工科创2b设计报告

小组成员：杨子腾 程云龙 陈屹扬 刘权

一、项目名称

全自动的走迷宫小车

二、小组成员信息及分工

杨子腾（组长），负责通信、对接、算法设计、地图小车制作、网页制作。



程云龙，负责图像处理，算法设计与编写(python)



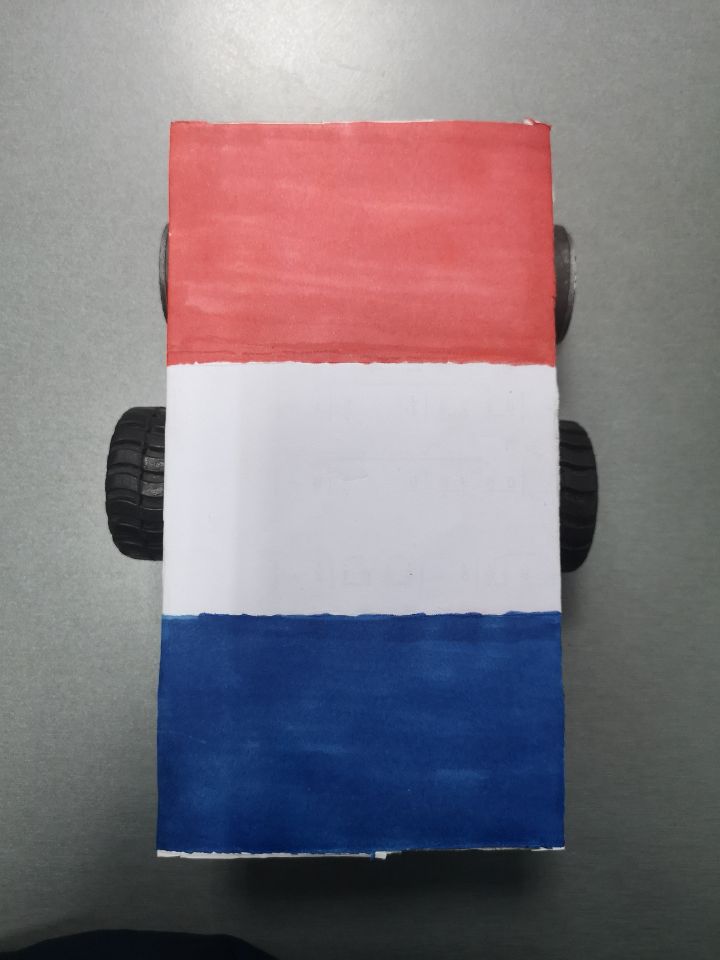
刘权，负责图像处理，算法设计与编写(C++)，算法优化（协助程云龙）



陈屹扬，负责单片机程序编写，通信、小车制作、撰写设计报告。



小车照片：



三、项目介绍及整体完成情况

计算机通过对地图和小车的图像识别，实时控制小车运行，从起点走过迷宫到达终点。为了实现这个目标，我们用分别用python和C++编写了两个功能类似的计算机程序，用energia编写了单片机程序，连接了电路，制作了小车的外壳。最终完成了课程目标。

在最终的测试中，我们实现了基本功能，即小车从起点到终点的全自动运行。我们还实现了迷宫起点和终点可以任意变更，以及地图可以在程序运行过程中旋转和位移。我们的程序只需要根据光照环境设置一个参数，可以说接近了自动适应光照条件，无需调节参数这一要求。

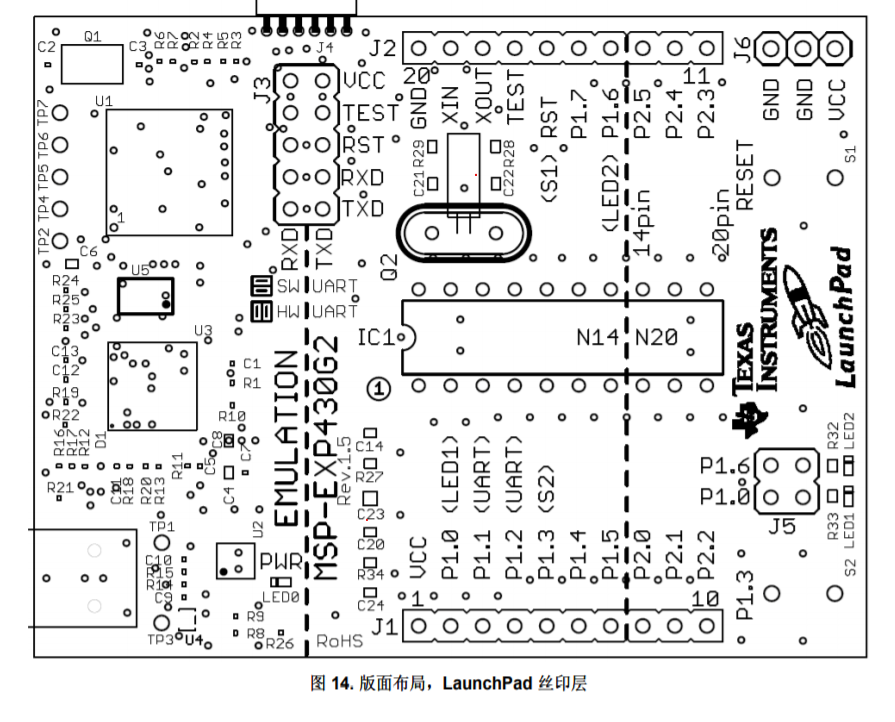
四、整体系统结构

**硬件部分：**编写单片机程序，通过单片机控制小车两个驱动轮的转速，实现小车的前进、后退和转向。通过蓝牙模块接收计算机发送的指令并控制小车进行相应的动作。

**软件部分：**在计算机上通过摄像头采集迷宫画面，监控小车位置，进行图像识别和路径规划。实时通过蓝牙对小车发布运动指令，控制小车走出迷宫。

**地图部分：**在白纸上用黑线画出迷宫，在四角贴上绿色圆片，用于图像识别中对地图的识别，迷宫的道路宽度大于小车宽度，边界均为直线，转角均为直角。

五、硬件部分介绍：



**接线：**

开发板如上图，端口3 RXD接蓝牙TXD, 端口4 TXD接蓝牙RXD, 再接上端口1 VCC和端口20 GND通电。再将开发板插到小车上，接线完成。

**代码：**

通过观察电路板和测试，得到9号端口对应右轮前进，10号端口对应右轮后退，13号端口对应左轮前进，12号端口对应左轮后退。在程序开头用#define定义数字端口的变量名，便于程序的编写。程序中设计7种动作，编写7个函数，对应7个控制命令‘a’到’g‘:

a: 前进，调用函数go(), 传递的参数为前进时间，单位ms，为了便于控制定为500ms，实现方法为左右轮前进端口输入高电平。

b: 左转，调用函数turnleft(), 传递的参数为旋转角度，为了便于控制定为30度，实际旋转角度可能会因电池电压的不同和纸面摩擦力的不同而有所差别。实现方法为右轮前进端口高电平，左轮后退端口高电平。

c: 右转，调用函数turnleft(), 与上述类似，区别在于左轮前进端口高电平，右轮后退端口高电平。

d: 左转微调，调用函数turnleftslight(), 无传递参数，控制小车左转约10度。

e：右转微调，调用函数turnrightslight(), 无传递参数，控制小车右转约10度

f: 前进微调，调用函数goslight(), 无传递参数，控制小车前进100ms

g: 后退微调，调用函数back(), 无传递参数，控制小车后退100ms

由于控制端口全程高电平的时候轮子转速过快，通过程序实现100hz的方波，高低电平各占一半，即不断重复写高电平，delay, 写低电平，delay的过程。上述所有的函数中相关端口都是写入这种方波，以次将小车的速度控制在便于操控的程度。

主程序每次检查串口是否由信息传递，若有，读入字符，判断是’a’到’g’中的哪条命令并调用函数，然后继续等待下一个命令。

六、软件部分介绍：

软件部分介绍分为两部分，一部分为python程序，一部分为C++程序，两个程序都能成功控制小车走出迷宫，而我们在演示和视频录制的时候采用的是效果更好的python程序。

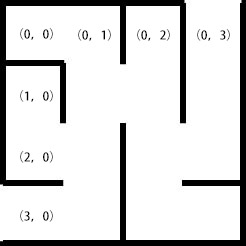
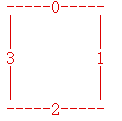
**python程序介绍**

我使用了python语言，利用python中的opencv接口实现该程序，程序介绍如下：

首先实现了两种数据结构：

* Node（）

因为大部分的迷宫中格子形状都是矩形，所以对每一个迷宫，我们都可以将其视为一个一个矩形组成的。每一个Node数据结构就代表着一个矩形，在其中储存的是每一个矩形的基本信息。Node中包含着以下几种矩形的基本信息：



1. self.pos: 蕴含着每一个矩形的位置，如上图所示，在左下角的节点位置为（3，0）.
2. self.state: 蕴含着矩形的各边信息，state是一个4个元素的数组，其中下标所对应的边如右图所示。其中如果对应下标处的边为黑边，那么数组中该元素即为1.例如左下角顶点（3，0），其state = [ 1, 0, 0, 1 ].
3. self.\_l\_son, self.\_r\_son: 指向该节点的下两个可以经过的节点。例如图中的（2，1）节点的下两个可以经过得到节点即为（2，0）和（1，1）.
4. self.to\_end: 其值为真或者假，其中记录的是从迷宫起点通向终点的路径中是否存在该节点。
5. self.len: state中值为1的元素的个数。
6. self.rnode: 判断该节点是否为关键节点。关键节点的意思是如果在该节点小车需要判断是否变向，那么该节点即为关键节点。图中的（3，1），（2，1），（2，0），（1，1），（0，1），（1，2），（2，2），（3，2），（2，3）为关键节点。

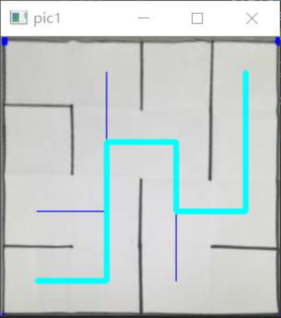
在节点中，我还定义了两种方法：

1. figure\_state: 判断该节点的self.state信息。
2. Is\_a\_rnode: 判断该节点是否为关键节点。

* Tree（）

正如tree的字面意思，该数据结构中保存的即为一颗遍历完整个迷宫的由关键节点组成的树。这棵树中有一个头节点（self.head），代表着迷宫的起始位置的节点。

在这棵树中，我也定义了多个方法，介绍如下：

1. search\_Node: 这个方法的功能即为从self.head开始遍历迷宫，搜索每一个关键节点。该方法的运行截图如下：

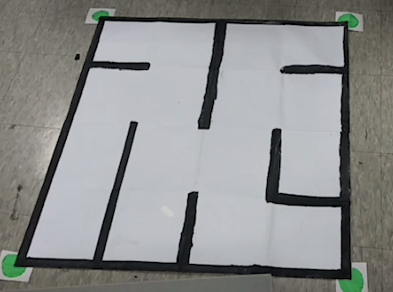
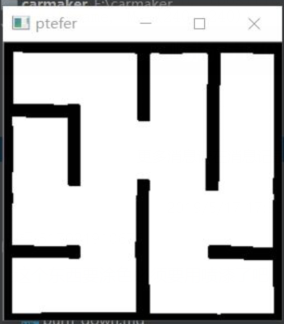
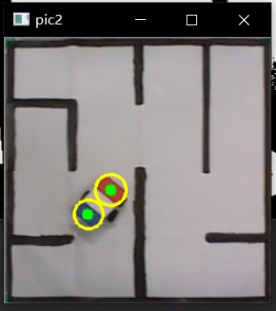
其中左下为迷宫起点，右上为迷宫终点。该方法的实现思路即为对每一个节点，判断其是否为关键节点，如果是关键节点，那么遍历该节点除了进入该节点的边的其他边，判断是否可以由这条边进入下一个关键节点，如果是，那么将该节点的\_l\_son或者\_r\_son指向下一个关键节点，再对下一个关键节点调用该函数。这个函数的实现算法实际上就是一个dfs算法。

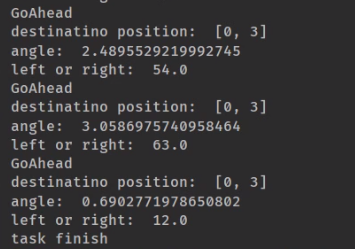
1. findPathToEnd: 该方法在利用search\_Node方法获取了整张地图的关键节点之后调用，并且它实际上是一个包装函数，实际实现功能的函数为pfindPathToEnd。该方法利用pfindPathToEnd返回的一条可以从起点走到终点的路径，在对应的frame中画出该路径。如上图，浅蓝色粗路径即为findPathToEnd画出的路径。

工具函数：

1. pfindPathToEnd: 本函数由findPathToEnd调用，其功能是在有节点信息的基础上找到一条可以走完迷宫的路径。本函数遍历tree中的所有关键节点，是一个递归函数，终止条件是找到了迷宫的终点，返回的上一个调用就知道终点的信息，进而将本次调用中的节点的to\_end信息修改为真。
2. drawline： 一个工具函数用来在frame中绘制遍历节点的路径。如上图中被部分覆盖的细蓝色线条。

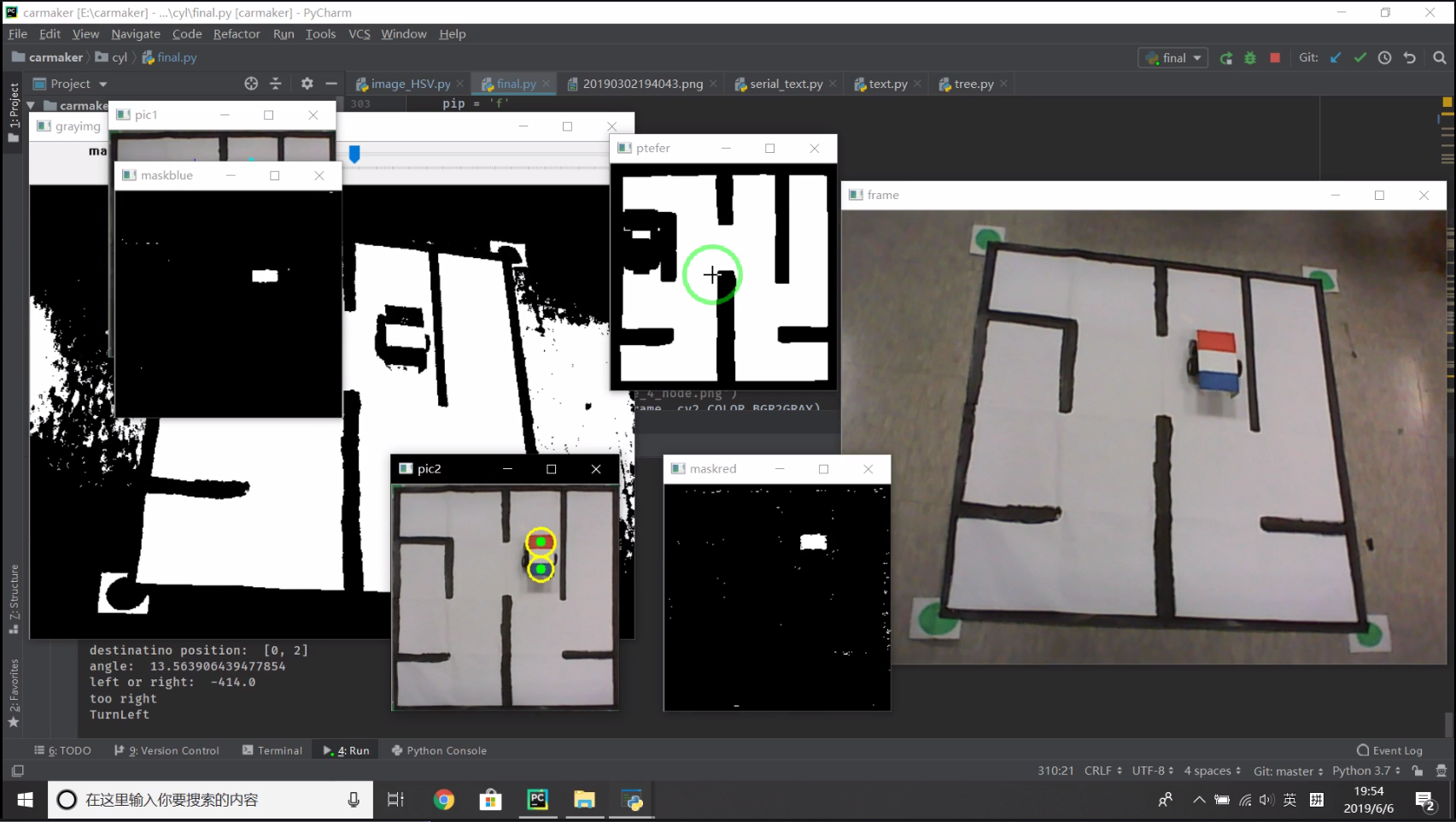
下面是主函数的运行 ：

1. 初始化
   1. 初始化蓝牙，连接上小车。
   2. 从摄像头中获取图片，将获取的frame转化为灰度图。
   3. 利用滑动条调整mask值，并用其将frame转化为二值化图片。
2. 调用GetTheMaze()函数获取地图信息。
   1. 识别图片利用的是图片四角的绿色方格，根据cv2.findCountours()获得的轮廓信息判断地图的方位。
3. 将获得的地图转化成含有240\*240个像素点的二值化图片，对这张图片进行两次腐蚀，去掉所得图片中黑色部分的少量白色像素点。以下为所获得的地图：
4. 不断重复过程2，3，获取地图的较好位置直至用户按‘y’确定。
5. 以迷宫的起点构造上述树形结构，运行search\_Node找到关键节点信息，运行findPathToEnd找到迷宫的正确路径。图片为介绍树时使用的图片。
6. 调用GetCarPosition获取小车的位置，并且在相应frame中将小车中的红色与蓝色中心标注出来。
7. 判断小车与下一个关键节点相差的距离，以及小车前进方向是否指向下一个目标节点，利用蓝牙发送对应指令控制小车运动。
   1. angel为小车红色中心，蓝色中心和下一目标节点所夹得夹角。
   2. destination position为目标节点的位置。
   3. left or right为判断小车左转右转的依据，数值为正则需要左转，为负则需要右转。



1. 当小车的中心与关键节点的中心接近到一定距离，调整关键节点到下一个关键节点。
2. 在小车运动的过程中同时不断获取迷宫的信息。
3. 重复过程5-9直至小车运行到终点。输出“task finish”.结束程序。
4. 如果用户中途按下了‘q’键，则程序立即终止。

以下为程序运行总截图：



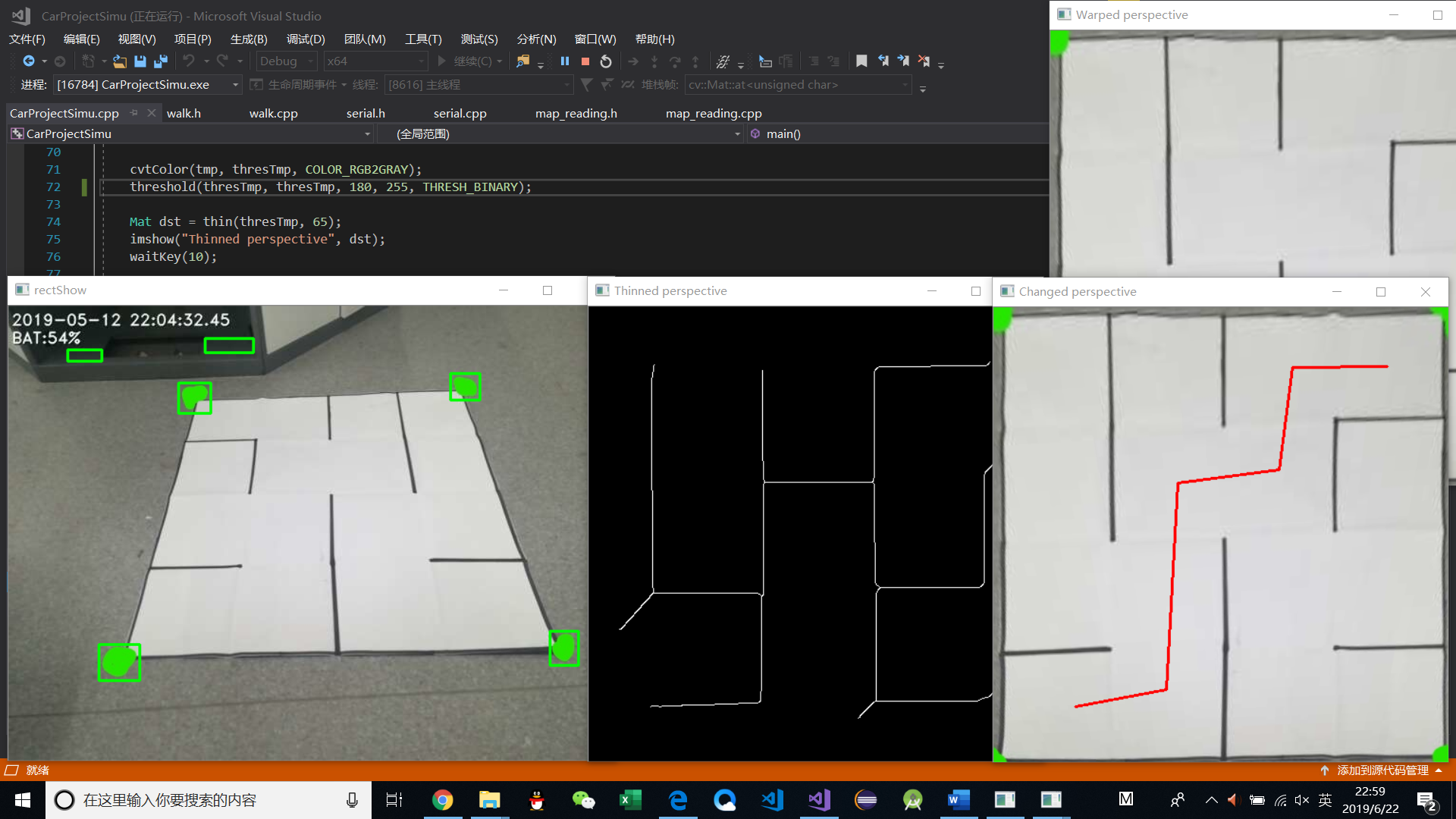
**C++程序介绍**

注1：由于是展示结束后才写的报告，小车不在手边，因此图片中是使用了预存的照片进行演示的效果，代码中注释了所有与小车和摄像头相关的语句。

注2：此程序运行时间相对较慢，但基本不用修改即可适应其他任意类型的迷宫，泛用性较强，但由于时间原因，未制作多个不同迷宫进行测试，在以后的工作中可以进一步测试。

1. 主程序思路：
   1. 读图：
      1. 开启摄像头，等待图像稳定；用于调整摄像头位置的一段代码，按q退出（可注释掉）；展示第一张图片（为左下图）；
      2. 以绿色色块为标志（主要以色度区分），找出绿色色块；
      3. 腐蚀+膨胀消除小块干扰；
      4. 以得到的边界为依据变换视角（为右上图）；
   2. 计算路径：
      1. 起点终点程序中给定；
      2. 使用指导中的细化函数细化，迭代多次完成细化（为中下图）；
      3. 基于dfs寻路，在细化图中找到起点附近到终点附近的路径，返回路径点：
         1. dfs中，在递归返回路径上每几个像素记录一个点，得到近似路径；
         2. 将近似路径上那些落在前后两点构成的直线附近的点去除，即去除直线段中间的所有点；
         3. 将聚集在一起的点取均值，即将一个转折处留存的点融合为一个；
         4. 返回处理完成后留存的所有点。
      4. 画出处理完成的路径（为右下图）。
   3. 循环运行中为小车导航：
      1. 读入一帧，以色块为标志得到小车位置与方向；
      2. 以小车位置、方向为根据计算出小车需要进行的动作；
      3. 使用蓝牙通信将指令传输给小车；
      4. 循环整个过程，到达终点时跳出循环。
2. 其余程序思路：
   1. 源文件的头文件中均有详细解释。

程序运行截屏：



七、系统测试情况：

经多次测试，小车运行流畅，前进快速，转弯平滑而精确，能正确到达终点，途中没有进入错误的岔路，也不会碰到迷宫的边界，始终走在道路的中间。

可以任意更改迷宫的起点和终点，比如起点和终点对调，起点和终点设置在迷宫中间而非边缘，小车始终能通过正确的道路从起点走到终点。

小车运行过程中地图的旋转和位移不影响地图识别和小车的运行，由于地图较大旋转麻烦，我们通过旋转、移动摄像头来达到相同的效果。在摄像头的抖动、旋转和移动中，小车仍然正确走完了路线。

八、系统的不足及可能的改进

**不足**

1 在程序运行中，整张地图，特别是四个角的圆片，必须出现在摄像头画面中，否则程序就会出错。

2 程序不能自适应光照环境，若不调节那一个光照参数，在不同的光照环境下可能导致程序失败

3 单片机程序只接1 收由字符代表的命令而不接收参数比如前进时间或旋转角度，延长前进时间等操作需要在电脑程序中重复发送命令。

**对应的改进方案**

1 在电脑程序中修改代码，当地图被部分遮挡无法识别时，使用之前未被遮挡的地图图像，结合当前的小车位置，继续运行程序。或者暂停发送命令，直到地图不再被遮挡。总之加入合适的错误处理代码防止程序报错而直接终止运行。

2 可以设置地图上白纸的颜色为参考颜色，在不同的光照环境下自动调节相关参数，直到经参数处理后白纸的颜色为程序设定的标准白色。通过这种方法实现自适应光照环境

3 在单片机程序中添加代码读取串口中的字符串类型数据，并转化为数据类型传递到函数中，实现前进时间和旋转角度的指定。

九、小组成员的自我评价

刘权：

**自我评价**：本课程中，我主要负责的部分是电脑端的程序与算法设计，这部分由我和程云龙共同负责，互相监督，互相学习，互相促进，最终我们两个分别实现了两个不同的程序。我在项目开始之初投入时间较多，对于识图、尤其是识别色块方面有了成果。后来我在其他部分中的成果进展较为缓慢，虽然实现了一套新的算法，但因为投入时间不足而导致稳定性不够，最终未录进视频。另外，在工作过程中，我和程云龙互相帮助解决问题的团队合作过程，我认为也是值得一提的。

**收获**：我在课程之初，对于如何能完成这样一个项目，没有任何想法，因此当初胡乱查找了很多资料，自己臆想把任务的难度看得很高，把自己唬住了。现在看，那些知识多数是用不到的。最终，我通过自己对于任务层级结构的分析，以及与组员间的学习与讨论，我逐渐明晰了一点：这个任务的实际难度实际并不大，对于运行的精度要求不强，解决问题需要的知识量也不多，问题在于如何明确定位我们所需的知识。这时候，互联网以及组员的帮助就显得特别重要了。幸运的是我身边有这群十分热心的组员，没有他们的帮助我是不可能完成这些任务的。我十分感谢我的组员们，也十分感谢课程与老师，给了我这次实现工程项目的经历。

杨子腾：

在前期的学习过程中，我和组员共同学习了OponCV函数库的各种原理和接口，同时也学习了energia单片机程序编写、蓝牙串口通信以及如何将小车程序与电脑程序进行联合与通信，整个学期相当充实，收获颇丰，组员们也特别厉害、好学，组员间相互监督相互学习，虽然我的贡献可能不如写程序的同学大，但在参与算法优化的讨论、分析同学的想法当中确实学到了太多以前完全想不到的方法；

另一方面作为组长，也兼顾了及时接收课程消息、组织小组定期进行学习等任务，我认为没有失职。

陈屹扬：

**自我评价:** 在前期的学习准备中，我初步了解了OpenCV和单片机的程序编写，由于我在大一下的工程学导论中尝试过arduino版的开发，对这方面较为熟悉，所以选择了单片机编程和通讯的工作。在此次项目过程中，我较早地写好了单片机程序，提供材料制作了便于识别的小车外壳，方便计算机程序的测试与改进。由于硬件有时会出现各种各样的问题，我能不断调试，耐心找出问题，不影响小组的进度。我积极参加每一次讨论，帮助测试程序，改进程序。感谢组员们的共同努力与合作，最后的成品有极高的完成度和稳定性。我也从中收获了知识与技能，相信在以后类似的小组合作中，我能做出更好的作品。

**建议：**工科创2b在第一个星期上了一节绪论课后，就交由我们独立完成。我建议可以在前半学期上几节课介绍所需的背景知识和基本制作方法。这有利于我们更快上手，少走弯路，效率更高，质量更好地完成任务。

程云龙：

**自我评价与收获**：在本次工科创课程中，我主要负责的是图像处理以及算法设计与实现这一部分。因为我之前就对opencv函数库有所接触，所以图像处理这一部分并没有对我们造成很大问题。在算法方面，我设计了一种基于树的迷宫遍历算法，可以根据“节点”（在软件介绍部分详细解释）迅速地遍历迷宫，运行情况良好。在整个课程中，我认为我的贡献还是比较大的。通过本次课程地学习，我也收获了许多，收获了如何与组员齐心协力完成一个一个分划出来的小目标，收获了对opencv函数库的更加熟练的使用能力，以及收获了对树这种数据结构的更深入的理解。

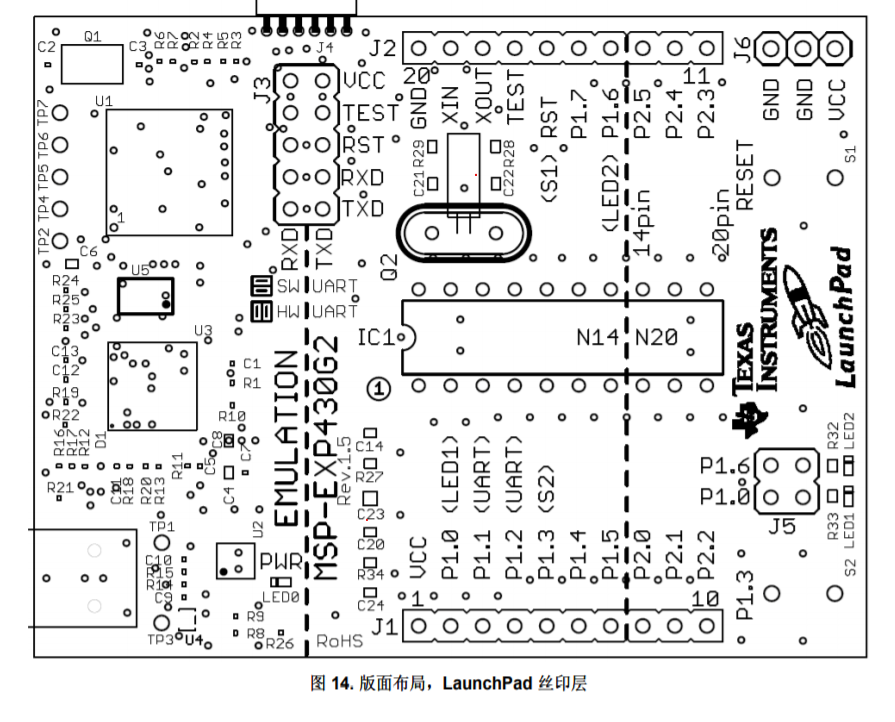
十、详细系统使用说明

将地图在地面上铺开，将摄像头与电脑连接，启动程序，显示出摄像头拍摄的画面，根据当前光照环境设置好唯一一个参数，调整摄像头位置直到整张地图，尤其是地图四个角的四张圆片出现在画面中并固定。

小车装上电池，打开开关，通过开发板的LED灯和蓝牙模块的LED灯确定小车工作正常。

程序的迷宫起始点已经设好，如有必要可以在程序更改。将小车放到起始点，运行程序，等待小车前进至终点，程序结束，小车走迷宫的任务完成。

十一、系统电路图



蓝牙

VCC

RXD

TXD

GND

十二、参考资料

1 工科创2b网站提供的参考资料中的energia和OpenCV相关资料

2 energia官方网站的文档 <http://energia.nu/reference/>

3 《MSP-EXP430G2 LaunchPad 实验板用户指南》

<http://www.ti.com.cn/cn/lit/ug/zhcu010c/zhcu010c.pdf>

4 OpenCV官网Tutorial <https://docs.opencv.org/master/d9/df8/tutorial_root.html>

注：由于硬件和软件程序源代码过长，没有放在设计报告中，详见网页版附录