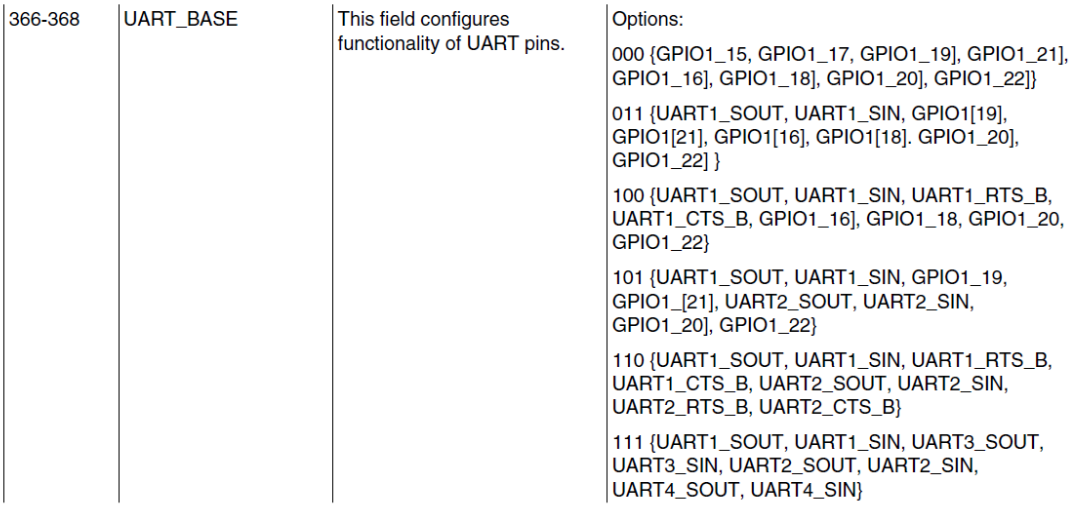
LS1043 GPIO support in Linux SDK

This document introduces how to configure RCW to support GPIO on LS1043 platform, how to configure Linux Kernel to load Linux GPIO driver to access GPIO from SYSFS and using loopback method to do verification on the target board.

1. RCW configuration to support GPIO

On LS1043 UART and GPIO signal multiplexing, users need to configure RCW[UART_BASE] to GPIO, there are 6 options defined in LS1043 reference manual.



366-368	UART_BASE	This field configures functionality of UART pins.	Options:
			000 {GPIO1_15, GPIO1_17, GPIO1_19], GPIO1_21], GPIO1_16], GPIO1_18], GPIO1_20], GPIO1_22]}
			011 {UART1_SOUT, UART1_SIN, GPIO1[19], GPIO1[21], GPIO1[16], GPIO1[18]. GPIO1_20], GPIO1_22] }
			100 {UART1_SOUT, UART1_SIN, UART1_RTS_B, UART1_CTS_B, GPIO1_16], GPIO1_18, GPIO1_20, GPIO1_22}
			101 {UART1_SOUT, UART1_SIN, GPIO1_19, GPIO1_[21], UART2_SOUT, UART2_SIN, GPIO1_20], GPIO1_22}
			110 {UART1_SOUT, UART1_SIN, UART1_RTS_B, UART1_CTS_B, UART2_SOUT, UART2_SIN, UART2_RTS_B, UART2_CTS_B}
			111 {UART1_SOUT, UART1_SIN, UART3_SOUT, UART3_SIN, UART2_SOUT, UART2_SIN, UART4_SOUT, UART4_SIN}

Uses could choose RCW[UART_BASE] option from 0b011, 0b100 and 0b101.

For 0b011, 8 pins are defined, UART1_SOUT, UART1_SIN, GPIO1[19], GPIO1[21], GPIO1[16], GPIO1[18]. GPIO1[20], GPIO1[22], 2 pins for UART1 and 6 pins for GPIO. In this option, UART1 doesn't have RTS & CTS pins, so it doesn't support hardware flow control.

For 0b100, 8 pins are defined, UART1_SOUT, UART1_SIN, UART1_RTS_B, UART1_CTS_B, GPIO1[16], GPIO1[18], GPIO1[20], GPIO1[22], 4 pins for UART1 and 4 pins for GPIO. In this option, UART1 has RTS & CTS pins so it supports Hardware flow control.

For 0b101, 8 pins are defined, UART1_SOUT, UART1_SIN, GPIO1[19], GPIO1[21], UART2_SOUT, UART2_SIN, GPIO1[20], GPIO1[22], 2 pins for UART1, 2 pins for UART2, 4 pins for GPIO. In this option, 2 UART ports are enabled and with extra 4 GPIO pins. Both UART ports have no hardware flow control support.

The users need to configure RCW[UART_BASE] according to their requirement about UART and GPIO. If UART hardware flow control is required, users could choose 0b100 and use UART1. If no hardware flow control required, users could choose one of above three options according to GPIO pins number.

In Linux SDK, RCW[UART_BASE] is configured as 0b111 by default, the user needs to modify RCW configuration to support GPIO. The following introduces how to regenerate RCW binary file in Linux SDK, RCW[UART_BASE] will be configured as 0b100.

```
$ bitbake rcw -c cleansstate
$ bitbake rcw -c configure
```

Go to the source folder build_ls1043ardb/tmp/work/ls1043ardb-fsl-linux/rcw/git-r0/git/ls1043ardb/RR_FQPP_1455, configure UART_BASE as 4 in rcw_1600.rcw, go back to the build folder.

$ bitbake rcw
Get rcw_1600.bin in build_ls1043ardb/tmp/deploy/images/ls1043ardb/rcw/ls1043ardb/RR_FQPP_1455.

2. Configure GPIO driver in Linux Kernel

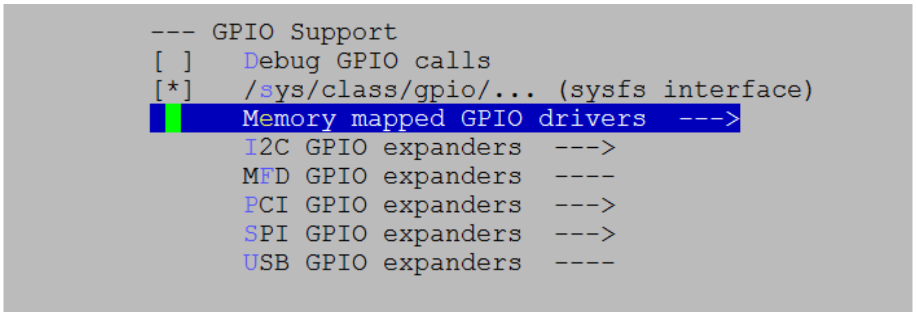
LS1043 platform could use GPIO driver for PowerPC architecture. Individual GPIO signals may be accessed from userspace, GPIOs are only made available to userspace by an explicit “export” operation. The user needs to configure the following Kernel options as the following.

```
CONFIG_ARCH_WANT_OPTIONAL_GPIOLIB=y
CONFIG_ARCH_REQUIRE_GPIOLIB=y
CONFIG_GPIOLIB=y
CONFIG_GPIO_DEVRES=y
CONFIG_OF_GPIO=y
CONFIG_GPIOLIB_IRQCHIP=y
CONFIG_GPIO_SYSFS=y
CONFIG_GPIO_GENERIC=y
CONFIG_GPIO_GENERIC_PLATFORM=y
CONFIG_GPIO_MPC8XXX=y
CONFIG_GPIO_XGENE=y
CONFIG_DEBUG_FS=y
```

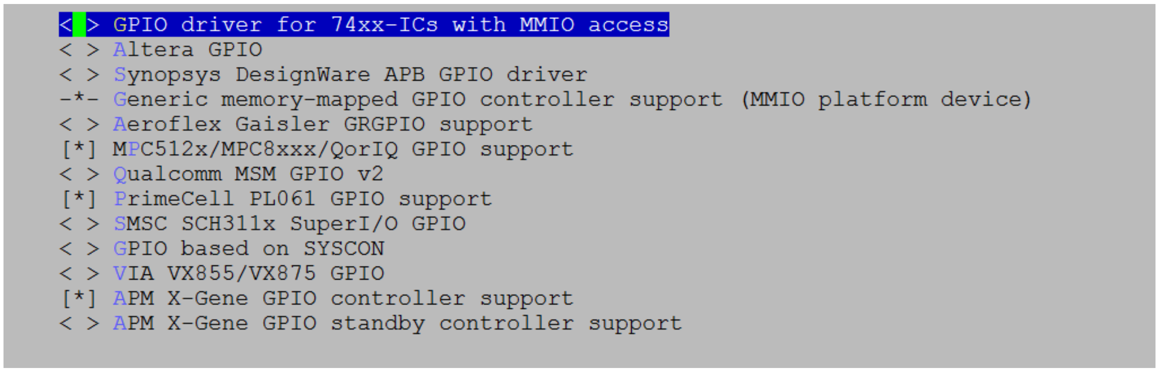
The user could use the following procedure to rebuild Linux Kernel with GPIO enabled.

```
$ bitbake virtual/kernel -c cleansstate
$ bitbake virtual/kernel -c configure
```

Go to the folder /home/b25805/test/QorIQ-SDK-V2.0-20160527-yocto/build_ls1043ardb/tmp/work/ls1043ardb-fsl-linux/linux-qoriq/4.1-r0/build/ to configure Kernel with the command “*make ARCH=arm64 menuconfig”.*



```
--- GPIO Support
[ ]   Debug GPIO calls
[*]   /sys/class/gpio/... (sysfs interface)
      Memory mapped GPIO drivers  --->
      I2C GPIO expanders  --->
      MFD GPIO expanders  ----
      PCI GPIO expanders  --->
      SPI GPIO expanders  --->
      USB GPIO expanders  ----
```



```
< > GPIO driver for 74xx-ICs with MMIO access
< > Altera GPIO
< > Synopsys DesignWare APB GPIO driver
-*- Generic memory-mapped GPIO controller support (MMIO platform device)
< > Aeroflex Gaisler GRGPIO support
[*] MPC512x/MPC8xxx/QorIQ GPIO support
< > Qualcomm MSM GPIO v2
[*] PrimeCell PL061 GPIO support
< > SMSC SCH311x SuperI/O GPIO
< > GPIO based on SYSCON
< > VIA VX855/VX875 GPIO
[*] APM X-Gene GPIO controller support
< > APM X-Gene GPIO standby controller support
```

Regenerate Kernel itb image

$ bitbake fsl-image-kernelitb

3. Verify GPIO on the target board

Program RCW on NOR flash and reset the target board.

```
=> tftp 82000000 b25805/ls1043ardb/rcw_1600.bin

=> protect off 60000000 +$filesize

=> erase 60000000 +$filesize

=> cp.b 82000000 60000000 $filesize

=> protect on 60000000 +$filesize

=> reset
```

u-boot console log is as the following:

```
U-Boot 2016.012.0+g187aa26 (Apr 05 2016 - 16:47:25 +0800)

SoC:  unknown (0x87920410)

Clock Configuration:

       CPU0(A53):1600 MHz  CPU1(A53):1600 MHz  CPU2(A53):1600 MHz

       CPU3(A53):1600 MHz

       Bus:      400  MHz  DDR:      1600 MT/s  FMAN:     500  MHz

Reset Configuration Word (RCW):

       00000000: 08100010 0a000000 00000000 00000000

       00000010: 14550002 80004012 e0025000 c1002000

       00000020: 00000000 00000000 00000000 00020800

       00000030: 00000000 00001101 00000096 00000001

I2C:   ready

Model: LS1043A RDB Board

Board: LS1043ARDB, boot from vBank 4
```

Under u-boot set CFG_MUX_UART_S0/1 under uboot to bring the GPIO pins out for testing.

```
=>  mw.b 7fb0005e  0x02

GPIO1_16     < ----- > JP1_5

GPIO1_18     < ----- >  JP1_4

GPIO1_19     < ----- >  JP3_5

GPIO1_21      < ----- > JP3_4

GPIO1_20      < ---- -> JP2_5
```

GPIO1_22   < ---- -> JP2_4

The following section describes how to verify GPIO under Linux.

GPIOs are defied as they are inside the kernel, using integers in the range 0 .. INT_MAX.

```
 /sys/class/gpio
     /export : asks the kernel to export a GPIO to userspace
     /unexport: to return a GPIO to the kernel
     /gpioN: for each exported GPIO #N
          /value: always readable, writes fail for input GPIOs
          /direction: r/w as in, out (default low); write(high, low)
          /edge: r/w as none, falling, rising, both
     /gpiochipN: for each gpiochip; #N is its first GPIO
          /base: (r/o) same as N
          /label: (r/o) descriptive, not necessarily unique
          /ngpio: (r/o) number of GPIOs; numbered N to N + (ngpio - 1)
```

GPIO block mapping could be get as the following.

```
QorIQ SDK (FSL Reference Distro) 2.0 ls1043ardb /dev/ttyS0
ls1043ardb login: root
root@ls1043ardb:~# ls /sys/class/gpio/
export       gpiochip384/ gpiochip416/ gpiochip448/ gpiochip480/ unexport
root@ls1043ardb:~# cat /sys/kernel/debug/gpio
GPIOs 384-415, /soc/gpio@2330000:
GPIOs 416-447, /soc/gpio@2320000:
GPIOs 448-479, /soc/gpio@2310000:
GPIOs 480-511, /soc/gpio@2300000:
```

GPIO496(480+16) corresponds with GPIO1_16.

Verify GPIO as output as the following.

```
root@ls1043ardb:/sys/class/gpio# echo 496 > export
root@ls1043aqds:/sys/class/gpio# ls
export  gpio496  gpiochip384  gpiochip416  gpiochip448  gpiochip480  unexport
root@ls1043ardb:/sys/class/gpio# echo "out" > gpio496/direction
```

```
root@ls1043ardb:/sys/class/gpio# echo 1 >  gpio496/value
root@ls1043ardb:/sys/class/gpio# echo 0 >  gpio496/value
root@ls1043ardb:/sys/class/gpio# echo "in" > gpio496/direction
root@ls1043ardb:/sys/class/gpio# cat gpio496/value
1
```

GPIO is used as input and check the interrupt counter.

```
root@ls1043ardb:/sys/class/gpio# ls /sys/class/gpio/
export  gpio496  gpiochip384  gpiochip416  gpiochip448  gpiochip480  unexport le
root@ls1043ardb:/sys/class/gpio# cd gpio496/
root@ls1043ardb:/sys/class/gpio/gpio496# ls
active_low  direction  edge  power  subsystem  uevent  value
root@ls1043ardb:/sys/class/gpio/gpio496# echo out > direction
root@ls1043ardb:/sys/class/gpio/gpio496# echo both > edge
[  562.634440] GPIO chip /soc/gpio@2300000: gpiochip_lock_as_irq: tried to flag a GPIO set as output for IRQ
root@ls1043ardb:/sys/class/gpio/gpio496# cat /proc/interrupts | grep 8xxx
 70:          1          0          0          0  mpc8xxx-gpio  16 Edge      gpiolib, gpiolib
root@ls1043ardb:/sys/class/gpio/gpio496# echo 1 > value
root@ls1043ardb:/sys/class/gpio/gpio496# echo 0 > value
root@ls1043ardb:/sys/class/gpio/gpio496# echo 1 > value
root@ls1043ardb:/sys/class/gpio/gpio496# cat /proc/interrupts | grep 8xxx
 70:          4          0          0          0  mpc8xxx-gpio  16 Edge      gpiolib, gpiolib
root@ls1043ardb:/sys/class/gpio/gpio496#
```