S3C6400/6410
Scenario Based APPLICATIONS
User's Manual
(Linux)

S3C6400/6410

July 5, 2008

REV 1.00

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product

S3C6400/6410 RISC Microprocessor
User's manual

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City Gyeonggi-Do, Korea
446-711

Home Page: http://www.samsungsemi.com/

E-Mail: mobilesol.cs@samsung.com

Printed in the Republic of Korea

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
1.00	Initial Version	-	Jiun Yu	2008-07-05

Contents

1 Introduction

1.1 Purpose

The purpose of the document explains how to test applications.

1.2 Scope

The scope of this document is to describe

- Data flow of applications
- How to test applications

1.3 Intended Audience

Intended Audience	Tick whenever Applicable
Project Manager	Yes
Project Leader	Yes
Project Team Member	Yes
Test Engineer	Yes

1.4 Definitions, Acronyms, and Abbreviations

Abbreviations	Description
JPEG	Joint Photographic Exports Grout
MFC	Multi Format Codec
API	Application Program Interface

1.5 References

Number	Reference	Description

2 Data flow of Applcations

2.1 Overview

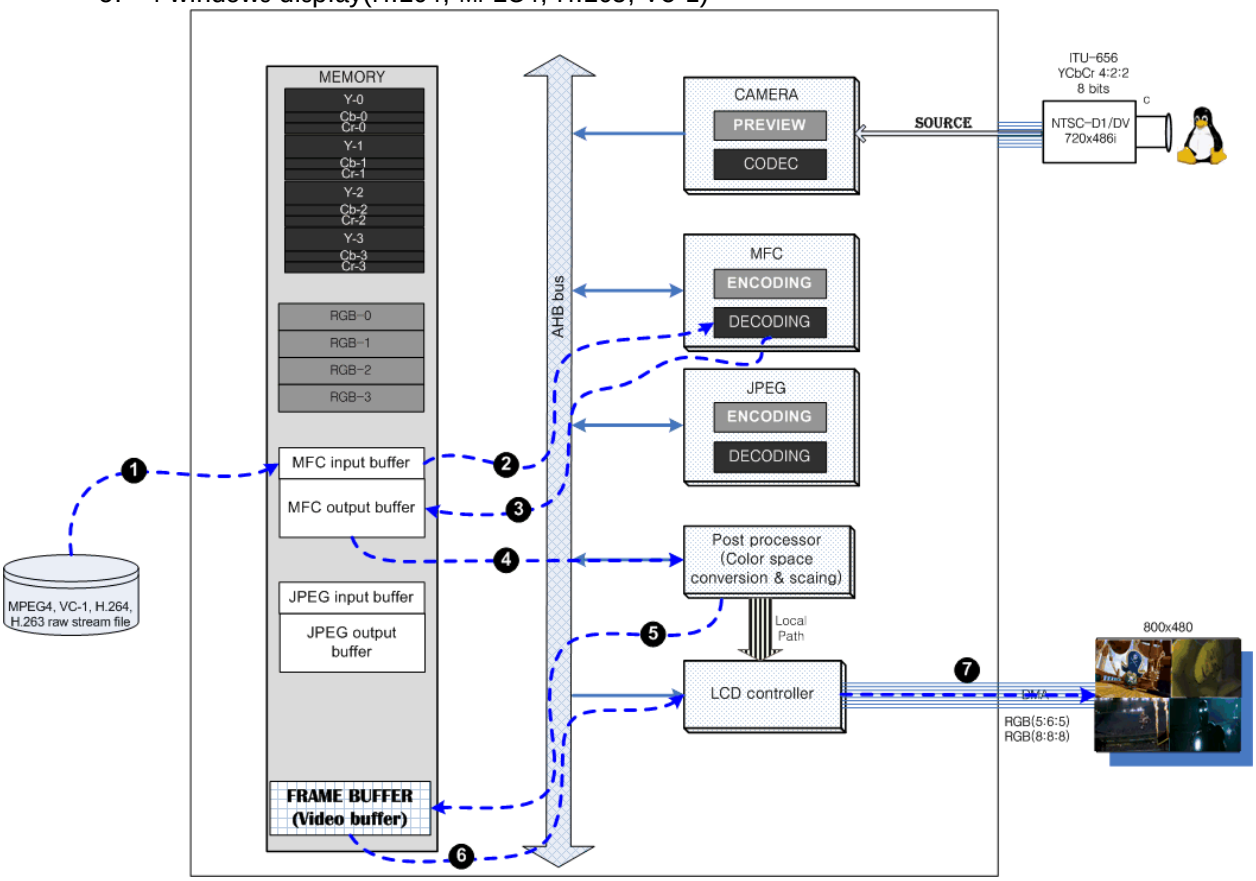
There are 12 scenario based applications. It uses several drivers these are Camera, MFC, post processor, display and JPEG driver. Each application has own scenario.

1. H.264 display
2. MPEG4 display
3. H.263 display
4. VC-1 display
5. 4 windows display(H.264, MPEG4, H.263, VC-1)
6. H.264 display using local path
7. H.264 display using double buffering
8. Camera preview & MFC encoding
9. H.264 display & Camera preview
10. Camera preview & MFC encoding/decoding
11. Camera input and JPEG encoding(Capture)
12. JPEG file dispaly on LCD

2.2 Basic display application

It includes

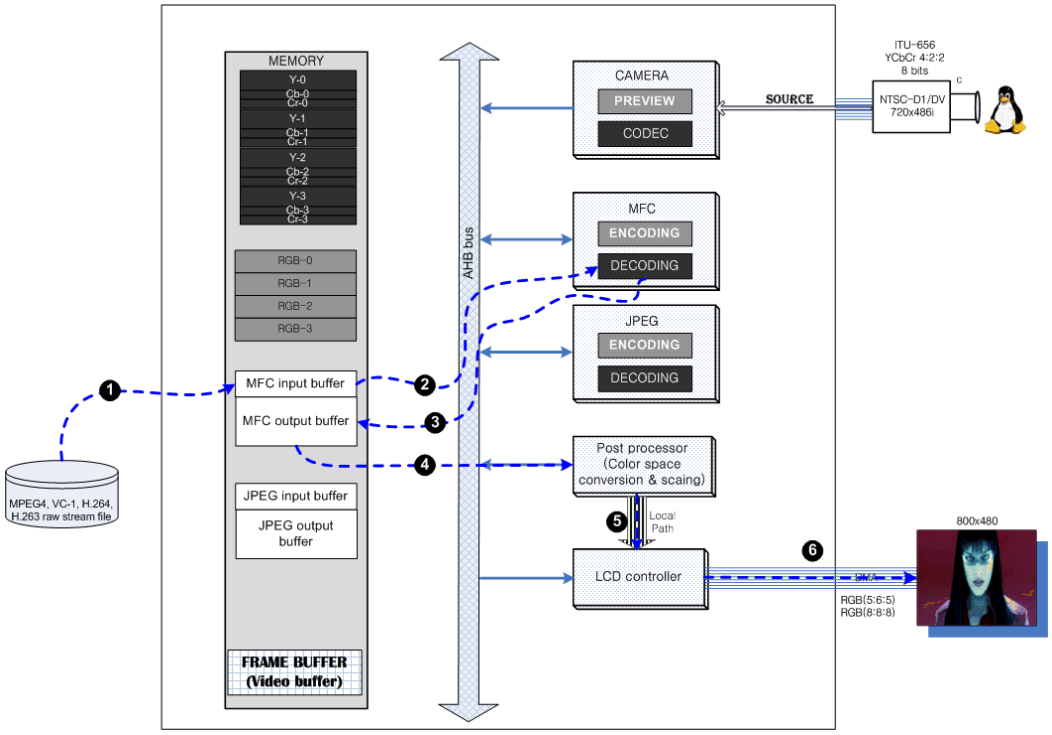
1. H.264 display
2. MPEG4 display
3. H.263 display
4. VC-1 display
5. 4 windows display(H.264, MPEG4, H.263, VC-1)



2.3 H.264 display using local path

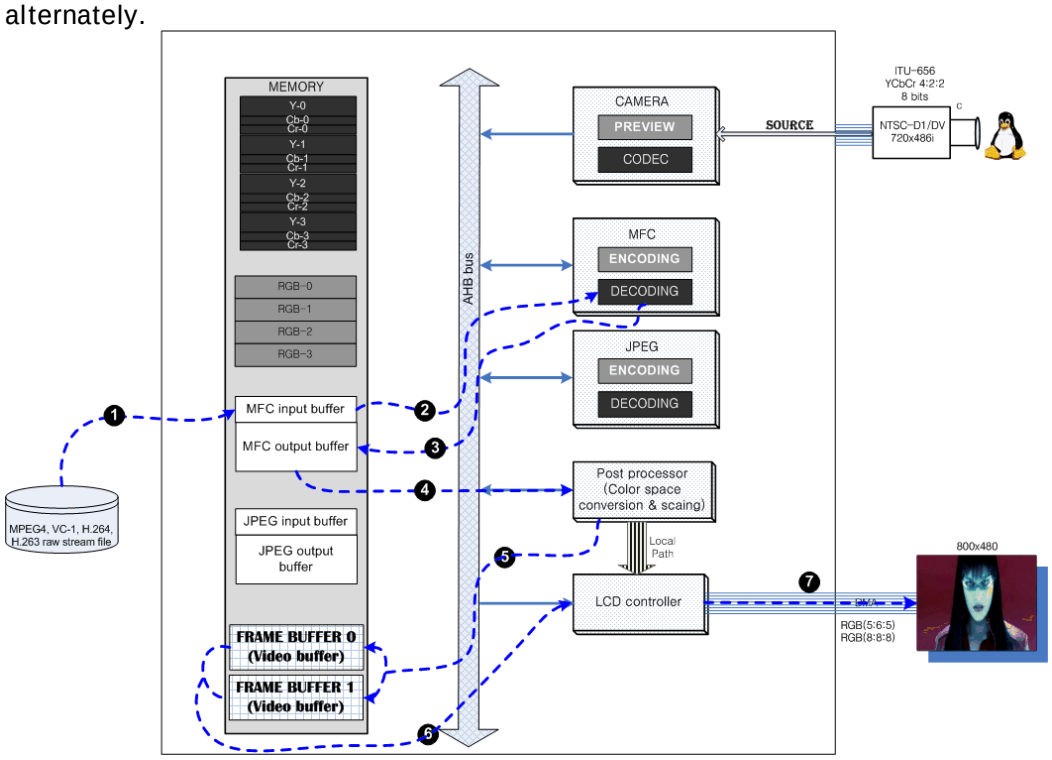
This application uses local path between post processor and display controller. Post processor's output data go through LCD controller directly. But it has 2 limitations.
These are :

1. Only Window 0 can be used when application uses local path
2. Only 24bpp support(RGB888)



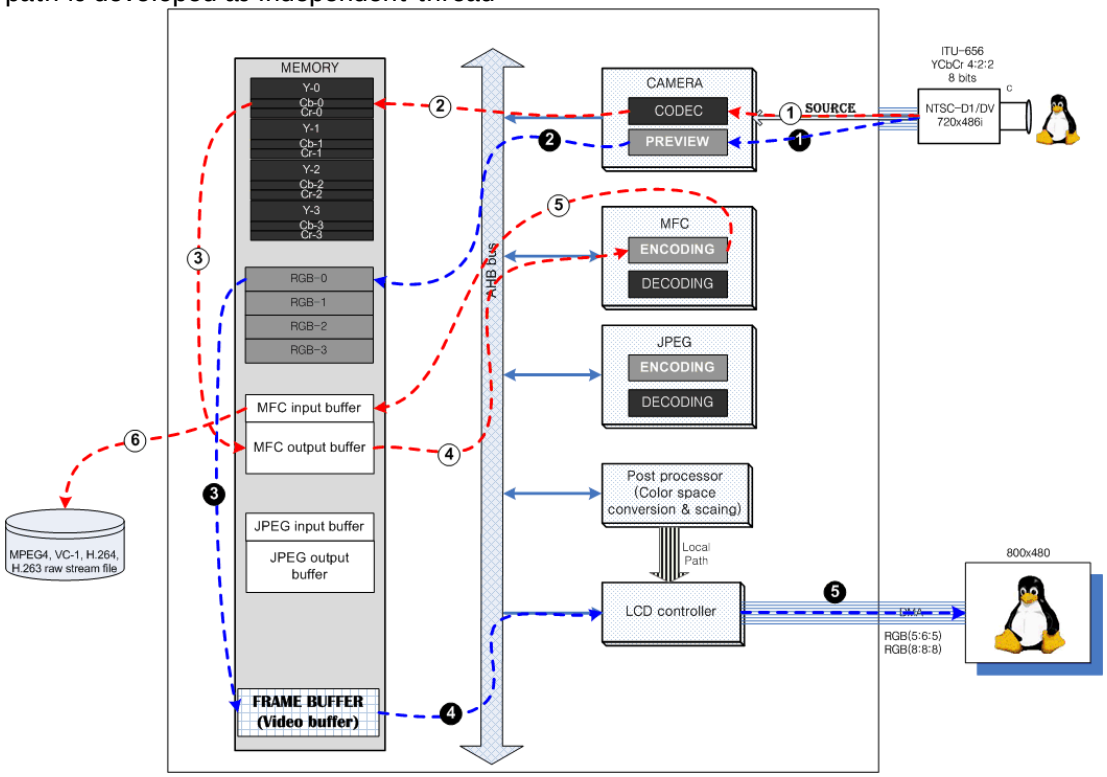
2.4 H.264 display using double buffering

This application uses 2 frame buffers(RGB buffer). LCD controller read data from frame buffer alternately.



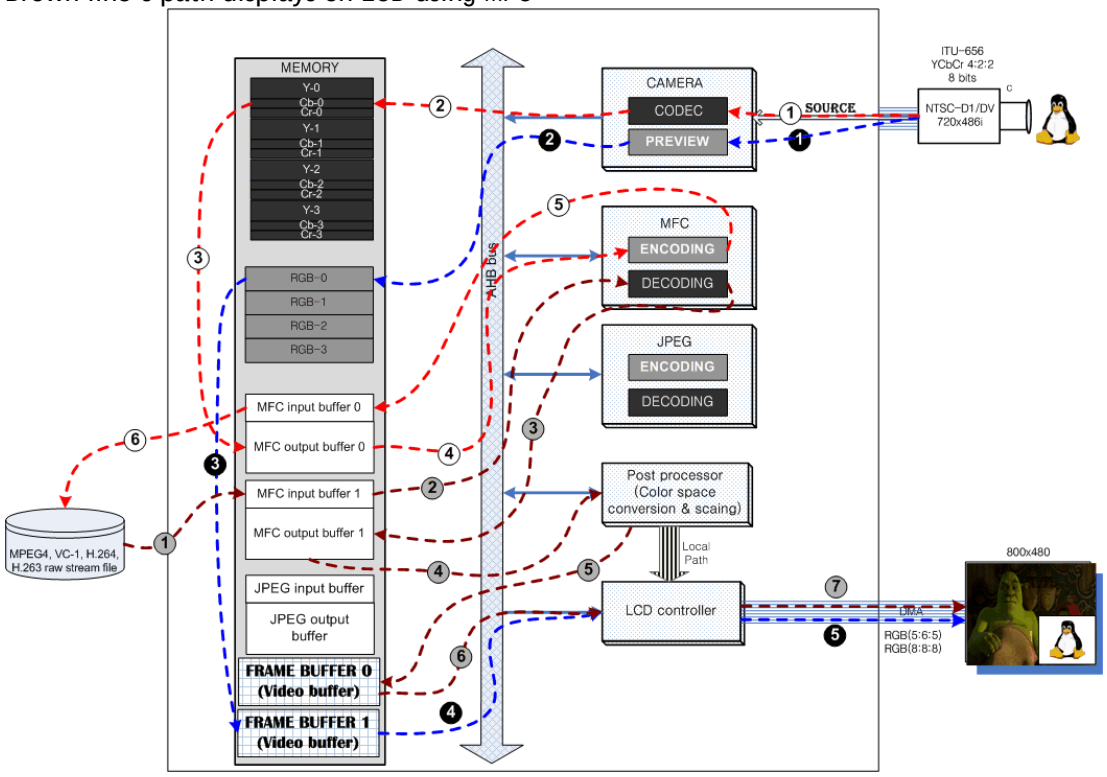
2.5 Camera preview & MFC encoding

This application does previewing and encoding the current frame simultaneously. Blue line is previewing in below figure and red line is to encode the current frame from camera codec path. Each path is developed as independent thread



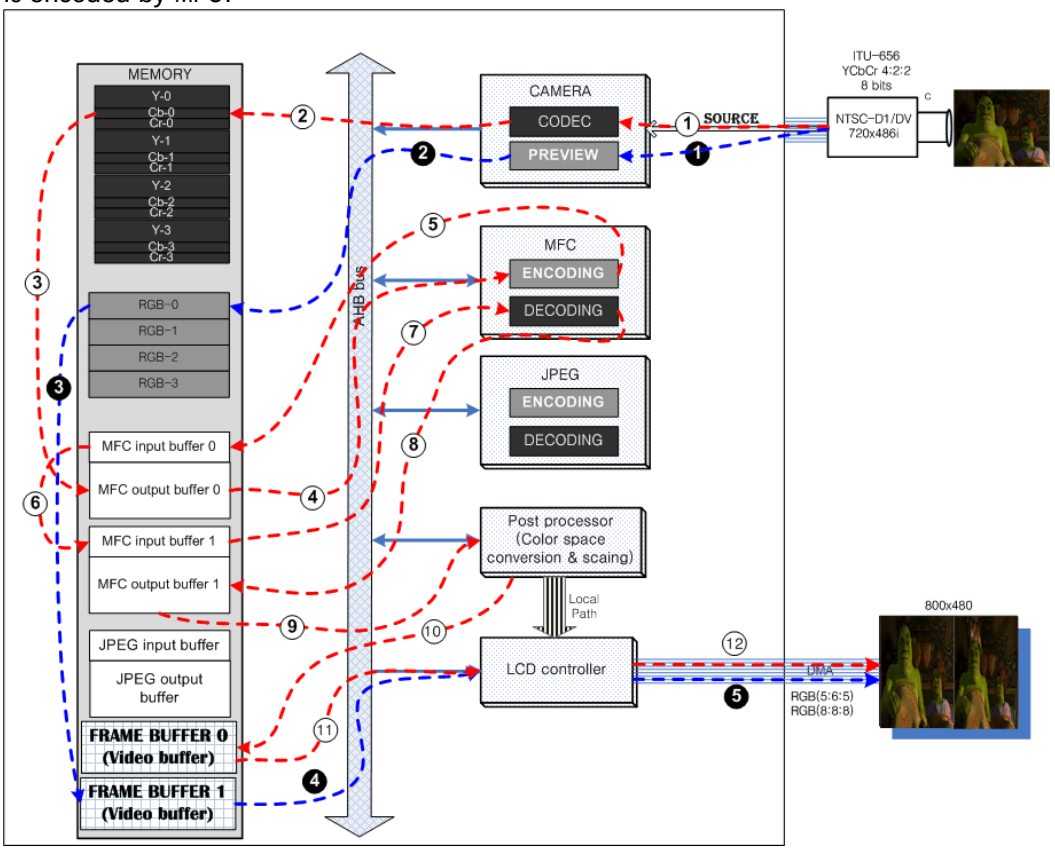
2.6 H.264 display & Camera preview

Camera previewing, MFC's encoding and decoding are operated in this application at the same time. Brown line's path displays on LCD using MFC



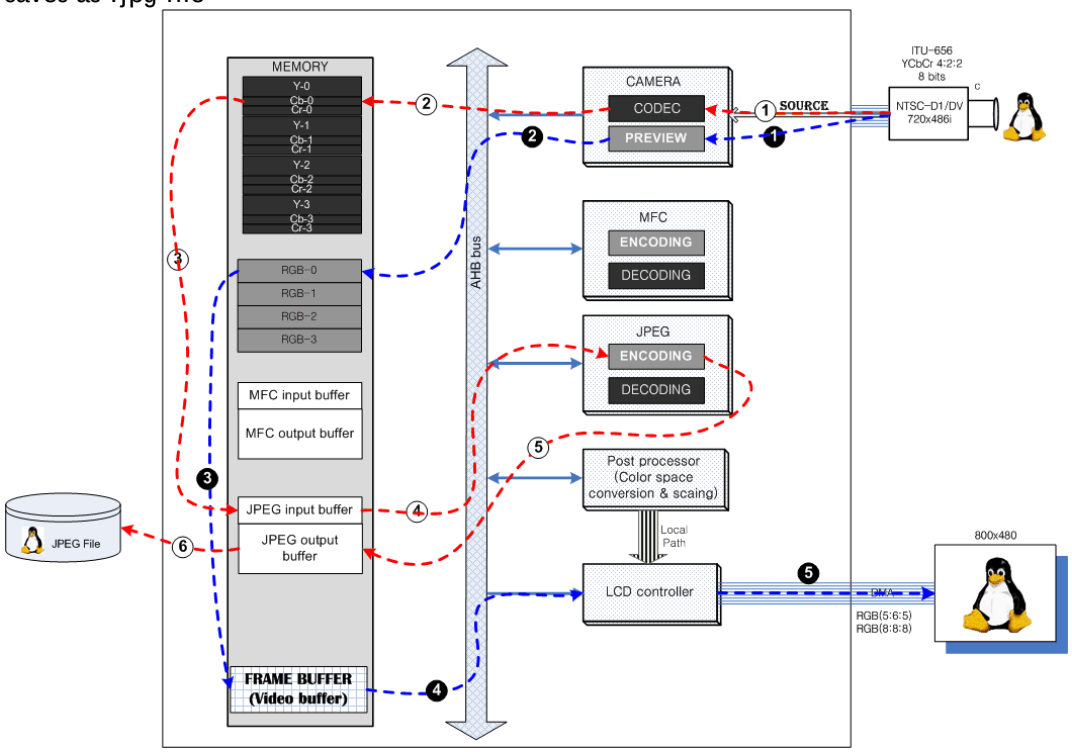
2.7 Camera preview & MFC encoding/decoding

Camera previewing is displayed on half of LCD and the other half of LCD displays decoded frame that is encoded by MFC.



2.8 Camera input and JPEG encoding

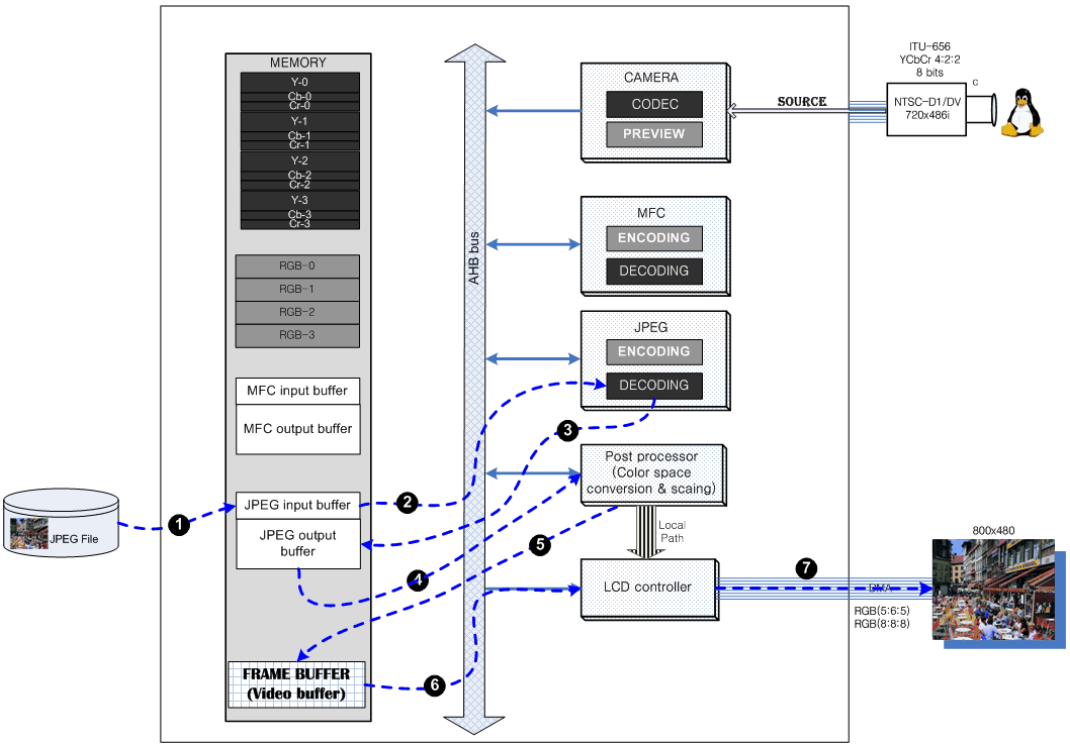
This application encodes the YUV frame from Camera codec path and then encoded file(JPEG file) saves as .jpg file



SAMSUNG ELECTRONICS

2.9 JPEG file display on LCD

This application displays on LCD after decoding jpg file



3 Package Guidelines

3.1 Directory Structure

Directory	Files	Description
/APPLICATIONS/Common	*.c, *.h	Common files of applications
/ APPLICATIONS/FrameExtractor	*.c, *.h	Frame extractor for decoder
/ APPLICATIONS/MFC_API	*.c, *.h	MFC API
/APPLICATIONS/JPEG_API	*.c, *.h	JPEG API
/ APPLICATIONS/TestVectors	*.jpg,, *.264, *.m4v, *.263, *.rcv	Test vectors for applications
/ APPLICATIONS/doc	*.doc, *.pdf	Documents for Scenario based applications
/ APPLICATIONS/	*.c, *.h	Source codes of the Scenario based applications

4 How to test Scenario based applications

4.1 Procedure to build kernel, API and test application

1. kernel compilation
2. module compilation
3. Test application compilation
4. loading kernel
5. insert module into kernel
6. Execute test application binary

4.2 Procedure to test

4.2.1 Kernel configuration and building

4.2.1.1 Configure reserved memory layout

Some devices need reserved memory because they have to allocate physically continuous memory. So you must setup reserved memory layout using below file named "reserved_mem.h" before kernel compilation.

```
/*
 * Defualt reserved memory size
 * MFC      : 6 MB
 * Post     : 8 MB
 * JPEG     : 8 MB
 * Camera   : 15 MB
 * These sizes can be modified
 */


//#define CONFIG_RESERVED_MEM_JPEG
//#define CONFIG_RESERVED_MEM_JPEG_POST
//#define CONFIG_RESERVED_MEM_MFC
//#define CONFIG_RESERVED_MEM_MFC_POST
//#define CONFIG_RESERVED_MEM_JPEG_MFC_POST
//#define CONFIG_RESERVED_MEM_JPEG_CAMERA
//#define CONFIG_RESERVED_MEM_JPEG_POST_CAMERA
//#define CONFIG_RESERVED_MEM_MFC_CAMERA
//#define CONFIG_RESERVED_MEM_MFC_POST_CAMERA
#define CONFIG_RESERVED_MEM_JPEG_MFC_POST_CAMERA
```

[NOTE]
1. This file is in
Include/asm-arm/arch-s3c64xx/reserved_mem.h (Linux2.6.16)
Include/asm-arm/arch-s3c2410/reserved_mem.h (Linux2.6.21)

2. For detailed information about how to build Linux kernel and how to download kernel image and cramfs, please refer to related porting guide documents.

You should modify DMA-consistent memory region in "include/asm-arm/memory.h" file.

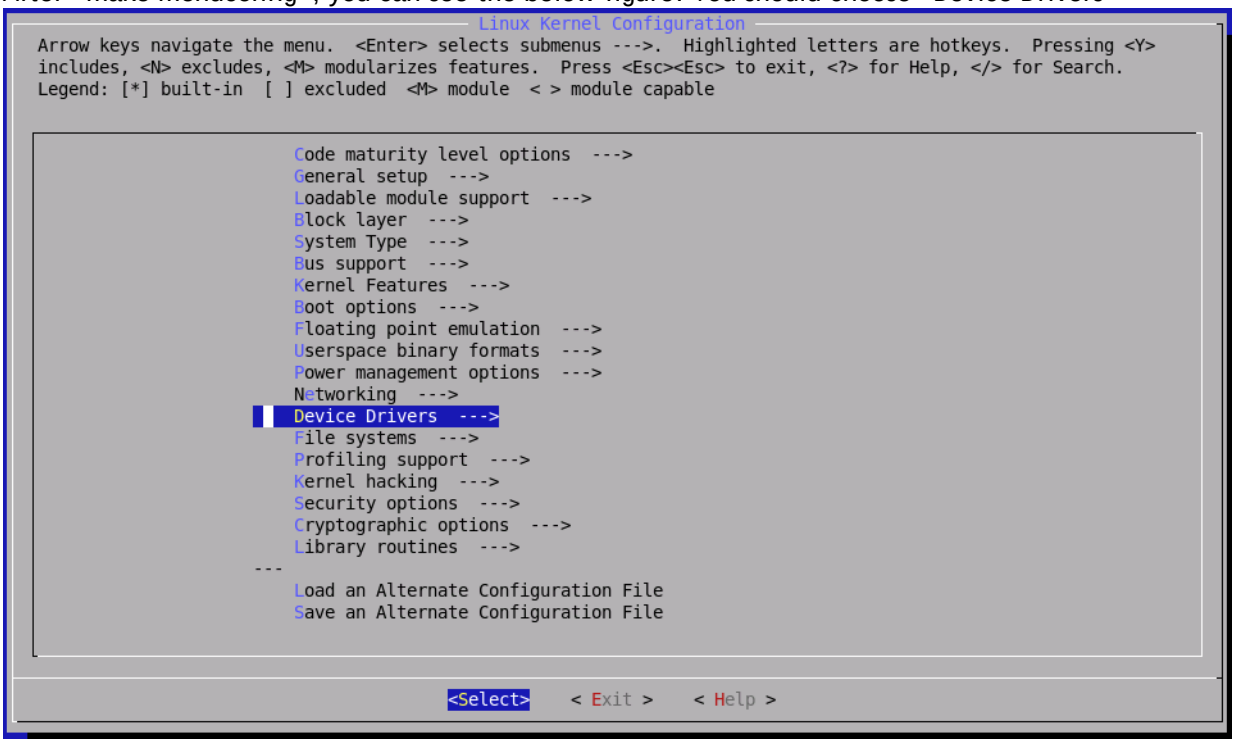
```
00112: /*
00113:  * Size of DMA-consistent memory region.  Must be multiple of 2M,
00114:  * between 2MB and 14MB inclusive.
00115:  */
00116: #ifndef CONSISTENT_DMA_SIZE
00117: #define CONSISTENT_DMA_SIZE (SZ_8M + SZ_4M)
00118: #endif
```

4.2.1.2 Configure Graphics options

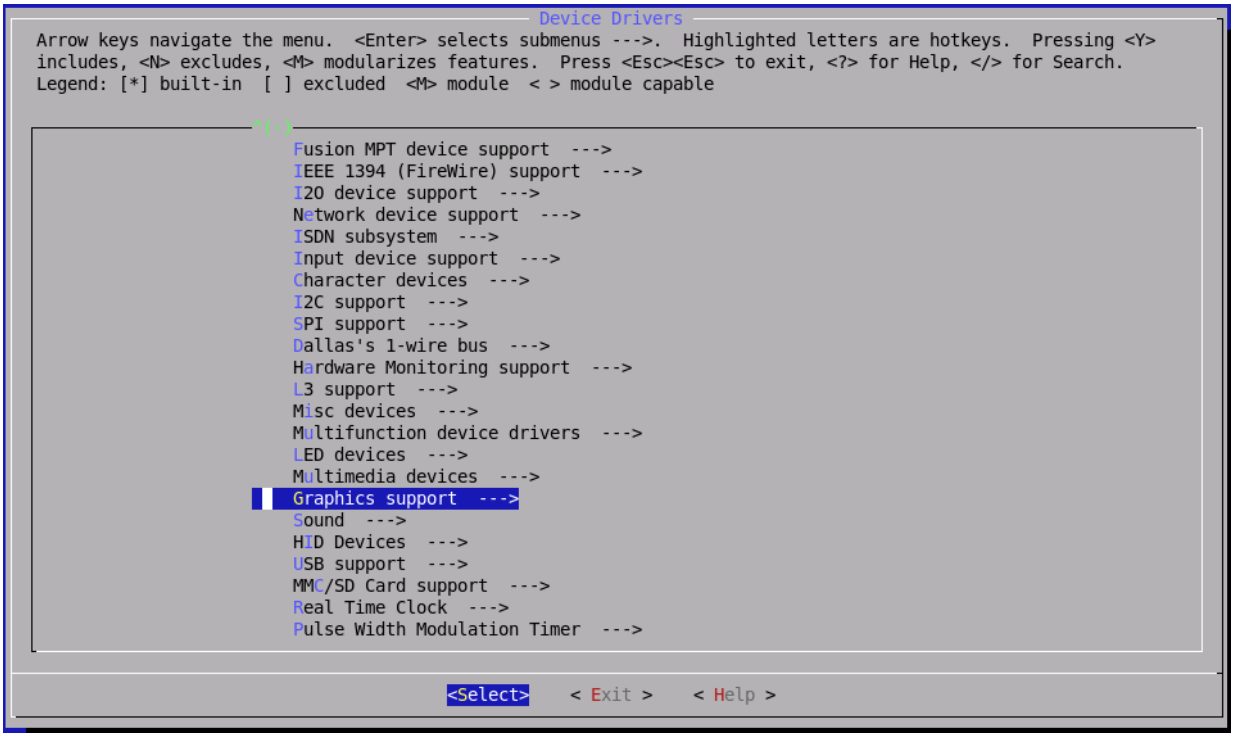
You should command "make menuconfig"

```
[root@localhost s3c-linux-2.6.21]# make menuconfig
```

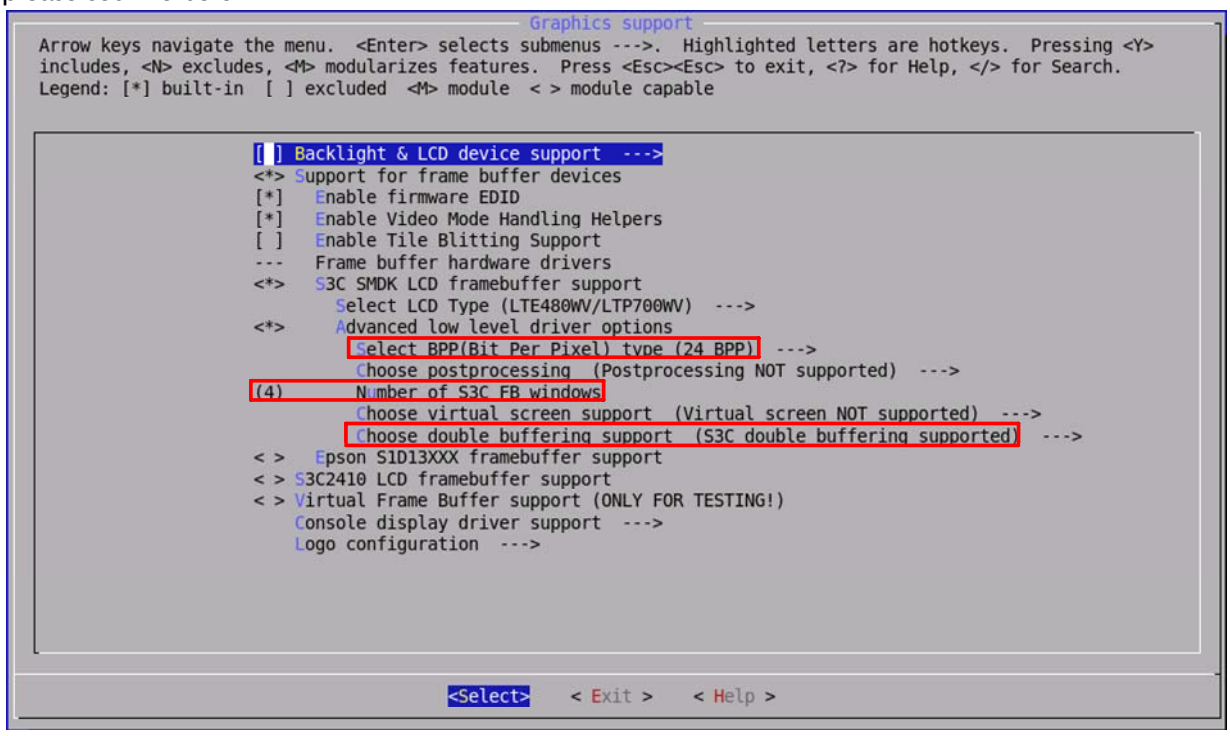
After "make menuconfig", you can see the below figure. You should choose "Device Drivers"



and then, please choose "Graphics support"



please set like below



4.2.2 Module compilation

4.2.2.1 JPEG driver compilation

JPEG device driver should be compiled as a kernel module.

1. Modify Makefile
You should modify "Makefile" to set configuration according to your environment.

```
[root@localhost s3c-linux-2.6.21]# cd [module directory]/Multimedia_DD/JPEG_V1.01/jpeg_drv

[root@localhost jpeg_drv]# vi Makefile
```

```
#####################################################
# Makefile for JPEG Driver
# 2007 (C) Samsung Electronics
# Author : Jiun. Yu <jiun.yu@samsung.com>
#####################################################

KERNEL_DIR := /home/mobile/workspace/s3c-linux-2.6.21
TOPDIR     := /home/mobile/workspace/s3c-linux-2.6.21

obj-m              := s3c_jpeg.o

s3c_jpeg-y := LogMsg.o JPGOpr.o JPGMisc.o JPGMem.o s3c-jpeg.o

PWD                        := $(shell pwd)

here:
        (cd $(KERNEL_DIR); make SUBDIRS=$(PWD) modules)

clean:
        rm -rf *.ko
        rm -rf *.mod.*
        rm -rf .*.cmd
        rm -rf *.o
        rm -rf Module.*
```

2. Module compilation
After compilation, you can find newly created files. "s3c_jpeg.ko" file is module of JPEG device driver.

```
[root@localhost jpeg_drv]# make
```

4.2.2.2 MFC driver compilation

MFC device driver should be compiled as a kernel module.

1. Modify Makefile
You should modify "Makefile" to set configuration according to your environment.

```
[root@localhost s3c-linux-2.6.21]# cd [module directory]/Multimedia_DD/FIMV_MFC_V1.0/mfc_drv

[root@localhost mfc_drv]# vi Makefile
```

```
#####################################################
# Makefile for MFC Driver
# 2007 (C) Samsung Electronics
# Author : Jiun. Yu <jiun.yu@samsung.com>
#####################################################

KERNEL_DIR := /home/mobile/workspace/s3c-linux-2.6.21
TOPDIR     := /home/mobile/workspace/s3c-linux-2.6.21
```

```
CFLAGS += -DLINUX
#CFLAGS += -DDIVX_TEST

obj-m              := s3c_mfc.o

s3c_mfc-y := Prism_0503.o BitProcBuf.o DataBuf.o FramBufMgr.o \
                          LogMsg.o MFC_HW_Init.o MFC_Inst_Pool.o MFC_Instance.o MfcMemory.o
MfcMutex.o MfcSfr.o     \
                          s3c-mfc.o MfcIntrNotification.o MfcSetConfig.o

PWD                       := $(shell pwd)

here:
        (cd $(KERNEL_DIR); make SUBDIRS=$(PWD) modules)

clean:
        rm -rf *.ko
        rm -rf *.mod.*
        rm -rf .*.cmd
        rm -rf *.o
        rm -rf Module.*
```

2. Module compilation
After compilation, you can find newly created files. "s3c_mfc.ko" file is module of MFC device driver.

```
[root@localhost mfc_drv]# make
```

4.2.2.3 Post processor driver compilation

Post processor device driver should be compiled as a kernel module.

1. Modify Makefile
You should modify "Makefile" to set configuration according to your environment.

```
[root@localhost s3c-linux-2.6.21]# cd [module directory]/Multimedia_DD/PP_V2.5/pp_drv

[root@localhost pp_drv]# vi Makefile
```

```
#######################################################
# Makefile for Post Processor
# 2007 (C) Samsung Electronics
# Author : Jiun. Yu <jiun.yu@samsung.com>
#######################################################

KERNEL_DIR := /home/mobile/workspace/s3c-linux-2.6.21
TOPDIR     := /home/mobile/workspace/s3c-linux-2.6.21

obj-m              := s3c_pp.o

s3c_pp-y := s3c_pp_common.o s3c_pp_6400.o

PWD                       := $(shell pwd)

here:
        (cd $(KERNEL_DIR); make SUBDIRS=$(PWD) modules)

clean:
        rm -rf *.ko
```

```
rm -rf *.mod.*
rm -rf .*.cmd
rm -rf *.o
rm -rf Module.*
```

2. Module compilation
After compilation, you can find newly created files. "s3c_pp.ko" file is module of Post processor device driver.

```
[root@localhost pp_drv]# make
```

4.2.3 Test application compilation

After compilation, you can see the "app" which is executable file.

```
[root@localhost s3c-linux-2.6.21]# cd [module directory]/Multimedia_DD/APPLICATIONS

[root@localhost APPLICATIONS]# make
```

4.2.4 Test application execution(In target side)

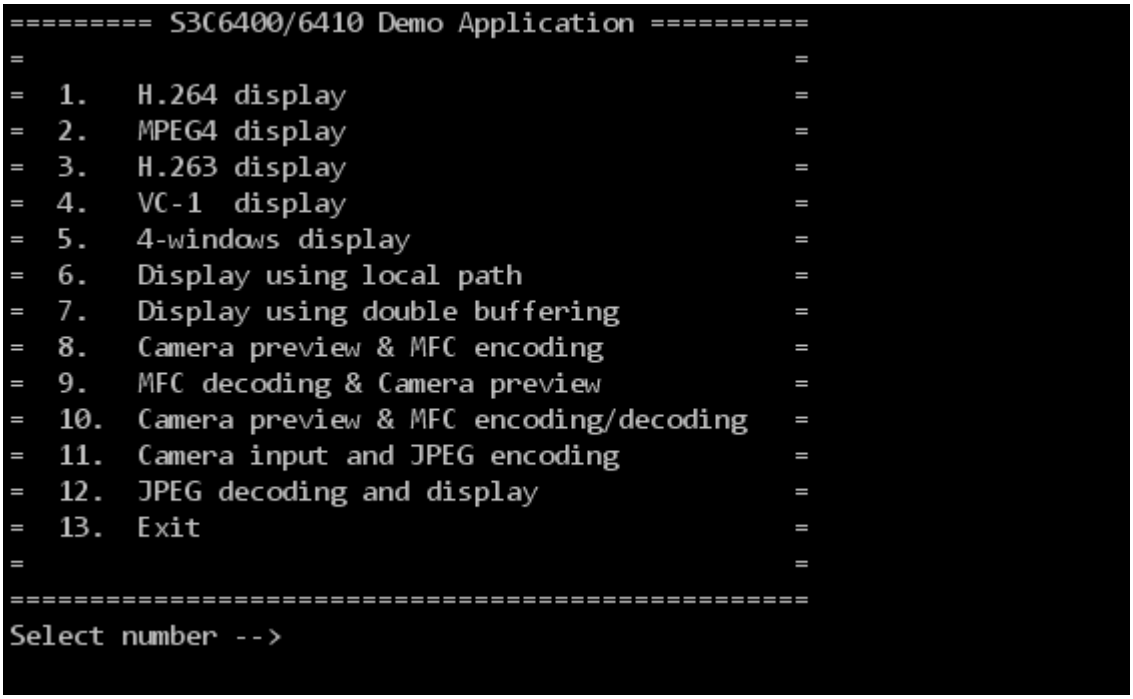
After kernel booting, you should load driver modules(MFC, post processor, JPEG) firstly.

```
[root@Samsung Multimedia_DD]# insmod FIMV_MFC_V1.0/mfc_drv/s3c_mfc.ko
S3C6400 MFC Driver, (c) 2007 Samsung Electronics
S3C6400 MFC driver module init OK.
[root@Samsung Multimedia_DD]# insmod PP_V2.5/pp_drv/s3c_pp.ko
S3C PostProcessor Driver, (c) 2007 Samsung Electronics
[root@Samsung Multimedia_DD]# insmod JPEG_V1.01/jpeg_drv/s3c_jpeg.ko
S3C JPEG Driver, (c) 2007 Samsung Electronics
[root@Samsung Multimedia_DD]#
```

Second, you should execute "app" file which is in "Multimedia_DD/APPLICATIONS/"

```
[root@Samsung Multimedia_DD]# cd APPLICATIONS/
[root@Samsung APPLICATIONS]# ./app
```

And then, please select application you want to test


========= S3C6400/6410 Demo Application ==========
=   =
= 1.   H.264 display   =
= 2.   MPEG4 display   =
= 3.   H.263 display   =
= 4.   VC-1 display   =
= 5.   4-windows display   =
= 6.   Display using local path   =
= 7.   Display using double buffering   =
= 8.   Camera preview & MFC encoding   =
= 9.   MFC decoding & Camera preview   =
= 10. Camera preview & MFC encoding/decoding   =
= 11. Camera input and JPEG encoding   =
= 12. JPEG decoding and display   =
= 13. Exit   =
=   =
==
Select number -->