



# Frequentie-meter

M. Dohmen

R. Koekoek

Het hier beschreven programma kan een frequentie meten tot, let wel, 100 kHz met een afwijking van niet meer dan één trilling op de 65536. De frequentie wordt in vijf digits uitgelezen. Het programma is geschreven voor de KIM, maar is ook goed bruikbaar voor andere 6502-microcomputers, met name de SYM en de Junior. Eenmaal gestart meet het programma een frequentie continu, wanneer toets „6” wordt ingedrukt en eenmalig, wanneer toets „0” wordt ingedrukt.

## Overflow

In dit programma is gebruik gemaakt van een eigenschap van de 6502, die tot nog toe niet veel werd toegepast. Deze microprocessor bezit namelijk een SO (Set Overflow) pen 38, waarmee van buiten af de overflow-flag kan worden gezet. Deze is op de meeste computers direct naar buiten uitgevoerd via een connector (RO, pen E-5 bij de KIM; SO, pen 14-c bij de Junior). De overflow-flag wordt gezet, wanneer het signaal op deze ingang van hoog naar laag gaat.

Dit vormt een snelle en directe toegang tot de microprocessor, waarbij de rest van de computer niet wordt betrokken. Als de overflow-flag is gezet, kan deze alleen nog via de software worden beïnvloed.

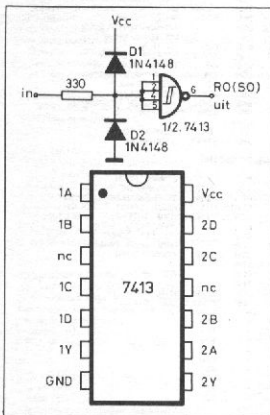
## Buffering

Omdat rechtstreeks in schakelingen moet worden gemeten, is buffering en beveiliging van de RO-ingang wel noodzakelijk. Het gemakkelijkst kan dit worden gerealiseerd met een Schmitt-trigger poort, de 7413. Deze wordt dan nog voorzien van een spannings- en stroombeveiliging volgens de afbeelding. Op deze manier kunnen signalen van elke golfvorm worden gemeten, mits de amplitude ervan voldoende groot is en het signaal rond de 2,5 V ligt.

## Werking

Het programma start op locatie 0200 hex. Eerst wordt de interruptvector gezet op het beginadres van de interruptroutine (0298 hex). Vervolgens wordt een teller geïnitieerd voor het aantal interrupts, waarna de uitlezing wordt gewist. Dan wordt de timer gezet op ongeveer 1/4 seconde, waarbij interrupts worden toegestaan. Voor een volledige cyclus worden vier interrupts verwerkt, wat resulteert in een totale tijd van ongeveer 1 seconde. Om deze tijd exact te maken en om de meter te kunnen

ijken volgt een extra vertragingroutine van ongeveer 3 ms. De accumulator, de carry- en overflowflag worden nul gemaakt en vervolgens wordt het aantal overflows geteld, dat plaats vindt tot aan het eerst volgende interrupt. Er wordt binair geteld met de minst significante bits in het X-register, de volgende in het Y-register en de meest significante bits in de accumulator. Doordat deze routine zo kort mogelijk is gehouden, ligt de maximaal te meten frequentie op ongeveer 100 kHz. De fout van één op de 65536 ontstaat doordat eenmaal op de FFFF hex maal de accumulator moet worden opgehoogd. Dit maakt de telroutine iets langer. Wanneer een interrupt komt, wordt naar de interruptroutine gesprongen. Hierin wordt de timer opnieuw gestart voor een 1/4 seconde, mits dit niet het vierde interrupt was. Was dit wel het geval dan wordt de binaire waarde in de teller omgezet naar decimaal. Dit gebeurt nu pas, omdat bij direct decimaal tellen alles moet gebeuren in de accumulator en dat kost meer tijd. De conversie van binair naar decimaal gaat als volgt: in een tabel staat de binaire voorstelling van respectievelijk de decimale waarden 1, 10, 100, 1000 en 10000. Eerst wordt 10000 van de tellerstand, die tijdelijk in een aantal geheugenlocaties is opgeslagen, afgetrokken. Zolang de uitkomst nog niet negatief is, wordt een „10000-teller” opgehoogd. De uitkomst wordt telkens weer in de oorspronkelijke teller terug gezet. Als de uitkomst negatief is, wordt de waarde van de teller, die voor de aftrekking aanwezig was, genomen en wordt daarvan de volgende waarde uit de tabel afgetrokken. Dit gaat zo door totdat de teller nul is. De decimale waarde wordt rechtstreeks bijgehouden in de registers voor de uitlezing. Na de conver-





sie licht de uitlezing op, waarbij de waarde zichtbaar wordt. Hierna wordt gekeken of een toets was ingedrukt. Zo neen, dan wordt de vorige toets opgehaald. Zo ja, dan stopt bij een „0” het programma met meten en blijft bij een „6” de computer continu de frequentie aangeven.

## Afregeling

De afregeling van de frequentiemeter gaat als volgt. Een delerschakeling, bestaande uit twee 10-delers, wordt verbonden met de 1 MHz klok van de computer. De frequentie aan de uitgang hiervan zal dan in de buurt van de 10 kHz liggen. Dit signaal wordt toegevoerd aan de meet-

ingang en het programma wordt gestart. Door de waarden in de geheugenplaatsen 021C en 021E van de extra verpagingsroutine te veranderen, wordt de uitlezing afgesteld op 10000. Gekozen is voor een 100-deler, omdat dan de accumulator nog niet in het spel is bij het tellen. De meter is zo op zijn nauwkeurigst. Door de 100-deler te schakelen tussen de buffer (7413) en de RO (SO) ingang wordt het meetgebied van de frequentieteller uitgebreid tot 10 MHz. Belangrijk is ook, dat om de timer te laten werken, bij de KIM PB7 (A-15) met NMI (E-6) moet worden verbonden en bij de Junior NMI (12-c) met IRQ (12-a). De timer genereert namelijk een interrupt, dat moet kunnen worden gehonoreerd.

## Wijzigingen voor de Junior

Het programma, zoals dat staat in de lijst, werkt op de KIM. Om het voor de Junior geschikt te maken moeten de volgende veranderingen worden aangebracht:

Adres	KIM	Junior
0203	FB	7B
0204	17	1A
0208	FA	7A
0209	17	1A
0219	0F	FF
021A	17	1A
0283	1F	8E
0284	1F	1D
02A0	0F	FF
02A1	17	1A

MICRO-WARE ASSEMBLER 6502-1.0 PAGE 01

```

0010:
0020:
0030:
0040:
0050:
0060:
0070:
0080:
0090:
0100:
0110:
0120: 0200      ORG      $0200
0130:
0140:
0150:
0160: 0200      INTCHT * $0000
0170: 0200      ZEROP * $0001
0180: 0200      TEMPC * $00E0
0190: 0200      TEMPB * $00E1
0200: 0200      TEMPA * $00E2
0210: 0200      INL * $00F8
0220: 0200      INH * $00F9
0230: 0200      POINTL * $00FA
0240: 0200      POINTH * $00FB
0250: 0200      CHAR * $00FE
0260:
0270:
0280:
0290: 0200      TABLO * $02AF
0300: 0200      TABHI * $02B0
0310:
0320:
0330:
0340: 0200      TIMER * $170F
0350:
0360:
0370:
0380: 0200      INTLO * $17FA
0390: 0200      INTHI * $17FB
0400:
0410:
0420:
0430: 0200      SCANDS * $1F1F
0440:
0450:
0460:
0470: 0200 A9 02      INIT      LDRAIN #02
0480: 0202 8D FB 17      STA      INTHI
0490: 0205 A9 98      LDRAIN #98      ZET INTERRUPTVECTOR
0500: 0207 8D FA 17      STA      INTLO
0510: 020A A9 FC      LDRAIN #FC      ZET INTERRUPTTELLER
0520: 020C 85 D0      STA      INTONT      OP -4
0530: 020E A9 00      LDRAIN #00
0540: 0210 85 F9      STA      INH      MARK DISPLAY EN DUS OOK
0550: 0212 85 FA      STA      POINTL    DE TELLER SCHOON
0560: 0214 85 FB      STA      POINTH
0570: 0216 A9 F5      LDRAIN #F5      ZET TIMER EN MARK
0580: 0218 8D 0F 17      STA      TIMER    INTERRUPTS MOGELIJK
0590:
0600:
0610:
0620: 0218 A2 03      LDIXIN #03      PAS DEZE WAARDEN AAN
0630: 021D A0 00      BULOOP LDVIM #00      VOOR DE VEREISTE
0640: 021F 88      BULOOP DEV      NAUWKEURIGHEID
0650: 0220 D0 FD      BNE      BULOOP
0660: 0222 CA      DEX
0670: 0223 D0 F8      BNE      BULOOP
0680:
0690:
0700:
0710: 0225 B8      CLU
0720: 0226 18      CLC
0730: 0227 98      TYA      MARK ACCU SCHOON
0740: 0228 50 FE      OVERFL BUC      OVERFL OVERFLOW GESET?
0750: 022A B8      CLU
0760: 022B E8      INX      HOOG X OP NET 1
0770: 022C D0 FA      BNE      OVERFL    IS X = FF?
0780: 022E C8      INY      HOOG Y OP NET 1
0790: 022F D0 F7      BNE      OVERFL    IS Y = FF?
0800: 0231 69 01      ADCIM #01      HOOG ACCU OP NET 1
0810: 0233 D0 F3      BNE      OVERFL    IS ACCU = FF?

```

```

0830:
0840:
0850: 0235 85 E2
0860: 0237 84 E1
0870: 0239 86 E0
0880: 023B A2 08
0890: 023D A5 E2
0900: 023F C3 00
0910: 0241 D0 0C
0920: 0243 A5 E1
0930: 0245 D0 B0 02
0940: 0248 D0 05
0950: 024A A5 E0
0960: 024C D0 AF 02
0970: 024F 90 2D
0980: 0251 38
0990: 0252 A5 E0
1000: 0254 FD AF 02
1010: 0257 85 E0
1020: 0259 A5 E1
1030: 025B FD B0 02
1040: 025E 85 E1
1050: 0260 A5 E2
1060: 0262 E3 00
1070: 0264 85 E2
1080: 0266 8A
1090: 0267 48
1100: 0268 4A
1110: 0269 4A
1120: 026A 4A
1130:
1140:
1150:
1160: 026B 80 04
1170: 026D F6 F9
1180: 026F D0 09
1190: 0271 85 F9
1200: 0273 69 0F
1210: 0275 95 F9

```

BINAIR NAAR DECIMAAL CONVERSIE

```

BINDEC STA TEMPA BEWAAR ACCU
STV TEMPB BEWAAR Y
STX TEMPC BEWAAR X
LDXIM #08 ZET POINTER
POS LDA TEMPA
CMPIH #00 IS DE TELLER
BNE SUBT NOG STEEDS
LDA TEMPB GROTER DAN
CMFAX TABHI DE HUIDIGE
BNE SUBT WAARDE
LDA TEMPC IN DE
CMFAX TABLO TABEL
SUBT BCC NEXT NEEN, HAAR VOLGENDE TABELWAARDE
SEC
LDA TEMPC DE HUIDIGE
SBCAX TABLO WAARDE UIT
STA TEMPC
LDA TEMPB DE TABEL
SBCAX TABHI DE
STA TEMPB STAND IN
LDA TEMPA DE TELLER
SBCIM #00
STA TEMPA EN ZET TELLERSTAND TERUG
TYA
PHA
LSRA
LSRA
TAX BEWAAR DE TABELPOINTER
IN X OP DE STACK
BEPAL IN WELK DIGIT DEZE
TELWAARDE MOET KOMEN
DOOR POINTER IN X

```

TELLERSTAND DECIMAAL IN DISPLAY

```

BCS TIENT TIENTIAL?
INCZX INH NEEN, EEN ERBIJ
BNE NOG EN TERUG
TIENT LDZX INH HAAR VORIGE STAND
ADCIM #0F EN TEL TIENTIAL OP
STAX INH DISPLAY
NOG PLA HAAR TABELPOINTER VAN STACK
TAX EN PLAATS DIE IN X
BPL POS ALLES KLAAR?
DEX TABELPOINTER NAAR
DEX VOLGENDE WAARDE
BPL POS EN TERUG
JSR SCANDS DISPLAY TELLER
BNE KEY TOETS INGEDRUKT?
LDA INL NEEN, HAAR VORIGE
BNE DI WAS DIT NUL?
DEC ZEROP NEEN, DELAY
BNE DISPL JA, BLIJF DISPLAYEN
JMP START OPNIEUW
LSRA
STA INL BEWAAR TOETS
JMP START OVERNIEUW

```

INTERRUPTROUTINE

```

PHA BEWAAR ACCU
INC VERLAAG DE TELLER
BEQ UIER ZIJN ER VIER GEMEEST?
LDRAIN #F5 NEEN, LAADT TIMER EN
STA TIMER MARK INTERRUPT MOGELIJK
PLA HAAR ACCU TERUG
RTI EN KEER TERUG
PLA HAAR ACCU TERUG
STA CHAR BEWAAR DATA
PLA
PLA ZET STACK GOED
PLA
LDA CHAR HAAR DATA OP
JMP BINDEC CONVERTEER
= $01
= $08
= $09
= $0B
= $0D
= $0E
= $0F
= $10
= $11
= $12
= $13
= $14
= $15
= $16
= $17
= $18
= $19
= $1A
= $1B
= $1C
= $1D
= $1E
= $1F
= $20
= $21
= $22
= $23
= $24
= $25
= $26
= $27
= $28
= $29
= $2A
= $2B
= $2C
= $2D
= $2E
= $2F
= $30
= $31
= $32
= $33
= $34
= $35
= $36
= $37
= $38
= $39
= $3A
= $3B
= $3C
= $3D
= $3E
= $3F
= $40
= $41
= $42
= $43
= $44
= $45
= $46
= $47
= $48
= $49
= $4A
= $4B
= $4C
= $4D
= $4E
= $4F
= $50
= $51
= $52
= $53
= $54
= $55
= $56
= $57
= $58
= $59
= $5A
= $5B
= $5C
= $5D
= $5E
= $5F
= $60
= $61
= $62
= $63
= $64
= $65
= $66
= $67
= $68
= $69
= $6A
= $6B
= $6C
= $6D
= $6E
= $6F
= $70
= $71
= $72
= $73
= $74
= $75
= $76
= $77
= $78
= $79
= $7A
= $7B
= $7C
= $7D
= $7E
= $7F
= $80
= $81
= $82
= $83
= $84
= $85
= $86
= $87
= $88
= $89
= $8A
= $8B
= $8C
= $8D
= $8E
= $8F
= $90
= $91
= $92
= $93
= $94
= $95
= $96
= $97
= $98
= $99
= $9A
= $9B
= $9C
= $9D
= $9E
= $9F
= $A0
= $A1
= $A2
= $A3
= $A4
= $A5
= $A6
= $A7
= $A8
= $A9
= $AA
= $AB
= $AC
= $AD
= $AE
= $AF
= $B0
= $B1
= $B2
= $B3
= $B4
= $B5
= $B6
= $B7
= $B8
= $B9
= $BA
= $BB
= $BC
= $BD
= $BE
= $BF
= $C0
= $C1
= $C2
= $C3
= $C4
= $C5
= $C6
= $C7
= $C8
= $C9
= $CA
= $CB
= $CC
= $CD
= $CE
= $CF
= $D0
= $D1
= $D2
= $D3
= $D4
= $D5
= $D6
= $D7
= $D8
= $D9
= $DA
= $DB
= $DC
= $DD
= $DE
= $DF
= $E0
= $E1
= $E2
= $E3
= $E4
= $E5
= $E6
= $E7
= $E8
= $E9
= $EA
= $EB
= $EC
= $ED
= $EE
= $EF
= $F0
= $F1
= $F2
= $F3
= $F4
= $F5
= $F6
= $F7
= $F8
= $F9
= $FA
= $FB
= $FC
= $FD
= $FE
= $FF

```