

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

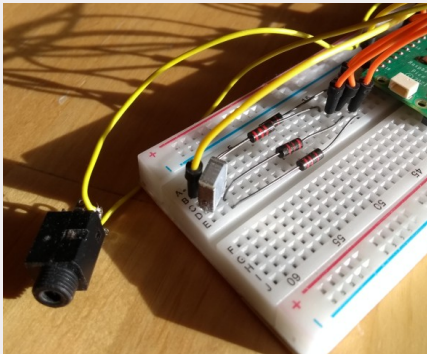
"Ik heb nu **NOFORTH** op de **PICO** draaien, wat kan ik ermee doen?"

Je kan muziek vastgelegd in de **MANX** notatie (bekend van de tingeltangels) afspelen met de PICO

"Hoe dan?"

Wat is daarvoor nodig?

- Om te beginnen: je moet geluid kunnen maken met de PICO
  - daarvoor is een stukje software nodig -- ik heb daarvoor fanfare.fs geschreven op noforth T en de PIO assembler
  - en een heel klein stukje hardware, als minimaal begin -- op dit moment 4 weerstanden, een condensator een mono jack en een USB speaker met 3.5 mm plug



### Geluid maken met de PICO

De pico heeft 8 PIO's (*Programmed I/O*) die zelfstandig kunnen draaien - die kunnen dus tonen genereren zonder dat de processor iets hoeft te doen.

Fanfare bevat PIO assembler opdrachten om de frequentie van de PIO's te berekenen en in te stellen en ze aan en uit te zetten. Daarnaast heeft fanfare woorden om muziknoten af te spelen, woorden om het tempo in te stellen en om akkoorden te maken:

#### Toonvorming:

tonen:	c d e f g a b x
kruis:	^
mol:	v
octaaf:	n 0 0+ 0-
transpositie:	n HOGER

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

akkoorden:	MAJ SUS4 AUG DIM etc... of: { } max 4 noten tussen accolades tegelijk afspelen
------------	--

### Timing:

nootduur:	/1 /2 /4 3/4 /8 3/8 7/8 /12 /16 /24 /32 /64 etc...
telsnelheid:	=tpm (tellen per minuut)
tel:	=telnoot (de noemer van de maatsoort)
rust:	z
maten:	ATTENTIE:
maatsoorten:	MS3/4 MS4/4 MS2/2 MS9/4 etc...

### Voorbeeld:

4 0 MS3/4 150 =tpm Attentie: /16 b b b z e e e z f f f z   /8 b e f f e b   /4 { c e g } { c d a } MAJ c   MEL
---

## Muzieknotatie

Nou is het aardig als je bestaande muziek kan afspelen, dan hoef je niet alles zelf te schrijven. Er zijn verschillende computer leesbare muziek notaties, bijvoorbeeld MANX, Beethoven430 of ABC.

Dus als je muziek in die notatie zou willen afspelen...

- dan moet je de muziek in MANX/Beethoven430/ABC notatie kunnen omzetten naar fanfare notatie
- en dan kan je die fanfare notatie afspelen met fanfare

Maar ja, MANX code ziet er heel anders uit:

```
PART
PDEFAULT: SILVER
||: 3 /4 measure whole rest /8 rest
|
/16 f2 e2 ges2 f2
     g2 fis2 as2 g2
/24 a2 gis2 bes2 a2 ces3 bes2 |
/24 c3 b2 des3 c3 d3 cis3
```

En Beethoven code weer anders:

pp :23T-
----------

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```
6r ' 'G#FGAB' C, E#DEGE4C2'D6C, B'CEC, A, FEFAF4D' '2E6D#CDFD, B, G#FGB' DG'D
#CD, BFE, C, B'CEG' C,
G#FGAB4' C2' F6E#DEGEC, , A#GA' C, A4F2' 'B6A#GA' C, AF, D#CDFD, B' ' 'FEFDC, B,
G#FGFDC' 'C, B' C, GEC;
```

en ABC code:

```
T: Maple Leaf Rag
C: Scott Joplin (1899)
M: 4/4
L: 1/8
"A"|: "(C7)"z2 |\
"F"zFcF Ac2F | "C7"cEGc- c4 | "F"zFcF Ac2F | "C7"cEGc- c4 |\
"Db"zF_A_d "C"zc2c | "Db"zF_A_d "C"zc3 |
```

(er zijn natuurlijk nog veel meer notaties beschikbaar:

[https://en.wikipedia.org/wiki/Category: Music\\_notation\\_file\\_formats](https://en.wikipedia.org/wiki/Category:Music_notation_file_formats) voor de liefhebbers)

Wat nu?

Met de hand vertalen? Dat blijkt behoorlijk tijdrovend te zijn. En een foutje is zo gemaakt. Maar wel een goede eerste stap om te zien wat er moet gebeuren bij een vertaling

Anders: vertalen of interpreteren van MANX, ABC en Beethoven430 met een FORTH programmaatje:

### Beschrijving MANX aanpak

- pak een beetje uitgebreid MANX .SCO bestand, bijvoorbeeld: drawals.sco
- zet dat om in een lijst van woorden met het script "sco-preprocessor":

```
# maak lijsten van tokens, gesorteerd op aantal en op alfabet
cat $1 | tr -cs .:/{ }\|1-9A-Za-z '\n' >tmp-765245
cat tmp-765245 | sort | uniq | sort >tokens.list
cat tmp-765245 | sort | uniq -c | sort -rn >tokens.counts
rm tmp-765245
```

Resultaat tokens.list:

```
{
|
||
```

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```
||:  
}  
11  
126  
/16  
16:12  
/2  
2  
27  
28  
3  
33  
/4  
/8  
a2  
Albert  
arranged  
b2  
by  
c3  
d2  
d3  
de  
DEFAULT:  
Dutch
```

Deze lijst omzetten in forth woorden die manx omzetten in fanfare woorden,  
sco2fanfare.fs:

```
\ conversie voor *.sco files naar fanfare codering  
\ principe: definieer onderstaande woorden en evalueer .sco file;  
\ niet relevante woorden worden genegeerd, de rest wordt vertaald  
\ naar fanfare codering  
\ eerste versie van de woordenlijst is gebaseerd op drawals.sco  
\ gebruik sco-preprocessor om sco files om te zetten naar tokens  
\ die gebruikt kunnen worden om onderstaande woorden aan te vullen  
\ met begrippen die nog ontbreken  
\ Het verdient aanbeveling om commentaar e.d. zoveel mogelijk te  
\ verwijderen uit de .sco file voor de transformatie  
  
variable oktaaf 0 oktaaf !  
: o space space dup oktaaf @ <> if  
    dup oktaaf ! . ." 0 "  
    else drop then ;  
  
: .. ;  
: { ." ( ) { " ; \ ( ) ivm e4thcom interpretatie van {
```

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```

: |      ." |" cr      ;
: ||     ." |" cr      ;
: ||:    ." |" cr      ;
: }      ." }"        ;
: /12    ." /12 "      ;
: 126    ." 126 =tpm " ;
: /16    ." /16 "      ;
: 16nov93 ;
: /2     ." /2 "       ;
: /24    ." /24 "      ;
: /4     ." /4 "       ;
: /8     ." /8 "       ;
: /8.    ." 3/16 "     ;
: a2     2 o ." a"      ;
: a3     3 o ." a"      ;
: a4     4 o ." a"      ;
: albert ;
: as2    2 o ." v a"    ;
: as3    3 o ." v a"    ;
: as4    4 o ." v a"    ;
: b2     2 o ." b"      ;
: b3     3 o ." b"      ;
: bell   6 o ." a"      ;
: bell-tree 6 o ." ^ a" ;
: bes2   2 o ." v b"    ;
: bes3   3 o ." v b"    ;
: bes4   4 o ." v b"    ;
: bis    ;
: boem,  ;
: boem   6 o ." b"      ;

```

... en dan krijgen we dit (sco-drawals.fs):

```

126 =tpm
ms3/4

```

attentie:

```

/1 z /8 z |
/16 2O f e v g f g ^f v a g /24 a ^g v b a 3O v c 2O v b |
/24 3O c 2O b 3O v d c d ^c v e d e ^d f e v g f g ^f v a g |
/4 { 2O f 3O f a } c 3/16 { c ^g } /16 a |
3/16 2O c /16 3O a 3/16 { c g } /16 a 3/16 { c g } /16 f |
/4 { 2O g 3O e 4O c } 3/16 3O c /16 b 3/16 { c v b } /16 a |

```

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```
/4 { 2Oc vb 3Og} c c |  
/4 { 2Og 3Oe vb} c/16 { c a} 3/16 vb |  
3/16 2Oc/16 3O vb3/16 { c a} /16 vb3/16 { c a} /16 g |  
3/16 { 2Oa 3Of 4Od} /16 c3/16 { 3Oc 4O vd} /16 c3/16 { 3Oc  
b} /16 4Oc |  
/4 { 2Of 3Oa} c c |  
/16 { 2Oa 3Oc} f/8 4Of/4 3Oc{ 2Oa 3Of 4Oc} |
```

En dat klinkt zo: (drawals)

Wat er nog ontbreekt: onder meer meerstemmigheid, maar ik ken MANX niet goed genoeg om een volledig beeld te hebben van wat er ontbreekt.

### Beschrijving Beethoven430 aanpak

Het Beethoven 430 programma van Albert en Willem (zie <https://home.hccnet.nl/anij/nof/m/bh430/music%20on%20the%20launchpad.html>) is geschreven in Noforth en hergebruik voor de PICO is gemakkelijk, vooral ook omdat het programma een gelaagde architectuur heeft met een besturingslaag voor de hardware en een interpretatielaag voor de Beethoven430 muziektaal.

De Beethoven430 hardware besturingslaag wordt vervangen door fanfare en de interpretator aangepast om fanfare woordjes te gebruiken:

```
: PLAY ( adr -- )  
\ adapted for fanfare, replaced code in blue, new code in red  
\ 0 to sd 0 to time  
\ true hx 2A ( p2dir) !  
\ 0 to OCTAVE  
\ 0 to @#  
\ XF 15 umin dup to XF XFI umin to XFI  
\ ." higher=" higher . ." xf=" xf .  
\ s1 0 .r ." /" s2 . cr  
begin  
true to IDLE?  
ch S over c@ = if SHOW space? then \ !DUTY then  
ch $ over c@ = if SHOW space? then \ !DUTY S2 S1 - to S1 then  
ch T over c@ = if SHOW space? dura 1000 * tel-us ! then  
          \ DURA to TIQ STAC >STAC then  
ch - over c@ = if SHOW 0 >STAC then \ articulation/accenten  
ch . over c@ = if SHOW space? 4 >STAC then  
dup c@ ch U ch [ within if SHOW 1- count ch T - 3 over < -  
>STAC then  
dup c@ ch 1 ch : within if >DURA 0 to IDLE? space? Then  
  \ numeric input, kept in DURA  
ch ' over c@ = if SHOW 0+ then \ 12 +to OCTAVE then
```

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```
ch , over c@ = if SHOW 0- then \ -12 +to OCTAVE then
ch # over c@ = if SHOW ^ then \ incr @# then
ch @ over c@ = if SHOW v then \ true +to @# then
dup c@ ch A ch H within if dup c@ 32 + c>NOOT-N tijd-nu (SPEEL)
  SHOW then \ dup c@ NOTE# >PERIOD TONE SHOW dup !XFI then
ch R over c@ = if SHOW tijd-nu Z then \ REST dup !XFI then
ch | over c@ = if cr? SHOW | then \ then
ch < over c@ = if cr? SHOW 0 swap dup then \ repeat
ch : over c@ = if SHOW 3 0 then \ 0 to OCTAVE then
ch ^ over c@ = if cr? SHOW 1AV then \ repeat prima volta
ch > over c@ = if cr? SHOW over if swap then nip then \ repeat
ch ; over c@ = if ( TIMING ) SHOW cr drop exit then
  \ end of input
IDLE? \ stop? or
until SHOW 1- cr u. ; \ timing tone-off true ?abort ;
```

Met deze kleine aanpassingen kunnen we Beethoven430 muziek op de PICO afspelen: ([#iibeethoven.fs](#))

Wat er nog ontbreekt: duty cycle aanpassing om de klankkleur te beïnvloeden

### Beschrijving ABC aanpak

ABC heeft weer een andere aanpak nodig. De syntax van ABC is niet erg geschikt om direct naar muziekwoorden te vertalen. ABC is inhoudelijk sterk gekoppeld met bladmuziek, dus sommige onderdelen moeten worden genegeerd. En veel zaken zoals maatsoort zijn niet strak gedefinieerd. De aanpak is een vertaalprogramma in gforth, zodat de vertaalde code nog kan worden nabewerkt in een bestand tot fanfare code. Een klein deel van de code:

```
: UNDERSTAND ( u a -- u a ) \ nr of characters in parse area and
address of parse area
dup to PARSADR
begin
  over to #PARS 0 to #ACTIONS
  \ cr ( ." >IN:" >IN @ . ) ." PARS#:" dup PARS# . ." c@:"
  dup c@ emit cr \ tracing
  \ look ahead for matches, longest first
  4 atleast if do4codes then
  3 atleast if do3codes then
  2 atleast if do2codes then
  1 atleast if do1codes then
  '; over c@ = if set-eop cr ." fin" cr exit then \ end of input
  \ check for end of parse area
  over 0< if .abcdata .s
    cr ." overshoot detected, string length now:" over . cr
```

## MANX, Beethoven430 en ABC muziek op de RP2040 (Raspberry Pico)

```

    \ should be 0 or positive, report
    swap drop 0 swap exit
    \ .. and set length to indicate end of parse area
then
over 0= if exit then \ end of parse area, do refill
#ACTIONS 0= if
    \ if no action was triggered in any of the doNcodes ...
    cr ." !! unmatched input:" NXTA cr key drop
    \ .. then just show char and continue (to prevent endless
    \   loop and to signal unprocessed input)
then
again ;
\ PARSE ( char "ccc<char>" -- c-addr u )
: PU ( -- ) \ Parse and Understand
    begin
        ' ; parse    cr ." \ " 2dup type cr    swap understand
        c@ ' ; = if drop exit then \ end of input, done
        0= if
            trace-details? if ." --get-new-data--" then
                refill    50 ms
            0= if ." --no-more-data-available--" 2drop exit then
then
    again ;
```

Met deze vertaler komt de Maple Leaf Rag er zo uit te zien (eerste stukje):

```

: maple-leaf-rag          \ met de hand toegevoegd

\ T: Maple Leaf Rag
\ C: Scott Joplin (1899)
\ M: 4/4
MS4/4 240 =tpm           \ met de hand toegevoegd
\ L: 1/8
/8
\ P: AABBAACC
\ K: F
\ "A"|: "(C7)"z2 |\
( A ) |
( (C7) 2/8 z |

\ "F"zFcF Ac2F | "C7"cEGc- c4 | "F"zFcF Ac2F | "C7"cEGc- c4 |\
( F ) /8 z 3 0 f 4 0 c 3 0 f a 4 0 2/8 c 3 0 /8 f |
( C7 ) 4 0 c 3 0 e g 4 0 c 4/8 c |
( F ) /8 z 3 0 f 4 0 c 3 0 f a 4 0 2/8 c 3 0 /8 f |
( C7 ) 4 0 c 3 0 e g 4 0 c 4/8 c |
```

En het klinkt zo: (MAPLE-LEAF-RAG)



Wat er nog ontbreekt: meerstemmigheid, herhaling van maten, begeleidingsakkoorden, juiste afhandeling van 'ties' etc...

### **Toekomstige ontwikkelingen**

Dit onderwerp, afspelen van bestaande muziek, is eigenlijk een zijstapje, wat ik wil bereiken is het realiseren van een portable muziekinstrument. Wel een belangrijk zijstapje, want het heeft me veel geleerd over muziektheoretische onderwerpen en conventies.

Onderwerpen voor de toekomst:

- meerstemmigheid en de maat herhaling
- ergonomie en bediening
- klankkleur en klankvorming
- MIDI, MPE, OSC...