

# **kitCON-167**

## **QuickStart Instructions**

**Using PHYTEC FlashTools 16W and the Keil  $\mu$ Vision2 Software  
Evaluation Development Tool Chain**

**Note: The PHYTEC Spectrum CD includes the electronic version of  
the English kitCON-167 Hardware Manual**

**Hinweis: Die PHYTEC Spectrum CD beinhaltet die elektronische  
Version des deutschen kitCON-167 Hardware Manuals**

**Edition: January 2001**

A product of a PHYTEC Technology Holding company

In this manual are descriptions for copyrighted products which are not explicitly indicated as such. The absence of the trademark (©) symbol does not infer 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 <a href="mailto:order@phytec.de">order@phytec.de</a>	+1 (800) 278-9913 <a href="mailto:info@phytec.com">info@phytec.com</a>
Technical Support:	+49 (6131) 9221-31 <a href="mailto:support@phytec.de">support@phytec.de</a>	+1 (800) 278-9913 <a href="mailto:support@phytec.com">support@phytec.com</a>
Fax:	+49 (6131) 9221-33	+1 (206) 780-9135
Web Site:	<a href="http://www.phytec.de">http://www.phytec.de</a>	<a href="http://www.phytec.com">http://www.phytec.com</a>

1<sup>st</sup> Edition: January 2001

---

<b>1</b>	<b>Introduction to the Development Kit.....</b>	<b>1</b>
1.1	Development Kit Documentation .....	1
1.2	Overview of this QuickStart Instruction.....	2
1.3	System Requirements .....	2
1.4	The PHYTEC kitCON-167.....	4
1.5	The Keil $\mu$ Vision2 Software Evaluation Development Tool Chain .....	5
<b>2</b>	<b>Getting Started.....</b>	<b>9</b>
2.1	Installing Development Kit Software .....	9
2.2	Interfacing the kitCON-167 to a Host-PC .....	16
2.3	Downloading Example Code with FlashTools 16W .....	18
2.3.1	“Hello” .....	23
2.3.2	“Blinky” .....	28
<b>3</b>	<b>Getting More Involved.....</b>	<b>31</b>
3.1	Starting the $\mu$ Vision2 Tool Chain.....	31
3.2	Creating a New Project and Adding an Existing Source File.....	32
3.3	Modifying the Source Code.....	39
3.4	Saving the Modifications .....	40
3.5	Setting Options for Target .....	40
3.6	Building the Project .....	43
3.7	Downloading the Output File .....	43
3.8	“Hello” .....	45
3.8.1	Creating a New Project.....	45
3.8.2	Modifying the Example Source .....	46
3.8.3	Setting Target Options.....	46
3.8.4	Building the New Project .....	46
3.8.5	Downloading the Output File .....	47
3.8.6	Starting the Terminal Emulation Program.....	48
<b>4</b>	<b>Advanced User Information .....</b>	<b>49</b>
4.1	FlashTools 16W .....	49
4.2	Start167.a66 .....	51
4.3	Linking and Locating.....	53
	 Figure 1: Development Board Jumper Settings .....	 16
	Figure 2: Power Connector .....	17



# 1 Introduction to the Development Kit

## This tutorial provides:

- general information on the PHYTEC kitCON-167 Single Board Computer,
- an overview of Keil's  $\mu$ Vision2 software evaluation development tool chain, and
- instructions on how to run example programs on the kitCON-167, in conjunction with  $\mu$ Vision2.

Please refer to the [kitCON-167 Hardware Manual](#) for specific information on such board-level features as [jumper configuration](#), [memory mapping](#) and [pin layout](#). Selecting the links on the electronic version of this document links to the applicable section of the kitCON-167 Hardware Manual.

## 1.1 Development Kit Documentation

This “Development Kit” includes the following electronic documentation on the enclosed “PHYTEC Spectrum CD-ROM”:

- the PHYTEC [kitCON-167 Hardware Manual](#)
- controller [User's Manuals and Data Sheets](#)
- this QuickStart Instruction with general “Development Kit” description, software installation hints and two example programs enabling quick out-of-the box start-up of the kitCON-167 in conjunction with the  $\mu$ Vision2 software development tools:

## **1.2 Overview of this QuickStart Instruction**

- 1) The “*Getting Started*” section uses two example programs “*Hello*” and “*Blinky*” - to demonstrate the download of user code to the Flash device using PHYTEC’s FlashTools 16W.
- 2) The “*Getting More Involved*” section provides step-by-step instructions on how to modify both examples, create and build new projects and generate and download output files to the kitCON-167 using the Windows-based  $\mu$ Vision2 tool.

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

## **1.3 System Requirements**

Use of this “Development Kit” requires:

- the PHYTEC kitCON-167 with the included DB-9 serial cable and AC adapter supplying 8-13 V/min 300 mA,
- the PHYTEC Spectrum CD,
- an IBM-compatible host-PC (486 or higher running at least Windows95/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](mailto:support@phytec.com) - or - [support@phytec.de](mailto:support@phytec.de)



<http://www.keil.com>  
[support@keil.com](mailto:support@keil.com)

## **1.4 The PHYTEC kitCON-167**

The kitCON-167 is a EURO-sized (100 x 160 mm) Evaluation Board for the Siemens C167x microcontroller of the C16x-microcontroller family in a MQFP-144 package. The kitCON-167 is pre-configured and equipped with all necessary connectors required for immediate start-up.

### **kitCON-167 Technical Highlights**

- Evaluation Board in EURO-card format 160 x 100 mm, including wrap-field (90x60 mm) for easy layout of user circuitry
- Improved interference safety through multi-layer technology
- Requires single unregulated 8V= to 12V=/500 mA power source
- 256Kbyte Flash on-board (PLCC-package)(optionally expandable to up to 2MB)<sup>1</sup>
- on-board Flash-programming
- No need for a dedicated programming voltage through use of 5V-Flash devices
- 64kByte SRAM on board (optionally expandable to up to 2MB)<sup>1</sup>
- 512Byte serial EEPROM
- On-chip bootstrap loader
- All controller ports, as well as data and address lines, are extended from the controller to the pins of the kitCON-Connector in the middle of the board
- RS-232 serial interface, available at DB9-socket P1
- Supplemental debug-interface (RS-232), available at DB9-plug P2
- If equipped with a C167Cx controller, a CAN-Interface is available at DB9-plug P2, too
- Two LED's to display the boards status
- Sixteen freely programmable LED's at port P2
- Real-Time Clock<sup>2</sup>
- Up to three free Chip-Select signals for easy connection of peripheral devices (controller-dependent)

---

<sup>1</sup>: For more information about additional configurations see the PHYTEC product catalog

<sup>2</sup>: The real-time clock is not available on boards included in Siemens' Starter Kits

---

- Provision for operating the board with an ICE/connect-16x for easy emulation of the controller
- Provision for operating the board with a Quad-Connector for easy emulation of the controller

## 1.5 The Keil $\mu$ Vision2 Software Evaluation Development Tool Chain

Keil  $\mu$ Vision2 fully supports the entire Infineon C166 microcontroller family. This includes a C compiler, macroassembler, Linker/Locator and the Simulator and Target Monitor within the  $\mu$ Vision2 IDE. Specific chips supported are the 161, 163, 164-CI, 165, 166 and 167CR. Future derivatives are easily accommodated due to the flexible Keil C compiler design.

$\mu$ Vision2 supports all in-circuit emulators that adhere to the Infineon OMF166 debugging specification. The Keil OH166 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 kitCON-167 target board.

$\mu$ Vision2 consists of the following executables:

- **C Compiler**    c166.exe
- **Assembler**    a166.exe
- **Linker**        l166.exe
- **Converter**    oh166.exe
- **$\mu$ Vision2**     Uv2.exe (a Windows-based application)

Once installed, the default destination location for the DOS based files is the *C:\Keil\C166\bin* directory while  $\mu$ Vision2 is in *C:\Keil\Uv2*. If using the professional (i.e. full) version of the Keil tool chain, the default destination location for these files is the *C:\C166\bin* directory. Access to these programs from Windows is accomplished with  $\mu$ Vision2. The entire tool set can be run from  $\mu$ Vision2 or directly from DOS with batch files. The Evaluation version is limited in code size to 8 kByte. Other than these restrictions, all features operate normally.

## **µVision2 IDE**

µVision2 is a Windows-based Graphical User Interface for the C compiler and assembler. All compiler, assembler and linker options are set with simple mouse clicks. µVision runs under Windows 95/98 and NT. This Integrated Development Environment (IDE) has been expressly designed with the user in mind and includes a fully functional editor.

All IDE commands and functions are accessible via intuitive pull-down menus with prompted selections. An extensive Help utility is included. External executables can be run from within µVision2, including emulator software.

## **C166 C Compiler for the entire Infineon 166/167 family**

The C166 ANSI compiler and A166 assembler are designed specifically for the Infineon 161, 163, C164CI, 165,166, 167, 167CR, and future derivatives. The C166 compiler easily integrates into the Keil RTOS and interfaces and passes debug information to the µVision2 Simulator and all in-circuit emulators. Extensions provide access to on-chip peripherals.

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

## **A166 Macroassembler**

The Professional Kit (PK) macroassembler is included with the PK Compiler package or is available separately. It is DOS-based or can be run from µVision2 and includes all utilities needed to complete your project.

## **Debug Environment**

µVision2 contains a software simulator supporting debugging either via software on a host-PC or in target hardware. When operated in conjunction with the Keil Monitor resident in target hardware µVision2 enable the following debugging functions:

- run/halt,
- set breakpoints,
- examine/change memory,
- view the stack,
- view/set peripheral information
- apply virtual external signals.

µVision2 has a performance analysis feature to ensure your code runs efficiently. In addition, µVision2 has a disassembler/assembler that allows the modification of user code without recompiling. The evaluation version of µVision2 is restricted to a 8 kByte in manipulable code. Other than this restriction the evaluation tool chain (EK) functions exactly as does the full (PK) 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 Keil products on the PHYTEC target board. The PK full version has no restrictions and is fully ANSI compliant.

## **FR166 Full-Function RTOS for the Infineon C166 Family**

The FR166 is a multi-tasking real-time operating system for the Infineon 166 family. You can manage multiple tasks on a single CPU making your programs much easier to develop. The RTX166 Full includes CAN libraries. The RTX166 Tiny is a subset of the RTX166 Full and is included with all C166 C Compiler Kits. The EK version of the tool chain does not include an RTOS.

### **CAN (Controller Area Network) Library**

The RTX166 Full RTOS supports CAN controllers with the included libraries. The CAN libraries are sold with the RTOS and support 11-bit (Basic 1.0 CAN) and 29-bit (Full 2.0B CAN) message identifiers. Keil 166 and 8051 C compilers interface with the RTOS and CAN libraries. Keil supports all CAN microcontrollers based on the Infineon C505C, C515C, C164-CI, and C167CR. Future CAN products based on these 8051 or C16x Families are easily supported due to the flexible Keil Compiler design.

## 2 Getting Started

What you will learn with this Getting Started example:

- installing Development Kit software
- starting PHYTEC's FlashTools 16W download utility
- interfacing the kitCON-167, mounted on the Development Board, to a host-PC
- downloading example user code in hexfile format from a host-PC to the external Flash-Memory using FlashTools 16W

### 2.1 Installing Development Kit Software

When inserting 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 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:

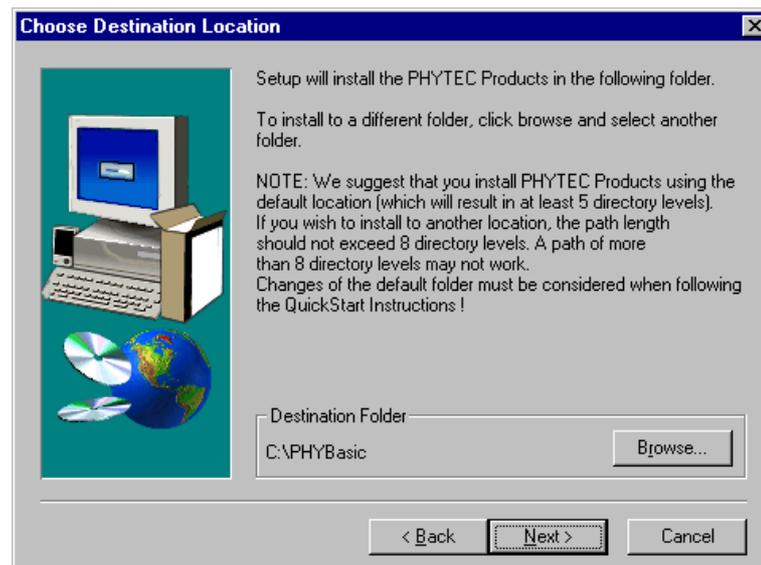


Choose *Install Basic Product Files* Button.

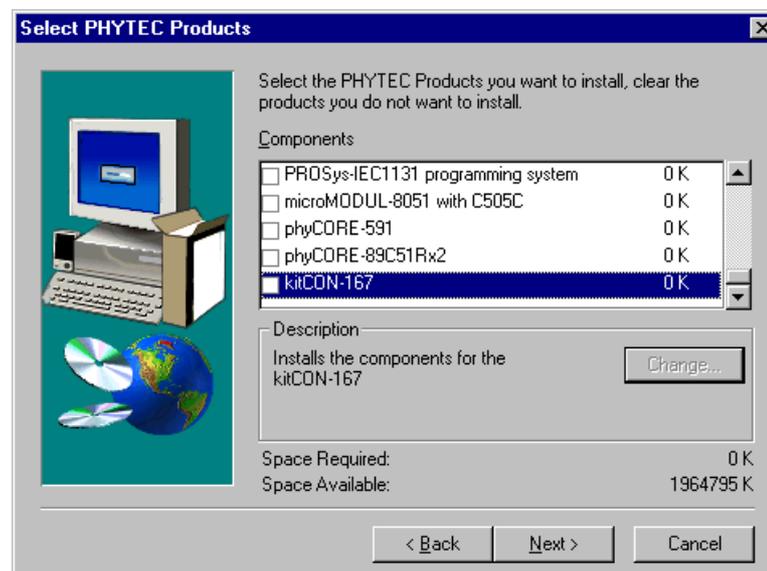
After accepting the Welcome window and license agreement select the destination location for installation of Rapid Development Kit software and documentation.

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.



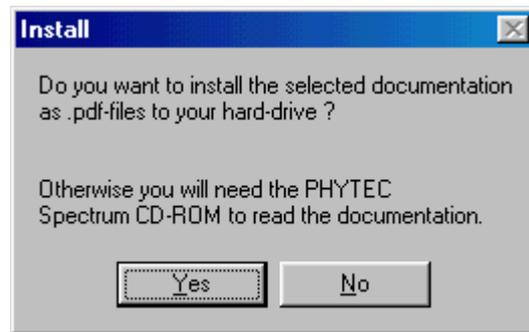
In the next window select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.



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 kitCON-167 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.

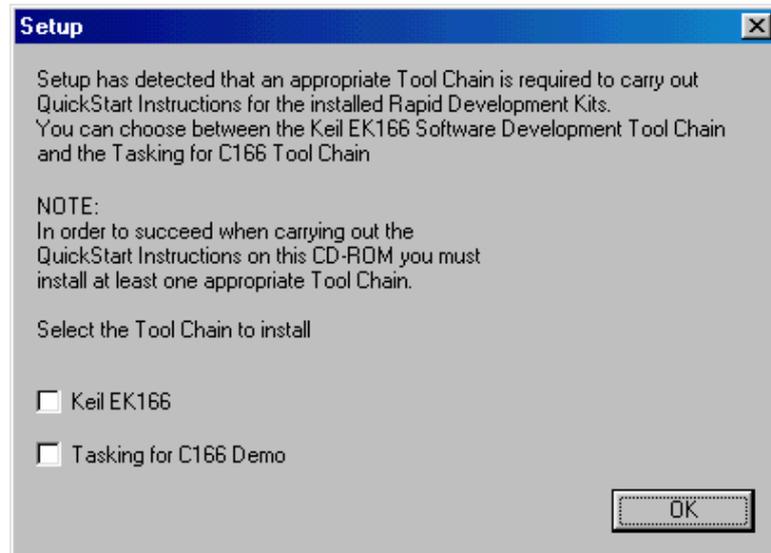


If you decide **not** to copy the documentation to your hard-drive you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

If you decide to copy the electronic documentation to your hard-drive, the documentation for this kitCON-167 Kit will also be installed to the Kit-specific subdirectory.

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

In the next window, you choose the Keil EK166 Software Development Tool Chain.



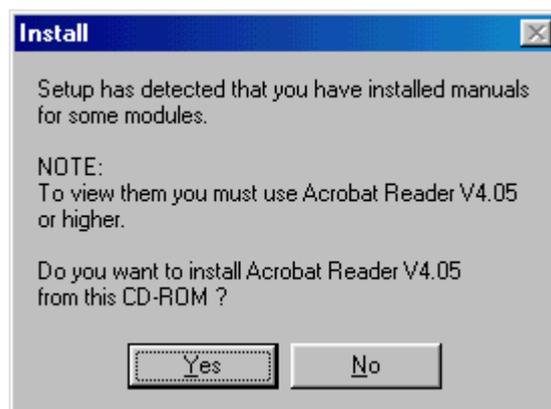
After accepting the Welcome window and license agreement select the destination location for installation of the Development Tool Chain.

Depending on the Rapid Development Kit software you have selected, the applicable Keil  $\mu$ Vision2 Evaluation Development tool chain will be installed to your hard-drive. Additional software, such as Adobe Acrobat Reader, will also be offered for installation.

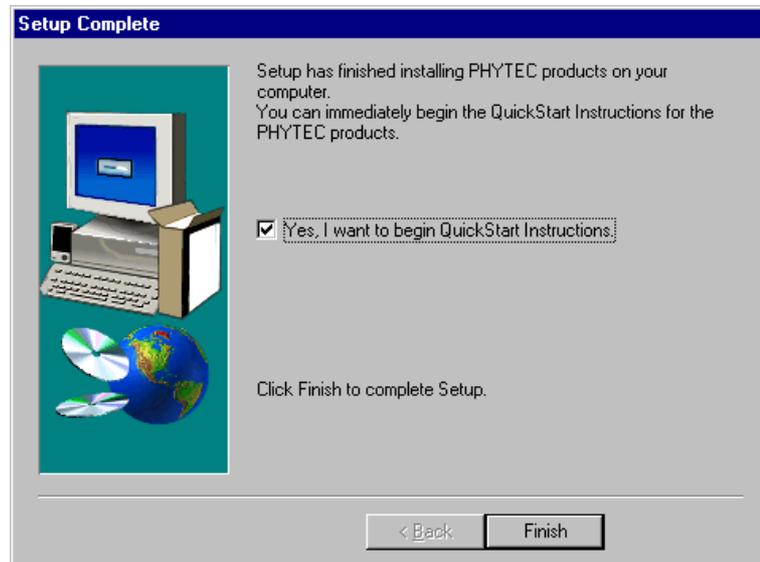
The applicable Keil tool chain must be installed to ensure successful completion of this QuickStart Instruction. Failure to install the proper software could lead to possible version conflicts, resulting in functional problems.

We recommend that you install  $\mu$ Vision2 from the Spectrum CD-ROM even if other versions of  $\mu$ Vision2 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 16W Software and the Acrobat Reader.



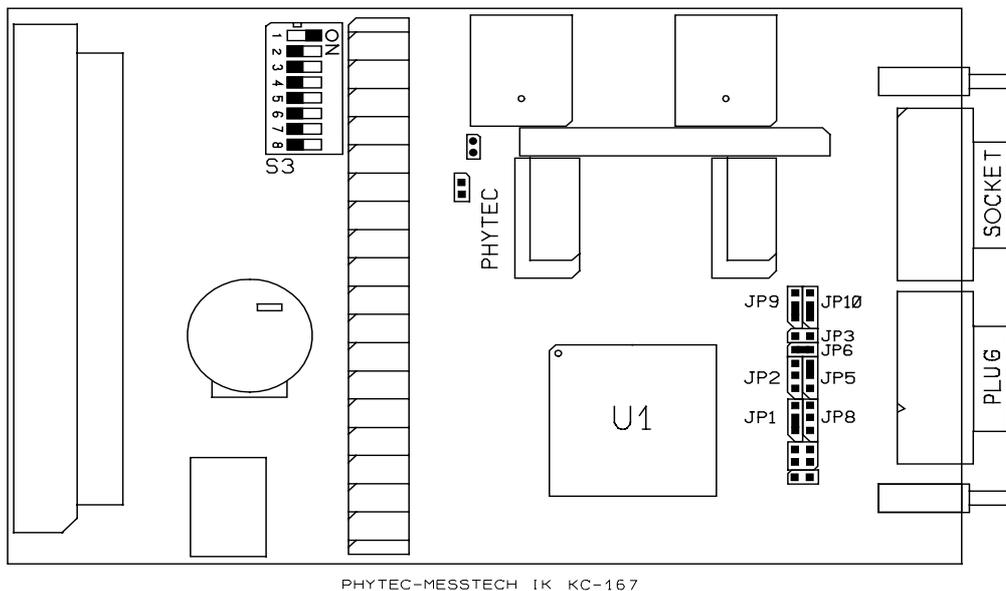
Press Finish to complete the installation and decide if you want to begin the QuickStart Instruction immediately.



## 2.2 Interfacing the kitCON-167 to a Host-PC

Connecting the kitCON-167 to your computer is simple:

- Configure the Jumpers on the Development Board as indicated below. This correctly routes the RS-232 signals to the DB-9 connector (P1) on the Development Board.
- Ensure that switch 1 of DIP switch S3 (left of the kitCON-Connector) is turned ON.

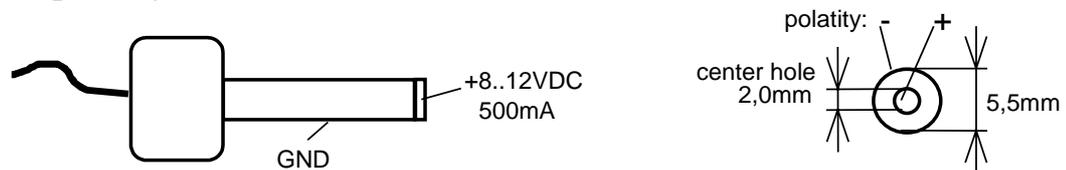


 Insertable Jumper

Figure 1: Development Board Jumper Settings

- Connect the RS-232 interface of your computer to the DB-9 RS-232 interface on the Development Board (P1) using the included serial cable.

- Using the included power adapter, connect the power socket on the board (P3) to a power supply (refer to *Figure 2* for the right polarity).



*Figure 2: Power Connector*

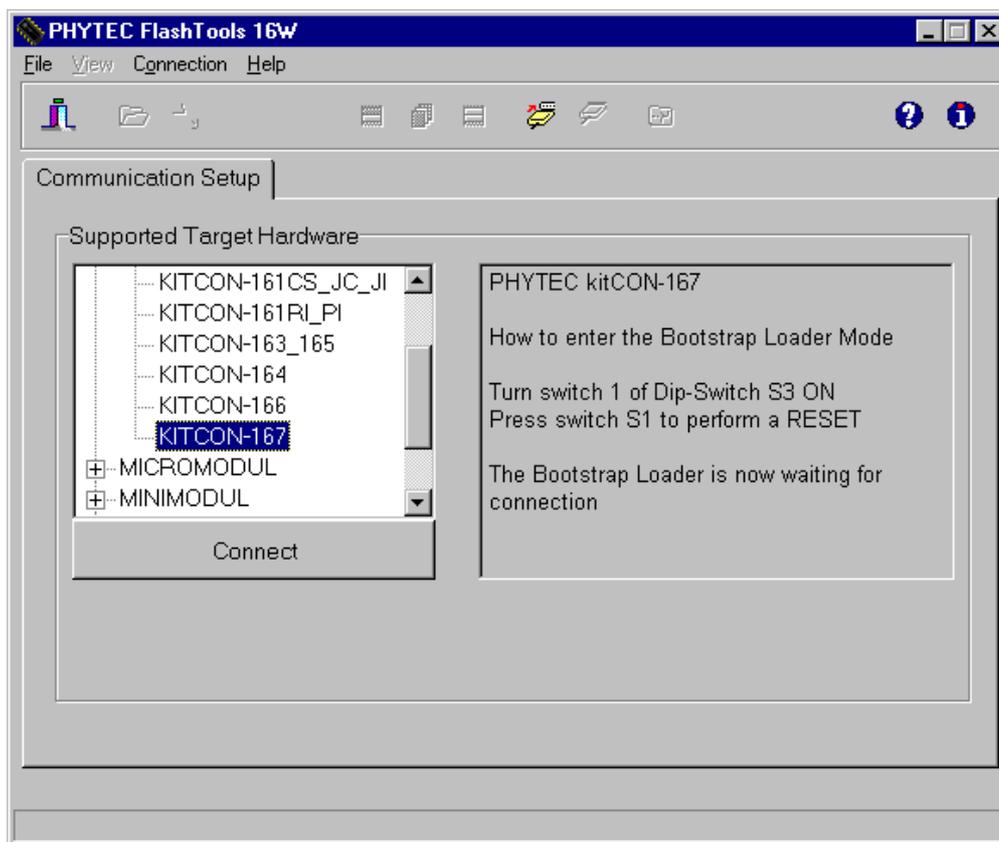
- Push switch S1 to perform a Reset and to start the Bootstrap Loader on the kitCON-167.

The kitCON should now be properly connected via the Development Board to a host-PC and power supply. After executing a Reset and rendering the board in Bootstrap mode, you are now ready to program the kitCON-167. This kitCON shall also be referred to as “target hardware”.

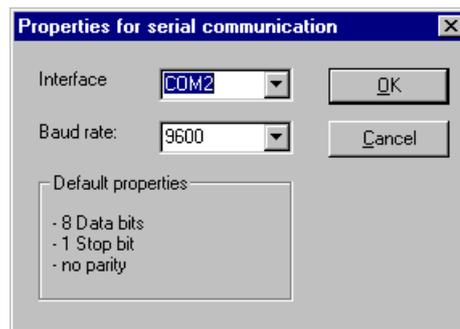
## 2.3 Downloading Example Code with FlashTools 16W

Start FlashTools 16W by double-clicking on the FlashTools 16W icon or by selecting *FlashTools 16W* from within the *Programs/PHYTEC* program group.

- The Connect tab of the FlashTools 16W Worksheet window will now appear.
- Double click on KITCON

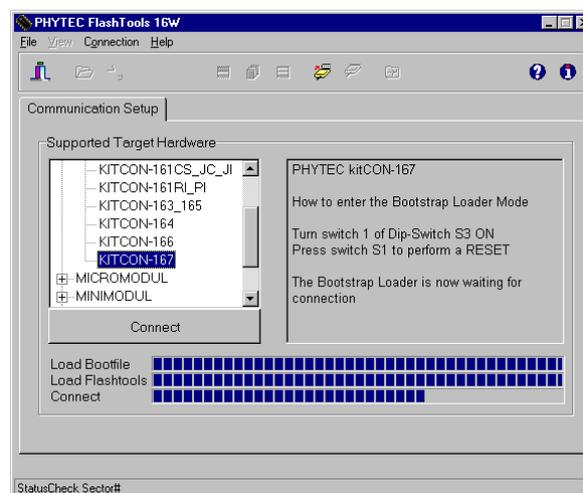


- Select the correct module from the list and press the *Connect* button.



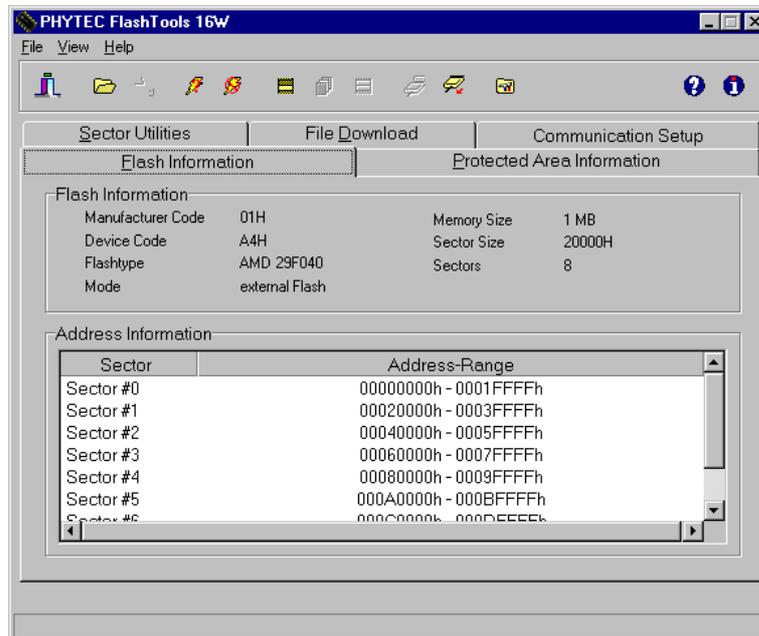
- Select the correct serial port for your host-PC and a 9600 baud rate.
- Click the *OK* button to load the module based part of the FlashTools to the target hardware.

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*.

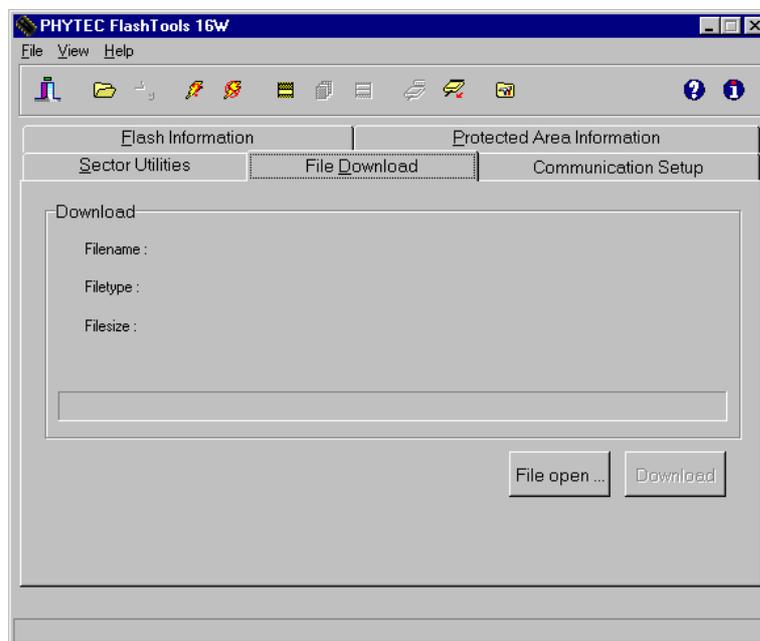


- After the data transfer you will see FlashTools 16W Worksheet window with the following tabs.

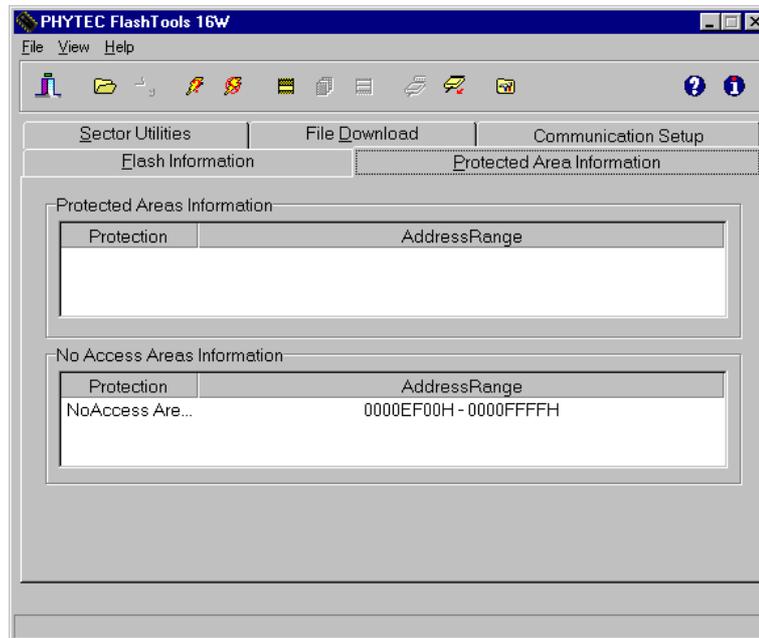
Flash Information shows sector and address ranges in Flash-Memory:



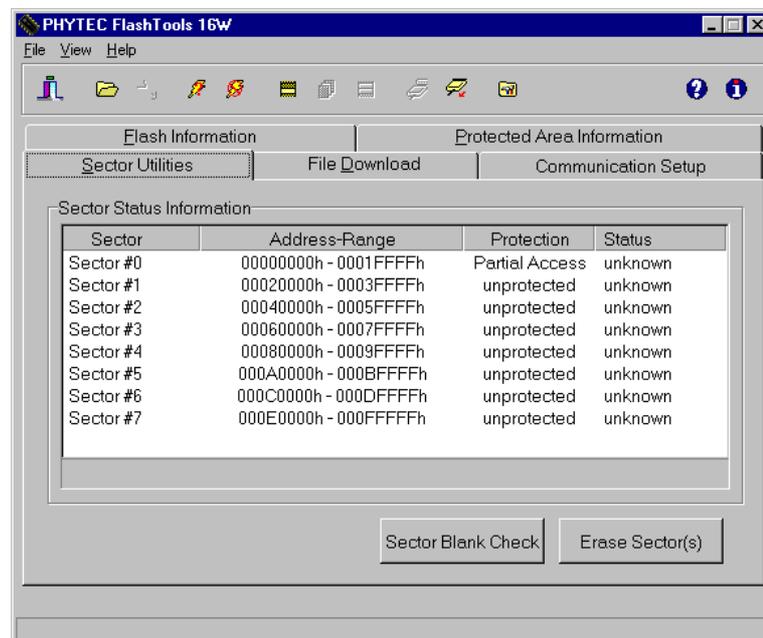
File Download downloads specified hexfiles to the target hardware:



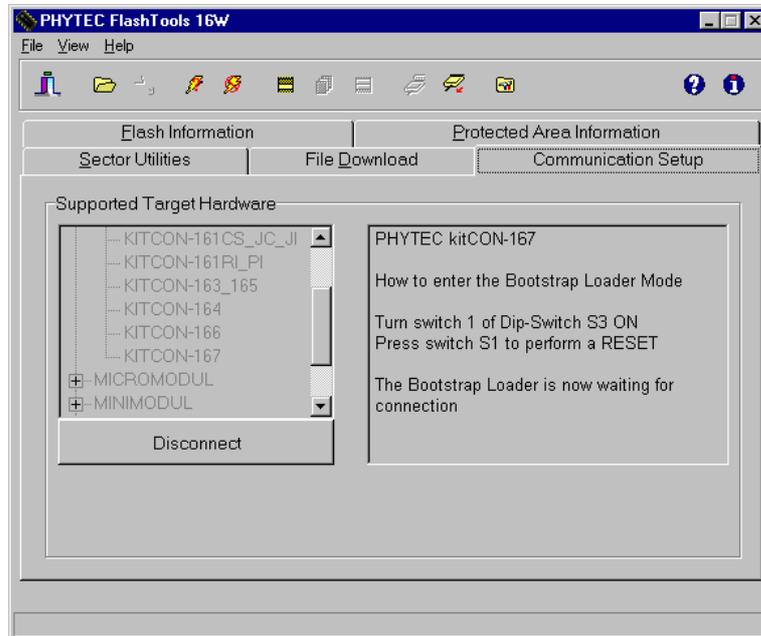
*Protected Areas Information* shows protected areas of Flash-Memory:



*Sector Utilities* allow erasure of individual sectors of Flash-Memory:



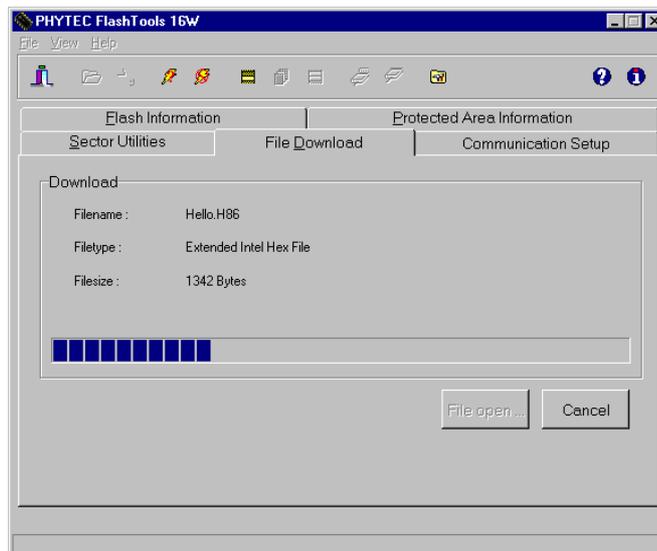
*Communication Setup* provides you with the possibility to disconnect the target and reconnect with an equal one. Programming of several targets is possible for you thereby.



### 2.3.1 “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 all sectors and click on the *Erase Sector(s)* button. You can see the Flash sectors 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 kitCON-167 Demo directory (default location *C:\PHYBasic\KC167\Demos\Keil\Hello\hello.h86*) and click *Open*.
- Click on the *Download* button. You can watch the status of the download of the *hello.h86* 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 16W 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 16W worksheet, highlight all sectors and erase them. Then repeat the download procedure.

- Returning to the *Communication Setup* tab, click on the *Disconnect* button and exit the FlashTools.
- Turn **OFF** switch 1 of DIP-switch S3

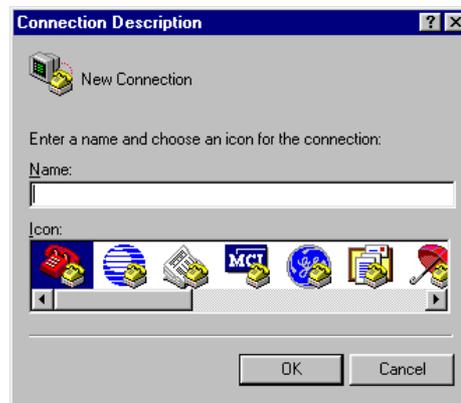
Monitoring the execution of the Hello demo requires use of a terminal program, such as the HyperTerminal included within Windows.

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



- Double-click on the HyperTerminal icon “Hypertrm” to create a new HyperTerminal session.

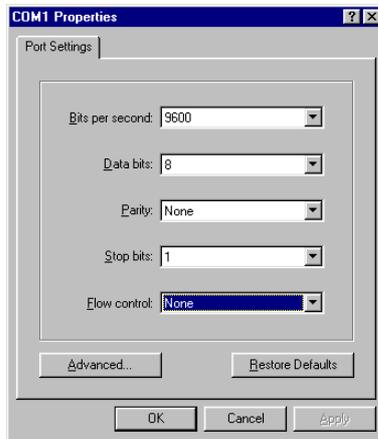
- The Connection Description window will now appear. Enter “COM Direct” in the “Name” combobox (be sure to specify the correct COM parameter for your system).



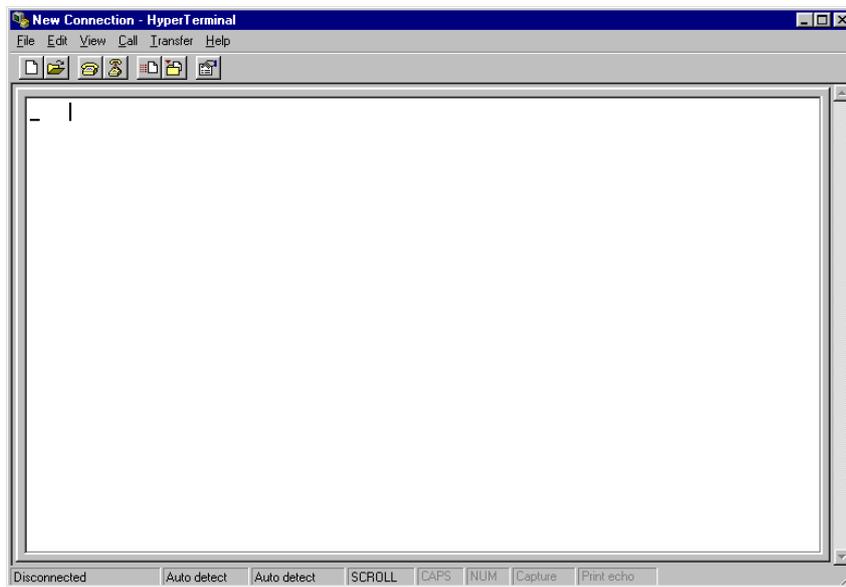
- Next press the *Ok* button. This creates a new HyperTerminal session named “COM Direct” and advances you to the next HyperTerminal window.
- The “New Connections Properties” window will now appear. Specify *Direct to COM1* under the *Connect Using* combobox (be sure to indicate the correct COM setting 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 “COM Direct – HyperTerminal” monitoring window. Notice the connection status report in the bottom lower corner of the window.



- Resetting the Development Board (at S1) will execute the *hello.h86* 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  in HyperTerminal toolbar and exit HyperTerminal.

If no output appears in the HyperTerminal window check the power supply, the COM parameters, the RS-232 connection and the DIP-switch S3.

The demo application within the file *hello.h86* initializes the serial port of your kitCON-167 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 kitCON-167 is populated with an oscillator with a frequency of just 5 MHz. An internal PLL (Phase Locked Loop) device renders an internal 20 MHz clock frequency. If your kitCON-167 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.

### 2.3.2 “Blinky”

The “Blinky” example sends a program to the Flash that, when executed, manipulates the LED D13 on the Development Board that is mounted next to the red LED D4. This second example program provides a review of the FlashTools 16W download procedure. For detailed commentary on each step, described below in concise form, refer back to the “Hello” example starting at *section 2.3.1*

- Ensure that the target hardware is properly connected to the host-PC and a power supply
- Render the target hardware into Flash Programming Mode by simultaneously pressing the Reset (S2) and Boot (S1) switches on the Development Board and then releasing first the Reset (S2) and, two to three seconds later, release the Boot (S1) switch
- Start FlashTools 16W
- Double click on KITCON
- Select the correct module from the list and press the *Connect* button.
- Select the correct serial port for your host-PC and a 9600 baud rate.
- Click the *OK* button to load the module based part of the FlashTools to the target hardware.
- Choose the *Sector Utilities* tab, highlight all sectors and click on the *Erase Sector(s)* button. You can see the Flash sectors being erased at the bottom left hand of the window.
- Wait until the status check in the lower left corner of the FlashTools 16W Worksheet finishes
- 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\KC167\Demos\Keil\Blinky\blinky.h86* folder (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 Reset (S2) on the Development Board
- Successful execution of the program will flash the LED with equal on and off durations

You have now successfully downloaded and executed two pre-existing example programs in \*.*h86*-file format.



### 3 Getting More Involved

What you will learn with this example:

- how to start the  $\mu$ Vision2 tool chain
- how to configure the  $\mu$ Vision2 IDE (Integrated Development Environment)
- how to modify the source code from our examples, create a new project and build and download an output \*.h86-file to the target hardware

#### 3.1 Starting the $\mu$ Vision2 Tool Chain

The  $\mu$ Vision2 evaluation software development tool chain should have been installed during the install procedure, as described in section 2.1.

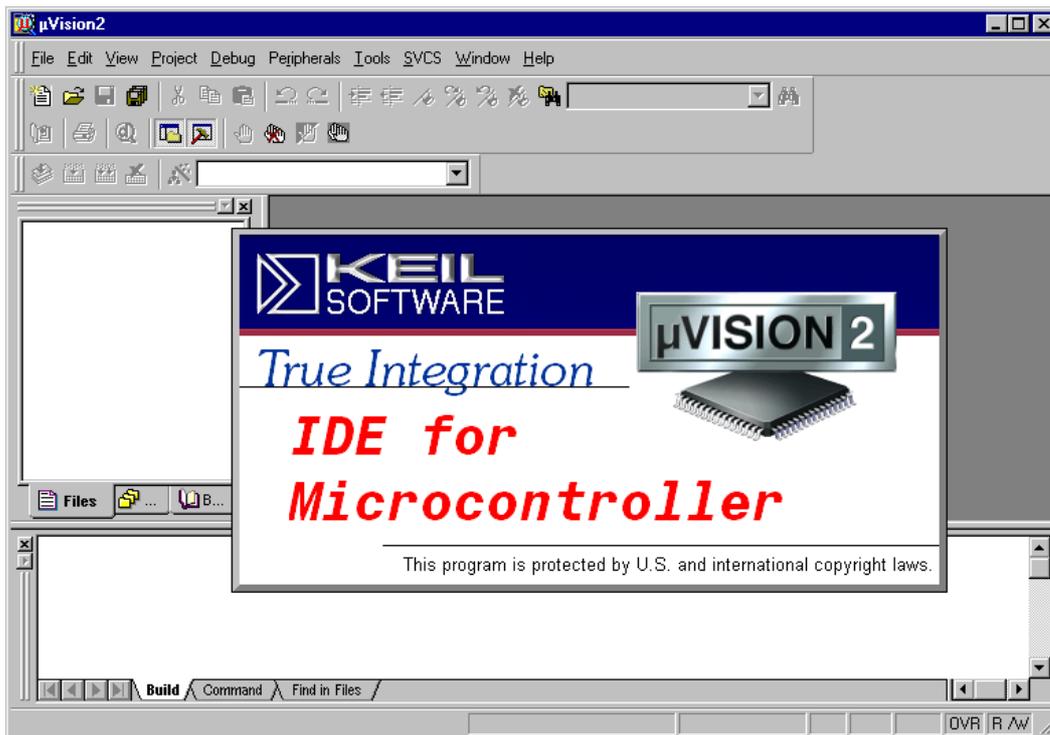
You can also manually install  $\mu$ Vision2 by executing *ek166.exe* from within the `\Software\Keil\Ek166` directory of your PHYTEC Spectrum CD.

**Attention:**

You must use the Keil tool chain provided on the accompanying Spectrum CD in order to complete this QuickStart Instruction successfully. Use of a different version could lead to possible version conflicts, resulting in functional problems.

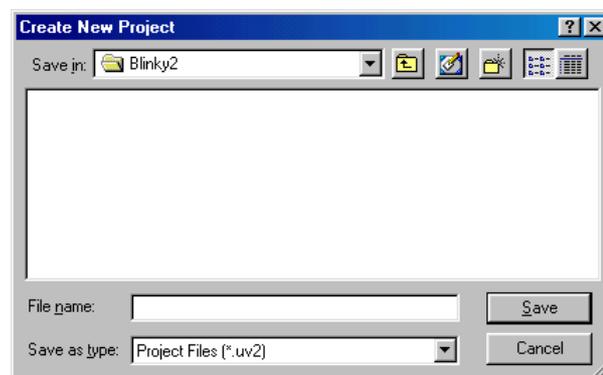
Start the tool chain by selecting *Keil  $\mu$ Vision2* from within the *Programs* group.

After you start  $\mu$ Vision2, the window shown below appears. From this window you can create projects, edit files, configure tools, assemble, link and start the debugger. Other 3<sup>rd</sup> party tools such as emulators can also be started from here.

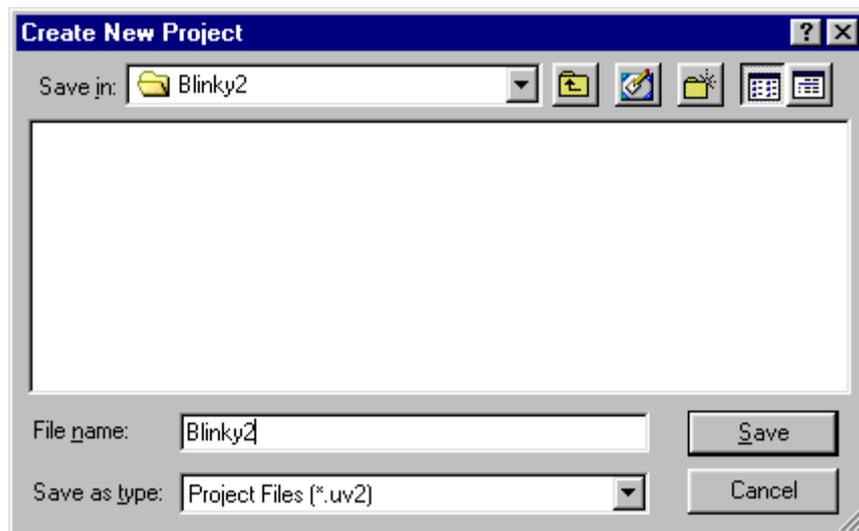


### 3.2 Creating a New Project and Adding an Existing Source File

- To create a new project file select from the  $\mu$ Vision2 menu **Project/New Project....** This opens a standard Windows dialog that asks you for the new project file name.
- Change to the project directory created by the installation procedure (default location **C:\Phybasic\KC167\Demos\Keil\Blinky2**).



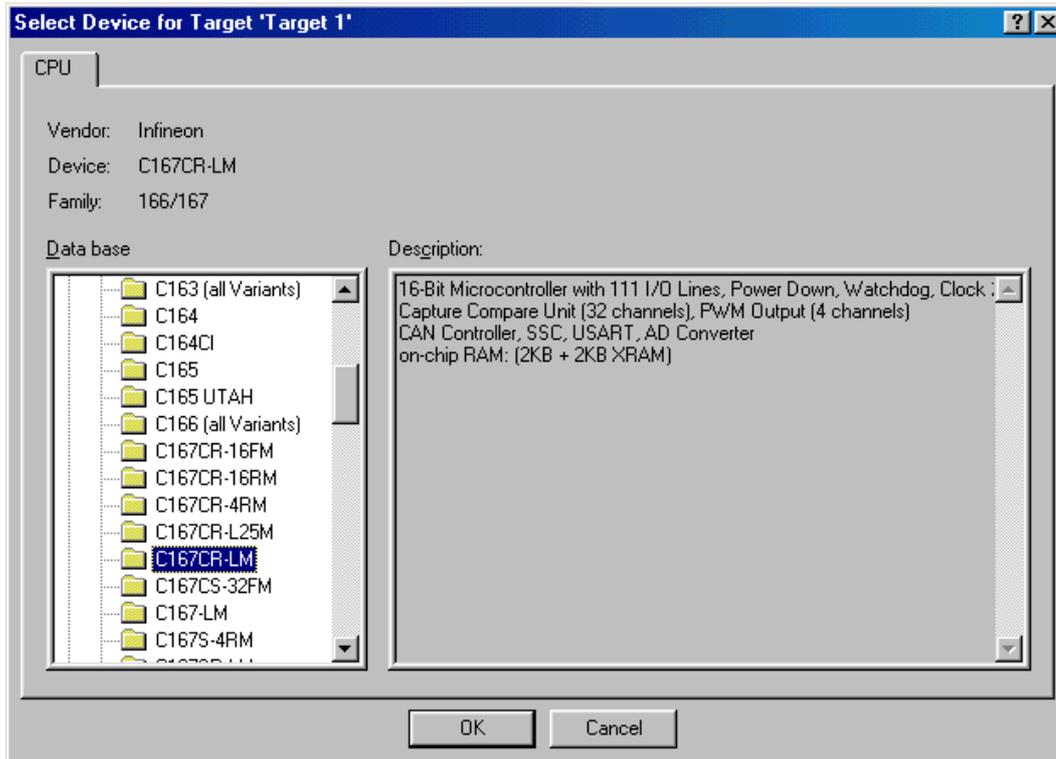
- In the combobox 'File name', enter the file name of the project you are creating. For this tutorial, enter the name ***Blinky2*** and press *Save*.



- Double click on *Infineon* as manufacturer for the CPU within the next window which opens automatically<sup>1</sup>. The kitCON-167 is equipped with a C167CR-LM CPU. Choose this controller type from the list as shown below. This selection sets necessary tool options for the C167CR-LM device and simplifies in this way the tool configuration.

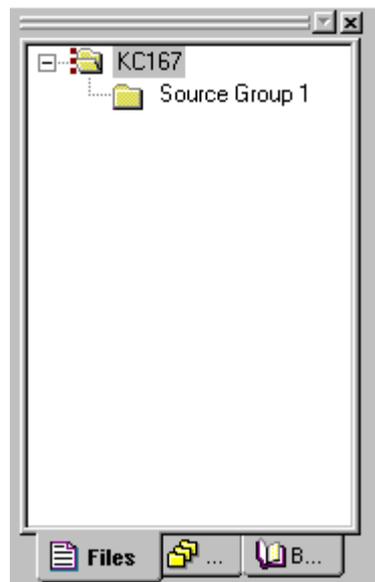
---

<sup>1</sup>: The same window opens by choosing Select Device for Target from the Project menu



- Press the *OK* button

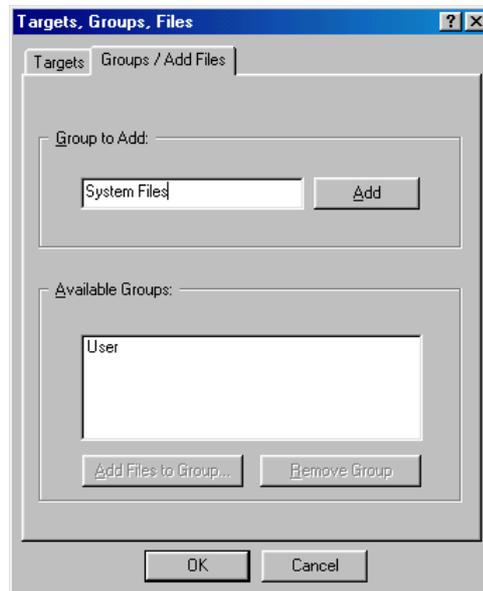
- To give the target of our project a name select the default *Target 1* in the project window and click on it. Change the name into *KC167* and press return.



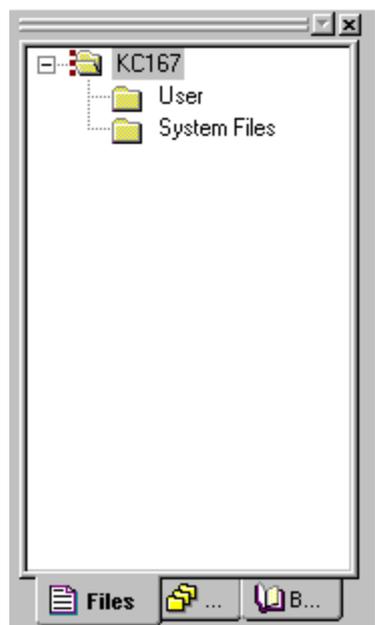
- Select the file group *Source Group 1* in the *Project Window – Files* page and click on it to change the name into *User*.
- Click with the right mouse key in the *Project Window* to open a local menu. Choose the option *Targets, Groups, Files...* .



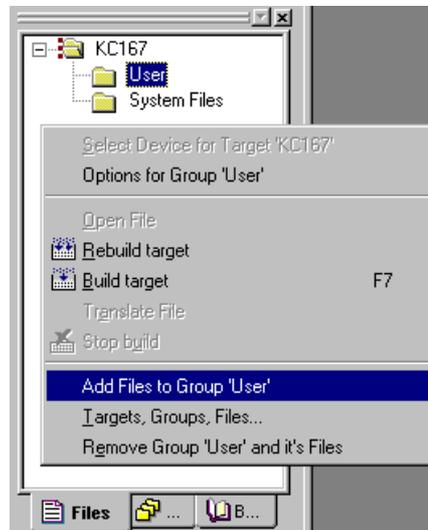
- Select the tab **Groups / Add Files** and type the new group name *System Files* in the *Group to Add:* section.



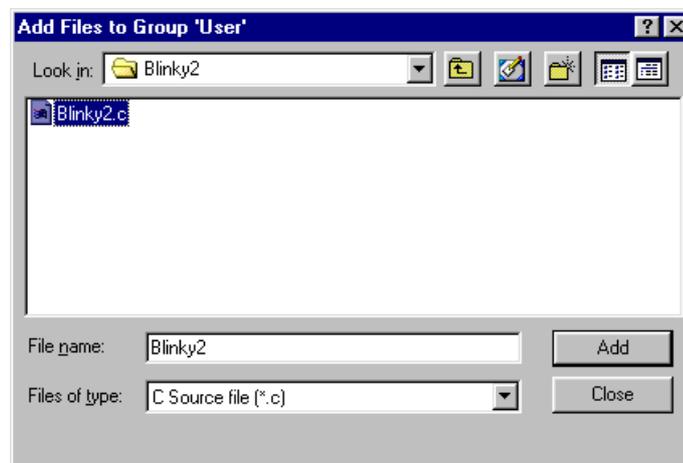
- Click on *Add* and then on *OK*.
- Your project file structure should now look like this:



- Now it's time to add some source code to our project. To do so, click with the right mouse key on the *User* group to open a local menu. The option *Add Files to Group 'User'* opens the standard files dialog.

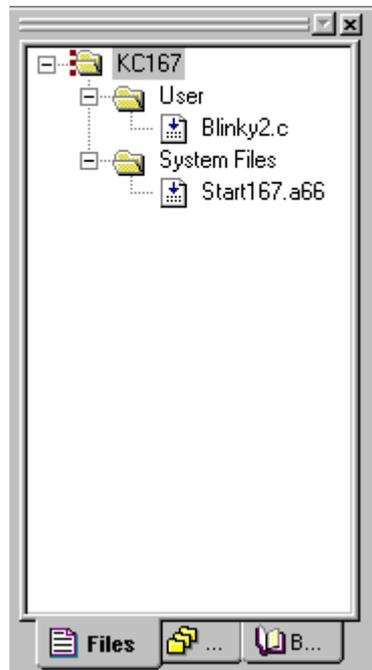


- Select the file *Blinky2.c*.



- Choosing *Add* adds the *Blinky2.c* file to your current project window.
- Close the window.

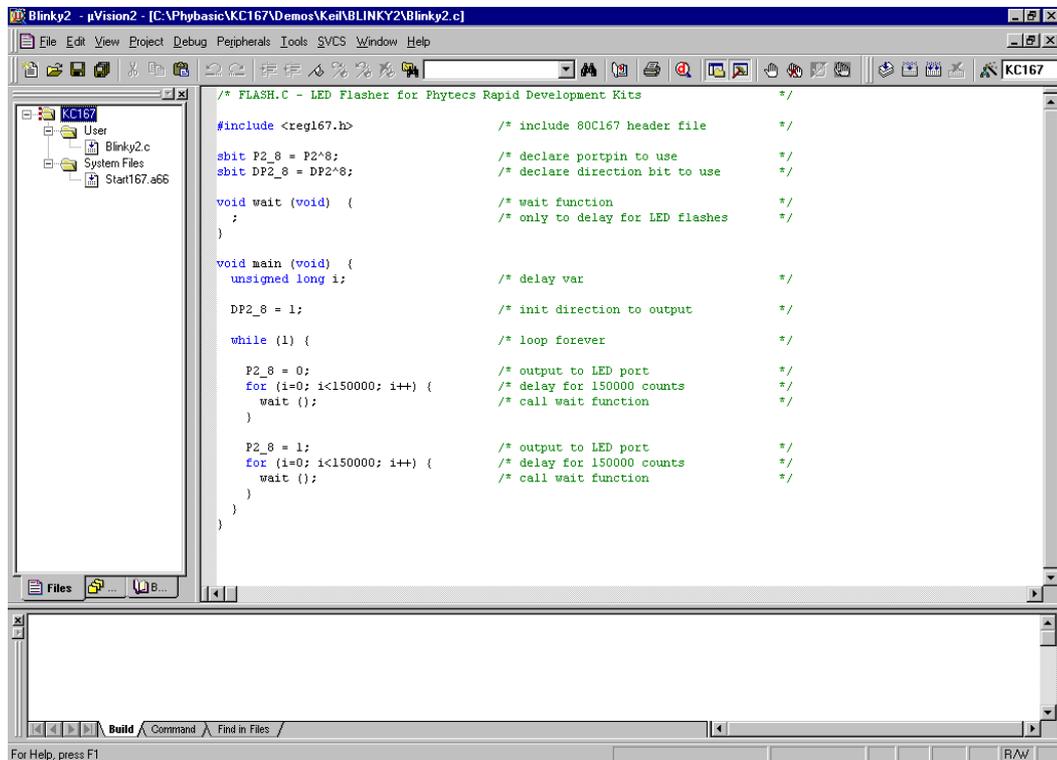
- Now right click on group *System Files* and add the File *start167.a66*. (You have to change the Filetype to “ASM source file \*.a” see this file).
- Your project window should now look like this:



At this point you have created a project called *blinky2.uv2* and added an existing C source file called *blinky2.c* and an existing Assembler file called *start167.a66*. 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

- Double click on *blinky2.c* to open it in the source code 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 */

P1_0 = 0; /* output to LED port */
for (i=0; i<225000; i++) { /* delay for 225.000 counts */
wait (); /* call wait function */
}

P1_0 = 1; /* output to LED port */
for (i=0; i<75000; i++) { /* delay for 75.000 counts */
wait (); /* call wait function */
}
}

```

### 3.4 Saving the Modifications

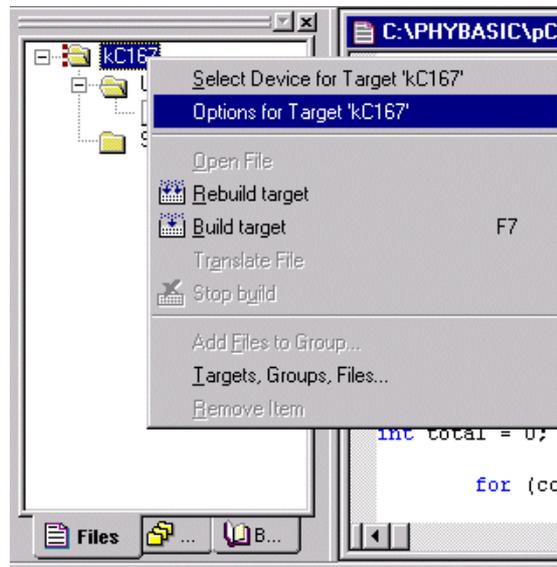
- Save the modified file by choosing *File/Save* or by clicking the floppy disk icon  .

### 3.5 Setting Options for Target

Keil includes a Make utility that can control compiling and linking source files in several programming languages. Before using the Make utility, macroassembler, C compiler or linker you must configure the corresponding options. Most of the options are set by specifying the device for the project. Only the external memory and output options must be set.

Enter the changes as indicated below and leave all other options set to their default values.  $\mu$ Vision2 allows you to set various options with mouse clicks and these are all saved in your project file.

- Click with the right mouse key in the “Project” window to open a local menu. Choose the option *Options for Target 'kC167'*.



**To configure the Target:**

- Open the *Project/Options for Target 'KC167'* menu and type the settings for the External Memory as shown below. Make sure that #1 is set to ROM.

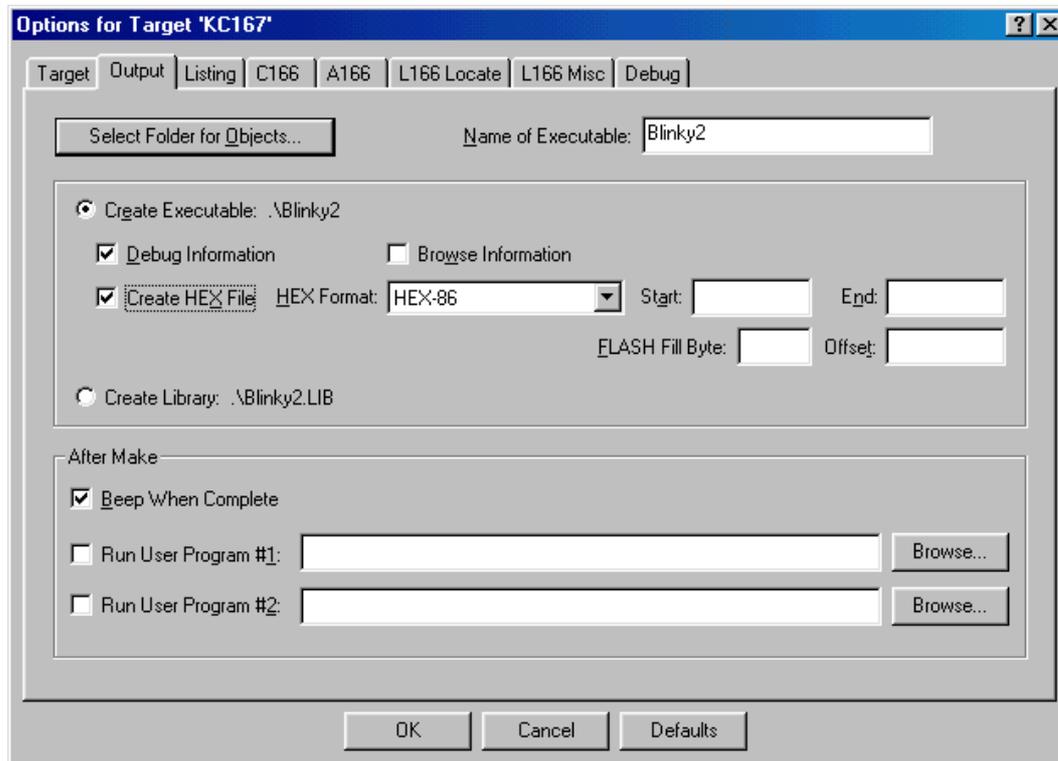
The screenshot shows the 'Options for Target 'KC167'' dialog box. The 'External Memory' section is the focus, with the following settings:

		Start:	Size:		Start:	Size:
#1:	ROM	0x0	0x3FFFF	#4:	RAM	
#2:	RAM	0x40000	0x0FFFF	#5:	RAM	
#3:	RAM			#6:	RAM	

Other settings visible in the dialog include: Clock (MHz): 20.0; Memory Model: Small: 'near' functions and data; Operating System: None; Data: near 6; Near Memory: 16 KB RAM, 16 KB ROM; and a checkbox for 'Use On-chip CAN+XRAM (0xE000 - 0xE7FF)' which is unchecked.

### To configure the Output options:

- Select the Output tab and activate the *Create HEX-File* option. With this option a *INTEL \*.HEX* file will be created for download.



- Click on *OK*

### 3.6 Building the Project

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

- Click on the *Build Target* icon  from the  $\mu$ Vision2 tool bar or press F7.
- If the program specified (*blinky2.c*) contains any errors, they will be shown in an error dialog box 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. This is shown in the Output Window, which indicates "*Blinky2*" - 0 Errors, 0 Warnings. The code to be downloaded to the board will be the name of the project with *.h86* as filename extension (in this case *blinky2.h86*).



```
Build target 'KC167'  
compiling Blinky2.c...  
assembling Start167.a66...  
linking...  
creating hex file from "blinky2"...  
"blinky2" - 0 Error(s), 0 Warning(s).
```

- 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

- Exit Keil  $\mu$ Vision2.
- Ensure that switch 1 of DIP switch S3 (left of the kitCON-Connector) is turned ON and push switch S1 to perform a Reset and to start the Bootstrap Loader on the kitCON-167.
- Start FlashTools 16W
- Select the correct module from the list and press the *Connect* button.
- Select the correct serial port for your host-PC and a 9600 baud rate.

- Click the *OK* button to load the module based part of the FlashTools to the target hardware.
- Returning to the FlashTools 16W Worksheet, choose the *Sector Utilities* tab, highlight all sectors and click on the *Erase Sector(s)* button. You can see the Flash sectors being erased at the bottom left hand 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\KC167\Demos\Keil\Blinky2\blinky2.h86* 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
- If the modified hexfile properly downloads and executes, the LED should now flash in a different mode with different on and off durations.

To start the demo program again execute the following steps:

- Turn **OFF** switch 1 of DIP-switch S3
- Press the Reset button (S1) on the Development Board to reset the target hardware and to start execution of the downloaded software

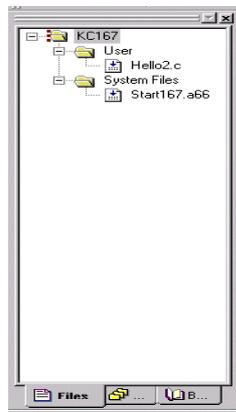
You have now modified source code, recompiled the code, created a modified download hexfile, and successfully executed this modified code.

## 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 3.2*.

### 3.8.1 Creating a New Project

- Start the Keil  $\mu$ Vision2 environment and close all projects that might be open.
- Open the Project menu and create a new project called *hello2.uv2* within the existing project directory  
*C:\PHYBasic\KC167\Demos\Keil\Hello2*  
(default location) on your hard-drive.
- Add *hello2.c* and *start167.a66* from within the project directory to the project *hello2.uv2*.
- Your project window should now look like this:



- Save the project

At this point you have created a project called *hello2.uv2* consisting of a C source file called *hello2.c* and an Assembler file called *start167.a66*.

### **3.8.2 Modifying the Example Source**

- Double click 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 Target Options**

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

- Exit Keil  $\mu$ Vision2.
- Ensure that switch 1 of DIP switch S3 (left of the kitCON-Connector) is turned ON and push switch S1 to perform a Reset and to start the Bootstrap Loader on the kitCON-167.
- Start FlashTools 16W
- Double click on KITCON
- Select the correct module from the list and press the *Connect* button.
- Select the correct serial port for your host-PC and a 9600 baud rate.
- Click the *OK* button to load the module based part of the FlashTools to the target hardware.
- Choose the *Sector Utilities* tab, highlight all sectors and click on the *Erase Sector(s)* button. You can see the Flash sectors being erased at the bottom left hand of the window. Wait until the status check in the lower left corner of the FlashTools 16W Worksheet finishes
- 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\KC167\Demos\Keil\Hello2\Hello2.h86* 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 16W.
- Turn **OFF** switch 1 of DIP-switch S3

### **3.8.6 Starting the Terminal Emulation Program**

- Start the HyperTerminal and connect 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.h86* file loaded into the Flash.
- Successful execution will send the modified character string "*PHYTEC... Stick It In!*" to the HyperTerminal window.
- Click the Disconnect icon 
- Close the Hyper Terminal program

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

## 4 Advanced User Information

This section provides advanced information for successful operation of the kitCON-167 with  $\mu$ Vision2.

### 4.1 FlashTools 16W

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 kitCON-167 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 kitCON-167 that, in turn, download user code into the Flash. Additionally, the re-programmable Flash device on the kitCON-167 enables easy update of user code and the target application in which the kitCON-167 has been implemented.

Currently the kitCON-167 can be populated with two different sized Flash devices: a 29F010 with 128 kByte or a 29F040 with 512 kByte. Since a Flash Bank always consists of two of such 8-bit devices to achieve 16-bit data width and there are two Flash banks, the kitCON-167 can be equipped with 256 kByte, 1 MB, 1.2 MB or 2 MB Flash.

The kitCON-167 is mounted with an external programmable device that functions as an address decoder. This can be used to derive all four used memory Chip Select signals used by the board from just one Chip Select signal supplied by the microcontroller (MODE = 0). Otherwise all four memory Chip Select signals must be supplied by the microcontroller using its internal Chip Select Unit (MODE = 1, default).

We recommend using the microcontroller's internal Chip Select Unit because of its high degree of flexible configuration. This is the default mode of the kitCON-167. FlashTools 16W will not properly function when using the external address decoder to derive the Chip Select signals.

The current version of FlashTools 16W does not support contiguous programming of both Flash banks. The standard FlashTools version is exclusively suitable to program the Flash bank 1.

FlashTools 16W always uses the Bootstrap mode to transfer the required microcontroller firmware to the kitCON-167. Hence, there is no restriction in using the Flash-Memory for storing user code.

Before Flash programming FlashTools 16W 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 Bank 1 using /CS0 addressable at 000000H-0FFFFFFH (up to 1 MB Flash)
- RAM Bank 1 using /CS1 addressable at 100000H-1FFFFFFH (up to 1 MB RAM)

Since there is no additional address translation, FlashTools 16W will download any hexfile to Flash bank 1 using its original addresses.

Using the Bootstrap mode to transfer the required microcontroller firmware to the kitCON-167 ensures that FlashTools 16W maintains its independent Flash programming capability.

## 4.2 Start167.a66

The code within the assembly file *start167.a66* 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.

It is very important that this code will execute prior to the execution of user code. To ensure this, it is recommended that the startup code occupy 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.

Since some of the settings are hardware-dependent, we recommend use of the prepared *start167.a66* from within the default location *C:\PHYBasic\KC167\Tools\Startup\Keil*. The properties of the external bus interface are already configured for the kitCON-167. You may want to change the values for the Chip Select Unit.

To accommodate the startup code to the needs of your application copy it from the directory described above to your project directory. You can then edit, modify and compile it using the Keil macroassembler.

Since the startup code will usually be implicitly taken into consideration from the default runtime libraries, you must now explicitly tell your linker to instead consider your individual startup object file. To do this we recommend adding your modified *start167.a66* file to your project. Be sure that it is always included in the Link/Lib process of your project (see how to add sourcecode to your project of  $\mu$ Vision2 in *section 3.2*).

The Keil tool chain collects all initialization information for pre-initialized variables (e.g. `int i=3;`) during compile and link time. This information is stored within a special segment in the form of a list of binary data entries. To mark the end of this list,  $\mu$ Vision2 adds a special entry to the list. Whenever the startup code finds this special entry it ends the initialization of variables and continues executing. Note that this end of list entry comes from the ***start167.obj*** object file. To ensure consideration of all of your pre-initialized variables, you must move ***start167.obj*** to the end of the list of your link files. Otherwise some of your pre-initialized variables may not be initialized by the startup code during runtime. It is usually very difficult to detect such a mistake.

We recommend that you always move ***start167.a66*** to the end of your projects file list.

### 4.3 Linking and Locating

The Linker has to combine several re-locatable 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.

To enable easy accommodation of the linking process the Linker collects segments of equal type to classes.

The Keil tool chain distinguishes the following classes:

- *x*CODE: code for several addressing modes  
(*x* = N, I, F, H or X)
- *x*DATA: not initialized data for several addressing modes  
(*x* = N, I, F, H or X)
- *x*DATA0: pre-initialized data for several addressing modes  
(*x* = N, I, F, H or X)
- *x*CONST: constant for several addressing modes  
(*x* = N, I, F, H or X)

It is required that all *x*DATA and *x*DATA0 classes segments are located to any internal RAM of the C167 or to any external RAM of the kitCON-167.

All *xCODE* and *xCONST* classes must also be located to any internal non-volatile memory (e.g. Flash, OTPROM) of the C167 or any external Flash-Memory of the kitCON-167.

A near address data area (NDATA, NDATA0) must reside in one data page (16 kByte). A near address code area (NCODE, NCONST) must reside in one code segment (64 kByte).

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 where to assume external RAM for locating data classes and Flash for locating code and constant classes.

The recommended operating mode of the kitCON-167 allows the use of the Chip Select Unit of the C167 to define the physical memory layout. By modifying the file *start167.a66* as part of your project you can adapt the memory layout to your needs.

The external use of the Chip Select signals is predefined by the hardware in the following way:

- Flash Bank 1 uses /CS0 (up to 1 MB Flash)
- Flash Bank 2 uses /CS3 (up to 1 MB Flash)
- RAM Bank 1 uses /CS1 (up to 1 MB RAM)
- RAM Bank 2 uses /CS2 (up to 1 MB RAM)

The default configuration of the kitCON-167 has 256 kByte Flash at Bank 1 (/CS0) and 256 kByte RAM at Bank 1 (/CS1).

Note that you will see multiple mirrors of a memory device that has a physical smaller address range than the associated address range of the Chip Select signal.

For instance if you adjust Chip Select Signal /CS1 to be active within an address range of 1 MB and the actually memory size populating the kitCON is just 256 kByte, you will get three mirrors of your RAM.

We recommend that you generate a *\*.m66* 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 *start167.a66* file.



---

**Document: kitCON-167 QuickStart Instructions**  
**Document number: L-537e\_1, January 2001**

---

**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 2001

Ordering No. L-537e\_1  
Printed in Germany