

CE

0

CONRAD

Nyomógombok által programozható vezérlés

Mikrokontrollert mindenütt találunk: a háztartási készülékekben, a szórakoztató elektronikai készülékekben, a gépkocsikban, a mérőkészülékekben, sőt az ember nélküli űrhajókban is. Mindenütt elvégzi azokat a teendőket, amelyeket egy program ajánl. Izgalmas dolog, hogy saját maga hozzon létre egyszer egyszerű vezérlőprogramokat.

Mindig az az első lépés, hogy megkeresse azt a mikrokontrollert vagy processzort, amely a lehető legjobban illik az adott feladathoz. Különböző cégek számos típusa között válogathat. És még a programozási nyelv is kiválasztható. Többnyire az Assemblert és a C-t ajánlják, sok esetben a Basic-ot vagy egy másik magas szintű nyelvet. Normál esetben a programozáshoz költséges szoftverre és egy programozókészülékre van szükség. A pénzráfordításon kívül nem elhanyagolható a betanulási idő sem.

Az itt alkalmazott mikrokontroller teljesen más. A programozáshoz két nyomógombos kapcsolón kívül nincs másra szüksége. A *nyomógomb-programozású vezérlés* (TPS = Tastenprogrammierbare Steuerung) csak viszonylag kevés parancsot ismer, amelyek könnyen megtanulhatók, és a kontrollerben lévő nyomógombok segítségével programozhatók. A programot bármikor megváltoztathatja különösebb segédeszköz használata nélkül.

A rendszer elsősorban a mérés, vezérlés és szabályzás kompakt feladataira alkalmas. Sok feladat oldható meg teljes értékűen ezzel a rendszerrel. Ehhez járul még az, hogy a mikrokontrollert a sikeres programozás után be lehet építeni saját kapcsolásaiba. Ehhez feltételezzük az elektronikai alapismeretek meglétét.

Ugyanakkor alkalmas a rendszer oktatási alapnak és a mikrokontroller-programozás első lépéseként is. Hamarabb ér a sikereket, mint más rendszerekkel. A szerkezete hasonló más programozási nyelvekéhez, úgyhogy könnyebbé válik az átmenet a későbbi átálláshoz.

Tartalomjegyzék

1	Bevezetés	5
2	Váltakozó villogó	8
3	Bináris számláló és PWM-kiadás	9
4	Analóg-digitális-átalakító	13
5	Véletlengenerátor	15
6	Impulzushosszúság mérése	17
7	Programok kiolvasása	19
8	Programok beadása	21
9	A példaprogramok helyreállítása	23
10	TPS-alapparancsok	23
11	Számítás változókkal	26
12	Ugrások és elágazások	28
13	A parancsok áttekintése	30
14	Számlálólóhurok	31
15	Összehasonlítások	32
16	AND, OR és XOR	33
17	Alprogramok	34
18	Szürkületi kapcsoló	36
19	LED-fényszabályzás	36
20	Számzár	38
21	Függelék	40

1 Bevezetés

A TPS-kontroller elve nagyon egyszerű. Van négy digitális bemenet (E1 - E4), és négy digitális kimenet (A1 - A4). Van ezenkívül két analóg bemenet (AD1 és AD2), továbbá egy kvázi-analóg PWM-kimenet. Egy reset-bemenet a hozzátartozó reset-gombbal visszaállítja a programokat a kezdeti állapotukra. A kontrollert három AA-méretű elem táplálja kb. 4,5 V feszültséggel, és a 2,2 V - 5,5 V tartományban működőképes.

Műszaki adatok:

Mikrokontroller: HT46F47

Órafrekvencia: 2 MHz

Belső EEPROM: 128 byte

Tápáramellátás VCC: 2,2 V - 5,5 V

Áramfelvétel: 1 mA; 4,5 V-on

4 kimeneti port: terhelhető 10 mA-ig

1 PWM-kimenet: terhelhető 10 mA-ig

4 bemeneti port: nyugalmi

állapotban 1 2 analóg

bemenet: 0 V ... VCC

2 gombbemenet: nyugalmi állapotban 1

A tanulócsomagban

lévő alkatrészek:

dugaszártya

elemtartó 3 db AA

elemhez

huzal

HT46F47 a TPS-

firmwarerel

3 db nyomógombos

kapcsoló

4 db LED, piros

1 db 5 mm-es LED, zöld

1 db LDR

3 db tárcsakondenzátor, 100 nF

1 db elektrolit kondenzátor, 47 μ F

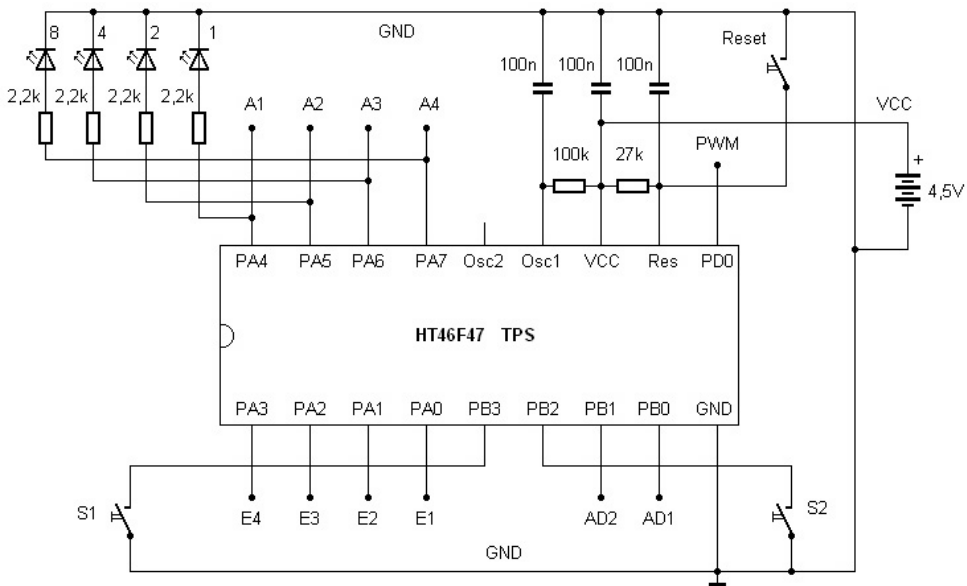
5 db ellenállás 2,2 k Ω

1 db ellenállás, 10 k Ω

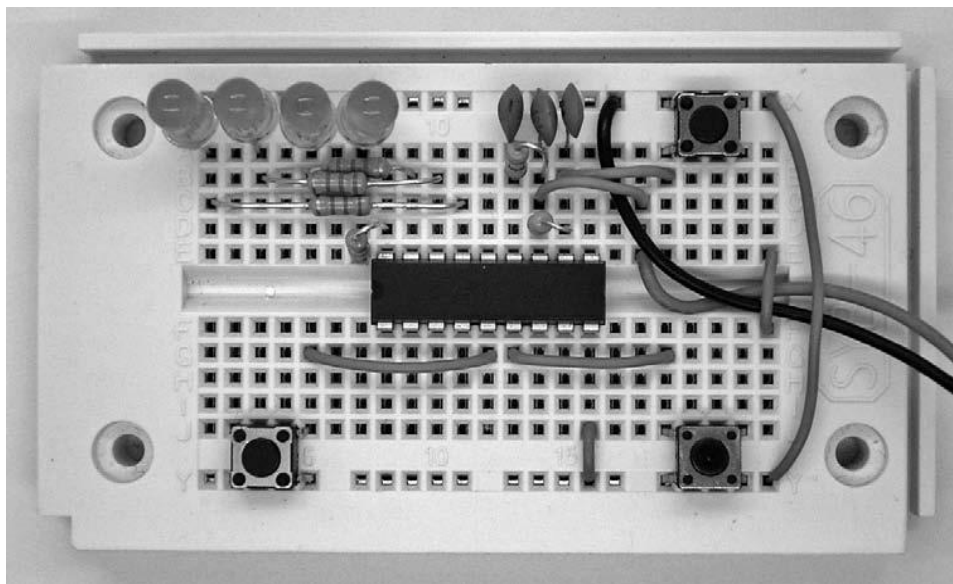
1 db ellenállás, 27 k Ω

2 db ellenállás 100 k Ω

A programozáshoz szükség van az S1 és az S2 nyomógombra, továbbá az A1 - A4 kimeneten lévő négy LED-ből álló egyszerű LED-kijelzőre. Összesen 14 egyszerű parancs van a hozzájuk tartozó adatokkal és alparancsokkal. A parancsok és az adatok minden esetben 4-bites bináris számként vannak kódolva a 0000 - 1111 (decimális 0 - 15) tartományban, és közvetlenül láthatók a LED-kijelzőn. A mindenkor számot programozáskor az S1 gombbal kell programozni. Az S2 gombbal a parancsok és az adatok között lehet váltani, és a parancssor címét növelni lehet. Az egész programszerkezet olyan egyszerű, hogy bizonyos gyakorlattal már fejből elvégezhető.



1. ábra 1: A rendszer alapkapcsolása



2. ábra 2: Alapfelépítés a nyomógombos kapcsolókkal

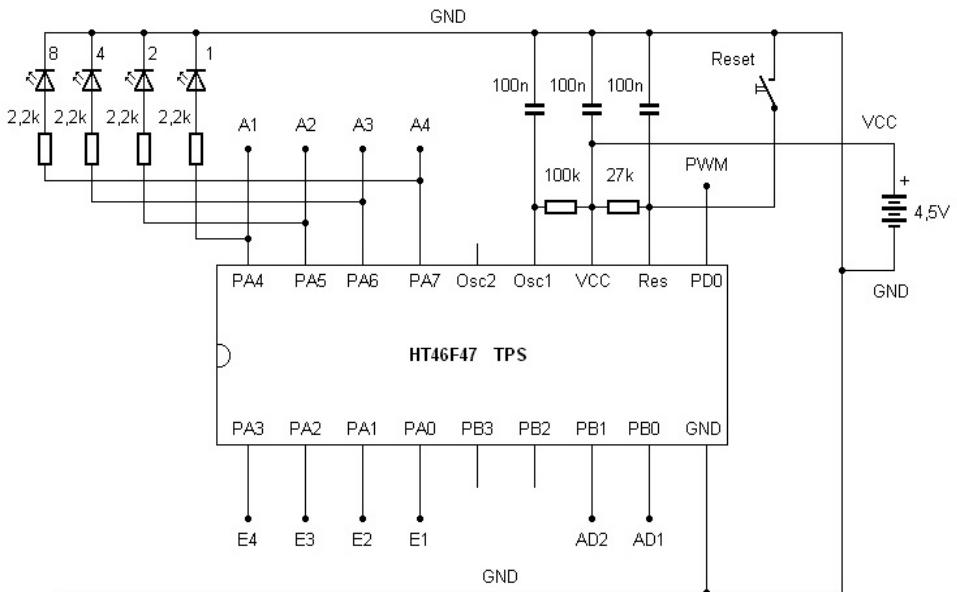
A szállításkor már van néhány alaprogram (default = alapértelmezés) a TPS-kontrollerben, amelyek közvetlenül elindíthatók. Ennek következtében a kontroller lépésről lépésre vehető használatba. Ismerkedjen meg először a hardver-funkciókkal, és csak utána kezdjen bele néhány programba.

Az első tesztek esetében a kontrollerben már készen elérhető kis programokat indítson el. A hozzájuk tartozó programjegyzékek első benyomást nyújtanak a lehetőségekről. Minden esetben magyarázatot is kap hozzájuk. Az egyes parancsok pontos ismertetését a következő fejezet tartalmazza.

Első kísérletként építse fel az alapkonfigurációt a kontrollerrel és a szükséges kiegészítő elemekkel a dugaszkartyán. Feltétlenül szükséges:

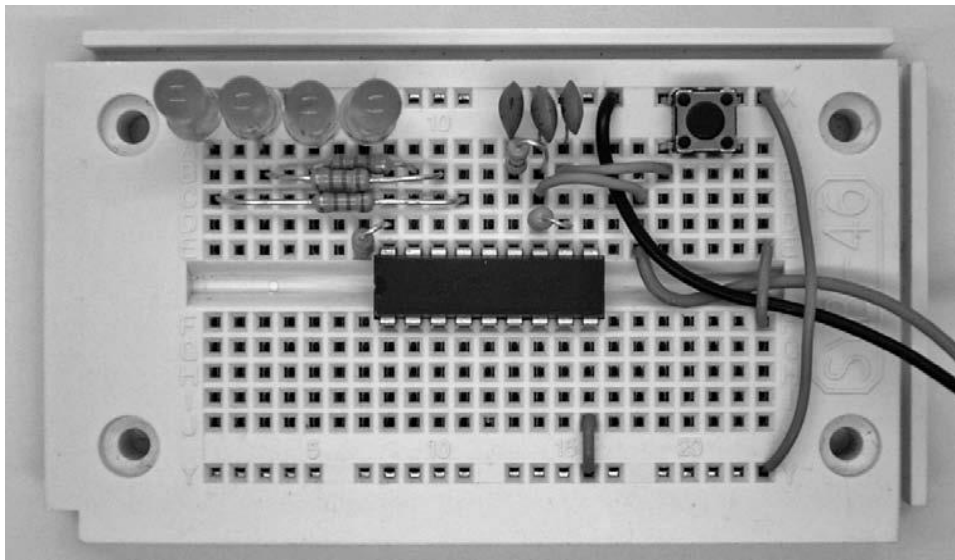
- A tápáramforrás csatlakoztatása a GND (mínusz) és a VCC (plusz) pontra.
- Egy 100 nF-os levezetőkondenzátor a VCC és a GND közé.
- Egy reset-ellenállás a VCC után és egy reset-kondenzátor a GND után.
- Oszcillátorellenállás, 100 kΩ a VCC után, és kondenzátor a GND után.

A HT46F47 mikrokontrollert saját belső RC-oszcillátorával működteti. Az Osc1-bemeneten lévő ellenállás határozza meg az órajel frekvenciáját. A 100 kΩ értékű ellenállással kb. 2 MHz frekvencia áll be. Szükség esetén kisebb vagy nagyobb sebességgel is lehet dolgozni. A csatlakoztatott kondenzátor csak levezetőül szolgál, és nincs befolyása az órajel-frekvenciára. Az Osc2 csatlakozó szabadon marad. Szükség esetén ide és a VCC-re is lehet kötni egy további ellenállást, és az órajel-frekvencia negyedével impulzusokat kicsatolni.



3. ábra 3: Négy LED a kimeneteken

Használja a felső és az alsó tápellátó sít a GND testcsatlakozások számára. Ide kell csatlakoztatni az elemtartó fekete kábelét, azaz a negatív pólust. A VCC pozitív vezetékét kösse össze az elemtartó piros csatlakozókábelével. Feltétlenül kerülni kell a helytelen polaritású csatlakozást, mert tönkremehet a controller. Szereljen be tehermentesítőnek egy rövid huzaldarabot. Az egyszerű már csatlakoztatott tápáramellátást lehetőleg mindig hagyja bekötve. A kikapcsoláshoz vegye ki az egyik elemet a tartóból.



4. ábra 4: A minimális beültetés a LED-ekkel

Ültesse még be a reset-gombot, és kösse be a négy LED-et előtétellenállásokkal (2,2 k Ω). Ezekre az első hardver-tesztekhez lesz már szükség. Figyeljen a sorrendre. Az A1-et a baloldali LED-re, míg az A4-et a jobboldalra kell csatlakoztatni. Ezzel van egy bináris kijelzője a legmagasabb helyértékű bittel. Ez elsősorban a későbbi programozásban lesz hasznos.

2 Váltakozó villogó

Rakjon be most három 1,5 V-os elemet, vagy pedig három NiMh-akkumulátort az elemtartóba. Ezzel elindítja az első példaprogramot a baloldali és a jobboldali LED-ből álló váltakozó villogóval. A villogási frekvencia mintegy 1 Hz. A program-kiírás ezt az egyszerű programot csak öt sorral jeleníti meg. Váltakozva a LED 1 és a LED 8 kapcsolódik be. Köztük van egy várakozási parancs 0,5 m-es várakozási idővel. A visszaugrás az elejére gondoskodik arról, hogy a villogás végtelenül ismétlődjön. Az egyes parancsokat alább még részletesebben is ismertetjük. Ezen a példán azonban már most felismerheti a programozás egyszerűségét. A controller firmwareje az egyszerű parancsokat felismerő és végrehajtó értelmező programot (interpreter) tartalmaz. A programok ennek következtében sokkal tömörebbek, mint más rendszerekben.

A példa a 20h-val (decimális 32) kezdődő címtartományt foglalja el. A felső címtartományba eső több program később ugyancsak saját alkalmazásból indítható. A címek saját programkóddal is felülírhatók. Szükség esetén azonban a controller ismét visszaállítható alapállapotába, amikor is az eredeti példaprogramok helyreállíthatódnak.

Cím	Parancs	Adatok	Megjegyzés
20	1	1	LED1
21	2	8	Várj 500 ms-t
22	1	8	LED8
23	2	8	Várj 500 ms-t
24	3	4	Ugorj -4

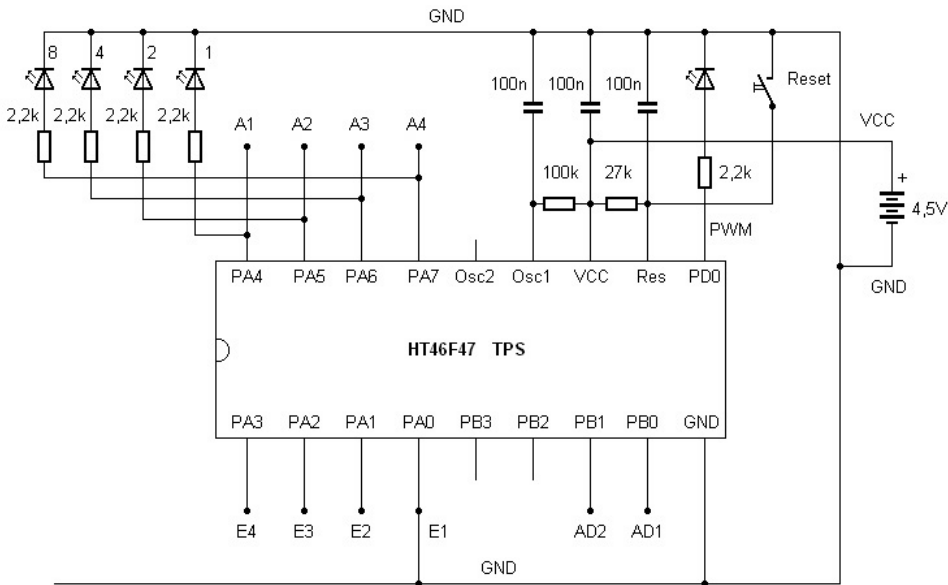
1. kilistázás: Váltakozó villogó

Ha elmarad a kívánt eredmény, vizsgálja meg először a LED-ek helyes polaritását. Segíthet egyes feszültségek megmérése is. Használjon például egy digitális multimétert a 20 V-os tartományban, és rakja a negatív csatlakozását a GND pontra. Az összes feszültséget így a GND ponthoz képest mérheti:

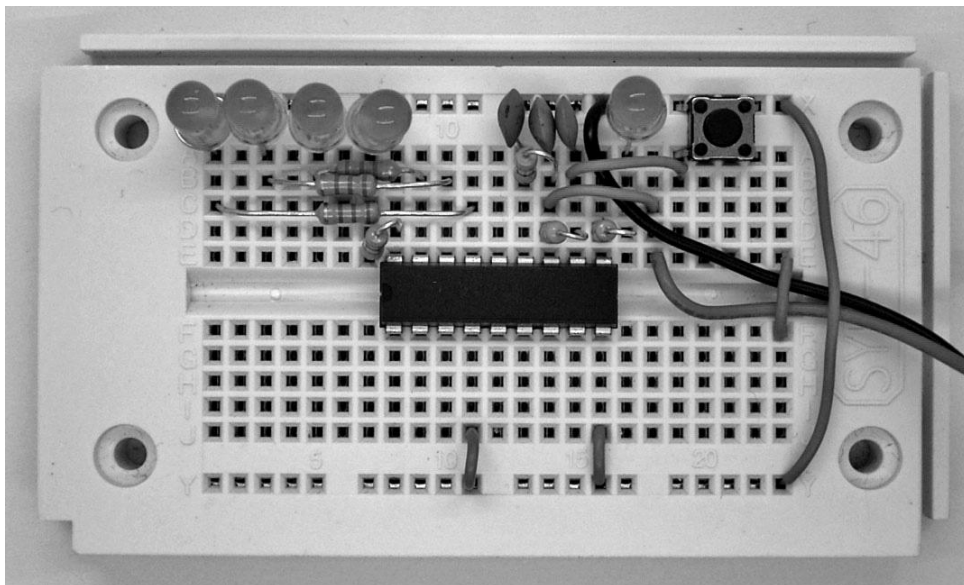
VCC: 4,5 V
reset: 4,5 V
Osc1: 1,5 V
E1 - E4: 4,5 V
A1: váltakozó
A2, A3: 0V
A3: váltakozó

3 Bináris számláló és PWM-kiadás

Az összes digitális bemenetet egy belső ellenállás a VCC-hez képest felhúzza (pull-up ellenállás), és a nyugalmi feszültsége az üzemi feszültség tájékán van. De bármelyik bemenetet egy huzallal vagy érintkezővel a GND szintre hozhatja. Indításkor az alapértelmezett program kiolvassa és kiértékeli a port állapotát. Némelyik csatlakozópont a GND szintre van hozva, így azokon a nulla-állapot olvasható ki. Az eredménytől függően különböző programok felhívása történik meg.



5. ábra 5: A PWM-LED alkalmazása



6. ábra. 6: A bináris számláló elindítása

Rakja az E1-et a GND szintre. Ezáltal egy visszaállítás után elindul a második példaprogram. Ez a kimeneti állapotokat számlálja binárisan felfelé. Folyamatosan végigfut a 0000-tól (decimális 0) a 1111-ig (decimális 15). A program az A változót használja az egyszerű összeadáshoz, és a digitális kimeneten, ill. a PWM-kimeneten történő kiadásra. A 7. és az 5. parancsnak alfunkciói vannak, amelyek adatokként vannak megírva.

Cím	Parancs	Adatok	Megjegyzés
25	7	1	A=A+1
26	5	4	Port=A
27	5	9	PWM=A
28	2	6	Várj 100 ms-t
29	3	4	Ugorj -4

2. listázás : Bináris számláló LED- és PWM-kiadással.

A számolóprogram gyakorlóprogramként használható bináris számok kiolvasására, amelyeket a saját programozáshoz kezelni kell tudni. A négy LED mindegyike egy-egy bitet jelent. Összesen tehát egy 4-bites számot jeleníthetnek meg. A LED-ek a kapcsolási rajzban értéküknek megfelelően a 8, 4, 2 és 1 jelölést viselik. Az adott értékek összeadása révén kapja meg a decimális számot. A hexadecimális írásmódban a 10 - 15 számot az A - F nagybetűkkel jelölik.

8	4	2	1	Decimális	Hexadecimális
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

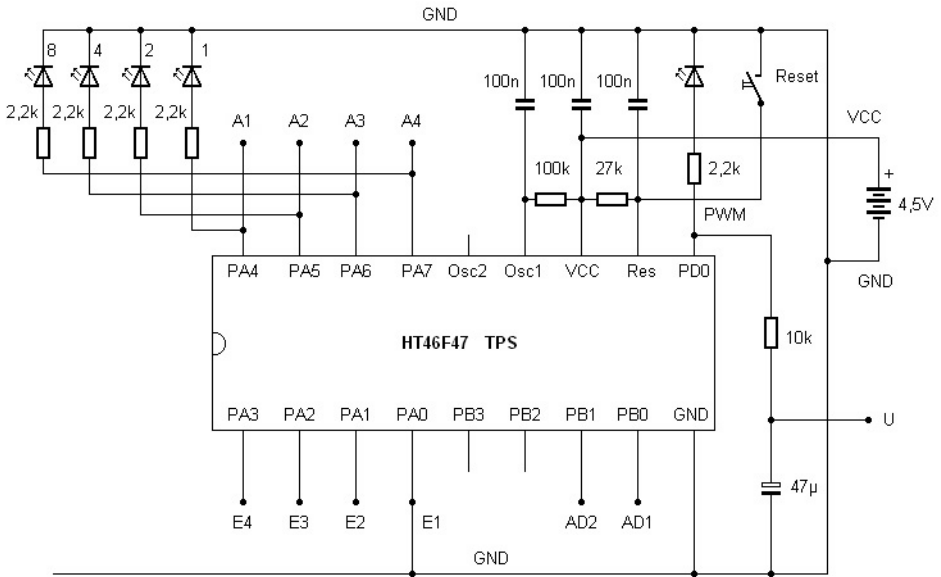
A program használható még különböző frekvenciájú villogóként is.

A következő legmagasabb kimenetnek mindig fele akkora a frekvenciája, ill. kétszer akkora a periódusideje:

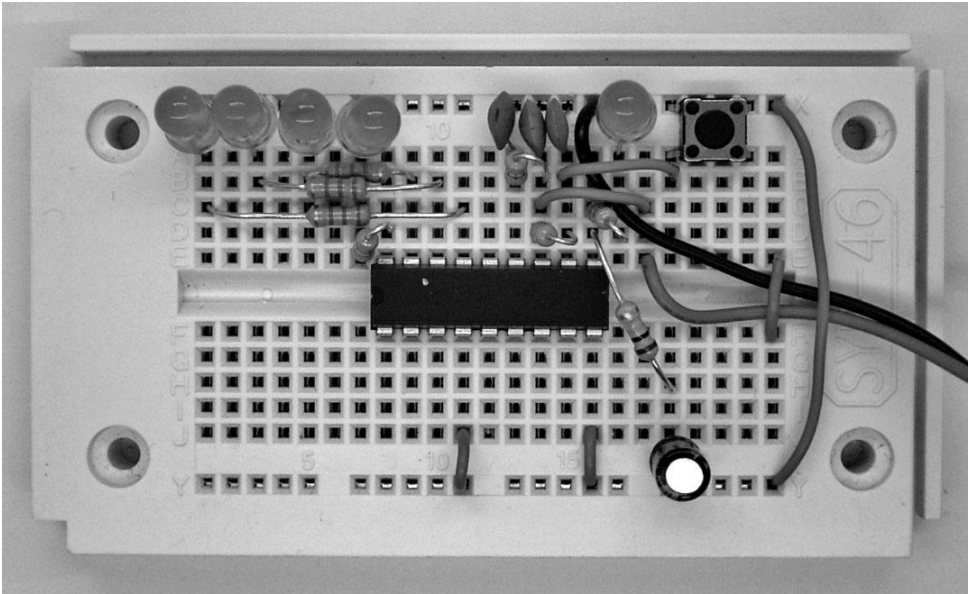
- A1: 200 ms
- A2: 400 ms
- A3: 800 ms
- A4: 1.600 ms

Ezenkívül a növekvő számértékek kiadásra kerülnek a PWM-kimeneten (impulzusszélesség-moduláció) is. A PWM-jel egy kb. 16 kHz frekvenciájú négyzetjel. Itt az impulzushossz szabályozása történik, úgyhogy az impulzus/szünet-arány határozza meg az átlagos bekapcsolás időt, és ezzel a LED fényességét. Az ide csatlakoztatott LED fényessége 15 fokozatban szabályozódik nulla és a teljes fényesség között.

A PWM-jel egy RC-aluláteresztő szűrő segítségével egyenfeszültséggé simítható. A PWM-kimenet ezáltal analóg kimenetté válik. Ezzel a programmal 0 V-tól 4,5 V-ig fokozatmentesen növekvő egyenfeszültséget kap. Kövesse a feszültség menetét egy műszerrel vagy egy oszcilloszkóppal.



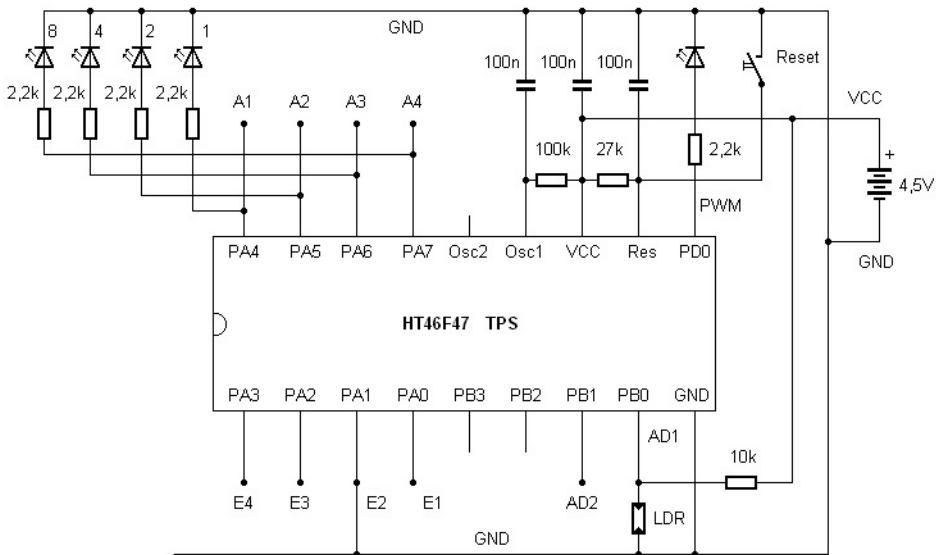
7. ábra 7: Aluláteresztő szűrő a PWM-kimeneten



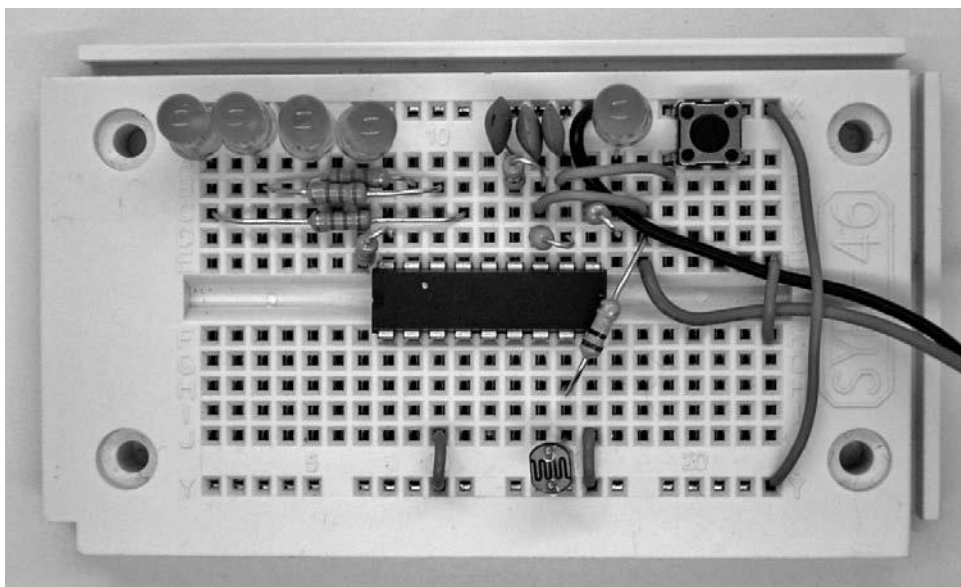
8. ábra 8: Simított PWM-kimenőfeszültség

4 Analóg-digitális-átalakító

Ha az E2-t a GND-re köti, és megnyomja a reset-gombot, elindít egy kis példaprogramot az analóg-digitális-átalakítóhoz (AD-átalakító). Az AD1 analóg bemeneten lévő analóg feszültséget a készülék megméri, és átalakítja digitális számértékké. Mivel a TPS-kontroller kivétel nélkül 4-bites értékekkel dolgozik, az analóg-digitális-átalakítás eredménye a 0-15 tartományba eső szám. A 0 eredmény a 0 bemenőfeszültséget jelenti, míg 15 eredmény az üzemi feszültségnek megfelelő bemenőfeszültséget, azaz pl. 4,5 V-ot. Az AD-érték bináris számként kerül kiadásra a négy LED-en, továbbá a PWM-kimeneten. Csatlakoztasson az AD1 analóg bemenetre egy fix ellenállásból és egy fényérzékeny érzékelőellenállásból (LDR) álló feszültségosztót.



9. ábra 9: A fényérzékelő csatlakoztatása



10. ábra 10: Az LDR az AD1 bemeneten

A példaprogram a digitális kimeneteken és a PWM-kimeneten történő kiadás miatt nagyon hasonló az előző fejezet programjához. Az első sorban azonban az analógérték átalakítási parancsa áll.

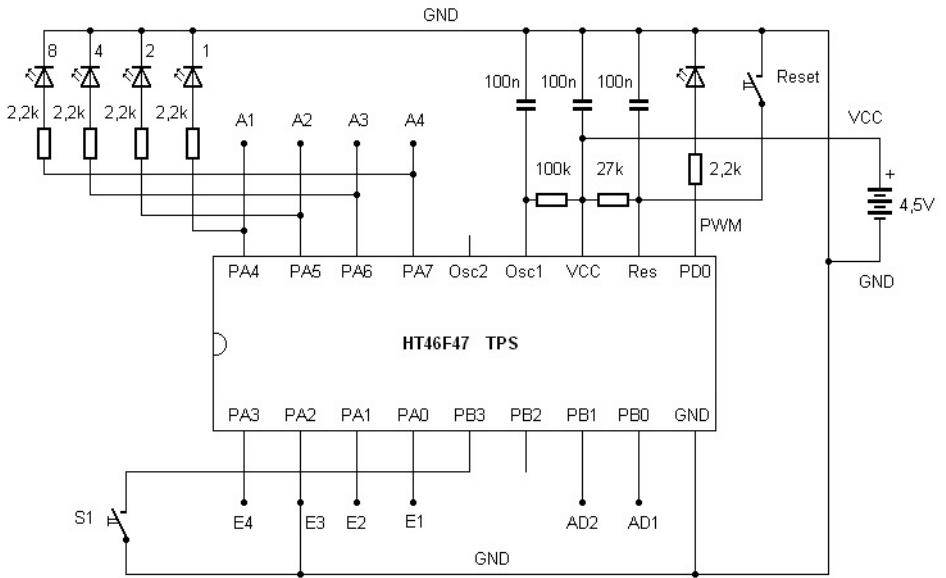
Cím	Parancs	Adatok	Megjegyzés
2A	6	9	A = AD1
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Várj 100 ms-t
2E	3	4	Ugorj -4

3. kilistázás: Az AD-átalakító és a PWM-kimenet

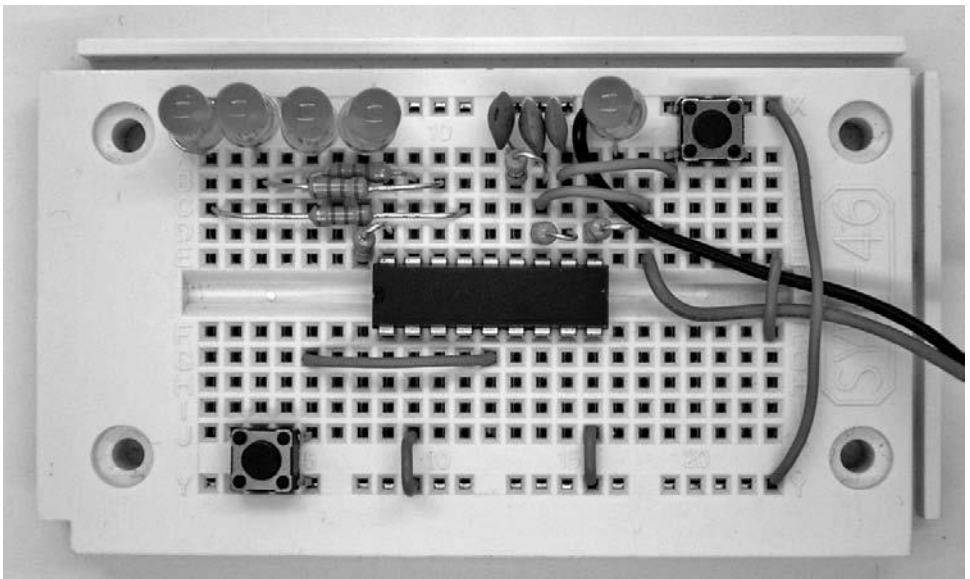
Vizsgálja a programot az érzékelő különböző megvilágításai mellett. Minél több fény esik az LDR-re, annál kisebb a feszültség az AD1-n. És megfordítva, sötétben kapja a maximális AD-értékeket, és ezzel a PWM-kimeneten lévő LED maximális fényességét. Olvassa le a bináris számot a LED-kijelzőről, és próbálja meg pl. a fényességet pontosan a tartomány felére beállítani. A digitális érték ekkor vagy 0111, vagy 1000. Egy kissé villódzó műfénynél előfordulhat, hogy az eredmény a két fokozat között váltakozik.

5 Véletlengenerátor

Az E3 és a GND közötti huzaláthidalással elindít egy példaprogramot egy véletlengenerátor számára. Itt az S1 gomb értékelődik ki. A hozzátartozó bemenet egy belső felhúzó-elenállással rendelkezik, amely a feszültséget a VCC-szintre húzza. A gomb a testre csatlakozik. Egy gombnyomás az S1 bemenetet nullára húzza.



11. ábra 11: A véletlenkapcsoló ellindítása



12. ábra 12: Huzaláthidalás az E3 és a GND között

A program egy feltételes ugrás parancsot alkalmaz. Ha az S1-bemenet állapota Egy, a következő parancs átúgrásra kerül. Ha megnyomja a gombot, az állapot Nulla lesz, és ez által megnő az A változó értéke. Ez a kimenet állapotának a gyors felfelé számlálásához vezet. A gomb felengedésekor az utolsó számlálóállás marad meg. A nagy számolási sebesség miatt nincs befolyásunk az eredményre, azaz az véletlenszerű.

Cím	Parancs	Adatok	Megjegyzés
30	5	4	Port = A
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Ugorj -3

4. kilistázás: Véletlengenerátor

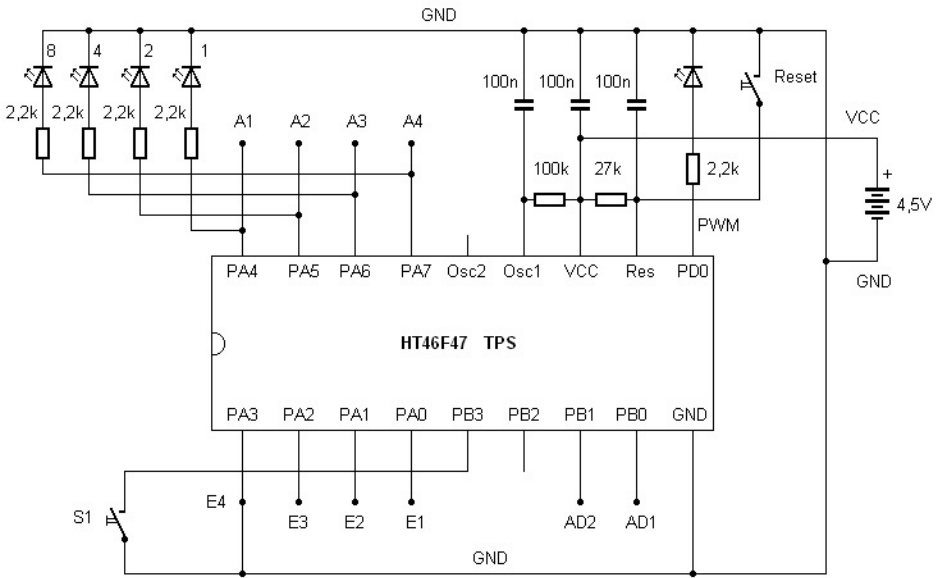
Nyomja meg folyvást a gombot, hogy új és új véletlen eredményt kapjon. Tesztelje a véletlenszerűségi funkciót az eredmények statisztikájának az összeállításával. A lefutások elegendően nagy száma esetén meg kell mutatkoznia annak, hogy az összes eredmény mintegy azonos gyakorisággal fordul elő. A program játékra is alkalmas, pl. "kockával" ki kell dobni a 1111 számot.

Ugyanakkor a program a lehetséges legnagyobb sebességű számláló is, mivel nem alkalmaz várakozási parancsot. Ennek következtében meg tudja vizsgálni ezen a példán a TPS-kontroller működési sebességét. Ameddig a gombot nyomva tartja, az A1 kimeneten egy kb. 133 Hz frekvenciájú és 7,5 ms periódusidejű négyszögjel jelenik meg. A port tehát 3,75 ms-onként változtatja meg az állapotát. A program a számlálókörökben négy parancson fut át. Parancsonként tehát mintegy egy milliszekundumra van szüksége. Az A4 utolsó kimenet 16,6 Hz frekvenciát mutat fel, amely még látható villódzasként ismerhető fel.

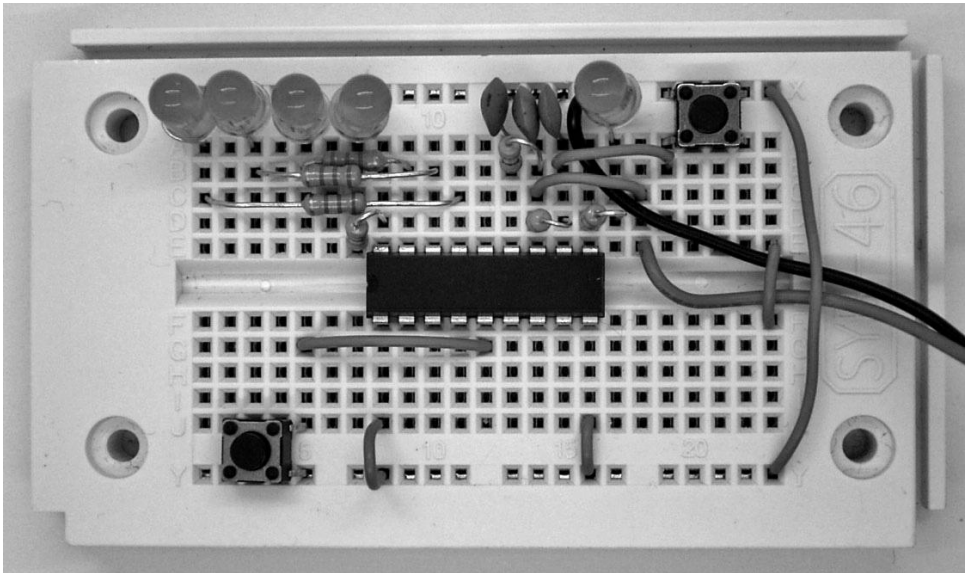
Ha időben kritikus feladatokhoz egyszer nagyobb működési sebességre lenne szüksége, az Osc1-en lévő ellenállás csökkentése révén megnövelheti a kontroller impulzusfrekvenciáját. A 100 kΩ ellenállással az impulzusfrekvencia 2 MHz. Cserélje le az ellenállást egy 27 kΩ-osra. Ezzel majdnem négyszer akkora impulzusfrekvenciát és kb. 0,25 ms parancsidőt kap. Normál esetben azonban a kontrollernek az Osc1-en 100 kΩ -al kell működnie. Ezzel kisebb az áramfelvétel és biztos a működés kisebb üzemi feszültség mellett is egészen 2,2 V-ig.

6 Impulzushosszúság mérése

Az E4 és a GND közötti huzaláthidalással egy reset után elindít egy példaprogramot az impulzushossz méréséhez. Itt is az S1 bemenet állapota értékelődik ki.



13. ábra 13: E4 a GND-ra kötve



14. ábra 14: Az impulzushosszúság mérésének az elindítása

Az időmérés az S1 = 0 állapotban történik, azaz megnyomott gomb mellett. Az 5 ms-os várakozási időhöz még egyszer kb. 5 ms adódik a számlálóhurok összesen öt parancsának a végrehajtásához. A mérés időegysége emiatt 10 ms.

Cím	Parancs	Adatok	Megjegyzés
34	2	2	Várj 5 ms-t
35	C	C	S1 = 0?
36	3	2	Ugorj -2
37	4	0	A = 0
38	2	2	Várj 5 ms-t
39	7	1	A = A + 1
3A	5	4	Port = A
3B	C	E	S1 = 1?
3C	3	4	Ugorj -4
3D	3	9	Ugorj -9

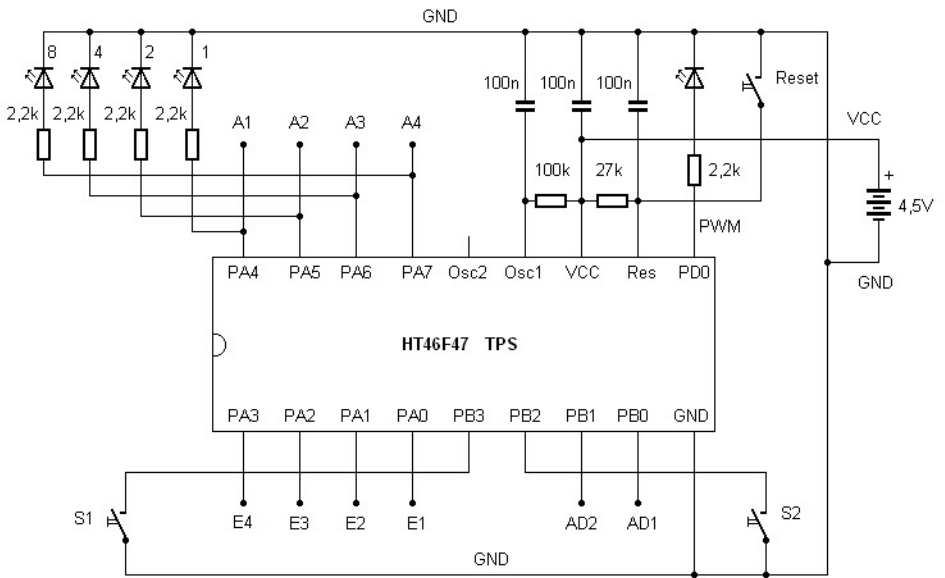
5. kilistázás: Időmérés

Nyomja meg lehetőleg röviden az S1 gombot. Eredményül például 1010-et kap, azaz a decimális 10-et. Mivel a program időegysége 10 ms, a kijelzés jelentése 100 ms. Egy kis edzéssel rövidebb időt is elérhet, lefelé egészen 50 ms-ig.

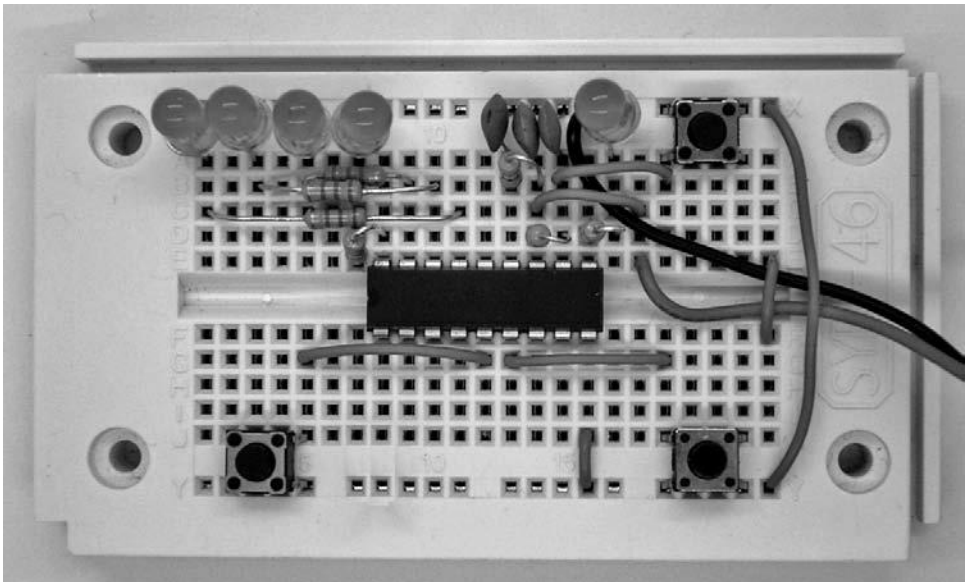
7 Programok kiolvasása

A programozáshoz az S1 (adatbeadás, balra) és az S2 (programozás, jobbra) gombot használhatja. Ezenkívül még a reset-gombra van szüksége. Csupán ezekkel a gombokkal olvashatók ki a programok, és adhatók be a tetszőleges programok. Némi gyakorlattal nagyon rövid idő alatt új programokat adhat be, és módosíthat meglévő programokat.

A programozási üzemmódba úgy juthat, hogy a nyomva tartott S2 programozógomb mellé megnyomja még a reset gombot, és az S2 gombot csak mintegy fél másodperccel a reset gomb után engedi fel. Egyedül az S2 gombbal tállózhat meglévő programban, és megnézheti a parancsokat és az adatokat. Minden cím eközben az S2 gomb kétszeri megnyomását igényli. Így válthat a parancsok és az adatok kijelzése között. Ezenkívül közben rövid időre mindig megjelenik az aktuális cím is.



15. ábra 15: S1 és S2 a programozási üzemmódoz



16. ábra 16: A három gomb és a LED-kijelző

- Az S2 gomb első megnyomása
- A cím (alsó négy bit) megjelenítése, 300 ms
- Kijelző ki, 300 ms
- A parancs megjelenítése
- Az S2 gomb második megnyomása
- Az adatok megjelenítése
- Az S2 gomb harmadik megnyomása
- A következő cím megjelenítése, 300 ms
- stb.

Ha pl. egy meglévő öt lépéses programot csak meg akar nézni, de nem akar módosítani, az S2 gomb összesen tízszeri megnyomásával jut a végére. Miután minden esetben rövid időre megjelenik az aktuális cím, könnyű a tájékozódás. Mindig lehet tudni, hogy a kijelzőn éppen egy parancs vagy adatok láthatók. Szállításkori állapotban az első öt címen a következő parancsok vannak. Ezek a programválaszték kezdetét adják meg az egyes példaprogramok elindításához.

Cím	Parancs	Adatok	Megjegyzés
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 14
03	8	0	AdrHi = 0
04	C	3	A = B?

6. kilistázás: A programkód az alapállapotban

Egy 4-bites parancs és a hozzátartozó 4-bites adatok együtt egy byte-ot képeznek, azaz egy 8-bites számot. Egy fél byte-ot »nibble« (tetrád) néven is hívják. A legmagasabb helyértékű nibble képezi a parancsot, a legkisebb helyértékű nibble a hozzátartozó adatokat. A kontroller EEPROM-ja összesen 128 byte-ot foglal magába. Ezáltal egy program maximum 128 parancsot tartalmazhat. Ez elegendő a legtöbb alkalmazáshoz, mivel a programkód rendkívül tömör. Sok hasznos program tíz sornál kevesebb parancsból áll.

Jelenítse meg az egyes parancsokat és adatokat a kijelzőn, és hasonlítsa össze a memória tartalmát. Majd nyomja meg újra a reset-gombot. A régi program változatlan alakban indul el.

8 Programok beadása

Az S1 gombot csak akkor használja, ha meg akar változtatni egy parancsot vagy az adatait, vagy újat akar beadni. Alapvetően csak 0 és 15 közé eső számértékek adhatók be. Az S1 gomb első megnyomásával 0-t állít be. Minden további gombnyomásra a szám 1-gyel nő. Az aktuális állást mindig a négy LED jelzi binárisan. Ha pl. 4-et akar beadni, összesen ötször kell megnyomnia az S1 gombot: 0, 1, 2, 3, 4. A bináris kijelzőn ekkor 0100 látható.

Ha ezen a módon vagy a parancsot vagy az adatokat, vagy mindkettőt újlag adta be, az S2 gomb második megnyomására ez a byte beprogramozódik az EEPROM-ba. Hogy ez nyilvánvalóvá váljon, a LED-kijelző 600 ms-ig kikapcsolódik, mielőtt a következő cím, majd utána a következő parancs megjelenne rajta. Ezt a szünetet ösztönösen programozási műveletként kell értelmezni. Tudatalattinkba belépülhet

az a képzet, hogy a rendszer takarékoskodik a kijelző energiájával, és a programozásra az EEPROM-ot használja. Ilyesmit ismerünk már autónkból is: Ha az önindítót működtetjük, rövid időre elmegy a világítás és a rádió.

Egy meglévő programnak lehet csak egy helyét módosítani. Az S2 gombbal legördítjük a sorokat a kívánt helyig, és az S1 gombbal megváltoztatjuk a parancsot vagy az adatokat, majd tároljuk őket az S2 gombbal.

Az első teszthez egy csak két parancsból álló programot adjon be. Ez bekapcsol három LED-et, majd egy végtelen hurkot hajt végre.

Cím	Parancs	Adatok	Megjegyzés
00	1	7	A1-4 = 0111
01	3	0	Ugorj 0

7. kilistázás: A LED-ek bekapcsolása

A részletes kilistázás helyett választhatja a rövidített írásmódot. Ekkor a két byte-ot hex számokká fogja össze: 17h, 30h. A továbbiakban alapvetően a hexadecimális írásmódot alkalmazzuk. A programokat emiatt rövidített írásmóddal, a hex-jel nélkül írjuk: 17 30

A beadáshoz a következőket kell beírni:

S2 + reset

2 x S1

S2

8 x S1

S2

4 x S1

S2

1 x S1

S2

Ha egyszer véletlenül túl sokszor nyomja meg az S1 gombot, ennek ellenére elérhető a helyes szám. Ilyenkor még egyszer el kell menni a 15-ön túl, mert utána a 0 értékre történő átmenet következik.

A beadás befejezése után az új programot a reset-gomb indítja el. Látja, hogy a három LED bekapcsolódik. Más nem történik. A kontroller most már nem is reagál az E1 - E4 bemenet állapotára, mivel az eredeti program részben át lett írva. Emiatt a példaprogramok sem indíthatók már el.

Mivel csak az első két memóriacímet változtatta meg, az eredeti program könnyen újra működésbe hozható. Ehhez csak az első két parancsot (64 51) adja be újra az utolsó fejezetben lévő listának megfelelően.

Tesztelje le a példaprogramok eredeti funkcióját. Adja be leginkább az új gyakorlóprogramot még egyszer. Rövid idő alatt bizosságra tesz szert a programbeadás műveleteiben.

9 A példaprogramok helyreállítása

Ha bizonyos idő múlva helyre akarja állítani a kontroller őseredeti állapotát, ezt két FF byte beadásával teheti meg. Valójában ez az állapot a beíratlan EEPROM-nak felel meg. A TPS-kontroller firmwareje egy indítási funkciót tartalmaz, amely először a két első címet vizsgálja meg, hogy felismerje, üres-e a memória. Ha itt két FF-byte olvasható, a kontroller abból indul ki, hogy még nem lett beadva program. Ebben az esetben az EEPROM-ba automatikusan betöltődik a példa-szoftver. Ez a funkció tulajdonképpen arra szolgál, hogy a kontrollert első elindításakor az EEPROM ellássa a példaprogramokkal. Azonban bármikor felhasználható az alapállapot helyreállítására is.

Cím	Parancs	Adatok	Megjegyzés
00	F	F	-
01	F	F	-

8. listázás: Visszatérés az alapállapotr

Indítsa el tehát a programozási üzemmódot az S2-gomb megnyomott állapota mellett végzett resettel. Majd adja be összesen négyszer az F értéket (decimális 15), amikor is világít az összes LED (A1 - A4). Az utolsó beadást is zárja le az S2 gombbal.

Nyomja meg ezután a reset gombot. A kontrollernek a megszokotthoz képest csak egy pillanattal hosszabb időre van szüksége ahhoz, hogy a példaprogramok összes byte-ját újra beprogramozza. Ezzel helyreállt a kontroller őseredeti állapota. Tesztelje pl. a bemenetek huzaláthidalói nélkül a 8. oldalon szereplő váltakozó villogót.

10 TPS-alapparancsok

A nyomógomb-programozású vezérlés összesen 14 parancsot ismer (1–14). Ezek közül sokhoz tartozik egy paraméter is, amely 0000 és 1111 (0–F) közötti négybites szám, azaz a 15-ig terjedő (decimális) számtartomány. Más parancsokhoz alfunkciók tartoznak, amelyek a paraméter formájában adhatók meg. Egy parancskód mögött emiatt akár 16 alparancs is rejtőzhet. Így pl. a 7 parancs mögött a »Rechne A = ...« rejlik. A parameter azt adja meg, hogy milyen számítási műveletet kell elvégezni.

Az alábbiakban a parancsok és az adatok együtt hexadecimális írásmódban egy byte-ként vannak beírva. Az 1. parancsból a 4. paraméterrel együtt így a 14h parancs lesz. A hexadecimális jelet elhagyjuk, mivel az összes parancs és cím alapvetően hexadecimális írásmódban áll.

Az első három parancs így néz ki:

10–1F: Közvetlen port-kiadás az A1–A4-re, 0–15, bináris 0000 - 1111

20–2F: várakozási idő 0–15

(1, 2, 5, 10, 20, 50, 100, 200, 500, 1.000, 2.000, 5.000, 10.000, 20.000, 30.000, 60.000 ms)

30–3F: ugrás vissza 0–15

Az 1. parancs egy állandó szám port-kiadására szolgál. Ezáltal tetszőleges bit-minta adható ki, és pl. több LED is bekapcsolható egyszerre.

A 2. parancs olyan paraméter, amely az időt milliszekundumokban, és egy 1-2-5-sorolást tartalmaz. A 0 -15 értékű csekély számátfogás ellenére ezen a módon egy milliszekundum és egy perc közötti késleltetési idők valósíthatók meg. Még ennél is hosszabb időket kell beprogramozni egy várakozási parancs többszöri végrehajtása céljából, pl. egy számlálólóhurokban.

A 3. visszaugrási parancs különösképpen egyszerű, és sok olyan feladathoz megfelel, ahol egy folyamatot végtelen sokszor ismételni kell. Az ugrás nagysága a 15-ig terjedő tartományra van korlátozva. Miután az ugrás nagysága az aktuális címhez van viszonyítva, a program egyes részei tetszőlegesen eltolhatók egy másik címre ezzel a visszaugrással.

A váltakozó villogó program ezzel a három paranccsal éri be. Itt kissé módosítva a 00-val kezdődő címtartományba kell beírni. A kiadási bit-minta és a várakozási idő is meg lett változtatva.

Cím	Parancs	Adatok	Megjegyzés
00	1	1	A1-4 = 0001
01	2	7	Várj 200 ms-t
02	1	4	A1-4 = 0100
03	2	7	Várj 200 ms-t
04	3	4	Ugorj -4

9. kilistázás: Villogó-program

Hexadecimális rövidített írásmóddal a program most így néz

ki: 11 27 14 27 34

Ennek az első három parancsnak az alapján már számos egyszerű program megírható. Analizálja és tesztelje a következő három programot. A cél az legyen, hogy ezt a parancsot ösztönösen tudja használni. Az ehhez hasonló egyszerű programlefutásokat bizonyos gyakorlattal már fejből is tud programozni és közvetlenül beadni. Egy példa erre egy egyszerű futófény négy kiadási mintával:

Cím	Parancs	Adatok	Megjegyzés
00	1	1	LED-ek 0001
01	2	8	Várj 500 ms-t
02	1	2	LED-ek 0010
03	2	8	Várj 500 ms-t
04	1	4	LED-ek 0100
05	2	8	Várj 500 ms-t
06	1	8	LED-ek 1000
07	2	8	Várj 500 ms-t

Cím	Parancs	Adatok	Megjegyzés
08	3	8	Ugorj -8

11 28 12 28 14 28 18 28 38

10. kilistázás: 1. futófény

Bővítse ki a programot két további kiadási mintával, úgyhogy a fénypont mindig oda- és visszafusson. Kísérletezzen más kiadási mintákkal és késleltetési időkkal.

Cím	Parancs	Adatok	Megjegyzés
00	1	1	LED-ek 0001
01	2	8	Várj 500 ms-t
02	1	2	LED-ek 0010
03	2	8	Várj 500 ms-t
04	1	4	LED-ek 0100
05	2	8	Várj 500 ms-t
06	1	8	LED-ek 1000
07	2	8	Várj 500 ms-t
08	1	4	LED-ek 0100
09	2	8	Várj 500 ms-t
0A	1	2	LED-ek 0010
0B	2	8	Várj 500 ms-t
0C	3	C	Ugorj -12

11 28 12 28 14 28 18 28 14 28 12 28 3C

11. listázás: 2. futófény, oda és vissza

Egy kapcsolóóra egy késleltetési paranccsal egy percig terjedhető késleltetést tartalmazhat. A végén egy visszaugrás áll 0 tartammal, azaz egy végtelen hurok tartalom nélkül, amely a program befejezéseként szolgál. Egy új indítást a reset-gombbal válthat ki. Bővítse ki a programot ezúttal egy háromperces konyhai időzítővé. Ekkor a hátralévő időt a világító LED-ek számával fényoszlopként jelezheti.

Cím	Parancs	Adatok	Megjegyzés
00	1	F	LED-ek 1111
01	2	F	Várj 1 percet
02	1	0	LED-ek 0000
03	3	0	Vége

1F 2F 10 30

12. kilistázás: Egyperces kapcsolóóra

11 Számítás változókkal

Eddig az egyes parancsok paramétereiben csak állandó számértékek nyertek alkalmazást. Ez ésszerű akkor, amikor egy programnak mindig egyformán kell lefutnia. A komplexebb programok azonban változó adatokkal dolgoznak. Pl. esetleg az $A = A + B$ számítást kell elvégezni. Az A és a B változó értékétől függően esetről esetre valami mást kapunk eredményül. Az eredmény vezérelheti pl. a kimeneteken lévő LED-eket.

A szabályzás négy változóval - A, B, C és D - rendelkezik. A legfontosabb változó az A. Szokták *akkumulátor* vagy *rövidenakku* névvel jelölni. Az A az összes számítási műveletben részt vesz és megkapja a számítás eredményét. Azonkívül az A-t az adatok szállítására is használjuk. A B-re főleg számítási feladatokhoz van szükség. A C és a D közbenő memóriaként szolgálhat, és később még számlálólul is használjuk számlálóhurokban.

Ezenkívül van még két analóg bemenet (AD1 és AD2) és egy PWM-kimenet. A feldolgozott adatok négy bitre vannak korlátozva, és csak az A változón keresztül férhetők hozzá ($A = AD1$, $PWM = A$). Az A akkumulátor közvetlenül is megtölthető egy számmal (parancs 40–4F). A B, C vagy D betöltéséhez először be kell tölteni az A-t, és tartalmát a többi változóhoz hozzá kell rendelni (51–53 parancs). Az A-val és a B-vel néhány számítási lépés (71–7A parancs) végezhető el.

A 40–4F parancs az A-hoz új értéket rendel hozzá. Az 51–5A parancscsoport átviszi az A tartalmát egy célhelyre, pl. egy másik változóra vagy a PWM-kimenetre. Ebben a csoportban vannak olyan parancsok is, amelyek a kimeneti port egyetlen bitjét állítják be.

Éppen a másik adatirány áll fenn a 61–6A parancscsoport esetében, amikor is egy forrás adatai kerülnek beolvasásra az A-ba. Végül a 71–7A parancscsoport néhány számolási műveletet hajt végre, amikor is az eredmény alapvetően az A-ban jelenik meg. A Dout kimeneti port az A1 - A4 négy kimenetet foglalja magába, amelyeket vagy közösen, vagy egyenként Dout.0 - Dout.3 egyedi bitként lehet felhívni. Az E1 - E4 bemenet hasonlóképp Din bemeneti portként szólítható meg.

40–4F: $A = 0-15$

51–5A: 1–9 cél = A

51: $B = A$

52: $C = A$

53: $D = A$

54: $Dout = A$ ($Dout = Dki$)

55: $Dout.0 = A.0$

56: $Dout.1 = A.0$

57: $Dout.2 = A.0$

58: $Dout.3 = A.0$

59: $PWM = A$

61–6A: $A = 1-10$ forrás

61: $A = B$

62: $A = C$

63: $A = D$

64: $A = Din$ ($Din = Dbe$)

65: A = Din.0
 66: A = Din.1
 67: A = Din.2
 68: A = Din.3
 69: A = AD1
 6A: A = AD2

71 –7A: A = 1–10 kifejezés
 71: A = A + 1
 72: A = A – 1
 73: A = A + B
 74: A = A – B
 75: A = A * B
 76: A = A / B
 77: A = A And B
 78: A = A Or B
 79: A = A Xor B
 7A: A = Not A

Az A változó alkalmazására egy példát talál a 3. fejezet példaprogramjai között. A program itt a nulla címre lett beállítva, és kissé ki lett bővítve. Hozzájött még egy 0 értékkel definiált kezdés az A változón. A 01 címen egy számolási parancsot talál, itt 1-gyel való növelésre. Az A változó tartalma ezután átadódik a PWM-kimenetre és a kimeneti portra.

Cím	Parancs	Adatok	Megjegyzés
00	4	0	A = 0
01	7	1	A = A + 1
02	5	4	Port = A
03	5	9	PWM = A
04	2	6	Várj 100 ms-t
05	3	4	Ugorj -4

40 71 54 59 26 34

13. listázás: Növelés 1-gyel

Egy további példát már bemutattunk a 4. fejezetben. Az adatok ott az AD1 analóg bemenetről érkeznek, és átvitelre kerülnek a kimeneti portra és a PWM-kimenetre. A módosított program még egy kiegészítő számolási lépést tartalmaz, éspedig az A változó tartalmának az invertálását. Ennek következtében a 0000 értékből az új 1111 érték lesz, azaz 0-ból 15 lesz, és fordítva. Egy növekvő bemeneti feszültség ezen a módon egy csökkenő PWM-kiadáshoz vezet.

Cím	Parancs	Adatok	Megjegyzés
00	6	9	A = AD1
01	5	4	Port = A
02	7	A	A = Not A
03	5	9	PWM = A
04	2	6	Várj 100 ms-t
05	3	5	Ugorj -5

69 54 7A 59 26 35

14. kilistázás: Invertálás

12 Ugrások és elágazások

Eddig csak egy egyszerű ugrásról volt szó (3. parancs), amely maximum 15 címig nyúlt vissza. Most egy abszolút ugrás (Jump) jön hozzá. Mivel az ugrási cél csak 4 bittel adható meg, van egy kiegészítő parancs, amely a cím legmagasabb helyértékű nibblejét határozza meg. Ezáltal a címtartományunk 0–255. Ez több, mint amire szükség van, mivel a kontrollér EEPROM-ja csak 128 byte-ot fog át, tehát a 00 - 7F (decimális 0 - 127) tartományt. A memória ezzel gyakorlatilag nyolc oldalra, a 0 - 7 oldalra (Page 0–7) van beosztva. Egy abszolút ugrás előtt meg kell határozni az ugrás céljának az oldalát.

Két számlálóhurok a C és a D változóval ugyancsak abszolút ugrást hajt végre, amikor itt is, mint korábban, meg kell határozni a cím oldalát.

A feltételes ugrások skip-parancsként (kihagyás, átugrás) működnek. Ha a mindenkor feltétel igaz, egy cím átugrásra kerül. Ott állhatna pl. egy ugrás-parancs vagy akár egy számítási parancs. Feltételként állhatnak rendelkezésre az A és B közötti összehasonlítások, valamint a bemeneti port közvetlen lekérdezései.

Ezenkívül van még egy alprogram-felhívás (Call) és a hozzátartozó visszaugrasi parancs (Return). Bár több alprogram van engedélyezve, de egy alprogramból nem szabad további alprogramot felhívni, mert az értelmező program mindig csak egy visszaugrasi címet jegyez meg.

80–8F: Adr-high = 0–15

90–0F: közvetlen ugrás (Jump) az Adr-high, Adr-low-ra (0–15)

A0–AF: számlálóhurok C-szer Adr-high, Adr-low (0–15)

B0–BF: számlálóhurok D-szer Adr-high, Adr-low (0–15)

C1–CF: feltételes ugrás: ha (feltétel 1–15) akkor átugrani

C1: if A > B then Adr = Adr + 1

C2: if A < B then Adr = Adr + 1

C3: if A = B then Adr = Adr + 1

C4: if Din.0 = 1 then Adr = Adr + 1

C5: if Din.1 = 1 then Adr = Adr + 1
 C6: if Din.2 = 1 then Adr = Adr + 1
 C7: if Din.3 = 1 then Adr = Adr + 1
 C8: if Din.0 = 0 then Adr = Adr + 1
 C9: if Din.1 = 0 then Adr = Adr + 1
 CA: if Din.2 = 0 then Adr = Adr + 1
 CB: if Din.3 = 0 then Adr = Adr + 1
 CC: if S1 = 0 then Adr = Adr + 1
 CD: if S2 = 0 then Adr = Adr + 1
 CE: if S1 = 1 then Adr = Adr + 1
 CF: if S2 = 1 then Adr = Adr + 1

D0–DF: alprogram felhívása (Call) Adr-high, Adr-low (0-15)

E0–EF: visszaugrás az alprogramból (Return)

A feltételes ugrás parancs alkalmazására egy példát talál a 6. fejezet példaprogramjai között. A program itt kissé módosítva a nulla címre lett beállítva. Mivel a cím felső része (Adr-hi) nyugalmi állapotban 0, a kontroller tehát a 0 oldalon kezd, a 80. parancsot itt nem szabad alkalmazni. Ismét egy gombnyomás hosszúságát méri és jelzi ki a program. Az összes várakozási parancs el lett távolítva a programból, úgyhogy most nagyobb időfelbontással dolgozik.

Cím	Parancs	Adatok	Megjegyzés
00	C	C	S1 = 0?
01	3	1	Ugorj -1
02	4	0	A = 0
03	7	1	A = A + 1
04	5	4	Port = A
05	C	E	S1 = 1?
06	3	3	Ugorj -3
07	3	7	Ugorj -7

CC 31 40 71 54 CE 33 37

15. kilistázás: Reakció az S1 gomb megnyomására

A 00 címbe lévő CC ugrásparancs az S1 gomb állapotát értékeli ki. Nyugalmi állapotában az S1 = 1. A feltétel tehát nem igaz, és a 01 címbe lévő parancs nem kerül átugrásra. Ott egy relatív parancs van a kezdetre való ugrásra. A program addig ismétli a 00 és a 01 címbe lévő parancsokat, amíg csak meg van nyomva a gomb. Ezután a feltétel igaz lesz, és a 01 cím átugrásra kerül. Ezzel elkezdődik a tulajdonképpeni mérési művelet. Az akkumulátor törlődik, majd újra és újra 1-gyel növekszik, és kiadódik a LED-ekre. A 05 címbe egy további feltételes ugrásparancs, a CE áll. Itt egy parancs átugrásának a feltétele S1 = 1. Mivel a gomb egyelőre még meg van nyomva, a feltétel nem igaz. A 06 címbe lévő parancs is végrehajtásra kerül, és a 03 címre történő visszaugráshoz vezet. A program csak akkor jut el a 07 címre, amikor felengedi a gombot, és ezáltal a kezdetre történő visszaugráshoz.

Adja be a programot, és tesztelje. A reakcióidő most lényegesen gyorsabb. Az időegység kb. 5 ms.

Az eredeti példaprogram még mindig megvan a memóriában a 34h címtől, mert csak az alacsonyabb címek lettek átírva. Írjon meg egy kis programot, amely csak egy ugrást tartalmaz ezen a címen. Itt először a 3. oldalt kell megadni. Az ezután következő abszolút ugrás a megadott 4 címmel a 34. tényleges címet célozza.

Cím	Parancs	Adatok	Megjegyzés
00	8	3	Oldal 3
01	9	4	Cím = 34

83 94

16. kilistázás: Abszolút ugrás az időmérő programra

Az eredeti példaprogramot ezzel újra felhívja. Tesztelje ezt egyszer más példákra is. Az összes alkalmazható program teljes áttekintését a Függelék tartalmazza.

13 A parancsok áttekintése

Az összes parancs egy pillantásra – ami leegyszerűsíti a munkát a kontrollerral. A következő táblázat tartalmazza a teljes parancskészletet tömör formában.

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port=	Várj	Ugorj	A=	... = A	A = ...	A = ...	Oldal	Ugrás	C*	D*	Hagyja ki, ha...	Hívás	Ret
0	0	1 ms	0	0				0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	A = A+1	1	1	1	1	A>B	1	
2	2	5 ms	2	2	C = A	A = C	A = A-1	2	2	2	2	A<B	2	
3	3	10 ms	3	3	D = A	A = D	A = A+B	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout = A	A = Din	A = A-B	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	A = A*B	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A/B	6	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B		8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B		9	9	9	Din.1 = 0	9	

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
A	10	2 s	10	10		A = AD2	A = Not A		A	A	A	Din.2 = 0	A	
B	11	5 s	11	11					B	B	B	Din.3 = 0	B	
C	12	10 s	12	12					C	C	C	S1 = 0	C	
D	13	20 s	13	13					D	D	D	S2 = 0	D	
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

14 Számlálóhurok

Egy folyamatot pl. pontosan ötször kell végrehajtani. Ehhez egy számlálóhurkot kell képezni. Egy ugrási parancsot ebben az esetben ötször kell végrehajtani, majd többé nem. A számlálótérítő neve C. Az 5 számlálási értéket először az A-ba kell betölteni, majd onnan a C-be. Az A2 parancs egy abszolút ugrást végez a 02 címre, és ugyanakkor 1-gyel csökkenti a C változó tartalmát. Ha a C elérte a 0 értéket, az ugrásra többet már nem kerül sor. Az abszolút ugrás címe a megadott oldalra vonatkozik. A 0 oldalon lévő program esetében elhagyható a 80. oldal parancsa is. Viszont feltétlenül szükség van rá, ha egy másik oldalra kell ugrani.

Cím	Parancs	Adatok	Megjegyzés
00	4	5	A = 5
01	5	2	C = A
02	1	5	Port = 0101
03	2	8	500 ms
04	1	A	Port = 1010
05	2	8	500 ms
06	8	0	Oldal 0
07	A	2	C-mal 02
08	3	0	Vége

45 52 15 28 1A 28 80 A2 30

17. kilistázás: Egy számlálóhurok

Testelje a programot. A LED-ek minden áthaladáskor a 0101 és a 1010 mintát mutatják. Mindenesetre ez a programrészlet nyilvánvalóan nem ötször, hanem hatszor fut le. A 07 címben lévő ugrásparancs ugyan ténylegesen ötször kerül végrehajtásra, azonban abból a célból, hogy az első alkalommal erre a helyre jusson, már egy villogási műveletnek meg kellett történnie. Emiatt a program összesen hatszor villog. Változtassa meg a számlálótérítő értékét 4-re, és testelje újra a programot. Most a LED-ek pontosan ötször villannak.

A számlálóhurkot úgy is lehet alkalmazni, hogy ne hátra, hanem előre ugorjon. Ezúttal a folyamatot ténylegesen ötször megy végbe, ha a C kezdeti értékű 5-öt töltünk be. A minden esetben átugrott 04 cím egy önmagára történő relatív ugrást tartalmaz, és ezzel egy végtelen hurkot, amely a program végéül szolgál.

Cím	Parancs	Adatok	Megjegyzés
00	4	5	A = 5
01	5	2	C = A
02	8	0	AdrHi = 0
03	A	5	C-mal 05
04	3	0	Vége
05	1	5	Port = 0101
06	2	8	Várj 500 ms-t
07	1	A	Port = 1010
08	2	8	Várj 500 ms-t
09	3	6	Ugorj -6

45 52 80 A5 30 15 28 1A 28 36

18. kilistázás: Ötszöri villogás

15 Összehasonlítások

Két számértéket kell összehasonlítani. Az összehasonlítás eredményétől függően kerül sor egy ugrásra. A két számértéknek az A-ban és B-ben kell lennie. A következő példában a B az 5 számmal van megtöltve. Az A az eredményét az AD1 analóg bemenetről kapja. Ide lehet csatlakoztatni például, mint a 4. fejezetben is, egy fényérzékelőt. A programnak most folyamatosan a következő műveletet kell végeznie:

Ha $AD1 > 5$,
akkor: az összes
LED világít,
különben: az
összes LED
kialszik.

Végeredményként egy szűrületi kapcsolót kapunk. Mivel az LDR rá van kötve a GND-ra, a nagyobb fényesség kisebb feszültséget ad az AD1-re. A LED-ek kialszanak, amint egy bizonyos értéknél nagyobb a fényesség, és ezzel együtt egy bizonyos értéknél kisebb a feszültség. A határérték 6, mivel a mérési eredménynek 5-nél nagyobbak kell lennie.

Cím	Parancs	Adatok	Megjegyzés
00	4	5	A = 5
01	5	1	B = A
02	8	0	AdrHi = 0

Cím	Parancs	Adatok	Megjegyzés
03	6	9	A = AD1
04	C	1	Hagyja ki, ha A>
05	9	8	Adr 08
06	1	F	LED-ek 1111
07	3	4	Adr 03
08	1	0	LED-ek 0000
09	3	6	Adr 03

45 51 80 69 C1 98 1F 34 10 36

19. kilistázás: Egyszerű szűrületi kapcsoló

Tesztelje úgy a programot, hogy a tenyerével többé vagy kevésbé eltakarja a fényérzékelőt. Megállapíthatja, hogy az alpműködés teljesül. Mindenesetre egy nem túl szép mellékkörülmény jelenik meg többnyire. Éppen a bekapcsolás-kikapcsolás határán összevissza villognak a LED-ek. Elsősorban műfénynél ingadozik a fényesség gyorsan egy bizonyos középérték körül. Ezt a fényingadozást ugyan a program korrekten értékeli, az eredmény azonban nem olyan, amilyent elvárnánk egy szűrületi kapcsolótól. Egy javított szűrületi kapcsolót mutat be a 18. fejezet.

16 AND, OR és XOR

Két bináris állapot egy új állapottá köthető össze. Egy példa erre az AND (és) funkció: Ha az 1. bit állapota 1, ÉS a 2 bit állapota 1, a kimenet állapota szintén 1. Több bit-ből álló bináris számok is összekapcsolhatók ezen a módon. A »10 AND 3 = 2« érthetővé válik, ha bináris számként írjuk:

1010 AND

0011 =

0010

A következő program összekapcsolja a bemenetek állapotát a 3 konstans számmal. Az AND-funkció hatása ekkor gyakorlatilag az, hogy a két alsó bitet maszkírozza (kiszűri). Nyugalmi állapotban a bemeneti port állapota 1111. Az AND-kapcsolat az 0011-el a LED-eken a 0011 állapotot adj ki. Ha azonban az E1 vagy E2 bemenetet a GND-ra kötjük, a 0-állapot a kimeneten is láthatóvá válik. Az E3-on és E4-en végzett változtatásoknak nincs hatásuk.

Cím	Parancs	Adatok	Megjegyzés
00	6	4	A=Din
01	5	1	B=A
02	4	3	A=3

Cím	Parancs	Adatok	Megjegyzés
03	7	7	A = A és B
04	5	4	Port = A
05	3	5	Ugorj -5

64 51 43 77 54 35

20. kilistázás: Az AND-funkció alkalmazása

Változtassa meg a programot, és teszteljen más logikai funkciókat is. Az OR (vagy) funkció (78) arra használható, hogy adott bemeneti állapotokat alapvetően 1-re állítson: 64 51 43 78 54 35

1010 OR

0011 =

1011

A XOR (kizáró vagy, 79) funkcióval egyes biteket lehet invertálni: 64 51 43 79 54 35

1010 XOR

0011 =

1001

17 Alprogramok

Ha egy program egyes részeit többször kell felhasználni, egy alprogramba írjuk be. Ezáltal gyakran memóriahelyet takarítunk meg, néha pedig sok gombnyomogatást. A következő példa egy alprogram alkalmazását mutatja be, amelyet a főprogram két helyén hívunk fel. Az alprogram ebben az esetben csak egyetlen utasítást ($A = A - 1$) és egy visszaugrási parancsot tartalmaz. Ezzel itt egyáltalán nem takarítunk meg memóriahelyet, hanem a példa csupán a CALL- és a RET-parancsot mutatja be.

Főprogram:

Cím	Parancs	Adatok	Megjegyzés
00	8	0	AdrHi = 0
01	D	8	Hívd 08
02	5	4	Kivétel
03	2	9	Várj 1 s-t
04	D	8	Hívd 08
05	5	4	Kivétel
06	2	8	Várj 0,5 s-t
07	3	7	Ugorj -7

Alprogram:

Cím	Parancs	Adatok	Megjegyzés
08	7	2	A = A-1
09	E	0	Ret

80 D8 54 29 D8 54 28 37 72 E0

21. kilistázás: Alprogramok felhívása

A program eredménye egy elágazó bináris számláló nem egyenlő időkéselettelésekkel. Teszteljen egyszer más parancsokat is az alprogramban.

A szállításkori állapotban lévő példaprogramok között sok hasznos, általános alkalmazású alprogram van. Ezek a Függelékben teljesen fel vannak sorolva. Saját használatra csak a beugrási címnek kell ismertnek lennie:

50: alprogram: hosszú hang

52: alprogram: rövid hang

53: alprogram: tetszőleges hang, hossza az A-ban

60: alprogram: az S1 gomb megnyomásának a figyelése

68: alprogram: az S2 gomb megnyomásának a figyelése

70: alprogram: számbeadás az S1-gyel és az S2-vel

A 60. cím feletti alprogramokat csak akkor alkalmazzuk, ha az S1 gomb által vezérelt számlálót építünk fel. A számlálóállás 0-val kezdődik. A főprogram viszonylag rövid, mivel a gombkérdés komplex feladata az alprogramba van kirámolva.

Cím	Parancs	Adatok	Megjegyzés
00	4	0	A = 0
01	5	4	Kivétel
02	7	1	A = A+1
03	8	6	Oldal 6
04	D	0	Hívd 60,Gomb S1
05	3	4	Ugorj -4

40 54 71 86 D0 34

22. kilistázás: Az S1 által vezérelt számláló

Tesztelje a programot. Ha tízszer megnyomja az S1 gombot, az eredménynek 1010-nek kell lennie. Változtassa meg úgy a programot, hogy a 68 című alprogramot alkalmazza. Most a számláló az S2 gombra reagál.

18 Szürkületi kapcsoló

Egy szürkületi kapcsolónak akkor kell bekapcsolnia a világítást, amikor a környezeti megvilágítás egy megadott határérték alá csökken. És megfordítva, ha kivilágosodik, a világításnak ismét ki kell kapcsolódnia. Gondoskodni kell arról, hogy a világítás ne kapcsolódjon be és ki a határérték környékén. Ezt egy hiszterézis alkalmazásával sikerül megoldani, azaz egy kis távolság hagyásával a bekapcsolási és a kikapcsolási megvilágítás között. Az itt bemutatott program az alábbi szabályok szerint működik:

- Ha az AD1 feszültsége nem nagyobb 5-nél, a világítás kikapcsolódik.
- Ha az AD1 feszültsége nem kisebb 9-nél, a világítás bekapcsolódik.

Ezzel van egy közbenső tartományunk, amelyben nem léphet fel változás a kimenet állapotában. Ez a hézag megakadályozza a LED-ek lobogását.

0–5: a LED-ek nem világítanak

6–8: a LED-ek állapotában nincs változás

9–15: a LED-ek világítanak

Cím	Parancs	Adatok	Megjegyzés
00	1	0	LED-ek 0000
01	4	5	A = 5
02	5	1	B = A
03	6	9	A = AD1
04	C	1	Hagyja ki, ha A>B
05	1	0	LED-ek 0000
06	4	9	A = 9
07	5	1	B = A
08	6	9	A = AD1
09	C	2	Hagyja ki, ha A<B
0A	1	F	LED-ek 1111
0B	3	A	Ugorj -10

10 45 51 69 C1 10 49 51 69 C2 1F 3A

23. kilistázás: Szürkületi kapcsoló hiszterézissel

19 LED-fényszabályzás

Ennek a példaprogramnak a célja egy szabályozható fényű LED-lámpa. A PWM-kimeneten lévő LED fényességét szabályozni kell tudni egy gombbal. Ekkor egy gombot röviden megérintve lehet nyomni ahhoz, hogy a következő fényességi fokozatot elérjük, vagy pedig folyamatosan nyomva lehet tartani, amikor is a fényesség folyamatosan változik.

A program magjában a már jól ismert ugrási parancs kerül alkalmazásra. Ha az adott gomb nincs megnyomva, az akkumulátor tartalmának a növelésére vagy csökkentésére irányuló hozzátartozó parancs átugrásra kerül. A nehézség abban van, hogy ekkor normál esetben felléphet az átfutás 15-ről 0-ra, vagy 0-ról 15-re. Egy kissé több ráfordítást igényel ennek az átfutásnak a megakadályozása. Ehhez ugyanis minden esetben le kell kérdezni, hogy már elértünk-e az alsó végponthoz (0) vagy a felső végponthoz (15). Mivel az összehasonlításban alapvetően az akkumulátor vesz részt, a tartalmát közbenső memóriában kell tárolni. Ehhez a C változót alkalmazzuk.

Cím	Parancs	Adatok	Megjegyzés
00	8	0	AdrHi = 0
01	5	9	PWM = A
02	2	7	200 ms
03	5	2	C = A
04	4	F	A = 15
05	5	1	B = A
06	6	2	A = C
07	C	2	Hagyja ki, ha A<B
08	9	B	Ugorj 0B
09	C	F	Hagyja ki, ha S2 = 1
0A	7	1	A = A + 1
0B	5	2	C = A
0C	4	0	A = 0
0D	5	1	B = A
0E	6	2	A = C
0F	C	1	Hagyja ki, ha A>B
10	9	0	Ugorj 00
11	C	E	Hagyja ki, ha S1 = 1
12	7	2	A = A - 1
15	9	0	Ugorj 00

80 59 27 52 4F 51 62 C2 9B CF 71 52 40 51 62 C1 90 CE 72 90

24. kilistázás: Fényességvezérlés

20 Számzár

Az itt bemutatott számzár bekapcsolja a PWM-kimenetet, ha a felhasználó a helyes számsort adta be. A számbeadásnak pontosan követnie kell az S1 és az S2 gombbal végzett programozás mintáját. Az alábbi program az egyes számoknak az S1 gombbal történő beadását szemlélteti. Mint a programozáskor is, az első gombnyomás a 0000 eredményt adja. Az S1 minden egyes további megnyomása 1-gyel növeli a kiadást. Az S2 gomb megnyomásával befejeződik a beadás. Ebben az esetben a program egy végtelen hurokkal fejeződik be.

Cím	Parancs	Adatok	Megjegyzés
00	C	C	S1 = 0?
01	3	1	Ugorj -1
02	4	0	A = 0
03	5	4	Dout = A
04	2	3	10 ms
05	C	E	S1 = 1?
06	3	2	Adr 04
07	C	F	S2 = 1?
08	3	0	Vége
09	C	C	S1 = 0?
0A	3	3	Adr 07
0B	7	1	A = A + 1
0C	2	3	10 ms
0D	C	C	S1 = 1?
0E	3	1	Adr 0D
0F	3	C	Adr 03

CC 31 40 54 23 CE 32 CF 30 CC 33 71 23 CC 31 3C

25. kilistázás: Egy szám beadása

A számbeadás kész alprogramként áll rendelkezésre a 70. cím alatt. A 08 sorban lévő végtelen hurok helyett itt egy RET-parancs van. Az alprogram az A-ba való számbeadás eredményével záródik.

Az alábbi számzár háromszor hívja fel a számbeadást, és összehasonlítja az eredményeket a programozott számokkal. Ebben a példában a helyes beadás 3, 5, 2. Ezután a PWM-kimenet a 15 értékkel teljesen felvezérlődik. Minden hibás beadás azonban egy végtelen hurokhoz vezet, amely csak egy reset paranccsal hagyható el.

A PWM-kimenetet ebben a példában egy normális digitális portként kezeljük. Erre azért van szükség, mert mind a négy, A1 - A4 kimenetre szükség van a számbeadáshoz. Minden egyes teljes beadás után a négy LED törlődik, hogy egy esetleges megfigyelő személynek lehetőleg kevés betekintése legyen a titkos számkombinációba.

Cím	Parancs	Adatok	Megjegyzés
00	8	7	Oldal 7
01	4	3	A = 3
02	5	1	B = A
03	D	0	Hívás 70
04	C	3	Hagyja ki, ha A=B
05	3	0	Vége
06	1	0	LED-ek ki
07	4	5	A = 5
08	5	1	B = A
09	D	0	Hívás 70
0A	C	3	Hagyja ki, ha A=B
0B	3	0	Vége
0C	1	0	LED-ek ki
0D	4	2	A = 2
0E	5	1	B = A
0F	D	0	Hívás 70
10	C	3	Hagyja ki, ha A=B
11	3	0	Vége
12	1	0	LED-ek ki
13	4	F	A = 15
14	5	9	PWM=A
15	3	0	Vége

87 43 51 D0 C3 30 10 45 51 D0 C3 30 10 42 51 D0 C3 30 10 4F 59 30

26. kilistázás: A számszár

21 Függelék

A példaprogramok kilistázása

Cím	Parancs	Adatok	Megjegyzés
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 1110
03	8	0	Oldal 0
04	C	3	A = B?
05	9	8	Ugorj 08
06	8	2	Oldal 2
07	9	5	Ugorj 25, »nagyszámok«
08	4	D	A = 1101
09	8	0	Oldal 0
0A	C	3	A = B ?
0B	9	E	Ugorj 0E
0C	8	2	Oldal 2
0D	9	A	Ugorj 2A, »AD/PWM«
0E	4	B	A = 1011
0F	8	1	Oldal 1

64 51 4E 80 C3 98 82 95 4D 80 C3 9E 82 9A 4B 81

o. oldal: A példaprogramok kiválasztása és elindítása

Cím	Parancs	Adatok	Megjegyzés
10	C	3	A = B?
11	9	4	Ugorj 14
12	8	3	Oldal 3
13	9	0	Ugorj 30, »véletlen«
14	4	7	A = 0111
15	8	1	Oldal 1
16	C	3	A = B?
17	9	A	Ugorj 1A
18	8	3	Oldal 3

Cím	Parancs	Adatok	Megjegyzés
19	9	4	Ugorj 34, »Stopperóra S1«
1A	4	3	A = 0011
1B	8	2	Oldal 2
1C	C	3	A = B?
1D	9	0	Ugorj 20 »Váltakozó villogó«
1E	8	4	Oldal 4
1F	9	0	Ugorj 40 »Stopperóra S1/S2«

C3 94 83 90 47 81 C3 9A 83 94 43 82 C3 90 84 90

1. oldal: A példaprogramok kiválasztása és elindítása

Cím	Parancs	Adatok	Megjegyzés
20	1	1	0001 »Váltakozó villogó«
21	2	8	Várj 500 ms-t
22	1	8	1000
23	2	8	Várj 500 ms-t
24	3	4	Ugorj -4
25	7	1	A = A + 1 »Magasszámok«
26	5	4	Port = A
27	5	9	PWM = A
28	2	6	Várj 100 ms-t
29	3	4	Springe -4
2A	6	9	A = AD1 »AD/PWM«
2B	5	4	Port = A
2C	5	9	PWM = A
2D	2	6	Várj 100 ms-t
2E	3	4	Ugorj -4
2F	F	F	-

11 28 18 28 34 71 54 59 26 34 69 54 59 26 34 FF

2. oldal: Példaprogramok: váltakozó villogó, felfelé számlálás, AD/PWM

Cím	Parancs	Adatok	Megjegyzés
30	5	4	Port = A »véletlen«
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Ugorj -3
34	2	2	Várj 5 ms-t »Stopperóra S1«
35	C	C	S1 = 0?
36	3	2	Ugorj - 2
37	4	0	A = 0
38	2	2	Várj 5 ms-t
39	7	1	A = A + 1
3A	5	4	Port = A
2B	C	E	S1 = 1?
3C	3	4	Ugorj -4
3D	3	9	Ugorj -9
3E	F	F	-
3F	F	F	-

54 CE 71 33 22 CC 32 40 22 71 54 CE 34 39 FF FF

3. oldal : Példaprogramok: véletlen, S1 stopperóra

Cím	Parancs	Adatok	Megjegyzés
40	8	6	Oldal 6 »Stopperóra Start/Stop«
41	D	0	Felszólítás »Várj S1«
42	4	0	A = 0
43	7	1	A = A + 1
44	5	4	Port = A
45	2	3	Várj 10 ms-t
46	C	D	S2 = 0?
47	3	4	Ugorj -4
48	D	8	Felszólítás »Várj S2«
49	4	0	A = 0
4A	5	4	Port = A
4B	3	B	Ugorj -11

Cím	Parancs	Adatok	Megjegyzés
4C	F	F	-
4D	F	F	-
4E	F	F	-
4F	F	F	-

86 D0 40 71 54 23 CD 34 D8 40 54 3B FF FF FF FF

4. oldal: Példaprogram: stopperóra start/stop

Cím	Parancs	Adatok	Megjegyzés
50	4	F	A = 15 »hosszú hang«
51	9	3	Adr 03
52	4	5	A = 5 »rövid hang«
53	5	3	D = A »változtatható hang«
54	1	9	A4 = 1
55	1	1	A4 = 0
56	2	1	2 ms
57	1	9	A4 = 1
58	1	1	A4 = 0
59	2	1	2 ms
5A	1	9	A4 = 1
5B	1	1	A4 = 0
5C	2	0	1 ms
5D	B	4	Dmal 04
5E	1	0	Dout 0
5F	E	0	Vissza

4F 93 45 53 19 11 21 19 11 21 19 11 20 B4 10 E0

5. oldal: Alprogram: hangkiadás

Cím	Parancs	Adatok	Megjegyzés
60	2	3	Várj 10 ms-t »Várj S1«
61	C	E	S1 = 1?
62	3	2	Ugorj -2
63	2	3	Várj 10 ms-t
64	C	C	S1 = 0?
65	3	1	Ugorj -1
66	E	0	Vissza
67	F	F	-
68	2	3	Várj 10 ms-t »Várj S2«
69	C	F	S2 = 1?
6A	3	2	Ugorj -2
6B	2	3	Várj 10 ms-t
6C	C	D	S2 = 0?
6D	3	1	Ugorj -1
6E	E	0	Vissza
6F	F	F	-

23 CE 32 23 CC 31 E0 FF 23 CF 32 23 CD 31 E0 FF

6. oldal: Alprogramok: S1 megfigyelése és S2 megfigyelése

Cím	Parancs	Adatok	Megjegyzés
70	C	C	S1 = 0? »gombbevitel«
71	3	1	Ugorj -1
72	4	0	A = 0
73	5	4	Port = A
74	2	3	Várj 10 ms-t
75	C	E	S1 = 1?
76	3	2	Ugorj -2
77	C	F	S2 = 1?
78	E	0	Vissza
79	C	C	S1 = 0?
7A	3	3	Ugorj -3
7B	7	1	A = A + 1

Cím	Parancs	Adatok	Megjegyzés
7C	2	3	Várj 10 ms-t
7D	C	C	S1 = 1?
7E	3	1	Ugorj - 1
7F	3	C	Ugorj -12

CC 31 40 54 23 CE 32 CF E0 CC 33 71 23 CC 31 3C

7. oldal: Alprogram: beadás nyomógombbal

Parancstáblázat

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port =	Várj	Ugorj-	A=	...=A	A=...	A=...	Oldal	Ugorj	C*	D*	Hagyja ki, ha	Hívás	Ret
0	0	1 ms	0	0				0	0	0	0		0	
1	1	2 ms	1	1	B = A	A = B	A = A+1	1	1	1	1	A>B	1	
2	2	5 ms	2	2	C = A	A = C	A = A-1	2	2	2	2	A<B	2	
3	3	10 ms	3	3	D = A	A = D	A = A+B	3	3	3	3	A = B	3	
4	4	20 ms	4	4	Dout.0 = A	A = Din	A = A-B	4	4	4	4	Din.0 = 1	4	
5	5	50 ms	5	5	Dout.0 = A.0	A = Din.0	A = A*B	5	5	5	5	Din.1 = 1	5	
6	6	100 ms	6	6	Dout.1 = A.0	A = Din.1	A = A/B	6	6	6	6	Din.2 = 1	6	
7	7	200 ms	7	7	Dout.2 = A.0	A = Din.2	A = A And B	7	7	7	7	Din.3 = 1	7	
8	8	500 ms	8	8	Dout.3 = A.0	A = Din.3	A = A Or B		8	8	8	Din.0 = 0	8	
9	9	1 s	9	9	PWM = A	A = AD1	A = A Xor B		9	9	9	Din.1 = 0	9	
A	10	2 s	10	10		A = AD2	A = Not A		A	A	A	Din.2 = 0	A	
B	11	5 s	11	11					B	B	B	Din.3 = 0	B	
C	12	10 s	12	12					C	C	C	S1 = 0	C	
D	13	20 s	13	13					D	D	D	S2 = 0	D	
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

Impresszum

© 2011 Franzis Verlag GmbH,
85540 Haar bei München, www.elo-
web.de

Autor: Burkhard Kainka

ISBN 978-3-645-10104-

2

Készült a Conrad Electronic SE megbízásából, Klaus-Conrad-Str. 1, 92240 Hirschau

Minden jog fenntartva, a fotómechanikus lejátszása és az elektronikus médiakon történő tárolása is. Csak a kiadó írásos engedélyével szabad másolatokat készíteni és terjeszteni papíron, adathordozókon vagy az interneten, különösen PDF-fájlként, ellenkező esetben büntetőjogi következményekkel járhat.

A hardver és szoftver termékmegnevezések többsége, valamint a jelen leírásban szereplő céges logók rendszerint bejegyzett termékmegjelölések, és akként kezelendők. A kiadó lényegében a gyártó írásmódját alkalmazza a termékmegnevezéseknél.

Az ebben a kézikönyvben bemutatott összes kapcsolást és programot a lehető legnagyobb gondossággal fejlesztettük ki, vizsgáltuk be és teszteltük. Ennek ellenére nem lehet teljesen kizárni a kézikönyvben és a szoftverben előforduló hibákat. A kiadó és a szerző a szándékos vagy hanyag magatartás miatt a törvény szabta felelősséggel tartozik. Egyebekben a kiadó és a szerző már csak a termékszavatosságnak megfelelően tartozik felelősséggel az élet, a test vagy az egészség sérelme, vagy a lényeges szerződéses kötelezettségek vétkes megsértése esetén. A lényeges szerződéses kötelezettségek megsértése miatti kártérítés a szerződésre jellemző előrelátható károokra korlátozódik, hacsak a termékszavatosság szerinti kényszerítő felelősség esete nem áll fenn.



Az elektromos és elektronikus készülékeket tilos a háztartási hulladékkal együtt eltávolítani. Az elhasznált terméket az érvényes törvényi előírásoknak megfelelően kell eltávolítani. Az elektromos készülékeit az eltávolítás céljára rendszeresített gyűjtőállomásokon ingyenesen leadhatja. Lakhelyén a hatóságoknál informálódhat, hol talál ilyen gyűjtőállomást.



A termék megfelel a vonatkozó CE irányelveknek, amennyiben azt a mellékelt használati útmutató szerint kezeli. Az útmutató a termékhez tartozik, és tovább kell adnia a készülékkel együtt, ha azt továbbadja.