

# DIGITAL IN & OUT USING MBED-OS

Handong university

Jong-won Lee

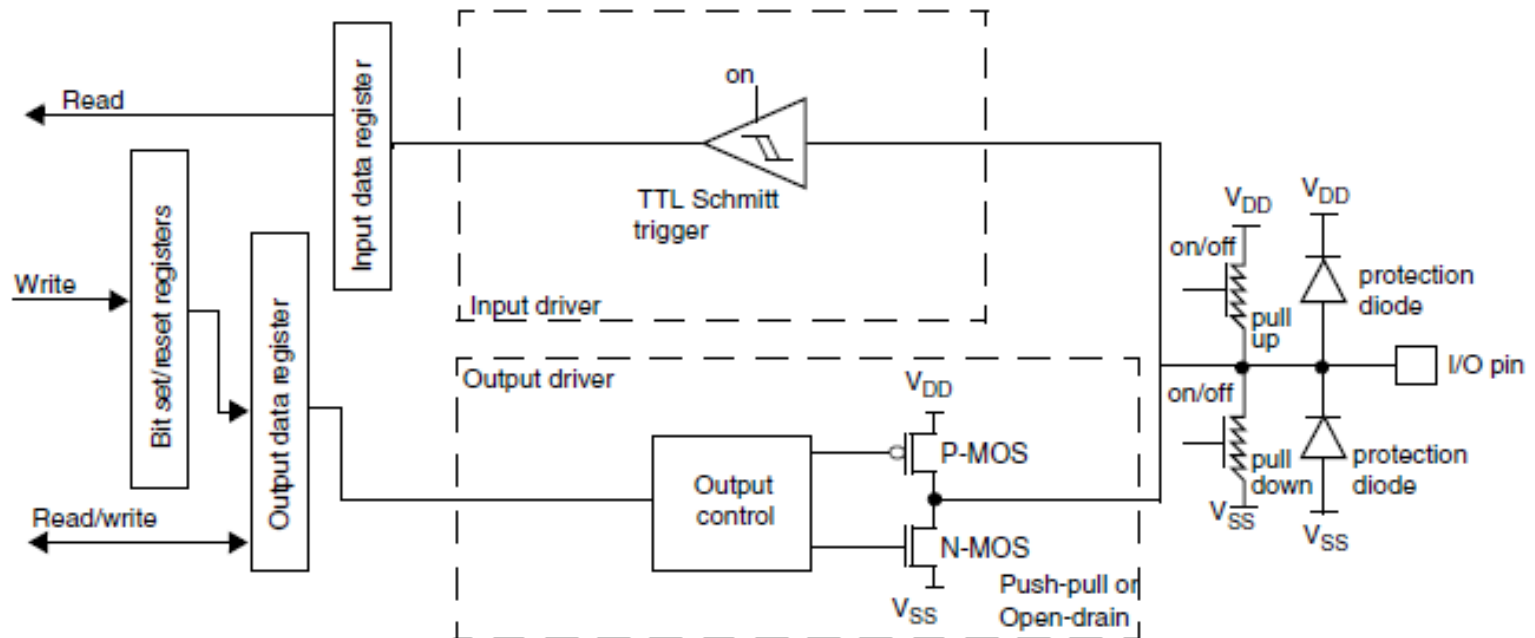
2

DigitalOut

# GPIO Output Configuration

3

- Push-pull or open-drain type
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx\_PUPDR register.
- A read access to the output data register gets the last written value



# DigitalOut API in mbed-OS

4

## □ DigitalOut class

생성자	<code>DigitalOut(PinName pin)</code> <code>DigitalOut (PinName pin, int value)</code>
설명	GPIO 포트의 어떤 pin을 디지털 출력 장치로 설정한다. 혹은 디지털 출력 장치의 초기값을 value로 한다. 초기값이 주어지지 않았을 경우 디지털 출력 장치의 초기값은 '0'으로 주어진다.
인자	pin: PA_0, ... PA_15, PB_0,..., PB_15 등 혹은 D0, ..., D15 (아두이노 확장 커넥터 이름) (Pinname에 대한 정의는 <a href="https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/TARGET_STM32F4/TARGET_STM32F411xE/TARGET_NUCLEO_F411RE/PinNames.h">https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/TARGET_STM32F4/TARGET_STM32F411xE/TARGET_NUCLEO_F411RE/PinNames.h</a> 참조)
사용 예	<code>DigitalOut led1(D2);</code> <code>DigitalOut led2(PA_5, 0);</code>

# DigitalOut API in mbed-OS

5

## □ DigitalOut class

함수	<code>void write(int value)</code>
설명	디지털 출력 장치로 설정된 GPIO 포트의 핀을 '0' 혹은 '1'의 값으로 출력한다. (value 값이 0이 아니면 출력 값은 '1'에 해당 하는 값이 출력된다.)
사용 예	<code>led1.write(0);</code> <code>led1.write(1);</code>
함수	<code>int read()</code>
설명	디지털 출력 장치로 설정된 GPIO 포트의 핀에서 입력 신호를 읽는다. (코드를 살펴보면 GPIOx_IDR 레지스터 값을 읽는다.)
사용 예	<code>int a = led1.read();</code>
함수	<code>DigitalOut&amp; operator= (int value)</code>
설명	= 연산자에 대한 재정의 함수로 write() 함수를 간단하게 사용할 수 있도록 정의한 함수이다.
사용 예	<code>led1 = 1;     //the same as led1.write(1)</code>

# DigitalOut API in mbed-OS

6

## □ DigitalOut class

함수	<code>DigitalOut&amp; operator= (DigitalOut&amp; rhs)</code>
설명	= 연산자에 대한 재정의 함수로 <code>write()</code> 함수를 간단하게 사용할 수 있도록 정의한 함수이다. (rhs에도 <code>DigitalOut</code> 객체가 오게 되는 경우로 rhs의 <code>DigitalOut</code> 객체로부터 값을 읽어서 <code>DigitalOut</code> 객체로 해당 값을 출력시키는 경우에 해당한다.)
사용 예	<code>led1 = !led2</code>
함수	<code>operator int()</code>
설명	이것은 일반적으로 C++에서 변환 연산자(conversion operator)이다. <code>DigitalOut</code> 클래스에서는 <code>read()</code> 를 간단하게 하기 위하여 사용된다. (변환 연산자의 의미로 설명한다면 <code>DigitalOut</code> 형태를 <code>int</code> 형태로 변환하는 것이다.)
사용 예	<code>int a = led1;     //the same as int a = led1.read();</code>

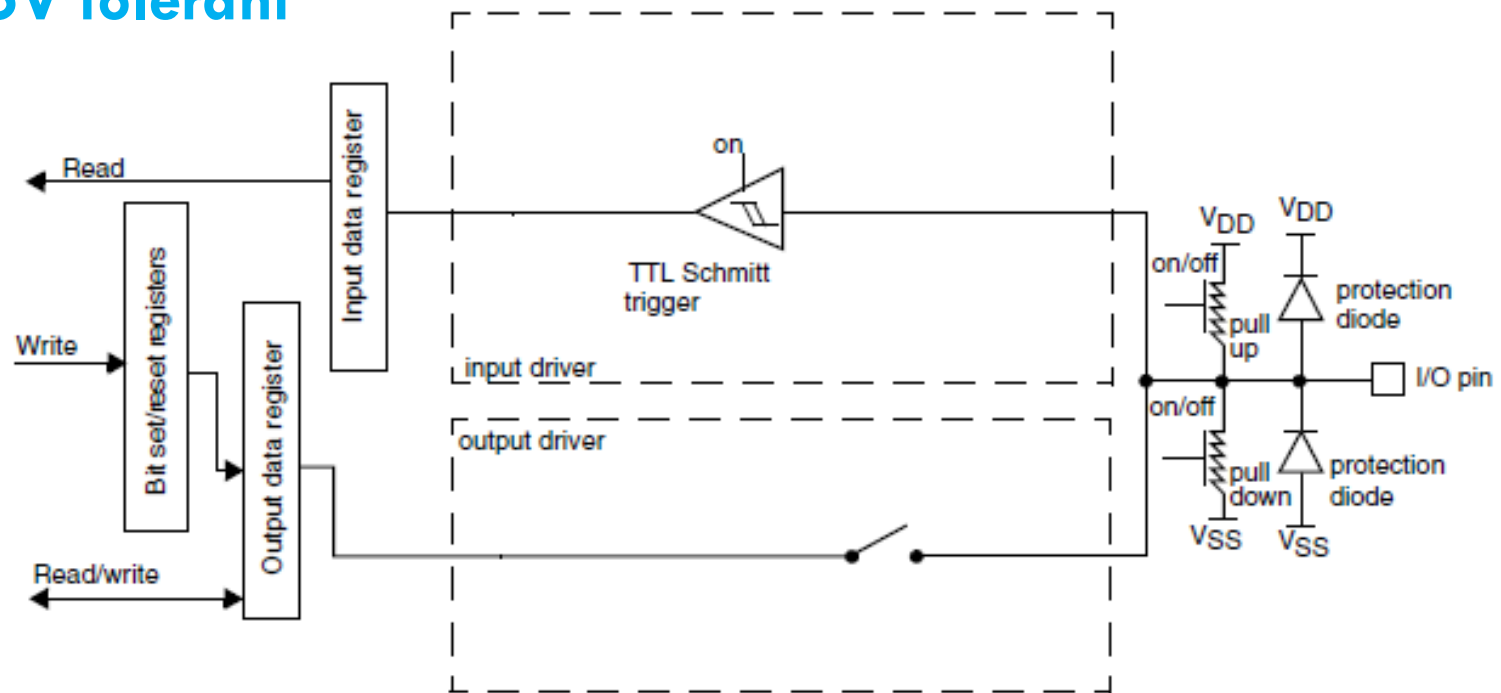
7

DigitalIn

# GPIO Input Configuration

8

- The output buffer is disabled
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx\_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- **5V tolerant**





# DigitalIn API in mbed-OS

9

## □ DigitalIn class

생성자	<code>DigitalIn(PinName pin)</code> <code>DigitalIn (PinName pin, PinMode mode)</code>
설명	GPIO 포트의 pin을 디지털 입력 장치로 설정한다. 혹은 디지털 입력 장치의 pin의 입력 신호 형태를 mode로 한다. mode 값이 주어지지 않을 경우에는 PullNone이 주어진 경우와 동일하다. 즉 풀업과 풀다운이 없는 일반적인 형태이다.
인자	PinName: PA_0, ... PA_15, PB_0,..., PB_15 등 혹은 D0, ..., D15 (아두이노 확장 커넥터 이름) (Pinname에 대한 정의는 <a href="https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/TARGET_STM32F4/TARGET_STM32F411xE/TARGET_NUCLEO_F411RE/PinNames.h">https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/TARGET_STM32F4/TARGET_STM32F411xE/TARGET_NUCLEO_F411RE/PinNames.h</a> 참조) PinMode: PullUp, PullDown, PullNone, OpenDrainPullUp, OpenDrainPullDown, OpenDrainNoPull
사용 예	<code>DigitalIn button(PC_13);</code> <code>DigitalIn button(PC_13, PullUp);</code>

# DigitalIn API in mbed-OS

10

## □ DigitalIn class

함수	<code>int read()</code>
설명	디지털 입력 장치로 설정된 GPIO 포트의 핀에서 입력 신호를 읽는다.
사용 예	<code>int a = button.read();</code>
함수	<code>void mode(PinMode pull)</code>
설명	입력 핀의 모드를 설정한다.
사용 예	<code>button.mode(PullUp);</code>
함수	<code>operator int()</code>
설명	이것은 C++에서의 변환 연산자이다. DigitalIn 클래스에서는 <code>read()</code> 를 간단하게 하기 위하여 사용된다.
사용 예	<code>int a = button;        //the same as int a = button.read();</code>

# Mbed API: wait functions

11

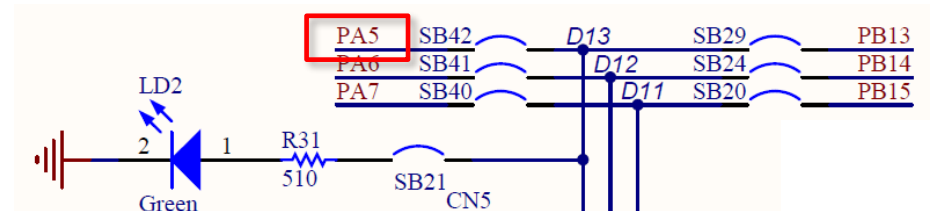
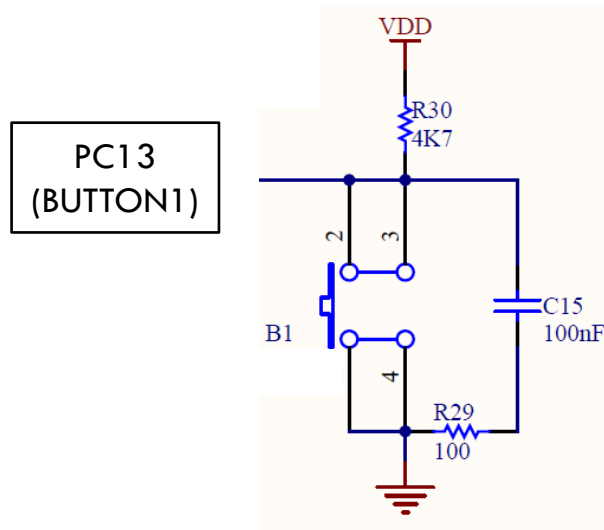
## □ Functions related to wait.

- ▣ The OS scheduler will put the current thread in waiting state, allowing another thread to execute.
- ▣ void **ThisThread::sleep\_for** (Kernel::Clock::duration\_u32 rel\_time )
  - Sleep for a specified time period
  - Ex.:
    - ThisThread::sleep\_for(500ms), ThisThread::sleep\_for(5s)
- ▣ void **ThisThread::sleep\_for** (uint32\_t millisec)
  - Sleep for a specified time period in millisec:
  - **Deprecated**
- ▣ void **wait\_us**(int us);
  - Waits a number of microseconds.
- ~~▣ void **wait**(float s);~~
- ~~▣ void **wait\_ms**(int ms);~~

# Lab. 1-1: Digital Output and Input

12

- Control the blinking rate of the LD2 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 → 250 ms on/off → 500 ms on/off → ...
- BUTTON1 & LED1

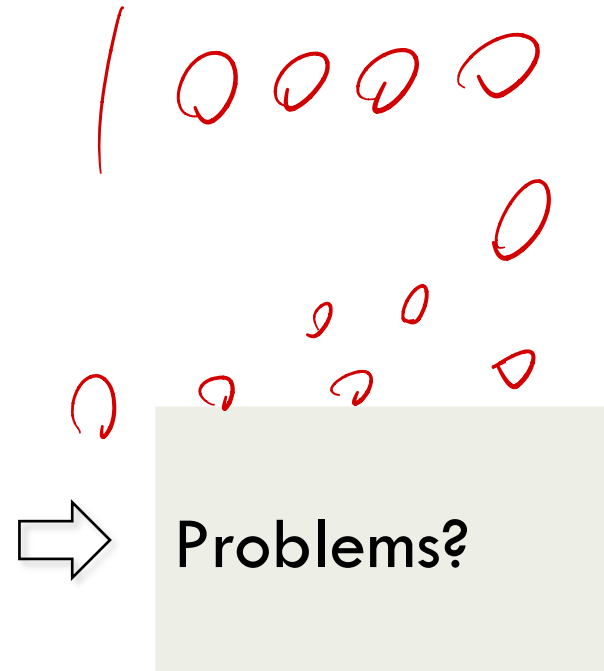


# Lab. 1-1: Digital Output and Input

13

## □ A sample code

```
main.cpp X
1 #include "mbed.h"
2
3 DigitalOut led(LED1);
4 DigitalIn button(BUTTON1);
5
6 // main() runs in its own thread in the OS
7 int main() {
8     int interval = 500;
9
10    while (true) {
11
12        if (button == 0) {
13            interval = interval << 1;
14            if (interval > 2000)
15                interval = 250;
16        }
17
18        led = !led;
19        ThisThread::sleep_for(
20            std::chrono::milliseconds(interval));
21    }
22 }
```



14

# Digital Input & Output

# DigitalInOut API in mbed-OS

15

## □ DigitalInOut class

생성자	<code>DigitalInOut(PinName pin)</code> <code>DigitalInOut (PinName pin, PinDirection dir, , PinMode mode, int value)</code>
설명	GPIO 포트의 어떤 핀을 디지털 입출력 장치로 설정한다. <code>PinDirection</code> 이 주어지지 않는 경우에는 디폴트 상태가 디지털 입력 장치로 동작한다. 혹은 디지털 입출력 장치를 입력 장치 혹은 출력 장치로 초기화시키며, 해당 핀의 모드를 설정하고, 출력 장치일 때 출력 신호의 초기값을 <code>value</code> 로 한다. 입력 장치인 경우 <code>value</code> 는 무의미하다.
인자	<code>pin</code> : PA_0, ... PA_15, PB_0,..., PB_15 등 혹은 D0, ..., D15 (아두이노 확장 커넥터 이름) <code>dir</code> : PIN_OUTPUT, PIN_INPUT <code>mode</code> : PullUp, PullDown, PullNone, OpenDrainPullUp, OpenDrainPullDown, OpenDrainNoPull
사용 예	<code>DigitalInOut inOut(D2);</code> <code>DigitalInOut inOut(PA_5, PIN_OUTPUT, PullNone, 0)</code>

# DigitalInOut API in mbed-OS

16

## □ DigitalInOut class

함수	<code>void write(int value)</code>
설명	디지털 출력 장치로 동작할 때, GPIO 포트의 핀을 '0' 혹은 '1'의 값으로 출력한다. (value 값이 0이 아니면 출력 값은 '1'에 해당 하는 값이 출력된다.)
사용 예	<code>inOut.write(0);</code> <code>inOut.write(1);</code>
함수	<code>DigitalInOut&amp; operator= (int value)</code>
설명	= 연산자에 대한 재정의 함수로 write() 함수를 간단하게 사용할 수 있도록 정의한 함수이다.
사용 예	<code>inOut = 1;     //the same as inOut.write(1)</code>
함수	<code>DigitalInOut&amp; operator= (DigitalInOut&amp; rhs)</code>
설명	= 연산자에 대한 재정의 함수로 write() 함수를 간단하게 사용할 수 있도록 정의한 함수이다.
사용 예	<code>inOut1 = inOut2;     //the same as inOut1.write(inOut2.read());</code>
함수	<code>int read()</code>
설명	디지털 입력 혹은 출력 장치로 동작할 때, GPIO 포트의 핀에서 입력 신호를 읽는다. (GPIOx_IDR 레지스터 값을 읽는다.)
사용 예	<code>int a = inOut.read();</code>



# DigitalInOut API in mbed-OS

17

## □ DigitalInOut class

함수	<code>operator int()</code>
설명	DigitalInOut 클래스에서는 <code>read()</code> 를 간단하게 하기 위하여 사용된다.
사용 예	<code>int a = inOut;     //the same as int a = inOut.read();</code>
함수	<code>void input()</code>
설명	디지털 입력 장치로 동작하도록 설정한다.
사용 예	<code>inOut.input();</code>
함수	<code>void output()</code>
설명	디지털 출력 장치로 동작하도록 설정한다.
사용 예	<code>inOut.output();</code>
함수	<code>void mode(PinMode pull)</code>
설명	해당 포트의 모드를 결정한다. (PullUp/PullDown/PullNone/ OpenDrain)
사용 예	<code>inOut.mode(PullUp);</code>

# DIGITAL IN & OUT USING MBED-OS

Handong university

Jong-won Lee

2

# Interrupt Input

# Mbed InterruptIn API

3

## □ InterruptIn class

생성자	<code>InterruptIn(PinName pin)</code>
설명	인터럽트 처리를 위한 구조체를 형성하고, 기본적인 인터럽트 설정을 수행한 다음, 해당 GPIO 핀을 디지털 입력 장치로 설정한다.
인자	pin: PA_0, ... PA_15, PB_0,..., PB_15 등 혹은 D0, ..., D15 (아두이노 확장 커넥터 이름)
사용 예	<code>InterruptIn button(PC_13);</code>

# Mbed InterruptIn API

4

## □ InterruptIn class

함수	<code>void fall(Callback&lt;void()&gt; func)</code>
설명	하강 에지 이벤트 발생 때 실행될 ISR인 <code>func</code> 를 등록한다.
사용 예	<code>button.fall(&amp;button_pressed);</code>
함수	<code>void rise(Callback&lt;void()&gt; func)</code>
설명	상승 에지 이벤트 발생 때 실행될 ISR인 <code>func</code> 를 등록한다.
사용 예	<code>button.rise(&amp;button_released);</code>
함수	<code>void enable_irq()</code>
설명	인터럽트 입력 장치로 설정된 핀에 대한 인터럽트를 활성화시킨다.
사용 예	<code>button.enable_irq();</code>
함수	<code>void disable_irq()</code>
설명	인터럽트 입력 장치로 설정된 핀에 대한 인터럽트를 비활성화시킨다.
사용 예	<code>button.disable_irq();</code>

# Mbed InterruptIn API

5

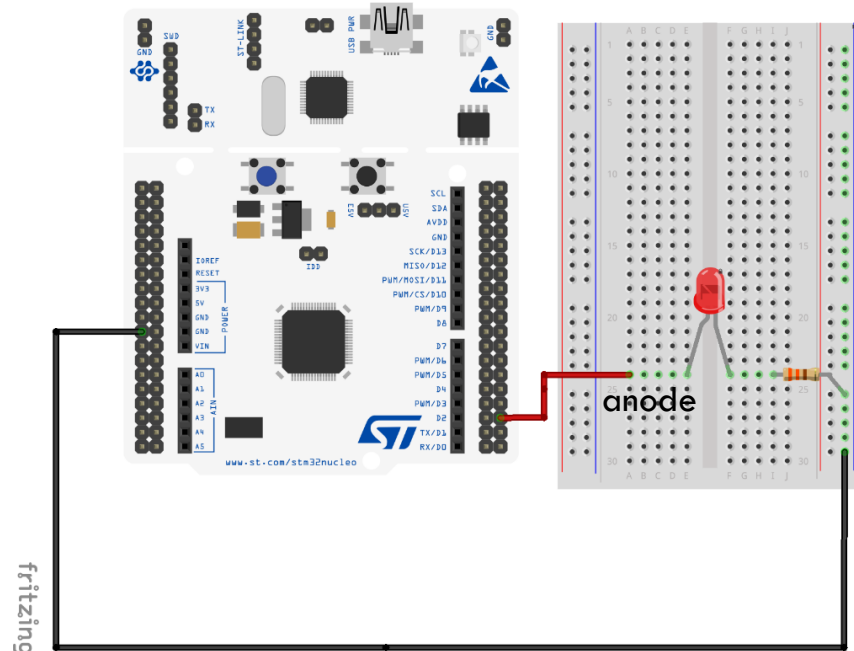
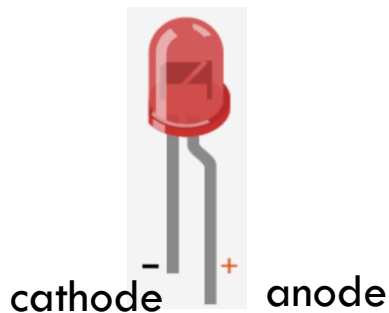
## □ InterruptIn class

함수	<code>int read()</code>
설명	인터럽트 입력 장치로 설정된 핀에서 입력 신호 값을 읽는다.
사용 예	<code>int a = button.read();</code>
함수	<code>void mode(PinMode pull)</code>
설명	입력 핀의 모드를 설정한다.
사용 예	<code>button.mode(PullUp);</code>
함수	<code>operator int()</code>
설명	<code>read()</code> 를 간단하게 하기 위하여 사용된다.
사용 예	<code>int a = button; //the same as int a = button.read();</code>

# Lab. 1-2 Interrupt Input

6

- Green LED: blinking it at a rate of 1 sec. (500ms on/ 500ms off)
  - ▣ Green LED는 board에 있는 LED를 이용한다.
- Red LED: toggle the LED whenever a user button is pressed.
  - ▣ Red LED는 D2 (Arduino Uno connector)에 연결한다. (아래 그림 참조.)
    - LED 극성을 조심하여 연결하고, 저항은 220 ohm 정도를 사용한다.

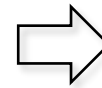


# Lab. 1-2 Interrupt Input

7

## □ A sample code

```
main.cpp X
1 #include "mbed.h"
2
3 DigitalOut redLED(D2);
4 DigitalOut greenLED(LED1);
5 InterruptIn button(BUTTON1);
6
7 volatile int interval;
8
9 void buttonPressed()
10 {
11     redLED = !redLED;
12 }
13
14 // main() runs in its own thread in the OS
15 int main() {
16     button.fall(&buttonPressed);
17
18     while (true) {
19         greenLED = !greenLED;
20         ThisThread::sleep_for(500ms);
21     }
22 }
```



Button을 20 ~ 30번 정도 눌렀을 때, (대부분의 경우는 발생하는) 비정상적인 현상은 어떤 것이 있는가 관찰하여라. 그 원인은 무엇이라고 생각하는가?



8

BusOut, BusIn, BusInOut  
PortOut, PortIn, PortInOut

# BusOut API in mbed-OS

9

## □ BusOut class

- ▣ 다수의 디지털 출력 장치를 하나의 논리적인 장치로 구성하여 사용하는 것으로, 최대 16개의 디지털 출력 장치를 수용할 수 있다.

생성자	<code>BusOut (PinName p0, PinName p1=NC, PinName p2=NC, PinName p3=NC, PinName p4=NC, PinName p5=NC, PinName p6=NC, PinName p7=NC, PinName p8=NC, PinName p9=NC, PinName p10=NC, PinName p11=NC, PinName p12=NC, PinName p13=NC, PinName p14=NC, PinName p15=NC)</code>
생성자	<code>BusOut (PinName pins[16])</code>
사용 예	<code>BusOut motorCtl(D2, D3, D4, D5);</code>

- ▣ The order of pins in the constructor is the reverse order of the pins in the byte order. So if you have `BusOut(a,b,c,d,e,f,g,h)`, then the order of bits in the byte would be determined with a being bit 0, b being bit 1, c being bit 2 and so on.

# BusOut API in mbed-OS

10

## □ BusOut class

함수		설명
void	write(int value)	버스 값을 출력한다
int	read()	버스 값 읽는다.
	operator int()	read() 함수의 간략한 형태이다.
DigitalOut&	operator= (int value)	write() 함수의 간략한 형태이다.
DigitalOut&	operator= (DigitalOut& rhs)	write() 함수의 간략한 형태이다.

# Lab. 1-3 BusOut

11

- Red LED, Green LED, Yellow LED가 각각 D2, D3, D4 port에 연결되어 있다.
- Red LED는 1초 주기 (0.5초 On/ 0.5초 Off),
- Green LED는 2초 주기,
- Yellow LED는 4초 주기로 점멸하는 프로그램을 BusOut을 이용하여 구현하시오.

# Lab. 1-3 BusOut

12

## □ A sample code

```
main.cpp X
1 #include "mbed.h"
2
3 // D2: Red LED, D3: Green LED, D4: Yellow LED
4 PinName pins[3] = {D2, D3, D4};
5 BusOut RGYLeds(pins);
6 //BusOut RGYLeds(D2, D3, D4);
7
8 // main() runs in its own thread in the OS
9 int main() {
10     while (true) {
11         for(int color = 0; color < 8; color++) {
12             RGYLeds = color;
13             ThisThread::sleep_for(500ms);
14         }
15     }
16 }
```

# BusIn API in mbed-OS

13

## □ BusIn class

- ▣ 다수의 디지털 입력 장치를 하나의 논리적인 장치로 구성하여 사용하는 것으로, 최대 16개의 디지털 입력 장치를 수용할 수 있다.

생성자 혹은 함수	설명
<code>BusIn (PinName p0, PinName p1=NC, PinName p2=NC, PinName p3=NC, PinName p4=NC, PinName p5=NC, PinName p6=NC, PinName p7=NC, PinName p8=NC, PinName p9=NC, PinName p10=NC, PinName p11=NC, PinName p12=NC, PinName p13=NC, PinName p14=NC, PinName p15=NC)</code>	생성자
<code>BusIn (PinName pins[16])</code>	생성자

# BusIn API in mbed-OS

14

## □ BusIn class

int	read()	버스 값을 읽는다.
	operator int()	read() 함수의 간략한 형태이다.
void	mode(PinMode mode)	입력 핀들의 모드를 설정한다.
DigitalIn&	operator[] (int index)	특별한 입력 비트를 읽는다.

# BusInOut API in mbed-OS

15

## □ BusInOut class

<b>BusInOut</b> (PinName p0, PinName p1=NC, PinName p2=NC, PinName ... p13=NC, PinName p14=NC, PinName p15=NC)	constructor
<b>BusInOut</b> (PinName pins[16])	constructor
void <b>write</b> (int value)	value
int <b>read</b> ()	Return value
void <b>output</b> ()	Set all the pins as output.
void <b>input</b> ()	Set all the pins as input.
void mode(PinMode mode)	Set all input pin mode
operator int()	A shorthand for read()
BusInOut & operator= (int value)	A shorthand for write
DigitalInOut& operator[] (int index)	Access to a particular bit



# PortOut API in mbed-OS

16

## □ PortOut class

- 하나의 포트(즉 GPIOA, GPIOB 등)에 속한 다수의 디지털 출력 장치를 하나의 논리 장치로 구성할 수 있다.
  - BusOut: 임의의 GPIO 포트의 핀들로 하나의 논리 장치로 구성할 수 있다.
  - BusOut/BusIn/BusInOut 장치들은 구성에서 유연성이 있는 반면, PortOut/PortIn/PortInOut 장치는 동작 속도에서 장점이 있다.

생성자	<code>PortOut (PortName port, int mask=0xFFFFFFFF)</code>
인자	<p>port: PortA/PortB/.../PortH 값을 지닐 수 있다. 다음 헤더 파일에 정의되어 있다. (<a href="https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/PortNames.h">https://github.com/ARMmbed/mbed-os/blob/master/targets/TARGET_STM/PortNames.h</a>)</p> <p>mask: 한 포트에서 어떤 비트들이 포함되었는가를 표시한다. (비트의 값이 '0'인 비트는 포함되지 않는다.)</p>
사용 예	<p>PB_3, PB_4, PB_5와 PB_6를 포트 출력으로 사용하려 할 때: <code>PortOut motorCtl(PortB, 0x0078);</code></p>

# PortOut API in mbed-OS

17

## □ PortOut class

함수		설명
void	write(int value)	포트 값을 출력한다.
int	read()	포트 값 읽는다.
	operator int()	read() 함수의 간략한 형태이다.
DigitalOut&	operator= (int value)	write() 함수의 간략한 형태이다.
DigitalOut&	operator= (DigitalOut& rhs)	write() 함수의 간략한 형태이다.

# Lab. 1-4 PortOut

18

- Red LED, Green LED, Yellow LED가 각각 D3, D5, D4 port에 연결되어 있다.
  - ▣  $D3 = PB\_3$ ,  $D5 = PB\_4$ ,  $D4 = PB\_5$
- Red LED는 1초 주기 (0.5초 On/ 0.5초 Off),
- Green LED는 2초 주기,
- Yellow LED는 4초 주기로 점멸하는 프로그램을 PortOut을 이용하여 구현하시오.

# Lab. 1-4 PortOut

19

## □ A sample code

```
main.cpp X
1 #include "mbed.h"
2
3 // D3 = PB_3, D5=PB_4, D4=PB_5
4 // D3: Red , D5: Green, D4: Yellow
5 PortOut RGYLeds(PortB, 0x0038);
6
7 // main() runs in its own thread in the OS
8 int main() {
9     while (true) {
10         for(int color = 0; color < 8; color++) {
11             RGYLeds = color << 3;
12             ThisThread::sleep_for(500ms);
13         }
14     }
15 }
```

178

6 5 4

3 2 1 0

0000

0000

0011

1000

(1)

0000

1000

# PortIn API in mbed-OS

20

## □ PortIn class

생성자 혹은 함수		설명
PortIn (PortNmae port, int mask=0xFFFFFFFF)		생성자
int	read()	포트 값 읽음.
operator int()		read() 함수의 간략한 형태
void	mode(PinMode mode)	입력 핀들의 모드 설정함.

# Mbed PortInOut API

21

## □ PortInOut class

<b>PortInOut</b> (PortName port, int mask=0xFFFFFFFF )	Constructor (include a port with '1' )
void <b>write</b> (int value)	value
int <b>read</b> ()	Return value
void <b>output</b> ()	Set all the pins as output.
void <b>input</b> ()	Set all the pins as input.
void mode(PinMode mode)	Set input pin mode
operator int()	A shorthand for read()
PortInOut & operator= (int value)	A shorthand for write

# DIGITAL IN & OUT USING MBED-OS

Handong university

Jong-won Lee

2

# Labs: Exercises



# Lab. 1-5

3

## □ 실습 내용 (30점)

- ▣ Red LED를 1초 간격 (500ms on/ 500ms off)으로,
- ▣ Green LED를 2초 간격으로 , Yellow LED를 4초 간격으로 점멸하도록 프로그램을 작성하시오.

## □ 회로 구성

- ▣ D3 핀에 Red LED와 저항(220 ohm),
- ▣ D4 핀에 Green LED와 저항,
- ▣ D5 핀에 Yellow LED와 저항 연결

0011 (000)

- PortOut을 이용하여 구현하여야 한다.
- 가능한 코드의 길이를 짧게 작성하여야 한다.

0X0038 = 0011 / 000

	Y	G	R		
Σ 900	0	0	0	000	0
	0	0	1		1
	0	0	0		0
	0	0	1		1
	0	1	0	23234456767	2
	0	1	1		3
	0	1	0		2
	0	1	1		3
	0	0	0		4
	1	0	1		4
	1	0	0		4
	1	0	1		5
	1	1	0		6
	1	1	1		7
	1	1	0		6
	1	1	1		7

# Lab. 1-6

4

## □ 실습 내용 (30점)

- ▣ Red LED를 1초 간격 (500ms on/ 500ms off)으로, Yellow LED를 2초 간격으로, Green LED를 4초 간격으로 점멸하기.
- ▣ User button을 누르 때마다, 각 LED의 점멸 주기는 다음과 같이 변함.
  - 1초 간격 => 2초 간격 => 4초 간격 => 1초 간격 ...

## □ 회로 구성

- ▣ D3 핀에 Red LED와 저항,
  - ▣ D4 핀에 Green LED와 저항,
  - ▣ D5 핀에 Yellow LED와 저항 연결
- PortOut을 이용하여 구현.
- InterruptIn 을 이용하여 구현함.