

# TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

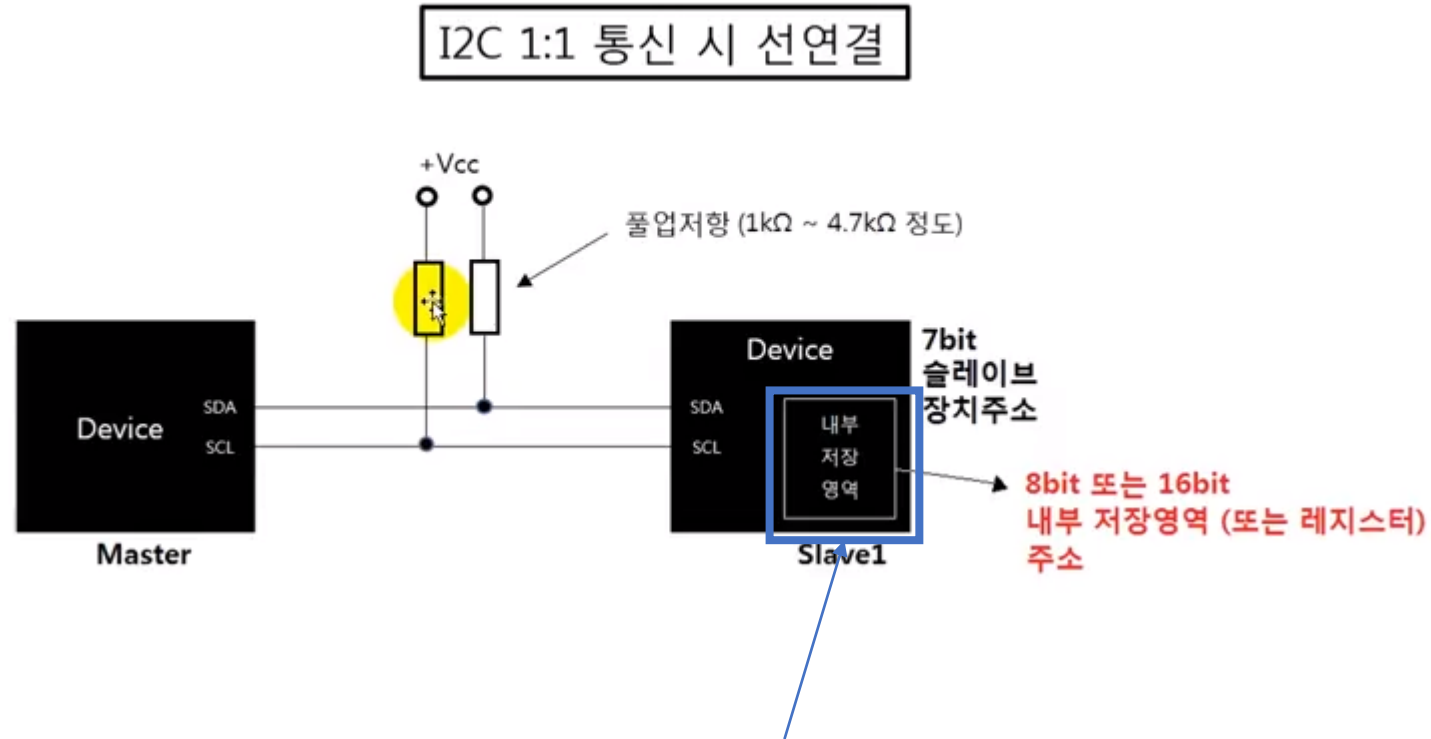
## I2C Interface - EEPROM

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

학생 – 안상재  
sangjae2015@naver.com

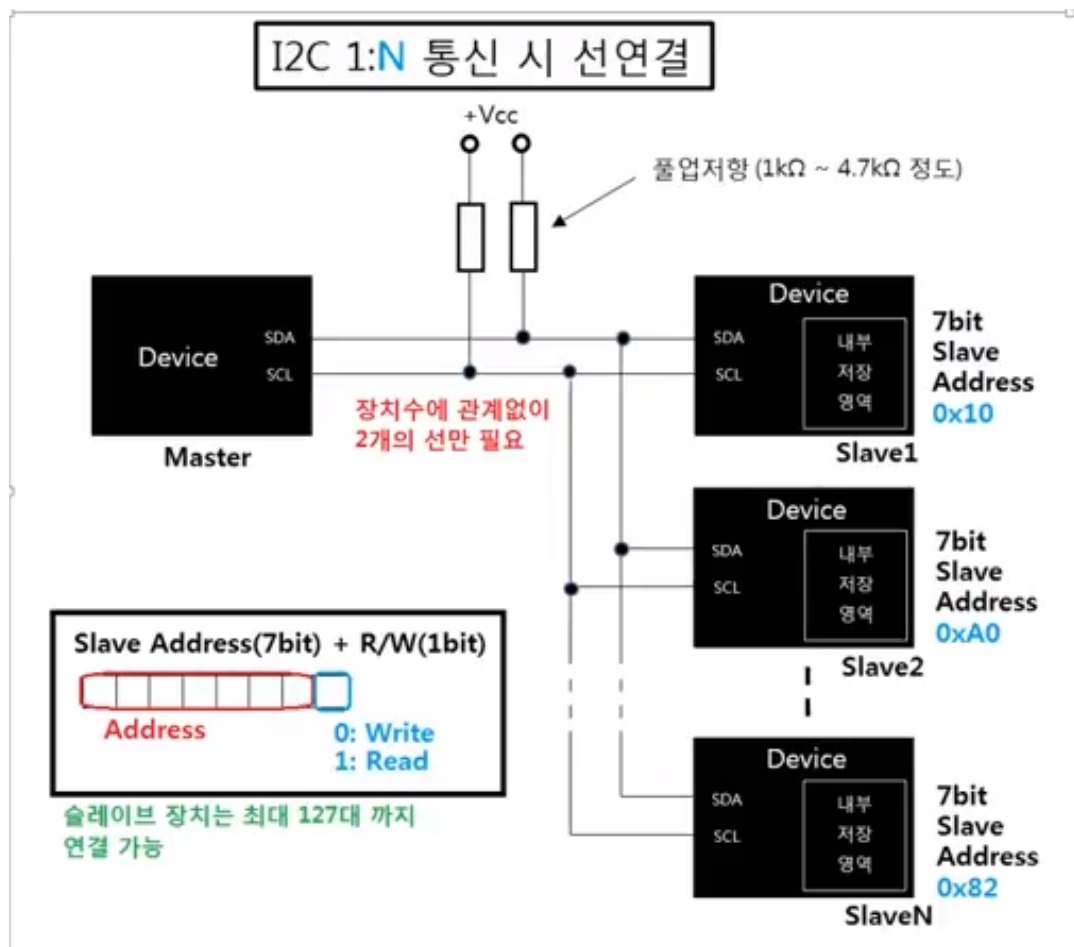
## 1. I2C 통신 연결

- I2C 통신 1:1 연결

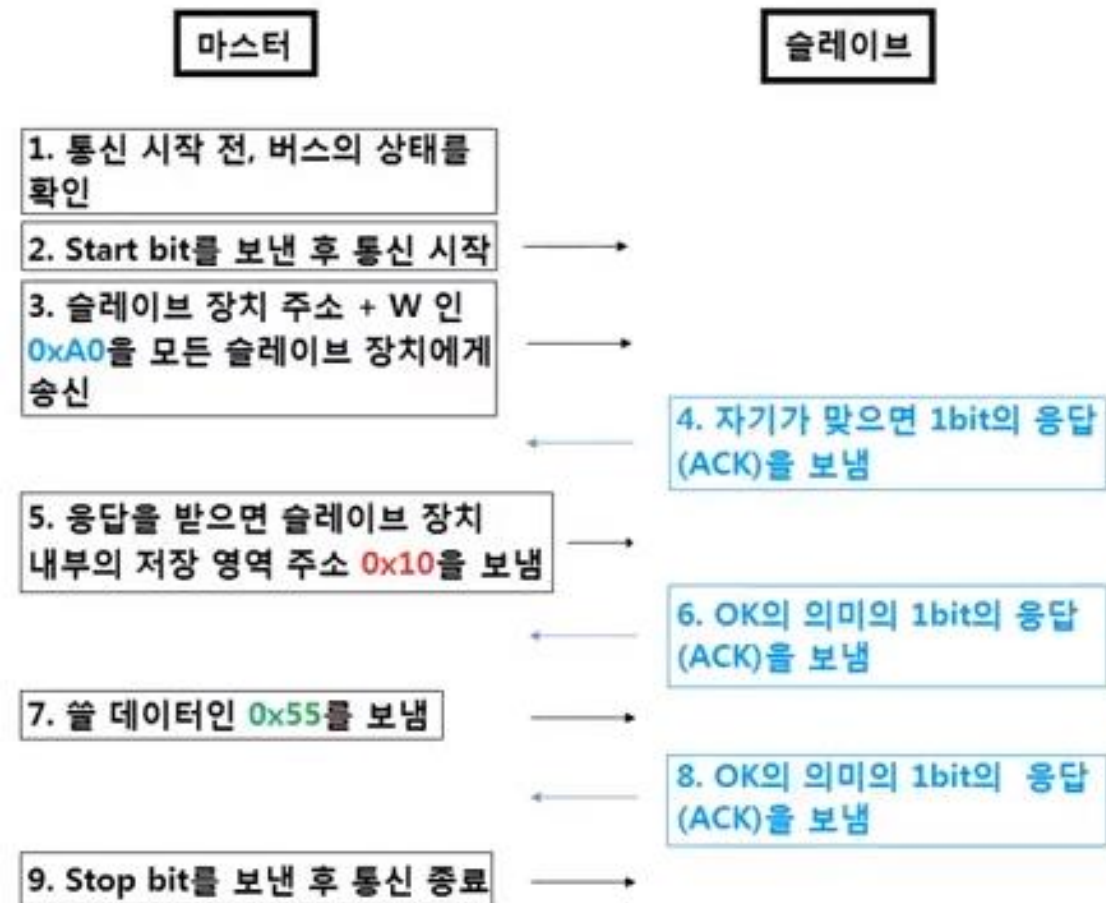


슬레이브의 내부 메모리를 접근해서 write 또는 read함!

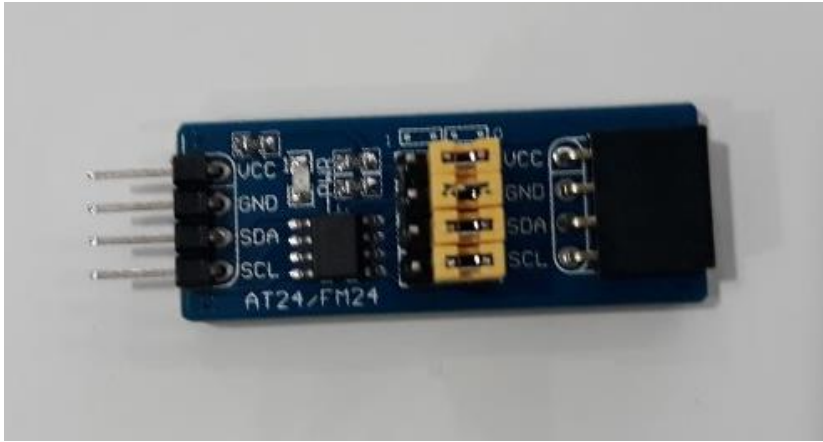
## - I2C 통신 1:N 연결



ex) 0xA0 주소를 가진 슬레이브 내부의 0x10 저장영역에 0x55라는 데이터를 쓰기



## 2. I2C 통신 모듈 (EEPROM)



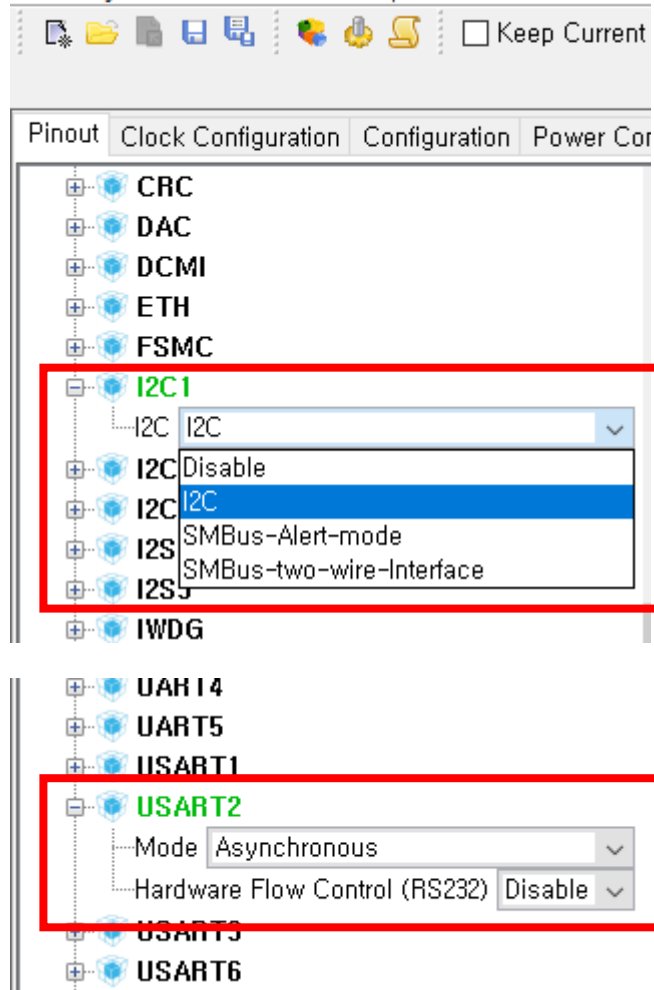
**AT24CXX EEPROM Board**

<https://www.eleparts.co.kr/goods/view?no=436527>

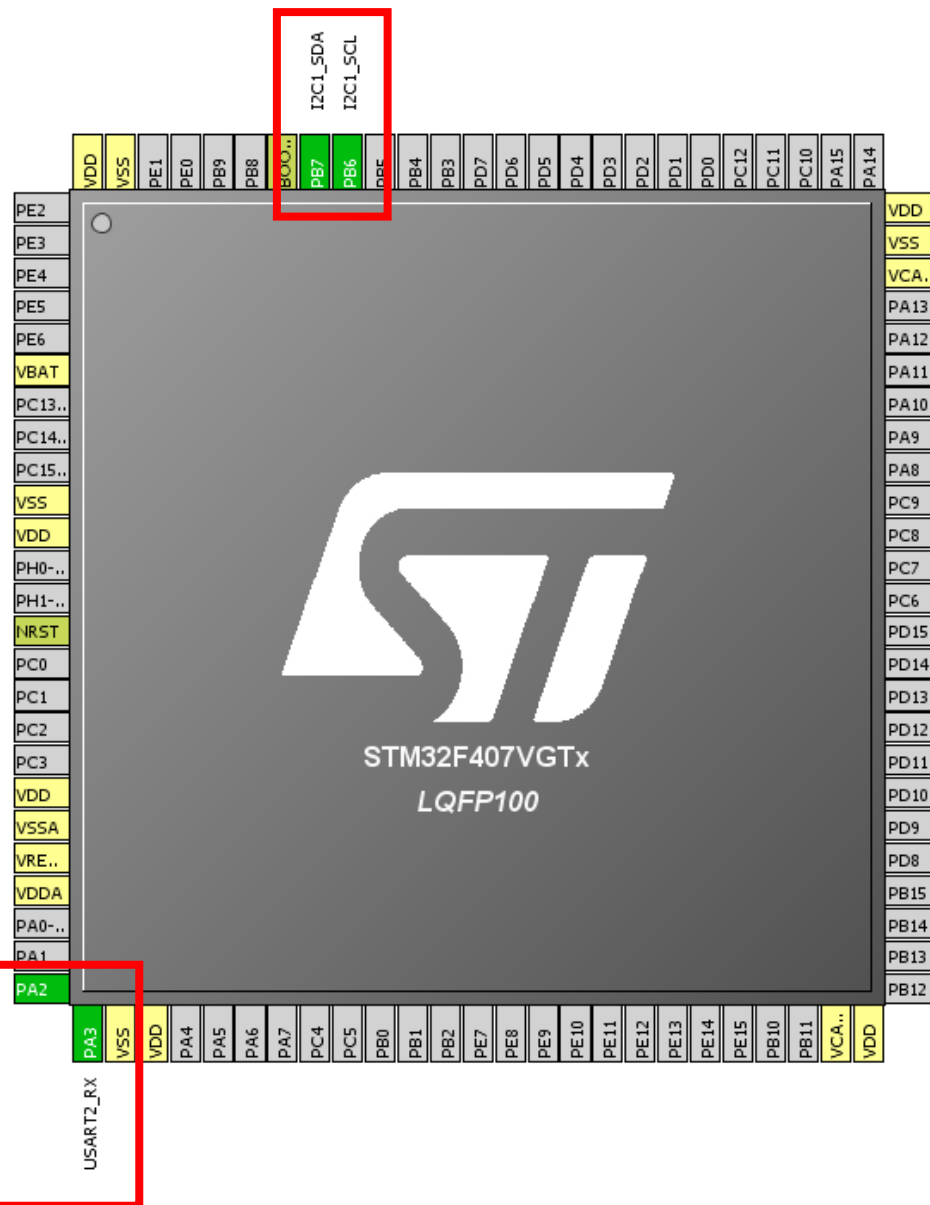
### 3. CubeMX 설정

STM32CubeMX I2C\_EEPROM.ioc: STM32F407VGTx

File Project Pinout Window Help

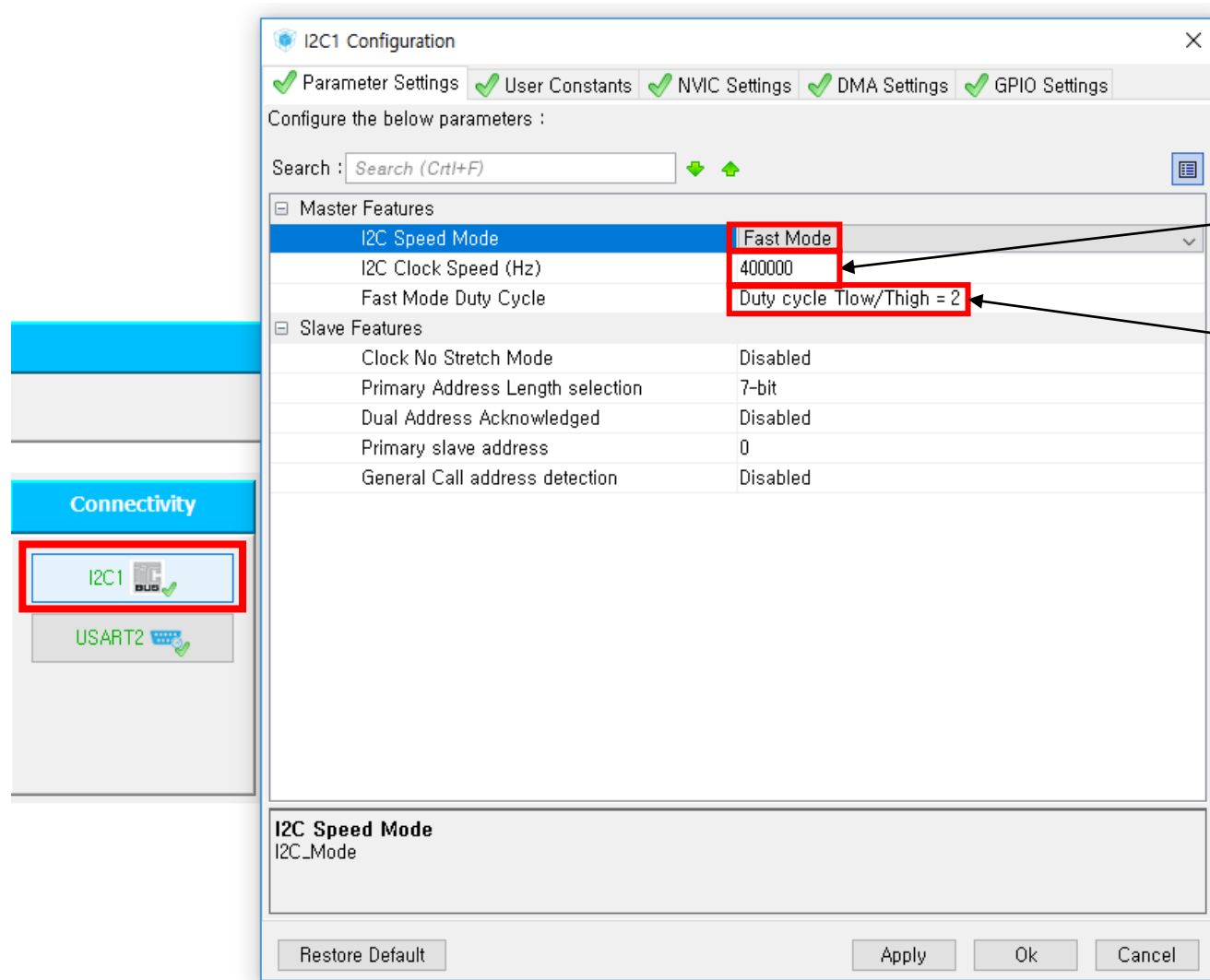


I2C 핀



USART2 (I2C 통신 확인)

## Configuration 탭



통신 속도 : 400KHz

High가 low 보다 2배 길게 Duty비 설정  
통상 이 정도 비율로 하면 문제 없음

## 4. 소스 코드

`#define EEPROM_ADDR 0xA0` ← AT24CXX EEPROM 모듈의 주소

EEPROM에 write할 데이터

```
int main(void)
{
    /* USER CODE BEGIN 1 */
    uint8_t eeprom[10] = {0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99};
    uint8_t data[10] = {0};
    uint8_t str[30], i;
    /* USER CODE END 1 */
```

EEPROM로부터 read한 데이터를 저장할 변수

main 문 안의 while문 위

EEPROM의 0x55번지에 eeprom 배열의 10개 데이터를 write함

```
HAL_I2C_Mem_Write(&hi2c1, EEPROM_ADDR, 0x55, I2C_MEMADD_SIZE_8BIT, &eeprom[0], 10, 10);
HAL_Delay(1000);
```

EEPROM에 데이터가 write되려면 시간이 필요하므로 1초 대기함!

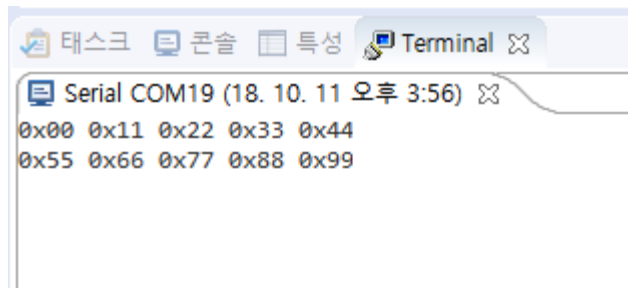
```
HAL_I2C_Mem_Read(&hi2c1, EEPROM_ADDR, 0x55, I2C_MEMADD_SIZE_8BIT, &data[0], 10, 10);
```

0x55번지로부터 데이터를 10개 read함

```
sprintf(str, "0x%02x 0x%02x 0x%02x 0x%02x 0x%02x\r\n", data[0], data[1], data[2], data[3], data[4]);
HAL_UART_Transmit(&huart2, str, sizeof(str), 10);

sprintf(str, "0x%02x 0x%02x 0x%02x 0x%02x 0x%02x\r\n\n", data[5], data[6], data[7], data[8], data[9]);
HAL_UART_Transmit(&huart2, str, sizeof(str), 10);
```

## 4. 결과 화면



- EEPROM에 write한 데이터가 그대로 read 되는 것을 확인할 수 있다.
- EEPROM은 비휘발성 메모리이기 때문에 전원을 껐다가 다시 read해도 같은 값이 출력된다.