

## 1 VersaKit-30xx contents:

The VersaKit-30xx development kit is a plug-and-play evaluation system for the VRS51L3xxx series of high performance, fully-integrated, FRAM-Enhanced™ 8051 microcontrollers (MCUs). The VersaKit-30xx provides a complete and comprehensive programming and development platform, with ample prototyping space and easy access to chip peripherals and I/Os.

The VersaKit-30xx ships complete with:

- Development board that supports the VRS51L3074 (or VRS51L2070)
- VJTAG-USB programming/debugging interface
- USB cable
- 6V power supply (Euro or U.S. compatible)

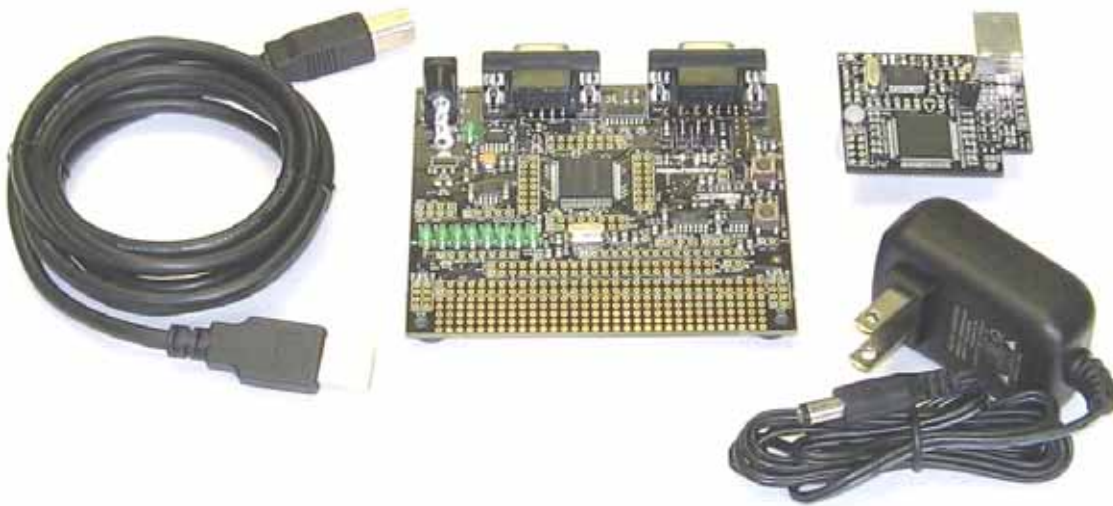


Figure 1: Complete VersaKit-30xx kit

## 2 Software and Documentation

The VersaKit-30xx Getting Started Guide is a quick reference guide to setup the VersaKit-30xx.

Before you begin, visit the Ramtron web site at [http://www.ramtron.com/doc/Products/Microcontroller/Support\\_Tools.asp](http://www.ramtron.com/doc/Products/Microcontroller/Support_Tools.asp) to download the following:

- VersaKit-30xx User Guide
- Versa Ware JTAG software

The user guide provides a complete description of the VersaKit-30xx feature set, as well as the development board schematics. The Versa Ware JTAG Software provides an easy-to-use platform for device programming and in-circuit debugging of the code. Include files and demonstration programs for the VRS51L3074 can be downloaded from the abovementioned web page.

Download the VRS51L3074 datasheet at [http://www.ramtron.com/doc/Products/Microcontroller/MCU\\_list.asp](http://www.ramtron.com/doc/Products/Microcontroller/MCU_list.asp).

### 3 Overview: VersaKit-30xx Development Board and the VJTAG Interface

The figure below presents an overview of the VersaKit-30xx development board and its principal features:

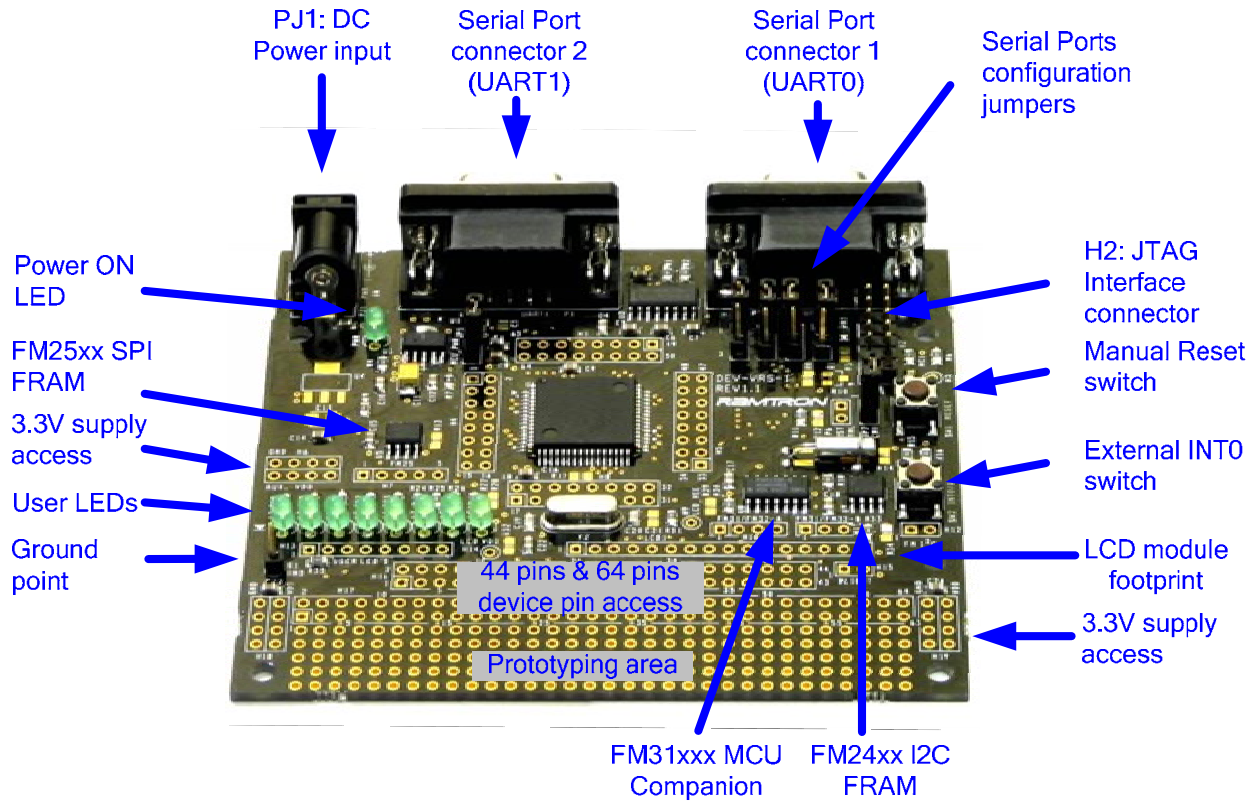


Figure 2: VersaKit-30xx development board

The VJTAG-USB board is a USB-based JTAG interface that enables in-circuit programming and debugging of the Ramtron VRS51L3074 and other JTAG-based MCUs. The VJTAG-USB connects directly to the VersaKit-30xx development board and can be used on prototype/production boards featuring any Ramtron JTAG-based MCU.

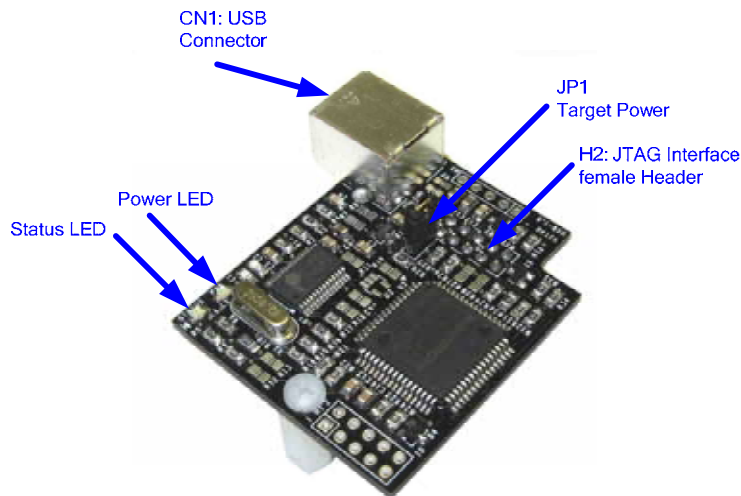


Figure 3: VJTAG-USB interface

## 4 Basic Development Board and VJTAG Setup

The setup of the VersaKit-30xx is quick and easy. However, the order in which the operations are performed is important and should be followed.

First, insert the VJTAG-USB into Header H2 on the devboard so that the CN1, USB connector sits directly on top of the devboard's rightmost DB9 connector. If the VJTAG-USB is to provide power to the devboard, then a jumper should be inserted at position JP1. If the development board's is to have an external power supply, the jumper should be removed from JP1. Consult the VersaKit-30xx for more details on power supply configuration.



Figure 4: Installing the VJTAG-USB on the devboard

Second, connect the USB cable to the VJTAG-USB board.



Figure 5: Connecting the USB cable to the VJTAG-USB

From this point, it is important not to remove the VJTAG-USB board from the development board without first disconnecting the USB cable. Doing so may cause Windows® OS instability, and require a computer reboot.

## 5 Overview: Versa Ware JTAG Software

Versa Ware JTAG is a Windows®-based software tool that provides a user-friendly development platform for all Ramtron microcontrollers featuring a JTAG interface (the VRS51L2xxx, VRS51L3xxx and future derivatives).

The Versa Ware JTAG Software is composed of two parts:

- **Versa Ware JTAG Programmer**

The Versa Ware JTAG Programmer is used to perform operations such as erase, program, read, etc., on the target device's Flash memory.

- **Versa Ware JTAG Debugger**

The Versa Ware JTAG Debugger is a user interface that links the in-circuit debugger and the source code. All Ramtron MCUs with a JTAG interface include an integrated debugger that enables in-application debugging of the device via its JTAG interface.

The Versa Ware JTAG Debugger is compatible with the SDCC, Keil and Ride compilers.

The Versa Ware JTAG Software has been developed under Windows XP, but should also run on Windows Vista and Windows 2000 operating systems, as well as Windows 98 SE Operating Systems, as long as the PC has a USB interface. Systems without a USB interface can use Ramtron's parallel port version of the JTAG interface. Please contact Ramtron for details.

The following window shows the in-circuit debugger in action when the source code XRAM, SFR page 0 and watch list windows are open and the program is halted at a breakpoint.

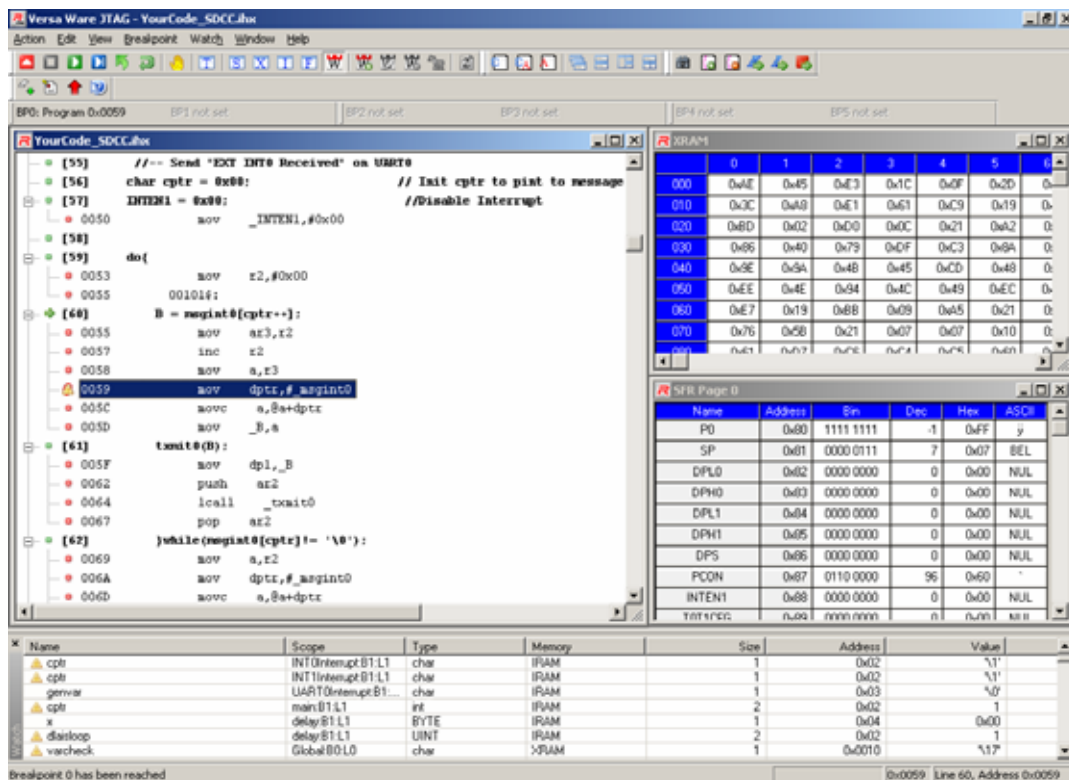


Figure 6: Versa Ware JTAG Debugger window

The Versa Ware JTAG Debugger provides a comprehensive set of configuration options allowing the user to tailor the user interface, the watch variables and the breakpoints settings.

## 6 Installing and Using the Versa Ware JTAG Software

Installation of the Versa Ware JTAG Software is handled automatically by the set up program, which installs the:

- Versa Ware JTAG Software
- Prolific PL2303X USB driver

As is the case with most software/driver installations, we recommend creating a system restore point before running the setup. To proceed with Versa Ware JTAG installation, download from the Ramtron web site and run the following file: **Versa\_Ware\_JTAG\_3x\_SETUP.exe**.

Once the software is installed, it can be run directly from the setup program or by clicking on the Versa Ware JTAG shortcut created during the installation process.



Versa Ware JTAG

Figure 7: Versa Ware JTAG icon

### 6.1 Versa Ware JTAG Programming Interface

Upon startup, the software will attempt to connect to the VJTAG-USB interface.

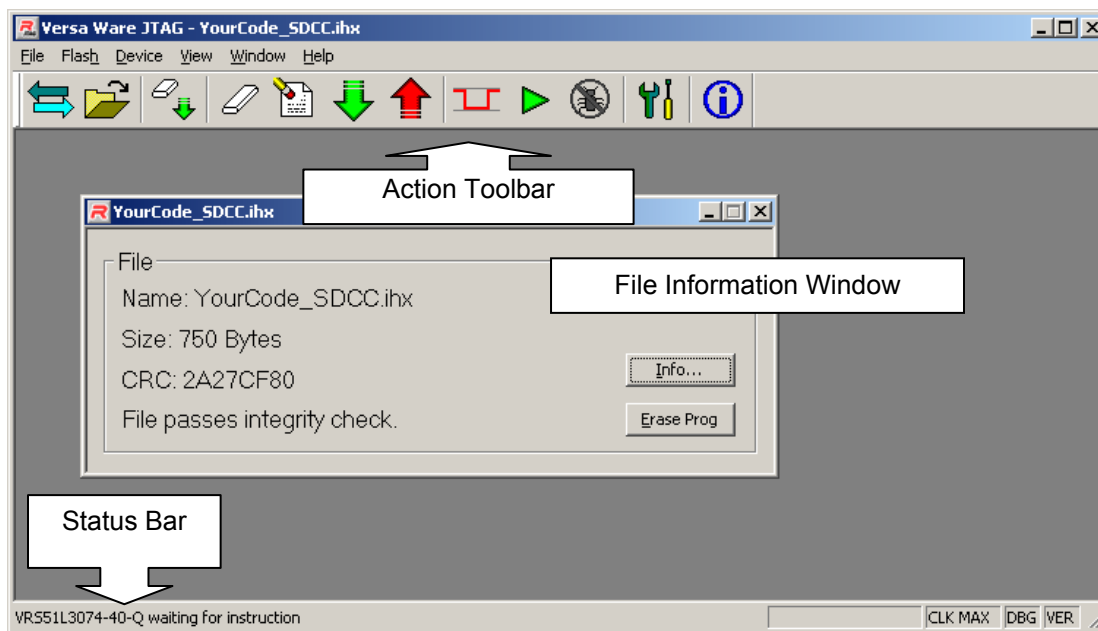








Figure 8: Versa Ware JTAG programming interface

Most of the functions provided by the Versa Ware JTAG Software are accessible through the action toolbar.

To download a HEX file into the VRS51L3074:

1. Make sure that the Versa-JTAG interface is properly connected to the H2 header of the devboard.
2. Click on the Synchronize  button. The status bar should show: "VRS51L3074-40-Q waiting for instruction".
3. Click Open  to select the HEX file to be programmed into the VRS51L3074.
4. Click on Erase then Program  to erase and program the Flash. By default, after this process is complete, the program will start.

The  Synchronize button can be used to halt execution of the VRS51L3074 program and put the device into program mode. The  Run button restarts program execution.

The Options  button allows configuring the programming options, set the Flash security options, and activates the in-circuit debugger.

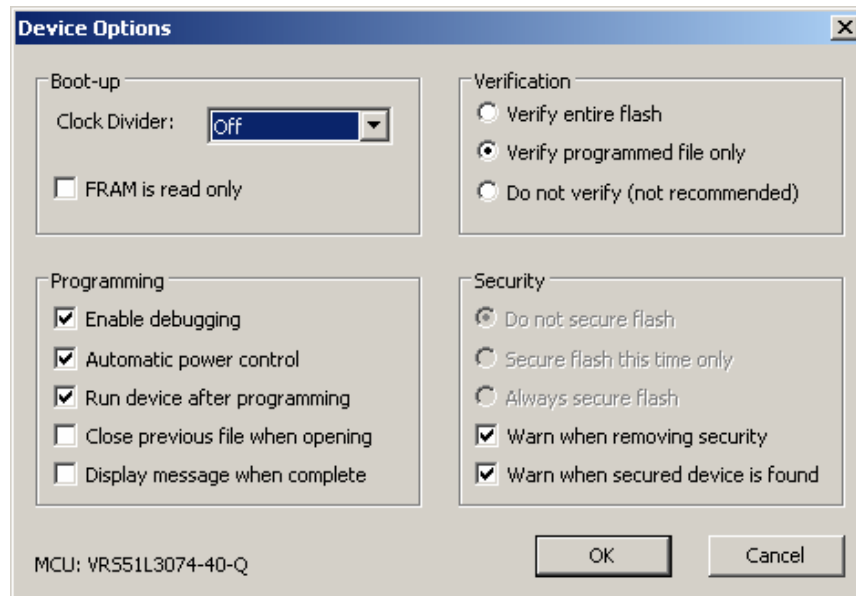




Figure 9: Versa Ware JTAG device options

Changes to any device options will become effective the next time an Erase then Program  operation is initiated.

## 6.2 Using the Versa Ware JTAG Debugger

Once the program is loaded into the VRS51L3074 Flash memory and the debugger is enabled, activate the debugger by clicking the Debugger  button. Upon startup, the debugger will halt the processor at address 0x0000 and wait.

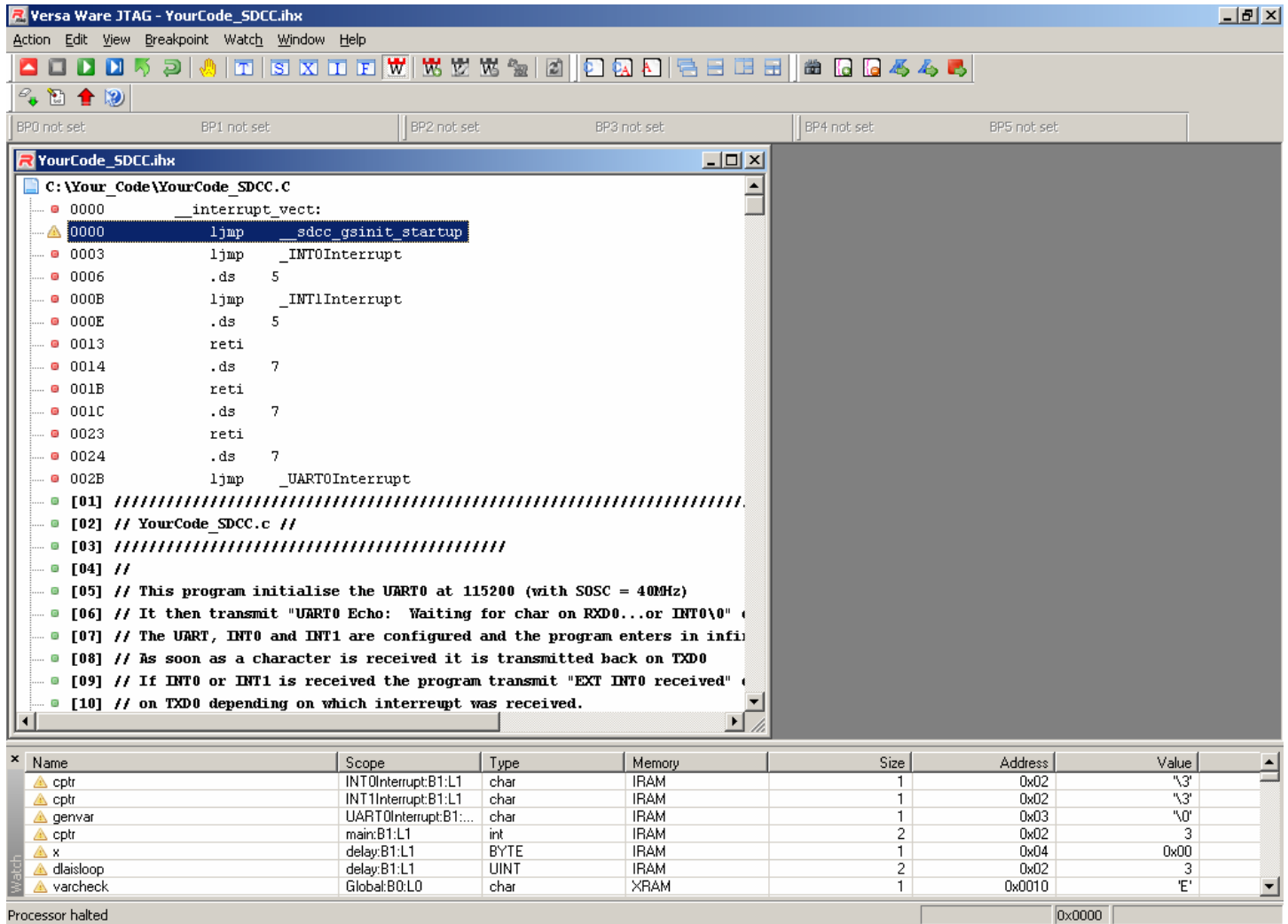



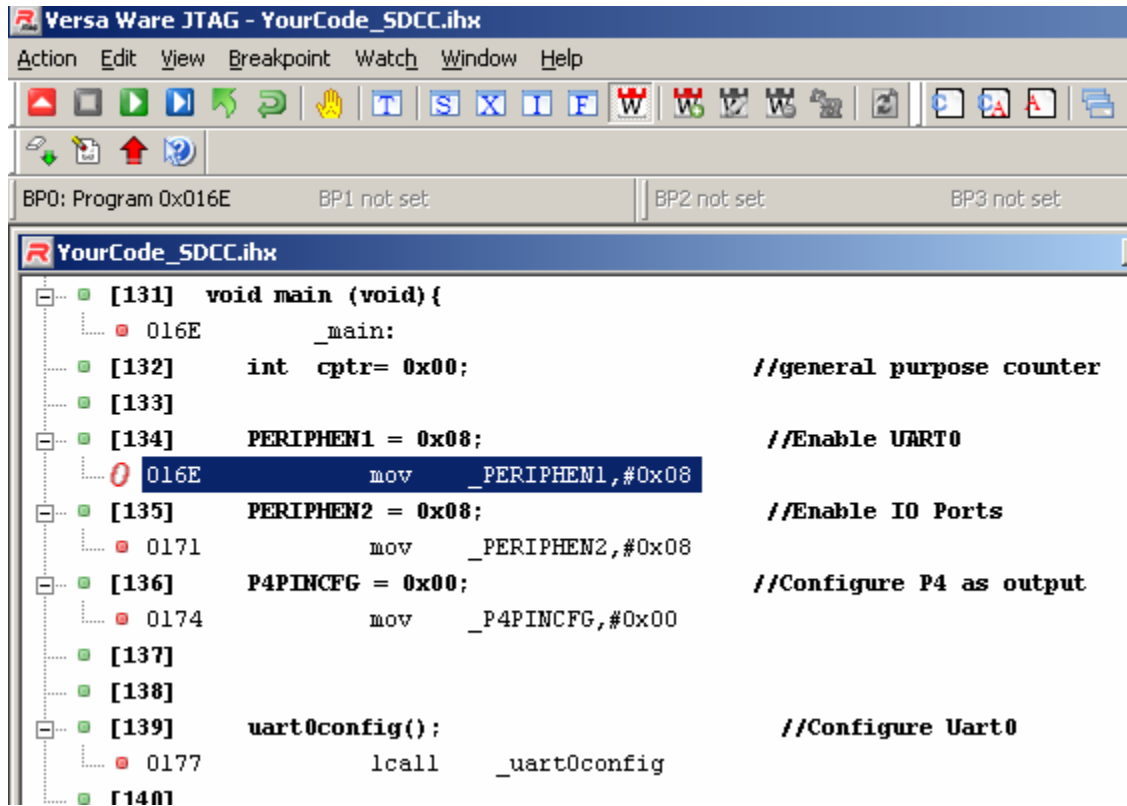


Figure 10: Versa Ware Debugger upon startup

Once the debugger has started, either run the program at full speed by clicking on the Run  button or set a breakpoint anywhere in the code by either double clicking on a specific code line and then clicking the Run  button or by setting the breakpoint manually. Alternatively, you can manually set breakpoints anywhere in the code by clicking the Breakpoint Setting button .

When a breakpoint is set in the code, the breakpoint reference number will be displayed in red next to the code line. The breakpoint toolbar will then display the active breakpoint. Double clicking again on a given code line where a breakpoint has been set will disable that breakpoint.



```


Versa Ware JTAG - YourCode_SDCC.ihx
Action Edit View Breakpoint Watch Window Help

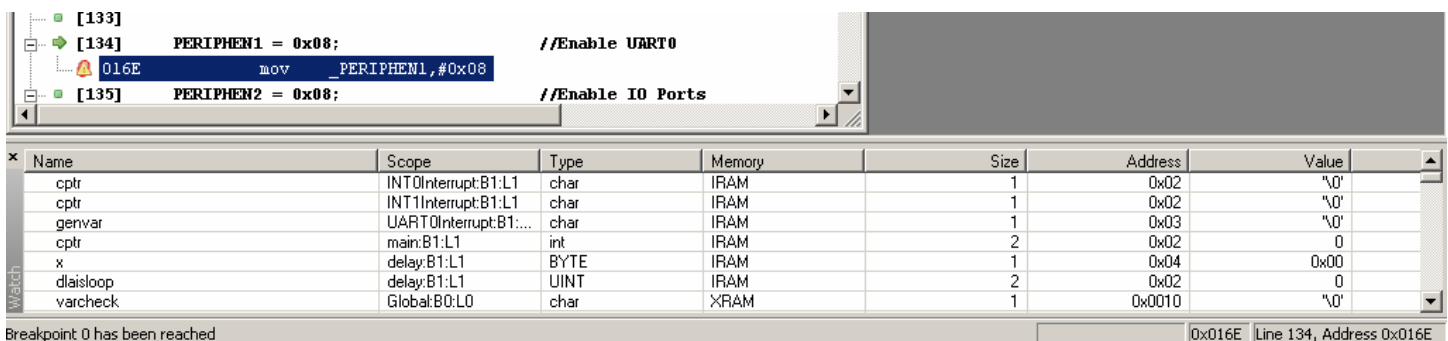
BP0: Program 0x016E   BP1 not set   BP2 not set   BP3 not set

YourCode_SDCC.ihx
[131] void main (void){
  [132] int cptr= 0x00;           //general purpose counter
  [133]
  [134] PERIPHEN1 = 0x08;       //Enable UART0
  016E mov  _PERIPHEN1,#0x08
  [135] PERIPHEN2 = 0x08;       //Enable IO Ports
  0171 mov  _PERIPHEN2,#0x08
  [136] P4PINCFG = 0x00;        //Configure P4 as output
  0174 mov  _P4PINCFG,#0x00
  [137]
  [138]
  [139] uart0config();         //Configure Uart0
  0177 lcall _uart0config
  [140]

```

Figure 11: Set breakpoint by double clicking on a code line

Once a breakpoint is set, click on the Run  button. The processor will then start executing code at full speed until the breakpoint is reached.



```

[133]
[134] PERIPHEN1 = 0x08;           //Enable UART0
016E mov  _PERIPHEN1,#0x08
[135] PERIPHEN2 = 0x08;           //Enable IO Ports

```






Name	Scope	Type	Memory	Size	Address	Value
cptr	INT0Interrupt:B1:L1	char	IRAM	1	0x02	'\0'
cptr	INT1Interrupt:B1:L1	char	IRAM	1	0x02	'\0'
genvar	UART0Interrupt:B1:...	char	IRAM	1	0x03	'\0'
cptr	main:B1:L1	int	IRAM	2	0x02	0
x	delay:B1:L1	BYTE	IRAM	1	0x04	0x00
dlaishoop	delay:B1:L1	UINT	IRAM	2	0x02	0
vcheck	Global:B0:L0	char	XRAM	1	0x0010	'\0'


Breakpoint: 0 has been reached

0x016E Line 134, Address 0x016E

Figure 12: Breakpoint reached

Once a breakpoint is reached, you can either:

- Restart the program by clicking again on the Run  button.
- Single step through the code by clicking on the Step  button.
- View/edit SFR, IRAM, XRAM memory locations by clicking on SFR , IRAM , XRAM  buttons.

When the Step  button is clicked, the processor will execute the current instruction and step to the next instruction (and actually stop).

SFR, IRAM, XRAM or FRAM memory content can be viewed and edited anytime a breakpoint is reached. This includes in step mode. The debugger provides table and list view options for the SFR, IRAM, XRAM and FRAM memory areas. It is possible to open more than one window of each one of these memory areas.

Editing a given SFR, IRAM, XRAM and FRAM memory cell is simple:

- Double click on the memory location to edit.
- Type-in the new value to write.
- Press <Enter> or double click on another memory location.

#### **Notes about the SFR edition:**


In the case of the SFR memory location, it is important to note that some registers are partially or totally read or write only or can be accessed only when the related peripheral is activated. The SFR edition feature allows the user to control a peripheral up to a certain extent.

### **6.3 Updating the Code without exiting the debugger**

If during a debug session, you find issue in your code and want to modify it, recompile and reload the code into the Flash memory. The Versa Ware JTAG allows you to reload the code in the Flash memory without exiting the Versa Ware JTAG Debugger environment. To do this, simply go to your editor, modify the code, recompile it, return to Versa Ware

Debugger, and click on the Erase then Program  button. Note that all the breakpoints you set will be lost.

### **6.4 Exiting the Debugger**

To exit the debugger, click on the Stop Debugging  button. This will return you to the Versa Ware JTAG programming environment.

#### **Disclaimers**

**Right to make change** - Ramtron reserves the right to make changes to its products - including circuitry, software and services - without notice at any time. Customers should obtain the most current and relevant information before placing orders.

**Use in applications** - Ramtron assumes no responsibility or liability for the use of any of its products, and conveys no license or title under any patent, copyright or mask work right to these products and makes no representations or warranties that these products are free from patent, copyright or mask work right infringement unless otherwise specified. Customers are responsible for product design and applications using Ramtron parts. Ramtron assumes no liability for applications assistance or customer product design.

**Life support** — Ramtron products are not designed for use in life support systems or devices. Ramtron customers using or selling Ramtron products for use in such applications do so at their own risk and agree to fully indemnify Ramtron for any damages resulting from such applications.

PC is a trademark of Koninklijke Philips Electronics NV.