FND 디바이스 제어

조 진 성
경희대학교 컴퓨터공학과
Mobile & Embedded System Lab.

FND

❖ FND(Flexible Numeric Display)

- 일정 개수 이상의 LED를 사용하여 아라비아 숫자 및 영문자를 표시할 수 있는 장치

- 7-Segment LED
  - FND의 한 종류
  - FND를 표시하는데 있어 최소한으로 사용되는 7개의 LED를 조합한 것
    - FND에서 Segment는 7개보다 더 많을 수도 있음

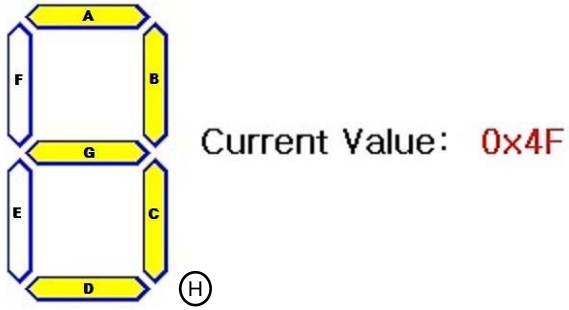
- FND 기능 및 구조
  - FND는 2개의 사각형이 상하로 연결된 형태
    - 위쪽 사각형의 아래 획과 아래쪽 사각형의 위쪽 획이 합쳐진 모양을 가짐
  - LED로 구성된 7개의 획은 각각 꺼지거나 켜질 수 있음
  - 7-세그먼트의 종류는 공통(Common) 단자에 인가되는 전원에 따라 다음과 같이 분류
    - Anode(+공통)형 : 공통 단자에 VCC(+5V)에 연결하고 입력 단자에 0V를 인가 시, LED 점등
    - Cathode(-공통)형 : 공통 단자에 접지(0V)를 연결하고 입력 단자에 +5V를 인가 시, LED 점등

❖ FND(Flexible Numeric Display)

▪ FND 기능 및 구조

- 7-세그먼트 표시 장치의 각 획은 맨 위쪽 가로 획부터 시계방향으로, 그리고 마지막 가운데 가로 획까지 각각 A부터 G로 명칭
  - 사용 예1) 7-세그먼트에 '3'을 표시: A,B,G,C,D 의 LED를 ON
  - 사용 예2) 7-세그먼트에 '7'을 표시: F,A,B,C 의 LED를 ON
- 소수를 나타내기 위해서 숫자의 오른쪽 아래에 소수점이 붙기도 함
  - 소수점은 DP(Decimal Point) 또는 H로 명칭

< 7-세그먼트를 통해 표현한 '3' >



< 7-Segment LED에 16진수를 표현하기 위한 비트 값 >

16진수	7-Segment의 비트값								데이터 값 (HEX)
	H	G	F	E	D	C	B	A	
0	0	0	1	1	1	1	1	1	0X3F
1	0	0	0	0	0	1	1	0	0X06
2	0	1	0	1	1	0	1	1	0X5B
3	0	1	0	0	1	1	1	1	0X4F
4	0	1	1	0	0	1	1	0	0X66
5	0	1	1	0	1	1	0	1	0X6D
6	0	1	1	1	1	1	0	1	0X7D
7	0	0	1	0	0	1	1	1	0X27
8	0	1	1	1	1	1	1	1	0X7F
9	0	1	1	0	0	1	1	1	0X67
A	0	1	1	1	0	1	1	1	0X77
B	0	1	1	1	1	1	0	0	0X7C
C	0	0	1	1	1	0	0	1	0X39
D	0	1	0	1	1	1	1	0	0X5E
E	0	1	1	1	1	0	0	1	0X79
F	0	1	1	1	0	0	0	1	0X71

FND

❖ FND(Flexible Numeric Display)

- 각 숫자에 해당하는 7세그먼트 표시 장치의 모습

FND

❖ FND(Flexible Numeric Display)

▪ FND Array

- 7- Segment는 7개의 LED로 1개의 FND의 제어가 가능
- FND를 사용하는 장치들을 살펴보면, 여러 개의 FND가 사용되는 경우 다수
- 여러 개의 FND를 사용하는 경우
  - DP 까지 8개의 LED가 포함되어 8핀을 사용하는 경우 8의 배수 형태로 핀의 개수가 필요
  - 예를 들어, 8핀을 사용하는 4개의 FND를 제어하기 위해서는 무려 32개의 신호 데이터 핀 필요
- 다수 FND의 사용으로 핀의 개수가 증가하는 것을 해결하기 위해 Array 방식의 FND 고안

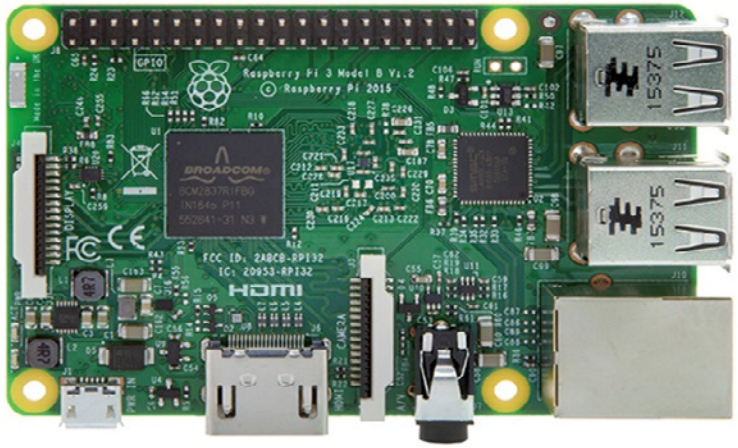
- 즉, FND Array는 출력 포트의 효율적 사용을 위해 FND를 구성하는 몇 개의 데이터 전송 핀과 점등할 FND를 선택하는 몇 개의 핀을 사용하여 다수의 FND를 제어하는 것
  - 데이터의 출력은 공통으로 연결하고 출력되는 FND 위치를 별도로 지정

▪ 본 강의에서 사용하는 7-SEG 모듈은 총 6개의 FND로 구성

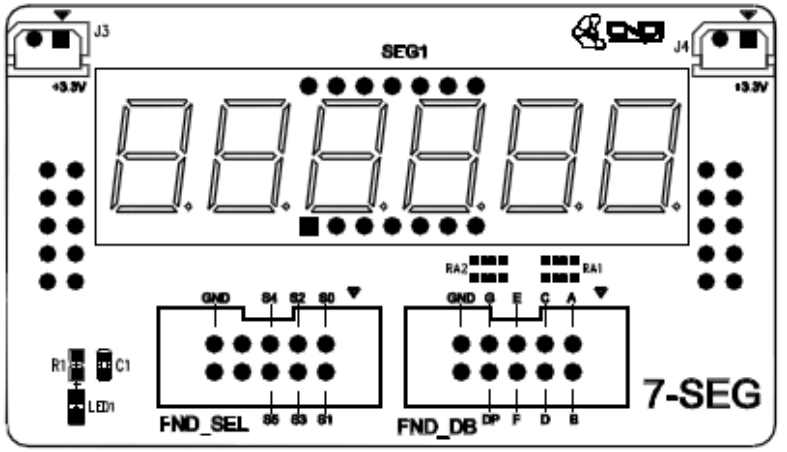
- FND_SEL 포트의 S0~S5 핀에 Low 신호를 출력하여 FND를 선택
- FND_DB 포트의 A~DP 핀을 통해 해당하는 LED Segment에 High 신호를 출력하여 On/Off

하드웨어 구성

❖ Raspberry Pi 3 Model B

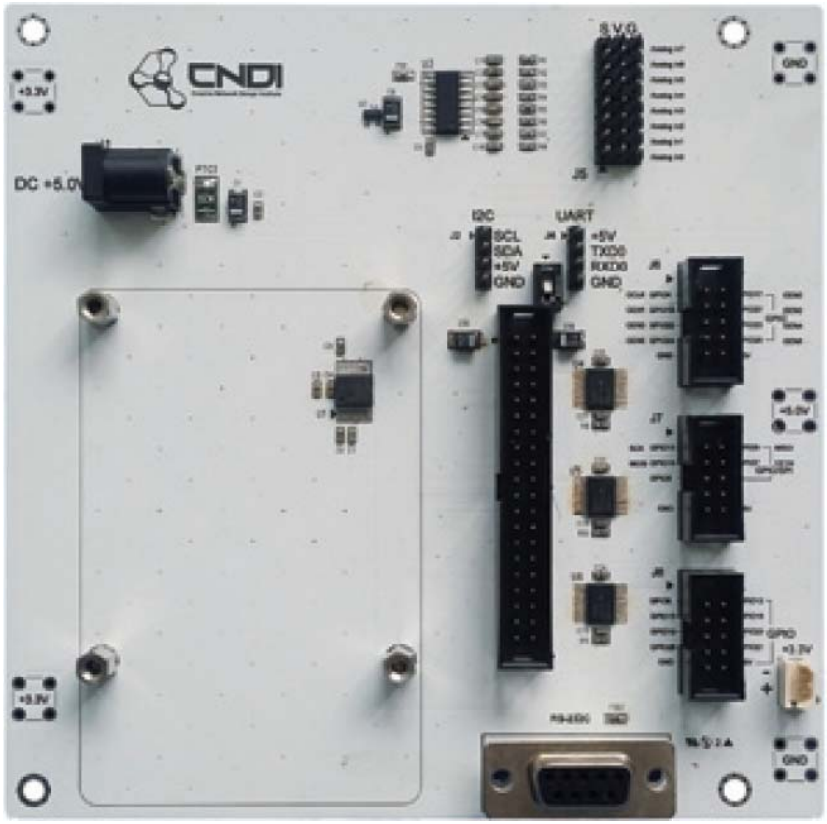


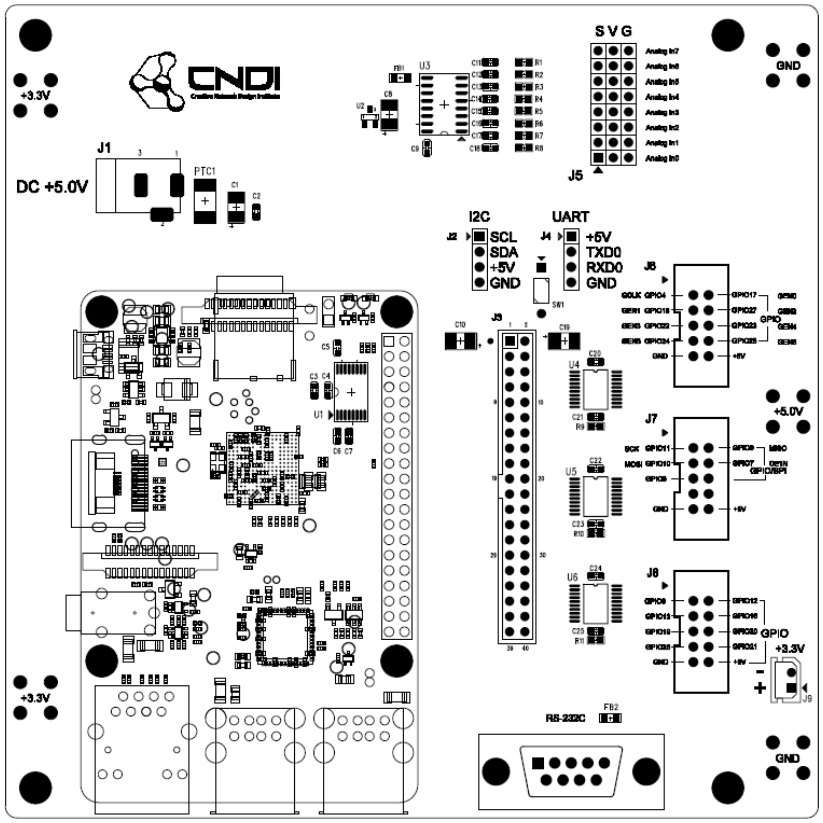
❖ FND(Flexible Numeric Display) 모듈



하드웨어 구성

❖ Raspberry Pi Adapter



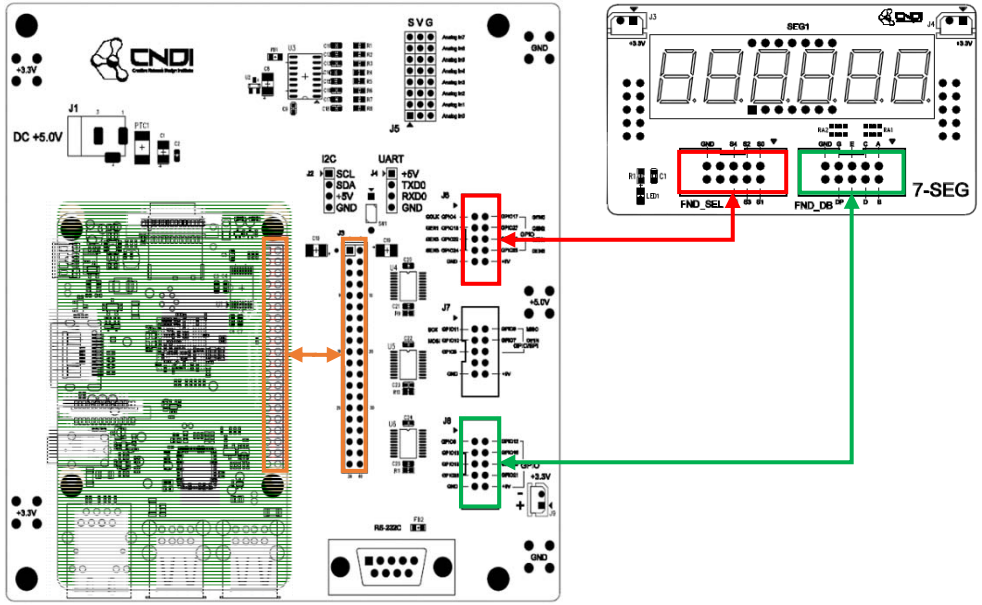


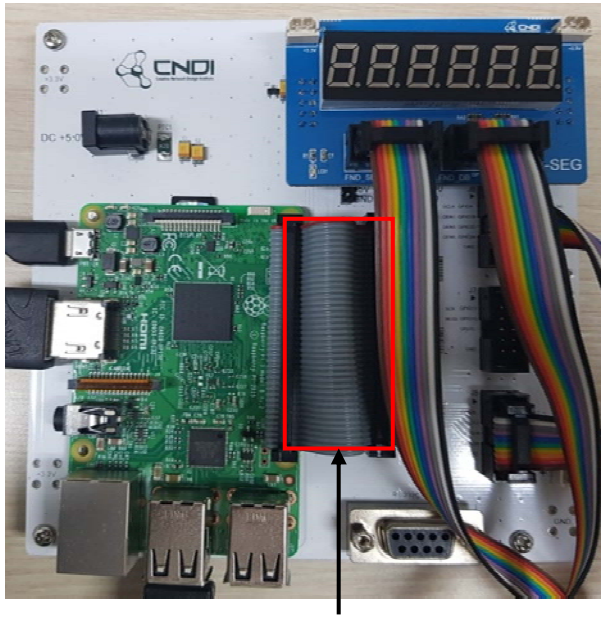
하드웨어 구성

❖ 실습 예제 (1), (2) 위한 하드웨어 구성

▪ 실습을 위한 Raspberry Pi 3와 LED 결선 방법

- Raspberry Pi 3와 Raspberry Pi Adapter 보드를 20 x 2 케이블을 이용하여 연결
- Raspberry Pi Adapter 모듈의 포트 J6(10핀)을 FND 모듈의 포트 FND_SEL(10핀)에 연결
- Raspberry Pi Adapter 모듈의 포트 J8(10핀)을 FND 모듈의 포트 FND_DB(10핀)에 연결

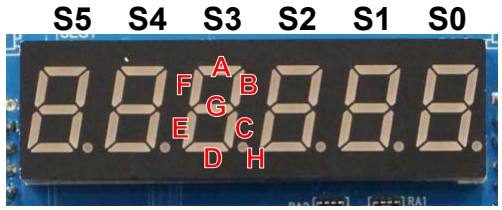




실습 예제 (1)

❖ FND 한 개만 작동

- FND(S3)에 숫자 출력
- lab2-4_1.c

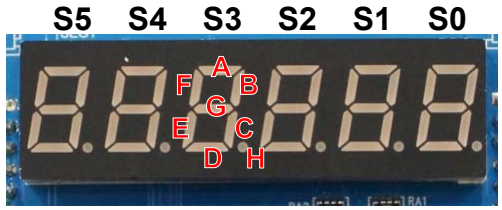



```
// GPIO Access Library 헤더파일 선언
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>

// 각 FND와 연결된 라즈베리파이 핀(S0, S1, …, S5)
const int FndSelectPin[6] = { 4, 17, 18, 27, 22, 23 };
// FND의 LED와 연결된 라즈베리파이 핀(A, B, …, H)
const int FndPin[8] = { 6, 12, 13, 16, 19, 20, 26, 21 };
// FND에 출력되는 문자 (0~9) 배열
const int FndFont[10] = { 0x3F, 0x06, 0x5B, 0x4F, 0x66,
                          0x6D, 0x7D, 0x07, 0x7F, 0x67 };

// 초기화 함수, WiringPi 라이브러리 초기화, Select 핀 및 LED 핀 초기화를 담당)
void Init() {
        int i;

        if( wiringPiSetupGpio() == -1 ) {
                printf( "wiringPiSetupGpio() error\n");
                exit(-1);
        }

        for( i = 0; i < 6; i++ ) {
                pinMode( FndSelectPin[ i ], OUTPUT );       // Select 핀을 출력으로 설정
                digitalWrite( FndSelectPin[ i ], HIGH);     // Select 핀 OFF
        }
```

실습 예제 (1)

❖ FND 한 개만 작동

- FND(S3)에 숫자 출력
- lab2-4_1.c

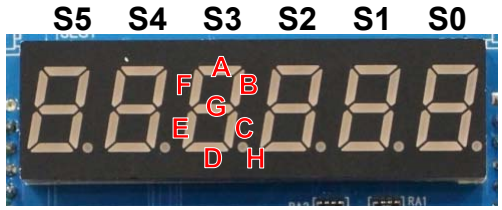



```
        for( i = 0; i < 8; i++ ) {
                pinMode( FndPin[ i ], OUTPUT );  // LED 핀을 출력으로 설정
                digitalWrite( FndPin[ i ], LOW );   // LED 핀을 OFF
        }
}

// FND를 출력하는 함수
void FndDisplay(int position, int num) {
        int i, j;
        int flag = 0;  // FndPin[ ]을 ON/OFF
        int shift = 0x01;   // FndFont와 And 연산하여  출력할 LED의 상태 결정

        for( i = 0; i < 8; i++ ) {
                flag = ( FndFont[num] & shift );  // i = 0, FndFont[ 0 ] = 0x3F라 하면  (0b00111111 & 0b00000100 = 1) 이다.

                digitalWrite( FndPin[ i ], flag );  // FndPin[ ]을 flag( 0또는 1 )로 ON/OFF

                shift <<= 1;  // 왼쪽으로 한 비트 쉬프트한다. i = 0이라 하면, ( shift = 0b00000001 )에서 ( shift = 0b00000010)로 변한다.
        }

        digitalWrite( FndSelectPin[ position ], LOW );
}
```

실습 예제 (1)

FND 한 개만 작동

- FND(S3)에 숫자 출력
- lab2-4_1.c

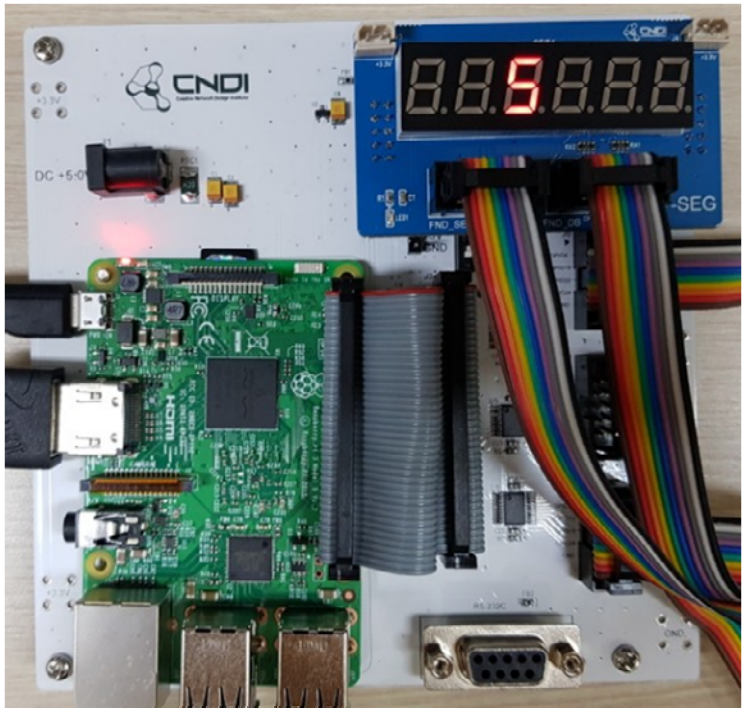



```
int main( int argc, char **argv ) {
        if( argc != 3) {
                printf( "Usage: %s  [ position ]  [ number ] ", arvg[ 0 ] );
                exit( -1 );
        }

        Init();

        // argv[1] = position, argv[2] = number, argv는 문자 스트링이므로 정수로 변환
        // int  atoi( const  char  *string )
        FndDisplay( atoi( argv[1] ), atoi( argv[2] ) );

        return 0;
}
```

실습 예제 (1)

FND 한 개만 작동

- 제어 프로그램 컴파일
  - gcc –o lab2-4_1 lab2-4_1.c –lwiringPi
- 제어 프로그램 실행
  - ./lab2-4_1 3 5
- 동작 화면



실습 예제 (2)

❖ FND 여러 개 작동

- FND(S0~S5)에 숫자 출력하기
- lab2-4_2.c

```
// GPIO Access Library 헤더파일 선언
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>

// 각 FND와 연결된 라즈베리파이 핀(S0, S1, …, S5)
const int FndSelectPin[6] = { 4, 17, 18, 27, 22, 23 };
// FND의 LED와 연결된 라즈베리파이 핀(A, B, …, H)
const int FndPin[8] = { 6, 12, 13, 16, 19, 20, 26, 21 };
// FND에 출력되는 문자 (0~9) 배열
const int FndFont[10] = { 0x3F, 0x06, 0x5B, 0x4F, 0x66,
                          0x6D, 0x7D, 0x07, 0x7F, 0x67 };

// 초기화 함수, WiringPi 라이브러리 초기화, Select 핀 및 LED 핀 초기화를 담당)
void Init() {
        int i;

        if( wiringPiSetupGpio() == -1 ) {
                printf( “wiringPiSetupGpio() error\n”);
                exit(-1);
        }

        for( i = 0; i < 6; i++ ) {
                pinMode( FndSelectPin[ i ], OUTPUT );       // Select 핀을 출력으로 설정
                digitalWrite( FndSelectPin[ i ], HIGH );    // Select 핀 OFF
        }
```

실습 예제 (2)

❖ FND 여러 개 작동

- FND(S0~S5)에 숫자 출력하기
- lab2-4_2.c

```
        for( i = 0; i < 8; i++ ) {
                pinMode( FndPin[ i ], OUTPUT );  // LED 핀을 출력으로 설정
                digitalWrite( FndPin[ i ], LOW );   // LED 핀을 OFF
        }
}

// FND를 선택하는 함수, S0 ~ S5 중 파라미터(position)에 해당하는 FND 선택
void FndSelect (int position) {
        int i;

        for( i = 0; i < 6; i++ ) {
                if( i == position ) {
                        digitalWrite( FndSelectPin[ i ], LOW );  // 선택된 FND의 Select 핀 ON
                }
                else {
                        digitalWrite( FndSelectPin[ i ], HIGH );  // 선택되지 않은 FND의 Select 핀 OFF
                }
        }
}

// FND를 출력하는 함수
void FndDisplay(int position, int num) {
        int i, j;
        int flag = 0;  // FndPin[ ]을 ON/OFF
        int shift = 0x01;   // FndFont와 And 연산하여  출력할 LED의 상태 결정
```

실습 예제 (2)

❖ FND 여러 개 작동

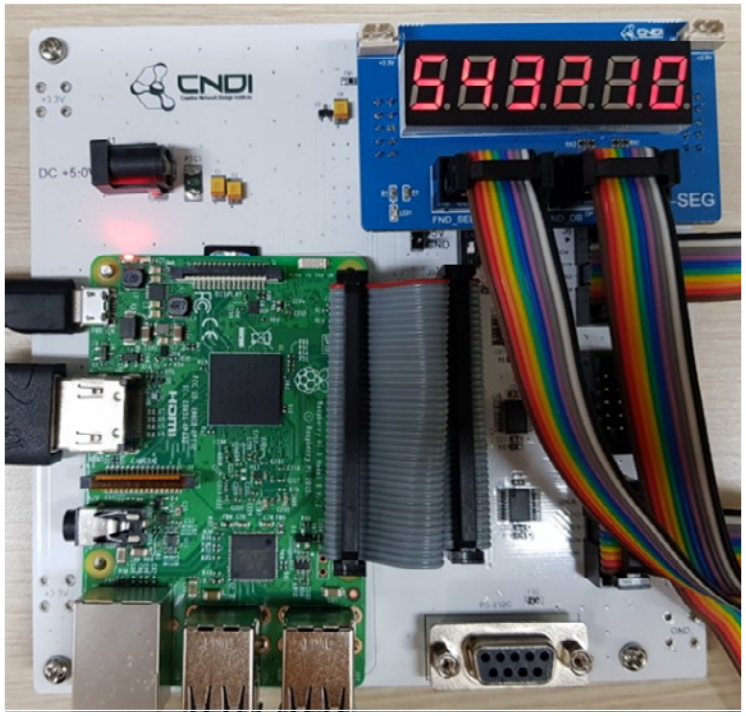
- FND(S0~S5)에 숫자 출력하기
- lab2-4_2.c

```
        for( i = 0; i < 8; i++ ) {
                flag = ( FndFont[num] & shift );  // i = 0, FndFont[ 0 ] = 0x3F라 하면  (0b00111111 & 0b00000100 = 1) 이다.

                digitalWrite( FndPin[ i ], flag );  // FndPin[ ]을 flag( 0또는 1 )로 ON/OFF

                shift <<= 1;  // 왼쪽으로 한 비트 쉬프트한다. I = 0이라 하면, ( shift = 0b00000001 )에서 ( shift = 0b00000010)로 변한다.
        }

        FndSelect( position ) ;
}

int main() {
        int pos;
        int data[6] = { 0, 1, 2, 3, 4, 5 };  // 출력할 문자 데이터

        Init();

        while(1) {
                for( pos = 0; pos < 6; pos++ ) {
                        FndDisplay( pos, data[ pos ] );
                        delay(1);  // WiringPi 라이브러리에서 정의된 delay() 함수, void delay( unsinged  int  howLong )
                }
        }

        return 0;
}
```

실습 예제 (2)

FND 여러 개 작동

- 제어 프로그램 컴파일
  - gcc –o lab2-4_2 lab2-4_2.c –lwiringPi
- 제어 프로그램 실행
  - ./lab2-4_2
- 동작 화면



실습 과제

❖ 글자 출력 및 shift하기

- 아래 제시된 'HELLO'를 FND에 출력

- 0.5초 간격으로 왼쪽으로 shift하고, 다시 오른쪽 부터 나타남이 반복

❖ 1/100 초시계

- 0000.00 부터 시작하여, 1/100 초시계 동작

Q & A

http://mesl.khu.ac.kr