

See releasenotes.txt in the Dynamic C root directory for more details.

DYNAMIC C 7.33P3 PREMIER RELEASE HIGHLIGHTS

FUNCTIONAL CHANGES

Added support for BL2500 with RCM3200 core module option.

Added support for EG2110 Rabbit Link.

DYNAMIC C 7.33 PREMIER RELEASE HIGHLIGHTS

NEW FEATURES

Samples, libraries, and documentation for new RCM3400 core module added.

Samples, libraries, and documentation for new BL2500 added.

DYNAMIC C 7.32 PREMIER RELEASE HIGHLIGHTS

NEW FEATURES

New Download Manager Sample Programs -- One for single 256K byte flash boards added, and one using FTP instead of RS232.

Separate Instruction and Data space - The compiler can now take advantage of an MMU feature available on Rabbit 2000A and later chips that effectively doubles the root memory space available.

Simple Network Management Protocol (SNMP) -- Implements SNMP v1 basic protocol (using UDP as transport mechanism) plus a database for the MIB tree. (Dynamic C Premier only)

TCP/IP Multicasting Support -- For IGMPv1 and IGMPv2. This allows Internet enabled Rabbit products to send packets with multiple destinations.

Ethernet packet driver support added for SMSC LAN91C113 10/100 and ASIX AX88796 10/100.

TCP/IP Multiple Interface Support -- Allows

simultaneous use of multiple Ethernet ports
or Ethernet and PPP.

Samples, libraries, and documentation for new
RCM3200 core module added.

Samples, libraries, and documentation for new
OP7200 controller with QVGA added.

Extended memory data handling functions added.

OTHER IMPROVEMENTS

General TCP/IP Improvements -- Includes:

- * Congestion avoidance improvements
- * Significantly faster transmission.
- * ARP now handles ICMP redirect messages
- * ...and more

Significant speed-ups to compiling and
downloading

DYNAMIC C 7.26 RELEASE HIGHLIGHTS

Samples, libraries, and documentation for the new
LP3500 series and BL2000, BL2100, OP6800 and SmartStar
series controllers added.

Improvements to target communication robustness.

DYNAMIC C 7.25 RELEASE HIGHLIGHTS

Samples, libraries, and documentation for new RCM3000 core
module added.

STDIO output can now be directed to a serial port for
debugging under run mode.

The baud rate used for downloading is now negotiated between
the PC and the target. This may speed up downloading,
particularly if a USB converter is used where rates up to
460800 baud are possible.

The baud rates for debugging with the BIOS have been expanded
from 57600 and 115200 to 9600, 19200, 38400, 57600, and
115200. This aids in debugging with user designed boards with
slower crystals.

Dynamic C now works with a wider selection of USB to RS232
converters. The quality of drivers for converters varies
widely, and Dynamic C attempts to compensate for these
differences.

Support was added for more large-sector flash products.

Alt+F7 and Alt+F8 can now force the debugger to step by c statements rather than by instruction when the assembly window is open. (Introduced in 7.10).

DYNAMIC C 7.21 RELEASE HIGHLIGHTS

Root memory usage reduction in BIOS and libraries.

Tech Note 219 - Root memory usage reduction tips added.

uC/OS-II release notes from Jean Labrosse added

DYNAMIC C 7.20 RELEASE HIGHLIGHTS

NEW FEATURES

Project Files - Project files facilitate working on multiple Dynamic C projects. Project level information such as compiler options, libraries used, files open, and project scope #define macros can now be saved in project files. Use the project command on the file menu.

Support for Enums - Dynamic C now supports the standard intrinsic enum type.

Precompilation - Functions can be listed in the file precompile.lib to save recompiling them every time.

µC/OS-II Upgrade - The Rabbit/Dynamic C port of Jean Labrosse's popular real-time operating system for embedded systems, µC/OS-II, has been upgraded to version 2.51. New features include mutex semaphores and event flags. µC/OS-II for Rabbit is available only in Dynamic Premier, and may be used license and royalty free in end products.

TCP/IP Improvements - Many improvements to the robustness and performance of the TCP/IP stack have been made. (Most of these were introduced in version 7.06P2) See releasenotes.txt in the Dynamic C root directory for details.

Command Line RFU Version - A GUI-less Rabbit Field Utility for loading programs without Dynamic C is now available.

Clear All Breakpoints Command - A command on the Run menu to remove all breakpoints at once has been added.

New System Macros - __LINE__, __FILE__, __DATE__, __TIME__ compiler defined macros have been added.

Library encryption - A utility program that is not released with Dynamic C, but is available from Z-World Tech Support will encrypt libraries in a format that Dynamic C can decrypt. This is primarily of interest to people who want to make arrangements with Z-World to resell Dynamic C as a programming tool for a programmable end product.

OTHER FUNCTIONAL IMPROVEMENTS

Faster Debugging - Single stepping through code and printf are now noticeably faster.

Large Sector Flash Support - Dynamic C now Supports a larger variety of flash types, including several very-large , non-uniform sector size types. This change required some fundamental changes in the way the debugger interacts with the target. Single-stepping and breakpoint setting used to cause flash writes to occur. This no longer happens.

Faster, More Flexible Cloning - Cloning is a feature that allows one Rabbit target to copy itself to another using a special cloning board available from Z-World. (The simple circuit is easily designed into board test fixtures.) Previously, a "fast" cloning option was available but had some restrictions. Now, all cloning is fast (up to 921.6 kbaud), and options have been added for: creating "sterile" clones, copying the User and System ID blocks, running after copying, copying a second flash, and more. See the cloning configuration options near the top of rabbitbios.c or the Rabbit 2000 Designer's Handbook.

Constant String Optimization - If an identical literal string is defined in code multiple times, only one instance of it will be generated and used.

OTHER IMPORTANT FUNCTIONAL CHANGES

Assembly level debugging - Previously, #asm blocks embedded in C functions took on the "debug/nodebug" characteristic of that C function. The #asm directive to begin an assembly block now can take a "debug" or "nodebug" keyword, and the default is nodebug, regardless of the how the C function is declared. This means that a breakpoint can not be set inside an #asm block unless "#asm debug" is explicitly used. Due to the changes to support large sector flash, using debug on the assembly block now causes RST 28Hs to be placed in between instructions so that breakpoints may be set. This may have the side effect of making short jump instructions out of range (an assembler error message will be generated). This can be fixed by changing jr to jp, and will not break pre-existing code since the debug keyword was not usable with #asm before. Nodebug Assembly blocks may still be single-stepped through in the disassembly window.

Targetless Compilation - Compiling with "Use Attached Target" no

longer compiles the bios, and requires that the target have a running bios/program. The Options menu now has a "Define Target Configuration Command." The "Compile to *.bin file" command now uses the target configuration defined in the new option to compile, instead of popping the Target Configuration Dialog up for every compilation.

RabbitLink Changes - The RabbitLink has had a major overhaul in Dynamic C 7.20 to bring it in sync with the Ethernet Loading utility of the DeviceMate, to allow for future compatibility, and to allow pure-Ethernet loading, without ever needing a serial port, through the use of DHCP. Because of this overhaul, it will be necessary to upgrade the firmware on the RabbitLink hardware for it to be able to communicate with Dynamic C 7.20 or later. The New RabbitLink firmware has the flexibility to allow for future changes in communication method between the Dynamic C and the target and avoid future incompatibilities.