

基于 RISC-V 处理器内核微控制器的智能教学平台^①



陈宏铭¹, 王宜怀², 杨 勇³, 施连敏², 彭 彤¹

¹(浙江海洋大学 信息工程学院, 舟山 316022)

²(苏州大学 计算机科学与技术学院, 苏州 215006)

³(南京沁恒微电子股份有限公司, 南京 210012)

通信作者: 陈宏铭, E-mail: buildgates@zjou.edu.cn

摘 要: 本文以沁恒微电子内置 RISC-V 处理器内核的 CH32V307 微控制器为基础, 利用 AHL-CH32V307 硬件系统阐述嵌入式技术实践。首先简要介绍嵌入式系统的知识体系, 降低具有高技术门槛的嵌入式系统开发工作, 实现嵌入式人工智能的敏捷开发生态系统。接着给出并测试嵌入式开发硬件, 以多功能嵌入式集成开发环境编译、下载与运行第一个嵌入式程序的直观体验, 学员可以开始进入嵌入式系统的学习之旅。对应开发套件的硬件系统, 文中给出一些常见嵌入式系统被控单元如彩灯、红外传感器与汇编工程的树型结构的基本原理、电路接法和编程实践。利用 CH32V307 微控制器设计一套简单实用基于图像识别的嵌入式物体认知系统, 可作为人工智能的快速入门系统。本文所介绍的教学案例适用于高等学校嵌入式系统的教学或技术培训, 也可以提供给嵌入式系统技术人员作为研发参考。

关键词: RISC-V 处理器; 微控制器; 嵌入式系统; 嵌入式人工智能

引用格式: 陈宏铭, 王宜怀, 杨勇, 施连敏, 彭彤. 基于 RISC-V 处理器内核微控制器的智能教学平台. 计算机系统应用, 2022, 31(9): 50-56. <http://www.c-s-a.org.cn/1003-3254/8839.html>

Intelligent Teaching Platform Based on Microcontroller with RISC-V Processor Core

CHEN Hong-Ming¹, WANG Yi-Huai², YANG Yong³, SHI Lian-Min², PENG Tong¹

¹(School of Information Engineering, Zhejiang Ocean University, Zhoushan 316022, China)

²(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

³(Nanjing Qinheng Microelectronics Co. Ltd., Nanjing 210012, China)

Abstract: This study expounds on the practice of embedded technology by using the AHL-CH32V307 hardware system based on the CH32V307 microcontroller with a RISC-V processor core from Nanjing Qinheng Microelectronics. Firstly, the study briefly introduces the knowledge system of the embedded system, reduces the development work of the embedded system with a high-tech threshold, and realizes the agile development ecosystem of embedded artificial intelligence. Then, the embedded development hardware is given and tested. With the intuitive experience of compiling, downloading, and running the first embedded program in a multi-functional embedded integrated development environment, students can start their journey to learn embedded systems. In terms of the hardware system corresponding to the development kit, this study describes the basic principle, circuit connection, and programming practice of some common controlled units in the embedded system, such as color lights, infrared sensors, and the tree structure of assembly project. Furthermore, a simple and practical embedded object recognition system based on image recognition is designed by using the CH32V307 microcontroller, and it can be used as a quick-start system of artificial intelligence. The teaching

^① 本文由“RISC-V 技术及生态”专题特约编辑武延军研究员、宋威副研究员、张科高级工程师以及邢明杰高级工程师推荐。

收稿时间: 2022-01-31; 修改时间: 2022-04-25, 2022-05-16; 采用时间: 2022-06-01; csa 在线出版时间: 2022-07-22

cases in this study are applicable to the teaching or technical training of embedded systems in colleges and universities, and they can provide a reference for application technicians in developing embedded systems.

Key words: RISC-V processor; microcontroller; embedded system; embedded artificial intelligence

嵌入式计算机系统最初源于传统测控系统对计算机的需求,随着以微处理器(mini processor unit, MPU)为内核的微控制器(microcontroller unit, MCU)制造技术不断进步,计算机领域在通用计算机系统与嵌入式计算机系统这两大分支分别得以发展.通用计算机已经在科学计算、通信、日常生活等各个领域产生重要影响.在后PC时代,嵌入式系统的广阔应用是计算机发展的重要特征.嵌入式系统的技术基础是要完成一个嵌入式系统产品的设计,需要有硬件、软件及行业领域相关知识.但是,随着嵌入式系统中软件规模日益增大,对嵌入式底层驱动软件的封装提出了更高的要求,可复用性与可移植性受到特别的关注,嵌入式软件构件化开发方法逐步被业界所重视^[1].

2015年以来,RISC-V架构处理器在我国快速发展^[2-5],基于沁恒微电子RISC-V架构CH32V307微控制器构建通用嵌入式计算机(general embedded computer, GEC)生态系统,形成相对完备的教学及实践系统.配合可以直接实践的硬件系统AHL-CH32V307,具备简洁、便利、边学边实践等优点,克服了实验箱模式的冗余、不方便带出实验室、不易升级等缺点,逐步探索嵌入式教学新模式^[6].

1 从概念到实践通用嵌入式计算机 GEC

1.1 嵌入式系统开发具有较高技术门槛

智能终端及系统开发具有较高技术门槛.芯片生产厂家往往提供用户一份长篇幅的参考手册,少则几百页,多则可达近千页.许多厂家也给出庞大软件开发工具包(software development kit, SDK),但设计人员需要花费许多精力从中提取出所需要的信息.智能终端开发人员通常花费太多的精力在基于芯片级硬件设计及基于寄存器级的底层驱动设计上,开发方式存在软件开发颗粒度低、可移植性弱等问题.

特别是系统开发时,会涉及到许多复杂技术问题,如传感器应用设计、微控制器编程、终端(user equipment, UE)通信、数据库系统、个人计算机(personal computer, PC)的程序设计、人机交互系统(human

computer interaction, HCI)的软件设计等,具有较高技术门槛,是许多中小企业技术转型的重要瓶颈之一.

1.2 降低门槛的方法

遵循人的认识过程由研究共性技术,实现降低门槛之目标.为此,需要深入研究3个提高的方法:

(1) 提高硬件设计的颗粒度.需要把硬件设计元件级过渡到硬件构件级为主的设计场景.

(2) 提高软件编程颗粒度.需要把软件编程颗粒度从寄存器级提高到知识要素级.

(3) 提高软硬件可移植性.需要把嵌入式软件工程基本原理应用其中,提高软硬件的可移植性.

在此基础上,建立其相应的生态系统,就可有效降低嵌入式开发的技术门槛.

1.3 共性抽取的研究过程

1.3.1 如何完成3个提高

(1) 如何提高硬件设计的颗粒度?若能能将MCU硬件最小系统及面向领域应用的共性电路封装成一个整体,则可提高UE的硬件设计颗粒度.硬件设计则可从元件级过渡到硬件构件级为主的场景.

(2) 如何提高软件编程颗粒度?若能设计出基于模块共性知识要素的构件,屏蔽寄存器级之差异,则可把软件编程颗粒度从寄存器级提高到知识要素级.

(3) 如何提高软硬件可移植性?若能从共性知识要素角度研究可移植性问题,屏蔽芯片生产厂家、编译器类型等因素,遵循软硬件构件的设计原则,则可提高软硬件的可移植性.

1.3.2 提出GEC概念的时机

GEC概念的实质是把面向寄存器编程提高到面向知识要素编程,提高了编程颗粒度.但是,这样做也会降低实时性.弥补实时性降低的方法是提高芯片的运行时钟频率.目前MCU的总线频率是早期MCU总线频率几十倍,因此,更高的总线频率给提高编程颗粒度提供了物理支撑.

另一方面是软件构件技术的发展与认识的普及,也为提出GEC概念提供了机遇.嵌入式软件开发人员越来越认识到软件工程对软件开发的重要支撑作用,

也意识到掌握和应用软件工程的基本原理对嵌入式软件的设计、升级、芯片迭代与维护等方面,具有不可或缺的作用.因此,从“零”开始的编程,将逐步分化为构件制作与构件使用两个不同层次,也为嵌入式人工智能提供先导基础.

1.3.3 GEC 概念的特点

与 MCU 相比, GEC 具有硬件可测性、编程快捷性与可移植性等基本特点.

(1) 硬件直接可测性. 与一般 MCU 不同, GEC 类似 PC 机, 通电后可直接可运行内部基本输入输出系统 (basic input output system, BIOS) 程序, user 框架程序中包含类似于 PC 程序调试的 printf 语句, 通过串口向 PC 机输出信息, 实现了 GEC 硬件的直接可测性.

(2) 用户软件编程快捷性. 与一般 MCU 不同, GEC 内部驻留的 BIOS 与 PC 机上电过程类似, BIOS 内驻留了嵌入式常用驱动及 RTOS, 并提供了函数原型级调用接口. 利用 user 程序不同框架, 用户软件不需要从“零”编起, 而是在相应框架基础上, 充分应用 BIOS 资源, 实现快捷编程.

(3) 用户软件可移植性. 与一般 MCU 不同, GEC 的 BIOS 软件由 GEC 提供者研发完成, 随 GEC 芯片而提供给用户, 即软件被硬件化了, 具有通用性. 在此基础上, 只要遵循软件工程基本原则, GEC 用户软件则具有较高的可移植性.

1.4 主要解决问题

这里提到解决的问题是第 2 节的基础技术, 有了这些技术才能实现第 2 节的通用编程框架. 这也是符合软件工程原理的框架, 为实现嵌入式产品的应用层软件可移植性提供了技术基础.

1.4.1 解决 GEC 硬件共性结构及 BIOS 驻留机制

(1) 解决基于硬件构件概念研究 GEC 硬件设计方法, 结合 BIOS 驻留, 为将来 GEC 流片提供技术基础.

(2) 解决 BIOS 功能定位, 即确定内核时钟与总线时钟、启动一个定时器, 提供时间函数调用接口、给出面向知识要素的常用底层驱动驻留与调用方式、启动 RTOS 及其他功能.

(3) 解决从 BIOS 转入 user 程序执行, 在 user 正常运行情况下, 转入 BIOS 的机制.

(4) 解决实时操作系统 (real-time operating system, RTOS) 的驻留方法、映射接口模式及统一 (application programming interface, API) 接口格式, 为实现不同

RTOS 下程序可移植提供技术支撑.

1.4.2 解决基于串口的下载与调试方法

为实现类 PC 机程序的开发模式, 需要:

(1) 解决基于串口的下载、调试模式, 同时提出类似 PC 机程序开发中的 printf 语句功能, 从而实现类 PC 机程序的开发模式. 这里涉及中断继承与重定位机制、内存分割机制等. 探索基于串口的断点调试模式方法.

(2) 解决动态命令机制, 为 user 功能的在线扩展提供支撑. 研究多通信模式下的程序更新机制及远程调试方法. 动态命令机制 (dynamic command mechanism, DCM) 机制是指在不影响嵌入式终端正常运行的前提下, 将一段具有特定功能的机器码通过一定的通信方式临时注入至指定存储区域并运行的机制. 这一段临时的、具有一定功能的机器码就称为动态命令.

(3) 解决针对 NB-IoT、Cat1、4G、5G、WSN、WiFi 等无线通信方式下的程序更新方法.

1.4.3 解决 GEC 开发生态系统

针对国外 IDE 的芯片相关性:

(1) 解决面向不同 MCU 内核的统一编译模式的集成开发环境的设计方法, 研究基于软件工程基本原理的 user 程序框架, 使得用户程序可移植.

(2) 解决面向 NB-IoT、Cat1、4G、5G、WSN、WiFi 通信的共性编程框架, 形成基于 GEC 体系面向物联网的快速开发生态系统, 为降低开发门槛、节约开发成本、缩短开发时间提供技术支撑.

(3) 解决面向基于 GEC 体系面向嵌入式人工智能的快速开发方法, 实现嵌入式人工智能的敏捷开发生态系统.

2 嵌入式系统的实现

嵌入式系统不单独以通用计算机面目出现, 而是隐含在各类具体的智能产品中, 如手机、机器人、自动驾驶系统等. 嵌入式系统在嵌入式人工智能、物联网、工厂智能化等产品中起核心作用.

由于嵌入式系统是一门理论与实践密切结合的课程, 教学配合基于 RISC-V 架构 AHL-CH32V307 嵌入式开发套件 (图 1). AHL-CH32V307 体积很小, 学员可以利用这个套件的硬件、软件、文档、开发环境等资源, 用于嵌入式系统入门学习.

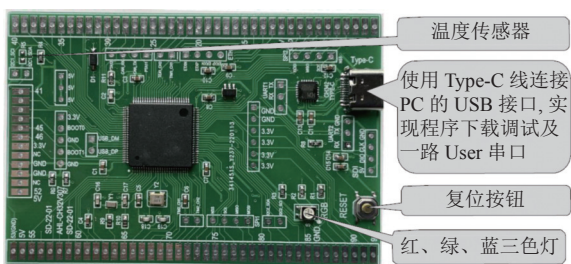


图1 AHL-CH32V307 嵌入式开发套件

本文所设计的嵌入式教学系统,通过通用嵌入式计算 GEC 概念,将 RISC-V 指令集计算机衔接到打造多年的 ARM 体系.可以复用 ARM 体系的生态资源,实现 RISC-V 指令集生态系统的快速建立与推广应用.同时,给出了适应 RISC-V 指令集的汇编程序框架、操作系统的移植与驻留、printf 函数打桩调试等快速实现.因此,所设计的嵌入式教学系统体现 RISC-V 指令集的特点与优势.

AHL-CH32V307 以 CH32V307VCT6 微控制器为核心,辅以硬件最小系统集成红、绿、蓝三色灯,复位按钮,二路 TTL-USB 串口,外接 Type-C 线,从而形成完整的通用嵌入式计算机.为了满足学校实验室建设要求,还制作增强型套件,增加了 9 个外接组件,包括声音传感器、加速度传感器、人体红外传感器、循迹传感器、振动马达、蜂鸣器、四按钮模块、彩灯及数码管等,可完成物联网相关实验.亦可通过主板上的开放式外围引脚,外接其他接口模块进行创新性实验.该开发套件与一般嵌入式系统实验箱不同,不仅可以作为嵌入式系统教学使用,还是一套较为完备的嵌入式微型计算机应用开发系统.

2.1 测试嵌入式开发硬件

因为出厂时已经将测试程序烧录到嵌入式计算机内,只要给供电嵌入式开发套件内的程序就可以运行,从运行效果可以体会到小小嵌入式计算机的功能.

步骤 1. 使用 Type-C 数据线给主板供电.将 Type-C 数据线的小端连接主板,另外一端接可以是笔记本电脑或个人计算机 PC 的工具机 USB 接口.

步骤 2. 观察程序运行效果.现象如下:红、绿、蓝各灯每 5 s、10 s、20 s 状态变化,对外表现为三色灯的合成色,其实际效果如图 2 所示.即开始时为暗,依次变化为红、绿、黄(红+绿)、蓝、紫(红+蓝)、青(蓝+绿)、白(红+蓝+绿),周而复始.

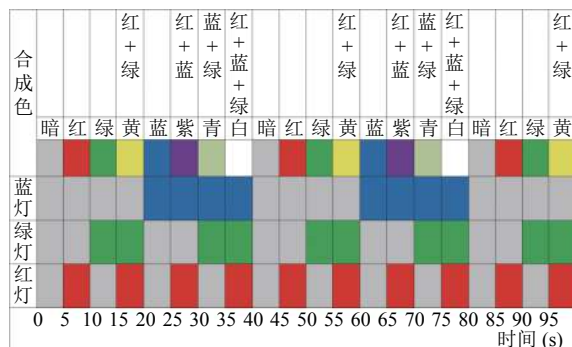


图2 三色灯实际效果

2.2 集成开发环境

嵌入式软件开发有别于个人计算机(PC),软件开发的一个显著的特点在于需要一个交叉编译和调试环境,即工程的编辑和编译所使用的工具软件通常在 PC 上运行,这个工具软件通常称为集成开发环境(integrated development environment, IDE),而编译生成的嵌入式软件的机器码文件则需要通过写入工具下载到目标机上执行,本文的目标机就是 AHL-CH32V307 开发套件.

本文介绍的集成开发环境为 AHL-GEC-IDE,具有编辑、编译、链接等功能.配合硬件可直接运行、调试程序,根据芯片型号不同兼容常用嵌入式集成开发环境.

2.3 编译、下载与运行第一个嵌入式程序

在正确安装 AHL-GEC-IDE 可以进行第一个嵌入式程序编译、下载与运行,以便直观体会嵌入式程序的运行.

步骤 1. 硬件接线.将 Type-C 数据线的小端连接主板的 Type-C 接口,另外一端接通用计算机的 USB 接口.

步骤 2. 打开环境,导入工程.打开集成开发环境 AHL-GEC-IDE,单击菜单“文件”→“导入工程”,选择“AHL-CH32V307-Test”.导入工程后,左侧为工程树形目录,右侧为文件内容编辑区,初始显示 main.c 文件内容,如图 3 所示.

步骤 3. 编译工程.单击菜单“编译”→“编译工程”,就开始编译.正常情况下,编译后会显示“编译成功!”.

步骤 4. 连接 GEC.单击菜单“下载”→“串口更新”,将进入更新窗体界面.单击“连接 GEC”查找目标 GEC,若提示“成功连接...”,可进行下一步操作.

步骤 5. 下载机器码.单击“选择文件”按钮,导入被编译工程目录下 Debug 中的.hex 文件,然后单击“一键

自动更新”按钮,等待程序自动更新完成.当更新完成之后,程序将自动运行.

步骤 6. 通过串口观察运行情况.

① 观察程序运行过程. 在 IDE 的顶部菜单栏目, 单击“工具”→“串口工具”, 选择其中一个串口, 波特率设为 115200 并打开, 串口调试工具页面会显示三色灯的状态、温度等信息;

② 验证串口收发. 关闭已经打开的串口, 打开另一个串口, 波特率选择默认参数, 在“发送数据框”中输入字符串, 单击“发送数据”按钮. 正常情况下, 主板会回送数据给 PC, 并在接收框中显示, 效果如图 4 所示.

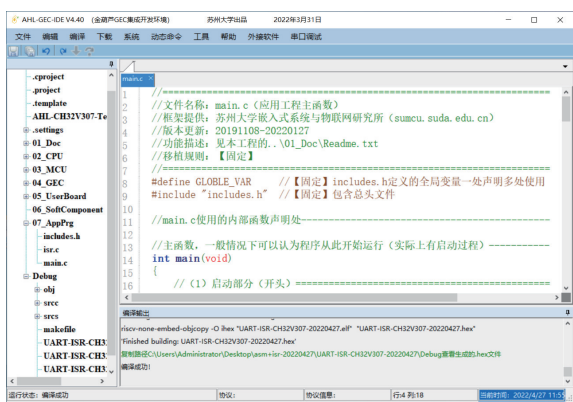


图 3 IDE 界面及编译结果

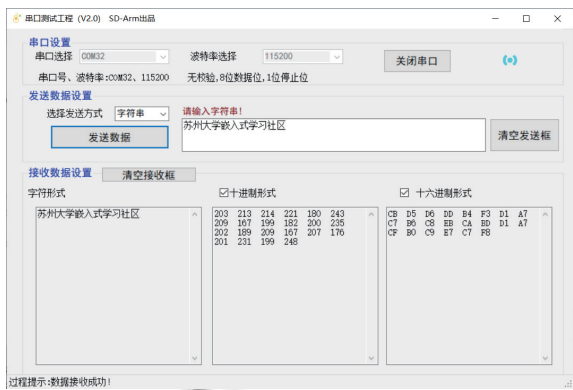


图 4 IDE 内嵌的串口调试工具

3 外接传感器及执行部件的编程方法

对应 AHL-CH32V307 开发套件的硬件系统, 以下给出一些常见嵌入式系统被控单元 (传感器) 的基本原理、电路接法和编程实践.

3.1 开关量输出类驱动构件

3.1.1 彩灯

彩灯的控制电路与 RGB 芯片集成在一个 5050 封

装的元器件中, 构成了一个完整的外控像素点, 每个像素点的三基色颜色可实现 256 级亮度显示.

像素点内部包含了智能数字接口数据锁存信号整形放大驱动电路、高精度的内部振荡器和可编程定电流控制部分, 有效保证像素点光的颜色高度一致, 数据协议采用单线归零码的通讯方式, 通过发送具有特定占空比的高电平和低电平来控制彩灯的亮暗.

如图 5 所示, 彩灯使用串行级联接口, 能够通过一根信号线完成数据的接收与解码. VDD 是电源端, 用于供电; DOUT 是数据输出端, 用于控制数据信号输出; VSS 接地; DIN 控制数据信号的输入端.

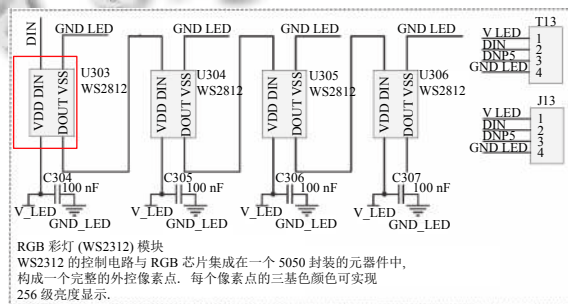


图 5 彩灯电路原理

3.1.2 人体红外传感器

任何发热体都会产生红外线, 辐射的红外线波长跟物体温度有关, 表面温度越高, 辐射能量越强. 人体都有恒定的体温, 所以会发出特定波长 10 μm 左右的红外线, 人体红外传感器通过检测人体释放的红外信号, 判断一定范围内是否有人体活动^[7,8]. 默认输出是低电平, 当传感器检测到人体运动时, 会触发高电平输出, 小灯亮有 3 s 左右延迟.

如图 6 所示, V_PIR1 用于供电; REF 为输出引脚; 实验现象: 当用手靠近靠近人体红外传感器, 红灯亮; 远离, 延迟 3 s 左右, 红灯灭.

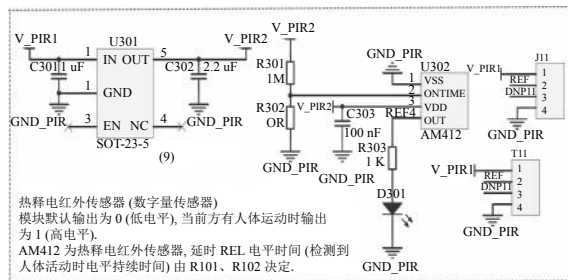


图 6 人体红外传感器电路原理

3.1.3 第一个汇编语言工程: 控制小灯闪烁

GPIO 汇编程序是一个有 RISC-V 特色的实验, 相对于 C 语言编程, 汇编在编程的直观性、编程效率以及可读性等方面都有所欠缺, 控制小灯闪烁的汇编语言编程方法在开发环境中, 程序是通过工程的方式组织起来的. 汇编工程通常包含芯片相关的程序框架文件、软件构件文件、工程设置文件、主程序文件及抽象构件文件等.

图 7 给出小灯闪烁汇编工程的树型结构, 主要包括 MCU 相关头文件夹、底层驱动构件文件夹、Debug 工程输出文件夹、程序文件夹等. 汇编工程仅包含一个汇编主程序文件, 该文件名固定为 main.s, 包含有以下内容:

- (1) 工程描述. 工程名、程序描述、版本、日期等.
- (2) 包含总头文件. 声明全局变量和包含主程序文件中需要的头文件、宏定义等.
- (3) 主程序. 主程序一般包括初始化与主循环两大部分. 初始化包括堆栈初始化、系统初始化、I/O 端口初始化、中断初始化等.
- (4) 内部直接调用子程序.

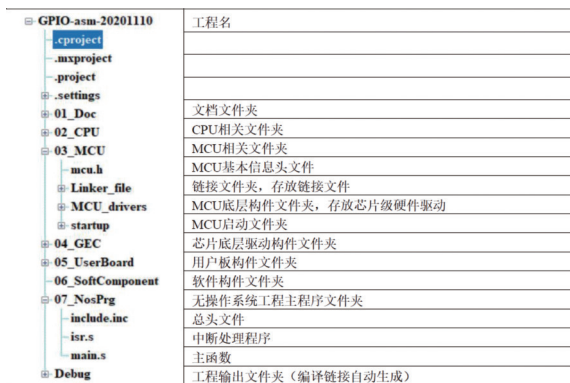


图 7 小灯闪烁汇编工程的树型结构

4 嵌入式人工智能简明实例

嵌入式人工智能是指将人工智能的基本学习或推理算法应用于嵌入式产品, 利用 CH32V307 微控制器设计了一套成本低、原理清晰、简单实用, 基于图像识别的嵌入式物体认知系统, 可作为人工智能的快速入门系统. 本文实现的嵌入式人工智能教学平台总体结构如图 8 所示. 在整体系统中以微控制器为核心, 通过硬件最小系统保证能够基本运行, 使用摄像头模块采集图像数据, 并使用通信模组提供远程通信功能, 并

与电源转换电路等模块共同构成了硬件平台. 识别终端同时也提供其他不同的外设传感器等输入输出设备使用接口, 实现不同执行机制的并行使用, 完成对物体的认知以及结果输出^[9,10].

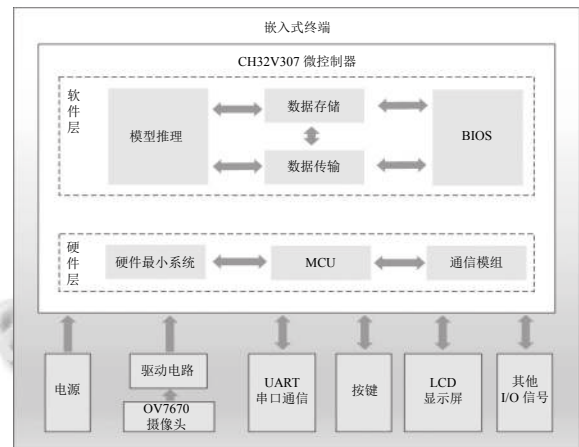


图 8 嵌入式人工智能教学平台总体结构

基于图像识别的嵌入式物体认知系统是利用嵌入式计算机通过摄像头采集物体图像, 利用图像识别相关算法进行训练、标记, 可进行推理来完成对图像的识别. AHL-EORS 主要目标用于嵌入式人工智能入门教学, 试图把复杂问题简单化, 利用最小的资源、最清晰的流程体现人工智能中“标记、训练、推理”的基本知识要素. 同时, 提供完整源码、编译及调试环境, 期望达到“学习汉语拼音从啊 (a)、喔 (o)、鹅 (e) 开始, 学习英语从 A、B、C 开始, 学习嵌入式人工智能从物体认知系统开始”之目标.

学员可以将测试工程下载到 MCU 芯片中, 进行 0-9 十个数字识别, 测试步骤如下:

步骤 1. 通电. 电压为 5V, 可选择计算机、充电宝等.

步骤 2. 测试. 上电后, 正常情况下, 如图 9 所示, LCD 彩色屏幕会显示出图像, 可识别 0-9 数字.

5 结论与展望

近年来, 由于嵌入式技术的飞速发展和广泛应用, 我国的许多高等院校和职业技术培训机构也开设了嵌入式系统实验教学的课程, 使得实验系统的设计开发成为嵌入式领域教学急需解决的问题. 本文首先介绍所实现的通用嵌入式计算机 GEC, 主要解决 GEC 硬件共性结构及 BIOS 驻留机制、基于串口的下载与调试方法、GEC 开发生态系统等问题, 并提出 AHL-

CH32V307 嵌入式实验教学解决方案. 基于 RISC-V 架构的高性价比教学实验系统通过合理应用资源搭建性价比高的系统平台, 在这一平台上编写大量教学性实验. 这些实验针对性强, 层次清晰, 易于学习并进行二次开发.

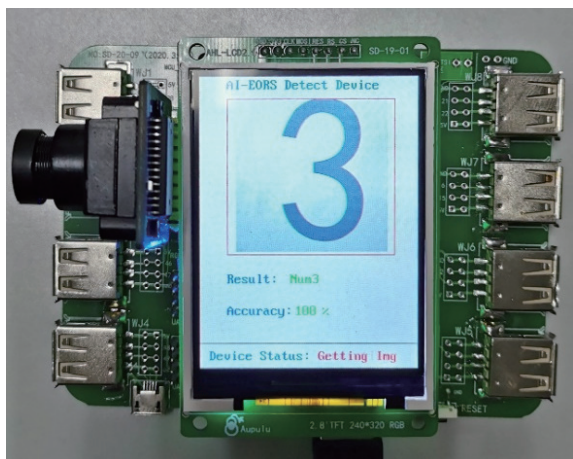


图9 LCD 彩色屏幕显示数字3图像

实验教学是本文最重要的设计部分, 配套开发与硬件模块结合紧密的两个层次实验, 第1个层次是外接传感器及执行部件编程方法的教学实验. 第2个层次嵌入式人工智能简明实例. 接下来的研究工作是基于套件开发更多贴近现实生活中的物联网应用. 如此既能推广 RISC-V 处理器的技术, 又能增加学员的学习兴趣. 甚至适用于 GEC SoC 芯片的加密方法来提高数据传输的安全性, 也是有价值的研究方向.

参考文献

- 1 王宜怀, 李跃华, 徐文彬, 等. 嵌入式技术基础与实践——基于 STM32L431 微控制器. 第6版, 北京: 清华大学出版社, 2021.
- 2 Waterman A, Lee Y, Avizienis R, *et al.* The RISC-V instruction set manual, Volume 2: Privileged architecture Version 1.7. Berkeley: University of California at Berkeley, 2015.
- 3 Patterson D, Waterman A. RISC-V 手册: 一本开源指令集的指南. 勾凌睿, 黄成, 刘志刚, 译. 2021. <http://riscvbook.com/chinese/RISC-V-Reader-Chinese-v2p1.pdf>. [2022-01-31].
- 4 陈彦漆, 钱璐遥, 陈麒安, 等. MicroPython 用于 RISC-V 微控制器的研究与设计. 中国集成电路, 2020, 29(3-4): 44-51, 56.
- 5 陈宏铭. SiFive 经典: RISC-V FE310 微控制器原理与实践. 北京: 电子工业出版社, 2020.
- 6 沁恒微电子股份有限公司. CH32FV2x_V3x 系列应用手册. www.wch.cn/downloads/CH32FV2x_V3xRM_PDF.html. [2022-01-31].
- 7 陈宏铭, 李水竹, 陈宏维, 等. 适合低功耗高精度红外测温 SoC 芯片与方案实现. 中国集成电路, 2012, 21(11): 50-58.
- 8 陈彦漆, 陈麒安, 赵振华, 等. 基于 RISC-V 微控制器与 FreeRTOS 红外耳温量测系统. 中国集成电路, 2020, 29(5-6): 55-62.
- 9 王宜怀. 嵌入式实时操作系统: 基于 RT-Thread 的 EAI&IoT 系统开发. 北京: 机械工业出版社, 2021.
- 10 陈宏铭, 胡进, 徐滔, 等. 利用 RISC-V 构建信任方案解决安全性落差. 中国集成电路, 2019, 28(11): 51-63, 75.

(校对责编: 孙君艳)