StellarisWare Release Notes

Literature Number: SW-RLN-5450
December 02, 2009

Copyright

Texas Instruments
108 Wild Basin, Suite 350
Austin, TX 78746
Main: +1-512-279-8800
Fax: +1-512-279-8879
http://www.luminarymicro.com

Revision Information

This is version 5450 of this document, last updated on December 02, 2009.

Table of Contents

1 Release Notes for StellarisWare Revision 5450 (December 2, 2009)

1.1 New Features in Stellaris Boot Loader Library

1.1.1 Added Internal Pull up/down to boot loader. (Reference 10876)

The boot loader has the ability to check an external pin to see if it is set to a high or low voltage level and remain in the boot loader. However, the boot loader did not have a way to enable an internal pull-up or pull-down when using this feature. Two additional configuration options were added to enable an internal pull-up resistor by defining FORCED_UPDATE_WPU or an internal pull-down by defining FORCED_UPDATE_WPD value. These values are mutually exclusive and only one or the other should be used when using an internal pull resistor. If an external resistor pull-up or pull-down is used then neither valued should be defined.

1.2 New Features in Stellaris Peripheral Driver Library

1.2.1 Added IntPendSet() and IntPendClear() APIs (Reference 10694)

The IntPendSet() and IntPendClear() APIs have been added to the interrupt controller driver. They allow an interrupt to be pended (made to appear as if it has occurred) or unpended (made to appear as if it has not occurred). Unpending an interrupt is sometimes useful prior to enabling it (if the process of configuring the interrupt source causes a false trigger prior to the interrupt being enabled).

1.3 Bug Fixes in Stellaris Peripheral Driver Library

1.3.1 I2CMasterErr returned I2C_MASTER_ERR_NONE if arbitration was lost (Reference 10821)

Function I2CMasterErr previously assumed that bit 2 of the I2CMCS register would be set in all error conditions and, if this bit was clear, assumed no error had occurred. Unfortunately, this bit only indicates an ACK error so the function would return I2C_MASTER_ERR_NONE if the controller lost arbitration. This has been fixed.

1.3.2 USBFIFOFlush() fails to flush endpoints. (Reference 10853)

The USBFIFOFlush() function was improperly checking the state of the FIFO and was not allowing endpoints to flush the endpoint's FIFO. This affected all endpoints other than endpoint zero.

1.3.3 Use of non-MOTO formats in SSIConfigSetExpClk() can cause minor errors in the actual clock rate (Reference 10922)

If a non-MOTO format was specified in a call to the SSIConfigSetExpClk() function, two lower bits of a clock divisor register could be corrupted. The result was a small error in the actual clock rate. This has been fixed.

1.3.4 Added support for PWM3 trigger to ADC (Reference 10943)

Add the ability to configure an ADC sample sequence to be triggered by the event generated by the PWM3 generator, when present. The ADC_TRIGGER_PWM3 is now supported by ADCSequenceConfigure().

1.3.5 Corrected definition of NUM_INTERRUPTS (Reference 10908)

The definition of NUM_INTERRUPTS was off by one, resulting in improper treatment of the last interrupt (GPIOJ) by the IntRegister()/IntUnregister() functions, and assertion failures by the remaining Interrupt driver APIs.

1.4 Bug Fixes in Stellaris Utility Library

1.4.1 Fixed trailing slash handling errors in fswrapper (Reference 10892)

Three related changes have been made to fix problems experienced when using fswrapper and the "cd" command on the qs-checkout application serial command line. Previously fswrapper would result in a NULL pointerin ppcFSFilename if the pcName string passed did not contain a second

slash character. It now returns a pointer to the terminating NULL instead, preventing a corrupt string being displayed by the application. Additionally, file.c in qs-checkout has been modified to ensure that it never leaves a trailing slash at the end of the current working directory string and will also detect and fail attempts to change into non-FAT directories since the internal file system image access functions do not support directory navigation and listing.

1.5 New Features in DK-LM3S9B96 Firmware Package

1.5.1 qs-checkout application updated to support FPGA/Camera daughter board (Reference 10647)

Minor changes have been made to the qs-checkout example application to ensure that it operates correctly with the FPGA/Camera daughter board attached. In this case, no external RAM is available to store a directly accessible copy of the photo gallery file system image so this web site is not available and the “Image Viewer” function reports “No file system image”.

1.6 Bug Fixes in DK-LM3S9B96 Firmware Package

1.6.1 JPEG decode example rebranding (Reference 10614)

The JPEG image used in the showjpeg example application has been replaced with a version including the Texas Instruments logo rather than the Luminary Micro logo.

1.6.2 Fixed trailing slash handling errors in fswrapper (Reference 10892)

Three related changes have been made to fix problems experienced when using fswrapper and the “cd” command on the qs-checkout application serial command line. Previously fswrapper would result in a NULL pointerin ppcFSFilename if the pcName string passed did not contain a second slash character. It now returns a pointer to the terminating NULL instead, preventing a corrupt string being displayed by the application. Additionally, file.c in qs-checkout has been modified to ensure that it never leaves a trailing slash at the end of the current working directory string and will also detect and fail attempts to change into non-FAT directories since the internal file system image access functions do not support directory navigation and listing.

1.6.3 Corrected error in IAR linker script for ext_demo_1 and ext_demo_2 (Reference 10978)

An error in the definition of the size of the SRAM on the Flash/SRAM/LCD daughter board was fixed in the IAR toolchain linker scripts for example applications ext_demo_1 and ext_demo_2. The previous versions defined the SRAM to be 64KB in size when it should have been 1MB.

1.6.4 eflash tool added to dk-lm3s9b96 release (Reference 10979)

Although documentation for the “eflash” tool was included in previous StellarisWare releases for dk-lm3s9b96, the actual source and binary for the tool was omitted. These files have now been included and can be found in the “tools” subdirectory after StellarisWare installation.

1.7 Bug Fixes in EK-LM3S6965 Rev A Firmware Package

1.7.1 Fix handling of invalid page in uip web server (Reference 10981)

Fixed a cut-paste error in the setting of the page length for an invalid page. Also modified the invalid page to return very simple “page not found” content.

1.8 Bug Fixes in EK-LM3S6965 Firmware Package

1.8.1 Fix handling of invalid page in uip web server (Reference 10981)

Fixed a cut-paste error in the setting of the page length for an invalid page. Also modified the invalid page to return very simple “page not found” content.

1.9 Bug Fixes in EK-LM3S8962 Firmware Package

1.9.1 Fix handling of invalid page in uip web server (Reference 10981)

Fixed a cut-paste error in the setting of the page length for an invalid page. Also modified the invalid page to return very simple “page not found” content.

1.10 Bug Fixes in EK-LM3S9B90 Firmware Package

1.10.1 Fix handling of invalid page in uip web server (Reference 10981)

Fixed a cut-paste error in the setting of the page length for an invalid page. Also modified the invalid page to return very simple “page not found” content.

1.11 Bug Fixes in EK-LM3S9B92 Firmware Package

1.11.1 Fix handling of invalid page in uip web server (Reference 10981)

Fixed a cut-paste error in the setting of the page length for an invalid page. Also modified the invalid page to return very simple “page not found” content.

1.12 Bug Fixes in RDK-BDC Firmware Package

1.12.1 Properly handle overflow in PID controller (Reference 8535)

An overflow of the output of the PID controller will now clip instead of wrapping. The wrapping behavior could cause unexpected behavior, like the motor being driven in the wrong direction. With clipping, the motor behaves as expected.

1.13 New Features in RDK-IDM Firmware Package

1.13.1 Display and touchscreen driver updated to support the ILI9328 controller (Reference 10894)

The display and touchscreen drivers for rdk-idm have been updated to support the new KWH028Q02-F02 display module. This features an ILI9328 display controller which is compatible with the ILI9325 found on the previous module, KWH028Q02-F05. Minor updates in these drivers ensure that all three display controller IDs are handled correctly allowing a single binary to run on IDMs equipped with any of the displays.

1.14 Bug Fixes in RDK-IDM-SBC Firmware Package

1.14.1 qs-blox web site now updates correctly using IE7 (Reference 10613)

Special-case JavaScript has been added to the web site served by the qs-blox example application to ensure that the dynamically updated fields are correctly displayed in Internet Explorer 7. This browser insists on interpreting the XML response from the board as plain text so the JavaScript which receives the request now constructs an XML document around the response text and parses out the individual fields from it.

1.14.2 JPEG decode example rebranding (Reference 10614)

The JPEG image used in the showjpeg example application has been replaced with a version including the Texas Instruments logo rather than the Luminary Micro logo.

1.14.3 QS-Blox web site now updates in IE (Reference 10738)

A workaround for an Internet Explorer problem has been added to the web site served by the qs-blox example application. This allows the page to update automatically without a manual reload being required. The previous version of the web site operated correctly on standards-compliant browsers.

1.14.4 Fixed trailing slash handling errors in fswrapper (Reference 10892)

Three related changes have been made to fix problems experienced when using fswrapper and the “cd” command on the qs-checkout application serial command line. Previously fswrapper would result in a NULL pointerin ppcFSFilename if the pcName string passed did not contain a second slash character. It now returns a pointer to the terminating NULL instead, preventing a corrupt string being displayed by the application. Additionally, file.c in qs-checkout has been modified to ensure that it never leaves a trailing slash at the end of the current working directory string and will also detect and fail attempts to change into non-FAT directories since the internal file system image access functions do not support directory navigation and listing.

1.15 Bug Fixes in RDK-S2E Firmware Package

1.15.1 Fix storage of baud rate in RFC2217 code (Reference 10839)

In RFC2217, the baud rate parameter is a four octet parameter, which arrives in network (big-endian) order. The code that saves this parameter data as it arrives has been corrected to properly store it in host (little-endian) order.

2 Release Notes for StellarisWare Revision 5228 (October 1, 2009)

2.1 Bug Fixes in Stellaris Boot Loader Library

2.1.1 Ensure vector table is not compressed in IAR boot loader builds (Reference 10345)

The file bl_link.icf was updated to ensure that the IAR tools never compress the relocated boot loader vector table and initialized data sections. Using version 5.3 of Embedded Workbench for ARM, compression of these sections was noted in some cases during development of a boot loader supporting execution from external flash and, since relocation is not handled using IAR's code, this caused the boot loader to crash when run.

2.1.2 Use read-modify-write when configuring pins and peripherals in the boot loader (Reference 10348)

The boot loader now uses read-modify-write when enabling or disabling peripherals and configuring pins. This ensures that any peripheral or pin that has been configured during an application-specific hook function will not be unintentionally disabled or reconfigured by the boot loader code.

2.2 New Features in Stellaris Peripheral Driver Library

2.2.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.2.2 Added SSIBusy() function (Reference 9606)

This function determines if the SSI peripheral is busy transmitting data.

2.2.3 EPIConfigNoModeSet renamed to EPIConfigGPModeSet (Reference 10247)

The DriverLib API function EPIConfigNoModeSet has been renamed EPIConfigGPModeSet to ensure that the terminology used in the driver matches the part datasheets. Similarly, several labels defined in epi.h have been renamed to ensure consistency with the microcontroller documentation. The previous function and labels defined in epi.h have been deprecated but macros have been added to the header file to ensure that existing code using the function will still build and operate.

2.2.4 EPINonBlockingWriteCount renamed EPIWriteFIFOCountGet (Reference 10248)

The previous API EPINonBlockingWriteCount() has been renamed EPIWriteFIFOCountGet() to clarify operation. All EPI writes are via the write FIFO. If space is available, they do not block, otherwise they block until they can be added. The previous API has been deprecated but is mapped to the new function via a macro in epi.h to ensure backwards compatibility.

2.2.5 EPI driver function additions (Reference 10064)

Several changes have been made to the EPI driver. New configuration parameter flags have been added to EPIConfigHB8Set() and EPIConfigSDRAMSet() to allow configuration of features found in the EPIHB8CFG2 and EPISDRAMCFG2 registers respectively.

2.2.6 CANMessageSet() now provides flag to enable FIFO mode. (Reference 10431)

The CANMessageSet() did not allow configuring a set of message objects as a FIFO for transmitting or receiving CAN messages. This change adds the MSG_OBJ_FIFO flag value to tag message objects as part of a FIFO and not the final entry in a FIFO. This allows multiple message objects to be linked together to transfer or receive more than 8 bytes at a time.

2.3 Bug Fixes in Stellaris Peripheral Driver Library

2.3.1 Remove uDMAIntStatus() and uDMAIntClear() APIs (Reference 10148)

The APIs that use the DMA_CHIS register have been deleted because this register is no longer available for use and has been removed from the data sheet.

2.3.2 I2C_MASTER_CMD_BURST_RECEIVE_ERROR_STOP definition corrected (Reference 10434)

The value of I2C_MASTER_CMD_BURST_RECEIVE_ERROR_STOP was incorrect; the correct value is now provided.

2.3.3 CanBitRateSet() was incorrectly checking requested bit rate inputs. (Reference 10439)

The function CanBitRateSet() function was incorrectly checking the limits for possible bit rates given an input clock rate. The correct checking is now in place as ASSERT() checks to allow the parameter checking code to be removed in non-DEBUG builds.

2.3.4 Error in epi.h address size definitions corrected (Reference 10461)

An error in epi.h has been corrected and affected example applications have been updated. Labels EPI_ADDR_PER_SIZE_512MB and EPI_ADDR_RAM_SIZE_512MB have been replaced with the correct definitions for those register bit patterns, EPI_ADDR_PER_SIZE_256MB and EPI_ADDR_RAM_SIZE_256MB.

2.4 New Features in Stellaris Host Tools Library

2.4.1 pnmtoc now supports grayscale “PGM” files (Reference 10402)

The pnmtoc tool used to convert PBM/PNM format images into C arrays for use with the Stellaris Graphics Library has been updated to support conversion of grayscale “Portable Gray Map” (identifier “P5”) images. The previous version supported only color PBM images with identifier “P6”. Gray map images are created by the open source giftopnm tool when the input GIF image contains only shades of gray.

2.4.2 Windows USB example application rebranding (Reference 10500)

The Windows USB examples and their installer have been changed to present TI branding rather than Luminary Micro. This rework involved no functional changes in any application. Applications now install into “C:\Program Files\Texas Instruments\Stellaris” by default and shortcuts can be found under “Start\All Programs\Texas Instruments\Stellaris”.

2.5 Bug Fixes in Stellaris USB Library

2.5.1 USB host applications using OTG mode will hang if connected as a device. (Reference 10142)

USB application that are using OTG mode without having initialized device mode operation will hang if they are connected as a device to another USB host. This bug will affect all USB host example applications on kits that provide a USB OTG connector.

2.6 Bug Fixes in Stellaris Utility Library

2.6.1 Additional pointer checking added to fswrapper (Reference 10374)

When using a position independent file system image, some additional safety checks have been added in fswrapper and the qs-checkout application for dk-lm3s9b96 to prevent a fault exception from occurring if the file system image is corrupted. The file system code now checks that pointers calculated from the linked list in the file system image are actually within the bounds of the image itself before dereferencing them. Although a corrupt file system is itself a serious error, this change allows the qs-checkout application to boot successfully even with a bad file system image in SSI flash and allows the user to update the bad image via TFTP.

2.7 Bug Fixes in Stellaris Third Party Libraries

2.7.1 JPEG decoder reworked to use new ExtRAMAlloc/Free functions. (Reference 10262)

The JPEG decoder has been reworked to use functions ExtRAMAlloc() and ExtRAMFree() in place of the now-deprecated SDRAMAlloc() and SDRAMFree().

2.8 New Features in DK-LM3S9B96 Firmware Package

2.8.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.8.2 Higher data rate audio files can now be played in qs-checkout (Reference 10151)

Recent optimizations in the USB library have allowed the previous 64KB/S data rate restriction when playing uncompressed audio files from a USB flash stick to be lifted. Smooth audio playback is now possible from USB flash sticks for stereo files at 44.1KHz and 48KHz while running the qs-checkout example application.

2.8.3 Support added for SRAM/Flash/LCD Daughter Board (Reference 10307)

Three new examples and a new driver have been added to the StellarisWare release for dk-lm3s9b96 in support of the new SRAM/Flash/LCD daughter board. The examples are a sample boot loader (boot_eth_ext) allowing images to be written to external flash and booted from there and two small example applications that will operate with that boot loader, ext_demo_1 and ext_demo_1. A new driver, extflash.c, is included to support erasing and programming the external flash.

2.8.4 EPIConfigNoModeSet renamed to EPIConfigGPModeSet (Reference 10247)

The DriverLib API function EPIConfigNoModeSet has been renamed EPIConfigGPModeSet to ensure that the terminology used in the driver matches the part datasheets. Similarly, several labels defined in epi.h have been renamed to ensure consistency with the microcontroller documentation. The previous function and labels defined in epi.h have been deprecated but macros have been added to the header file to ensure that existing code using the function will still build and operate.

2.8.5 PinoutSet() now performs dynamic EPI configuration. (Reference 10042)

Function PinoutSet() in drivers/set_pinout.c has been reworked to offer dynamic configuration of the Extended Peripheral Interface (EPI) based upon information read from an I2C-connected EEPROM device. The new code is intended to support future daughter cards for the dk-lm3s9b96 board which will feature this EEPROM device and is used in all dk-lm3s9b96 example applications by default. To allow simplified EPI and pinout configurations and to remove the overhead of the new dynamic configuration code, a second implementation of the PinoutSet() function is provided which merely configures the pinout and EPI based on a hardcoded configuration. This may be enabled by building the file with label SIMPLE_PINOUT_SET defined.

2.8.6 Embedded web site rework (Reference 10517)

The embedded web sites served by the enet_io and enet_lwip example applications have been reworked to use 8.3 filenames. As a result, these sites can now be copied to SDCard and served from there when using the enet_lwip application.

2.8.7 New example applications ext_demo_1 and ext_demo_2 added (Reference 9968)

Two new example applications illustrating execution from EPI-connected flash have been added to the dk-lm3s9b96 release. Both are intended for use with the SRAM/Flash/LCD daughter board and the external flash Ethernet boot loader (boot_eth_ext).

2.8.8 Display driver updated to support SRAM/Flash/LCD daughter board (Reference 9511)

The kitronix320x240x16_ssd2119_8bit.c display driver for dk-lm3s9b96 has been updated to support the SRAM/Flash/LCD daughter board and dynamically switch between GPIO-based and EPI-based access to the display depending upon the hardware detected. The display type determination is made inside function PinoutSet() which can be found in file drivers/set_pinout.c.

2.8.9 Added an example boot loader targeting external flash (Reference 9513)

Example application boot_eth_ext has been added to the dk-lm3s9b96 StellarisWare package. This is a version of the Ethernet boot loader which can be used to download and run applications linked to run from the address space of the external flash found on the optional SRAM/Flash/LCD daughter board.

2.9 Bug Fixes in DK-LM3S9B96 Firmware Package

2.9.1 Error in epi.h address size definitions corrected (Reference 10461)

An error in epi.h has been corrected and affected example applications have been updated. Labels EPI_ADDR_PER_SIZE_512MB and EPI_ADDR_RAM_SIZE_512MB have been replaced with the correct definitions for those register bit patterns, EPI_ADDR_PER_SIZE_256MB and EPI_ADDR_RAM_SIZE_256MB.

2.10 New Features in EK-LM3S811 Firmware Package

2.10.1 Added support for RIT display on new ek-lm3s811 boards (Reference 10106)

The display driver for the ek-lm3s811 board has been updated to support both the OSRAM display found on the existing boards and the RIT display use on newer revisions. Since the driver is no longer specific to the OSRAM display, it has been renamed from osram96x16x1 to display96x16x1 and all example applications have been updated accordingly. Macros are provided in the new display96x169x.h header file to ensure that the previous API calls will be correctly remapped to the new functions.

This change causes the code size of the display driver to increase slightly but the new driver offers the ability to build for either one or other display by defining either OSRAM_ONLY or RIT_ONLY via the project file or makefile when building. If neither label is defined, the driver will include support for both displays and will determine which is required at runtime.

2.11 Bug Fixes in EK-LM3S811 Firmware Package

2.11.1 Quickstart application rebranded (Reference 10397)

The opening graphic displayed by the qs_ek-lm3s811 application has been changed to show the Texas Instruments logo rather than the Luminary Micro logo.

2.12 New Features in EK-LM3S2965 Firmware Package

2.12.1 Add a CAN FIFO example. (Reference 10430)

Two new CAN examples were added that use FIFO mode to transfer data to and from the main board to the CAN device board. The can_fifo example is run on the main board and the can_device_fifo is run on the CAN device board. These applications demonstrates how to use the DriverLib CAN APIs to use multiple CAN message objects as a FIFO for transferring data.

2.13 New Features in EK-LM3S6965 Rev A Firmware Package

2.13.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.13.2 Embedded web site rework (Reference 10524)

The embedded web site served by the enet_lwip example application has been reworked to use 8.3 filenames. As a result, this site can now be copied to SDCard and served from there when using the enet_lwip application.

2.14 Bug Fixes in EK-LM3S6965 Rev A Firmware Package

2.14.1 Embedded web site rebranding (Reference 10538)

The web sites served by the enet_io and enet_ptpd example applications have been rebranded to show Texas Instruments logos and information rather than the Luminary Micro equivalents. In the process, filenames were also updated to conform to the 8.3 naming convention to ease use of the sites from SDCards rather than internal flash.

2.15 New Features in EK-LM3S6965 Firmware Package

2.15.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.15.2 Embedded web site rework (Reference 10523)

The embedded web sites served by the enet_ptpd, enet_io and enet_lwip example applications have been reworked to use 8.3 filenames. As a result, these sites can now be copied to SDCard and served from there when using the enet_lwip application.

2.16 New Features in EK-LM3S8962 Firmware Package

2.16.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.16.2 Add a CAN FIFO example. (Reference 10430)

Two new CAN examples were added that use FIFO mode to transfer data to and from the main board to the CAN device board. The can_fifo example is run on the main board and the can_device_fifo is run on the CAN device board. These applications demonstrates how to use the DriverLib CAN APIs to use multiple CAN message objects as a FIFO for transferring data.

2.16.3 Embedded web site rework (Reference 10523)

The embedded web sites served by the enet_ptpd, enet_io and enet_lwip example applications have been reworked to use 8.3 filenames. As a result, these sites can now be copied to SDCard and served from there when using the enet_lwip application.

2.17 New Features in EK-LM3S9B90 Firmware Package

2.17.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.17.2 Add uDMA support to the enet_uip example application (Reference 10196)

The enet_uip example has been modified to add support for using uDMA with the Ethernet controller. This change is meant to demonstrate how an application can use the uDMA controller with the Ethernet controller in a general application.

2.17.3 EPIConfigNoModeSet renamed to EPIConfigGPModeSet (Reference 10247)

The DriverLib API function EPIConfigNoModeSet has been renamed EPIConfigGPModeSet to ensure that the terminology used in the driver matches the part datasheets. Similarly, several labels

defined in epi.h have been renamed to ensure consistency with the microcontroller documentation. The previous function and labels defined in epi.h have been deprecated but macros have been added to the header file to ensure that existing code using the function will still build and operate.

2.17.4 Embedded web site rework (Reference 10525)

The embedded web sites served by the enet_ptpd and enet_lwip example applications have been reworked to use 8.3 filenames and present TI brand information.

2.18 New Features in EK-LM3S9B92 Firmware Package

2.18.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.18.2 Add uDMA support to the enet_uip example application (Reference 10196)

The enet_uip example has been modified to add support for using uDMA with the Ethernet controller. This change is meant to demonstrate how an application can use the uDMA controller with the Ethernet controller in a general application.

2.18.3 EPIConfigNoModeSet renamed to EPIConfigGPModeSet (Reference 10247)

The DriverLib API function EPIConfigNoModeSet has been renamed EPIConfigGPModeSet to ensure that the terminology used in the driver matches the part datasheets. Similarly, several labels defined in epi.h have been renamed to ensure consistency with the microcontroller documentation. The previous function and labels defined in epi.h have been deprecated but macros have been added to the header file to ensure that existing code using the function will still build and operate.

2.18.4 Embedded web site rework (Reference 10525)

The embedded web sites served by the enet_ptpd and enet_lwip example applications have been reworked to use 8.3 filenames and present TI brand information.

2.19 Bug Fixes in RDK-ACIM Firmware Package

2.19.1 Change motor kit GUI install and start menu locations (Reference 10547)

The motor kit GUI programs install location has been changed from “Luminary Micro” to “Texas Instruments/Stellaris” in the Program Files directory. The Windows start menu location has likewise been changed so that the programs now appear under “Texas Instruments/Stellaris”.

2.20 New Features in RDK-BLDC Firmware Package

2.20.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.21 Bug Fixes in RDK-BLDC Firmware Package

2.21.1 Change motor kit GUI install and start menu locations (Reference 10547)

The motor kit GUI programs install location has been changed from “Luminary Micro” to “Texas Instruments/Stellaris” in the Program Files directory. The Windows start menu location has likewise been changed so that the programs now appear under “Texas Instruments/Stellaris”.

2.22 New Features in RDK-IDM-SBC Firmware Package

2.22.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.22.2 Functions SDRAMAlloc and SDRAMFree have been renamed. (Reference 10268)

The SDRAM memory management functions provided in drivers/sdram.c have been renamed to ExtRAMAlloc and ExtRAMFree to match the similar functions provided by the dk-lm3s9b96 software

release. These functions are used by the shared JPEG decoder software so must match across all releases which use JPEG. The previous functions are deprecated but macro definitions in sdram.h will remap them to the new names, ensuring that existing code which uses them will continue to build and operate.

2.23 Bug Fixes in RDK-IDM-SBC Firmware Package

2.23.1 Documentation correction (Reference 10139)

The documentation for the idm-checkout example application in the rdk-idm-sbc release of StellarisWare incorrectly stated that the board comes with a web site image already stored in the serial flash device. This is not the case - the user must download this image using TFTP if desired. The documentation has been updated to reflect this.

2.24 New Features in RDK-S2E Firmware Package

2.24.1 Add new GPIOPinTypeEthernetLED API (Reference 10090)

Add new API, GPIOPinTypeEthernetLED, to GPIO driver in driverlib. In this function the default drive strength for the Ethernet LED pins is now 8ma. Modified all of the sample applications that were enabling the Ethernet LED pin functions “manually” to now use this new API.

2.24.2 Embedded web site rework (Reference 10528)

The embedded configuration web site for the ser2enet application has been reworked to present TI brand information and the factory default name for the module has been changed from “Luminary Micro Serial2Ethernet Module” to “TI Stellaris Serial2Ethernet Module”. No functional changes were made to the application.

2.25 Bug Fixes in RDK-STEPPER Firmware Package

2.25.1 Fix Stack Overflow in RDK-Stepper Application (Reference 10459)

Under some circumstances the stack could overflow in the qs-stepper application. The stack size has been increased to accommodate the additional required stack space.

2.25.2 Change motor kit GUI install and start menu locations (Reference 10547)

The motor kit GUI programs install location has been changed from “Luminary Micro” to “Texas Instruments/Stellaris” in the Program Files directory. The Windows start menu location has likewise been changed so that the programs now appear under “Texas Instruments/Stellaris”.

3 Release Notes for StellarisWare Revision 4905 (July 30, 2009)

3.1 New Features in Stellaris Boot Loader Library

3.1.1 Improved boot loader performance for dk-lm3s9b96 (Reference 9842)

The boot loader was reworked to offer the ability to replace the low level flash sizing, erase and programming functions. For Tempest-class devices such as the lm3s9b96, the boot loader flash programming function has been replaced with a version which makes use of the flash write buffer, improving download performance dramatically compared to the previous release. Note that the ROM-based boot loaders for Tempest-class devices were already making use of the flash write buffer - this change only affects boot loaders built to run from flash or SRAM.

3.1.2 Boot loader now allows vector table to be initialized in SRAM (Reference 9993)

A new label, VTABLE_START_ADDRESS, was added to the list supported by bl_config.h to allow an application to specify that its vector table should be relocated to SRAM prior to the boot loader transfering control to it. This is intended to support applications running out of EPI-connected memory where the vector table at the beginning of the application image is not accessible by the NVIC.

3.2 Bug Fixes in Stellaris Peripheral Driver Library

3.2.1 Missing configuration options added for EPIConfigHB8Set and EPIConfigNoModeSet (Reference 9778)

Additional flags have been added for use in the ulConfig parameter passed to EPIConfigNoModeSet and EPIConfigHB8Set. These allow selection of word access mode and, for the HostBus8 case, allow the function of CS to be defined. Prior to this addition, direct access to an EPI configuration register was required to set these options.

3.2.2 USBDevEndpointConfigSet() does not properly configure isochronous endpoints (Reference 9856)

The USBDevEndpointConfigSet() had an incorrect conditional statement that causes isochronous endpoints to be configured incorrectly. This causes the endpoint to acknowledge incoming packets, which results in the host controller not transmitting any more packets to that endpoint.

3.3 Bug Fixes in Stellaris Graphics Library

3.3.1 Rendering of 1bpp and 4bpp compressed images (Reference 9642)

GrImageDraw() now properly displays 1bpp and 4bpp compressed images. Due to a pair of rounding errors, previously it would incorrectly display images that were not a multiple of 8 pixels wide (1bpp) or 2 pixels wide (4bpp).

3.4 Bug Fixes in Stellaris Host Tools Library

3.4.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not

specifically handled and defaulted to MIME type text/plain.

3.5 New Features in Stellaris USB Library

3.5.1 Add support for USB audio class in device mode. (Reference 9894)

This update adds support for USB audio device class to the USB library. The USB audio class support only audio playback with volume control implemented as well. The audio device class is implemented as a generic USB audio class and will work with any operating system that supports USB audio class devices without the need for additional operating system drivers.

3.6 Bug Fixes in Stellaris USB Library

3.6.1 USB library enumeration code not properly clearing FIFO flags. (Reference 10044)

When handling standard device request, the USB library was not always clearing the FIFO read flag when a new request was received. This could effect any commands that followed the request that failed to clear the FIFO read flag.

3.6.2 USB library not handling VBUS errors in OTG mode. (Reference 10100)

The USB library is not properly handling VBUS error interrupts during host enumeration in OTG mode. If VBUS error occurs during enumeration, it causes the USB library code to hang in an intermediate state and not properly turn off power and retry the connection.

3.7 New Features in Stellaris Utility Library

3.7.1 New function ustrnicmp added to the ustdlib module (Reference 9862)

An implementation of the standard C runtime strnicmp (compare strings without regard to case) has been added to the ustrlib module.

3.7.2 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of

transfering files to and from an application over Ethernet.

3.8 Bug Fixes in Stellaris Utility Library

3.8.1 Fix UDP-Only configuration of LWIP (Reference 9898)

Add conditional compilation wrapper to the code that supports the TCP timer. This will prevent undefined references when compiling the LWIP library for UDP only (i.e. no TCP).

3.9 Bug Fixes in Stellaris Third Party Libraries

3.9.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.10 New Features in DK-LM3S9B96 Firmware Package

3.10.1 Improved boot loader performance for dk-lm3s9b96 (Reference 9842)

The boot loader was reworked to offer the ability to replace the low level flash sizing, erase and programming functions. For Tempest-class devices such as the lm3s9b96, the boot loader flash programming function has been replaced with a version which makes use of the flash write buffer, improving download performance dramatically compared to the previous release. Note that the ROM-based boot loaders for Tempest-class devices were already making use of the flash write buffer - this change only affects boot loaders built to run from flash or SRAM.

3.10.2 qs-checkout example now uses general-purpose TFTP server (Reference 9976)

The qs-checkout example application for dk-lm3s9b96 has been updated to make use of the new, general-purpose TFTP module (utils/tftp.c).

3.10.3 EPI configuration moved to PinoutSet() function (Reference 10012)

Configuration of the External Peripheral Interface (EPI) has been moved from individual drivers into the PinoutSet() function. This ensures that any external memories attached via EPI are accessible from early in the application startup process.

3.10.4 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.11 Bug Fixes in DK-LM3S9B96 Firmware Package

3.11.1 Missing configuration options added for EPIConfigHB8Set and EPIConfigNoModeSet (Reference 9778)

Additional flags have been added for use in the ulConfig parameter passed to EPIConfigNoModeSet and EPIConfigHB8Set. These allow selection of word access mode and, for the HostBus8 case, allow the function of CS to be defined. Prior to this addition, direct access to an EPI configuration register was required to set these options.

3.11.2 Sound driver improperly calls buffer callback function. (Reference 10010)

The sound driver could call a buffer's callback function after the buffer had already been used. The buffer callback will now only be called if the buffer pointer is still valid.

3.11.3 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.11.4 Fix compatibility problem with recent Keil compiler in the usb_stick_update example application (Reference 10038)

The static declaration on a function was removed to work around a problem in the recent Keil toolchain (3.50) that was causing an internal fault in the compiler.

3.12 Bug Fixes in EK-LM3S3748 Firmware Package

3.12.1 Fix compatibility problem with recent Keil compiler in the usb_stick_update example application (Reference 10038)

The static declaration on a function was removed to work around a problem in the recent Keil toolchain (3.50) that was causing an internal fault in the compiler.

3.13 New Features in EK-LM3S6965 Firmware Package

3.13.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.14 Bug Fixes in EK-LM3S6965 Firmware Package

3.14.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.15 New Features in EK-LM3S8962 Firmware Package

3.15.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.16 Bug Fixes in EK-LM3S8962 Firmware Package

3.16.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.17 New Features in EK-LM3S9B90 Firmware Package

3.17.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.18 Bug Fixes in EK-LM3S9B90 Firmware Package

3.18.1 Missing configuration options added for EPIConfigHB8Set and EPIConfigNoModeSet (Reference 9778)

Additional flags have been added for use in the ulConfig parameter passed to EPIConfigNoModeSet and EPIConfigHB8Set. These allow selection of word access mode and, for the HostBus8 case, allow the function of CS to be defined. Prior to this addition, direct access to an EPI configuration register was required to set these options.

3.18.2 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.18.3 Fix compatibility problem with recent Keil compiler in the usb_stick_update example application (Reference 10038)

The static declaration on a function was removed to work around a problem in the recent Keil toolchain (3.50) that was causing an internal fault in the compiler.

3.19 New Features in EK-LM3S9B92 Firmware Package

3.19.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.20 Bug Fixes in EK-LM3S9B92 Firmware Package

3.20.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.20.2 Fix compatibility problem with recent Keil compiler in the usb_stick_update example application (Reference 10038)

The static declaration on a function was removed to work around a problem in the recent Keil toolchain (3.50) that was causing an internal fault in the compiler.

3.21 New Features in RDK-IDM Firmware Package

3.21.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.22 Bug Fixes in RDK-IDM Firmware Package

3.22.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.23 New Features in RDK-IDM-SBC Firmware Package

3.23.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.23.2 Add USB Memory Stick Updater Application (Reference 10048)

Two new applications have been added to demonstrate the ability to perform a firmware update over USB from a memory stick. The application usb_stick_update is the updater, and the application usb_stick_demo provides an example that can be loaded from a USB memory stick.

3.24 Bug Fixes in RDK-IDM-SBC Firmware Package

3.24.1 Sound driver improperly calls buffer callback function. (Reference 10010)

The sound driver could call a buffer's callback function after the buffer had already been used. The buffer callback will now only be called if the buffer pointer is still valid.

3.24.2 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

3.25 New Features in RDK-S2E Firmware Package

3.25.1 General purpose TFTP server module added (Reference 10053)

A new general purpose TFTP server module has been added to the utils directory for all kits supporting Ethernet. This module makes use of the lwIP TCP/IP stack and offers a simple method of transfering files to and from an application over Ethernet.

3.26 Bug Fixes in RDK-S2E Firmware Package

3.26.1 MIME type for icons is now correctly set (Reference 10021)

The lwIP HTTPD server and makefsfile tool have both been updated to correctly recognize files with extension ".ico" and report these as type image/x-icon. In previous releases, these were not specifically handled and defaulted to MIME type text/plain.

4 Release Notes for StellarisWare Revision 4781 (June 30, 2009)

4.1 New Features in Stellaris Peripheral Driver Library

4.1.1 Add API for ADC Digital Comparators (Reference 9668)

In the Tempest class devices, a Digital Comparator feature has been added to the ADC module. This feature allows ADC samples to be sent to a digial comparator. This comparator can be programmed to trigger on low-band, mid-band or high-band values, and the trigger can be used to generate an interrupt or trigger a fault condition to the PWM module. The ADC API has been expanded to provide support for this new feature. Additional details about the operation of the digital comparator can be found in the data sheets for the Tempest Class Stellaris devices.

4.1.2 Added support devices that support 32 USB endpoints. (Reference 9666)

Newer devices allow for more endpoints with up to 32 IN/OUT endpoints now available. Support for the additional endpoints was added to the DriverLib USB functions and examples of using the new APIs were added to the USB library. The main change to the DriverLib API was to deprecate the interrupt handling functions because they could not support 32 endpoints as defined. The deprecated APIs are USBIntDisable(), USBIntEnable(), USBIntStatus() which can still be used with older devices that have only 8 endpoints. There are six added APIs that provide the same functionality, except the new APIs are split between control interrupts and endpoint interrupts. The new APIs are the following: USBIntDisableControl(), USBIntEnableControl(), USBIntStatusControl(), USBIntDisableEndpoint(), USBIntEnableEndpoint(), and USBIntStatusEndpoint(). The flags to use with the new USB control interrupt functions start with USB_INTCTRL_ while the new USB endpoint interrupt functions use the USB_INTEP_ flags.

4.2 Bug Fixes in Stellaris Peripheral Driver Library

4.2.1 SysCtlClockGet() provided incorrect results in some cases (Reference 9555)

If the PLL is enabled, SysCtlClockGet() now applies the system divider to the computed PLL output frequency even if the USESYSDIV bit in RCC is not set. It is possible to configure RCC where the PLL is enabled and USESYSDIV is not set, but the device forces the use of the system divider (since the PLL is enabled). This change causes SysCtlClockGet() to mimic the behavior of the device and therefore provide correct results in this case.

4.2.2 USBDevEndpointConfig() deprecated in favor of USBDevEndpointConfigSet() (Reference 9297)

The function USBDevEndpointConfig() has been marked as DEPRECATED and the name has been changed to USBDevEndpointConfigSet() to be symmetrical with the USBDevEndpointConfigGet() API. This has no affect on any current code however the definitions for USBDevEndpointConfig() may be removed at some point in the future.

4.2.3 SysCtlPeripheralPresent() did not properly handle USB (Reference 9756)

The SysCtlPeripheralPresent() API added a case to handle the USB peripheral because the current definition will incorrectly indicate the presence of the USB0 peripheral even when there is no USB controller present.

4.3 Bug Fixes in Stellaris Graphics Library

4.3.1 WidgetRemove() now clears the widget's next pointer (Reference 9615)

The next pointer on a widget is now cleared when it is removed from the widget tree with WidgetRemove(). If the widget is later added back to the widget tree with WidgetAdd(), it will no longer corrupt the widget tree since the next pointer no longer points to a potentially invalid widget.

4.3.2 GrStringSet() did not properly handle the ulSize parameter (Reference 9630)

The GrStringGet() was not using the ulSize parameter in all cases and was allowing the function to write beyond the end of the buffer provided to the function. This could cause other variables or data to overwritten with data for a given string.

4.4 New Features in DK-LM3S9B96 Firmware Package

4.4.1 Add USB Memory Stick Updater Application (Reference 9722)

Two new applications have been added to demonstrate the ability to perform a firmware update over USB from a memory stick. The application usb_stick_update is the updater, and the application usb_stick_demo provides an example that can be loaded from a USB memory stick.

4.5 New Features in EK-LM3S3748 Firmware Package

4.5.1 Add USB Memory Stick Updater Application (Reference 9722)

Two new applications have been added to demonstrate the ability to perform a firmware update over USB from a memory stick. The application usb_stick_update is the updater, and the application usb_stick_demo provides an example that can be loaded from a USB memory stick.

4.6 New Features in EK-LM3S9B90 Firmware Package

4.6.1 Add USB Memory Stick Updater Application (Reference 9722)

Two new applications have been added to demonstrate the ability to perform a firmware update over USB from a memory stick. The application usb_stick_update is the updater, and the application usb_stick_demo provides an example that can be loaded from a USB memory stick.

4.7 New Features in EK-LM3S9B92 Firmware Package

4.7.1 Add USB Memory Stick Updater Application (Reference 9722)

Two new applications have been added to demonstrate the ability to perform a firmware update over USB from a memory stick. The application usb_stick_update is the updater, and the application usb_stick_demo provides an example that can be loaded from a USB memory stick.

4.8 Bug Fixes in RDK-IDM Firmware Package

4.8.1 sd_card application was not properly configuring the Graphics Library (Reference 9793)

The sd_card application was failing to properly configure the Graphics Library before calling GrStringDraw() which was causing the application to halt. This problem occurred whenever a re-

quest to update the firmware was received from the Ethernet controller.

4.9 Bug Fixes in RDK-IDM-SBC Firmware Package

4.9.1 Corrected text misalignment in usb_host_mouse and usb_host_keyboard (Reference 9787)

The text strings in the status panel at the bottom of the display for the IDM-SBC versions of usb_host_mouse and usb_host_keyboard have been moved to prevent possible overlap.

4.10 Bug Fixes in Stellaris Firmware Development Package

4.10.1 Updated project files for Sourcery G++ for Stellaris (Reference 9667)

The project files for Sourcery G++ for Stellaris have been updated to reflect the requirements of the new version of CodeSourcery's tool chain. This fixes some of the project/workspace import problems that occurred with the previous version of StellarisWare and Sourcery G++ for Stellaris.

5 Release Notes for StellarisWare Revision 4694 (May 27, 2009)

5.1 New Features in Stellaris Graphics Library

5.1.1 Additions to the ImageButton widget (Reference 9484)

New functionality has been added to the ImageButton widget offered by the Graphics Library. The widget now supports IB_STYLE_FILL for drawing a background color and new macros allow the button background and keycap images to be enabled and disabled. These changes have been implemented to keep the interface backwards compatible, hence the use of style flags IB_STYLE_KEYCAP_OFF and IB_STYLE_IMAGE_OFF (since the previous version of the widget assumed that both images were disabled unless a NULL pointer was used to populate the relevant image pointer).

5.2 New Features in Stellaris USB Library

5.2.1 USB host event driver added to USB library (Reference 9534)

An event driver was added to the USB host library to provide notification of important system events and class specific events that were previously not visible to the application.

5.3 Bug Fixes in DK-LM3S9B96 Firmware Package

5.3.1 i2s_demo application report wrong elapsed time for 8bit wav files. (Reference 8973)

There was a problem in the interpretation of the byte rate of .wav files being played that caused the byte rate calculation to be incorrect by a factor of 2 for 8 bit .wav files.

6 Release Notes for StellarisWare Revision 4674 (May 19, 2009)

6.1 Bug Fixes in Stellaris Boot Loader Library

6.1.1 Ethernet boot loader hangs in some cases (Reference 9240)

A delay has been added between enabling the Ethernet controller and accessing it. If the boot loader was configured to enable the Ethernet LEDs, this was performing the function of that delay. If the LEDs were not used, a fault would occur since the Ethernet controller was accessed too quickly after being enabled. This delay resolves that problem in the case that the LEDs are not used.

6.2 New Features in Stellaris Peripheral Driver Library

6.2.1 Added two new uDMA API functions to support the interrupt status register in Tempest (Reference 9179)

Added two new functions to the uDMA API: uDMAIntStatus() and uDMAIntClear() to support the new DMA interrupt status register that is available in Tempest class parts.

6.2.2 Add CAN Bit Rate API (Reference 9315)

A simplified CAN Bit Rate API, CANBitRateSet() was added to provide an easier method of setting the CAN bit timing as opposed to having to fully specify the CAN bit timing with the CANBitTimingSet() API. The new API can directly set the CAN bit rate based on the clock source for the CAN controller. The CANBitTimingSet() API can still be used if more precise timing parameters are needed by an application.

6.2.3 Added I2S and EPI drivers to DriverLib (Reference 9419)

Drivers have been added for the I2S and EPI peripherals that are available on the new Stellaris parts.

6.3 Bug Fixes in Stellaris Graphics Library

6.3.1 Corrected operation of ListBoxLock() and ListBoxUnlock() (Reference 9441)

In previous releases, the operation of the ListBoxLock() and ListBoxUnlock() macros was reversed. This has now been corrected.

6.3.2 Corrected operation of Lock and Unlock macros for Slider and JPGWidget (Reference 9471)

In previous releases, the operation of the SliderLock()/SliderUnlock() and JPEGWidgetLock()/JPEGWidgetUnlock() macros were reversed. This has now been corrected.

6.4 New Features in Stellaris Host Tools Library

6.4.1 New board locater tool for Ethernet-based applications (Reference 9094)

A board locater tool has been created that will search the Ethernet network for Stellaris-based boards running code which utilizes Ethernet and the board locater service. This allows an easy method to discover the presence, IP address, and MAC address of the Ethernet-based boards on the network, as well as a description of the application that is running on that board.

6.4.2 Windows USB Examples have moved to the tools directory (Reference 9388)

The Windows USB example applications which were previously found in the StellarisWare/boards/ek-lm3s3748/windows_examples directory have moved to the StellarisWare/tools directory instead. Previously, these examples only applied to the lm3s3748 kit but, with the introduction of new lm3s9b90, lm3s9b92 and lm3s9b96 kits, they are now required by several boards so this change ensures that they are in a single, common location for all boards that make use of them.

6.4.3 Update to Red Suite Project Import XML Files (Reference 9445)

Red Suite Version 2 upgrades the compiler tools to version 4.3.2. For building the boot loader, the -Os option should be specified now (same as for Code Sourcery and GCC). Also, the way that compiler defines are specified has changed and the new XML files will support these changes.

6.5 Bug Fixes in Stellaris Host Tools Library

6.5.1 Makefsfile updated to prevent generation of invalid C code (Reference 8651)

The makefsfile tool was updated to ensure that filenames containing characters which are not legal within C variable names would be correctly translated into something that is valid C. In the previous version, only spaces, dots and slashes were replaced with underscores. The new version adds all the non-alphanumeric (shifted) characters to this list. Without this change, it was possible to generate a C file system image file which would not compile if filenames in the directory being imaged contained characters such as ‘-’, ‘+’.

6.5.2 makefsfile tool now adds correct headers to XML files (Reference 9361)

The makefsfile tool, used to generate images for internal file systems which can be used by the lwIP HTTP server, would previously describe any file with a '.xml' extension as 'text/plain' rather than 'text/xml'. This caused problems when using AJAX since the XMLHttpRequest object would not have the responseXML field set when the asynchronous request completed (the response would be stored only in the responseText field).

6.5.3 Library files for Windows USB DLLs have been added to the release (Reference 9386)

In previous releases, the library files lmusbdll.lib and lmdfu.lib were missing making it impossible to build some of the USB example applications without having access to the Windows Device Driver Kit. These files have now been added to the appropriate directories under StellarisWare/tools allowing the examples to be built. Additionally, copies of the files have been included in the Windows device driver package (SW-USB-windrivers) which already includes the DLLs that these library files relate to, lmusbdll.dll and lmdfu.dll.

6.6 Bug Fixes in Stellaris USB Library

6.6.1 Fixed a USB host MSC bug causing a hang on multi-block reads (Reference 9411)

A bug existed in the previous release which would cause USBHMSCBlockRead() and USBHMSCBlockWrite() to hang if passed a ulNumBlocks value greater than 1. This was due to an assumption in usbhscsi.c that all reads and writes would be performed on a block-by-block basis resulting in incorrect block numbers being written to the read and write command blocks generated in USBHSCSIRead10() and USBHSCSIWrite10().

6.6.2 USBDCDInit() now disconnects before reconnecting (Reference 9442)

The USBLib device initialization function USBDCDInit() now explicitly disconnects the device from the bus and delays approximately 100mS before connecting it once again. The previous version of the function did not perform this disconnect operation and, as a result, if the function was called when the device was already connected to the USB bus, it would not be reenumerated resulting in missing callbacks to the application and resulting application confusion.

6.7 New Features in Stellaris Utility Library

6.7.1 Added function fs_map_path() to fswrapper module (Reference 9322)

The fswrapper module offers a method to give multiple file system images user-friendly names in web URLs. It can support FAT logical drives and binary file system images but only provides the subset of file system operations typically required by a web server. To allow access to the more advanced functions provided by FatFS for FAT logical drives, a new API has been provided, fs_map_path(), which will map a path in the fswrapper namespace to the equivalent path at the FatFS level (for mount points that correspond to FAT logical drives). For example, passing the string "/sdcard/index.htm" would return "0:/index.htm" assuming the mount point name "sdcard" is associated with FAT logical drive number 0.

6.8 Bug Fixes in Stellaris Utility Library

6.8.1 Change between static and DHCP IP sometimes fails (Reference 9438)

The function, lwIPNetworkConfigChange, does not always switch properly between static IP and Auto IP (with DHCP). This is due to the fact that the variable that retains the current IP mode setting is not properly saved. This variable, g_ulIPMode, is now saved at the end of the function for all cases.

6.9 New Features in Stellaris Third Party Libraries

6.9.1 Added support for AES ROM tables in Tempest class parts (Reference 9089)

Modified the AES code in third_party to use the AES tables from ROM for Tempest class devices. Also modified the AES example applications for Tempest based boards.

6.10 Bug Fixes in Stellaris Third Party Libraries

6.10.1 Closed lwIP HTTPD timing hole that could cause hangs on connection shutdown (Reference 9256)

A race condition in the lwIP HTTPD server which could cause a NULL pointer to be dereferenced in some cases during connection termination was fixed.

6.10.2 lwIP HTTP server now sends correct headers for XML files (Reference 9358)

The lwIP HTTPD server previously described XML files using header “text/plain”. This caused problems for AJAX browser applications since the XML responses were not parsed correctly when received. The server now uses the correct “text/xml” header with any file whose extension is “.xml”.

6.11 New Features in DK-LM3S9B96 Firmware Package

6.11.1 Added support for AES ROM tables in Tempest class parts (Reference 9089)

Modified the AES code in third_party to use the AES tables from ROM for Tempest class devices. Also modified the AES example applications for Tempest based boards.

6.12 Bug Fixes in DK-LM3S9B96 Firmware Package

6.12.1 Library files for Windows USB DLLs have been added to the release (Reference 9386)

In previous releases, the library files lmusbdll.lib and lmdfu.lib were missing making it impossible to build some of the USB example applications without having access to the Windows Device Driver Kit. These files have now been added to the appropriate directories under StellarisWare/tools allowing the examples to be built. Additionally, copies of the files have been included in the Windows device driver package (SW-USB-windrivers) which already includes the DLLs that these library files relate to, lmusbdll.dll and lmdfu.dll.

6.12.2 Corrected operation of Lock and Unlock macros for Slider and JPGWidget (Reference 9471)

In previous releases, the operation of the SliderLock()/SliderUnlock() and JPEGWidgetLock()/JPEGWidgetUnlock() macros were reversed. This has now been corrected.

6.12.3 Web server opens Luminary Micro site in the wrong frame (Reference 9488)

In various applications supporting an embedded web server (depending upon the kit, enet_io, enet_lwip, qs-checkout and idm-checkout) used to open the link to `http://www.luminarymicro.com` within a frame. The sites have been updated to open this link in the top level window instead.

6.13 New Features in EK-LM3S3748 Firmware Package

6.13.1 Windows USB Examples have moved to the tools directory (Reference 9388)

The Windows USB example applications which were previously found in the StellarisWare/boards/ek-lm3s3748/windows_examples directory have moved to the StellarisWare/tools directory instead. Previously, these examples only applied to the lm3s3748 kit but, with the introduction of new lm3s9b90, lm3s9b92 and lm3s9b96 kits, they are now required by several boards so this change ensures that they are in a single, common location for all boards that make use of them.

6.14 Bug Fixes in EK-LM3S3748 Firmware Package

6.14.1 Library files for Windows USB DLLs have been added to the release (Reference 9386)

In previous releases, the library files lmusbdll.lib and lmdfu.lib were missing making it impossible to build some of the USB example applications without having access to the Windows Device Driver Kit. These files have now been added to the appropriate directories under StellarisWare/tools allowing the examples to be built. Additionally, copies of the files have been included in the Windows device driver package (SW-USB-windrivers) which already includes the DLLs that these library files relate to, lmusbdll.dll and lmdfu.dll.

6.14.2 Stack overflow in usb_dev_serial example (Reference 9446)

The stack size allocated for the usb_dev_serial example was increased to prevent an overflow which had been seen occasionally in a previous version of the application.

6.14.3 Bitband example was failing to run on all tool chains. (Reference 9443)

The bitband example was failing on some tool chains due to the stack not being large enough. The stack size was increased to prevent the stack overflow from causing the application to crash.

6.15 Bug Fixes in EK-LM3S6965 Rev A Firmware Package

6.15.1 enet_ptpd web server occasionally returns too much data (Reference 9435)

The file system module in the enet_ptpd application used strlen() to determine the amount of data that should be served up by the web server instead of the file size that is stored in the file system structure. Return the stored size instead since the strlen() size since the later can be incorrect at times (if there is no trailing NULL in the file data).

6.15.2 Replace use of strstr with ustrstr (Reference 9447)

To avoid potential runtime library issues that vary from toolchain to toolchain, replace the use of the strstr function with ustrstr, which is provided in the utils folder.

6.16 Bug Fixes in EK-LM3S6965 Firmware Package

6.16.1 enet_ptpd web server occasionally returns too much data (Reference 9435)

The file system module in the enet_ptpd application used strlen() to determine the amount of data that should be served up by the web server instead of the file size that is stored in the file system structure. Return the stored size instead since the strlen() size since the later can be incorrect at times (if there is no trailing NULL in the file data).

6.16.2 Replace use of strstr with ustrstr (Reference 9447)

To avoid potential runtime library issues that vary from toolchain to toolchain, replace the use of the strstr function with ustrstr, which is provided in the utils folder.

6.17 Bug Fixes in EK-LM3S8962 Firmware Package

6.17.1 enet_ptpd web server occasionally returns too much data (Reference 9435)

The file system module in the enet_ptpd application used strlen() to determine the amount of data that should be served up by the web server instead of the file size that is stored in the file system structure. Return the stored size instead since the strlen() size since the later can be incorrect at times (if there is no trailing NULL in the file data).

6.17.2 Replace use of strstr with ustrstr (Reference 9447)

To avoid potential runtime library issues that vary from toolchain to toolchain, replace the use of the strstr function with ustrstr, which is provided in the utils folder.

6.18 New Features in EK-LM3S9B90 Firmware Package

6.18.1 Added applications for new evaluation board (Reference 9348)

A suite of example applications has been added for the new evaluation board.

6.18.2 Added support for AES ROM tables in Tempest class parts (Reference 9089)

Modified the AES code in third_party to use the AES tables from ROM for Tempest class devices. Also modified the AES example applications for Tempest based boards.

6.19 Bug Fixes in EK-LM3S9B90 Firmware Package

6.19.1 Library files for Windows USB DLLs have been added to the release (Reference 9386)

In previous releases, the library files lmusbdll.lib and lmdfu.lib were missing making it impossible to build some of the USB example applications without having access to the Windows Device Driver Kit. These files have now been added to the appropriate directories under StellarisWare/tools allowing the examples to be built. Additionally, copies of the files have been included in the Windows device driver package (SW-USB-windrivers) which already includes the DLLs that these library files relate to, lmusbdll.dll and lmdfu.dll.

6.20 New Features in EK-LM3S9B92 Firmware Package

6.20.1 Added applications for new evaluation board (Reference 9348)

A suite of example applications has been added for the new evaluation board.

6.20.2 Added support for AES ROM tables in Tempest class parts (Reference 9089)

Modified the AES code in third_party to use the AES tables from ROM for Tempest class devices. Also modified the AES example applications for Tempest based boards.

6.21 Bug Fixes in EK-LM3S9B92 Firmware Package

6.21.1 Library files for Windows USB DLLs have been added to the release (Reference 9386)

In previous releases, the library files lmusbdll.lib and lmdfu.lib were missing making it impossible to build some of the USB example applications without having access to the Windows Device Driver Kit. These files have now been added to the appropriate directories under StellarisWare/tools allowing the examples to be built. Additionally, copies of the files have been included in the Windows device driver package (SW-USB-windrivers) which already includes the DLLs that these library files relate to, lmusbdll.dll and lmdfu.dll.

6.22 Bug Fixes in RDK-BLDC Firmware Package

6.22.1 Enhance Hall Sensor Speed Calculation (Reference 9476)

Modify the speed calculation algorithm to use every rising/falling edge of a Hall sensor input, instead of just the rising edge of Hall Sensor A. This improves the granularity of the speed calculation, and allows the PI loop to respond to changes in speed more quickly.

6.23 Bug Fixes in RDK-IDM-SBC Firmware Package

6.23.1 Corrected operation of Lock and Unlock macros for Slider and JPG-Widget (Reference 9471)

In previous releases, the operation of the SliderLock()/SliderUnlock() and JPEGWidget-Lock()/JPEGWidgetUnlock() macros were reversed. This has now been corrected.

6.23.2 Web server opens Luminary Micro site in the wrong frame (Reference 9488)

In various applications supporting an embedded web server (depending upon the kit, enet_io, enet_lwip, qs-checkout and idm-checkout) used to open the link to

`http://www.luminarymicro.com` within a frame. The sites have been updated to open this link in the top level window instead.

6.24 Bug Fixes in RDK-S2E Firmware Package

6.24.1 Change between static and DHCP IP sometimes fails (Reference 9438)

The function, lwIPNetworkConfigChange, does not always switch properly between static IP and Auto IP (with DHCP). This is due to the fact that the variable that retains the current IP mode setting is not properly saved. This variable, g_ulIPMode, is now saved at the end of the function for all cases.

6.25 New Features in Stellaris Firmware Development Package

6.25.1 Add SourceryG++ for Stellaris project files (Reference 9469)

Project files (.sgxx) and workspace files (.sgxw) for use by the SourceryG++ for Stellaris IDE are now provided for the libraries, applications, and boards that are provided in StellarisWare.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TIŠs terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TIŠs standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or “enhanced plastic.” Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer’s risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265