

dátum:

a mérést végezte:

A PIC16F876 I/O portok használata.

– m é r é s i j e g y z ő k ö n y v –

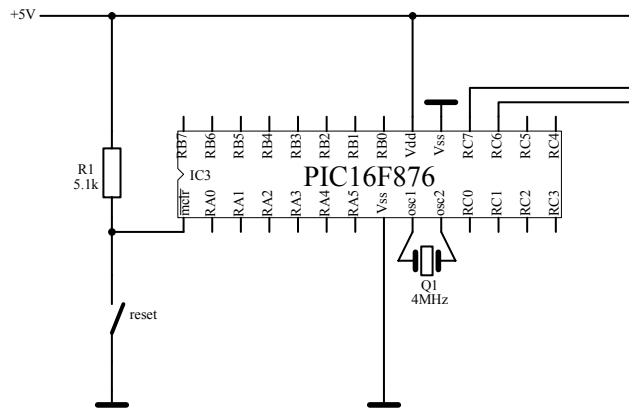
Elsőként elevenítsük fel az előző laboron megszerzett ismereteinket a rendelkezésre álló fejlesztő környezettel kapcsolatosan:

Mérési összeállítás

A laboratóriumi mérési összeállítást sikerült úgy leegyszerűsíteni, hogy még programozó egység használatára se legyen szükség! A *flash* programmemóriával rendelkező PIC-ekben futó program képes módosítani a programmemória tartalmát. Ezért ezekbe a PIC-ekbe elegendő, ha egyszer írunk programozó egységgel egy ún. "*boot*" programot.

A *boot* program kicsi és a programmemória felső részén foglal helyet, szabadon hagyva a memória "alsó" felének nagy részét a felhasználói alkalmazás számára. Feladata, hogy bekapcsolást ill. resetet követő néhány másodpercben figyelje a PIC soros perifériáját (*Usart*), hogy óhajt-e vele valaki adatot cserélni. Ha igen, és sikeresen azonosította a PC-n futó *bootloader* programot, akkor megkezdődik a PC felől jövő adatfolyam programmemóriába írása (a PIC programozása), ellenkező esetben lezárja, vagy meg sem kezdi a nem létező soros kommunikációt. Ilyenkor (az esetek többségében, amikor nem programozzuk, "csak" egyszerűen használjuk a PIC-et) néhány másodperces türelmi idő letelte után, a program végrehajtás a PIC-ben található felhasználói programmal folytatódik.

A laboratóriumban használt "minimális mérési összeállítás" az alábbi ábrán látható. A kapcsolási rajz jobb oldalán látható rész – tranzistoros változatban -egy külön nyomtatott áramkörön van felépítve, melyet egy csatlakozóval tudunk a PIC-hez kapcsolni. **A PIC 20-as lábát a csatlakozónk megjelölt lábához kapcsoljuk, a további három láb a 19,18,17-es lábhoz kerüljön.**



"minimális mérési összeállítás"

A nyomtatott áramkörön felépített rész soros kábelen keresztül kapcsolódik a számítógép RS232-es portjához. Ezen a kábelén keresztül töltjük le a számítógépen előállított (megírt és lefordított) programot a mikrovezérlőbe, és ugyanezen kábelén keresztül kommunikálhat a mikrovezérlőben futó felhasználói program pl. egy PC-s programmal.

A programírás, fordítás lépései:

Indítsuk el a PC-n a *mikroPascal* programot. Ha egy új programot szeretnénk elkezdni, akkor a *mikroPascal* által megnyitott legutóbb használtat csukjuk be (*File/Close All*). Kezdjünk egy új projectet (*Project/New Project...*) Adjuk meg az új programunk nevét (*Project Name:*), elérési útvonalát (*Project Path:*), a mikrovezérlő típusát (*Device:PI16F876*), és a kristályrezonátor frekvenciáját (*Clock: 4MHz*), majd nyomjuk meg az *Ok* gombot.

Megj.: Új programot mindig új mappában kezdjünk!

A fenti módon létrehozott új project két file-t hoz létre a választott mappában. Egy *.ppp* kiterjesztésű project file-t és egy *.ppas* kiterjesztésű pascal forrásfile-t. A pascal forrásfile-t látjuk megnyitva a szerkesztőben. Írjuk meg a programunkat. Szerkesztés közben sokat segít az integrált fejlesztői környezet *kódkiegészítő* funkciója, mely Ctrl-Szököz-el hívható elő az egyes utasítások valamint függvény/eljárások gépelése közben (beleértve a beépített könyvtárak és a magunk által deklarált függvény/eljárásainkat is). A beírt függvény/eljárás nevet követő nyitó zárójel máris előhívja a használandó paraméterlistáról tájékoztató sűgöt. (Egyébként ez utóbbi funkció Ctrl-Shft-Szököz-el csalogatható elő).

F1-el pedig bármikor előhívható a *mikroPascal* sűgője, amiben részletes segítséget találunk egyrészt a pascal szintaktika és használható utasítások, másrészt a beépített könyvtárak függvény/eljárásait illetően.

A program fordítása *Project/Build* menüpontokkal lehetséges, az esetleges szintaktikai hibákról a fordító a szerkesztő alatti *Messages* ablakban tájékoztat. Ugyanitt jelenik meg a sikeres fordítás után előálló – PIC-be beírandó – programkód hossza, valamint a felhasznált RAM méret.

Megj.: A PIC RAM memóriája – változóink helye – igencsak véges: **368 byte!** Ezt a tényt végig tartsuk szem előtt a program tervezése és írása közben! Nem egy PC-re írunk "mamutprogramot" mindenféle összetett adatstruktúrák és objektumok deklarálásával, hanem egy mikrovezérlőre!

A kész, hiba nélkül lefordított programjainkat "debuggolhatjuk", a változók, regiszterek értékeit nyomon követhetjük a futás során. Ehhez előbb indítsuk el a nyomkövetőt Run/Start Debugger, majd a Watch ablakban válasszuk ki a megfigyelni kívánt változók és regiszterek neveit. A szerkesztő ablakban levő programsorokat a szövegkurzorral F4-el egyhuzamban végrehajtja, F7 és F8 a lépésenkénti végrehajtást szolgálja, azzal a különbséggel, hogy függvény/eljárás esetén az előbbi belép, míg az utóbbi egylépésben hajtja végre. Mindezek mellett a szerkesztőablak bal margóján töréspontokat (Breakpoints) is bekapcsolhatunk.

Az előbbi sorokban említett nyomkövető rendszer egy nagyon hatékony eszköz programunk logikai (amikor nem úgy működik, ahogy elképzeltük) hibáinak kiszűrésére.

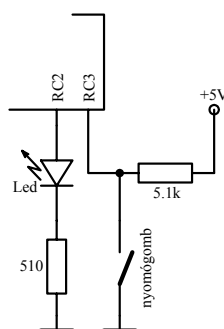
A kész lefordított program letöltése PIC-be

Tools/mikroBootloader menüpontokkal indítsuk el a letöltő programot. Az egérkurzorral álljunk rá, de még ne nyomjuk meg a *Connect* gombot! Előbb a PIC resetjét nyomjuk meg a próbalaapon, majd ezt követően néhány másodpercen belül a *Connect* gombot. Ha jól csináltuk (PIC táp alatt, soros kábel a helyén, gombnyomkodás sorrendje helyes) a *History Window*-ban azt látjuk: "*Connected*", az *Open HEX file*-al válasszuk ki a letöltendő hex formátumba fordított PIC programot, majd *Start bootloader*. Ekkor látható módon elindul a program letöltése. A letöltés végét a PIC újraindítására figyelmeztető üzenet jelzi. Hajtsuk végre a PIC újraindítását a reset gombbal!

Feladatok:

1. Led villogtatás, azaz a mikrovezérlő digitális ki- bemeneteinek megismerése, használata

A műszaki jellemzők között láttuk, hogy mikrovezérlőnk A, B, C portokkal (*PortA*, *PortB*, *PortC*) rendelkezik, amelyek rendre 5, 8, 8 bitesek. Ezen bitek mindegyike használható digitális ki- és bemenetként. A portokhoz kapcsolódó irányregiszterek tartalmi szabják meg azt, hogy egy adott bit éppen bemenet-e, avagy kimenet. Kimenet akkor, ha az irányregiszter megfelelő bitje nulla, egyébként bemenet. Alapértelmezésben a portok összes bitjei bemenetek. Az irányregiszterek elnevezései *TrisA*, *TrisB* és *TrisC*, ezek mindegyike programból szabadon írható olvasható.



LED és nyomógomb csatlakoztatása a PIC-hez

Minimális mérési összeállításunkat egészítsük ki az előbbi kis ábrán látható kapcsolással, azaz a C port 3.-ik bitjét használjuk bemenetként és egy nyomógommbal (vagy kapcsolóval) határozhatjuk meg logikai állapotát (elengedett nyomógomb → magas szint → logikai 1, megnyomott gomb → alacsony szint → logikai nulla)!

C portunk 2.-ik bitje viszont legyen digitális kimenet, ha éppen magas logikai szintű Ledünk világítani fog, ha alacsony a Led nem világít!

A hardver összeállításához írjunk programot úgy, hogy elengedett nyomógomb mellett egy bizonyos kitöltési tényezővel villogjon a Led, megnyomott nyomógomb esetén egy másik kitöltési tényezővel!

```
program led_villog;
var kapcsolo:boolean;
{-----}
procedure Villan(tbe,tki:word);
begin
  PortC.2:=1; VDelay_ms(tbe);
  PortC.2:=0; VDelay_ms(tki);
end;
{-----}
begin
  TRISC:=TRISC AND %11111011;      //C2 legyen kimenet
  TRISC:=TRISC OR  %00001000;     //C3 legyen bemenet
  while true do begin
    kapcsolo:=PortC.3;
    if kapcsolo then Villan(300,200) else Villan(50,450);
  end;
end.
```

A főprogram elején beállítjuk a C irányregiszter 2. és 3. bitjét. A másodikat töröljük, a harmadikat 1-be állítjuk, a többit nem bántjuk! Szokjuk meg ezt a "programozási fogást" még akkor is, ha jelen helyzetben mondhattuk volna, hogy bekapcsolás után a TrisC összes bitje 1, azaz nekünk csak a 2.-at kell nullázni, tehát: `TRISC:=$FB;` írjuk a két TrisC-s sor helyett.

Ezután a főprogram egy végtelen ciklussal folytatódik, vagyis rendre villantjuk a Ledet a nyomógomb állásától függő tbe, tki paraméterekkel meghatározott módon.

2. Az előre elkészített LED sorral készítsünk alkalmazást (futófény , stb.) – amely egy kapcsoló állásától függően különbözően működik.

A programunkat és a működő alkalmazást dokumentáljuk és mutassuk be a gyakorlatvezetőnek.