uMMT - The Main Code in short on One Page v7 2016_05_01

Easy for copy and paste - MPE VFX 430 LITE version, should work with 4e4th

```
\ Dictionary | C!   | CSET | CCLR | @   | LSHIFT | DUP    | DROP   | MS |
\            | AND  | OR   | XOR  |INVERT | RSHIFT | BEGIN | UNTIL | IF |
\       | :    | ;    | .S | .    |  U.    | KEY?    |  ? | HEX | DECIMAL |
\ HEX \ change to HEX where needed - do not forget ! Error else
\ 11 22 33    \ put some data on to the stack
\ .S          \ and check that it is there

\ 00 2E c! \ set port2 to just be GPIO, no internal specials used \ setgpio
\ 80 2A cset \ set bit7 DDR to output.   40/20/10 for other bits  \ bit7out
\ 80 29 cset \ set bit 7 HIGH. Replace by 40/20/10 for other bits \ bit7h
\ 80 29 cclr \ set bit 7 LOW.  Replace by 40/20/10 for other bits \ bit7l

\ variable OUTVA2     \ define a variable OUTV which holds some values
\ 77 OUTVA2 !         \ to test, load the variable with 77
\ Outva2 @ .          \ and display the data in OUTL - visible on output LEDs

\ : OUTL2 OUTVA2 @ 4 lshift  29 c! ; \ Output variable OUTV to OUTL \  LEDs

\ 0F 2A cclr \ set upper nibble to OUT, lower nibble to IN       \ setoutin
\ 0F 2F cset \ enable the 4 internal resistors, inputs not open    \ enres
\ 0F 29 cset \ ensure the 4 INPUTS are enabled                     \ set4in
: INIT2  00 2e c!  F0 2a C!  0F 2F cset 0F 29 cset ; \ as one word \ init
: 4SHIFTL2  4 LSHIFT  0F OR  ;  reposition IN to OUT              \ 4shiftl
: INTOOUT  INIT2  BEGIN  28 C@  4SHIFTL2  29 C!  KEY? UNTIL ;      \ intoout

\ 1000 ms 80 29 cset  \ 1 second delay to set bit 7 high (set hex/decimal?)
\ 500  ms 80 29 cclr  \ one half second delay after CR to set bit 7 to low

: DELAY begin dup ms 80 29 cset dup ms 80 29 cclr key? until drop; \x delay

DECIMAL \ set to decimal for easier calculations \                \ decimal
: DIT   80 29 cset  300 ms  80 29 cclr 100 ms  ; \ LEDon 300ms,off 100 \ dit
: DAA   80 29 cset  900 ms  80 29 cclr 100 ms  ; \ LEDon 900ms,off 100 \ daa
: SP    300 ms  ;  \ short pause between letters                       \ sp
: LP    999 ms  ;  \ long  pause between words                         \ lp
: MorseSOS begin dit dit dit sp daa daa daa sp dit dit dit sp key? until ;
: AND01b BEGIN 28 c@ DUP 1 rshift AND 01 AND 4shiftl 29 c! key? until ; \and01
: OR01b BEGIN 28 C@ DUP 1 RSHIFT OR 01 AND 4SHIFTL 29 C! KEY? UNTIL ;  \ or01
: XOR01b BEGIN 28 C@ DUP 1 RSHIFT XOR 01 AND 4SHIFTL 29 C! KEY? UNTIL ; \xor01
: INVERT0b  BEGIN  28 C@  INVERT  01 AND  4SHIFTL  29 C! KEY? UNTIL ;  \ inv0
: ToSc  Decimal begin 1 . 1000 ms key? until ; \ to screen 1 1 / sec   \ tosc

: LedCount

: INITADC  00 2E c!  F0 2A c!  0F 2F cset  0F 29 cset ; \ Init our simple ADC
Variable ADCV                                            \ Define variable ADCV
: ADC  initadc  0 ADCV !  BEGIN  1 ADCV +!  1 28 cget  UNTIL  ADCV @ . ; \ Go

Xxxx Variable ON    \ PWM OUT ON  - values: DAC xxx1, SERVO xxx2, SOUND xxx3
Yyyy Variable OFF   \ PWM OUT OFF - values: DAC yyy1, SERVO yyy2, SOUND yyy3
: DAC begin ON @ 0 do loop 80 29 cset OFF @ 0 do loop 80 29 cclr key? until ;

: FR begin dup @ 0 do loop 80 29 cset dup @ 0 do loop 80 29 cclr key? until ;
```

Our Small Debugger

```
: ??? 2E @ . 2A @ . 28 @ . OUTV @ . 29 @ . base @ >r hex .S r> base ! ; \ ???
  \   GPIO    DDR     INPUTS OUTV      LEDs            STACK

Other Words used: 0 commit - empty - freeled - flashled - application -
```

ExMark - Juergen Pintaske - www.exemark.com - 07736 70 76 74 v5_2016_03_26

```
\ From here onwards starts a file INCLUDE which contains new words
\ First type WORDS, see which FORTH words are available, difference to later
\ generate folder c:\ummt, save this page to end with ForthED as file INCLUDE
\ in uMMT type INCLUDE C:\ummt\inlcude
\ this will read the file compile and include the new words
\    and add them to WORDS that were there before
\ now type WORDS again, the new WORDS added via INCLUDE are the first ones
\ new Words are: SETGPIO  FREELED FLASHLED  OUTV     OUTL     SETOUTIN  INIT
\                BIT7OUT  BIT7L   BIT7H     BIT6OUT  BIT6L    BIT6H
\                BIT5OUT  BIT5H   BIT5L     BIT4OUT  BIT4L    BIT4H     INTOOUT
\                4SHIFTL  SET4IN  SETOUTIN  ENRES    DELAY    TOSC      ???
\                DIT      DAA     SP        LP       MORSESOS

: freeled  0 ledactive ! ;  \ decouple LED from Interrupt - free to be used
: flashled 1 ledactive ! ;  \ back to interrupt driven LED
: setgpio 00 2e c! ;     \ set port2 to using the General Purpose IO only
\                          - no internal specials used for now like counters …
\ IO control for OUTPUT; we use in uMMT PORT2. 4 bits OUTPUT, 4 bits INPUT
\ uMMT using Port 1 needs 2a to xx and 29 to yy - NOT BIT 11 AND 12 AS RX TX
\ Bit7-out3 Bit6-out2 Bit5-out1 Bit4-out0 Bit3-in3 Bit2-in2 Bit1-in1 Bit0-in0
: bit7out 80 2a cset ;   \ set bit 7 to be an output /  Texas Instruments
: bit7h   80 29 cset ;   \ set bit 7 to HIGH level   /  MSP430G2553 20 PIN DIL
: bit7l   80 29 cclr ;   \ set bit 7 to LOW  level   /  Plus 1 +      20 GND
: bit6out 40 2a cset ;   \ set bit 6 to be an output /  A0/x 2 10   26 19 O2
: bit6h   40 29 cset ;   \ set bit 6 to HIGH level   /    RX 3 11   27 18 O4
: bit6l   40 29 cclr ;   \ set bit 6 to LOW  level   /    TX 4 12      17 Tst
: bit5out 20 2a cset ;   \ set bit 5 to be an output /    S2 5 13      16 RESS1
: bit5h   20 29 cset ;   \ set bit 5 to HIGH level   /   AD4 6 14   17 15 S3
: bit5l   20 29 cclr ;   \ set bit 5 to LOW  level   /   FRQ 7 15   16 14 PWM
: bit4out 10 2a cset ;   \ set bit 4 to be an output /  IN0  8 20   25 13 O1
: bit4h   10 29 cset ;   \ set bit 4 to HIGH level   /  IN1  9 21   24 12 O0
: bit4l   10 29 cclr ;   \ set bit 4 to LOW  level   /  IN2 10 22   23 11 IN3
variable outv            \ define a variable outv  - OUTPUT Variable
: outl  outv2 @ 4 lshift 29 c! ; \ define outl to output outv to LEDs
: setoutin  0f 2a cclr ;  \ set the Data Direction Register to 4 OUT 4 IN
: enres     0f 2f cset ;  \ now enable the 4 internal resistors on IN lines
: set4in    0f 29 cset ;  \ confirm that the bits 0 to 3 are set to IN
: init3 00 2e c!  f0 2a c!  0f 2f cset  0f 29 cset ;  \ the same as word INIT
: 4shiftl3  4 lshift  0f or ;  \ shift 4 IN bits 4x left to position OUT bits
: intoout  init begin 28 c@ 4shiftl 29 c! key? until ;  \ show IN on OUT bits
: delay begin dup ms 80 29 cset dup ms 80 29 cclr key? until drop ; \ x delay
\ 100 delay  ok
: dit 80 29 cset 300 ms 80 29 cclr 100 ms ;  \ morse code short, 100ms
: daa 80 29 cset 900 ms 80 29 cclr 100 ms ;  \ morse code long,  300ms
: sp 300 ms ;                                \ short pause,      100ms
: lp 1100 ms ;                               \ long pause,      1100ms
: MorseSOS  begin dit dit dit sp daa daa daa sp dit dit dit lp key? until ;
: ToSC  decimal begin 1 . 1000 ms key? until ; \ send 1 to display, 1 per sec
 \ tosc 1 1 1 1 1 1 1 1 1 1 1 1  ok
\ MiniDebugger
: ???  hex 2e @ . 2a @ .  28 @ . outv @ . 29 @ . base @ >r hex .s r> base ! ;
\          0      F0      FF     77       FF                   <1> -3E22 ok -
\          GPIO   DDR     INPUTS OUTv     OUTl                 STACK
\ displays the registers GPIO  DDR  INPUTs variable OUTV LEDs and Stack

\ ExMark - Juergen Pintaske – www.exemark.com - 07736 70 76 74  v5_2016_03_26
```

Additional Examples and Descriptions

```
: count3  0 begin  dup 4shiftl 29 c! 1000 ms 1+ key? until ;
: contf3  0 begin  dup 4shiftl 29 c!  100 ms 1+ key? until ;
```

Values for sound and RC Servos   - to be done

Description AD Converter   - to be done

Additional Debugger for Port1 and Port 3   - to be done

Links:

MPE Forth VFX LITE – The Software running on the chip
http://www.mpeforth.com/xc7lite.htm

Most of the project data
https://www.forth-ev.de/wiki/doku.php/en:projects:microbox:start

One of the sponsors even has stored more, including the video
http://lb-eccom-865556685.eu-west-1.elb.amazonaws.com/blog/posts/page-3

Starting Forth on the Forth Inc Website
https://www.forth.com/starting-forth/0-starting-forth/

Starting Forth for print on Exmark website item number 3
http://www.exemark.com/FORTH.htm

Some eBooks about Forth
http://goo.gl/o3hH2g

4e4th Wiki
https://www.forth-ev.de/wiki/doku.php/en:projects:4e4th:start
from there how to save the complete contents of your chip including application:
Unfortiunately for this you need a friend who owns a TI Launchpad

Saving your Program

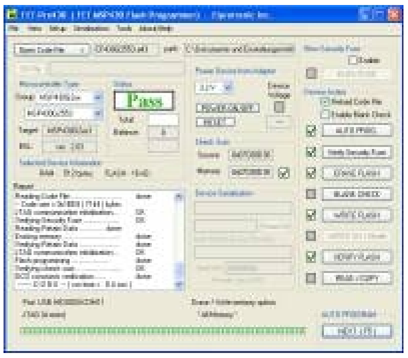
The only thing you have to do to save your program is: type SAVE <Enter>. SAVE saves your program on the MSP430G2553 FLASH memory.
It is possible to upload and save the MCUs image using a Programmer.
To remove your program, type WIPE. Be careful. This is a powerful instruction. When typing WIPE, your work of may be several hours will be gone. That's why 4E4th-Terminal logs all your typing, and you will be able to download your work again.
Another way to remove your programs is by pressing the LaunchPad's S2 button and then the RESET button while holding S2.
There is a special characteristic in handling the FLASH in MSP430 MCUs, and FLASH memory in general: The FLASH always can be correctly written to only once! It is thus not as comfortable as RAM or FRAM. Practically seen, that means, if you forget to erase the FLASH properly before you start programming (compiling) again, the program crashes. Or, with other words: If there is no $FFFF at IHERE nothing can be written there. Use WIPE to compile into the fresh FLASH.

Programmer

You find the 4E4th binary image at the Forth-Gesellschaft Repository: http://www.forth-ev.de/repos/, see section Bootstrap Loading for a link. It has to be loaded into the MCU using a suitable programmer.

Windows XP and Windows 7

FET-Pro430 Lite version FREE. http://www.elprotronic.com/fetpro430.html
Set the 4E4th image (*open code file*) and path, choose the MSP430G2553 MCU (*Microcontroller Type*). Activate the box *reload code file*. Choose the right connection at menu *setup connections*. Click on *AUTOPROG* - programming starts.
Here a screen shot:



Linux

At Linux the TI LaunchPad should come up with /dev/ttyACM0. This entry in the /dev directory should appear, irrespective of whether your Launchpad is able to communicate or contains a useful program.
Under Linux you can use the program mspdebug to flash and debug the LaunchPad MCU. With some luck, your distribution knows the package and you can use a command such as

```
 $ apt-get install mspdebug
```

to install it.
Version 0.19 or higher is preferred, http://mspdebug.sourceforge.net/ has the latest sources if it comes to this.
Loading the hex file image onto the LaunchPad MCU can be done script-style:

```
 $ mspdebug rf2500 "prog 4e4th.a43"
```

Alternatively you can use mspdebug interactively:

```
    $ mspdebug rf2500
```

Safest is the combination of

```
    (mspdebug) erase all
    (mspdebug) prog 4e4th.a43
```

Then – still in mspdebug – you can check the content of the flash memory.

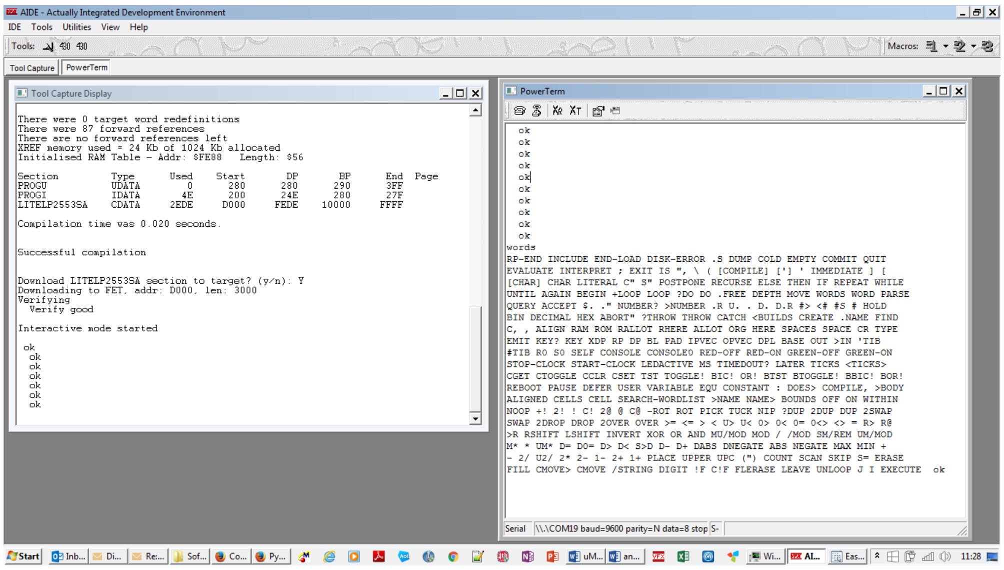
Potential problems:

1. Some modern Linux Systems Background Programs (one example is the ModemManager) immediately occupy the Interface installed for Launchpad and try to talk to LaunchPad as to a modem (See Linux today: too much plug and play; http://fedetft.wordpress.com/2011/01/07/linux-today-too-much-plug-and-play/). (cas)
2. In some distributions user might not have access rights to /dev/ttyACM0 so mspdebug should be run via sudo as shown below.
3. If you have an older version (0.13) of mspdebug the model name rf2500 will not be recognized and the command will be rejected. You can force the loading with:

```
 $ sudo mspdebug --fet-force-id "prog 4e4th.a43" rf2500
```

Mac OSX (intel)

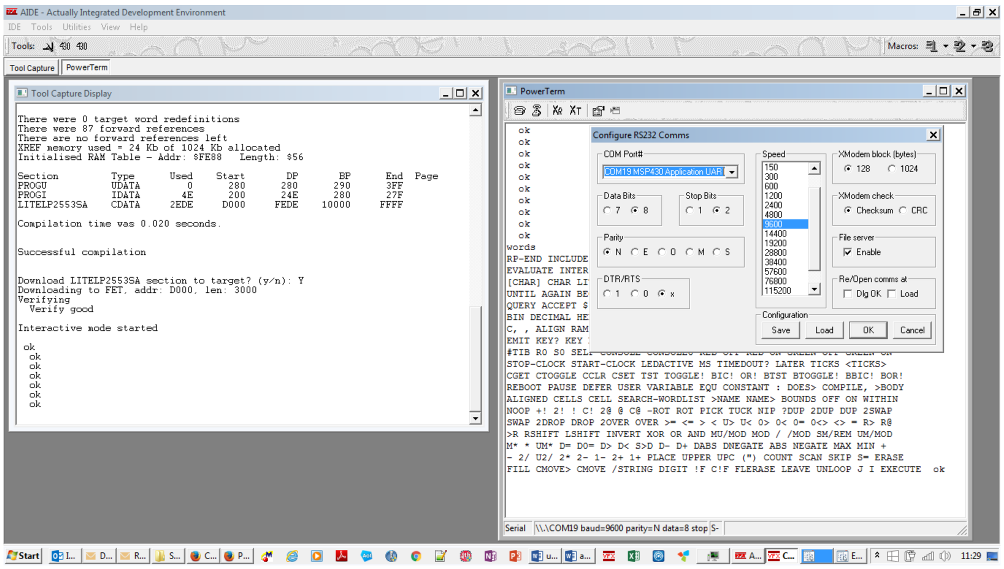
There is a group which is collecting tools: http://processors.wiki.ti.com/index.php/MSP430_LaunchPad_Mac_OS_X


AIDE - Actually Integrated Development Environment
IDE Tools Utilities View Help
Tools:
Macros:
Tool Capture
PowerTerm
Tool Capture Display
PowerTerm

```
There were 0 target word redefinitions
There were 87 forward references
There are no forward references left
XREF memory used = 24 Kb of 1024 Kb allocated
Initialised RAM Table - Addr: $FE88   Length: $56

Section         Type      Used    Start      DP        BP      End  Page
PROGU           UDATA        0      280     280       290      3FF
PROGI           IDATA       4E      200     24E       280      27F
LITELP2553SA    CDATA     2EDE     D000    FEDE     10000     FFFF

Compilation time was 0.020 seconds.


Successful compilation


Download LITELP2553SA section to target? (y/n): Y
Downloading to FET, addr: D000, len: 3000
Verifying
  Verify good

Interactive mode started

 ok
  ok
  ok
  ok
  ok
  ok
  ok
```

```
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
words
RP-END INCLUDE END-LOAD DISK-ERROR .S DUMP COLD EMPTY COMMIT QUIT
EVALUATE INTERPRET ; EXIT IS ", \ ( [COMPILE] ['] ' IMMEDIATE ] [
[CHAR] CHAR LITERAL C" S" POSTPONE RECURSE ELSE THEN IF REPEAT WHILE
UNTIL AGAIN BEGIN +LOOP LOOP ?DO DO .FREE DEPTH MOVE WORDS WORD PARSE
QUERY ACCEPT $. ." NUMBER? >NUMBER .R U. . D. D.R #> <# #S # HOLD
BIN DECIMAL HEX ABORT" ?THROW THROW CATCH <BUILDS CREATE .NAME FIND
C, , ALIGN RAM ROM RALLOT RHERE ALLOT ORG HERE SPACES SPACE CR TYPE
EMIT KEY? KEY XDP RP DP BL PAD IPVEC OPVEC DPL BASE OUT >IN 'TIB
#TIB R0 S0 SELF CONSOLE CONSOLE0 RED-OFF RED-ON GREEN-OFF GREEN-ON
STOP-CLOCK START-CLOCK LEDACTIVE MS TIMEDOUT? LATER TICKS <TICKS>
CGET CTOGGLE CCLR CSET TST TOGGLE! BIC! OR! BTST BTOGGLE! BBIC! BOR!
REBOOT PAUSE DEFER USER VARIABLE EQU CONSTANT : DOES> COMPILE, >BODY
ALIGNED CELLS CELL SEARCH-WORDLIST >NAME NAME> BOUNDS OFF ON WITHIN
NOOP +! 2! ! C! 2@ @ C@ -ROT ROT PICK TUCK NIP ?DUP 2DUP DUP 2SWAP
SWAP 2DROP DROP 2OVER OVER >= <= > < U> U< 0> 0< 0= 0<> <> = R> R@
>R RSHIFT LSHIFT INVERT XOR OR AND MU/MOD MOD / /MOD SM/REM UM/MOD
M* * UM* D= D0= D> D< S>D D- D+ DABS DNEGATE ABS NEGATE MAX MIN +
- 2/ U2/ 2* 2- 1- 2+ 1+ PLACE UPPER UPC (") COUNT SCAN SKIP S= ERASE
FILL CMOVE> CMOVE /STRING DIGIT !F C!F FLERASE LEAVE UNLOOP J I EXECUTE  ok
```

Serial | \\.\COM19 baud=9600 parity=N data=8 stop | S-



Configure RS232 Comms

COM Port#: COM19 MSP430 Application UART

Data Bits: 7 [✓ 8]

Stop Bits: 1 [✓ 2]

Parity: [✓ N] E O M S

DTR/RTS: 1 0 [✓ x]

Speed: 150 300 600 1200 2400 4800 [✓ 9600] 14400 19200 28800 38400 57600 76800 115200

XModem block (bytes): [✓ 128] 1024

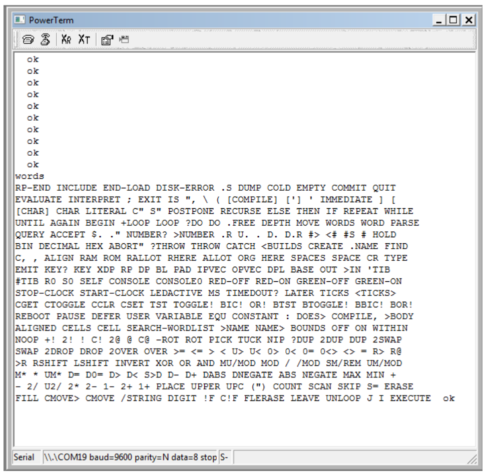
XModem check: [✓ Checksum] CRC

File server: [✓ Enable]

Re/Open comms at: Dlg OK ☐ Load ☐

```
There were 0 target word redefinitions
There were 87 forward references
There are no forward references left
XREF memory used = 24 Kb of 1024 Kb allocated
Initialised RAM Table - Addr: $FE88   Length: $56

Section         Type      Used    Start      DP        BP      End  Page
PROGU           UDATA        0      280     280       290      3FF
PROGI           IDATA       4E      200     24E       280      27F
LITELP2553SA    CDATA     2EDE     D000    FEDE     10000     FFFF

Compilation time was 0.020 seconds.


Successful compilation


Download LITELP2553SA section to target? (y/n): Y
Downloading to FET, addr: D000, len: 3000
Verifying
  Verify good

Interactive mode started

 ok
  ok
  ok
  ok
  ok
  ok
  ok
```

Serial | \\.\COM19 baud=9600 parity=N data=8 stop | S-



```
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
  ok
words
RP-END INCLUDE END-LOAD DISK-ERROR .S DUMP COLD EMPTY COMMIT QUIT
EVALUATE INTERPRET ; EXIT IS ", \ ( [COMPILE] ['] ' IMMEDIATE ] [
[CHAR] CHAR LITERAL C" S" POSTPONE RECURSE ELSE THEN IF REPEAT WHILE
UNTIL AGAIN BEGIN +LOOP LOOP ?DO DO .FREE DEPTH MOVE WORDS WORD PARSE
QUERY ACCEPT $. ." NUMBER? >NUMBER .R U. . D. D.R #> <# #S # HOLD
BIN DECIMAL HEX ABORT" ?THROW THROW CATCH <BUILDS CREATE .NAME FIND
C, , ALIGN RAM ROM RALLOT RHERE ALLOT ORG HERE SPACES SPACE CR TYPE
EMIT KEY? KEY XDP RP DP BL PAD IPVEC OPVEC DPL BASE OUT >IN 'TIB
#TIB R0 S0 SELF CONSOLE CONSOLE0 RED-OFF RED-ON GREEN-OFF GREEN-ON
STOP-CLOCK START-CLOCK LEDACTIVE MS TIMEDOUT? LATER TICKS <TICKS>
CGET CTOGGLE CCLR CSET TST TOGGLE! BIC! OR! BTST BTOGGLE! BBIC! BOR!
REBOOT PAUSE DEFER USER VARIABLE EQU CONSTANT : DOES> COMPILE, >BODY
ALIGNED CELLS CELL SEARCH-WORDLIST >NAME NAME> BOUNDS OFF ON WITHIN
NOOP +! 2! ! C! 2@ @ C@ -ROT ROT PICK TUCK NIP ?DUP 2DUP DUP 2SWAP
SWAP 2DROP DROP 2OVER OVER >= <= > < U> U< 0> 0< 0= 0<> <> = R> R@
>R RSHIFT LSHIFT INVERT XOR OR AND MU/MOD MOD / /MOD SM/REM UM/MOD
M* * UM* D= D0= D> D< S>D D- D+ DABS DNEGATE ABS NEGATE MAX MIN +
- 2/ U2/ 2* 2- 1- 2+ 1+ PLACE UPPER UPC (") COUNT SCAN SKIP S= ERASE
FILL CMOVE> CMOVE /STRING DIGIT !F C!F FLERASE LEAVE UNLOOP J I EXECUTE  ok
```

Serial | \\.\COM19 baud=9600 parity=N data=8 stop | S-

```
  ok
words
RP-END INCLUDE END-LOAD DISK-ERROR .S DUMP COLD EMPTY COMMIT QUIT
EVALUATE INTERPRET ; EXIT IS ", \ ( [COMPILE] ['] ' IMMEDIATE ] [
[CHAR] CHAR LITERAL C" S" POSTPONE RECURSE ELSE THEN IF REPEAT WHILE
UNTIL AGAIN BEGIN +LOOP LOOP ?DO DO .FREE DEPTH MOVE WORDS WORD PARSE
QUERY ACCEPT $. ." NUMBER? >NUMBER .R U. . D. D.R #> <# #S # HOLD
BIN DECIMAL HEX ABORT" ?THROW THROW CATCH <BUILDS CREATE .NAME FIND
C, , ALIGN RAM ROM RALLOT RHERE ALLOT ORG HERE SPACES SPACE CR TYPE
EMIT KEY? KEY XDP RP DP BL PAD IPVEC OPVEC DPL BASE OUT >IN 'TIB
#TIB R0 S0 SELF CONSOLE CONSOLE0 RED-OFF RED-ON GREEN-OFF GREEN-ON
```

```
STOP-CLOCK START-CLOCK LEDACTIVE MS TIMEDOUT? LATER TICKS <TICKS>
CGET CTOGGLE CCLR CSET TST TOGGLE! BIC! OR! BTST BTOGGLE! BBIC! BOR!
REBOOT PAUSE DEFER USER VARIABLE EQU CONSTANT : DOES> COMPILE, >BODY
ALIGNED CELLS CELL SEARCH-WORDLIST >NAME NAME> BOUNDS OFF ON WITHIN
NOOP +! 2! ! C! 2@ @ C@ -ROT ROT PICK TUCK NIP ?DUP 2DUP DUP 2SWAP
SWAP 2DROP DROP 2OVER OVER >= <= > < U> U< 0> 0< 0= 0<> <> = R> R@
>R RSHIFT LSHIFT INVERT XOR OR AND MU/MOD MOD / /MOD SM/REM UM/MOD
M* * UM* D= D0= D> D< S>D D- D+ DABS DNEGATE ABS NEGATE MAX MIN +
- 2/ U2/ 2* 2- 1- 2+ 1+ PLACE UPPER UPC (") COUNT SCAN SKIP S= ERASE
FILL CMOVE> CMOVE /STRING DIGIT !F C!F FLERASE LEAVE UNLOOP J I EXECUTE  ok
```

Just a few of those words are used here:

```
COMMIT ; \ THEN IF UNTIL AGAIN BEGIN WORDS . BIN DECIMAL HEX SPACE CR
KEY? BASE RED-OFF RED-ON GREEN-OFF GREEN-ON LEDACTIVE MS CCLR CSET
VARIABLE : C! @ DUP SWAP DROP RSHIFT INVERT XOR OR AND + ok
```

In sequence as they are used in the examples:

```
HEX
.S
LEDACTIVE
C!
CSET
CCLR
VARIABLE
@
.
:
;
LSHIFT
OR
BEGIN
UNTIL
KEY?
MS
DUP
RSHIFT
AND
OR
XOR
INVERT
```