

=== 선언부 ===

- └ include SoftwareSerial (꼭 선언해야하는지 Test필요)
- └ 초음파센서 핀 설정
- └ State답을 전역변수 선언 (int 형? String형?, Serial을 통해 NodeMcu에서 넘어온 값)

=== Setup ===

- └ 초음파 센서 pinMode설정
- └ 모터 바퀴 pinMode 설정
  - └ 바퀴 두쪽
- └ Serial.begin(9600);

=== Loop ===

- └ if(Serial.available() > 0) 일 경우! 값을 읽고있다가 (읽은값을 state에 저장해둔 상태)
  - └ switch(state)문으로 상태 제어 (
    - └ case on:
      - powerOn();
    - └ case off:
      - stop();
    - └ case auto:
      - autoDrive();
    - └ case manual:
      - manualDrive();

=== 함수부 ===

- └ powerOn();
  - └ led를 켜다.(물리적으로 전기 공급을 중단할 수가 없음.대용으로 led사용)
- └ stop();
  - └ led를 끄고, 아두이노 움직임 정지
- └ autoDrive();
  - └ 자동주행 시작
- └ manualDrive();
  - └ 전(g), 후(b), 좌(l), 우(r) 로 동작하도록 구현  
(방향을 정의한 변수는 여기서만 사용)

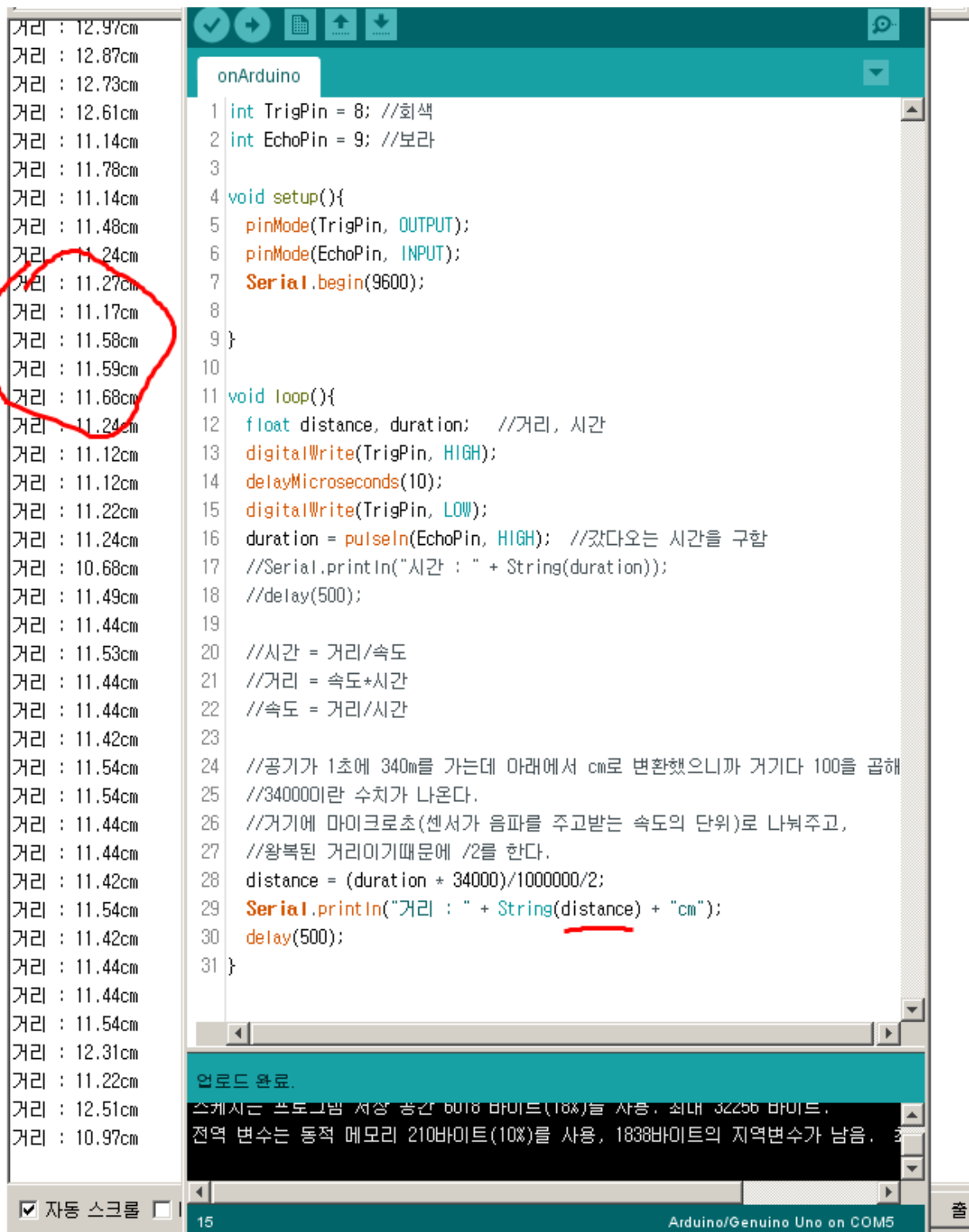
=== 보류 ===

- └ 간식급여, 멜로디재생, 압력센서, 투석기 만드는거 보류함

## 1. 초음파 센서 달기

- 초음파 센서 연결하기  
센서의 Trig핀이 아두이노의 8번으로 Echo핀이 9번핀으로 갔다.  
-> Trig핀 12번, Echo핀이 13번으로 수정
- 센서값 시리얼로 확인하기  
갔다가 돌아오는 값을 기준으로 거리를 측정한다.  
Serial을 보면 String형태로 distance가 변환되어 출력된다.

c. 이 값은 이후에 자동주행 부분에서 사용할 예정!!

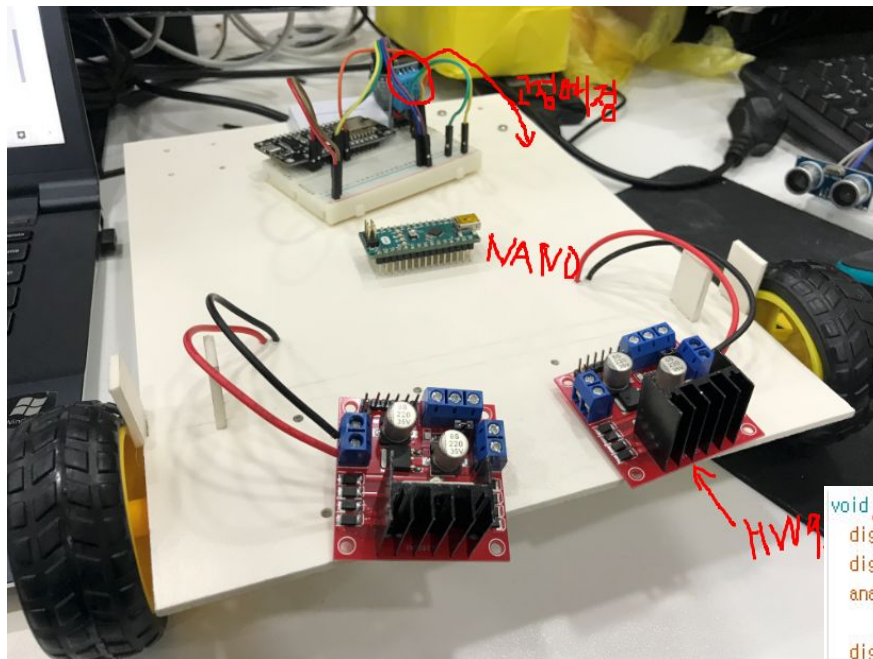


## 2. 1.에서 구한 distance값으로 자동주행

- a. 자동차 밀 판 만들기 ok
  - 추가로 구매해야 되는 나사

- HW95 고정용 나사 볼트 너트 총 8쌍
- 온습도계 고정용 볼트 너트 1쌍
- 앞 왼쪽 바퀴 나사 8개 (하나도 고정 안되었음)
- 앞 오른쪽 바퀴 나사 2개 (대각선으로 2쪽만 연결되었음)
- 볼트 너트는 구매하지 못함
  - >>> 철사를 사서 구매했다. 철물점을 3~4군데 다녀봤는데 이렇게 조그만건 팔지 않는다고 인터넷에 알아보려고 했지만 철사로 하면 될거같아서 구매해서 해보니 다행히 고정은 잘 됐다.

b. 일단 Go(전진), Back(후진), Stop(정지) 함수부터 구현했다.



```
void Go(){
  digitalWrite(8, 1);
  digitalWrite(7, 0);
  analogWrite(11, 255);

  digitalWrite(6, 1);
  digitalWrite(5, 0);
  analogWrite(3, 255);
}

void Back(){
  digitalWrite(8, 0);
  digitalWrite(7, 1);
  analogWrite(11, 255);

  digitalWrite(6, 0);
  digitalWrite(5, 1);
  analogWrite(3, 255);
}

void Stop(){
  digitalWrite(8, 0);
  digitalWrite(7, 0);

  digitalWrite(6, 0);
  digitalWrite(5, 0);
}
```

```

void loop(){
  if(Serial.available()){
    Mode = Serial.read();
    switch(Mode){
      case '1':
        Go();
        break;
      case '2':
        Back();
        break;
      case '3':
        Stop();
        break;
    }
  }
}

```

--> 그리고 루프에서 1~3의 값을 입력해 제대로 작동되는지 Test한 결과 제대로 작동하였다.  
(제대로 작동하는지 확인OK)

※핀 설정 참고

```

1 #include <SoftwareSerial.h>
2 //왼쪽 바퀴
3 int Lin1 = 8;
4 int Lin2 = 7;
5 int LENA = 11;
6 //오른쪽 바퀴
7 int Rin1 = 6;
8 int Rin2 = 5;
9 int RENA = 3;
0
1 char Mode;
2

```

- c. 이제 초음파 센서로 구한 거리를 이용하여 자동주행을 구현해야 한다.
  - i. 일단 두개의 함수를 추가했다.
    1. AutoDrive()
    2. TurnLeft()

```

void AutoDrive(){
    driving = true;
    while(driving){
        if(driving = false){
            Stop();
        }
        digitalWrite(TrigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(TrigPin, LOW);

        duration = pulseIn(EchoPin, HIGH); //갓다오는 시간을 구함
        distance = (duration * 34000)/1000000/2;
        Serial.println("거리 : " + String(distance) + "cm");
        delay(500);

        if (distance >= 20) {
            Go();
        } else if (distance < 20) {
            Back();
            delay(3000);
            TurnLeft();
            delay(3000);
        }
    }
}
}

```

```

void TurnLeft(){
    driving = true;
    while(driving){
        if(driving = false){
            Stop();
        }
        //왼쪽바퀴는 정지한 상태에서 오른쪽바퀴만 돌면됨
        digitalWrite(8, 0);
        digitalWrite(7, 0);

        digitalWrite(6, 1);
        digitalWrite(5, 0);
        analogWrite(3, 255);
    }
}
}

```

그리고 전역변수로 값을 계속 저장해 놓기 위해서 distance와 duration 을 선언해놓고 AutoDrive()는 Loop()안에서 계속 거리를 측정하며 자동주행을 시작한다.

- ! 근데 자동주행은 되는데 가만히 보니 초음파 센서가 정면 가운데 있어서 앞쪽 양옆 모서리쪽의 정면은 감지를 잘 못하는거 같다. 일단 집에서 테스트했기 때문에 학원같은 넓은 환경에서는 어떻게 작동할지도 테스트 해봐야 할듯!

To-Do ▶ 기존에 작은 RC카와 달리 전면의 면적도 크기 때문에 초음파 센서 하나로 그 면적을 모두 감지해서 자동주행을 하는건 불가능하다. 앞부분에 총 3개정도를 달게 된다면 구현이 가능할 같은데 이 부분을 작성하고있는 지금, 그 부분보다는 팀원들과 회의를 한 뒤 WebSocket부분을 구현하는것이 낫다고 생각이 들었다.

### 3. NodeMcu에 찍히는 씨리얼 읽기( 공유, Master(NodeMcu) - Slave(Arduino) )

**읽은 후에는 아두이노도 동작해야됨!**

## -아두이노 우노 & NodeMcu Lua Master-Slave 연결하기

참고 영상 <https://www.youtube.com/watch?v=SiU-QZwik8w>

## 참고 영상

<https://www.hackster.io/pawan-kumar3/serial-communication-between-nodemcu-and-arduino-640819>

a. 일단 핀 설정

## i.

Arduino	NodeMcu
TX	RX
RX	TX

찾아본 자료에서는 GND까지 서로 연결시켜줬지만 다른곳에서 나가서 그런지 GND를 연결하지 않았지만 제대로통신이 가능했다.

ii. Node에 찍힌 값이 아두이노에게 가는 것이기 때문에 다른 함수를 일단

주석처리하고 loop()에 값을 찍어봤다.

```
132 void loop() {
133 // receiveWiFiData();
134 //
135 // while(WiFi.status() == WL_CONNECTED){
136 //   receiveMessageFromServer();
137 //   checkTemp();
138 // }
139 Serial.println("node -> arduino Serial Test");
140 delay(1000);
141 }
```

[illegible]

- iii. 값은 잘 찍히고 아두이노쪽도 동일한 시리얼이 나오는지 체크해 보았다.  
(아두이노 홈페이지에서 제공하는 온라인 툴을 이용)

```
void loop(){
// if(Serial.available()){
//   Mode = Serial.read();
//   switch(Mode){
//     case '1':
//       Go();
//       break;
//     case '2':
//       Back();
//       break;
//     case '3':
//       Stop();
//       break;
//     case '4':
//       AutoDrive();
//       break;
//   }
// }
// AutoDrive();
if (Serial.available()) {
  Serial.write(Serial.read());
}
}
```

```
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
node -> arduino Seiral Test
```

- iv. 일단 값을 동일하게 들어온다.

V. nodeMcu측에서 와이파이가 연결된 상태라면 receiveMessageFromServer()에서 GET으로 0~4(mode)를 가지고 오는데 그값이 node쪽 serial엔 계속 출력된다.

그 값이 arduino에서도 계속 찍히게 된다면 arduino에선 그값을 읽고 그에 맞는 동작만 수행하면 됨.

VI. 일단 노드에 찍힌 데이터(아두이노 상태값)은 노드에서 제대로 찍힌뒤 아두이노까지 Serial 통신으로 전달되어 아두이노 port의 serial에서도 동일한 값이 찍히는 것을 확인하였다. OK  
(1~4까지의 데이터는 초기에 char형으로 선언해 놓았고, 읽는 부분(Arduino)에서는 단순히 Serial.abaliavle()을 이용해서 그 값을 읽어들이고 switch(Mode)에서 그 값을가지고 작동하면 된다.

-> 소스를 보면 알겠지만 정상적으로 해당 uri에서 값을 읽어와서 Mode라는 변수에 저장된다.

그 값을 가지고 switch문을 돌리면 정상적으로 동작하는 것을 확인 할 수 있었다.

