

# practice05

| 2021011158 김선희

## Related Concepts

### Memory-Mapped I/O

: I/O address space and memory address space exist in one memory space.

- Treat the memory and registers of an I/O device as part of the whole memory.
- UART address: The address which is used for data transfer between embedded board and pc.
  - UART: Handles interrupts from I/O devices such as keyboards and mouse and converts signal from them into signals that the computer can read.

## Kernel Porting | UART Settings

### UART Register Setting

To control hardware, we use registers. To use these registers, we need to know exact address of each register points to.

Currently, the address for UART register is not explicitly set. So we need to set the address so that it maps to the correct address in memory space, when we use it.

#### UART clock & control register setting

##### *serial.c*

```
char getc(void)
{
    char c;
    unsigned long rxstat;
    /* Write getc func */
    while (UARTFR & UARTFR_RXFE);

    c = (char) UARTDR;

    rxstat = UARTRSR & UART_ERR_MASK;

    /* End getc func */

    // while (pop_idx == push_idx);

    // c = serial_buff[pop_idx++];

    // if (pop_idx == SERIAL_BUFF_SIZE)
    //     pop_idx = 0;

    // return c;
}

void vh_serial_init(void)
{
    // set baud rate
    unsigned int idiv, fdiv;
    /* baud rate Here */
    float divisor = (float) UART_CLK / (16 * UART_BAUDRATE);
    idiv = (unsigned int) divisor;
    fdiv = (unsigned int) ((divisor - idiv) * 64 + 0.5);

    /* baud rate End */
}
```

```

UARTIBRD = idiv;
UARTFBRD = fdiv;

// set UART ctrl regs
UARTLCR_H = 0x76;
UARTCR = 0x301;
UARTMSC = 0xF9EF;
UARTIFLS = 0x4;

// clear buffer
push_idx = 0;
pop_idx = 0;
for(int i=0; i<SERIAL_BUFF_SIZE; i++)
    serial_buff[i] = '\0';
}

```

## UART base address setting

### vh\_io\_hal.h

```

/*****
UART address
*****/

#define UART_BASE_ADDR    0x09000000
#define UART_CLK          24000000

#define UARTDR            (*(volatile unsigned int*)(UART_BASE_ADDR + 0x000))
#define UARTRSR           (*(volatile unsigned int*)(UART_BASE_ADDR + 0x004))
#define UARTFR            (*(volatile unsigned int*)(UART_BASE_ADDR + 0x018))
#define UARTIBRD          (*(volatile unsigned int*)(UART_BASE_ADDR + 0x024))
#define UARTFBRD          (*(volatile unsigned int*)(UART_BASE_ADDR + 0x028))
#define UARTLCR_H         (*(volatile unsigned int*)(UART_BASE_ADDR + 0x02C))
#define UARTCR            (*(volatile unsigned int*)(UART_BASE_ADDR + 0x030))
#define UARTIFLS          (*(volatile unsigned int*)(UART_BASE_ADDR + 0x034))
#define UARTMSC           (*(volatile unsigned int*)(UART_BASE_ADDR + 0x038))
#define UARTRMIS          (*(volatile unsigned int*)(UART_BASE_ADDR + 0x040))
#define UARTRICR          (*(volatile unsigned int*)(UART_BASE_ADDR + 0x044))

/* Register Bit Setting */
#define UARTFR_TXFF       (1 << 5)
#define UARTFR_RXFE      (1 << 4)

/* your choice */
#define UART_BAUDRATE     115200
#define UART_ERR_MASK     0XF

```

## Result

```

20210111158> go 40000000
## Starting application at 0x40000000 ...

*****
*  QURIX version 3.0    xx/10/2012    *
*****

Race condition value = 0

Shell>typing available

```