Pluto 6 Release Notes

Document No. 80-17739   Issue 15

Current Issue :-   19/06/09

Previous Issues :-   14

If your query is not covered in this Manual, or you require further information, please email Heber Customer Support: support@heber.co.uk

The latest version of this document and other technical information can be found on the Heber website: www.heber.co.uk

The information contained in this document is believed to be accurate and reliable. However, Heber Ltd. assumes no responsibility for its use, and reserves the right to revise the documentation without notice.

Precise specifications may change without prior notice.

HEBER LIMITED
Belvedere Mill
Chalford Stroud
GL6 8NT
UK

Tel +44 (0) 1453 886000
Fax +44 (0) 1453 885013
Email   support@heber.co.uk
Website   www.heber.co.uk

CONTENTS

This page intentionally left blank.

1 INTRODUCTION

The software that is installed with the Pluto 6 Development Kit is the current version at the time that the product was released. Periodically the Heber Pluto 6 software will be revised. The updated software can be downloaded from the Heber website: www.heber.co.uk

It is recommended that you check the website and download the latest versions of the software drivers before starting game development.

If you experience problems with the software during game development please check the website to ensure that you are using the latest versions of the software.

This document provides information regarding changes made for each version of the software. This document is updated after each release with a brief description of the problem that was identified. From time to time, when a better description is required Heber will include special notes on the problem, side effects or code description.

For any further information or to report any problems with the software contact Heber Customer Support: support@heber.co.uk

2 HARDWARE LIBRARY (81-16616)

2.1 Issue 9 (27/01/2004)

First Release

2.2 Issue 10 (26/02/2004)

Video driver:

Internal changes to SaveBackground function to work with 8 or 16 bit screen width.

Security feature:

Security files added to provide security functionality to Pluto 6 hardware library.
Function: UnlockSecurity()

2.3 Issue 10r1 (09/03/2004)

Mbus driver:

Mbus read and write functions can now be used to access the E2 device on the Pluto 6 board. This is a temporary fix until an E2 driver has been released.

2.4 Issue 10r2 (16/04/2004)

Video driver:

Calypso 32: Timing fix to improve Cremson to video memory bandwidth. The fix involves modifying register settings to optimise performance with ISSI 42S16400 SDRAMs. For more information please refer to:

> *Pluto 6 Technical Bulletin: Pluto 6/Calypso 32 Graphics Controller Limitations (80-17644-1).*

2.5 Issue 10r3 (25/05/2004)

Video driver:

1) Bug fix in SaveBackground function to work properly with 16bit sprites.

2) In LoadSprite function, the mask created is only for an 8bit colour sprite
One parameter was added to reflect the colour depth of the image and create a correct mask for 8bit or 16bit sprites according to this parameter.

3) As the video driver is currently provided, users need to create their own mask for their picture if they want to use transparency features with PutSprite function.
An API function, CreateMask, was implemented for all users to use the same function to create a mask. To use a mask properly with PutSprite, it should be created after the image in the video memory.

LONG CreateMask(const UBYTE* startaddr, LONG offset, LONG size ,LONG bytes_per_pixel)
Where the parameters are:

1) Address of sprite image
2) Offset from start of video memory to beginning of mask
3) Size of sprite image to mask in bytes
4) 16 or 8 bit colour

And the return value is the Offset from start of video memory to end of mask

2.6 Issue 10r4 (11/06/2004)

DUART driver:

Functions RxAEmpty() and RxBEmpty were using the transmit buffer pointer instead of the receive buffer pointers.

Video driver:
BltBusy() function now correctly returns “1” while the bit block transfer is in progress.

2.7 Issue 10r5 (15/06/2004)

Internal Heber release.

2.8 Issue 10r6 (01/07/2004)

General comments:
This release is a very important one as it involves modification to interrupt priority levels.
Anyone who updates their version of the hardware library to 10r6 or above needs to ensure that:

1. fpga.c and fpga.h files from current project folder are identical to the one included in project
    a. demo version 6

2. lnkdram.x file from current project folder is identical to the one included in project
    a. demo version 6

3. Modify your project makefile to target correct libraries

```
lhdw=  ../../libs/hardware/$(target)/81-16616-10r6
ihdw=  ../../libs/hardware/$(target)/81-16616-10r6
```

Video Driver:
Function BltBusy, in some occasions doesn’t return. Function was amended to return all the time. (Method: check display list continuously to see when display list gets emptied).

Internal driver:
Earlier versions of the hardware library were set to have the 10 ms interrupt on a higher level than the video interrupt. In this case, when playing sound in a double buffered loop, a video flicker is created.
Modifications involve setting the 10ms interrupt and the DUART interrupt at the same level.

To achieve these changes, the vector table was modified. It is mandatory to update file “fpga.c” to a newer version (as described above). Not modifying this file will lead in immediate failure of the board when trying to run your game.

2.9 Issue 10r8 (05/10/2004)

ATA Driver:
Because of the variety of Compact Flash cards on the market, a modification to the driver was required to avoid delay in data transfer caused by some cards. Even though this change has improved the driver for some CF cards, it is still highly recommended to test any brand of CF card before going to production of a game.

2.10 Issue 11 (12/11/2004)

Issue 11 is a major release for Issue 2 of the Pluto 6 Development Kit.

Advanced Video Driver:
The Calypso32 Advanced Video Driver is included in this issue of the library. For information on the driver and API functions please refer to Issue 3 of the Pluto 6 Software User Manual (80-17314).

2.11 Issue 11r1 (19/11/2004)

Multiplexed inputs driver:
When the TestDir function is called with an active high input (e.g. TestDir(IP0 | ACT1)), true is always returned. The function was amended to return the correct reading in this case.

2.12 Issue 14r1 (31/5/2005)

ATA Driver:
Compact Flash write capability has been added to the ATA driver. The following functions have been added:

ATASTS WriteSector(const DEVICE* dev, ULONG lba, UWORD *buffer)

Where dev is a pointer to ATADevice, lba is the logical block address to write to on the Compact Flash card, and buffer is a pointer to the data to write. The function returns ATA_OK on success or BSY_TIMEOUT on failure. The function is intended to be used by the function FilePutSector() in the updated FAT32 driver of the interface library.

2.13 Issue 15r3 (27/9/2005)

Dual Video Driver:
The Pluto 6 Dual Video Driver software extends the capabilities of the Pluto 6 Software by providing support for Dual Video systems within the hardware library.

Pluto 6 boards with the part number 01-17802 have dual video capability and can be fitted with two Calypso 32 video boards.

Software support for dual video operation has been added to the existing Advanced Calypso Driver where all the API functions include a device parameter. This identifies the video board to which the function refers:

Calypso32SomeFunction(const DEVICE *dev, parameter list...)

Where dev is one of the following:

&Calypso32Plus Device Single video system
&Calypso32Dual0 Device Dual video system: First board
&Calypso32Dual1 Device Dual video system: Second board

Please refer to the Pluto 6 Software Manual (80-17314 issue 7 for more details).

Dual video is handled by a *separate vertical interrupt handler function* for each board. The appropriate handlers are assigned at driver installation using the function pointers passed via the configuration structures. As a result, every customer using the hardware library post 15r3 MUST update their current projects according to the details in section 5.5 of the current document (Mandatory update for all projects using hardware library post 15r3)

List of devices:
In order to use the new HSF driver, a device was added in the hardware library.

2.14 Issue 16 (18/11/2005)

Fpga Uart Driver:
Bug fix with handshake signals – Creation of three functions (statusfpuartCTS - clearfpuartRTS - setfpuartRTS) to handle these signals.

RS485 Driver:
New driver written to enable or disable RS485 communication (EnableRS485Tx - DisableRS485Tx)

Lamp Driver:
Internal modifications to some functions. These changes should not affect any customer. Please contact Heber Customer Support (support@heber.co.uk) if you are experiencing new problems with this driver.

2.15 Issue 16r1 (09/12/2005)

Lamp Driver:
The functions deleted in issue 16 were restored since some customers are using the deleted functions.

2.16 Issue 16r2 (06/02/2006)

PIC18:
Bug Fix: The index provided to GetPIC18Log function was causing problems. A test within the function was added and reports an error in case a wrong index is entered.
The documentation was modified to reflect changes.

2.17 Issue 16r3 (10/02/2006)

Lamp Driver:
A flicker would occur when multiple fast call to the Lamp() function is made. For example:

```
while(1)
{
Lamp(LAMP(LC1,LR0),ON);
}
```

The order of operation within the driver was modified (segment not turned off unnecessarily) in order to remove the flicker.

2.18 Issue 17 (28/02/2006)

Advanced video driver:
Advanced line functionalities (using the Cremson registers) added (see documentation)

CalypsoAdvanceLine – CalypsoSetLineProperties – CalypsoAdvanceRectangle

2.19 Issue 17r1 (25/04/2006)

Advanced video driver:
Bug fix: Function PutUnicodeStr.

2.20 Issue 17r2 (18/07/2006)

Duart driver:
Bug fix: General Error flag not cleared correctly. This would be noticed in case of an overrun error.

2.21 Issue 18 (22/11/2006)

HardReset function updated in order to write to the correct FPGA register. Now, a direct call to the HardReset function within our libraries will correctly reset the platform.

2.22 Issue 19 (20/02/2007)

Advanced video driver:
Advanced line functions to draw lines without anti-aliasing.

CalypsoAdvanceSetLineAntiAlias - CalypsoAdvanceClearLineAntiAlias

2.23 Issue 20 (25/04/2007)

Device Manager:
New devices have been defined in the device list for ccTalk Device Drivers.

```
#define CCTALKDEVICE  0x21
#define CCTALKDRIVER  0x01
#define CCTALKBILL    0x02
#define CCTALKCOIN    0x03
#define CCTALKHOPPER 0x04
```

2.24 Issue 21 (01/11/2007)

Support for Enhanced 2 stage security:
Functionality has been added to support an enhanced version of the Pluto 6 security. This functionality requires an Enhanced Security CSD.

Security Driver -
A new security function has been added that takes a single start and end address. This function is used for second stage security when using enhanced security.

Entry Code
First stage security code has been added to the entry code. A global flag has been added to enable users to disable or enable this functionality. The flag should be set as follows:

1) Enabled - this is for enhanced security boot loaders.
2) Disabled - this is for project code and standard security boot loaders.

This implementation allows a single version of the hardware library to be used for all projects and boot loaders, and to allow backwards compatibility with standard security projects.

NOTE: any projects built against issue 21 and above of the hardware library need to set the global flag to an appropriate value for the type of project.

The flag should be included in the project file config.c as follows:

```
        const ULONG SecurityFlag = SECURITY_ENABLED;
```

and set to the appropriate value for the type project.

The following defined values need to set in the file config.h at project level:

```
/* boot security */
#define SECURITY_ENABLED 0x00000001
#define SECURITY_DISABLED 0x00000000
```

The following line needs to be added to the file fpga.c at project level:

```
BYTE*const  Xsec=CRCCHECK;
```

2.25 Issue 22 (04/01/2008)

PIC driver:
A finer level of granularity has been added to the PIC real time clock (RTC). 1/100ths of a second as well as seconds since the Pluto 6 epoch may now be retrieved.

A new function GetPIC18TimeFine has been added to the PIC driver which accesses the fine grained RTC to retrieve an extended time value from the PIC. When the hundredths are retrieved the value is populated in the P18TIME structure.

All existing PIC 18 API functions are still supported with CSD designs that implement the finer granularity RTC.

2.26 Issue 23 (09/02/2009)

PIC driver:
RTC bug fixed that caused date corruption during leap year role over.

2.27 Issue 25 (21/05/2009)

DUART driver:
Support added for the following Baud rates:

- 57600
- 115200

Please download version 13 of the Software User Manual 80-17314 and refer to the Serial Defined Values page.

3 INTERFACE LIBRARY (81-17292)

3.1 Issue 4 (27/01/2004)

First Release.

3.2 Issue 4r2 (05/10/2004)

Sound driver:

Some issues have been identified and solved with this version of the interface layer:

- A distortion was audible when mixing audio samples.
- A distortion was audible when stopping and restarting a sound sample continuously.
- Random reduced speed playback.
- QuietChannel stops any currently playing WAV files on the specified audio channel as expected but also closes the sound file. This behaviour is not expected and time consuming from a game developer's perspective so an extra function was created:

  ```
  void StopAudioChannel( BYTE audioChannel )
  ```

  This function silences any audio currently playing on the requested channel. The differences to QuietChannel() are:
  - does NOT close file
  - rewinds file to start

Some customers have reported in some occasions a non-existing fault with the sound driver. This fault is often described as a clicking noise at the end of the sample. As described in the Pluto 6 software documentation, a period of silence at the end of a sound sample, or fading the sample, is required with the Pluto 6 board.

When a sample stops playing, the audio amplifier will return to a 0 volt crossing. When a sample is played right after, the voltage will then rise. If a sample is stopped at a voltage > 0 volts (i.e. not silence) the audio amp needs time to drop to 0 volts before a new sample is played. If this time is not present the drop to 0 volts and subsequent rise in voltage when the new sample is played will result in an audible click.

3.3 Issue 5 (12/11/2004)

Issue 5 is a major release for Issue 2 of the Pluto 6 Development Kit.

FAT32 driver:

Improvements have been made to the driver to reduce the amount of data placed on the stack during directory searching operations.

Sound driver:

The sound driver has been modified to recognise .wav audio files that are encoded using both A-Law and U-Law compression. Using U-Law or A-Law compressed audio reduces the time taken to load files from the Compact Flash card. Using the .wav file header information, the driver detects if compression has been applied. For further information please refer to Issue 3 of the Pluto 6 Software User Manual (80-17314).

3.4 Issue 5r1 (14/1/2005)

Windows Bitmap Loader: Fix for problem of incorrectly transparent pixels

The bitmap loader truncates 24 bit colour values to 15 bit. An unwanted side-effect is the conversion of near black pixels to black pixels (r=g=b=0), which are transparent.

The fix prevents conversion of near-black pixels to black pixels.

3.5 Issue 6 (28/6/2005)

FAT32 driver

The driver has been modified to take advantage of the Compact Flash writing capability added to the ATA driver in the hardware library. Writing to files via the use of file descriptor structures is made possible by the modification of the following functions:

```
FILE_DESCRIPTOR *FileOpen(const DEVICE* dev, BYTE *filename, FILE_DESCRIPTOR *file,
                                        UBYTE mode)
```

This function is used as in the documentation, except the following modes are now available:

```
READ_ONLY
READ_OVERWRITE
OVERWRITE_ONLY
```

The following functions have been added:

```
ULONG FilePutSector(const DEVICE* dev, FILE_DESCRIPTOR *file, UWORD *mem)
```

This function writes one sector of data to a file, where dev is a pointer to ATADevice, file is a pointer to a FILE_DESCRIPTOR structure and mem is a pointer to the data to write to the file. The FILE_DESCRIPTOR structure file must have been initialised by calling FileOpen using either READ_OVERWRITE or OVERWRITE_ONLY mode.

Only block (sector) oriented writing is supported. In order to improve block oriented control over file access the following functions have been added:

```
ULONG FileSeekSector(const DEVICE* dev, FILE_DESCRIPTOR *file, ULONG sector_offset)
```

This function moves the current file read/write position, where dev is a pointer to ATADevice, file is a pointer to a FILE_DESCRIPTOR structure and sector_offset is the offset in sectors from the start of the file. Zero is returned on success.

```
ULONG FileCurrentSector(const DEVICE* dev, FILE_DESCRIPTOR *file)
```

This function returns the current read/write position as an offset in sectors from the start of the file, where dev is a pointer to ATADevice and file is a pointer to a FILE_DESCRIPTOR structure.

Sound driver

The sound driver has been modified to increase reliability. Processing of sound samples is now spread across several 10ms interrupt service routines using a state machine. The states are based around which channel to process in the current audio interrupt service routine.

Two new API functions have been added which use FILE_DESCRIPTOR structures directly rather than WAVFILEINFO structures, which reduces RAM usage when handling multiple sound samples.

```
BYTE SearchForWavFile( BYTE *filename, FILE_DESCRIPTOR *file )
```

This function is equivalent to InitWavFile(), except instead of passing a pointer to a WAVFILEINFO structure, use a pointer to a FILE_DESCRIPTOR instead.

```
BYTE PlaybackWavFile( BYTE channel, FILE_DESCRIPTOR *file)
```

Again this function is equivalent to PlaybackCachedWavFile(), except instead of passing a pointer to a WAVFILEINFO structure, use a pointer to a FILE_DESCRIPTOR instead.

There are a set of guidelines for the usage of the new sound driver which should be followed for best results. See *Pluto 6 Technical Bulletin – Streamed Sound Driver Recommended Usage (80-18329)*.

3.6 Issue 6r4 (27/9/2005)

Heber sound format driver (HSF audio driver):
The Heber Sound File audio driver is an alternative streaming sound driver, which provides the capability to stream up to 4 channels of audio from a Compact Flash card.

The new driver supports a proprietary Heber audio format, which has the following features:

The Heber Sound Files are pre-processed - header information has been reduced to a single length field.
The audio data is compressed using CCITT-ULAW.
The files are a multiple of 512 bytes in size, to ensure they do not cross sector boundaries on the Compact Flash card (this reduces file reading complexity within the FAT32 driver).
Stereo samples are stored in separate HSF files.
The reason for these characteristics is to reduce the complexity of the driver, and hence reduce CPU load/file accesses within interrupt routines. The simplicity of this driver will also increase reliability.

This new driver can only be used with a hardware library post 15r3. Please refer to the Pluto 6 Software Manual (80-17314 issue 7) for more details on how to use this new driver.

3.7 Issue 7 (18/11/2005)

Bitmap loader driver:
Three functions added to library (EnableAlphaBlending, DisableAlphaBlending, StatusAlphaBlending) in order to load a bitmap image from Compact Flash with or without the Alpha Blending Bit set.

3.8 Issue 7r1 (05/12/2005)

HSF driver:
Bug Fix: Zeroing DMA buffers within the DMA interrupt causes the DMA interrupt to block lower level interrupt subroutines.. (lamp flicker)

3.9 Issue 7r2 (13/02/2006)

Heber internal release

3.10 Issue 8 (21/11/2006)

HSF driver:
Bug Fix: The driver was reported to lose the sound on some occasions. After investigation, it was proved that under certain circumstances, the DMA interrupt will return earlier and cause the state table to transit unconventionally. A set of semaphores was added within the driver and the issue is now resolved.

3.11 Issue 8r1 (28/02/2007)

HSF driver:
Bug Fix: The driver was reported to cause sound “clicks” and “pops” at the end of the HSF file when played. After investigation, it was proved when a sound file of a specific file length is used it is possible for data to be read beyond the end of the file. This extra data would then
be decompressed to the DMA buffer as non-silence, causing an audible "Pop", "Blip" or "Click". A test was added within the driver to check that the correct length of data is read and the issue is now resolved.

3.12 Issue 9 (15/08/2007)

ccTalk driver:
Initial implementation of the ccTalk interface library including support for unencrypted coin and bill acceptors.

3.13 Issue 9r1 (09/02/2009)

ccTalk driver:
Bug Fix: There were a few compilation issues when certain ccTalk commands were included. These issues have now been resolved in this release.

Rebuilt against issue 12 Peripherals Library and Issue 23 Hardware Library.

3.14 Issue 9r2 (18/06/2009)

Rebuilt against issue 13 Peripherals Library and Issue 25 Hardware Library.

4 PERIPHERALS LIBRARY (81-16631)

4.1 Issue 7 (27/01/2004)

First Release

4.2 Issue 8 (12/11/2004)

Issue 8 is a major release for Issue 2 of the Pluto 6 Development Kit.

E2ROM Driver:

A driver has been added to the peripherals library that provides functions to access the 512 bytes E2PROM device that can be fitted into socket U7 on the Pluto 6 board. For information on the driver and API functions please refer to Issue 3 of the Pluto 6 Software User Manual (80-17314).

4.3 Issue 9 (28/6/2005)

This issue of the library has been built against the latest release 14r1 of the hardware library. The latest release 6 of the interface library has dependencies on this issue number of the peripherals library. Therefore this release should be used with release issue 14r1 of the hardware library and release issue 6 of the interface library.

4.4 Issue 9r1 (27/9/2005)

This issue of the library has been built against the latest releases of the hardware (dual video) and interface (HSF driver) libraries to ensure a correct dependency in between all Heber's libraries.

4.5 Issue 10 (27/9/2005)

Touch Screen Driver:

Fix to handle problem with some ELO touch screens. These sometimes transmit incomplete touch packets when in streaming mode, causing errors in reported screen XY co-ordinates.

4.6 Issue 11 (15/8/2007)

ccTalk Driver:

An initial release of the ccTalk peripheral library including support for coin and bill acceptors.

4.7 Issue 12 (09/02/2009)

Peripherals Library rebuilt against issue 23 of the Hardware Library.

4.8 Issue 13 (18/06/2009)

Rebuilt against issue 25 of the Hardware Library.

5 UPDATING PLUTO 6 LIBRARIES

After downloading the latest library, copy the content of the zip file to[1]

C:\heber\pluto6\libs\hardware\pluto6\81-16616-XX
Or
C:\heber\pluto6\libs\interface\pluto6\81-17292-XX
Or
C:\heber\pluto6\libs\peripherals\pluto6\81-16631-XX
Or
C:\heber\pluto6\libs\customperipherals\pluto6\81-17315-XX

Where XX the current issue of the library.

Update your project makefile to target the correct libraries as follows:

```
lhdw=   ../../libs/hardware/$(target)/81-166616-XX
lper=   ../../libs/peripherals/$(target)/81-16631-XX
lint=   ../../libs/interface/$(target)/81-17292-XX
lcper=  ../../libs/customperipherals/$(target)/81-17315-XX
ihdw=   ../../libs/hardware/$(target)/81-16616-XX
iper=   ../../libs/peripherals/$(target)/81-16631-XX
iint=   ../../libs/interface/$(target)/81-17292-XX
icper= ../../libs/customperipherals/$(target)/81-17315-XX
```

Ensure the linker script lnkdram.x is targeting the latest hardware library

```
vectors_start    = .;
../../libs/hardware/pluto6/81-16616-XX/vectors.o(.text)
vectors_end      = .;
```

Open a DOS prompt where your project lives and type the following commands:

"touch *c" to update all the dependencies
"make" to remake your current project using the latest libraries

[1] The following explanation assumes the development kit was installed in C:\heber\

6 PROJECTS

6.1 Introduction:

Pluto 6 Development Kit was released on the 27/01/2004.

At this time, the versions of the libraries were:

1. Hardware: v9
2. Interface: v4
3. Peripherals: v7

The versions of the three projects provided with the development kit were:

1. Bootloader: v1
2. Demo: v4
3. Video: v1

6.2 First update (17/07/2004):

6.2.1 Reason for changes:

For the last six months, some of the technical support queries have involved important aspects of the Pluto 6 board such as:

- Some modifications in order to achieve security settings
- Data backed up in the sysram or idram area being erased on startup
- Problems securing a large area of data in memory
- Problems working at high video resolution
- Problems while using video and sounds.

Most of the faults have been identified and successfully fixed by up-issuing the hardware library.

Some fixes could only be implemented by modifying some important parameters directly into the project folders. On the 19/07/2004, all projects are re-issued and can be downloaded directly from the Heber website.
The versions of the three projects are currently:

1. Bootloader: v3
2. Demo: v6
3. Video: v2

It is mandatory before continuing development to carry out the following:

1. When the development kit was released, it came with a pre-programmed Compact Flash card. The Compact Flash cards contains by default 2 files (video.s19, demo.s19) and 2 folders (images – audio). Modify your Compact Flash card with latest “s19 files” coming from video project and demo project.
2. Reprogram your flash memory (U1) with the correct bootloader project. For that you will need an EPROM programmer.
3. Amend your current projects with following files from the demo project:
   a. fpga.c and fpga.h
   b. lnkdram.x
4. Amend your makefile to target latest libraries.

6.2.2 Explanation of changes:

Sysram or idram erased on startup:

The memory map defined by the linker file in the bootloader project and any other (video – demo) contains an overlapping area of memory where the idram and sysram should be backed up. Each time the boot loader starts, it erases the content of this common area in memory.
An area called keepram was created at a different address range for each project. Each project now backs up the sysram and idram into its own keepram area. Data stored in these areas is no longer overwritten on system start up.

Security settings and secure area.

Fpga.h and fpga.c files were modified to use security settings from our libraries. The refresh rate of the idram was also modified in order to secure a larger area of memory if necessary.

Interrupt levels:

Fpga.c was modified to set the interrupt priority of the 10ms interrupt at the same level as the DUART priority. This amendment stops video flicker when sound is played at the same time as video double buffering.

6.3 New Issues of projects (12/11/2004):

6.3.1 Existing projects:

The previously released demo and video projects have been re-built against the latest issues of the library, and re-issued.

Demo Project 81-17269-7:
Tests have been added to the demo project to test the E2ROM driver.
The interrupt mask values in the file config.c have been updated to reflect changes in the driver code.

Video Project 81-17456-3:
The interrupt mask values in the file config.c have been updated to reflect changes in the driver code.

Boot Loader Project 81-17443-5:
The interrupt mask values in the file config.c have been updated to reflect changes in the driver code.

6.3.2 New projects:

Pluto 6 demo project 2 81-17951-3:
A new project has been added which demonstrates the use of the Advanced Video Driver and Audio API functions.

6.4 Recommended update for all projects (24/1/2005)

All projects need to be modified in order to ensure that the dram refresh rates are correct. The DRAM configuration structure in the project fpga.c file should contain the following:

```
/* DRAM Controller Module */
const DRAMCFG dram_cfg=
{
        0x0025,         /* DCRR - DRAM CONTROLLER REFRESH REGISTER
                           The DCRR controls the frequency of refresh requests.
```

6.5 New Issues of projects (09/02/2009):

6.5.1 Existing projects:

The projects listed below have been re-built against the following issues of the libraries:

6.5.1.1 Libraries

Hardware Library 81-16616-23
Peripherals Library 81-16631-12
Custumperipherals Library 81-17315-6
Interface Library 81-17292-9r1

6.5.1.2 Projects

81-17269-9 -Development Kit Standard Demo Project
Tests have been added to the demo project to test the fine granularity RTC.

81-183753-4 -Development Kit Dual Video Demo

81-17443-7 - Standard Boot Loader Project

81-18328-3- Touchscreen Project

81-17951-4 - Video Audio Project

6.6 New Issues of projects (18/06/2009):

6.6.1 Existing projects:

The projects listed below have been re-built against the following issues of the libraries:

6.6.1.1 Libraries

Hardware Library 81-16616-25
Peripherals Library 81-16631-13
Custumperipherals Library 81-17315-8
Interface Library 81-17292-9r2

6.6.1.2 Projects

81-17269-10 -Development Kit Standard Demo Project

81-183753-5 -Development Kit Dual Video Demo

81-17443-8- Standard Boot Loader Project

81-18328-4 Touchscreen Project

81-17951-5 - Video Audio Project