430eForth43n6vis – VOCs and ITEMs HOWTO – Creating a DS1621 Library

Manfred Mahlow

Zusammenfassung

Some decades ago, when I first discovered Forth and became familiar with the vocabulary concepts of Fig-Forth and F83, I was missing a way to bind a vocabulary (a namespace) to a word representing data, to create data types or simple objects.

Over the years I found a way to do it without using state-smart words, first for PC Forth Systems and during the last years also for Flash based MCU Forth Systems. VOCs and ITEMs [2] are the core elements of the final concept and 430eForth43n6vis [3] is at present the smallest Forth System fully supporting both.

This HOWTO should give an idea, how to use VOCs and ITEMs (on a MSP 430 LaunchPad) to create a library.

A Library for the DS1621 Digital Thermometer and Thermostat

The DS1621 [1] provides 9-bit readings, which indicate the temperature of the device. The device has three temperature registers, T for the last measured temperature, TL and TH for thermostat settings. Data is read/written via a 2-wire serial interface (open drain I/O lines). Three address inputs allow to address 8 devices per bus. The base address is 48h. The device operation is controlled by an 8-bit configuration register. Two conversion modes are supported, a 1SHOT mode and a continous mode and the thermostat status is available as a digital output.

1 I2C (2-WIRE) Library required

The DS1621 is an I2C Bus device. So some words from an I2C Library are required for read and write access.

```
\ DS1621 Digital Thermometer and Thermostat with I2C Bus Interface      MM-180804
\
\ Up to 8 DS1621 devices can be connected to a single I2C bus.
\

#require I2C    \ loading an I2C library  ( I2C Bus Master )

{ The following words are used from the I2C library:

  I2C ?ADDR ( u a1 -- a2 )
  Calculate the I2C slave address a2 from a given base address (a1) and a
  modfier (u). Throw an error if u > 7.

  I2C ?BUSY ( -- )
  Wait while the I2C Bus Master is busy (required when using I2C hardware).

  I2C INIT ( -- )
  Initialise the I2C Bus interface.

  I2C READ ( a +n sid -- c1 .. cn )
  Read +n bytes from address a of the I2C slave sid, sid is the slaves 7 bit
  identifier or address. An ambiguous condition exists for n = 0.

  I2C WRITE ( cn .. c1 +n sid -- )
  Write +n bytes to the I2C slave sid. sid is the slaves 7 bit id or addr. An
  ambiguous condition exists for n = 0.
}
```

2 A Namespace for the DS1621 Library

A separate namespace is used for the library words. It's created with a vocabulary prefix (VOC), so that all library words can be accessed with this prefix.

```

  ONLY FORTH DEFINITIONS  DECIMAL

  VOC DS1621  \ the library namespace
```

3 DS1621 read/write access

Words to write to and read from a DS1621 device:

```
  DS1621 DEFINITIONS  HEX

\ Return the I2C slave identifier of DS1621 device u ( u = [0-7] ).
  : SID ( u -- sid ) $48 I2C ?ADDR I2C ?BUSY I2C INIT ;

\ Read n bytes from address a of DS1621 device u ( u = [0-7] ).
  : READ ( u a +n -- c1 .. cn )
    ROT DS1621 SID ( a +n sid ) I2C READ ;

\ Write n bytes to address a of DS1621 device u ( u = [0-7] ).
  : WRITE ( cn -- c1 u a n+1 -- )
    ROT DS1621 SID ( cn .. c1 n+1 sid -- ) I2C WRITE ;
```

4 DS1621 CONFIG/STATUS Register

The DS1621 configuration and status is hold in an 8-bit register. It's the only 8-bit register in that device. So creating a data type for 8-bit DS1621 registers makes no sence here but using a vocabulary prefix makes sence.

```

  DS1621 DEFINITIONS

  VOC CONFIG  \ CONFIG/STATUS Register access

  DS1621 CONFIG DEFINITIONS

\ Read the CONFIG/STATUS register of the DS1621 device u (u = [0-7]).
  : READ ( u -- c ) $AC 1 DS1621 READ ;

\ Write the byte c to the CONFIG register of the DS1621 device u (u = [0-7]).
  : WRITE ( c u -- ) $AC 2 DS1621 WRITE ;
```

5 DS1621 Temperature Registers

The DS1621 has three 16-bit temperature registers using a special format. So it makes sence, to use a VOC to create a data type for this registers.

5.1 Defining a data type for the temperature registers

```
  DS1621 DEFINITIONS

 VOC TR16  \ Temperature Register acccess

  DS1621 TR16 DEFINITIONS

\ Read from the temperature register a of the DS1621 device u (u = [0-7]).
  : READ ( u a -- msb lsb ) 2 DS1621 READ ;

\ Write to the temperature register a of the DS1621 device u (u = [0-7]).
  : WRITE ( msb lsb u a -- ) >R >R SWAP R> R> 3 DS1621 WRITE ;

\ Read and print the temperature from register a of the DS1621 device u.
  : ? ( u a -- )
    BASE @ >R DECIMAL  DS1621 TR16 READ ( msb lsb )
    IF [ CHAR 5 ] LITERAL ELSE [ CHAR 0 ] LITERAL THEN
    <# HOLD [ CHAR . ] LITERAL HOLD ( S>D) #S #> SPACE TYPE  R> BASE ! ;

\   Note: # and #S expect single not double precision numbers in 430eForth.

\ Create an identifier for a DS1621 temperature register with address a.
  : ITEM: ( "name" a -- ) \ name ( -- a , NS: DS1621 TR16 )
    ITEM CONSTANT ;

  DS1621 DEFINITIONS
```

5.2 Using the data type for the temperature registers

```
  DS1621 DEFINITIONS

\ DS1621 Temperature Registers :
  $AA DS1621 TR16 ITEM: T  ( -- a , NS: DS1621 TR16 )
  $A1 DS1621 TR16 ITEM: TH ( -- a , NS: DS1621 TR16 )
  $A2 DS1621 TR16 ITEM: TL ( -- a , NS: DS1621 TR16 )
```

6 Control and Monitoring

```
\ Start the temperature conversion at DS1621 device u ( u = [0-7] ).
\ Default on power up is the continous conversion mode.
  : START ( u -- ) $EE 1 DS1621 WRITE ;

\ Stop the tempeature conversion at DS1621 device u ( u = [0-7] ).
  : STOP ( u -- ) $22 1 DS1621 WRITE ;

\ Read the conversion status of DS1621 device u ( u = [0-7] ). Conversion
\ complete if f <> 0, in progress otherwise.
  : DONE ( u -- f ) DS1621 CONFIG READ 080 AND ;

FORTH DEFINITIONS  DECIMAL

\ Last Revision: MM-191212 minor changes
\                MM-181120 converted to upper case
\                MM-180906 changed to use ITEMs
\                MM-180804 changed to used 430eForth43n1vi VOCs
```

7 Examples how to access a DS1621 Device

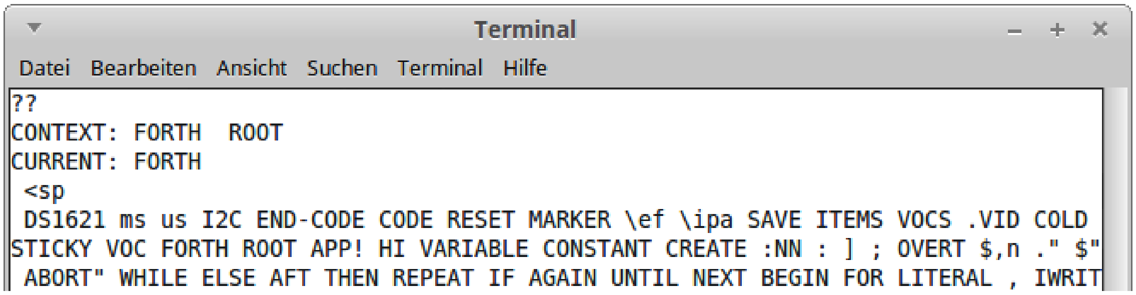
```

\\ Some examples how to access a DS1621 sensor:

1. Direct access with the sensor id:

   ( u -- ) DS1621 <method>
```

```


2. Access with an context switching constant:

   ( u –– ) DS1621 ITEM CONSTANT TS

   TS ( u –– , NS: DS1621 ) <method>


3. Access with a context switching word defined with a colon definition:

   DS1621 ITEM : TS ( –– u ) ....... ;

   TS ( –– u , NS: DS1621 ) <method>


4. Using a sensor defining word:

   DS1621 DEFINITIONS

   : SENSOR ( "name" u –– ) ITEM CONSTANT ;

  FORTH DEFINITIONS

   ( u –– ) DS1621 SENSOR TS
```

8 The DS1621 Library in a Forth Terminal

8.1 ??

is the dictionary browser that shows the current search order, the compiler context, the data stack and the words of the top wordlist in the seach order (not all words are shown here).

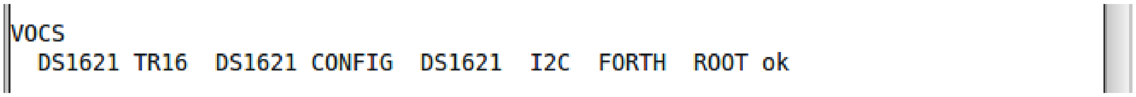


```
??
CONTEXT: FORTH  ROOT
CURRENT: FORTH
 <sp
 DS1621 ms us I2C END-CODE CODE RESET MARKER \ef \ipa SAVE ITEMS VOCS .VID COLD
STICKY VOC FORTH ROOT APP! HI VARIABLE CONSTANT CREATE :NN : ] ; OVERT $,n ." $"
 ABORT" WHILE ELSE AFT THEN REPEAT IF AGAIN UNTIL NEXT BEGIN FOR LITERAL , IWRIT
```

DS1621 and I2C are the libraries that were loaded into the FORTH namespace.

8.2 VOCS

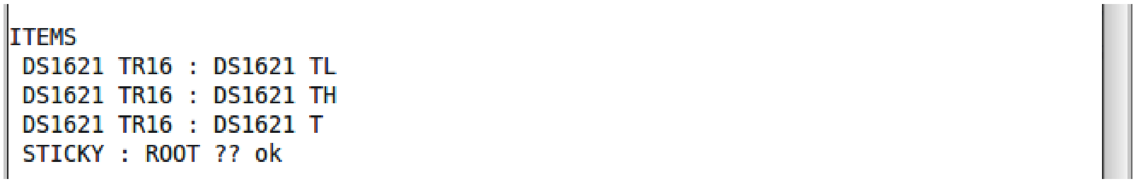
lists all vocabulary prefixes (all namespaces) that are defined in the dictionary.



```
VOCS
  DS1621 TR16  DS1621 CONFIG  DS1621  I2C  FORTH  ROOT ok
```

8.3 ITEMS

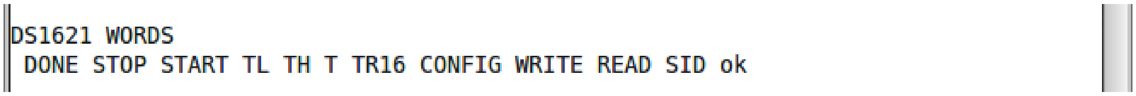
lists all context switching items that are defined in the dictionary.



```
ITEMS
 DS1621 TR16 : DS1621 TL
 DS1621 TR16 : DS1621 TH
 DS1621 TR16 : DS1621 T
 STICKY : ROOT ?? ok
```

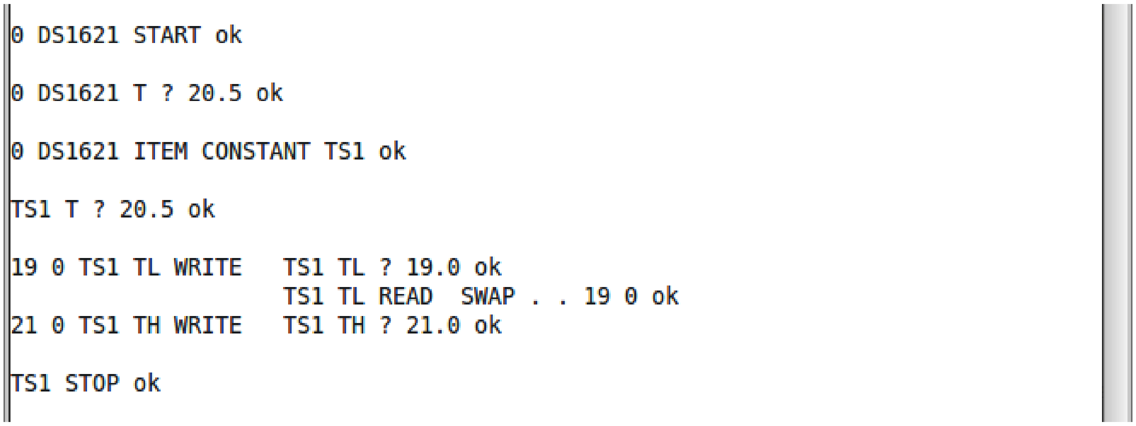
?? is the dictionary browser defined in the ROOT namespace. It's a STICKY word, i.e. it does not change the actual search order.

8.4 DS1621 WORDS



```
DS1621 WORDS
 DONE STOP START TL TH T TR16 CONFIG WRITE READ SID ok
```

8.5 DS1621 connected as device 0



```
0 DS1621 START ok

0 DS1621 T ? 20.5 ok

0 DS1621 ITEM CONSTANT TS1 ok

TS1 T ? 20.5 ok

19 0 TS1 TL WRITE   TS1 TL ? 19.0 ok
                    TS1 TL READ  SWAP . . 19 0 ok
21 0 TS1 TH WRITE   TS1 TH ? 21.0 ok

TS1 STOP ok
```

Literatur

[1] DS1621 Data Sheet
https://pdfserv.maximintegrated.com/en/ds/DS1621.pdf

[2] Manfred Mahlow, *VOCS, ITEMS und STICKY Words*, Forth Tagung 2019
https://wiki.forth-ev.de/lib/exe/fetch.php/events:ft2019:vocs_items_und_sticky_words.pdf

[3] Manfred Mahlow, *Namespaces and Context Switching for a Tiny Forth*, VD 4/2019