|  |
| --- |
| **人工智能小车** |

**实**

**验**

**报**

**告**

**报告人： 张宇宁**

**学号:2015202036**

1. **实验目的**

人工智能是指由人制造出来的机器所表现出来的智能，被称为“第四次工业革命”，我们要实现的小车基础功能是：

1. 能够自动前进，不原地打转；
2. 遇到障碍物时，能够通过超声波规避障碍物；
3. 能够使用电脑或者手机蓝牙遥控小车，使其完成前进、左右转弯、后退、停止、启动；
4. 能够向电脑或者手机传拍摄的图像以及视频；

在实现了小车基础功能后，我们要实现的小车高级功能是：

1. 通过装人工智能软件，使小车能够沿固定的轨道行驶；
2. 实现声控人工智能小车
3. 图像识别，选用了TSL1401CL线性CCD图像采集模块，该模块采用串行通信方式与主控CPU连接，智能车能根据采集到的图像分析前方路径及障碍而实现智能驾驶。
4. 小车可以自动追踪目标，前方物体跑得慢小车减速，前方物体跑得快小车加速。若前方物体距离小车很近，则停车。
5. **实验步骤**

1、元件准备

UNO板\*1

蓝牙模块\*1

超声波探测器\*1

扩展板\*1

面包板\*1

9V蓄电池\*1

充电宝\*1

亚力克板车架\*1（配有马达、轮胎）

焊接笔\*1

螺丝刀\*2

Mp3扩展板\*1

发光二极管若干

杜邦线若干

2、配置软件环境并导入代码

我们在电脑上下载Arduino id软件，将UNO与电脑相连，将代码导入UNO板。

3、小车拼装

根据老师在课堂上讲解的内容以及拍摄的照片，我们使用杜邦线将各个部分电路连接起来。具体操作如下：

1. 拼装小车

一共有两个马达，三个轮胎，我们使用螺丝钉将轮胎以及马达固定在了小车底盘。

1. 安装UNO板、扩展板以及面包板

我们将扩展板插在UNO板上，使用橡皮筋固定在了车架上的后面。面包板固定在车架上的前面。

1. 安装蓝牙和传感器

我们将蓝牙和传感器错位插在面包板上。

1. 连接电路

蓝牙插件插在面包板上，共有四个引脚，两个引脚用于供电，两外两个用于传输信息。输入输出口与UNO板输出输入口对应，剩余两个引脚与扩展板左上角相连。

超声波探测器插入在面包板上，四个引脚，GND口接到扩展板SER1串口的负极，ECHO口接到扩展板SER1串口的S极，TRIG口扩展板SER1串口的S极，UCC口扩展板SER1串口的正极。

我们将两个马达的正负极与UNO电机驱动扩展板M1、M4相连。

将蓄电池正负极分别与UNO电机驱动扩展板的GND，+M两个口相连接，不要连错。正常供电时，扩展板绿灯亮。

**三、总结反思**

1、我们开始想做四轮驱动小车，后面经过讨论，还是选择两轮的了。两轮小车灵活轻便，在进行避障时效果会好一点。

2、我们考虑到橡皮机固定小车不稳定，所以考虑使用支架固定小车。

3、我们配置环境开始编译时，找不到头文件AFMotor.h下载后放置在同一目录下即可。

4、组装完毕后发现小车没有动，我们开始调试小车。首先查找是不是马达或者电池的问题，我们单独用蓄电池为马达供电，发现马达正常工作，故排除这种可能。再与其他组交流后，发现我们电机驱动板绿灯不亮，所以发现是电机驱动板的问题。我们购置新的电机驱动板，但是目前尚未到货。

5、我们进一步确定是电机驱动板与蓄电池相连的串口坏了，所以用M1、M4串口连接蓄电池，发现小车可以跑。但是蓝牙发烫，并且蓝牙的灯没有亮，我们发现蓝牙被烧坏了。

我们遇到的问题是：小车能够直走，但是有时候由于接触不良以及摩擦，导致两个轮胎转速不同。

6、经过多次实验，我们发现GND和+M口工作不稳定。比如：第一次调试小车时，GND和+M接口不能正常工作。为了解决此问题，我们将两个马达的正负极各引出一根线分别与蓄电池的正负极相连。原理是：电池的正负极与扩展板的GND和+M口相连，扩展板M2和M3接口与马达的正负极相连，电池通过扩展板给马达供电。现在我们直接将电池正负极与两个马达正负极相连，实现电池直接给马达供电。

7、为了使小车显得长一点，我们将两个马达换了方向。调试小车时，我们发现小车倒着走。首先，我们对代码进行修改，将小车初始状态向前改为向后，发现并没有起作用。后来，我们尝试改变电池与马达的连线方式。定义马达上面的接口为a，下面的接口为b，我们开始将电池的正极接马达a口，负极接马达b口，现在我们调换一下顺序，将电池正极接马达b口，负极接马达a口。同时将代码修改回来，发现小车是向前走的，该问题解决。

8、在调试小车避障功能过程中，小车的右轮跑掉了两次，我们将它又安了上去。因为小车的轮子不能通过螺丝固定，小车撞上障碍物时发生碰撞，把轮子碰松了。同时我们发现，固定马达的螺丝钉会与车轮内侧发生摩擦。

9、刚开始小车没有动，经过多次尝试，我们发现新购买的扩展板的M1、M2、M3、M4接口工作不稳定。于是，我们使用二极管对它们进行测试，发现这些接口都不亮。我们又使用二极管对老的扩展板进行测试，发现它的M1、M4接口无法正常工作，M2、M3接口正常工作。所以我们对接线和代码进行调整，将两个马达正负极分别与老扩展板的M2、M3接口相连，并将代码中M1、M4改为M2、M4。通过不断尝试，小车可以前进了。

10、因为新扩展板GND和+M接口无法正常工作，我们使用电池直接给马达供电。通过多次实验，发现有时候小车左侧马达的两极未与电池正负极相连时，右侧马达两极与电池正负极连接后，右轮并没有转动，反而左轮缓慢地转动。我们将左侧马达与电池相连，发现右轮依然没有转动，左轮急速地转动。但是，有时候左右轮正常转动，我们推测是因为连接马达同一极的两根导线接触而形成了回路，使右轮短路而无法正常转动。

11、在测试小车避障功能时，我们发现小车无法识别障碍物。通过反复实验并与其他小组交流，我们发现面包板的前后两部分是不连通的。我们将杜邦线和传感器都插在了前半部分。

12、在测试小车避障功能时，为了方便控制小车，我们通过修改代码，将小车的速度从原来的80降至50。多次实验中，我们发现小车有轻微停顿和小角度转弯，我们猜测是由于小车速度较低使转弯效果不明显。所以，我们将小车的速度提高至60，并增加了转弯的时间。

13、在多次实验中，我们发现杜邦线的接触不良以及有些线是坏的，这对我们造成了困扰，我们通过换线、用胶带固定、焊接等方式进行改善。

14、在修改代码过程中，我们遇到了代码上传无法完成的情况，程序报错。我们对UNO板进行reset操作，重新上传发现还是不可以。后来我们分析后，如果IDE出现avrdude: stk500\_recv(): programmer is not responding的提示的时候，可能是扩展板的数字0和数字1插着东西。如果数码管接到了数字0和数字1上，每次上传代码时，都需要要拔下来，因为数据交互借用了这两个脚。也就是说，如果你接线时用了串行，upload时需要先拔下0,1口的接线，上传完成后再接回去。为了验证我们的想法，我们使用新的扩展板上传代码，新扩展板的0和1接口未进行焊接，发现代码正常上传，所以证实了我们的想法。我们发现自己的UNO开发板串口RX/TX是有其他连接或复用的（蓝牙连接），为了解决此问题，故直接摘下电机驱动扩展板，再次传输，成功。

15、实验过程中，我们发现传感器无法正常工作，经过多次分析和比对后，我们发现rx和tx接错了，正确的接法是rx接tx，我们误将rx与rx连接，tx与tx连接。重新连接后，传感器正常工作。

**四、实验结果**

我们实现了以下功能：自动前进、超声波避障、蓝牙遥控（前进后退左右转停止等）、手机向电脑实时传图、自动追踪目标。完成了图像识别的初步准备工作，生成了测试集和训练集。

我们通过超声波测距，使小车随机转向 0.05 - 1秒。通过蓝牙，发送0，表示人工控制；1，表示自动行驶；2，表示追踪前方物体；人工控制中1表示前进，2表示后退，3表示向右转，4表示向左转，5表示停车；6表示自动控制开始，7表示自动控制结束，8表示避障，9表示自动追踪目标。

自动追踪目标时，最初小车会随机停等 0.1 - 0.5秒，如果判断与前方物体距离太近（小于20），停车；如果前方物体移动速度快（两次测距差值大），则小车加速；如果物体移动速度慢（两次测距差值小），则小车减速。如果是正常情形，小车跟着前方目标直行。

**五、代码**

#include <AFMotor.h>

AF\_DCMotor Rback\_motor(4);

AF\_DCMotor Lback\_motor(1);

float RLRatio = 0.8;

int maxSpeed = 100;//最大速度

char getstr;//串口数据

int IntervalTime;//测距时间

int trig = 9;//sero1保留端口

int echo = 10;//sero2保留端口

int control=1;//0:人工控制;1:自动控制;2:追踪正前方物体

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

Rback\_motor.setSpeed(0);

Rback\_motor.run(RELEASE);

Lback\_motor.setSpeed(0);

Lback\_motor.run(RELEASE);

delay(2000);

//超声波测距

pinMode(echo, INPUT);

pinMode(trig, OUTPUT);

}

void loop() {

getstr = Serial.read();

driver();

if(control==1){

ultrasonicCar();

}

else if(control==2){

follow();

}

}

void forward() {

Rback\_motor.run(BACKWARD);

Lback\_motor.run(BACKWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

}

void backward() {

Rback\_motor.run(FORWARD);

Lback\_motor.run(FORWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

}

void stopcar() {

Rback\_motor.setSpeed(0);

Lback\_motor.setSpeed(0);

Rback\_motor.run(RELEASE);

Lback\_motor.run(RELEASE);

}

void left(){

Rback\_motor.run(FORWARD);

Lback\_motor.run(BACKWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

}

void right(){

Rback\_motor.run(BACKWARD);

Lback\_motor.run(FORWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

}

void driver() {

if ((getstr == '-')&&(RLRatio>0.1)) {

RLRatio -= 0.1;

Serial.println("decrease right");

Serial.println(RLRatio);

}

if (getstr == '+') {

RLRatio += 0.1;

Serial.println("increase right");

Serial.println(RLRatio);

}

if (getstr == '5') {

Serial.println("stopcar");

stopcar();

control=0;

}

if (getstr == '1') {

Serial.println("forward");

forward();

control=0;

}

if (getstr == '2') {

Serial.println("backward");

backward();

control=0;

}

if (getstr == '3') {

Serial.println("right");

right();

control=0;

}

if (getstr == '4') {

Serial.println("left");

left();

control=0;

}

if (getstr=='6'){

control=1;

Serial.println("Auto Control On");

}

if(getstr=='7'){

control=0;

Serial.println("Auto Control Off");

}

if(getstr=='8'){

float s;

s = getdistance();

Serial.println(s);

}

if(getstr=='9'){

control=2;

Serial.println("follow up....");

}

}

float getdistance() {

digitalWrite(trig, 1);

delayMicroseconds(15);

digitalWrite(trig, 0);

IntervalTime = pulseIn(echo, HIGH);

return IntervalTime / 58.00;

}

void ultrasonicCar() {

float s = 0;

s = getdistance();

ultrasonicCar(s);

}

void follow() {

//跟踪正前方物体

float s1 = 0;

float s2 = 0;

s1 = getdistance();

long n = random(100,500);//随机停等 0.1 - 0.5秒

delay(n);

Rback\_motor.setSpeed(0);

Lback\_motor.setSpeed(0);

s2 = getdistance();

// 距离太近，停车

if(s2<20){

stopcar();

}

// 物体向前方移动，距离变化很大，加速

else if(s2-s1>=60){

if(maxSpeed<150){

maxSpeed+=5;

}

forward();

}

// 物体向前方移动，距离变化很小，减速

else if(s2-s1<10){

if(maxSpeed>=50){

maxSpeed-=5;

}

forward();

}

// 正常情形，跟着前方目标直行就可以了

else{

maxSpeed=100;

forward();

}

}

void ultrasonicCar(float s) {

if ((s < 40)&&(s>0)){//20cm以内

randomright();

}

else{

forward();

}

}

void randomright(){

Rback\_motor.run(BACKWARD);

Lback\_motor.run(FORWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

long n = random(50,1000);//随机转向 0.05 - 1秒

delay(n);

Rback\_motor.setSpeed(0);

Lback\_motor.setSpeed(0);

}

void randomleft(){

Rback\_motor.run(FORWARD);

Lback\_motor.run(BACKWARD);

Rback\_motor.setSpeed(maxSpeed\*RLRatio);

Lback\_motor.setSpeed(maxSpeed);

long n = random(50,1000);//随机转向 0.05 - 1秒

delay(n);

Rback\_motor.setSpeed(0);

Lback\_motor.setSpeed(0);

}