

[아두이노 스마트 자동차]

목 차

1. 스마트 자동차 전진
2. 여러 가지 동작(바퀴제어)
3. 스마트 자동차 연결동작
4. 스마트 자동차 스위치 누른 후 부저 경보
5. 스마트 자동차 블랙 라인 추적
6. 스마트 자동차 적외선 장애물 회피 1(기본 장애물)
7. 스마트 자동차 적외선 장애물 회피 2(물체로 향함)
8. (역 회전 방지와) 스마트 자동차 적외선 장애물 회피 테스트(3)
9. 스마트 자동차 초음파 실험
10. 스마트 자동차 초음파 장애물 회피 실험(NO Servo)
11. 스마트 자동차 초음파 장애물 회피 테스트(Servo)
12. 적외선 원격 제어 스마트 자동차
13. 스마트 자동차 블랙라인 추적 및 초음파 장애물 회피
14. 기본적인 Bluetooth 연결(1)-시리얼모니터 출력
15. 기본적인 Bluetooth 연결(2)-스마트 폰 출력
16. Bluetooth를 이용한 스마트 자동차 컨트롤





개 요

- ⇒ 스마트 자동차의 모터를 제어하면서 여러 가지 동작을 구현해본다.
- ⇒ 스위치와 부저를 이용한 제어를 해본다.
- ⇒ 적외선 센서를 이용한 장애물 회피동작을 구현해본다.
- ⇒ 초음파 센서를 이용한 장애물 회피동작을 구현해본다.
- ⇒ 적외선 센서를 이용해 자동차를 원격제어 해본다.
- ⇒ Bluetooth를 이용해 자동차를 제어해본다.

부품목록

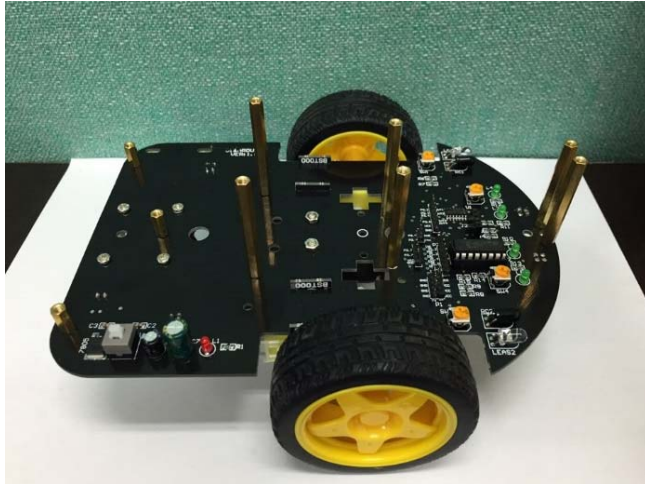
NO	부품명	수량	상세설명
1	아두이노 우노	1	아두이노 호환보드
2	건전지	1	9V 배터리
3	초음파센서	1	HC-SR04
4	점퍼케이블	2(묶음)	점퍼케이블(암-수)
5	9V 배터리 연결 배럴 잭	1	배럴 잭
6	자동차 키트(부품키트 포함)	1	자동차 키트
7	서보모터	1	서보모터
8	Battery universal charger	2	건전지 충전기
9	적외선센서 & 리모콘	1	적외선센서 & 리모콘
10	Bluetooth 모듈	1	HC-06

부품명	아두이노 우노	건전지	9v 배터리 배럴 잭	자동차 키트
파트				
	x1	x2	x1	x1

부품명	초음파센서	서보모터	점퍼케이블	Battery universal charger
파트				
	x1	x2	x1	x1

하드웨어 Making

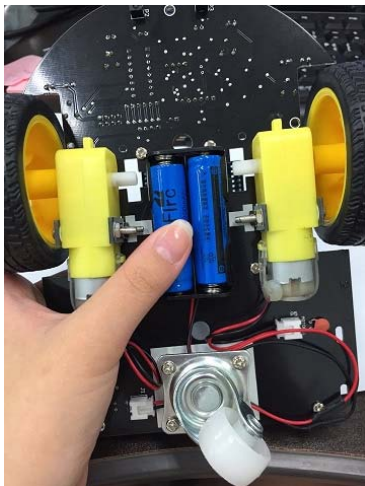
■기본 자동차 키트



1. 아두이노를 장착해줍니다.



2. 자동차 아랫부분에 건전지를 끼워줍니다.



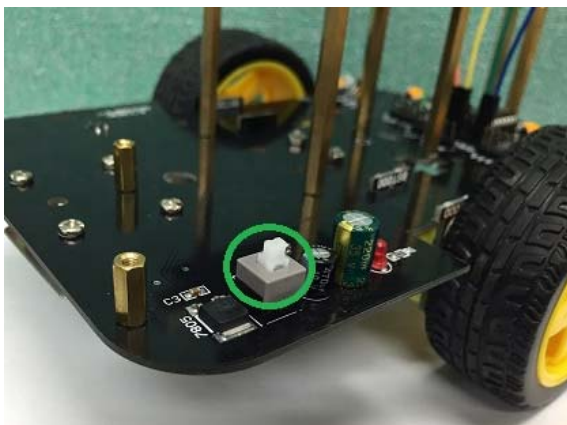
01_스마트 자동차 전진(advance)

①모터와 아두이노 연결

- UNO의 D8 & 자동차키트의 IN1 P1.2
- UNO의 D9 & 자동차키트의 IN2 P1.3
- UNO의 D10 & 자동차키트의 IN3 P1.6
- UNO의 D11 & 자동차키트의 IN4 P1.7



* 아래의 스위치를 눌러야만 동작한다.



②소스코드

01_스마트 자동차 전진

```
//#include <Servo.h>
int Left_motor_back=8;    //Left_motor_back(IN1)
int Left_motor_go=9;      //Left_motor_go(IN2)
int Right_motor_go=10;     //Right_motor_go(IN3)
int Right_motor_back=11;   //Right_motor_back(IN4)

void setup()
{
    pinMode(Left_motor_go,OUTPUT);
    pinMode(Left_motor_back,OUTPUT);
    pinMode(Right_motor_go,OUTPUT);
    pinMode(Right_motor_back,OUTPUT);
}
void run(int time)        //전진
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //PWM은 0~255속도조절, 좌우 차이가 증감
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200); //PWM은 0~255속도조절, 좌우 차이가 증감
    analogWrite(Left_motor_back,0);
    delay(time * 100);
}

void loop()
{
    delay(500);
    run(10); //전진
}
```

02_여러 가지 동작(바퀴제어)

① 연결은 [01_스마트 자동차 전진]과 동일

② 소스코드

02_여러 가지 동작(바퀴제어)

```
void setup()
void run(int time) //전진
//여기까지 [01]과 동일

void brake(int time) 브레이크
{
    digitalWrite(Right_motor_go,LOW); //Right, Left 모두 LOW
    digitalWrite(Right_motor_back,LOW);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    delay(time * 100);
}

void left(int time) //좌회전(오른쪽 바퀴 회전식으로)
{
    digitalWrite(Right_motor_go,HIGH); // Right_motor_go가 HIGH
    digitalWrite(Right_motor_back,LOW); // Right_motor_back이 LOW
    analogWrite(Right_motor_go,200); // Right_motor_go의 속도 200
    analogWrite(Right_motor_back,0); //PWM0~255속도조절
    digitalWrite(Left_motor_go,LOW); //Left는 LOW
    digitalWrite(Left_motor_back,LOW); //Left는 LOW
    analogWrite(Left_motor_go,0); // Left_motor_go의 속도 0
    analogWrite(Left_motor_back,0); //PWM0~255 속도조절
    delay(time * 100);
}

void spin_left(int time) //좌회전(Spin)
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); // Right_motor_go의 속도 200
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,200); // Left_motor_back의 속도 200
    delay(time * 100);
}
```

```

void right(int time)          //우회전(왼쪽 바퀴 회전식으로)
{
    digitalWrite(Right_motor_go,LOW);    // Right_motor_go가 LOW
    digitalWrite(Right_motor_back,LOW);  // Right_motor_back이 LOW
    analogWrite(Right_motor_go,0);        // Right_motor_go의 속도 0
    analogWrite(Right_motor_back,0);      //PWM0~255속도조절
    digitalWrite(Left_motor_go,HIGH);     // Left_motor_go HIGH
    digitalWrite(Left_motor_back,LOW);    // Left_motor_go LOW
    analogWrite(Left_motor_go,200);       // Left_motor_go의 속도 200
    analogWrite(Left_motor_back,0);       //PWM0~255속도조절
    delay(time * 100);
}

void spin_right(int time)      //우회전(Spin)
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,200);    // Right_motor_go의 속도 200
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0);       // Left_motor_back의 속도 0
    delay(time * 100);
}

void back(int time)           //후진
{
    digitalWrite(Right_motor_go,LOW);     //오른쪽 바퀴 뒤로
    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,150);    //속도 150
    digitalWrite(Left_motor_go,LOW);      //왼쪽 바퀴 뒤로
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,150);     //속도 150
    delay(time * 100);
}

void loop()
{
    delay(2000); //2초 대기
    back(10);    //후진한 뒤 1초
    brake(5);    //정지 0.5초
}

```



```
run(10);      //전진 1초  
brake(5);     //정지 0.5초  
left(10);     //좌향 좌 1초  
right(10);    //우향 우 1초  
spin_right(20); //오른쪽으로 회전 2초  
spin_left(20); //왼쪽으로 회전 2초  
brake(5);     //정지 0.5초  
}
```

03_스마트 자동차 연결동작

①연결은 [01_스마트 자동차 전진(advance)]과 동일

②소스코드

03_스마트 자동차 연결동작

```
void setup()
void run(int time)
void brake(int time)
void left(int time)
void spin_left(int time)
void right(int time)
void spin_right(int time)
void back(int time)
//여기까지 [02]와 동일

void loop()
{
    int i; //변수 선언
    delay(2000); //2초 대기
    run(10);      //1초 전진
    back(10);     //1초 후진
    brake(5);     //0.5초 정지

    for(i=0;i<5;i++)
    {
        run(10); //전진 1초
        brake(1); //정지 0.1초
    }

    for(i=0;i<5;i++)
    {
        back(10); //후진1초
        brake(1); //정지 0.1초
    }

    for(i=0;i<5;i++)
    {
        left(10); //좌향 좌 1초
        spin_left(5); //왼쪽으로 회전 0.5초
    }
}
```

```

for(i=0;i<5;i++)
{
    right(10); //우향 우 1초
    spin_right(5); //오른쪽으로 회전 0.5초
}

for(i=0;i<10;i++)
{
    right(1); //우향 우 0.1초
    brake(1); //정지 0.1초
}

for(i=0;i<10;i++)
{
    left(1); //좌향 좌 0.1초
    brake(1); //정지 0.1초
}

for(i=0;i<10;i++)
{
    left(3); //좌향 좌 0.3초
    right(3); //우향 우 0.3초
}

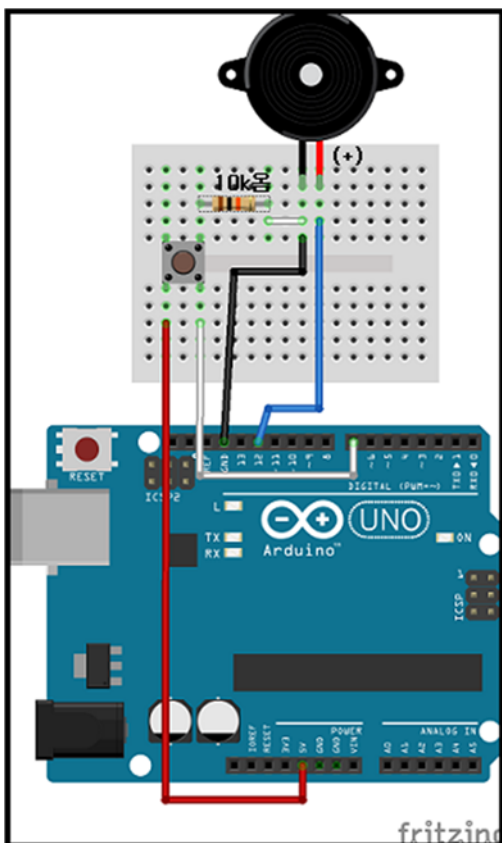
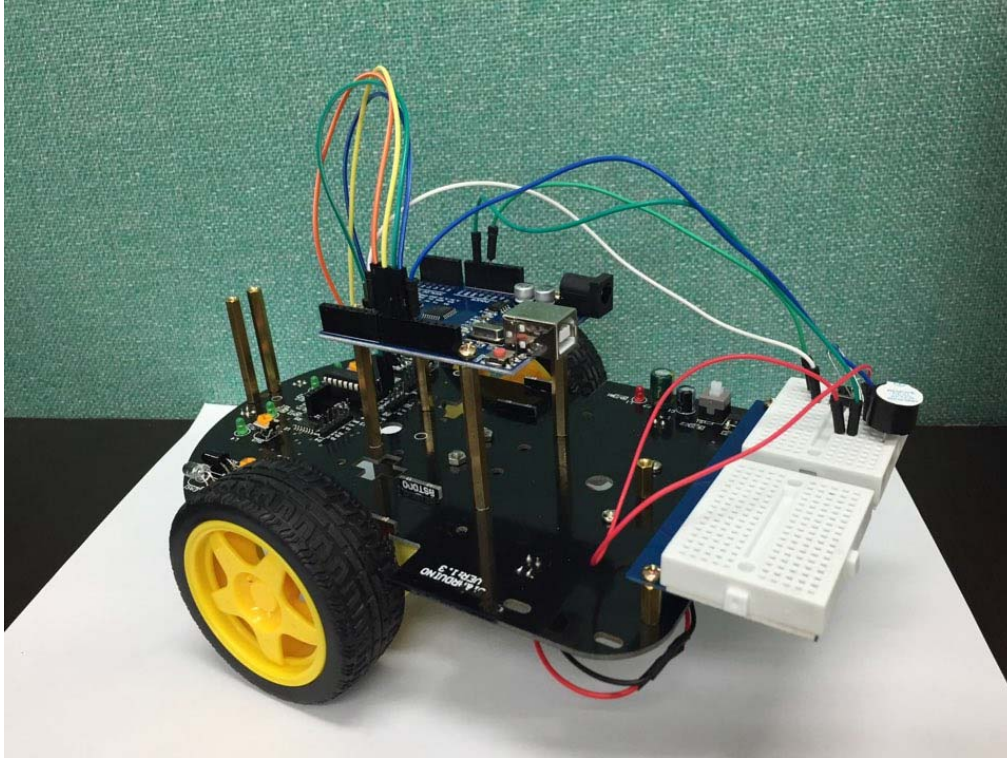
for(i=0;i<10;i++)
{
    spin_left(3); //왼쪽으로 회전 0.3초
    brake(3); //정지 0.3초
}

for(i=0;i<10;i++)
{
    spin_right(3); //오른쪽으로 회전 0.3초
    brake(3); //정지 0.3초
}
}

```

04_스마트 자동차 스위치 누른 후 부저 경보

①스위치&부저 연결



②소스코드

04_스마트 자동차 스위치 누른 후 부저 경보

```
int key=7; //D7(갈색 점퍼선)
int beep=12; //D12(회색 점퍼선)
void setup()
{  pinMode(key,INPUT);
   pinMode(beep,OUTPUT);
}
void run(int time)
void brake(int time)
void left(int time)
void spin_left(int time)
void right(int time)
void spin_right(int time)
void back(int time)
//여기까지 03과 동일
void keysacn()
{
  int val;
  val=digitalRead(key);
  while(!digitalRead(key))
  {

    val=digitalRead(key);
  }
  while(digitalRead(key))
  {
    delay(10);
    val=digitalRead(key);
    if(val==HIGH)
    {
      digitalWrite(beep,HIGH);
      while(!digitalRead(key))
        digitalWrite(beep,LOW);
    }
    else
      digitalWrite(beep,LOW);
  }
}
```

```
}
```

```
void loop()
```

```
{
```

```
  keysacn();
```

```
  back(10);
```

```
  brake(5);
```

```
  run(10);
```

```
  brake(5);
```

```
  left(10);
```

```
  right(10);
```

```
  spin_left(20);
```

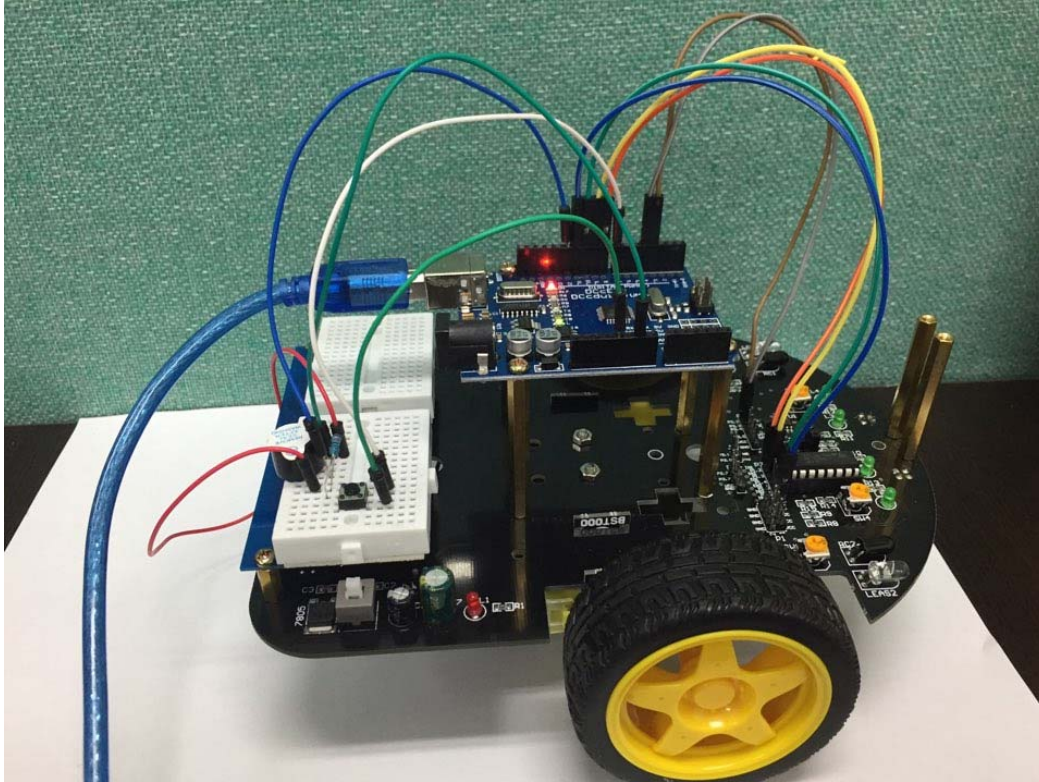
```
  spin_right(20);
```

```
  brake(5);
```

```
}
```

05_스마트 자동차 블랙 라인 추적

①연결



②소스코드

05_스마트 자동차 블랙 라인 추적

```
const int SensorRight = 3;  //(P3.2 OUT1)
const int SensorLeft = 4;   //(P3.3 OUT2)
int SL;    //SensorLeft
int SR;    //SensorRight

void setup()
{
  pinMode(SensorRight, INPUT);
  pinMode(SensorLeft, INPUT); //05_블랙라인 추적
}

void run()
void brake()
void left()
void spin_left(int time)
void right()
void spin_right(int time)
void back(int time)
//여기까지 [03]과 동일
```

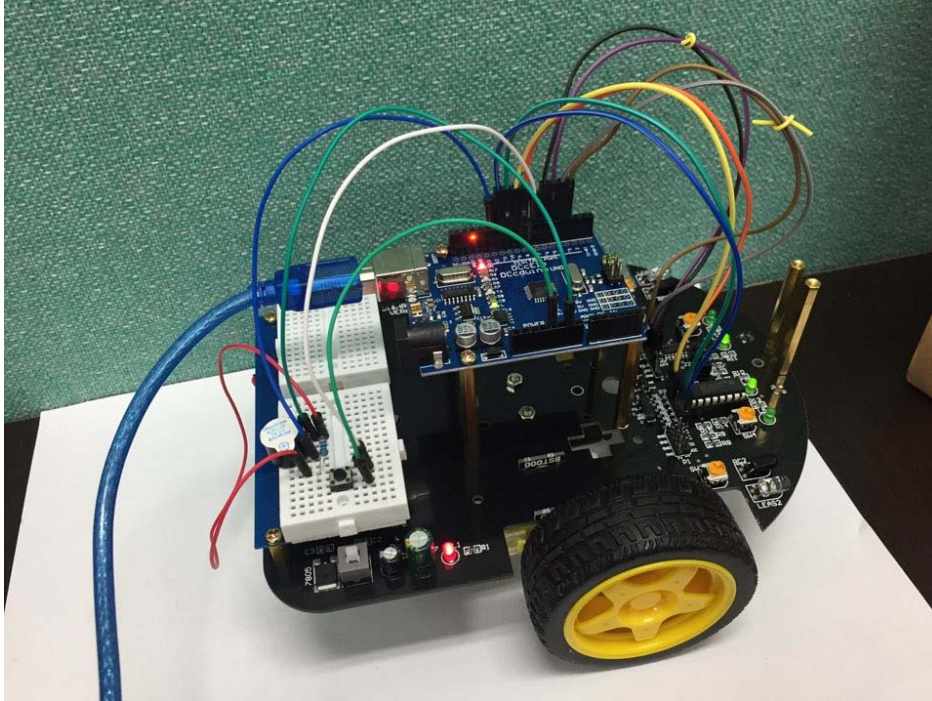
```

void keysacn()
//여기까지 [04]와 동일
void loop()
{
    keysacn();
    while(1)
    { //신호 있으면 LOW, 신호 없으면HIGH
        SR = digitalRead(SensorRight); //흰색이면 L3켜지고, 그렇지 않으면 L3꺼진다.
        SL=digitalRead(SensorLeft);
        if (SL == LOW&&SR==LOW) //왼쪽이 '0', 오른쪽이 '0'이면
            run(); //run함수 호출
        else if (SL == HIGH & SR == LOW) // 왼쪽이 '1', 오른쪽이 '0'이면
            left(); //left함수 호출
        else if (SR == HIGH & SL == LOW) //왼쪽이 '0', 오른쪽이 '1'이면
            right(); //right함수 호출
        else //
            brake(); //brake함수 호출
    }
}

```


06_스마트 자동차 적외선 장애물 회피1(기본 장애물)

①연결



②소스코드

06_스마트 자동차 적외선 장애물 회피1(기본 장애물)

```
const int SensorRight_2 = 6;  //P3.5 OUT4
const int SensorLeft_2 = 5;   //P3.4 OUT3
int SL_2;
int SR_2;

void setup()
{
    pinMode(SensorRight_2, INPUT);
    pinMode(SensorLeft_2, INPUT);
}

void run()
void brake(int time)
void left()
void spin_left(int time)
void right()
void spin_right(int time)
void back(int time)
void keysacn()
//여기까지 [05]과 동일
```

```
void loop()
{
  keysacn();
  while(1)
  {    //신호있으면LOW  신호없으면HIGH
    SR_2 = digitalRead(SensorRight_2);
    SL_2 = digitalRead(SensorLeft_2);
    if (SL_2 == HIGH&&SR_2==HIGH)
      run();
    else if (SL_2 == HIGH & SR_2 == LOW) //오른쪽에 장애물이 있으면 왼쪽으로 돈다.
      left();
    else if (SR_2 == HIGH & SL_2 == LOW) //왼쪽에 장애물이 있으면 오른쪽으로 돈다.
      right();
    else
      back(3);
  }
}
```

07_스마트 자동차 적외선 장애물 회피2 (물체로 향함)

①연결은 [06_스마트 자동차 적외선 장애물 회피1]과 동일

②소스코드

07_스마트 자동차 적외선 장애물 회피2(물체로 향함)

```
void setup()
void run()
void brake()
void left()
void spin_left(int time)
void right()
void spin_right(int time)
void back(int time)
void keysacn()
//여기까지 [06]과 동일
void loop()
{
    keysacn();
    while(1)
    {
        SR_2 = digitalRead(SensorRight_2);
        SL_2 = digitalRead(SensorLeft_2);
        if (SL_2 == LOW&&SR_2==LOW)
            run();
        else if (SL_2 == HIGH & SR_2 == LOW)
            right();
        else if (SR_2 == HIGH & SL_2 == LOW)
            left();
        else
            brake();
    }
}
```

08_(역 회전 방지와)스마트 자동차 적외선 장애물 회피 테스트(3)

①연결은 [06_스마트 자동차 적외선 장애물 회피1]과 동일

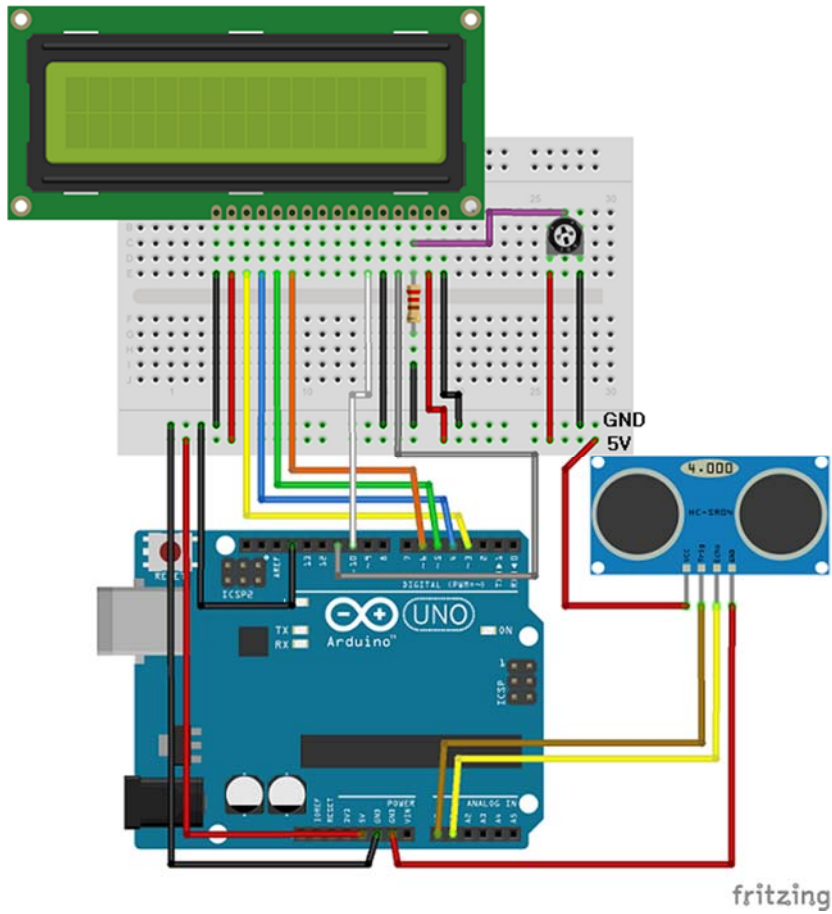
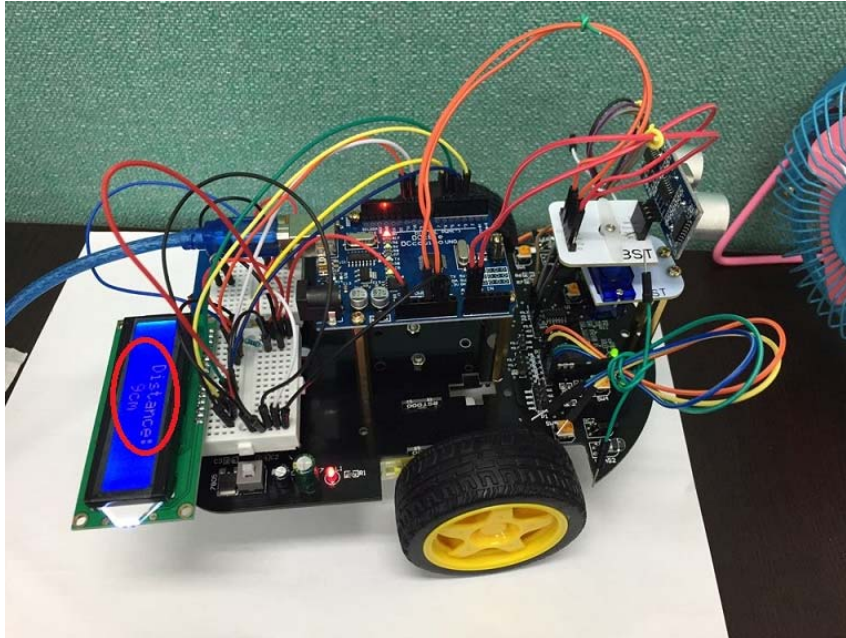
②소스코드

07_스마트 자동차 적외선 장애물 회피2(물체로 향함)

```
void setup()
void run()
void brake()
void left()
void spin_left(int time)
void right()
void spin_right(int time)
void back(int time)
void keysacn()
//여기까지 [07]과 동일
void loop()
{
  keysacn();
  while(1)
  {
    SR_2 = digitalRead(SensorRight_2);
    SL_2 = digitalRead(SensorLeft_2);
    if (SL_2 == HIGH&&SR_2==HIGH)
      run();
    else if (SL_2 == HIGH & SR_2 == LOW)
      left();
    else if (SR_2 == HIGH & SL_2 == LOW)
      right();
    else
    {
      Back(4.5);
      Spin right(4.5);
    }
  }
}
```

09_스마트 자동차 초음파 실험

①연결



②소스코드

09_스마트 자동차 초음파 실험

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(11,10,6,5,4,3);

int Echo = A1;
int Trig =A0;
int Distance = 0;
void setup()
{
    Serial.begin(9600);
    pinMode(Echo, INPUT);
    pinMode(Trig, OUTPUT);
    lcd.begin(16,2);
}

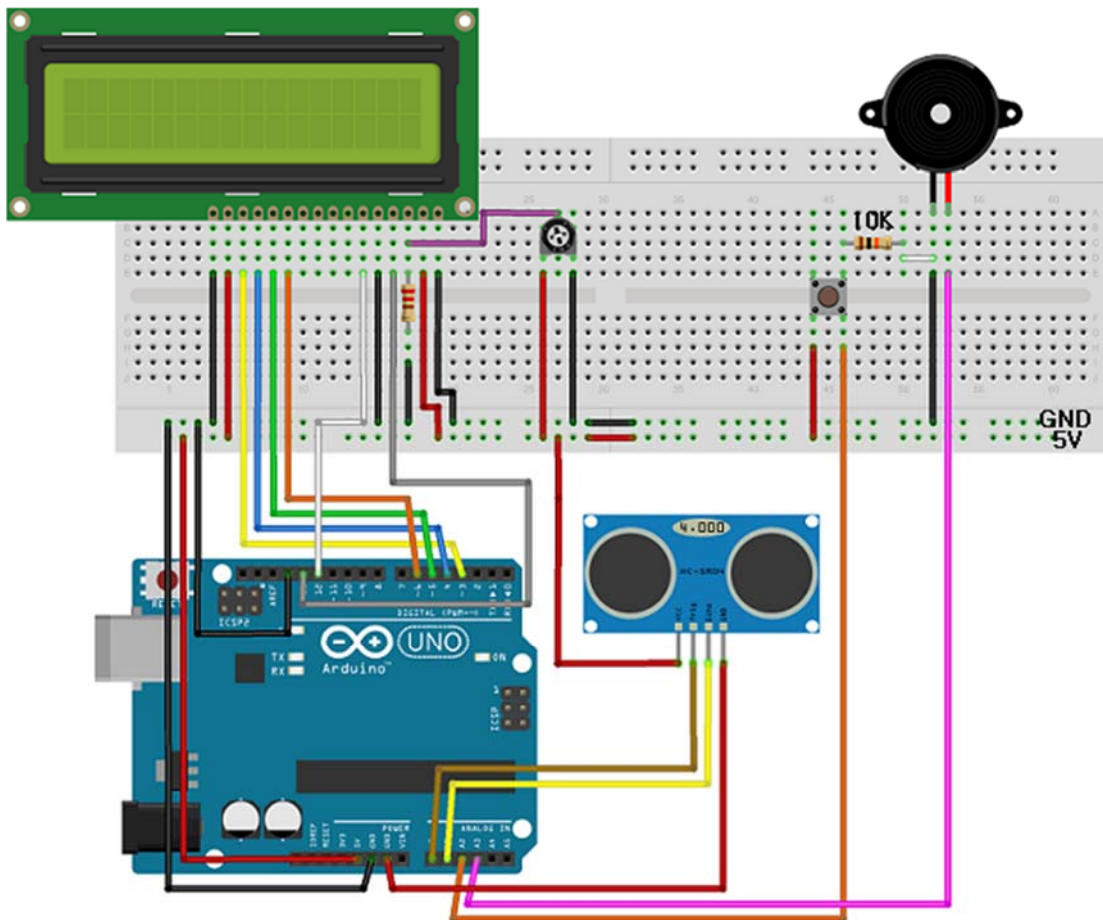
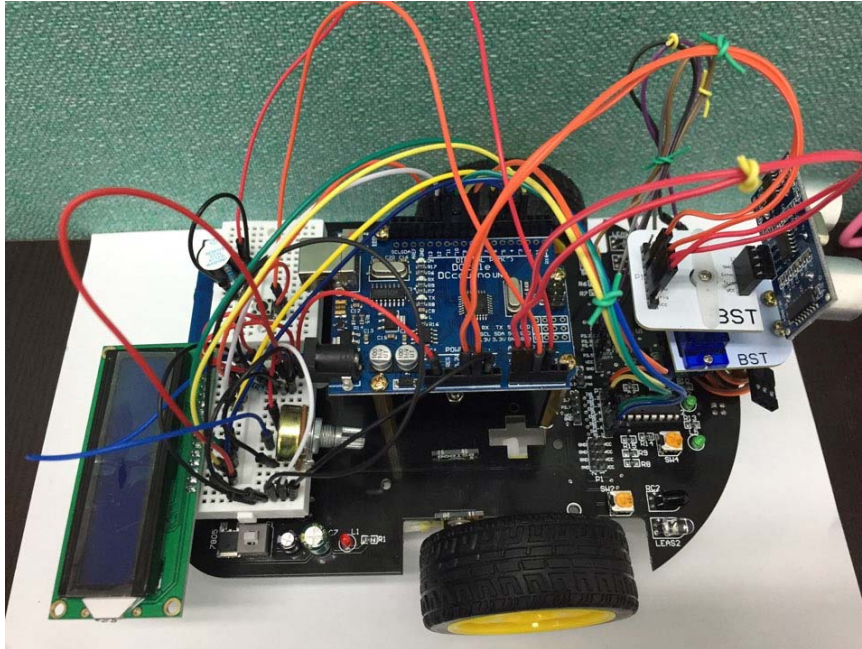
void Distance_test()
{
    digitalWrite(Trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    //Echo값을 받기 전에 이상한 신호가 발생하는 것을 처리함
    float Fdistance = pulseIn(Echo, HIGH);
    // 단위는 usec //pulseIn(핀번호, HIGH&LOW, 감지시간)
    //감지시간이란 몇초동안 값이 안오면 0을 반환하겠다. 라는 것. 따로 값을 설정 안 하면 자동1초설정
    Fdistance= Fdistance/58;
    //거리(d)=시간(t)*속도(v)라서
    //d=0.0001*170m/s=17/1000이므로 Fdistance의 단위가usec이기때문에 17/1000을 곱해주어(1000/17을
    나눠주어) 단위를 cm으로 나타내준다.
    Serial.print("Distance :"); //cm
    Serial.println(Fdistance); //시리얼 모니터에 Fdistance 출력
    Distance = Fdistance;
}

void loop()
{
    Distance_test();
    if((2<Distance)&(Distance<400)) //2cm이상, 4m미만
    {
        lcd.home();
        lcd.print(" Distance: "); LCD에 "Distance: " 출력
    }
}
```

```
    lcd.setCursor(6,2); //커서는 2행 6열에 위치
    lcd.print(Distance); //2행6열에 거리(distance) 출력
    lcd.print("cm"); //LCD에 "cm"출력
}
else //2미만, 400이상이면
{
    lcd.home();
    lcd.print("!!! Out of range"); "!!!Out of range" 출력
}
delay(250);
lcd.clear();
}
```


10_스마트 자동차 초음파 장애물 회피(NO Servo)

①연결



fritzing

②소스코드

10_스마트 자동차 초음파 장애물 회피(NO Servo)

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,6,5,4,3);
int Echo = A1;
int Trig =A0;
int Distance = 0;

int Left_motor_back=8;    IN1
int Left_motor_go=9;     IN2
int Right_motor_go=10;    IN3
int Right_motor_back=11;  IN4

int key=A2;
int beep=A3;

//const int SensorRight = 3;
//const int SensorLeft = 4;
//const int SensorRight_2 = 6;
//const int SensorLeft_2 = 5;
//int SL;
//int SR;
//int SL_2;
//int SR_2;

void setup()
{
    // pinMode(SensorRight, INPUT);
    // pinMode(SensorLeft, INPUT);
    //pinMode(SensorRight_2, INPUT);
    //pinMode(SensorLeft_2, INPUT);

    pinMode(Echo, INPUT);
    pinMode(Trig, OUTPUT);
    lcd.begin(16,2);
}

void run()
void brake(int time)
void left()
void spin_left(int time)
void right(int time)
void spin_right(int time)
```

```

void back(int time)
void keysacn()
//까지 [08]과 동일

void Distance_test()
{
    digitalWrite(Trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    float Fdistance = pulseIn(Echo, HIGH);
    Fdistance= Fdistance/58;

    Serial.print("Distance:");
    Serial.println(Fdistance);
    Distance = Fdistance;
}

void Distance_display()
{
    if((2<Distance)&(Distance<400))
    {
        lcd.home();
        lcd.print("    Distance: ");
        lcd.setCursor(6,2);
        lcd.print(Distance);
        lcd.print("cm");
    }
    else
    {
        lcd.home();
        lcd.print("!!! Out of range");
    }
    delay(250);
    lcd.clear();
}

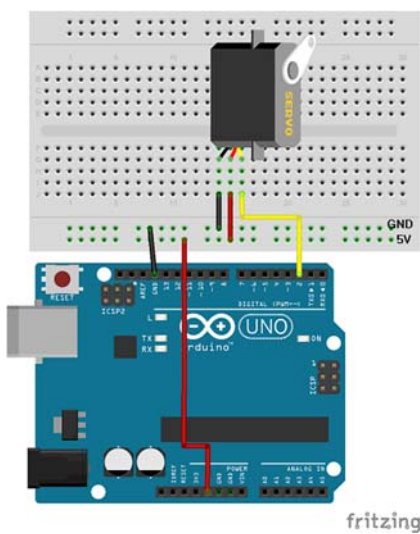
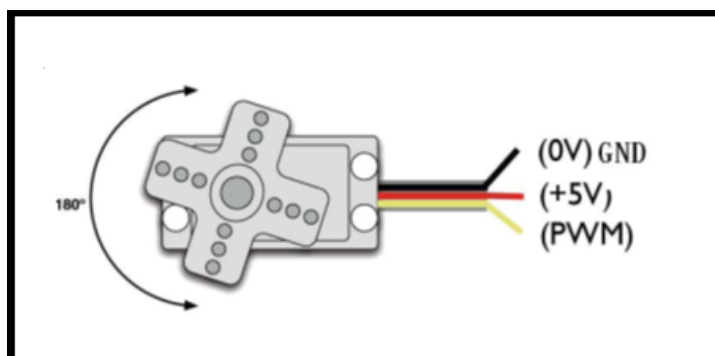
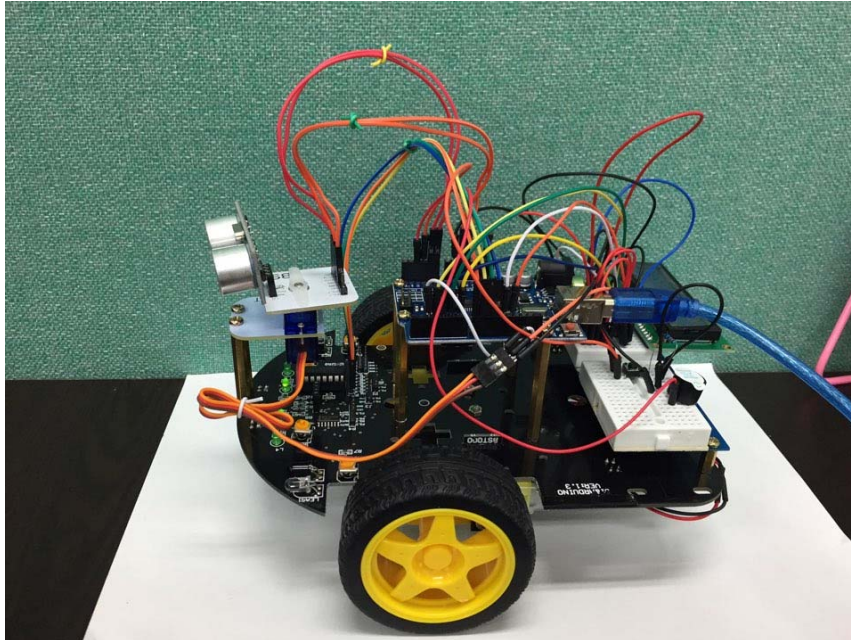
void loop()
{
    keysacn();
    while(1)
    {

```

```
Distance_test();  
Distance_display();  
if(Distance < 60)  
    while(Distance < 60)  
    {  
        right(1);  
        brake(1);  
        Distance_test();  
        Distance_display();  
    }  
else  
    run();  
}  
}
```

11_스마트 자동차 초음파 장애물 회피 테스트(Servo)

①연결



②소스코드

11_스마트 자동차 초음파 장애물 회피 테스트(Servo)

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,6,5,4,3);
int Echo = A1;
int Trig =A0;
int Front_Distance = 0;
int Left_Distance = 0;
int Right_Distance = 0;

int servopin=2;
int myangle;
int pulsewidth;
int val;

void setup()
{
  pinMode(servopin,OUTPUT);
}
void run()
void brake(int time)
void left()
void spin_left(int time)
void right(int time)
void spin_right(int time)
void back(int time)
void keysacn()
//까지 [10]과 동일

float Distance_test()
{
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  float Fdistance = pulseIn(Echo, HIGH);
  Fdistance= Fdistance/58;
  Return Fdistance;
}

void Distance_display(int Distance)
```

```

{
    if((2<Distance)&(Distance<400))
    {
        lcd.home();
        lcd.print("    Distance: ");
        lcd.setCursor(6,2);
        lcd.print(Distance);
        lcd.print("cm");
    }
    else
    {
        lcd.home();
        lcd.print("!!! Out of range");
    }
    delay(250);
    lcd.clear();
}

void servopulse(int servopin, int myangle)
{
    pulsewidth=(myangle*11)+500; //500-2480
    digitalWrite(servopin,HIGH);
    delayMicroseconds(pulsewidth);
    digitalWrite(servopin,LOW);
    delay(20-pulsewidth/1000);
}

void front_detection()
{
    for(int i=0;i<=5;i++) //PWM
    {
        servopulse(servopin,90);
    }
    Front_Distance = Distance_test();
}

void left_detection()
{
    for(int i=0;i<=15;i++)
    {
        servopulse(servopin,175); //PWM
    }
    Left_Distance = Distance_test();
}

```

```

void right_detection()
{
    for(int i=0;i<=15;i++)
    {
        servopulse(servopin,5);
    }
    Right_Distance = Distance_test();
}

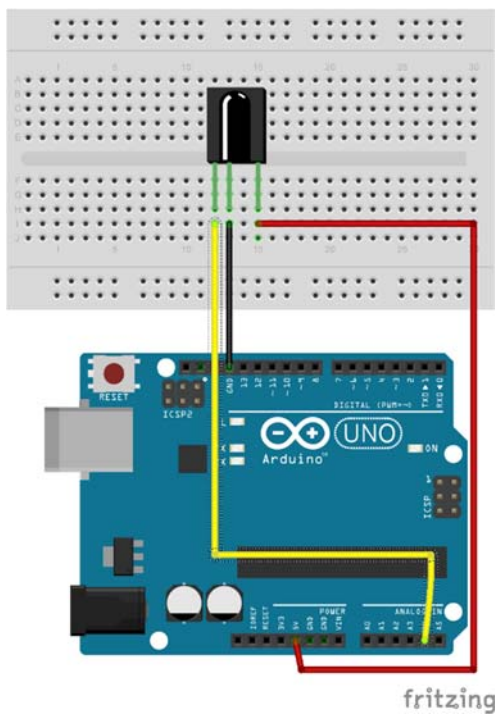
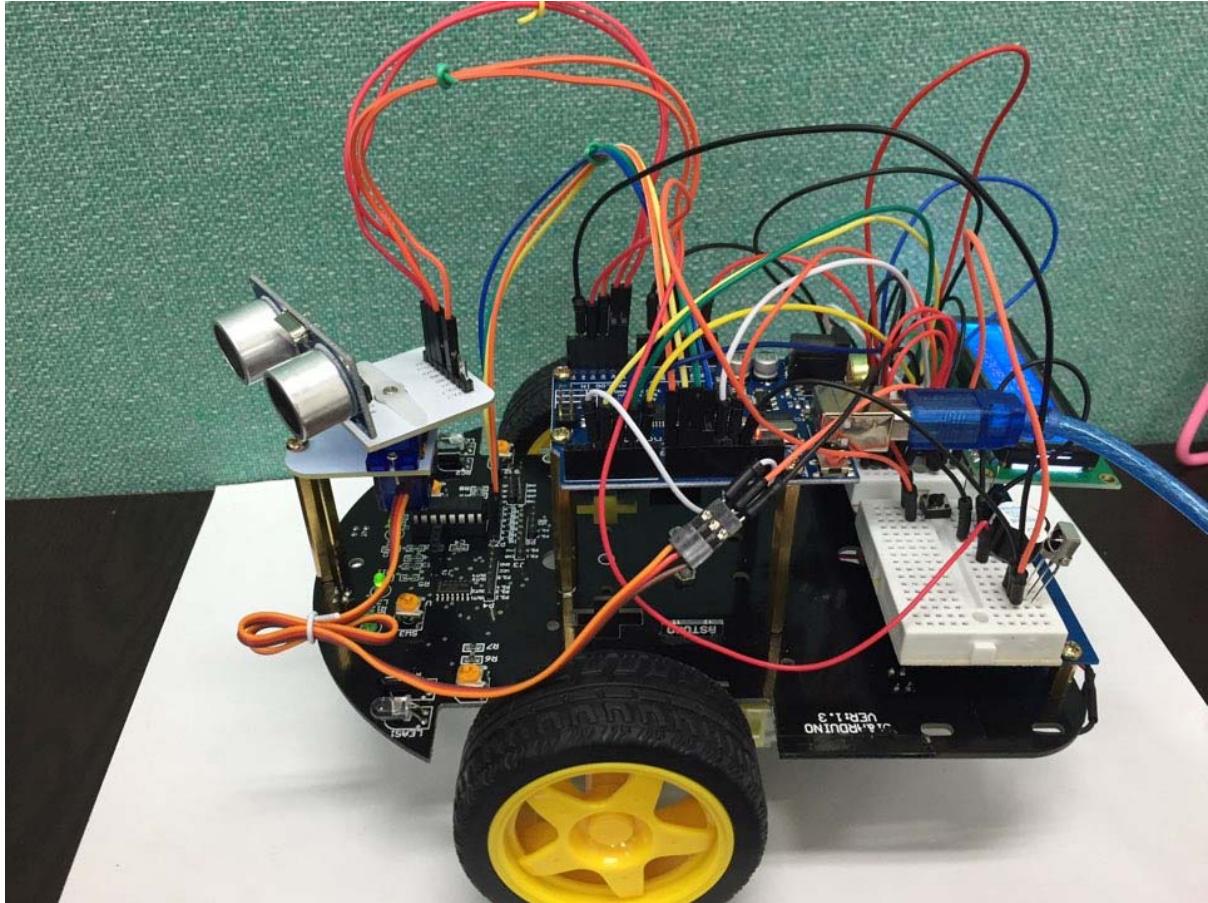
void loop()
{
    keysacn();
    while(1)
    {
        front_detection();
        if(Front_Distance < 32)
        {
            back(2);
            brake(2);
            left_detection();
            Distance_display(Left_Distance);
            right_detection();
            Distance_display(Right_Distance);

            if((Left_Distance < 35 ) &&( Right_Distance < 35 ))
                spin_left(0.7);
            else if(Left_Distance > Right_Distance)
            {
                left(3);
                brake(1);
            }
            else
            {
                right(3);
                brake(1);
            }
        }
        else
        {
            run();
        }
    }
}

```

12_적외선 원격 제어 스마트 자동차

①연결



②소스코드

12_적외선 원격 제어 스마트 자동차

```
#include <IRremote.h>
int RECV_PIN = A4;
IRrecv irrecv(RECV_PIN);
decode_results results;
int on = 0; //flag
unsigned long last = millis();

long run_car = 0x00FF18E7; //2번
long back_car = 0x00FF4AB5; //8번
long left_car = 0x00FF10EF; //4번
long right_car = 0x00FF5AA5; //6번
long stop_car = 0x00FF38C7; //5번
long left_turn = 0x00ff30CF; //1번
long right_turn = 0x00FF7A85; //3번
//=====
int Left_motor_back=8; //IN1
int Left_motor_go=9; //IN2

int Right_motor_go=10; //IN3
int Right_motor_back=11; //IN4

void setup()
{
    pinMode(13, OUTPUT);
    irrecv.enableIRIn(); // Start the receiver
}

void run()
void brake()
void left()
void spin_left()
void right()
void spin_right()
void back()
void dump(decode_results *results)
{
    int count = results->rawlen;
    if (results->decode_type == UNKNOWN)
    {
        brake();
    }
}
```

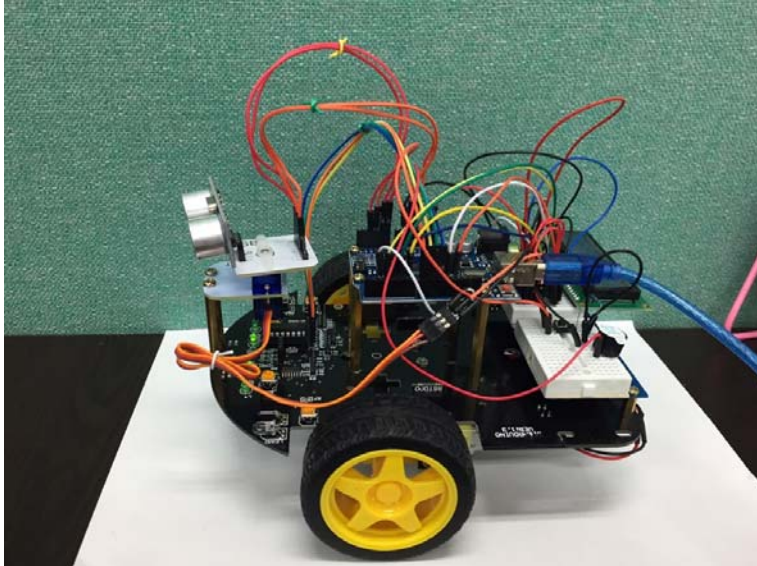
```

void loop()
{
  if (irrecv.decode(&results)) //라이브러리 함수 디코딩: 호출
  {
    // If it's been at least 1/4 second since the last
    // IR received, toggle the relay
    if (millis() - last > 250)
    {
      on = !on;
      digitalWrite(13, on ? HIGH : LOW);
      dump(&results); //적외선신호 디코드
      if (results.value == run_car ) //2번
        run();
      if (results.value == back_car ) //8번
        back();
      if (results.value == left_car ) //4번
        left();
      if (results.value == right_car ) //6번
        right();
      if (results.value == stop_car ) //5번
        brake();
      if (results.value == left_turn ) //1번
        spin_left();
      if (results.value == right_turn ) //3번
        spin_right();
      last = millis();
      irrecv.resume(); // Receive the next value
    }
  }
}

```

13_스마트 자동차 블랙라인 추적 및 초음파 장애물 회피

①연결



②소스코드

13_스마트 자동차 블랙라인 추적 및 초음파 장애물 회피

```
#include <Servo.h>
int Left_motor_back=8;    //IN1
int Left_motor_go=9;      //IN2
int Right_motor_go=10;    //IN3
int Right_motor_back=11;  //IN4

int key=7;
int beep=12;
const int SensorRight = 3;
const int SensorLeft = 4;
int SL;
int SR;
int Echo = A1;
int Trig =A0;
int Distance = 0;

void setup()
{
    pinMode(Left_motor_go,OUTPUT); // PIN 8
    pinMode(Left_motor_back,OUTPUT); // PIN 9
    pinMode(Right_motor_go,OUTPUT); // PIN 10
```

```

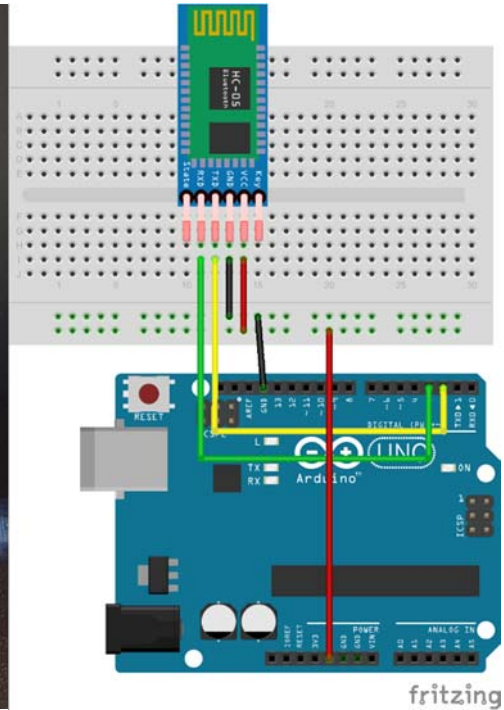
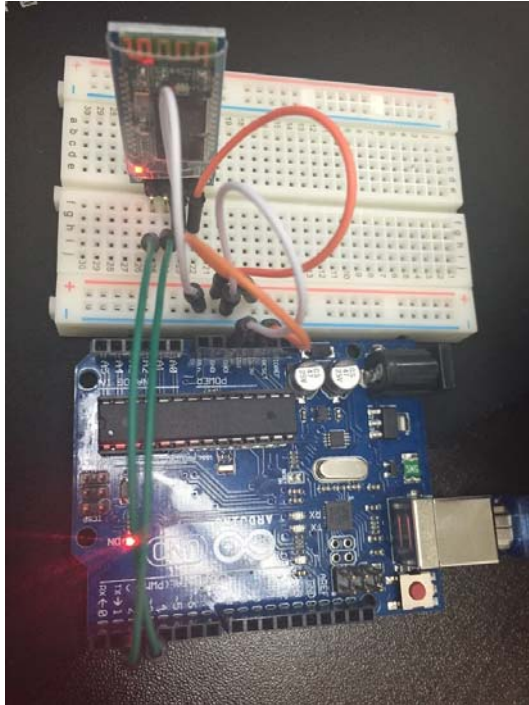
pinMode(Right_motor_back,OUTPUT); // PIN 11
pinMode(key,INPUT);
pinMode(beep,OUTPUT);
pinMode(SensorRight, INPUT);
pinMode(SensorLeft, INPUT);
Serial.begin(9600);
pinMode(Echo, INPUT);
pinMode(Trig, OUTPUT);
}
void run()
void brake()
void left()
void spin_left()
void right()
void spin_right()
void back()
void keysacn()
void Distance_test()  //[11]과 동일
//[12]와 동일
void loop()
{
    keysacn();
    while(1)
    {
        Distance_test();

        SR = digitalRead(SensorRight);
        SL = digitalRead(SensorLeft);
        if((Distance < 10)||(Distance > 400))
            brake();
        else
        {
            if (SL == LOW&&SR==LOW)
                run();
            else if (SL == HIGH & SR == LOW)
                left();
            else if (SR == HIGH & SL == LOW)
                right();
            else
                brake();
        }
    }
}

```

14_기본적인 Bluetooth 연결(1)-시리얼모니터 출력

①연결



②소스코드

14_기본적인 Bluetooth 연결(1)-시리얼 모니터 출력

```
#include<SoftwareSerial.h>
int BTS_RX=3;
int BTS_TX=2;
SoftwareSerial BTSerial(BTS_TX, BTS_RX);
void setup(){
    Serial.begin(9600);
    BTSerial.begin(9600);
}
void loop(){
    if(BTSerial.available())
        Serial.write(BTSerial.read());
    if(Serial.available())
        BTSerial.write(Serial.read());
}
```

③과정

- 휴대폰으로 Bluetooth를 연결하지 않은 상태에서 업로드 한 후 시리얼 모니터를 켜다.
- 기본적인 설정을 해준다.
 1. 테스트
명령어: AT 반환 값: OK
 2. 버전확인
명령어: AT+VERSION 반환 값: OKlinvorV1.8
 3. 이름변경
명령어: AT+NAME이름 반환 값: Oksetname
 4. 보드레이트 설정
명령어: AT+BAUD보드레이트 메뉴 값 반환 값: OK보드레이트
(1-1200, 2-2400, 3-4800, 4-9600, 5-19200)
 5. PIN(비밀번호)설정
명령어: AT+PIN네자리숫자 반환 값: OKsetPIN
(기본 비밀번호는 0000 또는 1234)
- 설정이 끝나면 휴대폰의 Bluetooth pro 어플을 켜다. (iphone과 HC-06은 연결X)
- Bluetooth를 켜고
설정 3번에서 이름을 설정했다면 그 이름으로 나오고, 설정하지 않았다면 NULL혹은 HC-06으로 나온다.
- 다시 시리얼 모니터를 켜 후 시리얼 모니터에 쓴 글을 전송하면 휴대폰에 글이 오고, 그 반대도 가능하다.

15_기본적인 Bluetooth 연결(2)-스마트 폰 출력

①연결은 14_기본적인 Bluetooth 연결(1)과 동일

②소스코드

15_기본적인 Bluetooth 연결(2)-스마트 폰 출력

```
#include<SoftwareSerial.h>
int BTS_RX=3;
int BTS_TX=2;
SoftwareSerial BTSerial(BTS_TX, BTS_RX);
void setup(){
    Serial.begin(9600);
    BTSerial.begin(9600);
}
void loop()
{
    int data;
    if(BTSerial.available())
    {
        data = BTSerial.read();
        BTSerial.write(data);
    }
}
```

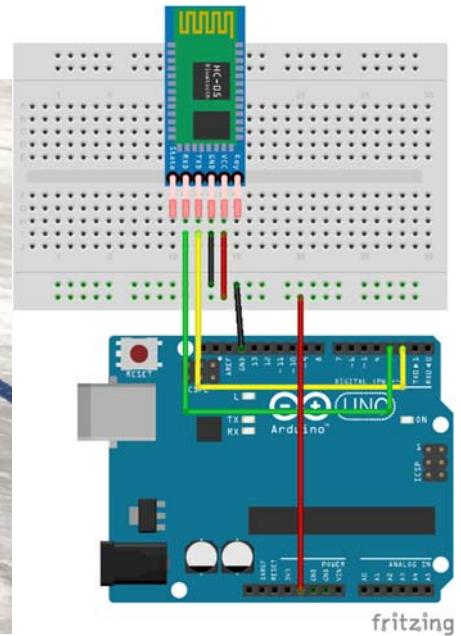
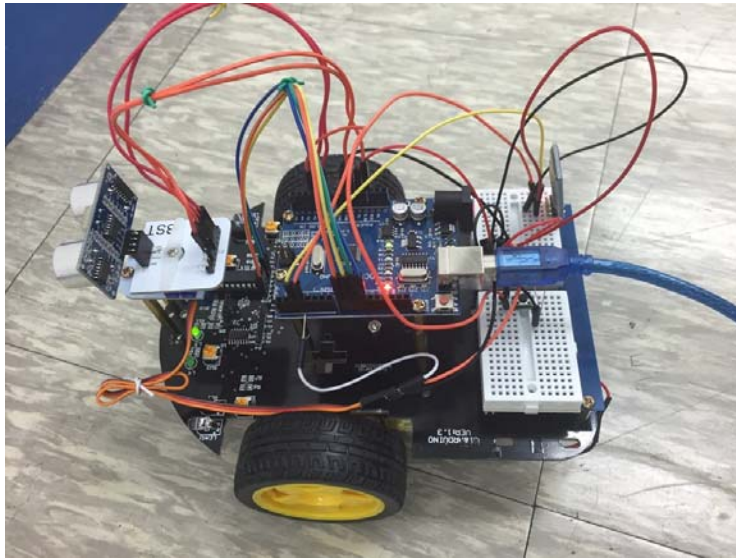
③과정

- Bluetooth와 연결되지 않은 상태에서 업로드 한다.
- Bluetooth Controller 이라는 어플에 들어간다.
- Bluetooth를 연결하고, <키 설정>을 눌러 이름과 데이터를 설정한 후 ok를 누른다. (오른쪽 사진)
- 키를 각각 누르면 해당하는 데이터가 화면에 뜬다.



16_Bluetooth를 이용한 스마트 자동차 컨트롤

①연결



②소스코드

16_Bluetooth를 이용한 스마트 자동차 컨트롤

```
#include<SoftwareSerial.h>
int BTS_RX=3;
int BTS_TX=2;
SoftwareSerial BTSerial = SoftwareSerial(BTS_TX, BTS_RX);
#define FORWARD '1' //forward
#define FORWARD_L '2' //LEFT
#define FORWARD_R '4' //RIGHT
#define BACKWARD '5' //BACK
#define STOP '3' //STOP

int Left_motor_back=8;
int Left_motor_go=9;
int Right_motor_go=10;
int Right_motor_back=11;

void run() ;
void brake() ;
void left() ;
void spin_left() ;
void right() ;
void spin_right();
```



```
void back()
```

```
//[13]과 동일
```

```
void setup(){
```

```
    Serial.begin(9600);
```

```
    BTSerial.begin(9600);
```

```
    pinMode(Left_motor_go,OUTPUT);
```

```
    pinMode(Left_motor_back,OUTPUT);
```

```
    pinMode(Right_motor_go,OUTPUT);
```

```
    pinMode(Right_motor_back,OUTPUT);
```

```
}
```

```
void loop(){
```

```
    char direct;
```

```
    if(BTSerial.available()>0)
```

```
{
```

```
    Direct = BTSerial.read();
```

```
    if(direct == FORWARD)
```

```
        Run();
```

```
    Else if(direct ==FORWARD_L)
```

```
        Left();
```

```
    else if(direct ==FORWARD_R)
```

```
        right();
```

```
    else if(direct ==BACKWARD)
```

```
        back();
```

```
    else if(direct ==STOP)
```

```
        brake();
```

```
    else;
```

```
}
```

```
    delay(100);
```

```
}
```