

一. 设计概述

1.开发背景

在现代的大棚种植技术中，温度、湿度、二氧化碳浓度和光照是大棚蔬菜能否茁壮生长的四要素，这四个方面的制约关系完全可以构成一个学科体系，需要掌握四者之间的微妙关系才能使得蔬菜增产。温室大棚是用来栽培农作物的设施，它能够外界的四季变化和恶劣气候，为农作物的生长创造适宜的条件。温室大棚作为高效农业的重要组成部分，已经成为大棚行业研究的重要课题之一。现在我国大棚生产规模虽然空前巨大，但是有些农村蔬菜大棚的设备仍然比较陈旧，温度采集方式落后，并且费时费力，不利于大棚生产规模的扩大，也不利于信息化程度的提高。在湿度和二氧化碳浓度的采集和控制上，也难以达到资源的优化配置，在好多农村的蔬菜大棚中甚至没有湿度采集模块，二氧化碳浓度的检测也较为简单，形不成一套整体上的、信息化的、集成程度高的监控和管理系统。这在一定程度上影响了蔬菜大棚的产量，更为严重的是，对湿度和二氧化碳浓度监控和管理的不到位，影响到了菜农本身的身体健康，受潮湿高温的大棚环境的影响，好多菜农的肌肉和关节等出现病变，严重者甚至会丧失劳动能力，于是，我国大部分有温室大棚历史的农村，迫切需要一套温室大棚的智能监控系统，对大棚内的温度、湿度、二氧化碳浓度做一个有效的监控和管理。

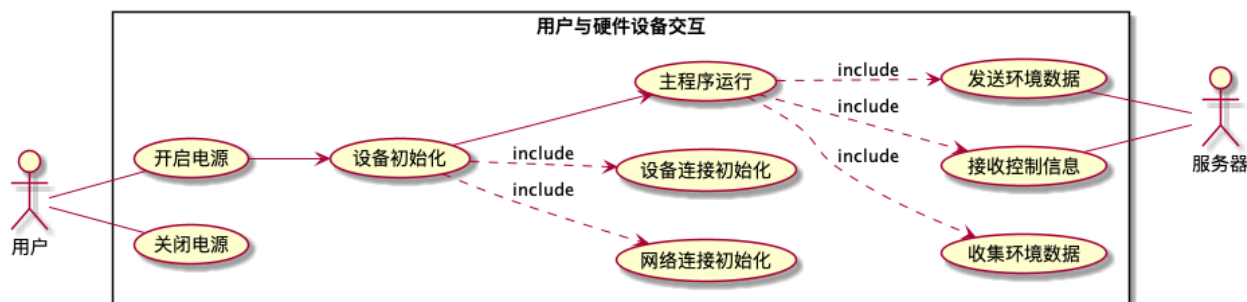
2.简介

目前，在农业生产过程中，最具标志性的生产技术是温室大棚种植技术传统农业大棚存在设备简陋、从业人员劳动强度大，智能化程度低、农产品产量低等问题，智能温室大棚不仅能够有效解决我国对农产品需求日益增高的问题，也解决了以往的农业生产中智能化、信息化程度低的难题，能够很好地节约生产管理过程中的人力成本。

Arduino是近年流行的一个基于开放原始代码的Simple I/O平台，支持多种传感器扩展板，操作简单，功能多样，广泛应用于电子系统设计和互动产品开发等方面。Arduino是一款便捷灵活、方便开发者使用和上手的开源电子原型平台，包含各种型号的Arduino板及扩展板硬件资源和Arduino IDE软件资源。Arduino不仅仅是全球最流行的开源硬件，也是一个优秀的硬件开发平台，更是当前硬件开发的趋势。本文设计的智慧农业温室大棚系统以Arduino Uno为主控单元，把最新的嵌入式技术、移动通信技术与农业温室种植相结合，采用温度传感器和土壤湿度传感器对大棚内部植物生长环境要素，如温度、湿度、CO₂浓度等，进行实时检测，并将大棚的各种状态数据通过GSM移动通信模块实时发送到管理人员手机终端，实现对大棚实时、高效的监测;通过温湿度传感器监测大棚温湿度。同时，管理人员也可以通过GSM模块将一些控制信号(如温度命令、湿度命令等)发送给Arduino，驱动相应外围设备工作，实时调整大棚内的温度、湿度以及CO₂浓度等数值，从而实现对大棚的远程控制。

二. 需求分析

1. 用户与硬件设备交互用例



网络连接初始化

功能:在设备开机后,自动通过网络连接与服务器建立通讯

前置条件:电源开启

后置条件:主程序运行

设备连接初始化

功能:在设备开机后,自动与有线或无线连接的设备建立通讯

前置条件:电源开启

后置条件:主程序运行

主程序运行

功能:在设备初始化完成后,进入稳定运行状态,定时完成规定的目标操作

前置条件:设备初始化完成

后置条件:设备主函数进入循环中

收集环境数据

功能:从所连接的传感器处获取实时环境数据

前置条件:与传感器设备成功建立连接

后置条件:收到传感器发来的环境数据

发送环境数据

功能:向服务器发送收集到的环境数据

前置条件:与服务器连接正常,收到各个连接的传感器发来的数据并已经处理完成,

后置条件:收到服务器的确认信息

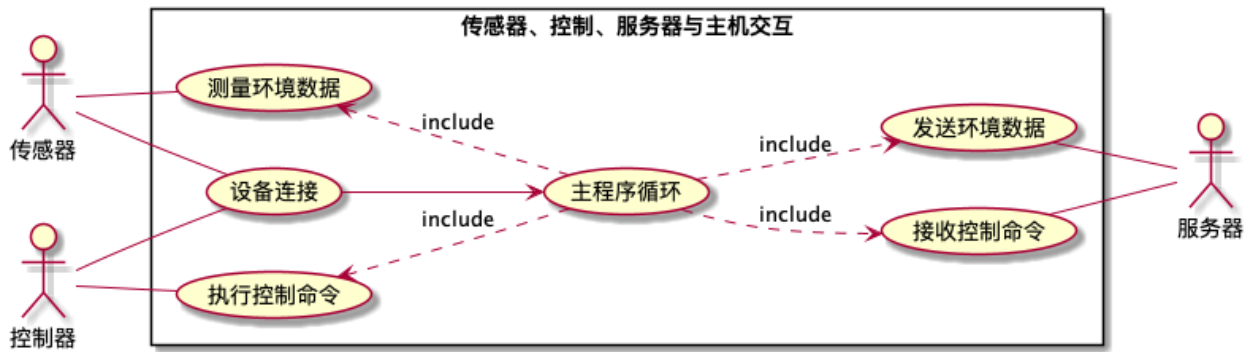
接收控制命令

功能:接收服务器发来的控制数据,并要求连接的控制器执行相应操作

前置条件:监听接收端口,收到服务器发来的控制信息

后置条件:执行相应操作

2. 传感器与主机交互用例



测量环境数据

功能:通过连接的传感器设备测量环境的多方面数据

前置条件:定时器到时,与传感器连接正常

后置条件:得到包含传感器设备信息的环境数据

发送环境数据

功能:将收集到的环境数据集合整理交付连接的服务器处理

前置条件:与服务器连接正常,收到各个连接的传感器发来的数据并已经处理完成

后置条件:收到服务器的确认信息

接收控制命令

功能:接收服务器发来的控制数据,并要求连接的控制器执行相应操作

前置条件:监听接收端口,收到服务器发来的控制信息

后置条件:执行相应操作

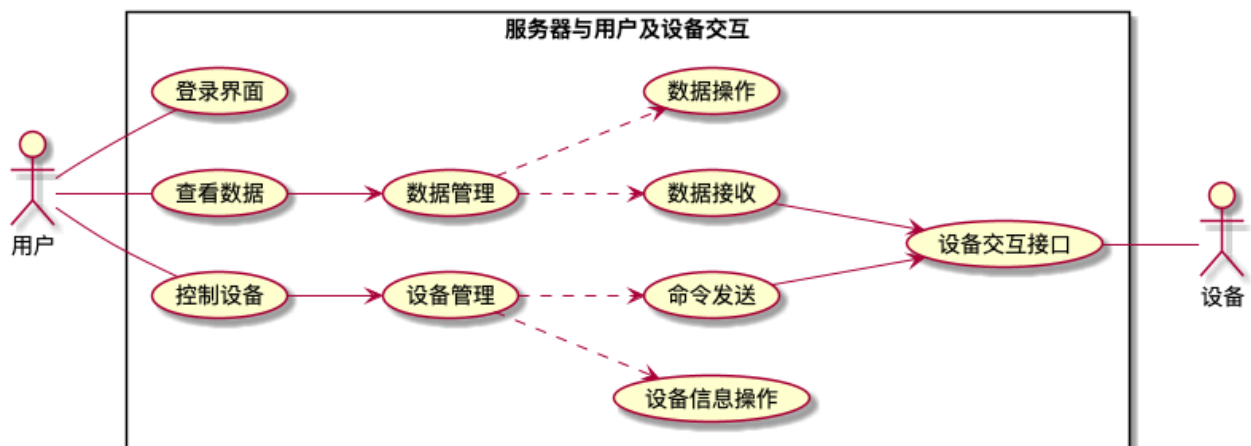
执行控制命令

功能:将服务器发来的控制命令转化为实际的机械操作

前置条件:收到服务器发来的命令,与控制设备连接正常

后置条件:命令得到执行,环境被自动调节

3. 服务器与用户及设备交互用例



登录界面

功能:确保用户安全正确地登录服务端

前置条件:用户输入正确的账户信息

后置条件:显示内部管路界面

查看数据

功能:显示用户账户下的设备信息,以及设备上传的环境数据

前置条件:用户登入管理界面,设备数据正确及时上传到服务器

后置条件:显示用户的设备信息,设备的运行状态,温室内的环境信息

控制设备

功能:用户可以预先设定运行规则,或是直接远程控制设备

前置条件:远程设备已连接服务器

后置条件:向设备发送控制指令或是在服务器中记录运行规则

数据接收

功能:接收设备发来的数据,交付主程序处理

前置条件:监听端口,接收设备发来的数据

后置条件:得到设备的运行数据

数据操作

功能:管理内部数据(不可修改记录,只能删除无用旧数据),或是供程序进行分析处理

设备信息操作

功能:管理连接的设备

后置条件:设备记录得到更新

命令发送

功能:将经过分析处理得到的命令结果,或是用户所确认的操作命令,发送给对应的设备执行

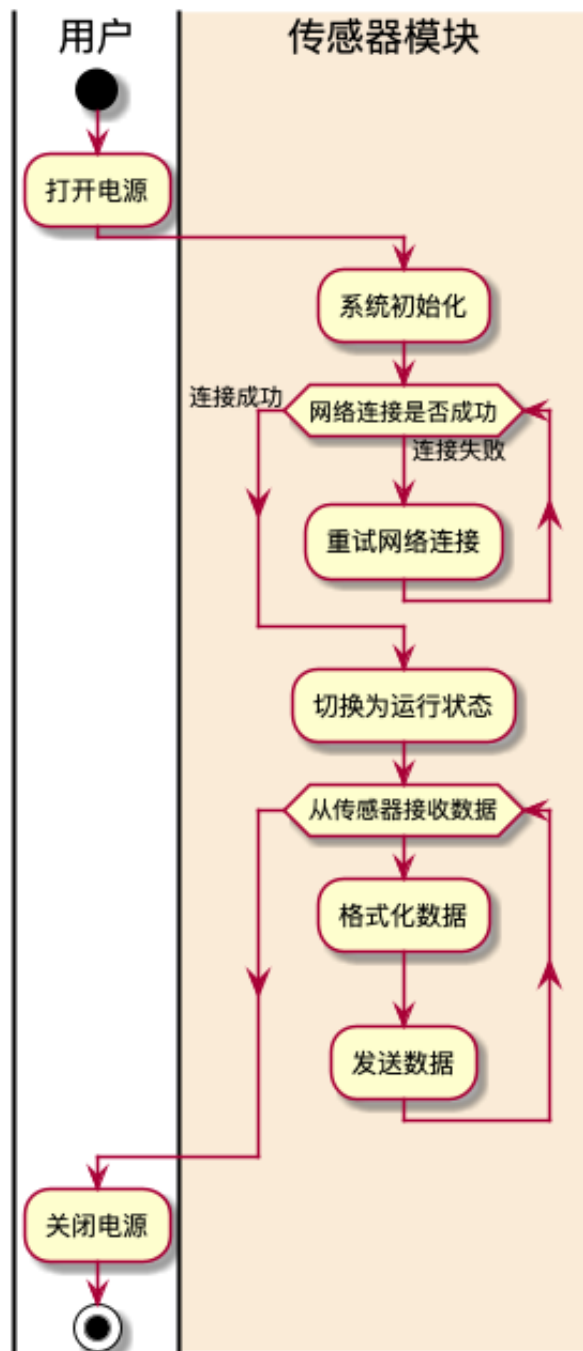
前置条件:有命令信息待发送

后置条件:设备得到需要执行的命令

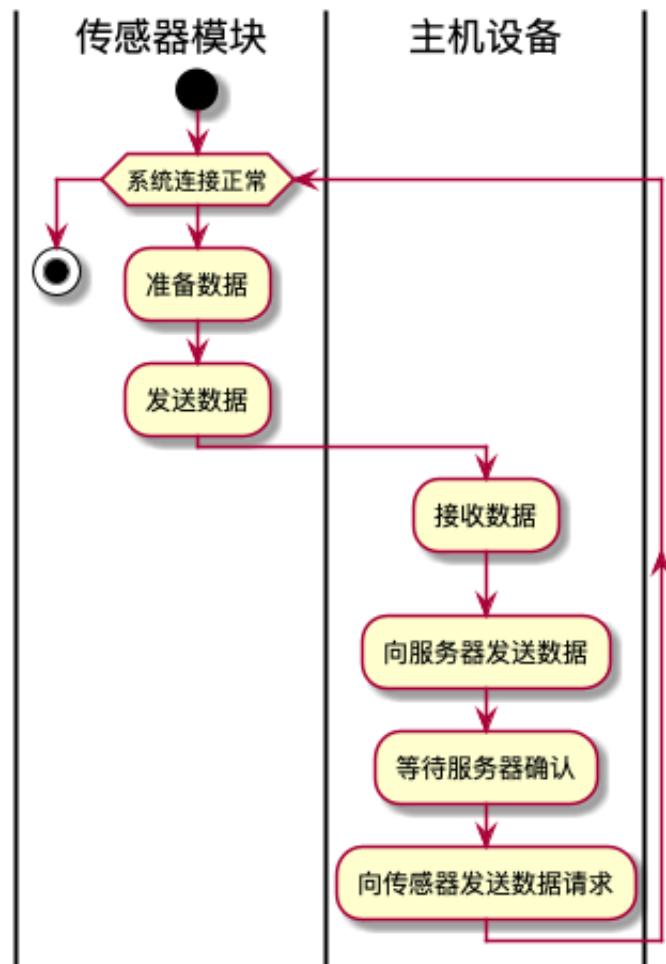
三. 对象分析

1. 对象活动图

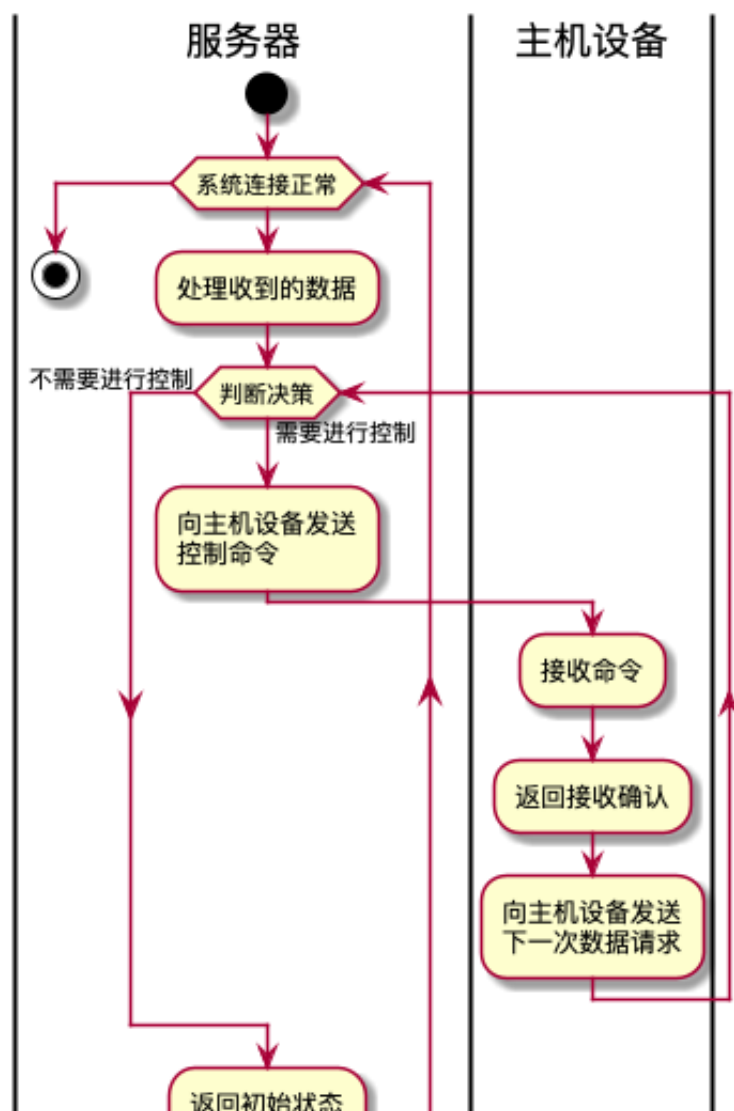
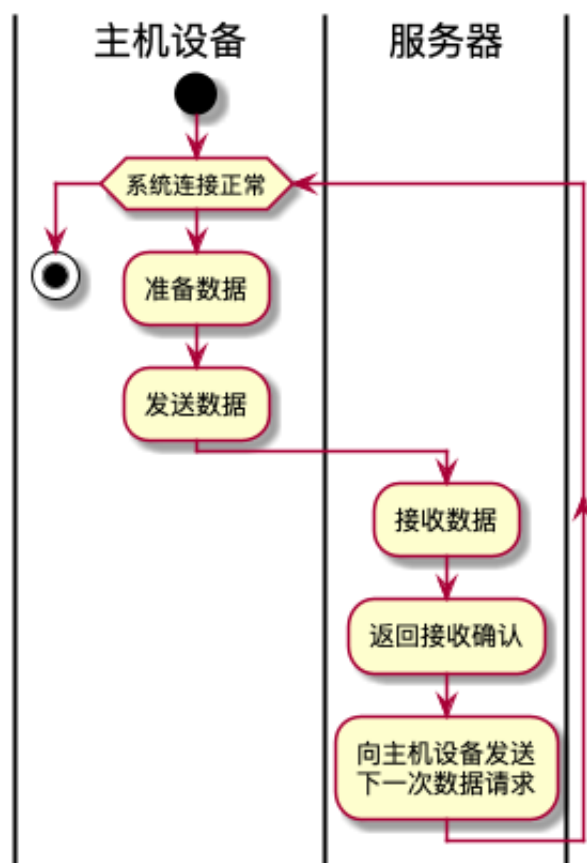
用户与传感器模块交互活动图



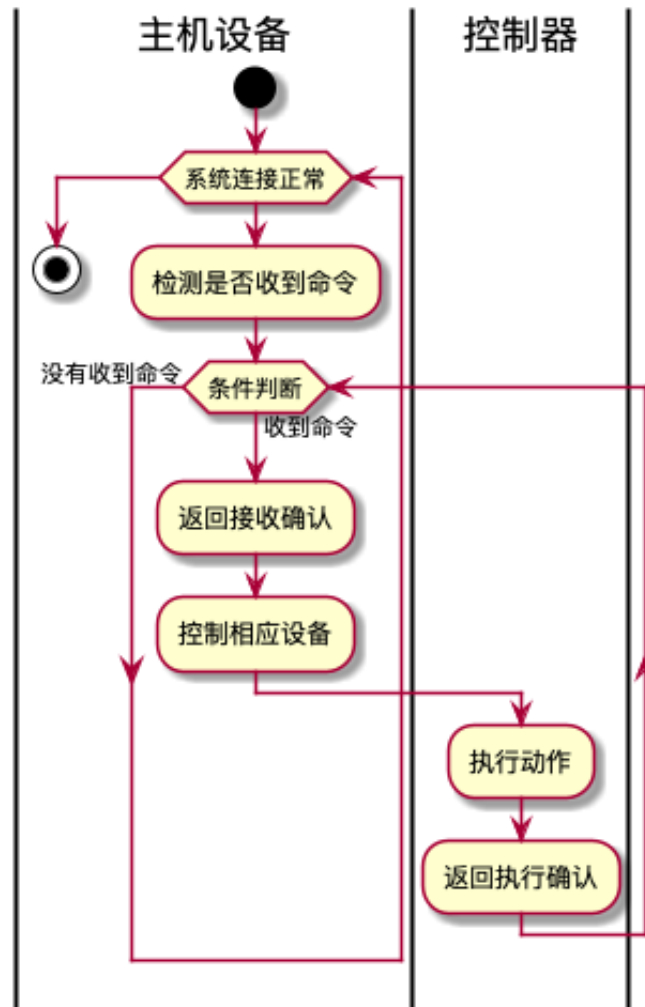
传感器模块与主机交互活动图



主机设备与服务器交互活动图

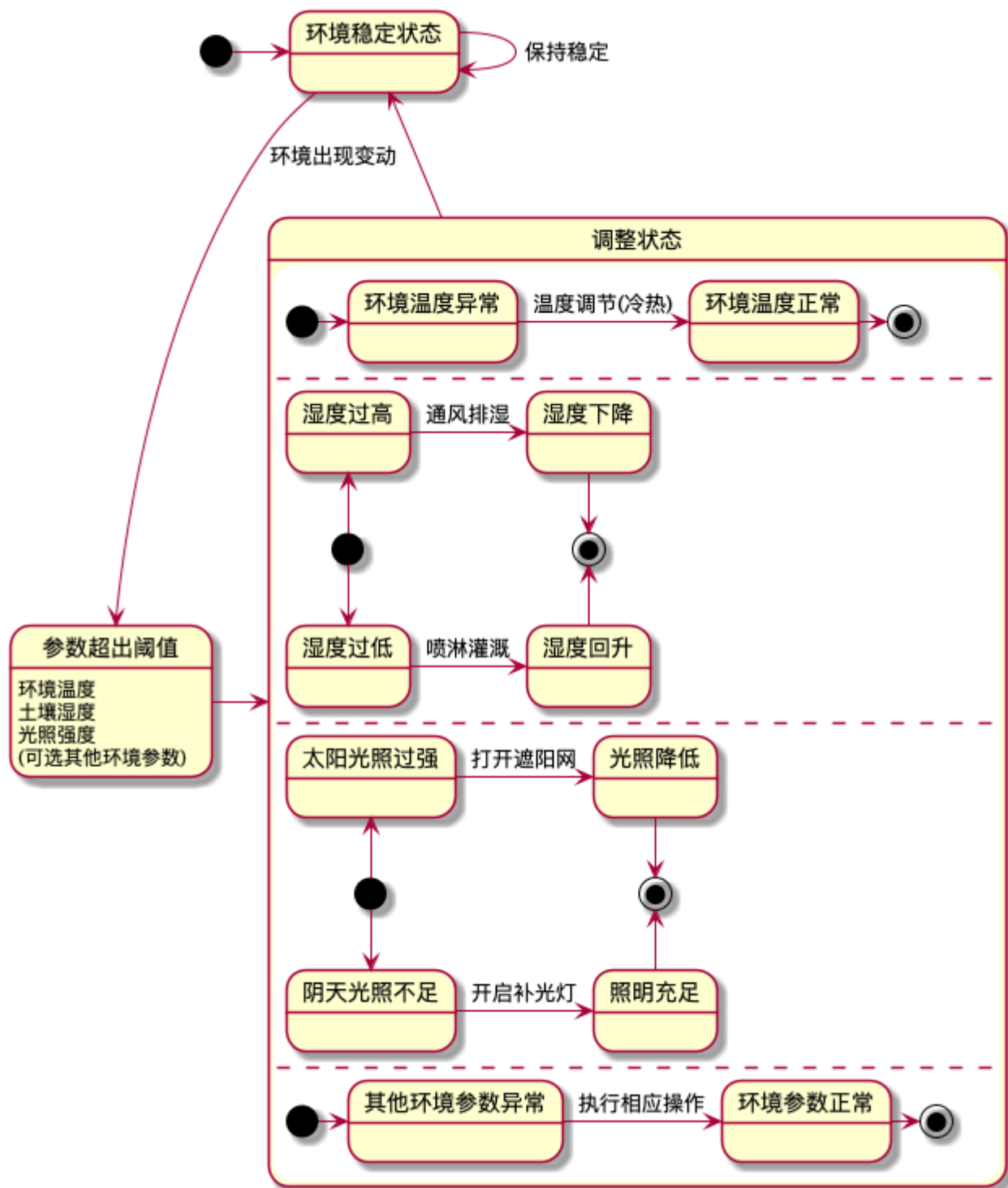


主机设备与控制器交互活动图

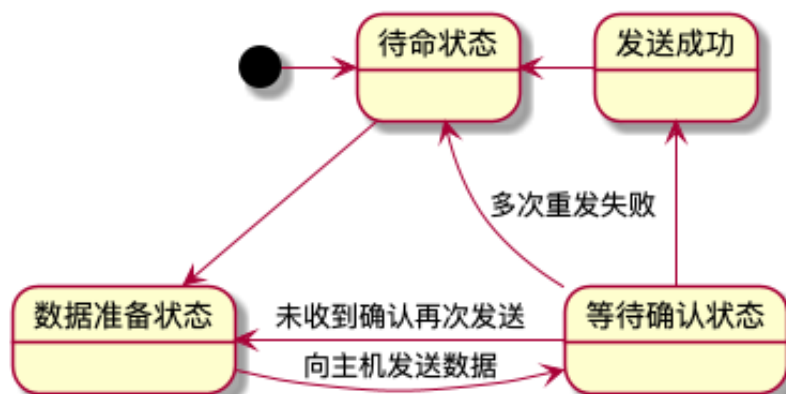


2. 对象状态图

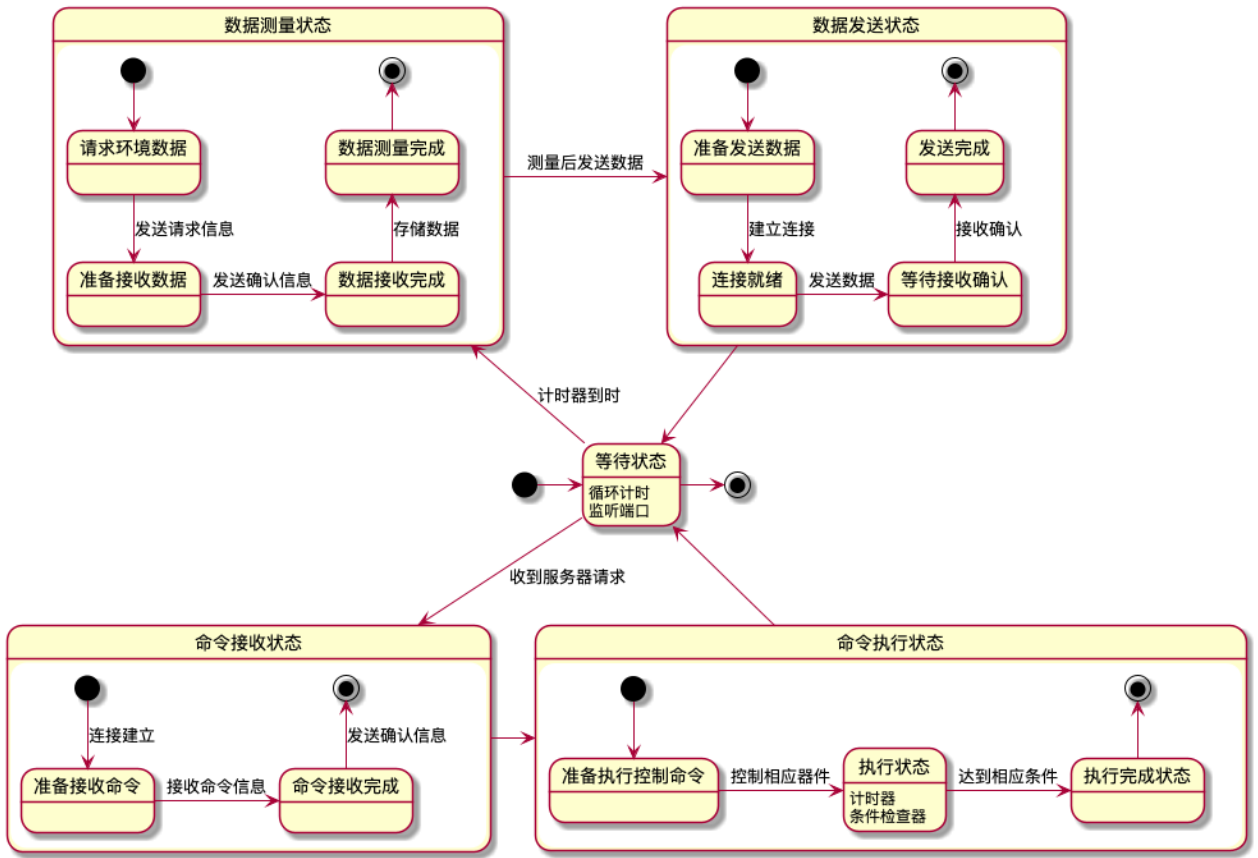
内部环境状态图



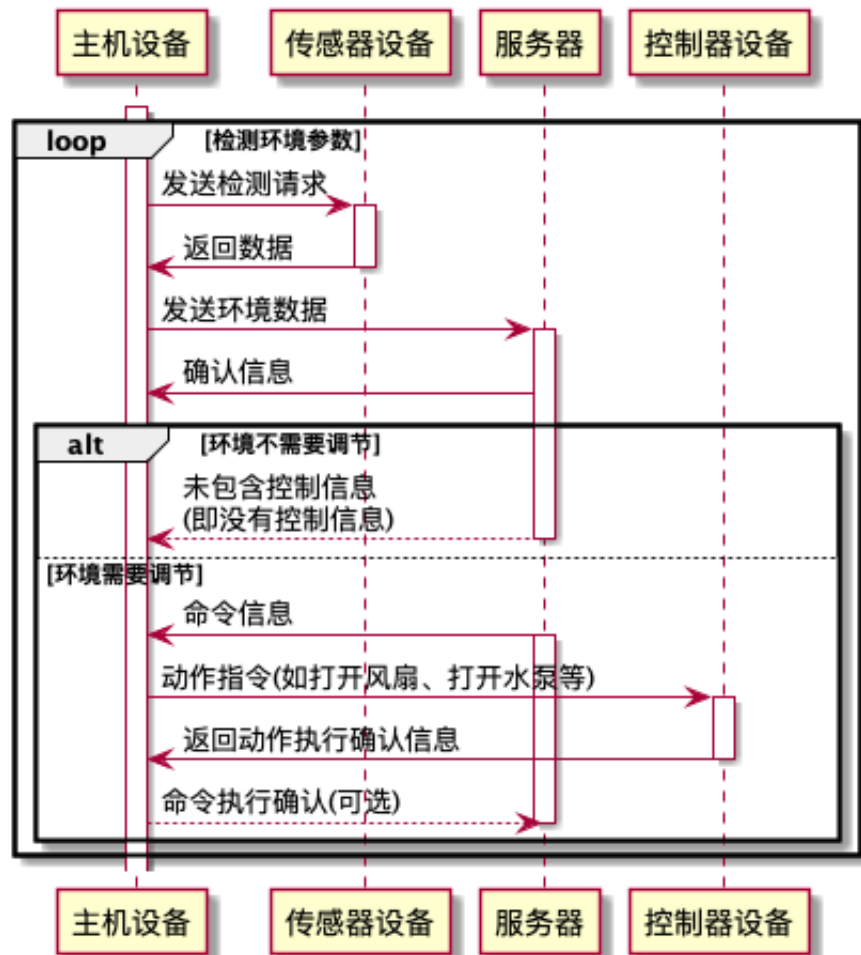
传感器运行状态图



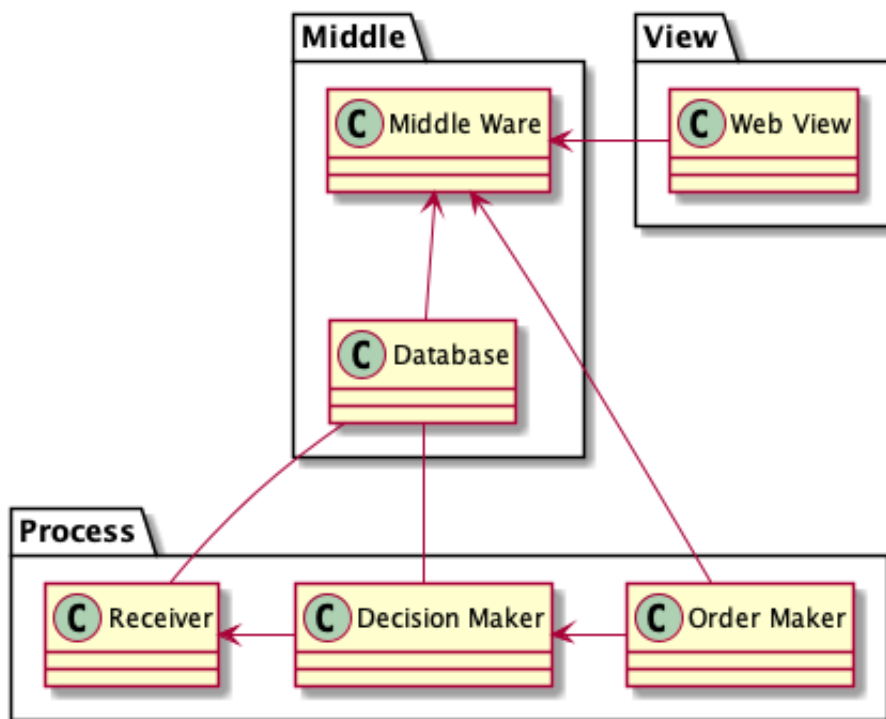
主机状态图



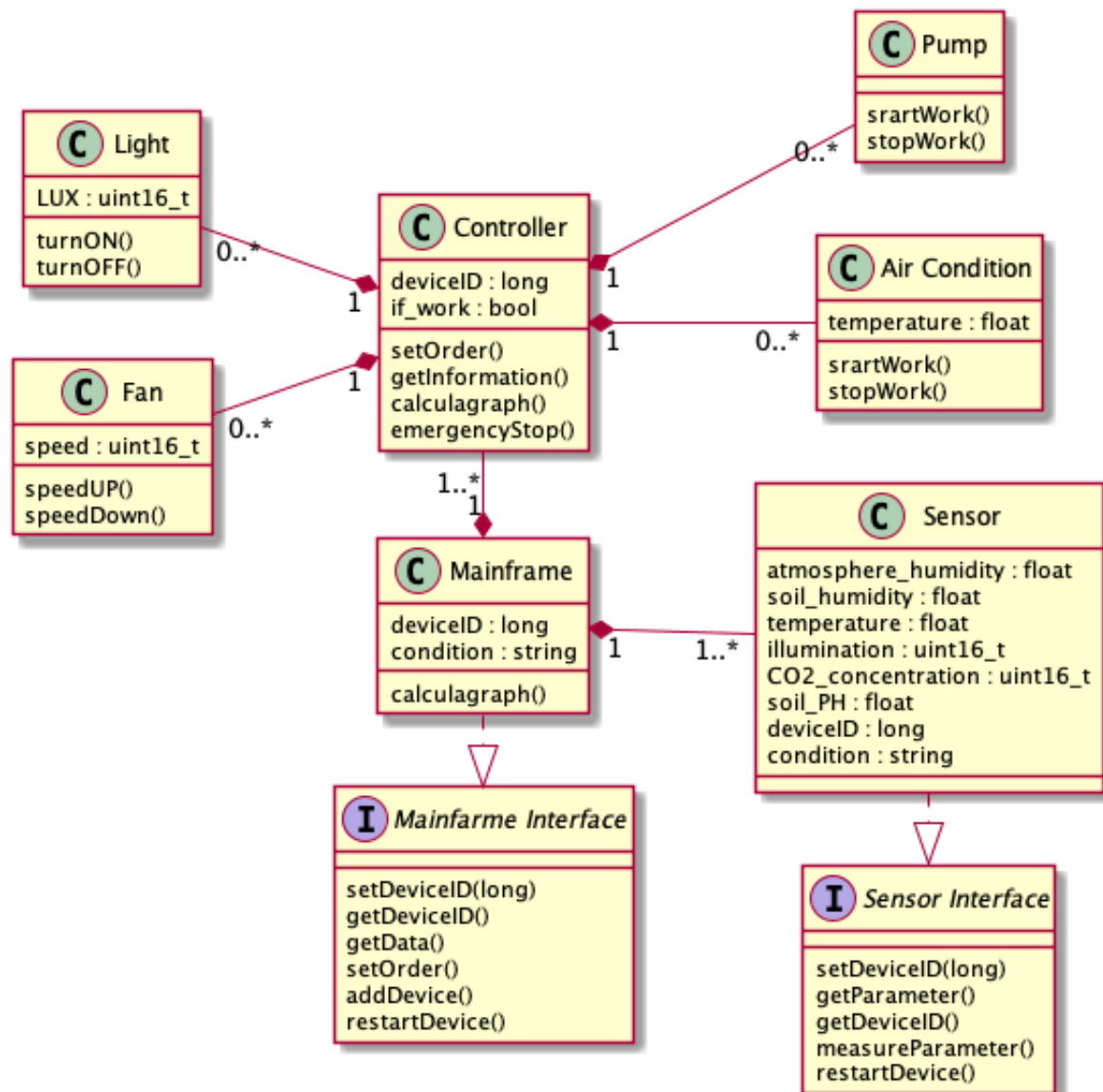
3. 对象顺序图



四. 详细设计



1. 主机设备分析设计



1.1 Sensor 类

功能:主要负责传感器部分运行,可以被主机设备接入并设置设备号

Sensor Interface 接口

setDeviceID(long)

功能:设置传感器设备的 ID,方便明确数据采集对象

输入:long 型设备 ID

getParameter()

功能:生成包含环境参数信息的比特流,并向外部设备发送

输出:规定格式的比特流

getDeviceID()

功能:为外部设备提供设备 ID

输出:long 型设备 ID

measureParameter()

功能:立即从串口获取传感器测量得到的环境参数,存入对象缓存区中

输出:True--测量成功;False--测量失败

restartDevice()

功能:重启传感器设备,并清除相应缓冲区

输出:True--重启成功;False--重启失败

1.2 Mainframe 类

功能:主机设备通过内部定时器执行循环操作,即从所连接的传感器设备读取参数信息(比特流),解析并向服务器发送得到数据;同时接收服务器的控制命令,通过连接的控制器来执行控制操作

calculagraph()

功能:作为内部计时器,定时激活相应内部操作

Mainfarme Interface 接口

setDeviceID(long)

功能:设置传感器设备的 ID,在新增设备并入网络后由服务器设置设备信息

输入:long 型设备 ID

getDeviceID()

功能:为服务器设备提供设备 ID

输出:long 型设备 ID

getData()

功能:为服务器提供获取主机设备记录数据的接口

输出:JSON 格式包含了主机设备信息以及设备信息的文件

addDevice()

功能:添加链接到主机的所有设备

输出:True--添加成功;False--添加失败

restartDevice()

功能:重启传感器设备,但不清除连接设备的记录信息

1.3 Controller 类

功能:控制所连接控制设备,检测设备的运行状态,为主机设备执行控制命令

setOrder()

功能:服务器向主机设备发送的命令,向对应控制器转发控制指令来控制环境

输入:包含控制信息的JSON 文件

输出:返回接收确认信息

getInformation()

功能:服务器获取控制器运行信息####

输出:包含运行信息的JSON 文件

calculagraph()

功能:内部计时器来确定设备运行时间

emergencyStop()

功能:在发生紧急情况时停止一切控制器件的操作

1.3.1 Fan 类

speedUP()

功能:增加风扇转速

speedDown()

功能:降低风扇转速

1.3.2 Light 类

turnON()

功能:开启照明

turnOFF()

功能:关闭照明

1.3.3 Pump 类

srartWork()

功能:开启水泵

stopWork()

功能:关闭水泵

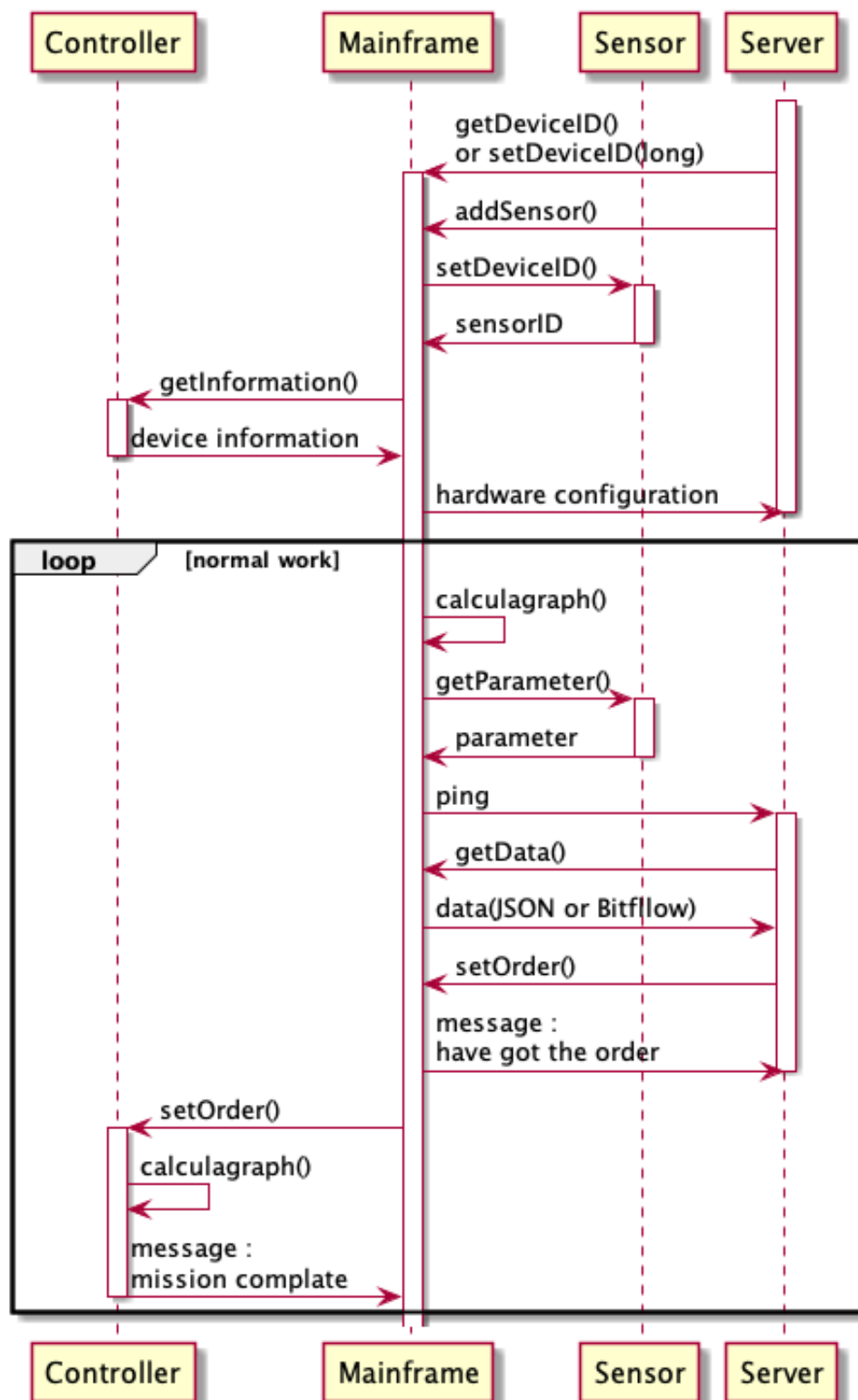
1.3.4 Air Condition 类

srartWork()

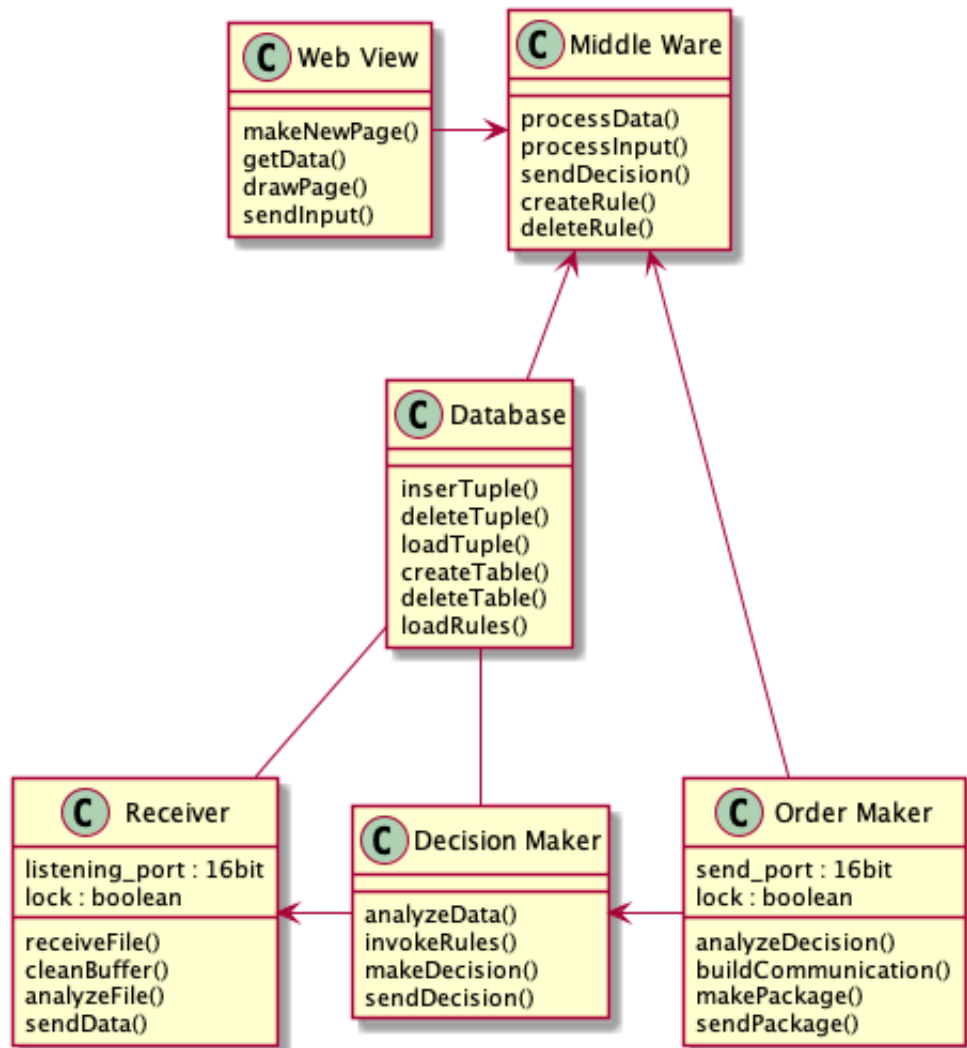
功能:开启空调

stopWork()

功能:关闭空调



2. 服务器端分析设计



2.1 Receiver

功能:监听端口,接收并处理发来的信息,并交付后层处理

receiveFile()

功能:接收发来的信息信息

输出:接收到的文件或比特流

cleanBuffer()

功能:清空缓冲区

输入:缓冲区地址

输出:True--清空成功;False--清空失败

analyzeFile()

功能:解析发来的文件或比特流,将内容转化为可被服务器处理的符合数据库格式的记录

输入:文件的地址或记录的头指针

输出:固定格式的记录

sendData()

功能:将解析好的记录转送给数据库以及处理程序

输入:数据库地址,处理程序入口

输出:True--发送成功;False--发送失败

2.2 Decision Maker

analyzeData()

功能:分析收到的环境数据,读取出地块标识,环境参数,时间戳等多种信息

输入:固定格式的记录

invokeRules()

功能:从数据库读取对应地块的处理规则

输入:地块的标识信息

输出:可以用于决策的规则

makeDecision()

功能:通过预设的规则以及实际参数信息判断此地块设备需要进行的操作或保持的状态

输入:环境参数,时间信息,处理规则

输出:决策结果

sendDecision()

功能:将处理程序得到的决策结果发送给数据库保存,以及命令发送程序处理

输入:数据库地址,发送程序入口

2.3 Order Maker

功能:将处理程序得到的决策以及用户确定的猝发命令发送给连接的设备

analyzeDecision()

功能:分析决策命令,提取出命令与目标字段

输入:决策命令

buildCommunication()

功能:与设备建立通讯连接,以便发送数据

输出:True--连接建立成功;False--连接建立失败

makePackage()

功能:将分析好的命令以互联网规定的格式封装起来

输出:规定格式的数据包

sendPackage()

功能:将封装好的数据包发送给对应的设备

2.4 Database

功能:存储一段时间内的环境参数以及决策命令,以便调取或分析研究

inserTuple()

功能:在相应数据表中插入新的记录

输入:参数信息或是决策命令

输出:True--插入成功;False--插入失败

deleteTuple()

功能:在相应数据表中删除相应记录

输入:需要删除的记录的主键信息

输出:True--删除成功;False--删除失败

loadTuple()

功能:读取一条记录

输入:查询条件

输出:记录信息

createTable()

功能:新建一张对应于实际设备的参数或命令表

输入:设备的唯一标识信息

输出:True--建立成功;False--建立失败

deleteTable()

功能:删除一张对应于实际设备的参数或命令表

输入:设备的唯一标识信息

输出:True--删除成功;False--删除失败

loadRules()

功能:读取所有相关的规则记录

输入:目标规则唯一标识信息

输出:规则记录信息

2.5 Web View

功能:负责生成视图化的浏览器页面

makeNewPage()

功能:新建页面

输入:页面配置文件

getData()

功能:获取页面需要显示的数据

输入:显示数据文件

drawPage()

功能:根据页面配置文件以及实时的数据渲染出显示的页面

输出:页面文件

sendInput()

功能:将用户在页面中输入的数据信息传递给中间层

输入:用户键入

输出:键入信息

2.6 Middle Ware

功能:负责为页面提供页面需要的信息,接收用户输入的信息,处理从数据库读取和接收端发来的信息,发送要被存储和执行的信息

processData()

功能:处理数据库和接收端发来的信息

输入:数据库的记录或是接收端发来的信息

输出:可被页面读取的数据

processInput()

功能:处理用户输入

输入:界面发来的键入信息

输出:可被程序理解的命令或规范化的数据库记录

sendDecision()

功能:将处理好的用户输入发送给相应的程序或数据库

输入:目标程序入口

createRule()

功能:为目标地块设备创建新的判断规则,

输入:用户输入

输出:规则记录条目

deleteRule()

功能:删除目标地块设备的判断规则
输入:用户输入

