Cortex-A7 i.MX6ULL Series Software Guide

Version: 1.0

2021 Sep.

Copyright © Artila Electronics Co., Ltd. All Rights Reserved.

Trademarks

The Artila logo is a registered trademark of Artila Inc. All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Artila.

Artila provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Artila reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Artila assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Model Support Listing

Category	Model	Descriptions
IoT Device Platform	M-X6ULL-B.	Linux-ready Cortex-A7 System on Module
IoT Device Platform	SBC-7530	Linux-ready, a turnkey total off-the-shelf compact ARM-based Cortex-A7 solutions, Single Board Computer
IoT Gateway	Matrix-752	Linux-ready Cortex-A7 Highly Integrated Industrial IoT Gateway

Document Amendment History

Revision	Date	Remark
V 1.0	2021 Sep.	Initial

Table of Contents

1. Overview 6
2. Access the USB Serial Console 8
2.1 USB Serial Console Introduction 8
2.2 USB Serial Console Log-in 9
3. Network Settings 10
3.1 Config the Network Interface 10
3.2 Configure the DNS Server 11
4. Access the SSH Console 12
5. Check Linux Kernel Version 13
6. File System Information 14
7. Serial Port Settings 15
7.1 Port Mapping 15
7.2 Configure the Serial Port 15
8. System Time and Real-Time Clock(RTC) 16
8.1 Adjust System Time by data Command 16
8.2 Adjust RTC by hwclock Command 16
8.3 Synchronize System Time by NTP Server 16
8.3.1 Install the ntpdate utility 16
8.3.2 Using the ntpdate utility 17
9. Insert Kernel Modules 18
10. Insert Software Package 19
11. Mount/Unmount an SD Card 20
12. Mount/Unmount a USB Card 21
13. Web Server Settings 22
13.1 Nginx Web Server 22
13.2 Root Web Page Directory 22
13.3 PHP Support 23
14. Auto-execute User Applications/Shell Scripts 24
14.1 Modify the /etc/rc5.d directory 24
14.2 Modify the /etc/profile 24
15. Change the Welcome Message 25
16. Reboot the System 26
17. User Application Development 27
17.1 Install C/C++ Cross Compilation Toolchain 27
17.2 Using the C Cross Compiler 27
17.3 Using the C++ Cross Compiler 28
17.4 Using the Native C Compiler 28

17.5 Using the Native C++ Compiler 29
17.5.1 Install the Native C++ Toolchain 29
17.5.2 Using the Native C++ Compiler 29
17.6 Using the Python Interpreter 29
17.6.1 Python 3 Support 29
18. GPIO Operation 30
19. Install an USB Wi-Fi Dongle 32
19.1 Install Hardware Driver 32
19.2 Modify the network interface configuration 32
19.3 Modify the WPA configuration 33
19.4 Restart the wireless network interface 33
20. Webmin Support 34
21. Setup Eclipse IDE 35
21.1 Configure the Eclipse IDE 35
21.2 Install the Eclipse Yocto Plug-in 37
21.3 Configuring the Cross-Compiler Options 38
21.4 Create a Hello World Project 39
22. Setup SIM card 42
23. Setup “STATUS” LED indicator 44
24. Setup Digital Input / Digital Output 45
25. Setup CAN port 46
26. Setup Audio Out 47
27. Setup Desktop Environment 48
28. Restore to Factory Default 49
29. Backup the “File System” 51
30. Restore the “File System” 53

1. Overview

This software guide applies for Artila's Cortex-A7 i.MX6ULL related models. It includes of M-X6ULL-B SoM (System on Module), Matrix-75x series Industrial IoT gateway and SBC-7530 single board computer.

- **Operation System**
  - Linux kernel 5.4.x (or Above)
  - Supports bootup from eMMC or SD card
  - Support Backup/Restore via SD card or USB device
  - Boot Loader: Barebox
  - File System: EXT4
  - GUI Engine: X11

- **Software Development**
  - Toolchain: gcc 9.3 + glibc 2.31
  - Supports in-place C/C++ code compilation

- **Package Management**
  - Package repository: Artila self-maintained repository
  - Command: Using standard apt-get command

- **Popular Packages**
  - Web server: Apache/Nginx/Lighttpd
  - Database: MySQL/SQLite3/PostgreSQL
  - Script Language: PHP/Python/Perl/NodeJS
  - Text editor: vim/nano/sed
  - Administration: Webmin

- **Protocol Stacks**
  - IPV4, ICMP, ARP, DHCP, NTP, TCP, UDP, FTP, HTTP, PPP, PPPoE, CHAP, PAP, SNMP V1/V3, SSL, SSH 1/2

- **Utilities**
  - Bash: Shell Command
  - Telnet: Telnet client program
  - Busybox: Linux utility collection
  - FTP: FTP client program

- **Daemon**
  - pppd: Dial In/out over serial port and PPPoE
  - snmpd: SNMP agent program
  - ftpd: FTP server program
  - nginx: Web server program
  - sshd: secured shell server
  - iptables: Firewall service manager

- **Standard Device Drivers**
  - ttymxc0: serial console port (CORTEX-A7 SERIES debug port)
  - ttymxc1~ttymxc4: serial ports (CORTEX-A7 SERIES UART0~UART3)
  - gpio: General Purpose I/O
  - mmc: SD/MMC:
  - rtc: Real Time Clock
  - sda: USB flash memory disk
  - ttyACM: USB Modem
  - ttyUSB: USB RS-232 adaptor
  - spi: spi bus

- **I/O devices Control**

  Use standard I/O device control to access following devices:
  - Ethernet: eth1
  - Serial Ports: ttymxc1, ttymxc2, ttymxc3, ttymxc4
  - Serial Console Port: ttymxc0
  - Real time clock: rtc0
  - USB Flash Disk: sda, sda1, sdb, sdb1
  - SD memory Card: mmc0
  - USB Serial Cable: ttyUSB0, ttyUSB1
  - SPI bus: spi0

- **Default Setting**
  - IP Default setting:
    - eth1: 192.168.2.127 (Netmask: 255.255.255.0)
  - ssh Login: root
  - Password: root
  - Terminal type: VT100

2. Access the USB Serial Console

2.1 USB Serial Console Introduction

All the Cortex-A7 Series based Matrix IoT gateways come with a USB client port (micro-USB connector), which is used as the serial console. Please prepare a USB-to-microUSB cable to connect the Matrix IoT gateway to a PC.
When the Matrix IoT gateway finished its boot up process, it will automatically emulate an USB CDC compatible serial device.

Cortex-A7 Series based Matrix IoT gateway comes with a USB client port	Use a standard on-the-shelf USB-to-MicroUSB cable to connect to the Matrix-7XX	Linux/Windows/OSX Desktop/Notebook PC

The identifier name of the CDC serial port varies depending on your computer's operation system and the numbers of the serial ports which are already installed on your computer.

On Linux system, the serial port name appears like ttymxc0, etc.
On OSX system, the serial port name appears like tty.usbmodem1421, tty.usbmodem1422, etc.
On Windows system, the serial port name appears like COM3, COM4, etc.

The serial communication parameters are: **115200**, **N81**, **VT100**. Use your preferred serial terminal tools to access the Matrix IoT gateway's serial console.
For example:
On Windows system, use **putty** or **teraterm**.
On Linux/OSX system, use **minicom** utility.

> ☞ Note
> For Linux, Mac OSX and Windows 10 computers, the CDC serial driver is already built-in and will be activated automatically.

2.2 USB Serial Console Log-in

User name: **root**

Password: **root**

Following example by Matrix-752

```
 Welcome to

      **                      **  **
      **                **          **
    **  **              **          **
    **  **       ****   ****   **   **     ******
  **       **     **     **     **   **            **
  **       **     **     **     **   **     ********
  **********      **     **     **   **   **       **
**           **   **     **     **   **   **       **
**           **   **       **   **   **     ********

For further information check:
http://www.artila.com/

Poky (Yocto Project Reference Distro) 3.3 matrix_120601 ttymxc0

Matrix_120601 login: root
Password:
Last login: Fri Jan 01 21:20:24 UTC 2021 on ttymxc0

     **                    **  **
     **              **          **
    **  **           **          **
    **  **      ****  ****   **   **     ******
  **      **     **    **     **   **           **
  **      **     **    **     **   **     ********
  **********      **    **     **   **   **       **
**          **   **    **     **   **   **       **
**          **   **      **   **   **     ********

http://www.artila.com/

[root@matrix_120601 ~]#
```

3. Network Settings

3.1 Config the Network Interface

The Cortex-A7 Series based Matrix IoT gateways come two Ethernet ports, the default network settings are shown below:

Ethernet Type	Port Label	Device mapping	IP mode	IP address
10/100Mbit	LAN	eth0	DHCP	auto
10/100Mbit	LAN	eth1	static	192.168.2.127

Users may need to modify the network settings to meet their LAN environment. The network interface configuration file path is **/etc/network/interfaces**. Edit and save the configuration file, then use **ifdown** and **ifup** command to ON/OFF the specific network interface to activate the network settings.

```
[root@matrix_120601 ~]#cat /etc/network/interfaces
# /etc/network/interfaces -- configuration file for ifup(8), ifdown
(8)

# The loopback interface
auto lo
iface lo inet loopback

# Wired or wireless interfaces
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
       address 192.168.2.127
       netmask 255.255.255.0
       network 192.168.2.0
       #gateway 192.168.2.1

[root@matrix_120601 ~]#
```

The following screen capture shows the eth0 of the Matrix-752 got a valid IP: 192.168.1.93.

```
[root@matrix_120601 ~]#ifdown eth0
[root@matrix_120601 ~]#ifup eth0
udhcpc: option -h NAME is deprecated, use -x hostname:NAME
udhcpc (v1.24.1) started
Sending discover...
Sending select for 192.168.1.93...
Lease of 192.168.1.93 obtained, lease time 86400
/etc/udhcpc.d/50default: Adding DNS 208.67.220.220
/etc/udhcpc.d/50default: Adding DNS 208.67.222.222

[root@matrix_120601 ~]#
```

3.2 Configure the DNS Server

The DNS configuration file path is **/etc/resolv.conf**. Users may edit the file according to their specific network environment.

```
[root@matrix_120601 ~]#cat /etc/resolv.conf

[root@matrix_120601 ~]#ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:13:48:12:06:02
          inet addr:192.168.2.127  Bcast:192.168.2.255  Mask:255.255.25
5.0
          inet6 addr: fe80::213:48ff:fe03:84a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:126 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14966 (14.6 KiB)  TX bytes:3770 (3.6 KiB)
          Interrupt:41 Base address:0xc000

[root@matrix_120601 ~]#ping google.com
ping: unknown host google.com

[root@matrix_120601 ~]#echo 'nameserver 8.8.8.8' > /etc/resolv.conf

[root@matrix_120601 ~]#cat /etc/resolv.conf
nameserver 8.8.8.8

[root@matrix_120601 ~]#ping google.com
PING google.com (216.58.200.238) 56(84) bytes of data.
64 bytes from tsa03s01-in-f14.1e100.net (216.58.200.238): icmp_seq=1 t
tl=52 time
=13.9 ms
64 bytes from tsa03s01-in-f238.1e100.net (216.58.200.238): icmp_seq=2
ttl=52 tim
e=15.3 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 13.910/14.616/15.322/0.706 ms
```

Warning: Please be noted that, the /etc/resolv.conf is physically located in the RAM disk, so the content of the file will disappear after system reboot.

4. Access the SSH Console

Most Linux/OSX computers come with built-in SSH client utility. For Windows users, it is highly recommended to use **putty** as an SSH client.

User name: **root**

Password: **root**

```
$ ssh root@192.168.1.64
The authenticity of host '192.168.1.64 (192.168.1.64)' can't be establ
ished. ECDSA key fingerprint is SHA256:gQQ9QzBGV0F0fZCmP5qLxioRkbPlRqJ
DLnLuklLZVhQ. Are you sure you want to continue connecting (yes/no)? y
es Warning: Permanently added '192.168.1.64' (ECDSA) to the list of kn
own hosts. root@192.168.1.64's
password:
Last login: Fri May 6 20:47:14 2016 from 192.168.1.54 Welcome to

     **                  **  **
     **             **       **
   **  **           **       **
   **  **      ****  ****  **  **    ******
  **      **    **    **    **  **         **
  **      **    **    **    **  **    ********
  **********    **    **    **  **  **      **
**          **  **    **    **  **  **      **
**          **  **      **  **  **    ********

For further information check:
http://www.artila.com/

[root@matrix_120601 ~]#
```

5. Check Linux Kernel Version

```
[root@matrix_120601 ~]#uname -a
Linux Matrix_120601 5.10.59- rt51 #1 PREEMPT_RT Sun Aug 15 12:00:27 UT
C 2021 armv7l armv7l armv7l GNU/Linux

[r root@matrix_120601 ~]#uname -v
#1 PREEMPT_RT Wed Nov 18 18:20:34 UTC 2021

[root@matrix_120601 ~]#uname -r
5.10.59- rt51
[root@matrix_120601 ~]#
```

6. File System Information

The Cortex-A7 Series based Matrix IoT gateways come with 16GB on-board eMMC Flash memory, which contains boot loader, Linux kernel, root file system and user disk (/home).

```
[root@matrix_120601 ~]#lsblk
NAME       MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0    179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mtdblock0   31:0    0  8.3M  0 disk
mtdblock1   31:1    0  8.2M  0 disk
mtdblock2   31:2    0  7.7M  0 disk
mtdblock3   31:3    0  7.7M  0 disk
mtdblock4   31:4    0  7.6M  0 disk
mtdblock5   31:5    0  3.9M  0 disk

[root@matrix_120601 ~]#
```

```
[root@matrix_120601 /]#ls -F
bin/  boot/  dev/  etc/  gpio/  home/  lib/  lost+found/  media/  mnt/
proc/  run/  sbin/  swapfile  sys/  tmp@  usr/  var/
[root@matrix_120601 /]#
```

```
[root@matrix_120601 ~]#df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        13G  906M   11G   8% /
devtmpfs        249M     0  249M   0% /dev
tmpfs           249M  152K  249M   1% /run
tmpfs           249M  296K  249M   1% /var/volatile

[root@matrix_120601 ~]#
```

7. Serial Port Settings

7.1 Port Mapping

The Cortex-A7 Series based Matrix IoT gateways come with four or eight serial communication ports. The first four serial ports are CPU native serial ports. Some Cortex-A7 Series based Matrix IoT gateways provide more serial ports via USB-to-Serial chip. The serial port mapping information is listed below:

Port 1 → /dev/ttymxc1

Port 2 → /dev/ttymxc2

Port 3 → /dev/ttymxc3

Port 4 → /dev/ttymxc4

7.2 Configure the Serial Port

Please use the built-in ***setuart*** utility to display/modify the operation mode (RS-232/485) and communication parameters of the first four serial ports (ttymxc1/2/3/4).

```
[root@matrix_120601 ~]#setuart -h
Artila utility: setuart
Usage: setuart [OPTION]

 -h      display this help and exit
 -v      print version number and exit
 -p      uart port number
 -t      uart interface type [232,485]
 -b      set baudrate, up to 921600bps

Examples:
  setuart -p 1                    display port 1 type and baudrate
  setuart -p 1 -t 485 -b 115200  set port 1 type RS-485 and baud to 115200
  setuart -p 1 -t 232 -b 9600   set port 1 type to RS-232 and baud to 9600
```

☞ Caution

The serial port's mode and associated communication parameters will go back to factory default after system reboot.

8. System Time and Real-Time Clock(RTC)

8.1 Adjust System Time by data Command

The Cortex-A7 Series based Matrix IoT gateways support standard ***date*** command to adjust the Linux system time manually. A typical usage is: ***date MMDDhhmmYYYY***.

```
[root@matrix_120601 ~]#date 050717132016
Sat May 7 17:13:00 UTC 2016
```

8.2 Adjust RTC by hwclock Command

To adjust the on-board Real-time clock (RTC), please follow the steps shown below: First, to adjust the system time by using the ***date*** command. Then use the ***hwclock*** command to synchronize the system time to the RTC.

A typical usage is: ***hwclock -w***.

```
[root@matrix_120601 ~]#hwclock
Thu May 26 15:31:49 2016  0.000000 seconds

[root@matrix_120601 ~]#date
Thu May 26 15:32:00 UTC 2016

[root@matrix_120601 ~]#hwclock -w

[root@matrix_120601 ~]#
```

8.3 Synchronize System Time by NTP Server

8.3.1 Install the ntpdate utility

The Cortex-A7 Series based Matrix IoT gateways support the ***ntpdate*** NTP client utility to synchronize the system date with specified NTP server. Users need to install the ***ntpdate*** utility first by executing the ***apt-get install ntpdate*** command.

```
 [root@matrix_120601 ~]#apt-get install ntpdate
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ntpdate
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/41.0 kB of archives.
After this operation, 0 B of additional disk space will be used.
Selecting previously unselected package ntpdate.
(Reading database ... 17344 files and directories currently installe
d.)
Preparing to unpack .../ntpdate_4.2.8p9-r0_armhf.deb ...
Unpacking ntpdate (4.2.8p9-r0) ...
Setting up ntpdate (4.2.8p9-r0) ...
```

8.3.2 Using the ntpdate utility

The following example shows how to use the ***ntpdate*** utility to synchronize the system with the NTP server **0.pool.ntp.org**.

```
[root@matrix_120601 ~]#date
Mon Apr 10 07:17:31 UTC 2017

[root@matrix_120601 ~]#date 050717132016
Sat May  7 17:13:00 UTC 2016

[root@matrix_120601 ~]#ntpdate 0.pool.ntp.org
10 Apr 07:18:36 ntpdate[1025]: step time server 61.216.153.106 offset
29167497.848661 sec

[root@matrix_120601 ~]#date
Mon Apr 10 07:18:45 UTC 2017

[root@matrix_120601 ~]#date;hwclock
Mon Apr 10 07:18:59 UTC 2017
Mon Apr 10 07:18:58 2017  0.000000 seconds

[root@matrix_120601 ~]#hwclock -w

[root@matrix_120601 ~]#date;hwclock
Mon Apr 10 07:19:15 UTC 2017
Mon Apr 10 07:19:15 2017  0.000000 seconds
```

9. Insert Kernel Modules

Users can use command ***lsmod*** to list all installed kernel modules.

```
[root@matrix_120601 rc5.d]#lsmod
Module                  Size  Used by
usb_f_mass_storage     25809  2
usb_f_acm               4064  2
u_serial                7750  3 usb_f_acm
libcomposite           33643  12 usb_f_acm,usb_f_mass_storage
nfsd                  251055  11
auth_rpcgss            39359  1 nfsd
oid_registry            2441  1 auth_rpcgss
exportfs                3541  1 nfsd
nfs_acl                 2510  1 nfsd
lockd                  53405  1 nfsd
grace                   1627  2 nfsd,lockd
sunrpc                175725  16 auth_rpcgss,nfsd,nfs_acl,lockd
atmel_usba_udc         15098  0
udc_core               10846  5 usb_f_acm,usb_f_mass_storage,atmel_usba_
udc,u_serial,libcomposite
```

To load additional kernel modules during the system boot-up, you can modify the file: ***/etc/modules***.

```
[root@matrix_120601 ~]#cat /etc/modules
atmel_usba_udc
#g_serial
#mt7601Usta
```

10. Insert Software Package

The Cortex-A7 Series based Matrix IoT gateways support standard **apt** (Advanced Package Tool) package management utility. With this utility, users can easily install, upgrade, remove software packages. Artila provides a self-maintained software repository. The apt configuration file path is ***/etc/apt/sources.list***.

```
[root@matrix_120601 ~]#ls /etc/apt
apt.conf  apt.conf.d  preferences.d  sources.list  sources.list.d

[root@matrix_120601 ~]#cat /etc/apt/sources.list
deb [trusted=yes] http://www.artila.com/download/imx6ul/linux/deb/cor
texa7hf-neon cortexa7hf-neon main
deb [trusted=yes] http://www.artila.com/download/imx6ul/linux/deb/cor
texa7t2hf-neon cortexa7t2hf-neon main
deb [trusted=yes] http://www.artila.com/download/imx6ul/linux/deb/cor
texa7t2hf-neon-mx6ul cortexa7t2hf-neon-mx6ul main
deb [trusted=yes] http://www.artila.com/download/imx6ul/linux/deb/all
 all main
deb [trusted=yes] http://www.artila.com/download/imx6ul/linux/deb/m_x
6ull m_x6ull main
```

** Please be noted the last line of the ***/etc/apt/sources.list*** varies according to specific model name.

Commonly used apt commands are listed below:

- ***apt-get install*** <package> to install package
- ***apt-get remove*** <package> to remove package
- ***apt-cache search*** <package> to search package
- ***apt-get update*** to update the package list
- ***apt-get upgrade*** to upgrade installed packages

11. Mount/Unmount an SD Card

The Cortex-A7 Series based Matrix IoT gateways support SD card access. If an SD card is inserted, you can use ***lsblk*** command to find the device identifier name. And then use ***mount*** command to mount the SD card to a folder.

Before SD Insertion

```
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

After SD Insertion

```
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mmcblk1     179:24   0  1.9G  0 disk
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

Mount mmcblk1 to /media.

```
[root@matrix_120601 ~]#mount /dev/mmcblk1 /media
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mmcblk1     179:24   0  1.9G  0 disk /media
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

Unmount /media.

```
[root@matrix_120601 ~]#umount /media
```

12. Mount/Unmount a USB Card

The Cortex-A7 Series based Matrix IoT gateways support generic USB drives. If an USB drive is inserted, you can use ***lsblk*** command to find the device identifier name. And then use ***mount*** command to mount the USB drive to a folder.

Before USB drive Insertion

```
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

After USB drive Insertion

```
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    1 14.5G  0 disk
`-sda1        8:1    1 14.5G  0 part
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

Mount sda1 to /media.

```
[root@matrix_120601 ~]#mount /dev/sda1 /media
[root@matrix_120601 ~]#lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    1 14.5G  0 disk
`-sda1        8:1    1 14.5G  0 part /media
mmcblk0     179:0    0  7.3G  0 disk
`-mmcblk0p1 179:1    0  7.3G  0 part /
mtdblock0    31:0    0  8.3M  0 disk
mtdblock1    31:1    0  8.2M  0 disk
mtdblock2    31:2    0  7.7M  0 disk
mtdblock3    31:3    0  7.7M  0 disk
mtdblock4    31:4    0  7.6M  0 disk
mtdblock5    31:5    0  3.9M  0 disk
```

Unmount /media.

```
[root@matrix_120601 ~]#umount /media
```

13. Web Server Settings

13.1 Nginx Web Server

The Cortex-A7 Series based Matrix IoT gateways come with pre-installed ***nginx*** web server. The configuration file is **/etc/nginx/nginx.conf**.

13.2 Root Web Page Directory

The default root web page directory is ***/var/www/localhost/html***. This path can be changed by modifying the above configuration file.

```
[root@matrix_120601 ~]#ls /var/www/localhost/html
50x.html  index.html

[root@matrix_120601 ~]#
```

13.3 PHP Support

The Cortex-A7 Series based Matrix IoT gateways support commonly used server-side script languages, including Perl, PHP and Python. Per and Python support are built-in, while PHP support needs to be installed manually using ***apt-get*** command.

- ***apt-get install -y php php-fpm php-fpm-apache2 php-cli***

```
[root@matrix_120601 ~]#php-cgi -v
PHP 7.4.16 (cgi-fcgi) (built: Mar  2 2021 10:35:17)
Copyright (c) The PHP Group

[root@matrix_120601 ~]#php-fpm -v
PHP 7.4.16 (fpm-fcgi) (built: Mar  2 2021 10:35:17)
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

[root@matrix_120601 ~]#php -v
PHP 7.4.16 (cli) (built: Mar  2 2021 10:35:17) ( NTS )
Copyright (c) The PHP Group
```

14. Auto-execute User Applications/Shell Scripts

14.1 Modify the /etc/rc5.d directory

To automatically start user applications after system boot-up, please edit a shell script to execute the program, and put that script file to the folder: ***/etc/rc5.d***.

```
[root@matrix_120601 rc5.d]#ls
S01networking@   S19nfscommon@   S60php-fpm@      S99stop-bootlogd@
S02dbus-1@       S20atd@         S90crond@        S99usbgadget@
S09sshd@         S20hwclock.sh@  S92nginx@        S99webmin@
S12rpcbind@      S20nfsserver@   S99readyled@
S15mountnfs.sh@  S20syslog@      S99rmnologin.sh@
```

14.2 Modify the /etc/profile

To automatically start user shell scripts after system boot-up, please modify the ***/etc/profile*** accordingly.

15. Change the Welcome Message

The welcome message file is ***/etc/motd***, the default content is shown below, modify the content at your will.

```
[root@matrix_120601 ~]#cat /etc/motd
Welcome to

      **                    **  **
      **              **        **
    **  **            **        **
    **  **       ****  ****  **  **    ******
  **      **    **    **    **  **          **
  **      **    **    **    **  **    ********
  **********    **    **    **  **  **      **
**          **  **    **    **  **  **      **
**          **  **      **  **  **    ********

For further information check:
http://www.artila.com/
```

16. Reboot the System

To re-boot the system, use the ***reboot*** command.

```
[root@matrix_120601 ~]#reboot

Broadcast message from root@Matrix752 (ttyGS0) (Sun May 8 15:51:47 201
6):
The system is going down for reboot NOW!
```

17. User Application Development

17.1 Install C/C++ Cross Compilation Toolchain

The following instructions are based on ***64-bit*** Ubuntu Linux environment:

Step 1, Download the toolchain installation script from Artila's website, the URL is:

$wget

http://www.artila.com/download/imx6ul/linux/sdk/poky-glibc-x86_64-meta-toolchain-cortexa7t2hf-neon-m-x6ull-toolchain-3.3.1.sh

Step 2, Execute the toolchain installation script.

$ sh

poky-glibc-x86_64-meta-toolchain-cortexa7t2hf-neon-m-x6ull-toolchain-3.3.1.sh

Step 3, Activate the environment settings.

$ source

poky-glibc-x86_64-meta-toolchain-cortexa7t2hf-neon-m-x6ull-toolchain-3.3.1.sh

17.2 Using the C Cross Compiler

Step 1: Execute ***$CC*** command to compile the C source file.

Step 2: Execute ***scp*** command to upload the compiled binary file to the Matrix IoT gateway.

```
$ cat hello.c
#include <stdio.h>

int main(){
printf("Hello World!\n");
return 0;
}

$ $CC -o hello_c hello.c

$ scp hello_c root@192.168.1.70:/home/root
root@192.168.1.70's password:
hello_c                                   100% 9800     9.6KB/s   00:00


$
```

17.3 Using the C++ Cross Compiler

Step 1: Execute ***$CXX*** command to compile the C++ source file.

Step 2: Execute ***scp*** command to upload the compiled binary to the Matrix IoT gateway.

```
$ cat hello.cpp
#include <iostream>
using namespace std;
int main() {
cout << "Hello! World!\n";
return 0;
}

$ $CXX -o hello_cpp hello.cpp

$ scp hello_cpp root@192.168.1.70:/home/root
root@192.168.1.70's password:
hello_cpp                                     100%   11KB  10.9KB/s   00:00

$
```

17.4 Using the Native C Compiler

User application can also be directly developed on the **CORTEX-A7 SERIES** based Matrix IoT gateways. By default, ***gcc*** toolchain is pre-installed on the **CORTEX-A7 SERIES** based Matrix IoT gateways.

```
[root@matrix_120601 ~]#cat hello.c
#include <stdio.h>
int main()
{
 printf("Hello World!\n");
}

[root@matrix_120601 ~]#gcc -o hello hello.c

[root@matrix_120601 ~]#./hello
Hello World!
```

17.5 Using the Native C++ Compiler

17.5.1 Install the Native C++ Toolchain

Users can install the native C++ toolchain via ***apt-get*** command. Two packages are needed to build a C++ source file, the ***g++*** package and the ***g++-symlinks*** package.

- ***apt-get install g++ g++-symlinks***

17.5.2 Using the Native C++ Compiler

```
[root@matrix_120601 ~]#cat hello.cpp
#include <iostream>
using namespace std;
int main() {
cout << "Hello World!\n";
return 0;
}

[root@matrix_120601 ~]#g++ -o hello_cpp hello.cpp

[root@matrix_120601 ~]#./hello_cpp
Hello World!

[root@matrix_120601 ~]#
```

17.6 Using the Python Interpreter

The **CORTEX-A7 SERIES** based Matrix IoT gateways come with pre-built Python 3 interpreter.

```
[root@matrix_120601 ~]#python3
Python 3.8.2 (default, Feb 25 2020, 10:39:28)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The Python ***pip3*** package manager is also included by default. To upgrade the pip itself, execute the following command:

- ***pip3 install –upgrade pip***

18. GPIO Operation

The CORTEX-A7 series comes a bunch of GPIO (General Purpose IO) pins.

By implementation, there are 15 GPIO pins are controlled in user space.

GPIO definitions are shown as below:

No.	Software Definition	Hardware Functions	Pin location on M-X6ULL-B
GPIO1	gpio40	GPIO2_8	(CN1 pin 10)
GPIO2	gpio41	GPIO2_9	(CN1 pin 12)
GPIO3	gpio42	GPIO2_10	(CN1 pin 14)
GPIO4	gpio43	GPIO2_11	(CN1 pin 16)
GPIO5	gpio44	GPIO2_12	(CN1 pin 18)
GPIO6	gpio45	GPIO2_13	(CN1 pin 20)
GPIO7	gpio111	GPIO4_15	(CN1 pin 15)
GPIO8	gpio107	GPIO4_11	(CN1 pin 17)
GPIO9	gpio46	GPIO2_14	(CN1 pin 19)
GPIO10	gpio47	GPIO2_15	(CN1 pin 21)
GPIO11	gpio9	GPIO1_9	(CN1 pin 23)
GPIO12	gpio30	GPIO1_30	(CN3 pin 31)
GPIO13	gpio31	GPIO1_31	(CN3 pin 33)
GPIO14	gpio109	GPIO4_13	(CN3 pin 35)
GPIO15	gpio110	GPIO4_14	(CN3 pin 37)

It means,

- the GPIO**2_11**, which is located on the SoM M-X6ULL's pin16 of the CN1 connector, is mapped to number gpio**43**;
- the GPIO**1_31**, which is located on SoM M-X6ULL's pin31 of the CN3 connector, is mapped to number **gpio30**

All information is listed as following:

https://www.kernel.org/doc/Documentation/gpio/sysfs.txt

```
[root@matrix_120601 gpio]#pwd
/gpio
[root@matrix_120601 gpio]#ls
GPIO1  GPIO10  GPIO11  GPIO12  GPIO13  GPIO14  GPIO15  GPIO2
GPIO3  GPIO4  GPIO5  GPIO6  GPIO7  GPIO8  GPIO9
[root@matrix_120601 gpio]#cd GPIO1
[root@matrix_120601 gpio1]#ls
Active_low  device  direction  edge  power  subsystem  uevent  value
```

For Example,

to set gpio31 as output:

```
[root@matrix_120601 ~]#cd /sys/class/gpio/

[root@matrix_120601 gpio]#ls
export   gpiochip0  gpiochip128  gpiochip32  gpiochip64  gpiochip96  un
export

[root@matrix_120601 gpio]#echo 31 > export

[root@matrix_120601 gpio]#ls
export      gpiochip128 gpiochip64  gpio31
gpiochip0  gpiochip32   gpiochip96  unexport

[root@matrix_120601 gpio]#cd gpio31

[root@matrix_120601 gpio31]#ls
active_low  device  direction  edge  power  subsystem  uevent  value

[root@matrix_120601 gpio31]#echo 'out' > direction

[root@matrix_120601 gpio31]#echo 1 > value

[root@matrix_120601 gpio31]#echo 0 > value

[root@matrix_120601 gpio31]#cd ..

[root@matrix_120601 gpio]#ls
export      gpiochip128 gpiochip64  gpio31
gpiochip0  gpiochip32   gpiochip96  unexport

[root@matrix_120601 gpio]#echo 31 > unexport

[root@matrix_120601 gpio]#ls
export   gpiochip0  gpiochip128  gpiochip32  gpiochip64  gpiochip96  un
export
```

19. Install an USB Wi-Fi Dongle

The CORTEX-A7 SERIES based Matrix IoT gateways support USB Wi-Fi dongles.

19.1 Install Hardware Driver

The USB Wi-Fi driver can be installed via apt-get utility.

The following example is using RT8192 wifi dongle.

```
[root@matrix_120601 ~]#apt-get install kernel-module-rtl8xxxu linux-firmware-rtl8192cu
```

19.2 Modify the network interface configuration

The network interface configuration file path is ***/etc/network/interfaces***. A typical configuration example is listed below:

```
# Wireless interfaces
auto wlan0
iface wlan0 inet dhcp
    wireless_mode managed
    wireless_essid any
    wpa-driver nl80211, wext
    wpa-conf /etc/wpa_supplicant.conf
```

Be noted the last line of the above example, which specifies an additional configuration file for WPA settings. In this example, the WPA configuration file path is ***/etc/wpa_supplicant.conf***.

19.3 Modify the WPA configuration

Modify the ***/etc/wpa_supplicant.conf*** according to the Wi-Fi environment of your factory/office. A typical configuration example is listed below:

```
# WPA configuration

ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
ap_scan=1

# WEP example
network={
  ssid="Artila"
  key_mgmt=NONE
  wep_key0=ABCABCABC
}

# WPA/WPA2 example
Network={
  ssid="Artila"
  key_mgmt=WPA-PSK
  auth_alg=OPEN
  psk="ABCABCABC"

}
```

19.4 Restart the wireless network interface

```
[root@matrix_120601 ~]#ifdown wlan0

[root@matrix_120601 ~]#ifup wlan0
```

20. Webmin Support

The CORTEX-A7 SERIES based Matrix IoT gateways support the Webmin, which is a browser_based system management tool.
To access the Webmin, please visit https://192.168.2.127

Username: admin
Password: admin

21. Setup Eclipse IDE

Users can integrate the CORTEX-A7 SERIES tool chain into the Eclipse IDE. It can be downloaded the Eclipse IDE for **C/C++** Developers (**Luna**) from https://www.eclipse.org/downloads/packages/release/Luna/SR2

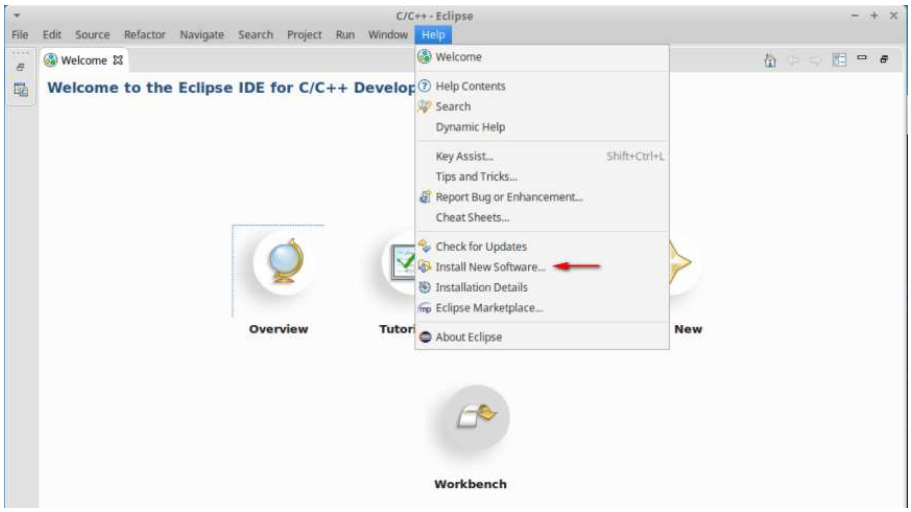
21.1 Configure the Eclipse IDE

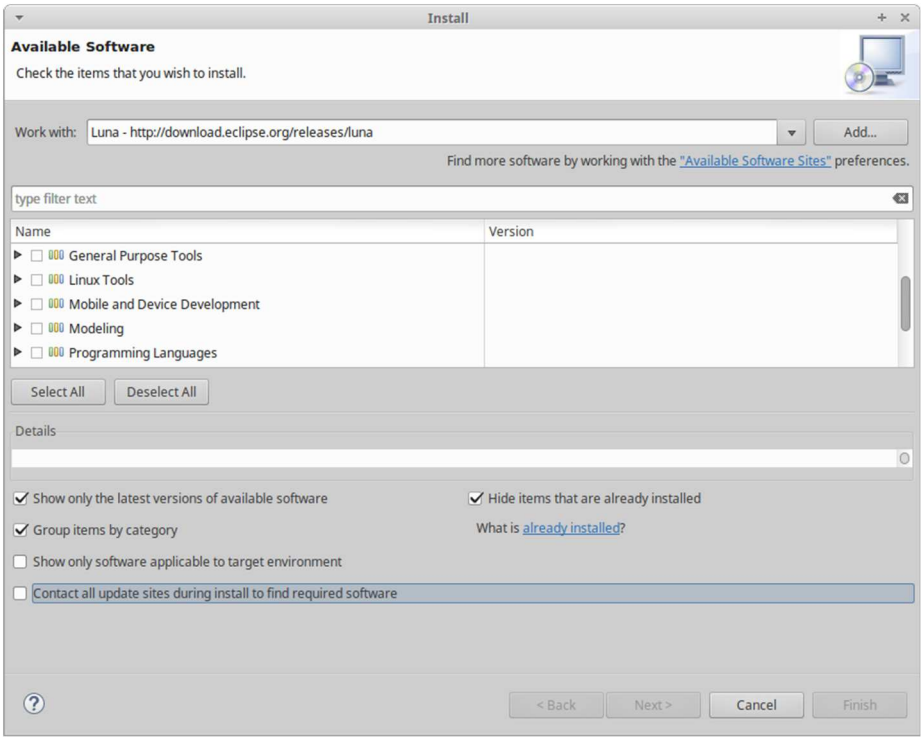
Step 1, Start the Eclipse IDE.

Step 2, From “Help” menu select “Install New Software”

- Add “Luna - http://download.eclipse.org/releases/luna”.
- Select the following items (If these selections do not appear in the list, that means the items are already installed.)
  - Linux Tools
    - Linux Tools LTTng Tracer Control
    - Linux Tools LTTng Userspace Analysis
    - LTTng Kernel Analysis
  - Mobile and Device Development
    - C/C++ Remote Launch (Requires RSE Remote System Explorer)
    - Remote System Explorer End-user Runtime
    - Remote System Explorer User Actions
    - Target Management Terminal (Core SDK)
    - TCF Remote System Explorer add-in
    - TCF Target Explorer
  - Programming Languages
    - C/C++ Autotools Support
    - C/C++ Development Tools

Step3, Complete the installation and restart the Eclipse IDE



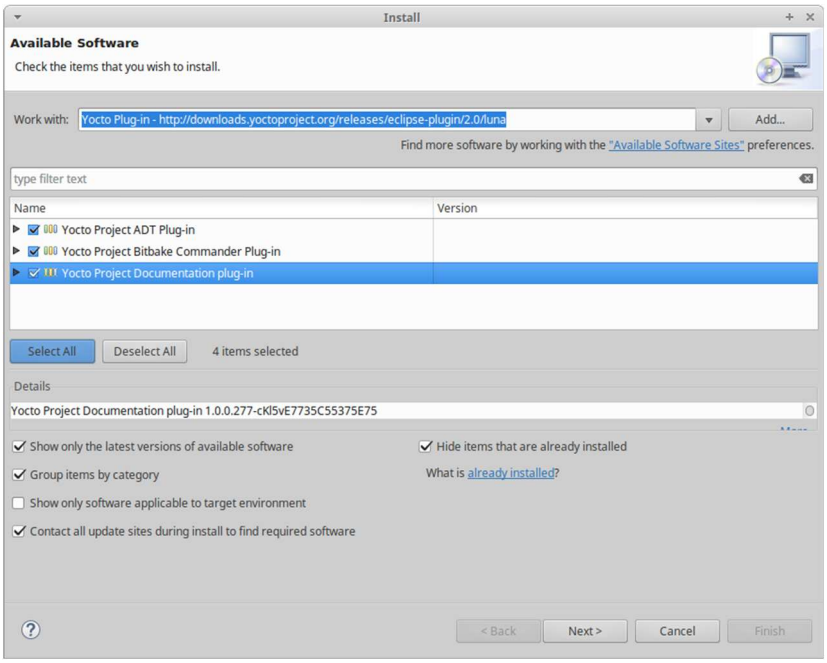


21.2 Install the Eclipse Yocto Plug-in

Step1, From the “Help” menu select “Install New Software”

- Add URL “http://downloads.yoctoproject.org/releases/eclipse-plugin/2.0/luna” and provide a meaningful name.
- Select the following items
  - Yocto Project ADT Plug-in,
  - Yocto Project Bitbake Commander Plug-in
  - Yocto Project Documentation plug-in.

Step2, Complete the installation and restart the Eclipse IDE



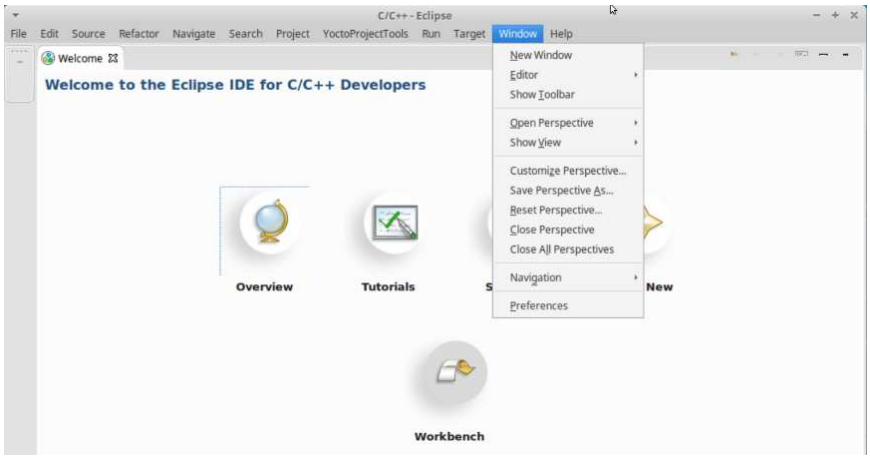
21.3 Configuring the Cross-Compiler Options

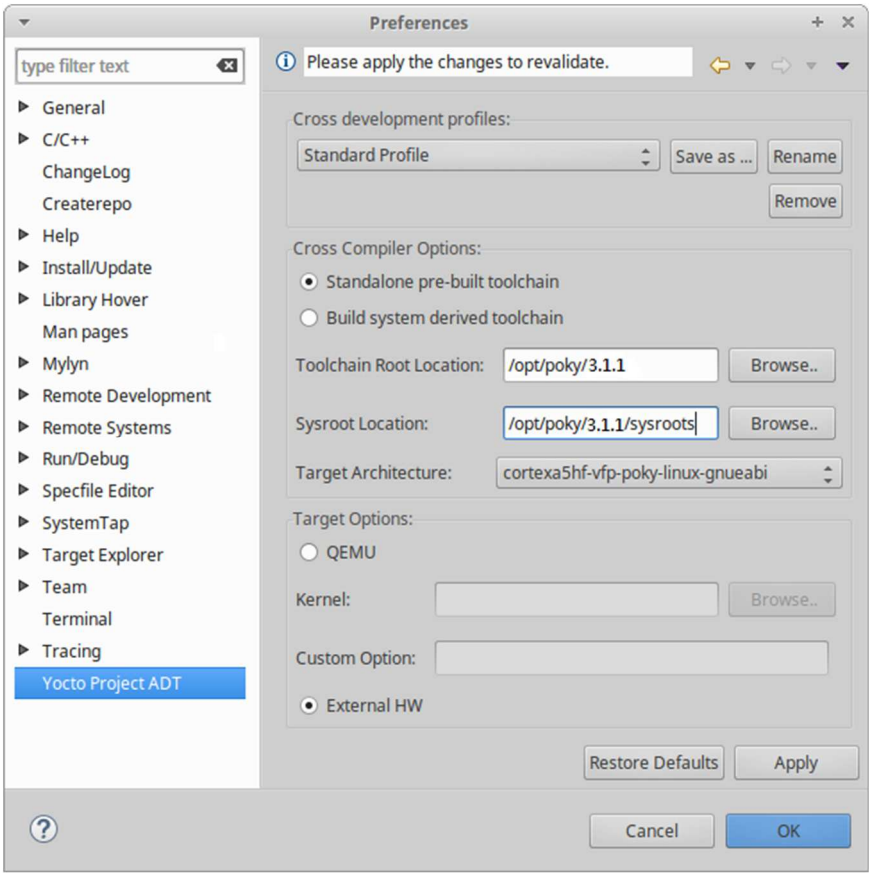
Step1, From the “Windows” menu select “Preferences”

Step2, Click "Yocto Project ADT" to display the configuration screen

Step3, Selecting the Toolchain Type: **Standalone pre-built toolchain**

- Point to the Toolchain: **/opt/poky/3.1.1**
- Specify the Sysroot Location: **/opt/poky/3.1.1/sysroots**
- Select the Target Architecture: **cortexa7hf-vfp-poky-linux-gnueabi**





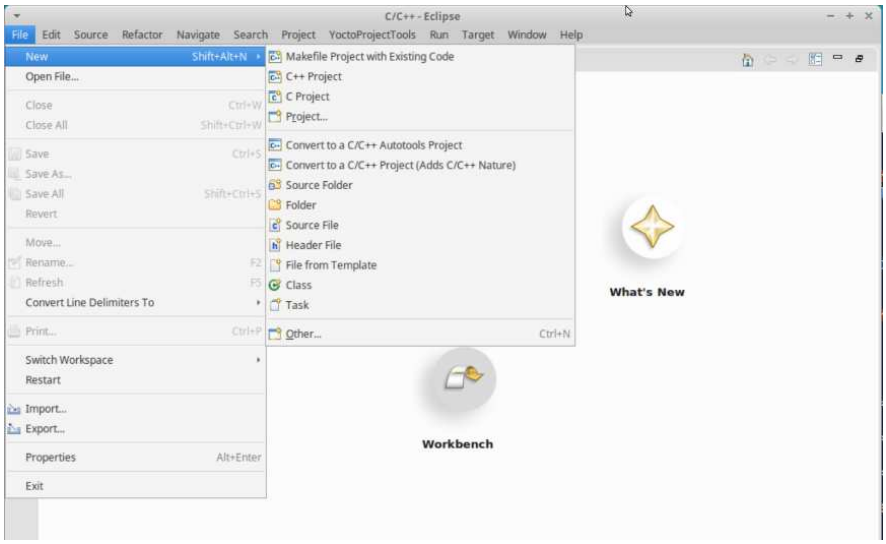
21.4 Create a Hello World Project

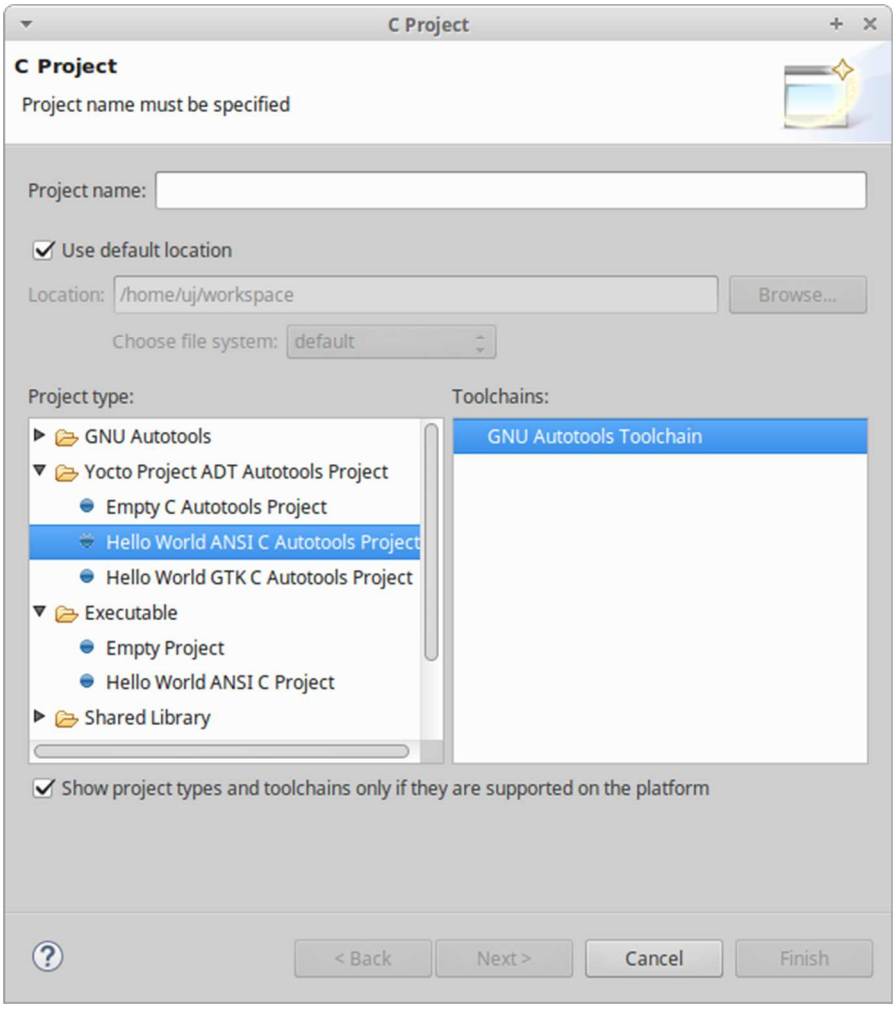
Step1, Select "Project" from the "File -> New" menu

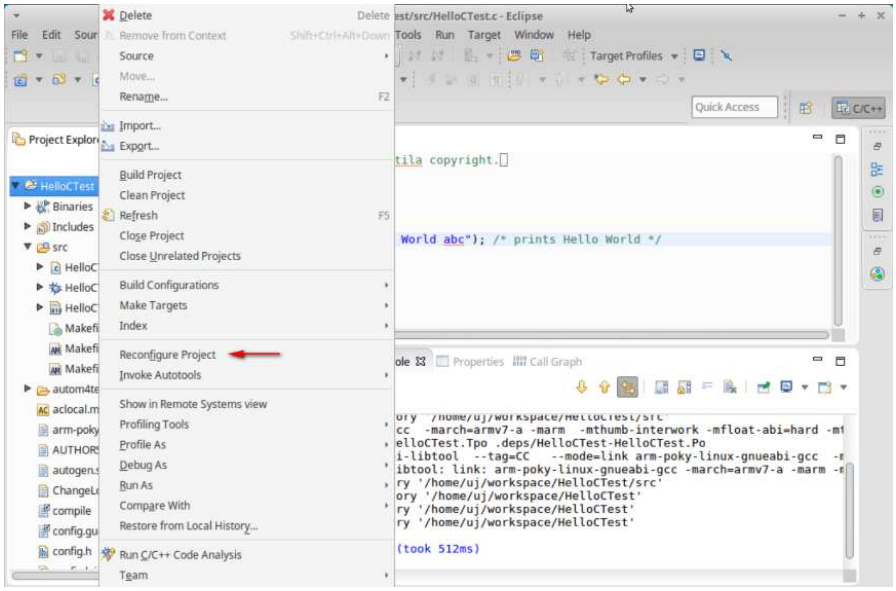
- Double click C/C++
- Double click C Project to create the project
- Expand Yocto Project ADT Autotools Project
  select Hello World ANSI C Autotools Project.
  (This is an Autotools-based project based on a Yocto template)
- Put a name in the Project name field.
  (Do not use hyphens as part of the name)
- Click "Next".
- Add information in the Author and Copyright notice fields.
- Click "Finish".

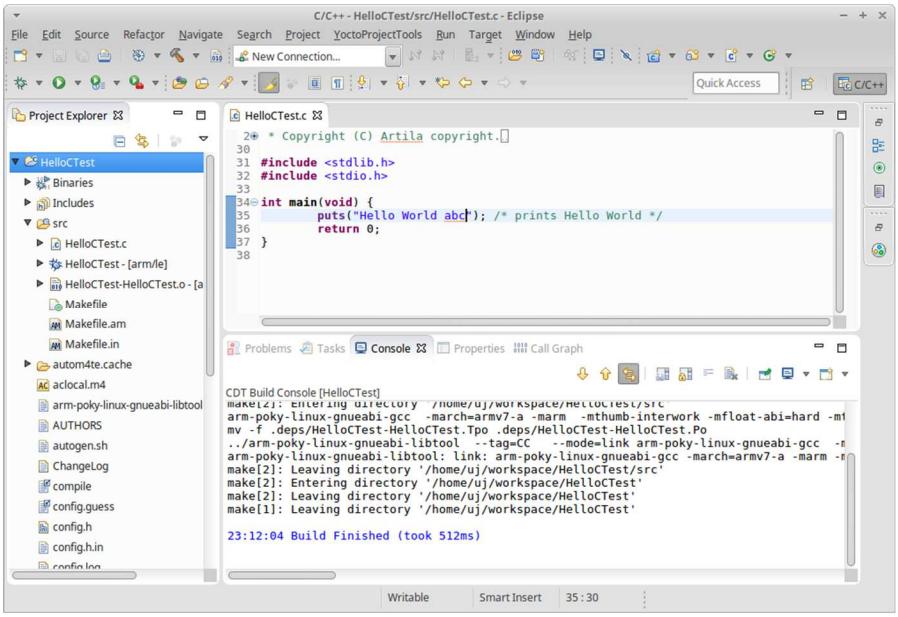
Step2, Right-click in the navigation pane and select "Reconfigure Project" from the pop-up menu. This selection reconfigures the project by running autogen.sh in the workspace for your project.

Step3, To build the project select "Build Project" from the "Project" menu.






Delete
Remove from Context
Source
Move...
Rename...
Import...
Export...
Build Project
Clean Project
Refresh
Close Project
Close Unrelated Projects
Build Configurations
Make Targets
Index
Reconfigure Project
Invoke Autotools
Show in Remote Systems view
Profiling Tools
Profile As
Debug As
Run As
Compare With
Restore from Local History...
Run C/C++ Code Analysis
Team
Target Profiles
Quick Access
C/C++
HelloCTest
Binaries
Includes
src
World abc"); /* prints Hello World */
Properties
Call Graph
(took 512ms)


C/C++ - HelloCTest/src/HelloCTest.c - Eclipse
File Edit Source Refactor Navigate Search Project YoctoProjectTools Run Target Window Help
New Connection...
Quick Access
C/C++
Project Explorer
HelloCTest
Binaries
Includes
src
HelloCTest.c
HelloCTest - [arm/le]
HelloCTest-HelloCTest.o - [a
Makefile
Makefile.am
Makefile.in
autom4te.cache
aclocal.m4
arm-poky-linux-gnueabi-libtool
AUTHORS
autogen.sh
ChangeLog
compile
config.guess
config.h
config.h.in
HelloCTest.c
* Copyright (C) Artila copyright.
#include <stdlib.h>
#include <stdio.h>
int main(void) {
puts("Hello World abc"); /* prints Hello World */
return 0;
}
Problems
Tasks
Console
Properties
Call Graph
CDT Build Console [HelloCTest]
arm-poky-linux-gnueabi-gcc -march=armv7-a -marm -mthumb-interwork -mfloat-abi=hard -mf
mv -f .deps/HelloCTest-HelloCTest.Tpo .deps/HelloCTest-HelloCTest.Po
../arm-poky-linux-gnueabi-libtool --tag=CC --mode=link arm-poky-linux-gnueabi-gcc -m
arm-poky-linux-gnueabi-libtool: link: arm-poky-linux-gnueabi-gcc -march=armv7-a -marm -m
make[2]: Leaving directory '/home/uj/workspace/HelloCTest/src'
make[2]: Entering directory '/home/uj/workspace/HelloCTest'
make[2]: Leaving directory '/home/uj/workspace/HelloCTest'
make[1]: Leaving directory '/home/uj/workspace/HelloCTest'
23:12:04 Build Finished (took 512ms)
Writable
Smart Insert
35 : 30

22. Setup SIM card

The CORTEX-A7 SERIES based Matrix IoT gateways, Model Matrix-752 & SBC-7530, support the miniPCIe slot can be install LTE/4G module for communication.
SIM Card setting is necessary before access by following:

NOTICE: Please unlock SIM PIN code first

Example: SIM7100E

apt install -y kernel-module-option kernel-module-qmi-wwan kernel-module-cdc-wdm kernel-module-usbnet libqmi

reboot

usb-devices

```
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  2 Spd=480 MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  2
P:  Vendor=1e0e ProdID=9001 Rev=02.32
S:  Manufacturer=SimTech, Incorporated
S:  Product=SimTech, Incorporated
S:  SerialNumber=0123456789ABCDEF
C:  #Ifs= 7 Cfg#= 1 Atr=80 MxPwr=500mA
I:  If#=0x0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=option
I:  If#=0x1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
I:  If#=0x2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
I:  If#=0x3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
I:  If#=0x4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=option
I:  If#=0x5 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=qmi_wwan
I:  If#=0x6 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=42 Prot=01 Driver=(none)
```

/etc/qmi-network.conf

```
APN=internet
```

/etc/network/interfaces/

```
auto wwan0
iface wwan0 inet dhcp
    pre-up qmi-network /dev/cdc-wdm0 start
    pre-down qmi-network /dev/cdc-wdm0 stop
```

Verify that SIM is inserted properly

echo -ne "AT+CPIN?\r" | eval /usr/bin/microcom -t 500 /dev/ttyUSB2

```
AT+CPIN?
+CPIN: READY

OK
```

Get signal strength

echo -ne "AT+CSQ\r" | eval /usr/bin/microcom -t 500 /dev/ttyUSB2

```
AT+CSQ
+CSQ: 17,99


OK
```

23. Setup “STATUS” LED indicator

The CORTEX-A7 SERIES based Matrix IoT gateways, Model SBC-7530, support three “STATUS” LED indicators for user definition control by GPIO4.IO27,GPIO4.IO28

Example 1, Enable and turn on the LED1

echo 0 > /sys/class/leds/user1/brightness

Example 2, Turn off and disable the LED1

echo 1 > /sys/class/leds/user1/brightness

24. Setup Digital Input / Digital Output

The **CORTEX-A7 SERIES** based Matrix IoT gateways, Model Matrix-752 & SBC-7530, support digital input and digital output port.

Matrix-752
DI / Digital Input (GPIO: GPIO4.IO23, GPIO4.IO25)
DO / Digital Output (GPIO: GPIO1.IO31, GPIO1.IO30)

SBC-7530
DI / Digital Input (GPIO: GPIO4.IO23, GPIO4.IO25)
DO / Digital Output (GPIO: GPIO4.IO19)

Example 1, Read value of DI1

cat /gpio/DI1/value

Example 2, Set High of DO1

echo 0 > /gpio/DO1/value

25. Setup CAN port

The CORTEX-A7 SERIES based Matrix IoT gateways support CAN port that can be setup as following:

Command:

apt-get install apt-get -y install kernel-module-can-raw kernel-module-can can-utils

Baudrate setting:

ip link set can0 up type can bitrate 125000

Example:

read: ***candump can0***

send: ***cansend can0 7df#02010d***

26. Setup Audio Out

The CORTEX-A7 SERIES based Matrix IoT gateway, Model SBC-7530 support one Audio out as line-out R/L port, optional earphone R/L.

Command: ***aplay (support format type: voc, wav, raw or au)***

Example: ***aplay sample.wav***

27. Setup Desktop Environment

The CORTEX-A7 SERIES based Matrix IoT gateway, Model SBC-7530 and M-X6ULL-B support “MatchBOX” as desktop environment.

Command: ***apt install -y packagegroup-core-x11 packagegroup-core-x11-matchbox matchbox-wm***

28. Restore to Factory Default

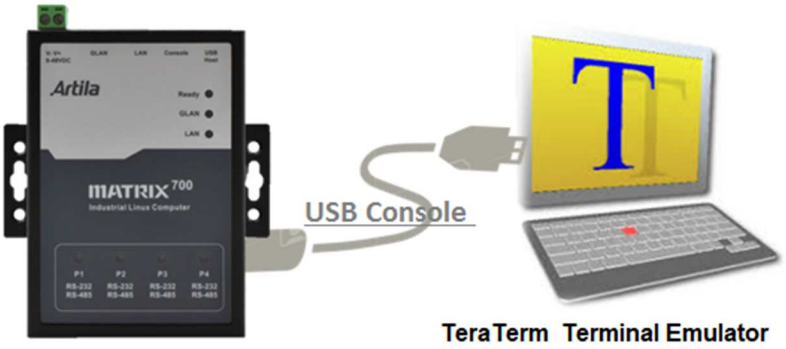
The following information shows how to restore to factory default:

Step 1,

Power-on Matrix-752/SBC-7530/M-X6ULL-B (“Ready” LED in green)

After system ready (“Ready” LED turns to yellow),

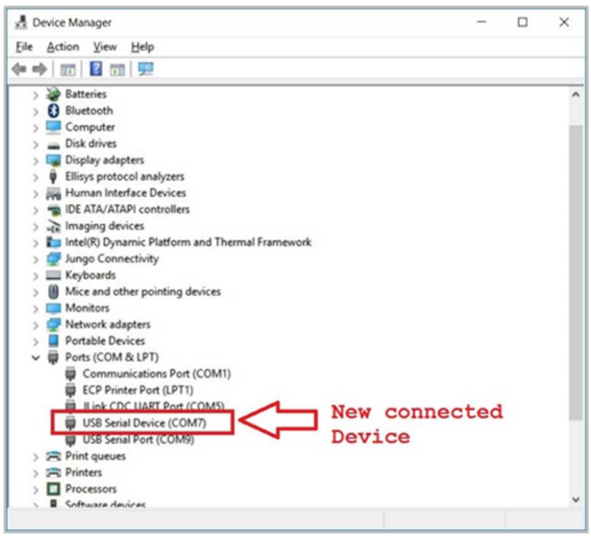
Connect the Matrix-752/SBC-7530/M-X6ULL-B’s USB/serial console to a PC



Step 2,

It shows *USB Serial Device “COMx” at Device Manager / PC* Windows:

X : depends on your PC’s environment



Communicate via this Serial device (EX. By TeraTerm @115200,N81).

Step 3,

Execute the command “**restore factory” and “y”**, this command will force Matrix-752/SBC-7530/M-X6ULL-B to boot again to start the restore process.

```
# restore factory
Restore from factory, Sure?(y/n)
y
……
```

The **READY** LED will be blinking during the restore process

This process may take 20~30minutes.

USB/serial console is NOT accessible during the restore process.

After completed above steps, the Matrix-752/SBC-7530/M-X6ULL-B will automatically reboot again when the backup process is finished.

29. Backup the “File System”

The following information shows how to restore to factory default:

Step 1,

Insert an empty 16GB(or above) USB drive (FAT32 format). Let’s say the USB drive maps to **/dev/sda1**.

Step 2,

Power-on Matrix-752/SBC-7530/M-X6ULL-B (“Ready” LED in green)

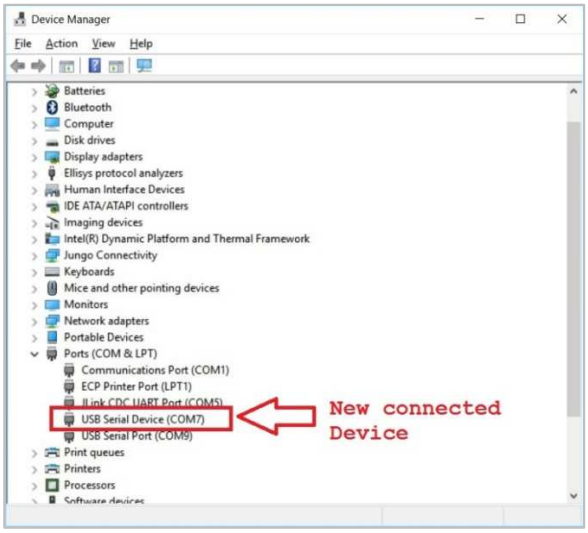
After system ready (“Ready” LED turns to yellow),

Connect the Matrix-752/SBC-7530/M-X6ULL-B’s USB/serial console to a PC

Step 3,

It shows *USB Serial Device "COMx" at Device Manager / PC* Windows:

X : depends on your PC's environment



Communicate via this Serial device (EX. By TeraTerm @115200,N81).

Step 4,

Execute the command "**backup /dev/sda1"** and **"y"**, this command will force Matrix-752/SBC-7530/M-X6ULL-B to boot again to start the backup process.

```
# backup /dev/sda1
Backup to /dev/sda1, Sure?(y/n)
y
…….
```

The **READY** LED will be blinking during the backup process

This process may take **20~30** minutes

USB/serial console is NOT accessible during the backup process

After completed above steps, the Matrix-752/SBC-7530/M-X6ULL-B will automatically reboot again when the backup process is finished.

Remove the USB drive and keep it properly as the golden copy.

30. Restore the “File System”

The following information shows how to restore to factory default:

Step 1,

Insert the golden copy USB drive. Let’s say the USB drive maps to **/dev/sda1**.

Step 2,

Power-on Matrix-752/SBC-7530/M-X6ULL-B (“Ready” LED in green)

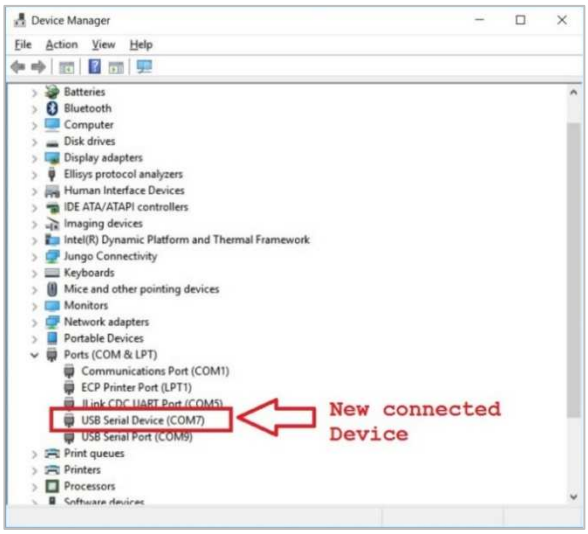
After system ready (“Ready” LED turns to yellow)

Connect the Matrix-752/SBC-7530/M-X6ULL-B’s USB/serial console to a PC

Step 3,

It shows *USB Serial Device "COMx" at Device Manager / PC* Windows:

X : depends on your PC's environment



Communicate via this Serial device (EX. By TeraTerm @115200,N81).

Step 4,

Execute the command "**restore /dev/sda1"** and **"y"**, this command will force Matrix-752/SBC-7530/M-X6ULL-B to boot again to start the restore process.

```
# restore /dev/sda1
Restore from /dev/sda1, Sure?(y/n)
y
……
```

The **READY** LED will be blinking during the restore process,

This process may take 20~30minutes

USB/serial console is NOT accessible during the restore process.

After completed above steps, the Matrix-752/SBC-7530/M-X6ULL-B will automatically reboot again when the restore process is finished.

Remove the USB drive.