

I2C Interface

PCF8591 ADC/DAC 모듈 (가변저항, 조도센서, 온도센서)

I2C 인터페이스

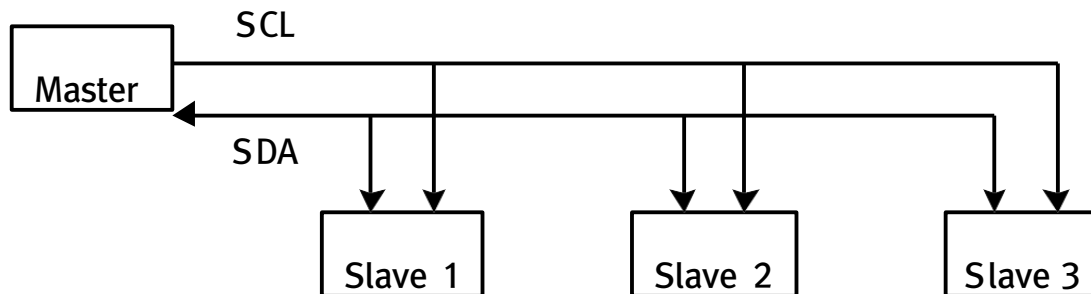
* 라즈베리파이 보드

- 라즈베리파이의 GPIO는 디지털 입출력만 가능하며, ADC도 제공되지 않음
- 아날로그 신호원의 디바이스를 취급하려면 ADC필요
- ADC/DAC 모듈은 I²C 통신으로 동작

I2C 인터페이스

* I2C(Inter integrated Circuit)

- IC들 간의 데이터 통신 방식
- 2개 신호선 만을 사용하여 통신토록 정의 SDA(Serial data), SCL(Serial Clock)
- 회로 간단, 저 비용, 저 잡음 등의 장점
- 송/수신측은 하나의 master와 다수의 slave로 구성



I2C 인터페이스

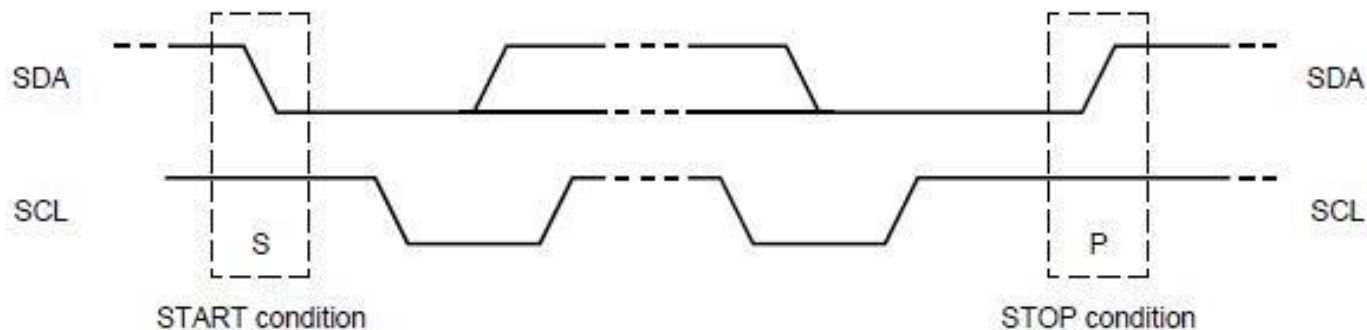
* Start 및 Stop 조건

- 비 통신시 DAT, SCL에 High 신호 유지(IDLE 상태)
- 통신을 위해 반드시 시작 조건으로 개시
- 두 조건사이의 전송 데이터 바이트 수는 제한 없음
- 시작(Start) 조건 : master가 slave에게 통신 개시의 의미 상태

SCL은 High 상태인 동안, SDA는 High-to-Low 천이

- 종료(Stop) 조건 : master가 slave에게 통신 종료의 의미 상태

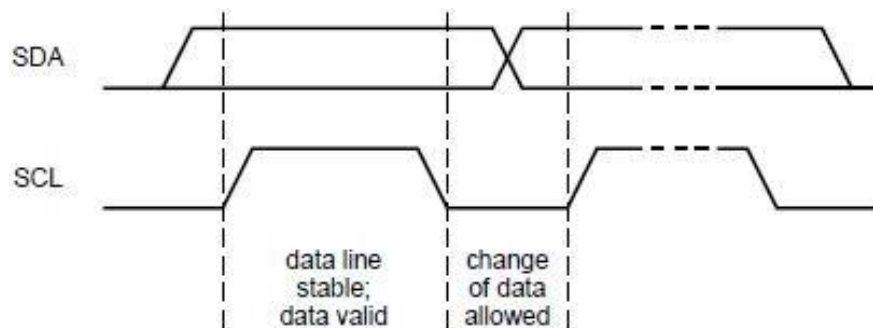
SCL은 High 상태인 동안, SDA는 Low-to-High 천이



I2C 인터페이스

* 데이터 비트 전송

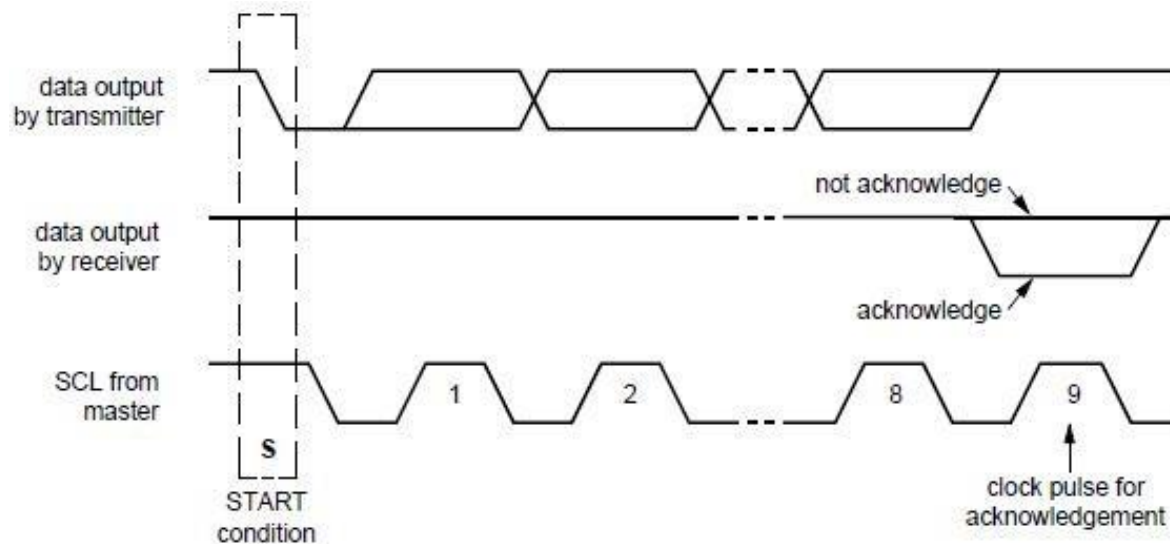
- 하나의 데이터 비트는 매 클럭펄스(SCL이 High) 동안 전송
- SDA에 데이터 비트는 SCL이 High인 동안 안정적으로 유지되어야 함
(SCL이 High인 동안 SDA 신호가 안정하도록 유지)



I2C 인터페이스

* ACK 신호 전송

- 전송되는 데이터는 바이트마다 상대방으로부터 ACK 신호가 따라와야 해야 함.
- 한 바이트 전송후 그다음 SCL의 펄스동안에 ACK 신호 전송
- ACK 신호는 ADA에 High 비트 전송 의미
- 마지막 바이트의 전송후에는 ACK 신호 전송 않음



I2C 인터페이스

* 슬레이브 디바이스의 주소 포맷

- PCF8591의 주소 형식

7	6	5	4	3	2	1	0
0	1	0	0	1	A2	A1	A0

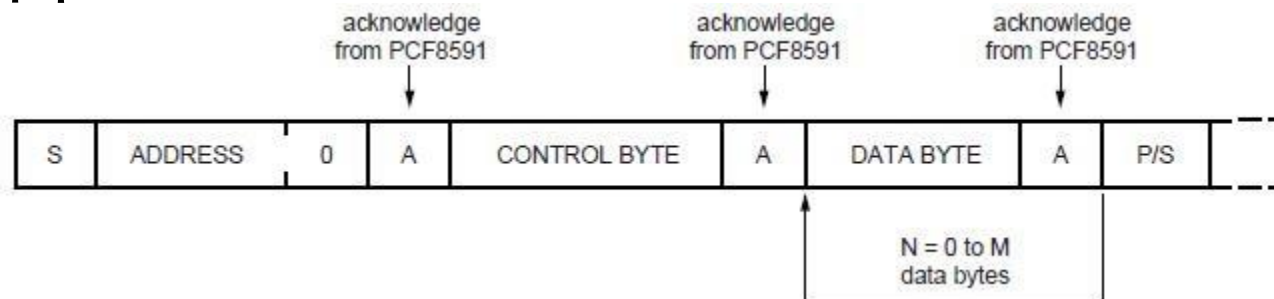
참고) I2C 통신에서 전송될 때의 주소바이트 형식 (위 주소가 좌쉬프트 + R/W비트)

7	6	5	4	3	2	1	0
1	0	0	1	A2	A1	A0	R/W

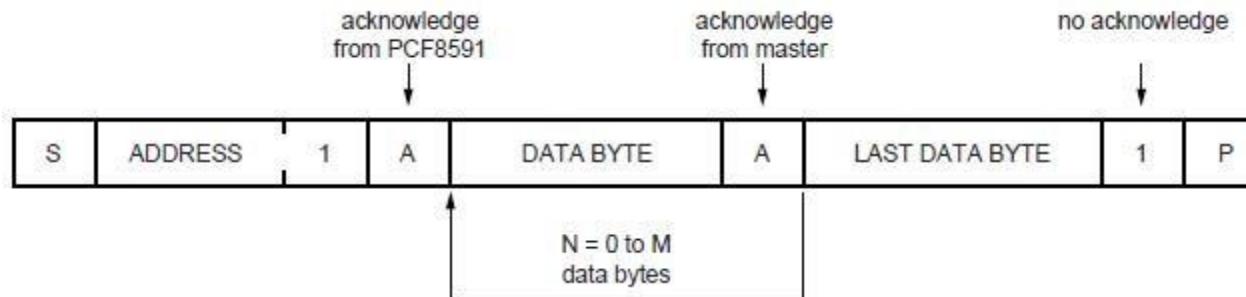
I2C 인터페이스

* 데이터 전송 포맷

• 기록 모드



• 판독 모드



I2C 인터페이스

* wiringPiI2C 라이브러리 함수

- I2C 통신을 위한 슬레이브 주소를 지정하여 초기화하는 함수

```
// devID는 i2cdetect 명령을 통해 얻은 슬레이브 디바이스의 I2C 주소
int wiringPiI2CSetupInterface(const char *device, int devId) ;
int wiringPiI2CSetup(const int devId) ;
```

- 슬레이브 디바이스에 제어 바이트 등을 전송하는 함수

```
// 단순 데이터를 전송 가능, 레지스터를 지정하여 1바이트 혹은 2바이트 전송
int wiringPiI2CWrite(int fd, int data) ;
int wiringPiI2CWriteReg8(int fd, int reg, int data) ;
int wiringPiI2CWriteReg16(int fd, int reg, int data) ;
```

- 슬레이브 디바이스로부터 데이터를 수신 받는 함수

```
int wiringPiI2CRead(int fd) ;
int wiringPiI2CReadReg8(int fd, int reg) ;
int wiringPiI2CReadReg16(int fd, int reg) ;
```

ADC/DAC 모듈

* ADC/DAC(YL-40) 모듈

- PCF8591 IC 장착

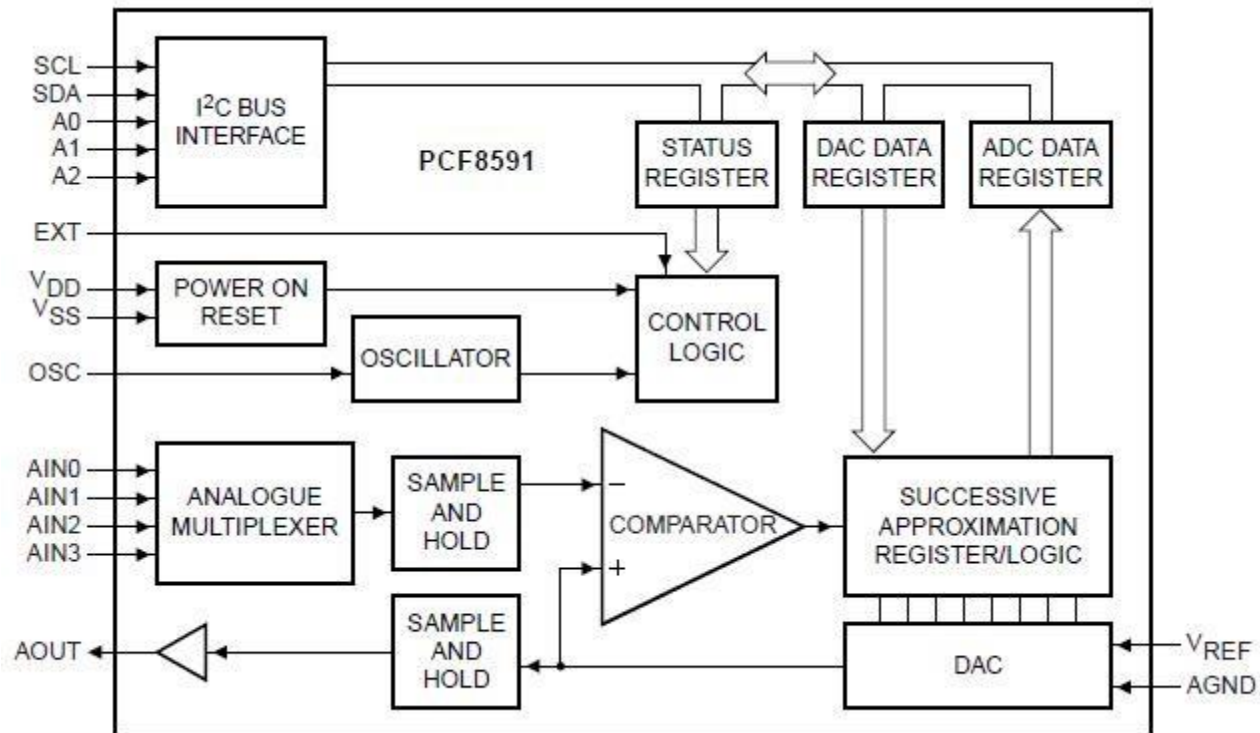


- 3개 아날로그 센서 장착(조도, 온도, 가변저항)
- 이들 센서와 아날로그 채널, 점퍼선과의 관계

채널	내장디바이스	점퍼선	비고
AIN0	조도센서	P5	
AIN1	온도센서	P4	
AIN2	-	-	
AIN3	가변저항	P6	

ADC/DAC 모듈

* PCF8591 IC(ADC/DAC) 블록 다이어그램



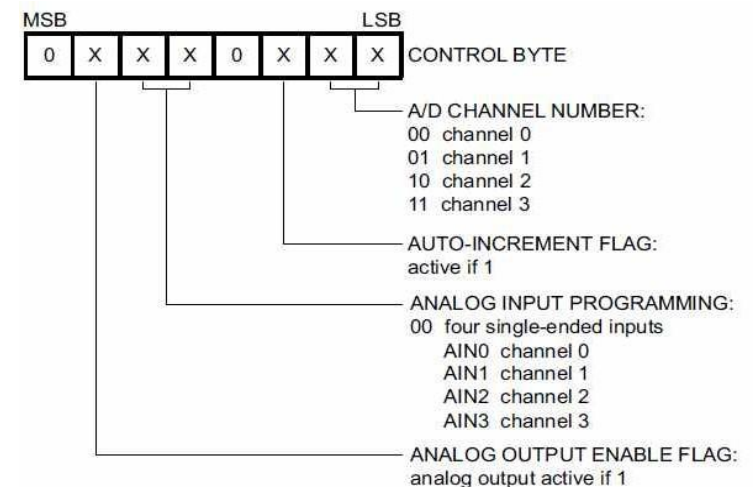
ADC/DAC 모듈

* PCF8591 IC (ADC/DAC)(계속)

- 전송 바이트
- 첫 바이트 : 디바이스의 주소 및 R/W비트
- 2번째 바이트 : 제어 바이트(채널선택 등)

7	6	5	4	3	2	1	0
1	0	0	1	A2	A1	A0	R/W

- 3번째 바이트 : DAC 변환시 이용



ADC/DAC 모듈

* 회로구성

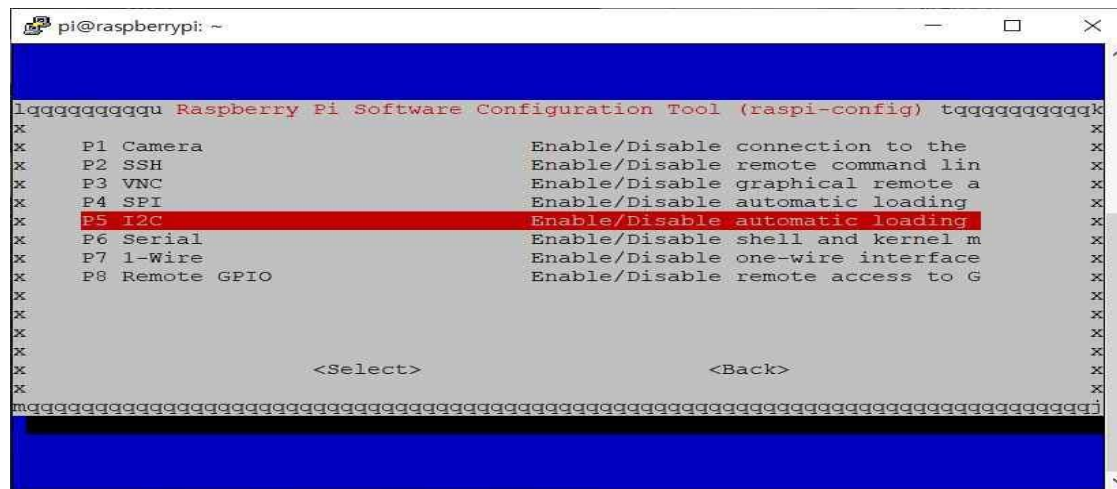
- SDA 단자는 SDA1(wiringPi #8, BCM_GPIO #2)에,
- SCL 단자는 SCL1(wringPi #9, BCM_GPIO #3)에 연결
- 전원 단자를 연결

ADC/DAC 변환

* I²C 활성화

\$ sudo raspi-config

: Interfacing Options항목 선택후, I2C 항목 선택하여 활성화



\$ sudo reboot

ADC/DAC 모듈

* I²C 활성화 확인

- 부팅할 때 활성화 내역

```
$ cat /boot/config.txt
```

```
dtparam=i2c_arm=on
```

- 부팅할 때 적재할 커널 모듈

```
$ cat /etc/modules
```

```
i2c-dev
```

ADC/DAC 모듈

* I2C 모듈의 주소 확인

- 모듈을 연결하고, 전원을 인가 후 다음의 명령을 통해 I2C 주소 확인

```
$ i2cdetect -y 1
```

```
pi@raspberrypi:~/IFC415/10 I2C/adc $ i2cdetect -y 1
```

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
10:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
20:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
30:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
40:	--	--	--	--	--	--	--	--	48	--	--	--	--	--	--	--
50:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
60:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
70:	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

```
pi@raspberrypi:~/IFC415/10 I2C/adc $
```


ADC/DAC 모듈

* I2C초기화 및 채널 선택

- 모듈 제어위한 I2C 디바이스 주소를 설정및 초기화

```
if( (fd = wiringPiI2CSetup(0x48) ) < 0) {  
  
    // 특정 채널선택을 위해 ..      0 0 0 0 0 0 x x  
    wiringPiI2CWrite(fd, 0x00 | a2dChannel);      // 채널선택  
  
    // 각 채널을 자동 증가 방식으로 선택  
    (자동 증가 비트와 아날로그 출력 인에이블 비트를 1로 설정)  
    0 1 0 0 0 1 x x  
    wiringPiI2CWrite(fd, 0x44);      // 채널 자동 증가 모드
```

ADC/DAC 변환

〈실습1〉 가변저항 통한 전압의 ADC 변환 : 모듈의 P6 단자를 점퍼선으로 연결(AIN3 채널 선택)

```
#include <wiringPi.h>
#include <wiringPiI2C.h>

int main (void) {
    int fd;
    int i, cnt;
    int a2dChannel = 3;    // analog channel AIN3, VR
    int prev, a2dVal; float a2dVol; float Vref = 5.0;
    printf("[ADC/DAC(YL-40) Module testing.....]\n");
    if((fd = wiringPiI2CSetup(0x48))<0) {
        printf("wiringPiI2CSetup failed:\n");
    }
    cnt = 0;
    while(1) {
        wiringPiI2CWrite(fd, 0x00 | a2dChannel);    // 0000_0011
        prev = wiringPiI2CRead(fd);    // Previously byte, garbage
        printf("[%d] previous = %d, ", cnt, prev);
        a2dVal = wiringPiI2CRead(fd);
        printf("2nd a2dVal = %d, ", a2dVal); a2dVol = 5.0 - (a2dVal * Vref / 255); printf("a2dVol = %f[V]\n", a2dVol);
        delay(1000);
        cnt++;
    }
}
```

ADC/DAC 변환

〈실습2〉 조도센서 출력을 ADC변환 : 모듈의 P5 단자를 점퍼선으로 연결(AIN0 채널 선택)

```
#include <stdio.h>
#include <wiringPi.h>
#include <wiringPiI2C.h>

int main (void) {
    int fd;
    int i, cnt;
    int a2dChannel = 0;    // analog channel AIN0, CDS sensor
    int prev, a2dVal;
    int threshold = 180;

    printf("[ADC/DAC(YL-40) Module testing.....]\n");

    /*
    if((fd = wiringPiI2CSetup(0x48))<0) {
        printf("wiringPiI2CSetup failed:\n");
    }
    */
    if((fd = wiringPiI2CSetupInterface("/dev/i2c-1",0x48))<0) {
        printf("wiringPiI2CSetupInterface failed:\n");
    }
}
```

continue

ADC/DAC 변환

```
cnt = 0;

while(1) {
    wiringPi2CWrite(fd, 0x00 | a2dChannel);    // 0000_0000

    prev = wiringPi2CRead(fd);    // Previously byte, garbage
    a2dVal = wiringPi2CRead(fd);
    printf("[%d] prev = %d, ", cnt, prev);
    printf("a2dVal = %d, ", a2dVal);

    if(a2dVal < threshold) {
        printf("Bright!!\n");
    } else {
        printf("Dark!!\n");
    }
    delay(1000);
    cnt++;
}
```

ADC/DAC 변환

〈실습3〉 온도센서 출력을 ADC II :: 모듈의 P4 단자를 점퍼선으로 연결(AIN1채널 선택)

```
#include <stdio.h>
#include <wiringPi.h>
#include <wiringPiI2C.h>

int main (void) {
    int fd;
    int i, cnt;
    int a2dChannel = 1;    // analog channel AIN1, thermistor sensor
    int prev, a2dVal;
    int threshold = 200;

    printf("[ADC/DAC(YL-40) Module testing.....]\n");
    if((fd = wiringPiI2CSetupInterface("/dev/i2c-1", 0x48))<0) {
        printf("wiringPiI2CSetupInterface failed:\n");
    }

    cnt = 0;

    while(1) {
        wiringPiI2CWrite(fd, 0x00 | a2dChannel);    // 0000_0001

        prev = wiringPiI2CRead(fd);    // Previously byte, garvage
        a2dVal = wiringPiI2CRead(fd);
        printf("[%d] prev = %d, ", cnt, prev);
        printf("a2dVal = %d, ", a2dVal);
```

continue

ADC/DAC 변환

```
    if(a2dVal > threshold)
        printf("Hot!!\n");
    else
        printf("Normal!!\n");

    delay(1000);
    cnt++;
}
```

ADC/DAC 변환

〈실습4〉 ADC 변환(자동 증가 모드): 자동 증가 플래그비트를 1로 하여 4개 채널

```
int main (void) {
    int fd;
    int i, cnt;
    int prev, ain[4];
    float a2dVol;
    float Vref = 5.0;
    printf("[ADC/DAC(YL-40) Module testing.....Auto Inc Mode]\n");

    if((fd = wiringPi2CSetup(0x48))<0) {
        printf("wiringPi2CSetup failed:\n");
    }
    cnt = 0;
    while(1) {
        wiringPi2CWrite(fd, 0x44);          // 0100_01xx, Auto-Inc

        prev = wiringPi2CRead(fd);          // Previously byte, garbage
        for(i=0; i<4; i++)                  // for 4 channels
            ain[i] = wiringPi2CRead(fd);
        printf("[%d] ", cnt);
        for(i=0; i<4; i++)
            printf("ain%d = %d, ", i, ain[i]);
        printf("\n");
        delay(1000);
        cnt++;
    }
}
```