【项目目标】

能力目标:

- (1) 熟悉 IAR 编译环境。
- (2) 掌握仿真调试器驱动的安装方法。
- (3) 能够对 PxSEL、PxDIR、PxINP 寄存器进行编程。
- (4) 掌握中断使能的步骤。
- (5) 能够通过设置定时器 1、3 寄存器用中断方式来实现灯闪烁。
- (6) 能够串口发送和接收数据。
- (7) 能够使用 CC2530 中的温度传感器采集温度。
- (8) 能够使用 ZigBee CC2530 驱动 LCD12864 液晶屏。

素质目标:通过学习制作智能水杯,学生能够掌握 CC2530 芯片的功能,具备硬件测试的能力,提高实际动手能力。

重点难点:

- (1) 了解常见的 ZigBee 开发的片上系统。
- (2) 熟悉 CC2530 片上系统的结构及内部模块功能。
- (3) 掌握 IAR 编译软件的安装、编辑与修改。
- (4) 掌握通用 I/O 端口的输入输出设置。
- (5) 熟悉 PxSEL、PxDIR、PxINP 寄存器的应用。
- (6) 了解中断的概念。
- (7) 掌握中断屏蔽寄存器、中断标志寄存器的应用。
- (8) 了解定时器的概念和基本功能。
- (9) 掌握定时器 1、3 寄存器的工作原理和应用。
- (10) 熟悉串行通信接口的概念、特点、工作方式。
- (11) 掌握串行通信接口寄存器的具体应用。
- (12) 掌握用 CC2530 中的温度传感器采集温度的方法。
- (13) 掌握用 ZigBee CC2530 驱动 LCD12864 液晶屏的方法。

【知识纲要】

CC2530 片上系统、IAR、Smart Flash、PxSEL/PxDIR/PxINP 寄存器、中断屏蔽寄存器、 中断标志寄存器、定时器 1 和 3 寄存器、串行通信接口、温度传感器、液晶显示。

【项目分析】

许多家庭给新出生的宝宝冲奶粉的时候总是掌握不好水的温度,不同牌子的奶粉有不同的配比说明,但一般来说都是用 40℃~60℃的温水冲比较好。这个温度不仅有利于加快化学反应的速度,促使糖、奶粉等在液体里的溶解,调出比较均匀的溶液,且能保证奶粉里的营养物质不被破坏。冲好牛奶后,等温度降到约 37℃时再给宝宝喝。可是爸爸妈妈们如何掌握好水温呢?

本项目就来设计一款智能水杯,主要针对有婴儿或者老人的家庭不能直观地感知水温 的问题,结合传感器等技术实现。

该智能水杯使用温度传感器读取被测温度值,系统采用 CC2530 片上系统,显示部分采用 LCD 数码管动态扫描方式显示温度。涉及的知识点主要有 CC2530 片上系统的结构及内部模块功能,IAR 编译软件的安装、编辑与修改。为了掌握 CC2530 的使用,为今后的学习 打好基础,在本项目中我们做了以下几个实验:通过 CC2530 片上系统 I/O 端口控制来实现 点亮 LED 和按键控制 LED 交替闪烁的操作;通过 CC2530 的按键控制灯亮灭来实现外设中断的操作;通过 CC2530 的不同定时器实现传统的点灯操作;通过 CC2530 的串行通信接口 来测试数据的发送和接收。最后使用集成在芯片 CC2530 中的温度传感器采集到内部温度传感器信息通过串口发送到上位机,并用 ZigBee CC2530 来驱动 LCD12864 液晶屏,使用 LCD 液晶显示屏来显示温度的数值。

任务1 智能水杯开发平台

1.1.1 任务描述

在各种各样制作智能水杯的技术解决方案中,我们需要选择一款合适的开发平台,需要有支持 ZigBee 协议栈的硬件应用到 ZigBee 无线传感器网络中,所以该智能开发采用 TI 公司的 CC2530 无线片上系统。

1.1.2 任务目标

- (1) 了解 ZigBee 的概念和特点。
- (2) 了解常见的 ZigBee 开发的片上系统。
- (3) 熟悉 CC2530 片上系统的结构及内部模块功能。
- (4) 掌握 IAR 编译软件的安装、编辑与修改方法。

1.1.3 知识学习

1. ZigBee 标准概述

ZigBee 标准是一种新兴的短距离无线网络通信技术,基于 IEEE 802.15.4 协议栈,主要 面向低速率无线个人区域网(Low Rate Wireless Personal Area Network, LRWPAN)。它功耗

低,典型特征是近距离、低功耗、低成本、低传输速率,主要适用于工业监控、远程控制、 传感网络、家庭监控、安全系统和玩具等领域,目的是满足小型廉价设备的无线联网和控 制需要。ZigBee 技术采用三种频段: 2.4GHz、868MHz 和 915MHz。2.4GHz 频段是全球通 用频段,868MHz 和 915MHz 是用于美国和欧洲的 ISM 频段,这两个频段的引入避免了 2.4GHz 附近各种无线通信设备的相互干扰。

2. 常见 ZigBee 开发的片上系统

随着集成电路技术的发展,无线射频芯片厂商采用片上系统(System On Chip, SOC)的办法对高频电路进行了大量集成,大大简化了无线射频应用程序的开发。ZigBee 开发板(如图 1-1 所示)旨在开发、演示各种 ZigBee 相关产品应用,支持 CC2430、CC2431、CC2530(如图 1-2 所示)等芯片的 ZigBee 开发,但并不局限于此,板上丰富的硬件资源允许用户评估、开发、演示其他类型的射频产品等。



图 1-1 CC2530 芯片



图 1-2 基于 CC2530 的 ZigBee 开发板

目前比较通用的 ZigBee 基本是 TI 公司的 2.4GHz 的芯片,如 CC2420、CC2430 和 CC2431,其中 CC2431 有定位功能,CC2430 和 CC2431 用起来都比较方便。最具代表性的 是 TI 公司开发的 2.4GHz IEEE 802.15.4 ZigBee 片上系统解决方案 CC2530 无线单片机。TI 公司提供完整的技术手册、开发文档、工具软件,使得开发者开发无线传感网络应用成为 可能。TI 公司不仅提供实现 ZigBee 网络的无线单片机,而且免费提供符合 ZigBee 2007 协议规范的协议栈 Z-Stack 和较为完整的开发文档。

3. CC2530 片上系统的结构及内部模块功能

CC2530 是用于 2.4GHz IEEE 802.15.4、ZigBee 和 RF4CE 应用的一个真正的片上系统 (SOC)解决方案。它能够以非常低的总的材料成本建立强大的网络节点。CC2530 结合了 领先的 RF 收发器的优良性能、业界标准的增强型 8051 CPU、系统内可编程闪存、8KB RAM 和许多其他强大的功能。CC2530 有四种不同的闪存版本: CC2530F32、CC2530F64、 CC2530F128、CC2530F256,分别具有 32KB、64KB、128KB、256KB 的闪存。CC2530 具 有不同的运行模式,使得它尤其适应超低功耗要求的系统。运行模式之间的转换时间短进

一步确保了低能源消耗。

(1) CC2530 的主要功能特性。

- CPU 配置: 片内集成增强型高速 8051 内核,支持代码预取; 256KB Flash 程序存储器,支持最新版 ZigBee 2007 Pro 协议; 8KB 数据存储器;支持硬件调试。
- 电源供电:支持 2.0~3.6V 供电区间,具有 3 种电源管理模式:唤醒模式、睡眠模式和中断模式,包括处理器和智能片内外设在内的模块超低功耗。
- 片内外设:片内集成 5 通道 DMA; MAC 定时器;4 个振荡器用于系统时钟和定时操作;1 个通用 16 位和 2 个 8 位定时器;1 个红外发生电路;32kHz 睡眠定时器;电源管理与片内温度传感器;8 通道 12 位 A/D 转换器;看门狗等智能外设。高密度集成化电路节约成本。
- 中断:片内中断控制器控制18个中断源,4优先级结构。
- 接口方面:有2个可编程串口,用于主从 SPI 或 UART 操作; I/O 控制器控制 21 个 I/O 接口,可以根据需要灵活配置,其中2个具有20mA 的电流吸收或电流供给能力。
- 通信方面:信息帧同步字检测/插入;数据 CRC-16 校验;信号强度检测/数字 RSSI; 硬件支持避免冲突的载波侦听多路存取(CSMA-CA)。
- 数据加密:片内集成1个AES协处理器,以支持IEEE 802.15.4 MAC 层安全所需 的加密操作。
- 应用对象:应用范围包括 2.4GHz IEEE 802.15.4 系统、RF4CE 远程控制系统、ZigBee 网络、家居自动化、照明系统、工业测控、低功耗 WSN 等领域。

(2) CC2530 结构分析。

如图 1-3 所示是 CC2530 的功能模块图。这些模块大致可以分为三类: CPU 和内存相关的模块;外设、时钟和电源管理相关的模块,以及无线电相关的模块。

(3) 功能模块介绍。

1) CPU 和内存。

CC2530 芯片系列中使用的 8051CPU 内核是一个单周期的 8051 兼容内核。它有三种不同的内存访问总线:单周期访问 SFR、DATA 和主 SRAM。它还包括一个调试接口和一个输入扩展中断单元。

中断控制器: 总共提供 18 个中断源, 分为 6 个中断组, 每个与 4 个中断优先级之一相关。当设备从活动模式回到空闲模式时, 任一中断服务请求就被激发, 甚至还可以通过一些中断把设备从睡眠模式中唤醒。

32/64/128/256KB 闪存块:为设备提供了内电路可编程的非易失性程序存储器,映射到 XDATA 存储空间。除了保存程序代码和常量以外,非易失性存储器允许应用程序保存必须 保留的数据,这样设备重启之后可以使用这些数据。例如可以利用已经保存的网络具体数 据,就不需要经过完全启动、网络寻找和加入过程。

2) 外设: CC2530 包括许多不同的外设,允许应用程序设计者开发先进的应用。

 调试接口:执行一个专有的两线串行接口,用于内电路调试。通过这个调试接口, 可以执行整个闪存存储器的擦除、控制使能哪个振荡器、停止和开始执行用户程 序、执行 8051 内核提供的指令、设置代码断点,以及内核中全部指令的单步调试。 使用这些技术,可以很好地执行内电路的调试和外部闪存的编程。



图 1-3 ZigBee 功能模块图

- I/O 控制器:负责所有通用 I/O 引脚。CPU 可以配置外设模块是否控制某个引脚或 它们是否受软件控制,如果是的话,每个引脚配置为一个输入还是输出,是否连 接衬垫里的一个上拉或下拉电阻。CPU 中断可以分别在每个引脚上使能。每个连 接到 I/O 引脚的外设可以在两个不同的 I/O 引脚位置之间选择,以确保在不同应用 程序中的灵活性。
- 五通道 DMA 控制器:系统可以使用一个多功能的五通道 DMA 控制器,使用 XDATA 存储空间访问存储器,因此能够访问所有物理存储器。每个通道(触发器、 优先级、传输模式、寻址模式、源和目标指针、传输计数)可以用 DMA 描述符 在存储器的任何地方配置。许多硬件外设(AES 内核、闪存控制器、USART、定 时器、ADC 接口)通过使用 DMA 控制器在 SFR 或 XREG 地址和闪存、SRAM 之间进行数据传输,获得高效率操作。
- 定时器 1: 是一个 16 位定时器,具有定时器和 PWM 功能。它有一个可编程的分频器,一个 16 位周期值,和 5 个各自可编程的计数器/捕获通道,每个都有一个 16 位比较值。每个计数器/捕获通道可以用作一个 PWM 输出或捕获输入信号边沿

的时序。它还可以配置在 IR 产生模式,计算定时器 3 周期,输出是 ANDed,定时器 3 的输出是用最小的 CPU 互动产生调制的消费型 IR 信号。

- MAC 定时器 (定时器 2): 专门为支持 IEEE 802.15.4 MAC 或软件中其他时槽的协议设计。定时器有一个可配置的定时器周期、一个 8 位溢出计数器(可以用于保持跟踪已经经过的周期数)、一个 16 位捕获寄存器(用于记录收到/发送一个帧开始界定符的精确时间或传输结束的精确时间)和一个 16 位输出比较寄存器(可以在具体时间产生不同的选通命令(开始 RX、开始 TX 等)到无线模块)。
- 定时器 3 和定时器 4: 是 8 位定时器,具有定时器、计数器和 PWM 功能。它们有一个可编程的分频器,一个 8 位的周期值,一个可编程的计数器通道,具有一个 8 位的比较值。每个计数器通道可以用作一个 PWM 输出。
- 睡眠定时器:是一个超低功耗的定时器,计算 32kHz 晶振或 32kHz RC 振荡器的 周期。睡眠定时器在除了供电模式 3 的所有工作模式下不断运行。这一定时器的 典型应用是作为实时计数器或作为一个唤醒定时器跳出供电模式 1 或 2。
- 看门狗:一个内置的看门狗允许 CC2530 在固件挂起的情况下复位自身。当看门狗 定时器由软件使用时,它必须定期清除,否则当它超时就会复位设备。

3)无线设备: CC2530 具有一个 IEEE 802.15.4 兼容无线收发器。RF 内核控制模拟无 线模块。另外,它为 MCU 和无线设备之间提供了一个接口,可以发出命令、读取状态、自 动操作和确定无线设备事件的顺序。无线设备还包括一个数据包过滤和地址识别模块。

(4) 引脚描述。

CC2530 的引脚排布如图 1-4 所示,对各个引脚的简单描述如表 1-1 所示。



引脚名称	描述
AVDD1	2V~3.6V 模拟电源连接
AVDD2	2V~3.6V 模拟电源连接
AVDD3	2V~3.6V 模拟电源连接
AVDD4	2V~3.6V 模拟电源连接
AVDD5	2V~3.6V 模拟电源连接
AVDD6	2V~3.6V 模拟电源连接
DCOUPL	1.8V 数字电源去耦
DVDD1	2V~3.6V 数字电源连接
DVDD2	2V~3.6V 数字电源连接
GND	接地衬垫必须连接到一个坚固的接地面
GND	连接到 GND
P0_0	端口 0.0
P0_1	端口 0.1
P0_2	端口 0.2
P0_3	端口 0.3
P0_4	端口 0.4
P0_5	端口 0.5
P0_6	端口 0.6
P0_7	端口 0.7
P1_0	端口 1.0(20mA 驱动能力)
P1_1	端口 1.1(20mA 驱动能力)
P1_2	前口 1.2
P1_3	前口 1.3
P1_4	端口 1.4
P1_5	端口 1.5
P1_6	端口 1.6
P1_7	端口 1.7
P2_0	端口 2.0
P2_1	靖山口 2.1
P2_2	前口 2.2
P2_3/	端口 2.3/32.768kHz XOSC
P2_4/	端口 2.4/32.768kHz XOSC
RF_N	RX 期间负 RF 输入信号到 LNA
RF P	RX 期间正 RF 输入信号到 LNA

表 1-1 引脚描述

引脚名称	描述
RBIAS	参考电流的外部精密偏置电阻
RESET_N	复位,活动到低电平
XOSC_Q1	32 MHz 晶振引脚 1 或外部时钟输入
XOSC_Q2	32 MHz 晶振引脚 2

(5) CPU介绍。

CC2530 用一个增强型工业标准的 8 位 8051 微控制器内核代替了传统的标准 8051 微控 制器,提高 MCU 处理能力,以期更好地满足在执行协议栈、网络和应用软件时的要求,增 强型 8051 的运行时钟为 32MHz,但是使用的依然是标准 8051 指令集。原因在于:第一, 因为每个时钟周期为一个机器周期,而标准 8051 中是 12 个时钟周期为一个机器周期,这 可以大大提高机器周期的可利用性;第二,除去被浪费掉的总线状态的方式外,使用标准 8051 指令集的 CC2530 增强型 8051 内核比标准 8051 内核的性能强出 7 倍,由于指令周期 在可能的情况下包含了取指令操作所需的时间,故绝大多数单字节指令均可在一个时钟周 期内完成。除了速度的提高外,CC2530 增加内核还有两处改进:增加第二个数据指针;扩 展了 18 个中断源。CC2530 的 8051 内核的目标码与标准 8051 完全兼容,也可以使用标准 8051 的汇编器和编译器进行软件开发,所有 CC2530 的 8051 指令在目标码和功能上与同类 的标准 8051 产品完全等价。但是,由于 CC2530 的 8051 内核使用不同于标准的指令时钟,因此在编程时与标准 8051 的代码略有差异,原因在于外设如定时器的配置等方面不同于标 准 8051。

4. 相关硬件设计

(1) 主电路设计。由于 CC2530 内部已经建成大量必要的电路,因此其外围只需较少的电路即可完成功能实现。

1) 外围电路设计。外围电路设计的原理图如图 1-5 所示,所用参数已标注在图上。

电容 C11~C17 和电感 L2~L4 组成了电路中的非平衡变压器以满足 RF 输出/输入匹配 电阻 E1(50Ω)的要求。C21 和 C22 连接在约 32MHz 的晶振电路上; C2 为内部数字稳压 器的去耦电容,用于电源滤波来提高芯片工作的稳定性; R3 和 R4 为偏置电阻,主要用来 为 32MHz 的晶振提供合适的工作电流。

2) 电源电路设计。电源电路设计的原理图如图 1-6 所示,所用参数已标注在图上。

通过此电路可以由外接电源为模块供电。工作电压为 2.6V~3.6V,额定为 3.3V。

3) 串口电路设计。串口电路设计的原理图如图 1-7 所示,所用参数已标注在图上。

串口电路的设计在于能使模块与 PC 机进行通信。该系统的串口可以通过 RS232 连接 PC 机, PC 机就通过串口向模块发送控制信息,而模块向 PC 通过串口反馈自己的状态。同时,也可以利用 PC 机通过串口向单片机发送各种数据,再通过模块的协调器节点向外广播。11 针串口电路需要 3.3V 供电,与板子上的供电电源匹配。



(2) PCB 的设计。在画各元器件的封装时尽量选用最小的贴片封装,以减小芯片板的面积。进行 PCB 的排版,板子大小长度为 3.6cm×2.6 cm,完成后的 PCB 如图 1-8 所示。



图 1-8 PCB

5. IAR 编译软件的安装

(1) IAR 简介。IAR Systems 是全球领先的嵌入式系统开发工具和服务供应商。公司 成立于 1983 年,提供的产品和服务涉及嵌入式系统的设计、开发和测试的每一个阶段,包括:带有 C/C++编译器和调试器的集成开发环境(IDE)、实时操作系统和中间件、开发套件、硬件仿真器和状态机建模工具。嵌入式 IAR Embedded Workbench 是一个非常有效的集成开发环境(IDE),它使用户可以充分有效地开发并管理嵌入式应用工程。

(2)IAR7.51 的安装。在网上下载 IAR7.51.rar, 解压。在 ew8051 文件夹中单击 Setup.exe 进行安装,如图 1-9 所示。在图 1-10 所示的界面中单击 Next 按钮进入图 1-11 所示的界面,单击 Accept 按钮进入图 1-12 所示的界面。此时在 IAR7.51 文件夹下双击 key 文件夹下的 keygen.exe 打开注册码程序,如图 1-13 所示,将图 1-13 中 licence number 下的数字序列拷 贝替换如图 1-12 所示的 Licence#后面的数字序列,单击 Next 按钮进入图 1-14 所示的界面。将图 1-13 中 Licence number 下的序列拷贝到图 1-14 中的 Licence key 中,单击 Next 按钮进入图 1-15 所示的界面。一直单击 Next 按钮直到进入安装进度界面,如图 1-16 所示。安装 进度满后进入图 1-17 所示界面,单击 Finsh 按钮完成安装。



AR Systems Software License Agreement	Enter User II
Press the PAGE DOWN key to see the rest of the agreement.	Enter your na
SOFTWARE LICENSE AGREEMENT [concerning Products: IAR Embedded Workbench(R), visualSTATE(R) and IAR PowerPac[TM]) from IAR Systems AB	N <u>a</u> me: [
PREAMBLE THIS SOFTWARE LICENSE AGREEMENT (THE "AGREEMENT") APPLIES TO PRODUCTS LICENSED BY IAR SYSTEMS AB INCLUDING ANY OF ITS SUBSIDIARIES (TAR SYSTEMS") OR DISTRIBUTORS ("AR DISTRIBUTORS"), TO	Company:
Do you accept all the terms of this License Agreement? If you don't accept, the installat cancelled. To install IAR Embedded Workbench for MCS-51, you must have read, understood, and accepted this agreement.	ion is
allShield	InstallShield

图 1-11 安装协议









图 1-13 输入 key

图 1-14 输入注册码



图 1-15 选择安装路径

图 1-16 安装进度



图 1-17 确认安装完成

安装完 IAR7.51 后在"开始"→"所有程序"中打开 IAR,如图 1-18 所示。

📾 IAR Systems	> 📵 IAR Systems License Activation
💼 卡巴斯基反病毒软件 7.0	🔸 🛷 IAR Systems License Manager
😝 千千静听	🚎 IAR Embedded Workbench for MCS-51 7.51A 🔹 🔀 IAR Embedded Workbench
🧾 Microsoft Firewall Client 管理	📷 IAR Embedded Workbench for MCS-51 Evaluation 🔸 🎇 IAR Embedded Workbench Example project
🚗 UT_TECU Coftware	TAP Draduat Information Lag Dila
	图 1-18 打开软件 IAR

6. IAR 工程和项目的建立

(1)新建第一个 IAR 工程。

现在可以通过桌面上的快捷方式或在"开始"菜单中选择程序来启动 IAR 软件开发环 境了。

使用 IAR 开发环境应先建立一个新的工作区。在一个工作区中可以创建一个或多个工 程。用户打开 IAR Embedded Workbench 时已经建好了一个工作区,一般会显示如图 1-19 所示的窗口,可以选择打开最近使用的工作区或向当前工作区添加新的工程。

选择 File→New→Workspace 命令,现在用户已经建好一个工作区,可以创建新的工程 并把它放入工作区。

🔏 IAR Embedded Workbench IDE Embedded Workbench Startup X <u>F</u>ile <u>E</u>dit <u>View</u> <u>Project</u> <u>T</u>ools <u>W</u>indow <u>H</u>elp Add Files. 0 🗳 日 🕼 Create new project in current workspace 1 Add Group .. Import File List ... D Add existing project to current workspace Edit Configurations. Files Remove 4 Open existing workspace Create New Project. A Add Existing Project. Example workspaces Options. 41++F7 Source Code Control ۲ Recent workspaces: workspace_cc2430 compiletest Make F7 Open Compile Ctrl+F7 Rebuild All Clean Batch build. F8 Stop Build Debug Ctrl+D Do not show this window at startup. Make & Restart Debugger Cancel

单击 Project 菜单,选择 Create New Project 命令,如图 1-20 所示。

图 1-20 建立一个新工程

弹出如图 1-21 所示的"建立新工程"对话框,确认 Tool chain 栏已经选择 8051,在 Project templates 栏中选择 Empty project,单击 OK 按钮。

Create New Project		
<u>I</u> ool chain: <mark>8051 Project templates:</mark>	Ţ	
Empty project		~
Description: Creates an empty project.	ОК	Cancel

图 1-21 选择工程类型

根据需要选择工程保存的位置,更改工程名,如 ledtest,单击"保存"按钮,如图 1-22 所示,这样便建立了一个空的工程,如图 1-23 所示。

系统产生两个创建配置:调试和发布。在这里我们只使用 Debug 即调试。项目名称后的 星号(*)指示修改还没有保存。选择 File→Save→Workspace 命令保存工作区文件并指明存

图 1-19 打开一个工作区

另存为 ? 🗙 保存在 (I): 🔁 8051 - 🖛 🗈 💣 📰 bin Ì config doc 我最近的文档 doc drivers inc lib plugins 桌面 src tutor 我的文档 **夏**夏 我的电脑 网上邻居 ledtest 文件名(M): -保存(S) Project Files (*. ewp) • 保存类型(T): 取消

🔀 IAR Embedded Workbench IDE <u>F</u>ile <u>E</u>dit <u>V</u>iew <u>P</u>roject <u>S</u>imulator <u>T</u>ools <u>W</u>in D 🛩 🖬 🕼 🚳 👗 🖻 🖻 🖉 🕫 -Debu 2:: B: Files 🔊 ledtest - Debug * ¥

图 1-22 保存工程

图 1-23 工作区窗口中的工程

Save Vorksp	ace As			? 🗙
保存在 (1): 表最近的文档 受 東面 教的文档 受 教的文档 の文档 の文档 の文档 の文档 の文档 の文档 の文档 の	Din bin	<u>.</u>	← Ē 💣 ⊞•	
	文件名 @): 保存类型 (I):	Workspace Files (*. eww)	• [呆存 (S) 取消

放路径,这里把它放到新建的工程目录下,单击"保存"按钮保存工作区,如图 1-24 所示。

图 1-24 保存工作区

(2) 添加文件或新建程序文件。

选择菜单 Project→Add File,或者在工作区中右击工程名,在弹出的快捷菜单中选择 Add File, 弹出文件打开对话框, 选择需要的文件并单击"打开"按钮。

如果没有建好的程序文件,也可以选择菜单 File→New→File 新建一个空文本文件,向 文件里添加程序清单1中的代码。

程序清单1如下:

#include "ioCC2530.h" void Delay(unsigned char n) £ unsigned char i; unsignedint j;

```
for(i=0; i<n;i++)
  for(j=1;j;j++);
}
void main(void)
ł
  // CC2530 中, I/O 口作普通 I/O 使用时和每个 I/O 端口相关的寄存器有 3 个, 分别是 PxSEL
  //功能选择寄存器、PxDIR 方向寄存器、PxINP 输入模式寄存器,其中 x为 0、1、2,这里选
  //择 P1.0 上的红色 LED 作为 I/O 测试
  SLEEP &= ~0x04;
  while(!(SLEEP & 0x40));
                             //晶体振荡器开启且稳定
                             //选择1~32MHz晶体振荡器
  CLKCON &= ~0x47;
  SLEEP \models 0x04;
  P1SEL = 0x00;
                             //P1.0 为普通 I/O 口
  P1DIR = 0x01;
                             //P1.0 输出
  while(1)
  {
     P1_0 = 1;
     Delay(10);
     P1 0 = 0;
     Delay(10)
  }
}
```

选择菜单 File→Save, 弹出"另存为"对话框, 如图 1-25 所示。

任列						
保存在(工)	Source		•	(†) 📥 🛄 -	
最近的文档						
國						
我的文档						
我的自脑						
网上邻居						
		and a		_	-	(現存(の))
	文件名 (11):	ftest. c			<u> </u>	1#17 (2)

图 1-25 保存程序文件

新建一个 source 文件夹,将文件名改为 test.c 后保存到 source 文件夹下。按照前面添加 文件的方法将 test.c 添加到当前工程里,完成的结果如图 1-26 所示。

(3) 设置工程选项参数。

选择 Project→Options 配置与 CC2530 相关的选项。

1) Target 标签。

按图 1-27 所示配置 Target,选择 Code model 和 Data model,以及其他参数。

Lie Euric Ylew Froject Simu D 📽 🖬 🕼 🕼 🕼 🖡 Workspace	<u>1</u> 466 <u>1</u> 66 1 ທ ຕ	×	test.c
Debug		-	#include "ioCC2430.h"
Files	82	107: 10	void Delay(unsigned char
■ ledtest - Debug *			<pre>{ unsigned char i; unsigned int j; for(i = 0; i < n; i+, for(j = 1; j; j+, for(j = 1,</pre>

图 1-26 添加程序文件后的工程

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Texas Instrument: Infineon ROM-Monitor Analog Devices Silabs Simulator	Target Data Pointer Code Device information Device: CC2530 CPU core: Plain Code model Banked Data model Large Number of virtual 8	Bank Output Library Configur: Do not use extended st: Extended stack = 0x002000 Calling convention XDATA stack reentrant Location for constants and s CAM memory RAM memory CDDE memory
---	---	---

图 1-27 配置 Target

单击 Derivative information 栏右边的按钮,选择程序安装位置,如这里是 IAR Systems\Embedded Workbench 4.05 Evaluation Version\8051\config\derivatives\chipcon下的文件 CC2530.i51。

2) Data Pointer 标签。

如图 1-28 所示,选择数据指针数 1 个,16 位。

Antereal Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Chipcon ROM-Monitor Analog Devices Silabs Simulator	Target Data Pointer Number of DPTRs: Size © 15 bit C 24 bit DPTR addresses © Shadowad C Separate Configure	Arithmetic Unit Code Bank Output	<u> </u>
---	--	--------------------------------------	--------------

图 1-28 数据指针选择

3) Stack/Heap 标签。

如图 1-29 所示,改变 XDATA 栈大小到 0x1FF。

General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Chipcon ROM-Monitor Analog Devices Silabs Simulator	Library Configuration Lib Stack sizes IDATA: 0x40 PDATA: 0x80 XDATA: 0x1FF Extended: 0x3FF	Heap Star XDATA: Far: Huge:	oxFF OxFF OxFFF OxFFF

图 1-29 Stack/Heap 设置

单击 Options for node "ledtest"中左边框架内的 Linker 选项,配置相关的选项。 4) Output 标签。

选中 Override default 可以在下面的文本框中更改输出文件名。如果要用 C-SPY 进行调 试,则选中 format 下的 Debug information for C-SPY,如图 1-30 所示。

Category:		Factory Settings
General Options C/C++ Compiler Assembler Custom Build Build Actions Unker Debugger Third-Party Driver Texas Instrument: Infineon ROM-Monitor Analog Devices Silabs Simulator	Output Extra Output #define Output file © Ogerride default (cc530.hex Format © Bebug information for O I with guines corrol I with guines corrol I with L/O esultion Euffored termina C Other Output int Format variagt: Non Module-local Incluse	Diagnostics List Confi Secondary output file: Offone for the selected for SPY mod notput icettra output file el-extended

图 1-30 输出文件设置

5) Config 标签。

如图 1-31 所示, 单击 Linker 栏文本框右边的按钮, 选择正确的连接命令文件, 如表 1-2 所示。

ategory:	Factory Settings
ieneral Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Texas Instruments Infineon ROM-Monitor Analog Devices Silabs Simulator	Output [Extra Output] #define Diagnostics List Confit > Generate extra output file Queput file

图 1-31 选择连接命令文件

表 1-2	Code Model	关系表
-------	------------	-----

Code Model	File
Near	lnk51ew_cc2530.xcl
Banked	lnk51ew_cc2530b.xcl

6) Debugger 标签。

单击 Options for node "SerialApp"中左边框架内的 Debugger 选项,配置相关的选项。 在 Setup 标签内按图 1-32 所示设置。

Category: General Options C/C++ Compiler Assembler Custom Build Build Actions Linker Debugger Third-Party Driver Texas Instruments Infineon ROM-Monitor Analog Devices Silabs Simulator	Factory Setting
	Device Description file v Qveride default \$TOOLKIT_DIR\$\config\devices\Texas Instruments\CC25K

图 1-32 配置调试器

在 Device Description file 中选择 CC2530.ddf 文件,其位置在程序安装文件夹下,例如 C:\ Program Files\IAR Systems\Embedded Workbench 4.05 Evaluation Version\8051\config\derivatives \chipcon。

最后单击 OK 按钮保存设置。

- 1.1.4 任务实施
- 1. 编译、连接、下载

选择 Project→Make 命令或按 F7 键编译和连接工程,如图 1-33 所示。



成功编译工程并且没有错误信息提示后,按照如图 1-34 所示连接硬件系统(注"CC2430 的 ZigBee 模块"部分可以换成网关、扩展板等与仿真器引脚相配的器件)。



选择 IAR 集成开发环境中的菜单 Project→Debug(如图 1-33 所示)或按快捷键 Ctrl+D 进入调试状态,也可以单击工具栏中的按钮进行程序下载,程序下载完成后 IAR 将自动跳转至仿真状态。

2. IAR 工程的编译和修改

(1) 编译项目。在工作区中选择需要编译的文件,选择 Project→Compile 菜单项,或 者单击工具栏上的 Compile 按钮,对源程序进行编译。编译过程中的各种信息将显示在屏 幕下方的 Build 信息窗口中,如图 1-35 所示。

Messages
Fib.c
Done. 0 error(s). 0 warning(s)

图 1-35 Build 信息窗口

出现错误时,可以按照提示进行相应修改。

(2) 连接项目和调试。进行 Debugger 和 J-Link/J-Trace 设置,如图 1-36 和图 1-37 所示。

Vorkspace		×	DW710C-2.d79 DW710C-2.sim marcic
Files	10C-2 - Debug lude	* <u></u>	Com_PC_Send_Bu } for(1=0;1<10+Com_PC_Se {
	ptions for node Category:	*DW710C-2*	Factory Settings
	General Options C/C++ Complex Assembler Custom Build Build Actions Linker Debugger Simulator Angel IAR ROM-monitor J-UnK/J-Trace LMT FTDI Macragor RDI Third-Party Driver	Setup Download Extra Options Flugi Driver Functions Flugi J-Link/J-Trace Function Sgtup sacros Sgtup sacros S	nx -0; dt Lon -0; //f. PC, NT, PC, T, Send -5, -5, -5, -5, -5, -5, -5, -5,
	v710C-2.dep	[JART E Com_PC OKCancel Dreak:

图 1-36 Debugger 设置



图 1-37 J-Link/J-Trace 设置

选择项目文件夹并右击,选择 Make 命令,如图 1-38 所示。



图 1-38 选择 Make 命令连接项目

弹出如图 1-39 所示的界面,之后进行调试操作,用 J-Link 将目标板和计算机连接之后 单击工具栏中的 Debugger 按钮 💭,如图 1-40 和图 1-41 所示。

🄏 IAR Embedded Workber	ich IDE			
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>T</u> o	ols <u>¥</u> indow	Help		
🗅 🚅 🖬 🕼 🖓 🖓	B 00		- 🙆 🌬 👫 👯 🔀 🗶 🚳	
Workspace	,	font.h main.c LCD.c Key.c manage_485_asset_addr.c LunXu	in.c	•
Debug	•			-
Files	2: Di 🔺	if(Meter_Valid_Flag==METER_IS_VALID)		-
DW710C-2 - Debug		{		
La include				
- D linker		//读取一次当前的时间		
Dinkarm flach vol		RTC_Read_Date_Time(&Time_Rtc)	12	
Inkam ran vd		Meter_32_D1_And_Time.Hour	= Time_Rtc.nour;	
		Neter_32_D1_And_line.Hinute	= lime_Rtc.minute;	
	1	Neter_32_D1_And_11me.Second	= lime_Rtc.second;	
		Meter_32_Date.Day	= Tear_Hon_Day.Day;	
		Meter 32 Date Vear	- Year Mon Day Year:	
He la Dspi.c		Necer_32_Date. rear	- Tear_non_bay.Tear;	
		Com BJ Send Buff01 - STADT CHAI	p •	
THE I TRASN.C		com_bo_send_bar[o] - sinati_ond	~	
THE INTOLC	-	ChangeAddrToAA(sMeter 485 Addr)	[0].«Com BJ Send Buf[1]):	
He gpio.c		Con BJ Send Buf[7] = START	CHAR;	
He la key.c		Con BJ Send Buf[8] = 0X01;	-	
		Con BJ Send Buf[9] = 0X02;		
		Con BJ Send Buf[10] = 0X52;		
LunXun.c		Con_BJ_Send_Buf[11] = 0XC3;		
He main.c	*	Con_BJ_Send_Buf[12] = 0;		
- manage_485_a	*	<pre>for(tenp_i = 0;tenp_i<12;tenp</pre>	p_i++)	
Here Depute tool.c	*	{		
Here in the second	*	Com_BJ_Send_Buf[12] -	+= Com_BJ_Send_Buf[temp_i];	
Head_dl.c	*)		
Here in the second seco		Con_BJ_Send_Buf[13] = 0X16;		
I ⊨—⊞ Mi Spi Flash.c	* <u>×</u>	Con_BJ_Send_Buf[14] = 0X16;		
DW710C-2		[fo] [4]		
Messages				File
Building configuration:	W710C-2-	obug		
Lindeting build tree	54471002	endd		
71. inite				
71x_inits				
/ix_it.c				
		10		
Build Debug Log Find in Files				
eady			Errors 0, Yarnings 0	RUM
🛃 开始 🔰 😒 😂 🚳	" XIAR	mbedded Wor 🦉 22. JPG - 画图		EN 🛛 🖞 🔇 🔶 🗭 🗾 9:15
	Concernation of the second			

图 1-39 单击 Make 后界面

🔀 IAR Embedded Workbench IDE			X
<u>File Edit View Project Tools Window</u>	Help		
🗅 😅 🖬 🕼 🐇 🐘 🛍 🗠 🗠	■ ペ > > > > > > > > > > > > > > > > > >	🔞 📴 🛤 🖬 💁 🛐	0
Workspace 🛛 🗙	font.h main.c LCD.c Key.c manage_485_asset_addr.c LunXu	n.c	+ x
Debug ▼ Files ?:: № ∩ Image: State of the state	sk_2(); sk_3(); Task_7(); Task_7(); :Str711_Init() 功能播述:对主芯片进行初始化*/ void) r711_Init(); r711_Init();	// //在谈到无线命令时,参	117 <i>10/2</i>
H H Bisspic H Bicc H Bitshc H Biotc W710C2	从flash 中读出电施表的485 地址、资产编号、以及33 I, Read J=Link Driver I, Read I_Read I_Read	2 <i>快表计的有效标志</i> r[0],6); et_Number[0],6); _Valid_Flag),1);	لر ل
Messages Building configuration: DW710C-2 - D Updating build tree Configuration is up-to-date.	Jebug		
<u></u> <			>
Build Debug Log Find in Files			×
Ready	Errors 0, Warnings 0		NUM //
软件			

图 1-40 项目调试

完成后出现如图 1-42 所示的情形。

至此,可以进行单步调试,也可以单击 Stop Debugger 按钮在目标板上进行测试。



图 1-41 项目调试



图 1-42 调试完成

1.1.5 任务小结

本次任务主要学习了使用 IAR 集成开发环境进行 ZigBee 开发的基本流程,以及工程的 建立、源文件的添加、编译与调试。

【思考题】

- 1. 简述 CC2530 各引脚的功能。
- 2. 建立新的 IAR 工程和项目。

任务 2 测试智能水杯开发平台

1.2.1 任务描述

为了保证智能水杯开发平台的稳定性,在安装调试完 IAR 软件平台后,还需要安装仿 真调试器的驱动并进行调试,进行 IAR 工程源码的烧写。

1.2.2 任务目标

- (1) 掌握仿真调试器驱动的安装方法。
- (2) 掌握 IAR 工程源码的烧写方法。
- (3) 掌握程序仿真调试的方法。

1.2.3 知识学习

编译好 IAR 工程后就可以调试程序了,需要先连接硬件平台才能进行调试,在计算机 与 ZigBee 硬件系统连接前要在计算机上安装仿真器驱动。

1.2.4 任务实施

1. 仿真调试器驱动的安装

安装仿真器前要确认 IAR 已经安装,将仿真器通过开发系统附带的 USB 电缆连接到 PC 机上,系统找到新硬件后提示如图 1-43 所示的对话框,选择"从列表或指定位置安装"单选项,单击"下一步"按钮,然后单击"浏览"按钮,选择驱动文件所在路径,如图 1-44 所示。

找到新的硬件向导	
	欢迎使用找到新硬件向导
	这个向导帮助您安装软件:
10th	Chipcon SRF04EB
	如果您的硬件带有安装 CD 或软盘,请现在将 其插入。
	您期望向导做什么?
a states whites	○ 自动安装软件 (推荐) (L)
	●从列表或指定位置安装(高级)(S)
Harrison and the second	要继续,请单击"下一步"。
	<上一步 ® 下一步 ® > 取消

图 1-43 选择安装驱动的方式