#include <Servo.h>

#define SER1\_BAUD 115200 //驱动引脚

#define PWMB\_IN1 11 //定义IN1引脚

#define PWMB\_IN2 6 //定义IN2

#define PWMA\_IN1 5 //定义IN3

#define PWMA\_IN2 3 //定义IN4

#define Servor\_Pin 7

#define BEEP\_PIN 12

#define NLED\_MS\_BETWEEN 500

#define DUOJI\_MS\_BETWEEN 20.000

#define IR1 A1

#define IR2 A0

#define IR3 A3

#define IR4 A2

#define IR5 10

int Out1,Out2,Out3,Out4,Out5;

int pwm\_value=1500;

int Echo = A5; // Echo回声脚()

int Trig = 13; // Trig 触发脚()

int Distance = 0;

char val=0;

Servo myservo; //创建舵机伺服对象数组

unsigned char handle\_ms\_between( unsigned long \*time\_ms, unsigned int ms\_between);

void handle\_nled();

void dida(u8 times, u8 frequency);

void Motor\_Forward(char motor,char pwm);

void Motor\_Back(char motor,char pwm);

void Motor\_Left(char motor,char pwm);

void Motor\_Right(char motor,char pwm);

void Motor\_Turn\_Left(char motor,char pwm);

void Motor\_Turn\_Right(char motor,char pwm);

void Motor\_Stop(char motor);

void setup()

{

pinMode(BEEP\_PIN, OUTPUT);

pinMode(Echo, INPUT); // 定义超声波输入脚

pinMode(Trig, OUTPUT); // 定义超声波输出脚

Serial.begin(SER1\_BAUD);

myservo.attach(Servor\_Pin);

myservo.writeMicroseconds(1500);

dida(1, 1000);

delay(1000);

}

void loop()

{

while(1)

{

Out1 = digitalRead(IR1);

Out2 = digitalRead(IR2);

Out3 = digitalRead(IR3);

Out4 = digitalRead(IR4);

Out5 = digitalRead(IR5);

Distance\_test(); //测量前方距离

if(Distance < 25) //数值为碰到障碍物的距离

{

delay(10);

Distance\_test(); //测量前方距离

while(Distance<25) //再次判断是否有障碍物，若有则转动方向后，继续判断

{

Motor\_Stop(1);

Motor\_Stop(2);

myservo.writeMicroseconds(1500);

delay(400);

Distance\_test();//测量前方距离

}

}

else

{

if(Out1==1 && Out2==1 && Out3==0 && Out4==1 && Out5==1)

{

Motor\_Forward(1,190);

Motor\_Forward(2,190);

myservo.writeMicroseconds(1500);

}

else if(Out1==1 && Out2==1 && Out3==1 && Out4==0 && Out5==1)

{

Motor\_Forward(1,190);

Motor\_Forward(2,190);

myservo.writeMicroseconds(1700);

}

else if(Out1==1 && Out2==1 && Out3==1 && Out4==1 && Out5==0)

{

Motor\_Forward(1,190);

Motor\_Forward(2,190);

myservo.writeMicroseconds(2000);

}

else if(Out1==1 && Out2==0 && Out3==1 && Out4==1 && Out5==1)

{

Motor\_Forward(1,190);

Motor\_Forward(2,190);

myservo.writeMicroseconds(1300);

}

else if(Out1==0 && Out2==1 && Out3==1 && Out4==1 && Out5==1)

{

Motor\_Forward(1,190);

Motor\_Forward(2,190);

myservo.writeMicroseconds(1000);

}

else if(Out1==0 && Out2==0 && Out3==0 && Out4==0 && Out5==0)

{

Motor\_Stop(1);

Motor\_Stop(2);

myservo.writeMicroseconds(1500);

}

else

{

Motor\_Forward(1,160);

Motor\_Forward(2,160);

}

}

}

}

unsigned char handle\_ms\_between( unsigned long \*time\_ms, unsigned int ms\_between) {

if(millis() - \*time\_ms < ms\_between) {

return 0;

} else{

\*time\_ms = millis();

return 1;

}

}

void dida(u8 times, u8 frequency) {

for(byte i = 0; i < times; i++ ) {

digitalWrite(BEEP\_PIN, LOW);

delay(frequency);

delay(frequency);

digitalWrite(BEEP\_PIN, HIGH );

delay(frequency);

delay(frequency);

}

}

void Distance\_test() // 量出前方距离

{

digitalWrite(Trig, LOW); // 给触发脚低电平2μs

delayMicroseconds(2);

digitalWrite(Trig, HIGH); // 给触发脚高电平10μs，至少10μs

delayMicroseconds(10);

digitalWrite(Trig, LOW); // 一直给触发脚低电

float Fdistance = pulseIn(Echo, HIGH); // 读取高电平时间(单位：微秒)

Fdistance= Fdistance/58; //除以58等于厘米是因为Y米=（X秒\*344）/2

Serial.print("Distance:"); //输出距离（厘米）

Serial.println(Fdistance); //显距离

Distance = Fdistance;

}

void Motor\_Forward(char motor,char pwm)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,pwm);

analogWrite(PWMA\_IN2,255);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,pwm);

analogWrite(PWMB\_IN2,255);

}

}

void Motor\_Right(char motor,char pwm)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,pwm);

analogWrite(PWMA\_IN2,255);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,255);

analogWrite(PWMB\_IN2,255);

}

}

void Motor\_Left(char motor,char pwm)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,255);

analogWrite(PWMA\_IN2,255);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,pwm);

analogWrite(PWMB\_IN2,255);

}

}

void Motor\_Turn\_Right(char motor,char pwm)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,pwm);

analogWrite(PWMA\_IN2,255);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,255);

analogWrite(PWMB\_IN2,pwm);

}

}

void Motor\_Turn\_Left(char motor,char pwm)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,255);

analogWrite(PWMA\_IN2,pwm);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,pwm);

analogWrite(PWMB\_IN2,255);

}

}

void Motor\_Stop(char motor)

{

if(motor==1)

{

analogWrite(PWMA\_IN1,255);

analogWrite(PWMA\_IN2,255);

}

else if(motor==2)

{

analogWrite(PWMB\_IN1,255);

analogWrite(PWMB\_IN2,255);

}

}