

分 类 号 _____

学号 M201873228

学校代码 10487

密级 公开

华中科技大学

硕士学位论文

智慧校园卡关键驱动实现技术

学位申请人：李丹迪

学 科 专 业：计算机技术

指 导 教 师：胡贯荣 副教授

答 辩 日 期：2020.6.6

**A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree for the Master of Engineering**

**Key driving implementation technology of
smart campus card**

Candidate : Li Dandi

Major : Computer Technology

Supervisor : Assoc.Prof Hu Guanrong

**Huazhong University of Science & Technology
Wuhan 430074, P.R.China
June, 2020**

摘要

随着传感、通信、计算机等技术的发展，嵌入式相关技术应用领域广阔，依赖于嵌入式的物联网技术得到了发展，广泛应用物联网技术的智慧校园开始备受关注。智慧校园目的是使用信息技术提高教学效果。在智慧校园中，智慧校园卡承担着信息收发和课堂交互的作用。研制出智慧校园卡，可以辅助教师教学，提高课堂效率。

智慧校园卡软件系统分为应用层，支撑层，驱动层，重点研究和实现智慧校园卡的关键驱动。软件开发平台使用 Keil 搭建工程，使用 Jlink 将代码烧录到开发板中，进行调试开发。在了解硬件原理的基础上，选择对应的通信协议，构建数据传输通道，进而完成智慧校园卡关键驱动的实现。

首先对智慧校园卡的硬件组成和软件系统进行介绍，并且设计了驱动开发框架。其次是显示屏的驱动实现，显示屏支持 SPI 总线，在驱动初始化时完成 SPI 初始化和芯片的显示设置，并且实现了用于传输命令和数据的接口和显示数据传输接口。此外还设计了功耗管理策略。通过这些设定，设定了显示缓存区域，规定了黑白显示功能，使得显示屏能够显示教学内容。最后是加速度传感器模块的驱动实现，通过编写的读写接口，对寄存器进行读写，设定芯片地址和工作模式，完成芯片的初始化，并提供获取三轴加速度值的接口。为了避免误差，还设计了一种极端温度下的补偿策略。最终芯片使能，向上层提供准确加速度值。

关键词：智慧校园卡，通信协议，显示驱动，加速度传感器驱动

Abstract

With the development of sensor, communication, computer technology. Embedded technology has wide application fields. Relying on Internet of things technology has been developed. Smart campus using Internet of things technology has attracted more and more attention. The purpose of smart campus is to improve teaching effect by using information technology. In the smart campus, the smart campus card can realize information sending and receiving and classroom interaction. Manufacturing smart campus card can assist teachers in teaching and improve classroom efficiency.

Smart campus card software system is divided into application layer, support layer and drive layer. Focus on research and implementation of key drivers. The software development platform uses Keil to build the project, JLINK is used to download the code into the development board for debugging. Understanding the principle of hardware, we can choose the communication protocol, build the data transmission channel, and then complete key driver of smart campus card.

Firstly, the hardware composition and software system of smart campus card is introduced, and the driving development framework is designed. Secondly, the driver of the display screen is completed. The display supports SPI bus. SPI initialization and chip display setting are completed during driver initialization. The interface for transmitting command and data and transmission interface of the display data are implemented. In addition, the power management strategy is also designed. Through these settings, the display buffer area is setted, and the black and white display function is specified, so that the display can display the teaching content. Finally, the driver of the acceleration sensor module is completed. Through compiling read-write interface, reading and writing the value of register, setting chip address and working mode, the initialization of the chip can be completed. And the interface to obtain the triaxial acceleration value is provided. In order to avoid the error, a compensation strategy under extreme temperature is designed.

Finally, the chip provides accurate acceleration value for the upper layer.

Keywords: Smart campus card, communication protocol, display drive
acceleration sensor drive

目录

摘要.....	I
ABSTRACT.....	II
1 绪论	
1.1 研究背景.....	(1)
1.2 国内外研究概况.....	(2)
1.3 课题主要研究工作.....	(5)
2 系统整体架构	
2.1 智慧校园系统简介.....	(6)
2.2 硬件设计.....	(8)
2.3 软件总体设计.....	(11)
2.4 驱动框架的设计.....	(12)
2.5 开发环境.....	(14)
2.6 本章小结.....	(15)
3 墨水屏显示驱动实现	
3.1 墨水屏操作流程.....	(16)
3.2 墨水屏传输方案.....	(17)
3.3 墨水屏驱动初始化.....	(21)
3.4 墨水屏传输实现.....	(28)
3.5 墨水屏功耗管理策略.....	(30)

3.6 本章小结.....	(32)
---------------	------

4 加速度传感器驱动实现

4.1 加速度传感器工作流程.....	(33)
---------------------	------

4.2 加速度传感器传输方案.....	(35)
---------------------	------

4.3 加速度传感器的传输接口.....	(37)
----------------------	------

4.4 加速度传感器驱动初始化.....	(41)
----------------------	------

4.5 传感器加速度值获取接口的实现.....	(45)
-------------------------	------

4.6 本章小结.....	(48)
---------------	------

5 测试

5.1 测试环境.....	(49)
---------------	------

5.2 显示屏模块的测试.....	(49)
-------------------	------

5.3 加速度传感器模块的测试.....	(50)
----------------------	------

5.4 本章小结.....	(51)
---------------	------

6 总结与展望

6.1 总结.....	(52)
-------------	------

6.2 展望.....	(53)
-------------	------

致谢.....	(54)
---------	------

参考文献.....	(55)
-----------	------

1 绪论

1.1 研究背景

随着传感器、通信、计算机等技术的发展，嵌入式相关技术应用领域广阔，依赖于嵌入式的物联网技术也得到了飞速发展。物联网技术广泛地应用于交通、医疗、工业、农业、教育等领域，引申出来了智慧交通、智慧医疗、智慧校园等多个概念^[1]。智慧校园首先由浙江大学提出，描述了无处不在的网络学习环境、融合创新的网络科研、透明高效的校务治理、丰富多彩的校园文化、方便周到的校园生活^[2-4]。

2018 年国家公布了智慧校园建设总体框架，在这个框架中提出了智慧校园所需的功能，包括校园管理，校园服务，教学资源，校园环境。框架要求，智慧校园需要有云计算平台、PC 管理平台、智能设备以及数据中转基站。基站将智能设备产生的信息转发至云平台，智能设备之间互相传输信息辅助课堂教学，共同组建交互共享的智慧校园。涉及到的相关智能设备，就有校园一卡通。框架提出，一卡通应该具备校园业务数据的生成、采集、传输、汇总、分析等基本需求，并且要配合校园教学环境，辅助课堂教学^[5]。

根据框架的要求，需要制作出一款独具学习管理功能的智慧校园卡。智慧校园卡主要具备教学互动和智能管理两大功能。教学互动功能主要是辅助课堂中学生与教师的互动，包括课程通知、在线答题、素质评价、短信通知。智能管理功能主要实现对学生的信息数据的生成、采集、管理、推送。这些数据包括学生课堂表现评价数据、短信数据、出勤数据、位置数据。将记录的数据上传到云平台，教师和家长之间共享，便于对学生的日常活动和教学信息进行管理。

本课题将围绕着智慧校园卡而展开，在智慧校园卡的设计实现中，针对驱动层技术进行研究。在嵌入式相关产品中会经常用到相关硬件的驱动设计，根据不同硬件，需要开发出相应的软件驱动程序，供产品的上层应用实现某种功能。若从中抽取出一套可以持续利用的驱动框架，根据此框架来设计完成智慧校园卡的驱动，开

发出高效的，适配性高的智慧校园卡关键驱动代码，也便于智慧校园卡的产品更新迭代，具有广阔的研究前景。

1.2 国内外研究概况

1.2.1 校园卡发展概述

在大学校园中，校园一卡通比较常用。一卡通能为学生的生活提供帮助，可以实现借书还书，食堂消费，图书馆和宿舍的门禁刷卡^[6]。它还可以采集数据，方便学校对学生的校园生活进行管理。这些依赖于射频识别技术和 J2EE 技术。通过射频识别技术实现管理系统与用户卡之间的非接触双向通信，从而达到用户识别、数据记录的目的^[7]。J2EE 技术主要应用于软件方面，学生信息出现变化时，需要一个平台去记录、查询、维护这些数据。J2EE 技术其本质实际上是 Java 技术，它可以提供相应的容器和组件，方便开发者使用该技术进行应用设计和开发^[8,9]。一卡通软件系统的设计就是基于 J2EE 技术，借此建立起可扩展的为多用户服务的平台，方便对全体学生的信息进行管理^[10]。这种类型的校园卡被大中小学校广泛利用，但是该类型学生卡只是一种辅助类型的卡片，不能显示学生的数据信息，有一定的局限性。

随着技术的发展，人们开始考虑使用虚拟卡片来替代校园卡。将 NFC 技术与 APP 相结合，使用 APP 上的虚拟卡片充当校园卡。NFC 即为近场通信技术，属于近距离无线通信，通信时间短，速度快，能实现芯片识别和双向通信的功能^[11]。NFC 将非接触卡、非接触读卡器以及点对点功能集成到一块小芯片上，自身的成本低廉，所以广泛的嵌入到智能终端产品上使用^[12,13]。每一个带有 NFC 的手机，均可以读取到 NFC 标签数据，该标签数据可以通过 HTTP 协议进行数据交互。手机上专用的 APP 收集感知的内容信息和用户信息，通过加密算法发送到校园后台的数据云服务中心^[14]。之后要对收到的信息进行判断，读取信息进行信息验证，解密后截取出用户信息和内容信息，判断是否是合法用户，如果是，则根据内容信息进行下一步操作。最后这些操作的结果会反馈到用户的 APP 上，供用户查看。这种类型的校园卡解决了实体卡带来的不便，但是需要全体学生使用带有 NFC 功能的智能手机并且下

载对应的 APP，所以这种方式有一定的限制。

随着手环的兴起，设计一款手环一卡通就很有应用价值。基于 STC12C5A60S2 芯片和蓝牙技术设计了手环式一卡通，它不仅可以实现普通的校园一卡通功能，还可以实现显示功能，使得数据可视化^[15]。手环采用 STC12C5A60S2 芯片作为该系统的控制模块，该芯片是一款是高速，低功耗，超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051，内部集成专用复位电路、看门狗、时钟源、定时器，功能全面，价格可观^[16-18]。在数据交换方面，使用蓝牙模块实现系统与手机终端和电脑终端的数据交换，蓝牙传输是比较成熟的一种方式，随着蓝牙 5 标准的出现，蓝牙传输技术也具有更快的传输速度，更灵活的广播数据包^[19]。同时该手环还具有显示功能，手环上负载有 OLED 显示屏。OLED 显示屏可以显示字符、文字、图案等多种元素。显示屏将蓝牙收到的数据包的数据读取出来，显示在手环上，方便学生查看信息。该设计还具备普通手环的功能，增加了加速度感应模块用来统计步数。手环在使用上更加便捷，还具备显示功能，这种设计值得学习。

通过对国内外的校园卡的产品研究，带有显示屏的实体卡更加适合学生使用。虚拟卡片需要使用智能手机，会分散学生注意力，而便捷性更高的手环，显示屏小不方便使用。智慧校园卡在此基础上，需要选择合适的硬件产品，通信手段，在软件上完成相应的驱动实现和支撑技术。因此需要相关通信技术，驱动技术和支撑技术的研究探索，组合成智慧学生卡的设计方案。

1.2.2 关键技术

由上文的研究现状来看，智慧校园系统是复杂的系统，校园卡的功能也丰富多样。为了实现这些功能，需要多种技术相互辅助。

(1) 通信技术

校园卡和校园系统的通信是设计中重要的一环。现有的无线通信技术有低功耗广域网、蜂窝数据、蓝牙和 BLE、RFID、WIFI、Zigbee 和其他网状协议。就目前来看，智慧校园类产品多采用蓝牙和 BLE，RFID 这两类技术。

RFID 技术,就是射频识别技术。它利用无线电波进行这种不接触的信息交互,完成对卡片身份的识别,以此连接数据库,进行数据交换,达到非接触双向通信的效果^[20]。一个 RFID 系统包括电子标签、读写器、应用系统三个部分^[21-23]。电子标签是存储数据的地方,它能接受到信号,实现对数据的读写操作。读写器包含天线,主要是用于接受信号,发送数据。应用系统,可以发送工作指令,对标签进行识别,并且计算、存储采集到的所有数据。这类通信技术数据可以及时更新,读写速度快,使用寿命长,体积小,因此多被嵌入到卡产品中。

蓝牙技术发展至今,已经非常成熟,尤其是低功耗蓝牙技术(BLE)的出现,解决了传统蓝牙的一些劣势。低功耗蓝牙可以超长待机,快速连接,发送接收功耗也比较低。相较于传统蓝牙,低功耗蓝牙建立连接的速度只需要 3 毫秒,可以快速连接,并且收发速度很快,在连接结束后还会立即暂停发射无线,一定程度上减少了收发功耗^[24-26]。在广播包的数据长度方面,低功耗蓝牙数据包的长度可以在 8 到 31 字节灵活变动,这样在传输时也减少部分功耗。因为低功耗,数据包灵活,连接快速等特点,这类蓝牙被广泛采用^[27,28]。

(2) 数据管理平台

智慧校园系统需要数据平台来存储数据,并且进行计算的。故搭建一个辅助系统运作的管理平台,也是重要的一环。

在传统的系统中,采用 J2EE 搭建软件管理平台。它提供了基于组件的方式来设计、开发、组装和部署企业应用,可以高效地对 Web 应用开发提供支持^[29]。在此基础上,开发一个数据管理平台,对校园卡传递过来的数据进行更改、存储、计算、反馈。这种平台技术上非常成熟,但是为了对这些数据进行存储,本地资源消耗很多,有一定的成本。

而现在,针对数据的计算和存储平台,有了进一步的改进。云计算平台,它通过网络将数据处理交给其他服务器来做,计算完成后再返回给用户。云计算就是一种提供资源的网络,把计算资源集中起来,参与数据的处理,所以云计算技术可以在很短的时间内完成对数以万计的数据的处理^[30,31]。像这种可以提供计算和存储功

能的云平台，减少了企业本地存储资源的消耗，对于用户来说，用户不需要了解其中细节，直接可以获取想要的数据库。

1.3 课题主要研究工作

课题的主要方向是智慧校园卡的设计，为中小学生的日常生活和教育教学做辅助。需要研究的工作有：

（1）智慧校园卡总体设计

智慧校园卡整体可以分为四层，包括硬件层，驱动层，支撑层和应用层。在硬件上选择了显示屏，加速度感应这几个功能模块芯片，用于实现产品需求，比如基本的显示和动作判断。着重关注智慧校园卡驱动框架设计，归纳抽象出一个适用于功能芯片的驱动设计框架，方便未来的驱动移植和再设计。之后可以借助此框架，对这几个模块驱动进行设计。

（2）显示驱动设计

校园卡需带有显示屏，用于学生在使用时显示课表、短消息、题目、时间等信息，这就涉及到字符、汉字、图片等显示内容。在驱动设计方面，要完成显示屏与主控芯片的时钟控制和数据传输，需要对显示屏进行驱动初始化。同时还需提供缓存数据接口，方便进行上层业务数据的显示。

（3）加速度传感器驱动设计

加速度传感器的设计，主要是为上层业务服务。上层业务根据加速度值的变化，可以获取晃动次数和卡片的动作信息。在驱动设计中，需要将加速度传感器使能，完成驱动初始化，并且提供加速度值的获取接口，更加准确的获取用户对卡片的操作，方便上层业务设计。

2 系统整体架构

本章首先对整个智慧校园系统进行介绍。其次阐述智慧校园卡主要的系统框架和功能，系统框架主要介绍该设计所用的硬件选择和软件模块结构，功能主要是描述智慧校园卡所具备的使用功能。然后抽象出一个驱动设计框架用于后期开发。最后对开发环境进行了介绍。

2.1 智慧校园系统简介

智慧校园系统主打互动课堂，为老师开展互动教学提供更优的方式方法，通过课堂互动的方式帮助老师随时了解全体学生对知识的掌握情况。学生在学校的相关数据都可以记录保存，被记录的数据可以由管理人员、家长、教师等抽取检查，了解学生们的学习情况，这种方式也能辅助解决当前家长、学生、老师之间的合作沟通问题。

智慧校园系统总体可以由这几个部分组成：智慧校园卡，数据基站，云计算平台，教学终端。智慧校园系统框架如图 2.1 所示，智慧校园系统的相应业务也有所标识。智慧校园卡是信息采集和接收的终端之一，它接收从基站发来的命令，完成课堂互动等多种功能，并将获取的学生信息发送给基站。教学终端有两种，教师端可以发送教学命令，家长端可以获取学生在校表现。物联网数据基站是信息中转站，它接收教学终端发送的命令转发给校园卡，校园卡把信息反馈给基站。

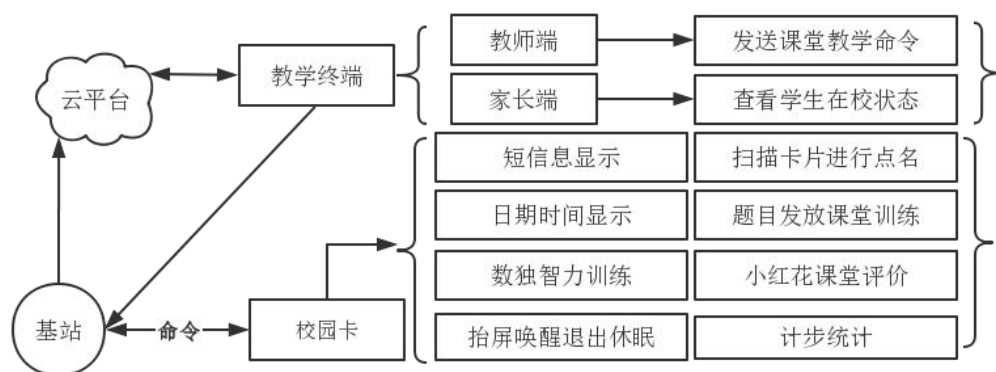


图 2.1 智慧校园系统框架

在学生进课堂的场景中，学生进入课堂后，物联网数据基站会扫描学生出勤情况，与智慧校园卡进行交互，自动统计出未出勤学生的名单并将数据上报，老师也可结合实际对考勤数据做手动调整后上报给云平台，无须增加额外的刷卡设备，也无须学生主动刷卡，一切均自动完成。在上课的场景中，老师可通过教师使用的终端平台或手机，向学生发起互动，比如教师可以向学生发起抢答，则学生使用智慧校园卡按键抢答，或者老师也可针对选择题、判断题发起全员评测，学生在智慧校园卡上通过选择与判断按键进行回答。这些答题数据，包括学生学号，学生的答案，会通过校园卡传输到数据基站。物联网数据基站在接收到校园卡的反馈数据后，通过安装在手机、Pad、以及校园内安置的显示屏上的互动课堂软件展示答题结果，老师可实时评估学生对知识的掌握情况并及时做出教学调整。下课后，学生的课堂考勤及互动数据都将被上传、保存到云平台。通过云平台及互动课堂云数据服务，增强了教学展示效果，并且可以对数据进行统计和分析，用于跟踪记录每一个学生的学习情况。相关数据，老师、学生、家长们也可通过手机、Pad 的软件进行查询，实现学校、老师及家长的数据共享。

智慧校园卡除了具有上述的答题、接收命令、传输数据的功能之外，还有其他的一些辅助功能。比如卡片能够完成基本的通知提醒功能，包括显示年月日星期等时间信息、显示教师和家长发送的短信，显示课表等。在触摸功能上，卡片除了能够

点击选项和判断正误，还能进行上下左右滑动的识别，在后期，可以加载在手写数字识别算法，使得小学生能练习基本的口算速算技巧。此外还辅助增加加速度传感器模块，实现对学生每日的步数统计，并且通过加速度传感器感应晃动次数，通过这个晃动可以在课堂快速响应命令。再加上定位，进出门检测，电量提醒等功能，以此来丰富校园卡的使用体验。

通过基站中转，教学终端向智慧校园卡发送命令，智慧校园卡将学生身份、出勤状态、素质评价、位置信息等数据反馈到云计算平台，云平台将这些数据共享给终端用户。通过这种方式为师生提供课堂签到、普通答题、抢答、教学统计分析等功能，真正实现互动课堂。

2.2 硬件设计

智慧校园卡的硬件主板需要具有传输，控制功能。为了兼顾业务功能，还需要显示触摸等其他功能模块，故硬件主板需要多种芯片配合完成。硬件结构如图 2.2 所示。

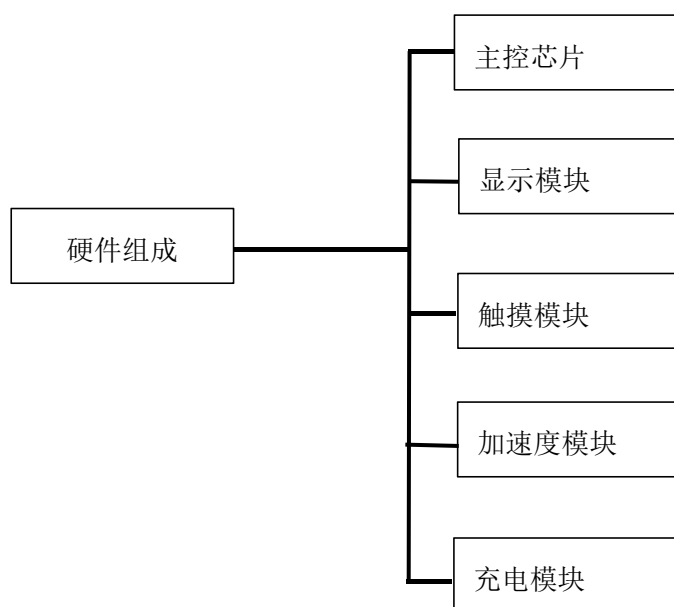


图 2.2 硬件结构图

(1) 主控芯片

主控芯片使用的是 STM32WB55CGU6。STM32WB 平台是在市场上超低功耗

STM32L4 系列的基础上发展而来。该系列芯片带有双核 MCU，富有多种通信特性，支持 USB2.0，适配性很强，可以用于多种嵌入式产品的研发^[32]。采用该芯片的原因有以下几点：①多达 72 个 GPIO，给开发者在设计开发中提供了便利。②具有多样的低功耗模式，适合校园卡的设计需求。③STM32WB 能够选择的传输协议也多种多样，内嵌通用 MAC 允许使用 IEEE 802.15.4 专有协议栈或其它专有协议，无线技术可支持 BLE/OpenThread 协议，在校园卡产品设计时，考虑到功耗问题，需要使用低功耗蓝牙进行传输，STM32WB 提供了完好的 BLE 支撑环境。④安全功能，这款 MCU 支持多种加密方式，比如 256 位 AES，校园卡在信息传递的时候，需要对信息进行加密，所以需要一款能够支持简单加密功能的主控芯片。该芯片针对无线相关的软件处理、资源使用的灵活性、电源管理都进行了优化，在开发中具有很高的适配性和灵活性，并且提供了多种资源供开发者设计使用，满足校园卡需要。

（2）显示模块

显示模块的硬件选择需要考虑到校园卡使用时长，所以在硬件选择的时候需要把功耗考虑进去，同时也要满足基本的显示功能，要做到小屏幕也清晰可视。选择 HINK-E0154A45 作为显示屏元件。它是一款墨水屏，内含许多微囊，本质是一种墨水状的悬浮物^[33-35]。每个微囊里包含多个黑色粒子和多个白色粒子，两者各带相反电荷。当需要在屏幕上显示黑色内容时，通过电压驱动黑色粒子向上移动到屏幕顶部，当显示白色内容时，通过电压驱动白色粒子向上移动到屏幕顶部^[36,37]。墨水屏具有以下几个优点：①文字显示切合实际。墨水屏的微囊，使得显示出来的文字效果贴近自然纸张上的文字，做成校园卡的显示屏能够让学生有更好的感官，达到舒适护眼的效果。②功耗少。墨水屏在文字刷新以后，会长期停留在屏幕上，屏幕在显示的时候不耗电，一定程度上减少了功耗。③抗摔不易碎裂。智慧校园卡由学生使用，难免会有磕碰，所以需要有一个耐摔的屏幕，防止碎屏影响显示效果。综上所述，选择墨水屏作为显示器件非常适合校园卡的需求。

（3）触摸模块

触摸模块采用的是 FT3267 芯片。采用该芯片做为触摸模块的原因有以下几点：①高灵敏度。人们已经习惯了使用大屏的智能产品，在触摸习惯偏向大屏的使用习

惯，如果触摸区域小，比较容易断触，故需要一款高灵敏度的触摸设备，弥补这一缺陷，FT3267 恰好具有高灵敏度。②抗干扰能力强。该芯片能确保在电源和其他设备的各种干扰下实现稳定的触控效果。智慧校园卡元件众多，所以抗干扰能力是十分必要的。③低功耗。该芯片在使用的时候耗能少，满足校园卡的需求。因为其高灵敏度，低功耗，强抗干扰能力，所以选择 FT3267 作为校园卡的触摸元件。

（4）加速度传感器模块

加速度传感器选择的是 BMA253，这是一种普遍使用的加速度传感器，它可以测量 3 个相互垂直方向的加速度值，应用于运动检测，方向检测，振动控制，计步，撞击检测，倾斜测量等方面。BMA253 是一种电容式传感器，由可运动质量块与固定电极组成一个电容，当受加速度作用时，质量块与固定电极之间的间隙会发生变化，电容也会发生变化^[38]。这类传感器的灵敏度高，抗干扰能力强，并且受温度影响小^[39,40]。此外 BMA253 封装结构微小，接口也被设计得符合多种硬件设计的要求，故选择该产品成为校园卡加速度传感器的器件。

（5）充电模块

充电模块，使用的是充电芯片 RT9527。该芯片具有以下特点：①电池类型。RT9527 可以从交流适配器或 USB 获取电源输入，为锂离子电池充电，校园卡采用的就是锂电池。②具有电源保护功能。在充电时，芯片可以承受的最大电压为 28V，比一般的芯片电压范围要广，并且还具有过压保护，欠压保护功能，尽最大可能的保护电池，延长电池使用时间。③充电模式多样。该芯片可以实现三种充电模式，包括预充模式，快充模式和恒压模式，这样的模式组合，一方面保证了充电速度，另一方面，在充电完成后可以保持恒压模式，确保始终满电。综上所述，该芯片可以在过压和欠压时对电池进行保护，在充电时也能保证合适的充电速率，延长了电池的使用时间，故选择该芯片作为充电元件，尽可能的延长校园卡的使用寿命，符合校园卡需求。

2.3 软件总体设计

在设计中，为了给校园卡提供更多的功能支撑，配合校园教学活动等多种业务需求，需要驱动程序做好相应的数据采集处理工作，上层业务根据这些数据进行逻辑判断。智慧校园卡的设计需要至少要具备以下功能：通过蓝牙完成收发通信；课堂教学交互中显示题目进入应答并发送回答；短消息（小红花和评语）的接收显示存储功能。为此软件设计可以分为驱动层，支撑层，应用层。下层为上层服务，共同为业务支撑，软件总体设计如图 2.3 所示。

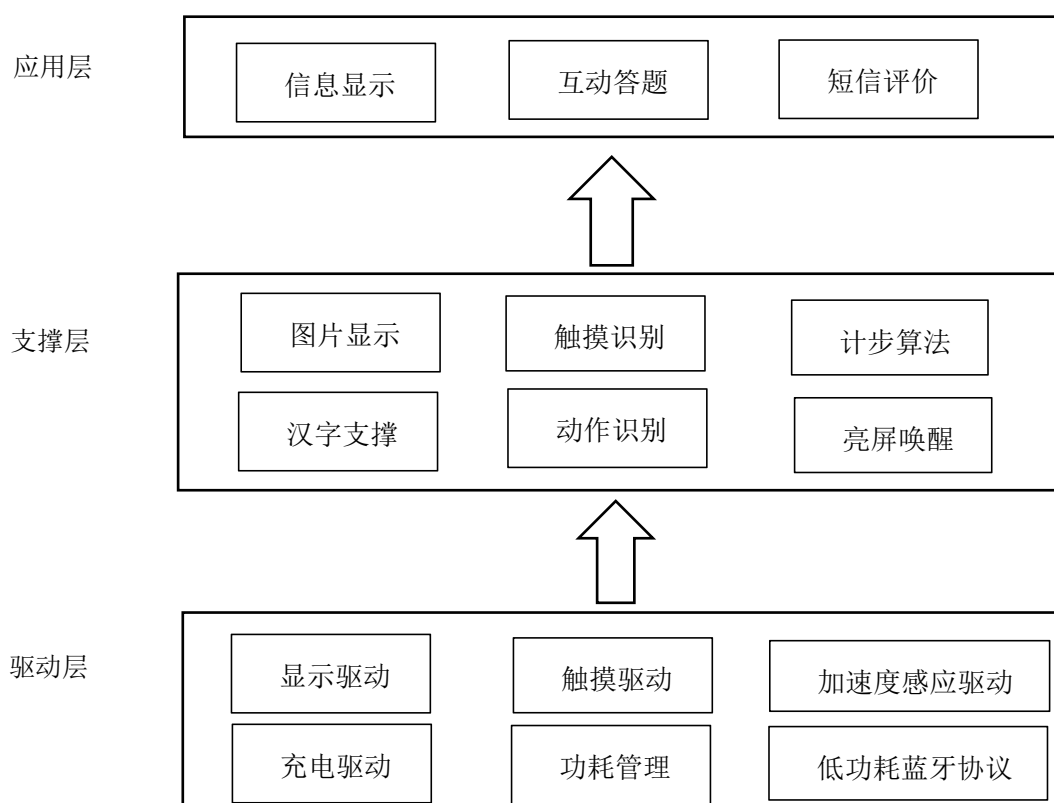


图 2.3 软件总体设计

驱动层需要给底层硬件编写对应的驱动程序，让硬件模块使能，完成主控芯片与各模块之间的数据传输，并且提供接口来输出采集到的数据，方便上层通过这些数据来进行业务判别。故需要设计对应的显示驱动，触摸驱动，加速度传感器驱动，充电驱动。主控模块要完成蓝牙传输协议的设定，智慧校园卡在课堂交互中主要依

靠蓝牙接收命令，比如教师答题命令，教师评价信息等，采用低功耗蓝牙协议完成蓝牙通信，满足课堂互动需求。在各个模块设计的同时，还要考虑功耗问题，延长校园卡使用待机时间。

支撑层提供智慧校园卡的技术支撑。校园卡需要显示图片，比如每一级目录需要有图标显示，这需要图片显示功能。在课堂交互中，有教师评价操作，可以对学生发送评语，这就需要汉字字库和汉字显示，汉字显示同图片显示类似，但是汉字字库需要根据存储空间的大小进行裁剪。此外需要触摸功能，动作识别功能。触摸功能需要识别使用者点击的位置，触发确认返回键；动作识别需要根据触摸轨迹，判断出上下左右的滑动，实现翻页功能。关于计步功能和亮屏唤醒功能，是通过加速度传感器模块实现的。计步功能需要结合相应的计步算法，故支撑层需要实现合适的计步算法，供应用层进行步数统计。亮屏唤醒功能根据感应加速度值的变换，判断卡片处于抬起倾斜的角度，触发亮屏操作。

在应用层，智慧校园卡专注辅助教学任务，主要应用有：①信息显示：比如日期时间，课表的显示，学生可以从校园卡上看时间和课表，确定上课时间和内容。②课堂提问应答：在课堂上通过蓝牙接收题目，触摸选择选项，并发送答案，教师还可以发起抢答，增加课堂互动。③短信评价：依赖于汉字显示和蓝牙收发功能，校园卡可以接收来自教师的评语，或者课业通知，以短信的形式呈现给学生。④附加应用：校园卡还增加了进校考勤，校园定位等功能，方便学生校园生活。为了辅助学生学习，增加了计数器，数独模块，口算练习等功能，学生在课余时间还可以使用学生卡进行简单是数学练习和思维开发。

2.4 驱动框架的设计

智慧校园卡已经更迭了几代产品，在产品功能和使用上越来越完善。初代产品采用的是 CC2640 作为主控芯片，并且不具备显示功能，只能与教师终端发生答题互动。之后的版本其主控芯片为 nRF52832，增加了显示模块。本次设计由 STM32WB 作为主控芯片，其他模块也与之前产品有所不同。通过智慧校园卡的环境，搭建一

个嵌入式的驱动框架。

驱动层开发一般指利用单片机的硬件资源完成控制功能，对其软硬件开发。常见的控制手段，比如时钟控制、定时器、AD 转换、GPIO 和各种通讯协议，包括 RS232、RS485、UART、SPI、IIC 等。多数情况下还需要外设驱动，比如电机控制、电磁阀、继电器控制、键盘扫描、LED 和 LCD 显示等。将这些驱动完成，给外界提供一个获取数值和进行控制的接口，上层就可以直接调用该接口，而不需要关心硬件底层的实现方式及实现过程。

在一般的设计流程中，对硬件产品的进行软件设计，也要尽可能降低耦合，对软件进行分层，方便后期的维护和移植。对于驱动设计，一般可以分为协议层和接口支撑层。协议层用于完成对芯片的数据传输，一般不同的硬件会支持一种或者多种通信协议，协议层主要完成对所选择的通信协议的初始化，模拟 IO，重要的是向上层提供封装好的读写接口，能够完成对不同长度的数据的传输读写，方便传输读取命令数据。接口支撑层主要是为硬件所需要完成的业务逻辑提供相应的功能接口，根据之前完成好的通信协议，完成对应引脚功能的使能，对芯片的初始化，并且将硬件采集到的数据进行输出或者显示。

协议层的设计中，首先是根据协议的时序图观察协议需要用到哪些信号线及时钟源，对通信协议使用到的信号线的相位和极性，时钟，以及通信过程中的各种起止确认状态标志等进行设置。了解了通信协议之后，就要编写代码进行 IO 模拟通信，进行协议的设定。比如 I2C，SPI，对其进行基本初始化，一般都会有固定的结构体，包括定义芯片地址，模式，传输的数据大小，帧格式等等。初始化之后，根据通信协议时序图，了解其数据传输机制，通过对时钟信号上升沿下降沿的控制进行数据传输，完成通信中读、写、开关控制等操作函数的设置^[41]。

接口支撑层就可以使用协议层定义好的操作函数，对硬件驱动进行设置。首先是对芯片进行初始化，根据数据手册，通过对寄存器写数据，或者写命令，达到对芯片的电源模式，时钟，中断，通信的波特率等参数设置，激活芯片，完成初始化。之后根据芯片的具体功能，接口支持层需要给应用层提供函数调用接口，比如加速度传感器需要提供读取加速度值的接口。这样应用层可以通过时间片轮询，前后台

执行，或者简单的顺序执行的方式调用这些接口去访问硬件，完成上层的业务逻辑。驱动框架如图 2.4 所示。

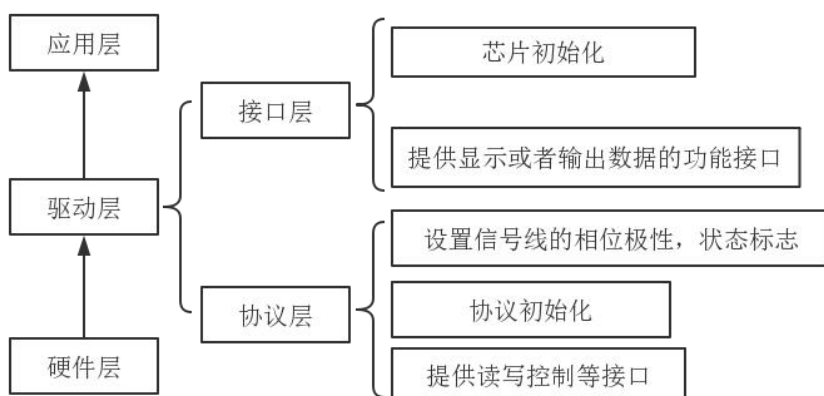


图 2.4 驱动框架图

根据这种分层思想，在设计驱动时，同样的硬件在移植时，只需要对头文件的引脚定义和寄存器映射做相应改变，源代码文件的各种接口不需要改动就直接可用。扩展到大多数硬件，在嵌入式开发的过程中，通信协议是固有的几种，故协议层可以不变。后期在对应的接口支撑层使用协议层定义好的操作函数，完成芯片驱动的初始化和数据输出接口。在开发中，根据分层解耦合的思想，尽量封装，提供接口，对后期的维护和移植都大有便利。

2.5 开发环境

（1）软件开发平台

使用 Keil 做为智慧校园卡的软件开发平台。Keil 的界面十分明了，集合了编译器、连接器、库管理、仿真调试器等功能，为开发者提供便利。通过此平台搭建项目文件，进行编译调试，对智慧学生卡进行仿真开发。

（2）软件开发工具包

STM32WB 系统对开发者提供了软件开发工具包，包含了 HAL 驱动和蓝牙协议栈。在工具包中提供了多个项目工程，包含了 STM32 常用的控制功能示例，方便开

发者熟悉芯片，解决开发中遇到的问题。

(3) 烧写与调试工具

程序的烧录使用 Jlink 仿真器。同时它还可以用于调试代码，通过在代码中增加打印语句，烧录进开发板开始打印调试信息，比如广播包的内容，或者状态标志。开发者可以随时调整打印语句，根据输出的信息，找到代码错误位置进行更改。

2.6 本章小结

本章节简要说明了智慧校园系统的组成和功能，大致构建出一个智慧校园系统，并且描述校园卡在其中发挥的功能。先是阐述了校园卡的软硬件构成，简要介绍芯片的信息以及优势，并且对校园卡的软件系统分层，对每一层的相关技术和应用进行介绍。其次对驱动层设计进行归纳，总结出驱动层框架。最后简要的介绍了开发环境。

3 墨水屏显示驱动实现

本章主要阐述了墨水屏显示驱动的设计与实现。首先设计了墨水屏与主控芯片的连接，并且进行了传输协议的选择。其次完成了墨水屏初始化，结合指令表使用 SPI 数据传输接口完成显示设置。最后编写了显示数据的传输接口，设计了功耗管理策略。

3.1 墨水屏操作流程

市面上有很多显示屏产品，它们各有优势，根据器件不同的特性，适应于不同的应用场景。校园卡为卡类产品，在选择上需要考虑清晰度，尺寸大小等，根据对比，选择电子墨水屏 HINK-E0154A45 作为显示屏。它是一种有源矩阵电泳显示器（AMEPD），带有接口和可参考系统设计。HINK-E0154A45 带有 1.5 英寸的显示区域，200×200 像素，配置有黑白全显示能力。芯片还提供了串行外设接口，内置温度传感器，可以获取外界温度加以使用。并且具有超宽视角，超低功耗，超低电流深度休眠模式，电源电压检测等功能。

HINK-E0154A45 作为显示元件，墨水屏的操作流程如图 3.1 所示。首先是对整个墨水屏进行供电处理，把电源引脚和地线等电源接口对应连接起来，完成基本的芯片连接。接下来是对墨水屏进行复位操作，重置墨水屏的设置。为了信息的传输，需要配置时钟信号，时钟信号由与主控芯片连接的引脚提供，可以同步传递信息。同时设置命令数据引脚，用于控制传输数据的类型，实现数据传输通道。接下来就是对墨水屏进行初始化操作，主要是完成屏幕分辨率，波特率，显示相关参数的设置，规划出一个显示缓冲区域。在显示流程中，当需要显示某些数据时，就刷新屏幕，清除原本缓存，再把接收到的文字数据或者图片数据，存储到这个缓冲区域里。最后调用显示接口，将信息显示到显示屏。

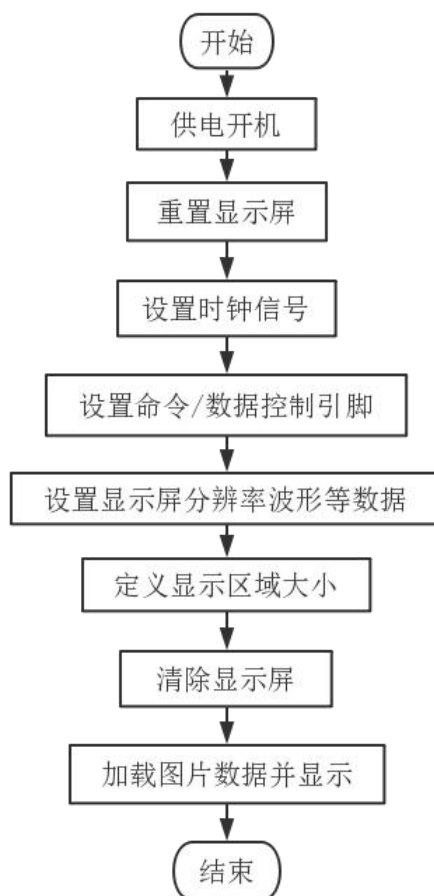


图 3.1 墨水屏操作流程

3.2 墨水屏传输方案

3.2.1 连接配置

HINK-E0154A45 作为显示元件，需要接收主控芯片传输过来的数据进行显示，则墨水屏的引脚需要和主控芯片构建连接，构建出信号连接通道，才能完成主设备与从设备之间的数据通信。主设备是 STM32WB55CGU6，从设备是 HINK-F0154A45。要实现 HINK-F0154A45 驱动，需要进行配置的引脚有 SCL、SDA、CS、DC 和 RES。下图简化地绘制芯片的连接引脚图，墨水屏的引脚连接如图 3.2 所示。

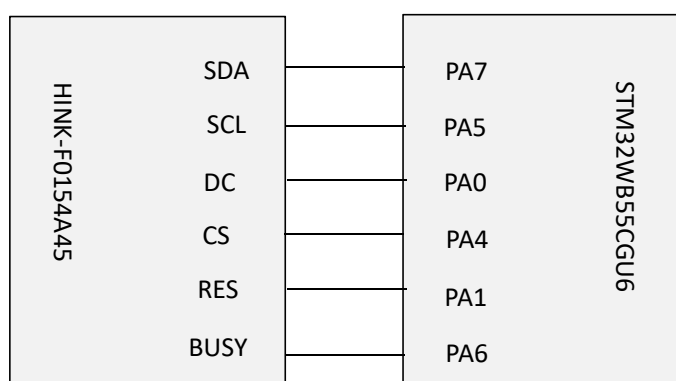


图 3.2 墨水屏引脚连接图

STM32WB55CGU6 中具有 16 个连接引脚 PA0~PA15, 可以提供给外设用于连接。将 HINK-F0154A45 所需要连接的 SDA、SCL、CS、DC、RES 等引脚, 与 STM32WB 单片机的这些提供给外设的引脚进行连接, 完成控制信号的传输。

SCL: 时钟引脚, 在传输过程中, 与引脚图中 STM32WB 的 PA5 连接, 由主设备 STM32WB 产生同步时钟信号, 控制数据交换。

CS: 片选引脚与 PA4 连接, 由 STM32WB 芯片 PA4 引脚提供片选信号, 通过片选信号选中从设备, 来达到对墨水屏的控制。当墨水屏被选中时, 片选信号需要低电平, 当使用完毕时, 要拉高电平。

SDA: 主输出从输入引脚, 使用 PA7 引脚与之连接, 主设备通过该信号线将数据送至墨水屏。

DC: 数据/命令控制引脚, 与之连接的是 PA0 引脚。对于一个字节数据, 当引脚被拉高时, 该字节将被解释为数据。当引脚拉低时, 该字节将被解释为命令。

RES: 复位信号引脚, 与之连接的是 PA1, STM32WB 通过该引脚传递低电平时, 墨水屏进行复位。

通过这样的连接, 架构好硬件之间的物理连接, 为之后的软件设计提供硬件连接基础。

3.2.2 传输方式

(1) 墨水屏传输协议的选择

在嵌入式设计当中，出现过很多通信方式，而芯片也与时俱进，满足多种通信方式的需求。在嵌入式系统的通信中，经常被使用的通信协议有两种，它们是 SPI (Serial Peripheral Interface)通信协议和 I2C (Inter-Integrated Circuit) 通信协议。这两种通信协议适合近距离低速芯片间的通信。

HINK-F0154A45 可以支持 3 线/4 线串行外围设备接口。在模块中，墨水屏的传输协议的选择可以通过 BS1 引脚进行。BS1 引脚控制功能如表 3-1 所示，当 BS1 引脚为低电平时，选择 4 线 SPI 传输协议，高电平时，为 3 线 SPI 传输协议。

表 3-1 BS1 引脚控制功能

BS1 引脚状态	协议选择
L	4 线串行外设接口
H	3 线串行外设接口(9 位串行接口)

SPI 总线是全双工同步串行外围接口，采用主从模式架构。SPI 总线结构如图 3.3 所示。SPI 支持一主多从的应用模式，多从机间通过片选信号区分，多主机也可以实现，只不过需要增加信号线。一般 SPI 总线有 2 根数据线（MOSI 和 MISO），2 根信号线（SCK 和 CS）。时钟由主机通过 SCK 信号线发出，在时钟脉冲下，数据在这两根数据线上按位传输，数据传输本质上是通过两个移位寄存器的数据交换实现的。

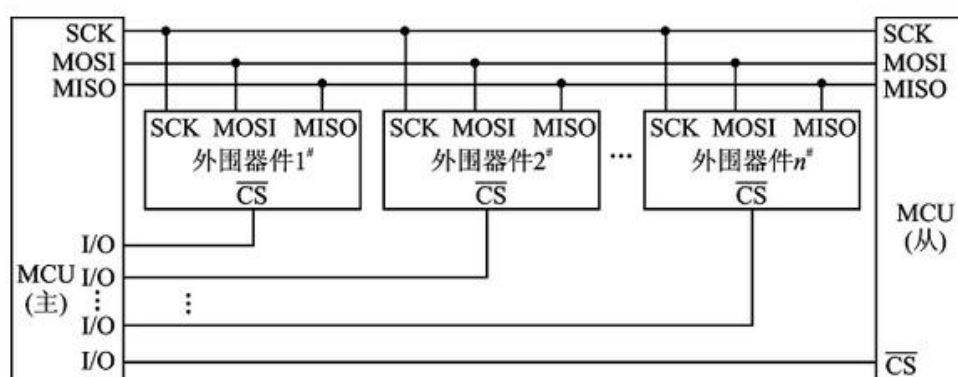


图 3.3 SPI 总线结构图

SPI 总线支持全双工通信，接口简单，四根信号线可以完成数据传输。虽然多个从设备的情况下，需要增加片选信号线，但是校园卡从设备比较少，不需要担心占

用太多口线，并且在智慧校园卡系统里只有一个主设备，也不需要额外的逻辑和线路。此外 SPI 具有数据传输效率高，速度快，传输稳定等优点，其传输速率在几 Mbps 到 10 Mbps 左右的范围。而 I2C 通信协议受到总线限制，最高传输速率为 3.4 Mbps，对比之下，I2C 的速度较慢。智慧校园卡显示屏模块使用时传输的数据包括文字和图片等，体量大，并且在一些课堂交互中使用的频率高，所以需要传输速率更高，传输稳定的总线方式。为了主从芯片的传输交互，为了更快速稳定的传输状态，选择 SPI 作为通信协议。

(2) 墨水屏在 SPI 协议下的传输过程

HINK-F0154A45 的串行外设接口(4 线串行接口)读写模式下可以传输数据字节和命令字节。4 线 SPI 包括时钟信号 SCL、串行数据信号 SDA、数据/命令信号 DC 和片选信号 CS。表 3-2 描述了写入命令或数据时 SPI 中的信号线状态，在引脚连接中，L 为低电平，H 为高电平，箭头代表时钟上升沿。传输过程中，DC 信号线高位为是写入数据，低位是写入命令。

表 3-2 写入功能下的 4 线 SPI 信号线状态

功能	SCL	SDA	DC	CS
写入命令	↑	命令位	L	L
写入数据	↑	数据位	H	L

在 4 线 SPI 模式下的写过程中，传输的数据为 8 位，可以用 D7, D6.....D0 代表这 8 个数据位。SCL 的每个上升沿，SDA 的数据会被移位到一个 8 位的移位寄存器中，顺序依次为 D7, D6.....D0。在这个字节的传输中，DC 电平应该保持不变，确保这一字节为完整的命令位或者数据位。在移位寄存器的数据字节会被写入到数据字节寄存器或者命令字节寄存器。

在 4 线 SPI 模式下的读取过程中，首先需要将 CS 驱动至低电平，单片机需要定义要读取的寄存器。接着 SDA 在 SCL 的每个上升沿将数据按 D7、D6 到 D0 的顺序移位到 8 位移位寄存器中，DC 保持低电平。SCL 变为低电平时，在寄存器的最后一位读取完毕，DC 需要驱动至高电平。SDA 在 SCL 的每个下降沿以 D7, D6 到 D0 的顺序移出一个 8 位数据。根据寄存器类型，可以读取超过 1 个字节。读取完所有

字节后，CS 需要驱动至高电平以结束选中。

3.3 墨水屏驱动初始化

墨水屏的初始化主要有以下几步：供电，引脚设置，协议初始化，显示设置。在完成芯片必需的功能引脚的连接后，选择 SPI 协议作为传输协议，需要对 SPI 协议进行初始化，完成相应的传输接口的实现，最后使用这个传输接口传输命令，完成显示相关设置。墨水屏初始化流程如图 3.4 所示。

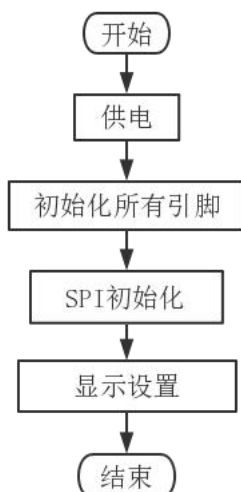


图 3.4 墨水屏初始化流程

3.3.1 SPI 初始化

对 SPI 进行初始化，是整个墨水屏初始化的第一步。SPI 初始化之后才能借此完成传输接口的实现，为后面的显示设置做准备。SPI 的初始化分为两步：SPI 功能函数的封装和 SPI 参数设定。SPI 的功能函数的封装，主要是完成 SPI 相关功能函数的整合封装实现，SPI 参数的设定可以完成工作模式，时钟等内容的设置，设置好的

参数模式，如果合法，则执行设定。SPI 初始化流程如图 3.5 所示。

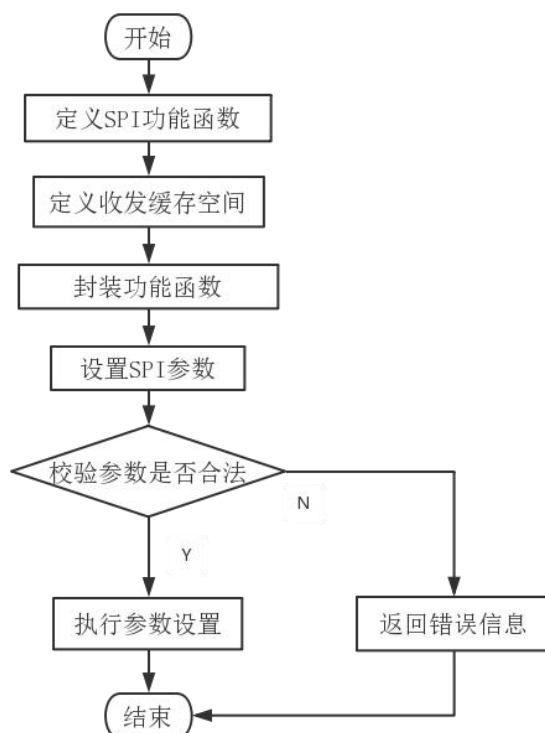


图 3.5 SPI 初始化流程

(1) SPI 功能函数的封装

定义 SPI_HandleTypeDef（功能函数结构体），这里面封装了 SPI 所需要的功能函数表。比如 TypeDef（类型定义），类型可选择 SPI1 或 SPI2，由于 SPI 功能不需要扩展太多，故选择 SPI1。此为还有传输过程所需的 TxBuffer（发送缓存），TxSize（发送缓存大小），RxBuffer（接收缓冲），RxSize（接收缓存大小）的设置。最后还需要实现收发缓存对应回调函数，DMA（直接存储访问）处理函数，状态标识函数，错误提示函数等几个功能函数。其中错误提示函数可以辅助调试开发，提示 SPI 错误信息。

(2) SPI 参数设定

SPI 参数的设置，需要 SPI_InitTypeDef（初始化参数结构体）。该结构体中存放着 SPI 相关参数，规定了 SPI 的工作模式。初始化结构体如下：

```
typedef struct
{
    uint32_t Mode;           //主从模式
    uint32_t Direction;      //传输方向
    uint32_t DataSize;
    uint32_t CLKPolarity;    //时钟控制
    uint32_t CLKPhase;
    uint32_t NSS;
    uint32_t BaudRatePrescaler; //波特率
    uint32_t FirstBit;
    uint32_t TIMode;
    uint32_t CRCCalculation;  //CRC 校验
    uint32_t CRCPolynomial;
    uint32_t CRCLength;
    uint32_t NSSPMode;
} SPI_InitTypeDef;
```

在初始化中，这些参数的设置是这样的：**Mode** 用于选择主从模式，值为 0 时为从模式，值为 4 时是主模式，在这里选择主模式。**Direction** 控制双向传输模式，一共有 2LINES 和 1LINES, 2LINES_RXONLY 三种选择，在这里选择 2LINES。**DataSize** 选择数据大小，定义为 8 位。**BaudRatePrescaler** 设置波特率分频器，可选选项有 2、4、8、16 分频等等，选择 2 分频即可。**FirstBit** 指定数据传输是从 MSB 位还是 LSB 位开始，在这里选择 MSB 在前。由于 SPI 不需要一些扩展功能，所以 TI 模式，NSSP 模式，CRC 校验等功能直接选择禁用。

3.3.2 SPI 传输接口的实现

SPI 传输接口涉及到底层引脚的控制，所以需要对引脚进行读写，控制引脚状态。在墨水屏的传输中，比较常用到写操作，用来传输命令和数据。

(1) 底层引脚读写

在初始化之后，需要对相关引脚进行设定，包括 DC、CS、SDA、SCL、RES。
4 线 SPI 读写过程中需要信号线的高低电平变换，故需要定义函数将信号线拉高或者拉低，以满足后续传输过程的需要。

STM32WB 系列提供的驱动函数库中，提供设置引脚状态值函数 HAL_GPIO_WritePin 和读取引脚状态值函数 HAL_GPIO_ReadPin，通过这两个函数对 STM32WB 芯片 16 个引脚的状态值进行读和写，进而实现对信号线电平的控制。

①写引脚状态函数

HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)。

该函数为对引脚的写函数，可以设置或清除选定的数据端口位。GPIO x，其中 x 可以是 A 到 F，以选择 STM32WB 系列的 GPIO 外围设备。GPIO_PIN 指定要写入的端口，端口可以是 PA0 到 PA15，为主控芯片可操作的 16 个引脚。PinState 指定要写入选定位的值。枚举两个状态值，分别 GPIO_PIN_SET 和 GPIO_PIN_RESET。GPIO_PIN_RESET 代表清除端口状态，用于置引脚为低电平，GPIO_PIN_SET 用于设置端口为高电平。

②读引脚状态函数

HAL_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)。该函数可以读取指定引脚的状态值。GPIO x 选择 GPIO 外围设备。GPIO_PIN 指定要读取的端口，可以是 PA0 到 PA15。函数最终返回指定端口状态值。

通过使用读写引脚接口，采用宏定义的方式，将这些读写函数重命名，与引脚名对接。比如 PA7 引脚连接的是显示屏的 SDA 引脚，所以把该引脚置高电平命名为 SDA_H。定义如下：

```
#define SDA_H HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7, GPIO_PIN_SET)
```

用同样的方式，使用 Write 函数对 CS、DC、SCL、SDA 等引脚分别编写它们的高电平和低电平的设置函数，同时使用 Read 函数实现对 BUSY 和 SDA 这两个引脚状态的读取，这些将用于 SPI 写操作的实现。

(2) SPI 写操作的实现

使用 SPI 通信协议在处理器和显示屏之间进行数据传输，贯穿于整个显示屏的工作流程中。智慧校园卡需要写命令函数和写数据函数。写命令函数传输命令控制字，不同的命令字对应不同的控制类型。写数据函数可以通过写入数据确定该控制类型的具体模式。写命令函数和写数据函数一般同时使用。

写操作主要是通过收发数据接口完成的。该接口为：

```
HAL_StatusTypeDef HAL_SPI_TransmitReceive_DMA(SPI_HandleTypeDef *hspi,
uint8_t *pTxData, uint8_t *pRxData, uint16_t Size)。
```

在该收发函数中使用 DMA 以非阻塞模式短时间发送和接收大量数据。hspi 指向包含 SPI 模块的配置信息。*pTxData 与 *pRxData 分别对应传输和接收缓冲区的指针，Size 为字节数。使用该函数时，需要两个缓冲区来保存数据，缓冲区的大小可以任意。通过 DMA 的方式，在传输数据时不会对 CPU 造成影响，效率也更高。

通过此收发接口，数据可以由主控芯片发出，写入墨水屏芯片中。墨水屏中 DC 引脚，可以将传输的数据可以分为两种类型，分别是数据传输和命令传输。根据这种区别可以编写写命令函数和写数据函数。

编写命令传输函数 SPI4W_WRITECOM(unsigned char INIT_COM)。在写命令函数中，INIT_COM 为需要写的命令字节，在显示屏使用中，命令字长度为 1 个字节。定义一个数组缓存该字节，并定义一个发送缓存数组。接着使用 nCS_L，让 CS 信号为低，完成显示屏选中。使用 nDC_L 语句对 DC 信号设置为低，代表该数据是命令。修改完信号线的状态，调用收发数据接口将缓存数组中的命令字节进行传输。

数据传输函数 SPI4W_WRITEDATA(unsigned char INIT_DATA)与命令传输函数一致，唯一的区别就是使用 DC 引脚的电平不同，为了确保传输的字节为数据，需要将 DC 置为高电平，即采用 nDC_H 语句。最后调用收发数据接口再将数据进行发送，完成写数据过程。

3.3.3 显示设置

墨水屏初始化中重要的一环就是对显示相关参数进行设置，包括电压驱动表，

输出顺序，显示空间的地址等。最终划分出一块用于缓冲显示数据的空间，大小是400*200。墨水屏大小为200*200，设置一个两倍大的空间便于显示时翻页。显示设置的流程可以归纳为图3.6。

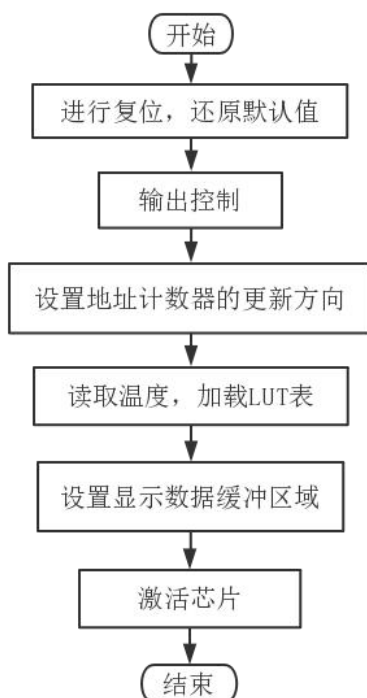


图 3.6 显示设置流程

这些设置需要使用的是墨水屏的指令表。在指令表中有多个命令字，每个控制命令下都有不同的数据，不同的数据代表不同的模式选择。找到相应的控制命令，按照指令表的命令和数据形式，可以对应地修改显示设置。

(1) 复位

根据墨水屏的指令表信息，使用写命令接口传输命令字0X12，可以触发复位操作。在墨水屏初始化中，首先要把原本的命令和参数重置为默认值。

(2) 输出控制

使用写命令函数，传输指令0X01，可以进入输出控制的相关设定。使用写数据函数传输0XC7, 0X00, 0X00这三个数据字节，分别对应不同的输出方式。其中0XC7

与 0X00 组合为数据位的 A0~A8，代表 200MUX。另外的数据 0X00，有效位可以用 B[2]、B[1]、B[0]代表。数据是 0X00，则 B[2]、B[1]、B[0]这三位分别为 0。B[2]=0，设定 G0 是第一个输出通道。B[1]=0，设定顺序 G0，G1，G2，G3...G199。B[0]=0，设定从 G0 扫描到 G199。从而完成输出通道和输出顺序的设置。

(3) 规定计数器的方向

写入 0X11 命令字节，进入计数器方向的设定。一般在初始化里默认使用 Y 递增 X 递减。计数器方向有四种组合方式，可以写入不同数据进行选择。00 代表 Y 递减，X 递减；01 代表 Y 递减，X 递增；10 代表 Y 递增，X 递减；11 代表 Y 递增，X 递增。在这里没有采用默认值，而是写入的数据 0X03，设定输入序列为 Y 递增，X 递增，地址计数器在 X 方向更新。

(4) 加载温度和 LUT 表

墨水屏的显示原理是基于温度检测和灰度值对应的 LUT 表。LUT 表为电压驱动表，其中记录有不同灰度与驱动电压的对应关系，在不同的驱动电压下，墨水屏的微囊粒子移动的距离不同，显示的灰度也不同。但是此表在用于灰度对应时需要参考当前温度，以免出现误差，故需要获取温度值。温度值是通过芯片的温度传感器感应的。如果温度值 MSByte 位 D11=0，则温度为正值，写入数据值为+(温度值)/16。如果温度值 MSByte 位 D11=1，则温度为负值，写入数据值为-(温度值的补码)/16。这些数据值会通过对应的映射表将转换为 HEX 格式。使用命令字 0x1A，进入温度传输功能，将这些 HEX 格式的温度值传输进去与 LUT 表一起使用。

(5) 设置缓冲空间

显示的数据需要存储在缓冲空间，所以要对缓冲空间进行设置，包括它的 X 和 Y 方向的起止位置和数据存入之后的像素设置。命令字 0x44，该命令用于控制缓存区域在 X 方向的开始和结束位置的设置。使用写数据函数传入两个数据，将开始地址设置为 00h，结束地址设置为 127h。使用 SPI 写命令函数传输命令字 0x45，进入缓存区域的 Y 方向地址开始和结束位置的设置。使用写数据函数传入起止位置值，将 Y 方向的起始位置设置为 127h，结束位置设置为 00h。同时还需要在 RAM 中设置 X 与 Y 的地址计数器，写入命令字 0x4E，0x4F，对应 X 和 Y 的地址计数器设置。

接着需要设置数据与像素的对应关系，写入 0x24 命令字，控制数据项被写入 BW (black white) RAM，随着数据的写入，地址指针将相应地前进，对于白色像素写入 RAM 的内容为 1，对于黑色像素写入 RAM 的内容为 0。

(6) 芯片激活

最后是进行主控芯片激活，使用命令字 0x22，在该命令字下有多种数据，对应多种显示模式和操作顺序。最终选择显示模式 1 下对应的 LUT 表，启用时钟信号，加载温度值，并且禁用时钟信号，被禁用的时钟信号当需要再次进行显示数据传输时会被开启，接收到相应的显示数据，进行显示。

结合上面这些步骤，通过使用编写的写命令函数和写数据函数，进行显示相关参数的设置，并且规划好了一块用于存储显示数据缓存空间，至此墨水屏的驱动初始化完成。

3.4 墨水屏传输实现

在墨水屏的显示方案中，当上层业务需要显示某些内容时，先将内容进行预处理，获取点阵数据，放在可以存储的位置，比如 RAM 或者 Flash，之后将存储的显示数据通过显示接口转换成屏幕影像显示出来。显示信息的传输流程如图 3.7 所示。

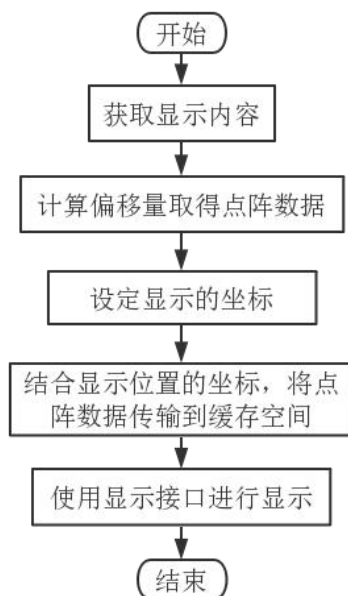


图 3.7 显示信息传输流程

墨水屏显示的内容可以分为图片，字符和汉字。为了能够使用到墨水屏上，还需要对这些显示内容做一些处理。对于图片，要把现在的图像处理成二值图像，处理图像的方式有多种，在这里采用的是 matlab 里的 `im2bw` 函数，该函数可以使用阈值变换法把灰度图像转换成二值图像。之后使用取模软件 `pctolcd2002`，把图片进行取模，获得对应的点阵数据。将这些点阵数据存储在头文件中，用于以后使用。对于字符，字符采用 24×16 的点阵数据，点阵数据通过字模软件生成。字符数据一共有 95 个，对应 `ascii` 码值 32 到 126 的字符，将这些点阵数据整合成数组与图片存放在同一头文件中。对于汉字，汉字采用 24×24 的点阵数据，经过纵向取模，高位在下的取模方式处理之后，一个汉字的点阵数据为 72 个字节数据，学生卡使用的汉字采用 `GBK` 编码，点阵数据也按此编码存放，经过对字库裁剪整合，将汉字的点阵数据生成一个 `hex` 文件，烧录在开发板中。

当获取到显示内容后，计算偏移量，寻址获取这些图片、字符、汉字的点阵数据。在初始化时，已经规划出一块 400×200 的缓存区域用于存放这些寻址到的点阵数据，现在需要将寻址到的点阵数据写到缓存中。编写显示数据写接口，伪代码如下

下：

```
    HINK_Write(const char *data, unsigned int x, unsigned int y, unsigned int w,  
unsigned int h){  
    for( int row = 0; row < h; ++row){  
        取对应的每一行数据  
        for(int col = 0; col < w/8 - 1; ++col){  
            将一维点阵数组数据按照位置对应关系转录到二维缓存区域中  
        }  
    }  
}
```

在此接口中，参数包括显示位置的横坐标 x ，纵坐标 y ，点阵数据 $*data$ ，宽度 w ，高度 h 。对于图片来说，高度宽度在对图片做处理时就获取到了，保存起来。对于汉字，高度宽度都为 24，对于字符，高度为 24，宽度为 16。这里的 x 、 y 、 w 、 h 是像素值，而数据是字节值，为了对应好位置，需要除以 8 来转换坐标。最外层的每次循环对应显示内容的每一行，内层循环就把每一行的像素按照对齐的原则拼成方便屏幕显示的格式，将一维数组数据与二维显示内容一一对应。通过该接口，完成了显示点阵数据 $data$ 到缓存中的写入过程。

3.5 墨水屏功耗管理策略

当前的墨水屏生产商在生产设计时就考虑到了功耗问题，尤其是 HINK-E0154A45，该产品主打特点是更小延迟更低功耗。墨水屏的工作模式一共有两种，一种的是普通模式，一种是休眠模式。在正常使用时，普通模式下的墨水屏由于本身特性，功耗较少。在休眠模式下，墨水屏直接熄屏，减少功耗。

墨水屏的功耗管理策略就可以借助普通模式和休眠模式的转换来进行。功耗管理策略如图 3.8 所示。

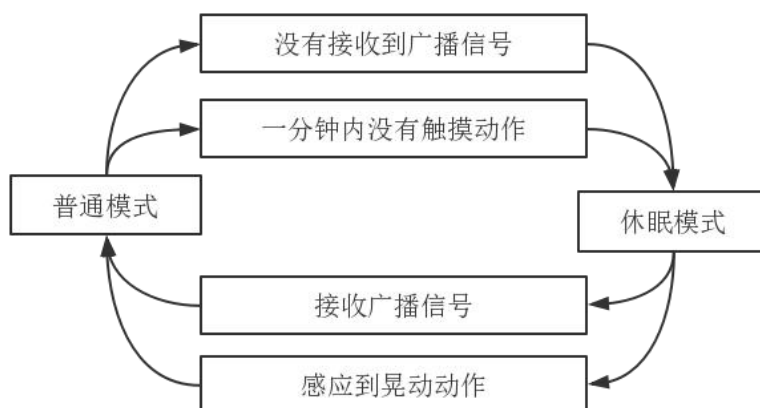


图 3.8 墨水屏功耗管理策略

进入休眠模式的判断围绕着定时器进行设计。方案一：设置扫描广播的定时器。当检测到校园卡在 3 分钟内没有扫描到广播时，这意味着外界没有发出和校园卡进行交互的业务，则让墨水屏直接休眠。方案二：设置感应触摸屏的中断的定时器。在定时器的时间范围内，一般设置为 1 分钟，如果有触摸而产生的中断，就重新开始计时，如果在时间范围内没有感应到中断，触发中断任务，墨水屏进入休眠模式。

当需要唤醒时，可以主动唤醒或者被动唤醒。被动唤醒是接收到广播包，要完成广播包的业务，就会退出休眠。主动唤醒需要人为摇晃智慧校园卡，通过加速度传感器模块感应到晃动动作，触发晃动中断，唤醒显示屏。

在模式转变上，使用 `enterdeepsleep` 函数进入休眠模式。在该函数中，不需要传入任何参数，使用写命令函数和写数据函数组合完成模式的设置。使用写命令函数通过命令字 `0X10` 进入模式选择状态。在此状态下，有两个数据字节可控制模式，分别为 `0X00`，`0X01`。数据 `0X00` 代表普通模式，数据是 `0X01` 时，代表休眠模式。故使用写数据函数，写入数据 `0X01`，控制芯片进入休眠模式，最后使用语句 `nCS_H`，拉高片选信号，取消选中。

3.6 本章小结

本章主要就智慧校园卡驱动设计和实现进行阐述。首先是对墨水屏的操作流程进行介绍。其次完成了墨水屏与主控芯片的连接配置，并且选择 SPI 为传输协议。在驱动实现中，首先是对 SPI 进行初始化，之后实现了 SPI 数据传输函数，通过该函数，使用指令表的命令，完成显示设置。最后实现了显示屏显示数据的传输接口，设计了功耗管理采用的办法。

4 加速度传感器驱动实现

本章主要是实现加速度传感器的驱动。加速度感应模块主要是对校园卡设计一些辅助功能，例如抬屏唤醒功能，计步功能。首先需要完成芯片的引脚连接，选择 I2C 作为传输协议，编写 I2C 传输函数和对加速度传感器寄存器的读写函数。使用读写函数对传感器驱动相关的参数进行设置。最后设计了误差补偿策略，实现了加速度值获取函数，方便上层应用随时获取准确的加速度值。

4.1 加速度传感器工作流程

博世公司出产了一系列加速度传感器，比如 BMA 系列，该系列适用于各类移动终端，被用于运动检测，角度倾斜计算，亮屏唤醒等。采用的芯片是 BMA253。BMA253 是低重力、数字式加速度传感器，能够测量物品在三维坐标系中 XYZ 三轴的加速度大小。该芯片长宽高都在毫米级别，引脚微小，具有超小的占位面积，适合扁平封装，因此非常适合校园卡这类卡片类产品。BMA253 的工作流程如图 4.1 所示。

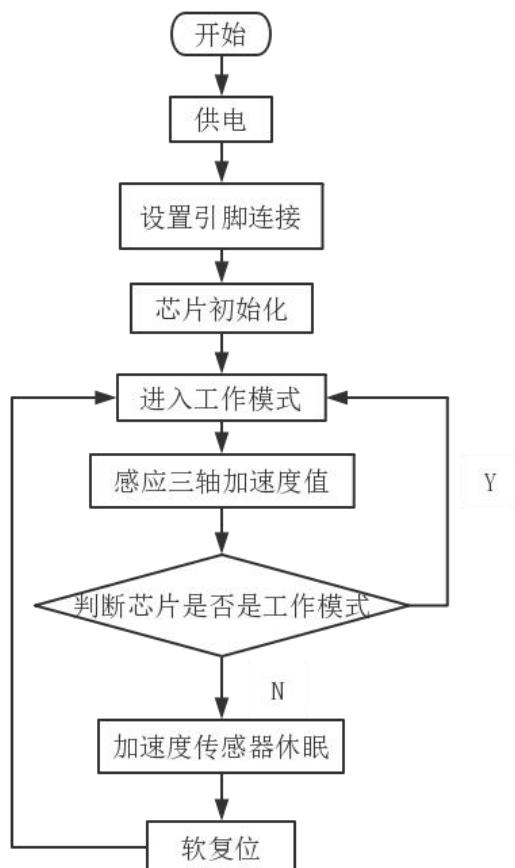


图 4.1 BMA253 工作流程图

BMA253 供电范围从 1.2V 到 3.6V 的可变，供电后，将芯片 PS、SCK、SDA 等信号设置完毕，即可开始初始化。在初始化中，主要是对芯片的模式性能进行设定。BMA253 加速度范围有 $\pm 2g$ ， $\pm 4g$ ， $\pm 8g$ ， $\pm 16g$ ，根据应用情况选择一个加速度范围，则可测量三轴的加速度值，经过读取整合，这些值可以递交给上层应用进行业务逻辑判断。BMA253 具备超低功耗，唤醒时间短，系统电源管理的高级功能。当主控芯片判别当前模式不是工作模式时，加速度传感器休眠，可通过软复位的方式退出休眠。

4.2 加速度传感器传输方案

4.2.1 BMA253 传输协议选择

BMA253 支持两种串行数字接口协议（SPI 和 I2C），用作主机设备与从机的通信。可以通过引脚 PS 来完成协议选择，引脚电平为 0 时是 SPI 协议，引脚电平为 1 时是 I2C 协议。默认情况下，SPI 以标准 4 线配置运行，它可以通过软件重新配置以 3 线模式工作。两个传输协议共享相同的管脚，选择不同的传输协议，引脚的功能不同。

I2C 是一种半双工的同步串行通信总线，共有两根线，分别是串行时钟线 SCL 和串行数据线 SDA，如图 4.2 所示。主控芯片负责规定时钟信号，并且发起通信，通过数据线向从设备传输信息。I2C 数据传输速率最高可以为 3.4 Mbps，传输速率基本满足嵌入式芯片的需求。

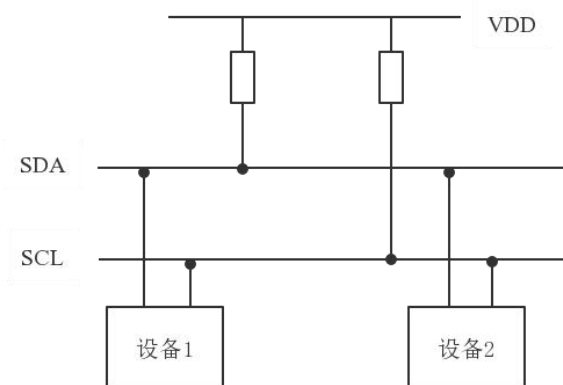


图 4.2 I2C 总线结构图

I2C 总线最大的优点是整体架构比较简单，只需要两根信号线，所需要的接口少，但是 I2C 的传输速度不如 SPI。SPI 协议适合用于数据量大，交互传输次数多，传输速度高的通信中，但是它比较消耗接口资源。加速度传感器获取的数据不需要经常

性通信，I2C 的传输速度可以接受。并且数据量也不是很大，还需要节省接口供其他硬件使用，故选择 I2C 传输协议。

在 I2C 的传输过程中，需要如下这几个信号。

START: 总线上的数据传输从 SDA 线路上的高到低转换开始，而 SCL 保持高位，故 SDA 电平的转换是 I2C 总线主机传输的启动条件。一旦发送启动信号，总线就被认定为 BUSY。

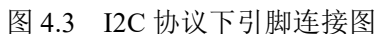
STOP: 每个数据传输都应该由主设备产生的停止信号来终止。STOP 信号是 SDA 线电平从低到高的转换。不支持 START 紧接着一个 STOP。如果发生这种情况，则设备无法识别 STOP。

ACK: 在数据的传输过程中，每次数据交互都需要确认信号 ACK，接收端发送确认信号表示已接收到数据。在确认脉冲期间，必须释放 SDA 线路，而接收器必须将 SDA 线路拉低，在确认周期内保持稳定的低电平。

在信号的辅助下，采用 I2C 的主从设备是这样进行数据传输的。数据的传输是由主机发送 START 信号开始的。从机从总线上获得开始信号后，就不断地监听总线，等待其他信号。接着，主机发送一个 8 位的数据帧，这 8 位由 7 位从机地址和一位读写状态位组成。当从机接收数据后，将数据帧中的从机地址与自己的地址进行对比。如果相同，则发送一个确认信号 ACK。如果不同，从机则等待 STOP 信号，准备退出传输过程。在确认地址相符后，便可以借助总线传输数据。最后主机通过发送 STOP 信号，结束传输流程。

4.2.2 BMA253 芯片的连接方式

由于不同的协议对应不同的引脚功能，所以引脚对应的连接方式不同。针对智慧校园卡，BMA253 采用的是 I2C 的传输协议，故引脚连接方式如图 4.3 所示。其中 11 号引脚是 PS，功能为协议选择，连接地线是选择 SPI 协议，连接 VDDIO 选择 I2C 协议，在这里连接 VDDIO。剩余的为供电引脚和数据传输引脚，按照规定方式连接好即可。



4.3 加速度传感器的传输接口

4.3.1 I2C 的收发操作

(1) 总线信号电平设置

I2C 所需信号线较少，只使用 SCL（串行时钟）和 SDA（串行数据输入输出）信号线。在 I2C 的读写过程中，需要对这两个信号线的电平进行高低转换，所以需要在这两个信号线进行高低电平设定。采用的方式同第三章，借助 GPIO 设置引脚状态函数 HAL_GPIO_WritePin 完成引脚电平设定。如下代码，SCL 与 PA0 引脚连接，通过写入 GPIO_PIN_SET，设定 SCL 的高电平。

```
#define IIC_SCL_H HAL_GPIO_WritePin(GPIOB,GPIO_PIN_0,GPIO_PIN_SET)
```

依照此方式逐个定义出 IIC_SCL_H（SCL 高电平），IIC_SCL_L（SCL 低电平），IIC_SDA_H（SDA 高电平），IIC_SDA_L（SDA 低电平）。

（2）I2C 起止应答信号设置

传输数据的过程中需要 START，STOP，ACK 等信号，用于主从设备之间传输过程的开始结束和确认应答。故在读写操作实现之前，需要实现这些信号函数。

开始信号：总线上的数据传输从 SDA 线路上的高到低转换开始，SCL 保持在高位。通过对这两个信号线高低电平的改变，可以完成开始信号的发送。具体步骤为：①将 SDA 设置为高电平。②将时钟信号 SCL 拉高，延时 4us。③将 SDA 电平拉低，使得电平由高变低，触发开始信号。④将 SCL 电平拉低，钳住总线。

停止信号：当 SCL 保持在高位时，STOP 条件是 SDA 线的电平从低到高的转换。故模仿上文开始信号的设置，依次对信号线进行更改。通过这三条语句 IIC_SDA_L，IIC_SCL_H，IIC_SDA_H，将 SDA 电平由低转变到高，完成结束信号的设置。

依次根据协议标准，结合对 SCL，SDA 高低电平设置函数和延迟函数，完成对 START、STOP、确认信号 ACK、等待确认信号 WAIT_ACK、无确认 NACK 等信号函数的设置，方便完成对 I2C 的读数据和写数据的接口设置。

（3）I2C 收发接口的实现

I2C 总线中数据线是用来传递数据，通过收发接口，主机可以在数据线上传递数据。

对于发送数据的接口 IIC_Send_Byte(unsigned char txd)，其中 txd 是要发送的数据。调用该函数，可以将数据发送至数据总线。在函数中，首先将时钟信号拉低。传输过程需要一个 for 循环来传输 8 位数据。将要传输的数据 txd 与 10000000 按位

相与，再右移 7 位取得传输数据中的最高位。如果最高位是 1，SDA 拉高，之后时钟信号处于高位，触发上升沿，开始传输该数据，传输完毕再把时钟信号拉低锁死。如果这位是低位，则直接 SDA 拉低。每传输完一位数据就进行移位，这样就完成了发送数据接口的设置。

对于接收接口 `IIC_Read_Byte(unsigned char ack)`，`ack` 是状态值。该接口用于接收读取数据线上的一个字节。使用该函数时，将 SDA 设置为输入，当 SDA 信号线有数据，就读取保存数据。使用 `for` 循环来读取这 8 位数据，采取的方式是进行移位，再相加，完成数据的读取，最终将数据作为返回值进行返回。同时根据传入的 `ack` 值反馈对应信号，由于是在接收数据，要在接收完毕之后回复相应确认信息。如果再调用时传入 `ack` 为 0，则发送确认信号，`ack` 不为 0，发送无确认信号。在实际接收数据时，一般都是需要在接收数据后发送确认信号。

4.3.2 加速度传感器的读写操作

BMA253 的读数据接口主要是通 `IIC_Read_Byte` 和 `IIC_Send_Byte` 这两个接口实现的。实现函数 `BMA253_ReadOneByte(uint8_t regaddr)`，输入的是想要读取的寄存器的地址值，返回值是读到的数据。BMA253 读数据流程见图 4.4。

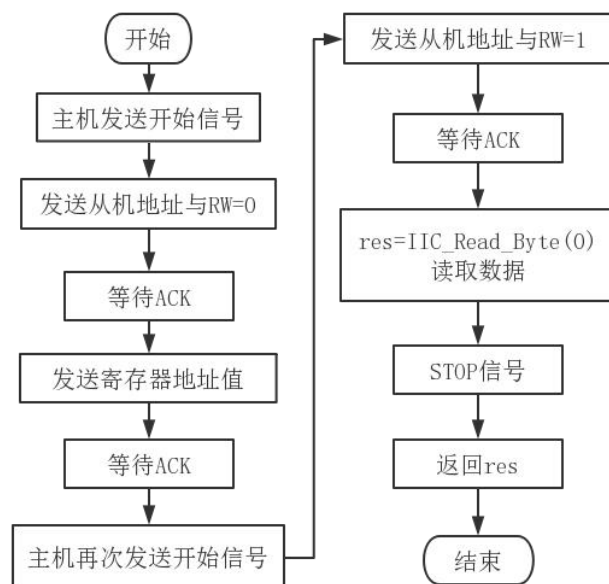


图 4.4 BMA253 的读数据流程

读取过程可用于在一个序列中读取一个或多个数据字节。对于一个字节的读取，由 I2C 写入阶段和 I2C 读取阶段组成。这两个阶段必须由重复开始条件（Sr）分开。第一步是写入阶段，主机发送开始信号，启动传输过程，并且使用 IIC_Send_Byte 发送从机地址并传输读取位 RW=0。接收到确认信号后使用 IIC_Send_Byte 发送要读取的寄存器地址，等待从机确认信号。第二步是读取过程，主机再次生成开始信号，并将从机地址与读取位（RW=1）一起发送。收到确认信号后，主机释放总线，等待 IIC_Read_Byte 读取字节数据。最后主机可以生成 STOP 信号并终止传输，返回读取的数据。

BMA253 的写数据接口主要是通 IIC_Send_Byte 这个接口实现的。实现函数 BMA253_WriteOneByte(uint8_t regaddr, uint8_t data)，regaddr 是寄存器地址，data 是要写的的数据。BMA253 写数据流程见图 4.5。

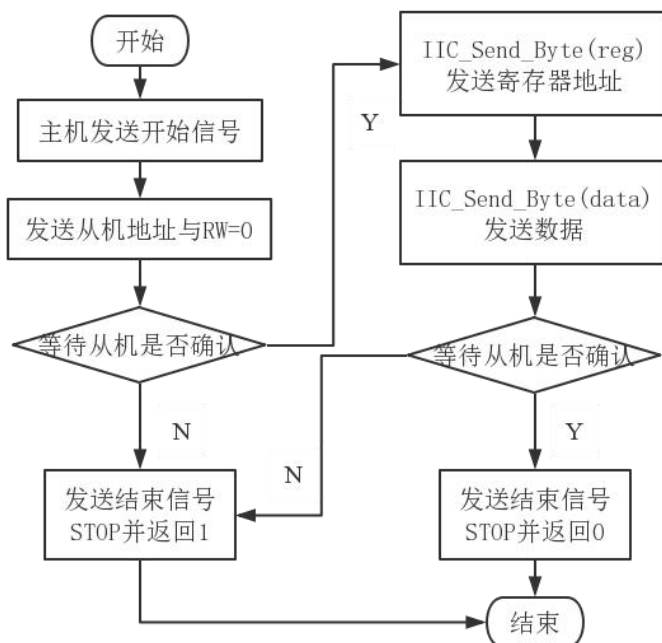


图 4.5 BMA253 写数据流程

BMA253 的写过程用于在一个序列中写入字节数据。主机发送开始信号，开启写数据过程。接下来使用 IIC_Send_Byte 传输 7 位从设备地址和一个写入位(RW=0)。之后等待从机发送一个确认信号，收到后释放总线（如果没有收到确认信号则直接结束写过程）。然后主机发送单字节寄存器地址。从机收到后再次发送确认信号并等待 8 位数据写入指定的寄存器地址。等待从机再次确认收到数据字节后，主机产生停止信号，并终止写入过程。

4.4 加速度传感器驱动初始化

加速度传感器驱动的初始化可以分为两步：第一步进行 I2C 的初始化，完成协议的设置；第二步是完成芯片的初始化，激活芯片，使得芯片功能使能。

4.4.1 I2C 初始化

加速度传感器的初始化中，首先要对 I2C 传输协议进行初始化设定。在 I2C 的

初始化中需要完成 I2C 功能函数的封装和 I2C 参数设置，完成这些之后，判断参数是否合法，合法则执行。I2C 初始化流程如图 4.6 所示。

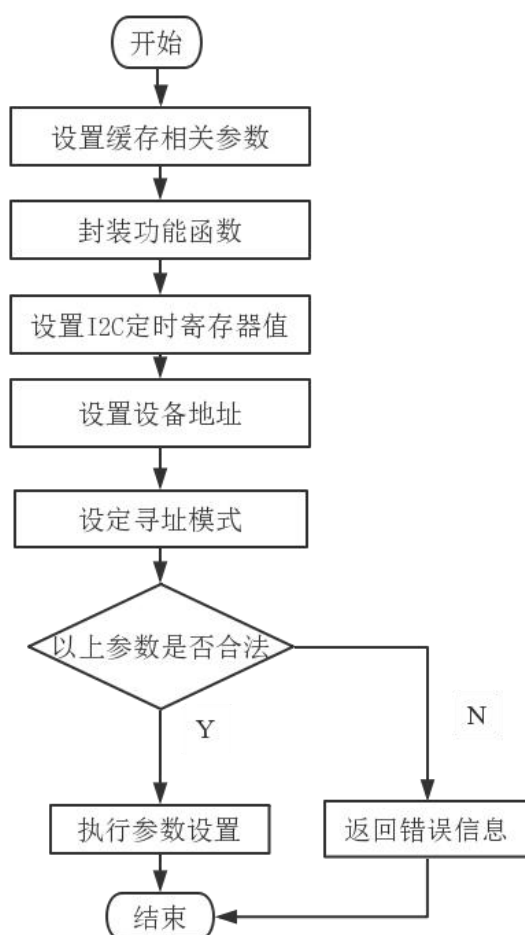


图 4.6 I2C 初始化流程

（1）I2C 功能函数的封装

I2C 协议在传输过程中需要设置缓存空间的位置和大小，缓存空间的计数器，用于传输过程。同时还需要一些功能函数，包括主设备发送回调函数，接收回调函数，以及传输结束回调函数，从机地址匹配回调函数，错误提示函数等。通过这些功能函数，完成了缓冲区域的设置，状态标志的设置，以及调试信息提醒，辅助 I2C 传

输协议的实现。

(2) I2C 参数设定

I2C 相关的参数通过一个结构体定义，主要是设置地址和传输模式，具体如下结构体所示：

```
typedef struct
{
    uint32_t Timing;           //定时器设置
    uint32_t OwnAddress1;      //设备地址
    uint32_t AddressingMode;
    uint32_t DualAddressMode;
    uint32_t OwnAddress2;
    uint32_t OwnAddress2Masks;
    uint32_t GeneralCallMode;
    uint32_t NoStretchMode;
} I2C_InitTypeDef
```

在此结构体中，Timing 用于指定 I2C 定时寄存器值。OwnAddress1 为设备地址值，设定为 0x70。AddressingMode 用于指定寻址模式，由于从机地址是选择的 7 位，所以寻址模式也是 7 位。DualAddressMode 用于指定双寻址模式，校园卡不需要双寻址，故禁用。因为没有使用双寻址，OwnAddress2（第 2 地址）为空，与双寻找相关的 OwnAddress2Masks（掩码地址）也不需要设置，GeneralCallMode，NoStretchMode 等全部禁用。

4.4.2 BMA253 的初始化

BMA253 芯片初始化是通过寄存器来控制的。在多个寄存器中，每个寄存器都控制不同的参数，完成对芯片的控制。这些寄存器在芯片设计时就被提前规定好，寄存器的宽度为 8 位，它们映射到 64 个地址的公共空间，地址是从 0x00 到 0x3F。在这些寄存器中，地址从 0x00 到 0x0E 的寄存器是只读的，对这些寄存器的任何写

入尝试都将不能执行，并且也有一些寄存器必须配置为只写寄存器。通过加速度传感器的写操作，将这些寄存器配合存入不同的数据，可以用来完成芯片的初始化。

BMA253 初始化流程如图 4.7 所示。

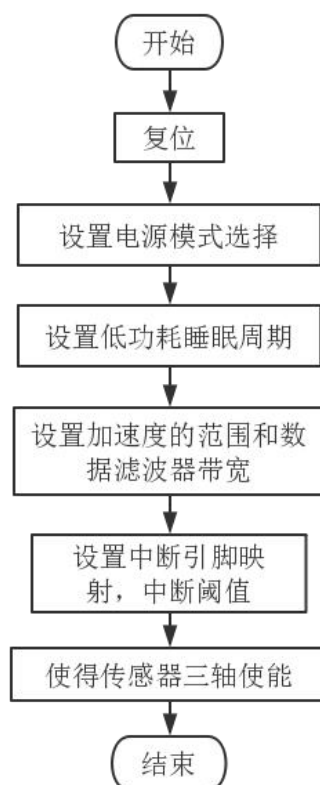


图 4.7 BMA253 初始化流程

首先进行加速度传感器的复位操作。地址为 0X14 的寄存器是控制复位的，写入的数据是 0xB6，该数据用于触发复位。接下来就是传感器模式的设置，使用到的寄存器地址为 0x11, 0x12, 0x0F, 0x10。地址为 0x11 寄存器是控制主电源模式选择及低功耗睡眠周期，传感器为了满足功耗要求，一共有五种低功耗模式和一种正常模式，传入数据为 0x00，选择主电源模式是正常模式。寄存器 0x12 用于配置低功耗模式，传入数据为 0x00，代表选择 LPM1 模式。寄存器 0x0F 与 0x10 分别设定加速度传感器的范围和带宽，设置加速度传感器的范围是 2g，带宽是 31.25Hz。

接下来是中断设置，使用中断相关的寄存器，进行可选数据源的中断的定义。寄存器 0X21 用于控制中断复位和中断模式选择，写入数据 0x03，选择中断有效，时间为 1 秒。关于中断引脚 INT1 和 INT2，在初始化中首先保持禁用，还原到初始状态。0x20 寄存器包含中断引脚的行为配置，将中断 1 和中断 2 都设置为低电平，把梯度信号映射到中断 1。再对中断的阈值进行设定，设定为 0.5g。当需要中断时，可以再次打开，按照之前设定的参数使用。

最后通过地址为 0x16 的寄存器将加速度传感器三轴使能，三轴能够正常采集数据，加速度传感器的初始化完成。

4.5 传感器加速度值获取接口的实现

4.5.1 加速度值的补偿策略

在测量数据时，难免会有信号失调的情况，导致数据出现偏差，为此，需要进行一定的补偿。具体的补偿方式有四种，分别是慢速补偿、快速补偿、手动补偿，在线校准。慢速补偿是基于高通滤波器来实现的，通过地址为 0X37 的寄存器可以更改带宽。快速补偿是一次性过程，在出厂之前，计算 16 个加速度的平均值，将平均值和目标值的偏差写入三轴对应的偏移寄存器中，在程序中设置偏移触发命令，进行快速补偿。人工补偿即手动通过传输接口，将自定义的补偿值写入偏移寄存器中。在线校准工作流程是每当校准一次失调，就将补偿值存储起来，当再一次失调时，即可取用以前的补偿值进行校准，直到被其他补偿方式取代。考虑到校园卡的使用场景，测量值出现误差的原因是温度极端，由于学生卡自身工作产生的热量不高，在校使用时，高温环境比较少见，故卡片的高温情况比较少见，反而是在冬季夜间容易出现低温情况，所以需要在低温情况下进行补偿。

选择使用快速补偿作为加速度值的补偿策略。首先，使用芯片的自测功能，通过内部施加力的方式，对正负方向进行自检，确保器件完好。然后在低温环境下，测量出加速度值，共测 16 组取平均，将该值和正常温度下的加速度值进行比对，计算出偏移值，存放在 0X38、0X39、0X3A 这三个寄存器中。一旦判断温度极端，则

使用地址为 0X36 的寄存器来控制补偿。在该寄存器中，存入是 8 位数据。0 到 3 位依次是 0111，目的是让三轴补偿使能。第 4 位设置为 0，启用快速补偿，第 5 到 6 位可以设置为 01、10、11，分别代表对 X、Y、Z 轴进行补偿。第 7 位设置为 0，代表使用存放在偏移寄存器的偏移量进行补偿。通过对该寄存器写入不同的数据，激活三轴的快速补偿，就可以一次性完成对三轴加速度值的调整。

4.5.2 加速度值的获取

BMA253 传感器通过重力的变化而获取加速度值，芯片嵌入校园卡后，三轴的加速度矢量的定义如图 4.8 所示。

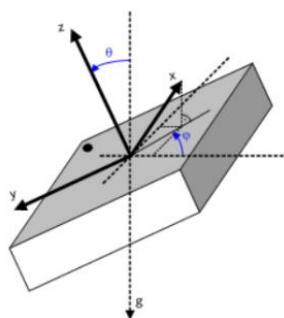


图 4.8 三轴加速度矢量定义图

传感器采集到数据使用的单位是 256LSB/g，也就是采用 2G 的范围。x 轴方向采集到的数据是 256 的话，则 X 轴加速度是 1g。三轴加速度传感器感应到的数据是存放在寄存器当中，地址为 0x02 到 0x07。寄存器对应关系如表 4-1 所示。

表 4-1 加速度传感器三轴寄存器对应关系

数轴	寄存器名称及地址	
X 轴	ACCD_X_LSB 0X02	ACCD_X_MSB 0X03
Y 轴	ACCD_Y_LSB 0X04	ACCD_Y_MSB 0X05
Z 轴	ACCD_Z_LSB 0X06	ACCD_Z_MSB 0X07

ACCD_X_LSB，该寄存器包含 X 方向加速度读出值的最低有效位。除了在通电和深度睡眠期间，加速度的数据可以在任何时候从寄存器 ACCD_X_LSB 读取。在 ACCD_X_LSB 的 8 位数据中，4 到 7 位是加速度回读值的最低有效 4 位，第 1 到 3

位数据为忽略数据，未定义的随机数据，第 0 位是标志位，0 代表自上次读取加速度值以来，尚未更新该值，1 代表加速度值自上次读取后已更新。ACCD_X_MSB 寄存器包含 X 方向加速度读出值的最高有效位，存储的 8 位数据代表加速度回读值的最高有效 8 位。剩余的 Y 轴 Z 轴的 4 个寄存器数据分布同理。

根据补偿策略和寄存器的有效位，加速度值的获取流程如图 4.9 所示。

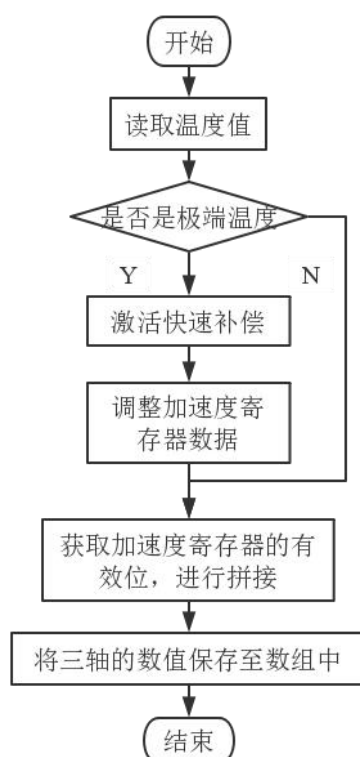


图 4.9 加速度值获取流程

首先，使用加速度传感器读数据接口 BMA253_ReadOneByte 读取地址为 0X08 的温度寄存器的数值，判断当前温度是否是低温。如果是，使用地址为 0X36 的寄存器激活快速补偿，使用存储好的偏移值，对存储在加速度寄存器的值进行调整。如果不处于低温环境，则直接使用读数据接口读取这六个加速度寄存器的值，将有效位进行拼接。对应 X 轴，有效位是这样获取的：存放 ACCD_X_MSB 中的数据左移 4 位，存放 ACCD_X_LSB 中的数据右移 4 位，将移位后数据进行拼接，即可获得 x

轴上的加速度数据。其他两轴同理。最后将拼接后的数据保存至提前定义好的数组中。

4.6 本章小结

本章主要实现加速度传感器的驱动。使用 BMA253 芯片，介绍了芯片的结构，功能和连接方式。选择 I2C 总线进行数据传输控制，完成了总线传输函数，通过这些函数实现加速度传感器的驱动初始化。最后设计了误差补偿策略，完成了加速度值获取接口，以便于上层使用准确的加速度值触发相应功能。

5 测试

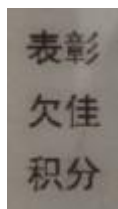
本章节主要是对之前设计实现的模块进行测试，确认功能是否能够使用。

5.1 测试环境

使用烧写工具连接到 PC，烧写项目代码到校园卡中对功能进行验证。通过智慧校园卡显示屏的显示状况进行显示屏模块的测试，通过 PC 端输出调试信息，读取加速度传感器感应到的数值。

5.2 显示屏模块的测试

显示屏需要显示图片，字符，汉字等信息，能显示这些信息，就覆盖了显示屏的所有显示业务。如图 5.1 到 5.3 所示，分别是墨水屏显示汉字、字符、图片等信息。



5.1 汉字信息显示

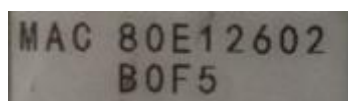


图 5.2 字符信息显示



图 5.3 图片信息显示

在 UI 界面中，需要字符、图片与汉字等信息组合显示，共同构成整个界面的内容。不同的信息在屏幕上显示的位置不同，按照合理的坐标分布开来，如图 5.4 时间显示界面所示，将时间数字的坐标设置在屏幕中部，其他汉字和图片在空余处排开。



图 5.4 时间显示界面

经过反复调取显示各类界面信息，字符、图片与汉字的显示功能能正常实现。

5.3 加速度传感器模块的测试

加速度传感器模块的测试主要是在卡片不同姿态下感应三轴的加速度的值。加速度传感器的测试值见表 5-1。

表 5-1 加速度传感器测试表

卡片姿态	x	y	z
水平（卡片横向正面朝上）	-2	-10	-256
水平（卡片横向背面朝上）	0	6	256
垂直（卡片横向正面朝人）	3	-256	-17
垂直（卡片横向背面朝人）	1	256	-5
垂直（卡片竖立挂绳朝上）	254	-3	-23
垂直（卡片竖立挂绳朝下）	-255	-10	-16

卡片的三轴，以卡片水平放置为例，X 轴负方向为横向右手侧（挂绳方向），Y 轴正方向为垂直 X 轴向上，Z 轴正方向垂直水平面向上。当卡片静止时，会有某一轴受到重力加速度的影响，数值为 256，代表 1g。忽略在调整姿态时产生的数值变动，单独测试一轴。水平朝上放置，Z 轴负方向受到重力加速度，值为-256，水平朝下时，Z 轴的正方向受到重力加速度，为 256。同样的，当垂直放置卡片横向时，Y 轴负方向受到重力加速度，为-256，逆转方向，值为 256。在垂直卡片竖立时，检测 X 轴，挂绳朝上为负方向。竖立挂绳朝上时，X 轴正方向受到重力加速度，数值应为 256，由于手持的误差，可能不是完全垂直，故数值略小，逆转方向同理。根据不

同姿态对比测试，判断三轴可以正常测量。

5.4 本章小结

本章根据校园卡的需求对显示模块和加速度感应模块进行测试，测试了校园卡不同显示内容的显示效果，测试了加速度模块在卡片不同姿态下三轴的数值，测试结果表明能够正常工作。

6 总结与展望

本章节主要是对上文的归纳总结，同时归纳一些不足和后期的展望。针对当前智慧校园系统，主要研究实现智慧校园卡的驱动技术。在校园卡设计的过程中，针对关键驱动进行设计，并且要根据实际功能做出一些具体的调整，设计出符合产品功能需求的驱动。

6.1 总结

随着智慧校园的提议被推广到各个省市的中小学，智慧校园系统所需要的设备必须随之设计使用，以便于智慧教学的推进。校园卡作为智慧校园系统中常见的终端设备，承担了教师与学生进行信息交互的责任。在智慧校园卡项目的开发中，针对驱动实现进行阐述。本文的主要内容如下：

首先对校园卡的研究背景、研究意义、实际使用场景进行介绍，描绘了一个智慧校园卡辅助的智慧教学场景，讲述了智慧校园卡的主要功能。校园卡的整体框架，包括硬件底层的构成和软件架构。硬件方面主要说明了硬件主板的主要模块，介绍了采用的芯片的型号和相应的特性。软件方面主要是介绍校园卡软件系统构成，包括应用层，支撑层和驱动层，介绍了具体的功能需求。最后对驱动的设计提出一种设计思路，抽象出驱动设计框架，通过框架去实现校园卡的驱动。

其次是显示屏的驱动实现。首先进行硬件连线，即确定主控芯片和显示屏引脚连接，例如时钟引脚，片选引脚，数据命令引脚。其次需要选择传输协议，墨水屏适用 4 线 SPI 总线，要配置 SPI 参数，进行初始化，完成写数据和写命令的接口。通过芯片初始化的流程，结合指令表，完成显示模式的选择，缓存区域的规划，将显示屏初始化。最后实现显示数据的传输接口，将汉字、字符、图片数据写入缓存。并且还对显示屏的功耗管理进行设计，尽可能的让其在使用过程中减少电量损耗，延长待机时间。

最后是加速度模块的驱动设计，选择 BMA253 作为传感器芯片，该芯片在移动

终端中广泛使用，并且支持多种总线。选择 I2C 总线作为通信方式，对总线进行初始化，设定读写接口。之后根据芯片的原理，对不同的寄存器写入数据，规定电源模式和中断信号，完成对芯片驱动的初始化。最后设计了补偿策略，编写加速度值获取接口，给上层提供这个接口，方便获取准确的加速度值。

6.2 展望

就目前的设计看，还有一些需要改进的地方。首先，硬件的存储空间有限，还要存储一些课堂数据，再加上后期的业务代码，所以在驱动代码上，或许需要进行精简，删除一些不需要的接口。比如在显示功能上，缓存接口和显示接口需要更加精简的方式，目前这种不断对缓存进行清除再显示的操作在实际使用时会影响实时性。在实际测试中，要根据使用情况对驱动进行调整。比如在加速度传感器模块，需要准确的加速度值。针对加速度值的极端环境下的补偿，由于现在没有低温环境，无法实现校园卡低温下的使用场景，所以无法对补偿策略进行实际测试。

致谢

来到华中科技大学读书的这两年，我度过了一段非常有意义的时光，两年的校园时光，有任课老师讲授专业知识，有导师的悉心指导，有朋友同学们的陪伴关照，让我的学业生涯充满了乐趣。在此，我要对出现在我这两年学习生涯中的人表示感谢。

首先感谢我的指导老师胡贯荣老师，胡老师是我的研究生学业生涯中的指明灯，在平时生活和科研学习中给予我关怀和帮助。胡老师知识渊博，在生活上也有很多经验，在这两年中，他悉心教导，让我成长了很多。

其次感谢师门下各位学长学姐，同届同学们对我的帮助，在学长学姐们的帮助下，我懂得了研究生生涯该如何去规划，在同学们的陪伴下，我的研究生生活变得丰富起来。

最后感谢在疫情期间帮助我论文进展的每一个人，虽然距离很遥远，但是心意还是那么温暖，他们的关心、帮助，让我总是充满信心，无比坚定。

参考文献

- [1]陈文智.嵌入式系统原理与设计.第2版.北京:清华大学出版社,2017:4-8.
- [2]蓝均政.中小学智慧校园建设的探索.教育信息技术,2020,40(Z1):142-144.
- [3]Atif Y , Mathew S S , Lakas A . Building a smart campus to support ubiquitous learning. Journal of ambient intelligence and humanized computing, 2015, 6(2):223-238.
- [4]王同聚, 张嘉志. 中小学"智慧校园"建设和应用模式探索. 中小学信息技术教育, 2014, 000(001):61-64.
- [5]国家市场监督管理总局, 国家标准化管理委员会.智慧校园总体框架 (GB/T 36342-2018).第1版.北京:中国标准出版社, 2018:3-5.
- [6]王智慧, 刘沅元. 物联网技术在智慧校园中的应用. 电子技术与软件工程, 2018, 000(006):20.
- [7]李成渊.射频识别技术的应用与发展研究.无线互联科技,2016,12(20):146-148.
- [8]Boranbayev A S . Defining methodologies for developing J2EE web-based information systems. Nonlinear Analysis Theory Methods & Applications, 2009, 71(12):33-37.
- [9]Wu Jie,Yong Jianping,Wang Liangmin. An AOP Based Light-weight J2EE Architecture[J]. Journal of Convergence Information Technology,2012,7(20):328-334.
- [10]张文增, 曾艳丽. 基于物联网的校园一卡通的设计. 电子技术与软件工程,2019,7(24):13-14.
- [11]张相飞,周芝梅,王永刚,等.NFC技术原理及应用.科技风,2019,31(05):69-70+75.
- [12]吴雨欣.基于手机NFC校园手机一卡通系统.电子世界,2019,40(16):178-179.
- [13]陈阳.基于手机NFC技术的行业卡充值实现与能力开放平台构建.软件导刊,2020,19(01):203-206.
- [14]唐钢,张龙信,文鸿.基于NFC的校园信息管理系统设计与应用.电子世界,2019,40(04):117-118.
- [15]黄盼,刘淑娴,陆明中,等.STC12和RFID的手环式校园一卡通系统设计.单片机与嵌入式系统应用,2019,19(08):76-80.

- [16]代少玉, 吴兴林. 单片机原理与应用技术: STC12 系列 Keil C 工程实验. 西安: 西安电子科技大学出版社, 2017:48-49.
- [17]Long Guo Jin, Lian Tao Cui. Development and Application on Intelligent Sub-Station of Automatic Production Line Based on the STC12C5A60S2 MCU.Advanced Materials Research, 2013, 2493:807-810.
- [18]杨佳丽,史恩秀,王晔,等.基于无线网络的室内移动机器人定位方法研究.机械科学与技术,2013,32(03):457-461+468.
- [19]Cliff Drubin. Bluetooth 5 Will Lead to Widespread Deployments on IoT Landscape. Microwave Journal,2016,59(9);59-60.
- [20]晏军,刘鸿源,曹阳.基于 RFID 技术的无人超市系统设计与实现.电子技术与软件工程,2019,7(24):72-73.
- [21] David Flint. RFID tags, security and the individual. Computer Law and Security Review: The International Journal of Technology and Practice,2006,22(2):165-168.
- [22]Amjad Ali Alamr,Firdous Kausar,Jongsung Kim,Changho Seo. A secure ECC-based RFID mutual authentication protocol for internet of things. The Journal of Supercomputing,2018,74(9):4281-4294.
- [23]Changjiang Xiao, Nengcheng Chen, Dandan Li, et al. SCRMS: An RFID and Sensor Web-Enabled Smart Cultural Relics Management System.Sensors, 2016, 17(1):60.
- [24]Nicole Todtenberg,Rolf Kraemer. A survey on Bluetooth multi-hop networks. Ad Hoc Networks,2019,93:101922.
- [25]苑尧尧,曹佃国,迟祥,等.基于低功耗、双模蓝牙通信的心电监护系统设计.通信技术,2019,52(10):2573-2580.
- [26]Muhammad Imran,Sheikh Tahir Bakhsh,Sabeen Tahir,et al. A reconfigurable scatternet formation and maintenance scheme with heterogeneous services for smart Bluetooth devices. Sustainable Cities and Society,2018,40:589-599.
- [27]张凯.基于 STM32 的无线蓝牙传输设计.物联网技术,2020,10(03):40-41+44.
- [28]Nikodem Maciej,Bawiec Marek. Experimental Evaluation of Advertisement-Based

- Bluetooth Low Energy Communication. Sensors (Basel, Switzerland),2019,20(1).
- [29]王灵.财政收入信息管理系统设计与实现.电子世界,2013(23):138-139.
- [30]许子明,田杨锋.云计算的发展历史及其应用.信息记录材料,2018,19(8):66-67.
- [31]Shin Kamada, Takumi Ichimura, Tetsuya Shigeyasu,et al. Registration system of cloud campus by using android smart tablet. SpringerPlus, 2014, 3 (1):1-13.
- [32]单祥茹.意法半导体 STM32 家族首次引进片上无线通信功能.中国电子商情(基础电子),2018,23(11):17-18.
- [33]贺轶烈,许晓荣,楼丁漂,等.基于电子墨水屏的无线电子标签设计.电子设计工程,2019,27(06):138-141.
- [34]赵威,孙绍辉,曹勇.基于超低功耗无线通信技术的电子墨水标签显示系统研究.电力信息与通信技术,2018,16(04):54-58.
- [35]张志永.电子墨水技术在智慧城市应用研究.工业控制计算机,2016,29(03):106-107.
- [36]黄虎,陈诗瑶,卿松,等.一种基于 NFC 的电子墨水屏货物标签显示牌.单片机与嵌入式系统应用,2019,19(02):60-63.
- [37]浩钿,王猛,陈聪,等.基于无线通信和墨水屏的电子标识技术研究.无线互联科技,2019,16(09):3-5.
- [38]贺志强. 基于加速度传感器的人体能耗检测系统的设计与实现:[硕士学位论文].哈尔滨工业大学:哈尔滨工业大学图书馆,2016.
- [39]陈启山,许杰,杨德俊.一种石英三角剪切标准加速度传感器的研制.电子质量,2019,39(11):89-94.
- [40]肖咸盛,卞玉民.一种高 g 值压阻式加速度传感器.微纳电子技术,2017,54(04):261-267+284.
- [41]柯化成. 嵌入式系统全系统模拟器框架设计与实现:[硕士学位论文].浙江大学:浙江大学图书馆,2006.