第3章

物端开发实验

3.1 实验环境配置

3.1.1 硬件环境概述

物联网感知层的核心设备是开发板,这类设备通常由芯片、通信模组、操作系统等模块 组成。本章立足于国产技术和产品,基于华为 LiteOS 介绍开发实验,因此芯片和通信模组 的选择也应与 LiteOS 适配。

华为 LiteOS 支持多种芯片架构,包括 ARM 的 Cortex-M0、Cortex-M0+、Cortex-M3、 Cortex-M4、Cortex-M7、Cortex-A7、Cortex-A9、Cortex-A53 系列、ARM64 的 Cortex-A72 系列、RISC-V 的 RV32 系列、C-SKY 的 CK802 系列等。一般来讲,嵌入式设备不仅芯片差 异大、外设种类多,而且资源有限,因此物联网操作系统(如 LiteOS)无法像计算机操作系统 (如 Windows、Linux 等)那样,适配并集成所有驱动,通常的做法是对部分开发板完成适配, 而对其他开发板进行操作系统的移植。

为了将目标聚焦于物联网系统的开发而无须考虑系统移植的问题,本章选择已适配且 应用较为广泛的小熊派系列开发板作为实验硬件,该开发板具有简单易用、生态良好、模块 集成度高等优点。需要说明的是,本章的实验参考了小熊派开发板的用户手册。以下对该 开发板进行简单介绍。

小熊派高性能物联网开发板由南京小熊派智能科技有限公司与华为技术有限公司联合 出品。该开发板基于 STM32L431RCT6 设计,从物联网的感知层角度来讲,具有设备多样 性和可延展性,可以为开发者提供一个评估和设计物联网产品的平台。

1. 开发板的功能

将小熊派开发板进行功能区的划分,如图 3-1 所示。 其中:

(1) SD card。指示 SD 卡卡槽区域,可插 SD 卡。

(2) Flash。指示存储器区域,可存储程序等。

(3) ST-link。指示单片机的下载器区域。

(4) USB Power supply。指示 USB 接口区域,可在下载、调试代码时提供电源。

(5) Power LED。指示 LED 区域,可根据上电、下载、用户定义状态等亮灯。

(6) E53 interface。指示 E53 接口区域,可兼容具有 E53 接口的传感器扩展板。

(7) UART SW。指示串口开关区域,可用来调试通信模块。



图 3-1 小熊派开发板的功能区划分①

(8) LCD。指示显示屏区域。

(9) WAN Module interface。指示通信扩展接口区域,可接入 NB-IoT、2G、Wi-Fi 等采 用不同通信方式的通信扩展板。

(10) MCU: STM32L431RCT6。指示采用了 STM32L 系列单片机。

(11) Reset Key。指示系统的 Reset 按键,可实现系统的自动复位重启功能。

(12) User's Key。指示功能键区域,可通过 F1、F2 按键实现定义的功能。

2. 开发板的框架

小熊派开发板的系统框图如图 3-2 所示。



图 3-2 小熊派开发板的系统框图^②

① 图片来源于网络。

② 图片来源于网络。

其中,各部件的连接关系为:

(1) 开发板经由 USB 5V 供电并通过内置 DC-DC(直流电源变换器)降压至 3.3V 进行 系统内部供电。

(2) ST-Link 通过 SWD(串行线调试接口)接口与 MCU 互连。

(3) 8MB Flash 通过 QSPI(4 线串行外设接口)与 MCU 互连。

(4) SD 卡通过 3 线 SDMMC 与 MCU 互连。

(5) E53 扩展支持 SPI(串行外设接口)、I²C、UART(通用异步收发器)协议。

(6) 1.44 英寸的 LCD(液晶显示器)通过 SPI 与 MCU 互连。

(7) 通信模块扩展接口支持 SPI、UART 协议。

(8) LED、按键部件直连 MCU 的 GPIO(通用输入/输出)。

3. 主要外设接口

基于后续实验的需要,下面介绍 MCU 主板的主要外设接口,包括 USB 接口、按键、E53 接口、通信扩展接口、SPI 接口和预留接口。

1) USB 接口

开发板含有一个 USB 接口,为 USB ST-Link 接口,作为软件下载、调试和系统供电的 输入口。USB ST-Link 接口除了给系统提供电源之外,还是开发板的下载接口,与 STM32F103 的 USB 接口相连接,用 USB 数据线连接至 PC 之后会映射出一个 COM 口设备,用来进行开发板和 PC 端之间的交互,打印开发板的调试信息,下载 MCU 程序,调试通 信模组。

2) 按键

开发板带有两个功能按键和一个系统 Reset 按键。功能按键可以提供给开发者进行功能定义开发,均使用 GPIO 口,方向为输入,低电平有效。Reset 按键是直接接入 STM32F103 和 MCU 的硬件复位 Pin,按下 Reset 按键,系统将自动重启复位。

3) E53 接口

开发板设计有 E53 接口的传感器扩展板接口,该接口可兼容所有 E53 接口的传感器扩展板,实现不同案例场景的快速搭建。该接口可接入 UART、SPI、I²C、ADC(模拟数字转换器)等通信协议的传感器。

4) 通信扩展接口

开发板设计有通信扩展板的扩展接口,该接口可接入 NB-IoT、2G、Wi-Fi 等采用不同通 信方式的通信扩展板,以满足不同场景下运行的需求。

5) SPI

SPI是 LCD 显示屏的接口,开发板板载一个 FPC 材质的 LCD 屏幕,屏幕的分辨率为 240×240dpi。

6) 预留接口

预留接口有一组 UART 和一组 I²C 接口以及两个通用 I/O 口,可供开发者自定义开发 使用。

3.1.2 软件环境概述

软件的编辑需要文本编辑器,编译需要编译器,汇编需要汇编器,链接需要链接器,

第 3 章 可执行文件需要软件工具来加载文件,同时软件还需要一些函数库、中间件等。为了使 开发更便捷、简单,几乎所有的 MCU 芯片都会有对应的集成开发环境(IDE),该环境囊括 了软件开发从编辑到可执行文件的所有工具,同时还包括常用的库、调试工具、在线调试 工具链等。

STM32开发主流的集成开发环境有两种: MDK(微控制器开发套件)和 IAR(Systems 公司的一款 IDE),此外,华为自研的 LiteOS Studio 也是一个不错的选择,下面简要介绍这 三种集成开发环境。

1. MDK-ARM

MDK-ARM软件是 Keil 公司推出的一款产品,为基于 Cortex-M、Cortex-R4、 ARM7、ARM9处理器的设备提供了一个完整的开发环境。MDK-ARM专为微控制器 应用而设计,不仅易学易用,而且功能强大,能够满足大多数苛刻的嵌入式应用需求。 MDK-ARM有四个可用版本,分别是 MDK-Lite、MDK-Basic、MDK-Standard、MDK-Professional。所有版本均提供一个完善的 C /C++开发环境,其中 MDK-Professional 还 包含大量的中间库。

2. IAR for ARM

IAR for ARM 全名为 IAR Embedded Workbench for ARM,是一款由瑞典 IAR Systems 公司推出,专为微处理器开发的优秀集成开发环境,能够支持 ARM、AVR、MSP430 等多种芯片内核平台。

3. IoT Studio

华为 IoT Studio 是华为 LiteOS 提供的一款 Windows 下的图形化开发工具。它以 Visual Studio Code 的社区开源代码为基础,根据 C 语言编程特点和华为 LiteOS 嵌入式系 统软件的业务场景开发。它提供了代码编辑、组件配置、编译、烧录、调试等功能,可以对系 统关键数据进行实时跟踪、保存与回放。

小熊派开发板使用 ST-Link 作为烧录器,因此需要提前下载安装 ST-Link 驱动,根据 PC 的系统选择对应的安装程序,以 64 位安装程序为例,如图 3-3 所示。

名称	修改日期	类型	大小	
📕 amd64	2017/8/3 8:35	文件夹		
📕 x86	2017/8/3 8:35	文件夹		
🚿 dpinst_amd64.exe	2015/5/20 23:42	应用程序	665 KB	
🍣 dpinst_x86.exe	2015/5/20 23:42	应用程序	540 KB	
🚮 stlink_dbg_winusb.inf	2015/5/20 23:42	安装信息	4 KB	
🚮 stlink_VCP.inf	2015/5/20 23:42	安装信息	2 KB	
stlink_winusb_install.bat	2016/9/14 19:14	Windows 批处理	1 KB	
stlinkdbgwinusb_x64.cat	2015/5/20 23:42	安全目录	11 KB	
🗣 stlinkdbgwinusb_x86.cat	2015/5/20 23:42	安全目录	11 KB	
stlinkvcp_x64.cat	2015/5/20 23:42	安全目录	9 KB	
stlinkvcp_x86.cat	2015/5/20 23:42	安全目录	9 KB	

图 3-3 开发板安装包

安装完驱动后,将开发板通过 Micro-USB 线与 PC 连接,打开 Windows 的设备管理器,如果在设备管理器中能找到 ST 端口,证明驱动安装成功,如图 3-4 所示。

▲ 设备管理器	-	X
文件(F) 操作(A) 查看(V) 帮助(H)		
desktop-ulp772 desktop-ulp772		
Sound, video and game controllers		- 1
> 🔐 安全设备		- 1
> 🦉 便携设备		- 1
> 🔲 处理器		- 1
🔉 🔜 磁盘驱动器		- 1
> 🎥 存储控制器		- 1
> 🖻 打印队列		- 1
> 🗃 电池		- 1
∽ 💭 端口 (COM和 LPT)		- 1
ELTIMA Virtual Serial Port (COM5->COM6)		- 1
ELTIMA Virtual Serial Port (COM6->COM5)		- 1
STMicroelectronics STLink Virtual COM Port (COM3)		- 1
> 置 固件		- 1
> 🔄 计算机		- 1
> 🛄 监视器		- 1
		- 1
> 10 蓝牙		- 1
		- 1
		- 1
		- 1
		- 1
		- 1
		- 1
▼ 型用中门心线以田 ■ ST Link Dobug		
Y STELIK DEBUG		
· The Statute of the State of		
		_

图 3-4 驱动安装成功示意图

3.1.3 IoT Studio 的使用

IoT Studio 是华为公司研发的,用于支持 LiteOS 嵌入式系统进行软件开发的工具,提供了代码编辑、编译、烧录及调试等一站式开发功能,支持 C、C++、汇编等多种开发语言,可让开发者快速、高效地进行物联网开发。具体安装步骤如下。

1. 华为 IoT Studio 下载安装

华为 IoT Studio 的下载地址为 https://developer.obs.cn-north-4.myhuaweicloud.com/idea/IoT-Studio.zip。完成安装后会提示是否安装 ST-Link 驱动及 JLINK 调试器,可以选择程序自动安装或后续手动安装。

2. 安装 ST-Link 驱动

小熊派开发板使用 ST-Link 来进行程序读写、烧录等操作,因此需要安装 ST-Link 驱动才能使计算机识别并使用小熊派开发板。

3. 安装 JLINK 调试器

华为 IoT Studio 需要使用 JLINK 调试工具进行链接及调试,需要保证计算机上已成功 安装 JLINK 调试器。

4. 安装 OpenOCD 及 GNU Make

OpenOCD 是小熊派烧录程序,GNU Make 是程序编译程序,可以自行在官网找到对应 自己机器平台及架构的版本,或使用安装工具自动安装,安装工具的默认安装位置为"~/ openSourceTools"。 47

第 3

章

注:安装工具也会自动安装 ST-Link 驱动及 JLINK 调试器。

5. 链接小熊派开发板

使用 Micro-USB 数据线连接小熊派开发板的 Micro-USB 口及计算机的 USB 接口。若 驱动正确安装,在计算机的"设备管理器"中可以成功识别开发板,如图 3-5 所示。



图 3-5 在"设备管理器"中成功识别小熊派开发板

调试程序时,请确保 AT 拨块拨至 AT-MCU 位置,如图 3-6 所示。



图 3-6 WIFI8266 模块

当使用串口终端调试外接模块(如图 3-6 的 WIFI8266 模块)时,可将拨块拨至 AT-PC 位置,因本实验不涉及,在实验中请始终保持拨块位于 AT-MCU。

3.2 实验一 Hello World 初体验

本章实验基于简单的 hello_world 案例,对华为 LiteOS 设备开发初步体验,在实验开始 之前请确保实验环境搭建成功。 (1) 实验目标:通过使用小熊派开发板,完成简单的 Hello World 信息打印功能。

(2) 实验准备:硬件采用小熊派 BearPi IoT 开发板和 WIFI8266 通信模组,软件推荐采用 IoT Studio。

3.2.1 搭建环境

1. 连接硬件

如图 3-7 所示,连接好 WIFI8266 通信模组,注意通信模组标记朝外,不要遮盖住开发 板的 LED 显示屏。如图 3-7 和图 3-8 所示,将开发板右上角的拨块拨至 AT-MCU 位置,使 用可传输数据的 USB 线将开发板与主机相连。



图 3-7 小熊派 BearPi IoT 开发板(左)和 WIFI8266 通信模组(右)



图 3-8 开发板与主机连接

2. 创建项目

基于小熊派开发板创建 LiteOS 的工程项目。如图 3-9 所示,选择"创建工程"命令进入 "新建 IoT 工程"对话框,其中"工程名称"和"工程目录"可以自由选择,但注意不要有空格、中 文等特殊符号,例如本实验使用的硬件平台是 STM32L431_BearPi,实验内容是 Hello World 项 目,故工程名称为 STM32L431_BearPi_HelloNpuers,"SDK 版本"选择 IoT_LINK_1.0.0,"硬件 平台"选择 STM32L431_BearPi,选择"基于示例工程构建"的 hello_world_demo。

```
第3章
```

1BearPI_LiteOS T_LINK_1.0.0		····
T_UNK_1.0.0		*
TM32L431_BearPi		~
▲ 基于云磷模型构建		
	▲ 基于云端模型构建 olate ed_template	▲ 基于云端模型构建 Delate ed_template

图 3-9 "新建 IoT 工程"对话框

注意,在创建项目后,需要对相关设置进行检查和确认,具体步骤为:进入创建好的 LiteOS 工程,选择"文件"→"首选项"命令,查看 Studio 设置是否正确。

3. 管理 SDK

"SDK 管理"页面如图 3-10 所示, SDK 的默认选择为创建项目时选择的 IoT_LINK_ 1.0.0,其中路径为程序自动从计算机上读取 SDK 的安装路径,不需要手动设置。

tudio设置					
	SDK路径:	C:\Users\ZSH\.icode\sc	lk		
のの日理	*	版本名称	版本号	状态	
T目磁路22	 Image: A start of the start of	IoT_LINK_1.0.0	1.0.0	更新	
网络配带					
版本更新					
代码编辑					
LAN TANAT					
					and compare
					安装/更新

图 3-10 "SDK 管理"页面

在"SDK 管理"页面中查看 SDK 版本是否需要更新,"状态"栏中若出现"更新",则选中 需更新的版本。单击右下角的"安装/更新"按钮,安装完后会显示"已安装"。更新后关闭当 前工程,然后基于新的 SDK 创建新的工程。

4. 依赖工具

如图 3-11 所示,在"依赖工具"页面可以查看依赖项,单击"JLink 下载"或者"STLink/ V2 驱动下载"可进行依赖下载,这里不需要重复安装。

Studio设置					×
Q (4	衣赖工具:				
SDK管理	资源名称	版本	操作		
依赖工具	▼ 依赖项下载				
工具链路径	JLink下载	V6.34f+	÷.		
	STLink/V2驱动下载	V4.3.0	±.		
版本更新					
代码编辑					
			26621	1923年 1923585855858558585855858558585585585585585	
			WH IV		

图 3-11 "依赖工具"页面

5. 工具链路径

"工具链路径"页面如图 3-12 所示,其中"JLink 目录"和"OpenOCD 路径"会默认选择 计算机上安装的路径,不需要手动设置。

进入创建的项目工程,可以看到最上方的菜单栏(见图 3-13 的方框区域)从左到右依次 为编译、重新编译、停止编译、烧录、重启开发板、启动调试、停止调试。

其中,编译指对当前打开的工程进行编译,并生成编译后的文件。重新编译指删除上一次编译生成的文件,再次执行编译。停止编译即停止正在进行的编译。烧录指将程序烧录 至目标开发板。重启开发板即对开发板进行重启操作,注意在重新启动开发板之后,需要再 次烧录代码。启动调试可启动或继续进行代码调试。停止调试后调试终止,调试信息会清 空,但断点信息会保留。

3.2.2 工程配置

(1)选择菜单栏中的"工程"→"工程配置"命令进入"工程配置"对话框,如图 3-14 所示。进入"串口配置"页面,选择"端口"为 COM3(也有可能为 COM4,与个人计算机有关)、
 "波特率"为 115200。

51 第 3

章

Studio设置			×
Q	工具链路径:		
SDK B理 依赖工具	JLink目录:	C:\Program Files (x86)\SEGGER\JLink_V644h	
工具链路径	OpenOCD路径:	C:\Users\ZSH\openSourceTools\openocd\bin-x64\openocd.exe	
版本更新 代码编辑			
		确认 取消 应用	

图 3-12 "工具链路径"页面



图 3-13 工具栏

工程配置							×
Q	端口:	COM3	~	波特率:	115200	~	
通用							
SDK配置	数据位:	8	~	停止位:	1	~	
编译输出							
目标板	奇偶:	None	~	流控:	None	· · · · ·	
编译器							
烧录器	日志目录:	C:\Users\lan\.icode\logs				自动保存	
调试器							
串口配置							
				确认	取消	应用	

图 3-14 "工程配置"对话框