**小车编队组网与协同工作若干问题研究**

**阶段研究报告**

[一、选题背景 3](#_Toc510405216)

[二、项目计划达到的目标和内容 3](#_Toc510405217)

[*（一）组网与操控* 3](#_Toc510405218)

[*（二）编队协同* 3](#_Toc510405219)

[三、实施进程 4](#_Toc510405220)

[四、成果内容 5](#_Toc510405221)

[*（一）应用层：用户操作平台* 5](#_Toc510405222)

[1.1数据交互模式 5](#_Toc510405223)

[1.2提供的功能 5](#_Toc510405224)

[*（二）从基础的通讯指令集到ABC语言* 10](#_Toc510405225)

[2.1基础的通讯指令集 10](#_Toc510405226)

[2.2加入循环、选择、判断和记录指令的通讯指令集 12](#_Toc510405227)

[2.3扩展的通讯指令集的意义与应用 15](#_Toc510405228)

[2.4高级语言设计：ABC语言雏形（下一步重点研究） 18](#_Toc510405229)

[*（三）网络通讯层：通讯与组网* 19](#_Toc510405230)

[3.1硬件层 19](#_Toc510405231)

[3.2网络层 22](#_Toc510405232)

[*（四）硬件层：小车向上提供的功能* 23](#_Toc510405233)

[4.1硬件材料使用型号说明 23](#_Toc510405234)

[4.2基本运动控制、误差控制 23](#_Toc510405235)

[4.3定位 26](#_Toc510405236)

[4.4障碍物检测 27](#_Toc510405237)

[四、创新点 28](#_Toc510405238)

[五、实施过程中的得失成败 29](#_Toc510405239)

# 一、选题背景

我们的项目主要以4种场景作为应用背景：巡逻、地图绘制、区域搜索、无人车辆交通。

巡逻是指使用无人车进行指定路线或区域的异常监测，这种异常可能是火情（或是可能诱发火情的东西）、某种气体浓度超标、禁入区中有人进入、发生犯罪或是某一处发生了破坏等。这在未来的智能城市中可能会被应用于无人化的监测体系。地图绘制是指对于一片指定区域，标记障碍物的分布情况，绘制出可以出入的路线。这对于一些难以由人进入的区域或者是在发生灾难时一些可能经常发生动态变化的区域比较适合由无人小车来完成。在现实中也经常会发生人或物失踪的情况，这时如果动用无人车进行搜索便可以节省许多人力和时间。同时在无人车辆也会在不久的未来越来越多的出现在我们的身边，这其中可能有客运也可能有货运。

上述的这些任务显然不会仅仅只由一辆小车完成。在巡逻任务中，多辆小车可能分开巡逻增加异常事件的响应与巡逻的范围，也可能搭载不同设备共同巡逻扩展单辆小车的能力；在地图绘制中，多辆小车共同协作不仅能增加地图绘制的效率还能提升信度；在区域搜索中，多辆小车协同工作能缩短搜索时间。

# 二、项目计划达到的目标和内容

## *（一）组网与操控*

这要求任意辆小车都可以进行组网，在实际中使用若干辆小车进行实际的组网实现。小车之间可以进行的可靠的信息传递。

对小车使用自行设计的ABC（Arduino-Based Car Language）语言进行较为复杂的逻辑策略控制（围绕巡逻、搜索、无人交通和地形勘测的应用背景）。

另外开发一个可以操控、调试、模拟小车的平台，这个平台需要能够可视化的显示小车的运动和障碍物的位置分布，另外这个平台需要提供从ABC语言到小车可执行指令集之间的编译。

## *（二）编队协同*

若干辆小车之间要围绕巡逻、搜索、无人交通和地形勘测的应用背景进行一些能够被ABC语言表达的协同工作，这同时要求了需要扩展ABC语言的表达能力。

# 三、实施进程

2017年 11月：

学习使用HMC5883L电子罗盘模块，尝试使用它来进行自动的误差校准（失败）。

学习使用光电编码器，在直行的校准上取得了一定的效果。

学习使用GPS模块，我们决定将项目的重点不再放在误差的自动校准上。

学习使用HC-SR04超声波传感器和舵机。

2017年12月：

完善通讯指令集在Arduino主板中的实现，使得小车可以封装的向上提供接口。使用HC-05蓝牙与继电器模块完成3辆小车通过蓝牙的跳连组网通讯。学习使用各种液晶显示屏模块，其中使用OLED12864液晶模块收集超声波传感器数据进行了简单的人机交互图形化显示（后放弃）。

我们决定暂时放弃蓝牙改用WIFI来进行组网。

2018年1-2月：

烧写ESP8266WIFI模块使之成为TCP客户端，使用java搭建TCP服务端使两者之间能够通讯，进一步的使用多线程技术使得服务端可以与多个客户端通讯，并实现任意客户端之间的通讯。使ESP8266WIFI模块可以与Arduino主板进行稳定通信从而使得服务端可以抽象的直接与安置在小车上的Arduino主板直接通信。最后使用滑动窗口模型实现消息传递过程中的确认机制并规定通讯指令的指令头部从而实现任意辆小车通过WIFI进行组网可靠通讯。

2018年3月：

搭建小车操作平台，数据库设计与建表，可显示小车状态、可调式小车参数、可直接与小车通讯、可直接与小车通讯、可显示小车位置与障碍物分布的模拟图。

将通讯指令集加入循环、选择、判断控制指令，ABC（Arduino-Based Car Language）语言显现雏形。

# 四、成果内容

## *（一）应用层：用户操作平台*

### 1.1数据交互模式

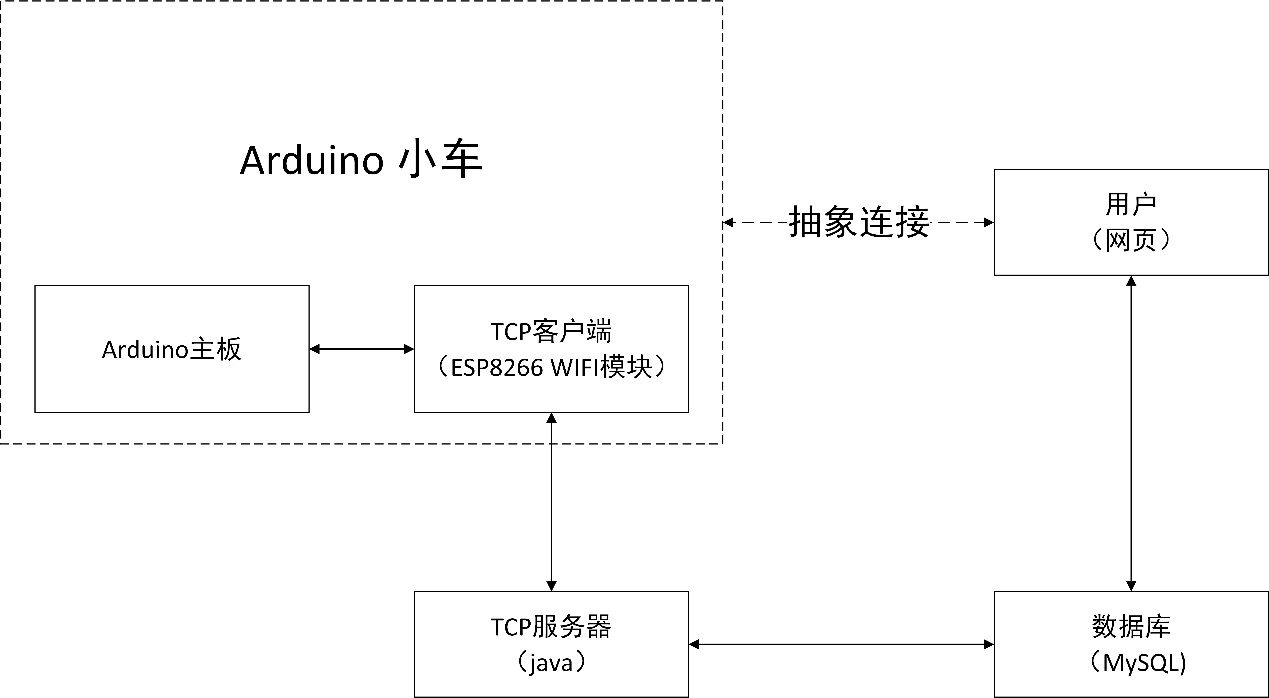


图 1

### 1.2提供的功能

#### 1.2.1直接与小车通讯

当一辆小车与服务器建立连接后，服务器首先会询问其编号car\_id，当小车回复其car\_id后，便正式建立了连接。在小车与服务器正式连接之后，所有服务器待发送的消息会存储在raw\_send这张表中，其中表中存储了发送方的编号、接收方的编号和该发送方和该接收方之间的消息编号，status表明该消息的状态，为0表示未发送，为1表示已发送但未收到确认，为2表示已发送且已收到确认。

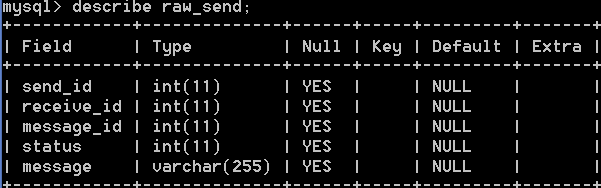


图 2

每个子线程的服务器都会拥有一个定时器程序，它每隔一定时间就会查看raw\_send中是有应该发送给与它连接的小车的还未确认的消息，如果有就发出去。同时这个服务器会不停的接收消息，如果收到了确认帧，那么就会把raw\_send中该消息的状态改为已发送已确认。

用户可以使用网页中提供的接口向小车发送消息，或

我们使用telnet工具模拟1号小车与服务器建立连接，当我们给服务器发送数据时，会在网页中显示接收到的消息，同时使用网页中的接口，可以给小车发送消息，如图显示模拟的1号小车收到了一则消息。

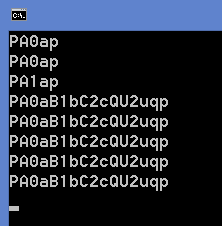


图 3

由于这时我们迟迟没有回复，因此服务器在不断重发这段消息，同时网页上显示该消息的状态为已发送、未确认。

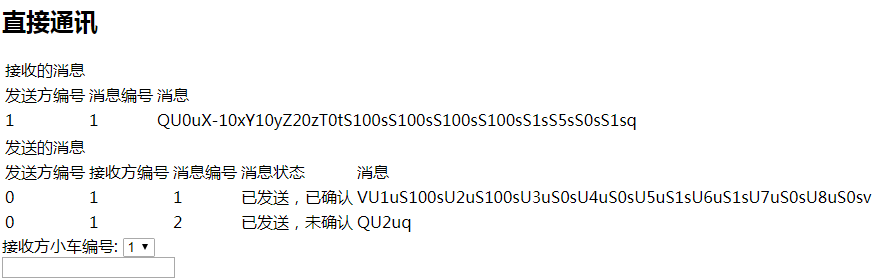


图 4

现在模拟的1号小车给出确认。

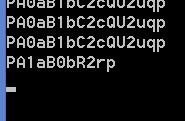
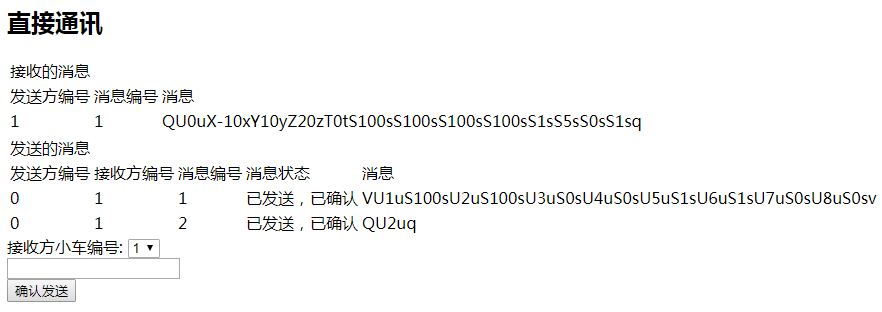


图 5

刷新网页可以看到，消息状态变为了已发送、已确认。



#### 1.2.2提供调试小车的接口

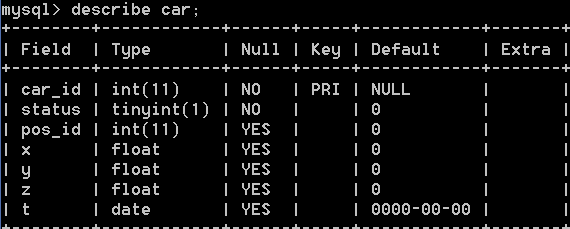


图 6

由于目前我们总共搭建了3辆小车，所以数据库中就存入了3辆小车的数据。

没有小车连接时可以看到小车状态均为未连接。

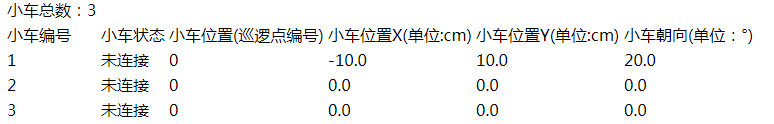


图 7

现在我们使用telnet模拟2号小车与服务器建立连接。

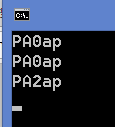


图 8

然后可以看到网页上显示2号小车已连接。

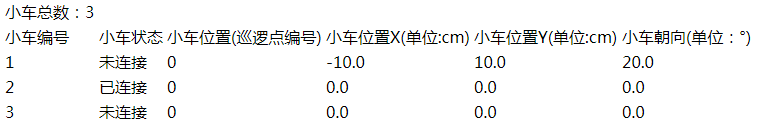


图 9

小车详细参数中提供了直接修改小车参数的接口，同时可以更新小车的参数（包括位置、方向、开机时间）

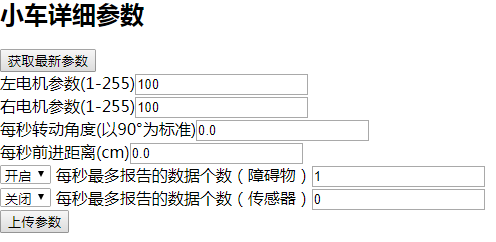


图 10

这里的参数存储在了car\_detail这个表中，update\_status表明了当前存储在表中的信息是否应当被更新到小车，或者是否应当被更新到网页。当一辆小车断连时，这个值会被置为0，表明下一次连接时服务器会重新给小车发送设置这些参数的命令。

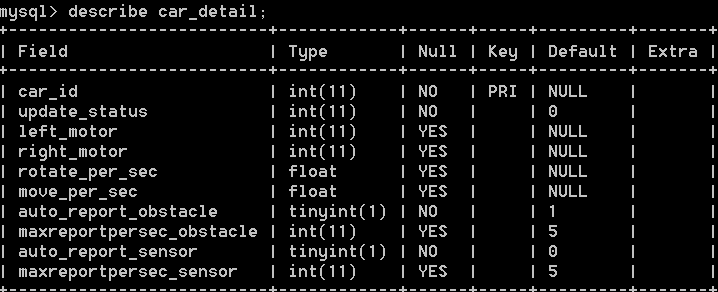


图 11

#### 1.2.3快速操作

网页还提供了快速操作小车的接口，使得平时调试使用小车更加方便。



图 12

#### 1.2.4小车位置与障碍物分布模拟图

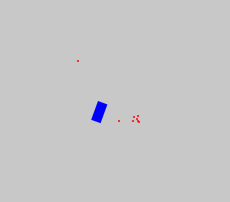


图 13

网页上提供了展示小车位置和障碍物分布的模拟图，根据小车返回的障碍物数据，存在obstacle表中，网页上使用<svg>标签来绘制这些障碍物的位置（用红点表示），小车则用一个蓝色的矩形表示。网页中长宽的1个单位对应1cm。

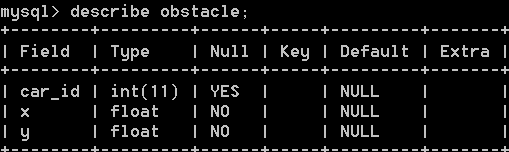


图 14

## *（二）从基础的通讯指令集到ABC语言*

### 2.1基础的通讯指令集

我们设计了一套通讯的指令集来控制小车，这其中主要包括了指令头部和指令主体部分。

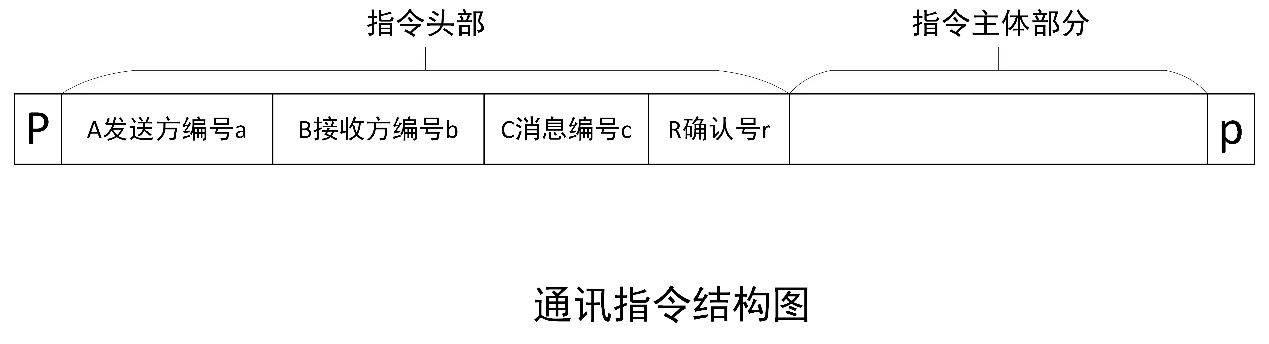


图 15

指令的内容主要包括了对小车基本运动的描述（距离、时间）、调用其传感器返回数值（超声波传感器与其他传感器）、指定小车之间的通讯、修改小车参数、询问小车状态。

通讯指令集的规则如下：

（a）P开始p结束界定了一则指令

（b）指令头：指令头必须包含A，可以不包含B，但是只有B存在时，C才有效，C有效时，指令主体部分才有效

（c）确认帧：基于当前的实现，如果指令头中含有R，那么不可以存在C，且指令主体部分无效。

（d）指令主体部分：

一条合法的指令需要符合以下的2条规则：

规则1 合法的指令以大写字母开始，（大写字母的）小写字母结束。

规则2 合法的指令除去开始的大写字母与结束的小写字母之后，要么是若干合法的指令，要么为空，要么是数字。



图 16

在设计上这种指令是可以多层嵌套的，但是在当前的使用与实现中，很少会多层嵌套的必要（不过在稍后的扩展指令集中，会更多的使用到这种指令的嵌套）

另外对于理论上当中内部的标志字母能否与外部的重复，我们还没有深入研究，所以在实际的实现中，我们选取了XYZUST这6个作为内部参数，其他的指令标志则避免发生重复。

|  |  |  |  |
| --- | --- | --- | --- |
| 指令标志 | 可选参数 | 含义 | 备注 |
| A |  | 消息发送者编号 | 指令头部  0为用户 |
| B |  | 消息接收者编号 | 指令头部  0为用户 |
| C |  | 消息编号 | 指令头部 |
| D | U=0前进/后退S/T距离时间；  U=1旋转Z角度；  U=2指定目标点XY(Z)；  U=3停止(S)(T)信号/时间； | 基本运动控制 | U=2在指定完目标点后会继续执行之后的指令 |
| E |  | 记录指令 | 扩展的指令 |
| F |  | 循环控制指令 | 扩展的指令 |
| G |  |  | 暂时未用  可能后续作为编队控制指令 |
| H | U=0绝对位置XY  U=1相对位置XYZS | 障碍物指令 | 扩展的Q指令含有其功能 |
| I |  | 选择控制指令 | 扩展的指令 |
| J |  | 判断指令 | 扩展的指令 |
| K |  | 车载传感器参数 | 扩展的Q指令含有其内容 |
| L |  |  |  |
| M |  |  |  |
| N |  |  |  |
| O | U指定接收方编号 | 通讯指令 | 当前实现为默认让目标小车前往当前小车位置 |
| P |  | 指令开始结束符 |  |
| Q | 基础的通讯指令集：  U=0表示答XYZT为固定答案S扩展参数  U=1询问XY位置Z方向T开机时间  U=2在1的基础上详细询问其他参数  扩展的通讯指令集：  T=0 答  U-U-S 大类编号-小类编号-值  T=1 问  U-U 大类编号-小类编号 | 查询指令 |  |
| R |  | 确认的消息号 |  |
| S |  |  | 内部参数 |
| T |  |  | 内部参数 |
| U |  |  | 内部参数 |
| V | U=1-8 分别表示小车的8个参数 | 修改小车参数 |  |
| W |  |  |  |
| X |  |  | 内部参数 |
| Y |  |  | 内部参数 |
| Z |  |  | 内部参数 |

### 2.2加入循环、选择、判断和记录指令的通讯指令集

#### 2.2.1选择指令

标志I

选择指令格式如下

IJ判断指令内容j选择指令的执行内容i

若J判断为真则执行I的内容，否则跳过I

应用举例：

I

J

U2uU0uT1tS100s

j

DU3ud

i

I

J

U2uU0ut2tS100s

j

DU0US100sd

i

上述指令的意思也就是

IJ正前方障碍物小于1米j停止Iij正前方障碍物大于1米j前进1米i

执行的效果就是：

如果正前方障碍物小于1米就停下，否则就再前进1米

选择指令一定要和判断指令结合使用才有意义。

ABC语言逻辑形式：

If(Obstacle.front<100){

brake();

}

If (Obstacle.front>100){

move(100);

}

#### 2.2.2判断指令

标志J

将对特定对象进行考察（周围障碍物情况/当前是否有小车与之连接/小车本身位置/传感器参数/是否收到特定信号量）

多个判定内容写在J中是与的关系，同一个I之中多个J是或的关系

第1个U表示考察的对象大类编号（全体存储在小车中的参数），第2个U表示考察的对象小类编号，T表示考察的方式（0等于1小于2大于3小于等于4大于等于）S为考察对象的数值

下面写成表格（大类参数-小类参数-返回值-含义）

U=0 小车参数（位置、开机时间、8个详细参数共计12个参数）

U=1 障碍物数据 （第2个U表示方向获得内容为该方向障碍物的距离）

第2个U的范围不能超过舵机允许的超声波传感器测量角度范围

U=2 传感器数据 （第2个U直接写0当前传感器的数据）

U=3 通讯情况（第2个U表示与小车连接的总数、以及某一编号的小车是否与之连接）

0 连接的小车数量总数（若为-1则表示没有与服务器连接，≥0表示与服务器连接）

1 2 3 与其他小车连接情况（自身恒为1）

U=4 记录数据 （指定具体记录变量号）

#### 2.2.3循环指令

F..f中为循环执行的内容

S表示循环次数（若S为空则为死循环）

U表示提前跳出当前循环，这一般要与选择控制指令结合使用才有意义。

举例：

在前面的例子上加上循环指令

F

Ss

I

J

U2uU0uT1tS100s

j

DU3ud

i

I

J

U2uU0uT2tS100s

j

DU0US100sd

i

f

就变成了小车不断前进（每次前进1米），直到障碍物离它距离小于了1米，当然如果障碍物又远离了超过1米，则再次前进

如果不考虑其他复杂因素，这可以作为直线运动上的小车尾随其他小车或移动物体的简单策略。

ABC语言逻辑形式：

while(true){

If(Obstacle.front<100){

brake();

}

If (Obstacle.front>100){

move(100);

}

}

#### 2.2.4记录指令

标志E

记录指令U的选择与J中一致，T=0表示存，T=1表示读，S表示记录到的变量编号

比如ET0tU0uU0uS1seET0tU0uU1uS2se表示记录当前的XY坐标到变量1和2中

而ET1tU0uU0uS1seET1tU0uU1uS2se表示读取先前存储在变量1和2中的XY坐标

使用指令

D

U2u

X

ET1tU0uU0uS0se

x

Y

ET1tU0uU1uS1se

y

d

表示小车前往先前存储的XY坐标

ABC设计形式

int posx=Car.posx,posy=Car.posy;

……

move\_to(posx,posy);

（5）扩展指令集中的Q查询

T=1表示回答后续的UUS如此结构表示类别和值

T=0表示询问后续的UU如此结构依次表示类别

比如QT0tU4uU1uU4uU1uq表示存储在变量1和2中的值

而

Q

T1t

U4uU1u

S

ET1tU0uU0uS1se

s

U4uU1u

S

ET1tU0uU1uS2se

s

q

表示回答存储在变量1和2中的值

### 2.3扩展的通讯指令集的意义与应用

基础的通讯指令集可以实现对小车的基本运动控制，并可以获取其参数。可以说这可以使用一切硬件提供的功能，通过这个通讯指令集可以描述许多小车的行为，指派小车进行许多操作。

但是如果脱离中央服务器的命令，小车如果还想要自主的有一些运行策略，则需要事先将这些策略烧写到硬件芯片中。试想一些应用场景，我们并不能总是召回小车进行有线的烧写，另外我们也不能保证小车一直能够接受服务器的指挥。

所以我们在基础的通讯指令集上加入了循环、选择、判断和记录4种指令，一下子扩充了指令集的表达能力，我们可以使用扩展的通讯指令集编写一些程序，并无线的烧写到小车上，小车便可以执行我们编写的策略，我们随时可以改变策略并烧写新的程序，同时我们对于在网络范围外的小车不必要过于担心，因为将会继续按照先前的策略执行，并在重新回归网络时按照策略中的要求汇报那些离线期间获得数据。

由于当前的实现底层是基于Arduino的，因此我们将它称作Arduino-Based Car Language，简称ABC语言。扩展的通讯指令集是ABC语言的底层表达形式，我们希望进而设计更上层的语言表达形式（并提供编译途径）。

扩展的通讯指令集实际上可以作为一种程序来看待了。当我们上传这样一段指令到小车上时，小车便会按照程序中的逻辑运行。使用扩展的通讯指令集的好处一方面是封装了底层的硬件使用，另一方面我们可以使用无线的方式“烧写”小车的程序，而不需要每一次都召回小车通过USB线进行烧写，这样在实际应用中我们需要使用的小车可能离我们非常遥远，或者由于紧急情况等不及小车回归再改写程序了，使用扩展的通讯指令集就可以直接远距改变小车的行动策略。

下面举几个应用的例子:

小车在4个点顺次巡逻，如果途中有障碍物则从右方绕开它（假定这个障碍物可以从右方绕开）

首先是一个巡逻的例子。假设小车的任务是小车在2个点（设坐标分别为（0,0）和(100,100）)之间往返巡逻监测是否发生异常情况（异常可能表示火情，则为温度或烟雾或火焰传感器，或者可能表示有人（在一个要求无人的地方），或者可能搭载摄像头经过简单数字图像处理后为标定为异常的情况），总之假定小车搭载的传感器在监测到异常时会返回1，否则返回0，要求其在巡逻途中发现异常后记录异常位置（假定小车处于组网状态）。

使用扩展的通讯指令集编写的结果是：

F

Ss

F

Ss

DU2uX0xY0yd

I

J

U0uU0uT0tS0s

U0uU1uT0tS0s

j

Uu

i

I

J

U2uU0uT0tS1s

j

ET0tU2uU0uS0se

i

f

F

Ss

DU2uX100xY100yd

I

J

U0uU0uT0tS100s

U0uU1uT0tS100s

j

Uu

i

f

f

然后是一个搜索的例子。假定小车在一片给定区域（（0,0）（0,100），（100,0）（100,100））中搜索某一个物品，以下程序可以遍历给定区域。

//前往（0,0）点

F

Ss

DU2uX0xY0yd

I

J

U0uU0uT0tS0s

U0uU1uT0tS0s

j

Uu

i

I

J

U2uU0uT0tS1s

j

ET0tU2uU0uS0se

i

f

//使方向朝向(0,100)

DU2uX0xY100yd

F

Ss

//前进100，即对某个X，遍历Y

F

Ss

DU0uS100sd

I

J

U0uU1uT0tS100s

j

Uu

i

f

//相当于X++

DU1uZ-90zd

DU0uS1sd

DU1uZ-90zd

//判断是否遍历完成（X=101）

I

J

U0uU0uT0tS101s

j

Uu

i

//再次遍历Y（蛇形）

F

Ss

DU0uS100sd

I

J

U0uU1uT0tS0s

j

Uu

i

f

//X++

DU1uZ90zd

DU0uS1sd

DU0uZ90zd

//判断是否遍历完成（X=101）

I

J

U0uU0uT0tS101s

j

Uu

i

f

### 2.4高级语言设计：ABC语言雏形（下一步重点研究）

Arduino-Based Car Language的设计目标是为扩展的通讯指令集提供更易理解与编写的高级语言形式。

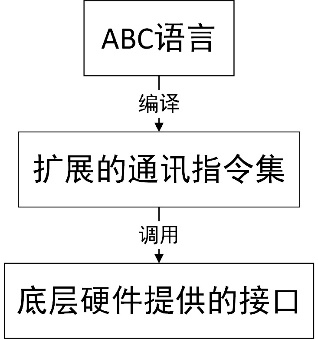


图 17

## *（三）网络通讯层：通讯与组网*

### 3.1硬件层

#### 3.1.1蓝牙跳连组网

首先，蓝牙主机和蓝牙从机可以自动连接，对于HC-05型号的蓝牙，可以使用AT设置其主从属性，同时蓝牙从机可以与手机中蓝牙串口助手进行蓝牙建立连接（实验中我们主要使用华茂科技公司开发的蓝牙串口助手）。

由于我们目前使用的蓝牙模块支持一对一连接通讯（最新的蓝牙是支持的一对多的），所以如果我们想要3辆及以上的小车组网，则只能采用跳连组网的方式，即2辆小车建立连接后一段时间断开连接再与另一辆小车建立连接。

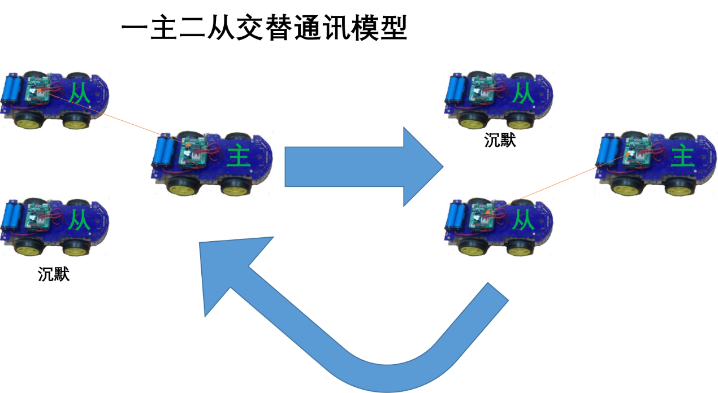


图 18

断开连接的实现是继电器+蓝牙。

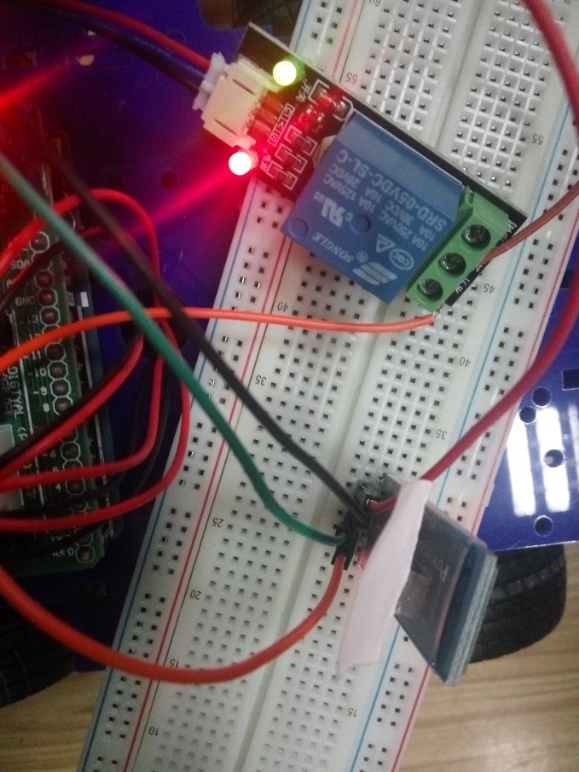


图 19

但是有一种可能就是断连后再连接的还是上一辆小车，这时会采取一种惩罚策略，即错误的连接发生后，对方会被施加一个递增时间的沉默（在这段时间内不可开启蓝牙）。

通过以上方法，抽象而言消息可以在任意2辆小车之间传递从而完成组网。

在一主二从的情况下，主机小车会轮流与2辆从机小车建立连接。

这样跳连组网的缺点是消息传递效率较低。因此之后我们又尝试了使用WIFI模块。

#### 3.1.2ESP8266WIfI模块

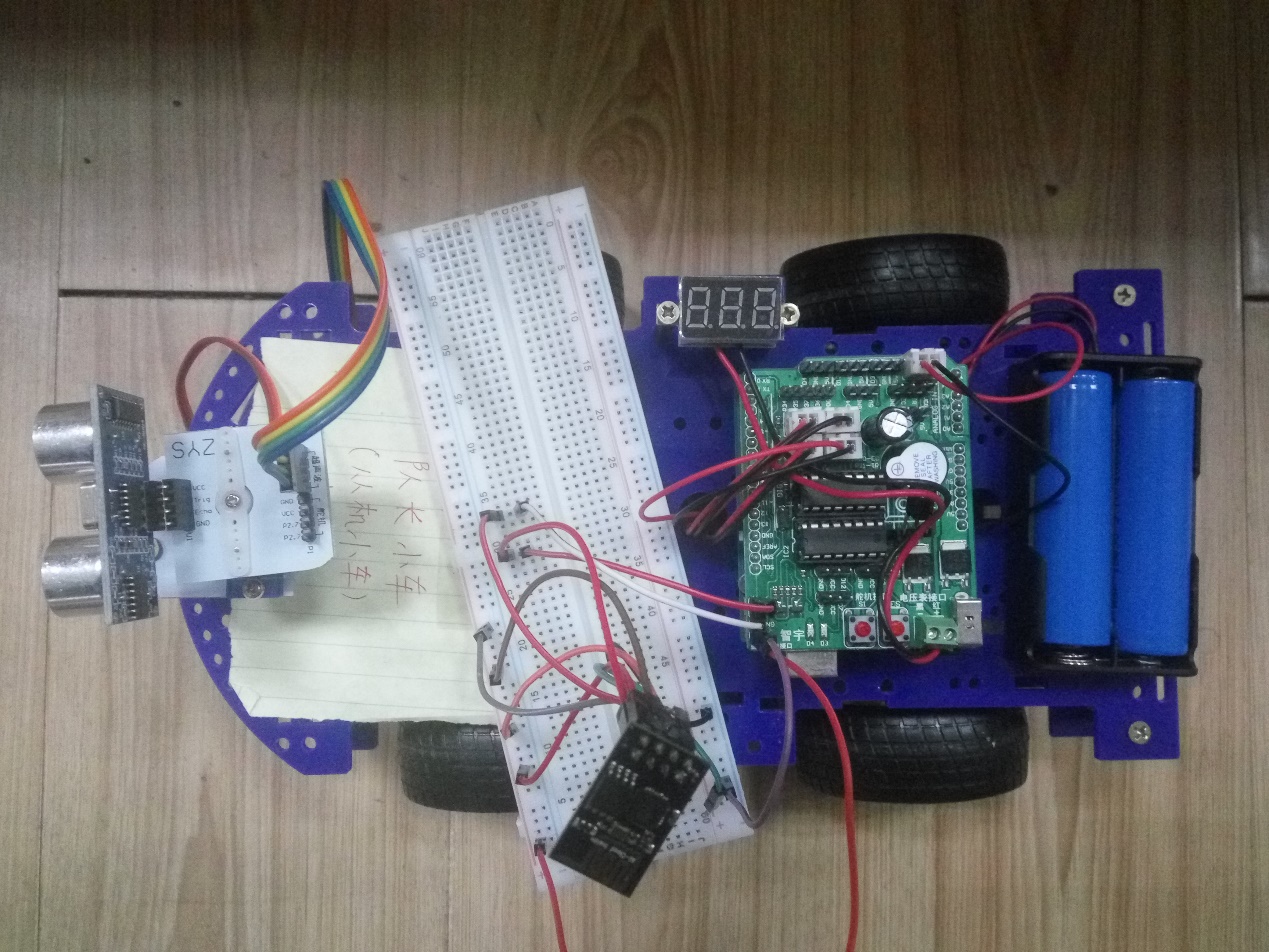


图 20

首先用Arduino语言编写的程序烧写进ESP8266 WIFI模块作为TCP客户端，接着我们用Java语言写了TCP服务器，两者进行套接字通讯，接着我们使得ESP8266 WIFI模块能够与Arduino主板进行通讯，从而使得WIFI模块成为服务器与Arduino主板之间通讯的桥梁，这样便可以把Arduino与WIFI模块一起看成是TCP客户端。

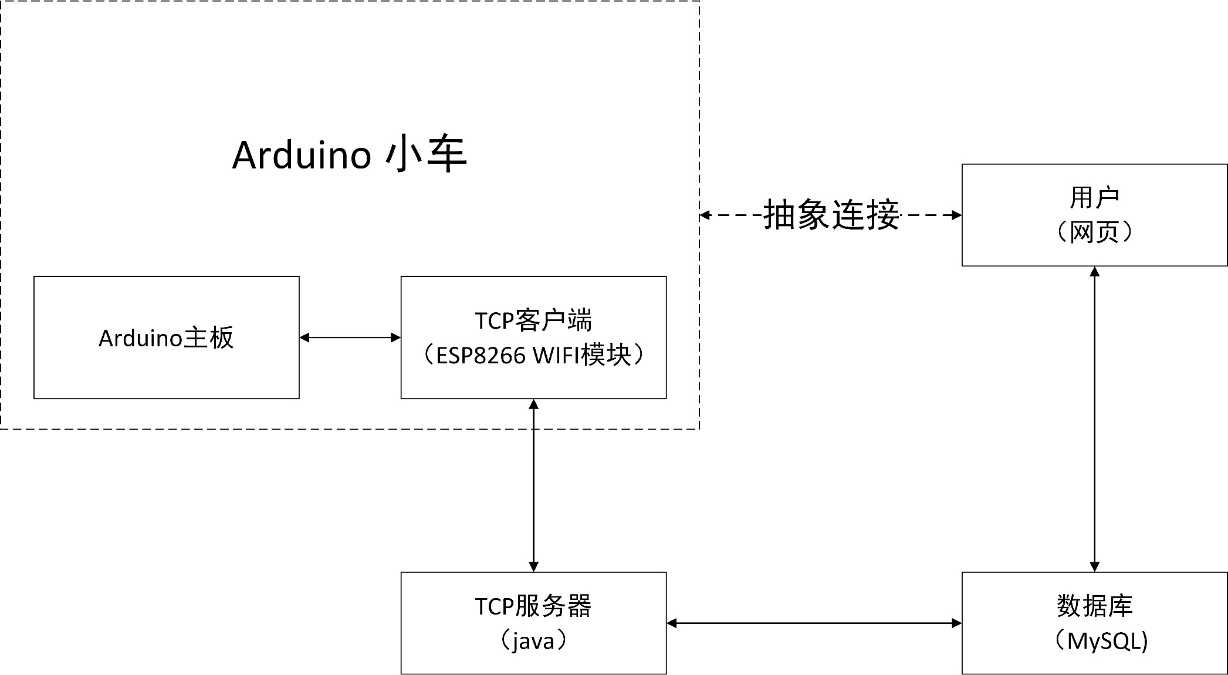


图 21

为了显示小车的接收信息情况，我们使用命令行telnet工具模拟小车的接收。

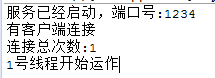


图 22

打开服务器，端口打开为1234。

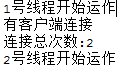


前面的telnet按下回车，可以看到：



同时我们有多辆小车需要与服务器进行连接，所以我们改写java程序使用多线程达到服务器可以一对多客户端。

此时可以再打开一个telnet，可以看到又出现了一个子线程的服务器端。



另外服务器可以作为小车与小车之间通讯的中介。

这里演示1号小车现在发送消息给2号小车。



图 23

可以看到模拟小车2号收到了消息，由于其一直没有回复确认，所以该消息被一直重发。

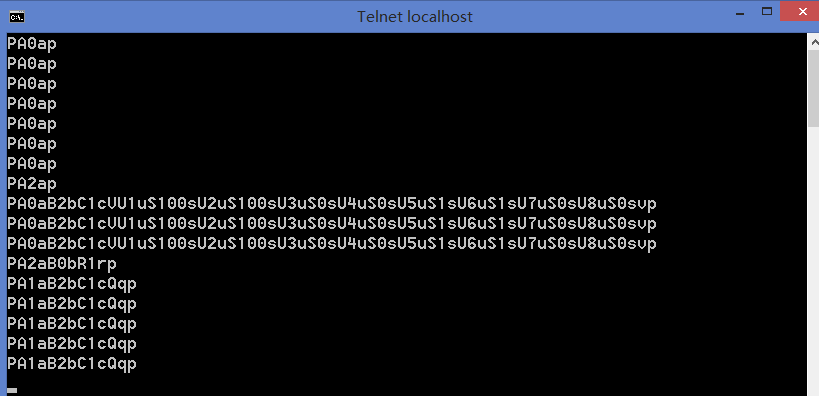


图 24

现在模拟小车2号给出确认帧。



图 25

可以看到模拟小车1号收到了确认帧



图 26

### 3.2网络层

虽然套接字或蓝牙通讯一般而言是可靠的，但是偶尔会发生丢失或较大延迟或错误的情况，所以我们对每辆小车对另一辆小车的信息进行编号，要求接受的小车对收到的消息返回确认，在实际的实现中这种确认信息必须是单独的即暂时不支持捎带确认。如果小车接受的消息编号已经是处理过的，小车将对此不作处理，但是仍然会对此返回确认帧。为实现的简单起见我们采用了窗口大小为1的滑动窗口模型。一辆小车只要没有收到确认帧就会不断重发该消息。

如果小车断开了连接，那么其他小车会取消等待其确认帧。

我们这里采用心脏检测的技术来确定其他小车是否保持连接，即不断发送最小单元的无关信息，如果成功发送（即收到确认）则表明连接维持。

## *（四）硬件层：小车向上提供的功能*

### 4.1硬件材料使用型号说明

小车组件：小车底盘、轮胎、Arduino主板、（电机&电池）功能转接板（湖南智宇科技公司生产，我们为方便使用对其进行了简单的焊接改造）

通讯器件：HC-05（蓝牙主从一体机）、HC-06（蓝牙从机）、ESP8266 WIFI模块传感器：传感器：HC-SR04超声波传感器、MH光电编码器、GPS模块

其他：舵机

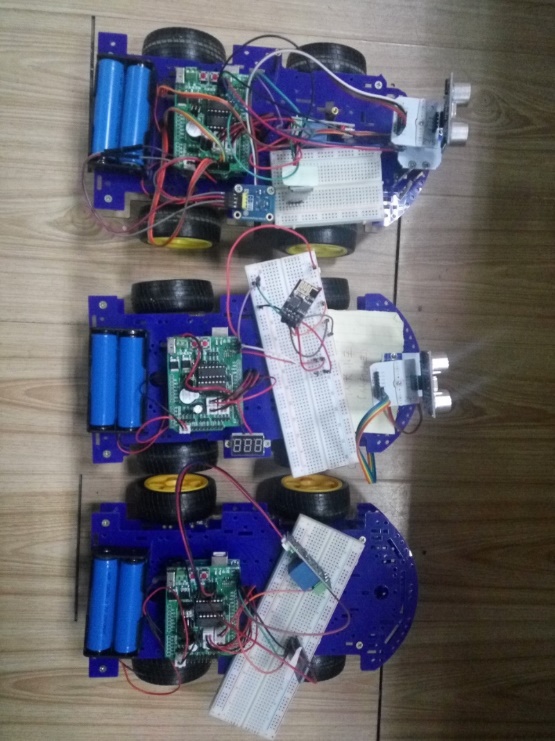


图 27

### 4.2基本运动控制、误差控制

#### 4.2.1误差控制的要求

由于实际应用中的定位是由GPS模块完成的，考虑到GPS模块的精度一般都有3米，因此实验中误差控制只关注3米以内，对长距离的偏差不予考虑。

#### 4.2.2直行

通过使双侧电机向前转动与反向转动可以实现小车前进与后退。虽然被2个不同电机控制，同一侧的2个轮胎在实际运动中速度将是一样的，但是不同侧的轮胎转速可能会有差异，从而导致小车在实际运动中偏向某一侧。这可以通过对电机进行PWM控制来调整转速差异，问题在于小车需要准确地知道当前它是偏向哪一侧。

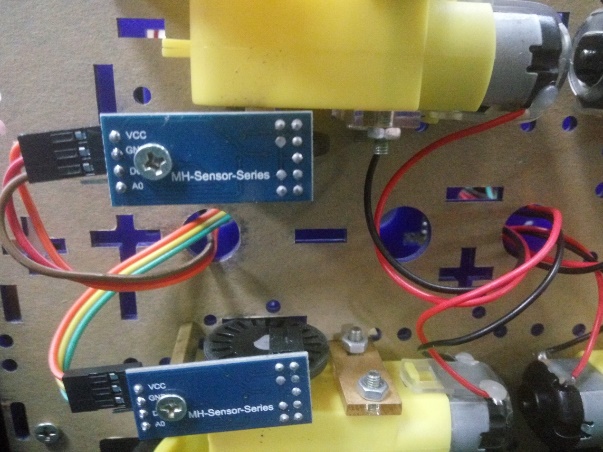


图 28

我们使用光电编码器，当安装在电机上的测速码盘遮挡光电编码器时会使之原本持续发射的脉冲信号消失（表现为低脉冲信号），通过计算某一时间内的低脉冲个数可以对左右进行比较从而即时调整左右电机转速。

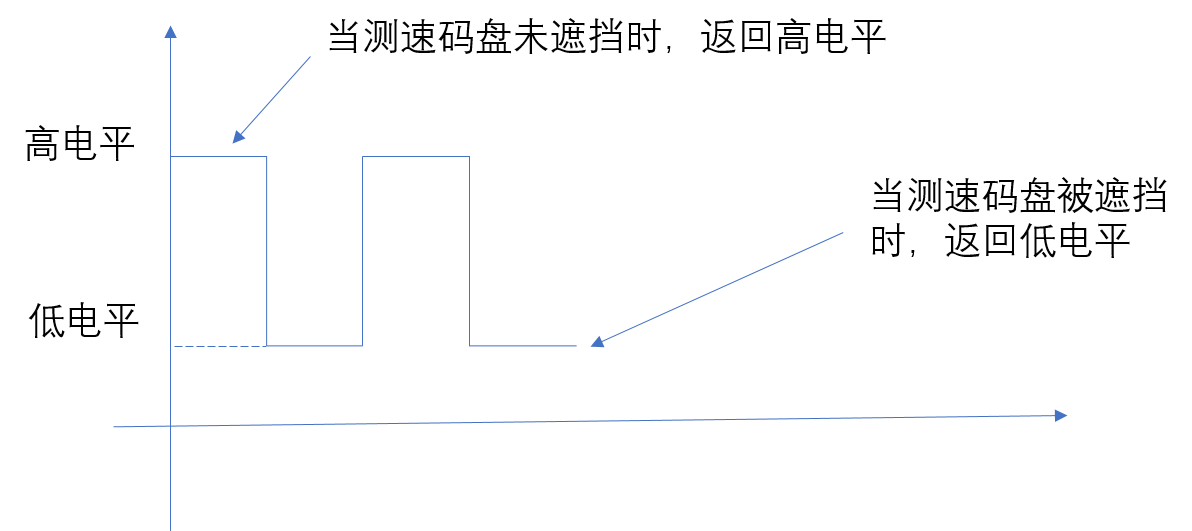


图 29

坏消息是在实际实验中会发现，比如对于事实左偏的一次运动，它有可能会报告左边较快也有可能报告右边较快。好消息是当使用这种策略时，要求其直行一般总会偏向同一侧，这种恒常性说明如果我们对左右光电编码器的结果乘以某个合适的系数，则会更靠近直行。这个系数是要人工调试的，不过在调试完后，小车可以在很长一段时间内进行自动的直行校准。最终能够达到前进1米左右偏移在5厘米以内。

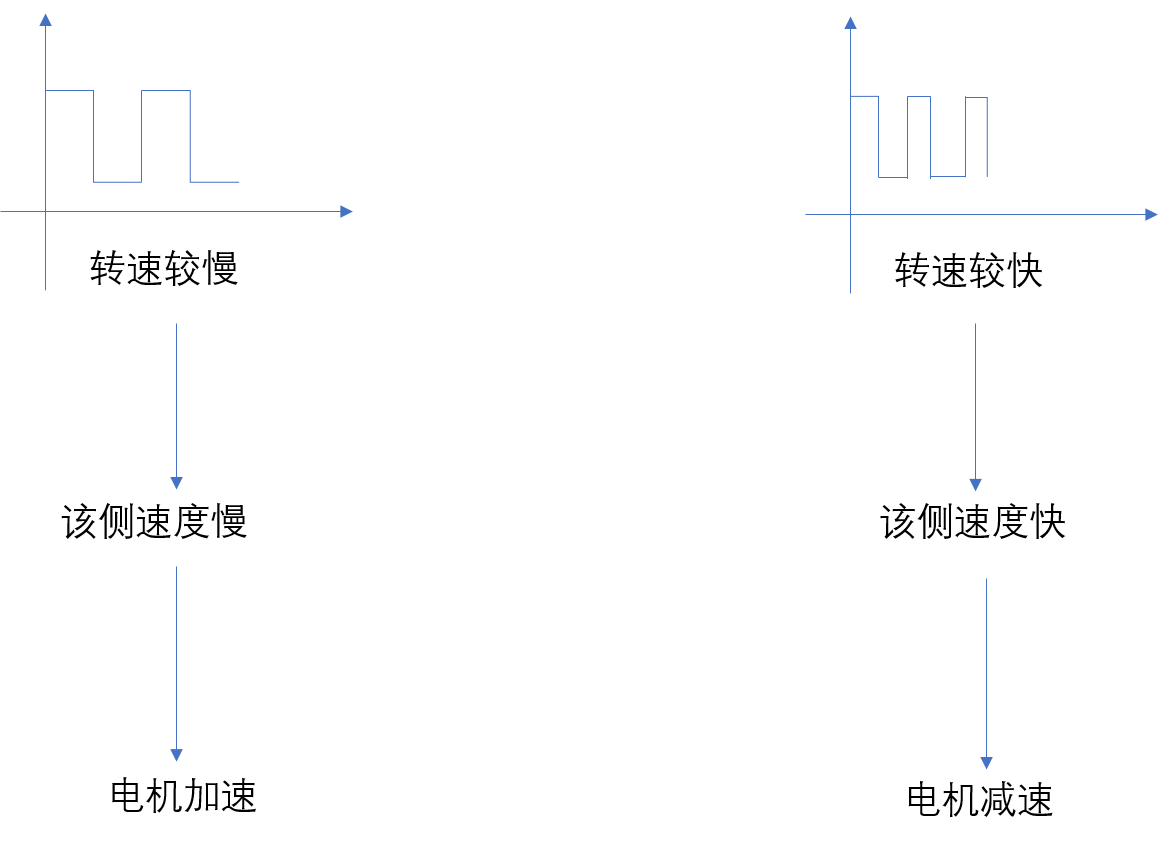


图 30

另外需要调试的一个参数是单位时间直行的距离，在实验中我们的单位是厘米，用户使用时指定的距离都是现实中的距离，所以小车会根据这个参数来转换到电机运转的时间，当然对于不同转速的电机这个参数的值是不同的，我们的处理是：



另外，考虑小车在启动与刹车时是加速与减速的过程，直接使用匀速时的单位时间前进距离对于长距离的运动误差不会太大，但是对于短距离的运动则会有较大偏差。

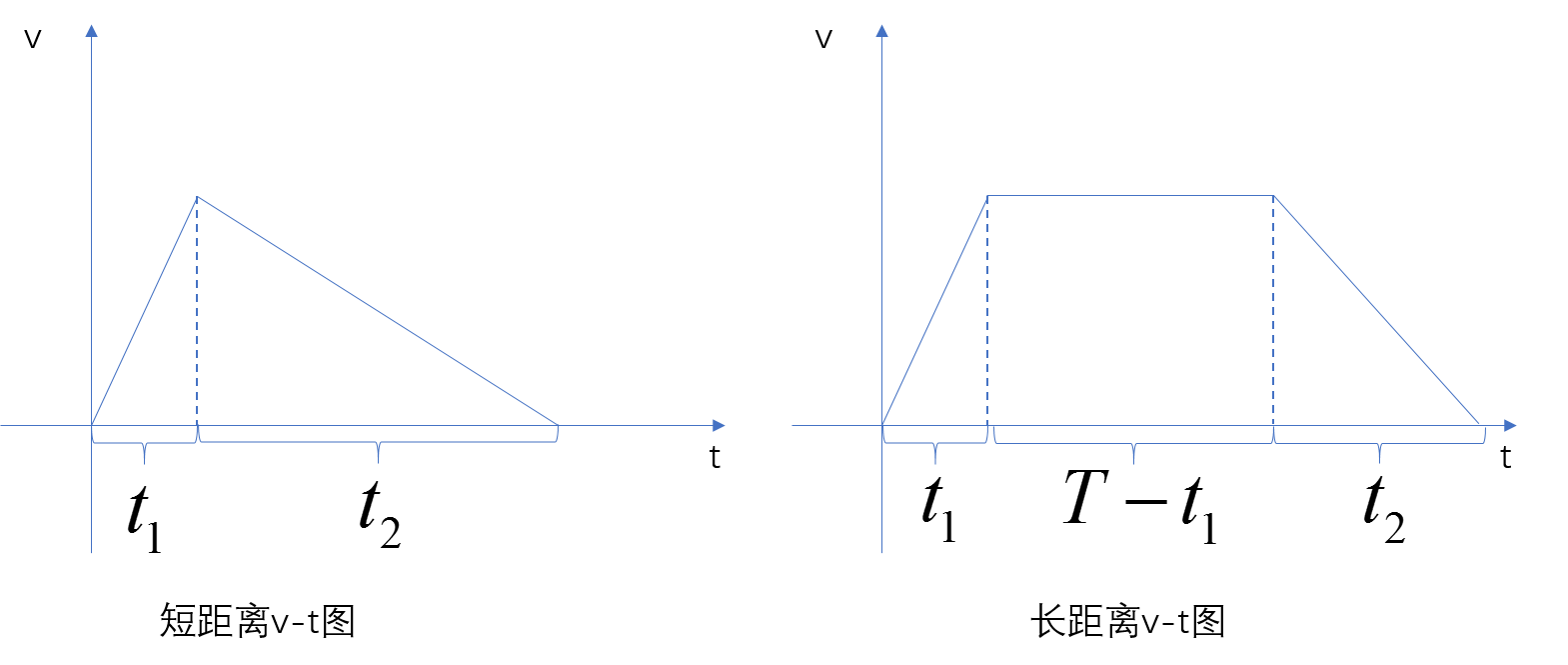


图 31

假设我们知道了小车启动的时间为，则刹车时间也是，则对于给定的前进的的要求，匀速运动下单位时间前进距离，计算应该让电机转动的时间的过程如下（这里注意不涵盖后面的刹车时间）：





一般刹车时间会比启动时间要长，接下来的问题就是如何测量小车启动时间和刹车时间。

测量小车启动时间的方法是，使得电机转动一个较小时间后小车不会运动，再找一个大一些的时间使得小车会运动，然后二分测试小车恰运动的那个值。测量小车刹车时间的方法是让小车返回关闭电机时的已开机时间，然后在小车运动的路线上放一个超声波传感器，输出测障距离与其开机时间的数对，这样时候就可以得到小车的刹车时间与刹车距离。

不过总体而言，这个误差控制主要针对短距离的精度有所提高。

#### 4.2.3旋转

使小车旋转有2种方式，一种是单侧电机旋转，另一种是两侧电机反方向旋转。出于误差控制的考虑，我们采用双侧电机反转的方式进行旋转。这么做有2个好处：

首先，双侧旋转时旋转中心为小车中心，而单侧旋转时则不是，所以采用双侧旋转可以使得小车原地旋转后对自身坐标的偏移尽可能小。

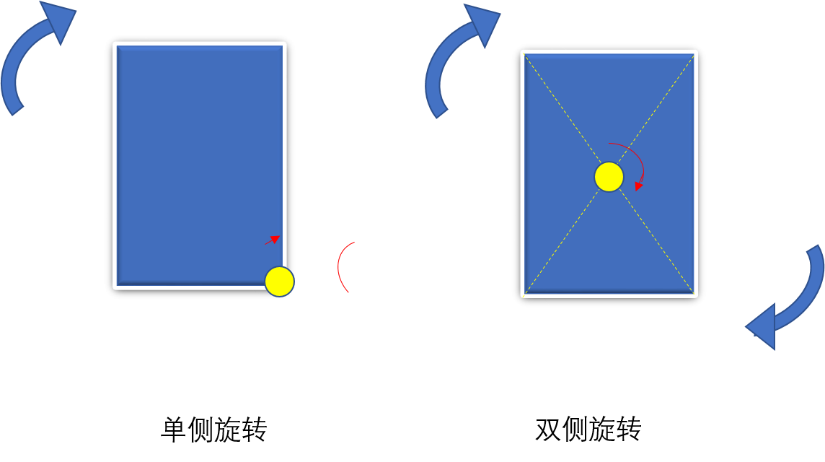


图 32

其次，像直行控制中考虑到启动和刹车一样，双侧旋转使得启动时的加速度更大从而由于启动时间而产生的误差会稍小一些。

不像直行中启动和刹车对整体运动不是很重要，旋转时这两者的影响较大，这是因为每次旋转几乎都是电机的短时间运转，于是电机启动与刹车时所占时间的比重要比大多数直行任务中要大。这里要测量单位时间旋转度数。电机启动时间可以用上面类似的方法测量，刹车时间不是很好测量，可以用电机启动时间乘以系数后用看旋转角度是否符合预期。这里不考虑启动与刹车时间的话人工校准的结果一般只能适用某个特定角度周围，对于其他角度则会发生较大偏差。公式与上面类似：



在实验中，也可以只是针对一些特定角度校准，保证实验效果即可。

另外，在旋转误差控制中使用电子罗盘的努力也基本以失败告终，这是因为我们使用的电子罗盘模块的读数基本只能精确到某半个象限，对于现实中的4个象限，电子罗盘测得的读数并不是均与分布的，往往有2个象限刚好90°而一个120°一个60°。

### 4.3定位

在做过上述误差控制的努力，我们平时计算位置时便不过多考虑偏差，每个小车维护一个位置（初始位置给定），小车根据初始位置和其执行的直行与旋转操作推算所在位置。

但是在项目后续，这个简单的操作可以得到一定的升级，也就是借助周围障碍物辅助自身定位。

由于一般的超声波传感器在合适的检测距离下精度可以精确到毫米，那么小车根据身旁的障碍物可以更准确的对自己进行定位（即当有足够有理由相信这是一个静止的障碍物时，可以以其为固定点，将与其的偏移距离作为自身的位移距离，而不是用自己的参数进行推算，同时如果这个测量距离足够可信的话，那么还可以更新自身的参数。）

虽然在实验中受制于实验环境较少使用GPS传感器，不过还是对其进行了测试。GPS传感器会返回所在地的经纬度。

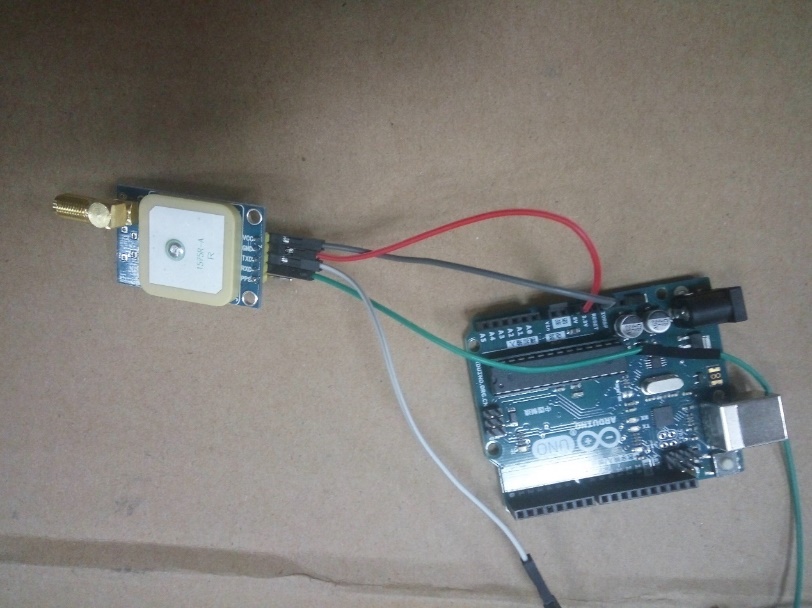


图 33

### 4.4障碍物检测

实验中我们使用的超声波传感器已经能达到很高的精度（毫米），除了对超远和超近距离的障碍物有较大偏差，对合适距离障碍物其读数非常可信。

为了使得小车能够探测除了前方的障碍物，又装载了舵机云台，不过由于其转动精度较低，所以平时只使用正前方、正左方、正右方3个方向，这个转动的参数均需要事先人工校准。（后续可能开放左上45°与右上45°）

# 四、创新点

项目中比较创新的首先是本项目的立足点不是研究单辆小车而是研究多辆小车之间的协同。这在不同的应用背景下会有不同的好处，如在巡逻任务中，多辆小车可能分开巡逻增加异常事件的响应与巡逻的范围，也可能搭载不同设备共同巡逻扩展单辆小车的能力；在地图绘制中，多辆小车共同协作不仅能增加地图绘制的效率还能提升信度；在区域搜索中，多辆小车协同工作能缩短搜索时间。

然后，项目搭建了可视化的小车调试平台，该平台可以直接操控小车、与小车进行通讯、修改小车参数并可视化的显示小车与障碍物分布的模拟图。这使得各种实验可以方便的进行、更便捷的调试小车同时更好的展示实验效果。

最后是从基础指令集到ABC语言的升级，基础的通讯指令集最初的产生是为了对于小车的基本必要控制，但是当我们对其加入选择、判断、记录、循环等扩展规则时，我们便可以使用它编写复杂的策略设计并无线的上传至小车使之执行。

# 五、实施过程中的得失成败

误差控制的目标是小车能够走直线、旋转预期的角度，这样才能使得小车到达预期坐标。

在项目刚开始的时候，我们采用人工校准的方法，这比较烦琐，于是我们花费了非常多的时间研究误差控制的问题，尤其是尝试进行自动化的误差修正，但事实证明这种尝试是徒劳的与不必要的。

一方面对误差自动控制有高精度要求是难以达到的，我们思考了很多可能解决问题的算法，这些算法都停留在软件层面，最终我们认为在我们使用的硬件精度下，对误差自动控制的高精度要求是几乎不可能达到的。

一次失败的尝试是使用电子罗盘模块，在直行的过程中检测角度是否发生变化从而调快或调慢某一侧的电机转速。这在有时能表现较好的自动校准结果，但在更多时候小车会以不可预知的方式乱跑，最后我们查明即时在同一条直线上，电子罗盘模块获取的读数也可能不同，在有些地方这种差异可能较少从而小车几乎在一条直线上，但在有些地方这种读数会差异到90°以上。另外它会受到许多干扰，比如将手机靠近电子罗盘模块时，它的读数会立刻发生10°以上的变化。使用更高精度的电子罗盘模块可能会使小车达成直行，但是对于我们的项目我们已经放弃了使用电子罗盘模块作为校准小车直行的传感器。

另一方面对误差自动控制有高精度要求是不必要的，使用GPS模块可以使定位误差保持在3m以内，在实际应用中对于我们的项目的应用（巡逻、搜索）来说是完全足够的。

但是我们项目中的误差自动控制并非完全没有意义。

从实验的角度来说，我们并不能总是在非常开阔的场地进行实验，于是在狭小的场地中，我们仍然需要高于实际应用时的精度来使实验效果能够得以展示，同时这可以节省实验时间。

从实际应用的角度来说，对于GPS模块暂时失效的情景下，使得小车的误差尽可能小仍然有一定意义。

总结来说，误差自动控制的结果好于不进行误差自动控制，但是不如人工校准。

我们考虑到了人机交互的重要性所以前期我们使用这个液晶显示屏把超声波传感器的数据及一些其他数据展示在显示屏中，方便调试与展示，不过后来我们逐渐开始使用网页进行人机交互，这是因为一方面液晶显示屏的表现实现比较复杂，且实现出来的效果也不是很好，最致命的是其与Arduino主板组合使用如果不适用其他读写设备（比如SD卡读写器，我们还真买了并进行了研究），那么存储空间非常小。