# 上位机

上位机部分硬件采用拆解后的搭载有Intel™ core duo®处理器的笔记本电脑。视觉图像的获取采用普通的USB摄像头, 通过特定的软件将图像进行回传处理。上位机通过USB与Arduino进行通讯。



## 通讯部分

为了能够使通讯更加的稳定并且能够应对各种突发性的状况, 我们规定了一整套的通讯协议来实现单片机和PC之间的通讯，基本的原则是发什么回什么，完成之后返回1。具体的每条指令也都有详细的规定，在这里就不一一列出。

# 策略

由于计算机性能捉急，为了能够尽可能地压缩时间，加快速度，我们尽可能地将计算机中较为浪费时间的视觉识别部分和机械运动部分进行同步，这样就能够加快全场的工作。于此同时，我们对每个区都采用了不同的抓取、识别和放置的策略以便于高效地完成各个区的抓取工作。

## 全局部分

### 购物车赛

最开始我们为了能够尽可能地减小识别在抓取过程中的时间浪费，我们采用了下面的工作流程：



但是在实际的过程中发现巡线才是浪费时间的最大部分，所以我们最终，采用流水账式的程序风格，方便于后续的抓取策略的修改，事实证明这样的方式确实有利于后续策略的调整，比如最开始我们是希望按照顺序的方式抓取，但是后来为了能够抓取到更多的物品我们按照逆序的方式进行抓取，但是这里代码的修改只是将两个部分的位置进行了调换。我们的全局代码框图如下：



#### A区策略

四个作为一组进行拍摄识别，全部拍完识别后进行抓取。

#### B区策略

四个作为一组进行拍摄识别，每拍一组进行抓取。

#### C区策略

四个作为一组进行拍摄识别，每拍一组进行抓取

#### D区策略

单个单个地识别，识别完成之后统一抓取。

### 挑战赛

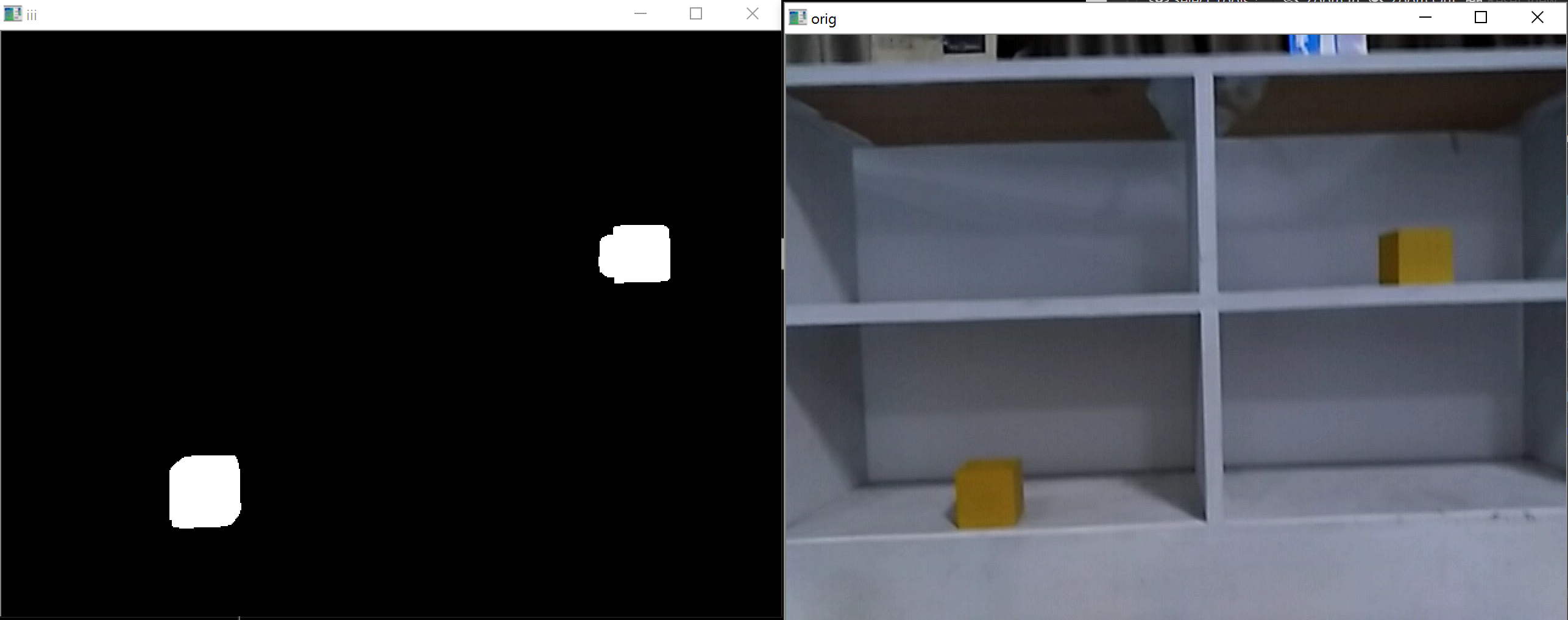
在挑战赛中我们采用的策略是可以自由选择的，下面是我们程序的框图



# 识别功能

## 颜色筛选

对于A区B区的木块，我们采用颜色筛选的方式来进行匹配。在实际操作的过程中使用了开操作和闭操作，并且只选取其中面积超过一定阈值的连通区域，这样就能够保证准确地识别出所有的木块。



## 模板匹配

模板匹配虽然在识别准确度上不及surf，但是在实践的过程中由于物品的相对位置始终是处于一个固定的值，所以通过模板匹配能够非常准确地找到对应的位置。并且模板匹配的速度要远远快于surf，所以在C区我们采用了模板匹配的方式。下面是模板匹配的效果示意图：







## SURF特征点匹配

通过surf特征点匹配，我们可以获得下面的的抽取效果，三十由于有几个物品极其相似，我们采用了二次匹配以增大识别的准确率，比如下面的纯牛奶与高钙奶：

