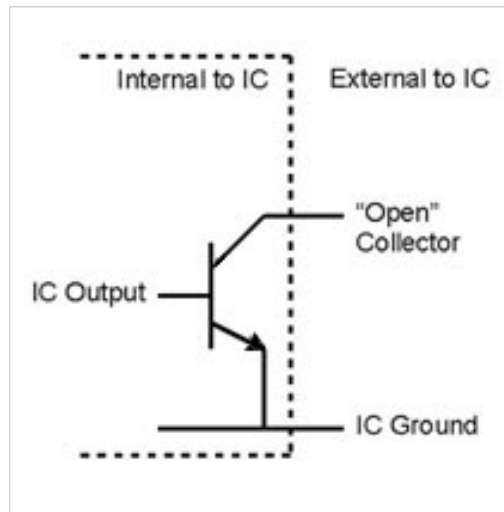


Mikrokontrollerek tételsor

"A" kérdések

Milyen a nyitott kollektoros kimenet?

[1]



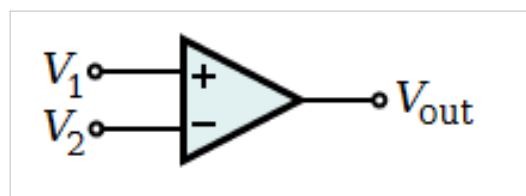
Gyakorlatilag egy tranzisztor aminek a kollektora az IC-n kívülre néz, könnyen lehet vele AND és OR kapcsolásokat csinálni. Általában a tranzisztor bázisa egy IC kimenetére csatlakozik, emittére a földre, a kollektort pedig "nyitott", azaz alaphól nincs kötve semmihez. Egy felhúzó ellenállással fel lehet kötni bármilyen feszültségre (nem kell, hogy az IC tápfeszültsége legyen), így összekapcsolható az adott nyitott kollektoros kimenettel rendelkező IC-nk egy másik feszültségen működő áramkörrel. (Ugyanis ha a tranzisztor nyitott állapotban van, akkor a felhúzó ellenálláson keresztül kapott feszültséget kapjuk, egyébként meg a földet.)

Hogyan készíthetünk stabilizált tápfeszültség forrást?

Talán ilyenekre gondolnak? [2] Esetleg egy egyszerű teljesítmény tranzisztor, ami a feszültségkülönbséget hővé alakítjuk. [3]

Ismertesse az analóg komparátor működését!

Talán ilyesmikre gondolnak? [4] [5] A lényeg, hogy két bejövő feszültség esetén a kimenetet ahhoz mérten állítja be, hogy melyik nagyobb. Ha a V_1 nagyobb, mint a V_2 , akkor az elérhető legnagyobb feszültséget adja ki (ez általában a tápfesz), fordított esetben pedig a legkisebbet (ez meg általában a föld).



Az 555-ös időzítő áramkör felépítése és alkalmazásai.

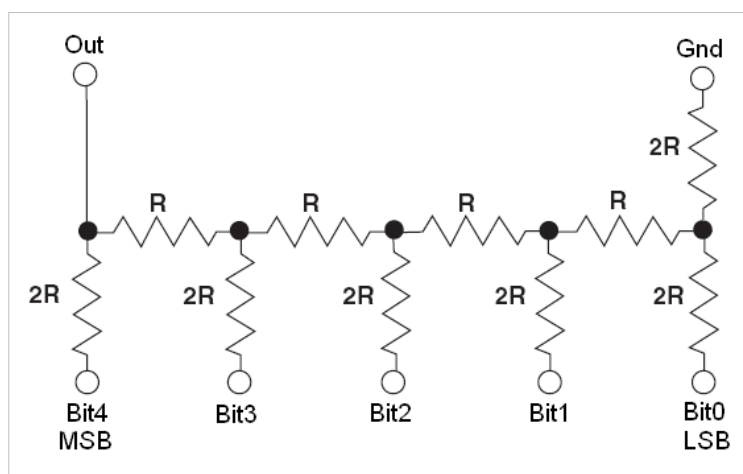
[6] [7]

- Lehet használni monostabilként: bejön egy jel, és egy meghatározott impulzust ad ki ennek hatására [8].
- Lehet használni astabilként: meghatározott frekvenciájú négyszögjelet ad ki ekkor [9].
- És lehet bistabilként használni: ekkor a triggerre magasba megy, resetre visszajön belőle [10].

A beütésszám átlagmérőknél, digitális méréstechnika laboron például 555-öt használtunk monostabil üzemmódban ahhoz, hogy a beérkező beütéseket azonos lefutású impulzusokká konvertáljuk.

Ellenállás létra és R-2R felépítése és használata D/A konverterekben.

[11]



És $V_{OUT} = V_{REF} \times VAL / 2^N$, ahol V_{REF} a referencia feszültség (általában +5V), VAL az ábrázolt szám (pl. 3), N a használható bitek száma (pl. 2 - két biten a maximum ábrázolható érték 3). Ekkor: $V_{OUT} = 5 \times 3 / 4 = 3.75$

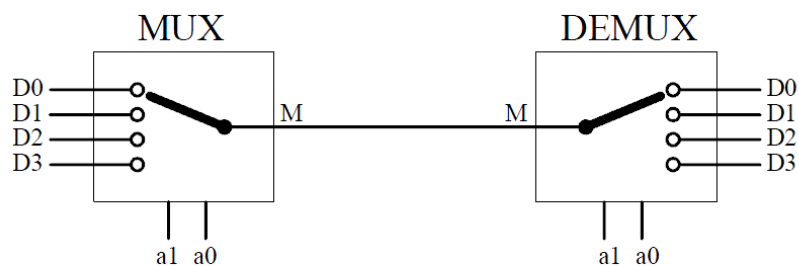
Műveleti erősítők alkalmazási lehetőségei TTL környezetben.

Logikailag más a működési karakterisztikájuk, ezért TTL vagy CMOS környezetben csak kisebb trükkökkel építhetők be: [12]. Valamint megfelelő kiegészítő áramköri elemekkel különböző feszültségeken lehet működtetni őket, ezért a különböző feszültségen működő áramkörök kapcsolhatóak össze velük.

Magyarázza meg, mik a multiplexerek, demultiplexerek, dekódolók

[13] [14]

- **Multiplexerek, demultiplexerek**



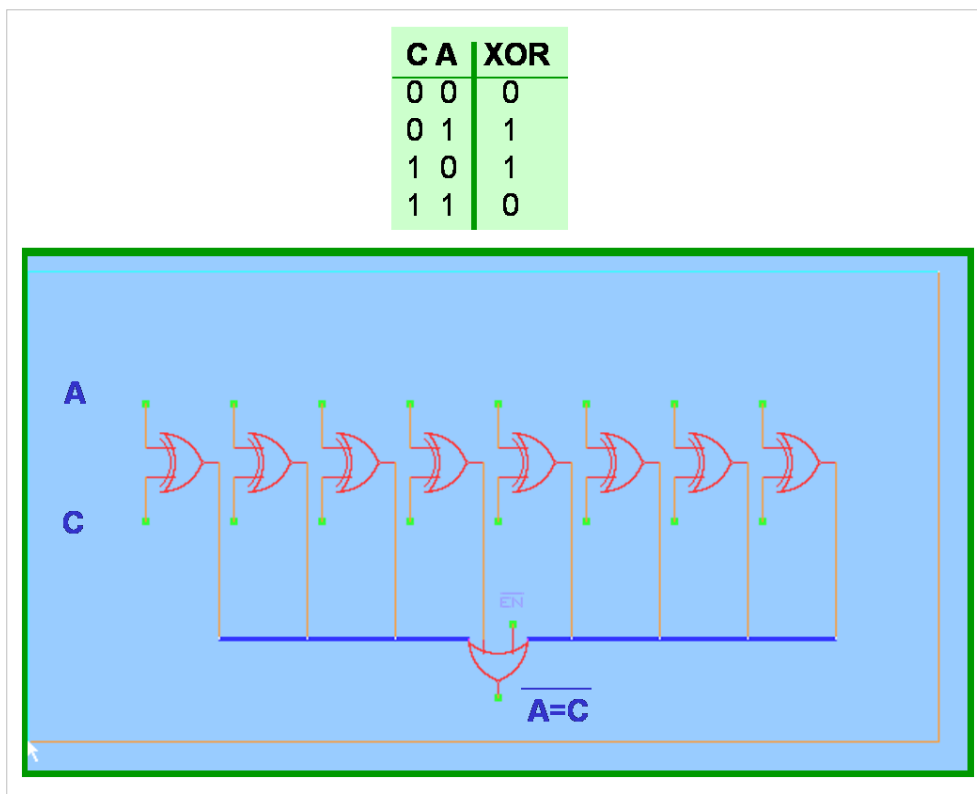
Alkalmazás:

- átviteltechnika
- tetszőleges logikai fgv. megvalósítása MUX segítségével

A demultiplexerek speciális dekódolók, például címeket lehet velük dekódolni.

Hogyan működik a digitális komparátor?

[15] A digitális komparátor összehasonlítja két bejövő, bináris jelet és attól függően ad ki nullát vagy egyet, hogy a két bejövő jel megegyezik, vagy nem. (Nullát, ha megegyeznek, és egyet, ha nem). Ezt általában úgy érik el, hogy a két jelet kivonják egymásból. Ha megegyeztek, akkor nulla az eredmény. Ha az eredmény nem nulla, akkor egyet ad vissza a rendszer.

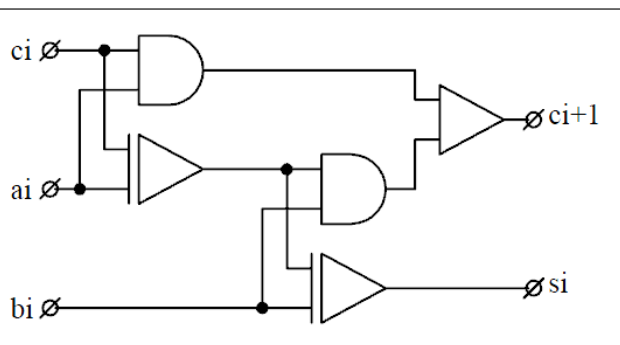


Hogyan működik a teljes összeadó?

[16] Szemben a félösszeadóval, ez lekezelet a carry-t (maradékot) ezért láncolható, így elvileg tetszőlegesen hosszú számok összeadása megvalósítható vele.

Teljes összeadó

c_i	a_i	b_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

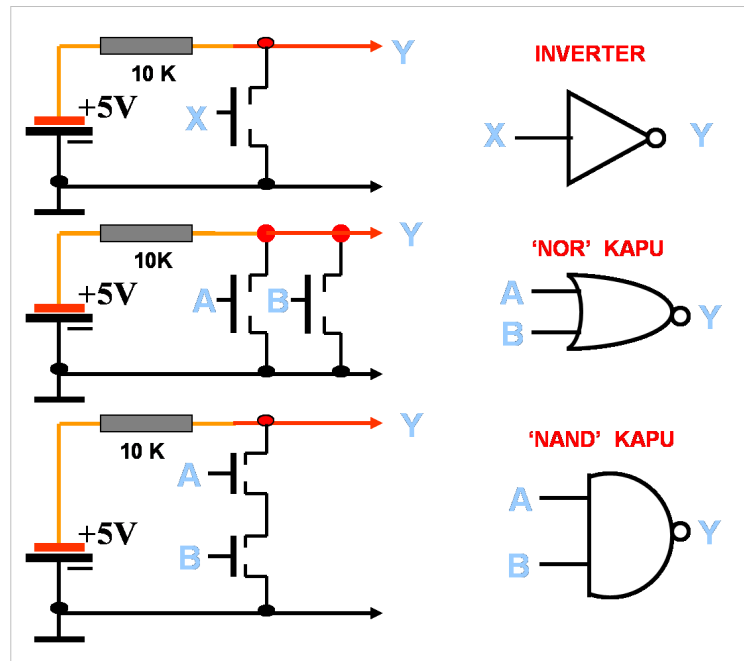


Milyen feladatokra használhatók a tároló regiszterek és osztók?

A tároló regiszter adatot tárol, és ezeken mennek az ALU-ban a behuzalozott műveletek. Az osztó arra jó, hogy amikor adott egy számláló, és a különböző helyiértékű biteiről veszünk le jelet, akkor az az alap órajel kettő hatványú leosztásával fog változni.

Hogyan lehet nyomógombos kapcsolót TTL bemenethez illeszteni?

Felhúzó ellenállással a következő módokon:



Hogyan lehet láthatóvá tenni egy kimenet logikai állapotát?

A legáttekinthetőbb megoldás, ha ráteszünk két műveletierősítőt, egyiket a föld, másikat a táp felé, és ezekről hajtunk meg LED-eket. Ekkor nem terheljük túl, a kisáramra tervezett alkatrészeket.

Ismertesse az I/O ciklus idődiagramját!

Először az Address Enable bit felmegy magasba (a negáltja lemegy alacsonyba). A be vagy kimenet akkor fejeződik be, ha ez a bit visszaáll az eredeti értékére. Kimenetkor először az Adat vezeték felveszi az adat által definiált értéket, majd az IOW negáltja lemegy, és az ezutáni felfutásnál megy ki az adat.

Bemenetnél hasonlóan megy, csak rövidebb időszelvényben és az IOR bittel szabályozva.

Az output port felépítése.

Ha az adatbusz által adott cím egyezik az adott port címével, akkor az adatbusz komparátora magasat ad, és ha ekkor az IOW is magas (a negáltja alacsony) akkor engedélyeződik az adat kivitele ezzel az AND-el jellel mint órajellel vezérelve az adatbuszon lógó fliflopokat.

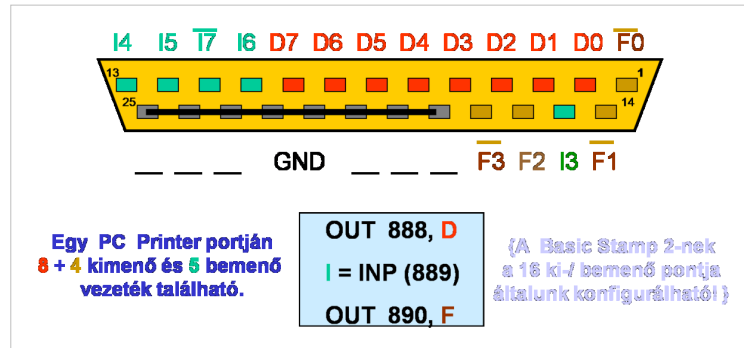
Az input port felépítése.

Hasonló az Output porthoz, csak az engedélyezést az IOR bit adja, és az adatbuszon nem fliflopok, hanem ugynevezett tristate-ekből [17] alkotott bufferek ülnek. Ezek nem csak 1 és 0 szintet tudnak felvenni, hanem harmadik állapotként magas impedanciájú szintet is, aminek hatására tulajdonképpen lekapcsolódnak az áramkörrel. Ez azt jelenti, hogy sok ilyen buffert lehet rákötni egyetlen buszra, és azok (ha csak 1 van kiválasztva, a többi a

magas impedanciájú állapotban van) akkor nem fogják egymást zavarni.

A PC printer portjának felépítése.

Ez gyakorlatilag az előző két portfajta egybeépítése, annyi extrával, hogy az adatvezeték 2 bitjét használják még két féle input és a bemenet kiválasztására.



Ismertesse az ALU felépítését és működését!

Az ALU az aritmetikai és logikai egység végzi a legalapvetőbb adat manipulációs műveleteket regiszterek adatai között. Vázlatosan: az utasítás dekódolásával kiválasztja a végrehajtandó műveletet (és esetleg forrás és célregiszter is) és az órajel megérkezésekor aktiválja azt az áramkört ami az utasításnak megfelelő műveletet végrehajtja. Az átmeneti tároló a W (working) regiszter a PIC esetén. Valamint elvégzi az interruptok és a stack kezelését is.

Ismertesse a CLEAR/SET utasítás működését.

Ha itt a BCF (Bit Clear f) és BSF (Bit Set f) PIC utasításokra gondolnak, akkor ezek az f regiszter b-edik bitjét állítják be, vagy törlik ki (b 3 biten jön be, ez címzi a 8 bitjét f-nek, f pedig a fileregiszter azonosítja 7 bit-en) . f szerint multiplexelés van, azaz az utasításban az f-et kódoló rész válsztja ki az aktív áramkört, a 3 b-t pedig szintén multiplexelik, hogy az adott bitet kiválaszthassák.

Milyen alapvető digitális áramkörökből épül fel a számítógép?

Multiplexerek, demultiplexerek, komparátorok, logikai kapuk, regiszterek...

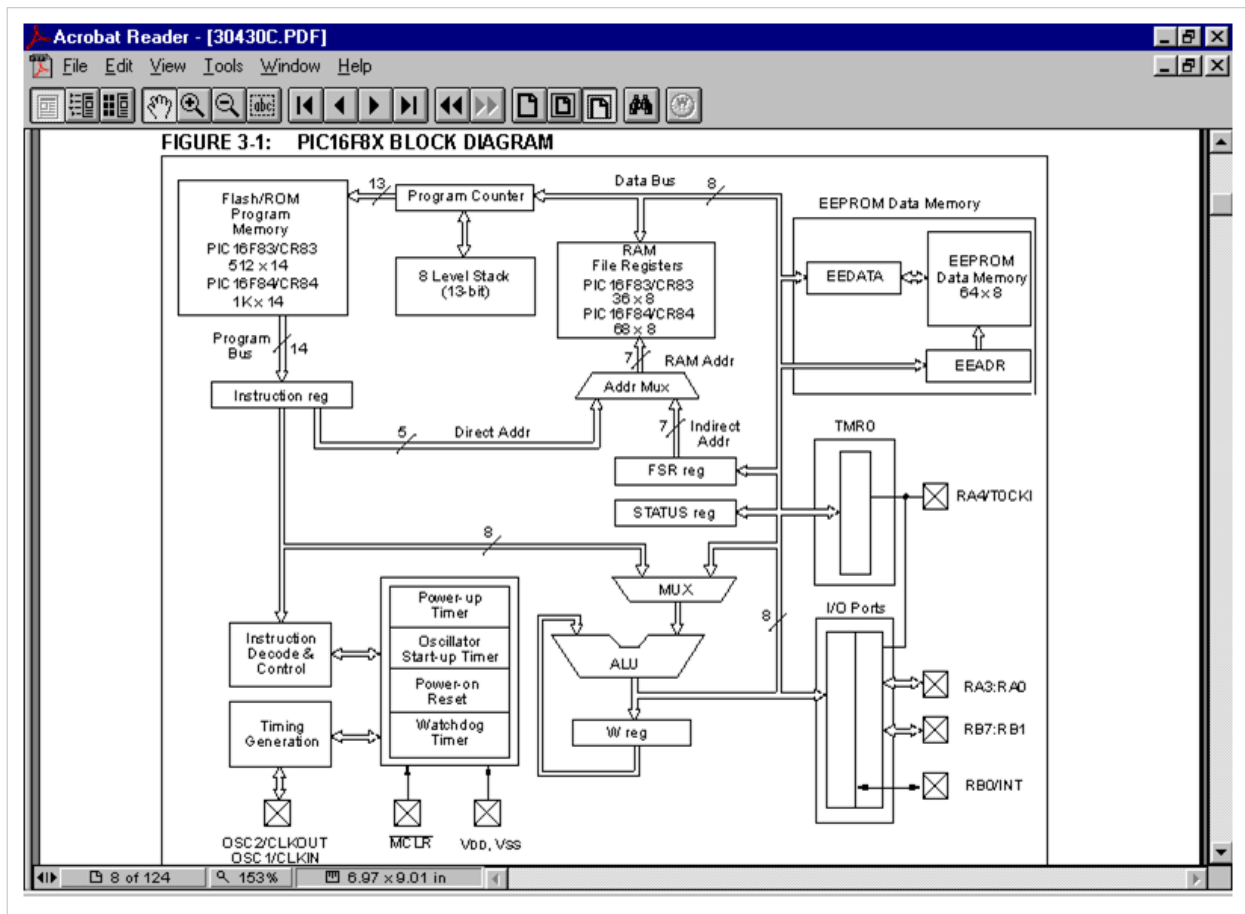
Magyarázza el, mi a Harvard-felépítés és hasonlítsa össze a Neumann-felépítésű számítógéppel

[18] [19]

A Harvard az utasításokat (programkódot) független helyen és független dedikált eléréssel különíti el az adatoktól. Ezzel szemben a Neumann közös memóriát és sávcsélt használ a programkód és az adatok elérésére, változtatására, ezáltal felxibilisebb, de valamivel lassabb. A mai komplex processzorok Neumann elvűek, de specifikus chipekben, pl jelfeldolgozásban, ahol a program ritkán kell változzon Harvardot használnak.

A blokkvázlat alapján mutassa be a PIC mikrovezérlők működését!

Logic.ppt 42. oldal



Csoportosítsa a PIC mikrovezérlők lábkievezéseit!

Alapvető működéshez: Táp, föld, reset, oszcillátor Kimeneti/bemeneti csatlakozók..

A jegyzet csoportosítása szerint: adatbusz, címbusz, kimenetek, bemenetek.

Milyen lépésekből áll egy utasítás végrehajtása, és mi a pipe-line?

Utvégrehajtás lépései: utelőkészítés, lehívás; utszámáláló reg tartalmának növelése; műveleti kód értelmezése, dekódolás, operandus címének meghatározása; a művelethez szükséges adatok előkészítése; végrehajtás; az eredmény elhelyezése. Pipe-line: míg az egyik utasítást végrehajtjuk, addig lehetséges a következő utasítás memóriából történő lehívása és dekódolása; ha az éppen végrehajtandó utasítás ugró utasítás, akkor nem jól működik (ilyenkor a következő utasítást az ugrás helyéről kell betölteni, a már lehívott utasítást pedig el kell dobni).

Csoportosítsa, és röviden mutassa be a PIC mikrovezérlők utasításkészletét!

Bitekre ható műveletek, Byteokra ható műveletek, és vezérlési, illetve "literal" műveletek. Ez utóbbiak pl. forráskódba írt konstans kifejezések (pl. számok).

Milyen feladatot lát el a veremtár?

A verem egy last in first out tároló. A függvényhívások, és visszatérések kezelésekor játszik szerepet, ugyanis itt tárolódnak a visszatérési címek, regiszterek értékei, esetleg átadandó paraméterek.

Két operandust igénylő utasítások esetén hol képződik az eredmény?

Egy d bittől függően vagy a W regiszterbe, vagy az utasításban használt file regiszterbe kerül.

Magyarázza el a státuszbitek jelentését!

Legfőképpen 3 státuszbit található a PIC-ben, az egyik a Z (Zero) bit, ez logikai összehasonlításoknál, illetve egyéb más utasításoknál állítódik be (például egyenlőség tesztje esetén) és ezt felhasználja egy későbbi utasítás (például ugrásra): Ez a PIC-ben az increment és decrement utasításoknál van például használva, ahol ha a számlálás nullát ad, akkor az utána következő utasítást nem hajtja végre, átugorja azt, ez lehet például egy ciklusból való kilépés.

A másik két DC (Digit Carry) és C (Carry) a maradékok kezelését látja el, összeadásnál és kivonásnál.

Miért van szükség memórialapok kialakítására, és mi az a lapváltás?

Ha több a memória, mint amit az adott operandusban indexelni lehet, akkor a hiányzó biteket valahonnan máshonnan olvassa az utasítás. A PIC-en van két bit (1. alább), ezeket használja az utasításbeli 11 mellé. A lapozás folyamata, amikor ezeket a biteket megváltoztatjuk, onnantól minden címzés máshova fog hivatkozni, más lapra.

Mir az RP0, RP1, IRP bitek

Az első kettő az említett kiegészítő bitek, ezek adják meg, hogy melyik lapon dolgozunk. Innen következik az is, hogy 4 lap van a PIC-ben. Az IRP bit az indirekt címzéshez kell, ekkor a filregister+ az IRP bitje határozza meg a lapot és azon belül a címet az utasítás végrehajtásához (gyakorlatilag az f regiszter által mutatott címet használja).

Mi a RESET feladata, és hogyan működik ez a PIC-eknél?

Alaphelyzetbe állítja a PIC-et. Ha van bootloader program az indul el először. Minden port bemenetre állítódik, az adatregisztereket nullára állítja, az ADC kontrol regiszter is nullára állítódik (azaz analóg bemenet lesz).

Órajel generálása PIC-eknél.

- Kisfrekvenciás kvarzkristályos: alacsony fogyasztás.
- Kvarc vagy kerámia rezonátor: normál működés.
- Nagyfrekvenciás kvarc vagy rezonátor: gyors működéshez (4 MHz felett).
- Külső RC elemekből felépített oscillátor.

Hogyan valósítható meg az indirekt címzés

Lásd feljebb.

Milyen célt szolgál a Watch Dog Timer?

A WDT egy önálló, független oszcillátorról járatott számláló, amely ha lejár, akkor reseteli a PIC-et. Ez például valamilyen külső behatásból, ami megakasztja, vagy kiakasztja a PIC-et, szolgál visszatérésként.

Mi a polling? Mi a megszakítás (interrupt-IT)?

Pollingnak nevezzük azt a megoldást, amikor folyamatosan olvassuk egy külső bemenetről, hogy ott van-e változás, vagy érkezik-e adat. Interrupt az, amikor ha beérkezik egy jel, akkor az megszakítja a programkód vérehajtását, elugrik egy bizonyos másik programkódrészhez, azt végrehajtja, majd ha végzett vele, akkor folytatja az eredeti programot. Interruptokkal lehet például hatékonyan megoldani a perifériák kezelését.

Milyen időbeli viszonyokra kell ügyelnünk a megszakítások használatánál?

Interrupt feldolgozása alatt másik interrupt nem jöhet, addig le kell tiltani a megszakításokat. Továbbá van egy rövid 3 usec-es késés, belső utasítások lefutása miatt.

Hogyan lehet programozni a PIC-eket?

Feltöltünk a bootloaderbe egy olyan programot, ami soros USART kommunikációt próbál meg létesíteni a PIC és valami külső eszköz között. Ha sikerül, akkor kezeli ezt, és például programot tölt le, ha nem sikerül, akkor egy korábban feltöltött programkód végrehajtásával folytatja a működést. Az áttöltött programkód gépkódú utasítások sora, ezt valamilyen más eszközön, például PC-n állítjuk elő, általában valamilyen szerkesztő és fordító programmal.

Ismertesse röviden a PIC perifériákat.

- 3 Port (A, B, C), ezek közül az egyik digitális és analóg is lehet, a másik kettő csak digitális. További portok is vannak, nagyrészt nem foglalkoztunk velük (D, E, ezek külön programozható lábak, ki, bemenetre).
- 3 Timer (2 8 bites, egy 16 bites), átskálázható.
- 10 bites AD konverter.
- Univerzális Szinkron-Aszinkron Fogadó-Küldő Port (USART).
- Jelszélesség Moduláló (PWM).

Van más is, azokkal nem foglalkoztunk.

PIC I/O portok működése.

Használat előtt a portokhoz tartozó beállító regisztereken (TRISX) be kell állítani, hogy ki- vagy bemenetként szeretnénk őket használni. Ezen felül az A portnál beállítható analóg üzemelés is. Ezek után a PortX regisztereken keresztül írhatjuk, olvashatjuk a portokat.

Számlálók és időzítők a PIC-ben. TMR0, TMR1, TMR2

A számláló áramköröket perifériaként használva a processzort tudjuk tehermentesíteni: a számláló bemenetére egy külső esemény esetén megjelenő jelváltás kapcsolva, a számláló önállóan képes a külső események számlálására. A processzornak csak le kell kérdezni a számláló tartalmát. Ha a kontrollert működtető leosztott órajelet jutattunk erre a számláló bemenetre, akkor a számláló tartalma és az órajel periódus idejének szorzata az eltelt idővel arányos. A számláló által átfogható számlálási tartomány növelése miatt a számlálót alkotó flip-flopokból álló számlálólánc hosszát növelik meg.

- TMR0: időzítő áramkör; 2 bemenetű multiplexe; kontroll reg-gel beállítani: ha 0 - belső jel, ha 1 - külső jel.

- TMR1: 16 bites + felbontást növelő elosztó + bemenet lehetőség: egy kisfrekvenciás kvarcosszc.; létezik asszinkron számláló üzemmód.
- TMR2: 8 bites, soros adatátviteli egység ütemadójaként használjuk; periódusreg:n adott értékelérése->saját magát törli; van elő és utóosztója

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

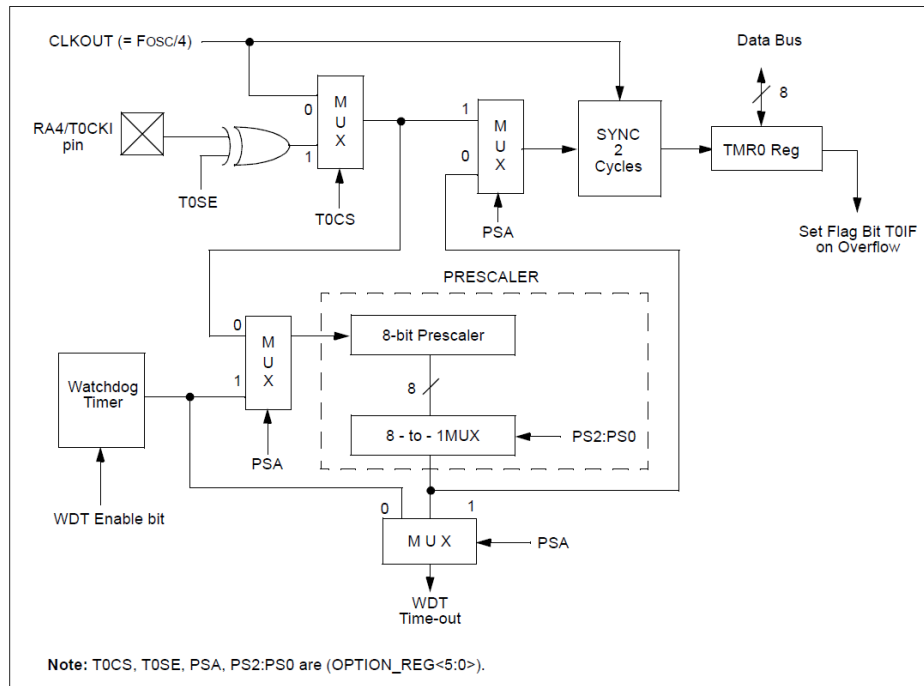
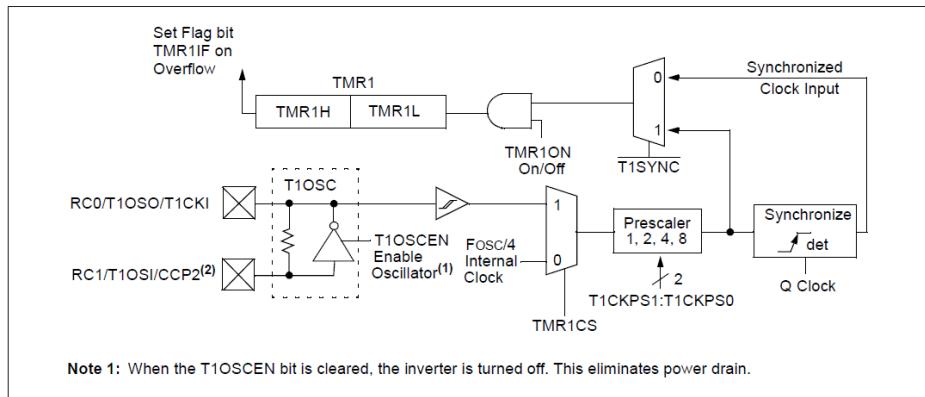
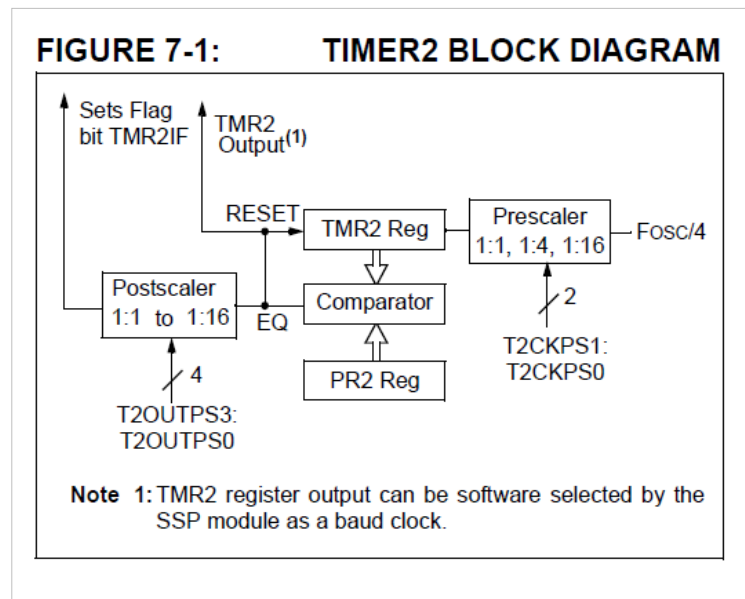


FIGURE 6-2: TIMER1 BLOCK DIAGRAM





PWM modulátor működési elve és felhasználási lehetőségei.

A PWM a négyszögjel szélességét tudja szabályozni, adott kitöltési tényező szerint (10-bites pontossággal), továbbá a frekvenciája is beállítható.

[20]

Használható telekommunikációban, különböző szélességű jelek kiadására, vagy teljesítmény finomszabályozására (pl. villanykörte fényereje), és használják még kapcsolóüzemű tápegységekben feszültség szabályozásra.

"B" kérdések

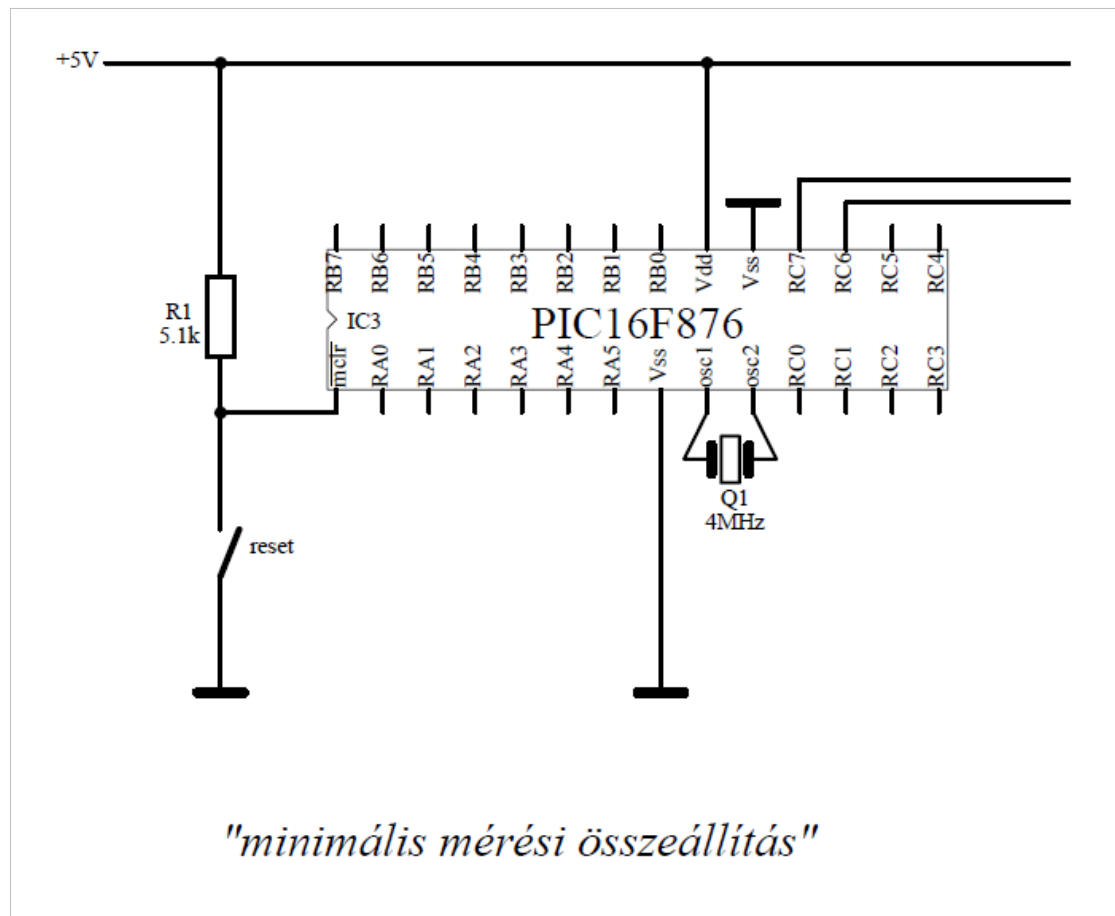
mikroPascal utasításkészlete és a fordító használata

A sztenderd pascal utasításokon kívül (melyek azért csak redukáltan érhetőek el), a mikroPascal tartalmaz egy sor beépített könyvtárat a különböző perifériák kezeléséhez és a különböző regiszterek programozásához. A perifériák kezeléséhez leggyakrabban használt könyvtárak az USART és az LCD könyvtárak. Ezekben találhatóak a kiíratásra, ki-, és beolvasásra stb. szolgáló parancsok. A különböző portok, regiszterek stb. a dokumentációban közölt néven érhetőek el. Például a `PortA.1 = 0;` utasítás az A port első kimenetét nullára állítja.

A PIC16F86-os mikrovezérlő (továbbiakban PIC) felépítése

Nincs jobb ötletem, lásd feljebb.

A PIC működéséhez szükséges minimális hardware követelmények



Digitális I/O megvalósítás PIC-kel

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 -

Eből kiderül, hogy az alábbi laboron megszerzett ismereteket a rendelkezésre álló fejlesztési környezetben megvalósítottuk:

Mérés leírása

A laboratóriumi mérési összeállítás sikerült úgy megszervezni, hogy még programozó eszköz használata se legyen szükség! A flash programmemóriával rendelkező PIC-kben flash program képes módosítani a programmemória tartalmát. Ezért ekkor a PIC-kbe elengedő, ha egyszer flash programozó eszközzel egy ún. "boot" programot.

A boot program pici és a programmemória felül részén foglal helyet, szabadon hagyva a memória "alsó" felétnek nagy részét a felhasználói alkalmazás számára. Feladata, hogy bekapcsolást ill. resetet követő néhány másodpercen belül a PIC soros perifériáját (Uart), hogy átadja vele valaki adatot cserélje. Ha igen, és sikeresen azonosította a PC-t flash bootloader programot, akkor megkezdődik a PC felől jövő adatfolyam programmemória írás (a PIC programozása), ellenkező esetben lezárja, vagy meg sem kezd a nem létező soros kommunikációt. Ilyenkor (az esetek többségében, amikor nem programozunk, csak) egyszerűen használjuk a PIC-et néhány másodperces időre álló letétre utal, 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 – transzmissziós villamosan – egy külön nyomtatott áramkörön van felépítve, melyet egy csatlakozóval tudunk a PIC-hoz kapcsolni. A PIC 20-as lábait a csatlakozónak megfelelő lábbal kapcsoljuk, a további három láb a 19,18,17-es lábbal kerüljön.

Ehhez mindössze annyit fűznék hozzá, hogy a digitális I/O megvalósítása egyszerűen a portok megfelelő irányregisztereinek beállításával érhető el. Az A port esetén kell egyedül külön megmondani, hogy digitális, vagy analóg módban szeretnénk használni.

Analóg feszültség mérése PIC-kel



1. Feszültségmérés a kontrollrel:
Mikrovezetőnk egy 10 bites AD konvertert épített, amelyet megoldó egy 5 bitesnél analóg multiplexerrel, így akár 5 különböző analóg forrásból származó feszültséget is rendre az AD konverter bemenetére kapcsolhatunk. A multiplexer bemenetét az A port A_0, A_1, A_2, A_3, A_4 bityein érijük el. Az ADCON0 regiszter tartalma határozza meg, hogy ezekből melyik analóg és melyik a már korábban megemlített digitális bemenet. Azaz beállíthatjuk, hogy az AD konverzió során – az AD konverter egység – mit tekintsen referenciának.

ADCON1 REGISTER (ADDRESS 9Fh)							
U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

4.7 ADFM: A/D Result Format Select bit

1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.

Unimplemented: Read as 0

bit 3-0	PCFG3:PCFG0: A/D Port Configuration Control bits
---------	--

PCFGs	REC1	REC2	REC3	REC4	REC5	REC6	REC7	REC8	REC9	REC10	REC11	REC12	REC13	REC14	REC15	REC16	REC17	REC18	REC19	REC20	REC21	REC22	REC23	REC24	REC25	REC26	REC27	REC28	REC29	REC30	REC31	REC32	REC33	REC34	REC35	REC36	REC37	REC38	REC39	REC40	REC41	REC42	REC43	REC44	REC45	REC46	REC47	REC48	REC49	REC50	REC51	REC52	REC53	REC54	REC55	REC56	REC57	REC58	REC59	REC60	REC61	REC62	REC63	REC64	REC65	REC66	REC67	REC68	REC69	REC70	REC71	REC72	REC73	REC74	REC75	REC76	REC77	REC78	REC79	REC80	REC81	REC82	REC83	REC84	REC85	REC86	REC87	REC88	REC89	REC90	REC91	REC92	REC93	REC94	REC95	REC96	REC97	REC98	REC99	REC100	REC101	REC102	REC103	REC104	REC105	REC106	REC107	REC108	REC109	REC110	REC111	REC112	REC113	REC114	REC115	REC116	REC117	REC118	REC119	REC120	REC121	REC122	REC123	REC124	REC125	REC126	REC127	REC128	REC129	REC130	REC131	REC132	REC133	REC134	REC135	REC136	REC137	REC138	REC139	REC140	REC141	REC142	REC143	REC144	REC145	REC146	REC147	REC148	REC149	REC150	REC151	REC152	REC153	REC154	REC155	REC156	REC157	REC158	REC159	REC160	REC161	REC162	REC163	REC164	REC165	REC166	REC167	REC168	REC169	REC170	REC171	REC172	REC173	REC174	REC175	REC176	REC177	REC178	REC179	REC180	REC181	REC182	REC183	REC184	REC185	REC186	REC187	REC188	REC189	REC190	REC191	REC192	REC193	REC194	REC195	REC196	REC197	REC198	REC199	REC200	REC201	REC202	REC203	REC204	REC205	REC206	REC207	REC208	REC209	REC210	REC211	REC212	REC213	REC214	REC215	REC216	REC217	REC218	REC219	REC220	REC221	REC222	REC223	REC224	REC225	REC226	REC227	REC228	REC229	REC230	REC231	REC232	REC233	REC234	REC235	REC236	REC237	REC238	REC239	REC240	REC241	REC242	REC243	REC244	REC245	REC246	REC247	REC248	REC249	REC250	REC251	REC252	REC253	REC254	REC255	REC256	REC257	REC258	REC259	REC260	REC261	REC262	REC263	REC264	REC265	REC266	REC267	REC268	REC269	REC270	REC271	REC272	REC273	REC274	REC275	REC276	REC277	REC278	REC279	REC280	REC281	REC282	REC283	REC284	REC285	REC286	REC287	REC288	REC289	REC290	REC291	REC292	REC293	REC294	REC295	REC296	REC297	REC298	REC299	REC300	REC301	REC302	REC303	REC304	REC305	REC306	REC307	REC308	REC309	REC310	REC311	REC312	REC313	REC314	REC315	REC316	REC317	REC318	REC319	REC320	REC321	REC322	REC323	REC324	REC325	REC326	REC327	REC328	REC329	REC330	REC331	REC332	REC333	REC334	REC335	REC336	REC337	REC338	REC339	REC340	REC341	REC342	REC343	REC344	REC345	REC346	REC347	REC348	REC349	REC350	REC351	REC352	REC353	REC354	REC355	REC356	REC357	REC358	REC359	REC360	REC361	REC362	REC363	REC364	REC365	REC366	REC367	REC368	REC369	REC370	REC371	REC372	REC373	REC374	REC375	REC376	REC377	REC378	REC379	REC380	REC381	REC382	REC383	REC384	REC385	REC386	REC387	REC388	REC389	REC390	REC391	REC392	REC393	REC394	REC395	REC396	REC397	REC398	REC399	REC400	REC401	REC402	REC403	REC404	REC405	REC406	REC407	REC408	REC409	REC410	REC411	REC412	REC413	REC414	REC415	REC416	REC417	REC418	REC419	REC420	REC421	REC422	REC423	REC424	REC425	REC426	REC427	REC428	REC429	REC430	REC431	REC432	REC433	REC434	REC435	REC436	REC437	REC438	REC439	REC440	REC441	REC442	REC443	REC444	REC445	REC446	REC447	REC448	REC449	REC450	REC451	REC452	REC453	REC454	REC455	REC456	REC457	REC458	REC459	REC460	REC461	REC462	REC463	REC464	REC465	REC466	REC467	REC468	REC469	REC470	REC471	REC472	REC473	REC474	REC475	REC476	REC477	REC478	REC479	REC480	REC481	REC482	REC483	REC484	REC485	REC486	REC487	REC488	REC489	REC490	REC491	REC492	REC493	REC494	REC495	REC496	REC497	REC498	REC499	REC500	REC501	REC502	REC503	REC504	REC505	REC506	REC507	REC508	REC509	REC510	REC511	REC512	REC513	REC514	REC515	REC516	REC517	REC518	REC519	REC520	REC521	REC522	REC523	REC524	REC525	REC526	REC527	REC528	REC529	REC530	REC531	REC532	REC533	REC534	REC535	REC536	REC537	REC538	REC539	REC540	REC541	REC542	REC543	REC544	REC545	REC546	REC547	REC548	REC549	REC550	REC551	REC552	REC553	REC554	REC555	REC556	REC557	REC558	REC559	REC560	REC561	REC562	REC563	REC564	REC565	REC566	REC567	REC568	REC569	REC570	REC571	REC572	REC573	REC574	REC575	REC576	REC577	REC578	REC579	REC580	REC581	REC582	REC583	REC584	REC585	REC586	REC587	REC588	REC589	REC590	REC591	REC592	REC593	REC594	REC595	REC596	REC597	REC598	REC599	REC600	REC601	REC602	REC603	REC604	REC605	REC606	REC607	REC608	REC609	REC610	REC611	REC612	REC613	REC614	REC615	REC616	REC617	REC618	REC619	REC620	REC621	REC622	REC623	REC624	REC625	REC626	REC627	REC628	REC629	REC630	REC631	REC632	REC633	REC634	REC635	REC636	REC637	REC638	REC639	REC640	REC641	REC642	REC643	REC644	REC645	REC646	REC647	REC648	REC649	REC650	REC651	REC652	REC653	REC654	REC655	REC656	REC657	REC658	REC659	REC660	REC661	REC662	REC663	REC664	REC665	REC666	REC667	REC668	REC669	REC670	REC671	REC672	REC673	REC674	REC675	REC676	REC677	REC678	REC679	REC680	REC681	REC682	REC683	REC684	REC685	REC686	REC687	REC688	REC689	REC690	REC691	REC692	REC693	REC694	REC695	REC696	REC697	REC698	REC699	REC700	REC701	REC702	REC703	REC704	REC705	REC706	REC707	REC708	REC709	REC710	REC711	REC712	REC713	REC714	REC715	REC716	REC717	REC718	REC719	REC720	REC721	REC722	REC723	REC724	REC725	REC726	REC727	REC728	REC729	REC730	REC731	REC732	REC733	REC734	REC735	REC736	REC737	REC738	REC739	REC740	REC741	REC742	REC743	REC744	REC745	REC746	REC747	REC748	REC749	REC750	REC751	REC752	REC753	REC754	REC755	REC756	REC757	REC758	REC759	REC760	REC761	REC762	REC763	REC764	REC765	REC766	REC767	REC768	REC769	REC770	REC771	REC772	REC773	REC774	REC775	REC776	REC777	REC778	REC779	REC780	REC781	REC782	REC783	REC784	REC785	REC786	REC787	REC788	REC789	REC790	REC791	REC792	REC793	REC794	REC795	REC796	REC797	REC798	REC799	REC800	REC801	REC802	REC803	REC804	REC805	REC806	REC807	REC808	REC809	REC810	REC811	REC812	REC813	REC814	REC815	REC816	REC817	REC818	REC819	REC820	REC821	REC822	REC823	REC824	REC825	REC826	REC827	REC828	REC829	REC830	REC831	REC832	REC833	REC834	REC835	REC836	REC837	REC838	REC839	REC840	REC841	REC842	REC843	REC844	REC845	REC846	REC847	REC848	REC849	REC850	REC851	REC852	REC853	REC854	REC855	REC856	REC857	REC858	REC859	REC860	REC861	REC862	REC863	REC864	REC865	REC866	REC867	REC868	REC869	REC870	REC871	REC872	REC873	REC874	REC875	REC876	REC877	REC878	REC879	REC880	REC881	REC882	REC883	REC884	REC885	REC886	REC887	REC888	REC889	REC890	REC891	REC892	REC893	REC894	REC895	REC896	REC897	REC898	REC899	REC900	REC901	REC902	REC903	REC904	REC905	REC906	REC907	REC908	REC909	REC910	REC911	REC912	REC913	REC914	REC915	REC916	REC917	REC918	REC919	REC920	REC921	REC922	REC923	REC924	REC925	REC926	REC927	REC928	REC929	REC930	REC931	REC932	REC933	REC934	REC935	REC936	REC937	REC938	REC939	REC940	REC941	REC942	REC943	REC944	REC945	REC946	REC947	REC948	REC949	REC950	REC951	REC952	REC953	REC954	REC955	REC956	REC957	REC958	REC959	REC960	REC961	REC962	REC963	REC964	REC965	REC966	REC967	REC968	REC969	REC970	REC971	REC972	REC973	REC974	REC975	REC976	REC977	REC978	REC979	REC980	REC981	REC982	REC983	REC984	REC985	REC986	REC987	REC988	REC989	REC990	REC991	REC992	REC993	REC994	REC995	REC996	REC997	REC998	REC999	REC1000
60002	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A																																																																																																																																																																																																																																																																																																																																																																																																																															

3311	0	0	0
------	---	---	---

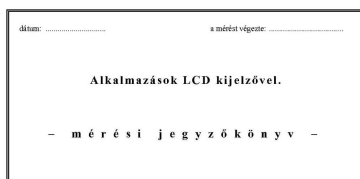
A = Analog Input D = Digital I/O

PCFG3...PCFG0 bitek jelentését tanulmányozzuk a fenti táblázatban. Ha pl. a konverzió pontossága nem olyan fontos és megelégszünk, a mikrovezérlő típusfelismerésével referencia gyanánt, akkor használjuk nyugodtan

-2-

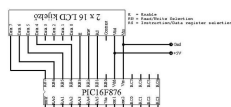
Teljes leírásként linkelném be ide

LCD panel illesztése PIC-hez



1. LCD kijelző csatlakoztatása a mikrokonrollerhez:

A kijelző 1-es lábát a mikrokontroller 19-es lábától csatlakoztatjuk a breadboardon. Teszteljük le a kijelzőt.



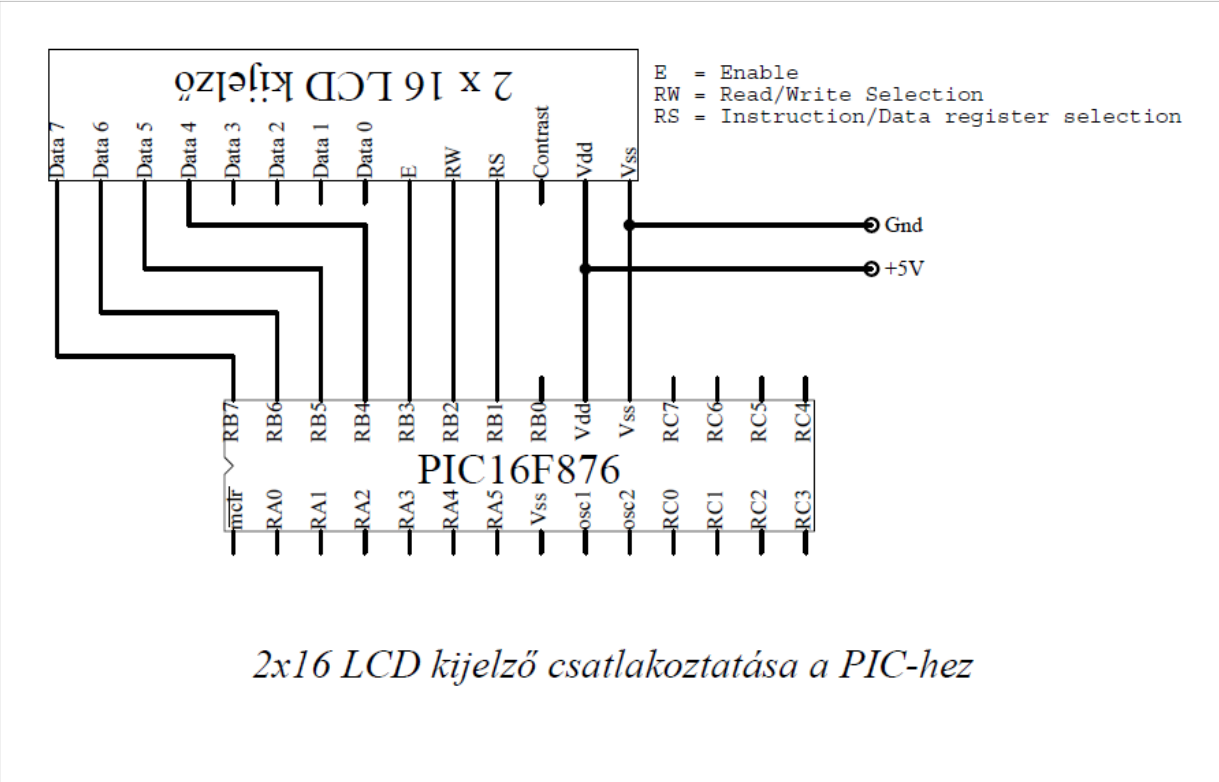
2x16 LCD kijelző csatlakoztatása a PIC-hez

```
var i:byte;
t1:=taring[16];
t2:=taring[16];

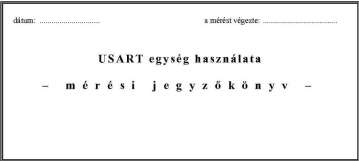
begin
i:=0;
t1:='*** LCD testt ***';
t2:='';
Lcd_Config(PORTB,7,6,5,4,PORTB,1,2,3);

while True do begin
t2[i and 16]:=32+2;
Lcd_Cmd(LCD_CLEAR);
Lcd_Cmd(LCD_CURSOR_OFF);
Lcd_Out(1,1,t1);
Lcd_Out(2,1,t2);
i:=i+1;
Delay_ms(500);
end;
end;
```

-2-



Soros kommunikáció PIC és PC között



1. Soros kommunikáció a PIC-ben futó felhasználói program és a PC között, az USART egység használata

A PIC-be épített USART (USART=Universal Synchronous Asynchronous Receiver Transmitter) egységet regisztereken keresztül az alábbi módokban lehet felprogramozni és használni:

- asinkron kommunikáció,
- szinkron kommunikáció - Master üzemmód,
- oldalsó kommunikáció - Slave üzemmód.

Az elől többnyire PIC és PC közötti soros adatátvitelhez célszerű használni, míg a következő kettő PIC és perifériák (AD konverter, EEPROM) közötti kommunikációt segíti.

A mikroPascal szűrekre tartalmaz egy USART Library beépített könyvtárat az USART egység kezelésére, így annak beállításait a PIC regiszterekhez írták helyett egyszerű parancsokkal végeztethetjük.

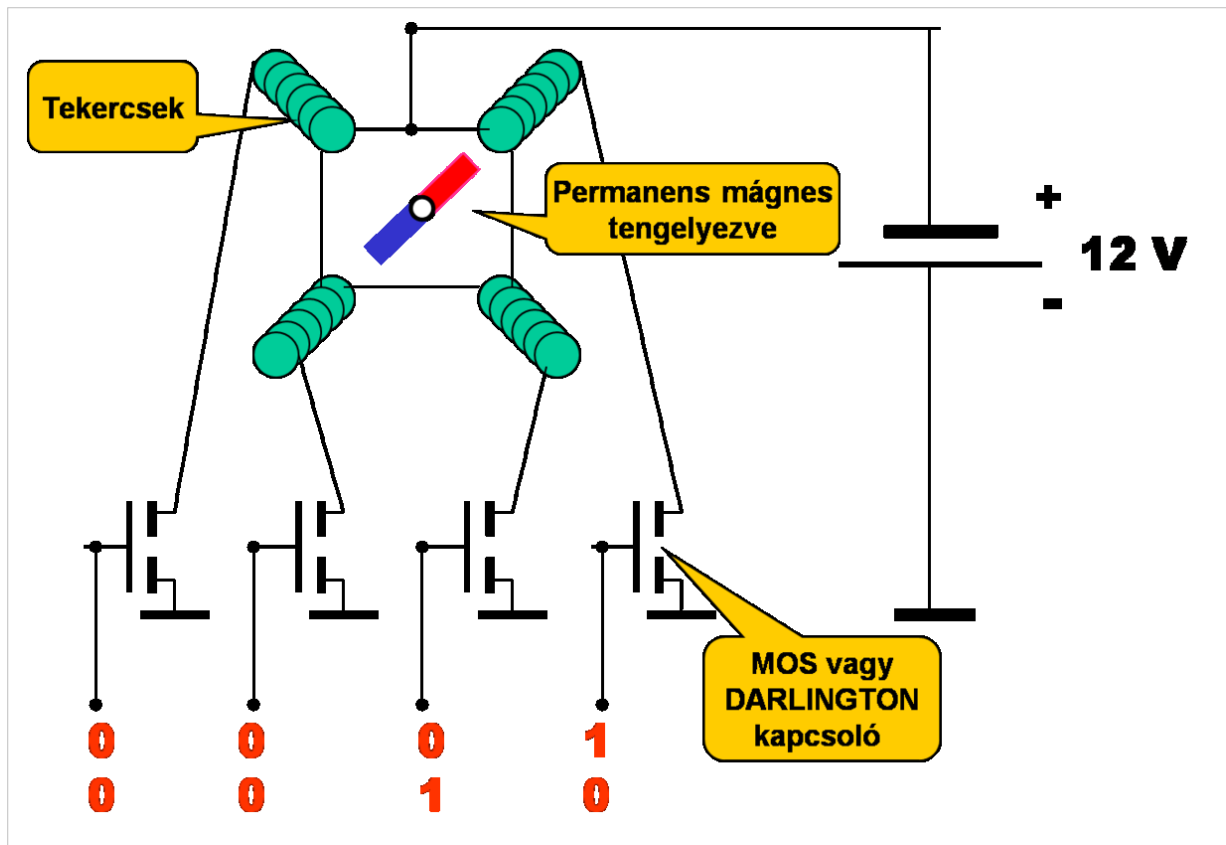
Egy nagyon egyszerű feladatot tűztünk ki magunk elé elől, a PIC-ben lévő program kioldjon adatokat (számokat) a PC-n futó terminálból számításra.

```
program USART_hwart;
var i:byte; t1, t2:string(20);
{-----}
begin
  t1:="Számokat kioldtárat"; t2:="ak a következőkben";
  USART_Init(9600);
  USART_Write_Sent(t1);
  USART_Write_Sent(t2);
  while true do begin
    USART_Read(t1);
    Delay_ms(500);
  end;
end;
```

A program elején felöltünk kezdőértéssel két karakterláncot (t1 és t2), majd az USART egység 9600 baud átviteli sebességre való felprogramozás után (USART_Init), ezt a két láncot kioldják a PC szoros portja felé. A továbbiakban hallgatunk a végföld célzatra, ahol félmásodpercenként egy byte-on változó léptékkel növelt értékű továbbiak sorozat.

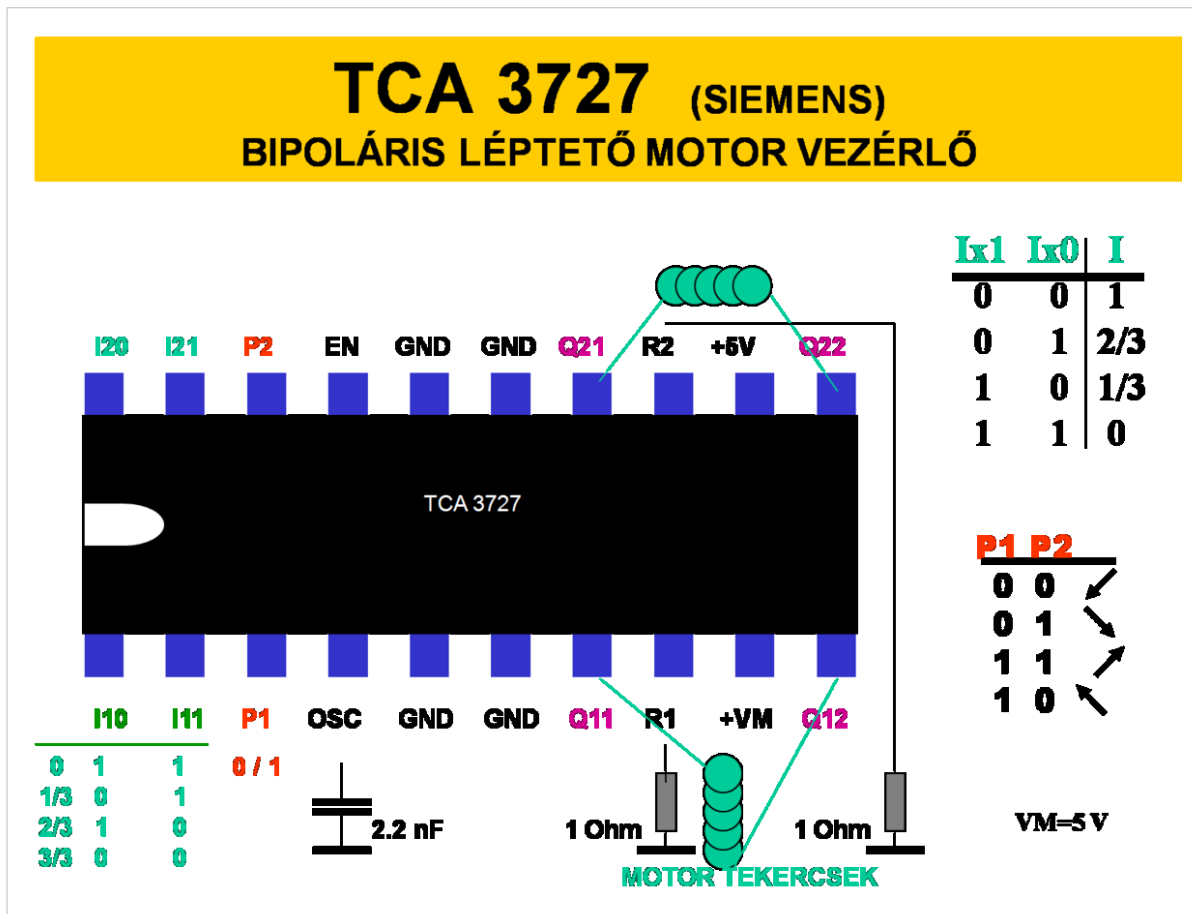
Bővebb leírás található itt: DataSheet^[21] A 97. oldalon.

Léptetőmotorok működése



Unipoláris vezérlőmotor vezérlése PIC-kel

Egyelőre a jegyzetben csak ilyet találtam, ha valaki tud jobbat, ne habozzon megosztani. :)



Szervómotorok működése

Meg kell adni, hogy milyen irányban kell elmozdulnia (pl. egy ellenállás változásával), és pl. egy jelszélesség modulált jelet (hogy milyen gyorsan menjen a motor).

Hivatkozások

- [1] http://en.wikipedia.org/wiki/Open_collector
- [2] http://en.wikipedia.org/wiki/Voltage_stabilizer
- [3] <http://en.wikipedia.org/wiki/Transistor>
- [4] http://en.wikipedia.org/wiki/Operational_amplifier_applications#Comparator
- [5] <http://en.wikipedia.org/wiki/Comparator>
- [6] http://en.wikipedia.org/wiki/555_timer_IC
- [7] <http://www.eleinmec.com/article.asp?1>
- [8] <http://www.eleinmec.com/article.asp?4>
- [9] <http://www.eleinmec.com/article.asp?3>
- [10] <http://www.eleinmec.com/article.asp?5>
- [11] http://en.wikipedia.org/wiki/Resistor_ladder#R-2R_resistor_ladder_network
- [12] <http://www.ecelab.com/interfacing-opamp-ttl-cmos.htm>
- [13] <http://en.wikipedia.org/wiki/Multiplexer>
- [14] <http://en.wikipedia.org/wiki/Multiplexing>
- [15] http://en.wikipedia.org/wiki/Digital_comparator
- [16] [http://en.wikipedia.org/wiki/Adder_\(electronics\)#Full_adder](http://en.wikipedia.org/wiki/Adder_(electronics)#Full_adder)
- [17] http://en.wikipedia.org/wiki/Three-state_logic
- [18] http://en.wikipedia.org/wiki/Harvard_architecture

-
- [19] http://en.wikipedia.org/wiki/Von_Neumann_architecture
 - [20] http://en.wikipedia.org/wiki/Pulse-width_modulation
 - [21] <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>
-

Szócikkek forrása és közreműködői

Mikrokontrollerek tételsor *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?oldid=223> *Közreműködők:* Csega

Képek forrásai, licencei és közreműködői

Kép:Opencollector.jpg *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Opencollector.jpg> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:Op-Amp_Comparator.png *Forrás:* http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Op-Amp_Comparator.png *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:R2R.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:R2R.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:(de)multiplexer.png *Forrás:* [http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:\(de\)multiplexer.png](http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:(de)multiplexer.png) *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:digitkomparator.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Digitkomparator.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:teljesosszeado.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Teljesosszeado.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:logkapuk.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Logkapuk.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:printerport.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Printerport.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:picblockdiag.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Picblockdiag.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:tmr0.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Tmr0.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:tmr1.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Tmr1.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:tmr2.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Tmr2.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:picmin.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Picmin.png> *Licenc:* ismeretlen *Közreműködők:* Csega
fájl:IOportok.pdf *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:IOportok.pdf> *Licenc:* ismeretlen *Közreműködők:* Csega
fájl:ADconv.pdf *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:ADconv.pdf> *Licenc:* ismeretlen *Közreműködők:* Csega
fájl:LCD.pdf *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:LCD.pdf> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:LCD.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:LCD.png> *Licenc:* ismeretlen *Közreműködők:* Csega
fájl:usart.pdf *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Usart.pdf> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:leptetomotor.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Leptetomotor.png> *Licenc:* ismeretlen *Közreműködők:* Csega
Kép:bipolarmotor.png *Forrás:* <http://mafihe.hu/~wiki/wiki/index.php?title=Fájl:Bipolarmotor.png> *Licenc:* ismeretlen *Közreműködők:* Csega