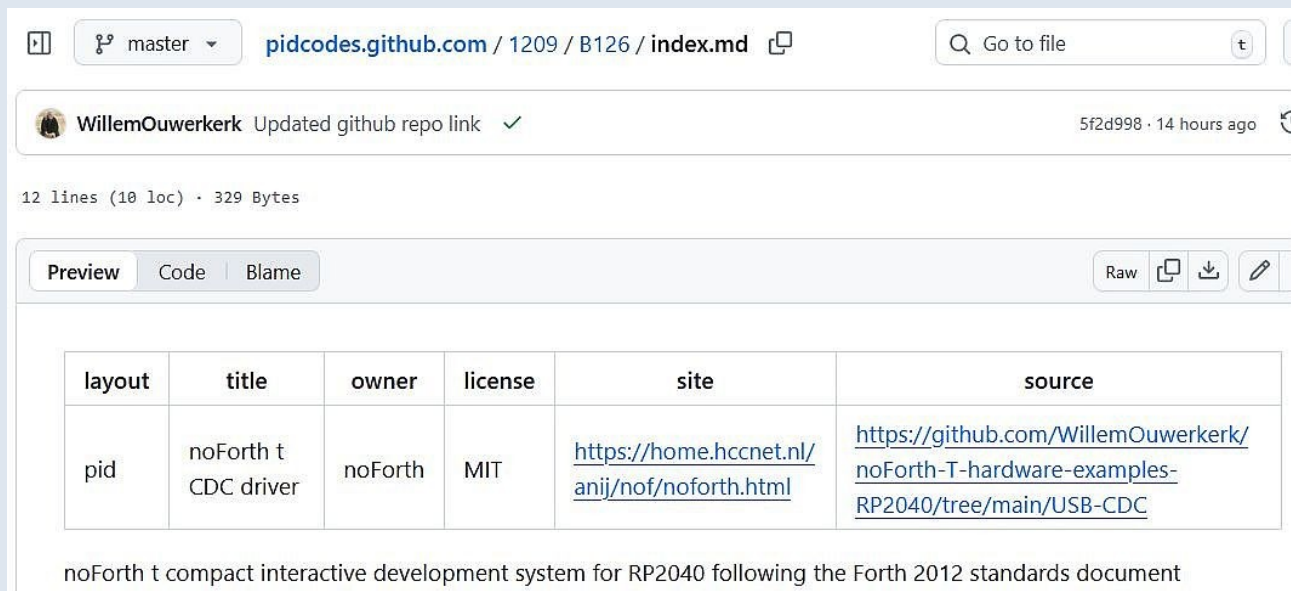


Werkgroepen verslag

USB werkgroep

De documentatie is nu ook af en onze werkgroep is daarmee afgerond. Vandaag kunnen jullie deze documentatie inkijken. De eerste versie is nu af.



12 lines (10 loc) · 329 Bytes

Preview Code Blame

layout	title	owner	license	site	source
pid	noForth t CDC driver	noForth	MIT	https://home.hccnet.nl/anij/nof/noforth.html	https://github.com/WillemOuwerkerk/noForth-T-hardware-examples-RP2040/tree/main/USB-CDC

noForth t compact interactive development system for RP2040 following the Forth 2012 standards document

We hebben nu ook een eigen Open Source Product ID.

Later in de lezing vertel ik over hoe we de low level USB ACK/NAK handshake hebben toegepast in onze driver.

De code voor de ontvangst van data in de CDC-driver. Om de driver code compact te houden is er van **:NONAME** gebruik gemaakt.

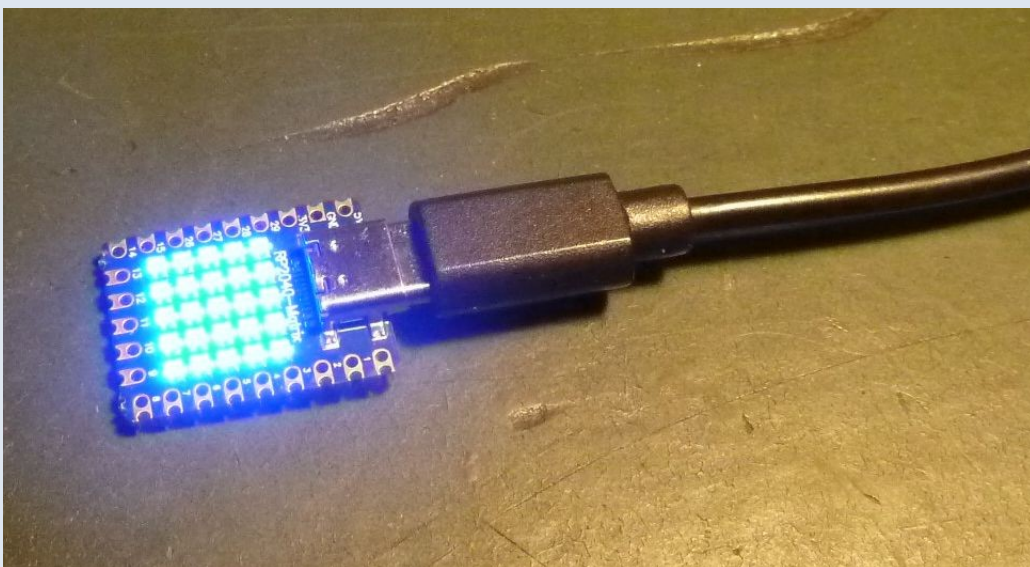
```
:noname \ USB-HANDLER ( -- ) \ Handle USB setup & receiving of chars
( start-usb ) [ r> compile, ] false usb-state ! false ( if=0 )
begin
  requests pause dup if \ Next RX packet wanted?
    5010,009C c@ >r \ Data bytes arrived in EP3
    #L #tx - r@ > \ Yes, enough space in TX buffer?
    #L #rx - r@ > and if \ And next RX packet fits too?
      ep3 >epbuf @ r@ \ Yes, fill RX buffer
      for c@+ >rx pause next
      2drop false \ Done
      ep3 prep-rcv \ And allow next RX packet
    then rdrop
  then
again ;
```

noForth documentatie

De noForth documentatie is ook geheel vernieuwd, deze kunnen jullie zo meteen inzien. Het is een hele waslijst...

- pico rp2040 dev board
- rp2040 asm notation
- starten met noforth t
- configuratie van noforth t
- commacode voor noforth t
- library v2 a short overview
- bibliotheek van noforth t
- multitasker van noforth t
- noforth t plus
- starting with the pio
- pio assembler docs v2

Voor mensen die het leuk vinden met lampjes te spelen is onderstaand RP2040 bordje heel geschikt.



noForth t ontwikkel systeem met 25 LEDjes

[Hier is hij te bestellen voor nog geen €7,50:](#)

<https://nl.aliexpress.com/item/1005005861666155.html>

NEED

Weer een woordje uit de bibliotheek. Dit maal eentje om de stacks van achtergrond taken te manipuleren.

Stacks benaderen van achtergrond taken

```
v: inside also extra definitions

: STK      ( +n task -- addr )
  ?task >r  ?dup if          \ Not TOS?
    cells negate            \ Yes, address of Nth element
    r> his s0 @ cell- +
  else
    r> his trp @ cell+ cell+ \ No, read address TOS
  then ;

v: fresh
```

Voorbeeld:

Let op, de referentie voor de stack diepte is altijd op het punt waar van taak gewisseld wordt. Het punt waar **PAUSE** staat of een woord waar **PAUSE** in opgenomen is zoals **MS**

```
task: ONE          \ Definieer een taak en code, en start deze
: TELLER 0 begin 1+ 100 ms again ;
' teller one start-task
```

```
1 one stk @ . many \ Lees en toon het 2e stack element (de teller)
0 1 one stk !      \ Zet de teller weer op nul
```

Het Forth klasje

*/	123.45	.F	2DROP	procenten	D+	Celsius
D.	dubbels	DNEGATE	Fahrenheit	π	inhoud	12345.
tweecellig	2>R	schaling	coördinaten	breuken	S>D	
2DUP	vaste komma	vorm	eencellig	gemengd		

Onderwerpen:

- Turnkey in (no)Forth
- Parsing [IF] etc.
- */ gebruik voor schaling en vaste komma
- EVALUATE
- WORDLIST (VOCABULARY)
- EXECUTE
- INVERT vs NOT (13 april 2024)
- MIN vs UMIN (13 april 2024)
- Dubbele getallen
- CATCH gebruiken
- KEY
- OK lus
- Getal uitvoer bouwen met <# # #S #>
- +LOOP voor string of COUNT
- CREATE DOES>
- Control flow stack
- S"
- BEGIN WHILE WHILE REPEAT THEN
- []
- COMPILE,

Vandaag behandelen we de
programmeer opdracht: Invoerstream