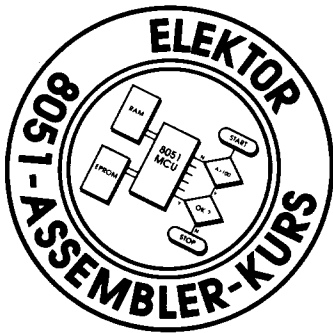


8051-es mikrokontroller- és assembler-tanfolyam

3. rész



A tanfolyam e részében a 8051 utasításkészletével kapcsolatos ismereteinket fogjuk bővíteni és megismerjük az eddig még nem tárgyalt bitcímzést, valamint annak használatát. Az új utasítások már olyan programok megírását teszik lehetővé, melyek a valós felhasználásokat is megközelítik (például servo-vezérlőimpulzusok generálása és analóg jel-output). Mindezek kapcsán sor kerül majd kiegészítő kártya alkalmazására és valós idejű tesztelésére is.

Állapotjelzők és bitcímzés

A program által végrehajtandó tevékenység gyakran egy meghatározott jel állapotától függ. Ez lehet egy külső jel, például egy komparátor kimenete, mely egy Input-Portra csatlakozik, de belső jellel (Bit-ről, Flag-ról) is lehet szó, például az akkumulátorban található (ACC.3-mal jelölt) 3. bit-ről. Ilyen egy bites információk kiértékelése céljából a 8051-es sorozat egy további címzési móddal rendelkezik. Ez a bitcímzés. Ha a biteket nem csak lekérdeznünk akarjuk, hanem logikai kapcsolatba (pl. VAGY kapcsolat) is kívánjuk hozni, akkor a bitmanipulációs utasításokat használhatjuk.

A 8051 sorozatba tartozó mikrokontrollerek a 000H-tól 0FFH-ig terjedő címeken 256 bitet tudnak megcímezni. A közvetlen címzéshez hasonlóan a 128-nál kisebb, illetve a 127-nél nagyobb címek itt is eltérő jelentésűek. A 0-tól 127-ig terjedő címekkel a belső RAM bitjei kerülnek kijelölésre. A 0 bit-cím ennek során a belső RAM-ban található 20H című Byte 0. bitjéhez való hozzáférést jelenti. A 127 bit-cím a belső RAM 2FH című Byte-jának 7-es bitjére mutat. A belső RAM 20H-tól 2FH-ig terjedő 16 Byte-ja tehát a 0-tól 127-ig terjedő bitek tárolására szolgál.

A 128-tól 255-ig terjedő bitcímekkel a speciális funkcióregiszterekben tárolt bitekre hivatkozhatunk. A tényleges bitcím az SFR cím és a lehívni kívánt bitszám összege teszi ki. Az akkumulátorban (SFR OE0H) található, a 3-as bit kijelölésére tehát a OE3H bitcím kell használni. Az assembler az összeadási munkát átveszi, ha a bitcímzés során a „PONT-NOTÁCIÓ”-t használjuk.

Ennek során a megfelelő bitszámot egyszerűen az SFR cím után ponttal elválasztva adjuk meg.

Példa:

```
JB ACC.3, HOVA'; ugorj a HOVA-ra;
                    ha az akkumulátor 3. BIT-je
                    logikai 1-es
```

(Az ACC konstanshoz megelőzően az EQU-val az akkumulátor SFR(címét hozzá kell rendelni).

Nem lehet az összes SFR-t bitcímezni, csupán azokat, melyek címében az utolsó három bit 0.

A PSW program-állapotszó

Az úgynevezett program-állapotszó a 8051 ODOH című SFR-jének helyén található. Benne olyan bitek (állapotjelzők) szerepelnek, amelyek arról nyújtanak felvilágosítást, hogy a legutóbbi összeadásnál vagy kivonásnál keletkezett-e átviteli (angolul: Carry) bit. Ez az információ a 7-es bitben kerül tárolásra és így a PSW.7 bitcímrel hivatkozhatunk rá. Maga a (C-állapotjelzőnek is nevezett) bit számos bitmanipulációs utasítással elérhető. A 8051-es sorozatnál ez a bit bizonyos értelemben egy bitmanipulációra szolgáló egy bites akkumulátor szerepét tölti be. Benne kerül tárolásra a bitek közötti logikai kapcsolatok (VAGY, ES és NEM) eredménye.

A Parity-Flag (PSW.0) akkor kap logikai „1” értéket, ha az akkumulátorban páratlan számú egyes található. Az OV (PSW.2) és AC (PSW.6) állapotjelzők ellenőrzésre szolgálnak az előjeles vagy BCD számokkal végzett számítások során. A PSW 3-as és 4-es bitjéről lehet megállapítani, hogy melyik regiszterbankra fogunk hivatkozni. Tanfolyamunk során normális körülmények között mindig a 0 bankot használjuk. Ezt a két bitet tehát nem kell megváltoztatni. Az 1-es és az 5-ös (általános célú) bitek szabadon használhatók, így tetszőlegesen beírhatók és lekérdezhetők. A bitek jelentését az 1. ábrán szereplő táblázatban összefoglaltuk. Kiegészítésként a 2. ábrán azokat az utasításokat is megadtuk, amelyek a PSW-ben található állapotjelzőket befolyásolják.

A bitek és a program-állapotszóban található állapotjelzők területére tett kirándulásunk után térjünk most át a további utasításokra. Ezek alkalmazása a 8. és a 9. ábrák listáin bemutatott BSP6 és BSP7 programokban kerül példaként bemutatásra. A programok forráskódjai (BSP6.A51 és BSP7.A51) természetesen a tanfolyam diszkettjén ugyancsak megtalálhatók. A beírási munka tehát ismét megtakarítható és rögtön lehet kezdeni.

Bits (Flags) im Programmstatuswort PSW

CY	PSW.7	Carry Flag	: Ueberlauf-Flagge , Bit Akkumulator C
AC	PSW.6	Auxiliary Carry	: Hilfsflagge fuer BCD Addition
FO	PSW.5	FLAGO	: frei fuer allgemeine Anwendungen
RS1	PSW.4	Reg. Bank select 1	: Anwahl der Register-Bank Bit 1
RS0	PSW.3	Reg. Bank select 0	: Anwahl der Register-Bank Bit 0
OV	PSW.2	Overflow	: Ueberlauf-Flagge
	PSW.1		: frei fuer allgemeine Anwendungen
P	PSW.0	Parity	: Paritaet im Akkumulator

Register-Bank-Anwahl

RS1	RS0	Register-Bank	Adressen im internen RAM
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

910109-4-11

1. ábra. Az állapotjelzők jelentése a PSW program-állapotszóban

1. ábra.

- Bits (Flags) im Programmstatuswort PSW = Bitek (állapotjelzők) a PSW program-állapotszóban
- Ueberlauf-Flagge, Bit Akkumulator C = Átviteli állapotjelző, C bitakkumulátor
- Hilfsflagge fuer BCD Addition = Segéd-állapotjelző BCD összeadáshoz
- frei fuer allgemeine Anwendungen = szabad felhasználású, általános célokra
- Anwahl der Register-Bank Bit 1 = Regiszterbank megválasztása, 1. bit
- Anwahl der Register-Bank bit 0 = Regiszterbank megválasztása, 0. bit
- Ueberlauf-Flagge = Túlcsordulás állapotjelző
- frei fuer allgemeine Anwendungen = szabad felhasználású, általános célokra
- Paritaet im Akkumulator = Paritás az akkumulátorban

- Register-Bank-Anwahl = Regiszterbank megválasztása
- Adressen im internen RAM = címek a belső RAM-ben

2. ábra.

- Flagbeeinflussende Befehle = Állapotjelzőt befolyásoló utasítások
- wird beeinflusst = Befolyásolásra kerül
- wird gesetzt = „1”-es értéket kap
- wird zurueckgesetzt = nullázásra kerül
- das invertierte Bit wird verknuepft = Az invertált bit kerül beírásra
- Notiz = Megjegyzés
- Auch das Beschreiben unter SFR Adresse ODOH veraendert die Flags = A ODOH cím alatti SFR beírás is megváltoztatja az állapotjelzőket

2. ábra. Ezek az utasítások befolyásolják a PSW állapotjelzőket

Flagbeeinflussende Befehle

INST	OP	C	OV	AC
ADD		x	x	x
ADDC		x	x	x
SUBB		x	x	x
MUL		0	x	
DIV		0	x	
DA		x		
RRC		x		
RLC		x		
SETB	C		1	
CLR	C		0	
CPL	C	x		
ANL	C,bit	x		
ANL	C,/bit	x		
ORL	C,bit	x		
ORL	C,/bit	x		
MOV	C,bit	x		
CJNE		x		

- x : wird beeinflusst
- 1 : wird gesetzt
- 0 : wird zurueckgesetzt
- /bit : das invertierte Bit wird verknuepft

Notiz:

Auch das Beschreiben unter SFR Adresse ODOH veraendert die Flags.

910109-4-12

A programok működésének magyarázatára a későbbiekben kerül sor.

Feltételes ugrási utasítások

Meghatározott jelektől vagy eseményektől függő különböző akciók indítására a programon belül ugrási utasításokat használunk. Amennyiben egy feltételes ugrási utasítás esetén az adott feltétel teljesült, úgy a program folytatására az utasításban megadott címnél kerül sor. Ha a feltétel nem teljesült, akkor a program egyszerűen a következő utasítással folytatódik. A 8051 a következő ugrási utasításokat ismeri:

JZ adr;	ugorj az adr címre, ha az akkumulátor = 0
JC adr;	ugorj, ha az átvitelflag (PSW.7) = 1
JB bit, adr;	ugorj, ha a bit bitcímű bit = 1
JBC bit, adr;	mint JB, de egyúttal töröld a bitet is

Ezek közül az utasítások közül az első három negált (N-nel jelölt) ugrási feltétellel is léteznek. (Jelölésük: JNZ, JNC, ill. JNB). A megadott ugrási címet, az úgynevezett ugrási célt rövid ugrással kell elérni (-128-tól +127-ig terjedő ugráskülönbség, lásd a tanfolyam 2. részét, SJMP). Feltételtől függő hosszú ugrást tehát csak egy (rövid) feltételes ugrás és egy azt követő (feltétel nélküli) hosszú ugrás egymás utáni alkalmazásával valósíthatunk meg.

Összehasonlítási utasítás

A 8051 egy további ugrási utasítása a következő:

CJNE op1, op2, adr;	ha op1 nem egyenlő op2-vel, akkor ugorj az adr címre
---------------------	--

Az op1 és op2 operandusok az utasítástáblázatból vehetők ki. Ezzel az utasítással például könnyen ellenőrizhető, hogy a regiszter az előre megadott értéket tartalmazza-e, majd ezt követően megfelelő ugrás hajtható végre.

A DJNZ számlálási utasítás

Programozás során sok esetben szükségessé válik az úgynevezett hurok használata: bizonyos utasításokat ciklikusan többször meg kell ismételnünk. Ha az ismétlések száma 1 és 255 közé esik, akkor a ciklus-szervezés a következő utasítással könnyen elérhető:

DJNZ Rn,rel

- 3. ábra.**
- Taste B gedrückt = „B” gomb megnyomva
 - Taste A gedrückt = „A” gomb megnyomva
 - Mittelstellung = középállás
 - Rechts langer Impuls = Jobbra hosszú impulzus
 - Links kurzer Impuls = Balra rövid impulzus
 - Pause ca. 10ms = Szünet kb. 10 ms

- 4. ábra.**
- Impulse = impulzusok
 - Datenbus = adatbusz
 - „kurz” = „rövid”
 - „lang” = „hosszú”
 - Compuboard = Compuboard
 - Ergänzungsplatine = kiegészítő panel

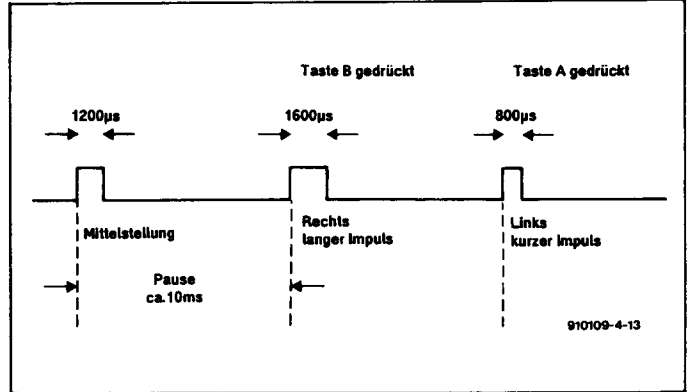
Ez az utasítás az Rn (számláló) regiszter tartalmát először 1-gyel csökkenteni (dekrementálja), majd megvizsgálja, hogy a regiszter tartalma nulla-e. Ha nem, akkor a programnak a megadott ugrási címnél (rel) való folytatására kerül sor, egyébként a program a soronkövetkező utasítással folytatódik. A számlálóregiszter helyett, ahogy az az utasítástáblázatban látható, közvetlen címezhető Byte is használható.

A BSP6 programban ez az utasítás például időkésleltetés megvalósítása céljából kerül felhasználásra.

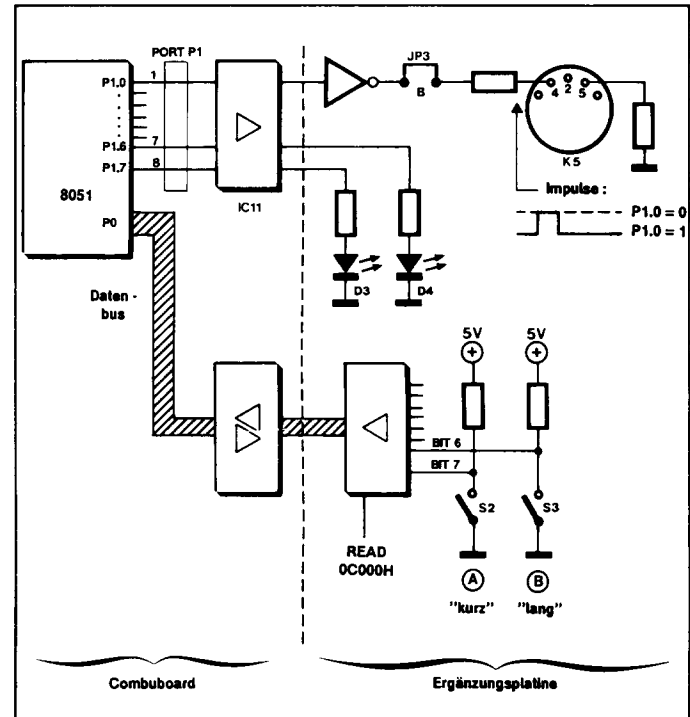
Ehhez a regisztert először egy olyan értékre töltjük fel, amely a késleltetés hosszához szükséges. Ezt követően egy pár „időhúzó utasítás” kerül végrehajtásra és végül a DJNZ utasítással a hurok újra indul. Azt, hogy a hurok a 29-től 33-ig terjedő sorokon hány-szor fut végig, éppen az R1 regiszterben a hurok kezdetén tárolt érték határozza meg. Mivel a hurok belül az utasítások végrehajtásához 10 mikroszekundum szükséges (a lista T oszlopában szereplő idő összege), a WARTE hurok 10 R1 mikroszekundum késleltetést valósít meg (ez az adat nem egészen pontos, mivel csak a hurokidővel számoltunk annak ellenére, hogy korrekt számítás érdekében az ACALL és RET utasítások idejét is figyelembe kell venni ugyanúgy, mint a CLR, illetve SETB utasítások idejének egy részét is; a tényleges impulzus időtartam így 5 mikroszekundummal hosszabb).

Logikai kapcsolatok

Logikai kapcsolatok céljaira a 8051 az ÉS (AND), VAGY (OR) és kizáró VAGY (EXOR) kapcsolatokra szolgáló utasításokat tartalmazza. A kapcsolatot a két operandus valamennyi



3. ábra. Szervevezérlő Impulzusok



4. ábra. Az impulzusgenerátor jelfolyam ábrája

bitjére vonatkozóan végrehajtásra kerül. Az utasítások a következők:

- ANL cél, forrás; a cél helyére (cél AND forrás) kerül
- ORL cél, forrás; a cél helyére (cél OR forrás) kerül
- XRL cél, forrás; a cél helyére (cél EXOR forrás) kerül

Célként és forrásként megint minden esetben az utasítástáblázatban megadott kombinációk használata engedélyezett. Induljunk ki abból, hogy az akkumulátor az 10010111B = 97H = 151 értéket tartalmazza, a RO regiszterben pedig az 11110010B = F2H = 242 érték szerepel. Ebben az esetben az utasítások a következő eredményekhez vezetnek:

Az ANL utasítás például egy Byte-on belül egy meghatározott bit törlésére és a többi bit változatlanul hagyására használható, a kívánt bit pl. egy ORL utasítás

segítségével történő utólagos beírása céljából. Az ilyen maszkolási műveletek az assembler programozás kedvelt segédeszközét képezik. Annak tesztelése céljából például, hogy az akkumulátorban a 0., 1., 2. vagy 7. bit logikai „1” értékű-e, a következő program készíthető:

```
ANL A, #10000111;
      7, 2, 1, 0 biteket
JNZ VAN
EGYES; A nem nulla,
      ha valamelyik
      bit = 1
```

Az akkumulátoron végrehajtott

```
CLR A; töröld az
      akkumulátort
CPL A; invertáld az akku-
      mulátor bitjeit
```

utasításokkal az akkumulátor törölhető (nullázható), illetve annak valamennyi bitje invertálható.

Utasítás	ANL A, RO	ORL A, RO	XRL A, RO
A előzőleg:	10010111	10010111	10010111
RO előzőleg:	11110010	11110010	11110010
A utána:	10010010	11110111	01100101

Bitmanipulációs utasítások

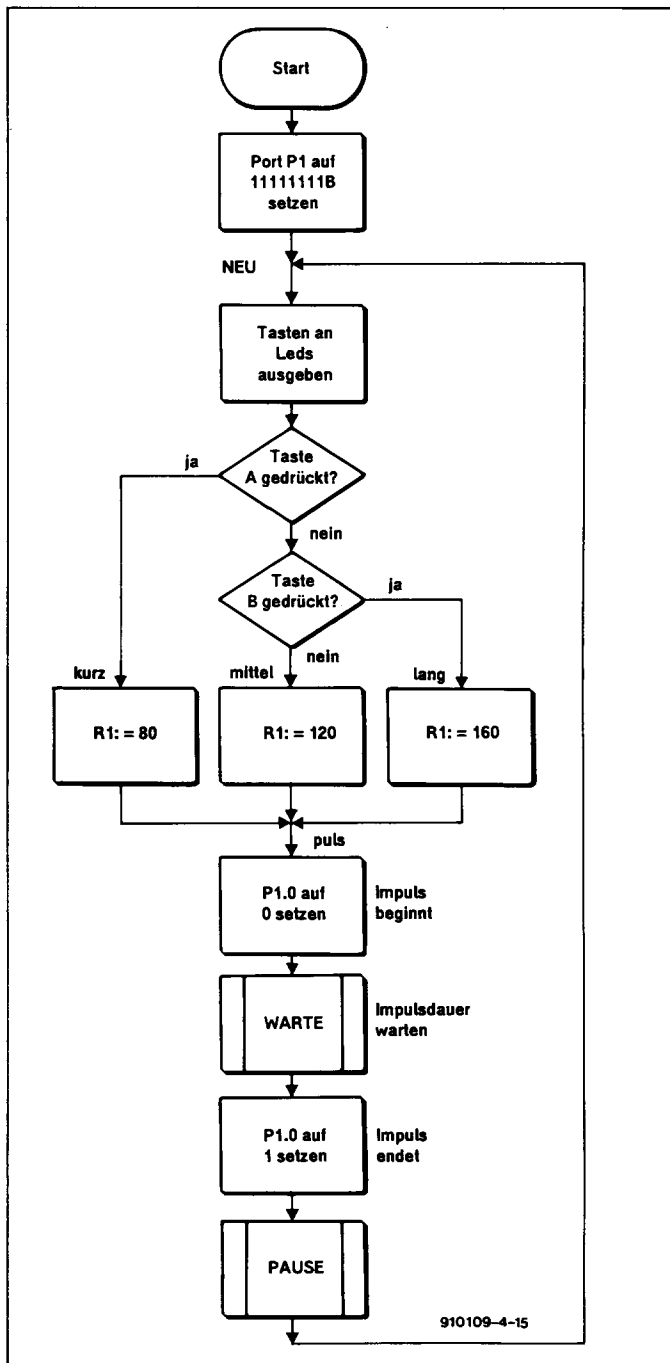
A bitmanipulációs utasítások az alábbiak:

CLR bit-operandus;	töröld a bitet (írj be 0-t)
SETB bit-operandus;	írd be a bit-be 1-et
CPL bit-operandus;	invertáld a bitet
ANL C, bit-operandus;	C helyett (C AND bit) irándó
ORL C, bit-operandus;	C helyett (C OR bit) irándó
MOV bit-operandus, C;	bit helyett C irándó
MOV C, bit-operandus;	C helyett bit irándó

Ezek az utasítások ugyanazokat a manipulációkat teszik lehetővé az egyes bitekkel, mint amelyek a logikai utasítások a Byte-okkal tesznek lehetővé. A logikai kapcsolatok esetén a C bitet (PSW-ben a 7-es bit) használjuk eredménytárolóként. A bitutasításokat gyakran használják egyes bitek (pl. Output-Port-ok egyes bitjei) beírására vagy

5. ábra.

- Port P1 auf 11111111B setzen = P1 Portot 11111111B-vel betölteni
- NEU = ÚJ
- ja = igen
- Tasten an LEDs ausgeben = Nyomógombokat a LED-ekre kiadni
- Taste A gedrückt? Meg van nyomva az „A” gomb?
- nein = nem
- Taste B gedrückt? Meg van nyomva a „B” gomb?
- kurz = rövid
- mittel = közepes
- lang = hosszú
- puls = impulzus
- P1.0 auf 0 setzen = P1.0-t 0-ra állítani
- Impuls beginnt = impulzus kezdődik
- WARTE = VÁRJ
- Impulsdauer warten = impulzustartamot kívárni
- P1.0 auf 1 setzen = P1.0-t 1-re állítani
- Impuls endet = impulzus befejeződik
- PAUSE = SZÜNET



5. ábra. Az impulzusgenerálás folyamatábrája

KAPCSOLÓK, TÁVADÓK, ÉRZÉKELŐK

A svájci CONTRINEX KÖZELÍTÉSKAPCSOLÓI, valamint a legkorszerűbb elven működő NYOMÁSÉRZÉKELŐK

0,6...400 bar

FOLYADÉKSZINTMÉRŐK

100 m mélységig

HŐMÉRSÉKLET-ÉRZÉKELŐK

-30...500 °C

TÁVADÓK

0-10 V, 4-20 mA

FELDOLGOZÓ ELEKTRONIKÁK

az INTERBIP INVEST

Mikroelektronikai RT-től,

mely a Contrinex termékeinek

kizárólagos

magyarországi forgalmazója.

1047 Budapest, Fóti út 56.

Tel/Fax: 160-3420.

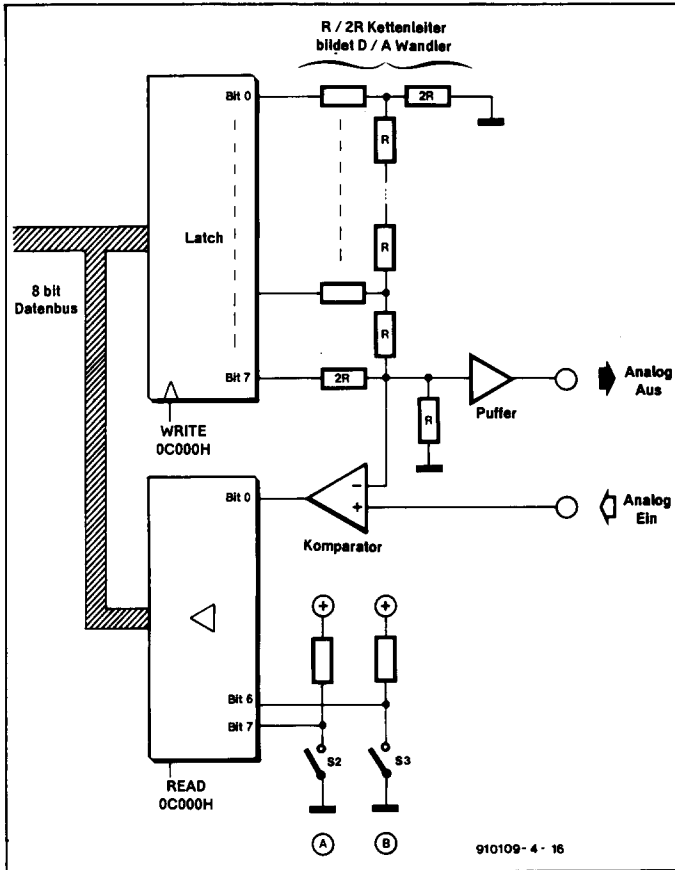
Egyedi igények kielégítése, alkalmazási tanácsadás.

törlesztésre anélkül, hogy ennek során a többi bitet megváltoztatnák.

A szervo tesztelése

A BSP6 program feladata impulzusok generálása egy távvezérlő szervo (kormányzógép) teszteléséhez (3. ábra). Ennek során 800 mikroszekundumos rövid impulzusokat kell létrehozni („kormányzás balra”) az A nyomógomb, illetve kapcsoló működtetésével. Hosszú, (jobbra); 1600 mikroszekundum időtartamú impulzusokat kell generálni a B gomb megnyomásakor. Amikor egyik gomb sincs megnyomva, akkor 1200 mikroszekundum hosszúságú impulzusokat (középállás) kell előállítani. Az impulzusok között kb. 10 milliszekundumos impulzusszünetet kell tartani.

A már régebben ismertetett hardverkiegészítés a feladat megoldása során most kerül alkalmazásra. Az A és a B nyomógombok szerepét a multifunkciós kártyán található S2 és S3 nyomógombok veszik át. Ezeket így a 0C00H cím alatt (hozzáférés a külső RAM-hoz, Memory mapped I/O) lehet lekérdezni. A jeleket a MIDI kimeneten (5-ös hüvely 4-es pontja a kiegészítőpanelon) kívánjuk rendelkezésre bocsátani. Ahhoz, hogy a jel a csatlakozóhüvelyre jusson, a kiegészítőpanelon a JP3 Jumpert B helyzetbe kell állítani. A kártyán található inverternek tulajdonítható, hogy a P1.0 Porton megjelenő 0 output alacsony szintet eredményez a K5 csatlakozón. A kiadó jelfolyamat vázlatosan a 4. ábrán, a program logikai lefutását az 5. ábrán mutatjuk be. A megvalósítás a 8. ábrán szereplő listából érthető meg.



6. ábra. Így csatlakozik a D/A átalakító a Compuboard-ra

A P1 Portról történő inicializálás után (Uj címkenél) végtelen hurok kezdődik. Először megjelenik a nyomógombok állapota a P1 Port LED diódáin. Ennek során a 14. sorban rögtön egy ORL utasítás kerül alkalmazásra. Ez gondoskodik arról, hogy P1.0 impulzus-kimeneti bitünk még „1” szinten maradjon. A 16. és 17. sorban aztán az előzőekben megbeszélt feltételes ugrási utasítások kerülnek alkalmazásra, mégpedig mindjárt az akkumulátor bitcímzésével kombinálva. Ezekkel állapítjuk meg a nyomógombok állapotát. Ha az A gomb megnyomott állapotban van, az az A akkumulátorban a 7-es bitnél nullát eredményez, mely a 16. sorban található utasítással kerül lekezdésre.

Attól függően, hogy melyik gomb van megnyomva, az R1 regiszterbe különböző érték kerül beírásra, mely azt adja meg, hogy az egyes impulzusok időtartama milyen hosszú legyen. A 23. sorban ezután megtörténik a P1.0 output-bit nullázása és ezzel a MIDI kimeneten magas (pozitív) szint jelenik meg; az impulzus megkezdődik.

A VARJ alprogram és vele együtt az impulzus időtartamát meghatározó, előírt késleltetés indítására a 24. sorban kerül sor. Az előírt idő eltelte után a 25. sor utasításával az impulzus befejeződik. A kimeneti bitek beírása és törlése az éppen most ismertett bitutasítások segítségével történik.

6. ábra.

- R/2R Kettelenleiter bildet D/A Wandler = R/2R négypólus-lánc képezi az A/D átalakítót
- 8 bit Datenbus = 8-bités adatbusz
- Analog Aus = analóg kimenet
- Analog Ein = analóg bemenet

7. ábra.

- DA-WERT auf 128 setzen = DA-ERTEK-et 128-ra állítani
- NEU = UJ
- DA-WERT an DA-Wandler ausgeben = DA-ERTEK-et D/A átalakítóra kiadni
- 5ms warten = 5 ms-ig várni
- Komparator Ausgänge an LEDS ausgeben = komparátor kimeneteket a LED-ekre kiadni
- Taste A gedrückt? = Meg van nyomva az „A” gomb?
- ja = igen
- kann nicht erniedrigt werden! = nem csökkenthető!
- DA-Wert = 0?
- DA-ERTEK = 0?
- nein = nem
- erniedrige DA-WERT = csökkentsd a DA-ERTEK-et
- Taste B gedrückt? = Meg van nyomva a „B” gomb?
- erhöhe DA-WERT = növeld a DA-ERTEK-et
- DA-WERT = 0?
- DA-ERTEK = 0?
- zu viel erhöht! túl magasra növelve!
- DA-WERT: = 255 = DA-ERTEK: = 255

Végül a SZUNET alprogram behívása útján megtörténik a kb. 10 milliszekundumos impulzus-szünet kivárása. Ezután az Uj címkehez való ugrással az egész folyamat újra előlről kezdődik.

Az ebben a programban található mindkét időhurok (adott hosszúságú impulzus és 10 ms impulzusszünet) esetében DJNZ utasítások kerülnek alkalmazásra.

Aki kormányzógéppel nem rendelkezik, az ezt a programot természetesen oszcilloszkóppal is tesztelheti.

Kiseb változtatás után a program hanggenerálásra is használható. Ebből a célból a 36. sor utasítását a

```
MOV R2,#1
```

utasítással cseréljük fel és a 23., valamint 25. sorokban a P1.0 bit helyett P1.1 bitet írunk. Ezzel azt érjük el, hogy a (P1.1 pontra kötött) hangszóróban olyan hang jelenik meg, amelynek

magassága a két gomb megnyomásától függ.

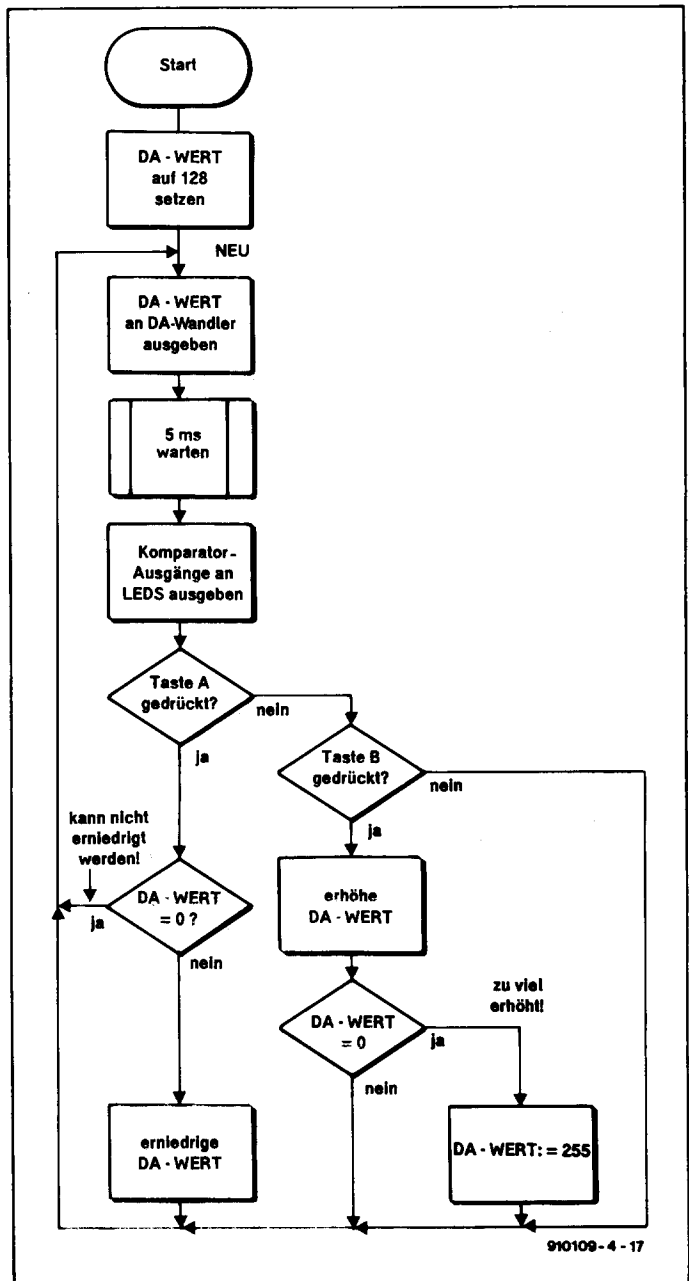
Nyilvánvaló, hogy ugyanezekkel a programozási megoldásokkal tetszőleges impulzusok és impulzus-sorozatok generálhatók. A programozható impulzusgenerátor felé tehát már megtettük az első lépést.

Számlálás

Az egy Byte által kifejezhető, 0-tól 255-ig terjedő számtartományon belüli számlálásra a következő utasítások állnak rendelkezésre:

INC Byte-operandus;	növeld a Byte-ot 1-gyel
DEC Byte-operandus	csökkentsd a Byte-ot 1-gyel
INC DPTR;	növeld a 16-bités DPTR-t 1-gyel

7. ábra. A D/A-output folyamatábrája



```

***** LISTING of EASM51 (BSP6) *****
LINE LOC OBJ T SOURCE
1 0000 ; ***** DATEI BSP6.A51 *****
2 0000 ;
3 0000 ACC EQU 090H ; SFR Adresse des Akkumulators
4 0000 P1 EQU 090H ; SFR PORT1 Adresse ist 090H
5 0000 TASTEM EQU 0C000H ; Tasten Adresse im externen Datensp.
6 0000 kurz EQU 80 ; kurzer Impuls 800 Microsec.
7 0000 mittel EQU 120 ; mittellanger Impuls 1000 Microsec.
8 0000 lang EQU 160 ; langer Impuls 1600 Microsec.
9 0000 ;
10 0000 ;
11 4100 75 90 FF [2] START ORG 4100H ; 1. Programm steht spaeter ab 4100H
12 4103 90 C0 00 [2] MOV P1,#11111111B ; Alle Port Bits 1
13 4106 E0 [2] NEU MOV DPTR,#TASTEM ; Tasten Adresse
14 4107 44 0F [1] MOVX A,#DPTR ; Tasten einlesen (Bits 4,5,6,7)
15 4109 F5 90 [1] ORL A,#00001111B ; Bits 0,1,2,3 auf 1 setzen
16 410B 30 E7 07 [2] MOV P1,A ; LEDS entsprechen Tasten, P1.0=1
17 410E 30 E8 08 [2] JNB ACC.7,TASTEa ; Taste A = Kurzer Impuls
18 4111 79 78 [1] JNB ACC.6,TASTEb ; Taste B = Langer Impuls
19 4113 80 06 [2] MOV R1,#mittel ; keine Taste, Impuls mittellang
20 4115 79 50 [1] SJMP PULS ; Puls ausgegeben
21 4117 80 02 [2] TASTEa MOV R1,#kurz ; andere Laenge
22 4119 79 A0 [1] SJMP PULS ;
23 411B C2 90 [1] TASTEb MOV R1,#lang ; andere Laenge
24 411D 31 25 [2] CLR P1.0 ; P1.0 wird LOW, MIDI wird High
25 411F D2 90 [1] PULS ACALL WARTE ; Puls-laenge (R1) abwarten
26 4121 31 30 [2] SETB P1.0 ; Impuls beenden
27 4123 80 DE [2] ACALL PAUSE ; 10 Millisekunden Pause
28 4125 ; SJMP NEU ; von vorne los
29 4125 AF E0 [2] WARTE MOV R7,ACC ; Unterprogramm R1*10 Microsecunden
30 4127 AF E0 [2] MOV R7,ACC ; jeweils 2 microsecunden
31 4129 AF E0 [2] MOV R7,ACC ; Zeit verschuenden
32 412B AF E0 [2] MOV R7,ACC ;
33 412D D9 F6 [2] DJNZ R1,WARTE ; R1 mal 10 Zyklen a 1 Mikrosec.
34 412F 22 [2] RET ; Ende des Unterprogramms
35 4130 ;
36 4130 7A 0A [1] PAUSE MOV R2,#10 ; 10 Millisekunden
37 4132 79 64 [1] PAUSE1 MOV R1,#100 ; sind 10*100*10 Mikrosecunden
38 4134 31 25 [2] ACALL WARTE ; hier 1000 microsecunden warten
39 4136 DA FA [2] DJNZ R2,PAUSE1 ; aeußere Schleife fortsetzen
40 4138 22 [2] RET ; Ende des Unterprogramms
41 4139 ;
42 4139 ; END
***** SYMBOLTABLE (14 symbols) *****
ACC :00E0 P1 :0090 TASTEM :C000 kurz :0050
mittel :0078 lang :00A0 START :4100 NEU :4103
TASTEa :4115 TASTEb :4119 PULS :411B WARTE :4125
PAUSE :4130 PAUSE1 :4132

```

8. ábra. A BSP6 program a kormányzógép (szervó) vezérléséhez olyan impulzusokat generál, melyek különböző hosszúságúak attól függően, hogy melyik gombot nyomjuk meg

Mint minden esetben, a megengedett Byte-operandusokat most is tanfolyamunk 2. részének utasítástáblázatából kell kiválasztani. Az első két utasítás nem befolyásol állapotjelzőket, még az átviteli állapotjelzőt sem. Ha a megnövelendő Byte értéke 255, akkor az eredmény 0 lesz, a 0 érték eggyel való csökkentése a 255 értéket eredményezi. E két utasítás használatára egy kis példa a BSP7 programban található. Rendelkezésre áll továbbá a 16-bites Datapointer eggyel való megnövelésére vonatkozó harmadik utasítás is. Ez gyakran kerül alkalmazásra olyan esetekben, amikor a programmemóriában vagy az adatmemóriában található táblázatokat lassan, folyamatosan, lépésről-lépésre kívánjuk megcímezni.

A D/A átalakító tesztelése

A kiegészítő panelon elhelyezkedő egyszerű D/A átalakító a 6. ábrán látható vázlatos kapcsolásnak megfelelő felépítésű. Kimeneti jele a (⇒) jelű analóg kimeneten leválasztva áll rendelkezésre. A jel egyidejűleg a komparátorokra is eljut. (Az ábrán csak egy komparátor szerepel). Így a D/A átalakító kimeneti feszültsége az 1-től 3-ig terjedő bemenetekre adott feszültségekkel hasonlítható össze. A BSP7 programnak most a kö-

vetkező feladatot kell elvégeznie:

Olyan feszültséget kell szolgáltatni, melynek értéke az A, illetve B nyomógombok (6. ábra) működésének hatására növelhető, illetve csökkenthető. Ennek során egyidejűleg a három komparátor kimeneti jelét a LED-eken meg kell jeleníteni.

Így tehát lehetőségünk van feszültség programvezérelt módon történő előállítására és annak megállapítására, hogy ez a három bemenetre adott feszültségeknél nagyobb- vagy kisebb-e.

A program folyamatábrája a 7. ábrán látható. Ebben a programban ismét újabb utasítások fordulnak elő. Első alkalommal kerül sor a belső RAM egyik memóriarekeszének használatára is. Ezért megegyeszer nézzük át röviden a 9. ábra listájának egyes utasításait: Az aktuális feszültségérték tárolása a belső RAM-ban 050H cím alatt történik (DA ERTEK szimbolikus név alatt, ld. 9-es sor). A program kezdetén (12. sor) erre a helyre 128 kerül beírásra. Az Új címkével kezdődő programhurokban (14-től 16-ig terjedő sorok) először ennek a tárolócellának az értéke kerül ki a D/A átalakítóra. Ezután 5 millisekondum várakozás következik. Végül a 19-től 22-ig terjedő sorokban megtörténik a komparátorok kimeneti jelének a LED-ekre való továbbítása. A 23-tól 25-ig terjedő sorokban az kerül megállapításra, hogy valamelyik gomb meg van-e nyomva.

8. ábra.

– DATEI BSP6.A51 = BSP6.A51 FILE
– SFR Adresse des Akkumulators = az akkumulátor SFR címe
– SFR PORT1 Adresse ist 090H = PORT1 SFR címe 090H
– TASTEM = GOMBOK
– Tasten Adresse im externen Datensp. = nyomógombok címe a kulso adatmemoriaban
– kurz = rovid
– kurzer Impuls 800 Microsec. = rovid impulzus (800 mikrosec.)
– mittel = kozep
– mittellanger Impuls 800 Microsec. = kozeposszu impulzus (800 mikrosec.)
– lang = hosszu
– langer Impuls 1600 Microsec. = hosszu impulzus (1600 mikrosec.)
– 1. Programm steht spaeter ab 4100H = 1. program kesobb, 4100H-tol kezdodik
– alle Port Bits 1 = valamennyi Port bit 1
– NEU = UJ
– Tasten Adresse = nyomógombok címei
– Tasten einlesen (Bits 4,5,6,7) = nyomógombokat beolvasni (4,5,6,7 bit)
– Bits 0,1,2,3 auf 1 setzen = 0,1,2,3 bitek helyere 1-et beírni
– LEDS entsprechen Tasten, P1.0 = 1 a LED-ek a nyomógomboknak felelnek meg, P1.0 = 1

– Taste A = Kurzer Impuls = „A” nyomógomb = rövid impulzus
– Taste B = Langer Impuls = „B” nyomógomb = hosszú impulzus
– mittel = közep
– keine Taste, Impuls mittellang = egyik gomb sem, közep-hosszu impulzus
– PULS = PULZUS
– Puls ausgegeben = impulzus kiadva
– TASTEa = aGOMB
– kurz = rovid
– andere Laenge = mas hossz-szusag
– TASTEb = bGOMB
– P1.0 wird LOW, MIDI wird High = P1.0 ALACSONY, MIDI MAGAS lesz
– WARTE = VARJ
– Puls-laenge (R1) abwarten = impulzushosszot (R1) kivarni
– Impuls beenden = impulzust befejezni
– PAUSE = SZUNET
– 10 Millisekunden Pause = 10 millisekondum szunet
– von vorne los = kezd elolrol
– Unterprogramm R1*10 Microsecunden = R1*10 mikrosecundumos alprogram
– jeweils 2 microsecunden = mindig 2 mikrosecundum
– Zeit verschuenden = ido telik el
– R1 mal 10 Zyklen a 1 Mikrosec. = R1-szer 10, egyenkent 1 mikrosec. ciklus
– Ende des Unterprogramms = alprogram vege
– 10 Millisekunden = 10 millisekondum
– PAUSE 1 = SZUN 1
– sind 10x100x10 Mikrosecunden = osszesen 10x100x10 mikrosecundum
– hier 1000 Mikrosecunden warten = itt 1000 mikrosecundumot varni
– PAUSE 1 = SZUN 1
– aeußere Schleife fortsetzen = kulso hurkot folytatni
– Ende des Unterprogramms = alprogram vege
– mittel = kozep
– TASTEa = aGOMB
– PAUSE = SZUNET
– lang = hosszu
– TASTEb = bGOMB
– PAUSE 1 = SZUN 1
– TASTEN = GOMBOK
– PULS = PULZUS
– kurz = rovid
– NEU = UJ
– WARTE = VARJ

```

***** LISTING of EASM51 (BSP7) *****
LINE LOC OBJ T SOURCE
1 0000 ; ***** DATEI BSP7.A51 *****
2 0000 ;
3 0000 ACC EQU 090H ; SFR Adresse des Akkumulators
4 0000 P1 EQU 090H ; SFR PORT1 Adresse ist 090H
5 0000 TASTEN EQU 0C000H ; Tasten Adresse im externen Datensp.
6 0000 DA_ADR EQU 0C000H ; DA-Wandler Adresse
7 0000 ;
8 0000 ; RAM
9 0000 DA_WERT EQU 060H ; Ram speicher Vereinbarungen
10 0000 ;
11 0000 ; reserviere 1 Byte bei Adresse 50H
12 4100 75 50 80 [2] START ORG 4100H ; also ueber dem RAM von EMON51
13 4103 NEU MOV DA_WERT,#128 ; Programm steht spaeter ab 4100H
14 4103 90 C0 00 [2] EQU $
15 4106 E5 50 [1] MOV DPTR,#DA_ADR ; Adresse des DA-Wandlers
16 4108 F0 [2] MOV A,DA_WERT ; hole momentanen Wert in AKKU
17 4109 90 00 05 [2] MOVX @DPTR,A ; gib an DA-Wandler aus
18 410C 31 32 [2] MOV DPTR,#5 ; 5 millisekunden warten
19 410E 90 C0 00 [2] ACALL ZEIT
20 4111 E0 [2] MOV DPTR,#TASTEN
21 4112 C4 [1] MOVX A,@DPTR ; hole Komparatoren in Bits 0,1,2
22 4113 F5 90 [1] SWAP A ; Austausch der Bits 0-3 mit Bits 4-7
23 4115 E0 [2] MOV P1,A ; an LEDS ausgeben
24 4116 30 E7 05 [2] MOVX A,@DPTR ; Tasten holen
25 4119 30 E6 0B [2] JNB ACC.7,TASTEa ; Taste A gedrueckt ?
26 411C 80 E5 [2] JNB ACC.6,TASTEb ; Taste B gedrueckt ?
27 411E E5 50 [1] SJMP NEU ; keine Taste, also von vorne los
28 4120 60 E1 [2] TASTEa MOV A,DA_WERT ; Wert soll erniedrigt werden
29 4122 14 [1] JZ NEU ; ist vielleicht null, dann nichts tun
30 4123 F5 50 [1] DEC A ; erniedrige
31 4125 80 DC [2] MOV DA_WERT,A ; und speichere neuen wert im internen
32 4127 E5 50 [1] SJMP NEU ; von vorne
33 4129 04 [1] TASTEb MOV A,DA_WERT ; hole Wert aus internem RAM
34 412A 70 02 [2] INC A ; erhoehe
35 412C 74 FF [1] JNZ IST_OK ; wenns nicht 0 ist ists ok
36 412E F5 50 [1] MOV A,#255 ; sonst wars und ists 255
37 4130 80 D1 [2] IST_OK MOV DA_WERT,A ; speichere neuen verz
38 4132 SJMP NEU ; und wieder los
39 4132 ; MONITOR INTERFACE ; das uebliche
40 4132 ccLTIME EQU 021H ; MONITOR Kommando , DPTR Millisek. Zeit
41 4132 COMAND EQU 030H ; MONITOR Kommando Speicherstelle
42 4132 MON EQU 0200H ; MONITOR Einsprungadresse
43 4132 ;
44 4132 75 30 21 [2] ZEIT MOV COMAND,#ccLTIME
45 4138 LJMP MON
46 4138 END
***** SYMBOLTABLE (14 symbols) *****
ACC :00E0 P1 :0090 TASTEN :C000 DA_ADR :C000
DA_WERT :0050 START :4100 NEU :4103 TASTEa :411E
TASTEb :4127 IST_OK :412E ccLTIME :0021 COMAND :0030
MON :0200 ZEIT :4132

```

Amennyiben igen, úgy a program megfelelő részére történő ugrás következik be, mely a DA ERTEK tárolócellában található szám növelését (32-től 36-ig terjedő sorok) vagy csökkentését (27-től 31-ig terjedő sorok) váltja ki. Ennek során minden esetben gondoskodás történik arról, hogy a maximális érték feletti és a minimális érték alatti érték ne léphessen fel.

Új feladat

Új feladatként megkísérelhető a BSP6 és BSP7 programok oly módon való kombinálása, hogy a kormányzógép helyzetét (a BSP6 program által generált impulzusok hosszát) úgy lehessen a nyomógombokkal befolyásolni, mint a BSP7 programban a kimeneti feszültséget. Aki akarja, természetesen megpróbálhatja a D/A átalakítón át fűrészel vagy bármilyen más, számára érdekes jelalak outputjának megvalósítását.

Most már itt az ideje annak, hogy az assembler irányelvekkel is foglalkozzunk. Ehhez ajánlatos a tanfolyam diszketijén található assembler használati útmutatót még egyszer elolvasni és olyan kis programokat írni, melyeknél például a startcím megválasztása történik másképpen, mint a mi példánkban.

Előrejelzés

Tanfolyamunk következő részében a még hátralevő aritmetikai utasításokat fogjuk megbeszélni. Egyidejűleg bemutatjuk, hogyan végezhető el a 8051-en azok az egyszerű számítási műveletek, melyekre mérési értékek feldolgozásánál gyakran van szükség. Emellett néhány programtechnikát fogunk megbeszélni és programozni fogjuk a V24- vagy MIDI-adatok soros outputját. ■

9. ábra. A BSP7 program a D/A átalakító útján változtatható kimeneti feszültséget bocsát rendelkezésre

9. ábra.

- DATEI BSP7.A51 = BSP7.A51 FILE
- SFR Adresse des Akkumulators = az akkumulátor SFR címe
- SFR PORT1 Adresse ist 090h = PORT1 SFR címe 090H
- TASTEN = GOMBOK
- Tasten Adresse im externen Datensp. = nyomógombok címe a külső adatmemóriában
- DA ADR = DA CIM
- DA-Wandler Adresse = DA átalakító címe
- RAM Speicher Vereinbarungen = RAM taroló direktívák
- DA-WERT = DA ERTEK
- reserviere 1 Byte bei Adresse 50H = foglalj le 1 Byte-ot az 50H címmel
- also ueber dem RAM von EMON51 = tehát az EMON51 RAM területen
- 1. Programm steht spaeter ab 4100H = 1. program később, 4100H-tól kezdődik
- NEU = UJ
- Adresse des DA-Wandlers = DA átalakító címe
- hole momentanen Wert in AKKU = hozd a pillanatnyi értéket az AKKU-ba
- gib an DA-Wandler aus = output a DA átalakítón
- 5 Millisekunden warten = 5 millisekundumot vari
- Zeit = ido
- TASTEN = nyomógombok

- hole Komparatoren in Bits 0,1,2 = hozz komparatorokat a 0,1,2 bitekre
- Austausch der Bits 0-3 mit Bits 4-7 = a 0-3 bitek felcserélése a 4-7 bitekkel
- an LEDS ausgeben = a LED-ekre kiadni
- Tasten holen = nyomógombokat behozni
- Taste A gedrueckt? = meg van nyomva az „A” gomb?
- keine Taste, also von vorne los = egyik gomb sem, tehát kezd elölrol
- TASTEa = aGOMB
- Wert soll erniedrigt werden = az értéket csökkenteni kell
- ist vielleicht Null, dann nichts tun = ha nulla, akkor nincs mit tenni
- erniedrige = csokkents
- und speichere neuen Wert im internen RAM = es az új értéket tarold a belso RAM-ban
- von vorne = elolrol
- TASTEb = bGOMB
- hole Wert aus internem RAM = hozz értéket a belso RAM-ból
- erhoehe = novelj
- IST-OK = RENDBEN
- wenn nicht 0 ist, ists ok = ha nem 0, akkor rendben
- sonst wars und ists 255 = kulonben 255 volt es maradt
- speichere neuen Wert = tarolj új értéket
- und wieder los = es ujra mehet
- das uebliche = a szokasos

- MONITOR Kommando, DPTR Millisek. Zeit = MONITOR parancs, DPTR millisek. ido
- MONITOR Kommando Speicherstelle = MONITOR parancs tarrekesz
- MONITOR Einsprungadresse = MONITOR beugrocim
- ZEIT = IDÓ
- DA WERT = DA ERTEK
- TASTEb = bGOMB
- IST OK = RENDBEN
- ZEIT = IDO
- TASTEN = GOMBOK
- NEU = UJ
- DA ADR = DA CIM
- TASTEa = aGOMB

INCOMP

ELECTRONICS EXPORT-IMPORT

Halmágyi József

ELEKTRONIKAI BERENDEZÉSEK SZERVIZE, ÁRUHÁZA

TV • VIDEO • HIFI • SZÁMÍTÓGÉP • SATELLIT

ALKATRÉSZ-ÁRUSÍTÁS, COMPUTER- ÉS VIDEOSZERVIZ

2120 Dunakeszi, Fő út 35.

Tel./Fax: (27) 42-407