

## Övningsuppgifter – STYRNING - i Mikrodator teknik för U2 2010

### 1. Uppgift M10 (5p) aug-09

Skriv ett program i PIC-assembler som

- gör PortB till utport
- ettställer bit 0, 3 och 4 i PortB
- nollställer bit 5 och 6 i PortB
- inverterar bit 1 och 2 i PortB

Bit 7 i PortB ska vara oförändrad. Använd de logiska instruktionerna AND, IOR, COM eller XOR för bitmanipuleringen.

### 2. Uppgift M11 (5p) aug-09

Beskriv övergripande vad vidstående program utför.  
Ange därefter PortB:s värde om  $(PortC) = 75_{16}$ .

```
org 0
goto start

init    clrfs PORTB
        bsf  STATUS, RP0
        clrfs TRISB
        movlw 0F
        movwf TRISC
        bcf  STATUS, RP0
        retlw 0

L0      movlw 0x01
        movwf PORTB
        retlw 0
L1      movlw 0x02
        movwf PORTB
        retlw 0
L2      movlw 0x03
        movwf PORTB
        retlw 0
L3      movlw 0x04
        movwf PORTB
        retlw 0

start   call  init
loop    btfsc PORTC, 0
        call  L0
        btfsc PORTC, 1
        call  L1
        btfsc PORTC, 2
        call  L2
        btfsc PORTC, 3
        call  L3
        goto loop

end
```

**3. Uppgift M12 (4p) aug-09**

Skriv ett väl kommenterat program i PIC-assembler, som bestämmer hur många bitar i minnescell (filregister) 0x0020 som antar värdet 1. Placera resultatet i minnescell (filregister) 0x0021.

**4. Uppgift M13 (8p) aug-09**

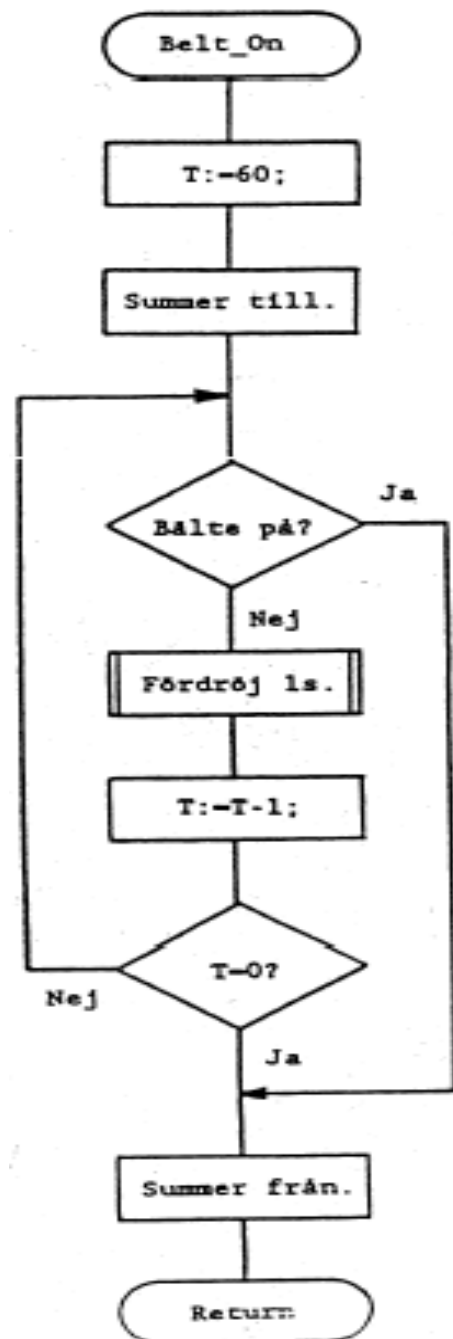
I ett mikrodatorstyrts elsystem i en bil finns en summer, som uppmanar föraren att ta på sig säkerhetsbältet. I elsystemet finns också en givare, som indikerar om bältet satts på.

Då föraren sätter sig i sätet anropas subrutinen BeltOn enl. vidstående flödesschema. Summern aktiveras genom att skriva 0 i bit 3 (RB3) på PortB och stängs av genom att skriva 1 i bit3 (RB3). Givaren är kopplad till bit0 (RC0) på PortC och ger värdet 1 om bältet har satts på, och värdet 0 om bältet ej satts på.

Till hjälp finns en färdig subrutin, DelB, som skapar en fördröjning på 1 sek.

Skriv subrutinen BeltOn (väl kommenterad) i PIC-assembler.

Om föraren ej har tagit på sig säkerhetsbältet inom en minut skall summern stängas av och återhopp ske.



**5. Uppgift M11 (4p) april-09**

Skriv ett väl kommenterat program i PIC-assembler som jämför talen i adresserna TAL1 och TAL2, och lägger det största av dem i adress MAX. Varje tal utgöres av en byte.

Ledning: Använd SUBWF f,d som ger C=1 om differensen  $\geq 0$  och C=0 om differensen  $< 0$ .  
C - carryflaggan

**6. Uppgift M12 (5p) april-09**

Förklara **övergripande** vad vidstående program utför.

Ange också vad innehållet blir i filregister SUMM efter exekvering om från början innehållen i filregistren DELS1 = 0xCA, DELS2=0x21 resp. DELS3=0xDF. Talen anges på 2-komplementsform. Motivera väl.

|       |       |         |
|-------|-------|---------|
| MAIN  | CLRW  |         |
|       | BTFSS | DELS1,7 |
|       | GOTO  | NEXT1   |
|       | COMF  | DELS1,1 |
|       | INCF  | DELS1,1 |
| NEXT1 | ADDWF | DELS1,W |
|       | BTFSS | DELS2,7 |
|       | GOTO  | NEXT2   |
|       | COMF  | DELS2,1 |
|       | INCF  | DELS2,1 |
| NEXT2 | ADDWF | DELS2,W |
|       | BTFSS | DELS3,7 |
|       | GOTO  | NEXT3   |
|       | COMF  | DELS3,1 |
|       | INCF  | DELS3,1 |
| NEXT3 | ADDWF | DELS3,W |
|       | MOVWF | SUMM    |
| STOP  | GOTO  | STOP    |

**7. Uppgift M14 (7p) april-09**

Skriv ett väl kommenterat program i PIC-assembler för en mät databearbetare. Data ligger samlat i en tabell med start på symboliska adressen TAB. Antal värden är 8.

Programmet ska läsa av en tryckknapp (aktivt låg), som är ansluten till PortC bit0 (RC0).

Först när denna knapp blivit nedtryckt (=0) startas inhämtningen av data. Bearbetningen av data görs i subrutinen CALC. Efter bearbetningen skickas resultatet av databehandlingen ut på PortB m h a subrutinen SEND.

a) Skriv huvudprogrammet samt alla nödvändiga initieringar som behövs för att få programmet att fungera.

b) Skriv subrutinen CALC som summerar talen i tabellen (summan antas mindre än 256) och placerar summan på adress 0x25.

c) Skriv subrutinen SEND som hämtar data på adress 0x25, skickar ut data på PortB (utport) samt tänder en lysdiod ansluten till PortC bit6 (RC6).

|     |       |       |    |
|-----|-------|-------|----|
| TAB | ADDWF | PCL,1 |    |
|     | RETLW | 0X05  | ;0 |
|     | RETLW | 0X06  | ;1 |
|     | RETLW | 0X05  | ;2 |
|     | RETLW | 0X06  | ;3 |
|     | RETLW | 0X05  | ;4 |
|     | RETLW | 0X06  | ;5 |
|     | RETLW | 0X05  | ;6 |
|     | RETLW | 0X06  | ;7 |

**8. Uppgift M11 4 poäng jan-09**

I ett 8-bitars tal är **en** bit 1, alla övriga 0, och talet finns i filregister TAL (adress 0x20). Skriv ett väl kommenterat program i PIC-assembler, som i filregister POS1 (adress 0x21), efter exekvering anger ettans bitposition som ett tal (0 – 7).

Ex.           0010 0000 → (POS1) = 5  
              0000 0100 → (POS1) = 2

**9. Uppgift M12 5 poäng jan-09**

Vad utför nedanstående program? Beskriv övergripande (inte instruktion för instruktion) vad programmet gör. Ange också innehållet i filregistret A1, B1 och C1 efter exekvering.

Talen i tabellen anges på 2-komplementsform.

```

LIST          P=16F877A
INCLUDE       <P16F877A.INC>

A1            EQU      0X20
B1            EQU      0X21
C1            EQU      0X22

START        ORG      0X00
             CLRF     A1
             CLRF     B1
             CLRF     C1
             CLRW

LOOP         CALL     TAB
             MOVWF    A1
             SUBLW   0
             BTFSS   STATUS,Z
             GOTO    TEST
             GOTO    SLUT
TEST        MOVF     A1,W
             ANDLW   0X80
             SUBLW   0
             BTFSC  STATUS,Z
             GOTO    NEXT
             INCF   B1,1
NEXT        INCF     C1
             MOVF   C1,W
             GOTO   LOOP
SLUT        GOTO    SLUT

TAB         ADDWF   PCL,1
             RETLW  0XD7
             RETLW  0XF3
             RETLW  0X1A
             RETLW  0X0A
             RETLW  0XCA
             RETLW  0X00
             END

```

**10. Uppgift M14 5 poäng jan-09**

Till PORTB ansluts 3 lysdioder och en knapp. Lysdioderna ansluts till RB5-RB3 och knappen till RB0. Då knappen aktiveras ska ett avbrott av typen RB0/INT genereras och lysdioderna inkrementeras (räknas upp binärt 0-7 för att sen börja om). För varje aktivering av knappen ska man räkna upp ett steg. Övriga bitar ska ej påverkas.

a) Skriv ett väl kommenterat program i PIC-assembler som initierar PORTB och aktiverar RB0/INT avbrott. Skriv också ett huvudprogram som endast ska utgöras av en oändlig loop (vilket gör att inga registervärden behöver sparas undan i avbrottsrutinen enl. b)). Initieringen antas börja på adress 0x0010 och huvudprogrammet följer omedelbart därefter. Bilaga: STATUS- och INTCON-registren

b) Skriv den kompletta avbrottsrutinen som inkrementerar lysdioderna. Kvittera avbrott.

c) Till vilken adress går processorn då den får avbrottsbegäran?  
Skriv in en lämplig instruktion på denna adress.

**11. Uppgift M10 (4p) dec-07**

Betrakta följande programkod:

|       |        |      |       |
|-------|--------|------|-------|
|       | ORG    |      | 0X00  |
|       | GOTO   |      | START |
|       | ORG    |      | 0X10  |
| START | MOVLW  | 0X17 |       |
|       | CALL   |      | ASUB  |
| STOP  | GOTO   |      | STOP  |
|       | ORG    |      | 0XFF  |
| ASUB  | XORLW  | 0XFF |       |
|       | RETURN |      |       |

Förklara vad programmet utför och ange vad W-registret innehåller efter det att programmet exekverats.

**12. Uppgift M11 (6p) dec-07**

a) Initiera PortB efter följande hårdvaruritning. PortB är kopplad till följande enheter:

|        |        |      |      |        |      |      |      |
|--------|--------|------|------|--------|------|------|------|
| Knapp0 | Knapp1 | LED0 | LED1 | Knapp2 | LED2 | LED3 | LED4 |
| RB7    | RB6    | RB5  | RB4  | RB3    | RB2  | RB1  | RB0  |

b) Skriv kod för att läsa av knapp2. Om knapp2 är nedtryckt (aktiv låg) så ska LED2 och LED4 tändas, utan att påverka resten av registret. Om knapp2 ej är nedtryckt ska avläsningen av knapp2 upprepas.

**13. Uppgift M12 (5p) dec-07**

Skriv två program för att kontrollera statusen på en ingång IN7 ansluten till RB7 pinnen och visar IN7:s status på en LED ansluten till RB1 pinnen (PortB). De två programmen skall implementeras så att:

a) IN7 kontrolleras med polling

b) IN7 kontrolleras med avbrott (interrupt).

Bilaga: INTCON-registret

**14. Uppgift M13 (5p) dec-07**

Skriv ett program som hittar det största talet (binär representation) i TABELL. Placera det största talet på adressen MAX.

|        |       |       |                  |    |
|--------|-------|-------|------------------|----|
| TABELL | ADDWF | PCL,1 | ;Positionsnummer |    |
|        | RETLW | 0X12  |                  | ;0 |
|        | RETLW | 0X32  |                  | ;1 |
|        | RETLW | 0X33  |                  | ;2 |
|        | RETLW | 0X22  |                  | ;3 |
|        | RETLW | 0X10  |                  | ;4 |
|        | RETLW | 0X63  |                  | ;5 |
|        | RETLW | 0X42  |                  | ;6 |
|        | RETLW | 0X65  |                  | ;7 |
|        | RETLW | 0X34  |                  | ;8 |
|        | RETLW | 0X11  |                  | ;9 |

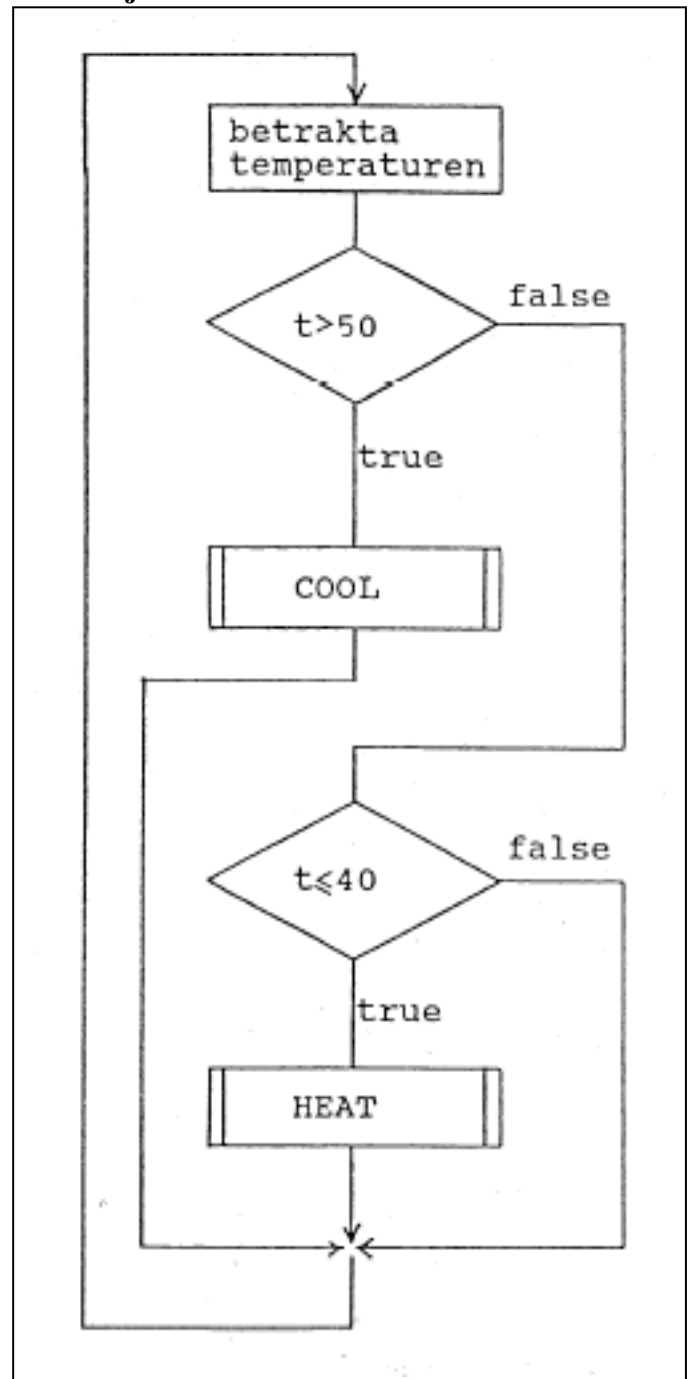
**15. Uppgift M12 (5p) maj-07**

Skriv en väl kommenterad instruktionssekvens i PIC-assembler, som upprepat övervakar temperaturvärdet på adress 0x20.

Om temperaturen är högre än 50°C skall subrutinen COOL exekveras och om den är lägre än eller lika med 40°C skall subrutinen HEAT exekveras (allt enligt flödesplanen).

Temperaturvärdet är ett binärt heltal.  
Subrutinerna COOL och HEAT behöver ej skrivas.

Bilaga: Status register



**Lösningförslag:**

1.            **UppgiftM10    aug-09**

```

INIT            BSF            STATUS,RP0
                 CLRF            TRISB                            ;PORTB utport
                 BCF            STATUS,RP0
MAIN            MOVLW        0X19                            ;1-ställer RB0,3 o. 4
                 IORWF        PORTB
                 MOVLW        0X9F                            ;0-ställer RB5 o. 6
                 ANDWF        PORTB
                 MOVLW        0X06                            ;inverterar RB1 o. 2
                 XORWF        PORTB
SLUT            GOTO            SLUT
  
```

2.            **UppgiftM11    aug-09**

1711.  
5P

PORTB: UTPORT, PORTC: BIT 7-4 UT, BIT 3-0 IN  
(w/ 0-ställas)

RITADNA RC0-RC3 TESTAS, OM =1 LÄGG

|   |    |    |       |    |         |
|---|----|----|-------|----|---------|
| 1 | UT | PS | PORTB | OM | RC0 = 1 |
| 2 | "  | "  | "     | "  | RC1 = 1 |
| 3 | "  | "  | "     | "  | RC2 = 1 |
| 4 | "  | "  | "     | "  | RC3 = 1 |

(PORTC) = 0X95 = 0111 0101 ⇒ RC2 & RC0 = 1 ⇒  
⇒ (PORTB) KORTER XFF ALTERNERA MELLAN 1 & 0

3.            **UppgiftM12    aug-09**

```

X            EQU            0X20
SUM          EQU            0X21
ANTAL        EQU            0X22

                  ORG            0
                  CLRF          SUM                            ;0-ställ SUM
                  MOVLW        8
                  MOVWF        ANTAL                        ;Ladda ANTAL
LOOP          RRF            X,F                            ;Högerskifta X
                  BTFSC        STATUS,C                    ;Kolla C-flaggan
                  INCF          SUM,F                        ;Om C=1, öka SUM
                  DECFSZ        ANTAL,F                    ;Minska ANTAL
                  GOTO          LOOP
SLUT          GOTO          SLUT
                  END
  
```

```

4.      UppgiftM13  aug-09
T      EQU      0X20

INIT   ORG      0
      BSF      STATUS,RP0
      CLRF     TRISB           ;PORTB utport
      MOVLW    0XFF
      MOVWF    TRISC          ;PORTC inport
      BCF      STATUS,RP0

MAIN   CALL     BELTON
      GOTO     MAIN

BELTON MOVLW    0X3C           ;60 till T
      MOVWF    T
      BSF      PORTB,3       ;Aktivera summern
LOOP   BTFSC   PORTC,0       ;Kolla om bältet är på
      GOTO     OFF
      CALL     DELB          ;Vänta 1sek
      DECFSZ  T,F           ;Dekrementera T
      GOTO     LOOP
OFF    BCF      PORTB,3       ;Stäng av summern
      RETURN

DELB   RETURN                ;Fördröjningsrutin

      END

```

```

5.      UppgiftM11  april-09
TAL1   EQU      0X20
TAL2   EQU      0X21
MAX    EQU      0X22

MAIN   ORG      0x00         ;RESETVEKTOR
      MOVF     TAL1,W        ; (TAL1) --> (W)
      SUBWF   TAL2,W        ; (TAL2) - (W) --> (W) OBS! (W) >=0 --> C=1
      BTFSS  STATUS,C      ;KOLLA C-FLAGGAN (W) < 0 --> C=0
      GOTO     MAX1
MAX2   MOVF    TAL2,W        ; (TAL2) --> (W)
      MOVWF   MAX           ; (W) --> (MAX)
      GOTO     SLUT
MAX1   MOVF    TAL1,W        ; (TAL1) --> (W)
      MOVWF   MAX           ; (W) --> (MAX)
SLUT   GOTO     SLUT
      END

```



7. UppgiftM14 april-09

```

ANTAL EQU 0X20
SUM EQU 0X25

INIT ORG 0x00 ;RESETVEKTOR
      BSF STATUS,RP0 ;Bank1
      CLRF TRISB ;PortB UT, RC6 UT
      BCF TRISC,6
      BCF STATUS,RP0 ;Bank0
      CLRF PORTB
      CLRF PORTC

MAIN CALL CALC
      CALL SEND
STOP B STOP

CALC MOVLW 8 ;8 tabellelement
      MOVWF ANTAL
      CLRF SUM ;0-ställ summan
LOOP DECF ANTAL,1 ;korrigera offsetvärdet
      MOVF ANTAL,W
      CALL TAB
      ADDWF SUM,1 ;summera
      MOVF ANTAL,W
      SUBLW 0 ;kolla om alla tabellvärden summerats
      BTSS STATUS,Z
      GOTO LOOP
      RETURN

SEND MOVF SUM,W
      MOVWF PORTB ;lägg ut summan på PortB
      BSF PORTC,6 ;tänd lysdiod på RC6
      RETURN

TAB ADDWF PCL,1
      RETLW 0X05 ;0
      RETLW 0X06 ;1
      RETLW 0X07 ;2
      RETLW 0X08 ;3
      RETLW 0X09 ;4
      RETLW 0X04 ;5
      RETLW 0X05 ;6
      RETLW 0X03 ;7

      END

```

6. UppgiftM12 april-09

PROGRAMMET SUMMERAR ABSOLUTVÄRDEN AV  
TALEN 1 (DECS1), (DECS2) OCH LÄGGER  
SUMMAN I (SUMM)

(DECS1) = 0XCA = 1100 1010<sub>2</sub> ⇒ 0011 0110 = 0X36  
2-kompl

(DECS2) = 0X29 = 0010 1001 = 0X29

(DECS3) = 0XDF = 1101 1111<sub>2</sub> ⇒ 0010 0001 = 0X21  
2-kompl

∴ (SUMM) = 0X78

8. UppgiftM11 jan-09

|       |       |          |                           |
|-------|-------|----------|---------------------------|
| TAL   | EQU   | 0X20     |                           |
| POS1  | EQU   | 0X21     |                           |
|       | ORG   | 0X00     |                           |
| START | CLRF  | POS1     | ;0-ställ resultatcell     |
| LOOP  | MOVF  | TAL,W    | ;läs in talet             |
|       | ANDLW | 1        | ;maska fram bit0          |
|       | BTFSS | STATUS,Z | ;kolla om 0 eller 1       |
|       | GOTO  | KLAR     | ;1 --> klar               |
|       | INCF  | POS1     | ;0 --> öka på positionsnr |
|       | RRF   | TAL,1    | ;högerskifta talet        |
|       | GOTO  | LOOP     |                           |
| KLAR  | GOTO  | KLAR     |                           |
|       | END   |          |                           |

9. UppgiftM12 jan-09

PROGRAMMET BEHÄNDERAR ANTALET NEGATIVA  
TAL I TABELLEN SÖT AVSLUTAS MED VÄRDET 0.

A1 = 0      R1 = 3      C1 = 5  
TABELLENS    ANTAL    OFFSETVÄRDE  
SLUTVÄRDE    OVRS. TAL

10. UppgiftM14 jan-09

```

ORG 0X00
GOTO INIT

; INTERRUPTVEKTORN
ORG 0X04
GOTO HANDLER

; INITIERING AV PORTB OCH INTERRUPT
INIT    CLRF    PORTB      ; 0-STÄLL PORTB
        BSF    STATUS,RP0 ; BANK1
        MOVLW  0XC7      ; RB7-6 IN, RB5-3 UT, RB4-0 IN
        MOVWF  TRISB
        BCF    STATUS, RP0 ; BANK0
        BSF    INTCON, 4  ; INTE=1, RB0-AVBROTT
        BSF    INTCON, 7  ; GIE=1

; HUVUDPROGRAM: OÄNDLIG LOOP
MAIN    GOTO    MAIN

; INTERRUPTROUTIN: INKREMENTERA LED2-LED0
HANDLER MOVE    PORTB,W      ; LÄS IN PORTB, (PORTB)-->(W)
        ADDLW  0X08        ; RÄKNA UPP MED ETT BIT3 + 1 DVS ADDERA 0X08
        ANDLW  0XB8        ; MASKA BORT BIT6, 7+1=0, EJ 8 !
        MOVWF  PORTB      ; SPARA PORTB, (W)-->(PORTB)
        BCF    INTCON,1    ; 0-STÄLL INTF, TILLÅT NYTT INTERRUPT
        RETFIE
        END

```

12. UppgiftM11 dec-07

```

TEMP    EQU    0X20

        ORG    0x00          ; RESETVEKTOR

; PORTINITIERING
INIT    BSF    STATUS, RP0  ; ÖVERGÅNG TILL BANK1
        MOVLW  0XC8
        MOVWF  TRISB      ; RB7, 6, 3 IN, RB5, 4, 2, 1, 0 UT
        BCF    STATUS,RP0  ; ÅTERGÅNG TILL BANK0

MAIN    MOVE    PORTB,W      ; (W)<--(PORTB)
        ANDLW  0X08        ; MASKA FRAM BIT3
        MOVWF  TEMP        ; (TEMP)<--(W)
        BTFSS  TEMP,3      ; KOLLA BIT3
        B      MAIN        ; BIT3=0 => HOPPA TILL MAIN
        IORLW  0X05        ; BIT3=1 => 1-STÄLL BIT2 o. 0
        MOVWF  PORTB      ; (PORTB)<--(W)

STOP   GOTO    STOP

```

11. UppgiftM10 dec-07

|              |        |            |                          |      |            |
|--------------|--------|------------|--------------------------|------|------------|
| START        | MOVLW  | 0X17       | ; (W) ← 0X17             | 0001 | 0111       |
|              | CALL   | ANRO       |                          |      |            |
| STOP         | GOTO   | STOP       |                          |      |            |
| ANRO         | KORLW  | 0XFF       | ; (W) ← 1110 1000 = 0XEB |      |            |
|              | RETURN |            |                          |      |            |
| INNEBÄR LLET | 1      | KRÖ. REG W |                          |      | INVERTERAS |

14. UppgiftM13 dec-07

```

MAX EQU 0X20
ANTAL EQU 0X21 ; (ANTAL) VÄLJES HÄR TILL 10
TEMP EQU 0X22
ORG 0x00 ; RESETVEKTOR

MAIN MOV LW 0X0A
      MOVWF ANTAL
      CLRF MAX
LOOP  DECF ANTAL, W ; FIXA TILL OFFSETVÄRDET
      CALL TABELL ; ANROPA TABELL
      MOVWF TEMP ; SPARA TABELLVÄRDET I TEMP
      SUBWF MAX, W ; (MAX - (W)) --> (W)
      BTFSC STATUS, C ; KOLLA C, >=0 C=1, <0 C=0
      GOTO HOPP ;
      MOVF TEMP, W ; (TEMP) --> (W)
      MOVWF MAX ; (W) --> (MAX)
HOPP  DECFSZ ANTAL, 1
      GOTO LOOP
SLUT  GOTO SLUT ;

TABELL ADDWF PCL, 1 ; Positionsnummer
        RETLW 0X12 ; 0
        RETLW 0X32 ; 1
        RETLW 0X33 ; 2
        RETLW 0X22 ; 3
        RETLW 0X10 ; 4
        RETLW 0X63 ; 5
        RETLW 0X42 ; 6
        RETLW 0X65 ; 7
        RETLW 0X34 ; 8
        RETLW 0X11 ; 9

```

```

13.          UppgiftM12  dec-07
              ORG      0x00          ;RESETVEKTOR
;PORTINITIERING
INIT        BSF      STATUS, RP0    ;ÖVERGÅNG TILL BANK1
              MOVLW   0XFD
              MOVWF   TRISB         ;RB1 UT, övriga IN
              BCF      STATUS,RP0   ;ÅTERGÅNG TILL BANK0

MAIN        MOVF     PORTB,W        ; (W) <-- (PORTB)
              BTFSC   PORTB, 7     ;KOLLA RB7
              GOTO    ON            ;RB0=1
OFF         ANDLW   0XFD           ;RB7=0, 0-STÄLL RB1
              MOVWF   PORTB        ; (PORTB) <-- (W)
              GOTO    SLUT         ;
ON          IORLW   0X02           ;1-STÄLL RB1
              MOVWF   PORTB        ; (PORTB) <-- (W)
SLUT       GOTO    MAIN
TEMP       EQU     0X20

              ORG      0x00          ;RESETVEKTOR
              GOTO    INIT

              ORG      0X04
              GOTO    HANDLER

              ORG      0X10
;PORTINITIERING
INIT        BSF      STATUS, RP0    ;ÖVERGÅNG TILL BANK1
              MOVLW   0XFD
              MOVWF   TRISB         ;RB1 UT, övriga IN
              BCF      STATUS,RP0   ;ÅTERGÅNG TILL BANK0
;AVBROTTSINITIERING
              BSF     INTCON, 7     ;GIE=1
              BSF     INTCON, 3     ;RBIE=1

MAIN        GOTO    MAIN

HANDLER    MOVF     PORTB,W        ; (W) <-- (PORTB)
              BTFSC   PORTB, 7     ;KOLLA RB7
              GOTO    ON            ;RB0=1
OFF         ANDLW   0XFD           ;RB7=0, 0-STÄLL RB1
              MOVWF   PORTB        ; (PORTB) <-- (W)
              GOTO    SLUT         ;
ON          IORLW   0X02           ;1-STÄLL RB1
              MOVWF   PORTB        ; (PORTB) <-- (W)
SLUT       BCF     INTCON, 0       ;RBIF=0
              RETFIE

```

```
15.          UppgiftM12  maj-07
TEMP        EQU        0X20

                ORG        0X00
START       MOVF        TEMP, W      ;läs in temp.
                SUBLW     0X32      ;50-temp
                BTFSC    STATUS, C   ;>=0 -->C=1, <0 -->C=0
                GOTO     NEXT
                CALL     COOL
                GOTO     START
NEXT        MOVF        TEMP, W
                SUBLW     0X28      ;40 - temp
                BTFSC    STATUS, C
                CALL     HEAT
                GOTO     START

COOL        RETURN
HEAT        RETURN

                END
```