

BBC MicroBit - now runs Mecrisp Forth under Windows

– and try VFXTESTAPPbit as a first Test Program

Juergen could get hold of a MicroBit via his STEM Ambassador Activity in the UK. Matthias Koch agreed to give it a go and try to port Mecrisp to the MicroBit - hardly any documentation was available then - and as result a MicroBit changed country to visit Germany. A couple of weeks later the port was successfully finished. As a first step to run Mecrisp under Linux. And we tested it here in the UK as well.

But it should run under Windows as well, so some more head scratching was required. Jan Coombs and Ulrich Hoffmann got interested and found the right links. And now anybody can run Mecrisp Forth under Linux or as well now under Windows by installing it on the MicroBit.

Mecrisp can be found at <http://mecrisp.sourceforge.net/>

All of the information to set up the serial interface is available on the BBC MicroBit website at <https://www.microbit.co.uk/td/serial-library>

To install the Mecrisp code is easy, but takes some time.

5 steps have to be executed:

install the serial library – get and install the Mecrisp Forth Program - get and install TeraTerm – get, install and set up TeraTerm – download and install VFXTESTAPPbit.

Then start playing using VFXTESTAPP – more examples to follow.

1 Connect the MicroBit to your PC via the USB cable:

- the yellow light at the back of the Microbit will indicate power
- a new folder with the name MicroBit will show on the screen of the PC.

The MicroBit appears on the PC as a memory stick

2 Files:

DETAILS.TXT which contains data about the MicroBit software

DAPLink Firmware - see <https://mbed.com/daplink>

Version: 0234

Build: Oct 12 2015 14:53:22

MICROBIT.HTM , which is a link to <https://www.microbit.co.uk/>

2 Click on this link to go to the MicroBit website, which then leads to www.microbit.org

This activity is not needed to install Mecrisp Forth, but to look around on the MicroBit website

3 BUT, more important for us now is to get Mecrisp Forth installed and running;
a serial library is needed, which will allow for the MicroBit to talk to the PC via the serial USB.

Look for the serial library link <https://www.microbit.co.uk/td/serial-library>

Follow the instructions at <https://developer.mbed.org/handbook/Windows-serial-configuration>

in order to install the device driver

and follow the instructions there.

This will download the file **mbedWinSerial_16466.exe** onto your PC.

Double click on it to run the program

The installation will take a couple of minutes, so be patient

4 Now download the Mecrisp Hex Program File **mecrisp-stellaris-microbit.hex**

It is located at the same site where you have found the PDF you read now.

Save this file onto your PC desktop for easy access.

When you connected the Microbit to the PC, a MicroBit Folder had opened.

Copy the file **mecrisp-stellaris-microbit.hex** from your desktop and paste it into this MicroBit folder

Something funny will happen:

Mecrisp.hex will be visible shortly and then disappear.

This is normal. The MicroBit has transferred the information from this folder to a place on the Microbit.

There is a lot more information about Mecrisp available, we just want to keep it short here.

See the full information at <http://mecrisp.sourceforge.net/> You can see the microbit files as part of the package.

5 Now we have to set up the Terminal Function on the PC:

we suggest TeraTerm.

Start TeraTerm with a setting of 11520 Baud. 8 (bits) N (no parity) 1 (stop bit)

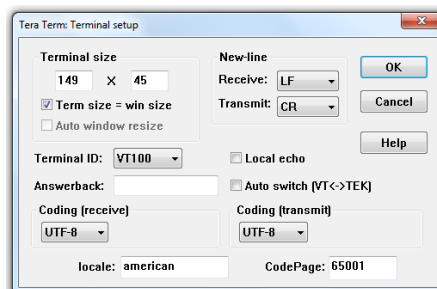
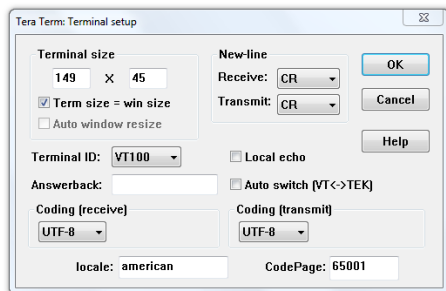
select LF, the default was CR

This is the serial Terminal Function needed, as the Microbit will communicate with the PC, and the PC will act like a keyboard and a display for the MicroBit.

If you had not installed TeraTerm before, you can download it from the link shown in the PDF and click on the exe <https://www.microbit.co.uk/td/serial-library>

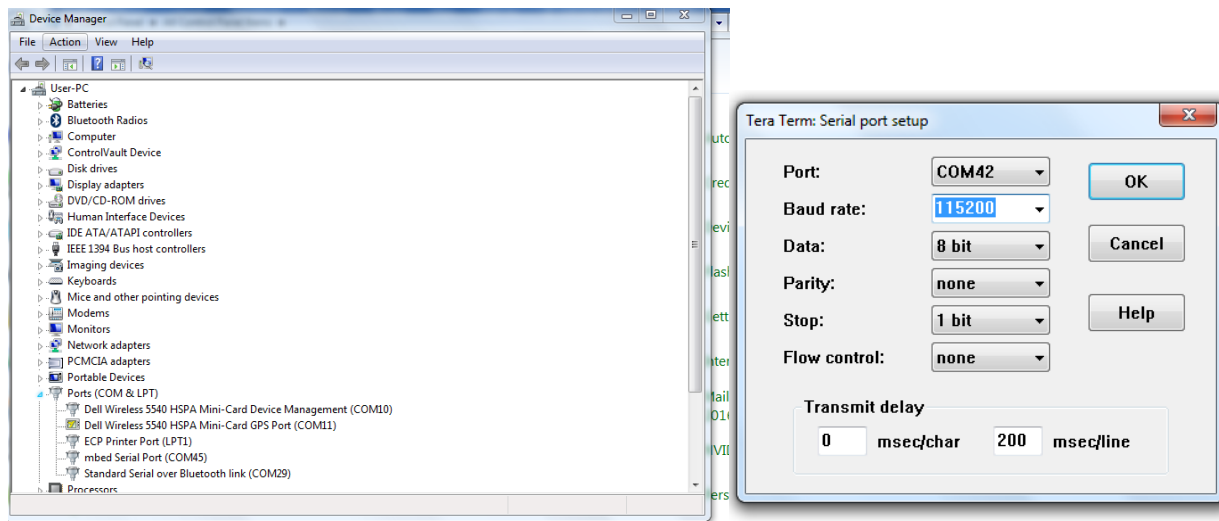
BEFORE

AFTER



Select the correct Port. Open the Device Manager on your PC, under Ports(COM&LPT) you see here mbed Serial Port (COM45), this might be a different COMxx on your PC.

Open Port on TeraTerm, select e.g. 45 from the drop down menu, TeraTerm had suggested 42 here



Then push the RESET BUTTON on the back of the MicroBit and you should see the

ok on your screen.

This indicates that Mecrisp Forth is running now and is waiting for your commands.

Hit the return key again, and a new **ok** should show

Now type into the TeraTerm window

: Hello ." hello forth world " ; do not forget the spaces - and hit return

The answer is again **ok**

Now type `hello` and hit the return key, and on the screen you can see

`hello forth world`

each time you hit the *hello* `<cr>` you will see the same message.

Now download the file `VFXTESTAPPbit.f`, and save onto the desktop. Open it with a suitable ASCII editor program – this program I can actually open with Microsoft Word.

Then I selected the whole text, copy it, and paste it into the TeraTerm Window.

A few things will happen, which indicate that Mecrisp is “digesting” this information copied in.

And if all went well you should hear the PC Bell signal, and then another hello Forth World, the display will change and you see some text and a series of 0 and ones altogether 16.

All of the explanations you can find at <https://wiki.forth-ev.de/doku.php/en:projects:a-start-with-forth:staterere> it is for MPE VFX and not for MicroBit, but the rest is the same.

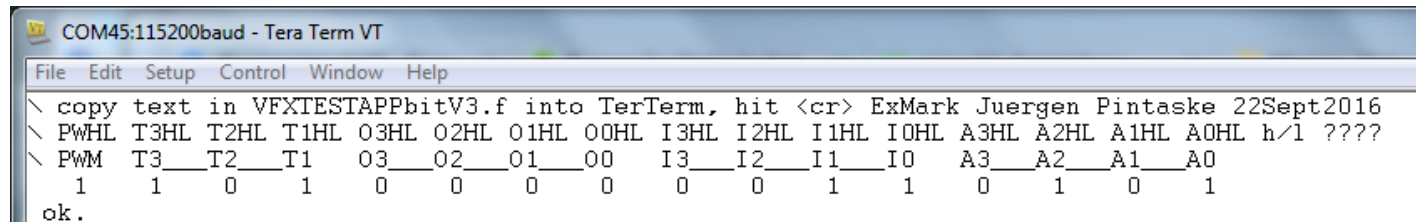
But anybody who does not have a MicroBit can play with VFXTESTAPP by downloading the exe and learn a bit about Forth as well – no age limits. Just download and run – no installation required. More information will be made available as time allows.

There, the 35 Words of Forth used for this little application are explained.

Set and reset the bits, start a counter – fixed speed or set speed, try the logic examples and there is as well a small debugger. Have fun and we will add more examples as time allows.

No warranty given – this is spare time work. Complaints, corrections, additions, expansions to epldfpga@aol.com. They will be dealt with as spare time allows.

Our small application:



```
COM45:115200baud - Tera Term VT
File Edit Setup Control Window Help
\ copy text in VFXTTESTAPPbitV3.f into TerTerm, hit <cr> ExMark Juergen Pintaske 22Sept2016
\ PwHL T3HL T2HL T1HL 03HL 02HL 01HL 00HL I3HL I2HL I1HL IOHL A3HL A2HL A1HL A0HL h/1 ???
\ PWM T3__T2__T1__ 03__02__01__00__ I3__I2__I1__I0__ A3__A2__A1__A0__
1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1
ok.
```

This **FORTH LOVE IF HONK THEN** was a secret sticker the Forth community has used over the last 30+ years, and you will understand soon what it means:

FORTH LOVE IF HONK THEN - FORTH LOVE IF HONK THEN - FORTH LOVE IF HONK THEN - FORTH LOVE IF HONK THEN - FORTH LOVE IF HONK THEN - FORTH LOVE IF HONK THEN


```

$4004A000 constant PORTB_PCR
$4004A080 constant PORTB_GPCLR
$4004A084 constant PORTB_GPCHR

$4004B000 constant PORTC_PCR
$4004B080 constant PORTC_GPCLR
$4004B084 constant PORTC_GPCHR

$4004C000 constant PORTD_PCR
$4004C080 constant PORTD_GPCLR
$4004C084 constant PORTD_GPCHR

$4004D000 constant PORTE_PCR
$4004D080 constant PORTE_GPCLR
$4004D084 constant PORTE_GPCHR

: init                                \ initialize
  Cr                                  \ send a cr first
  Flamingo                            \ then the flamingo
  Cr                                  \ then a cr again
  ." Have a nice day !" cr            \ and Have anice day plus a cr
;

: init-led                            \ initialize LED
$0100 1 4 * PORTD_PCR + ! \ Port D1 als GPIO aktivieren
                                \ activate Port D1 as GPIO
$2 GPIOD_PDDR ! \ Port D1 als Ausgang schalten
                                \ switch Port D1 as Output
$0100 18 4 * PORTB_PCR + ! \ Port B18 als GPIO aktivieren
                                \ activate Port B18 as GPIO
$0100 19 4 * PORTB_PCR + ! \ Port B19 als GPIO aktivieren
                                \ activate Port B19 as GPIO
1 18 lshift 1 19 lshift or GPIOB_PDDR ! \ Port D1 als Ausgang schalten
                                \ Switch Port D1 to Output
;

: Bunt
  init-led
  begin
    key
    dup
    case
      [char] r of 1 18 lshift GPIOB_PTOR ! endof
      [char] g of 1 19 lshift GPIOB_PTOR ! endof
      [char] b of          2 GPIOD_PTOR ! endof
    endcase
    27 =
  until
;

: systick ( ticks -- ) \ Ticks on 20.97 MHz
  $E000E014 ! \ How many ticks between interrupts ?
  7 $E000E010 ! \ Enable the systick interrupt.
;

: systick/16 ( ticks -- ) \ Ticks on 1.31 MHz
  $E000E014 ! \ How many ticks between interrupts ?
  3 $E000E010 ! \ Enable the systick interrupt, Coreclock/16.
;

```

```

: systick-1Hz ( -- ) 1310625 systick/16 ; \ Tick every second

: cornerstone ( Name ) ( -- )
  <builds begin here $3FF and while 0 , repeat
  does>   begin dup $3FF and while 4 + repeat
          eraseflashfrom
;

cornerstone Rewind-to-Basis

compiletoram
init

: tick ( -- ) ." Tick" cr ;

: clock ( -- )
  ['] tick irq-systick !
  systick-1Hz
  eint
;

```



```

: A0H ANI @ $1 OR ANI ! MBV2 ; \ here end the words that Set Bits HIGH - the next ones set the same bits LOW
: PWL PSWI @ $7 AND PSWI ! MBV2 ; \ x x x x 0 1 1 1 AND e.g. 1111 gives 0111
: T3L PSWI @ $B AND PSWI ! MBV2 ; \ 1 0 1 1 AND e.g. 1111 gives 1011
: T2L PSWI @ $D AND PSWI ! MBV2 ; \ 1 1 0 1 AND e.g. 1111 gives 1101
: T1L PSWI @ $E AND PSWI ! MBV2 ; \ 1 1 1 0 AND e.g. 1111 gives 11110
: O3L OUTP @ $7 AND OUTP ! MBV2 ;
: O2L OUTP @ $B AND OUTP ! MBV2 ;
: O1L OUTP @ $D AND OUTP ! MBV2 ;
: O0L OUTP @ $E AND OUTP ! MBV2 ;
: I3L IN @ $7 AND IN ! MBV2 ;
: I2L IN @ $B AND IN ! MBV2 ;
: I1L IN @ $D AND IN ! MBV2 ;
: I0L IN @ $E AND IN ! MBV2 ;
: A3L ANI @ $7 AND ANI ! MBV2 ;
: A2L ANI @ $B AND ANI ! MBV2 ;
: A1L ANI @ $D AND ANI ! MBV2 ;
: A0L ANI @ $E AND ANI ! MBV2 ;
\ Set I1 and/or I0 of the INPUTs, then call AND01, OR01, XOR01, INVERT0 and see the result of the logic result in O0
: AND01 IN @ DUP 1 RSHIFT AND 01 AND OUTP ! MBV2 ; \
: OR01 IN @ DUP 1 RSHIFT OR 01 AND OUTP ! MBV2 ; \
: XOR01 IN @ DUP 1 RSHIFT XOR 01 AND OUTP ! MBV2 ; \
: INVERT0 IN @ INVERT 01 AND OUTP ! MBV2 ; \

```

\ Forth Words used

```

\ 0 INCLUDE
\ 1 HEX
\ 2 \
\ 3 :
\ 4 ."
\ 5 CR
\ 6 ;
\ 8 Variable
\ 9 DUP
\ 10 $n
\ 11 AND
\ 12 IF
\ 13 ELSE
\ 14 THEN
\ 15 LSHIFT
\ 16 SPACE
\ 17 DROP
\ 18 DV
\ 19 @
\ 20 ?DO
\ 21 LOOP
\ 22 PAGE
\ 23 BEGIN
\ 24 1+
\ 25 !
\ 26 MS
\ 27 KEY?
\ 28 UNTIL
\ 29 EMIT
\ 30 .S
\ 31 .
\ 32 >R
\ 33 R>
\ 34 ( )

```

\ - this covers all of the Forth words used. Not programmed optimally - not the target - beginner's code for beginners

sos page cr cr ." Hello World - and wait a second" cr cr 1000 ms \ send SOS and Hello World

\ type SOS <cr>, COUNTER <cr> to start COUNTER, 200 SCOUNTER <cr> or start setting BITS using the control words with H/L

\ PWH PWL T3H T3L T2H T2L T1H T1L O3H O3L O2H O2L O1H O1L O0H O0L I3H I3L I2H I2L I1H I1L I0H I0L DEBUG ????

(end)