# 系统概要设计

## 系统的功能设计

**1.1.1人员管理模块**

人员在进入阅览室前，需要刷卡，只有具有权限的卡才可以打开门，同时数据库记录该条信息，包括时间，人员id。

**1.1.2书籍管理模块**

当某个人打开书柜时，记录书柜书籍情况，当书柜关闭时，在记录下书柜的书籍情况，通过两次对比可得哪些书在何时被哪个人借阅，将此条信息记录下来。当发生还书事件时，处理方法同上。

**1.1.3环境监测模块**

实时记录405阅览室的温度，湿度，烟雾浓度信息，当发生温度骤升，烟雾浓度骤升时，发送火情预警信息。

## 系统技术架构设计

基于现有的设备和需求分析中的要求，我们的整个系统主要分为感知层，传输层，服务平台层以及应用层。

* 感知层

感知层主要为现有的传感器，包括烟雾传感器，温湿度传感器，门磁传感器，RFID等等。

* 传输层

传感器的内部数据传输采用I2C总线传输，每个传感器通过串口与上位机链接，上位机之间形成组网，起到数据融合和数据过滤等功能。

* 服务平台层

该层次为本系统的大脑和核心，应用层的功能都要基于该层次的数据处理实现，接收来自传输层的数据，对数据进行整合和存储，同时负责向感知层传输来自应用层的命令，如开门，摄像等等。

* 应用层

主要实现数据的展示盒突发情况的通知，其中数据展示包括书籍信息，人员信息，室内环境信息。在发生火情、漏雨、书籍被带出阅览室等情况产生推送。

## 系统的存储设计

选用MongoDB对数据进行存储，设计两个库，一个存储405的元数据和状态，另一暂时性存储历史数据用于将来导入OLAP服务器进行分析。

表1.3-1状态数据库存储设计

|  |  |
| --- | --- |
| User | {  id: Object\_id  name: String  card\_num: String // 持有的rfid卡编号  info: {  student\_num: String // 学号  phone: String // 手机  QQ: String  email: String  ...other informations about this user  }  } |
| Book | {  id: Object\_id  bookinfo: { // 对于相同的书，这个字段存在冗余  ISBN: String  name: String  ...other infomations about this book  }  position: User // 此字段为空表示书位于书柜  } |

表1.3-2 日志数据库存储设计

|  |  |
| --- | --- |
| Temperature | {  timestamp: Integer  value: Number  isSpecial: Bool // 是否超出阈值，冗余字段，便于装入合并处理时减小计算量  } |
| Pressure | {  timestamp: Integer  value: Number  isSpecial: Bool  } |
| Humidity | {  timestamp: Integer  value: Number  isSpecial: Bool  } |
| RFID\_Bookdoor | {  timestamp: Integer  value: String  } |
| RFID\_Door | {  timestamp: Integer  value: String  } |
| RFID\_Book | {  timestamp: Integer  value: Array<String>  } |
| Hall\_Bookdoor | {  timestamp: Integer  value: Bool  } |
| Hall\_Door | {  timestamp: Integer  value: Bool  } |
| Hall\_Bookdoor | {  timestamp: Integer  value: Bool  } |
| Camera | {  timestamp: Integer  content: Blob // 拍摄的照片  } |
| 以上为原始流水日志，以下为冗余的事务处理日志 | |
| Command | {  timestamp: Integer  target: enum{'door','bookdoor'}  content: String  } |
| Alert | {  timestamp: Integer  type: String // 警报类型，比如温度超过上限，门长时间没关等  value: Any // 和type相关的参数，比如温度超过上限时的具体温度值等。  } |
| Book\_Leading | {  timestamp: Integer  book: Book  borrower: User  } |
| Book\_Return | {  timestamp: Integer  book: Book  } |
| Door\_Open | {  timestamp: Integer  } |
| Door\_Close | {  timestamp: Integer  } |

# 系统详细设计



## 系统感知层详细设计

## 系统感知层详细设计

本系统采用了温湿度传感器、烟雾传感器、门磁传感器、RFID、继电器、气压传感器。在本节中，将对以上传感器进行原理、使用、数据传输等方面的描述。

### 温湿度传感器



图 .1-1温湿度传感器原理图

本系统中，温湿度传感器用来检测室内的环境特征值，实时监控室内的情况。辅以烟雾传感器，可以判断火灾的发生。

温湿度传感器需要将DATA引脚和SCK引脚与单片机相连，所以使用单片机的P0\_6和P0\_7引脚分别连接到DATA和SCK口，并且给两个引脚发送相应的时序就能控制温湿度传感器。

要驱动SHT10进行采样必须发送如下命令：首先，向SHT10发送“启动传输”时序，完成数据传输的初始化。如图4-3所示，时序包括当SCK时钟高电平时，DATA发转为低电平；紧接着SCK变为低电平，随后在SCK时钟高电平时，DATA翻转为高电平。初始化之后，单片机便可以向SHT10发送命令。通常的命令包括3个地址位（目前只支持“000”）和5个命令位，具体将在后面的代码中进行介绍。SHT10会以下述方式表示已正确的接收到命令：在第8个SCK时钟的下降沿之后，将DATA下拉为低电平，并且在第9个SCK时钟的下降沿之后，将DATA位恢复为高电平。

启动程序完成之后，SHT10便会以串行数据的方式与单片机进行通信，时序图如图4-4所示。DATA三态门用于数据的读取，DATA在SCK时钟下降沿之后改变状态，并仅在SCK时钟上升沿有效。数据传输期间，在SCK时钟高电平时，DATA必须保持稳定。为避免信号冲突，单片机应驱动DATA在低电平。在上升沿之前把数据写入，上升沿时数据有效，在下降沿时把数据发送给传感器。

* + 计算温度的公式为： Temperature = a + b \* X

式中，a，b为参数；X为传感器采集的温度数据。

对于本实验采用3V电压、14位采样精度，查芯片手册可知：a = --39.67，b = 0.01。带入计算即可得到以摄氏度为单位的温度值。在极端工作条件下测量温度时，可使用进一步的补偿算法以获取高精度。可参阅应用说明“相对湿度与温度的非线性补偿”。

* + 计算湿度的公式为： Humidity = c + d \* Y + e\* Y^2

式中，c，d，e为参数；Y为传感器采集的湿度数据。

对于本实验采用12位采样精度，查芯片手册可知：c = -4，d = 0.0405，e = -2.8 \* 10^-6。

温湿度传感器程序流程图如下：



图 ‑1-2 温湿度传感器程序流程图

### 烟雾传感器



图.1-3 烟雾传感器原理图

本系统中采用烟雾传感器，检测火灾的发生，并产生及时的预警。辅以温度的数据，可以有效排除由于吸烟等其他情况产生的误差。

物联网实验箱选用的烟雾传感器型号为MQ2。MQ2属于半导体气敏式烟雾传感器。它由微型AL2O3陶瓷管、SnO2敏感层，测量电极和加热器构成的敏感元件固定在塑料或不锈钢制定的腔体内，加热器为气敏元件提供了必要的工作条件。封装好的气敏元件有6只针装管脚，其中4个用于信号取出，2个用于提供加热电流。MQ2气体传感器所使用的气敏材料是在清洁空气中电导率较低的二氧化锡（SnO2）。当传感器所处环境中存在可燃气体时，传感器的电导率随空气中可燃气体浓度的增加而增大。使用简单的电路即可将电导率的变化转换为该气体浓度相对应的输出信号。该传感器常用于家庭和工厂的气体泄漏装置，适用于液化气、丁烷、丙烷、甲烷、酒精、氢气、烟雾等的探测，是一款适合多种应用的低成本传感器。

首先，烟雾传感器的加热丝预热一段时间后，才能正常检测烟雾。要求预热时间不小于24小时。烟雾传感器在A、B处检测周围空气的烟雾浓度，气敏材料二氧化锡（SnO2）在清洁空气中电导率较低，当可燃气体浓度增加时，气敏材料电导率生高，气敏元件电阻阻值变小，电极A、B两端的分压很小，则B端输出的模拟信号电压升高。对B端输出的模拟电压信号进行ADC采样，即可以采集到环境中可燃气体浓度。

烟雾传感器程序流程图如下：

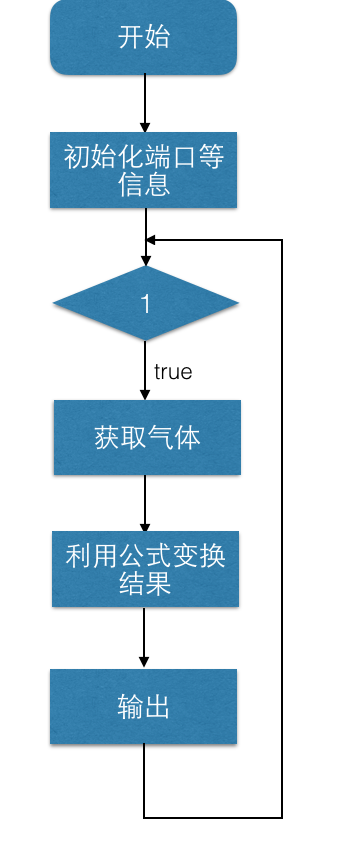
****

图 .1-4烟雾传感器程序流程图

### 门磁传感器



图 .1-5 门磁传感器原理图

本系统采用门磁传感器，对书柜门是否关闭进行检测。在书柜门上设置门磁传感器，在门上设置永磁体。通过门磁传感器对永磁体的检测，可以判断书柜门的状态，从而影响报警和作为其他控制的条件（图书数目扫描、门锁等）。

当磁铁靠近门磁传感器时，门磁传感器检测到磁，其信号输出管脚3输出低电平，当磁铁远离门磁传感器时，门磁传感器未检测到磁，其信号输出管脚3输出高电平。用单片机采集管脚3的电平变化，即可检测到磁铁是否从门磁传感器上移走，从而发出报警。管脚3与CC2531的P0\_6相连。

门磁的工作原理：当两个门磁距离1.5cm以内时，门磁线1和门磁线2之间导通，当两个门磁距离超过1.5cm时，门磁线1和门磁线2之间断开。实际安装时，应保证门闭合时，两个门磁之间距离不超过1cm。如图1，当门闭合时，门磁线1和门磁线2导通，P0\_7读入数据为0，当门打开时，门磁线1和门磁线2断开，P0\_7读入数据为1，程序以P0\_7读入数据判断门是否打开。

门磁传感器程序流程图如下：

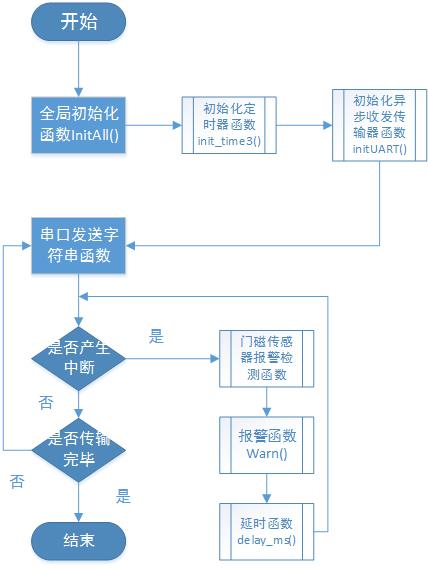


图 2.1-6 门磁传感器程序流程图

### 继电器

本系统中采用继电器对书柜门进行控制。继电器通过COM端口的输入控制电磁铁的通断，当书柜门关闭且到达锁门延迟时，继电器通电使电磁铁产生磁性从而吸合；当检测到借阅者的RFID卡并验证成功时，继电器断路关闭电磁铁，从而松开书柜门。

继电器（relay）是指当输入量（激励量）的变化量达到规定要求时，在电气输出电路中被控量发生预定的阶跃变化的一种电器。

继电器是一种点控制器件。它具有控制系统（又称输入回路）和被控制系统（又称输出回路）之间的互动关系。通常应用于自动化的控制电路中，它实际上使用小电流去控制大电流运作的一种“自动开关”。故在电路中起着自动调节、安全保护、自动化、运动、遥控、测量和通信等装置中。

继电器的输入量可分为电气量（如电流、电压、频率、功率等）及非电气量（如温度、压力、速度等）两大类。当输入量（如电压、电流、温度等）达到规定值时，继电器使被控制的输出电路导通或断开。

继电器程序流程图如下：

图 .1-7 继电器程序流程图

### 气压传感器

本系统采用气压传感器对室内的气压进行监控，提供环境数据。

物联网实验箱选用的气压传感器型号为MPL3115A2。MPL3115A2具有三项主要功能。一是测量环境大气压，二是测量高度，三是测量温度。

气压传感器与主设备通信使用I2C协议，硬件设计主要是把气压传感器的数据线、时钟线与CC2531连接起来，其中数据线SDA与CC2531的P0\_1相连，时钟线SCL与CC2531的P0\_0相连。

气压值表达为三个字节，例如，寄存器0x01的值为0x61, 寄存器0x02的值为0xF8, 寄存器0x03的值为0xE0。应用时将这三个字节组合成0x61F8E，即20位二进制位，丢弃最后四位二进制位。0x61F8E = 401294，此值401294 \* 0.25 = 100323.5（帕斯卡）。

气压传感器流程图如下：

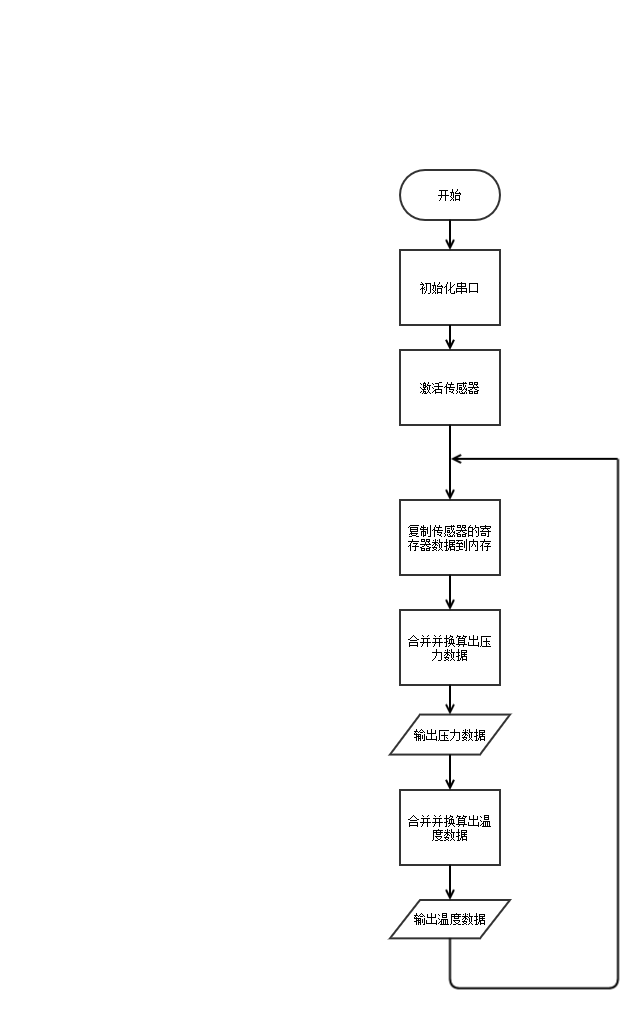


图 .1-8 气压传感器程序流程图

### RFID

本系统中采用RFID对图书和用户进行唯一标识。

每本图书上拥有唯一标识RFID卡，书柜内的读卡器进行扫描，获取图书的借阅状态。

每名用户拥有唯一的RFID卡，只有当卡验证成功时，才可以打开书柜门取图书，并在取走后将该用户与其拿走的图书进行记录。

PC机与RFID模块之间的通信采用UART，实现PC机对RFID模块的控制，从而实现标签的读数据、写数据操作。

PC机和RFID模块的通信采用面向字节的异步通信协议数据格式。规定主机发给RFID模块的数据帧为命令，RFID模块返回给主机的数据帧为响应。命令或响应数据帧是变长字节数，采用组包方法并用异或校验方法进行检错。

数据链路层具体规定命令帧和响应帧的类型与数据格式。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D\_LEN  1Byte | 设备类型  1Byte | 消息ID  2Bytes | 流水号  4bytes | DATA | CHECK  1byte |

Figure ‑ RFID数据格式

1. D\_LEN：该字段长度为1字节，表示DATA部分的长度。
2. 设备类型：区分ZigBee、RFID等不同设备；RFID：0x01；ZigBee：0x02。
3. 消息ID，区分不同设备的消息类型，不同设备类型的ID可以重复；

## 系统网络层详细设计

### 2.2.1 传感器与上位机

传感器与上位机之间采用串行通信方式，每个传感器对应一个上位机，传感器通过串口将数据发送到上位机进行接收，经过处理后将数据发送到服务器，接收到的服务器指令同样通过串口发送到传感器。

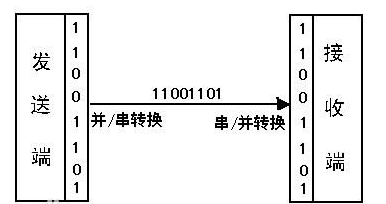


图2.2-1 串口通信过程图

由于 CPU 与接口之间按并行方式传输，接口与外设之间按串行方式传输，因此，在串行接口中，必须要有“接收移位寄存器”（串→并）和“发送移位寄存器”（并→串）。在数据输入过程中，数据1位1位地从外设进入接口的“接收移位寄存器”,当“接收移位寄存器”中已接收完1个字符的各位后，数据就从“接收移位寄存器”进入“数据输入寄存器”。CPU 从“数据输入寄存器”中读取接收到的字符。（并行读取，即D7~D0同时被读至累加器中）。“接收移位寄存器”的移位速度由“接收时钟”确定。

在数据输出过程中，CPU 把要输出的字符（并行地）送入“数据输出寄存器”，“数据输出寄存器”的内容传输到“发送移位寄存器”，然后由“发送移位寄存器”移位，把数据1位1位地送到外设。“发送移位寄存器”的移位速度由“发送时钟”确定。

### 2.2.2 上位机与服务器

机与服务器通过TCP/IP协议进行数据通信，上位机将数据融合的传感器信息通过Socket发送给服务器，上位机也可以将用户指令通过服务器建立Socket传送给上位机。

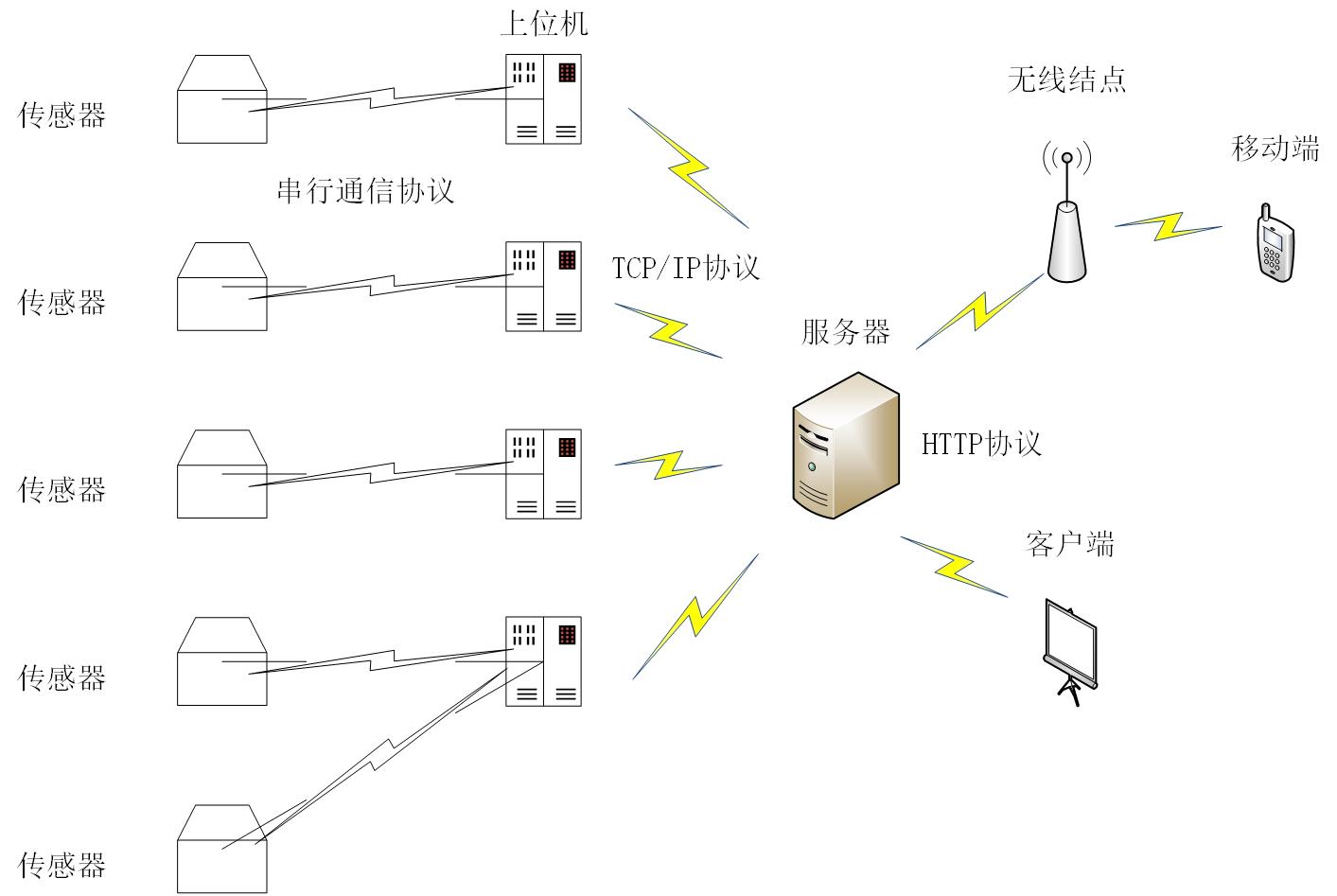


图2.2-2 智能监控系统组网图

TCP是面向连接的通信协议，通过三次握手建立连接，通讯完成时要拆除连接，由于TCP是面向连接的所以只能用于端到端的通讯。TCP提供的是一种可靠的数据流服务，采用“带重传的肯定确认”技术来实现传输的可靠性。TCP还采用一种称为“滑动窗口”的方式进行流量控制，所谓窗口实际表示接收能力，用以限制发送方的发送速度。如果IP数据包中有已经封好的TCP数据包，那么IP将把它们向‘上’传送到TCP层。TCP将包排序并进行错误检查，同时实现虚电路间的连接。TCP数据包中包括序号和确认，所以未按照顺序收到的包可以被排序，而损坏的包可以被重传。TCP将它的信息送到更高层的应用程序，例如Telnet的服务程序和客户程序。应用程序轮流将信息送回TCP层，TCP层便将它们向下传送到IP层，设备驱动程序和物理介质，最后到接收方。

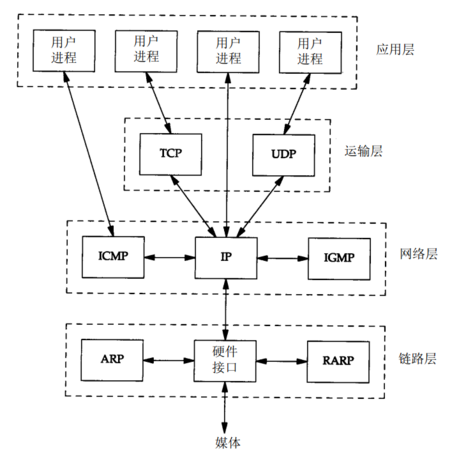


图2.2-3 TCP/IP协议栈图

### 2.2.3 服务器与客户端

服务器与客户端通过发送HTTP请求的方式获取对方的指令数据或监控数据。HTTP协议是基于请求/响应范式的。一个客户机与服务器建立连接后，发送一个请求给服务器，请求方式的格式为，统一资源标识符、协议版本号，后边是MIME信息包括请求修饰符、客户机信息和可能的内容。服务器接到请求后，给予相应的响应信息，其格式为一个状态行包括信息的协议版本号、一个成功或错误的代码，后边是MIME信息包括服务器信息、实体信息和可能的内容。其实简单说就是任何服务器除了包括HTML文件以外，还有一个HTTP驻留程序，用于响应用户请求。你的浏览器是HTTP客户，向服务器发送请求，当浏览器中输入了一个开始文件或点击了一个超级链接时，浏览器就向服务器发送了HTTP请求，此请求被送往由IP地址指定的URL。驻留程序接收到请求，在进行必要的操作后回送所要求的文件。在这一过程中，在网络上发送和接收的数据已经被分成一个或多个数据包（packet），每个数据包包括：要传送的数据；控制信息，即告诉网络怎样处理数据包。TCP/IP决定了每个数据包的格式。

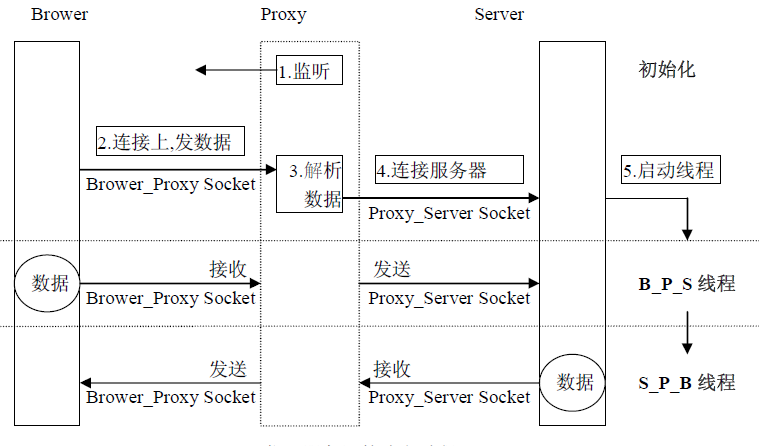


图2.2-4 HTTP工作原理示意图

## 系统应用层详细设计

**2.3.1静态图**

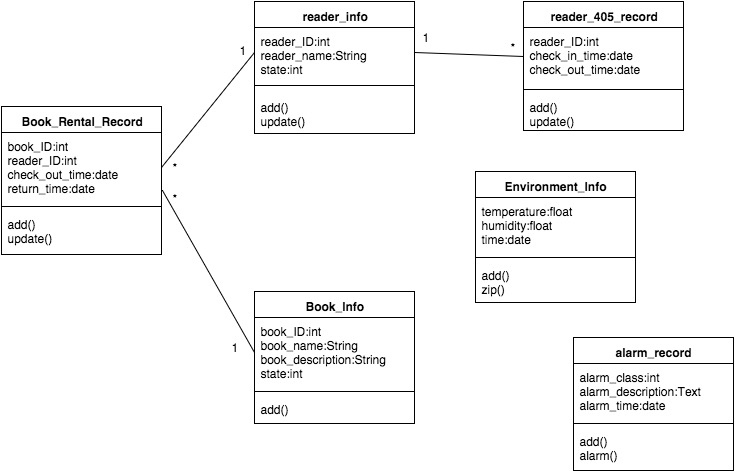


图2.3-1应用层类图

六个类：读者信息、书籍信息、405环境信息、书籍借阅记录、读者进出405记录、警报

**2.3.2动态图**



图2.3-2时序图



图2.3-3活动图



图2.3-4协作图



图2.3-5状态图