

Dual USB op noForth t met spinlock's

```
COM40:115200bps - Tera Term VT
File Edit Setup Control Window Help

* noForth tv# duo 250902 *
M0+ core with ARMv6-M ISA

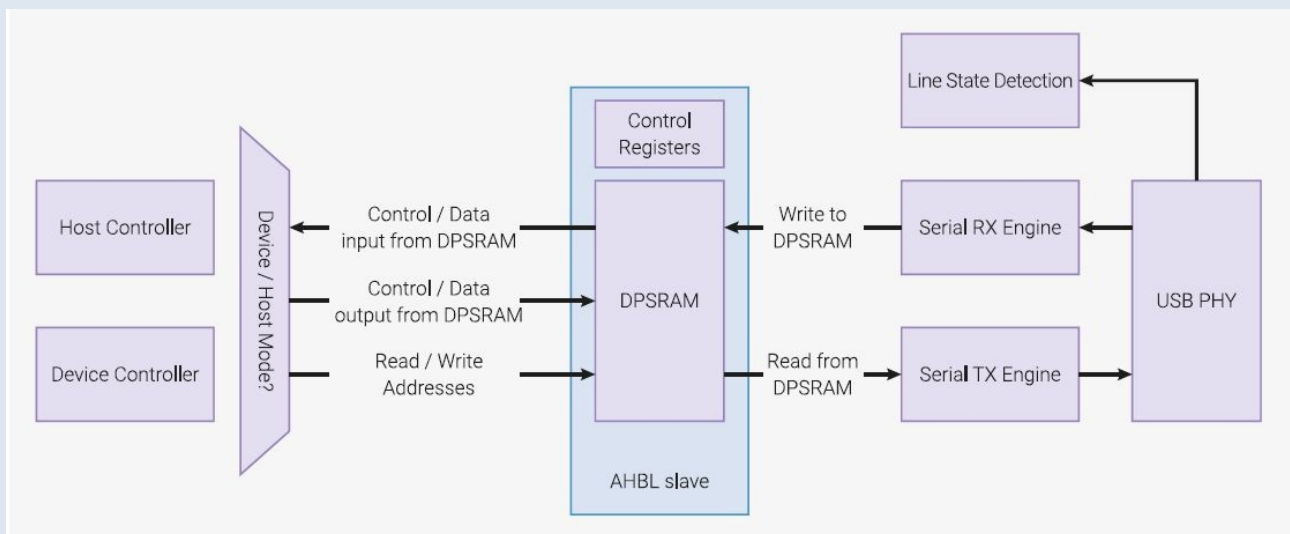
*
OK.0
@>words
TOOLS\ [THEN] [IF] [ELSE] .S NEED\ CDC\ NOFORTH\ TO DOES> [']
IMMEDIATE ;CODE CODE REPEAT ELSE WHILE BEGIN THEN WORDLIST [CHAR]
\ ' EVALUATE VOCABULARY QUIT MS >NUMBER FIND .( ( CHAR PARSE
REFILL ACCEPT " S" C" DEPTH ] [ RECURSE POSTPONE /STRING HEX
DECIMAL .R U.R . U. D. #> #5 # SIGN HOLD <# TYPE SPACES SPACE
CR EMIT KEY? KEY CELLS CELL+ >BODY ALIGN COMPILER, C, , ALLOT
HERE ALIGNED ABORT CATCH THROW MOVE FILL */ */MOD MOD / /MOD
FM/MOD M* D+ DABS DNEGATE ABS NEGATE D2/ D2* - + 2/ 2* 1- 1+
RSHIFT LSHIFT INVERT XOR OR AND S>D 0<> 0= <> = 0< WITHIN U<
< U> > FALSE TRUE UMAX UMIN MAX MIN ROT PICK NIP TUCK SWAP
OVER DROP ?DUP DUP 2SWAP 2OVER 2DUP 2DROP 2R@ R@ 2R> 2>R R>
>R COUNT @ C@ +! ! C! SOURCE-ID >IN STATE BASE BL UM/MOD *
UM* I +LOOP LOOP UNLOOP LEAVE ?DO DO AGAIN UNTIL IF AHEAD ;
2LITERAL LITERAL VALUE VARIABLE CONSTANT :NONAME : CREATE EXIT EXECUTE
OK.0
@>

COM41:115200bps - Tera Term VT
File Edit Setup Control Window Help

+ noForth tv# duo 2 +
M0+ core with ARMv6-M ISA

+
OK.0
@>words
TOOLS\ [THEN] [IF] [ELSE] .S NEED\ CDC1\ NOFORTH\ TO DOES> [']
IMMEDIATE ;CODE CODE REPEAT ELSE WHILE BEGIN THEN WORDLIST [CHAR]
\ ' EVALUATE VOCABULARY QUIT MS >NUMBER FIND .( ( CHAR PARSE
REFILL ACCEPT " S" C" DEPTH ] [ RECURSE POSTPONE /STRING HEX
DECIMAL .R U.R . U. D. #> #5 # SIGN HOLD <# TYPE SPACES SPACE
CR EMIT KEY? KEY CELLS CELL+ >BODY ALIGN COMPILER, C, , ALLOT
HERE ALIGNED ABORT CATCH THROW MOVE FILL */ */MOD MOD / /MOD
FM/MOD M* D+ DABS DNEGATE ABS NEGATE D2/ D2* - + 2/ 2* 1- 1+
RSHIFT LSHIFT INVERT XOR OR AND S>D 0<> 0= <> = 0< WITHIN U<
< U> > FALSE TRUE UMAX UMIN MAX MIN ROT PICK NIP TUCK SWAP
OVER DROP ?DUP DUP 2SWAP 2OVER 2DUP 2DROP 2R@ R@ 2R> 2>R R>
>R COUNT @ C@ +! ! C! SOURCE-ID >IN STATE BASE BL UM/MOD *
UM* I +LOOP LOOP UNLOOP LEAVE ?DO DO AGAIN UNTIL IF AHEAD ;
2LITERAL LITERAL VALUE VARIABLE CONSTANT :NONAME : CREATE EXIT EXECUTE
OK.0
@>
```

noForth tv# duo via USB CDC



RP2040 USB controller

Opgeloste dual CDC pijnpunten

1. Het fabrikant- en het produkt-ID zijn aangepast (VID & PID)
2. Bescherming van lees/schrijf acties (**spinlock's**)

VID & PID voor dual CDC

Hebben we het vorige keer over gehad. Door een VID & PID toe te passen waar de host (PC) iets mee kan, wordt een correcte USB CDC driver geladen. Wil je meer dan dual CDC doen, dan moet je zelf een werkende driver bouwen. Op Windows kan dat door een *.INF bestand te schrijven. Weet je niet hoe je dat moet doen, vraag het aan b.v. chatGPT of een andere AI-agent

Driver type	VID	PID
TinyUSB dual CDC	CAFE	4020
VOTI CDC-ACM	16C0	27DD
Open source noForth t ID	1209	B126

Beschermen van lees/schrijf acties

Vorige keer bleek dat bij gebruik van meerdere processoren problemen ontstaan. In dit geval kunnen beide processoren tegelijkertijd dezelfde geheugen locatie proberen te veranderen. Zie hier het gevolg:

```
501006A0 000 0BA 0BA
501007AC 000 050 050
501006A0 000 0BA 0BA
501007AC 000 050 050
501006A0 000 0BA 0BA
501007AC 000 050 050
501006A0 000 0BA 0BA
501007AC 000 050 050 ← Ok
501006A0 000 0BB 0BB
501007AC 000 0C2 0CA ← Fout
501006A0 000 0BB 0BB
501007AC 000 0C2 0CA
```

De pointers van de ringbuffers raken dan ontregeld, waardoor onze CDC-driver niet of niet correct meer functioneert.

De oplossing bleek het spinlock-device van de RP2040. Er zijn er 32, en met twee stuks daarvan werd onze driver stabiel.

Wat is een spinlock

Een spinlock is een geheugencel met een speciale eigenschap. Is hij vrij, dan geeft hij bij lezen de waarde één terug en wordt hij gelijk bezet. Is hij bezet dan geeft elke volgende lees actie een nul. Wordt er wat dan ook naar de spinlock geschreven, dan wordt hij weer vrij gegeven.

Lezen beide processoren de spinlock tegelijkertijd, dan wint processor-0. Dat betekent dat de tweede processor een nul te zien krijgt, ofwel deze spinlock is bezet. Een voorbeeld:

```
: MINE? ( +n -- s ) cells D000,0100 + @ ;  
: YOURS ( +n -- ) cells D000,0100 + dup ! ;
```

Data tabellen voor de dubbele CDC

\ Name:	CNT	ORG	PKT	PID	INBUF-CTRL	OUTBUF-CTRL	EPxBUF
\ Offsets:	00	04	08	0C	10	14	18
create EP0	4	cells	allot		5010,0080	dup , 4 + ,	5010,0100 ,
\ CDC-0							
create EP1	4	cells	allot		5010,0088	dup , 4 + ,	5010,0180 ,
create EP2	4	cells	allot		5010,0090	dup , 4 + ,	5010,0200 ,
create EP3	4	cells	allot		5010,0098	dup , 4 + ,	5010,0280 ,
\ CDC-1							
create EP4	4	cells	allot		5010,00A0	dup , 4 + ,	5010,0300 ,
create EP5	4	cells	allot		5010,00A8	dup , 4 + ,	5010,0380 ,
create EP6	4	cells	allot		5010,00B0	dup , 4 + ,	5010,0400 ,

Ontvangen van data

```
: ENDPOINTS ( 0 -- if ) \ Handle used endpoints
  5011,0058 @           \ Read interrupt flags
  dup 04 and if zlp> then \ EP1 active?
  dup 5011,0058 !       \ Clear active flags
  2080 and or ;         \ EP3 and/or EP6 active?
```

```
: USB-RX1 ( if0 -- if1 ) \ Restart RX1 after reception
  dup 2000 and if         \ EP6 active?
  5010,00B4 c@ >r        \ #Data arrived in EP6
  #L #tx1 - r@ >         \ Enough space in TX1 buffer?
  #L #rx1 - r@ > and if   \ And next RX1 packet fits too?
  0 mine? if             \ Buffer available?
    ep6 >buf @ r@        \ Yes, fill RX1 buffer
    for c@+ >rx1 next drop
    0 yours 8 us 2000 xor \ Done
  ep6 usb-receive        \ Allow next RX1 packet
  then
  then rdrop
  then ;
```

Een bit op de stack geeft aan, dat een pakket ontvangen is op EP6. Dit pakket wordt alleen bevestigd als het in de ringbuffer past en deze niet (door de andere processor) wordt gebruikt.

Hierdoor gebruikt onze driver de laag niveau ACK/NAK USB handshake, zoals die in het USB protocol zit. Met Windows betekent dit, dat je ongeveer 0,6 seconden vertraging per ontvangen USB pakket mag hebben. Effectief zorgt dit voor een bruikbare handshake, zolang de host maar niet een hele file ineens stuurt.

Een toets lezen

```
: USB-KEY ( -- c )
  begin pause #rx1 0> 0 mine?
  and until
  rx1> 0 yours ;
```

Het afhandelen van een toets is ook beschermd. Een toets lezen is alleen toegestaan, wanneer de andere processor geen USB ontvangst aan het afhandelen is.

Data verzenden

```
: USB-TX1      ( -- )
  begin 4 us usb1? until 0A ms
  begin
    begin pause usb1? until \ Still connected?
    #tx1 if \ EP5 Any chars to send?
      ep5 gone? if \ Yes, previous packet gone?
        20 us 1 mine? if \ TX1 buffer free to use
          ep5 >buf @ #tx1 40 umin \ Packet place & size
          2dup ep5 prepare bounds \ Init. transmit pointers
          ?do tx1> i c! loop 1 yours \ data to EP5-buffer
          ep5 usb-packet \ Send to host
        then
      then
    then
  again ;
```

Een karakter uitvoeren

```
: USB-EMIT ( c -- )
  begin pause #L #tx1 - until \ Is there space in send ring buffer
  begin pause 1 mine? until \ Wait for access allowed?
  >tx1 1 yours ; \ Put char in ring buffer and free buffer
```

De CDC0 afhandelaar is nagenoeg identiek. Natuurlijk met gebruik van andere endpoints en ringbuffers.

Een korte demo...

- 1) Start noForth t duo en open twee terminals
- 2) WORDS en CHAPTERS in beide vensters
- 3) Laad assembler vanuit library
- 4) Laad decompiler via drag en drop

Nog vragen?