

# MPE xARM/Cortex

---

## First steps – Olimex STM32-P107 board



---

# First steps – STM32-P107 board

---

## Book name

Manual revision 7.10

Date 05 April 2012

## Software

Software version 7.10

<b>Package</b>	<b>Number:</b>	XARMCTX/71
----------------	----------------	------------

## For technical support

Please contact your supplier

## For further information

MicroProcessor Engineering Limited  
133 Hill Lane, Southampton  
SO15 5AF, UK  
Tel: +44 (0)23 8063 1441  
Fax: +44 (0)23 8033 9691

e-mail: [mpe@mpeforth.com](mailto:mpe@mpeforth.com)  
[tech-support@mpeforth.com](mailto:tech-support@mpeforth.com)

web: [www.mpeforth.com](http://www.mpeforth.com)

---

## **Acknowledgements**

MPE would like to thank the following people for their involvement in the production of this product:

Winfried Clemens

---

# Contents

---

<b>1 Setting up</b>	<b>1</b>
AIDE tools.....	1
AIDE PowerTerm.....	2
Flash Programmer.....	3
Talking to the board.....	3
What you did.....	3
<b>2 Compiling and testing</b>	<b>5</b>
Compile and download.....	5
Adding application code.....	6

---

# 1

## Setting up

---

We assume that you are using a supported board. For the STM32 family, this is usually the Olimex STM32-P107 board.

Find the directory containing the MPE control file

STM32-P107.ctl

and note it down. It should be in

<xArmCortex>/Cortex/Hardware/STM32

### AIDE tools

AIDE is a program that provides manages external tools and provides a terminal emulator. Make sure that everything is set up correctly.

- Run AIDE
- In AIDE, use  
IDE -> Cofigure Edit/Locate  
to set up your favourite editor.
- If the STM32 is not on AIDE's toolbar, add a new tool using:  
IDE -> External Tools

To use the MPE control file

STM32-P107.ctl

set the start directory to the one containing STM32F4disco.ctl. You will need quotation marks if there are spaces in the path. Some people prefer to uninstall and reinstall to a directory with no spaces in the pathname. The set up in my development system is:

BMP file: c:\buildkit.dev\software\Aide\CM4.bmp

Compiler: c:\buildkit.dev\software\compiler\xArmCortexDev.exe

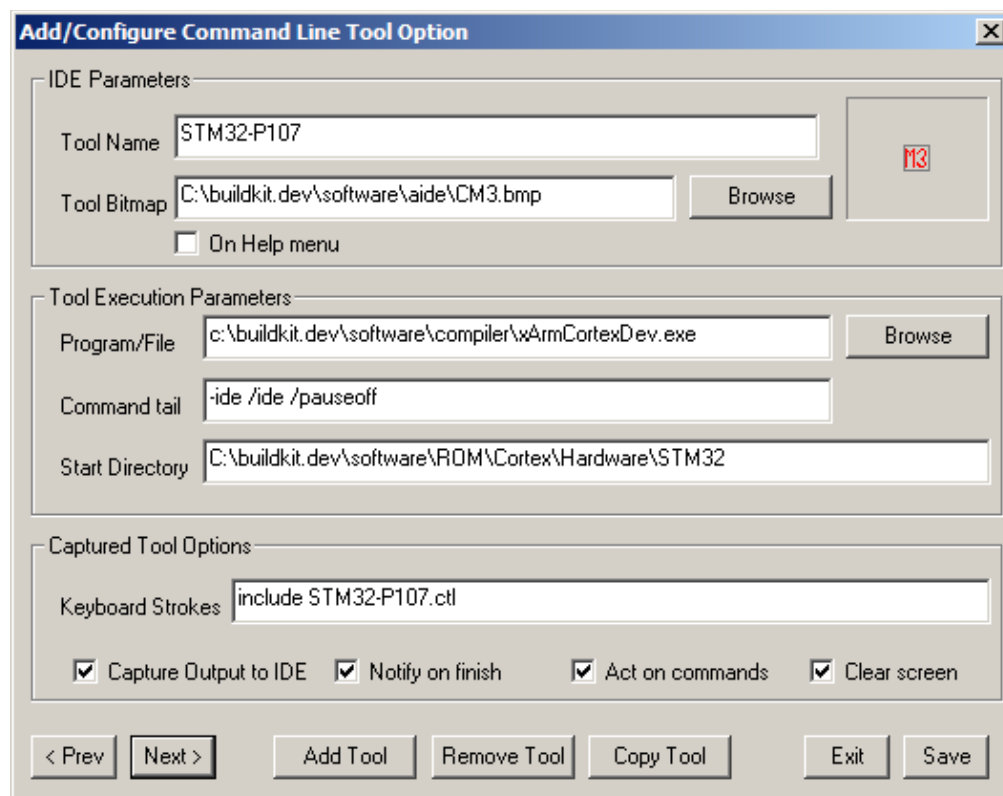
Command tail: -ide /ide /pauseoff

Start dir: C:\buildkit.dev\software\ROM\Cortex\Hardware\STM32

Keybd Strokes: include STM32-P107.ctl

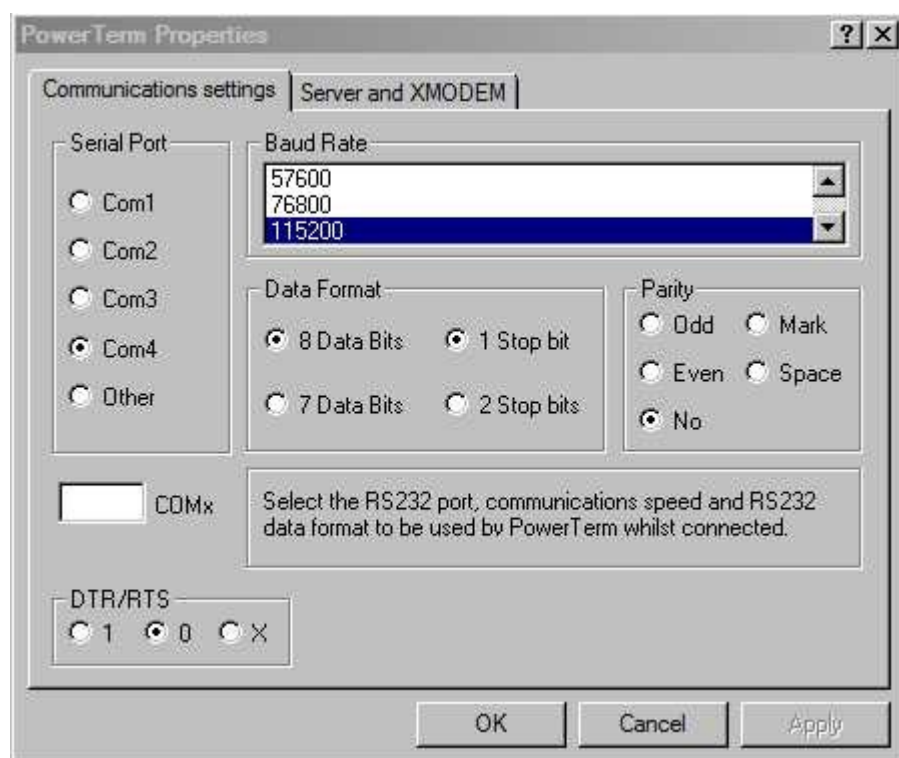
All four checkboxes in the "Captured Tool Options" box are checked. See the following picture for my complete settings. If you are using the Stamp edition of the compiler, change the compiler line:

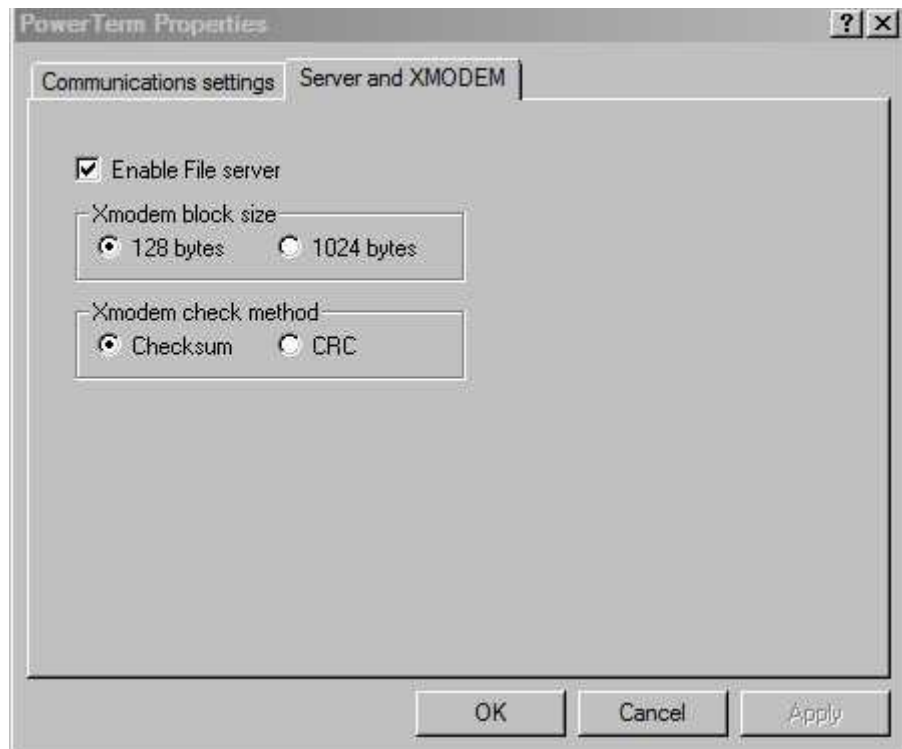
Compiler: c:\buildkit.dev\software\compiler\xArmCortexStamp.exe



## AIDE PowerTerm

Configure PowerTerm for 115200 baud, N81, COMx using the Properties button that is second from the right on the PowerTerm toolbar. Switch to the "Server and XMODEM" page, and check "Enable File Server" with 128 byte and Checksum Xmodem selected.





## Flash Programmer

We use the low-cost ST-LINK/V2 device from ST for programming the board for the first time. Download the STM32 ST-LINK Utility from the ST web site. At the time of writing (March 2012), we obtained this from:

<http://www.st.com/internet/evalboard/product/219866.jsp>

The utility uses the ST-LINK to program the Flash on the chip. If you do not want to use the ST-LINK, you can use other programmers or the bootloader built into the STM32 devices. The programmer should be able to program a binary image file or an Intel HEX file.

## Talking to the board

Use a simple serial cable to the connector labelled “RS232” on the board. You can also use a serial to USB adapter cable such as one supplied by FDTI.

## What you did

You now have the Forth cross compiler tool set up. There will be a corresponding button on the toolbar.

When you click the compiler button the file

STM32-P107.ctf

is included by the cross-compiler. This file is a control file. It tells the cross compiler how to compile the target and what to compile. Because the tool capture checkboxes were checked when the tool was set up, the compiler runs in the “Tool Capture” window.





## 2 Compiling and testing

### Compile and download

You now have a tool set up. There will be a corresponding button on the toolbar. When you click it, the file

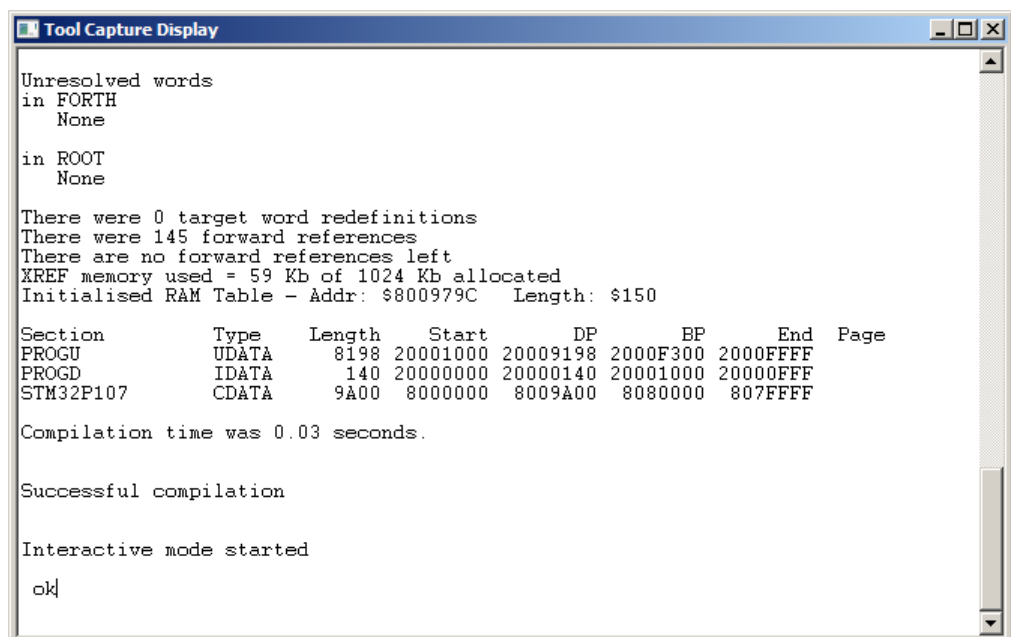
STM32-P107.ctl

is included by the cross-compiler. This file is a control file. It tells the cross compiler how to compile the target and what to compile. When the compiler has finished, there will be two output files in your STM32F4 folder

STM32F4DISCO.img

STM32F4DISCO.hex

The cross compiler always produces binary image files with a ".img" extension. The control file tells the compiler also to produce an Intel Hex file with a ".hex" extension. The binary files are used by MPE tools, the hex files by some external tools for Flash programming.



AIDE's Tool Capture window should be visible - the compiler is still alive so that you can disassemble words and use the cross reference tools such as **XREF** and **LOCATE**. Make sure that PowerTerm is also visible, but is disconnected.

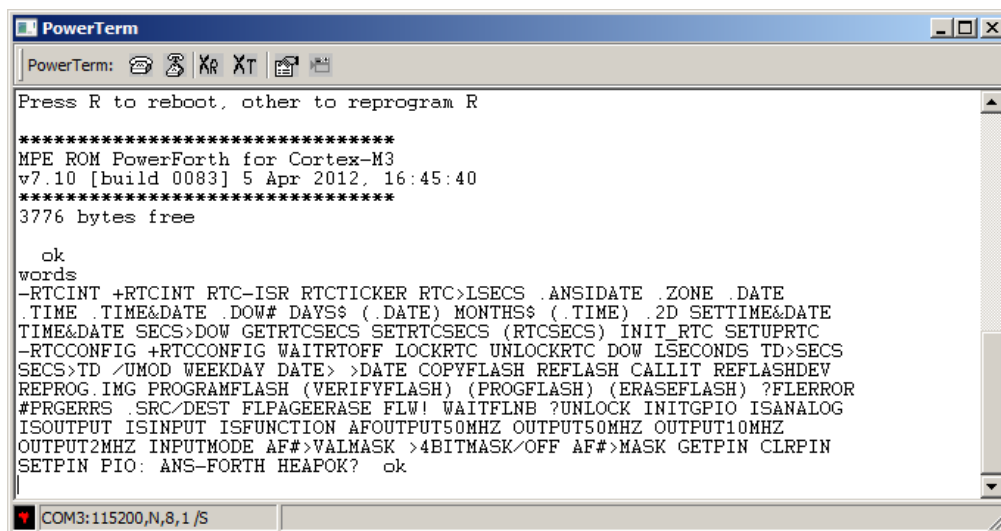
You now need to program one of the output files into the STM32 Flash – the STM32 ST-LINK Utility is the most convenient option.

Once the board is programmed and powered, press the PowerTerm connect button (the telephone) and then reset the board. You should now get the MPE PowerForth sign on message. This is the target Forth on the board. You can use this as a normal Forth, and you can even compile code on it. However, since the cross compiler takes

a fraction of a second, and the Forth has facilities to reflash itself, it is quicker just to recompile and reflash. To use the Forth facilities, just type

### REFLASH

AIDE will put up a box asking for an image file. Navigate to the STM32F4 folder and select STM32F4DISCO.img. AIDE will download this to the target and reflash the target. If the process has no errors, at the end you will be offered R to reboot. The new target image will then run.



```

PowerTerm
PowerTerm: [Phone] [Printer] [X] [T] [F] [E]
Press R to reboot, other to reprogram R

*****
MPE ROM PowerForth for Cortex-M3
v7.10 [build 0083] 5 Apr 2012, 16:45:40
*****
3776 bytes free

ok
words
-RTCINT +RTCINT RTC-ISR RTCTICKER RTC>LSECS .ANSIDATE .ZONE .DATE
.TIME .TIME&DATE .DOW# DAYS$ (.DATE) MONTHS$ (.TIME) .2D SETTIME&DATE
TIME&DATE SECS>DOW GETRTCSECS SETRTCSECS (RTCSECS) INIT_RTC SETUPRTC
-RTCCONFIG +RTCCONFIG WAITRTOFF LOCKRTC UNLOCKRTC DOW LSECONDS TD>SECS
SECS>TD /UMOD WEEKDAY DATE> >DATE COPYFLASH REFLASH CALLIT REFLASHDEV
REPROG.IMG PROGRAMFLASH (VERIFYFLASH) (PROGFLASH) (ERASEFLASH) ?FLERROR
#PRGERRS .SRC/DEST FLPAGEERASE FLW! WAITFLNB ?UNLOCK INITGPIO ISANALOG
ISOUTPUT ISINPUT ISFUNCTION AFOUTPUT50MHZ OUTPUT50MHZ OUTPUT10MHZ
OUTPUT2MHZ INPUTMODE AF#>VALMASK >4BITMASK/OFF AF#>MASK GETPIN CLRPIN
SETPIN PIO: ANS-FORTH HEAPOK? ok

COM3:115200,N,8,1/5
  
```

You just repeat the recompile and reflash cycle as required.

If the **REFLASH** operation fails, you will have to use the ST-LINK Utility tool to put the image or hex file into the target again.

## Adding application code

Control files and text macros are important in MPE Forth systems. They are documented in the main cross compiler manual. Please read these sections of the manual.

Although you can just add new lines to the MPE control files, we do not recommend this. You run the risk of losing your changes when you update the compiler. Make your own project. You do this by making a copy of STM32F4disco.ctl and using it as the basis of your project. MPE suggests that you leave our code alone and make a new project that is **not** in the cross compiler folder.

To do this, make a new folder and put the renamed control file in it, (say) MySTM32F4disco .ctl. Also copy over STM32F4disco.no and rename it as (say) mySTM32F4disco .no. Use AIDE's "External Tools" "Copy Tool" button to make a new project button and edit the start directory and include file name.

Edit the control file's text macros that start at line 50 so that they point to the right folders. Edit the build file name on line 332. Now use your new button to recompile and then reflash the new image.