

## 6. DC모터와 RC서보모터

### 1) DC모터와 RC서보개요

DC모터는 바퀴 구동형 로봇을 만들 때 필요한 모터로서 360도 무한회전하는 모터이다. 반면에 RC서보모터는 로봇팔과 같은 관절형 로봇을 제작하는 데 사용된다.

### 2)연결방법

DC모터의 연결 핀과 서보 모터의 연결 핀의 위치는 아래와 같다.



**-DC모터:** 붉은색 사각형으로 표시된 위치가 DC모터가 연결되는 핀이다. 2개의 DC모터를 제어할 수 있다. M1과 M2는 DC모터를 별도로 제어할 수 있는 핀이지만 M1과 M3는 하나의 DC모터 드라이버를 공유하고 있어 별개로 제어할 수 없으므로 M1이나 M3 어디에나 연결해도 상관이 없다. M2와 M4 역시 동일한 DC모터 드라이버를 공유하고 있어 각각 별도로 제어할 수 없다. 배선할 때 극성은 위의 그림처럼 연결하면 된다.

**-SERVO모터:** 주황색(PORTD)과 파란색(PORTA) 사각형의 위치가 서보모터를 연결하는 핀의 위치이다. 주황색 핀과 파란색 핀의 차이는 전압에 있는데 주황색 핀에 연결하는 경우에는 서보모터에 외부 배터리 전압이 그대로 공급되고 파란색 핀에 연결하는 경우에는 5V전압이 서보모터에 공급된다. 만약 제어기에 연결된 배터리전압이 7V 인데 연결한 서보모터의 정격전압이 5V인 경우 서보모터의 파손이나 작동시 떨림이나 과열이 발생할 수 있으므로 주황색 위치에 연결할 때에는 유의해야 한다.

### 3) 아두이노함수설명

#### - 라이브러리추가하기

DC모터를 구동하기 위해서는 라이브러리가 필요하다. 아두이노 스케치의 메뉴상단에서

스케치 -> include library 에서 DCM 라이브러리를 추가하도록 한다. 특이한 점은 라이브러리 추가시 DCM.h 뿐만아니라 MsTimer2.h까지 자동적으로 추가된다는 점인데 이것은 DCM라이브러리가 MsTimer2 라이브러리를 내부에서 이용하고 있어서 함께 빌드되어야하기 때문이므로 삭제하지 않고 그대로 두어야한다. 라이브러리 추가 후에는 아래와 같이 객체를 생성하는 코드를 작성하도록 한다.

```
DCM mymotor1;
```

```
DCM mymotor2;
```

모터 객체의 이름은 당연히 달라질 수 있다.

서보모터를 사용하기 위해서도 역시 라이브러리 추가가 필요하다. 아두이노 스케치의 메뉴 상단에서 스케치→ include library에서 servo를 찾아 클릭하면 라이브러리가 추가 된다. 각각의 서보모터들을 제어하기 전에 먼저 아래와 같이 객체를 생성해야 한다.

```
Servo myservo1;
```

```
Servo myservo2;
```

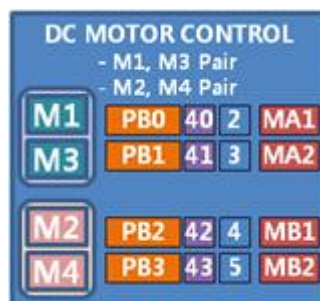
```
Servo myservo3;
```

역시 서보모터 객체의 이름은 임의로 정할 수 있다.

## -멤버함수: DC모터

```
attachPins(핀번호0, 핀번호1);
```

모터 1개를 구동하는 데 필요한 Logic핀을 설정한다. Logic 신호는 모터의 방향을 결정하는데 필요한 것이며 1개의 모터에 2개의 신호가 필요하다.. 아래 표를 보면 C와 D 두개의 핀에서 신호가 드라이버로부터 들어오고 있으며 각 핀의 신호상태에 따라 정회전, 역회전, 모터정지, 혹은 브레이크로 기능이 달라지는 것을 볼수있다.



Inputs		Function
$V_{en} = H$	$C = H ; D = L$	Forward
	$C = L ; D = H$	Reverse
	$C = D$	Fast Motor Stop
$V_{en} = L$	$C = X ; D = X$	Free Running Motor Stop

L = Low

H = High

X = Don'tcare

스마트인벤터보드에서는 DC모터드라이버가 내장되어 있으므로 logic핀 역시 고정되어 있

다. M1(M3)는 2번과 3번 핀이, M2(M4)는 4번과 5번 핀이 할당되어 있다. 따라서 아래와 같이 코딩하면 두 개의 모터를 사용할 수 있을 것이다.

```
mymotor1.attachPin(2, 3);  
mymotor2.attachPin(4, 5);
```

void write(모터방향, 속도);

모터의 속도와 방향을 지정해 주는 멤버이다. 첫번째 파라미터는 모터의 방향을, 두번째 파라미터 value는 모터의 속도를 지정한다. 모터의 방향은 아래와 같은 매크로 상수를 사용한다.

CW: 정회전 / CCW: 역회전 / STOP: 빠른정지 / LOOSE: 느린정지

속도는 0부터 100까지의 값을 사용한다.

### **-멤버함수: Servo모터**

attachPin(핀번호)

서보모터를 제어할 핀들을 설정한다. 서보모터를 연결한 핀의 번호를 파라미터로 입력하면된다. 예를 들어 PORTD의 27번과 28번에 서보모터를 연결하였다면 아래와 같이 입력하면 될 것이다.

```
myservo1.attach(27);  
myservo2.attach(28);
```

myservo1.write(서보모터각도)

파라미터에는 0 ~ 180도의 서보모터의 각도 값이 들어가는데 서보모터의 제조사나 종류마다 서보모터가 실제동작할 수 있는 각도 값은 다양하며 같은 값을 입력하였어도 서보모터의 종류마다 움직이는 각도의 위치는 다 다를수있다. 따라서 이 파라미터의 값은 사실상 상대 값이라고 보아야한다.

.

myservo1.detach( );

서보모터를 사용하다가 어떤 이유로 인가 제어기의 전원이 켜져있는 상태에서 서보모터의 축력을 풀어야 할 경우도있다. 이와 같은 경우엔 다음과 같은 멤버 함수를 사용한다. 이때 서보모터의 축을 돌려보면 힘이 풀려 쉽게 돌아가는 것을 볼수있다.

#### 4) 프로그래밍

**예제1:** IR리모컨을 사용하여 DC모터를 구동하는 예제이다. 보통 2륜 구동형 로봇을 리모컨으로 제어할 때 사용하는 전형적인 프로그램이다. 리모컨 버튼 구성에 대해서는 5장 IR 리모컨을 참고하기 바란다.

```
#include <IRremote.h>
#include <TVRemocon.h>           // TVRemocon 라이브러리추가
#include <MsTimer2.h>
#include <DCM.h>                 // DCM라이브러리추가
DCM motor1;                     // 객체선언
DCM motor2;

void setup()
{
    TVRemocon.begin();
    motor1.attachPins(2,3);      // 핀의설정은달라질수있음
    motor2.attachPins(4,5);
}

void loop()
{
    int data = TVRemocon.recieve();
    if(data >= 0)
    {
        if(data == KEY_U)       // U버튼이눌려진경우모터2개모두정회전
        {
            motor1.write(CW, 70); motor2.write(CW, 70);
        }
        else if(data == KEY_D)  // D버튼이눌려진경우모터2개모두역회전
        {
            motor1.write(CCW, 70); motor2.write(CCW, 70);
        }
        else if(data == KEY_L)  // L버튼은왼쪽모터가역회전하여방향전환
        {
            motor1.write(CCW, 70); motor2.write(CW, 70);
        }
    }
}
```

```

else if(data == KEY_R)    // R버튼은오른쪽모터가역회전하여방향전환
{
    motor1.write(CW, 70); motor2.write(CCW, 70);
}
else
{
    motor1.write(STOP, 70); motor2.write(STOP, 70);
}
}
}

```

**예제검토:** 위의프로그램은 리모컨 수신부에서 오는 리모컨 버튼 값을 입력받을 뿐 아니라 DC모터를 구동하기도 한다. 따라서 IR리모컨과 DCM 두종류의 라이브러리를 사용하며 이에 따른 setup()에서의 설정도 해주어야 한다.

**예제2:** RF리모컨으로 서보모터 각도 조절하기

*필요부품: 서보모터*

*연결방법:서보모터-> 27번*

아래에는 리모컨의 특정버튼을 눌렀을 때 서보모터가 반응하고 버튼에서 손을 떼었을때 다시원래의 각도로 돌아오는 프로그램이다.

```

#include <IRremote.h>
#include <TVRemocon.h>          //TVRemocon 라이브러리추가
#include <Servo.h>

Servo servo1;

void setup()
{
    TVRemocon.begin( );
    servo1.attach(27);
}

void loop()
{
    int data = TVRemocon.receive( ); //리모컨으로부터데이터수신
    If(data > 0)                      //새로운데이터를받았는가
    {

```

```
    If(data == KEY_R) servo1.write(10);  
    else if(data == KEY_L) servo1.write(170);  
    else servo1.write(90);  
  }  
}
```