microMODUL-8051

QuickStart Instructions

Using PHYTEC FlashTools98 for Windows and the Keil EK51 Software Evaluation Development Tool Chain

Note: The PHYTEC Spectrum CD includes the electronic version of the English microMODUL-8051 Hardware Manual

Hinweis: Die PHYTEC Spectrum CD beinhaltet die elektronische Version des deutschen microMODUL-8051 Hardware Manuals

Edition: May 2000

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.

	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

3rd. Edition: May 2000

Table of Contents

Figures

1 Introduction to the Rapid Development Kit

This tutorial provides:

- general information on the PHYTEC microMODUL-8051 Single Board Computer,
- an overview of Keil's PK51 software evaluation development tool chain, and
- instructions on how to run example programs on the microMODUL-8051, mounted on the PHYTEC Development Board, in conjunction with Keil tools.

Please refer to the microMODUL-8051 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 microMODUL-8051 Hardware Manual.

1.1 Documentation Overview

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

- the PHYTEC microMODUL-8051 Hardware Manual and Development Board Hardware Manual
- controller User's Manuals and Data Sheets
- this QuickStart Instructions with general "Rapid Development Kit" description, software installation hints and three example programs enabling quick out-of-the box start-up of the microMODUL-8051 in conjunction with the Keil software development tools.
  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 FlashTools98 for Windows.
  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 microMODUL-8051 using the Windows-based Keil tools.

3) The Debugging section provides a third example program – "Debug" - to demonstrate monitoring of the board and simple debug functions using the Keil debug environment.

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

1.2 System Requirements

Use of this "Rapid Development Kit" requires:

- the PHYTEC microMODUL-8051,
- the PHYTEC Development Board with the included DB-9 serial cable and power supply (8-13 V / min 250 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:

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

http://www.keil.com
support@keil.com

1.3 The PHYTEC microMODUL-8051

The microMODUL-8051 represents an affordable yet highly functional Single Board Computer (SBC) solution in a minimal formfactor (51 x 36 mm). Standard versions of the microMODUL-8051 feature the following controllers:

- the Infineon SAB C501 or Dallas 80C320 controllers, both of which offer up to 3.3 times the speed of the standard 80C51;
- the Infineon SAB C504, featuring an 8-channel on-chip A/D converter with 10-bit resolution;
- the SST89C58FlashFlex51 controller, boasting 36 kByte on-chip Flash; and
- the Oki MSM80C154, boasting a lower power draw

Upon request, the microMODUL-8051 can be populated by any 8051 or 80C52/80C32-compatible controllers in a PLCC44- or QFP44-package. All applicable data/address lines and applicable signals extend from the underlying logic devices to standard-width (2.54 mm/0.10 in) pin headers lining the circuit board edges. This enables the microMODUL-8051 to be plugged like a “big chip” into target hardware.

The standard memory configurations of the microMODUL-8051 features 32 (to 128) kByte external SRAM and 128 (to 512) kByte external Flash for storage of user code. The Flash device allows direct on-board programming. Free chip-select signals are available for external I/O connectivity. The module communicates by means of an RS-232 transceiver and operates within a standard industrial range of 0 to +70 degrees C and requires only a 250 mA power source. PHYTEC FlashTools98 enables easy on-board download of user programs.

microMODUL-8051 Technical Highlights

- SBC in matchbox-sized dimension (51 x 36 mm) achieved through advanced SMD technology
- fitted with a 44-pin (PLCC or QFP-packaged) Infineon SAB-C5xx, Dallas 8xC32x, SST895x, Oki MSM80C154 or other 8051-derivative controllers
- up to 40 MHz. clock speed
- 32 (to 128) kByte external SRAM
- 128 (to 512) kByte external Flash supporting on-board downloading of user code from a host-PC in conjunction with PHYTEC FlashTools98 firmware
- RS-232 serial interface (optionally configurable as RS-485)
- three Chip-Select signals for external I/O devices
- optional 8-channel, 10-bit A/D-converter (if mounted with the C504)
- requires only +5V/250 mA power source
- operates in a temperature range of 0... 70° C (optional -40... 85° C temperature range available)

The PHYTEC Development Board, in EURO-card dimensions (160 x 100 mm), is fully equipped with all mechanical and electrical components necessary for the speedy and secure insertion and subsequent programming of most PHYTEC microMODUL and miniMODUL series Single Board Computers. Simple jumper configuration readies the Development Board's connectivity to any PHYTEC module, which plug pins-down into the contact strips mounted on the Development Board. One set of strips accommodates microMODULs, the other miniMODULs.

Development Board Technical Highlights

- a RESET-switch
- a BOOT-switch
- a low voltage socket and voltage regulator accepting an un-regulated input voltage in a range from 8 to 13 VDC.
- a DB-9 connector (configured as an RS-232 interface)
- a second DB-9 connector configurable as an RS-485, CAN or a second RS-232 interface depending on the implemented Single Board Computer and user needs
- a VG96-connector
- simple jumper configuration allowing use of the Development Board with various PHYTEC Single Board Computers
- a wire wrap field (60 x 60 mm/2.4 x 2.4 in) supporting development of user-designed peripheral hardware.

1.4 The Keil EK51 Software Evaluation Development Tool Chain

The professional (PK51) and evaluation (EK) versions of the Keil software development tool chains fully support the entire 8051 and 8051-derivative microcontroller family, including the Infineon C500 family. The Keil tool chain consists of the following executables:

- **µVision** uvw51e.exe (GUI shell for DOS-level programs)
- **C Compiler** C51.exe (C compiler)
- **Assembler** A51.exe (macroassembler)
- **Linker** bl51.exe (code-banking linker/locator)
- **Converter** oh51.exe (object-to-hex converter)

Once installed, the default destination location for these files is the ***c:\c51eval\bin*** directory for the EK evaluation version. If using the professional (PK) version of the Keil tool chain, the default destination location for these files is the ***c:\c51\bin*** directory. They are DOS-level programs that can be accessed from Windows via the µVision GUI-shell. The entire tool chain can be run from µVision or directly from DOS. The Keil tools support all in-circuit emulators that adhere to the Infineon OMF51 debugging specification. The Keil OH51 Object-to-Hex Converter converts an absolute object file into an Intel-HEX file that is suitable for programming into an EPROM device or downloading into external Flash on the PHYTEC microMODUL-8051 target board. The evaluation version of the Keil tool chain is restricted to a manipulable code size of 2 kByte. In addition the code will automatically be located at 0x4000 in order to prevent unauthorized use of this version for programming of common devices with internal code memory less or equal than 2 kByte. Other than these restrictions, the evaluation tool chain functions exactly as the full version does, enabling full evaluation of the features and functionality of Keil development tools. The full version has no such restrictions, both are fully ANSI compliant.

µVision IDE

µVision is a Windows based front-end Graphical User Interface (GUI) 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 µVision, including emulator software.

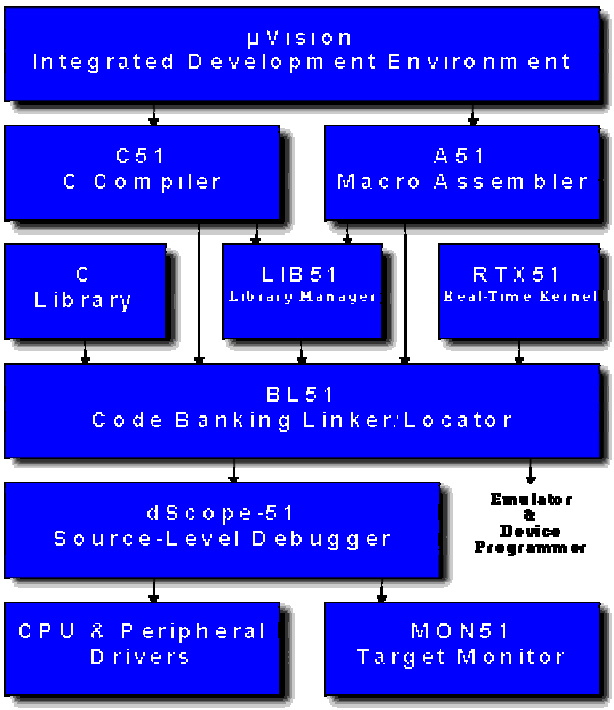


Figure 1: Keil Tool Chain Overview

C51 C Compiler

The C51 ANSI compiler with the A51 assembler are both specifically designed for the 8051 microcontroller family. The compiler can be run in either DOS mode or started from the Windows-based µVision.

A51 Macro Assembler

The full (PK) Keil macroassembler is available with the Keil compiler package or as a stand-alone tool. The assembler is DOS-based or can be run from within µVision.

BL51 Code Banking Linker/Locator

The BL51 linker/locator is used to join relocateable object modules and library files together and to locate them to fixed memory locations. It supports Code Banking for easy management of multiple code-banks to allow applications with more than 64 kByte of code. The library and object files created with the C51 compiler or the A51 assembler are provided by Keil. This process results in absolute object modules. The BL51 is DOS-based and can be run within DOS or Windows using µVision. A ****.map***-file can be produced, giving details of the memory structure. The object file may be specified to contain debugging information as required by simulators, debuggers and emulators. The BL51 also supports an overlay mechanism for variables to optimize the use of the size-restricted internal RAM.

OH51 Object Hex Converter

The Keil OH51 object-to-hex converter transforms an absolute object file produced by the BL51 Banking Linker into a standard Intel ****.hex***-file. This file is suitable as an input to the PHYTEC FlashTools98 or for programming into an EPROM or an emulator. OH51 is a DOS file and can be run by µVision and started with suitable arguments.

Debug Environment

dScope is 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, dScope becomes tScope. dScope and tScope enable the following debugging functions:

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

dScope has a performance analysis feature to ensure your code runs efficiently. In addition, dScope has a disassembler/assembler that allows the modification of user code without recompiling.

2 Getting Started

What you will learn with this Getting Started example:

- installing Rapid Development Kit software
- starting PHYTEC's FlashTools98 for Windows download utility
- interfacing the microMODUL-8051, mounted on the Development Board, to a host-PC
- downloading example user code in hex-file format from a host-PC to the external Flash memory using FlashTools98

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 folder of the PHYTEC Spectrum CD.

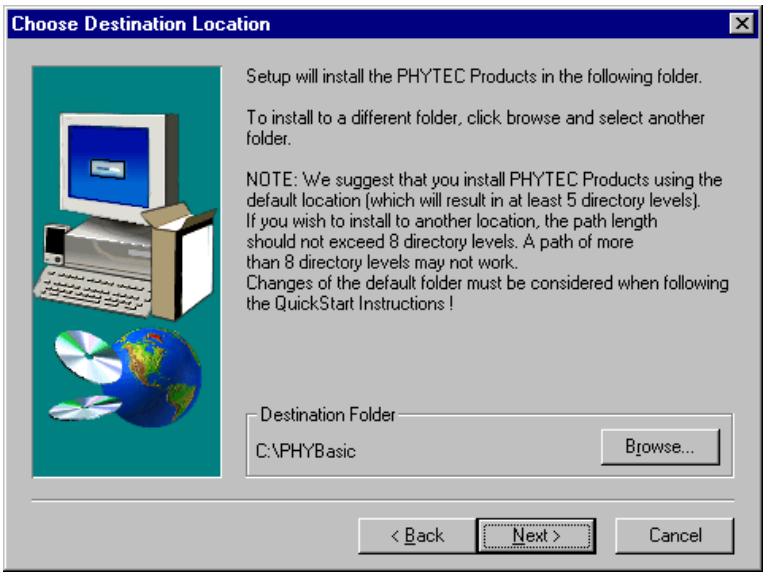
The following window appears:

Choose *Install Basic Product Files* Button.

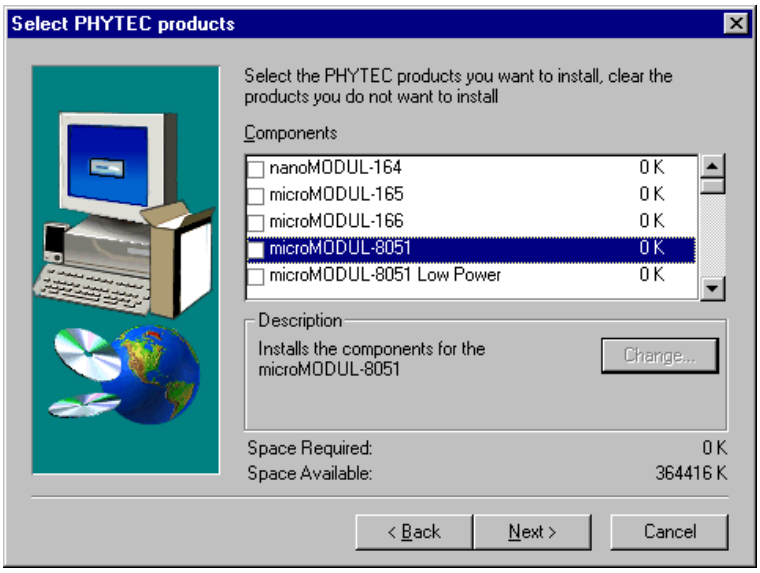
After accepting the Welcome window and license agreement select the destination location for installation of Rapid Development Kit software and documentation.

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.



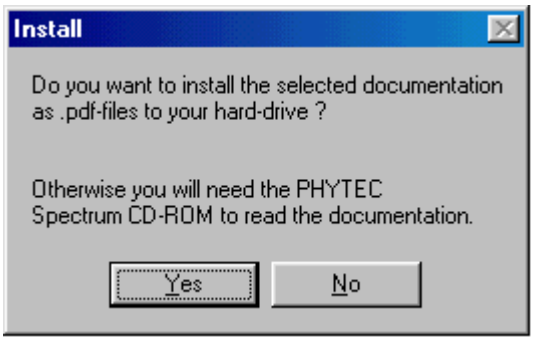
In the next window select your Rapid Development Kit of choice from the list of available products. By using the *Change* button, advanced users can select in detail which options should be installed for a specific product.



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

All software and tools for this microMODUL-8051 Kit will be installed to the ***\PHYBasic*** folder 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 documentation. On the Spectrum CD.

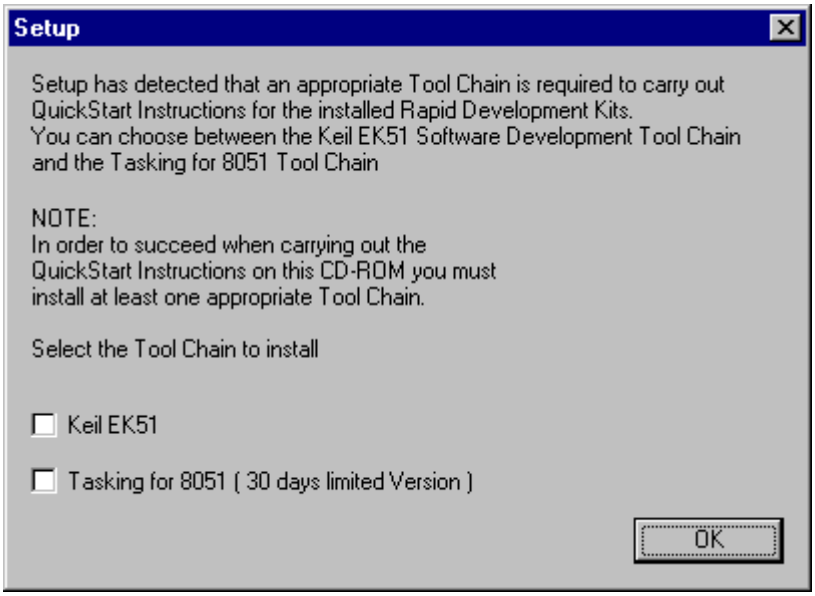


If you decide **not** to copy the documentation to your hard-drive you will need the PHYTEC Spectrum CD-ROM each time you want to access these documents. The installed links will refer to your CD-ROM drive in this case.

If you decide to copy the electronic documentation to your hard-drive, the documentation for this microMODUL-8051 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\DevBmM***) 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 the Keil EK51 Software Development Tool Chain.



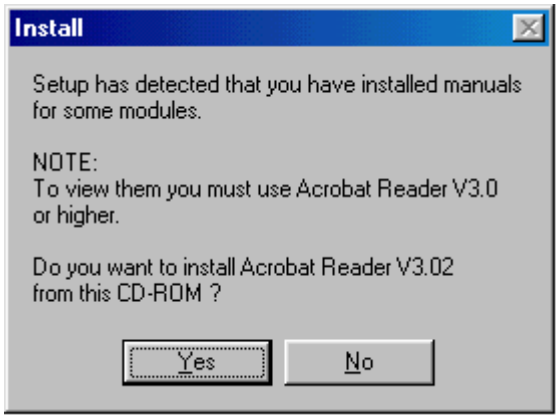
After accepting the Welcome window and license agreement select the destination location for installation of the Development Tool Chain.

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

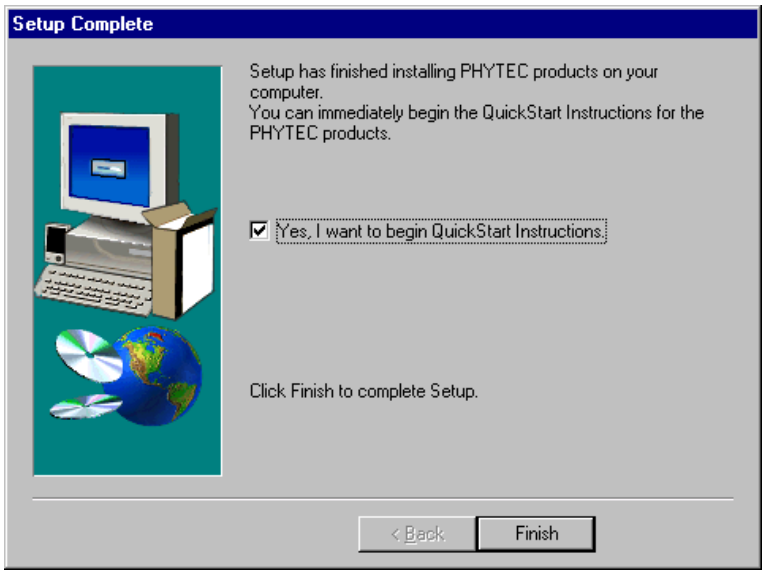
The applicable Keil tool chain must be installed to ensure successful completion of this QuickStart Instruction. Failure to install the proper software could lead to possible version conflicts, resulting in functional problems.

We recommend that you install the Keil tool chain from the Spectrum CD-ROM even if other versions 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 FlashTools98 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 microMODUL-8051 to a Host-PC

Connecting the microMODUL-8051, mounted on the PHYTEC Development Board, to your computer is simple:

- As shown in the figure below, if the microMODUL is not already pre-installed, mount it pins-down onto the Development Board's interior receptacle footprint (X2).

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

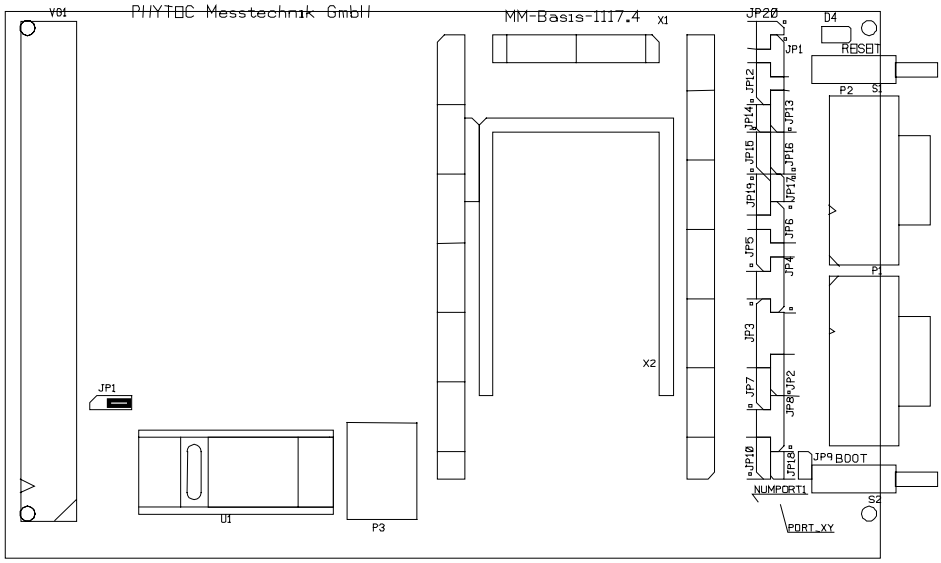


Figure 2: Development Board Overview

- 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.

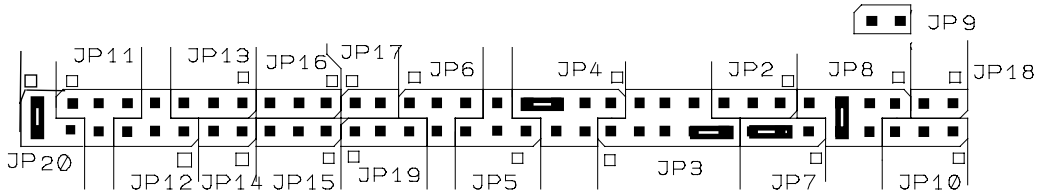


Figure 3: Suitable Development Board Jumper Settings

- Configure Jumper JP1 on the Development Board as indicated 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 included serial cable.
- Using the included power adapter, connect the power socket on the board (P3) to a power supply (*refer to Figure 4 for the right polarity*).

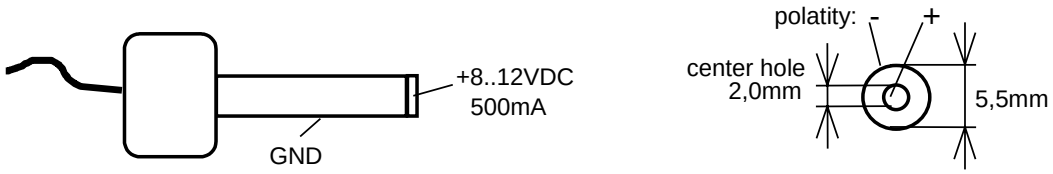


Figure 4: Power Connector

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

This sequence of pressing and releasing the Reset (S1) and Boot (S2) switches always renders the microMODUL-8051 into the Flash Programming Mode. Use of FlashTools98 requires that the microMODUL-8051 be in Flash Programming mode. Ensure that FlashTools are only started after rendering the microMODUL-8051 in this mode. See *section 5.1* FlashTools98 for more details.

The microMODUL should now be properly connected via the Development Board to a host-PC and power supply. You are now ready to program the microMODUL-8051, as mounted on the Development Board. This microMODUL/Development Board combination shall also be referred to as “target hardware”.

2.3 Starting PHYTEC FlashTools98 for Windows

FlashTools98 should have been installed during the initial setup procedure as described in *section 2.1*.

Alternately, you can manually install FlashTool98 by executing ***setup.exe*** from within the ***\Software\Flasht98*** directory of your PHYTEC Spectrum CD.

FlashTools98 for Windows 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.

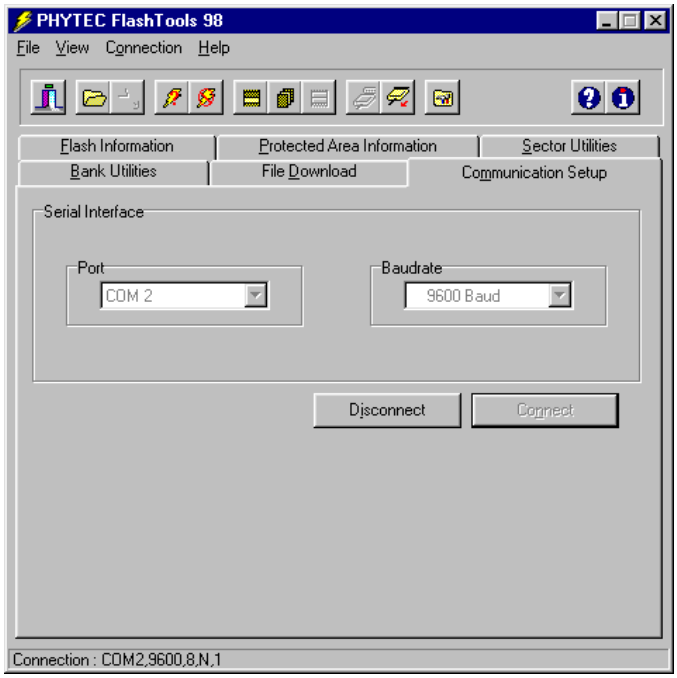
FlashTools98 consists of a firmware resident in the external Flash and corresponding software installed in the host-PC. Proper connection of a PHYTEC Single Board Computer (SBC) to a host-PC enables the software portion of FlashTools98 to recognize and communicate to the firmware portion.

- You can start FlashTools98 by selecting it from the *Programs* menu using the Windows *Start* button.

It is recommended that you drag the FlashTools98 icon onto the desktop of your PC. This enables easy start of FlashTools98 by double-clicking on the icon.

2.4 Downloading Example Code with FlashTools

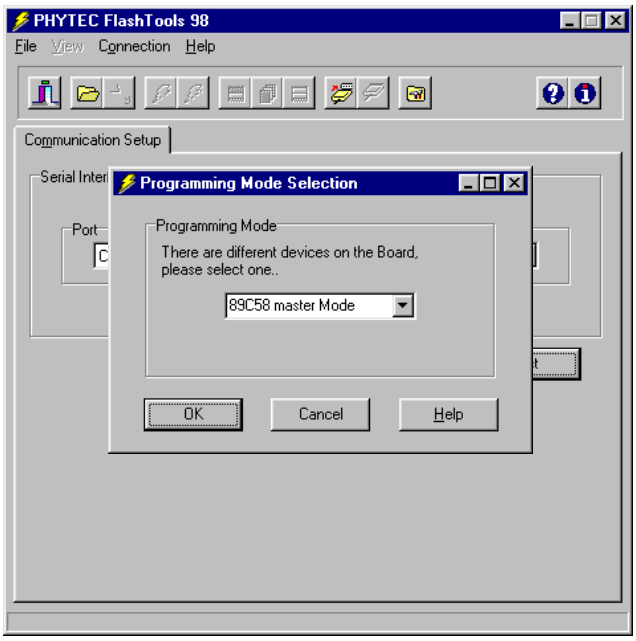
- Start FlashTools98 for Windows by double-clicking on the Flash-Tools98 icon or by selecting *FlashTools98* from within the ***Programs|PHYTEC FlashTools98*** program group.
- The Communication Setup tab of the FlashTools98 Worksheet window will now appear. Here you can specify connection properties to the microMODUL-8051.



- Choose the correct serial port for your host-PC and a 9,600 baud-rate.
- Click the *Connect* button to establish connection to the target hardware.

The microcontroller firmware 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 Flash Programming Mode as described in *section 2.1*.

- **Optional:** if FlashTools98 detects multiple Flash devices on the target hardware (e.g. controller derivatives with internal Flash) the following *Programming Mode Selection* dialog window will appear:



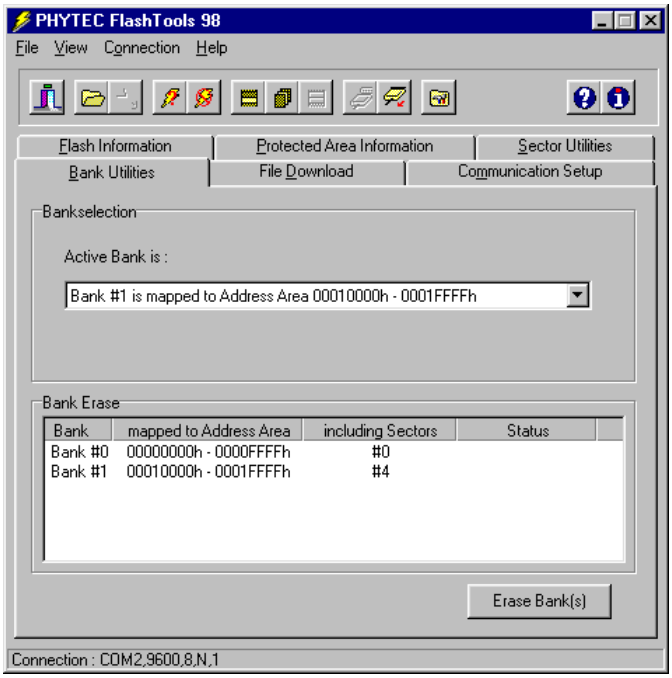
- If the *Programming Mode Selection* window appears choose *xxx external Flash* where *xxx* describes the Flash device currently mounted on the microMODUL-8051.

Some 8051 controller derivatives used on the microMODUL-8051 offer on-chip Flash. An appropriate Programming Mode option will only appear if on-chip memory is detected by the FlashTools98 firmware. For this example, it is assumed that no on-chip Flash is available on the controller populating the microMODUL-8051.

- Click on the *OK* button to select download of code to the external memory.

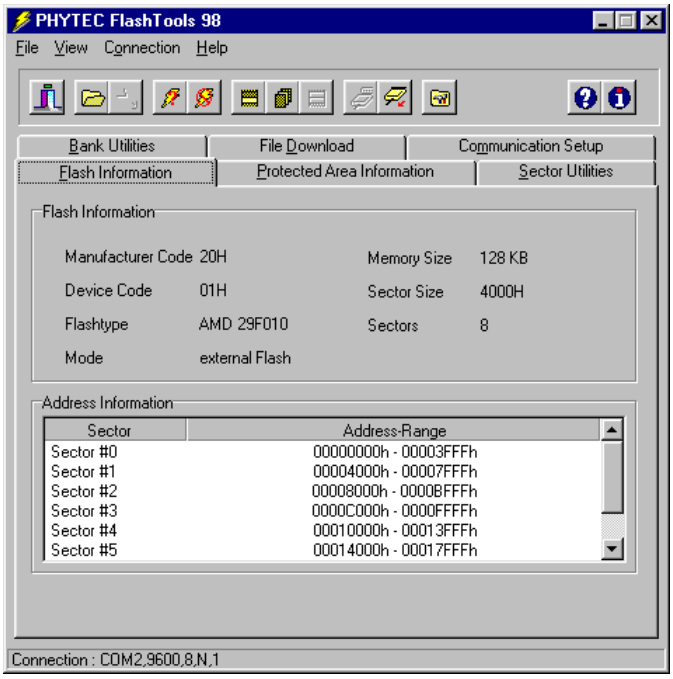
Returning to the FlashTool98 Worksheet window, you will see tabs for the following:

Bank Utilities enable erasure and status check of whole banks of memory specified by the user:

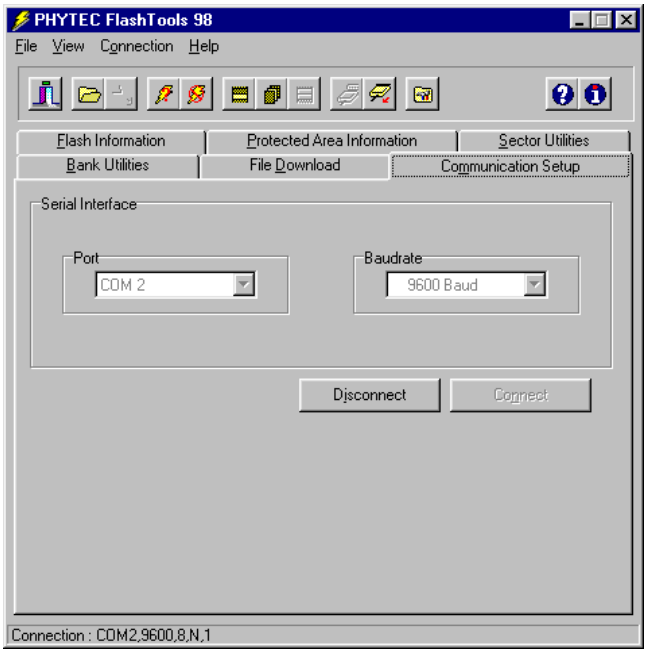


This tab is called *Bank Utilities* if external Flash has been selected in the Programming Mode window. If internal Flash has been selected for a controller supporting on-chip Flash, this tab is called *Block Utilities*. These tabs enable erasure of memory banks and blocks, respectively. The *Block Utilities* also supports chip erase of the entire on-chip Flash.

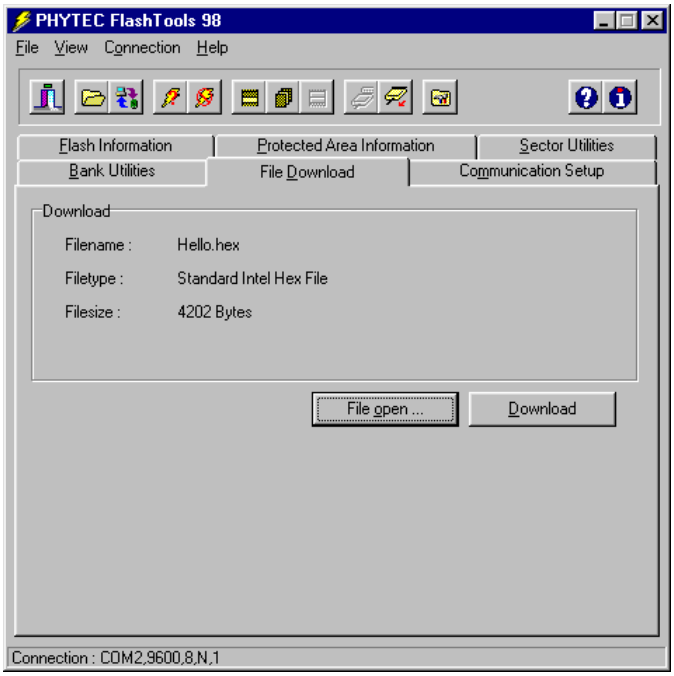
Flash Information shows sector and address ranges in Flash memory:



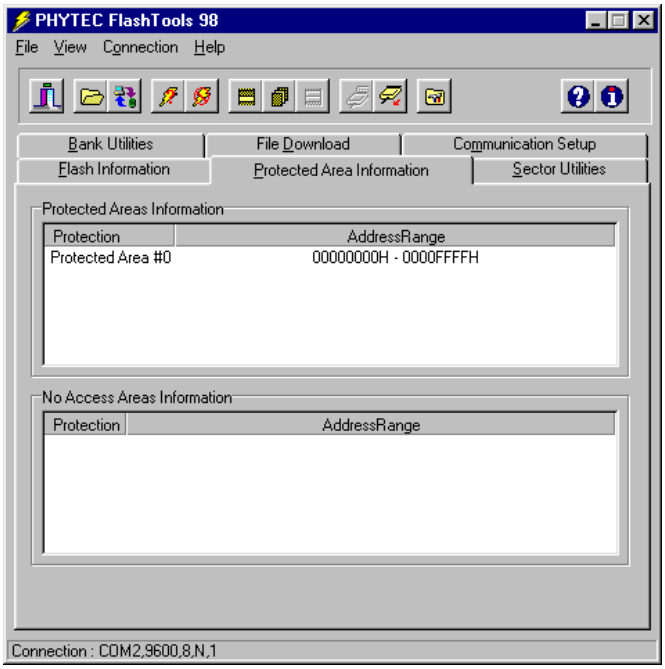
Communication Setup allows selection of the serial port and speed (this is the same window that was used when you first entered Flash-Tools98 for Windows):



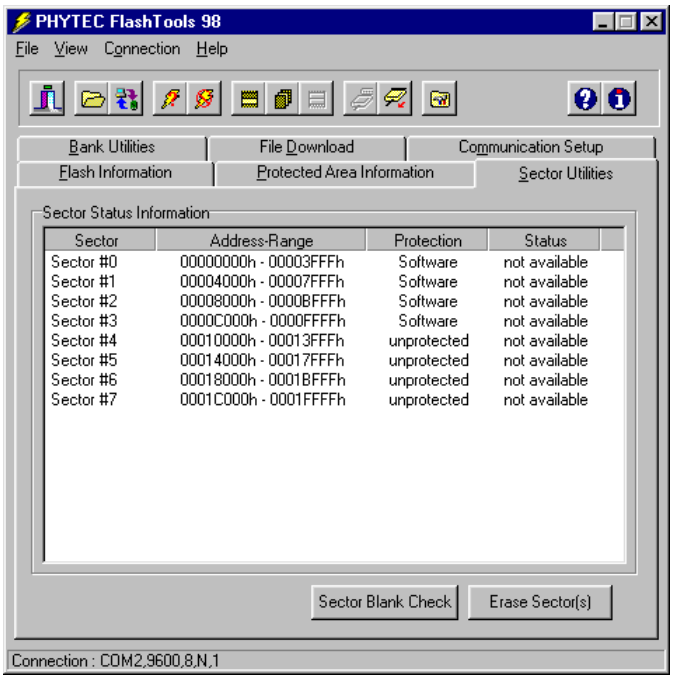
File Download downloads specified hex-files to the target hardware:



Protected Areas Information shows protected areas of Flash memory:



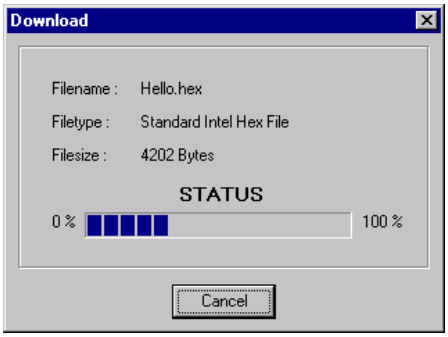
Sector Utilities allow erasure of individual sectors of Flash memory:



2.4.1 "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 *Bank Utilities* tab, highlight *Bank #1* within the *Bank Erase* section, and click on the *Erase Bank(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 microMODUL-8051 Demo directory (default location ***C:\PHYBasic\uM8051\Demos\Keil\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 Flash-Tools98 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 *Bank Erase* tab from the FlashTools98 worksheet, highlight bank 1 and erase the bank. Then repeat the download procedure.

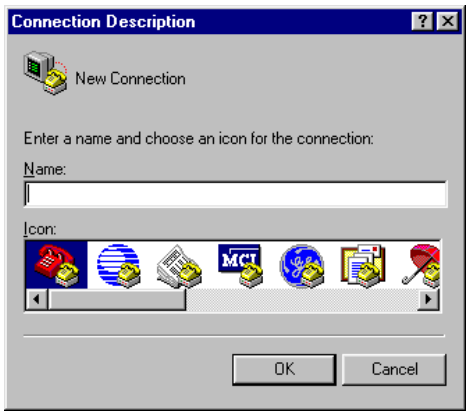
- Returning to the *Communication Setup* tab, click on the *Disconnect* button.

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

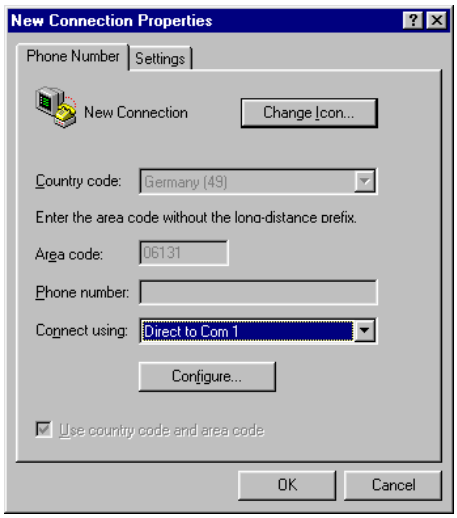
- Start the HyperTerminal program within the *Programs|Accessories* bar.
- The HyperTerminal main window will now appear:

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

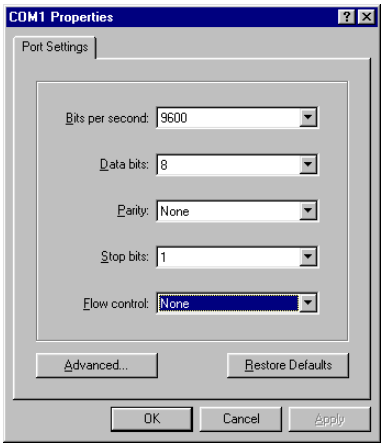
- The Connection Description window will now appear. Enter “COM Direct” in the “Name” combobox (be sure to specify the correct COM parameter for your system).



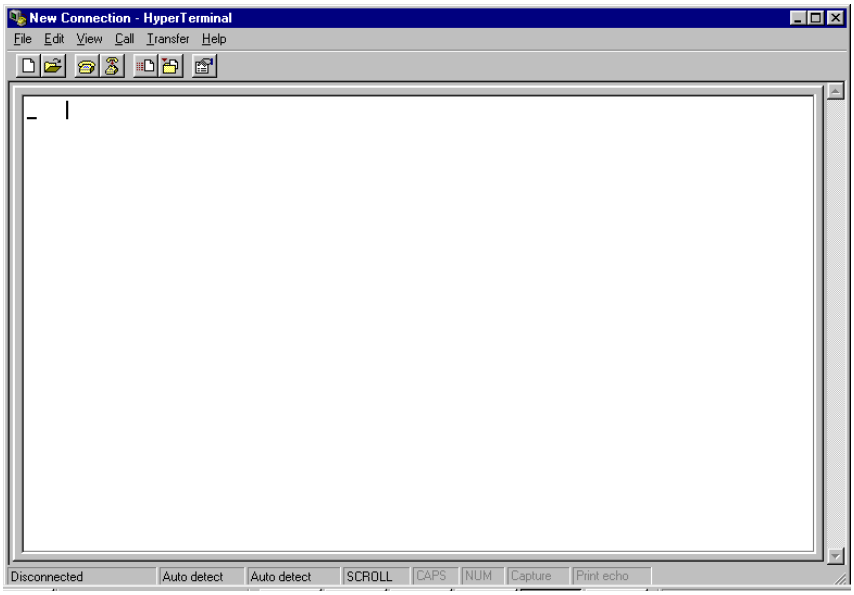
- Next press the *Ok* button. This creates a new HyperTerminal session named “COM Direct” and advances you to the next HyperTerminal window.
- The *New Connections Properties* window will now appear. Specify *Direct to COM1* under the *Connect Using* combobox (be sure to indicate the correct COM setting 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) and pressing the <Space> bar on your keyboard will execute the ***hello.hex*** file loaded into the Flash. Pressing the <Space> bar allows the micro-MODUL-8051 to detect the baud rate for connection to the target hardware.
- Successful execution will send the character string "*Hello World*" from the target hardware to the HyperTerminal window.

Pressing any other key than the <Space> bar leads to an improper baud rate since the automatic baud rate detection is based on the timing measurement during the transmission of a well known character – the <Space> character. As a result you may get incoherent characters in 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 and the RS-232 connection.

2.4.2 "Blinky"

The "Blinky" example sends a program to the Flash that, when executed, manipulates the single user SMD-LED D3 on the Development Board which is mounted near the Boot (S2) switch. This second example program provides a review of the FlashTools98 download procedure:

- 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 (S1) and Boot (S2) switches on the Development Board and then releasing first the Reset (S1) and, two to three seconds later, release the Boot (S2) switch
- Start FlashTools98 for Windows

- At the Serial Interface tab of the FlashTools98 Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Connect* button to establish connection to the target hardware
- If the optional Programming Mode Selection window appears (only for controllers with on-chip Flash), choose *External Flash Memory*
- Returning to the FlashTools98 Worksheet, choose the *Bank Utilities* tab, highlight *Bank #1* in the *Bank Erase* section of the tab and click on the *Erase Bank(s)* button to erase this memory bank
- Wait until the status check in the lower left corner of the FlashTools98 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\uM8051\Demos\Keil\Blinky\blinky.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 to reset the target hardware and to start execution of the downloaded software
- 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 ****.hex***-file format.

3 Getting More Involved

What you will learn with this example:

- how to start the Keil tool chain
- how to configure the Keil tools within the µVision IDE (Integrated Development Environment)
- 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

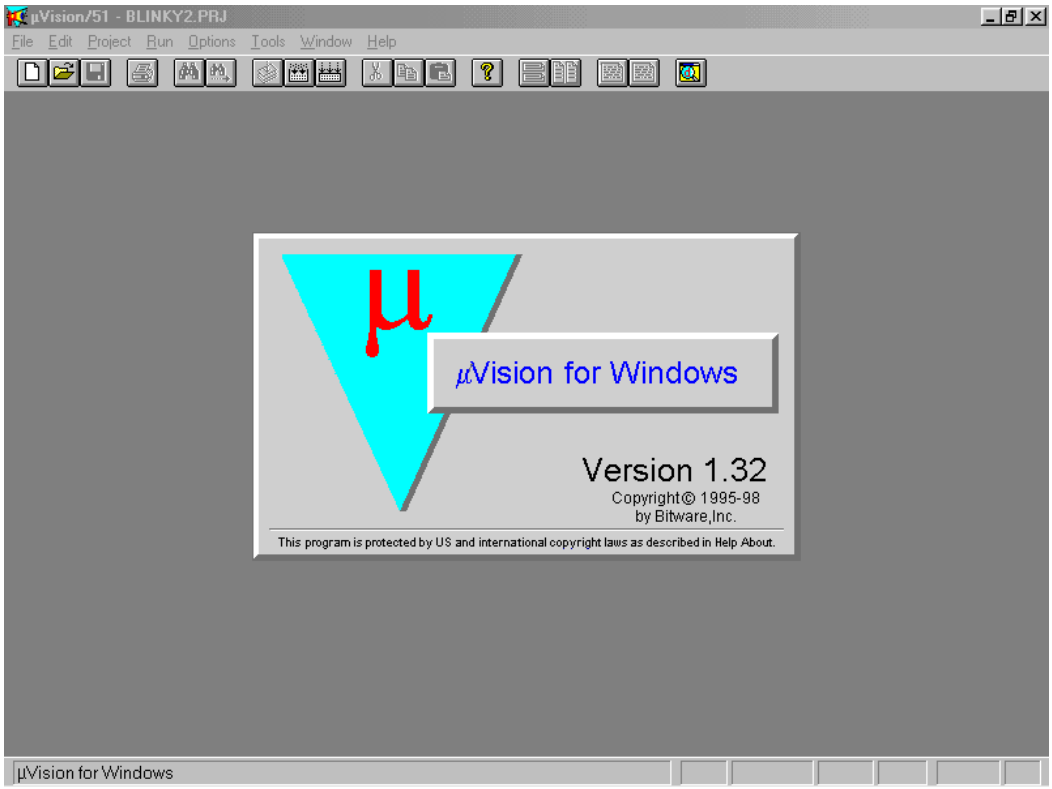
3.1 Starting the Keil Tool Chain

The Keil EK51 software development tool chain 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\Keil\Ek51*** directory of your PHYTEC Spectrum CD.

Start the tool chain by clicking *µVision-51* from within the ***Programs|Keil PK51 – Eval*** program group.

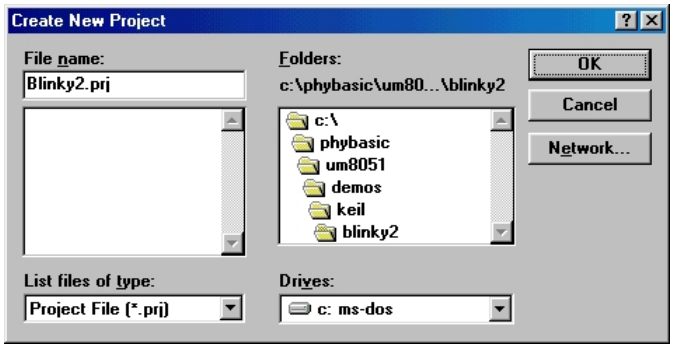
After you start µVision, the window shown below appears. From this window you can create projects, edit files, configure tools, assemble, link and start the debugger. Other 3rd party tools such as emulators can also be started from here.



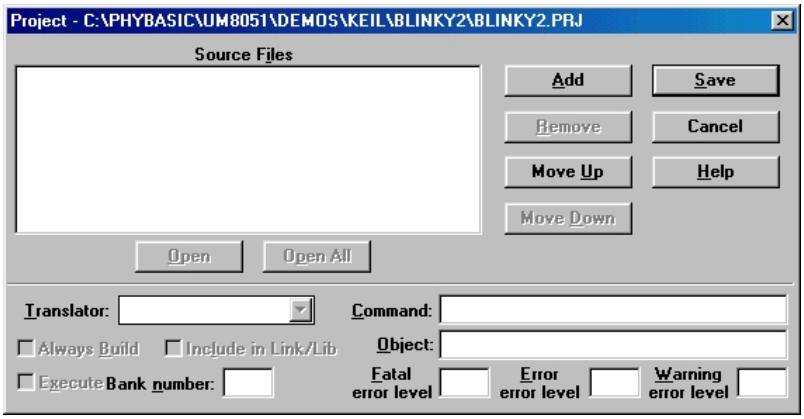
3.2 Creating a New Project and Adding an Existing Source File

- Open the *Project* menu and choose *New Project*. The window shown below appears.
- Change to the project directory created by the installation procedure (default location ***C:\PHYBasic\uM8051\Demos\Keil\Blinky2***).

- Enter the file name of the project you are creating. For this tutorial, enter the name ***Blinky2.prj*** and press *Ok*

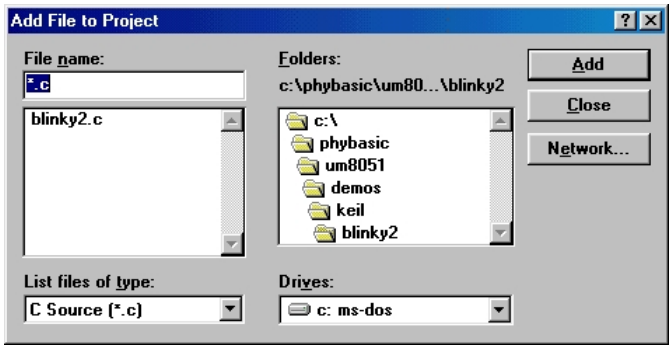


The following window will appear:



This window is used to add various files to your project. These include ASCII files, C and assembly source and macros and various other file types. The list is quite extensive and is found in the Translator combobox as soon as you have added any file to the project and selected it within the Source Files window. Note that this combobox is blanked out at this time. This project window is accessible at any time by selecting *Project|Edit Project*, which enables easy edit of your file list.

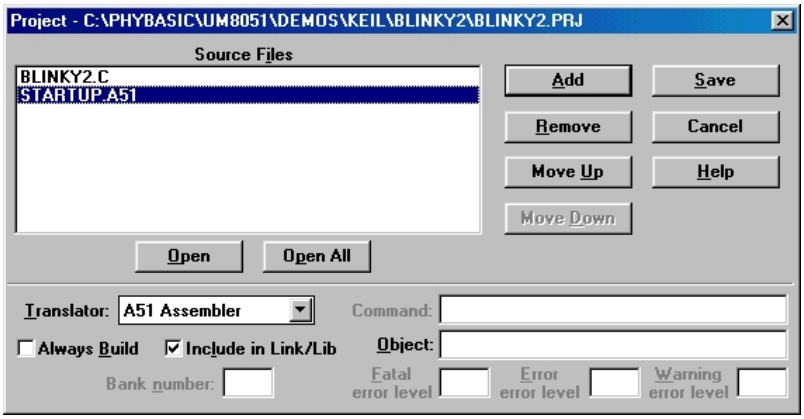
- Choosing *Add* results in the following window. This is where you add files to the project.



- Make sure you are in the correct directory: the default location is ***C:\PHYBasic\uM8051\Demos\Keil\Blinky2***.
- Select the file: ***blinky2.c*** and press <Enter> (or choose *Add*).
- Choose Close. ***blinky2.c*** will now be listed in the Source Files window. Ensure that the *Include in Link/Lib* option is enabled for your source files.
- Repeat the Add steps to also include ***startup.a51*** to your project. Change the file type to *Assembly Source* within the *Add File to Project* window to see the ***startup.a51*** file.

The ***startup.a51*** file must be the last file in the list. You might have to reposition it using the *Move Up* and *Move Down* buttons (*see section 5.2 for details*)

Your project window should now look like this:

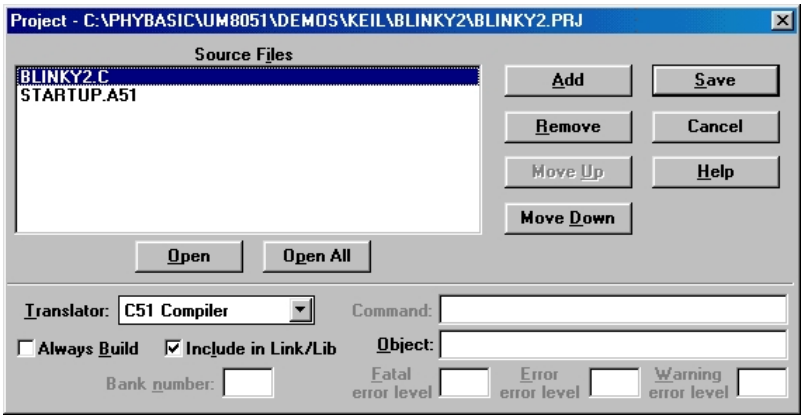


- Choose *Save* to save this new project.

At this point you have created a project called ***blinky2.prj*** and added an existing C source file called ***blinky2.c*** and an existing Assembler file called ***startup.a51***. The next step is to modify the C source before building your project. This includes compiling, linking, locating and creating the hex file.

3.3 Modifying the Source Code

- Open the *Project* menu and choose *Edit Project*. The window shown below appears. This is where you added the two source files to your project.



- Select the ***blinky2.c*** file within the Source Files window.
- Click on *Open* to open the selected source file.
- Click on *Cancel* to close the project edit window.
- Activate the edit window containing your source file. Locate the following code section and modify the bold and italic formatted value from 12,000 counts to new rates:

```
while (1) {                     /* loop forever              */

    P4_0 = 0;                   /* output to LED port        */
    for (i=0; i<18000; i++) {   /* delay for 12000 counts    */
      wait ();                  /* call wait function        */
    }

    P4_0 = 1;                   /* output to LED port        */
    for (i=0; i<6000; i++) {    /* delay for 12000 counts    */
      wait ();                  /* call wait function        */
    }
  }
```

This will change the LED on/off ratio.

3.4 Saving the Modifications

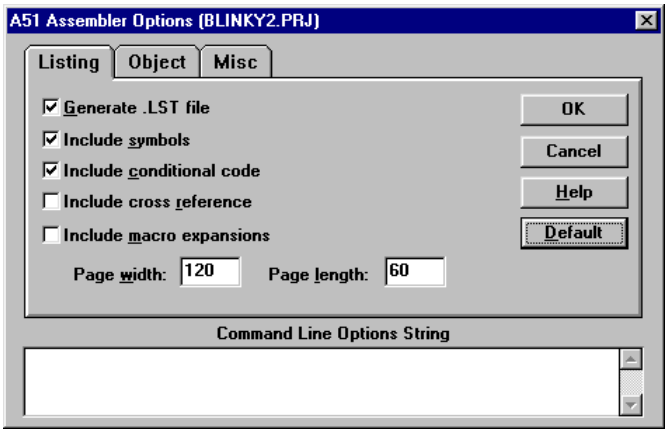
- Save the modified file by choosing *File*|*Save* or by clicking the floppy disk icon 
  Note that the icon is active as soon as you modify the file.
- Close the edit window

3.5 Setting Tool Chain Options

Keil Software has a Make utility that can control compiling and linking source files of different programming languages. Before using the Make utility, macroassembler, C compiler or linker you must configure the corresponding options. Enter the changes as indicated and leave all other options set to their default values. µVision allows you to set various options with mouse clicks and these are all saved in your project file.

To configure the A51 Assembler:

- Open the *Options* menu and choose *A51 Assembler*. Note the resulting command line options string at the bottom of the window.
- Adjust the default settings by clicking *Default*.

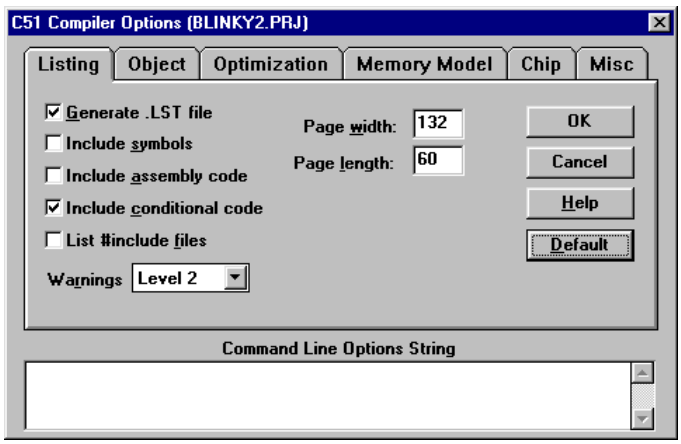


The A51 compiler default options are suitable for the ***Blinky2*** project, enabling you to build a object file without taking into account debugging settings. The options lead to the generation of a ****.lst*** file with included symbol information and prevent inclusion of additional debug information to the object file.

- Leave the A51 Assembler Options menu by clicking *Ok*.

To configure the C51 Compiler:

- Open the *Options* menu and choose *C51 Compiler*. Note the resulting command line options string at the bottom of the window.
- Adjust the default settings by clicking *Default*.

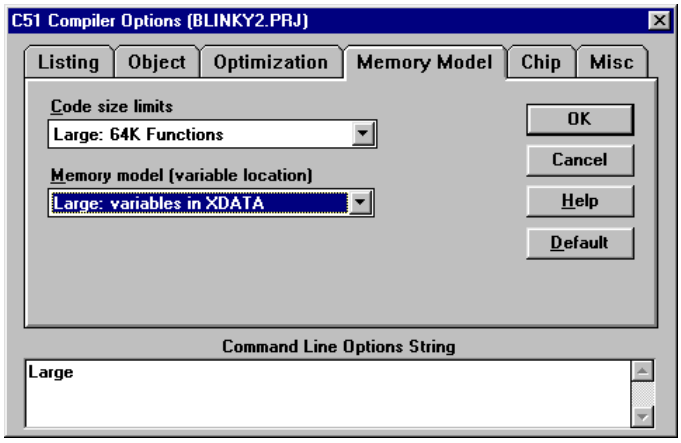


The C51 compiler default options are suitable for the ***Blinky2*** project, enabling you to build a object file without taking into account debugging settings. The options lead to the generation of a ****.lst*** file without included symbol or assembler information, allow the inclusion of interrupt vectors to the object file, prevent inclusion of additional debug information to the object file and limit the code up to 64 kByte.

However the large memory model can be used because the microMODUL-8051 is populated with at least 32 kByte of external RAM. This means that all variables are located to external XDATA memory.

In order to configure the large memory model, the following steps must be undertaken:

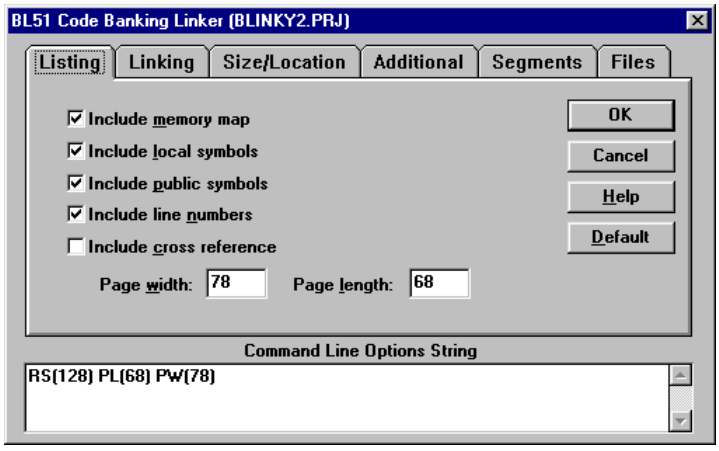
- Enter the *Memory Model* tab and select memory model *Large* as shown below



- Leave the C51 Compiler Options menu by clicking *Ok*.

To configure the BL51 Code Banking Linker:

- Open the *Options* menu and choose *BL51 Code Banking Linker*. Note the resulting command line options string at the bottom of the window.
- Adjust the default settings by clicking *Default*.

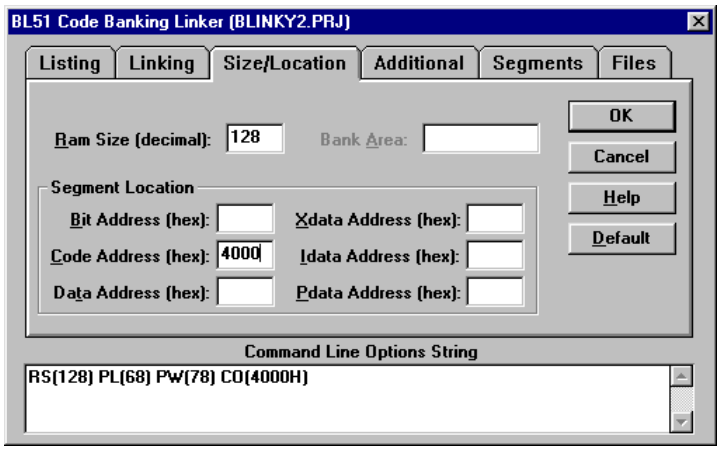


The Code Banking Linker options are suitable for the ***Blinky2*** project, enabling you to build an absolute object file without taking into account debugging settings. The options lead to the generation of a ****.m51*** map file that includes memory map and symbol information, default libraries and assumes availability of internal RAM with a size of 128 Byte.

The restrictions of the evaluation tool chain allow a code size of up to 2 kByte of code. To prevent programming of small devices with internal code memory with up to 2 kByte, the Keil tool chain automatically locates code to 0x4000 without any user action.

Our project is built with enabled Variable Overlay. Hence the code segments of our project will be located to start at 0x4000. Even though we can not override this limitation of the evaluation version (EK) of the Keil tool chain, it is recommended that you set the Linker settings to accomplish a similar project result even if you would start using the full (PK) version of the Keil tool chain.

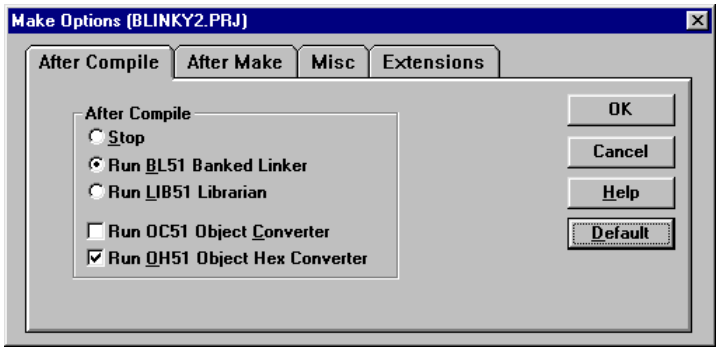
- Enter the *Size/Location* tab
- Enter 4000 at *Code Address (hex):* input field



- Leave the BL51 Code Banking Linker Options menu by clicking *OK*

To configure the Make utility:

- Open the *Options* menu and choose *Make*
- Adjust the default settings by clicking *Default*



The Make default options are suitable for the ***Blinky2*** project. Make will run the Banked Linker and the Object-Hex-Converter to build an absolute hexfile which we can program to the on-board Flash of the microMODUL-8051.

- Leave the Make Options menu by clicking *Ok*.

3.6 Building the Project

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

- Click on the *Build All* icon from the µVision tool bar  or open *Project* and select *Make:Build Project.*
- 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 Project Status dialog box, which indicates "Make successful - HEX file created". 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*** will be selected.
- Click on *OK* in the Project Status dialog box. Note that a machine-readable, executable hex-file has been created. Other files (e.g. list-files ****.lst*** and map-files ****.m51***) 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.

You are now ready to download the new machine-readable, executable hex-file ***blinky2.hex*** to the target hardware by repeating the download steps described in *section 2*

3.7 Downloading the Output File

- Exit Keil µVision.
- 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, releasing first the Reset (S1) and then the Boot (S2) switch
- Start FlashTools98 for Windows
- At the Serial Interface tab of the FlashTools98 Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Connect* button to establish connection to the target hardware

- At the optional Programming Mode Selection window, choose *External Flash Memory* if the window appears
- Returning to the FlashTools98 Worksheet, choose the *Bank Utilities* tab, highlight Bank #1in the Bank Erase section of the tab and click on *Erase Bank(s)*
- Wait until the status check in the lower left corner of the FlashTools98 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
- Be sure to enter the file ***blinky2.hex***, located in the ***C:\PHYBasic\uM8051\Demos\Keil\Blinky2*** directory (default location)
- Click on the *Download* button and view the download procedure in the status window
- Return to the *Communication* tab and click on the *Disconnect* button
- Press the Reset button (S1) on the Development Board
- If the modified hex-file properly executes, the LED should now flash in a different mode with different on and off duration's.

You have now modified source code, recompiled the code, created a modified download hex-file, 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 Keil µVision environment and close all projects and source file windows that might be open
- Open the Project menu, choose the ***C:\PHYBasic\uM8051\Demos\Keil\Hello2*** path (default location) and create a new project called ***hello2.prj*** within the existing project directory on your hard-drive
- Add ***hello2.c***, ***serinit.lib*** and ***startup.a51*** from within the project directory to the project ***hello2.prj***. Remember to move ***startup.a51*** to the end of the list of all files in the project. Also ensure that the *Include in Link/Lib* option is enabled for each file.
- *Save* the project

At this point you have created a project called ***hello2.prj*** consisting of a C source file called ***hello2.c***, a library called ***serinit.lib*** and an assembler file called ***startup.a51***.

3.8.2 Modifying the Example Source

- Open the project window by selecting *Project* | *Edit Project*
- Open the file ***hello2.c*** from within the project window
- Use the editor to modify the *printf* command as follows:

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

to

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

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

3.8.3 Setting Tool Chain Options

- Adjust the default options for the A51 Assembler by clicking *Default*
- Adjust the default options for the C51 Compiler by clicking *Default* and, afterwards, set the Memory Model to “Large” in order to locate variables in XDATA
- Adjust the default options for the BL51 Code Banking Linker by clicking *Default* and, subsequently, define the Code Address to 4000(hex)
- Adjust the default Make options

3.8.4 Building the New Project

- Build the project
- If any source file in the project contains 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 µVision
- Reset the target hardware and renders it into 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, several seconds later, the Boot (S2) switch
- Start FlashTools98 for Windows
- At the Serial Interface tab of the FlashTools98 Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Connect* button to establish connection to the target hardware
- At the optional Programming Mode Selection window, choose *External Flash Memory* if the window appears.
- Returning to the FlashTools98 Worksheet, choose the *Bank Utilities* tab, highlight *Bank #1*, and click on the *Erase Bank(s)* button
- Next choose the *File Download* tab and click on the *File Open* button
- Download the file ***hello2.hex*** from the project directory ***C:\PHYBasic\uM8051\Demos\Keil\Hello2*** (default location).
- Click on the *Download* button and view the download procedure in the status window
- Disconnect from the target hardware after the download is finished
- Exit FlashTools98

3.8.6 Starting the Terminal Emulation Program

- Start HyperTerminal and open a session 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
- Press the <Space> bar on the keyboard to proceed the automatic baud rate detection
- Successful execution will send the character string "***PHYTEC... Stick It In!***" from the target hardware to the HyperTerminal window

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

4 Debugging

The Keil debugger has two possible modes:

- as dScope, the Keil debug environment is a software debugger and simulator
- as tScope, the Keil debug environment installed on a host-PC communicates with the target hardware via a Monitor kernel that is downloaded to the Flash memory on the target hardware

Keil dScope is the initial mode when starting the Keil debugging environment. The debugger becomes tScope after communication with the target hardware is successfully established. This example utilizes the tScope debug environment, which automatically downloads a special Monitor kernel to the target hardware using the Bootstrap mode.

This example utilizes the tScope debug environment, which first requires download of a special ****.hex***-file to the target hardware.

4.1 Preparing the Target Hardware to Communicate with tScope

- Ensure that the target hardware is properly connected to the host-PC and a power supply
- Reset the target hardware and 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, several seconds later, the Boot (S2) switch
- Start FlashTools98 for Windows
- At the Serial Interface tab of the FlashTools98 Worksheet, specify the proper serial port and transmission speed for communication between host-PC and target hardware and click the *Connect* button to establish connection to the target hardware
- At the optional Programming Mode Selection window, choose *External Flash Memory* if the window appears.

- Returning to the FlashTools98 Worksheet, choose the *Bank Utilities* tab, highlight *Bank #1*, and click on the *Erase Bank(s)* button
- Next choose the *File Download* tab and click on the *File Open* button
- Download the file ***mon51.hex*** from the tools directory ***C:\PHYBasic\uM8051\Tools\Mon\Keil\32KB*** (default location).

There may also exist some other subdirectories to the ***C:\PHYBasic\uM8051\Tools\Mon\Keil*** which contain other versions of the ***mon51.hex***. Please refer to *readme* files within the directories for details.

- Click on the *Download* button and view the download procedure in the status window

If download is successful, the Monitor kernel has been programmed into the external Flash memory. The target hardware is now prepared to communicate with the Keil software installed on the host-PC.

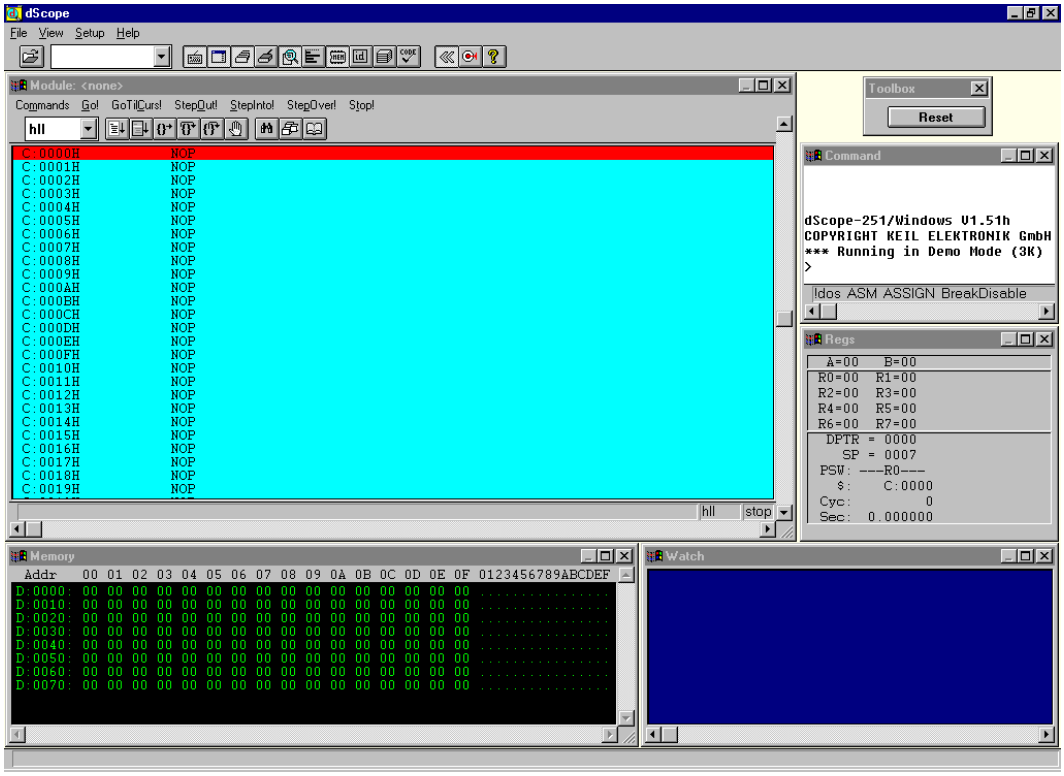
- Disconnect from the target hardware after the download has finished either by clicking the disconnect icon or choosing disconnect from menu.
- Exit FlashTools98
- Reset the Target Hardware

4.2 Starting the Debugger

The correct communication driver and interface must be selected within the Keil debug environment before the Monitor kernel that has been downloaded to the target hardware can communicate with the tScope debugging environment on the host-PC.

- To start the Keil debug environment, click on the debugger icon on the µVision toolbar  or select dScope-51 from within the *Programs|Keil PK-51 – Eval* group.

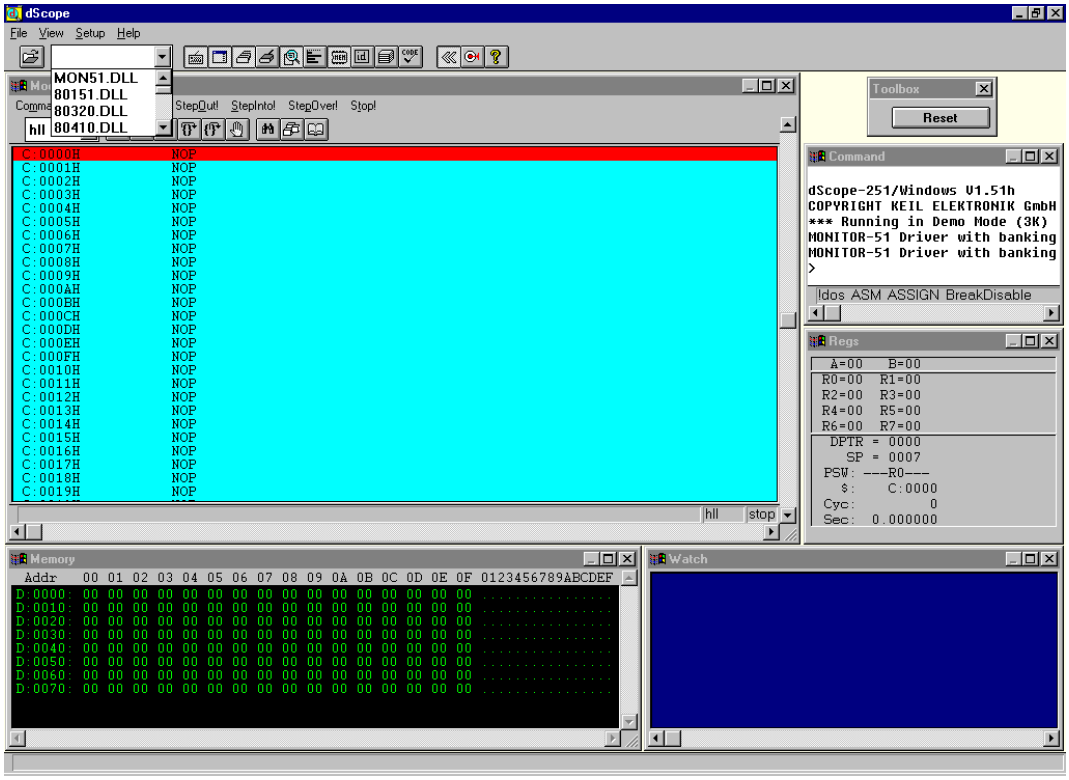
- A dScope window similar to that shown below appears. The upper left-hand corner of the following figure shows that the debugger is initially in the dScope mode for this example.



You may want to move and resize the windows. The window marked *"Module: <none>"* is the Debug window. The Command window can be used to enter commands. It will not be used during this portion of the example. You may need to open, resize and /or move some windows to make your screen look something like to the screen capture. The Debug, Memory, Watch, Command and Regs windows will be used. You may have to open any invisible window by opening the *View* menu and selecting the window to view. You may also want to open the *Commands* menu in the Debug Window and select *Show Dialog bar*, *Show Status bar* and *Show Tool bar*.

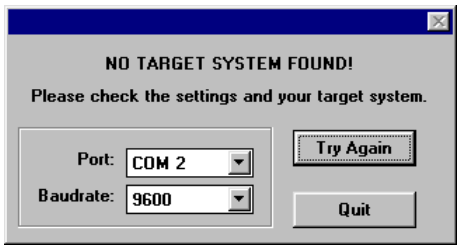
To establish communication with the target hardware within tScope:

- Choose the correct driver using the pull-down list on the left portion of the debugger toolbar as shown below



- Select the appropriate ***.dll** file – in this case ***mon51.dll*** - and serial link prompts to enable dScope to try to communicate with the Monitor kernel located in the Flash memory on the target hardware
- If communication is successful, tScope replaces dScope in the upper left hand corner of the debug tool bar

- If communication is not successful, dScope will error out as indicated in the "No Target Hardware Found!" dialog box. If this happens, try other baud-rates. The correct speed should be 9,600 baud. Click on *Try Again* after reentering communication parameters.



- For troubleshooting please refer to *Appendix A1*.

Selecting ***8051.dll*** maintains the dScope environment. This results in a debug environment run on the host-PC rather than on the target hardware.

The serial FIFO buffer in Windows 95 can cause transmission problems. dScope may be unable to complete the communication initialization process. Refer to *Appendix A* for further information and troubleshooting.

4.3 Loading the Example File to the Debugger Environment

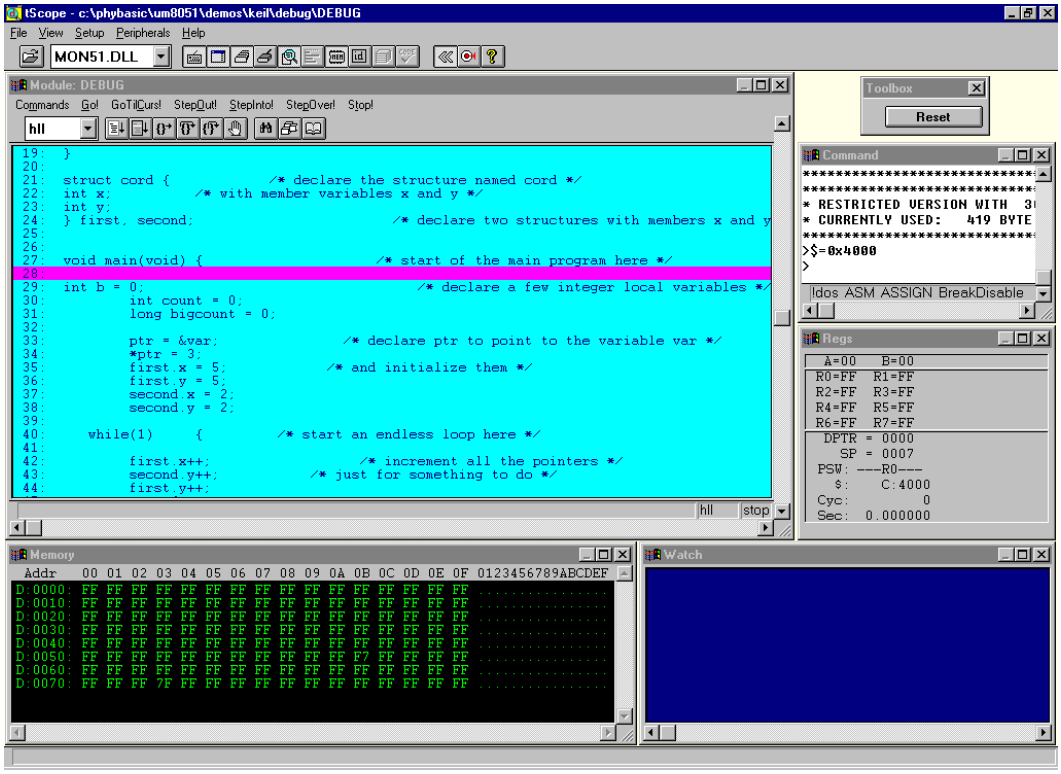
Once communication has been established, you are now ready to run the debugger to download example user code to the target hardware.

Within the default location
C:\PHYBasic\uM8051\Demos\Keil\Debug
you will find an already built project called ***debug.***

Note that some C51 Compiler options should be used to build a project suitable for use with the dScope/tScope debugger: *Include symbols*, *Include debug information* and *Include extended debug information*. These enable you to view your project at source-level within the debugger.

Whenever debugging using tScope and its target connection, please note the shared memory properties of the monitor hexfile ***mon51.hex*** that is downloaded to your target hardware. Once downloaded, the Monitor kernel and your user application both reside within the same target and memory. This requires special configuration of the memory. Disregarding this special configuration may lead to malfunction of user code. See *section 5.3* for details.

- Open the File menu and choose *Load Object file*
- In the Window provided, select the file: ***debug.*** The file is downloaded to the board and the debugger. You will see the source code on your screen. Note that with some code, you might not see your source code because the program starts at 004000H, which is a condition of the evaluation version of the Keil Compiler. You may see the LJMP instruction at C:0000H. The full version of the Keil tool chain will start at 0x0000 or wherever you set it.
- If necessary enter the following command in the Command window: $=0x4000 to see the code. This command sets the Program Counter ($) to hex 4000. You can also use the command *$=main* if you prefer. The program counter will advance to *main*. You can press the up arrow key to get the history of keys typed in the Command Window.
- To see the Command window, if not already visible, select the *View|Command* Window. To see the source code, select the Debug window in the same fashion
- You should now see the source in the Debug window below, depending on how you set your windows. You are now ready to use tScope to step through code, set breakpoints, and issue the Go command to start program execution. You can examine special function registers, memory locations, and register values, etc. For instance, you can single step through code using the StepInto icon.



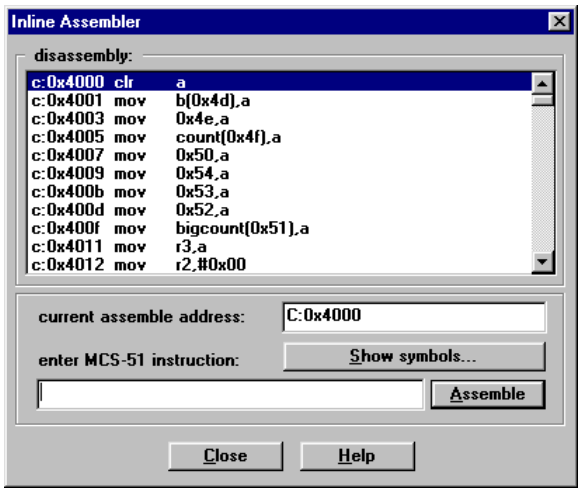
4.4 Using the Keil tScope Debug Features

4.4.1 Breakpoints

- Click on a memory location such as line number 42 first.x++. A colored bar appears marking this position.
- You could click on *Go Til Curs!* to reach this point or you double-click and set a breakpoint. Set a breakpoint here with the double-click. The [BR0] mark on the right hand of the selected line indicates the first breakpoint.
- Click on *GO* and the program will run and stop at the breakpoint.
- Double click on the breakpoint to remove it.

4.4.2 In Line Assembler

- Open the *Commands* option and click on *Inline assemble...* or if Show Dialog bar is active, click on the *Assemble...* button on the right hand of the Debug window.
- The Inline Assembler windows opens as indicated below.



- Note that you could enter mnemonics in the window titled *enter MCS-51 instruction:* Do not do this at this time. Just note the ability to make small changes in the program without recompiling the program.
- Click on *Close*.

4.4.3 Single Stepping

- tScope uses "*Step Into*" to single step one instruction at a time. "*Step Into"* is also used to enter a function in the same fashion.
- "*Step Over*" means to skip over a function in which you are not interested
- "*Step Out"* is used to exit a function you are currently in. "*Step Out"* 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 tScope gets stuck - do a reset using the RESET box on the Toolbox or click on the reset icon  next to the Help icon. Then reload the file and/or the ***.dll***. You can also try using the ***8051.dll*** instead of ***mon51.dll*** to isolate any problems from the target hardware. You can also use the hardware reset button on the board.
- With the cursor on line 42, click on *StepInto* until you enter the function *stepout( )*. Note that you must click on *StepInto* ten times to exit the *for* loop within the function *stepout( )*.
- Repeat the process using *StepOver* and you will not enter the function although it will be executed. Use the command *$=\42* within the Command Window to reposition the program counter to line 42 prior to further code execution. This command takes a line number preceded by a backslash to set the program counter. The *StepOver* feature is useful to skip function calls you are not interested in debugging.

Note that the *StepOut* button is grayed out and not available in tScope. *StepOut* is available in the simulator dScope and provides a quick escape from a function by executing the next *return* instruction.

4.4.4 Memory Window

- If not already visible view the Memory Window by selecting *View|Memory window* at the tScope menu
- The Memory Window shows as the default the data space starting at D:0x0000. This is the area where data variables are kept in this example.
- As you step through the program, please note that the contents of the memory changes as the variable values are adjusted
- You can change the memory area with the *Display* command in the Command window. Enter d c:0x00 and the code area will be displayed.
- Change back to the data area with d d:0x00

4.4.5 Watch Window

- If not already visible view the Watch Window by selecting *View|Watch window* at the tScope menu
- The *Watch* window displays memory contents as specified by name. Structures can also be displayed
- Open the *Setup* menu and select *Watchpoints*. The window shown below appears.

- In the *Expr.:* window, enter the following variable symbols each, followed by clicking on *Define watch* :
  - *bigcount*
  - *first*
  - *second*

- Note that if you select radio button *10* instead of the default value *0xnn* prior to clicking on *Define watch*, the values will be displayed in decimal rather than hexadecimal notation. Try this for the following variable symbol:
  - *var*

- Close the *Watchpoint* window
- The variables will now be displayed in the *Watch* window

- Click on *StepOver* and the variable values will change as appropriate.

If the *Update Memory Window* and *Update Watch Window* are activated from within the Setup menu in dScope, the variable values will change as the program is running. This feature is not available if tScope is running on a target hardware board.

- Open the *Peripherals* menu and select *Configuratio*n. Activate *use serial interrupt* and click on *Apply* then *Close*. This allows tScope to communicate with the target hardware even if there is a user application running.
- Execute a reset and start over as before. Press *Go* and then *Stop* and note that the values change in both the Watch and Memory windows when the processor is stopped

5 Advanced User Information

This section provide advanced information for successful operation of the microMODUL-8051 in conjunction with the Keil tool chain.

5.1 FlashTools98

Flash is a highly functional means of storing nonvolatile data. One of its advantages among many others is the possibility of on-board programming. Programming tools for the Flash device are always included with the microMODUL-8051 in the form of a pre-programmed Flash with a resident microcontroller firmware and a counterpart software serving as the user interface on a host-PC. Once the firmware communicates with the PC-based software, FlashTools98 allows the download of user code from the host-PC into the Flash. Additionally, the reprogrammable Flash device on the microMODUL-8051 allows you to easily update your own code and target the application in which the microMODUL-8051 has been implemented.

Currently the microMODUL-8051 can be populated by two different sized Flash devices: a 29F010 with 128 kByte or a 29F040 with 512 kByte. To support the entire memory area of these devices the address decoder of the microMODUL-8051 is equipped with an integrated banking mechanism that allows code-bank switching in code-banks of 64 kByte each.

Please note that the FlashTools98 always occupies the first 64 kByte bank (bank 0, FA[18..15] = 0000b) of the Flash memory in which the microcontroller firmware resides. This bank is already pre-programmed upon delivery of the microMODUL-8051. The remaining banks are available to house your application. This makes available one user application bank if the microMODUL-8051 is mounted with a 29F010 and seven user application banks if the microMODUL-8051 is mounted with a 29F040 Flash memory device. Multiple user application banks can easily be managed by using the Code Banking mechanism of the Keil tool chain.

The following description is valid only for the FlashTools98 included with the microMODUL-8051and is not intended as guidelines for using any other program.

The FlashTools98 incorporates a safety mechanism that ensures that its system bank (bank 0), in which the firmware is resident, can not be overwritten during programming of the available user banks of the Flash device.

Resetting the microMODUL-8051 also activates the system bank (bank 0) of the Flash device, which automatically starts the Flash-Tools98 firmware. Then the module either enters the Flash programming mode or it starts your user application.

To distinguish between download and execution modes, the firmware checks immediately for the presence of a pull-up resistor (usually 4.7kΩ) connected to pin D0 (Data 0) of the microMODUL-8051 after reset. This resistor is located on the Development Board and can be connected to the microMODUL-8051 by pressing the Boot (S2) switch. Hence to enter the Flash programming mode you must simultaneously press the Reset (S1) and the Boot (S2) switch, release the Reset (S1) switch first and then, several seconds later, release the Boot (S2) switch.

Execution of your user application will always start in the second 64 kByte bank (bank 1, FA[18..15] = 0010b). This is to be noted when preparing a software copy of the contents of the address decoder's internal write-only registers.

The extended features of the address decoder of the microMODUL-8051 allows you to flexibly configure the memory model according to your needs and to address additional Flash code-banks. The Tools directory of the PHYTEC Spectrum CD contains various sample programs that illustrate the use of this feature.

Do not use Flash-bank 0 in your application program in order to preserve the FlashTools98 microcontroller firmware and the associated Flash reprogramming capability.

5.2 STARTUP.A51

The code within the assembly file ***startup.a51*** initializes your C project. This includes setting of the system stack, initialization of variables and clearing of memory areas.

It is very important that this code executes simultaneous to the execution of your user code. This requires that the startup.a51 code occupy the Reset Vector of the application. This 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.

To accommodate the startup code to the needs of your application, copy it from the ***Lib*** directory of the Keil tool chain (must be the full PK version, this directory is not included in the EK Evaluation version) 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 explicitly instruct the linker to instead consider your individual startup object file. To do this we recommend adding your modified ***startup.a51*** file to your project. Ensure that this file is always included into the Link/Lib process of your project (see options within the Project window of the Keil tool chain in *section 3.5*).

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 as a list of binary data entries. To mark the end of this list, the tool chain 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 ***startup.obj*** object file. To get all of your pre-initialized variables into consideration you must move ***startup.obj*** to the end of the list of your project 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 move ***startup.a51*** or ***startup.obj*** (depending on the kind of file you want to add to the project) to the end of your project's file list.

5.3 Linking and Locating

The Linker must 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 and data to fixed address locations within the address range in regards to the memory types of the microcontroller. This ensures the natural or explicitly declared properties of these segments. Data segments always must be located to Random Access Memory (e.g. RAM), code segments should be located in non-volatile memory (e.g. Flash). The 8051 family has a Harvard memory architecture that distinguishes between non-volatile and modifiable memory and has two physically different signals for separate fetching of data and code.

The Keil Tool Chain distinguishes the following segment types:

- CODE: code
- XDATA: external data (max. 64 kByte)
- DATA: direct addressable on-chip data (max. 128 Byte)
- IDATA: indirect addressable on-chip data (max. 256 Byte)
- BIT: bit-addressable on-chip data (max. 128 Bits)

The segment types DATA, IDATA and BIT always reside in the on-chip RAM of the controller.

The segment types XDATA and CODE will usually reside in external memory devices.

To ensure proper execution of your application it is required that all XDATA segments are located to the external RAM of the microMODUL-8051 and that all CODE segments are located to the external Flash memory of the microMODUL-8051. Exceptions may occur if you use a 8051 derivative with on-chip portions of XDATA (e.g. internal XRAM) or CODE (e.g. internal ROM).

Since the microMODUL-8051 is equipped with a software configurable address decoder instead of simple programmable logic you can configure the memory model to your needs at runtime.

To ensure proper execution of your application you must take the runtime memory model into consideration when linking and locating. This means that you must instruct the Linker where to assume external RAM for locating data segments and Flash for locating code segments.

The standard configuration of the microMODUL-8051 is equipped with 32 kByte of external RAM and 128 kByte of external Flash. During runtime the RAM will be addressable at 0x0000 to 0x7FFF, the user bank (bank 1, FA[18..15] = 0010b) will be addressable at 0x0000 to 0xFFFF. This default runtime memory model actually requires no additional linker settings because both RAM and Flash start at 0x0000. This is also the default start address of the linkers segment types.

Since you can not define any end address you should always ensure that the size of the segments fits within the available size of the mounted memory devices. For instance all XDATA segments should end below 0x7FFF if a 32 kByte RAM device is mounted on the microMODUL-8051. We recommend generation of a ****.m51*** map-file for your project and inspection of the memory map information within this file.

Whenever you modify the memory model (e.g. use Von-Neumann rather than Harvard memory), which leads to different start addresses of code or data memory, you must configure this in the linker settings.

5.4 Debugging using *mon51.hex*

Whenever you decide to use the target Monitor in tScope to debug your application, special precautions must be taken to ensure proper code execution of your application. This is because the Keil Monitor firmware and user code share some memory locations within the target hardware. If user code is not properly linked in relation to the ***mon51.hex***, you may get conflicting memory use. An improper link leads to variables containing not their assigned value, functions returning bad results and modified code. In order to avoid such errors, you should link your application to prevent any overlapping of all memory range types:

- **CODE** and **XDATA:** Ensure that the CODE and XDATA portions of your application do not overlap the XDATA or CODE portions of the Monitor. The Monitor's XDATA and CODE portions are usually linked to the top of the memory area, leaving as much memory free space as possible for user code.

- **CODE (interrupt vector):** The Monitor occupies, and may use, an interrupt for serial communication (in most cases serial channel 0, which corresponds to CODE address 0023H). Hence it is necessary to not link any user code to the interrupt vector used by the Monitor. Please note that this is not shown in the memory map-file, as the interrupt vector location is only occupied during run-time.

- **Internal RAM:** Because the Monitor uses internal RAM nearly up to 40H you should add a *Size/Location* statement to your *BL51 Code Banking Linker* options to set the *Data address (hex):* to 40.

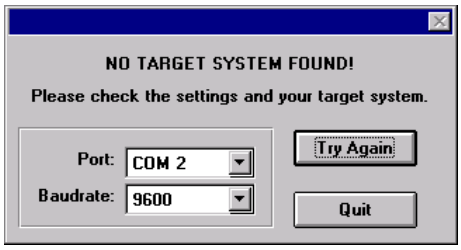
- **Bit-addressable memory:** do not use bit-addressable memory for your own code, as the Monitor occupies most of the bit-addressable memory.

To enable correct definition of the Monitor's memory requirement, PHYTEC delivers the corresponding memory map file ***mon51.m51*** in addition to the ***mon51.hex*** executable file. This file contains a detailed memory map of the Monitor.

A Troubleshooting

A1 tScope: NO TARGET SYSTEM FOUND!

If tScope's communication to the target system is not successful, it will error out as indicated in the "No Target System Found!" dialog box shown below. If this happens, check if the proper serial port is selected or try other baud rates. The correct speed should be 9,600 baud. Click on *Try Again* after reentering communication parameters and resetting the target system.



The serial FIFO buffer in Windows 95 can cause transmission problems. tScope may have problems completing the communication initialization process. This can be intermittent. The FIFO can be disabled under *Control panel|System|Device Manager|Port Settings|Advanced*. Make sure *Use FIFO buffers* in this menu is not activated.

Document: microMODUL-8051 QuickStart Instructions
Document number: L-381e_3, May 2000

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

Ordering No. L-381e_3
Printed in Germany