地 平 线
Horizon Robotics

X3

MU-2520-2-X3 Platform

System Software User Manual

Rev. 1.0.0

2021-02

Important Notice and Disclaimer

Information in this document is provided solely to enable system and software implementers to use Horizon products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

All statements, information and recommendations in this document are provided "AS IS". Horizon makes no warranty, representation or guarantee of any kind, express or implied, regarding the merchantability, fitness or suitability of its products for any particular purpose, and non-infringement of any third party intellectual property rights, nor does Horizon assume any liability arising out of the application or use of any product, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

"Typical" parameters that may be provided in Horizon datasheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Buyers and others who are developing systems that incorporate Horizon products (collectively, "Users") understand and agree that Users shall remain responsible for using independent analysis, evaluation and judgment in designing their applications and that Users have full and exclusive responsibility to assure the safety of Users' applications and compliance of their applications (and of all Horizon products used in or for Users' applications) with all applicable regulations, laws and other applicable requirements.

User agrees to fully indemnify Horizon and its representatives against any claims, damages, costs, losses and/or liabilities arising out of User's unauthorized application of Horizon products and non-compliance with any terms of this notice.

Horizon Robotics, Inc.

https://www.horizon.ai

Revision History

Time	Version	Details
2020.06	V0.5	Document Created
2020.08	V0.5.2	Updated Camera test application command; Corrected typo
2021.02	V1.0.0	Release 1.0

Contents

Cautions

1 Quick Deployment Guide

This quick deployment guide will introduce the initial deployment process of development boards. The detailed description of system software usage is presented in the remaining chapters.

1.1 PC Environment Setup

1.1.1 Serial/Ethernet Connection Software

The PC connecting to development board must be equipped with software that can read and write through serial port or through Ethernet port via telnet/ssh protocol.

Serial Port Baud rate: 115200/921600

1.1.2 Horizon Upgrade Tool

Horizon Upgrade tool will be used to program image onto SOM on development boards.

1.1.3 tftp Server Setup and File transfer via tftp Protocol

1.1.3.1 Windows tftp Server

Use "tftp_win32" or any tftp server application on Windows preferred

1.1.3.2 Linux tftp Server

```
sudo apt-get install tftpd-hpa
sudo mkdir  /tftpboot
# Edit tftpd-hpa Configuration
sudo vi /etc/default/tftpd-hpa
```

After inside vi, input the following configuration in the file. The root directory of tftp server will be "/tftpboot"

```
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/tftpboot"
TFTP_ADDRESS="[::]:69"
```

```
TFTP_OPTIONS="-l -c -s"
```

Configure IP address to be 192.168.1.1:

```
ifconfig 192.168.1.1
```

Restart tftp server and apply configuration:

```
sudo service tftpd-hpa restart
```

Put the image file relevant to kernel to directory "/tftpboot"

1.1.3.3 Download via tftp from Board

1.1.3.3.1 In Uboot

```
setenv serverip 192.168.1.1
setenv ipaddr 192.168.1.10
tftp <target addr> <relative file path>
```

1.1.3.3.2 In Kernel

```
Usage: tftp [OPTIONS] HOST [PORT]
Transfer a file from/to tftp server
       -l FILE Local FILE
       -r FILE Remote FILE
       -g      Get file
       -p      Put file
```

For Example:

```
# Download from PC
tftp -g -r <relative path to file in the server directory> 192.168.1.1
# Send file to PC
tftp -p -l <path to file> 192.168.1.1
```

1.1.4 NFS Server Setup and Transfer File via NFS

1.1.4.1 Windows nfs Server

Use any nfs application on Windows preferred to set up nfs server on Windows.

1.1.4.2 Linux nfs Server

Install nfs:

```
sudo apt-get install nfs-kernel-server
```

Configure nfs:

```
cd ~
mkdir nfs
sudo vim /etc/exports
/home/gs/nfs 192.168.*.*(sync,rw,no_root_squash)
sudo exportfs -a
```

Restart rpcbind Service

```
sudo service rpcbind restart
```

Restart nfs Service

```
sudo service nfs-kernel-server restart
```

1.1.4.3 Mount nfs Server from Board

Configure IP Address:

```
ifconfig eth0 192.168.1.10 netmask 255.255.255.0
```

Mount PC nfs Server Folder to "/mnt/nfs":

```
mkdir -p /mnt/nfs
mount -t nfs -o nolock 192.168.1.1:/home/gs/nfs /mnt/nfs
```

After mounting the folder successfully, the files in the folder on PC can be read/write/execute (after permission is configured correctly) from board directly

1.2 X3DVB

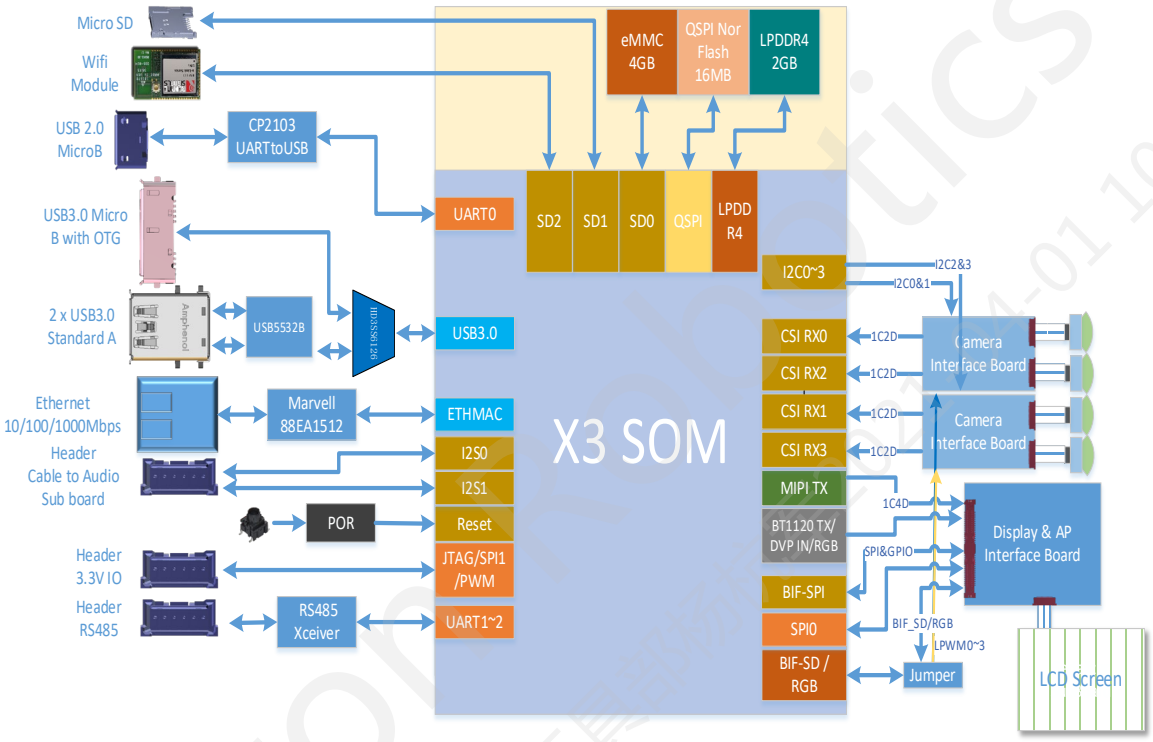
1.2.1 Cautions

Refer to: *Cautions*

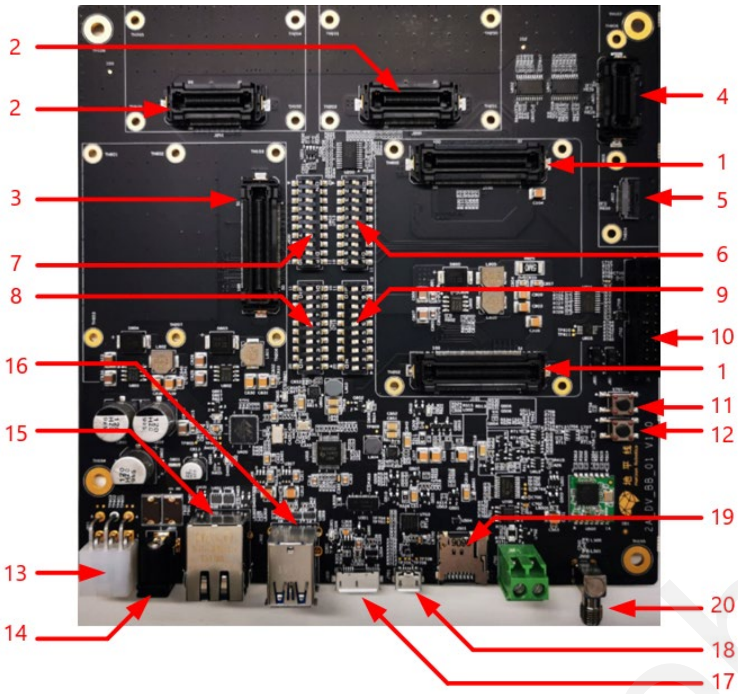
1.2.2 Hardware Setup

1.2.2.1 Development Board Introduction

1.2.2.1.1 System Design Topology



1.2.2.1.2 Image of X3DVB

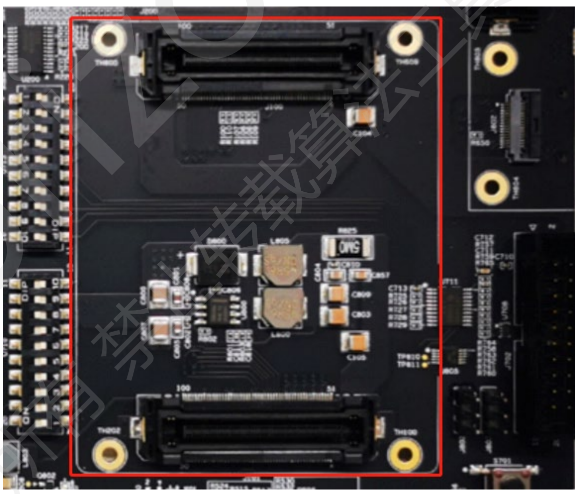


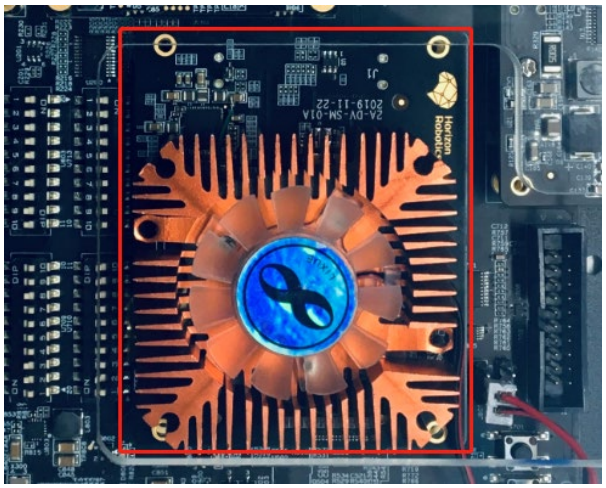
1	SOM Socket
2	Camera Socket
3	Display Socket
4	Audio Socket for Audio Plug of X3 DVB
5	Audio Connector for Audio Board of BOOTPRINT
6	JTAG Bypass Switch
7	BIFSD Bypass Switch
8	Boot Strap Switch 1
9	Boot Strap Switch 2 Fan Power Switch USB Select Switch
10	JTAG Connector
11	Reset Key
12	Wake Up Key
13	Power Connector 6 Pin
14	Power Jack 2.5/5.5mm
15	Ethernet Connector RJ45
16	USB 3.0 Standard-A x2
17	USB 3.0 Micro-AB
18	Debug Uart - Micro-USB 2.0
19	Micro SD Card Connector
20	WIFI Antenna Connector SMA-F

1.2.2.2 X3DVB Hardware Setup

Please setup X3-DVB according to the following procedures:

- Attach heat sink to X3-SOM board. Plug the X3-SOM board to the SOM socket indicated in the above image, below is a enlarged view of the SOM socket;





X3SOM comes with anti-reverse insertion protection, but proceed with caution.

- Locate the Micro-USB socket in the location demonstrated in the above image and plug in the Micro-USB cable. The other end of the Micro-USB cable should connect to the USB port to the PC you are using;
- Locate the Ethernet port in the location demonstrated in the above image, plug in Ethernet cable comply with RGMII. The other end of the Ethernet cable should connect to the PC you are using;
- If needed, locate the Micro SD card slot at the location demonstrated in the above image. Insert the Micro SD card.

1.2.2.3 Camera Sensor and Display Board Setup

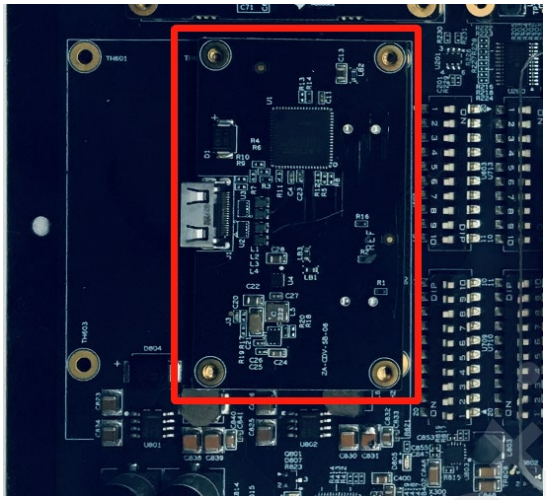
If needed, set up the display board and camera by the following steps:

1.2.2.3.1 Display Board

All display boards are equipped with anti-reverse insertion protection. In the above image, the display board connected is X2-LCD Display board.

1.2.2.3.2 HDMI Display Board

Locate the Display Board socket in the location above, insert X3-HDMI display board，demonstrated be low is the correct installation of HDMI display board;



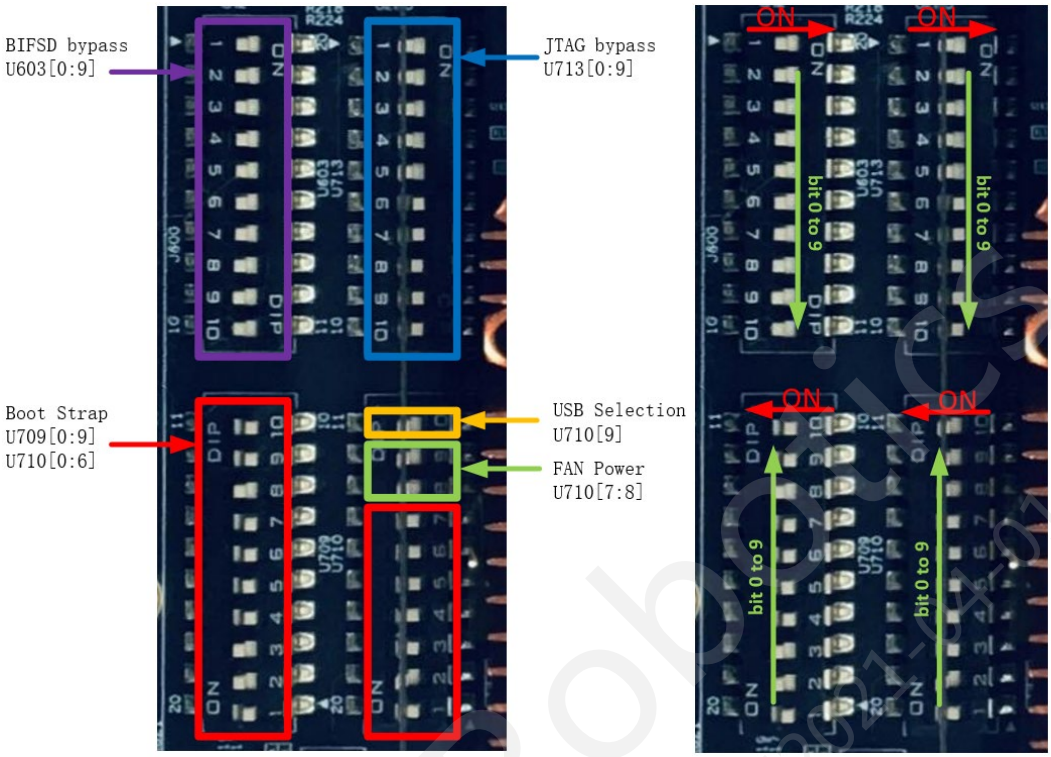
1.2.2.3.3 LCD Display Board

Locate the Display Board socket in the location above, insert X3-LCD display board. Demonstrated below is the correct connection of the LCD display board and the LCD screen;

1.2.2.3.4 Camera Sensor Setup

1.2.2.4 X3-DVB Boot Mode Configuration

X3 DVB is equipped with 4 DIP switches, among which switch No.3 is the Boot mode configuration switch. As demonstrated below, DIP switch No.3 is located in the bottom left corner, the switch on/off position is demonstrated in the picture on the right:



The detailed Boot mode configuration is shown below, the default position of DIP switch 3 should be all off:

Function	Switches Involved	Switch Status and Meaning
DDR Identifier for DDR Identification	Switch 1	0: Hynix 1: Micron
UART Baud Rate	Switch 8	0: 921600 1: 115200
UART Boot	Switch 2，3，4	Switch 2: 1 Switch 3: 1 Switch 4: 0
eMMC Boot	Switch 2，3，4	Switch 2: 0 Switch 3: 0 Switch 4: 0
USB Boot	Switch2，3，4	Switch 2: 0 Switch 3: 0 Switch 4: 1
NOR Boot	Switch 2，3，4	Switch 2: 1 Switch 3: 0 Switch 4: 1

NAND Boot	Switch 2，3，4	Switch 2: 1 Switch 3: 0 Switch 4: 0

The functions of other DIP switches are listed in X3DVB User Manual.

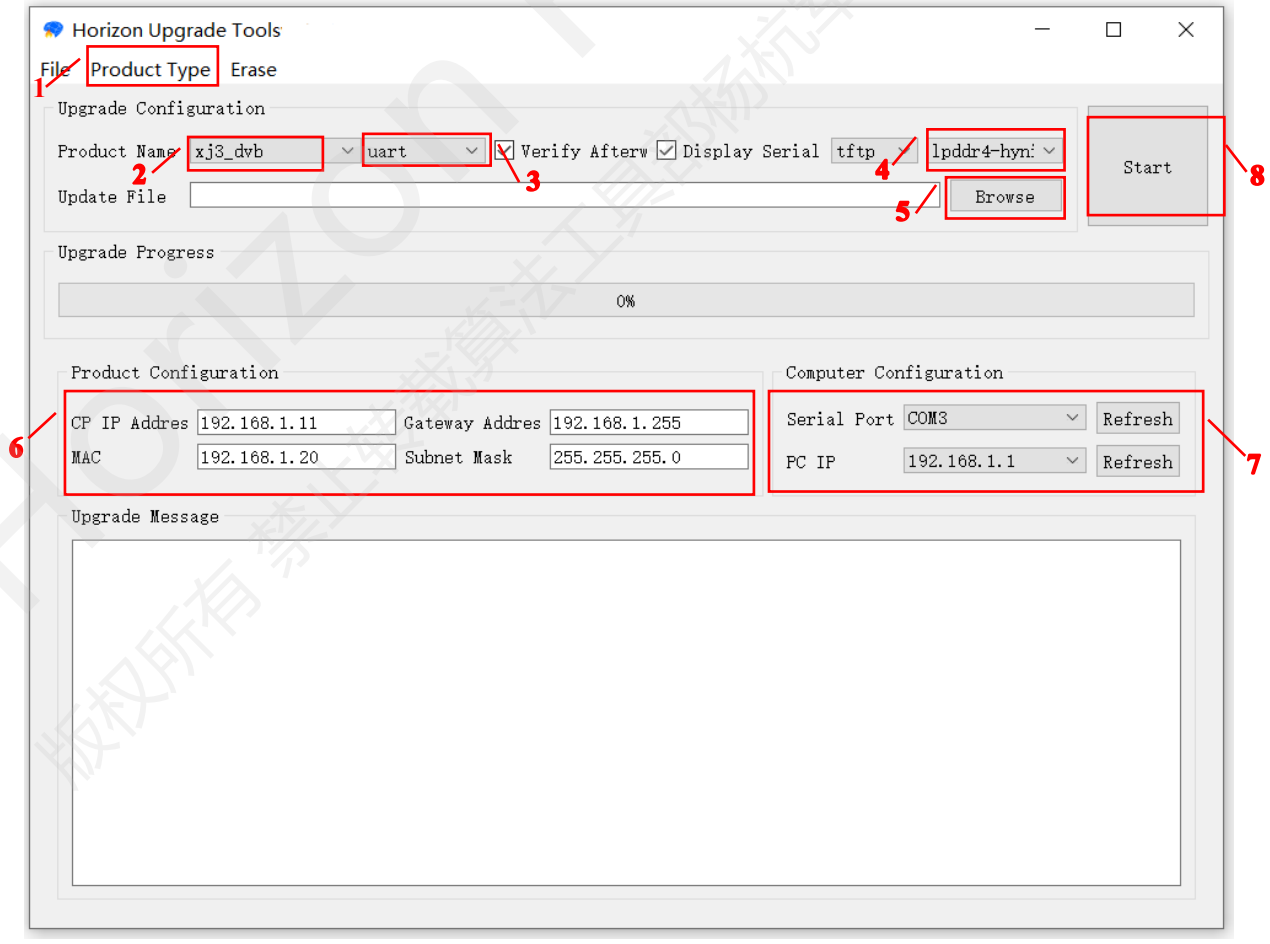
1.2.3 Burn Image

Burning image to development boards using Horizon Upgrade tool is recommended.

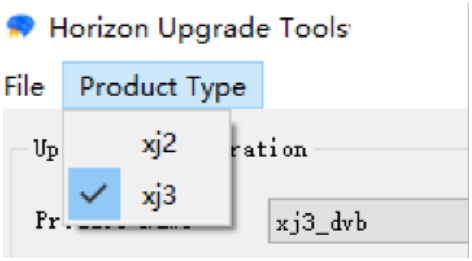
1.2.3.1 UART Mode

UART Mode is suitable for boards powered up for the first time, SOM boards with no functional system or boards with corrupted systems which can no longer boot to U-Boot stage. Burning image in UART mode is time-consuming.

1.2.3.2 Upgrade Tool Configuration



1. As shown above, click "Product Type" and choose "xj3":



2. As shown above, choose "xj3_dvb" for Product Name
3. As shown above, choose "uart" for upgrade mode
4. As shown above, choose the DDR manufacturer of SOM board using the dropdown list on top of "Browse" button
5. As shown above, click "Browse" button, select the upgrade image file. After the upgrade file is selected, the path to the upgrade file will be displayed in the text box to the left of the button
6. Configure Product Configuration, the IP address of CP should be in the same network segment as the PC it's connected to. Please proceed with cautions with the other network configurations. Please refer to update tool manual for details
7. Configure PC configuration, Serial Port should be the COM port appeared right after the X3-DVB is powered on and connected to PC via USB port. The IP Address of PC should be in the same network segment as the IP address of CP.
8. Click "Start" button to start the upgrade process

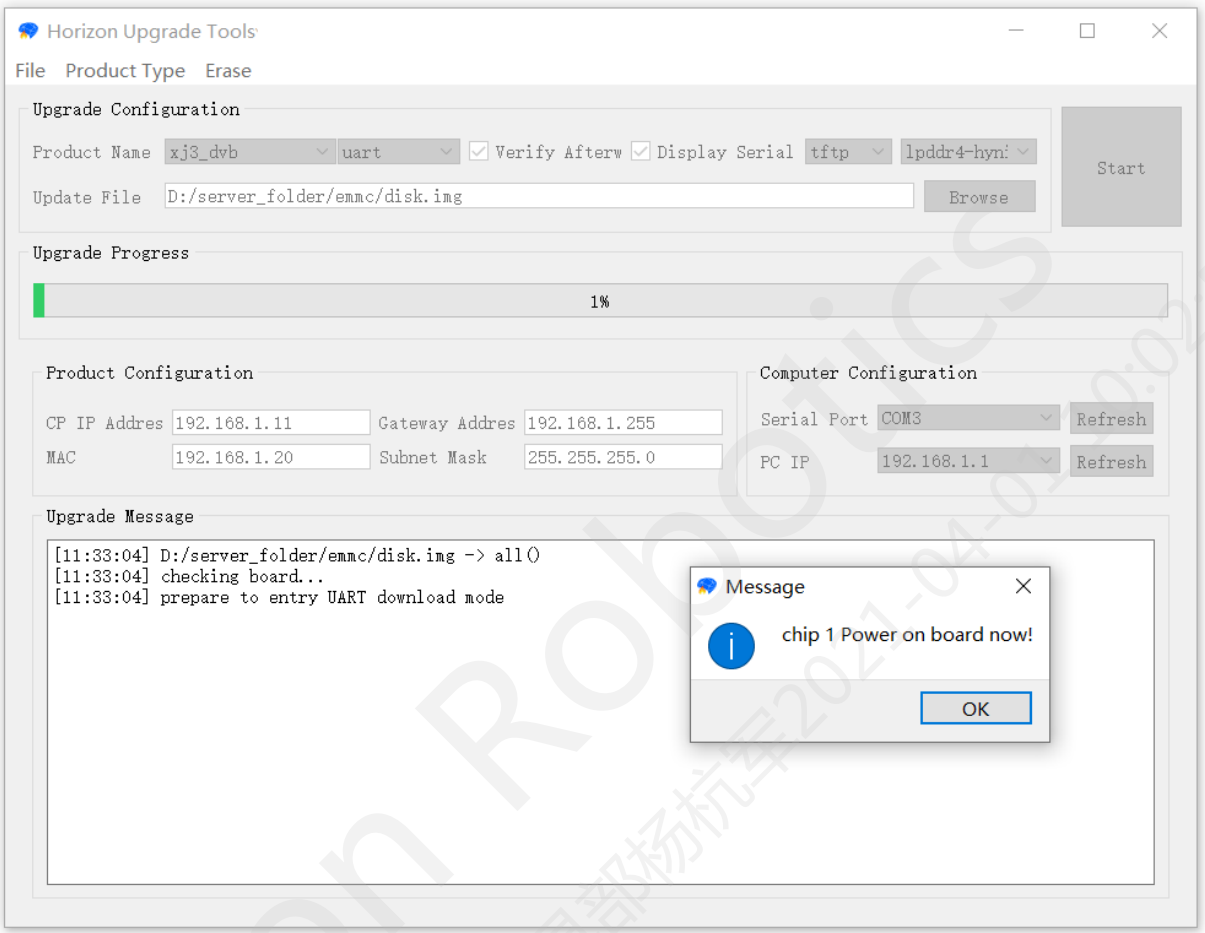
Note: When "Display Serial Port Output" option is selected, the output of serial port will be printed out to the text box "upgrade message" which can be used for debugging.

1.2.3.3 Burn Image Procedure

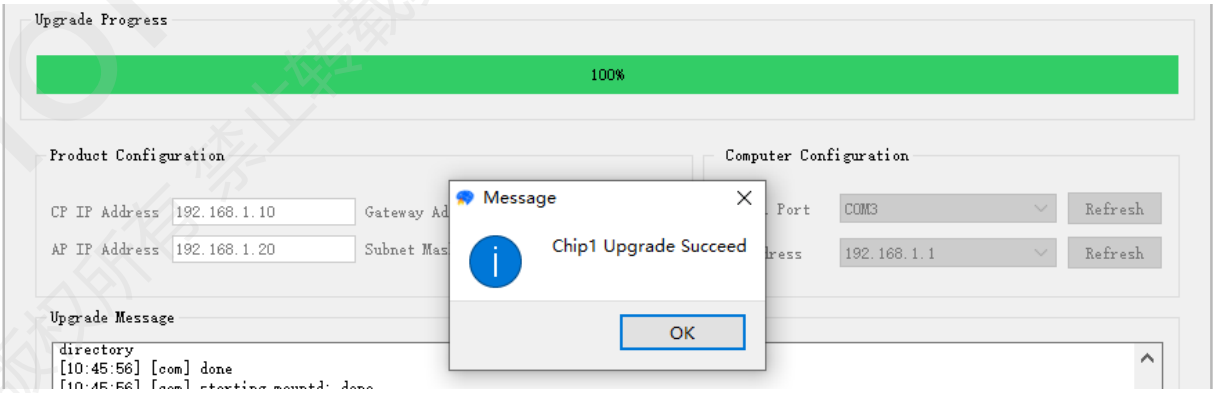
1. Before upgrade starts, **power OFF** the development board. Setup the board and Horizon Upgrade tool according to the previous manual.
2. Click "Start" button, and the following prompt should be shown:

3. Confirm that the development board is powered OFF, click "OK" button and the

following prompt and print should appear:



4. After powering ON the development board, click the "OK" button and wait until the upgrade process is completed.
5. After the upgrade has completed, the following should appear:



1.2.3.4 U-Boot Mode

U-Boot Mode is designed for systems that can boot to U-Boot stage. Except for

"upgrade mode" ought to be configured as "uboot", the other configurations and burning steps are the same as those mentioned in "uart" mode.

1.3 System Check After Boot

1.3.1 Check System Version

Use the following command to check current system version:

```
cat /etc/version
```

1.3.2 Check BPU Usage

Use System Tool hrut_bpuprofile" to check the usage of BPU:

-b: specifies the BPU checking, 0 - BPU0; 1 - BPU1; 2 - BPU0+BPU1

-r: specifies the times checked, 0 is unlimited

For more details, refer to: *hrut_bpuprofile*

```
hrut_bpuprofile -b 2 -r 0
```

2 Horizon Upgrade Tool

2.1 Abstract

X3-DVB comes with CP2103 "USB to Serial" adapting chip. Install the corresponding USB driver and set baud rate to 115200bps/921600bps.

2.2 Windows Version

Unzip hbupdate_win64_v*.zip, execute the unzipped executable "hbupdate.exe". Logs will be stored under "log" directory.

Upgrade Process:

1. Choose product type
2. Choose upgrade mode
3. Configure options for PC and Product connection: serial port, IP address etc.
4. Choose Upgrade file
5. Start Upgrade

2.3 MAC Version

Unzip hbupdate_MAC_v*.zip and drag the unzipped app file to "Applications" (Changing settings in "Settings -> Security and Privacy" to allow installation might be needed). If the app would not launch or "Permission Denied" message is displayed, execute the application in terminal with "sudo" privilege:

```
sudo /Applications/hbuppdate.app/Contents/MacOS/hbupdate
```

Logs will be stored in the directory "/Applications/hbupdate.app/Contents/MacOS/hbupdate/log"

2.4 FAQ

1) Read the "Horizon Upgrade Tool User Manual" before upgrading any boards
2) Before upgrade starts, make sure tftp service is properly set up and serial port to the board is NOT in use
3) File name of the upgrade image should contains only English characters and underscore(without any spaces or tabs)
4) To achieve optimal upgrade speed and stability, connect your board with PC via cable instead of wireless connection

5) If upgrade/boot failed, retry using "uart" mode with full image (disk.img)
6) Make sure PC network card connecting the board is setup properly under the same subnet with the board
7) If network connection fails repetitively, please turn off firewall and try again

2.5 Horizon Upgrade Tool

Refer to "Horizon Upgrade Tool User Manual" for details

3 Serial port and multiple console support

X3 support serial port connection. System defaults to log onto serial console. (Use SecureCRT/minicom to connect to X2J2 from PC) System is integrated with "screen" to support multiple console from serial connection.

Note: Under minicom, Ctrl-A overlaps the hotkey of screen. Please double tab Ctrl-A or change minicom configuration using Ctrl-A + O.

3.1 screen Manual

```
# screen [-AmRvx -ls -wipe][-d <session name>][-h <line number>][-r <session name>][-s ][-S <session name>]
```

3.1.1 Synopsis

```
# screen [-AmRvx -ls -wipe][-d <session>][-h <line>][-r <session>][-s ][-S <session>]
Command line options
-A adjust all windows to fit current terminal
-d <session> disconnect specified session
-h <num> specifies the history scroll-back buffer to be num lines high
-m creation of a new session is enforced, regardless whether screen is called from within another screen session or not
-r <pid.tty.host> resumes a detached screen session
-R attempts to resume the first detached screen session it finds. If no detached session exists, starts a new session using the specified options
-s sets the default shell to the program specified.
-S <sessionname> specify a meaningful name for the session.
-v Print version info
-x Attach to a not detached screen session. (Multi display mode).
-ls or –list prints a list of pid.tty.host strings identifying your screen sessions
-wipe does the same as "screen -ls", but removes destroyed sessions instead of marking them as 'dead'
```

More options please use command line help to check:

```
screen -h
```

```
root@x3dvbx3-micron4G-2666:~# screen -h
Use: screen [-opts] [cmd [args]]
 or: screen -r [host.tty]

Options:
-a            Force all capabilities into each window's termcap.
-A -[r|R]     Adapt all windows to the new display width & height.
-c file       Read configuration file instead of '.screenrc'.
-d (-r)       Detach the elsewhere running screen (and reattach here).
-dmS name     Start as daemon: Screen session in detached mode.
-D (-r)       Detach and logout remote (and reattach here).
-D -RR        Do whatever is needed to get a screen session.
-e xy         Change command characters.
-f            Flow control on, -fn = off, -fa = auto.
-h lines      Set the size of the scrollback history buffer.
-i            Interrupt output sooner when flow control is on.
-l            Login mode on (update /var/run/utmp), -ln = off.
-ls [match]   or
-list         Do nothing, just list our SockDir [on possible matches].
-L            Turn on output logging.
-Logfile file Set logfile name.
-m            ignore $STY variable, do create a new screen session.
-O            Choose optimal output rather than exact vt100 emulation.
-p window     Preselect the named window if it exists.
-q            Quiet startup. Exits with non-zero return code if unsuccessful.
-Q            Commands will send the response to the stdout of the querying process.
-r [session]  Reattach to a detached screen process.
-R            Reattach if possible, otherwise start a new session.
-s shell      Shell to execute rather than $SHELL.
-S sockname   Name this session <pid>.sockname instead of <pid>.<tty>.<host>.
-t title      Set title. (window's name).
-T term       Use term as $TERM for windows, rather than "screen".
-U            Tell screen to use UTF-8 encoding.
-v            Print "Screen version 4.06.02 (GNU) 23-Oct-17".
-wipe [match] Do nothing, just clean up SockDir [on possible matches].
-x            Attach to a not detached screen. (Multi display mode).
-X            Execute <cmd> as a screen command in the specified session.
```

3.2 Examples

3.2.1 Create a New screen Session

```
screen -S Hobot
```

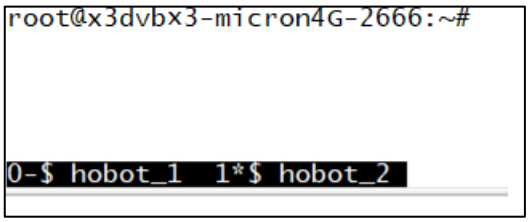
After screen has launched, window No.0 is created with system default shell running (usually bash). Thus, after you used screen command, you will return to command line prompt immediately.

When you are in screen, all commands start with (Ctrl-A). Use (Ctrl-A+?) to check the information of all hotkey bindings.

```
                         Screen key bindings, page 1 of 1.

                         Command key:  ^A   Literal ^A:  a

  break       ^B b       history     { }          other       ^A           split       S
  clear       C          info        i            pow_break   B            suspend     ^Z z
  colon       :          kill        ^K k         pow_detach  D            time        ^T t
  copy        ^[ [       lastmsg     ^M m         prev        ^H ^P p ^?   title       A
  detach      ^D d       license     ,            quit        ^\           vbell       ^G
  digraph     ^V         lockscreen  ^X x         readbuf     <            version     v
  displays    *          log         H            redisplay   ^L l         width       W
  dumptermcap .          login       L            remove      X            windows     ^W w
  fit         F          meta        a            removebuf   =            wrap        ^R r
  flow        ^F f       monitor     M            reset       Z            writebuf    >
  focus       ^I         next        ^@ ^N sp n   screen      ^C c         xoff        ^S s
  hardcopy    h          number      N            select      '            xon         ^Q q
  help        ?          only        Q            silence     _

^]   paste .
"    windowlist -b
-    select -
0    select 0
1    select 1
2    select 2
3    select 3
4    select 4
5    select 5
6    select 6
7    select 7
8    select 8
9    select 9
]    paste .
|    split -v
:kB: focus prev
```

3.2.2 Create a New sub-Session in Screen

Use Ctrl-A + C to create new sub-session, then use Ctrl-A + W to print a list of current sub-sessions:



3.2.3 Switch between sub-Sessions

Use (Ctrl-A Ctrl-A) to switch between the two most recently used sub-session.

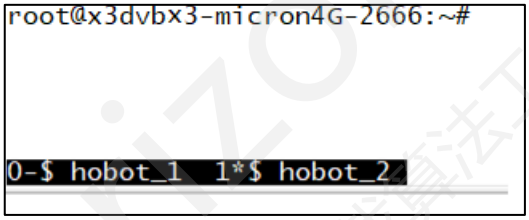
Use (Ctrl-A + P) (Ctrl-A + N) to switch to the previous/next sub-session accordingly

Use (Ctrl-A + A) to set the name of current sub-session

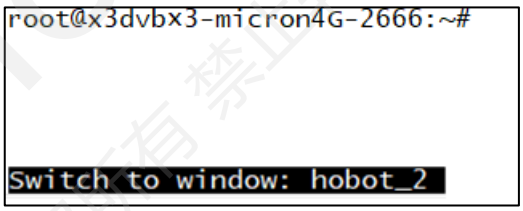
Use (Ctrl-A + ') to switch to a sub-session by name

For examples:

List of sub-sessions after name changed:



Switch to window by name:



3.2.4 Splitting Window

```
Ctrl-A + S          ### Split window horizontally
Ctrl-A + |          ### Split window vertically
Ctrl-A + <tab>      ### Switch between split windows
```

```
Ctrl-A + X            ### Close the current window
Ctrl-A + Q            ### Close all but current window
```

4 Use gdbserver to Debug Application

System is integrated with gdbserver, follow the below steps to setup:

4.1 Preparation

4.1.1 Confirm IP Address of the Board

Make sure the internet connection between the board and PC is up and running by pinging:

```
root@x3dvbx3-micron4G-2666:/mnt/gdbserver_example# ping 192.168.1.1 -c 5
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=128 time=0.982 ms
64 bytes from 192.168.1.1: seq=1 ttl=128 time=0.493 ms
64 bytes from 192.168.1.1: seq=2 ttl=128 time=0.623 ms
64 bytes from 192.168.1.1: seq=3 ttl=128 time=0.351 ms
64 bytes from 192.168.1.1: seq=4 ttl=128 time=0.392 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.351/0.568/0.982 ms
```

```
yu@Horizon:~$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.320 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.488 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.398 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=0.427 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=64 time=0.314 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=64 time=0.314 ms
64 bytes from 192.168.1.10: icmp_seq=6 ttl=64 time=0.537 ms
64 bytes from 192.168.1.10: icmp_seq=6 ttl=64 time=0.537 ms
64 bytes from 192.168.1.10: icmp_seq=7 ttl=64 time=0.344 ms
64 bytes from 192.168.1.10: icmp_seq=8 ttl=64 time=1.12 ms
64 bytes from 192.168.1.10: icmp_seq=9 ttl=64 time=0.842 ms
^C
--- 192.168.1.10 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8149ms
rtt min/avg/max/mdev = 0.314/0.532/1.124/0.260 ms
yu@Horizon:~$
```

4.1.2 Test Application

Write test application on PC and compile it with -g. e.g.: hello.c

```
#include<stdio.h>

int main()
{
    printf("hello world");
    getchar();
    return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"hello.c" [noeol] 8L, 89C                                    1,1           All
```

Compiling:

```
yu@Horizon:/mnt/pc/gdbserver_example$ aarch64-linux-gnu-gcc -g hello.c -o hello.out
yu@Horizon:/mnt/pc/gdbserver_example$
```

4.2 Configure Board

Use tftp or scp to copy application onto the board or alternatively, use nfs to mount PC directory containing the application. Execute gdbserver and wait for PC connection:

```
gdbserver :8000  <myapp>
```

```
root@x3dvbx3-micron4G-2666:/mnt/gdbserver_example# gdbserver :8000 hello.out
Process /mnt/gdbserver_example/hello.out created; pid = 1931
Listening on port 8000
```

4.3 PC configuration

4.3.1 Connect PC to board via gdb

Run gdb on PC and use "target remote <board IP>:<port>" to connect to board.

```
target remote 192.168.1.10:8080
```

```
yu@Horizon:/mnt/pc/gdbserver_example$ aarch64-linux-gnu-gdb
GNU gdb (Linaro_GDB-2018.12) 8.2.1.20190107-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-unknown-linux-gnu --target=aarch64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word".
(gdb) target remote 192.168.1.10:8000
Remote debugging using 192.168.1.10:8000
Reading /mnt/gdbserver_example/hello.out from remote target...
warning: File transfers from remote targets can be slow. Use "set sysroot" to access files locally instead.
Reading /mnt/gdbserver_example/hello.out from remote target...
Reading symbols from target:/mnt/gdbserver_example/hello.out...done.
Reading /lib/ld-linux-aarch64.so.1 from remote target...
Reading /lib/ld-linux-aarch64.so.1 from remote target...
Reading symbols from target:/lib/ld-linux-aarch64.so.1...Reading /lib/ld-2.28.so from remote target...
Reading /lib/.debug/ld-2.28.so from remote target...
(no debugging symbols found)...done.
0x0000007fbf6d3100 in ?? () from target:/lib/ld-linux-aarch64.so.1
(gdb)
```

4.3.2 Debug

Use gdb to debug:

```
0x0000007fbf6d3100 in ?? () from target:/lib/ld-linux-aarch64.so.1
(gdb) l
1       #include<stdio.h>
2
3       int main()
4       {                                                    PC
5           printf("hello world");
6           getchar();
7           return 0;
8       }(gdb) b 6
Breakpoint 1 at 0x4005e4: file hello.c, line 6.
(gdb) c
Continuing.
Reading /lib/libc.so.6 from remote target...
Reading /lib/libc-2.28.so from remote target...
Reading /lib/.debug/libc-2.28.so from remote target...

Breakpoint 1, main () at hello.c:6
6           getchar();
(gdb) n
```

```
root@x3dvbx3-micron4G-2666:/mnt/gdbserver_example#  gdbserver :8000 hello.out
Process /mnt/gdbserver_example/hello.out created; pid = 1491
Listening on port 8000
Remote debugging from host 192.168.1.1
hello world                                          Board
```

5 Frequently Used Tools

5.1 hrut_utilities

This is a toolset used on CP to set/collect system info

5.1.1 hrut_boardid

Used to check the boardid of current board. Detailed list is attached below:

	Definition	Length	Value Range
Model	DDR Manufacturer	4Bits	0x0: Auto-Detection 0x1:海力士 0x2:镁光 0x3:三星
DDR Type	DDR Model	4Bits	0x0: Auto-Detection 0x1: LPDDR4 0x2: LPDDR4X 0x3: DDR4 0x4: DDR3L
Frequency	DDR Frequency	4Bits	0x0: Auto-Detection 0x1: 667 0x2: 1600 0x3: 2133 0x4: 2666 0x5: 3200 0x6: 3733 0x7: 4266
Capacity	DDR Size	4Bits	0x0: Auto-Detection 0x1: 1GB 0x2: 2GB 0x3: 4GB
ECC	ECC Configuration	4Bits	0x0: Default 0x1: inline ECC All 0x2: inline ECC option1 0x3: inline ECC option2
SOM Type	SOM Model	4Bits	0x0: Auto-Detection 0x1: X3

			0x2: J3
Alternative	Alternation Configurations	4Bits	0x0: Default 0x1: Config1
Board Type	Development Board Type	4Bits	0x0: Auto-Detection 0x1: X3 DVB 0x2: J3 DVB 0x3: CVB

This utility can also be used to manually change Boardid

Note:

Boardid will affect the initialization of SOM during boot, please proceed with cautions.

```
Example:
    hrut_boardid g
    hrut_boardid s 103
Options:
    g  get board id(veeprom)
    s  set board id(veeprom)
    G  get board id(bootinfo)
    S  set board id(bootinfo)
    c  clear board id
    C   clear board id(bootinfo)
    h  display this help text
Values:
    [ 101|103|202|203|301|302....]
```

5.1.1.1 Read boardid(veeprom)

```
hrut_boardid  g
```

Examples:

```
root@x3dvbx3-hynix1G-2666:~# hrut_boardid g
11410101
```

5.1.1.2 Set boardid(veeprom)

```
hrut_boardid s board_id
```

Examples:

```
hrut_boardid s 11410101
```

The following printout should show on console:

```
set board id = 11410101
```

5.1.1.3 Read boardid(bootinfo)

```
hrut_boardid  G
```

Examples:

```
root@x3dvbx3-hynix1G-2666:~# hrut_boardid G
00
```

5.1.1.4 Set boardid(bootinfo)

```
hrut_boardid S board_id
```

Examples:

```
root@x3dvbx3-hynix1G-2666:~# hrut_boardid S 11410101
```

The following printout should show on console:

```
change board id from 0 to 11410101
```

5.1.2 hrut_bpuprofile

hrut_bpuprofile is used to set BPU status: frequency modulation, power on/off, clock on/off etc. hrut_bpuprofile is integrated and can be used under any directory you are in.

Usage:

```
BPU PROFILE HELP INFORMATION
>>> -b/--bpu     [BPU CORE,0--bpu0,1--bpu1,2--ALL BPU] (required)
>>> -p/--power    [POWER OFF/ON,0--OFF,1--ON]
>>> -c/--clock    [CLOCK OFF/ON,0--OFF,1--ON]
>>> -e/--enabletime [GET FC TIME/ON,0--OFF,1--ON]
>>> -t/--time    [GET FC TIME,NO ARGUMENT]
>>> -f/--frq     [SET BPU FREQUENCY,ARGUMENT:N]
>>> -r/--ratio    [BPU RATIO,N--N TIMES,0--FOREVER]
```

5.1.2.1 Check the utilization of the 2 BPUs

```
hrut_bpuprofile -b 2 -r 0
```

5.1.2.2 Set frequency of BPU0 to 400M

```
hrut_bpuprofile -b 0 -f 400000000
```

5.1.2.3 Power off BPU1

```
hrut_bpuprofile -b 1 -p 0
```

5.1.2.4 Check the execution time of fc

Disabled by default, need to enable this feature first)

5.1.2.4.1 Enable driver to start recording fc execution time

```
hrut_bpuprofile -b 2 -e 1
```

5.1.2.4.2 Check fc execution time, disable if no longer required

```
hrut_bpuprofile -b 2 -t
```

5.1.3 hrut_ddr

5.1.3.1 DDR QoS Configuration

DDR QoS on X3 added VPU and AR support, the port number corresponds to modules is listed below:

20191111_0	axi_0	axi_1	axi_2	axi_3	axi_4					axi_5		axi_6		axi_7
block	cpu / R5	noc	cnn0	cnn1	gdc0	pym	isp	ipu0	gdc1	JPG	VPU	sif	iar	peri

Note: AXI6 and AXI4 is configurable through register located at address 0xA4000038. Use "devmem" command to check and configure the register:

```
devmem 0xA4000038
```

The module DDR usage is direct to AXI6 when value is 1, 0 means the output is directed to AXI4. Bit17-31 corresponds to modules listed below:

```
// [17]: sif
// [18]: isp_0_m0
// [19]: isp_0_m1
// [20]: isp_0_m2
// [21]: isp_1_m0
// [22]: isp_1_m1
// [23]: isp_1_m2
// [24]: gdc_0
// [25]: t21_0
// [26]: gdc_1
// [27]: sif 2
// [28]: ipu_0
// [29]: ipu_1
// [30]: pym
// [31]: iar
```

5.1.3.2 DDR QoS Sysfs Interface

The priority value of QoS ranges from [0, 15], the larger the value, the higher the priority. The priority of each port is configured through the DDR_PORT_READ/WRITE_QOS_CTRL register of Perf Monitor. Perf Monitor will configure the DDR controller through hardware, software does not handle DDR controller.

The value of DDR QoS can be accessed and configured through sysfs interface of ddr_monitor. For example:

X3 DDR Monitor supports 9 ports in total:

Port	all	bifdma	bpu0	bpu1	cpu	peri	vio0	vio1	vpu

```
# ddr write
cat /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/all
00021100:
P0_CPU: 0
P1_BIFDMA: 0
P2_CNN0: 1
P3_CNN1: 1
P4_VIO0: 2
P5_VPU: 0
P6_VIO1: 0
P7_PERI: 0

# ddr read
cat /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/all

# Port "all" can be used to configure all ports at once, the lowest 4bits corresponds to P0_CPU,
the
# highest 4 bits corresponds to P7_PERI
echo 8f9a998d> /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/all
```

```
echo 0e56550d > /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/all

# The rest of the ports configures each port separately, the value range is [0, 15] in integer
ls /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/
all     bifdma  bpu0    bpu1    cpu     peri    vio0    vio1     vpu

# e.g:
echo 13 > /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/cpu
echo 0  > /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/bifdma
echo 5  > /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/bpu0
echo 5  > /sys/bus/platform/drivers/ddr_monitor/write_qos_ctrl/bpu1

echo 13 > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/cpu
echo 8  > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/bifdma
echo 9  > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/bpu0
echo 9  > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/bpu1
echo 10 > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/vio0
echo 9  > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/vpu
echo 15 > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/vio1
echo 8  > /sys/bus/platform/drivers/ddr_monitor/read_qos_ctrl/peri
```

5.1.3.3 Usage

DDR Monitor driver implements per monitor device driver. The perf monitor is used to check the information of each port such as DDR bandwidth, delay, etc.

5.1.3.3.1 Use hrut_ddr to read bandwidth of each port

hrut_ddr is a system tool providing visualized bandwidth data:

hrut_ddr usage information:

```
# hrut_ddr -h
DDR MONITOR HELP INFORMATION
>>> -t/--type [SAMPLE TYPE,cpu,bif,bpu0,bpu1,vio0,vpu,vio1, peri,sum,all]
>>> -p/--period [SAMPLE PERIOD, 1 - 100ms]
```

hrut_ddr can read all ports at once of each port separately, the sampling rate is 10Hz to1KHz.

For example:

The following example is presented after application is executed, when no workload is executed, the output is highly likely to be all 0s.

Accessing the DDR bandwidth usage of port "cpu", sampling rate 10Hz:

```
root@Unknown:/userdata# hrut_ddr -t cpu -p 100
[100009]cpubw R:3607MB/s W:1580MB/s
[200005]cpubw R:3752MB/s W:1611MB/s
[300010]cpubw R:3586MB/s W:1557MB/s
[400008]cpubw R:3623MB/s W:1487MB/s
[500010]cpubw R:3654MB/s W:1594MB/s
[600009]cpubw R:3654MB/s W:1587MB/s
[700010]cpubw R:3678MB/s W:1581MB/s
```

Accessing the bandwidth usage of all ports supported, the total bandwidth is printed at the end, sampling rate 10Hz:

```
root@x3dvbx3-hynix4G-2666:~# echo 1 > /proc/sys/kernel/printk
root@x3dvbx3-hynix4G-2666:~# hrut_ddr -t all -p 100
[100011]cpu R:1431MB/s W:1139MB/s bif R:0MB/s W:0MB/s bpu0 R:675MB/s W:361MB/s bpu1 R:705MB/s
W:419MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2962MB/s W:1920MB/s
[200040]cpu R:1349MB/s W:1083MB/s bif R:0MB/s W:0MB/s bpu0 R:788MB/s W:514MB/s bpu1 R:619MB/s
W:397MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2899MB/s W:1938MB/s
[300011]cpu R:1393MB/s W:1086MB/s bif R:0MB/s W:0MB/s bpu0 R:648MB/s W:352MB/s bpu1 R:757MB/s
W:438MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:118MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2948MB/s W:1878MB/s
[400009]cpu R:1352MB/s W:1083MB/s bif R:0MB/s W:0MB/s bpu0 R:792MB/s W:514MB/s bpu1 R:592MB/s
W:330MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2882MB/s W:1872MB/s
[500037]cpu R:1445MB/s W:1152MB/s bif R:0MB/s W:0MB/s bpu0 R:639MB/s W:347MB/s bpu1 R:772MB/s
W:494MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:3002MB/s W:1995MB/s
[600011]cpu R:1368MB/s W:1104MB/s bif R:0MB/s W:0MB/s bpu0 R:804MB/s W:520MB/s bpu1 R:586MB/s
W:303MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2904MB/s W:1874MB/s
[700018]cpu R:1397MB/s W:1102MB/s bif R:0MB/s W:0MB/s bpu0 R:641MB/s W:352MB/s bpu1 R:783MB/s
W:530MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:118MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2971MB/s W:1984MB/s
[800006]cpu R:1371MB/s W:1089MB/s bif R:0MB/s W:0MB/s bpu0 R:816MB/s W:525MB/s bpu1 R:602MB/s
W:302MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2935MB/s W:1868MB/s
[900013]cpu R:1433MB/s W:1147MB/s bif R:0MB/s W:0MB/s bpu0 R:635MB/s W:346MB/s bpu1 R:775MB/s
W:524MB/s vio0 R:0MB/s W:0MB/s vpu R:0MB/s W:0MB/s vio1 R:115MB/s W:0MB/s peri R:0MB/s W:0MB/s
sum R:2990MB/s W:2011MB/s
```

5.1.4 hrut_mac

[LINUX] Use hrut_mac utility to save MAC parameters of the network card of current board.

Usage:

【Set】

```
hrut_mac s aa:bb:cc:dd:ee:ff
```

【Get】

```
hrut_mac g
```

5.1.5 hrut_somstatus

X3 comes in the form of SOM and complete PCB. Now status is acquired by hrut_somstatus. For temperature sensors, different board has different detect point for temperature sensor.

Examples:

X3DVB (Partial Information is shown below):

```
root@x3dvbx3-hynix1G-2666:~# hrut_somstatus
====================1====================
temperature-->
       BOARD    : 40.8 (C)
       CPU      : 49.0 (C)
cpu frequency-->
             min       cur     max
       cpu0: 500000    800000  1200000
       cpu1: 500000    1000000 1200000
       cpu2: 500000    800000  1200000
       cpu3: 500000    800000  1200000
bpu status information---->
            min       cur            max            ratio
       bpu0: 400000000 1000000000     1200000000     0
       bpu1: 400000000 1000000000     1200000000     0
```

5.1.6 hrut_ipfull

[LINUX] Used by application code to store IP parameters. After configuration, the next boot of the board will automatically configure the IP parameters (IP&MASK) and add gateway to router.

Note: Some product's IP address is fixed and cannot configured by this utility.

Usage:

【Set】

```
hrut_ipfull s IP MASK GATEWAY
e.g. hrut_ipfull s 192.168.1.10 255.255.255.0 192.168.1.255
```

【Get】

```
hrut_ipfull g
```

The following printout should show on console:

```
ip=192.168.1.10
mask=255.255.255.0
gw=192.168.1.255
```

【Clear】

```
hrut_ipfull c
```

Once cleared, board will use default IP address 192.168.1.10 after reboot

Note:

Once configured, several files will appear under /tmp saving the parameters after reboot:

```
/tmp/ip_mac
/tmp/ip_ip
/tmp/ip_mask
/tmp/ip_gw
```

6 Camera Manual

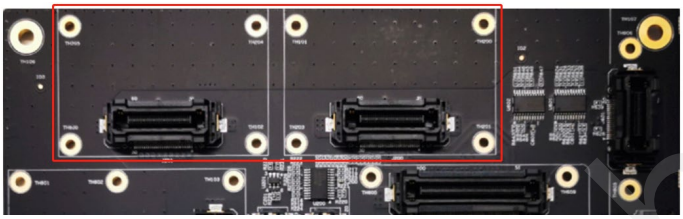
X3 Camera to display functionality verification.

6.1 Cautions of Hardware Setup

6.1.1 X3DVB

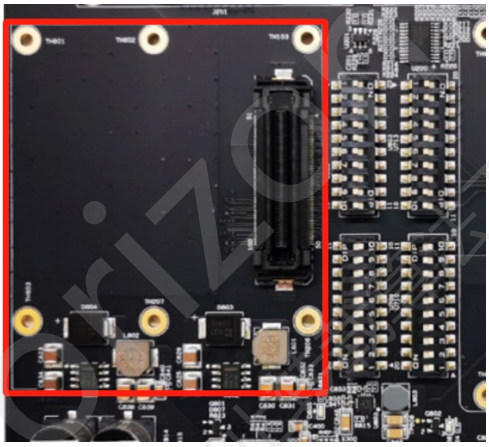
6.1.1.1 Sensor Board Sockets

X3DVB is equipped with 2 sensor board sockets as shown below:



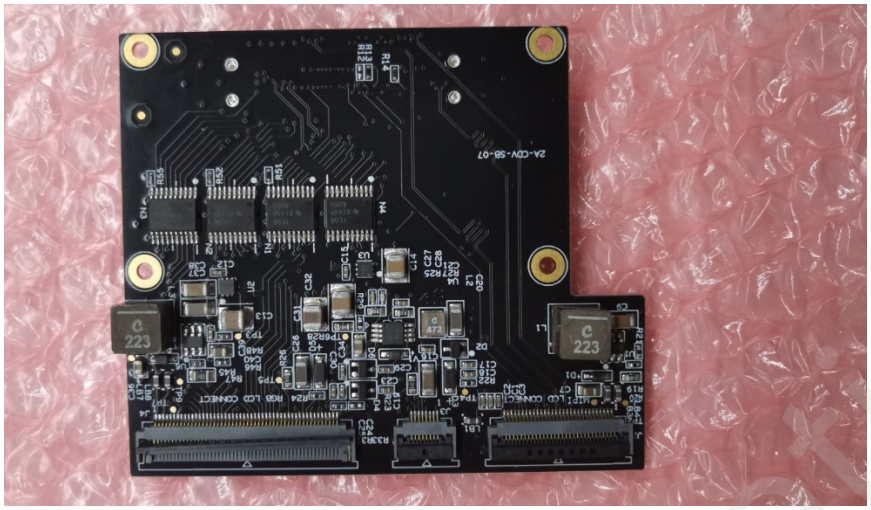
6.1.1.2 Display Board Socket

X3DVB is equipped with 1 display board socket as shown below:

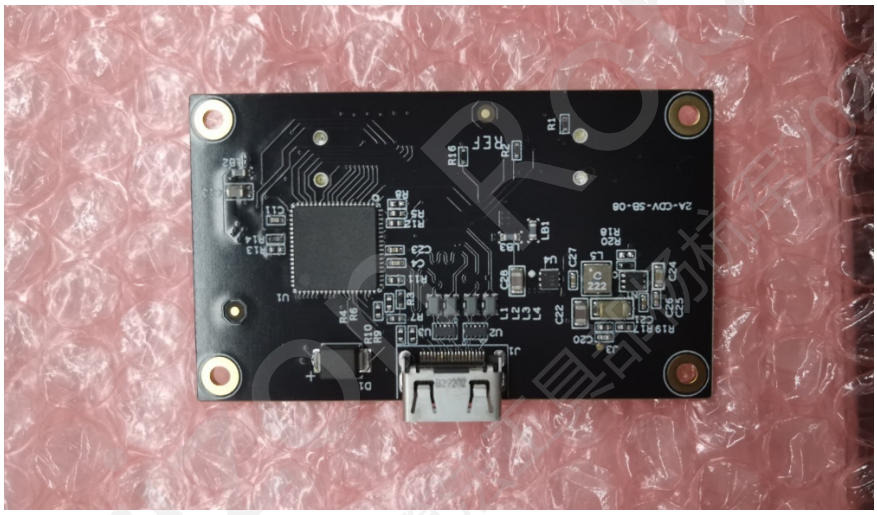


There are 2 kinds display boards:

- LCD Display board:

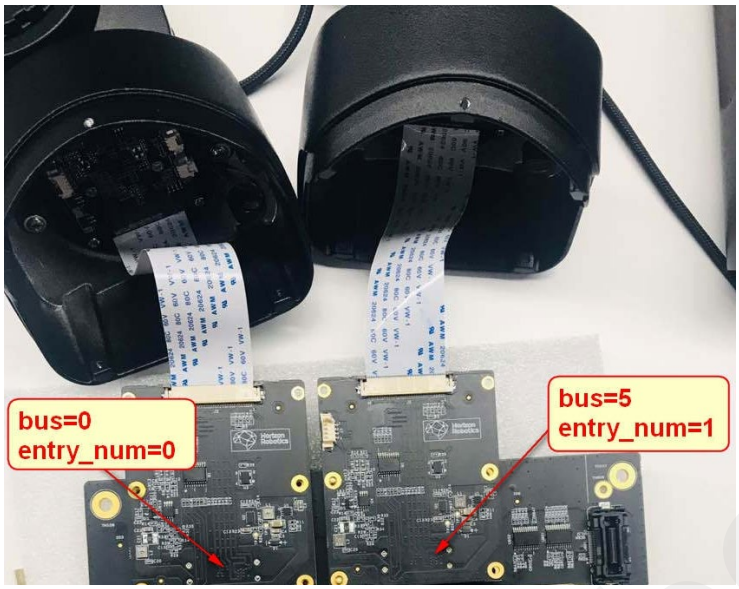


- HDMI Display board:



6.1.2 IMX327

IMX327 Module and IMX327 sensor board is properly set up as shown below. Please check if the flexible flat cable is compatible with the module. IMX327 sensor board can be inserted to either sockets:



6.2 Test Application Manual

6.2.1 Configure Displays

The default configuration of display is: 800*480 screen. To change the display configuration:

Enter UBoot, change bootargs to change display mode. Insert the following arguments to bootargs: "video=hobot:<Display Type>"

For example, configure display to HDMI output, use printenv to check if the configuration is set properly:

```
setenv bootargs ${bootargs} video=hobot:hdmi
printenv bootargs
```

Use saveenv to make this configuration permanent, or it will be effective for only one time.

```
saveenv
```

Boot to use the display configured:

```
boot
```

Display options supported:

- "video=hobot:lcd", 800*480 LCD screens. This is the default configuration. Config file used: "/etc/iar/iar_x3_lcd.json"
- "video=hobot:720p", MIPI-720*1280-DSI screen. Config file used: "/etc/iar/iar_x3_mipi720p.json"

- "video=hobot:1080p", MIPI-1080*1920-DSI screen. Config file used: "/etc/iar/iar_x3_mipi1080p.json"
- "video=hobot:hdmi", HDMI (when HDMI is preferred, bootargs is not required). Config file used: "/etc/iar/iar_x3_hdmi_1080p.json"
- "video=hobot:disable", disable display module.

To change the display configuration for each option, modify the config file mentioned above. The detailed display configuration is descripted in API manual.

6.2.2 IMX327 Verification

After successfully configured display to use HDMI, when the board boots up, the screen connected should light up. Before executing the test application, execute the following command first:

If IMX327 is inserted to socket i2c0:

```
echo 1 >/sys/class/vps/mipi_host0/param/stop_check_instart
```

If IMX327 is inserted to socket i2c5:

```
echo 1 >/sys/class/vps/mipi_host1/param/stop_check_instart
```

Based on the socket where IMX327 is inserted, modify the camera config file entries: "bus_num" and "entry_num", the config file is listed under:

```
/etc/cam/
```

If IMX327 is inserted to socket i2c0:

```
"bus_num":0
"entry_num":0
```

If IMX327 is inserted to socket i2c5:

```
"bus_num":5
"entry_num":1
```

If ds0 of vio pipeline0 is needed to be displayed, modify the "display_cam_no" option of "/etc/iar/iar_x3_hdmi_1080p.json" to 0, and "display_addr_type" to 2. (Note: the corresponding vio config file needs to be configured to enable ds0 and 1080p resolution). Save the modification and reboot. The detailed description of vio config file is listed in API manual.

Another way to configure is to use sysfs interface:

```
echo pipe param1param2param3 > /sys/devices/virtual/graphics/iar_cdev/iar_test_attr
```

Parameters in the command is:

- "param1" specifies the video layer of iar, range[0, 1]
- "param2" specifies the pipeline number of vio, range[0, 3]
- "param3" specifies the channel of vio range[0, 38] (The detailed description of each channel, please refer to API manual, "display_addr_type")

For example, to enable ds0 of video level0 from camera 1, the following command can be used:

```
echo pipe012 > /sys/devices/virtual/graphics/iar_cdev/iar_test_attr
```

After system boot successfully, use the following command to verify the camera to display functionality:

```
/app/bin/vio/vio_test -v /etc/vio/imx327_raw_12bit_1952x1097_online_Pipeline.json -r 50 -c 1 -p
1 -t 1 -l 0 -m 1 -f 1 -g 1 -D 1
```

If HDMI is used, use "-c" option to specify the exact display config file:

```
/app/bin/vio/vio_test -v /etc/vio/imx327_raw_12bit_1952x1097_online_Pipeline.json -r 50 -c 1 -p
1 -t 1 -l 0 -m 1 -f 1 -g 1 -D 1 -C /etc/iar/iar_xj3_hdmi_1080p.json
```

Among the above parameters, "-c" specifies the config number listed in the camera configurations for each board in the path /etc/cam, for example: /etc/cam/hb_xj3dev.json. For detailed introduction for the json configuration files, please refer to X3J3 Image Media Tuning Manual.

Note:

If mipi check hs error occurs during command execution, it is highly likely there is a bad connection between the sensor board and development board. Please use screws to tighten the sensor board.

7 Thermal Configuration Manual

There are 3 cooling devices in X3 in total:

```
cooling_device0: cnn0.
cooling_device1: cnn1.
cooling_device2: cpu.
```

The default strategy ("Stepwise") can be accessed by the following command:

```
cat /sys/class/thermal/thermal_zone0/policy
```

The strategies available can be accessed by the following command:

```
cat /sys/class/thermal/thermal_zone0/available_policies
```

There are 2 strategies supported: "user_space", "step_wise", please configure the strategy to suit your need. Strategy can be defined during compilation or use sysfs to configure dynamically.

For example:

Use sysfs wo configure strategy to "userspace":

```
echo user_space > /sys/class/thermal/thermal_zone0/policy
ls -l  /sys/devices/virtual/thermal/thermal_zone0
```

There are 3 trip_point in total, the default trip-point is trip_pint_1_temp(Threshold at 75℃)

```
trip_point_*_hyst （*:0 - 2）
trip_point_*_temp (*: 0 - 2)
trip_point_*_type (*: 0 - 2)
```

Use "echo" to change threshold of trp_point_1_temp:

```
echo 85000 >  /sys/devices/virtual/thermal/thermal_zone0/trip_point_1_temp
```

8 CPU Frequency Strategy

The default CPU frequency strategy is "ondemand", which would adjust the CPU frequency automatically based on work load. If necessary, CPU frequency can be fixed at maximum frequency or any other supported value.

CPU frequency fixed at maximum:

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

CPU frequency fixed at certain value:

First, set the frequency strategy to "userspace":

```
echo userspace> /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Print CPU frequency supported:

```
cat /sys/devices/system/cpu/cpufreq/policy0/scaling_available_frequencies
```

Set CPU frequency to 500M Hz:

```
echo 500000 > /sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed
```

Print current CPU Frequency:

```
cat /sys/devices/system/cpu/cpufreq/policy0/cpuinfo_cur_freq
```