

分数: _____

华中科技大学

软件工程大型作业

作业名称: 电梯控制系统软件设计

学 生 姓 名:	<u>胡思勖</u>
学 号:	<u>U201514898</u>
专 业:	<u>计算机科学与技术</u>
同 组 成 员:	<u>无</u>

目录

I 摘要	1
II Abstract	1
III 前言	2
III.1 课题的来源、目的、意义	2
III.2 主要解决的问题	2
III.3 国内外发展状况	2
III.4 设计的指导思想	3
IV 可行性分析	4
IV.1 问题定义	4
IV.2 现有系统描述	5
IV.3 新系统可行性	6
IV.4 可行方案	6
IV.4.1 方案一	6
IV.4.2 方案二	7
IV.4.3 方案三	7
IV.5 方案选择	8
IV.6 开发计划	8
V 需求规格说明	9
V.1 引言	9
V.1.1 文档目的	9
V.1.2 范围限定	9
V.1.3 术语	9
V.1.4 综述	10
V.2 总体描述	10
V.2.1 交互模型	10
V.2.2 产品功能	13
V.2.3 用户角色	13
V.2.4 限制	14
V.3 详细需求	14
V.3.1 对象模型	14
V.3.2 功能需求	15
V.3.3 性能需求	18
V.3.4 外部接口需求	18
V.3.5 假设条件	20

VI 概要设计	21
VI.1 设计思想	21
VI.2 系统功能与目的	21
VI.3 总体结构设计	23
VI.3.1 用例执行流程	23
VI.3.2 状态转换图	31
VI.3.3 软硬件平台设计	35
VII 详细设计	36
VII.1 功能模块设计	36
VII.1.1 功能说明	39
VII.1.2 功能协作图	40
VII.1.3 功能状态图	46
VII.2 算法设计	49
VII.2.1 算法简介	49
VII.2.2 算法应用	50
VIII 界面设计	51
IX 系统测试	53
IX.1 测试目的	53
IX.2 测试方法	53
IX.2.1 单元测试	53
IX.2.2 集成测试	53
IX.3 确认测试	53
X 项目管理	57
X.1 软件规模度量	57
X.1.1 计算未调整功能点数	57
X.1.2 计算技术复杂因子	57
X.2 人员安排	58
X.3 项目进度计划	58
附录 A 参考文献	61
附录 B 体会与建议	62

I 摘要

随着现代化城市的高速发展，电梯的需求量越来越大，与此同时电梯的控制系统的设计与分析的问题越来越突出。随着建筑物的体日益增大，电梯控制系统也变得十分庞大，调度算法越来越复杂。正是由于这种复杂性，面对新的设计挑战，设计者难以充分利用已有的模块进行快速而有效的设计，而且对于原有模块的更改和重构十分困难，会造成人力资源的浪费，因此，我们急需一种全新的，高效的电梯系统调度软件来协助我们管理电梯。本文提供了一种新的面向对象的电梯调度系统的设计方案以及具体实现，为未来的电梯管理系统的设计提供了一种新的思路。

关键词： 电梯控制，调度，面向对象方法

II Abstract

With the explosive development of modern city, the demand for elevators is increasing rapidly, meanwhile, the problem shown on the analysis and designation of elevator controlling system is more and more prominent. As the buildings' volume becoming more and more huge, the elevator controlling system is getting large consequently, the algorithm is getting more complex as well. As a result of this complexity, designers found it hard to do fast and efficient design using the already-have modules when facing the problem, and the changing and refactoring on the old modules is hard to perform, which will cause a waste on manpower and resources, thus, we are urged to build a brand new and efficient elevator dispatch system to help us to manage elevators. This article offers designation and implementation of a new kind of object-oriented elevator dispatch system, which provides a new way of thinking for future elevator management systems' designation.

keywords: elevator managing system, dispatching, object-oriented

III 前言

III.1 课题的来源、目的、意义

在现代的城市生活中，我们不可避免的要使用到电梯，在所要到达的楼层较高的时候更是如此。但是电梯控制系统软件的表现不总是尽如人意的。比如在有多部电梯的楼层中，经常会碰到需要等待的时间非常长，电梯空间不足但是电梯调度系统无法对此及时做出反应，电梯调度算法不合理导致效率低下，电梯损坏但是未能得到及时维修，被困在电梯中难以获得救援等状况。在问题严重的时候，电梯出现的故障可能导致生命危险。同时，由于建筑物体积的增大，一个建筑可能有多部电梯用以分散人群压力，如哈里发塔，更是有惊人的 57 部电梯¹。此时电梯管理系统将变得更为复杂。因此，一个高效，安全的电梯控制系是十分必要的，本文所描述的电梯管理系统软件也正是由于这些因素的存在才产生的。

本文旨在描述一种有效、稳定、安全、具有通用性的电梯控制系统软件的设计与实现，用于管理现代建筑物中的电梯的运作方式，并弥补传统的电梯管理系统的缺陷，同时增加多种特性和功能，能够有效的在各种情况下正确的管理和引导电梯的运作，同时，为今后的电梯设计提供一个可以借鉴的、易于修改的良好解决方案。

III.2 主要解决的问题

- 在不需要对软件进行较大的改动的情况下，对任意楼层数任意电梯数的统一管理。
- 尽可能高效率的，根据实际情况（人员情况和电梯状况）调度电梯，减少乘客的平均等待时间。
- 减少安全隐患，在出现安全问题的时候能够及时通知相关人员处理。
- 在发生紧急情况的时候（如火灾等）能够做出紧急应对方案。

III.3 国内外发展状况

在国内，对于电梯系统的硬件方面的研究已经比较成熟，尤其是给予嵌入式系统的电梯硬件设计。以基于 CAN 总线的电梯控制系统为例，CAN 总线种成本合理、可靠性高、功能完善、技术先进，被广泛应用于自动化控制的各个领域。CAN 总线技术特点还包括成本低；总线利用率高；数据传输距离远（长达 10 km）；数据传输速率高（高达 1 Mbps）；具有自动退出总线功能；完善的错误检查和处理机制，利用 CAN 总线能够方便的实现分布式控制 [5]。与之类似的还有基于 μ -COS 的电梯控制系统。总而言之，我国在嵌入式电梯系统设计方面的发展已经比较完善。

对于电梯调度系统软件的算法方面，被研究的较为深入的部分为 PID 控制、多目标模糊控制算法等。以比较有代表性的《电梯群控系统多目标模糊控制算法的设计》所提出的多目标模糊控制算法为例，在此控制算法中，首先根据交通流特征描述量（单位时间上行乘客数量与下行乘客数量），进行交通流模式识别与模式切换，在 1 天内的客流量按照 4 种不同交通流模式进行有规律的切换 [3]。而在《基于匈牙利算法的群控电梯调度的实现》中提出的算法同时考虑到了候梯时间、乘梯时间、能耗、梯内拥挤度 [2]。虽然能够较为有效的解决效率问

¹Burj Khalifa. (2017). En.wikipedia.org. Retrieved 24 March 2017, from https://en.wikipedia.org/wiki/Burj_Khalifa

题，但这些算法均没有从更高以一级的层面推算最佳解决方案，即这些算法未考虑到人流量随时间分布在随着日期的变化是不一样的。

相较而言，在 *Improving elevator performance using reinforcement learning* 中提到的对电梯组的每一台电梯使用深度学习算法，同时整个电梯组接受一个总的深度学习的算法控制，能够有效的根据一天内不同的时间段以及不同的日期预测人流量并调整电梯的行为 [1]。在《基于遗传算法模糊神经网络的电梯群控调度》中提到的给予遗传算法的控制具有自学习功能，并可以自动适应人流的变化，进行平滑的调整 [4]。这一类的算法自动化和智能化程度更高，更能够做出经济高效的决策。在本文所提出的电梯系统中正是借鉴了这种方法使电梯的管理更为简单。

III.4 设计的指导思想

本电梯控制系统软件的设计遵循以下指导思想

1. 软件应具有可用性，能够正常的运作。
2. 软件应具有通用性，能应用于绝大部分的电梯系统中。
3. 软件应具有健壮性，能够自动处理大部分的异常，并在出现致命性错误的时候应有健全的保护机制。
4. 软件应结构清晰，易于维护。在出现错误的时候能够快速定位错误位置并进行修复。
5. 软件应具有高效性，对于不同的状况均能够分析最有效的解决方案进行解决。

IV 可行性分析

IV.1 问题定义

基于目前大部分楼房存在的电梯控制系统软件存在的缺陷，设计实现一种新的电梯控制系统软件，其研发成本、安装成本、在房屋使用年限之内的维护成本之和应该小于直接使用现有系统的安装成本和建筑使用年限之内的维护成本之和，并能够提高用户体验，减少事故。假定新系统的安装成本和旧系统的安装成本相同，主要的目标就是减少电梯的故障维修成本和事故成本，以及改善调度的合理性。

在安全方面，中国整体电梯故障率尚不明确。但根据 *Consumerwatch.com* 提供的数据，在美国，每年约有 27 人因电梯致死，10,200 人因电梯受伤，而美国总体大约有 90 万部电梯¹，以此为依据，计算出平均每台电梯每年造成的事故伤害到约 1/9 人，而维护和赔偿所需的费用不会少于 3000 元/人/年。根据《国有土地使用权出让和转让暂行条例》第 12 条的规定，居住用土地使用权出让的最高年限为 70 年²。计算出每台电梯在建筑使用期限内的维护花费平均至少为 23000 元，加上无故障例行检修成本约 6000 元/年，则总成本为 29000 元。而新型电梯的开发不可能使事故发生率低于 0，因此这一方面对新系统的预算贡献为 29000 元。

在调度合理性方面，若能够减少 20% 的运行距离，接可以节省 20% 的运行费用，以及将问题的发生率降低到原来的 80%，我国目前平均每部电梯每天耗电 40 千瓦时³，以每度电 0.6 元计算，一年电梯所需消耗的电费为 8760 元，调度和理性对于新系统预算的贡献为 122640 元。

考虑到年限较长，需要考虑通货膨胀，根据目前中国的通货膨胀率 2.3%⁴，计算出实际的预算为 $(122640 + 29000) \times (\frac{1}{1.023})^{70} \approx 31000$ 元/部电梯。而实际系统开发完成后不可能仅用于一部电梯，因此实际的项目规模应该更大

开发的具体问题、规模和初步目标见IV-1-1。

表 IV-1-1: 关于电梯控制系统软件的规模和年目标

项目名称	细节阐述
问题	目前存在的电梯控制系统软件不够高效、稳定、智能化
项目目标	研究开发一种易于管理的更优良的电梯控制系统软件的可能性
项目规模	开发总成本平均到打算安装的每部电梯不应该超过 31000 元
初步设想	使用 C++ 或者 Python 设计一个管理系统软件，最好可运行于嵌入式设备上
可行性研究	进行大约历时一周的可行性研究，研究成本平均到打算安装的每部电梯不超过 2000 元

¹Elevator Accidents & Deaths | ConsumerWatch.com. (2017). Consumer Watch. Retrieved 25 March 2017, from <http://consumerwatch.com/workplace-public-safety/elevators/>

²中华人民共和国城镇国有土地使用权出让和转让暂行条例. (2017). Mlr.gov.cn. Retrieved 25 March 2017, from http://www.mlr.gov.cn/zwgk/flfg/tdglflfg/200601/t20060119_642175.htm

³電梯更需要節能減排 _ 經濟學人 - 前瞻網. (2017). Big5.qianzhan.com. Retrieved 29 March 2017, from <http://big5.qianzhan.com/analyst/detail/220/20111214-0bd833976f460ab8.html>

⁴各国通货膨胀率列表. (2017). Zh.wikipedia.org. Retrieved 29 March 2017, from <https://zh.wikipedia.org/wiki/各国通货膨胀率列表>

IV.2 现有系统描述

大部分电梯管理系统软件，其接受的参数均为用户输入数据以及系统现在的状态，经过控制系统软件处理后，输出数据为对于电梯的调度以及显示输出，用图来表示，即为图4.2.1。

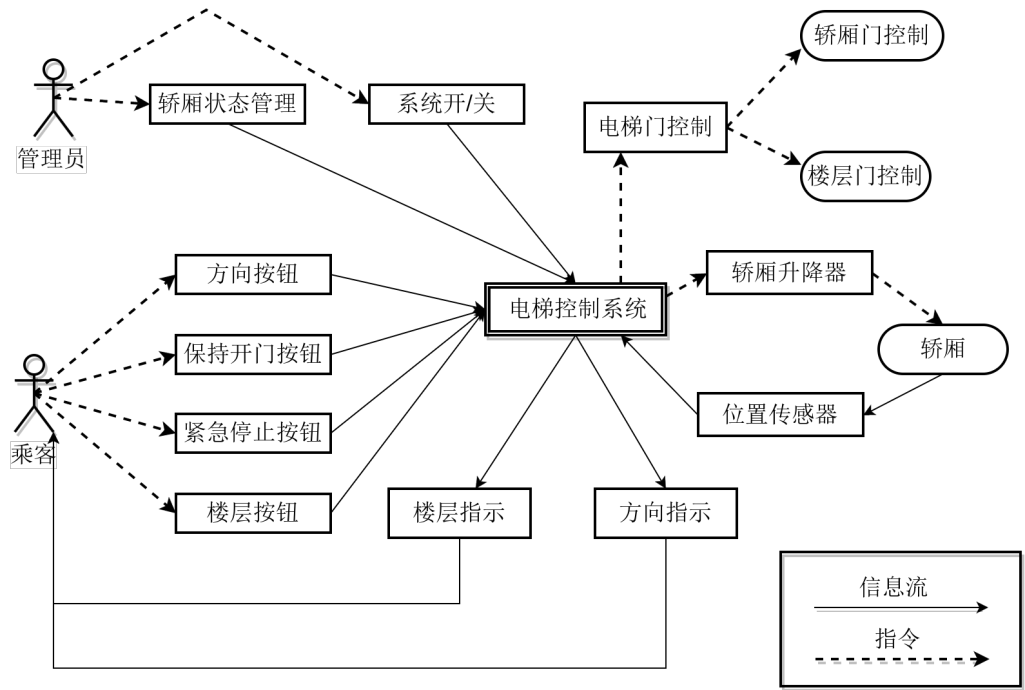
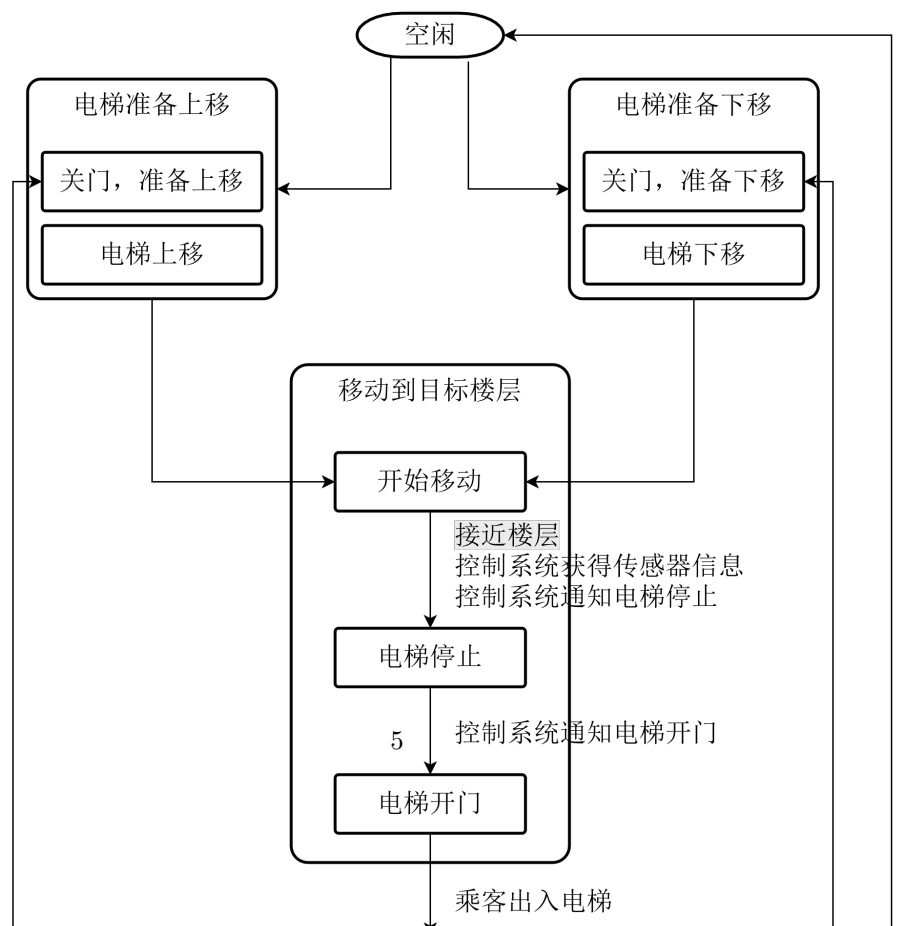


图 4.2.1: 电梯控制系统黑盒流程

查阅资料后，对电梯管理系统软件进行研究，并对图4.2.1中的黑盒进行展开，以了解目前电梯管理系统软件的具体工作细节，展开后黑盒黑内部的内容如图4.2.2所示，其为电梯控制系统软件的基本工作框架。



如4.2.2展示的为单个电梯系统的基本工作流程。可以看出，现有的单个电梯系统只是简单响应用户的输入，并对其做出反应。而现存的多电梯系统多为多个单电梯系统的组合，即每个电梯都只是遵循此流程进行响应。即使有多个电梯共用一个呼叫面板，面板也只是根据电梯的空闲状况将呼叫信号分给其中一个电梯，本质上还是多个这样的系统的简单组合，不能做到足够的智能化控制。

IV.3 新系统可行性

现在我们对于用户的问题以及现有的系统已经有了比较深入的了解，下面从多个方面进行可行性分析：

- 技术可行性：到目前为止，对于开发此电梯控制系统软件所需的知识所在的各个领域的研究已经比较成熟，无论是控制理论，调度算法，机器学习，还是编程语言，软件所运行的平台，嵌入式技术等，都有较为成熟的理论成果，因此本电梯控制系统软件在技术上是完全可行的。
- 操作可行性：短时间大面积更换新型电梯控制系统并不是十分可行：若新系统没有得到足够的实践检验而出现故障，则大面积维修的代价高昂，并且不容易取得大面积更换新系统的经济支持。但是在小范围内更换新的电梯管理系统是可行的：它所需的资金较少，维护成本低，且能够充分的对其进行各种测试，待新的电梯系统测试基本成熟后逐步扩大更换范围是可行的。
- 经济上，根据章节IV.1对于项目可行性以及项目规模的初步研究，只要平均每部电梯的研究成本和安装、维护成本不超过 30000 元即是经济可行的。而根据现有的材料和人力资源的市场价格，初步预计开发成本平均到每部电梯不会超过 30000 元，因此在经济上本系统是可行的。

IV.4 可行方案

经过初步的可行性研究后，得出以下三种可行的方案待选择，其成本依次升高，每一种方案都有自己的优缺点，最终经过斟酌后将在这三种方案中选一种最为合理，最符合现在实际情况的解决方案。

IV.4.1 方案一

直接在现有的电梯系统上进行更改，更换电梯的控制系统使之接近新设计的系统。

优点：

- 减少了购买原材料的成本，只有软件开发、替换、维护的成本。
- 对于现有的系统改动最少，更换时间较短。
- 若新系统出现严重问题，可以迅速回滚为原来的系统。

缺点：

- 不能够保证兼容性。现存的电梯控制系统的结构不尽相同，与现有系统的差异不同，实现起来十分复杂。
- 维护过程复杂。由于运行的软硬件平台各不相同，对于不同的系统不能复用同一个维护方案，难以实现自动维护。
- 实现后系统稳定性不能保证。由于现有的硬件系统不一定能够胜任新的软件的计算速度以及响应速度，因此不能够保证更换后系统运行的稳定性。

IV.4.2 方案二

仅更换控制系统所需的硬件，并将新的软件应用其上，保留原来的机械性部件和终端部件（如升降机、显示面板、按钮等）。

优点：

- 由于控制系统运行平台的统一，控制系统的维护较为方便，可以采用统一的维护方案，甚至实现自动维护。
- 成本适中，仅在方案一的成本基础上增加了控制系统硬件的安装、布线成本。
- 普适性较强，只需考虑终端型硬件的接口不同，控制部分软件将与之隔离，可使用同样的软件。

缺点：

- 安装步骤较为繁琐，需要考虑不同类型的电梯的布线以及硬件接口的问题。
- 难以回滚到原来的系统，需要对原来的硬件进行重新安装，回滚成本较高。
- 硬件可扩展性不强，原来的电梯设计可能没有留出对应位置来安装新的终端型硬件（如指示面板等）

IV.4.3 方案三

完全更换所有的硬件（包括电梯轿厢、升降系统、控制系统、终端型硬件等），并使用新设计的软件和硬件。优点：

- 所有的电梯系统均完全相同，十分易于维护。
- 能够做到对电梯系统的完全控制，可扩展性极强。
- 安装过程统一，易于管理。

缺点：

- 由于需要对整个电梯进行更换，施工时间长，对用户造成的影响较大。
- 成本偏高，不仅有安装成本和研发成本，还有对于原来的电梯系统的拆卸和处理成本，回收成本需要的时间较长。

IV.5 方案选择

以上三个方案均可以在预定的项目预算中完成，且各有利弊。经过与客户的交流后决定使用第二种解决方案，即只更换控制系统所需要的硬件。第一种方案虽然成本较低，但安装完成以后的不可预计性较高，后期的维护和修补费用可能增大，且不利于管理，可行性较低。第三种方案从长远来看是一个比较好的选择，它实现了统一的安装、管理、维护等过程，但其成本较高，需要较长的时间收回成本，而在这段时间内可能会对于此电梯系统做出更新，在收益上来看不划算。第二种方案虽然不能做到完全的控制，但是它最符合实际情况，既能较快的收回成本，又能较快的接近最终的需求，并且保留了一定的可扩展性。因此最终选择第二种方案。第二种方案的成本/效益分析如表IV-5-2所示。注意，虽然本方案选择包含了软件和硬件的费用，但是本项目只负责软件的开发，虽然开发会尽量覆盖到尽可能多的硬件平台，并且测试会在部分硬件平台上进行，但本项目不负责对于硬件设计以及硬件接口的研究。

表 IV-5-2: 方案二的成本/效益分析

项目	明细	金额
开发成本	人力（2 人月，8000 元/人月）	1.2 万元
	硬件设备	1.0 万元
	开发用软件版权费	0.2 元
安装和运行费用	安装费用	0.1 万元/部
	维护费用	0.1 万元/部
每年节省的费用	电费	0.2 万元/部
	误工费	0.2 万元/部
投资回收期（以 4 部电梯计算）		1.5 年
回收投资后年均收入		0.4 万元/部

IV.6 开发计划

根据选择的最佳方案，我们制定了如表IV-6-3所示的开发计划：

表 IV-6-3: 实现方案二的粗略开发计划

阶段	所需时间/月
可行性研究	0.5
需求分析	0.7
概要设计	0.5
详细设计	1.0
实现	2.0
总计	4.7

这一开发计划并非最终的开发计划，但总体开发进度不应与此开发计划有较大的差别。

V 需求规格说明

V.1 引言

V.1.1 文档目的

本文档的目的是通过以面向对象的方法，对于将要开发的电梯控制系统软件提供一个具有完整性和一致性的详细说明。本说明将以文字的方式来解释各个概念，使用图表的方式来阐释复杂的软件组成与交互，并使用表格来提供相关信息。

本文档的受众是所有与电梯控制系统软件开发有关的项目的持股人，这包括但不限于软件开发人员、项目管理者、质量监测人和消费者。

V.1.2 范围限定

本电梯控制系统软件（以下简称控制软件）需要确保对于电梯系统中所有其他组成部分的安全且高效的管理。软件将运行于控制器上，而控制软件的主要职责为接受来自于电梯其他部分硬件的输入，运行软件并输出对应的结果到其他部分的硬件。控制软件的两个主要目标为记录用户输入队列和调度多个电梯轿厢使之做出对应的反应。控制软件的另一个重要目标为确保使用电梯系统的用户的安全。以上的职责和目标需要通过与其他控制、监视、显示等部件的配合来完成。

处理恶意的高频用户输入不是控制软件的职责。控制软件将对正常情况下的低密度和高密度用户输入做出对应的处理，但对于恶意的、超过预定范围的高密度输入，本电梯控制系统将采用用于处理正常情况下高密度输入的算法来处理。

本规格需求说明仅对控制软件做出要求，而并不包含对于控制器（控制硬件）以及其他组成部分做出要求。它们之所以被提到是因为控制软件需要基于控制硬件运行，并与电梯的其他组成部分有交互与合作的成分，但本规格需求说明并不包含对于控制硬件与电梯的其他组成部分的要求。

V.1.3 术语

- 位置标志：** 位置标志是在电梯井壁上的金属或非金属标志，它将被电梯轿厢外部的传感器所探测。每个电梯井内都有一列位置标志均匀分布在整個电梯井内，传感器可以通过探测他们并将信号发送给处理器，处理器经过计算后便可以知道轿厢的位置。
- 电梯门：** 电梯门指的是电梯轿厢内部的门和电梯所停靠楼层的门的组合。由于电梯总是同时开关电梯轿厢门和所停靠楼层的门，因此不需要区分“内部门”和“外部门”，统称为“电梯门”即可。

- 显示面板：** 显示面板指的是电梯轿厢内部的显示面板和各个楼层的显示面板的组合，显示面板主要显示电梯所在的楼层以及运行状态。由于电梯内部以及每层楼的显示面板所显示的信息总是相同的，因此不需要区分“内部显示面板”和“外部显示面板”，统称为“显示面板”即可。
- 楼层请求：** 楼层请求是指暂时未被满足的需要轿厢到达某个楼层的请求。当对应的按钮被按下时楼层请求将会被激活，当电梯到达所请求的楼层时楼层请求将自动被释放。

V.1.4 综述

本需求规格说明剩下的将以以下形式组织其内容：

章节V.2将提供一些有关本电梯控制软件的粗略信息，并为其后的章节做出铺垫。它包含电梯控制软件、电梯控制器以及与之交互的所有其他电梯控制系统的信息，并包含**关系图**和**用例图**来阐释他们间的关系，同时，此章节还包含电梯控制系统的功能以及限制。

章节V.3包含了对于本软件的更多需求，并用**类图**、**事件跟踪图**等多种图表、展示了需求中的对象模型等更多的细节，同时，这一部分详细描述了电梯控控制器和同其他部分之间的信号传递细节，这一章节对于开发者、测试者和维护者有更大的参考价值。

V.2 总体描述

V.2.1 交互模型

电梯控制系统与以下的电梯其他组成部分有直接的交互：

按钮

- 呼叫按钮** 呼叫按钮是在电梯外的，位于每层楼的，用于让将要乘电梯的乘客呼叫电梯的按钮。除了最高层和最低层外，每一层楼有两个按钮：上行按钮和下行按钮。最底层只有上行按钮，最高层只有下行按钮。控制器接收来自于这些按钮的信号，信号包括了请求的方向和所在的楼层。同时，控制器会向按钮上的指示灯发送信号指示按钮是否被按下。
- 楼层按钮** 楼层按钮是位于轿厢内部的按钮，楼层按钮的个数与电梯可能停靠的楼层数相等。所有的电梯按钮位于电梯内部的一个面板上，进入电梯的乘客可以通过他们来选择自己想要去哪几层楼。控制器接收来自于这些按钮的信号，信号包括用户指示的楼层数以及信号是从哪一个轿厢发出的。

开关门按钮 开关门按钮是两个按钮位于轿厢的内部，当电梯停靠完成时，乘客可以保持此开门按钮的按压状态以使电梯门持续开启，在电梯门处于开启状态时关门按钮可以使电梯门关闭。控制器接收来自于开门按钮的按下信号和释放信号，以及关门按钮的按压信号，这些信号包含它们是从哪一个轿厢发出的。

紧急停止按钮 紧急停止按钮位于轿厢的内部，当他被按下时，不论轿厢在电梯井的那个位置，都会立即停止。控制器接收此按钮的按压信号，信号包含它是从哪一个轿厢发出的。

紧急呼叫按钮 紧急呼叫按钮位于轿厢的内部。当有乘客被困于轿厢中时，紧急呼叫按钮可以用于通知电梯外部的人有人被困于电梯内部，以便采取行动。控制器接收来自此按钮的按压信号，以及信号是从哪一个轿厢发出的。

显示器

楼层显示器 楼层显示器同时位于轿厢的内部和每层楼的电梯外部。它指示了电梯当前处于哪一个楼层。控制器向楼层指示器发送信号，信号包括电梯目前处于哪一楼层。

方向显示器 方向显示器同时位于轿厢的内部和每层楼的外部，并与楼层显示器一起位于显示面板上。它指示了当前的电梯运行方向。控制器向楼层指示器发送信号，信号包括电梯的运行方向：向上、向下或者停止。

传感器

压力传感器 压力传感器位于轿厢的底部，它用于监测任意时刻电梯内容物的总重量。每当电梯中的压力有变化时，控制器接收来自于压力传感器的信号，信号包括更新后的压力数值，以及信号是从哪个轿厢发出的。

位置传感器 每一个轿厢外部都有一个位置传感器，它通过检测位置标志（见章节V.1.3来得知轿厢所在的位置。每当位置传感器检测到位置标志时，他会向控制器发送一个信号，信号的内容包括当前所在的楼层以及信号是从哪个轿厢发出的。

红外传感器

红外传感器位于电梯门 (见章节V.1.3) 的侧部。红外传感器用于检测电梯门间是否有阻挡物，以决定电梯门是否可以关闭。每当出现阻挡物或阻挡物移除时，红外传感器会向控制器发送信号，信号包括当前电梯门是否被阻挡以及信号是从哪个轿厢发出的。

机械装置

电梯门控制器

每一个电梯轿厢的顶部有一个电梯门控制器，用于控制电梯门的开闭。控制器向电梯门发送信号，通知电梯门开启或闭合。同时，每当门完全开启或闭合时，控制器还会接收到来自于电梯门控制器的信号，指示当前电梯门的状态以及信号是从哪一个轿厢发出的。

升降机

升降机位于电梯井的顶部，并通过绳索、滑轮、配重块等一系列机械装置控制电梯的升降。控制器向升降机发送信号，信号包括电梯轿厢将要移动的方向和速度，加速和减速过程由升降机自行控制。

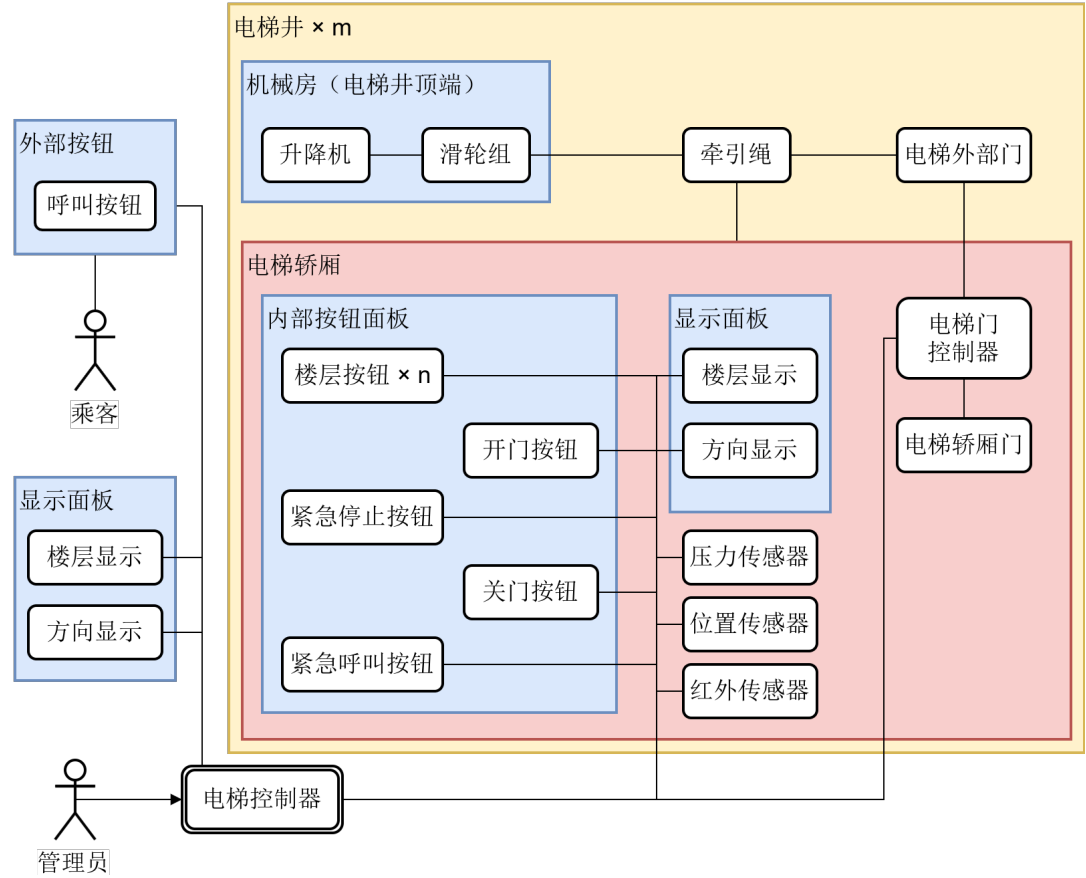


图 5.2.1: 组成部分关系图

V.2.2 产品功能

产品的主要功能是接收来自于整个电梯系统的不同部件的信号，并将之处理，然后将处理所得的结果以控制信号的方式发送给有关的电梯部件，以此达到控制整个电梯系统的目的。信号的传递与交换也是整个系统正常运作的基石。

控制系统主要通过以下几种方式与电梯系统其他部件进行信号的传递和交换，以完成其功能：

- 控制每个升降机以控制每个轿厢的移动速度和位置。
- 通过呼叫按钮和楼层按钮接收信号，将呼叫请求和楼层请求加入队列。
- 处理压力传感器传送的信号，并保证轿厢内容物在任意时刻都不会超重。
- 通过处理位置传感器发送的信号来监视电梯的位置。
- 通过向按钮上的指示灯和指示面板发送信号，来告知乘客必要的信息（如呼叫是否成功，电梯运行方向，电梯所在楼层等。
- 通过接收并处理传感器的信号并响铃来告知乘客电梯超重。
- 通过发送信号给远程设备以告知工作人员有人被困于电梯内部。
- 通过与电梯门控制器的信号交换控制电梯门的开关。

V.2.3 用户角色

对于控制器以及运作其上的控制软件而言，用户实际上是与之进行信号交换的其他电梯部件。这些电梯部件必须能够采用合适的方式与控制器进行通信。它们必须使用相同的通信协议、以及相同格式的信号。同时，类似于电梯门和升降机一类的部件必须能够及时的处理由电梯控制器发送的信号，否则整个电梯系统将不能有效的进行工作。

电梯控制器所控制的电梯系统最终是为乘客服务的，因此有必要阐明乘客与整个电梯系统之间的关系。对于由电梯控制器以及所有其它部件组成的电梯系统而言，其用户是乘客，而乘客可以对电梯系统进行以下的操作：

- 进行电梯呼叫：这一行为在电梯井外部（通常是楼道内）通过按钮进行，它包含了对电梯运行方向的请求，同时，按钮上的指示灯会表明请求是否成功。
- 进行楼层请求：在乘客进入电梯轿厢内部的时候，通过按下楼层按钮请求电梯在某一楼层进行停靠，同时对应按钮上的灯会指示请求是否成功。
- 开／关电梯门：电梯停靠时，在电梯的内部，乘客可以通过开／关门按钮控制电梯门的开关，并可以通过持续按压开门按钮使电梯门保持打开。
- 紧急停止电梯：当发生意外状况时，在轿厢内部的乘客可以通过紧急停止按钮时电梯执行紧急停止。
- 执行紧急呼叫：当被困于电梯内部时，可以通过按压紧急响铃按钮执行紧急呼叫，通知外部人员。

此外，维修人可以通过直接接触控制器来对电梯系统进行检查，或者关闭／开启电梯系统等，具体用例见图5.2.2。

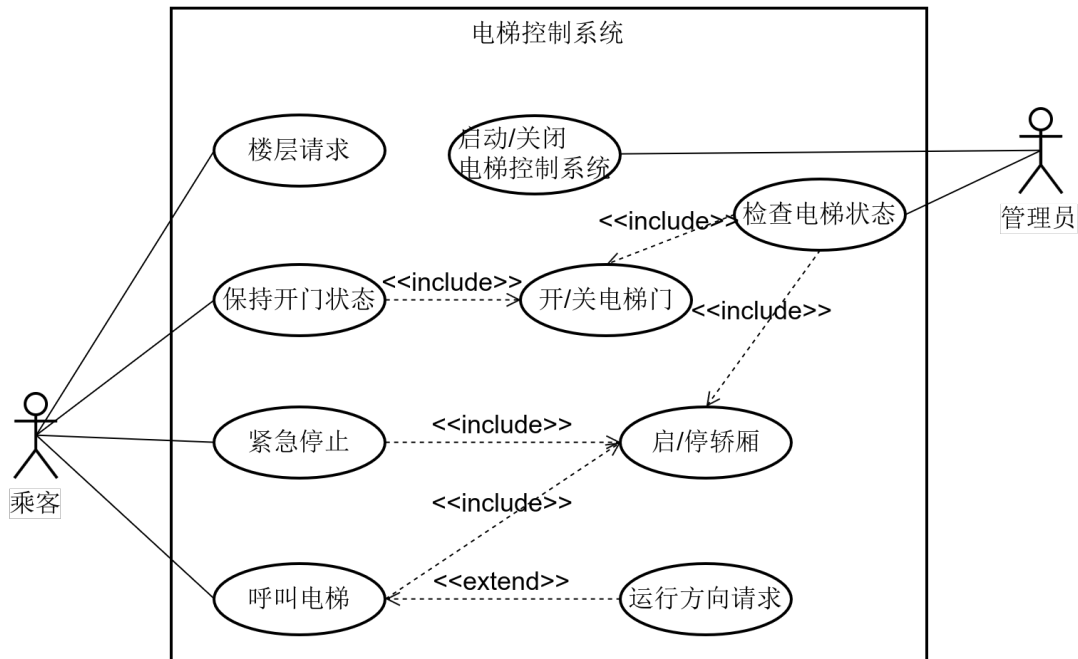


图 5.2.2: 电梯系统用例图

V.2.4 限制

控制软件的运行平台—控制器的硬件选择会对系统的设计做出一些限制，尤其是性能方面的限制，同时也会对信号传输的效率做出限制。

一些处于安全因素的考虑也会对此电梯控制系统做出限制，这些限制必须在设计的时候被考虑，它们主要体现在如下几个方面：

- 电梯门只有在电梯轿厢停靠完毕，并且停靠在正确的楼层后才能进行开关，运行时必须保持关闭状态。
- 电梯井外部的门只能在电梯轿厢停靠在此楼层时与电梯轿厢门同步开关，避免人员坠入电梯井。
- 电梯的加速度和减速度必须控制在一定的范围内以保证不对乘客造成伤害。
- 如果轿厢内容物超重，电梯必须暂停运行，直到内容物质量低于质量上限。
- 如果控制系统检测到任何关键性部件的失灵，电梯必须立即停止运行。

V.3 详细需求

V.3.1 对象模型

虽然图5.2.1已经描述了电梯系统各个组件之间的关系，但我们需要更为详细和结构化地建立整个电梯系统的对象模型，详细的了解整个电梯系统的结构对电梯控制系统的设计十分重要。从整个电梯系统角度看去的对象模型如图5.3.1所示。

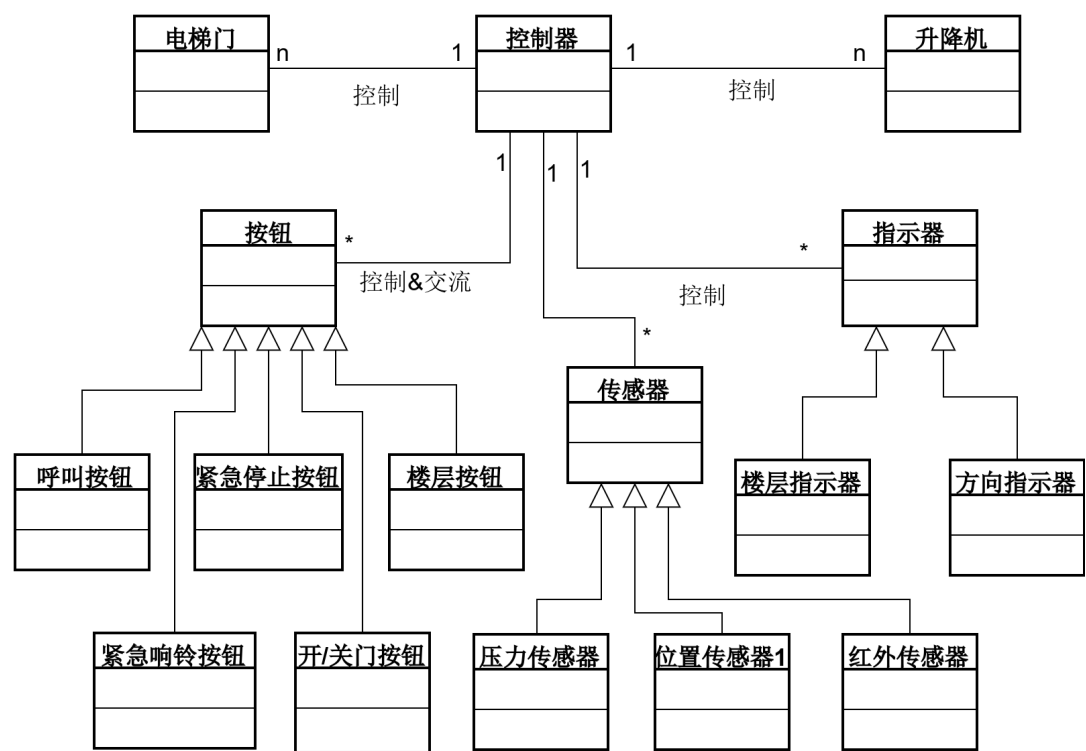


图 5.3.1: 类图-从整个电梯系统的角度看

这一类图直接描述了整个电梯系统的所有功能部件及其关系：要控制轿厢的移动和停止，就需要通过升降机类和控制器类，类似的，要控制门的开闭，就通过电梯门类以及控制器类完成，以此类推。此对象模型有助于理解整个电梯系统是如何运作的，从而进一步理解软件的需求。此外，此对象模型还说明了一些继承关系，虽然它们为电梯系统其他部分的分类继承关系而非控制软件内部的继承关系，但在软件设计是考虑此继承关系可以简化软件的设计过程，提高部分模块的复用率。

V.3.2 功能需求

这一部分将主要使用脚本和事件跟踪图描述电梯控制系统将如何与其他电梯部件合作，从模拟事件的过程中体现对功能的需求。

电梯正常运作 表V-3-6展示了在正常情况下，用户从一个楼层呼叫电梯、进入电梯、按下楼层按钮、走出电梯的过程。图5.3.2是对应的事件跟踪图。可以看出，在正常工作的情况下，所有的终端型硬件都参与了运行，以保证整个系统的正常工作。

表 V-3-6: 电梯正常运作脚本

乘客在一楼按下上行呼叫按钮，此时电梯 A 停靠在九楼，电梯内无人。

乘客观察到按钮被激活，同时指示面板显示电梯开始下行。

乘客观察到楼层从九楼不断降低到一楼。

电梯到达一楼，电梯门打开，乘客进入电梯。

乘客按下到七楼的按钮，一段时间后电梯门关闭。

乘客观察到指示面板显示电梯开始上行，楼层从一楼不断增加到七楼。

电梯门打开，乘客走出电梯，电梯门关闭。

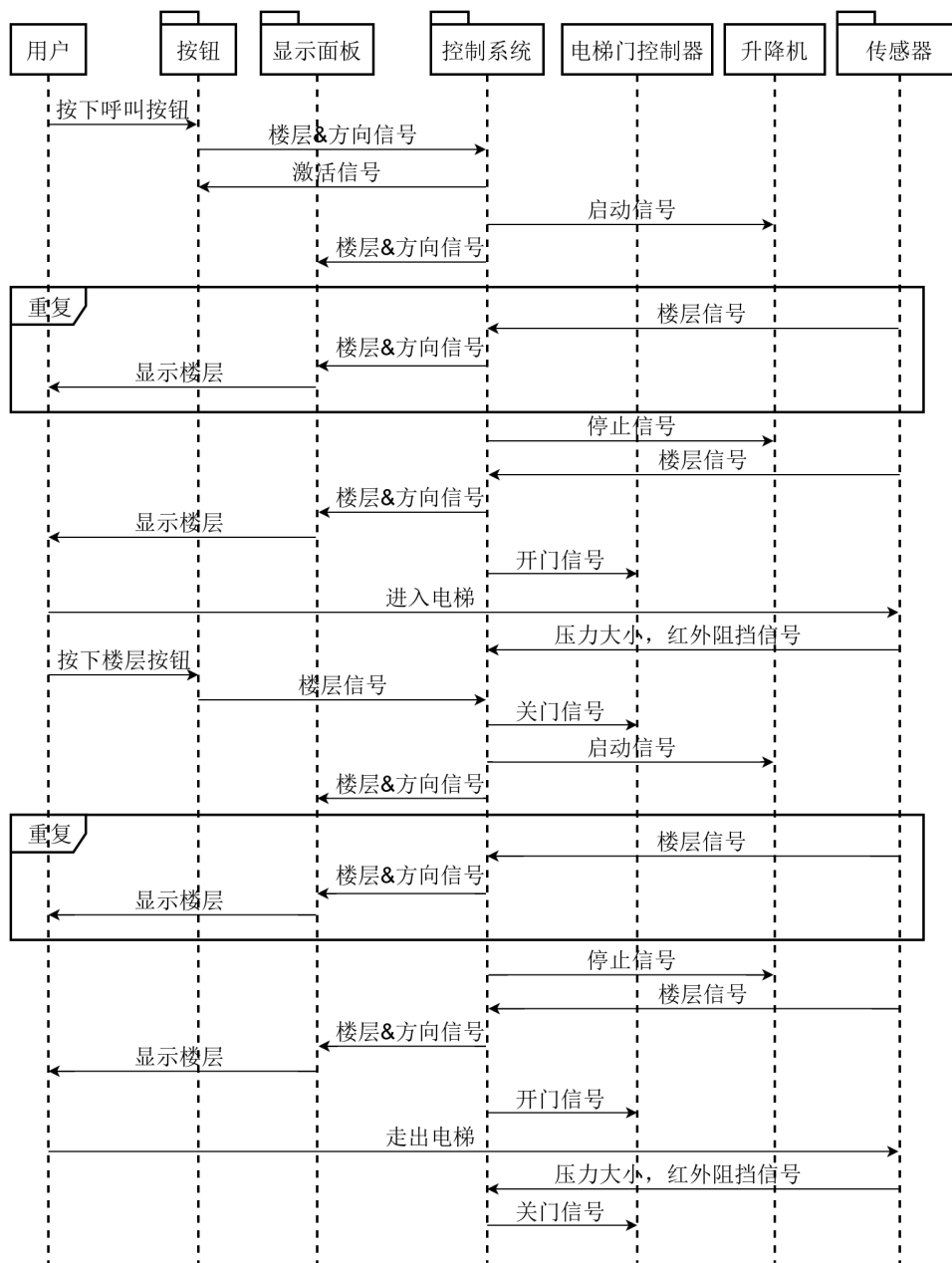


图 5.3.2: 事件跟踪图-电梯正常运作

复杂情况下的电梯正常运作 表V-3-6以及图5.3.2仅展现了电梯在最简情况下的正常运作方式，而实际请款远比这复杂。如表V-3-7展示了在较为复杂情况下的正常运作方式，其中假设大楼一共有 17 层，其中有 3 部电梯。由于时间跟踪图过于复杂，在此不再给出。

表 V-3-7: 复杂情况下电梯正常运作脚本

41 名乘客在一楼呼叫电梯，此时电梯 A 停靠在 14 楼，电梯 B 停靠在 12 楼，电梯中均无人。电梯 C 在 6 楼，其中有 2 名乘客，正欲去往 17 楼。
一楼的乘客观察到呼叫按钮被激活，此时指示面板显示电梯 B 开始下行。
电梯 B 下行到 12 楼时 6 楼有呼叫电梯按钮被按下，此时有两名乘客在 6 楼等待下行，电梯 C 已运行到 9 楼。
电梯 B 继续下行，在 6 楼停靠，电梯 B 门开启，两名乘客进入电梯，并按下 1 楼的楼层按钮。
电梯 B 关门，启动，继续下行，此时在 3 楼有人按压上行呼叫按钮。
电梯 C 在顶楼 17 楼停靠，电梯中的 2 名乘客走出电梯，电梯停靠在 17 楼。
电梯 B 直接下行到 1 楼，开门，电梯中的两名乘客走出电梯。
40 名乘客依次进入电梯 B，当进入到第 21 名乘客后，电梯蜂鸣提示超重，第 21 名乘客退出电梯。
电梯 B 门关闭，电梯 B 内的 5，10，17 楼楼层按钮被按下。退出的第 21 名乘客决定走楼梯。
一楼的上行按钮再次被按下，电梯 A 开始下行。
电梯 B 忽略 3 楼上行请求直接上行到 5 楼时，开门，6 名乘客走出电梯 B，电梯 B 关门，继续上行，此时电梯 A 下行到 14 楼。
电梯 B 上行到 10 楼，开门，7 名乘客走出电梯，电梯 B 关门，继续上行。此时电梯 A 下行到 7 楼，4 楼上行呼叫按钮被按下，3 名乘客欲上行。
电梯 A 下行到 1 楼，开门，1 楼剩下的 20 名乘客走入电梯，电梯 A 关门，此时电梯 A 内的 5，10，17 楼楼层按钮被按下。
控制系统指示电梯 C 下行，电梯 C 开始下行。
电梯 B 上行到 17 楼，开门，所有乘客走出电梯，电梯 B 停靠在 17 楼。
电梯 A 忽略 3 楼和 4 楼的上行信号直接上行到 5 楼，开门，10 人走出电梯，电梯 A 关门，继续上行，此时电梯 C 下行到 5 楼。
电梯 C 下行到 3 楼，开门，3 楼的两名乘客进入电梯，并按下 17 楼楼层按钮，电梯 C 关门，开始上行。此时电梯 A 上行到 8 楼。
电梯 A 上行到 10 楼，开门，5 人走出电梯，电梯 A 关门，继续上行。
电梯 C 上行到 4 楼，开门，3 人进入电梯，见 17 楼按钮已被按下，便直接按下关门按钮。电梯 C 关门，开始上行。

电梯 A 上行到 17 楼，开门，所有人走出电梯。电梯 A 关门，并停靠在 17 楼。

电梯 C 一直上行到 17 楼，开门，所有人走出电梯，电梯 C 关门，停靠在 17 楼。

电梯异常运作 表V-3-8展示了一部电梯在异常情况下的运作脚本。

表 V-3-8: 电梯异常运作脚本
电梯 A 停靠在 7 楼，此时一楼一名乘客按下呼叫按钮。
从显示面板上看到电梯从 7 楼下到一楼。
电梯门开启，乘客进入电梯，并按下 6 楼的楼层按钮，电梯门关闭。
电梯开始上行，上行到 3、4 楼之间时，控制系统监测到升降机出现故障，电梯紧急停止，乘客被困于电梯内。
乘客按下紧急响铃按钮，信号被发送给电梯外的工作人员。
工作人员检查电梯升降系统后，手工控制电梯控制器，使电梯门打开，将人员救出。
工作人员关闭整个电梯系统，开始对其进行维修。
维修完毕并检测无误后，工作人员重新开启电梯系统。

从以上的脚本可以对控制系统的功能需求有更直观的了解：首先，控制系统需要保证能满足所有乘客的运送需求；其次，控制需要保证乘客的安全；最后，控制系统需要电梯时刻正常运行，在非正常情况下需要停止运行并向外界发送请求修理的信息。

V.3.3 性能需求

控制系统除了完成其功能外，还需要在以下几个方面尽可能提高其性能，以节约资源，提升用户体验。

- 为了提升用户体验，需要在一定时间段内，使用户的平均等待时间最小。
- 由于电梯运行最耗能时是在加速以及减速阶段，因此需要在一定时间段内，使所有轿厢加速以及减速的次数最少。
- 为了节省能源，除了使加速和减速的次数最少外，还需尽可能使每次电梯的载客量达到最大。
- 对于电梯控制软件要求起响应速度大于其他所有与之交流的部件响应速度的最大值，否则信号传递效率过低，电梯无法正常工作。

V.3.4 外部接口需求

为了满足电梯控制系统的功能需求和性能需求，对于电梯控制系统的外部接口必须做出规定。

- 为了实现正常的控制功能，各个外部电梯部件需要根据其功能能够发送控制器能够接受并解释的信号，或者能够接受并解释控制器发送的信号，对于不同的电梯控制系统信号不一定需要统一内容，但一定需要能够被转换为能够被控制器解释的内容，或者能够通过控制器将需要发送的信号转换成对应的格式。对于需要的信号内容见表V-3-9。
- 为了使控制系统能够与电梯的各个部件正常的通信，各个部件的接口应该使用统一的通信协议和信号格式。
- 为了实现通信的高效性，需要选择一种传输效率较高的通信协议。
- 由于控制器是固定的，而轿厢是移动的，且距离较远，通信协议应该保证信号能够远距离线缆传输而不失真，或者直接采用无线通信协议。
- 为了保证各个监视器的数值能够不间断的传达到控制器，接口间需要选用一种稳定的通信协议。
- 控制软件应该留出软件接口以保证可扩展性，以及能够使用其他工具对控制器实现直接控制。

表 V-3-9: 输入输出信号需求

输入信号		
组件	信号	说明
维修开关	turnOn()	维修开关打开
	turnOff()	维修开关关闭
呼叫按钮	pressed(floor, direction)	呼叫按钮按下（楼层，方向）
	released(floor, direction)	呼叫按钮弹起（楼层，方向）
楼层按钮	pressed(floor, cabID)	楼层按钮按下（楼层，电梯标号）
开门按钮	pressed(cabID)	开门按钮按下（电梯标号）
	released(cabID)	开门按钮弹起（电梯标号）
紧急停止按钮	pressed(cabID)	紧急停止按钮按下（电梯标号）
紧急响铃按钮	pressed(cabID)	紧急停止按钮弹起（电梯标号）
电梯门控制器	doorOpened(cabID)	电梯门完全开启（电梯标号）
	doorClosed(cabID)	电梯门完全关闭（电梯标号）
位置传感器	markDetected(cabID)	检测到位置标志（电梯标号）
压力传感器	weightChanged(cabID, weight)	压力发生变化（电梯标号，更新后压力）
红外传感器	blocked(cabID)	电梯门被阻挡（电梯标号）
	unBlocked(cabID)	电梯门未被阻挡（电梯标号）
输出信号		
组件	信号	说明
升降机	move(direction, speed)	升降机移动（方向，速度）

电梯门控制器	open()	开门
	close()	关门
方向显示器	show(direction)	显示方向 (方向)
楼层显示器	show(floor)	显示楼层 (楼层)
呼叫按钮	lightOn()	亮灯
	lightOff()	灭灯
楼层按钮	lightOn()	亮灯
	lightOff()	灭灯
紧急状况警铃	ring()	响铃
超载警铃	ring()	响铃

V.3.5 假设条件

在整个电梯控制系统的设计过程中，以下的假设成立。这些对于电梯系统的假设同时会影响电梯系统的设计。

- 电梯所工作的建筑有 m 层楼，其中 $m \geq 2$ 。
- 一共有 n 部电梯在建筑中工作，其中 $n \geq 1$ ，每一个电梯工作于一个独立的电梯井中。
- 电梯控制器应该使电梯系统合理的工作，并尽可能的优化性能，但是电梯系统不必要对不合理的或者突然增多的乘客做出特殊的动作。
- 电梯的其他部件必须通过实现设计好的同一种协议和信号格式进行交流。
- 电梯有部分应急措施，这些应急措施将被分别对待，整个电梯系统没有全局紧急模式。
 - 紧急响铃按钮被按下：这只会触发紧急响铃以及发送信号给外部人员，不会导致电梯系统做出其他动作。
 - 紧急停止按钮被按下：这会导致被按下按钮所在的轿厢执行紧急停止，并进入应急状态，其他的轿厢不会受到影响，依然会正常工作。紧急停止按钮被多次按下时，只有第一次按压是有效的。
 - 压力传感器检测到超重：这只会导致超重的轿厢停止运行并响铃，不进入紧急状态，也不影响其他轿厢。当压力恢复正常时电梯继续正常工作。
 - 红外传感器检测到门被阻挡：与压力传感器检测到超重相同，不会影响其他电梯运行，并且当阻挡被移除时电梯恢复正常运行。

VI 概要设计

VI.1 设计思想

需求说明书总体上是将电梯控制系统看做一个黑盒，并通过阐明控制系统与其他部分的关系来说明电梯控制系统所应该拥有的功能。在概要设计的过程中则采用面向对象的设计方法，将电梯控制系统的各个组成部分以及整个电梯系统的其他部分抽象成类，类之间通过信号进行交流，以此为依据，根据需求分析对需求进行详细展开，从而完成对控制系统的概要设计。

VI.2 系统功能与目的

章节V.2.2已经对产品功能做出了简述，此处再次对电梯控制系统的功能与目的做出详细的描述。电梯控制系统需要由调度模块、安全控制模块、升降机控制模块、按钮控制模块、电梯门控制模块、显示控制模块和传感器控制模块组成。其中按钮控制模块和传感器控制模块又根据它们的信息交换对象的不同被分为不同的子模块，他们各自实现如下的功能。

- **调度模块**：调度器是电梯控制系统中最重要部件，它不对实际的电梯组件进行直接控制，而是通过一些列算法对传感器模块接收到的信号进行处理，并通过控制其他控制模块对整个电梯系统进行控制。
- **安全控制模块**：安全控制模块不对实际的电梯组件进行直接控制，而是通过读取传感器模块接收到的信号进行处理，并通过控制其他控制模块达到对整个电梯系统的安全保障。
- **按钮控制模块**：按钮控制模块是电梯控制系统中与外部按钮交互的子模块，依据功能又被分为呼叫按钮控制模块、楼层按钮控制模块、紧急停止按钮控制模块和紧急响铃按钮控制模块，它直接接收来自于外部按钮的信号并向外部按钮发送灯光信号以控制外部按钮的行为。
- **传感器控制模块**：传感器模块是电梯控制系统中与外部传感器直接交互的子模块，它接收来自于外部传感器的信号并向电梯控制系统中的其他子模块提供信息。
- **电梯门控制模块**：电梯门控制模块直接与电梯门进行交互，控制电梯门的开关，并接收电梯门的完全开闭信号，为控制系统中的其他模块提供必要信息。
- **升降机控制模块**：接收来自于调度模块的信号，直接对升降机发出信号对其进行控制。
- **显示控制模块**：向各个需要向用户显示的部件发送对应的信号，更新显示的内容（主要是当前楼层以及电梯运行方向）

这些模块以其与外部组件的交互为依据被划分到各个子系统内，包括信号处理子系统、安全子系统、探测子系统以及一个维修控制子系统。图6.2.1展示了包含子系统的用例。此图并不包含本章所描述所有的模块，仅仅包含与外部组件进行交互的模块。具体的模块关系见章节VI.3。同时，本章节所描述的模块又根据其功能有被更详细的划分为子模块，详见章节VII。

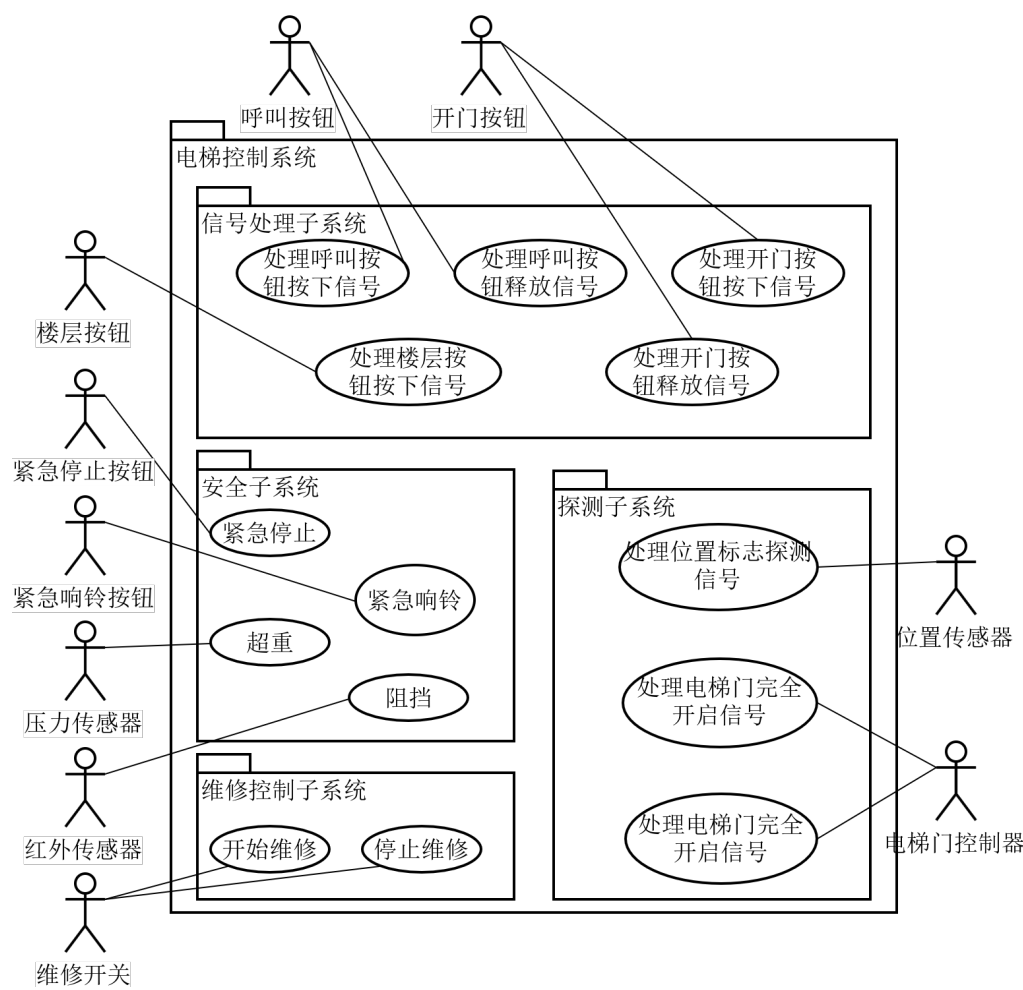


图 6.2.1: 子系统分组用例图

VI.3 总体结构设计

图5.3.1已经对于电梯控制系统系统与外部设备的控制关系以及继承关系做出了描述。图6.3.1将对电梯控制系统内部的结构进一步做出更详细的描述。从此图中可以看出电梯控制系统内部各初步划分模块之间的继承关系以及其与外部模块之间的交互关系。

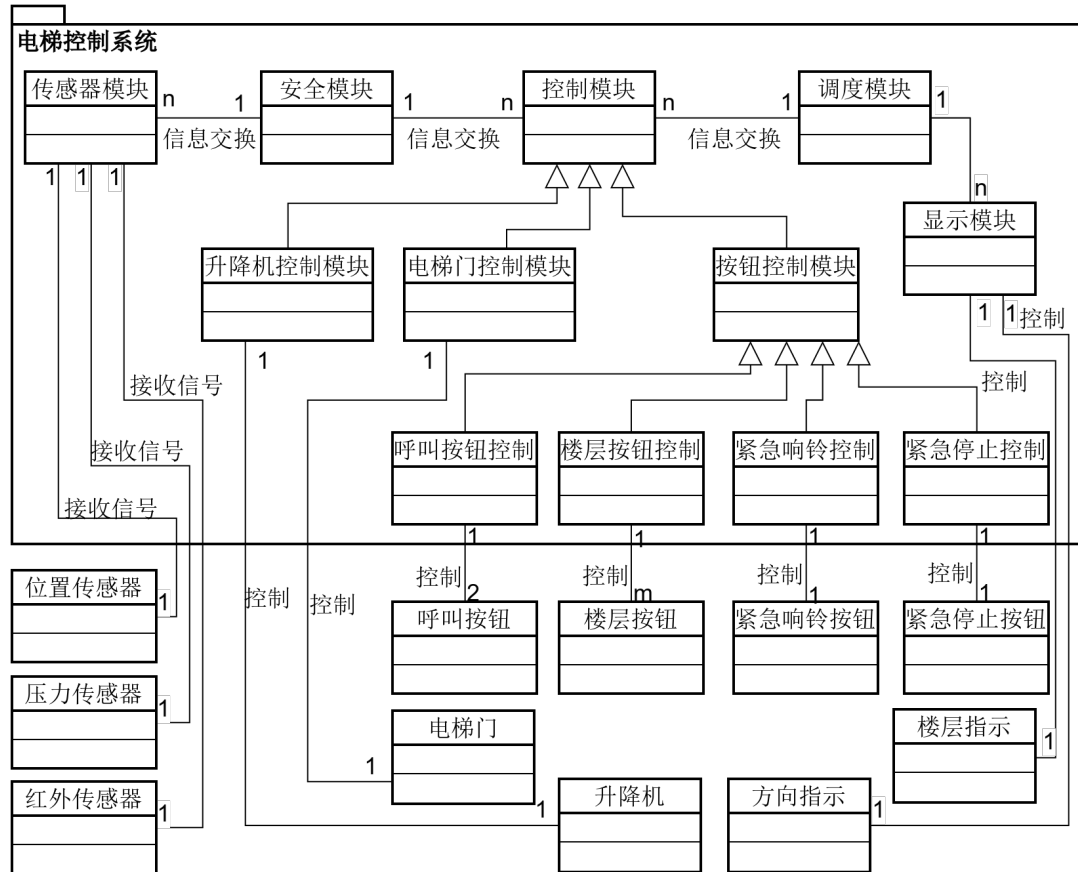


图 6.3.1: 类图-电梯控制系统

VI.3.1 用例执行流程

为了进一步说明电梯调度系统的运作方式，本节将从不同的用例入手，使用顺序图的方式展示用例的执行流程，为详细设计以及实际对电梯控制系统的实现搭好框架。为了使顺序图不至于过于复杂，本节会将用例所执行的功能拆解为较小的功能部分，表VI-3-10。展示了这些部分。其中，指示功能和感应功能的响应随时处于激活状态，也就是说，在控制功能中这些功能会同时作用，因此不再在控制功能的顺序图中画出。

表 VI-3-10: 用例及其所属功能部分

功能分类	用例	
控制功能	呼叫信号处理	开/关电梯门
	楼层信号处理	紧急停止
	启动/停止轿厢	紧急响铃
指示功能	指示电梯位置	指示电梯移动方向
感应功能	处理压力改变信号	处理红外感应改变信号
	处理位置改变信号	

呼叫信号处理 呼叫信号处理包括两种可能性，一种是电梯正在朝着乘客希望方向的同方向移动，另一种则是朝反方向移动。两种可能性的顺序图是一样的，唯一不同的是用户的等待时间。

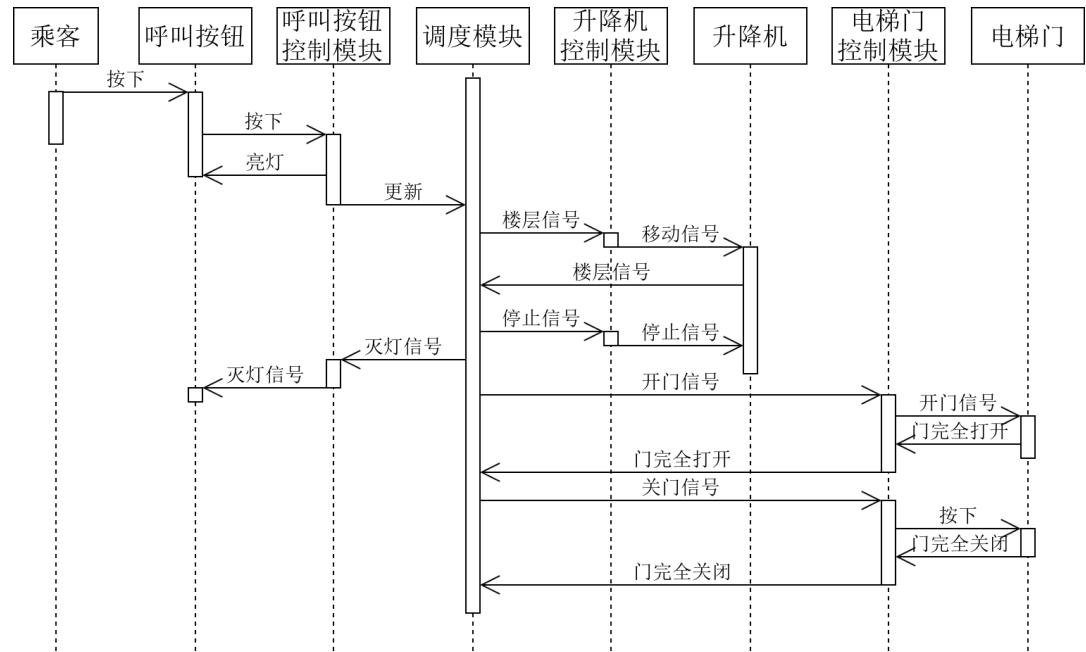


图 6.3.2: 呼叫信号处理

楼层信号处理 与呼叫信号处理相同，楼层信号处理包括两种可能性，一种是电梯正在朝着乘客希望方向的同方向移动，另一种则是朝反方向移动。两种可能性的顺序图是一样的，唯一不同的是用户的等待时间。

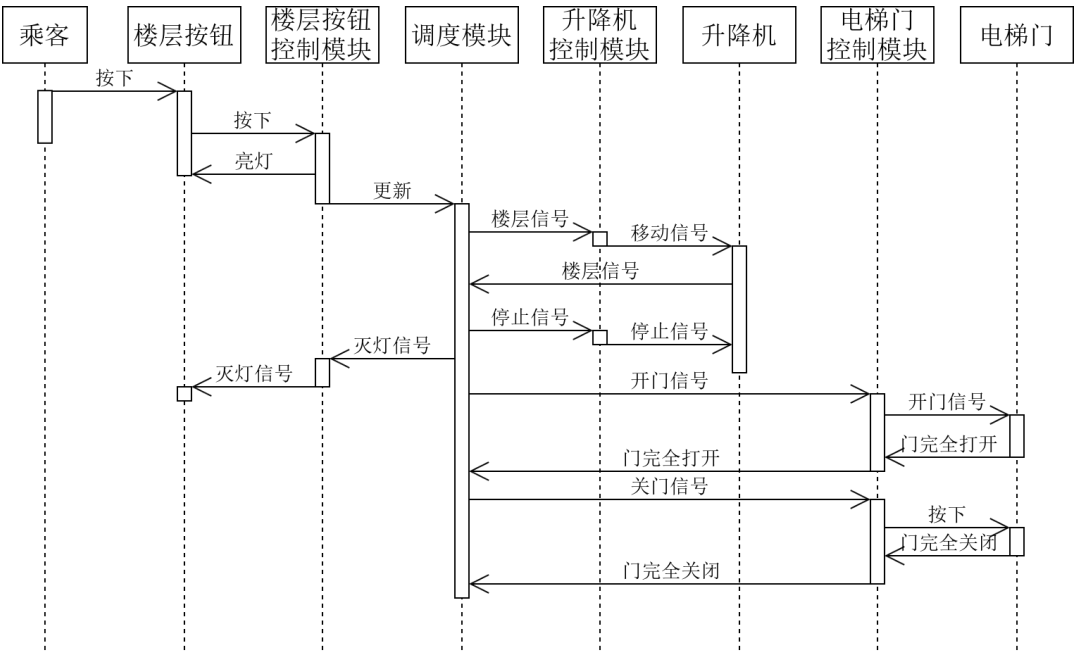


图 6.3.3: 楼层信号处理

启动/停止轿厢 对于轿厢的启动，移动速度应该在限定的速度和加速度范围内从慢到快，而对于轿厢的停止，应该在限制范围内从快到慢。无论是启动还是停止，其移动方向都有向上或者向下两种，两种的顺序图相同，仅仅是方向不同。

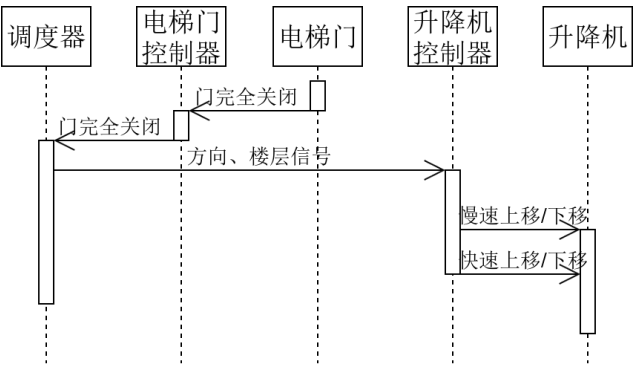


图 6.3.4: 启动轿厢

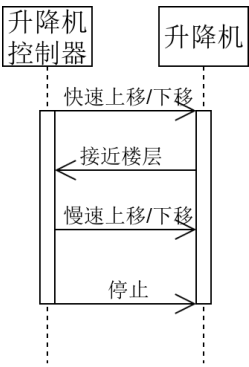


图 6.3.5: 停止轿厢

开/关电梯门 当电梯从运行状态到达停止状态并且停靠在某一楼层时，电梯门应该打开一段时间，当超时或者关门按钮被按下时，电梯门应该关闭；此外，在门未被完全关闭时，如果开门按钮被按下，则应该停止关门，重新打开电梯门。关门必须在门未被阻挡的情况下进行。在任何情况下若门被阻挡，则立刻停止关门。顺序图6.3.6和6.3.7不考虑门被阻挡的情况。

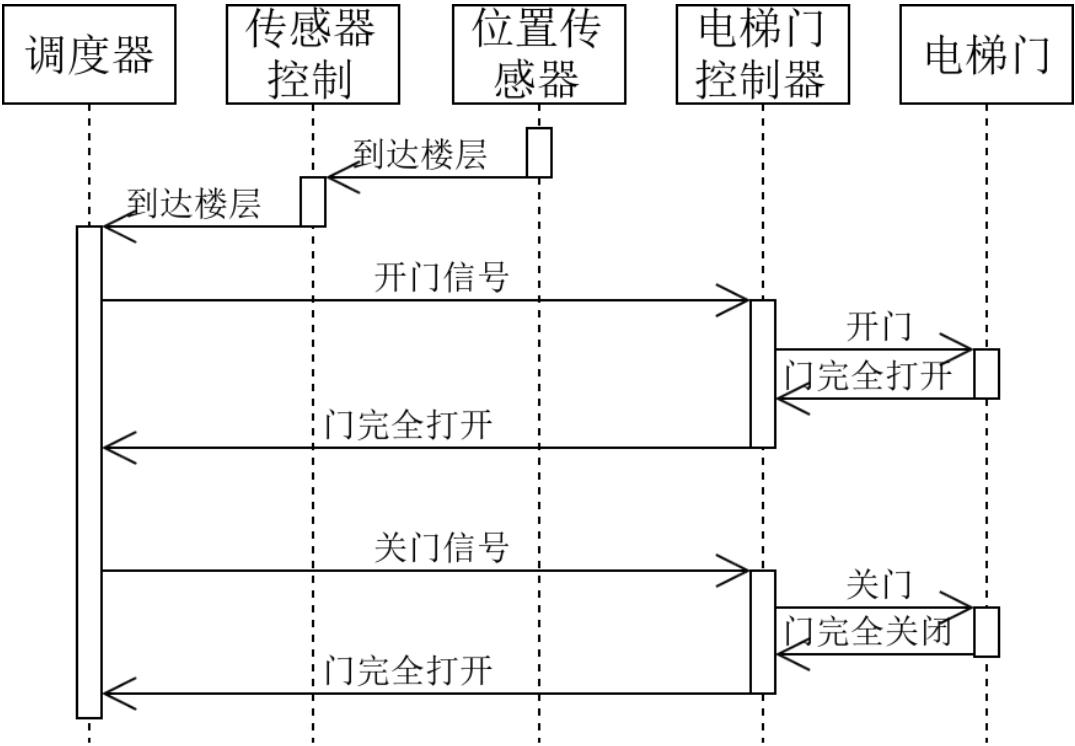


图 6.3.6: 开门/关门

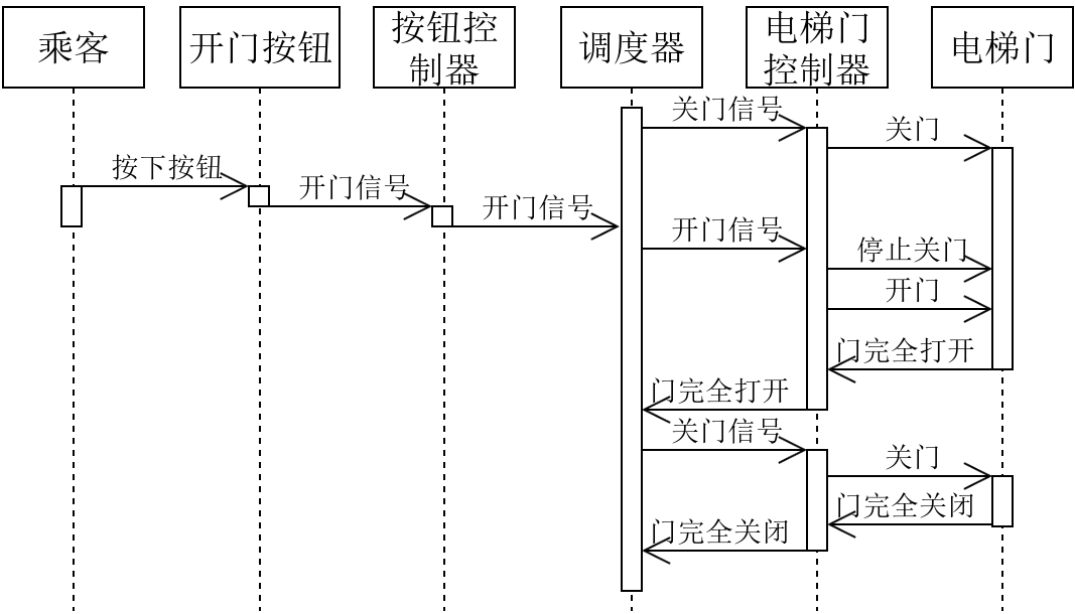


图 6.3.7: 在门未完全关闭时开门

紧急停止 紧急停止在三种情况下会被触发：紧急停止按钮被按下、轿厢应该停止但未停止、门在轿厢运行时打开。这三种情况见图6.3.8、6.3.9和6.3.10

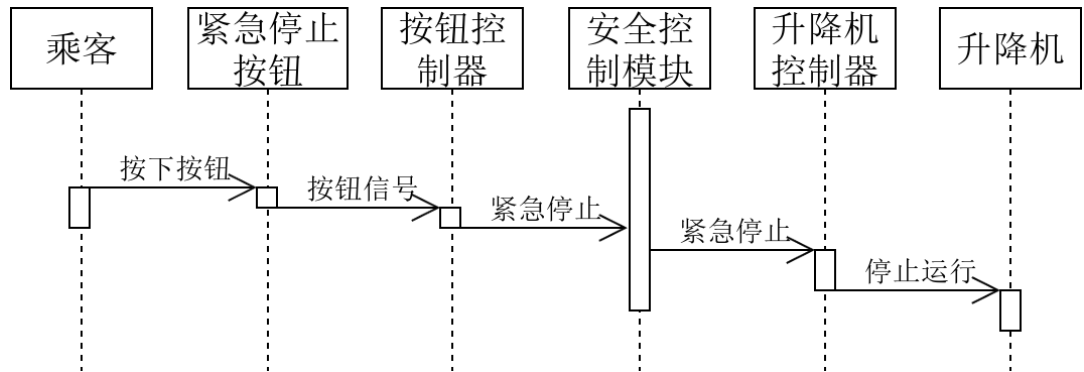


图 6.3.8: 紧急停止按钮被按下

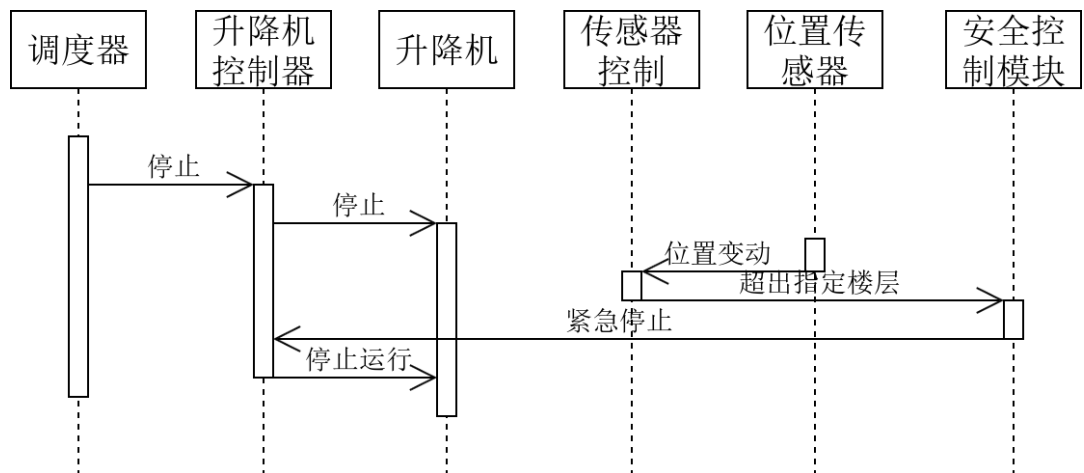


图 6.3.9: 轿厢应该停止但未停止

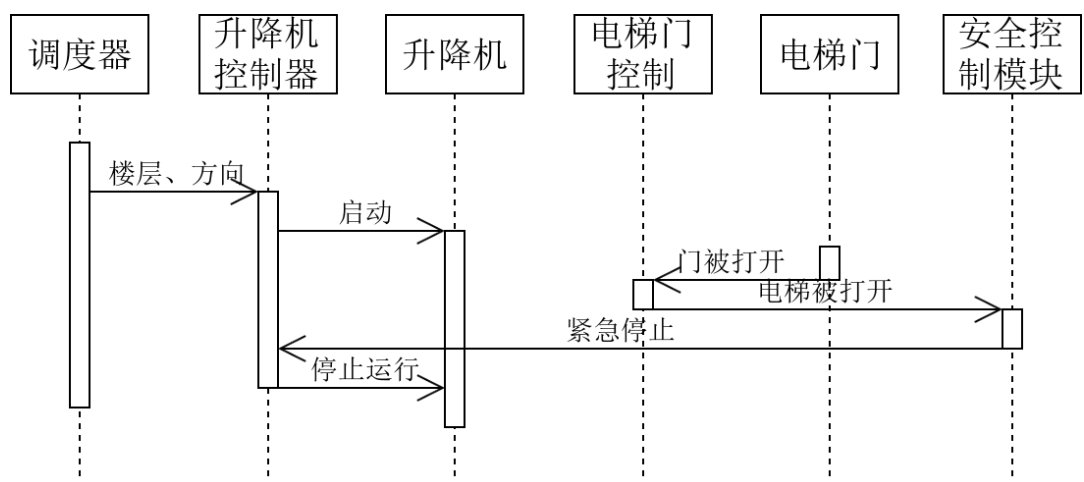


图 6.3.10: 运行时轿厢门被打开

紧急响铃 紧急响铃按钮独立于调度器运行，当用户按下响铃按钮，按钮模块直接发送信号给响铃模块，并发送信号给外部设备以通知外部人员。

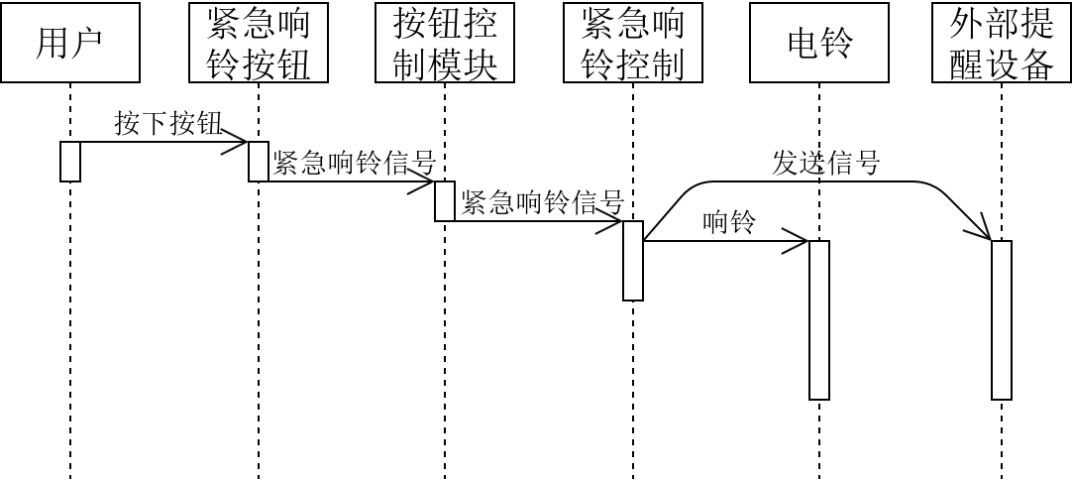


图 6.3.11: 紧急响铃按钮被按下

指示电梯位置 每当位置传感器检测到位置标志时，需要更新电梯内外的位置指示器以告知用户目前的电梯位置。

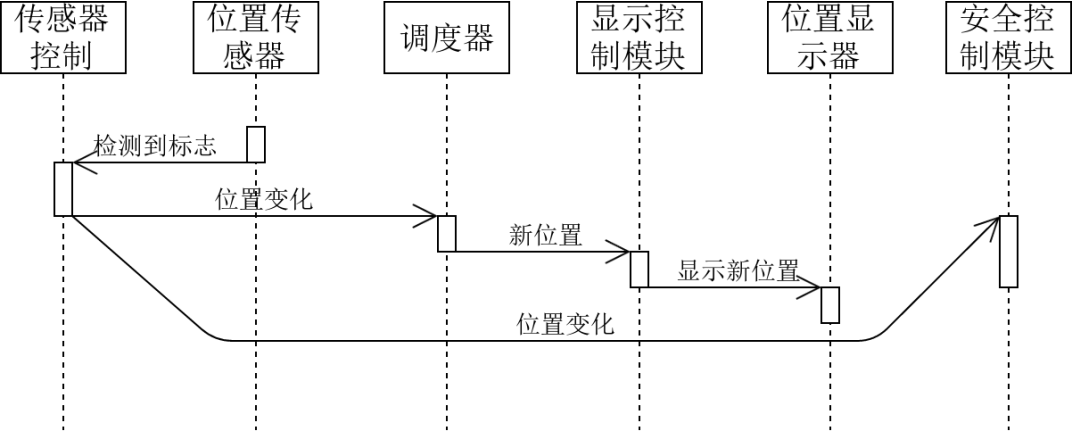


图 6.3.12: 更新电梯位置

指示电梯方向 当电梯的方向改变时，需要更新电梯的方向。方向有三种情况：向上、向下或静止（不显示）。电梯方向的改变总是由调度器决定的并且只会在运行状态（启动、停止）改变的时候改变。

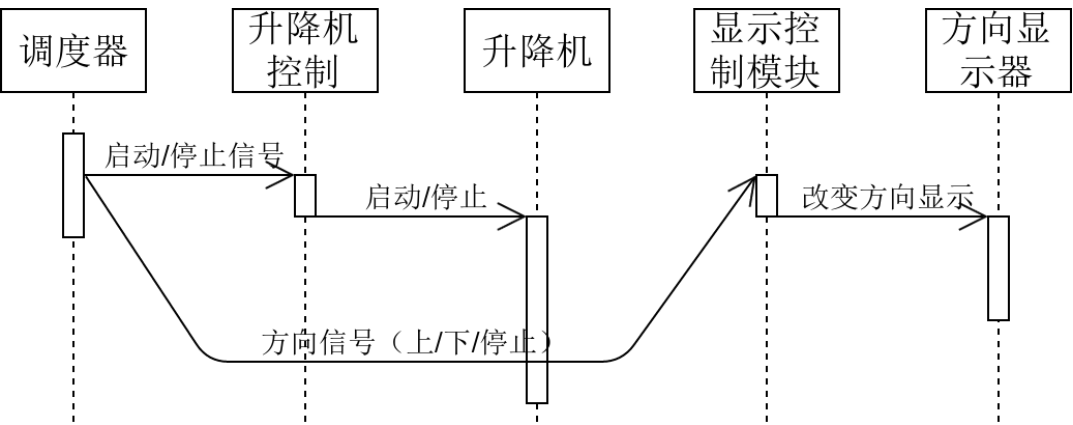


图 6.3.13: 电梯启动时更新电梯方向

处理压力改变信号 当电梯压力传感器的信号发生改变时，电梯需要做出是否超重的判断，以决定电梯是否继续运行。

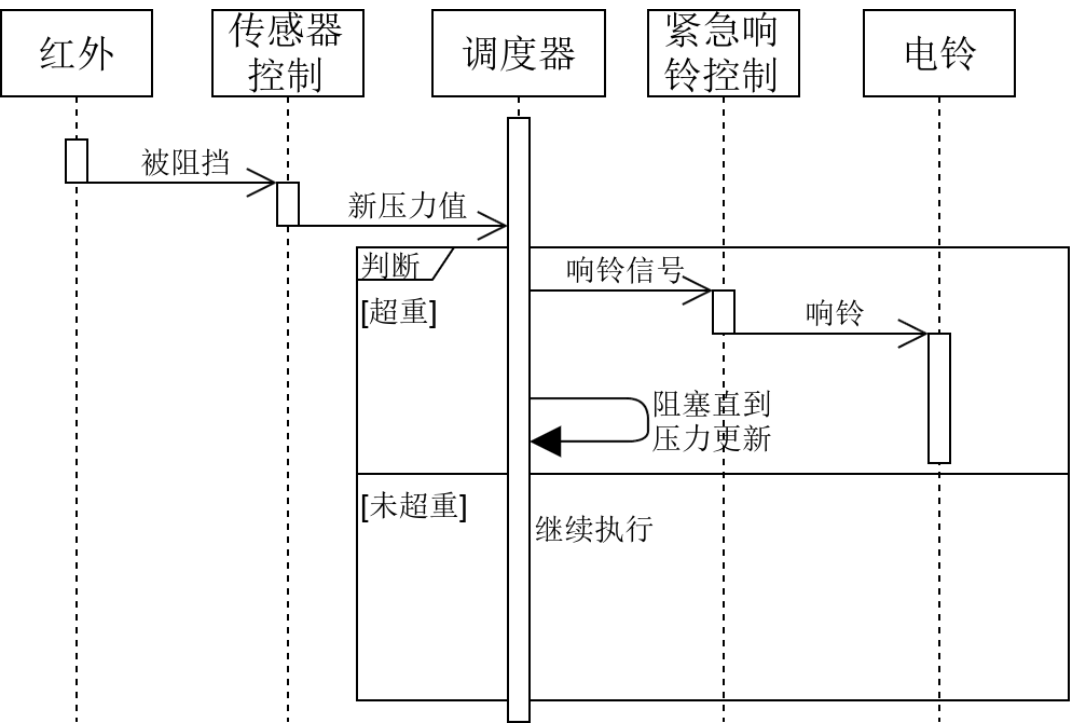


图 6.3.14: 处理压力传感器信号

处理红外感应改变信号 当红外传感器发出被阻挡的信号时，电梯控制系统需要立刻阻塞，直到阻挡信号被移除，才可以继续执行关门，启动电梯等一系列活动。

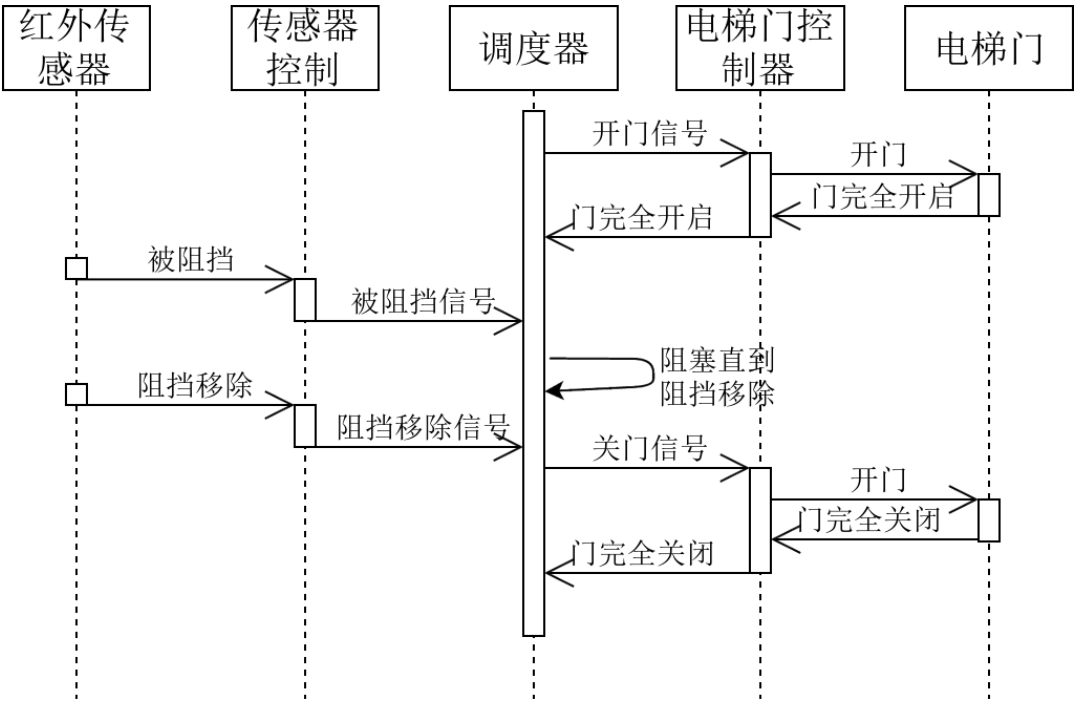


图 6.3.15: 红外感应信号改变

处理位置改变信号 当位置发生变化时，需要改变电梯位置指示，通知调度器 and 安全管理模块。有关位置改变信号的功能已在图6.3.2 ~ 图6.3.15中给出，此处不再重复绘制。

VI.3.2 状态转换图

图6.3.16展示了整个系统的状态转换图，虽然此图未能显示细节内容，但其极为重要。它展示了电梯控制系统中的并行组件，并行对于此电梯系统十分重要，因为绝大部分针对电梯控制系统的输入信号需要在输入发生后立刻被处理。此外，信号未被处理造成丢失的可能性将变小，因为整个电梯控制系统在任意时刻必然处于一个特定的状态。图6.3.16中展示的五种状态必须全部并行的执行。

通常情况下，当一个程序的并行化较高时，信号同步将会成为较为棘手的问题。但在本系统中此问题发生的概率较小。所有的重要信号会存放于队列中。两个记录子状态只负责向队列末尾添加信号，而调度子状态负责从队列头移除并处理信号。另外两个子状态可以同步的被执行而并不会对整个控制器造成影响。也就是说，即便一个轿厢被紧急停止或进入紧急状态，并不会影响调度系统对于其他轿厢以及其他组件的控制。当这一轿厢从紧急状态恢复时，又可以立即投入运行。

注：以下状态图中加粗的状态为一个子状态图。



图 6.3.16: 状态图-电梯控制系统

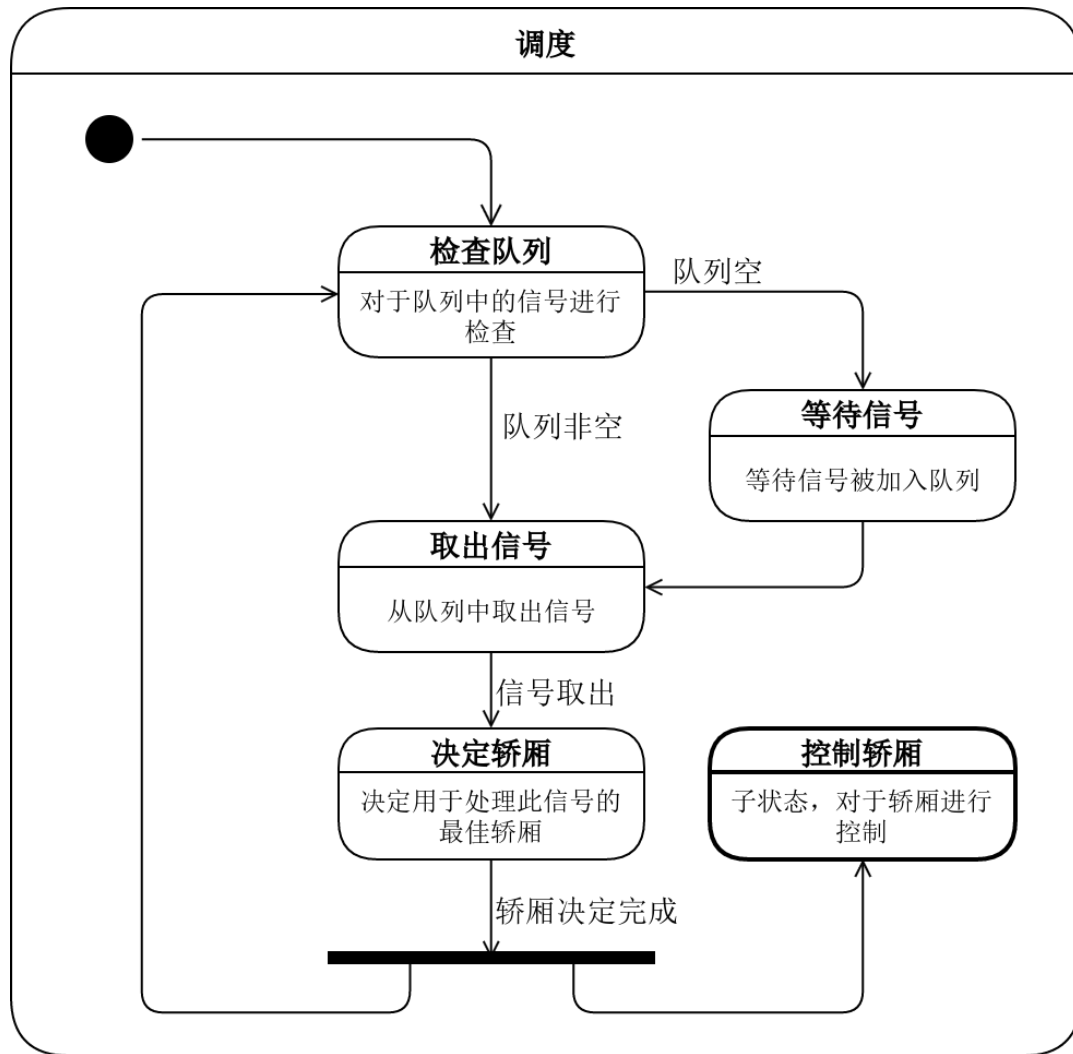


图 6.3.17: 状态图-调度

图6.3.17中的 fork 十分重要。它意味着当用于处理这个信号的轿厢被决定后，当前进程/线程出现两个分支。一个分支用于处理信号，另一个分支继续等待信号，这是电梯控制系统能够同时操作多部电梯的基础。当一个信号被处理时，主线程不会阻塞在对这个信号的处理上，这样才能做到对于多部电梯的并行处理。

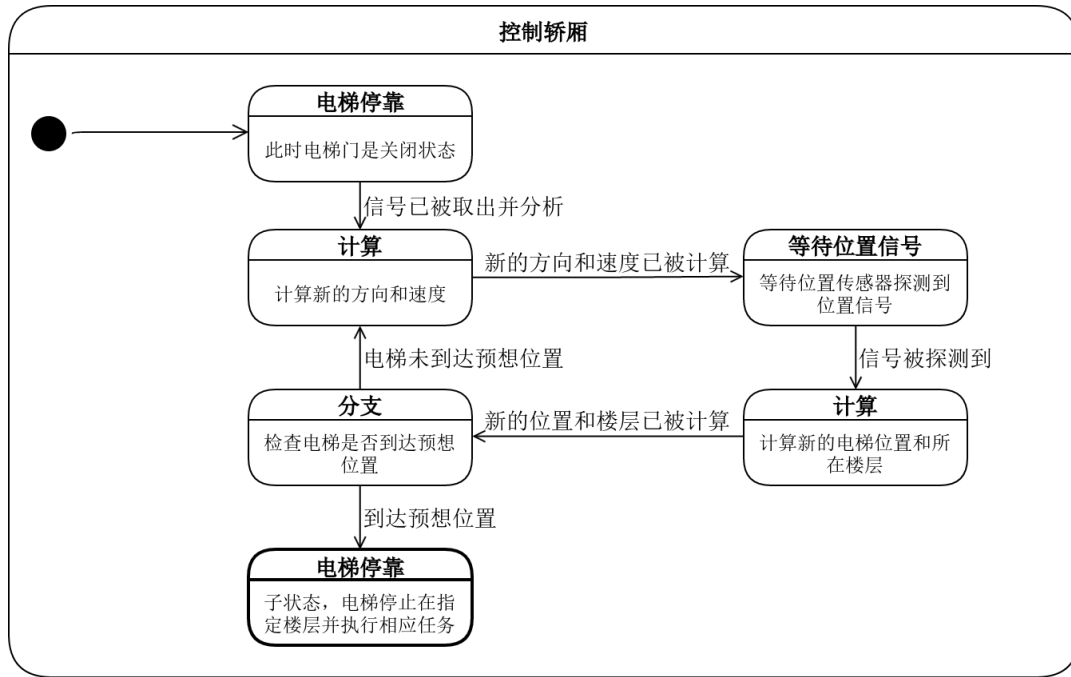


图 6.3.18: 状态图-控制轿厢

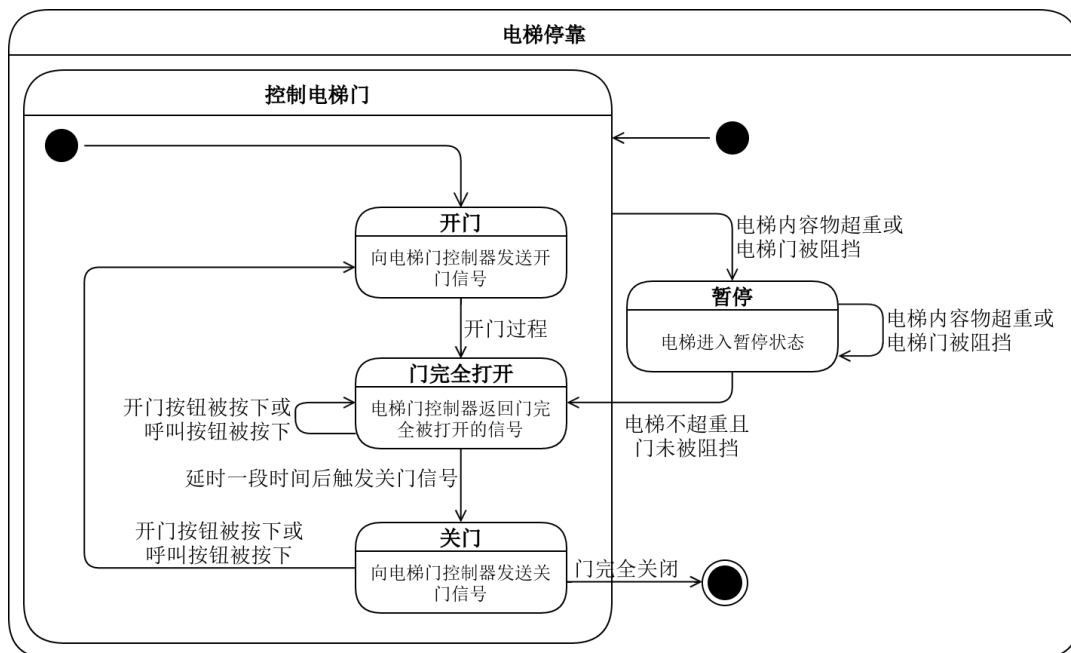


图 6.3.19: 状态图-电梯停靠

在图6.3.19中我们可以发现暂停状态可以由开门、门完全开启和关门中的任意一个状态进入。这是因为在三种状态中门都有可能被阻挡，尤其是在关门状态，在门没有完全被关闭时可能有乘客跑进电梯。由暂停状态恢复只会进入开门状态，以保证安全。

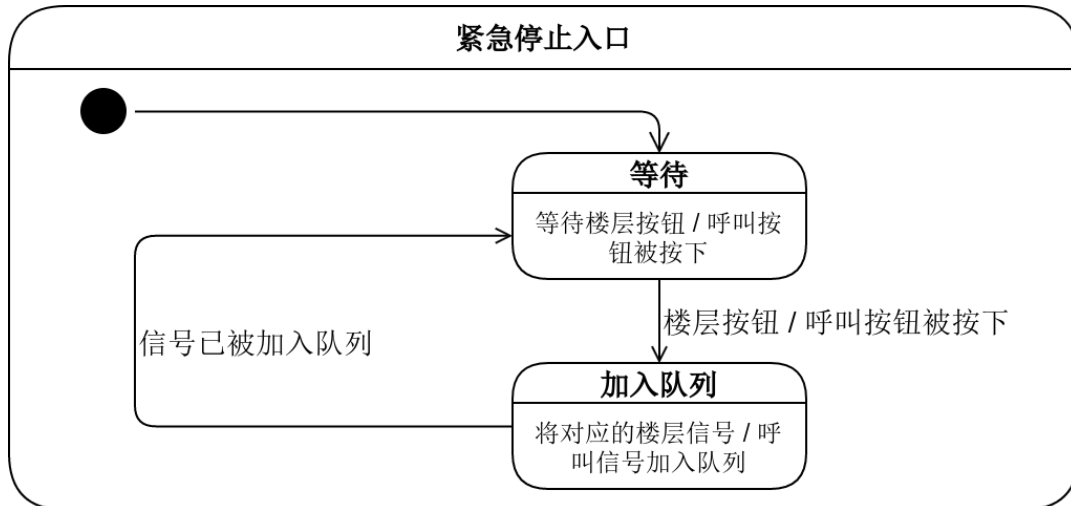


图 6.3.20: 记录楼层/呼叫信号

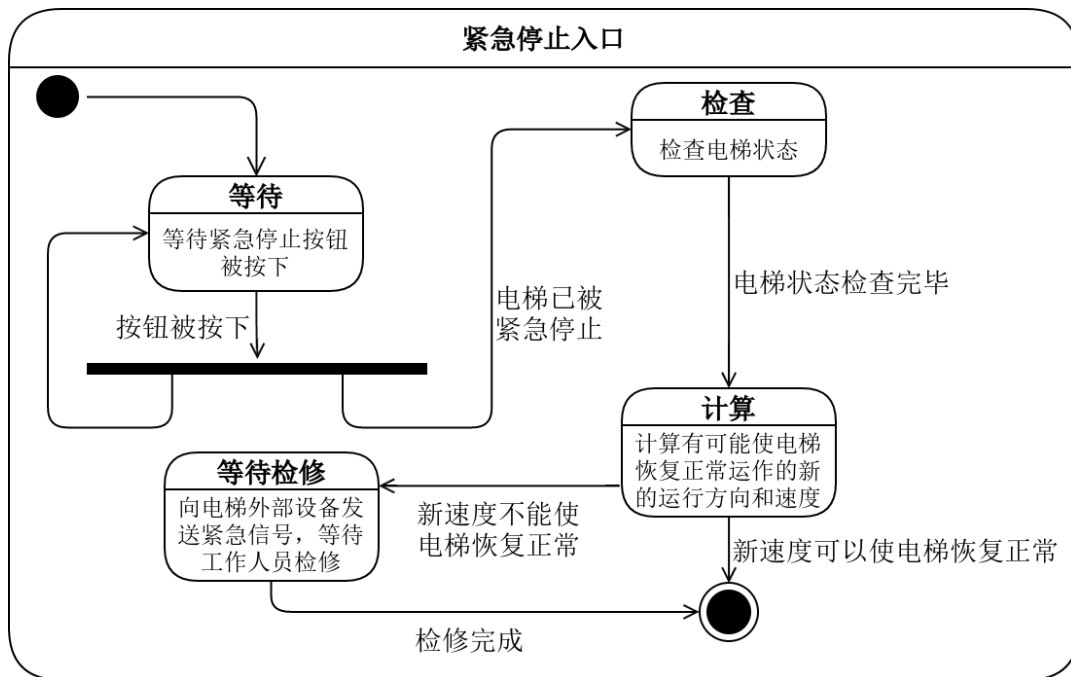


图 6.3.21: 处理紧急停止信号

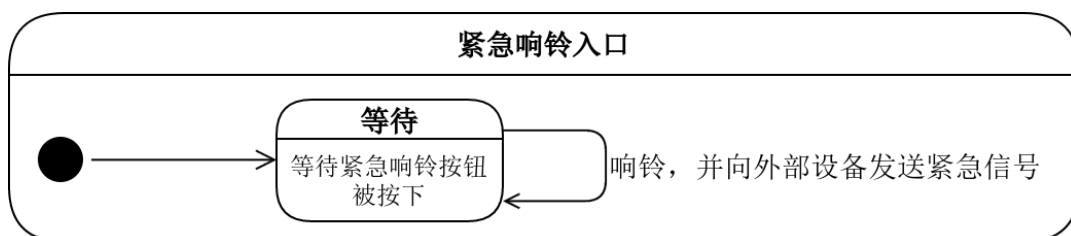


图 6.3.22: 处理紧急响铃信号

VI.3.3 软硬件平台设计

硬件方面，本电梯控制系统拟运行于 ARM® Cortex® -M9 系列之上，一方面，目前对于 ARM 架构的研究十分成熟，ARM® Cortex® 系列芯片的文档也十分齐全，另一方面，从性价比来看，在能够完全胜任对于整个电梯系统的调度的情况下，此系列芯片的价格较低，性价比高，可供选择的子型号也较多。

软件方面，本电梯控制系统拟在硬件平台上运行 μ CLinux 或 μ COS 等嵌入式操作系统，然后使用 Python 或 C++ 编写程序。使用嵌入式操作系统是出于维护以及管理的考虑，运行于操作系统比裸机运行程序更容易进行远程调试与管理，使用 C++ 或 Python 则是出于面向对象以的考虑，这两种语言都能够较为容易的编写基于面向对象的程序，调试工具种类齐全，十分方便。

VII 详细设计

VII.1 功能模块设计

章节VI已经对系统的功能有了一个比较系统化、结构化的介绍。图7.1.1将通过协作图的形式对于系统各个部件及其关系做出更详尽的解释说明。表VII-1-11是对图7.1.1中各个信号的详细说明。

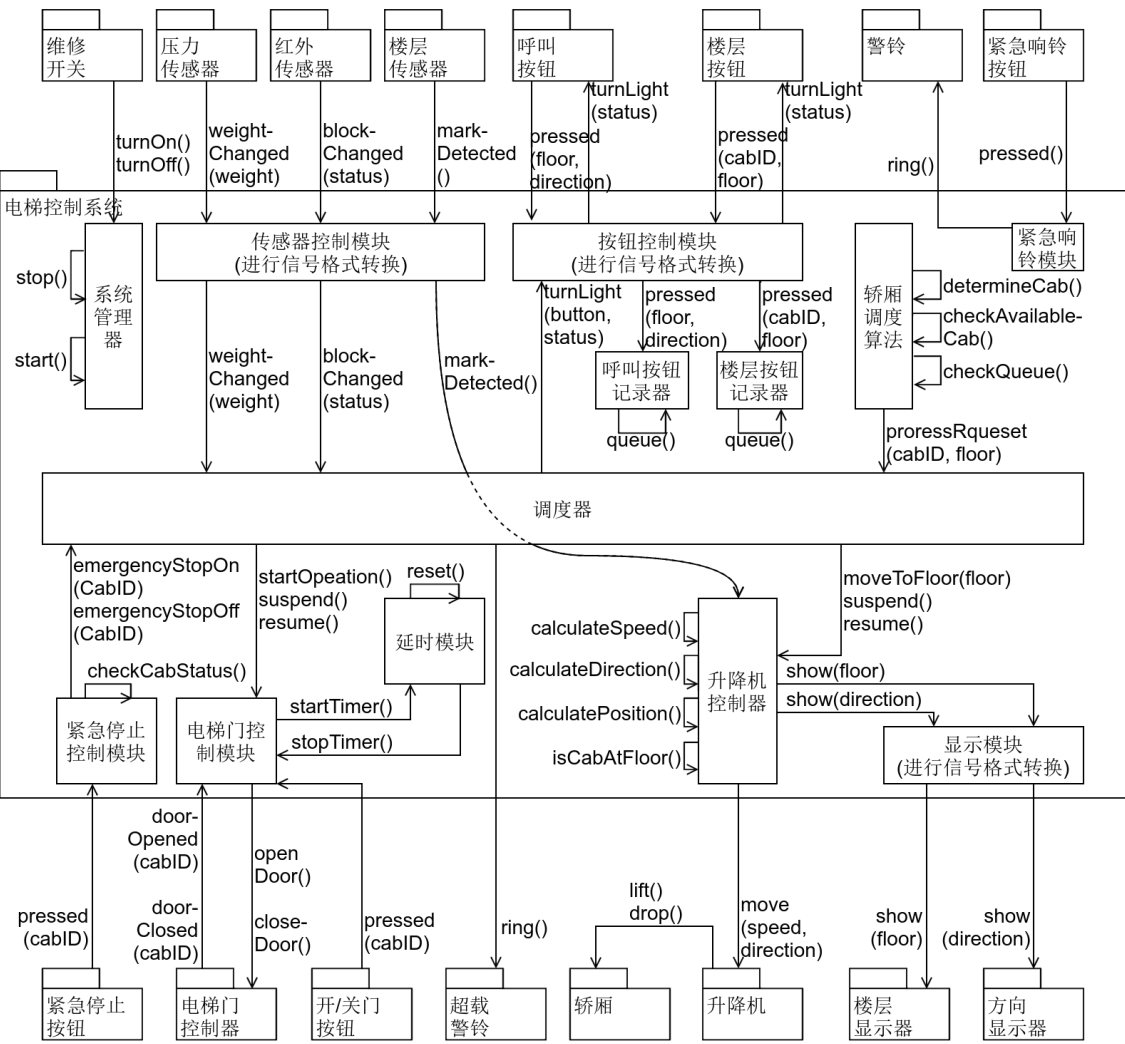


图 7.1.1: 电梯控制系统协作图

表 VII-1-11: 电梯控制系统协作图说明

模块	接收信号	信号发送者	信号说明
系统管理器	turnOn()	维修开关	发送启动电梯系统信号
	turnOff()	维修开关	发送停止电梯系统信号
	start()	自身	启动电梯系统
	stop()	自身	停止电梯系统

传感器控制模块	weightChanged(weight)	压力传感器	压力发生变化, 传入压力数值
	blockChanged(weight)	红外传感器	阻挡信号发生变化, 传入阻挡状态
	markDetected()	位置传感器	探测到位置标志
按钮控制模块	pressed(floor, direction)	呼叫按钮	呼叫按钮被按下, 传入楼层以及方向
	pressed(cabID, direction)	楼层按钮	楼层按钮被按下, 传入轿厢编号即楼层
	turnLight(button, status)	调度器	指定按钮按钮亮灯/灭灯
	turnLight(status)	按钮控制模块 → 楼层按钮	亮灯/灭灯
	turnLight(status)	按钮控制模块 → 呼叫按钮	亮灯/灭灯
呼叫按钮记录器	pressed(floor, direction)	按钮控制模块	呼叫按钮被按下, 传入处理过的信号
	queue()	自身	将信号加入队列
楼层按钮记录器	pressed(floor, direction)	按钮控制模块	楼层按钮被按下, 传入处理过的信号
	queue()	自身	将信号加入队列
轿厢调度算法	determineCab()	自身	决定用于处理信号的最优轿厢
	checkAvailableCab()	自身	检查可用的轿厢
	checkQueue()	自身	检查队列中的信号
紧急响铃模块	pressed()	紧急响铃按钮	紧急响铃按钮被按下
	ring()	紧急响铃模块 → 警铃	响铃
调度器	weightChaged(weight)	传感器控制模块	处理过的压力数值
	blockChanged(status)	传感器控制模块	处理过的阻挡信号
	processRequest(cabID, floor)	轿厢调度算法	选取标号为 cabID 的轿厢处理信号, 目的地是 floor 楼
	emergencyStopOn(cabID)	紧急停止控制模块	标号为 cabID 的轿厢执行紧急停止

	emergencyStopOff(cabID)	紧急停止控制模块	标号为 cabID 的轿厢恢复正常运行
	ring(cabID)	调度器 → 超载警铃	超载警铃响铃
紧急停止控制模块	checkCabStatus()	自身	检查轿厢的运行状态
	pressed(cabID)	紧急停止按钮	紧急停止按钮被按下
电梯门控制模块	startOpeation()	调度器	开始实行开门-关门预设程序
	suspend()	调度器	暂停开-关门程序并打开电梯门
	resume()	调度器	继续执行开-关门预设程序
	stopTimer()	延时模块	接收到超时信号
	doorOpened()	电梯门控制器	电梯门完全打开信号
	doorClosed()	电梯门控制器	电梯门完全关闭信号
	openDoor()	电梯门控制模块 → 电梯门控制器	开始开门过程
	closeDoor()	电梯门控制模块 → 电梯门控制器	开始关门过程
	pressed(cabID)	开/关门按钮	开/关门按钮被按下
延时模块	reset()	自身	重置定时器
	startTimer()	电梯门控制模块	开始倒计时
升降机控制器	calculateSpeed()	自身	计算升降机将要运行的速度
	calculateDirection()	自身	计算升降机将要运行的运行方向
	calculatePosition()	自身	计算升降机当前位置
	isCabAtFloor()	自身	判断轿厢是否到达特定楼层
	moveToFloor(floor)	调度器	移动轿厢到 floor 楼
	suspend()	调度器	当前升降机控制器暂停运行
	resume()	调度器	当前升降机控制器恢复运行
	move(speed, direction)	升降机控制器 → 升降机	控制升降机以特定的方向和速度移动
	markDetected()	传感器模块	探测到位置标志 (信号已转换)
显示模块	show(floor)	升降机控制器	显示指定楼层

show(direction)	升降机控制器	显示指定方向
show(floor)	显示模块 → 楼层显示器	显示指定楼层（信号格式已转换）
show(direction)	显示模块 → 楼层显示器	显示指定方向（信号格式已转换）

VII.1.1 功能说明

为了对整个系统功能有更为深入的理解，做出如图7.1.2的带有实体关系的系统协作图，此图是对电梯控制系统内部结构及其关系以数据为中心的更深入的解释。

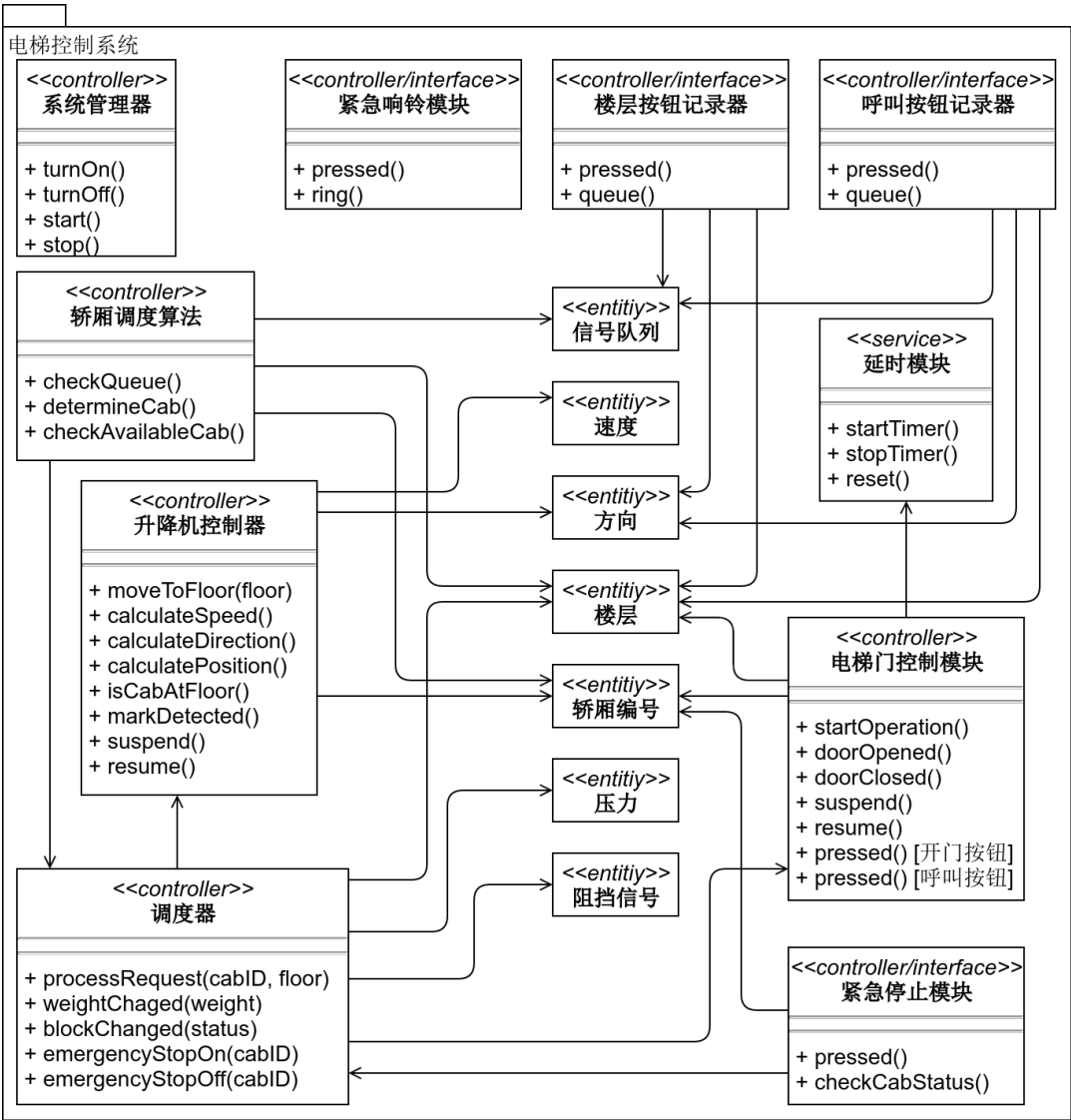


图 7.1.2: 协作图-以数据为中心

VII.1.2 功能协作图

由于不同的用例调用的功能模块不尽相同，因此有必要给出不同在不同的用例情况下，系统的协作关系，在测试时，也可以以这些协作图作为依据，对于各个模块的功能以及各个模块之间的协作进行有效的测试。

用例-呼叫按钮被按下 当呼叫按钮被按下后，除了呼叫信号被处理外，还有一系列的功能会被触发，如进行轿厢的调度，调度完成以后电梯门的控制等等，因此参与协作的组件较多。图7.1.3展示了呼叫按钮被按下后进行工作的模块以及他们之间需要进行传递的信号。

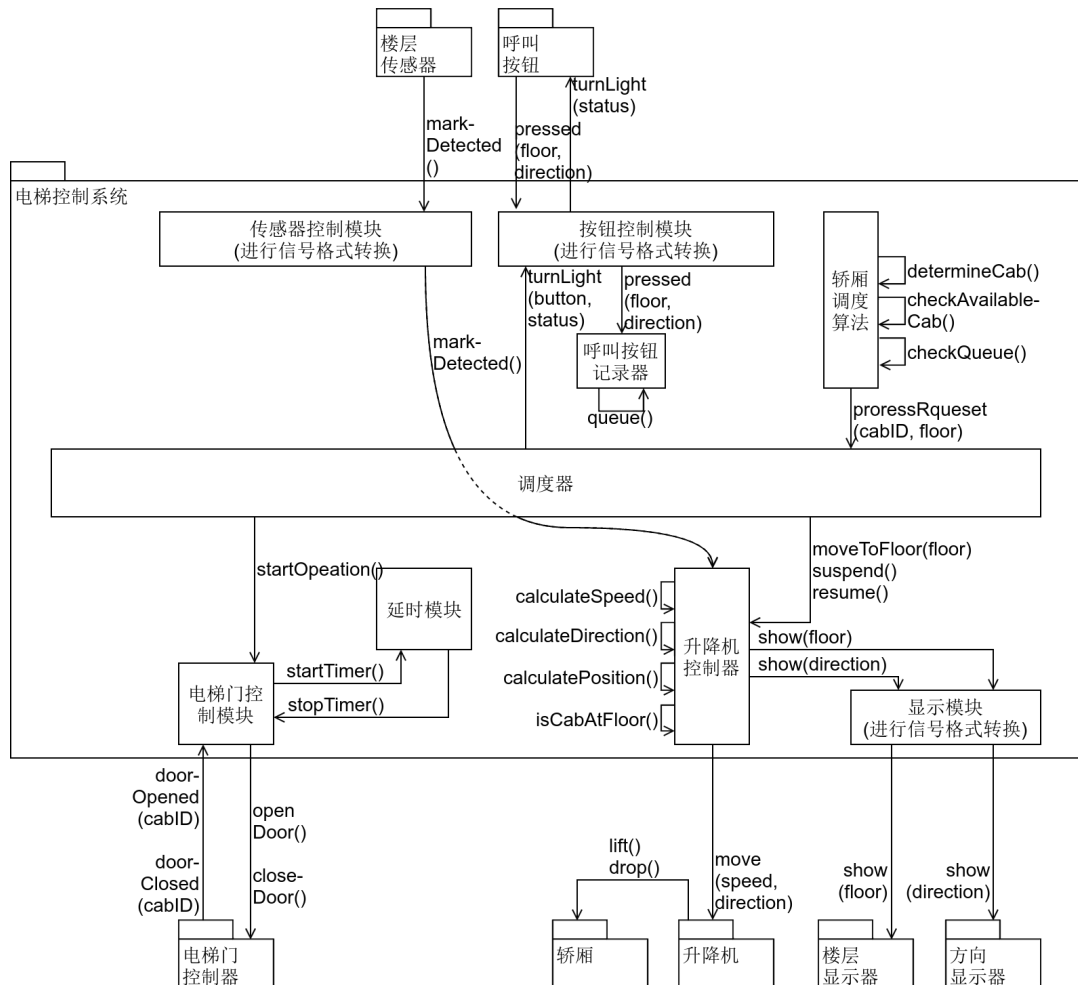


图 7.1.3: 协作图-呼叫按钮被按下

用例—楼层按钮被按下 与呼叫按钮被按下相似，轿厢的调图，以及门的控制会被触发。不同的是这一信号是在轿厢内发出的，不同的轿厢有轿厢编号的区分，因此模块间传送的信号有所不同。图7.1.4展示了楼层按钮被按下后的模块关系图。

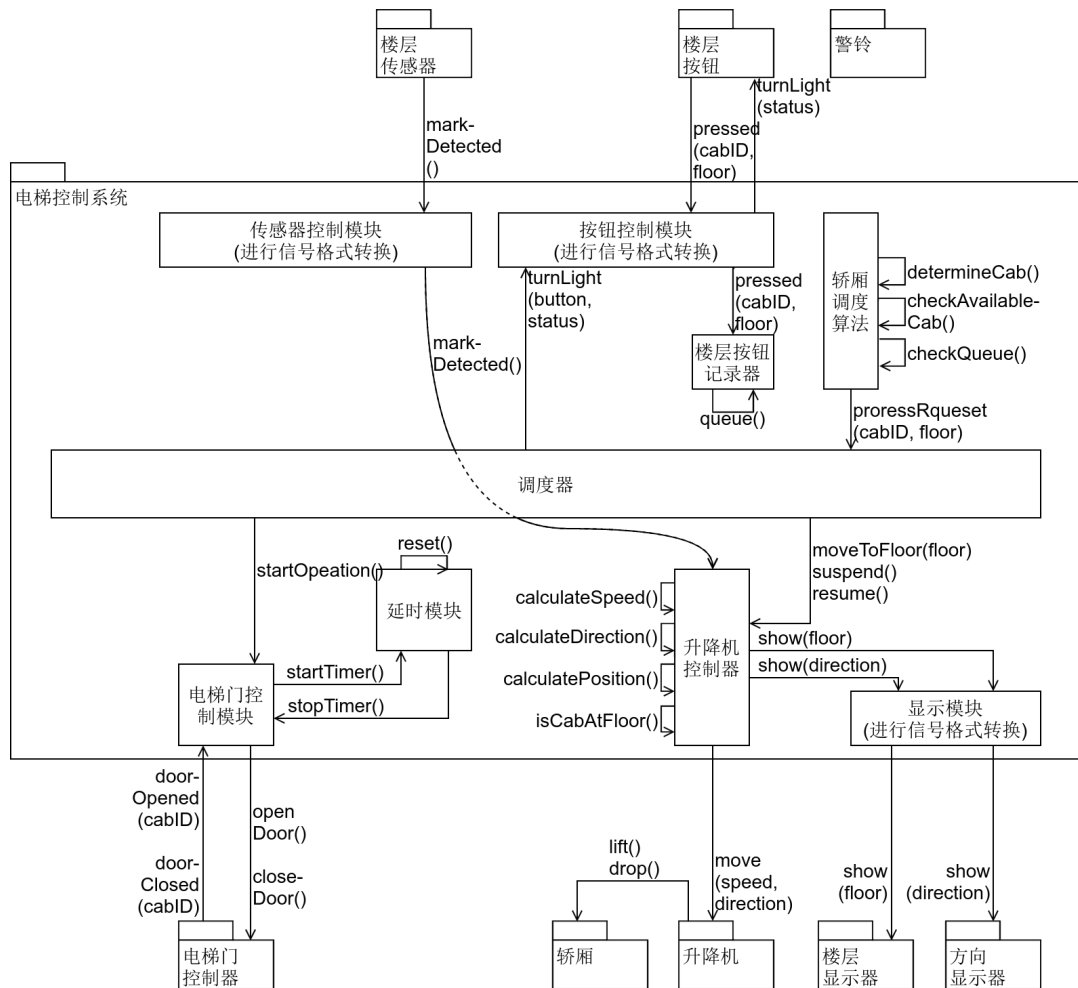


图 7.1.4: 协作图-楼层按钮被按下

用例-开门按钮被按下 当开门按钮被按下时，只有电梯门控制器参与工作，调度器并不参与。调度器等待电梯门完全关闭是通过检测信号队列中的信号状态进行的，这个过程中没有信号交换，因此调度器不参与工作。

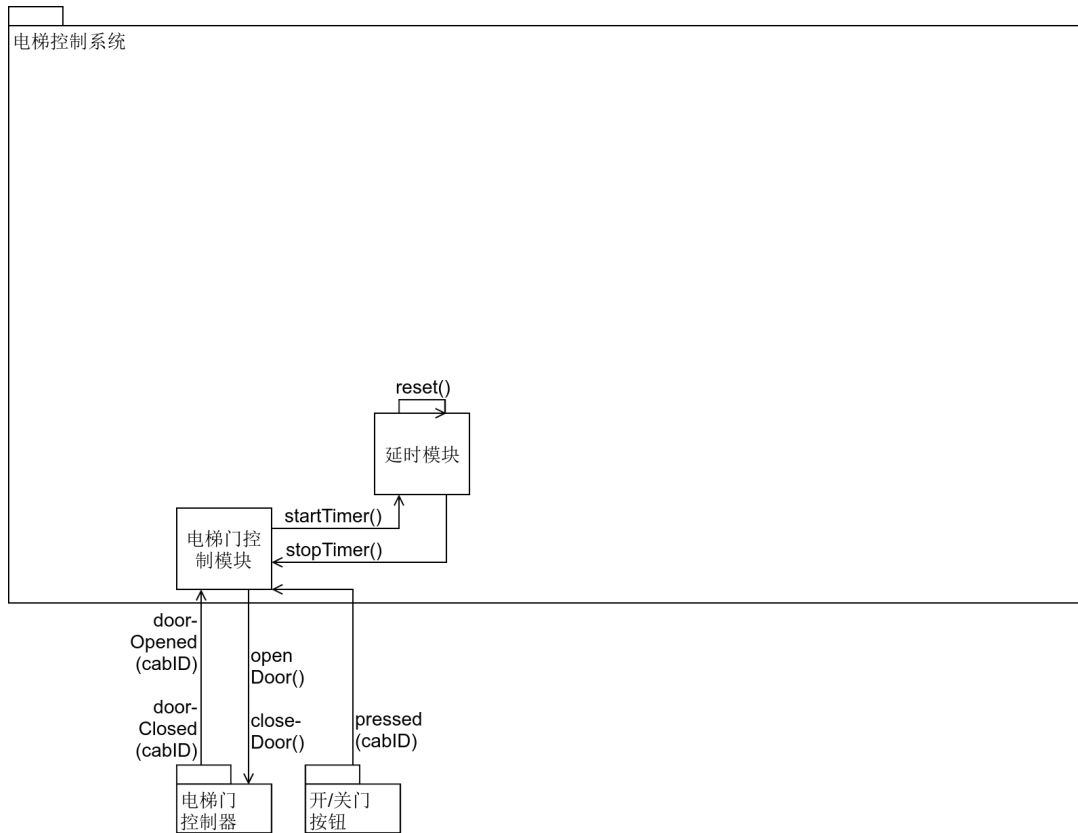


图 7.1.5: 协作图-开门按钮被按下

用例-紧急停止按钮被按下 当紧急停止按钮被按按下时，紧急停止控制模块会给调度器发送信号，调度器会发信号给电梯门控制器以及轿厢控制器使之进入暂停状态。图7.1.6展示了紧急停止按钮被按后系统的协作图，此图包括了紧急停止按钮被按下到系统恢复正常运行过程中的模块协作关系。

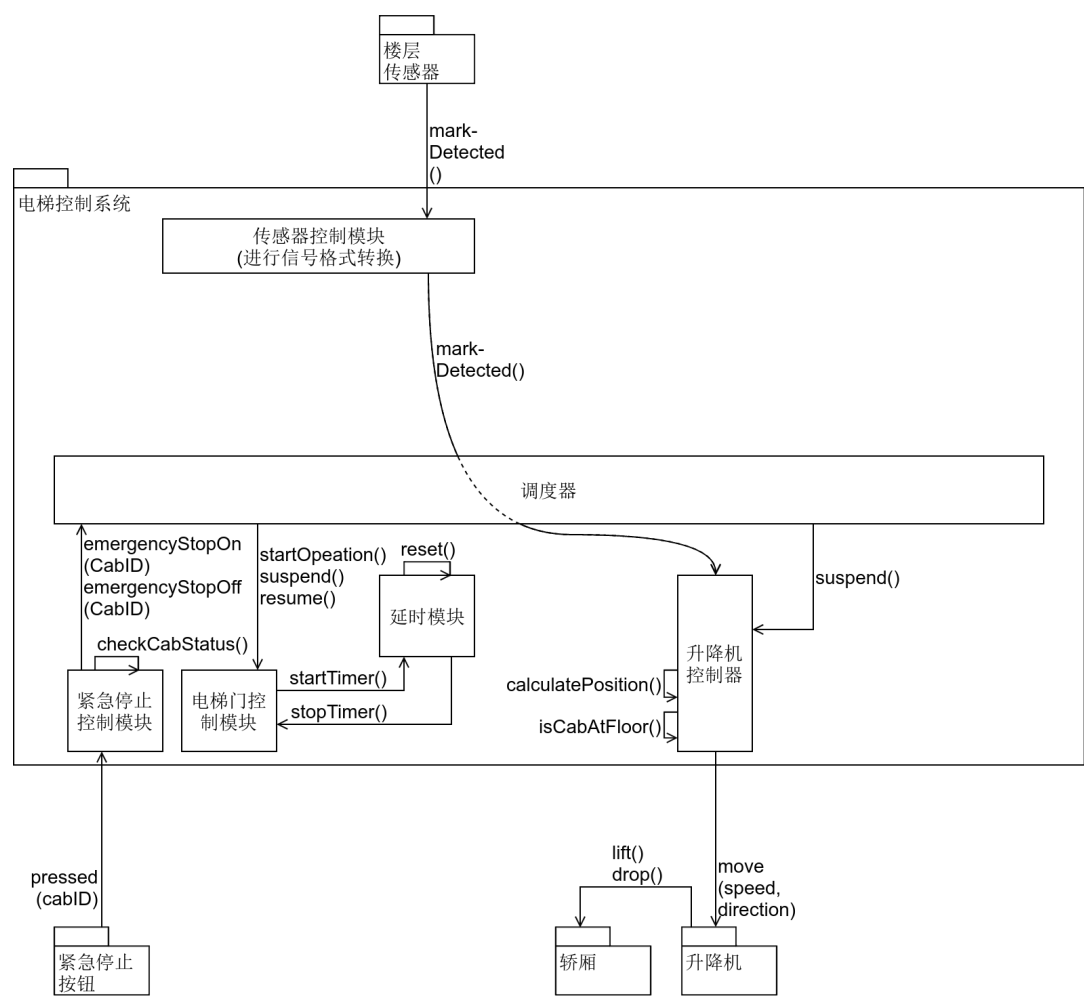


图 7.1.6: 协作图-紧急停止按钮被按下

用例-压力传感器超载 在压力传感器检测到超载时一定是电梯停靠在特定楼层时，因为电梯运行时电梯内的质量不可能增加。因此此时升降机控制器不参与协作，仅有传感器模块、警铃以及电梯门控制模块参与工作。图7.1.7展示了压力传感器检测到超载后的系统协作图。

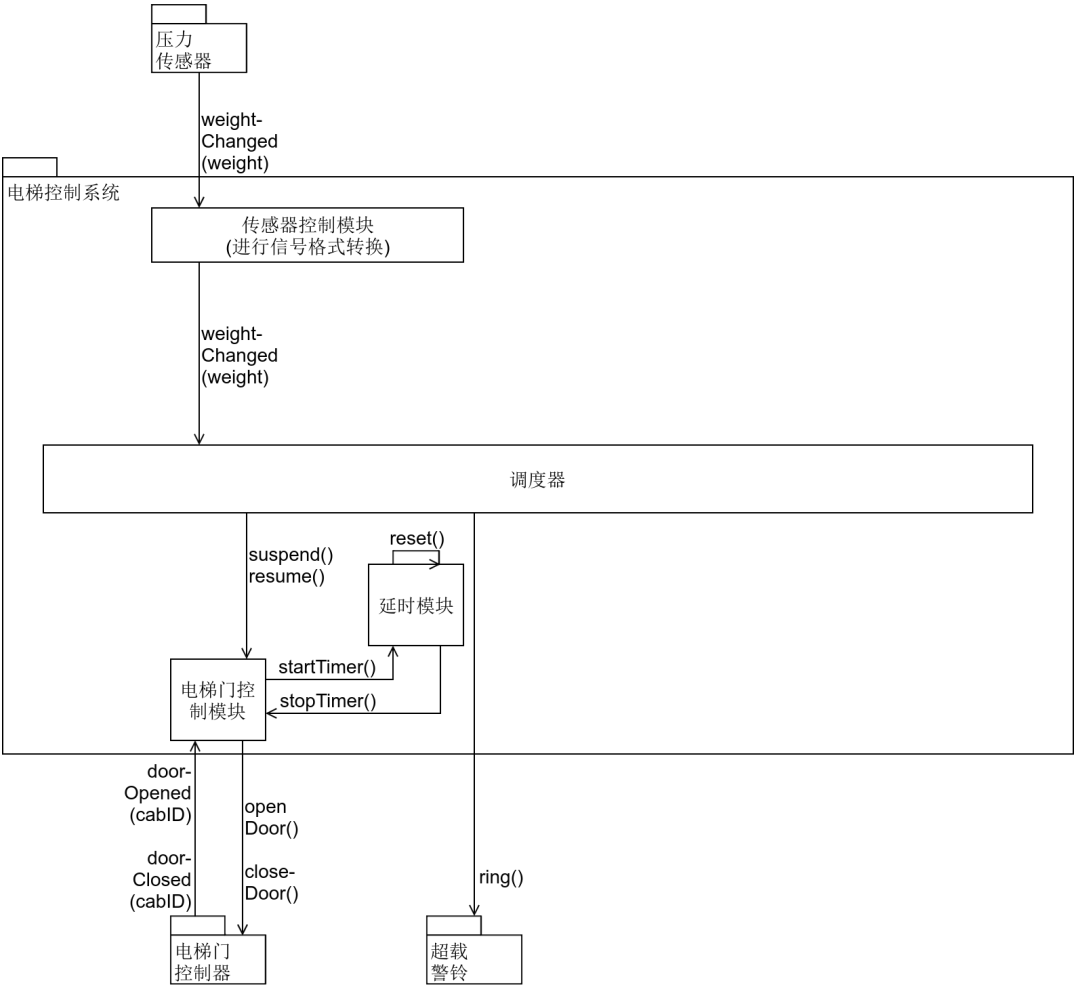


图 7.1.7: 协作图-压力传感器超载

用例—红外传感器被阻挡 与压力传感器类似，红外传感器被阻挡只可能是电梯停靠在特定楼层，并且在电梯门未完全关闭的情况下发生，因此升降机控制器不参与协作。不同的是，此时超载警铃不再参与工作。图7.1.8展示了红外传感器被阻挡时的系统协作图。

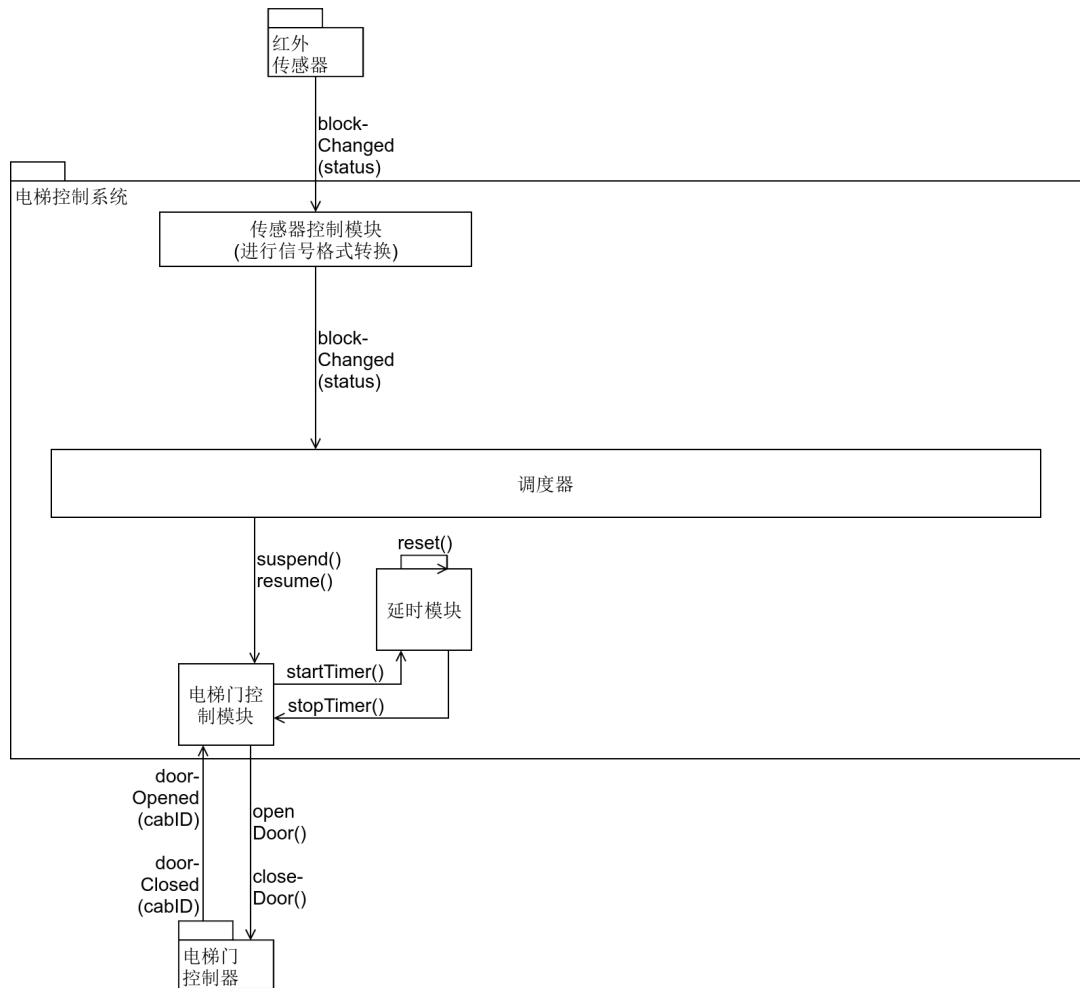


图 7.1.8: 协作图—红外传感器被阻挡

用例-维修开关状态改变、紧急响铃按钮被按下 在这两个用例中，参与工作的组件均为独立工作，不与其他组件有任何信号交换，因此系统协作图也较为简单。图7.1.9展示了在这两种情况下的系统协作图。

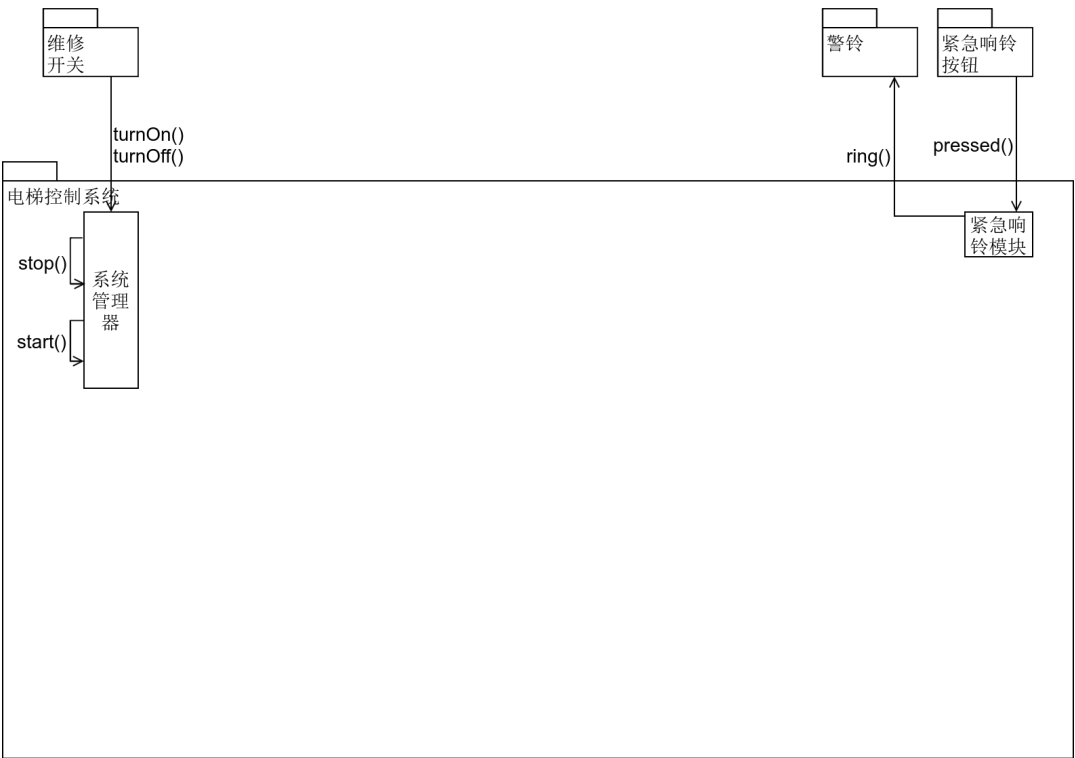


图 7.1.9: 协作图-维修开关以及紧急响铃按钮

VII.1.3 功能状态图

对于图7.1.1中所描述的电梯控制系统中的各个模块，作出其在处理信号时的功能状态图，同时，这也是对于概要设计中章节VI.3.2的状态转换图的详细解释以及扩充。

系统整体状态图 图7.1.10为控制整个电梯控制系统开关的状态图，在电梯控制系统中处于最高一级。

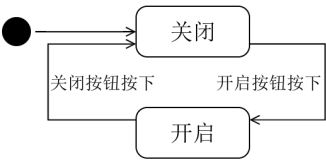


图 7.1.10: 状态图-系统整体

呼叫/楼层按钮记录器状态图 图7.1.11为呼叫按钮记录器和楼层按钮记录器的状态图。它们展示了两个记录器的总体运行状态。

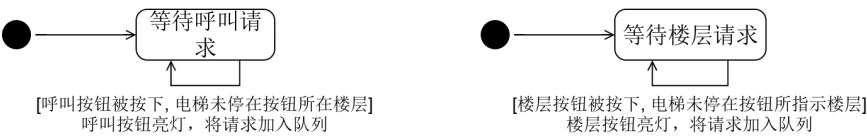


图 7.1.11: 状态图-呼叫/楼层按钮记录器

轿厢调度算法状态图 图7.1.12是轿厢调度算法模块的状态转换图，只要接口格式一致，具体的轿厢调度算法可以随时被替换。

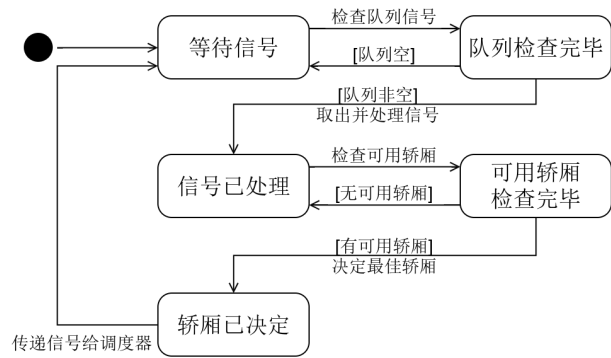


图 7.1.12: 状态图-轿厢调度算法

调度器状态图 图7.1.13展示了调度器的状态转换图，调度器为整个电梯控制系统的核心部件，需要与大部分系统模块进行交流，因此起状态图较为复杂。

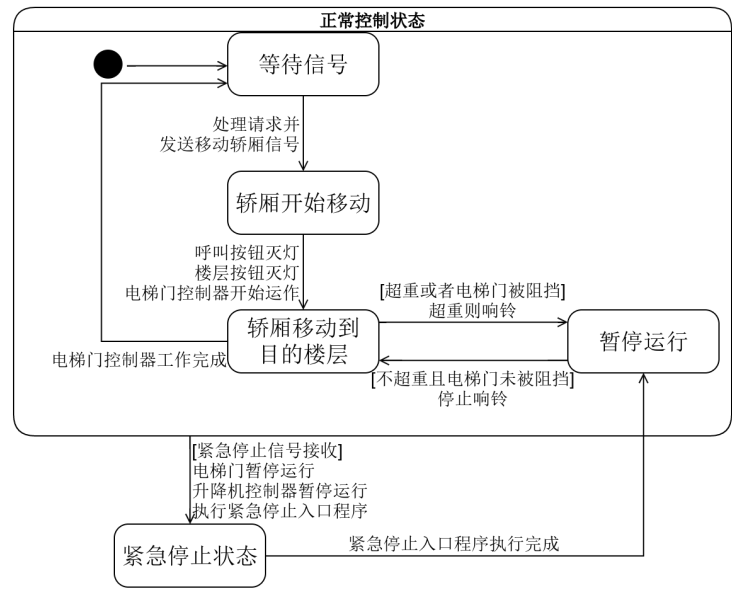


图 7.1.13: 状态图-调度器

升降机控制器状态图 图7.1.14展示了升降机控制器的状态转化关系，其主要有两个状态：运行状态和暂停状态，两个主要状态之间由调度器发来的暂停信号和恢复信号控制。

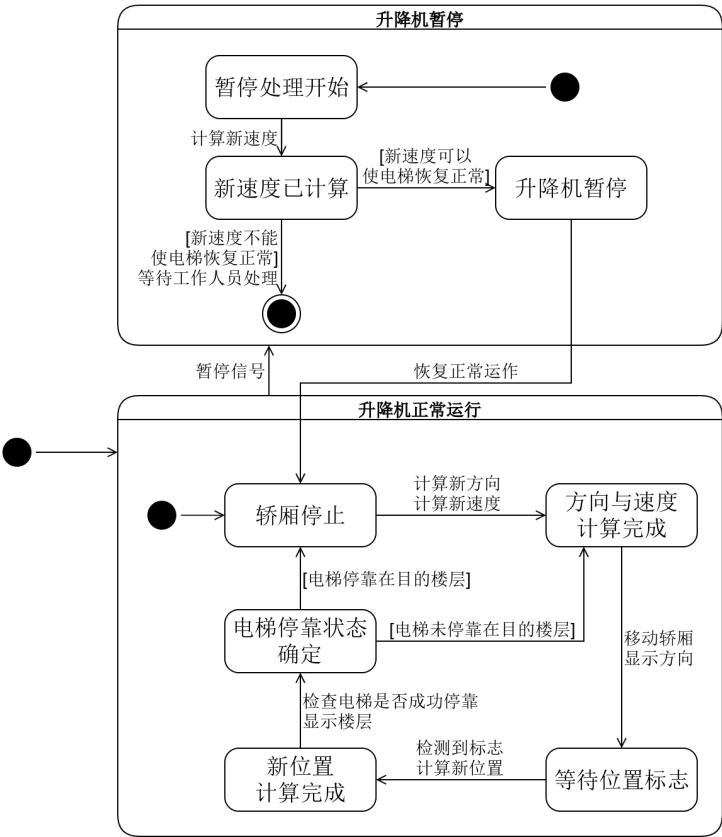


图 7.1.14: 状态图-升降机控制器

电梯门控制器状态图 图7.1.15为电梯门控制器的状态转换图，电梯门状态受到多个模块的控制，同时也存在暂停状态，使其状态转换图较为复杂。

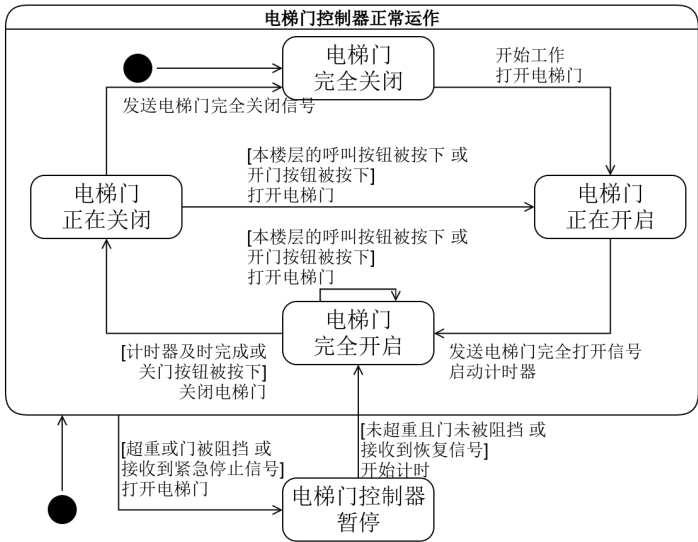


图 7.1.15: 状态图-电梯门控制器

计时器、紧急停止模块以及紧急响铃按钮 这三个模块中、计时器是为电梯门控制器提供服务的，紧急停止模块作为调度器与紧急停止按钮之间的中介为调度器提供服务，而紧急响铃按钮独立与调度器运行，因此它们的状态图均较为简单。

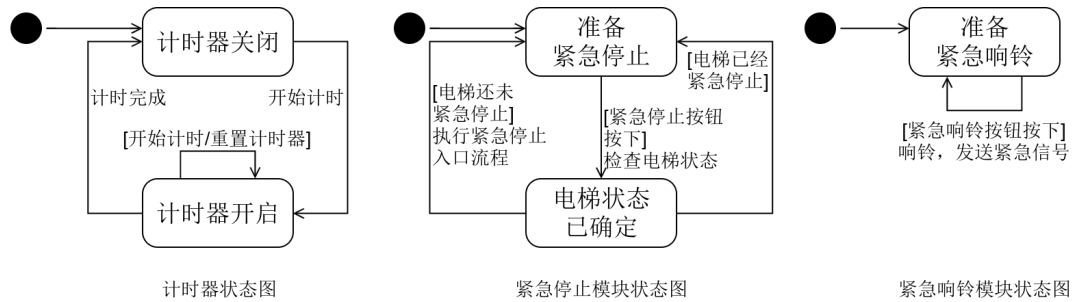


图 7.1.16: 状态图-计时器、紧急停止模块以及紧急响铃模块

VII.2 算法设计

本节针对电梯调度系统中的轿厢调度算法模块进行详细的说明。

为了使电梯轿厢的运行能够尽可能减少用户的等待时间，同时能够尽可能减少电梯的能耗，本电梯控制系统拟采用匈牙利算法。虽然对于电梯调度系统而言已经有神经网络算法和遗传算法等可以使用，但是这些算法难以实现同时对于多个电梯的调度，而匈牙利算法能够在很大情况下解决电梯的群控问题，具有较高的实用性

VII.2.1 算法简介

匈牙利算法是一种在多项式时间内求解任务分配问题的组合优化算法，并推动了后来的原始对偶方法。美国数学家哈罗德·库恩于 1955 年提出该算法。此算法之所以被称作匈牙利算法，是因为算法很大一部分是基于以前匈牙利数学家 Dénes König 和 Jenő Egerváry 的工作之上创建起来的。¹

以矩阵 C 为例：

$$C = \begin{bmatrix} 2 & 11 & 3 & 8 \\ 5 & 6 & 1 & 9 \\ 7 & 4 & 10 & 2 \\ 8 & 10 & 7 & 5 \end{bmatrix}$$

计算步骤如下：

1. 先让矩阵 C 中每行元素减去该行元素中的最小值，让每列元素减去该列元素中的最小值，这样每行必然会产生至少一个零元素。

$$\begin{bmatrix} 2 & 11 & 3 & 8 \\ 5 & 6 & 1 & 9 \\ 7 & 4 & 10 & 2 \\ 8 & 10 & 7 & 5 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 9 & 1 & 6 \\ 4 & 5 & 0 & 8 \\ 5 & 2 & 8 & 0 \\ 3 & 5 & 2 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 7 & 1 & 6 \\ 4 & 3 & 0 & 8 \\ 5 & 0 & 8 & 0 \\ 3 & 3 & 2 & 0 \end{bmatrix}$$

¹匈牙利算法. (2017). Zh.wikipedia.org. Retrieved 16 April 2017, from <https://zh.wikipedia.org/wiki/匈牙利算法>

2. 找出仅有一个“0”元素的行,并划去与该“0”同列的其他“0”元素,然后找出仅有一个“0”元素的列,并划去与该“0”同行的其他“0”元素。

$$\begin{bmatrix} 0 & 7 & 1 & 6 \\ 4 & 3 & 0 & 8 \\ 5 & 0 & 8 & 0 \\ 3 & 3 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 7 & 1 & 6 \\ 4 & 3 & 0 & 8 \\ 5 & 0 & 8 & \emptyset \\ 3 & 3 & 2 & 0 \end{bmatrix}$$

3. 观察矩阵中的零元素的个数是否等于矩阵的阶数,若两者相等则算法结束,否则需要对矩阵进行变化。直到矩阵中零元的个数与矩阵的阶数相等则算法结束。

$$\begin{bmatrix} 0 & 7 & 1 & 6 \\ 4 & 3 & 0 & 8 \\ 5 & 0 & 8 & \emptyset \\ 3 & 3 & 2 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

VII.2.2 算法应用

综合考虑电梯控制系统的优化,考虑乘客乘梯时间、等待时间、电梯内拥挤程度(使用压力传感器获取)和电梯能耗四个指标,将匈牙利算法应用于轿厢调度上,将四个指标加权后可以得到:

$$F = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 \quad (1)$$

式1中 F 为总评价函数, f_1, f_2, f_3, f_4 分别为候梯时间评价函数、乘梯时间评价函数、能量消耗评价函数数和梯内拥挤度小评价函数。 w_1, w_2, w_3, w_4 为各评价函数的权重系数且 $w_1 + w_2 + w_3 + w_4 = 1$ 。修改 w_x 的值可以修改各个评价函数的权重,从而影响整个系统的调度行为。

在每次需要决定使用哪个轿厢响应信号时,先取得轿厢当前的状态,然后根据当先的状态和信号队列中的信号计算出电梯的评价矩阵,然后利用匈牙利算法解出需要调用的轿厢。电梯的评价矩阵为

$$C = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ F_{21} & F_{22} & \cdots & F_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ F_{m1} & F_{m2} & \cdots & F_{mn} \end{bmatrix}$$

其中 F_{ij} 代表第 i 个电梯在处理信号队列中第 j 个信号时的评价指标(由式1计算得出)。矩阵 C 由匈牙利算法计算得出最优分配方案,然后进行轿厢的调度。

由于匈牙利算法要求待解的矩阵为方阵,因此需要对非方阵进行处理。对于 $m > n$ 的情况,则假设信号槽末尾还有 $m - n$ 个虚拟信号,且这些虚拟信号的评价值均为 0,这样可以保证算法的结果不受虚拟信号的影响。而对于 $m < n$ 的情况,则假设有 $n - m$ 个虚拟轿厢,这些轿厢的评价值为 0。在最终的结果中,对于虚拟部分的值不作处理(丢弃)即可。

VIII 界面设计

由于电梯控制系统不是完全运行于计算机上的计算机软件，而是运行于嵌入式设备上对硬件进行控制的软件，因此其没有严格的界面。但对于电梯控制系统模块的外部接口而言，需要有一个统一的标准以便于进行信号的传递，符合这一标准的终端型硬件则可以称作电梯控制系统软件的界面。对于此电梯控制系统而言，至少需要：

- 在每建筑的每楼层都有每个轿厢的位置和行驶方向的显示面板
- 每层楼都有至少一个呼叫按钮面板，其上有上行呼叫按钮和下行呼叫按钮。
- 一楼的呼叫按钮面板只有上行呼叫按钮，顶楼的呼叫按钮面板只有下行呼叫按钮。
- 轿厢内部至少需要有一个按钮面板，其上有所有楼层的楼层按钮，紧急停止按钮，紧急响铃按钮。
- 轿厢内部至少需要一个显示面板，显示当前轿厢的位置和行驶方向。
- 轿厢内部至少需要有一个警铃来响应响铃信号。

图8.0.1和图8.0.2是符合标准的电梯内部界面和电梯外部界面的样例。

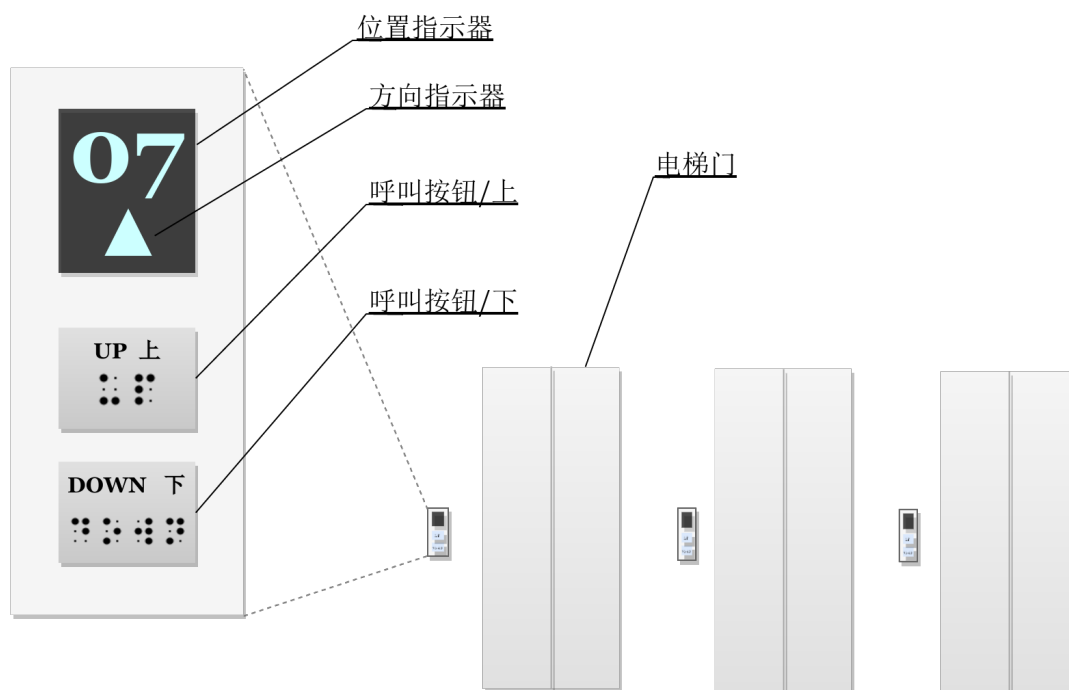


图 8.0.1: 界面-电梯外部

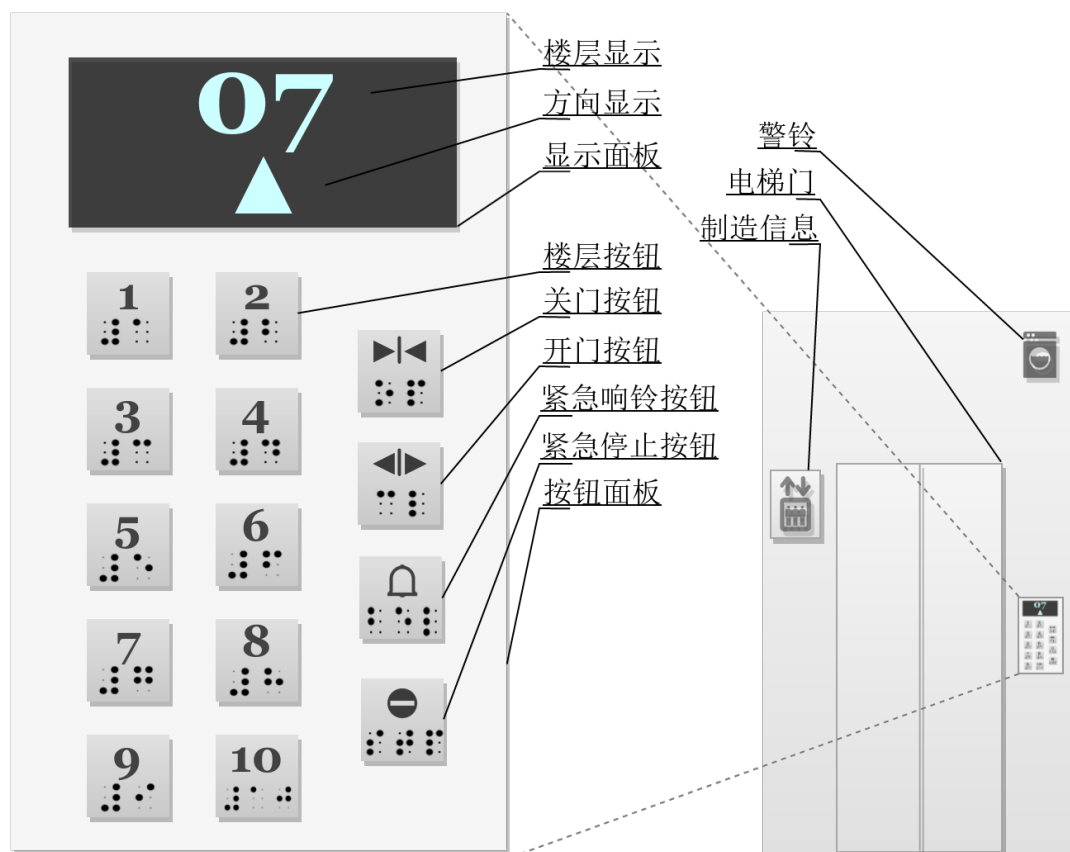


图 8.0.2: 界面-电梯内部

IX 系统测试

IX.1 测试目的

这一测试的目的是为了找出应用程序中的缺陷，从而提高电梯控制系统的质量。对于本控制系统不可能对于每一个细节均进行测试，但需要在尽可能广泛的范围内进行测试，其中，测试并对程序做出修改后至少需要保证程序可用，且具有较高的可移植性。

IX.2 测试方法

由于本电梯测试系统包含有多个模块，因此系统的测试应该先从小模块开始，在代码的编写过程中对于代码中的每一个模块进行单元测试。在开发完成后，依据详细设计中的状态图与合作图，对于每一个模块进行集成测试，并采用自底向上的测试方法。最后，使用黑盒测试方法，让非开发人员以及用户进行确认测试。只有在这三种测试完成后，才能够确定电梯控制系统能够投放使用。对于测试用例的设计需要考虑边界值的选取以及等价类的划分。

IX.2.1 单元测试

单元测试主要是测试代码是否存在语法错误、逻辑错误等。其主要目的是验证进行代码编写时是否能够正确执行而产出符合预期值的结果，在本控制软件的代码编写的过程中对于电梯控制系统实现中的所有类方法进行的单元测试应在软件编写的过程中进行。

IX.2.2 集成测试

IX.3 确认测试

对于确认测试而言，其主要目的是测试电梯控制每一个功能的完整性。因此，根据表IX-3-12所示的电梯系统的功能对其进行测试。对于每一个功能，根据起功能描述在限制条件下对其进行测试，测试结果应与“功能描述/语气结果中描述的结果相同”

表 IX-3-12: 功能表

功能名称	功能描述/预期结果	细节与限制
轿厢响应呼叫	升降机控制器控制电梯轿厢完全到达呼叫发出的楼层，呼叫由呼叫按钮被按下完成。	根据轿厢调度算法，综合评分最高的轿厢应该被调度。
轿厢响应楼层请求	当轿厢内部的一个楼层按钮被按下后，轿厢需要最终移动到指定的楼层。	轿厢可以先在其他楼层停靠。
关门后轿厢立即移动	在电梯门关闭后，轿厢需要尽快地进行移动，以确保运输需求	轿厢移动的目的地是楼层请求指示的楼层或者呼叫请求发出的楼层。

开门按钮打开电梯门	当开门按钮被按下时, 电梯门应该被打开。	电梯必须首先在某个楼层停靠。如果电梯门没有完全关闭, 那么开门按钮的按下会再一次将电梯门打开。
开门按钮被按住	当开门按钮被按下时, 电梯门会保持最大打开状态不会关闭。	电梯必须首先在某个楼层停靠。
到达呼叫楼层时开门	当电梯到达呼叫信号被发出的楼层时, 电梯门应该打开, 便于乘客进出。	无
到达请求楼层时开门	当电梯内的楼层请求按钮被按下, 并且电梯到达响应楼层时, 电梯应该开门, 便于乘客进出。	无
呼叫按钮打开电梯门	当电梯按钮停靠在呼叫按钮所在的楼层, 且电梯门未完全关闭时, 按下此楼层的呼叫按钮会重新打开电梯门。	此时的呼叫按钮和开门按钮的功能相同。
呼叫按钮被按住	当电梯按钮停靠在呼叫按钮所在的楼层, 且电梯门未完全关闭时, 按住此楼层的呼叫按钮会使电梯门保持最大打开状态不会关闭。	此时的呼叫按钮和开门按钮的功能相同。
电梯门自动关闭	当电梯门打开达到一定时间且无信号接收, 电梯门自动关闭。	在此过程中若开关门按钮, 呼叫按钮被按下, 压力传感器、红外传感器发送信号则停止自动关闭过程, 打开电梯门并重新计时。
紧急停止按钮停止轿厢	当紧急停止按钮被按下时, 轿厢应该立即停止。	紧急停止按钮若已被按下过则再次按下无作用。
紧急响铃按钮触发响铃	当紧急响铃按钮被按下时, 电梯应该立即响铃并向外界发送信号。	无
超载时响铃响铃	当压力传感器检测到压力超过限制时, 电梯需要响铃。	无
超载时暂停工作	当压力传感器检测到压力超过限制时, 升降机控制器以及电梯门控制器应该暂停工作, 直到压力传感器发出压力正常信号。	无
呼叫按钮亮灯	当呼叫按钮被按下时, 呼叫按钮上的灯应亮起。	呼叫按钮的灯必须实现处于灭状态。

呼叫按钮灭灯	当轿厢到达呼叫按钮所在楼层时，呼叫按钮上的灯应灭掉。	呼叫按钮的灯必须实现处于亮状态。
楼层按钮亮灯	当楼层按钮被按下时，楼层按钮上的灯应亮起。	楼层按钮的灯必须实现处于灭状态。
楼层按钮灭灯	当轿厢到达楼层按钮所指示楼层时，楼层按钮上的灯应灭掉。	楼层按钮的灯必须实现处于亮状态。
方向显示	当电梯移动的方向或即将移动的方向发生改变时，方向指示器上的方向应该发生改变，指示新的方向。	当电梯不移动时，方向显示器不显示方向。
楼层显示	当电梯的楼层发生变化时，无论轿厢是否运动，楼层显示器的楼层都应该发生变化，指示新的楼层。	无
手工控制	当维修开关被打开时，电梯可以接受手工控制	维修开关必须事先处于关闭状态
停止手工控制	当维修开关被关闭时，电梯不接受手工控制	维修开关必须事先处于打开状态

除了功能性测试以外，测试人员还应对于非功能性部件进行测试，以保证整个电梯系统各个部件之间能够正常协作。非功能需求见表IX-3-13。

表 IX-3-13: 非功能需求

名称	功能描述/预期结果	细节与限制
电梯加速度	为了电梯内乘客的安全，在任何时刻电梯的加速度或者减速度大小不能超过 $3m^2/s$	无
轿厢门开启	为了电梯内乘客的安全，应该使用机械性结构保证在电梯有运行时电梯内部的门不能够被打开。	无
外部门开启	为了电梯外的人员的安全，在轿厢未在某一楼层停靠时，应该使用机械性结构保证电梯外部门（电梯井门）不能够被打开。	无
轿厢运行	当电梯门未完全关闭时，轿厢不能够开始运行。	无

轿厢暂停工作	当检测到电梯的关键组件发生故障时，轿厢应该停止运行。	无
响应时间	硬件应该保证电梯其他组件发出的信号必须能够在 300 毫秒内得到处理。	无
信号处理	电梯其他组件发出的信号必须被处理，无论控制器处于什么状态，不能出现信号丢失的情况。	控制器只负责对于接收到的信号进行处理，而不对在传输过程中发生的信号丢失负责。
轿厢停靠	当交响停靠在某一楼层时，必须保证电梯外部门与内部门的高度对齐，以保证乘客能够安全的出入。	只在电梯正常停靠时有效，不适用于紧急停止。

X 项目管理

X.1 软件规模度量

为了对本电梯控制系统软件进行信息与特性和软件复杂性评估,从而估算软件规模,本节将采用功能点技术对于软件规模进行度量。

X.1.1 计算未调整功能点数

对于本软件而言,其信息域特性如表X-1-14所示。

表 X-1-14: 电梯控制系统信息域特性

信息域特性	功能点数	复杂级别	特性系数
输入项数 (Inp)	5	中等	4
输出项数 (Out)	5	中等	5
查询数 (Inq)	0	简单	3
主文件数 (Maf)	1	简单	7
外部接口数 (Inf)	18	复杂	10

根据:

$$UFP = a_1 \times Inp + a_2 \times Out + a_3 \times Inq + a_4 \times Maf + a_5 \times Inf \quad (2)$$

可得:

$$UFP = 5 \times 4 + 5 \times 5 + 0 \times 3 + 1 \times 7 + 18 \times 10 = 232 \quad (3)$$

X.1.2 计算技术复杂因子

表X-1-15展示了软件的技术因素及其影响值。影响值分布在 0 5 之间,0 表示无影响,5 表示有很大影响,中间的值影响力均匀分布。

表 X-1-15: 技术因素

序号	F_i	技术因素
1	5	数据通信
2	0	分布式数据处理
3	3	性能标准
4	2	高负荷的硬件
5	4	高处理率
6	0	联机数据输入
7	1	终端用户效率
8	0	联机更新
9	1	复杂的计算
10	4	可重用性
11	5	安装方便
12	4	操作方便
13	4	可移植性
14	5	可维护性

根据:

$$\begin{cases} DI = \sum_{i=1}^{14} F_i \\ TCF = 0.65 + 0.01 \times DI \end{cases} \quad (4)$$

可得:

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i = 1.03 \quad (5)$$

又根据式3和式5可得:

$$FP = UFP \times TCF = 232 \quad (6)$$

又根据 kemerer 模型, 计算得:

$$E = 60.62 \times 7.728 \times 10^{-8} \times 232^3 = 58.5 \quad (7)$$

根据 COCOMO2 模型 $T = 2.5 \times 232^{0.35} = 16.82$ 计算, 正常情况下完成这一项目需要大约 17 个月。项目拟聘用 5 个人开发。

X.2 人员安排

对于本电梯系统的开发, 使用 5 个人进行开发。一方面, 人数太多会增加开发开发时的沟通成本, 反而可能增加开发时间。另一方面, 电梯控是系统的各个模块之间的通信较多, 开发人员过多容易出现代码接口混乱, 代码难以管理的情况, 开发人员较少则便于管理和交流。

在分组方面, 由于没有很多的开发人员, 因此不需要细分小组, 直接采用如图10.2.1的民主制现代程序员小组即可。其中技术组长负责保证整个程序功能的正确性以及开发时间。行政组长负责管理组内的非技术事物, 如与甲方进行交流以及考虑产品的法律问题。对于被管理的程序员而言, 他们主要向技术组长汇报工作的进展状况以及工作中遇到的难题。

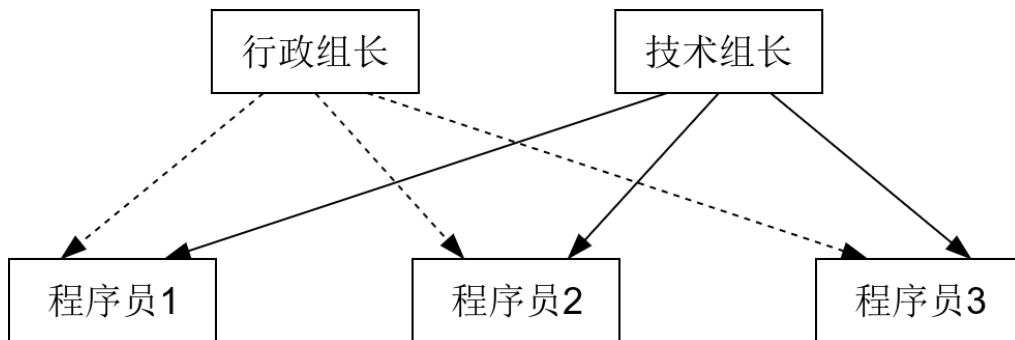


图 10.2.1: 小组组织结构

X.3 项目进度计划

鉴于电梯控制软件中调度器的核心位置, 它必须被最先开发, 且应首先将器接口设计完成, 以供其他模块调用。在此后的过程中, 由于大部分模块都与不同的外部设备打交道, 且模块间的信号交流格式以及函数都已确定, 因此可以同步开发。每一步开发完成后, 都可以将其被调度器控制或与调度器的代码加入工程中, 即对调度器代码的补充。对于轿厢调度算法模块, 可以留待最后开发, 因其不受其他任何模块的影响也不影响任何模块, 并且是系统中较为

重要和复杂的模块, 留待最后开发可以将其仔细研究、换用多种算法等, 提高开发的成功率以及软件的运行效率。总体的开发工程网络图见图10.3.1

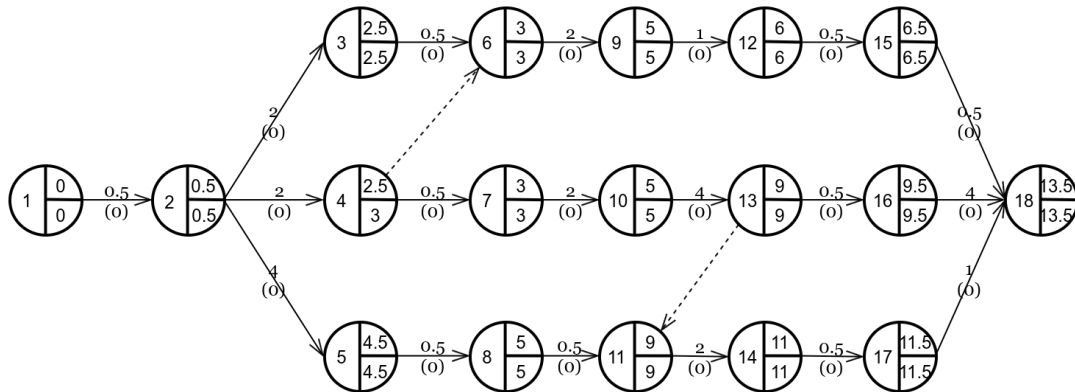


图 10.3.1: 工程网络图

对于图10.3.1中的编号做出如下解释:

表 X-3-16: 工程网络图过程

过程	开发部分
1→2	调度器接口
2→3	传感器控制模块
2→4	按钮控制模块
2→5	电梯门控制模块
3→6	调度器-传感器控制模块部分
4→7	调度器-按钮控制模块部分
5→8	调度器-电梯门控制模块部分
6→9	呼叫按钮记录器
7→10	楼层按钮记录器
8→11	延时模块
9→12	紧急停止模块
10→13	升降机控制器
11→14	显示模块
12→15	调度器-紧急停止模块部分
13→16	调度器-升降机控制器部分
14→17	系统管理器
15→18	控制器-轿厢调度算法部分
16→18	轿厢调度算法模块
17→18	紧急响铃模块

由于整个工程中的交叉交叉依赖关系较少, 且大部分时间都在并行开发, 因此工程网络图中的 3 条主要路径均为关键路径, 要减少开发时间, 就需要从对结束时间影响最大的路径减少开发时间。

对于本软件的预想开发过程, 使用图10.3.2所示 Gantt 图表示。

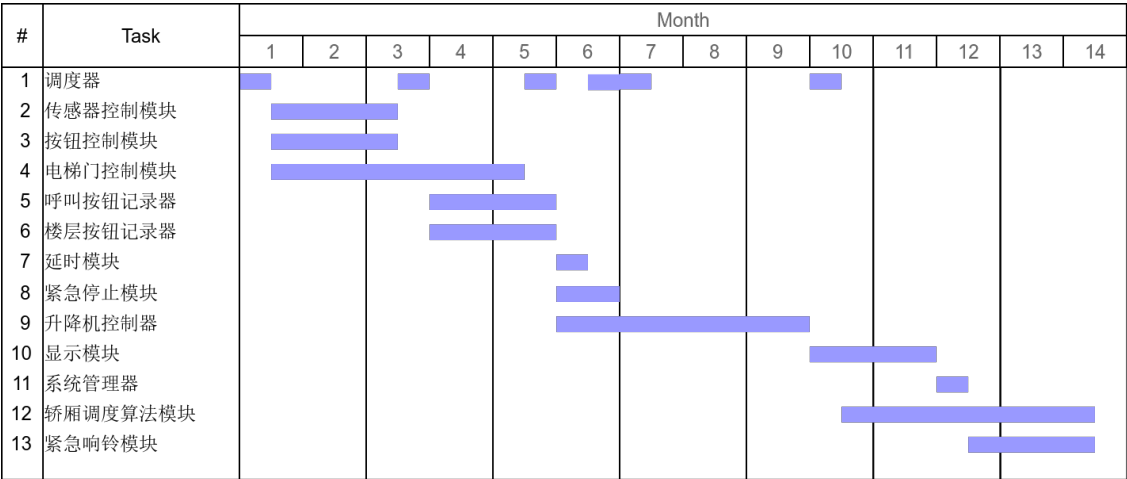


图 10.3.2: 电梯控制系统软件开发 Gantt 图

附录 A 参考文献

- [1] Robert H Crites and Andrew G Barto. “Improving elevator performance using reinforcement learning”. In: *Advances in neural information processing systems* 8 (1996).
- [2] 刘美菊, 刘冬, 刘剑, et al. “基于匈牙利算法的群控电梯调度的实现”. In: *沈阳建筑大学学报 (自然科学版)* 2013 年 05 (2013), pp. 955–960.
- [3] 宗群 and 王朝阳. “电梯群控系统多目标模糊控制算法的设计”. In: *制造业自动化* 22.7 (2000), pp. 23–25.
- [4] 宗群 et al. “基于遗传算法模糊神经网络的电梯群控调度”. In: *天津大学学报: 自然科学与工程技术版* 36.5 (2003), pp. 585–589.
- [5] 李军生 et al. “基于 ARM+ Linux 的智能电梯控制系统及运行环境”. In: *测试技术学报* 2 (2014), p. 015.

附录 B 体会与建议

这次软件工程实作业我体会到了软件工程的方方面面。在开始一个项目前需要问题定义和可行性分析，需要谨慎地分析项目是不是值得去做，成本多少，解决了什么问题等等，真正地把一个项目上升到工程的水准。对于将要开发的项目，详细的工程文档能够快速解决开发者之间的沟通问题，能够事先合理地构建出工程的结构，方便开发者后续的快速开发。开发者按照文档的开发计划进行开发，在相关接口设计上按照文档的说明去做，可以统一项目工程的数据结构和接口，节省开发的时间精力等资源。

在此项目的设计过程中，我的综合能力也得到了锻炼，对于问题的解决考虑得更加周全。不仅如此，借由这个项目，我对于神经网络算法、遗传算法以及匈牙利算法等可以用于调度的算法有了更深入的了解，这是我一开始没有预料到的，可见软件工程航安范围之广。

在编写此报告的过程中，我发现很多方面与书本上的描述不完全相同，虽然大体内涵相同，但是细节的不同也给此报告的编写带来了不小的困扰，因此希望在以后的课程中可以对报告编写的细节方面作一些描述。