i.MX 6UL/6ULL Development FAQs

Based on MYIR's i.MX6UL/6ULL Series Products

V1.0

E-mail: support@myirtech.com

Table of Contents

1.Set up the development environment 1

1.1 Requirements to the virtual machine 1

1.2 The recommended Ubuntu version 1

1.3 How to connect PC to the serial ports of the development board? 1

1.4 Naming rules of MYIR's boards and explanation of relevant DTB files 1

2. Modify and recompile the kernel, uboot, and the file system 2

2.1 How to modify Yocto after the source code of kernel or uboot is modified? 2

2.2 How to tailor "make menuconfig" of kernel? How to add new device? 3

2.3 After compiling the kernel or uboot separately we get many files, which one to use? 5

2.4 How to find "myd-y6ull-boot-mmc0-tftp.txt"? How to use? 5

3. Downloading and programming in the system 6

3.1 How to program via tftp under uboot? 6

3.2 When updating OS via SD card or programming with mfgtool, how to replace the files? 7

3.3 How to transfer files between Windows PC / Ubuntu and the development board? 7

4. Development of applications 10

4.1 How to modify if we want to use another serial port? 10

4.2 How to debug I2C? 10

4.3 How to debug SPI? 11

4.4 How to debug RS485? 12

4.5 How to debug ADC? 12

4.6 How to debug GPIO? 14

4.7 The corresponding relationship between ttymxc and uart 15

4.8 How to calculate GPIO numbers, how to use the GPIO? 15

4.9 How to enable/disable the Ethernet ports? 15

4.10 How to add other baud rate for serial port? 16

4.11 PWM control program 17

4.12 What are the LED1 & LED2 of Ethernet port for? 17

4.13 How to make MYD-6ULX support LVDS? 17

5. Protocols and tools associated with applications 19

5.1 How to add database, FTP, sqlite3 or other applications in Yocto? 19

5.2 How to modify “rc.local” in Yocto? 20

5.3 How to generate ubi file system in Yocto?.. 21

5.4 How to compile .c file using the cross-compiler toolchain?.. 21

5.5 How to modify default GPIO voltage in the source code of uboot? 21

5.6 How to modify the debug serial port to another? 22

5.7 How to set a permanent MAC address while Linux generates random MAC addresses? 22

5.8 Is there a SN in the system of every MYC-Y6ULX CPU Module? 22

5.9 How to set static IP address?.. 23

1.Set up the development environment

1.1 Requirements to the virtual machine

A: The practical using case with Windows PC is to install a virtual machine such as Oracle VM VirtualBox. The VM should have 4G memory at least, 100G hard disk approximately. Connect the VM to the internet. The internet is needed because when compiling in YOCTO, Ubuntu needs to be connected to the internet.

Yocto-downloads only contain part of the resources, other resources need to be downloaded via internet.

1.2 The recommended Ubuntu version

A: Ubuntu 16.04 64-bit release version.

1.3 How to connect PC to the serial ports of the development board?

A: Use USB-TTL serial port cable to connect PC with DEBUG serial port (JP1) on the development board. Do not use RS232 DB9 cable to connect with JP1 because the voltage of JP1 is 3.3V while the voltage from DB9 is 12V.

1.4 Naming rules of MYIR’s boards and explanation of relevant DTB files

A: MYC means CPU modules. MYB means base boards (carrier boards). MYD means development boards which are consisted of CPU modules and carrier boards.

- MYC-Y6ULG2-256N256D-50-I is based on 528MHz NXP i.MX6UL G2 sub family processors with 256MB DDR3, 256MB Nand Flash of industrial working temp.

- MYC-Y6ULY2-256N256D-50-I is based on 528MHz NXP i.MX6ULL Y2 sub family processors with 256MB DDR3, 256MB Nand Flash of industrial working temp.

Explanation of relevant DTB files:

- If you are using development boards based on i.MX6UL, modify myd-y6ul-gpmi-weim.dtb; myb-y6ul-14x14.dts
- If you are using development boards based on i.MX6ULL, modify myd-y6ull-gpmi-weim.dtb; myb-y6ull-14x14.dts

2. Modify and recompile the kernel, uboot, and the file system

2.1 How to modify Yocto after the source code of kernel or uboot is modified?

A: When we recompile in Yocto after the source code of kernel or uboot is modified, we need to modify commit ID of the kernel or uboot.

- How to get the commit ID of the kernel or uboot?

  In the directory of the kernel or uboot:

```
git add . (commit all the modification)
git config - -gobal user.email “your Email address” (commit the email address of the modifier)
git config - - global user.name “your name” (commit the name of the modifier)
git commit -m “comment”(add comment)
git log (get the commit ID)
```

- If the kernel is modified, we need to modify the “SRCREV” under below address:

```
/home/roy/MYD-Y6ULX-devel/04-Source/fsl-release-Yocto/sources/meta-myir-imx6ulx/recipes-kernel/linux/linux-mys6ulx_4.1.15.bb
```

Example code:

```
# Copyright (C) 2013-2016 Freescale Semiconductor
# Released under the MIT license (see COPYING.MIT for the terms)
SUMMARY = "Linux Kernel for MYiR MYS6ULx board"
DESCRIPTION = "Linux Kernel provided and supported by Freescale with focus on \
i.MX Family Reference Boards. It includes support for many IPs such as GPU, VPU and IPU."
require recipes-kernel/linux/linux-imx.inc
require recipes-kernel/linux/linux-dtb.inc
DEPENDS += "lzop-native bc-native"

LOCALVERSION = "-1.2.0"
SRCREV = "d87b5be6bfc5a78cd45d8efa044fddcd7f4b2ac1"
SRCBRANCH = "mys-6ulx"
SRC_URI = "git:///${HOME}/MYiR-iMX-Linux;protocol=file;branch=${SRCBRANCH} \
                  file://defconfig \
             "
DEFAULT_PREFERENCE = "1"
COMPATIBLE_MACHINE = "(mx6ull|mx6ul)"
```

- In Yocto, the directory to modify the commit ID of uboot:

```
/home/roy/MYD-Y6ULX-devel/04-Source/fsl-release-Yocto/sources/meta-myir-imx6ulx/recipes-bsp/u-boot/u-boot-mys6ulx_2016.03.bb
```

2.2 How to tailor “make menuconfig“ of kernel? How to add new device?

A: Below is how to configure the kernel and how to add new device.

- 3 ways to configure the kernel:

1. make config

2. make menuconfig

3. make xconfig (QT is needed)

Either of above 3 ways works, “make menuconfig” is recommended.

Input this command in the terminal for VM to open the kernel configuration interface: make menuconfig.

- How to operate the kernel configuration interface：

1. Pressing y to select.

2. Pressing n not to select.

3. Pressing m to make it a module.

4. Press Esc to back to upper level page.

5. Press the arrow keys to select.

6. [*] means already selected.

7. [ ] means not selected yet.

Generally, we tailor the kernel according to actual demand. We introduce some necessary items here.

“General setup-->System V IPC (IPC: Inter Process Communication)” is necessary.

After all the necessary items have been configured, press Esc to quit, choose Yes to save.

After setting up the cross-compiler, input the command in the terminal: make. Then wait for a long period of time. After the compilation is completed, “zImage” would be generated under directory “. /arch/arm/boot/”.

- Add new device in the kernel (take LED driver for example)

Create a LED directory under “kernel\drivers\char\”, put the LED driver code in this directory.

Modify the “Makefile” in “kernel\drivers\char\” to include the LED directory: add “obj-y += led/” in the “Makefile”.

- obj-y means get “foo.o” from compiling “xx.c” or “foo.s” file and connect “foo.o” to the kernel.

- obj-m means to compile this file as a module.

Object files except “y” and “m” won’t be compiled.

Add “Makefile” file and “Kconfig” file to LED directory, then add below contents to these 2 files:

“Makefile” file:

```
obj-$(CONFIG_MY_LED_DRIVER) += my-led.o
```

“Kconfig” file：

```
config MY_LED_DRIVER
bool "my led driver"
default y
help
compile for leddriver, y for kernel, m for module.
```

The needed knowledge in practice is far more than we have introduced here, to read relevant books or contents online is suggested.

2.3 After compiling the kernel or uboot separately we get many files, which one to use?

A: “zImage” file under directory “~/MYiR-imx-Linux/arch/arm/boot/” is from compiling kernel.

“u-boot.imx” file under directory “~/MYiR-iMX-uboot/” is from compiling uboot.

2.4 How to find “myd-y6ull-boot-mmc0-tftp.txt”? How to use?

A: When we boot Linux from SD card using u-boot, the system would check the file “boot.scr”. When creating SD card booting file, if nand flash is used on the board, “boot.scr” is not needed. If eMMC flash is used on the board, “boot.scr” is needed.

Generally, the content of “boot.scr” won’t be modified. The content of “boot.scr” is not included in the CDs which MYIR delivers with MYD-Y6UL & 6ULL development boards, you may edit it by yourself as below.

```
setenv mmcroot '/dev/mmcblk0p2 rootwait rw rootdelay=5 mem=256M'
run mmcargs
tftpboot 0x83000000 zImage
tftpboot 0x84000000 myd-y6ull-gpmi-weim.dtb
bootz 0x83000000 - 0x84000000
```

3. Downloading and programming in the system

3.1 How to program via tftp under uboot?

A: We may use tftp to download when programming from the internet. Example code:

Configure the IP under uboot:

```
setenv ipaddr 192.168.30.106      (example only)
setenv serverip 192.168.30.103      (example only)
setenv ethaddr 00:01:03:A0:03:11    (example only)
saveenv
```

kernel:

```
        tftp ${loadaddr} zImage-myd-y6ull
        nand erase 0x600000 0xA00000            //to erase
        nand write ${loadaddr} 0x600000 0xA00000      //to write
```

dtb:

```
        tftp ${fdt_addr} zImage-myd-y6ull-14x14-gpmi-weim.dtb
        nand erase 0x1000000 0x100000
        nand write ${fdt_addr} 0x1000000 0x100000
```

rootfs:

```
        tftp 0x85000000 rootfs.ubi      (ubi file system)
        nand erase 0x1100000 0x9000000
```

```
            nand write.e 0x85000000 0x1100000 0x9000000
```

Please note above address bits are examples only, you may modify according to the partition of your PC.

3.2 When updating OS via SD card or programming with mfgtool, how to replace the files?

A: When making the files in the SD card to update OS, please choose relevant "mfgimages-myd*" folder. There is a file named "Manifest" in each folder. In the file "Manifest" we can find the naming rules of uboot, kernel, dtb and file system. When moving new files to "mfgimages-myd*" folder, naming rules from "Manifest" must be obeyed.

When programming with mfgtool in Windows, the new uboot, kernel, dtb, file system files should be move to "MYD-Y6ULX-mfgtools-20180810\Profiles\Linux\OS Firmware\files".

3.3 How to transfer files between Windows PC / Ubuntu and the development board?

- **How to connect Ubuntu to the Internet?**

Run Oracle VM VirtualBox (this is what I am using, you may use other VirtualBox). Open the settings in the supervisor, choose network, enable network connection, connection type: bridge network card, the name of the interface: Select by actual occurrence.

If you want ubuntu to use fixed IP, you may set it up in "Ubuntu/etc/network/interface".

Reference:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
auto enp0s3
iface enp0s3 inet static
address 192.168.30.109
netmask 255.255.255.0
getway 192.168.30.1
```

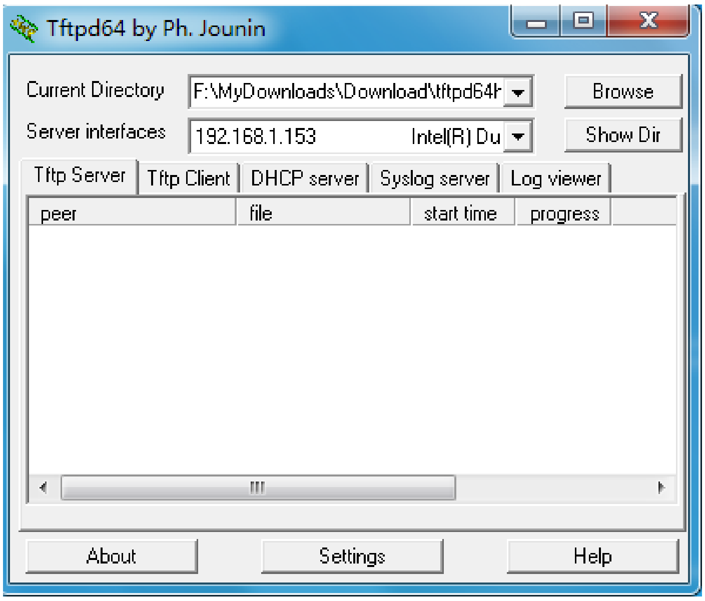
- **How to transfer files between Windows PC and the development board?**

The board and PC should be connected to the same network.

Install tftpd64.exe software on the PC.

Current Directory, choose the path to upload files.

Server interface, choose the IP of PC. Execute “tftp 192.168.1.153 -g -r test.sh” in Xshell when downloading files to the board. Execute “tftp 192.168.1.153 -p -r test.sh” when upload files from the board to PC.



Tftpd64 by Ph. Jounin

Current Directory F:\MyDownloads\Download\tftpd64r

Server interfaces 192.168.1.153 Intel(R) Du

3-3-1 tftpd64 Server configuration

- **How to transfer files between Ubuntu and the development board?**

The board and Ubuntu should be connected to the same network.

Execute scp file to transfer files:

“scp -r /home/roy/rs485 root@192.168.1.223:/home/root” .

Tips: this command means copy folder ubuntu/home/roy/rs485 to directory /home/root of board with IP 192.168.1.223.

4. Development of applications

4.1 How to modify if we want to use another serial port?

A: The driver is ready on the MYD-Y6UL/6ULL development board. We only need to modify dts file. The specific pin depends on practical case.

- Open the kernel source code file “/arch/arm/boot/dts/myb-y6ull-14x14.dts”

UART: take the example of adding UART3. Please note to delete the 2 lines about UART3 in original dts file “pincrl_uart2” because there is only 1 usage mode for 1 pin.

- In “myb-y6ull-14x14.dts” , add UART3 referring to existing UART.

```
 pinctrl_uart3: uart3grp {               fsl,pins = <
                      MX6UL_PAD_UART3_TX_DATA__UART3_DCE_TX    0x1b0b1
                      MX6UL_PAD_UART3_RX_DATA__UART3_DCE_RX    0x1b0b1
              >;
        };
........
&uart3 { pinctrl-names = "default";
 pinctrl-0 = <&pinctrl_uart3>;
 status = "okay";
};
```

4.2 How to debug I2C?

A: Determine which pin to use according to the hardware design. Open the kernel source code file “/arch/arm/boot/dts/myb-y6ull-14x14.dts” .

The codes for I2C1 and I2C2 are provided in “myb-y6ull-14x14.dts” . So here we take the example of I2C3. Disable fec2 because I2C3 uses fec2.

```
&i2c3 {              clock-frequency = <100000>;
                pinctrl-names = "default";
                pinctrl-0 = <&pinctrl_i2c3>;
                status = "okay";
```

```
    };
    ……….
            pinctrl_i2c3: i2c3grp {                                    fsl,pins = <
MX6UL_PAD_ENET2_RX_DATA0__I2C3_SCL 0x4001b8b0
MX6UL_PAD_ENET2_RX_DATA1__I2C3_SDA 0x4001b8b0
    }
```

4.3 How to debug SPI?

A: Determine which pin to use according to the hardware design. Open the kernel source code file "/arch/arm/boot/dts/myb-y6ull-14x14.dts".

Modify dts file and the following example codes. Which SPI to use and pin configuration depends on practical demand.

Enable spi_dev in "make menuconfig".

Directory: SPI SUPPORT/User mode SPI device driver support

```
                pinctrl_ecspi1: ecspi1grp {
                fsl,pins = <
                        MX6UL_PAD_CSI_DATA07__ECSPI1_MISO 0x100b1
                        MX6UL_PAD_CSI_DATA06__ECSPI1_MOSI 0x100b1
                        MX6UL_PAD_CSI_DATA04__ECSPI1_SCLK 0x100b1
                >;
        };

        pinctrl_ecspi1_cs: ecspi1cs {
                fsl,pins = <
                        MX6UL_PAD_CSI_DATA05__GPIO4_IO26    0x80000000
                >;
        };
    ……..
    &ecspi1 {
     compatible = "fsl,imx6ul-ecspi";
     fsl,spi-num-chipselects = <1>;
     cs-gpios = <&gpio4 26 0>;
     pinctrl-names = "default";
     pinctrl-0 = <&pinctrl_ecspi1 &pinctrl_ecspi1_cs>;
     status = "okay";

     spidev@0x00{
        #address-cellss=<1>;
        #size-cells=<1>;
```

```
            compatible = "spidev";
            spi-max-frequency = <8000000>;
            reg = <0>;
      };
     };
```

4.4 How to debug RS485?

A: Determine which pin to use according to the hardware design. Open the kernel source code file "/arch/arm/boot/dts/myb-y6ull-14x14.dts".

Example code:

```
            pinctrl_uart3: uart3grp {
                  fsl,pins = <
                        MX6UL_PAD_UART3_RX_DATA__UART3_DCE_RX     0x1b0b1
                        MX6UL_PAD_UART3_TX_DATA__UART3_DCE_TX     0x1b0b1
                  /*    MX6UL_PAD_UART1_CTS_B__GPIO1_IO18         0x1b0b1 RS485 RE/DE */
                  >;
            };
      ……………………
      &uart3 {
       pinctrl-names = "default";
       pinctrl-0 = <&pinctrl_uart3>;
       fsl, rs485-gpio-txen = <&gpio1 18 GPIO_ACTIVE_HIGH>;
       linux,rs485-enable-at-boot-time;
       status = "okay";
      };
```

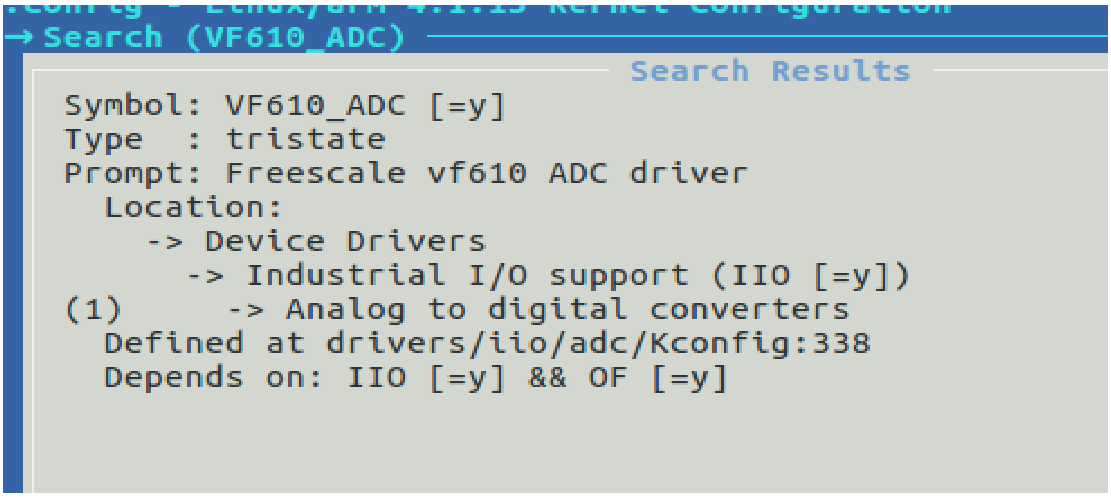
4.5 How to debug ADC?

A: Determine which pin to use according to the hardware design. Open the kernel source code file "/arch/arm/boot/dts/myb-y6ull-14x14.dts".

```
       regulators {
            compatible = "simple-bus";
            #address-cells = <1>;
            #size-cells = <0>;
            reg_can_3v3: regulator@0 {
                  compatible = "regulator-fixed";
                  reg = <0>;
                  regulator-name = "can-3v3";
                  regulator-min-microvolt = <3300000>;
```

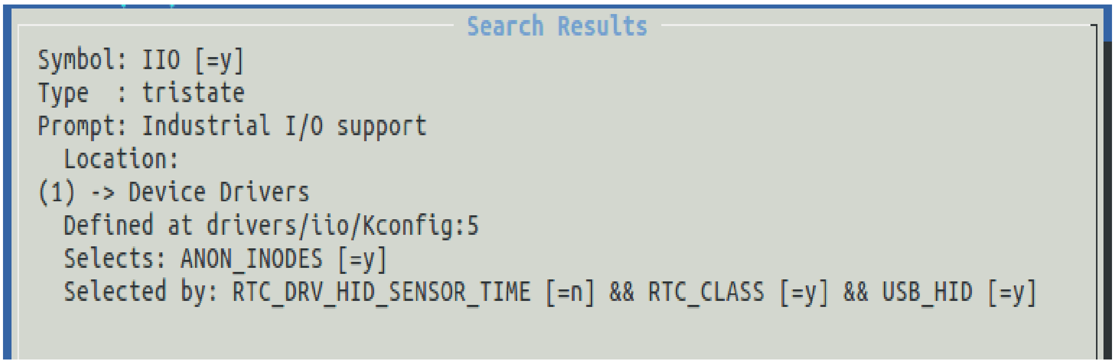
```
                regulator-max-microvolt = <3300000>;
        };
                reg_vref_3v3: regulator@3 {
                        compatible = "regulator-fixed";
                        regulator-name = "vref-3v3";
                        regulator-min-microvolt = <3300000>;
                        regulator-max-microvolt = <3300000>;
                }
}

pinctrl_adc1: adc1grp {
                fsl,pins = <
                        MX6UL_PAD_GPIO1_IO01__GPIO1_IO01            0xb0
                >;
        };
.....
&adc1 {
 pinctrl-names = "default";
 pinctrl-0 = <&pinctrl_adc1>;
 num-channels = <1>;
 vref-supply = <&reg_vref_3v3>;
 status = "okay";
};
```

Enable iio and vf610_adc through “make menuconfig”, then compile to generate new kernel and dtb file.



```
.config - Linux/arm 4.1.15 Kernel Configuration
→ Search (VF610_ADC)
                              Search Results
 Symbol: VF610_ADC [=y]
 Type  : tristate
 Prompt: Freescale vf610 ADC driver
   Location:
     -> Device Drivers
       -> Industrial I/O support (IIO [=y])
 (1)     -> Analog to digital converters
   Defined at drivers/iio/adc/Kconfig:338
   Depends on: IIO [=y] && OF [=y]
```

4-5-1 Enable VF610_ADC in kernel



```
Search Results
Symbol: IIO [=y]
Type  : tristate
Prompt: Industrial I/O support
  Location:
(1) -> Device Drivers
  Defined at drivers/iio/Kconfig:5
  Selects: ANON_INODES [=y]
  Selected by: RTC_DRV_HID_SENSOR_TIME [=n] && RTC_CLASS [=y] && USB_HID [=y]
```

4-5-2 Enable IIO in kernel

Then read value and set parameters from directories including

“/sys/bus/iio/devices/iio\:device0/”

4.6 How to debug GPIO?

A: Determine which pin to use according to the hardware design. Open the kernel source code file “/arch/arm/boot/dts/myb-y6ull-14x14.dts” .

GPIO: set LCD_DATA0 as GPIO.

```
    &iomuxc {
     pinctrl-names = "default";
     pinctrl-0 = <&pinctrl_hog_1>;
     imx6ul-evk {
          pinctrl_hog_1: hoggrp-1 {
               fsl,pins = <
                    MX6UL_PAD_UART1_RTS_B__GPIO1_IO19 0x17059 /* SD1 CD */
                    MX6UL_PAD_JTAG_MOD__GPIO1_IO10         0x17059 /* WiFi module power */
                    MX6UL_PAD_NAND_CE1_B__GPIO4_IO14  0x17059 /* LTE Reset */
                    MX6UL_PAD_GPIO1_IO00__ANATOP_OTG1_ID       0x17059 /* USB OTG1 ID */
                    MX6UL_PAD_GPIO1_IO09__GPIO1_IO09    0x1b0b0 /* LCD_DISP */
                    MX6UL_PAD_GPIO1_IO02__GPIO1_IO02              0x10b1
                    MX6UL_PAD_LCD_DATA00__GPIO3_IO05          0x1b0b0     (Set LCD_DATA0 as
GPIO)
    ……
    &lcdif {
     pinctrl-names = "default";
     pinctrl-0 = <&pinctrl_lcdif_dat_16bits
                    &pinctrl_lcdif_ctrl
                    &pinctrl_lcdif_reset>;
     display = <&display0>;
     status = "disabled";     (Disable previous usage of LCD_DATA0)
```

After above modification to dts file, load the tool chain to compile.

4.7 The corresponding relationship between ttymxc and uart

A: UART1 corresponds to ttymxc0

UART2 corresponds to ttymxc1

UART3 corresponds to ttymxc2

…

4.8 How to calculate GPIO numbers, how to use the GPIO?

A: Make sure the GPIO in dts is not used before using the GPIO. Define the GPIO in dts, generate new dtb, program it to the board, then the board is able to input and output via the GPIO.

Calculation formula for GPIO number: (n-1)*32 +m

Examples:

LCD_DATA14 is gpio3.io19.

(M-1)*32+n = (3-1)*32+19=83

Set GPIO to output, set high/low voltage of GPIO:

```
echo 83 > /sys/class/gpio/export                (Set the GPIO number)
echo out > /sys/class/gpio/gpio83/direction      (Set GPIO to output)
cat /sys/class/gpio/gpio10/value                 (Check the high/low voltage of GPIO)
echo 0 > /sys/class/gpio/gpio10/value            (Set high/low voltage of GPIO)
```

4.9 How to enable/disable the Ethernet ports?

A: The eth0 is brought out from MYC-Y6ULX CPU module directly. If only one Ethernet port is needed, we may disable other Ethernet ports in dts. Meanwhile, you need to move the configuration for mdio in dts to the ethernet port which you want to use.

Two RJ45 connectors on MYD-6ULX board: CN2 is eth0, CN1 is eth1.

Example code for modifying one Ethernet port:

```
&fec1 {
 pinctrl-names = "default";
 pinctrl-0 = <&pinctrl_enet1>;
 phy-mode = "rmii";
 phy-handle = <&ethphy0>;
 phy-reset-gpios = <&gpio5 9 GPIO_ACTIVE_LOW>;
 phy-reset-duration = <26>;
 status = "okay";
        mdio {
        #address-cells = <1>;
        #size-cells = <0>;
        ethphy0: ethernet-phy@0 {
               compatible = "ethernet-phy-ieee802.3-c22";
               smsc,disable-energy-detect;
               reg = <0>;
        };
};
```

4.10 How to add other baud rate for serial port?

A: Add kernel source code in “/driver/tty/serial/serial_core.c”. Example code:

```
...
static const struct baud_rates baud_rates[] = {
 { 921600, B921600 },
 { 460800, B460800 },
 { 230400, B230400 },
 { 115200, B115200 },
 {   57600, B57600   },
 {   38400, B38400   },
 {   19200, B19200   },
 {     9600, B9600     },
 {     4800, B4800     },
 {     2400, B2400     },
 {     1200, B1200     },
 {           0, B38400   }
...
```

4.11 PWM control program

A: We only provide the output configuration of PWM below, this is not enough, you need to add PWM node according to the pin you used by modifying dts.

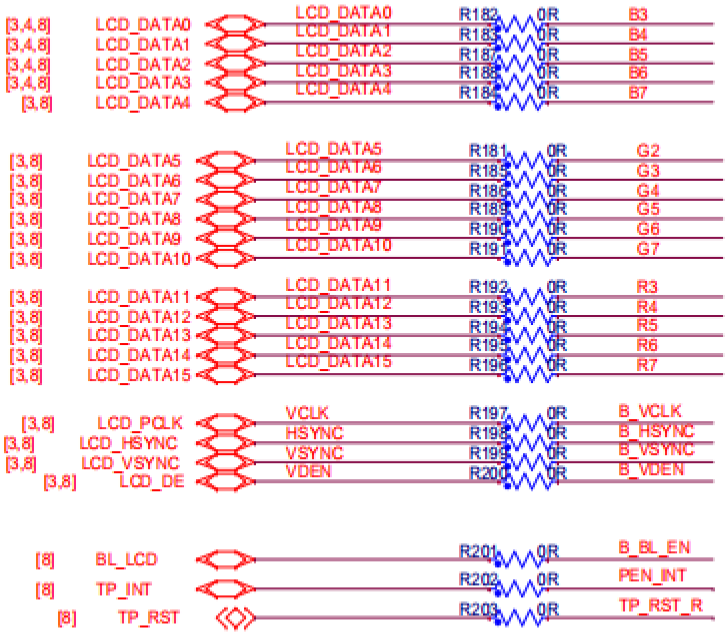
```
echo 100000 > /sys/class/pwm/pwmchip0/pwm0/period
echo 50000 > /sys/class/pwm/pwmchip0/pwm0/duty_cycle
echo 0 > /sys/class/pwm/pwmchip1/export
echo 100000 > /sys/class/pwm/pwmchip1/pwm0/period
echo 50000 > /sys/class/pwm/pwmchip1/pwm0/duty_cycle
echo 0 > /sys/class/pwm/pwmchip2/export
echo 100000 > /sys/class/pwm/pwmchip2/pwm0/period
echo 50000 > /sys/class/pwm/pwmchip2/pwm0/duty_cycle
echo 0 > /sys/class/pwm/pwmchip3/export
echo 100000 > /sys/class/pwm/pwmchip3/pwm0/period
echo 50000 > /sys/class/pwm/pwmchip3/pwm0/duty_cycle
```

4.12 What are the LED1 & LED2 of Ethernet port for?

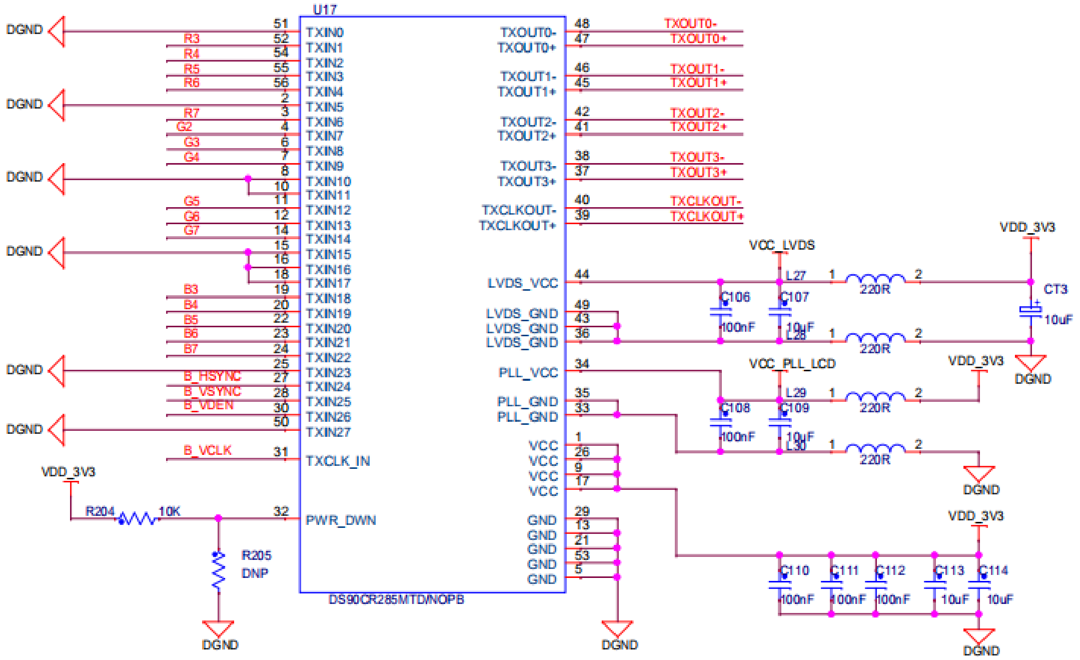
A: LED1: data transmission indicator (green). LED2: connection indicator (orange).

4.13 How to make MYD-6ULX support LVDS?

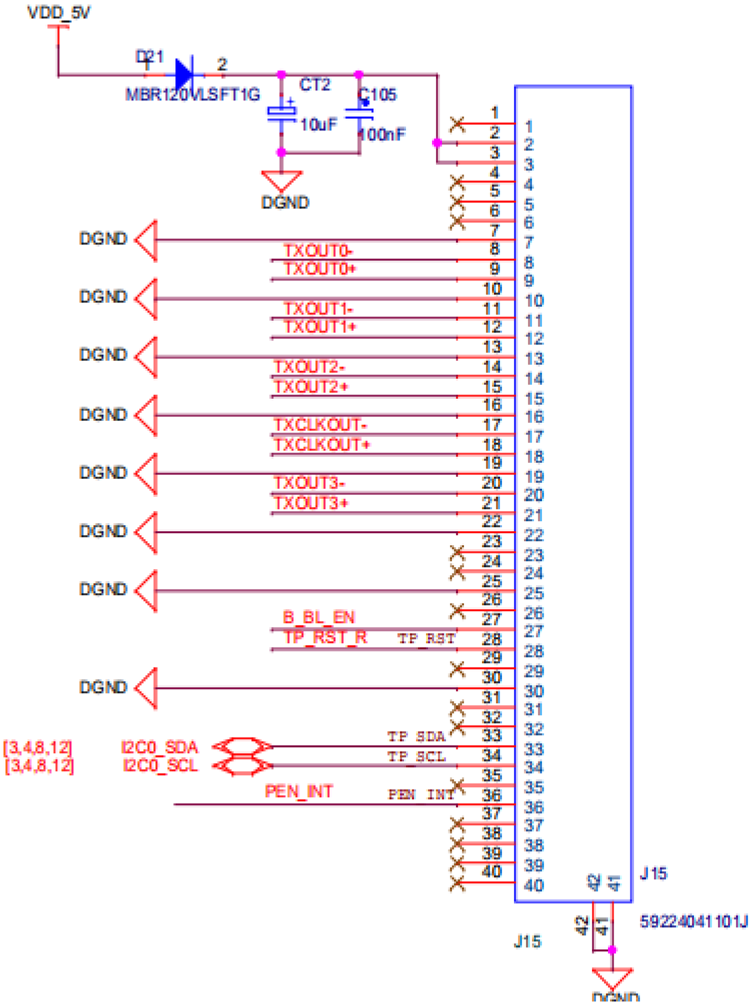
A: Convert RGB to LVDS. Below circuit designs are for your reference.


[3,4,8] LCD_DATA0 LCD_DATA0 R182 0R B3
[3,4,8] LCD_DATA1 LCD_DATA1 R183 0R B4
[3,4,8] LCD_DATA2 LCD_DATA2 R187 0R B5
[3,4,8] LCD_DATA3 LCD_DATA3 R188 0R B6
[3,8] LCD_DATA4 LCD_DATA4 R184 0R B7
[3,8] LCD_DATA5 LCD_DATA5 R181 0R G2
[3,8] LCD_DATA6 LCD_DATA6 R185 0R G3
[3,8] LCD_DATA7 LCD_DATA7 R186 0R G4
[3,8] LCD_DATA8 LCD_DATA8 R189 0R G5
[3,8] LCD_DATA9 LCD_DATA9 R190 0R G6
[3,8] LCD_DATA10 LCD_DATA10 R191 0R G7
[3,8] LCD_DATA11 LCD_DATA11 R192 0R R3
[3,8] LCD_DATA12 LCD_DATA12 R193 0R R4
[3,8] LCD_DATA13 LCD_DATA13 R194 0R R5
[3,8] LCD_DATA14 LCD_DATA14 R195 0R R6
[3,8] LCD_DATA15 LCD_DATA15 R196 0R R7
[3,8] LCD_PCLK VCLK R197 0R B_VCLK
[3,8] LCD_HSYNC HSYNC R198 0R B_HSYNC
[3,8] LCD_VSYNC VSYNC R199 0R B_VSYNC
[3,8] LCD_DE VDEN R200 0R B_VDEN
[8] BL_LCD R201 0R B_BL_EN
[8] TP_INT R202 0R PEN_INT
[8] TP_RST R203 0R TP_RST_R

4-13-1 Example circuit 1



4-13-2 Example circuit 2



4-13-3 Example circuit 3

5. Protocols and tools associated with applications

5.1 How to add database, FTP, sqlite3 or other applications in Yocto?

A: Refer to the Yocto source code which MYIR provides. Below is the directory in QT file system which we will modify.

Add vsftpd, ftp, sqlite3 in

"sources/meta-myir-imx6ulx/recipes-fsl/images/fsl-image-qt5.bbappend"

Example code:

```
    DESCRIPTION = "Freescale Image - Adds Qt5"
    LICENSE = "MIT"
    inherit populate_sdk_qt5
    require recipes-fsl/images/fsl-image-qt5-validation-imx.bb
    IMAGE_FEATURES += "package-management ssh-server-dropbear "
    IMAGE_INSTALL += " \
            imx-kobs \
            tslib \
            tslib-calibrate \
            tslib-conf \
            tslib-tests \
            memtester \
            bzip2 \
            gzip \
            canutils \
            dosfstools \
            mtd-utils \
            mtd-utils-ubifs \
            ntpdate \
            vlan \
            tar \
            net-tools \
            ethtool \
            evtest \
            i2c-tools \
            iperf3 \
            iproute2 \
            iputils \
            udev-extraconf \
            iperf \
            openssl \
            v4l-utils \
            alsa-utils \
            ppp \
            ppp-quectel \
            sqlite3 \
            libmodbus \
            libxml2 \
            dbus \
            openobex \
            hostapd \
            iptables \
            vsftpd \
            openobex \
            myir-rc-local \
    ${@base_contains("MACHINE", "mys6ull14x14", "rtl8188eu-driver", "", d)} \
```

Below is the file we need to modify in core-base file system:

```
    sources/meta-myir-imx6ulx/recipes-core/images/core-image-base.bbappend
```

Then build the file system referring to the development manual, the new file system would include these applications.

5.2 How to modify "rc.local" in Yocto?

A: The dictionary of rc.local:

sources/meta-myir-imx6ulx/recipes-myir/myir-rc-local/myir-rc-local/rc.local.etc

Example code:

```
    #!/bin/sh -e
    #
    # rc.local
    #
    # This script is executed at the end of each multiuser runlevel.
    # Make sure that the script will "exit 0" on success or any other
    # value on error.
    #
    # In order to enable or disable this script just change the execution
    # bits.
    #
    # By default this script does nothing.

    # demo start
    export TSLIB_TSDEVICE=/dev/input/event1
    #ts_calibrate && ts_test &

    export TSLIB_TSDEVICE=/dev/input/event1
    export TSLIB_CONFFILE=/etc/ts.conf
    export TSLIB_CALIBFILE=/etc/pointercal
    export TSLIB_PLUGINDIR=/usr/lib/ts
    export TSLIB_CONSOLEDEVICE=none
    export QT_QPA_FB_TSLIB=1
    export QT_QPA_GENERIC_PLUGINS=tslib:/dev/input/event1

    #add command to display the real time
    date

    exit 0
```

5.3 How to generate ubi file system in Yocto?

Example code:

```
    +++ b/sources/meta-myir-imx6ulx/conf/distro/include/myir-imx-base.inc
@@ -15,7 +15,14 @@ LOCALCONF_VERSION = "1"
 IMX_DEFAULT_DISTRO_FEATURES = "largefile opengl ptest multiarch bluez"
 IMX_DEFAULT_EXTRA_RDEPENDS = "packagegroup-core-boot"
 IMX_DEFAULT_EXTRA_RRECOMMENDS = "kernel-module-af-packet"
-IMAGE_FSTYPES = "tar.bz2 tar.xz ext4 sdcard"
+IMAGE_FSTYPES = "tar.bz2 tar.xz ext4 sdcard ubi"
+
+# Use the expected value of the ubifs filesystem's volume name in the kernel
+UBI_VOLNAME = "rootfs"
+# The biggest NANDs on current modules are 256MB.
+# This sums up to 2048 LEBs available for the ubifs (-c)
+MKUBIFS_ARGS = "-F -m 2048 -e 126976 -c 2048"
+UBINIZE_ARGS = "-m 2048 -p 128KiB -s 2048 -O 2048"
```

5.4 How to compile .c file using the cross-compiler toolchain?

A: Please refer to “$CC -o test test.c”.

5.5 How to modify default GPIO voltage in the source code of uboot?

A: File to modify: uboot/board/myir/myd_y6ull/myd_y6ull.c.

Example code:

```
/* WiFi Reset */
gpio_direction_output(IMX_GPIO_NR(4, 16) , 0);
udelay(3000);
gpio_direction_output(IMX_GPIO_NR(4, 16) , 1);

/* LCD Power */
imx_iomux_v3_setup_multiple_pads(lcd_pwr_pads, ARRAY_SIZE(lcd_pwr_pads));
gpio_direction_output(IMX_GPIO_NR(3, 4) , 1);

/* LTE module */
imx_iomux_v3_setup_multiple_pads(lte_pwr_pads, ARRAY_SIZE(lte_pwr_pads));
/* LTE wakeup */
gpio_direction_output(IMX_GPIO_NR(5, 8) , 1);
/* LTE power */
gpio_direction_output(IMX_GPIO_NR(5, 5) , 1);
/* LTE reset */
gpio_direction_output(IMX_GPIO_NR(4, 14) , 1);
udelay(150000);
gpio_direction_output(IMX_GPIO_NR(4, 14) , 0);
```

5.6 How to modify the debug serial port to another?

A: Modify the source code of uboot by modifying “bootargs=console=ttymxc0” of environment variables, modify “ttymx0” to the code of another serial port.

5.7 How to set a permanent MAC address while Linux generates random MAC addresses?

A: We may buy MAC from IEE and program it into fuse of chip, fuse can be programmed only once. Attention please, program it improperly may affect booting.

5.8 Is there a SN in the system of every MYC-Y6ULX CPU Module?

A: No preset SN in it.

5.9 How to set static IP address?

A: File to modify: /etc/network/interfaces.

Example code (below IP is for reference only):

```
# Wired or wireless interfaces
auto eth0
auto eth1
iface eth1 inet dhcp
iface eth0 inet static
address 192.168.30.122
netmask 255.255.255.0
getway 192.168.30.1
```

MYIR Tech Limited
Room 04, 6th Floor, Building No.2, Fada Road, Yunli Smart Park,
Bantian, Longgang District, Shenzhen, Guangdong, China 518129
E-mail: sales@myirtech.com (Sales Team)
support@myirtech.com (Support Team)
Phone: +86-755-22984836
Fax: +86-755-25532724
Website: http://www.myirtech.com