Development Tools Guide 2005

Microprocessor

ARM®

TriCore™

C167

XC166

ST10

MSP430

8051/C500

68HC05/11

68HC08

68HC12

MC9S12

ST7

CR16C

x86

Embedding Software Quality
Welcome to the Microprocessor Development Tools Guide

You will find all the tools and support services required to ensure the success of your project and keep development time as short as possible. To simplify your selection of the perfect tool, we have created the Hitex SCIL categories (Safety-Complexity-Integrity-Level). Using a testing tool that is under-specified for the task in hand can prejudice quality and waste time and money.

The limited space available here is complemented by more comprehensive background information and detailed specifications on our website. If you have any questions on anything that appears in this publication, please contact us directly.

Your Hitex Development Tools Team

Contents

Introduction ........................................................................................................................................ 2
Getting the right tool for the job... ................................................................................................ 3
Estimating the Safety-Complexity-Integrity Level (SCIL) of your project... ......................... 4
Choosing the Appropriate Debug Tool .................................................................................... 5
HiTOP – Integrated Development Environment for all Tools ........................................... 6
Hitex Development Tools Overview ....................................................................................... 8
Tool chain for ARM ................................................................................................................... 10
C166 Architecture .................................................................................................................... 14
XC166 Architecture .................................................................................................................. 15
ST10 Architecture .................................................................................................................... 16
TriCore Architecture ................................................................................................................ 17
8051 Architecture .................................................................................................................... 18
VCTP Architecture ................................................................................................................... 19
Freescale / ST 8-Bit Architecture ........................................................................................... 20
Freescale 16-Bit Architecture ................................................................................................. 21
MSP430 Architecture ............................................................................................................... 22
C805C Architecture ................................................................................................................ 23
Connecting A Hitex Emulator ... ........................................................................................... 24
USB Protocol Analyzer .......................................................................................................... 25
IEEE1394 (FireWire) Protocol Analyzer................................................................................ 26
Software Quality Tools .......................................................................................................... 27
Tessy ............................................................................................................................................ 27
Development Assistant for C ................................................................................................... 28
RiskCAT and IEC 61508 .......................................................................................................... 29
Embedded C/C++ Compilers ................................................................................................. 29
Keil ............................................................................................................................................ 29
IAR ............................................................................................................................................. 30
Cosmic ....................................................................................................................................... 30
Training and Consulting.......................................................................................................... 31

© Hitex Development Tools 2005
Test and Debug Tools
Getting the right tool for the job...

When starting out on an important new microcontroller development there is a bewildering choice of different types of debug tools available. Choosing the wrong debug type can save money in the short term but later in the project it can become a major obstacle to achieve reliable embedded software. If you are working in a high integrity field the choice of debug approach is especially critical but, even for mainstream commercial projects, getting it right is still important.

Here is a summary of the attributes of the main tool types:

<table>
<thead>
<tr>
<th>Feature</th>
<th>In-circuit emulator</th>
<th>BDM / OCDS / JTAG</th>
<th>Monitor</th>
<th>Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advised for high-integrity systems?</td>
<td>yes</td>
<td>possible</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Debugging capabilities</td>
<td>high</td>
<td>Low w/o Trace</td>
<td>Medium w/ Trace</td>
<td>low</td>
</tr>
<tr>
<td>Initial debugging of new hardware</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Real-time debugging</td>
<td>yes</td>
<td>yes</td>
<td>restricted</td>
<td>no</td>
</tr>
<tr>
<td>Target hardware needed?</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Occupies target hardware resources?</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>n/a</td>
</tr>
<tr>
<td>Trace</td>
<td>yes</td>
<td>seldom</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Performance Analysis</td>
<td>often</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Coverage</td>
<td>often</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

What sort of debugging tool is required for my project?

At the beginning of a project it is worth taking some time to consider how sophisticated the tools required will need to be. The most important factors that influence the choice of debugging tools in particular are the nature of the software i.e. its complexity and real-time content plus the safety implications of any malfunction. To help in the selection of tools, the Hitex Safety-Complexity-Integrity Level (SCIL) rating of the project can be useful. This allows projects to be allocated to one of four SCIL classes, numbered SCIL1 to 4. Please note that the SCIL levels have been defined by Hitex with the goal to facilitate the choice of the proper tool. The Hitex SCIL levels are similar to (but should not be confused with) the SIL1-4 scheme proposed in IEC61508 and DEFStan 00-55/56, but in addition to taking into account the seriousness of the consequences of failure, it also considers the complexity and real-time content of the software. Thus the overall SCIL number indicates both the difficulty of debugging the system and the consequences of any subsequent failure due to undiscovered bugs.
Estimating the Safety-Complexity-Integrity Level (SCIL) of your project...

Almost all professional projects will fit into one of four safety-complexity-integrity levels (SCIL). Assess your project against the following criteria to decide what level of debugging tool is required. The SCIL rating will be determined by the highest group in which your project has a matching attribute. For example, if your system will not cause injury or damage when it fails but has a very strong real-time content, then it might be SCIL3 or SCIL4 rather than just SCIL2. Likewise, if the project is relatively simple but could cause serious injury if it failed, it should still be classed as SCIL3.

**SCIL1:**
Example:
Evaluation of new CPU, pure research, student project
- No significant real-time content
- Less than 15 modules/50 functions
- Failure will cause no loss of life
- Failure will cause no inconvenience
- Failure will not damage associated circuitry
- Not controlling mechanical hardware
- Not closed loop control
- Likely Software Standard: None

**SCIL2:**
Example:
Point of sale equipment, small consumer goods
- Some real-time content
- Failure will cause inconvenience or disappointment to members of the public
- Failure will not seriously damage controlled mechanical hardware
- Not closed loop control
- Likely Software Standard: In-house ISO9001

**SCIL3:**
Example:
Automotive engine management, general industrial motor drives, major consumer goods, non-flight critical aerospace
- Strong real-time content
- Failure will cause serious injury to a number of persons
- Failure will cause major disruption to normal activities of the public
- Failure will seriously damage controlled mechanical hardware
- Failure will destroy associate high current drivers
- Project requires algorithm development
- CAN network is an important component
- PID (or other) control loops
- Likely Software Standard: TickIT, ISO9001
SCIL 4:
Example:
Flight-critical aerospace, life-critical medical systems, transport control systems, hazardous process control systems, automotive braking systems, large motor drives
- Time-critical content with mechanically determined task scheduled deadlines
- Failure will kill one or more persons
- Failure will destroy immediate controlled mechanical hardware and associated systems
- Failure will destroy associate high current drivers and electromechanical devices
- Mission or time-critical data derived from CAN network
- Project requires complex algorithm development
- Complex, multiple PID (or other) adaptive control loops
- Likely Software Standard: TickiT, ISO9001, DO178/B, DefStan 00-55/56, TS0-C106, IEC 61508

Choosing the Appropriate Debug Tool
Now that you have established the SCIL rating of your application, use the following to determine the correct performance level for the debug tool type you are considering:

Monitor or simulator debugger
Basic version of in-circuit emulator without trace or BDM/JTAG/OCDS debugger without trace
In-circuit emulator with trace buffer or BDM/JTAG/OCDS debugger with trace
In-circuit emulator with large trace buffer, coverage and real-time performance analysis
Monitor or simulator debugger not permitted

Simple Profiling
Displays the proportion of CPU time spent in user-specified address ranges but absolute runtimes not available. Gives a quick and approximate guide to program bottlenecks in medium integrity commercial projects.

Real-Time Performance Analysis
Displays run times in microseconds for user-specified address ranges as a table or histogram of max, min and average values, plus shows number of executions. Also able to measure distributions of run times for single objects. An accurate tool for use in high-integrity or hard real-time projects.

Make a note of the SCIL rating of your project and use it as a guide to which development and debug tools have the right capabilities and facilities
HiTOP – Integrated Development Environment for all Tools

HiTOP is IDE and user interface for all debugging tools from Hitex. It is available for in-circuit emulators, BDM / JTAG / OCDS tools, simulators etc. For each type of debugging tool, HiTOP gives access to the specific debugging tool features. The exact functionality of HiTOP depends on the debug system, e.g. a pure JTAG debugger will not feature a Trace display, but an in-circuit emulator will.

More Than 20 Architectures Supported

HiTOP users can move between more than 20 different microprocessor architectures.

More Than A Standard Debugger

HiTOP conforms to industry standards like Microsoft Visual C++ and features drag&drop, docking windows, toolbars editable tabs, and explorer-like trees to structure information. HiTOP integrates an editor featuring syntax colouring.

When it comes to debugging, HiTOP features of course all what you can expect from a nowadays source level debugger, like stepping, easy breakpoint control, read and write of registers and memory, thus allowing program download and flash programming. Furthermore, a lot of small, but very useful features ease debugging, e.g. automatic display of local variables when stepping into a function. Also user-definable windows allow the “plain text” access to the Special Functions Registers (SFR) of the microcontroller.

Integrated Development Environment

HiTOP includes a project management facility and provides an integrated development environment (IDE) for compilers like GNU, Keil, Tasking, IAR, Cosmic and others.

RTOS Support

HiTOP supports many real-time operating systems (RTOS) for use on various microcontroller architectures. This includes the displaying of objects of the RTOS (such as tasks, semaphores, etc.)

Link To Partner Tools

HiTOP links to many other partner tools, e.g. Tessy from Razorcat.
**Smart Emulator – Intelligent Resource Manager**

HiTOP manages the resources of the in-circuit emulator in an intelligent way, thus freeing the user from selecting the appropriate emulator feature. An example: A normal code breakpoint halts the emulation when a specific instruction of the application is executed for the first time.

HiTOP automatically selects the optimal feature provided by the specific emulator or microcontroller for this purpose, e.g. a software breakpoint, a breakpoint established by the hardware breakpoint logic of the in-circuit emulator, or a breakpoint implemented by the on-chip debug system of the microcontroller. The selected kind of breakpoint heavily depends on the respective emulator or microcontroller in use.

If the user now wishes the same breakpoint to halt the emulation not on the first occurrence, but, say, on the fifth, the user simply extends the definition of the already existing breakpoint. HiTOP takes care of the necessary actions.

**HiSCRIPT Test Harness Language**

HiSCRIPT is an integral part of the HiTOP debugger used on all Hitex emulators. It allows complete control of the debug system’s operation and complex test harnesses to be created that can exercise embedded software in a consistent manner, using test data derived from ordinary PC disk files.

Test harness techniques are required by most current software standards and guidelines and HiSCRIPT has been designed to allow engineers to easily build powerful and flexible test routines that DO NOT influence target program operation. This is in stark contrast to the software probe-based approaches used in conventional test suites that inevitably alter the program under test from what will be shipped to the customer.

Via the Remote Control Interface (RCI) of HiTOP, any remote program may control the in-circuit emulator by using HiSCRIPT.
The MX Range
Hitex MX have multi-channel trigger systems able to detect FETCH, READ and WRITE accesses across address and data ranges. External logic probe inputs allow port and other IO signals to be combined with CPU-based states. Unique to Hitex are opcode decoders that allow on-chip RAM READ/WRITE triggering even on the 8051 family plus interrupt servicing during halted emulation. All MX units have break-before-execute breakpoints to eliminate overruns past function calls.

Suitable for: 

The AX Range
Hitex AX class emulators build on the feature set of the MX series by adding more triggers and crucially for high-integrity applications, non-intrusive coverage and performance analysis. These two additions are essential for those projects conducted to a recognised software quality standard such as the DTI’s TickIT and MOD’s DEFStan 00-55/56.

Suitable for: 

The Modular DProbe/DBox Range
The DProbe range use advanced ASIC techniques to give a smaller footprint, higher speed and greater modularity. DProbes are hand-held units that contain the emulation memory, breakpoint and triggering systems directly on top of the CPU, allowing operation up to maximum CPU clock frequency, so providing basic emulator functions. For projects at higher integrity levels (SCIL3 and SCIL4), DProbes become the emulation probes of a larger system, based on a DBox mainframe unit, which yields a full laboratory software analysis system.

DProbe suitable for: 

DBox-based systems represent the current emulation “state of the art”, offering auto-documented code and data coverage measurements plus non-statistical performance analysis. These two additions are essential for those projects conducted to a recognised software quality standard such as the DTI’s TickIT and MOD’s DEFStan 00-55/56.

DProbe with DBox suitable for:
**Tanto™ is all you need**

For microcontrollers with on-chip debugging features, a highly modular and universal system like Tanto is a must. Changing over from one microcontroller to another requires nothing more than a swap of the Tanto PL (Port Link) and starting the corresponding HiTOP user interface. The same hardware can then be used throughout.

What’s outstanding about Tanto is the fact that the entry-level product, Tanto Base, which is itself a high performance debugging system, can be expanded by the Tanto PT (Port Trace) to a debugger with port trace.

- Tanto Base suitable for:
- Tanto Port Trace with port trace module

**The Tantino™ Tool Class**

Tantino is the newest member of test and analysis tools from Hitex. Tantino is available for various microcontroller architectures with on-chip debugging features.

The Tantino system consists of a small module with a debug cable that allows the direct connection of the test and analysis tool to the JTAG, BDM, Nexus port in the target system. The system communicates to the host via USB.

- Suitable for:
Tool chain for ARM®
ARM7, ARM9, ARM9E, ARM11, XScale

Chip Manufacturers
various

Supported Compilers
ARM Developer Suite
ARM GNU Tool Chain
Green Hills
HighTec
IAR
Keil
Metrowerks

Host Interface: USB, Ethernet

<table>
<thead>
<tr>
<th>Content</th>
<th>Hitex Developer Suite</th>
<th>Starter Kits</th>
<th>Tantino for ARM</th>
<th>Tanto for ARM</th>
<th>Tanto Port Trace for ARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITOP IDE</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>HISIM</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>GNU C</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>RTOS support</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Tessy</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Tantino JTAG Tool</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Tanto JTAG Tool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Tanto Port Trace</td>
<td>Tool</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Evaluation Board</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Starter Kits with evaluation boards and examples for different architectures are available. Some configurations include code size or time limited versions.

Hitex Developer Suite for ARM
This suite includes all software tools you need to start the development of your ARM projects:

- HiSIM® ARM v4T instruction set simulator
- HITOP® powerful IDE and debugger
- GNU C compiler
- Tessy® automated unit test tool

* 16k code-size limited version upgradable to full version

Hitex Instruction Set Simulator for ARM – HiSIM
This simulator lets you debug your application under the Hitex HITOP IDE:

- includes GNU C Compiler
- Support ARM and Thumb mode of the ARM v4T architecture
- Little and Big Endian Support
- Support of high and low exception vector table
- Preload of simulated memory
- Logging of simulator actions
- Counters for simulated instructions and cycles
- IRQ and FIQ trigger control
- Breakpoints for code, data read, data write, data read/write
**Hitex Starter Kits for ARM**

The Hitex Starter Kits contain everything you need to set up your first ARM project at low cost:

- Tantino for ARM JTAG debugger
- Evaluation Board for ARM with Atmel or Philips ARM7TDMI
- Quick Start Guide
- Installation CD with Hitex Developer Suite* for ARM and examples
- Power supply
- JTAG and power supply cable

* 16k code-size limited versions of HiTOP, HiSIM and Tessy

<table>
<thead>
<tr>
<th>Feature</th>
<th>Starter Kit Atmel</th>
<th>Starter Kit Philips</th>
<th>Starter Kit STMicroelectronics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hitex Developer Suite</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Tantino JTAG Tool</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Cable</td>
<td>USB, JTAG, Power</td>
<td>USB, JTAG, Power</td>
<td>USB, JTAG, Power</td>
</tr>
<tr>
<td>Power Supply</td>
<td>Power supply for evaluation board</td>
<td>Power supply for evaluation board</td>
<td>Power supply for evaluation board</td>
</tr>
</tbody>
</table>
| Evaluation Board         | Atmel AT91M40800 ARM7TDMI Microcontroller:  
  > 8 kByte On-chip SRAM  
  > Vectored Interrupt Controller  
  > GPIO Interface with 32 Lines  
  > 3-channel 16-bit Timer/Counter  
  > 2 USART  
  > Programmable Watchdog Timer  
  > Fully Static Operation up to 40MHz  
  > 512Kbyte SRAM  
  > 32MHz Quartz  
  > Serial Port Driver for 2 Ports  
  > 4 LED’s (user programmable)  
  > 3 buttons (user programmable)  
  > reset button and LED  
  > Power Regulator (5V external power)  
  > 40 Pin Expansion Connector  
  > JTAG Connector | Philips LPC2129 ARM7TDMI Microcontroller:  
  > 16 kByte On-chip SRAM  
  > 256 kByte On-chip FLASH  
  > Vectored Interrupt Controller  
  > GPIO Interface with up to 48 Lines  
  > Two 32-bit Timer  
  > PWM unit  
  > Two UART interfaces  
  > NVIC Interface  
  > Two SPIs  
  > Two CAN interfaces  
  > 4-channel 10-bit AD converter  
  > Programmable Watchdog Timer  
  > Real Time Clock  
  > Fully Static Operation up to 60MHz  
  > 12MHz Quartz  
  > Port Driver for 2 UART Ports and 2 CAN Ports  
  > 8 LED’s (user programmable)  
  > IRQ and Reset button  
  > Power Regulators (min. 6V external power)  
  > Wire-Wrap pins for 47 port signals  
  > JTAG Connector  
  > ETM Trace connector (Mictor connector) | ST Microelectronics STR710F ARM7TDMI  
  > Microcontroller:  
  > 64 kByte On-chip SRAM  
  > 256 kByte On-chip FLASH  
  > Vectored Interrupt Controller  
  > GPIO Interface with up to 48 Lines  
  > Four 16-bit Timer  
  > Four UART Interfaces  
  > Two PMC interfaces  
  > Two SPIs  
  > CAN Interface  
  > USB 2.0 full speed Interface  
  > HDLC Interface  
  > Programmable Watchdog Timer  
  > Real Time Clock  
  > 4-channel 12-bit AD converter  
  > Fully Static Operation up to 16MHz  
  > 16MHz Quartz  
  > Port Driver for 2 UART and 1 CAN Port  
  > 8 LED’s (user programmable)  
  > IRQ and Reset button  
  > Power Regulators (min. 6V external power)  
  > Wire-Wrap pins for 48 port signals  
  > JTAG Connector  
  |
**Tantino for ARM JTAG Debugger**

Tantino for ARM is our compact and cost-effective JTAG debug interface with complete HiTOP support:

- Supports ARM7, ARM9 and ARM9E processor cores
  (ARM11 and XScale in preparation)
- JTAG interface supports target voltages from 1.8 - 5.0V and clocks up to 30MHz
- USB2.0 (full speed) interface
- USB powered
- Fast program download up to 100kByte/s
- Fast FLASH programming for internal and external FLASH devices
- Supports On-chip hardware and unlimited software breakpoints

**TANTINO-ARM: JTAG Debug Interface**

---

**Tanto for ARM**

**Advanced JTAG Development Tool**

Tanto for ARM is our high-performance modular JTAG development tool including Ethernet:

- Supports ARM7, ARM9 and ARM9E processor cores
  (ARM11 and XScale in preparation)
- JTAG interface supports target voltages from 1.8 - 5.0V and clocks up to 30MHz
- USB2.0 (full speed) interface
- Ethernet 10/100MBit interface
- Fast program download with up to 400kByte/s
- Fast FLASH programming internal and external FLASH devices
- Supports On-chip hardware and unlimited software breakpoints
- Expandable with Tanto PortTrace for ARM module

**TANTO-ARM: JTAG Development Tool**

---
**Tanto Port Trace for ARM**

*Advanced real-time debugging with filtered trace.*

- Tanto PortTrace for ARM is an extension module for the Tanto ARM base module for real-time trace capture:
- Support of ETM7 and ETM9
  - with CPU clocks up to 250 MHz
  - with normal and half-rate clock modes
  - with 4- and 8-bit trace port width
- Trace memory for up to 2 million CPU cycles
- Support of ETM configuration
- Trace display with linking to HLL source
- Trace forward and backward stepping

TANTOPT-ARM: JTAG Development Tool with 128 k or 1 MB trace module

<table>
<thead>
<tr>
<th>Feature</th>
<th>Tantino for ARM</th>
<th>Tanto for ARM</th>
<th>Tanto Port Trace for ARM Extension Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM7, ARM9, ARM9E Support</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Debug Interface</td>
<td>JTAG (30 MHz, 1.8-5.0 V)</td>
<td>JTAG (30 MHz, 1.8-5.0 V)</td>
<td>JTAG (30 MHz, 1.8-5.0 V) ARM ETM Trace</td>
</tr>
<tr>
<td>PC Host Interface</td>
<td>USB 2.0 (Full Speed)</td>
<td>USB 2.0 (Full Speed) LAN 10/100 MBit</td>
<td>(via Tanto for ARM)</td>
</tr>
<tr>
<td>Power Supply</td>
<td>USB powered</td>
<td>External, 100-240V</td>
<td>(via Tanto for ARM)</td>
</tr>
<tr>
<td>Debug Support: Level 1: Program Run Control</td>
<td>Level: &gt; run/halt &gt; single step/line step &gt; max. HW breakpoints &gt; unlimited SW breakpoints &gt; watchpoints</td>
<td>Level: &gt; run/halt &gt; single step/line step &gt; max. HW breakpoints &gt; unlimited SW breakpoints &gt; watchpoints</td>
<td>Level2: &gt; ETM7 and ETM9 &gt; CPU clock max. 250 MHz &gt; 4- and 8-bit port normal, half-rate mode &gt; 2 data comparators &gt; 4 address comparators &gt; 2 counters &gt; instruction and data trace filtering &gt; trace memory for up to 2 million CPU cycles &gt; time stamp 25 ns resolution</td>
</tr>
<tr>
<td>HITOP IDE Support</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
**C166 Architecture**

**C16x**

**DProbe167**

- 33 MHz operation (40 MHz option)
- Uses replaceable E3 bondout chip
- Full support of all C167 bus modes
- Semi-ruggedized hand-held format
- Independent time reference
- C/C++ debug support
- Expandable to full software analysis system
- HiTOP debugger
- Press-on adaptation

DProbe167-HT-E3: 1 MB RAM, hardware/software breakpoints, HiTOP

**DTrace167**

- Trace module for DProbe167
- Internal/external code and data trace
- Optional 48-Bit Timestamp

DP167-TRA-1: 64K x 144-bit trace module
DP167-TRA-2: 128K x 144-bit trace module

**DMemory167**

- Dual-port RAM module for DProbe167

DP167-MEM-1: 1 MB dual port RAM module
DP167-MEM-2: 2 MB dual port RAM module

**DBox167**

- Absolute performance and coverage analysis
- 8 channel trigger unit
- Unique timer trigger
- 4-level sequencer
- Up to 512K x 196 bit trace
- Up to 48 external signal inputs
- 48-Bit Timestamps with 10 ns resolution

DBox-167-1: 256K trace
DBox-167-2: 256K trace, 16 MB Code Coverage, Performance
DBox-167-3: 512K trace, 16 MB Code Coverage, Performance Analysis

**Chip Manufacturers**

Infineon

**Supported Compilers**

Cosmic
HighTec
Keil
Tasking

**Host Interface:**

PCI, PC-CARD
**DProbeXC**

**XC161Cl, XC164CS, XC164CM, XC167Cl and future XC16x derivatives**

- Emulation device, fully transparent, non-intrusive debugging
- Supports voltages from 2.7 up to 5.5 Volts
- 1 MByte emulation memory, mapable in complete 16 MByte address range, selectable granularity (1 kByte up to 32 kByte), zero wait state access, write protection (ROM emulation)
- Selectable ROM size up to 512 kByte, on-chip Flash support
- 32 hardware execution breakpoints (break before make)
- 256 software breakpoints
- On-the-fly accesses to registers, internal and external RAM and Flash memory
- Programmable clock generator (XTAL1), XTAL3 support integrated on DProbeXC
- Enable/disable control for RESETIN, RESETOUT, NMI, XTAL2
- Adaptation via quad connector system to all existing adapter and chip packages. Compatible with Hitex PressOn technology

DPROBEXC: 1 MB RAM, hardware/software breakpoints, HITOP

---

**DTraceXC**

- Trace module for DProbeXC
- 128k frames, 192 channels, internal/external code and data
- Timestamp resolution 10 ns, 48 bit
- Break, trigger, qualify trace, start/stop trace, change level
- Optional 8 I/O channel data probe
- 4 complex triggers, up to 32 trace regions
- Performance analysis, display of interrupt levels

DTRACEXC-128: 128 k trace module

---

**DCoverageXC**

- Code coverage in specific memory ranges
- Data coverage in specific memory ranges
- 16 x 128k memory available, can be mapped in whole address range of the XC166
- Out of range detection
- Timer trigger

DCOVERAGEXC: Coverage module for DProbeXC

---

**TantinoXC/TantinoXC Plus**

**XC161Cl, XC164CS, XC164CM, XC167Cl and future XC16x derivatives**

- JTAG interface for OCDS (2 performance options available)
- Fast break for access on-the-fly
- On-chip Flash support
- OCDS hardware breakpoint, software breakpoints unlimited
- Task specific debugging
- Adaptation via PressOn adapter possible, if no OCDS connector is implemented

TANTINOXC: OCDS unit, HITOP
TANTINOXCP: OCDS unit, HITOP

---

**Chip Manufacturers**

- Infineon

**Supported Compilers**

- Cosmic
- HighTec
- Keil
- Tasking

---

**Host Interface:**

- PCI, PC-CARD
- USB
ST10 Architecture

Chip Manufacturers

STMicroelectronics

Supported Compilers

Keil
Cosmic
HighTec
Tasking

Host Interface:
PCI, PC-CARD, LPT

DProbeST10

ST10163, ST10R165, ST10x167, ST10F168, ST10R172, ST10R262, ST10F269,
ST10R272/L, ST10F276, ST10x275, ST10x280

- 50MHz operation
- Full support of all ST10 bus modes
- Support for MAC-unit
- Semi-ruggedised hand-held format
- Independent time reference
- C/C++ debug support
- Expandable to full software analysis system
- HiTOP debugger
- Press-on adaptation

DPROBEST10: 1 MB RAM, hardware/software breakpoints, HiTOP

DTraceST10 and DMemST10

For the DProbeST10 the DTrace167 and DMem167 modules are used

DBoxST10

For technical specifications see DBox167

DBOX-ST10-1: 256K trace
DBOX-ST10-2: 256K trace, 16MB Code Coverage, Performance Analysis
DBOX-ST10-3: 512K trace, 16MB Code Coverage, Performance Analysis

Chip Manufacturers

STMicroelectronics

Supported Compilers

Keil
Cosmic
HighTec
Tasking

Host Interface:
PCI, PC-CARD, LPT

DProbeST10

ST10163, ST10R165, ST10x167, ST10F168, ST10R172, ST10R262, ST10F269,
ST10R272/L, ST10F276, ST10x275, ST10x280

- 50MHz operation
- Full support of all ST10 bus modes
- Support for MAC-unit
- Semi-ruggedised hand-held format
- Independent time reference
- C/C++ debug support
- Expandable to full software analysis system
- HiTOP debugger
- Press-on adaptation

DPROBEST10: 1 MB RAM, hardware/software breakpoints, HiTOP

DTraceST10 and DMemST10

For the DProbeST10 the DTrace167 and DMem167 modules are used

DBoxST10

For technical specifications see DBox167

DBOX-ST10-1: 256K trace
DBOX-ST10-2: 256K trace, 16MB Code Coverage, Performance Analysis
DBOX-ST10-3: 512K trace, 16MB Code Coverage, Performance Analysis
**Tanto TriCore**

- Supports all TriCore derivatives, such as TC1775, TC1130 etc.
- Full OCDS Level 1 support for TriCore and PCP
- Support of maximum frequency
- On-the-fly access to any memory location or internal register in real-time
- 2 Code triggers or 1 code range trigger
- 2 Data triggers or 1 data range trigger
- Virtually unlimited software breakpoints
- Task-specific debugging
- External signals: trigger in/trigger out
- Flash programming
- Multi-core debugging

TANTO-TC: Tanto Base unit, HiTOP, Tanto Port Link target connection

---

**TantoPT TriCore**

- Trace depth of 1M frames
- TriCore and PCP trace for detailed analysis of CPU and peripheral activities, intelligent trace filter for optimal trace utilization, qualified trace recording based on TantoPT’s triggers, recording of external signals, timestamps for performance measurement
- 4 Triggers with 16 bit counters on OCDS2 event messages, 32Bit indirect branch address, external signals
- TriCore and PCP Triggers allow qualified trace recording of specific modules, detailed analysis of program flow in real-time
- 4 Sequence levels
- 8 External lines, configurable as 4 inputs + 4 outputs or 8 inputs, pulse or static with definable polarity, EMUL and BRKIN output

TANTOPT-TC: Tanto PortTrace Module, HiTOP extension for trace capabilities

---

**Chip Manufacturers**
- Infineon

**Supported Compilers**
- Diab Data
- Greenhills
- HighTec
- Tasking

**Host Interface:**
- Ethernet, USB

---

**Host Interface:**
- Ethernet, USB
**8051 Architecture**

**8051, C500, DS320**

### Chip Manufacturers
- Atmel
- Dallas
- Infineon
- Intel
- Philips
- and many others ...

### Supported Compilers
- Keil
- Tasking
- IAR
- and many others ...

**Host Interface:**
- RS232, PCI, USB

---

**MX51**
Mid-Range 8051 Emulator
- ROM/ROMless emulation to 42MHz
- 2.7 - 5.5V
- Dual-ported RAM and trace
- Opcode decoder for on-chip RAM triggering
- Break-before-execute hardware breakpoints
- Real-time internal access
- Up to 16 code banks (MX51-B/BH only)
- Execution profiler
- Support of entire 8051 family via adaptors
- Unique ICE/connect® adaptation

- **MX51SE:** 16MHz, 256KB RAM, no trace
- **MX51-A:** 16MHz, 128KB RAM, 2K trace
- **MX51-AH:** 24MHz, 128KB RAM, 2K trace
- **MX51-B:** 30MHz, 256KB RAM, 8K trace, code banking
- **MX51-BH:** 42MHz, 256KB RAM, 8K trace, code banking

---

**AX51**
High Performance 8051 Emulator
- ROM/ROMless emulation to 42MHz
- 2.7 - 5.5V
- Dual-ported RAM and trace
- Opcode decoder for on-chip RAM triggering
- Break-before-execute hardware breakpoints
- Real-time internal access
- Up to 16 code banks
- Execution profiler
- Absolute performance analysis
- Absolute coverage analysis
- Support of entire 8051 family via adaptors
- Unique ICE/connect® adaptation

- **AX51:** 30MHz, 256KB RAM, 32K trace, code banking
- **AX51-H:** 42MHz, 256KB RAM, 32K trace, code banking

---

Chip Manufacturers:
- Atmel
- Dallas
- Infineon
- Intel
- Philips
- and many others ...

Supported Compilers:
- Keil
- Tasking
- IAR
- and many others ...

Host Interface:
- RS232, PCI, USB

Chip Manufacturers:
- Atmel
- Dallas
- Infineon
- Intel
- Philips
- and many others ...

Supported Compilers:
- Keil
- Tasking
- IAR
- and many others ...

Host Interface:
- RS232, PCI, USB
DProbeHS

- For high-speed derivatives from Atmel up to 70 MHz
- and applications with demand for huge overlay memory size
- Up to 1 MB dual ported overlay memory,
- Breakpoints on code and data
- Optional DTrace16 oder DBox16

DPROBEHS: 256KB RAM, 70MHz

DTrace16

- Optional plug-on 32K trace buffer

DProbeHS/DBox16

- 4 complex triggers with sequence
- Unique timer trigger
- Enhanced trace unit with filter feature and time stamps
- Read-before-write detection
- Execution profiler or performance analysis
- Coverage monitor
- External logic inputs

DBOX16: Advanced function module

VCTP Architecture

TantinoVCTP

VCTP

- JTAG interface for OCDS
- Fast break for access on-the-fly
- On-chip Flash support
- OCDS hardware breakpoint
- Software breakpoints, virtually unlimited
- Task specific debugging

TANTINOVCTP: OCDS unit, HiTOP

Chip Manufacturers

Micronas

Supported Compilers

IAR, Keil, Tasking

Host Interface:

USB
Freescale / ST 8-Bit Architecture
68HC11, 68HCo8, 68HC05, ST7

Chip Manufacturers
Freescale
STMicroelectronics

Supported Compilers
Cosmic
Metrowerks

AX68n, AX-ST7
- Internal visibility of processor registers
- Break-before-execute hardware breakpoints
- Code banking
- 8MHz E-clock
- Dual-ported RAM and trace
- Performance analysis
- Coverage analysis

AX68n-A, AX-ST7-A: 8MHz, 64KB, 2 trigger, 8K trace
AX68n-B, AX-ST7-B: 8MHz, 256KB, 4 triggers, 32K trace, banking
AX68n-C, AX-ST7-C: 8MHz, 256KB, 4 triggers, 64K trace performance & coverage

MX6808
- High performance at low cost
- Break-before execute hardware breakpoints
- Rapid support for new variants
- Dual ported RAM and trace
- Built in EEPROM programming

MX6808-L: 8MHz, 64KB, 1 trigger, 2K trace, EEprogrammer
MX6808: 8MHz, 64KB, 4 triggers, 8K trace, EEprogrammer

MX-ST7
- First third party emulator for ST7
- Break-before execute hardware breakpoints
- Uses ST bondout chips
- Dual ported RAM and trace

MX-ST7-L: 8MHz, 64KB, 1 trigger, 2K trace
MX-ST7: 8MHz (16MHz), 64KB, 4 triggers, 8K trace

Host Interface:
RS232, PCI, USB
**Freescale 16-Bit Architecture**

**68HC12/MC9S12**

**DProbeHC12/DProbeS12**
- Modular emulator with low entry cost
- Built in BDM connection and flash programming
- Breakpoints on code and data
- Break-before-execute hardware breakpoints
- Very small footprint
- Plug 'n Play
- Supports all CPU modes
- EE & FLASH programming
- OSEK task trace
- QuadConnect, Yamaichi or replacement adapter

DPROBEHC12: 16MHz, 1 MB, BDM and flash programming
DPROBESES12: 25MHz, 512KB, BDM and flash programming

**DTrace16**
- Optional plug-on 32K trace buffer

DTRACE16: 32K trace frames module

**DProbeHC12/DBox16 and DProbeS12/DBox16**
- 4 complex triggers with sequence
- Unique timer trigger
- Enhanced trace unit with filter feature and time stamps
- Read-before-write detection
- Performance analysis
- Coverage monitor
- External logic inputs

DBOX16-1, DBOX16-2, DBOX16-3: Advanced functions module

**TantinoS12**
- Hardware and software breakpoints
- 2.7 - 5.5v support
- Bank-aware breakpoints
- “Hot pluggable”
- Integrated FLASH and EEPROM programming

TANTINOS12: BDM debugger with FLASH/EEPROM programmer

**EVA-S12**
- Evaluation board for MC9S12 controller

EVA-S12: Evaluation board for MC9S12 controller
MSP430 Architecture

MSP430

Chip Manufacturers
Texas Instruments

Supported Compilers
IAR
HighTec

DProbe430

- Break-before-execute breakpoints
- Data breakpoints
- 64KB emulation memory
- 64K trace

DPROBE430-BAS: 8MHz, 64KB RAM, 64K trace, memory protection
DPROBE430-DP110: module for MSP430x11x, MSP430x11x1 and MSP430x31x
DPROBE430-DP300: module for MSP430x31x, MSP430x32x and MSP430x33x
DPROBE430-DP140: module for MSP430x13x and MSP430x14x

DProbe430/DBox16

Advanced functions module for DProbe430

- 4 complex triggers with sequence
- Unique timer trigger
- Enhanced trace unit with filter feature and time stamps
- Read-before-write detection
- Performance Analysis
- Coverage monitor
- External signal inputs

DBox16: Advanced functions module
DBox16-IF430: necessary interface to connect DBox16 with DProbe430

Host Interface:
RS232, PCI, USB
CR16C Architecture

Tantino CR16C

- Small and compact Nexus class 1 debug hardware
- Support the maximum operating frequency without wait-states
- Fast program download
- FLASH programming support
- Optional 'smart' access to variables with minimum real-time violation
- Virtually unlimited number of software breakpoints for code located in the target RAM
- 8 complex hardware breakpoints for code, data read or data write for the complete 16 MB address range
- 4 break ranges for code, data read or data write for the complete 16 MB address range
- External signals: trigger in and trigger out
- Power supply: USB bus powered

TANTINO-CR16C
TANTINO-CR16C-DECT

Chip Manufacturers
National Semiconductor

Supported Compilers
IAR
National Semiconductor

Host Interface: USB
Connecting A Hitex Emulator...
... to your Target PCB

CPU is mounted in a socket.
1. Simple adaptors that replace CPU in socket.
   - Dual In-Line socket adaptor
   - PGA socket adaptor
   - PLCC socket adaptor
   - SMD socket adaptor

CPU is soldered down, even on prototypes...
1. **QuadConnect®**
   SMD pads arranged around CPU that accept a low-cost adaptor.

2. **PressOn®**
   Attaches to CPU which is then tristated.
   Use where space does not permit QuadConnect®.
   Can be fitted to any PCB.

3. **ICE/connect® (ROMless 8051 family only)**
   Special connector on PCB accepts a low-cost emulator probe.
   One adaptation for most derivatives.

4. **JTAG Connection**
   Standard connection for all JTAG based targets.

5. **Mictor® Trace Connection**
   Standard connector for ARM trace port.

6. Where CPU is not fitted (prototypes) ...
   (Pack layout for CPU)
   - Replacement Adaptor
   Solder-in adaptor with same footprint as CPU.

Host Computer Interfaces For Hitex Emulators
You can use a variety of connection methods to your PC or workstation:

<table>
<thead>
<tr>
<th>MX/AX and DProbeHC12, -251, -HS Systems:</th>
<th>RS232, PCI, USB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAPCI: High speed PCI parallel interface</td>
<td></td>
</tr>
<tr>
<td>USB2PARA: USB to parallel interface converter</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DProbe167, DProbeST10, DProbeXC Systems:</th>
<th>PCI, PC-CARD, LPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPHIF-2: PCMCIA II interface</td>
<td></td>
</tr>
<tr>
<td>DPHIF-3: PCI interface</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tanto Systems:</th>
<th>Ethernet, USB</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Tantino Systems:</th>
<th>USB</th>
</tr>
</thead>
</table>
USB Protocol Analyzer
Explores the Secrets of USB

USBAgent

USBAgent — the efficient USB bus & protocol analyzer — will become your preferred development tool for any Universal Serial Bus design. The USBAgent records non-intrusively all USB traffic either at low or full speed. The easy-to-use Windows® driven user interface with all its helpful analytical features helps you to reveal quickly any potential problem in your USB system.

- Records and analyses USB traffic and USB protocol
- Operates with both portable and desktop PCs via LPT port
- Provides recording of real-time USB traffic in full-speed and/or low-speed branches
- Allows powerful triggering on user-definable conditions or external input
- Connects non-intrusively to any branch of a USB system through a high-impedance probe
- Comes with 8Mbit capture buffer (which stores both data lines)
- Includes a one-year warranty and hotline customer support
- The easy-to-use Windows® driven software monitors all bus traffic and static states, and also signals abnormal bus conditions
- Simply connect the USBAgent into any branch of your system

USB AGENT: USB Analyzer with Software

USBIO USB Driver Development Software

Universal USB Plug 'n Play Driver Kit For Windows 98 Windows 2000 and Windows XP.
USBIO driver provides a standard Windows API-like interface that allows the USB interface to be controlled through standard C functions like ReadFile, WriteFile or DeviceControl.

- Supports USB 1.1 and USB 2.0
- Fast and easy development of USB PC applications
- Supports all USB functionality including Plug 'n Play and Power Management
- Complies with the WIN32 Driver Model (WDM)
- Supports all USB transfer types, provides an interface to USB pipes that is similar to files
- Supports multi-configuration and multi-interface devices
- Allows multiple applications and multiple devices at the same time
- No run-time royalties

USBIO: Universal USB driver
IEEE1394 (FireWire) Protocol Analyzer

**FireSpy400, FireSpy800**

If you’re developing hardware or software based on IEEE1394, then FireSpy is your best friend. This development and debugging tool in the form of a compact highly integrated system will connect to a PC host or a MAC with USB interface. A power-packed system with 128 Mbyte buffer, multiple trigger and filter features that will not burn a hole in your pocket. With FireSpy400 data transfer rates of up to 400 Mbps are supported (even in the generator function) and the system is IEEE1394a compliant. The FireSpy800 is IEEE1394b compliant with 800 Mbits/sec. The user interface is comfortable and easy to work with and won’t let you down. All information on the IEEE1394 bus and its traffic can be displayed in various degrees of detail, from an overview against a time scale to transaction or packet view.

FireSpy consists of four systems in one:

**Monitor**

The Monitor provides an overview of bus activity. Packets shown are differentiated with respect to their speed and type. Concerning errors, the amount of packets associated with each error type is displayed. Bus voltage is also monitored.

**Recorder**

This is the most crucial tool to guarantee performance on an IEEE1394 device. The recorder samples IEEE1394 bus traffic and stores the data in the internal buffer. The recording process can be controlled by powerful trigger and filter options. Various viewing options provide insight into the recorded data stream.

**Commander**

The commander initially reveals the IEEE1394 system’s topology. In Phy Register View or Memory View, it’s possible to examine and modify the contents of registers and memory locations of all IEEE1394 devices connected to the system. The commander also allows all types of individual packets to be sent or received.

**Generator**

The Generator enables large amounts of isochronous or asynchronous packets to be sent. Manual editing of these packets is not always necessary since it’s possible to resend pre-recorded data streams, even if they’ve been altered. The isochronous generator has a convenient graphical editor while the asynchronous generator has the intelligence to wait for specified events to occur before running.

FIRESPY SPB2: Protocol support for SPB2
FIRESPY AVC: Protocol support for AV/C
FIRESPY IP4: Protocol support for IP4
FIRESPY IIDC: Protocol support for IIDC
Tessy

Tessy performs automated unit testing of embedded software as well as facilitating all other aspects of software testing. This tool can save an embedded development project tremendous amounts of time, particularly if regression testing is to be performed. Regression testing itself is a key feature to achieve software quality. Tessy may be used to perform module tests according to IEC 61508.

Tessy analyses the interface of a function under test. Then Tessy lets the user specify the test cases. Each test case consists of the input values for the function under test and the expected results. The data is automatically saved in a database. Tessy generates the test driver for the function under test and – if necessary – provides stub functions, in case the function under test calls subroutines, which are not implemented yet. Everything is compiled and linked using a suitable compiler for the embedded system. Using the link into HiTOP, the resulting executable is exercised on the debug system, normally an in-circuit emulator.

Tessy executes all test cases and checks, if the expected results are obtained or not and generates a test report. If a test case failed, Tessy can cause HiTOP to set a breakpoint at the function entry and re-run the test. So, the user ends up in HiTOP at the function entry after the function was called with a set of parameters the user knows to generate unexpected results. The powerful features of HiTOP and the debug system can now be used to find the problem.

Regression testing reveals if new errors were introduced during further development of the application. Because Tessy can re-use the test data even if the interface of the function under test changes, and because Tessy can operate in batch mode, Tessy is extremely well-suited for regression testing.

Tessy generates test documentation in various formats (Word, HTML, Excel, ...)

Tessy at a glance
- Automatic test execution
- Test data management
- Seamless integration with HiTOP
- Powerful regression testing
- Unlimited number of test cases
- Code coverage
- Export / import of test data to/from Excel
- Batch mode
- ASAP2 support

Tessy includes the Classification Tree Editor (CTE), a graphical tool to specify test cases according to the Classification Tree Method.

TESSY: Tessy including CTE

More information: www.hitex.de/perm/tessy.htm
Development Assistant for C

The Development Assistant for C (DAC) from RistanCASE is used to develop, review, document and maintain C projects. DAC allows you to browse through your project files and edit them. DAC displays your C program in structured form or in flow chart form. DAC features include Software Metrics, which allow you to rate the quality of your software. The Static Code Analysis feature points out possible errors that a compiler may not be able to find. DAC can check your sources for compliance with the MISRA rules.

DAC lets you easily navigate in your code to find definitions and declarations of variables, caller/callee-pairs etc.. DAC can automatically generate documentation for your project. It supports more than 90 dialects of C compilers for embedded applications.

DAC-NL: Node locked version of DAC
DAC-FL: Floating license version of DAC

More information: www.hitex.de/perm/dac.htm

RiskCAT and IEC 61508

RiskCAT supports the application of the IEC 61508 standard. This standard covers a wide range of measures that can be applied in the development of safety-critical systems. The degree of obligation of each measure varies with the Safety Critical Level (SIL) the system is supposed to achieve. The SIL to be achieved depends on the risk the system bears, i.e. the probability and the consequences of a system failure.

RiskCAT is also available for DIN EN 50129 (only in German) and DIN EN 50128 (in English and German), incorporates parts of DIN EN 50126:

- Assists you in determining the risk of the system according to IEC 61508
- Automatically derives the Safety Integrity Level (SIL) of the system according to IEC 61508
- Indicates the degree of obligation (recommended, mandatory, ...) for each measure
- Enables the user to select the measures that shall be applied
- Documents all decisions for review
- Generates checklists for the selected measures

RiskCAT is also available for DIN EN 50126 and DIN EN 50128 (RiskCAT Railway, only in German)

Store/Reload enables you to start work with user-defined preselections of measures (e.g. company-wide guidelines).

More information about RiskCAT: www.hitex.de/perm/riskcat.html including a white paper and evaluation software.
Embedded C/C++ Compilers

Keil

Compiler:  
Debugger:  

The µVision IDE from Keil Software combines project management, make facilities, source code editing, program debugging, and complete simulation in one powerful environment. µVision helps you get programs working faster than ever while providing an easy-to-use development platform. The editor and debugger are integrated into a single application and provide a seamless embedded project development environment. Keil Software development tools support every level of developer from the professional applications engineer to the student just learning about embedded software development.

The Keil development tools for ARM support the most popular ARM7 derivatives and help you get your projects completed on schedule. The industry-standard Keil C Compilers, Macro Assemblers, Debuggers, Real-time Kernels, and Single-board Computers support all 8051, 251 and C16x derivatives.

Tasking

Compiler:  
Debugger:  

TASKING products from Altium Limited are the choice for embedded software development across industry standard computer platforms. Our embedded development environment, compiler, assembler and debugger, embedded Internet and RTOS offerings, coupled with our global support and services, are the most comprehensive available. Worldwide, more embedded software applications have been developed using TASKING tools than anyone else’s.

Integrates with DA-C, PC-Lint.
IAR

Compiler: SCL4

Debugger: SCL1

The IAR Embedded Workbench IDE with its C/C++ compiler and C-SPY debugger is the most complete and easy-to-use set of development tools available today for professional embedded applications. IAR Embedded Workbench offers the same intuitive user interface - a truly Integrated Development Environment.

The IAR Embedded Workbench Professional supports you through all the different phases of embedded application development. It gives you the benefits of the integrated development environment with its C/Embedded C++ compiler, assembler, linker, librarian, editor, project manager and debugger plus a graphical system design tool, a test tool and an automatic system documentation tool, all in one package!

All above listed IAR compilers are compatible with Hitex development tools.

Cosmic

Compiler: SCL4

Debugger: SCL1

Cosmic are specialists in compilers for 8-bit and 16-bit Freescale and derivative CPUs such as the ST7. By using custom code generators and optimisers for each family, code efficiency is high and the subtleties of the instruction set catered for. The IDEA workbench coordinates the edit, compilation, linking and debugging phases and contains the power ZAP debugger. All Cosmic compilers are compatible with Hitex emulators.

Host platform: Windows and Unix

Integrates with DA-C, PC-Lint.
Training and Consulting

HiSCHOOL Hitex Training

HiSCHOOL has been offering comprehensive seminars for more than 6 years now. This includes extensive hands-on training on our products as well as important topics around the Embedded World. Our Trainers will be pleased to help you become familiar with your new emulation system. We provide workshops for ARM, 8051, XC166 and TriCore architectures together with our partners and chip manufacturers. We also provide courses for popular bus technologies such as USB, IEEE1394, CAN and TTP. Also offered are courses for Tessy, the unit and regression test tool.

The current training dates can be found in the event calendar. Most of our Seminars can be provided also in english. If you are interested in a training in one of the topics above, please contact our training department training@hitex.de. Following our company’s locations around the world, the courses are held at different places. It goes without saying that we also provide on-site seminars upon request.

First Class Consultancy

As one of the leading emulator manufacturers Hitex has years of considerable experience in the embedded market. Therefore Hitex also offers professional consultancy for your new projects. The Hitex consultants will help you to find the optimal solution for the special requirements of your project. Take advantage of this unique service and contact us for further details.