

西电机器人夏令营 SUMMER CAMP 2018

3组技术报告

进度安排表

项目	工期	开始时间	结束时间
1、规划、方案讨论	1天	7.26	7.27
2、方案初步设计			
机械：确定机械方案、验证可行性、方案设计以及加工	2天	7.27	7.30
嵌入式：小车行走迷宫的功能实现，超声波传感器的距离测量，以及机械臂的驱动。	5天	7.27	8.01
算法：实现魔方颜色的识别，距离地视觉测量	5天	7.27	8.01
3、方案的讨论、迭代			
机械：改进机构、结构优化	8天	8.01	8.09
嵌入式：激光测距模块的实现	3天	8.01	8.03
更换方案	5天	8.04	8.09
算法：调整代码	4天	8.01	8.04
通讯协议	4天	8.05	8.08
4、测试与调试			
5、总装完成、优化	1天	8.09	8.10

目录

- 1 机械部分3
 - 1.1 设计动机.....3
 - 1.2 设计需求.....3
 - 1.3 设计方案.....4
 - 1.4 方案的优点与不足.....5
- 2 嵌入式部分7
 - 2.1 整体方案.....7
 - 2.2 运动学解算方法.....7
 - 2.3 机械臂与控制方案.....7
 - 2.4 功能模块说明..... 10
 - 2.5 难点与不足..... 11
- 第 3 章 算法部分 12
 - 3.1 开发环境介绍..... 12
 - 3.2 整体技术方案概述..... 12
 - 3.3 算法整体框架设计..... 12
 - 3.4 算法功能模块说明..... 13
 - 3.5 测试结果..... 13
 - 3.6 可优化方案..... 14
- 4 夏令营感想、总结 15

1 机械部分

1.1 设计动机

比赛大约分为两个部分，最主要的目标是使小车走完迷宫并尽可能用最少的时间，比赛最终的胜负就是用时最短的一队。第二个目标为从放置在迷宫角落的四个魔方中选出拼好的一个并使用机械臂等结构把魔方带走，如果成功的带走拼好的魔方，则会在原来所耗时间上减少一定时间作为奖励。

根据比赛的规则，小车在走出迷宫的过程中，需要用到红外测距等识别技术，还有摄像头，其中为了保证摄像头能清晰地识别出拼好的魔方，所以要把摄像头安置在一个视野宽广的地方，在保持稳定的前提下尽可能地提供更为宽广的视野。为了保证小车行进的平稳性，需要对小车底板上主控板，树莓派，电池以及其他结构进行合理布局，使其几何中心与质心尽可能重合来方便程序算法的编写。

1.2 设计需求

各需求的优先级

大概的设计时间

主要目标需求：1.尽可能的高速行驶。

2.重心降低，平稳行驶。

3.麦轮不打滑，四轮平均受力。转向平滑自如。

次要目标需求：1.摄像头位置适当，视野开阔无遮挡。

2.主控板陀螺仪位置应在质量中心。

3.机械臂运行速度适中，抓取成功率高。抓取魔方后固定良好。

1.3 设计方案

1.3.1 底盘机构

小车底盘采用3D打印材料亚力克板，为方便放置主控板以及使陀螺仪位于中心位置，加置二层平台，材料同样为亚力克版，用四根螺柱固定。为方便转向以及夹取魔方，轮子使用麦轮。电池重量和体积较大，故而放置于底板下面，树莓派放在底盘后部，摄像头放置在底盘前部。

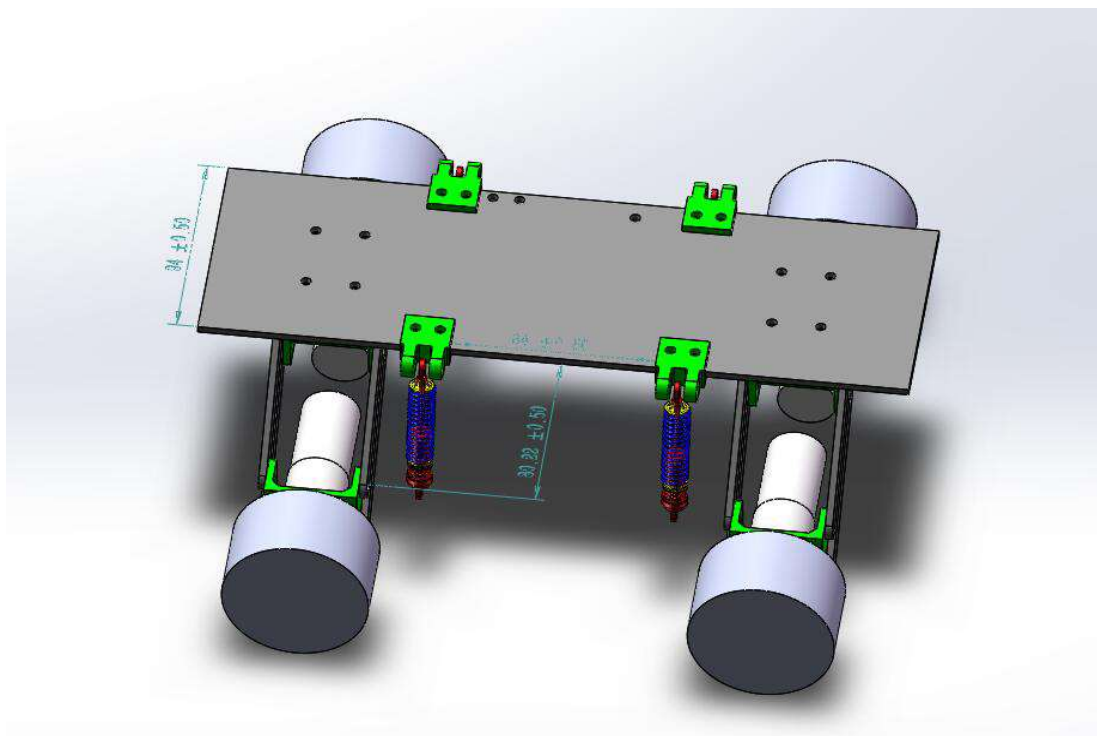


图1.1 小车底盘

1.3.2 机械臂设计

在实用为第一目的的基础上，我们抓取魔方的方案最后确定为拉伸杆勾取方案。使用齿轮齿条相配合，用步进电机驱动，进行拉伸。在齿条头部装上舵机，用舵机控制一个挡板，平时是竖起来的，在舵机控制下旋转后平放，再用步进电机控制齿条拉回，使魔魔方掉进早已准备好的盒子中。

相较于夹爪等方案，此方案有以下特点：

1. 结构较为简单，相较于多齿轮的机械夹爪和带有气缸并有一定危险

性的气动夹爪，此结构只使用了一次齿条齿轮配合。

2. 调整余地大，由于使用了勾取方案，对算法精准度的要求大大降低，不再需要十分精准的控制侧边与魔方间距。

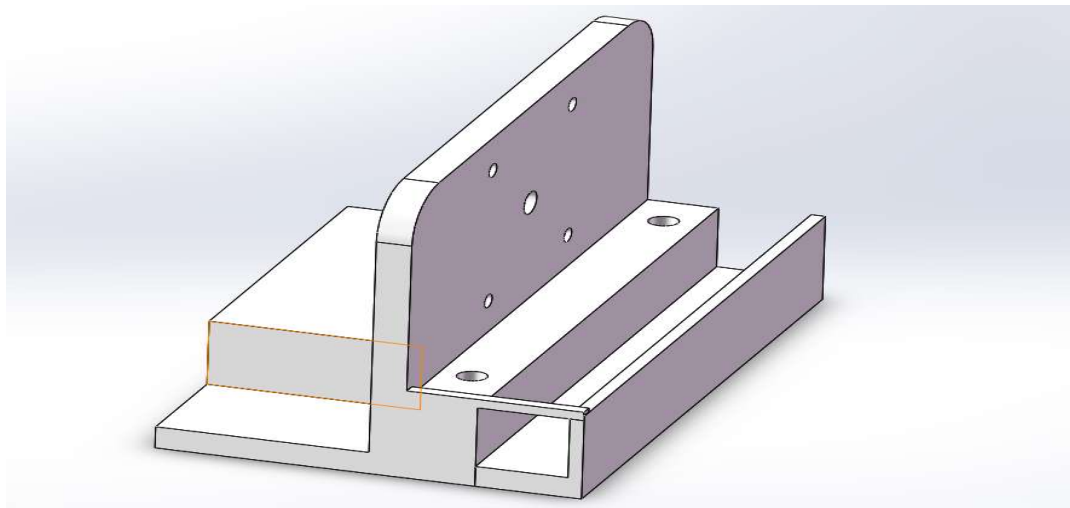


图1.2 齿轮导轨固定装置

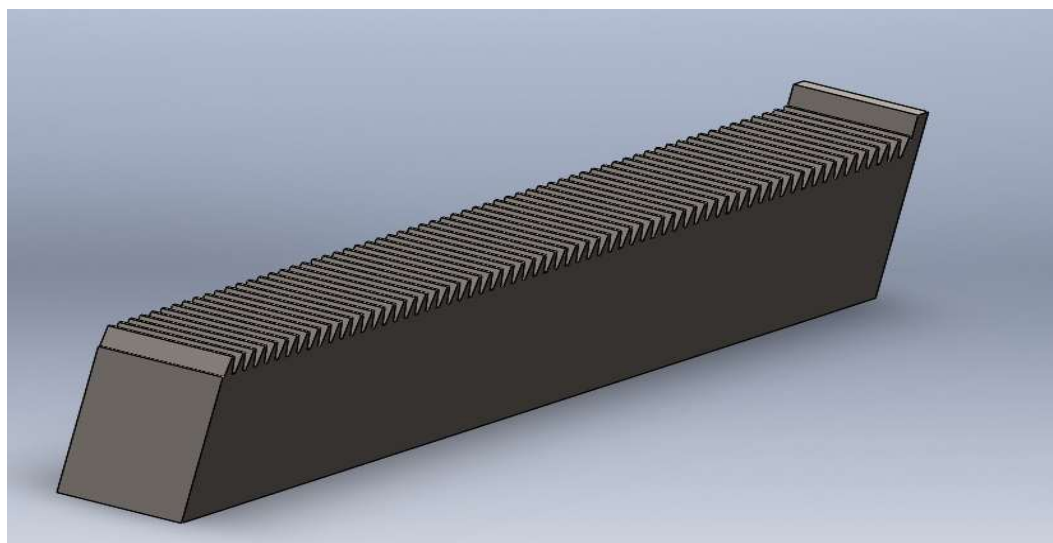


图1.3 导轨

1.4 方案的优点与不足

1. 由于板子空间有限，结构安放需要十分紧凑，打孔间距较短，不方便打孔。

2. 麦轮和底板连接结构稳定性较差，四轮受力不均，经常出现一轮空转

情况，抓地能力差。

2 嵌入式部分

2.1 整体方案

整体需要的算法与方案

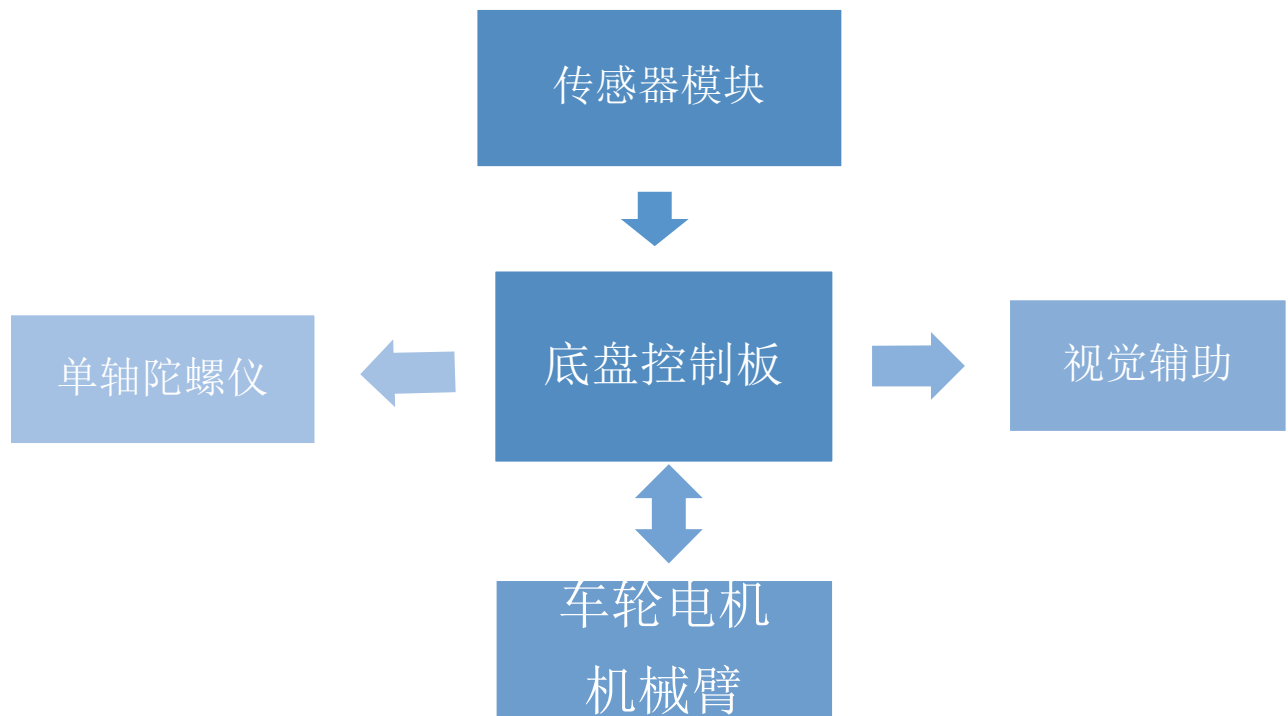


图2.1 整体算法流程图

2.2 运动学解算方法

我们并没有对麦克纳姆轮进行运动解算。我们是定时器驱动电机，并用基础定时器进行各个轮子的速度的测量，最后再用pid返回速度进行控制。并用IMU测量车的底盘转向角度【相对于y轴的0-360度的测量】。

2.3 机械臂与控制方案

机械臂与底盘所使用的控制方法

如何解决PID的超调、震荡等问题

2.3.1 机械臂控制方法

我们的机械臂的控制方案是利用直流电机带动齿轮推出导轨并使导轨顶端固定的舵机超过魔方的位置，舵机带动的挡板旋转平放后，再利用步进电机反转拉回导轨，使魔方可以掉入安装在车下方的承载装置中。

2.3.1.1 直流电机的控制方法

直流电机的控制方法，由通用定时器输出pwm波来驱动直流电机，利用转速和齿轮周长计算出需要转动的时长，再规定时长后结束pwm的输出，以达到推出和收回伸缩臂的效果。

2.3.1.2 舵机的控制方法

舵机的控制一般需要一个20秒左右的时基脉冲，该脉冲的高电平部分一般为0.5ms-2.5ms范围内的角度控制脉冲部分，总间隔为2ms。

我们购买的舵机的最大旋转角是180度，以正中位置设为零位，所以对应的控制关系如下：

0.5ms———— -90度；

1.0ms———— -45度；

1.5ms———— 0度；

2.0ms———— 45度；

2.5ms———— 90 度；

即

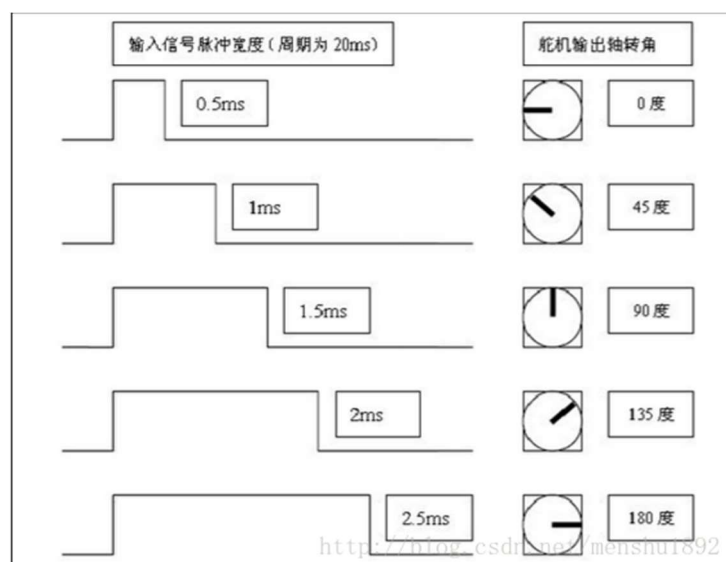


图2.2 信号脉冲宽度与舵机输出轴转角关系图

对于stm32f429，使用定时器pwm输出来控制舵机，并通过调整高电平占空比来控制舵机旋转的角度。

2.3.2底盘控制方案

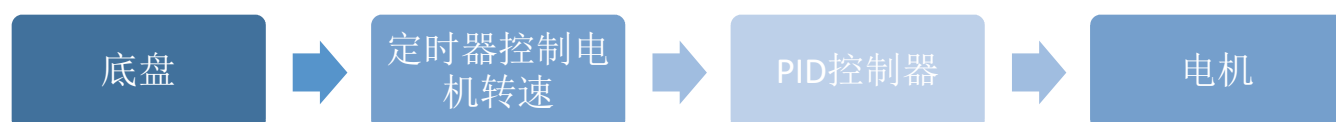


图2.3.1 底盘控制方案示意图

我们组对于底盘的控制十分简单，利用通用定时器输出pwm波，并通过控制pwm波的占空比控制各轮转速，再由陀螺仪计算出小车相对于给定y轴的偏斜角度，并用PID来保证小车在迷宫之内的直行以及转弯。

如图所示，是

利用传感器传回的距离值控制小车对于魔方抓取时位置的确定和伸出

的导轨长度。

2.4 功能模块说明

2.4.1 陀螺仪测角度模块

关于陀螺仪与加速度计的组合姿态解算我们具体参考文件
IMU(加速度计和陀螺)组合姿态解算.pdf



IMU（加速度计和
陀螺）组合姿态解算

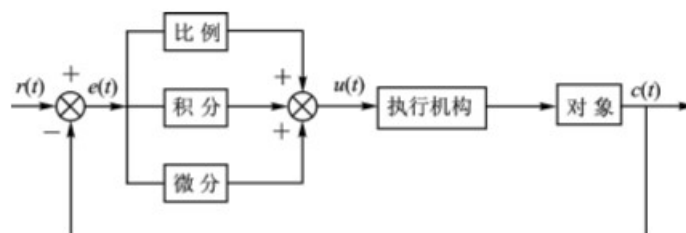
在此不做赘述。

下图为陀螺仪测量角度的部分代码：

```
215 void IMU_Get_Data()
216 {
217     int16_t ax, ay, az, gx, gy, gz;
218     uint8_t mpu_buff[14];
219
220     MPU6500_Read_Regs(MPU6500_ACCEL_XOUT_H, mpu_buff, 14);
221
222     imu_data.ax = mpu_buff[0]<<8 | mpu_buff[1];
223     imu_data.ay = mpu_buff[2]<<8 | mpu_buff[3];
224     imu_data.az = mpu_buff[4]<<8 | mpu_buff[5];
225
226     imu_data.temp = mpu_buff[6]<<8 | mpu_buff[7];
227
228     imu_data.gx = (mpu_buff[8]<<8 | mpu_buff[9]) - MPUGyroX_Offset;
229     // IMU_current_Gx = (float)imu_data.gx / 32.768 * pi / 180;
230     imu_data.gy = (mpu_buff[10]<<8 | mpu_buff[11]) - MPUGyroY_Offset;
231     // IMU_current_Gy = (float)imu_data.gy / 32.768 * pi / 180;
232     imu_data.gz = (mpu_buff[12]<<8 | mpu_buff[13]) - MPUGyroZ_Offset;
233     // IMU_current_Gz = (float)imu_data.gz / 32.768 * pi / 180;
234
235     MPU6500_getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
236     IMU_original_Ax = (float)ax / 16.384f * g;
237     IMU_original_Ay = (float)ay / 16.384f * g;
238     IMU_original_Az = (float)az / 16.384f * g;
239     IMU_current_Ax = IMU_original_Ax - Ax_offset;
240     IMU_current_Ay = IMU_original_Ay - Ay_offset;
241     IMU_current_Az = IMU_original_Az;
242     IMU_original_Gx = (float)gx / 32.768f / 4;
243     IMU_original_Gy = (float)gy / 32.768f / 4;
244     IMU_original_Gz = (float)gz / 32.768f / 4;
```

图2.4.1 IMU函数的部分代码

2.4.2 PID调整模块



2.4.2.1 本模拟反馈控制回

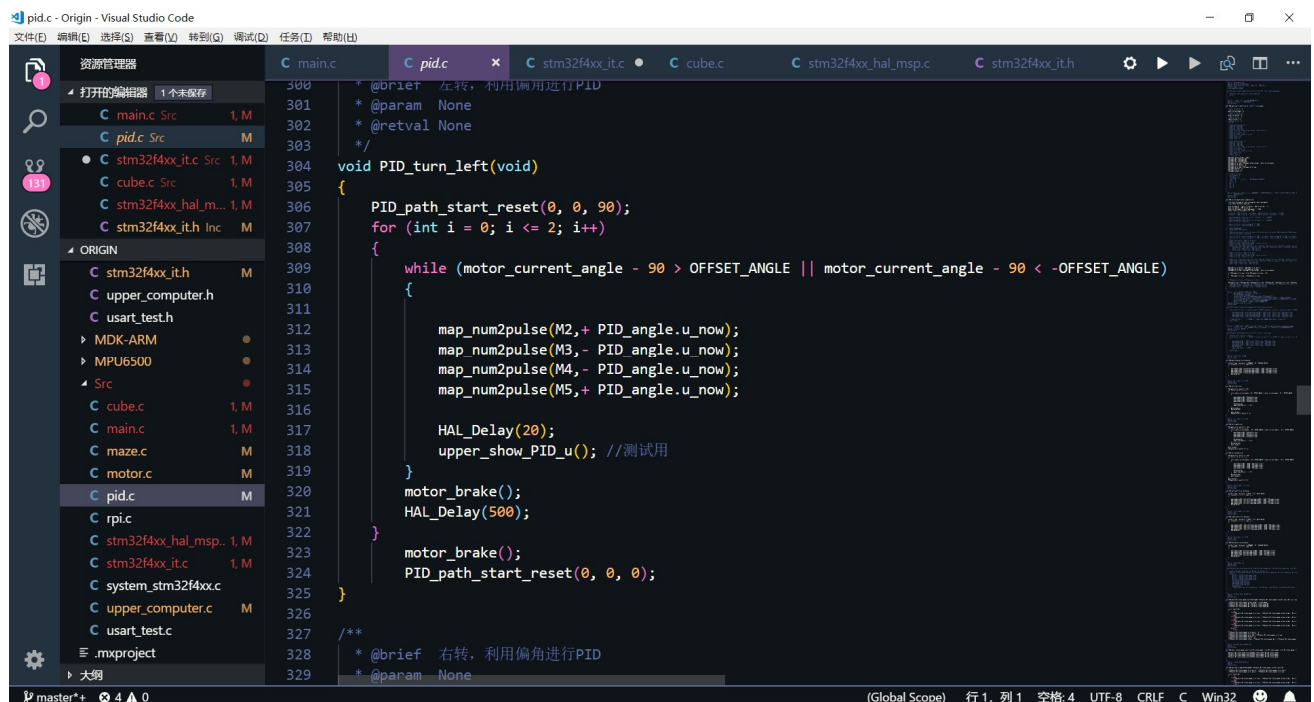
PID 调节器是一种线性调节器，它将给定值 $r(t)$ 与实际输出值 $c(t)$ 的偏差的比例(P)、积分(I)、微分(D)通过线性组合构成控制量，对控制对象进行控制。

PID 调节器的微分方程：

$$u(t) = K_P \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right]$$

式中 $e(t) = r(t) - c(t)$

下图这是PID控制小车向左旋转90°的部分代码：



2.5 难点与不足

- 1、 利用PID控制小车的旋转时，向左转和向右转精度不同，向右旋转90°会产生较大的误差。
- 2、 在执行平移走迷宫的时候，小车在向右平移的时候会自主后退。
- 3、 激光测距模块传输数据速度过慢。

第3章 算法部分

3.1 开发环境介绍

本次夏令营的算法开发环境配置主要是Linux环境与树莓派的配置，但是本次夏令营的主要在windows上Visual studio上配上Opencv的库，linux上的Opencv的配置是主要依据老学长的配置教程与网上的csdn教程。主要的坑是在配置树莓派的配置，问题有两个：

一是：安装树莓派的tf卡，第一次没成功后格式化后会可能破坏tf的分区，导致无法安装上。解决方案如下(网址)：

<https://blog.csdn.net/z3881006/article/details/72590539>

二是：树莓派的配置，一是树莓派Ip地址的问题-解决方案先检查树莓派是否安装好，再检查动态Ip与静态Ip，真的找不到可以用路由器的方法。

3.2 整体技术方案概述

整体方案的思路-先识别到魔方，然后发给电控与魔方的距离，和与小车的中心的距离（使小车对准魔方），因为团队方案与本身算法的能力水平的限制，没有做视觉辅助的功能，将摄像头放置于小车的侧边以配合机械的侧边抓取的方案。

主要技术的难点是识别魔方和与stm32的通信和算法的稳定性，算法要全程执行，时间过长可能会导致出bug。

3.3 算法整体框架设计

算法主体是在建立在开源代码的基础上，第一周是主要对Opencv算法的学习，第二周是在写魔方识别的代码方案，第三周是想完成与电控的

通信与修改代码。

本次夏令营的算法包含魔方识别与串口通信。魔方识别的主要方案是先识别到魔方的九个小块，再转到hsv的颜色空间，再识别每个小块的颜色方差，当方差小于某个阈值的就判断是同一个颜色。之后是对摄像头进行标定，标定的方案是依据张正友标定法，（标定程序见资料）得到相机的内参矩阵与畸变系数，来进行相机的测距。通过相机标定的结果来得出，相机焦距 f_x, f_y 。从而根据相似三角形来计算魔方与小车的中心的水平距离与直线距离。之后通过调用树莓派上的wiringPi上的serial库函数来实现其与stm32的通信，设置相同的波特率与通信协定来传输以上的数据。由于算法不需要对小车走迷宫进行视觉辅助，因此对二维码的识别就深入去做。同时在小车走迷宫的时候树莓派的程序需要自启动，因此再通过网线进行ssh连接就会不方便，因此需要对其程序进行自启动设置。

3.4 算法功能模块说明

在本次夏令营的算法模块没有实现的很明确，主要代码实现在cube.cpp的getube部分，

代码主体分为主函数与一个cube函数，对魔方照片进行识别分析，处理和对距离的测算。

而与stm32的通信是在主函数中实现的。对本次算法的实现没有达到预期的效果，算法的模块化也没有很好的实现。按预期的想法分为一个主函数和至少三个分函数，一个实现对魔方的抓取，一个是对魔方的颜色识别，一个是通过与通信实现。

3.5 测试结果

一是对魔方本身的识别的效果是挺好的，但是识别的距离要在50cm以内才会有比较好的效果，对颜色的识别由于方差的效果还是可以的。二是对通信的传输有点问题，在与电脑的通信来说是只传输两个值，不太稳

定。

3.6 可优化方案

之后优化应该是主要集中于算法的模块化与通信的优化，同时对二维码的识别的功能进行实现。主要是算法的模块化与通信，现在的算法模块化程度太低，出错误后查找的效率太低，同时代码的优化不方便。通信这方面对原理的还是不太清楚，目前是只调用模块，通信的效率不高，稳定性较差。

4 夏令营感想、总结

这次夏令营历时整整三周，占去了暑假的一多半，却也很快就结束了。

在经历了第一周staff们的教授课程以及老校区各种参观丰富了我们的见识，加深了我们对各自专业的理解，第二周第三周我们就进入了抓紧的备赛时期，我们也从一开始的各干各的转变到最后的一起熬夜爆肝的战友。

两个星期的备赛时间在我们我们第3组全体组员不分日夜的工作中一晃而过。我们各组人员在不断地修改方案，改进优化方案，并且追赶进度，最终完成了夏令营布置的任务，这三周真的很辛苦组员了。

虽然这不是我第一次接触项目管理，但确实是我第一次完全一人去管理一个项目，也让我明白了管理一个项目完全不像表面上看上去那么简单。

从了解目标，拆分各组任务到了解组员需求，进行物料采购，都必须认真地去思考决策；最开始由于对于规则的误解以及我对任务了解不够透彻，导致前期浪费了挺多时间，再者加上现实与理想的差距，让我们不得不多次修改方案，不过后来在一次彻底的交谈过后，我们的项目进度得到了完全的统一，项目推进速度也大幅度提高。

组员之间的沟通交流也是十分重要的，组内机械也曾经出现过商量好的方案理解错误导致打印件直接报废，不得不抓紧时间重新打印一个，也浪费了不少时间。

我在此次夏令营中不断领悟、不断学习如何去做一个合格的项管，并且觉得自己获益匪浅，与此同时我也获得了很多关于嵌入式知识。

以上是我作为一个项管对于此次夏令营的感受，下面是来自我们各部分的负责人的一些感想：

算法：感谢这群队友们，一起熬夜，一起做车，这次夏令营是自己第一次做主导所参加的活动，自己的进度时常困扰着自己，希望以后自己可

以变的更好。

电控：这次夏令营的可以说是给我打开了一个新世界的大门，从理论知识到积极动手实践，在机器人小车的制造过程学到了很多。不仅学习了嵌入式知识，而且认识了一群小伙伴。从一条咸鱼变成了一条有梦想的咸鱼。

机械：这次的夏令营，对我来说是一次很好的锻炼，但从我个人来说，并没有达到我个人所期望的目标。经过这三周的学习，我掌握了solidworks的使用方法及一部分的机械设计制造知识，这对于本身就是机械专业的我来说有很大的帮助。但也正是在这一过程中，我充分认识到了个人知识及能力方面的差距。可能比起所学到的知识，这一心态上的变化才是这次夏令营带给我的最大收获吧。

此次夏令营的两周内，我们光是抓取魔方的机械臂的方案在一周之内就修改了五次之多，最终做出来的成品更是有三批，我们选取了最简单、稳妥的方案来组装了我们的机械臂；同样，关于小车运动的代码及方式，我们也试了三种，最后选择了较为稳妥最后的一周，我们组的电控以及算法的同学几乎都是通宵Debug。我们在不断的翻车中成长，在缺少了两个队友的情况下，依然斗志高涨，努力将一切任务做到最好，不论比赛结果如何，感谢RM让我们相遇，让我们拥有一个如此充实的暑假。