

SB35 Development Platform for Yocto Linux User Guide

June. 2023

Version 0.1

Document Revisions

Date	Revision Number	Document Changes
06/2023	0.1	Initial version

Table of Contents

1	Overview	4
2	Setting up your computer.....	4
3	Setting up the repo utility	4
4	Building the Yocto Linux for SB35	5
4.1	Downloading the i350 source code.....	5
4.2	Get SB35 patch file from Innocomm.....	5
4.3	Building Yocto Linux images.....	5
4.4	Install AIoT Tools.....	5
5	Flashing SB35 image.....	6
6	Booting SB35.....	9
7	Serial console terminal	10
8	Function verify	10
9	Appendix	13
10	Reference	14

1 Overview

This tutorial guides new developers how to build Yocto Linux with the MTK i350 based development platform – SB35 board. It provides manuals for:

- Setting up a Linux® OS build machine.
- Building SB35 Yocto Linux images.
- Flashing the images to SB35 development board.

2 Setting up your computer

To build the Yocto Linux source files, you need a 64-bit version of Ubuntu (18.04 is recommended. But compatible with 16.04).

After installing the computer running Linux OS, check whether all the necessary packages are installed.

```
$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-  
multilib build-essential chrpath socat cpio python python3 python3-pip  
python3-pexpect xz-utils debianutils iputils-ping python3-git python3-  
jinja2 libegl1-mesa libsdl1.2-dev pylint3 xterm g++ libstdc++6  
lib32stdc++6 libpulse-dev libevent-dev ninja-build rpm2cpio libswitch-  
perl
```

gn tools install

```
$ sudo apt-get install libssl-dev  
$ wget -O gn http://storage.googleapis.com/chromium-  
gn/3fd43e5e0dcc674f0a0c004ec290d04bb2e1c60e  
$ sudo chmod 777 gn
```

Put the **gn** in build server `/usr/bin/`

Install adb and Fastboot

```
$ sudo apt-get install android-tools-adb android-tools-fastboot
```

Note: If your Ubuntu use default dash shell, please install bash shell to build image.

3 Setting up the repo utility

Repo is a tool built on top of Git that makes it easier to manage projects that contain multiple repositories, which do not need to be on the same server. Repo complements very well the layered nature of the Yocto Project, making it easier for users to add their own layers to the BSP.

To install the “repo” utility, perform these steps:

1. Create a bin folder in the home directory.

```
$ mkdir ~/bin (this step may not be needed if the bin folder already exists)  
$ curl https://storage.googleapis.com/git-repo-downloads/repo >  
~/bin/repo  
$ chmod a+x ~/bin/repo
```

2. Add the following line to the `.bashrc` file to ensure that the `~/bin` folder is in your PATH variable.

```
$ export PATH=~/bin:$PATH
```

4 Building the Yocto Linux for SB35

4.1 Downloading the i350 source code

Download MTK i350 source code from Gitlab server.

```
$ mkdir i350-sb35
$ cd i350-sb35
$ repo init -u https://gitlab.com/mediatek/aiot/bsp/manifest.git -b
refs/tags/rity-kirkstone-v23.0 -m default.xml --no-repo-verify
$ repo sync
$ ls
src
```

4.2 Get SB35 patch file from Innocomm

Please contact your Innocomm contact window to download the SB35 patch file

Note: Innocomm SB35 patch only supports rity-kirkstone-v23.0 version.

4.3 Building Yocto Linux images

- **Full build**

```
$ cd i350-sb35
$ patch -p1 < inno_for_sb35_rity-kirkstone-v23.0-v001.patch
$ patch -p1 < inno_for_sb35_rity-kirkstone-v23.0-v002.patch
$ export PROJ_ROOT=`pwd`
$ export TEMPLATECONF=$PROJ_ROOT/src/meta-rity/meta/conf/
$ source src/poky/oe-init-build-env build
$ DISTRO=rity-demo bitbake rity-demo-image
```

Note:

If you want a clean build, remove the build and sstate-cache folder, and start from the beginning.

The images will be located in the folder below, if the build is successful.

```
build/tmp/deploy/images/i350-sb35
```

Note: All files under this folder are needed to flash the board

4.4 Install AIoT Tools

Before flashing the image to SB35 EVB, you need to install AIoT tools first.

Git

```
$ add-apt-repository ppa:git-core/ppa
$ apt update
$ apt-get install git
```

Python3

If you don't have Python and pip installed on your Ubuntu, run the following commands to install them.

```
$ sudo apt update
$ sudo apt-get install python3
$ sudo apt-get install python3-pip
```

You can check Python and pip versions by running the following commands:

```
$ python3 --version
Python 3.9.2
$ pip3 --version
pip 21.2.4 from /usr/bin/pip3 (python 3.9)
```

If your version of pip3 is older than 20.3. Please upgrade it by running:

```
$ pip3 install --upgrade pip
```

USB Device rules

Add new udev rule and add your user account to **plugdev** group:

```
$ echo 'SUBSYSTEM=="usb", ATTR{idVendor}=="0e8d", ATTR{idProduct}=="201c",
MODE="0660", $ GROUP="plugdev"' | sudo tee -a /etc/udev/rules.d/96-
rity.rules
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
$ sudo usermod -a -G plugdev $USER
```

udev is a device manager for the Linux kernel. It can be used to grant a user or a group permission to access a device. In this case, we added an udev rule to assign **plugdev** group to the newly created SoC USB device. In order to access the SoC, your user account must be a member of the **plugdev** group.

AIoT Tools

After finishing installation of required packages and necessary configuration, we are now ready to install AIoT Tools:

```
$ pip3 install -U -e "git+https://gitlab.com/mediatek/aiot/bsp/aiot-
tools.git@1.3.0#egg=aiot-tools"
```

More detail information can reference website in below.

[Setup Tool Environment \(Linux\) — IoT Yocto documentation \(mediatek.gitlab.io\)](#)

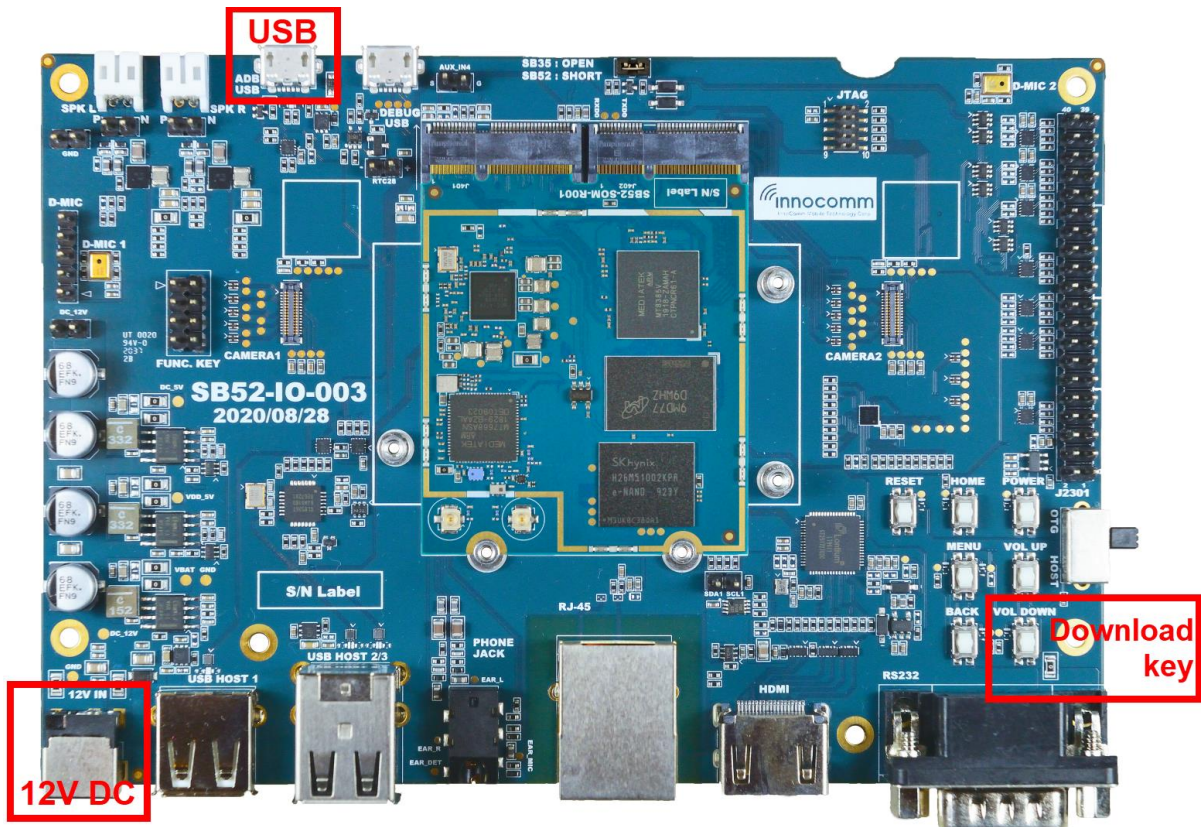
[AIoT Tools — IoT Yocto documentation \(mediatek.gitlab.io\)](#)

5 Flashing SB35 image

Use the command below to flash the board

```
$ aiot-flash --load-dtbo gpu-mali.dtbo --load-dtbo video.dtbo
```

- Full flash steps:



1. Plug out SB35 power cable
2. Plug in USB cable
3. Run the command

```
lion@ThinkCentre-M72e:~/images/i350-sb35$ aiot-flash --load-dtbo gpu-
mali.dtbo --load-dtbo video.dtbo
AIoT Tools: v1.3.dev24+ga10bc6a
Yocto Image:
  name:      Rity Demo Image (rity-demo-image)
  distro:    Rity Demo Layer 23.0-release (rity-demo)
  codename:  kirkstone
  machine:   i350-sb35
  overlays:  ['gpu-mali.dtbo', 'video.dtbo']
```

WARNING:aiot:[Errno 0] Success: 'error creating GPIO chip iterator'

4. Press and hold **download key** (to enter download mode)
5. Plug in SB35 power cable. Image flash will start...

```
lion@ThinkCentre-M72e:~/images/i350-sb35$ aiot-flash --load-dtbo gpu-
mali.dtbo --load-dtbo video.dtbo
AIoT Tools: v1.3.dev24+ga10bc6a
Yocto Image:
  name:      Rity Demo Image (rity-demo-image)
  distro:    Rity Demo Layer 23.0-release (rity-demo)
  codename:  kirkstone
```

```
machine: i350-sb35  
overlays: ['gpu-mali.dtbo', 'video.dtbo']
```

```
WARNING:aiot:[Errno 0] Success: 'error creating GPIO chip iterator'  
Looking for MediaTek SoC matching USB device 0e8d:0003  
Opening /dev/ttyACM0 using baudrate=115200  
Connected to MediaTek MT8168 SoC  
Sending bootstrap to address: 0x201000  
Jumping to bootstrap at address 0x201000 in AArch64 mode  
erasing mmc0  
< waiting for any device >  
erasing 'mmc0'...  
(bootloader) request sz: 0x747c0000, real erase len: 0x0  
OKAY [ 1.112s]  
finished. total time: 1.112s  
erasing mmc0boot0  
erasing 'mmc0boot0'...  
(bootloader) request sz: 0x400000, real erase len: 0x400000  
OKAY [ 0.009s]  
finished. total time: 0.009s  
erasing mmc0boot1  
erasing 'mmc0boot1'...  
(bootloader) request sz: 0x400000, real erase len: 0x400000  
OKAY [ 0.008s]  
finished. total time: 0.008s  
flashing mmc0=rity-demo-image-i350-sb35.wic.img  
target reported max download size of 268435456 bytes  
sending sparse 'mmc0' 1/8 (242564 KB)...  
OKAY [ 7.600s]  
writing 'mmc0' 1/8...  
OKAY [ 8.454s]  
sending sparse 'mmc0' 2/8 (262140 KB)...  
OKAY [ 8.211s]  
writing 'mmc0' 2/8...  
OKAY [ 8.927s]  
sending sparse 'mmc0' 3/8 (262140 KB)...  
OKAY [ 8.207s]  
writing 'mmc0' 3/8...  
OKAY [ 8.557s]  
sending sparse 'mmc0' 4/8 (241312 KB)...  
OKAY [ 7.559s]  
writing 'mmc0' 4/8...  
OKAY [ 7.919s]  
sending sparse 'mmc0' 5/8 (262140 KB)...  
OKAY [ 8.208s]  
writing 'mmc0' 5/8...  
OKAY [ 8.550s]  
sending sparse 'mmc0' 6/8 (262140 KB)...  
OKAY [ 8.207s]  
writing 'mmc0' 6/8...  
OKAY [ 8.550s]  
sending sparse 'mmc0' 7/8 (262140 KB)...  
OKAY [ 8.205s]  
writing 'mmc0' 7/8...  
OKAY [ 8.537s]  
sending sparse 'mmc0' 8/8 (40508 KB)...  
OKAY [ 1.271s]  
writing 'mmc0' 8/8...  
OKAY [ 1.331s]
```



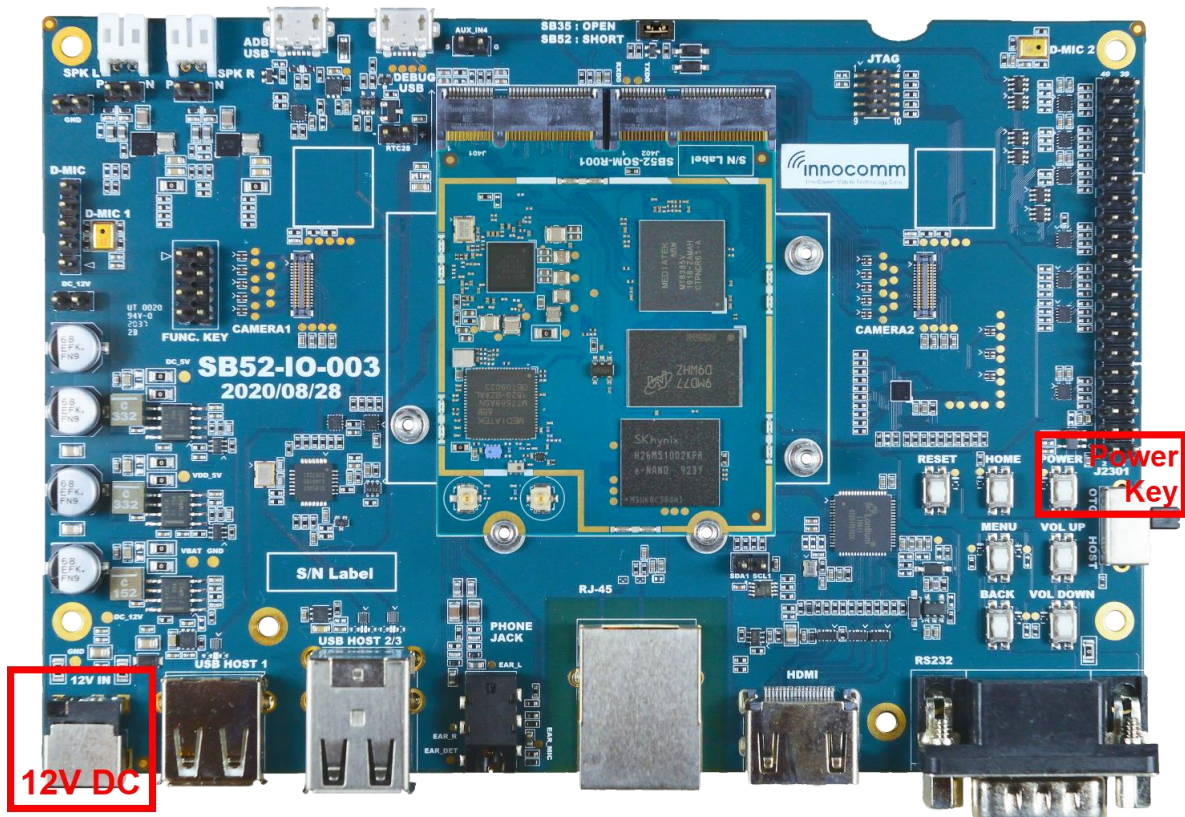
```
finished. total time: 118.294s
flashing mmc0boot0=b12.img
target reported max download size of 268435456 bytes
sending 'mmc0boot0' (136 KB)...
OKAY [ 0.006s]
writing 'mmc0boot0'...
OKAY [ 0.013s]
finished. total time: 0.019s
flashing mmc0boot1=u-boot-env.bin
target reported max download size of 268435456 bytes
sending 'mmc0boot1' (4 KB)...
OKAY [ 0.002s]
writing 'mmc0boot1'...
OKAY [ 0.006s]
finished. total time: 0.008s
rebooting...
```

```
finished. total time: 0.252s
```

```
lion@ThinkCentre-M72e:~/images/i350-sb35$
```

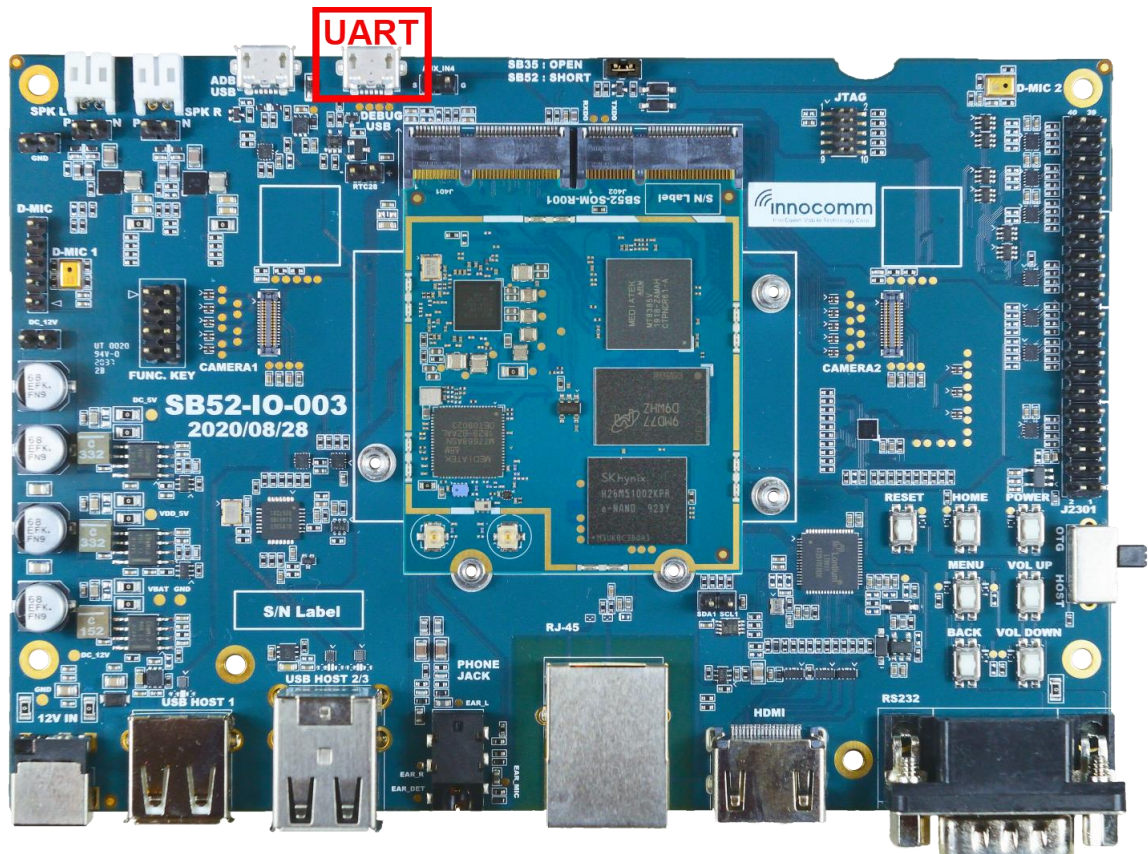
6 Booting SB35

SB35 development board is powered by 12V DC from DC Jack. Long press power key to boot SB35.



7 Serial console terminal

The serial console is a helpful tool for debugging your board and reviewing system log information. The console is the default output for kernel log messages (i.e., dmesg), and it also provides access to a full shell prompt that you can use to access commands such as logcat.



Recommended tools for serial communication terminal:

- **Putty** for Windows.
- **Minicom** for Ubuntu. (\$ sudo apt-get install minicom)

Configure the serial port as follows:

- Baud rate: 921600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Hardware Flow Control: No
- Software Flow Control: No

8 Function verify

● Shell command

1. ADB over USB

ADB is enabled by default in Yocto system. Input "adb shell" in your terminal console.

```
lion@ThinkCentre-M72e:~/images/i350-sb35$ adb devices
List of devices attached
```

```
0123456789 device
```

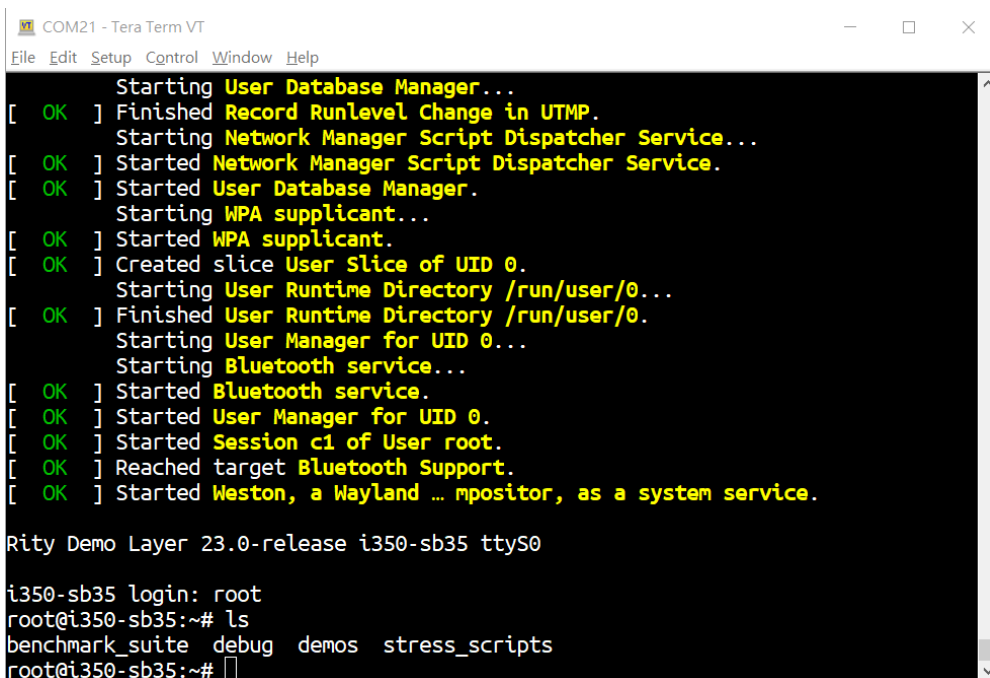
```
lion@ThinkCentre-M72e:~/images/i350-sb35$ adb shell
sh-5.1# ls -al
total 72
drwxr-xr-x 20 root root 4096 Jun  8  2023 .
drwxr-xr-x 20 root root 4096 Jun  8  2023 ..
drwxr-xr-x  2 root root 4096 Mar  9  2018 bin
drwxr-xr-x  2 root root 4096 Mar  9  2018 boot
drwxr-xr-x 19 root root 4120 Apr 28 17:42 dev
drwxr-xr-x 52 root root 4096 Apr 28 17:42 etc
drwxr-xr-x  4 root root 4096 Mar  9  2018 home
drwxr-xr-x 11 root root 4096 Mar  9  2018 lib
lrwxrwxrwx  1 root root    4 Mar  9  2018 lib64 -> lib/
drwx----- 2 root root 16384 Jun  8  2023 lost+found
drwxr-xr-x  2 root root 4096 Mar  9  2018 media
drwxr-xr-x  2 root root 4096 Mar  9  2018 mnt
drwxr-xr-x  3 root root 4096 Mar  9  2018 opt
dr-xr-xr-x 196 root root    0 Jan  1  1970 proc
drwxr-xr-x 17 root root  640 Apr 28 17:42 run
drwxr-xr-x  3 root root 4096 Mar  9  2018/sbin
drwxr-xr-x  2 root root 4096 Mar  9  2018/srv
dr-xr-xr-x 13 root root    0 Jan  1  1970/sys
drwxrwxrwt  9 root root  180 Apr 28 17:42/tmp
drwxr-xr-x 10 root root 4096 Mar  9  2018/usr
drwxr-xr-x  9 root root 4096 Apr 28 17:42/var
sh-5.1#
```

2. UART console

The i350 platform uses the UART0 and USB cable to establish serial communication with a PC.

Login as root:

Sb52 login: root



```
COM21 - Tera Term VT
File Edit Setup Control Window Help
Starting User Database Manager...
[ OK ] Finished Record Runlevel Change in UTMP.
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
[ OK ] Started User Database Manager.
Starting WPA supplicant...
[ OK ] Started WPA supplicant.
[ OK ] Created slice User Slice of UID 0.
Starting User Runtime Directory /run/user/0...
[ OK ] Finished User Runtime Directory /run/user/0.
Starting User Manager for UID 0...
Starting Bluetooth service...
[ OK ] Started Bluetooth service.
[ OK ] Started User Manager for UID 0.
[ OK ] Started Session c1 of User root.
[ OK ] Reached target Bluetooth Support.
[ OK ] Started Weston, a Wayland ... mpositor, as a system service.

Rity Demo Layer 23.0-release i350-sb35 ttyS0

i350-sb35 login: root
root@i350-sb35:~# ls
benchmark_suite debug demos stress_scripts
root@i350-sb35:~#
```

- **Audio**

Before you run the command you should put your sample wav file to SB35 EVB, here my sample file is test.wav

```
$ adb push test.wav /tmp/
```

1. Playback Audio (Output to Speaker)

```
aplay -D hw:0,1 /tmp/test.wav
```

2. Playback Audio (Output to Headphone)

```
aplay -D hw:0,0 /tmp/test.wav
```

3. Record Audio via Earphone

```
arecord -D hw:0,2 -r 16000 -c 2 -f S32_LE -d 10 /tmp/01.wav
```

4. Record Audio via on-board Microphones

```
arecord -D hw:0,3 -r 16000 -c 2 -f S32_LE -d 10 /tmp/02.wav
```

- **WiFi**

Setting SSID and Password to /etc/wpa_supplicant.conf , then run the command below.

```
wpa_supplicant -Dnl80211 -iwlan0 -c/etc/wpa_supplicant.conf &
```

- **Ehternet**

Plug ethernet cable to RJ45 connector will auto connect to internet.

- **RS232**

Loopback test Plug a RS232 loopback plug/connector to the RS232 DTE connector of SB52 andperform the test with linux-serial-test utility.

```
root@i350-sb35:~# linux-serial-test -c -o 1 -i 2 -e -p /dev/ttyS1 -b 115200
Linux serial test app
Opening /dev/ttyS1 without tty line discipline
Error setting RS-232 mode: Inappropriate ioctl for device
Stopped transmitting.
Stopped receiving.
/dev/ttyS1: count for this session: rx=4111, tx=4111, rx err=0
/dev/ttyS1: TIOCGICOUNT: ret=0, rx=4111, tx=4111, frame = 0, overrun = 0, parity = 0, brk = 0, buf_overrun = 0
```

- **SPI**

The data rate of SPI is up to 12MBps in Yocto Linux

Before the SPI test please short SB52 Jumper and keep it open when you test the other



functions except SPI.

And then short pin 19(MOSI) and 21(MISO) of J2301 connector on SB52 carrier board, and perform the loopback test with spidev_test utility

```
spidev_test -D /dev/spidev0.0 -s 12000000 -v -I 1
```

- Camera

Raw sensor is not support in Yocto Linux
 USB UVC camera is supported.

- HDMI

Just work when plugin the HDMI monitor user can operate the computer with USB mouse and keyboard via USB host port.

- RaspberryPi 7“ Display

Since RaspberryPi 7” Display has three model, we only support Part number : PH800480T013-IAC01 (D/5624KIT-NOLOGO) and touch is not work now

Part Description	Part number	Approved
LCD Module	PH800480T013-IAC01 (D/5624KIT-NOLOGO)	1 July 2015
LCD Module	YB-TG800480C43A-C-A1	1 November 2018
LCD Module	PH800480T013-IAC21 (D/5624/KIT-LC)	21 February 2019

More detail information please reference link in below.

https://www.raspberrypi.org/app/uploads/2019/04/RPI-PCN-Display_v.1-002.pdf

9 Appendix

- Video Playback Using GStreamer

To use the software codec:

```
gst-launch-1.0 -v filesrc location=<your-video-path> ! parsebin !
avdec_h264 ! \
videoscale ! video/x-raw,width=1920,height=1080 ! fpsdisplaysink video-
sink=waylandsink text-overlay=false
```

To use the hardware codec(not work now):

```
gst-launch-1.0 -v filesrc location=<your-video-path> ! parsebin !
v4l2h264dec ! \
v4l2convert output-io-mode=5 ! video/x-raw,width=1920,height=1080 !
fpsdisplaysink video-sink=waylandsink text-overlay=false
```

- JPEG Image Playback Using GStreamer

To use the software codec:

```
gst-launch-1.0 -v filesrc location= out-160x128-NV12.jpg ! jpegparse !
jpegdec ! waylandsink
```

To use the hardware codec (not work now):

```
gst-launch-1.0 -v filesrc location=out-160x128-NV12.jpg ! jpegparse !  
v4l2jpegdec ! waylandsink
```

10 Reference

[MediaTek IoT Yocto Developer Guide — IoT Yocto documentation](#)