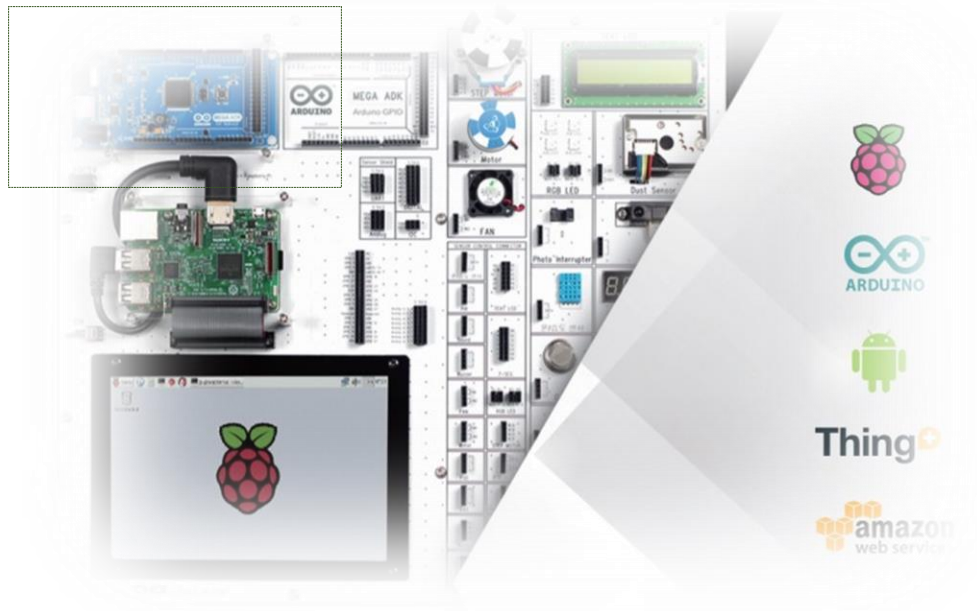


마이크로프로세서

(강의자료 #9)



교과목명 : 마이크로프로세서 (01)

담당교수 : 이 수 형

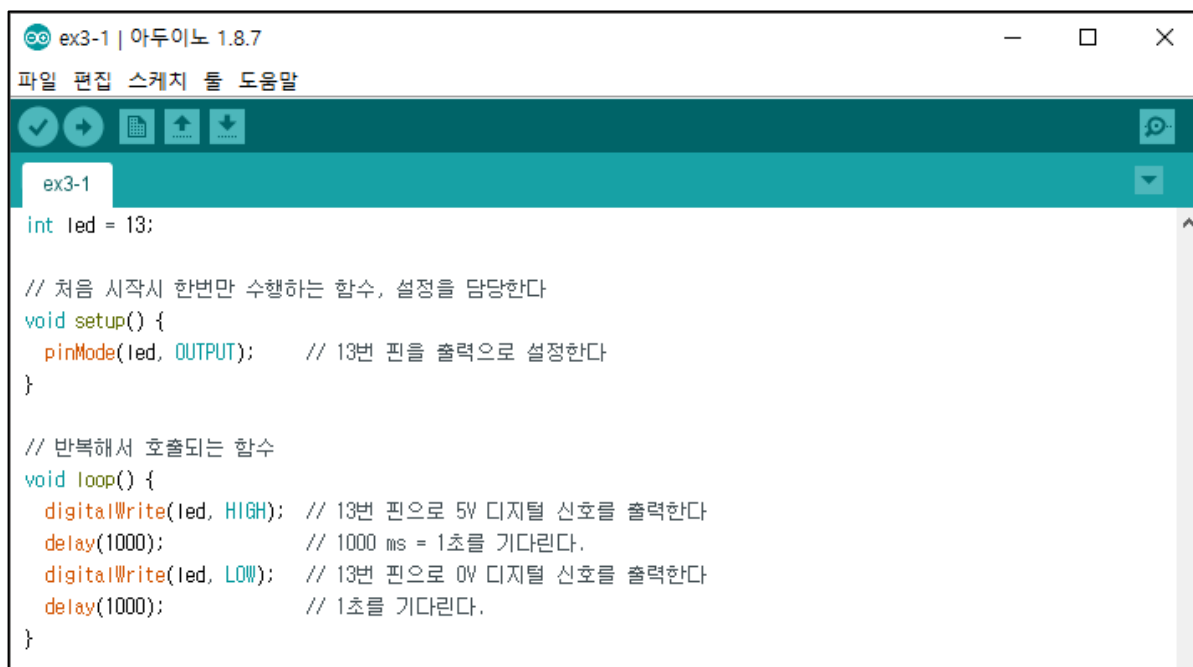
E-mail : soohyong@uu.ac.kr

교재명 : 스마트기기 개발을 위한 아두이노,
이수형. (LINC+ 사업단 배포)

수강생 안내

- 본 교과목은 기본적으로 “대면” 수업입니다.
- ‘코로나바이러스감염증-19’ 등의 이유로 대면수업을 받지 못하는 학생들을 위해 ‘비대면 실시간 수업’을 병행합니다. 기본적으로 대면 수업이므로 대면 수업에 참석한 학생들은 교실에서 출석을 체크하면 되며, 참석하지 못하고 ‘비대면 수업’을 듣는 학생들은 ‘실시간 온라인 강의’에 참석하면 출석이 반영됩니다.
- 수강생 여러분들은 강의 자료를 온라인 배포 및 게재 등의 행위를 할 시 저작권 문제가 발생할 수 있사오니 이에 유념하여 강의 자료를 공유하는 행위는 삼가 바랍니다.
- 교재는 “스마트기기 개발을 위한 아두이노”이며, 나누어준 키트는 각자 잘 관리하면서 실습실(대면) 및 집(비대면)에서 실험/실습을 수행할 수 있도록 진행합니다.

7. 모터 구동



The screenshot shows the Arduino IDE interface with a sketch named 'ex3-1'. The code is written in C++ and is designed to control a motor using a digital pin (pin 13). The code includes a setup function to initialize the pin and a loop function to toggle the pin state every 1000ms (1 second).

```
ex3-1 | 아두이노 1.8.7
파일 편집 스케치 툴 도움말

ex3-1
int led = 13;

// 처음 시작시 한번만 수행하는 함수, 설정을 담당한다
void setup() {
  pinMode(led, OUTPUT);    // 13번 핀을 출력으로 설정한다
}

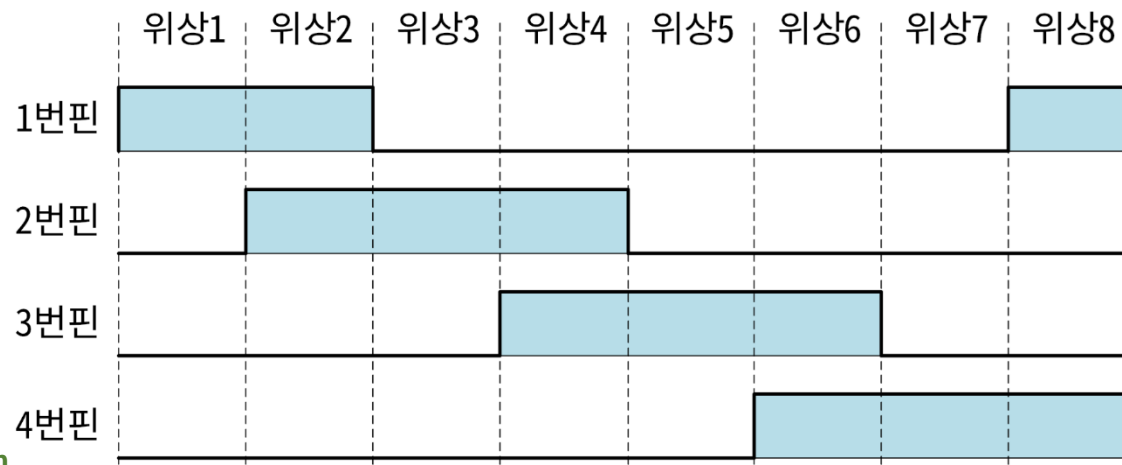
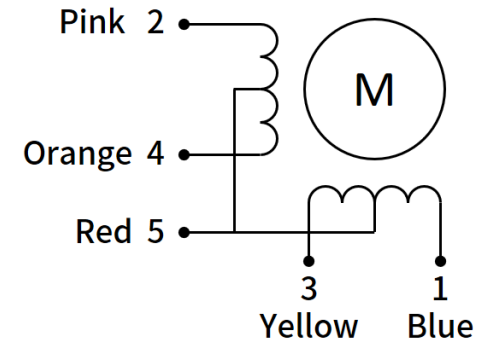
// 반복해서 호출되는 함수
void loop() {
  digitalWrite(led, HIGH); // 13번 핀으로 5V 디지털 신호를 출력한다
  delay(1000);             // 1000 ms = 1초를 기다린다.
  digitalWrite(led, LOW);  // 13번 핀으로 0V 디지털 신호를 출력한다
  delay(1000);             // 1초를 기다린다.
}
```

7.1 스텝모터

- 모터
 - 외부 기기들을 물리적으로 움직이게 하는 구동장치
 - DC 모터 : 직류전원사용 연속적이며 빠른 회전
 - 서보모터(servo motor) : PWM 신호를 이용하여 회전각을 제어, 정밀한 회전각
 - 스텝모터(step motor) : 정밀한 제어용 4개의 신호선을 통하여 펄스를 인가하여 제어
- 아두이노에서 사용
 - 모터는 많은 양의 전류 소요 : 추가 드라이버 모듈 사용

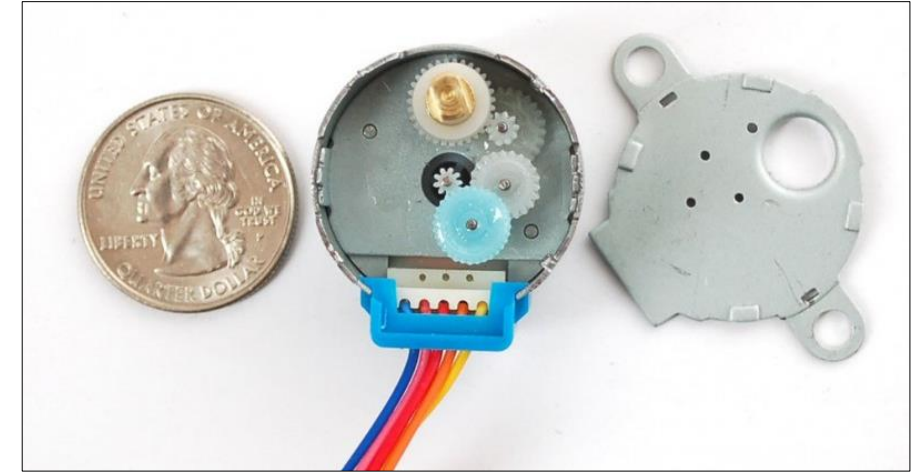
7.1 스텝모터

- 스텝모터의 구동
 - 4개의 펄스선을 통하여 펄스를 규칙적으로 인가
 - 8개의 위상 순서에 따라 CCW(counter clockwise), CW(clockwise) 회전



7.1 스텝모터

- 28BYJ-48 step motor spec.
 - Rated Voltage : 5V DC
 - Number of Phases : 4
 - Stride Angle
 - Half-step : $5.625^{\circ}/64 \Rightarrow 360 / 5.625 * 64 = 4096$
 - Full-step : $11.25^{\circ}/64 \Rightarrow 360 / 11.25 * 64 = 2048$
 - Frequency : 100Hz
 - Pull in torque : 300 gf.cm
 - Insulated Power : 600VAC/1mA/1s
 - Coil : Unipolar 5 lead coil



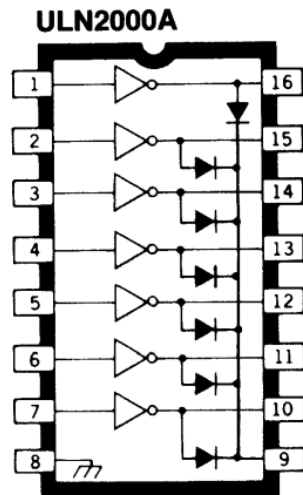
7.1 스텝모터

- 모터 드라이버

- 모터 구동에는 많은 양의 전류가 필요하기 때문에 아두이노의 입/출력 핀들을 바로 연결할 수 없음 $\Rightarrow$ 입력신호와 동일한 신호를 내 보내면서 대전류 사용 가능

- ULN2003

- 스텝 모터 제어용 드라이버, 5V ~ 12V



ULN20X3* (Each Driver)

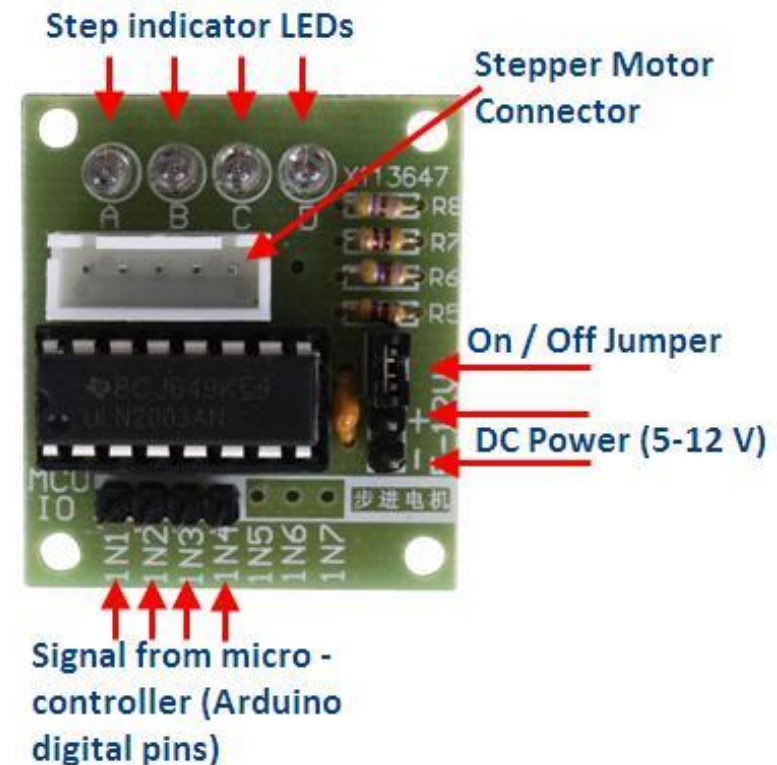
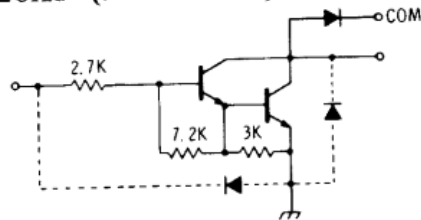
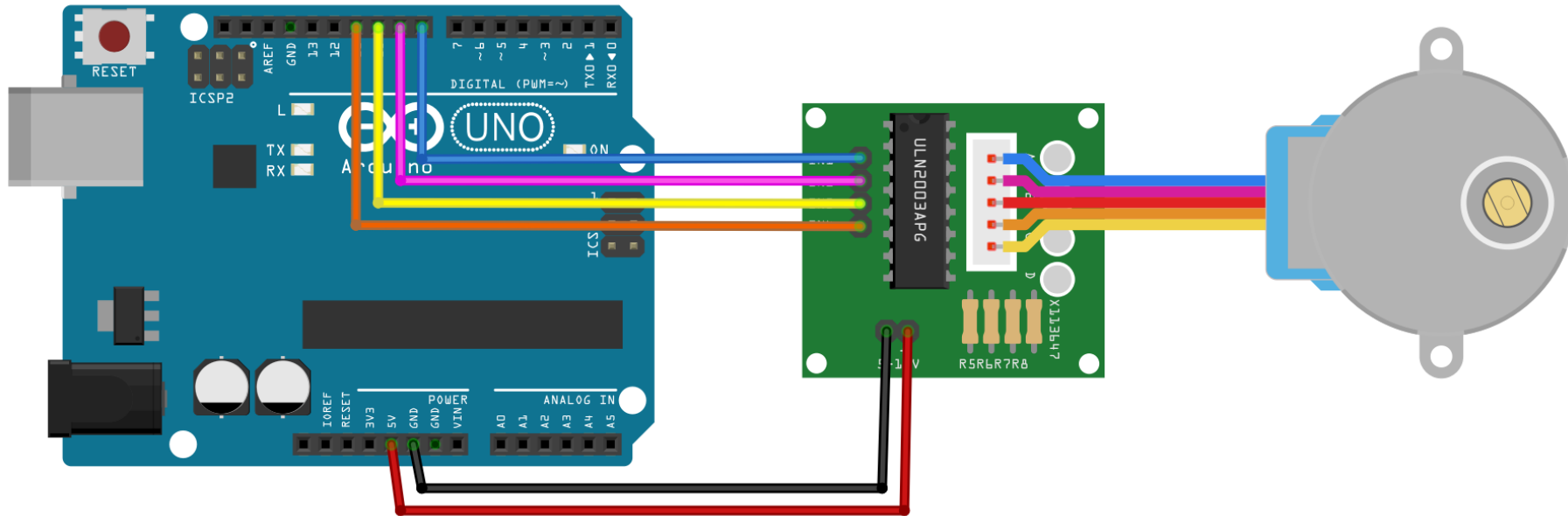


사진 출처 : http://mechasolution.com/shop/goods/goods_view.php?goodsno=538483&category=140015014

데이터시트 : <https://pdf1.alldatasheet.com/datasheet-pdf/view/125994/ETC1/ULN2003.html>

예제 7-1

- 회로구성



아두이노 우노 : 1개, 스텝모터 : 28BYJ-48, 5V DC 1개, 모터 드라이버 보드 : ULN2003 1개, 점퍼선 여러 개

예제 7-1: 스케치

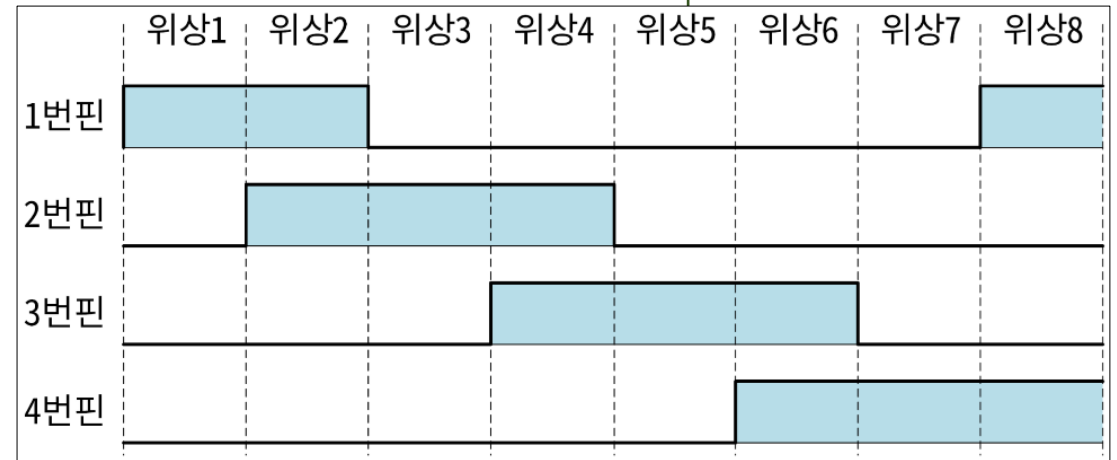
예제 7-1. 스텝모터 구동

// 스텝모터의 4개의 핀 정의

```
int pins[4] = { 8, 9, 10, 11 };
```

// 스텝모터의 위상(phase) 상태를 정의함, 8개의 상태 + 정지 상태

```
int signal[9][4] = {  
  { 1, 0, 0, 0 }, // 위상 0  
  { 1, 1, 0, 0 }, // 위상 1  
  { 0, 1, 0, 0 }, // 위상 2  
  { 0, 1, 1, 0 }, // 위상 3  
  { 0, 0, 1, 0 }, // 위상 4  
  { 0, 0, 1, 1 }, // 위상 5  
  { 0, 0, 0, 1 }, // 위상 6  
  { 1, 0, 0, 1 }, // 위상 7  
  { 0, 0, 0, 0 }  // 모터 정지 신호  
};
```



예제 7-1: 스케치

```
void setup() {  
    pinMode(13, OUTPUT); // 아두이노 보드의 LED로 정회전/역회전 방향 표시  
    // 스텝모터 제어용 핀을 출력용으로 설정, 배열을 사용하여 한번에 설정함  
    for(int i=0; i<4; i++)  
        pinMode(pins[i], OUTPUT);  
}  
void loop() {  
    int i, j;  
    unsigned long ms; // 동작 시간 측정용 변수  
  
    // 1. 시계 반대방향 회전 : 위상을 0~7까지 순차적으로 설정  
    digitalWrite(13, HIGH);  
    // 현재 시각을 기록함, 단위 [ms]  
    ms = millis();  
    // 매 반복마다 새로운 시각을 읽어서 3초가 지나지 않은 경우 반복  
    while(millis() - ms < 3000) {  
        for(i=0; i<8; i++) { // i : 0, 1, 2, 3, 4, 5, 6, 7  
            // 4개의 출력핀을 지정된 펄스 패턴대로 출력  
            for(j=0; j<4; j++) {  
                digitalWrite(pins[j], signal[i][j]);  
            }  
            delayMicroseconds(1000); // 회전 속도를 조절할 수 있는 값 (1000~5000)  
        }  
    }  
    delay(100);  
}
```

예제 7-1: 스케치

```
// 2. 시계 방향 회전: 위상을 7~0까지 반대로 설정
digitalWrite(13, LOW);
// 현재 시각을 기록함, 단위 [ms]
ms = millis();
// 매 반복마다 새로운 시각을 읽어서 3초가 지나지 않은 경우 반복
while(millis() - ms < 3000) {
    for(i=7; i>=0; i--) { // i : 7, 6, 5, 4, 3, 2, 1, 0
        // 4개의 출력핀을 지정된 펄스 패턴대로 출력
        for(j=0; j<4; j++) {
            digitalWrite(pins[j], signal[i][j]);
        }
        delayMicroseconds(1000); // 회전 속도를 조절할 수 있는 값 (1000~5000)
    }
}
delay(100);
}
```

예제 7-1

- 추가 사용 함수

- millis() : 아두이노 시작 후 현재까지 시간을 밀리초 단위로 읽음

- unsigned long ms; $\Rightarrow$ 32 bit 부호 없는 정수형 변수 ($0 \sim 2^{32}-1 = 4,294,967,296$)

```
// 현재 시각을 기록함, 단위 [ms]
ms = millis();
// 매 반복마다 새로운 시각을 읽어서 3초가 지나지 않은 경우 반복
while(millis() - ms < 3000) {
    .....
}
```

- delayMicroseconds(us) : 마이크로초 단위로 시간지연 사용

참고 : unsigned int 연산

- 8 bit 정수형의 예
 - 0000 0000에서 1씩 증가
 - 최대값 : 1111 1111 ($2^8 - 1 = 255$)
 - 다음 수는 1 0000 0000 $\Rightarrow$ 0000 0000 (8 bit만 저장 가능)

0 0000 0000	0
0 0000 0001	1
...	...
0 1111 1110	254
0 1111 1111	255
1 0000 0000	0
1 0000 0001	1

- 8bit 정수형 연산

$$\begin{array}{r}
 1000 \ 1110 \quad 135 \\
 + \ 0010 \ 1110 \quad 55 \\
 \hline
 1011 \ 1110 \quad 190
 \end{array}$$

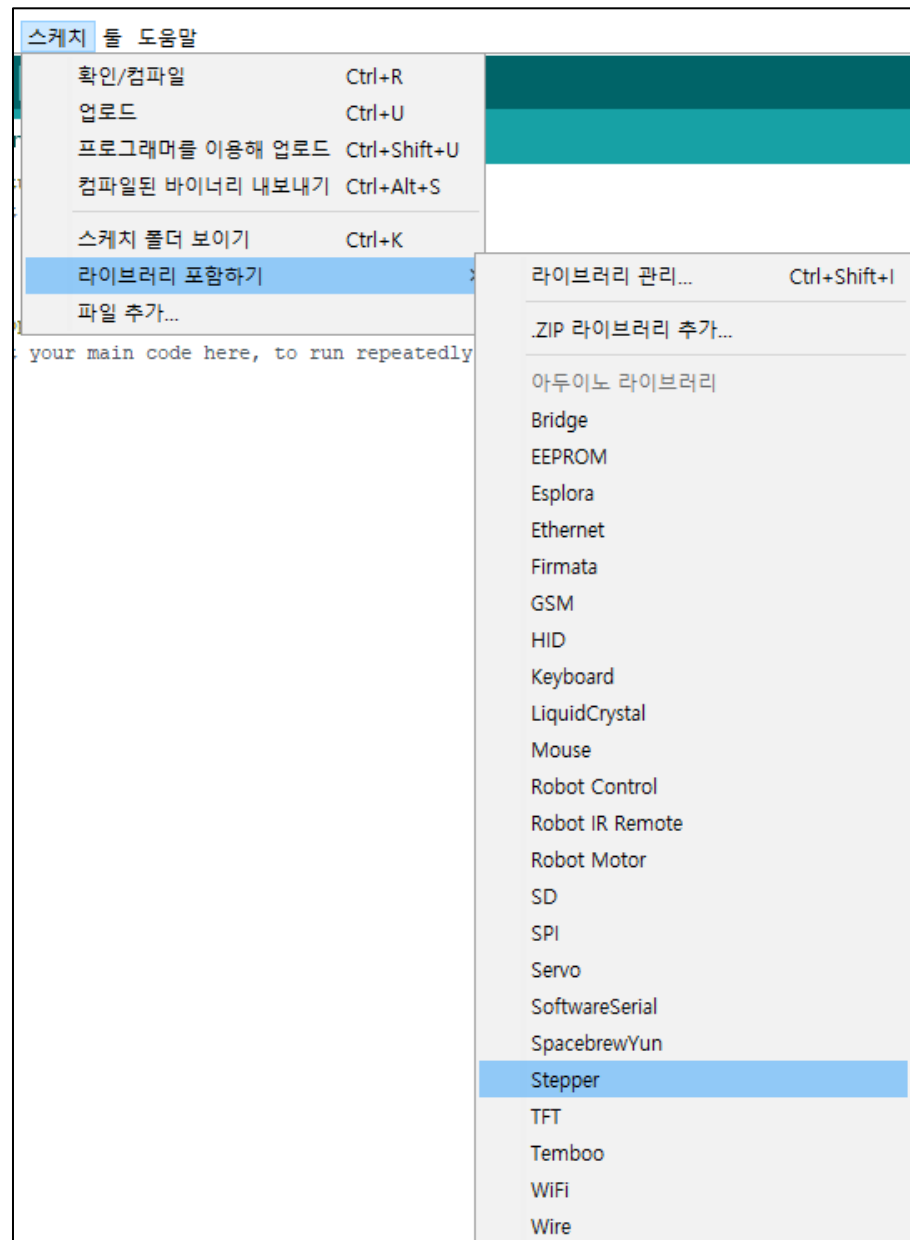
$$\begin{array}{r}
 1011 \ 1110 \quad 190 \\
 - \ 1000 \ 0111 \quad 135 \\
 \hline
 0010 \ 1110 \quad 55
 \end{array}$$

$$\begin{array}{r}
 1011 \ 1110 \quad 190 \\
 + \ 0111 \ 0000 \quad 112 \\
 \hline
 \cancel{1} \ 0010 \ 1110 \quad \cancel{302} \ 46
 \end{array}$$

$$\begin{array}{r}
 1 \ 0010 \ 1110 \quad 46 \\
 - \ 0111 \ 0000 \quad 112 \\
 \hline
 1011 \ 1110 \quad 190
 \end{array}$$

예제 7-2: Stepper 라이브러리

- 아두이노의 스텝모터 라이브러리
 - 복잡하게 위상 정보를 정의하고 순차적으로 출력하는 과정을 라이브러리로 간단하게 제공
 - 메뉴에서 라이브러리 포함 ⇒



예제 7-2: 스케치

```
#include <Stepper.h>

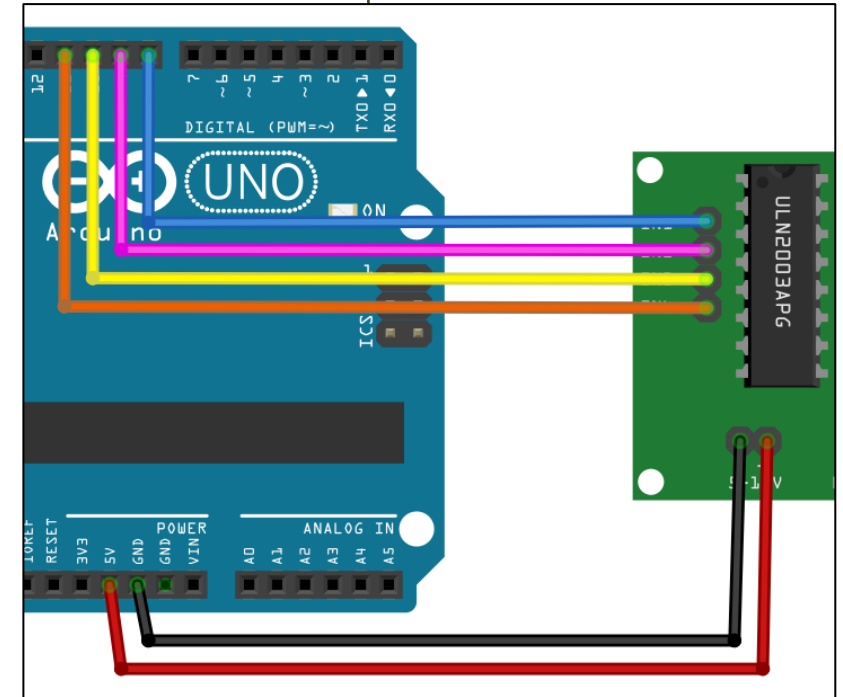
// 모터가 1회전(360도) 하는데 필요한 스텝의 수 지정
const int steps = 2048;

// 모터 드라이브에 연결된 핀 IN4, IN2, IN3, IN1
Stepper myStepper(steps, 11, 9, 10, 8);

void setup() {
    myStepper.setSpeed(14);
}

void loop() {
    // 시계 반대 방향으로 한바퀴 회전
    myStepper.step(steps);
    // 1초 대기
    delay(100);

    // 시계 방향으로 한바퀴 회전
    myStepper.step(-steps);
    // 1초 대기
    delay(100);
}
```

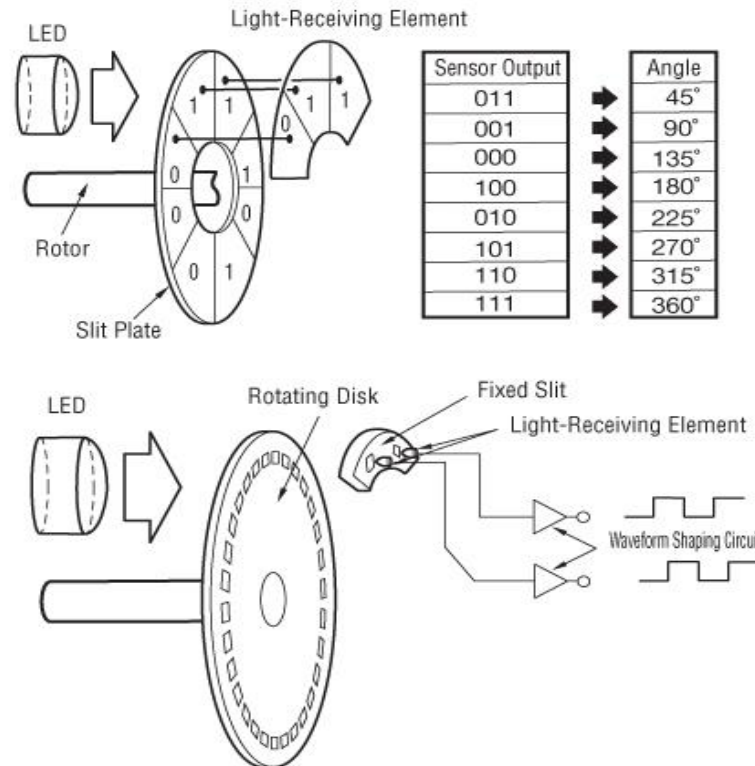


7.2 서보모터

- 서보모터 (servo motor)
 - 모터 내부에 기어 및 엔코더/가변저항을 이용한 피드백 포함
 - 정밀한 위치제어 가능, 강한 토크(회전력)가 필요한 곳에 사용
 - 로봇관절 제어, CNC 위치 제어, RC 자동차 또는 비행기 제어

- 엔코더

- 모터의 회전 각도 측정용
 - Absolute Encoder (절대각도)
 - Incremental Encoder (상대각도)



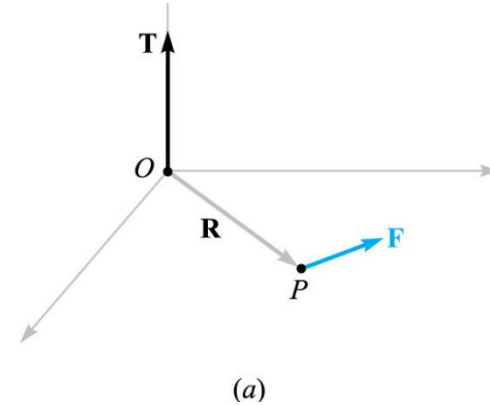
7.2 서보모터

- SG90 모델
 - 동작 전압 : DC 4.8 ~ 6.0V(5V 권장)
 - 회전 각도 : 0 ~ 180도(실제는 170도 정도)
 - 속도 : 0.1s/60도 (4.8V) / 0.12s/60도 (6V)
 - 토크 : 1.8kgf.cm (4.8V)
 - 크기 : 22.2mm x 11.8mm x 31mm
 - 무게 : 9g



- 토크 (회전력)

- $T = R \times F$ (ref: 전기자기학)
단위 $[N \cdot m]$



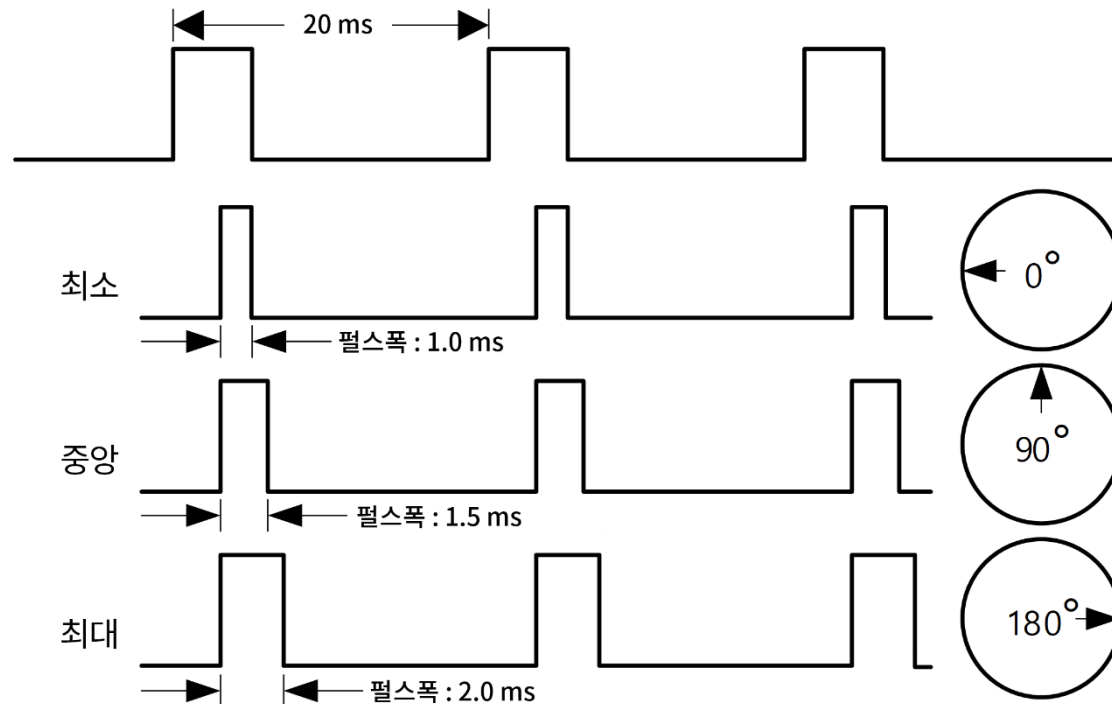
- $1 \text{ Kgf} \cdot \text{cm}$: 1cm의 막대 끝에 1Kg의 물체가 달려있을 때 막대의 끝에서 받는 토크

- 단위 변환

- $1 \text{ Kgf} \cdot \text{cm} = 1 \text{ Kgf} \times \frac{9.8 \text{ N}}{1 \text{ Kgf}} \cdot \text{cm} \times \frac{1 \text{ m}}{100 \text{ cm}} = 0.098 \text{ N} \cdot \text{m}$

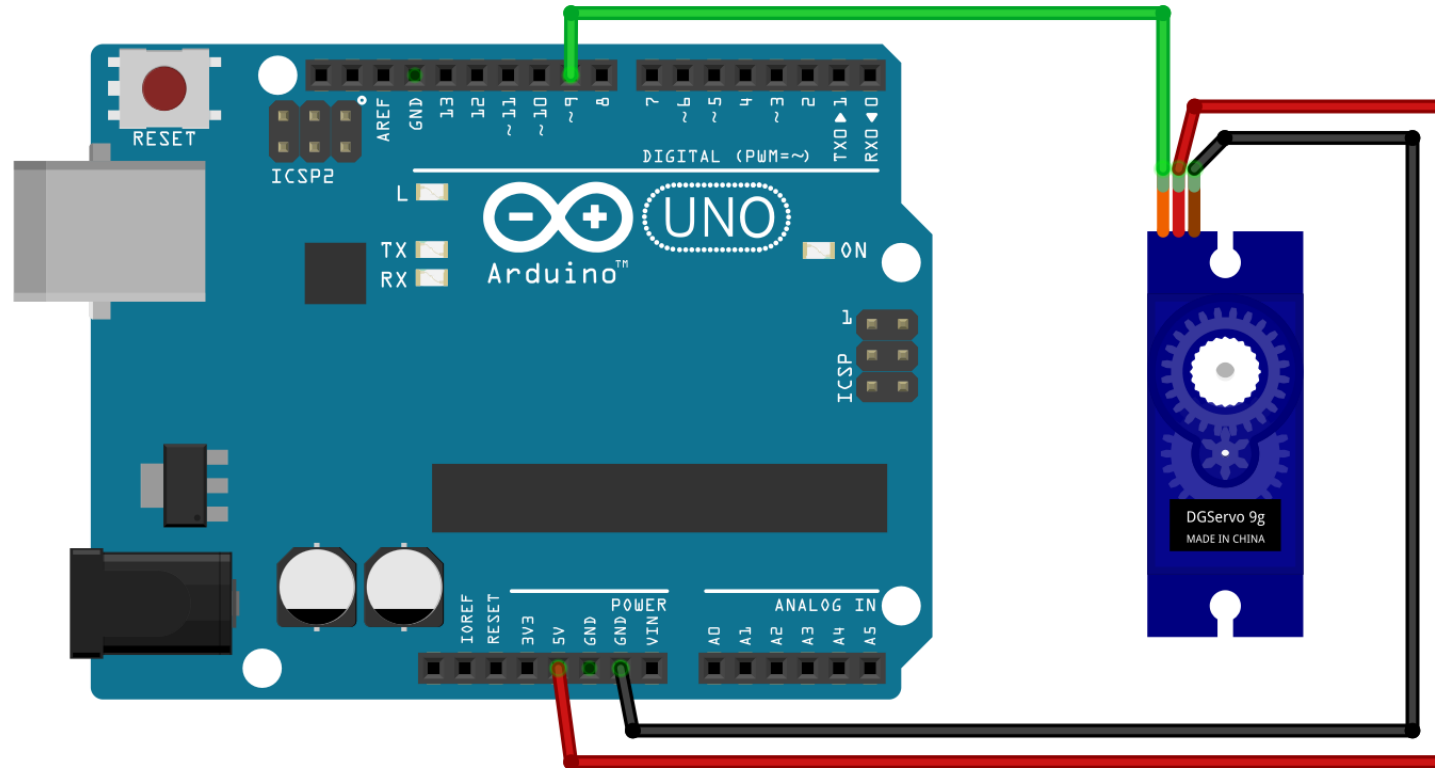
7.2 서보모터

- 서보모터의 제어
 - PWM 신호를 사용하여 제어
 - 각 펄스의 주기는 20ms로 생성하고 펄스값이 HIGH인 시간 폭을 1ms~2ms로 조절하면 회전각이 변화함



예제 7-3

- 회로구성



아두이노 우노 : 1개, 서보모터 : SG90, 점퍼선 여러 개

예제 7-3: 서보모터 구동 스케치

```
#include <Servo.h>

// 서보모터의 출력 핀 (PWM만 가능)
int pin = 9;

Servo motor;

void setup() {
    // 서보모터의 출력핀을 지정
    motor.attach(pin);
}

void loop() {
    // 30도로 회전
    motor.write(30);
    // 1초 대기
    delay(1000);

    // 150도로 회전
    motor.write(150);
    // 1초 대기
    delay(1000);
}
```

Servo 라이브러리 추가

- Servo 클래스 사용

- 아두이노 Servo 라이브러리에 정의된 클래스
- Servo::attach(핀번호) : 서보모터에 연결된 PWM 핀으로 초기화
- Servo::write(값) : 원하는 각도를 지정 (값이 200미만/이상)
 - 값이 200미만의 경우 : 각도로 인식
예] Servo::write(90);
 - 값이 200이상인 경우 : 마이크로초 단위의 펄스폭 사용
예] Servo::write(1500);