

Application Report

Feb 2012

External Video (Encoder, Decoder) Device Configuration for DM81XX DVRSDK

Video Surveillance Applications

ABSTRACT

For TMS320DM81XX there are two ways to control external devices using i2c, i2c devices can be controlled or configured from A8 as well as from M3. Configuring from M3 side needs a device specific driver to be present which can work with bios. On the other hand, Linux on A8 provides i2c-dev device node which can be used for i2c device configuration from Linux user space. It allows user to choose different i2c controlled devices which can be interfaced with TMS320DM81XX and it removes the overhead of writing a bios driver for the external device.

Video Surveillance Applications	1
ABSTRACT	1
1 Overview	1
2 Changes needed in Application	1
2.1 Parameter changes	1
2.2 Additional APIs	2
3 MCFW I2C APIs	2
4 Code reference	2
5 I2C Utility	2

1 Overview

DVR RDK GA release supports configuration of external devices like TVP5158, Sii9022A and THS filters from A8 side by default. Using the i2c APIs provided in MCFW similar external devices can be controlled from Linux user space.

2 Changes needed in Application

2.1 Parameter changes

Set following parameters to FALSE in the demo application as shown below

```
/* By Default enableConfigExtVideoDecoder is set to TRUE*/
Vcap_params_init(&vcapParams);
vcapParams.enableConfigExtVideoDecoder = FALSE;
Vcap_init(&vcapParams);

/* By Default enableConfigExtVideoEncoder and enableConfigExtThsFilter is set to TRUE*/
Vdis_params_init(&vdisParams);
vdisParams.enableConfigExtVideoEncoder = FALSE;
vdisParams.enableConfigExtThsFilter = FALSE;
Vdis_init(&vdisParams);
```

This assumes external devices will be controlled by a driver/application external to MCFW and MCFW will not do i2c initialization for TVP5158, Sii9022A and THS filters. Here user takes responsibility for i2c initialization and controls devices using custom external device driver or a user space application.

The Linux user space application/function to initialize custom devices should run before VCAP and VDIS initialization or before VCAP and VDIS starts processing

2.2 Additional APIs

Please ensure you are calling `Vcap_setVideoSourceStatus()` from your demo application. This API will set channel specific information returned by your device. Based on this information MCFW internal parameters like `isPalMode` are set. For details of `Vcap_setVideoSourceStatus()` please refer `dvr_rdk/mcfw/interfaces/ti_vcap.h`

3 MCFW I2C APIs

A user space application can be written to do i2c read writes to a device using `OSA_i2c` APIs. Following i2c APIs are supported

```
int OSA_i2cOpen(OSA_I2cHndl *hndl, Uint8 instId);
int OSA_i2cRead8(OSA_I2cHndl *hndl, Uint16 devAddr, Uint8 *reg, Uint8 *value, Uint32 count);
int OSA_i2cWrite8(OSA_I2cHndl *hndl, Uint16 devAddr, Uint8 *reg, Uint8 *value, Uint32 count);
int OSA_i2cRawWrite8(OSA_I2cHndl *hndl, Uint16 devAddr, Uint8 *value, Uint32 count);
int OSA_i2cRawRead8(OSA_I2cHndl *hndl, Uint16 devAddr, Uint8 *value, Uint32 count);
int OSA_i2cClose(OSA_I2cHndl *hndl);
```

These APIs in turn call in standard Linux i2c APIs through ioctls provided in `<linux/i2c-dev.h>`.

4 Code reference

Please have a look at Linux user side implementation present in MCFW to control TVP5158, Sii9022A and THS filters for reference. It is present in

`dvr_rdk/mcfw/src_linux/devices/`

Similarly, a Linux user space application can be written for custom devices.

5 I2C Utility

DVR RDK provides a Linux user space i2c utility which allows user to read write to a particular i2c device. This utility is very useful especially if you are adding your custom device. This utility uses same `OSA_i2c` APIs to do i2c transactions.

When MCFW is built Binary for this utility gets transferred to target file system at

`/opt/dvr_rdk/ti814x/bin/i2c_rdwr.out`

Usage

```
i2c_rdwr.out -r|-w <devAddrInHex> <regAddrInHex> <regValueInHex or
numRegsToReadInDec>
```