

# “时间管理者”系统设计报告

姓名：胡成成      学号：41724260

北京科技大学计通学院通信 1701 班

---

**摘要：**针对人们对时间观念的弱化，尤其是大学生对时间的荒废问题，本文设计了一套帮助他们管理时间的物联网系统——“时间管理者”。通过设置贴片形式的传感器节点，基于 Android 设计实现软件功能，与传感器节点交互，并通过基于 Nginx 搭建的与服务器提供相关同步服务，通过 SQL 语言设计关系型数据库，最终选择 ZigBee 通信方式进行软硬件的数据通信。来帮助大学生管理好时间，完成每天任务，充实每一天的生活。

**关键字：**传感器节点设计；Android；Nginx；SQL；ZigBee。

---

**Abstract:** Aiming at the weakening of people's conception of time, especially the problem of college students' lack of time, this paper designs a set of Internet of Things system to help them manage time--"time manager". By setting up sensor nodes in the form of patches, implementing software functions based on the Android design, interacting with the sensor nodes, and providing related synchronization services with the server based on Nginx, designing a relational database through SQL language, and finally choosing ZigBee communication method to achieve software and hardware Data communication. To help college students manage time, complete daily tasks, and enrich each day's life.

## 1. 引言

---

本文设计灵感源于生活，应用于生活。着重解决生活中遇到的痛点问题，通过物联网技术所学的知识来尝试解决问题，锻炼工程实践与系统设计的能力。

随着人们的生活水平不断提高，生活的方式越来越多样化，人们对时间的观念反而越来越薄弱。有句古话说得好“时间就是金钱”。现代人忙碌着怎么去赚钱，却忽略了时间这一宝贵的财富。本文就针对现在很多对时间的忽视，耗费大把时间荒废，没有规划很迷茫的人，特别是在校大学生，更应当规划好自己的时间，为自己的人生负责，为自己未来的路打好基础，收获丰硕的知识，以便在自己的行业做出一番成就。

因此，本文针对这一现象，将从物联网设计的角度思考问题，通过专业知识解决发现的问题。本文第一节对全文进行概括，第二节介绍系统的研究背景，第三节对系统架构进行设计，第四节进行技术方案实现与分析，第五节进行方案与工程分析，第六节进行总结与展望。

## 2. 研究背景

---

### 2.1 问题引入

---

随着人们的生活水平不断提高，生活的方式越来越多样化，人们对时间的观念反而越来越薄弱。常常出现一天好像什么也没干就过去了的现象，大把的时间浪费在没有意义的事情上，自己却没有及时发现。尤其是正处于青春年华的大学生，更是要对自己的生活有很好的规划。然而，很多大学生却没有这样的观念，在大学荒废了大把的时间，没学到什么东西，生活过的也不充实，但是反思自己却又发现不了自己的时间都花费在哪里了，久而久之，产生恶性循环，对时间管理的忽视导致对他们的大学生活甚至未来的发展产生很大的危害，例如：自制力下降，自信心缺失，懒惰等等坏习惯。针对这一现象，是否可以设计出一种物联网系统，通过铺设很多小小的贴片，粘在自己的日常生活用品上，通过设计一款手机软件与贴片进行通信，从而达到提醒自己在规定时间去完成自己的计划？例如：在自己的书本上粘上贴片，在专有的软件上为这个贴片设置一个标签(Tag)，每个标签就是一个任务，可命名为“看书”，可以在软件上设置该任务的时间，

比如从上午 9:00 到 10:00 看书，这样贴片会在 9:00 发出类似于“滴滴”的提醒音，提醒使用者完成“看书”的任务，同时贴片和手机开始计时，记录你实际完成任务的时间，并保存在软件内，以日历的形式记录每天每个时段的时间都在花在哪里，并通过数据处理的方法及时分析出自己每天、每周以及每月在每个任务上花费的时间占比，从而更好地了解自己的时间都花费在哪里了，便于下一步的决策与规划。

## 2.2 痛点分析

---

对于问题引入中的现象，可以看出，主要的校园生活痛点问题是大多数人忽视时间的管理，究其原因，主要包括：

- (1) 时间管理太麻烦，凭借传统的“手账”记录方式较繁琐，难以坚持
- (2) 自制力差，大量时间花费在娱乐方面，没有相应的提醒与措施
- (3) 对花费的时间没有总体的认知，不知道自己所干的事花费的时间比重
- (4) 过于依赖手机，手机上的诱惑信息很多，仅凭借手机提醒自己，很容易导致很多时间的浪费，例如：微博发送了一条推送内容，忍不住打开去看，从而导致计划的打乱与时间的浪费。
- (5) 过于迷茫，没有规划，没有生活规划的任务推荐
- (6) 办事注意力难以集中，效率低下

## 2.3 需求分析

---

根据痛点分析，针对这些问题的原因，提出相应的用户需求：

- (1) 需要简化任务的记录方式，构建任务实现的奖励机制
- (2) 对于娱乐任务，需要设置时间限制提醒，必要时需要强制关闭的程序
- (3) 需要时间数据的处理分析，生成时间管理报告
- (4) 将任务提醒机制尽可能从手机移到简单便宜的物联网设备上
- (5) 根据时间管理报告，通过智能算法推荐生活任务，帮助提升自我
- (6) 播放大自然的声音，设置任务倒计时，通过添加紧迫感让用户更专注

## 3. 架构设计

---

### 3.1 功能架构

对于系统的功能设计，从硬件和软件两个角度分别有不同的功能，以及软硬件交互的特殊功能等等，甚至有些需要进行云端处理的操作都要进行详细的架构设计。

#### 3.1.1 贴片功能性设计

首先贴片的物理特性应该具有小巧，便携，功耗低，可重复利用等特点。具体包括以下功能：

- (1) 可轻易吸附任何材质表面，不怕磕碰，有可识别编号
- (2) 可在未使用是自动休眠，降低功耗
- (3) 能够发出提示音，同步手机软件定位任务需要的工具
- (4) 能够计时，可通过按压贴片开启和关闭计时，并同步软件端控制
- (5) 有指示灯进行任务终止强提醒，对一些限时任务（玩游戏等）
- (6) 可通信，连接手机并接入网络将数据上传云端
- (7) 可定位，定位发出预设声音和指定颜色灯光

#### 3.1.2 软件功能设计

软件服务应与贴片相匹配，具有可控制，轻量级，简约化的特性。具体包括以下功能：

- (1) 可注册登录，通过云使得不同设备上登入的用户数据同步
- (2) 授权设备可与多个贴片通信，并可为不同贴片设置标签和信息，以及控制贴片状态
- (3) 可计时，可提醒（闹钟形式），可存储任务时间数据
- (4) 预设各类任务与自定义任务，可设置时段，并同步手机日历
- (5) 可视化统计时间数据，生成数据分析报告与图表
- (6) 给日常计划进行重要性排序，对重要性等级高的任务与计划提醒强度增加
- (7) 可根据时间管理分析报告智能提供一些任务建议

- (8) 预设一些音频，在完成一些需要专注度高的可开启“专注模式”，锁定手机，播放音乐开始任务

### 3.1.3 云端服务功能

云端的服务，主要包括数据备份，数据存储和数据计算等，包括：

- (1) 数据存储，存储用户时间数据，便于同步设备与数据的备份
- (2) 数据计算，通过一些智能算法，对时间数据处理分析，反馈给用户一些建议

## 3.2 软件架构

软件架构主要从基础组件层、通用业务组件层、业务层、App 壳这四层设计软件的基础架构。如图 1 所示。

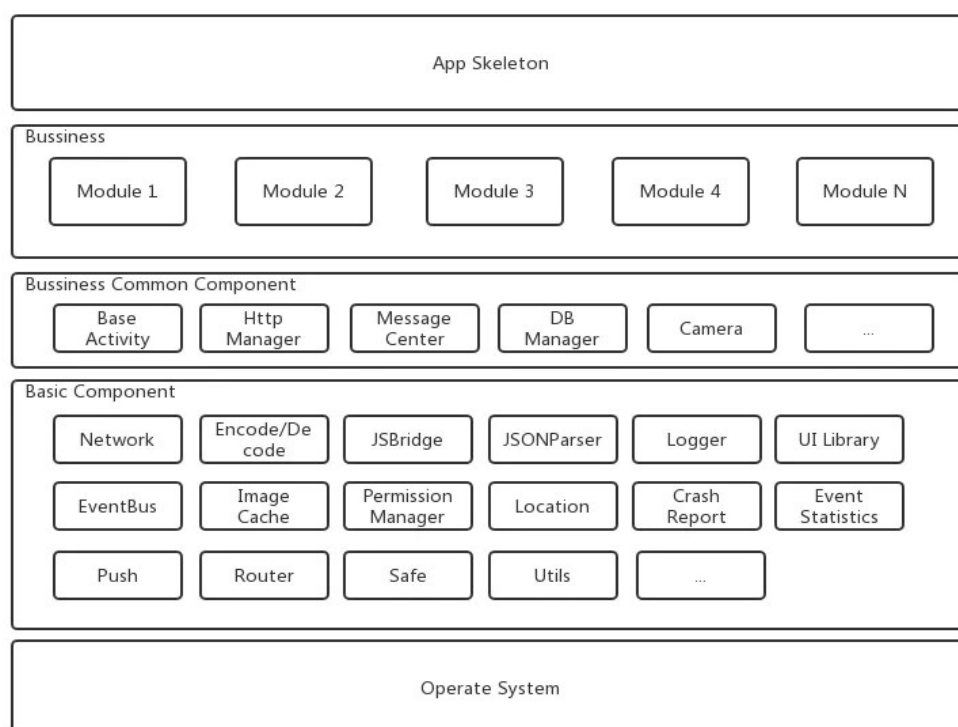


图 1 APP 架构设计图

- (1) 基础组件层：每个模块功能单一；不包含业务逻辑；模块互相独立，没有依赖关系；组件化，模块化良好，代码可复用程序高。
- (2) 通用业务组件层：对基础组件模块的封装；通用的业务逻辑下沉；通用的 App UI 相关组件；依赖基础组件层。

- (3) 业务层：根据业务及功能，划分多个相对独立业务模块；业务模块之间，减少互相依赖；业务模块互相调用，通过 Router 组件解耦；MVC/MVP/MVVM。
- (4) App 壳：App 入口，主界面，首屏；App 整体功能输出配置，构建；ToB 形态 App 的功能自定义组合，UI 风格自定义等需求

### 3.3 硬件架构

硬件架构主要包括控制系统，传感器和物理硬件设备的设计。如图 2 所示。

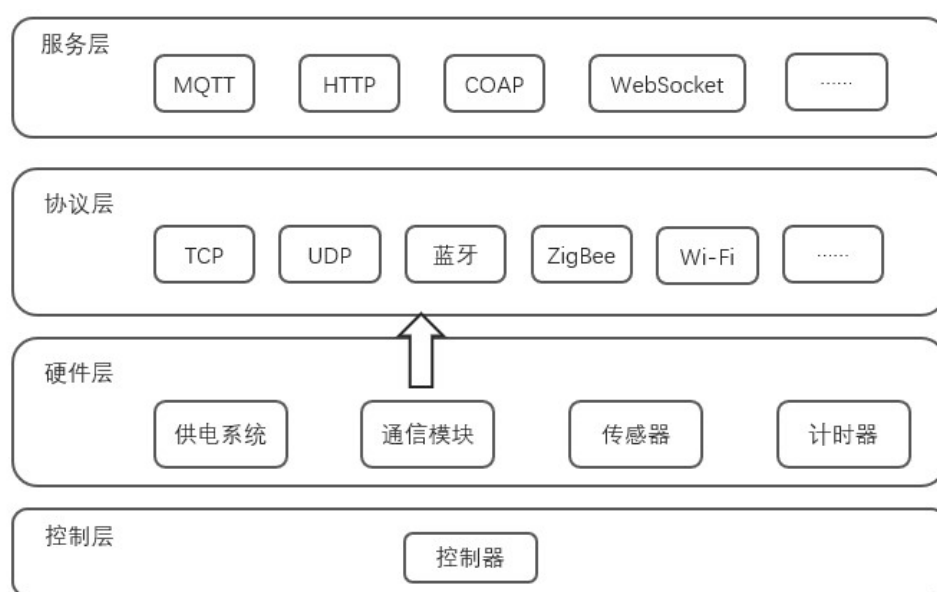


图 2 硬件传感器架构设计图

- (1) 控制层：提供系统逻辑控制，电路控制。
- (2) 硬件层：获取、发送传感器数据，执行指令。
- (3) 协议层：协调硬件层与服务器的通信，并负责处理部分数据。
- (4) 服务层：提供网络连接，和相关服务。

## 4. 技术方案

### 4.1 硬件实现方案

根据传感网的基本单元结构——传感器节点的组成，如图 3 所示。

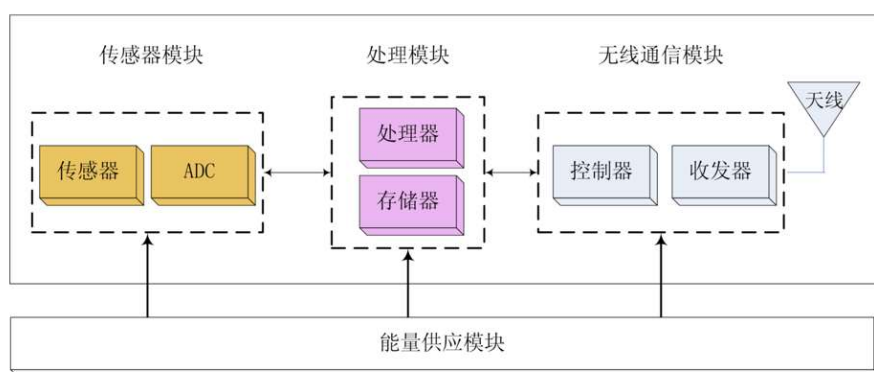


图 3 传感器节点组成

一般的大规模长时间部署传感器的设计需求包括：低成本与微型化，低功耗，高可靠性（鲁棒、健壮、可生存），灵活性和可扩展性的特点。

#### 4.1.1 传感器模块

随着科学技术的发展，目前市面上的传感器种类丰富，性能优越，功耗低。传感器制造厂商也十分多。由于我们要设计的系统要求体积轻小，目前生产微处理器的国内厂商有歌尔声学、上海威尔泰、杭州士兰、深圳鑫精诚科技等。鉴于贴片需要的传感器精度和要求并不需要过高，在性能上追求低功耗，体积薄且小，在选择传感器上可以降低精度的追求。

根据要设计的物联网系统的需求来看，需要使用的传感器包括：

- （1）微型发声装置：产生提示音
- （2）压力传感器：感知压力，启动/关闭计时器计时
- （3）晶振计时器：任务计时，启动中断程序
- （4）微型可变色 LED 灯：指示与提醒作用

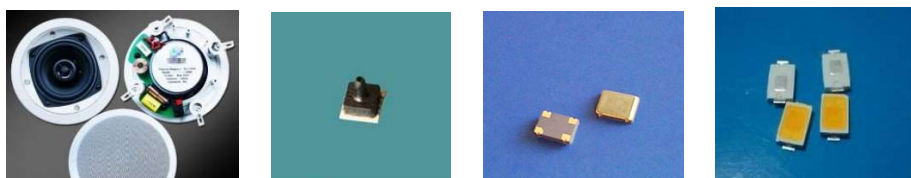


图 4 需求传感器示意图

对于无线传感器的工作模式设计如下：

- （1）睡眠模式：各节点敏感功能关闭，无线功能关闭，节点上的微处理器模块近似处于停机状态，保持极低能耗，在一个较长唤醒周期后自主



唤醒,打开无线接收功能,侦听信道中是否有含自身 ID 号的唤醒指令,如没有则关闭无线接收功能,再次睡眠等待下一个唤醒周期;如侦听到唤醒指令,则进入待机模式。

- (2) 待机模式:各传感器采集节点在被唤醒后,敏感功能仍关闭,节点上的微处理器模块开始工作,无线功能按唤醒要求打开,各传感器采集节点准备接收由数据接收控制器发送的周期广播信号。
- (3) 工作模式:数据接收控制器上电初始化完成后,自动周期发送广播信号,使各传感器采集点进入工作模式。在此模式下,各传感器采集节点按照预定设定的时间依次打开敏感和无线功能,向无线数据接收控制器发送数据。在工作模式下,当传感器采集节点未能接收到数据控制器发送的周期广播信号,则返回到待机模式。

#### 4.1.2 处理模块

微处理器是无线传感器节点中负责计算的核心。目前的微处理器芯片同时也集成了内存、闪存、模/数转换器、数字 IO 等。而影响节点工作整体性能的关键特性包括:功耗特性,唤醒时间,供电电压,运算速度,内存大小。著名的微处理器厂商 Atmel, Ember, TI 等公司。

而对于处理器的选型,选择够用就好的原则,在满足实现功能的条件下,需要考虑系统的低功耗和小型化设计要求,主要依据功耗,调制方式,通讯距离以及传输速率等因素。处理器的功耗主要由工作时需要的工作电压、时钟频率、制作工艺决定,所以在系统不工作时,处理器需要支持睡眠模式。实验表明,当处理器处于睡眠状态时,处理器对能量的损耗比在工作状态时低 3-5 个数量级。<sup>[1]</sup>

#### 4.1.3 无线通信模块

无线通讯模块的功耗占无线传感器网络功耗的大部分。通信模块只有在睡眠状态才会关闭,在待机状态也同样需要监测信号通道,确认是否有数据传送过来。而在无线通讯模块处于发送状态时功耗是最大的,待机状态与接收状态的功耗基本一样。所以为了减少功耗,在不必要的接收与发送时,尽量使无线通讯模块处于睡眠状态,减少不必要的监听,这样可以降低网络节点的功耗。但简单地关闭无线通讯模块会造成数据的延时,实时性较差。无线通讯的能量损耗与距离的关



系为 $E = kd^n$ ，通常  $n=3$ ，即能耗与距离的 3 次方成正比。<sup>[2]</sup>因此在满足通信速率的前提下，应尽量减少通信距离。

#### 4.1.4 能量供应模块

有效的电源管理设计可以明显降低无线传感器的功耗。一方面可以从元器件的选型入手，尽可能选择符合使用要求，同时低功耗的元器件；另一方面在单机工作模式的设计上，做到电源管理的精确化、智能化，在需要的时间段供电，只给需要的功能模块供电。

实现低功耗电源管理的主要措施：

- (1) 通过处理器的 I/O 端口电压稳压后为敏感元件供电，这样可以达到供电可控。当不需要采样和发送数据时，处理器进入低功耗模式，同时控制 I/O 端口关闭电压输出，切断给敏感元件的供电，从而有效地控制了电能的消耗，即在不需要工作的时候，断开电源，起到节能目的。
- (2) 通过程序的控制，使处理器在工作模式、睡眠模式、低功耗运行模式之间进行合理切换，只有在信号发射时，才需要提供较大的电流输出给电路。

#### 4.1.5 最终可实践方案

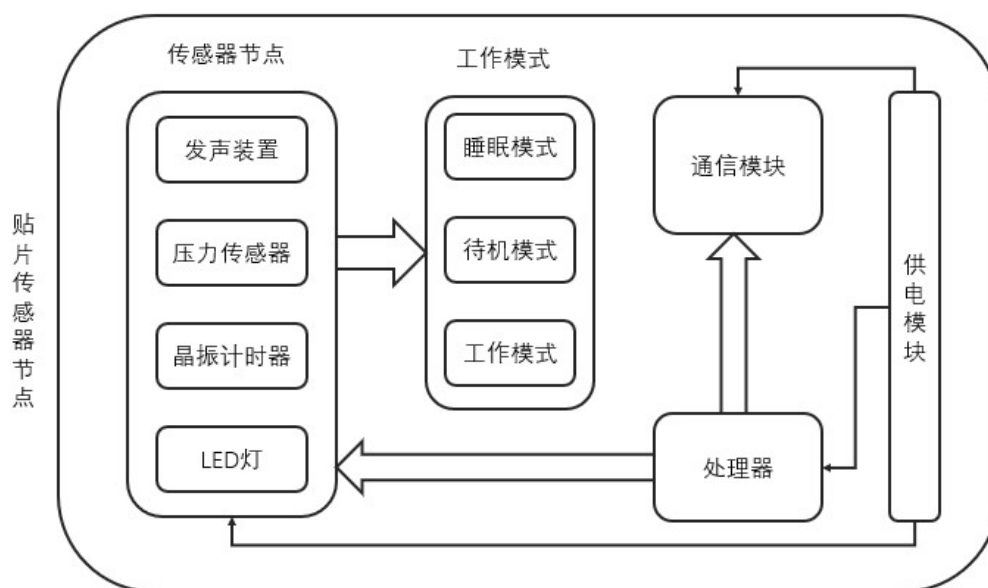


图 5 最终传感器节点设置方案

## 4.2 软件实施方案

### 4.2.1 操作系统

目前应用于移动手机端的主流操作系统有 Android 和 IOS，鉴于个人使用的是安卓手机且有安卓开发经历，软件的设计基于 Android 操作系统，开发工具可选择 Android Studio。

### 4.2.2 基础组件与通用业务组件

**基础组件：**根据 App 的需求功能可以看出，需要页面，存储，网络，通知，服务，缓存，安全等多个基础组件的使用。通过调用 Android SDK 中的相应的 Java 类包可编程实现基础组件的设计与开发。

- 页面：Activity 是 Android 展示给用户的 UI 容器，负责与用户直接交互的组件。
- 存储：Android 存储方式有五种，文件存储，SharedPreferences（存少量格式非常简单的数据），SQLite 数据库存储，ContentProvider（用于不同的程序之间实现数据共享），网络存储。
- 网络：Android 主要的网络通信框架有 okhttp, retrofit, android-async-http, volley。
- 通知：Android 通知组件 Notification 可提供手机通知栏的通知功能。
- 服务：Service 是一个长期运行在后台，没有用户界面的应用组件，即使切换到另一个应用程序或者后台，服务也可以正常运行。因此，服务适合执行一段时间不需要显示界面的后台耗时操作（需要另启子线程）。
- 缓存：Android 缓存分为内存缓存和文件缓存（磁盘缓存）。包括二级缓存机制。
- 安全：通信采用 https 协议进行的网络请求，使用签名和加密数据。

**通用业务组件：**除了基础组件外，基于这些组件针对不同的软件应用需求需要提供相应的业务组件，针对本文设计的系统，主要包括如表 1 的通用业务组件组件。

表格 1 通用业务组件

组件	Java 类包（库）
闹钟服务	AlarmManager
日历组件	CalendarView
数据可视化	MPAndroidChart（第三方开源库）
蓝牙	Bluetooth

### 4.2.3 UI 设计

在概念性产品中，UI 为其中较为突出的一种，应用 UI 的主要目的为实现机器与人的无障碍交流。UI 设计的流程如下<sup>[3]</sup>：

- （1）了解用户需求，提高用户认知度
- （2）利用视觉设计完成设计研究
- （3）完善设计细节

具体的 UI 设计根据产品的应用场景，应该有以下特点：

- （1）界面设计简单清新，核心功能突出
- （2）色彩不花哨，不分散用户注意力
- （3）记录效果突出，任务是否完成的标记和完成时间重点标注

## 4.3 服务实现方案

服务实现主要是云端数据的存储、操作以及查询，因此主要是对使用的数据库和服务器软件的选择与设计。

### 4.3.1 服务器选择

对于服务器主要是使用云服务器，而对于服务器软件，目前主流的服务器软件有：

- **Apache**：世界使用排名第一的 Web 服务器软件。Apache 的特点是简单、速度快、性能稳定，并可做代理服务器来使用。适合处理 php 页面。
- **IIS（Internet Information Server）**：微软公司主推的服务器，能够利用 Windows Server 和 NTFS 内置的安全特性，建立强大，灵活而安全的 Internet 和 Intranet 站点。

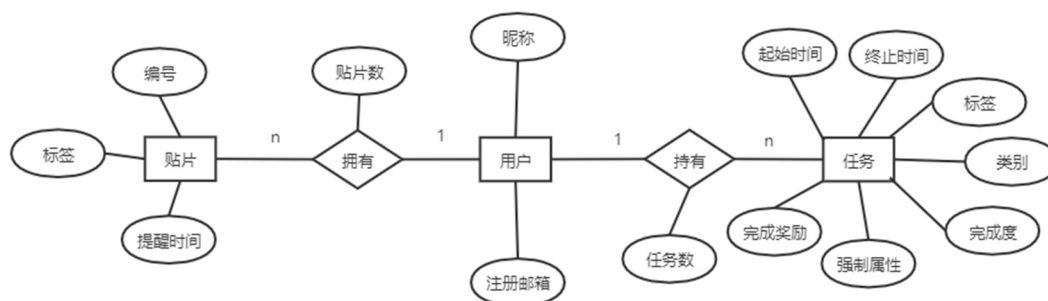
- **Nginx:** 不仅是一个小巧且高效的 HTTP 服务器，也可以做一个高效的负载均衡反向代理，通过它接受用户的请求并分发到多个 Mongrel 进程可以极大提高 Rails 应用的并发能力。
- **Lighttpd:** 基于 BSD 许可的开源 WEB 服务器软件，其根本的目的是提供一个专门针对高性能网站，安全、快速、兼容性好并且灵活的 web server 环境。具有非常低的内存开销，CPU 占用率低，效能好，以及丰富的模块等特点。

根据我们的服务资源大多是静态资源，针对服务器软件权衡，选择 Nginx 最合适。主要是 Nginx 包括稳定性、丰富的功能集、示例配置文件和低系统资源的消耗等强大的性能，适合静态内容多的服务。

#### 4.3.2 数据库设计

对于数据库的设计，针对用户和其可能产生的数据进行分析，使用关系型数据库设计，通过数据库设计基本流程设计该物联网系统的数据库。

##### (1) ER 图设计:



##### (2) 关系模式:

表格 2 用户实体关系模式

字段	命名	数据类型
昵称	User_name	Varchar(255)
注册邮箱	User_email	Varchar(50)
拥有贴片数	Patch_num	int
持有任务数	Task_num	int

表格 3 贴片实体关系模式

字段	名称	数据类型
编号	Patch_id	int
标签	Patch_tag	Varchar(255)
提醒时间	Patch_in_time	Datetime

表格 4 任务实体关系模式

字段	名称	数据类型
标签	Task_tag	Varchar(255)
类别	Task_class	Varchar(255)
起始时间	Task_in_time	Datetime
终止时间	Task_out_time	Datetime
完成度	Task_done	Boolean
强制属性	Task_must	Boolean
完成奖励	Task_done_reward	Varchar(255)

(3) 数据库实现：通过 SQL Server 2008 实现上述数据库。

#### 4.3.3 推荐算法的实现

通过设计推荐算法实现对用户的任务推荐，帮助用户合理分配时间。目前的推荐算法主要包括<sup>[4]</sup>：

- (1) 基于协同过滤的个性化推荐
- (2) 基于图结构的个性化推荐
- (3) 基于社交网络的个性化推荐

## 5. 方案与工程分析

### 5.1 通信方案分析

根据本文设计的物联网系统，拟采用的是 ZigBee、Bluetooth 或 Wi-Fi 的通信方式。下面对比这两种通信方式的特点进行择优选择。

表格 5 拟采用通信方式对比样表

属性	ZigBee	Bluetooth	Wi-Fi
传输速率	100Kbps	1Mbps	11-54Mbps
通讯距离	2-20m	20-200m	20-200m
频段	2.4GHz	2.4GHz	2.4GHz
安全性	中等	高	低
功耗	5mA	20mA	10-50mA
成本	5\$	2-5\$	25\$

通过对比, 综合考虑各项指标, ZigBee 通信方式更适合大量使用传感器节点的使用, 与本文设计的传感器系统的需求更匹配。

## 5.2 数据库方案分析

本文采用关系型数据库设计, 主要是由于传感器数据关系并不复杂, 方便查询, 数据量相比传统的数据采集型传感器并不是很大, 采用正常的 SQL 数据可设计即可, 并不需要像 Redis 这样的非关系型, 易处理高并发的数据库来处理。

因此, 本文的数据库采用 SQL 关系型数据库建立, 基本满足设计需求。

## 5.3 服务器方案分析

本文采用 Nginx 搭建服务器, 主要是由以下几点原因:

- (1) Nginx 的中间件架构, 是开源, 高性能、可靠的、代理服务中间件。
- (2) Nginx 轻量级, 功能模块少, 代码模块化。
- (3) Nginx 的 CPU 亲和。
- (4) Nginx 的 sendfile 工作机制。

## 5.4 工程分析

综合整个系统的设计过程, 从工程的角度来看:

- (1) 首先, 需要考虑实际的设计成本, 例如本文所需要的成本主要来源于贴片的硬件成本和云服务器的租赁成本, 设计需要的贴片数量多, 实际分摊在每个贴片上的成本完全可以接受;

- (2) 其次，需要考虑项目带来的效益，本项目主要通过贴片于手机软件的结合，达到实现时间系统管理的目的，让人们更好地规划最记得人生，尤其是针对在校大学生，可见系统带来的实际效益还是十分高的，具有一定的现实意义；
- (3) 最后，需要考虑系统的安全性，整个系统在硬件上并不存在高危电子设备，在软件上来自用户地私密信息并不多，仅仅是注册邮箱和一些时间数据，不存在用户隐私安全地问题。

## 6. 项目总结

---

通过对“时间管理者”物联网系统的设计，对物联网技术课程所学的知识应用到实际的调研与研究开发中，通过自主设计，对物联网技术中传感器技术，无线通信技术，数据库系统，云计算与云存储等进一步深入了解。

本次设计过程中也存在一些问题待后续学习改进：

- (1) 并未实际实现该系统，所有设计都是基于理论，待后续实验验证可行性。
- (2) 设计中由于对市场适合该系统的小型芯片不了解，并未实际选取相应型号的控制系統。
- (3) 对具体的编程细节和 Bug 调试有待优化调整。

## 7. 设计工具

---

- 编辑工具：Word2019
- 作图工具：PowerPoint2019 & ProcessOn
- 搜索平台：Baidu & CNKI
- 编程工具：Android Studio
- 数据库工具：SQL Server 2008
- 服务器工具：Nginx
- 算法实现工具：Python



## 8. 参考文献

---

- [1] 王晓莲.无线传感器网络低功耗设计[J].自动化技术与应用,2017,36(05):144-146.
- [2] 涂巧玲,张杰,潘建权,曾宪建.无线传感器网络节点低功耗设计策略[J].电子测量技术,2009,31(01):158-161.
- [3] 李丹骏.UI 交互设计在信息科技中的探究[J].戏剧之家,2020(08):229.
- [4] 陈豪,王泽珺.个性化推荐算法综述[J].企业科技与发展,2019(02):56-57.