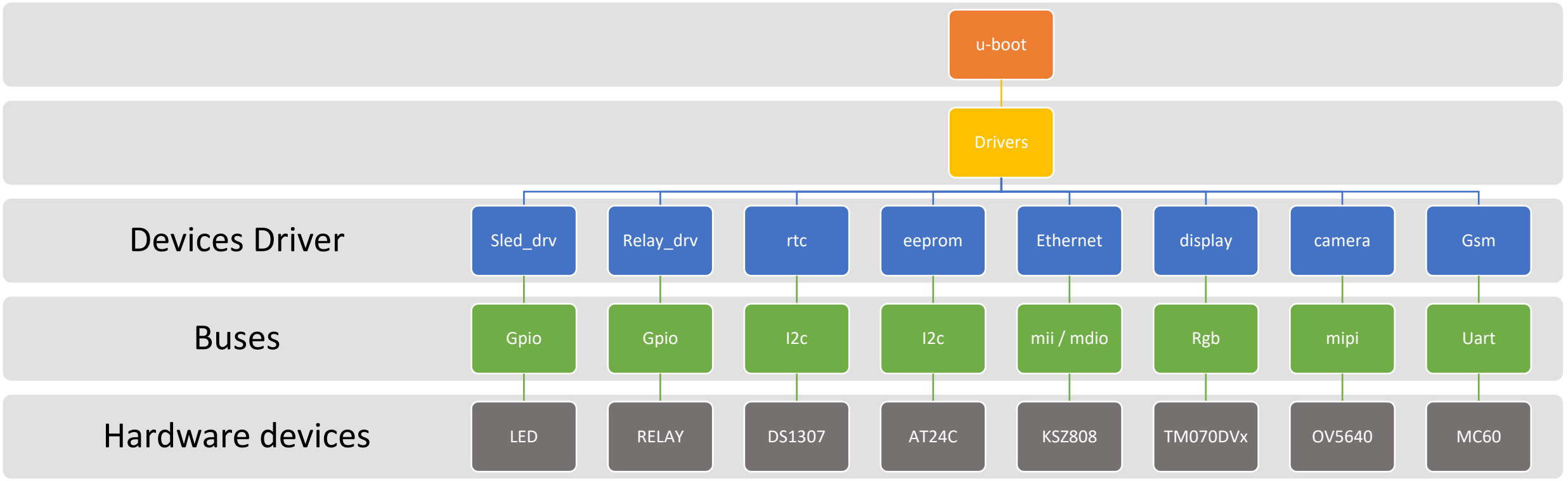
Embedded Linux Porting (C6)

Bootloader U-Boot3

Agenda

- Modifying Bootloader to add new features
- Understanding the bus drivers in U-BOOT.
- Add new driver sled in Bootloader
- Add command file sled_cmd to test the new driver
- Test the new driver on RB-A5D2x (P)

U-BOOT Bus Drivers



U-BOOT Bus Drivers

Function	Description
int gpio_**request**(unsigned gpio, const char *label);	Request use of a specific GPIO
int gpio_**free**(unsigned gpio);	Frees the GPIO that was in use
int gpio_**direction_input**(unsigned gpio);	Makes the GPIO an input
int gpio_**direction_output**(unsigned gpio, int value);	Makes the specified GPIO an output and sets its value
int gpio_**get_value**(unsigned gpio);	Gets the value of the GPIO (either input or output)
int gpio_**set_value**(unsigned gpio, int value);	Sets the value of an output GPIO (0 or 1)

Driver Code Flow

File	Description
u-boot/driver/gpio/at91_gpio.c	Atmel GPIO Driver core bus driver
u-boot/driver/gpio/gpio-uclass.c	U-Boot GPIO Subsystem HAL
u-boot/driver/led/sled.c	Sled device driver which used gpio bus driver
u-boot/command/sled_cmd.c	Test app / command implemented to test sled driver

Adding new Driver in U-Boot

#Step-1: Define your device in dts file

```
$ vim <uboot_path>/arch/arm/dts/rugged_board_a5d2x.dts

leds {
        compatible = "sled-testing";
        status = "okay";

        UserLed {
            label = "UserLed";
                        sled-default-state = "blink";
        };
    };
```

Step-2: Define your driver sled.c in uboot/driver folder

$ vim <uboot_path>/driver/led/sled.c

copy the sled.c code

Step-3: Add sled configuration in Kconfig file

$ vim <uboot_path>/driver/led/Kconfig

```
config SLED
        bool "SLED support for LEDs"
        depends on LED
        help
          Sled driver on RuggedBOARD-A5D2x
```

Step-4: Add sled configuration in Kconfig file

$ vim <uboot_path>/driver/led/Makefile

```
obj-$(CONFIG_SLED) += sled.o
```

Step-5: Write a test code cmd_sled.c under command folder and which calls the driver functions

$ vim <uboot_path>/command/cmd_sled.c

#implement do_sled() & register using U_BOOT_CMD

U-boot Compilation

Browse Source: https://github.com/rugged-board/uboot-rba5d2x

Compiling U-Boot for RuggedBOARD

```
#Set the toolchain path first
$ . env_setup.sh

# Download uboot Source
$ git clone https://github.com/rugged-board/uboot-rba5d2x.git
$ cd uboot-rba5d2x
$ git checkout origin/uboot-rba5d2x

# Configure u-boot bootloader for RB-A5D2x
$ make rugged_board_a5d2x_mmc1_defconfig          # For SD Card
Or
$ make rugged_board_a5d2x_qspiflash_defconfig     # For NOR Boot

# Compile u-boot bootloader
make
```

U-boot Flashing on RB-A5D2x (TFTP)

Power on board and stop at bootlaoder prompt

#check network connection by pining host PC

u-boot$ ping <serverip>

Download uboot image from PC to Board RAM

u-boot$ tftp 0x21FF0000 u-boot.bin

#erase serial flash(NOR) u-boot partition

u-boot$ sf erase 0x20000 0x80000

copy from uboot image from RAM to NOR Flash

u-boot$ sf write 0x21FF0000 0x20000 0x80000

U-boot Experiments

1. Modify the U-Boot Command Prompt & Test
2. Test the behaviour bootdelay env variable
3. Modify the Baudrate of the UART U-BOOT Console
4. Modify the bootcmd variable to load the kernel from TFTP Server
5. Modify the bootargs to take the RFS only from SDCard & Test
6. Modify the bootargs to take the RFS only from NFS Server
7. Add fwupdate command to U-Boot
8. Add sled driver to U-Boot
9. Add sled_cmd to test sled driver

Open Discussions

Connect-us