

```

;*****
;
;Repeater programma PI3ZOD Versie: 1.0 van 970326
;
;Org : Johan van Dijk PA3ANG, Emmen 870709   V1.0
;Rev : 870731 PI2ASN                          V1.1
;      871124   ""                            V2.0
;      880409   ""                            V2.1
;      970326 PI3ZOD
;Copyright (C) 1987, 1997 by PA3ANG
;
;
;*****
;
;RIOT bedrading
;PA0 - Tone output      (uit)
; 1 - Watchdog reset   (uit)
; 2 - 1750 Hz dekodeer (in)
; 3 - Receiver squelch(in)
; 4 - Operator switch  (in)
; 5 - Audio switch     (uit)
; 6 - Tone switch      (uit)
; 7 - 1 sec clock      (in)
;
;PB3 - Operator lamp    (uit)
; 4 - Local pcb lamp    (uit)
; 5 - Operator lamp     (uit)
; 6 - Transceiver on    (uit)
; 7 - Squelch off       (uit)
;
;MODES
; 0 - Normaal = 1750 Hz toonslot, 3 / 30 seconden
; 1 - T
;*****
;
;Programma variabelen (assembler)
SECOND =1           ;Duur van seconde in seconden
MINUTE  =60         ;Duur minuut in seconden
SPTIMS  =1          ;Spreektyd kort (min.)
SPTIML  =10        ;Spreektyd lang (min.)
JTIME   =2          ;Storingstyd bij aangaan relais (sec.)
SPEEDC  =200       ;Snelheid morse >> is trager
SPEEDR  =100       ;Snelheid rogerpiep
TONE    =7         ;Toonhoogte >> is lager
BCTIM   =30        ;Baken herhalingstyd (min.)
NCLTIM  =2         ;Binnen x min geen herstart met call
CLTIME  =5         ;Call sequence tyd (max = 8 minuten)
CWMIN   =2         ;Min. tyd carrier voor geven roger beep
MINUT2  =MINUTE/2  ;Duur minuut 2 seconden timer
TRHRLY  =3         ;Threshold tx na squelch dicht (sec.)
CDTIME  =30        ;Closedown tyd (sec.)
ELAPM   =100/2     ;Minimale elapstyd voor long closedown
CWREPT  =3         ;Max. carrier wave TXON (players)
TSTTIM  =180       ;Maximale testtyd in sec. (max = 255)
TSTCLT  =60        ;Call-sequence tyd in teststand
FLTIMS  =4         ;Flutter detektie bij x keer
FLTIM   =2         ;Flutter tyd-gap
SQTIM   =2         ;Squelch off TYD
ALSIGS  =10        ;Aantal waarschuwingsbeeps bij timeout
CMDTIM  =5         ;Tijd dat commando's geveenen mogen worden
;
;System hardware
PA      =$800      ;RIOT
DRA     =PA+1

```

```

PB      =PA+2
DRB     =PA+3
IRQADR  =PA+7
;
;System temporary adress.
;
;Note: System stack from 01FF down
;      Ram from 000-07F (080-0FF etc 32 maal)
TEMP1   =$0           ;General purpose TEMP addresses
TEMP2   =TEMP1+1     ;van CWRTN
TEMP3   =TEMP1+2     ;--
CDOWN1  =TEMP1+3     ;General purpose conter on IRQRTN
CDOWN2  =TEMP1+4     ;--
CDOWN3  =TEMP1+5     ;Elapstyd (aftellen per 2 seconden)
TIM1    =TEMP1+6     ;Beaontyd
TIM2    =TEMP1+7     ;Spreektyd
STIME   =TEMP1+9     ;Spreektyd (volgens mode)
RSTAT1  =TEMP1+10    ;Status : Relais aan
RSTAT2  =TEMP1+11    ;Status : Squelch off
SECFE   =TEMP1+12    ;Seconden flipflop
TEMP4   =TEMP1+13    ;General flaggetje
TEMP5   =TEMP1+14    ;Flutter counter
CDOWN4  =TEMP1+15    ;Testtoon timer
TEMP6   =TEMP1+16    ;Relais open houden DLOOP
SPEED   =TEMP1+17    ;Snelheid morse
MODE    =TEMP1+18    ;Mode relais
CDOWN5  =TEMP1+19    ;Command timer
;
;System initialisation
.ORG $FFFA
.WORD NMIRTN ;Set jumpvektors
.WORD RESET
.WORD IRQRTN
;
;System startup
.ORG $F000
RESET   LDX #$FF      ;Reset stackpointer
        TXS
        CLD
        SEI          ;Disable interrupt
;
;RIOT initialisation
LDA    #%01100011 ;PA port
STA    DRA
LDA    #%11111000 ;PB port
STA    DRB
STA    IRQADR      ;IRQ on PA7 rise
;
;Relais initialisation
LDA    #%01000000 ;Alles uit behalve Tone high
STA    PA          ;Reset tevens de watchdog.
LDA    #%01000000 ;Zet de zender uit
STA    PB
LDA    #SPEEDC     ;Standaard snelheid
STA    SPEED
;
CLI
;
;Start relais programma
RELAIS JSR TXON      ;Meld restart programma
        JSR WAITS     ;Wacht 1 seconde
        LDX #BEACON-CALL
        JSR CWRTN     ;Call
        JSR TXOFF

```

```

;
;De bakenloop.
CLOOP LDA #0
      STA RSTAT1      ;Relais uit
      JSR RSTOFF
      JSR SQON
      JSR AUDOFF
      LDA #BCTIM      ;Hoeveel keer 1 minuut aftellen ?
      STA TIM1
CLOOP1 LDA #MINUTE    ;1 minuut laten aftellen door IRQRTN
      STA CDOWN1
CLOOP2 JSR CH1750
      BEQ ULOOP      ;1750 ontvangen. Zet relais aan.
      JSR OPKEY      ;Operator wil in relais mode?
      BEQ ULOOP
      LDA CDOWN1     ;Minuut uitgeteld ?
      BNE CLOOP2
      DEC TIM1       ;Alle minuten al gehad ?
      BNE CLOOP1     ;Nee dan terug.
      ;Het verzenden van het baken bericht
      JSR TXON       ;Ga bakenmessage verzenden.
      JSR WAITS      ;Ga 1 seconde wachten
      JSR TLOW
      LDX #BEACON-CALL
      JSR CWRTN
      JSR TXOFF
      JMP CLOOP      ;Continueer closedown loop
;
;Aanzetten relais, prepareer de pointer etc.
ULOOP JSR TXON       ;Zet de zender meteen aan
      LDA #1         ;Status relais = aan
      STA RSTAT1
      JSR RSTON      ;Led aan
      JSR JLOOP      ;Wacht even op luisteren tstn.
      JSR AUDON      ;Audio mag aan
      LDA #MINUT2*2
      STA CDOWN3     ;Cw id over 2 minuten
      LDA TIM1       ;Mag relais open zonder call ?
      CMP #BCTIM-NCLTIM
      BCS ULOOPZ     ;Ja,
      JSR WAITS
      LDX #CALL-CALL ;Nee, ga eerst de call sienen
      JSR CWRTN
      LDA #MINUT2*CLTIME
      STA CDOWN3     ;Cw id over 5 minuten
ULOOPZ JSR OPKEY      ;Wil de operator in de testmode ?
      BNE ULOOP3     ;Nee
      JMP TLOOP
ULOOP3 LDA #SPTIMS   ;Korte spreektijd
      STA STIME
      JSR CHSQ       ;Welke status heeft het relais ?
      BNE ULOOP4     ;Kort..
      LDA #SPTIML    ;Lang..
      STA STIME
ULOOP4 LDA #0         ;Init flags and counters
      STA RSTAT2     ;Squelch off flag
      STA TEMP4      ;Play counter
      STA TEMP5      ;flutter counter
;
;De receive loop. Het hoofdprogramma.
RLOOP LDX #1         ;Geen status door aan IRQRTN
      STX RSTAT1     ;CDOWN3 wordt dan afgeteld
      JSR TRIGCK
      LDA #TRHRLY    ;Na x seconden tx-off

```

```

        STA CDOWN1
        ;
        ;De lus bij geen 'ontvang', zender aan.
RLOOP1 LDA CDOWN1
        BNE RLOOPB
        JMP DLOOP      ;Zender uit (carrier operated)
RLOOPB LDA CDOWN3      ;Moet de call tydens het wachten
        BNE RLOOP0
        JMP RLOOPN
RLOOP0 JSR CHSQ
        BEQ RLOOP1
        ;Init ontvang
RLOOP2 LDA STIME      ;Spreektyd in minuten
        STA TIM2
        JSR TRIGCK    ;Trigger voor roger beep delay
        LDA #MINUTE
        STA CDOWN1
        LDA #CMDTIM
        STA CDOWN5
        JMP RLOOPL    ;Sla RLOOPK over
RLOOPK LDA #0
        STA TEMP5     ;Clear flutter counter
RLOOPL LDA #FLTIM     ;Entry na squelch < 1 seconde dicht.
        STA CDOWN2    ;Laad maximale flutter tyd
        ;
        ;De lus bij constante 'ontvang'
RLOOP3 LDA CDOWN2
        BEQ RLOOPK
        JSR OPKEY     ;Wil operator stoppen ?
        BNE RLOOPI    ;Nee,
        JMP DLOOP7    ;Stop relais onmiddellijk met cw-id
RLOOPI LDA CDOWN1     ;Counter uitgeteld ?
        BNE RLOOPD
        DEC TIM2      ;Alle minuten gehad ?
        BNE RLOOP4    ;Ja, wurgen !!
        JMP WLOOP
RLOOP4 LDA #MINUTE    ;Nee, ga volgende minuut in.
        STA CDOWN1
RLOOPD LDA CDOWN3     ;Check call-id time limit
        BNE RLOOPA    ;0 ?
        LDX #CALL-CALL ;ga de call onder de mod. seinen
        JSR CWRTN
        LDA #MINUT2*CLTIME
        STA CDOWN3    ;Reset elaps timer
RLOOPA CMP #MINUT2/2  ;4,5 minuten voorbij sinds cw-id ?
        BCS RLOOP5    ;Nee, doorgaan
        LDA #1        ;Ja,
        STA TEMP3     ;Maak flag 1 (wordt 0 door CWRTN)
RLOOP5 JSR CH1750    ;Wordt er opnieuw gefloten
        BNE RLOOP6    ;Nee, continue RLOOP
        JSR AUDOFF    ;Ja, zet audio uit.
        JSR TRIGCK    ;Wil station in testrtn ?
        LDA #3        ;minstens drie seconden 1750 Hz
        STA CDOWN4
RLOOPY LDA CDOWN4     ;Timer uitgeteld ?
        BNE RLOOPX    ;Nee, doorgaan
        JMP TLOOP     ;Ja, naar testloop
RLOOPX JSR CH1750
        BEQ RLOOPY    ;Blijf afvragen
        LDA CDOWN5
        BEQ RLOOPM
        INC MODE
RLOOPM JSR AUDON
RLOOP6 LDA RSTAT2     ;Staat squelch uit ?

```

```

    BEQ RLOOPE      ;Nee, doorgaan
    LDA CDOWN2     ;Ja, check tydlimit
    BNE RLOOP3     ;Tyd voorbij
    ;Squelch controle
    JSR SQON       ;Tyd voorbij, zet squelch in
    JSR CHSQ       ;Is er nog signaal
    BNE RLOOPH     ;Laat squelch uit staan
    LDA #0         ;Nee, reset counter en flag
    STA TEMP5      ;en ga gewoon verder.
    STA RSTAT2
RLOOPE JSR CHSQ    ;Signaal nog steeds in de lucht ?
    BNE RLOOP3     ;Ja, blijf in de spreektyd loop
    ;Signaal weg
    ;De beslisroutine voor rogerbeep-call
    ;en squelch aan-uit
    JSR TRIGCK     ;Nee, ga roger geven
    INC TEMP5      ;Dedekteer eerst flutter
    LDA TEMP5
    CMP #FLTIMS
    BNE RLOOPF     ;Geen flutter
RLOOPH LDA #SQTIM ;Wel flutter, laad squelch counter
    STA CDOWN2
    STA RSTAT2     ;en zet flaggen
    JSR SQOFF      ;Zet squelch uit
    BNE RLOOPF
    JMP RLOOP3     ;Vervolg programma zonder roger-beep
RLOOPF LDA #SECOND ;Signaal langer dan 1 seconde weg ?
    STA CDOWN2
RLOOP7 JSR CHSQ
    BEQ RLOOPM
    JMP RLOOPL     ;Nee, dan spreektyd aanhouden.
RLOOPM LDA CDOWN2
    BNE RLOOP7    ;Nee, blijf afvragen.
    LDA TIM2      ;Ja, geef wel of geen roger beep
    CMP STIME     ;Meer dan een minuut ?
    BNE RLOOP8    ;Ga beslist een roger beep seinen
    LDA CDOWN1    ;Voldoende carriertyd voor roger beep?
    CMP #MINUTE-CWMIN
    BPL RLOOPZ    ;Nee, geen roger beep
RLOOP8 LDA #SPTIML ;Zet max. spreektyd op normale mode.
    STA STIME
    LDA #0        ;Reset play counter
    STA TEMP4
    STA TEMP5     ;Reset flutter counter
    LDA TEMP3
    BEQ RLOOP9    ;Geen call-id nodig
RLOOPN LDX #CALL-CALL
    JSR CWRTN
    LDA #MINUT2*CLTIME
    STA CDOWN3
    JMP RLOOP     ;Geen roger
RLOOP9 JSR WAITS
    JSR CHSQ      ;Station alweer in de lucht ?
    BNE RLOOPZ    ;geef maar geen roger meer
    LDX #ROGER-CALL
    JSR RGRTN     ;Ja, geef available status
RLOOPZ JMP RLOOP ;en terug naar RLOOP
    ;
    ;De carrier operated routine
DLOOP JSR TXOFF
    LDA CDOWN3    ;hoe lang is relais up na call ?
    CMP #MINUT2*CLTIME-ELAPM
    BCC DLOOP1    ;Ga closedown doen met call
    LDA #1        ;Nee, closedown zonder call

```

```

        STA TEMP1          ;Flaggetje zetten
        BNE DLOOP2
DLOOP1 LDA #0             ;Closedown met call
        STA TEMP1
DLOOP2 STA TEMP5          ;Reset flutter-counter ook maar.
        INC TEMP4
        LDA TEMP4
        CMP #CWREPT       ;Relais versneld uit ?
        BEQ DLOOP4
        LDA #CDTIME       ;Closedown relais na x seconden
        STA CDOWN1
DLOOP3 LDA CDOWN1
        BEQ DLOOP5        ;Einde carrier operating
        JSR OPKEY         ;Operator einde ?
        BEQ DLOOP4
        JSR CHSQ          ;Draaggolf ontvangen ?
        BEQ DLOOP3        ;Nee
        JSR TXON          ;Ja, herstart relais
        JMP RLOOP         ;Ga opnieuw in relais loop
        ;
        ;Relais uitzetten. Einde repeater.
DLOOP4 JSR OPKEY          ;Heeft de operator soms nog de
        BEQ DLOOP4        ;ingedrukt. Dan wachten.
DLOOP7 JSR TXON          ;Zender aan
        LDA #0
        STA RSTAT1
        JSR AUDOFF
        JSR WAITS
        LDX #CLOSE-CALL
        JSR CWRTN         ;en closedown info
        JSR RSTOFF
        JSR TXOFF
DLOOP5 JMP CLOOP         ;Terug in de beaconloop
        ;
        ;De testloop.
TLOOP  JSR JLOOP         ;Wacht tot er weer geluisterd wordt
        JSR WAITS
        LDA #0            ;Zet counter even op nul.
        STA CDOWN2
        STA RSTAT1       ;Repeater funcyie is nu uit
        JSR RSTOFF
        LDA #TSTTIM      ;Laad toegestane testtyd
        STA CDOWN1
TLOOP1 LDA CDOWN1        ;Blijf toegestane tijd aan staan.
        BEQ TLOOP5
        LDA CDOWN2       ;Moet testmessage geseind ?
        BNE TLOOP2       ;Nee, doorgaan
        LDX #TEST-CALL   ;Ja,
        JSR CWRTN
        LDA #MINUT2*CLTIME
        STA CDOWN3       ;Reset call timer
        JSR TSTON        ;Laat led2 aan staan
        LDA #TSTCLT     ;Laad counter opnieuw
        STA CDOWN2
TLOOP2 JSR OPKEY         ;Wil operator stoppen ?
        BEQ TLOOP5
        JSR CH1750       ;Wil tegenstation stoppen ?
        BNE TLOOP1
        JSR JLOOP
        JSR WAITS        ;Voor evt. Rogerbeep
        JSR AUDON        ;Terug in RLOOP met audio weer aan
        JSR RSTON
        JMP RLOOP9
TLOOP5 JSR WAITS        ;Geef Close cw

```

```

        LDX #CLOSE-CALL
        JSR CWRTN      ;en zender uit..
        JSR TXOFF
        JMP CLOOP      ;Jump naar CloseLOOP
        ;
        ;De wurger...
WLOOP  LDA #ALSIGS    ;Geef de alarmbeeps
        STA TEMP4
WLOOP1 LDA #SECOND    ;in een secondetempo
        STA CDOWN2
WLOOP2 JSR CH1750     ;Wordt er gefloten ?
        BEQ WLOOP8     ;Ja, hertrigger
        JSR CHSQ
        BEQ WLOOP7
WLOOP3 LDA CDOWN2
        BNE WLOOP2     ;Weer een beep ?
        STA RSTAT1     ;Geen verzwakking cw !
        LDX #TIME-CALL
        JSR CWRTN      ;Ja,
        DEC TEMP4
        LDA TEMP4
        BNE WLOOP1     ;alle beeps gehad ?
        JSR AUDOFF     ;Ja, zet zender uit
        JSR TSTON
        JSR TXOFF
WLOOP4 JSR CH1750     ;alsnog 1750 ?
        BEQ WLOOP9     ;Ja, zet relais weer aan
        JSR CHSQ       ;Nee, wacht tot station uit de lucht
        BNE WLOOP4
        JSR RSTOFF
        LDA #0
        STA RSTAT1     ;Clear tevens de relais-flag
        JSR TXON
        JSR WAITS      ;Hij is weg, geef Close CW
        LDX #CLOSE-CALL
        JSR CWRTN      ;zender uit en in CLOOP
        JSR TXOFF
        JMP CLOOP
WLOOP7 JMP RLOOP      ;Ga weer in RLOOP met huidige cndx
WLOOP8 JSR TSTOFF
        JSR AUDON
        JSR TXON
        JMP RLOOP
WLOOP9 JMP ULOOP
        ;
        ;Programma subroutine's
        ;Wachtlus tot station weer luisterd.
JLOOP  LDA #JTIME
        STA CDOWN2
JLOOP1 LDA CDOWN2
        BEQ JLOOP2     ;Jammer het duurt te lang
        JSR CHSQ
        BNE JLOOP1
JLOOP2 RTS
        ;
        ;Reset clockcounter for correct counting
TRIGCK PHA
        LDA PA
        ORA #%00000010
        STA PA
        AND #%11111101
        STA PA
        PLA
        RTS

```

```

;Wachtlus (halt programma)
WAITS LDA #SECOND
WAIT JSR TRIGCK
STA CDOWN1 ;Door de IRQ routine wordt deze
WAIT1 LDA CDOWN1 ;variabele afgeteld.
BNE WAIT1
RTS
;
;Besturings subroutine's
;Check 1750 Hz detektor
CH1750 LDA PA
EOR #$FF ;Inverteren
AND #%00001100 ;Maskeren juiste bit (Squelch ook open !)
CMP #%00001100
RTS ;De caller neemt een beslissing.
;Check squelch
CHSQ LDA PA
EOR #$FF
AND #%00001000 ;Maskeer juiste bit
RTS
;Operator switch
OPKEY LDA PA
AND #%00010000 ;Maskeer juiste bit
RTS
;Status led 1 aan
RSTON LDA PB
ORA #%00100000 ;Led 1 aan
STA PB
RTS
;Status led 1 uit
RSTOFF LDA PB
AND #%11011111 ;Led 1 uit
STA PB
RTS
RTS
;Status led 2 aan
TSTON LDA PB
ORA #%00001000 ;Zet operator led2 aan
STA PB
RTS
;Status led 2 uit
TSTOFF LDA PB
AND #%11110111 ;Zet led2 uit
STA PB
RTS
;TX-on
TXON LDA PB
AND #%10111111 ;Zenderstuurbit laag
STA PB
RTS
;TX-off
TXOFF LDA PB
ORA #%01000000 ;Zenderstuurbit hoog
STA PB
RTS
;Audio-on
AUDON LDA PA
ORA #%00100000
STA PA
RTS
;Audio-off
AUDOFF LDA PA
AND #%11011111
STA PA

```



```

RTS
;Tone low
TLOW LDA PA
AND #%10111111
STA PA
RTS
;Tone high
THIGH LDA PA
ORA #%01000000
STA PA
RTS
;Squelch on
SQON LDA PB
AND #%01111111
STA PB
LDY #80
JMP DELAY
;Squelch off
SQOFF LDA PB
ORA #%10000000
STA PB
LDY #10
DELAY LDX #$FF
DELAY1 DEX
BNE DELAY1
DEY
BNE DELAY
RTS
;
;IRQ routine 1. Clear interrupt.
;                2. Toggle local led.
;                3. Decrement counter.
IRQRTN PHA
BIT IRQADR ;Clear interrupt
LDA PB
EOR #%00010000 ;Toggle local led only
STA PB
LDA CDOWN1
BEQ IRQRT1 ;Alleen aftellen bij <> 0.
DEC CDOWN1
IRQRT1 LDA CDOWN2
BEQ IRQRT2
DEC CDOWN2
IRQRT2 LDA SECF ;Ja, ga CDOWN3 verlagen
EOR #1
STA SECF ;in een 2 seconden tempo
LDA CDOWN3
BEQ IRQRT3 ;Lager dan 0 kan niet
SEC
SBC SECF
STA CDOWN3
IRQRT3 LDA CDOWN4
BEQ IRQRT4
DEC CDOWN4
IRQRT4 PLA
CLI
RTI
;
;NMI routine. Reset watchdog !!
NMIRTN JSR TRIGCK
RTI
;CW generatie programma
;
;De tekst staat direkt gecodeerd in het programma.

```

```

;
;Code's 0-FC ?-32 J-78 T-C0
;        1-7C A-60 K-B0 U-30
;        2-3C B-88 L-48 V-18
;        3-1C C-A8 M-E0 W-70
;        4-0C D-90 N-A0 X-98
;        5-04 E-40 O-F0 Y-B8
;        6-84 F-28 P-68 Z-C8
;        7-C4 G-D0 Q-D8 /-94
;        8-E4 H-08 R-50 -FF
;        9-F4 I-20 S-10 =-8C
;        beginteken-AC
;        sluitteken-54
;
LDA #SPEEDR
STA SPEED
JSR CWRTN
LDA #SPEEDC
STA SPEED
RTS
CWRTN STX TEMP2      ;In X-reg staat message offset
JSR TSTOFF
NXT   LDX #2         ;Karakter space-1
JSR TRIGCK          ;Voorkom een NMI gedurende CWRTN
LDA RSTAT1
BEQ SPC
JSR CHSQ
BEQ SPC
STA TEMP6
JSR TLOW
SPC   CLC
JSR ELEM            ;Ga space uitvoeren.
LDX TEMP2
INC TEMP2           ;Verhoog pointer
LDA CALL,X         ;Lees karakter
BEQ CWRDY          ;0 dan stoppen
STA TEMP1          ;Ga verwerken
CMP #$FF           ;Spatie ?
BNE SHFT
LDX #7              ;Ja, laad wachttyd
BNE SPC
SHFT  ASL TEMP1     ;Ga de info verschuiven
BEQ NXT
LDX #1              ;Punttyd
BCC DOT
LDX #3              ;Streeptyd
DOT   SEC
JSR ELEM            ;Ga toon van die lengte genereren
CLC
INX
JSR ELEM            ;Punttyd=tekenspatie wachten
BEQ SHFT           ;en terug in de karakterloop.
;
;Generen toon met lengte in X-reg
ELEM  SEI
BCC ELEMA
JSR TSTON
BCS ELEMB
ELEMA JSR TSTOFF
ELEMB PHP
LDA RSTAT1
BEQ ELEMD
JSR CH1750
BNE ELEMC

```

```

        JSR AUDOFF
        JMP ELEMD
ELEM3 JSR AUDON
        JMP ELEMD      ;Zo houden we de tyd gelijk
ELEM4 PLP
ELEM5 LDY SPEED      ;Snelheid
ELEM6 LDA #TONE      ;Toon
        STA TEMP3
ELEM7 DEC TEMP3
        BNE ELEM2
        BCC QUIET
        LDA PA        ;Wegzetten info in RIOT
        EOR #1
        STA PA
QUIET  DEY            ;Aftellen Y-reg voor lengte/snelheid
        BNE ELEM1
        DEX
        BNE ELEME
        CLI
        RTS
CWRDY  JSR THIGH
        JMP TSTOFF    ;en terug caller
        ;
        ;Resident CW messages
        ;PI3ZOD (PA3ANG)
CALL   .BYTE $68
        .BYTE $60
        .BYTE $1C
        .BYTE $60
        .BYTE $A0
        .BYTE $D0
        .BYTE $0
        ;E (= roger beep)
ROGER  .BYTE $40
        .BYTE $0
        ;T (= Timeout)
TIME   .BYTE $C0
        .BYTE $0
        ;SK (= closedown)
CLOSE  .BYTE $10
        .BYTE $B0
        .BYTE $0
        ;Beacon message
        ;PI3ZOD Emmen JO32KS 75m ASL pwr 1W
BEACON .BYTE $FF
        .BYTE $68
        .BYTE $60
        .BYTE $1C
        .BYTE $60
        .BYTE $A0
        .BYTE $D0
        .BYTE $FF
        .BYTE $FF
        .BYTE $40
        .BYTE $E0
        .BYTE $E0
        .BYTE $40
        .BYTE $A0
        .BYTE $FF
        .BYTE $78
        .BYTE $F0
        .BYTE $1C
        .BYTE $3C
        .BYTE $B0

```

```
.BYTE $10
.BYTE $FF
.BYTE $C4
.BYTE $04
.BYTE $E0
.BYTE $FF
.BYTE $60
.BYTE $10
.BYTE $48
.BYTE $FF
.BYTE $68
.BYTE $70
.BYTE $50
.BYTE $FF
.BYTE $7C
.BYTE $70
.BYTE $FF
.BYTE $0
;Test message
;VVV de PI3ZOD JO32KS
TEST .BYTE $FF
      .BYTE $18
      .BYTE $18
      .BYTE $18
      .BYTE $FF
      .BYTE $90
      .BYTE $40
      .BYTE $68
      .BYTE $60
      .BYTE $1C
      .BYTE $60
      .BYTE $A0
      .BYTE $D0
      .BYTE $FF
      .BYTE $78
      .BYTE $F0
      .BYTE $1C
      .BYTE $3C
      .BYTE $B0
      .BYTE $10
      .BYTE $FF
      .BYTE $0
.END
```