자율주행 자동차 프로젝트 가이드

**프로젝트 가이드**

Auto Pilot Car Project Guide

제작 : 손계원

ver. 1.5

목차

[자율주행 자동차 프로젝트 가이드 1](#_Toc490079810)

[Auto Pilot Car Project Guide 1](#_Toc490079811)

[1. 공통 사항 4](#_Toc490079812)

[1. 방향 관련 4](#_Toc490079813)

[2. 변수 및 함수 이름 관련 4](#_Toc490079814)

[3. 변수 선언 4](#_Toc490079815)

[2. Servo Motor 5](#_Toc490079816)

[1. 변수 선언 5](#_Toc490079817)

[3. DC Motor 5](#_Toc490079818)

[1. 변수선언 5](#_Toc490079819)

[2. 구동 관련 5](#_Toc490079820)

[4. Bluetooth 6](#_Toc490079821)

[1. 통신핀 6](#_Toc490079822)

[2. 시리얼 이름 6](#_Toc490079823)

[5. 초음파 6](#_Toc490079824)

[1. 변수명 6](#_Toc490079825)

[6. 적외선 센서 6](#_Toc490079826)

[1. 변수명 6](#_Toc490079827)

[7. 기타 센서 6](#_Toc490079828)

[8. 함수 7](#_Toc490079829)

[1. 반복적인 기능, 기본 기능은 모두 함수로 만듦 7](#_Toc490079830)

[2. 위치 7](#_Toc490079831)

[3. 함수, 제어문 및 반복문 사용시 7](#_Toc490079832)

[4. Servo 함수 8](#_Toc490079833)

[5. 초음파 함수 8](#_Toc490079834)

[6. Motor 함수 9](#_Toc490079835)

[1. 속도 9](#_Toc490079836)

[2. 기본 기능 9](#_Toc490079837)

[부록 11](#_Toc490079838)

[1. Bluetooth 기본 설정 12](#_Toc490079839)

[2. 시리얼 데이터 저장 16](#_Toc490079840)

[A. 시리얼 모니터 값 TXT 파일 만들기 16](#_Toc490079841)

# 공통 사항

## 방향 관련

* + 1. 정면에 센서 사용시 Left, Center, Right를 사용
       1. L, C, R로 앞 글자만 대문자로 사용
    2. 전 방향에 사용시 Front, Rear, Left Side, Right Side
       1. Front, Rear로 사용
       2. L\_Side, R\_Side
    3. 방향을 앞에 표시
       1. Ex) L\_Motor, L\_Side\_Echo

## 변수 및 함수 이름 관련

* + 1. a,b,c, 또는 x,y와 같이 명확하지 않은 이름 지양
    2. 단어간 연결, 방향 관련은 \_(Underline)을 사용
    3. 단어의 첫 글자는 대문자 사용
    4. 변수 선언 시 무엇에 관련된 변수 인지 주석 달기
    5. 함수 이름은 하나의 단어로 구성될 경우 소문자로만 작성
    6. set, get 등으로 시작하는 함수에는 \_없이 사용

## 변수 선언

* + 1. 변수 선언 시 같은 묶음 단위 변수 선언 후 공백
    2. 3글자 이하인 센서는 모두 대문자로 선언

# Servo Motor

## 변수 선언

* + 1. 서보 모터
       1. 방향 제어 서보 모터
          1. Steer\_Servo
       2. 기타 서보
          1. (기능명 또는 센서 이름)\_Servo
          2. Ex) IR\_Servo(적외선 센서와 관련 있는 서보)[[1]](#footnote-1)

# DC Motor

## 변수선언

* + 1. L\_Motor, R\_Motor, C\_Motor[[2]](#footnote-2)

## 구동 관련

* + 1. Forward, Backward, Velocity[[3]](#footnote-3), Stop 을 사용
    2. 각각의 앞 글자만 따서 F, B, V, S로 사용
       1. L\_Motor\_F //좌측 모터 전진
       2. R\_Motor\_B //우측 모터 후진

# Bluetooth[[4]](#footnote-4)

## 통신핀

* + 1. TX, RX

## 시리얼 이름

* + 1. BT

# 초음파

## 변수명

* + 1. ECHO, TRIG 대문자로 사용
    2. 2개 이상 사용시 1.1과 1.2의 방향 관련 규칙에 따라 센서 위치에 따라 접두사 사용

# 적외선 센서

## 변수명

* + 1. IR\_Sensor
    2. 2개 이상 사용시 1.1의 방향 관련 규칙에 따라 센서 위치에 따라 접두사 사용

# 기타 센서

* 1. 센서이름\_센서

# 함수

## 반복적인 기능, 기본 기능은 모두 함수로 만듦

* + 1. forward(), backward(), turn\_Left(), turn\_Right(), setSpeed()[[5]](#footnote-5), getDistamce()[[6]](#footnote-6)등등

## 위치

* + 1. setup과 loop사이를 기본으로 하되, loop이후 정의 시 set이후 선언 후 loop문 이후 정의

## 함수, 제어문 및 반복문 사용시

* + 1. 수정 용이성과 { } 가독성을 위해 BSD 또는 GNU 방법으로 코딩

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6 | if(...)      {          처리();      }    GNU | [cs](http://colorscripter.com/info#e) |

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6 | if(...)  {          처리();  }    BSD | [cs](http://colorscripter.com/info#e) |

## Servo 함수[[7]](#footnote-7)

* + 1. turn\_Right : 우회전
       1. turn\_Right\_R : 후진 우회전
    2. turn\_Left : 좌회전
       1. turn\_Left\_R : 후진 좌회전
    3. turn\_U
       1. U 턴

## 초음파 함수

* + 1. 거리 구하기[[8]](#footnote-8)

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9 | float / int getDistance(char TRIG, char ECHO)  {  digitalWrite(TRIG,HIGH);  delay(10);  digitalWrite(TRIG,LOW);  float / int  distance = pulseIn(ECHO,HIGH)/58.2;  return distance;  } | [cs](http://colorscripter.com/info#e) |

## Motor 함수[[9]](#footnote-9)

### 속도

|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5 | void setSpeed(char speed)  {      analogWrite(R\_Motor\_V, speed);      analogWrite(L\_Motor\_V, speed);  } | [cs](http://colorscripter.com/info#e) |

### 기본 기능[[10]](#footnote-10)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1  2  3  4  5  6  7  8 | void forward()  {      Steer\_Servo.write(정면 각도);      digitalWrite(L\_Motor\_F, HIGH) ;      digitalWrite(L\_Motor\_B, LOW);      digitalWrite(R\_Motor\_F, HIGH) ;      digitalWrite(R\_Motor\_B, LOW);  } | | [cs](http://colorscripter.com/info#e) | |
| 1  2  3  4  5  6  7  8 | void backward()  {      Steer\_Servo.write(정면 각도);      digitalWrite(L\_Motor\_F, LOW) ;      digitalWrite(L\_Motor\_B, HIGH);      digitalWrite(R\_Motor\_F, LOW);      digitalWrite(R\_Motor\_B, HIGH);  } |  | |

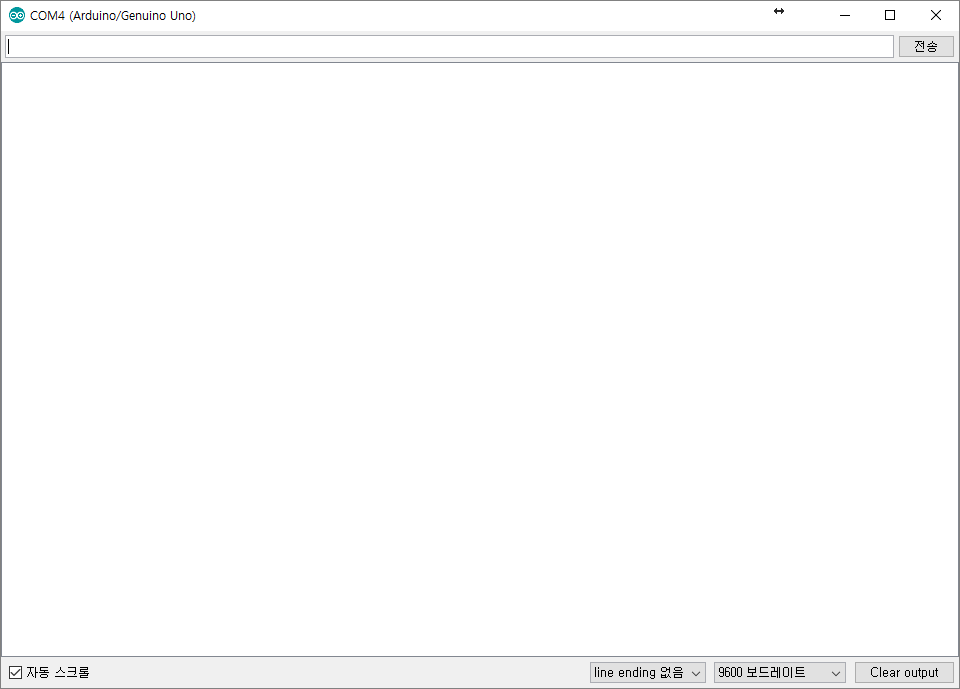
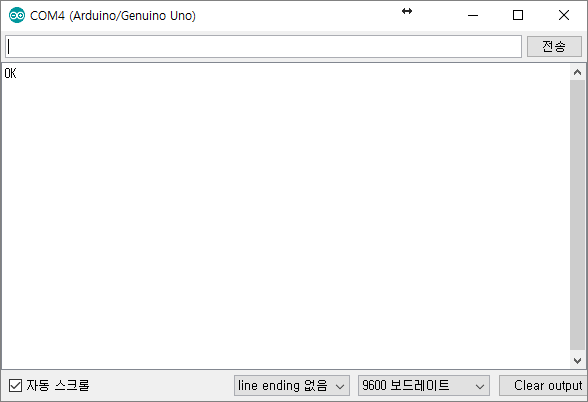
|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8 | void stop(int time)  {      digitalWrite(L\_Motor\_F, HIGH) ;      digitalWrite(L\_Motor\_B, HIGH);      digitalWrite(R\_Motor\_F, HIGH);      digitalWrite(R\_Motor\_B, HIGH);     delay(time);  } |  |

# 부록

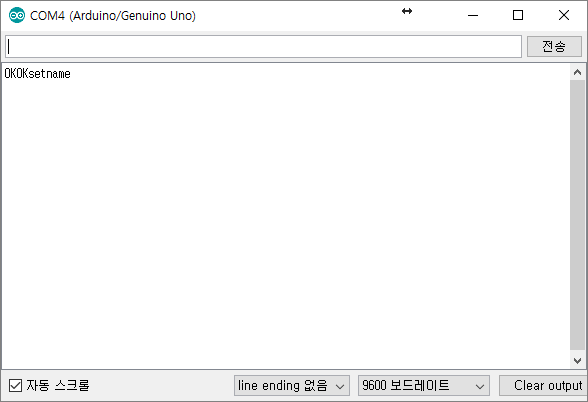
# Bluetooth 기본 설정

* 1. 이름 및 암호 변경
     1. 설정 프로그램 업로드[[11]](#footnote-11)

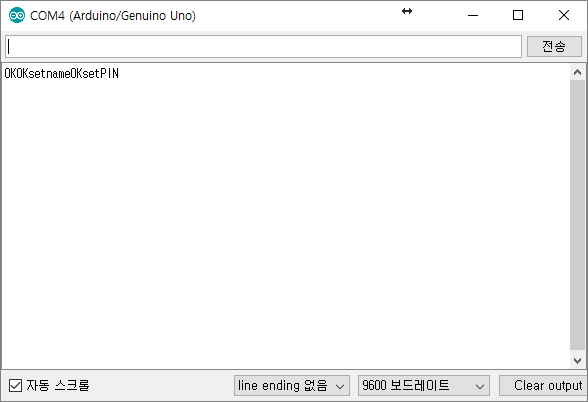
|  |  |  |
| --- | --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | #include <SoftwareSerial.h>  #define TX pinNo // 자유롭게 사용  #define RX pinNo  SoftwareSerial BTSerial(RX, TX);    void setup()  {    Serial.begin(9600);    BTSerial.begin(9600);  }  void loop()  {    if (BTSerial.available())     Serial.write(BTSerial.read());      if (Serial.available())     BTSerial.write(Serial.read());  } | [cs](http://colorscripter.com/info#e) |

* + 1. 기본 설정
       1. 시리얼 모니터를 실행
       2. 블루투스 현재 상태확인

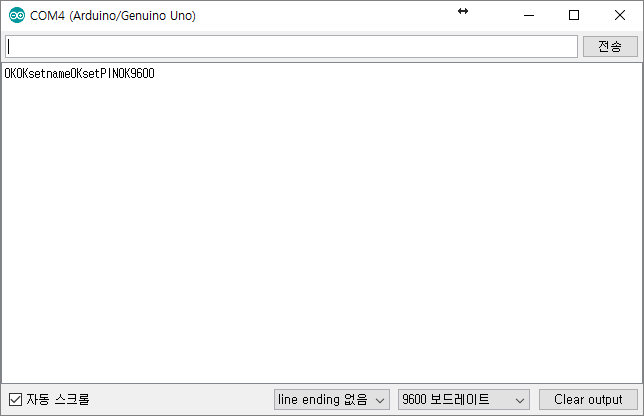
OK가 출력되면 응답 확인

* + - 1. 이름 변경

AT+NAME변경할이름

* + - 1. PIN 번호 변경

AT+PIN변경할핀

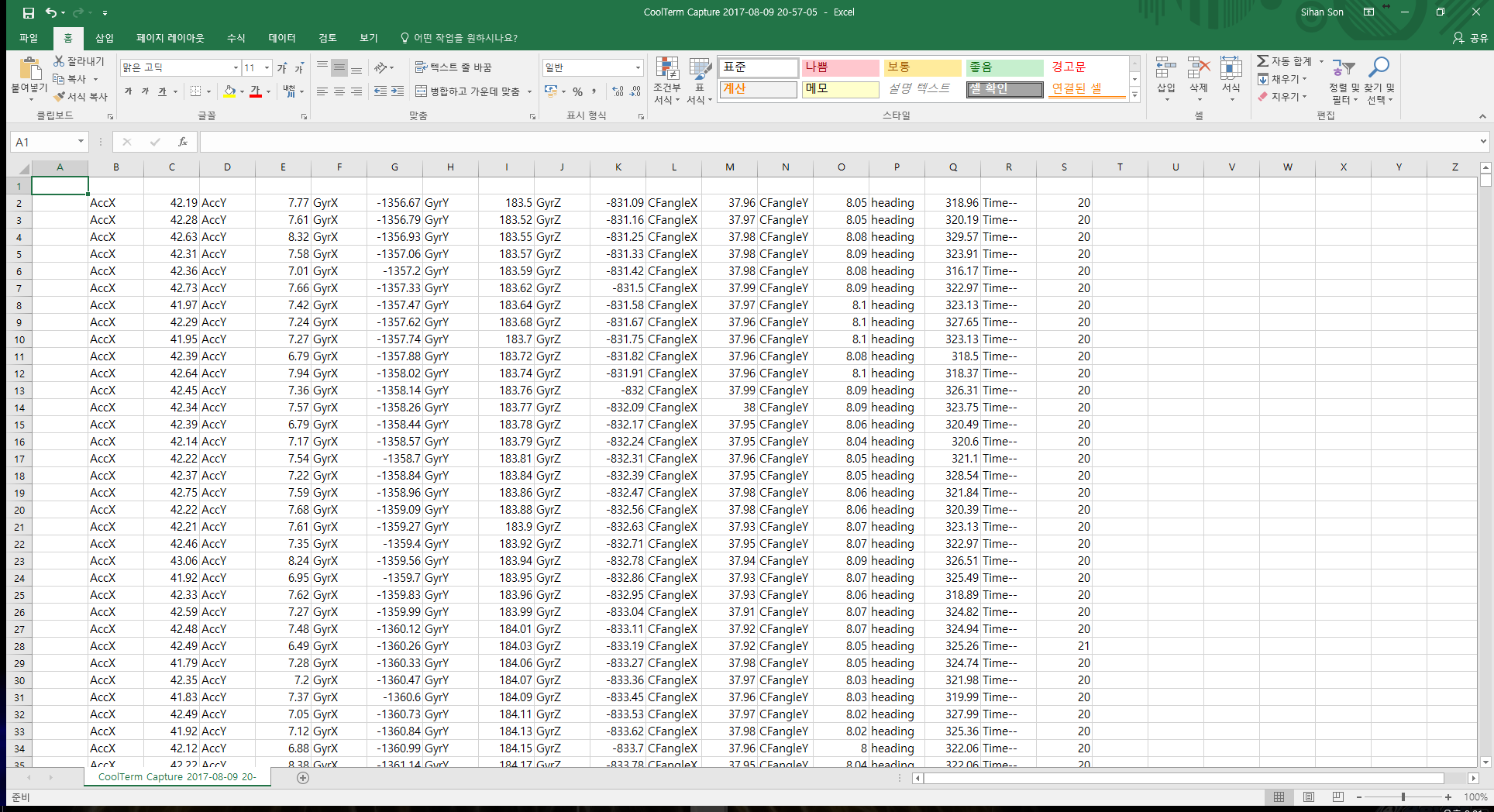
* + - 1. 통신 속도 변경

AT+BAUD통신속도(1~8)[[12]](#footnote-12)

# 시리얼 데이터 저장

## 시리얼 모니터 값 TXT 파일 만들기

* + 1. Util 폴더의 CoolTerm 또는 공식홈페이지인 <http://freeware.the-meiers.org/> 에서 프로그램 다운로드
    2. Connection -> Options -> Serial port에서 포트 설정
       1. port에서 아두이노가 연결된 포트로 설정
          1. 블루투스 연결시 장치관리자에서 수신 포트 확인
          2. 아두이노가 연결된 포트가 뜨지 않을 시 하단의 Re-Scan 버튼을 클릭
       2. Baudrate에서 통신속도를 맞춤
    3. 상단 메뉴바에서 Connect 클릭
       1. 아두이노에서 시리얼 모니터를 열고 있으면 Error 발생, 다른 시리얼 모니터 종료 후 클릭
    4. Connection -> Capture to Textfile -> Start(Ctrl + R, 단축키)로 텍스트 파일로 시리얼 모니터의 값들이 저장,
       1. 충분한 데이터가 모였으면 정지
    5. Disconnect로 시리얼 모니터 값을 읽어오는 것을 중지
    6. TXT 파일을 엑셀에서 열어 구분자들로 CSV파일 제작



△ CSV 파일로 만든 예시, 자이로 센서 값

원본 및 업데이트는 <https://github.com/Sihan-Son/AutoPilotCar_rasp>에서 확인하세요

1. 초음파는 UW를 사용 [↑](#footnote-ref-1)
2. 모터 하나로 두 바퀴를 제어 할 경우 [↑](#footnote-ref-2)
3. Speed, Stop 모두 S로 시작해 혼동을 막기 위해 변수 이름엔 Velocity 사용 [↑](#footnote-ref-3)
4. 부록 참고 [↑](#footnote-ref-4)
5. 모터 구동 속도 설정 [↑](#footnote-ref-5)
6. 초음파 센서 사용시 거리 구하는 함수 [↑](#footnote-ref-6)
7. 구제적인 구현 방법은 개인에 맡김 git 참조 [↑](#footnote-ref-7)
8. 프로그램 내에서 거리를 사용하는 변수는 1.1과 1.2의 규칙에 따라 방향\_Dis를 사용한다. [↑](#footnote-ref-8)
9. 2개의 모터를 개별 제어하는 것을 전제로 함 [↑](#footnote-ref-9)
10. 방향전환은 8.3참조 [↑](#footnote-ref-10)
11. 부록의 A 파일 [↑](#footnote-ref-11)
12. 1 -> 1200   2 -> 2400    3 -> 4800    4 -> 9600(기본값)    5 ->19200    6 -> 38400    7 -> 57600    8 -> 115200

    다른 장치에 연결 시 통신속도를 맞춰 주어야 한다 [↑](#footnote-ref-12)