

## [마이크로프로세서및실습] 과제4 Dot Matrix 2020136129 최수연

### 1. 과제 제목

- Dot Matrix

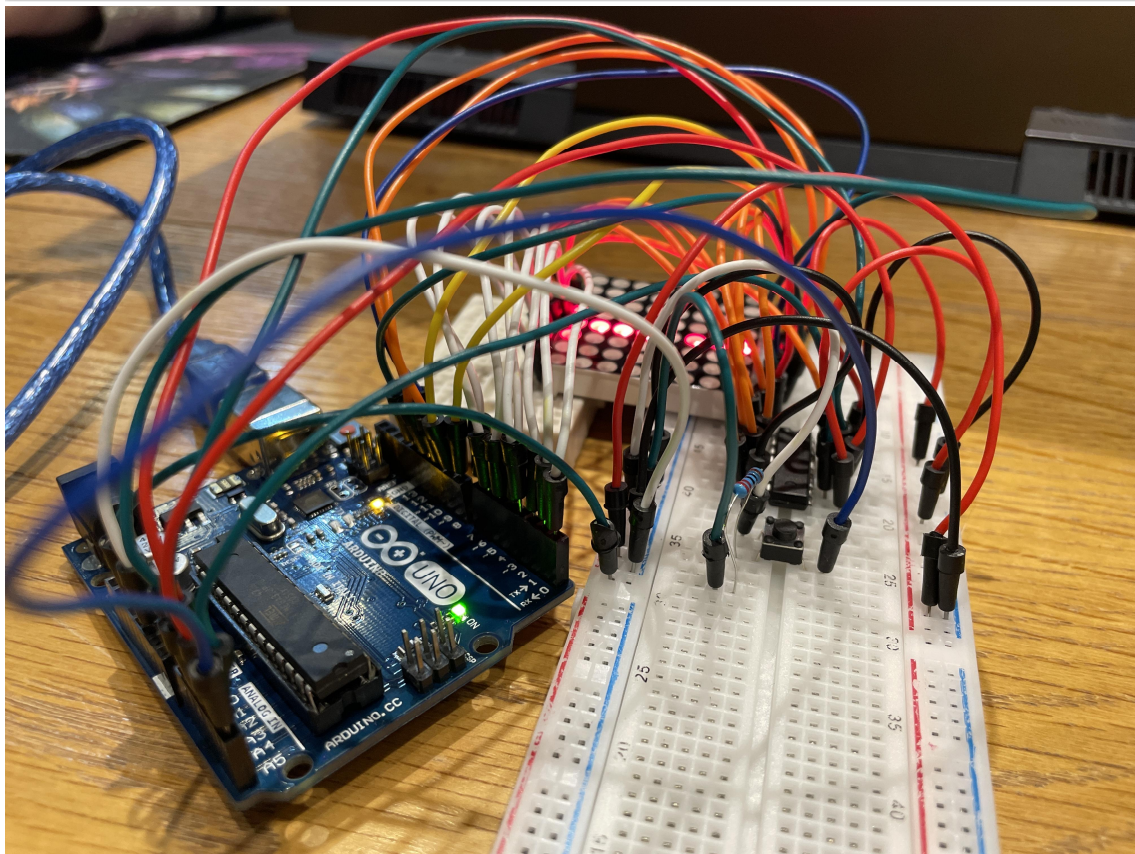
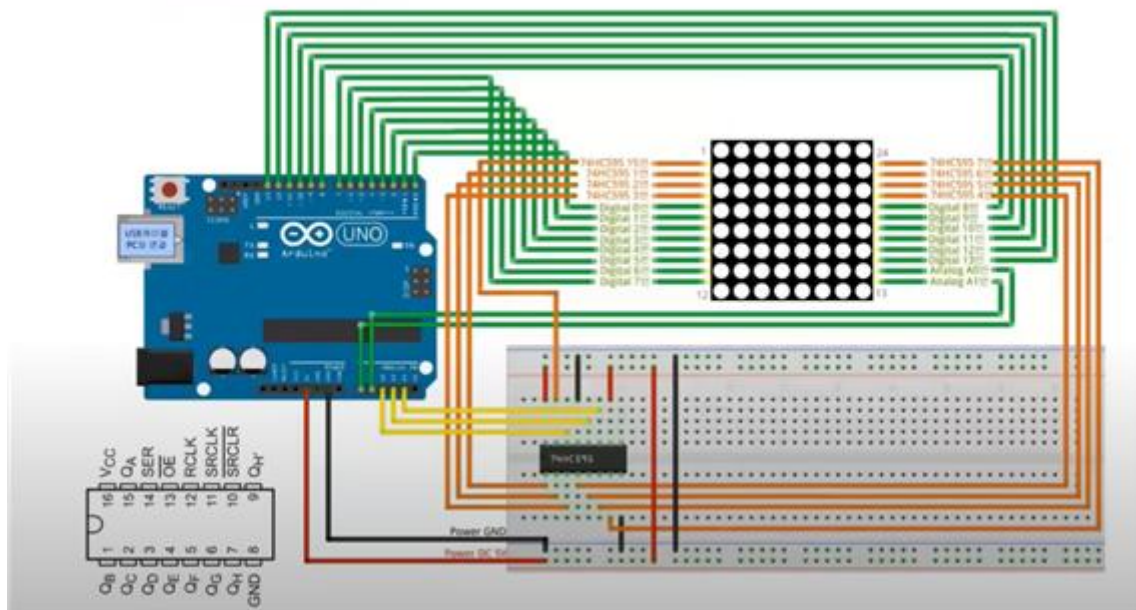
### 2. 부품 리스트

- 아두이노 우노 R3
- 830 포인트 브레드보드
- 숫자 점퍼 케이블
- 아두이노 USB 케이블
- 탭트 스위치
- Dot Matrix
- DC 모터 드라이버 IC

### 3. 구현하고자 하는 기능 및 설명

- Dot Matrix, 74HC595, Switch를 사용하여 아두이노에 연결한다.
- 다음 조건을 만족하도록 프로그래밍하여 dot matrix를 동작시킨다.
  - Switch를 누르면 다음 내용을 차례로 화면에 표시한다.
    - 학번을 차례로 하나씩 표시한다.
    - 이모티콘 3개를 차례로 표시한다.
  - 표시간격은 1초이내로 한다.
  - 표시방법은 자유로 한다.
    - 예를 들면, 스크롤 업/다운, shift left/right, fade in/out, etc..

#### 4. 부품과 아두이노 우노 보드와의 연결도



## 5. 프로그램 소스

```
#include <LedControl.h>
#include <FrequencyTimer2.h>
/* 타이머/카운터 인터럽트 사용을 위해 FrequencyTimer2.h 헤더 파일을 include 한다. */

// 표현할 문자를 2진수로 정의 새로운 문자 추가 가능.
/* 각 핀(Pin) 정의, 프로그램 가독성을 위하여 배열로 선언 */
int pins[17] = { -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15};
// 가로 열 핀(Pin) 정의(col[xx] of leds = pin yy on led matrix), ex) pins[1]은 디지털0
// 번핀
// 세로 핀은 74HC595 IC를 이용하여 제어하므로 제외 됨.
int GreenCols[8] = {pins[1], pins[2], pins[3], pins[4], pins[5], pins[6], pins[7], pins[8]};
int RedCols[8] = {pins[9], pins[10], pins[11], pins[12], pins[13], pins[14], pins[15],
pins[16]};
const int DS = 16;
const int ST_CP = 17;
const int SH_CP = 18;
int count = 0; //제어
int Button = 19;
boolean sw_stop = false;
/* 표현할 문자 패턴, numPatterns 의 개수에 맞게 표시
   Define부에서 각 문자 별로 정의 됨 */

byte num_string[13][8] = {
    B00011100, B00100010, B00100000, B00100000, B00010000, B00001000, B00000100,
    B00111110, // 2
    B00111000, B01000100, B01000100, B01000100, B01000100, B01000100, B01000100,
    B00111000, // 0
    B00011100, B00100010, B00100000, B00100000, B00010000, B00001000, B00000100,
    B00111110, // 2
    B00111000, B01000100, B01000100, B01000100, B01000100, B01000100, B01000100,
    B00111000, // 0
    B00010000, B00011000, B00010000, B00010000, B00010000, B00010000, B00010000,
    B00111000, // 1
    B00011100, B00100010, B00100000, B00011000, B00100000, B00100000, B00100010,
    B00011100, // 3
    B00011100, B00100010, B00000010, B00011110, B00100010, B00100010, B00100010,
    B00011100, // 6
    B00010000, B00011000, B00010000, B00010000, B00010000, B00010000, B00010000,
    B00111000, // 1
    B00011100, B00100010, B00100000, B00100000, B00010000, B00001000, B00000100,
```

```

B00111110, // 2
    B00111000, B01000100, B01000100, B01000100, B01111000, B01000000, B01000100,
B00111000, // 9
    B00111100, B01000010, B10100101, B10000001, B10100101, B10011001, B01000010,
B00111100, //이모티콘1
    B00111100, B01000010, B10100101, B10000001, B10111101, B10000001, B01000010,
B00111100, //이모티콘2
    B00111100, B01000010, B10100101, B10000001, B10011001, B10100101, B01000010,
B00111100 //이모티콘3
};
unsigned int NumCnt = 0;
#define Row_Sig_AllOff 8
void shiftRegister(unsigned char data) // 74HC595 시프트 레지스터 IC 구동 함수
{
    int i = 0;
    digitalWrite(ST_CP, LOW); // 래치 OFF
    for (i = 0; i < 8; i++)
    {
        digitalWrite(SH_CP, LOW); // 플립플롭 클럭 핀(Pin) 로우(LOW, DC 0V)
        if (data & (0x80 >> i))
        {
            digitalWrite(DS, HIGH); // 시리얼 데이터 핀(Pin) 하이(HIGH, DC 5V)이면
        }
        else
        {
            digitalWrite(DS, LOW); // 시리얼 데이터 핀(Pin) 로우(LOW, DC 0V)이면
        }
        digitalWrite(SH_CP, HIGH); // 플립플롭 클럭 핀(Pin) 하이(HIGH, DC 5V)
    }
    digitalWrite(ST_CP, HIGH); // 래치 ON
}
void RowPulseControl(unsigned char data) // 세로 행(Row) 핀(Pin) 출력 제어 함수
{
    /* 공통애노드 타입 8x8 도트매트릭스 LED 모듈로 변경 됨
    인터럽트 서비스 루틴을 나가기 전에 세로 행(Row) 핀(Pin)에 해당 핀(Pin)에만 하이
    (HIGH, DC 5V)를
    출력하고 그 외 핀(Pin)들은 로우(LOW, DC 0V)를 인가하도록 한다 */
    switch (data)
    {
        case 0:
            shiftRegister(0x80); break; // 첫 번째 ROW
    }
}

```

```

case 1:
    shiftRegister(0x40); break; // 두 번째 ROW
case 2:
    shiftRegister(0x20); break; // 세 번째 ROW
case 3:
    shiftRegister(0x10); break; // 네 번째 ROW
case 4:
    shiftRegister(0x08); break; // 다섯 번째 ROW
case 5:
    shiftRegister(0x04); break; // 여섯 번째 ROW
case 6:
    shiftRegister(0x02); break; // 일곱 번째 ROW
case 7:
    shiftRegister(0x01); break; // 여덟 번째 ROW
case Row_Sig_AllOff:
    shiftRegister(0x00); break; // 전 ROW 핀(Pin) 신호 OFF
}
}

void DotmatrixDisplay() // 8x8 도트매트릭스 모듈 문자 복사 함수
{
    /* 1바이트에 해당하는 0~7번비트, 즉 각행에 비트 값이 0인지 1인지를 판단하여
    0~7 포트 핀에 해당 신호를 출력 한다. 이때 각행이 1이면 로우(LOW, DC 0V) 신호,
    0 이면 하이(HIGH, DC 5V) 신호를 내 보내어 LED를 켜거나 끈다.
    각행에 비트검사가 이루어지면서 각행 핀에 신호 출력 후 공통신호인 해당 열에
    하이(HIGH, DC 5V) 신호를 출력 한다.
    1열부터 8열까지 위와 같이 동작 하도록 된다.
    8열까지 신호를 출력한 후, 문자가 표현되며, 다음문자는 NumCnt 값이 타이머에 변경
    된 후
    나타나도록 설계 되어 있다. */
    for (int i = 0; i < 8; i++)
    {
        for (int j = 0; j < 8; j++) // 1Byte에서 1bit별로 체크하여 핀에 하이 혹은 로우 신호
        를 출력
        {
            if ((num_string[NumCnt][i] & 1 << j)) // 0번비트부터 7번비트까지 비트 체크
            {
                //Active Low, Because Common Anode Dot Matrix Led 8*8, LED ON
                digitalWrite(RedCols[j], LOW);
            }
            else digitalWrite(RedCols[j], HIGH); // LED OFF
        }
    }
}

```

// 각 열에 공통애노드 신호인 하이(HIGH, DC 5V) 신호 74HC595 IC를 이용하여 출력한다.

// 해당하지 않는 핀은 로우(LOW, DC 0V)를 출력

RowPulseControl(i);

delay(1); // LED 휘도 보장을 위함

RowPulseControl(Row\_Sig\_AllOff); //출력을 보낸 후 LED 잔상을 제거 하기 위함

}

}

void clearLeds() // 8x8 도트매트릭스 모듈 메모리 초기화 함수

{

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++) //1Byte에서 1bit별로 체크하여 핀에 하이 혹은 로우 신호를 출력

{

digitalWrite(GreenCols[j], HIGH);

RowPulseControl(i);

}

}

for (int i = 0; i < 8; i++)

{

for (int j = 0; j < 8; j++) //1Byte에서 1bit별로 체크하여 핀에 하이 혹은 로우 신호를 출력

{

digitalWrite(RedCols[j], HIGH);

RowPulseControl(i);

}

}

}

void Timer() // 1ms마다 Timer()함수가 호출(Generate Interrupt Service routine per 1ms)

{

static unsigned int cnt = 0; //정적 사용자 변수 초기화

cnt++; // 1ms 증가

if (cnt == 1000) //1초가 되면

{

cnt = 0;

NumCnt++; //문자 선택 카운터

if (NumCnt == 13)

{

NumCnt = 0; // 문자 선택 초기화

```

    }
}
}
void setup() // 초기화
{
    /* 8x8 도트매트릭스 모듈과 연결된 핀(Pin)들 출력 포트로 설정 */
    pinMode(0, OUTPUT); // 도트매트릭스 LED 모듈 - 5번 핀, 초록 1열
    pinMode(1, OUTPUT); // 도트매트릭스 LED 모듈 - 6번 핀, 초록 2열
    pinMode(2, OUTPUT); // 도트매트릭스 LED 모듈 - 7번 핀, 초록 3열
    pinMode(3, OUTPUT); // 도트매트릭스 LED 모듈 - 8번 핀, 초록 4열
    pinMode(4, OUTPUT); // 도트매트릭스 LED 모듈 - 9번 핀, 초록 5열
    pinMode(5, OUTPUT); // 도트매트릭스 LED 모듈 - 10번 핀, 초록 6열
    pinMode(6, OUTPUT); // 도트매트릭스 LED 모듈 - 11번 핀, 초록 7열
    pinMode(7, OUTPUT); // 도트매트릭스 LED 모듈 - 12번 핀, 초록 8열
    pinMode(8, OUTPUT); // 도트매트릭스 LED 모듈 - 20번 핀, 빨강 1열
    pinMode(9, OUTPUT); // 도트매트릭스 LED 모듈 - 19번 핀, 빨강 2열
    pinMode(10, OUTPUT); // 도트매트릭스 LED 모듈 - 18번 핀, 빨강 3열
    pinMode(11, OUTPUT); // 도트매트릭스 LED 모듈 - 17번 핀, 빨강 4열
    pinMode(12, OUTPUT); // 도트매트릭스 LED 모듈 - 16번 핀, 빨강 5열
    pinMode(13, OUTPUT); // 도트매트릭스 LED 모듈 - 15번 핀, 빨강 6열
    pinMode(14, OUTPUT); // 도트매트릭스 LED 모듈 - 14번 핀, 빨강 7열
    pinMode(15, OUTPUT); // 도트매트릭스 LED 모듈 - 13번 핀, 빨강 8열
    pinMode(SH_CP, OUTPUT); // 래치 클럭 핀(Pin) 출력 포트로 설정
    pinMode(ST_CP, OUTPUT); // 플립플롭 클럭 핀(Pin) 출력 포트로 설정
    pinMode(DS, OUTPUT); // 시리얼 데이터 입력 핀(Pin) 출력 포트로 설정
    pinMode(Button, INPUT); //버튼을 인풋으로 설정
}
void loop() // 무한 루프
{
    if(digitalRead(Button)==HIGH) {
        sw_stop = true;
    }
    else {
        sw_stop = false;
        //FrequencyTimer2::setOnOverflow(Timer);
        //clearLeds();
    }
    if(sw_stop == true){
        // 타이머/카운터 인터럽트 주기 설정 1000us = 1ms
        FrequencyTimer2::setPeriod(1000);
        // 인터럽트 서비스 루틴 함수 선언
    }
}

```

```
    FrequencyTimer2::setOnOverflow(Timer);  
    // 8x8 도트매트릭스 모듈 사용 메모리 초기화  
    clearLeds();  
    count=1;  
}  
if(count==1){  
    DotmatrixDisplay(); // 8x8 도트매트릭스 LED 모듈 출력 함수  
}  
}
```



## [마이크로프로세서및실습] 과제4 Character LCD 2020136129 최수연

### 1. 과제 제목

- Character LCD

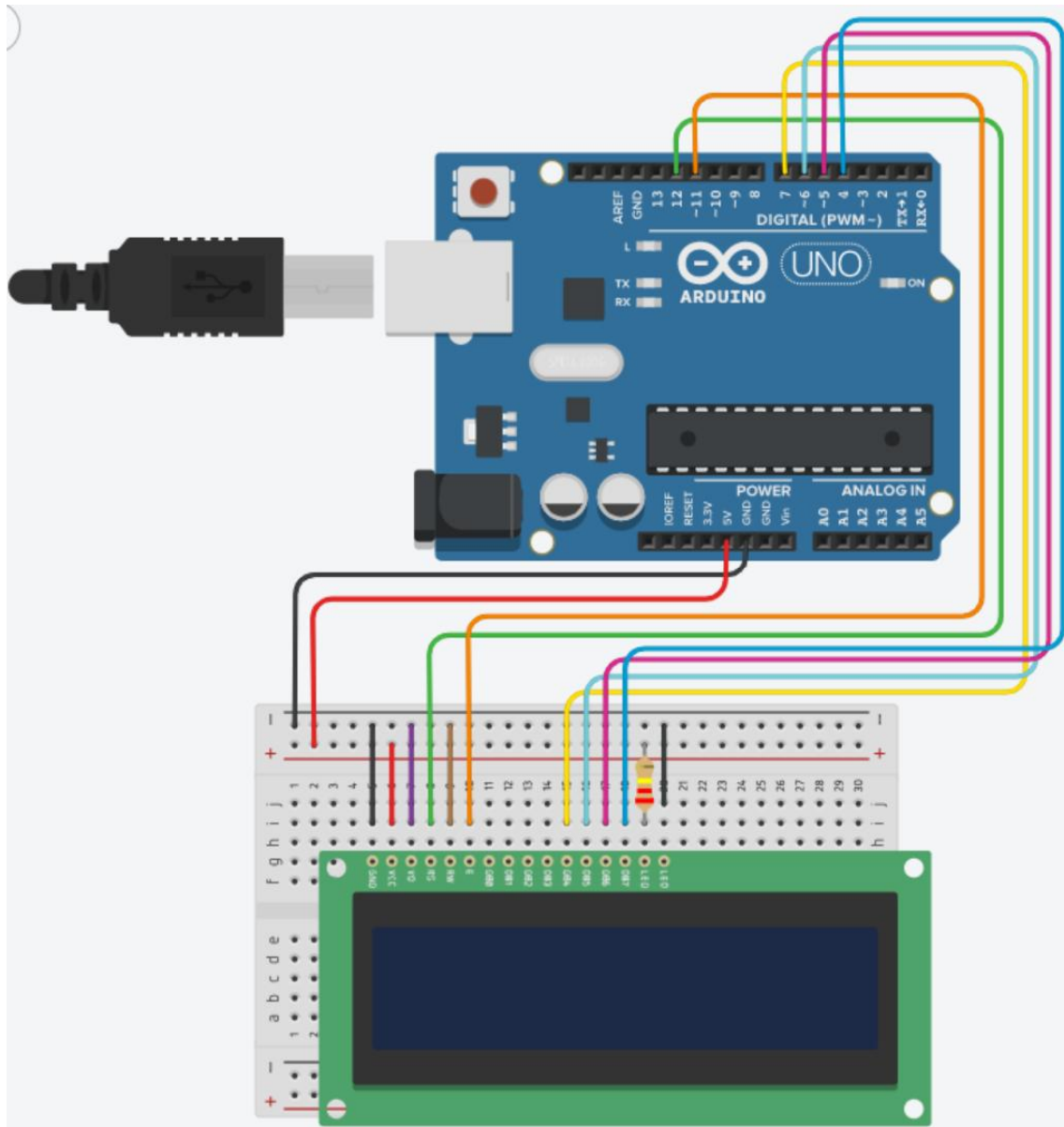
### 2. 부품 리스트

- 아두이노 우노 R3
- 830 포인트 브레드보드
- 슛스텀 점퍼 케이블
- 아두이노 USB 케이블
- 탭트 스위치
- Character LCD

### 3. 구현하고자 하는 기능 및 설명

- Character LCD, Switch를 아두이노와 연결한다.
- 다음 내용을 형식에 맞춰 LCD에 표시한다.
  - ① Switch를 첫번째로 누르면 “학번과 영문이름”을 LCD의 좌상단에서 부터 2줄로 연속적으로 표시한다. 2라인을 초과한 경우, 표시가능한 부분까지만 표시한다.
  - ② Switch를 2번째로 누르면 학번과 영문이름을 shift left 한다. 좌 상단에서 부터 한 글자씩 사라지고, 우 하단에는 표시되지 않은 내용이나, ‘학번과 영문이름’을 다시 표시하기 시작한다.
  - ③ Switch를 3번째로 누르면, shift left 동작이 정지한다.
  - ④ Switch를 4번째로 누르면, 정지된 상태가 해제되어 화면표시를 계속한다.

#### 4. 부품과 아두이노 우노 보드와의 연결도



## 5. 프로그램 소스

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(7,8,9,10,11,12);
```

```
int button = 13;
```

```
int btn_now = 0, btn_last=0;
```

```
int count = 0;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    lcd.begin(16, 2);
```

```
    lcd.clear();
```

```
    pinMode(button, INPUT);
```

```
}
```

```
void loop() {
```

```
    if(digitalRead(button)==LOW){
```

```
        count++;
```

```
        if(count == 4) count = 2;
```

```
    }
```

```
    if(count == 1){
```

```
        lcd.clear();
```

```
        Serial.println(digitalRead(button));
```

```
        lcd.setCursor(0,0);
```

```
        lcd.print("2020136129      2020136129      ");
```

```
        lcd.setCursor(0,1);
```

```
        lcd.print("CHOI SOO YEON   CHOI SOO YEON   ");
```

```
        delay(500);
```

```
    }
```

```
    if(count == 2){
```

```
        lcd.scrollDisplayLeft();
```

```
        delay(500);
```

```
    }
```

```
    if(count == 3){
```

```
        delay(1000);
```

```
    }
```

```
    if(count == 4){
```

```
        lcd.scrollDisplayLeft();
```

```
    }
```

```
}
```