메이크존 강좌좋음: <http://cafe.naver.com/makezone/113>

http://blog.whattomake.co.kr/163

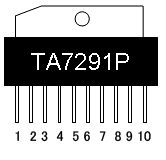
쇼핑몰: <http://artrobot.co.kr/>

쇼핑몰: <http://www.plughouse.co.kr/shop/>

http://blog.daum.net/\_blog/BlogTypeView.do?blogid=0aBw8&articleno=9&\_bloghome\_menu=recenttext#ajax\_history\_home

엊그제께 주문한 DC모터 드라이브 IC가 도착했다.

TA7291P



1 GND 아두이노 GND

2 출력단자1 DC모터단자  
3 미사용

4 PWM단자 아두이노 아날로그출력

5 입력단자1

6 입력단자2

7 아두이노 5V전원

8 외부전원

9 비접속

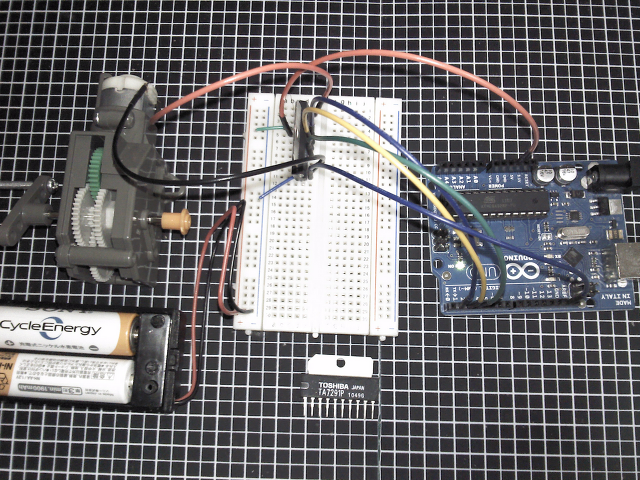
                                           10 출력단자2 DC모터단자

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 입력1 | 입력2 | 출력1 | 출력2 | 기능 |
| 0 | 0 | 무한대 | 무한대 | 정지 |
| 1 | 0 | H | L | 정전 |
| 0 | 1 | L | H | 역전 |
| 1 | 1 | L | L | 브레이크 |

모터용 전원은 아두이노에 공용으로 접속해도 된다. 사진찍을때 선이 복잡하게 꼬여서 알기쉽게 2.4V건전지로 해놓았는데... 역시 너무 느리다.

타미야 DC모터기어박스에 연결한 사진(서보모터아님)

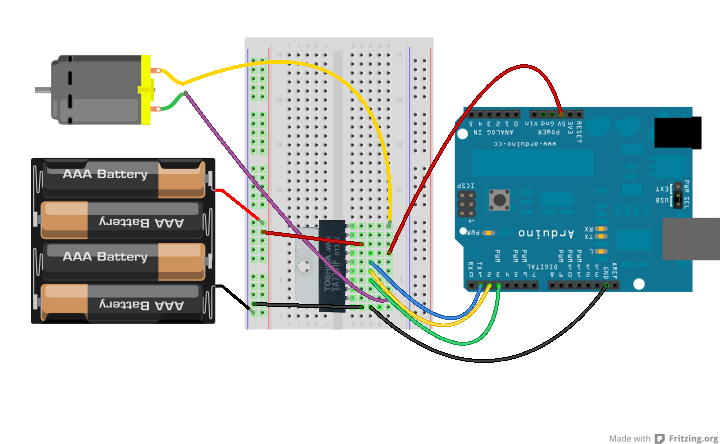
아래의 또하나 TA7291P까지 사용해서 2개모터 제어해야 탱크움직일텐데...



선이 안보인다. 이왕 블로그쓴김에 알기쉽게 툴로 그렸다.

Fritzing라는 아두이노 모듈이 이 갖춰진 무료 툴.아래 싸이트에서 다운받을 수 있다.

<http://fritzing.org/download/>



아두이노 스케치

테스트용으로 빨리 작성한거라서 실제로 TA7291P의 입력핀 틀렸다.

역방향/정방향이 반대로 움직임.

이부분

TA7291P의5번핀(입력1) - 아두이노 디지털출력1

TA7291P의6번핀(입력2) - 아두이노 디지털출력2

void setup(){

  //출력포트지정

  pinMode(1,OUTPUT);//TA7291P의5 -> 아두이노 입력1

  pinMode(2,OUTPUT);//TA7291P의6 -> 아두이노 입력2

  pinMode(3,OUTPUT);//TA7291P의4 -> 아두이노 입력3:출력조절(1-255)

}

void loop(){

    //3PIN 출력지정(랜덤으로 출력파워)

    int randNumber = random(255);

    analogWrite(3,randNumber);

    //모터정지

    digitalWrite(1,LOW);

    digitalWrite(2,LOW);

    delay(3000);

    //정방향

    digitalWrite(1,HIGH);

    digitalWrite(2,LOW);

    delay(3000);

    //역방향

    digitalWrite(1,LOW);

    digitalWrite(2,HIGH);

    delay(3000);

    //브레이크

    digitalWrite(1,HIGH);

    digitalWrite(2,HIGH);

    delay(3000);

}

준비물

아두이노

Bluetooth Mate Gold(<http://www.sparkfun.com/products/9358>)

LED

USB 블루투쓰 동글(데톱PC에서 아두이노에 접속/확인/테스트용 목적)

1.아두이노에 USB를 연결한후 아래코드를 업로드

(아직 블루투쓰 모듈 끼우지말것.같은 시리얼포트사용하게되므로)

#define LED 13

void setup(){

    Serial.begin(9600);

    pinMode(LED,OUTPUT);

}

void loop(){

    int length  = Serial.available();

    if(length>0){

        byte c=Serial.read();

        digitalWrite(LED,HIGH);

        delay(500);

        digitalWrite(LED,LOW);

        delay(500);

        Serial.write(c);

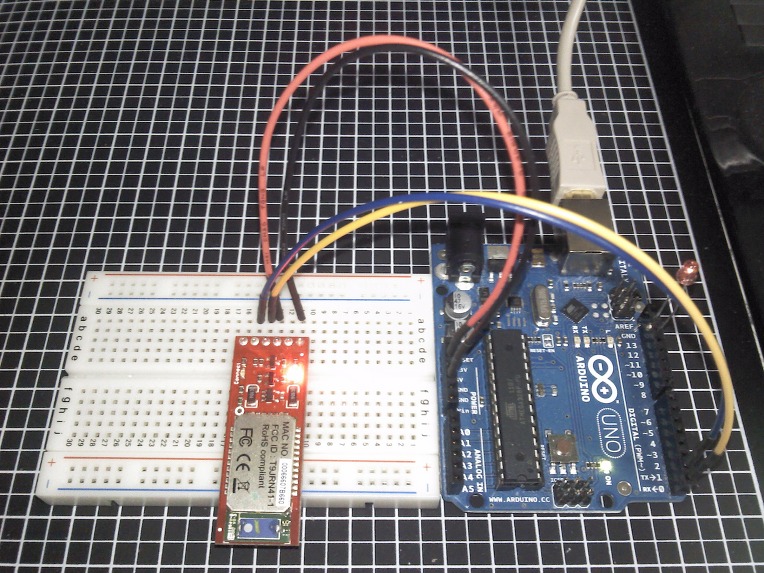
   }

}

2.LED,블루투쓰연결

블루투쓰모듈에 붙어있는 스테이트LED는 현재 상태를 보여주는데, 처음전원을 넣으면 빠르게 점멸을 반복(커맨드 환경설정모드:접속속도,이름등등)하다 느린점멸상태(일반사용모드)로 바뀐다로 전환된다.

구입 초기상태에서는 1분간 커맨드상태, 이후 일반모드로 전환된다. 즉,1분 후 사용가능한 상태가된다.



초기 시리얼포트 속도가 115K로 지정되어있다.

블루투쓰모듈이 커맨드상태일때(state LED가 빠르게 점멸) 아래의 명령으로 변경경설정이 가능하다.

->데탑블루투쓰로 연결후, $를 입력해서 커맨드상태 접속

D: 현재 설정상태표시

$

D

\*\*\*Settings\*\*\*

BTA=000XXXXXXXX

BTName=FireFly-XXX

Baudrt(SW4)=115K

Parity=None

Mode  =Slav

Authen=0

Encryp=0

PinCod=1234

Bonded=0

Rem=NONE SET

---END

SU,96

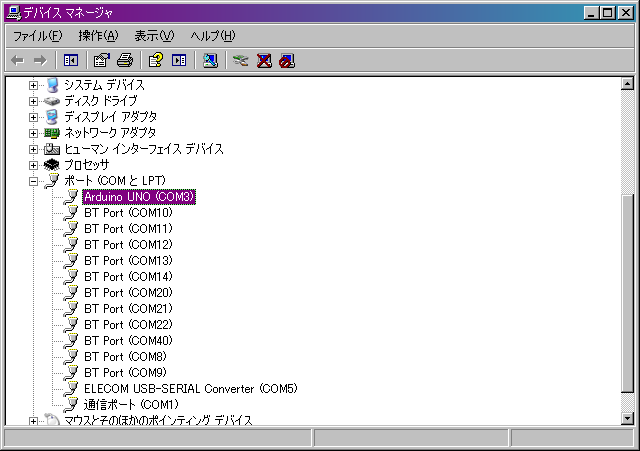
ST,15

D

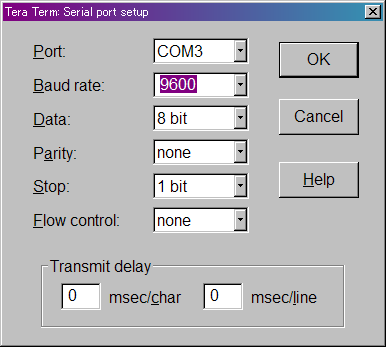
SU: 시리얼속도변경

ST: 커맨드모드 시간(15초,초기60초)

3.아두이노 IDE,나 제어판에서 아두이노 포트를 확인



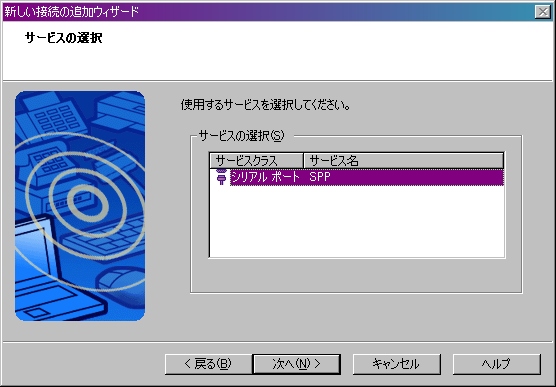
4.테라텀을 실행하고 포트를 지정한다.(아두이노 블루투쓰모듈 모니터링)



5.데톱PC에 저렴한 SPP지원되는 블루투쓰USB동글을 인스톨한다.(시험에 사용한건 IO Data USB-BT21)



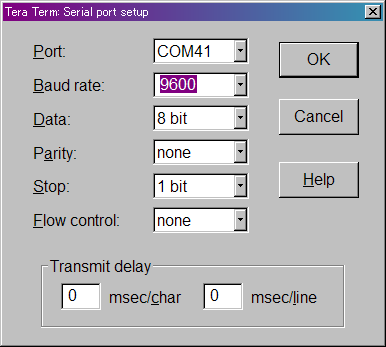
6.데톱에서 접속설정(FireFly-xxx 명의 아두이노쪽에 접속설정을한다.)



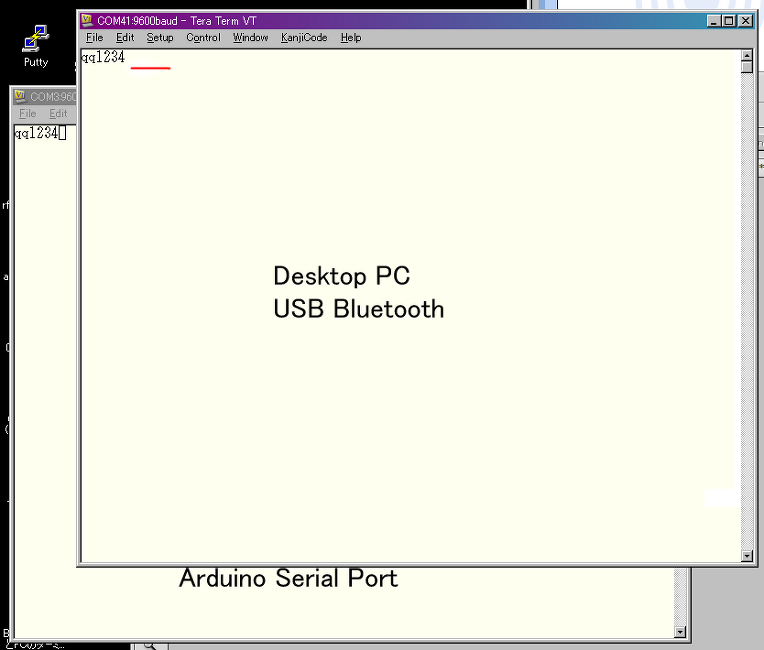
7.아두이노와 41번포트로 연결되었다.



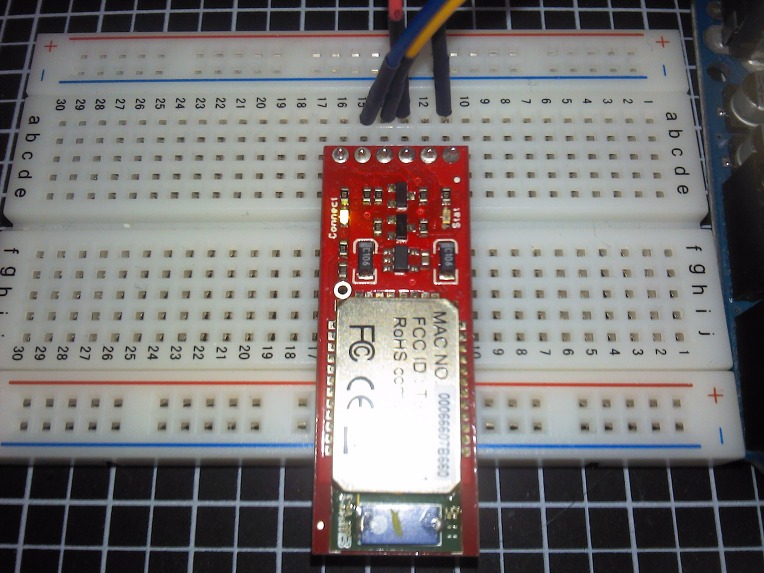
8.테라텀을 실행하고 시리얼 41번포트를 지정



9.데톱에서 입력한게 아두이노쪽 테라텀에도 표시되는지 확인

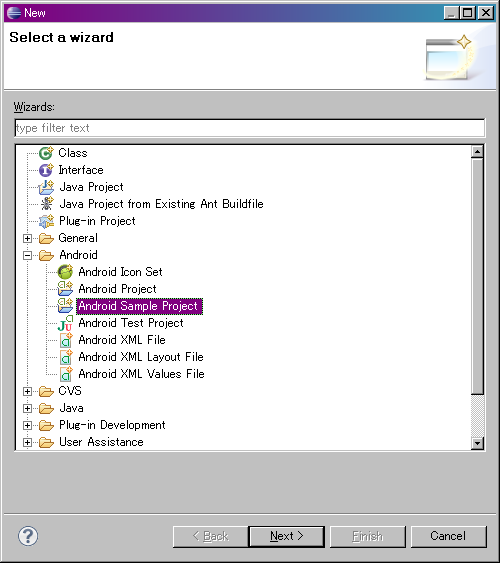


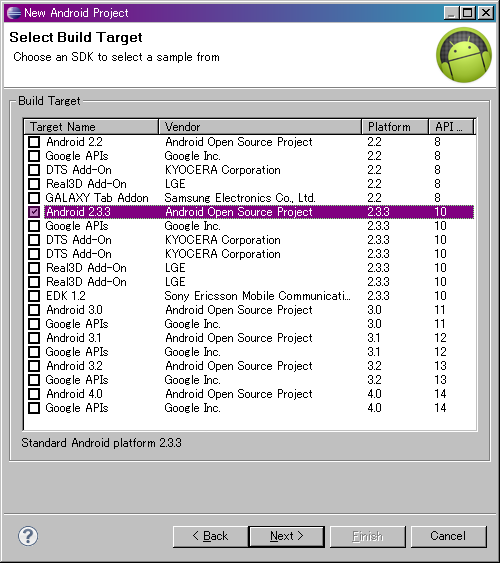
정상적으로 접속이 된상태

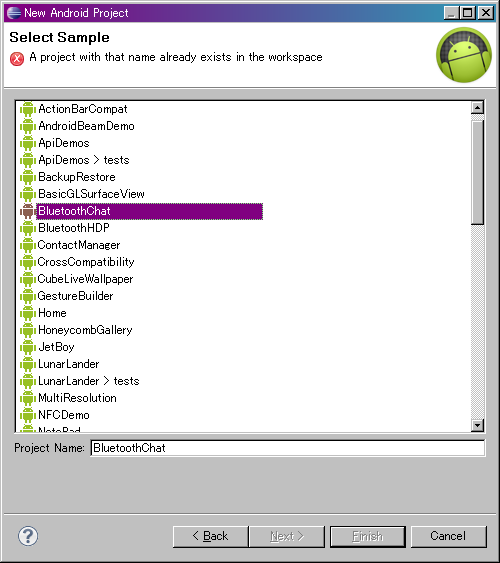


10.안드로이드 기기에서 확인

안드로이드 개발용 Eclipse 를 실행후, 샘플프로젝트로 블루투쓰채팅어플을 뽑아낸다.





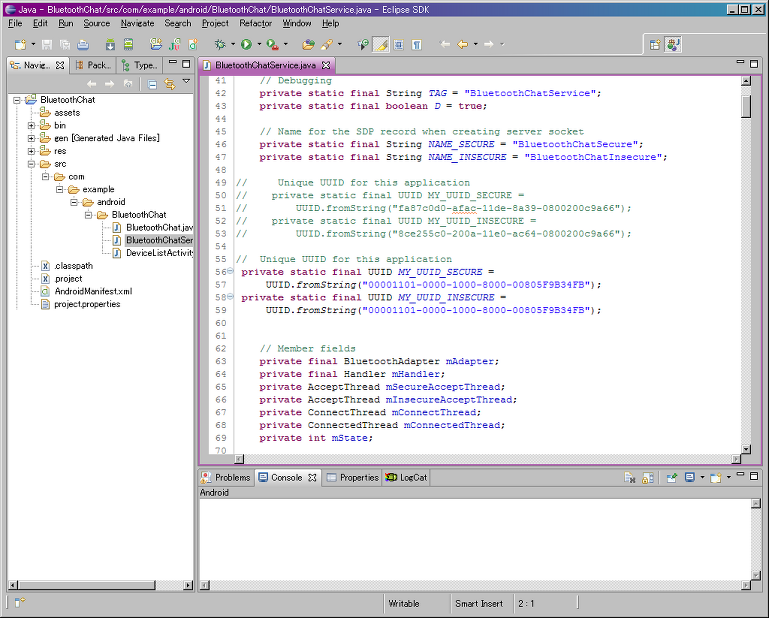


블루투쓰기기에는 정해진 번호가 있어서 이범위값을 넣지않으면 블루투쓰장치로 인식이되지 않는 문제가 있다고한다.

UUID를 아래의 번호로 변경한다.

 private static final UUID MY\_UUID\_SECURE =     UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

 private static final UUID MY\_UUID\_INSECURE =     UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

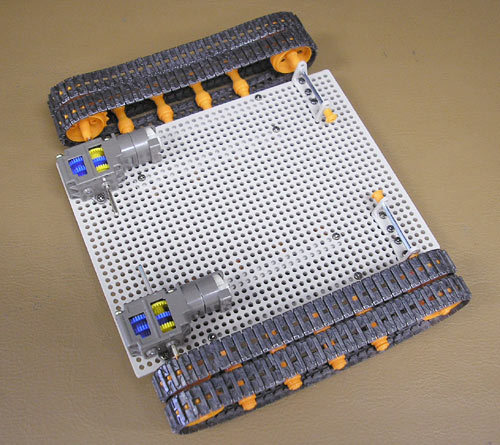


Export한후 안드로이드 어플팩키지 APK로 뽑아낸다.

안드로이드기기(핸드폰)에서 ES파일브라우저등으로 업로드후 인스톨.

위의 테라텀과 같이 접속후 채팅이 가능하다.

최근 타미야 유니버셜 보드를 구입했다.



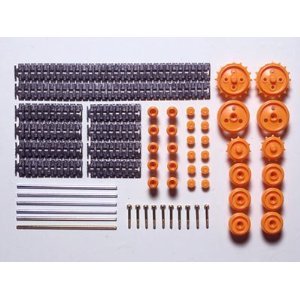
유니버셜세트는 어릴적 과학상자/ 레고 비슷한 것이라고 생각하면된다. 장난감 메이커답게 다양한 기어세트가 존재한다.



유니버셜기판



탱크용 바퀴 고무체인



탱크용 더블 모터



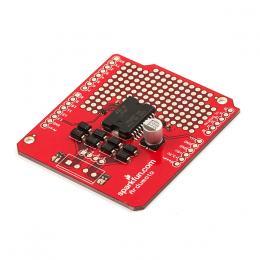
아무런 지식도없이 서보모터 제어와 같은줄알고 탱크를 사버렸다. DC모터제어가 안된다는 사실을알고 DC모터제어 IC를 구입했다.

도시바 TA7291P

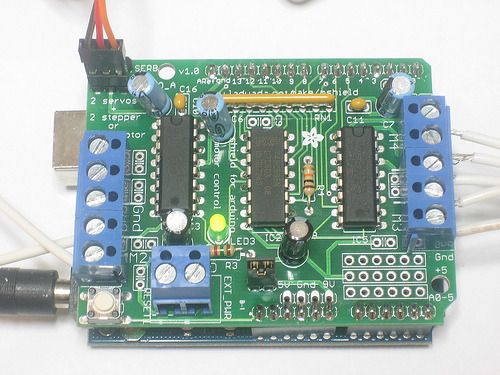


아래는 아두이노호환 모터 실드

달랑 DC모터2개 지원 -.-



다양한 모터를 지원하지만, 버뜨... 구입할 수 가없다.



초보답게 모터드라이버 IC부터 천천히 공부하면서 순서대로 하는게 나을듯...

s

**Arduino 탱크완성 & 회고**

프로젝트명 보롱 탱크 -\_-v

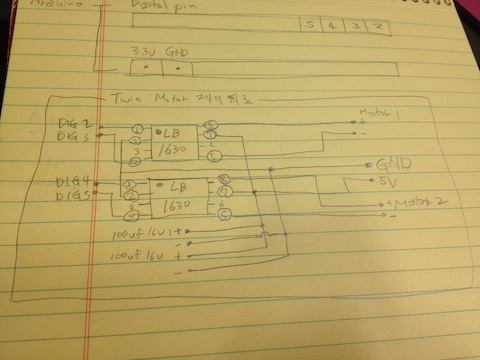
간단한 Outline sketch

* suapapa님 블로그에서 보고 급 땡겨서 간단하게 만들어보고자했다.
* 안드로이드 블루투스 제어방법이 좀 필요함 앞뒤 좌우 시그널을 어떤것으로 해야할지 정해야한다.

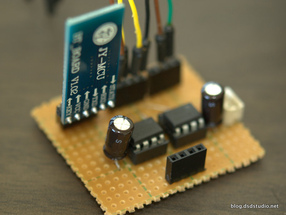
재료

* LB1630 모터제어 IC x 2
* 10uf 세라믹 캐패시터 x 2
* HC-06 블루투스 slave 모듈
* Arduino uno
* 아카데미 m1a1 모터라이즈 탱크
* 빵판 / 편의성을위한 IC 소켓 x 2, 전원 소켓 x 1, 소켓 헤더 여러개, 연결을 위한 점퍼선
* 3.7v Li-on 배터리 x 2개

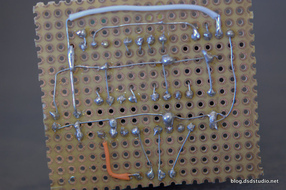
모터제어 회로도



모터제어 회로



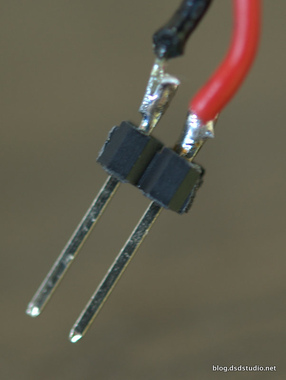
뒷면



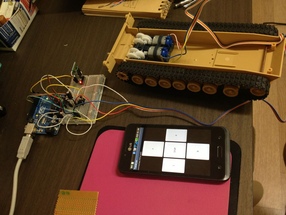
전원부는 모터가 전기를 워낙에 많이 빨아먹기때문에 3.7v arduino 입력만으로는 제대로 동작하지않는다. 전원 이중화 필수 !! 자주쓰지않는 핸드폰 리튬배터리를 재활용했다 ;)



헤더핀은 이런식으로 연선을 이어다가 쓰면 유용하다.



그래서 나온 프로토타입이 요기 !



오픈카형태로 조립후 동작영상

느끼거나 배운점

### 제 5강 DC 모터 구동

|  |
| --- |
| 수업에 필요한 물품목록 : Hello Board, 브레드 보드, ULN2803, 모터 구동모듈, DC 기어드 모터  [상세 사항 및 구매 링크](https://spreadsheets.google.com/spreadsheet/ccc?hl=en_US&hl=en_US&key=tbNjz_KikyXgUU3YG9LvHvA&authkey=CKrp7t8O#gid=1)  모터를 구동해보자.  **1. 모터 구동 기본**      우선 모터는 각각의 모터사양에 따라 구동전압이 다르다. 즉  5V 에 동작되는 모터는 5V 전압을 주면 되고 12V 에 동작되는 모터는 12V 를 주면된다.      하지만 우리는 단순히 모터가 동작되는 것을 구현하고자 하는 것이 아닌 사용자의 인터액션에 따라 모터를 구동하고자 한다.      이를 위해서 아두이노에서 보내주는 5V 신호를 각각의 모터에 맞는 전압으로 바꾸어주는 회로가 필요한 것이다.        간략한 회로는 아래와 같으며 더 자세한 내용은 아래 이미지의 출처인 [피지컬 컴퓨팅](http://book.naver.com/bookdb/book_detail.nhn?bid=4482067) 이라는 책을 참조하라.  [https://sites.google.com/site/bbaanng/_/rsrc/1303462969592/arduino/je-4gang-dc-motor-gudong/Motor_TR.JPG?height=210&width=400](https://sites.google.com/site/bbaanng/arduino/je-4gang-dc-motor-gudong/Motor_TR.JPG?attredirects=0) 출처 :  [피지컬 컴퓨팅](http://book.naver.com/bookdb/book_detail.nhn?bid=4482067)      어려울 것 없다. 우리는 이미 [제 2강 Full color LED bar](https://sites.google.com/site/bbaanng/arduino/2gang-analogwrite-maxmsp) 를 구동하기 위해 구성한 회로를 그대로 이용할 것이기 때문이다.      즉 Full color LED bar 부분을 Motor 로 대체하면 된다는 것이다. 참 쉽다.      비교를 위해 이전 Full color LED bar 의 연결 도식을 첨부한다. 두 이미지를 비교해보라.  https://sites.google.com/site/bbaanng/_/rsrc/1303461336963/arduino/je-4gang-dc-motor-gudong/full%20color%20led%20bar.png?height=266&width=400https://sites.google.com/site/bbaanng/_/rsrc/1303461361722/arduino/je-4gang-dc-motor-gudong/motorCtrl_basic.jpg?height=287&width=400      아니 오히려 회로구성이 줄었다. 기존에 RGB 세개의 포트가 1개로 줄었다.(빨간 LED 는 출력이 잘 나가는지 test 용으로 연결)  위와 같이 연결하고 아두이노의 Fading 이라는 예제를 실행하면 모터가 천천히에서 빠르게 빠르게에서 천천히 동작하는 것을 볼 수 있다.  기본예제는 아래와 같으며 포트(모터가 연결된 포트) 번호는 연결에 맞추어 바꾸어 주어야 한다.  int ledPin = 11;  void setup()  {  }  void loop()  {          for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5)          {                  analogWrite(ledPin, fadeValue);                  delay(30);          }          for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5)          {                  analogWrite(ledPin, fadeValue);                  delay(30);          }  }  **2. 센서값에 따른 모터 속도 제어**  위 코드에서 약간 변형하여 헬로보드의 슬라이드 센서 (A2) 아날로그 2번에 연결된 센서값에 의해 모터의 속도를 제어해보자.  코드는 아래와 같다.  int motorPort = 11;  int sensorVal = 0;  void setup()  {  }  void loop()  {          sensorVal = analogRead(A2);          sensorVal = map(sensorVal,0,1023,0,255);          analogWrite(motorPort,sensorVal);  }    슬라이드 센서에 의해 모터가 속도를 달리함을 알 수 있다.  2강의 설명처럼 모터구동에도 TR 의 증폭개념이 들어간다. 이는 앞 강의를 참조하자.  여기서.. 질문..  모터 속도는 센서에 의해 쉽제 제어를 하긴 했는데.. 방향제어를 하고 싶다면?  여기서 부터 약간 복잡해진다.  모터의 방향을 제어하기 위한 간략 회로는 아래와 같다.  **3. 모터 방향제어**  [https://sites.google.com/site/bbaanng/_/rsrc/1303463083164/arduino/je-4gang-dc-motor-gudong/Motor_Hbridge.JPG?height=364&width=400](https://sites.google.com/site/bbaanng/arduino/je-4gang-dc-motor-gudong/Motor_Hbridge.JPG?attredirects=0) 출처 :  [피지컬 컴퓨팅](http://book.naver.com/bookdb/book_detail.nhn?bid=4482067)  자세한 내용은 원서에 있지만 간략히 설명해보자.  모터용전압에서 12V 가 나온다. 여기서 1번과 4번이 연결이 된다면 모터용 전압이 모터의 오른쪽 선을 거쳐 왼쪽으로 나와 4번연결을 거쳐 그라운드로 빠진다.  그러면 모터는 하나의 방향(여기서은 오른쪽으로 정하자) 즉 오른쪽으로 모터가 회전한다.  다른 경우 3번과 2번이 연결이 된다면 모터용 전압이 모터의 왼쪽 선을 거쳐 오른쪽으로 나와 2번 연결을 거쳐 그라운드로 빠진다.  그러면 모터는 왼쪽으로 회전한다.  이러한 H-bridge 구조를 통해 모터의 방향을 제어하는 것이다.  이론은 간단하다. 하지만 전기에 대해 조금 아는 사람이라면 여기서 아주 위험한 상황을 발견하였을것이다.  그것은 바로 1번과 2번 이 같이 연결되었을때인데 이는 모터용 전압이 1번과 2번을 바로 거쳐 그라운드로 연결되는 쇼트회로를 구서하는 것이다.  이는 집에서 220V 전원코드에 젓가락을 꼿는 어린이의 아주 위험한 행동에 해당된다.  아 복잡하다.. 타이밍이 문제다. 1번 켜고 2번 켜면 절대 안되고 1번 켜고 4번 켜고 다시 방향 틀려면 4번 끄고 1번 끈 다음 2번 켜고 3번켠다.  복잡하다..  뭐 여러가지 방법이 있지만 여기서 우리는 이러한 것들을 이론.. ' 아 이렇게 모터의 방향을 제어하는구나! ' 하고 흐름만 알자  **4. 모터 방향제어에 모터 구동드라이버 이용하기**  그래서 준비한 것이 모터 구동 드라이버이다. 가격은 2만원대 2개의 모터를 방향제어와 속도제어가 가능하다.  대단하다.. (사실 회로를 구성할 줄 안다면 직접 해당 칩을 구매하여 회로를 구성해도 된다.) 이러한 기능을 2만원에.. ^^  우리가 실습에 쓰고자 하는 드라이버는 아래와 같다. 구입처는 디바이스 마트  [DC 모터 구동모듈 Dual 2A (AM-DC2-2D)](http://devicemart.co.kr/mart7/mall.php?cat=017012005&query=view&no=18870)  위와 같다. 앗 복잡하다..  거두절미하고. 아래 핀 번호 맵을 보자.  **5. 모터 방향제어 회로 구성하기**  1번 : PWM1 : 첫번째 모터의 속도를 제어하는 핀이다. 아두이노 코드로는 analogWrite(10,128); 10번핀에 128의 속도(0~255) 이므로 중간정도의 속도  2번 : DIR1 : 이곳에 신호를 HIGH(5V) 를 주면 방향이 정방향 그리고 반대로 LOW(0V) 를 주면 역방향으로 회전한다. 아두이노 코드로는 digitalWrite(11,HIGH);  3번 : /Enable1 : 말 그대로 모터를 켤껀지 끌껀지를 결정한다. 즉 PWM 의 신호와 상관없이 이것이 ON 되어야 모터가 동작한다.                          유의할 적음 Enable 앞에 ' / ' 붙었다. 이는 신호를 반대로 인식한다는 의미이다. 즉. 모터를 켜고자 하면 LOW(0V) 를 끄고자하면 HIGH(5V)                          를 주어야 한다는 것이다. 아두이노 코드로는 digitalWrite(12,LOW);  5,6,7번 : 위와 동일 2번째 모터 구동.  9번 : 모터 구동 드라이버의 전원 연결 : 이곳에 5V 를 연결하여야 구동된다.  10번 : 모더 구동 드라이버의 그라운드 연결 : 이곳에 0V(GND) 를 연결  [https://sites.google.com/site/bbaanng/_/rsrc/1303467604916/arduino/je-4gang-dc-motor-gudong/motor_circuit.jpg](https://sites.google.com/site/bbaanng/arduino/je-4gang-dc-motor-gudong/motor_circuit.jpg?attredirects=0)  위와 같이 핀헤더에 래핑와이어를 연결하여 아래의 모터 구동 드라이버에 연결할 수 있도록 준비합니다.  래핑와이어의 색으로 핀을 구별합니다.      흰색    :  /Enable1(활성화)         Arduino : pin 13      파란색 : DIR1(방향)                  Arduino : pin 12      노란색 : PWM1(속도)                Arduino : pin 11          흰색    :  /Enable2(활성화)        Arduino : pin 7      파란색 : DIR2(방향)                  Arduino : pin 6      노란색 : PWM2(속도)                Arduino : pin 5  아래의 드라이버에 핀에 맞추어 연결합니다.    [https://sites.google.com/site/bbaanng/_/rsrc/1303467742113/arduino/je-4gang-dc-motor-gudong/2007111915263811%20-%20%EB%B3%B5%EC%82%AC%EB%B3%B8.jpg](https://sites.google.com/site/bbaanng/arduino/je-4gang-dc-motor-gudong/2007111915263811%20-%20%EB%B3%B5%EC%82%AC%EB%B3%B8.jpg?attredirects=0)  **6. 모터 방향제어 코드 작성하기**  위와 같이 연결하고 아래의 코드를 넣습니다.  #define          ON 1  #define OFF 0    const int Motor\_A\_Enable = 13;  const int Motor\_A\_DIR = 12;  const int Motor\_A\_PWM = 11;  int button = 4;  int sensorVal;  int buttonStatus;  void setup()  {  pinMode(Motor\_A\_DIR, OUTPUT);  pinMode(Motor\_A\_Enable, OUTPUT);  Serial.begin(9600);  }  void loop()  {  sensorVal = analogRead(A2);  sensorVal = map(sensorVal,0,1023,0,255);            buttonStatus = digitalRead(button);            if(buttonStatus==LOW)          {                  digitalWrite(Motor\_A\_Enable,LOW);          }          else          {                  digitalWrite(Motor\_A\_Enable,HIGH);          }            analogWrite(Motor\_A\_PWM,sensorVal);  delay(2);  }  위와 같이 코드를 작성하고 업로드를 하면 Motor 는 헬로보드의 버튼(디지털4번에 연결됨)이 눌려져 있을때만 동작되며  슬라이드센서(A2,아날로그2번에 연결됨)의  위치에 따라 모터의 속도가 0에서 255의 속도로 조절된다. 여기서 숫자 255는 모터의 최대 속도를 말한다.  그럼 여기서 버튼을 모터의 Enable(활성화) 가 아닌 DIR(방향제어) 핀에 연결하여 버튼을 누를때마다 방향이 바뀌는 동작을 구현해 보자.  코드는 아래와 같다.  #define          ON 1  #define OFF 0    const int Motor\_A\_Enable = 13;  const int Motor\_A\_DIR = 12;  const int Motor\_A\_PWM = 11;  int button = 4;  int sensorVal;  int buttonStatus;  void setup()  {  pinMode(Motor\_A\_DIR, OUTPUT);  pinMode(Motor\_A\_Enable, OUTPUT);          digitalWrite(Motor\_A\_Enable,LOW);  Serial.begin(9600);  }  void loop()  {  sensorVal = analogRead(A2);  sensorVal = map(sensorVal,0,1023,0,255);            buttonStatus = digitalRead(button);            if(buttonStatus==LOW)          {                  digitalWrite(Motor\_A\_DIR,LOW);          }          else          {                  digitalWrite(Motor\_A\_DIR,HIGH);          }            analogWrite(Motor\_A\_PWM,sensorVal);  delay(2);  }  이처럼 상황에 따라 Enable 핀과 DIR 핀을 제어해주면 된다. 코드는 보통의 LED 를 켜고 끄는 로직과 같다.  여기까지 기본적인 구현은 끝이났다..  **6. 2개의 모터를 MaxMSP(윈도우 프로그램)에서 제어하기**  아래는 작품의 다양성을 위해 MaxMSP 에서 받은 값들(방향,활성화,속도)을 바로 제어가능하도록 연동하는 코드를 만들어 보았다.  쉽게 MaxMSP 에서 모터를 제어할 수 있다. 게다가 2개씩이나..  [https://sites.google.com/site/bbaanng/_/rsrc/1335446678633/arduino/je-4gang-dc-motor-gudong/DCmotorCtrlMaxCapture.png](https://sites.google.com/site/bbaanng/arduino/je-4gang-dc-motor-gudong/DCmotorCtrlMaxCapture.png?attredirects=0)  위 패치의 기본구조는 2강의 Full color LED 제어와 흡사하다. 여러개의 data 를 묶어서 아두이노로 보내주는 형식.  서로 포멧을 정해서 255, A모터 활성화, A모터 방향, A모터 속도, B모터 활성화, B모터 방향, B모터 속도, 254  의 형식으로 데이터를 보낸다. 이를 아두이노에서 각각의 정보를 받아 2개의 모터를 제어한다.  위의 MaxMSP 패치와 연동되는 아두이노 코드는 아래와 같다.  #define        ON 1  #define        OFF 0  #define        N\_DATA        8  const int Motor\_A\_Enable = 13;  const int Motor\_A\_DIR = 12;  const int Motor\_A\_PWM = 11;  const int Motor\_B\_Enable = 7;  const int Motor\_B\_DIR = 6;  const int Motor\_B\_PWM = 5;  int sensorVal;  unsigned int f\_motor\_A\_Enable,f\_motor\_B\_Enable,f\_motor\_A\_Direction,f\_motor\_B\_Direction,f\_motor\_A\_Pwm,f\_motor\_B\_Pwm;  int i;  unsigned int Data[8];  void setup()  {          pinMode(Motor\_A\_DIR, OUTPUT);          pinMode(Motor\_A\_Enable, OUTPUT);          pinMode(Motor\_B\_DIR, OUTPUT);          pinMode(Motor\_B\_Enable, OUTPUT);          Serial.begin(9600);  }  void loop()  {          if (Serial.available() > 0)          {                  Data[i]=Serial.read();                  if(Data[0]==255)                  {                          if(i<(N\_DATA-1))  i++;                          else                          {                                  if(Data[(N\_DATA-1)]==254)                                  {                                          f\_motor\_A\_Enable = Data[1];                                          f\_motor\_A\_Enable = ! f\_motor\_A\_Enable;                                          f\_motor\_A\_Direction = Data[2];                                          f\_motor\_A\_Pwm = Data[3];                                          f\_motor\_B\_Enable = Data[4];                                          f\_motor\_B\_Enable = ! f\_motor\_B\_Enable;                                          f\_motor\_B\_Direction = Data[5];                                          f\_motor\_B\_Pwm = Data[6];                                  }                                  i=0;                          }                  }                  else  i=0;          }          digitalWrite(Motor\_A\_Enable,f\_motor\_A\_Enable);          digitalWrite(Motor\_B\_Enable,f\_motor\_B\_Enable);          digitalWrite(Motor\_A\_DIR,f\_motor\_A\_Direction);          digitalWrite(Motor\_B\_DIR,f\_motor\_B\_Direction);          analogWrite(Motor\_A\_PWM,f\_motor\_A\_Pwm);          analogWrite(Motor\_B\_PWM,f\_motor\_B\_Pwm);  }  MaxMSP code  //코드 첨부함.  위와같이 실행하면 MaxMSP 에 의해 모터를 제어할 수 있을 것이다. 우선 컴퓨터 키보드 'Q'키에 의해 A 모터의 Enable 제어와 'W' 키로 A 모터의 DIR(방향)제어를 할 수 있다. 마찬가지로 키보드의 'O' 키에 의해 B 모터의 Enable 과 'P' 키로 B 모터의 DIR 을 제어할 수 있다.  오른쪽 상단에 위치한 number box 에 마우스를 가져가서 드래그하면 값이 바뀌면서 각각의 모터의 속도가 변하는 것을 볼 수 있다.  자 이제 내가 준비한 기술적인 것들은 끝이 났다.  이제 다음은 여러분들이 작품을 만들 시간이다.  (이해와 여러가지 가능성을 염두함에 따라 컴퓨터의 제어(MaxMSP)로 준비하였다. 위 내용들을 모두 이해한다면 아두이노 독립적으로 각각의 모터가 동작되는  재미난 작품을 만들 수 있을 것이다.  그럼 나머지는 여러분들의 작품 링크가 실리는 일만 남았다.  작품 링크 : ............................  코멘트나 의견은  이메일로 받겠다.(아직 위키 블로그 사용법을 모름.)  kinsomnia@gmail.com |