

西电机器人夏令营

SUMMER CAMP 2018

一组技术报告

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	5天	7.28	8.1
机械：确定机械方案、验证可行性、方案设计以及加工	5天	7.28	8.1
嵌入式：	5天	7.28	8.1
算法：编写魔方识别程序，编写串口程序。	5天	7.28	8.1
3、方案的讨论、迭代			
机械：改进机构、结构优化	3天	8.2	8.4
嵌入式：改进算法，调试小车路线	3天	8.4	8.4
算法：优化并改进算法	3天	8.4	8.4
4、测试与调试	3天	8.5	8.7
小车各部分的调试	2天	8.5	8.6
小车总调试	1天	8.7	
5、总装完成、优化	3天	8.7	8.9
组装并改进结构问题	1天	8.7	
优化整体	2天	8.8	8.9

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3
 - 1.4 方案的优点与不足4
- 2 嵌入式部分.....5
 - 2.1 整体方案5
 - 2.2 运动学解算方法8
 - 2.3 机械臂与控制方案 10
 - 2.4 功能模块说明 11
 - 2.5 难点与不足 12
- 第 3 章 算法部分..... 13
 - 3.1 开发环境介绍 13
 - 3.2 整体技术方案概述 13
 - 3.3 算法整体框架设计 13
 - 3.4 算法功能模块说明 13
 - 3.5 测试结果 17
 - 3.6 可优化方案 17
- 4 夏令营感想、总结..... 18

1 机械部分

1.1 设计动机

为了完成使小车能在迷宫中识别四个魔方中完整拼好的那个并将其抓取，携带完整的魔方走出迷宫这一比赛目标。机械组需要组装出完整的智能小车并能稳定搭载其他组实现目标所使用的部件（主控板，树莓派，摄像头等）并加以改进，使小车能发挥其最好的性能，机械组还需要设计出能准确抓取并携带魔方的机械臂，以便能在算法组识别完整魔方后在嵌入式的控制下，抓取魔方。

1.2 设计需求

比赛的功能需求

- a. 小车能走出迷宫
- b. 机械臂可以抓取魔方
- c. 机械臂抓取到拼好的魔方

各需求的优先级 $a > b > c$

大概的设计时间 7.28~8.3 7天

1.3 设计方案

详细的机械方案

根据107参观工程车的机械臂产生思路，设计出由一个两个关节组成的关节式机械手，关于底盘，设计灵感来源淘宝：多功能水陆遥控车。前后减震，防拖地设计，大板载。这代车我们将它命名为小车三。

小车1是原型车，小车2是平板车，它仅仅停留在了设计层。这里很大一部分原因是电控和机械的不妥协：电控需要用到陀螺仪。

小车三的设计过程经过了电机架结构外形设计，前后轮分别设计，尺寸调整到相互配合，底盘钻孔的位置经过多次严格测量，更改多次，争取不用手钻保证板载的稳定性。

底板经过一次大的改变:由于前后轮轮距大于左右轮（5cm）部分影响了电控解算，人为增加困难。并且此时对于美观的要求也更高。顺应要求我们进行了第二次改底盘。

成品：板高120mm；左右轮间距215mm；前后轮间距230mm；

稳定性分析

小车整体稳定性较高，没有出现晃动及断裂的情况，之前由于涉及问题后轮不太紧密，已经通过改造底盘和调整修复了这个问题，小车的机械臂较重，会略微影响小车的稳定，但在测试中未产生过大影响。

1.4 方案的优点与不足

优点

- 1.机械臂组成简易分明，结实厚重，易于电控的控制，损坏的概率小，易于发现并调整产生的问题。
- 2.机械臂夹取的强度大，携带模仿稳定，不易掉落。
- 3.改装后的车外型美观缩小了体积，使小车更加灵巧便于转向。

缺点

1. 机械臂质量体积略大，可能会使小车的重心不稳，抓取过程中可能会使小车晃动。
2. 固定小车轮子的枢纽不紧，有松动，有时会脱落。

2 嵌入式部分

2.1 整体方案

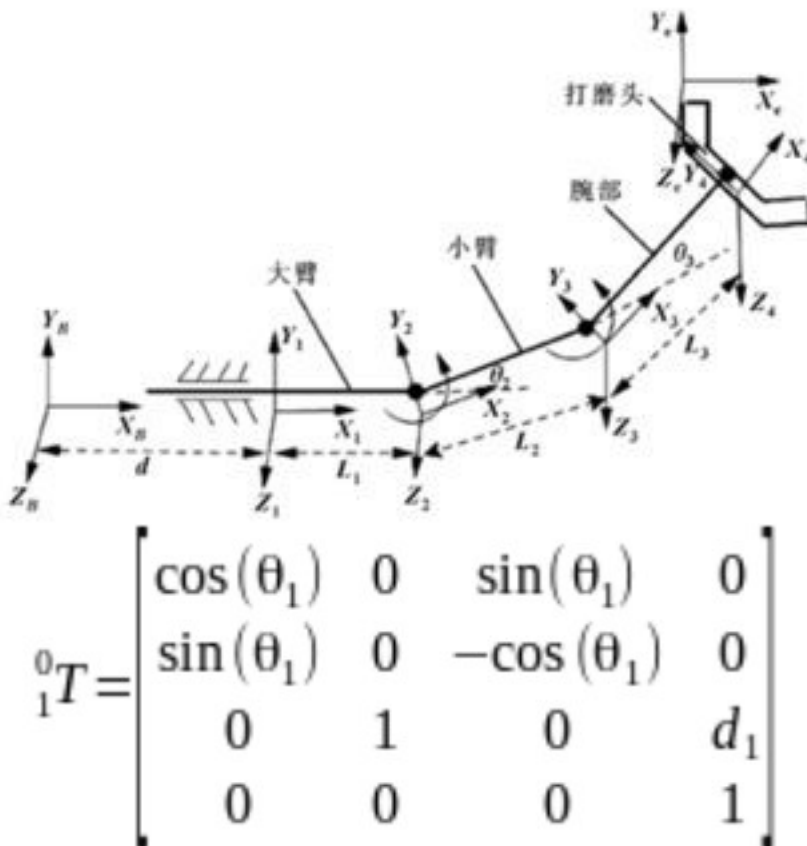
机械臂位姿解算

每个机械臂的每个轴可以用四个关节参量去表示，即连杆之间的扭转角，相邻轴之间公垂线的长度，连杆偏距，以及两个连杆的夹角

这样给机械臂的基座设一个零坐标系，以连杆的方向设立x轴，以轴的方向设立z轴（通过右手定则确定y轴的方向）再给第一个轴设一个1号坐标系，第二个轴设2号坐标系，以此类推，最后末端执行器设为n号坐标，再根据每个关节的四个关节参量去确定不同坐标系之间的位姿，通过矩阵的连乘就可以得到末端执行器相对于基座的位姿

在我们的机械臂里面，除了连杆之间夹角是可变的，其他的都是固定的

摄像头得到魔方的位置后经过一系列变换可以得到魔方与机械臂基座的位姿关系，这样令机械臂末端执行器的位姿态矩阵与魔方的位姿态矩阵相等，就可以求出不同连杆之间的夹角，再通过改变占空比去使舵机转过相应的角度即可，这样，通过位姿解算的方法可以使得机械臂可以精确的抓到魔方



$${}^1_2T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2_3T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PID算法

由于计算机控制是一种采样控制，它只能根据采样时刻的偏差值计算控制量。

在计算机控制系统中，PID 控制规律的实现必须用数值逼近的方法。当采样周期相当短时，用求和代替积分、用后向差分代替微分，使模拟 PID 离散化变为差分方程。

- (1) 数字 PID 位置型控制算法
- (2) 数字 PID 增量型控制算法

1.位置型控制算法

$$\text{PID控制律 } u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right]$$

为了便于计算机实现，必须将上式变换成差分方程。因此，假设 T 是采样周期， k 为采样序号。我们作如下近似：

$$\begin{cases} \int_0^t e(t) dt \approx \sum_{i=0}^k T e(i) \\ \frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T} \end{cases}$$

提供了执行机构的位置 $u(k)$ ，称为**数字PID位置型控制算法**

将近似表达式带入得数字PID的位置型控制算式：

$$u(k) = K_p \left[e(k) + \frac{1}{T_I} \sum_{i=0}^k T e(i) + T_D \frac{e(k) - e(k-1)}{T} \right]$$

2.增量型控制算法

$$\text{由位置型控制算法 } u(k) = K_p \left[e(k) + \frac{1}{T_I} \sum_{i=0}^k T e(i) + T_D \frac{e(k) - e(k-1)}{T} \right]$$

$$\text{易知 } u(k-1) = K_p \left[e(k-1) + \frac{1}{T_I} \sum_{i=0}^{k-1} T e(i) + T_D \frac{e(k-1) - e(k-2)}{T} \right]$$

增量满足 $\Delta u(k) = u(k) - u(k-1)$

$$\text{于是有 } \Delta u(k) = K_p [e(k) - e(k-1)] + K_p \frac{T}{T_I} e(k) + K_p \frac{T_D}{T_I} [e(k) - 2e(k-1) + e(k-2)]$$

增量式控制算法的优点

(1) 增量算法不需要做累加，控制量增量的确定仅与最近几次误差采样值有关，计算误差或计算精度问题，对控制量的计算影响较小。而位

置算法要用到过去的误差的累加值，容易产生大的累加误差。

(2) 增量式算法得出的是控制量的增量，例如阀门控制中、只输出阀门开度的变化部分，误动作影响小，必要时通过逻辑判断限制或禁止本次输出，不会严重影响系统的工作。而位置算法的输出是控制量的全量输出，误动作影响大。

(3) 采用增量算法，易于实现手动到自动的无冲击切换。

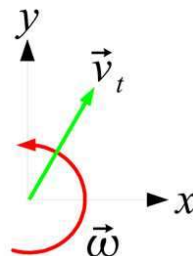
(4) 利用增量算法，也很容易得出位置算法 $u(k) = u(k-1) + \Delta u(k)$

2.2 运动学解算方法

麦克纳姆轮解算

一、底盘运动的分解

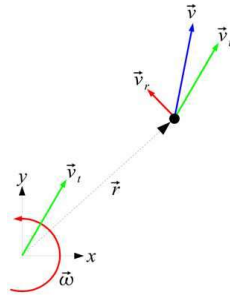
- 刚体在平面内的运动可以分解为三个独立分量：X轴平动、Y轴平动、yaw 轴自转。如下图所示，底盘的运动也可以分解为三个量：
- v_{tx} 表示 X 轴运动的速度，即左右方向，定义向右为正；



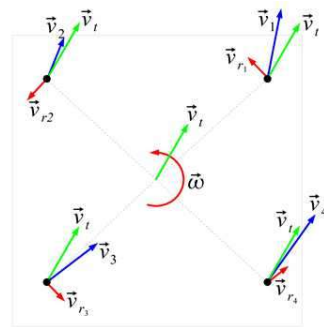
- v_{ty} 表示 Y 轴运动的速度，即前后方向，定义向前为正；
- Ω 表示 yaw 轴自转的角速度，定义逆时针为正。
- 以上三个量一般都视为小车的几何中心相对于地面的速度。

二、计算出轮子轴心位置的速度

- 定义：
- r 为从几何中心指向轮子轴心的矢量；
- V 为轮子轴心的运动速度矢量；
- V_r 为轮子轴心沿垂直于 r 的方向（即切线方向）的速度分量；
- 那么可以计算出：
- $V = V_t + \omega r$



- 分别计算 X、Y 轴的分量为：
- $V_x = V_{tx} - \omega r_y$
- $V_y = V_{ty} + \omega r_x$



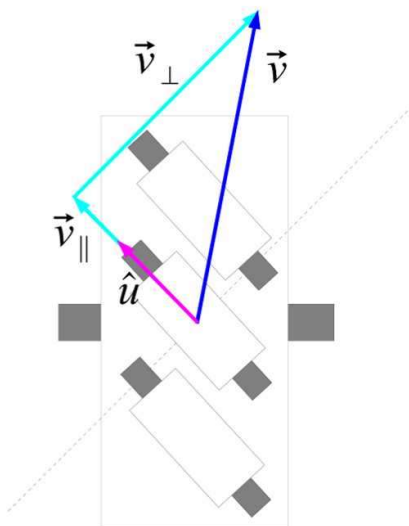
- 同理可以算出其他三个轮子轴心的速度。

三、计算辊子的速度

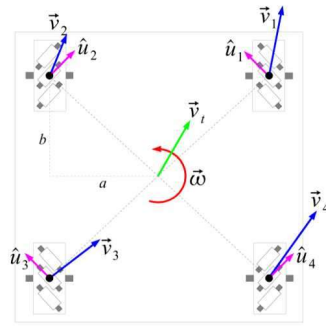
根据轮子轴心的速度，可以分解出沿辊子方向的速度 V_p 和垂直于辊子方向的速度 V_v 。其中 V_v 是可以无视的，而

$$V_p = V \cdot U = -1/\sqrt{2} V_x + 1/\sqrt{2} V_y$$

其中 U 是沿辊子方向的单位矢量。



四、计算轮子的速度



- $V\omega = -V_x + V_y$
- 根据图所示的 a 和 b 的定义
- , 有: $V_x = V_{tx} + \omega b$

$$V_y = V_{ty} - \omega a$$

结合以上四个步骤，可以根据底盘运动状态解算出四个轮子的转速：

$$V_{w1} = V_{ty} - V_{tx} + \omega(a+b)$$

$$V_{w2} = V_{ty} + V_{tx} - \omega(a+b)$$

$$V_{w3} = V_{ty} - V_{tx} - \omega(a+b)$$

$$V_{w4} = V_{ty} + V_{tx} + \omega(a+b)$$

2.3 机械臂与控制方案

机械臂与底盘所使用的控制方法

解决PID的超调

(1) 在控制器中引入积分环节可以消除稳态误差。控制在微分控制中，控制器的输出与输入误差信号的微分（即误差的变化率）成正比关系。自动控制系统在补偿误差的调节过程中可能会出现振荡甚至失稳。其原因是由于存在较大惯性环节（如水波影响）或滞后单元（如通讯设备的延迟），它们具有抑制误差的作用，即其变化总是落后于误差的变化，而解决的办法是提前抑制误差的变化。

在控制器中仅仅引入比例环节是不够的，因为比例环节的作用仅仅是改变误差的幅值，并不能起到提前抑制误差的作用。因此还需要引入微分环节，其作用是预测误差变化的趋势。这样，具有“比例+微分”的控制器就能抑制惯性环节和滞后单元对本系统的影响，从而改善了系统在自调节过程中的动态性能。

解决震荡问题

(1) 一般来说，在整定中，观察到曲线震荡很频繁，需把比例带增大以减少震荡；当曲线最大偏差大且趋于非周期过程时，需把比例带减少

(2) 当曲线波动较大时，应增大积分时间；曲线偏离给定值后，长时间回不来，则需减小积分时间，以加快消除余差。

(3) 如果曲线震荡的厉害，需把微分作用减到最小，或暂时不加微分；曲线最大偏差大而衰减慢，需把微分时间加长而加大作用

(4) 比例带过小，积分时间过小或微分时间过大，都会产生周期性的激烈震荡。积分时间过小，震荡周期较长；比例带过小，震荡周期较短；微分时间过大，震荡周期最短

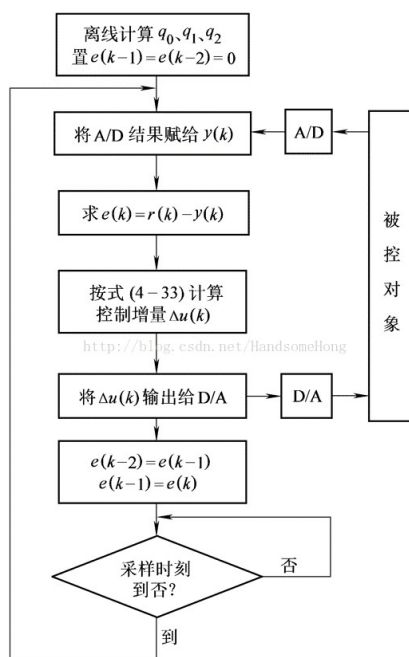
(5) 比例带过大或积分时间过长，都会使过渡过程变化缓慢。比例带过大，曲线如不规则的波浪较大的偏离给定值。积分时间过长，曲线会通过非周期的不正常途径，慢慢回复到给定值。

2.4 功能模块说明

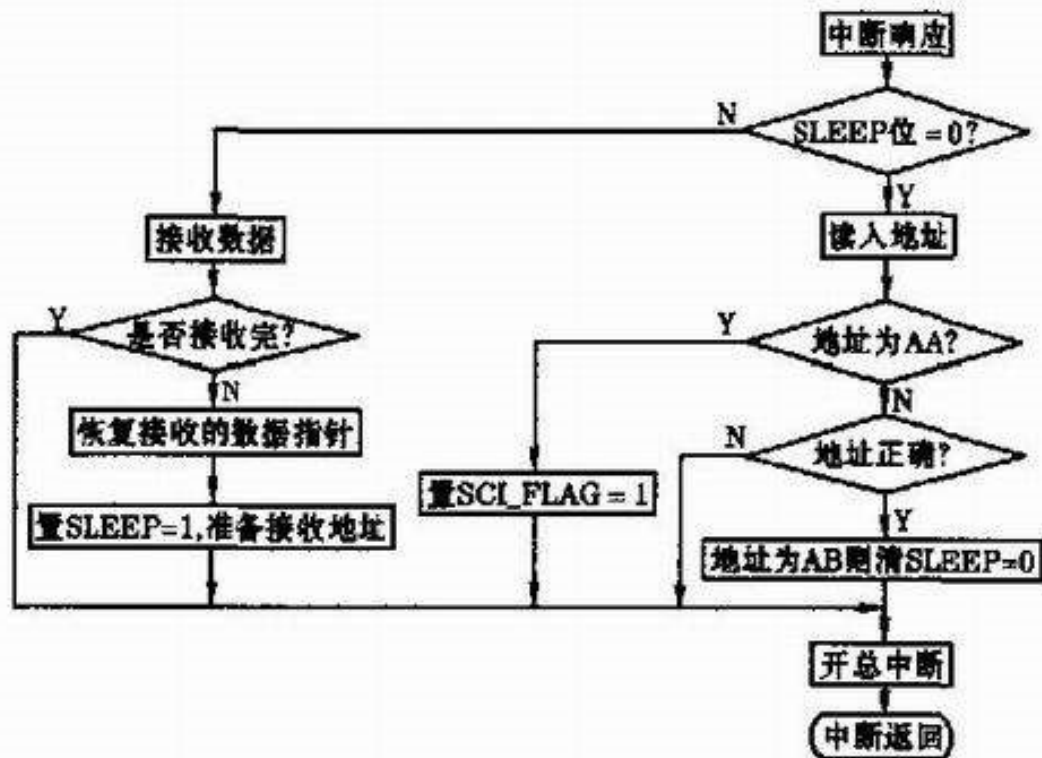
我组电控的功能模块包括通过超声波控制直线行走，通过pid控制智能小车的轮组，以及串口通信。

1. 通过控制前方的超声波来探测前方墙壁的距离，从而可以很好的规划路线，左右的超声波通过函数可以确保小车在直线上行走。

2. PID控制轮组转动



3. 通过串口通信与视觉组进行交流，接受视觉提供的信息并作出反应。



2.5 难点与不足

设计的过程中遇到的问题

1. 不同设备调试产生结果不同，两台电脑编译所结果差将近两倍。
2. 小车转向有一定误差，后改用小车平移然后倒着前行的方式来视线转向。
3. 陀螺仪的积分飘，没法确定，不好保证陀螺仪积分的准确性，经过计算大概10秒左右会飘一度。

还有那些问题没有解决

1. 编写了几种其他方案但都因为函数无法始终调用最终放弃了方案。
2. 陀螺仪飘的问题没能解决，通过滤波只能缩小一部分误差。，放弃了陀螺仪的方案。
3. 串口通信的实现在数据传输的时候出现了无法解决的问题。

第3章 算法部分

3.1 开发环境介绍

两位算法人员分别使用Windows10和Linux系统，使用基于Ubuntu16.0.4下的openCV3.1使用C++和C语言，在树莓派3B+中载入opencv3.1完成算法的内容。

3.2 整体技术方案概述

使用寻找轮廓函数冰取距离最近九个块加和的方法进行魔方识别。

用HSV颜色通道分离的方法来进行颜色识别。

使用Apriltag标记算法来识别迷宫口的标识图。

3.3 算法整体框架设计

<1>.魔方识别:用到了寻找轮廓函数，对于找到的所有正方形轮廓进行到中心的距离加和排序，对于中心距离加和排序排出距离和最小的九个块作为魔方一面的九个块。

<2>.颜色识别：对手摄像头识别到的色块进行HSV颜色通道转换及分离，对于每种色块中所有HSV通道值分析并求出平均数，从而对得到的数据进行判断是否符合颜色标准。

<3>.迷宫标识图识别：使用Apriltag标记算法，对于寻找到的所有Apriltag块做出图像长宽比以及图像大小限制，从而选出摄像头中最稳定，最准确的标识图。

3.4 算法功能模块说明

分为魔方识别模块和串口模块。

1. 魔方识别模块

提取HSV通道对每个模块求平均

```
cv::Mat colorhandle::getFrame_f(){

    cv::Mat imageout;

    if(!camera_f_stop){

        capture_f>>frame_f;

        // cv::blur(frame_f, imageout, cv::Size(2,2));

        //      cv::medianBlur(frame_f, imageout, 3);

        imageout = frame_f;

        return imageout;

    }

}

cv::Mat colorhandle::splitColor(cv::Mat templge){
```

```
char vr,vg,vb;
```

```
cv::Mat temp;
```

```
temp= cv::Mat(frame_f.size(),frame_f.type(),cv::Scalar::all(0));
```

```
for(int r=0;r<templge.rows;r++){
```

```
    cv::Vec3b *pixVal = templge.ptr<cv::Vec3b>(r);
```

```
    for(int c=0;c<templge.cols;c++){
```

```
        vr = pixVal[c][2];
```

```
        vb = pixVal[c][1];
```

```
        vg = pixVal[c][0];
```

```
        if(abs(vr-vb)>pix_val_gal || abs(vg-vb)>pix_val_gal){
```



```
temp.at<cv::Vec3b>(r,c)[2] =vr;

temp.at<cv::Vec3b>(r,c)[1] =vb;

temp.at<cv::Vec3b>(r,c)[0] =vg;

    }

}

}

return temp;

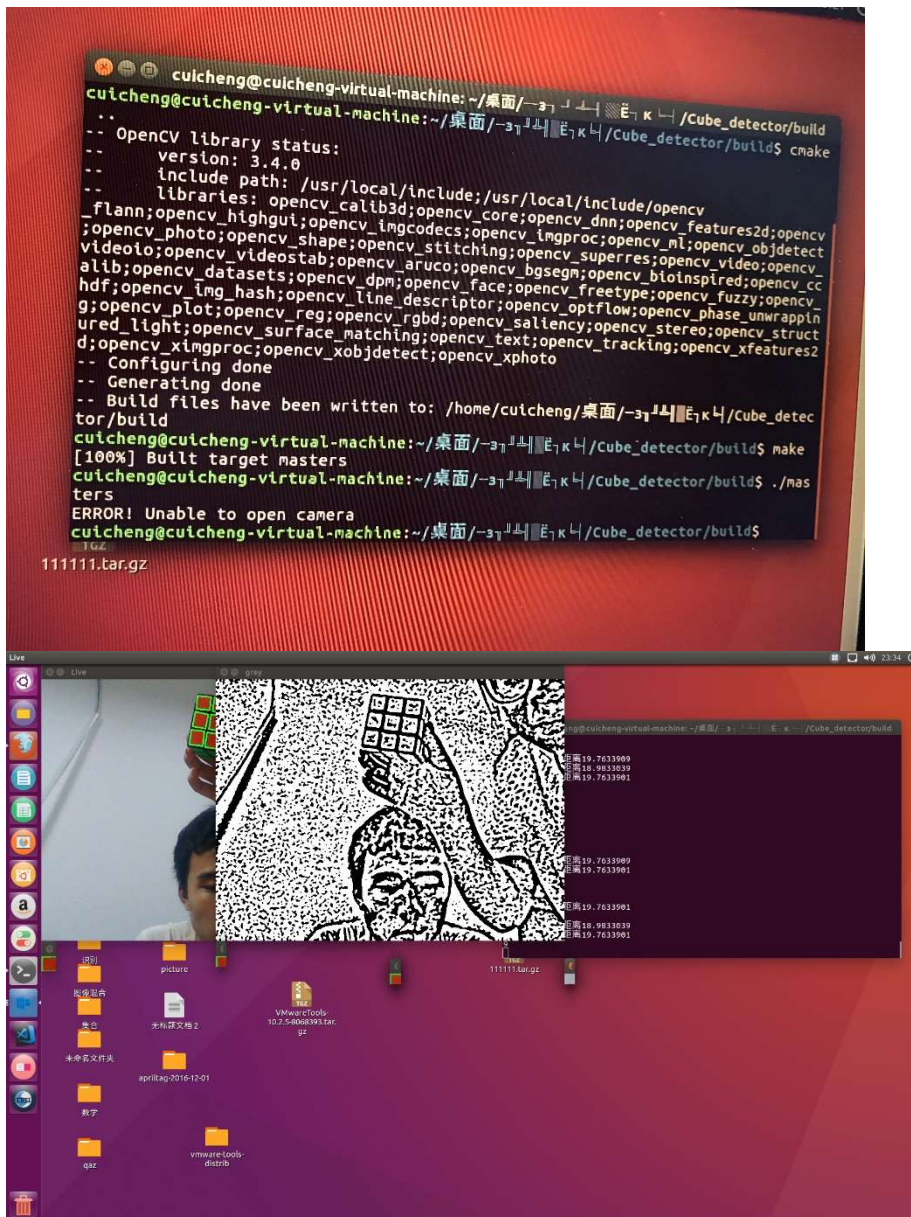
}

}
```

2. 串口通信模块

通过串口通信的方式将信息发给电控组，辅助电控组作出反应。

3.5 测试结果



3.6 可优化方案

- 1.对于HSV的通道分离可以做的更精确，对红色蓝色等暗色应该提高灵敏度。
- 2.可以尝试串口通信，通过short变量而不是float变量可能更容易实现。
- 3.对于标识识别的Apriltag的方法可以更多的学习和应用，通过视觉来识别路口标识来记忆路线，比电控纯记路线会更小几率产生误差。

4 夏令营感想、总结

这次机器人的暑期夏令营着实让我收获了不少，首先我收获了知识，前几天学长悉心的授课，我每个组相关的课都听了，对知识大体有了了解，感觉说不上学懂了很多，但至少了解了不少。也让我感觉这个假期很有意义没有被我荒废。更重要的我在这个夏令营里收获了与组员之间的友情，我们在合作分歧中逐渐产生了默契。我是一个经管院的男生，对我来说这样的和一个组的成员合作去完成一个偏向于科研的成果，对我来说是很难得的，甚至是第一次，但这次给我留下了深刻的印象，让我也对这方面产生了不少兴趣。从前的我总认为去研究一个东西是枯燥的乏味的，可能一群人在一起各自埋着头，不分白天黑夜的去做。但这次的夏令营改变了我的看法，我们一起熬夜去工作，一起去讨论一个问题，在看似枯燥的东西面前大家也有说有笑，都不会觉得累。当我们遇到了困难，我们的摄像头烧了，我的组员和我在市区里跑遍了所有的电子市场，回来紧接着又熬着通宵，让我看到大家为了一个共同的目标，不惜余力的去奋斗，让我觉得很感动。

当然这次夏令营除去结果，在过程中也有一些不足，首先经费我们考虑的不太周到，前期买摄像头之类的耗费过多，以至于最后出状况的时候产生了经费不太充足的问题。还有就是组员们在使用摄像头的时候并不是很严谨，导致出了问题。还有我自身，我在后面几天开始对于写感想和录制视频的事不太上心，连着几天都没交。还有就是技术报告我前期就应该分给组员们，但我当时考虑前期大家都在忙着学东西，我希望自己去完成技术报告，但在后面几天发现后面更加忙碌而且技术报告中很多东西我根本没法完成，所以不得已在最后几天去咨询我的组员们让他们给我说我来记录，但他们也抽不开身，我只能通过他们大概描述和我查阅资料给出来完成技术报告，所以后面很紧张。以后应当前期合理规划好，不要将任务向后拖，这样只会更加手忙脚乱。

总的来说这次的夏令营，我度过了非常充实的二十一天，我相信在以后的生活中这都是我难忘的一个回忆，这里学到的东西我也会在以后受益。这里学长们的帮助指导和组员的付出，也是我以后所学习的目标。希望还有机会能加入这样的夏令营，让我们能不断提升自我。