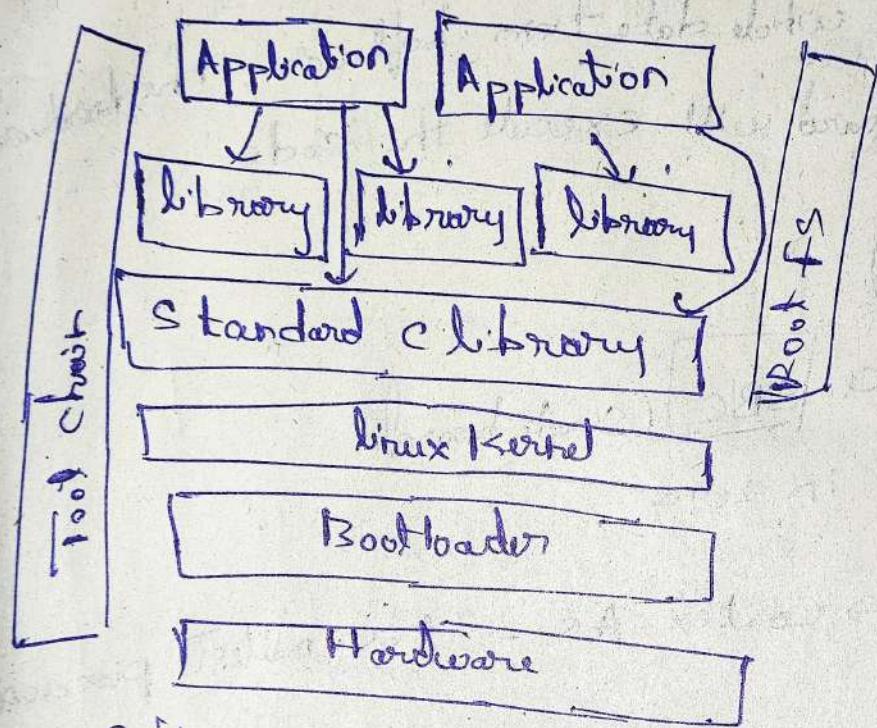


Porting

→ Porting means → Customizing Linux for given hardware.

Linux System Arch

→ Procedure for sending Image from Software i.e. Linux to hardware i.e. rugged board.



Software Components

① RootFS

It is the storage location of our image in software.

The image contains → Logical like 'vecto' type of language.

- (i) Application.
- (ii) Library.
- (iii) Standard C library.

These are present in stack

② Tool chain

To transmit data from software to hardware, we need to enable toolchain.

Hardware Components

(i) Linux Kernel

→ acts as an interface b/w Software & hardware.

(ii) Boot Loader

→ Receive data from the Software to hardware.

(iii) Hardware

→ After receiving whole data from software, the hardware i.e. Rugged Board will execute the code.

~~Libre~~

① Rugged board

→ It is a SBC (Single board)

introduced in 2019.

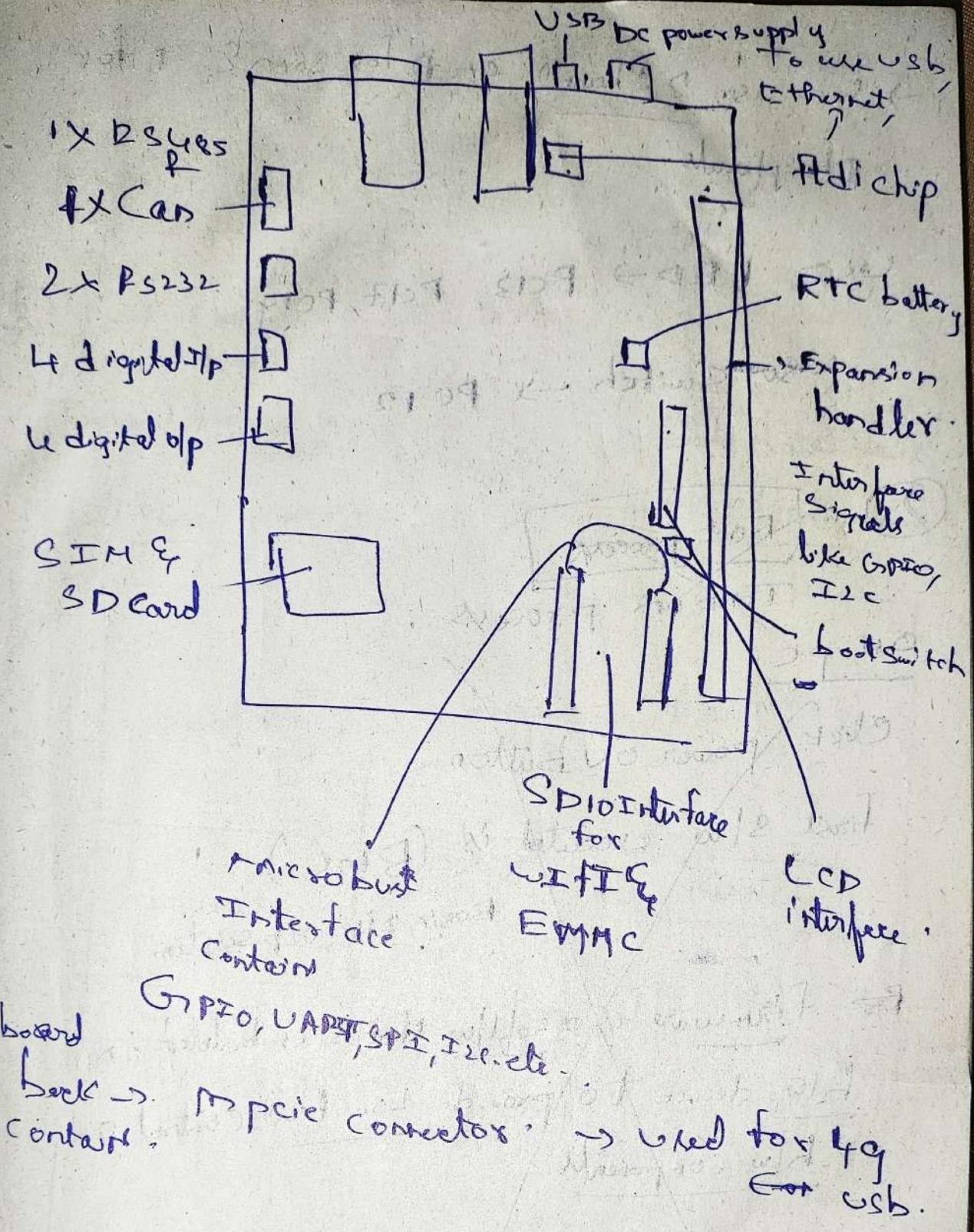
Processor → Cortex-A5 → it is smallest processor

Industrial variant → IoT, Embedded

RAM → 64 MB

ROM / Flash → 32 MB

Peripherals



→ It is first made by Phytec

→ By using Boot Switch we can select the
SD ~~or~~ flash (or) NOR flash

→ we have switching regulator which converts
5V to 3.3V

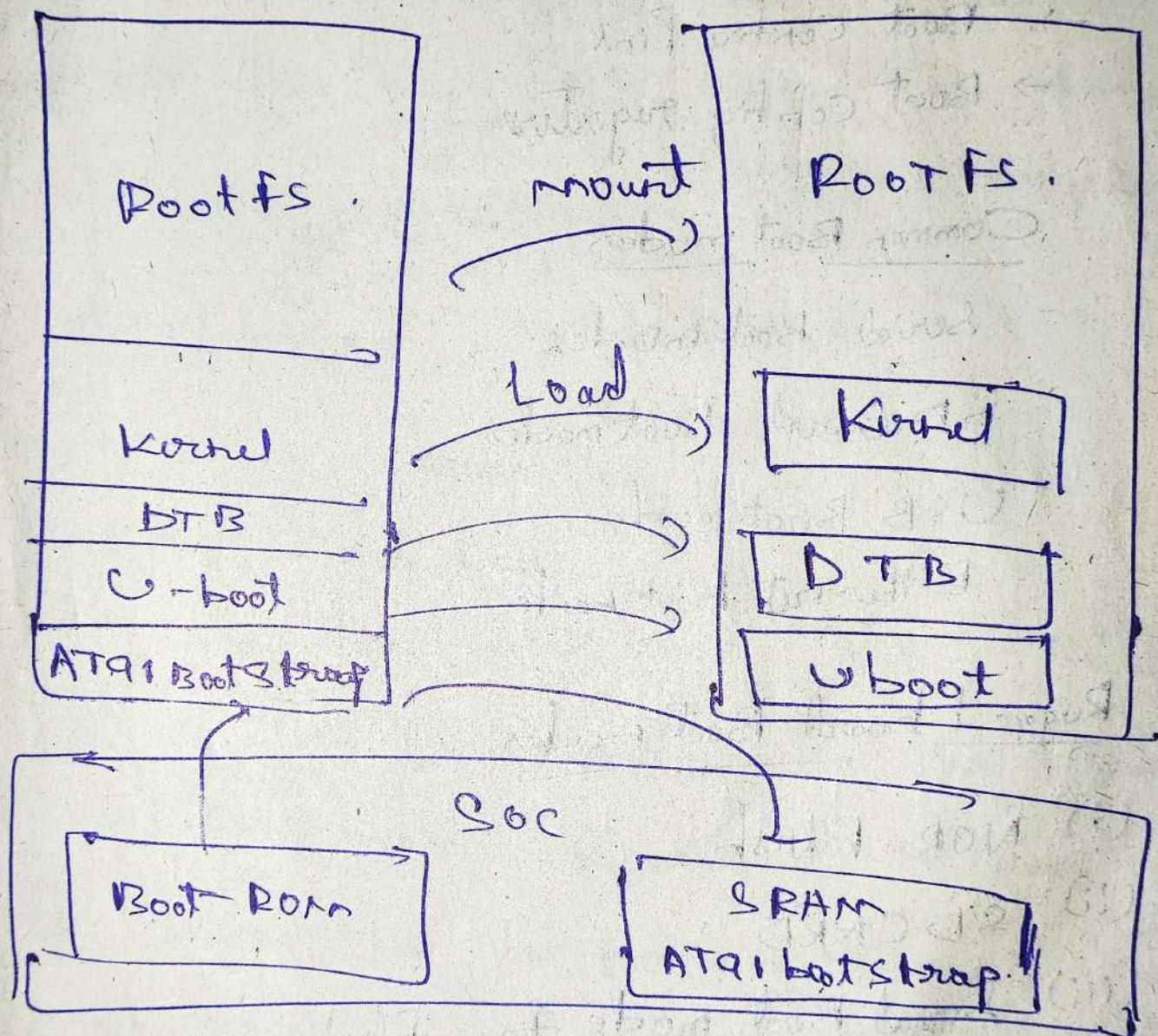
② Boot process

Rugged board

Flash memory

Double data Rate

DDR / Main memory



→ Power on

→ Boot ROM Loads AT91 bootstrap to
SRAM (Primary Boot Loader)

→ Primary Boot Loader initialized to DDR

- Primary Bootloader will Load Secondary Bootloader
i.e U-boot to DDR
- U-boot will Load DTB & Kernel
- Kernel mounts rootfs.

③

Boot Options

boot bypass

Boot mode of SOC

- Boot Config Pin
- Boot Config register

Common Boot modes

Serial Boot modes.

SD Card Boot modes

USB Boot mode.

Ethernet Boot mode.

Rugged board Boot modes

(i) NOR Flash

(ii) SD CARD

(iii) Serial Boot mode for flashing image

Using

SAM-BA Tool,

④ TFTP for flashing images & Binaries.

- Previously we use Serial boot mode for flashing images & Binaries
- ↓
Communication by using single (or) pair of wires.
- Now, we are using Ethernet & USB for fast communication.
- TFTP (Trivial file Transfer protocol) is used to transfer files b/w Two hosts using TCP network.
- ↓
(Transmission Control protocol)
- Client & Server
↓
Request Service Provide Service

UDP → is connectionless

TCP → connection

↳ 3 way Hand Shaking → ask .

① → SYN (Synchronize) (client to server)
initiates connection b/w client to server.

② SYN-ACK (client req to server)
Server ACK client request (server to client)
send

③ ACK (client to server) → receive
Client ACK Server, upon receiving data .

Package → combination of ~~file~~ file classes
modules for programming language

+ 12/24

④ Flashing images or RB (Serial)

→ connect board & power on click reset button on RB.

U-boot Flashing or RB (SD)

Means → changing U-boot image.

① To change board name in boot strap (NOR)

Go to github, tagged board.

download at91bootstrap 3.8.12

→ go to at91bootstrap - 3.8.12

Open in terminal

↳ Ctags -R * Create binary
 S S → To get binary file
 for changes

enable tool chain . 10 10 12 1c

↳ make myproper → To clear previous configurations

↑ au in gitHub, rb
\$ make rugged_board_asd2x0qspi-u-boot_

defconfig
(To say NOR flash)

{ vi .config

edit banner name (In Pradeep)

→ \$make (To save changes)

→ \$cd binaries.

→ \$18

→ Rename .bin file as BOOT.BIN

→ mv ru---.bin BOOT.BIN

{ CP BOOT.BIN /var/lib/tftpboot

Connect rugged board to TFTP

Reset button.

Enter

⇒ setenv ipaddr 192.168.1.10

⇒ setenv serverip 192.168.1.15

⇒ ping 192.168.1.15 → avail check. run

⇒ sf Probe → serial flash.

⇒ tftp 0x21FF0000 BOOT.BIN

Image to RAM.

⇒ set erase 0x20000 0x80000
erase already thor image

⇒ SF write 0x21FF008 - 0x40000 0x80000
Copus image
⇒ reset

for sd card flash
we need to change.

make rugged board -> dex_sd1_u-boot
ifconfig

sf probe commands

② To execute a file ^{(or) code} in using NOR

first goto Normal Terminal, To create exec file

\$ gcc file.c -o exec

\$ cp exec /lib/ftrboot

Connect rugged board (NOR)

\$ Root → cd /data

\$ ifconfig eth0 ~~serverip~~ ~~ipaddr~~
192.168.1.15 192.168.1.10

\$ ping 192.168.1.15. Type ipaddr

\$ tftp -r exec -s ~~serverip~~ 192.168.1.15

\$ chmod 777 exe

\$./exe .

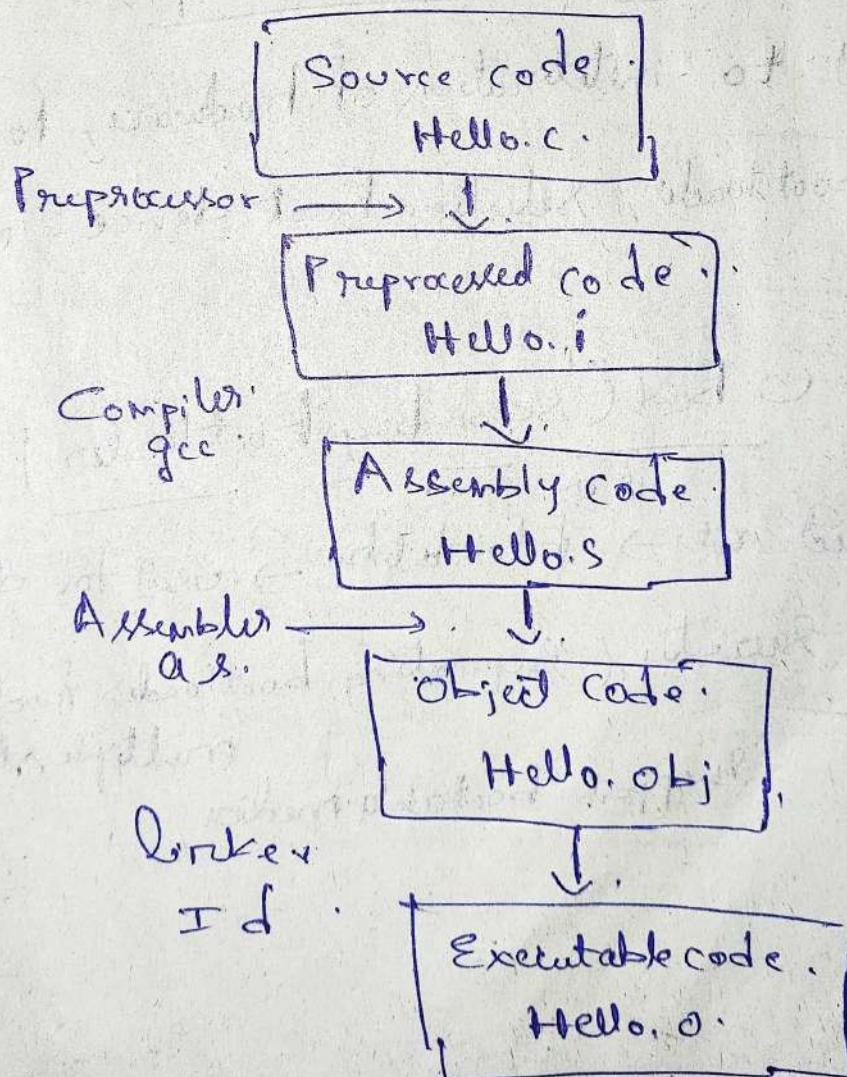
③ Tool chain

Converts Source code to executable code

Tasks → creating, building, testing &

deploying Software.

Ex:- Hello.c code



(4) ROM Code boot Sequence
which flash should done first
Selecting

(5) Bootstrap posting on Newboard

The process of adapting & configuring
a bootstrap to run on a new hardware
- board (or) platform

(6) Primary bootloader (Bootstrap)

Used to initialization of hardware, load secondary
bootloader, Selecting boot device, error handling

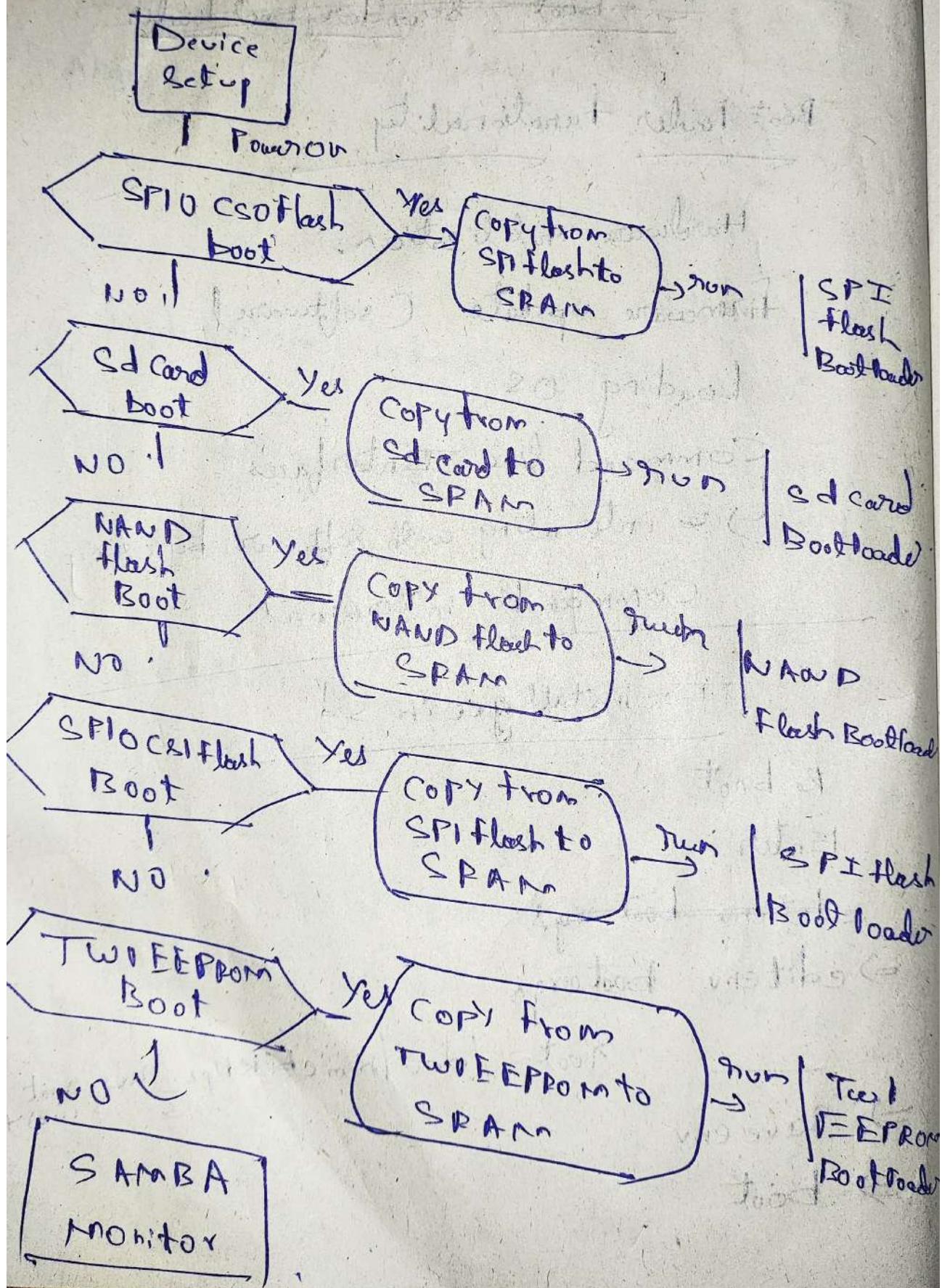
(7) U-boot (Secondary Bootloader)

Used ~~for~~ ^{it is} flexibility ^{Purposes} \rightarrow used for different
Security, (Separating bootloader functionality into
multiple stages),
Support Bootable media

12/2/24

① ROM Code Boot Sequence

→ It says which boot mode the board will initialize after ~~power on~~ Reset power on.



② why bootloader / function

firmware update → software

Hardware initialization.

Loading OS.

Command line interface → interacting by using
Commands in terminal

Primary bootloader

③ Primary bootloader (introduction)

download from git.

github.rugged board

download at 91boot strap. 3 & 12

Open in terminal.

ls

board → all specific board codes are here

config → build configurations.

driver → all drivers of interfaces which

host utilities → for binary conversions.

include → headers

scripts → build systems.

→ first file used is startup code in categories

followed by main.c, it main.c will

call your board in it

\$ cd Board

\$ dir.

\$ cd rugged-board-05d2x1

{ ls

↳ rugged-board-05d2x.c → Board file

def

2 config files → one for SD &

other for NOR

\$ cd .. / ..

\$ cd drivers

{ ls

↳ Here you will found

our rugged board

files.
↳ You will found all peripherals.

\$ del

\$ ls *.o

↳ files which are compiled
at q1-qsp1.o

→ To access NOR

sdhc.o

→ To access SD card

\$ cd ..

\$ ls

board { } → bsp folders
drivers }

When you want to build a new board, we
need to change board & drivers.
and odd things

\$ cd fs/

\$ ls

\$ cd src/

\$ ls

fat → we do isd & parted by using
we use fat only because it is ^{fat} light-weight

\$ cd .. / ..

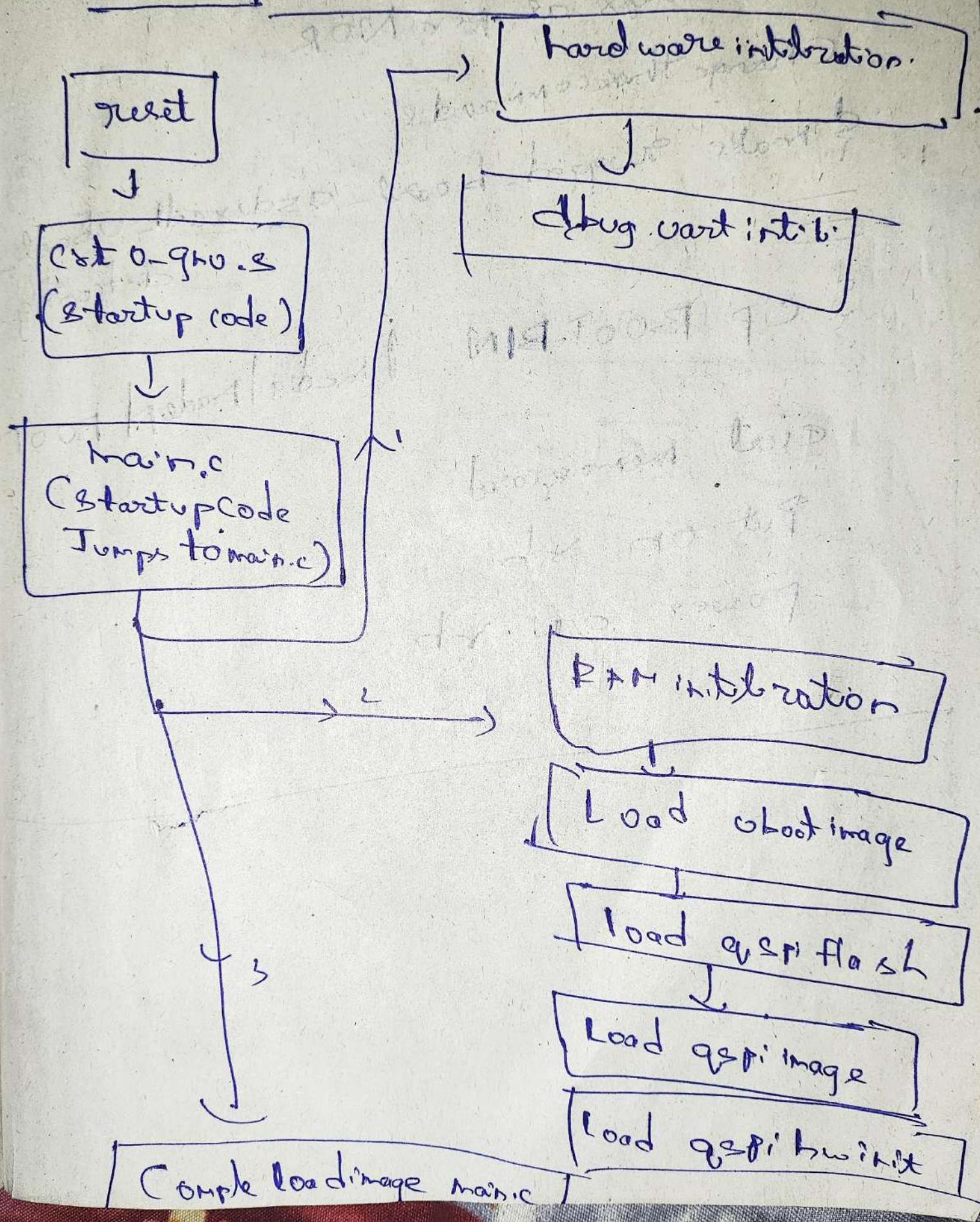
\$ ls scripts/

\$ cd host-utilities/

\$ ls

↳ Tools to generate parameters.

④ Boot strap code flow.



(5) Change banner using sd card (Boot strap)

Same steps as per NOR

Change these command - g.

\$ make swigged-boot-~~sd2xsd1~~_uboot
=====

\$ cp BOOT-BIN /media/Pradeep/BOOT

ejet memorycard

Put on vb

Power on vb

Bootloader

Ex:-

U-boot

Red Boot

Super Vivi

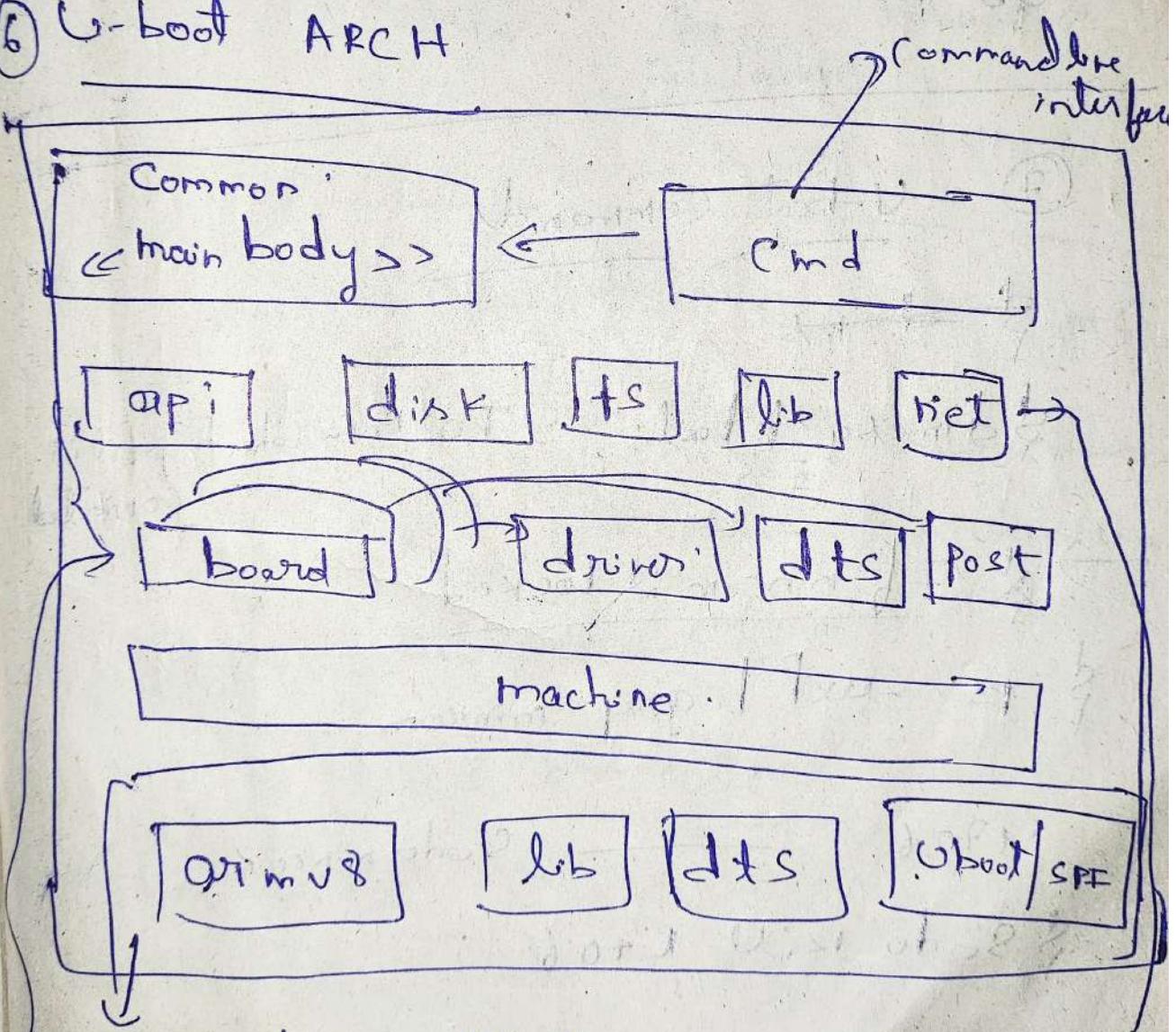
Biod

BareBox

Little Kernel

Many more

⑥ U-boot ARCH



Hardware drivers initialization
→ board
Loading OS

In Command

- first we will go to cmd (Command Line Interface) i.e Terminal
- By S.w Targ we write code,
- In Board O.S The code is dumped
Then we
- After intalling hardware drivers we will get the

⑦

U-boot Commands

~~\$ dmesg | tail.~~

\$ dmesg | tail. → To know which port is connected

→ ~~\$ if minicom is locked~~

~~\$ ps -aux | grep minicom~~

4706

Sudo minicom

\$ Sudo kill 4706

- In boot loader prompt to know all commands
 - ⇒ help
 - ⇒ bdinfo → board information

=> env → environment Variables

Change board name

⇒ Setenv bdname Hello

⇒ Printenv bdname

SF probe → Serial flash → Take default
flash

c freed

Ram
address.

Kernel address

file address.

Adding sled driver in u-boot (NXP)

download u-boot in github. nuxx board

download sled driver. from classroom

Stage - 1 Adding driver.

cp sled.c /drivers/led

Add in makefile → obj-y += sled.o

Add in k config → in pdf

cp .dtsi file
cp .dtsi /arch/arm/dts.

Stage-2 Now adding command

cp sled-test.c /uboot/cmd/

Add in makefile → obj-y += sled-test.o

Add in Kconfig → in prf

Stage 3 add menuconfig & compile
Compile & a

Enable Toolchain.

make rugged-board-asd2x-qspiflash -
defconfig
(NOR)

make rugged-board-asd2x-mmc1-defconfig (for sd)

make menuconfig

Command line interface → * sled test

device drivers → ledSupport → * 1st
* 2nd
* 3rd

make -j4

CP u-boot.bin /var/lib/tftpboot/

Stage - 4 Execution

Power on rugged board

Click Enter button

Servers ipaddr 192.168.1.10

Servers serverip 192.168.1.15.

Ping 192.168.1.15

~~st probe~~ → Uboot Commands in NOR

in developer's rugged

board.

reset

Click on Enter

Shed dsl or

Check led.

Kernel (Adding driver)

Statically

download linux-rbasdx from GitHub (rugged board)

download sled driver 2_ from Classroom

Open Terminal in Linux-rbasdx

enable toolchain.

make distclean

make clean

cp sled.c drivers/misc

Add in makefile → Obj-\$(CONFIG_SLED) += sled.o

Add in Kconfig → Fdt

Add common.dtsi file in arch/arm/boot/dts

\$ make rbsdx_defconfig

make menuconfig

No changes.

make -j4

ZImage is in /arch/arm/boot

~~now~~ → dts image in arch/arm/boot/dts
Go copy images in sd card

After open power on

it will Led on when kernel is booted

Same images for both Ep, Sd Card