

**Automatic\_Coffee\_Machine**

**Software Project Summary Report**

**Organizations：** [RTCSD2017-Group03](https://github.com/RTCSD2017-Group03/Automatic-Coffee-Machine)

**Address：**[Github/Automatic\_Coffee\_Machine](https://github.com/RTCSD2017-Group03/Automatic-Coffee-Machine)

**Group\_Member:** 张照博、彭彦毓、陈志平、李佳杰、刘玉松

**Version：** 2017\_V2.0

**Time：** 2017/10/20-2017/11/10

**Teacher:** Bing Chen

目录

[1. 项目信息 4](#_Toc497605754)

[1.1 项目背景 4](#_Toc497605755)

[1.1.1 立项目标 4](#_Toc497605756)

[1.1.2 类似产品说明 4](#_Toc497605757)

[1.2 项目前景 6](#_Toc497605758)

[1.3项目定义与缩写 7](#_Toc497605759)

[2. 项目说明 8](#_Toc497605760)

[3. 项目周期 8](#_Toc497605761)

[3.1 项目要求提交阶段 8](#_Toc497605762)

[1）第一阶段 8](#_Toc497605763)

[2）第二阶段 8](#_Toc497605764)

[3）第三阶段 8](#_Toc497605765)

[4）第四阶段 8](#_Toc497605766)

[5）第五阶段 9](#_Toc497605767)

[6）第六阶段 9](#_Toc497605768)

[3.2 项目完成情况 10](#_Toc497605769)

[3.2.1 第一周 10](#_Toc497605770)

[3.2.2 第二周 12](#_Toc497605771)

[3.2.3 第三周 14](#_Toc497605772)

[4. 软件开发和管理过程 15](#_Toc497605773)

[4.1 过程说明 15](#_Toc497605774)

[4.2 过程改进建议 17](#_Toc497605775)

[4.2.1首先是上传的权限不明 17](#_Toc497605776)

[4.2.2 Push较为集中 17](#_Toc497605777)

[4.2.3管理不当 18](#_Toc497605778)

[5. 开发工具和环境 18](#_Toc497605779)

[5.1 开发工具 18](#_Toc497605780)

[5.2 开发环境 19](#_Toc497605782)

[6. 项目个人总结 20](#_Toc497605784)

[6.1 张照博 20](#_Toc497605785)

[6.2 彭彦毓 20](#_Toc497605786)

[6.3陈志平 20](#_Toc497605787)

[6.4 李佳杰 20](#_Toc497605788)

[6.5 刘玉松 20](#_Toc497605789)

# 项目信息

## 项目背景

本项目是华中科技大学2017年14级机械本科《实施控制软件设计》课程第四次作业，也是课程设计的作业内容。

### 1.1.1 立项目标

本项目的预期目标：用户在咖啡机面板上选择自己想要喝的咖啡（美式、拿铁、卡布奇诺...）并按下开始按钮后，咖啡机开始按照设定的配方制作咖啡。

### 1.1.2 类似产品说明

#### 图书馆自动咖啡机

主校区图书馆一楼大厅进门右侧就有一台自动咖啡机，支持联网付费，全自动咖啡现磨冲泡。支持多品种选择，价格从10元到30元不等。与本项目产品高度相似，但是外部无操作界面，全部由服务器通过移动端进行操作，远程控制咖啡机进行生产。



#### 德龙咖啡机



## 项目前景

全自动咖啡机是整个咖啡机行业里发展最快的。从1999年GAGGIA发布了第一台能制作Espresso（意式浓缩咖啡）的全自动咖啡机，各个不同的咖啡机厂商都在致力于研究开发，使得其功能不断的完善，已经有能加热牛奶并把它按比例配在咖啡里的高端机型面市。好的全自动咖啡机制作出来的咖啡完全可以和商用专业机相媲美，而因其能自动磨豆且相对于专业机来说价格又低很多，所以从问世以来便一直受到家庭及办公场所的青睐。

数据显示，咖啡机线上市场最近12个月总销售额（淘宝天猫）约5.4亿元，其中小型咖啡机最近12个月总销售额（淘宝天猫）约3.6千万元，小型咖啡机2017年6月销售额为283万元，占整个咖啡机市场的2%。

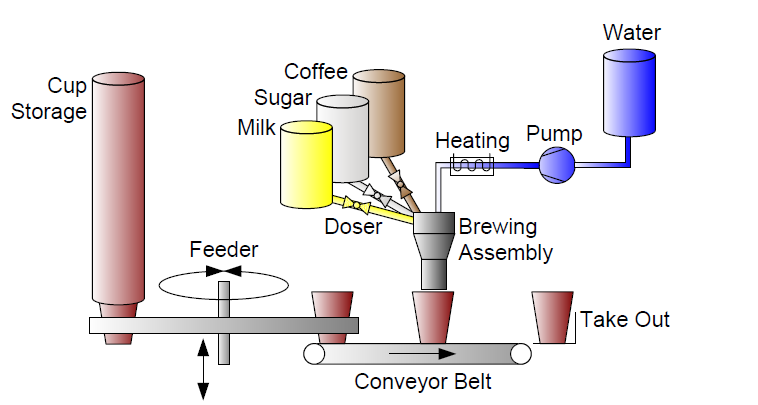


咖啡机线上市场近12个月的年同比增长12.8%，整体市场呈现上升趋势，而小型咖啡机线上市场近12个月的年同比增长率为156.2%，2017年6月销售额同比去年同月增长43.5%，可以明显看出小型咖啡机市场的快速增长趋势。

小型咖啡机细分市场中有31个竞争品牌，Top10品牌的销售额占整个小型咖啡机市场的92.83%，几乎占据整个市场，市场集中度高。其中德龙、小熊、rlrmafreda市场份额排名前三，分别占据了40%、22%、8%的市场份额，德龙、小熊在小型咖啡机市场的份额远高于其他品牌。

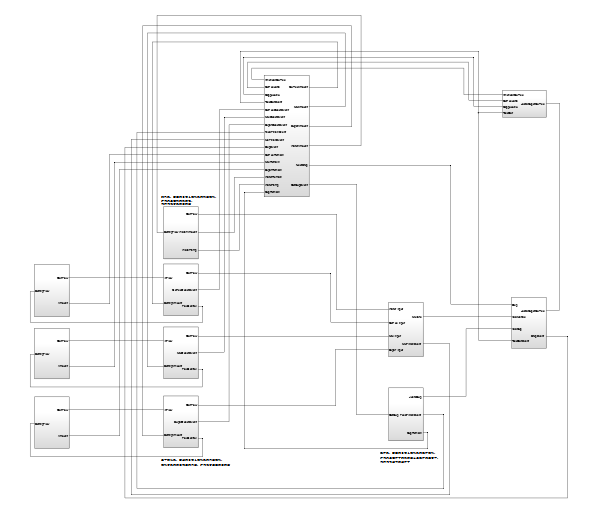
## 1.3项目定义与缩写

本项目预期实现通过控制接受人体输入，电路配合机械控制的全自动化多品种咖啡机。项目产品名为：Automatic Coffee Machine，简写为ACM。

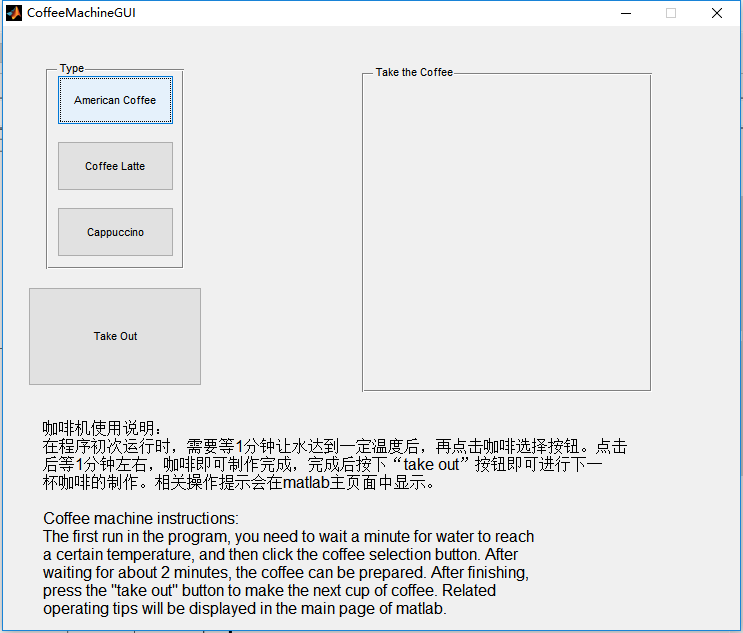


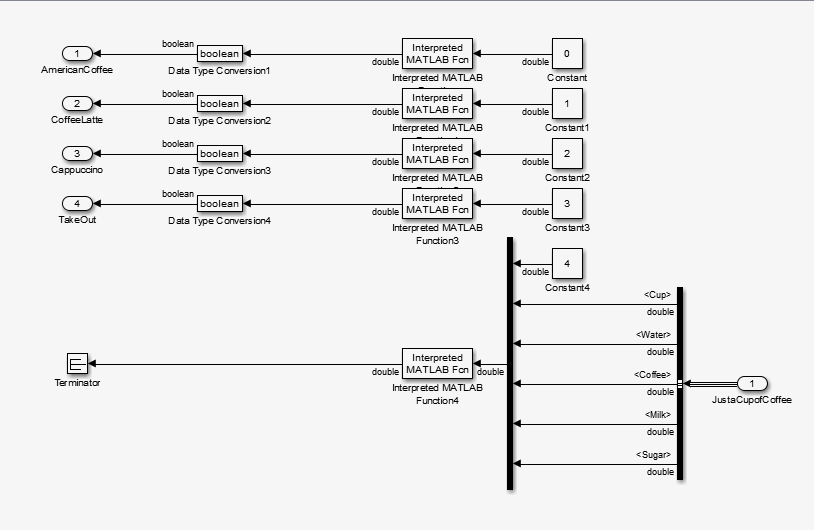
# 项目说明

项目整体布局如图所示：

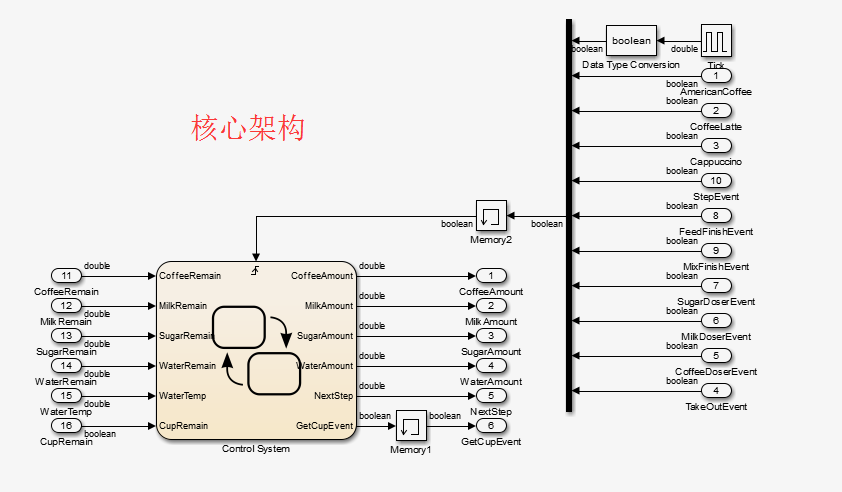


下面我们按照整个自动咖啡机的工作流程进行具体的模块讲述。

首先在可视化界面接口或者是机械按钮的人机I/O接口上，输入一个信号量。其中信号量有两种，一种是三种咖啡型号的选择，另一种是咖啡冲好之后已经被拿走的信号量输入（由传送带模块提供）。在I/O接口处的按钮模块有如下的内部控制：



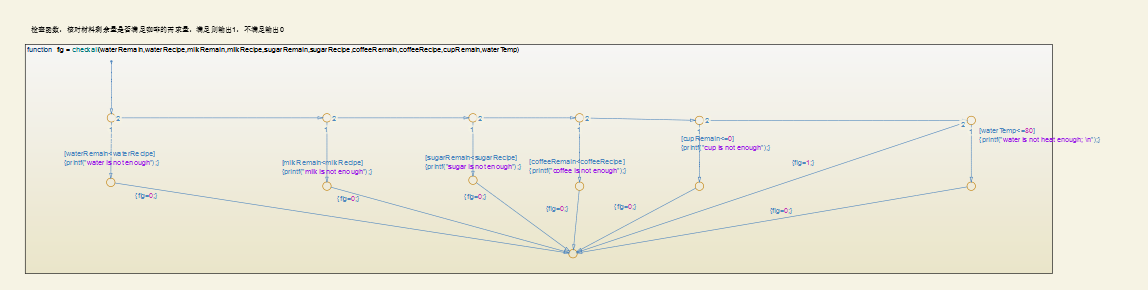
接受外界输入三种咖啡的选择处理完毕之后，I/O模块将信号量输入到核心。



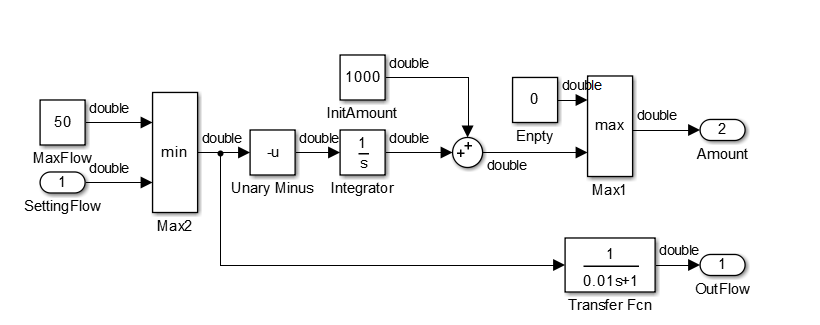
核心接受到信号量并且完成内部初始化之后。根据相对应的信号量，就开始进行配方的选择。配方的选择是通过初始化之后，以三条并联的通道，各自接受一种信号量作为状态转换新号来进行配方的获取。



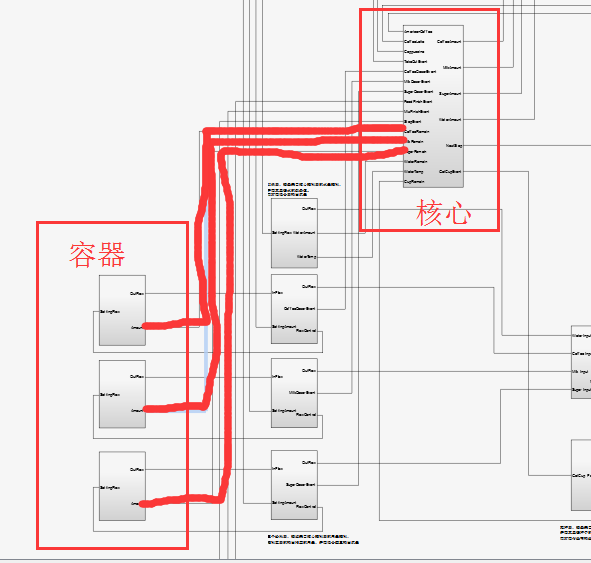
取得配方之后，我们再来进行判断我们的机器是否符合制造此配方的要求。其中机器是否符合此配方的生产要求有如下函数进行判断：



若最后函数返回值为True，那么就可以进入下一阶段—生产。而在此函数执行过程中，我们又需要外置的模块来进行响应，提供参数来对函数传入实参。

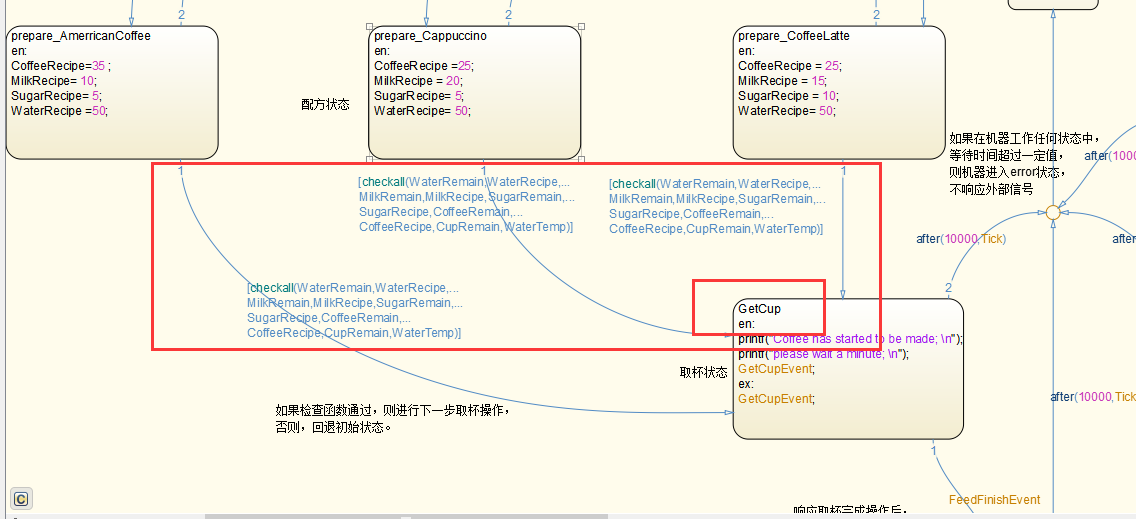


首先我们实时监控所有容器中，各种配料的剩余量。通过容器模块，返回到控制器核心。

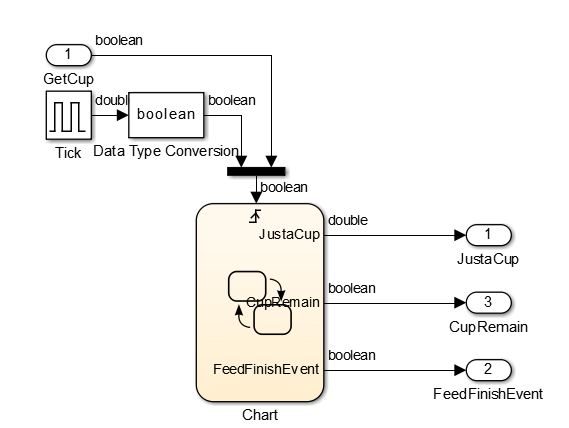


然后函数的输入就有了。即当前剩余量由外部容器模块提供，需求量由核心模块真值表提供，两相对比之下，函数就可以顺利返回bool变量！

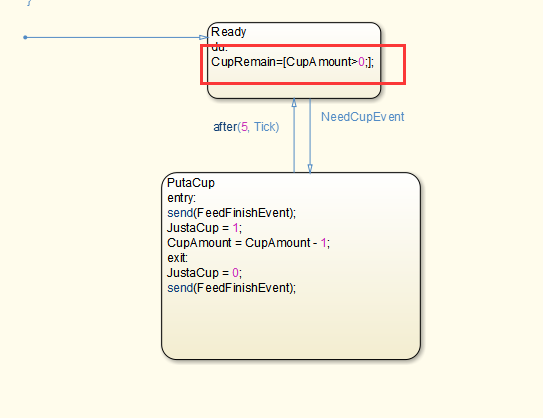
这一个函数分布在三条并联的通路上，在核心根据输入的，配方选择信号之后。如果可以通过此函数进行状态变化，那么我们就可以进入取杯状态。

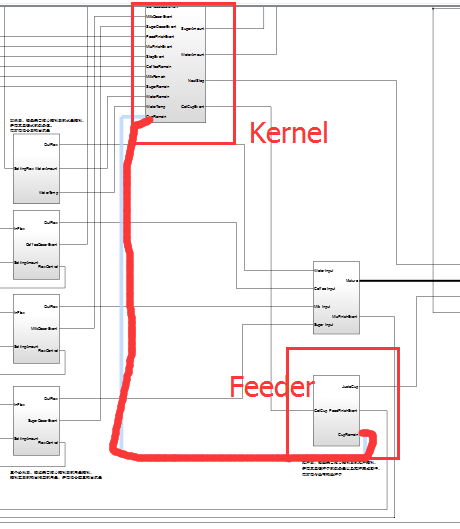


取杯的过程，我们首先需要判断，杯子容器中是否有剩余的杯子。这又需要一个外置模块Feeder来进行判断。

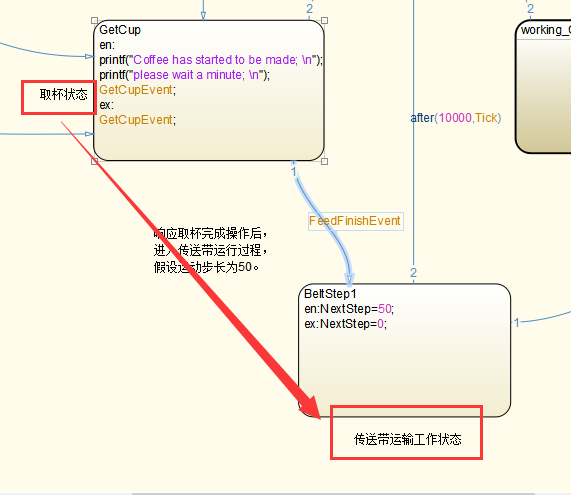


当Feeder的模块，给出杯中余量大于零的信号量的时候，我们就可以将此信号输入到核心模块。

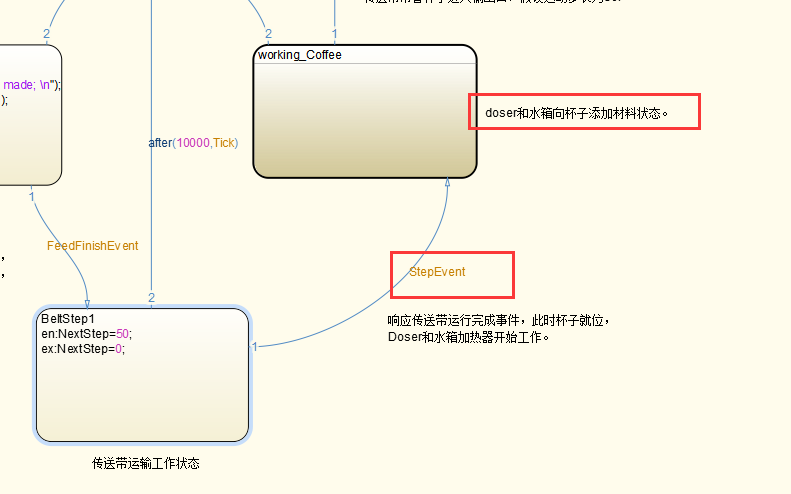
核心模块可以收到一个“取杯结束”的事件的信号量，从而从取杯状态过渡到传送带运输工作状态。



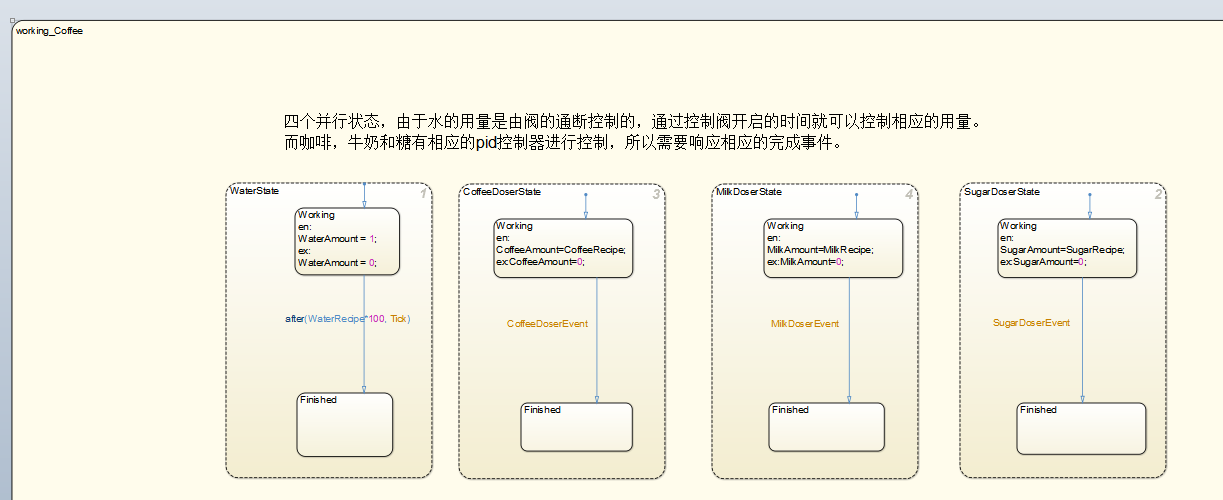
在传送带运输工作状态中，我们设置每一步的步长以及事件结束后的步长。这样我们可以精确的控制传送带，最后停止的位置。



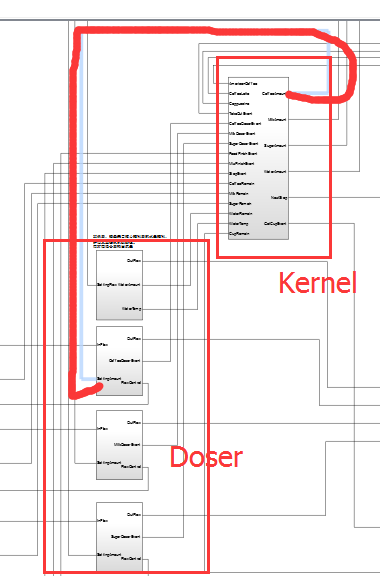
当传送带运行完成之后，此时杯子就位，那么接下来Doser和水箱加热器开始工作。



而在加热以及配比的过程中，我们设置了四个并行状态，由于水的用量是由阀的通断控制的，通过控制阀开启的时间就可以控制相应的用量，而咖啡牛奶和糖有相应的PID控制器进行控制，所以需要响应相应的完成事件。

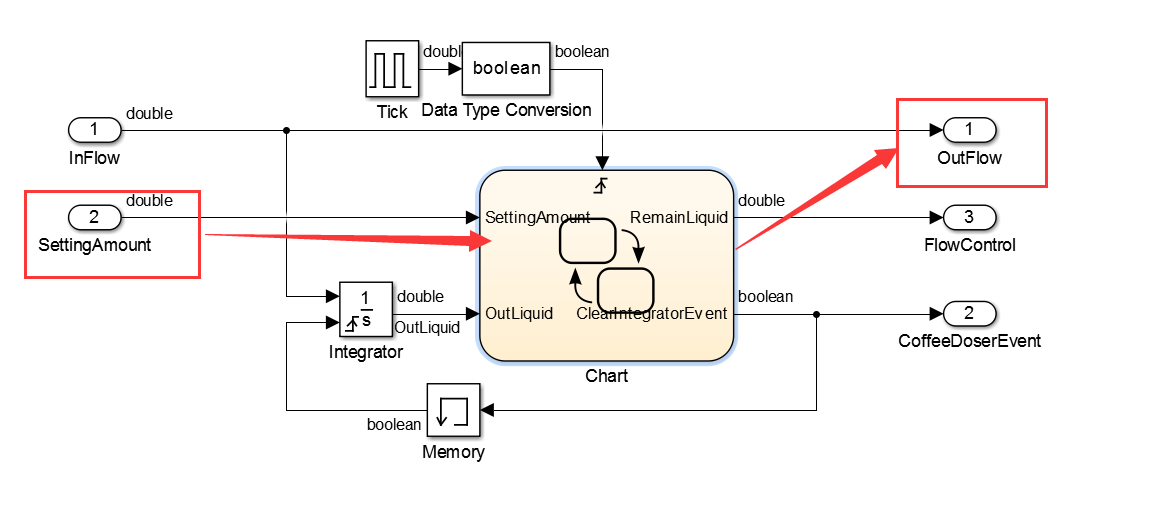


在实现各种配料混合之前，我们需要传一个信号量，给我们的Doser控制模块。

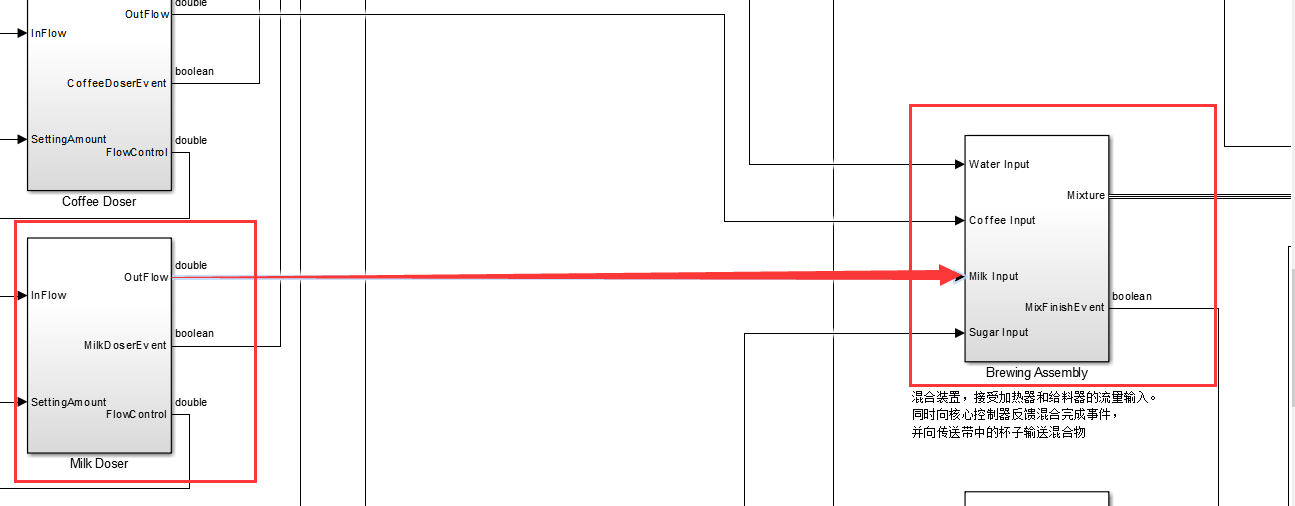


其具体的数据由配方提供，在前面已经决定，存储在本次事件中的变量中。

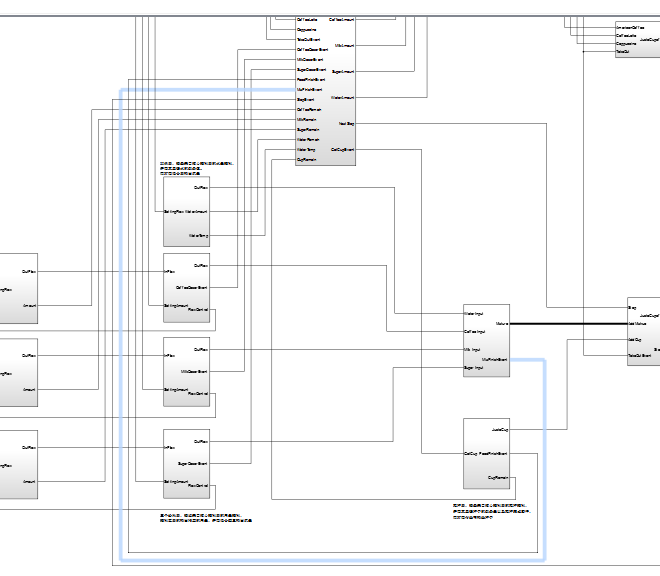
然后各个Doser模块，根据相应的数据来实现控制流出。



所有的Doser模块，全部将自己的流量输入到混合模块中。

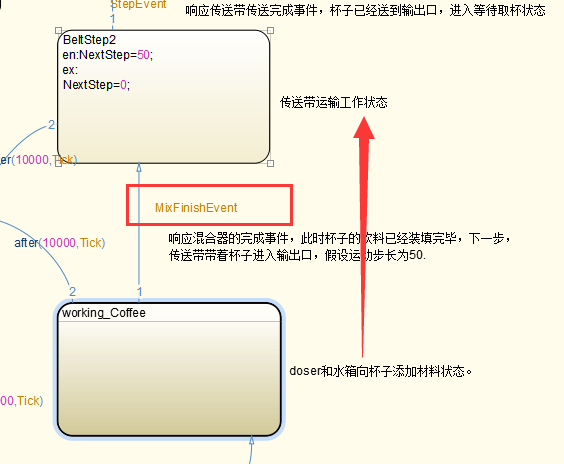


在混合模块中，经过一系列的事件控制。如果成功混合，那么在最后，就给出一个会混合完成事件的信号量返回到核心。

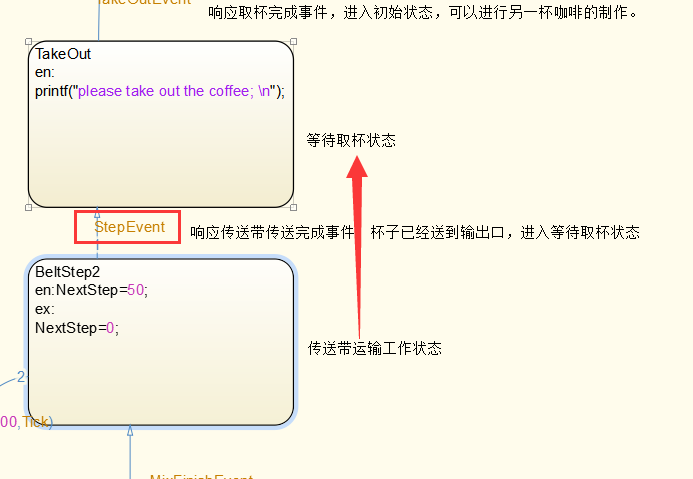


当混合完成之后，也就意味着我们，已经将成品咖啡输入到了杯中。

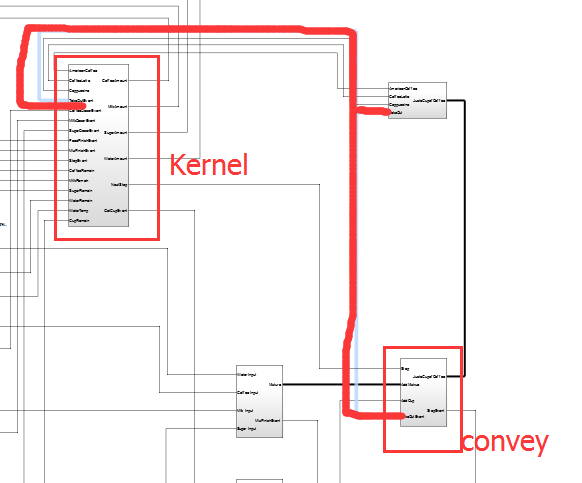
那么实际的咖啡是流入到了传送带模块上，而此时混合完成事件返回核心控制模块后意味着响应混合器的完成事件，此时杯子的饮料已经装填完毕，下一步，传送带带着杯子进入输出口，假设运动步长为50.。



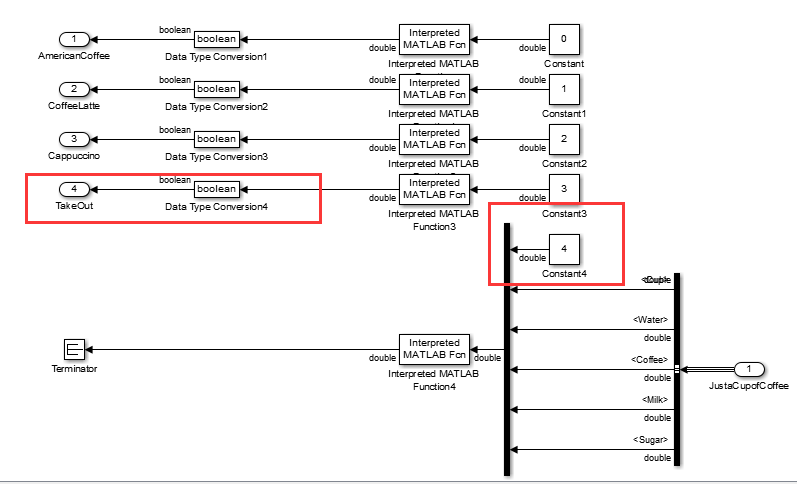
具体情节参照前面取杯后倒，混合前的过程中的传送带动作。



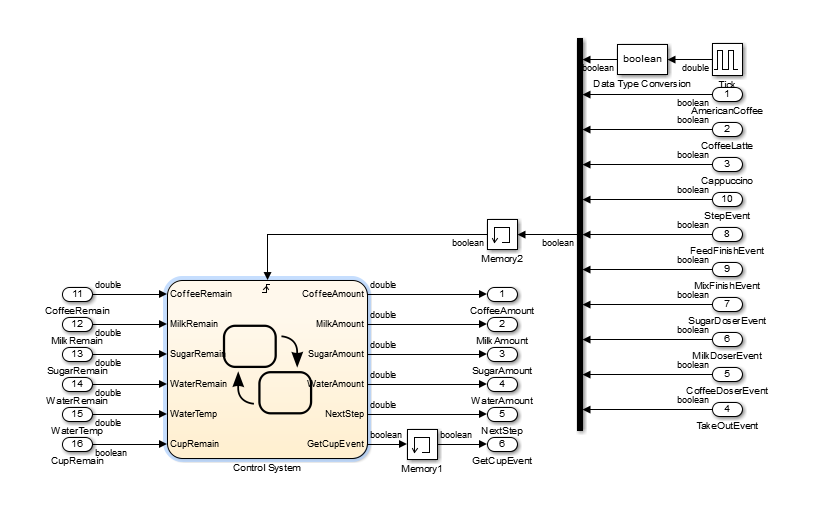
当一次步长完成之后，我们进入等待取出状态，当传感器检测到杯子被取走，就会返回一个TakeOutEvent事件信号量。此时我们又回到了初始状态，即可以进下一次事件动作。



在取杯完成的过程中，又是一次人机交互的过程。大家可以明显的观察到，从传送带模块输出的信号分流到了两个地方，一个是人机交互接口，另外一个是核心模块。回到核心模块很容易理解，那就是使核心全部回归初始化状态。那么到人机交互界面模块是什么意思呢？这里解释一下，在一次咖啡的制作过程中，我们不能让别人按下第二次咖啡，从而使得核心模块被迫的初始化，那么我们就需要屏蔽来自人机交互接口的输入信号量，直到本次咖啡制作事件全部完成为止！所以当我们发现杯子被取走之后，那么传送带模块就可以提示人机交互模块已经可以接收下一次的信号输入了。从而防止错漏。



所以与我们最开始的按钮状态合并成四个信号量一起组成了整体的人机交互接口。这样我们就完成了一次咖啡制作的流程。本项目所有目的全部达到。



# 项目周期

## 3.1 项目要求提交阶段

1）第一阶段，软件团队与咖啡机产品经理（老师和助教）沟通软件开发需求，形成软件设计规格书，清晰的定义控制软件的各项功能、硬件输入和输出接口。（本阶段交付件：软件设计规格书）

2）第二阶段，在Matlab/Simulink环境中，实现咖啡机的物理建模，同时完成控制软件的概要设计，划分成若干个软件模块，确保模块覆盖设计规格书的所有功能，清晰定义各个模块的接口。（本阶段交付件：咖啡机物理模型文件和概要设计说明书）

3）第三阶段，基于simulink/stateflow建模方法，设计各个软件模块的控制逻辑和算法，并与咖啡机物理模型进行协同仿真，实现整个咖啡制作流程。（本阶段交付件：控制软件模块stateflow模型文件）

4）第四阶段，基于模块的stateflow模型，手动编写或用自动代码生成工具，完成C代码设计工作。（本阶段交付件：模块C代码及代码自动生成报告）

5）第五阶段，基于MATLAB工具，对模块C代码进行Software in Loop测试或Processor in Loop测试，得到测试报告。（本阶段交付件：SIL/PIL测试报告）

6）第六阶段，提交项目总结和软件说明文档。（本阶段交付件：项目总结和软件说明文档）

## 3.2 项目完成情况

### 3.2.1 第一周

完成了软件设计规格说明书，并且将概要设计说明书递交；分工完毕，递交第一周个人周报。

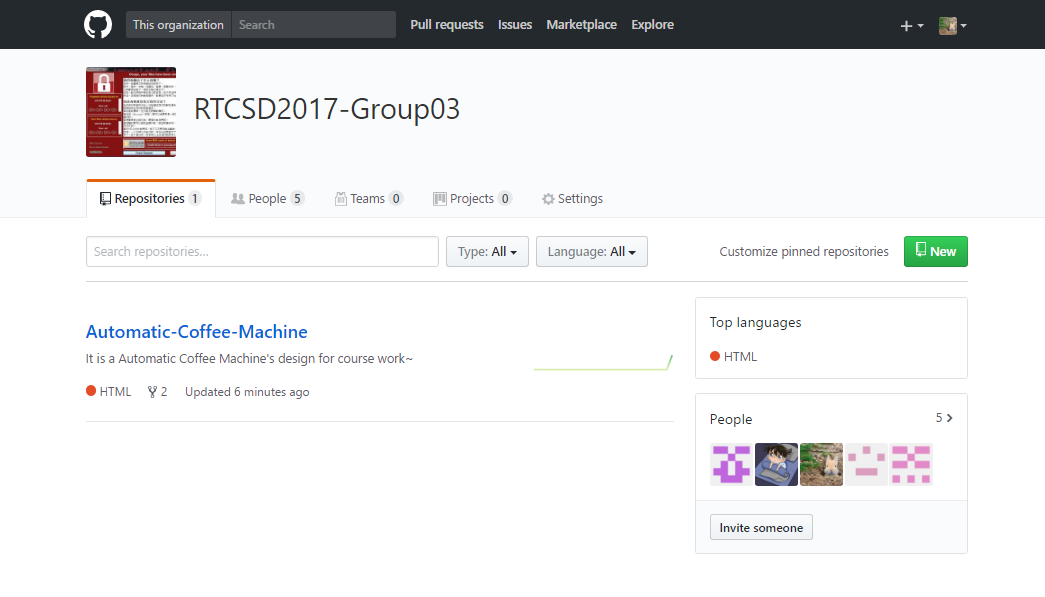
1. 软件规格说明书

[自动咖啡机项目-软件规格说明书](http://www.cnblogs.com/hustwolfzzb/p/7711409.html) 其他具体内容见Github 项目的docs文件夹下。



1. Github 团队的创建。

<https://github.com/RTCSD2017-Group03/Automatic-Coffee-Machine.git>



3、项目内容的框架搭建（最终版本）

Automati\_Coffee\_Machine

|---configs

|---scripts

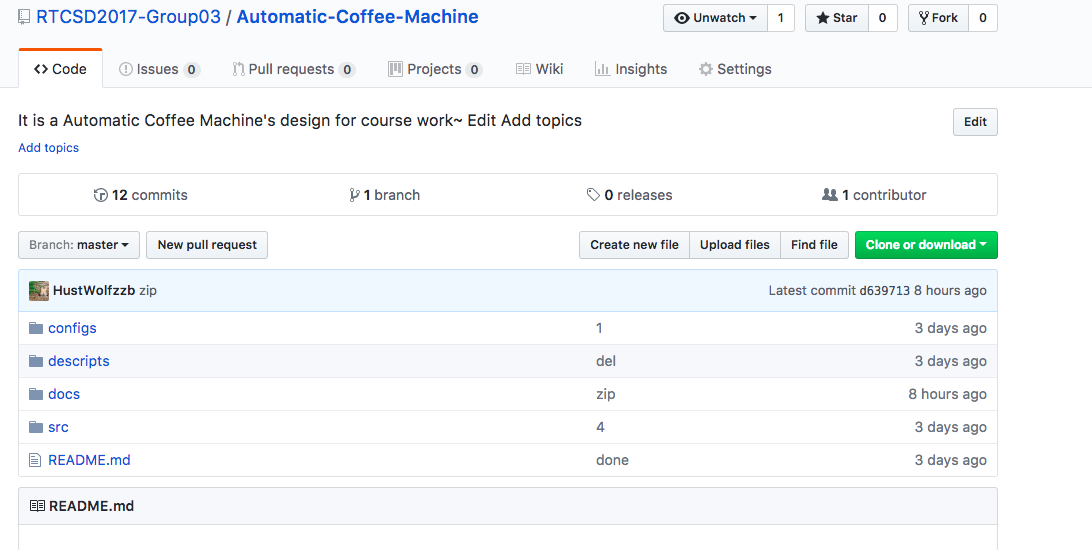
|---docs

|---srcs

|---model

|---README.md

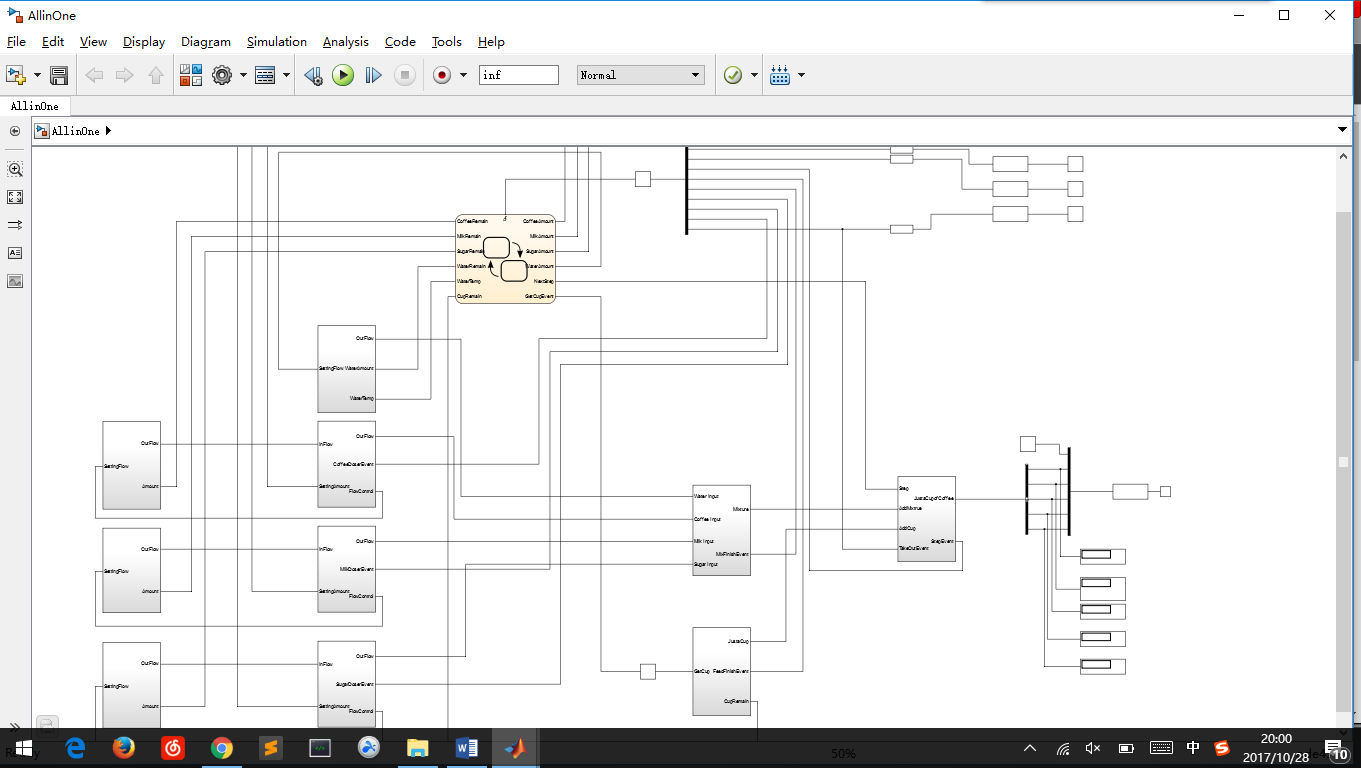
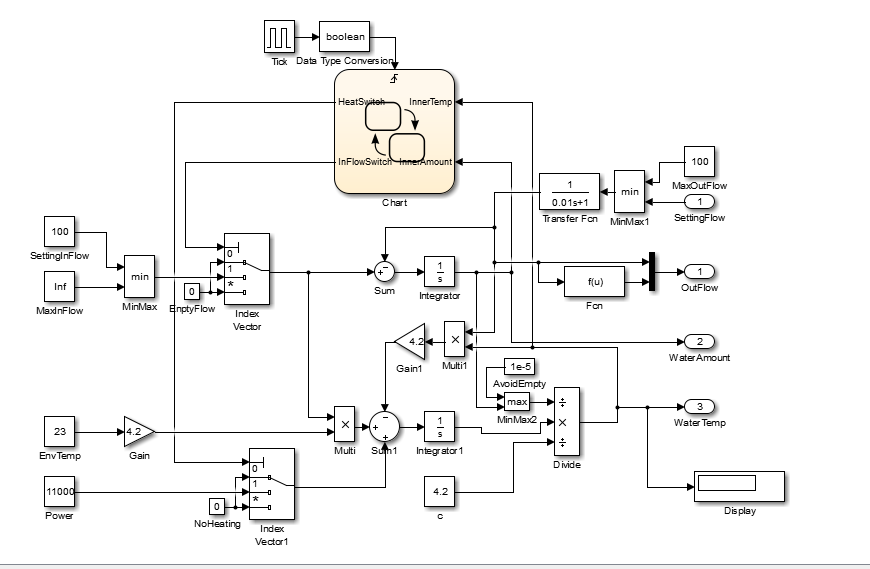
|--- [Software Project Summary Report.docx](https://github.com/RTCSD2017-Group03/Automatic-Coffee-Machine/blob/master/Software%20Project%20Summary%20Report.docx)

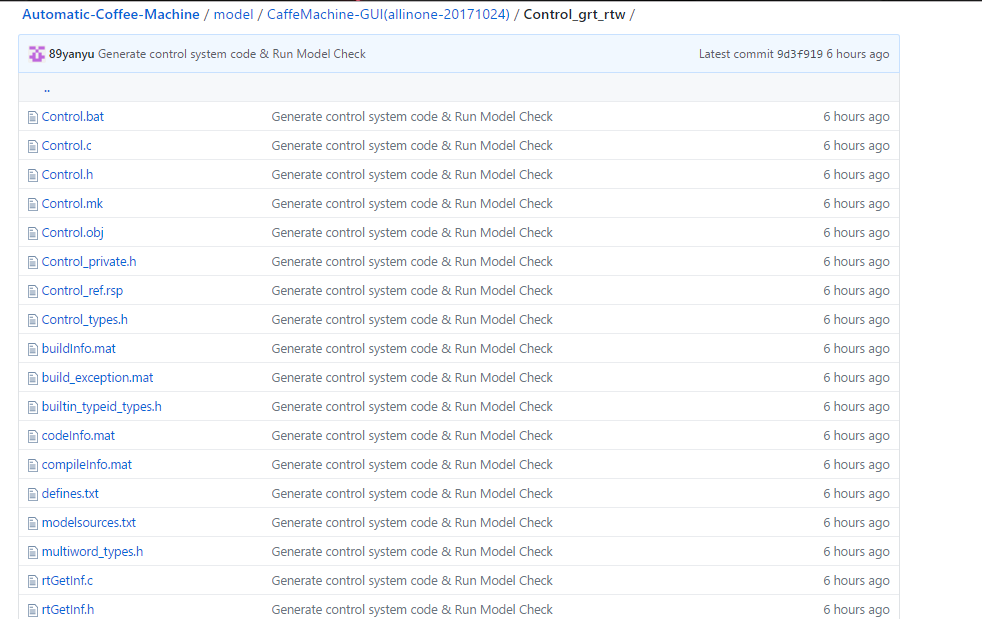


（本阶段交付件：软件设计规格书和概要设计说明书）

### 3.2.2 第二周

完成物理建模，基于simulink/stateflow建模方法，设计各个软件模块的控制逻辑和算法，并与咖啡机物理模型进行协同仿真，实现整个咖啡制作流程。基于模块的stateflow模型，手动编写或用自动代码生成工具，完成C代码设计工作。





（本周交付件：模块C代码及代码自动生成报告，控制软件模块stateflow模型文件）

### 3.2.3 第三周

基于MATLAB工具，对模块C代码进行Software in Loop测试或Processor in Loop测试，得到测试报告、提交项目总结和软件说明文档。（本阶段交付件：SIL/PIL测试报告、项目总结和软件说明文档）

# 软件开发和管理过程

## 过程说明

在软件开发的过程中，我们经历了上面所说的三个阶段。也就是根据老师给出的分阶段性任务来规划了自己三周内的三个阶段的任务。首先第一周，在懵懂未知的情况下，我们接受了任务，并且开始对任务进行各种各样的粗略任务分配。在第二周的时候，大家开始进入工作状态，还开了组会来进行具体的任务分配以及时间规划和任务精细化。



到了第三周由于各种各样的原因，我们暂时停摆了几天，不过到了周末大家开始认真工作，各展神通，为整个项目添砖加瓦，完成任务。

我们的软件是基于Github以及在Windows、Mac、Ubuntu下，各式各样的Git Client。甚至为了详述Github的使用过程，我还特意写了一篇科普文，来帮助大家提高对Git的认识以及理解。

[**《实时控制软件》Git相关操作从注册到Push全教程**](http://www.jianshu.com/p/7f7836264174)

那么说一下，我们在这个过程中，是如何实现分布式协作的。首先每个同学都建立自己的给他的账号，然后加入我创建的小组：[***RTCSD2017-Group03***](https://github.com/RTCSD2017-Group03/Automatic-Coffee-Machine)

然后在自己的电脑上安装Git的客户端。Windows下需要下载相应的软件，而在Mac和Ubuntu内就一般内置了git的程序。在每一次开始本地制作之前，我们需要先在本地初始化一个的本地仓库。然后在仓库内，与远程库形成映射关系，从而成为远程库的开发人员之一。然后把远程库拉到本地库进行合并：git pull origin master

按照各自分配的任务再开始自己的开发流程。本地开发完毕后，再上传到远端库，形成一个分支。上传过程如下：

* 第一步是用git add filename(文件名)把文件添加进去，实际上就是把文件修改添加到暂存区；
* 第二步是用git commit -m "随便写，最好是你本次的更改内容"提交更改，就是把暂存区的所有内容提交到当前分支。因为我们创建Git版本库时，Git自动为我们创建了唯一一个master分支，所以，现在，git commit就是往master分支上提交更改。
* 第三步git push origin master。登录Github去你的远程仓库看，就会发现有了最近的更新。

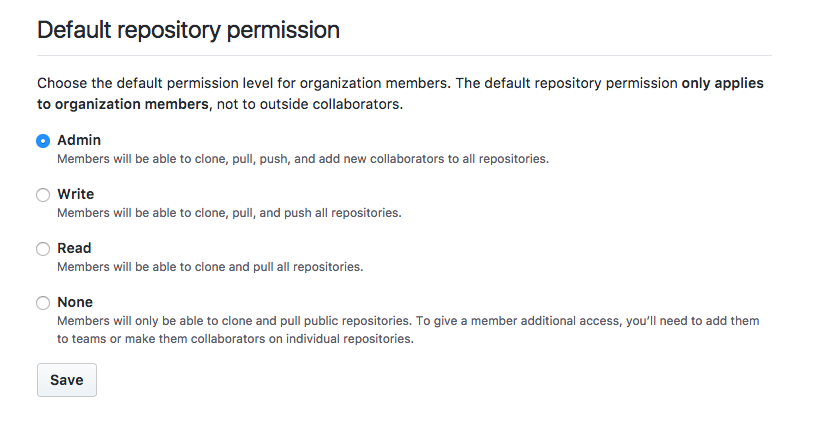
我作为远程库的创建者，在此过程中则进行版本控制与分支合并。最后进入成品准备阶段，测试完毕之后，我们把所有的Branch全部合并，并且去除掉老旧版本。从而形成只有一个master的版本。项目就进入了尾声，这就是整个软件开发的过程说明。

## 4.2 过程改进建议

在此过程中，我们发现了以下几个问题：

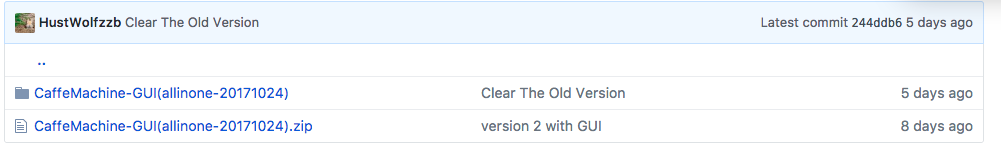
### 4.2.1首先是上传的权限不明

组内组员在本地无法用master分支直接铺市，最新的修改。只有个别同学能够成功上传，之后，改为了全员都可以上传。



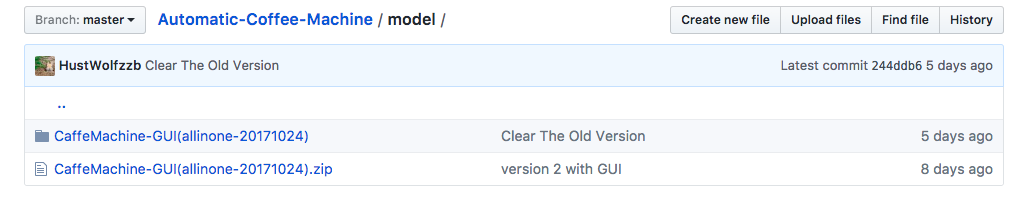
### 4.2.2 Push较为集中

因为版本控制较为繁琐，为了避免形成版本冲突，以及分支修改合并之类的繁琐事务，基本上是由我和彭彦毓两位同学进行上传更改，保持版本的更新。可以改进为，每一次版本更新按照日期分配进行，每天或者隔几天由一个同学负责，等到后面熟练了，就可以各自进行最新的push任务。



### 4.2.3管理不当

由于我的失误，导致模型（model）文件夹下相当部分文件相当错乱，各个版本重叠在一起，给助教带来了很大的麻烦，这一点应当改进。所以后期被提出后我特地建立了old version版本。



# 开发工具和环境

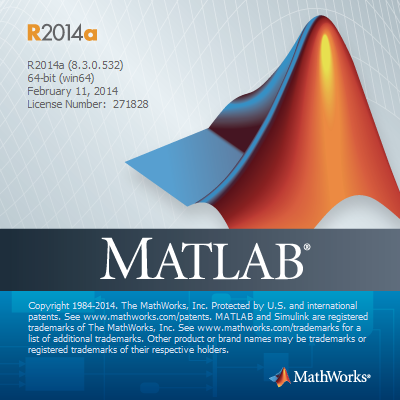
## 5.1 开发工具

**Matlab/Simulink/StateFlow**

## 

## 5.2 开发环境

**基于windows的matlab开发平台**



**基于ubuntu的C语言编译平台**

# 

# 项目个人总结

## 张照博

[《实时软件控制设计》大作业总结](http://www.jianshu.com/p/b495e12cdf4f)**—张照博**

## 6.2 彭彦毓

## 6.3陈志平

## 6.4 李佳杰

## 6.5 刘玉松