蓝牙智能小车

王元桢 2017202026

在选择A，B作业时觉得A作业，也就是小车会更有意思，所以就选择了小车这个作业，但是在实现的时候，由于对开发板，线路这些基础的电子知识没有什么认识，也遇到了一些问题，其中电压的问题就让我遇到了两次莫名奇妙的问题。下面是我如何组装蓝牙小车的过程。

1，首先先从网上买来所需的器件，小车套装，开发板，蓝牙，杜邦线，电机驱动板，控制开发板等原件。接着开始拼装小车，按照教程将车轮和马达固定在板上，这块问题不大，根据教程很快就拼装好了，然后开始考虑线路的连接，由于不了解电子线路，不知道该如何连接线路，于是去网上找教程，很快也拼好了线路，完成小车的拼装。这一块主要是线路不熟悉，后来明白了端口之间的关系后就可以较好地拼接。

2，然后开始做了蓝牙的部分，首先确定蓝牙控制的代码，利用Ardunio将对应的代码上传到开发板里，前进，后退和停止通过控制电机两边的电压使车轮转动，左转时左轮停止，右轮前进，右转时右轮停止，左轮前进。然后再下载一个蓝牙串口助手到手机里，手机将利用这个APP发送蓝牙信号给蓝牙模块，在手机助手里进行相应的设置，自定义按键和对应的信号。

3，然后将蓝牙的线路也连接到开发板上，一开始用充电宝尝试，但充电宝只有5V电压，不够，索然对应的灯都可以亮，手机发送信号也的确发出去了，但是马达并没有转动，不知道出了什么问题，后来舍友换了块9V的电池后，就可以转动了，我也换了块9V的电池，马达可以转动了，可能是驱动板的电压要求问题，充电宝达不到要求。解决了该问题后就可实现蓝牙小车了。

4，在小车上添加超声波模块，利用线路将其连接到开发板上，需要注意超声波的发射端和接收端所对应的端口，然后在写代码时需要注意发射端和接收端端口间的对应，否则将无法使用发射端和接收端。加蓝牙模块，在第一次作业时已经添加了蓝牙模块并设置了相应的按键，所以这次只需要加上超声波模块。

5，加入超声波模块，修改程序然后重新导入开发板，在程序里加上超声波的处理，利用超声波来探测前方阻碍物的距离，如果探测到距离小于一定值则不再前进，可以修改这个值来改变小车的灵敏性，由于没有舵机，只能探测小车前方的物体，所以没有办法知道左转和右转哪个更好，所以我这里只有右转，也就是遇到障碍物时指挥右转，如果右转后仍然有障碍物，则仍然右转，直到前方没有障碍物为止，可能会出现原地打滚的情况。最后可实现利用超声波只能避障。

在实现的过程中碰到了一些问题，因为对这方面缺少知识，所以出问题时经常不知道出了什么问题。

1，电压问题，无法用充电宝供电，由于老师说可以用充电宝供电，但是我插上充电宝后毫无反应，也不知道到底是什么问题，后来舍友用9V电池后小车可以跑动了，所以我也改用9V电池实现供电，至于老师的充电宝也能成功，可能是开发板或连接线路时的区别，这个我也不知道真正的原因。

2，线路问题，对电路板不是很了解，在连接电路时不知道如何对应端口，测试时出现端口冲突等现象，于是上网查询对应资料并进行多次测试，后来明白开发板给我们的接口也无非就是接口的对应关系罢了，只要对应好端口就可以了，这个问题倒是挺好解决。

3，超声波问题，添加超声波后不会使用，查询资料后其实也就是控制发出端口和接受端口，然后计算距离就可以了，调用几个对应的函数就可以解决。利用超声波就可以实现小车的避障功能了。

4，我想实现既能自动驾驶，又可进行蓝牙控制，但想了想觉得需要多线程，上网查了查资料，有说UNO开发板不适合用多线程，可能会出现问题，后来借鉴了老师PPT里的代码，引入了新的控制变量和改变了代码模块的设置，解决了这个问题。

5，一开始实现了自动驾驶避障后，应该是由于电池的原因，电压提供不上，我的小车出现了只能一只轮动，无法两只轮同时动的情况，查看代码也找不到什么问题，但怎么测试都不对，也不知道到底是什么问题，将线路重新组装一遍后仍然解决不了，后来测试多次后，电池慢慢用完了，突然想到会不会之前是电池没电了所以出现了一系列的问题，买了一块新的电池后，解决了问题，感觉这个问题真是出现的莫名奇妙，不知道哪里出了问题和如何修改。

最后实现了蓝牙智能小车，可以自动驾驶实现避障功能，也可以人工进行蓝牙控制，通过实现蓝牙小车，我也了解了一些电子基础知识，学到了一些东西，并且这个实验也很有趣，感觉颇有收获。

**实现小车图片：**



**相关代码：**

#define STOP 0

#define FORWARD 1

#define BACKWARD 2

#define TURNLEFT 3

#define TURNRIGHT 4

#define AUTO 5

int leftMotor1 = 6;

int leftMotor2 = 7;

int rightMotor1 = 8;

int rightMotor2 = 9;

int inputPin=11; // 定义超声波信号接收接口

int outputPin=12; // 定义超声波信号发出接口

int isauto = 0;

void setup() {

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

//串口初始化

Serial.begin(9600);

//测速引脚初始化

pinMode(leftMotor1, OUTPUT);

pinMode(leftMotor2, OUTPUT);

pinMode(rightMotor1, OUTPUT);

pinMode(rightMotor2, OUTPUT);

//超声波控制引脚初始化

pinMode(inputPin, INPUT);

pinMode(outputPin, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

//if(Serial.available()>0)

//{

//char cmd = Serial.read();//读取蓝牙模块发送到串口的数据

//Serial.print(cmd);

// motorRun(cmd);

//}

avoidance();

}

void motorRun(int cmd)

{

switch(cmd){

case AUTO:

Serial.println("AUTO"); //输出状态

isauto = 1;

avoidance();

break;

case FORWARD:

Serial.println("FORWARD"); //输出状态

isauto = 0;

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

break;

case BACKWARD:

Serial.println("BACKWARD"); //输出状态

isauto = 0;

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, HIGH);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, HIGH);

break;

case TURNLEFT:

Serial.println("TURN LEFT"); //输出状态

isauto = 0;

digitalWrite(leftMotor1, HIGH);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

break;

case TURNRIGHT:

Serial.println("TURN RIGHT"); //输出状态

isauto = 0;

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, HIGH);

digitalWrite(rightMotor2, LOW);

break;

case STOP:

Serial.println("STOP"); //输出状态

isauto = 0;

digitalWrite(leftMotor1, LOW);

digitalWrite(leftMotor2, LOW);

digitalWrite(rightMotor1, LOW);

digitalWrite(rightMotor2, LOW);

}

}

void avoidance()

{

int dis;//距离

motorRun(FORWARD);

dis=getDistance();

if(dis<30)

{

motorRun(TURNRIGHT);

delay(500);

}

}

int getDistance()

{

digitalWrite(outputPin, LOW); // 使发出发出超声波信号接口低电平2μs

delayMicroseconds(2);

digitalWrite(outputPin, HIGH); // 使发出发出超声波信号接口高电平10μs，这里是至少10μs

delayMicroseconds(10);

digitalWrite(outputPin, LOW); // 保持发出超声波信号接口低电平

int distance = pulseIn(inputPin, HIGH); // 读出脉冲时间

distance= distance/58; // 将脉冲时间转化为距离（单位：厘米）

Serial.println(distance); //输出距离值

if (distance >=30)

{

//如果距离小于30厘米返回数据

return 30;

}//如果距离小于30厘米小灯熄灭

else

return distance;

}