

5. Analogni vhodi

Spoznali boste vmesnik mikrokrmilnika, ki omogoča uporabo analognih vhodov mikrokrmilnika. Naučili se boste konfigurirati posamezne priključke vrat PORTA za izbiro analognih vhodov. Naučili se boste uporabljati 10 ali 8-bitni rezultat in ločevati med desno oz. levo poravnavo ter interpretirati številčni rezultat AD pretvorbe.

A/D pretvornik je 10-bitni, vsebuje 5-kanalni multiplekser, modul pa ima tudi pozitivni in negativni referenčni vhod [4].

Registri A/D modula so:

- register višjih bitov rezultata A/D pretvorbe (A/D Result High Register, **ADRESH**)
- register nižjih bitov rezultata A/D pretvorbe (A/D Result Low Register, **ADRESL**)
- A/D kontrolni register 0 (A/D Control Register 0, **ADCON0**)
- A/D kontrolni register 1 (A/D Control Register 1, **ADCON1**)

ADCON0 nadzoruje delovanje A/D modula, **ADCON1** pa nastavi funkcije priključkov. Registra **ADRESH:ADRESL** vsebujeta 10-bitni rezultat pretvorbe. Ko je A/D pretvorba zaključena, se rezultat naloži v oba registra, bit $\overline{\text{GO/DONE}}$ (**ADCON0<2>**) se postavi na '0', bit ADIF se postavi na '1'.

5.1. Delovanje in uporaba AD pretvornika

Postopek uporabe pretvornika je opisan v naslednjih točkah:

1. Konfiguriraj A/D modul:

- Konfiguriraj analogne priključke/napetostne reference in digitalne vhode/izhode (**ADCON1**)
- Izberi vhodni kanal A/D pretvornika (**ADCON0**)
- Izberi takt A/D pretvorbe (**ADCON0**)
- Vključi A/D modul (**ADCON0**)

2. Po potrebi konfiguriraj prekinitve, ki jih proži A/D pretvornik

- Zbriši bit **ADIF**
- Postavi bit **ADIE**
- Postavi bit **GIE**

3. Počakaj, da se inicializacija izvede

4. Poženi pretvorbo _____

- Postavi bit $\overline{\text{GO/DONE}}$ oz. **ADGO** (**ADCON0**)

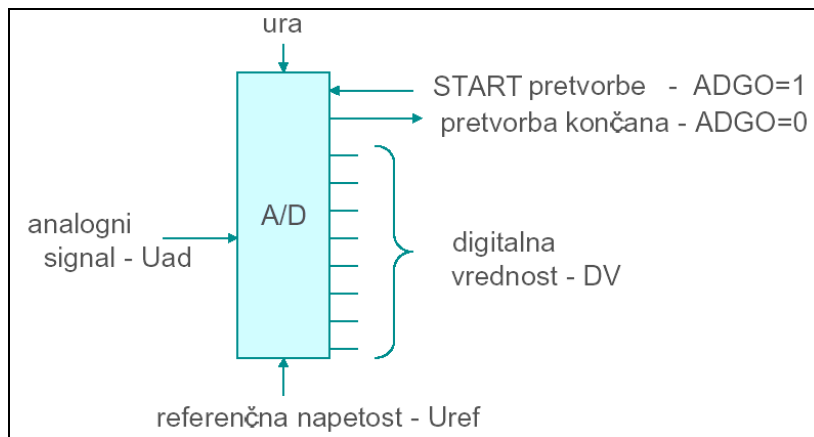
5. Počakaj, da se A/D pretvorba konča, obstajata dva načina:

- Čakaj, da se zbriše bit $\overline{\text{GO/DONE}}$ oz. **ADGO** ali
- Čakaj na A/D prekinitvev

6. Preberi rezultat v registrih (**ADRESH:ADRESL**), po potrebi zbriši bit **ADIF**.

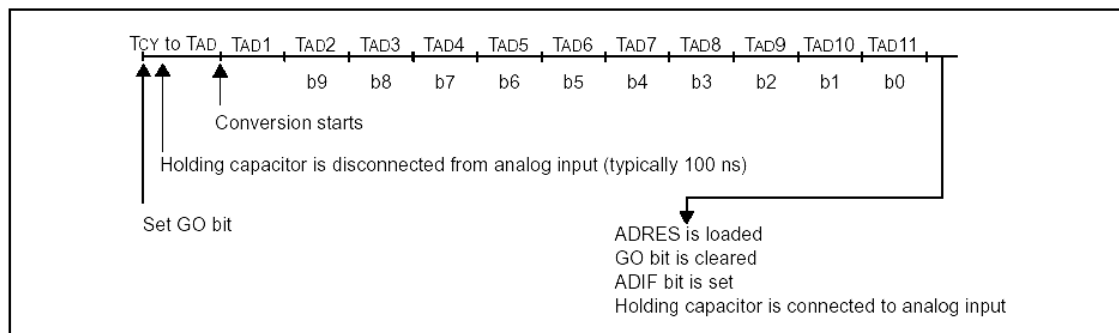
7. Za naslednjo pretvorbo se vrni v korak 1 ali 2 (kateri ustreza).

Običajno uporabljamo A/D pretvornik tako, da ga »programsko startamo« (postavitev bita 2 v registru **ADCON1**) in zatem počakamo na zaključek pretvorbe, tako da programsko testiramo isti statusni bit **ADGO** (Slika 5-1).



Slika 5-1: Simbolični prikaz delovanja A/D pretvornika

Kratek čas (največ 2 μs) po postavitvi bita **ADGO** (GO/DONE), se **zajame (trenutno otipa) vrednost na analognem vhodu** (širina okna je približno 100 ns) in se shrani v »zadrževalni kondenzator«. Zatem se prične proces pretvorbe (Slika 5-2) po postopku »zaporednega približevanja«.



Slika 5-2: Časovni potek delovanja A/D pretvorbe

Analogno digitalna pretvorba traja okrog 20 μs ($12T_{AD}$, pri čemer je interval T_{AD} enak 1,6 μs). Ko se zaključi A/D pretvorba, **mora miniti vsaj $2T_{AD}$ (3,2 μs)**, preden lahko sprožimo naslednje zajemanje analognega vhoda.

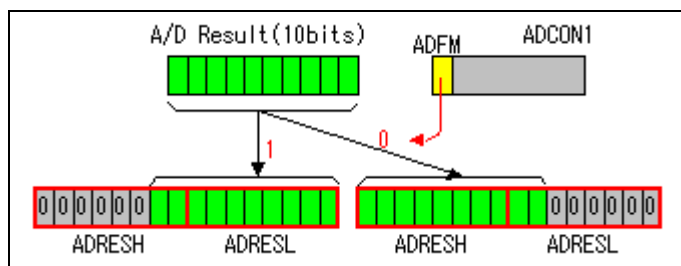
Rezultat (digitalna vrednost – DV) je seveda **odvisen od referenčne napetosti** (izbiramo lahko med notranjo – napajanje 5,0 V ali zunanjo - na vhodih V_{ref-} in V_{ref+}), kakor tudi od kvalitete tiskanega vezja, dolžine in oblike povezav ter od **preciznosti** samega pretvornika.

Napetost na vhodu izračunamo na osnovi 8 ali 10-bitne vrednosti na naslednji način:

$U_{ad} = DV \cdot q$, pri čemer je: **$q = U_{ref}/256$** (8-bitni rezultat), **$q = U_{ref}/1024$** (10-bitni rezultat);

DV je pri **8-bitni** vrednosti rezultata med **0 in 255**, pri **10-bitni** pa med **0 in 1023**.

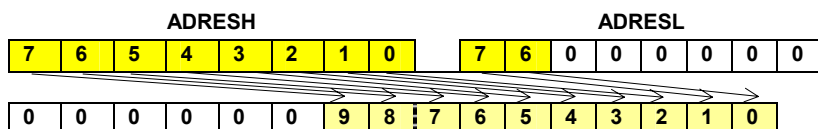
Vrednost kvanta: **$q = 4,883 \text{ mV}$** (10-bitni rezultat) oz. **$q = 19,53 \text{ mV}$** (8-bitni rezultat) ob predpostavki, da je referenčna napetost natanko **$U_{ref} = 5,00 \text{ V}$** .

Način shranjevanja rezultata AD pretvorbe (Slika 5-3):

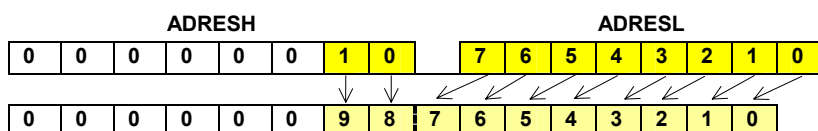
Slika 5-3: Desna ali leva poravnava rezultata A/D pretvorbe

V kolikor se zadovoljimo z **8-bitno vrednostjo rezultata (0 do 255)**, izberemo levo poravnavo in rezultat prenesemo iz registra **ADRESH** v 8-bitno spremenljivko. Pri tem je ločljivost A/D pretvorbe enaka 19,53 mV (pri območju - referenci 5,0 V).

Če je pomemben popoln **10-bitni rezultat (0 do 1023)**, je primernejša izbira **desne poravnave**, postopek shranjevanja pa je prikazan spodaj. Pri tem je ločljivost A/D pretvorbe enaka 4,88 mV (pri območju - referenci 5,0 V).

Postopek za prenos levo poravnane rezultata v 16-bitno spremenljivko (Slika 5-4):

Slika 5-4: Shranjevanje levo poravnane rezultata v 16-bitno spremenljivko

Postopek za prenos desno poravnane rezultata v 16-bitno spremenljivko (Slika 5-5):

Slika 5-5: Shranjevanje desno poravnane rezultata v 8 ali 16-bitno sprem.

5.2. Registri AD pretvornika in primer programa**5.2.1. Register ADCON0**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit7	6	5	4	3	2	1	bit0

Legenda: R = omogočeno branje bita, W = omogočeno pisanje v bit, U = neuporabljivi bit, beri kot '0', -n = vrednost ob POR resetu

bit 7-6: **ADCS1:ADCS0**: bita za izbiro takta A/D pretvorbe

00 = FOSC/2

01 = FOSC/8

10 = FOSC/32 (priporočljiva izbira pri 20 Mhz taktu)

11 = FRC (zunanji RC oscilator)

5. Analogni vhodi

Uvod v programiranje mikrokrmilnikov, zbrano gradivo za predavanja

bit 5-3: **CHS2:CHS0**: biti za izbiro analognega kanala

- 000 = kanal 0, (RA0/AN0)
- 001 = kanal 1, (RA1/AN1)
- 010 = kanal 2, (RA2/AN2)
- 011 = kanal 3, (RA3/AN3)
- 100 = kanal 4, (RA5/AN4)

bit 2: **GO/DONE**: Statusni bit A/D pretvorbe (bit se lahko imenuje tudi **ADGO**)

Če je ADON = 1

- 1 = A/D pretvorba poteka (postavitev tega bita sproži A/D pretvorbo)
- 0 = A/D pretvorba ne poteka (ta bit se avtomatsko postavi na '0' ko je A/D pretvorba končana)

bit 1: **Neuporabljen**: beri kot '0'

bit 0: **ADON**: A/D On bit

- 1 = A/D pretvornik deluje
- 0 = A/D pretvornik je izključen

5.2.2. Register ADCON1

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

bit7

bit0

Legenda: R = omogočeno branje bita, W = omogočeno pisanje v bit, U = neuporabljen bit, beri kot '0', -n = vrednost ob POR resetu

bit 7: **ADFM**: izbira formata rezultata A/D pretvorbe

- 1 = **Desno poravnan**. 6 najbolj pomembnih (most significant) bitov **ADRESH** se prebere kot '0'.
- 0 = **Levo poravnan**. 6 najmanj pomembnih (least significant) bitov **ADRESL** se prebere kot '0'.

bit 6-4: **Neuporabljeni**: Beri kot '0'

bit 3-0: **PCFG3:PCFG0**: Biti za konfiguracijo A/D vmesnika (Configuration Control bits)

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN / Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100⁽³⁾	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x⁽³⁾	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

Legenda: A = analogni vhod, D = digitalni vhod/izhod

Opomba 1: Ti kanali na čipih z 28 nožicami (16F876) niso na voljo, ampak pri PIC mikrokrmilnikih s 40 nožicami (16F877)

2: Stolpec označuje število analognih vhodov in število analognih vhodov, ki jih uporabljamo kot vhode za referenčno napetost

3: V krepkem tisku sta vrstici, ki pomenita najbolj verjetno izbiro.

Popoln zgled programa za A/D pretvornik za izvajanje na MPU-PIC16F876:

Primer programa v zbirnem jeziku	Primer programa v C-jeziku
<pre> ;----- ; Naziv programa: Demo program za AD pretvorbo: AN3 shrani v RezH ; in RezL , v casu AD pretvorbe vklopljena LD4 na RA5 izhodu ;----- list p=16f876 ;izbira tipa cipa include <p16f876.inc> ;vkljucitev datoteke z definicijami ; simbolov ;----- ;--- spremenljivke RezH equ 0x20 ;zgornjih 8 bitov rezultata RezL equ 0x21 ;spodnjih 8 bitov rezultata ;--- Nastavitev vektorjev org 0x00 ;RESET vektor goto Main ;skok na glavni program Main org 0x05 ;zacetek programske kode Main ;----- Inicializacija registrov vhodno/izhodnih vmesnikov bsf STATUS,RP0 ;banka 1 movlw b'00011011' ;0 - izhod, 1 - vhod movwf TRISA ;RA5-izhod, RA4-vhod, ; RA3/AN3-vhod, RA2-izhod, RA1/AN1-vhod, RA0/AN0-vh. clrf TRISB ;vsi pini porta B so izhodni clrf TRISC ;vsi pini porta c so izhodni ;--- inicializacija A/D - AN3 movlw b'10000100' ;desna poravnava, vhod AN3 je ; analogni, tudi AN1 in AN0 movwf ADCON1 ;bit7=1 -> desna poravnava 10- ; bitnega rezultata bcf STATUS,RP0 ;nazaj v banko 0 movlw b'10011001' ;takt (f/32,takt=20MHz), izbira ; anal. kan. (AN3), AD pretv. je omogocen movwf ADCON0 ;----- clrf INTCON ;prekinitev ne bomo uporabljali clrf PORTC ;izhode porta C postavi na 0 ;----- jedro programa Zanka bsf PORTA,5 ;vklopi LD4 na RA5 ;--- Start AD pretvorbe in cakanje na rezultat bsf ADCON0,2 ;start AD pretvorbe (postavi bit 2 (GO_DONE) na 1 pocakaj btfsc ADCON0,2;testiraj GO_DONE goto pocakaj ;cakanje na izvedbo AD pretvorbe (pribl. 20 us) ;--- pretvorba je koncana in rezultat pripravljen bcf PORTA,5 ;izklopi LD4 na RA5 movf ADRESH,w ;zgornjih 8-bitov (dejansko le 2 bita!) rezultata -> W movwf RezH ;premik visjih osmih bitov iz delovnega registra w v sprem RezH bsf STATUS,RP0 ;v banko1, kjer je ADRESL movf ADRESL,w ;spodnjih 8-bitov rezultata -> w bcf STATUS,RP0 ;nazaj v banko0 movwf RezL ;premik nizjih 8-bitov v RezL Test nop ;«Test» simbol za Breakpoint goto Zanka ;ponovitev AD pretvorbe end ;psevdo ukaz za konec programa </pre>	<pre> /* ----- Demo program za AD pretvorbo vhoda AN3, rezultat shrani v 16-bit sprem. Rez ; v casu AD pretvorbe je vklopljena LD4 na RA5 izhodu -----*/ #include <pic1687x.h> /* vkljucitev datoteke z definicijami simbolov */ unsigned int Rez; /* sprem. za rezultat pretvorbe */ void main(void) { /* ----- Inicializacija registrov vhodno/izhodnih vmesnikov */ TRISA=0x1b; /*b'00011011' RA5-izhod, RA4-vhod, RA3/AN3-vhod, RA2-izhod, RA1/AN1-vhod, RA0/AN0-vhod */ TRISB=0; /* vsi pini porta B so izhodni */ TRISC=0; /* vsi pini porta C so izhodni */ /*--- inicializacija A/D - AN3 */ ADCON1=0x84; /* b'10000100' desna poravnava, vhod AN3 je analogni, tudi AN1 in AN0 */ ADCON0=0x99; /* b'10011001' takt(f/32,takt=20 MHz), izbira anal. kan. (AN3) */ PORTC=0; /* izhode porta C postavi na 0 */ INTCON=0; /* prekinitev ne bomo uporabljali */ /* ----- jedro programa */ while (1) /* neskoncna zanka */ { RA5=1; /* vklop LD4 */ /*--- Start AD pretvorbe in cakanje na rezultat */ ADGO=1; /* start AD pretvorbe, postavi bit 2 (GO_DONE) na 1 */ while (ADGO==1); /* cakanje na izvedbo AD pretvorbe (~ 20 us) */ RA5=0; /* izklop LD4 */ Rez = ADRESL (ADRESH<<8); /* rezultat AD pretvorbe */ asm("test nop"); /* moznost nastav. Breakpoint na simb. "test" */ } /* ponovitev AD pretvorbe */ } </pre>

Program 5-1: Popoln primer programa v zbirnem in C-jeziku za A/D pretvornik

Za razumevanje tematike je potrebno tudi predznanje iz osnov programiranja in številskih sistemov. Dodatna znanja iz programiranja v zbirnem jeziku so v: [2][4][6][7]. Za razumevanje programov, ki so kodirani v C-jeziku, je potrebno poznavanje osnov ANSI C strukturnega programiranja: [15][11][13][5]. Veliko rešitev praktičnih nalog z uporabo PIC mikrokrmilnikov, je na spletnem portalu http://www.interq.or.jp/japan/se-inoue/e_pic6.htm. Podrobnejši opis delovanja A/D pretvornika s primeri uporabe v PIC mikrokrmilnikih najdemo tudi v: [8][16][17][18].

Vprašanja za utrjevanje:

Napišite podprogram z imenom InAD1 za inicializacijo AD pretvornika (vhod AN1) z desno poravnavo rezultata.

Rešitev:

```

;----- Inicializacija registrov vhodno/izhodnih vmesnikov
InAD1 bsf    STATUS,RP0    ; 1 -> RP0, Bank1 za dostop do TRISA in ADCON1
      bsf    TRISA,1      ; 1 -> TRISA.1, RA1/AN1- vhod
;--- inicializacija A/D - AN1
      movlw  b'10000100'  ;priprava vredn. za ADCON1, vhod AN1-analogni, tudi AN3 in AN0
      movwf  ADCON1      ;bit7=1 -> desna poravnava 10-bitnega rezultata
      bcf    STATUS,RP0    ;0 ->RP0, nazaj v banko 0
      movlw  b'10001001'  ;takt (f/32,Fosc=20MHz), izbira anal. kan. AN1, AD omogocen
      movwf  ADCON0      ;konec inicializacije
      return             ;konec podprograma

```

1. *Napišite podprogram z imenom InAD3 za inicializacijo AD pretvornika (vhod AN3) z levo poravnavo rezultata.*
2. *Napišite podprogram (ki se navezuje na točko 1) za Start AD pretvorbe vhoda AN3 in shranjevanje 8-bitnega rezultata v register W.*
3. *Kolikšna napetost je na analognem vhodu, če smo pri 8-bitni pretvorbi dobili rezultat 7Fh ?*
4. *Kolikšna napetost je na analognem vhodu, če smo pri 10-bitni pretvorbi dobili rezultat 300h in je $V_{ref+} = 4,00 V$?*
5. *Kolikšen je čas AD pretvorbe (približno) in po kolikem času lahko sprožimo ponovno AD pretvorbo ?*