



NAVODILA ZA UPORABO

Profesionalni učni komplet za mikrokontrolerike Conrad

Kataloška št.: 19 22 86



Programirljiv krmilnik s tipkami

Mikrokrmilnike najdemo povsod: v gospodinjstkih aparatih, napravah zabavne elektronike, vozilih, merilnikih in celo v vesoljskih plovilih brez posadke. Povsod počnejo stvari, ki jim jih ukaže program. Zelo zabavno je tudi samostojno ustvarjanje enostavnih krmilnih programov.

Prvi korak je vedno izbira mikrokrmilnika ali procesorja, ki se karseda natančno ujema z želeno nalogo. Izbirate lahko med nešteti vrstami, ki jih ponujajo različni proizvajalci. Izberete lahko tudi programski jezik. Ponavadi sta v ponudbi programska jezika Assembler in C, v številnih primerih tudi Basic ali kak drug jezik. Ponavadi za programiranje potrebujete drago programsko opremo ali programirno napravo. Poleg velikega finančnega zalogaja pa ne smemo zanemariti tudi časa uvajanja.

Mikrokrmilnik, ki je uporabljen v tem učnem kompletu, je čisto nekaj drugega. Za programiranje ne potrebujete ničesar drugega kot dve tipkali (stikali s tipko). Krmilnik, ki ga programirate s tipkami, pozna relativno malo ukazov, ki se jih z lahkoto naučite in jih programirate neposredno v krmilnik s pomočjo tipk. Program lahko kadarkoli spremenite in to brez uporabe posebnih pripomočkov.

Sistem je posebej primeren za kompaktne aplikacije na področju meritev, krmiljenja in regulacije. Številne naloge so s tem sistemom že v celoti rešljive. Poleg tega lahko po uspešnem programiranju mikrokrmilnik vgradite v lastna vezja. Pri tem se predpostavlja osnovno poznavanje elektronike.

Sistem je hkrati primeren tudi kot osnova za izobraževanje in za prve korake pri programiranju mikrokrmilnikov. Uspeh je hitreje viden kot pri drugih sistemih. Vendar pa so strukture podobne kot pri drugih programskih jezikih, s čimer je kasnejši prehod olajšan.

Kazalo

Programirljiv krmilnik s tipkami	2
Kazalo	2
Uvod	3
Izmenična utripalka	6
Binarni števec in PWM izhodni signal	7
Analogno-digitalni pretvornik	10
Naključni generator	12
Merjenje dolžine impulza	14
Branje programov	15
Vnašanje programov	17
Obnovitev vzorčnih programov	18
Osnovni ukazi za programirljiv krmilnik s tipkami	18
Računanje s spremenljivkami	20
Skoki in razcepi	22
Pregled ukazov	24
For zanke	25
Primerjave	26
AND, OR in XOR	26
Podprogrami	27
Stikalo za mrak	28
LED-zatemnilnik	29
Kodna ključavnica	30
Priloga	31
Garancijski list	36

Uvod

Načelo programirljivega krmilnika s tipkami je enostavno. Na voljo imate štiri digitalne vhode od E1 do E4 in štiri digitalne izhode od A1 do A4. Poleg tega sta na voljo tudi dva analogna vhoda AD1 in AD2 ter en kvazi-analogni PWM-izhod. Ponastavitveni vhod (Reset) s priklučeno ponastavitveno tipko program ponastavi na začetek. Napajanje krmilnika poteka s tremi baterijami tipa AA s pribl. 4,5 V in lahko deluje na območju od 2,2 V do 5,5 V.

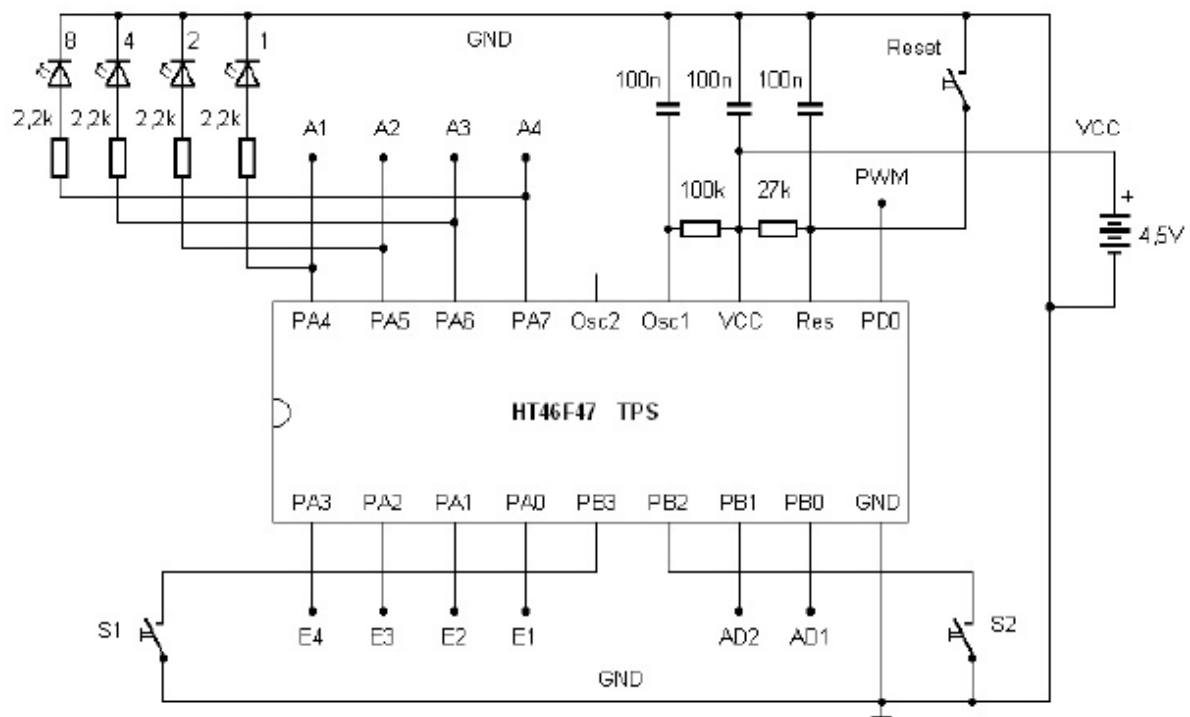
Tehnični podatki:

Mikrokrmilnik: HT46F47
Taktna frekvenca: 2 MHz
Notranji EEPROM: 128 bajtov
Napajanje VCC: 2,2 V do 5,5 V
Poraba toka: 1 mA pri 4,5 V
4 izhodna vrata: obremenljiva do 10 mA
1 PWM-izhod: obremenljiv do 10 mA
4 vhodna vrata: mirovanje 1
2 analogna vhoda: 0 V ... VCC
2 vhoda za tipke: mirovanje 1

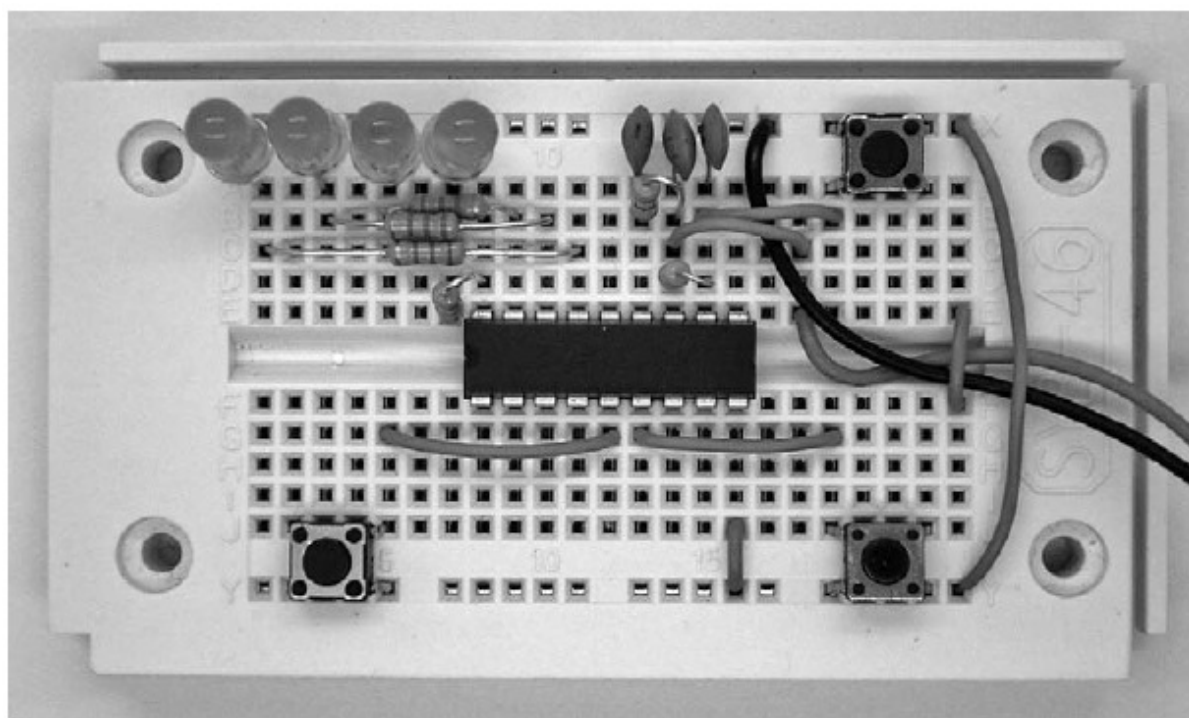
Komponente v učnem kompletu:

Preizkusna ploščica
Baterijski predal za 3 baterije tipa AA
Žica
HT46F47 s strojno-programsko opremo za programiranje s tipkami
2 tipkali
4 LED-diode, 5 mm, rdeče
1 LED-dioda, 5 mm, zelena
1 svetlobno občutljiv upor (LDR)
3 ploščati kondenzatorji, 100 nF
1 elektrolitski kondenzator, 47 μ F
5 uporov, 2,2 k Ω
1 upor, 10 k Ω
1 upor, 27 k Ω
2 upora, 100 k Ω

Za programiranje potrebujete tipki S1 in S2 ter enostavni LED-prikaz, ki je sestavljen iz 4 LED-diod, na izhodih od A1 do A4. Skupno je na voljo 14 enostavnih ukazov s pripadajočimi podatki in podukazi. Ukazi in podatki so posamezno kodirani kot 4-bitna binarna števila na območju od 0000 do 1111 (decimalno od 0 do 15) in so neposredno vidni na LED-diodah za prikaz. Posamezno število pri programiranju programirate s pritiski na tipko S1. S tipko S2 preklapljate med posameznim ukazom in podatki in povišate naslov ukazne vrstice. Celotna struktura programa je tako enostavna, da jo lahko z nekaj vaje obvladate kar iz glave.



Slika 1: Osnovna vezava sistema



Slika 2: Standardna sestava s tipkali

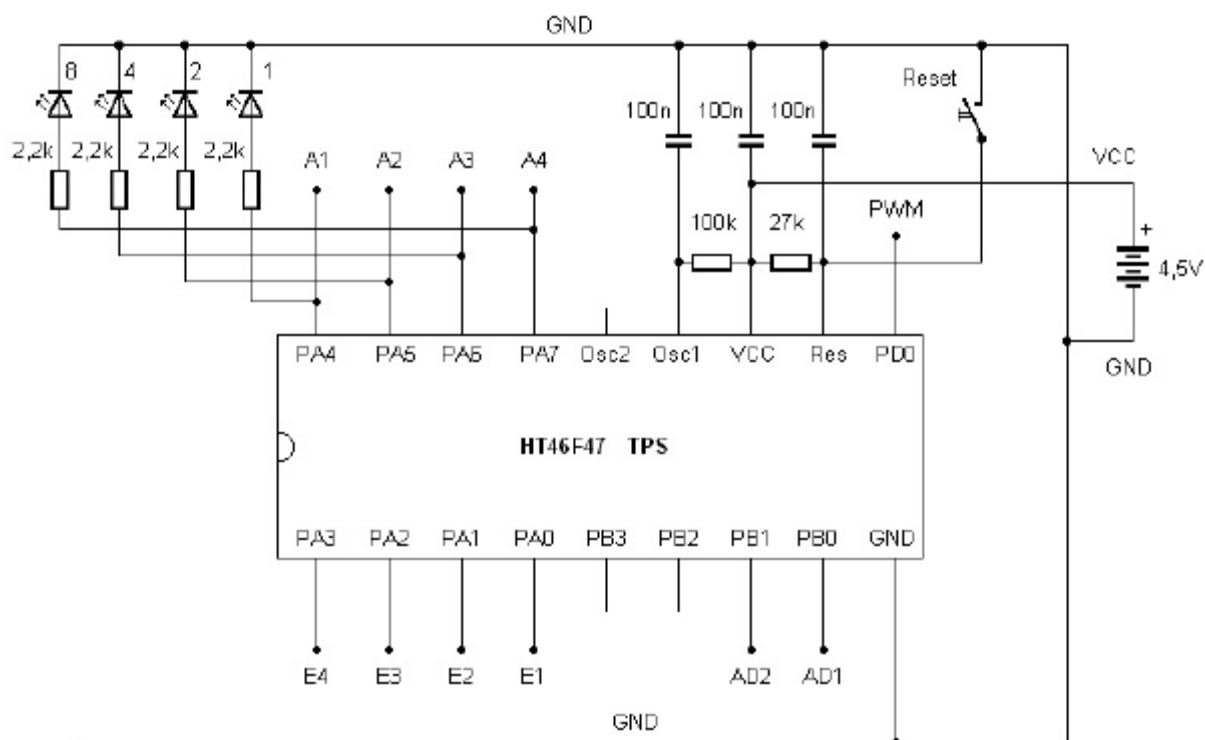
Nekaj osnovnih programov je že v stanju tovarniških nastavitvev (standardnih) na voljo v programirljivem krmilniku s tipkami in jih lahko neposredno zaženete. Tako lahko krmilnik začnete uporabljati korak za korakom. Najprej se seznanite s funkcijami strojne opreme, šele nato se lotite lastnih programov.

Pri prvih testih se programi, ki so že v stanju tovarniških nastavitvev na voljo v krmilniku, ne zaženejo. S pripadajočimi seznanji programov boste dobili prvi vtis o možnostih. Vsak posamezni program bo na kratko razložen. Natančno razlago posameznih ukazov najdete v naslednjih poglavjih.

Za prvi preizkus na preizkusni ploščici sestavite samo osnovno konfiguracijo s krmilnikom in potrebnimi dodatnimi elementi. Nujno potrebujete naslednje:

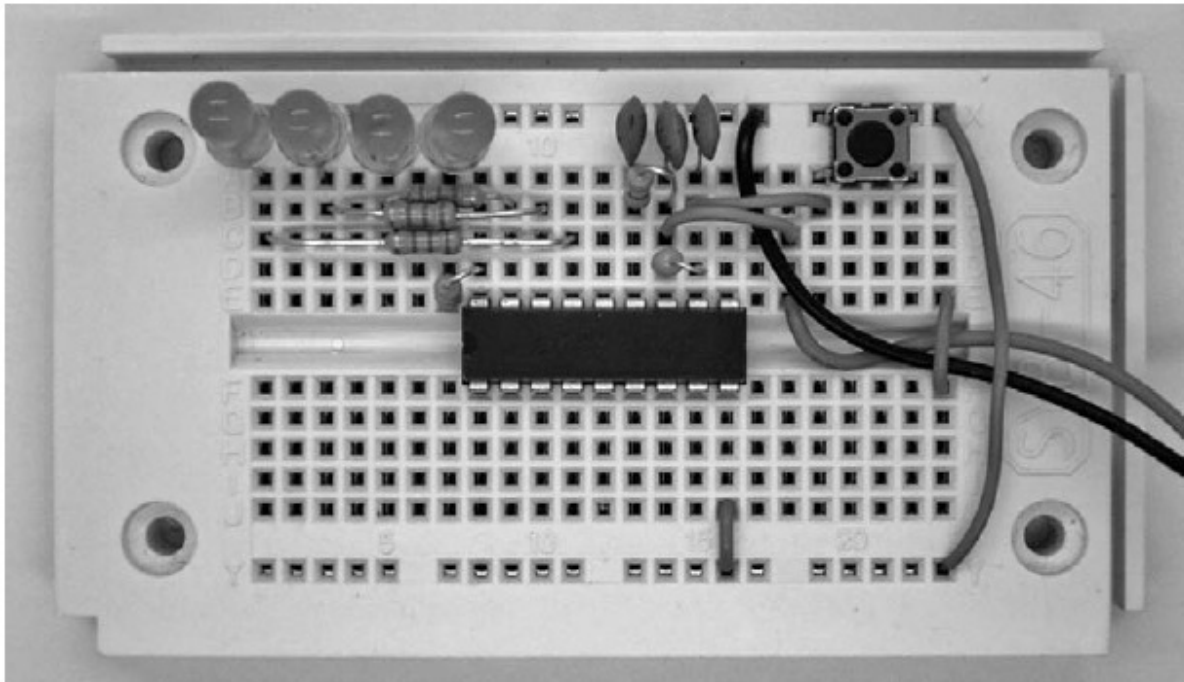
- Priključitev napajanja na GND (minus) in VCC (plus)
- Blokirni kondenzator s 100 nF med VCC in GND
- Ponastavitveni upor za VCC in ponastavitveni kondenzator za GND
- Upor za oscilator s 100 kΩ za VCC in kondenzator za GNC

Napajanje mikrokrmilnika HT46F47 poteka z notranjim RC-oscilatorjem. Upor na vhodu Osc1 določa taktno frekvenco. S 100 kΩ je nastavljena frekvenca pribl. 2 MHz. Po potrebi lahko delate z večjo ali manjšo hitrostjo. Priključen kondenzator služi samo blokiranju in nima vpliva na taktno frekvenco. Priključek Osc2 ostane prost. Po potrebi pa lahko tukaj priključite dodaten upor proti VCC in ločite impulze s četrtno taktne frekvence.



Slika 3: Štiri LED-diode na izhodih

Kot priključek za maso GND uporabite zgornjo in spodnjo napajalno vodilo na preizkusni ploščici. Tukaj priključite črni kabel baterijskega predala, torej minus pol. Plus vodnik VCC povežite z rdečim priključnim kablom baterijskega predala. Nujno se izognite priključitvi napačnih polov, sicer se lahko krmilnik uniči. Vgradite kratek kos žice, ki bo služil kot zaščita pred potegom. Ko enkrat priključite napajanje, potem naj bo po možnosti vedno povezano. Za izklop iz baterijskega predala preprosto odstranite eno izmed baterij.



Slika 4: Minimalno vezje z LED-diodami

Dodatno že vstavite ponastavitveno tipkalo in priključite štiri LED-diode s predupori po 2,2 k Ω . Potrebujete jih za prva testiranja strojne opreme. Bodite pozorni na zaporedje. A1 je treba priključiti na levo LED-diodo in A4 na desno. S tem imate binarni prikaz z najvišjim bitom na levi strani. To je koristno predvsem pri kasnejšem programiranju.

Izmenična utripalka

V baterijski predal vstavite tri 1,5 V baterije ali po želji tri NiMH akumulatorske baterije. S tem boste zagnali prvi vzorčni program z eno izmenično utripalko z levo in desno LED-diodo. Frekvenca utripanja znaša pribl. 1 Hz. Izpis programa prikazuje enostavni program s samo 5 vrsticami. Izmenično se vklapljata LED 1 in LED 8. Vmes se nahajajo čakalni ukazi s čakalnim časom 0,5 sekund. Vrnitev na začetek poskrbi za to, da se utripanje v nedogled ponavlja. Posamezni ukazi so še natančneje razloženi v nadaljevanju. Vendar lahko že na tem primeru razberete enostavnost programiranja. Strojno-programaska oprema krmilnika ima interpreta, ki zazna in izvede enostavne ukaze. Programi so zato nekajkrat bolj kompaktni kot pri drugih sistemih.

Primer zaseda naslovno območje od 20h naprej (decimalno 32). Kasneje lahko iz lastnih aplikacij zaženete več programov v zgornjem naslovnem območju. Naslove lahko po želji tudi prepisete z lastno programsko kodo. Po potrebi lahko krmilnik tudi povrnete v osnovno stanje, pri čemer se izvorni vzorčni programi ponovno vzpostavijo.

Naslov	Ukaz	Podatki	Komentar
20	1	1	LED 1
21	2	8	Čakaj 500 ms
22	1	8	LED 8
23	2	8	Čakaj 500 ms
24	3	4	Skoči -4

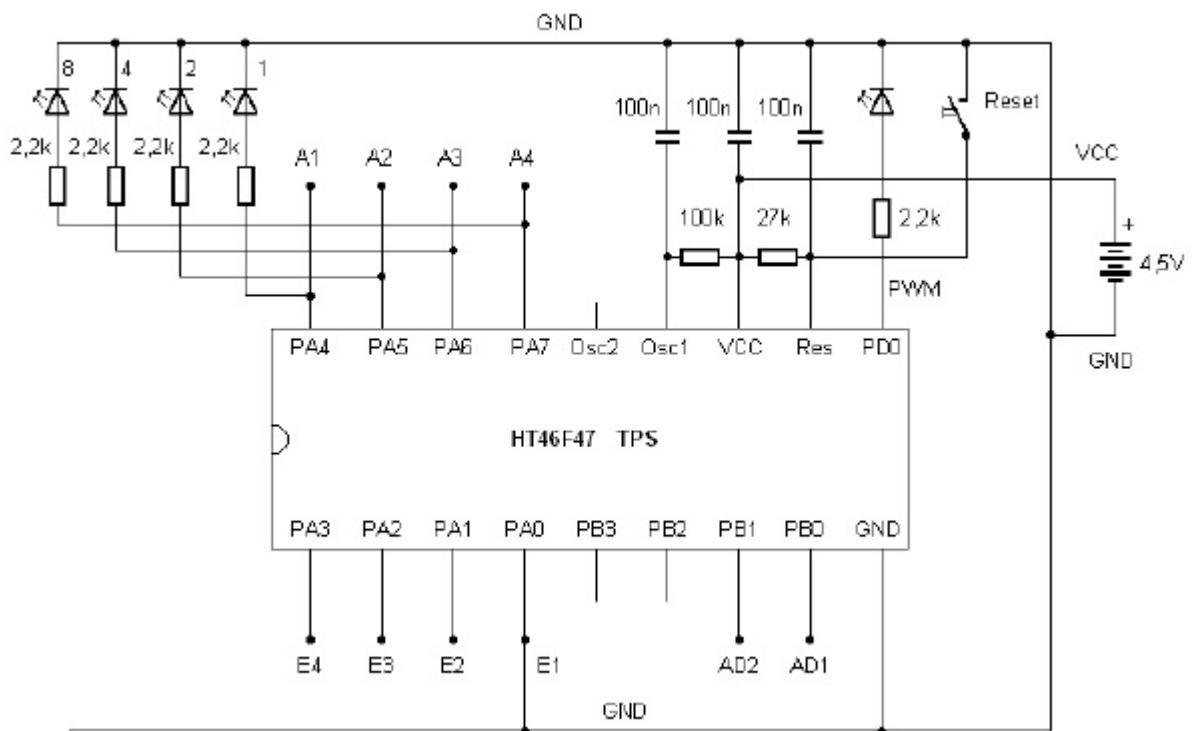
Izpis programa 1: Izmenična utripalka

Če ne dosežete zelenega rezultata, najprej preverite pravilno priključitev polov LED-diod. Koristna je tudi meritev nekaterih napetosti. Uporabite npr. digitalni multimeter v 20 V območju in pustite minus priključek na GND. Vse napetosti se s tem merijo proti GND:

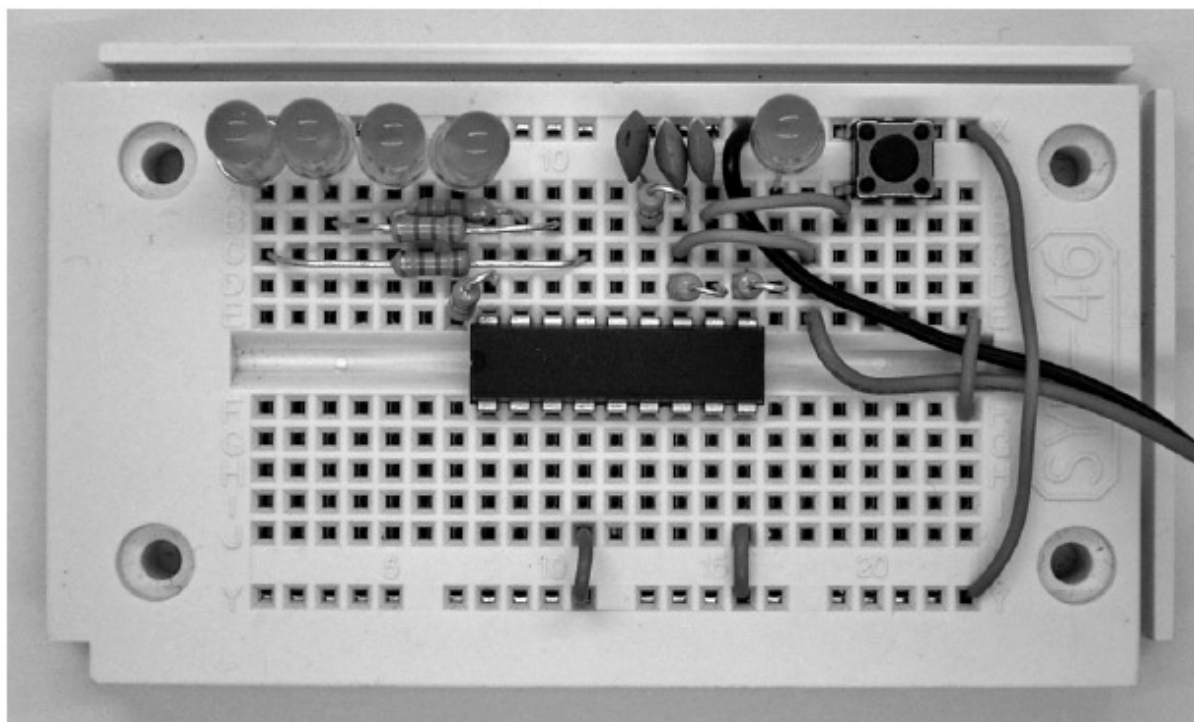
VCC: 4,5 V
 Reset: 4,5 V
 Osc1: 1,5 V
 E1 do E4: 4,5 V
 A1: izmenična
 A2, A3: 0 V
 A3: izmenična

Binarni števec in PWM izhodni signal

Vsi digitalni vhodi so z notranjim uporom vezani proti VCC (pull-up upor) in imajo mirovno napetost v višini obratovalne napetosti. Vendar lahko vsakega izmed vhodov priključite na GND z žico ali kontaktom. Pri zagonu standardni program prebere stanje vrat in jih analizira. Posamezne priključke lahko povežete z GND, tako da program tukaj prebere ničelno stanje. V odvisnosti od rezultata se priključijo različni programi.



Slika 5: Uporaba PWM LED-diode



Slika 6: Zagon binarnega števca

E1 priključite na GND. S tem se po ponastavitvi zažene drugi vzorčni program. Binarno prišteva stanja izhodov. Tekoče preleti stanja od 0000 (decimalno 0) do 1111 (decimalno 15). Program uporablja spremenljivko A za enostavno seštevanje in za izpisovanje na digitalne izhode ter na PWM-izhod. Ukaza 7 in 5 imata podfunkcije, ki se zapisujejo kot podatki.

Naslov	Ukaz	Podatki	Komentar
25	7	1	A = A + 1
26	5	4	Vrata = A
27	5	9	PWM = A
28	2	6	Čakaj 100 ms
29	3	4	Skoči -4

Izpis programa 2: Binarni števec z LED- in PWM-izhodom

Števni program lahko uporabljate kot vadbeni program za branje binarnih števil, ki jih morate obvladati za lastno programiranje. Vsaka izmed štirih LED-diod predstavlja en bit. Skupno je torej lahko prikazano 4-bitno število. LED-diode so v vezalnem načrtu označene v skladu s svojo vrednostjo z 8, 4, 2 in 1. S seštevanjem posameznih vrednosti dobite decimalno število. V heksadecimalnem zapisu so števila od 10 do 15 prikazana z velikimi tiskanimi črkami od A do F.

8	4	2	1	Decimalni	Heksadecimalni
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

8	4	2	1	Decimalni	Heksadecimalni
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

Program lahko uporabljate tudi kot utripalnik za različne frekvence.

Vsak naslednji višji izhod ima polovično frekvenco ali dvojno periodo:

A1: 200 ms

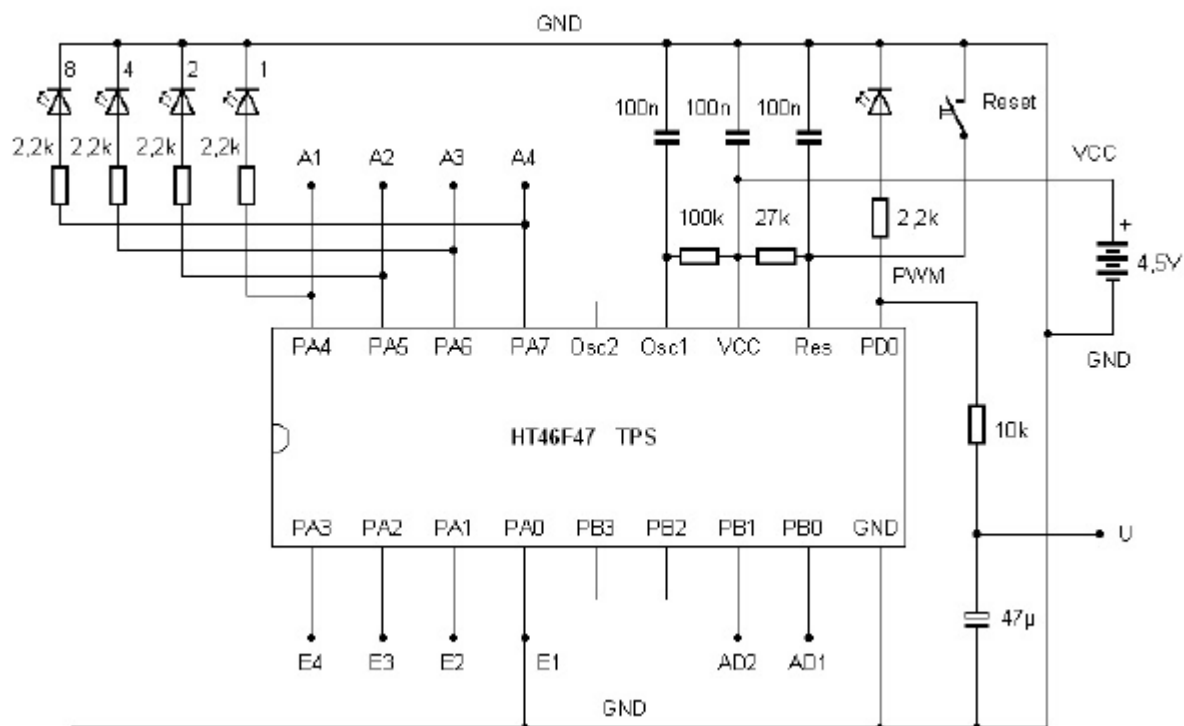
A2: 400 ms

A3: 800 ms

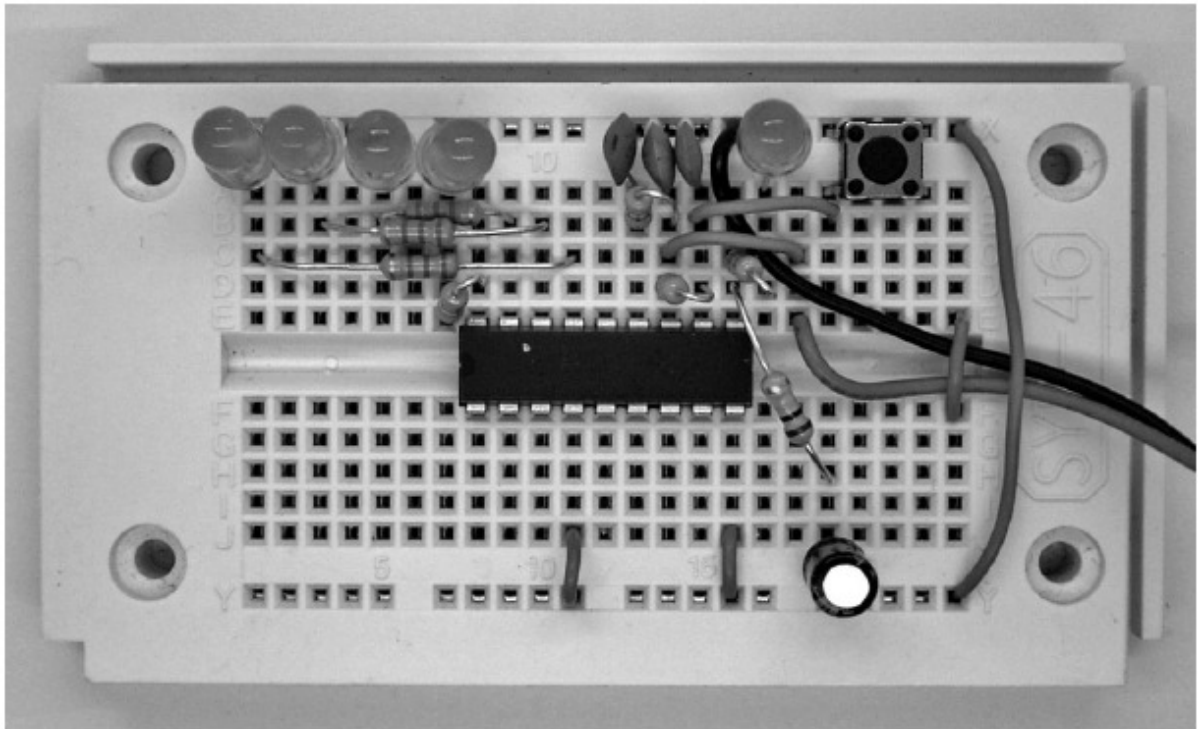
A4: 1.600 ms

Dodatno se naraščajoče številske vrednosti pošiljajo tudi na PWM-izhod (PWM = pulzno-širinska modulacija). PWM-signal je pravokotni signal s frekvenco pribl. 16 kHz. Pri tem se uravnava dolžina pulza, tako da razmerje med pulzom in premori določa povprečno trajanje vklopa in s tem svetilnost LED-diod. Svetilnost LED-diod, ki so tukaj priključene, se uravnava v 15 stopnjah med nič in polno svetilnostjo.

PWM-signal je možno izravnati na enosmerno napetost s pomočjo RC nizkoprepustnega filtra. PWM-izhod s tem postane analogni izhod. S tem programom boste dobili enosmerno napetost, ki postopoma narašča od 0 V do 4,5 V. Poteku napetosti lahko sledite s pomočjo merilnika ali osciloskopa.



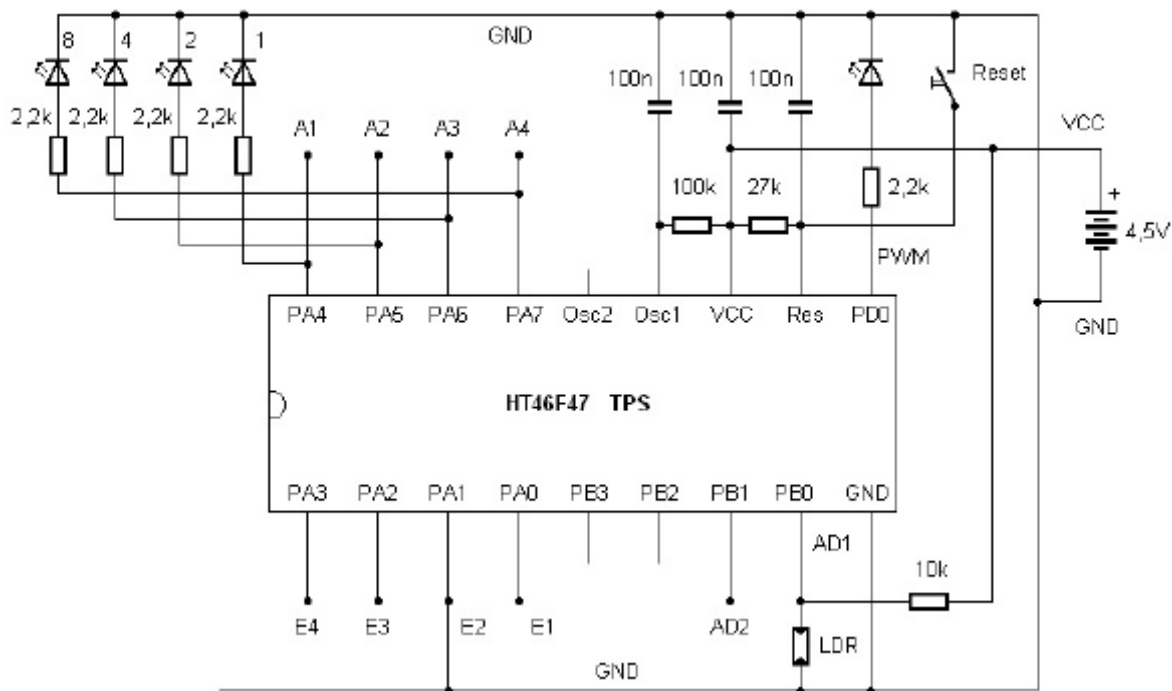
Slika 7: Nizkoprepustni filter na PWM-izhodu



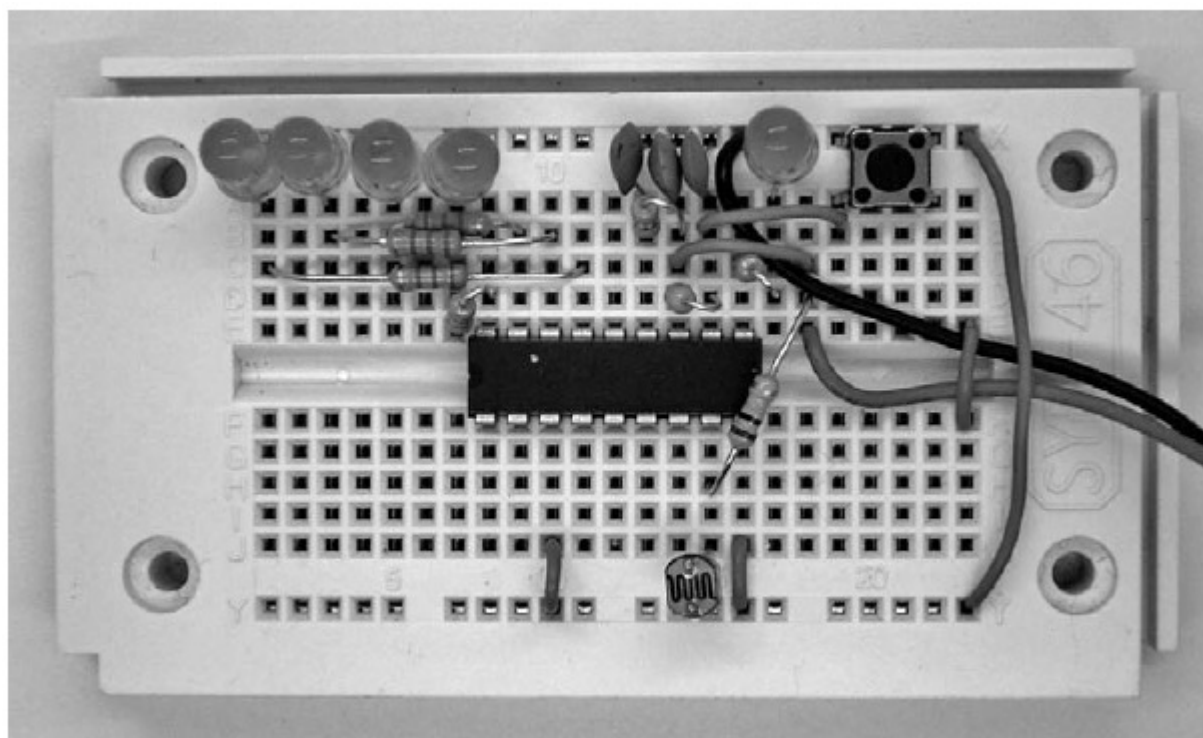
Slika 8: Izravnana PWM izhodna napetost

Analogno-digitalni pretvornik

S povezavo E2 proti GND in s pritiskom na ponastavitveno tipko boste zagnali majhen vzorčni program za analogno-digitalni pretvornik (AD-pretvornik). Pri tem se meri analogna napetost na analognem vhodu AD1 in se pretvori v digitalno številsko vrednost. Ker programirljiv krmilnik s tipkami dela izključno s 4-bitnimi vrednostmi, je rezultat analogno-digitalnega pretvarjanja število na območju od 0 do 15. Rezultat 0 predstavlja vhodno napetost 0, rezultat 15 pa napetost, ki ustreza obratovalni napetosti, torej npr. 4,5 V. AD-vrednost je prikazana na štirih LED-diodah v obliki binarnega števila, dodatno pa se posreduje tudi na PWM-izhod. Na analogni vhod AD1 priključite delilnik napetosti iz fiksnega upora in svetlobno občutljivega senzorskega upora (LDR).



Slika 9: Priključitev svetlobnega senzorja



Slika 10: Svetlobno občutljiv upor (LDR) na vhodu AD1

Vzorčni program je zaradi izhodnega signala, ki se pošilja na digitalne izhode in PWM-izhod, zelo podoben programu iz prejšnje točke. Vendar pa se v prvi vrstici nahaja ukaz za pretvorbo analogne vrednosti.

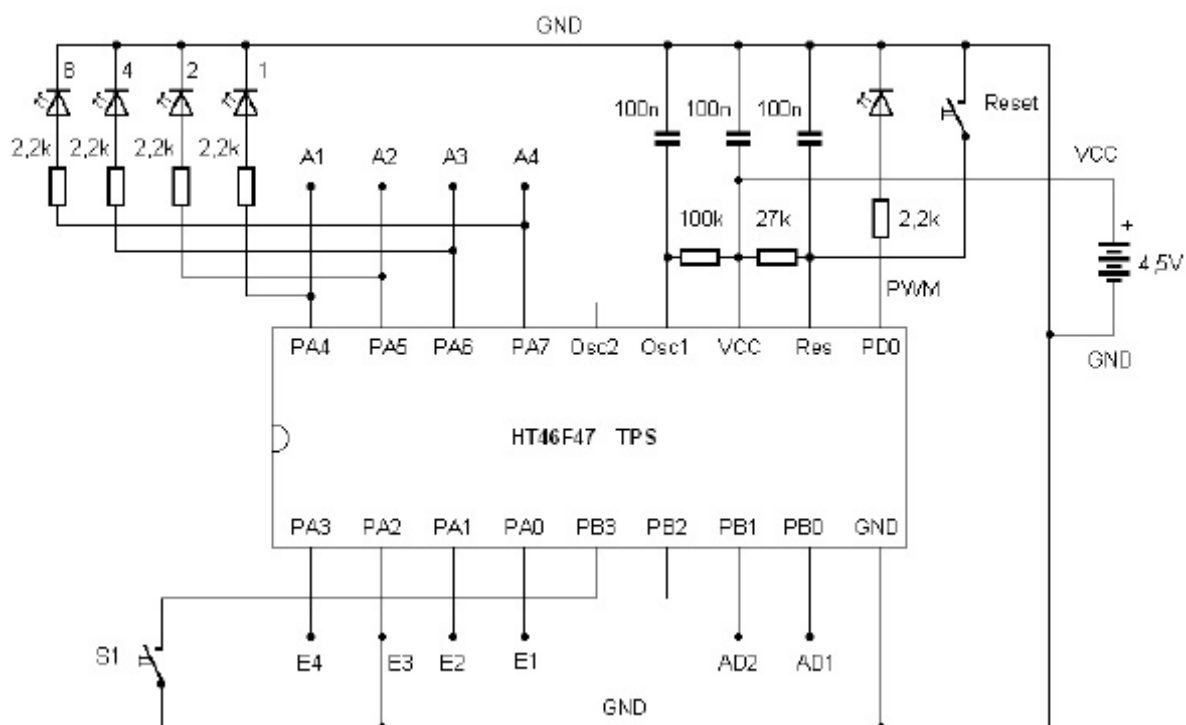
Naslov	Ukaz	Podatki	Komentar
2A	6	9	A = AD1
2B	5	4	Vrata = A
2C	5	9	PWM = A
2D	2	6	Čakaj 100 ms
2E	3	4	Skoči -4

Izpis programa 3: AD-pretvornik in PWM-izhod

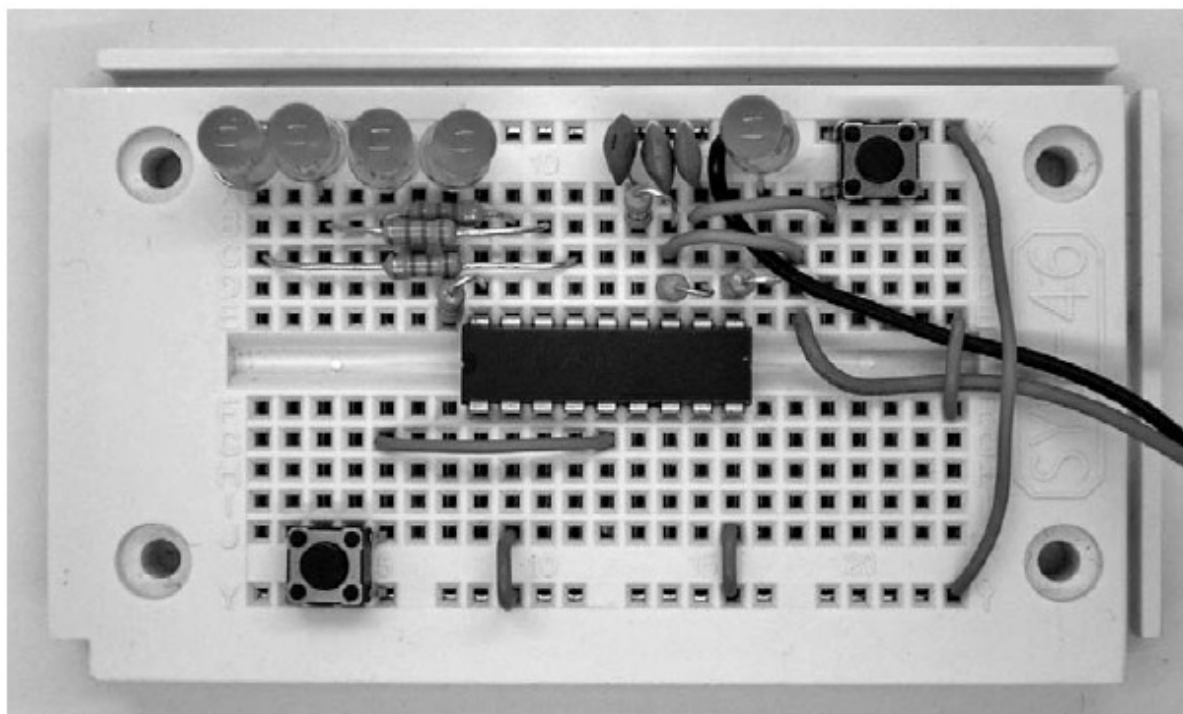
Testirajte program z različno osvetlitvijo senzorja. Več svetlobe kot pade na svetlobno občutljiv upor (LDR), manjša je napetost na vhodu AD1. Obratno pa v temi prihaja do maksimalnih AD-vrednosti in s tem do maksimalne svetilnosti LED-diod na PWM-izhodu. Binarna števila odčitajte z LED-prikazovalnika in poskušajte npr. nastaviti svetilnost natanko na polovici območja. Digitalna vrednost nato znaša 0111 ali 1000. Pri malce brleči umetni svetlobi se lahko zgodi, da rezultat preklaplja med dvema stopnjama.

Naključni generator

Z žičnim mostičkom E3 proti GND boste zagnali vzorčni program za naključni generator. Tukaj se analizira tipka S1. Ustrezni vhod ima notranji pull-up upor, ki veže napetost na VCC-nivo. Tipka je priključena proti masi. En pritisk tipke premakne vhod S1 na ničlo.



Slika 11: Zagon naključnega generatorja



Slika 12: Žični mostiček med E3 in GND

Program uporablja pogojni stavek goto. Če je stanje vhoda S1 ena, potem se naslednji ukaz preskoči. Če pritisnete tipko, je stanje nič, s tem pa se izvede povečanje spremenljivk A. To vodi v hitrejšo prištevanje stanja izhoda. Ko tipko izpustite, se zadnje stanje števca ohrani. Zaradi visoke hitrosti štetja nimate vpliva na rezultat, ki je torej naključen.

Naslov	Ukaz	Podatki	Komentar
30	5	4	Vrata = A
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Skoči -3

Izpis programa 4: Naključni generator

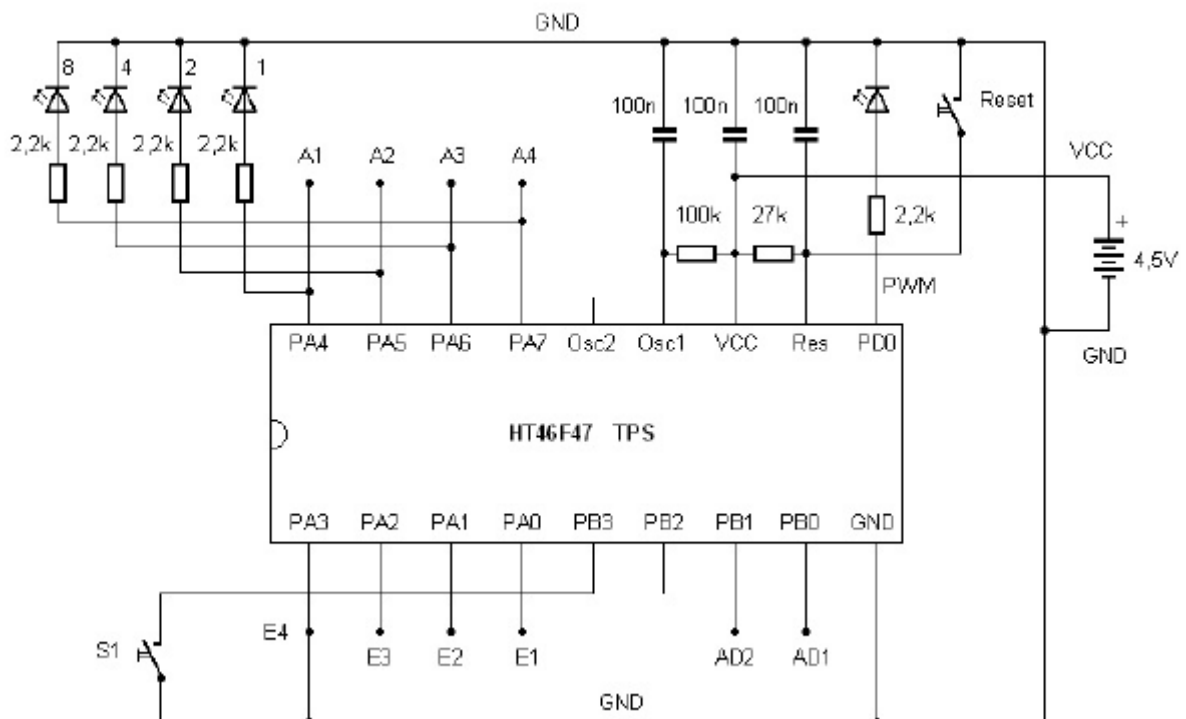
Za novi naključni rezultat vsakič kratko pritisnite tipko. Testirajte naključno funkcijo, tako da sestavite statistiko rezultatov. Po zadostnem številu prehodov bi se morale pokazati, da se vsi rezultati približno enako pogosto ponavljajo. Program je primeren tudi kot igra, pri katerem se je treba potegovati npr. za število 1111.

Hkrati je program tudi števec z maksimalno možno delovno hitrostjo, saj se ne uporablja noben čakalni ukaz. Iz tega razloga lahko na tem primeru raziščete delovno hitrost programirljivega krmilnika s tipkami. Dokler je tipka pritisnjena, se na izhodu A1 pojavi pravokotni signal s frekvenco pribl. 133 Hz in periodo 7,5 ms. Vrata torej čez vsakih 3,75 ms spremenijo svoje stanje. Program v for zanki preide skozi štiri ukaze. Za posamezni ukaz torej potrebuje pribl. eno milisekundo. Zadnji izhod A4 prikazuje 16,6 Hz frekvenco, kar lahko prepoznate kot vidno brljenje.

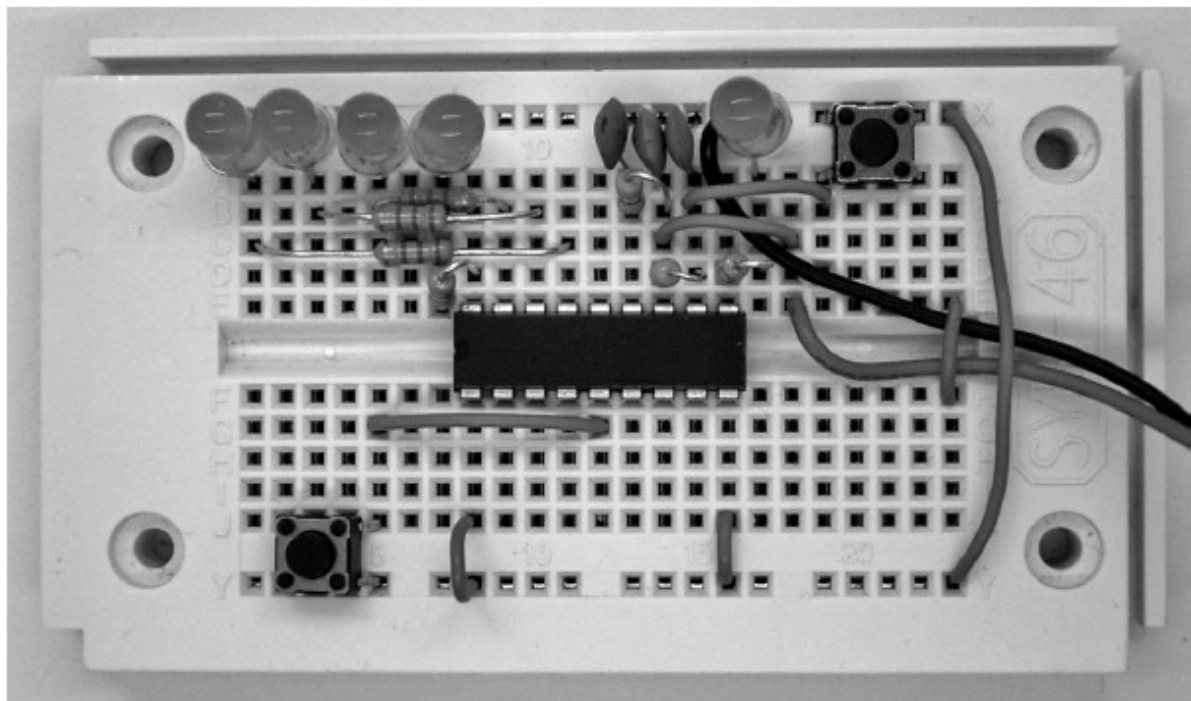
Če so kdaj za kritične naloge potrebne višje delovne hitrosti, potem lahko povečate taktno hitrost krmilnika, tako da zmanjšate upor na Osc1. S 100 kΩ imate na voljo hitrost takta 2 MHz. Upor nadomestite z drugim uporom s 27 kΩ. S tem boste dosegli skoraj štirikrat višjo hitrost takta in čas ukaza pribl. 0,25 ms. Vendar pa naj v običajnem načinu krmilnik deluje s 100 kΩ na Osc1. S tem je zagotovljena nizka poraba toka in varno delo tudi pri majhni obratovalni napetosti vse do spodnje meje 2,2 V.

Merjenje dolžine impulza

Z E4 na GND se po ponastavitvi zažene vzorčni program za merjenje dolžine impulza. Tudi tukaj se ponovno analizira stanje na vhodu S1.



Slika 13: E4 na GND



Slika 14: Začetek meritve dolžine impulza

Meritev časa teče v stanju $S1 = 0$, torej ko je tipka pritisnjena. K čakalnemu času 5 ms se še enkrat prišteje pribl. 5 ms za izvedbo skupno petih ukazov v for zanki. Časovna enota meritve je torej 10 ms.

Naslov	Ukaz	Podatki	Komentar
34	2	2	Čakaj 5 ms
35	C	C	S1 = 0?
36	3	2	Skoči -2
37	4	0	A = 0
38	2	2	Čakaj 5 ms
39	7	1	A = A + 1
3A	5	4	Vrata = A
3B	C	E	S1 = 1?
3C	3	4	Skoči -4
3D	3	9	Skoči -9

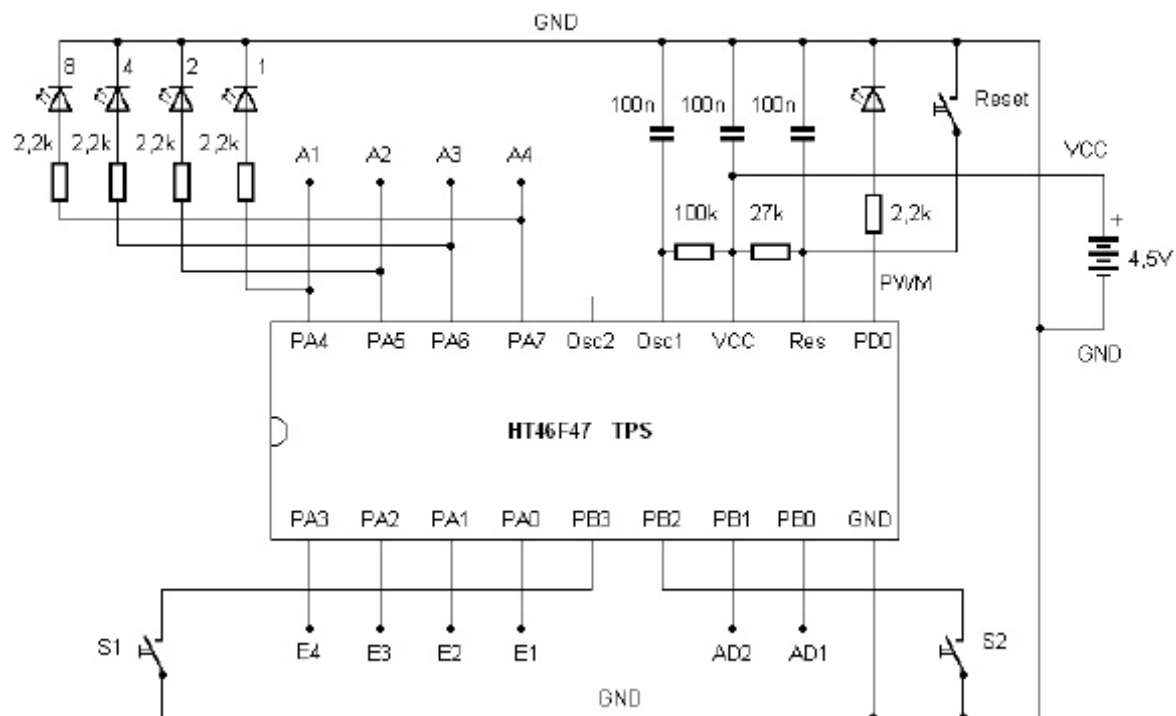
Izpis programa 5: Meritev časa

Karseda kratko pritisnete tipko S1. Rezultat 1010 bo npr. prikazan kot decimalno število 10. Ker znaša časovna enota programa 10 ms, potem ta prikaz pomeni 100 ms. Z nekaj vaje lahko dosežete tudi krajše čase vse do spodnje meje 50 ms.

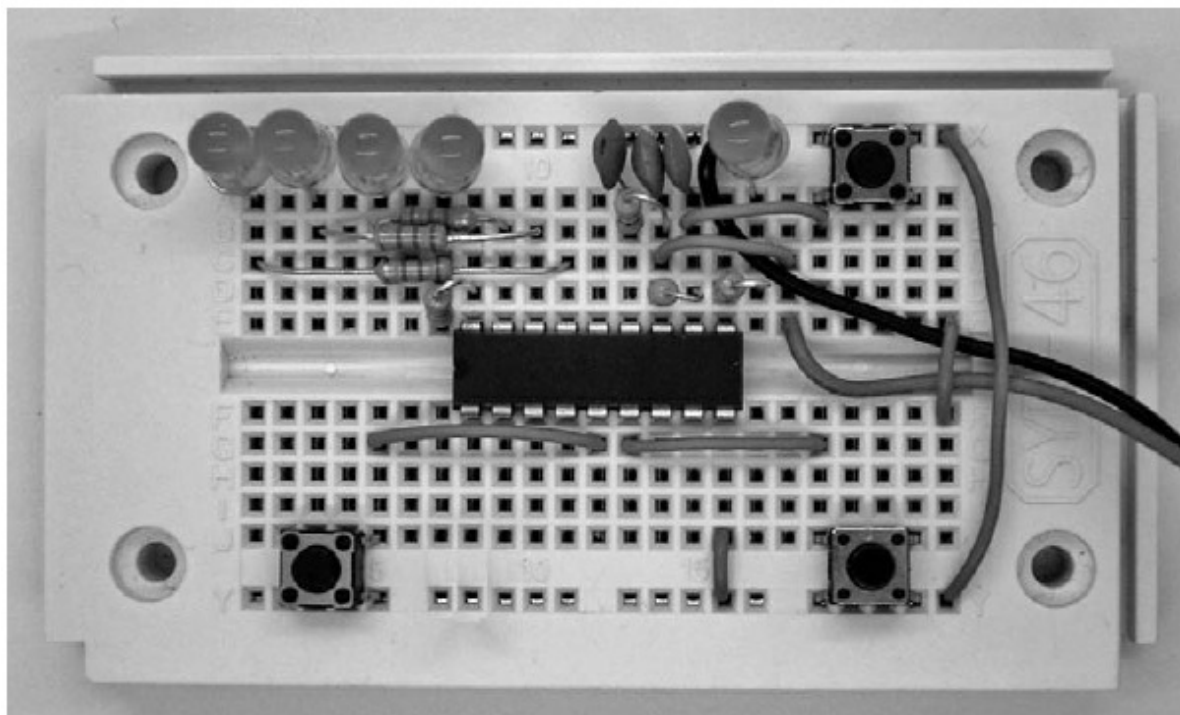
Branje programov

Za programiranje potrebujete tipko S1 (vnos podatkov, levo) in tipko S2 (programiranje, desno). Poleg tega potrebujete tudi ponastavitveno tipko. Že samo s temi tipkami lahko berete programe in tudi vnašate poljubne programe. Z nekaj vaje lahko že v zelo kratkem času vnašate programe in spreminjate obstoječe programe.

V načinu programiranja izvedete ponastavitev, tako da pritisnete programirno tipko S2, pri čemer lahko tipko S2 izpustite šele čez pribl. pol sekunde po ponastavitvi. Nato lahko s samo tipko S2 listate po obstoječem programu in si ogledate ukaze ter podatke. Pri tem morate za vsak naslov dvakrat pritisniti tipko S2. Tako preklapljate med prikazom ukaza in podatkov. Poleg tega se vsakič za kratek čas prikaže aktualni naslov.



Slika 15: S1 in S2 za način programiranja



Slika 16: Tri tipkala in LED-prikaz

- Prvi pritisk tipke S2
- Prikaz naslova (pod štirimi bitni), 300 ms
- Izklop prikaza, 300 ms
- Prikaz ukaza
- Drugi pritisk tipke S2
- Prikaz podatkov
- Tretji pritisk tipke S2
- Prikaz naslednjega naslova, 300 ms
- ltd.

Če si želite npr. samo ogledati obstoječi program s petimi koraki, vendar ga ne želite spreminjati, potem s skupno desetimi pritiski tipke S2 pridete do konca programa. Ker se vsakič za kratek čas prikaže aktualni naslov, je orientacija enostavna. Tako vedno veste, če prikaz trenutno prikazuje ukaz ali podatke. V stanju tovarniških nastavitvev se v prvih petih naslovih nahajajo naslednji ukazi (prikazani v spodnji tabeli). Pri tem gre za začetek izbirnega programa za zagon posameznih vzorčnih programov.

Naslov	Ukaz	Podatki	Komentar
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?

Izpis programa 6: Programska koda v osnovnem stanju

4-bitni ukaz in pripadajoči 4-bitni podatki skupaj tvorijo en bajt, torej 8-bitno število. Polovici bajta pravimo tudi polbajt oz. nibble. Zgornji polbajt vsakič tvori ukaz, spodnji polbajt pa pripadajoče podatke. EEPROM krmilnika skupno zajema 128 bajtov. Tako lahko ima program največ 128 ukazov. To za večino aplikacij zadostuje, saj je programska koda izjemno kompaktna. Za številne koristne programe zadostuje manj kot deset ukazov.

Prikažite posamezne ukaze in podatke ter primerjajte vsebino pomnilnika. Nato ponovno pritisnite ponastavitveno tipko. Stari program se nespremenjen zažene.

Vnašanje programov

Tipko S1 uporabljate samo takrat, ko želite spremeniti ali na novo vnesti ukaz ali njegove podatke. Načeloma lahko vnašate samo številske vrednosti med 0 in 15. S prvim pritiskom tipke S1 nastavite 0. Z vsakim naslednjim pritiskom tipke povečate število za 1. Aktualno stanje je vedno binarno prikazano s štirimi LED-diodami. Če želite vnesti npr. 4, potem je treba skupno petkrat pritisniti tipko S1: 0, 1, 2, 3, 4. Binarni prikaz nato kaže 0100.

Če ste na ta način na novo vnesli ali ukaz ali podatke ali oboje, potem z drugim pritiskom tipke S2 ta bajt programirate v EEPROM. Za ponazoritev programiranja se LED-prikaz za 600 ms izključi, nato pa se pojavi prikaz naslednjega naslova, kateremu sledi prikaz naslednjega ukaza. Ta majhen premor je treba intuitivno razumeti kot postopek programiranja. To si lahko predstavljate tako, kot da sistem prihrani energijo za prikaz in jo uporabi za programiranje EEPROM-pomnilnika. Nekaj podobnega poznamo tudi pri avtomobilu: Ko zaženete zaganjalnik, se za kratek čas izključijo luči in radio.

Že obstoječi program lahko spremenite samo na enem mestu. S tipko S2 se premaknete na želeno mesto in s tipko S1 spremenite ukaz ali podatke. Nato spremembe shranite s pritiskom tipke S2.

Za prvo testiranje se vnese program s samo dvema ukazoma. Vključi tri LED-diode, nato pa postopek ponavlja v neskončni zanki.

Naslov	Ukaz	Podatki	Komentar
00	1	7	A1-4 = 0111
01	3	0	Skoči 0

Izpis programa 7: Vkllop LED-diod

Namesto izčrpnega izpisa programa lahko izberete tudi kratek zapis. Pri tem se oba bajta združita v heksadecimalna števila: 17h, 30h. V nadaljevanju se načeloma uporablja heksadecimalno zapisovanje. Programi se zato pišejo v kratkem zapisu brez znakov "h":
17 30

Za vnos je treba vtipkati naslednje:

S2 + ponastavitvena tipka
2 x S1
S2
8 x S1
S2
4 x S1
S2
1 x S1
S2

Če pomotoma prevečkrat pritisnete tipko D1, lahko kljub temu dosežete pravilno število. Preprosto morate še enkrat iti čez 15, nato pa se števila ponovno začnejo pri vrednosti 0.

Po popolnem vnosu novi program zaženete s pritiskom ponastavitvene tipke. Vidite, da se vključijo tri LED-diode. Nato se ne zgodi nič. Krmilnik se sedaj tudi več ne odziva na stanja

na vhodih od E1 do E4, saj ste delno prepisali izvorni program. Tako tudi več ne morete zagnati vzorčnih programov.

Ker ste spremenili samo prva dva naslova pomnilnika, lahko enostavno spet zaženete izvorni program. To storite tako, da samo vnesete prva dva ukaza (64 51) v skladu z izpisom programa iz zadnje točke.

Testirajte izvorno delovanje vzorčnih programov. Najbolje je, da še enkrat vnesete novi vadbeni program. Čez kratek čas boste pri vnašanju programov že bolj prepričani vase.

Obnovitev vzorčnih programov

Če želite čez nekaj časa krmilnik povrniti v izhodiščno stanje, lahko to storite z vnosom dveh bajtov FF. To dejansko ustreza stanju praznega EEPROM-pomnilnika. Strojno-programaska oprema programirljivega pomnilnika s tipkami vsebuje zagonsko funkcijo, ki najprej preveri prva dva naslova, da lahko zazna prazen pomnilnik. Če tukaj prebere dva bajta FF, potem krmilnik sklepa, da še ni bil vnesen noben program. V tem primeru se EEPROM samodejno napolni z vzorčnimi programi. Ta funkcija je dejansko namenjena temu, da se EEPROM-pomnilnik pri prvem zagonu krmilnika napolni z vzorčnimi programi. Lahko pa jo tudi kadarkoli uporabite za povrnitev osnovnega stanja.

Naslov	Ukaz	Podatki	Komentar
00	F	F	-
01	F	F	-

Izpis programa 8: Vrnitev na osnovno stanje

Zaženite način programiranja, tako da pritisnete in držite tipko S2 ter hkrati pritisnete ponastavitveno tipko. Nato skupno štirikrat vnesite vrednost F (decimalno 5), pri kateri so vse LED-diode od A1 do A4 vključene. Tudi zadnji vnos zaključite s pritiskom tipke S2.

Nato pritisnite ponastavitveno tipko. Za novo programiranje vseh bajtov vzorčnih programov krmilnik potrebuje zgolj malce dlje kot običajno. S tem je ponovno vzpostavljeno izhodiščno stanje. Testirajte npr. izmenično utripalko s strani 8 brez žičnega mostička na vhodih.

Osnovni ukazi za programirljiv krmilnik s tipkami

Programirljiv krmilnik s tipkami pozna skupno 14 ukazov (1–14). K številnim ukazom sodi parameter v obliki 4-bitnega števila od 0000 do 1111 (0–F), torej s številskim območjem do 15 (decimalno). Drugi ukazi poznajo podfunkcije, ki so navedene v obliki parametra. Za ukazno kodo se lahko zato skriva do 16 podukazov. Tako npr. ukaz 7 predstavlja »Računaj A = ...«. Parameter označuje, katera računaska funkcija se mora izvesti.

V nadaljevanju so ukazi in podatki pisani skupaj kot en bajt v heksadecimalnem zapisu. Iz ukaza 1 skupaj s parametrom 4 se sestavi ukaz 14h. Znak "h" se izpusti, saj so vsi ukazi in naslovi načeloma zapisani v obliki heksadecimalnega zapisa.

Prvi trije ukazi so naslednji:

10–1F: neposredni izhodni signal vrat na A1–A4, 0–15, binarno od 0000 do 1111

20–2F: čakalni čas od 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: skok nazaj od 0–15

Ukaz 1 služi za izhodni signal iz vrat v obliki konstantnega števila. S tem se lahko na izhodu pojavljajo poljubni bitni vzorci, lahko pa tudi npr. vklopljate več LED-diod hkrati.

Čakalni ukaz 2 uporablja parameter, ki vključuje čas v milisekundah in v stopnjah 1-2-5. Kljub majhnemu obsegu števil od 0 do 15 lahko na ta način izvedete čase zakasnitve med eno milisekundo in eno minuto. Še daljše čase bi morali programirati z večkratno izvedbo čakalnega ukaza, npr. v for zanki.

Ukaz za vrnitev 3 je posebej enostaven in zadostuje za številne naloge, pri katerih se mora postopek neskončno ponavljati. Dolžina skoka je omejena na območje do 15. Ker dolžina skoka velja relativno glede na aktualni naslov, lahko s to vrnitvijo dele programa poljubno premaknete na druge naslove.

Program izmenične utripalke ima naslednje tri ukaze. Tukaj ga je treba rahlo spremenjenega zapisati v naslovno območje od 00 naprej. Spremenili so se tudi izhodni bitni vzorci in čakalni časi.

Naslov	Ukaz	Podatki	Komentar
00	1	1	A1–4 = 0001
01	2	7	Čakaj 200 ms
02	1	4	A1–4 = 0100
03	2	7	Čakaj 200 ms
04	3	4	Skoči –4

Izpis programa 9: Program utripalke

V kratkem heksadecimalnem zapisu je program sedaj videti tako:

11 27 14 27 34

Na podlagi teh prvih treh ukazov lahko že napišete številne enostavne programe. Analizirajte in testirajte naslednje tri programe. Cilj tega početja je, da lahko te ukaze kasneje intuitivno uporabljate. Enostavne poteke programov kot je ta lahko z nekaj vaje programirate kar na pamet in jih neposredno vnašate. Primer za to je enostavna tekoča luč s štirimi izhodnimi vzorci:

Naslov	Ukaz	Podatki	Komentar
00	1	1	LED-diode 0001
01	2	8	Čakaj 500 ms
02	1	2	LED-diode 0010
03	2	8	Čakaj 500 ms
04	1	4	LED-diode 0100
05	2	8	Čakaj 500 ms
06	1	8	LED-diode 1000
07	2	8	Čakaj 500 ms
08	3	8	Skoči –8

11 28 12 28 14 28 18 28 38

Izpis programa 10: Tekoča luč 1

Program razširite za dva dodatna izhodna vzorca, tako da svetlobna točka vedno teče sem in tja. Eksperimentirajte tudi z drugimi izhodnimi vzorci in časi zakasnitve.

Naslov	Ukaz	Podatki	Komentar
00	1	1	LED-diode 0001
01	2	8	Čakaj 500 ms
02	1	2	LED-diode 0010
03	2	8	Čakaj 500 ms
04	1	4	LED-diode 0100
05	2	8	Čakaj 500 ms
06	1	8	LED-diode 1000
07	2	8	Čakaj 500 ms
08	1	4	LED-diode 0100
09	2	8	Čakaj 500 ms
0A	1	2	LED-diode 0010
0B	2	8	Čakaj 500 ms
0C	3	C	Skoči -12

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

Izpis programa 11: Tekoča luč 2, sem in tja

Časovno stikalo lahko s čakalnim ukazom vsebuje zakasnitev do ene minute. Na koncu se nahaja vrnitev (povratni skok) z dolžino skoka 0, torej neskončna zanka brez vsebine, ki deluje kot konec programa. Novi zagon sprožite s pritiskom ponastavitvene tipke. Program lahko tudi nadgradite na 3-minutno kuhinjsko uro. Pri tem lahko preostali čas prikazujete s številom vključenih LED-diod.

Naslov	Ukaz	Podatki	Komentar
00	1	F	LED-diode 1111
01	2	F	Čakaj 1 min
02	1	0	LED-diode 0000
03	3	0	Konec

1F 2F 10 30

Izpis programa 12: Časovno stikalo za eno minuto

Računanje s spremenljivkami

Do sedaj so bile pri parametrih posameznih ukazov uporabljene samo konstantne številčne vrednosti. To je smiselno pri zmeraj istem poteku programa. Nasprotno pa kompleksnejši programi delujejo s spremenljivimi podatki. Možno je izvesti izračun, npr. $A = A + B$. Zaradi vsebine spremenljivk A in B bo rezultat zmeraj drugačen. Rezultat bi lahko npr. usmerjal LED-diode na izhodih.

Upravljanje ima na razpolago spremenljivke A, B, C in D. Najpomembnejša spremenljivka je A. Imenujemo jo akumulator. A sodeluje pri vseh računskih operacijah in prejme rezultat izračuna. Poleg tega se A uporablja tudi za prenos podatkov. B je potrebna predvsem pri računskih operacijah. C in D lahko služita kot medpomnilnik in se kasneje uporabljata tudi kot števec za for zanko.

Poleg tega obstajata tudi dva analogna vhoda (AD1 in AD2) in en PWM-izhod. Obdelani podatki so omejeni na štiri bite in dostopni le preko spremenljivke A ($A=AD1$, $PWM=A$). Akumulator je možno napolniti s številom tudi neposredno (ukazi 40-4F). Za polnjenje B, C ali D je potrebno naprej napolniti A in potem dodeliti vsebino ostalim spremenljivkam (ukazi 51-53). Z A in B je možno izpeljati nekaj računskih korakov (ukazi 71-7A).

Ukazi 40-4F dodelijo A novo vrednost. Skupina ukazov 51-5A prenese vsebino od A do določenega cilja, npr. do druge spremenljivke ali do PWM-izhoda. V tej skupini so tudi ukazi, ki določajo posamezni bit izhodiščnih vrat.

Ravno obratna smer podatkov je pri skupini ukazov 61–6A, kjer se podatki enega vira vnašajo v A. Skupina ukazov 71-7A izvaja namreč računske operacije, pri čemer se rezultat načeloma pojavi v A. Izhodiščna vrata Dout obsegajo štiri izhode od A1 do A4, skupaj ali pa kot posamezni biti Dout. Vhodi od E1 do E4 se odzivajo na enak način kot vhodna vrata Din.

40–4F: $A = 0-15$

51–5A: cilj 1–9 = A

51: $B = A$

52: $C = A$

53: $D = A$

54: $Dout = A$

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: $A = \text{Vir } 1-10$

61: $A = B$

62: $A = C$

63: $A = D$

64: $A = \text{Din}$

65: $A = \text{Din}.0$

66: $A = \text{Din}.1$

67: $A = \text{Din}.2$

68: $A = \text{Din}.3$

69: $A = AD1$

6A: $A = AD2$

71 –7A: $A = \text{Izhod } 1-10$

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: $A = A \text{ And } B$

78: $A = A \text{ Or } B$

79: $A = A \text{ Xor } B$

7A: $A = \text{Not } A$

Primer uporabe spremenljivke A najdete med vzorčnimi programi v 3. poglavju. Program je bil tukaj nastavljen na naslov nič in nekoliko razširjen. Dodan je bil tudi definiran pričetek z vrednostjo 0 v spremenljivki A. V naslovu 01 je možno najti računski ukaz, tukaj gre za zvišanje za 1. Vsebina spremenljivke A se zatem prenese na PWM-izhod in na izhodna vrata.

Naslov	Ukaz	Podatki	Komentar
00	4	0	A 0
01	7	1	A = A + 1
02	5	4	Vrata = A
03	5	9	PMW = A
04	2	6	Čakaj 100 ms
05	3	4	Skoči -4

40 71 54 59 26 34

Izpis programa 13: Zvišanje za 1

Dodatni primer je že bil prikazan v 4. poglavju. Podatki pri tem pridejo od analognega vhoda AD1 in se prenesejo na izhodna vrata in na PWM-izhod. Spremenjen program vsebuje še en dodatni računski korak, in sicer invertiranje vsebine spremenljivke A. Pri tem iz vrednosti 0000 nastane nova vrednost 1111, t.p. iz 0 nastane 15 in obratno. Zaradi višje vhodne napetosti je manj PWM-tiskanja.

Naslov	Ukaz	Podatki	Komentar
00	6	9	A = AD1
01	5	4	Vrata = A
02	7	A	A = Not A
03	5	9	PWM = A
04	2	6	Čakaj 100 ms
05	3	5	Skoči -5

69 54 7A 59 26 35

Izpis programa 14: Invertiranje

Skoki in razcepi

Do sedaj je obstajal le en enostaven povratni skok (ukaz 3), ki je segal maksimalno 15 naslovov nazaj. Temu se je sedaj pridružil absolutni skok (Jump). Ker je cilj skoka možno navesti le s štirimi biti, obstaja dodatni ukaz, ki določa zgornji polbajt naslova. S tem se dobi naslovni prostor 0-255. To je več, kot je potrebno, saj EEPROM krmilnika zajema samo 128 bajtov, se pravi področje od 00 do 7F (decimalno od 0 do 127). Pomnilnik je s tem dejansko razdeljen na osem strani, od strani 0 do strani 7 (Page 0–7). Pred absolutnim skokom je potrebno določiti stran cilja skoka.

Dve for zanki s spremenljivkama C in D izvajata prav tako absolutne skoke, pri čemer pa je tudi prej potrebno določiti stran naslova.

Pogojni skoki delujejo kot skip ukazi (izpust, preskok). Če je vsak pogoj resničen, se en naslov preskoči. Tam bi se lahko nahajal, npr. ukaz za skok ali tudi računski ukaz. Kot pogoji so na voljo primerjave med A in B, kot tudi neposredno bit poizvedovanje vhodnih vrat.

Poleg tega obstaja tudi priklic podprogramov (priklic) in pripadajoč ukaz za povratni skok (ponastavitvena tipka). Dovoljenih je sicer več podprogramov, ampak iz podprograma ni dovoljeno priklicati nobenega drugega podprograma, saj si prevajalnik zmeraj zapomni le en naslov povratnega skoka.

80–8F: Adr-high = 0–15

90–0F: Neposredni skok (Jump) na Adr-high, Adr-low (0–15)

A0–AF: For zanka C-krat Adr-high, Adr-low (0–15)

B0–BF: For zanka D-krat Adr-high, Adr-low (0–15)

C1–CF: pogojni skok: ko (pogoj 1-15) potem preskok

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 Din0,1 = 1 then Adr = Adr + 1

C6: if Din0,2 = 1 then Adr = Adr + 1

C7: if Din0,3 = 1 then Adr = Adr + 1

C8: if Din.0 = 0 then Adr = Adr + 1

C9: if Din0,1 = 0 then Adr = Adr + 1

CA: if Din0,2 = 0 then Adr = Adr + 1

CB: if Din0,3 = 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: Priklic podprogramov (priklic) Adr-high, Adr-low (0-15)

E0–EF: Povratni skok na podprogram (ponastavitvena tipka)

Primer uporabe pogojnih skočnih ukazov je možno najti v vzorčnem programu v 6. poglavju. Nekoliko spremenjen je bil postavljen na naslov 0. Ker je zgornji del naslova (Adr-hi) v stanju mirovanja, se torej krmilnik prične na strani 0, tukaj ni potrebno uporabljati ukaza 80. Ponovno se izmeri in prikaže trajanje pritiska tipke. Iz programa so bili odstranjeni vsi ukazi za čakanje, tako da sedaj deluje z višjo časovno ločljivostjo.

Naslov	Ukaz	Podatki	Komentar
00	C	C	S1 = 0?
01	3	1	Skoči -1
02	4	0	A = 0
03	7	1	A = A + 1
04	5	4	Vrata = A
05	C	E	S1 = 1?
06	3	3	Skoči -3
07	3	7	Skoči -7

CC 31 40 71 54 CE 33 37

Izpis programa 15: Odziv na tipko S1

Skočni ukaz CC na naslovu 00 oceni stanje na tipkalu S1. V stanju mirovanja je S1 = 1. Pogoj torej ni resničen in ukaz na naslovu 01 se ne preskoči. Tam se nahaja relativni skočni ukaz na začetek. Program tako dolgo ponavlja ukaze na naslovih 00 in 01, dokler ne pritisnete na tipko. Potem je pogoj resničen in naslov 01 se preskoči. S tem se prične dejanski postopek merjenja. Akumulator se izbriše, potem se vedno za 1 zviša in se posreduje do LED-diodo. Na naslovu 05 se nahaja dodatni pogojni ukaz za skok CE. Tukaj je pogoj za preskok enega ukaza S1 = 1. Ker je tipka na začetku še pritisnjena, pogoj ni resničen. Ukaz 06 se izvede in vodi v povratni skok na 03. Šele za tem, ko je tipka izpuščena, pride program na naslov 07 in do povratnega skoka na začetek.

Vstavite program in ga testirajte. Reakcijski čas je sedaj bistveno krajši. Časovna enota znaša pribl. 5 ms.

Prvotni vzorčni program se še nahaja v pomnilniku od naslova 34h naprej, saj se prepisejo le spodnji naslovi. Napišite kratek program, ki vsebuje le en skok na ta naslov. Tukaj je najprej potrebno določiti stran 3. Saj potem absolutni skok s podanim naslovom 4, ki sledi, cilja na dejanski naslov 34.

Naslov	Ukaz	Podatki	Komentar
00	8	3	Stran 3
01	9	4	Naslov = 34

83 94

Izpis programa 16: Absolutni skok na program časomera

Prvotni vzorčni program se tako ponovno priključuje. To testirajte tudi za druge primere. Celoten pregled vseh uporabljenih programov najdete v prilogi.

Pregled ukazov

Vsi ukazi na enem mestu – to delo s krmilnikom poenostavi. Naslednja tabela vsebuje vse ukaze v strnjeni obliki.

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port =	Wait	Jump -	A =	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	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	
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 = C 0	C	
D	13	20 s	13	13					D	D	D	S2 = D 0	D	

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
E	14	30 s	14	14					E	E	E	S1 = 1	E	
F	15	60 s	15	15					F	F	F	S2 = 1	F	

For zanke

En postopek bi se naj, npr. izpeljal natanko petkrat. Za to je potrebno sestaviti for zanko. Ukaz za skok se v tem primeru izpelje petkrat. Zatem pa več ne. Številčna spremenljivka se imenuje C. Vrednost števca 5 je potrebno najprej posredovati v A in od tam v C. Ukaz A2 izpelje absolutni skok na 02 in hkrati zmanjša vsebino spremenljivke C za 1. Ko C doseže vrednost, se skok več ne izvede. Absolutni naslov za skok se nanaša na podane strani. Pri programu na strani 0 je stranski ukaz 80 tudi dovoljeno izpustiti. Je pa ta ukaz nujno potreben pri skoku na drugo stran.

Naslov	Ukaz	Podatki	Komentar
00	4	5	A = 5
01	5	2	C = A
02	1	5	Vrata = 0101
03	2	8	500 ms
04	1	A	Vrata = 1010
05	2	8	500 ms
06	8	0	Stran 0
07	A	2	C-krat 02
08	3	0	Konec

45 52 15 28 1A 28 80 A2 30

Izpis programa 17: For zanka

Testirajte program. LED-diode pri vsaki prevodnosti prikazujejo vzorec 0101 in 1010. Vsekakor ta del programa očitno ne bo izveden petkrat, ampak natanko šestkrat. Sicer se ukaza za skok v naslovu 07 resnično izvede natanko petkrat, ampak za prvi prihod na ta mesta, se izvede postopek utripanja. Zato program utripne skupno šestkrat. Številčno spremenljivko spremenite na vrednost 4 in ponovno testirajte program. Sedaj LED-diode utripajo natanko petkrat.

For zanko je možno uporabiti tudi tako, da se ne skače nazaj, ampak naprej. Tokrat se postopek dejansko izvede petkrat, pri čemer se C na začetku napolni z vrednostjo 5. Vsak preskočen naslov 04 vsebuje relativni skok na samega sebe in s tem tudi neprekinjenost, ki služi kot konec programa.

Naslov	Ukaz	Podatki	Komentar
00	4	5	A = 5
01	5	2	C = A
02	8	0	Nasl.Z. = 0
03	A	5	C-krat 05
04	3	0	Konec
05	1	5	Vrata = 0101
06	2	8	Čakaj 500 ms
07	1	A	Vrata = 1010
08	2	8	Čakaj 500 ms
09	3	6	Skoči -6

45 52 80 A5 30 15 28 1A 28 36

Izpis programa 18: Petkrat utripanje

Primerjave

Potrebna je primerjava dveh številčnih vrednosti. Odvisno od rezultata primerjave se izvede skok. Obe številčni vrednosti se morata nahajati v A in B. V naslednjem primeru se B napolni s številom 5. A prejme svoj rezultat od analognega vhoda AD1. Tukaj je npr., kot je opisano v 4. poglavju, možno priključiti svetlobni senzor. Program bi naj sedaj nepretrgoma izvajal naslednja dejanja:

Ko je $AD1 > 5$
potem: vse LED-diode so vklopljene
drugače: vse LED-diode so izklopljene

V končnem rezultatu se ohrani stikalo za mrak. Ker je LDR priključen proti GND, vodi več svetlosti do manjše napetosti na AD1. LED-diode se izklopijo, kakor hitro se prekorači določena svetlost in se tako ne doseže določene napetosti. Mejna vrednost je 6, saj mora biti rezultat meritve večji od 5.

Naslov	Ukaz	Podatki	Komentar
00	4	5	A = 5
01	5	1	B = A
02	8	0	Nasl.Z. = 0
03	6	9	A = AD1
04	C	1	Preskoči, če $A > B$
05	9	8	Nasl. 08
06	1	F	LED-diode 1111
07	3	4	Nasl. 03
08	1	0	LED-diode 0000
09	3	6	Nasl. 03

45 51 80 69 C1 98 1F 34 10 36

Izpis programa 19: Enostavno stikalo za mrak

Testirajte program, tako da svetlobni senzor z roko bolj ali manj zasenčite. Ugotovili boste, da je osnovna funkcija izpolnjena. Vsekakor pa večinoma pride do nelepega stranskega učinka. Točno na meji med vklopom in izklopom LED-diode nekontrolirano utripajo. Predvsem pri umetni svetlobi svetlost na hitro zaniha na določeno srednjo vrednost. To plapolanje sicer program pravilno izmeri, vendar pa rezultat ni takšen, kakršnega bi se pričakovalo od stikala za mrak. Izboljšano stikalo za mrak je prikazano v 18. poglavju.

AND, OR in XOR

Dve binarni stanji se lahko povežeta v eno novo stanje. Primer za to je UND-funkcija: Če ima bit 1 stanje 1 in bit 2 stanje 1, bo izhodiščno stanje prav tako 1. Tudi binarne številke z več biti je možno tako povezati. Povezava »10 AND 3 = 2« je razumljiva, če je napisana v binarnih številkah:

1010 AND
0011 =
0010

Naslednji program povezuje izhodiščna stanja s konstantno številko 3. AND-funkcija pri tem praktično poskrbi za to, da sta oba spodnja bita maskirana (filtrirana). V stanju mirovanja imajo vhodna vrata stanje 1111. UNDS-povezava z 0011 prenese stanje 0011 do LED-diod.

Če je eden izmed vhodov E1 ali E2 nameščen na GD2, je stanje 0 vidno tudi na izhodih. Spreminjanje E3 in E4 nima nobenega učinka.

Naslov	Ukaz	Podatki	Komentar
00	6	4	A = Din
01	5	1	B = A
02	4	3	A = 3
03	7	7	A = A And B
04	5	4	Vrata = A
05	3	5	Skoči -5

64 51 43 77 54 35

Izpis programa 20: Uporaba AND-funkcije

Spremenite program in testirajte tudi druge logične funkcije. OR-funkcija (78) se lahko uporabi za postavitve določenih vhodnih stanj na 1: 64 51 43 78 54 35

1010 OR
0011 =
1011

Z XOR-funkcijo (ekskluzivno-ali, 79) lahko invertiramo posamezne bite: 64 51 43 79 54 35

1010 XOR
0011 =
1001

Podprogrami

Če bi naj bili deli programov večkrat uporabljeni, se vpišejo v podprogram: Tako se pogosto prihrani spominsko mesto, včasih pa tudi veliko tipkarskega dela. Naslednji primer prikazuje uporabo podprograma, katerega se na dveh mestih priključuje v glavni program. V tem primeru vsebuje podprogram samo ena navodila ($A = A-1$) in ukaz za vrnitev. Zato se tukaj spominsko mesto ne prihrani, ampak primer služi le za prikaz CALL-ukaza in RET-ukaza.

Glavni program:

Naslov	Ukaz	Podatki	Komentar
00	8	0	Nasl.Z. = 0
01	D	8	Priklic 08
02	5	4	Izhod
03	2	9	Čakaj 1 s
04	D	8	Priklic 08
05	5	4	Izhod
06	2	8	Čakaj 0,5 s
07	3	7	Skoči -7

Podprogram:

Naslov	Ukaz	Podatki	Komentar
08	7	2	$A = A-1$
09	E	0	Vrnitev

80 D8 54 29 D8 54 28 37 72 E0

Izpis programa 21: Podprogrami

08 7 2 A = A-1
09 E 0 Ret

Rezultat tega programa je padajoč binarni števec z neenakomernimi časovnimi zakasnitvami. Testirajte tudi druge ukaze v podprogramu.

Med vzorčnimi programi v stanju tovarniških nastavitvev je tudi več podprogramov za splošno uporabo. V prilogi so v celoti naštetih. Za lastno uporabo mora biti poznan le vektorski naslov:

50: Podprogram Zvok dolg
52: Podprogram Zvok kratek
53: Podprogram Zvok poljuben, dolžina v A
60: Podprogram Čakaj na pritisk tipke na S1
68: Podprogram Čakaj na pritisk tipke na S2
70: Podprogram Vnos števil s S1 in S2

Sedaj se uporablja podprogram od naslova 60 dalje za sestavo števca, katerega bo možno upravljati s tipko S1. Stanje števca se prične z 0. Glavni program je sorazmerno kratek, saj se je kompleksna naloga vnosa tipk prenese v podprogram.

Naslov	Ukaz	Podatki	Komentar
00	4	0	A = 0
01	5	4	Izhod
02	7	1	A = A+1
03	8	6	Stran 6
04	D	0	Priklic 60, Tipka S1
05	3	4	Skoči -4

40 54 71 86 D0 34

Izpis programa 22: Števec z upravljanjem preko S1

Testirajte program. Če desetkrat pritisnete S1, bi naj bil rezultat 1010. Program spremenite tako, da se bo uporabljal podprogram od naslova 68 naprej. Sedaj se števec odzove na S2.

Stikalo za mrak

Če svetlost okolice pade pod določeno mejno vrednost, je s stikalom za mrak možno vklopiti svetlobo. Če postane svetleje, pa je potrebno svetlobo ugasniti. Prepričajte se, da svetloba ne plapolna na meji med svetlim in temnim. To lahko rešite s pomočjo histereze, z določeno razdaljo med vklopno in izklopno svetlostjo. Tukaj predstavljeni program deluje po naslednjih pravilih:

- Če napetost na AD1 ni večja od 5, se izklopi.
- Če napetost na AD1 ni manjša od 9, se vklopi.

Tako dobimo srednje področje, v katerem ne more priti do spremembe izhodiščnega stanja. To vrzel zapolnjuje plapolanje LED-diod.

0–5: LED-diode izklopljene
6–8: LED-diode nespremenjene
9–15: LED-diode vklopljene

Naslov	Ukaz	Podatki	Komentar
00	1	0	LED-diode 0000
01	4	5	A = 5
02	5	1	B = A
03	6	9	A = AD1
04	C	1	Preskoči, če A>B
05	C	1	LED-diode 0000
06	4	9	A = 9
07	5	1	B = A
08	6	9	A = AD1
09	C	2	Preskoči, če A<B
0A	1	F	LED-diode 1111
0B	3	A	Skoči -10

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

Izpis programa 23: Stikalo za mrak s histerezo

LED-zatemnilnik

Cilj tega primera programov je možnost upravljanja LED-diode. Svetlost LED-diode na PWM-izhodu je potrebno nastaviti s tipkami. Za doseg naslednje stopnje svetlosti lahko vsakokrat na kratko pritisnemo eno tipko ali konstantno držimo tipko, pri čemer se svetlost stalno spreminja.

V jedru programa se uporabljajo že poznani skip ukazi. Če ne pritisnete konkretne tipke, se pripadajoči ukaz za zvišanje ali znižanje vsebine akumulacijske baterije preskoči. Težava je tudi v tem, da pri tem običajno pride tudi do prekoračitve iz 15 na 0 ali iz 0 na 15. Za preprečitev te prekoračitve je potrebnega nekaj več truda. Pri tem je namreč zmeraj potrebno poizvedeti, ali je bil že spodnji konec (0) ali zgornji konec (15) dosežen. Ker je pri primerjavi načeloma udeležen akumulator, je potrebno njegovo vsebino vmes zmeraj shranjevati. Za to se uporablja spremenljivka C.

Naslov	Ukaz	Podatki	Komentar
00	8	0	Nasl.Z. = 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	Preskoči, če A<B
08	9	B	Skoči 0B
09	C	F	Preskoči, če 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	Preskoči, če A>B
10	9	0	Skoči 00
11	C	E	Preskoči, če S1 = 1
12	7	2	A = A - 1
15	9	0	Skoči 00

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

Izpis programa 24: Upravljanje svetlobe

Kodna ključavnica

Tukaj predstavljena kodna ključavnica izklopi PWM-izhod, ko uporabnik vnese pravilno zaporedje števil. Vnos števil mora potekati točno po vzorcu programiranja s tipkama S1 in S2. Naslednji program prikazuje vnos posamezne številke s tipko S1. Kot pri postopku programiranja je rezultat prvega pritiska na tipko 0000. Vsak naslednji pritisk S1 zviša izhod za 1. S pritiskom tipke S2 se vnos konča. V tem primeru se program konča z neprekinjenostjo.

Naslov	Ukaz	Podatki	Komentar
00	C	C	S1 = 0?
01	3	1	Skoči -1
02	4	0	A = 0
03	5	4	Dout = A
04	2	3	10 ms
05	C	E	S1 = 1?
06	3	2	Nasl. 04
07	C	F	S2 = 1?
08	3	0	Konec
09	C	C	S1 = 0?
0A	3	3	Nasl. 07
0B	7	1	A = A + 1
0C	2	3	10 ms
0D	C	C	S1 = 1?
0E	3	1	Nasl. 0D
0F	3	C	Nasl. 03

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

Izpis programa 25: Vnos številke

Vnos števil je na voljo tudi kot dokončan podprogram od naslova 70 naprej. Namesto neprekinjenosti v vrstici 08 je tukaj RET-ukaz. Podprogram se z rezultatom vnosa števil v A zapusti.

Naslednja kodna ključavnica trikrat prikliče vnos števil in primerja rezultate s predhodno definiranimi števili. V tem primeru je pravilni vnos 3, 5, 2. Zatem se PWM-izhod z vrednostjo 15 popolnoma upravlja. Nasprotno vsak napačen vnos vodi v neprekinjenost, katero je možno ponovno zapustiti le s ponastavitvijo.

V tem primeru se PWM-izhod obravnava kot običajna digitalna vrata. To je potrebno, ker se vsi izhodi od A1 do A4 uporabljajo za vnos števil. Po vsakem popolnem vnosu se štiri LED-diode izbrisejo, da morebitnemu opazovalcu onemogočijo, da bi dobil napotke o skritih številskih kombinacijah.

Naslov	Ukaz	Podatki	Komentar
00	8	7	Stran 7
01	4	3	A = 3
02	5	1	B = A
03	D	0	Priklic 70
04	C	3	Preskoči, če A=B
05	3	0	Konec
06	1	0	LED-diode izklopljene
07	4	5	A = 5
08	5	1	B = A

Naslov	Ukaz	Podatki	Komentar
09	D	0	Priklic 70
0A	C	3	Preskoči, če A=B
0B	3	0	Konec
0C	1	0	LED-diode izklopljene
0D	4	2	A = 2
0E	5	1	B = A
0F	D	0	Priklic 70
10	C	3	Preskoči, če A=B
11	3	0	Konec
12	1	0	LED-diode izklopljene
13	4	F	A = 15
14	5	9	PWM=A
15	3	0	Konec

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

Izpis programa 26: Kodna ključavnica

Priloga

Seznam vzorčnih programov

Naslov	Ukaz	Podatki	Komentar
00	6	4	A = Din
01	5	1	B = A
02	4	E	A = 1110
03	8	0	Stran 0
04	C	3	A = B?
05	9	8	Skoči 08
06	8	2	Stran 2
07	9	5	Skoči 25, »EkspONENTI«
08	4	D	A = 1101
09	8	0	Stran 0
0A	C	3	A = B?
0B	9	E	Skoči 0E
0C	8	2	Stran 2
0D	9	A	Skoči 2A, »AD/PWM«
0E	4	B	A = 1011
0F	8	1	Stran 1

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

Stran 0: Izbira in zagon vzorčnih programov

Naslov	Ukaz	Podatki	Komentar
10	C	3	A = B?
11	9	4	Skoči 14
12	8	3	Stran 3
13	9	0	Skoči 30, »Naključje«
14	4	7	A = 0111
15	8	1	Stran 1
16	C	3	A = B?
17	9	A	Skoči 1A
18	8	3	Stran 3
19	9	4	Skoči 34, »Stoparica S1«

Naslov	Ukaz	Podatki	Komentar
1A	4	3	A = 0011
1B	8	2	Stran 2
1C	C	3	A = B?
1D	9	0	Skoči 20 »Izmenična utripalka«
1E	8	4	Stran 4
1F	9	0	Skoči 40 »Štoparica S1/S2«

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

Stran 1: Izbira in zagon vzorčnih programov

Naslov	Ukaz	Podatki	Komentar
20	1	1	0001 »Izmenična utripalka«
21	2	8	Čakaj 500 ms
22	1	8	1000
23	2	8	Čakaj 500 ms
24	3	4	Skoči -4
25	7	1	A = A + 1 »Eksponenti«
26	5	4	Vrata = A
27	5	9	PWM = A
28	2	6	Čakaj 100 ms
29	3	4	Skoči -4
2A	6	9	A = AD1 »AD/PWM«
2B	5	4	Vrata = A
2C	5	9	PWM = A
2D	2	6	Čakaj 100 ms
2E	3	4	Skoči -4
2F	F	F	-

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

Stran 2: Vzročni programi: Izmenična utripalka, eksponenti, AD/PWM

Naslov	Ukaz	Podatki	Komentar
30	5	4	Vrata = A »Naključje«
31	C	E	S1 = 1?
32	7	1	A = A + 1
33	3	3	Skoči -3
34	2	2	Čakaj 5 ms »Štoparica S1«
35	C	C	S1 = 0?
36	3	2	Skoči -2
37	4	0	A = 0
38	2	2	Čakaj 5 ms
39	7	1	A = A + 1
3A	5	4	Vrata = A
2B	C	E	S1 = 1?
3C	3	4	Skoči -4
3D	3	9	Skoči -9
3E	F	F	-
3F	F	F	-

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

Stran 3: Vzročni programi: Naključje, Štoparica S1

Naslov	Ukaz	Podatki	Komentar
40	8	6	Stran 3 »Stoparica Start/stop«
41	D	0	Priklic »Čakaj S1«
42	4	0	A = 0
43	7	1	A = A + 1
44	5	4	Vrata = A
45	2	3	Čakaj 10 ms
46	C	D	S2 = 0?
47	3	4	Skoči -4
48	D	8	Priklic »Čakaj S2«
49	4	0	A = 0
4A	5	4	Vrata = A
4B	3	B	Skoči -11
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

Stran 4: Vzorčni program Zagon/ustavitev štoparice

Naslov	Ukaz	Podatki	Komentar
50	4	F	A = »Zvok dolg«
51	9	3	Nasl. 03
52	4	5	A = 5 »Zvok kratek«
53	5	3	D = A »Zvok spremenljiv«
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	D-krat 04
5E	1	0	Dout 0
5F	E	0	Vrnitev

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

Stran 5: Podprogram Predvajanje zvoka

Naslov	Ukaz	Podatki	Komentar
60	2	3	Čakaj 10 ms »Čakaj S1«
61	C	E	S1 = 1?
62	3	2	Skoči -2
63	2	3	Čakaj 10 ms
64	C	C	S1 = 0?
65	3	1	Skoči -1
66	E	0	Vrnitev
67	F	F	-

Naslov	Ukaz	Podatki	Komentar
68	2	3	Čakaj 10 ms »Čakaj S2«
69	C	F	S2 = 1?
6A	3	2	Skoči -2
6B	2	3	Čakaj 10 ms
6C	C	D	S2 = 0?
6D	3	1	Skoči -1
6E	E	0	Vrnitev
6F	F	F	-

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

Stran 6: Podprograma Čakaj S1 in Čakaj S2

Naslov	Ukaz	Podatki	Komentar
70	C	C	S1 = 0? »Vnos s tipkami«
71	3	1	Skoči -1
72	4	0	A = 0
73	5	4	Vhod = A
74	2	3	Čakaj 10 ms
75	C	E	S1 = 1?
76	3	2	Skoči -2
77	C	F	S2 = 1?
78	E	0	Vrnitev
79	C	C	S1 = 0?
7A	3	3	Skoči -3
7B	7	1	A = A + 1
7C	2	3	Čakaj 10 ms
7D	C	C	S1 = 1?
7E	3	1	Skoči -1
7F	3	C	Skoči -12

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

Stran 7: Podprogram Vnos s tipkami

Tabela ukazov

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
	Port =	Wait	Jump -	A =	... = A	A = ...	A = ...	Page	Jump	C*	D*	Skip if ...	Call	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	

	1	2	3	4	5	6	7	8	9	A	B	C	D	E
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	



Električne in elektronske naprave ni dovoljeno metati med gospodinjske odpadke! Odslužen izdelek po izteku življenjske dobe odstranite v skladu z veljavnimi zakonskimi določili. Obstajajo lokalna zbirališča odpadkov, na katerih lahko brezplačno oddate odslužene električne naprave. O lokaciji takšnih zbirališč se pozanimajte na svoji občini.



Ta izdelek je v skladu z veljavnimi direktivami CE, v kolikor ga uporabljate v skladu s priloženimi navodili za uporabo. Ta navodila za uporabo sodijo k izdelku. Če izdelek predate v uporabo tretji osebi, priložite tudi ta navodila.

GARANCIJSKI LIST

Izdelek: **Profesionalni učni komplet za mikrokrmilnike Conrad**
Kat. št.: **19 22 86**

Conrad Electronic d.o.o. k.d.
Ljubljanska c. 66, 1290 Grosuplje
Fax: 01/78 11 250, Tel: 01/78 11 248
www.conrad.si, info@conrad.si

Garancijska izjava:

Proizvajalec jamči za kakovost oziroma brezhibno delovanje v garancijskem roku, ki začne teči z izročitvijo blaga potrošniku. **Garancija velja na območju Republike Slovenije.**

Garancija za izdelek je 1 leto.

Izdelek, ki bo poslan v reklamacijo, vam bomo najkasneje v skupnem roku 45 dni vrnilo popravljene ali ga zamenjali z enakim novim in brezhibnim izdelkom. Okvare zaradi neupoštevanja priloženih navodil, nepravilne uporabe, malomarnega ravnanja z izdelkom in mehanske poškodbe so izvzete iz garancijskih pogojev. **Garancija ne izključuje pravic potrošnika, ki izhajajo iz odgovornosti prodajalca za napake na blagu.**

Vzdrževanje, nadomestne dele in priklopne aparate proizvajalec zagotavlja še 3 leta po preteku garancije.

Servisiranje izvaja proizvajalec sam na sedežu firme CONRAD ELECTRONIC SE, Klaus-Conrad-Strasse 1, Nemčija.

Pokvarjen izdelek pošljete na naslov: Conrad Electronic d.o.o. k.d., Ljubljanska cesta 66, 1290 Grosuplje, skupaj z izpolnjenim garancijskim listom.

Prodajalec: _____

Datum izročitve blaga in žig prodajalca:

Garancija velja od dneva nakupa izdelka, kar kupec dokaže s priloženim, pravilno izpolnjenim garancijskim listom.

- Garancija velja na območju Republike Slovenije.
- Garancija ne izključuje pravic potrošnika, ki izhajajo iz odgovornosti prodajalca za napake na blagu.