

Ankerkettenzählwerk - PIC

1. Basis Annahmen

Um den Herstellungsaufwand für die Schaltung des Ankerkettenzählwerks verglichen mit der Lösung, welche mit passiven Zählern realisiert wurde, drastisch zu minimieren, wird hier eine aktive Lösung unter Verwendung des Microcontrollers PIC 16F627A gezeigt. Alle Forderungen im Bezug auf die Funktionsweise der Schaltung als auch alle Annahmen über die Kennwerte der Ankerwisch bleiben unverändert.

2. Diskretes Interface und PSU (Ref. Blatt 2)

Die Eingangssignale vom Reedsensor und für das „UP“ Kommando werden über Relais und nachfolgenden Entprellschaltungen geführt.

2.1. Reedsensor Signalverarbeitung

Aus Stabilitätsgründen und um Unabhängigkeit von der Impulsbreite des Reedsensorsignals garantieren zu können, wird als Beginn der Erfassung der Reedsensorimpulsen ihre abfallende Flanke vom Mikrokontroller verwendet.

2.2. PSU

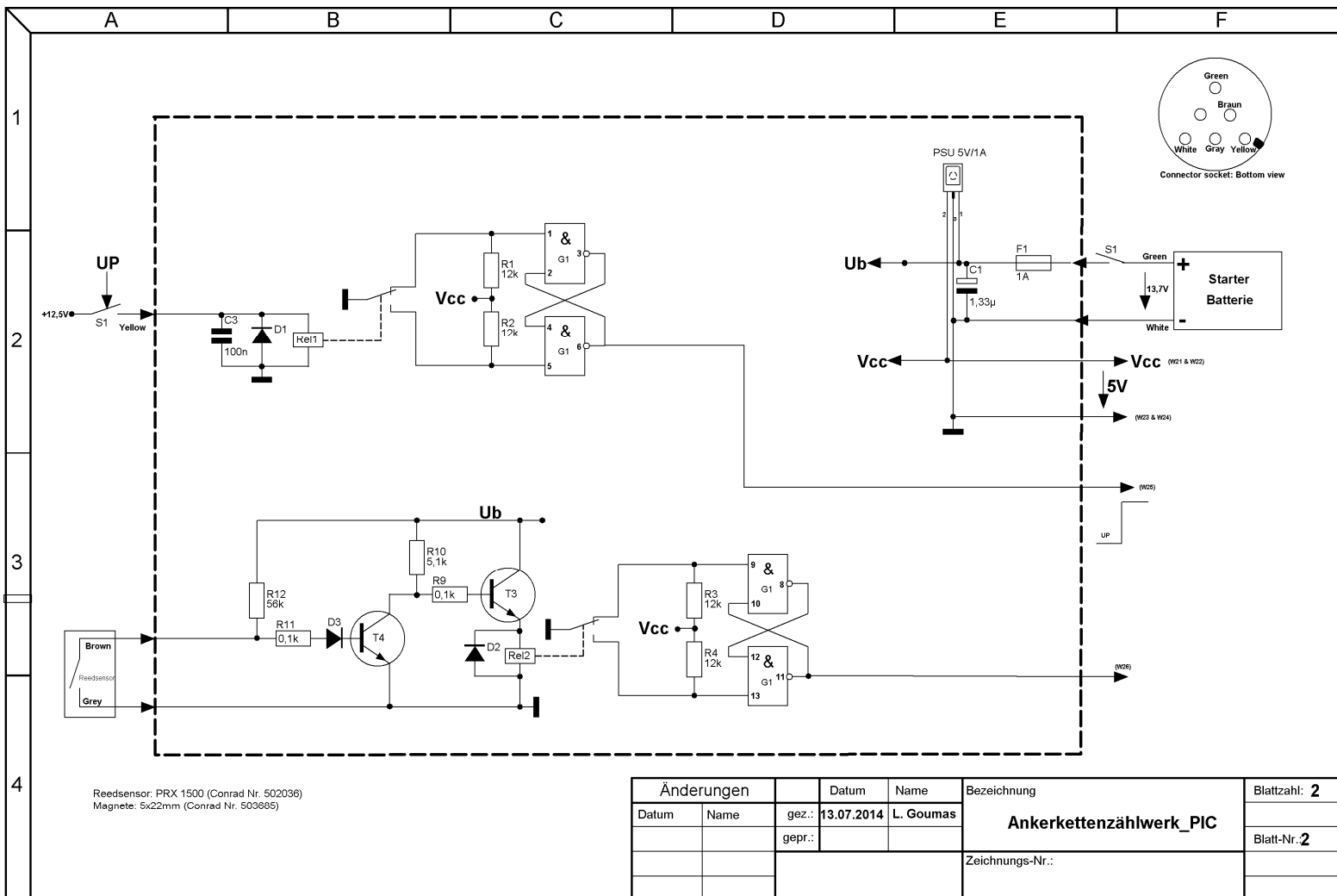
Als PSU wird der Motorola Regler 7805C verwendet. Für die Leistungsbilanz müssen in worst case das Brennen von 3x7 LED- Segmenten a 20mA, die maximal zulässige Verlustleistung des PIC von 800mW sowie ein zusätzlicher Strombedarf für die Peripherie von etwa 100 mA berücksichtigt werden. Dies führt zu einem maximalen Strombedarf von:

$$3*7*23\text{mA} + 0,8\text{W}/5\text{V} + 100\text{mA} = 743\text{mA}.$$

Erlaubt man einen maximalen Strom von 0,8A, so beträgt die maximale Verlustleistung am Regler, bei einer maximalen Eingangsspannung von 14V, $P_{\text{max}} = (14\text{V} - 5\text{V}) * 0,8\text{A} = 7,2\text{W}$. Für das TO220 Gehäuse des Reglers gibt der Hersteller eine maximal zulässige Sperrschichttemperatur von 150 °C und einen Wärmeübergangswiderstand zwischen Sperrschicht und Gehäuse von 5 °C/W an. Geht man von einer maximalen Umgebungstemperatur von 50 °C aus, so muss der Regler auf einen Kühlkörper mit folgendem maximalen Wärmeübergangswiderstand montiert werden:

$$R_{\theta ca} < \frac{\theta_j - \theta_a}{P_{\text{max}}} - R_{\theta jc} < \frac{150^\circ\text{C} - 50^\circ\text{C}}{7,2\text{W}} - 5^\circ\text{C}/\text{W} < 8,89^\circ\text{C}/\text{W}$$

Gewählt wird der Fischer Elektronik Kühlkörper FK 202 SA-CB mit einem Wärmeübergangskoeffizient von 8°C/W (Conrad 187925)



Ankerkettenzählwerk - PIC

3. Microcontroller und Kettenlänge Anzeige (Ref. Blatt 1)

3.1. Anzeige der Kettenlänge

Um die Tageslichtlesbarkeit der Anzeige verbessern zu können, werden hierbei Anzeigenelemente des Typs SA10-21GWA eingesetzt, welche 2 grüne LEDs pro Segment verwenden. Jedes Segment wird dann mit mindestens 20mA beaufschlagt.

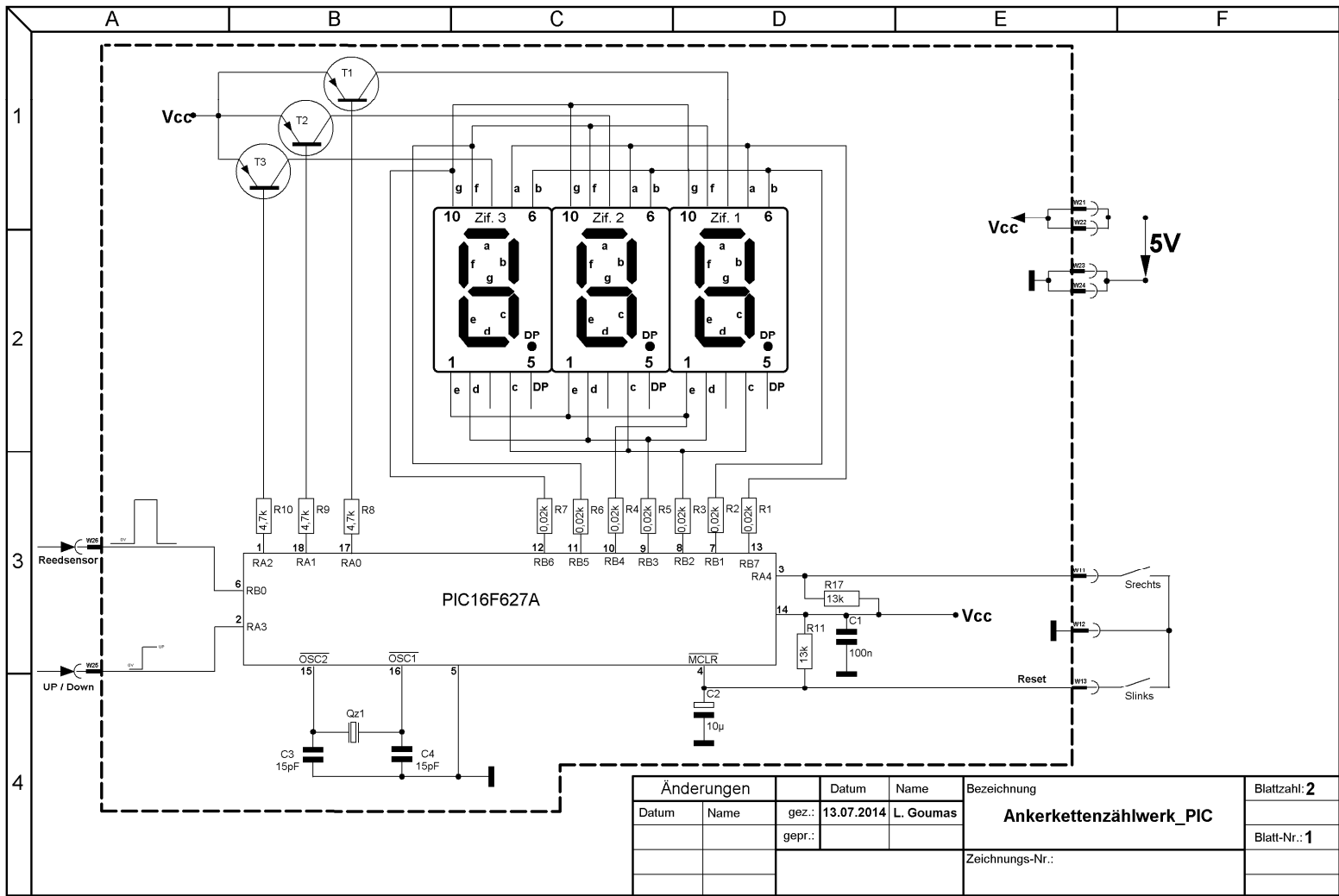
Um sicher zu stellen, dass das Flimmern der Anzeige möglichst gering gehalten werden kann, erfolgt die Ansteuerung der LEDs über Interrupts im Zyklus von 8msec. Damit beträgt die Update rate der Anzeige 24 msec, entsprechend 42 Hz.

3.2. PIC 16F627A

Es handelt sich hierbei um einen 8 bit Microcontroller der Firma Microchip, welcher im vorliegenden Fall mit einem Crystaloszillator von 2,048 MHz im XT- mode betrieben wird.

Da der Pin **RB4** als digitales I/O verwendet werden soll, muss der „low voltage programming mode“ ausgeschaltet werden, d.h. in den „Assembler directives“ muss unter den Konfigurationsangaben auch „**LVP_OFF**“ aufgenommen werden.

Schließlich muss auf Grund der Tatsache, dass die Pins **RA<3:0>** ebenfalls als digitale I/O's verwendet werden sollen, darauf geachtet werden, dass die bits **CM2: CM0** im „Special Function Register CMCON“ der BANK0 auf „**High**“ gesetzt und damit ihre Benutzung als Komparatoreingänge ausgeschlossen wird.



Änderungen		Datum	Name	Bezeichnung	Blattzahl: 2
Datum	Name	gez.: 13.07.2014	L. Goumas	Ankerkettenzählwerk_PIC	Blatt-Nr.: 1
		gepr.:			
				Zeichnungs-Nr.:	

Ankerkettenzählwerk - PIC

4. ASM-Listings

```
;KETTENWERK.asm Kettenzählwerk mit 3-Stellige 7-Segment Anzeige
;RA<3:0> Digital I/O d.h. CM<2:0> in CMCON auf"111"
;RA<2:0> LED Anoden / Wenn RA3=0 Kette "Down" sonst "Up"
;RA4. LED Test (888)
;RB<7:1> LEG Kathoden
;RB0. Reedsensoreingang (External interrupt)
;RB4. digitales I/O d.h. "LVP_OFF"
;2,048 MHZ / 1,953125 µs (Osc > 500kHz d.h. XT_OSC)
;Prescaler 16 / TMRO = 8 ms / LED update 42Hz (3x8=24mSec)
;*****

;Assembler directives
list,P16F627
#include <p16F627A.inc> ;Controller header file
_CONFIG_CP_OFF & _PWRTE_OFF & _WDT_OFF & _XT_OSC & _LVP_OFF
ERRORLEVEL -302 ;Supress Bank SELECTION MESSAGES
;*****

;Zuordnung der verwendeten Register

w_copy equ 0x20 ;saved w register
s_copy equ 0x21 ;saved STATUS register
countdown equ 0x22 ;Trommelzählregister "Down"
countup equ 0x23 ;Trommelzählregister "Up"
down equ 0x24 ;Kettenlängezählregister "Down"
up equ 0x25 ;Kettenlängezählregister "Up"
chain equ 0x26 ;Deployed chain
value equ 0x27 ;Wertigkeitskennziffer
display equ 0x28 ;Anzuzeigende Zahl
;*****

;Ziele der Register Operationen

w equ 0 ;w=Zielregister
f equ 1 ;f=Zielregister
;*****

;Macro- Definitionen

bank0 macro
    bcf STATUS, 5
    bcf STATUS, 6
endm

bank1 macro
    bsf STATUS, 5
endm
;*****

    org 0x00 ;Programmbeginn bei Adresse 00h
    goto start
;*****

    org 0x04 ;Interrupt service Adresse

;Interrupt service vector

    bcf INTCON, 7 ;global interrupt disabled
    movwf w_copy ;save w in 0x20
    movf STATUS, 0 ;Verschiebe STATUS in w
    movwf s_copy ;save STATUS in 0x21

;interrupt routine start
```

```

        btfsc  PORTA,3           ;Wurde "Up" kommandiert?
        goto  $+0A
        bcf   STATUS, 2        ;Flag Z löschen
        incf  countdown, 1
        movlw .6
        subwf countdown, 0    ;Countdown Inhalt minus .6 in w
        btfss STATUS, 2      ;ergebnis "Null"?
        goto  intrend        ;Nein
        incf  down, 1         ;Down Register um "1" erhöhen
        clrf  countdown      ;Countdown zurücksetzen
        goto  abs            ;Gesetzte Kettenlänge

        bcf   STATUS, 2        ;Flag Z löschen
        incf  countup, 1
        movlw .6
        subwf countup, 0     ;Countup Inhalt minus .6 in w
        btfss STATUS, 2      ;ergebnis "Null"?
        goto  intrend        ;Nein
        incf  up, 1          ;Up Register um "1" erhöht
        clrf  countup        ;Countup zurücksetzen
abs     bsf   STATUS, 0        ;Flag c auf 1 setzen
        movf  up, 0          ;Inhalt von Up in w
        subwf down, 0        ;Subtrahiere Up von Down
        btfsc STATUS, 0      ;Ergebnis negativ ?
        goto  $+02          ;Nein
        sublw .256
        movwf chain         ;Deployed chain

;Interrupt routine ende

intrend    bcf   INTCON, 1    ;RB0/INT Flag (INTF) cleared

;restore context and return to the main programm

        movf  s_copy, 0;Hole STATUS zurück in w
        movwf STATUS         ;Daten zurück in STATUS
        movf  w_copy, 0      ;Akku Daten sichern
        retfie               ;interrupt ende und GIE enabled
;*****

;Subroutines

;Initialisierung

init      clrf  countdown    ;Trommelzählregister "Down" löschen
        clrf  countup       ;Trommelzählregister "Up" löschen
        clrf  down          ;Down Zählregister löschen
        clrf  up            ;Up Zählregister löschen
        clrf  chain         ;Deployed chain cleared
        clrf  value        ;Wertigkeitskennziffer löschen
        clrf  display      ;Anzuzeigende LED Zahl löschen
        movlw 0x07
        movwf CMCON         ;törn off comparators & enable I/O
        bankl                ;gehe zu bank 1
        movlw B'00011000'
        movwf TRISA        ;RA<2:0> Ausgänge/RA<4:3> Eingänge
        movlw B'00000001'
        movwf TRISB        ;RB<7:1> Ausgänge / RB0 Eingang
        bank0                ;gehe zu bank 0
        movlw .15
        movwf PORTA        ;RA<3:0> auf 1
        movlw .254
        movwf PORTB        ;RB<7:1> auf 1

;TMRO voreinstellen und starten
;TMRO overflow alle 1,953125 x 16 x 256 = 8 ms

```

Ankerkettenzählwerk - PIC

```

bcf     INTCON,      7      ;global interrupt disabled
bank1   ;gehe zu bank 1
bcf     OPTION_REG, 5 ;intern. instruct. cycle clock
bcf     OPTION_REG, 6 ;Interrupt from RB0 trailing edge
bcf     OPTION_REG, 2
bsf     OPTION_REG, 1
bsf     OPTION_REG, 0 ;PS2,PS1,PS0="011"/prescaler 1:16
bcf     OPTION_REG, 3 ;prescaler assigned to TMRO
bank0   ;zurück zu bank 0
return

```

;7 Segment Tabelle

```

segmente addwf PCL, 1 ;Erhöhung von PCL um den Wert von w.
;Programm springt auf eine der
;nächsten 10 Zeilen
retlw   B'01000000' ;0
retlw   B'11111000' ;1
retlw   B'00100100' ;2
retlw   B'00110000' ;3
retlw   B'10011000' ;4
retlw   B'00010010' ;5
retlw   B'10000010' ;6
retlw   B'01111000' ;7
retlw   B'00000000' ;8
retlw   B'00011000' ;9

```

;Hauptprogramm

```

start   call    init

        movlw   B'10010000'
        movwf   INTCON      ;RB0 & GIE enabled/T0IF cleared

loop    clrf    TMRO
        clrf    value      ; Wertigkeit löschen
        incf   chain, 0 ;Deployed chain um "1" erhöht in w
        movwf  display
        decf   display, 1;Copy of deployed chain in display
        bsf   STATUS, 0 ;Flag C setzen
        clrf  value      ;Wertigkeit löschen
        incf  value, 1 ;Wertigkeit um "1" erhöht
        movlw .100
        subwf display, 1
        btsc  STATUS, 0 ;Display negativ?
        goto  $-04 ;Nein
        decf  value, 0 ;Wertigkeit in w
        call  segmente
        movwf PORTB
        bcf   PORTA, 2 ;Anode der LED 3 an
        btfs PORTA, 4 ;RA4 auf "1"
        goto LEDtest3;Nein
        goto  $+04 ;Ja
LEDtest3 movlw .8
        call  segmente
        movwf PORTB
        btfs INTCON, 2 ;TMRO overflowed?
        goto  $-01 ;Nein
        bsf   PORTA, 2 ;Anode der LED 3 "Aus"
        bcf   INTCON, 2 ;TMR0 overflow Flag cleared

        incf  chain, 0 ;Deployed chain um "1" erhöht in w
        movwf display
        decf  display, 1;Copy of deployed chain in display

```

```

    bsf    STATUS, 0      ;Flag C setzen
    movlw  .100
    subwf  display, 1
    btfs   STATUS, 0      ;Display negativ?
    goto   $-03          ;Nein
    movlw  .100
    addwf  display, 1;Wert von display 2-Stellig positiv
    clrf   value         ;Wertigkeit löschen
    bsf    STATUS, 0      ;Flag C setzen
    incf   value, 1 ;Wertigkeit um "1" erhöht
    movlw  .10
    subwf  display, 1
    btfs   STATUS, 0      ;Display negativ?
    goto   $-04          ;Nein
    decf   value, 0 ;Wertigkeit in w
    call   segmente
    movwf  PORTB
    bcf    PORTA, 1      ;Anode der LED 2 an
    btfs   PORTA, 4      ;RA4 auf "1"
    goto   LEDtest2;Nein
    goto   $+04          ;Ja
LEDtest2 movlw  .8
    call   segmente
    movwf  PORTB
    btfs   INTCON, 2     ;TMRO overflowed?
    goto   $-01          ;Nein
    bsf    PORTA, 1      ;Anode der LED 2 "Aus"
    bcf    INTCON, 2     ;TMR0 overflow Flag cleared

    incf   chain, 0 ;Deployed chain um "1" erhöht in w
    movwf  display
    decf   display, 1;Copy of deployed chain in display
    bsf    STATUS, 0      ;Flag C setzen
    movlw  .100
    subwf  display, 1
    btfs   STATUS, 0      ;Display negativ?
    goto   $-03          ;Nein
    movlw  .100
    addwf  display, 1;Wert von display 2-Stellig positiv
    bsf    STATUS, 0      ;Flag C setzen
    movlw  .10
    subwf  display, 1
    btfs   STATUS, 0      ;Display negativ?
    goto   $-03          ;Nein
    movlw  .10
    addwf  display, 1;Wert von display 1-Stellig positiv
    clrf   value         ;Wertigkeit löschen
    bsf    STATUS, 0      ;Flag C setzen
    incf   value, 1 ;Wertigkeit um "1" erhöht
    movlw  .1
    subwf  display, 1
    btfs   STATUS, 0      ;Display negativ?
    goto   $-04          ;Nein
    decf   value, 0 ;Wertigkeit in w
    call   segmente
    movwf  PORTB
    bcf    PORTA, 0      ;Anode der LED 1 an
    btfs   PORTA, 4      ;RA4 auf "1"
    goto   LEDtest1;Nein
    goto   $+04          ;Ja
LEDtest1 movlw  .8
    call   segmente
    movwf  PORTB
    btfs   INTCON, 2     ;TMRO overflowed?
    goto   $-01          ;Nein
    bsf    PORTA, 0      ;Anode der LED 2 "Aus"

```


Ankerkettenzählwerk - PIC

```
bcf    INTCON, 2    ;TMR0 overflow Flag cleared  
goto  loop
```

```
end
```

```
*****
```

