
Embedded System Design

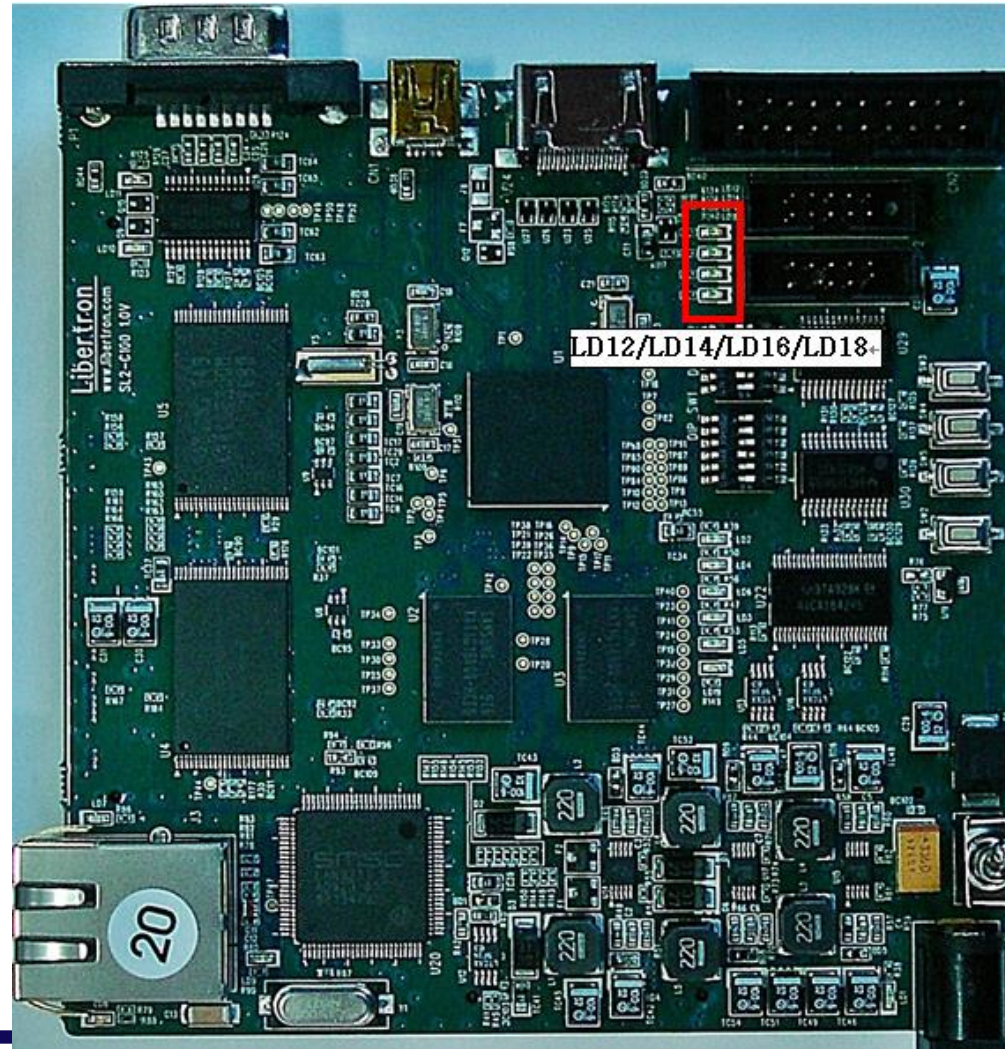
Practice 9

TaeWook Kim & SeokHyun Hong
Hanyang University

GPIO (LED)



LED in SP5PC100 Board



GPIO & LED

- LED is connected to pins 0 ~ 3 of GPJ2

SP5PC100 Chip



No.	Reference	Description
12	LD12	XmsmDATA/CF_DATA0
13	LD14	XmsmDATA/CF_DATA1
14	LD16	XmsmDATA/CF_DATA2
15	LD18	XmsmDATA/CF_DATA3

User LED (GPIO)

Define Register Address

- **GPIO related registers**
 - GPJ2CON
 - Control register of GPJ2 Port
 - Set the input/output mode of pins of each port
 - GPJ2DAT
 - Data register of GPJ2 Port
 - Output pin : When a value is written to the corresponding register, the data is transmitted to the outside via the pin
 - Input pin : Indicates current pin status

GPJ2CON

- **Port Group GPJ2 Configuration Register (0xe0300240)**

- Set each pin of the corresponding port as an input pin or an output pin
- If the specific hardware is physically connected, connect it to the hardware.
ex) UART

Field	Bit	Description	Reset Value
GPJ2CON[0]	[3:0]	0000 = Input, 0001 = Output, 0010 = MSM_D[0], 0011 = Reserved, 0100 = CF_D[0] 1111 = NWU_INT18[0]	0000
GPJ2CON[1]	[7:4]	0000 = Input, 0001 = Output, 0010 = MSM_D[1] , 0011 = Reserved, 0100 = CF_D[1] , 1111 = NWU_INT18[1]	0000
GPJ2CON[2]	[11:8]	0000 = Input, 0001 = Output, 0010 = MSM_D[2] , 0011 = Reserved, 0100 = CF_D[2] , 1111 = NWU_INT18[2]	0000
GPJ2CON[3]	[15:12]	0000 = Input, 0001 = Output, 0010 = MSM_D[3] , 0011 = Reserved, 0100 = CF_D[3] , 1111 = NWU_INT18[3]	0000

GPJ2DAT

- **Port Data Register (0xe0300244)**

- Consists of 8 bits. Bit correspond to each pin
- When pin is used as input, it indicates the state of the current pin.
- When pin is used as output , it stores data to be output

Field	Bit	Description	Reset Value
DAT[n] (n=0~7)	[n]	If the bit is configured as input, it represents the pin state. If the bit is configured as output, the pin state is the same as the value of the bit. If the port is configured as functional pin, an undefined value is read.	-

LED Initialization

- 1. Set pins 0 to 3 as output pins in GPJ2CON**
 - Write 0001 to the field corresponding to each pin

- 2. Write 0x0 to GPJ2DAT**
 - Initialize the GPJ2DAT register

LED On/Off

- Store 1 in the desired bit (pin) of the GPJ2DAT register
→ LED On
- Store 0 in the desired bit (pin) of the GPJ2DAT register
→ LED Off

VPOS_kernel_main()

- **Functions**

- Initialize the VPOS kernel data structure
- Initialize hardware such as serial device and timer
- Enable interrupt
- Print boot message
- Create a shell thread
- Enter scheduler calling
VPOS_start routine

- **Source code location**

- vpos/kernel/kernel.start.c

```
void VPOS_kernel_main( void )
{
    pthread_t p_thread, p_thread_0, p_thread_1, p_thread_2;

    /* static and global variable initialization */
    vk_scheduler_unlock();
    init_thread_id();
    init_thread_pointer();
    vh_user_mode = USER_MODE;
    vk_init_kdata_struct();

    vk_machine_init();
    set_interrupt();

    printk("%s\n%s\n%s\n", top_line, version, bottom_line);

    /* initialization for thread */
    race_var = 0;
    pthread_create(&p_thread, NULL, UPOS_SHELL, (void *)NULL);
    pthread_create(&p_thread_0, NULL, race_ex_1, (void *)NULL);
    pthread_create(&p_thread_1, NULL, race_ex_0, (void *)NULL);
    pthread_create(&p_thread_2, NULL, race_ex_2, (void *)NULL);

    UPOS_start();

    /* cannot reach here */
    printk("OS ERROR: VPOS_kernel_main( void )\n");
    while(1){}
}
```

vk_machine_init()

- **Code**

- Hardware device initialization
- vh_serial_init(): Initializing UART
- vh_timer_init(): Initializing Timer
- **vh_LedInit() : Initializing LED and
Turn on LED**

```
void vk_machine_init(void)
{
    vh_LedInit();
    vh_serial_init();
    vh_timer_init();
}
```

- **Location**

- vpos/kernel/machine_init.c

vh_LedInit()

- **Purpose**
 - Initialize GPIO setting and LED
 - Test that all four LEDs work properly
 - Turn on LEDs 0 ~ 3 in order. Repeat this 5 times
- **Location**
 - vpos/hal/io/led.c



vh_LedInit()

- Code

```
#define DELAY          0x10000
```

```
void vh_LedInit(void)
{
```

```
    int i,j, dly;
```

```
    vh_GPJ2CON =
```

```
    vh_GPJ2DAT = 0x0;
```

```
    for (i=0; i<5; i++) {
```

```
        for (j=0; j<4; j++) {
```

```
            vh_LedSet(j);
```

```
            for(dly=0; dly<DELAY; dly++);
```

```
        }
```

```
    }
```

```
    vh_GPJ2DAT = 0;
```

```
}
```

Set pins 0 to 3 as output pins

We have to define
GPIO registers

Turn On the j-th LED

Turn Off the LED

vh_LedSet()

- **Purpose**
 - Turn on the LED corresponding to the parameter
- **Location**
 - vpos/hal/io/led.c

vh_LedSet()

- Code

```
void vh_LedSet(unsigned char data)
{
    switch(data) {
        case 0: vh_GPJ2DAT = 0x1; break;
        case 1: vh_GPJ2DAT = ; break;
        case 2: vh_GPJ2DAT = ; break;
        case 3: vh_GPJ2DAT = ; break;
    }
}
```

Turn on 0th LED

Thank you

