

nanoMODUL-164

QuickStart Instructions

**Using PHYTEC FlashTools for Windows and the TASKING
EK166 Software Evaluation Development Tool Chain**

Note: The PHYTEC Spectrum CD includes the electronic version of
the nanoMODUL-164 English Hardware Manual

Hinweis: Die PHYTEC Spectrum CD beinhaltet die elektronische
Version des deutschen nanoMODUL-164 Hardware Manuals

Edition: August 1999

A product of a PHYTEC Technology Holding company

In this manual are descriptions for copyrighted products that are not explicitly indicated as such. The absence of the trademark (™) and copyright (©) symbols does not imply that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual.

The information in this document has been carefully checked and is believed to be entirely reliable. However, PHYTEC Meßtechnik GmbH assumes no responsibility for any inaccuracies. PHYTEC Meßtechnik GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. PHYTEC Meßtechnik GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, PHYTEC Meßtechnik GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. PHYTEC Meßtechnik GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2001 PHYTEC Meßtechnik GmbH, D-55129 Mainz.

Rights - including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part - are reserved. No reproduction may occur without the express written consent from PHYTEC Meßtechnik GmbH.

	EUROPE	NORTH AMERICA
Address:	PHYTEC Technologie Holding AG Robert-Koch-Str. 39 D-55129 Mainz GERMANY	PHYTEC America LLC 255 Ericksen Avenue NE Bainbridge Island, WA 98110 USA
Ordering Information:	+49 (800) 0749832 order@phytec.de	+1 (800) 278-9913 info@phytec.com
Technical Support:	+49 (6131) 9221-31 support@phytec.de	+1 (800) 278-9913 support@phytec.com
Fax:	+49 (6131) 9221-33	+1 (206) 780-9135
Web Site:	http://www.phytec.de	http://www.phytec.com

1st Edition: August 1999

1	Introduction to the Rapid Development Kit	1
1.1	Documentation Overview	1
1.2	Overview of this QuickStart Instruction.....	2
1.3	System Requirements	2
1.4	The PHYTEC nanoMODUL-164.....	4
1.5	The TASKING C166/ST10 Embedded Development Environment (EDE).....	6
2	Getting Started.....	9
2.1	Installing Rapid Development Kit Software.....	9
2.2	Interfacing the nanoMODUL-164 on the Development Board to a Host-PC.....	16
2.3	Downloading Example Code.....	18
2.3.1	Starting PHYTEC FlashTools for DOS	18
2.3.2	Downloading Example Code with FlashTools for DOS	19
2.3.2.1	‘Hello’	24
2.3.2.2	‘Blinky’	29
2.3.3	Downloading Example Code with FlashTools for Windows.....	31
2.3.3.1	‘Hello’	35
2.3.3.2	‘Blinky’	40
3	Getting More Involved.....	41
3.1	Starting the TASKING Software Tool Chain.....	41
3.2	Creating a New Project and Adding an Existing Source File.....	43
3.3	Modifying the Source Code.....	46
3.4	Saving the Modifications.....	47
3.5	Setting Tool Chain Options	47
3.6	Building the Project	51
3.7	Downloading the Output File	51
3.7.1	Downloading with FlashTools for DOS.....	51
3.7.2	Downloading with FlashTools for Windows.....	54
3.8	‘Hello’.....	55
3.8.1	Creating a New Project.....	55
3.8.2	Modifying the Example Source	56
3.8.3	Setting Tool Chain Options	56
3.8.4	Building the New Project	56

3.8.5	Downloading the Output File	57
3.8.5.1	Downloading with FlashTools for DOS	57
3.8.5.2	Downloading with FlashTools for Windows	59
3.8.6	Starting the Terminal Emulation Program	60
4	Debugging	61
4.1	Creating a Debug Project and Preparing the Debugger	61
4.2	Preparing the Target Hardware to Communicate with CrossView Pro.....	63
4.3	Starting the Debugger.....	64
4.4	Using the TASKING CrossView Pro Debug Features	65
4.4.1	Breakpoints.....	65
4.4.2	Data Window.....	65
4.5	Single Stepping.....	66
4.5.1	Memory Window.....	66
4.5.2	Stack Window	67
4.5.3	Simulated I/O	67
4.6	Resetting CrossView Pro and the nanoMODUL-164	68
5	Advanced User Information.....	69
5.1	FlashTools	69
5.1.1	FlashTools for DOS	69
5.1.2	FlashTools for Windows	70
5.2	Start.asm	72
5.3	Linking and Locating	73
5.4	Debugging Using Monitor Kernel.....	74
5.5	Using ROM Debug Monitor in Flash Memory	75
	Appendices.....	77
A	Troubleshooting.....	77
A.I	Initialization Error: ROM Monitor DLL could not Initialize or Access the Target Board.....	77
A.II	Program download never stops	77
A.III	Placing Code Breakpoint at 0xXXXXXX Failed: Memory not Writeable	78

Figures

Figure 1: Development Board Overview	16
Figure 2: Suitable Development Board Jumper Settings	17
Figure 3: Power Connector	17
Figure 4: Suitable Development Board Jumper Settings for Debugging ...	63

1 Introduction to the Rapid Development Kit

This tutorial provides:

- general information on the PHYTEC nanoMODUL-164 Single Board Computer
- an overview of the TASKING C166/ST10 Embedded Development Environment (EDE) evaluation version, and
- instructions on how to run example programs on the nanoMODUL-164, mounted on a PHYTEC Development Board, in conjunction with TASKING tools.

Please refer to the [nanoMODUL-164 Hardware Manual](#) for specific information on board-level features as such as [Jumper configuration](#), [memory mapping](#) and [pin layout](#). Selecting the links on the electronic version of this document takes you to the applicable section of the nanoMODUL-164 Hardware Manual.

1.1 Documentation Overview

This “Rapid Development Kit” includes the following electronic documentation on the “PHYTEC Spectrum CD”:

- the PHYTEC [nanoMODUL-164](#) and [Development Board Hardware Manual](#)
- controller [User's Manuals and Data Sheets](#)
- this QuickStart Instruction with general “Rapid Development Kit” description, software installation hints and three example programs enabling quick out-of-the box start-up of the nanoMODUL-164 in conjunction with the TASKING software development tools.

1.2 Overview of this QuickStart Instruction

- 1) The “*Getting Started*” section uses two example programs – ‘Helloand ‘*Blinky* demonstrate the download of user code to the Flash device using PHYTEC’s FlashTools for Windows.
- 2) Section “*Getting More Involved*” provides step-by-step instructions on how to modify both examples, create and build new projects and likewise download output files to the nanoMODUL-164 using the Windows-based TASKING tools.
- 3) The “*Debugging*” section provides a third example program “Debug” to demonstrate monitoring of the board and simple debug functions using the TASKING debug environment.

In addition to dedicated data for this Rapid Development Kit, this CD-ROM contains supplemental information on embedded microcontroller design and development.

1.3 System Requirements

Use of this ‘Rapid Development Kit’ requires:

- the PHYTEC nanoMODUL-164,
- the PHYTEC Development Board with the supplied DB-9 serial cable and AC adapter supplying 8-12 V./min. 500 mA.,
- the PHYTEC Spectrum CD,
- an IBM-compatible host-PC (486 or higher running at least Windows 95/NT)

For more information and example updates, please refer to the following sources:

PHYTEC

<http://www.phytec.com> - or - <http://www.phytec.de>
support@phytec.com - or - support@phytec.de

 **TASKING**
Quality Development Tools Worldwide

<http://www.tasking.com>
support_de@tasking.com (Germany)
support@tasking.com (International)

1.4 The PHYTEC nanoMODUL-164

The nanoMODUL-164 is intended for use in memory-intensive applications and in running within a CAN-bus network. Half of the size of a credit card (47 by 38 mm.), the standard board is equipped with the Infineon SABC164CI controller as well as a Real-Time Clock which, like the SRAM, can be buffered by an external battery for backup. The standard module runs at a 20 MHz internal clock speed (delivering 100 instruction cycles/ns) and offers 256 kByte (up to 1 MB) SRAM and 256 kByte (up to 1 MB) Flash on-board for DATA and CODE storage. PHYTEC FlashTools enable easy on-board download of user code to the external Flash device.

The nanoMODUL-164 is fitted with one RS-232 transceiver, an I²C RTC and a CAN-bus interface. All controller signals and ports extend to pin rows aligning two edges of the board. These pins provide a 16-bit bi-directional I/O port and 64 free port lines, including 8 analog inputs with 10-bit resolution. The nanoMODUL-164 has an industrial temperature range from 0 to 70 degrees C (with extended range from -40 to 85 degrees C except RTC only from -30 to 70 degrees) and requires only a 300 mA power source.

nanoMODUL-164 Technical Highlights

- SBC in matchbox-sized dimensions (47 x 38 mm.) achieved through modern SMD technology
- fitted with Infineon SABC164CI controller with Full 2.0B on-chip CAN
- Instruction cycle time of 100 ns. at 20 MHz. clock speed in 16-bit, multiplexed bus-mode
- 256 kByte (to 1 MB) SRAM on-board and 256 kByte (to 1 MB) Flash on-board
- RS-232 interface
- Full 2.0B CAN-bus interface
- battery-buffered Real-Time-Clock and SRAM
- provision for operation with in-circuit emulators (such as ICE connect-16x)
- requires only +5 V/300 mA. power source in a temperature range of 0 to 70 degree C.

The PHYTEC Development Board, in EURO-card dimensions (160 x 100 mm.) is completely equipped with all mechanical and electrical components necessary for the speedy and secure insertion and subsequent programming of the PHYTEC nanoMODUL series Single Board Computer. Simple Jumper configuration readies the Development Board's for connection to any PHYTEC module, which plugs connectors-down into the high density receptacle footprint connector mounted on the Development Board.

Development Board Technical Highlights

- a RESET-switch
- a BOOT-switch
- a low-voltage socket and voltage regulator accepting an unregulated input voltage in a range from 8 to 12 VDC.
- a DB-9 connector (configured as an RS-232 interface)
- a second DB-9 interface for CAN
- a VG96-connector
- simple jumper configuration
- a wire wrap field (64 x 69 mm.) supporting development of user-designed peripheral hardware.

1.5 The TASKING C166/ST10 Embedded Development Environment (EDE)

The TASKING software tool chain fully supports the entire Infineon C16X/ST Microelectronics ST10 microcontroller family. It includes a C/C++/EC++ compiler, macroassembler, Linker/Locator and the CrossView Pro Simulator and ROM Monitor Debugger within the EDE development environment. Specific chips supported are the Infineon 161, 163, 164, 165, 166, 167, and ST Microelectronics ST10/262. Future derivatives are easily accommodated due to the flexible TASKING C compiler design.

The TASKING software tool chain supports all in-circuit emulators that adhere to the IEEE-695 debugging specification or support the TASKING a.out format. The IHEX166 Object-to-Hex converter converts an absolute object file into an Intel-hexfile that is suitable for programming into an EPROM device or downloading into external Flash on the PHYTEC nanoMODUL-164 target board.

The TASKING software tool chain consists of the following executables:

- **C++/EC++ Comp.** cp166.exe (not in eval version)
- **C Compiler** c166.exe
- **Macro Preprocessor** m166.exe
- **Assembler** a166.exe
- **Linker/Locator** l166.exe
- **HEX converter** ihex166.exe
- **Make Utility** mk32.exe
- **Steering Program** cc166.exe
- **IEEE-695 converter** ieee166.exe
- **SREC converter** srec166.exe (not in eval version)
- **Disassembler** d166.exe (not in eval version)
- **Object dumper** dmp166.exe (not in eval version)
- **Library archiver** ar166.exe (not in eval version)
- **CrossView Pro** xfw166.exe (Windows-based)
- **EDE** ede.exe (Windows-based)

Once installed, the default destination location for these files is **C:\DC166\bin** for the evaluation version. If using the professional (i.e. full) version of the TASKING software tool chain, the default destination location for these files is **C:\C166\bin** . The executables are 32-bit Windows programs and run in command line mode except for CrossView Pro and EDE which are Windows-based applications. Access to these programs from Windows is done with EDE. The entire tool set can be run from EDE or directly from a command line using “batch” or “make” files. The limitations of the evaluation version are listed in the ‘Demo Package Readme’ file. Other than these restrictions, all features operate normally.

The TASKING tools are also available for the UNIX operating system.

Embedded Development Environment (EDE)

EDE is a Windows-based Graphical User Interface (GUI) for the C compiler and assembler portion of the TASKING software tools. All compiler, assembler and linker options are set with simple mouse clicks. EDE runs under Windows 95/98/2000 and NT. This Embedded Development Environment has been designed with the user in mind and includes a fully functional editor based on the popular Codewright from Premia.

All EDE commands and functions are accessible via pull-down menus with prompted selections. An extensive Help utility and the complete set of online manuals are included. External executables can be run from within EDE, including emulator software.

C166 C Compiler

The C166 ANSI compiler and A166 assembler are designed specifically for the Infineon SAB 161, 163, 164, 165, 166, 167 and ST Microelectronics ST10/262 and future derivatives.

The TASKING C166 compiler provides the fastest and smallest code using industry benchmarks.

M166 Macro Preprocessor/A166 Assembler

The macro preprocessor and the assembler are started by the EDE environment. It is also possible to pass a file directly to the assembler when it does not need any macro expansion..

Debug Environment

CrossView Pro is a software simulator/ROM monitor debugger supporting debugging either via software on a host-PC or in target hardware. CrossView Pro enables the following debugging functions:

- run/halt,
- set (complex) breakpoints,
- examine/change memory,
- view the stack,
- simulated I/O
- extended logging and scripting language
- powerful C like command language
- communication via RS-232 or CAN

The CrossView Pro Simulator supports code coverage and profiling to ensure your code runs efficiently, and high level language trace. The restrictions on evaluation version of the TASKING C166 tool set are described in the 'Demo Package Readme'. Other than these restrictions the evaluation tool chain functions exactly as does the full version. The evaluation version does not have a starting address restriction and produces useful object code. This allows you to fully evaluate the features and power of TASKING products on a PHYTEC target board. The full version has no restrictions and is fully ANSI compliant.

CAN (Controller Area Network) Library

The Infineon CAN protocol driver software routines including pre-built CAN libraries are supplied with the 32-bit Windows 95/NT version of this product. The file ap292201.pdf describes the usage of these libraries. This file is located in the pdf folder on the Spectrum CD.

2 Getting Started

What you will learn with this Getting Started example:

- installing the Rapid Development Kit software
- starting the PHYTEC FlashTools download utility
- interfacing the nanoMODUL-164 Rapid Development Kit, to a host-PC
- downloading example code in hexfile format from a host-PC to the external Flash memory using FlashTools

2.1 Installing Rapid Development Kit Software

When you insert the PHYTEC Spectrum CD into the CD-ROM drive of your host-PC, the PHYTEC Spectrum CD should automatically launch a setup program that installs the software required for the Rapid Development Kit as specified by the user. Otherwise the setup program *start.exe* can be manually executed from the root directory of the PHYTEC Spectrum CD.

The following window appears:

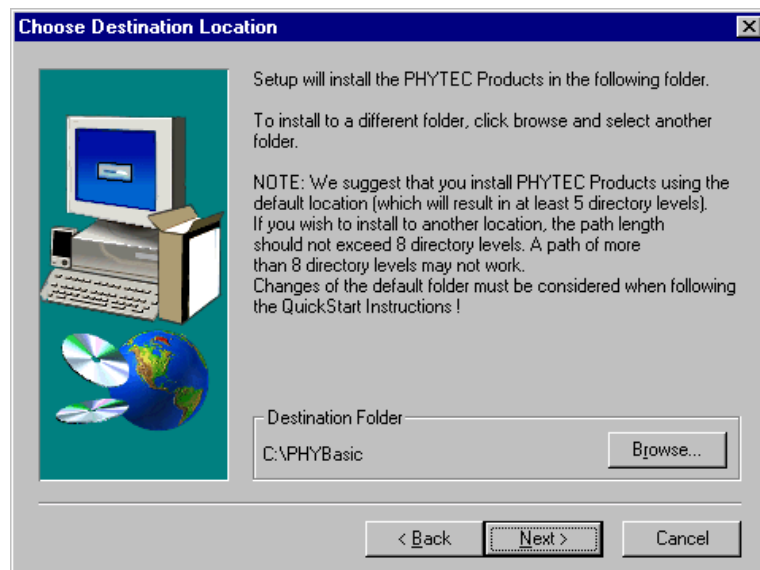


Choose *Install Basic Product Files* Button.

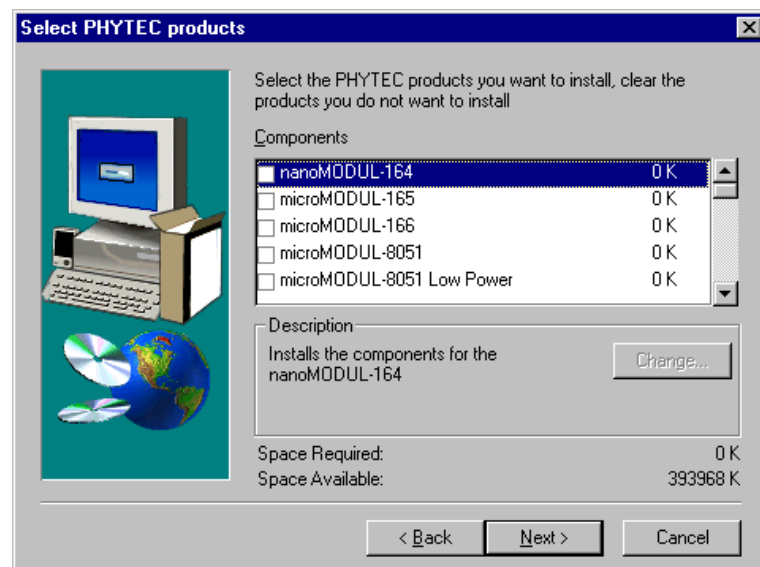
After accepting the Welcome window and license agreement select the destination location for installation of Rapid Development Kit software and documentation.

The default destination location is *C:\PHYBasic*. All path and file statements within this QuickStart Instruction are based on the assumption that you accept the default install paths and drives. If you decide to individually choose different paths and/or drives you must consider this for all further file and path statements.

We recommend that you accept the default destination location.



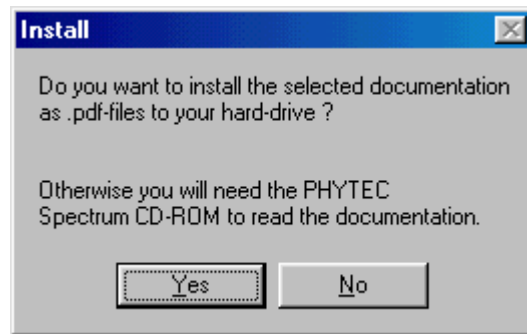
In the next window select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.



All Kit-specific content will be installed to a Kit-specific subdirectory of the Rapid Development Kit root directory that you have specified at the beginning of the installation process.

All software and tools for this nanoMODUL-164 Kit will be installed to the **|PHYBasic** directory on your hard-drive.

In the next dialog you must choose whether to copy the selected documentation as *.pdf-files to your hard-drive or to install a link to the doc. On the Spectrum CD.

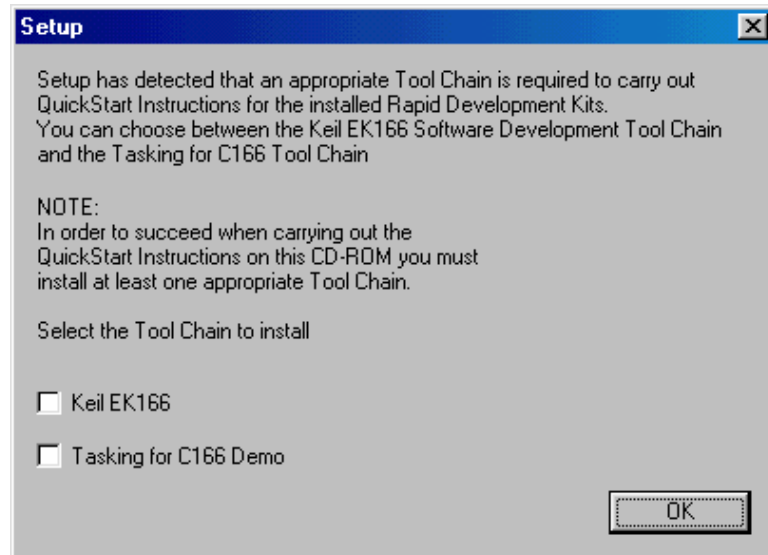


If you decide **not** to copy the documentation to your hard-drive you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

If you decide to copy the electronic documentation to your hard-drive, the documentation for this nanoMODUL-164 Kit will also be installed to the Kit-specific subdirectory. The manuals of the Development Boards are copied to their own specific subdirectories (e.g. **|PHYBasic|DevBnM**) because each Development Board is suitable for multiple Single Board Computers and is not dedicated to a specific Kit.

Setup will now add program icons to the program folder, named *PHYTEC*.

In the next window, you choose Tasking for C166 Demo Software Development Tool Chain.



After accepting the Welcome window and license agreement select the destination location for installation of the Development Tool Chain.

Depending on the Rapid Development Kit software you have selected, the applicable Tasking Evaluation Development tool chain will be installed to your hard-drive. Additional software, such as Adobe Acrobat Reader, will also be offered for installation.

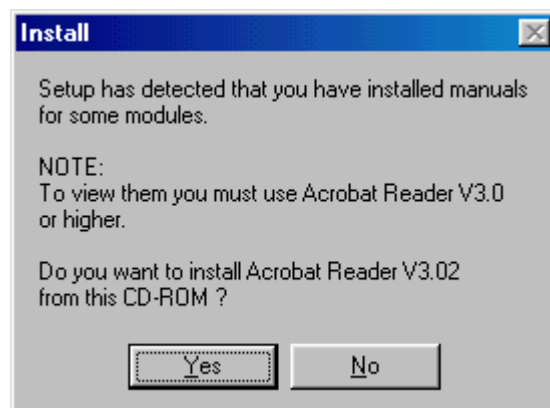
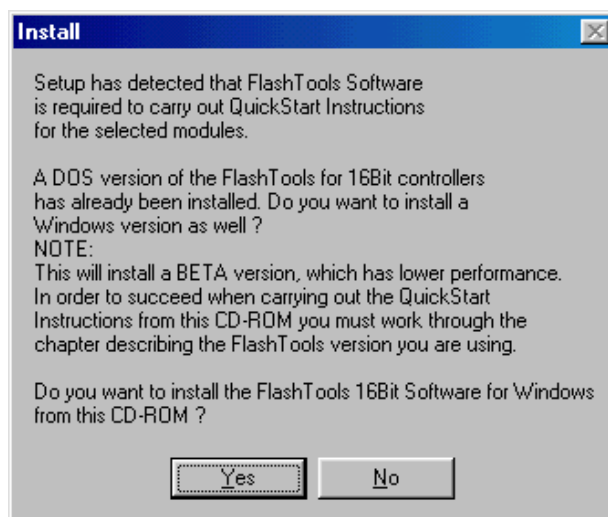
Note:

Tasking Evaluation Development tool chain is limited for 30 days since the first installation.

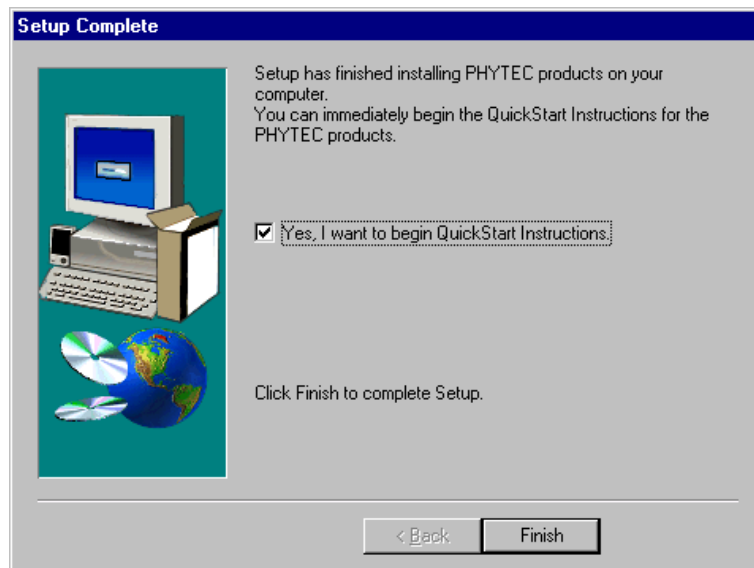
The applicable TASKING tool chain must be installed to ensure successful completion of this QuickStart Instruction. Failure to install the proper software could lead to possible version conflicts, resulting in functional problems.

We recommend that you install Tasking C166/ST10 Embedded Development Environment (EDE) evaluation version from the Spectrum CD-ROM even if other versions of EDE are already installed on your system. These QuickStart Instructions and the demo software included on the CD-ROM have been specifically tailored for use with one another.

In the following windows you can decide to install FlashTools for Windows Beta Version and the Acrobat Reader. For better performance, you can use the DOS Flashtools which have already been installed.



Press Finish to complete the installation and decide if you want to begin the QuickStart Instruction immediately.

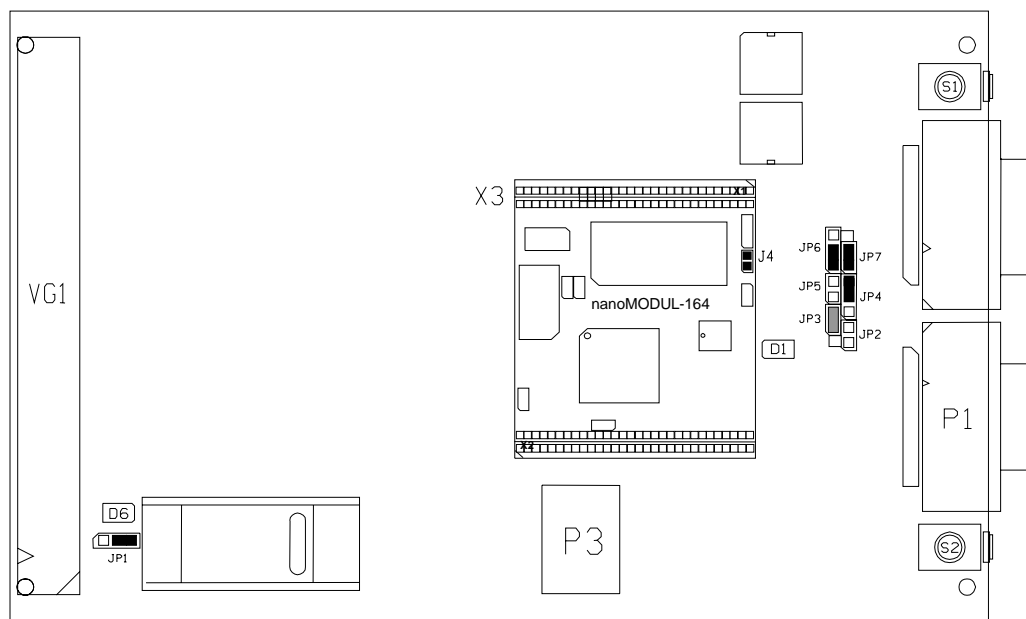


2.2 Interfacing the nanoMODUL-164 on the Development Board to a Host-PC

Connecting the nanoMODUL-164 on the PHYTEC Development Board, to your computer is simple:

- As shown in the figure below, if the nanoMODUL is not already pre-installed, mount it pins-down onto the Development Board's receptacle footprint (X3).

Ensure that there is a solid connection between the modul pins and the Development Board receptacle. Also take precautions not to bend the pins when the nanoMODUL is removed from and inserted onto the Development Board.



PHYTEC Meßtechnik GmbH Development Board nanoMODUL-164

Figure 1: Development Board Overview

- Set the Jumpers on the Development Board as shown below. This correctly routes the RS-232 signals to the DB-9 connector (P1) on the Development Board.

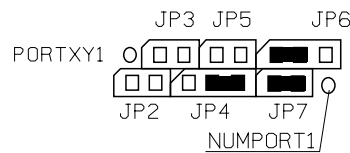


Figure 2: Suitable Development Board Jumper Settings

- Set Jumper JP1 on the Development Board as shown below to supply the board through the power socket (P3).



- Connect the RS-232 interface of your computer to the DB-9 RS-232 interface on the Development Board (P1) using the supplied serial cable.
- Using the supplied power adapter, connect the power socket on the board (P3) to a power supply (refer to Figure 3 for the right polarity).

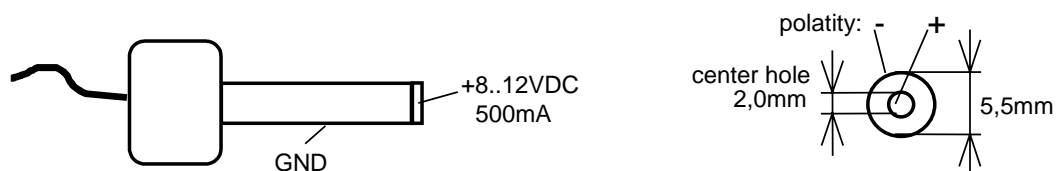


Figure 3: Power Connector

- Simultaneously press the Reset (S1) and Boot (S2) switches on the Development Board, first releasing the Reset (S1) and then, two or three seconds later, release the Boot (S2) switch.

This sequence of pressing and releasing the Reset (S1) and Boot (S2) switches puts the nanoMODUL-164 on the Development Board into the Bootstrap mode. FlashTools must always be operated in this mode. See *section 5.1.2* for more details. Ensure that the program `flasht.exe` is started only after first resetting the board.

The nanoMODUL-164 on the Development Board should now be properly connected via the Development Board to a host-PC and power supply. After executing a Reset and putting the board in Bootstrap mode, you are now ready to download a program to the nanoMODUL-164 on the Development Board.

2.3 Downloading Example Code

2.3.1 Starting PHYTEC FlashTools for DOS

FlashTools was copied to your hard drive during the initial setup procedure as described in *section 2.1*. A link also has been added to your *PHYTEC Rapid Development Kit/nanoMODUL-164* program group which allows you to easily start FlashTools by selecting the appropriate icon.

Alternately you can manually copy FlashTools from the `\nM164\Tools\Flasht` folder of your PHYTEC Spectrum CD. FlashTools consists of three files—*flasht.exe* (user interface as DOS executable file), *boot* (the second stage loader), and *flash* (the FlashTools microcontroller firmware). FlashTools for DOS (*flasht.exe*) is a utility program that allows download of user code in **.hex*-file format from a host-PC to a PHYTEC Single Board Computer (SBC) via an RS-232 connection. FlashTools uses the Bootstrap Loader mechanism of the nanoMODUL-164 to transfer and execute the firmware. FlashTools consists of a firmware transferred to the external RAM using the Bootstrap Loader and corresponding software executed at the host-PC. Proper connection of a PHYTEC SBC to a host-PC enables the software portion of FlashTools to recognize and communicate with the firmware portion.

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash you must use the three files from the sub-folder *FlashtMx* in the *Flasht* folder. This is because an external GAL device is used to configure large or small Memory-Models at a hardware level. Both Memory-Model types require a special version of *FlashTools*.

Each set of *boot* and *flash* files found in the controller specific folder of the PHYTEC Spectrum CD is specially tailored for use with a specific controller on a PHYTEC Single Board Computer. Do not interchange files because the wrong set will not work with other controllers and will lead to a configuration error. Refer to *section 5.1.1* for details.

- You start *FlashTools* for DOS by selecting either the *FlashTools (COM1)* or *FlashTools (COM2)* icon within the *Programs|PHYTEC Rapid Development Kit|nanoMODUL-164* program group. We recommend to use the installed icons.

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash use the icons *FlashTools Max (COM1)* or *FlashTools Max (COM2)*.

- You can also start *FlashTools* by running the executable from a DOS prompt by typing *flasht br(9600) x*. This sets the baud rate to 9600 baud and uses serial port *x* (*x* = appropriate COM port of your computer).

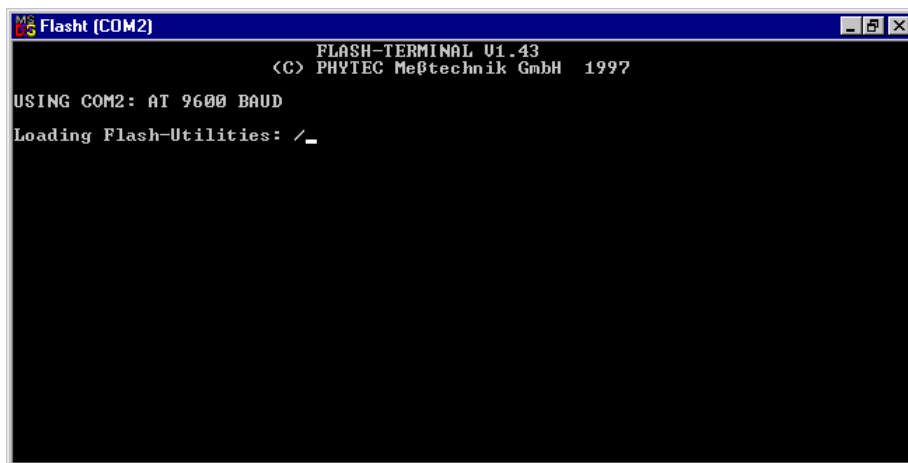
2.3.2 Downloading Example Code with *FlashTools* for DOS

- Start *FlashTools* for DOS for your nanoMODUL-164 on the Development Board by selecting either the *FlashTools (COM1)* or *FlashTools (COM2)* icon from within the ***PHYTEC Rapid Development Kit|nanoMODUL-164*** program group. Which icon to choose depends on the serial port used on your host-PC to connect to the target system.

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash use the icons *FlashTools Max (COM1)* or *FlashTools Max (COM2)*.

The microcontroller firmware tries to automatically adjust to the baud rate entered in the command line within the properties of the link. However it may occur that the predefined baud rate can not be reached. This results in FlashTools omitting the loading procedure and never showing the main menu. In this case, try incrementally lower baud rates to establish a connection. Before further connection attempts be sure to leave FlashTools by pressing function key <F1> and then to reset the target hardware and force it into Bootstrap mode as described in *section 2.2*.

- If properly started, FlashTools will load to RAM of the nanoMODUL-164. The active loading process is indicated by a rotating cursor bar.



- After FlashTools has loaded, the main menu appears on the screen.

```

Flashht (COM2)
=====
FLASH-Utility for 80C164 with 16-Bit Flash V 2.5
=====
(c) 1997, PHYTEC Meßtechnik GmbH, D-55129 Mainz

Flash-Devices:          AMD 29F200B
Flash-Area:            000000H-07FFFFH
Software-Protected-Areas #1: none
                      #2: none
                      #3: none
No-Access-Areas      #1: 00EF00H-00FFFFH
                      #2: none
                      #3: none

(1) Flash status information
(2) Erase entire Flash-Area
(3) Erase partial Flash-Area
(4) Load INTEL-Hexfile
(5) Software-Reset
(6) Run from address
(7) Erase, Load and Software-Reset

> Command:

```

At the FlashTools main menu you will see commands for the following:

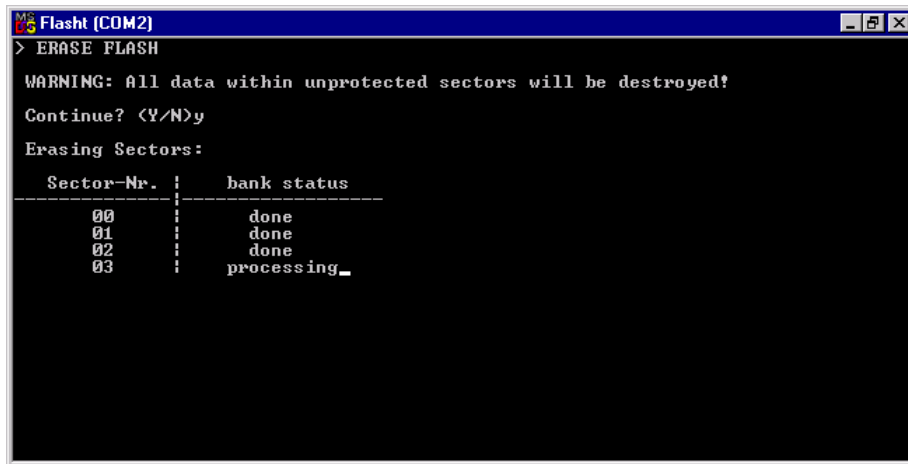
Flash Status information shows sector and address ranges in Flash memory:

```

Flashht (COM2)
> STATUS INFORMATION
Device information:
Manufacturer code = 01, Device code = 57
Flash-Device:      AMD 29F200B
Address range information:
Flash-Area:        000000H-07FFFFH
Software-Protected-Areas #1: none
                      #2: none
                      #3: none
No-Access-Areas  #1: 00EF00H-00FFFFH
                      #2: none
                      #3: none
Sector information: Access to 7 sectors
Sector-Nr.  : bank status
-----
00  : unprot. full acc. blank
01  : unprot. full acc. blank
02  : unprot. full acc. blank
03  : unprot. part. acc. blank
04  : unprot. full acc. blank
05  : processing

```

Erase entire Flash-area enables erasure of the unprotected sectors of the whole user addressable Flash:



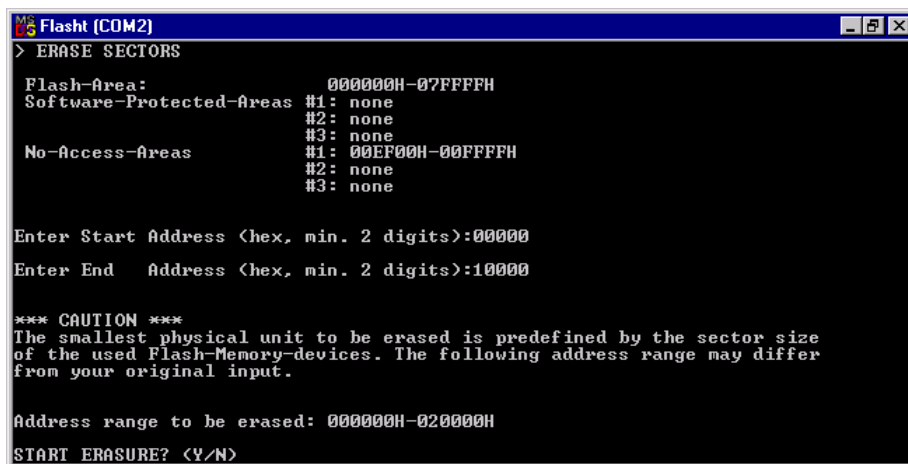
```
Flasht (COM2)
> ERASE FLASH

WARNING: All data within unprotected sectors will be destroyed!
Continue? <Y/N>y

Erasing Sectors:

Sector-Nr. | bank status
-----|-----
    00    | done
    01    | done
    02    | done
    03    | processing_
```

Erase partial Flash-area enables erasure of user specified sectors of memory by entering a memory address range:



```
Flasht (COM2)
> ERASE SECTORS

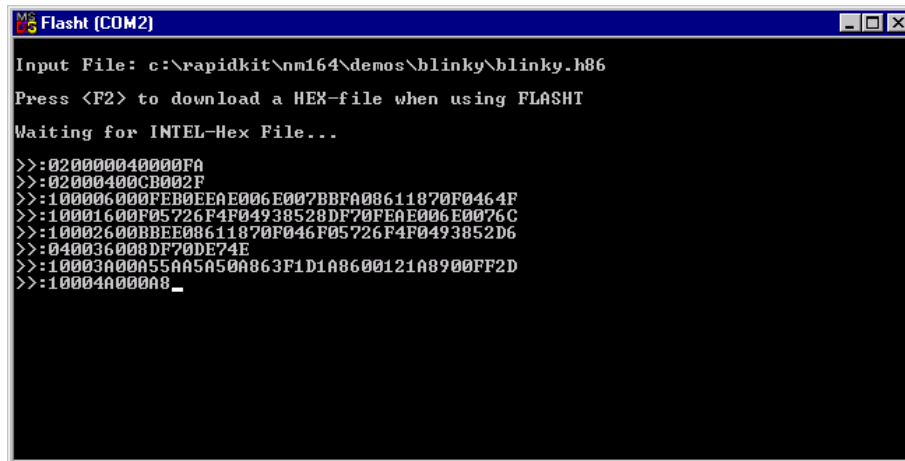
Flash-Area:          000000H-07FFFFH
Software-Protected-Areas #1: none
                        #2: none
                        #3: none
No-Access-Areas     #1: 00EF00H-00FFFFH
                        #2: none
                        #3: none

Enter Start Address <hex, min. 2 digits>:00000
Enter End   Address <hex, min. 2 digits>:10000

*** CAUTION ***
The smallest physical unit to be erased is predefined by the sector size
of the used Flash-Memory-devices. The following address range may differ
from your original input.

Address range to be erased: 000000H-020000H
START ERASURE? <Y/N>
```

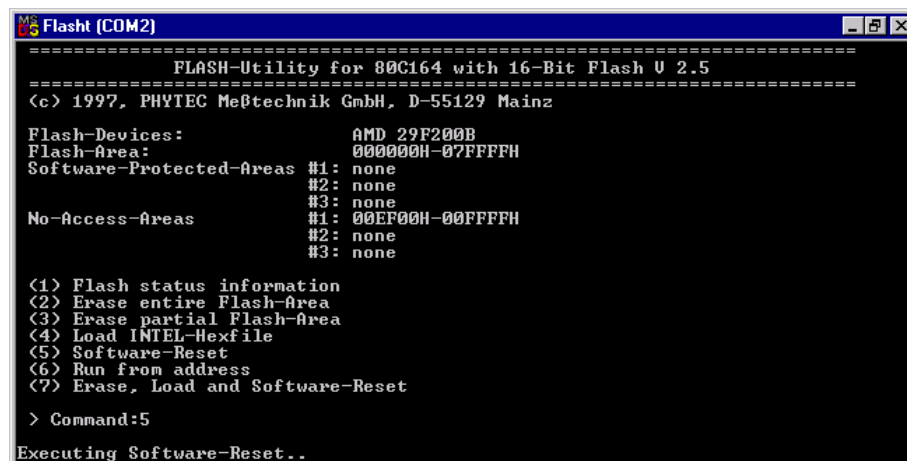
Load INTEL-hexfile downloads specified hexfiles to the nanoMODUL-164 on the Development Board:



```

Flasht (COM2)
Input File: c:\rapidkit\nm164\demos\blinky\blinky.hex
Press <F2> to download a HEX-file when using FLASHT
Waiting for INTEL-Hex File...
>>:020000040000FA
>>:02000400CB002F
>>:100006000FEB0EAE006E007BBFA08611870F0464F
>>:10001600F05726F4F04938528DF70FEAE006E0076C
>>:10002600BBEE08611870F046F05726F4F0493852D6
>>:040036008DF70DE74E
>>:10003A00A55AA5A50A863F1D1A8600121A8900FF2D
>>:10004A000A8_
  
```

Software-Reset forces a software-reset of the microcontroller, resulting in execution of the Flash memory content:



```

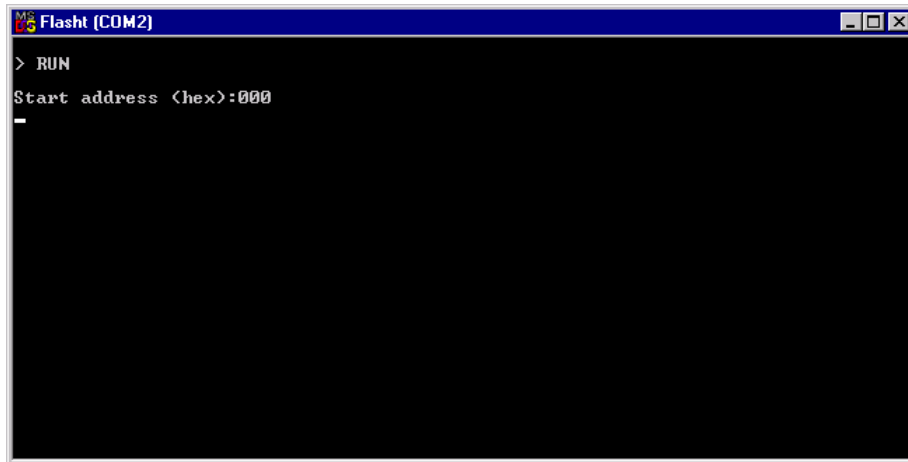
Flasht (COM2)
=====
FLASH-Utility for 80C164 with 16-Bit Flash U 2.5
=====
(c) 1997, PHYTEC Meßtechnik GmbH, D-55129 Mainz

Flash-Devices:          AMD 29F200B
Flash-Area:            000000H-07FFFFH
Software-Protected-Areas #1: none
                       #2: none
                       #3: none
No-Access-Areas       #1: 00EF00H-00FFFFH
                       #2: none
                       #3: none

<1> Flash status information
<2> Erase entire Flash-Area
<3> Erase partial Flash-Area
<4> Load INTEL-Hexfile
<5> Software-Reset
<6> Run from address
<7> Erase, Load and Software-Reset

> Command:5
Executing Software-Reset..
  
```

Run from address allows start of execution at a user specified address:



Erase, Load and Software-Reset results in automatic erasure of unprotected areas of the Flash, loading of a hexfile and a software-reset that executes the contents of the Flash memory.

Return to the main FlashTools menu to run an example program as described in the next section.

2.3.2.1 'Hello'

The 'Hello' example downloads a program to the Flash that, when executed, initialices the serial port to 9600 baud and sends a character string from the target hardware back to the host-PC. The character string can be viewed with a terminal emulation program.

- From the FlashTools main menu enter '2' to choose the *Erase entire Flash-Area* command and select 'Y' to confirm erasure of unprotected sectors of Flash memory. You can see the Flash sectors being erased in the FlashTools window. When all sectors are erased, the main menu reappears.
- Enter '4' to choose the *Load INTEL-hexfile* command and press function key <F2> to specify the input file for download.

- The hexfile has already been installed on your hard disk during the installation procedure. Enter the correct drive and path to the nanoMODUL-164 Demo file (default location *C:\PHYBasic\nM164\Demos\Tasking\Hello\hello.hex*). After pressing the <Return> key the download starts immediately and you can watch the process of the download as the code spills across the FlashTools window as it transmits via the RS-232 connection from the host to the on-board Flash:

```

Flasht (COM2)
Input File: c:\rapidkit\nm164\demos\hello\hello.h86
Press <F2> to download a HEX-file when using FLASHT
Waiting for INTEL-Hex File...
>>:020000040000FA
>>:10005400202063686172203D2025303262580A00F6
>>:0E0064001A48656C6C6F20576F726C640A004E
>>:1000100076E2000476E3000466E3FFF7E6B68000CC
>>:10002000E6B70000E65A4000E6D81180E6F8200066
>>:10003000DA004E06E6F86400E6F90000DA00500245
>>:10004000DA0_

```

- At the end of the download procedure the main menu reappears and the downloaded code can be executed.

If the selected Flash bank into which you wish to download code was not empty (i.e. erased), the following warning dialog box will appear: “!!! Target address location not empty : xxxx” (where xxxx is the erroneous address). If this occurs, press the <Space> bar to return to the main menu (this may take some time) and select the (2) *Erase entire Flash-Area* from the main menu to erase all unprotected sectors of the Flash. Then repeat the download procedure.

- From within the main menu press function key <F1>. Select ‘Y’ to exit FlashTools.

Monitoring the execution of the Hello demo requires use of a terminal program, such as HyperTerminal included with Windows.

- Start the HyperTerminal program within the *Windows Programs/Accessories* bar:
- The HyperTerminal main window will now appear:

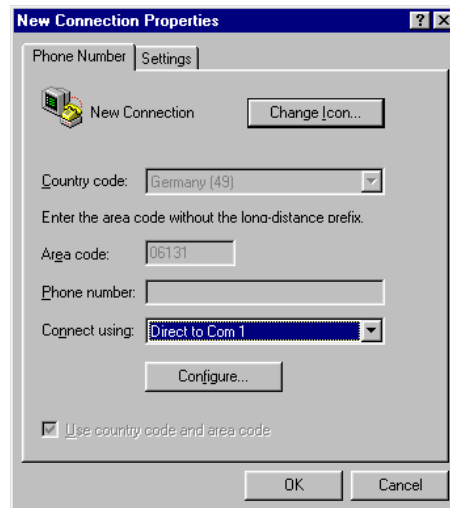


- Double-click on the *HyperTerminal* icon 'Hyperterm' to create a new HyperTerminal session.
- The Connection Description window will now appear. Enter "COM1 Direct" in the "Name" combobox (be sure to specify the correct COM parameter for your system).

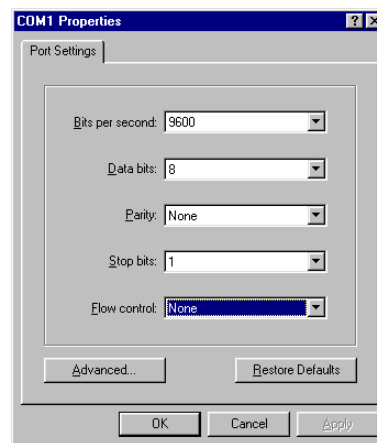


- Next press the *Ok* button. This creates a new HyperTerminal session named "COM1 Direct" and advances you to the next HyperTerminal window.

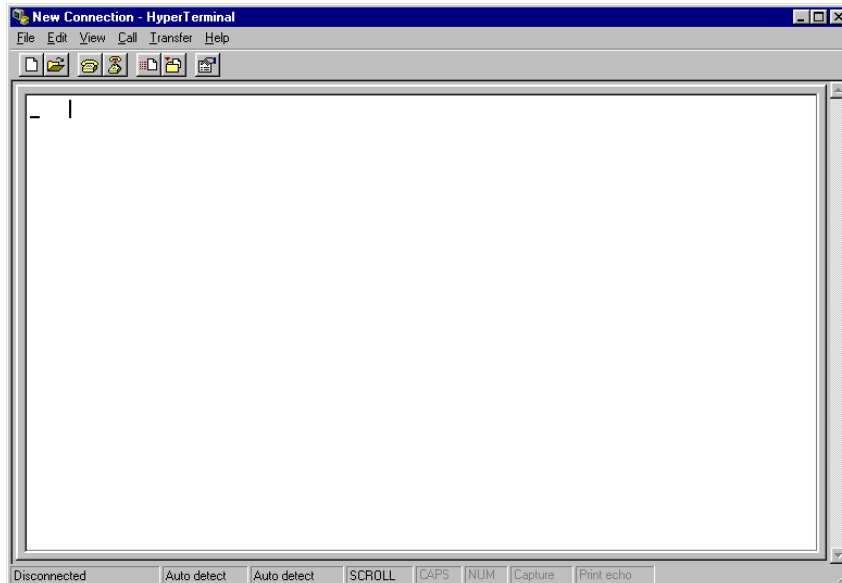
- The “New Connections Properties” window will now appear. Specify *Direct to COM1* under the *Connect Using* combobox (be sure to indicate the correct COM setting for your system).




- Select the *Configure* button in the “New Connections Properties” window to advance to the next window (COM1 Properties). Then set the following COM parameters: Bits per second = 9600; Data bits = 8; Parity = *None*; Stop Bits = 1; Flow Control = *None*.



- Selecting *OK* advances you to the “COM1 Direct – HyperTerminal” monitoring window. Notice the connection status report in the bottom left corner of the window.



- Resetting the Development Board (at S1) will execute the *hello.hex* file loaded into the Flash.
- Successful execution will send the character string "***Hello World***" from the target hardware to the HyperTerminal window.
- Click the disconnect icon 

If no output appears in the HyperTerminal window check the power supply, the COM parameters and the RS-232 connection.

The demo application within the file *hello.hex* initializes the serial port of your nanoMODUL-164 to 9600 baud. The initialization values are based on the assumption that the microcontroller runs at a 20 MHz internal clock frequency. Please note that the nanoMODUL-164 is populated with an oscillator with a frequency of just 5 MHz. An internal PLL (Phase Locked Loop) device puts an internal 20 MHz clock frequency. If your nanoMODUL-164 is equipped with an oscillator with a different frequency value, the demo application might transmit using another baud rate. This may lead to incoherent characters appearing in the HyperTerminal window following execution of code. Additional information can be found in the *readme.txt* file in the *Hello* project folder.

2.3.2.2 'Blinky'

The 'Blinky' example sends a program to the Flash that, when executed, manipulates the single user LED (D1) on the Development Board that is mounted near the Boot (S2) switch. This second example program provides a review of the FlashTools download procedure:

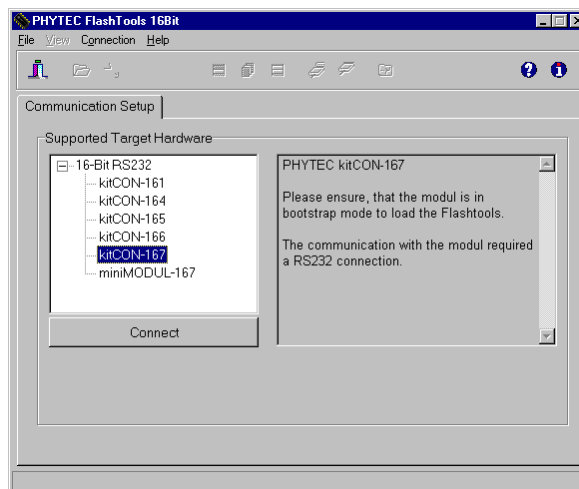
- Ensure that the target hardware is properly connected to the host-PC and a power supply
- Reset the target hardware and force it into Bootstrap mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, several seconds later, releasing the Boot (S2) switch
- Start FlashTools for DOS from within the *PHYTEC Rapid Development Kit/nanoMODUL-164* program group. For connection to the target hardware, remember to select the icon corresponding to the correct serial port on your host-PC
- After the loading process has finished and the main FlashTools menu appears, enter '2' to choose the *Erase entire Flash-Area* command. Select 'Y' to confirm erasure of unprotected sectors of Flash memory.

- Once the main menu reappears on the screen, enter '4' to choose the *Load INTEL-hexfile* command and press function key <F2> to specify the **blinky.hex** hex input file.
- The 'Blinky' hexfile has also already been installed on your hard disk during the installation procedure. Enter the correct drive and path to the demo file and press <Enter> (default location *C:\PHYBasic\nM164\Demos\Tasking\Blinky\blinky.hex*).
- Returning to the main menu, enter '5' to choose the *Software-Reset* command. This will render a software-reset to the target system and start execution of the download program.
- Successful execution of the program will flash the LED with equal on and off durations.
- Press the function key <F1> to exit FlashTools.

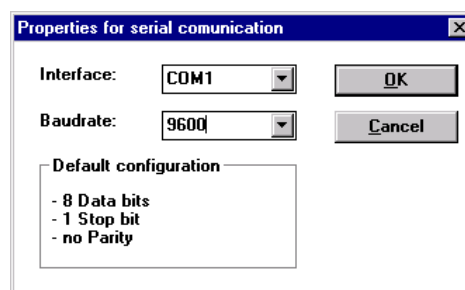
You have now successfully downloaded and executed two pre-existing example programs in **.hex*-file format.

2.3.3 Downloading Example Code with FlashTools for Windows

- Start FlashTools for Windows by double-clicking on the FlashTools for Windows icon or by selecting *FlashTools for Windows* from within the *Programs|PHYTEC FlashTools for Windows* program group.
- The Connect tab of the FlashTools for Windows Worksheet window will now appear.
- Double click on *16-Bit RS232*

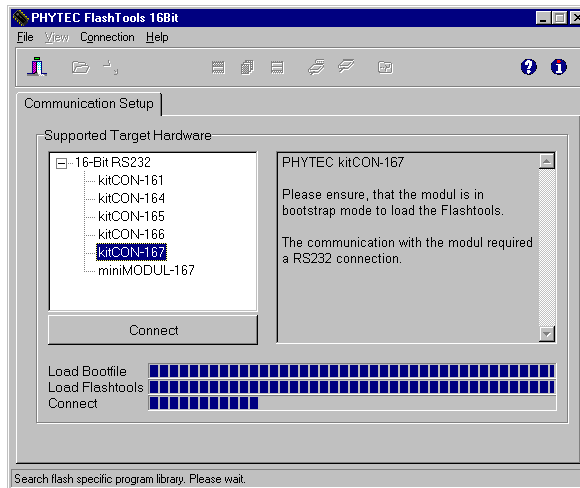


- Select the correct module from the list and press the *Connect* button.



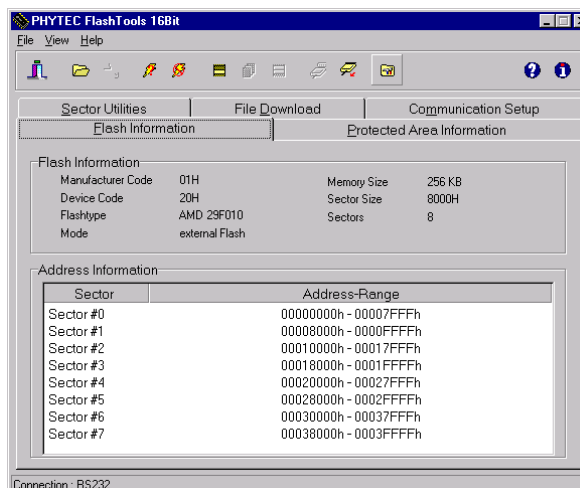
- Select the correct serial port for your host-PC and a 9600 baudrate.
- Click the *OK* button to load the module based part of the Flashtools to the target hardware.

The microcontroller tries to automatically adjust to the baud rate selected within the baud rate tab. However it may occur that the selected baud rate can not be reached. This results in a connection error. In this case, try other baud rates to establish a connection. Before further connection attempts, be sure to reset the target hardware and render it into Bootstrap mode as described in *section 2.2*.

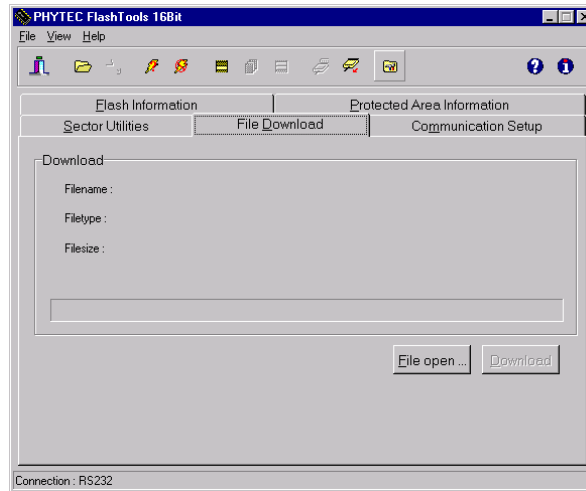


- After the data transfer you will see FlashTools for Windows Worksheet window with the following tabs.

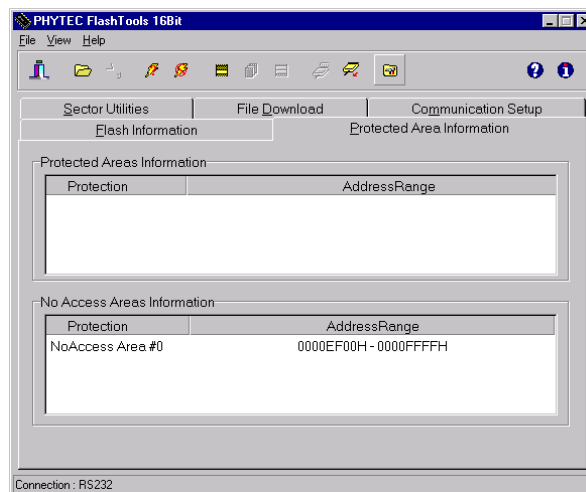
Flash Information shows sector and address ranges in Flash memory:



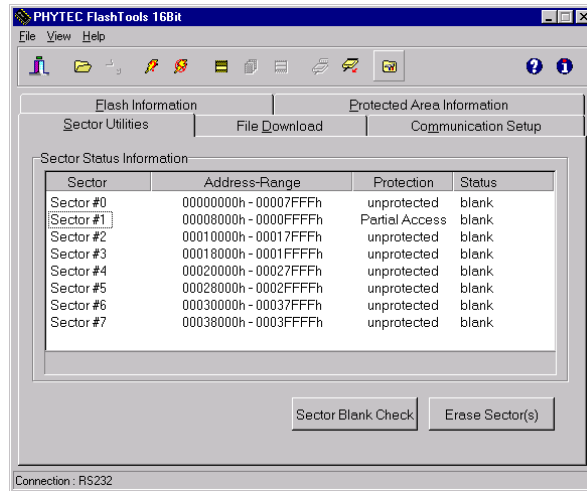
File Download downloads specified hexfiles to the target hardware:



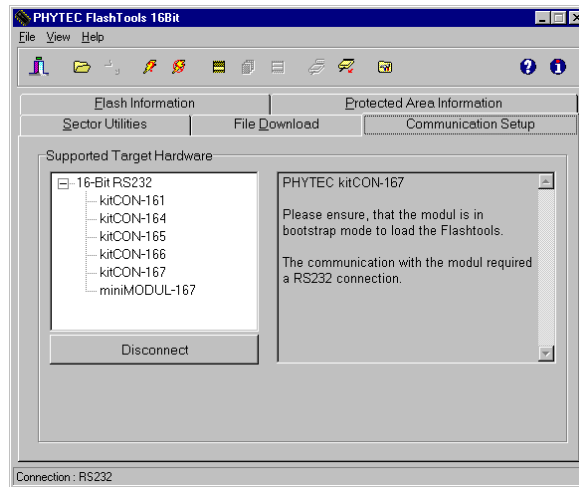
Protected Areas Information shows protected areas of Flash memory:



Sector Utilities allow erasure of individual sectors of Flash memory:



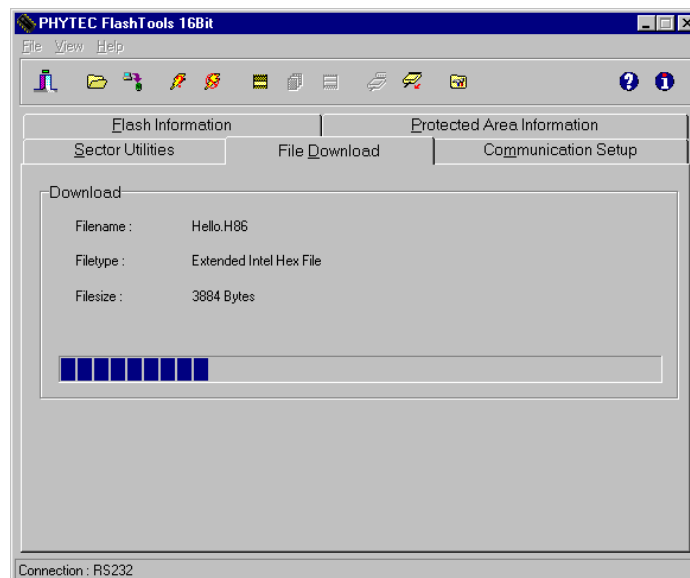
Communication Setup provides you with the possibility to disconnect the target and reconnect with an equal one. Programming of several targets is possible for you thereby.



2.3.3.1 'Hello'

The "Hello" example downloads a program to the Flash that, when executed, performs an automatic baud rate detection and sends a character string from the target hardware back to the host-PC. The character string can be viewed with a Terminal Emulation program.

- Choose the *Sector Utilities* tab, highlight *Sector #0* and click on the *Erase Sector(s)* button. You can see the Flash sector(s) being erased at the bottom left hand of the window. When the desired sectors are erased, the connection properties description returns to the lower left corner of the window.
- Choose the *File Download* tab, and click on the *File Open...* button.
- The hexfile has already been installed to your hard-drive during the installation procedure. Type in the correct drive and path to the nanoMODUL-164 Demo directory (default location *C:\PHYBasic\nM164\Demos\Tasking\Hello\hello.hex*) and click *Open*.
- Click on the *Download* button. You can watch the status of the download of the *hello.hex* into external Flash memory in the Download window.



- At the end of the download, a sector by sector status check of the Flash memory can be viewed in the left-hand corner of the FlashTools for Windows Worksheet window. Wait until the status check finishes before returning to work with the board. Once the status check is complete, the downloaded code can be executed.

If the selected Flash bank into which you wish to download code is not empty (i.e. erased), a warning dialog box will appear, indicating “Location not empty! Please erase location and try again.” In this event, select the *Sector Erase* tab from the FlashTools for Windows worksheet, highlight Sector #0 and erase the sector. Then repeat the download procedure.

- Returning to the *Communication Setup* tab, click on the *Disconnect* button and exit the FlashTools.

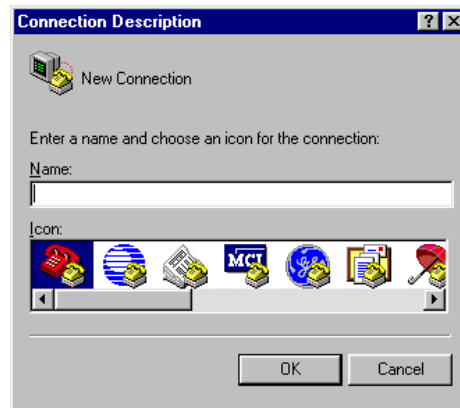
Monitoring the execution of the Hello demo requires use of a terminal program, such as HyperTerminal included with Windows.

- Start the HyperTerminal program within the *Windows Programs/Accessories* bar:
- The HyperTerminal main window will now appear:

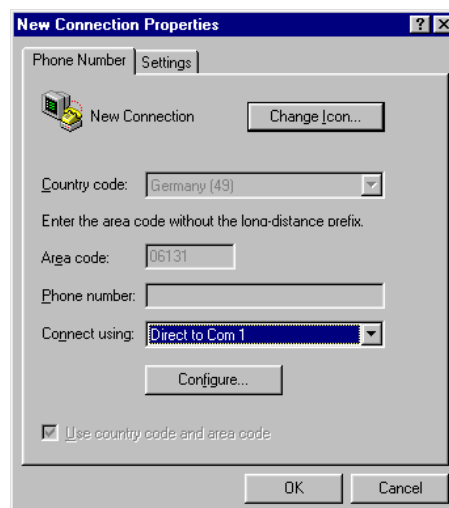


- Double-click on the *HyperTerminal* icon ‘Hyperterm’ to create a new HyperTerminal session.

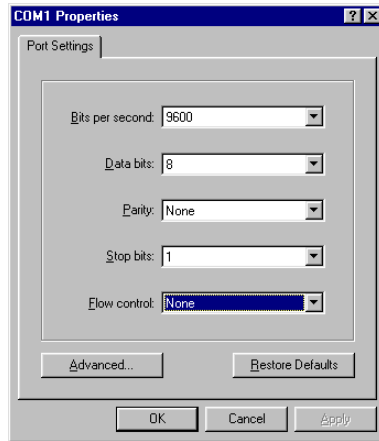
- The Connection Description window will now appear. Enter “COM1 Direct” in the “Name” combobox (be sure to specify the correct COM parameter for your system).



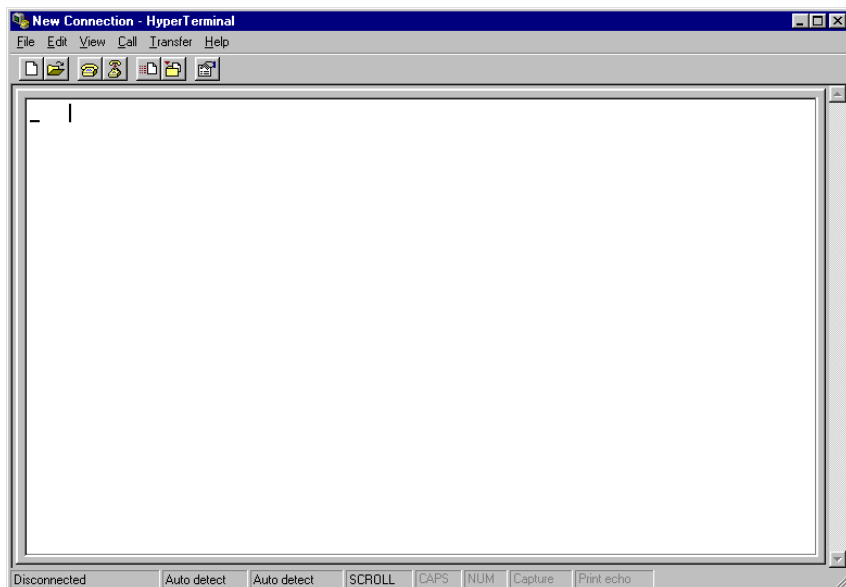
- Next press the *Ok* button. This creates a new HyperTerminal session named “COM1 Direct” and advances you to the next HyperTerminal window.
- The “New Connections Properties” window will now appear. Specify *Direct to COM1* under the *Connect Using* combobox (be sure to indicate the correct COM setting for your system).




- Select the *Configure* button in the “New Connections Properties” window to advance to the next window (COM1 Properties). Then set the following COM parameters: Bits per second = 9600; Data bits = 8; Parity = *None*; Stop Bits = *1*; Flow Control = *None*.



- Selecting *OK* advances you to the “COM1 Direct – HyperTerminal” monitoring window. Notice the connection status report in the bottom left corner of the window.



- Resetting the Development Board (at S1) will execute the *hello.hex* file loaded into the Flash.
- Successful execution will send the character string "**Hello World**" from the target hardware to the HyperTerminal window.
- Click the disconnect icon 

If no output appears in the HyperTerminal window check the power supply, the COM parameters and the RS-232 connection.

The demo application within the file *hello.hex* initializes the serial port of your nanoMODUL-164 to 9600 baud. The initialization values are based on the assumption that the microcontroller runs at a 20 MHz internal clock frequency. Please note that the nanoMODUL-164 is populated with an oscillator with a frequency of just 5 MHz. An internal PLL (Phase Locked Loop) device puts an internal 20 MHz clock frequency. If your nanoMODUL-164 is equipped with an oscillator with a different frequency value, the demo application might transmit using another baud rate. This may lead to incoherent characters appearing in the HyperTerminal window following execution of code. Additional information can be found in the *readme.txt* file in the *Hello* project folder.

2.3.3.2 ‘Blinky’

The “Blinky” example sends a program to the Flash that, when executed, manipulates the single user LED D3 on the Development Board that is mounted near the Boot (S2) switch. This second example program provides a review of the FlashTools download procedure:

- Ensure that the target hardware is properly connected to the host-PC and a power supply.
- Reset the target hardware and force it into Bootstrap mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, two or three seconds later, release the Boot (S2) switch.
- Start FlashTools for Windows from within the *PHYTEC Rapid Development Kit/nanoMODUL-164* program group. Remember to select the icon corresponding to the proper serial port on your host-PC for connection to the target hardware.
- After the loading process has finished and the main FlashTools menu appears, enter ‘2’ to choose the *Erase entire Flash-Area* command. Select ‘Y’ to confirm erasure of unprotected sectors of Flash memory.
- Once the main menu reappears on the screen enter ‘4’ to choose the *Load INTEL-hexfile* command and press the function key <F2> to specify the *blinky.hex* hex input file.
- The “Blinky” hexfile has also already been installed on your hard-drive during the installation procedure. Enter the correct drive and path to the demo file and press <Enter> (default location *C:\PHYBasic\nM164\Demos\Tasking\Blinky\blinky.hex*).
- Returning to the main menu, enter ‘5’ to choose the *Software-Reset* command. This will render a software-reset to the target system and start execution of the downloaded program.
- Successful execution of the program will flash the LED with equal on and off durations.
- Press the function key <F1> to exit the FlashTools.

You have now successfully downloaded and executed two pre-existing example programs in **.hex* hexfile format.

3 Getting More Involved

What you will learn with this example:

- how to start the TASKING software tool chain
- how to configure the TASKING tools within the C166/ST10 Embedded Development Environment (EDE)
- how to modify the source code from our examples, create a new project and build and download an output *.*hex*-file to the target hardware

3.1 Starting the TASKING Software Tool Chain

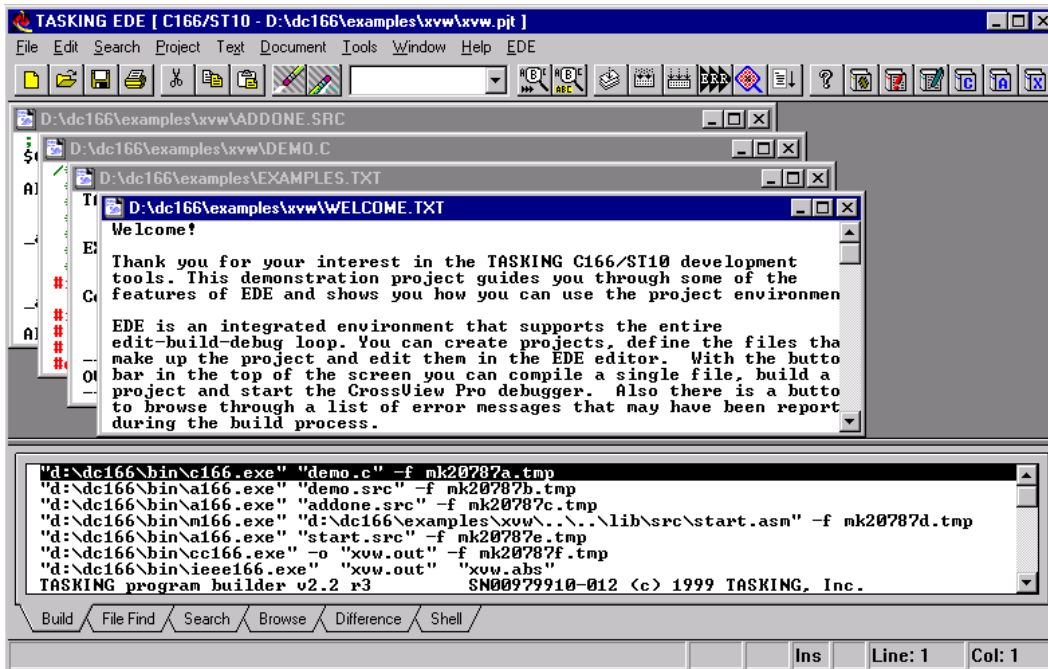
The TASKING C166/ST10 Embedded Development Environment (EDE) evaluation software should have been installed during the install procedure, as described in *section 2.1*.

You can also manually install the tool chain by executing *setup.exe* from within the `\Software\Tasking\DC166` folder of your PHYTEC Spectrum CD.

Start the tool chain by selecting *Embedded Development Environment* from within the *Programs/TASKING C166_ST10 Demo* program group.

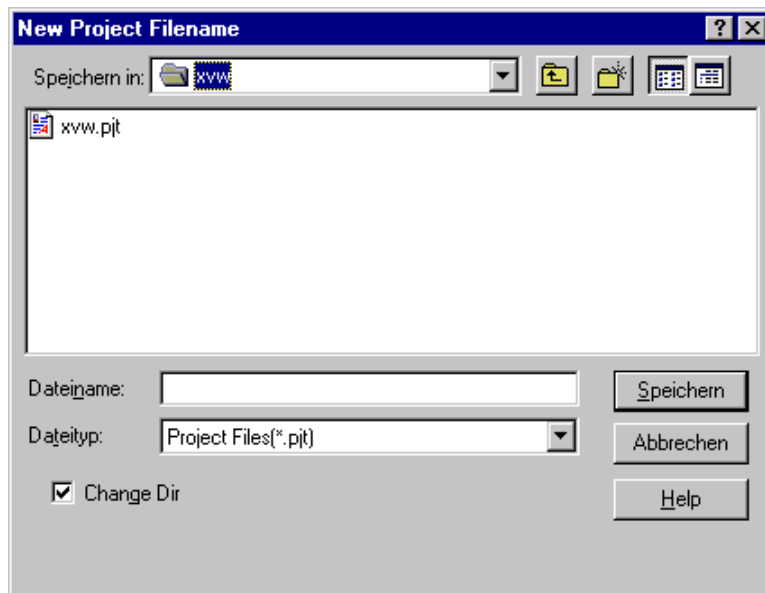
nanoMODUL-164 QuickStart Instructions

After you start EDE, the window shown below appears. From this window you can create projects, edit files, configure tools, compile, assemble, link and start the debugger. Other 3rd party tools such as emulators can also be started from here.

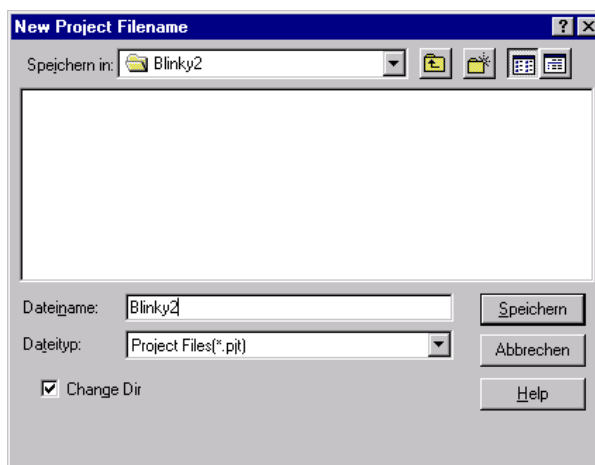


3.2 Creating a New Project and Adding an Existing Source File

- Open the *Project* menu and choose *New* and press the Browse... button. The window shown below appears.

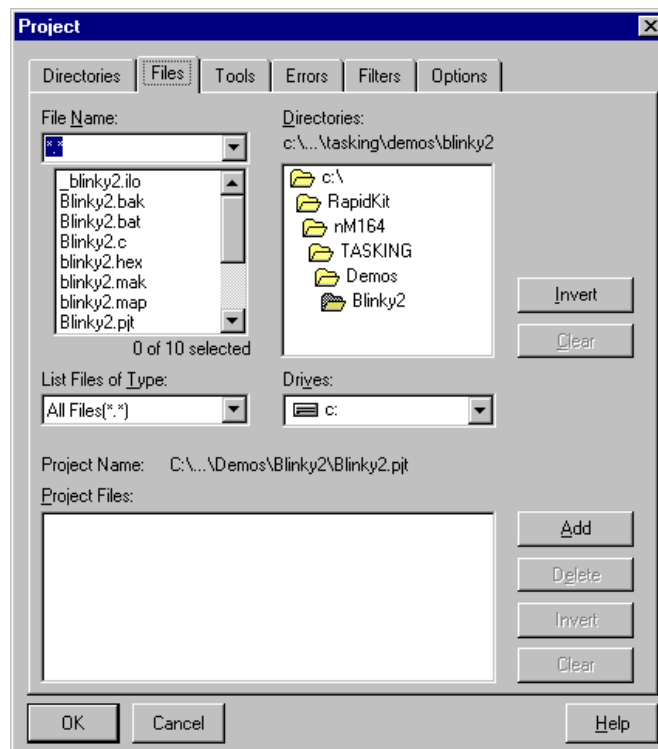


- In the 'Save in' section at the top of the window change to the project folder created by the installation procedure (default location *C:\PHYBasic\nM164\Demos\Tasking\Blinky2*).



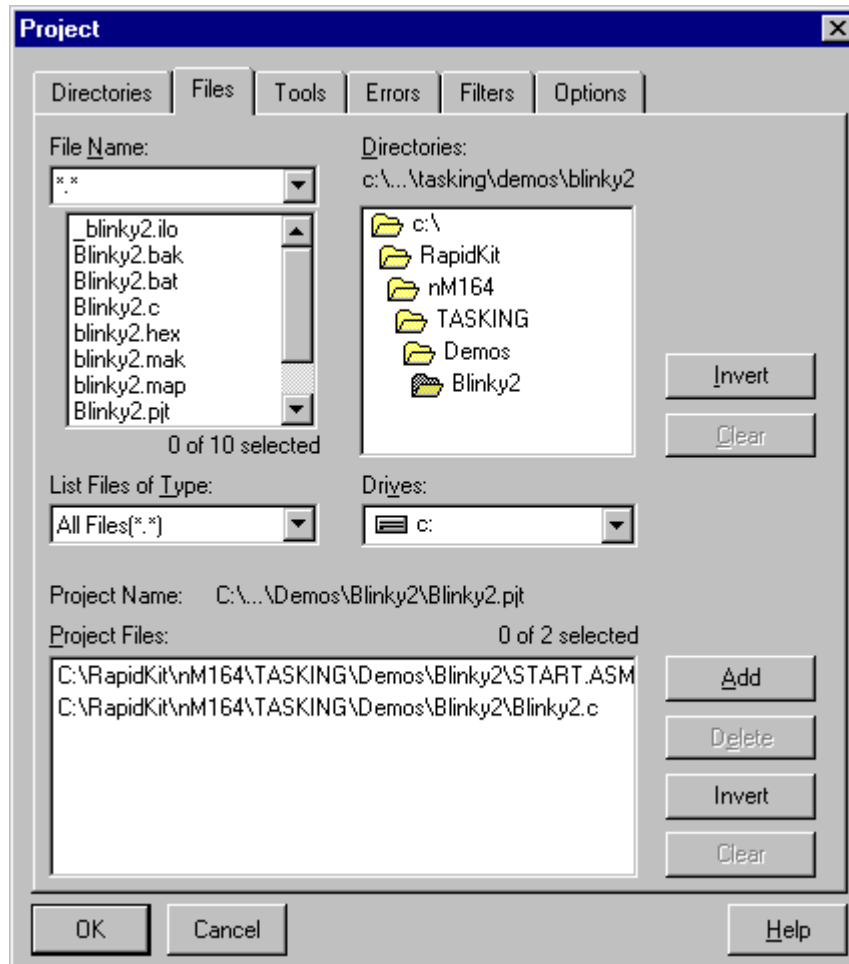
- In the 'File name', enter the file name of the project you are creating. For this tutorial, enter the name *Blinky2* and press *OK*

- The following window will appear:



This window is used to add various files to your project. These include ASCII files, C/C++ and assembly sourcefiles, assembler generated object files and library files. Note that the Project Files box is blanked out at this time. This project window is accessible at any time by selecting *Project/Properties|Files*, which enables easy edit of your file list.

- Double clicking on the files which should be part of the project (in this example *start.asm* and *Blinky2.c*) results in the following window.

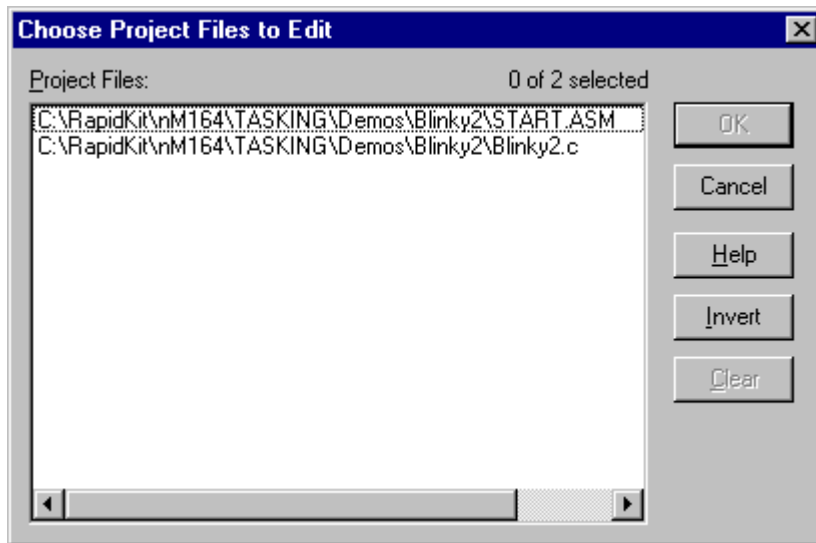


- Choose *OK* to save the new project.

At this point you have created a project called *blink2.pjt* and added an existing C source file called *blink2.c* and an existing Assembler file called *start.asm*. The next step is to modify the C source before building your project. This includes compiling, linking, locating and creating the hexfile.

3.3 Modifying the Source Code

- Open the *Project* menu and choose *Load Files*. The window shown below appears. This is where all project files are displayed. You can double click on the filename you would like to edit, or just highlight it and select *OK*. If you would like to load all project files into the editor you can select *Invert* and *OK*.




- Double click on *Blinky2.c* to load it into the editor. Locate the following code section. Modify the section shown below (the values shown in bold and italic) from the original (150,000) counts to the indicated values:

```
while (1) {                                     /* loop forever          */
    P2_8 = 0;                                    /* output to LED port     */
    for (i=0; i<225000; i++) {                 /* delay for 150000 counts */
        wait ();                                /* call wait function     */
    }
    P2_8 = 1;                                    /* output to LED port     */
    for (i=0; i<75000; i++) {                 /* delay for 150000 counts */
        wait ();                                /* call wait function     */
    }
}
```

This will change the on/off ratio of the *blinky* program.

3.4 Saving the Modifications

- Save the modified file by choosing *File/Save* or by clicking the floppy  disk icon.

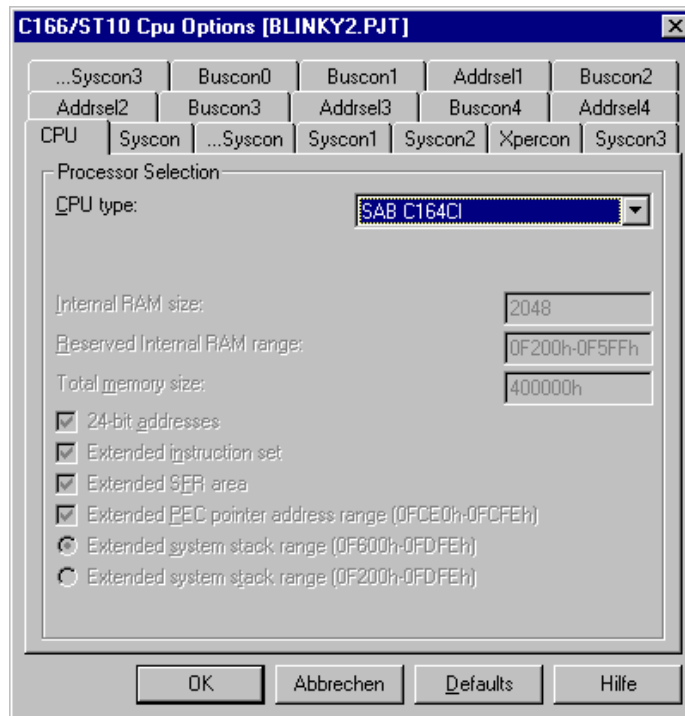
Note that the icon is active as soon as you modify the file.

3.5 Setting Tool Chain Options

TASKING includes a Make utility that can control compiling and linking source files in several programming languages. Before using the macro preprocessor, assembler, C compiler or linker/locator you must configure the corresponding options. Enter the changes as indicated below and leave all other options set to their default values. EDE allows you to set various options with mouse clicks and these are all saved in your project file.

To configure the CPU:

- Open the *EDE|CPU Options* menu and choose *CPU*. Adjust the default settings by clicking *Defaults*. In *CPU type* select *SAB C164CI*. Press *OK* to close the window after the modification.



To configure the C166 Compiler:

No modifications are necessary when default settings are set.

To configure the M166 Macro Preprocessor:

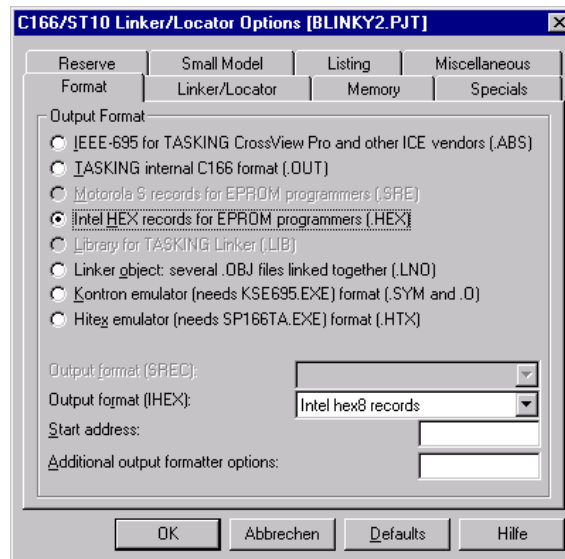
No modifications are necessary when default settings are set.

To configure the A166 Assembler:

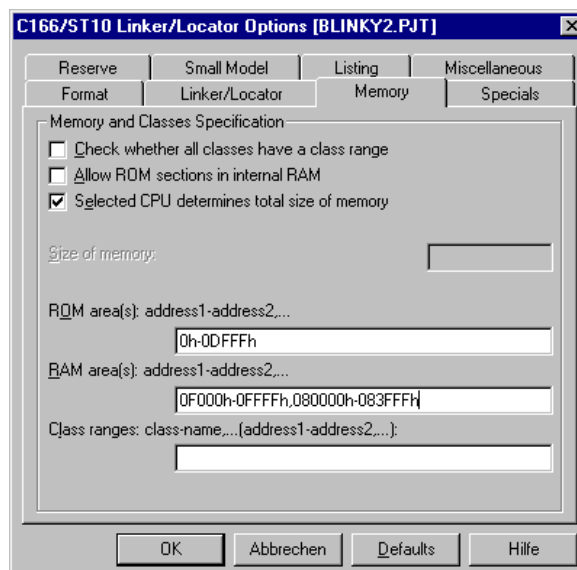
No modifications are necessary when default settings are set.

To configure the L166 Linker/Locator:

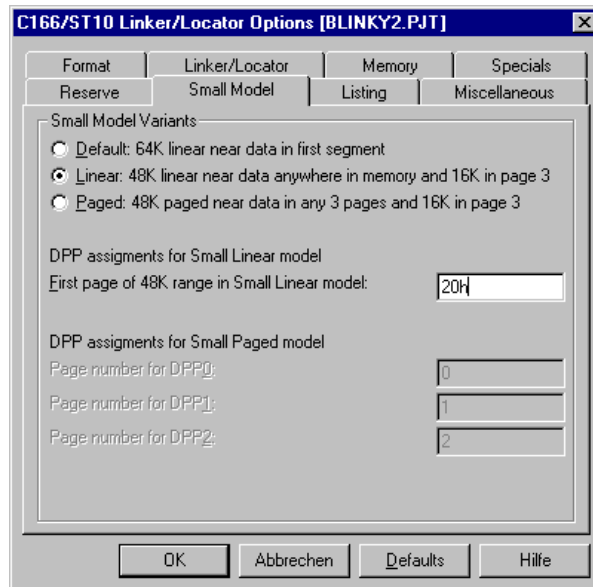
- Open the *EDE* menu and choose *Linker/Locator Options/Format*. Adjust the default settings by clicking *Default*. Select the *Intel Hex records for EPROM programmers (.HEX)* radiobutton. **Attention:** do not enter any value in *Start address*: since the flashtools will not accept this!



- Select the *EDE/Linker/Locator Options/Memory* and specify the following settings




- Enter *EDE/Linker/Locator Options/Small Model* Menu select *Linear: 48K linear near data anywhere in memory and 16K in page 3* and specify *First page of 48K range in Small Linear model: 020h*



The Linker options are now suitable for the *Blinky2* project, enabling you to build an absolute object file without taking into account debugging settings. The options lead to the generation of a **.map* map file with included memory map.

3.6 Building the Project

You are now ready to run the compiler and linker using the Make utility.

- Click on the *Execute 'Rebuild' Command* icon  from the EDE tool bar or open *Project* and select *Rebuild*.
- If the program specified (*blinky2.c*) contains any errors, they will be shown in the Build Output window on the screen.
- If there are no errors, the code is assembled and linked and the executable code is ready to be downloaded to the board. *No errors found.* is shown in the bottom line. The code to be downloaded to the board will be the name of the project with *.hex* as filename extension (in this case *blinky2.hex*).
- Note that a machine-readable, executable hexfile has been created. Other files (e.g. assembler input files **.src*, list files **.lst* and map file **.map*) are created to help the debugging or troubleshooting and error searching process.
- If a list of errors appears, use the editor to correct the error(s) in the source code and save the file and repeat this section.

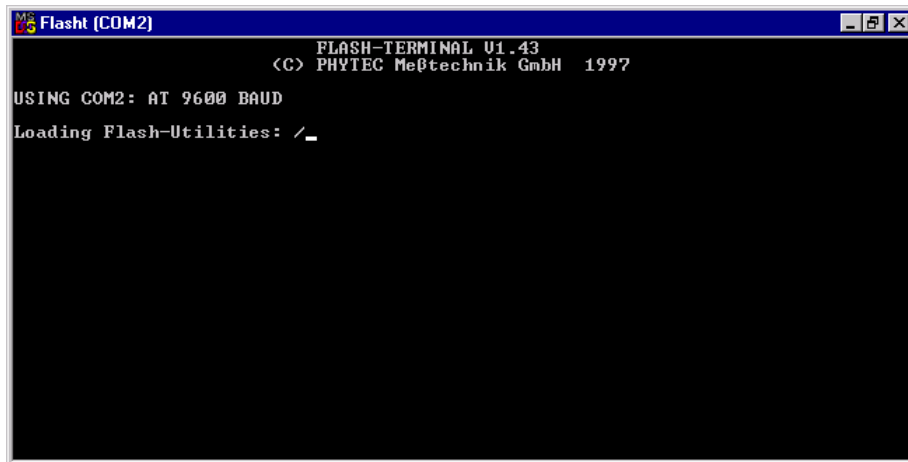
3.7 Downloading the Output File

3.7.1 Downloading with FlashTools for DOS

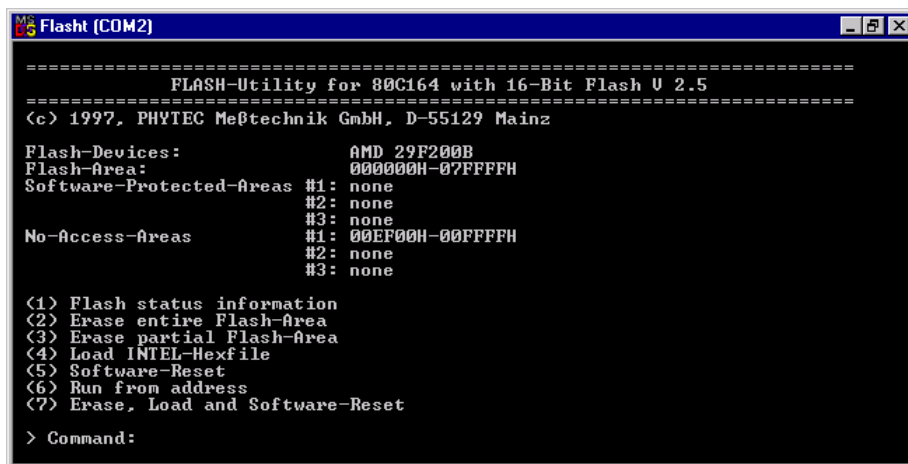
- Exit TASKING EDE environment.
- Reset the target hardware and force it into Flash programming mode by simultaneously pressing the Reset (*S1*) and Boot (*S2*) switches on the Development Board, releasing first the Reset (*S1*) and then the Boot (*S2*) switch
- Start FlashTools for DOS for your nanoMODUL-164 by selecting either the *FlashTools (COM1)* or *FlashTools (COM2)* icon within the *PHYTEC Rapid Development Kit/nanoMODUL-164* program group. Which icon to choose depends on the serial port used to connect to the target system.

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash use the icon's *FlashTools Max (COM1)* or *FlashTools Max (COM2)*.

- FlashTools will be loaded to the RAM of the nanoMODUL-164. The active loading process is indicated by a rotating cursor bar.



- After the loading process has finished the main menu appears on the screen.



- Enter '2' to choose the *Erase entire Flash-Area* command and enter 'Y' to confirm deletion of all unprotected sectors of Flash memory. You can see the Flash sectors being erased in the FlashTool window. When all sectors are erased, the main menu reappears.
- Enter '4' to choose the *Load INTEL-hexfile* command and press function key <F2> to specify the input file.

- The hexfile for download was created during the *Make* process and is located in the folder of the *Blinky2* project. Enter the correct drive and path to the *Blinky2* hexfile (default location *C:\PHYBasic\nM164\Demos\Tasking\Blinky2\blinky2.hex*) The download immediately starts. You can watch the process of the download in the FlashTools window as code transmits via the RS-232 connection from the host-PC to the on-board Flash:

```

Flasht (COM2)
Input File: c:\rapidkit\products\modul\nm164\demom\blinky2\blinky2.h86
Press <F2> to download a HEX-file when using FLASHT
Waiting for INTEL-Hex File...
>>:020000040000FA
>>:02000400C002F
>>:10000600FEB0E0A006E007BBFA08611870F0464F
>>:10001600F05726F4E386E38538DF70F0A006E0074E
>>:10002600BEE08611870F046F05726F4F8243851F4
>>:040036008DF70E74E
>>:10003A00055A05A050063F1D1A0600121A0900FF2D
>>:10004A000A890404E60A0CF0E6000000E601000048
>>:10005A000E6022000E60800FCB54A85B5E6F00090D5
>>:10006A000E6F90000E6F80000F01870192D29E005FD
>>:10007A00DC0998282D25F032DC099848AAF21BE001
>>:10008A00AF20EF0F0347C330EF306F300FDE0150D
>>:10009A004C549150685BB853084128213DF30DE751
>>:1000AA00DC099838DC44B9A30831184076F300C

```

If the selected Flash bank into which you wish to download code was not empty (i.e. erased), a warning dialog box will appear, indicating “!!! Target address location not empty : xxxx” where xxxx is the erroneous address. In this case, press the <Space> bar to return to the main menu (this may take some time) and select the (2) *Erase entire Flash-Area* from the main menu to erase all unprotected sectors of the Flash memory. Then repeat the download procedure.

- After the download is finished, FlashTools will return to the main menu automatically.
- Press function key <F1> and select ‘Y’ to confirm that you wish to exit FlashTools.
- Press the Reset button (S1) on the Development Board
- If the modified hexfile properly downloads and executes, the LED should now flash in a different mode with different on and off durations.

You have now modified source code, recompiled the code, created a modified download hexfile, and successfully executed this modified code.

3.7.2 Downloading with FlashTools for Windows

- Exit TASKING EDE environment.
- Render the target hardware and force it into Flash programming mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board, releasing first the Reset (S1) and two or three seconds later, releasing the Boot (S2) switch
- Start FlashTools for Windows
- At the Serial Interface tab of the FlashTools for Windows Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Load Flashtools* button to transfer the modul based part of the FlashTools for Windows to the target hardware.
- Select the right protocol (default RS232) for further use of the FlashTools for Windows.
- Returning to the FlashTools for Windows Worksheet, choose the *Sectors Utilities* tab, highlight *Sector #0* in the *Sectors Erase* section of the tab and click on the *Erase Sector(s)* button to erase this memory sector
- Wait until the status check in the lower left corner of the FlashTools for Windows Worksheet finishes, returning the connection properties description to the lower left corner of the window
- Next choose the *File Download* tab and click on the *File Open* button
- Type the complete pathway and name of the file you wish to download
C:\PHYBasic\nM164\Demos\Tasking\Blinky2\blinky2.hex
directory (default location)
- Click on the *Download* button and view the download procedure to the board in the status window
- Returning to the *Communication* tab, click on the *Disconnect* button
- Press the Reset button (S1) on the Development Board.

- If the modified hexfile properly downloads and executes, the LED should now flash in a different mode with different on and off durations.

You have now modified source code, recompiled the code, created a modified download hexfile, and successfully executed this modified code.

3.8 ‘Hello’

A return to the “Hello” program allows a review of how to modify source code, create and build a new project, and download the resulting output file from the host-PC to the target hardware. For detailed commentary on each step, described below in concise form, refer back to the “Blinky” example starting at section 3.1.

3.8.1 Creating a New Project

- Start the TASKING EDE environment and close all projects that might be open.
- Open the Project menu and create a new project called *hello2.pjt* within the existing project folder
C:\PHYBasic\nM164\Demos\Tasking\Hello2
(default location) on your hard drive.
- Add *hello2.c*, *serio.c* and *start.asm* from within the project folder to the project *hello2.pjt*.

At this point you have created a project called *hello2.pjt* consisting of C source files called *hello2.c*, *serio.c* and an assembler file called *start.asm*.

3.8.2 Modifying the Example Source

- Open the project window.
- Open the file *hello2.c* from within the project window.
- Use the editor to modify the *printf* command:

```
printf ("\x1AHello World\n")
```

to

```
printf ("\x1APHYTEC... Stick It In!\n")
```

- Save the modified file under the same name *hello2.c*.

3.8.3 Setting Tool Chain Options

The tool chain options can be set similar to the settings for the Blinky2 project described in *chapter 3.5*.

3.8.4 Building the New Project

- Build the project
- If any source file of the project contains any errors, they will be shown in an error dialog box on the screen. Use the editor to correct the error(s) in the source code and save the file and repeat this section.
- If there are no errors, the code is assembled and linked and the executable code is ready to be downloaded to the board.

3.8.5 Downloading the Output File

3.8.5.1 Downloading with FlashTools for DOS

- Reset the target hardware to put it in Flash Programming mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, two or three seconds later, release the Boot (S2) switch
- Start FlashTools for DOS for your nanoMODUL-164 by selecting either the *FlashTools (COM1)* or *FlashTools (COM2)* icon within the *PHYTEC Rapid Development Kit/nanoMODUL-164* program group. Choose the icon that corresponds to the serial port used on your host-PC to connect to the target system.

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash use the icon's *FlashTools Max (COM1)* or *FlashTools Max (COM2)*.

- FlashTools will load to the RAM of the nanoMODUL-164. The active loading process is indicated by a rotating cursor bar.
- After FlashTools has loaded, the main FlashTools menu appears on the screen.
- Enter '2' to choose the *Erase entire Flash-Area* command and enter 'Y' to confirm erasure of all unprotected sectors of Flash memory. You can see the Flash sectors being erased in the FlashTools window. When the all sectors are erased, the main FlashTools menu reappears.
- Enter '4' to choose the *Load INTEL-hexfile* command and press the function key <F2> to specify the input file.
- The hexfile for download was created during the Make process and is located in the folder of the **Hello2** project. Enter the correct drive and path to the **Hello2** hexfile (default location *C:\PHYBasic\nM164\Demos\Tasking\Hello2\Hello2.hex*) and press <Enter>. The download immediately starts. You can watch the download in the FlashTools window as code transmits via the RS-232 connection from the host-PC to the on-board Flash.

- At the end of the download the main menu reappears and the downloaded code can be executed.


If the selected Flash bank into which you wish to download code was not empty (i.e. erased), a warning dialog box will appear, indicating “!!! Target address location not empty : *xxxx*” where *xxxx* is the erroneous address. In this case, press the <Space> bar to return to the main menu (this may take some time) and select the (2) *Erase entire Flash-Area* from the main menu to erase all unprotected sectors of the Flash memory. Then repeat the download procedure.

- After the download is finished, FlashTools will return to the main menu automatically
- Press function key <F1> and select ‘Y’ to confirm that you wish to exit FlashTools

3.8.5.2 Downloading with FlashTools for Windows

- Exit TASKING EDE environment
- Reset the target hardware to render it in Flash Programming mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, two or three seconds later, release the Boot (S2) switch
- Start FlashTools for Windows
- At the Serial Interface tab of the FlashTools for Windows Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Load Flashtools* button to transfer the modul based part of the FlashTools for Windows to the target hardware
- Select the right protocol (default RS232) for further use of the FlashTools for Windows
- Returning to the FlashTools for Windows Worksheet, choose the *Sectors Utilities* tab, highlight *Sector #0* in the *Sectors Erase* section of the tab and click on the *Erase Sector(s)* button to erase this memory sector
- Wait until the status check in the lower left corner of the FlashTools for Windows Worksheet finishes, returning the connection properties description to the lower left corner of the window
- Next choose the *File Download* tab and click on the *File Open* button
- Type the complete pathway and name of the file you wish to download
C:\PHYBasic\nM164\Demos\Tasking\Hello2\Hello2.hex
directory (default location)
- Click on the *Download* button and view the download procedure to the board in the status window
- Returning to the *Communication* tab, click on the *Disconnect* button
- Exit the FlashTools for Windows.

3.8.6 Starting the Terminal Emulation Program

- Start the HyperTerminal and connect it to the target hardware using the following COM parameters: Bits per second = *9600*; Data bits = *8*; Parity = *None*; Stop Bits = *1*; Flow Control = *None*
- Resetting the Development Board (at S1) will execute the *hello2.hex* file loaded into the Flash.
- Successful execution will send the modified character string **‘PHYTEC...Stick it in!’** to the HyperTerminal window.
- Click the Disconnection icon 
- Close the HyperTerminal program

You have now modified source code, recompiled the code, created a modified download hexfile, and successfully executed this modified code.

4 Debugging

The TASKING debugger has two possible modes:

- the CrossView Pro simulator debugger environment is a software debugger and simulator
- the CrossView Pro ROM monitor debugger communicates with the target hardware via a monitor kernel that is downloaded to the RAM on the target system

This example utilizes the CrossView Pro ROM monitor debugger debug environment, which automatically downloads a special monitor kernel to the target hardware using the Bootstrap mode.

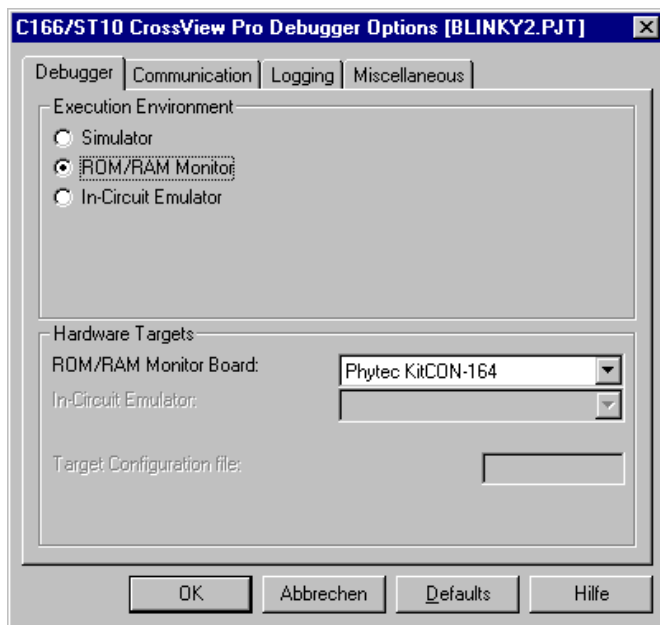
4.1 Creating a Debug Project and Preparing the Debugger

Create a new project in *C:\PHYBasic\nM164\Demos\Tasking\xvw* folder named *xvw*. Add the files *Start.asm*, *Addone.src* and *Demo.c* to the project and load *Demo.c* in the editor (*Project/Load Files...*).

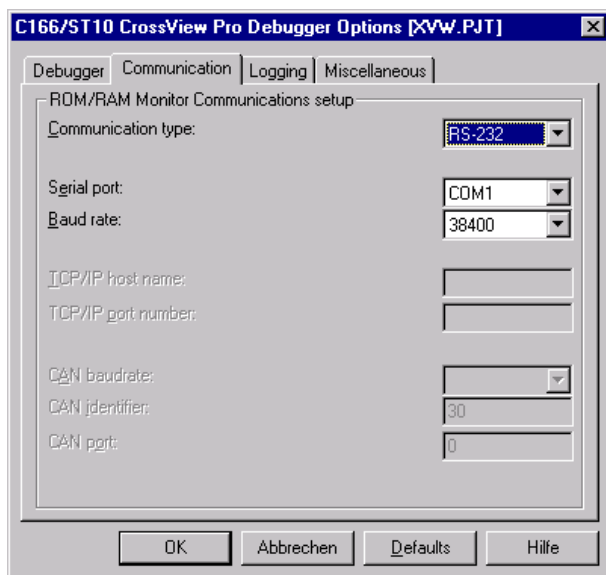
To be able to debug an application on target hardware the complete application (code and data) needs to be loaded into the target hardware's RAM. Therefore the application needs to be rebuilt with different settings compared to the stand-alone running version. Compared to the Blinky2 example settings (*see chapter 3.5*) the changes are listed below.

- *EDE/Linker/Locator Options/Format* select *IEEE-695 for TASKING CrossView Pro and other ICE vendors (.ABS)*
- *EDE/Linker/Locator Options/Memory*: No entries necessary in ROM/RAM areas
- *EDE/Linker/Locator Options/Small Model* select *Default: 64K linear near data in first segment*
- Rebuild the project

Since the required files for the download are specific to one specific target hardware the appropriate selection has to be made in *EDE/CrossView Pro Options/Debugger*. It has to be set to *ROM/RAM Monitor*. After this the target hardware can be selected in the *ROM/RAM Monitor Board* menu. In this case: Phytec kitCON-164 since the settings are compatible to the nanoMODUL-164.



The baudrate plus COM port can be set in *EDE/CrossView Pro Options/Communication* menu.



4.2 Preparing the Target Hardware to Communicate with CrossView Pro

- Set the Jumpers on the Development Board as indicated below.

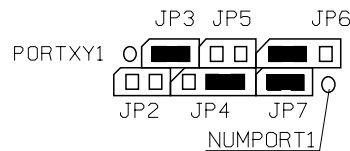



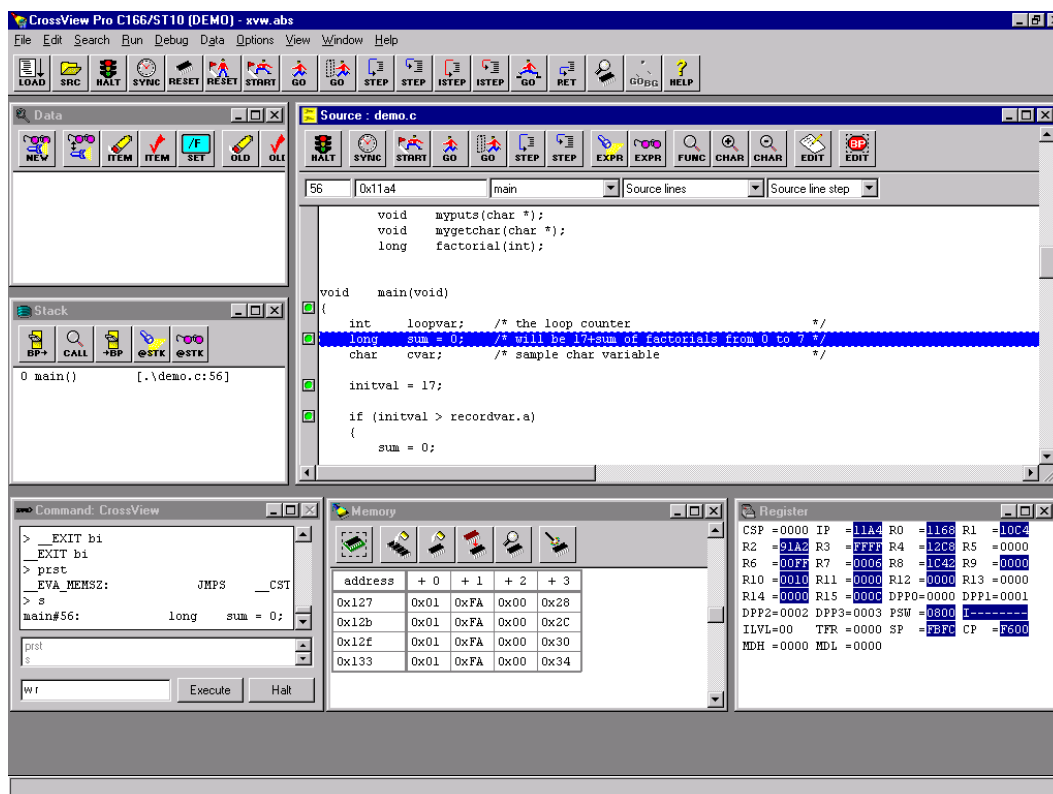
Figure 4: Suitable Development Board Jumper Settings for Debugging

- Ensure that the target hardware is properly connected to the host-PC and a power supply
- Reset the target hardware and put it in Bootstrap mode by simultaneously pressing the Reset (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, two or three seconds later, release the Boot (S2) switch

Since the required microcontroller portion to communicate with CrossView Pro will be automatically downloaded by CrossView Pro using the Bootstrap mode there is no further preparation of the target system.

4.3 Starting the Debugger

- To start the TASKING debug environment, click the debugger icon  on the EDE toolbar.
- The CrossView Pro debugger will download the monitor program automatically via bootstrap mode. Then the current opened project is downloaded.
- A CrossView Pro window similar to that shown below appears.




The window marked “Source: demo.c” is the Debug window. The *Command: CrossView* window can be used to enter commands. You may need to open, resize and /or move some windows to make your screen look something like the screen capture. You may have to open any invisible window by opening the *View* menu and selecting the window to view.

- For troubleshooting *please refer to Appendix A, Troubleshooting*.

4.4 Using the TASKING CrossView Pro Debug Features


4.4.1 Breakpoints

- Click on a memory location such as line number 69 *for* (`loopvar = 0; loopvar <= 8; ++loopvar`). A frame appears marking this position.
- You can click on *Run to cursor*  to reach this point or you can set a breakpoint by pressing the green button to the left. Set a breakpoint here. The green button will become red to indicate the first breakpoint.
- Click on *Go* and the program will run and stop at the breakpoint.
- Click on the breakpoint again to remove it.

4.4.2 Data Window

- If not already visible activate the data window by selecting *View/Data*
- To watch the value of variable *initval* either double click on the variable in the source window or press the *New* button in the data window and type in the name of the variable. Then select *Watch* and *Close* to display the variable in the data window.
- To change the value of a variable just double click on it in the data window or use the command line e.g. `initval = 10` to set the value to 10
- Structures can be viewed/modified in a similar way. Just double click on *recordvar* and select *watch* to display the structure in the data window. A '+' sign to the left shows a possible 'expansion' of the value. A double click on *recordvar* structmember *b* shows the value *recordvar.b* is pointing at. Now the '+' sign is exchanged by a '-' sign which shows that *recordvar.b* is expanded and a double click on it switches back to the 'compact' form.

4.5 Single Stepping

- CrossView Pro uses *Step (into function calls)* to single step one instruction at a time. *Step (into function calls)* is also used to enter a function in the same fashion.
- *Step (over function calls)* means to skip over a function that you are not interested in.
- *Return from function* is used to exit a function you are currently in. *Return from function* is very useful if you find yourself in a function you are not interested in and need to return quickly to your intended function.
- If CrossView Pro gets stuck, execute a reset using the *Reset application (but do not run)* button in the top toolbar . You can also use the hardware reset button on the board.
- As you step through the program - the appropriate values in the *Register* window will change color as their values are updated.

4.5.1 Memory Window

- If the Memory Window is not already visible, activate it by selecting *View/Memory* from the CrossView Pro tool bar.
- The Memory Window can be configured by pressing the *Setup Memory Display* button in the memory window
- Enter *Start address: 0x1000* in *Memory Window Setup* menu and *Base Type Long*. The addresses displayed in the Memory Window should change starting at this address. This is the memory area for the long array *table*. The recursive calculated factorial values of the variable *loopvar* are stored in this array. Single step through the for loop: *for (loopvar = 0; loopvar <= 8; ++loopvar)*.

4.5.2 Stack Window

- If not already visible view the Stack Window by selecting *View/Stack*.
- The Stack Window shows the current contents of the stack after the program has been stopped.
- The Stack Window displays the following information for each stack level: The name of the function that was called, all parameters specified to the function and the line number in the source code from which the function was called.
- Stepping into the factorial calculation *table[loopvar] = factorial(loopvar);* gives a good impression of how the Stack Window works.

4.5.3 Simulated I/O

- Simulated I/O is activated by default. Stream 0 is used for input and stream 1 is used for output. This feature lets you observe and simulate the input and output of your program before the hardware peripherals are in place.
- To activate the simulated I/O streams display in CrossView Pro select *Debug/Simulated I/O Setup* and double click on 0 and 1 in the *stream Nr* column.
- Now set a breakpoint at line *myputs("fail\n");* and press *Go*.
- Press *Step (over function calls)*. A new window should pop up which shows the contents of SIO stream: 1 (fail).
- The program now hangs in an endless loop *while (loopvar)* since *loopvar* is never zero.
- Set the variable *loopvar* to zero by modifying it's content to reach the end of the program (which includes a simulated input demonstration too).

4.6 Resetting CrossView Pro and the nanoMODUL-164

- The CrossView Pro RAM/ROM monitor debugger: The Monitor runs in the target hardware. When the *Reset the application (but do not run)* button is pressed the IP is set to zero. This command does NOT perform a reset of the target system, that is, no registers are modified except for the program counter. When the *Reset target CPU/system (but do not run)* button is pressed the target is initialized according to the power-up sequence for the processor. Almost all registers, including the system stack pointer and program counter are initialized.
- The PHYTEC Development Board does not have a hardware NMI button. A NMI is the most reliable way to stop a running program. This is even more important when you are using the Bootstrap version of the Monitor because it is difficult to re-synchronize CrossView Pro and the monitor program. For example, it is not possible for CrossView Pro to re-synchronize automatically by pressing the hardware-reset button (S1) if the monitor has been downloaded with the Bootstrap Loader. It must be restarted manually by setting the board in bootstrap mode and downloading the monitor program again.

5 Advanced User Information

This section provides advanced information for successful operation of the nanoMODUL-164 with the TASKING software tool chain.

5.1 FlashTools

5.1.1 FlashTools for DOS

Flash is a highly functional means of storing non-volatile data. One of its advantages is the possibility of on-board programming. Programming tools for the Flash device are delivered with the nanoMODUL-164 in the form a set of executable and binary files to run under DOS. This tools make use of the Bootstrap mode to transfer executable code to the nanoMODUL-164 that, in turn, download user code into the Flash. Additionally, the re-programmable Flash device on the nanoMODUL-164 enables easy update of user code and the target application in which the nanoMODUL-164 has been implemented.

Currently the nanoMODUL-164 can be populat with three different sized Flash devices: a 29F200 with 256 kByte, a 29F400 with 512 kByte or a 29F800 with 1 MB.

FlashTools always uses the Bootstrap mode to transfer the required microcontroller firmware to the nanoMODUL-164. Hence, there is no restriction in using the Flash memory for storing user code.

Before Flash programming FlashTools will adjust the internal Chip Select Unit and the bus configuration of the microcontroller during the Bootstrap download. This results in the following programming Memory-Model:

- ◆ Flash addressable at 00:0000H-07:FFFFH
(up to 512 kByte Flash)
- ◆ RAM addressable at 08:00000H-0F:FFFFH
(up to 512 MB RAM)

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash the Memory-Model is different:

RAM from 0x000000 to 0x0FFFFFF,
ROM from 0x1000000 to 0x1F7FFF).

Using the Bootstrap mode to transfer the required microcontroller firmware to the nanoMODUL-164 the FlashTools maintains its independent Flash programming capability.

5.1.2 FlashTools for Windows

Flash is a highly functional means of storing non-volatile data. One of its advantages is the possibility of on-board programming. Programming tools for the Flash device are delivered with the nanoMODUL-164 in the form a set of executable and binary files that run under Windows9x/NT. These tools make use of the Bootstrap mode to transfer executable code to the nanoMODUL-164 that, in turn, download user code into the Flash. Additionally, the re-programmable Flash device on the nanoMODUL-164 enables easy update of user code and the target application in which the nanoMODUL-164 has been implemented.

Currently the nanoMODUL-164 can be populat with three different sized Flash devices: a 29F200 with 256 kByte, a 29F400 with 512 kByte or a 29F800 with 1 MB.

FlashTools always uses the Bootstrap mode to transfer the required microcontroller firmware to the nanoMODUL-164. Hence, there is no restriction in using the Flash memory for storing user code.

Before Flash programming FlashTools will adjust the internal Chip Select Unit and the bus configuration of the microcontroller during the Bootstrap download. This results in the following programming Memory-Model:

- ◆ Flash addressable at 00:0000H-07:FFFFH
(up to 512 kByte Flash)
- ◆ RAM addressable at 08:00000H-0F:FFFFH
(up to 512 MB RAM)

If you have a nanoModul-164 with 1 MB SRAM or 1 MB Flash the Memory-Model is different:

RAM from 0x000000 to 0x0FFFFFF,
ROM from 0x1000000 to 0x1F7FFF).

Using the Bootstrap mode to transfer the required microcontroller firmware to the nanoMODUL-164 the FlashTools maintains its independent Flash programming capability.

5.2 Start.asm

The code within the assembly file *start.asm* and its include files is responsible for the initial controller configuration and the startup initialization of your C project. This includes adjusting the properties of the external bus signals and Chip Select signals, setting of the system stack, initialization of variables and clearing of memory areas.

This code will execute prior to the execution of user code. The startup code occupies the Reset Vector of the application, which is the location where the microcontroller starts execution after reset (0x0000). After performing the initialization steps your individual *main()* function is called by the startup code.

The configuration of the startup code is done via macro preprocessor defines which are generated automatically by EDE according to the settings in *EDE/CPU Options*.

Because the startup code is modified by macro preprocessor defines it is not necessary to copy the start.asm file (located in \lib\src folder) into the project folder. But it needs to be added to the project files.

5.3 Linking and Locating

The Linker has to combine several relocateable object modules contained in object files and/or libraries to generate a single absolute object.

In addition the Linker must locate several segments of code type, constants and data to fixed address locations within the address range of the microcontroller: This ensures the natural or explicitly declared properties of these segments.

Data segments must always be located in Random Access Memory (e.g. RAM), code and constant segments should be located in any kind of non-volatile memory (e.g. Flash). The C166 family has a von Neumann architecture which uses the same read signal to fetch data and also code or constants. To distinguish between non-volatile and modifiable memory, physically different memory devices must be addressable within different address ranges. A list of the compiler generated section names can be found in TASKINGs *C Compilers User's Guide* in chapter 3.2.2 *Section Allocation*.

It is required that all RAM data sections are located to any internal RAM of the C164 or to any external RAM of the nanoMODUL-164. All ROM code and –data sections must also be located to any internal non-volatile memory (e.g. Flash, OTPROM) of the C164 or any external Flash memory of the nanoMODUL-164.

The *small Memory-Model* which is the only one supported in the eval version of the TASKING C166/ST10 toolchain requires a special treatment of near RAM- and ROM-data sections. More detailed information on this subject can be found in TASKINGs *C Compilers User's Guide* in chapter 3.2.1.2 *Small Memory-Model*.

To ensure proper execution of your application you must take into account the runtime Memory-Model when linking and locating. This means that you must instruct the Linker/Locator where to assume external RAM for locating data classes and Flash for locating code and constant classes.

We recommend that you generate a **.map* file for your project and inspect the memory map information within this file. Accurately compare this information to the physical Memory-Model resulting from your settings selected within the *EDE/CPU Options* menu.

5.4 Debugging Using Monitor Kernel

Whenever you decide to use the CrossView Pro ROM monitor debugger to debug your application, some special precautions must be taken into consideration to ensure proper code execution of your application.

Your application and the ROM monitor kernel contained in the files *b167nrb.sre* and *m164r.sre* must share some memory locations within the target system. If you do not consider the physical Memory-Model already claimed by the kernel and the memory requirements of the kernel, you may get conflicts in memory use. This typically leads to variables containing not their assigned value, functions returning bad results and modified code.

To prevent this kind of conflicts you can instruct EDE to reserve the memory areas needed by the monitor kernel. This is done in *EDE/Project Options/Startup* menu by checking the *ROM or RAM monitor on evaluation board* box.

To obtain information about the memory requirements of the Monitor, you can take a look at *Reserve memory for ROM & monitor resources*: in *EDE/Linker/Locator Options/Reserve* menu. In addition the monitor uses the NMI and the serial interrupt. This interrupts may not be used by your application.

5.5 Using ROM Debug Monitor in Flash Memory

It is also possible to load the monitor kernel into Flash memory. In this case the bootstrap mode is not necessary anymore since the monitor program is running immediately after reset. *For more information about the monitor kernel in Flash memory you can take a look into the Addendum: C166/ST10 Family Target Boards chapter 5.1.2 ROM Debug Monitor Using Dual Vector Table of the CrossView Pro Debugger User's Guide.*

The sourcefiles of the monitor program are included in the full version of the product. The monitor kernel is royalty free and may be included in the customer application.

Appendices

A Troubleshooting

A.I Initialization Error: ROM Monitor DLL Could not Initialize or Access the Target Board

- board may not be in bootstrap mode. Try to activate the bootstrap mode again and press *Retry* button.
- check for correct power supply and serial connection
- the selection in *ROM/RAM Monitor Board* may be wrong in *EDE/CrossView Pro Options/Debugger* menu
- the baudrate in *EDE/Cross View Pro Options/Communication* may be set too high. With 'default' oscillator frequencies normally the maximum value is 38400 baud.
- the wrong COM port may be selected in *EDE/Cross View Pro Options/Communication* menu
- the serial FIFO buffer in Windows 95 can cause transmission problems. CrossView Pro may have problems completing the communication initialization process. This can be intermittent. The FIFO can be disabled under *Controlpanel/System/Device Manager/Port Settings/Advanced*. Make sure *Use FIFO buffers* in this menu is not activated.

A.II Program Download Never Stops

- the downloaded program may overwrite the monitor kernel. Verify whether *ROM or RAM monitor on evaluation board* is checked in *EDE/Project Options/Startup* menu or take a look into the .map file of your project.
- the address window settings may be wrong. No address window should be activated. Check BusconX and AddrSelX settings in *EDE|CPU Options*.

A.III Placing Code Breakpoint at 0xXXXXXX Failed: Memory not Writeable

When the CrossView Pro ROM monitor debugger is used the complete application which will be debugged must be allocated in RAM area. The error message is generated because parts of the application are allocated in read only memory. You can verify this by checking the projects mapfile (.map).

Document: nanoMODUL-164 QuickStart Instructions
Document number: L-455e_1, August 1999

How would you improve this manual?

Did you find any mistakes in this manual? _____ page

Submitted by:

Customer number: _____

Name: _____

Company: _____

Address: _____

Return to:

PHYTEC Technologie Holding AG
Postfach 100403
D-55135 Mainz, Germany
Fax : +49 (6131) 9221-33

Published by

PHYTEC

© PHYTEC Meßtechnik GmbH 1999

Ordering No. L-455e_1
Printed in Germany