A programunk legnagyobb hibája, hogy sem az időt, sem az ébresztést nem tudjuk jelenleg beállítani. Persze, jók erre a nyomógombok, csak sok kivezetést elfoglalnak. A megvalósítási tervet ábrázolhatjuk összefoglalásként:

Ébresztés kapcsoló	+	x	x	x	x
Ébresztés/órabeállító gomb	x	-	-	+	-
Óra gomb	x	+	-	+	-
Perc gomb	x	-	+	-	+
Fel gomb	x	+/-	+/-	+/-	+/-
Le gomb	x	+/-	+/-	+/-	+/-
Funkció	Ébresztés ki/bekapcsolás	Óra beállítása (óra)	Óra beállítása (perc)	Ébresztés beállítása (óra)	Ébresztés beállítása (perc)

Ahol a következő lehetőségek vannak: gomb bekapcsolva/benyomva (+), gomb elengedve/kikapcsolva (-), gomb nincs hatással a funkcióra (x).

5 gomb és egy kapcsoló! És ekkor nem is állítunk másodpercet, dátumot! Ezek újabb gombok lennének...

32.2. Az Arduino kivezetései lassan elfogynak...

Megoldás lehet a soros kommunikáció segítségével az idő beállítása. A pontos időt a T120452 sorral (az idő:12:04.52), míg a dátumot a D120324 (2012. március 24.), végül az ébresztést az A120600 (az ébresztés:12:06) utasítással állíthatnánk be... Az Arduino alatt ez nem olyan egyszerű, mert a szövegekkel, karakterláncokkal nagyon mostohán bánt...

40.1.1. Hardware-bővítés

Az Arduino család 28 lábú AVR központi chipje többféle lehet: ATmega8, ATmega168 vagy ATmega328. Ezek memóriamérete ebben a sorrendben nő. Jelenleg az ATmega8 chippel szerelt paneleket nagyítóval kell keresni, az ATmega168 és az ATmega328-as elterjedt. Az egyes chippek memóriaméretei:

Chip	Flash	SRAM	EEPROM
ATmega8	8 k	1 k	512 b
ATmega168	16 k	1 k	512 b
ATmega328	32 k	2 k	1 k

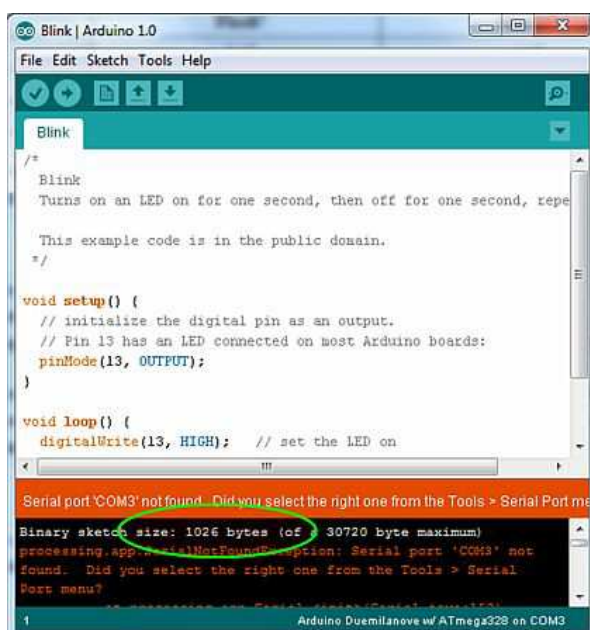
Az árkülönbség minimális közöttük, így ha most kell választani, akkor az ATmega328-as áramköröket válaszd. Az Atmel oldalán és a fejlesztői fórumban nem találtam utalást az esetlegesen fejlesztés alatti ATmega648-as chipre (64k Flash, 4k SRAM, 2k EEPROM). Pedig, de jó lenne használni :)!

40.1.2. Bootloader

A bootloader fejezetből már megismerhetted e kis program funkcionalitását. Ami számunkra most fontos: a Flash memória végéből csíp le, így a hasznos program-méret helyet csökkenti. Ez a standard bootloader esetén 2 kbyte, míg az UNO áramkörök esetén debütált Optiboot 0.5 kbyte helyfoglalású. Ha cipőskanállal fejlesztünk, akkor utolsó mentsvárként a bootloadert az Optiboot-ra cseréljük ki (persze, átírták már az összes chipre - így a feltelepített Arduino szoftver hardware/bootloader alatt megtalálhatjuk az új bootloadert).

Ha már ez sem elég és az utolsó bit helyre is szükségünk van a chipben, akkor a bootloadertől is megszabadulunk. Arduino 1.0 esetén már közvetlenül a programozóval is beégethetjük a lefordított programot a chipbe (*File - Upload using programmer*) - így akár le is zárhatjuk a chipet, hogy illetéktelen kódkalóz ne klónozza az eszközt. Mert ki szeretné a programját, többhavi munkáját a szomszéd weboldalon, kalózkidásként viszont látni? (*Példabeszéd*: ugyanez visszafele is igaz! Mind a két oldalon állhatunk!)

40.1.3. Fordító csere



Ha a mintaprogramunk nem akar beleférni a chipbe, akkor a programkódot összenyomhatjuk. Erre - az arduino kötöttségei miatt - nem sok lehetőségünk van. Próbáljuk ki az Arduino 1.0 helyett az Arduino 0023 verzióval (vagy akár régebbivel) való fordítást. Hogy mit nyerünk, azt a feltöltéskor láthatjuk...

Ezek az egyszerűbb, kézenfekvőbb megoldások voltak. A következő leckében már a programkódba nyúlunk bele....