

Infineon TC275 GPIO (General-Purpose IO)

Hyeongrae Kim
Architecture and Compiler for Embedded System LAB.
School of Electronics Engineering, KNU, KOREA

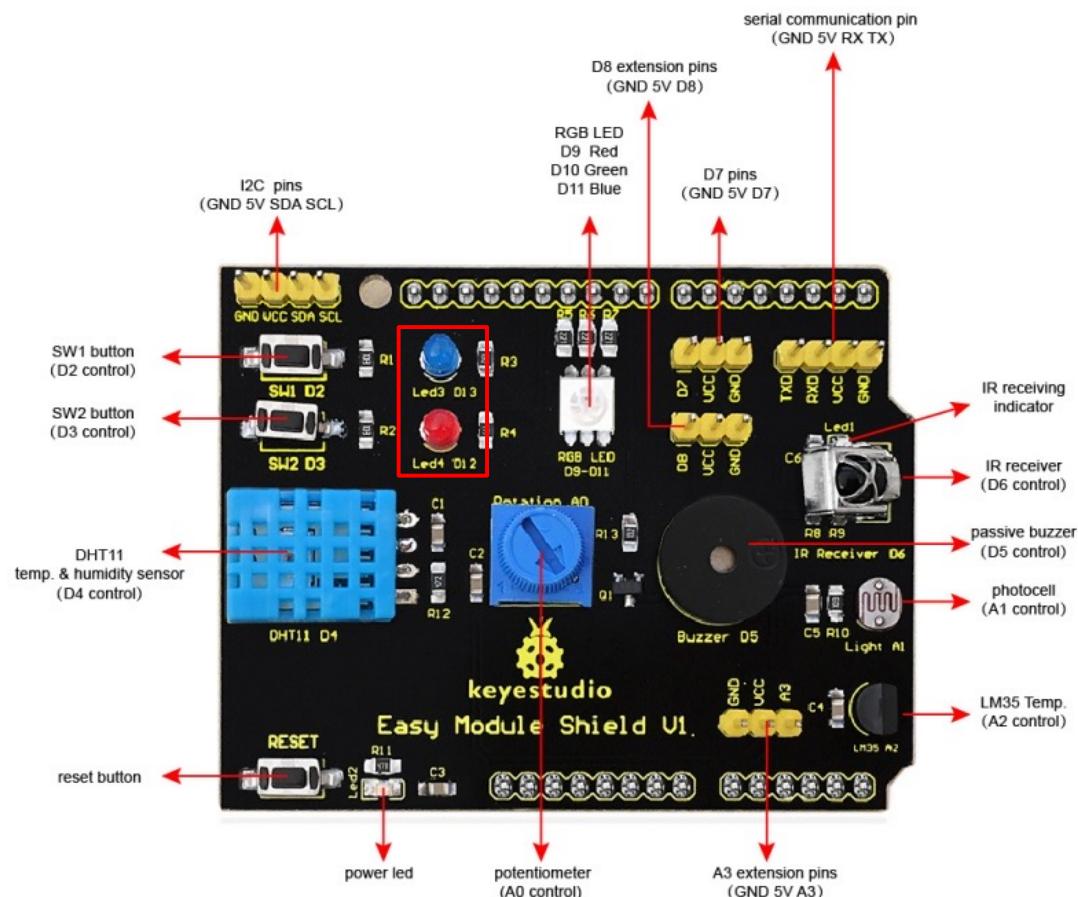
2021-05-11



LED Example

1. LED 연결 정보 파악

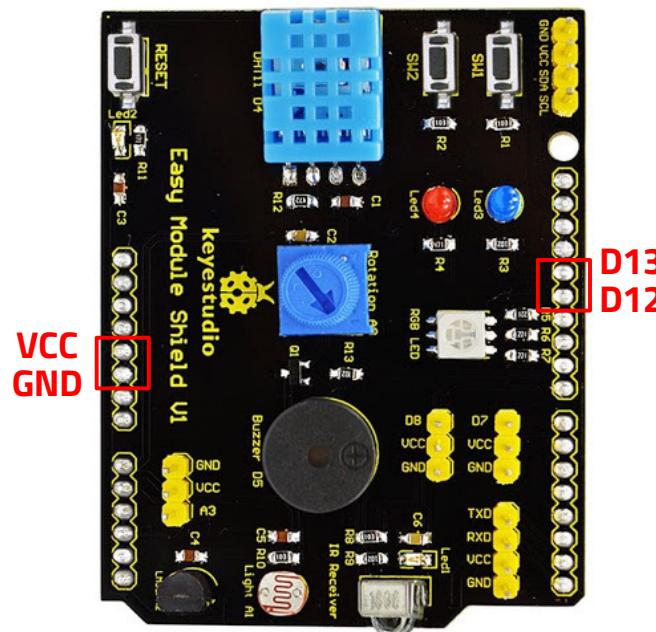
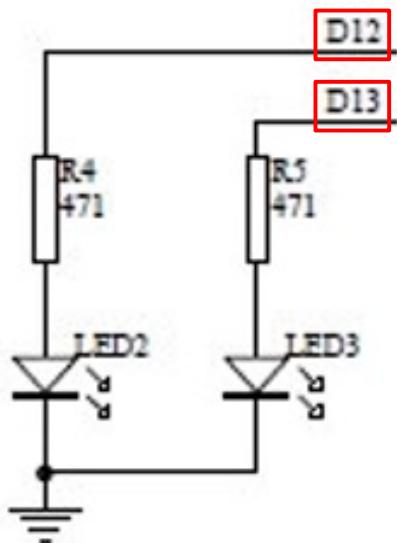
- ✓ 여러 LED를 사용하기 위해 Target Board가 아닌 **Easy Module Shield V1 확장 보드**의 LED를 사용한다.



LED Example

1. LED 연결 정보 파악

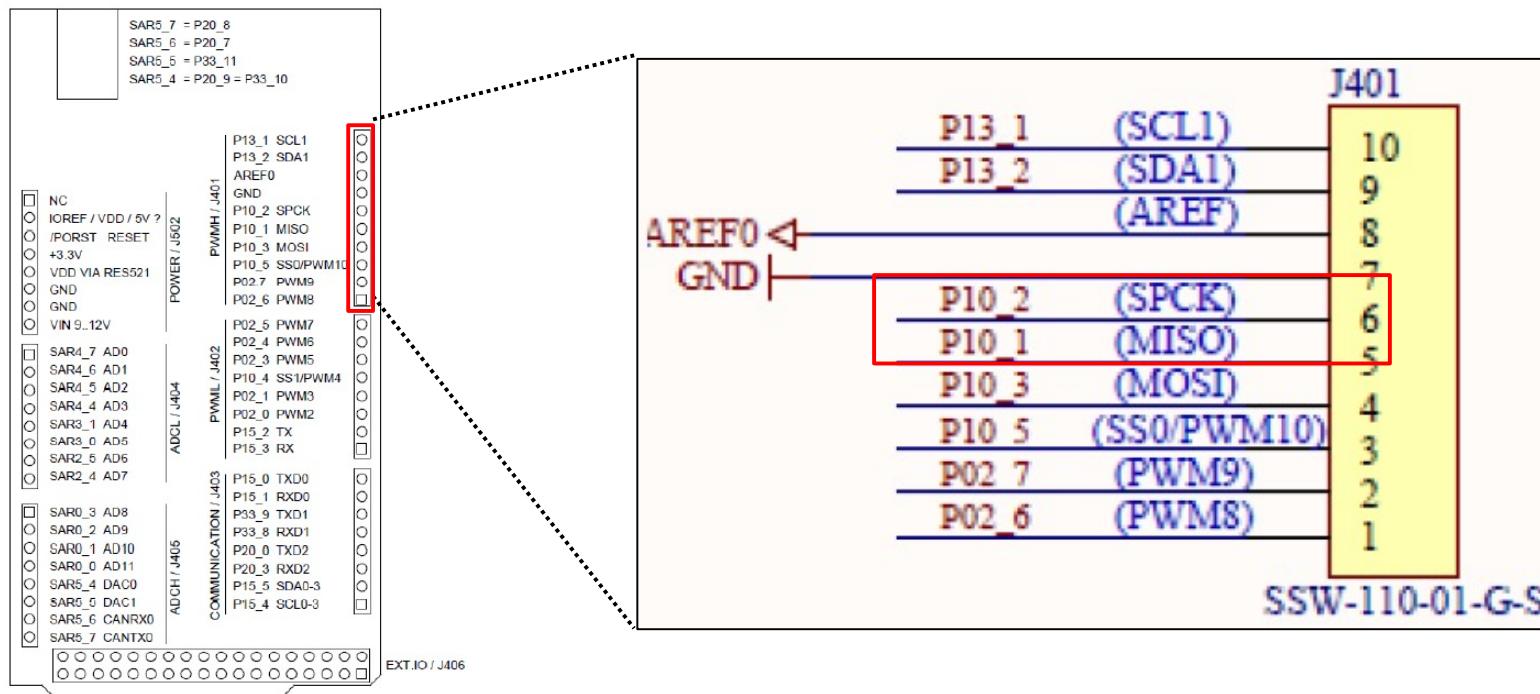
- ✓ LED는 Easy Module Shield V1 확장 보드의 **Pin D12(RED)/D13(BLUE)**과 연결되어 있다.
- ✓ 타겟 보드는 Easy Module Shield V1 확장 보드의 Pin D12/D13을 통해 LED 출력을 보낼 수 있다.
(정상적인 Switch 동작을 위해 VCC 및 GND도 연결해야 한다.)



LED Example

1. LED 연결 정보 파악

- ✓ TC275 보드의 Schematic과 Datasheet를 확인했을 때, Easy Module Shield V1 확장 보드의 Pin **D12/D13**과 연결되는 IO는 PORT10의 Pin 1-2다.
- ✓ 해당 Pin의 출력이 High-level 일 때 LED는 켜지고, Low-level 일 때 LED는 꺼진다.



LED Example

2. Data sheet 분석 : IO 설정

- ✓ LED RED를 사용하기 위해 연결된 Pin의 IO 설정이 필요하다.
- ✓ LED RED 제어를 위한 출력 신호를 보내내기 위해 해당 Pin을 **General-purpose output**으로 설정해야 한다.

Pin	Symbol	Ctrl	Type	Function
169	P10.1	I	MP+ / PU1 / VEXT	General-purpose input
	TIN103			GTM input
	MRST1A			QSPI1 input
	T5EUDB			GPT120 input
	P10.1			General-purpose output
	TOUT103			GTM output
	MTSR1			QSPI1 output
	MRST1			QSPI1 output
	EN01			MSC0 output
	VADCG6BFL1			VADC output
	END03			MSC0 output
	-			Reserved



LED Example

2. Data sheet 분석 : IO 설정

- ✓ LED BLUE를 사용하기 위해 연결된 Pin의 IO 설정이 필요하다.
- ✓ LED BLUE 제어를 위한 출력 신호를 내보내기 위해 해당 Pin을 **General-purpose output**으로 설정해야 한다.

170	P10.2	I	MP / PU1 / VEXT	General-purpose input
	TIN104			GTM input
	SCLK1A			QSPI1 input
	T6INB			GPT120 input
	REQ2			SCU input
	RXDCAN2E			CAN node 2 input
	SDI01			MSC0 input
	P10.2	O0		General-purpose output
	TOUT104	O1		GTM output
	-	O2		Reserved
	SCLK1	O3		QSPI1 output
	EN00	O4		MSC0 output
	VADCG6BFL2	O5		VADC output
	END02	O6		MSC0 output
	-	O7		Reserved



LED Example

2. Data sheet 분석 : PORT 설정 (1)

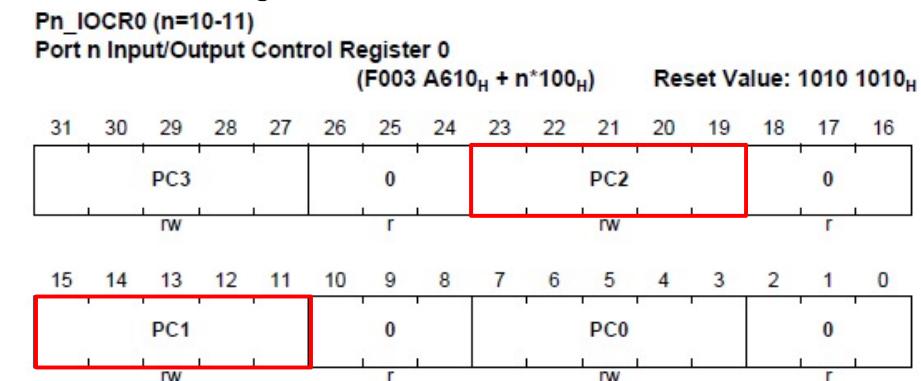
- ✓ P10_IOCR Register는 PORT10의 Input/Output을 설정한다.
- ✓ LED가 PORT10의 Pin 1-2에 연결되어 있기 때문에 **P10_IOCRO Register의 PC1, PC2 bits**를 설정한다.

Table 13-3 Registers Address Space

Module	Base Address	End Address	Note
P00	F003 A000 _H	F003 A0FF _H	13 pins
P01	F003 A100 _H	F003 A1FF _H	5 pins
P02	F003 A200 _H	F003 A2FF _H	12 pins
P10	F003 B000_H	F003 B0FF_H	9 pins
P11	F003 B100 _H	F003 B1FF _H	16 pins

P10_IOCRO Register 주소: F003_B010h (F003B000h + 10h)

P10_IOCRO Register 구조:



Field	Bits	Type	Description
PC0, PC1, PC2, PC3	[7:3], [15:11], [23:19], [31:27]	rw	Port Control for Port n Pin 0 to 3 This bit field determines the Port n line x functionality (x = 0-3) according to the coding table (see Table 13-5).
0	[2:0], [10:8], [18:16], [26:24]	r	Reserved Read as 0; should be written with 0.



LED Example

2. Data sheet 분석 : PORT 설정 (2)

- ✓ PORT10의 Pin 1-2를 General-purpose output (push-pull)으로 설정하기 위해 **PC1, PC2 bits**를 **10000b**로 설정한다.

Table 13-5 PCx Coding

PCx[4:0]	I/O	Characteristics	Selected Pull-up / Pull-down / Selected Output Function
10000 _B	Output	Push-pull	General-purpose output
10001 _B			Alternate output function 1
10010 _B			Alternate output function 2
10011 _B			Alternate output function 3
10100 _B			Alternate output function 4
10101 _B			Alternate output function 5
10110 _B			Alternate output function 6
10111 _B			Alternate output function 7
11000 _B	Open-drain		General-purpose output
11001 _B			Alternate output function 1
11010 _B			Alternate output function 2
11011 _B			Alternate output function 3
11100 _B			Alternate output function 4
11101 _B			Alternate output function 5
11110 _B			Alternate output function 6
11111 _B			Alternate output function 7



LED Example

2. Data sheet 분석 : PORT 출력 설정

- ✓ P10_OMR Register는 PORT10의 출력을 설정한다.
- ✓ PORT10의 Pin 1-2 출력을 설정하기 위해 **P10_OMR Register의 PCL1-2 bit와 PS1-2 bit**를 설정한다.
 - ✓ PCL1, PCL2 bit만 Set 하면 P10.1, P10.2 출력이 '0 (Low-level)'으로 Clear 된다.
 - ✓ PS1, PS2 bit만 Set 하면 P10.1, P10.2 출력이 '1 (High-level)'로 Set 된다.
 - ✓ PCL1, PCL2 bit와 PS1, PS2 bit를 동시에 Set 하면 P10.1, P10.2 출력이 Toggle 된다.

Table 13-3 Registers Address Space

Module	Base Address	End Address	Note
P00	F003 A000 _H	F003 A0FF _H	13 pins
P01	F003 A100 _H	F003 A1FF _H	5 pins
P02	F003 A200 _H	F003 A2FF _H	12 pins
P10	F003 B000_H	F003 B0FF_H	9 pins
P11	F003 B100 _H	F003 B1FF _H	16 pins
P12	F003 B200 _H	F003 B2FF _H	2 pins
P13	F003 B300 _H	F003 B3FF _H	4 pins
P14	F003 B400 _H	F003 B4FF _H	11 pins
P15	F003 B500 _H	F003 B5FF _H	9 pins

P10_OMR Register 주소: F003_B004_H (F003B000_H + 4_H)

P10_OMR Register 구조:

Pn_OMR (n=10-15) Port n Output Modification Register (F003 A604 _H + n*100 _H)																Reset Value: 0000 0000 _H
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16																
PCL 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																
W W W W W W W W W W W W W W W																
PS 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																
W W W W W W W W W W W W W W W																

Table 13-9 Function of the Bits PCLx and PSx

PCLx	PSx	Function
0	0	Bit Pn_OUT.Px is not changed.
0	1	Bit Pn_OUT.Px is set.
1	0	Bit Pn_OUT.Px is reset.
1	1	Bit Pn_OUT.Px is toggled.



LED Example

3. 프로그래밍

- 1) LED가 연결된 PORT10 Pin 1-2에 대한 IO 설정을 한다.

```
31 #define PORT10_BASE      (0xF003B000)
32 #define PORT10_IOCR0     (*(volatile unsigned int*)(PORT10_BASE + 0x10))
33 #define PORT10_OMR       (*(volatile unsigned int*)(PORT10_BASE + 0x04))
34
35 #define PC1              11
36 #define PC2              19
37
38 #define PCL1             17
39 #define PCL2             18
40 #define PS1              1
41 #define PS2              2
```

POR10 IO 설정관련 레지스터 주소 및 비트 필드 정의

```
45 /* Initialize LED (RED & BLUE) */
46 void init_LED(void)
47 {
48     /* Reset PC1 & PC2 in IOCR0*/
49     PORT10_IOCR0 &= ~((0x1F) << PC1);
50     PORT10_IOCR0 &= ~((0x1F) << PC2);
51
52     /* Set PC1 & PC2 with push-pull(2b10000) */
53     PORT10_IOCR0 |= ((0x10) << PC1);
54     PORT10_IOCR0 |= ((0x10) << PC2);
55 }
```

POR10 IO 설정 초기화 코드



LED Example

3. 프로그래밍

2) 동작에 따라 'main' 함수를 구현한다.

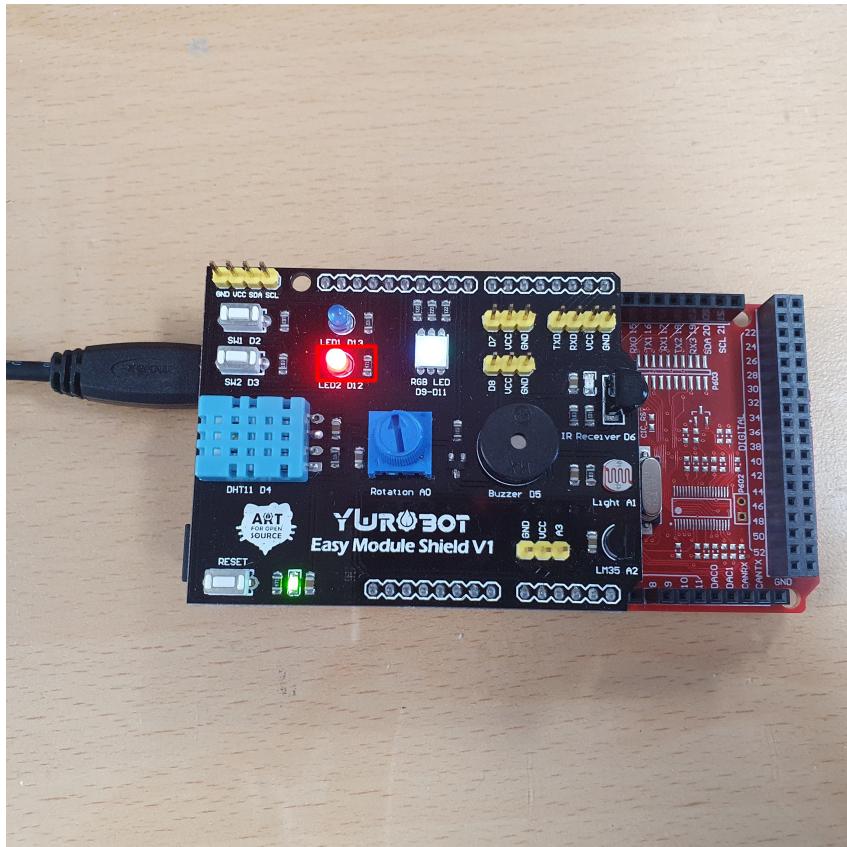
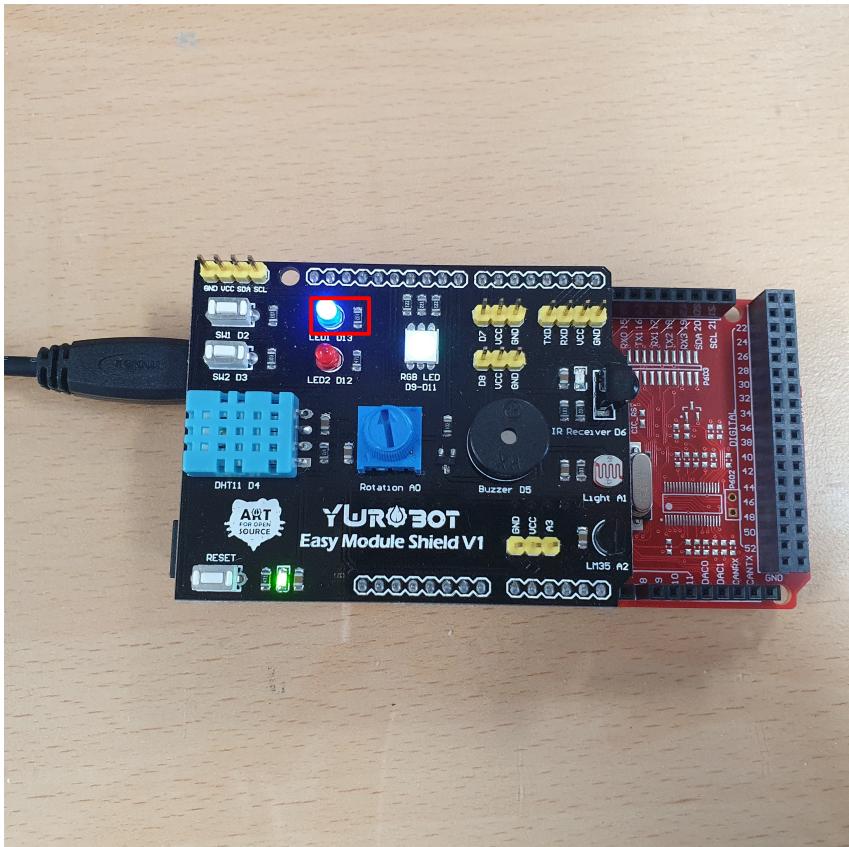
```
45 /* Initialize LED (RED & BLUE) */
46 void init_LED(void)
47 {
48     /* Reset PC1 & PC2 in IOCR0*/
49     PORT10_IOCR0 |= ~((0x1F) << PC1);
50     PORT10_IOCR0 |= ~((0x1F) << PC2);
51
52     /* Set PC1 & PC2 with push-pull(2b10000) */
53     PORT10_IOCR0 |= ((0x10) << PC1);
54     PORT10_IOCR0 |= ((0x10) << PC2);
55 }
56
57 int core0_main(void)
58 {
59     IfxCpu_enableInterrupts();
60
61     /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
62      * Enable the watchdogs and service them periodically if it is required
63      */
64     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
65     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
66
67     /* Wait for CPU sync event */
68     IfxCpu_emitEvent(&g_cpuSyncEvent);
69     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
70
71     volatile int cycle;
72
73     init_LED();
74
75     while(1)
76     {
77         PORT10_OMR |= ((1<<PCL1) | (1<<PS1));           // Toggle LED RED
78
79         for(cycle = 0; cycle < 5000000 ; cycle++);        // Delay
80
81         PORT10_OMR |= ((1<<PCL2) | (1<<PS2));           // Toggle LED BLUE
82
83         for(cycle = 0; cycle < 5000000 ; cycle++);        // Delay
84     }
85     return (1);
86 }
```



LED Example

4. 동작 확인

- ✓ Build 및 Debug 후 ('Resume' 버튼 클릭), LED 2개가 번갈아가며 깜빡이는 것을 확인한다.



Switch Example

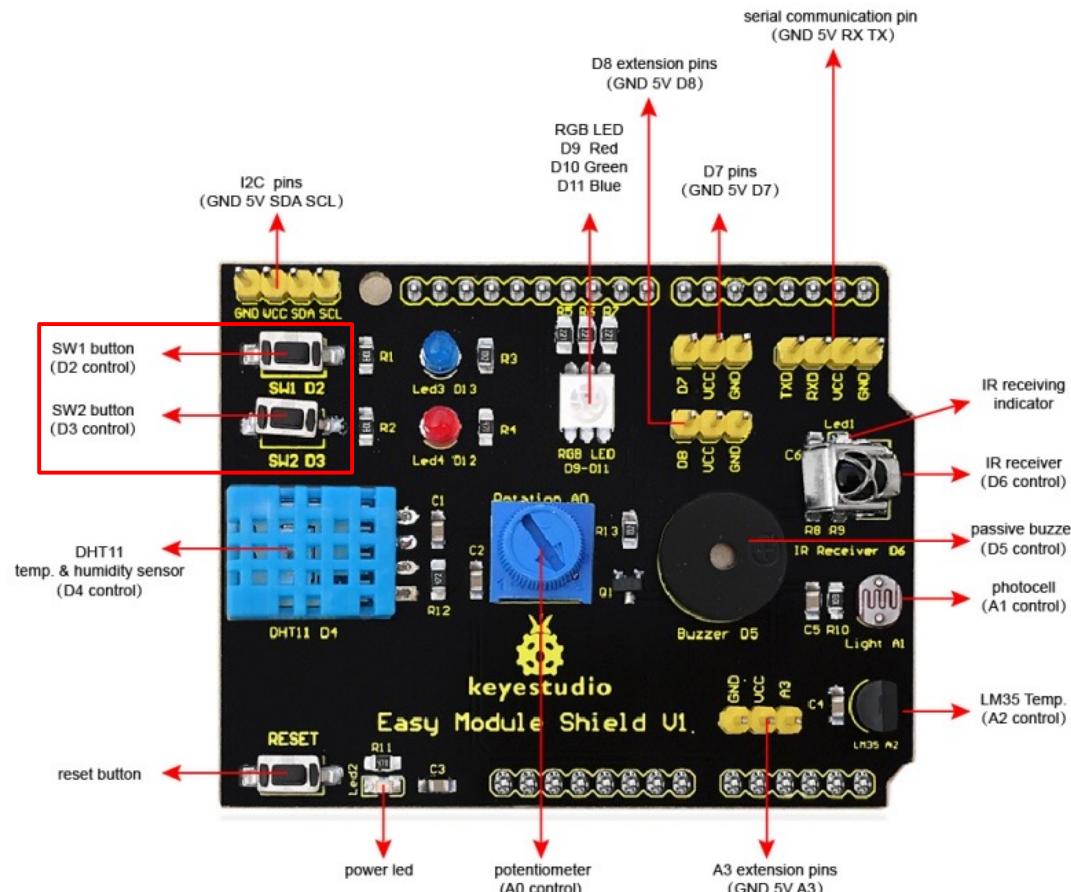
- Switch를 이용한 LED On/Off 제어
 1. 새로운 예제를 위한 프로젝트를 생성한다.
 2. 원하는 동작을 위해 레지스터와 메모리에 직접 접근해서 값을 써야한다.
 3. Switch 사용을 위해 Board Schematic과 Datasheet에서 Switch 연결 정보를 파악한다.
 4. Switch가 연결된 PORT의 메모리 맵을 분석한다.
 5. 분석 결과를 활용해 임베디드 프로그래밍을 한다.



Switch Example

1. Switch 연결 정보 파악

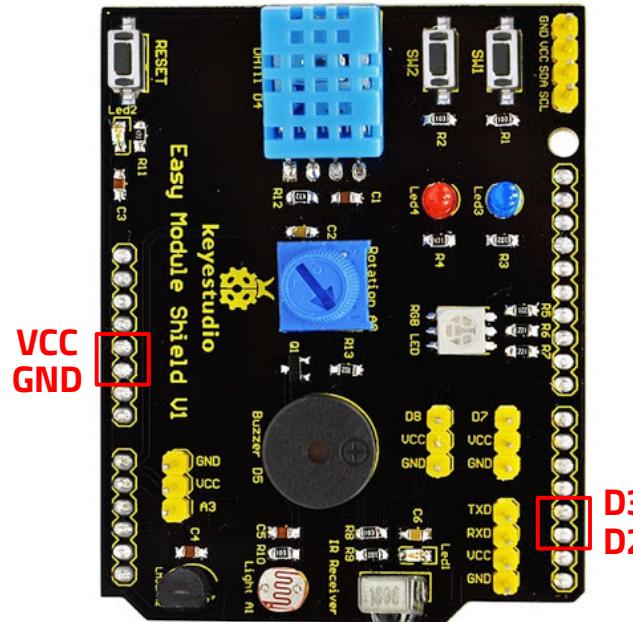
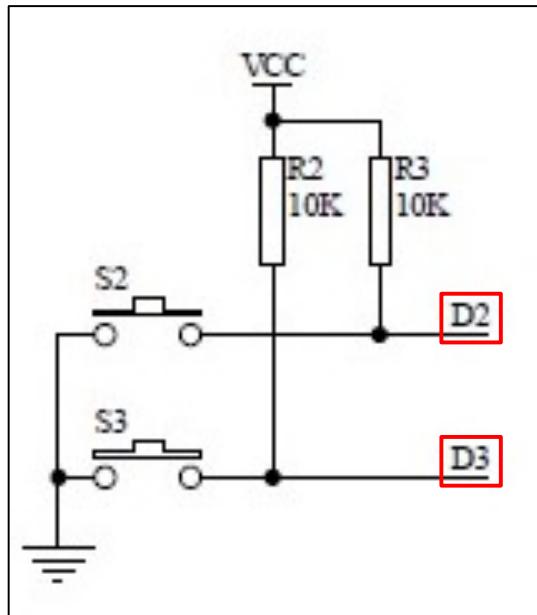
- ✓ 타겟 보드인 ShieldBuddy TC275에는 사용 가능한 Switch가 없기 때문에 **Easy Module Shield V1 확장 보드**의 Switch를 사용한다.



Switch Example

1. Switch 연결 정보 파악

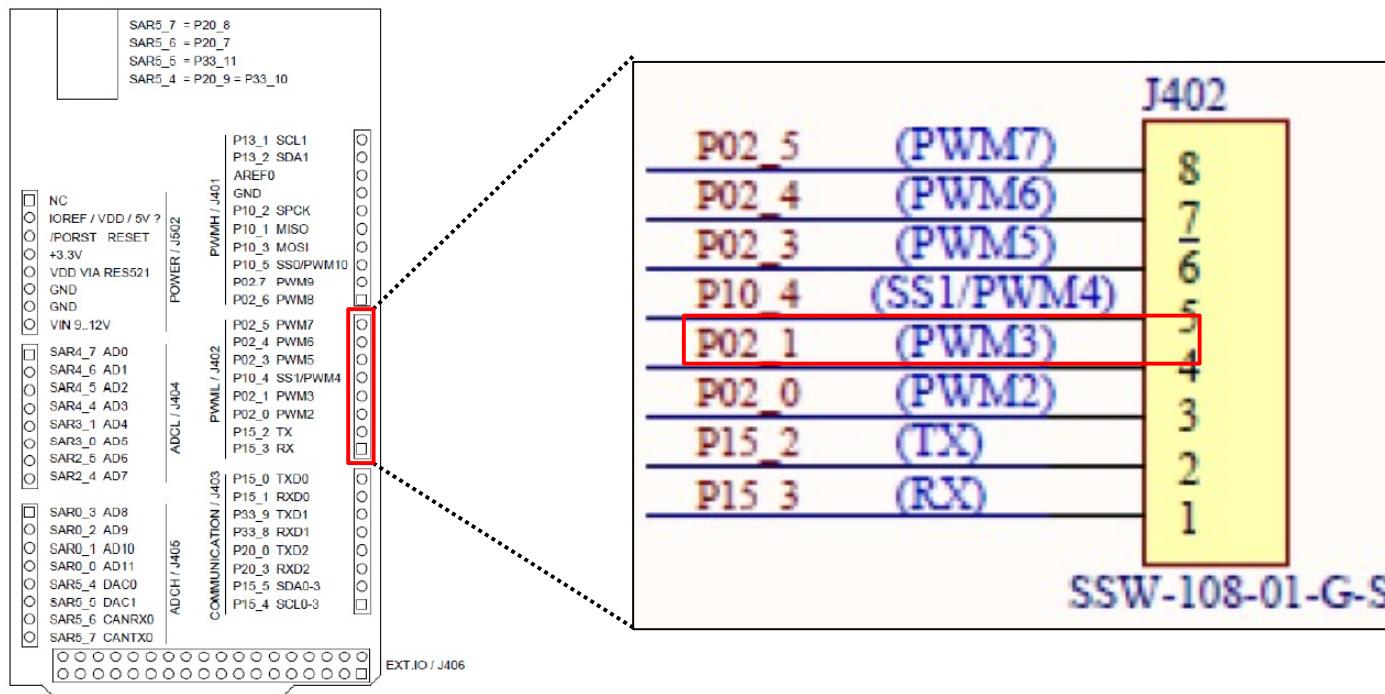
- ✓ Switch는 Easy Module Shield V1 확장 보드의 **Pin D2/D3**과 연결되어 있다.
- ✓ Switch가 눌리면 연결된 Pin은 Low-level이 되고, Switch가 눌리지 않으면 연결된 Pin은 High-level이 된다.
- ✓ 타겟 보드는 Easy Module Shield V1 확장 보드의 Pin D2/D3을 통해 Switch 입력을 받을 수 있다.
(정상적인 Switch 동작을 위해 VCC 및 GND도 연결해야 한다.)



Switch Example

1. Switch 연결 정보 파악

- ✓ TC275 보드의 Schematic과 Datasheet를 확인했을 때, Easy Module Shield V1 확장 보드의 **Pin D3**와 연결되는 IO는 **PORT02의 Pin 1**이다.



Switch Example

2. Data sheet 분석 : IO 설정

- ✓ Switch를 사용하기 위해 연결된 Pin의 IO 설정이 필요하다.
- ✓ Switch 신호를 입력 받기 위해 해당 Pin을 **General-purpose input**으로 설정해야 한다.

2	P02.1	I	LP / PU1 / VEXT	General-purpose input
	TIN1			GTM input
	REQ14			SCU input
	ARX2B			ASCLIN2 input
	RXDCAN0A			CAN node 0 input
	RXDA2			ERAY input
	CIFD1			CIF input
	P02.1	O0		General-purpose output
	TOUT1	O1		GTM output
	-	O2		Reserved
	SLSO32	O3		QSPI3 output
	DSCGPWMP	O4		DSADC output
	-	O5		Reserved
	-	O6		Reserved
	COUT60	O7		CCU60 output



Switch Example

2. Data sheet 분석 : PORT 설정 (1)

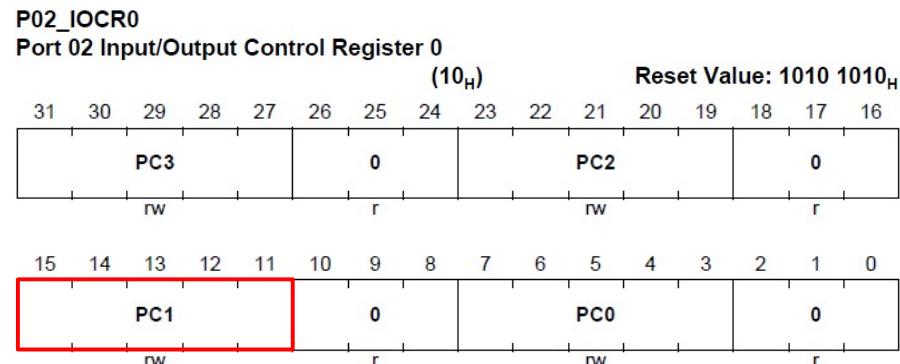
- ✓ P02_IOCR Register는 PORT02의 Input/Output을 설정한다.
- ✓ Switch가 PORT02의 Pin 1에 연결되어 있기 때문에 **P02_IOCRO Register의 PC1 bits**를 설정한다.

Table 13-3 Registers Address Space

Module	Base Address	End Address	Note
P00	F003 A000 _H	F003 A0FF _H	13 pins
P01	F003 A100 _H	F003 A1FF _H	5 pins
P02	F003 A200_H	F003 A2FF_H	12 pins
P10	F003 B000 _H	F003 B0FF _H	9 pins
P11	F003 B100 _H	F003 B1FF _H	16 pins
P12	F003 B200 _H	F003 B2FF _H	2 pins
P13	F003 B300 _H	F003 B3FF _H	4 pins
P14	F003 B400 _H	F003 B4FF _H	11 pins
P15	F003 B500 _H	F003 B5FF _H	9 pins

P02_IOCRO Register 주소: F003_A210h (F003A200h + 10h)

P02_IOCRO Register 구조:



Field	Bits	Type	Description
PC0, PC1, PC2, PC3	[7:3], [15:11], [23:19], [31:27]	rw	Port Control for Port n Pin 0 to 3 This bit field determines the Port n line x functionality (x = 0-3) according to the coding table (see Table 13-5).
0	[2:0], [10:8], [18:16], [26:24]	r	Reserved Read as 0; should be written with 0.

Switch Example

2. Data sheet 분석 : PORT 설정 (2)

- ✓ Easy Module Shield V1의 Switch는 pull-up device이다.
- ✓ 따라서, PORT02의 Pin 1을 General-purpose input으로 설정할 때 **PC1 bits**를 **0XX10b**로 설정한다.

Table 13-5 PCx Coding

PCx[4:0]	I/O	Characteristics	Selected Pull-up / Pull-down / Selected Output Function
$0XX00_B$	Input	–	No input pull device connected, tri-state mode
$0XX01_B$			Input pull-down device connected
$0XX10_B$			Input pull-up device connected ¹⁾
$0XX11_B$			No input pull device connected, tri-state mode



Switch Example

2. Data sheet 분석 : PORT 입력 확인

- ✓ P02_IN Register는 PORT02의 입력을 저장한다.
- ✓ PORT02의 Pin 1 입력을 확인하기 위해 **P02_IN Register**의 **P1 bit**를 확인한다.
 - ✓ 입력이 High-level인 경우, P1 bit는 '1'이다.
 - ✓ 입력이 Low-level인 경우, P1 bit는 '0'이다.

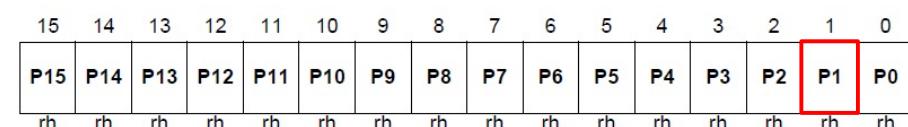
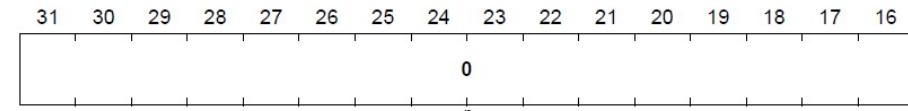
Table 13-3 Registers Address Space

Module	Base Address	End Address	Note
P00	F003 A000 _H	F003 A0FF _H	13 pins
P01	F003 A100 _H	F003 A1FF _H	5 pins
P02	F003 A200_H	F003 A2FF_H	12 pins
P10	F003 B000 _H	F003 B0FF _H	9 pins
P11	F003 B100 _H	F003 B1FF _H	16 pins
P12	F003 B200 _H	F003 B2FF _H	2 pins
P13	F003 B300 _H	F003 B3FF _H	4 pins
P14	F003 B400 _H	F003 B4FF _H	11 pins
P15	F003 B500 _H	F003 B5FF _H	9 pins

P02_IN Register 주소: F003_A224_H (F003A200_H + 24_H)

P02_IN Register 구조:

P0n_IN (n=0-2)
Port On Input Register (F003 A024_H + n*100_H) Reset Value: 0000 XXXX_H



Field	Bits	Type	Description
Px (x = 0-15)	x	rh	Port n Input Bit x This bit indicates the level at the input pin Pn.x. 0 _B The input level of Pn.x is 0. 1 _B The input level of Pn.x is 1.
0	[31:16]	r	Reserved Read as 0.



Switch Example

3. 프로그래밍

- 1) Switch가 연결된 PORT02 Pin 1에 대한 IO 설정을 한다.

```
40 /* Define PORT02 Registers for Switch2 */
41 #define PORT02_BASE      (0xF003A200)
42 #define PORT02_IOCR0     (*(volatile unsigned int*)(PORT02_BASE + 0x10))
43 #define PORT02_IN       (*(volatile unsigned int*)(PORT02_BASE + 0x24))
44
45 #define PC1             11
46 #define P1              1
```

POR02 IO 설정관련 레지스터 주소 및 비트 필드 정의

```
60 /* Initialize Switch2 */
61 void init_Switch(void)
62 {
63     /* Reset PC1 in IOCR0*/
64     PORT02_IOCR0 &= ~((0x1F) << PC1);
65
66     /* Set PC1 with push-pull(2b0xx10) */
67     PORT02_IOCR0 |= ((0x2) << PC1);
68 }
```

POR02 IO 설정 코드

Switch Example

3. 프로그래밍

2) 동작에 따라 'main' 함수를 구현한다. (LED 관련 부분은 이전 강의내용을 참고한다.)

```
31 /* Define PORT10 Registers for LED */
32 #define PORT10_BASE      (0xF003B000)
33 #define PORT10_IOCR0      (*(volatile unsigned int*)(PORT10_BASE + 0x10))
34 #define PORT10_OMR        (*(volatile unsigned int*)(PORT10_BASE + 0x04))
35
36 #define PC1              11
37 #define PCL1             17
38 #define PS1              1
39
40 /* Define PORT02 Registers for Switch2 */
41 #define PORT02_BASE      (0xF003A200)
42 #define PORT02_IOCR0      (*(volatile unsigned int*)(PORT02_BASE + 0x10))
43 #define PORT02_IN         (*(volatile unsigned int*)(PORT02_BASE + 0x24))
44
45 #define PC1              11
46 #define P1               1
```

```
50 /* Initialize LED (RED) */
51 void init_LED(void)
52 {
53     /* Reset PC1 in IOCR0*/
54     PORT10_IOCR0 &= ~((0x1F) << PC1);
55
56     /* Set PC1 with push-pull(2b10000) */
57     PORT10_IOCR0 |= ((0x10) << PC1);
58 }
59
60 /* Initialize Switch2 */
61 void init_Switch(void)
62 {
63     /* Reset PC1 in IOCR0*/
64     PORT02_IOCR0 &= ~((0x1F) << PC1);
65
66     /* Set PC1 with push-pull(2b0xx10) */
67     PORT02_IOCR0 |= ((0x2) << PC1);
68 }
```

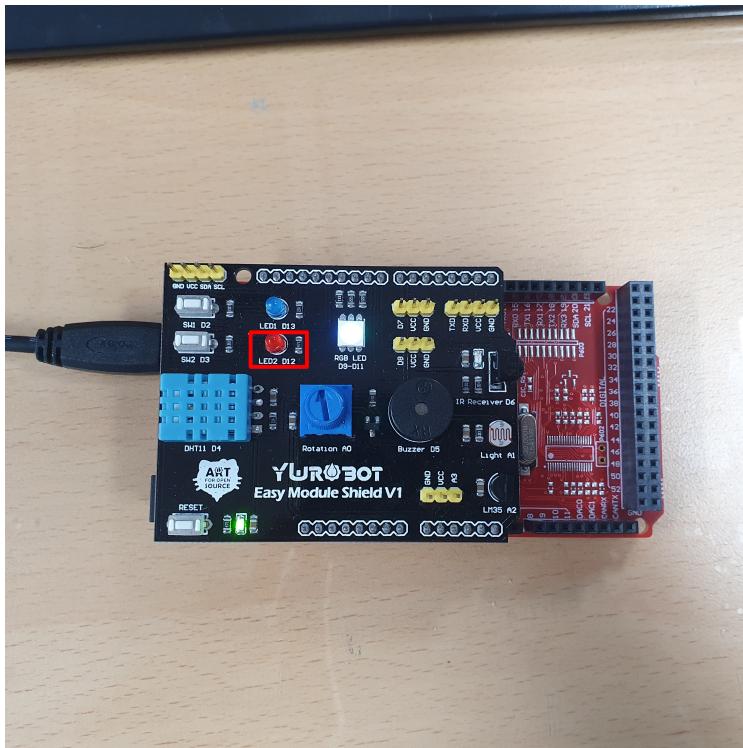
```
70 int core0_main(void)
71 {
72     IfxCpu_enableInterrupts();
73
74     /* !!!WATCHDOG AND SAFETY WATCHDOG ARE DISABLED HERE!!
75      * Enable the watchdogs and service them periodically if it is required
76      */
77     IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
78     IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
79
80     /* Wait for CPU sync event */
81     IfxCpu_emitEvent(&g_cpuSyncEvent);
82     IfxCpu_waitEvent(&g_cpuSyncEvent, 1);
83
84     init_LED();
85     init_Switch();
86
87     while(1)
88     {
89         if( (PORT02_IN & (1<<P1) ) == 0)           // Switch2 is pushed
90         {
91             PORT10_OMR |= (1<<PS1);                // LED RED on
92         }
93         else
94         {
95             PORT10_OMR |= (1<<PCL1);              // LED RED off
96         }
97     }
98 }  
99 }
```



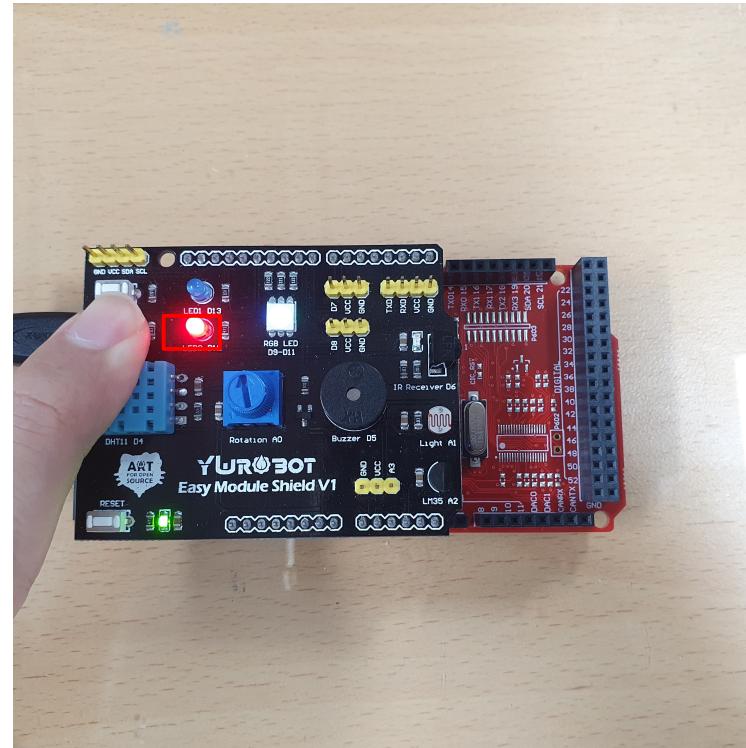
Switch Example

4. 동작 확인

- ✓ Build 및 Debug 후 ('Resume' 버튼 클릭), Switch를 누를 때 LED가 켜지는 것을 확인한다.



Switch is not pushed



Switch is pushed



Q & A

Thank you for your attention



Architecture and
Compiler
for Embedded Systems Lab.

School of Electronics Engineering, KNU

ACE Lab (hn02301@gmail.com)