const int irReceiverPin = 2; // 红外接收器的 OUTPUT 引脚接在 PIN2 接口 定义irReceiverPin变量为PIN2接口

IRrecv irrecv(irReceiverPin); // 设置irReceiverPin定义的端口为红外信号接收端口

decode\_results results; // 定义results变量为红外结果存放位置

int in3=6;//右轮

int in4=5;

int in1=10;//左轮

int in2=9;

int val0=15;//左轮快，调整差速

int val1=120;//walk

int val2=90;//画三角调整后退时的速度

int val3=150;//画圆时的差速

int val4=250;//画圆时的速度

int val5=180;

int servopin=11; //定义舵机接口数字接口11 也就是舵机的橙色信号线

int servopin\_bo=7;

int angle=40; //舵机抬笔角度

int len;

int wid;

int t=100; //矩形的单位时间

int voice=3; //蜂鸣器

int led=8; //LED灯

int Echo = A1; // Echo回声脚(P2.0)

int Trig =A0; // Trig 触发脚(P2.1)

//--------------------------------蓝牙模式-------------------------------//

void bluetooth()

{

char val;

while(1)

{

val = Serial.read();

if(val!=-1)

{

switch(val)

{

case 'A':

walk(in1,in2);

walk(in3,in4);

delay(50);

break;

case 'B':

walk(in1,in2);

walk(in3,in4);

delay(50);

break;

case 'C':

turnleft();

break;

delay(50);

break;

case 'D':

turnright();

delay(50);

break;

case 'F':

stopper(in1,in2);

stopper(in3,in4);

twinkle();

break;

default:

warn();

}

}

}

}

//--------------------------------蜂鸣器播放音乐-------------------------------//

int length = 15; // the number of notes

char notes[] = "ccggaagffeeddc "; // a space represents a rest

int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };

int tempo = 300;

void playTone(int tone, int duration) {

for (long i = 0; i < duration \* 1000L; i += tone \* 2) {

digitalWrite(voice, HIGH);

delayMicroseconds(tone);

digitalWrite(voice, LOW);

delayMicroseconds(tone);

}

}

void playNote(char note, int duration) {

char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };

int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

// play the tone corresponding to the note name

for (int i = 0; i < 8; i++) {

if (names[i] == note) {

playTone(tones[i], duration);

}

}

}

void playMusic(){

for (int i = 0; i < length; i++) {

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

playNote(notes[i], beats[i] \* tempo);

}

// pause between notes

delay(tempo / 2);

}

}

//--------------------------------控制舵机-------------------------------//

void servopulse(int servopin,int myangle)/\*定义一个脉冲函数，用来模拟方式产生PWM值舵机的范围是0.5MS到2.5MS 1.5MS 占空比是居中周期是20MS\*/

{

pulsewidth=(myangle\*11)+500;//将角度转化为500-2480 的脉宽值 这里的myangle就是0-180度 所以180\*11+50=2480 11是为了换成90度的时候基本就是1.5MS

digitalWrite(servopin,HIGH);//将舵机接口电平置高 90\*11+50=1490uS 就是1.5ms

delayMicroseconds(pulsewidth);//延时脉宽值的微秒数 这里调用的是微秒延时函数

digitalWrite(servopin,LOW);//将舵机接口电平置低

// delay(20-pulsewidth/1000);//延时周期内剩余时间 这里调用的是ms延时函数

delay(20-(pulsewidth\*0.001));//延时周期内剩余时间 这里调用的是ms延时函数

}

void down()

{

for(int i=1;i<=4;i++)

{

servopulse(servopin,0);

delay(1000);

}

}

void up()

{

for(int i=1;i<=3;i++)

{

servopulse(servopin,angle);

delay(1000);

}

}

//--------------------------------小车运动-------------------------------//

void walk(int pin1,int pin2)//当in1=1,in2=0,正转；in1=0,in2=1,反转；

{

digitalWrite(pin1,HIGH);

digitalWrite(pin2,LOW);

if ((pin1==in3)||(pin1==in4))

analogWrite(pin1,val1);

else

analogWrite(pin1,val1-val0);

}

void stepback(int val)

{

walk(in2,in1);

walk(in4,in3);

analogWrite(in2,val);

analogWrite(in4,val);

}

void stopper(int pin1,int pin2)//紧急制动，实际就是将电机两个端短接了

{

digitalWrite(pin1,HIGH);

digitalWrite(pin2,HIGH);

}

void left(int tim)

{

stopper(in1,in2);

walk(in3,in4);

analogWrite(in3,100);

delay(tim\*100);

}

void right(int tim)

{

stopper(in3,in4);

walk(in1,in2);

analogWrite(in1,100);

delay(tim\*100);

}

void turnleft()

{

walk(in3,in4);

walk(in2,in1);

}

void turnright()

{

walk(in1,in2);

walk(in4,in3);

}

//--------------------------------小车画图-------------------------------//

void drawtri()

{

for (int i=1;i<=2;i++)

{

down();

walk(in1,in2);

walk(in3,in4);

//analogWrite(in1,val5-val0);

//analogWrite(in3,val5);

delay(700);

up();

stopper(in1,in2);

stopper(in3,in4);

delay(500);

//walk(in1,in2);

//walk(in3,in4);

//analogWrite(in1,val2-val0);

//analogWrite(in3,val2);

//delay(100);

turnright();

//analogWrite(in1,val5-val0);

//analogWrite(in3,val5);

delay(400);

stepback(val2);

delay(370);

}

down();

walk(in1,in2);

walk(in3,in4);

//analogWrite(in1,val5-val0);

//analogWrite(in3,val5);

delay(700);

up();

stopper(in1,in2);

stopper(in3,in4);

delay(500);

turnright();

//analogWrite(in1,val5-val0);

//analogWrite(in3,val5);

delay(520);

stepback(val2);

delay(370);

}

void drawcir(int val)

{

down();

walk(in1,in2);

walk(in3,in4);

analogWrite(in3,val);

analogWrite(in1,val-val0-val3);

delay(2000); //time needs change

}

void drawrect(int l,int w)

{

down();

walk(in1,in2);

walk(in3,in4);

delay(w\*t);

up();

turnright();

delay(280);

stepback(val1);

delay(200);

down();

walk(in1,in2);

walk(in3,in4);

delay(l\*t);

up();

turnright();

delay(280);

stepback(val1);

delay(200);

down();

walk(in1,in2);

walk(in3,in4);

delay(w\*t);

up();

turnright();

delay(280);

stepback(val1);

delay(200);

down();

walk(in1,in2);

walk(in3,in4);

delay(l\*t);

up();

turnright();

delay(280);

stepback(val1);

delay(200);

}

//--------------------------------红外译码-------------------------------//

int number(int ircode)

{

switch (ircode)

{

case 0xFF6897:

return 1;

case 0xFF9867:

return 2;

case 0xFFB04F:

return 3;

case 0xFF30CF:

return 4;

case 0xFF18E7:

return 5;

case 0xFF7A85:

return 6;

case 0xFF10EF:

return 7;

case 0xFF38C7:

return 8;

case 0xFF5AA5:

return 9;

}

}

//--------------------------------蜂鸣器报警-------------------------------//

void warn()

{

for (int i=1;i<=2;i++)

{

digitalWrite(voice,HIGH);

delay(500);

digitalWrite(voice,LOW);

delay(500);

}

}

//--------------------------------LED灯闪烁-------------------------------//

void twinkle()

{

digitalWrite(led,HIGH);

delay(500);

digitalWrite(led,LOW);

delay(500);

digitalWrite(led,HIGH);

}

//--------------------------------超声波避障-------------------------------//

float 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，这里至少是10μs

delayMicroseconds(10);

digitalWrite(Trig, LOW); // 持续给触发脚低电

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

Fdistance= Fdistance/58; //为什么除以58等于厘米， Y米=（X秒\*344）/2

// X秒=（ 2\*Y米）/344 ==》X秒=0.0058\*Y米 ==》厘米=微秒/58

//Serial.print("Distance:"); //输出距离（单位：厘米）

//Serial.println(Fdistance); //显示距离

//Distance = Fdistance;

return Fdistance;

}

void front\_detection()

{

//此处循环次数减少，为了增加小车遇到障碍物的反应速度

for(int i=0;i<=5;i++) //产生PWM个数，等效延时以保证能转到响应角度

{

//Serial.println("www");

servopulse(servopin\_bo,90);//模拟产生PWM

}

Front\_Distance = Distance\_test();

//Serial.print("Front\_Distance:"); //输出距离（单位：厘米）

// Serial.println(Front\_Distance); //显示距离

//Distance\_display(Front\_Distance);

}

void left\_detection()

{

for(int i=0;i<=15;i++) //产生PWM个数，等效延时以保证能转到响应角度

{

servopulse(servopin\_bo,175);//模拟产生PWM

}

Left\_Distance = Distance\_test();

//Serial.print("Left\_Distance:"); //输出距离（单位：厘米）

//Serial.println(Left\_Distance); //显示距离

}

void right\_detection()

{

for(int i=0;i<=15;i++) //产生PWM个数，等效延时以保证能转到响应角度

{

servopulse(servopin\_bo,5);//模拟产生PWM

}

Right\_Distance = Distance\_test();

//Serial.print("Right\_Distance:"); //输出距离（单位：厘米）

//Serial.println(Right\_Distance); //显示距离

}

void chaoshengbo()

{

while(1)

{

front\_detection();//测量前方距离

if(Front\_Distance < 30)//当遇到障碍物时

{

stepback(val1);//后退减速

delay(200);

stopper(in1,in2);//停下来做测距

stopper(in3,in4);

delay(200);

left\_detection();//测量左边距障碍物距离

// Distance\_display(Left\_Distance);//液晶屏显示距离

right\_detection();//测量右边距障碍物距离

//Distance\_display(Right\_Distance);//液晶屏显示距离

if((Left\_Distance < 30 ) &&( Right\_Distance < 30 ))//当左右两侧均有障碍物靠得比较近

{

turnleft();//旋转掉头

analogWrite(in2,100);

analogWrite(in3,100);

delay(70);

}

else if(Left\_Distance > Right\_Distance)//左边比右边空旷

{

left(3);//左转

stopper(in1,in2);//刹车，稳定方向

stopper(in3,in4);

delay(100);

}

else//右边比左边空旷

{

right(3);//右转

stopper(in1,in2);//刹车，稳定方向

stopper(in3,in4);

delay(100);

}

}

else

{

walk(in1,in2);//无障碍物，直行

walk(in3,in4);

analogWrite(in1,100);

analogWrite(in3,100);

delay(1000);

}

}

}

//--------------------------------主程序-------------------------------//

void setup()

{

Serial.begin(9600);

irrecv.enableIRIn(); // 启动红外解码

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

pinMode(in3,OUTPUT);

pinMode(in4,OUTPUT);

pinMode(voice,OUTPUT);

pinMode(led,OUTPUT);

pinMode(servopin,OUTPUT);

pinMode(servopin\_bo,OUTPUT);

pinMode(Echo, INPUT); // 定义超声波输入脚

pinMode(Trig, OUTPUT); // 定义超声波输出脚

//下面程序开始时让控制端都为高电平，电机保持不动。

//digitalWrite(in1,HIGH);

//digitalWrite(in2,HIGH);

//digitalWrite(in3,HIGH);

//digitalWrite(in4,HIGH);

}

void loop()

{

digitalWrite(led,LOW);

up();

if (irrecv.decode(&results)) { // 解码成功，把数据放入results变量中

Serial.println(results.value,HEX);

switch (results.value)

{

case 0xFF6897: //1

digitalWrite(led,HIGH);

drawtri();

stopper(in1,in2);

stopper(in3,in4);

break;

case 0xFF9867: //2

digitalWrite(led,HIGH);

drawcir(val4);

stopper(in1,in2);

stopper(in3,in4);

break;

case 0xFFB04F: //3

digitalWrite(led,HIGH);

drawrect(6,6);

stopper(in1,in2);

stopper(in3,in4);

break;

case 0xFF30CF: //4

digitalWrite(led,HIGH);

drawrect(15,10);

stopper(in1,in2);

stopper(in3,in4);

break;

case 0xFF18E7: //5

digitalWrite(led,HIGH);

playMusic();

case 0xFF629D: //上

digitalWrite(led,HIGH);

walk(in1,in2);

walk(in3,in4);

break;

case 0xFFA857: //下

digitalWrite(led,HIGH);

stepback(val1);

break;

case 0xFF22DD: //左

digitalWrite(led,HIGH);

turnleft();

break;

case 0xFFC23D: //右

digitalWrite(led,HIGH);

turnright();

break;

case 0xFF02FD: //OK制停

stopper(in1,in2);

stopper(in3,in4);

twinkle();

break;

case 0xFF42BD:

bluetooth();

break;

case 0xFFFFFFFF:

break;

default:

warn();

}

irrecv.resume(); // 继续等待接收下一组信号

}

}