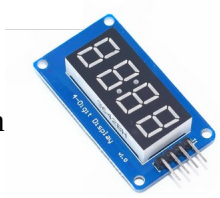
# Pi Pico klokradio

## door Roland Leurs

Tja … daar sta je dan. Heb je op de ROX zo'n leuk printje gekregen dat men Pi Pico noemt, maar wat moet je er mee? Wat kan je er mee? Je kon er een aantal extra onderdelen bij bestellen en dan heb je genoeg om op een knopje te drukken en dan gaat het lampje aan. Maar daar maak je weinig indruk mee op verjaardagen en andere familiefeestjes waar je de enige bent die nog iets met microcontrollers doet. Daarom gaan we hier aan de slag met een uitdagender project: een klokradio met de Pi Pico, geschreven in (hoe hoort het ook anders) BBC Basic.

Ik schrijf het natuurlijk een beetje cynisch met dat knopje en het lampje. Uiteindelijk is dat wel de basis waarmee elke processor of controller kan communiceren met de buitenwereld. Of je nu inderdaad een lampje wil laten branden op basis van een of ander invoersignaal, of een motor aansturen. Voor de controller maakt dat niets uit. Er komt input en er volgt output. Een knopje en een ledje zijn wat dat betreft laagdrempelig en eenvoudig aan te sluiten. En wat ook belangrijk is: ze zijn betaalbaar en je kunt niet veel stuk maken.

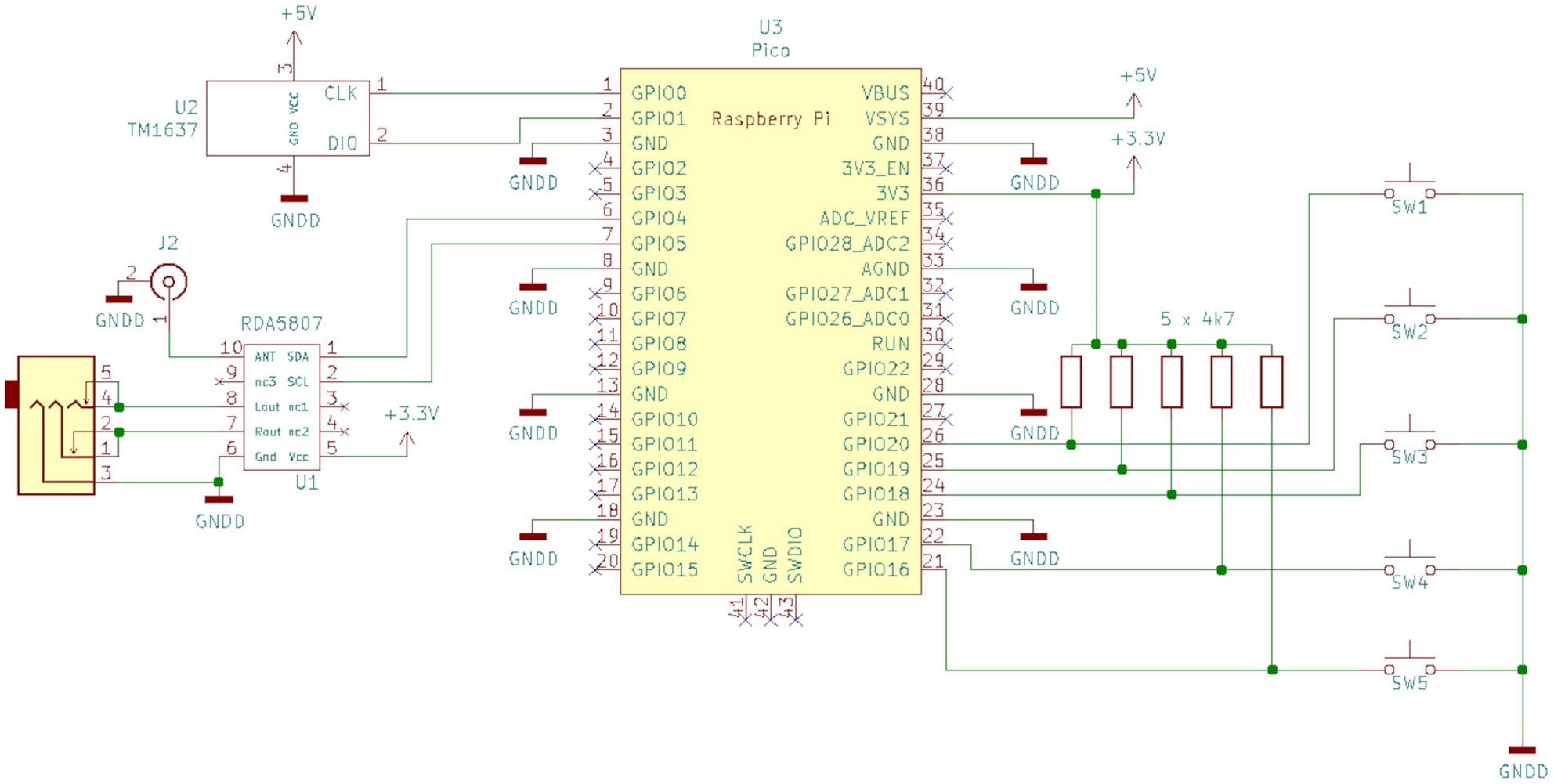


Ons klokradio project maakt gebruikt van vier 7-segmentdisplays inclusief een dubbele punt. Dat zijn gelijk al 30 ledjes. Deze displays zitten echter in een module "verpakt" die maar vier aansluitingen heeft: 5V voeding, 0V, datalijn en kloklijn. Doordat er maar vier draadjes nodig zijn, is het heel makkelijk om deze aan te sluiten op de Pico.


Het radiogedeelte is opgebouwd uit een RDA5807M. Dit is een klein printje van circa 1 $cm^2$ met daarop een complete FM radio ontvanger. Ook hier zijn maar vier aansluitingen voor nodig. Eigenlijk dezelfde als het display. Maar dan toch anders. Deze module werkt namelijk met het $I^2C$ protocol. Dit is een serieel protocol dat door Philips is bedacht om diverse IC's met elkaar te laten communiceren zonder dat daar al te veel printsporen voor nodig zijn. Het protocol van het display lijkt wel iets op $I^2C$ maar is het niet. Ik ga u niet vermoeien met de details van deze protocollen want u krijgt de benodigde bibliotheken van mij aangeleverd. Deze radio-ontvanger kan ook RDS data ontvangen, decoderen en doorgeven aan de processor of controller. Deze functionaliteit laten we voorlopig nog even links liggen.

Beide onderdelen zijn letterlijk voor een paar euro bij Ali-express te koop. Strikt genomen heb je aan een bread-board al voldoende om de hele schakeling op te bouwen. Voor het gemak heb ik er echter een printje voor gemaakt. Dit printje biedt ook nog plaats aan vijf druktoetsen die we later voor de bediening kunnen gebruiken. Het schema van deze schakeling staat op de volgende pagina.

Het proto-type printje is ook al ontworpen en gemaakt. Uiteraard zijn er tijdens het testen nog enkele verbeterpunten naar boven gekomen. Zo zet het display 4,5V op de datalijn. Dat is eigenlijk net iets te hoog voor de Pico, hoewel de mijne daar verder niet over klaagt. Ook zou een aantal ledjes niet misstaan om bijvoorbeeld een alarmfunctie, tuning en stereo-indicatie weer te geven. De komende pagina's geven in ieder geval al een mooie aanzet om te starten met je cadeautje en als je de schakeling op een bread board zet dan maak je ook al een goede indruk op de eerder genoemde feesten en partijen.


+5V
U2
TM1637
CLK
DIO
GND VCC
GNDD
U3
Pico
Raspberry Pi
GPIO0
GPIO1
GPIO2
GPIO3
GPIO4
GPIO5
GPIO6
GPIO7
GPIO8
GPIO9
GPIO10
GPIO11
GPIO12
GPIO13
GPIO14
GPIO15
VBUS
VSYS
3V3_EN
3V3
ADC_VREF
GPIO28_ADC2
AGND
GPIO27_ADC1
GPIO26_ADC0
RUN
GPIO22
GPIO21
GPIO20
GPIO19
GPIO18
GPIO17
GPIO16
SWCLK
SWDIO
+3.3V
5 x 4k7
SW1
SW2
SW3
SW4
SW5
J2
RDA5807
ANT SDA
nc3 SCL
Lout nc1
Rout nc2
Gnd Vcc
U1

# BBC Basic installeren op de Pico

De eerste stap is om BBC Basic te installeren op de Pico, voor zover dat nog niet gebeurd is. De ontwerpers van de Pico hebben er goed over nagedacht hoe je de Pico zo eenvoudig mogelijk voorziet van het programma (firmware) dat je wil gebruiken. Dit kan een programma zijn met een specifieke taak (zoals mijn Electron Video kaart), maar ook een omgeving met een programmeertaal zoals Micro Python of uiteraard BBC Basic.

Op de Pico zit een klein wit drukknopje om de boot-mode te selecteren. Als je de Pico aan zet zonder op dit knopje te drukken dan zal deze normaal opstarten. Druk je echter dit knopje in en je sluit dan de Pico aan op je computer, dan zal deze in de Finder (Apple MacOS) of Verkenner (Windows) als USB drive getoond worden. Het enige dat je dan nog hoeft te doen is het bestand met jouw programma of firmware te slepen naar het drive-ikoontje van de Pi Pico. Het bestand zal dan naar het flashgeheugen van de Pico gekopieerd worden en als dit klaar is herstart de Pico automatisch en wordt jouw programma of firmware gestart.


Laten we dit stap voor stap even uitvoeren:

- Download het bestand vanaf https://bbcmicro.nl/pico-klok-radio/bbcbasic_console_rpico.uf2
- Indien de Pico aangesloten is op je computer, koppel deze even los.
- Blader in de verkenner naar je Downloadmap waar het zojuist gedownloade bestand staat.
- Druk het witte knopje in en sluit met je andere hand de USB kabel weer aan op je computer.
- Er zal nu in de Finder of Verkenner een nieuw “drive” ikoontje verschijnen. Deze heet RPI-RP2.
- Sleur-en-pleur het uf2 bestand naar dit ikoontje.
- Even wachten totdat het bestand gekopieerd is en de Pico zich automatisch herstart.

Bij deze herstart kan jouw computer een melding geven dat je het medium eerst moet ontkoppelen (uitwerpen) alvorens het te verwijderen. Negeer deze melding maar, het bestand is goed gekopieerd en het kan geen kwaad dat het medium “koud” verwijderd wordt.

BBC Basic is nu geïnstalleerd en klaar voor gebruik.

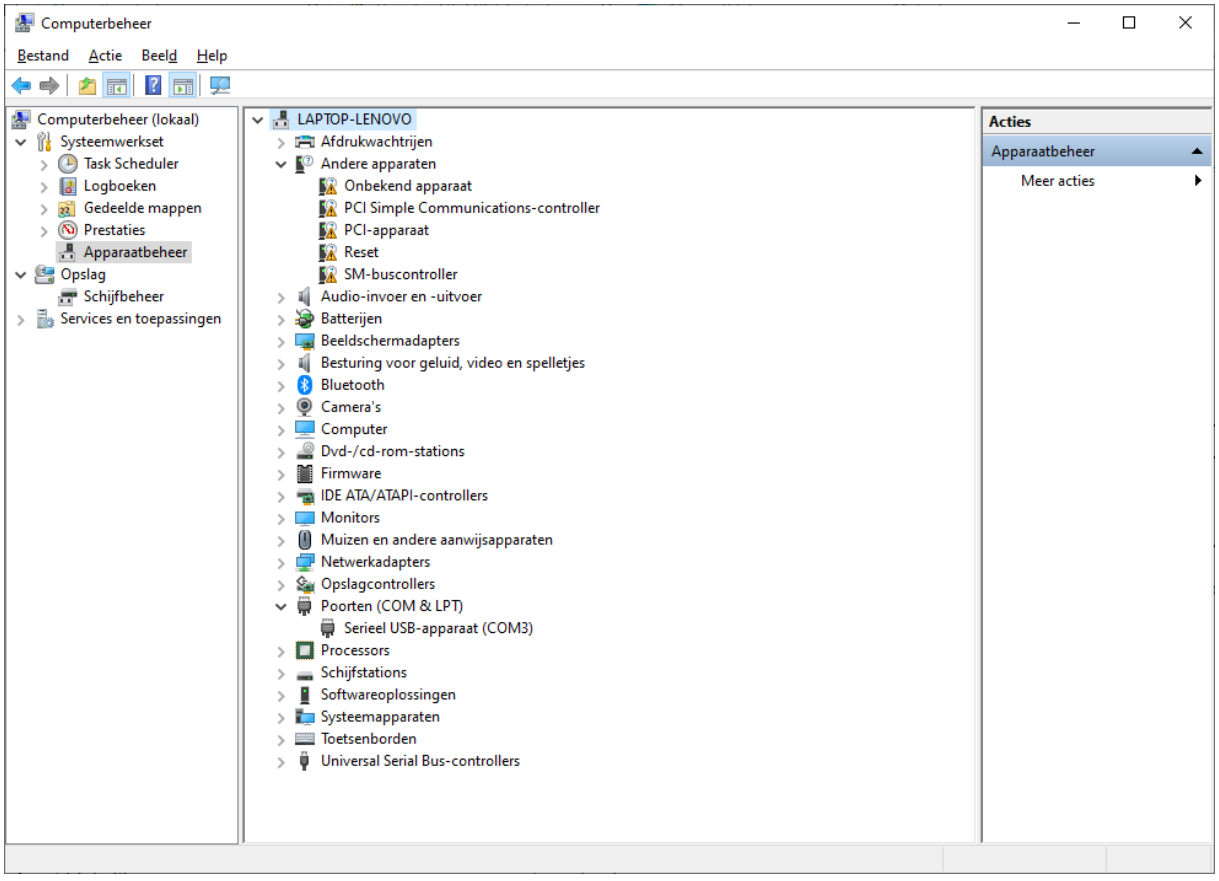
# BBC Basic gebruiken

Nu we BBC Basic op onze Pico hebben draaien willen we natuurlijk wel iets kunnen zien. Aangezien we geen toetsenbord en beeldscherm op onze Pico aan kunnen sluiten (althans niet in dit project) maken we gebruik van een terminal programma. Via de USB kabel kunnen we de Pico gebruiken alsof dit een apparaat is dat via een seriële poort is aangesloten. Als terminalprogramma kunnen we bijvoorbeeld *putty* onder Windows gebruiken en onder MacOS en Linux is *minicom* mijn favoriete programma hiervoor. Met gepaste schaamte kan ik u melden dat ik echt niet zou weten hoe dit op uw favoriete Acorn computer (of Raspberry Pi onder RiscOS) gedaan kan worden.

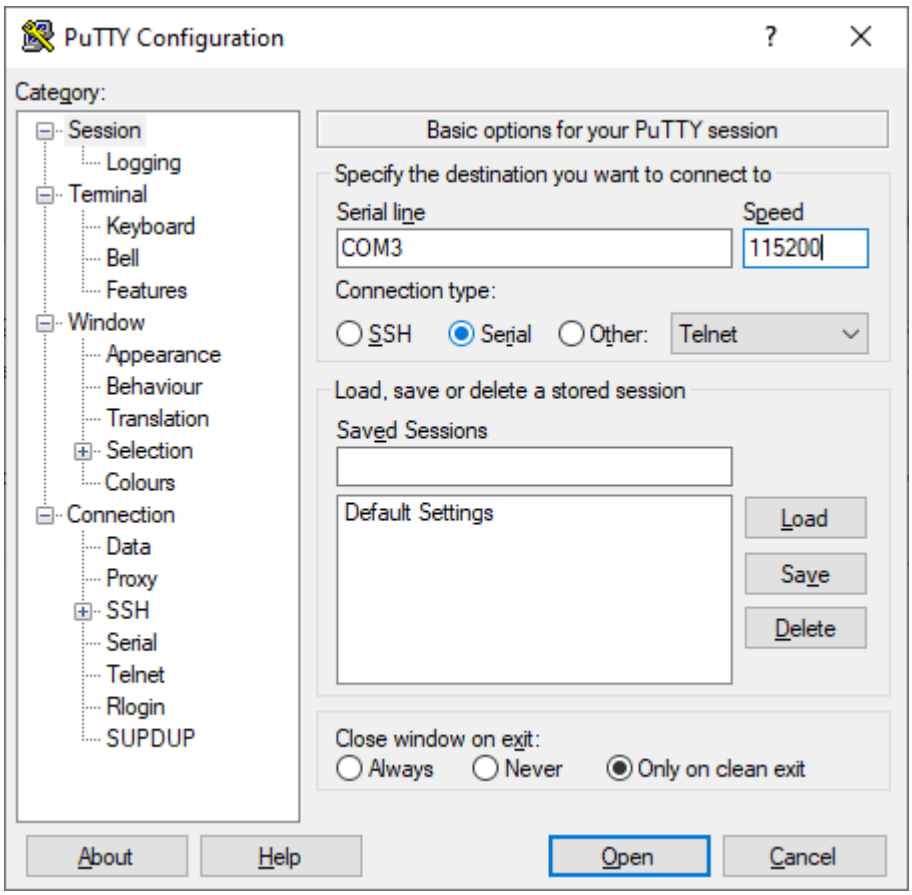
*Putty*

Putty is een eenvoudig programma dat normaliter gebruikt wordt voor het opzetten van beveiligde verbindingen (secure shell). Maar het is ook prima in staat om als seriële terminal te functioneren. Je kunt putty.exe downloaden van https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html. Kies voor het gemak voor de binary file putty.exe voor 64 bit x86 systemen. Dan hoef je verder niets te installeren. Uiteraard kies je voor de 32 bits versie als je een 32 bits Windows versie gebruikt.

Om de COM poort te bepalen open je Apparaatbeheer in Windows en blader je naar Poorten (COM & LPT):




In dit geval is de poort dus COM3. Als je putty start kies je voor een seriële verbinding. Je geeft dan de poort op die je in apparaatbeheer hebt gevonden. De baudrate is 115200.




En je klikt op Open om de communicatie te starten.

*Minicom*

Onder Linux zal minicom waarschijnlijk al geïnstalleerd zijn of anders is het beschikbaar via de package manager van jouw distributie. Onder MacOS is het het makkelijkst om minicom via ‘homebrew’ te installeren.

Installeer eerst homebrew. Open hiervoor een terminalvenster en geef het commando:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Installeer vervolgens minicom met de opdracht: `brew install minicom`

Als je dit gedaan hebt moet je nog even uitzoeken op welke poort jouw Pico aangesloten is. Voer hiervoor in het terminalprogramma de volgende opdracht uit:

```
ls dev/tty*
```

Je krijgt dan een lijst met de beschikbare tty (seriële) poorten. Dit kunnen fysieke poorten zijn, maar ook logische. De Pico heeft de naam tty.usbmodemXXXXX met op de plaats van de X een getal dat afhankelijk is van de poort waarop de Pico aangesloten is. Bij mij is dat 14101.

Je kunt minicom dan starten met: `minicom --device /dev/tty.usbmodem14101`

Na het starten van het terminal programma zal de Pico na een reset zich melden met:

```
BBC BASIC for Pico Console v0.41, Build Mar 18 2022, USB
Console, UART Console, Flash Filesystem, Serial devices,
SDL Sound, Stack Check 4
(C) Copyright R. T. Russell, 2022
>
```

Je kunt nu van start gaan met het programmeren in BBC Basic. Vrijwel alle bekende statements en functies zijn beschikbaar. Ook is een aantal van de bekende *-commando's ingebouwd. Zo kan je met *CAT een lijst opvragen met alle bestanden op het opslagmedium. Met LOAD en SAVE kan je programma's inlezen resp. opslaan.

Vreemd genoeg is het statement OLD niet beschikbaar. Let dus op dat je je programma dus kwijt bent als je NEW intypt of de Pico herstart voordat je je programma met SAVE opgeslagen hebt. Ik heb trouwens wel al een paar keer meegemaakt dat een BASIC programma de Pico enorm kan laten vastlopen zodat alleen een reset of power cycle de boel weer los krijgt. Ook dan gaan niet opgeslagen gegevens verloren. Doe dus ook even een SAVE voordat je een RUN doet als je hard gewerkt hebt aan je programma!

# Het 7-segment display aansturen

Om het display aan te sturen heb ik een aantal PROCedures geschreven die het leven makkelijk maken. Je hoeft dus niet zelf het seriële protocol uit te vlooien om het display te gebruiken. Voel je vrij om de routines uit mijn bibliotheek te gebruiken.

De volgende procedures zijn beschikbaar:

**PROCinit - Initialize the driver and display**
Deze procedure initialiseert de driver en het display. Als parameters geeft je de nummers van de GPIO pennen mee die als klok- en datalijn gebruikt worden.

**PROCshow - shows the hex digits in the data array**
Deze procedure toont de code op het display. De code is opgeslagen in de array data%. Het meest linker getal staat in element 0, het rechtse getal in element 3. Dus, om 1234 op het display te tonen voer je uit: `data%(0)=1:data%(1)=2:data%(2)=3:data%(3)=4:PROCshow`

**PROCclear - turns off all digits**
Spreekt voor zich, zet alle digits uit

**PROCsetbrightness - sets the brightness of the displays**
Stelt de helderheid van het display in. De helderheid wordt als parameter meegegeven in de aanroep van de prodedure: PROCsetbrightness(4) laat het display op (ongeveer) halve kracht branden.

**PROCsetdoublepointon - turns double point on**
Zet de dubbele punt aan.

**PROCsetdoublepointoff - turns double point off**
Vul zelf maar in ;-)  Juist, ja. Deze aanroep zet de dubbele punt uit.

Met deze basisfuncties kan je dus eenvoudig een digitale klok maken:

```
140 PROCinit(clockPin%, dataPin%)
150 data(0)=2:data(1)=1:data(2)=0:data(3)=0
155 PROCsetbrightness(7)
160 REPEAT
170   PROCshow
190   FOR s%=0 TO 59
200     T%=TIME:REPEAT UNTIL TIME-T%>95
205     PRINT s%
210     IF s% MOD 2 = 1 THEN PROCsetdoublepointon
220     IF s% MOD 2 = 0 THEN PROCsetdoublepointoff
240     PROCshow
245   NEXT s%
250   data(3)=data(3)+1
260   IF data(3)=10 THEN data(2)=data(2)+1:data(3)=0
270   IF data(2)=6 THEN data(1)=data(1)+1:data(2)=0
280   IF data(0)<2 AND data(1)=10 THEN data(0)=data(0)+1:data(1)=0
290   IF data(0)=2 AND data(1)=5 THEN data(0)=0:data(1)=0
300 UNTIL FALSE
```

Eerlijkheid gebiedt me te zeggen dat deze qua timing niet geheel accuraat is. Maar aangezien we een Pico W hebben (dus met WiFi) kunnen we Timo vragen om een NTP stack te schrijven zodat we voor de tijd een NTP server kunnen raadplegen.

Het gehele programma is te downloaden vanaf https://bbcmicro.nl/pico-klok-radio/tm1637-bbcbasic.txt. Het invoeren gaat het makkelijkst als je eerst een NEW commando opgeeft, dan de tekst vanuit de browser naar het klemboard kopieert en vervolgens in je terminalprogramma plakt. Daarna kan je het opslaan in de Pico.

# De radio module aansturen

De Pico heeft twee $I^2C$ poorten. Helaas is het mij niet gelukt om deze aan te sturen vanuit BBC Basic. Daarom heb ik een soortgelijke set van procedures als voor het display geschreven. De volgende routines bit-bangen een $I^2C$ interface in elkaar:

**PROCi2c_init – init the I2C interface**
Deze procedure initialiseert de $I^2C$ interface. Er is geen parameter nodig. Aangezien dit in Basic geprogrammeerd is ligt de klokfrequentie ergens tussen de 2 en 3 kHz, dat is ruim onder het maximum van 400 kHz.

**PROCi2c_start – send start bit**
Zet een startbit op de datalijn.

**PROCi2c_stop – send stop bit**
Sluit de communicatie af met een stopbit.

**PROCi2c_send – send byte to device**
Als parameter wordt hier een byte verwacht. Bij de communicatie wordt als eerste een adres van het aan te spreken device verstuurd. Na het versturen van de data wordt gewacht op de acknowledge van het geadresseerde device.

**FNi2c_read – read byte from device**
Deze functie leest een byte van een device.

**PROCradio_init**
Deze procedure initialiseert de radio door een aantal registers met hun standaardwaarden te vullen.

**PROCradio_tune – select channel**
Als parameter wenst deze procedure een kanaal. De radio wordt ingesteld op dit kanaal.

**PROCradio_freq – tune to frequency**
Deze procedure is een wrapper rondom radio_tune. Hierbij dient de frequentie in MHz opgegeven te worden. Deze procedure rekent die frequentie om naar het betreffende kanaal en roept dan radio_tune aan.

Met deze procedures en functies is het eenvoudig mogelijk om de radio aan te spreken. In grote lijnen komt het er op neer dat je een startbit stuurt, dan het adres, vervolgens de data en tenslotte een stopbit. Als je adres &10 gebruikt dan gaat de RDA5807 in seriële adresseringsmode. Dat wil zeggen dat de registerpointer bij het schrijven begint bij &02 en automatisch verhoogd wordt. Als je gaat lezen wordt gestart bij register &0A. De registers zijn 16 bits, daarom sturen we per register twee bytes. Het eerste is het MSB en daarna komt dus het LSB.

```
 10 REM BBC BASIC I2C
 40 PIN_SDA% = 16
 50 PIN_SCL% = 17
 60 ADDRESS%=&20
 80 REM Main program
100 configlength%=12
110 DIM config(configlength%) : REM register configuration
120 config(0)=&C1:config(1)=&03:REM register &02
130 config(2)=&00:config(3)=&00:REM register &03
140 config(4)=&0A:config(5)=&00:REM register &04
150 config(6)=&88:config(7)=&0F:REM register &05
160 config(8)=&00:config(9)=&00:REM register &06
170 config(10)=&42:config(11)=&02:REM register &07
246 PROCi2c_init(PIN_SDA%, PIN_SCL%)
248 PROCradio_init
249 INPUT "Frequentie ", f
250 PROCradio_freq(f)
260 END
```

# Basis Input/Output

Om dan toch nog even terug te keren naar de schakelaar en de led waarover ik het in het begin van dit verhaal gehad heb: in BBC Basic heb je de mogelijkheid om functies uit de C/C++ SDK te gebruiken met behulp van het SYS-statement.

Om een I/O pen te initialiseren gebruik je:

`SYS "gpio_set_dir", pin%, 0` (voor input) en `SYS "gpio_set_dir", pin%, 1` (voor output)

Om een I/O pen laag of hoog te zetten gebruik je:

`SYS "gpio_put", pin%, 0` (voor laag) en `SYS "gpio_put", pin%, 1` (voor hoog)

Om de waarde van een input-pin te lezen gebruik je:

```
SYS "gpio_get", pin% TO b%
```

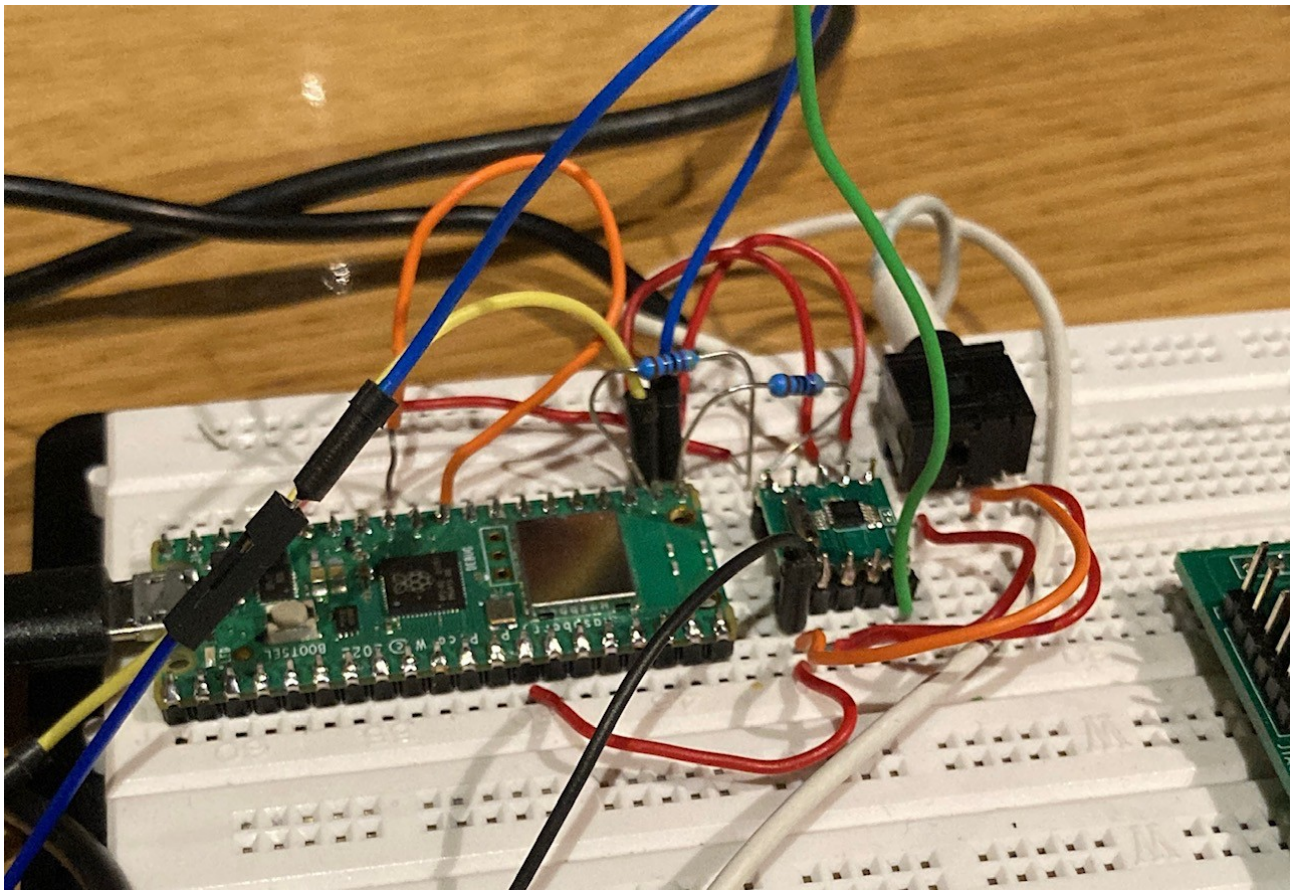
Waarbij pin% telkens het GPIO pinnummer is. Let er op dat dit iets anders is dan het pinnummer van de Pi Pico. Deze GPIO pinnummers staan overigens op de onderkant (handig!) van de Pico genoemd. De variabele b% in het laatste voorbeeld is een willekeurige variabele waar de waarde van het ingelezen bit in geplaatst wordt zodat deze waarde in het basic programma verder verwerkt kan worden.

Ik hoop dat ik u hiermee een aardige start heb kunnen meegeven om met de Pico aan de slag te gaan. Als u de onderdelen voor de klokradio in huis heeft kunt u deze op een bread board opbouwen maar u kunt ook bij mij een printje bestellen. Als er meerdere personen zijn die een printje bestellen kan ik eventueel ook complete kitjes aanleveren die u dan zelf in elkaar kunt solderen. De kosten hiervan zullen circa €10,00 bedragen. Daar kom ik later in een Asterisk nog wel op terug.

Ik wens u veel knutselplezier met uw Pi Pico!

Met de vriendelijke groeten van Roland.

roland@acornatom.nl



*Dit artikel is gepubliceerd in de Asterisk mei 2023. Dit is het clubblad van de Big Ben Club, de Nederlandse gebruikersvereniging van Acorn Computers.*