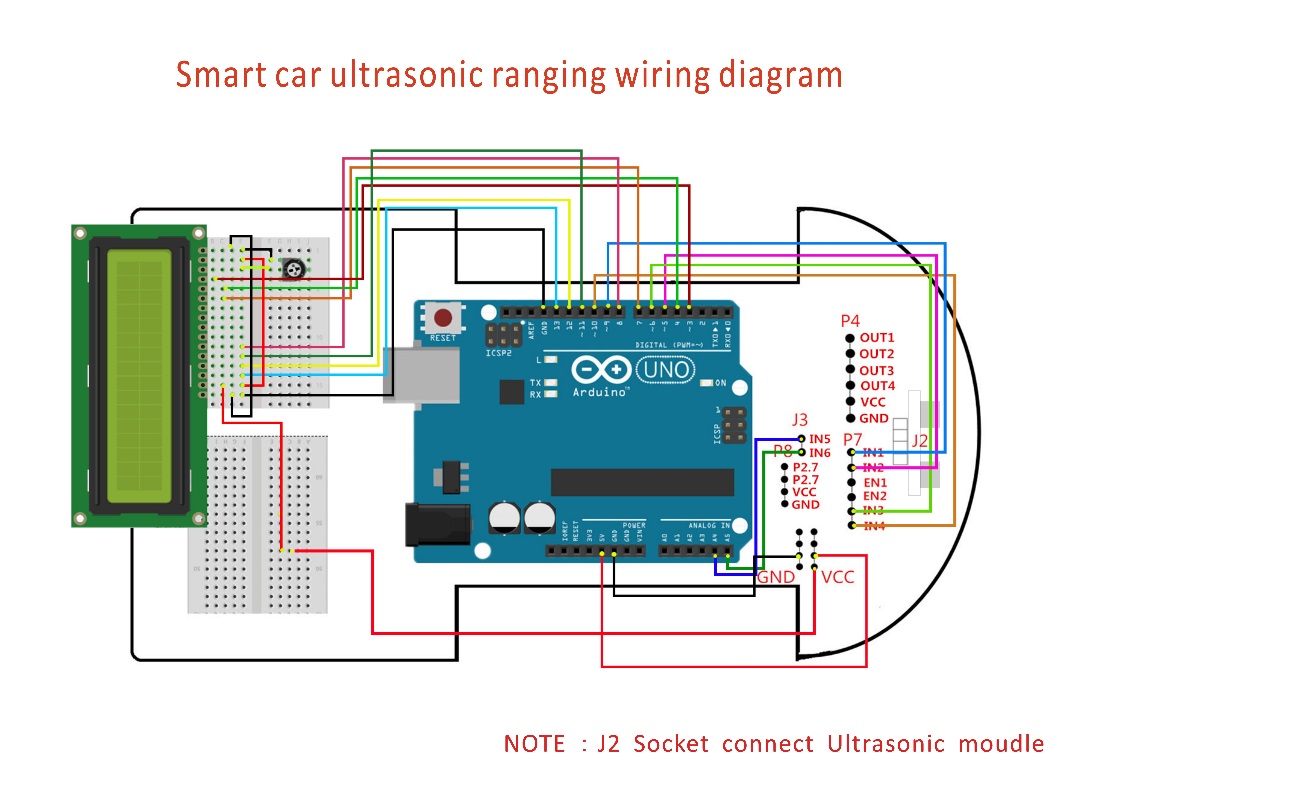
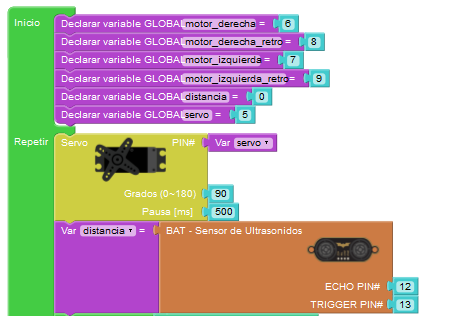
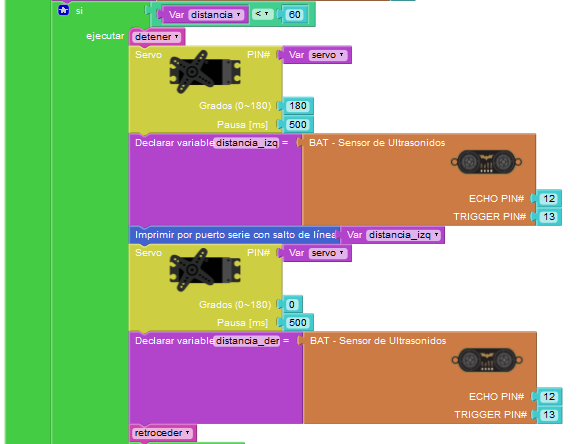
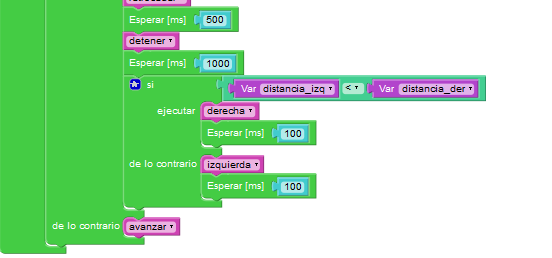
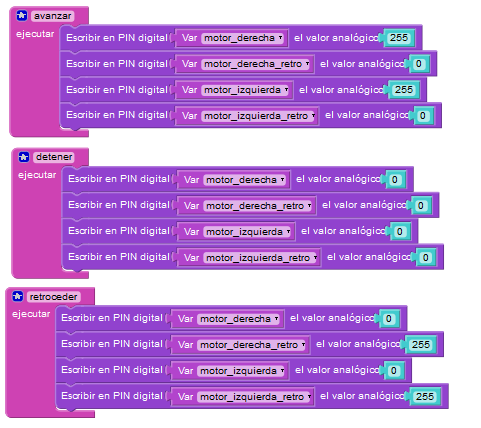
**Carro esquiva obstáculo**













Código  
#include <Servo.h>  
Servo servos[13];

/\*\*\* Global variables \*\*\*/  
int motor\_derecha=6;  
int motor\_derecha\_retro=8;  
int motor\_izquierda=7;  
int motor\_izquierda\_retro=9;  
int distancia=0;  
int servo=5;

/\*\*\* Function declaration \*\*\*/

//bqBAT

long TP\_init(int trigger\_pin, int echo\_pin);

long Distance(int trigger\_pin, int echo\_pin);

void avanzar ();

void detener ();

void retroceder ();

void izquierda ();

void derecha ();

void setup()

{

pinMode( 12 , INPUT );

pinMode( 13 , OUTPUT );

Serial.begin(9600);

}

void loop()

{

servos[servo].attach(servo);

servos[servo].write(90);

delay(500);

distancia=Distance(13,12);

if (distancia < 60) {

detener();

servos[servo].attach(servo);

servos[servo].write(180);

delay(500);

int distancia\_izq=Distance(13,12);

Serial.println(distancia\_izq);

servos[servo].attach(servo);

servos[servo].write(0);

delay(500);

int distancia\_der=Distance(13,12);

retroceder();

delay(500);

detener();

delay(1000);

if (distancia\_izq < distancia\_der) {

derecha();

delay(100);

}else {

izquierda();

delay(100);

}

}else {

avanzar();

}

}

/\*\*\* Function definition \*\*\*/

//bqBAT

long TP\_init(int trigger\_pin, int echo\_pin)

{

digitalWrite(trigger\_pin, LOW);

delayMicroseconds(2);

digitalWrite(trigger\_pin, HIGH);

delayMicroseconds(10);

digitalWrite(trigger\_pin, LOW);

long microseconds = pulseIn(echo\_pin ,HIGH);

return microseconds;

}

long Distance(int trigger\_pin, int echo\_pin)

{

long microseconds = TP\_init(trigger\_pin, echo\_pin);

long distance;

distance = microseconds/29/2;

if (distance == 0){

distance = 999;

}

return distance;

}

void avanzar () {

pinMode(motor\_derecha,OUTPUT);

analogWrite(motor\_derecha,255);

pinMode(motor\_derecha\_retro,OUTPUT);

analogWrite(motor\_derecha\_retro,0);

pinMode(motor\_izquierda,OUTPUT);

analogWrite(motor\_izquierda,255);

pinMode(motor\_izquierda\_retro,OUTPUT);

analogWrite(motor\_izquierda\_retro,0);

}

void detener () {

pinMode(motor\_derecha,OUTPUT);

analogWrite(motor\_derecha,0);

pinMode(motor\_derecha\_retro,OUTPUT);

analogWrite(motor\_derecha\_retro,0);

pinMode(motor\_izquierda,OUTPUT);

analogWrite(motor\_izquierda,0);

pinMode(motor\_izquierda\_retro,OUTPUT);

analogWrite(motor\_izquierda\_retro,0);

}

void retroceder () {

pinMode(motor\_derecha,OUTPUT);

analogWrite(motor\_derecha,0);

pinMode(motor\_derecha\_retro,OUTPUT);

analogWrite(motor\_derecha\_retro,255);

pinMode(motor\_izquierda,OUTPUT);

analogWrite(motor\_izquierda,0);

pinMode(motor\_izquierda\_retro,OUTPUT);

analogWrite(motor\_izquierda\_retro,255);

}

void izquierda () {

pinMode(motor\_derecha,OUTPUT);

analogWrite(motor\_derecha,0);

pinMode(motor\_derecha\_retro,OUTPUT);

analogWrite(motor\_derecha\_retro,0);

pinMode(motor\_izquierda,OUTPUT);

analogWrite(motor\_izquierda,255);

pinMode(motor\_izquierda\_retro,OUTPUT);

analogWrite(motor\_izquierda\_retro,0);

}

void derecha () {

pinMode(motor\_derecha,OUTPUT);

analogWrite(motor\_derecha,255);

pinMode(motor\_derecha\_retro,OUTPUT);

analogWrite(motor\_derecha\_retro,0);

pinMode(motor\_izquierda,OUTPUT);

analogWrite(motor\_izquierda,0);

pinMode(motor\_izquierda\_retro,OUTPUT);

analogWrite(motor\_izquierda\_retro,0);

}