



1918

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

Elektriamite ja jõuelektroonika instituut

PIC mikrokontrolleri õppesüsteem

PIC Assembler

Tallinn 2005

1. PIC Assembler

1.1 PIC mikrokontrolleri programmeerimise alused.

PIC Assembleris koosneb programm järgmistest etappidest:

- Määratakse kindlaks mikrokontrolleri tüüp
- Lisatakse fail mikrokontrolleri spetsiifiliste andmetega
- Mikrokontrolleri seadistamine
- Muutujate ja mälupiirkondade kindlaksmääramine
- Katkestus – alamprogramm
- Programmikood (võib sisaldada ka alamprogramme)
- Lõpp

list p=16f877a	;Määratakse kindlaks mikrokontrolleri tüüp
include "p16f877a.inc"	;Laetakse fail kirjeldustega (registrite nimed jne)
org 0x00	;Algusaadress (pärast reseti hakatakse koodi lugema siit)
goto main	;Põhiprogrammi juurde
org 0x04	;Katkestuse korral pöördutakse programmi juurest siia.
	;Siia kantakse kood, mida täidetakse katkestuse korral
main	;Põhiprogramm (võib sisaldada alamprogramme)
end	;Programmi lõpp

List p käsuga määratakse kompilaatori jaoks kindlaks mikrokontrolleri tüüp.

Include käsuga laetakse vastavat mikrokontrolleerit kirjeldav fail. Selles failis on kompilaatori jaoks kindlaks määratud tähtsamate seaderegistrite nimed ja aadressid. Eraldi nimed on antud ka tähtsamatele bittidele seaderegistrites.

Org käskudega määratakse kindlaks programmi algusaadress, kust alustatakse mikrokontrolleri (taas)käivitumisel ning katkestusprogrammi algus.

1.2 PIC Assembleri käsustik

PIC mikrokontrolleerid on lihtsa RISC arhitektuuriga seadmed, mis piirab ka PIC Assembleri käsustiku 37 käsuga. Vaatamata käskude vähesusele on sellegi poolest võimalik luua vägagi keeruliseid programme.

Enne käsustiku juurde suundumist tuleks selgitada lähemalt olulisemaid termineid ning süntaksiga seotud reegleid.

Käsustikuga seotud terminid:

- w – Tööregister (“working register”)
- f – Register (register, millega sooritatakse tehet)
- d – Lisatingimus käsule (d võimalikud väärtused 0 või 1)
- k – 8-bitine arv (0 – 255)
- b – Biti järjekorranumber 8-bitises arvus (0 – 7)

Mikrokontrolleris käivad kõik andmed läbi tööregistri. Kõigepealt omistatakse väärtus tööregistrile alles seejärel soovitud aadressiga registrile. Samamoodi toimub ka väärtuste lugemine registrist.

Lipud:

Lippudeks (flags) kutsutakse mitmesugustes registrites asuvaid kontrollbitte, mis toimivad “indikaatoritena”, andes infot erinevate programsete ja riistvaraliste protsesside kohta. Assembleris on tähtsad kaks lippu.

- c – “Carry flag” (c lipu väärtus on 1 kui toimus mõne registri ülejooks)
- z – “Zero flag” (z lipu väärtus on 1 kui viimase tehte väärtus oli 0)

Zero ja carry või siis eesti keeles null- ja ülekande lipud on bitid STATUS registris. C – lippu võib kasutada näiteks arvutustehete või 16 – bitise impulssloenduri realiseerimiseks.

Näide: 16 bitine loendur. 16 bitine loendustulemus jagatakse kahte 8-bitisesse registrisse, mis on jagatud “vanemaks baidiks” ja “nooremaks baidiks”. Iga impulss sisendis suurendab “noorema baidi” väärtust 1 võrra. 8-bitise registriga on võimalik loendada kuni 255 impulssi. Juhul kui nooremas baidis on maksimaalne väärtus 255 ja seda 1 võrra suurendada, nullitakse register ja c = 1. Programm kontrollib c lipu väärtust. Juhul kui c = 1, siis suurendatakse vanema baidi väärtust ühe võrra ning nullitakse c lipp. Vanemas baidis on noorema baidi ületäitumiste arv. Näiteks: kui on loendatud 300 impulssi, on toimunud 1 noorema baidi ülejooks, vanema baidi väärtus on 1 ning noorema baidi oma 45. Vanema baidi väärtus * 255 + noorema baidi väärtus = 1 * 255 + 45 = 300.

Süntaksireeglid:

- MPLAB tunnistab kommentaarina lõiku, mis algab semikooloniga.
- Rea alguses võivad olla ainult alamprogrammide nimed (hüppeaadressid) ja kommentaarid.
- Käsud peavad olema rea algusest vähemalt tühiku või tabulaatori kaugusel.

Arvused on võimalik kirjutada PIC Assembleris mitmetes süsteemides.

Näiteks:

- Kuueteistkümnendsüsteem 0x10 (kirjanduses 0x10h)
- Kümnendsüsteem d'10'
- Kahendsüsteem b'10000'

PIC mikrokontrollerite 37 käsku võib jaotada alamkategoriasse:

- Andmete käitlemise (kopeerimine, kustutamine) käsud

- Programmis liikumise käsud
- Aritmeetika- ja loogikakäsud
- Käsud üksikute bittidega opereerimiseks

1.2.1 Andmete käitlemise (kopeerimine, kustutamine) ja programmis liikumise käsud

Kopeerimiskäsud

Kopeerimiskäskudega liigutatakse andmeid.

MOVF

MOVF käsuga kopeeritakse registri f väärtus registrisse...

Süntaks: `movf f,d`

- `d=1` (Registri f väärtus kopeeritakse tööregistrisse w)
- `d=0` (Registri f väärtus kopeeritakse tagasi samasse registrisse f, kasutatav tegemaks kindlaks kas f väärtus on 0, kuna sellisel juhul z lipp STATUS registris=1)

Näide: Kopeeri aadressil 20h oleva registri sisu tööregistrisse.

```
movf 0x20, 1
```

Tehtega kasutatavad lipud: Z

MOVWF

MOVWF käsuga kopeeritakse tööregistri w väärtus registrisse f

Süntaks: `movwf f`

Näide: Kopeeri tööregistri sisu aadressil 20h olevasse registrisse.

```
movwf 0x20'
```

Tehtega kasutatavad lipud: Z

MOVLW

MOVLW käsuga kopeeritakse arv väärtusega 0 – 255 tööregistrisse.

Süntaks: `movlw k`

Näide: Kopeeri tööregistrisse arv 20h.

```
movlw 0x20
```

Tehtega kasutatavad lipud: Z

Kustutamiskäsud

Kustutamiskäskudega on võimalik nullida registreite sisu.

CLRF

CLRF käsuga kustutatakse registri f sisu

Süntaks: clrf f

Näide: Kustuta aadressil 20h oleva registri sisu.

clrf 0x20

Tehtega kasutatavad lipud: Z

CLRW

CLRW käsuga kustutatakse tööregistri w sisu

Süntaks: clrw

Näide: Kustuta tööregistri sisu.

clrw

z lipu väärtus pärast selle käsu täitmist ei muutu.

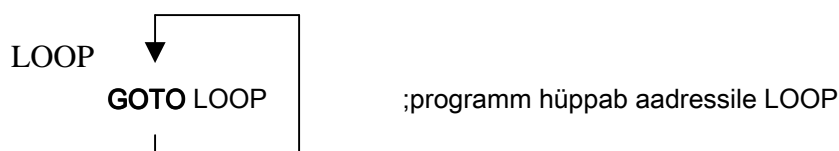
Programmis liikumise käsud

GOTO

GOTO käsuga hüpatakse programmis kindlaksmääratud aadressile.

Süntaks: GOTO k

Näide: Hüppa programmis aadressile LOOP

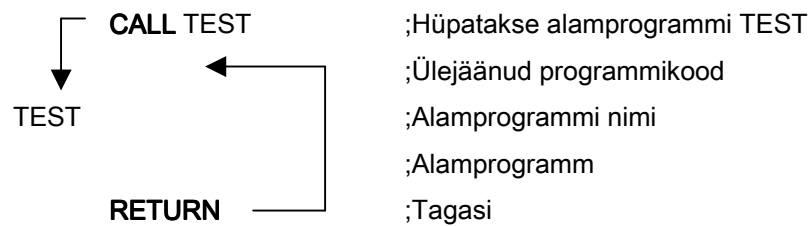


CALL

CALL käsuga minnakse alamprogrammi mille nime tähistab k, alamprogrammist väljutakse käsuga RETURN ning programmi täitmine jätkub CALL käsule järgnevast käsust.

Süntaks: CALL k

Näide: Kutsuda välja alamprogramm TEST



RETURN

RETURN käsuga väljutakse alamprogrammist ning pöörduakse tagasi esialgse programmi täitmise juurde.

Süntaks: RETURN

Näide: Vaata käsku CALL.

RETLW

RETLW käsuga väljutakse alamprogrammist ning pöörduakse tagasi esialgse programmi täitmise juurde ning tööregistrisse kirjutatakse 8-bitine arv k.

Süntaks: RETLW k

Näide: Vaata käsku CALL.

RETFIE

RETFIE käsuga väljutakse alamprogrammist, kuhu pöördueti pärast katkestust (katkestuste kohta täpsem informatsioon peatükis 6.7) ning pöörduakse tagasi esialgse programmi täitmise juurde.

Süntaks: RETFIE

1.2.2 Aritmeetika ja loogikakäsud

Aritmeetikakäsud

ADDWF

ADDWF käsuga liidetakse tööregistri w sisu registrile f.

Süntaks: ADDWF f, d

- d=0 (Liitmise tulemus kopeeritakse tööregistrisse w)
- d=1 (Liitmise tulemus kopeeritakse registrisse f)

Näide: Liida tööregistri sisu registrile aadressiga 20h.

addwf 0x20, 1

Tehtega kasutatavad lipud: C, Z

ADDLW

ADDLW käsuga liidetakse tööregistrile w arv l.

Süntaks: ADDLW k

Näide: Liida tööregistrile arv 20h.

addlw 0x20

Tehtega kasutatavad lipud: C, Z

SUBWF

SUBWF käsuga lahutatakse tööregistri w sisu registri f omast.

Süntaks: SUBWF f, d

- d=0 (Lahutamise tulemus kopeeritakse tööregistrisse w)
- d=1 (Lahutamise tulemus kopeeritakse registrisse f)

Näide: Lahuta registri 20h väärtusest tööregistri väärtus.

subwf 0x20, 1

Tehtega kasutatavad lipud: C, Z

SUBLW

SUBLW käsuga lahutatakse tööregistri w väärtusest arv l.

Süntaks: SUBLW k

Näide: Lahuta tööregistri väärtusest arv 3h.

sublw 0x03

Tehtega kasutatavad lipud: C, Z

INCF

INCF käsuga suurendatakse registri f väärtust 1 võrra.

Süntaks: INCF f, d

- d=0 (Tulemus kopeeritakse tööregistrisse w)
- d=1 (Tulemus kopeeritakse registrisse f)

Näide: Suurenda aadressil 20h oleva registri väärtust 1 võrra.

incf 0x20, 1

Tehtega kasutatavad lipud: Z

DECF

DECF käsuga vähendatakse registri f väärtust 1 võrra.

Süntaks: DECF f, d

- d=0 (Tulemus kopeeritakse tööregistrisse w)
- d=1 (Tulemus kopeeritakse registrisse f)

Näide: Vähenda aadressil 20h oleva registri väärtust 1 võrra.

decf 0x20, 1

Tehtega kasutatavad lipud: Z

INCFSZ

INCFSZ käsuga suurendatakse registri f väärtust 1 võrra ja hüpatakse järgmisest käsust üle kui pärast suurendamist registri väärtus $f = 0$ ning jätkatakse ülejärgmisega.

Süntaks: INCFSZ f, d

- $d=0$ (Tulemus kopeeritakse tööregistrisse w)
- $d=1$ (Tulemus kopeeritakse registrisse f)

Näide: Suurenda aadressil 20h oleva registri väärtust 1 võrra, juhul kui registri väärtus võrdub nulliga järgmine käsk vahele jätta.

incfsz 0x20, 1

DECFSZ

DECFSZ käsuga suurendatakse registri f väärtust 1 võrra ja hüpatakse järgmisest käsust üle kui pärast suurendamist registri väärtus $f = 0$ ning jätkatakse ülejärgmisega.

Süntaks: DECFSZ f, d

- $d=0$ (Tulemus kopeeritakse tööregistrisse w)
- $d=1$ (Tulemus kopeeritakse registrisse f)

Näide: Vähenda aadressil 20h oleva registri väärtust 1 võrra, juhul kui registri väärtus võrdub nulliga järgmine käsk vahele jätta.

decfsz 0x20, 1

COMF

COMF käsuga invertteeritakse kõik registri f bitid. Kui enne tehet oli registri f väärtus 00001100, siis pärast invertteerimist on registri f väärtus 11110011.

Süntaks: COMF f, d

- $d=0$ (Invertteerimise tulemus kopeeritakse tööregistrisse w)
- $d=1$ (Invertteerimise tulemus kopeeritakse registrisse f)

Näide: Inverteeri registri 20h väärtus.

comf 0x20, 1

Tehtega kasutatavad lipud: Z

Loogikakäsud

Loogikakäsud on mõeldud registrisisude ja arvude võrdlemiseks ning loogikateheteks.

ANDWF

ANDWF käsuga võrreldakse tööregistri w ja registri f väärtuseid loogikatehtega “JA” (AND). Kui võrreldava registri vastav bitt = 1 **ja** tööregistri vastava järjekorranumbriga bitt = 1, siis vastuse vastav bitt = 1.

Süntaks: ANDWF f, d

- d=0 (Võrdluse tulemus kopeeritakse tööregistrisse w)
- d=1 (Võrdluse tulemus kopeeritakse registrisse f)

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldava registri väärtus on 7 (00000111), siis tehte tulemusena saadakse 2 (00000010).

0	0	0	0	1	0	1	0	- W	} AND
0	0	0	0	0	1	1	1	- f	
0	0	0	0	0	0	1	0	- Tulemus	

Joonis 1.2.2.1 Tööregistri W ja registri f “AND” võrdlus

Näide: Võrdle aadressil 20h asuva registri sisu tööregistriga.

andwf 0x20, 1

Tehtega kasutatavad lipud: Z

ANDLW

ANDLW käsuga võrreldakse arvu l ja tööregistri w väärtuseid loogikatehtega “JA” (AND). Kui võrreldava arvu vastav bitt = 1 **ja** tööregistri vastava järjekorranumbriga bitt = 1, siis vastuse vastav bitt = 1.

Süntaks: ANDLW k

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldav arv on 7 (00000111), siis tehte tulemusena saadakse 2 (00000010).

0	0	0	0	1	0	1	0	- W	} AND
0	0	0	0	0	1	1	1	- L	
0	0	0	0	0	0	1	0	- Tulemus	

Joonis 1.2.2.2 Tööregistri W ja arvu L “AND” võrdlus

Näide: Võrdle arvu 7 tööregistriga.

andlw 0x07

Selle tehtega kasutatavad lipud: Z

IORWF

IORWF käsuga võrreldakse tööregistri w ja registri f väärtuseid loogikatehtega “VÕI” (OR). Kui võrreldava registri vastav bitt = 1 **või** tööregistri vastava järjekorranumbriga bitt = 1 **või** mõlemate võrreldavate registrite vastavad bitid on võrdsed ühega, siis vastuse vastav bitt = 1.

Süntaks: IORWF f, d

- d=0 (Võrdluse tulemus kopeeritakse tööregistrisse w)
- d=1 (Võrdluse tulemus kopeeritakse registrisse f)

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldava registri väärtus on 7 (00000111), siis tehte tulemusena saadakse 15 (00001111).

0	0	0	0	1	0	1	0	- W	} OR
0	0	0	0	0	1	1	1	- f	
0	0	0	0	1	1	1	1	- Tulemus	

Joonis 1.2.2.3 Tööregistri W ja registri f “OR” võrdlus

Näide: Võrdle aadressil 20h asuva registri sisu tööregistriga.

iorwf 0x20, 1

Tehtega kasutatavad lipud: Z

IORLW

IORLW käsuga võrreldakse arvu l ja tööregistri w väärtuseid loogikatehtega “VÕI” (OR). Kui võrreldava registri vastav bitt = 1 **või** tööregistri vastava järjekorranumbriga bitt = 1 **või** mõlemate võrreldavate registrite vastavad bitid on võrdsed ühega, siis vastuse vastav bitt = 1.

Süntaks: IORLW k

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldav arv on 7 (00000111), siis tehte tulemusena saadakse 2 (00000010).

0	0	0	0	1	0	1	0	- W	} OR
0	0	0	0	0	1	1	1	- L	
0	0	0	0	1	1	1	1	- Tulemus	

Joonis 1.2.2.4 Tööregistri W ja arvu L “OR” võrdlus

Näide: Võrdle arvu 7 tööregistriga.

iorlw 0x07

Selle tehtega kasutatavad lipud: Z

XORWF

XORLW käsuga võrreldakse arvu l ja tööregistri w väärtuseid loogikatehtega välistav VÕI (Exclusive OR). Kui võrreldava registri vastav bitt = 1 **või** tööregistri vastava järjekorranumbriga bitt = 1 **aga mitte mõlema** võrreldava registri vastavad bitid on võrdsed ühega, siis vastuse vastav bitt = 1.

Süntaks: IORWF f, d

- d=0 (Võrdluse tulemus kopeeritakse tööregistrisse w)
- d=1 (Võrdluse tulemus kopeeritakse registrisse f)

Süntaks: XORWF f, d

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldava registri väärtus on 7 (00000111), siis tehte tulemusena saadakse 13 (00001101).

0	0	0	0	1	0	1	0	- W	} XOR
0	0	0	0	0	1	1	1	- f	
0	0	0	0	1	1	0	1	- Tulemus	

Joonis 1.2.2.5 Tööregistri W ja registri f “XOR” võrdlus

Näide: Võrdle aadressil 20h asuva registri sisu tööregistriga.

xorwf 0x20, 1

Selle tehtega kasutatavad lipud: Z

XORLW

XORLW käsuga võrreldakse arvu l ja tööregistri w väärtuseid loogikatehtega välistav VÕI (Exclusive OR). Kui võrreldava registri vastav bitt = 1 **või** tööregistri vastava järjekorranumbriga bitt = 1 **aga mitte mõlema** võrreldava registri vastavad bitid on võrdsed ühega, siis vastuse vastav bitt = 1.

Süntaks: XORLW k

Näiteks: kui tööregistri väärtus on 10 (00001010) ja võrreldav arv on 7 (00000111), siis tehte tulemusena saadakse 13 (00001101).

0	0	0	0	1	0	1	0	- W	} XOR
0	0	0	0	0	1	1	1	- L	
0	0	0	0	1	1	0	1	- Tulemus	

Joonis 1.2.2.6 Tööregistri W ja arvu L “XOR” võrdlus

Näide: Võrdle arvu 7 tööregistriga.

```
xorlw 0x07
```

Selle tehtega kasutatavad lipud: Z

1.2.3 Käsud binaartehteks, muud käsud

Käsud binaartehteks

BSF

BSF käsuga antakse registri f bitile nr. b väärtus 1.

Süntaks: BSF f, b

Näide: Omista aadressil 20h olevasse registrisse salvestatud arvu 2. bitile väärtus 1. Enne tehet oli registri väärtus 0 (00000000), pärast tehet on registri väärtus 2 (00000010).

```
bsf 0x20,2
```

BCF

BCF käsuga antakse registri f bitile nr. b väärtus 0.

Süntaks: BCF f, b

Näide: Omista aadressil 20h olevasse registrisse salvestatud arvu 2. bitile väärtus 0. Enne tehet oli registri väärtus 2 (00000010), pärast tehet on registri väärtus 0 (00000000).

bcf 0x20,2

BTFSC

BTFSC käsuga kontrollitakse registri f bitti nr. b ning hüpatakse järgmisest käsust üle juhul kui $b = 0$. (Bit Test F, Skip if Clear).

Süntaks: BTFSC f, b

Näide: Kontrolli aadressil 20h oleva registri väärtust. Juhul kui registri esimese biti väärtus võrdub ühega kutsuda välja alaprogramm TEST1, vastupidisel juhul kutsuda välja alaprogramm TEST2 .

btfsc 0x20, 1

call TEST1

call TEST2

BTFSS

BTFSS käsuga kontrollitakse registri f bitti nr. b ning hüpatakse järgmisest käsust üle juhul kui $b = 1$. (Bit Test F, Skip if Set).

Süntaks: BTFSS f, b

Näide: Kontrolli aadressil 20h oleva registri väärtust. Juhul kui registri esimese biti väärtus võrdub ühega kutsuda välja alaprogramm TEST2, vastupidisel juhul kutsuda välja alaprogramm TEST1 .

btfsc 0x20, 1

call TEST1

call TEST2

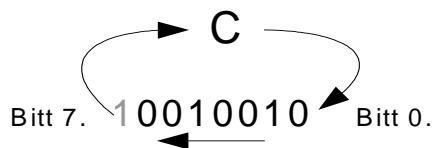
RLF

RLF käsuga antakse registri f esimesele bitile “carry” lipu väärtus, kõiki registri f bitte nihutatakse ühe positsiooni võrra vasakule ning “carry” lipule antakse viimase biti väärtus.

Süntaks: RLF f, d

- $d=0$ (Tehte tulemus kopeeritakse tööregistrisse w)
- $d=1$ (Tehte tulemus kopeeritakse registrisse f)

Näiteks: enne nihutamistehet oli registri f väärtus 10010010 ja “C = 0, siis pärast tehet on registri väärtuseks 00100100 ja C = 1.



Joonis 1.2.3.1 registri f bittide nihutamine vasakule.

Näide: Nihuta aadressil 20h oleva registri kõiki bitte 1 positsiooni võrra vasakule.

rlf 0x20, 1

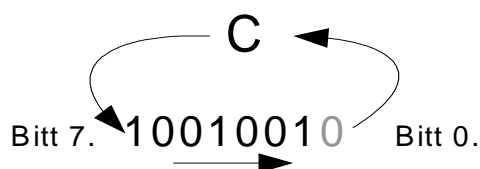
RRF

RRF käsuga antakse registri f viimasele bitile “carry” lipu väärtus, kõiki registri f bitte nihutatakse ühe positsiooni võrra paremale ning “carry” lipule antakse esimese biti väärtus.

Süntaks: RRF f, d

- d=0 (Tehte tulemus kopeeritakse tööregistrisse w)
- d=1 (Tehte tulemus kopeeritakse registrisse f)

Näiteks: enne nihutamistehet oli registri f väärtus 10010010 ja “C = 0, siis pärast tehet on registri väärtuseks 01001001 ja C = 0.



Joonis 1.2.3.2 registri f bittide nihutamine paremale.

Näide: Nihuta aadressil 20h oleva registri kõiki bitte 1 positsiooni võrra paremale.

rrf 0x20, 1

SWAPF

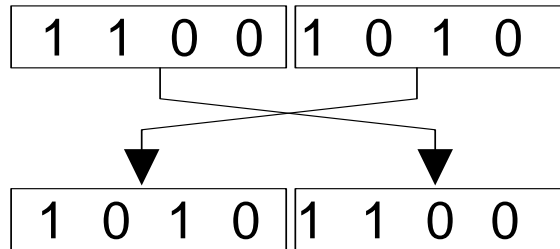
SWAPF käsuga vahetatakse omavahel registri f esimesed ja viimased 4 bitti.

Süntaks: SWAPF f, d

- d=0 (Tehte tulemus kopeeritakse tööregistrisse w)

- d=1 (Tehte tulemus kopeeritakse registrisse f)

Näiteks: enne vahetust oli registri f väärtus 11001010, pärast 10101100.



Joonis 1.2.3.3 registri f bittide nihutamine paremale.

Näide: Vaheta aadressil 20h oleva registri 4 esimese ja viimase biti asukohad omavahel.

swapf 0x20, 1

Muud käsud

NOP

NOP käsuga peatatakse programmi täitmine üheks tsükliks (No operation). Kasutatakse lühemate viivituste saavutamiseks.

Süntaks: NOP

Näide: Tekita 400 ns viivitus. Mikrokontrolleri taktsagedus on 20 MHz.

NOP

NOP

SLEEP

SLEEP käsuga peatub mikrokontrolleri programm ning voolu tarbimine väheneb. Mikrokontroller "äratatakse" kindlaksmääratud katkestuste abil (katkestused vt peatükk 6.7).

Süntaks: SLEEP

Näide: Lülita mikrokontroller patareide säästmise eesmärgil välja kuni järgmise nupulevajutuse poolt esilekutsutud katkestuseni.

SLEEP

Selle tehtega kasutatavad lipud: TO, PD (TO ja PD lipud on bitid STATUS registris, TO – time out, PD power down) TO=1 ja PD=0

CLRWDT

CLRWDT käsuga nullitakse valveloendurit. Juhul kui valveloenduri väärtust ei nullita kindla aja vältel, mikrokontroller taaskäivitatakse.

Süntaks: CLRWDT

Selle tehtega kasutatavad lipud: TO, PD (TO ja PD lipud on bitid STATUS registris, TO – time out, PD power down) TO=1 ja PD=1