# Embedded Linux Porting (C5)

Bootloader U-Boot2

# Agenda

- How to port Bootloader on ARM Board
- Add new command in Bootloader
- Compiling Bootloader
- Test the new Bootloader using TFTP Boot

# U-BOOT Porting

1. Identify the ARCH, CORE & SOC used in your board
2. Check the ARCH & Core support  in u-boot location /arch/arm/cpu
3. Check the SOC support location uboot/arch/arm/mach-<soc_family>
4. Create new board folder in u-boot/boards/<board_name>
5. Take ref of existing boards in uboot and develop the code for your board

   Add board.c, modify Kconfig & Makefile
6. Create a default configuration file for your board in u-boot/configs
7. Driver level modification if required  u-boot/drivers/
8. Make sure you did modified Makfiles corresponding to your code/file changes.

# Adding new command in U-Boot

**U_BOOT_CMD()** is the Macro used to add new command in u-boot.

```
U_BOOT_CMD(name, maxargs, repeatable, command, "usage", "help")
```

| | |
|---|---|
| name: | is the name of the command. THIS IS NOT a string. |
| maxargs: | the maximum numbers of arguments this function takes |
| command: | Command implementation Function pointer (*cmd)(struct cmd_tbl_s *, int, int, char *[]); |
| Usage: | Short description. This is a string |
| help: | long description. This is a string |

**Command Function Prototype:**

```
int do_funcname  (cmd_tbl_t *cmdtp, int flag, int argc, char *const  Argv[] )
```

cmdtp – Command table pointer (function vector table)
flag -- Unused
argc -- Argument count, including command name itself
argv[] -- Array of arguments (string).

# Adding new command in U-Boot

**Step-1 : create dummy.c in u-boot/cmd folder**

$ cd *<uboot_path>*/cmd
$ vim dummy.c

```
#include<common.h>
#include<command.h>

static int do_dummy(cmd_tbl_t *cmdtp, int flag, int argc,
char * const argv[])
{
	printf("Hello Rugged Board A5d2x\n");
	printf("This is dummy command implementation\n");
	return 0;
}

U_BOOT_CMD(dummy, 2, 1, do_dummy, "testing
hello","arg1 not needed");
```

**Step-2: Modify Kconfig file under cmd folder**

$vim Kconfig

```
config CMD_DUMMY
	bool “Dummy Command"
	default y
	help
	  This is testing the new command in rugged board..
```

**Step-3: Modify Makefile**

$vim Makefile // bootloader/uboot-rba5d2x/cmd

```
obj-$(CONFIG_CMD_DUMMY) += dummy.o
```

**Step-4: Compile & Flash**

**Step-5: Test the command on Target Bootloader prompt**

=> dummy

Hello Rugged Board A5d2x
This is dummy command implementation

# 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 compiling using Yocto

*#Configure for RuggedBOARD-A5D2x*

$ source sources/poky/oe-init-build-env

$ vi conf/local.conf

# Edit  MACHINE ?= "rugged-board-a5d2x-sd1“

*#Compile*

$ bitbake u-boot

*#Images  for NOR*

$ cd tmp/deploy/images/rugged-board-a5d2x/

#Follow NOR Flashing Tutorial..

# 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

# Open Discussions

# Connect-us