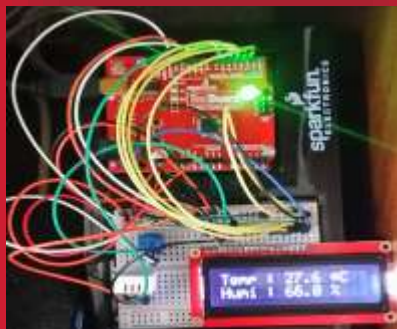
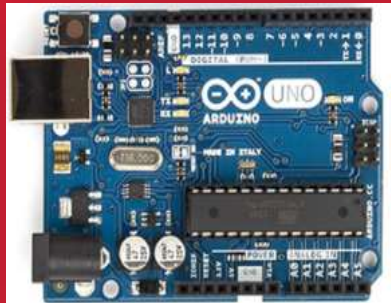




# Arduino-basic

## [wk08]

## LED – IV : DM



Learn how to code Arduino from scratch

Comsi, INJE University

1<sup>st</sup> semester, 2022

Email : [chaos21c@gmail.com](mailto:chaos21c@gmail.com)



# My ID (ARnn, github repo)

- [AR01 김정헌]
- [AR02 유석진]
- [AR03 김기덕]
- [AR04 강대진]
- [AR05 김성우]
- [AR06 김창연]
- [AR07 김창욱]
- [AR08 김태화]
- [AR09 박세훈]
- [AR10 박신영]

- [AR11 박제홍]
- [AR12 이승무]
- [AR13 이승준]
- [AR14 이재하]
- [AR15 이준희]
- [AR16 이현준]
- [AR17 임태형]
- [AR18 정동현]
- [AR19 정희서]
- [AR20 이한글]
- [AR21 황혁준]
- [AR22 김동영]

# wk06 : Practice-05 : ARnn\_Rpt05

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

제출폴더명 : **ARnn\_Rpt05**

### 제출할 파일들

- ① **ARnn\_4digit.fzz**
- ② **ex\_4\_5\_1.ino**
- ③ **ARnn\_4digit.ino**
- ④ **ARnn\_4digit\_9999.ino**
- ⑤ **ARnn\_4digit\_9999.png**

# wk07 : Practice-exam : ARnn\_Rpt06

## ◆ [Target of this week]

1. 4-digit FND 회로 완성
2. 0000, ... , 9999 출력 확인

-----

3. 직렬 통신으로 전송 받은  
4자리 수 출력 (사진 촬영)
4. Arnn\_Rpt06.zip을 email:  
[chaos21c@gmail.com](mailto:chaos21c@gmail.com)  
으로 전송

제출폴더명 : **ARnn\_Rpt06**

제출할 파일들

① **ARnn\_mid\_ex.ino**

② **ARnn\_mid\_ex.png**

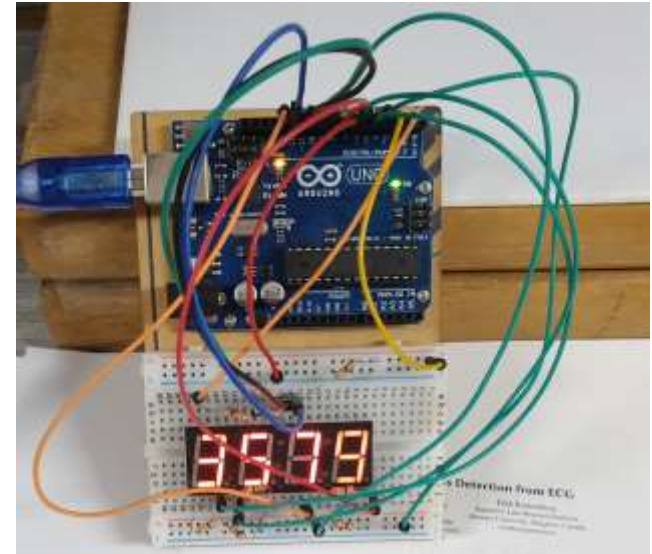
# wk07 : Practice-exam : ARnn\_Rpt06

## ◆ [RESULT]

COM3

```

전송할 4 자리 정수를 입력: 1234
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
전송할 4 자리 정수를 입력: 4567
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
전송할 4 자리 정수를 입력: 678
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
전송할 4 자리 정수를 입력: 7890
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
전송할 4 자리 정수를 입력: 2579
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
전송할 4 자리 정수를 입력: 3579
전송할 4 자리 정수를 입력: 4 자리 이하 정수를 입력하세요
  
```



# wk07 : Practice-exam : ARnn\_Rpt06

```

1 // pin no. connected to segment
2 int segpin[8] = {2,3,4,5,6,7,8,9};
3 //      a, b, c, d, e, f, g, h
4
5 // pin no. connected to digit
6 int dpin[4] = {10, 11, 12, 13};
7 //      pin  d1,d2,d3,d4
8
9 //      segment      a, b, c, d, e, f, g, h      number
10 const int arr[10][8] = {
11     {1, 1, 1, 1, 1, 1, 0, 0}, //0
12     {0, 1, 1, 0, 0, 0, 0, 0}, //1
13     {1, 1, 0, 1, 1, 0, 1, 0}, //2
14     {1, 1, 1, 1, 0, 0, 1, 0}, //3
15     {0, 1, 1, 0, 0, 1, 1, 0}, //4
16     {1, 0, 1, 1, 0, 1, 1, 0}, //5
17     {1, 0, 1, 1, 1, 1, 1, 0}, //6
18     {1, 1, 1, 0, 0, 0, 0, 0}, //7
19     {1, 1, 1, 1, 1, 1, 1, 0}, //8
20     {1, 1, 1, 1, 0, 1, 1, 0}, //9
21 };
22 int timer=200;
23 int dn[4];
24
25 void setup()
26 {
27     Serial.begin(9600); // baud
28
29     for (int i=0; i<=7; i++)
30     { pinMode(segpin[i], OUTPUT); }
31
32     for(int j=0;j<4;j++)
33     { pinMode(dpin[j], OUTPUT); }
34 }

```

```

36 void loop()
37 {
38     Serial.print("전송할 4 자리 정수를 입력: ");
39
40     while(Serial.available()== 0) { }
41
42     int num = (Serial.parseInt());
43     if(num>0) {
44         Serial.println(num);
45
46         dn[0] = num %10 ;           //digit at ones place
47         dn[1] = (num % 100) / 10 ; //digit at tens place
48         dn[2] = (num % 1000) / 100; //digit at hundreds place
49         dn[3] = num / 1000;        //digit at thousands place
50         delay(5);
51
52         for(int k=0; k<timer; k++)
53         {
54             for(int i=3; i=0; i--){
55                 fnd4display(i);
56             }
57         }
58         for(int j=0;j<4;j++)
59         { digitalWrite(dpin[j], HIGH); }
60     }
61     else{
62         Serial.println("4 자리 이하 정수를 입력하세요");
63     }
64 }

```



# 4. LED IV

## DM & DM module

4 1EA



7세그먼트 1채널

---

공통 음극 7세그먼트  
시계나 점수 등의 숫자를  
표현 할 때 많이 사용됩니다.

5 1EA



74HC595N

---

기본 메인보드입니다.  
74HC595N LED,  
도트매트릭스, NFD 제어 IC 입니다.

3 1EA



7세그먼트 4채널

---

7세그먼트가 4개 연결된 형태의  
부품입니다.  
총 12개의 핀을 사용합니다.

23 1EA



8x8 도트매트릭스 모듈

---

LED로 다양한 연출을  
할 수 있습니다.

8 X 8 Dot matrix



그림 4.8 실험에 사용할 도트매트릭스



## 8 X 8 Dot matrix

- ✓ 여러 개의 LED가 배열되어 문자나 기호를 표시하는 장치
- ✓ 8X8 Dot matrix는 64개의 LED를 이용
- ✓ LED를 빠르게 교차 출력하여 동시에 모든 LED가 제어되는 듯한 착시를 이용

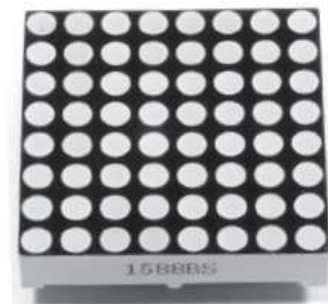


그림 4.8 실험에 사용할 도트매트릭스

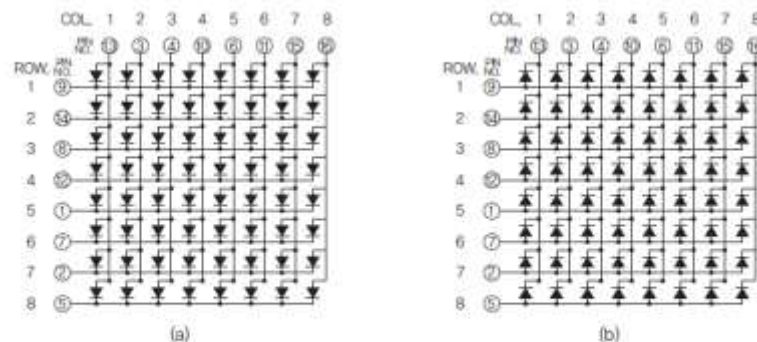
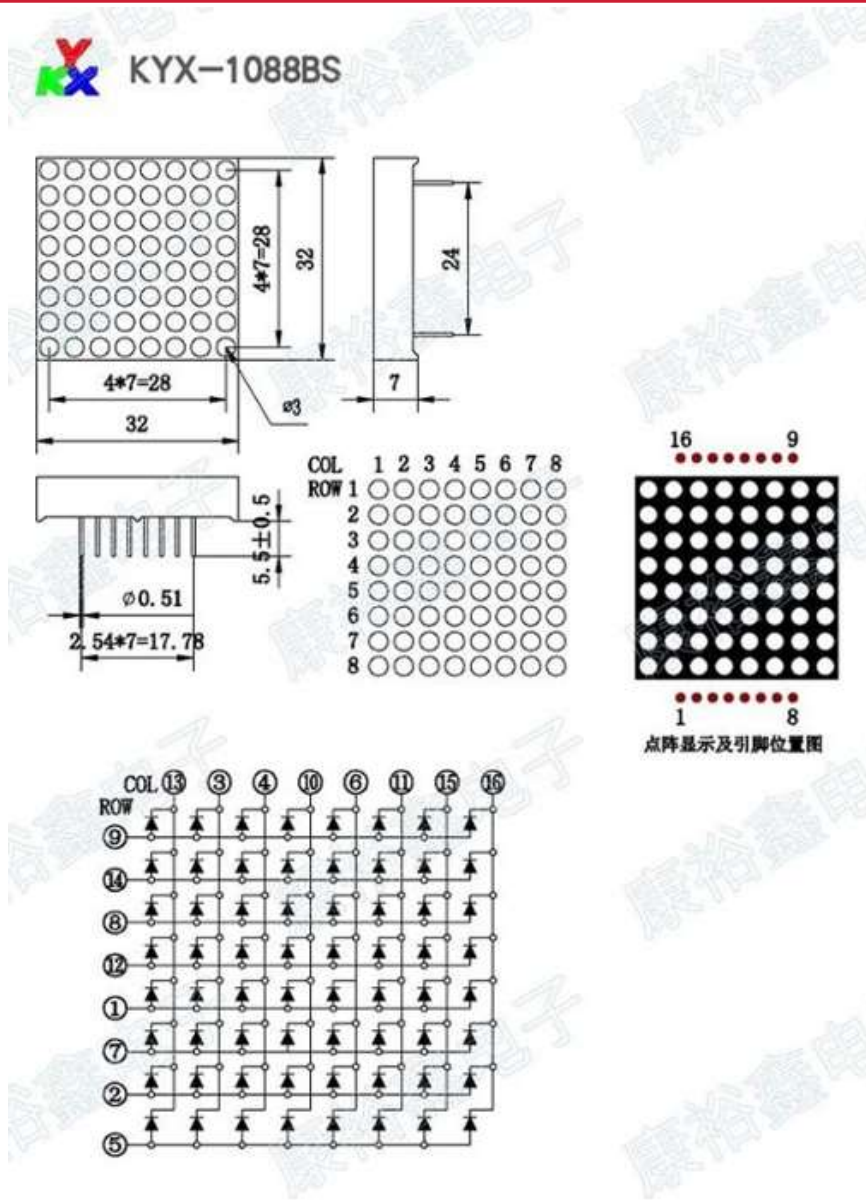


그림 4.9 행에 Anode(+연결)를 연결하고 열에 Cathode(-연결)를 연결한 형태(a)와 행에 Cathode(-연결)를 연결하고 열에 Anode(+연결)한 형태(b).

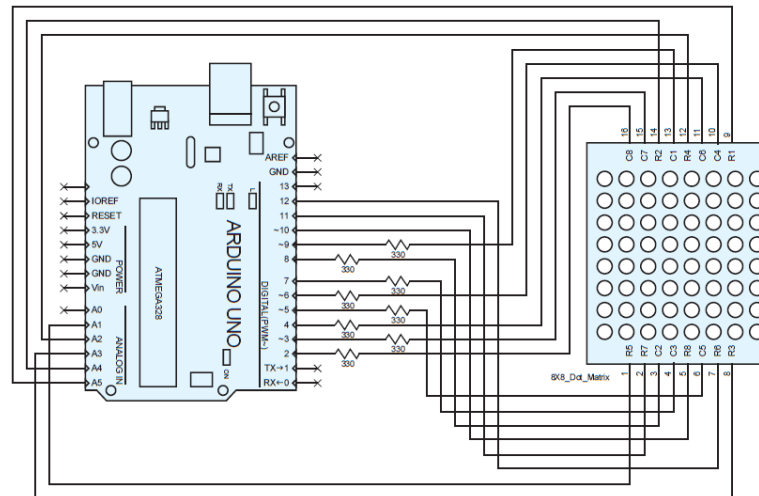


## EX 4.6

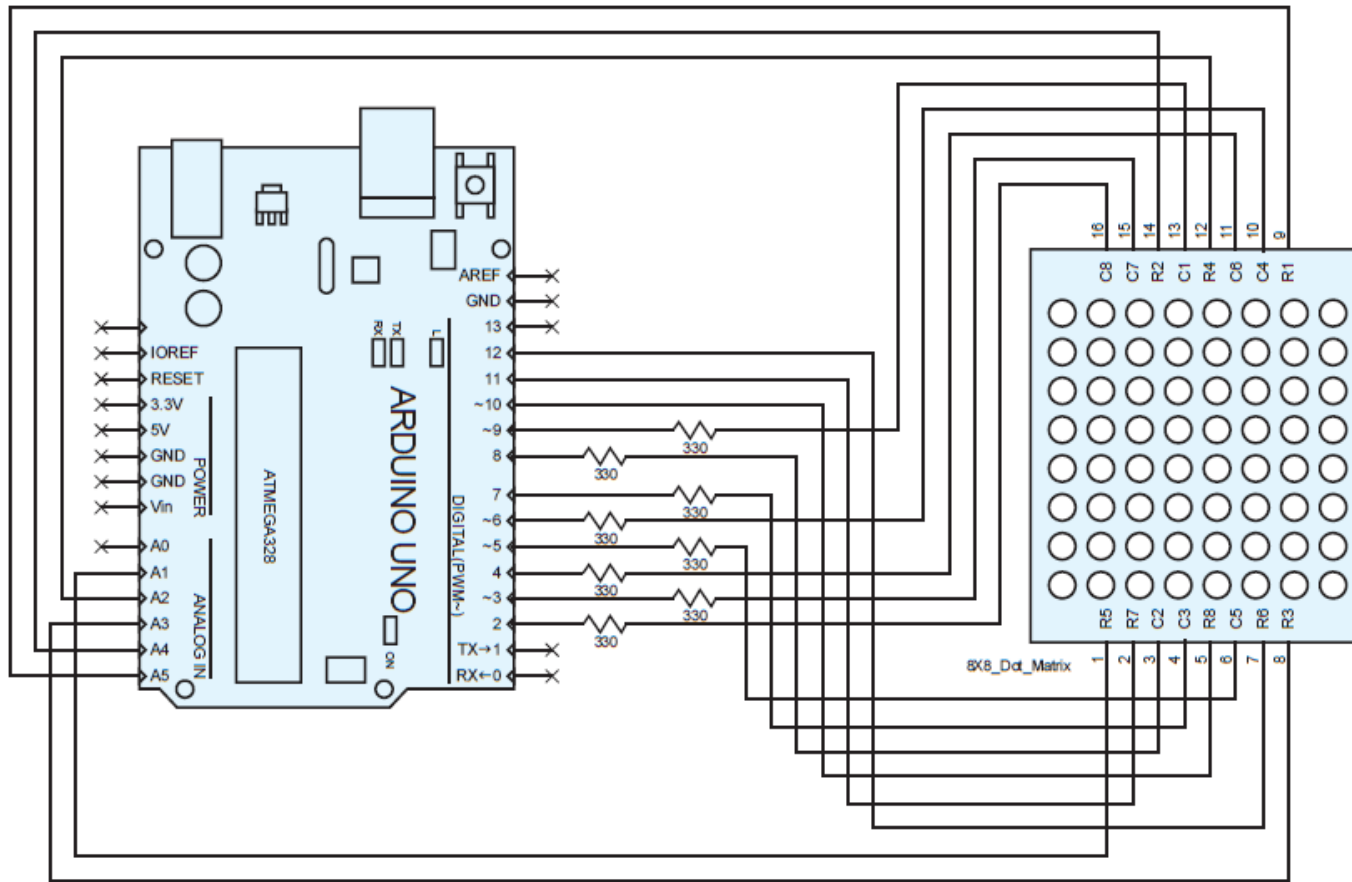
## Dot matrix 제어 (1/3)

**실습목표** 8x8 Dot matrix로 변화하는 막대그래프를 표현해 보자.

- Hardware**
1. 행은 2~9번핀에 연결하고 열은 10, 11, 12, A1~A5 번핀에 연결한다. 행을 연결할 때는 220 $\Omega$ 저항을 함께 연결한다.
  2. 실험에 사용할 8X8 Dot matrix는 행(column)에 Cathode, 열(row)에 Anode가 연결된 형태이다. 즉 행에 LOW신호, 열에 HIGH신호를 주어야 Dot LED가 켜진다.
  3. 특정 부분의 Dot LED를 점등하려면 그 부분의 행에 LOW신호, 열에 HIGH신호를 준다.



# 4.9.2 Dot matrix 제어



C1	C2	C3	C4	C5	C6	C7	C8
9	8	7	6	5	4	3	2
R1	R2	R3	R4	R5	R6	R7	R8
A5	A4	A3	A2	A1	12	11	10

## EX 4.6

## Dot matrix 제어 (2/3)

### Commands

- void 함수(변수1, 변수2, ...){  
};

'함수(변수1, 변수2)' 를 이용하여 '{ }' 내의 명령을 호출하여 사용한다. '변수1'과 '변수2'등을 함께 선언하면 함수 내에서 그 변수를 사용할 수 있다. 반복되는 구문을 설정해 놓고 호출하여 사용하면 편리하다.

- pinMode(핀번호, 설정)

핀의 입출력 모드를 설정한다. '핀번호'에는 설정하고자 하는 핀의 번호와 '설정'에는 입력으로 사용하기 위해선 'INPUT', 출력으로 사용하기 위해선 'OUTPUT', 입력이며 풀업 사용시 'INPUT\_PULLUP'을 설정한다.

- digitalWrite(핀번호, 값)

핀에 디지털 출력 (High or Low) 을 한다. '핀번호'에는 출력하고자 하는 핀의 번호를, '값'에는 'HIGH' 혹은 'LOW' 를 설정하여 High 혹은 Low 출력을 한다.

- for(변수=시작 값 ; 조건 ; 변수의 증분){    }

변수의 시작 값부터 조건이 만족하는 경우 '{    }' 내의 명령을 수행한다. '변수의 증분'에서는 1회 명령이 수행될 때 마다 변수를 증가 혹은 감소시킨다.

### EX 4.6

### Dot matrix 제어 (3/3)

- Sketch 구성
1. 8X8 Dot matrix는 그림 4.5의 (b) 그림의 행에 Cathode, 열에 Anode가 연결된 형태를 사용할 것이다.
  2. 행과 열에 출력에 사용할 핀을 모두 출력으로 설정한다.
  3. 점등하고자 하는 행에 LOW 신호를 준 뒤 열에 HIGH 신호를 주어 LED를 점등시킨다.
  4. 행을 하나씩 증가하여 점등시킨다.

실행 결과 C8 부터 C1로 한 칸씩 이동하면서 쌓이는 막대그래프가 출력된다.

응용 문제 Dot가 한 개씩 이동하는 스케치를 만들어보자.

ex\_4\_6\_start

```

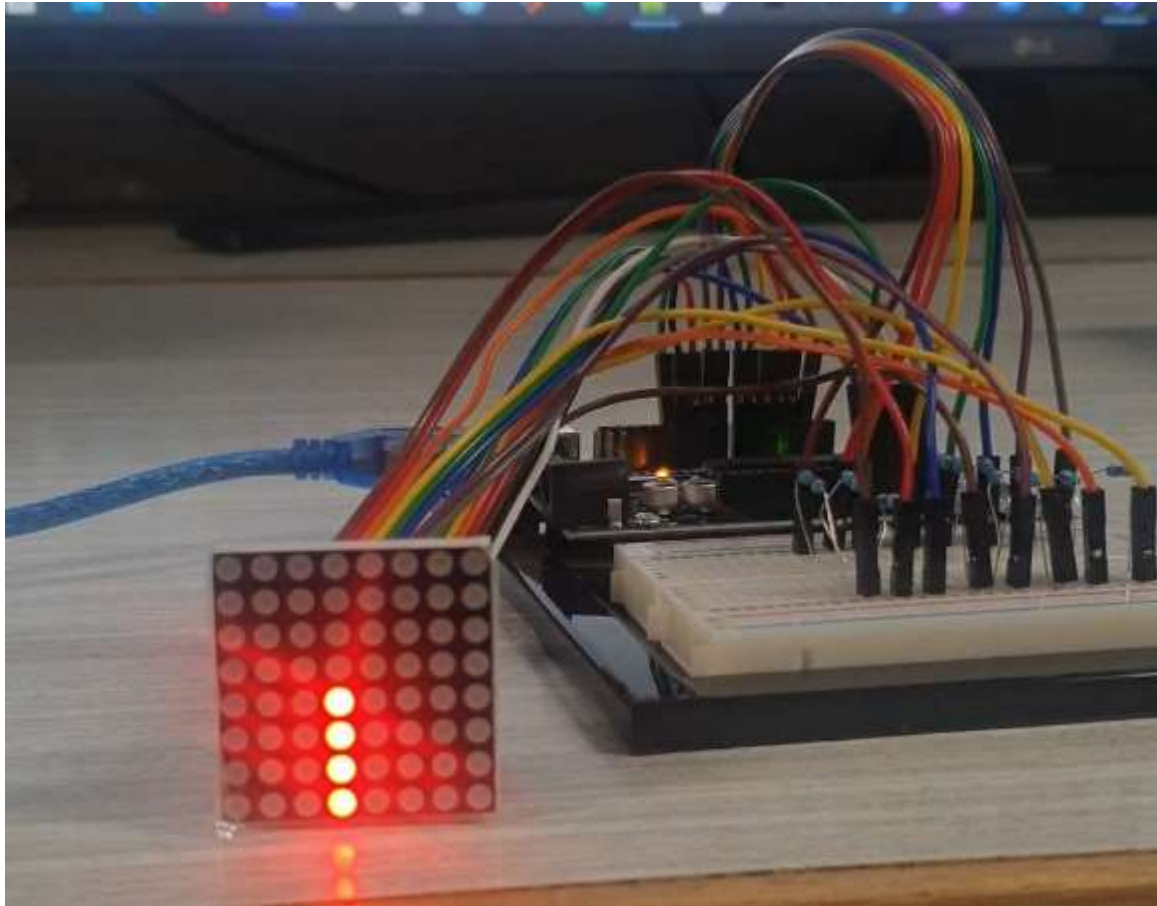
5
6 const int colPins[] = { 2, 3, 4, 5, 6, 7, 8, 9};
7 //          C8, C7, C6, C5, C4, C3, C2, C1
8 const int rowPins[] = { 10,11,12,15,16,17,18,19};
9 //          R8, R7, R6, R5, R4, R3, R2, R1
10
11 void setup() {
12   for (int i = 0; i < 8; i++)
13   {
14     // 행을 출력으로 설정한다
15     pinMode(colPins[i], OUTPUT);
16     // 열을 출력으로 설정한다
17     pinMode(rowPins[i], OUTPUT);
18   }
19 }
20

```

```

21 void loop() {
22
23   for (int column = 0; column < 8; column++)
24   {
25     // 행을 모두 초기화 한다
26     colClear();
27     // 현재의 행만 켜다
28     digitalWrite(colPins[column], LOW);
29
30     for(int row = 0; row < 8; row++)
31     {
32       // 열을 하나씩 켜다
33       digitalWrite(rowPins[row], HIGH);
34       delay(100);
35     }
36     // 열을 모두 초기화 한다
37     rowClear();
38   }
39   // 모든 행을 반복했으면 열을 모두 소등한다
40   rowClear();
41 }
42
43 // 행을 모두 초기화하는 루틴
44 void colClear(){
45   for(int i = 0; i < 8; i++){
46     digitalWrite(colPins[i], HIGH);
47   }
48 }
49
50 // 열을 모두 초기화하는 루틴
51 void rowClear(){
52   for(int i = 0; i < 8; i++){
53     digitalWrite(rowPins[i], LOW);
54   }
55 }

```

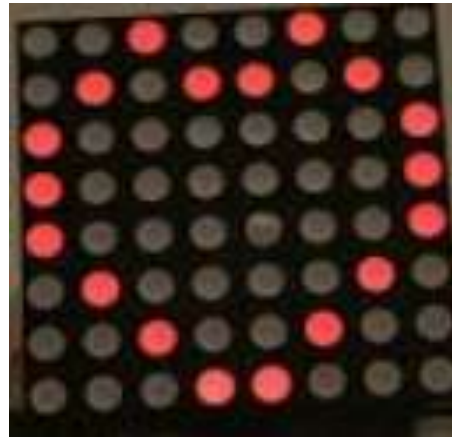


[ARnn\\_dm.png](#)

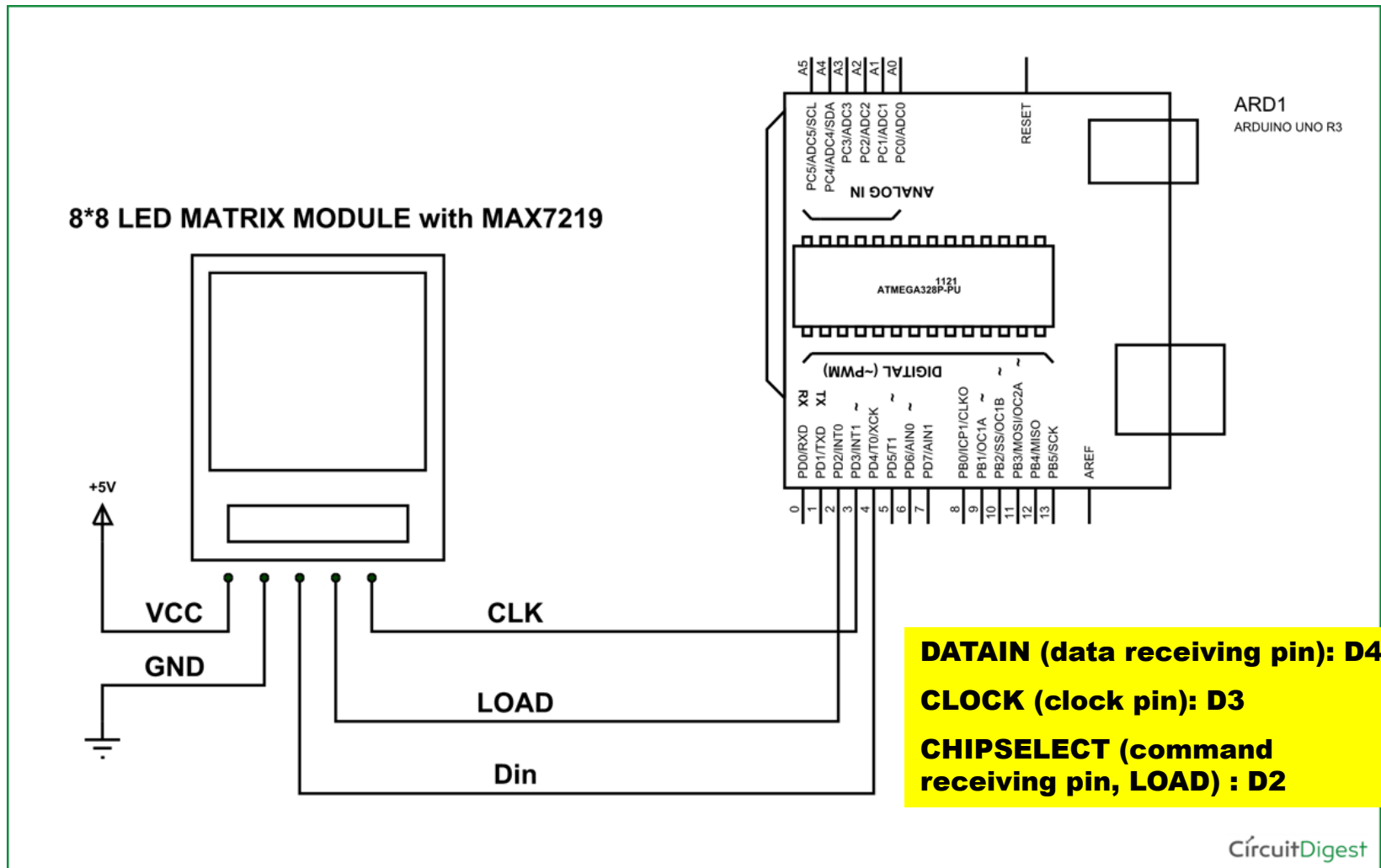
로 저장해서 제출.



8 X 8 Dot matrix

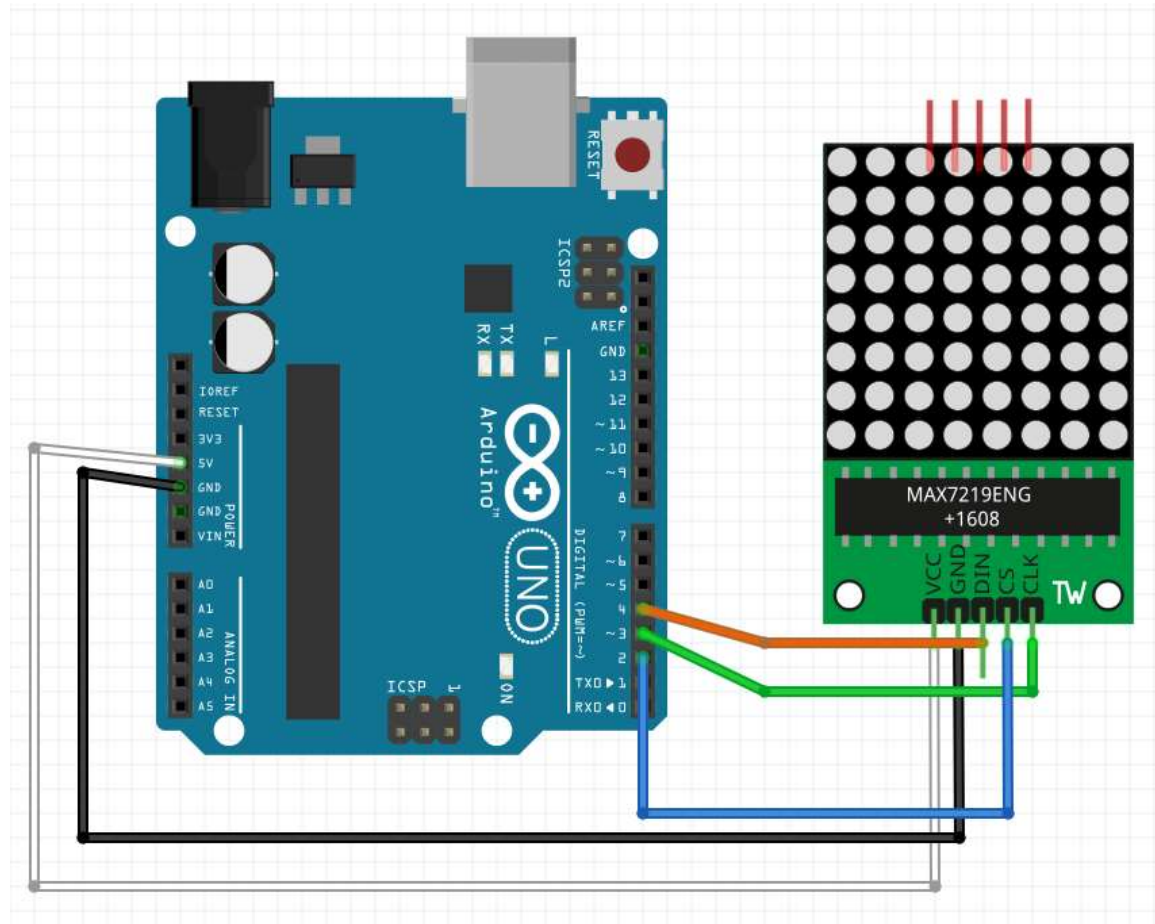


# 4.9.4 Dot matrix module 제어



<https://circuitdigest.com/microcontroller-projects/arduino-8x8-led-matrix>

# 4.9.4.1 Dot matrix module 제어



<http://forum.fritzing.org/t/max7219-dot-matrix-led-modul/1914>

Fritzing 으로 회로를 디자인하고

ARnn\_dm\_module.fzz 로 저장.

MAX7219 Dot Matrix Led Module 부품을 Fritzing에 연결

## 4.9.4.2 Dot matrix 제어 – code1

ex\_4\_7\_dm\_start

```
1 #include "LedControlMS.h"
2
3 // pin 4 is connected to the DataIn
4 // pin 3 is connected to the CLK
5 // pin 2 is connected to LOAD
6
7 #define NBR_MTX 1 //number of matrices attached is one
8 LedControl lc=LedControl(4,3,2, NBR_MTX);//
9
```



```
10 void setup()
11 {
12   for (int i=0; i< NBR_MTX; i++)
13   {
14     lc.shutdown(i,false);
15     /* Set the brightness to a medium values */
16     lc.setIntensity(i,8);
17     /* and clear the display */
18     lc.clearDisplay(i);
19
20     delay(100);
21   }
22 }
23
24 void loop()
25 {
26   //sending characters to display
27   lc.writeString(0,"Hello COMSI, AR00.");
28   //clearing the display
29   lc.clearAll();
30
31   delay(3000);
32 }
```

# 4.9.4.3 Dot matrix 제어 – code2.1

ex\_dm\_project

```

1 #include "LedControlMS.h"
2
3 #define NBR_MTX 1 //number of matrices attached is one
4
5 LedControl lc=LedControl(4,3,2, NBR_MTX); //
6
7 unsigned long delayTime=1200; // Delay between Frames
8
9 // Put values in arrays
10 byte invader1a[] =
11 {
12   B00011000, //
13   B00111100,
14   B01111110,
15   B11011011,
16   B11111111,
17   B00100100,
18   B01011010,
19   B10100101
20 };
21
22 byte invader1b[] =
23 {
24   B00011000, // Second
25   B00111100,
26   B01111110,
27   B11011011,
28   B11111111,
29   B00100100,
30   B01011010,
31   B01000010
32 };

```



```

34 byte invader2a[] =
35 {
36   B00100100, // First frame of invader #2
37   B00100100,
38   B01111110,
39   B11011011,
40   B11111111,
41   B11111111,
42   B10100101,
43   B00100100
44 };
45
46 byte invader2b[] =
47 {
48   B00100100, // Second frame of invader #2
49   B10100101,
50   B11111111,
51   B11011011,
52   B11111111,
53   B01111110,
54   B00100100,
55   B01000010
56 };

```

```

58 void setup()
59 {
60   lc.shutdown(0,false); // Wake up displays
61   // lc.shutdown(1,false);
62   lc.setIntensity(0,5); // Set intensity levels
63   // lc.setIntensity(1,5);
64   lc.clearDisplay(0); // Clear Displays
65   // lc.clearDisplay(1);
66 }
67
68
69 // Take values in Arrays and Display them
70 void sinvader1a()
71 {
72   for (int i = 0; i < 8; i++)
73   {
74     lc.setRow(0,i,invader1a[i]);
75   }
76 }
77
78 void sinvader1b()
79 {
80   for (int i = 0; i < 8; i++)
81   {
82     lc.setRow(0,i,invader1b[i]);
83   }
84 }

```

```

86 //void sinvader2a()
87 //{
88 //  for (int i = 0; i < 8; i++)
89 //  {
90 //    lc.setRow(1,i,invader2a[i]);
91 //  }
92 //}
93 //
94 //void sinvader2b()
95 //{
96 //  for (int i = 0; i < 8; i++)
97 //  {
98 //    lc.setRow(1,i,invader2b[i]);
99 //  }
100 //}

```

완성된 코드를 **ARnn\_dm\_project.ino**

로 저장해서 제출.

```

102 void loop()
103 {
104   // Put #1 frame on both Display
105   sinvader1a();
106   delay(delayTime);
107   // sinvader2a();
108   // delay(delayTime);
109
110
111   // Put #2 frame on both Display
112   sinvader1b();
113   delay(delayTime);
114   // sinvader2b();
115   // delay(delayTime);
116
117 }

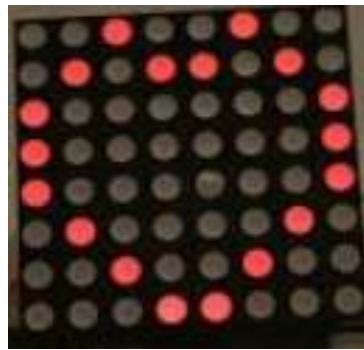
```

DIY

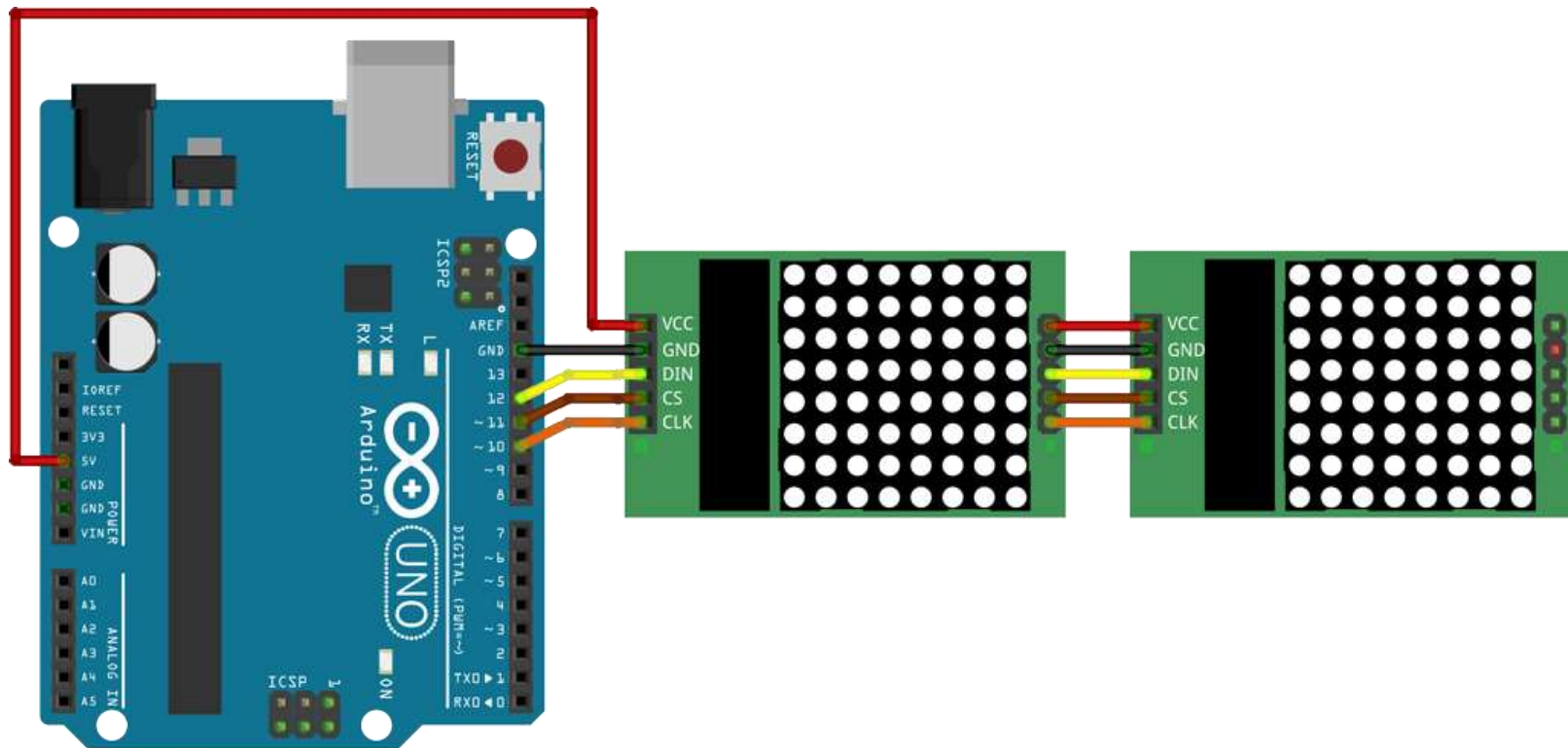
dot matrix에 ♥ 를 그리는 코드와

♥ 사진을 제출하시오.

완성된 코드를 `ARnn_heart.ino` & `ARnn_heart.png`  
로 저장해서 제출.







fritzing

<https://www.brainy-bits.com/how-to-control-max7219-led-matrix/>





# [Practice]

## ◆ [wk08]

- **Arduino LED – IV : DM & DM module**
- **Complete your project**
- **Submit folder : ARnn\_Rpt07**

# wk08 : Practice-07 : ARnn\_Rpt07

## ◆ [Target of this week]

- Complete your works
- Save your outcomes
- Upload all in github.

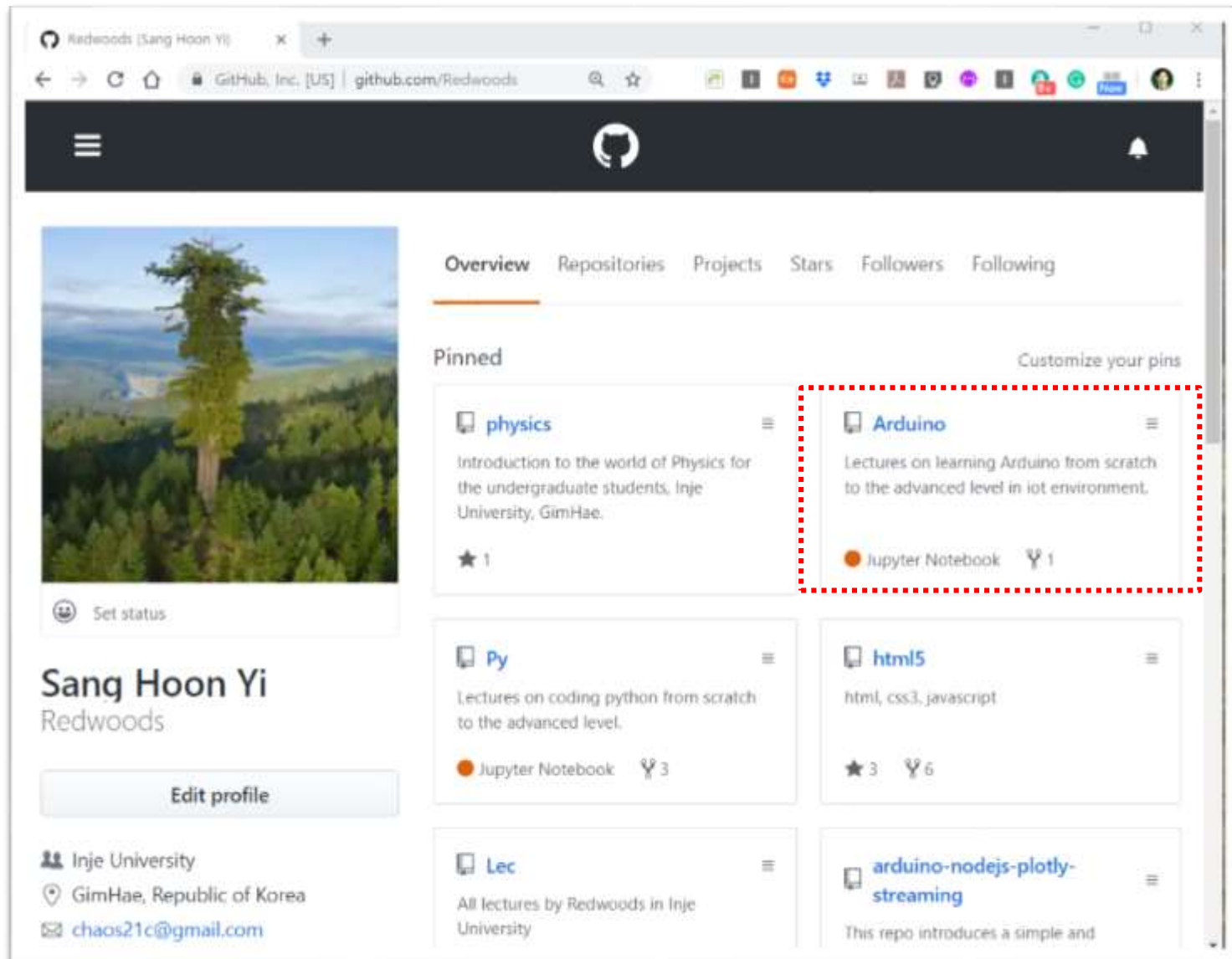
제출폴더명 : **ARnn\_Rpt07**

### 제출할 파일들

- ① ARnn\_dm.png
- ② ARnn\_dm\_module.fzz
- ③ Arnn\_dm\_project.ino
- ④ **ARnn\_heart.ino**
- ⑤ **ARnn\_heart.png**

## ● References & good sites

- ✓ <http://www.nodejs.org/ko> Node.js
- ✓ <http://www.arduino.cc> Arduino Homepage
- ✓ <http://www.w3schools.com> By w3schools
- ✓ <http://www.github.com> GitHub
- ✓ <http://www.google.com> Googling



Redwoods (Sang Hoon Yi)

GitHub, Inc. [US] | github.com/Redwoods

Overview Repositories Projects Stars Followers Following

Pinned Customize your pins

**physics**  
Introduction to the world of Physics for the undergraduate students, Inje University, GimHae.  
★ 1

**Arduino**  
Lectures on learning Arduino from scratch to the advanced level in iot environment.  
Jupyter Notebook 🍴 1

**Py**  
Lectures on coding python from scratch to the advanced level.  
Jupyter Notebook 🍴 3

**html5**  
html, css3, javascript  
★ 3 🍴 6

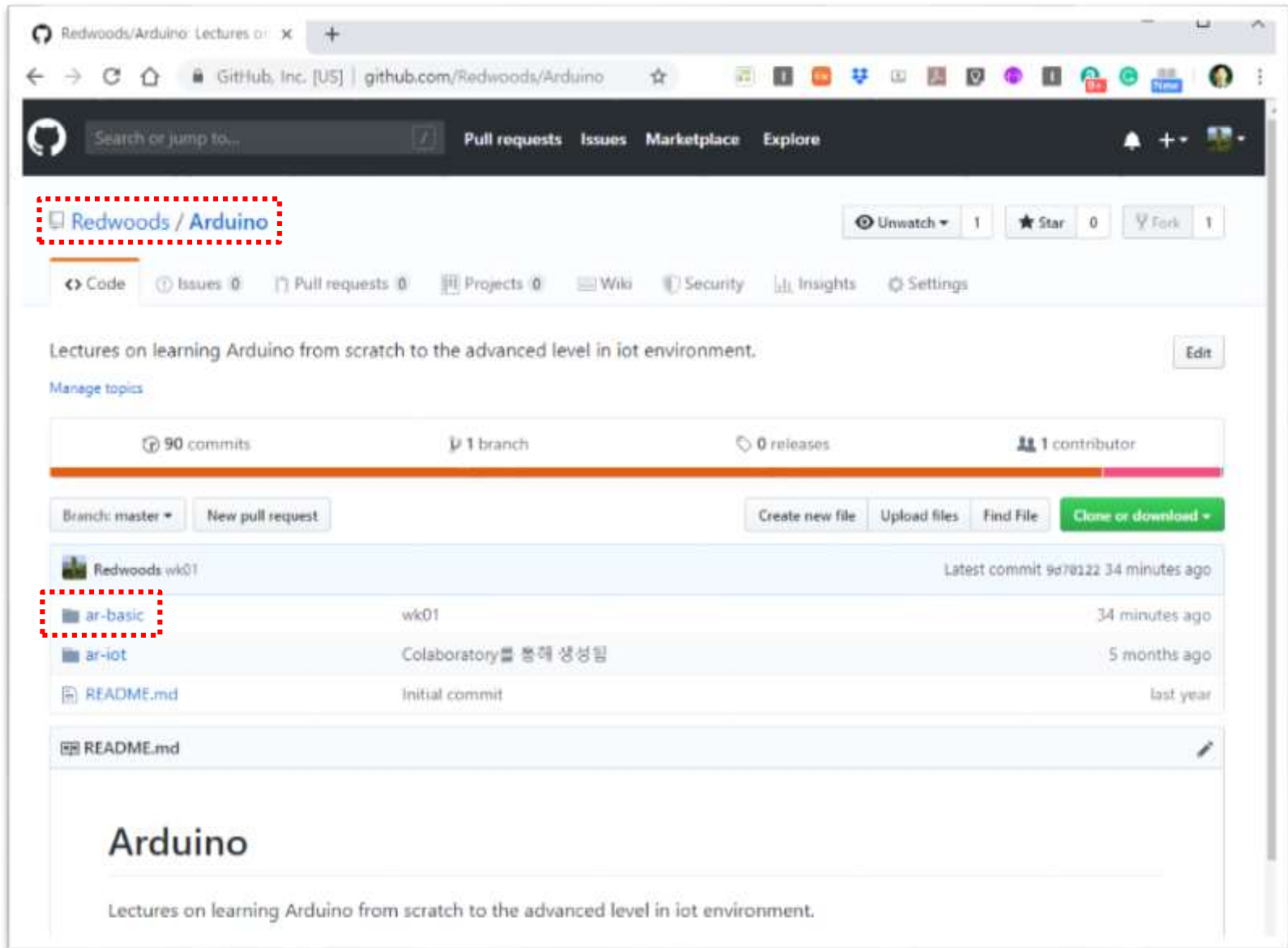
**Lec**  
All lectures by Redwoods in Inje University

**arduino-nodejs-plotly-streaming**  
This repo introduces a simple and

**Sang Hoon Yi**  
Redwoods

Edit profile

Inje University  
GimHae, Republic of Korea  
chaos21c@gmail.com



The screenshot shows the GitHub repository page for **Redwoods/Arduino**. The repository description is "Lectures on learning Arduino from scratch to the advanced level in iot environment." The repository has 90 commits, 1 branch, 0 releases, and 1 contributor. The file list includes **ar-basic** (34 minutes ago), **ar-iot** (5 months ago), and **README.md** (last year). The **ar-basic** file is highlighted with a red dashed box. The repository is currently on the **master** branch.

Redwoods/Arduino: Lectures on learning Arduino from scratch to the advanced level in iot environment.

Manage topics

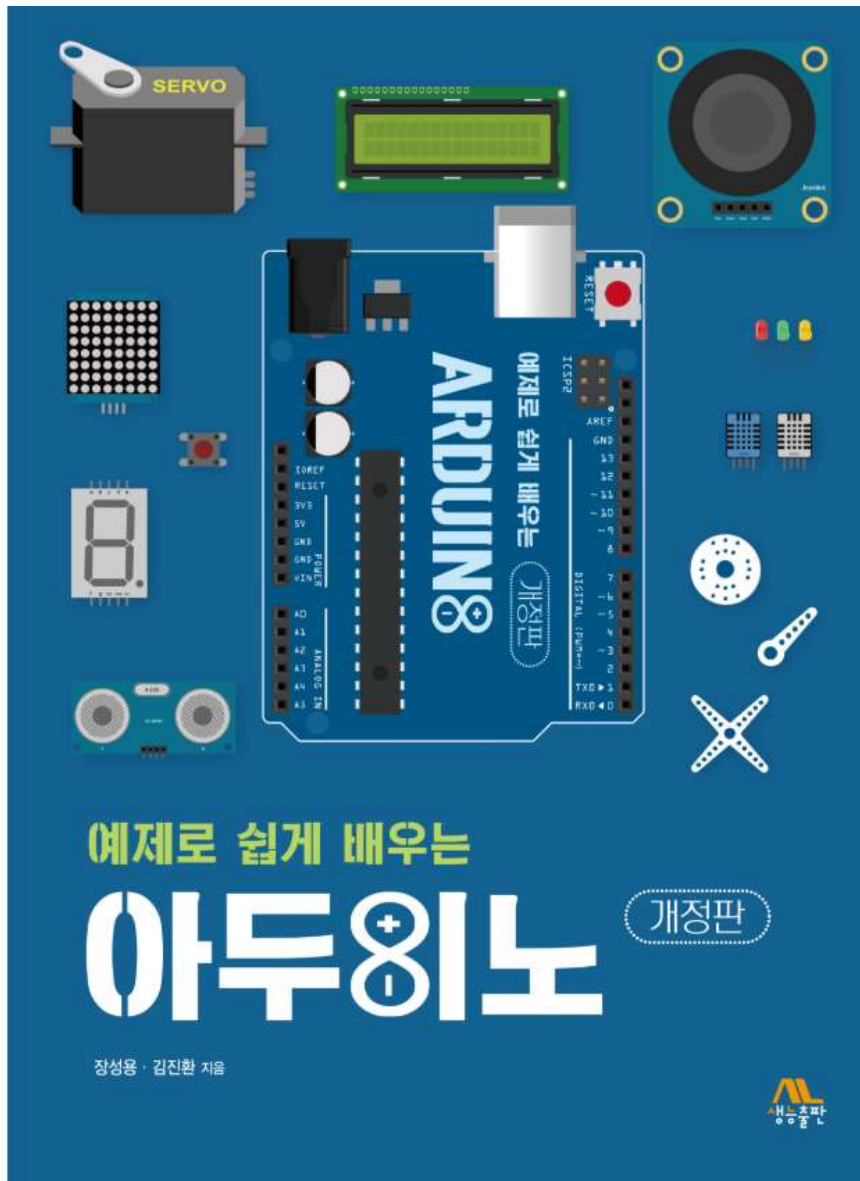
90 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find File Clone or download

File	Commit	Time
Redwoods (wk01)	Latest commit 9d70122	34 minutes ago
ar-basic	wk01	34 minutes ago
ar-iot	Colaboratory를 통해 생성됨	5 months ago
README.md	Initial commit	last year

## Arduino

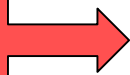
Lectures on learning Arduino from scratch to the advanced level in iot environment.



# 아두이노 키트(Kit)



web



<https://www.devicemart.co.kr/goods/view?no=12170416>

## 아두이노 레벨업키트(골드) 구성품

						
아두이노 UNO	USB 케이블	830핀 브레드보드	미니 브레드보드	점퍼와이어 세트	듀폰케이블 M/F	듀폰케이블 M/M
						
저항 220Ω	저항 1KΩ	저항 10KΩ	가변저항 10KΩ	빨강 LED	녹색 LED	파랑 LED
						
노랑 LED	RGB LED (CA)	RGB LED 모듈	1digit FND (CA)	4digit FND (CA)	8x8 도트 매트릭스	택트 스위치
						
택트 스위치 캡	눌 스위치	리드 스위치 센서	4x4키 매트릭스	5V 릴레이 모듈	조이스틱 모듈	수위 센서
						
온도센서 LM35	터미스터	온습도센서	CDS 조도센서	불꽃감지센서	적외선 수신기	IR 리모콘



# 아두이노 키트(Kit) : Part-2

						
TCRT5000 적외선 센서	인체감지센서 모듈	사운드센서	능동부저	수동부저	초음파센서	I2C 1602 LCD 모듈
						
서보모터	스텝모터	스텝모터드라이버	RFID 수신 모듈	RFID 카드	RFID 태그	DS1302 RTC 모듈
						
1N4001 다이오드	2N2222 트랜지스터	74HC595	1x40 핀헤더	9V 배터리 스냅	아크릴 고정판	

■ 아두이노 UNO × 1	■ USB 케이블 × 1	■ 830핀브레드보드 × 1	■ 미니 브레드보드 × 1	■ 점퍼와이어세트 × 1
■ 듀폰케이블 × 80 (M/F,M/M)	■ 저항 × 30	■ 가변저항 × 1	■ LED × 20	■ RGB LED × 1
■ RGB LED 모듈 × 1	■ 1digit FND(CA) × 1	■ 4digit FND(CA) × 1	■ 8×8도트 매트릭스 × 1	■ 탭스위치 × 5
■ 탭스위치 캡 × 5	■ 볼스위치 × 1	■ 리드 스위치 센서 × 1	■ 4×4 키 매트릭스 × 1	■ 5V 릴레이 모듈 × 1
■ 조이스틱 모듈 × 1	■ 수위 센서 × 1	■ 온도센서 LM35 × 1	■ 써미스터 × 1	■ 온습도센서 × 1
■ CdS 조도센서 × 1	■ 불꽃감지센서 × 1	■ 적외선 수신기 × 1	■ IR 리모컨 × 1	■ TCRT5000 적외선 센서 × 1
■ 인체감지센서 모듈 × 1	■ 사운드센서 × 1	■ 능동부저 × 1	■ 수동부저 × 1	■ 초음파센서 × 1
■ I2C 1602 LCD 모듈 × 1	■ 서보모터 × 1	■ 스텝모터 × 1	■ 스텝모터드라이버 × 1	■ RFID 수신 모듈 × 1
■ RFID 카드 × 1	■ RFID 태그 × 1	■ DS1302 RTC 모듈 × 1	■ 1N4001 다이오드 × 1	■ 2N2222 트랜지스터 × 1
■ 74HC595 × 1	■ 1x40 핀헤더 × 1	■ 9V 배터리 스냅 × 1	■ 아크릴 고정판 × 1	