

**Copyright 2021. (Univ. of Seoul) All rights reserved.**

- 수강생만 시청, 시청 후 삭제
- 변경, 복사, 배포 절대 금지

# 모형RC카 제작



# 모형차 제작, 아두이노로 구동

## • 1단계: 모형차 제작

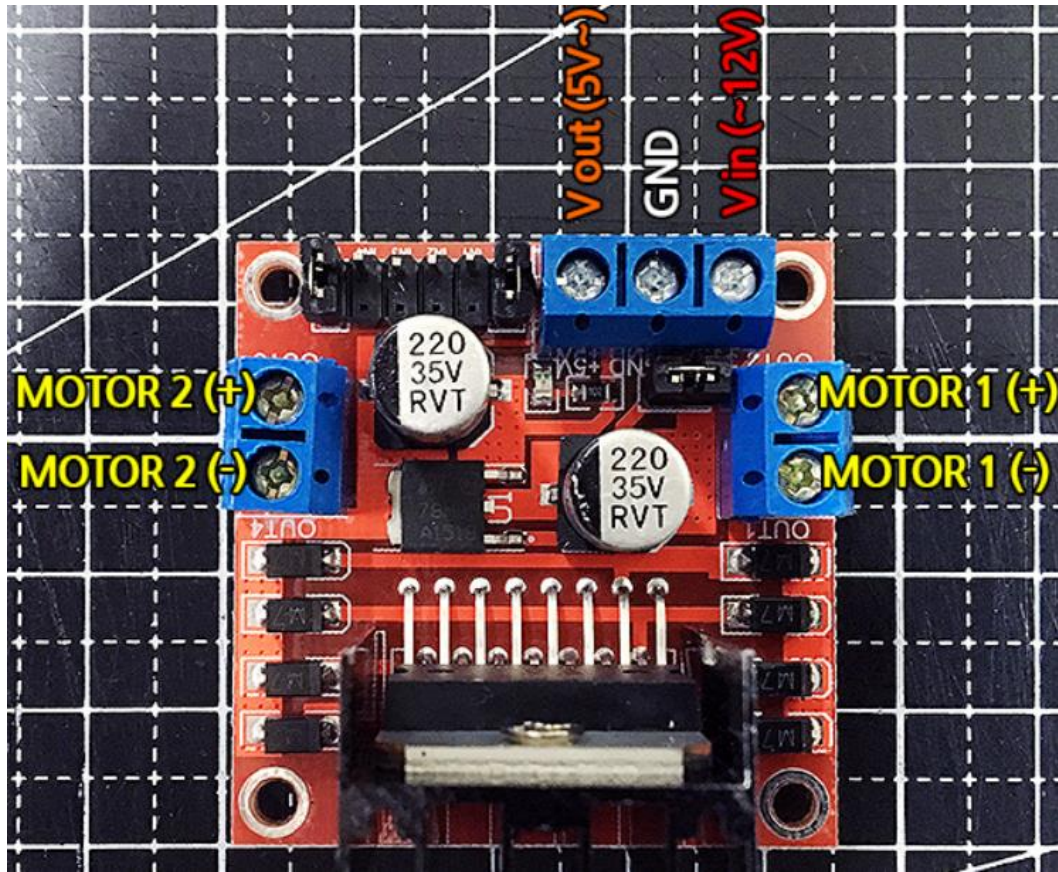
- 3가지 종류의 모형차 보급, 모터드라이버로 구분
  - L9110S 30세트
  - L298N 6세트 (미니멀 버전)
  - L293D 2세트
- L9110S와 L293D 모형차는 설명서대로 조립

## • 2단계: 아두이노로 모형차 구동

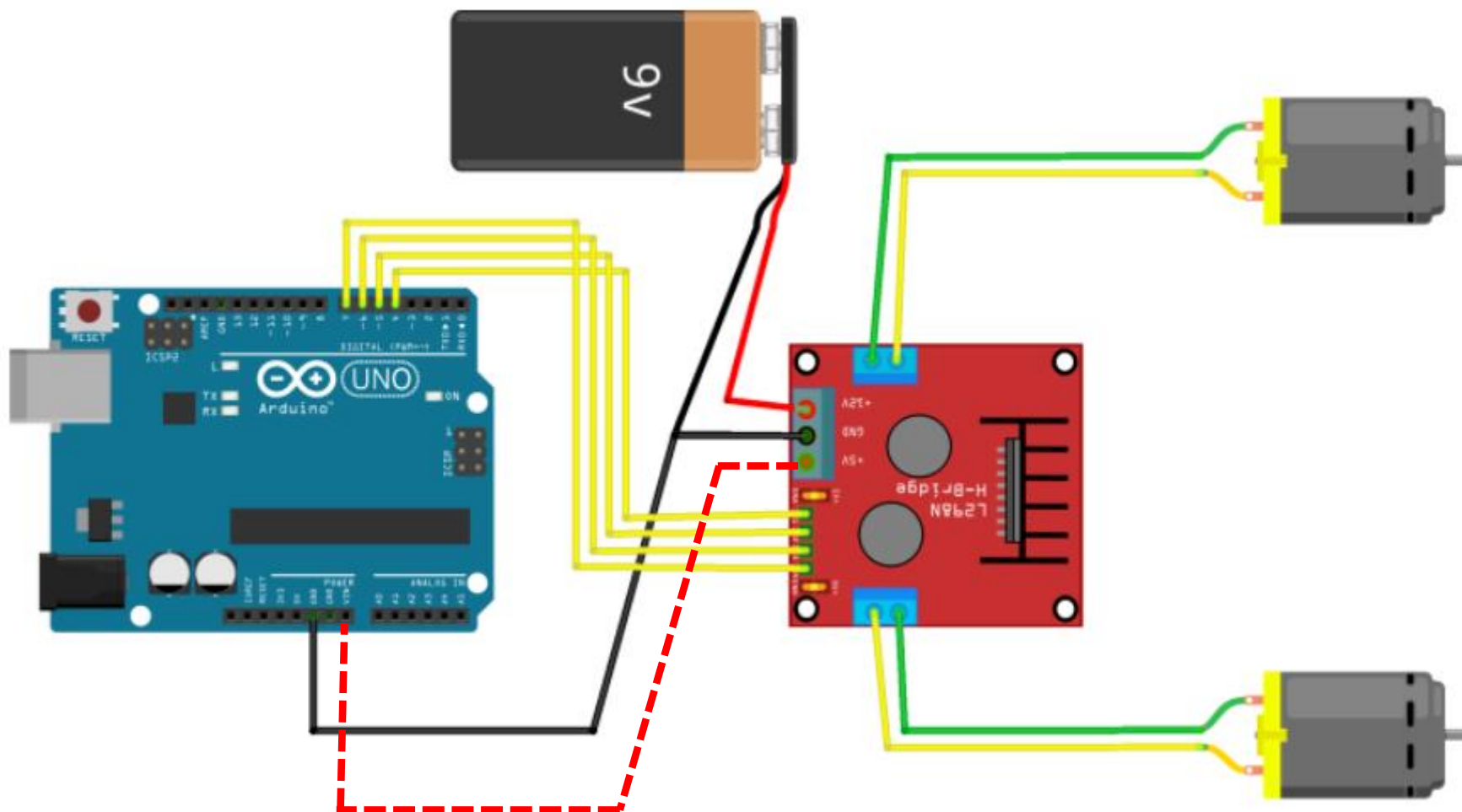
- L9110S와 L293D 모형차의 경우 설명서를 보고 아두이노 프로그램을 다운받아 실행
- L298N(미니멀 버전)은 뒤에 있는 슬라이드를 보고 아두이노 보드로 구동



# L298N 모터드라이버



# L298N 모터드라이버



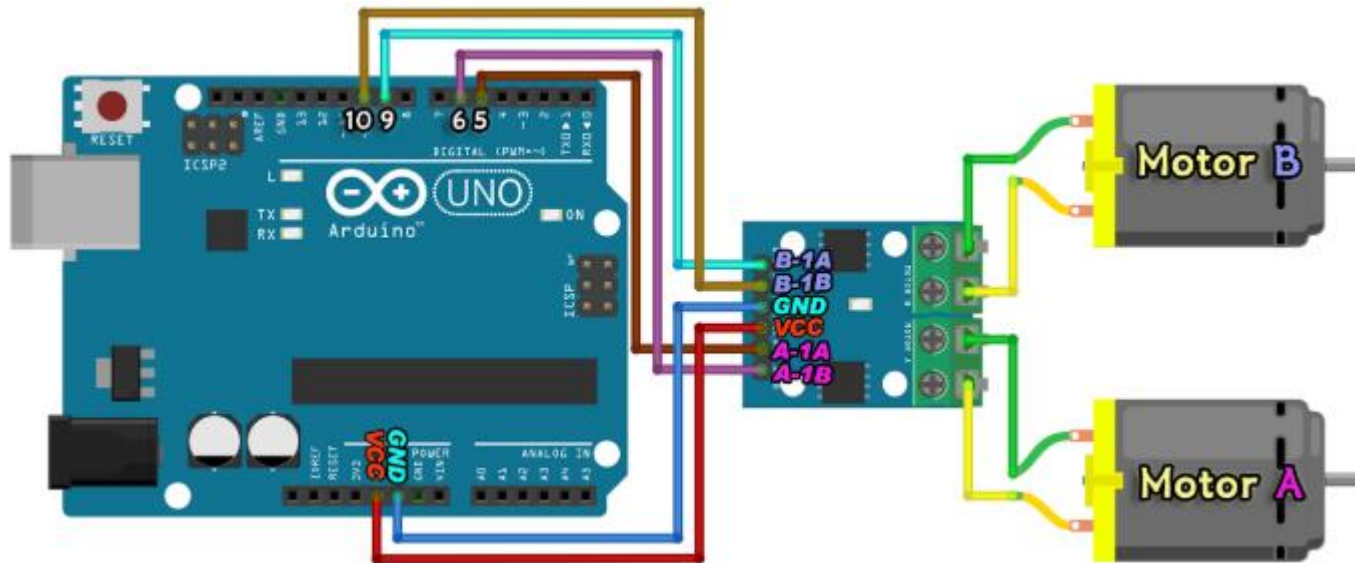
점선은 선택사항 (9V 배터리로부터 아두이노로 전원을 공급하는 경우에만 연결)

# L298N 모터드라이버

```
void setup() {  
    pinMode(4, OUTPUT);  
    pinMode(5, OUTPUT);  
    pinMode(6, OUTPUT);  
    pinMode(7, OUTPUT);  
}  
void loop() {  
    digitalWrite(4, HIGH);  
    digitalWrite(5, LOW);  
    digitalWrite(6, HIGH);  
    digitalWrite(7, LOW);  
    delay(3000);  
    digitalWrite(4, LOW);  
    digitalWrite(5, HIGH);  
    digitalWrite(6, LOW);  
    digitalWrite(7, HIGH);  
    delay(3000);  
}
```



# L9110S 모터드라이버



# L9110S 모터드라이버

```
int motorA1 = 5;
int motorA2 = 6;
int motorB1 = 9;
int motorB2 = 10;
int speed = 255;
```

```
Void setup() {
    pinMode(motorA1, OUTPUT);
    pinMode(motorA2, OUTPUT);
    pinMode(motorB1, OUTPUT);
    pinMode(motorB2, OUTPUT);
}
```

```
void loop() {
    /* 전진 */
    analogWrite(motorA1, speed);
    analogWrite(motorA2, 0);
    analogWrite(motorB1, speed);
    analogWrite(motorB2, 0);
    delay(2000);

    /* 후진 */
    analogWrite(motorA2, speed);
    analogWrite(motorA1, 0);
    analogWrite(motorB2, speed);
    analogWrite(motorB1, 0);
    delay(2000);
}
```



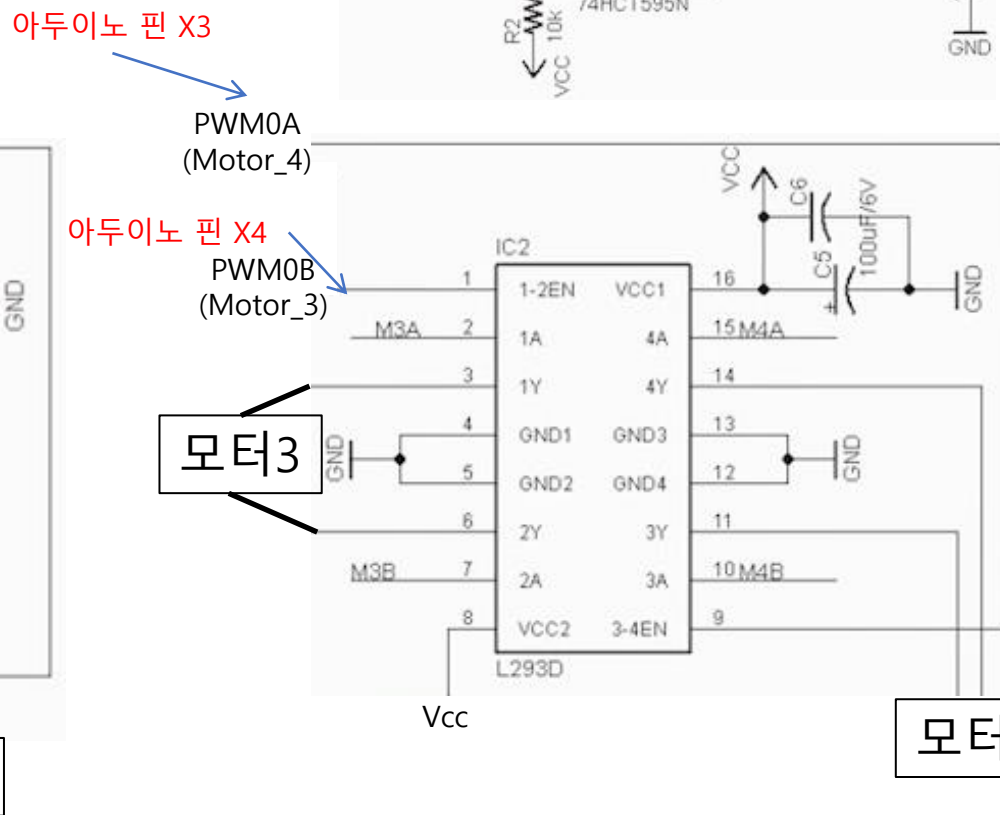
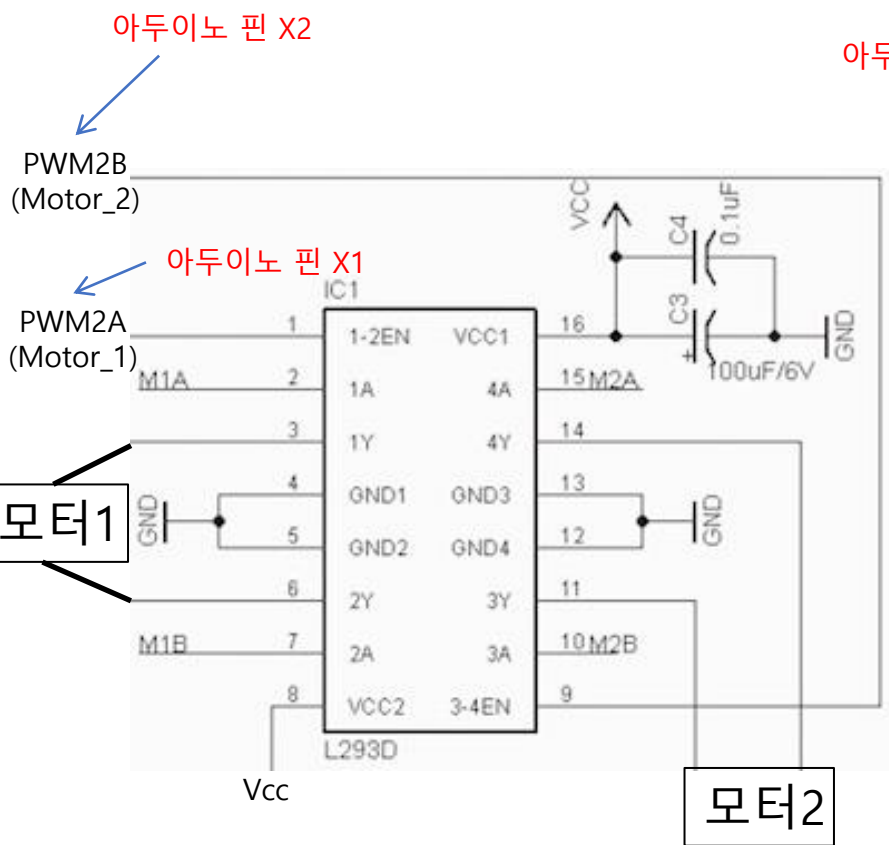


# L293D 모터드라이버шил드

## • Adafruit L293D Motor Driver Shield



- L293D 칩 2개,  
74HCT595N 칩 1개

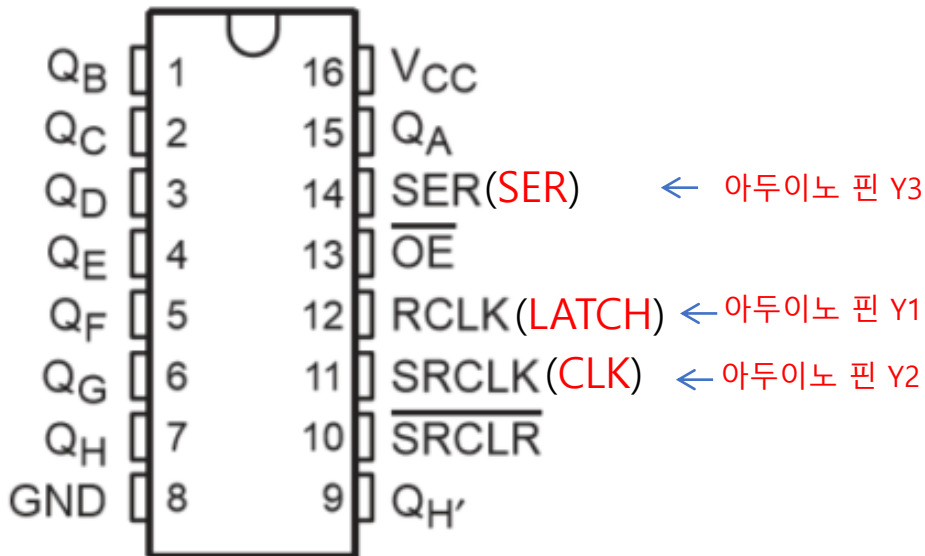


# 74HCT595 쉬프트 레지스터

## • 74HCT595 8bit 쉬프트 레지스터 칩

- 부족한 출력 핀을 확장
- SER 핀으로 1비트씩 제공하면서 CLK를 LOW에서 HIGH로 올림
- 위 과정을 8번 반복하면서  $Q_A \sim Q_H$ 까지 출력 값 설정
- LATCH를 LOW에서 HIGH로 올리면 설정 값이 실제로  $Q_A \sim Q_H$  핀으로 출력됨

74HC595 Shift Register



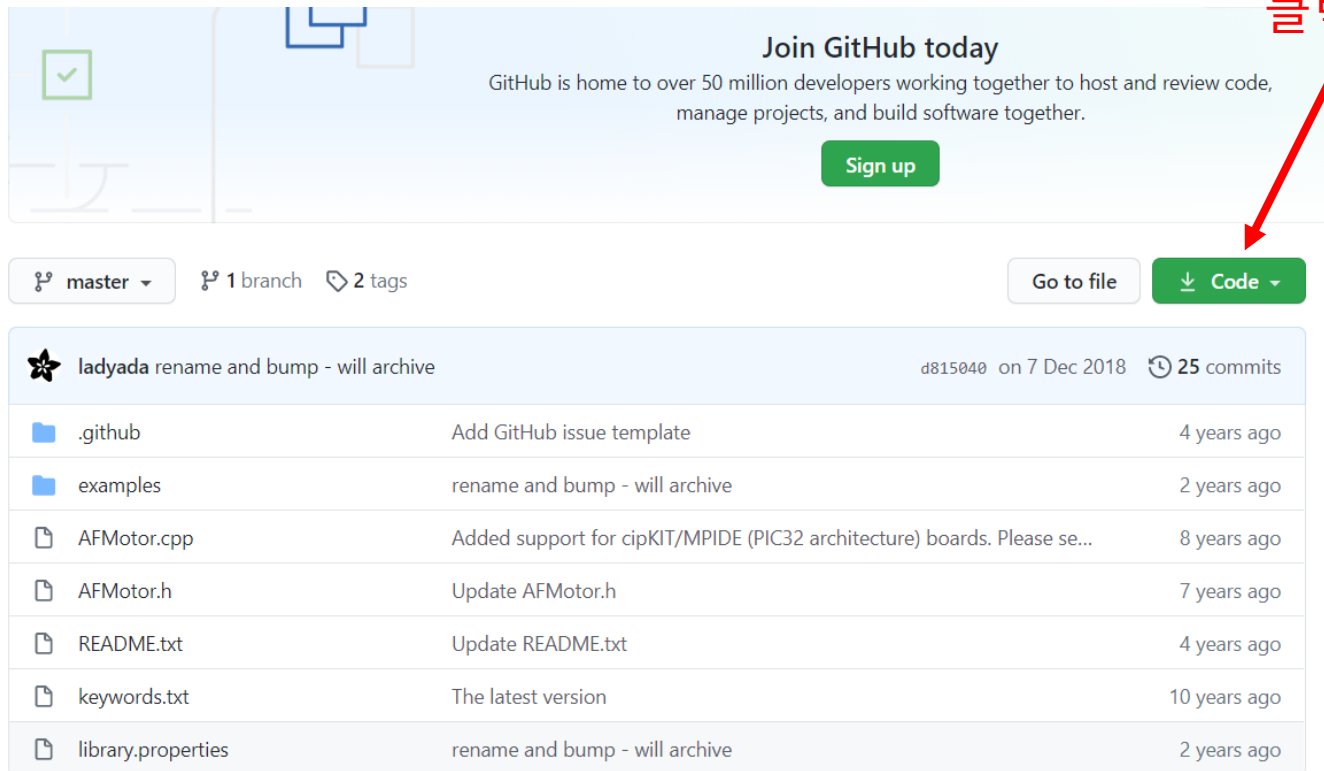
```
def _shift_write(self, value):
    GPIO.output(self._DIR_LATCH, GPIO.LOW)
    for x in range(0, 8):
        temp = value & 0x80
        GPIO.output(self._DIR_CLK, GPIO.LOW)
        if temp == 0x80:
            # data bit HIGH
            GPIO.output(self._DIR_SER, GPIO.HIGH)
        else:
            # data bit LOW
            GPIO.output(self._DIR_SER, GPIO.LOW)
        GPIO.output(self._DIR_CLK, GPIO.HIGH)
        value <<= 0x01 # shift left

    GPIO.output(self._DIR_LATCH, GPIO.HIGH)
```

# Adafruit 모터샐드 라이브러리

- 다운로드 사이트

- <https://github.com/adafruit/Adafruit-Motor-Shield-library>



Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

master 1 branch 2 tags

Go to file

Code

ladyada rename and bump - will archive d815040 on 7 Dec 2018 25 commits

.github	Add GitHub issue template	4 years ago
examples	rename and bump - will archive	2 years ago
AFMotor.cpp	Added support for cipKIT/MPIDE (PIC32 architecture) boards. Please se...	8 years ago
AFMotor.h	Update AFMotor.h	7 years ago
README.txt	Update README.txt	4 years ago
keywords.txt	The latest version	10 years ago
library.properties	rename and bump - will archive	2 years ago

클릭, 다운로드

# Adafruit 모터쉴드 라이브러리

- 압축해제, 아두이노 설치폴더 아래 libraries 폴더로 복사

보기

내 PC > 로컬 디스크 (C:) > arduino-1.8.13 > libraries

이름	수정한 날짜	유형
Adafruit_Circuit_Playground	2020-08-31 오후 3:02	파일 폴더
Adafruit-Motor-Shield-library-master	2020-10-05 오후 5:50	파일 폴더
Bridge	2020-08-31 오후 3:02	파일 폴더
Esplora	2020-08-31 오후 3:02	파일 폴더
Ethernet	2020-08-31 오후 3:02	파일 폴더
Firmata	2020-08-31 오후 3:02	파일 폴더
GSM	2020-08-31 오후 3:02	파일 폴더
Keyboard	2020-08-31 오후 3:02	파일 폴더
LiquidCrystal	2020-08-31 오후 3:02	파일 폴더
Mouse	2020-08-31 오후 3:02	파일 폴더
Robot_Control	2020-08-31 오후 3:02	파일 폴더
Robot_Motor	2020-08-31 오후 3:02	파일 폴더
RobotIRremote	2020-08-31 오후 3:02	파일 폴더

# Adafruit 모터쉴드 라이브러리

```
void loop() {
    uint8_t i;

    #include <AFMotor.h>

    AF_DCMotor motor1(1);
    AF_DCMotor motor2(2);

    void setup() {
        // turn on motor
        motor1.setSpeed(200);
        motor2.setSpeed(200);

        motor1.run(RELEASE);
        motor2.run(RELEASE);
    }

    motor1.run(FORWARD);
    motor2.run(FORWARD);
    for (i=0; i<255; i++) {
        motor1.setSpeed(i);
        motor2.setSpeed(i);
        delay(10);
    }
    motor1.run(BACKWARD);
    motor2.run(BACKWARD);
    for (i=0; i<255; i++) {
        motor1.setSpeed(i);
        motor2.setSpeed(i);
        delay(10);
    }
    motor1.run(RELEASE);
    motor2.run(RELEASE);
}
```

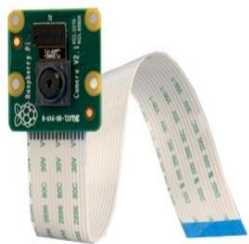
# 자율주행자동차 제작



# 자율주행자동차 제작

- 라즈베리파이(Rpi)와 아두이노를 USB 케이블로 연결
- 건전지를 모터실드와 연결, 모터실드와 아두이노 전원 공급
- Rpi는 보조 배터리로 전원 공급

파이카메라



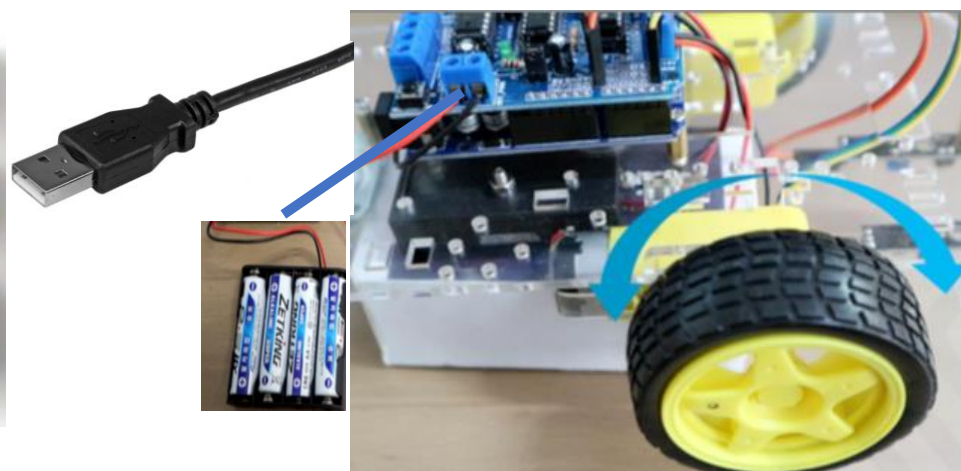
라즈베리파이(Rpi)  
보드



보조배터리



아두이노, 모터 실드와  
모형차





# 하드웨어 작업 분담

- 라즈베리파이(Rpi)는 카메라로 사진을 찍어 방향과 속도 결정
- Rpi는 USB 케이블로 아두이노에게 방향과 속도를 전달
  - "R10", "L20"과 같이 좌우 그리고 속도를 전달
  - 또는 직진과 같은 명령을 정의해서 사용할 것
- 아두이노는 명령 문자열을 받아 모터를 제어



# Rpi와 아두이노 간 USB 통신

- 강의록 게시판에서 Serial.ino를 다운로드
- (아두이노+모터쉴드)를 호스트 PC와 USB로 연결
  - Rpi와 연결이 끊어짐
- 다운 받은 Serial.ino를 스케치에서 컴파일하여 업로드, 아두이노가 Serial.ino를 실행하도록 함
- (아두이노+모터쉴드)에 배터리 연결, 전원이 끊기지 않도록 함
- PC에 연결된 USB를 빼서 Rpi에 연결
  - 아두이노와 Rpi가 USB를 통해 통신 가능
- 강의록 게시판에서 serial\_test.py를 다운로드
- Rpi에서 serial\_test.py를 실행



# Serial\_test.py

```
import serial
ser = serial.Serial('/dev/ttyUSB0',9600)
if ser is None:
    ser = serial.Serial('/dev/ttyUSB1',9600)
while True:
    cmd = ("R%d\n" % 30).encode('ascii')
    print("My cmd is %s" % cmd)
    ser.write(cmd)
    # For debugging, read cmd from arduino
    read_serial=ser.readline()
    print (read_serial)
```



# Serial.ino

```
String Speed;
char LorR;
int i, s;

byte DataToRead[6];

void setup() {
  Serial.begin(9600);
}

void loop() {
  DataToRead[5] = '\n';
  Serial.readBytesUntil(char(13), DataToRead, 5);

  /* For Debugging, send string to RPi */
  for (i = 0; i < 6; i++) {
    Serial.write(DataToRead[i]);
    if (DataToRead[i] == '\n') break;
  }
  /* End of Debugging */
```

```
LorR = DataToRead[0];
Speed = "";
for (i = 1; (DataToRead[i] != '\n')
&& (i < 6); i++) {
  Speed += DataToRead[i];
}
s = Speed.toInt();
if (LorR == 'L') {
  // Turn left wheel with spd s
}
else if (LorR == 'R') {
  // Turn right wheel with spd s
}
delay(2000);
}
```



# Serial\_test.py 변경 사항

- 라즈베리파이 OS 업그레이드 이후 /dev/ttyUSB0 이름이 변경됨
- USB 이름 찾기
  - 라즈베리파이와 아두이노 보드를 USB로 연결 => /dev/serial 디렉토리가 나타남
  - /dev/serial/by-id/ 디렉토리에 USB 인터페이스 이름이 보임
  - 만약 이름이 usb-Arduino\_Srl\_Arduino\_Uno\_75439333235351019202-if00 라면 serial\_test.py를 다음 슬라이드와 같이 수정



# Serial\_test.py 변경 사항

```
import serial
ser = serial.Serial('/dev/serial/by-id/usb-Arduino_Srl_Arduino_Uno_75439333235351019202-if00',9600)
speed = 0
while True:
    cmd = ("R%dWn" % speed).encode('ascii')
    print("My cmd is %s" % cmd)
    ser.write(cmd)
# For debugging, read cmd from arduino
    read_serial=ser.readline()
    print (read_serial)
    speed = speed+1
```

