

# 陇东学院



毕业论文（设计）

## 基于物联网STM32环境下的智能驾 校管理系统设计与实现

学 院 信息工程学院  
年 级 2016 级  
专 业 物联网工程  
学 号 2016631124  
姓 名 王富国  
指导教师 李芳芳  
完成日期

**批注 [11]:** 1. 论文题目过大，简单修改了一下，你再推敲推敲，物联网是一个很大的概念，论文题目要体现我们主要应用的技术和这项技术解决的问题。

**批注 [12]:** 论文格式只需要小幅度的修改。1. 主要参考文献部分，部分参考文献不完整，请按照格式要求补充完整。2. 表格里边的字号，有些是5号，有些是小5号，不统一。3. 不清晰的图要进行编辑，保证打印出来是可读的。4. 代码部分也要调整。

**批注 [13]:** 你论文的重复率为15.1%，符合本科论文的要求。

**批注 [14]:** 2 论文结构，看起来不太完整。从目录看你只有三个部分，我从目录给你批注了个参考的提纲，可能不太符合你的论文，可参考一下，调整文论结构。是论文层次更分明，更清晰，使论文结构上完整。



## 作者声明

本毕业论文（设计）是在导师的指导下由本人独立完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。对本论文（设计）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。因本毕业论文（设计）引起的法律结果完全由本人承担。

毕业论文（设计）成果归陇东学院所有。

特此声明。

作者专业：物联网工程

作者学号：2016631124

作者签名：

20 年 月 日（手填时间）

## 目 录

摘 要 .....	- 3 -
关键词 .....	- 3 -
Abstract .....	- 4 -
Key words .....	- 4 -
1 绪论 .....	- 5 -
1.1 研究背景及意义 .....	- 5 -
1.2 国内外的研究现状及发展趋势 .....	- 6 -
1.3 研究本课题的目的和基本内容 .....	- 7 -
2 系统总体设计方案 .....	- 8 -
2.1 系统需求分析 .....	- 8 -
2.2 系统总体架构设计 .....	- 9 -
3 系统整体实现方案 .....	- 14 -
3.1 使用版本控制器构建项目仓库 .....	- 14 -
3.2 数据持久层实现 .....	- 19 -
3.3 物联网模块实现 .....	- 26 -
3.4 Web 模块实现 .....	- 33 -
主要参考文献 .....	- 36 -
附录 1 数据库 E-R 关系图 .....	- 37 -
附录 2 主要英文缩写语对照表 .....	- 38 -

批注 [15]: 第 1 章绪论  
1.1 研究背景及意义  
1.2 研究现状  
1.3 主要研究内容  
1.4 论文结构  
第 2 章系统需求分析  
2.1 业务需求概述  
2.2 XX 管理  
2.2.1 XX 业务需求  
2.2.2 XX 功能需求与建模  
2.3 YY 管理  
2.3.1 YY 业务需求  
2.3.2 YY 功能需求与建模  
2.4 系统管理  
2.4.1 系统管理业务需求  
2.4.2 系统管理功能需求与建模  
2.5 本章小结  
第 3 章系统总体设计  
3.1 系统总体模块设计  
3.2 系统逻辑架构设计  
3.3 系统网络结构设计  
3.3.1 网络设计  
3.3.2 网络接入设计  
3.4 系统模块设计  
3.4.1 XX 业务管理模块设计  
3.4.2 YY 业务管理模块设计  
3.4.4 系统管理模块设计  
3.5 系统异常处理设计  
3.6 本章小结  
第 4 章系统数据库设计  
4.1 数据库总体设计...  
4.x 本章小结  
第 5 章系统详细设计与实现  
5.1 XX 管理模块  
5.1.1 活动过程分析  
5.1.2 对象交互过程分析  
5.1.3 XX 管理类图  
5.1.4 XX 管理功能实现  
5.2 YY 管理模块  
5.2.1 活动过程分析  
5.2.2 对象交互过程分析  
5.2.3 YY 管理类图  
5.2.4 YY 管理功能实现  
5.4 系统管理模块  
5.4.1 系统管理类图  
5.4.2 系统管理功能实现  
5.5 本章小结  
第 6 章部署与应用  
6.1 系统部署环境  
6.1.1 硬件环境  
6.2.2 软件环境  
6.2 系统测试  
6.3 系统应用

陇东学院 2016 届毕业论文（设计）

# 基于物联网环境的智能驾校管理系统

王富国

*Intelligent driving school management system based on IoT  
environment*

Wang Fu guo

2020 年 2 月 27 日

## 摘 要

现代社会是基于互联网的时代，从当前物联网横行的社会潮流来看，这是一个永恒不变的话题。驾照已经成为了我们人生中必不可少的证件之一，是我们安全出行的保证。随着“考证”人数的越来越多，驾校遍布，种类繁多，业务杂乱，驾校的人员管理越来越乱。目前主流的驾校管理系统在一定程度上减轻了人力的负担，但是还存在很多问题，例如预约练车系统容易崩溃、系统页面卡死，打卡定位计时不准确，学员信息存储不够安全，系统可维护性不高等，仅仅依靠 Web 端单一的管理方式，很难均衡。

本文基于物联网平台与 B/S 架构相结合设计了管理系统，有效降低了驾校人车不能互联的弊端，系统同时基于 STM32 开发板，通过 FTID 技术实现了练车打卡等功能，通过数据分析处理，实现了人卡合一，再次提升了集中统一化管理的效率，同时有效避免了学员间因为练车时间冲突等问题。本系统在驾校智能一体化管理方面起到了明显的作用，由于物联网范畴强大，可根据需要随时做软硬件结合的拓展。

**关键词：** 物联网；STM32；B/S；驾校管理；Web；RFID

## Abstract

Modern society is based on the era of the Internet. From the current social trend of the Internet of Things, this is an eternal topic. The driver's license has become one of the indispensable documents in our lives and a guarantee for our safe travel. With the increasing number of "examinations", driving schools are spreading, there are many types, business is chaotic, and the personnel management of driving schools is becoming more and more chaotic. The current mainstream driving school management system reduces the burden of manpower to a certain extent, but there are still many problems, such as the easy breakdown of the car training system, the system page stuck, the timing of the punch card positioning is not accurate, the student information storage is not secure, and the system can be maintained Sex is not high, it is difficult to balance just relying on a single management method on the Web side.

This article designs a management system based on the combination of the Internet of Things platform and the B / S architecture, which effectively reduces the disadvantages of driving schools and vehicles that cannot be interconnected. The system is also based on the STM32 development board and uses FTID technology to implement functions such as car driving and punching. Through data analysis and processing It realized the integration of people and cards, and once again improved the efficiency of centralized and unified management, while effectively avoiding problems such as conflicts between trainees and trainees. This system has played a significant role in the intelligent integrated management of driving schools. Due to the strong scope of the Internet of Things, the combination of software and hardware can be expanded at any time as needed.

**Key words:** Internet of things; STM32; B / S; driving school management; Web; RFID

## 1 绪论

人工智能，简称 AI，是指人类制造出来的机器表现出学习，演绎、推理、解决问题等的智能<sup>[1]</sup>。物联网在一定层次上起到了万物互联，通过硬件传感器设备收集数据的作用。人工智能与物联网是相辅相成的。在物联网大数据的支撑下，人工智能才能更加的智能便捷化。本课题研究的管理系统，在物联网的加持下，将进一步人性化驾校的管理。

### 1.1 概念解析

#### 1.1.1 物联网概述

物联网在维基百科中是这么定义的：物联网（The Internet of Things，缩写 IOT），又称 IOT 技术，是互联网、传统电信网等的信息载体，让所有能行使独立功能的普通物体实现互联互通的网络。物联网将现实世界数字化，一体化，万物互联，应用十分广泛。物联网拉近了世界万物间的距离，统整物与物之间分散的信息，主要应用领域有以下方面：智能环境（家居、办公、工厂）领域、物流运输领域、工业制造领域、医疗健康领域、个人以及社会公共安全领域等。

#### 1.1.2 STM32

#### 1.1.3 Java EE 体系概述

Java EE，Java 平台企业版，原名 J2EE，2018 年 3 月更名为 Jakarta EE。它是在 SUN 公司的领导下，多家公司参与共同制定的企业级分布式应用程序开发规范。是目前世界上主流的分布式应用平台解决方案。

该平台包含有 JDBC（Java 数据库连接）、Servlet（Java Servlet API）、WS（Web Service）、JTA（Java 事务 API）等相关解决方案。

应用程序架构包含 B/S 和 C/S 两种模式。C/S 是指客户端/浏览器模式，程序开发人员既要开发服务器端，还要开发客户端，客户端和服务端分开运行，用户要使用服务必须要安装由服务商提供的客户端应用程序。B/S 架构是指浏览器/服务器模式，开发人员只需开发服务器端程序，用户借助 IE、Chrome 等浏览器即可使用服务商提供的服务。目前，H5<sup>①</sup>标准的发布，使得 H5 应用更为广泛，微信小程序与 H5 的融合，使得 Web 应用程序的门槛更加的低，基于这种的架构的应用程序也更加的深受大众欢迎。

#### 1.1.1.2 研究背景及意义

驾校，全称驾驶人训练班或驾驶学校，英文名称：Driving School，是目前世界各国为机动车驾驶人提供的统一培训和教授练习驾驶技术的场所。驾校的开设现象，在

<sup>①</sup> H5，即 HTML5，是 HTML 的最新修订版本。

**批注 [16]:** 如果你有多个概念要介绍，就单独列为一个标题。此处建议弱化物联网和 javace，主要介绍 stm32，因为他才是你论文的核心和灵魂。写背景意义和研究现状时，要写目前这项关键技术的应用。

**带格式的：** 论文-二级标题，缩进：首行缩进： 0 厘米

**批注 [17]:** 物联网包括的内容太大了，它是一个学科，你论文主要是基于 stm32，建议把论文题目改一下，缩小范围到具体的某项技术。所以在这个地方应该着重讲解 stm32 及你论文中实现系统功能的关键的物联网技术。

**带格式的：** 正文，缩进：首行缩进： 0 厘米



我国普遍流行，随着大众化消费水平的提高，国家经济的快速发展，机动车产业迅速增长，在“人人有车”的社会前景下，驾校行业迅速扩大，服务内容也有所拓展，培训形式多种多样，最近几年兴起的练车预约、模拟驾考尤为流行，这些相对智能设备的辅助，让培训人在练车、学习等方面轻松了不少，也减轻了驾校的人力资源。

相对智能化的设备，例如语音模拟练车系统、灯光语音模拟考试系统等，相比传统纯人工监督的方式，更加的先进、便捷、智能化，然而随着人工智能的进一步发展，扩大物联网的范围，提供更多的数据支持，才能更好地利用互联网为社会造福。

#### 1.1.1 物联网概述

物联网在维基百科中是这么定义的：物联网（The Internet of Things，缩写 IOT），又称 IOT 技术，是互联网、传统电信网等的信息载体，让所有能行使独立功能的普通物体实现互联互通的网络。物联网将现实世界数字化，一体化，万物互联，应用十分广泛。物联网拉近了世界万物间的距离，统整物与物之间分散的信息，主要应用领域有以下方面：智能环境（家居、办公、工厂）领域、物流运输领域、工业制造领域、医疗健康领域、个人以及社会公共安全领域等。

#### 1.1.2 Java EE 体系概述

Java EE，Java 平台企业版，原名 J2EE，2018 年 3 月更名为 Jakarta EE。它是在 SUN 公司的领导下，多家公司参与共同制定的企业级分布式应用程序开发规范。是目前世界上主流的分布式应用平台解决方案。

该平台包含有 JDBC（Java 数据库连接）、Servlet（Java Servlet API）、WS（Web Service）、JTA（Java 事务 API）等相关解决方案。

应用程序架构包含 B/S 和 C/S 两种模式。C/S 是指客户端/浏览器模式，程序开发人员既要开发服务器端，还要开发客户端，客户端和服务端分开运行，用户要使用服务必须要安装由服务商提供的客户端应用程序。B/S 架构是指浏览器/服务器模式，开发人员只需开发服务器端程序，用户借助 IE、Chrome 等浏览器即可使用服务商提供的服务。目前，H5<sup>①</sup>标准的发布，使得 H5 应用更为广泛，微信小程序与 H5 的融合，使得 Web 应用程序的门槛更加的低，基于这种的架构的应用程序也更加的深受大众欢迎。

#### 1.2 1.3 国内外的研究现状及发展趋势

驾校管理系统，纵观国内外，属我国驾驶人培训行业形式最为严峻，我国交通管理条例全面，对驾驶人的要求严格，目前市面上的驾校管理系统，都只是在纯软件端对驾校的人员信息数据做了汇总，而且采用是陈旧的技术，例如 JSP 服务端页面技术，存在系统可移植性差，维护性不高，响应速度过慢等问题。很少有能将驾校硬件资源接入管理系统的先例，比如，在教练车上装有传感器，检测学员练车时的车速问

① H5，即 HTML5，是 HTML 的最新修订版本。

题，可以有效的保障学员的生命安全。

通过查询文献资料以及互联网上的资料分析，目前所有的驾校管理平台都是单一的网络服务预约系统<sup>[2]</sup>。伴随着互联网+概念的提出，为了实现网络信息办公化，也已经诞生了主流 SSM 框架搭建的综合化驾校业务管理平台<sup>[3]</sup>，充分实现了网络服务替代人工劳动，在减轻驾校成本，数据信息存储等方面取得了重大的进步。但目前驾校综合业务的拓展和系统的可维护性依然停留在 B/S 架构，单纯网络服务的领域，并没有应用到物联网的环境下，而且 Web 页面采用的是 JSP 技术，这种服务器端的页面框架已经成为过去式，对服务器的负荷重，维护成本高，目前 H5 已经占领了潮流，应用广泛，前后端分离开发模式已经占据了企业级应用的顶端。目前物联网正是火热时期，借助物联网平台，再来研究驾校的综合管理，使用软硬件结合的方式，将在一体化办公，智能服务的领域更进一步。相信不久的将来，随着物联网技术的进一步发展，驾校的综合化、一体化管理服务将会变得更加人性、更加便捷。

本文设计的驾校管理系统采用物联网硬件和 Web 网络服务平台相结合的方式，大大增强了系统的可拓展性。硬件收集数据，软件处理逻辑，软硬件结合的方式更加接近当今社会物联网时代的需求。物联网发展迅速，采用软硬件结合的方式，实现车联网、人联网，甚至万物互联，指日可待。

#### 1.3.1.4 研究本课题的目的和基本要求

本课题研究的智能化驾校管理系统，主要采用物联网平台与 Web 网络服务平台相结合的方式，在解决驾校日常运转各项业务，提高员工工作效率的前提下，更好的为广大的驾校管理者服务。本课题主要研究在 STM32 开发板环境下，RFID 射频识别技术采集数据，经过 ESP8266 网络设备通过 HTTP 协议将数据转发给 Web 服务器进行数据分析处理，响应给 ESP8266 设备，再由 STM32 芯片处理数据，展示在其它硬件设施（LED 灯光、蜂鸣器等）上。

本文系统中物联网平台基本内容如下：

- （1）采用 C 语言编码，驱动 STM32 芯片指挥各个硬件模块合理化工作。
- （2）通过 RC522 型号的 RFID 模块完成读写卡工作，来鉴别管理员和学员的身份。
- （3）使用 ESP8266 型号的网络模块将物联网平台的数据通过 HTTP 协议转发到 Web 服务器端存储
- （4）合理化分配开发板上的其余硬件设施(LED 灯，按键)配合主要模块完成功能交互。
- （5）本文系统中 Web 网络平台的基本内容如下：
- （6）服务器端采用 Java 语言的主流 Web 框架，Spring Boot 搭建。
- （7）前后台分离式开发部署，后端使用 Spring Boot、Spring MVC 和 MyBatis 整合（简称 SSM 框架），前端使用数据驱动形框架 Vue 开发。

批注 [18]: 下边应该重新编号。

- （8）服务器端使用 maven 一键构建、打包和部署。
- （9）前端使用 nginx 或者 apache 服务器做分离式部署。

## 2 系统总体设计方案

### 2.1 系统需求分析

该系统主要包括由 C 语言编写的物联网平台、Java 语言编写的 Web 服务端和由前端 vue 框架搭建的 Web 网页端，还需要数据库来存储系统数据。

驾校是一个服务型机构，主要为培训人提供培训、考试服务。具体业务流程如图 1 图 1 所示。

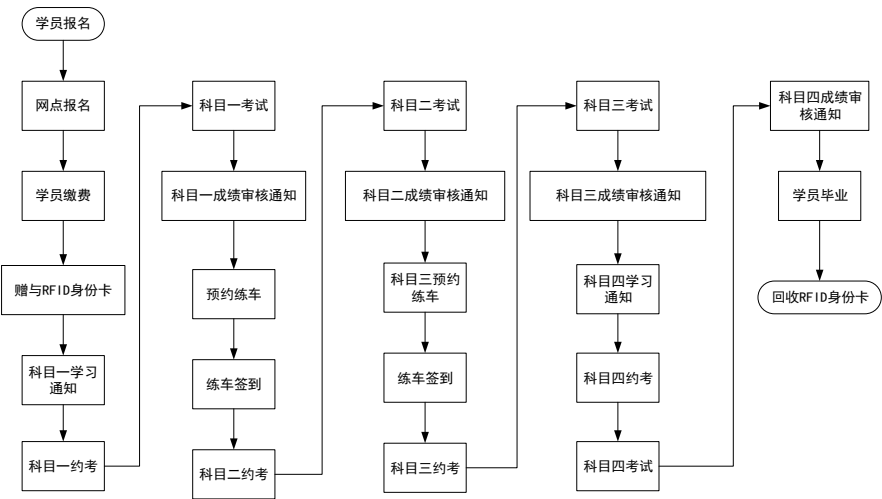


图 1 业务流程图

流程图中展示的大部分驾校数据信息处理的业务，在目前的各类驾校管理系统中已经较为成熟，且本论文实现的处理方式也基本类似，本文主要讨论驾校数据管理业务结合物联网环境平台的应用，根据这个初衷，需要研究以下内容：

（1）任何一套系统都离不开数据的支持，在现有驾校管理系统的基础上，融入物联网平台的内容，需要考虑物联网平台与 Web 服务器数据的衔接方式。本论文将使用开源免费的 MySQL 关系型数据库，通过构建物理模型来展示数据的交互过程，在下一章节将会详细介绍。

（2）本论文采用了物联网平台，需要用到开发板芯片处理数据，RFID 模块采集数据，ESP8266 模块传递数据等，具体的实现流程将会在系统硬件设计章节具体阐述。

（3）本系统的数据处理以及展示都采用了目前企业级应用开发的前后端分离架构处理，前端展示，后端处理，具体的业务实现需要大量的代码来完成，主要实现细

节，将在后续章节中提到。

（4）本论文研究的系统涉及物联网平台、Web 服务器端，Web 数据展示端等，本设计从项目搭建开始就将采用分布式版本控制工具 Git 来管理项目文件。物联网作为实践类科目，大部分功能都需要通过代码来实现<sup>[4]</sup>。本论文的重点在于专注物联网的应用，以及平台要实现的功能，本项目代码需求量大，耗时周期长，在项目搭建初始，就要考虑到代码的版本的迭代，以及代码的维护、拓展等，由于代码维护工作量太大，为了专注于功能的实现，Git 分布式版本控制工具就担当了代码维护的重要角色。

## 2.2 系统总体架构设计

### 2.2.1 数据建模和数据库 E-R 图

数据建模是对现实世界中的各类数据抽象的组织起来，确定数据库需要管辖的数据范围和对数据的组织形式等，最终转化为现实中数据库的过程。一套完整的系统设计，必须要有数据库的支撑，对数据的业务处理，是系统运转的核心。本节主要讨论数据字典的建立到转换为物理模型，再到 E-R 关系图。

驾校是一个服务型企业，涉及到的用户主要有三类：学员(Student)、教练(Coach)、以及驾校管理层人员(Administrator)，服务类型主要是以教练和学员为中心的数据服务，三类用户都有权访问驾校系统中的资源，可将这三类用户统一为驾校的用户(User)，驾校是提供付费培训的场所，支付费用才能获得对应的服务，因此每位用户都要有属于自己在驾校系统中的账户和密码。学员是驾校的用户主体，这三类人群构成了系统的三种用户角色(Role)。不同角色的用户可以在驾校系统中行使自己的权利，每个用户在系统中有着不一样的访问权限(Privilege)，根据以上的逻辑关系，可以构建用户表(driver\_user)的物理模型，对应数据字典如表 1 所示。

表 1 用户表 (driver\_user)

字段名	数据类型	长度	默认值	允许为空	自动递增	主/外键	备注
id	BIGINT	255		NO	YES	P	主键 id
login_id	VARCHAR	36	NULL	YES	NO		登录用户名
telephone	VARCHAR	36		NO	NO		电话号码 (可用于登录)
id_card	VARCHAR	20		NO	NO		身份证号 (可用于登录)
email	VARCHAR	255		NO	NO		邮箱 (可用于登录)
username	VARCHAR	30	NULL	YES	NO		用户昵称
password	VARCHAR	255	id_card 字段后 8 位	YES	NO		登录密码 (初始密码身份证后 8 位)

role	VARCHAR	200	DRIVER_ID_3	NO	NO		身份角色 (默认为学员)
created_time	DATETIME		当前时间	NO	NO		数据创建时间
enable	TINYINT	4	1	NO	NO		数据是否有效

系统共有三类用户，教练和学员属于典型的多对多关系，为了在系统中突出这一层关系，需要设计中间连接表来关联这一层关系，学员和教练作为驾校的主要人群，需要管理更多的用户信息，为了更好的实现多对多的关系，需要单独根据用户信息的需求设计学员、教练、管理员的信息表，由于管理员信息不是系统的主体业务，结构相对简单，不再列出，学员信息表（driver\_student\_archives）和教练信息表（driver\_coach\_archives）的数据字典分别如表 2 表 2 和表 3 表 3 所示。

表 2 学员信息表（driver\_student\_archives）

字段名	数据类型	长度	默认值	允许为空	自动递增	主/外键	备注
id	BIGINT	255		NO	YES	P	主键 id
user_id	BIGINT	255		NO		F	用户表主键
RFID_card_id	BIGINT	255		NO		F	RFID 卡片表主键
real_name	VARCHAR	50	NULL	YES	NO		真实姓名
birthday	DATETIME			YES	NO		出生日期
age	INT	4	NULL	YES	NO		年龄
gender	VARCHAR	10	NULL	YES	NO		性别
nation	VARCHAR	30	NULL	YES	NO		民族
address	VARCHAR	255	NULL	YES	NO		居住地址
paper_file_num	VARCHAR	255	NULL	YES	NO		纸质档案编号
register_id	BIGINT	255		NO	NO	F	报名表主键
created_time	DATETIME		当前时间	NO	NO		数据创建时间
enable	TINYINT	4	1	NO	NO		数据是否有效

表 3 教练表（driver\_coach\_archives）

字段名	数据类型	长度	默认值	允许为空	自动递增	主/外键	备注
id	BIGINT	255		NO	YES	P	主键 id
user_id	BIGINT	255		NO	NO	F	用户表主键

car_info_id	BIGINT	255		NO	NO	F	车辆信息表主键
RFID_card_id	BIGINT	255		NO	NO	F	RFID 卡片表主键
real_name	VARCHAR	50	NULL	YES	NO		真实姓名
birthday	DATETIME		当前时间	YES	NO		出生日期
age	INT	4	NULL	YES	NO		年龄
gender	VARCHAR	10	NULL	YES	NO		性别
nation	VARCHAR	30	NULL	YES	NO		民族
address	VARCHAR	255	NULL	YES	NO		居住地址
wage	NUMBER	10, 2	0	YES	NO		工资
driver_type	VARCHAR	100	NULL	YES	NO		驾照类型
created_time	DATETIME		当前时间	NO	NO		数据创建时间
enable	TINYINT	4	1	NO	NO		数据是否有效

通过中间桥表（student\_associate\_coach）的方式，将学员与教练建立多对多的关联关系，关联关系的数据字典，如表 4 表 4 所示。

表 4 学员教练关联表（student\_associate\_coach）

字段名	数据类型	长度	默认值	允许为空	自动递增	主/外键	备注
id	BIGINT	255		NO	YES	P	主键 id
student_id	BIGINT	255		NO		F	学员表主键
coach_id	BIGINT	255		NO	NO	F	教练表主键
associated_time	DATETIME		当前时间	NO	NO		关联时间
comment	VARCHAR	255	NULL	YES	NO		备注
created_time	DATETIME		当前时间	NO	NO		数据创建时间
enable	TINYINT	4	1	NO	NO		数据是否有效

RFID 模块是本系统中数据传递的重要组成部分，考虑到系统中用户数据的安全性，RFID 卡片扇区中不存储用户的任何数据，采用只读卡，不写卡的方式，将 RFID 的唯一识别卡号在数据库中记录，并与对应的用户信息进行绑定，每次读卡操作都将使用网络通信的方式，向服务器中请求该卡片对应的用户数据，并将用户的操作信息发送至服务器端，保存在数据库中。由上所述，需要建立 RFID 卡片的数据字典，如表 5 表 5 所示，并通过外键与用户数据相绑定。

表 5 RFID 卡片表（driver\_rfid\_info）

字段名	数据类型	长度	默认值	允许为空	自动递增	主/外键	备注
-----	------	----	-----	------	------	------	----

id	BIGINT	255		NO	YES	P	主键 id
RFID_num	VARCHAR	255	NULL	YES	NO		RFID 卡片唯一编号
color	VARCHAR	10	NULL	YES	NO		卡片颜色
publish_time	DATETIME		当前时间	YES	NO		卡片激活时间
comment	VARCHAR	255	NULL	YES	NO		备注
created_time	DATETIME		当前时间	NO	NO		数据创建时间
enable	TINYINT	4	1	NO	NO		数据是否有效

除了上述的几张表之外，系统正常运行还需要管理员表、权限表、RFID 读写卡记录表、科目表、成绩表、练车预约记录表、教练车信息记录表、用户操作记录表等，其余表字段简单，关联性不强，这里不再详细列出，以上数据字典都是本系统核心业务运行的基础数据表。

通过软件自动化构建物理模型非常方便，使用软件 Power Designer<sup>①</sup>构建物理模型，只需要根据数据字典将表和各个字段一一对应，再画出来表跟表之间的对应关系，就能创建出物理模型，然后一键生成 MySQL 可执行脚本文件，再使用数据库运行脚本文件，就可以创建出数据库以及生成对应的 E-R 关系图。

### 2.2.2 系统主要硬件设计

本系统采用的主要硬件设备清单如表 6 表-6 所示。

表 6 主要硬件清单

型号	数量	用途	主要功能
STM32F103ZET6 系列开发板	1 个	处理数据，控制开发板上的各个硬件工作	实现开发板各个引脚功能复用，驱动串口工作
MFRC-522 RFID 射频读写卡模块	1 个	读写 S50 系列 IC 卡、异形卡数据	读写射频卡数据
IC 射频卡	3 张	提供数据源	监测用户行为，记录用户信息
ESP8266 WIFI 模块	1 个	网络通信	通过 HTTP 协议，向远程服务器发送数据

本系统采用的开发板的 MCU 是基于 ARM 的 32 位通用增强型微控制器，引脚数目 144，闪存容量 512K 字节，具体型号为 STM32F103ZET6。这款开发板属于定制款，除了使用官方的仿真器下载程序，还可以通过 USB 直接下载程序，方便快捷，不需要单独购买仿真器。需要将开发板的 BOOT0 和 BOOT1 的电平位通过跳帽调整，切换

① Power Designer 是 Sybase 公司的 CASE 工具集。利用 Power Designer 可以制作数据流程图、概念数据模型、物理数据模型，还可以为数据仓库制作结构模型，也能对团队设计模型进行控制。

启动模式到系统存储器启动模式下，通过串口直接下载程序。使用好官方的固件库，更能减轻代码成本。本论文从服务商下载到了官方的 V3.5 版本的固件库，本系统的所有程序都是在此固件库基础上开发。[图 2 图-2](#)为基于 STM32 微控制器提供的物联网平台原理图。

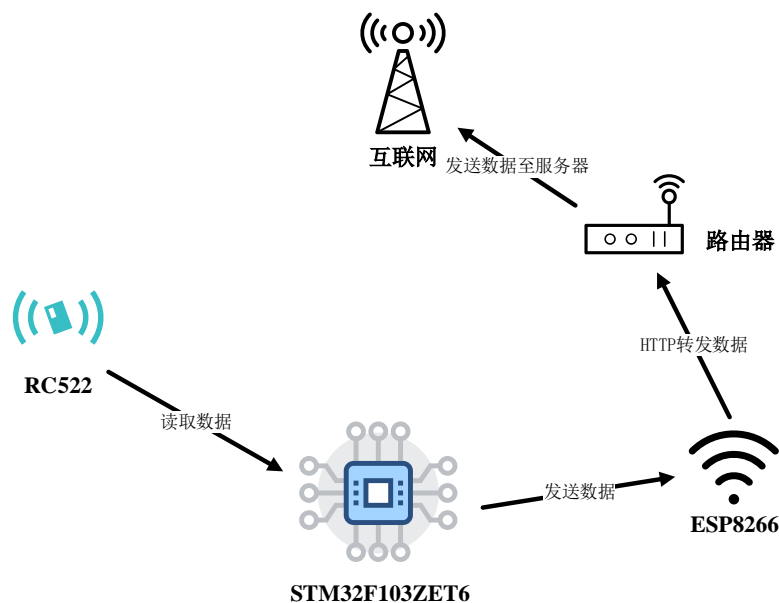


图 2 物联网平台原理图

### 2.2.3 系统 Web 端设计

Web 端主要分为两部分，Web 服务端和 Web 前端。Web 服务端是一台使用 Java 语言开发的服务器，用来接收 ARM 微控制器通过网络发送的数据，将接收到的数据分析之后存入数据库中，用户在 Web 前端查看对应的服务时，Java 服务器再将数据从数据库中读取出来，经过业务流程的筛选处理，渲染到前端页面上。Java 服务器基于轻量级开源框架 Spring 开发，能够完美支持多线程、高并发，提供了一站式解决方案，包括依赖注入、动态加载、切面编程、事务管理等。Web 模块主要的功能原理如[图 3 图-3](#)所示。



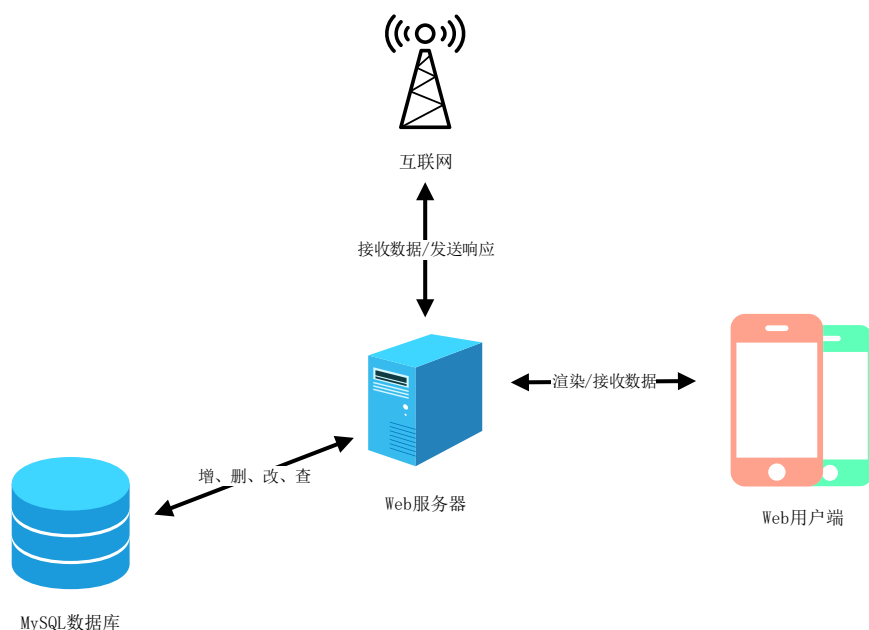


图 3 Web 模块原理图

### 3 系统整体实现方案

#### 3.1 使用版本控制器构建项目仓库

本系统从项目创建到最终开发完成，都采用了分布式版本控制器 git 来管理代码，以及控制项目版本的迭代，git 不仅仅具备版本控制的功能，还支持良好的团队协作，以及远程仓库关联同步等功能。git 是一款开源免费的软件，不管有没有网络环境，都能正常工作，创建好本地仓库之后，没有远程仓库也能够在本地图管理代码。面前互联网上能够与 git 配合使用的免费仓库有 Github 和 Gitee（中文名码云），同时也可以使用 gitLab 自己搭建 git 远程仓库，本系统使用的远程仓库是 github。完成项目版本控制，需要搭建以下环境，如表 7 所示。

表 7 版本控制器环境

名称	获取地址	性质	备注
Github 账户	<a href="https://github.com/">https://github.com/</a>	免费	注册账户 创建远程仓库
Git 客户端	<a href="https://git-scm.com/">https://git-scm.com/</a>	免费	git 命令环境

安装好 gitBash 环境，同样需要配置 git 的环境变量，配置好环境变量，打开命令行终端，在终端里输入 `git -version`，看到 git 的版本信息，如图 4 所示，就具

备了 git 版本控制器的环境。

```
d:\ $ git --version
git version 2.21.0.windows.1
```

图 4 查看 git 版本

### 3.1.1 初始化本地项目仓库

为了方便管理系统的全部代码，在开始真正编码之前，先创建项目的目录结构，将系统所有的文件整理归类，让 git 来统一管理，git 虽然不管理空文件夹，但当文件夹下有了内容，git 自然会去管理。

首先新建一个文件夹，根据表 8 表 8 所示清单，创建好如图 5 图 5 所示的目录。

表 8 项目目录结构说明

文件/文件夹名称	用途
back_end	Java 后端服务器代码存放目录
front_end	Web 前端页面代码存放目录
database	数据库相关内容存放（模型、sql 文件等）
stm32_client	STM32 ARM 程序存放目录
doc	设计说明文档存放目录
README.md	git 仓库项目简要描述说明文档

名称	修改日期	类型	大小
back_end	2020/3/21 23:11	文件夹	
database	2020/3/21 23:12	文件夹	
doc	2020/3/21 23:12	文件夹	
front_end	2020/3/21 23:16	文件夹	
stm32_client	2020/3/21 23:12	文件夹	
README.md	2020/3/21 23:13	Markdown File	1 KB

图 5 项目结构目录

打开命令行终端，使用命令 `git init` 将目录初始化为一个 git 仓库，执行完成后，该目录下会变成一个 git 管理的仓库，同时会产生 .git 文件夹，如图 6 图 6 所示

```
d:\Documents\new\driver $ git init
Initialized empty Git repository in d:/Documents/new/driver/.git/

d:\Documents\new\driver (master -> origin) $ ls -a
./ ../ .git/ back_end/ database/ doc/ front_end/ README.md stm32_client/
```

图 6 初始化项目为 git 仓库

然后使用命令将初始化仓库提交到本地版本库，提交时，git 会询问操作的用户，使用命令配置用户名和邮箱，再次提交到本地版本库，如图 7 图-7 所示。

```
d:\Documents\new\driver (master -> origin) $ git add .
d:\Documents\new\driver (master -> origin) $ git commit -m "初始化项目仓库"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'witt@DESKTOP-M9S9U4T.(none)')

d:\Documents\new\driver (master -> origin) $ git config user.name "lele"
d:\Documents\new\driver (master -> origin) $ git config user.email "fgwang.66@gmail.com"
d:\Documents\new\driver (master -> origin) $ git commit -m "初始化项目仓库"
[master (root-commit) 6fd9786] 初始化项目仓库
 1 file changed, 4 insertions(+)
 create mode 100644 README.md
```

图 7 提交项目目录结构

本系统的项目包括有 C 语言编写的 ARM 程序，Java 开发的 Web 服务器，前端框架开发的 Web 页面，版本库只需要管理这三个平台的源代码即可，例如 C 语言编译的过程文件，编译生成的可执行文件，下载到单片机开发板的 hex 文件，Java 编译后的 class 文件等，都可以从源代码再次编译产生，版本库需要忽略掉这些文件，需要在项目根目录下创建.gitignore 文件，在里面编写需要忽略追踪的文件的表达式，如图 8 图-8 所示。

批注 [19]: 论文中的代码，小五号，新罗马，单倍行距，加底纹（灰度第二格），边框 0.5 磅。你这个格式打印出来可能看不清吧。

带格式的：缩进：首行缩进： 0 字符

```
1  ### Keil项目相关的忽略文件 ###
2  *.plg
3  *.uvgui.*
4  JLinkLog.txt
5  *.o
6  *.d
7  *.crf
8  *.tra
9  *.axf
10 # *.hex
11 *.lnp
12 *.sct
13 *._i
14 *.dep
15 *.map
16 *.lst
17 *.iex
18 *.scvd
19 *.uvguix
20 *.dbg*
21 *.uvguix.*
22 *.mxproject
23 stm32_client/**/*.htm
24 stm32_client/driver/**/*.bak
25 stm32_client/**/*.log
26 stm32_client/**/*.zip
27 !*.c
28 !*.h
29 !*.s
30
31 ### Java Server项目相关的忽略文件 ###
32
33 # 日志文件务必忽略
34 **/back_end/driver-manager-system/logs/**
35 *.log
36 **/back_end/temp/**
37
```

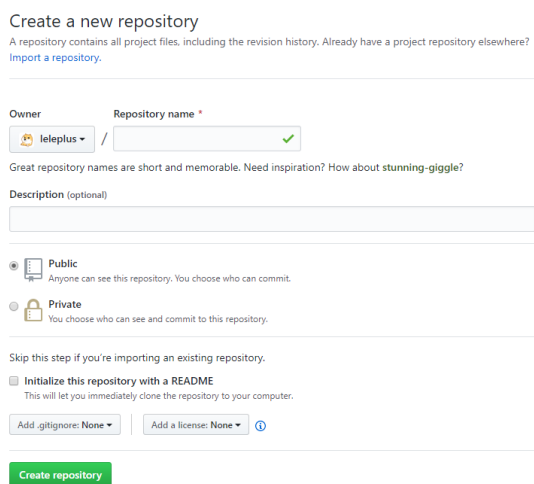
图 8 gitignore 文件配置

然后将.gitignore 文件同时提交到版本库中。

### 3.1.2 关联同步远程仓库

github 提供了免费的代码仓库管理服务，注册账户之后，新建代码仓库如[图 9](#)所示。

批注 [110]: 论文中的代码，小五号，新罗马，单倍行距，加底纹（灰度第二格），边框 0.5 磅。  
你这个格式红色和绿色打印出来都是灰色，可能看不清吧。



批注 [111]: 对比度得调整一下，图像看不清。

图 9 创建远程仓库

创建好之后，会得到一个远程仓库地址，如图 10 图-10 所示，有 SSH 和 HTTPS 两种协议的地址，我这里使用 SSH 协议地址

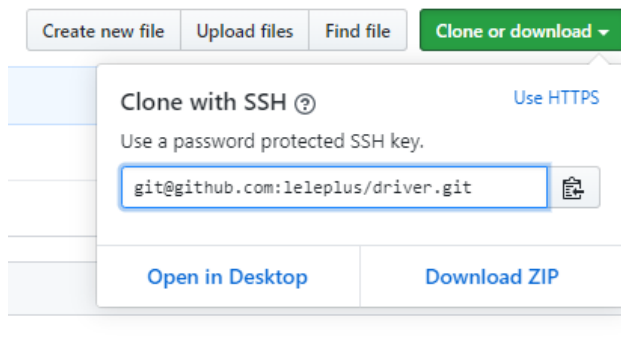


图 10 远程仓库地址

使用 SSH 协议之前，需要先将本地电脑的 SSH 公钥配置到 github 的 SSH Key 中，然后在创建好的本地仓库里，通过终端执行命令 `git remote add origin git@github.com:leleplus/driver.git`，如图 11 图-11 所示。

```
D:\Documents\new\driver (master -> origin) $ git remote add origin git@github.com:leleplus/driver.git
```

图 11 关联远程仓库

使用命令 `git push -u origin master` 将本地的代码仓库覆盖远程的仓库，后续将代

码推送到远程只需要使用命令 `git push`。

### 3.1.3 更新同步项目代码

本系统的版本控制，主要管理的内容有三部分，STM32 相关的 Keil 代码，Web 服务器和 Web 页面的代码，还有系统文档，对应这三部分内容，分别使用 git 创建三个分支，相关命令为 `git checkout -b 分支名`，在对应分支下编写对应模块的代码，编写完成后，将对应的分支使用 `git push` 推送到远程仓库管理，本地分支如图 12 图 12 所示。

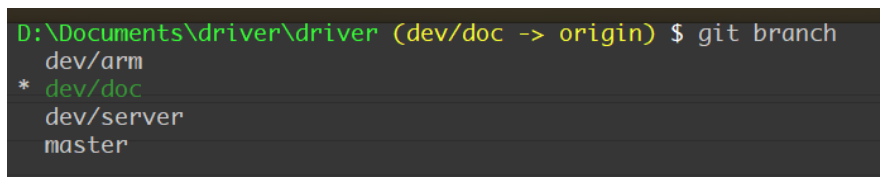


图 12 项目分支

管理好项目仓库之后，每次编写代码，只需要使用命令 `git push` 将代码推送到远程分支，更新代码时，使用命令 `git pull`，切换对应模块所在的分支，使用命令 `git checkout`，最终完成项目之后，使用 `git merge` 命令将各个模块所在分支的代码合并到 master 分支即可。

## 3.2 数据持久层实现

### 3.2.1 系统数据建模实现

按照系统的需求分析流程，编写出了系统的数据字典文档，按照数据字典文档使用物理模型构建工具 Power Designer 创建物理模型，实现流程如下：

(1) 安装好 Power Designer 工具，运行软件，如图 13 图 13 所示，新建一个模型。

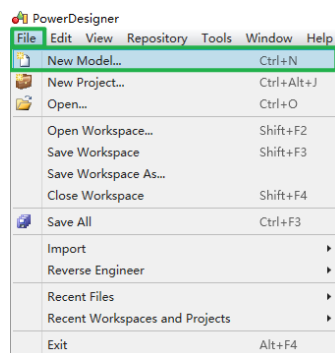


图 13 新建模型

(2) 选择模型的类型为物理模型，选择方式如图 14 图 14 所示。本系统

使用的 MySQL 版本为 5.7.29，这里必须要选择 MySQL 版本为 5.0。

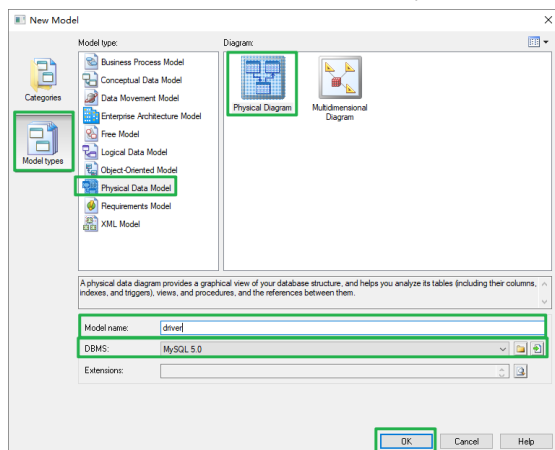


图 14 选择模型类型

（3）工具提供了在图形化界面通过拖拉拽的方式快速构建模型，如图 15 图 15 所示，在 Toolbox 菜单下，有表（Table）和关联关系（Reference）的按钮，使用鼠标，点击这两个按钮，能够方便的构建出物理模型以及实体之间的关系。

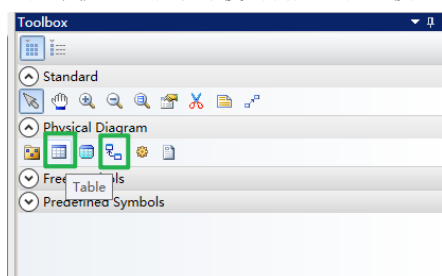


图 15 构建实体

（4）双击每一个 Table，在弹出的窗口中，编辑添加表对应的字段、类型、长度、等属性，如图 16 图 16 所示。

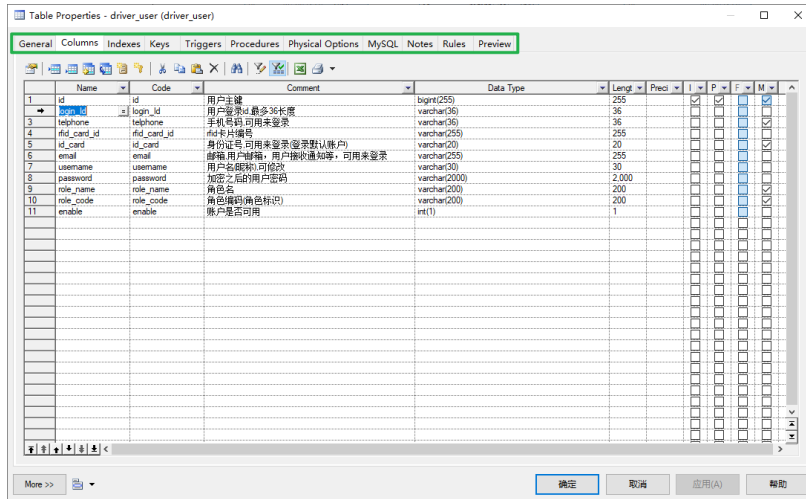


图 16 编辑 Table

(5) 创建完模型，按照图 17 图 18 所示方式，可直接预览物理模型对应的数据库 SQL 执行脚本，如图 18 图 18 所示。

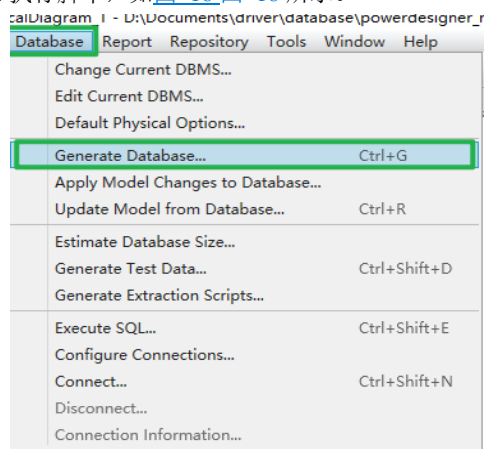


图 17 生成脚本



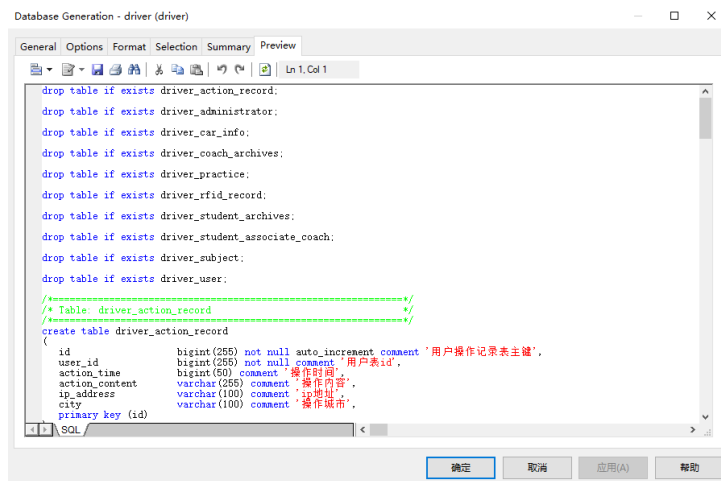


图 18 预览 SQL 脚本

(6) 设置编码，本系统使用的字符编码统一为 UTF-8，导出 SQL 脚本前设置脚本编码，如图 19 图 20 所示。

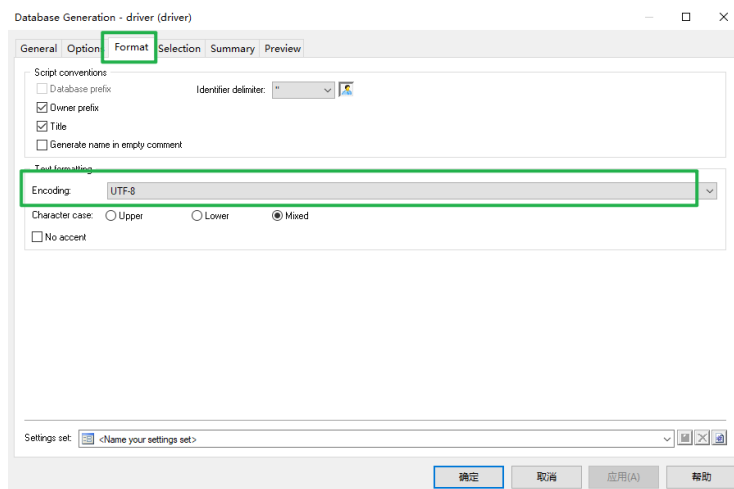


图 19 设置 SQL 脚本编码

(7) 选择导出 SQL 脚本的位置以及脚本文件的名称，如图 20 图 20 所示。点击确定之后，就会在对应的目录下生成好文件。

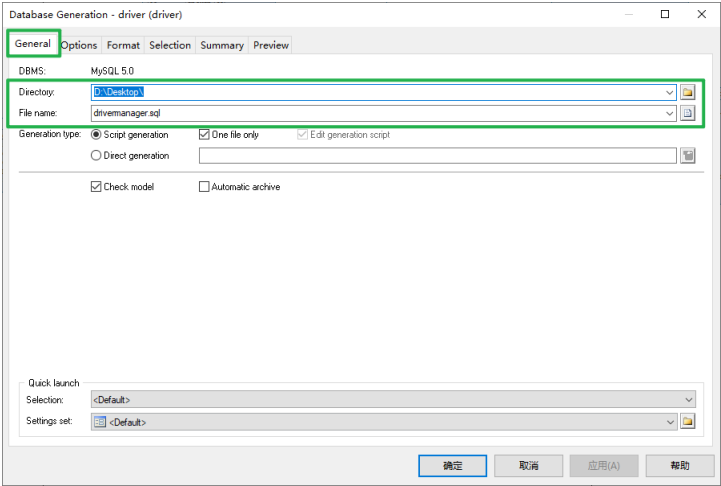


图 20 选择脚本位置设置文件名

3.2.2 数据库设计实现以及 E-R 图的生成

构建完物理模型，就已经有了数据库的脚本文件，直接在数据库中运行脚本文件，就会在数据库中创建好跟物理模型一样的数据库结构。完成数据库设计以及 E-R 生成需要如表 9 表 9 所示的软件清单。

表 9 数据库设计环境

软件名称	版本	用途	说明
JDK	1.8	Java 开发工具集以及 DBeaver 运行时环境	免费下载 需要配置环境变量
MySQL	5.7.29	数据库软件	免费开源 需要配置环境变量
DBeaver	7.0	数据库可视化管理工具	免费开源 基于 Java 开发，需要 JDK

设置了格式：字体：五号

带格式表格

设置了格式：字体：五号

设置了格式：字体：五号

设置了格式：字体：五号

具体实现步骤如下：

（1）安装完软件，开始创建数据库，如图所示，打开数据库可视化管理工具 DBeaver，使用账户密码连接到数据库，如图 21 图 24 所示。

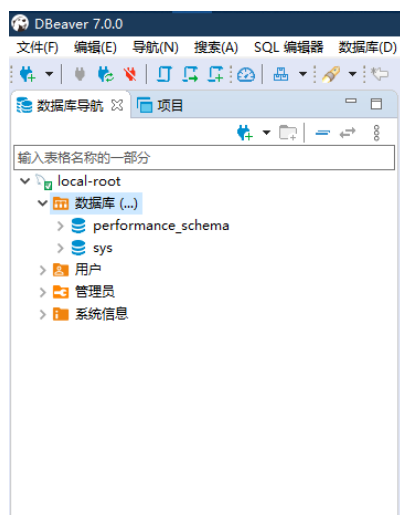


图 21 连接数据库

(2) 新建开发用的数据库，保证编码为 UTF-8，如[图 22](#)~~图 22~~所示。

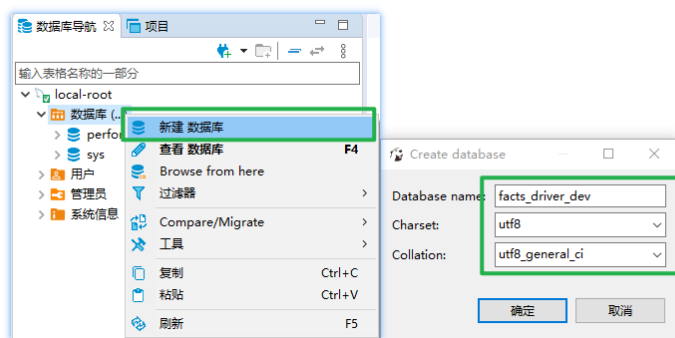


图 22 创建数据库

(3) 运行数据库脚本，按照[图 23](#)~~图 23~~所示的步骤，运行数据库脚本文件，如[图 24](#)~~图 24~~所示，需要指定 MySQL 的安装目录，以及物理模型生成的 SQL 脚本文件的位置，点击确定即可创建好数据库结构。

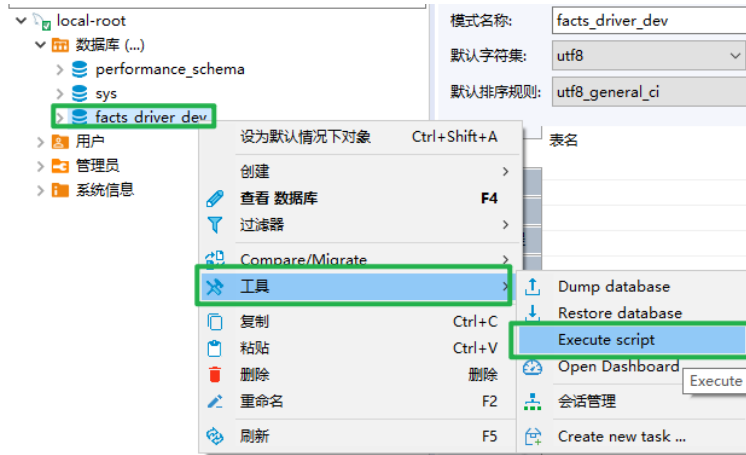


图 23 运行脚本

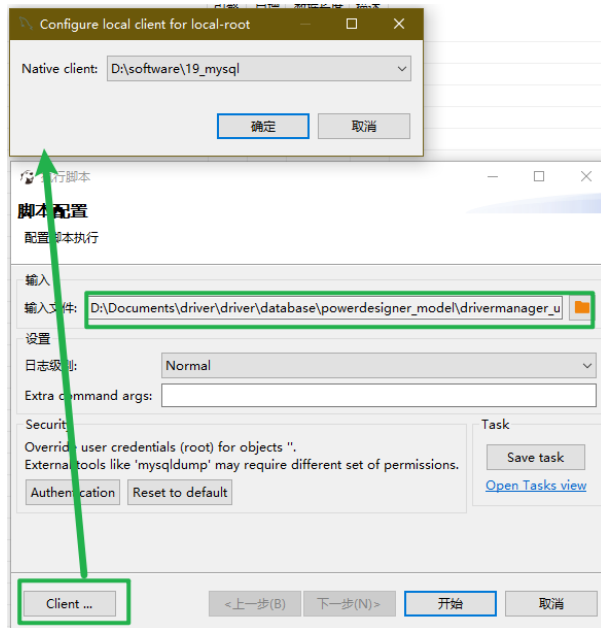


图 24 选择 mysql 客户端和脚本文件

(4) 数据库已经创建完成，创建好的表如图 25 图 25 所示。

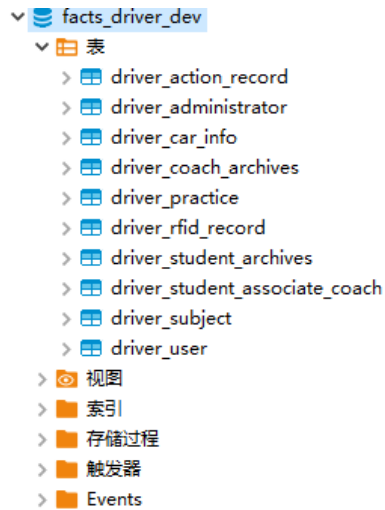


图 25 系统所需数据库的表

（5）生成 E-R 关系图，双击打开数据库，切换到 E-R 图，即可看到 E-R 图，步骤如 [图 26](#) ~~图 26~~ 所示。可以导出为图片等。

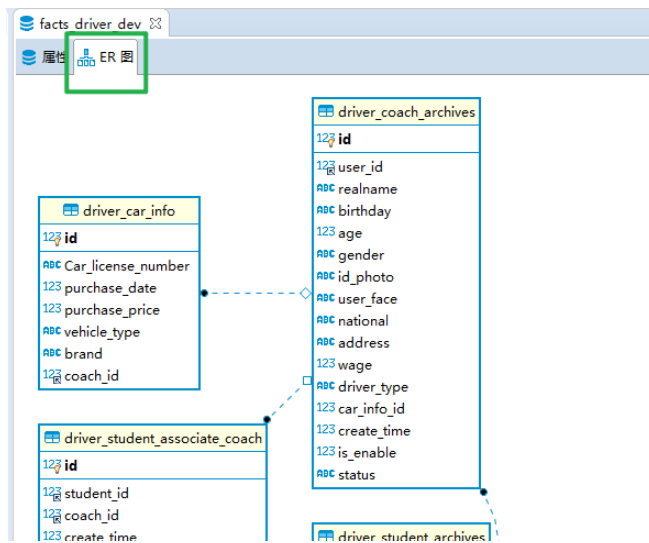


图 26 生成 E-R 图

系统最终使用的 E-R 关系图，见附录 1 所示。

### 3.3 物联网模块实现

#### 3.3.1 物联网模块硬件产品介绍

物联网模块主要是在 ARM 上做 C 语言开发，使用 Keil5<sup>①</sup>作为开发环境，同时需要安装 STM32F103ZE 系列的软件包来编译运行 C 语言代码，完成本模块需要的开发环境如表 10 所示。

表 10 STM32 开发环境清单

名称	用途	版本	来源
Keil-MDK	编程环境	V5.11	网络下载
STM32F1xx_DFP	编译执行 STM32F103 固件库代码	V2.1.0	网络下载
STM32F103ZE 固件原理图	查看引脚定义等	V1.1	随 STM32 购买附赠
STM32F103ZE 尺寸图	查看设备型号、接口等	V1.08	随 STM32 购买附赠
Micro-USB 下载线	下载程序		购买
杜邦线	连接模块和开发板		购买
固件库函数	编写 Keil 代码的底层参考程序	V3.5	随 STM32 购买附赠
CH340 驱动程序	下载程序驱动	V1.6.0	随 STM32 购买附赠
ISP 下载器 MCUISP	下载 HEX 文件到开发板	V0.99	随 STM32 购买附赠

3.3.1.1 STM32F103ZET6 核心板

本设计所使用的开发板，板载 MCU 为 STM32F103ZET6，主频为 72MHZ，闪存 512K,64K 的 SRAM，具备 4 个定时器，112 个 GPIO 引脚，5 个 UART，1 个 USB 接口，整体外观如图 27 所示。板载三个 LED 灯，四个按键（一个复位按键，三个独立按键），支持 ISP 一键下载程序，同时具备了 SWD 仿真功能，本论文主要使用 ISP 下载程序，后面章节详细描述。

① Keil 是德国知名软件公司 Keil（现已并入 ARM 公司）开发的微控制器软件开发平台，是目前 ARM 内核单片机开发的主流工具。Keil 提供了包括 C 编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器在内的完整开发方案，通过一个集成开发环境（uVision）将这些功能组合在一起。uVision 当前最高版本是 uVision3,它的界面和常用的微软 VC++的界面相似，界面友好，易学易用，在调试程序，软件仿真方面也有很强大的功能。因此很多开发 ARM 应用的工程师，都对它十分喜欢。

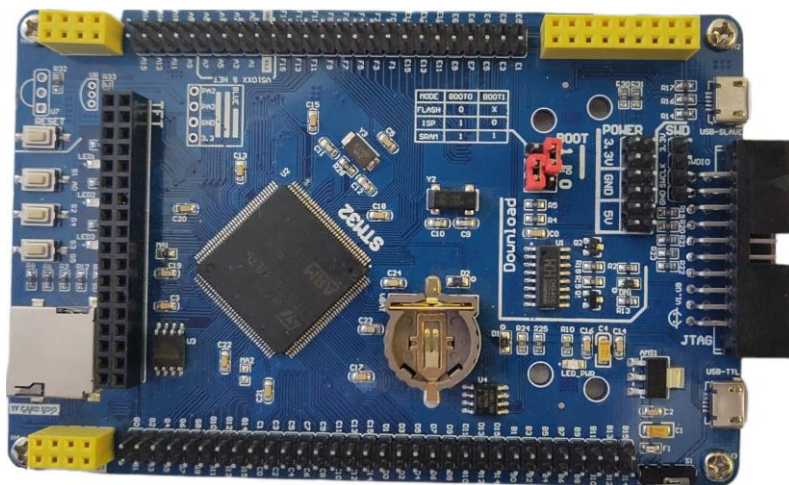


图 27 STM32F103ZET6 开发板外观图

### 3.3.1.2 ESP8266 网络通信模块

ESP8266 网络通信模块，尺寸为 5x5mm，模组需要的外围器件有：10 个电阻电容电感、1 个无源晶振、1 个 flash。工作温度范围：40~125℃。有 StandAlone 和 SIP 两种模式，CPU 主频支持 80M 到 160M，GPIO 输出电压 3.3V，具备两个 UART，UART0 和 UART1。整体外观如图 28 图 28 所示。



图 28 ESP8266 外观图

### 3.3.2 通过固件库创建工程

从零开始编写一套 ARM 程序，很显然任务量太重，本系统是基于官方固件库创建的工程，创建主要步骤如下

- (1) 下载好 STM32F10 系列的固件库源代码，如图 29 图 29 所示。



图 29 STM32V3.5 固件库

(2)新建文件夹 bin 和 src, bin 用来存放编译之后的文件, src 用来存放源代码, 如图 30 图 30 所示。

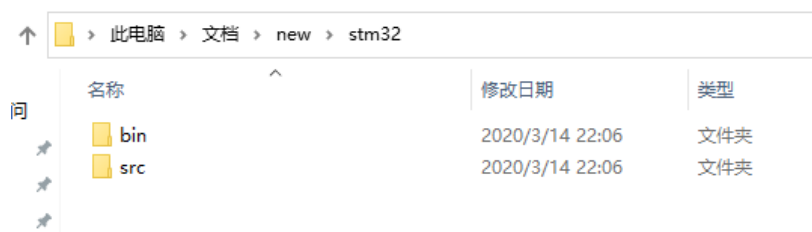


图 30 源码和编译文件目录

(3) 创建项目工程文件夹, 具体作用如表 11 表 11 所示, 创建完成后的效果如图 31 工程文件目录图 31 工程文件目录所示。

表 11 工程文件目录清单

文件夹名称	用途
CORE	存放 STM32 核心文件
FWLIB	各个外设的底层程序和源码
LIBRARY	自己定义引用的 STM32 库文件
STARTUP	STM32 启动文件
USER	自定义的程序以及主程序入口文件

带格式的: 行距: 多倍行距 1.25 字行

带格式表格

带格式的: 行距: 多倍行距 1.25 字行

带格式的: 行距: 多倍行距 1.25 字行

带格式的: 行距: 多倍行距 1.25 字行

带格式的: 行距: 多倍行距 1.25 字行

带格式的: 行距: 多倍行距 1.25 字行

带格式的: 行距: 多倍行距 1.25 字行



名称	修改日期	类型	大小
CORE	2020/3/14 22:06	文件夹	
FWLIB	2020/3/15 0:16	文件夹	
LIBRARY	2020/3/14 22:22	文件夹	
STARTUP	2020/3/14 22:22	文件夹	
USER	2020/3/14 22:22	文件夹	

图 31 工程文件目录

（4）再在每个文件夹下创建文件夹 inc 和 src，inc 用来存放.h 头文件,src 用来存放.c 文件，然后使用 Keil 创建工程，位置选择上一步创建的工程目录，如[图 32](#) ~~图 32~~所示。

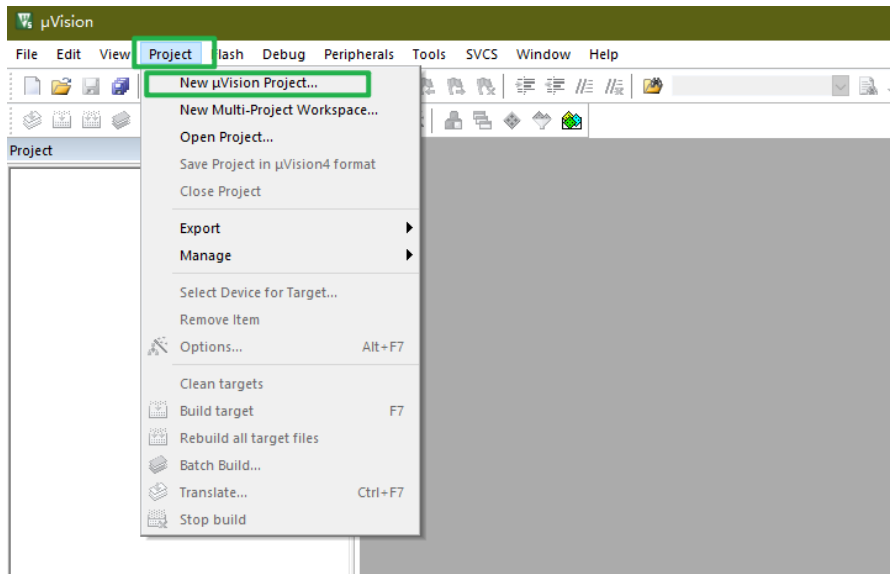


图 32 Keil 创建项目

（5）本系统使用的系统板为 STM32F103ZE 系列，新建项目需要使用对应的 Device 设备，如[图 33](#) ~~图 33~~所示。

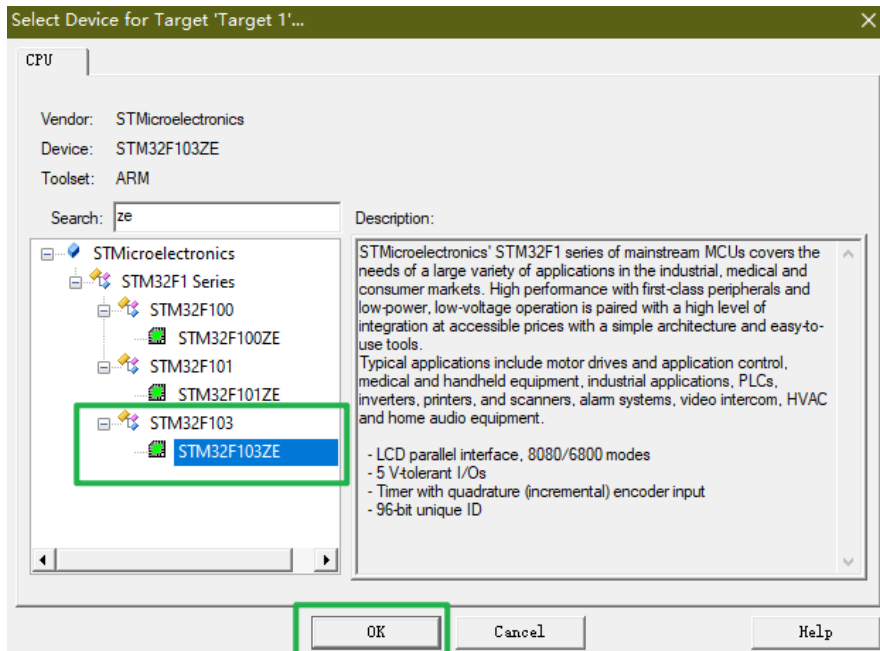


图 33 添加 STM32F103ZE Device 到项目中

(6) 将项目已有的程序文件关联到项目中，如图 34 图 34 所示。

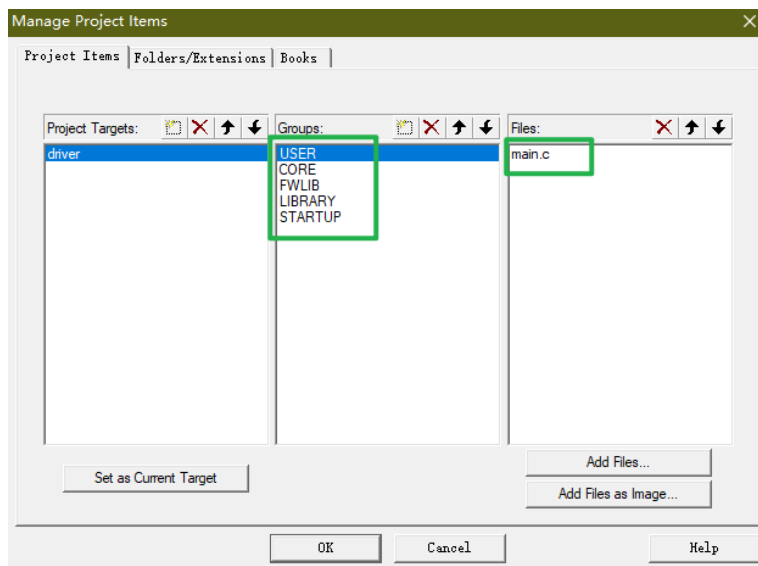


图 34 关联项目和工程

（7）关联完成后，整体项目目录结构如[图 35 图-35](#)所示

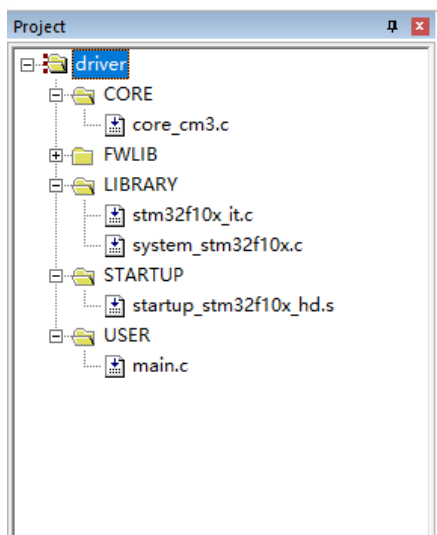


图 35 Keil 项目目录结构

（8）库函数在配置和选择外设的时候是通过宏定义来选择的，务必要配置一下系统全局的宏定义和头文件所在的目录，Keil5 的开发环境，已经预置了一部分宏定义，只需要配置 `USE_STDPERIPH_DRIVER` 就能完成编译，同时还要指定各个目录的 inc 文件夹位置，所有的头文件都存放在 inc 文件夹下，如[图 36 图-36](#)所示。

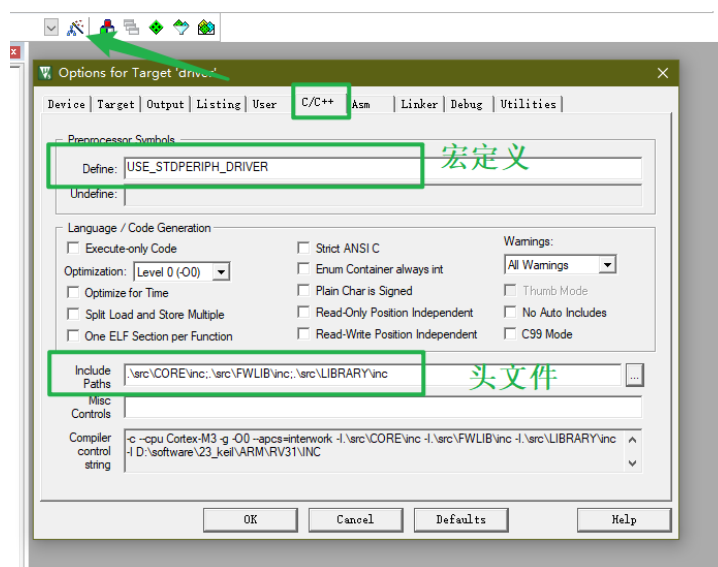


图 36 设置宏定义，关联头文件

### 3.3.3 ISP 串口下载程序

单片机下载程序的方法有好多种，本系统采用的是 ISP 方式下载，这种下载方式需要使用出厂预置在单片机内部的 BootLoader，所以需要更改单片机的启动模式。STM32 启动模式在开发板图例和文档中都有介绍，更改启动模式，只需要更换 BOOT 跳帽的接入方式，切换高低电平即可。如表 12 所示为开发板配置启动模式的跳帽接入方式。

表 12 STM32 启动模式

启动模式	BOOT0	BOOT1
FLASH	0	
ISP	1	0
SRAM	1	1

### 3.4 Web 模块实现

Web 服务端主要用到了 Java 语言，需要 JDK 的安装环境，首先需要安装 JDK，本设计使用的 JDK 版本为 JDK1.8，JDK 的安装非常简单，安装完成后需要配置环境变量，配置方式如图 11 所示，打开 Windows 运行框，输入 sysdm.cpl，确定后打开系统属性。

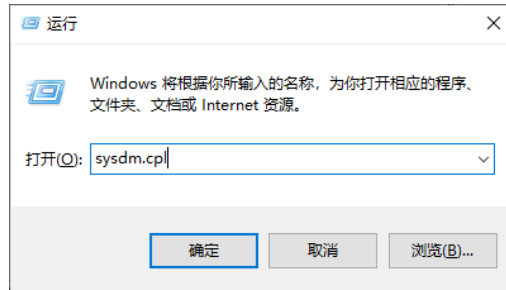


图 37 运行 sysdm.cpl

如图 12 所示，选择高级，环境变量。



图 38 设置环境变量

如图 13 所示，在系统环境变量下，依次新建环境变量，变量名 JAVA\_HOME,变量值 JDK 安装路径下能看到 bin 的路径，变量名 CLASSPATH，变量值.。

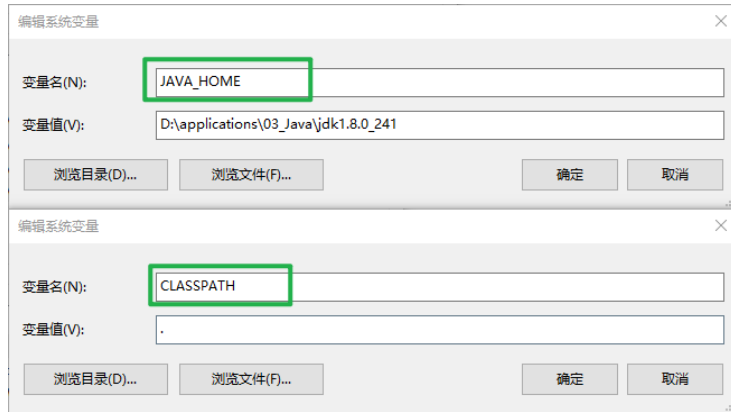


图 39 编辑环境变量

在 Path 变量下新建变量%JAVA\_HOME%\bin，引用下 JAVA\_HOME 的变量。如图 14 所示，运行 cmd，输入 java -version，能看到 Java 版本，JDK 安装成功。

```
d:\Documents\driver (dev/doc -> origin)
$ java -version
java version "1.8.0_241"
Java(TM) SE Runtime Environment (build 1.8.0_241-b07)
Java HotSpot(TM) 64-Bit Server VM (build 25.241-b07, mixed mode)
```

图 40 JDK 安装成功验证

本系统采用的 Java 开发 IDE 为 IDEA<sup>①</sup>，另外还使用了 maven、Tomcat、git 等工具部署管理项目

<sup>①</sup> IDEA 全称 IntelliJ IDEA，是 java 编程语言开发的集成环境。IntelliJ 在业界被公认为最好的 java 开发工具，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、各类版本工具(git、svn 等)、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能可以说是超常的。

## 主要参考文献

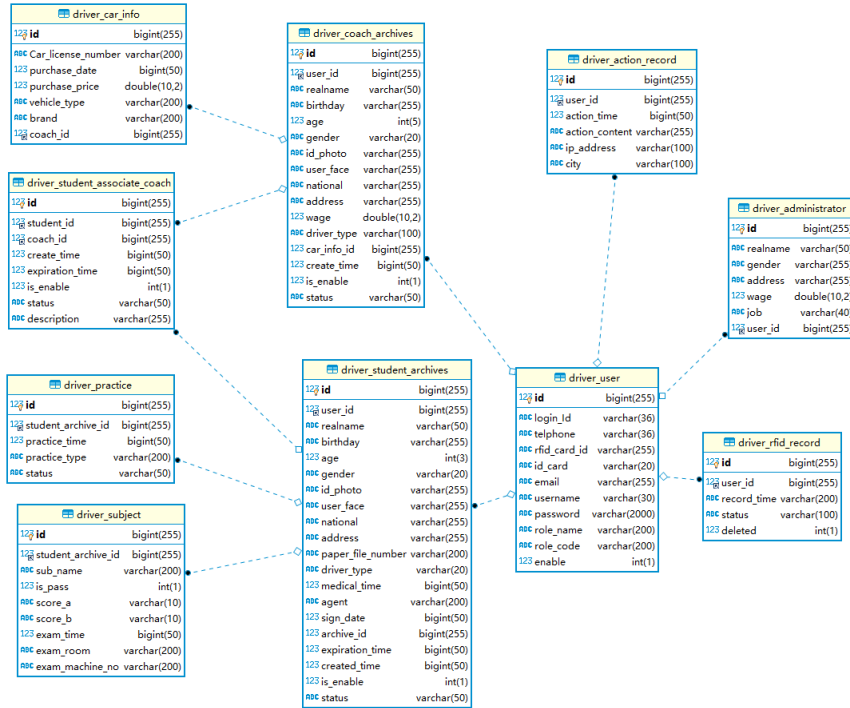
- [1] Luger, George. 人工智能：复杂问题求解的结构和策略. 由史忠植 等翻译 原书第 4 版. 北京：机械工业出版社. 2004. ISBN 7-111-12944-X （中文）.
- [2] 徐庆. 驾校网络预约服务系统设计与实现[D]. 吉林大学. 2015.
- [3] 孙根. (0). 基于 SSM 框架的驾校管理平台的研究与实现. (Doctoral dissertation). 武汉邮电科学研究院. 2017.
- [4] 庞双玉. Git 分布式版本控制实现机制探讨[J]. 信息系统工程, 298(10):55-56.
- [5] 张萍. 基于 ESP8266 和 OneNET 云平台的远程报警系统[J]. 单片机与嵌入式系统应用, 2017(12):70-73.
- [6] 牛小宝. 基于 MySQL 的云数据库设计与实现[D]. 2016.
- [7] 王振宇. 基于 Spring Boot 的整车出库管理系统设计与开发[D].
- [8] 贺倩. 人工智能技术的发展与应用[J]. 电力信息与通信技术, 2017, 000(009):P. 32-37.
- [9] 袁乐民. 基于 STM32 的 RFID 读卡器设计与实现[D]. 2015.
- [10] 王伟, 刘法胜, Dima. 浅谈驾照通用与国际出行[J]. 山东交通科技, 2014(05):17-19.
- [11] 王祎祎, 王梦倩, 雷馨媛, et al. 驾校的信息化发展水平分析[J]. 办公自动化, 2019, 24(09):31-33.
- [12] 王成. 基于 STM32 的超高频 RFID 读写器软件系统的设计实现[D].
- [13] 钱颖雪. 版本控制技术在物联网实践课程中的应用 [J]. 电脑与电信, 2016(10):38-40, 共 3 页.
- [14] 徐庆. 驾校网络预约服务系统设计与实现[D].
- [15] 史永哲. 基于 HTTP 协议的远程网站管理平台的实现[J]. 工业仪表与自动化装置, 2016(3):44-46.

批注 [112]: 部分参考文献不完整。请按照格式要求补充完整。

带格式的: 段落间距段后: 1 行

带格式的: 缩进: 左侧: 0 厘米, 悬挂缩进: 2 字符, 首行缩进: -2 字符

## 附录 1 数据库 E-R 关系图





附录 2 主要英文缩写语对照表

缩略语	中文全称	英文全称
ARM	微控制处理器	Advanced RISC Machine
GPIO	通用输入输出端口	General Purpose Input Output
USB	通用串行总线	Universal Serial BUS

设置了格式：字体：Times New Roman，五号

设置了格式：字体：Times New Roman，五号

设置了格式：字体：Times New Roman，五号

设置了格式：字体：Times New Roman，五号