



毕业论文（设计）

**基于物联网环境的智能驾校管理系统**

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

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 -](#_Toc35548979)

[关键词 - 3 -](#_Toc35548980)

[Abstract - 3 -](#_Toc35548981)

[Key words - 4 -](#_Toc35548982)

[1绪论 - 5 -](#_Toc35548983)

[1.1研究背景及意义 - 5 -](#_Toc35548984)

[1.2国内外的研究现状及发展趋势 - 6 -](#_Toc35548985)

[1.3 研究本课题的目的和基本内容 - 6 -](#_Toc35548986)

[2.系统总体设计方案 - 7 -](#_Toc35548987)

[2.1系统需求分析 - 7 -](#_Toc35548988)

[2.2系统总体架构设计 - 8 -](#_Toc35548989)

[3.系统整体实现方案 - 13 -](#_Toc35548990)

[3.1 数据持久层实现 - 18 -](#_Toc35548991)

[3.2 物联网模块实现 - 24 -](#_Toc35548992)

[3.3 Web模块实现 - 26 -](#_Toc35548993)

[主要参考文献 - 29 -](#_Toc35548994)

[附录1 数据库E-R关系图 - 30 -](#_Toc35548995)

[附录2 主要英文缩写语对照表 - 31 -](#_Toc35548996)

**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**

现代社会是基于是互联网的时代，是基于人工智能的时代，从当前物联网横行的社会潮流来看，探讨万物互联，顺应时代发展的规律，这是一个永恒不变的话题。从国际视角出发，人们生活水平普遍提高，汽车已经成为了家家户户必不可少的交通出行工具，各国关于驾车出行都有各种各样的条例，以我国最为典型，无证不可上路，是我国公民必须遵守的法律条例。

驾照已经成为了我们人生中必不可少的证件之一，是证明我们安全出行的保证，现代社会，驾校遍布，种类繁多，业务杂乱，随着“考证”人数的越来越多，驾校的人员管理越来越乱，经常出现了遗失学员档案，练车预约不到，教练车闲置等现象，现在网络通信技术的发展，以及物联网潮流的推进，各个驾校的管理方式越来越简单，随之而来的是学员档案的自动化管理，练车时间的自动化分配，练车时间自动定位打卡计时，智能调配等。目前流向的驾校管理系统还存在很多问题，例如预约练车系统容易崩溃、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。

论文首先对驾校管理系统的研究背景进行了论述，然后对市面上已有系统的技术进行了研究，并结合社会现状进行了分析，在此基础上构思了本文概述的系统设计，考虑到了已有系统在稳定性、可拓展性上的缺点，并对设计成果做了展望。

论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，同时基于STM32开发板，通过FTID技术实现了练车打卡，考勤录入等功能，RFID技术具有非接触性特点，在考勤过程中具，有唯一性的特点，加上一些数据的分析处理，可以提升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的问题。本系统在驾校智能一体化管理方面具体明显性的作用，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 and the era of artificial intelligence. From the current social trend of the Internet of Things, it is an eternal topic to explore the interconnection of all things and conform to the rules of the development of the times. From an international perspective, people ’s living standards have generally improved. Automobiles have become an indispensable transportation tool for every household. Various countries have various regulations on driving and traveling. The most typical of our country is the road without a license. It is a citizen of our country. Laws and regulations that must be followed.

Driving licenses have become one of the indispensable documents in our lives. They are a guarantee of our safe travel. In modern society, there are many driving schools, many types, and business chaos. With the increasing number of "examination certificate", driving school personnel The management is becoming more and more chaotic. There are often missing trainee files, inadequate training appointments, and idle coaches. Nowadays, with the development of network communication technology and the advancement of the Internet of things, the management methods of driving schools are becoming simpler. The following are the automated management of student files, the automatic allocation of training time, the automatic positioning of the training time, the timing of clocking, and intelligent deployment. There are still many problems with the current driving school management system. For example, the car-training system is prone to crash, the page is stuck, the timing of punch-in positioning is not accurate, the student information is not stored securely, and the system is not maintainable. It is difficult to balance.

The dissertation first discusses the research background of driving school management systems, and then studies the existing system technologies in the market, and analyzes the current situation in the society. Based on this, the system design outlined in this article is conceived. The shortcomings of the system in stability and scalability, and prospects for the design results.

The thesis 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 school cars that cannot be interconnected. At the same time, based on the STM32 development board, FTID technology is used to implement functions such as train driving, time attendance and entry, RFID technology It is non-contact and unique in the attendance process. In addition to the analysis and processing of some data, it can improve the efficiency of centralized and unified management, and effectively avoid the problem of conflicts between trainees due to train time. This system has a specific and obvious role in the intelligent integrated management of driving schools. Due to the strong scope of the Internet of Things, it can be expanded at any time according to needs.

**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[[[1]](#endnote-1)]。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. 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. 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. 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，客户端和服务器端分开运行，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[[2]](#footnote-1)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. 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[[[3]](#endnote-2)]。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[[[4]](#endnote-3)]，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. 1. 研究本课题的目的和基本内容

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：

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。

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。

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

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。

本文系统中Web网络平台的基本内容如下：

服务器端采用Java语言的主流Web框架，Spring Boot搭建。

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。

服务器端使用maven一键构建、打包和部署。

前端使用nginx或者apache服务器做分离式部署。

1. 系统总体设计方案
   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：

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。

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。

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。

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[[[5]](#endnote-4)]。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。

* 1. 系统总体架构设计
     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。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

表1 用户表（driver\_user）

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和表3所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 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 |

表3 教练表（driver\_coach\_archives）

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所示。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

表4 学员教练关联表（student\_associate\_coach）

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所示，并通过外键与用户数据相绑定。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

表5 RFID卡片表（driver\_rfid\_info）

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[[6]](#footnote-2)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。

* + 1. 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所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

表 6 主要硬件清单

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

* + 1. 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所示。



图 3 Web模块原理图

1. 系统整体实现方案
   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，如表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 获取地址 | 性质 | 备注 |
| Github账户 | https://github.com/ | 免费 | 注册账户  创建远程仓库 |
| Git客户端 | https://git-scm.com/ | 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，如图所示，就具备了git版本控制器的环境。

手机屏幕的截图

描述已自动生成

* + 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，根据表所示，创建好如图所示的目录。

|  |  |
| --- | --- |
| 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 |

手机屏幕截图

描述已自动生成

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，如图所示

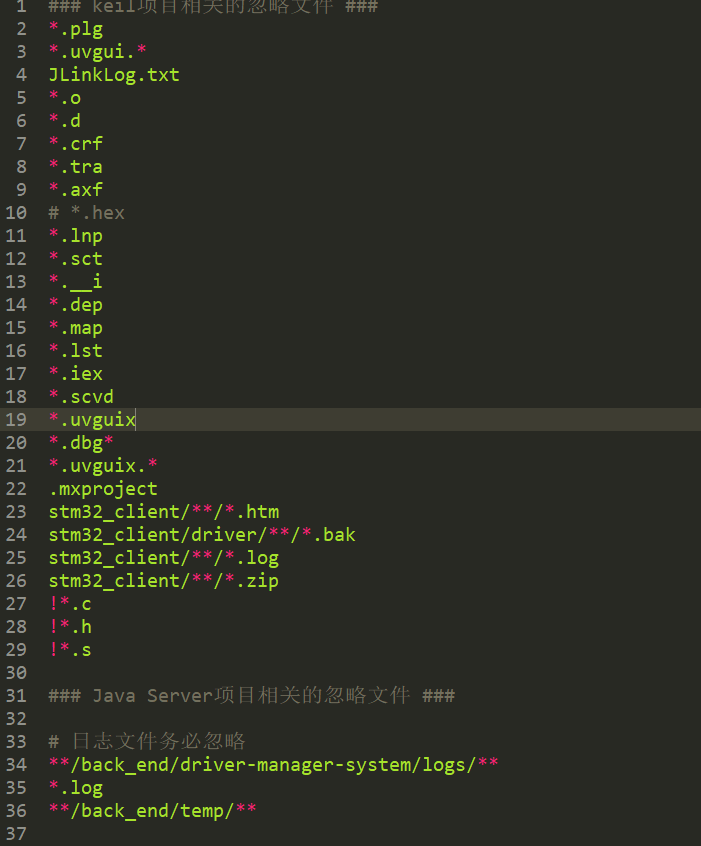
手机屏幕截图

描述已自动生成

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，如图所示。



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，如图所示。



然后将.gitignore文件同时提交到版本库中。

* + 1. 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，新建代码仓库如图所示。

手机屏幕截图

描述已自动生成

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

手机截图图社交软件的信息

描述已自动生成

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，如图所示。



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。

* + 1. 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，本地分支如图所示。

手机屏幕的截图

描述已自动生成

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。

* 1. 数据持久层实现
     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4所示，新建一个模型。

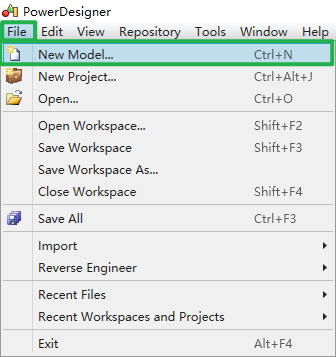


图 4 新建模型

1. 选择模型的类型为物理模型，选择方式如图5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。

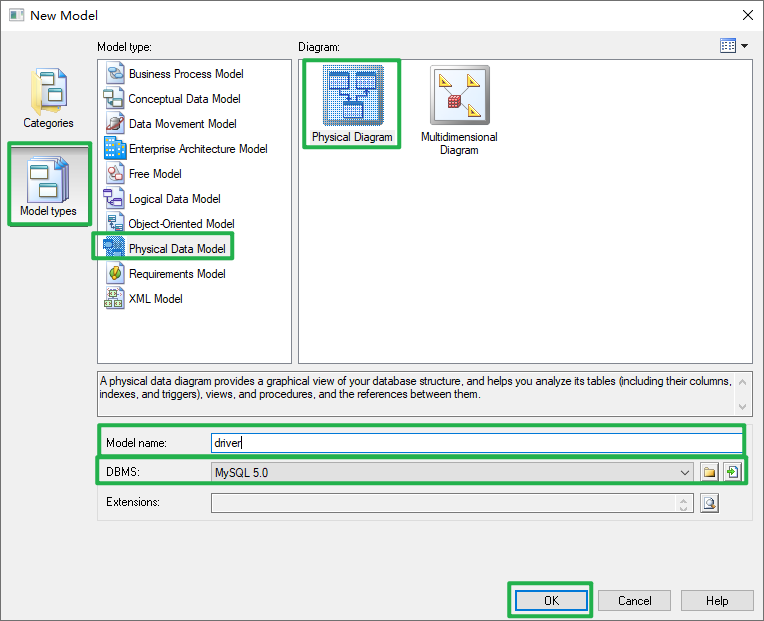


图 5 选择模型类型

1. 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6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。

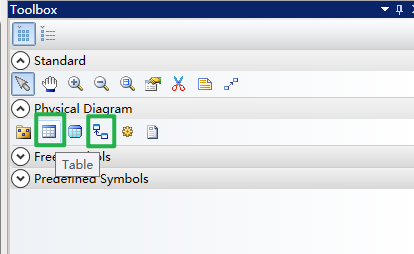


图 构建实体

1. 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7所示。

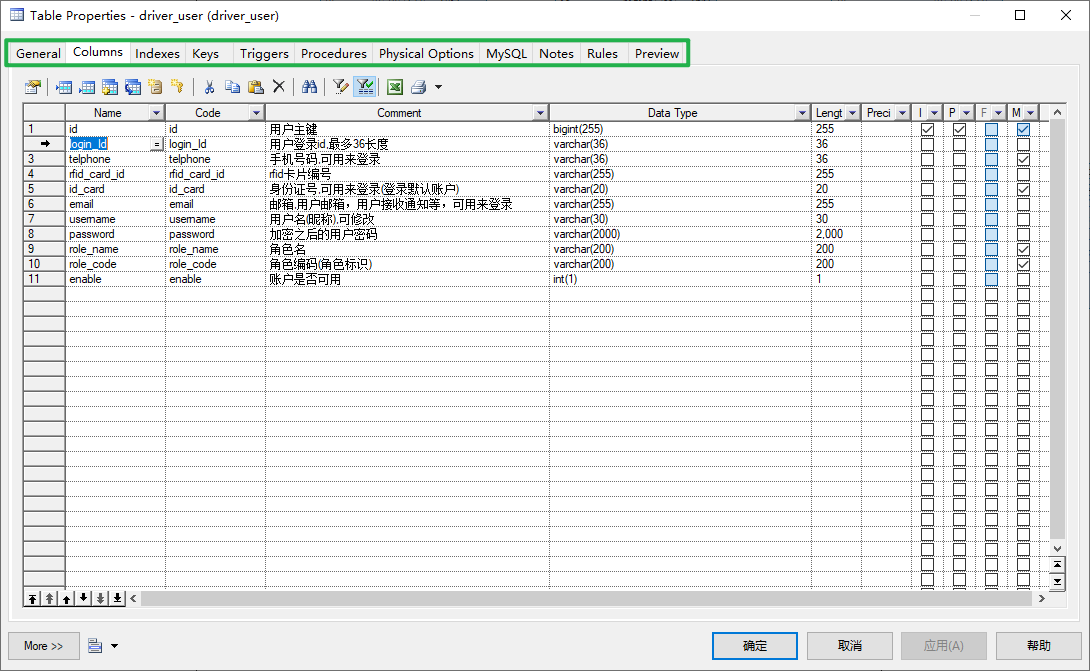


图 6 编辑Table

1. 创建完模型，按照图8方式，可直接预览物理模型对应的数据库SQL执行脚本，如图9所示。

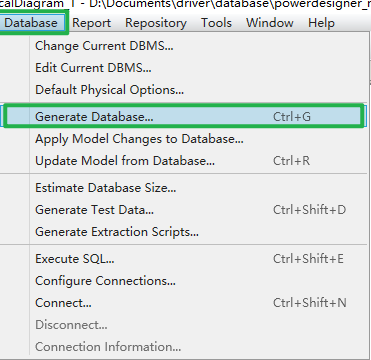


图 7 生成脚本

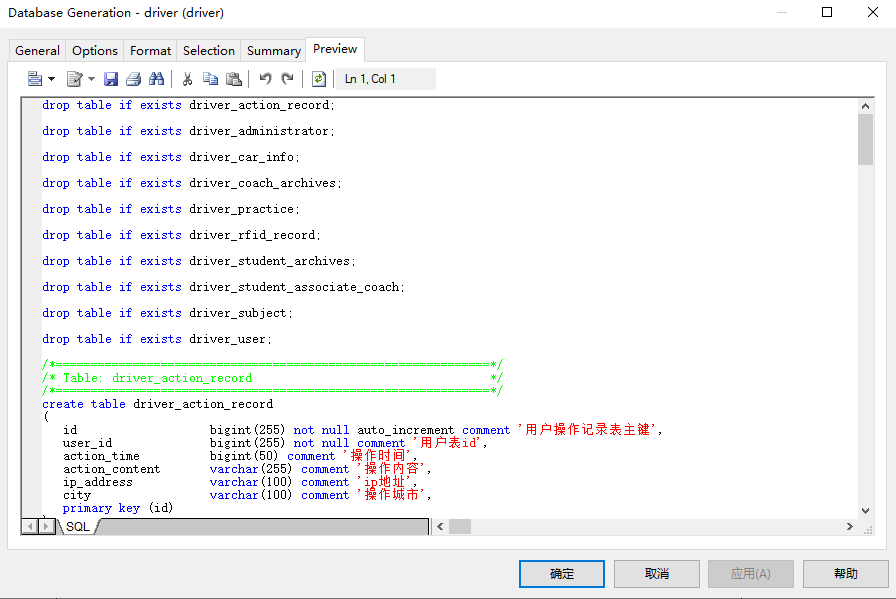


图 8 预览SQL脚本

1. 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10所示。

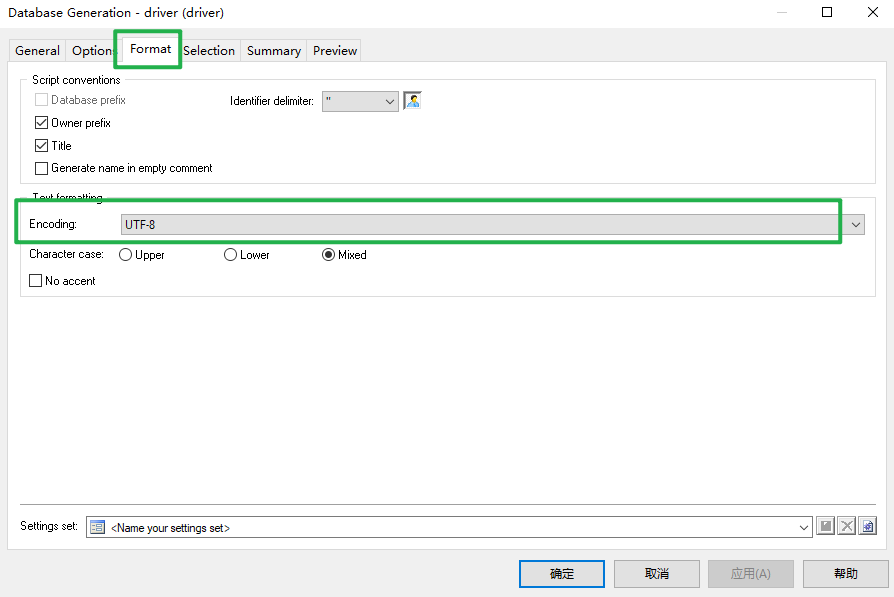


图 9 设置SQL脚本编码

1. 选择导出SQL脚本的位置以及脚本文件的名称，如图11所示。点击确定之后，就会在对应的目录下生成好文件。

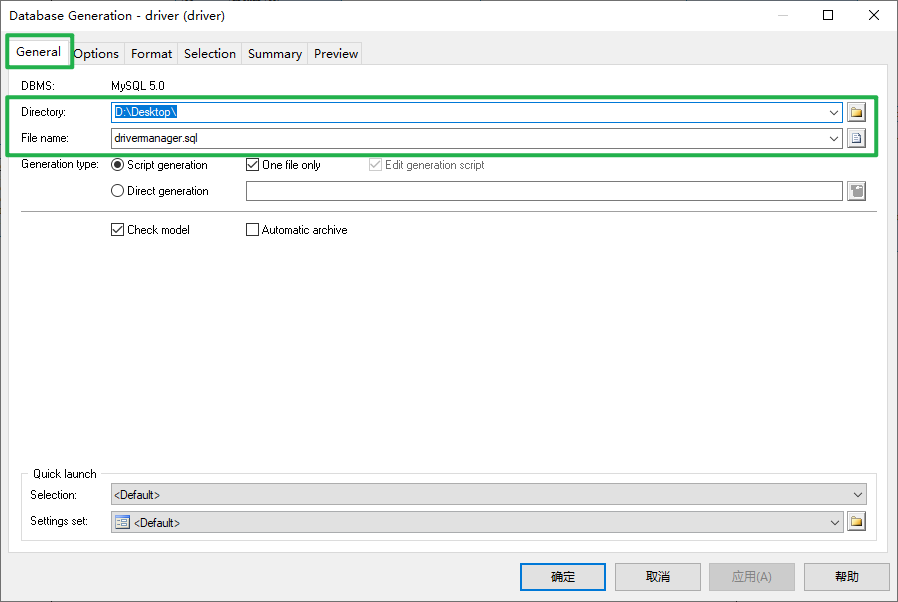


图 10 选择脚本位置设置文件名

* + 1. 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7所示的软件清单。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

表 7 数据库设计软件清单

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12所示。

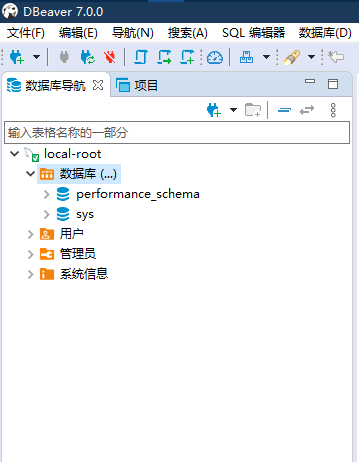


图 11 连接数据库

1. 新建开发用的数据库，保证编码为UTF-8，如图13所示。

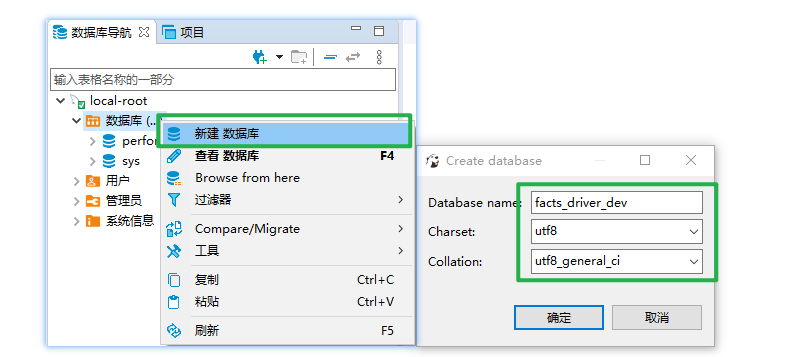


图 12 创建数据库

1. 运行数据库脚本，按照图14所示的步骤，运行数据库脚本文件，如图15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。

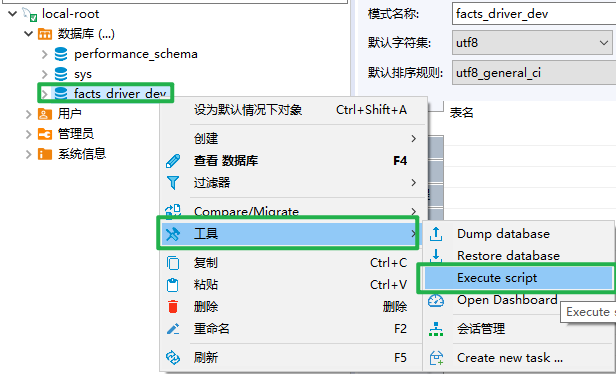


图 13 运行脚本

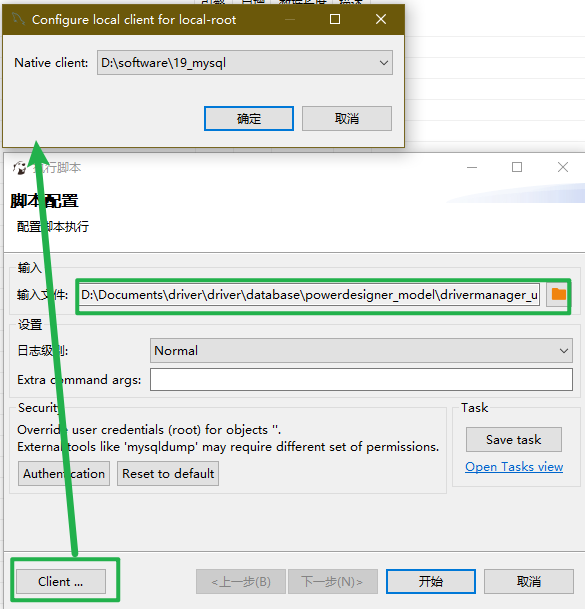


图 14 选择mysql客户端和脚本文件

1. 数据库已经创建完成，创建好的表16如图所示。

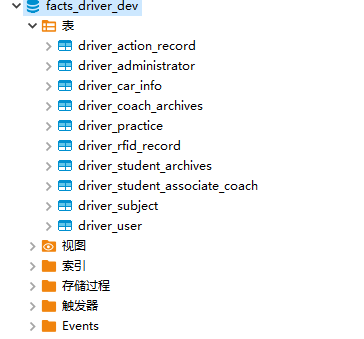


图 15 系统所需数据库的表

1. 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17所示。可以导出为图片等。

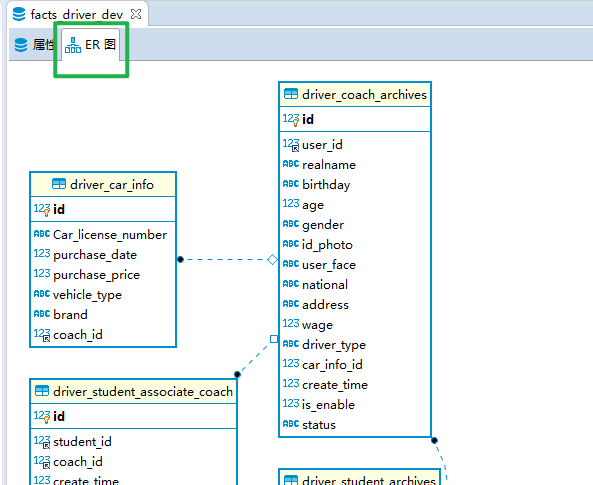


图 16 生成E-R图

系统最终使用的E-R关系图，见附录1所示。

* 1. 物联网模块实现
     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，使用Keil5[[7]](#footnote-3)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所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

* + - 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，整体外观如图所示。

尺寸

* + 1. 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 17所示。

手机屏幕截图

描述已自动生成

图 17

（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，如图所示。

手机屏幕截图

描述已自动生成

图 18

（3）创建项目工程文件夹，具体作用如表所示，创建完成后的效果如图所示。

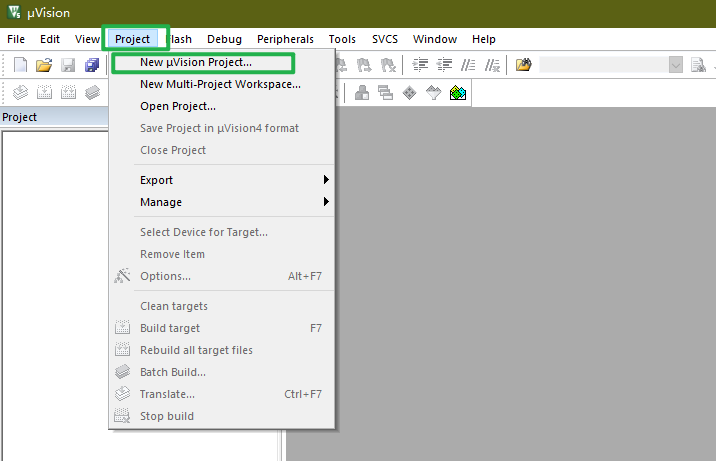
|  |  |
| --- | --- |
| 文件夹名称 | 用途 |
| CORE | 存放STM32核心文件 |
| FWLIB | 各个外设的底层程序和源码 |
| LIBRARY | 自己定义引用的STM32库文件 |
| STARTUP | STM32启动文件 |
| USER | 自定义的程序以及主程序入口文件 |

手机屏幕截图

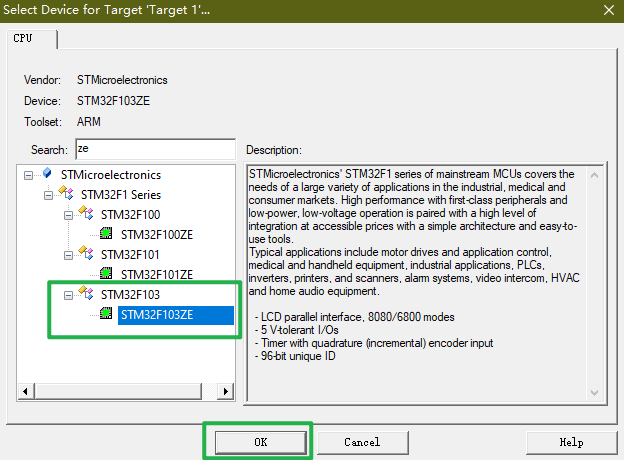
描述已自动生成

图 19

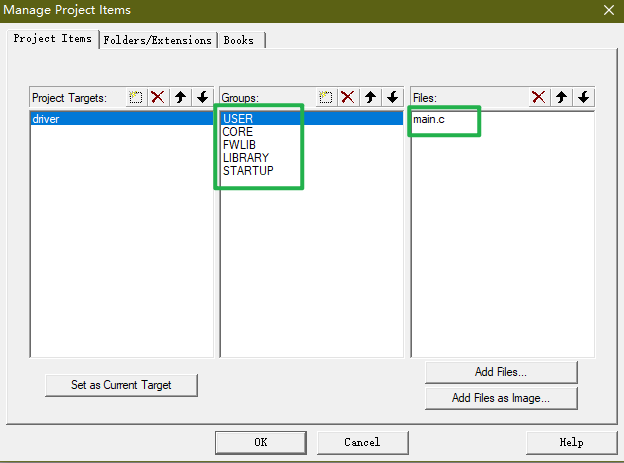
（4）再在每个文件夹下创建文件夹inc和src，inc用来存放.h头文件,src用来存放.c文件，然后使用Keil创建工程，位置选择上一步创建的工程目录，如图所示。



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，如图所示。



将项目已有的程序文件关联到项目中，如图所示。

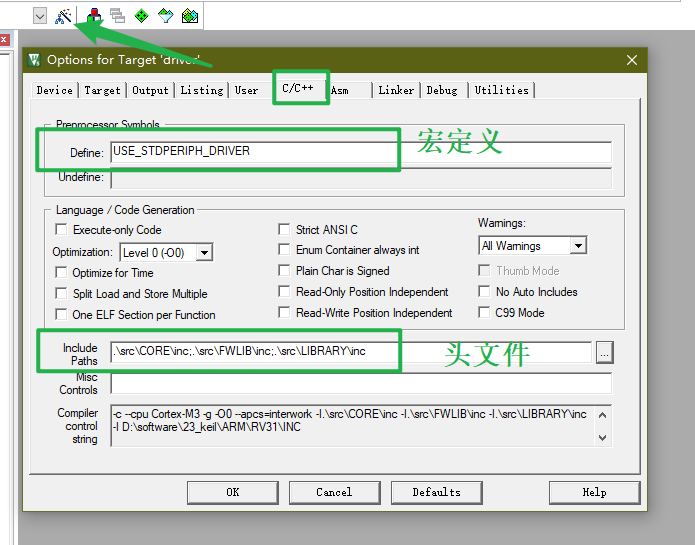


关联完成后，整体项目目录结构如图所示

手机屏幕截图

描述已自动生成

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，如图所示。



* + 1. 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。图示为

* 1. 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。

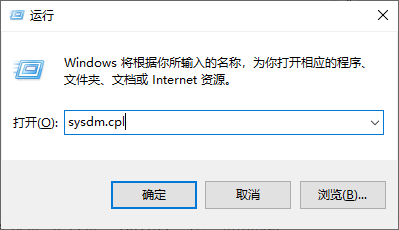


图 20 运行sysdm.cpl

如图12所示，选择高级，环境变量。



图 21 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.。

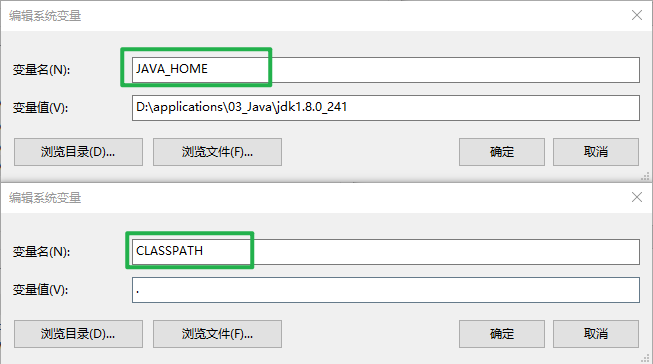


图 22 编辑环境变量

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。

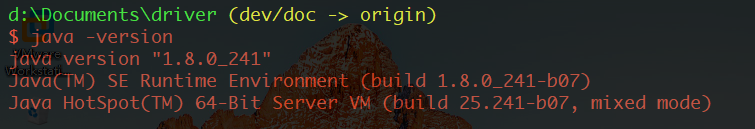
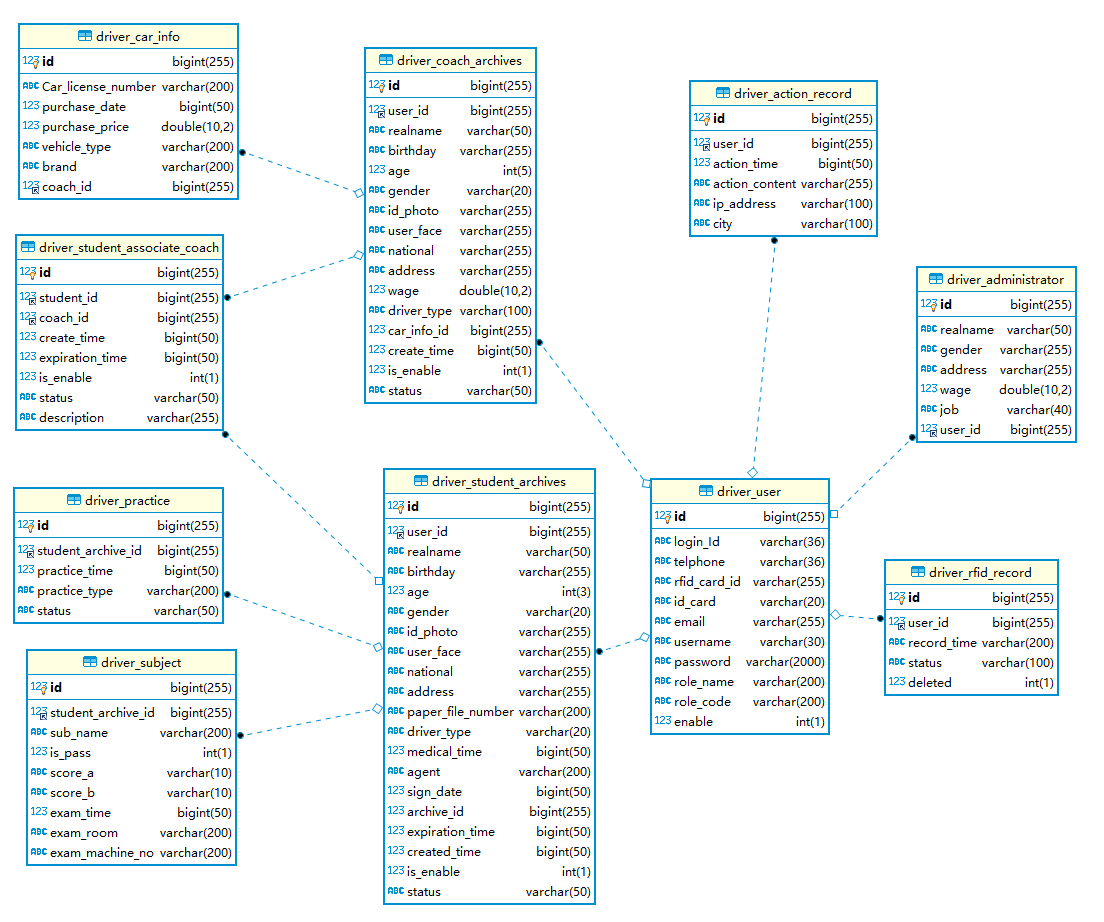


图 23 JDK安装成功验证

本系统采用的Java开发IDE为IDEA[[8]](#footnote-4)，另外还使用了maven、Tomcat、git等工具部署管理项目

**主要参考文献**

**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 |

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）. [↑](#endnote-ref-1)
2. H5，即HTML5，是HTML的最新修订版本。 [↑](#footnote-ref-1)
3. [] 徐庆. 驾校网络预约服务系统设计与实现[D]. 吉林大学. 2015. [↑](#endnote-ref-2)
4. [] 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. [↑](#endnote-ref-3)
5. [] 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. [↑](#endnote-ref-4)
6. 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。 [↑](#footnote-ref-2)
7. 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。 [↑](#footnote-ref-3)
8. 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。 [↑](#footnote-ref-4)