

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

工程实践与科技创新

SCIENCE AND TECHNOLOGY INNOVATION IIB

PC 端自动控制小车走黑线

THE AUTOMATIC TRACKING OF CAR CONTROLLED BY PC

项目报告

PROJECT REPORT



小组成员: 包伟铭 刘兴阳 罗晶 庞博

小组编号: II B-25

指导教师: 张 士 文

学 院: 电子信息与电气工程学院

目 录

1 项目简介	3
1.1 成员介绍	3
1.2 选题：小车走黑线。	3
1.3 实现功能：	3
2 实施方案.....	4
2.1 小车端实现	4
2.1.1 硬件部分	4
2.2.2 软件部分	7
2.2 主机端实现.....	9
2.2.1 概述	9
2.2.2 图像处理	10
2.2.3 跟踪小车	12
2.2.4 发送指令	14
3 特色及优化.....	17
3.1 项目特色	17
3.2 优化方案	18
3.2.1 转弯半径	18
3.2.2 小车防抖	18
3.2.3 舵机响应速度	18
3.3 后续改进	18
4 组员体悟.....	19
5 参考文献.....	21
6 致谢	21

1 项目简介

1.1 成员介绍

组号	二十五			
姓名	包伟铭（组长）	刘兴阳	罗晶	庞博
照片				
分工	OpenCV 代码调试，分工协调，小车组装，联合调试，课程项目报告（网页）	模块调试，Arduino 代码编写，小车组装，联合调试，课程项目报告	PC 端代码编写，小车组装，视频制作，跑道制作，联合调试，课程项目报告	PC 端代码调试，小车组装，视频制作，跑道制作，联合调试，课程项目报告

1.2 选题：小车走黑线。

PC 端软件调用摄像头获取路线信息，自动分析小车偏离情况，通过蓝牙交互控制小车循迹行进。

1.3 实现功能：

1. 自购组件组装小车，实现了 Arduino 平台下的“舵机控制”与“差速”两种转向方式。
2. 基于 OpenCV2.4 的主机端图像获取，二值化，冲蚀变换，角点识别，等图像处理及轨迹生成。
3. 小车与主机间的蓝牙串口通信。PC 端分析实时图像，调用 Windows Bluetooth API 发送控制指令，小车接收识别并进行对应操作。
4. 主机相对小车间的闭环控制：软件端比对小车在行进过程中与轨道的相对位置，生成闭环控制指令，实现小车姿态矫正。

2 实现方案

2.1 小车端实现

2.1.1 硬件部分

- Arduino Mega2560 单片机开发板



图 1：ARDUINO MEGA2560

我们组的小车采用 Arduino Mega2560 单片机平台作为小车的“大脑”，处理从主机端接收的信号，控制小车上的电机和舵机，使小车能够按照主机端的控制进行运动。

- HC-06 主从一体蓝牙模块

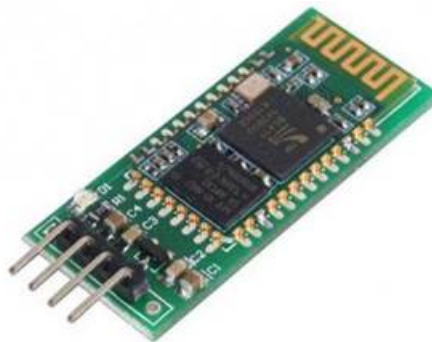


图 2：HC-06 蓝牙模块

HC-06 主从一体蓝牙模块采用低成本、低功耗 CSR 芯片，支持蓝牙 v2.0 协议标准，可以实现终端在空旷环境下 10M 内通讯，使用广泛。

在我们组的小车中，我们用 HC-06 与电脑配对，以此接收电脑传递的字符串，并通过单片机的 RX、TX 端口与单片机进行通信。

➤ Futaba S3003 舵机



图 3：FUTABA S3003 舵机

舵机，也称为伺服电机，主要由直流电机、减速齿轮组、传感器和控制电路组成，可根据接收到的 PWM 信号调整转过的角度。

我们小组的小车利用舵机来控制小车前进的方向，使用的最大角度范围约 $\pm 60^\circ$ 。

➤ L9110 两路电机驱动

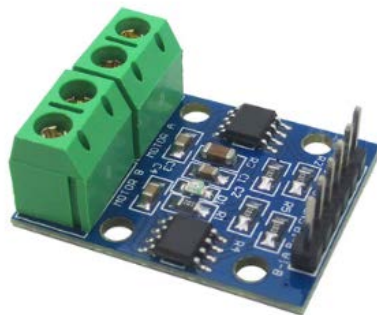


图 4：L9110 两路电机驱动

这是一个双 L9110S 的电机驱动，供电电压在 2.5~12V 之间，最大工作电流为 0.8A，可以同时驱动 2 个直流电机。

由于单片机端子不能提供足够大的电流，小车运动速度很慢，因此我们增加了该电机驱动模块，单片机提供控制信号，小车由电池直接供电。

➤ 整车组合

我们小车端电路系统连线如上图所示，通过在不同端口置高电位或者低电位，设置不同的 PWM 来控制电机、舵机，进而控制小车的运动。

一开始的测试阶段，我们采用移动电源为 Arduino 系统供电，但这种方法占位置且重量过大，因此我们自制了电源进行供电。解决了电力问题之后，我们又发现 Arduino 对电机供电不足，因此我们增加了电机驱动板解决了这个问题。

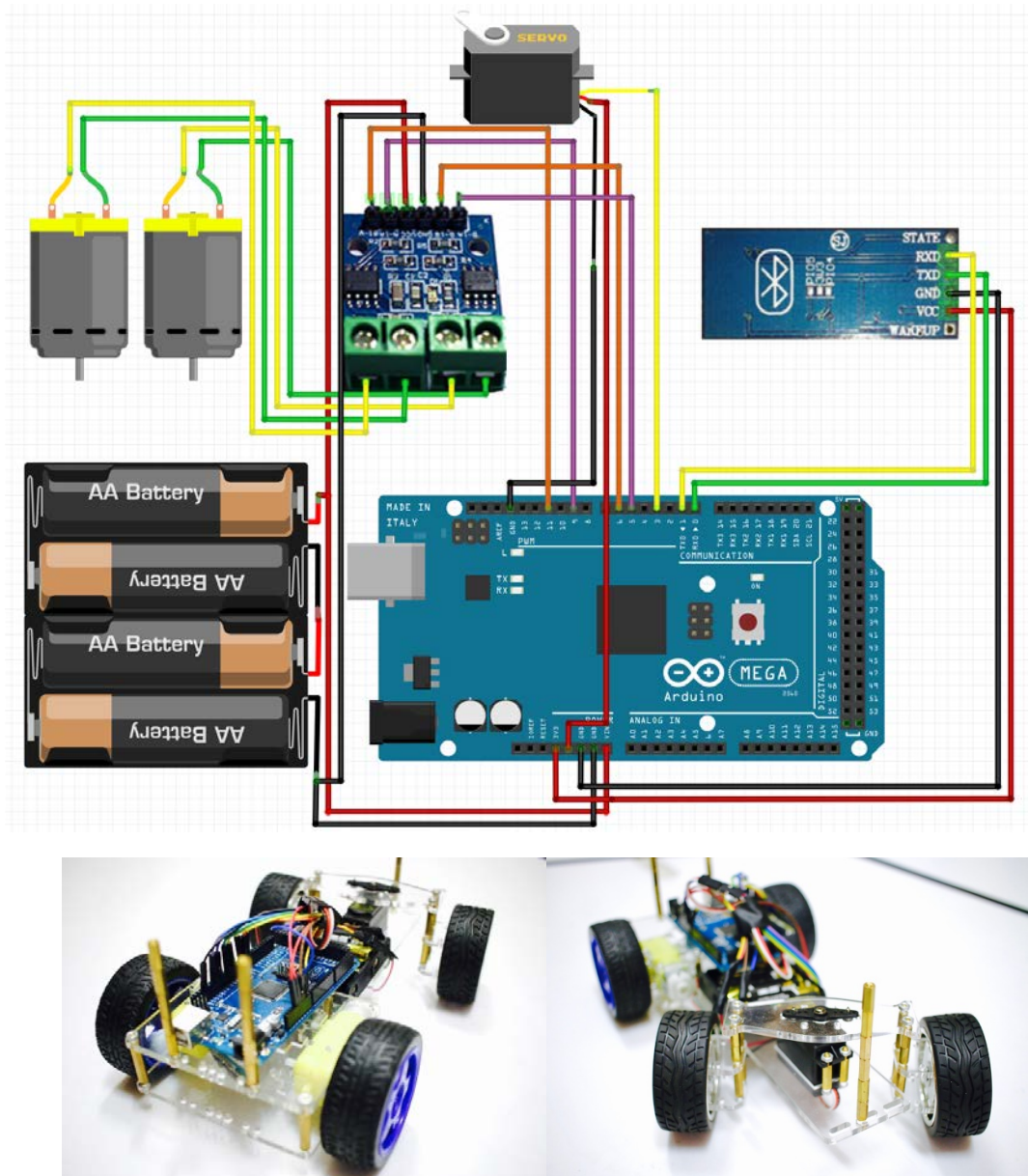


图 5：小车整体布局

2.2.2 软件部分

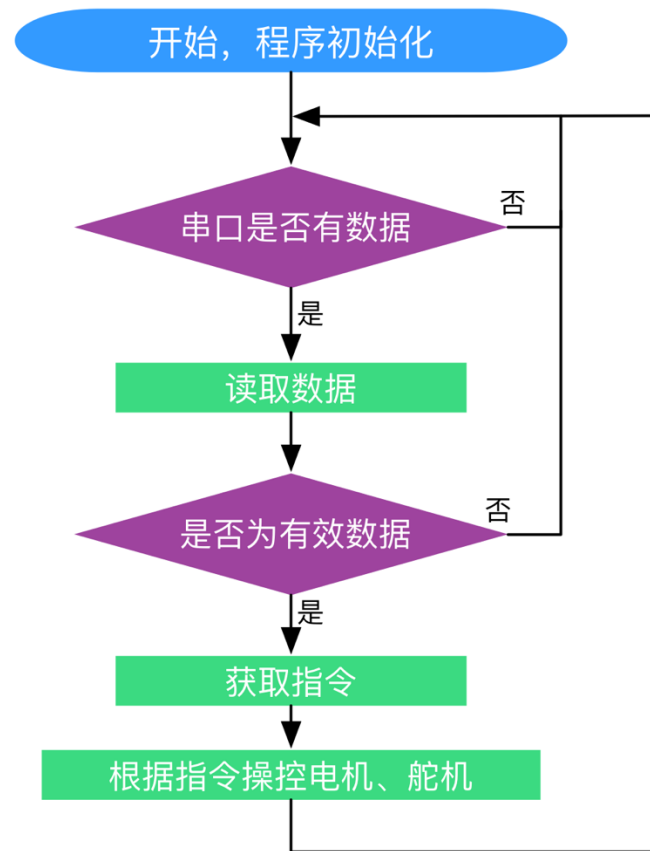


图 6：小车程序流程图

➤ 程序设计

我们用 Arduino 1.6.9 进行程序设计，主要思路是电脑通过蓝牙传送一个字符给单片机，单片机接收不同字符，进而控制电机与舵机状态。

主要有以下几种指令：

前进 (F)：电机正转，舵机正向

左转 (L)：电机正转，舵机左转

前进 (R)：电机正转，舵机右转

右后 (X)：电机反转，舵机右转

左后 (Z)：电机反转，舵机左转

停止 (S)：电机不转，舵机正向

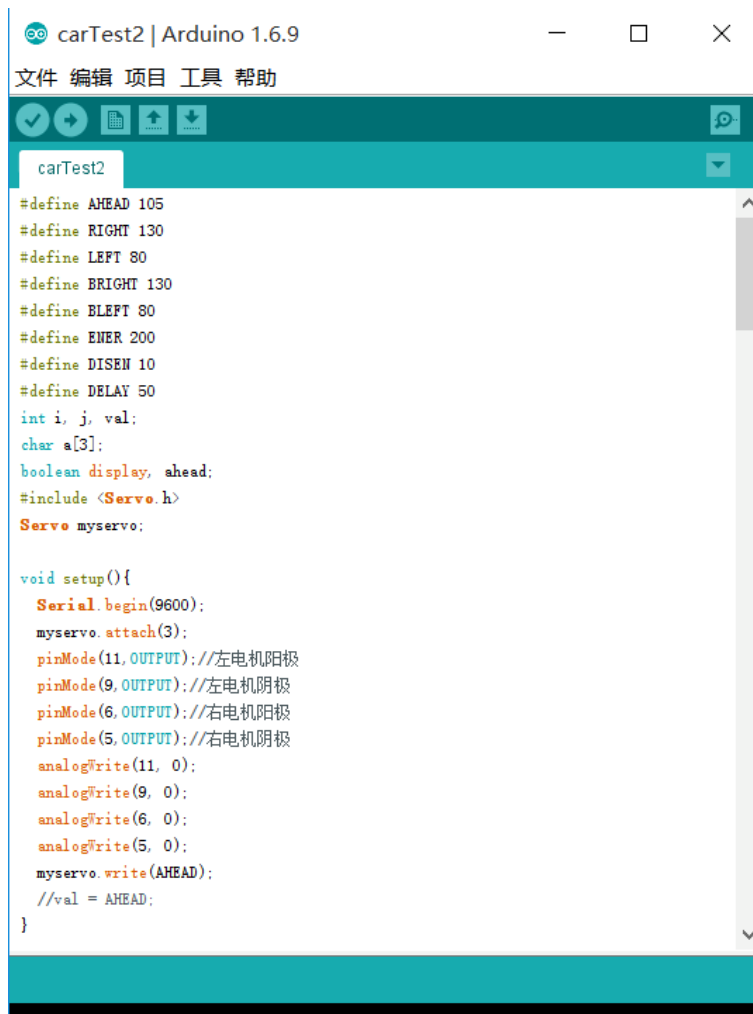
对于电机转速，我们通过控制发送至电机驱动板 I/O 端口的信号占空比来进行控制。


```

analogWrite(11, ENER);//将左边电机的正极置为高，占空比为ENER
analogWrite(9, 0);//保持左边电机的负极为低
analogWrite(6, ENER);//将右边电机的正极置为高，占空比为ENER
analogWrite(5, 0);//保持右边电机的负极为低

```

对于舵机的控制，我们使用了一个控制舵机的库——“servo.h”。先给自己的舵机建立一个对象 myservo，并用 myservo.attach(int I/O) 来绑定，通过函数 myservo.write(int angle) 来控制舵机的角度。部分代码如下图所示。



```

carTest2 | Arduino 1.6.9
文件 编辑 项目 工具 帮助

#define AHEAD 105
#define RIGHT 130
#define LEFT 80
#define BRIGHT 130
#define BLEFT 80
#define ENER 200
#define DISEN 10
#define DELAY 50
int i, j, val;
char a[3];
boolean display, ahead;
#include <Servo.h>
Servo myservo;

void setup(){
  Serial.begin(9600);
  myservo.attach(3);
  pinMode(11, OUTPUT);//左电机阳极
  pinMode(9, OUTPUT);//左电机阴极
  pinMode(6, OUTPUT);//右电机阳极
  pinMode(5, OUTPUT);//右电机阴极
  analogWrite(11, 0);
  analogWrite(9, 0);
  analogWrite(6, 0);
  analogWrite(5, 0);
  myservo.write(AHEAD);
  //val = AHEAD;
}

```

图 7：小车端代码变量常量定义及串口初始化

➤ 程序调试

在跟主机端一起调试之前，我们使用 SSCOM3.2 串口调试工具来进行调试。在用蓝牙跟 hc-06 建立了连接以后，选择对应的串口，发送指定的字符串，看下车的运行情况。

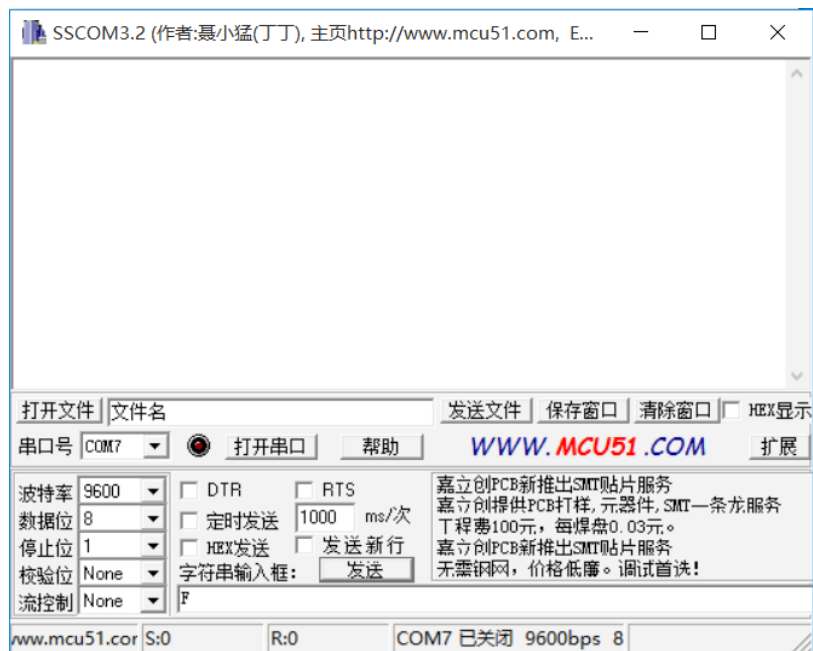


图 8：SSCOM 调试界面

经测试，发送指令后小车能按指令运动，则测试成功。

2.2 主机端实现

2.2.1 概述



图 9：工程流程图

2.2.2 图像处理

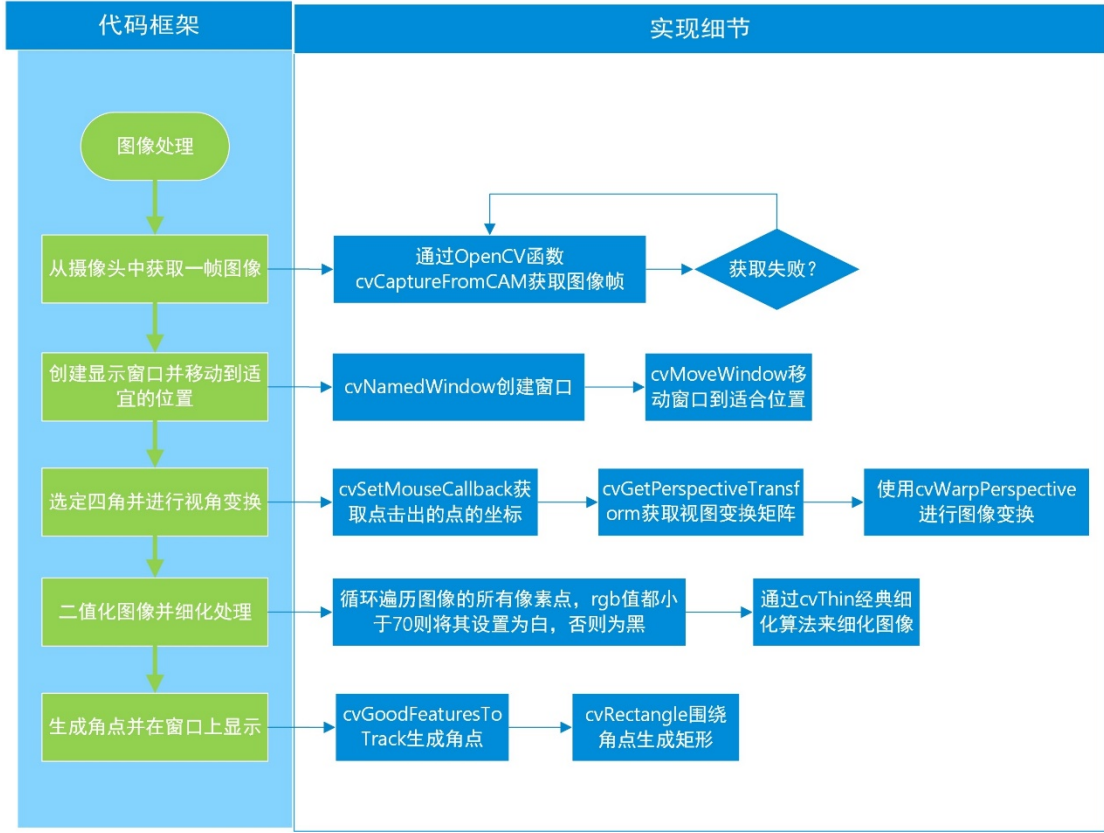


图 10：图形处理流程图

1、从摄像头中获取一帧图像

```
// 防止摄像头初始化失败
do {
    pCapture = cvCaptureFromCAM(1);
} while (!pCapture);
// 创建窗口并移动到指定位置
```

2、创建显示窗口并移动到适宜的位置

```
// 创建窗口并移动到指定位置
cvNamedWindow("colorfulImg", 1);
cvNamedWindow("lineImg", 1);
cvMoveWindow("colorfulImg", 0, 0);
cvMoveWindow("lineImg", 690, 0);
```

3、选定四角并进行视角变换

```
//step 1-4
cvSetMouseCallback("colorfulImg", mouseEvent);
```

```

void exchange(IplImage *img)
{
    if (step == 5)
    {
        newPoints[0] = cvPoint2D32f(0, 0);
        newPoints[1] = cvPoint2D32f(img->width, 0);
        newPoints[2] = cvPoint2D32f(0, img->height);
        newPoints[3] = cvPoint2D32f(img->width, img->height);
        cvGetPerspectiveTransform(originPoints, newPoints, transMat);
    }
    cvWarpPerspective(img, img, transMat);
}

```

4、二值化图像并细化处理

```

RgbImage  imgA(colorfulImg);
BwImage  imgB(lineImg);
for (int i = 0; i<colorfulImg->height; i++)
{
    for (int j = 0; j<colorfulImg->width; j++)
    {
        if (imgA[i][j].b<RGB && imgA[i][j].g<RGB && imgA[i][j].r<RGB)
        {
            imgB[i][j] = 255;
        }
        else
        {
            imgB[i][j] = 0;
        }
    }
}
cvThin(lineImg, lineImg, times);

```

5、生成角点并在窗口上显示

```

exchange(colorfulImg);
tmp = cvCreateImage(cvGetSize(colorfulImg), IPL_DEPTH_8U, 1);
tmp2 = cvCreateImage(cvGetSize(colorfulImg), IPL_DEPTH_8U, 1);
cvGoodFeaturesToTrack(lineImg, tmp, tmp2, corners, &mycount, 0.05, 70);
for (int i = 0; i < mycount; ++i)
{
    visited[i] = false;
    cvRectangle(colorfulImg, cvPoint((int)corners[i].x - 5, (int)corners[i].y - 5),
        cvPoint((int)corners[i].x + 5, (int)corners[i].y + 5),
        CV_RGB(0, 255, 0), 1, CV_AA, 0);
}

```

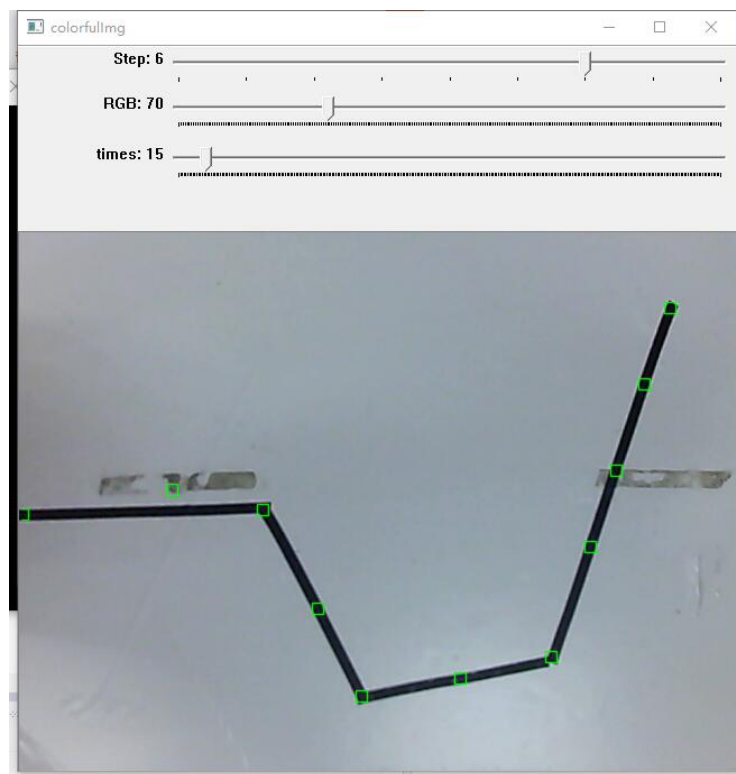


图 11：主机端生成的角点

2.2.3 跟踪小车

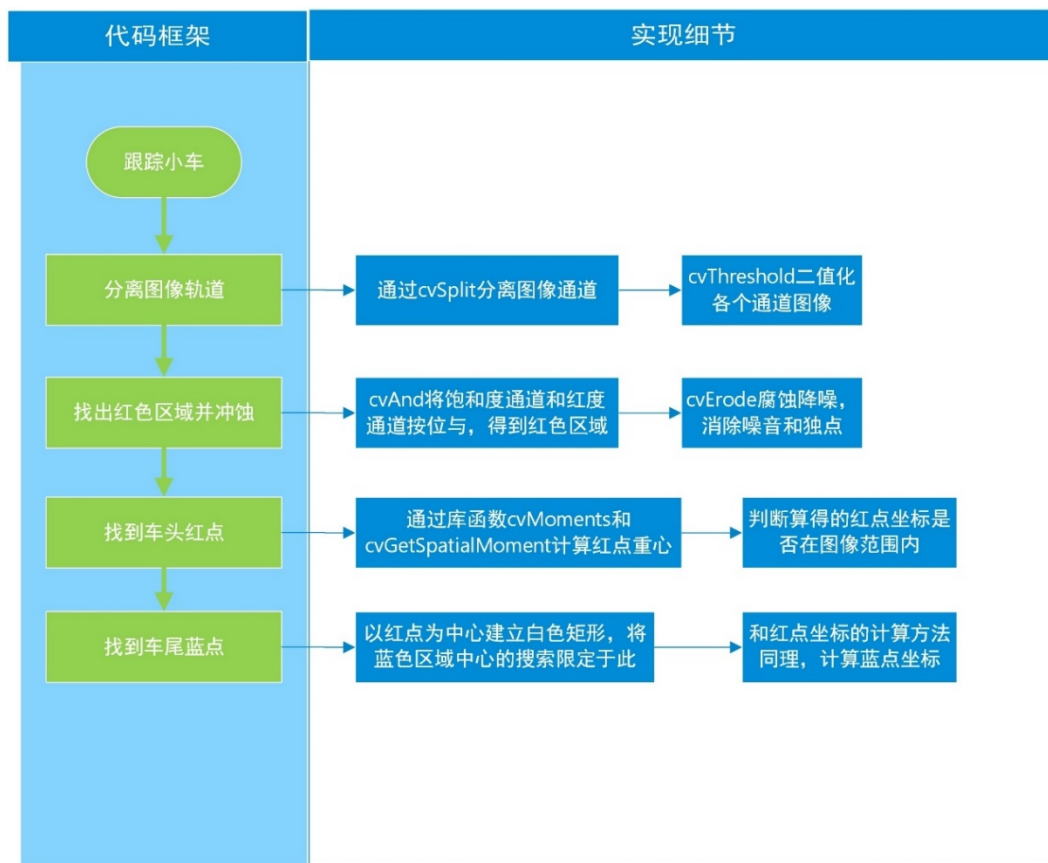


图 12：跟踪小车流程图

1、分离图像轨道，得到饱和度轨道和红值轨道，并二值化处理

```
cvSplit(img_pocessed, img_b, img_g, img_r, 0);
cvSplit(img_hsv, img_h, img_s, img_v, 0);
cvThreshold(img_r, img_r, 90, 255, CV_THRESH_BINARY); //90 需调试
cvThreshold(img_g, img_g, 90, 255, CV_THRESH_BINARY); //90 需调试
cvThreshold(img_b, img_b, 100, 255, CV_THRESH_BINARY);
cvThreshold(img_s, img_s, 90, 255, CV_THRESH_BINARY); //90 需调试
cvThreshold(img_h, img_h, 150, 255, CV_THRESH_BINARY_INV);
cvThreshold(img_v, img_v, 40, 255, CV_THRESH_BINARY);
```

2、找出红色区域并腐蚀降噪

```
//饱和度和红色二值化后的图像按位与
cvAnd(img_r, img_s, img_red);
//腐蚀，用于消除噪音和独点
cvErode(img_red, img_red, 0, 2);
```

3、找到车头红点中心

```
//使用cvMoments 求红色区域重心
CvMoments moments_red, moments_blue;
cvMoments(img_red, &moments_red, 1);
//cvShowImage("red", img_red);
double m00, m10, m01;
m00 = cvGetSpatialMoment(&moments_red, 0, 0);
m10 = cvGetSpatialMoment(&moments_red, 1, 0);
m01 = cvGetSpatialMoment(&moments_red, 0, 1);
//cvRound--将double 四舍五入转换成int
int red_x = cvRound(m10 / m00);
int red_y = cvRound(m01 / m00);
//找不到红点说明找不到车
if (red_x>img_size.width || red_x<0 || red_y>img_size.height || red_y<0)
{
    cout << "Fail to find the car!" << endl;
    return false;
}
```

4、以红点为中心建立一个小矩形，在这个范围内搜寻车尾蓝点

```
//以红点为中心建立一个白矩形，将蓝点的搜索范围限制于此
int length = 150; //length < 240
int x = (red_x - length>0) ? red_x - length : 0;
int y = (red_y - length>0) ? red_y - length : 0;
int width = (red_x + length < img_size.width) ? 2 * length : img_size.width - x;
int height = (red_y + length < img_size.height) ? 2 * length : img_size.height - y;
IplImage* mask = cvCreateImage(img_size, IPL_DEPTH_8U, 1);
cvSetZero(mask);
cvSetImageROI(mask, cvRect(x, y, width, height));
cvSet(mask, cvScalar(255, 0, 0, 0));
cvResetImageROI(mask);
cvAnd(img_s, mask, mask);
cvAnd(img_b, mask, img_blue);
cvErode(img_blue, img_blue, 0, 2);
```

2.2.4 发送指令

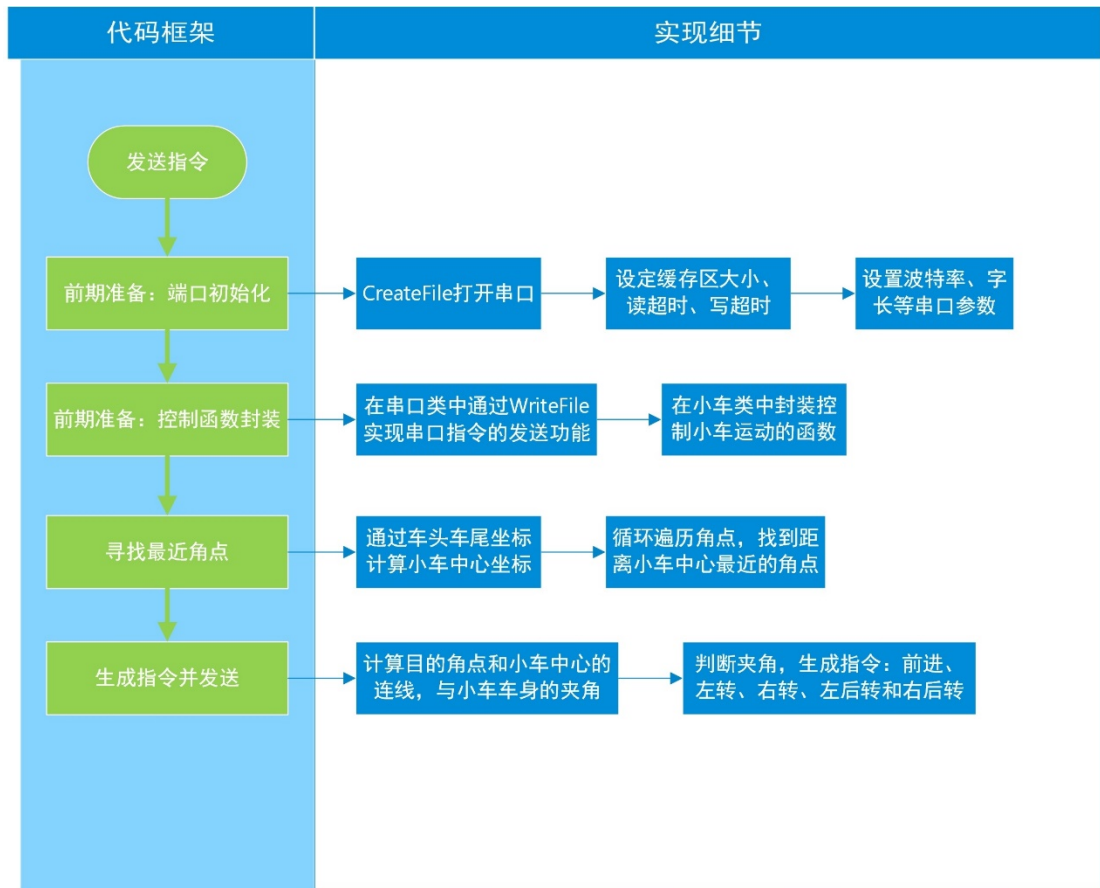


图 13：发送指令流程图

1、端口初始化

```
//打开串口
hCom = CreateFile(portName, //串口号
    GENERIC_READ | GENERIC_WRITE, //允许读和写
    0, //独占方式
    NULL,
    OPEN_EXISTING, //打开而不是创建
    FILE_ATTRIBUTE_NORMAL | FILE_FLAG_OVERLAPPED, //重叠方式：异步I/O结构
    NULL
);
if (hCom == INVALID_HANDLE_VALUE)
{
    dwError = GetLastError(); //取得错误代码
    cout << portName << " Opened Fail. Error code: " << dwError << endl;
}
else
{
    bComOpened = TRUE;
    cout << portName << " Opened Successful." << endl;
}
```

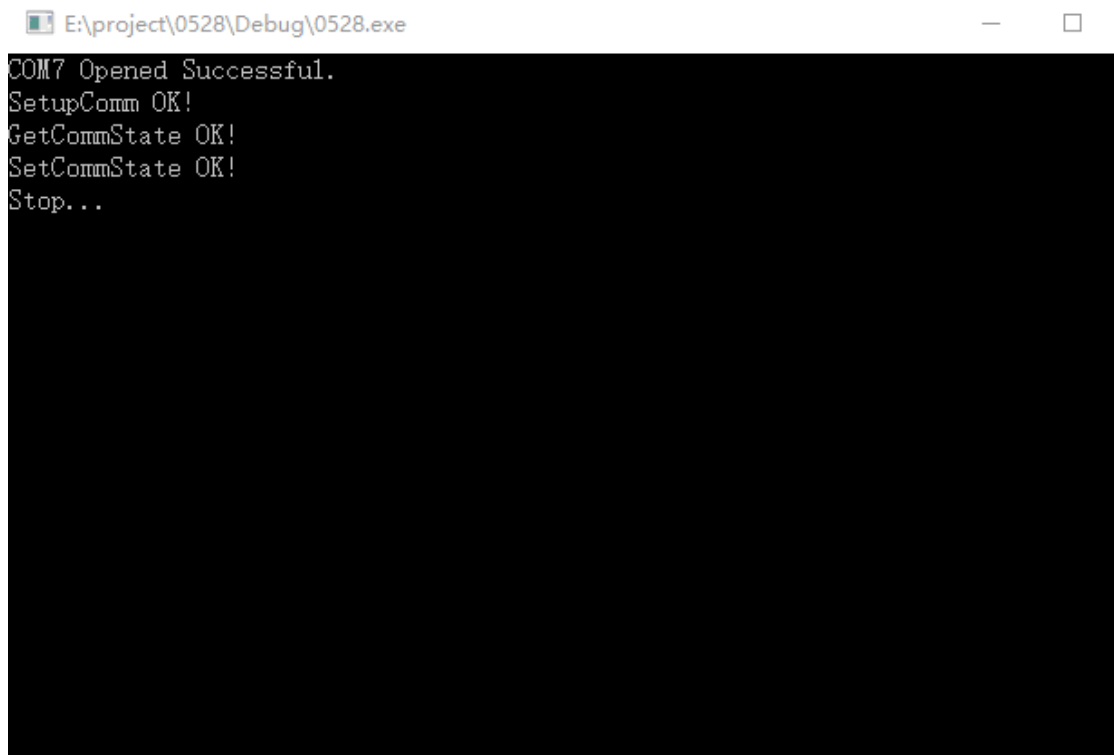


图 14：端口初始化显示界面

2、编写串口指令发送函数 Port.Send()

```
if (!WriteFile(hCom, pSendBuffer, dwWantSend, &dwRealSend, &wrOverlapped))
{
    dwError = GetLastError();
    if (dwError == ERROR_IO_PENDING)
    {
        while (!GetOverlappedResult(hCom, &wrOverlapped, &dwRealSend, FALSE))
        {
            if (GetLastError() == ERROR_IO_INCOMPLETE)
            {
                cout<<"Sending..."<<endl;
                continue;
            }
            else
            {
                cout << "Send Fail. Error code: " << dwError << endl;
                ClearCommError(hCom, &dwError, NULL);
                break;
            }
        }
    }
}
```

3、小车类的函数封装（除了左转、右转，相似的还有前进、停止、左后退和右后退）


```

void move_r()
{
    cout << "Right...\n";
    if (state == 'R')
        return;
    tmpSend = 'R';
    port.Send(tmpSend);
    state = 'R';
}
void move_l()
{
    cout << "Left...\n";
    if (state == 'L')
        return;
    tmpSend = 'L';
    port.Send(tmpSend);
    state = 'L';
}

```

4、主函数中，寻找最近角点

```

CvPoint2D32f centre = cvPoint2D32f((carPos_head.x + carPos_tail.x) / 2, (carPos_head.y + carPos_tail.y) / 2);
//初始化，找到最近的角点
if (lastPoint == -1)
{
    for (int i = 0; i < mycount; ++i)
    {
        if (lastPoint == -1)
        {
            lastPoint = i;
        }
        else if (getDistance(centre, corners[i]) < getDistance(centre, corners[lastPoint]))
        {
            lastPoint = i;
        }
    }
    visited[lastPoint] = true;
    return;
}

```

5、计算最近的角点和小车中心连线与车头车尾连线这两条线的夹角

```

Angle = calculateAngle(corners[minDis]);
CvPoint2D32f head, tail;
head.x = carPos_head.x;
head.y = carPos_head.y;
tail.x = carPos_tail.x;
tail.y = carPos_tail.y;
if (getDistance(corners[lastPoint], head) > getDistance(corners[lastPoint], tail))
{
    visited[lastPoint] = true;
    cout << lastPoint << endl;
}
lastPoint = minDis;
instruction(Angle);

```

6、根据实时角度发送所需的指令

```

void instruction(double Angle)
{
    if (Angle<=15 && Angle>=-15)
    {
        car.move_f();
    }
    else if (Angle > 15 && Angle < 70)
    {
        car.move_r();
        cvWaitKey(20);
    }
    else if (Angle >= 70)
    {
        car.move_lb();
        cvWaitKey(100);
    }
    else if (Angle < -15 && Angle > -70)
    {
        car.move_l();
        cvWaitKey(20);
    }
    else if (Angle <= -70)
    {
        car.move_rb();
        cvWaitKey(100);
    }
}

```

3 特色及优化

3.1 项目特色

1. **算法效率高。**采用即时发送即时反馈的算法，并未加入一些延时或是等待的命令，使得小车运行流畅，没有迟钝或是卡顿的现象。
2. **小车运行精度可控。**即可调整小车运行的容错率，以平衡精确度与稳定性。机理为通过调整角点的密度，实现运行路线的精度可控性，测试过程中我们使用过 50，70，100 三种角点密度（数字代表角点之间的图像像素数量）
3. **简约风格设计。**在提高美观度的同时也可以减少图像识别带来的错误，方便调试提高稳定性。
4. **一键开关式控制。**实现小车运行的全自动化

3.2 优化方案

3.2.1 转弯半径

- **问题：**自制小车过大，致使转弯半径过大。
- **解决方案：**小车很“大”可以说是我们的一个特色，但是为了追求“高富帅”还是要付出一定的代价。针对这个问题，我们拓展了小车的控制指令，加入了后退以及后退转弯等一系列新的指令。在急转弯处进行后退转弯可以减小转弯半径。

3.2.2 小车防抖

- **问题：**后退指令的加入导致小车易在阈值处前后抖动。
- **解决方案：**为了克服这个问题（即前后交替挪动而不行进），我们又加入了一些和滞回比较器或是施密特触发器原理相同的控制算法，使前进和后退的阈值之间存在一定的缓冲区，不会在前进瞬间又发出后退指令，成功解决了问题。

3.2.3 舵机响应速度

- **问题：**舵机转向较慢，跟不上发送指令的速度，导致转弯不及时。
- **解决方案：**在主机端给转弯指令加入一段延时，并根据实际情况调整延时时间，已达到最流畅的效果。

3.3 后续改进

- **问题：**

由于需要实时监测小车与轨迹间相对位置，闭环控制小车行进，所以算法的复杂度较高，对于主机负荷较大，受限于电脑配置，偶有运算跟不上的问题出现。所以仍需改进算法，减少资源依赖，这样也可以提高小车的最高运行速度。

➤ 问题：

小车高度过高，导致在摄像头拍摄角度偏斜时，小车行进路线有偏斜（取决于 β 及 h ）。后续可以通过调整相机角度或是降低色块位置改进。

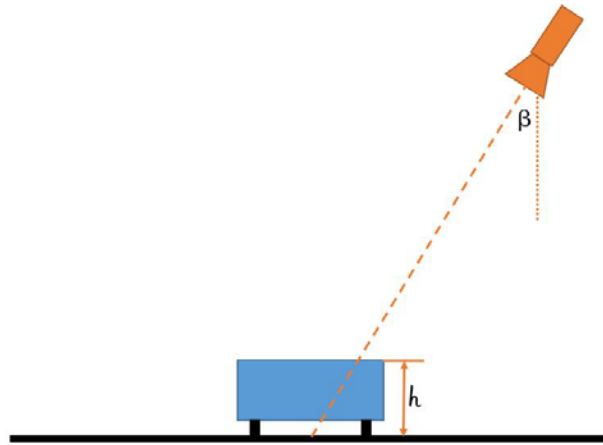


图 15：小车过高问题示意

4 组员体悟

➤ 包伟铭

本学期我们运用课程相关知识，实现了电脑主机自动控制小车循迹走黑线。综合考虑，我们选择了自制小车。电脑端基于 OpenCV 库函数，调用 Windows API，通过蓝牙连接小车发指令，并实时捕捉摄像头图像，处理实现闭环控制。小车端我们选用了相对熟悉的 Arduino 开发平台。由于自己之前没有接触过 OpenCV，在前期调试时，也碰到了不少问题，但辛苦是有价值的，收获良多。

小车端由于已修课程及课外已有所接触，但动手实践，实现这些功能，过程中仍会出现许许多多意料之外的突发状况。前期反复设想心中再有底，到实际动手是仍会出现或多或少的问题需要尝试解决。我想这也是工程的魅力所在。而只有潜下心，全情投入，面对种种出现的问题有恒心、毅力，不浮躁，轻言放弃，最终才能在这最困难也是最有收获的阶段真正收获更多。

感谢组员们全情参与及前人、老师的指导帮助，在实现整个项目的过程中，我收获了一份记忆深刻的体验，在相关理论、技术应用能力提高的过程中，更收获了许多兴奋与快乐！

➤ 庞博

通过学习《工程实践与科技创新 II-B》这门课程，使得我对于库函数的使用有了更深的了解，使我对编程的理解和使用都有所提高。这也是我第一次接触图像处理，本来以前认为非常深奥的东西，通过这一次的学习之后，我有了新的看法，也不再对这一块儿一无所知。这门课让我学到了很多。

实际成果的诞生给予了我们所有成员极大的成就感，提高了我学习编程语言的积极性和自信心，让我感受到了设计与创新的乐趣。

最后感谢小组成员的共同努力与配合，感谢老师的指导。

➤ 刘兴阳

工科创这个课程给了我们一个将理论知识付诸实践的机会，只有真正让代码在小车上跑起来，才会发现理论上优秀的代码要能够在实际环境中适用还需要一个很长的调试过程。原来，我们觉得非常合理的算法总会因为实际情况的限制（例如光线情况不佳、马达动力不足）而没有设想的效果好。当然，在发现问题与解决问题的过程中我们也学习了很多解决问题的思路，跟队友之间也建立了深厚的友谊。

我们小组自己采购并组装了小车，在这个过程中我们对硬件有了更深层次的认识，例如 Arduino 的开发、电机驱动板的使用等等。这是一个很有意思、很值得回忆的经历，仿佛回到了小时候自己组装四驱车的样子。

最后，非常感谢老师的指导，也非常感谢耐心而热情的助教。此外，我的队友（也是我的室友）也做出了很多的贡献，跟他们一起合作是一个非常快乐的经历。

➤ 罗晶

参与这个课程，实现小车走黑线是一个不小的挑战。之前我从未接触过 openCv 的图像处理功能，对蓝牙串口、小车端单片机也知之甚少。经过一个学期不间断的学习和探索，我终于初步掌握了这些知识并进行了实践。

总体做下来，最大的感受就是，写代码的时间远远没有调试的时间长，开始时小车会完全不按照预计的位置走，转弯也出现了很大的问题，十分考验自学能力和开脑洞的能力。还有在蓝牙模块花了不少时间找问题，电脑蓝牙时常故障，最终的解决方案是使用了一个外接的蓝牙适配器。

感谢老师的指导和帮助，助教在最后守着我们录视频也辛苦了。当然要感谢我的室友们，在很多环节如蓝牙连接、调试 PC 端代码中遇到困难和故障，是他们给予我支持和鼓励。

5 参考文献

- [1]. 学习 Opencv 2.4.9 (一) ——Opencv + vs2012 环境配置.
<http://blog.csdn.net/liukun321/article/details/38373277>. [2014-08-04]
- [2]. OpenCV 库函数大全.
<http://wenku.baidu.com/link?url=g7JNDFu0NeB6r6ddJVXWC2EJq68C2E11sqVSUHmf01-e22TtwalPwUQP4TKKIVgekfxJhx7p6cscJnq08IX71iopf4f1P-ypBdam8fG9LIq>. [2013-2-26]
- [3]. OpenCV 入门教程. <http://blog.csdn.net/column/details/opencv-tutorial.html>. [2014-6-22]
- [4]. OpenCV 官方 API. <http://docs.opencv.org/2.4/modules/refman.html>. [2016-5-25]
- [5]. Arduino Uno Introduction.
<https://www.arduino.cc/en/Main/ArduinoBoardUno>. [2016. 6. 24]
- [6]. Arduino Meya Introduction.
<https://www.arduino.cc/en/Main/ArduinoBoardMega>. [2016. 6. 24]
- [7]. 分享 ARDUINO 各版本引脚图 (2014 版). <http://www.geek-workshop.com/thread-11826-1-1.html>. [2014-12-4/2015-7-29]
- [8]. DIYer 修炼: 舵机知识扫盲. <http://www.guokr.com/article/5292/>. [2011-01-17/2015-7-29]
- [9]. Arduino 模块库文件自带 Example
- [10]. 《工程实践与科技创新 II-B》课程材料

6 致谢

感谢《工程实践与科技创新 II-B》课程组张士文等老师及助教的指导帮助。

感谢该门课程让我们获得这一份充实、有意义的学习体验。学习探索的过程中, 在实践能力、工程意识、团队协作、创新精神等方面都深感收获良多。

感谢课程中小组成员积极参与、交流探讨, 分工协作, 共同进步。