



8x8 LED Dot Matrix

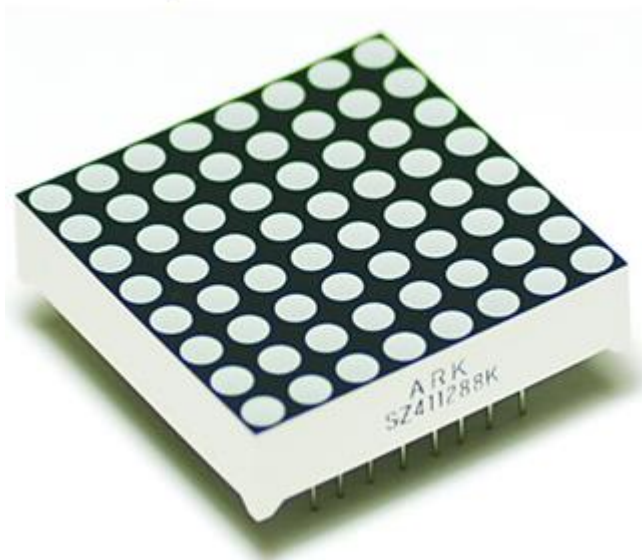
로봇SW 교육원

최상훈(shchoi82@gmail.com)

8x8 LED Dot Matrix

2

- 8x8 LED Dot Matrix
 - Cathode
 - RED



핀번호

3

- 품명이 마킹 되어 있는 면 왼쪽부터 반 시계 방향으로 넘버링

PIN NO. ①⑥ ①⑤ ①④ ①③ ①② ①① ①⑩ ①⑨



PIN NO. ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰

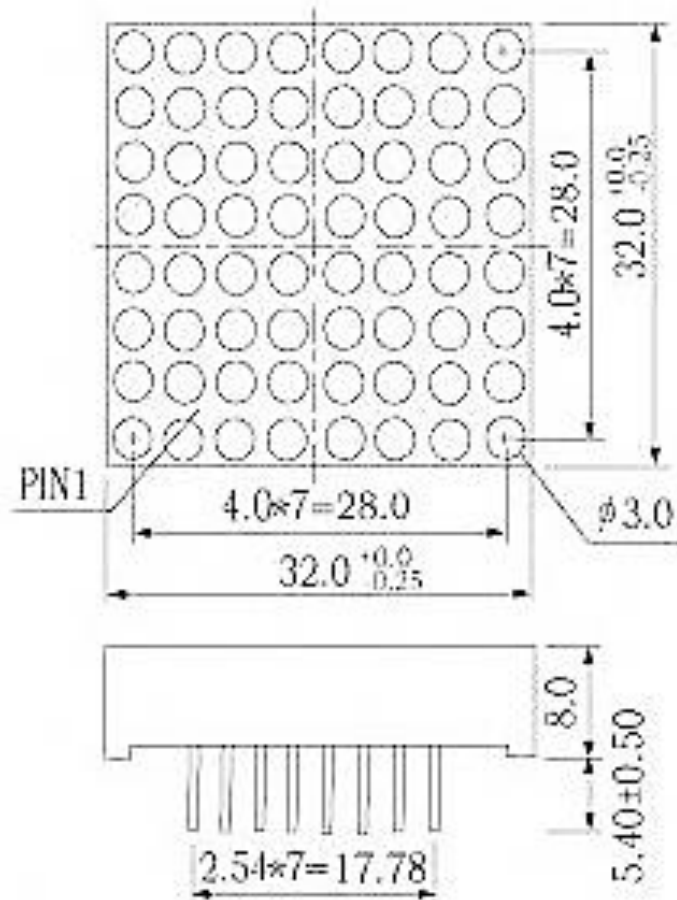


PIN NO. ① ② ③ ④ ⑤ ⑥ ⑦ ⑧

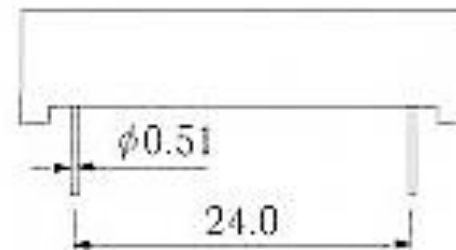


행(ROW)과 열(COL)

• 규격 및 도면



COL.	1	2	3	4	5	6	7	8	ROW
	○	○	○	○	○	○	○	○	1
	○	○	○	○	○	○	○	○	2
	○	○	○	○	○	○	○	○	3
	○	○	○	○	○	○	○	○	4
	○	○	○	○	○	○	○	○	5
	○	○	○	○	○	○	○	○	6
	○	○	○	○	○	○	○	○	7
	○	○	○	○	○	○	○	○	8

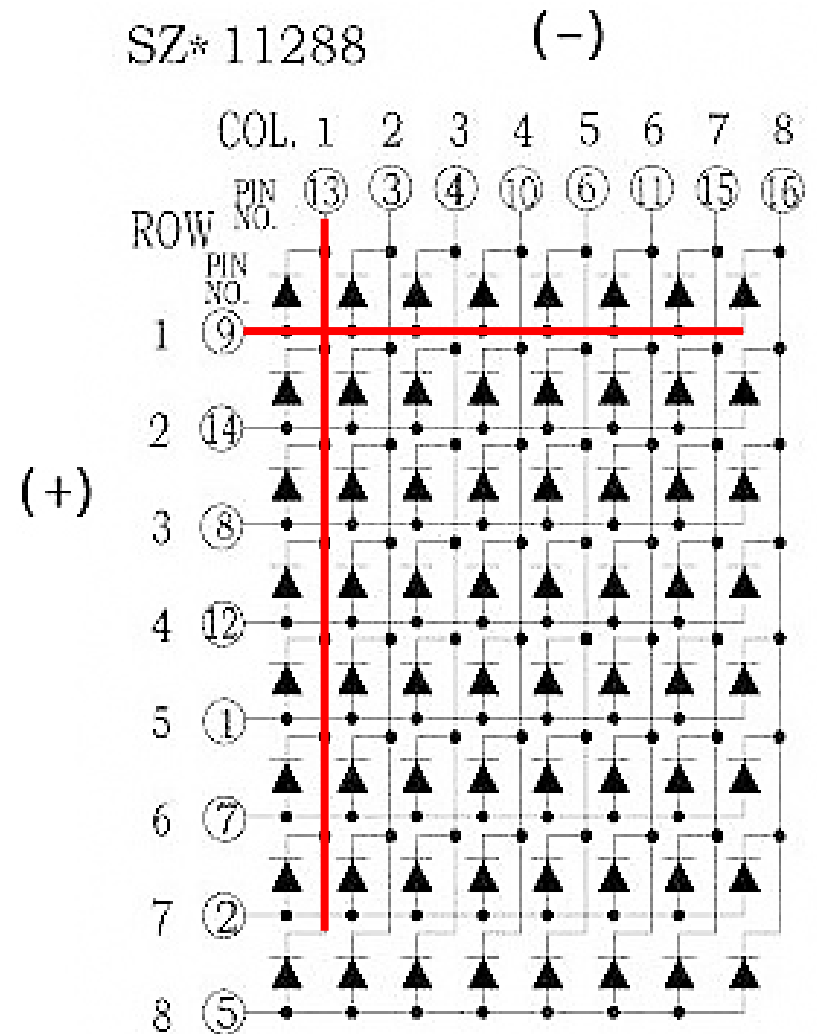
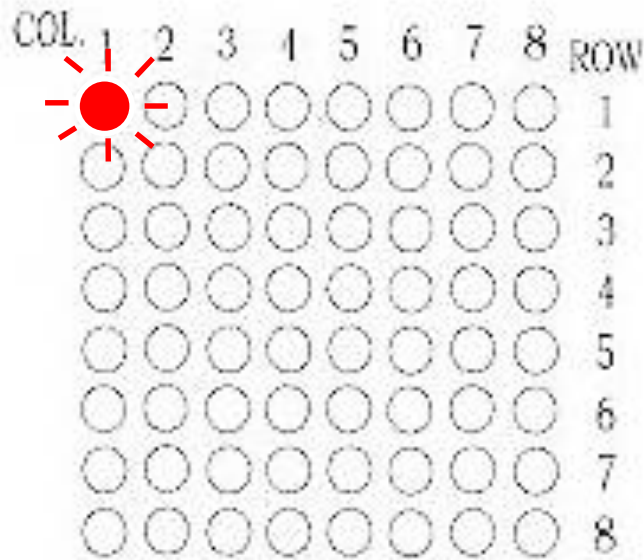


LED_{rc} 점등

5

• LED₁₁ 점등

- ⑨ HIGH
- ⑬ LOW



실습1-1 : LED₁₁ 점등

6

PIN NO. ①②③④⑤⑥⑦⑧⑨⑩⑪⑫⑬⑭⑮⑯⑰



PIN NO. ①②③④⑤⑥⑦⑧

PIN NO. ⑨⑩⑪⑫⑬⑭⑮⑯⑰⑱⑲⑳㉑㉒㉓㉔㉕㉖㉗㉘㉙㉚㉛㉜㉝㉞㉟㊱㊲㊳㊴㊵㊶㊷㊸㊹㊺㊻㊼㊽㊾㊿

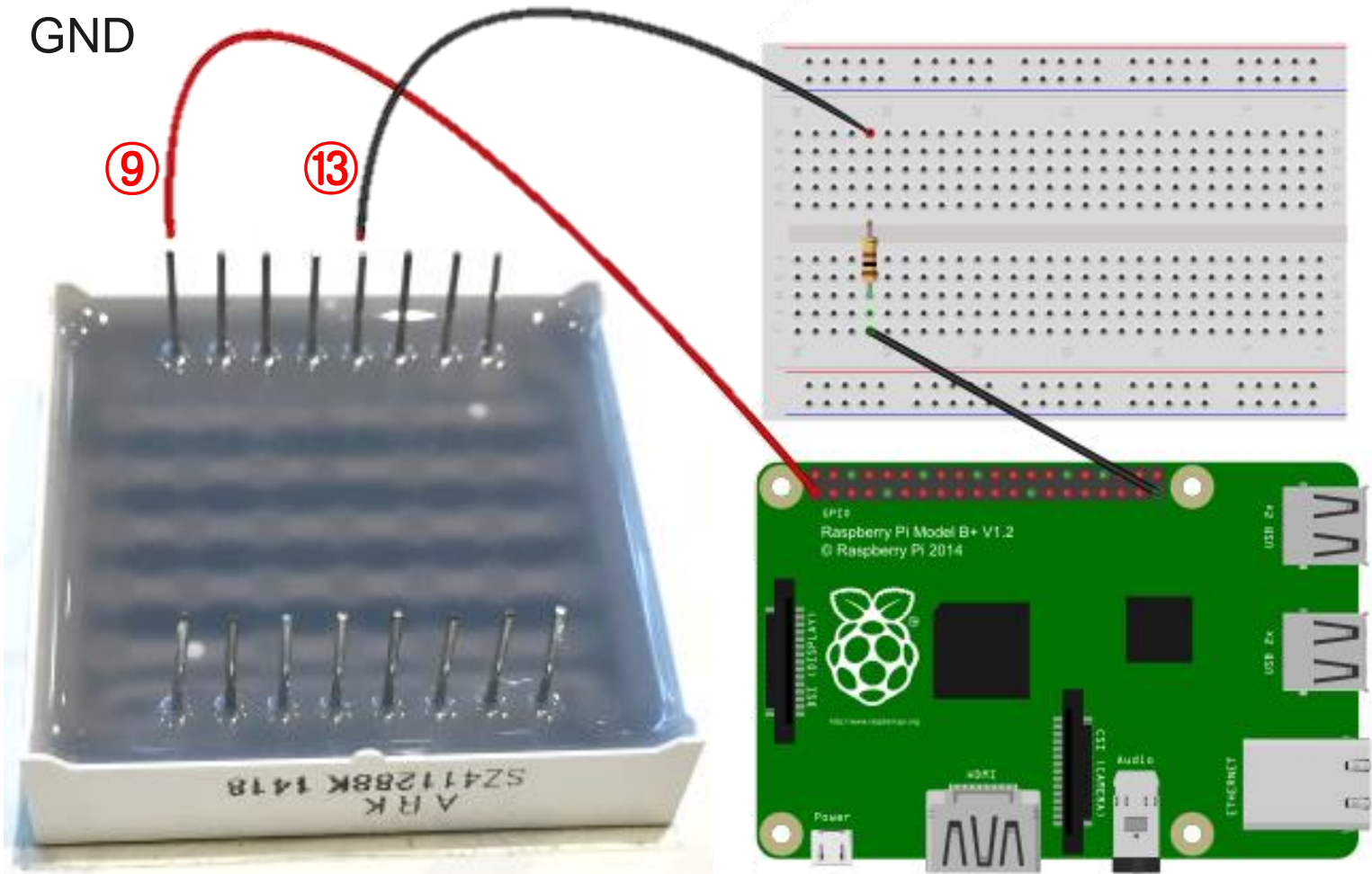
PIN NO. ①②③④⑤⑥⑦⑧



실습1-2 : LED₁₁ 점등

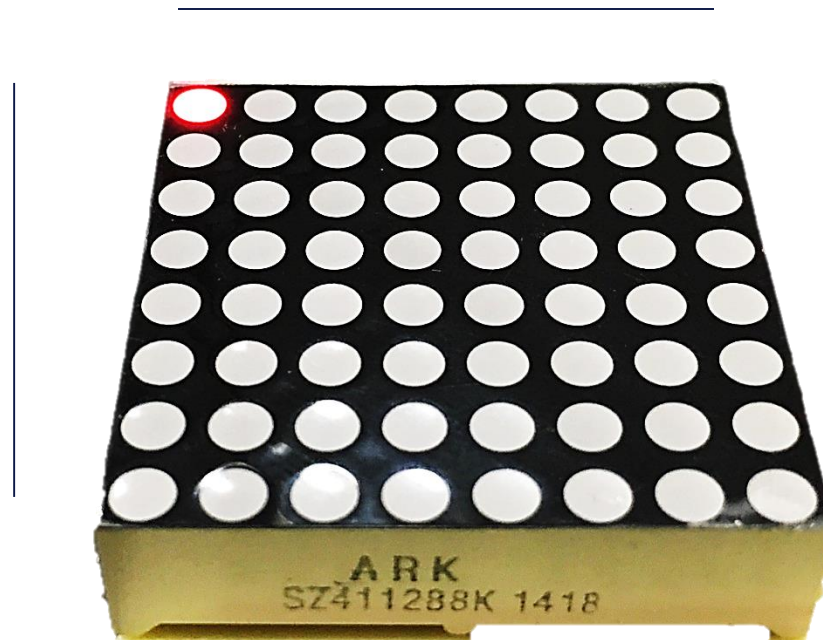
7

- 110Ω 저항
- 3.3v
- GND



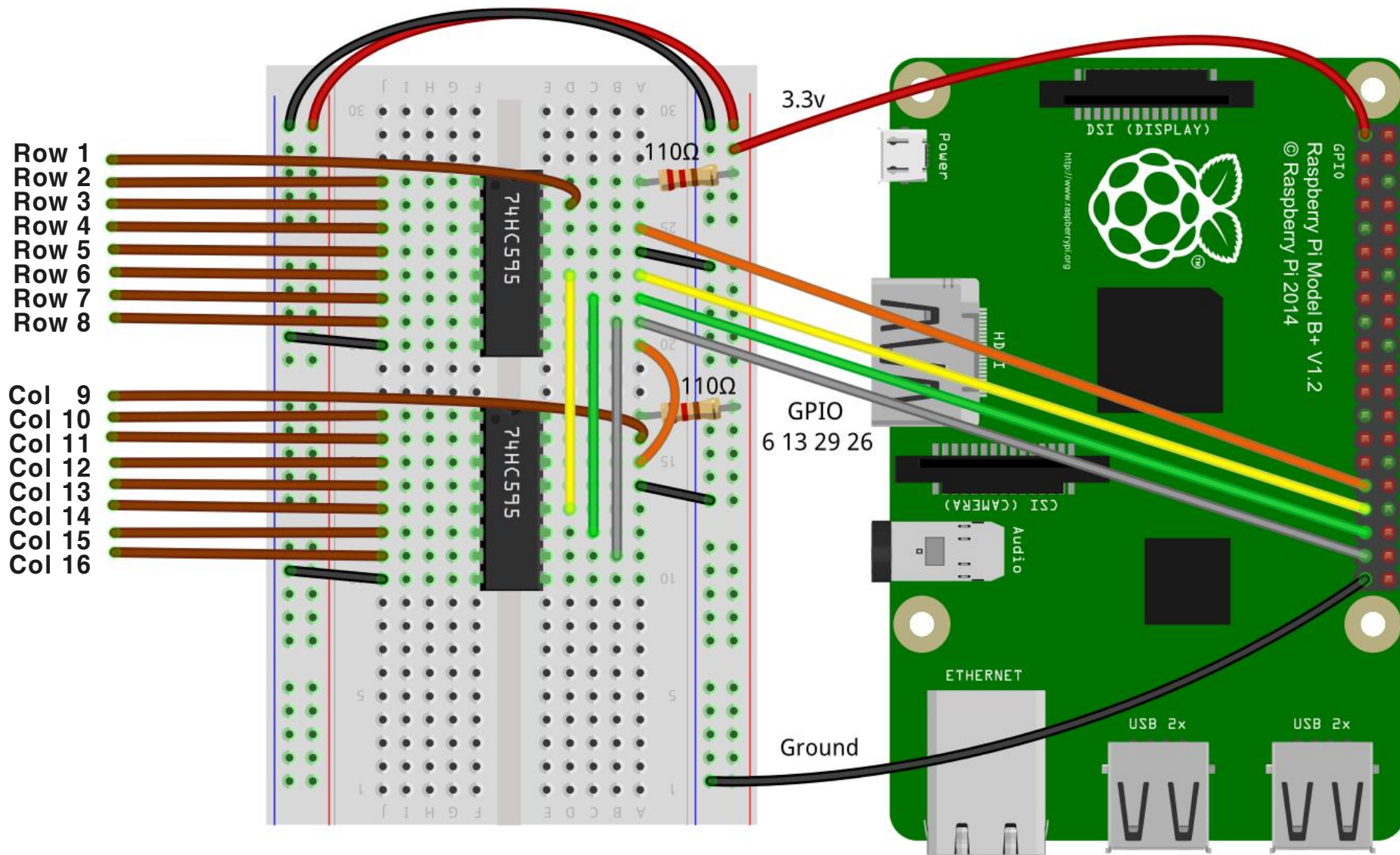
실습1-3 : LED₁₁ 점등

8



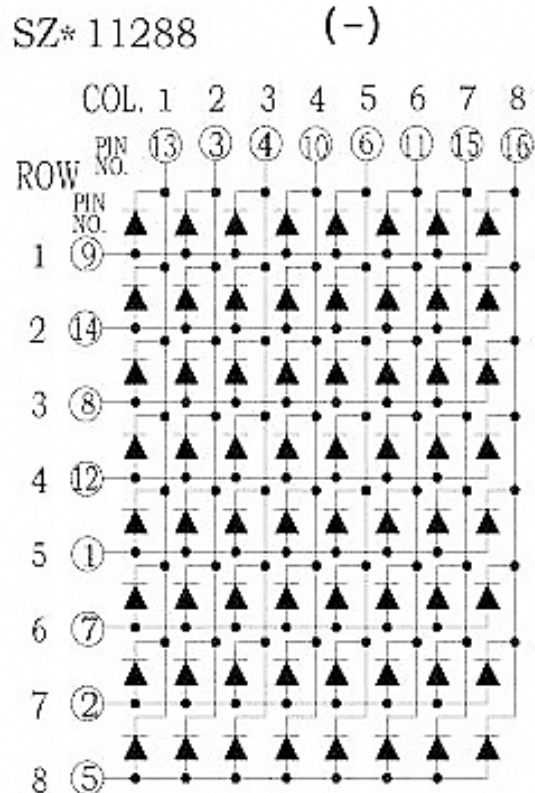
실습2-1 : Row 차례로 점등

9



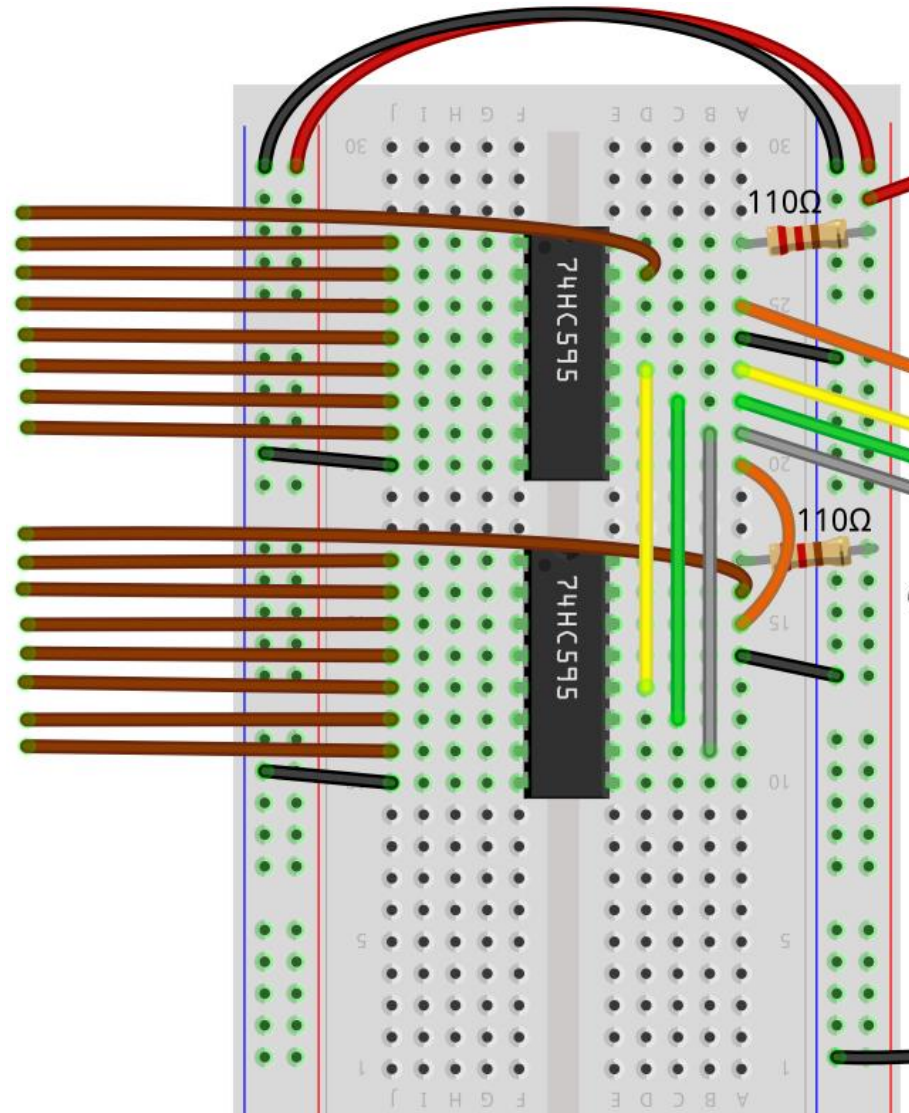
실습2-2 : Row 차례로 점등

10



Pin 9 - Row 1
 Pin 14 - Row 2
 Pin 8 - Row 3
 Pin 12 - Row 4
 Pin 1 - Row 5
 Pin 7 - Row 6
 Pin 2 - Row 7
 Pin 5 - Row 8

Pin 13 - Col 9
 Pin 3 - Col 10
 Pin 4 - Col 11
 Pin 10 - Col 12
 Pin 6 - Col 13
 Pin 11 - Col 14
 Pin 15 - Col 15
 Pin 16 - Col 16



실습2-3 : Row 차례로 점등

11

파일명 : 8x8LED_ex1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <stdint.h>
#define SDATA      6    // Serial Input Data
#define STR_CLK    13   // Storage Register Clock (LATCH)
#define SHR_CLK    19   // Shift Register Clock
#define SHR_CLEAR  26   // Shift Register Clear

void allclear(void);

void init(void)
{
    if(wiringPiSetupGpio() == -1){ // wiringPi
        fprintf(stderr, "wiringPiSetupGpio() error\n");
        exit(1);
    }
    pinMode(SDATA, OUTPUT);
    pinMode(STR_CLK, OUTPUT);
    pinMode(SHR_CLK, OUTPUT);
    pinMode(SHR_CLEAR, OUTPUT);
    allclear();
}

void allclear(void)
{
    digitalWrite(SHR_CLEAR, 0);
    digitalWrite(SHR_CLEAR, 1);
    digitalWrite(STR_CLK, 0);
    digitalWrite(STR_CLK, 1);    // latch
}
```

실습2-4 : Row 차례로 점등

12

파일명 : 8x8LED_ex1.c

```
void set16(uint16_t value)
{
    int i;

    for(i = 0 ; i < 16 ; i++){
        int mask = 0b1 << i;
        if((value & mask) == 0){
            digitalWrite(SDATA, 0);
        }
        else{
            digitalWrite(SDATA, 1);
        }
        digitalWrite(SHR_CLK, 0); //
        digitalWrite(SHR_CLK, 1); //
    }
    // latch
    digitalWrite(STR_CLK, 0); //
    digitalWrite(STR_CLK, 1); //
}
```

실습2-5 : Row 차례로 점등

13

파일명 : 8x8LED_ex1.c

```
int
main(void)
{
    int i;
    uint8_t row[] = {    0b10000000,
                        0b01000000,
                        0b00100000,
                        0b00010000,
                        0b00001000,
                        0b00000100,
                        0b00000010,
                        0b00000001};

    uint8_t col = 0b00000000;
    uint16_t tmp;
    init();

    for(;;)
        for(i = 0 ; i < 8 ; i++){
            tmp = (row[i]<<8) | col;
            set16(tmp);
            delay(100);
        }
    return 1;
}
```

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi 8x8LED_ex1.c -o 8x8LED_ex1
```

```
pi@robotcode ~ $ sudo ./8x8LED_ex1
```

<https://youtu.be/FWJkJ5ZABiY>

실습3-1 : 홀수 Row 차례로 점등 1

14

파일명 : 8x8LED_ex2.c

```
int
main(void)
{
    int i;
    uint8_t row[] = {    0b10000000,
                        0b01000000,
                        0b00100000,
                        0b00010000,
                        0b00001000,
                        0b00000100,
                        0b00000010,
                        0b00000001};

    uint8_t col = 0b00000000;
    uint16_t tmp;
    init();

    for(;;)
        for(i = 0 ; i < 8 ; i+=2){
            tmp = (row[i]<<8) | col;
            set16(tmp);
            delay(100);
        }
    return 1;
}
```

<https://youtu.be/jKjg4qwi41A><https://youtu.be/MVj7QRxIYdc><https://youtu.be/wRt01XJO5Wo>

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi 8x8LED_ex2.c -o 8x8LED_ex2
pi@robotcode ~ $ sudo ./8x8LED_ex2
```


실습4 : + 모양 점등 1

15

파일명 : 8x8LED_ex3.c

```

int
main(void)
{
    int i;

    uint16_t tmp;
    uint8_t rows[] = { 0b10000000,
                       0b01000000,
                       0b00100000,
                       0b00010000,
                       0b00001000,
                       0b00000100,
                       0b00000010,
                       0b00000001};

    uint8_t cols[] = { ~0b00011000,
                       ~0b00011000,
                       ~0b00011000,
                       ~0b11111111,
                       ~0b11111111,
                       ~0b00011000,
                       ~0b00011000,
                       ~0b00011000};

    init();
    while(1)
        for(i = 0 ; i < 8 ; i++){
            tmp = (rows[i]<<8) | cols[i];
            set16(tmp);
            delay(300);
        }
    return 1;
}

```

<https://youtu.be/Ay3qbk9u-KE>
<https://youtu.be/64XFm07rVB0>

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi 8x8LED_ex3.c -o 8x8LED_ex3
```

```
pi@robotcode ~ $ sudo ./8x8LED_ex3
```

실습5 : + 모양 점등 2

16

```

void dot(int row, int col)
{
    uint8 t row8, col8;
    uint16 t tmp;
    row8 = 1 << (8-row);
    col8 = ~(1 << (8-col));
    tmp = (row8<<8) | col8;
    set16(tmp);
}

int main(void)
{
    int i,j;

    char mat[8][8] = {
        0,0,0,1,1,0,0,0,
        0,0,0,1,1,0,0,0,
        0,0,0,1,1,0,0,0,
        1,1,1,1,1,1,1,1,
        1,1,1,1,1,1,1,1,
        0,0,0,1,1,0,0,0,
        0,0,0,1,1,0,0,0,
        0,0,0,1,1,0,0,0};

    init();
    while(1)
        for(i = 0 ; i < 8 ; i++)
            for(j = 0 ; j < 8 ; j++)
                if(mat[i][j] == 1)
                    dot(i+1,j+1);

    return 1;
}

```

파일명 : 8x8LED_ex4.c

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi 8x8LED_ex4.c -o 8x8LED_ex4
```

```
pi@robotcode ~ $ sudo ./8x8LED_ex4
```

https://youtu.be/b7_NeCGxnYc

실습6 : ♥ 모양 점등

17

파일명 : 8x8LED_ex5.c

```

int cnt;
void * thr_fn1(void *arg)
{
    while(1){
        usleep(500000);
        cnt++;
    }
    return((void *)0);
}

int main(void)
{
    int i,j;
    pthread_t tid1;
    char mat[8][8] = {
        0,0,0,0,0,0,0,0,
        0,1,1,0,0,1,1,0,
        1,1,1,1,1,1,1,1,
        1,1,1,1,1,1,1,1,
        1,1,1,1,1,1,1,1,
        0,1,1,1,1,1,1,0,
        0,0,1,1,1,1,0,0,
        0,0,0,1,1,0,0,0};

    init();
    pthread_create(&tid1, NULL, thr_fn1, NULL);
    while(1){
        for(i = 0 ; i < 8 ; i++){
            for(j = 0 ; j < 8 ; j++){
                if(mat[i][j] == cnt%2) dot(i+1,j+1);
            }
        }
        return 1;
    }
}

```

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi -lpthread 8x8LED_ex5.c -o 8x8LED_ex5
```

```
pi@robotcode ~ $ sudo ./8x8LED_ex5
```

<https://youtu.be/RDnx4STM530>

실습7-1 : 2차원 배열 review

18

파일명 : 2dimArray.c

```
// How to dynamically allocate a 2D array

#include <stdio.h>
#include <stdlib.h>
#define ROW 8
#define COL 8

char mat[ROW][COL]={
    0,0,0,0,0,0,0,0,
    0,1,1,0,0,1,1,0,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    1,1,1,1,1,1,1,1,
    0,1,1,1,1,1,1,0,
    0,0,1,1,1,1,0,0,
    0,0,0,1,1,0,0,0};

void method1(); // Using a single pointer
void method2(); // Using an array of pointers
void method3(); // Using pointer to a pointer
void method4(); // Using double pointer and one malloc call for all rows

int main(int argc, char *argv[]){
    int i, j;

    printf("[mat]\n");

    for(i = 0 ; i < ROW ; i++){
        for(j = 0 ; j < COL ; j++){
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }

    method1();      method2();      method3();      method4();

    return 0;
}
```

실습7-2 : 2차원 배열 review

19

파일명 : 2dimArray.c

```
// Using a single pointer
void method1()
{
    int i, j;
    char *matcopy;

    matcopy = (char*)malloc(ROW * COL * sizeof(char));

    for(i = 0 ; i < ROW ; i++)
        for(j = 0 ; j < COL ; j++)
            *(matcopy + (i * COL) + j) = mat[i][j];

    printf("\nmethod1\n");

    for(i = 0 ; i < ROW ; i++){
        for(j = 0 ; j < COL ; j++)
            printf("%d ", *(matcopy + (i * COL) + j));
        printf("\n");
    }
    free(matcopy);
}
```

실습7-3 : 2차원 배열 review

20

파일명 : 2dimArray.c

```
// Using an array of pointers
void method2()
{
    int i, j;
    char *matcopy[ROW];

    for(i = 0 ; i < ROW ; i++)
        matcopy[i] = (char*)malloc(COL * sizeof(char));

    for(i = 0 ; i < ROW ; i++)
        for(j = 0 ; j < COL ; j++)
//            *(matcopy[i] + j) = mat[i][j];
            matcopy[i][j] = mat[i][j];

    printf("\nmethod2\n");

    for(i = 0 ; i < ROW ; i++){
        for(j = 0 ; j < COL ; j++)
//            printf("%d ", *(matcopy[i]+j));
            printf("%d ", matcopy[i][j]);
        printf("\n");
    }
}
```


실습7-4 : 2차원 배열 review

21

파일명 : 2dimArray.c

```
// Using pointer to a pointer
void method3()
{
    int i, j;
    char **matcopy;

    matcopy = (char**)malloc(ROW * sizeof(char*));

    for(i = 0 ; i < ROW ; i++)
//      *(matcopy+i) = (char*)malloc(COL * sizeof(char));
        matcopy[i] = (char*)malloc(COL * sizeof(char));

    for(i = 0 ; i < ROW ; i++)
        for(j = 0 ; j < COL ; j++)
//          (*(matcopy+i) + j) = mat[i][j];
            matcopy[i][j] = mat[i][j];

    printf("\nmethod3\n");

    for(i = 0 ; i < ROW ; i++){
        for(j = 0 ; j < COL ; j++)
//          printf("%d ", (*(matcopy + i) + j));
            printf("%d ", matcopy[i][j]);
        printf("\n");
    }
}
```

실습7-5 : 2차원 배열 review

22

파일명 : 2dimArray.c

```
// Using double pointer and one malloc call for all rows
void method4()
{
    int i, j;
    char **matcopy;

    matcopy = (char**)malloc(ROW * sizeof(char*));

    // *(matcopy + 0) = (char*)malloc(ROW * COL * sizeof(char*));
    matcopy[0] = (char*)malloc(ROW * COL * sizeof(char*));

    for(i = 0 ; i < ROW ; i++)
        *(matcopy + i) = *matcopy + COL * i;
        matcopy[i] = *matcopy + COL * i;

    for(i = 0 ; i < ROW ; i++)
        for(j = 0 ; j < COL ; j++)
            // *(*(matcopy + i) + j) = mat[i][j];
            matcopy[i][j] = mat[i][j];

    printf("\nmethod4\n");

    for(i = 0 ; i < ROW ; i++){
        for(j = 0 ; j < COL ; j++)
            // printf("%d ", *(*(matcopy + i) + j));
            printf("%d ", matcopy[i][j]);
        printf("\n");
    }
}
```

```
$ ./makeFont_ex1  
.  
o o o .  
o . . . o  
o . . . o  
o . . . o  
o o o o o  
o . . . o  
o . . . o  
o . . . o  
o . . . o  
$
```

```
$ ./makeFont_ex2
. o o o .
o . . . o
o . . . o
o . . . o
o o o o o
o . . . o
o . . . o
o . . . o

o o o o .
o . . . o
o . . . o
o o o o .
o . . . o
o . . . o
o . . . o
o o o o .

. o o o .
o . . . o
o . . . .
o . . . .
o . . . .
o . . . o
. o o o .

o o o . .
o . . o .
o . . . o
o . . . o
o . . . o
o . . . o
o . . o .
o o o . .

o o o o o
o . . . .
o . . . .
o o o o .
o . . . .
o . . . .
o . . . .
o o o o o

$
```

실습10-1 : 5x8 Dot 문자열 출력

25

파일명 : makeFont_ex3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
#define FONT_COL 5          // 폰트의 Column
#define BOARD_ROW 8        // board의 Row

#define INTERVAL 1;        // 문자간의 기본간격

// 8x5 font 데이터
uint8_t font85[][FONT_COL] = {
    0x7F, 0x88, 0x88, 0x88, 0x7F, // A
    0xFF, 0x91, 0x91, 0x91, 0x6E, // B
    0x7E, 0x81, 0x81, 0x81, 0x42, // C
    0xFF, 0x81, 0x81, 0x42, 0x3C, // D
    0xFF, 0x91, 0x91, 0x91, 0x81}; // E

// board
uint8_t *board[BOARD_ROW];
int      board_size;

// board에 문자 writing, blank-이전 문자와의 간격
void boardWriter(int fontidx, int blank);
```

실습10-2 : 5x8 Dot 문자열 출력

26

파일명 : makeFont_ex3.c

```

int main(void)
{
    int i, j, total;

    // # of font data
    total = sizeof(font85) / FONT_COL / sizeof(uint8_t);

    boardWriter(0,10);    // 문자 a, 앞 여백 10
    boardWriter(1,0);     // 문자 b, 앞 여백 0
    boardWriter(2,0);     // 문자 c, 앞 여백 0
    boardWriter(3,0);     // 문자 d, 앞 여백 0
    boardWriter(4,0);     // 문자 e, 앞 여백 0
    boardWriter(3,5);     // 문자 d, 앞 여백 20
    boardWriter(2,0);     // 문자 c, 앞 여백 0
    boardWriter(1,0);     // 문자 b, 앞 여백 0
    boardWriter(0,0);     // 문자 a, 앞 여백 0

    printf("font data : %d\n", total);
    printf("board info : %d X %d\n", board_size, BOARD_ROW);
    printf("< B O A R D >\n");
    for(i = 0 ; i < BOARD_ROW ; i++){
        for(j = 0 ; j < board_size ; j++){
            printf("%c ", board[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```


실습10-3 : 5x8 Dot 문자열 출력

27

```
void boardWriter(int fontidx, int blank)
```

파일명 : makeFont_ex3.c

```
{
    int i, j, l;
    static int board_offset = 0; // 현재 문자의 offset
    uint8_t mask;

    if(blank)    board_offset += blank;

    // memory reallocation
    board_size = board_offset + FONT_COL + INTERVAL;

    for(i = 0 ; i < BOARD_ROW ; i++)
        board[i] = (uint8_t*)realloc(board[i], board_size);

    // writing on the board
    for(i = 7, l = 0 ; i >= 0 ; i--, l++){           // row
        mask = 0b1 << i;
        for(j = 0 ; j < FONT_COL ; j++){             // col
            if((mask & font85[fontidx][j])){
                board[l][j+board_offset] = 'o';
            } else {
                board[l][j+board_offset] = ' ';
            }
        }
    }
    board_offset = board_size;;
}
```

실습10-4 : 5x8 Dot 문자열 출력

28

파일명 : makeFont_ex3.c

```
shchoi82@shchoi82:~$ ./makeFont_ex3
```

```
font data : 5
```

```
board info : 69 X 8
```

```
< B O A R D >
```

```

      o o o   o o o o   o o o   o o o   o o o o o   o o o   o o o   o o o o   o o o
    o       o o       o o       o o       o       o       o       o       o       o
    o       o o       o o       o       o o       o       o       o       o       o
    o       o o o o o   o       o       o o o o o   o       o       o       o
    o o o o o o       o o       o       o       o       o       o       o o o o o o
    o       o o       o o       o       o o       o       o       o       o       o
    o       o o       o o       o o       o       o       o       o       o       o
    o       o o o o o   o o o   o o o   o o o o o   o o o   o o o   o o o o   o

```

```
shchoi82@shchoi82:~$
```

실습11-1 : 문자열 스크롤

29

파일명 : 8x8LED_ex6.c

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPi.h>
#include <stdint.h>
#include <pthread.h>
#include <unistd.h>

#define SDATA          6      // Serial Input Data
#define STR_CLK        13     // Storage Register Clock(LATCH)
#define SHR_CLK        19     // Shift Register Clock
#define SHR_CLEAR      26     // Shift Register Clear

void allclear(void);

void init(void)
{
    if(wiringPiSetupGpio() == -1){ // wiringPi
        fprintf(stderr, "wiringPiSetupGpio() error\n");
        exit(1);
    }
    pinMode(SDATA, OUTPUT);
    pinMode(STR_CLK, OUTPUT);
    pinMode(SHR_CLK, OUTPUT);
    pinMode(SHR_CLEAR, OUTPUT);
    allclear();
}
```

실습11-2 : 문자열 스크롤

30

파일명 : 8x8LED_ex6.c

```
void allclear(void)
{
    digitalWrite(SHR_CLEAR, 0);
    digitalWrite(SHR_CLEAR, 1);
    digitalWrite(STR_CLK, 0);
    digitalWrite(STR_CLK, 1);    // latch
}

void set16(uint16_t value)
{
    int i;

    for(i = 0 ; i < 16 ; i++){
        int mask = 0b1 << i;
        if((value & mask) == 0){
            digitalWrite(SDATA, 0);
        }
        else{
            digitalWrite(SDATA, 1);
        }
        digitalWrite(SHR_CLK, 0); //
        digitalWrite(SHR_CLK, 1); //
    }
    // latch
    digitalWrite(STR_CLK, 0); //
    digitalWrite(STR_CLK, 1); //
}
```

실습11-3 : 문자열 스크롤

31

파일명 : 8x8LED_ex6.c

```
void dot(int row, int col)
{
    uint8_t row8, col8;
    uint16_t tmp;
    row8 = 1 << (8-row);
    col8 = ~(1 << (8-col));
    tmp = (row8<<8) | col8;
    set16(tmp);
}

int cnt;
void * thr_fn1(void *arg)
{
    while(1){
        usleep(200000);
        cnt++;
    }
    return((void *)0);
}
```

실습11-4 : 문자열 스크롤

32

파일명 : 8x8LED_ex6.c

```
#define ROW 8
#define COL 8
int main(void)
{
    int i, j;
    int col;
    pthread_t tid;

    uint8_t mat[ROW][24] = {
        0,0,1,1,1,0,0,0,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,
        0,1,0,0,0,1,0,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,
        1,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,
        1,0,0,0,0,0,1,0,1,1,1,1,1,1,1,0,0,1,0,0,0,0,0,0,
        1,1,1,1,1,1,1,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,
        1,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,
        1,0,0,0,0,0,1,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1,1,0,0,
        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

    col = sizeof(mat) / ROW;

    init();
    pthread_create(&tid, NULL, thr_fn1, NULL);

    while(1)
        for(j = 0 ; j < COL ; j++)
            for(i = 0 ; i < ROW ; i++)
                if(mat[i][(j+cnt)%col] == 1)
                    dot(i+1,j+1);

    return 1;
}
```

```
pi@robotcode ~ $ gcc -Wall -W -lwiringPi -lpthread 8x8LED_ex6.c -o 8x8LED_ex6
pi@robotcode ~ $ sudo ./8x8LED_ex6
```

https://youtu.be/l4q1voZR_7M

미션

33

- **키보드로부터 문자열을 입력 받아 5x8 문자로 변환하시오.**
 - a~z, A~Z, 0~9 문자만 허용
 - 표준출력으로 출력
 - https://youtu.be/8EK04Ty_fgw
- **5x8 문자 파일을 읽어 8x8 LED Dot Matrix 로 출력**
 - 명령줄 첫번째 인자로 파일명을 입력받음
 - 좌에서 우로 스크롤링
 - <https://youtu.be/w6R3TjJInf8>