

面向云环境的敏捷物联网应用开发平台研究

张 腾, 金 光*, 江先亮, 朱家骅

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 针对物联网硬件设备多样性、差异化、开发环境部署繁琐过于复杂的问题, 本文设计并实现了面向云环境的敏捷物联网应用开发平台。该平台包括应用层、服务层以及设备层, 具有在线编程、远程下载、自动运行等功能。并且针对远程下载功能, 设计并实现了一套基于 MQTT 协议的代码在线推送方案及配套实验套件; 针对用户及设备安全性问题, 设计了一种基于 TEA 加密算法的动态口令认证方案。实测表明, 平台实现了预期功能, 达到了降低用户学习门槛以及提高开发效率的目的。

关键词: 云环境; 敏捷开发平台; 物联网应用; 微服务

中图分类号: TP319

文献标志码: A

文章编号: 1001-5132 (2021) 05-0016-06

随着物联网及嵌入式技术的发展, 物联网应用已遍布各行各业, 物联网应用开发教学也成为了当前信息技术教育的主流内容之一。不同于其他工程开发, 物联网应用开发中硬件及接口的差异化会造成学习者在设备选型、线路连接、环境部署时的困难, 进一步增加了学习成本及开发周期。

目前主流的快速物联网开发平台大多采用浏览器端编写代码, 云端服务器进行编译并烧录程序的方式。其中, OELab 采用 B/S 及 C/S 两种架构结合云端编译器及管理器, 实现了电源管理、代码下载及视频监控等功能^[1]。该平台针对专业嵌入式开发人员, 能够有效降低相关的测试及开发门槛, 但仍需在本地进行代码编写, 然后进行云端编译并下载代码运行。ARM mbed 可以完成基于 ARM 系列内核的开发板编程工作, 采用浏览器端代码编写并完成编译, 编译后的文件由本地 PC 端程序存入开发板并完成代码运行^[2]。Arduino Web Editor 是一款针对于 Arduino 开发套件的浏览器编程方案, 免去了开发环境的安装, 但需要通过浏览器插件进行设备连接及下载, 对系统及浏览器版本有一定要求^[3]。Tinyink 注重于降低初学者的硬件学习成本, 根据用户代码采用智能决策算法来生成所需

的硬件及其连接方案, 免去了用户设备选型的麻烦^[4]。但该系统需用户本地进行代码编写及硬件开发, 不适用于初学者。micro:bit 是一款针对于低龄物联网及嵌入式学习者的实验套件, 目前支持网页端的图形化编程及 Python 语言编程^[5]。该系统通过浏览器页面编程, 并将二进制文件下载到本地, 然后通过本地工具烧录进开发板中。该系统有较好的模块化设计, 但仍需在本地安装驱动及软件。LinkLab 通过浏览器页面进行代码编写, 并在云端完成编译, 编译后的文件发送给远程终端设备, 再通过串口完成物联网设备烧录^[6]。该方式适用于面向软件编程的用户, 缺少了硬件编程的直观性和趣味性。

综上所述, 目前缺乏一款面向初学者的免环境安装、接口统一、步骤简单、系统稳定可靠的物联网在线开发平台, 使初学者能更好地学习业务逻辑开发, 了解软件驱动与硬件之间的关系, 而不必在意开发及硬件烧录过程中的繁琐步骤。因此, 本文设计了面向云环境的敏捷物联网应用开发平台, 该平台是一个屏蔽硬件差异化的分布式远程物联网应用开发平台, 没有代码编译过程, 无需任何辅助设备及软件, 用户可直接在浏览器端进行

收稿日期: 2020-09-15.

宁波大学学报(理工版)网址: <http://journal1g.nbu.edu.cn/>

基金项目: 浙江省科技厅公益技术项目(LGG18F020007); 浙江省高等教育教学改革项目(JG20180070); 宁波市自然科学基金(202003N4085).

第一作者: 张腾(1994-), 男, 山东济宁人, 在读硕士研究生, 主要研究方向: 机器人及物联网系统开发. E-mail: ztdh88@126.com

*通信作者: 金光(1972-), 男, 浙江台州人, 教授, 主要研究方向: 无线网络及物联网. E-mail: jinguang@nbu.edu.cn

物联网开发、实验及教学,并且平台实现了在线编程、一键运行的功能,达到了屏蔽硬件、忽略差异化的目的。

1 平台架构设计

与上述技术及系统采用的硬件完全抽象化及部分步骤精简化不同,本文进一步分析了用户在学习及开发时的需求,将物联网开发中不太相关的繁琐部分进行了抽象化,并完整地保留对学习及开发者有用的部分,使得本方案能够有效地保证学习的完整性及高效性。

如图1所示,本平台与其他方案及系统相比,将环境安装、代码烧录等与物联网开发技术关联不强的繁琐程序进行抽象化,转移到云端平台进行。同时,针对于能够锻炼用户开发及具备有效体验实验效果的 WebIDE 及模块化开发板对用户可见。所以,本平台能够在极大地减少用户使用成本的同时,给予用户完整的物联网开发体验及流程,更便于用户了解物联网开发学习及实验时所需的知识点及操作步骤。

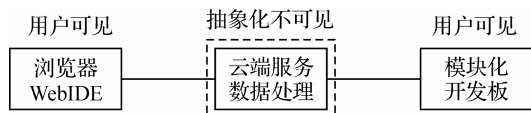


图1 平台可视化示意图

平台采用 B/S 架构,用户在浏览器端进行线上物联网实验及开发。相应系统架构如图2所示,平台分为应用层、服务层、设备层。应用层采用在线集成开发环境(Web Integrity Develop Environment, WebIDE)方式,在有效降低用户使用难度、减少学习成本的同时,达到跨平台的目的。服务层选用微服务架构^[7]进行设计和搭建,相比其他架构拥有更好的鲁棒性及扩展性,能够更好地满足用户的功能需求。设备层主要由配套实验开发板及实验模

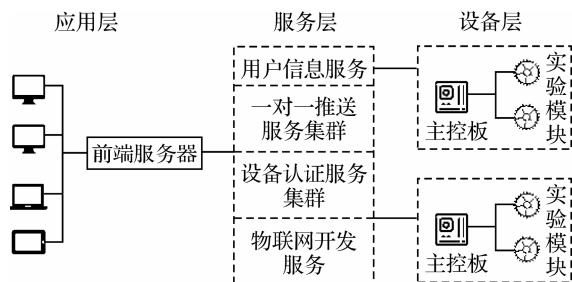


图2 平台系统架构

块组成,该实验套件具有接口统一、易用的特点,更加便于用户的学习及开发。

平台主要工作流程如图3所示。当用户将通过应用层浏览器编写完成之后的代码传送给服务层的云端微服务,其中的代码推送服务负责将代码从服务器推送到设备层开发板。手机 APP 用于第一次使用时对开发板的初始化配置工作。

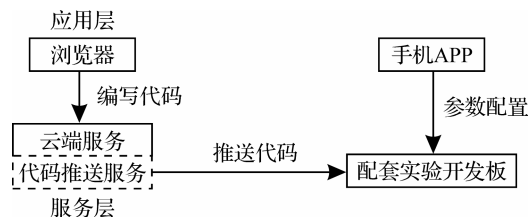


图3 系统工作流程概述

本系统无需编译流程,具体代码运行及反馈流程如图4所示。步骤1:用户在 WebIDE 编写完代码之后,前端服务通过 HTTP 协议将代码发送给图3服务层所示的一对一推送服务。步骤2:一对一推送服务对代码进行语法处理后,将代码通过 MQTT 服务器推送给开发板。步骤3:开发板运行 Python 代码后,通过控制 I/O 引脚完成物理上的实验现象演示。步骤4:通过 MQTT 协议将实验结果反馈给物联网开发服务。步骤5:物联网开发服务将数据进行格式化处理后,通过 WebSocket 协议将代码运行结果反馈给前端服务器,并在前端页面显示出来。

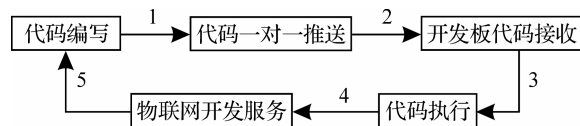


图4 代码运行及反馈流程

本平台主要功能如下:

(1)应用层及服务层设计与开发。应用层主要由前端页面及前端服务构成,包括登录注册功能的设计与开发、WebIDE 功能的设计与开发、用户中心功能的设计与开发。

服务层主要是后端微服务设计与开发,包括基于 Spring Cloud 架构进行系统架构设计,完成微服务相关组件搭建,完成鉴权认证服务、物联网开发服务、用户信息服务及一对一代码推送服务的设计与开发。

应用层前端功能与服务层对应服务见表1。

表1 应用层与服务层对应关系表

应用层	服务层
登录注册功能	鉴权认证服务
WebIDE 功能	物联网开发、一对一推动服务
用户中心功能	用户信息服务

(2)物联网代码推送机制设计. 旨在设计一种能够连接后端服务与开发板的推送机制, 该机制以微服务形式部署在线上编程系统的服务器端.

(3)设备层实验开发板设计与制作. 开发板设计与制作需针对功能要求进行 PCB(印刷电路板)的设计及制作, 并根据电路原理图完成开发板及相关外设的驱动开发及固件烧录.

针对于实验开发板参数初始化需要, 完成基于蓝牙通信手机的配套 APP 编写.

2 核心功能设计

本系统核心内容包括三部分: (1)设计适用于物联网开发环境下的一对一推送机制; (2)设计和实现支持拓展的易用性物联网实验套件; (3)设计基于 TEA 加密算法^[8]的动态口令认证协议^[9].

2.1 适用于高并发场景的一对一代码推送机制

满足高并发和一对一(一个用户针对一套开发板)的通信需要是本系统的关键点, 传统 TCP 连接服务器在合法性检测、确认重传机制的设计、对高并发量的支持等方面都无法满足实际需求. 此外, 适于物联网设备的 MQTT 协议虽具有消耗小、高并发的特点^[10], 但其协议采用发布/订阅的模式, 适于一对多通信的场景, 无法满足一对一通信要求. 针对此应用场景, 本文设计了一种基于 MQTT 的一对一代码推送机制.

如图 5 所示, 在对开发板进行配网时, 配网软件会获取开发板的唯一 ID, 当与用户 ID 进行绑定后, 存储到数据库中. 配网完成后, 开发板会订阅

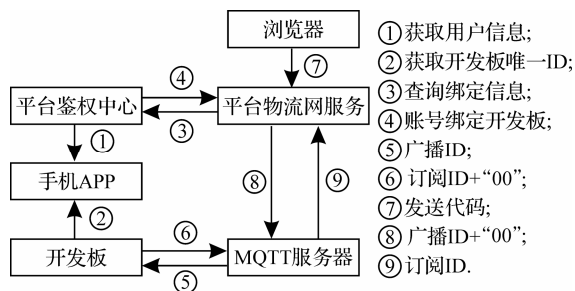


图5 一对一代码推送机制流程

以自身 ID+“00”命名的主题, 发布以自身 ID 为主题的广播. 用户完成线上代码编写, 并将代码发送至物联网服务器, 后端服务通过读取数据库以获取开发板 ID, 创建以开发板 ID+“00”为主题的广播进行代码传输, 订阅以开发板 ID 命名的主题完成开发板运行信息的接收, 达到双向一对一通信的目的.

2.2 模块化主控板电路设计

为实现用户代码完成推送后自动在开发板上的运行, 本文设计的开发板采用 MicroPython 系统为底层固件. MicroPython 是 Python 3 语言的精简高效实现, 可在微控制器和受限环境中运行^[11]. 因 Python 为解释性语言, 开发板在接收代码后, 可直接调用此 Python 代码, 实现自动运行功能.

开发板设计及开发架构如图 6 所示. 以 ESP32 为核心, 且根据需要可更换不同性能的芯片, 如 stm32、ARM 系列 Soc 等. 此外, 还根据功能需要设计了外设电路及外设模块, 且实验板与外设之间采用易用性接口连接.

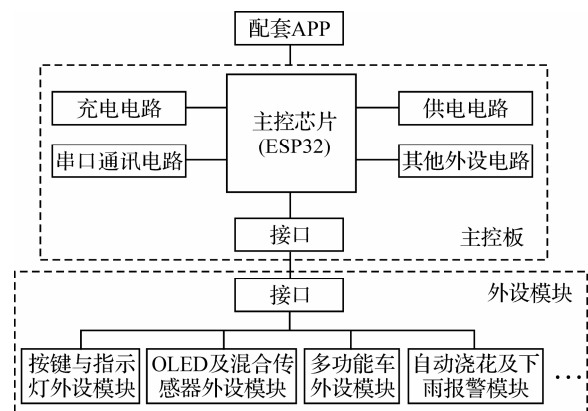


图6 开发板硬件系统结构

对于本场景, 本文针对主板的充电、稳压供电、下载通信、外设驱动等电路进行了重新设计, 其中开发板的主控芯片电路如图 7 所示.

2.3 基于 TEA 加密算法的动态口令认证协议

为防止非法设备接入及网络漏洞导致用户信息泄露等问题, 结合开发板核心性能不高的限制, 本文设计了一种基于 TEA 加密算法的动态口令认证协议.

该协议选用时间同步方式作为不确定因子, 采用 SHA256 哈希算法和 TEA 加密算法完成认证及传输加密, 以进一步提高系统安全性和破解难

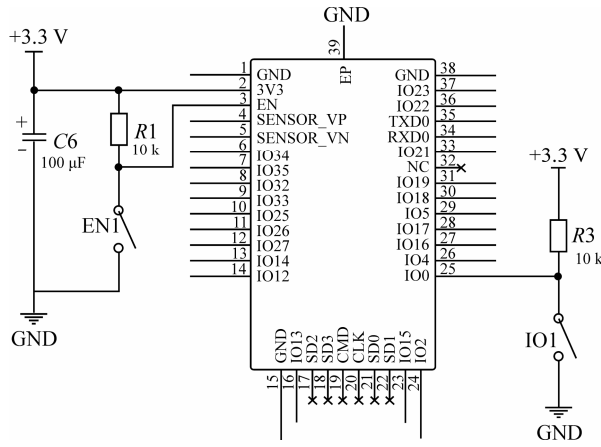


图7 主控芯片电路图

度, 本认证协议符号和相关表示见表2。

表2 认证协议算法符号和标识

符号	标识
C_i	开发板设备 i 标识
S_0	认证服务器标识
S_T	时间同步服务器标识
CID	开发板设备 ID 常量
Key	TEA 加密密钥常量
CB_MAC	开发板蓝牙 MAC 地址常量
CW_MAC	开发板 WiFi MAC 地址常量
T_C	开发板同步的时间变量
T_S	服务器同步的时间变量
REQS	SHA256 加密的请求变量
REQSE	TEA 加密后的请求数据变量
REQDE	TEA 解密后的请求数据变量
SE(x)	SHA256 加密函数
TE(x)	TEA 算法加密函数
TDE(Key, x)	TEA 解密函数

本文选用 CID、CB_MAC、CW_MAC 组合, 并

经过 SHA256 加密之后作为请求信息, 公式如下:

$$REQS = SE(CID, CB_MAC, CW_MAC). \quad (1)$$

在发送请求信息时, 将开发板获取的 T_C 、CID 信息进行 TEA 加密操作, 得到 REQSE 作为加密请求数据, 公式如下:

$$REQSE = TE(CID, T_C, REQS). \quad (2)$$

服务器接收到请求信息后, 对数据进行解密操作, 公式如下:

$$REQDE = TDE(Key, REQSE). \quad (3)$$

执行过程如图8所示。步骤1: 开发板 C_i 向时间同步服务器 S_T 获取时间。步骤2: 开发板将 CID、 T_C 数据与 SHA256 加密后的 CID、CB_MAC、CW_MAC 组合数据作为请求数据, 经过 TEA 加密算法加密后, 向认证服务器发出请求。步骤3: 服务器解密获取 T_C 、CID 及 REQSE 数据, 校验 T_C , 并比对数据库中的 REQSE 数据, 完成开发板认证。步骤4: 服务器将 T_S 以及 SHA256 加密后的 CID、CB_MAC、CW_MAC 数据发送给开发板。步骤5: 开发板验证 T_S 以及解压后的 REQSE。

步骤1: $C_i \leftarrow S_T: T_C$

步骤2: $C_i \rightarrow S_0: REQSE = TE(CID, T_C, SE(CID, CB_MAC, CW_MAC))$

步骤3: S_0 验证 C_i : $TED(Key, REQSE), T_C, CID$

步骤4: $S_0 \rightarrow C_i: REQSE = TE(T_S, SE(CID, CB_MAC, CW_MAC))$

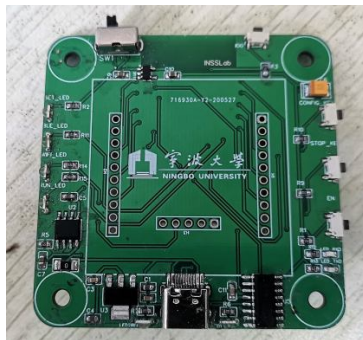
步骤5: C_i 验证 $S_0: TED(Key, REQSE), T_S$

图8 双向口令认证流程

3 系统部署及功能验证

3.1 系统部署

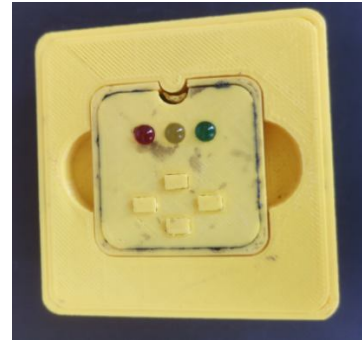
平台开发完成后, 我们将软硬件系统进行了



(a) 实验开发板电路板



(b) 样例外设电路板



(c) 开发板、外设实物

图9 开发板、外设电路及实物

部署. 软件部分, 在内存为 32 GB、CPU 为 E5-2630 V3 的服务器上使用 19.03.5 版本 Docker 容器化部署了线上编程系统各项服务, 使用 Android 6.0.1 系统安装了蓝牙配网 App. 硬件部分, 运行平台为 2.2 节所述物联网配套开发板.

我们以按键与指示灯外设模块为例进行实验测试, 以验证本方案主控电路、外设电路接口方案的可行性. 该外设模块使用易用性接口, 能更方便地完成安装和卸载工作. 具体开发板电路如图 9(a)所示, 外设模块电路如图 9(b)所示, 安装有外设模块的开发板实物如图 9(c)所示. 目前, 本系统已实现的实验套件外设模块见表 3.

3.2 功能测试

用户完成登录后, 进入线上编程界面, 输入测试代码, 并点击运行按钮开始运行程序. 待程序成功运行后, 可以看到如图 10 所示的实验现象: 开发模块上的红、黄、绿 3 种颜色 LED 灯以 1s 的间隔进行周期性闪烁.

3.3 系统对比

本文方案与 LinkLab、Arduino Web Editor、Arm Mbed 等系统的功能进行了对比. 由表 4 可见, 本系统具有跨设备、无需编译、无需软硬件辅助设备等优点, 能明显降低使用难度、提升用户体验.



图 10 配套物联网开发实验效果

4 结语

当前物联网应用系统开发门槛高、开发环境复杂, 导致物联网开发相关课程的实验教学难以在线上开展. 针对该类问题, 我们设计并实现了一套基于微服务架构的线上物联网实验平台. 考虑到已有通信协议无法满足实际需求, 同时设计了一种新的基于 MQTT 的一对一推送机制. 并根据实际需求, 设计并制作了一款基于 ESP32 的物联网开发板及手机端配网 APP. 此外, 设计了一种基于 TEA 加密算法的动态口令认证协议, 以进一步保障设备及用户信息的安全. 经过系统部署及实际功能测试, 表明该系统方案能够有效完成预期功能, 具有较好的实用价值, 为工科类尤其是物联网嵌入

表 3 目前已实现外设模块

模块名	功能	目的
按键及 LED 模块	十字按键功能及 LED 显示功能	供用户学习 GPIO 输入输出的学习, 点亮流水灯的学习等
OLED 及传感器模块	OLED 显示功能, 温湿度数据采集功能	IIC 通信协议的学习, AD 的学习, GPIO 的学习, UI 界面的学习等
智能小车模块	小车前进、转弯、后退功能	学习 PWM, 学习 GPIO 的控制, 学习完整项目的开发流程等
自动浇花及雨水报警模块	花草的自动浇水及下雨报警功能	学习传感器输入设备及输出设备的开发

表 4 本平台与主流系统功能对比

系统名	环境要求	下载、运行方式	辅助插件、烧录软件	辅助硬件、管理设备
本平台	无	在线下载、在线运行	不需要	不需要
LinkLab	无	在线下载、在线运行	管理设备需烧录环境	需要
Arduino Web Editor	需浏览器插件	本地下载、本地运行	需要	不需要
Arm Mbed	需烧录环境	本地下载、本地运行	需要	不需要
TinyLink	需烧录环境	本地下载、本地运行	需要	需要
micro:bit	需烧录环境	在线下载、本地运行	需要	不需要

式课程开展线上实验教学提供了一种新的方案。未来我们将针对该平台进一步深化设计细节, 拓展应用场景, 提升用户体验, 使其逐步得以广泛应用。

参考文献:

- [1] 杨民强. 开源嵌入式在线开发平台研究与实现[D]. 兰州: 兰州大学, 2012.
- [2] Toulson R, Wilmshurst T. Fast and effective embedded systems design: Applying the ARM mbed[M]. New York: Newnes, 2016.
- [3] Shah K, Chaudhari M P. Arduino based smart parking system[J]. International Research Journal of Engineering and Technology, 2017, 4(1):882-884.
- [4] Guan G, Dong W, Gao Y, et al. Tinylink: A holistic system for rapid development of iot applications[C]//Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, 2017.
- [5] Sentance S, Waite J, Hodges S, et al. "Creating cool stuff" — Pupils' experience of the BBC micro:bit[C]//Proceedings of the 48th ACM Technical Symposium on Computer Science Education (SIGCSE), 2017.
- [6] Gao Y, Zhang J, Guan G, et al. LinkLab: A scalable and heterogeneous testbed for remotely developing and experimenting IoT applications[C]//2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI), 2020.
- [7] 官东亮. 基于微服务的业务平台架构重构[J]. 电信科学, 2020(9):75-83.
- [8] Budiyo D, Putro P A W. Comparison of implementation Tiny Encryption Algorithm (TEA) and Advanced Encryption Standard (AES) algorithm on android based open source cryptomator library[C]//2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018.
- [9] 沈斌表, 毛海波, 刘柏嵩. 基于 PKI 的校园移动应用身份认证设计[J]. 宁波大学学报(理工版), 2014, 27(2):55-58.
- [10] Manandhar S. MQTT based communication in IoT[D]. Finland, Tampere: Tampere University of Technology, 2017.
- [11] Khamphroo M, Kwankeo N, Kaemarungsi K, et al. MicroPython-based educational mobile robot for computer coding learning[C]//2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), 2017.

An agile development platform for Internet-of-things applications in cloud environment

ZHANG Teng, JIN Guang^{*}, JIANG Xianliang, ZHU Jiahua

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: Beginners have been facing challenges in IoT development and applications due to the existing hardware diversity and complexity. The deployment under current circumstances proves to be inefficient and error-prone. To address this issue, an agile development platform for IoT in cloud environment is proposed. The platform is designed to achieve functions including the cross-device, interpretive execution, online running, etc. To implement the online code downloading, a fine-tuned MQTT protocol is adopted. In addition, a set of experiment kits are designed as a supporting system, all of which serve the functions of online coding and the real-time running on devices. To reduce risk of the access of unauthorized devices and the leak of user information, an authentication method based on the TEA encryption algorithm is presented. Compared with existing schemes, the platform has the advantages of simplicity, friendliness, and can improve the efficiency of IoT development.

Key words: cloud environment; agile development platform; IoT applications; microservice

(责任编辑 章践立)