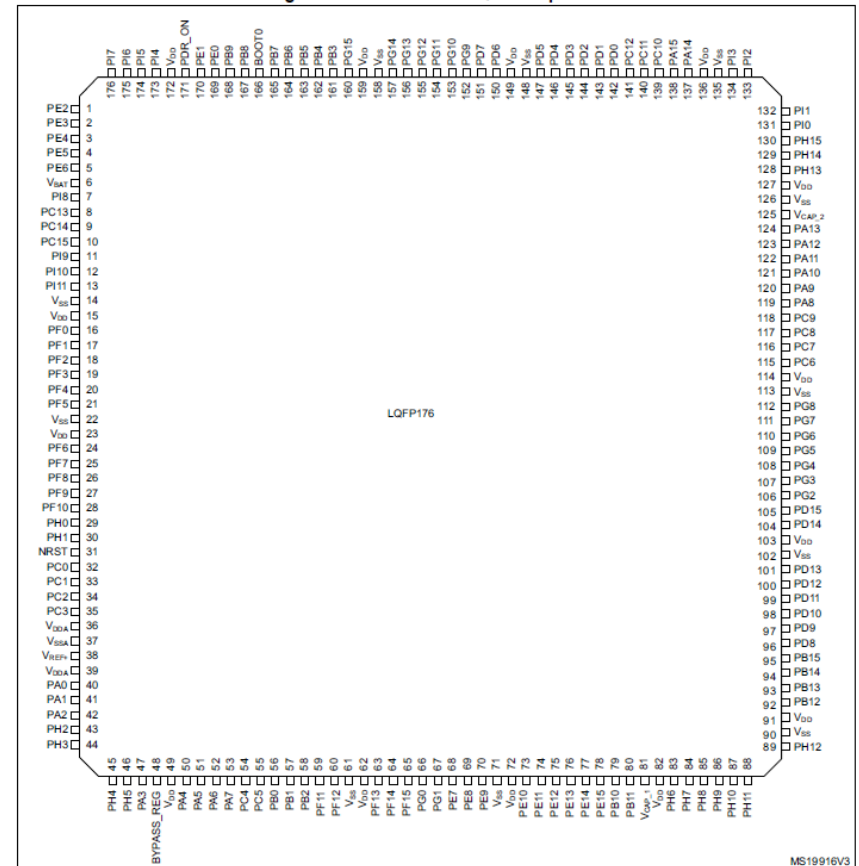


# STM32F407 GPIO (General-Purpose IO)



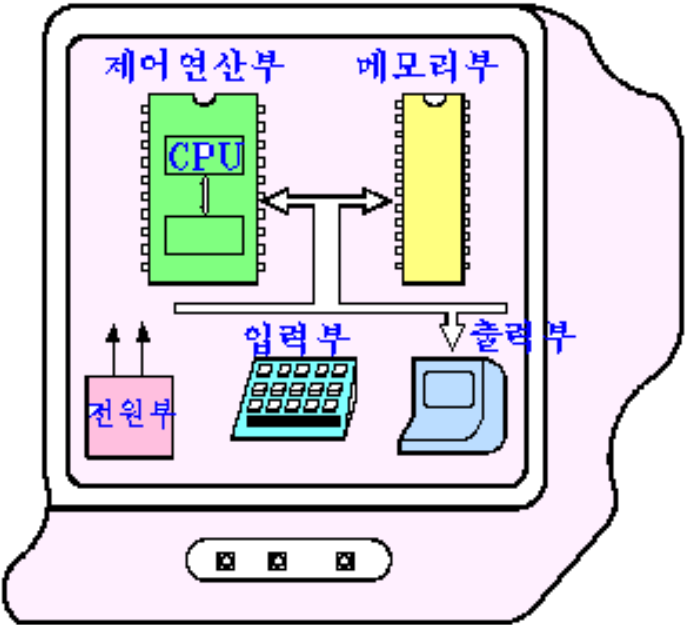
한국산업기술대학교 메카트로닉스공학과  
마이크로컴퓨터구조  
담당교수: 남윤석

## 0. GPIO 란?

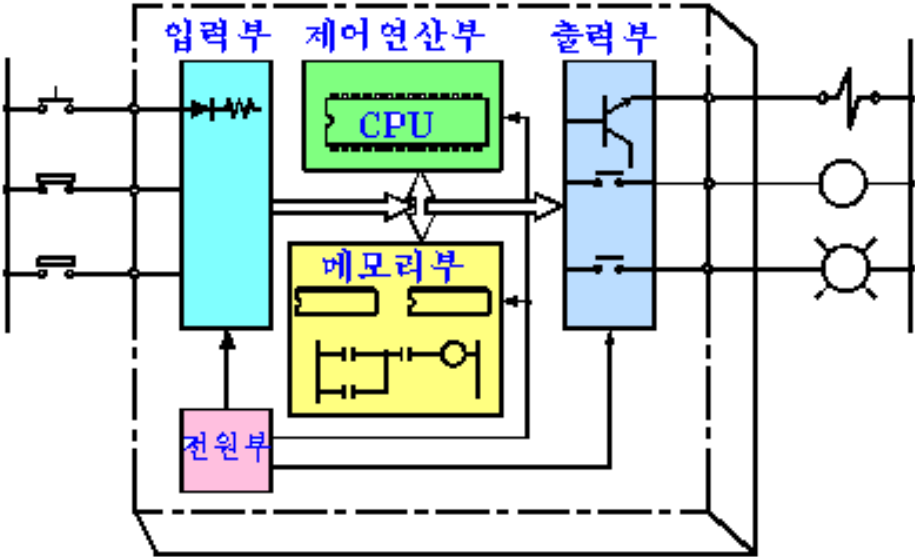
- $\mu$ -processor 나 MCU는 보통 데이터버스를 통해 메모리나 입출력 장치들과 정보를 주고 받음
- 입출력장치(회로)에 따라서는 **단선을 통해 디지털정보(H or L)**를 보내거나 받아야 하는 경우가 있음
- 이 경우를 대비해  $\mu$ -processor 나 MCU는 GPIO 기능을 내장
- GPIO는  $\mu$ -processor 나 MCU의 외부 핀을 통해 'High'나 'Low' 신호를 입출력회로에 보내거나 받는 역할 수행

# 0. GPIO 란?

## ●PLC와 임베디드 컴퓨터(컨트롤러)



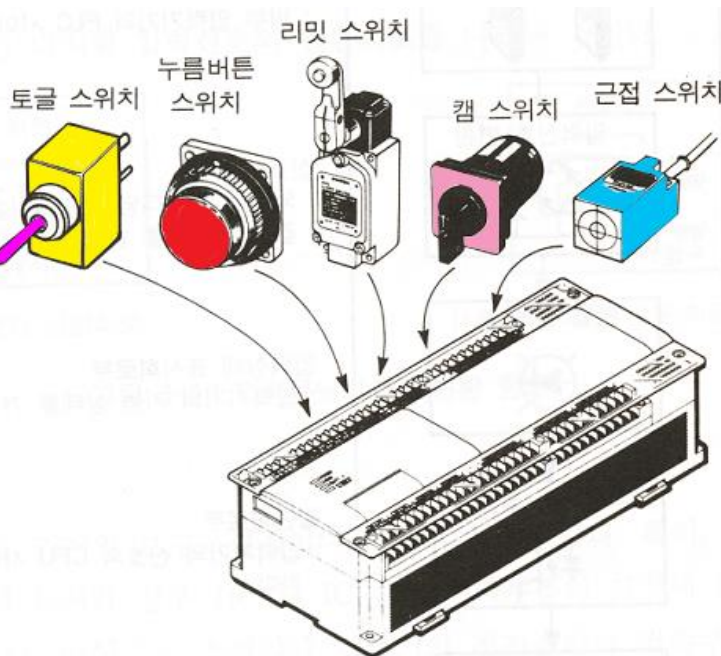
<마이크로 컴퓨터>



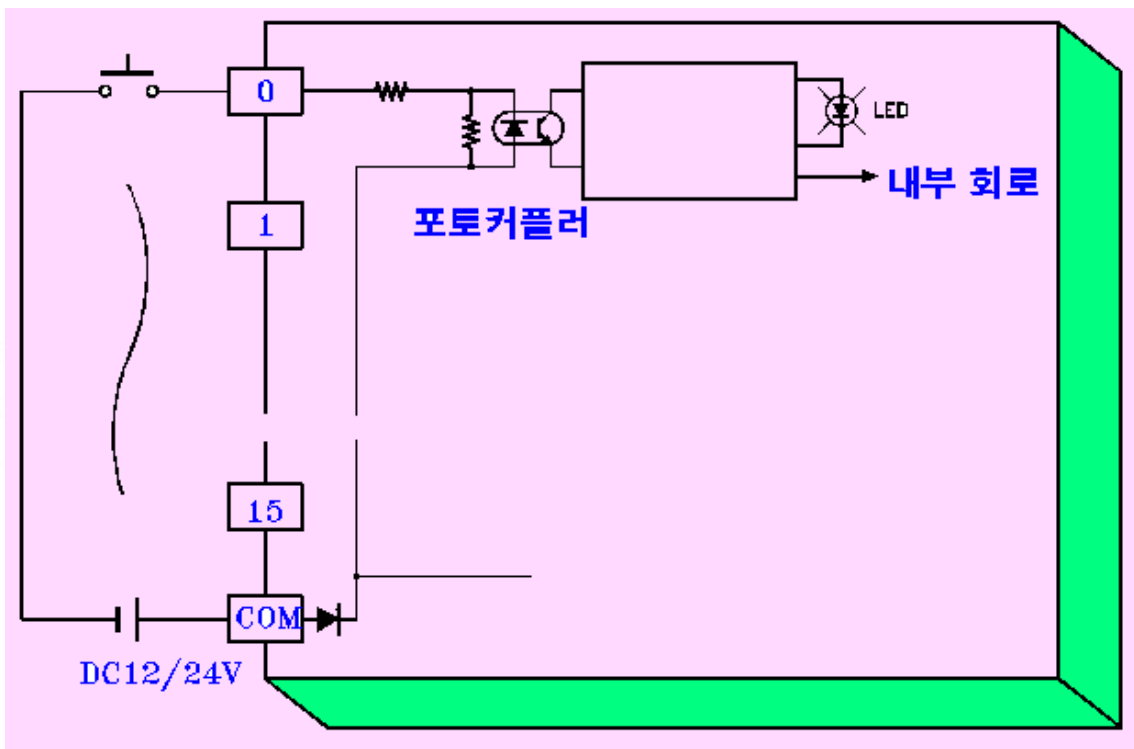
<PLC>

\*PLC제어(남대훈) 제공

●PLC의 I/O 접점 구조



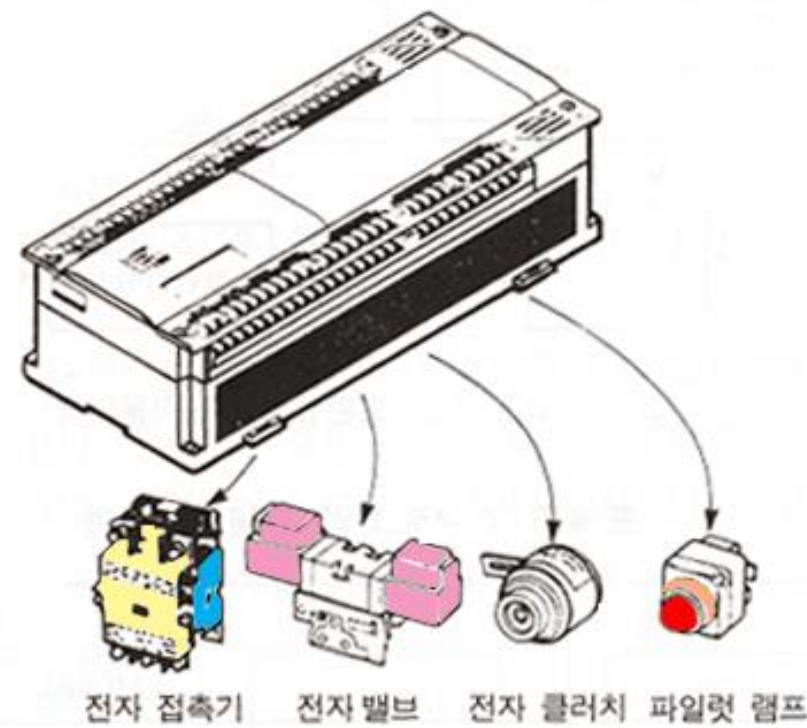
<PLC의 입력기기>



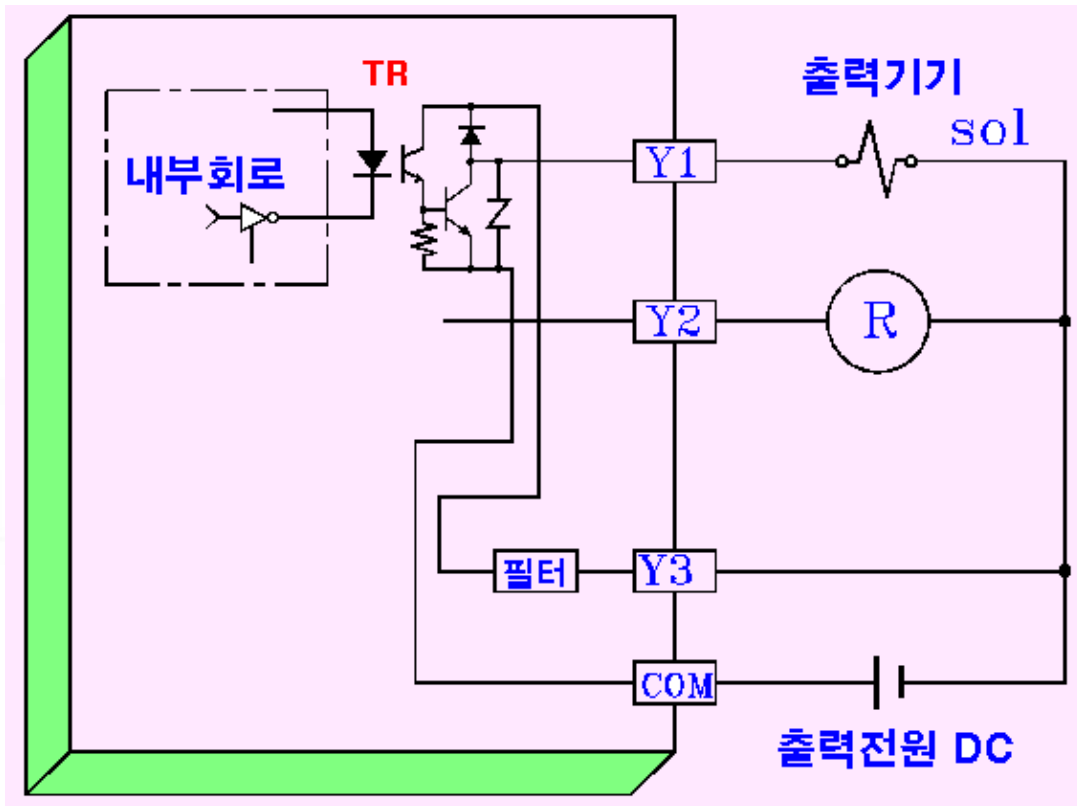
<DC 입력모듈의 내부 회로>

\*PLC제어(남대훈) 제공

●PLC의 I/O 접점 구조



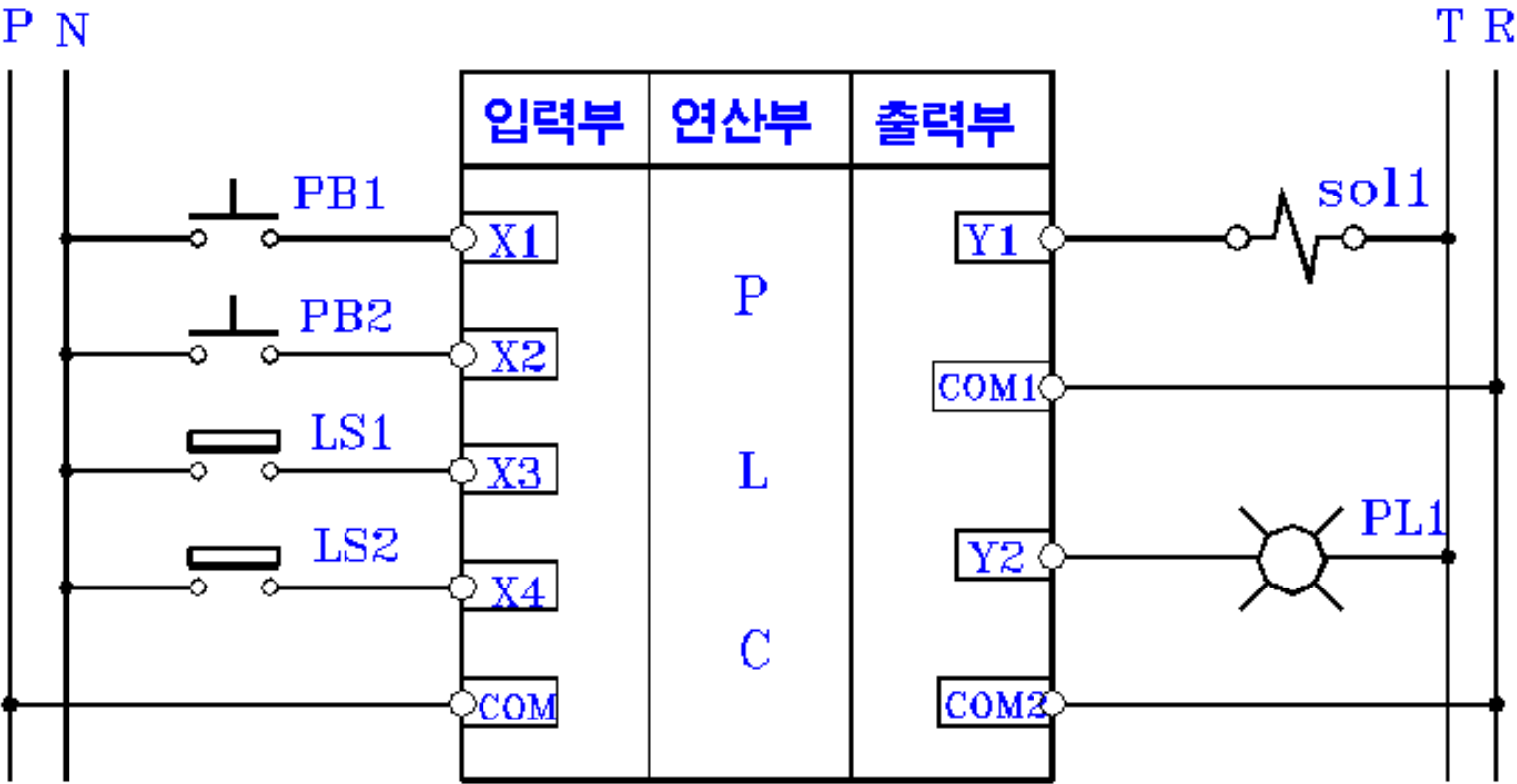
<PLC의 출력 기기>



<트랜지스터 출력 모듈>

\*PLC제어(남대훈) 제공

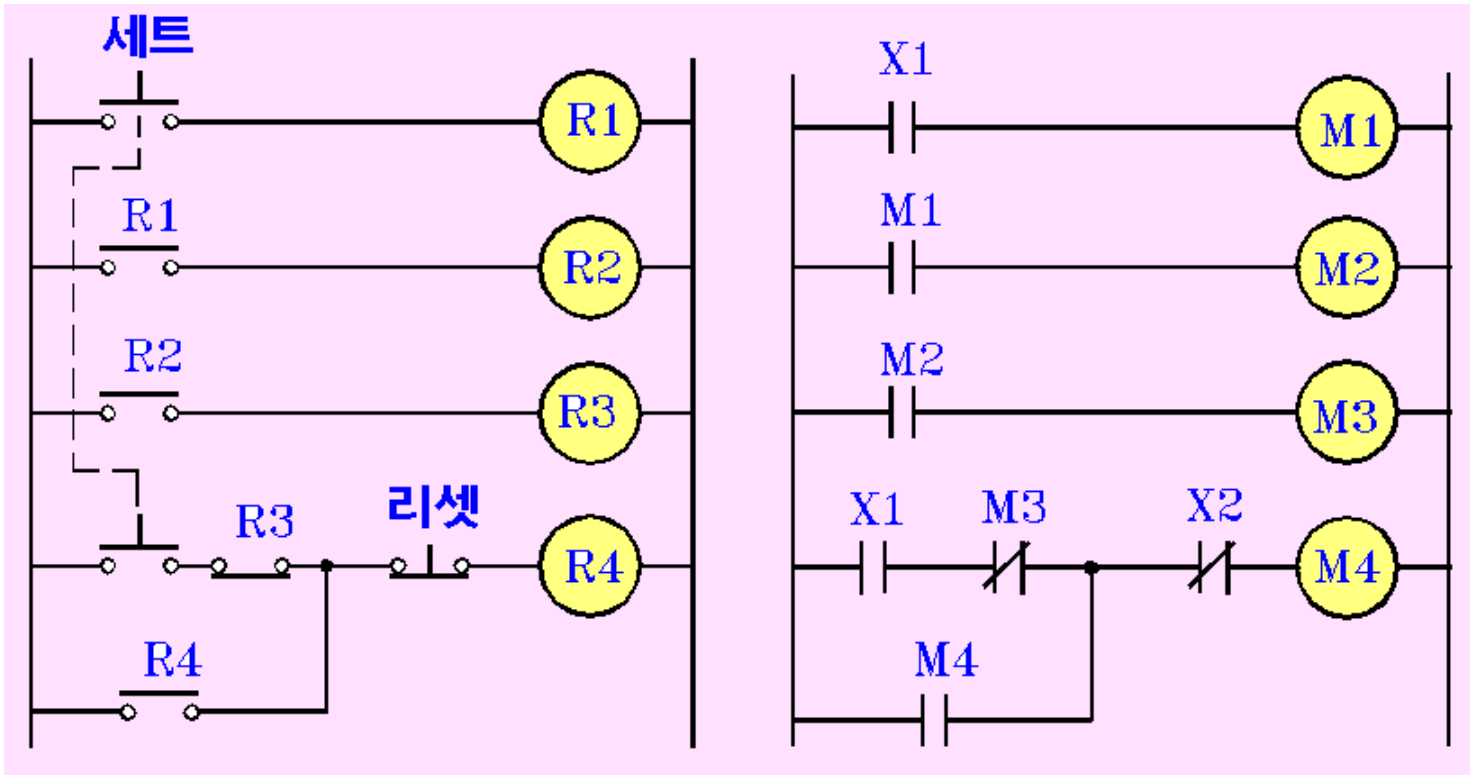
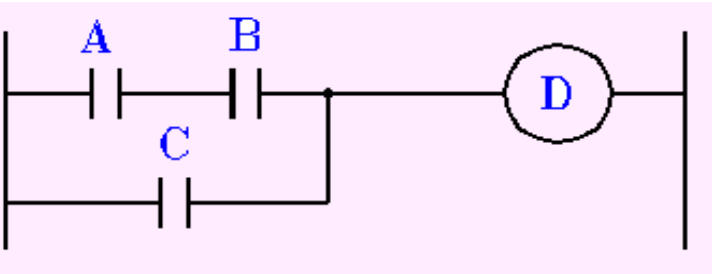
●PLC의 I/O 접점 구조



<PLC 제어배선도(예)>

\*PLC제어(남대훈) 제공

●PLC의 I/O 접점 구조



<PLC 프로그래밍: 래더다이아그램>

\*PLC제어(남대훈) 제공

# 1. STM32F407의 GPIO 응용



OUTPUT

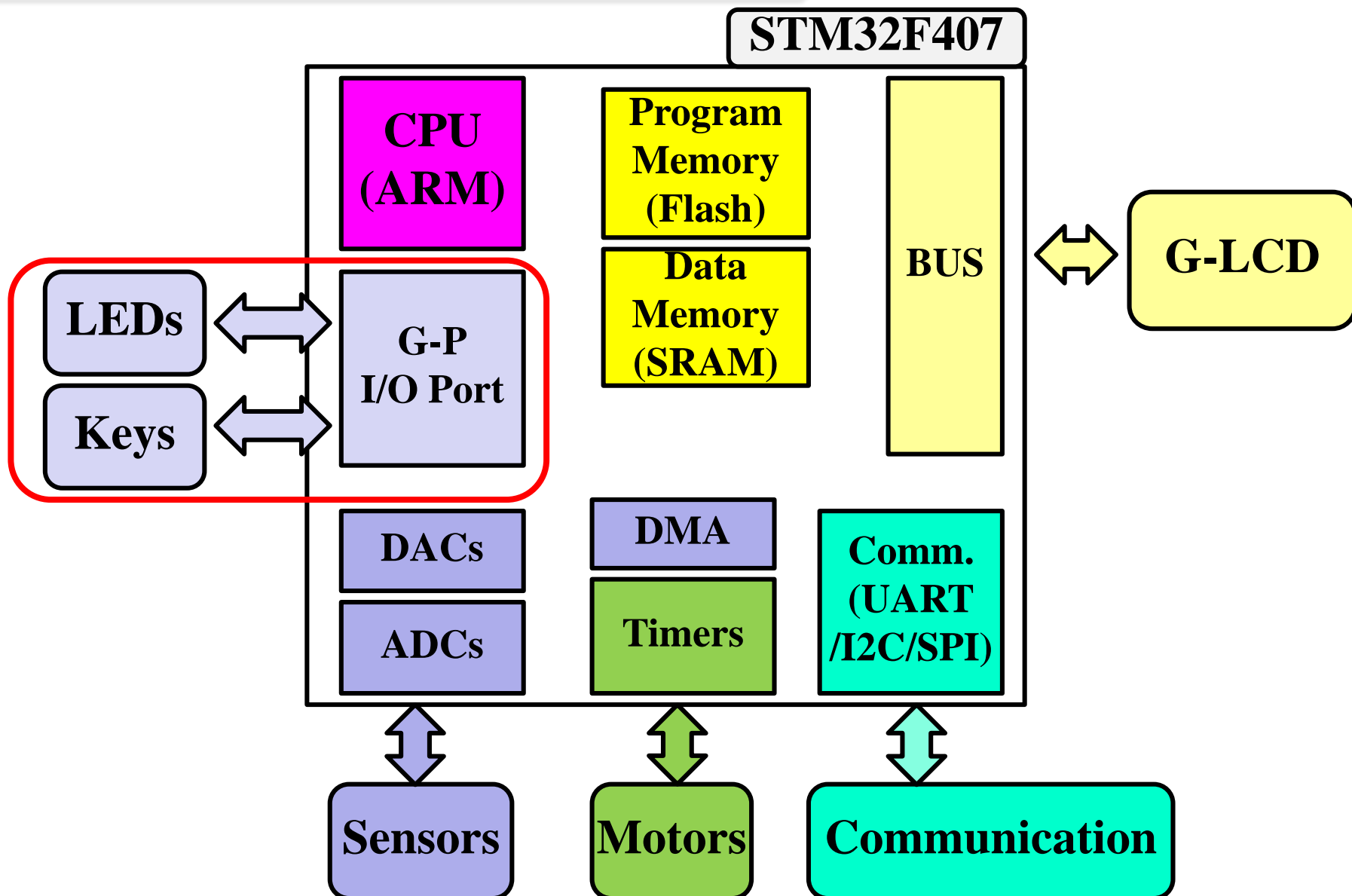


INPUT





## 2.1 STM32F407 GPIO : Block diagram



## 2.2 STM32F407 의 GPIO: 회로

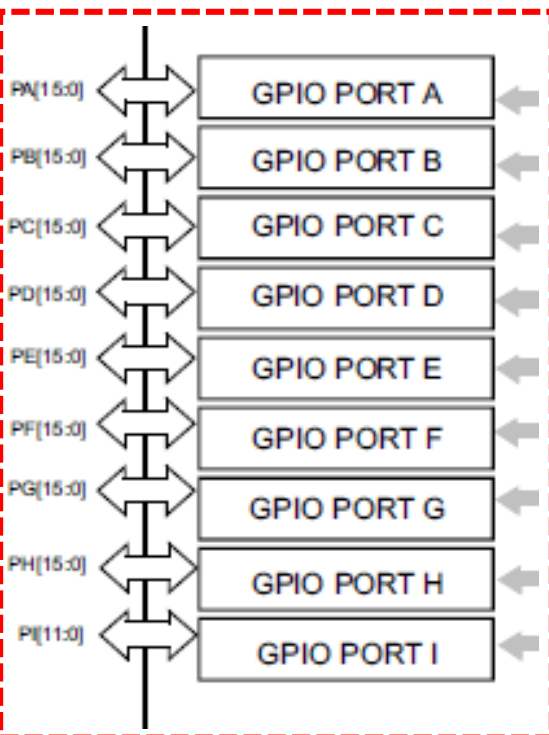
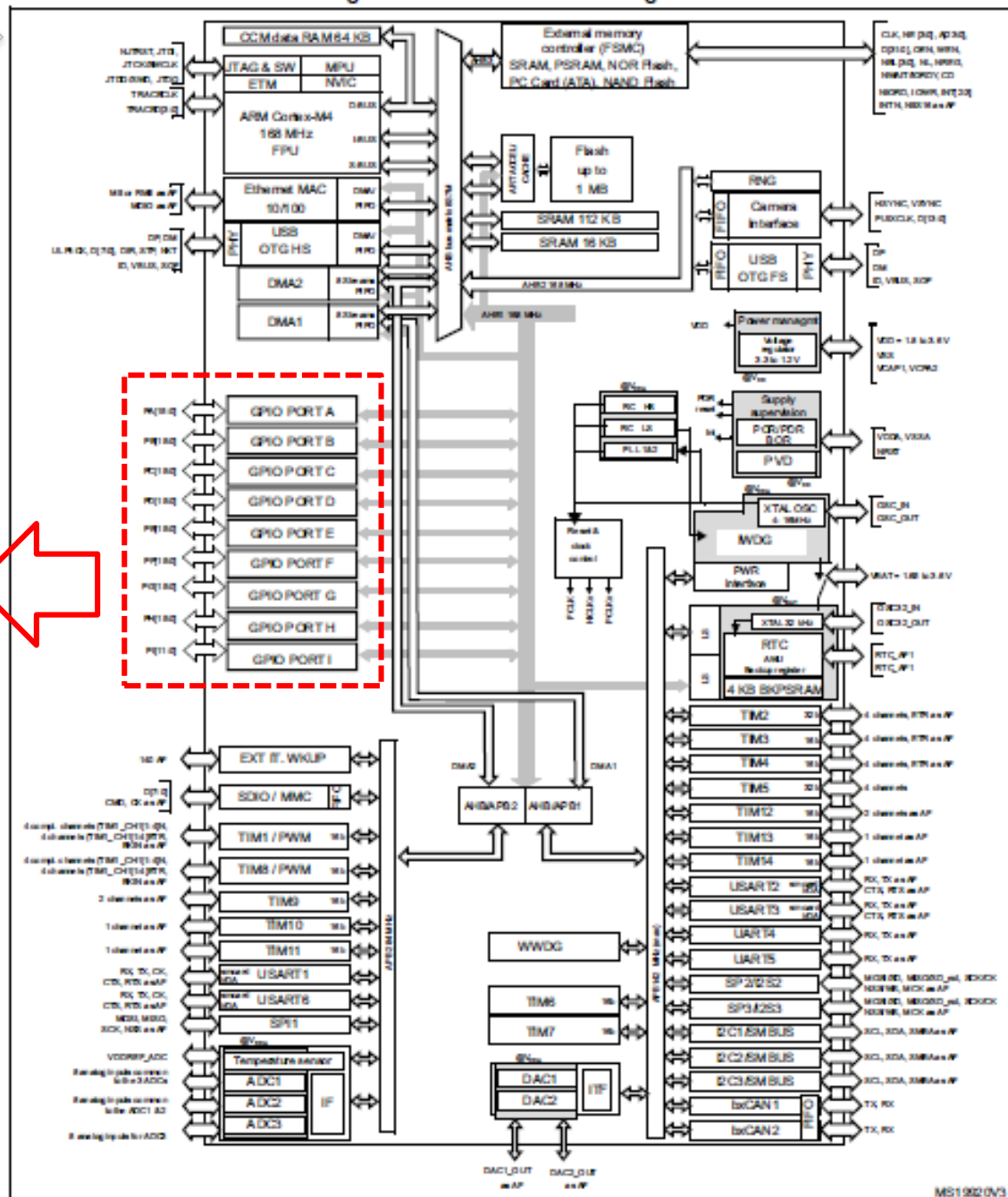
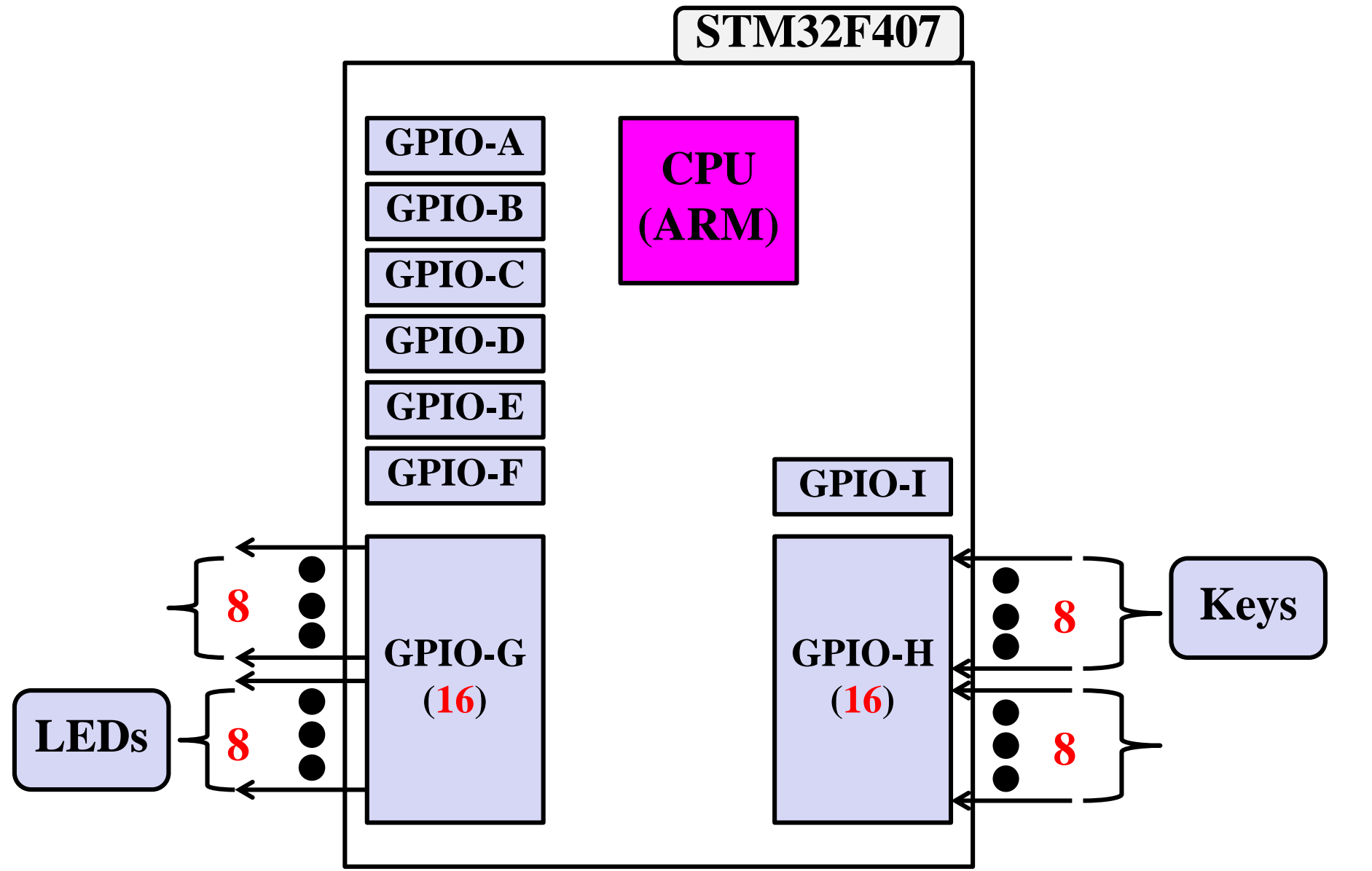


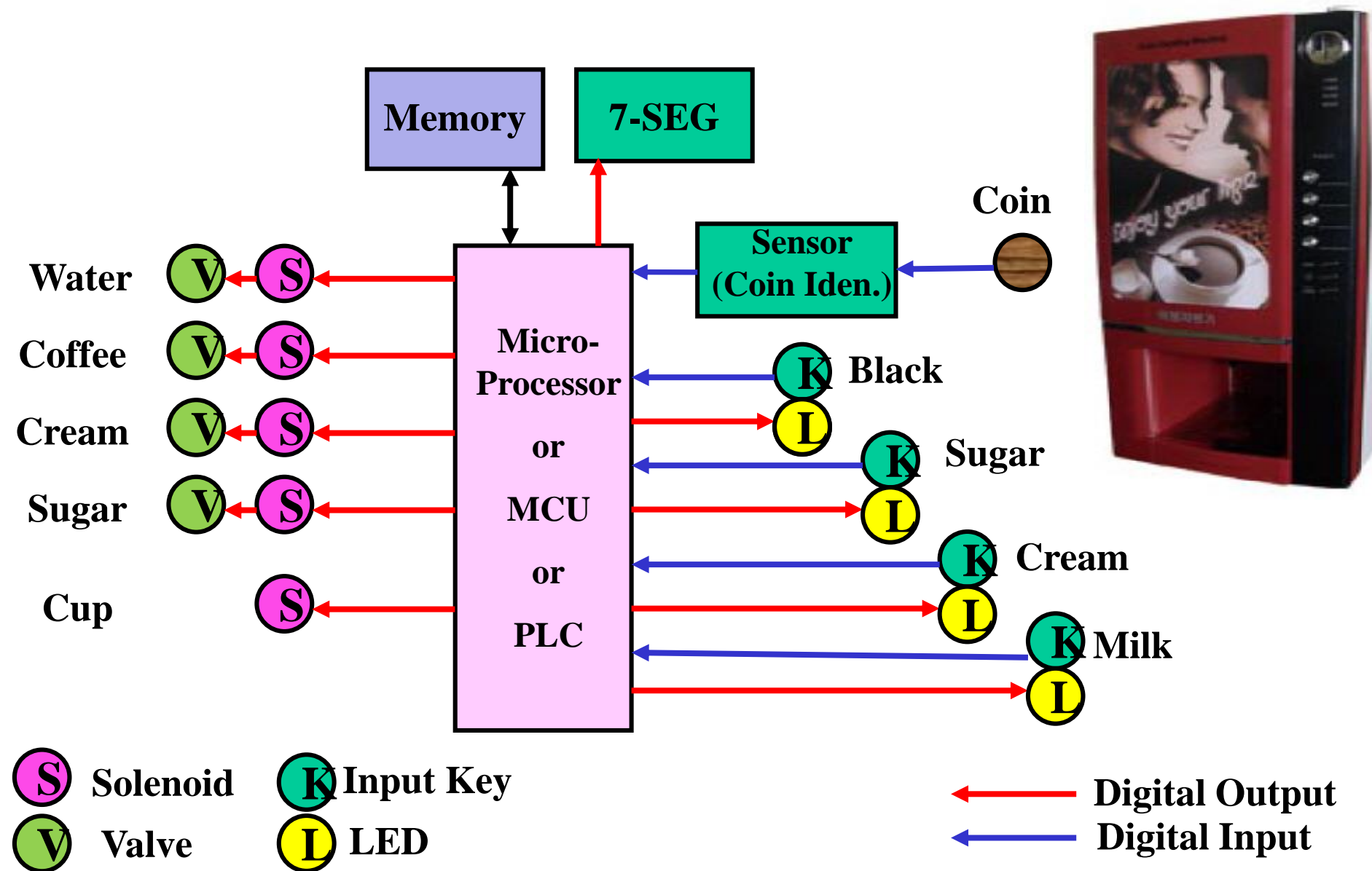
Figure 5. STM32F40x block diagram

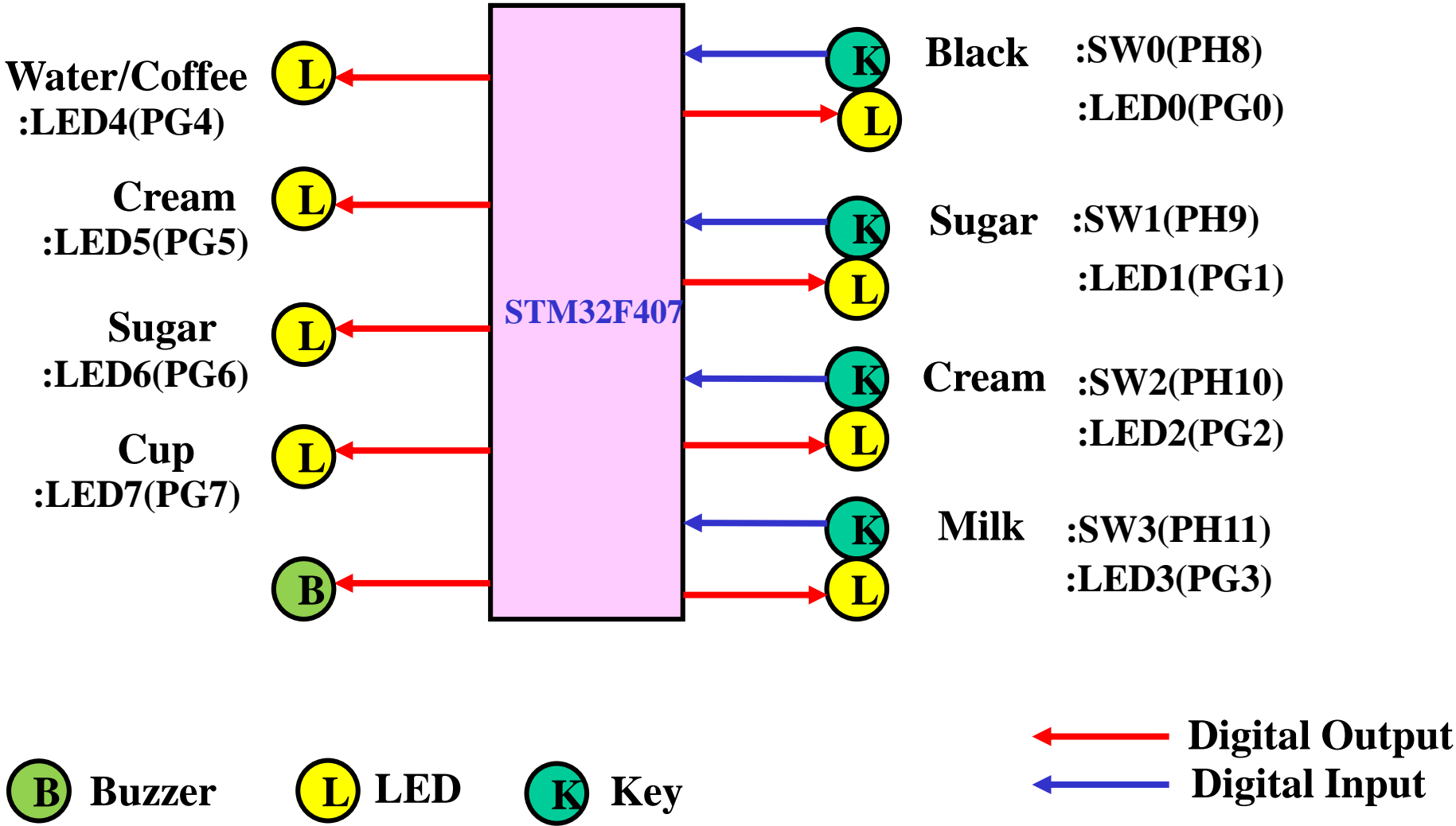


## 2.3 STM32F407 실습보드의 GPIO 회로



# 2.4 STM32F407 실습보드의 GPIO 사용 예: 커피자판기





<마이크로컴퓨터 제어배선도(커피자판기)>

```
Main()
```

```
{
```

```
    ????
```

```
}
```

<마이크로제어 **C-based** 프로그래밍>

### 3. STM32F407 GPIO 주요 특징

1. 9개의 Port (Port A/B/C/D/E/F/G/H/I), 총 140 pins
2. Port 당 16개의 I/O pin 할당(예외: Port I 12pins)
3. Output states : push-pull, open drain (with pull-up/down) 설정 가능
4. Input states : floating, pull-up, pull-down, analog 설정 가능
5. I/O Alternate function(open drain, push-pull) input/output 선택 레지스터 (\*GPIO 기능 이외의 특수기능)
6. Analog function
7. I/O pin마다 속도 설정 가능
8. Bit set/reset 레지스터
9. Locking 레지스터
10. Multiplexer에 의한 GPIO 또는 몇 가지 주변장치 중 하나를 선택 가능 (Multiplexed pin)
11. 모든 핀을 외부 인터럽트 핀으로 사용 가능(\* 해당 핀은 입력모드 설정되어야 함)

## 4. GPIO 기능 설명

### ● Mode Configuration 1

#### ➤ Input modes

- Input floating
- Input pull-up
- Input pull-down
- Analog (Input)

#### ➤ Output modes

- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability

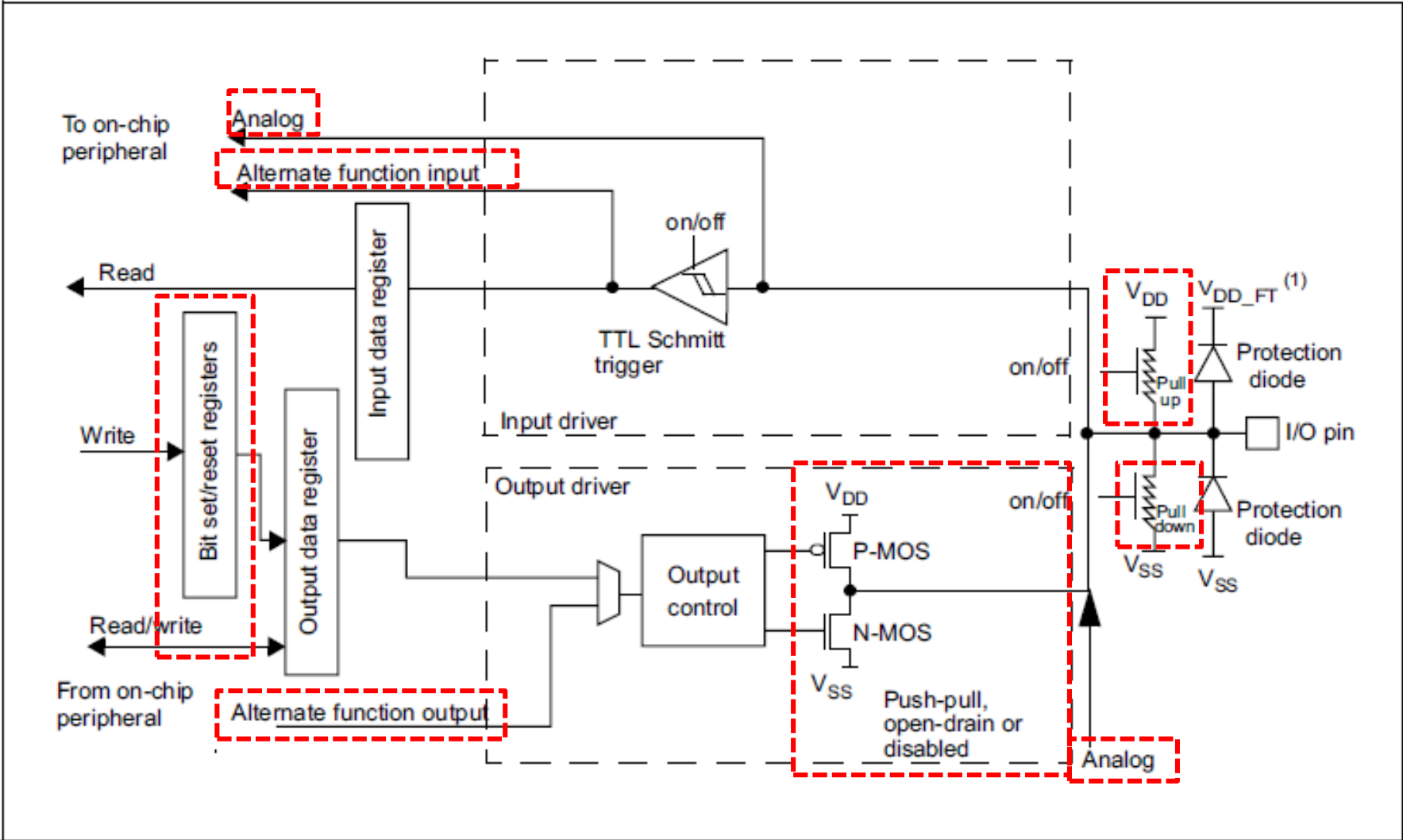
#### ➤ Alternate function modes

- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability



# ● Mode Configuration 1

Basic structure of a five-volt tolerant I/O port bit



## ● Mode Configuration 2

Port bit configuration table

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O configuration	
01	0	SPEED [B:A]		0	0	GP output	PP
	0			0	1	GP output	PP + PU
	0			1	0	GP output	PP + PD
	0			1	1	Reserved	
	1			0	0	GP output	OD
	1			0	1	GP output	OD + PU
	1			1	0	GP output	OD + PD
	1			1	1	Reserved (GP output OD)	
10	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	
00	x	x	x	0	0	Input	Floating
	x	x	x	0	1	Input	PU
	x	x	x	1	0	Input	PD
	x	x	x	1	1	Reserved (input floating)	
11	x	x	x	0	0	Input/output	Analog
	x	x	x	0	1	Reserved	
	x	x	x	1	0		
	x	x	x	1	1		

### • GPIO Pin 설정 레지스터

#### MODER

**00 : Input Mode**  
**01 : Output Mode**  
**10 : Alternate function Mode**  
**11 : Analog Mode**

#### OTYPER

**0 : Output Push-Pull**  
**1 : Output Open-drain**

#### OSPEEDR

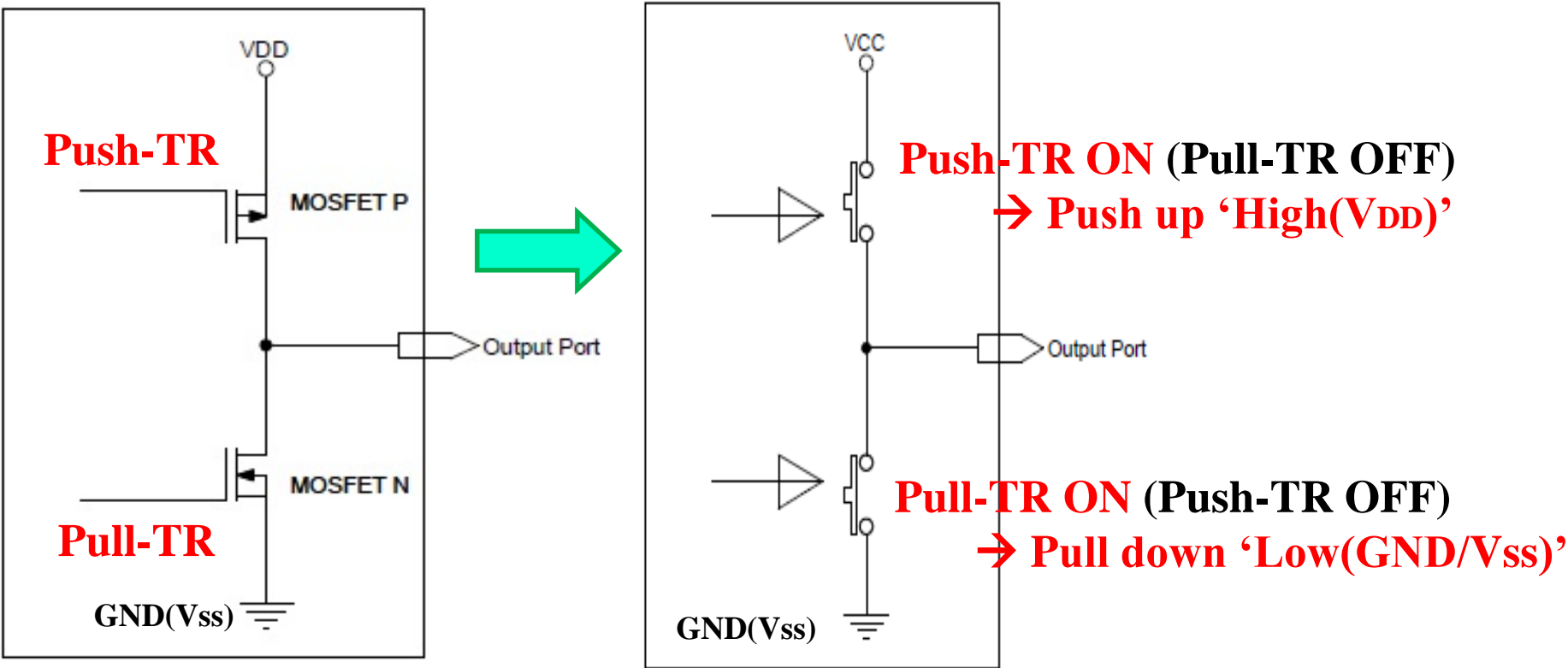
**00 : 2MHz Low speed**  
**01 : 25MHz Medium speed**  
**10 : 50MHz Fast speed**  
**11 : 100MHz High speed**

#### PUPDR

**00 : No pull-up/down**  
**01 : Pull-up**  
**10 : Pull-down**  
**11 : Reserved**

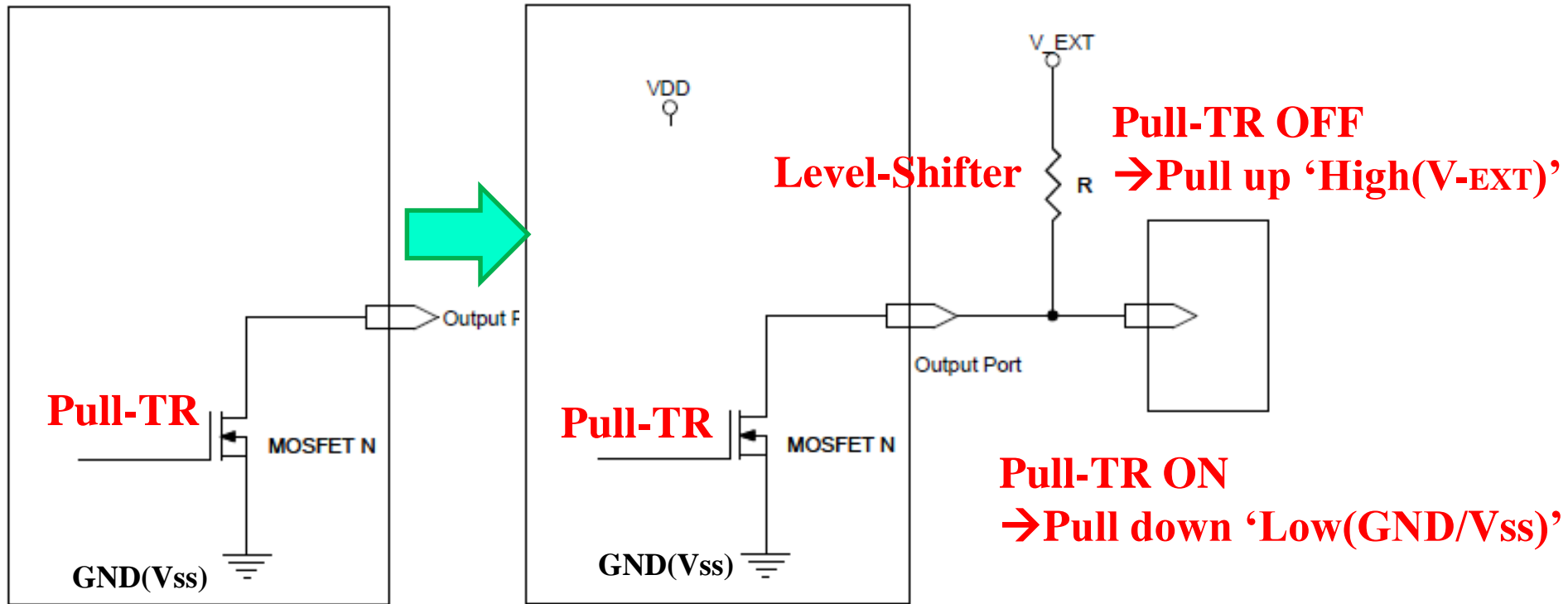
# ● Push-Pull, Open-Drive, Pull up/down -1

## • Push-Pull 출력의 구조 및 동작 원리



## ● Push-Pull, Open-Drive, Pull up/down -2

### •Open-Drive 출력의 구조 및 동작 원리



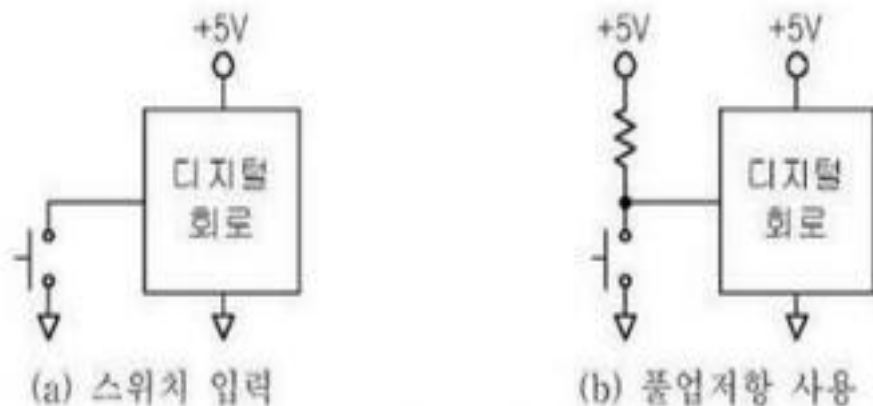
### •Open\_Drain 출력회로를 사용하는 이유

- Level-Shifter
- 출력 전류를 회로에 맞게 설계가능

•Level-Shifter: 전압기준 레벨을 VDD에서 V-EXT로 변환  
(예: VDD=3.3V, V\_EXT=5V, Box안 회로의 High는 3.3V,  
V\_EXT에 묶여있는 회로의 High는 5V임)

## ● Push-Pull, Open-Drive, Pull up/down -3

### • Pull up 회로의 구조와 동작원리

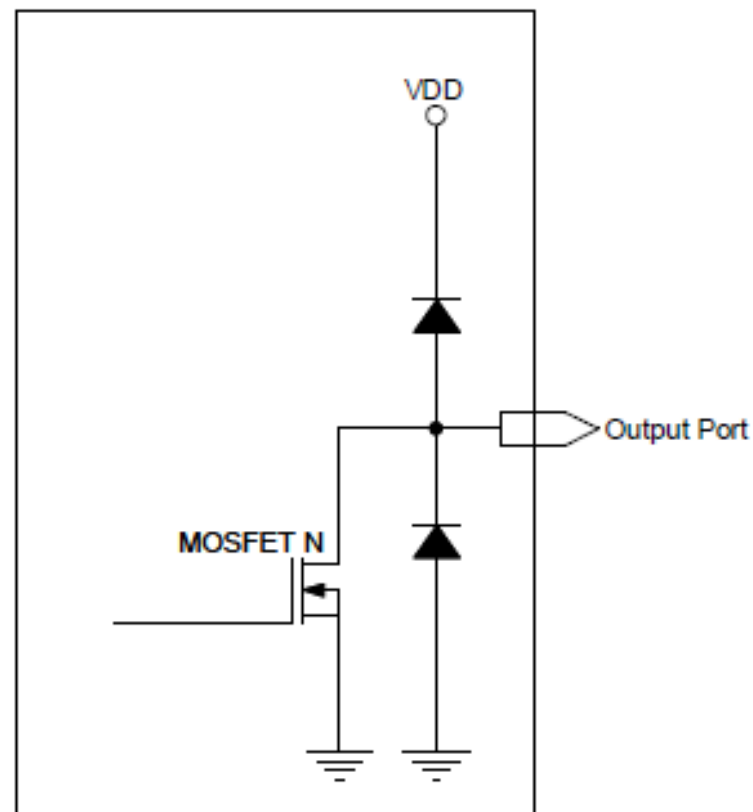


<그림 1> L 스위치 입력과 풀업 저항

스위치	ON	OFF
(a)그림	0V(Low)	Floating
(b)그림	0V(Low)	+5V(High)

• **Pull down** : 생략(저항이 **GND**에 연결되어 있음)

### • 정전기 보호 회로



-Diode의 역할: VDD보다 크거나, Vss(GND) 보다 작은 (음전압) 크기의 정전기 발생 시 우회회로 제공하는 정전기 보호 회로

\*출처:Daum

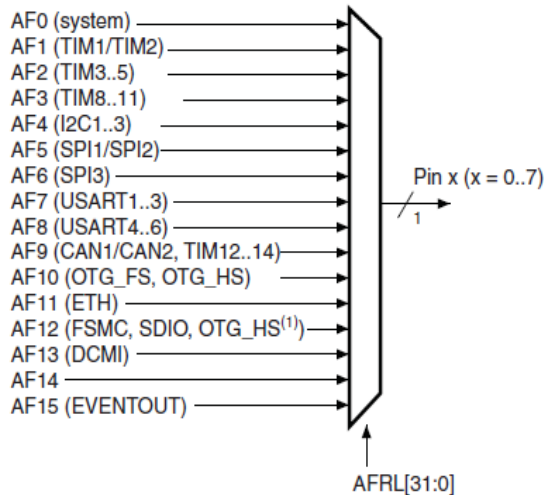
## ● 기타 GPIO 기능

- **Reset 직후 모든 핀들의 디폴트 모드: 'floating input' 모드**
  - \*JTAG(다운로드 및 디버깅에 사용되는 기능) 핀 제외
- **입출력 상태에 따른 핀 동작**
  - 입력 설정 경우:Input data register는 매 APB2 클럭(72MHz)마다 I/O 핀에서 데이터를 입력 받음
  - 출력 설정 경우:Output data register에 저장된 값이 I/O 핀에 출력
- **대체기능(Alternate function) 전환 방법**
  - Port Bit Configuration Register(GPIOx\_AFRL/AFRH/MODER/OTYPER/OSPEEDR/PUPDR) 설정 필요
- **대체기능 입력 전환 조건:** 해당 포트를 입력 모드(floating, pull-up/down)로 설정
- **대체 기능 출력 전환 조건:** 해당 포트의 비트를 대체기능 출력 모드(push-pull 또는 open-drain)로 설정
- **양방향(입출력)의 대체 기능 전환 조건:** 해당 포트의 비트를 대체기능 출력 모드(push-pull 또는 open-drain)로 설정(이 경우 입력 드라이버는 플로팅 입력 모드(input floating mode)로 설정됨)

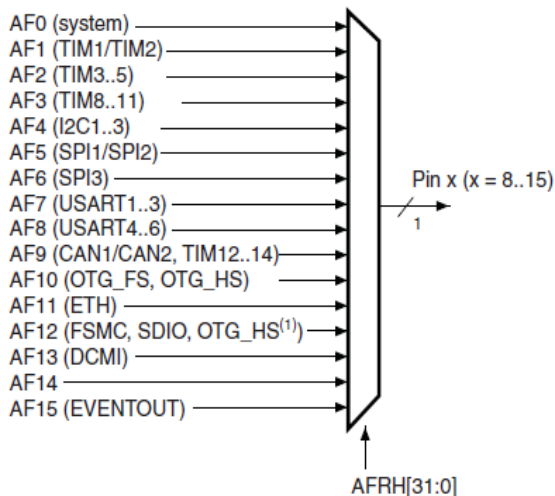
# I/O pin multiplexer and mapping 1

## Selecting an alternate function on STM32F405xx/07xx and STM32F415xx/17xx

For pins 0 to 7, the GPIOx\_AFRL[31:0] register selects the dedicated alternate function



For pins 8 to 15, the GPIOx\_AFRH[31:0] register selects the dedicated alternate function



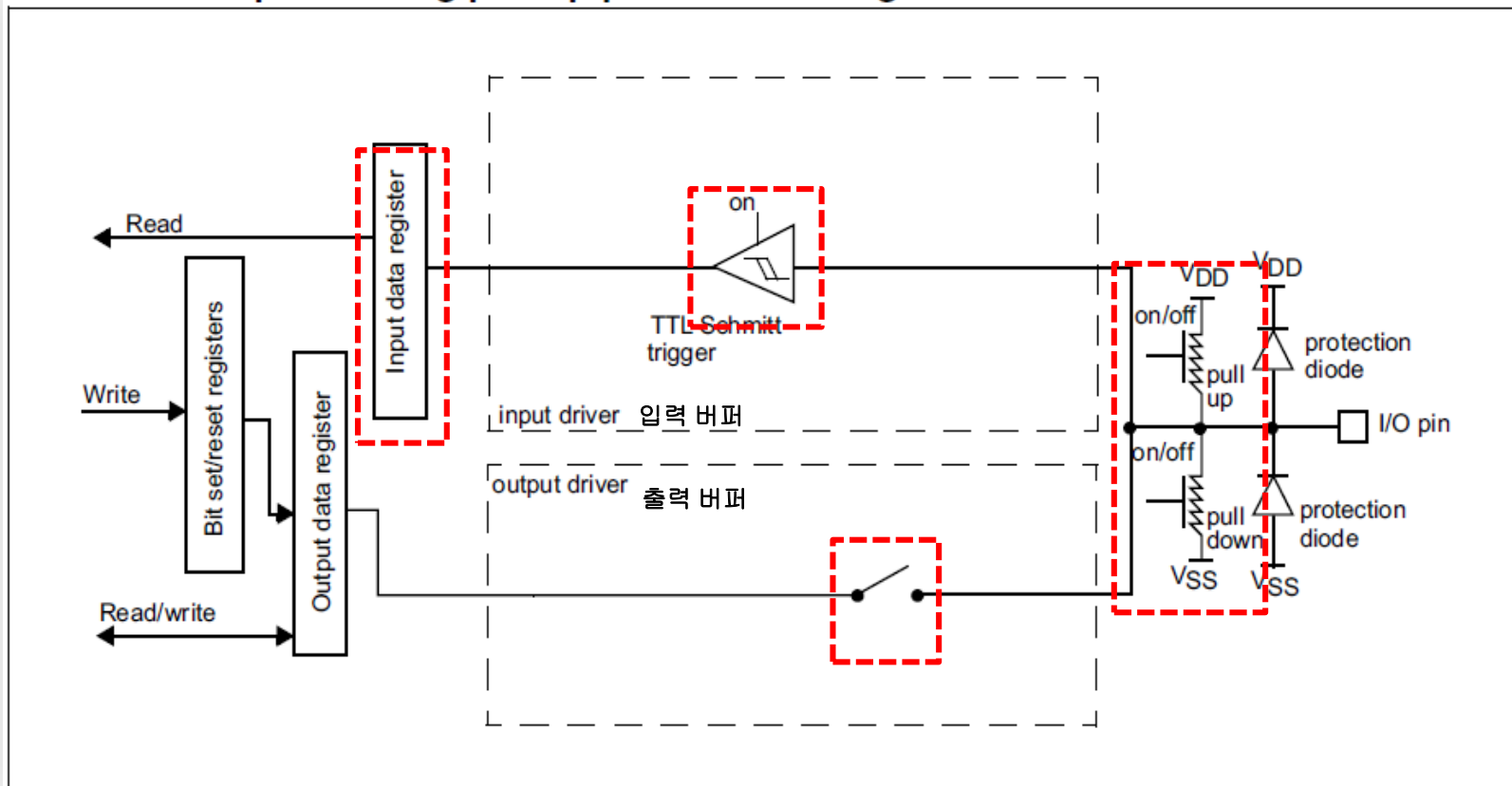
STM32F40x pin and ball definitions (continued)

Pin	Pin name (function after reset) <sup>(1)</sup>	Pintype	I/O structure	Notes	Alternate functions	Additional functions
UFBGA176	LQFP176					
H3	17	PF1	I/O	FT	FSMC_A1 / I2C2_SCL / EVENTOUT	
H2	18	PF2	I/O	FT	FSMC_A2 / I2C2_SMBA / EVENTOUT	
J2	19	PF3	I/O	FT	FSMC_A3/EVENTOUT	ADC3_IN9
J3	20	PF4	I/O	FT	FSMC_A4/EVENTOUT	ADC3_IN14
K3	21	PF5	I/O	FT	FSMC_A5/EVENTOUT	ADC3_IN15
G2	22	V <sub>SS</sub>	S			
G3	23	V <sub>DD</sub>	S			
K2	24	PF6	I/O	FT	TIM10_CH1 / FSMC_NIORD / EVENTOUT	ADC3_IN4
K1	25	PF7	I/O	FT	TIM11_CH1/FSMC_NREG / EVENTOUT	ADC3_IN5
L3	26	PF8	I/O	FT	TIM13_CH1 / FSMC_NIOWR / EVENTOUT	ADC3_IN6
L2	27	PF9	I/O	FT	TIM14_CH1 / FSMC_CD / EVENTOUT	ADC3_IN7
L1	28	PF10	I/O	FT	FSMC_INTR / EVENTOUT	ADC3_IN8
G1	29	PH0/OSC_IN (PH0)	I/O	FT	EVENTOUT	OSC_IN <sup>(4)</sup>
H1	30	PH1/OSC_OUT (PH1)	I/O	FT	EVENTOUT	OSC_OUT <sup>(4)</sup>
J1	31	NRST	I/O	RS T		
M2	32	PC0	I/O	FT	OTG_HS_ULPI_STP / EVENTOUT	ADC123_IN10
M3	33	PC1	I/O	FT	ETH_MDC / EVENTOUT	ADC123_IN11
M4	34	PC2	I/O	FT	SPI2_MISO / OTG_HS_ULPI_DIR / ETH_MII_TXD2 / I2S2ext_SD / EVENTOUT	ADC123_IN12

\* ADC / DAC 는 Alternate function에 해당 되지 않음.

# ● Input Configuration

Input floating/pull up/pull down configurations

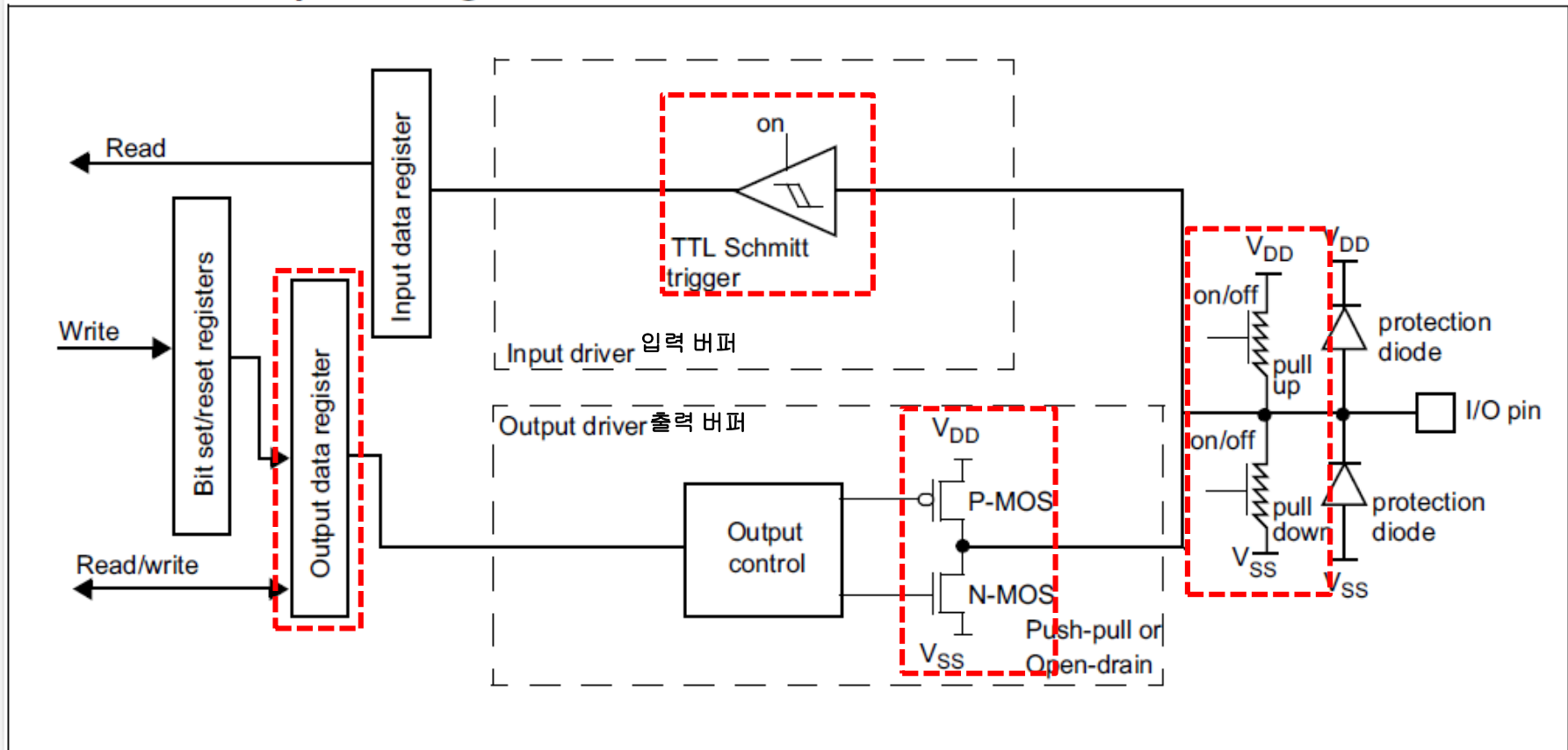


- 입력 으로 설정될 때 H/W 상태
- 출력 버퍼 비활성화 및 슈미트 트리거 입력 활성화
- GPIOx\_PUPDR 레지스터에 의해 풀업 풀다운의 상태 결정
- AHB1의 클럭에 동기되어 I/O 핀의 입력 값이 갱신
- Input data register를 read하면 I/O state를 획득 가능



# ● Output Configuration -1

## Output configuration

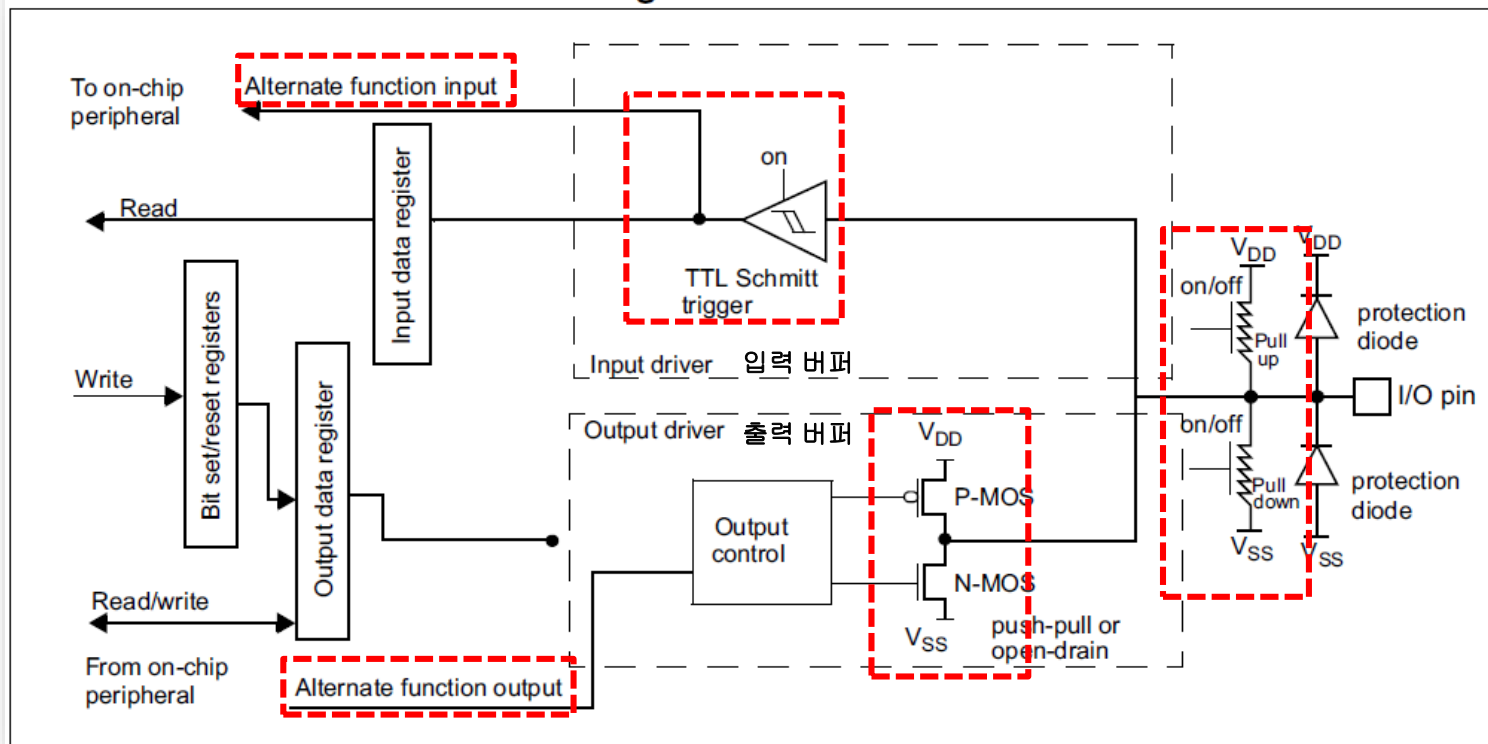


## ● Output Configuration -2

- 출력으로 설정될 때 H/W 상태
- 출력 버퍼 활성화
  - Open-Drain mode : Output register에 '0'을 기록하면 N-MOS가 활성화 되고, '1'을 기록하면 Hi-Z 상태가 됨
  - Push-pull mode : Output register에 '0'을 기록하면 N-MOS가 활성화 되고, '1'을 기록하면 P-MOS가 활성화 됨
- 슈미트 트리거 입력 활성화
- 내부 Pull-up / Pull-down 저항 활성화(GPIOx\_PUPDR 설정과 무관)
- AHB1의 클럭에 동기되어 I/O 핀의 입력 값이 갱신됨
- Input data register를 read하면 I/O state를 획득 가능
- Output data register를 read하면 최근의 written value 획득 가능

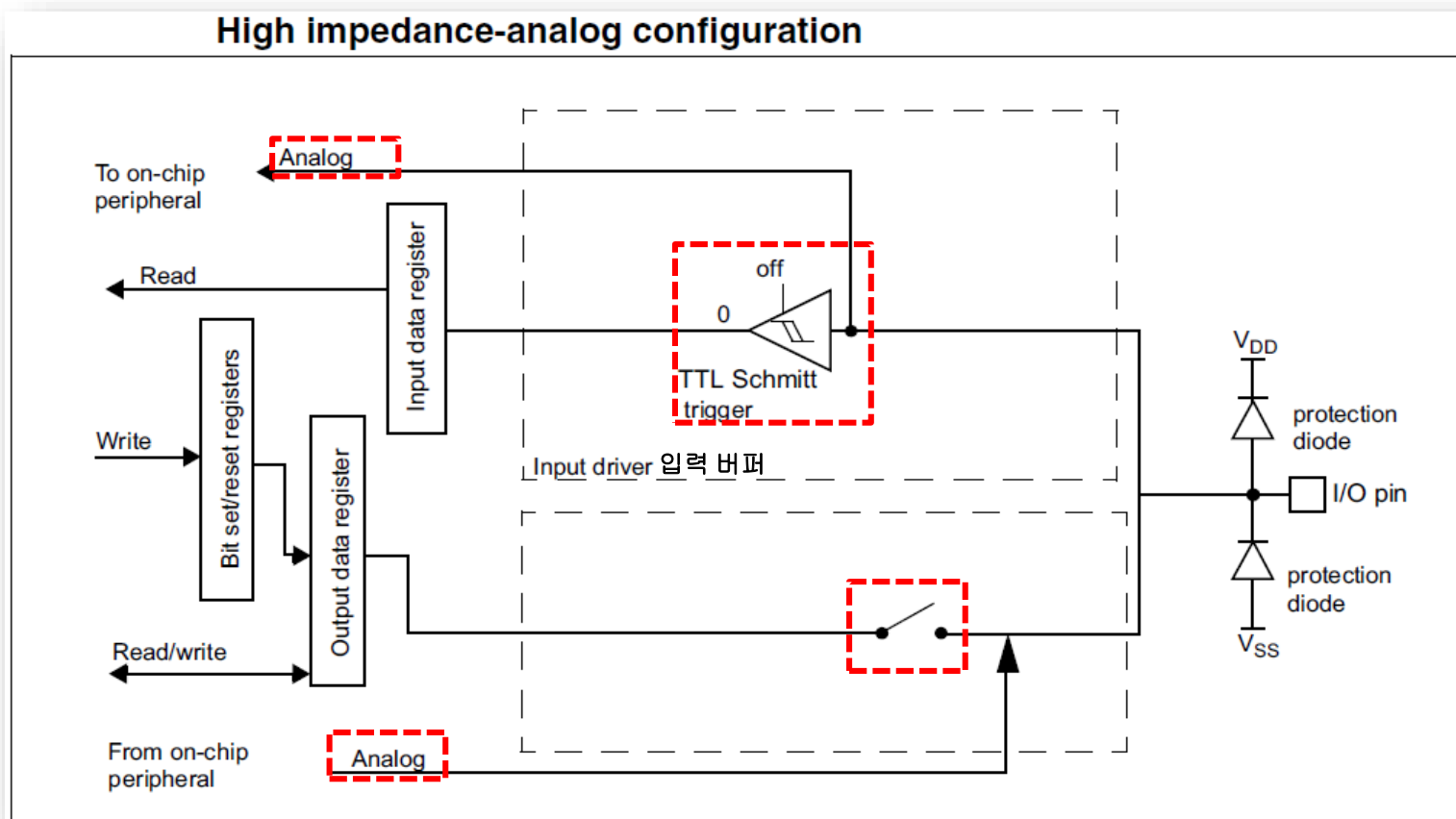
# ● Alternate function Configuration

Alternate function configuration



- 대체(부속) 기능으로 설정 될 때 H/W 상태
- 출력 버퍼는 Open-drain 혹은 Push-pull로 설정 가능
- 출력 버퍼는 주변장치에 의해 구동
- 슈미트 트리거 입력 활성화
- 내부 Pull-up / Pull-down 저항 활성화(GPIOx\_PUPDR 설정과 무관)
- AHB1의 클럭에 동기되어 I/O 핀의 입력 값이 갱신됨

## ● Analog Configuration



- 아날로그로 설정 될 때 H/W 상태
- 출력 버퍼 비활성화
- 슈미트 트리거 입력 비 활성화 (슈미트 트리거 의 출력은 '0' 설정)
- 풀업 풀다운 설정 금지
- Input data register를 읽을 경우 무조건 '0'을 얻음

## 5. GPIO 주요 레지스터

### ● GPIOx\_MODER

: GPIO port mode register (GPIOx\_MODER) (x = A..I)

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

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

• Bits 2y:2y+1 MODERy[1:0]: Port x configuration bits (y = 0..15)

I/O의 Input 이나 Output 방향을 설정하는 레지스터

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

# ● GPIOx\_OTYPER

: GPIO port output type register (GPIOx\_OTYPER) (x = A..I)

Reset value: 0x0000 0000

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
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

- Bits 31:16 Reserved, must be kept at reset value.
- Bits 15:0 OTy[1:0]: Port x configuration bits (y = 0..15)  
 I/O port의 output type을 설정해주는 비트들임  
 0: Output push-pull (reset state)  
 1: Output open-drain

## ● GPIOx\_OSPEEDR

**: GPIO port output speed register (GPIOx\_OSPEEDR)(x = A..I)**

**Reset values:**

- **0x0000 00C0 for port B**
- **0x0000 0000 for other ports**

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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
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]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

**• Bits 2y:2y+1 OSPEEDRy[1:0]: Port x configuration bits (y = 0..15)**

**I/O output speed를 설정하는 비트들임**

**00: 2 MHz Low speed**

**01: 25 MHz Medium speed**

**10: 50 MHz Fast speed**

**11: 100 MHz High speed on 30 pF (80 MHz Output max speed on 15 pF)**

## ● GPIOx\_PUPDR

: GPIO port pull-up/pull-down register (GPIOx\_PUPDR)(x = A..I)

Reset values:

- 0x6400 0000 for port A
- 0x0000 0100 for port B
- 0x0000 0000 for other ports

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	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
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	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

• Bits 2y:2y+1 PUPDRy[1:0]: Port x configuration bits (y = 0..15)

I/O Port Pin에 pull-up 이나 pull-down으로 설정하는 비트들임

00: No pull-up, pull-down

01: Pull-up

10: Pull-down

11: Reserved



## ● GPIOx\_IDR

: GPIO port input data register (GPIOx\_IDR)(x = A..I)

Reset value: 0x0000 XXXX (where X means undefined)

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
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

- Bits 31:16 Reserved, must be kept at reset value.
  - Bits 15:0 IDRx[15:0]: Port input data (y = 0..15)
- 이 비트들은 읽기모드만 가능하고 word단위의 접근만 허용되며 I/O port에 대응하는 입력 값을 가지고 있음

## ● GPIOx\_ODR

**: GPIO port output data register (GPIOx\_ODR)(x = A..I)**

**Reset value: 0x0000 0000**

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

- Bits 31:16 Reserved, must be kept at reset value
- Bits 15:0 ODRy[15:0]: Port output data (y = 0..15)  
이 비트들은 읽기 쓰기가 가능, 또한 GPIOx\_BSRR(x = A..I) 레지스터를 통해 ODRy 비트들을 개별적으로 set 또는 reset 가능

# ● GPIOx\_BSRR

: GPIO port bit set/reset register (GPIOx\_BSRR)(x = A..I)

Reset value: 0x0000 0000

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

## • Bits 31:16 BRy: Port x reset bit y (y = 0..15)

이 비트들은 오직 쓰기만 가능, word, half-word or byte 단위로의 접근이 가능  
이 비트들을 읽을 경우 0x0000이 리턴됨

0: ODRx 비트에 대응하는 값이 아무런 변화가 없음

1: ODRx 비트에 대응하는 값이 Reset 됨

-Note: 만약 BSx와 BRx가 둘다 set될 경우, BSx가 우선

## • Bits 15:0 BSy: Port x set bit y (y= 0..15)

이 비트들은 오직 쓰기만 가능, word, half-word or byte 단위로의 접근이 가능  
이 비트들을 읽을 경우 0x0000이 리턴됨

0: ODRx 비트에 대응하는 값이 아무런 변화가 없음

1: ODRx 비트에 대응하는 값이 Set 됨

## ● GPIOx\_AFRL(AFR[0])

: GPIO alternate function low register (x = A..I)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
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
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

0000: AF0

0001: AF1

0010: AF2

0011: AF3

0100: AF4

0101: AF5

0110: AF6

0111: AF7

1000: AF8

1001: AF9

1010: AF10

1011: AF11

1100: AF12

1101: AF13

1110: AF14

1111: AF15

## ● GPIOx\_AFRH(AFR[1])

: GPIO alternate function high register (x = A..I)

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]			
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
AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

0000: AF0

0001: AF1

0010: AF2

0011: AF3

0100: AF4

0101: AF5

0110: AF6

0111: AF7

1000: AF8

1001: AF9

1010: AF10

1011: AF11

1100: AF12

1101: AF13

1110: AF14

1111: AF15

## 6. STM32F407의 GPIO 프로그래밍 실습

## 6.0 프로그램에서의 GPIOx set-up 과정 및 레지스터 설정

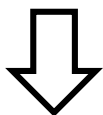
### RCC 설정

- **RCC→CR,CFGR,PLLCFGR (Clock소스/주파수 설정)**
- **RCC→AHB1ENR(GPIO Clock Enable)**



### GPIO Mode 설정

- **GPIOx→MODER (GPIO output/input 설정)**
- **GPIOx→OSPEEDR (Output speed 설정)**
- **GPIOx→OTYPER (Output type(P-P/O-D) 설정)**
- **GPIOx→PUPDR (Pull-up/Pull-down 설정)**



### GPIO 실행

- **출력: GPIOx→ODR, BSRR**
- **입력: GPIOx→IDR**

## 6.0 Reset 직후 레지스터 초기(디폴트) 상태

- **GPIO<sub>x</sub>→MODER :**
  - 0xA800 0000 for port A(PA.15~13: Alternate function, others:Input)
  - 0x0000 0280 for port B(PB.4/3: Alternate function, others: Input)
  - 0x0000 0000** for other ports (**All Input**)
- **GPIO<sub>x</sub>→OSPEEDR:**
  - 0x0000 00C0 for port B(PB.3:100MHz high speed, others:2MHz low speed)
  - 0x0000 0000** for other ports (**All 2MHz low speed**)
- **GPIO<sub>x</sub>→OTYPER:0x0000 0000** (**All Push-Pull**)
- **GPIO<sub>x</sub>→PUPDR:**
  - 0x6400 0000 for port A(PA.15:Pull-up, PA.14:Pull-down, PA.13:Pull-up, others: no)
  - 0x0000 0100 for port B(PA.4:Pull-up, others: no)
  - 0x0000 0000** for other ports (**All no Pull-up, Pull-down**)
- **GPIO<sub>x</sub>→ODR, BSRR : 0x0000 0000**



## 6.1 STM32F407의 GPIO port의 Address(Memory map)

Bus	Boundary address	Peripheral
AHB1	0x4004 0000 - 0x4007 FFFF	USB OTG HS
	0x4002 9400 - 0x4003 FFFF	Reserved
	0x4002 9000 - 0x4002 93FF	ETHERNET MAC
	0x4002 8C00 - 0x4002 8FFF	
	0x4002 8800 - 0x4002 8BFF	
	0x4002 8400 - 0x4002 87FF	
	0x4002 8000 - 0x4002 83FF	
	0x4002 6800 - 0x4002 7FFF	Reserved
	0x4002 6400 - 0x4002 67FF	DMA2
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 5000 - 0x4002 5FFF	Reserved
	0x4002 4000 - 0x4002 4FFF	BKPSRAM
	0x4002 3C00 - 0x4002 3FFF	Flash interface register
	0x4002 3800 - 0x4002 3BFF	RCC
	0x4002 3400 - 0x4002 37FF	Reserved
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2400 - 0x4002 2FFF	Reserved
	0x4002 2000 - 0x4002 23FF	GPIOI
	0x4002 1C00 - 0x4002 1FFF	GPIOH
	0x4002 1800 - 0x4002 1BFF	GPIOG
	0x4002 1400 - 0x4002 17FF	GPIOF
	0x4002 1000 - 0x4002 13FF	GPIOE
	0x4002 0C00 - 0x4002 0FFF	GPIOD
	0x4002 0800 - 0x4002 0BFF	GPIOC
	0x4002 0400 - 0x4002 07FF	GPIOB
	0x4002 0000 - 0x4002 03FF	GPIOA
	0x4001 5800 - 0x4001 FFFF	Reserved

## 6.2 STM32F407의 GPIO 관련 header file(stm32f4xx.h) 주요 부분

```
#define PERIPH_BASE          ((uint32_t)0x40000000) /* Peripheral base  
address in the alias region */  
  
/* Peripheral memory map */  
#define APB1PERIPH_BASE     PERIPH_BASE  
#define APB2PERIPH_BASE     (PERIPH_BASE + 0x00010000)  
#define AHB1PERIPH_BASE     (PERIPH_BASE + 0x00020000)  
#define AHB2PERIPH_BASE     (PERIPH_BASE + 0x10000000)  
  
/* AHB1 peripherals */  
#define GPIOA_BASE          (AHB1PERIPH_BASE + 0x0000)  
#define GPIOB_BASE          (AHB1PERIPH_BASE + 0x0400)  
#define GPIOC_BASE          (AHB1PERIPH_BASE + 0x0800)  
#define GPIOD_BASE          (AHB1PERIPH_BASE + 0x0C00)  
#define GPIOE_BASE          (AHB1PERIPH_BASE + 0x1000)  
#define GPIOF_BASE          (AHB1PERIPH_BASE + 0x1400)  
#define GPIOG_BASE          (AHB1PERIPH_BASE + 0x1800)  
#define GPIOH_BASE          (AHB1PERIPH_BASE + 0x1C00)  
#define GPIOI_BASE          (AHB1PERIPH_BASE + 0x2000)  
#define GPIOJ_BASE          (AHB1PERIPH_BASE + 0x2400)  
#define GPIOK_BASE          (AHB1PERIPH_BASE + 0x2800)  
#define RCC_BASE            (AHB1PERIPH_BASE + 0x3800)
```

## 6.2 STM32F407의 GPIO관련 header file 주요 부분

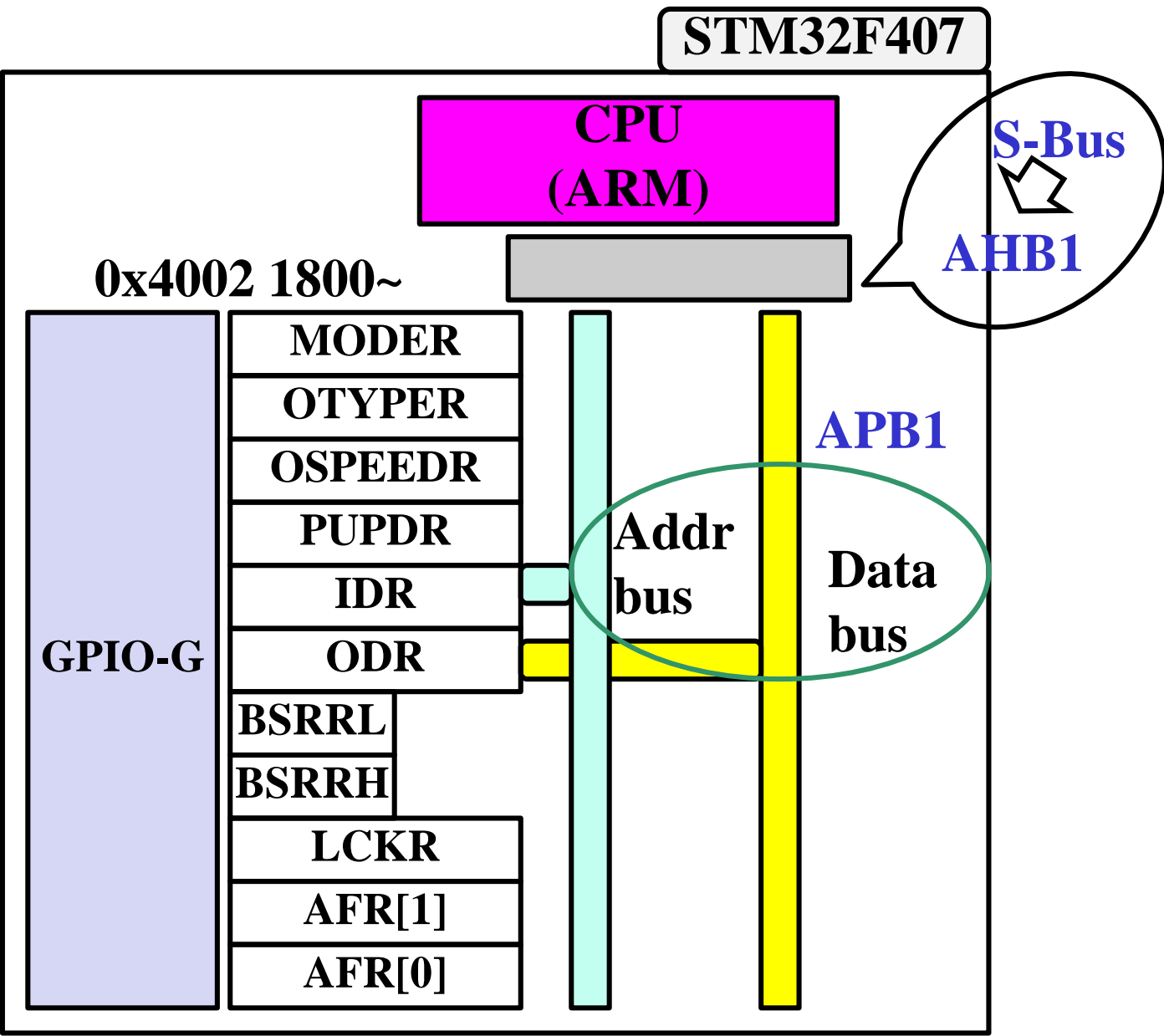
```
#define GPIOA      ((GPIO_TypeDef *) GPIOA_BASE)
#define GPIOB      ((GPIO_TypeDef *) GPIOB_BASE)
#define GPIOC      ((GPIO_TypeDef *) GPIOC_BASE)
#define GPIOD      ((GPIO_TypeDef *) GPIOD_BASE)
#define GPIOE      ((GPIO_TypeDef *) GPIOE_BASE)
#define GPIOF      ((GPIO_TypeDef *) GPIOF_BASE)
#define GPIOG      ((GPIO_TypeDef *) GPIOG_BASE)
#define GPIOH      ((GPIO_TypeDef *) GPIOH_BASE)
#define GPIOI      ((GPIO_TypeDef *) GPIOI_BASE)
#define GPIOJ      ((GPIO_TypeDef *) GPIOJ_BASE)
#define GPIOK      ((GPIO_TypeDef *) GPIOK_BASE)
#define RCC        ((RCC_TypeDef *) RCC_BASE)
```

## 6.2 STM32F407의 GPIO 관련 header file 주요 부분

```
typedef struct {  
    __IO uint32_t MODER; //GPIO port mode register, offset: 0x00  
    __IO uint32_t OTYPER; //GPIO port output type register, offset: 0x04  
    __IO uint32_t OSPEEDR; //GPIO port output speed register, offset: 0x08  
    __IO uint32_t PUPDR; //GPIO port pull-up/pull-down register, 0x0C  
    __IO uint32_t IDR; //GPIO port input data register, offset: 0x10  
    __IO uint32_t ODR; //GPIO port output data register, offset: 0x14  
    __IO uint16_t BSRRL; //GPIO port bit set/reset low register, 0x18  
    __IO uint16_t BSRRH; //GPIO port bit set/reset high register, 0x1A  
    __IO uint32_t LCKR; //GPIO port configuration lock register, 0x1C  
    __IO uint32_t AFR[2]; //GPIO alternate function registers, 0x20-0x24  
} GPIO_TypeDef;
```

```
typedef __IO uint32_t vu32;  
typedef __IO uint16_t vu16;  
typedef __IO uint8_t vu8;
```

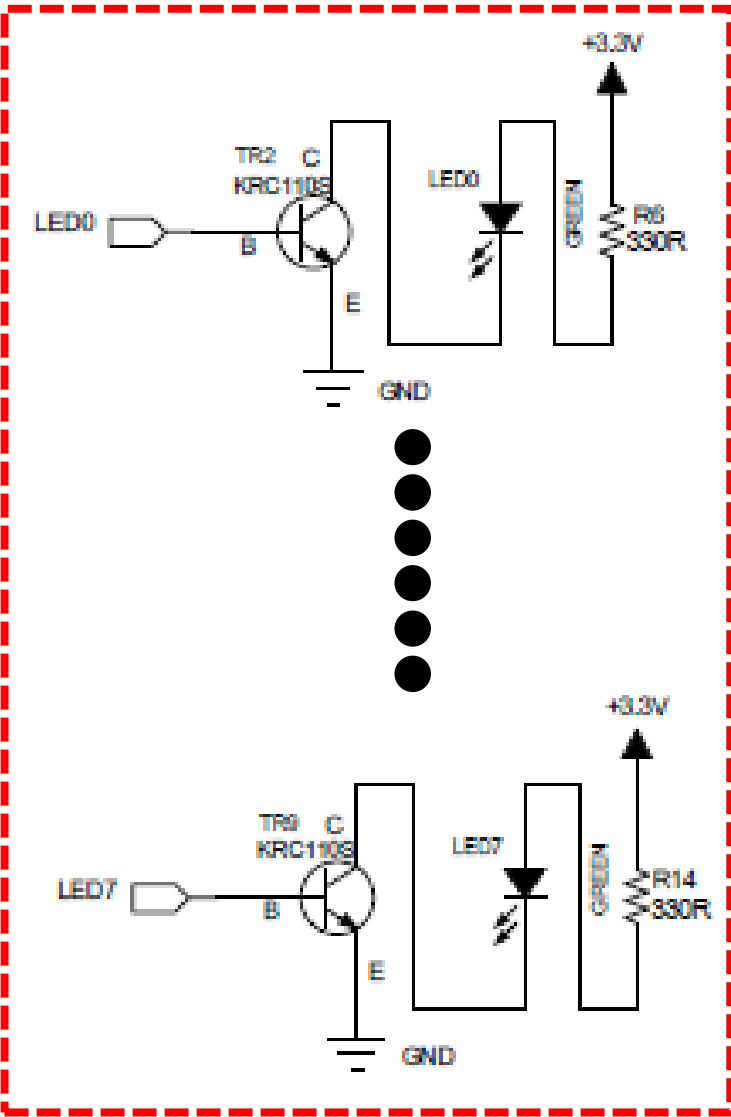
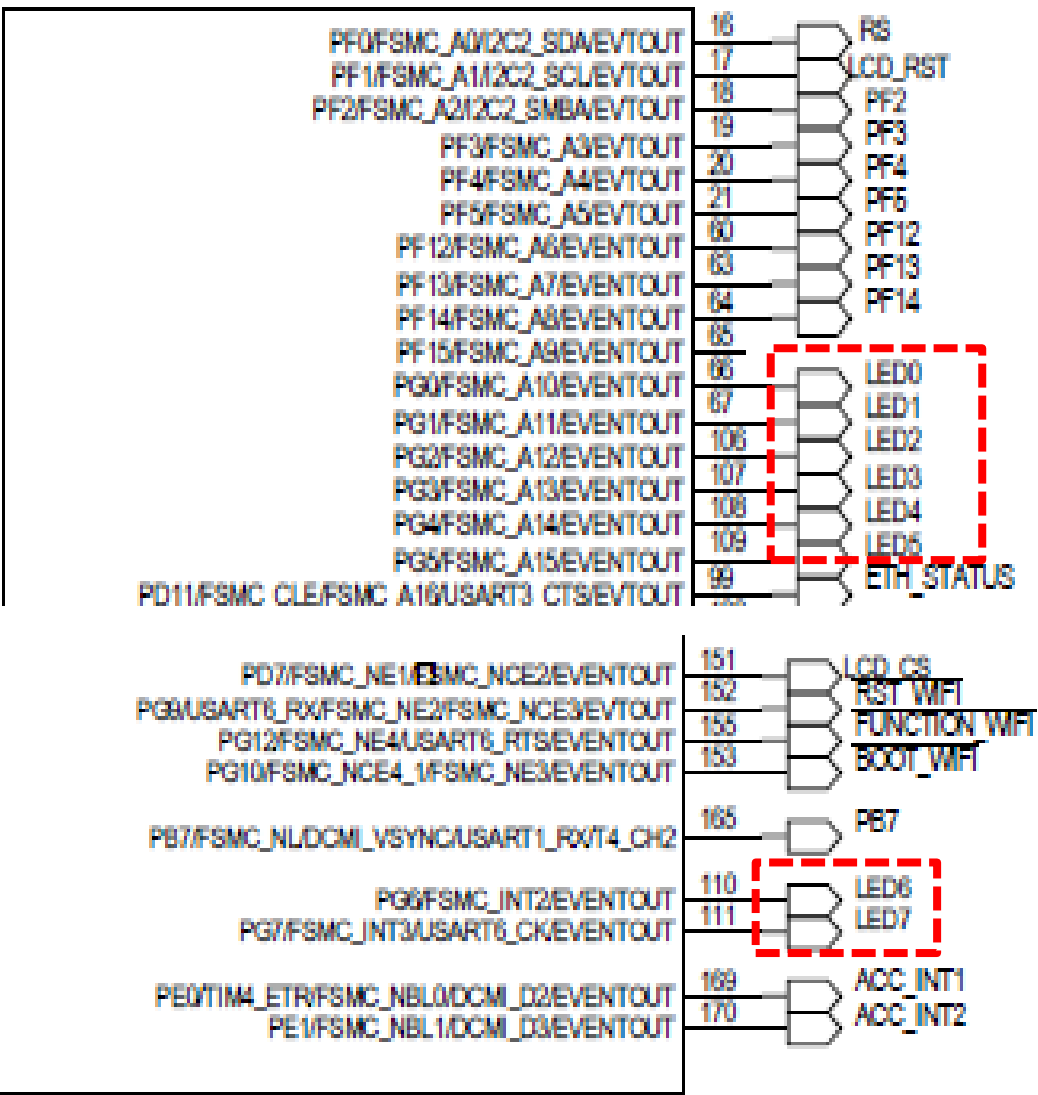
# 6.3 STM32F407의 GPIO port 내부 구조 (예: GPIOG)



# 6.4 GPIO 구동실습을 위한 실습보드의 출력인터페이스 회로

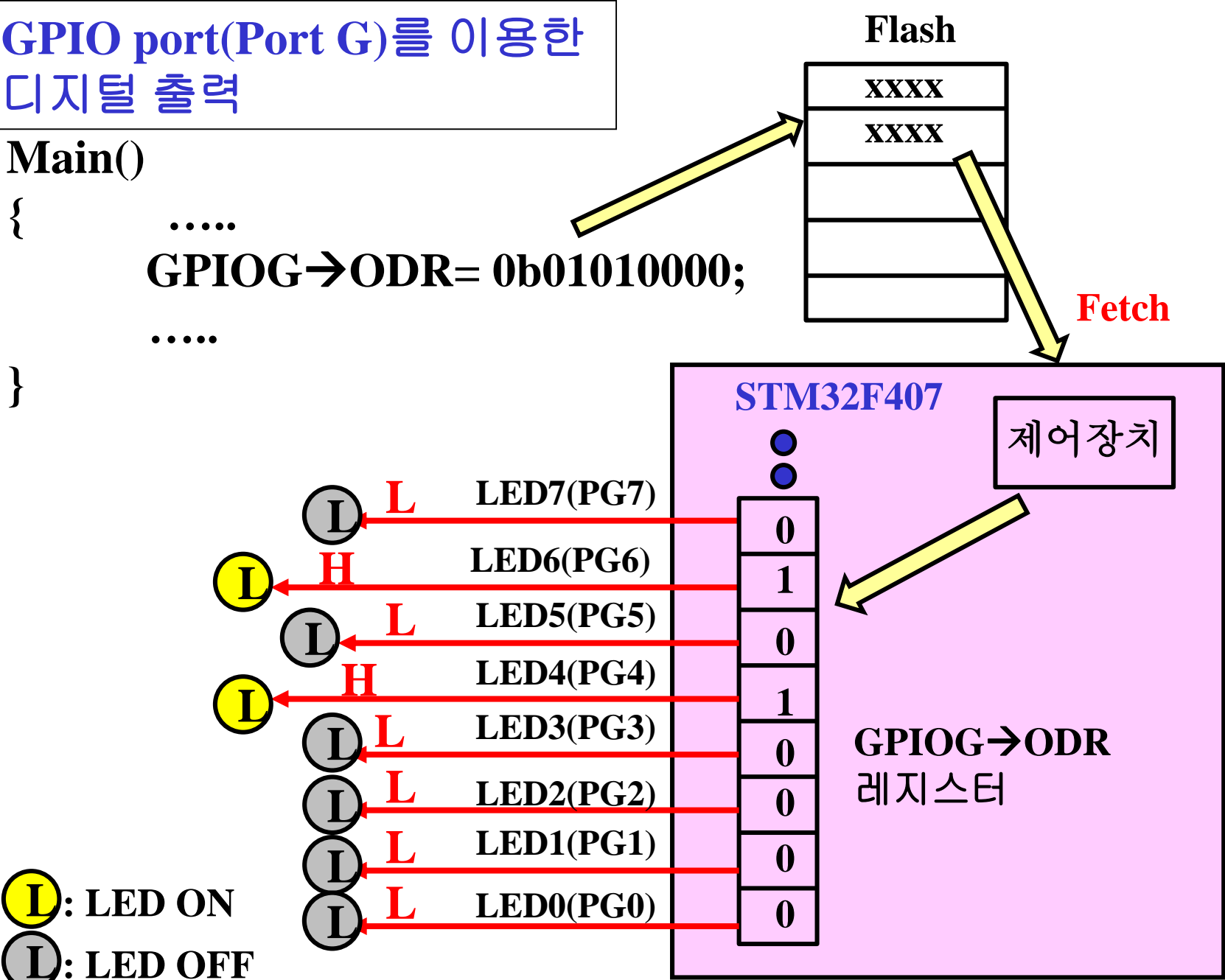
- Output: LED\*8 (LED7 ~ LED0), 관련 GPIO port : PG7~PG0

U1-C



# GPIO port(Port G)를 이용한 디지털 출력

```
Main()
{
    ....
    GPIOG→ODR= 0b01010000;
    ....
}
```



## \* 레지스터 정의(예)

**GPIOG→ODR**

**= (\*(volatile unsigned \*) GPIOG\_BASE+0x14)**

**= (\*(volatile unsigned \*) AHB1PERIPH\_BASE + 0x1800 +0x14)**

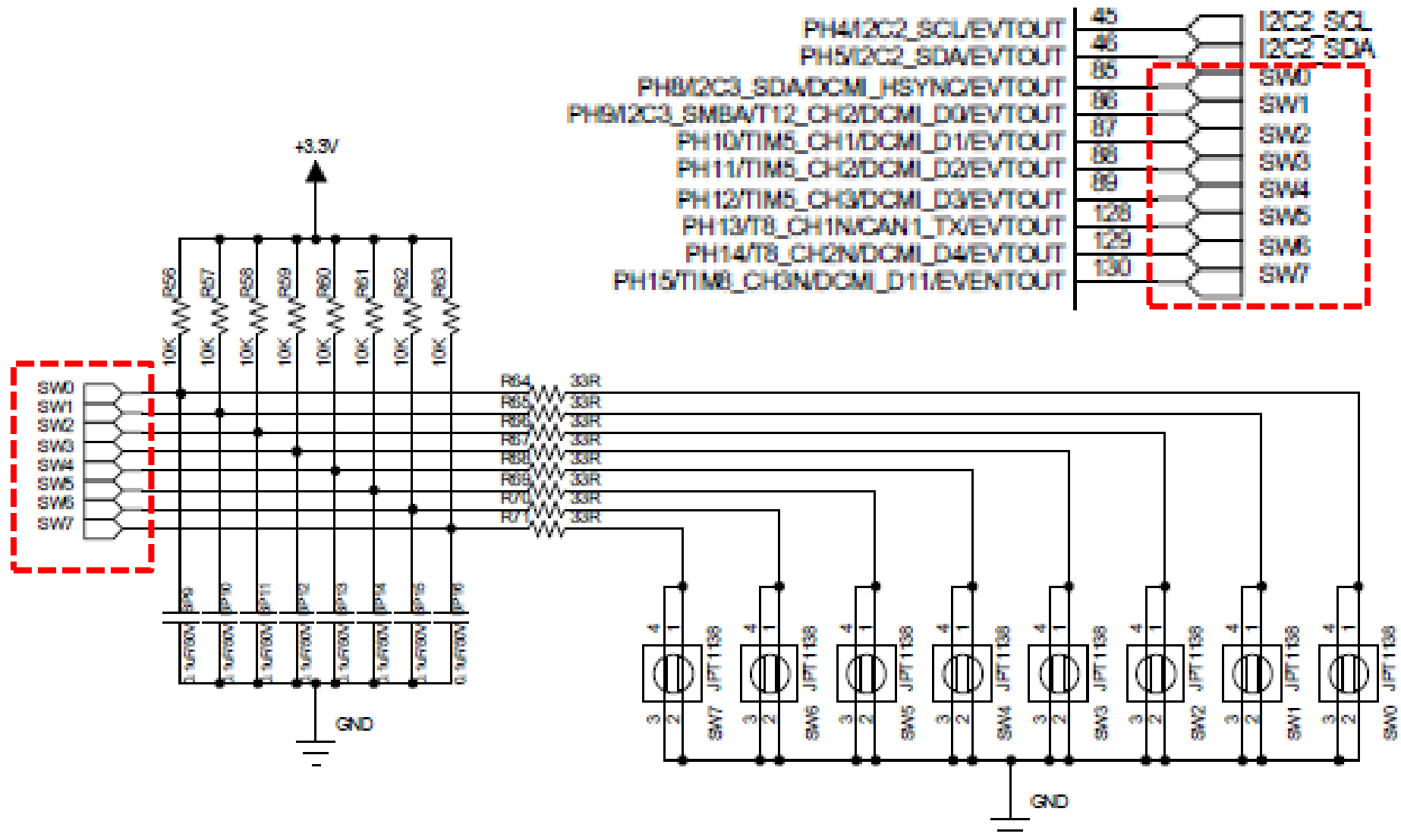
**= (\*(volatile unsigned \*) PERIPH\_BASE + 0x00020000+ 0x1800 +0x14)**

**= (\*(volatile unsigned \*) (uint32\_t)0x40000000 + 0x00020000 + 0x1800 + 0x14)**

**= (\*(volatile unsigned \*) (uint32\_t)0x40021814)**

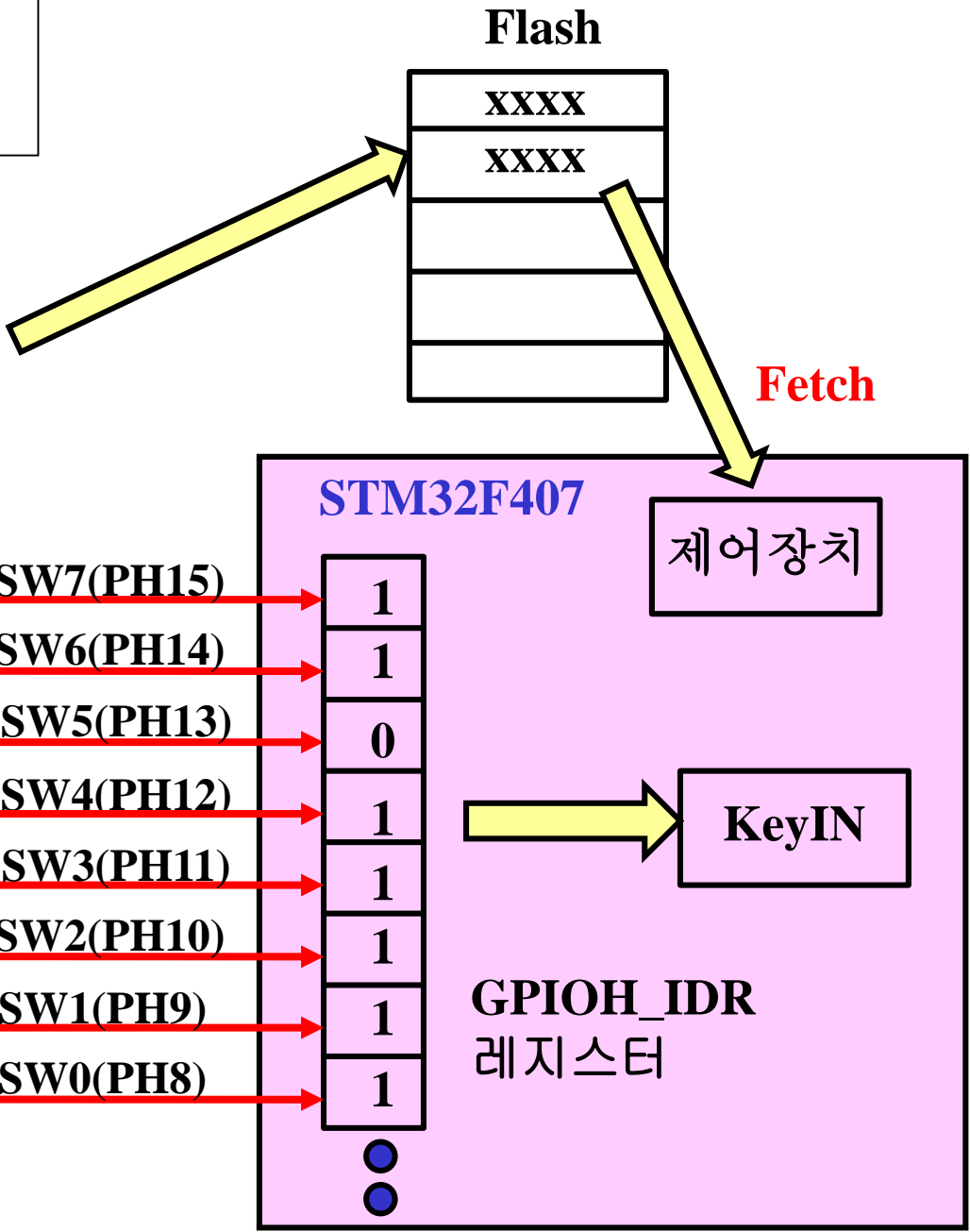


- **Input: SW\*8 (SW7 ~ SW0), 관련 GPIO port : PH15~PH8**



# GPIO port(Port H)를 이용한 디지털 입력

```
unsigned char KeyIN;  
main()  
{  
    ....  
    KeyIN= GPIOH_IDR;  
    ....  
}
```



**K** : Key ON  
**K** : Key OFF