MangoPi Guide for Setting Up LVGL

Prepare the repo:

Get the latest repo with all the latest BSP: https://github.com/RT-Thread/rt-thread.git this is the repo I used. (My case I have saved the repo at the location: /rtsmart/rtthread-smart/userapps)
Also update the toolchain to latest using the command from the tools folder.
*:~/rtsmart/rtthread-smart/tools$ python3 get_toolchain.py riscv64*

Also make sure that the environment is set correctly. *source ./smart-env.sh riscv64* must be executed at the correct folder.
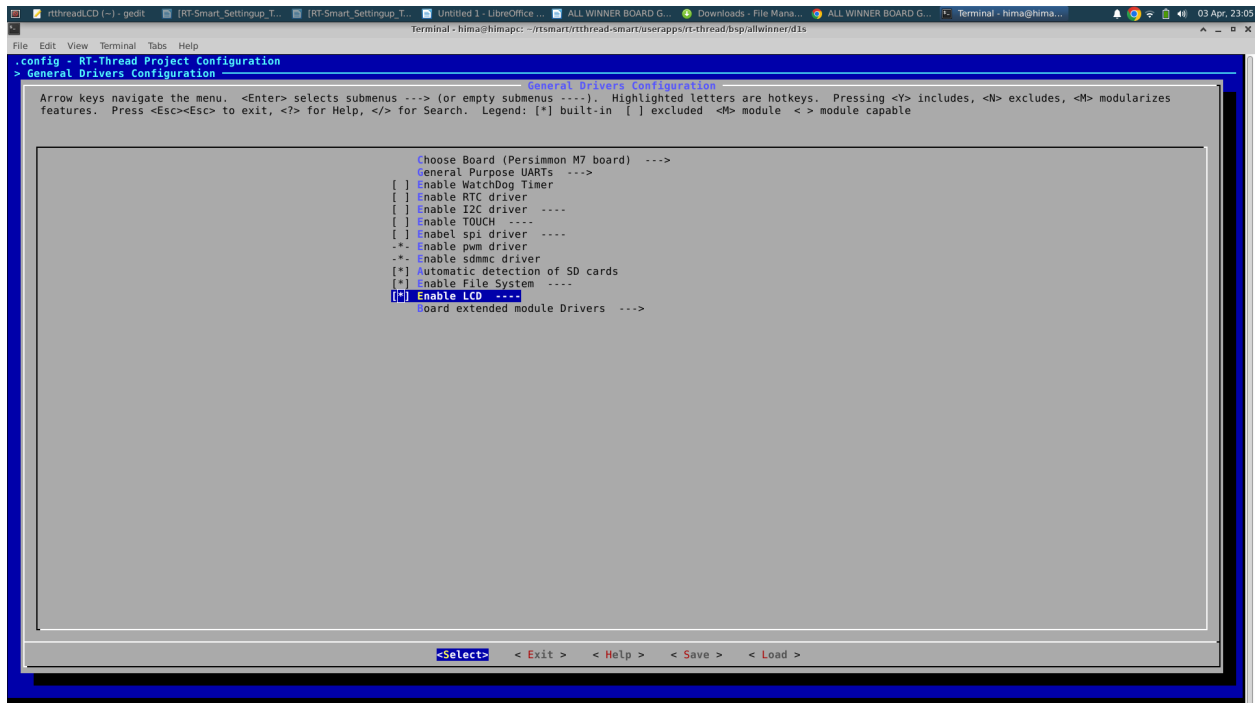
Build the BSP with LCD Enabled:

Build the BSP for the board: Now build the right BSP for your allwinner board. Navigate to the folder , in my case rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s

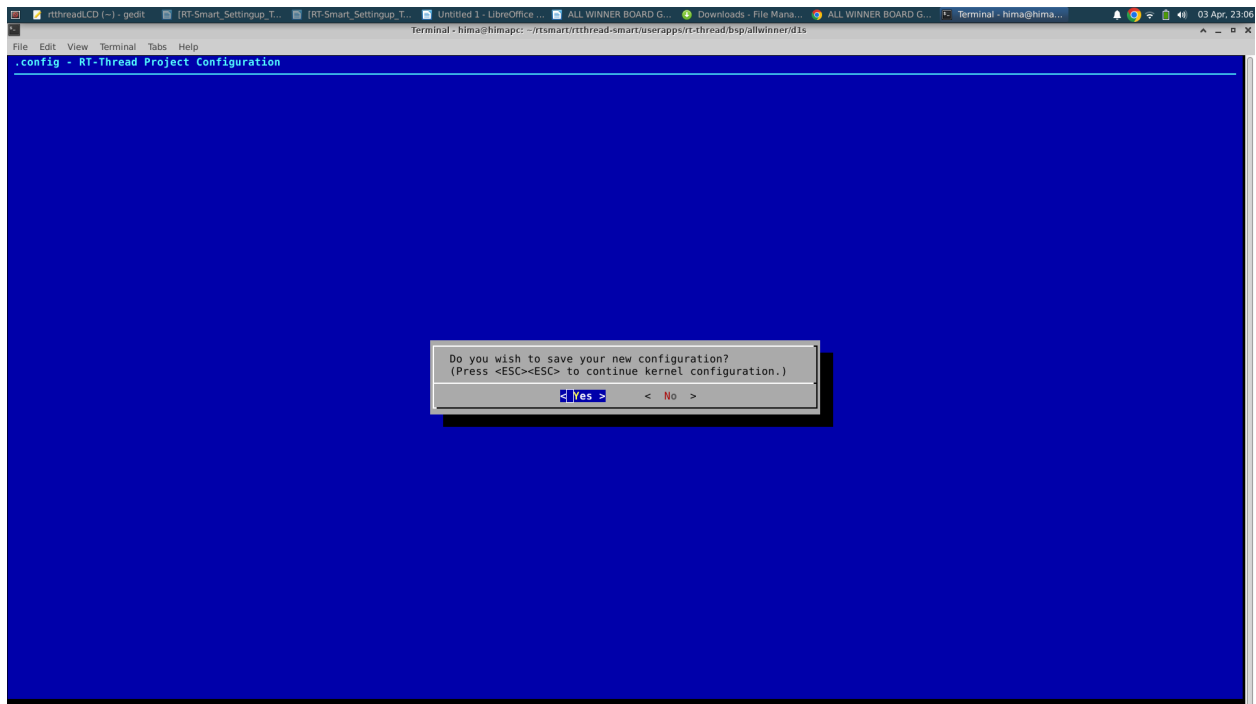/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s$ scons --menuconfig
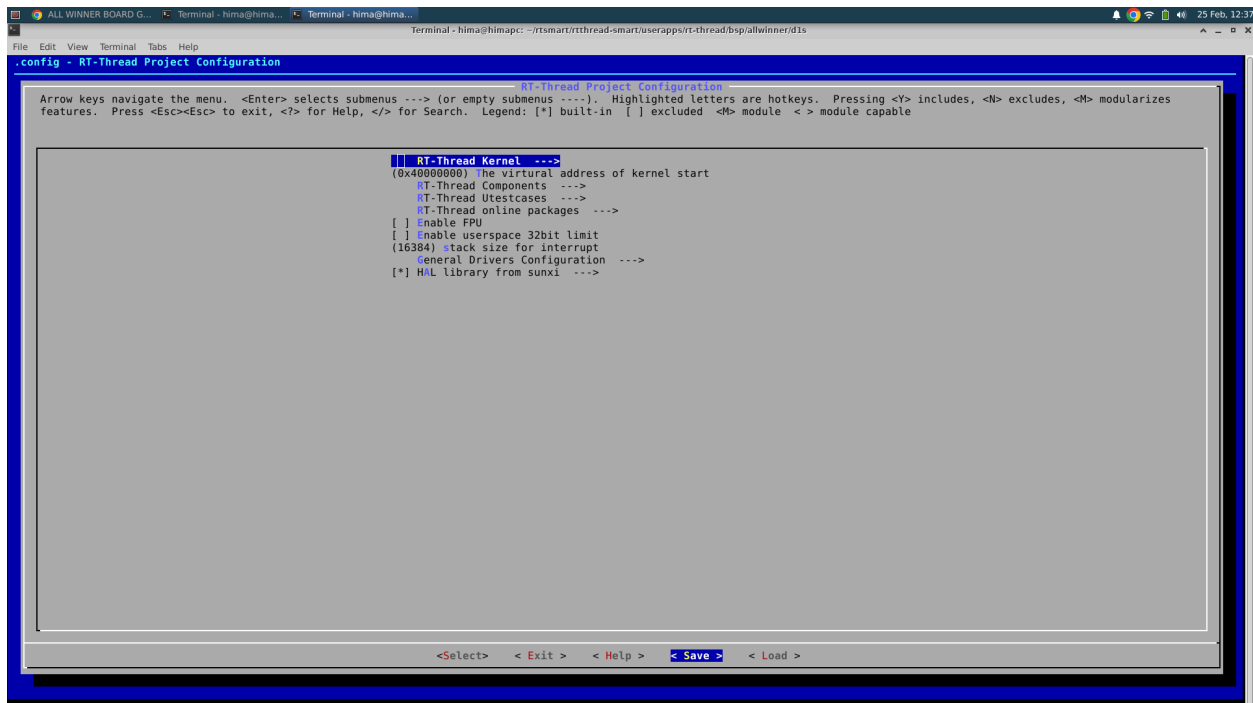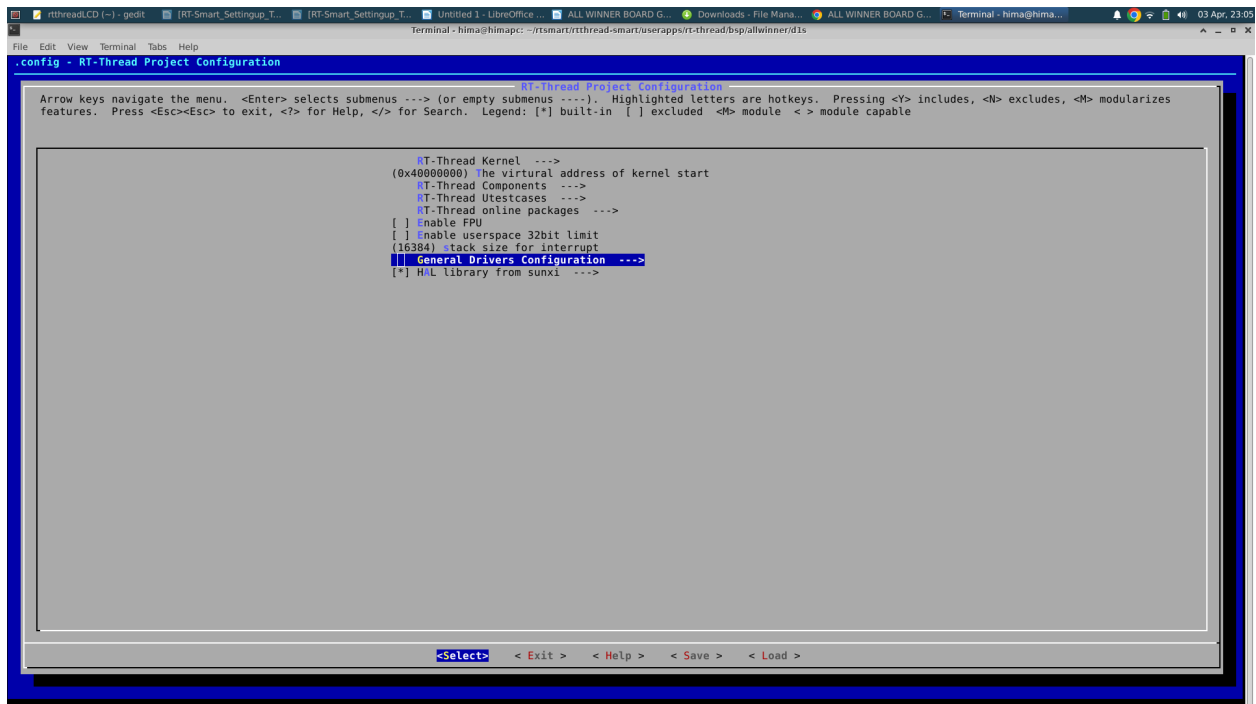
Select the "General Driver Configuration" Option

Select the "Enable LCD" optiion. To enable LCD you need press the key "Y".

For exiting the screen you need to press <Esc> <Esc> and repeat the same staep again. Then You will get a prompt for saving the configuration as shown in the image. Select "Yes" to save the file. Now LCD is enabled in the BSP. BSP should be build after this.

Execute the command "scons" to build the BSP.

```
 ~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s$ scons
……………..
……………..
 scons: done building targets.
```

Check for and sd.bin file which gets generated. This is the kernel image we are going to store in the SDCard, which will boot the RT-Thread kernel.

Preparing SD Card:

Connect the SD Card to your system. Find out the entry corresponding to your SD Card from /dev folder. For me the name was sdb.
Format the sdb disk using fdisk command.

```
:~$ sudo fdisk /dev/sdb
Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): o
Created a new DOS disklabel with disk identifier 0xbf06dddf.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-125542399, default 2048): 16384
Last sector, +/-sectors or +/-size{K,M,G,T,P} (16384-125542399, default 125542399):

Created a new partition 1 of type 'Linux' and of size 59,9 GiB.
Partition #1 contains a vfat signature.

Do you want to remove the signature? [Y]es/[N]o: Yes

The signature will be removed by a write command.

Command (m for help): q
```
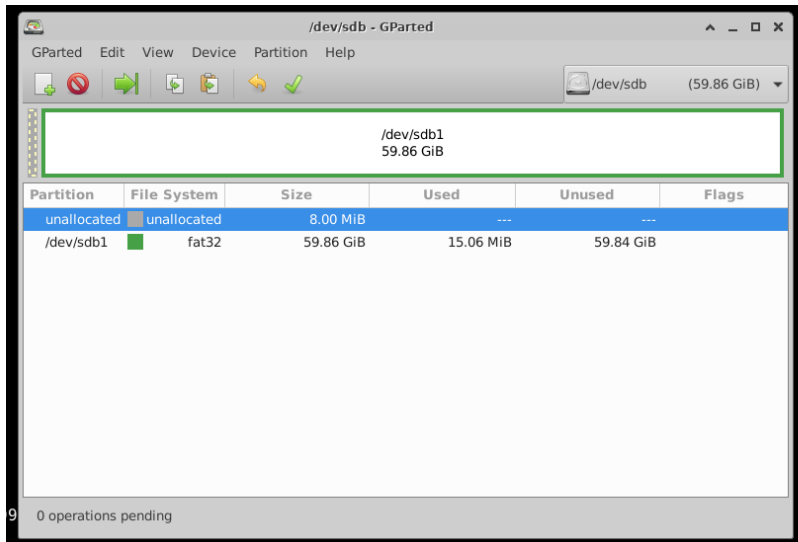
You can use *gparted* command to cross verify the partition.

Now Flash the sd.bin and boot image onto the SDCard. Use the following two commands from the right folder. Boot file is present inside the *rt-thread/bsp/allwinner/d1s/tools* folder and the sd.bin gets generated in the *rt-thread/bsp/allwinner/d1s* folder

sudo dd if=boot0_sdcard_sun20iw1p1_d1s.bin of=/dev/sdb bs=1024 seek=8
sudo dd if=sd.bin of=/dev/sdb bs=1024 seek=56

Command logs are followed.

hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s/tools$ sudo dd if=boot0_sdcard_sun20iw1p1_d1s.bin of=/dev/sdb bs=1024 seek=8
48+0 records in
48+0 records out
49152 bytes (49 kB, 48 KiB) copied, 0,0181441 s, 2,7 MB/s
hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s/tools$
hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s/tools$
hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s/tools$ cd ..
hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s$ sudo dd if=sd.bin of=/dev/sdb bs=1024 seek=56
772+0 records in
772+0 records out
790528 bytes (791 kB, 772 KiB) copied, 0,152477 s, 5,2 MB/s
hima@himapc:~/rtsmart/rtthread-smart/userapps/rt-thread/bsp/allwinner/d1s$

Build The User Application

The LVGL stands for light and Versatile Embedded Graphics Library. A sample application which shows graphics on the LCD, just to use as the demo is already aviailable in this git repo: https://github.com/Rbb666/RT-Smart-UserAPP

We are using this application. So download the repo to the folder userapps/media/lvgl

Now we need to edit Sconstruct file so that the lvgl application is also build. For that edit to the userapps/Sconstruct file and add the lvgl folder also to it , as shown in the following image

We are ready with the user apps. Now clean and build the user applications. Remember to run the smart-env.sh prior to the building.
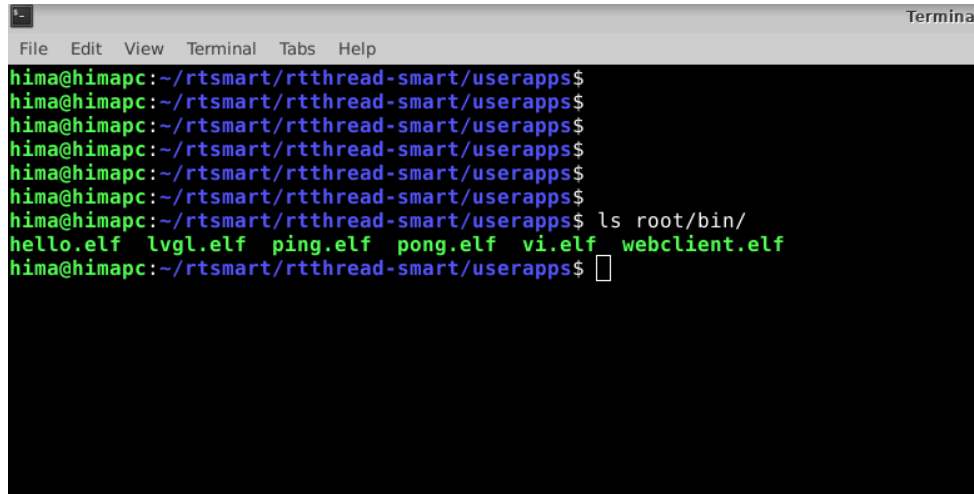
```
:~/rtsmart/rtthread-smart$ source smart-env.sh riscv64
Arch     => riscv64
CC       => gcc
PREFIX   => riscv64-unknown-linux-musl-
EXEC_PATH => /home/hima/rtsmart/rtthread-smart/tools/gnu_gcc/riscv64-linux-
musleabi_for_x86_64-pc-linux-gnu/bin

:~/rtsmart/rtthread-smart/userapps$ scons --clean
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Cleaning targets ...
Removed build/hello/main.o
Removed build/lvgl/lv_rtt_port/lv_demo.o
Removed build/lvgl/lv_rtt_port/lv_port_disp.o


...........................................................................
:~/rtsmart/rtthread-smart/userapps$ scons
scons: Reading SConscript files ...
scons: done reading SConscript files.
scons: Building targets ...
CC build/hello/main.o
CC build/lvgl/lv_rtt_port/lv_demo.o
CC build/lvgl/lv_rtt_port/lv_port_disp.o
................................................................................
CC media/lvgl/packages/LVGL-latest/src/widgets/win/lv_win.o
LINK root/bin/hello.elf
LINK root/bin/lvgl.elf
LINK root/bin/ping.elf
LINK root/bin/pong.elf
LINK root/bin/vi.elf
scons: done building targets.
```

The generated elf files are located in the root/bin folder. <u>We need to copy this bin folder also to the SD card. This can be just manually copied.</u>

<u>Setting Up The Hardware</u>

Plug the SD Card on to the allwinner Board, on the microSD Card slot.

Connect the LCD Module to the MangoPi board. I am using GL043026 Graphics LCD module. The connection is shown in the following figure.
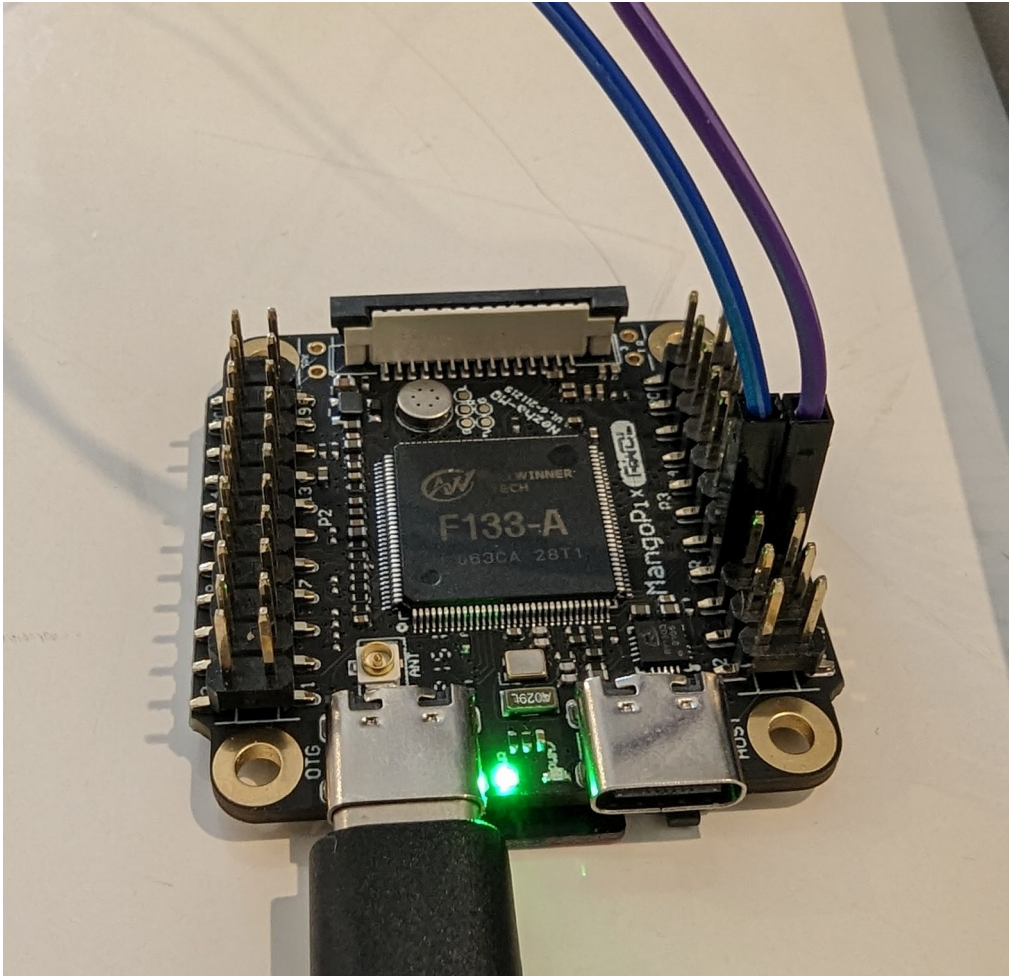
Also connect a UART-USB Bridge, so that you can check the UART Commands from your PC. I am using the CP2102 USB to TTL interface board as UART-USB Bridge.

RX (CP2102) → TX (P3.7) Allwinner
TX (CP2102)   → RX(P3.8) Allwinner

Refer to the following figure.

Check the UART messages on a suitable Terminal Application.
    Baud Rate: 500000

Open any Serial port terminal check if the board is booting properly. If everything works fine you can see the following messages on the serial port.


```
[00:02:21:190] [854]run_addr       = 0x40200000
[00:02:21:190] [855]index          = 0x0
[00:02:21:190] [856]Entry_end       = 0x3b454949
[00:02:21:199] [857]********************************************************
[00:02:21:199] [858]*******************TOC1 Item Message********************
[00:02:21:199] [860]Entry_name       = kernel
[00:02:21:199] [860]Entry_data_offset = 0x20600
[00:02:21:199] [861]Entry_data_len   = 0xe69c0
[00:02:21:199] [862]encrypt          = 0x0
[00:02:21:199] [863]Entry_type       = 0x0
[00:02:21:199] [863]run_addr         = 0x40400000
[00:02:21:199] [864]index            = 0x0
[00:02:21:206] [865]Entry_end        = 0x3b454949
[00:02:21:206] [865]********************************************************
[00:02:21:206] [868]Jump to second Boot.
[00:02:21:223]
[00:02:21:224] OpenSBI v0.9-167-gf45b7d4
[00:02:21:224]      ____                      ____    _ 
[00:02:21:230]     / __ \                    / ___|  _ \    |
[00:02:21:230]    | |  | |_ __   ___ _ __  | (___  | |_) | ||
[00:02:21:230]    | |  | | '_ \ / _ \ '_ \  \___ \ |  _ <  ||
[00:02:21:230]    | |__| | |_) |  __/ | | |  ___) || |_) | ||
[00:02:21:230]     \____/| .__/ \___|_| |_| |____/ |____/_|
[00:02:21:230]          | |
[00:02:21:236]          |_|
[00:02:21:236]
[00:02:21:245] Platform Name          : Allwinner D1 NeZha
[00:02:21:246] Platform Features       : medeleg
[00:02:21:251] Platform HART Count      : 1
[00:02:21:251] Platform IPI Device      : aclint-mswi
[00:02:21:251] Platform Timer Device    : aclint-mtimer @ 24000000Hz
[00:02:21:251] Platform Console Device  : uart8250
[00:02:21:251] Platform HSM Device      : ---
[00:02:21:251] Platform Reboot Device   : sunxi-wdt-reset
[00:02:21:258] Platform Shutdown Device : ---
```

```
Clear    ☐ Hex output  ☐ Logging to:  /root/cutecom.log

Device: /dev/ttyUSB0     Connection: 500000 @ 8-N-1
```

Now execute the bin/lvgl.elf application.

Upon execution, you can see a video being played on the graphics LCD Screen.