

Xilinx

Zynq FPGA

TI DSP MCU 기반의

프로그래밍 및 회로 설계 전문가

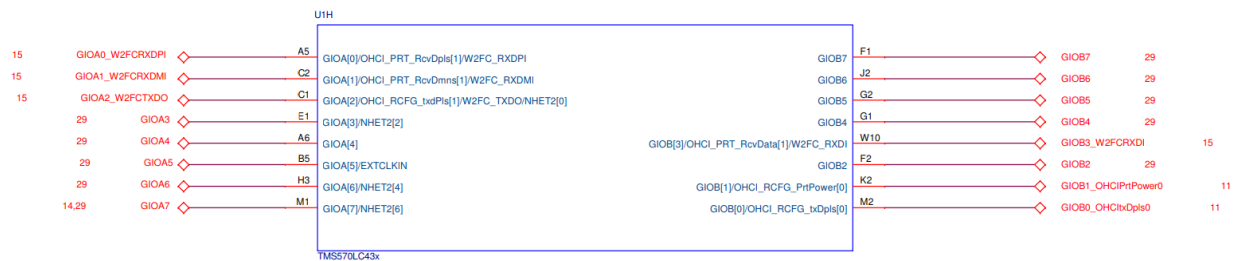
강사 이상훈

gcccompil3r@gmail.com



학생 김민호

minking12@naver.com



GIOA[4]

4 번핀을 이용하여 LED 센서 작동하기.

```
int main(void)
{
```

```
gioInit();
gioSetDirection(gioPORTA, 0xffffffff);
gioSetPort(gioPORTA, 0xffffffff);
gioSetBit(gioPORTA, 4, 1);
gioPOL(gioPOTRA, 0xffefffff)
```

```
while(1);
```

```
return 0;
}
```

gioInit 을 F3 으로 들어가보면

DIR → 출력인지 입력인지.

DOUT → 출력 (3.3 v)

DSET → 해당되는 비트를 출력으로 바꾼다.

DCLR → 출력중인 핀을 0 으로 바꿔서 출력을 멈춘다.

... 이 있다.

오늘 너무 티레드해서 메뉴얼 캡처를 못하겠습니다 ...

DOUT 연산법

그냥 삽입.

DSET 연산법

1 일때 1 , 0 이면 기존 저장되어 있는 값.

DCLR 연산법

1 일때 0 으로 , 0 이면 기존 저장되어 있는 값.

HAL Code Generator - C:\WEEWExternal_LED\External_LED.hcg - [GIO]

File Edit View Tools Window Help

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3

Device Explorer

TMS570LC4357ZWT

SYSTEM

- HL_hal_stdtypes.h
- HL_sys_common.h
- HL_reg_system.h
- HL_reg_flash.h
- HL_reg_l2ramw.h
- HL_reg_vim.h
- HL_reg_pbist.h
- HL_reg_stc.h
- HL_reg_efc.h
- HL_reg_pcr.h
- HL_reg_pmm.h
- HL_reg_dma.h
- HL_reg_ccmr5.h
- HL_sys_core.h
- HL_system.h
- HL_sys_vim.h
- HL_sys_mpu.h
- HL_sys_pmu.h
- HL_sys_pcr.h
- HL_sys_pmm.h
- HL_sys_dma.h
- HL_sys_core.asm
- HL_sys_intvecs.asm
- HL_sys_mpu.asm
- HL_sys_pmu.asm
- HL_sys_pcr.c
- HL_sys_pmm.c
- HL_sys_dma.c
- HL_system.c
- HL_sys_phantom.c
- HL_sys_startup.c
- HL_sys_vim.c
- HL_sys_main.c
- HL_notification.c
- HL_sys_link.cmd
- HL_reg_epc.h
- HL_reg_nmpu.h
- HL_reg_scm.h
- HL_reg_sdcmmr.h
- HL_epc.h
- HL_epc.c
- HL_nmpu.h
- HL_nmpu.c
- HL_errata.h
- HL_errata.c

ESM

Device Explorer File Explorer

Read CAP OVR

Port A Port B

High Priority: Enable: Rising Edge: Falling Edge: VIM: Low Priority:

Bit 3

DOUT: 0 DIR: PDR: PSL: GIOA[3]

DIN:

High Priority: Enable: Rising Edge: Falling Edge: VIM: Low Priority:

Bit 4

DOUT: 0 DIR: PDR: PSL: GIOA[4]

DIN:

High Priority: Enable: Rising Edge: Falling Edge: VIM: Low Priority:

Bit 5

DOUT: 0 DIR: PDR: PSL: GIOA[5]

DIN:

High Priority: Enable: Rising Edge: Falling Edge: VIM: Low Priority:

Bit 6

DOUT: 0 DIR: PDR: PSL: GIOA[6]

DIN:

High Priority: Enable: Rising Edge: Falling Edge: VIM: Low Priority:

Output

EQEP
FEE
AJSM
Code complete

For Help, press F1

```
Int main(void)
```

```
{
```

```
gioREG->GCR0 = 1U;
```

```
gio 핀을 리셋한다. 1u -> 리셋
```

```
gioREG->ENACLR= 0xFFU;
```

```
특정핀의 인터럽트를 끈다.
```

```
gioREG->LVLCLR= 0xFFU;
```

```
인터럽트를 최하위로 낮춘다.
```

```
gioPORTA->DIR=(uint32)((uint32)1U << 4U)
```

```
4 번비트를 출력으로 설정. 1U 가 작동
```

```
gioPORTA->DOUT=(uint32)((uint32)1U << 4U)
```

```
4 번 핀에 3.3v 출력.
```

```
while(1);
```

```
return 0;
```

```
}
```