Thor96 Camera Interface User Manual

i.MX8MQ Thor96 Reference Design

Date: March 10, 2022 | Version 1.1

CONTENTS

1 INTRODUCTION 4
1.1 Purpose of the Document 4
1.2 About the System 4
1.3 Intended Audience 4
1.4 Prerequisites 5
2 ENVIRONMENT SETUP 6
2.1 Hardware Installation 6
2.2 Open board's terminal- console (minicom) on x86 host PC 7
3 CAMERA DEMO 8
3.1 Live stream of AR0430 Camera on HDMI display 8
3.2 Live stream of ARX3A0 Camera on HDMI display 9
3.3 Capture image from camera 11
3.4 Validation of ISP controlled feature of AP1302: 13
3.4.1 Resolution, FPS and Data format 13
3.4.2 Brightness 14
3.4.3 Contrast 14
3.4.4 Saturation 14
3.4.5 Gamma 14
3.4.6 Exposure 15
3.4.7 Gain 15
3.4.8 White Balance 15
3.5 AP1302 Register Read / Write 15
3.5.1 Read AP1302 Registers: 15
3.5.2 Write AP1302 Registers: 16
4 LIMITATION 17

FIGURES

Figure 1: I.MX8MQ Thor96 board 4
Figure 2 : Thor96 board setup 6
Figure 3 : Thor96 – UART connection 7
Figure 4 : THOR96 SRT-Vision-AR0430 mezzanine setup 8
Figure 5: AR0430 Live stream in HDMI Display 9
Figure 6 : THOR96 SRT-Vision-ARX3A0 mezzanine setup 9
Figure 7 : ARX3A0 Live Stream on HDMI Display 10
Figure 8 : AR0430 Captured image 11
Figure 9: ARX3A0 Captured image 12

ACRONYMS AND ABBREVIATIONS

Definition/Acronym/Abbreviation	Description
cd	Change directory
scp	Secure copy over the network
BSP	Board Support Package
USB	Universal Serial Bus
HW	Hardware
FW	Firmware
NV RAM	Non-Volatile Random-Access Memory
API	Application Programming Interface
V4L2	Video for Linux second version
ISP	Image signal processor
SD	Secure Digital
HDMI	High-Definition multimedia interface
LTS	Long Term Support
UART	universal asynchronous receiver-transmitter
PC	Personal computer
FAT	File Allocation Table
FPS	Frames per second

1 INTRODUCTION

1.1 Purpose of the Document

Purpose of this document is to help developers flash firmware and demonstrate camera interface on i.MX8MQ-thor96 firmware. For demo, we have used AP1302 ISP with either one of the following camera sensors:

- SRT-Vision96-AR0430 mezzanine (AP-vision-AR0430)
- SRT-Vision96-ARX3A0 mezzanine (AP-vision-ARX3A0)
- SRT-Vision96-AR1335 mezzanine (AP-vision-AR1335)

1.2 About the System

This system is based on i.MX8MQ processor and supporting multiple interfaces. This can facilitate for Human-Machine Interface experience.

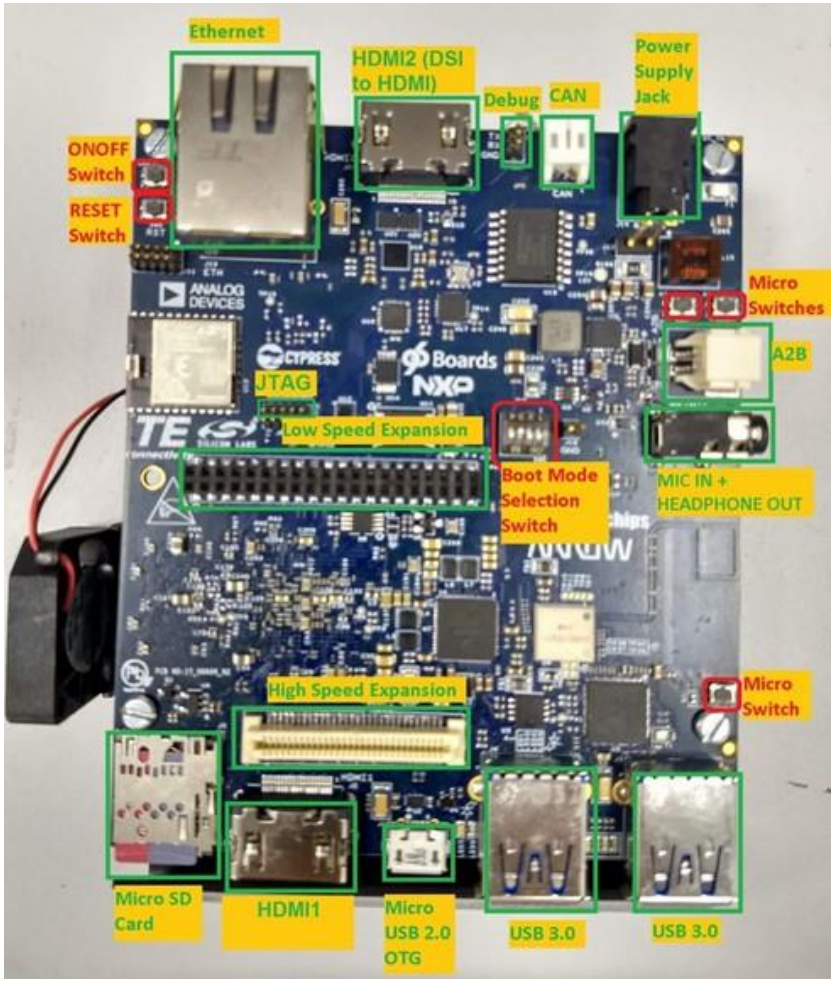

Figure 1: I.MX8MQ Thor96 board

1.3 Intended Audience

This document is for developers and end-users who want to understand/flash/demonstrate camera interface on i.MX8MQ-thor96 firmware.

1.4 Prerequisites

Below are the list of hardware and software needed to demonstrate the Camera interface on I.MX8MQ-Thor96 Board:

- Thor96 Board
- SD-card -32GB
- UART debug cable
- Power Supply - Power Accessories - 96Boards
- One of the following mezzanine cards:
  - SRT-Vision96-AR0430 mezzanine (AP-vision-AR0430)
  - SRT-Vision96-ARX3A0 mezzanine (AP-vision-ARX3A0)
  - SRT-Vision96-AR1335 mezzanine (AP-vision-AR1335)
- HDMI Display
- USB Keyboard
- USB Mouse

2 ENVIRONMENT SETUP

2.1 Hardware Installation

- Place hardware board on a clean anti-static surface
- Insert flashed SD card to J5 SD card slot
- Attach serial cable's micro end to board's J10 Connector (near Ethernet connector) and USB end to host system i.e. x86 PC's USB connector
- Attach Ethernet cable to board's Ethernet connector J12
- Provide 12V-5A power supply (provided with board) to board on J14 DC_IN connector. After all the other hardware setup is done and required interfaces are connected to board, Topview of the board will looks like as following:

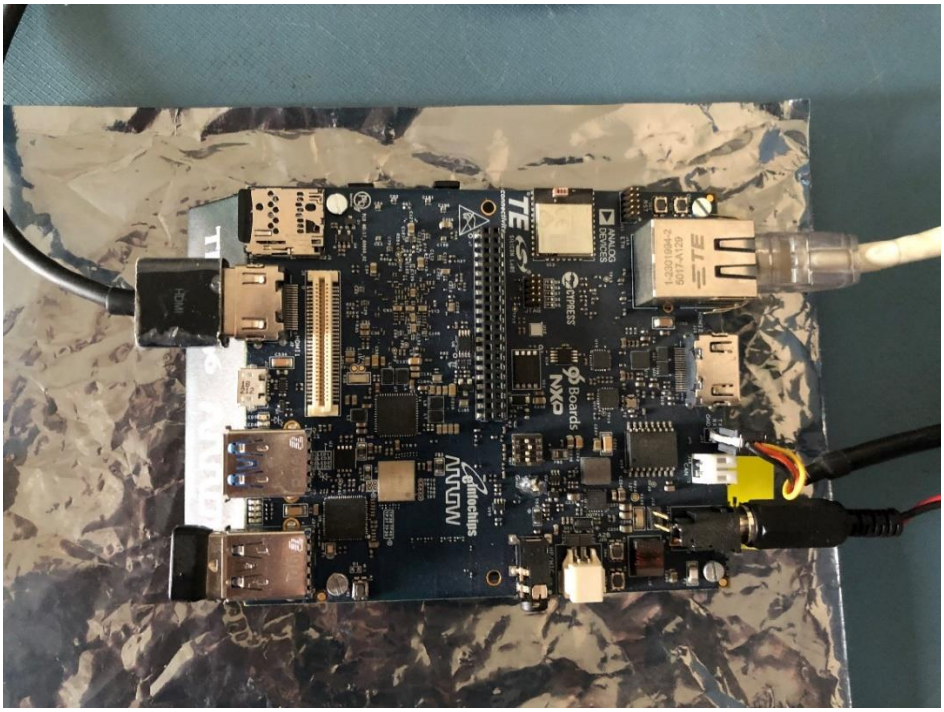

Figure 2 : Thor96 board setup

2.2 Open board's terminal- console (minicom) on x86 host PC

- Ensure SD card is flashed, and serial cable is plugged-in to the board as described in hardware setup section as following:

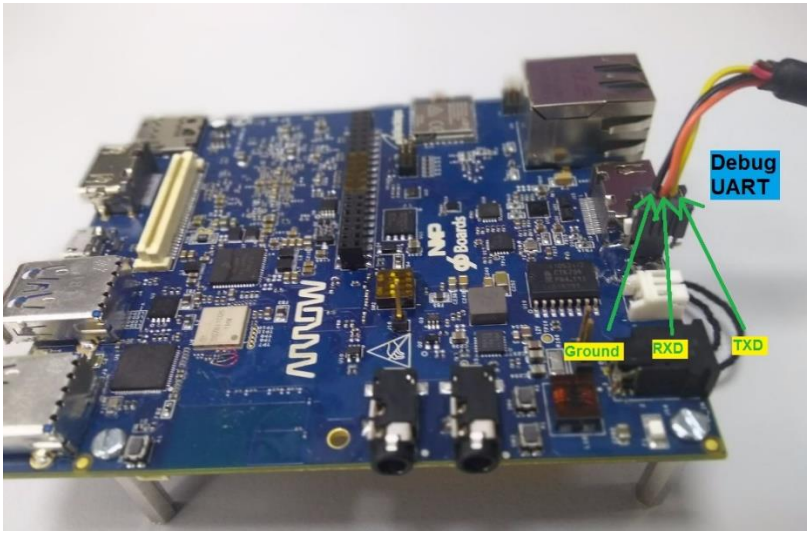

Figure 3 : Thor96 – UART connection

- Attach serial cable's USB end to host x86 PC's USB.
- Ensure minicom is installed in x86 Ubuntu PC (host system)
- Apply below command to open serial command's setting

```
$sudo minicom -s
```

- set baud rate and other setting as per below
  - baud rate=115200,
  - parity=none
  - hardware flow control = none
  - software flow control = none
  - serial device= /dev/ttyUSB0
  - save setup as dfl
- Once board gets powered-up, the above configured terminal will show logs on x86 and can interact with board using this open terminal

3 CAMERA DEMO

3.1 Live stream of AR0430 Camera on HDMI display

- To watch live stream of AR0430 camera sensor over the HDMI, connect HDMI Display, USB keyboard and Mouse to THOR96 board
- Attach **SRT-Vision96-AR0430 mezzanine (AP-vision-AR0430)** camera mezzanine module to high-speed expansion connector as following:

Figure 4 : THOR96 SRT-Vision-AR0430 mezzanine setup

- Make sure the DTB environment is set with **imx8mq-thor96-ap1302-ar0430.dtb**
- Power up the board and go to terminal of x86 host system and open board's console
- Hold boot on u-boot screen by pressing any key on host machine keyboard (immediate after boot within 3 seconds)
- Set DTB environment to **imx8mq-thor96-ap1302-ar0430.dtb** using following command:

```
$ u-boot > setenv fdt_file imx8mq-thor96-ap1302-ar0430.dtb
$ u-boot > saveenv
$ u-boot > boot
```

- From HDMI display, open command prompt of the target
- Then apply below command in the command prompt of the target.

```
$ gst-launch-1.0 v4l2src device=/dev/video0 ! glimagesink render-rectangle='<0, 0, 1920, 1080>'
sync=false
```

- And user will be able to see camera live stream on HDMI display as following:

Figure 5: AR0430 Live stream in HDMI Display

3.2 Live stream of ARX3A0 Camera on HDMI display

- To watch live stream over the HDMI, connect HDMI Display, USB keyboard and Mouse to THOR96 board
- Attach SRT-Vision96-ARX3A0 mezzanine (AP-vision-ARX3A0) camera mezzanine module to high-speed expansion connector as shown in the following Figure:

Figure 6 : THOR96 SRT-Vision-ARX3A0 mezzanine setup

- Make sure the DTB environment is set with imx8mq-thor96-ap1302-arx3a0.dtb

- Power up the board and go to terminal of x86 host system and open board's console
- Hold boot on u-boot screen by pressing any key on host machine keyboard (immediate after boot within 3 seconds)
- Set DTB environment to **imx8mq-thor96-ap1302-arx3a0.dtb using following command:**

```
$ u-boot > setenv fdt_file imx8mq-thor96-ap1302-arx3a0.dtb
$ u-boot > saveenv
$ u-boot > boot
```

- From HDMI display, open command prompt of the target
- Then apply below command in the command prompt of the target.

```
$ gst-launch-1.0 v4l2src device=/dev/video0 ! glimagesink render-rectangle='<0, 0, 1920, 1080>' sync=false
```

- And user will be able to see camera live stream on HDMI display as following:

Figure 7 : ARX3A0 Live Stream on HDMI Display

3.3 Capture image from camera

- To capture image from the camera, connect HDMI Display, USB keyboard and Mouse to Thor96 development board
- Configure appropriate DTBs either for AR0430 or ARX3A0 camera sensor as mentioned above
- Attach any one of the below mentioned camera-mezzanine modules to the Thor96 board
  - SRT-Vision96-AR0430 mezzanine (AP-vision-AR0430) – Refer *Figure 4*
  - SRT-Vision96-ARX3A0 mezzanine (AP-vision-ARX3A0) – Refer *Figure 6*
- Ensure Ethernet is plugged-in to get image from board to local x86 host pc
- Power up the board
- From HDMI display, open command prompt of the target
- Then apply following command in the command prompt of the target:

```
$ gst-launch-1.0 v4l2src device=/dev/video0 num-buffers=1 ! jpegenc ! filesink location=test0.jpg
```

- Above command will capture image named test0.jpg in /home/root/ location
- Copy image from board to local PC using below command

```
$ scp test0.jpg <user name of host pc >@<ip of host pc>:/home/<user>/Desktop
```

- Go to local PC's /home/user/Desktop and watch image into image viewer to verify captured image from board's camera

- **AR0430 Captured image:**

Figure 8 : AR0430 Captured image

- ARX3A0 Captured Image:

Figure 9: ARX3A0 Captured image

3.4 Validation of ISP controlled feature of AP1302:

User can check the available v4l2 controls related to the camera from the user space using the following command:

```
root@imx8mqthor96:~# v4l2-ctl -L

User Controls

                brightness 0x00980900 (int)    : min=0 max=65535 step=1 default=0 value=0
                  contrast 0x00980901 (int)    : min=0 max=65535 step=1 default=0 value=0
                saturation 0x00980902 (int)    : min=0 max=65535 step=1 default=4096 value=4096
                     gamma 0x00980910 (int)    : min=0 max=65535 step=1 default=0 value=0
                  exposure 0x00980911 (int)    : min=0 max=12 step=1 default=12 value=12
                      gain 0x00980913 (int)    : min=0 max=65535 step=1 default=256 value=256

Camera Controls

    white_balance_auto_preset 0x009a0914 (menu)   : min=0 max=1 default=1 value=1
                              0: Manual
                              1: Auto
```

Note: Above output is taken using AR0430 Camera sensor.

User will be able to get and set the values for each of the controls using following format.

- To get control value, Run the following command:

```
$ v4l2-ctl --get-ctrl <control_name>
```

- To set control value, Run the following command:

```
$ v4l2-ctl --set-ctrl <control_name> =<control_value>
```

3.4.1 Resolution, FPS and Data format

User can check the current resolution, FPS, and data format of AP1302 using the following command.

- AR0430:

```
root@imx8mqthor96: ~# v4l2-ctl --list-formats-ext -d /dev/video0
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture

        [0]: 'UYVY' (UYVY 4:2:2)
                Size: Discrete 2316x1746
                        Interval: Discrete 0.033s (30.000 fps)
```

- ARX3A0:

```
root@imx8mqthor96:~# v4l2-ctl --list-formats-ext -d /dev/video0
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture

        [0]: 'UYVY' (UYVY 4:2:2)
                Size: Discrete 600x600
                        Interval: Discrete 0.008s (120.000 fps)
```

3.4.2 Brightness

- To get the brightness value, Run the following command:

```
$ v4l2-ctl --get-ctrl brightness
```

- To set the brightness value, Run the following command:

```
$ v4l2-ctl --set-ctrl brightness=4096
```

3.4.3 Contrast

- To get the brightness value use the following command:

```
$ v4l2-ctl --get-ctrl contrast
```

- To set the brightness value use the following command:

```
$ v4l2-ctl --set-ctrl contrast=4096
```

3.4.4 Saturation

- To get the saturation value use the following command:

```
$ v4l2-ctl --get-ctrl saturation
```

- To set the saturation value use the following command:

```
$ v4l2-ctl --set-ctrl saturation=4096
```

3.4.5 Gamma

- To get the saturation value use the following command:

```
$ v4l2-ctl --get-ctrl gamma
```

- To set the gamma value use the following command:

```
$ v4l2-ctl --set-ctrl gamma=4096
```

3.4.6 Exposure

- To get the exposure value use the following command:

```
$ v4l2-ctl --get-ctrl exposure
```

- To set the exposure value use the following command:

```
$ v4l2-ctl --set-ctrl exposure=12
```

3.4.7 Gain

- To get the gain value use the following command:

```
$ v4l2-ctl --get-ctrl gain
```

- To set the gain value use the following command:

```
$ v4l2-ctl --set-ctrl gain=4096
```

3.4.8 White Balance

- To get the White balance value use the following command:

```
$ v4l2-ctl --get-ctrl white_balance_auto_preset
```

- To set the White balance value use the following command:

```
$ v4l2-ctl --set-ctrl white_balance_auto_preset =1
```

3.5 AP1302 Register Read / Write

User can read / write AP1302 registers directly through the SysFs interface provided by the AP1302 driver.

- The SysFs interfaces are available in the following path
  PATH: /sys/kernel/debug/ap1302.B-00SS/
  B – I2C Bus Number
  SS – I2C slave address
- User can write / read the address through reg_addr interface
- User can write / read the data through reg_data interface

3.5.1 Read AP1302 Registers:

- Write the address of the AP1302 register from which the data has to be read

```
$ echo 0xY00ZZZZ > /sys/kernel/debug/ap1302.B-00SS/reg_addr
```

Y - Type of register - 2 for 16bit,
4 for 32-bit
ZZZZ - Register address

- Read the value from the above written register address of AP1302

```
$ cat /sys/kernel/debug/ap1302.B-00SS/reg_data
```

Example:

- To write the address of register R0x7000 use the following command:

```
$ echo 0x2007000 > /sys/kernel/debug/ap1302.1-003d/reg_addr
```

- To read value from the register R0x7000 use the following command:

```
$ cat /sys/kernel/debug/ap1302.1-003d/reg_data
0x00000064
```

3.5.2 Write AP1302 Registers:

- Write the address of the AP1302 register to which the data has to be written

```
$ echo 0xY00ZZZZ > /sys/kernel/debug/ap1302.B-00SS/reg_addr
```

Y - Type of register - 2 for 16bit,
4 for 32-bit
ZZZZ - Register address

- Write the value to above written AP1302 register address

```
$ echo 0xXXXXXXXX > /sys/kernel/debug/ap1302.B-00SS/reg_data
```

XXXXXXXX is register address value which user wants to set.

Example:

- To write the address of register R0x7000 use the following command:

```
$ echo 0x2007000 > /sys/kernel/debug/ap1302.1-003d/reg_addr
```

- To write the value to the register R0x7000 use the following command:

```
$ echo 0x00000064 >  /sys/kernel/debug/ap1302.1-003d/reg_data
```

4 LIMITATION

NA