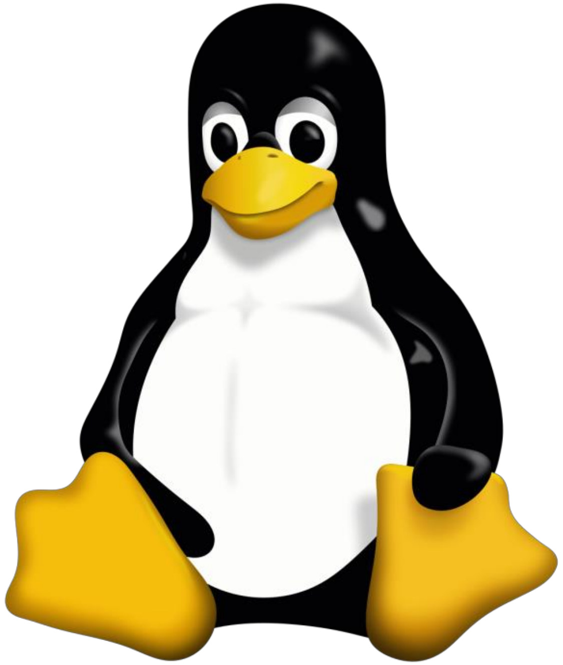
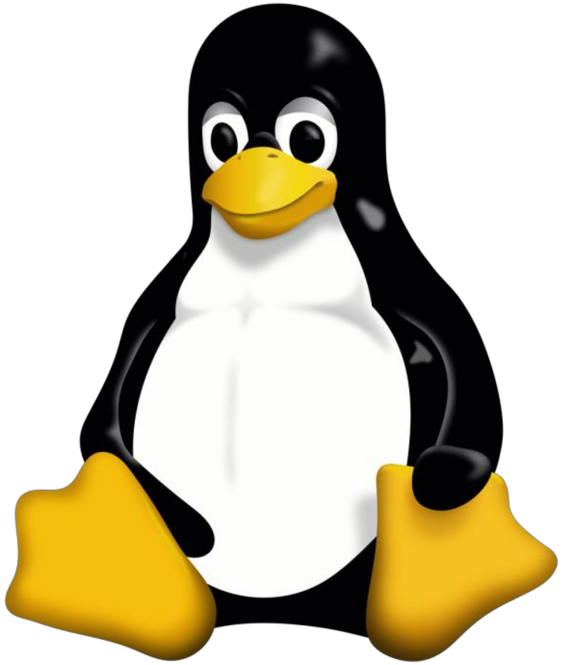


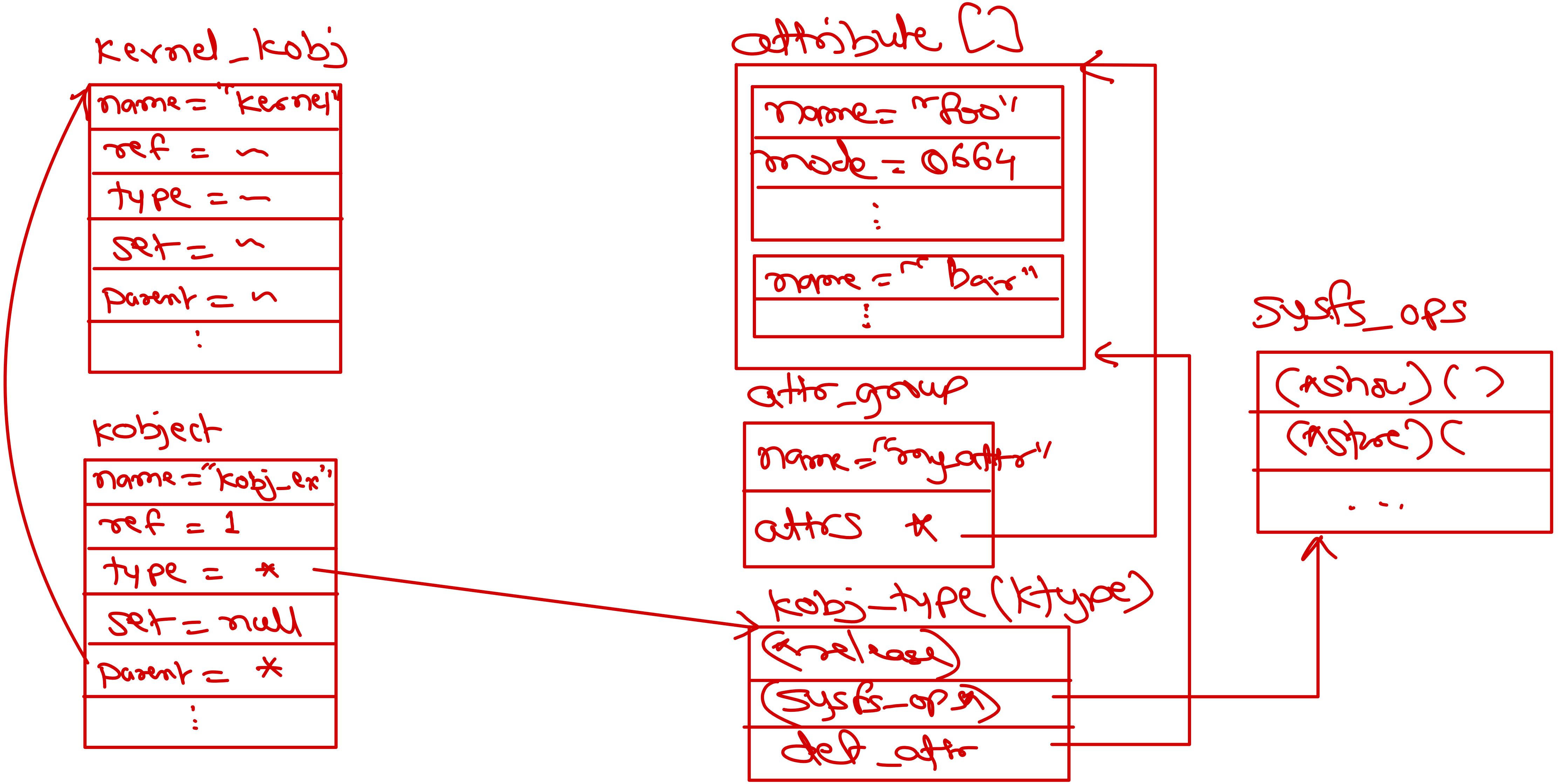
Linux Device Driver

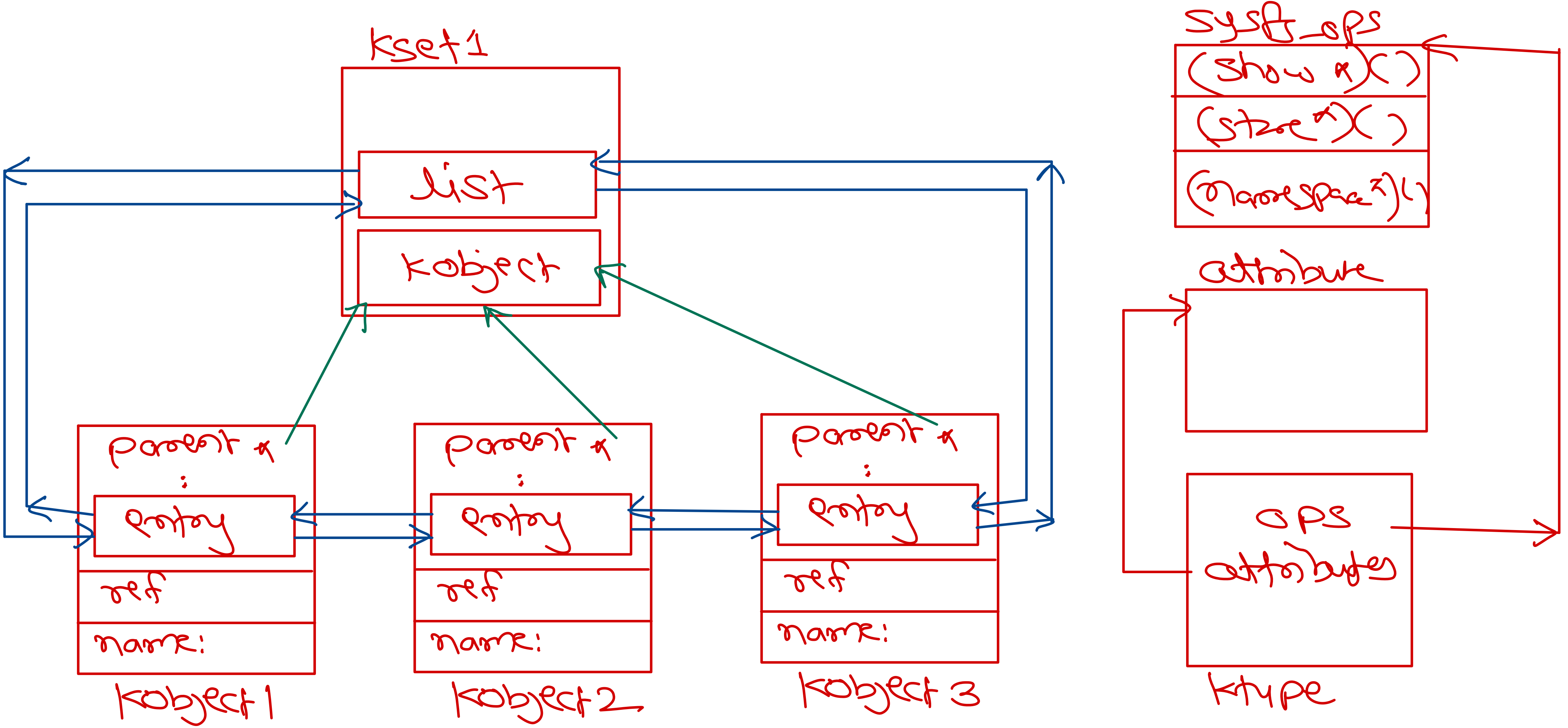
Sunbeam Infotech

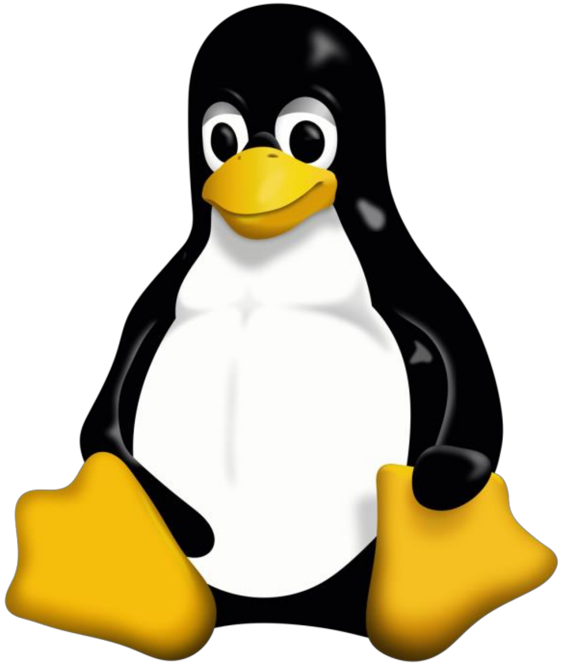


Linux Device Driver Model

Sunbeam Infotech


kernel_kobj
name = "kernel"
ref = n
type = –
set = n
parent = n
:
kobject
name = "kobj_ex"
ref = 1
type = *
set = null
parent = *
:
attribute []
name = "foo"
mode = 0664
:
name = " bar"
:
attr_group
name = "my_attr"
attrs *
kobj_type (ktype)
(*release)
(sysfs_ops)
def_attr
sysfs_ops
(*show)( )
(*store)(
. . .


kset1
list
kobject
parent *
:
entry
ref
name:
kobject1
parent *
:
entry
ref
name:
kobject2
parent *
:
entry
ref
name:
kobject3
sysfs_ops
(show *)( )
(store *)( )
(namespace *)( )
attribute
ops
attributes
ktype



Linux Platform Bus

Sunbeam Infotech

PCI bus – PC architecture

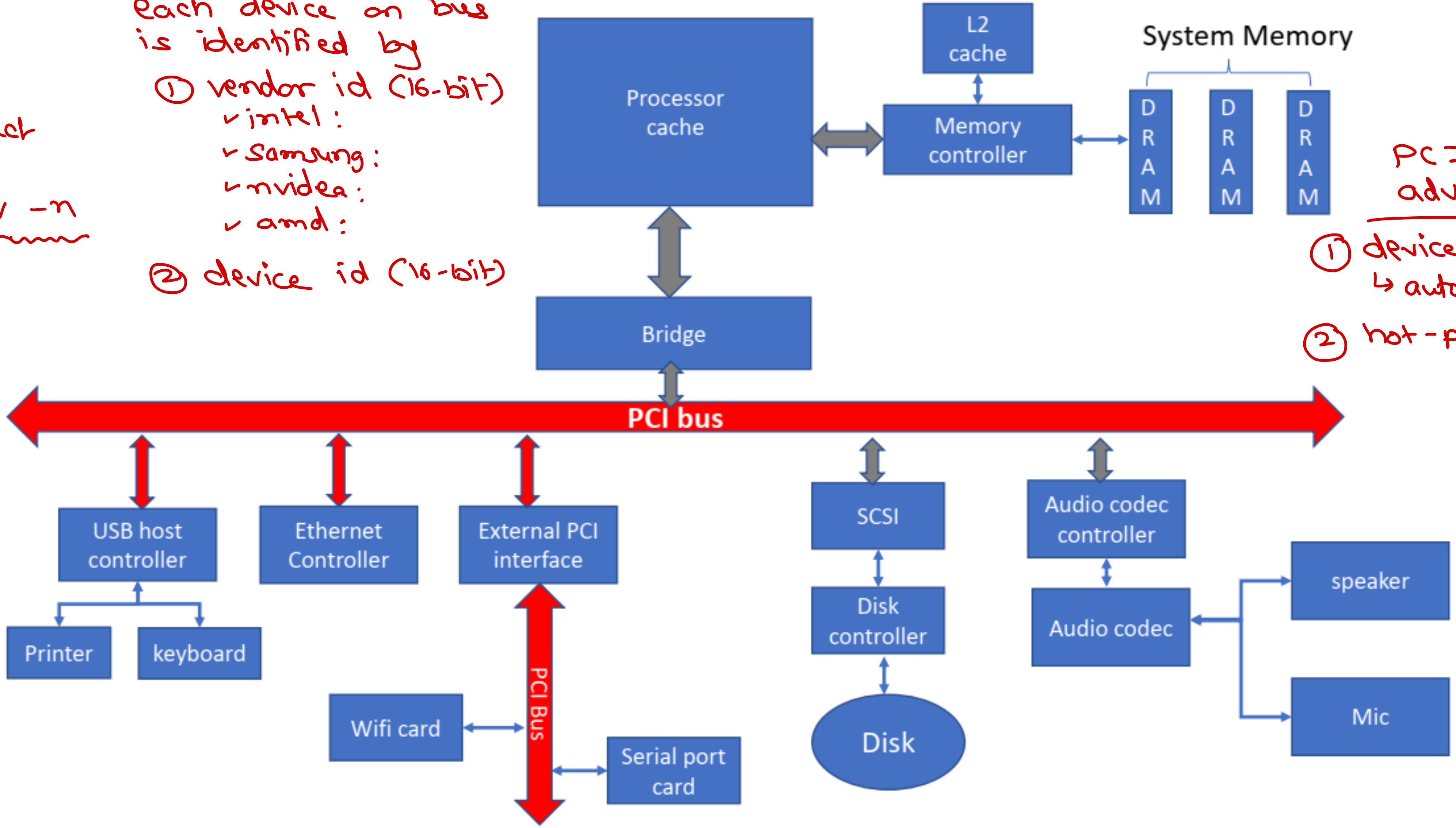
Peripheral
Component
Inter Connect

lspci -v -n

each device on bus is identified by

① vendor id (16-bit)
- ✓ intel :
- ✓ samsung :
- ✓ nvidea :
- ✓ amd :

② device id (16-bit)

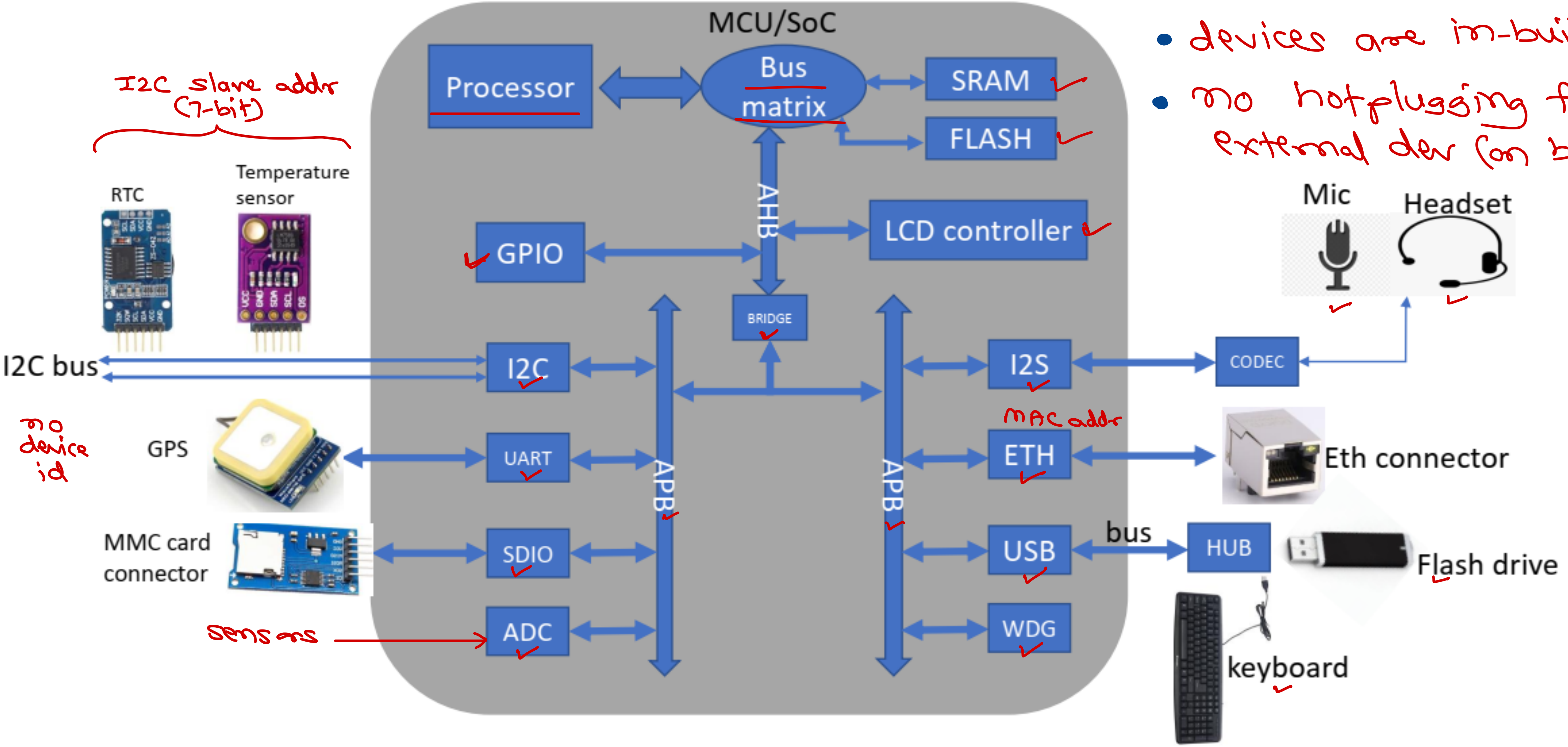


PCI bus advantages

① device identification
↳ auto-discoverable

② hot-plugging

Embedded – SoC



- devices are in-built
- no hotplugging for external dev (on bus).

struct kobject

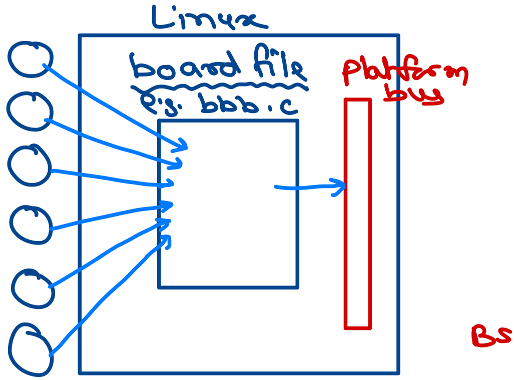
- Keeping track of various C struct objects is common need throughout the kernel.
- From Linux kernel 2.5 *struct kobject* is added for following functionalities.
- It provides following functionalities
  - Reference counting
  - Manage list of objects
  - Locking of sets
  - Exporting object properties to sysfs
- To avail these functionalities embed kobject into the desired struct.
- kobject functions: kobject_init(), kobject_get(), kobject_put(), kobject_add(), kobject_cleanup(), kobject_register(), kobject_unregister().

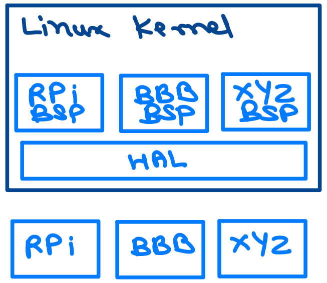
```
struct kobject {
   const char *k_name;
   struct kref kref;
   struct list_head entry;
   struct kobject *parent;
   struct kset *kset;
   struct kobj_type *ktype;
   struct sysfs_dirent *sd;
};
```

Platform bus, device and driver

- In Linux on PC architecture, most of the IO devices are connected over PCI and USB buses.
- PCI and USB buses are auto-discoverable (lspci, lsusb) and hot-pluggable (plug n play).
- Typical embedded Linux on ARM or other architecture do not have PCI bus.
- In embedded hardware (SoC) most of devices/buses are available on chip itself and are directly connected to CPU.
- Embedded buses like SPI, I2C, CAN, I2S are not discoverable/hot-pluggable.

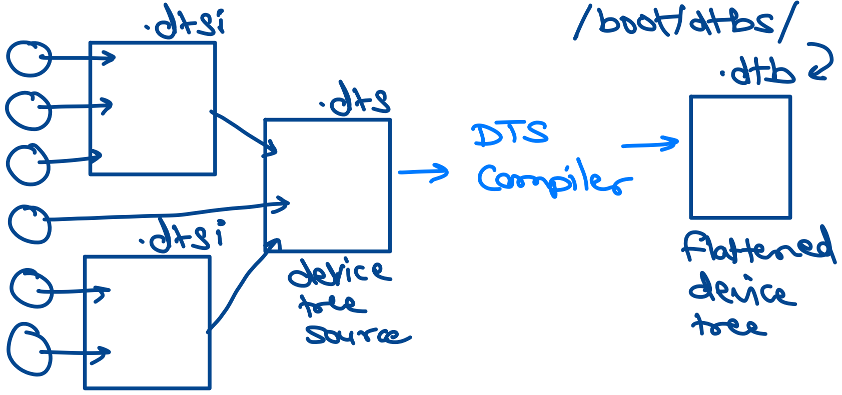
① Board File



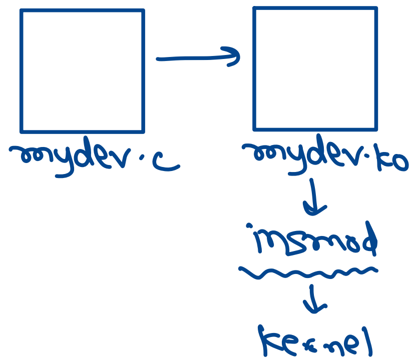


BSP = Board Specific Hw Intf / Drivers + HAL changes

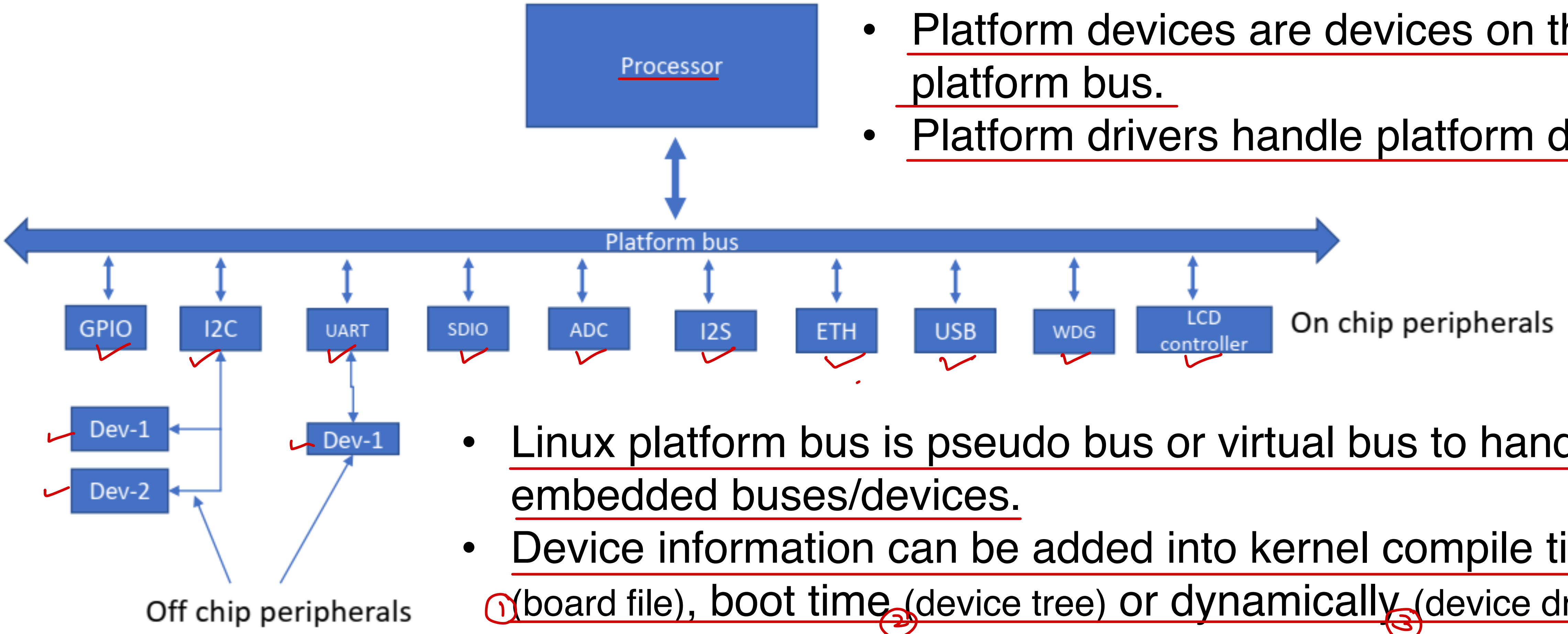
② Device tree (ksrc/arch/arm/boot/dts)



③ Device Driver



Platform bus

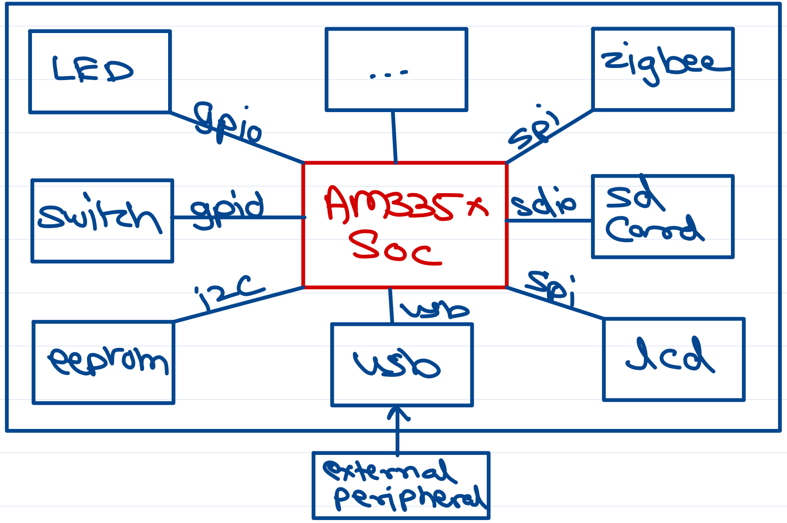


- Platform devices are devices on the platform bus.
- Platform drivers handle platform devices.

- Linux platform bus is pseudo bus or virtual bus to handle embedded buses/devices.
- Device information can be added into kernel compile time ①(board file), boot time ②(device tree) or dynamically ③(device driver).
  - Memory/IO address, IRQ number, Device Id, Device address, Pin configuration, Power/voltage information, etc.
- Device Tree: https://www.kernel.org/doc/Documentation/devicetree/usage-model.txt

Device tree

embedded board



Embedded Buses/Connectivity

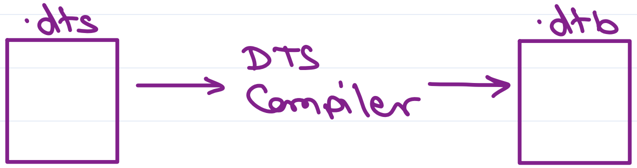
1. USB
2. RS232
3. SPI
4. I2C
5. CAN

} → platform devices

board file
↳ board specific
+ probe drivers
(recompile kernel for each board).

→ my_board_init( )
✓ add_device_serial()
✓ add_device_spi()
✓ add_device_eth()
✓ add_device_i2c()
✓ add_device_gpio()
+ device drivers (.ko)

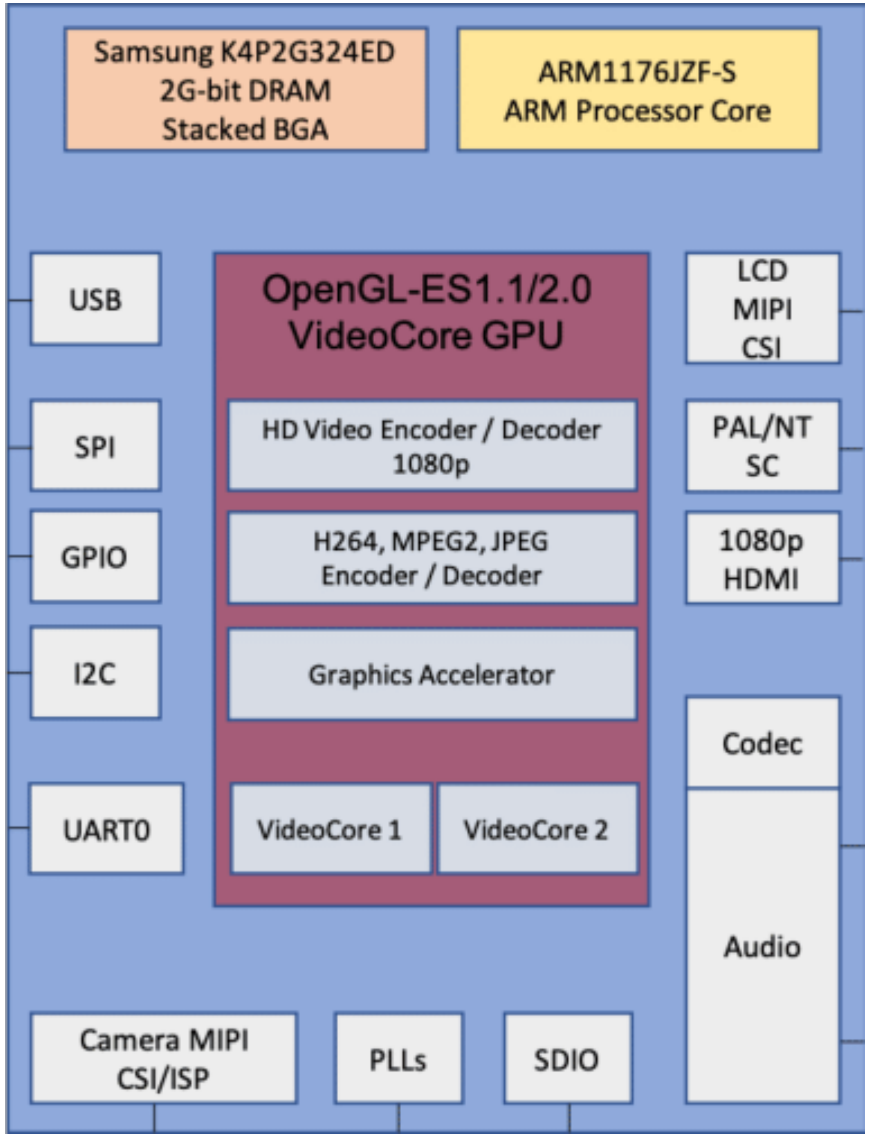
device tree source
↳ board specific
describe data struct.



Same kernel can be used to init diff boards with diff dtb files.

BCM2835 & AM335x

RPi



Samsung K4P2G324ED
2G-bit DRAM
Stacked BGA

ARM1176JZF-S
ARM Processor Core

USB

SPI

GPIO

I2C

UART0

OpenGL-ES1.1/2.0
VideoCore GPU

HD Video Encoder / Decoder
1080p

H264, MPEG2, JPEG
Encoder / Decoder

Graphics Accelerator

VideoCore 1

VideoCore 2

LCD
MIPI
CSI

PAL/NT
SC

1080p
HDMI

Codec

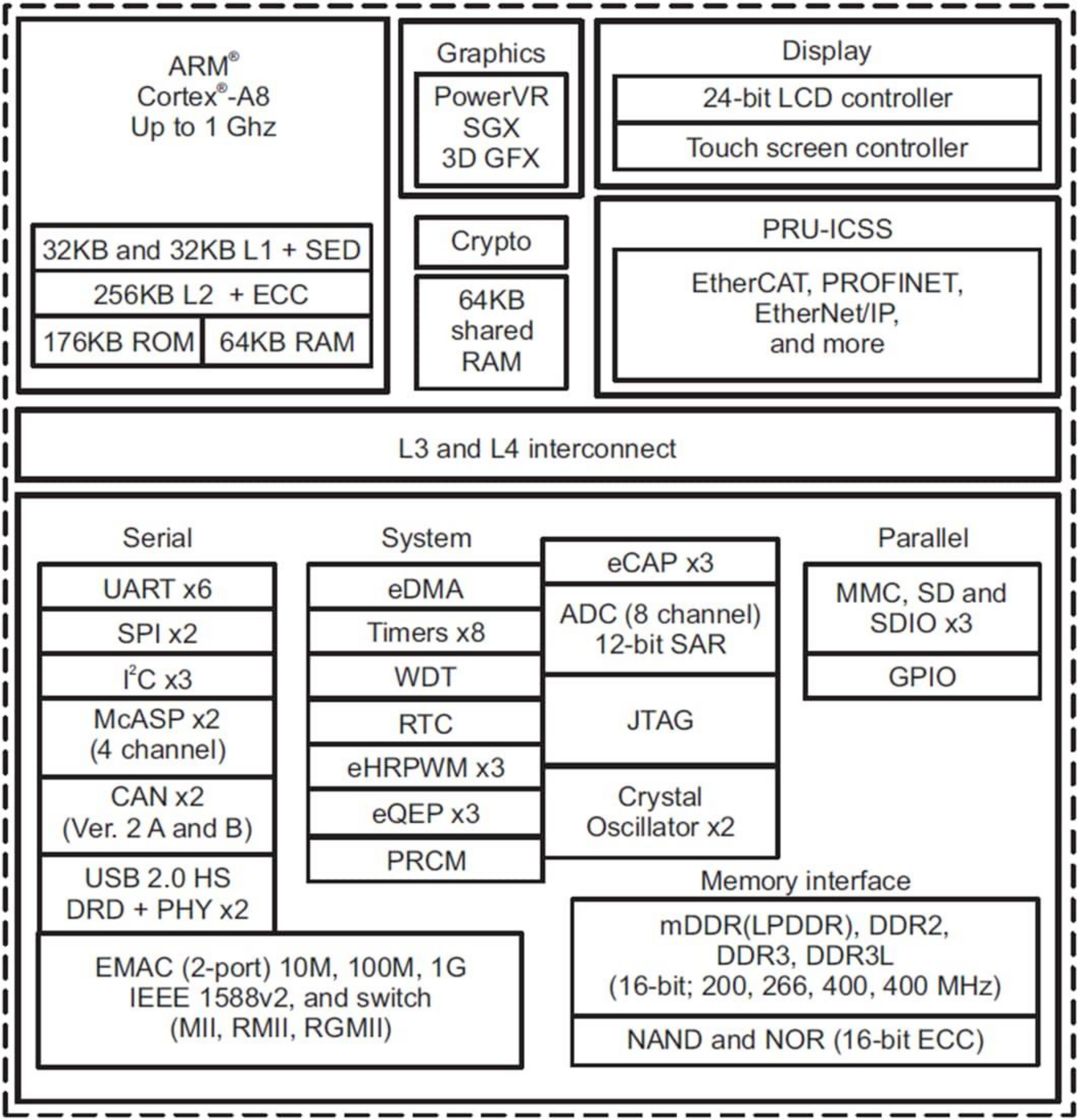
Audio

Camera MIPI
CSI/ISP

PLLs

SDIO

BBB →



ARM®
Cortex®-A8
Up to 1 Ghz

32KB and 32KB L1 + SED

256KB L2 + ECC

176KB ROM

64KB RAM

Graphics

PowerVR
SGX
3D GFX

Crypto

64KB
shared
RAM

Display

24-bit LCD controller

Touch screen controller

PRU-ICSS

EtherCAT, PROFINET,
EtherNet/IP,
and more

L3 and L4 interconnect

Serial

UART x6

SPI x2

I²C x3

McASP x2
(4 channel)

CAN x2
(Ver. 2 A and B)

USB 2.0 HS
DRD + PHY x2

System

eDMA

Timers x8

WDT

RTC

eHRPWM x3

eQEP x3

PRCM

eCAP x3

ADC (8 channel)
12-bit SAR

JTAG

Crystal
Oscillator x2

Parallel

MMC, SD and
SDIO x3

GPIO

EMAC (2-port) 10M, 100M, 1G
IEEE 1588v2, and switch
(MII, RMII, RGMII)

Memory interface

mDDR(LPDDR), DDR2,
DDR3, DDR3L
(16-bit; 200, 266, 400, 400 MHz)

NAND and NOR (16-bit ECC)

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>