第十二届江苏省大学生机器人大赛

无人快递车—智慧快递

技术报告

二〇二一年十一月

**前言**

机器人竞赛是一项体育与高科技结合的对抗项目，涉及机械电子、智能控制、计算机技术、人工智能等多种学科和研究领域，是培养信息、自动化科技人才，展示高科技成果，促进实用化和产业化的新途径。各类机器人大赛的举办，对于普及机器人科学技术，促进人工智能与机器人技术的研究和应用都将产生重要推动作用。

经过几年的不懈努力和各方面的大力支持，江苏省的大学生机器人水平已走在全国前列，在中国机器人大赛、CCTV 杯电视机器人大赛、亚太机器人挑战赛、ROBOTCUP 足球世界杯的国内外机器人大赛中屡创佳绩。

1. **方案论证**

根据设计要求，本系统主要由寻迹传感器模块、超声波避障模块、控制器模块、直流电机及其驱动模块等模块构成。

**1.1车体**

车子为四轮车，共四个电机，同侧电机保持轮子同步运转，一个正转，一个反转，实现转弯效果。为了能控制车轮的转速，可以采取代码内部调速机制。左右轮两个电机转速的配合可以实现小车的前进、倒退、转弯等功能。小车车底偏低，保持重心平稳，可以避免出现后轮过低而使两侧驱动轮动力不够的情况。对于车架材料的选择，绝大数选择了铝板。用有铝板的车架比塑料车架更加牢固，比铁制小车更轻便，美观。

**1.2传感器的安装**

**1.2.1 寻迹原理及选择**

红外传感器巡线的基本原理是利用物体的反射性质，我们本次实验是巡黑线行驶，当红外线发射到黑线上时会被黑线吸收掉，发射到其他的颜色的材料上会有反射到红外的接受管上。我们根据这点的不同写相应的代码完成小车巡线功能。我们本次实验采用的是四路红外传感器分别连接在Arduino主控板上的A1，A2，A3，A4口上。其中中间两路巡线是一直在黑线上，小车会直行，当任意一个出来，则小车会自动纠正，如果最外面的检测到黑线，则小车以更大速度纠正到正确黑线上面。

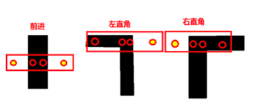


图1 处理直线、直角的传感器状态分析

**1.2.2 循迹传感器的安装**

根据比赛的规则，为了能达到比较好的寻迹效果，我们用7路红外传感器配合小车底下左右两个灰度传感器进行循迹,红外传感器只使用右边6路传感器：

A picture containing indoor

Description automatically generated

图2-1 红外传感器安装示意图

A picture containing electronics

Description automatically generated

图2-2 灰度传感器安装示意图

**1.3控制器模块**

采用[Arduino](https://baike.baidu.com/item/Arduino" \t "_blank)核心板作为主控制器，是一款便捷灵活、方便上手的开源电子原型平台。它构建于开放原始码simple I/O介面版，并且具有使用类似Java、C语言的Processing/Wiring开发环境。主要包含两个主要的部分：硬件部分是可以用来做电路连接的Arduino电路板；另外一个则是Arduino IDE，你的计算机中的程序开发环境。你只要在IDE中编写程序代码，将程序上传到Arduino电路板后，程序便会告诉Arduino电路板要做些什么了。



图3 Arduino主控板电路图

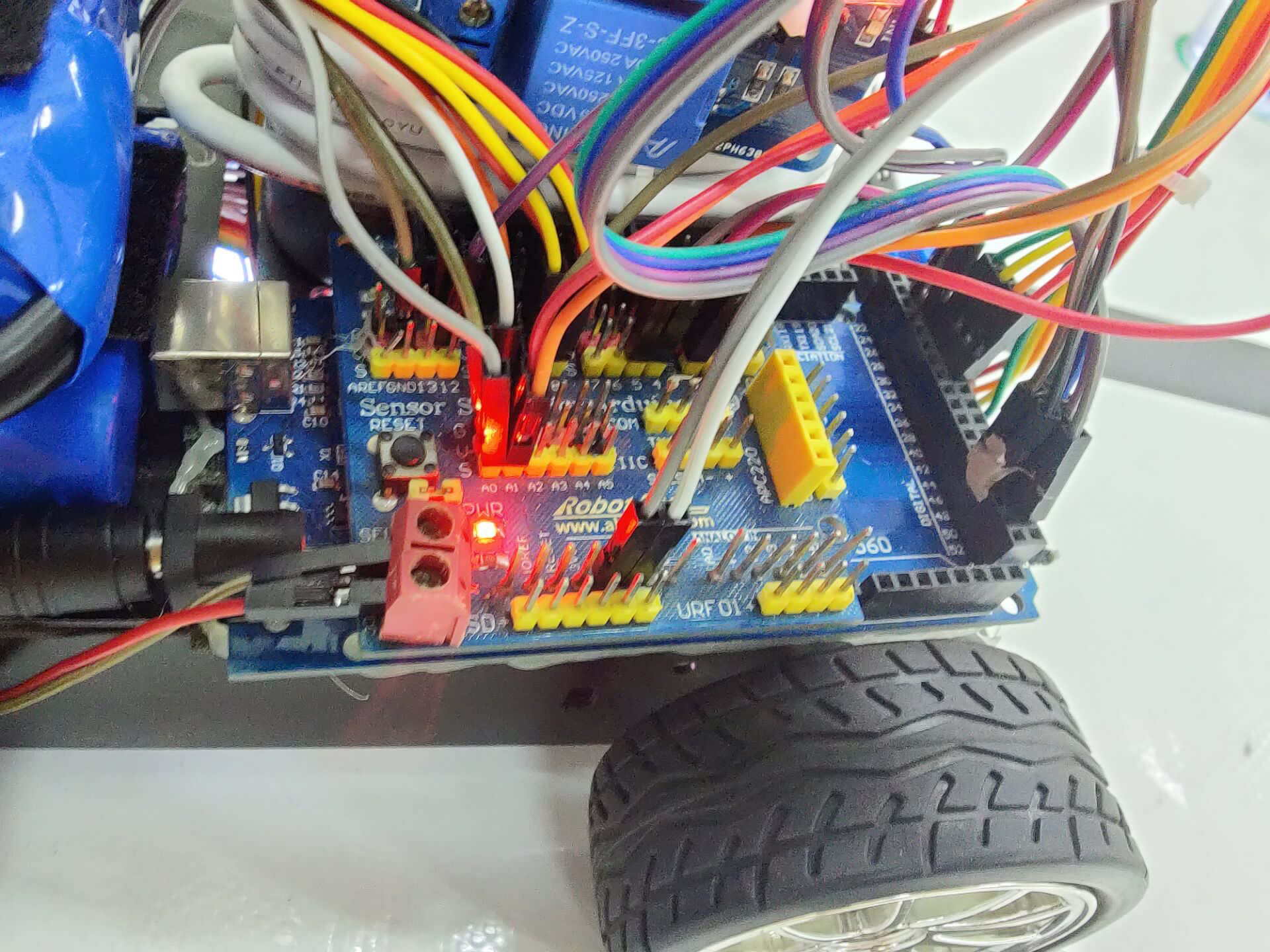


图4 Mega 2560板以及拓展板安装图

**1.4 电源模块**

采用2对3节3.7V锂电池分别给单片机系统、电机、电动推杆和舵机供电。

**1.5 升降台模块**

利用电动推杆（行程150mm，推力150N，空载速度为60mm/s）置顶起一端，打破平衡，实现运送货物的功能。



图5 升降台设计图

**1.6 超声波模块**

SR04是利用超声波特性检测距离的传感器。其带有两个超声波探头，分别用作发射和接收超声波。其测量的范围是3-500cm。

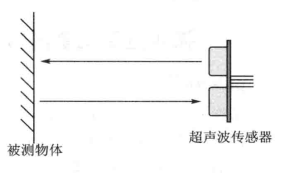


图6 超声波发射和接收示意图 图7 超声波发射和接收示意图

****

图8 超声波原理图

一旦检测到有回波信号则ECHO引脚会输出高电平。高电平持续的时间就是超声波从发射到返回的时间。此时可以使用pulseIn（）函数获取测距的结果，并计算出距被测物体的实际距离。公式: 距离=高电平时间\*声速(340M/S)/2。

1. **实验工作及数据分析**

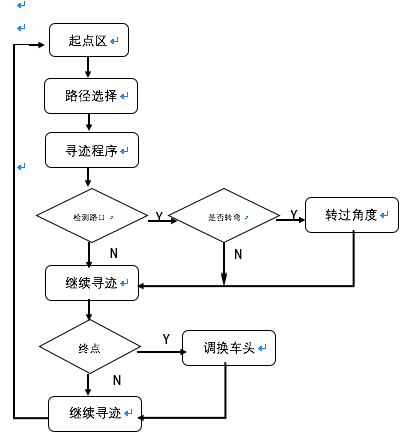
**1. 软件开发工具以及所使用的应用软件**

开发工具：程序编译软件：Arduino

应用软件：电路原理图及 PCB 制作：PROTEL DXP 2004

机械结构图绘制：Auto CAD 2014

**2.系统软件流程图**



**3.程序**

底层程序：

void loop()

{

Serial.print("小车当前位置x1 = ");

Serial.print(x1);

Serial.print(" y1 = ");

Serial.println(y1);

// 得到小车当前的时间

gnTmCurr = millis() ; //当前时间，单位毫秒

// 得到前面四个传感器的状态

TrackTest();

// 小车是否停止

if ( !gbRun ) return ;

//判断小车是否走完一圈

// 小车位于终点且跑完半程 || 小车位于起点且跑完全程

if ( (x1 == x9 && y1 == y9 && gnLoop % 2 == 0 ) || (x1 == x0 && y1 == y0 && gnLoop % 2 == 1) ) {

if ( gnLoop == 0 )setPath( x1 , y1 );

Serial.println("到达终点或起点");

// 下面的if-else控制小车到达终点或起点时,小车改变方向,小车背对箱子(终点)或者小车朝向北方(起点)

if (gbInitPos ) //停放位置，0左向右1右向左

{

Serial.println("开始左转");

MyRun(0, 0);

MyRun(-SPEED\_MAX + 50, SPEED\_MAX - 50);

control\_get\_lost();

MyRun(-SPEED\_MAX + 50, SPEED\_MAX - 50);

delay(100); // !

MyRun(0, 0);

}

else

{

Serial.println("开始右转");

MyRun(0, 0);

MyRun(SPEED\_MAX - 50, -SPEED\_MAX + 50);

control\_get\_lost();

MyRun(SPEED\_MAX - 50, -SPEED\_MAX + 50);

delay(100); // !

MyRun(0, 0);

}

x2 = x1;

y2 = y1;

brake(0);

if ( gnLoop % 2 == 0 ) //终点

{

MyRun(0, 0);

MyRun(-gnbackSpeed, -gnbackSpeed);

Serial.println("后退投喂");

delay(300);

MyRun(0, 0);

gnlastTime = millis() ;

// 投喂物品

food();

}

else//起点

{

MyRun(0, 0);

MyRun(-gnbackSpeed, -gnbackSpeed);

brake(0);

LedOn(LED\_RIGHT);

delay(1500);

LedOff(LED\_RIGHT);

Serial.println("到达起点");

//获取当前时间

gnlastTime = millis() ;

}

// 如果跑完全程

if ( gnLoop % 2 == 0 ) gnRunDir = 2;

else gnRunDir = 4;

// 小车完成一次半程行驶

gnLoop++;

}//if ( (x1 == x9 && y1 == y9 && gnLoop % 2 == 0 ) || x1 == x0 && y1 == y0 && gnLoop % 2 == 1 ) { //说明跑到终点位置

//前面这个if语句说明小车已经到了终点或者起点并且怎么处理

//这个表示小车的当前点已经到达了目标点，则需要去重新计算下一步的目标点

//MyLoop

if ( x1 == x2 && y1 == y2) {

TrackTest();

//检测5-25之间是否有障碍物，并且躲避掉

if (gnLoop <= 1) {

int num = 5;

while (num--) {

int dis = NewDistanceTest();

nDire = test\_hinder(dis);

if (nDire) break;

delay(5);

}

nDire == 0?Serial.println("前方无障碍"):Serial.println("前方有障碍");

}

//判断小车下一步要行动的方向

// 小车到达终点且跑完半程且小车初始左右方向行驶

if ( x1 == x9 && y1 == y9 && gnLoop % 2 == 1 && gbDireReturn == 0 ) {

// gbInitPos：停放位置，0：左向右 1：右向左

if (gbInitPos ) {

nDire = 2;

}

else {

nDire = -2;

}

}

else nDire = next\_direction();

Serial.print("nDire: ");

Serial.println(nDire);

//获得到小车下一步要走的点坐标

get\_next\_point(nDire);

Serial.print("小车下一个行驶到的点next point: x = ");

Serial.print(x2);

Serial.print(" y = ");

Serial.println(y2);

//记录一下原来的方向

nDirePre = gnRunDir;

//顺时针，右转，方向+

//逆时针，左转，方向-

//获得到改变后的方向

gnRunDir = get\_change\_direction(gnRunDir);

//得到小车前进，后退，左转，右转的标志 左转-1 右转1 前进 2 后退-2

carStatus = 0;

//拿到小车下一步的转向

carStatus = get\_car\_turn\_status(gnRunDir, nDirePre);

Serial.print("小车下一步的转向gnRunDir:");

Serial.println(gnRunDir);

Serial.print("小车状态carStatus:");

Serial.println(carStatus);

//小车调整一下转个向

if ( carStatus == -1) {//小车左转

Serial.println("小车左转");

LedOn(LED\_LEFT);

MyRun(0, 0);

MyRun(-SPEED\_LOW, SPEED\_LOW);

//保证转弯能有90度

control\_get\_lost();

MyRun(0, 0);

LedOff(LED\_LEFT);

} else if (carStatus == 1) { //小车右转

Serial.println("小车右转");

LedOn(LED\_TEST);

MyRun(0, 0);

MyRun(SPEED\_LOW, -SPEED\_LOW);

//保证转弯能有90度

control\_get\_lost();

MyRun(0, 0);

LedOff(LED\_TEST);

}

}//endif ( x1 == x2 && y1 == y2 )

if (carStatus == -2) { //小车后退

Serial.println("小车后退");

MyRun(0, 0);

MyRun(-gnbackSpeed, -gnbackSpeed);

//如果小车后退到下一个一个节点，也就是底下的传感器对到下面的黑线

while (1) {

TrackTest();

if (!TrackValue(-4) || !TrackValue(4)) {

x2 = x1;

y2 = y1;

break;

}

}

MyRun(0, 0);

}

// 如果小车当前点的坐标和下一个点的坐标不相等，则开始向下一个点移动

// 如果小车的横纵坐标不相等，则一直在里面循环

// 检测小车检测是否出错

is\_no\_node();

Serial.println("前往下一个点");

while ( x2 != x1 || y2 != y1 ) //找到下一个点

{

make\_car\_center\_1();

MyRun(SPEED\_ROTARY, SPEED\_ROTARY);

TrackTest();

//细微的调整小车走直线

smallChangeDirection();

//设置状态

TrackTest();

if ( TrackValue(3) && TrackValue(-3) && TrackValue(2) && TrackValue(-2) && !TrackValue(1) && !TrackValue(-1) && car\_status1 == 1) {

car\_status1 = 0;

}

//检测障碍物

if (!((x2 == 4 && y2 == 7) || (x2 == 4 && y2 == 0))) {//不是终点

Identifying\_junctions();

int dis1 = NewDistanceTest();

// ?

if ( dis1 > 15 && dis1 <= 25 && Identifying\_junctions()) //距离小于一定值,则需要后退

{

LedOn(LED\_LEFT);

LedOn(LED\_RIGHT);

car\_status1 = 1;

Serial.println("前方有路障，刹车");

MyRun(0, 0);

//说明不是个障碍物

setBarrier( x2, y2 );

//是个障碍物

//先判断小车的下面是否在黑线上，如果在，则小车不后退，

//继续判断方向

//小车后退

delay(100);

MyRun( -gnbackSpeed, -gnbackSpeed);

//目标位置设为障碍

while (1) {

//如果底下两个传感器有在黑线上，则停止后退

TrackTest();

if (!TrackValue(-4) || !TrackValue(4)) break;

}

MyRun(0, 0);

delay(100);

LedOff(LED\_LEFT);

LedOff(LED\_RIGHT);

//并且还原点坐标

x2 = x1;

y2 = y1;

//因为重新回到以前的点，所以需要重新计算下一个点,跳出此while循环

Serial.println("小车重新回到以前的点");

break;

}//如果是非路

}

Identifying\_junctions();

//如果前面没有障碍物，小车则向前继续前行，直到，小车达到下一个位置为止,小车底部的传感器

//达到黑线，则表示点坐标到了

}

//保证小车可以到终点或者起点

is\_car\_end();

//保证小车如果此时的点坐标是边缘，但是实际上并没有到边缘，该怎么办

check\_border();

car\_status1 = 1;

}

**第三章** **参考文献**

[1]邵贝贝.单片机嵌入式应用的在线开发方法[M].北京:清华大学出版社，2004.10.

[2]张崇巍,李汉强.运动控制系统[M].武汉:武汉理工大学出版社，2002.

[3]肖玲妮,袁增贵.Protel 99SE 印刷电路板设计教程[M].北京:清华大学出版社，2004.

[4]吴守箴,藏英杰著.电气传动的脉宽调制控制技术[M].北京:机械工业出版社，1997.