A Windows Guide to loading Tachyon Forth into Propeller P1

Introduction

Tachyon Forth is a powerful Forth programming tool set for the Parallax Propeller P1, created by Peter Jakacki. This is a guide by forth enthusiast Bob Edwards about getting Tachyon loaded into the P1 EEPROM, using Windows 10 tools. The method enables the user to alter the kernel if he wishes or choose what modules are loaded to suit the job in hand and to save precious memory.

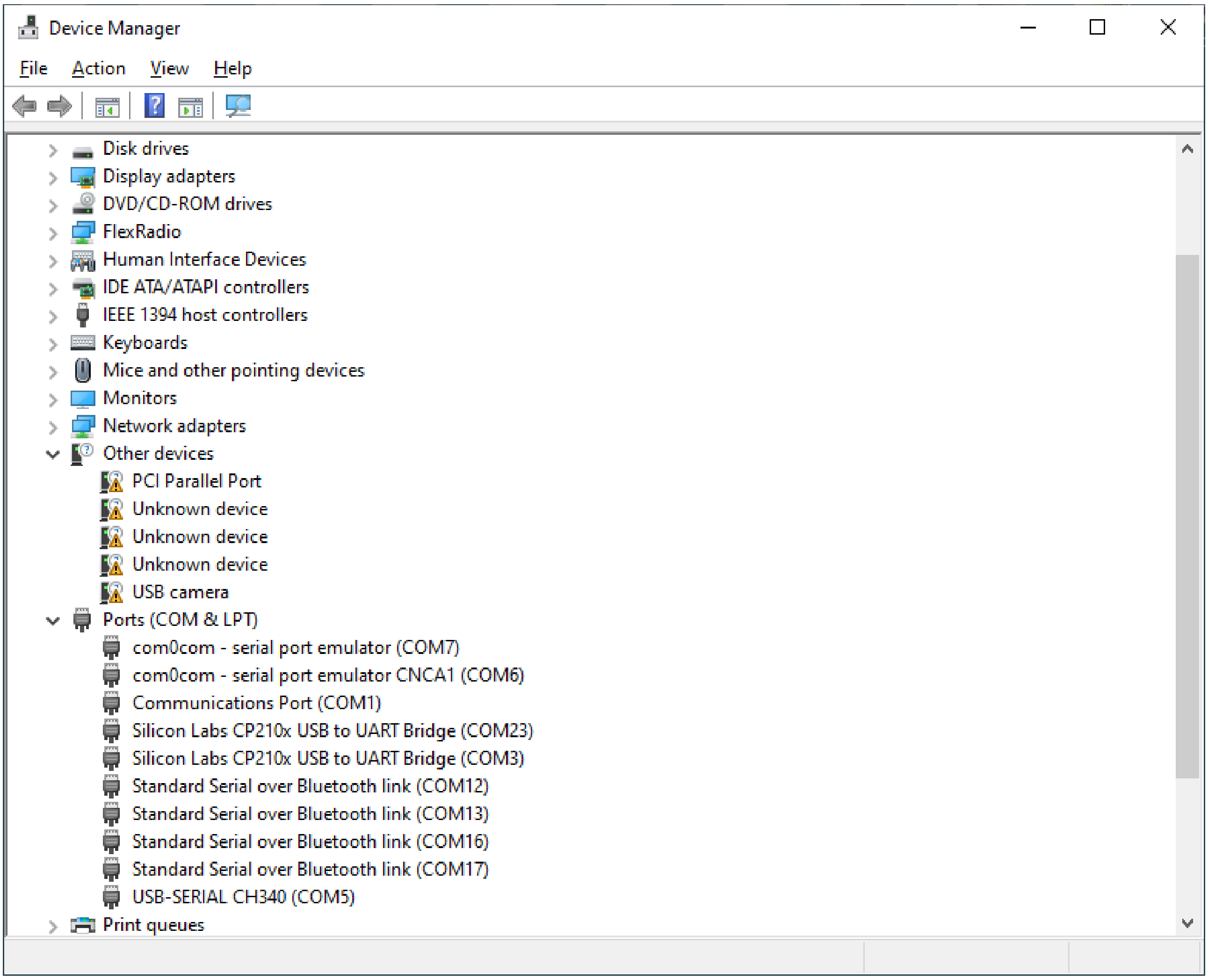
The Compiler

Tachyon is compiled and loaded using the BST tool currently available here www.fnarfbargle.com/bst.html . Download the latest code for bst, bstc and bstl. Unzip all three and place the programs in a common directory (there is no other installation required) I made a shortcut to bst on my desktop.

Connecting to the Windows PC

On his Sourceforge web page, Peter Jakacki describes how to compile and flash Tachyon onto the P1 using Linux tools. Something different is needed for Windows users.

1. Open Device Manager on the PC and check the Ports (COM & LPT) available
2. Connect the P1 up to a USB port of your PC direct (if onboard USB is present) else via a Parallax Prop Plug. The USB link has to be capable of resetting the P1 processor – the Prop Plug has this reset circuit built-in, so ftdi modules off ebay won't do, unless modified. Make sure your board is powered up. Be careful that any power provided by the USB link does not conflict with any external power supply to the P1 board.
3. In Device Manager, an extra port will have been created – in my case it's USB-SERIAL CH340 (COM5). Remember the com port number.



- Disk drives
- Display adapters
- DVD/CD-ROM drives
- FlexRadio
- Human Interface Devices
- IDE ATA/ATAPI controllers
- IEEE 1394 host controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
  - PCI Parallel Port
  - Unknown device
  - Unknown device
  - Unknown device
  - USB camera
- Ports (COM & LPT)
  - com0com - serial port emulator (COM7)
  - com0com - serial port emulator CNCA1 (COM6)
  - Communications Port (COM1)
  - Silicon Labs CP210x USB to UART Bridge (COM23)
  - Silicon Labs CP210x USB to UART Bridge (COM3)
  - Standard Serial over Bluetooth link (COM12)
  - Standard Serial over Bluetooth link (COM13)
  - Standard Serial over Bluetooth link (COM16)
  - Standard Serial over Bluetooth link (COM17)
  - USB-SERIAL CH340 (COM5)
- Print queues

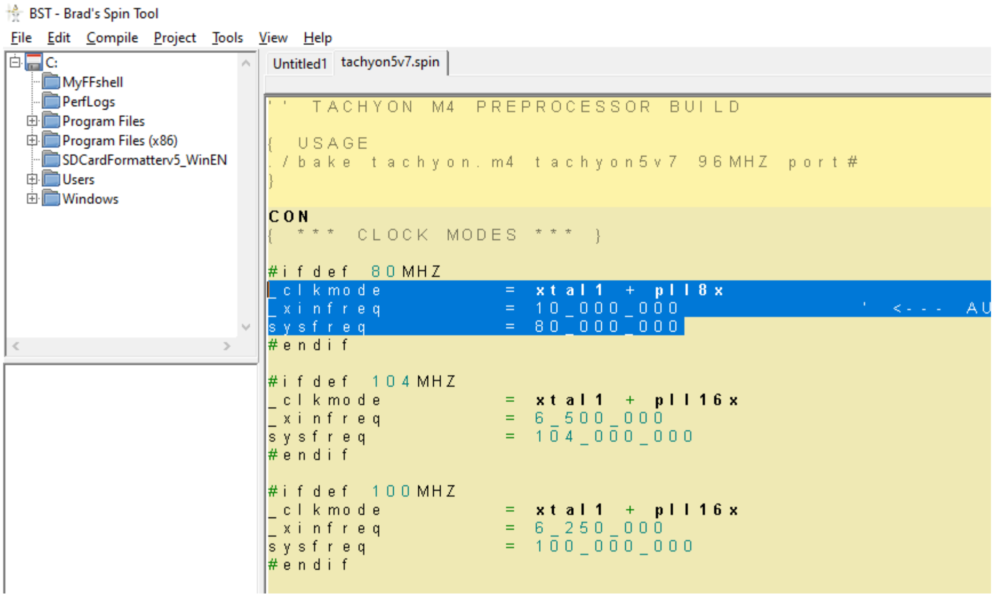
Tachyon Source Code

The Tachyon source code is available at [Tachyon Forth - Browse /Tachyon at SourceForge.net]().

1. From the kernel directory, download tachyon5v7.spin – the source code of the forth kernel.
2. From the Forth directory, download Extend.fth (Extensions to the kernel) and optionally, Easyfile.fth (An SD-card file system).

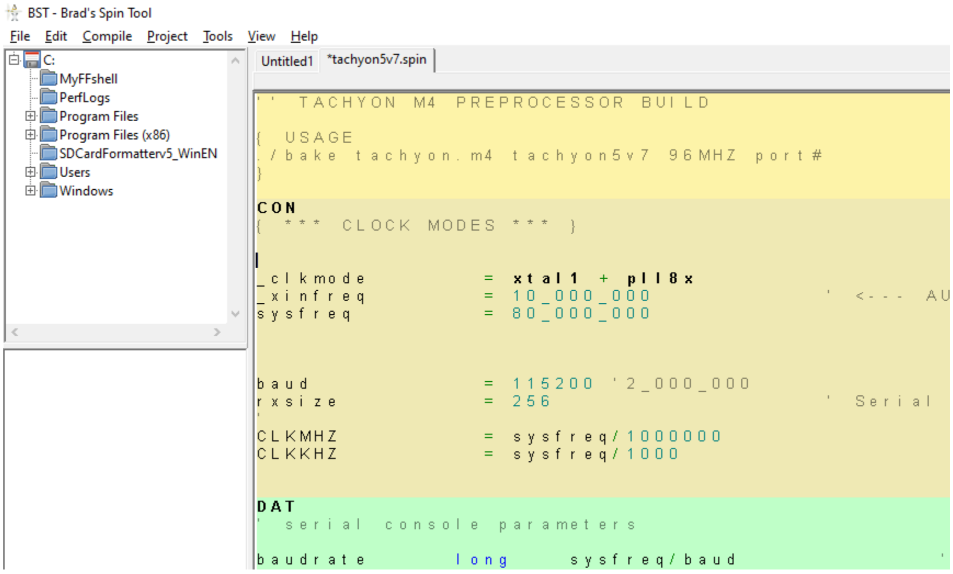
The first step is some minor alterations to the source code to define the clock rates on your P1 circuit board:-

1. Open BST and load tachyon5v7.spin



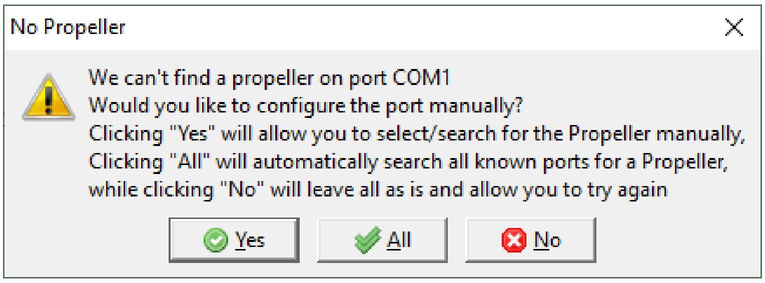
```
'' TACHYON M4 PREPROCESSOR BUILD

{ USAGE
./bake tachyon.m4 tachyon5v7 96MHZ port#
}

CON
{ *** CLOCK MODES *** }

#ifdef 80MHZ
_clkmode        = xtal1 + pll8x
_xinfreq        = 10_000_000                ' <--- AU
sysfreq         = 80_000_000
#endif

#ifdef 104MHZ
_clkmode        = xtal1 + pll16x
_xinfreq        = 6_500_000
sysfreq         = 104_000_000
#endif

#ifdef 100MHZ
_clkmode        = xtal1 + pll16x
_xinfreq        = 6_250_000
sysfreq         = 100_000_000
#endif
```

At the start of the code, there are multiple #ifdef - #endif statements that set values into _clkmode, _xinfreq and sysfreq. Delete all these, apart from the ‘80MHz’ code highlighted. You should be left with:-



```
'' TACHYON M4 PREPROCESSOR BUILD

{ USAGE
./bake tachyon.m4 tachyon5v7 96MHZ port#
}

CON
{ *** CLOCK MODES *** }

_clkmode        = xtal1 + pll8x
_xinfreq        = 10_000_000                ' <--- AU
sysfreq         = 80_000_000


baud            = 115200 '2_000_000
rxsize          = 256                       ' Serial
'
CLKMHZ          = sysfreq/1000000
CLKKHZ          = sysfreq/1000

DAT
' serial console parameters

baudrate        long    sysfreq/baud        '
```

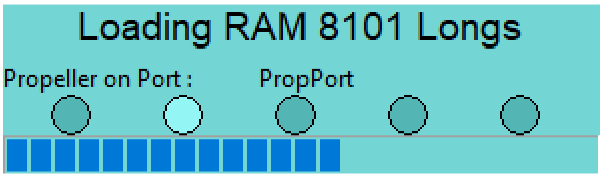
2. If your board has a 5 or 10 MHz crystal source for the P1, leave the xinfreq value as 10_000_000. Tachyon will autoadjust on reset. If you have another crystal frequency, then altering the _xinfreq value will be needed.
3. Save the source code.
4. From the menu bar, select Compile → Compile and Load EEPROM. You may get this popup:-



No Propeller

We can't find a propeller on port COM1
Would you like to configure the port manually?
Clicking "Yes" will allow you to select/search for the Propeller manually,
Clicking "All" will automatically search all known ports for a Propeller,
while clicking "No" will leave all as is and allow you to try again

5. Press All, the P1 should be autodetected, and the code should be loaded into the EEPROM – progress is shown by this pop-up:-



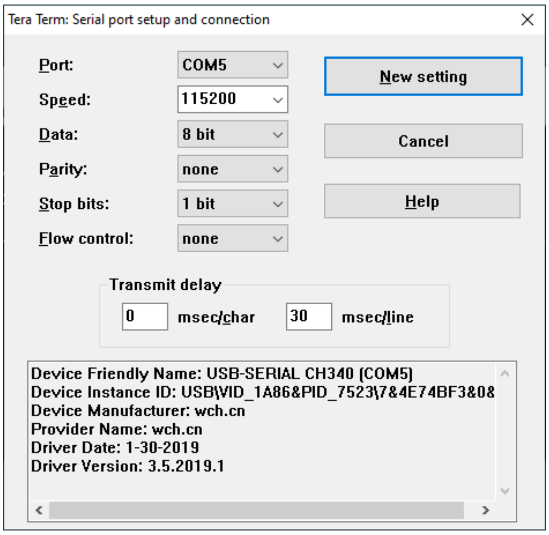
Loading RAM 8101 Longs

Propeller on Port : PropPort

6. After this popup disappears, we’re done using the compiler, so close it.

Setting up the terminal link

1. Download and install Tera Term from [Tera Term Open Source Project (osdn.jp)]
2. From the menu bar select Setup → Serial port. Set the Port to the com port that you found using Device Manager. Set Speed to 115200, data to 8 bit, Parity to none, Stop bits 1 bit, Flow Control none and Transmit delay 30 msecs/line. (The latter setting is conservative)



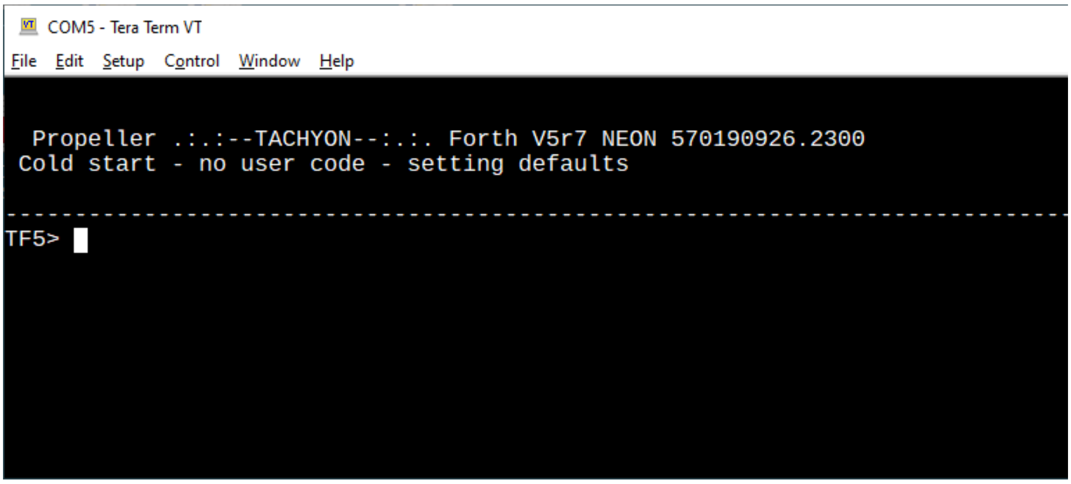
Tera Term: Serial port setup and connection

Port: COM5
Speed: 115200
Data: 8 bit
Parity: none
Stop bits: 1 bit
Flow control: none

Transmit delay
0 msec/char 30 msec/line

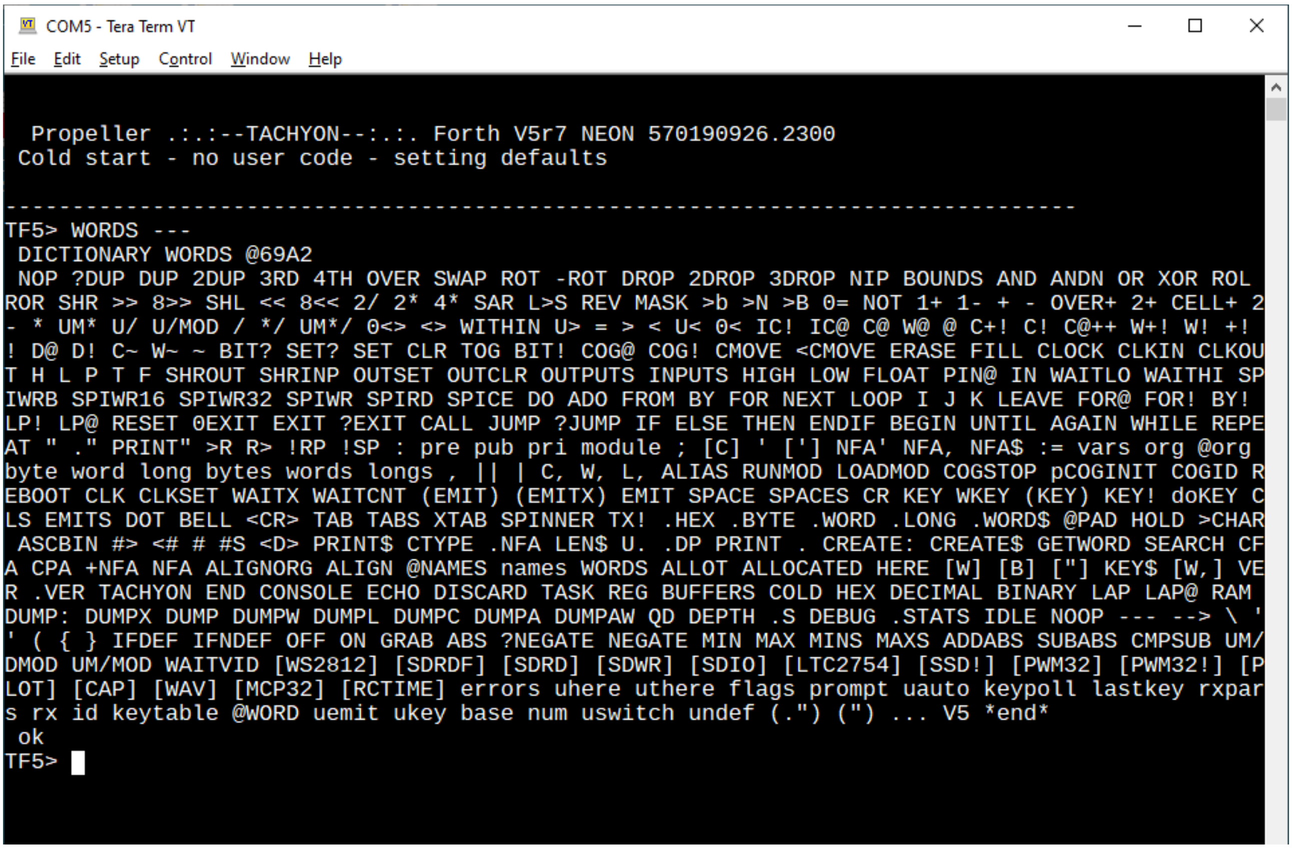
```
Device Friendly Name: USB-SERIAL CH340 (COM5)
Device Instance ID: USB\VID_1A86&PID_7523\7&4E74BF3&0&
Device Manufacturer: wch.cn
Provider Name: wch.cn
Driver Date: 1-30-2019
Driver Version: 3.5.2019.1
```

3. Press ‘New Setting’ to reset the port to the new values
4. Reset the P1 and you should see on the terminal:-



```
  Propeller .:.:--TACHYON--:.:. Forth V5r7 NEON 570190926.2300
 Cold start - no user code - setting defaults

-----------------------------------------------------------------------------
TF5> 
```

5. To check two way comms, type WORDS (must be capitals) and you should get:-

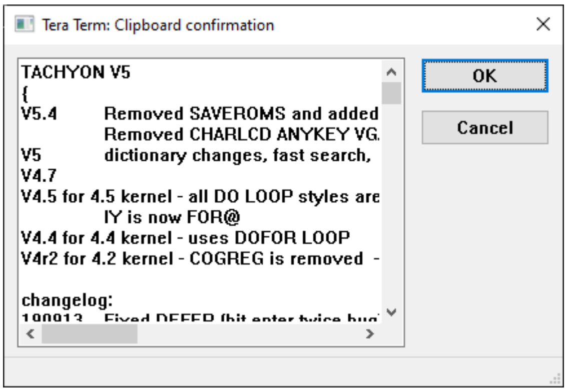


```
  Propeller .:.:--TACHYON--:.:. Forth V5r7 NEON 570190926.2300
 Cold start - no user code - setting defaults

-----------------------------------------------------------------------------
TF5> WORDS ---
 DICTIONARY WORDS @69A2
 NOP ?DUP DUP 2DUP 3RD 4TH OVER SWAP ROT -ROT DROP 2DROP 3DROP NIP BOUNDS AND ANDN OR XOR ROL
ROR SHR >> 8>> SHL << 8<< 2/ 2* 4* SAR L>S REV MASK >b >N >B 0= NOT 1+ 1- + - OVER+ 2+ CELL+ 2
- * UM* U/ U/MOD / */ UM*/ 0<> <> WITHIN U> = > < U< 0< IC! IC@ C@ W@ @ C+! C! C@++ W+! W! +!
! D@ D! C~ W~ ~ BIT? SET? SET CLR TOG BIT! COG@ COG! CMOVE <CMOVE ERASE FILL CLOCK CLKIN CLKOU
T H L P T F SHROUT SHRINP OUTSET OUTCLR OUTPUTS INPUTS HIGH LOW FLOAT PIN@ IN WAITLO WAITHI SP
IWRB SPIWR16 SPIWR32 SPIWR SPIRD SPICE DO ADO FROM BY FOR NEXT LOOP I J K LEAVE FOR@ FOR! BY!
LP! LP@ RESET 0EXIT EXIT ?EXIT CALL JUMP ?JUMP IF ELSE THEN ENDIF BEGIN UNTIL AGAIN WHILE REPE
AT " ." PRINT" >R R> !RP !SP : pre pub pri module ; [C] ' ['] NFA' NFA, NFA$ := vars org @org
byte word long bytes words longs , || | C, W, L, ALIAS RUNMOD LOADMOD COGSTOP pCOGINIT COGID R
EBOOT CLK CLKSET WAITX WAITCNT (EMIT) (EMITX) EMIT SPACE SPACES CR KEY WKEY (KEY) KEY! doKEY C
LS EMITS DOT BELL <CR> TAB TABS XTAB SPINNER TX! .HEX .BYTE .WORD .LONG .WORD$ @PAD HOLD >CHAR
 ASCBIN #> <# # #S <D> PRINT$ CTYPE .NFA LEN$ U. .DP PRINT . CREATE: CREATE$ GETWORD SEARCH CF
A CPA +NFA NFA ALIGNORG ALIGN @NAMES names WORDS ALLOT ALLOCATED HERE [W] [B] ["] KEY$ [W,] VE
R .VER TACHYON END CONSOLE ECHO DISCARD TASK REG BUFFERS COLD HEX DECIMAL BINARY LAP LAP@ RAM
DUMP: DUMPX DUMP DUMPW DUMPL DUMPC DUMPA DUMPAW QD DEPTH .S DEBUG .STATS IDLE NOOP --- --> \ '
' ( { } IFDEF IFNDEF OFF ON GRAB ABS ?NEGATE NEGATE MIN MAX MINS MAXS ADDABS SUBABS CMPSUB UM/
DMOD UM/MOD WAITVID [WS2812] [SDRDF] [SDRD] [SDWR] [SDIO] [LTC2754] [SSD!] [PWM32] [PWM32!] [P
LOT] [CAP] [WAV] [MCP32] [RCTIME] errors uhere uthere flags prompt uauto keypoll lastkey rxpar
s rx id keytable @WORD uemit ukey base num uswitch undef (.") (") ... V5 *end*
 ok
TF5> 
```

6. You’re in business with a very basic Tachyon system

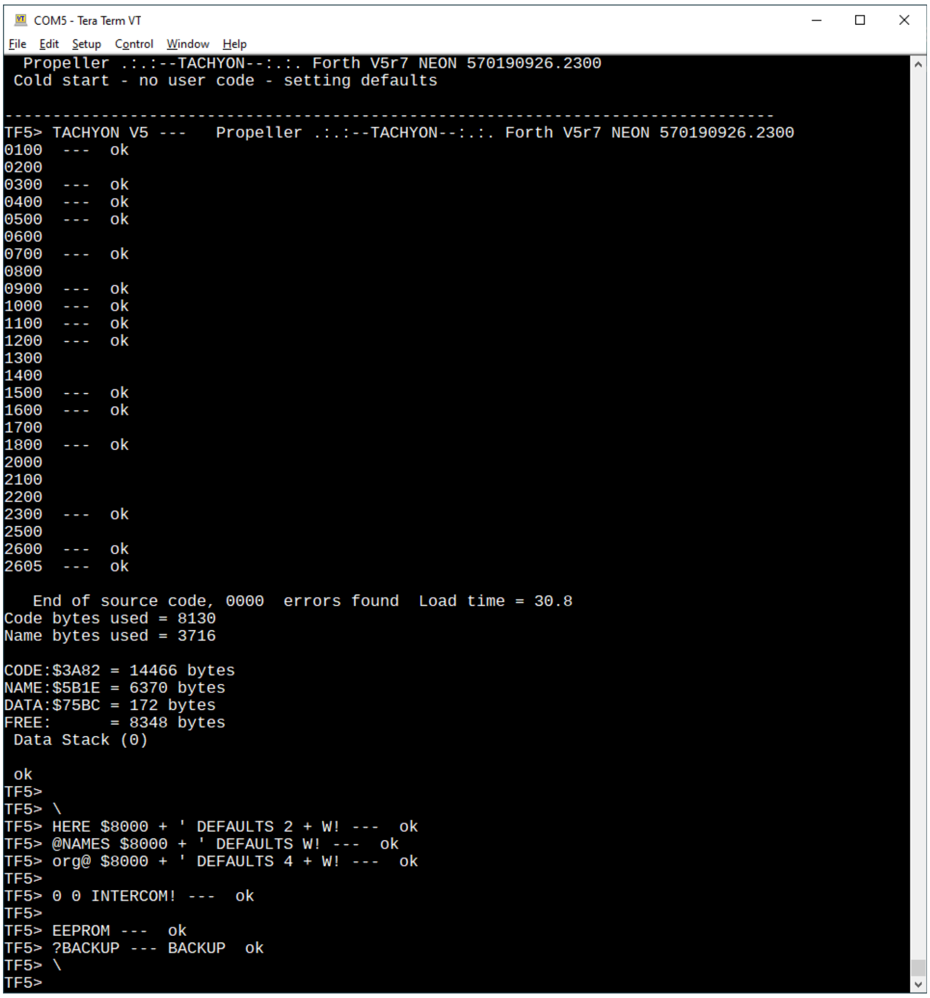
Extending the system

Download Notepad++ from [Notepad++ (notepad-plus-plus.org)](notepad-plus-plus.org) - it's a good multi-tab, syntax highlighting editor for wrangling forth source code. Load Extend.fth into it. You may like to make your own Extend.fth by cutting out sections that you're not interested in, but for now select all of the text with <ctrl> A, copy it to the clipboard with <ctrl> C and right click on Tera Term:-



```
TACHYON V5
{
V5.4        Removed SAVEROMS and added
            Removed CHARLCD ANYKEY VG.
V5          dictionary changes, fast search,
V4.7
V4.5 for 4.5 kernel - all DO LOOP styles are
            IY is now FOR@
V4.4 for 4.4 kernel - uses DOFOR LOOP
V4r2 for 4.2 kernel - COGREG is removed  -

changelog:
190913    Fixed DEFER (hit enter twice bug
```

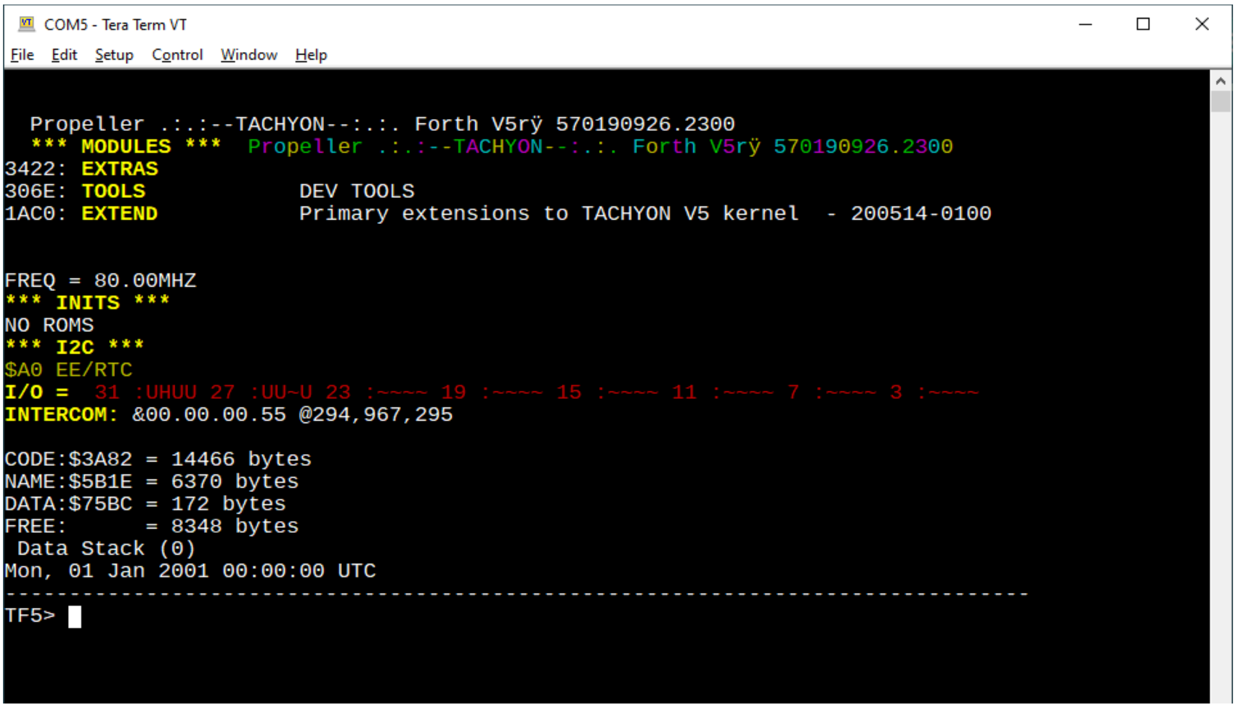
Press OK and the code will start loading into Tachyon with a series of counts displayed. (You'll be doing a lot of this when developing your own code) You'll eventually see:-



```
  Propeller .:.:--TACHYON--:.:. Forth V5r7 NEON 570190926.2300
 Cold start - no user code - setting defaults

--------------------------------------------------------------------------------
TF5> TACHYON V5 ---   Propeller .:.:--TACHYON--:.:. Forth V5r7 NEON 570190926.2300
0100  ---  ok
0200
0300  ---  ok
0400  ---  ok
0500  ---  ok
0600
0700  ---  ok
0800
0900  ---  ok
1000  ---  ok
1100  ---  ok
1200  ---  ok
1300
1400
1500  ---  ok
1600  ---  ok
1700
1800  ---  ok
2000
2100
2200
2300  ---  ok
2500
2600  ---  ok
2605  ---  ok

   End of source code, 0000  errors found  Load time = 30.8
Code bytes used = 8130
Name bytes used = 3716

CODE:$3A82 = 14466 bytes
NAME:$5B1E = 6370 bytes
DATA:$75BC = 172 bytes
FREE:      = 8348 bytes
 Data Stack (0)

 ok
TF5>
TF5> \
TF5> HERE $8000 + ' DEFAULTS 2 + W! ---  ok
TF5> @NAMES $8000 + ' DEFAULTS W! ---  ok
TF5> org@ $8000 + ' DEFAULTS 4 + W! ---  ok
TF5>
TF5> 0 0 INTERCOM! ---  ok
TF5>
TF5> EEPROM ---  ok
TF5> ?BACKUP --- BACKUP  ok
TF5> \
TF5>
```

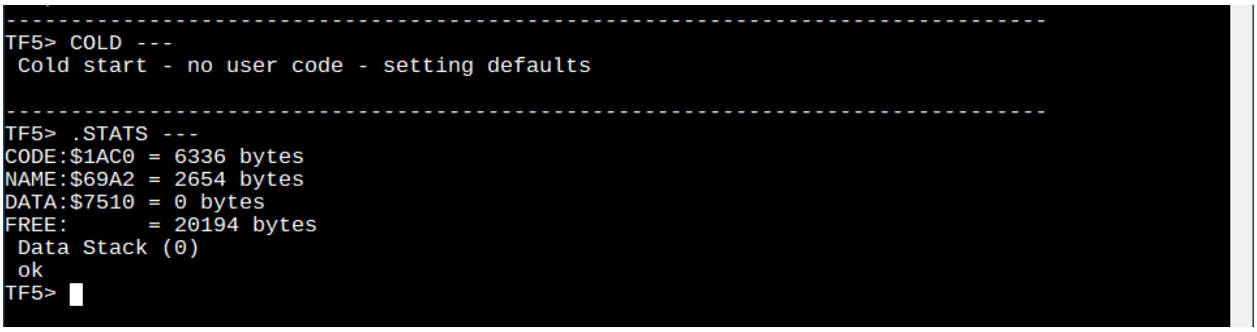
Extend.fth automatically saves the whole forth image to Flash using the ?BACKUP word. This word will not save to EEPROM if any compile error occurred, so watch out for that.

When the P1 is reset, the sign-on message will have changed to:-



```
  Propeller .:.:--TACHYON--:.:. Forth V5rÿ 570190926.2300
  *** MODULES ***  Propeller .:.:--TACHYON--:.:. Forth V5rÿ 570190926.2300
3422: EXTRAS
306E: TOOLS                DEV TOOLS
1AC0: EXTEND               Primary extensions to TACHYON V5 kernel  - 200514-0100


FREQ = 80.00MHZ
*** INITS ***
NO ROMS
*** I2C ***
$A0 EE/RTC
I/O =  31 :UHUU 27 :UU~U 23 :~~~~ 19 :~~~~ 15 :~~~~ 11 :~~~~ 7 :~~~~ 3 :~~~~
INTERCOM: &00.00.00.55 @294,967,295

CODE:$3A82 = 14466 bytes
NAME:$5B1E = 6370 bytes
DATA:$75BC = 172 bytes
FREE:      = 8348 bytes
 Data Stack (0)
Mon, 01 Jan 2001 00:00:00 UTC
--------------------------------------------------------------------------------
TF5> █
```

OK – that’s all there is to that. Notice there is 8348 bytes free for your application – which doesn’t sound much. However, providing your application doesn’t have a need for a lot of data space, that’s enough space for a large forth program - you’ll be surprised. More space can be regained, but you lose all the features of EXTEND.fth. Execute COLD and Tachyon restarts with just the kernel in the dictionary and a lot more free space – 20194 bytes for your program:-



```
--------------------------------------------------------------------------------
TF5> COLD ---
 Cold start - no user code - setting defaults

--------------------------------------------------------------------------------
TF5> .STATS ---
CODE:$1AC0 = 6336 bytes
NAME:$69A2 = 2654 bytes
DATA:$7510 = 0 bytes
FREE:      = 20194 bytes
 Data Stack (0)
 ok
TF5> █
```

Next time the P1 is reset, Tachyon will restart with all the Extend.fth words present again. The effects of executing COLD are only temporary.

SD Card Support

If you fit an SD card, Sandisk cards are known to work, don’t expect much from unmarked cards you found in your bits box. Format the card with SD card formatter from the SD Association to be sure that Tachyon can understand it, else some formats from other sources don’t work. Load Easyfile.fth to provide the SD card driver – this time you’ll have to manually BACKUP the system to keep the extra code after reset. Be warned, this reduces space for your program to a low level.

Conclusion

This has been guidance on installing Tachyon forth onto the Parallax P1 using Windows tools and loading some forth source code from file Extend.fth. This document was written by Bob Edwards, retired EMC engineer and ham radio operator G4BBY in SW U.K. in September 2022. Hope it’s been helpful and it gets you going with Tachyon!!