

# MS7724 Linux Golden Package BSP

RENESAS MPU  
SH4 Family / SH7724

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

### 1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between Products

Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

# How to Use This Manual

## 1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the MS7724 Linux Golden Package BSP. It is intended for users creating products using Gnu/Linux with the SH7724 processor

The following documents apply to the BSP.

<b>Document Type</b>	<b>Description</b>	<b>Document Title</b>	<b>Document No.</b>
User's Manual	Describes the technical details of the Linux BSP.	MS7724 Linux Golden Package BSP User's Manual	D009674_11
Quick Start Guide	Provides simple instructions to setup and run the demo provided on the MS7724 board.	MS7724 Linux Golden Package BSP Quick Start Guide	D009672_11
Host Setup Guide	Provides information on setting up your Linux host PC on order to use the BSP.	MS7724 Linux Golden Package BSP Host Setup Guide	D009673_11

## 2. List of Abbreviations and Acronyms

Abbreviation	Full Form
ALSA	Advanced Linux Sound Architecture
bps	bits per second
BSP	Board Support Package
CEU	Capture Engine Unit (Renesas hardware block)
DVI	Digital Visual Interface
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
FAAC	Freeware Advanced Audio Coder
FAAD	Freeware Advanced Audio Decoder
FTP	File Transfer Protocol
IIC	Inter-Integrated Circuit (I <sup>2</sup> C)
IP	Internet Protocol or Intellectual Property
LAN	Local Area Network
LCDC	LCD Controller (Renesas hardware block)
RTP	Real-Time Protocol
TFTP	Trivial FTP
UIO	User-space I/O
VEU	Video Engine Unit (Renesas hardware block)
VPU	Video Processing Unit (Renesas hardware block)

# Table of Contents

1. Overview.....	6
1.1 Introduction.....	6
1.2 Software.....	6
1.3 Hardware Support.....	7
1.4 Renesas Specific Software.....	8
1.4.1 Video Decoder Middleware.....	8
1.4.2 SH GStreamer plugins.....	8
1.5 Open Source Software.....	8
2. Installation.....	9
3. Building the BSP.....	10
3.1 Re-building.....	10
3.2 Building Qt demos.....	10
4. Using the BSP.....	11
4.1 U-Boot Boot Loader.....	11
4.2 Examples.....	12
4.2.1 Touch Screen.....	12
4.2.2 Video Codecs.....	12
4.2.3 Streaming Video.....	12
4.2.4 DirectFB.....	12
4.2.5 GStreamer.....	13
4.2.6 Qt Demo.....	13
5. Supplementary Information.....	14
5.1 Memory allocation.....	14
5.1.1 Camera Capture.....	14
5.1.2 Video Decode & Encode.....	14
5.2 U-Boot.....	14
5.2.1 Dynamic IP address.....	15
5.2.2 Bootp.....	15
5.2.3 Modifying U-Boot.....	15
5.2.4 Placing the kernel in flash memory.....	16
5.2.5 USB Root File System.....	16

# 1.Overview

## 1.1 Introduction

The BSP provides software for multimedia products using the SH-Mobile R2R (SH7724) device. The BSP is to be used on a linux host PC (see the Host Setup Guide) with the MS7724 development board. The BSP requires internet access to download some of the software packages. The BSP consists of:

- Root file system
- Linux kernel
- General open source software packages
- SH-Mobile specific software packages

## 1.2 Software

Linux systems can use a wide range of software, far too wide to cover here. The Renesas Linux BSP focuses on the multi-media capabilities of the SH-Mobile devices; the following two diagrams show a typical set up for video player and video recorder applications respectively.

The diagrams show the use of GStreamer for the multi-media framework. In some products this is not necessary, howe there are a significant number of open source GStreamer plugins that provide functionality such as audio codecs, container multiplexing & de-multiplexing, RTP streaming, and audio/video synchronization.

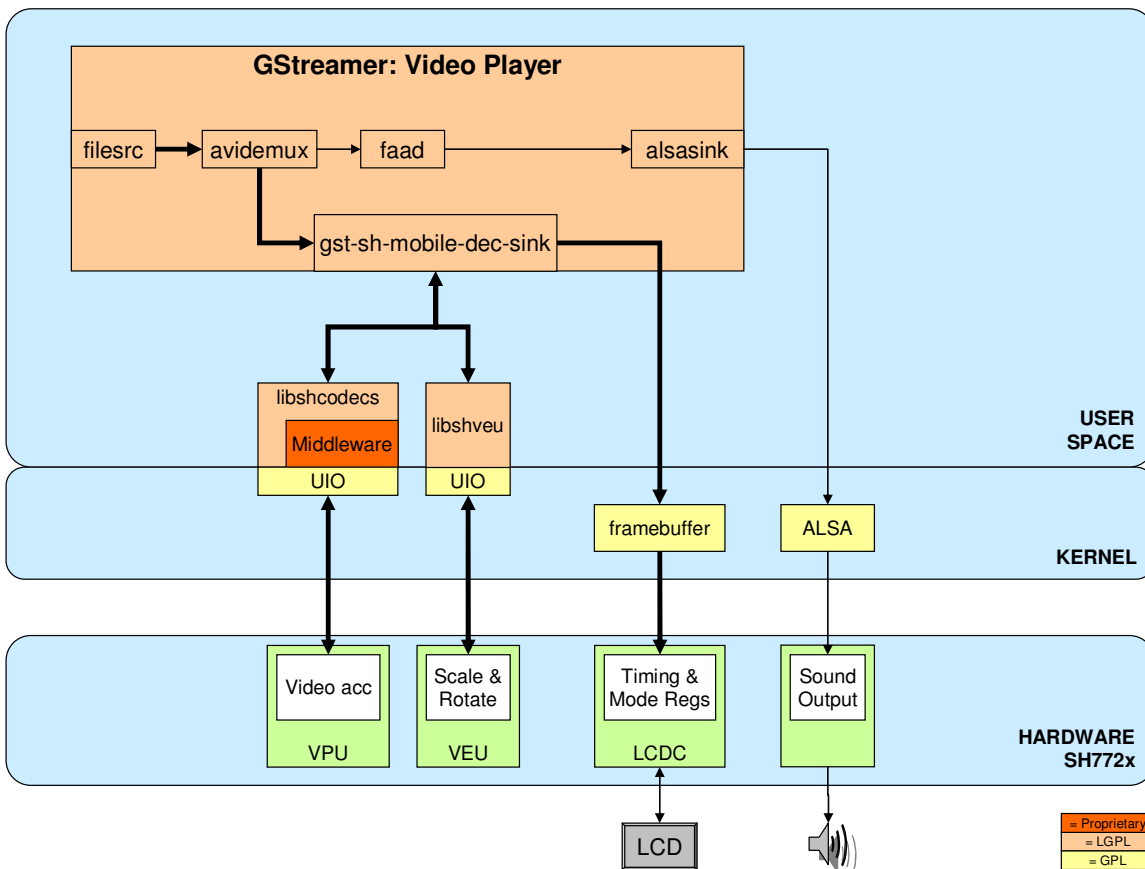


Figure 1 : Video Player Overview

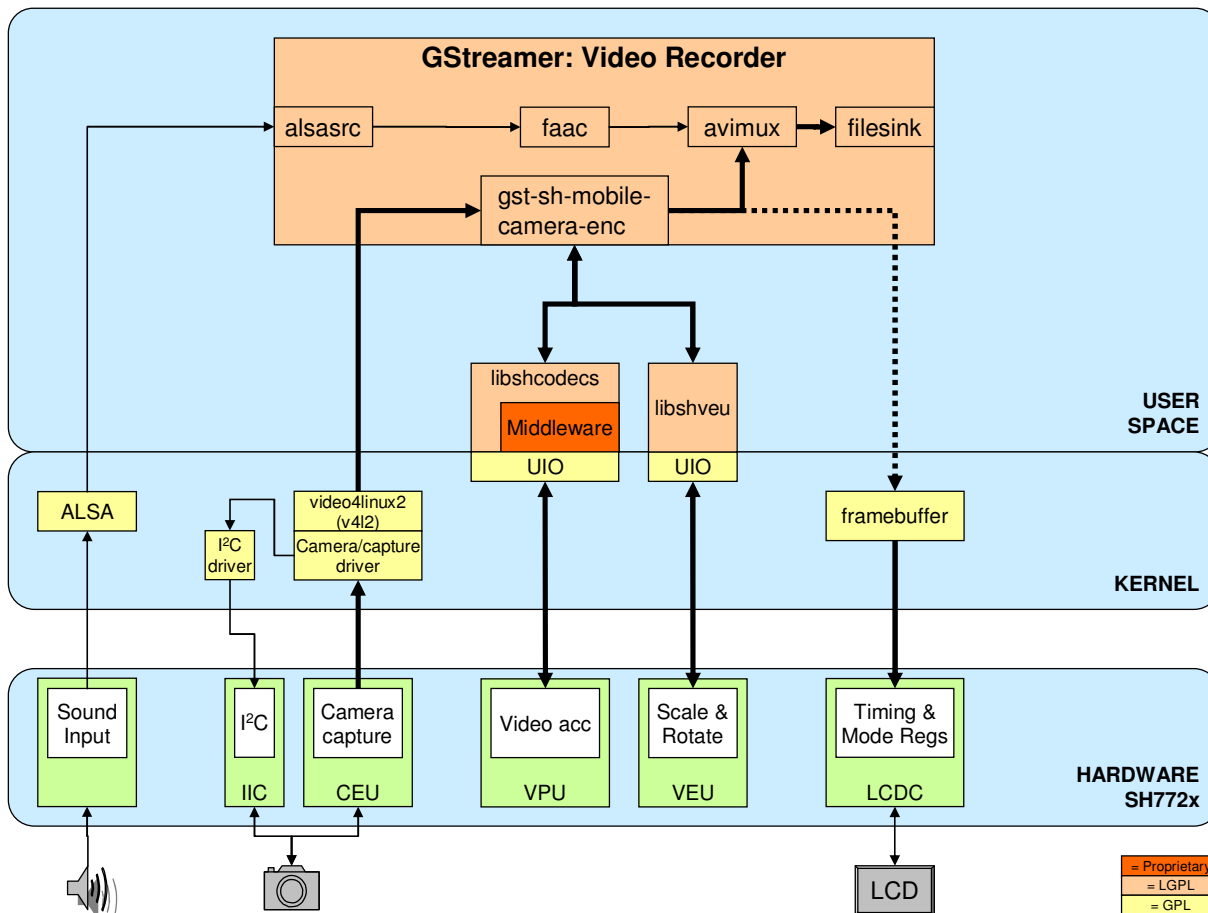


Figure 2 : Video Recorder Overview

### 1.3 Hardware Support

The following hardware has software support:

- USB x2 (HOST)
- Serial port (SCIF0)
- I2C x2
- LCD & DVI (LCDC)
- Ethernet
- Video input (tw9910)
- Camera capture (CEU x2)
- Sound I/O (FSI)
- Touch Screen
- Aptina camera module (mt9t112 x2)
- Audio Codec (DA7210)

## 1.4 Renesas Specific Software

Shown below is a list of Renesas specific software provided in the BSP.

<u>Software</u>	<u>Description</u>	<u>Source</u>
Accelerated Video decode/encode	Proprietary middleware provided by Renesas.	Please contact your Renesas representative if not on CD
	Open source libraries (libshcodecs)	<a href="http://github.com/pedwo/libshcodecs">http://github.com/pedwo/libshcodecs</a>
	Open source GStreamer plugins	<a href="http://github.com/pedwo/gst-sh-mobile">http://github.com/pedwo/gst-sh-mobile</a>
Accelerated Video scale/rotate	Open source libraries (libshveu)	<a href="http://github.com/pedwo/libshveu">http://github.com/pedwo/libshveu</a>
Accelerated Video blend/overlay	Open source libraries (libshbeu)	<a href="http://github.com/pedwo/libshbeu">http://github.com/pedwo/libshbeu</a>
Conflict manager for system resources, including UIO devices	Open source libraries (libuio mux)	<a href="http://github.com/pedwo/libuio mux">http://github.com/pedwo/libuio mux</a>

### 1.4.1 Video Decoder Middleware

For video streams that have consecutive I-frames, the video decoder middleware will output decoded frames to the same output buffer as the previous I-frame. Since libshcodecs allows clients to read one output frame whilst allowing the VPU to write the next frame, this will result in corrupted frames. Streams with consecutive I-frames are an highly unusual situation.

### 1.4.2 SH GStreamer plugins

SH specific video decode and encode plugins are provided to utilize the VEU and VPU hardware. For performance reasons, gst-sh-mobile contains additional GStreamer plugins to perform camera capture with encode, and video decode with output to the framebuffer. With these plugins it is not possible to use other GStreamer plugins to modify the video data prior to encode. Note that there is a SH video decoder and SH framebuffer sink plugin, but these plugins must be used together.

## 1.5 Open Source Software

A number of open source packages are included in the BSP. The choice of these packages is based on creating a GStreamer multimedia solution. The main packages are:

Touch screen	tslib	
Audio	ALSA	Advanced Linux Sound
	MAD	MP3 Decoder
	ID3TAG	MP3 ID3 tags
	FAAD	AAC Decoder
	FAAC	AAC Encoder
Others	GStreamer (with good, bad & ugly plugins)	Multimedia framework

Other packages are installed as required to meet the dependencies of the above packages.



## 2.Installation

Make sure you have your host PC set up with the relevant packages and services, see the Host Setup Guide.

The BSP is installed using two installers, one for the main features and an additional installer for licensed codec middleware. If the BSP has been downloaded, the middleware installer may not be included with the BSP. Please contact your local Renesas representative.

The BSP can be used without the middleware, though some features such as accelerated video decoding will not be supported. The installers install to the current directory, so ensure that you run the installers from the directory that you wish to install to (note that you can't use a path with a space in it). Assuming that you are installing from CD, mount the CD then run the installers:

```
/media/cd/bsp/sh772x-linux-installer.bin  
/media/cd/bsp/sh772x-linux-shcodecs.bin
```

where “/media/cd” is the mount point of the CD. The installer may take a while to complete.

The installers will create the following directory structure:

installer	Scripts to download & build the target software
installer/buildroot_dl	Buildroot packages
installer/demo	Demonstration application
shcodecs	Middleware binaries

By default the scripts will create a directory the same level for building source code:

```
src
```

## 3. Building the BSP

The scripts provided can be used to do everything for you. To create everything:

- Change to the install script directory

```
cd installer
```

- If you wish to change the directories used for extracted source code and TFTP dir, modify the `BUILDROOT_BASE` and `TFTP_ROOTFS` variables in the `defines.sh` file.
- Extract the toochain and then extract & build the packages by running

```
./build_all.sh
```

The `build_all.sh` script can take substantially over 1 hour to build everything. All of the software is built using Buildroot with the exception of the Renesas Qt demo, which is built afterwards.

For further information see <http://buildroot.uclibc.org>

The version of Buildroot used does not support the MS7724 board or the Renesas specific software packages; hence the scripts apply a patch once it has been extracted.

The scripts set the Buildroot `BUILDROOT_DL_DIR` environment variable to use the `installer/buildroot_dl` directory that is installed with the BSP. If you wish to use Buildroot after running our installation script, you should set this variable accordingly, for example:

```
export BUILDROOT_DL_DIR=/home/user/bsp/installer/buildroot_dl/
```

### 3.1 Re-building

If you wish to change the packages installed by Buildroot, anything built outside Buildroot (i.e. the Renesas Qt demo) will have to be re-installed to the rootfs. This can be done by running

```
./demo.sh
```

### 3.2 Building Qt demos

Buildroot does not provide an option to build the demos and examples supplied with Qt. If you wish to build a specific demo, it is quite easy to build it. The following example is for the `fluidlauncher` demo.

```
cd src/buildroot-2010.08/output/build/qt-*/demos/embedded/fluidlauncher
../../../../bin/qmake *.pro
sed -i 's|DEL_FILE\s*= |INSTALL_ROOT=/tftpboot/rootfs\nDEL_FILE = |' Makefile
make
sudo make install
```

Note that the `sed` command is used to specify the base of the target rootfs. If you have modified the `TFTP` variable in the `defines.sh` file, you will need to modify this line accordingly. This will install the files to the rootfs, normally `/usr/demos`.

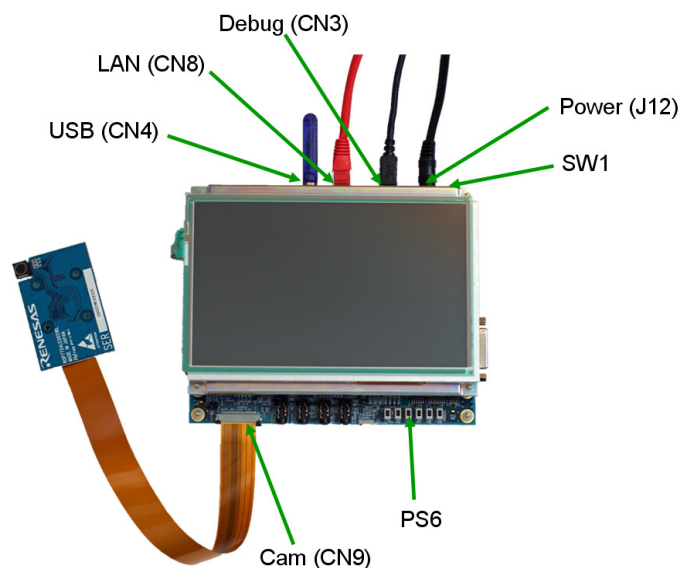
## 4. Using the BSP

In order to use the BSP, you need to connect the board to your linux host.

The Debug (CN3) connection is serial over USB and requires a driver to be installed on your host. Once installed, this provides a serial terminal connection which you can connect to with program like minicom.

The LAN (CN8) connection is used to access a TFTP server for downloading the kernel image, and for accessing the NFS root file system on your host PC. Make sure that this LAN just has your PC attached as the board is setup with a static IP address.

The USB Memory Stick (CN4) is not required as we use an NFS root file system for development. Do not power on the board (SW1) yet.



### 4.1 U-Boot Boot Loader

The linux kernel requires the system to be initialized into a known state before running. This is performed by a boot loader. The board has been programmed with U-Boot which is widely used in embedded systems. U-Boot is often used for both development and production environments as it allows booting from flash, USB, or a network. U-Boot can also be modified to use a splash screen whilst booting linux.

See section 5.1, <http://www.denx.de/wiki/U-Boot> and <http://www.denx.de/wiki/DULG/Manual> for further information on how to use U-Boot.

Once you have built the BSP, you can now run it on the board. U-Boot allows you to load the kernel image from a TFTP server. It can load a zImage or uImage kernel image, though using uImage is more flexible as it allows the Linux kernel to get its command line argument from the U-Boot bootargs environment variable. All of the instructions below assume that the host PC directory /tftpboot is served via the TFTP server. The instructions are run from the terminal program connected to the board's serial over USB debug port (e.g. minicom).

When you power up the board, you have to interrupt the boot sequence by pressing any key. Otherwise the board will boot into the demo program.

The following instructions tell U-Boot to load uImage from the host file system (in this case /tftpboot/rootfs/boot/uImage). In this example the bootargs variable specifies that the board uses a static IP address and a root file system on the host PC (/tftpboot/rootfs). Depending on your host IP address and target board you may need to modify these variables.

```
setenv bootcmd
setenv bootfile /rootfs/boot/uImage
setenv fileaddr 89000000
setenv serverip 192.168.10.30
setenv ipaddr 192.168.10.31
setenv bootargs console=ttySC0,115200 root=/dev/nfs
    nfsroot=${serverip}:/tftpboot/rootfs ip=${ipaddr} mem=248M
    memchunk.vpu=8m memchunk.veu0=4m memchunk.beu0=4m
saveenv
```

To load and then run the uImage run:

```
tftp
bootm
```

## 4.2 Examples

This section describes simple tests that can be run to ensure everything was built and installed correctly. It assumes you have followed all of the steps in section 3. All instructions below are run from the terminal program connected to the board's serial over USB debug port (e.g. minicom), after logging in as root (no password required), unless otherwise instructed.

We recommend that the blinking cursor is turned off:

```
echo 0 > /sys/class/graphics/fbcon/cursor_blink
```

and turn off messages about unaligned accesses:

```
echo 2 > /proc/cpu/alignment
```

### 4.2.1 Touch Screen

To setup the environment variables used by the touch screen library, tslib, run:

```
source /touch.env
```

To calibrate the touchscreen supported by tslib, run:

```
ts_calibrate
```

To test the touchscreen, run:

```
ts_test
```

### 4.2.2 Video Codecs

To play a sample video using the libshcodecs tools, run:

```
shcodecs-play /sample_media/movie/advert-h264-vga-25fps-2mbps.264 -s  
vga -r 24
```

To record a video from the camera, run:

```
cd /root  
shcodecs-record ctl/h264-video0-vga-stream.ctl > camera.264
```

To stop the recording, press Ctrl+C in the terminal.

To playback the recorded video, run:

```
shcodecs-play camera.264 -s vga -r 24
```

### 4.2.3 Streaming Video

The BSP includes the sighttpd streaming multimedia server. Sighttpd works with the shcodecs-record tool provided with libshcodecs.

To capture VGA data from one camera, resize this provide an additional QVGA source, encode both the VGA and QVGA data and stream this over HTTP, run:

```
sighttpd -f ctl/sighttpd-vga-qvga.conf
```

On your linux host, install mplayer and run:

```
mplayer http://192.168.10.31:3000/video0/vga.264 -fps 30
```

### 4.2.4 DirectFB

Note: DirectFB directly writes to the BEU registers rather than use the libshbeu library. This means it cannot be used at the same time as any other program using the BEU. Further, it doesn't write to all of the BEU registers so if another program that uses the BEU (e.g. shcodecs-record) has been used, DirectFB will not work correctly. Therefore, if you are following the steps in this document, the commands below will not work correctly without rebooting the hardware.

DirectFB for the SH772x uses a kernel module that needs to be loaded, run:

```
insmod /lib/modules/2.6.35/renesas/sh772x_gfx.ko
```

The BSP is set up to build only the df\_andi DirectFB example, run:

```
df_andi
```

To stop the demo, press Ctrl+C in the terminal.

### 4.2.5 GStreamer

GStreamer can connect together plugins based on its command line arguments or from compiled code. For our purposes we will just show a variety of GStreamer configurations specified on the command line.

#### 4.2.5.1 Encode camera to file with LCD preview

The BSP includes a script to perform this operation.

```
./rec-video.sh <filename>.avi
```

Note that when encoding, the size of the capture buffers and other encoding parameters are defined in the control file (specified with `cntl_file=`). However the encode size and frame rate is that specified in the GStreamer command line. The format specified on the command line must match that specified in the control file.

Note: The audio is stored as uncompressed PCM data.

#### 4.2.5.2 Decode recorded video

The BSP includes a script to perform this operation.

```
./play-video.sh <filename>.avi
```

### 4.2.6 Qt Demo

Perform the following steps:

```
source /touch.env  
./run_qt.sh ./menu
```

The `run_qt.sh` script sets up the `QWS_MOUSE_PROTO` and `QWS_KEYBOARD` environment variables based on what is currently connected to the board. This allows the board to be used with the touchscreen or USB mice/keyboards.

If you wish to make the demo run automatically on boot, you can add a script to the root file system in `/etc/init.d` containing the above commands. The scripts are run in order, so you should start the script name with `S90`.

Instructions for operating the demo are provided in the Quick Start Guide.

## 5. Supplementary Information

This section contains information that is useful but not essential to use the BSP.

### 5.1 Memory allocation

The multimedia libraries (libshcodecs, libshveu, libshbeu) and the SH multimedia plugins use User IO to access hardware registers and for memory used by the hardware IP blocks. The hardware IP blocks requires a special memory allocator as the memory must be physically contiguous.

The kernel Video For Linux (v4l2) interface is used for camera capture (CEU). For the best performance, the v4l2 USERPTR interface should be used so that captured frame buffers can be passed to other SH hardware IP blocks such as the VEU scaler.

UIO memory is managed by UIOMux; however the amount of memory available is specified to the kernel on the command line in the form 'memchunk.<device>=<size>'. If nothing is specified, relatively small defaults are used.

The following table provides a guide to the amount of UIO memory required to use the SH GStreamer plugins to perform common operations.

Resolution	Camera capture memchunk.veu0=	Video Decode & Encode memchunk.vpu=
QVGA (320x240)	1m	1m
VGA (640x480)	2m	3m
D1 (720x480)	Not supported	3m
SVGA (800x600)	2m	5m
720p (1280x720)	4m	8m

Note: DirectFB requires a block of memory to be reserved for its exclusive use. This is performed by setting the mem=248M kernel command line argument, leaving the upper 8MB for DirectFB.

#### 5.1.1 Camera Capture

The VEU memory size required for camera capture is approximately:

2 x (size of YCbCr420 frame)

Note: The current calculation of the capture buffer size rounds the frame size up to 16bpp.

Note: The camera sensor may not support the size you request.

#### 5.1.2 Video Decode & Encode

The VPU memory size when performing video encode is approximately:

For less than D1: 6 x (size of YCbCr420 frame)

Otherwise: 5.75 x (size of YCbCr420 frame)

## 5.2 U-Boot

The U-Boot documentation is very comprehensive; this section describes some typical uses.

It is worth pointing out that the command line arguments passed to the kernel is used to launch consoles, amongst other things. The examples below setup a single console attached to a SH772x serial port (ttySC0). If you want to see the text generated on the device whilst it is booting, you can add "console=tty1, " to the bootargs environment variable.

### 5.2.1 Dynamic IP address

This section describes setting up the board to use a dynamic IP address from a DHCP server.

<u>Item</u>	<u>Location</u>
U-Boot	Device flash memory
Target IP Address	Dynamic
Host IP Address	Static
Kernel	Loaded via TFTP from host PC
Root File System	Loaded via NFS from host PC

If the board has access to a DHCP server you can change the U-Boot environment variables so that the board gets its IP address from the DHCP server. The example below assumes that the DHCP server also sets the root-path option.

```
setenv bootfile /rootfs/boot/uImage
setenv fileaddr 89000000
setenv serverip 192.168.10.30
setenv bootargs console=ttySC0,115200 root=/dev/nfs ip=dhcp
                mem=248M memchunk.vpu=8m memchunk.veu0=4m
saveenv
```

To load and then run the uImage run:

```
tftp
bootm
```

### 5.2.2 Bootp

This section describes setting up the board to use the bootp protocol.

<u>Item</u>	<u>Location</u>
U-Boot	Device flash memory
Target IP Address	Dynamic
Host IP Address	Dynamic
Kernel	Loaded via TFTP from host PC
Root File System	Loaded via NFS from host PC

If you have a bootp server (normally part of the DHCP service), the board can send a bootp request broadcast to the server. The server will then send a bootp response with the IP addresses and uImage file.

```
setenv bootargs console=ttySC0,115200 root=/dev/nfs ip=dhcp
                mem=248M memchunk.vpu=8m memchunk.veu0=4m
saveenv
```

To load and then run the uImage run:

```
bootp
bootm
```

### 5.2.3 Modifying U-Boot

This section describes how to modify the U-Boot image stored in the device.

<u>Item</u>	<u>Location</u>
U-Boot	Loaded via TFTP from host PC, saved in device flash memory
Target IP Address	Static
Host IP Address	Static

You can re-program U-Boot, but be aware that a mistake or corrupt U-Boot file may result in your board failing to boot correctly. If your board can no longer boot after reprogramming U-Boot, please contact your Renesas representative. To program U-Boot, use the following commands:

```
setenv serverip 192.168.10.30
setenv ipaddr 192.168.10.31

tftp 89000000 /rootfs/boot/u-boot.bin
protect off A0000000 A001FFFF
erase A0000000 A001FFFF
cp.b 89000000 A0000000 <U-Boot size in hex>
protect on A0000000 A001FFFF
```

The U-Boot size is shown when you tftp it to the board.

Note that after re-programming U-Boot, the environment variables will resort to default values. Set them accordingly and save them.

### 5.2.4 Placing the kernel in flash memory

This section describes how to modify the kernel image stored in the device.

<u>Item</u>	<u>Location</u>
U-Boot	Device flash memory
Target IP Address	Static
Host IP Address	Static

You can program the kernel uImage into flash memory so you can create a standalone product. The following commands show the typical actions (you may need to change the size and location of the flash memory):

```
setenv serverip 192.168.10.30
setenv ipaddr 192.168.10.31

tftp 89000000 /rootfs/boot/uImage
protect off A0040000 A023FFFF
erase A0040000 A023FFFF
cp.b 89000000 A0040000 <uImage size in hex>
protect on A0040000 A023FFFF
```

The uImage size is shown when you tftp it to the board. To run the uImage run:

```
bootm A0040000
```

Alternatively you can set the U-Boot `bootcmd` environment variable to `bootm` to automatically boot the board. If you are setting `bootcmd`, you may want to set the `bootdelay` environment variable as well.

### 5.2.5 USB Root File System

You can tell the kernel to use a root file system mounted on a USB storage device by setting the `root`, `rootfstype` and `rootwait` kernel arguments, for example:

```
setenv bootargs console=ttySC0,115200 root=/dev/sda1
rootfstype=ext3 rootwait ip=192.168.10.31 mem=248M
memchunk.vpu=8m memchunk.veu0=4m
```

In this example, the USB storage device has to be mounted as `sda1`. If you have multiple USB storage device inserted, this may not be the case.



In order to create the root file system on the USB storage device, follow these instructions on your linux host:

**WARNING!** If a mistake is made in the device name, you may delete all of the data on your hard drive.

This requires a USB storage device larger than 512MB. The USB storage device is normally identified as '/dev/sdX', where X is the drive letter. Please check your environment for the correct device ID. Insert the USB storage device; if your system automatically mounts your USB storage device, unmount it first.

Note that the user interactions with are shown in **bold**.

### 5.2.5.1 Format USB storage device

```
sudo fdisk /dev/sdX
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 2024 MB, 2024275968 bytes
63 heads, 62 sectors/track, 1012 cylinders
Units = cylinders of 3906 * 512 = 1999872 bytes
Disk identifier: 0x63de1327
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	1012	1976405	b	W95 FAT32

```
Command (m for help): d
```

```
Selected partition 1
```

```
Command (m for help): n
```

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
P
```

```
Partition number (1-4): 1
```

```
First cylinder (1-1012, default 1): <Enter>
```

```
Using default value 1
```

```
Last cylinder, +cylinders or +size{K,M,G} (1-1012, default 1012): <Enter>
```

```
Using default value 1012
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

### 5.2.5.2 Copy root files system to the USB storage device

```
mkfs.ext3 /dev/sdX
```

```
mount -t ext3 /dev/sdX /mnt
```

```
cp -rd /tftpboot/rootfs/* /mnt
```

```
umount /mnt
```

<b>REVISION HISTORY</b>	<b>MS7724 Linux Golden Package BSP User's Manual</b>
-------------------------	--

Rev.	Date	Description	
		Page	Summary
2.00	Jun, 2010	—	Second Edition issued
2.01	Oct, 2010	12, 13	Use sighttpd for streaming video; Updated kernel to 2.6.35
		14	Clarified memory allocation.
		16	Updated size of flash used by kernel

---

MS7724 Linux Golden Package BSP User's Manual

Publication Date: Rev.2.02 08 Dec 2010

Published by: Renesas Electronics Corporation

---



## Renesas Electronics Corporation

### SALES OFFICES

<http://www.renesas.com>

---

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

**Renesas Electronics America Inc.**

2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**

1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: +44-1628-585-100, Fax: +44-1628-585-900

**Renesas Electronics Europe GmbH**

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**

7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**

Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**

Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**

7F, No. 363 Fu Shing North Road Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**

1 harbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: +65-6213-0200, Fax: +65-6278-8001

**Renesas Electronics Malaysia Sdn.Bhd.**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**

11F., Samik Laviel' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

# MS7724 Linux Golden Package BSP

