

Extended 8051 Support

During the past year, over 100 new 8051 devices have been released by 20 different silicon vendors. Keil Software is constantly enhancing our C51 products and adding support for these new devices.

Today, the Keil μ Vision2 Debugger supports debugging for almost every 8051 derivative using either the ROM monitor that is easily integrated into user hardware or the simulator that provides **full peripheral simulation**.

New Keil Optimizing Linker

Optimization levels in C51 Version 6 have set new standards for code density on 8051 applications.

Optimizer Level 10 (available August 2001) adds application-wide optimizations performed during the link phase. When enabled, the linker optimizes common code sequences throughout your entire program and replaces LJMP and LCALL instructions with AJMP and ACALL wherever possible.

Support for Extended 8051 Devices

The Keil PK51 Professional Developer's Kit includes support for new devices that extend the address space of the 8051. One such device, the new Philips 80C51MX **Memory eXtension**, is fully 80C51 code compatible and is optimized for C language programming.

Using the new Keil CX51 Compiler, variables may be located anywhere in the 16MB address space using the **far** and **const far** memory types. Additional CPU instructions support inline code for 3-byte pointer memory access. When recompiled, existing C51 programs are smaller and execute faster.

C51

Version 6.11
Extended Device Support

C251

Version 3
 μ Vision2 IDE & Debugger

C166

Version 4.11
with CAN Simulation

C51 Optimizer Benchmarks (Code Size)

Optimizer Level 6 (C51 V5)

Optimizer Level 9 (C51 V6)

Optimizer Level 10

C51 Version 6 reduces program size 5-15% by reusing common entry code and common block subroutines.

PK51 includes the LX51 Linker with Advanced Linker Optimization Technology that reduces code size another 10-25%.

This Newsletter Contains Information About:

C51 Extended Memory Support	Page 2
Dallas Semiconductor DS80C390 Support.....	Page 2
Philips MX Device Support.....	Page 2
C251 Version 3	Page 3
CAN Hardware Simulation.....	Page 4
CAN Message Traffic Simulation.....	Page 5
List of Supported Chips	Page 6
MCB167-NET Evaluation Board	Page 7
TCP/IP Support.....	Page 7

Extended Memory Support

The Keil C51 Compiler now provides two user-configurable memory types, **far** and **const far**, for accessing the extended address spaces of many new 8051 devices such as:

- Dallas Semiconductor DS80C390,
- Philips 80C51MX,
- Analog Devices ADuC812.

Alternatively, you may use **far** and **const far** to access special memory areas like:

- FLASH memory,
- EEPROM,
- Strings in code-banked ROM,
- Any weird memory configuration.

Once configured, you may use variables in these memory spaces as easily and transparently as variables in the standard 8051 memory spaces.

Configuration Details

Accesses to **far** and **const far** variables are handled automatically for the Dallas 390 and Philips MX devices. For other devices, these memory types are handled by eight Extended Memory Access Routines which you may customize.

These routines receive the address to be accessed as a 3-byte generic pointer passed in R1, R2, and R3. The value in R3 indicates the *memory type* and, in the case of address spaces larger than 64K, also forms part of the address.

The Extended Memory Access Routines need handle only the memory type values 0x02-0xFD. The other memory types are handled by the Keil run-time library.

Software Requirements

Programs that use **far** or **const far** must be linked with the LX51 Extended Linker/Locator which is supplied only with the PK51 Professional Developer's Kit.

Keil 8051 Development Tool Kits

Components	PK51	DK51	CA51
MACRO Assembler	✓	✓	✓
C Compiler	✓	✓	✓
BL51 Code Banking Linker	✓	✓	✓
µVision2 IDE	✓	✓	✓
µVision2 Debugger	✓	✓	
LX51 Extended Linker	✓		
Support for Extended Devices	✓		

This table lists the components that are included in each of the Keil 8051 Development Tool Kits.

Extended Memory Access Routines

Load Routine	Store Routine	Size of Value Accessed
?C?CLDXPTR	?C?CSTXPTR	1 byte (char)
?C?ILDXPTR	?C?ISTXPTR	2 bytes (int)
?C?PLDXPTR	?C?PSTXPTR	3 bytes (generic pointer)
?C?LLDXPTR	?C?LSTXPTR	4 bytes (long)

*User-configurable memory access routines allow you to customize the **far** and **const far** memory types.*

Values for Generic Pointer Memory Types

R3 Value	Memory Type	Address Range
00	data & idata	1:0x00-1:0xFF
01	xdata	X:0x0000-X:0xFFFF
02-7F	far	X:0x010000-X:0x7E0000
80-FD	far const	C:0x800000-C:0xFD0000
FE	pdata	One 256-byte page in XDATA
FF	code	C:0x0000-C:0xFFFF

*Generic pointers allow access to all memory areas including user-defined **far** and **const far** memory.*

The 251 architecture, a high-end 8-bit CPU, is 100% backward compatible with the 8051. The 251 supports up to 16MB of address space and an enhanced 8/16/32-bit instruction set.

Chip Vendors

Atmel Wireless & Microcontrollers and Intel offer several 251 devices that are 8051 pin-compatible. These chips provide an efficient upgrade path for existing 8051 designs and a true alternative to full 16-bit migration.

Performance and Compatibility

The 251 is designed to provide optimal performance when used with the C language. For example, 251 devices provide 64K of stack space, efficient stack addressing modes, linear memory space, a fast page-mode bus, and on-chip peripherals that are compatible with the 80C51Fx. Code that was written for the Keil C51 compiler can easily be reused on the 251.

C251 Compiler Features

The Keil C251 Compiler comes with high-speed single precision and double precision floating-point arithmetic support and offers two new memory models: **Tiny** and **XTiny**. The **XTiny** memory model is similar to the **LARGE** memory model but uses the 251's efficient memory addressing. With these memory models the code size of existing 8051 applications can be drastically reduced.

More information may be found at www.keil.com/appnotes in the following application notes:

- **I16:** Optimum Code Generation for 251 Applications Larger Than 64K.
- **I17:** Porting 8051 C Programs to the 251 Family.

251 Device Overview

Device	Clock Speed	On-Chip RAM	On-Chip ROM	I/O Pins	Counter Timer	Interrupt #/Level	Serial
Intel 80C251Sx	16 MHz	1056	16K	32	3+PCA	7/4	I
Intel 80C251Tx	24 MHz	1056	16K	32	3+PCA	8/4	2
AtmelWM C251G1	24 MHz	1056	16K	32	3+PCA	9/4	I+SPI
AtmelWM C251G2	24 MHz	1056	32K	32	3+PCA	9/4	I+SPI
AtmelWW C251SC (Smart Card)		4608	256K (FLASH)		2	7/4	I+SPI

8051 / 251 Microcontroller Comparison

Execution Speed

Program Code Size

Sieve Benchmark



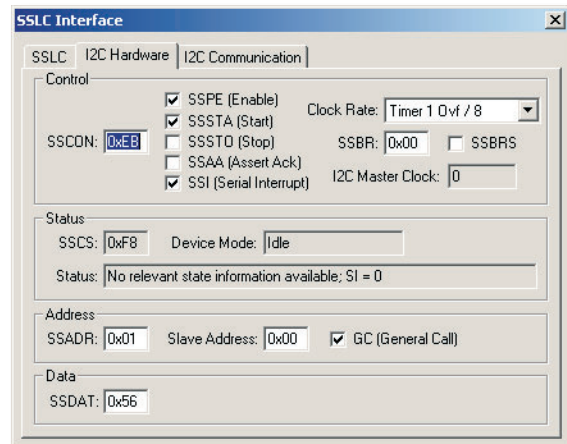
Dhrystone Benchmark



Whetstone Benchmark



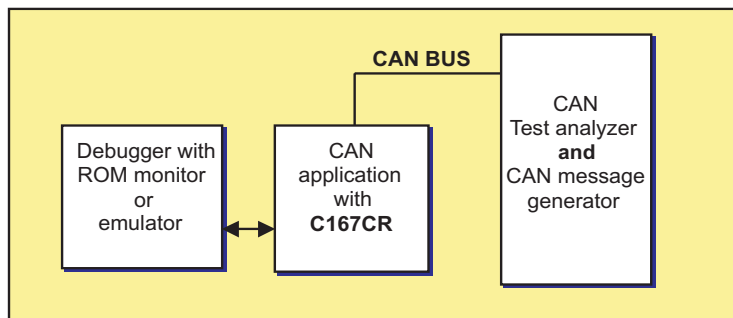
Execution times were measured at 12 MHz using a classic 8051 and a 251 operating in paged mode.
Memory Models: C51 Compiler (LARGE), C251 Compiler (XTINY)



The μ Vision2 Debugger allows you to simulate the on-chip peripherals of most 251 devices including the I2C Hardware of the Atmel Wireless & Microcontrollers 8xC251G1D

The Keil μ Vision2 IDE includes a robust source code editor and a universal debugger that may be used as a Simulator or Remote Target Debugger. The latest release of the μ Vision2 Debugger even includes simulation for a number of devices with on-chip CAN controllers.

CAN is a robust protocol for connecting multiple microcontroller systems. To test a CAN application, a complete system needs to be constructed including a CAN analyzer that generates the messages of other applications. This requires a complex hardware configuration as shown in the following figure.



In the new release, μ Vision2 **simulates a complete CAN environment using only your PC.** The debugger allows you to simulate the microcontroller hardware including memory and all on-chip peripherals.

Simulation Support

Simulation support is provided by the μ Vision2 Debugger in several ways.

- Dialog boxes allow you to view status and change the SFRs used for CAN communications.
- VTREGs (virtual target registers) allow you to change and monitor CAN traffic and status from the command line or from debugger functions you write.
- CAN messages may be generated on-the-fly using a dialog or debugger function. You may even write debugger functions that respond to output CAN messages from your application.

Input Signals

μ Vision2 allows you to simulate complex I/O signals by using VTREGs. VTREGs give you access to the microcontroller I/O and allow simple debugger commands like:

```
AIN0 = 3.5 // Set Analog Input AIN0 to 3.5 Volts
```

You may automatically generate input signals with signal functions. The signal function VoltageRamp increases the input voltage on AIN0 every 0.1 seconds.

```
SIGNAL void VoltageRamp (void) {
    float f;

    for (f = 0; f < 5.0; f += 0.1) {
        AIN0=f; /* current in volts */
        swatch (0.1); /* run app for 0.1 sec */
    }
}
```

This signal function is written using the μ Vision2 debug language and runs in parallel to the simulated microcontroller

μ Vision2 Simulates These CAN Microcontrollers

Atmel Wireless & Microcontrollers 89C51CC01,89C51CC02
Infineon C161CS, C164, C167CR, C167CS, ...
Infineon C505C, C515C, ...
Philips 80C591 (coming soon)
ST Microelectronics ST10167, ST10168, ST10169, ST10269

application. The signal function is synchronized using built-in debugger functions that allow the signal to execute with an exact timing relationship to the microcontroller program. The timing works regardless of breakpoints or whether or not you single step through your application.

The following table lists the built-in synchronization functions that are available in the μ Vision2 Debugger.

Debugger Function	Description
twatch	Delays a signal for the specified number of CPU cycles.
swatch	Delays a signal for the specified number of CPU seconds.
rwatch	Delays a signal until the specified address is read by the target program.
wwatch	Delays a signal until the specified address is written to by the user program.



N 2

The μ Vision2 Debugger allows you to manually create CAN network traffic using a dialog box. However, debug and signal functions (functions you create to run in the debugger) provide more ways to generate and analyze CAN messages in your simulated target system.

Debug and Signal Functions

You may create debug and signal functions that use VTREGs to access the simulated CAN controller. For example, you can create a debug function that sends a CAN message with three specific data bytes to your application program.

```
FUNC void SendMessage (void) {
    CANOL = 3; /* 3 Byte Message */
    CANOBO = 1; /* 1st data byte */
    CANOB1 = 2; /* 2nd data byte */
    CANOB2 = 3; /* 3rd data byte */
    CANOID = 0x44022 /* CAN ID */
}
```

Signal functions may be used to repeat messages. This is often required in CAN applications. The following example repeats a message 10 times at 1-second intervals.

```
SIGNAL void MessageRepeat (void) {
    int i;
    for (i = 0; i < 10; i++) {
        SendMessage(); /* send message */
        swtch (1.0); /* run app for 1 second */
    }
}
```

You may even create signal functions that respond to CAN messages transmitted by your target program.

CAN Application Notes

Keil Software provides several application notes that discuss CAN programming. Check for them on the web at www.keil.com/appnotes:

- I15: Programming the Infineon C167CR CAN Interface.
- I47: CAN Simulation with the μ Vision2 Debugger.

What is CAN?

CAN (Controller Area Network) is a serial data bus for real-time applications. It was originally developed for use in the automotive industry but is receiving widespread use in a wide variety of embedded applications where high-speed communication is required.

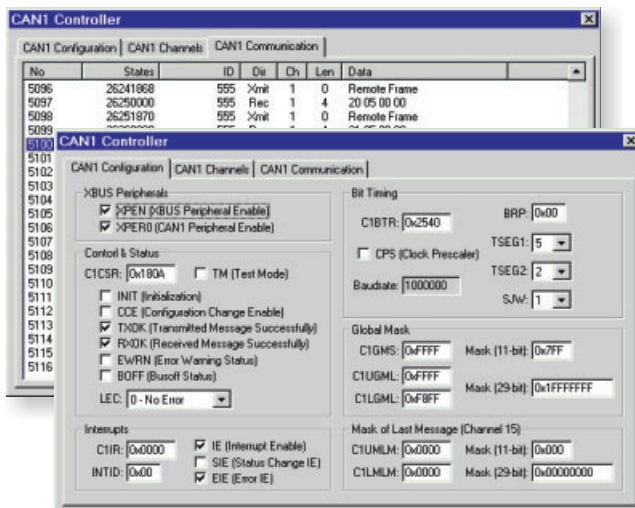
CAN is a serial bus that offers robust error handling. The number of nodes in a CAN network is limited only by the electrical characteristics of the bus.

In a CAN network, all nodes are peers and each receives every message transmitted on the bus. There are no node IDs, so messages do not include source and destination IDs.

CAN messages consist of an 11-bit or 29-bit **message ID** field and a 0-8 byte **data** field. The message ID serves several purposes and is of supreme importance. Message IDs are assigned by the system designer. Typically, every event or command in the system has a unique message ID. Parameters or data are sent in the data field.

Numerous silicon vendors are adapting CAN controllers for their microcontrollers. Intel, Philips, Infineon, Dallas Semiconductor, and Atmel Wireless & Microcontrollers are just a few of the companies who have CAN-based processors.

For more information about CAN: www.keil.com/can.



The status of the CAN controller is shown in dialog boxes. The CAN Communication page displays the CAN messages in your system.

The following devices (over 300 in all) are supported by the Keil Software development tools for the 8051, 251, and C16x/ST10 microcontroller families. Learn more about these devices from the Device Database™ at www.keil.com/dd.

- **Acer Lab** (8051 Family)
M6759
- **Analog Devices** (8051 Family)
ADuC812, ADuC824
- **Atmel** (8051 Family)
87F51, 87F51RC, 87F52, 89C1051, 89C1051U, 89C2051, 89C4051, 89C51, 89C52, 89C55, 89F51, 89F52, 89LS53, 89LS8252, 89LV51, 89LV52, 89LV55, 89S4D12, 89S53, 89S8252
- **Atmel Wireless & uC** (8051 Family)
80C31, 80C31X2, 80C32, 80C51, 80C51FP1, 80C51I2, 80C51RA2, 80C51RD2, 80C51U2, 80C52X2, 80C54X2, 80C58X2, 83/87C51RB2, 83/87C51RC2, 83/87C51RD2, 83/87C51U2, 83/87C52X2, 87C51, T83C5101, T83C5102, T87C5101, T87C5111, T87C5112, T89C51CC01, T89C51CC02, T89C51IB2, T89C51IC2, T89C51RB2, T89C51RC2, T89C51RD2
- **Atmel Wireless & uC** (251 Family)
87C251G1A, 8xC251A1, 8xC251G1D, 8xC251G2D
- **Cygnal Integrated Products** (8051 Family)
C8051F00x, C8051F01x
- **Cypress Semiconductor** (8051 Family)
EZ-USB (AN21XX), EZ-USB FX (CY7C646XX), EZ-USB FX2 (CY7C68XXX)
- **Dallas Semiconductor** (8051 Family)
DS5000(FP), DS5000(T), DS5001(FP), DS5002(FP), DS5240, DS80C310, DS80C320, DS80C323, DS80C390, DS87C520/DS83C520, DS87C530, DS87C550, DS89C420
- **Hyundai** (8051 Family)
GMS90C32, GMS90C51, GMS90C52, GMS90C54, GMS90C56, GMS90C58, GMS90L32, GMS90L51, GMS90L52, GMS90L54, GMS90L56, GMS90L58, GMS97C1051, GMS97C2051, GMS97C51, GMS97C51H, GMS97C52, GMS97C52H, GMS97C54, GMS97C54H, GMS97C56, GMS97C56H, GMS97C58, GMS97C58H, GMS97L1051, GMS97L2051, GMS97L51, GMS97L52, GMS97L54, GMS97L56, GMS97L58
- **Infineon** (8051 Family)
C501, C501G-1R / -E, C501G-L, C504-2R / -2E, C504-L, C505-2R, C505-L, C505A-4E, C505C-2R, C505C-L, C505CA-4E, C505L-4E, C508, C509-L, C511-R, C511A-R, C513-R, C513A--2R, C513A--L, C513A--R, C515-1R, C515-L, C515A-4R, C515A-L, C515B-2R, C515C-8R / -8E, C515C-L, C517A-4R, C517A-L, C540U-E, C541U-2E, SAB 80C515, SAB 80C515A, SAB 80C517, SAB 80C517A, SAB 80C535, SAB 80C537, SAB 83C515A-5, SAB 83C517A-5, SDA 30C16x/26x, SDA 555X TVText Pro
- **Infineon** (C16x/ST10 Family)
C161CS, C161JC, C161JI, C161K, C161O/OR, C161PI, C161RI, C161U, C161V, C163 (all Variants), C164CI, C164CL, C164SI, C164SL, C165, C165 UTAH, C166 (all Variants), C167-LM, C167CR-16FM, C167CR-16RM, C167CR-4RM, C167CR-L25M, C167CR-LM, C167CS-32FM, C167S-4RM, C167SR-LM, SDA 6000
- **Intel** (8051 Family)
80/87C51BH, 8031AH, 8032AH, 8051AH, 8052AH, 80C152JA, 80C152JB, 80C152JC, 80C152JD, 80C31BH, 80C51BH, 80C51FA, 80C51GB, 80C51SL, 80L51FA, 80L52, 81/83/87C51SL, 83/87C51FA, 83/87C51FB, 83/87C51FC, 83/87C51GB, 83/87C51RA, 83/87C51RB, 83/87C51RC, 83/87L51FA, 83/87L51FB, 83/87L51FC, 83C152JA, 83C152JC, 87C51FA, 87C54, 87L52
- **Intel** (251 Family)
8xC251SA, 8xC251SB, 8xC251SP, 8xC251SQ, 8xC251TA, 8xC251TB, 8xC251TP, 8xC251TQ
- **ISSI** (8051 Family)
IS80C31, IS80C32, IS80C51, IS80C52, IS80LV31, IS80LV32, IS80LV51, IS80LV52, IS89C51, IS89C52
- **Mentor Graphics Co.** (8051 Family)
M8051, M8051 E-Warp, M8051 Warp, M8052
- **OKI** (8051 Family)
80C154, 80C31F
- **Philips** (8051 Family)
80C32, 80C528, 80C552, 80C554, 80C562, 80C575, 80C591, 80C592, 80CE558, 83/87C451, 83/87C528, 83/87C552, 83/87C554, 83/87C557, 83/87C562, 83/87C575, 83/87C750, 83/87C751, 83/87C752, 83/89CE558, 83C591, 83C592, 87C552, 87C591, 87C652, 87C654, 87LPC762, 87LPC764, 87LPC767, 87LPC768, 87LPC769, 89C660, 89C662, 89C664, 8xC51/8xC51L, 8xC51FA/8xL51FA, 8xC51FB/8xL51FB, 8xC51FC/8xL51FC, 8xC51MA, 8xC51MB, 8xC51RA+, 8xC51RB+, 8xC51RB2, 8xC51RC+, 8xC51RC2, 8xC51RD+, 8xC51RD2, 8xC52, 8xC54, 8xC58
- **SST** (8051 Family)
SST89C54, SST89C58, SST89C59, SST89F54, SST89F58
- **ST Microelectronics** (C16x/ST10 Family)
ST10CT167, ST10F163, ST10F166, ST10F167, ST10F168, ST10F269, ST10R163, ST10R165, ST10R167, ST10R262/272
- **TDK** (8051 Family)
73M2910L
- **TI** (8051 Family)
TUSB3200
- **Triscend** (8051 Family)
TE502, TE505, TE512, TE520, TE532
- **Winbond** (8051 Family)
W77C32, W77E468, W77E58, W77LE58, W78C32B, W78C33B, W78C354, W78C374, W78C438C, W78C516, W78C51D, W78C52D, W78C54, W78C58, W78C801, W78E354, W78E374, W78E51B, W78E52B, W78E54B, W78E58, W78L32, W78L33, W78L51, W78L52, W78L54, W78L801, W78LE51, W78LE52, W78LE54, W78LE812

MCB167-NET Board

The Keil MCB167-NET evaluation board is a high-speed single-board computer that supports all variants of the Infineon C167 and ST Microelectronics ST10-167/168. It offers two CAN interfaces and an Ethernet controller for embedded TCP/IP applications.

The MCB167-NET runs at speeds up to 50MHz and is in-system programmable using I67 FLASH derivatives or external FLASH memory devices.

TCP/IP Stacks

Using the Keil C166 C Compiler, companies like US Software (www.ussw.com) and CMX (www.cmx.com) have developed TCP/IP stacks and other Internet solutions for the C16x/ST10 microcontroller family.

CMX provides a simple test application (described in Keil Application Note I55) that allows you to control the LEDs on the MCB167-NET board using a web browser.

Debugging Programs

Debugging programs for the MCB167-NET is simple. You may use the μ Vision2 Debugger to download and test your application programs using the MONI66 Target Monitor.

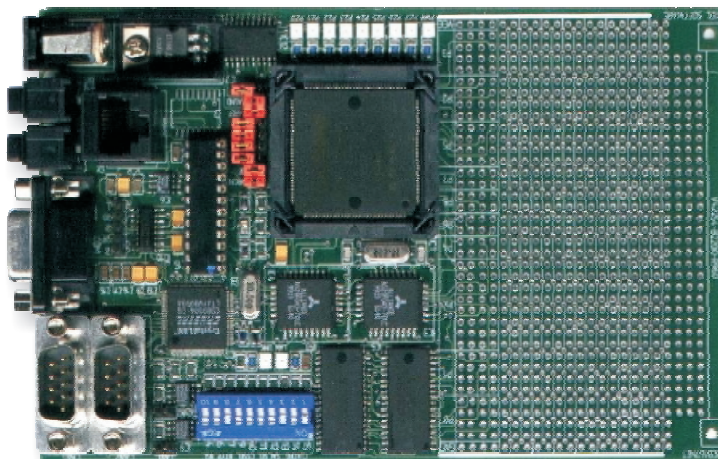
MONI66 comes pre-configured for a number of boards including those available from Keil Software, Phytex, Forth, and Infineon. Simply select the proper monitor configuration and the boot loader loads the monitor which, in turn, downloads your program to target memory.

Using the monitor you may single-step through your program, watch program variables and structures, set breakpoints, and debug even the most sophisticated programs.

For more information about MONI66, refer to www.keil.com/c166.



The CMX MicroNet TCP/IP stack allows you to create internet-ready applications for C16x/ST10 devices.



The MCB167-NET Evaluation Board supports CAN and provides an on-board Ethernet controller for testing your TCP/IP applications.

Technical Information

- Supports all C167 and ST10-F168 variants up to 50 MHz.
- 1 megabyte of high-speed RAM is provided (access time < 25 ns).
- Supports an optional 1 megabyte of FLASH memory.
- Includes an ethernet controller (Crystal CS8900A).
- Supports the dual CAN interface.
- Prototyping area: 3.75" x 2" (95mm x 50mm)

At Keil Software we **don't** make a compiler for every different architecture. We concentrate on just a few chip families that we can support very well. That's why Keil C compilers lead the way in code quality, flexibility, and reliability. Most of our customers will agree that we have the best development tools for the 8051, 251 and 166 microcontroller families.

Another plus is service and support. All Keil products come with one year of free technical support and updates.

On our web site at www.keil.com we provide current product information and a multitude of application notes. The technical support knowledge base contains over 1300 articles that answer most technical questions while the discussion forum allows you to share questions and ideas with colleagues around the world.

Only the μ Vision2 Debugger includes full peripheral simulation for on-chip peripherals of more than 100 different controllers. This makes it possible to test your whole target application on your desk - even before hardware is available. The following table exemplifies this with the Philips LPC peripheral set:

Peripherals Simulated	762	764	767	768	769
Interrupt System	✓	✓	✓	✓	✓
I/O Ports (all modes)	✓	✓	✓	✓	✓
Timers (including pin toggle)	✓	✓	✓	✓	✓
Watchdog Timer	✓	✓	✓	✓	✓
Pulse Width Modulation				✓	
Enhanced Serial UART	✓	✓	✓	✓	✓
I2C Interface	✓	✓	✓	✓	✓
Keypad Interrupt	✓	✓	✓	✓	✓
Analog Comparator	✓	✓	✓	✓	✓
A/D Converter			✓	✓	✓
D/A Converter				✓	✓
CPU Slowdown Modes	✓	✓	✓	✓	✓



1501 10th Street, Suite 110
Plano, TX 75074

Phone 800-348-8051
972-312-1107
FAX 972-312-1159

BULK RATE
U.S. Postage
PAID
Dallas, TX
Permit #4910