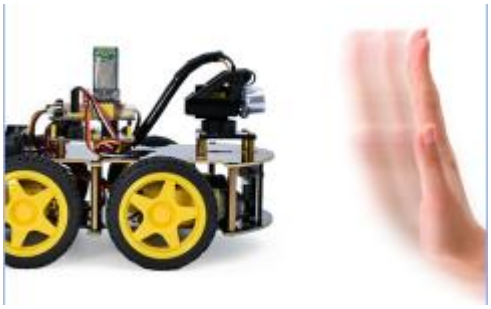
Project 12 Ultrasonic Following Smart Car



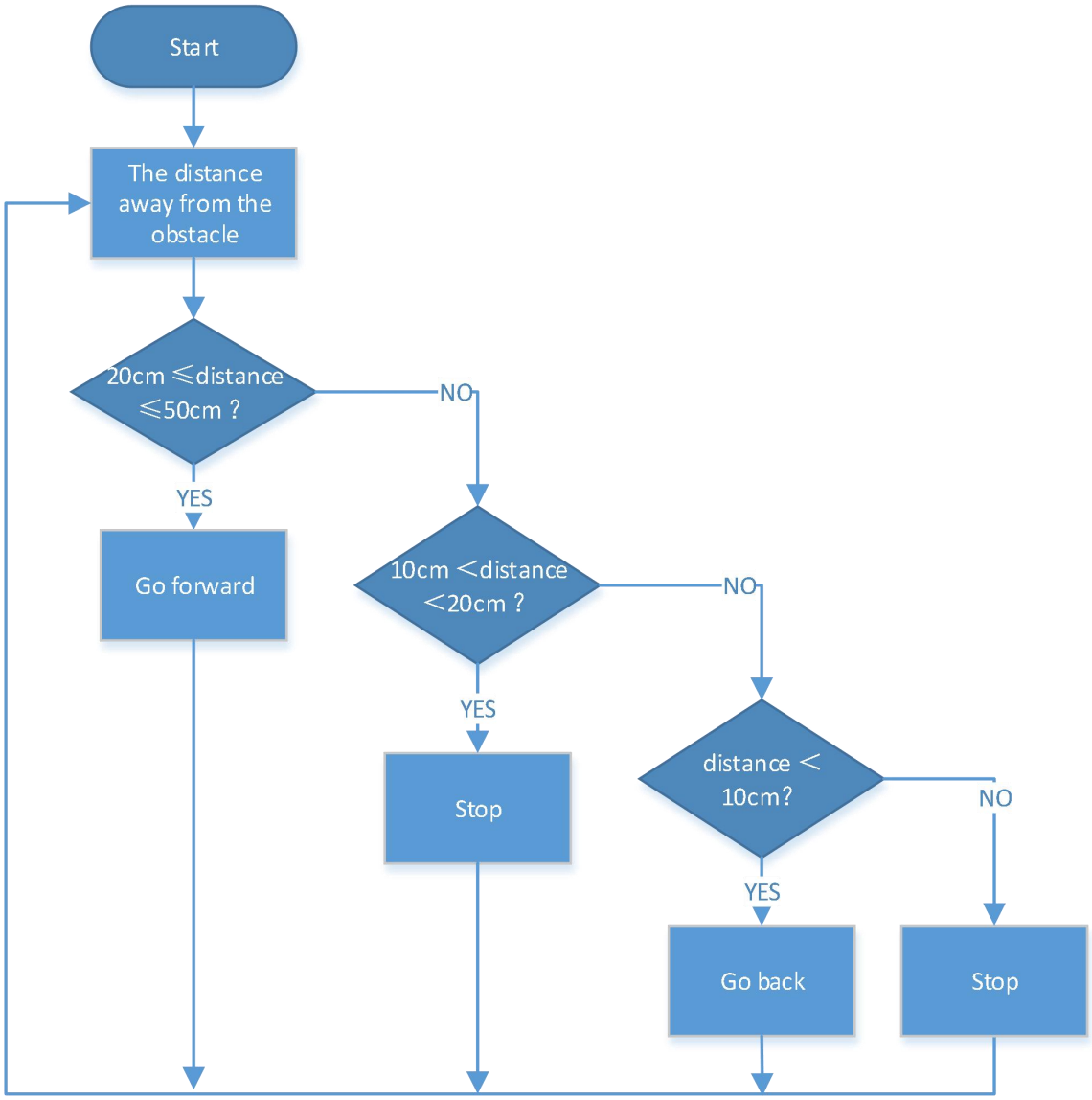
1.Description

In this project, we will look to detect the distance between the 4WD smart car and the obstacles ahead through an ultrasonic sensor to drive two motors in a way that make the car move and make the 8*8 LED board show a smile facial pattern.

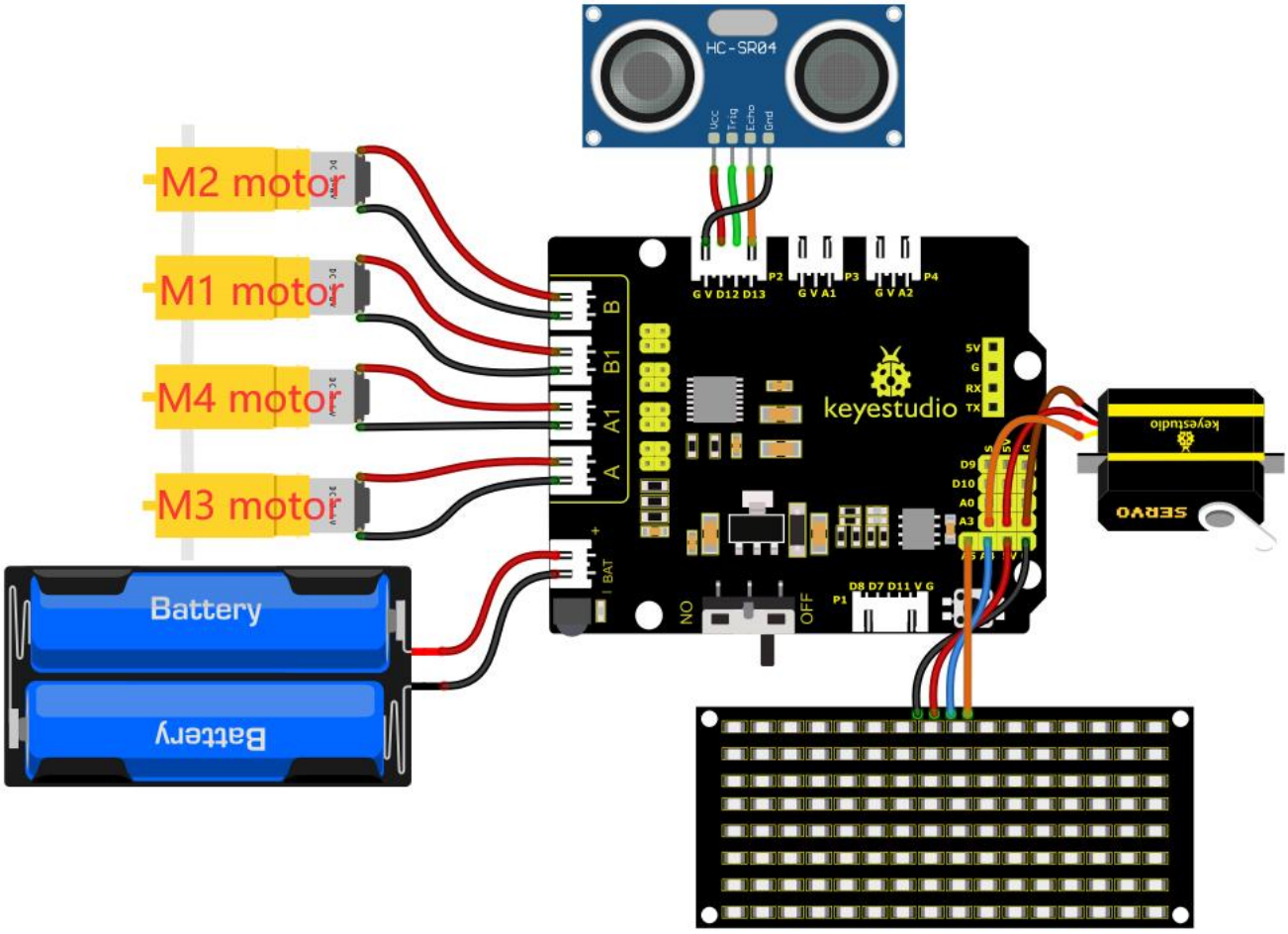
2.Flow Chart

Detection	Measured distance of front obstacles	distance (unit: cm)
Setting	8*16 LED board shows a smile pattern.	
	Set servo to 90°	
Condition	distance≥20 and distance≤50	
Status	Go forward	
Condition	distance > 10 and distance < 20	
	distance > 50	
Condition	stop	

Condition	distance≤10
Condition	Go back



3.Wiring Diagram



Wiring up:

1. GND, VCC, SDA and SCL of the 8*8 LED board are connected to G (GND), V (VCC), A4 and A5 of the expansion board.

2. VCC, Trig, Echo and Gnd of the ultrasonic sensor are connected to 5V(V), D12(S), D13(S) and Gnd(G)

3. The servo is connected to G, V and A3. The brown wire is interfaced

with Gnd(G), the red wire is interfaced with 5V(V) and the orange wire is interfaced with A3.

4. The power is connected to the BAT port

4.Test Code

```
//*****************************************************************************
/*
 keyestudio 4wd BT Car
 lesson 12
 Flowing Car
 http://www.keyestudio.com
*/
#define SCL_Pin  A5  //Set the clock pin to A5
#define SDA_Pin  A4  //Set data pin to A4

//Array, used to store the data of pattern, can be calculated by yourself or obtained from the modulus tool
unsigned char smile[] = {0x00, 0x00, 0x1c, 0x02, 0x02, 0x02, 0x5c, 0x40, 0x40, 0x5c, 0x02, 0x02, 0x02, 0x1c, 0x00, 0x00};

const int servopin = A3;//Set the pin of the steering gear

#include "SR04.h" //define the function library of ultrasonic sensor
#define TRIG_PIN 12// set the signal of ultrasonic sensor to D12
#define ECHO_PIN 13// set the signal of ultrasonic sensor to D13
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
long distance;

int left_ctrl = 2;//define the direction control pins of group B motor
int left_pwm = 5;//define the PWM control pins of group B motor
int right_ctrl = 4;//define the direction control pins of group A motor
int right_pwm = 6;//define the PWM control pins of group A motor

void setup() {
  pinMode(left_ctrl,OUTPUT);//set direction control pins of group B motor to OUTPUT
  pinMode(left_pwm,OUTPUT);//set PWM control pins of group B motor to OUTPUT
  pinMode(right_ctrl,OUTPUT);//set direction control pins of group A motor to OUTPUT
  pinMode(right_pwm,OUTPUT);//set PWM control pins of group A motor to OUTPUT
  pinMode(TRIG_PIN, OUTPUT); //Set the trig pin to output
  pinMode(ECHO_PIN, INPUT); //Set the echo pin to input
  pinMode(SCL_Pin,OUTPUT);//Set the clock pin to output
  pinMode(SDA_Pin,OUTPUT);//Set the data pin to output
```

```
servopulse(servopin,90);//Set the initial steering gear angle to 90°
  delay(500); //waits 500ms
  matrix_display(smile);  //display smiling expression pattern
}

void loop() {
  distance = sr04.Distance();//the distance detected by ultrasonic sensor
   if(distance <= 10)//if distance is less than 10
  {
    back();//go back
  }
  else if((distance > 10)&&(distance< 20 ))//if 10<distance<20
  {
    Stop();//stop
  }
  else if((distance >= 20)&&(distance <= 50))//if 20≤distance＜50
{
    front();//follow
  }
  else//otherwise
  {
    Stop();//stop
  }
}

void front()//define the status of going front
{
  digitalWrite(left_ctrl,HIGH);
  analogWrite(left_pwm,100);
  digitalWrite(right_ctrl,HIGH);
  analogWrite(right_pwm,100);
}
void back()//define the status of going back
{
  digitalWrite(left_ctrl,LOW);
  analogWrite(left_pwm,150);
  digitalWrite(right_ctrl,LOW);
  analogWrite(right_pwm,150);
}
void left()//define the status of turning left
{
  digitalWrite(left_ctrl, LOW);
  analogWrite(left_pwm, 100);
  digitalWrite(right_ctrl, HIGH);
```

```
  analogWrite(right_pwm, 155);
}
void right()//define the status of right turning
{
  digitalWrite(left_ctrl, HIGH);
  analogWrite(left_pwm, 155);
  digitalWrite(right_ctrl, LOW);
  analogWrite(right_pwm, 100);
}
void Stop()//define the state of stop
{
  digitalWrite(left_ctrl, LOW);
  analogWrite(left_pwm,0);
  digitalWrite(right_ctrl, LOW);
  analogWrite(right_pwm,0);
}

void servopulse(int servopin,int myangle)//Steering gear running angle
{
  for(int i=0; i<30; i++)
  {
    int pulsewidth = (myangle*11)+500;
    digitalWrite(servopin,HIGH);
    delayMicroseconds(pulsewidth);
    digitalWrite(servopin,LOW);
    delay(20-pulsewidth/1000);
  }
}

//this function is used for dot matrix display
void matrix_display(unsigned char matrix_value[])
{
  IIC_start();  //the function that calls the data transfer start condition
  IIC_send(0xc0);  //select address

  for (int i = 0; i < 16; i++) //the pattern data is 16 bytes
  {
    IIC_send(matrix_value[i]); //Transmit the data of the pattern
  }
  IIC_end();   //End pattern data transmission
  IIC_start();
  IIC_send(0x8A);  //Display control, select 4/16 pulse width
  IIC_end();
}
```

```
//Conditions under which data transmission begins
void IIC_start()
{
  digitalWrite(SDA_Pin, HIGH);
  digitalWrite(SCL_Pin, HIGH);
  delayMicroseconds(3);
  digitalWrite(SDA_Pin, LOW);
  delayMicroseconds(3);
  digitalWrite(SCL_Pin, LOW);
}
//Indicates the end of data transmission
void IIC_end()
{
  digitalWrite(SCL_Pin, LOW);
  digitalWrite(SDA_Pin, LOW);
  delayMicroseconds(3);
  digitalWrite(SCL_Pin, HIGH);
  delayMicroseconds(3);
  digitalWrite(SDA_Pin, HIGH);
  delayMicroseconds(3);
}
//transmit data
void IIC_send(unsigned char send_data)
{
  for (byte mask = 0x01; mask != 0; mask <<= 1) //Each byte has 8 bits and is checked bit by bit starting
at the lowest level
  {
    if (send_data & mask) { //Sets the high and low levels of SDA_Pin depending on whether each bit of the
byte is a 1 or a 0
      digitalWrite(SDA_Pin, HIGH);
    } else {
      digitalWrite(SDA_Pin, LOW);
    }
    delayMicroseconds(3);
    digitalWrite(SCL_Pin, HIGH); //Pull the clock pin SCL_Pin high to stop data transmission
    delayMicroseconds(3);
    digitalWrite(SCL_Pin, LOW); //pull the clock pin SCL_Pin low to change the SIGNAL of SDA
  }
}
//*****************************************************************************
```

5.Test Result

After successfully uploading the code to the V4.0 board, connect the wirings according to the wiring diagram, power on the external power then turn the DIP switch to ON. Set the servo to 90°, the smart car will move with the obstacles and the 8X16 LED board will show "smile".