

TI DSP, MCU, Xilinx Zynq FPGA ***기반의 프로그래밍 전문가 과정***

<펌웨어 프로그래밍>
2018.05.03 – 47일차

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 안상재
sangjae2015@naver.com

1. LED ON/OFF 프로그램(HAL Code Generator 코드 자동 생성기 사용)

```
#include "HL_sys_common.h"
#include "HL_gio.h"

int main(void)
{
    int i;
    gioInit();    // gio 레지스터 설정(불필요한 코드 다수 포함)

    while(1)
    {
        for(i=0;i<1000000;i++);        // delay 시간 동안 대기(led를 깜박이게 하기 위함)
        gioSetPort(gioPORTA, 0X10);    // gioPORTA의 4번째 bit를 1로 set
        for(i=0;i<1000000;i++);
        gioSetPort(gioPORTA, 0x00);    // gioPORTA의 4번째 bit를 0으로 set
    }
    return 0;
}
```

1-1. gioInit() 함수

```
void gioInit(void)
{
    /* USER CODE BEGIN (2) */
    /* USER CODE END */

    /** bring GPIO module out of reset */
    gioREG->GCR0 = 1U;
    gioREG->ENACLR = 0xFFU;
    gioREG->LVLCLR = 0xFFU;

    /** @b initialize @b Port @b A */

    /** - Port A output values */
    gioPORTA->DOUB = (uint32)((uint32)0U << 0U) /* Bit 0 */
        | (uint32)((uint32)0U << 1U) /* Bit 1 */
        | (uint32)((uint32)0U << 2U) /* Bit 2 */
        | (uint32)((uint32)0U << 3U) /* Bit 3 */
        | (uint32)((uint32)0U << 4U) /* Bit 4 */
        | (uint32)((uint32)0U << 5U) /* Bit 5 */
        | (uint32)((uint32)0U << 6U) /* Bit 6 */
        | (uint32)((uint32)0U << 7U); /* Bit 7 */

    /** - Port A direction */
    gioPORTA->DIR = (uint32)((uint32)0U << 0U) /* Bit 0 */
        | (uint32)((uint32)0U << 1U) /* Bit 1 */
        | (uint32)((uint32)0U << 2U) /* Bit 2 */
        | (uint32)((uint32)0U << 3U) /* Bit 3 */
        | (uint32)((uint32)1U << 4U) /* Bit 4 */
        | (uint32)((uint32)0U << 5U) /* Bit 5 */
        | (uint32)((uint32)0U << 6U) /* Bit 6 */
        | (uint32)((uint32)0U << 7U); /* Bit 7 */

    /** - Port A open drain enable */
    gioPORTA->PDR = (uint32)((uint32)0U << 0U) /* Bit 0 */
        | (uint32)((uint32)0U << 1U) /* Bit 1 */
        | (uint32)((uint32)0U << 2U) /* Bit 2 */
        | (uint32)((uint32)0U << 3U) /* Bit 3 */
        | (uint32)((uint32)0U << 4U) /* Bit 4 */
        | (uint32)((uint32)0U << 5U) /* Bit 5 */
        | (uint32)((uint32)0U << 6U) /* Bit 6 */
}
```

```

        | (uint32)((uint32)0U << 7U); /* Bit 7 */

/** - Port A pullup / pulldown selection */
gioPORTA->PSL = (uint32)((uint32)0U << 0U) /* Bit 0 */
        | (uint32)((uint32)0U << 1U) /* Bit 1 */
        | (uint32)((uint32)0U << 2U) /* Bit 2 */
        | (uint32)((uint32)0U << 3U) /* Bit 3 */
        | (uint32)((uint32)0U << 4U) /* Bit 4 */
        | (uint32)((uint32)0U << 5U) /* Bit 5 */
        | (uint32)((uint32)0U << 6U) /* Bit 6 */
        | (uint32)((uint32)0U << 7U); /* Bit 7 */

/** - Port A pullup / pulldown enable*/
gioPORTA->PULDIS = (uint32)((uint32)0U << 0U) /* Bit 0 */
        | (uint32)((uint32)0U << 1U) /* Bit 1 */
        | (uint32)((uint32)0U << 2U) /* Bit 2 */
        | (uint32)((uint32)0U << 3U) /* Bit 3 */
        | (uint32)((uint32)0U << 4U) /* Bit 4 */
        | (uint32)((uint32)0U << 5U) /* Bit 5 */
        | (uint32)((uint32)0U << 6U) /* Bit 6 */
        | (uint32)((uint32)0U << 7U); /* Bit 7 */

```

1-2. gpioSetPort(gioPORT_t *port, uint32 value)

```

void gpioSetPort(gioPORT_t *port, uint32 value)
{
    port->DOUT = value;
}

```

2. LED ON/OFF 프로그램 (최적화 프로그램)

- 코드를 최적화를 함으로써 불필요한 코드가 제거되고 성능이 향상된다.

```

#include "HL_sys_common.h"
#include "HL_gio.h"

int main(void)
{
    int i;
    gioREG->GCR0 = 1U; // GIO module의 레지스터를 설정하기 위해 GIOGCR0 레지스터를 1로 set함
    gioPORTA->DIR = 0x10; // gioPORTA 의 4번째 bit의 방향을 OUT으로 설정
4
    while(1)
    {
        for(i=0;i<1000000;i++);
        gioPORTA->DOUT = 0x10; // gioPORTA 의 4번째 bit를 1로 set
        for(i=0;i<1000000;i++);
        gioPORTA->DOUT = 0x00; // gioPORTA 모두 0으로 set
    }
    return 0;
}

```

* 결과 사진

