

GPIO IN & OUT VIA REGISTERS

Handong university

Jong-won Lee

References

2

- RM0383 Reference manual, STM32F411xC/E advanced Arm[®]-based 32-bit MCUs, Sep. 2017.
- STM32F411xC STM32F411xE Datasheet, Dec. 2017.

3

GPIO

- GPIO IN & OUT

GPIO

4

- GPIO Base address
 - GPIOA: 0x4002_0000
 - GPIOB: 0x4002_0400
 - GPIOC: 0x4002_0800
 - GPIOD: 0x4002_0C00
 - GPIOE: 0x4002_1000
 - GPIOH: 0x4002_1C00

GPIO Registers

21/21

5

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOx_MODER (where x = C..E and H)	MODER15[1:0]																															
0x04	GPIOx_OTYPER (where x = A..E and H)	Reserved																															
0x08	GPIOx_OSPEEDR (where x = C..E andH)	OSPEEDR15[1:0]	OSPEEDR14[1:0]	OSPEEDR13[1:0]	OSPEEDR12[1:0]	OSPEEDR11[1:0]	OSPEEDR10[1:0]	OSPEEDR9[1:0]	OSPEEDR8[1:0]	OSPEEDR7[1:0]	OSPEEDR6[1:0]	OSPEEDR5[1:0]	OSPEEDR4[1:0]	OSPEEDR3[1:0]	OSPEEDR2[1:0]	OSPEEDR1[1:0]	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0	
	Reset value	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0			

GPIO Registers

8

GPIOA
D ~ C5
[6bit]
(16bit port)

□ GPIO port mode register (GPIOx_MODER)

□ Address offset: 0x00

b15		30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

■ Configuration bits



- 00: Input (reset state except for PA13, PB3, and PB4)
- 01: General purpose output mode
- 10: Alternate function mode *다른 용도...*
- 11: Analog mode

[ADC/DAC]

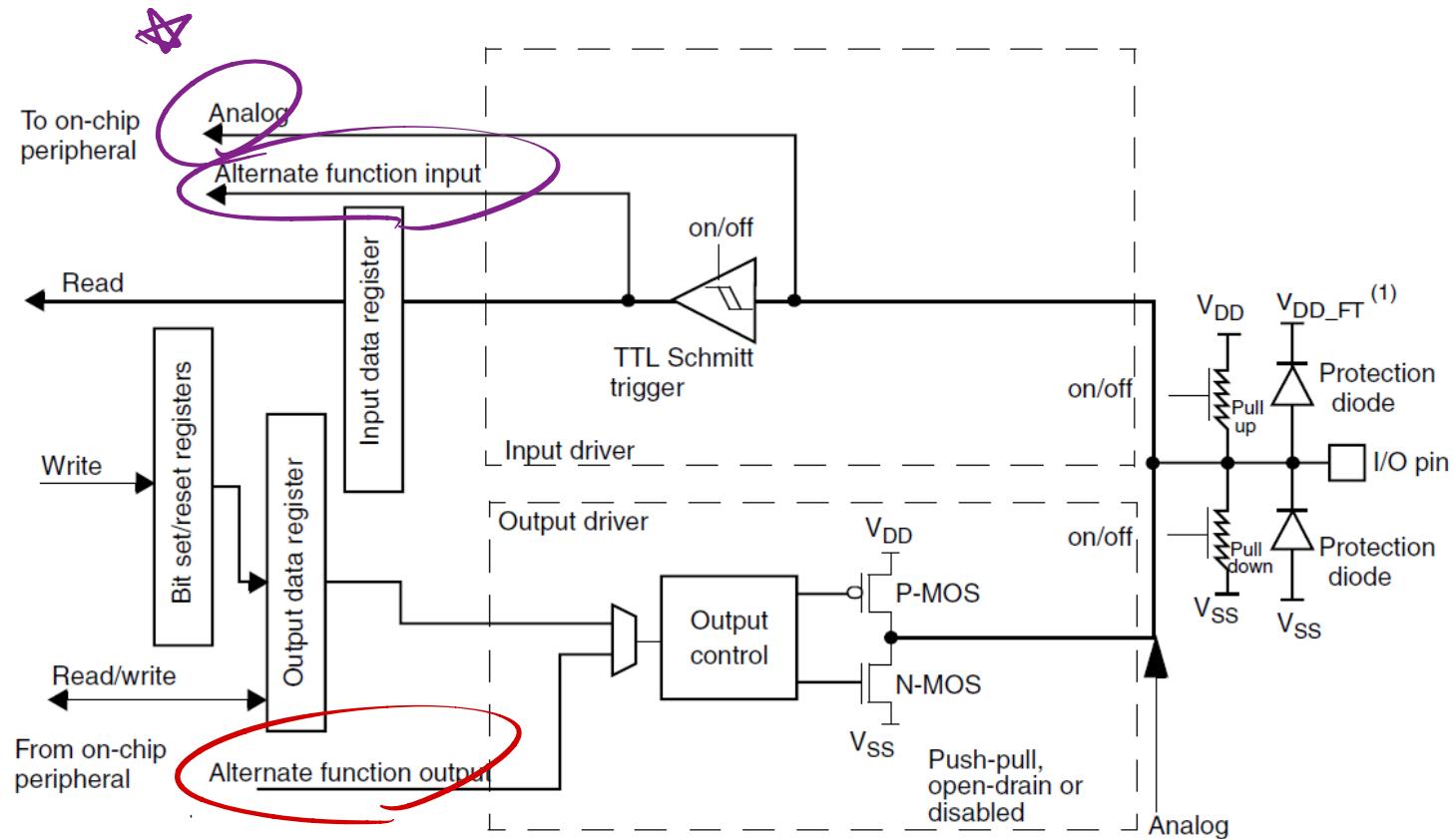
디지털 예외

GPIOA의
port(을)은 input

GPIO

9

Block diagram



GPIO Registers

10

- GPIO port output type register (GPIOx_OTYPER)
 - Address offset: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0

- Configuration bits

- 0: Output push-pull (reset state)
- 1: Output open-drain

Lecture (2019/7/10 09:00 AM)
How to control the pins
open drain)

GPIO Registers

11

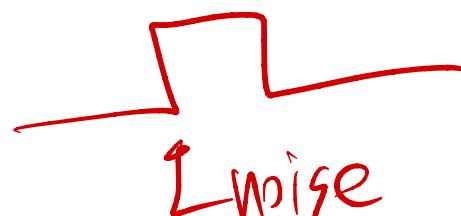
□ GPIO port output speed register (GPIOx_OSPEEDR)

□ Address offset: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]	OSPEEDR14 [1:0]	OSPEEDR13 [1:0]	OSPEEDR12 [1:0]	OSPEEDR11 [1:0]	OSPEEDR10 [1:0]	OSPEEDR9 [1:0]	OSPEEDR8 [1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]	OSPEEDR6[1:0]	OSPEEDR5[1:0]	OSPEEDR4[1:0]	OSPEEDR3[1:0]	OSPEEDR2[1:0]	OSPEEDR1 [1:0]	OSPEEDR0 [1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

■ Configuration bits

- {
 - 00: Low speed (reset state except for PA13 and PB3)
 - 01: Medium speed
 - 10: Fast speed
 - 11: High speed



GPIO Registers

12

- GPIO port pull-up/pull-down register (**GPIOx_PUPDR**)
 - Address offset: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw										

- Configuration bits

- 00: No pull-up, pull-down (reset state except for PA15, PA14, PA13, PB4)
- 01: Pull-up
- 10: Pull-down
- 11: Reserved

GPIO Registers

13

□ GPIO port input data register (GPIOx_IDR)

- Address offset: 0x10

~~232~~ word (4byte)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

- These bits are read-only and can be accessed in word mode only.

$a = \text{GPIO A} \rightarrow \text{IDR}$
↳ Int uint32_t

GPIO Registers

323

14

□ GPIO port output data register (**GPIOx_ODR**)

- Address offset: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

GPIO Registers

GPIO Register (\rightarrow)

15

GPIO port bit set/reset register (GPIOx_BSRR)

Address offset: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Configuration bits

Bits 31:16

- 0: No action on the corresponding ODRx bit
- 1: Resets the corresponding ODRx bit

Bits 15:0

- 0: No action on the corresponding ODRx bit
- 1: Sets the corresponding ODRx bit

Note: If both BSx and BRx are set, **BSx has priority**.

Quiz P.
설치 (설치)에 기관.

GPIO Registers

16

- GPIO port configuration lock register (**GPIOx_LCKR**)
 - Address offset: 0x1C
 - Used to lock the configuration of the each port bit when a correct write sequence is applied to bit 16.
 - When the LOCK sequence has been applied on a port bit, the value of this port bit can no longer be modified until the next MCU or peripheral reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	active high
Reserved																LCKK rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

2. SW 주제는 못바뀜

L bit 7은 못바뀜

L reset은 초기화만 가능

GPIO Registers

17

□ GPIO port configuration lock register (GPIOx_LCKR) (con't)

□ Configuration bits

■ Bits 16

- 0: Port configuration lock key not active
- 1: Port configuration lock key active.

■ Bits 15:0

- 0: Port configuration not locked
- 1: Port configuration locked

□ LOCK key sequence

■ WR LCKR[16] = '1' + LCKR[15:0]

■ WR LCKR[16] = '0' + LCKR[15:0]

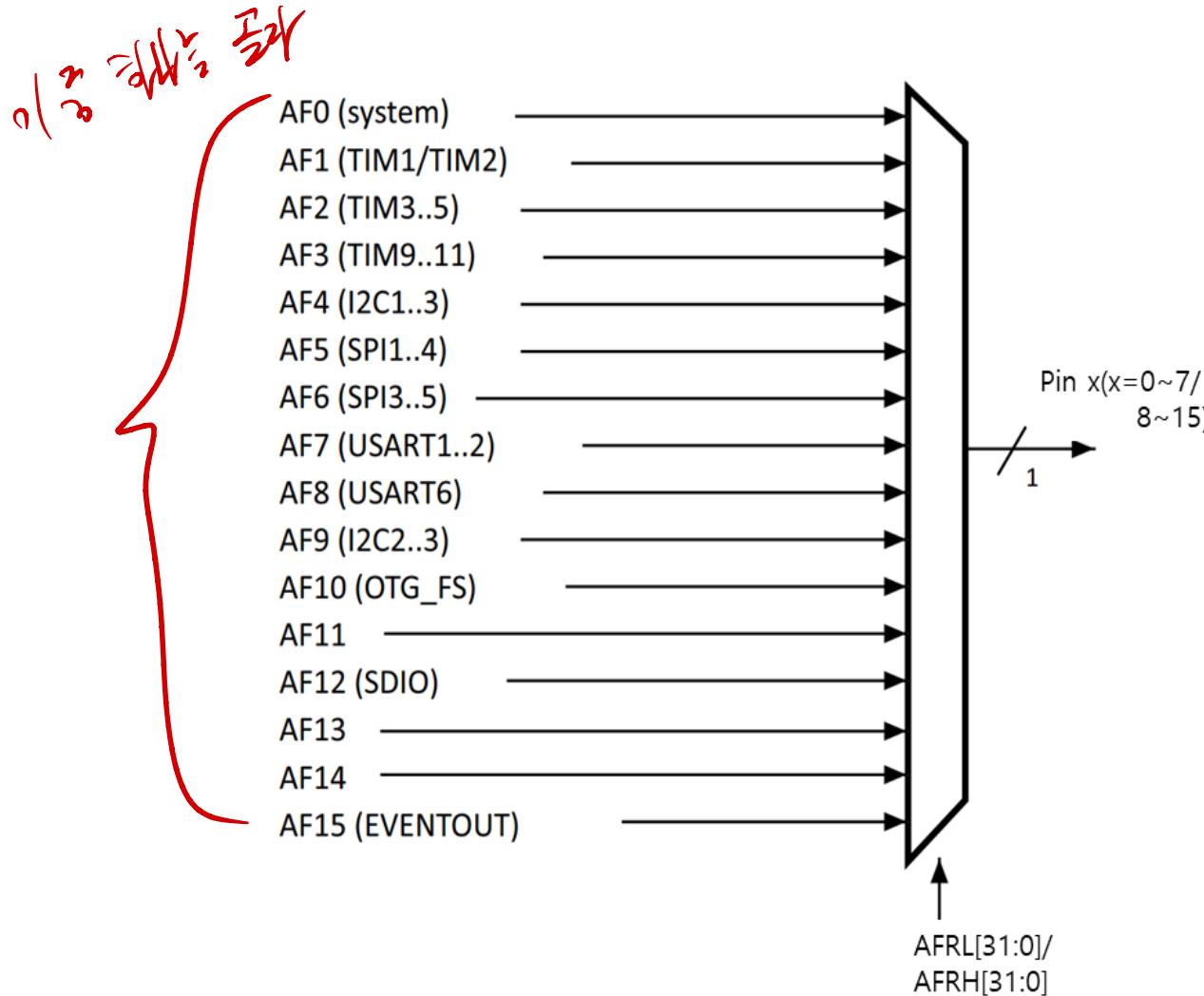
■ WR LCKR[16] = '1' + LCKR[15:0]

■ RD LCKR

■ RD LCKR[16] = '1' (this read operation is optional but it confirms that the lock is active)

GPIOx_AFRL/_AFRH

19



GPIO Alternate Functions

20

□ Examples for PA0 ~ PA4

PORT	AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
	SYS_AF	TIM1/ TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/ I2C2/ I2C3	SPI1/ I2S1/ SPI2/ I2S2/ SPI3/ I2S3	SPI2/ I2S2/ SPI3/ I2S3/ SPI4/ I2S4/ SPI5/ I2S5	SPI3/ I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS	SDIO				
PA0	-	TIM2_CH1/ TIM2_ETR	TIM5_CH1	-	-	-	-	USART2_CTS	-	-	-	-	-	-	-	EVENT OUT
PA1	-	TIM2_CH2	TIM5_CH2	-	-	SPI4_MOSI/ I2S4_SD	-	USART2_RTS	-	-	-	-	-	-	-	EVENT OUT
PA2	-	TIM2_CH3	TIM5_CH3	TIM9_CH1	-	I2S2_CKIN	-	USART2_TX	-	-	-	-	-	-	-	EVENT OUT
PA3	-	TIM2_CH4	TIM5_CH4	TIM9_CH2	-	I2S2_MCK	-	USART2_RX	-	-	-	-	-	-	-	EVENT OUT

GPIO Port Configuration

21

□ 01: output mode

MODER(i) [1:0]	OTYPER(i) Output type	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]	I/O configuration
01 01 01 01 11 11 11 11	0	SPEED [B:A]	0 0 no	GP output PP
	0		0 1	PP + PU pull up
	0		1 0	PP + PD pull down
	0		1 1 Reserved	
	1		0 0	GP output open-drain
	1		0 1	OD + PU
	1		1 0	OD + PD
	1		1 1	Reserved (GP output OD)

00 - input

01 - output

10 - Alternative func

11 - Analog

0 - push pull

1 - open drain

00 - low

01 - medium

10 - Fast

11 - High

00 - no

01 - pull-up

10 - pull-down

11 - reserved

GPIO Port Configuration

22

- 00: input mode, 10: AF mode

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O configuration	
10 <i>Alternative</i>	0	SPEED [B:A]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	Reserved	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	Reserved	
	x	x	x	0	0	Input	Floating ?
00 <i>input</i>	<i>out pin</i>	<i>alt pin</i>	x	0	1	Input	PU
	<i>pd</i>	<i>pu</i>	x	1	0	Input	PD
	<i>br</i>	<i>br</i>	x	1	1	Reserved (input floating)	

Input pullup/down

23

stm32f411xe.h

7月2

Data structures related to GPIOs

24

```
#define PERIPH_BASE          0x40000000U /*!< Peripheral base address in the alias region */  
#define AHB1PERIPH_BASE      (PERIPH_BASE + 0x00020000U)  
  
#define GPIOA_BASE           (AHB1PERIPH_BASE + 0x0000U)  
#define GPIOB_BASE           (AHB1PERIPH_BASE + 0x0400U)  
#define GPIOC_BASE           (AHB1PERIPH_BASE + 0x0800U)  
#define GPIOD_BASE           (AHB1PERIPH_BASE + 0x0C00U)  
#define GPIOE_BASE           (AHB1PERIPH_BASE + 0x1000U)  
#define GPIOH_BASE           (AHB1PERIPH_BASE + 0x1C00U)
```

```
29 }  
30  
31  
32 void led2_init(void){  
33     __HAL_RCC_GPIOA_CLK_ENABLE();  
34     GPIOA->MODER &= (~0b11<<2) << 10; //PA_5  
35     GPIOA->MODER |= 0b01 << 10; // Output mode  
36     GPIOA->  
37     MODER: volatile uint32_t  
38     AFR: uint32_t  
39     void le: uint32_t  
40     GPI: uint32_t  
41     }  
42     void bu: uint32_t  
43     _H: uint32_t  
44     GPIOC->MODER &= (~0b11<<2) << 26; // PC_13  
45     GPIOC->MODER |= 0b00 << 26; // Input mode  
46  
47     GPIOC->PUPDR &= (0b10<<26);  
48  
49     GPIOC->PUPDR |= (0b10<<26);
```

Data structures related to GPIOs

25

```
typedef struct {  
    __IO uint32_t MODER; /*!< GPIO port mode register, Address offset: 0x00 */  
    __IO uint32_t OTYPER; /*!< GPIO port output type register, Address offset: 0x04 */  
    __IO uint32_t OSPEEDR; /*!< GPIO port output speed register, Address offset: 0x08 */  
    __IO uint32_t PUPDR; /*!< GPIO port pull-up/pull-down register, Address offset: 0x0C */  
    __IO uint32_t IDR; /*!< GPIO port input data register, Address offset: 0x10 */  
    __IO uint32_t ODR; /*!< GPIO port output data register, Address offset: 0x14 */  
    __IO uint32_t BSRR; /*!< GPIO port bit set/reset register, Address offset: 0x18 */  
    __IO uint32_t LCKR; /*!< GPIO port configuration lock register, Address offset: 0x1C */  
    __IO uint32_t AFR[2]; /*!< GPIO alternate function registers, Address offset: 0x20-0x24 */  
} GPIO_TypeDef;
```

4byte 4h 증가

```
#define GPIOA ((GPIO_TypeDef *) GPIOA_BASE)  
#define GPIOB ((GPIO_TypeDef *) GPIOB_BASE)  
#define GPIOC ((GPIO_TypeDef *) GPIOC_BASE)  
#define GPIOD ((GPIO_TypeDef *) GPIOD_BASE)  
#define GPIOE ((GPIO_TypeDef *) GPIOE_BASE)  
#define GPIOH ((GPIO_TypeDef *) GPIOH_BASE)
```

GPIOA의 mode register reset? | GPIOA->MODER = 0

GPIO Register Handling

- LED
- Ex. 1: how to configure PA5 to output mode

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]	MODER14[1:0]	MODER13[1:0]	MODER12[1:0]	MODER11[1:0]	MODER10[1:0]	MODER9[1:0]	MODER8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]	MODER6[1:0]	MODER5[1:0]	MODER4[1:0]	MODER3[1:0]	MODER2[1:0]	MODER1[1:0]	MODER0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOA->MODER &= ~(0b11 << 10); 2bit masking (00~03 set) &
 GPIOA->MODER |= (0b01 << 10); (0~3 set)

10bit 왼쪽으로 shift

HAL_RCC_GPIOA_CLK_ENABLE();

?

CLK을 각각 따로 공급하되, 그 어떤 주사위 기능을 사용하지 않고

CLK을 공유해 줄수 있음

GPIO IN & OUT VIA REGISTERS

Handong university

Jong-won Lee

Lab. 2-1

2

- Control the blinking rate of the LED 2 by the User button.
 - Whenever the user button is pressed, change the blinking rate as follows: 500 ms on/off → 1 sec on/off → 2 sec on/off → 500 ms on/off → ...
- Implement the following functions by directly handling registers.

void led2_init(void);

void button_init(void);

uint32_t button_input(void); // return value: 0 or 1

void led2_toggle(void);

PA-5

PC-13

GPIOA-5 초기화

GPIOC-13 초기화

GPIOC-13의 값을 0으로
설정 | return

[값은 0/1]

Lab. 2-1

3

□ A skeleton code.

```
#include "mbed.h"

void led2_init(void);
void button_init(void);
uint32_t button_input(void);
void led2_toggle(void);

// main() runs in its own thread in the OS
int main() {
    int interval = 500;
    uint32_t val;

    led2_init();
    button_init();
```

```
    while (true) {
        val = button_input();

        if (val == 0) {
            interval = interval << 1;
            if (interval > 2000)
                interval = 500;
        }

        led2_toggle();

        ThisThread::sleep_for((std::chrono::milliseconds)interval);
    }
}
```