苏州大学

**《嵌入式应用技术》综合实验报告**

实验科目：嵌入式系统及应用 实验得分：

院 别：计算机科学与技术 专 业： 物联网工程

学生姓名： 张蓉 学 号： 1327404017

实验题目： 《嵌入式系统及应用》综合实验

指导教师：　　 曹金华、王宜怀

2015～2016学年第二学期

目 录

[第1章 功能概述与软硬件开发平台 1](#_Toc10459)

[1.1功能概述 1](#_Toc3022)

[1.2 软硬件开发平台 1](#_Toc11262)

[1.2.1 软件开发平台 1](#_Toc19587)

[1.2.2 硬件开发平台 1](#_Toc4903)

[第2章 总体设计 2](#_Toc1104)

[2.1 芯片最小系统及引脚接法 2](#_Toc26758)

[2.2 软件结构图 3](#_Toc16387)

[2.2.1 Target节点端软件结构 3](#_Toc27557)

[2.2.2 PC端软件结构 3](#_Toc22305)

[第3章 软件设计 4](#_Toc31601)

[3.1 概述 4](#_Toc1124)

[3.2 Target节点方软件设计 5](#_Toc18774)

[3.3 PC方软件设计 8](#_Toc22224)

[3.4 设计过程中技术难点分析 11](#_Toc26214)

[3.4.1 ADC转换 11](#_Toc26823)

[3.4.2组帧解帧 12](#_Toc685)

[第4章 综合测试 13](#_Toc29840)

[第5章 心得体会 15](#_Toc14359)

[参考文献 16](#_Toc23073)

[附录 使用说明 17](#_Toc17813)

# 第1章 功能概述与软硬件开发平台

## 1.1功能概述

ARM Cortex-M0+是ARM公司2012年3月14日于中国上海发布了一款拥有全球最低功耗的微控制器（简称Target节点） 。飞思卡尔的Kinetis L系列Target节点正是汲取该新型ARM Cortex-M0+处理器众多优质性能的系列产品，体现出该系列产品的卓越能效和易用性、功耗更低、价格更低、效率更高，且体现了Kinetis的优质性能，即多元化的外设、广泛的支持、易扩展性优质性能，KL25 Target节点正是该系列Target节点的代表产品。本报告描述一个Freescale KL25Target节点实验系统软件实例，可以通过串行通信与PC机连接，实现与PC机的相互控制。

设计时首先设计Target节点与PC机间的标志符号位，用于确定双方通信数据的含义。Target节点端程序编写完成后，使用串行通信调试器观察输出数据，输入控制数据观察Target节点工作状态。使用串行通信调试器调试成功后，再使用C#编写与其匹配的可视化程序。

实现联机验证最小系统、普通I/O口输入输出、驱动LCD\LED显示、串行通信数据接收与发送、定时器计时、PWM、键盘输入、AD转换、FLASH读写模块等功能。

## 1.2 软硬件开发平台

### 1.2.1 软件开发平台

软件开发基于Microsoft Visual Studio 2010编程环境，采用了C#语言进行编程，从而实现了PE机显示结果的界面。

C#是[面向对象](http://baike.baidu.com/view/125370.htm" \t "http://baike.baidu.com/view/_blank)的编程语言。它使得程序员可以快速地编写各种基于MICROSOFT .NET平台的应用程序，MICROSOFT .NET提供了一系列的[工具](http://baike.baidu.com/view/74746.htm" \t "http://baike.baidu.com/view/_blank)和服务来最大程度地开发利用计算与通讯领域。

C#使得C++程序员可以高效的开发程序，且因可调用由 C/C++ 编写的本机原生函数，因此不损失C/C++原有的强大的功能。C#与C/C++具有极大相似性，熟悉类似语言的开发者可以很快的转向C#。

### 1.2.2 硬件开发平台

Kinetis Design Studio是飞思卡尔于 2014 年开始推出的面向 ARM Cortex-M 内核的 Kinetis 系列微控制器的嵌入式集成开发环境。Kinetis Design Studio是一种附送的Kinetis Target节点集成开发环境，为设计提供强大的编辑、编译和调试功能。Kinetis Design Studio IDE基于包括Eclipse、GNU编译器套装(GCC)、GNU调试器(GDB)等免费开源软件，为设计人员提供了一种不限制代码大小的简单开发工具。此外，Processor Expert软件支持您利用其知识库进行设计，只需几次鼠标点击即可创建强大的应用。

# 第2章 总体设计

## 2.1 芯片最小系统及引脚接法

图2-1给出的是56引脚LGA封装的KW01的引脚图。每个引脚都可能有多个复用功能，有的引脚有两个复用功能，有的有四个复用功能，实际嵌入式产品的硬件系统设计时必须注意只能使用其中的一个功能。进行硬件最小系统设计时，一般以引脚的第一功能作为引脚名进行原理图设计，若实际使用的是其另一功能，可以用括号加以标注，这样设计的硬件最小系统就比较通用。

具体连线方式见图2-1。

|  |
| --- |
| 图2-1 最小系统原理图（本实验的连线图） |

## 

## 2.2 软件结构图

软件编写包括Target节点端与PC端2部分组成，下面分别介绍2部分软件的结构。

### 2.2.1 Target节点端软件结构

根据需求分析，对需要测试的功能进行编程，Target节点端包含开关、小灯、AD转换、FLASH读写、无线通信数据发送与接收等模块。

### 2.2.2 PC端软件结构

PC端使用C#编程，将数据以串口通信的方式发送给PCNode，PCNode再通过无线的方式，将数据帧转发给Target节点，从而实现对Target节点的控制和对Target节点反馈的信息进行处理，并反应在用户界面上。

根据Target节点提供的功能，编写程序匹配其工作。PC端程序包括串行通信数据输入输出、开关小灯控制、AD转换数据显示、Time（定时器）、FLASH读写模块。

# 第3章 软件设计

## 3.1 概述

本工程为测试KL25接收发送测试实例,从窗体界面来看分为六个部分，具体如下：

1. PC节点检测：

点击“检测PC节点”，即可检测PC节点和Target节点的无线通信状态。

状态包括：没有找到USB串口，有USB串口但无PC节点，串口号：PCNode（表示检测到PCNode）三种；

1. 数据收发

输入测试数据内容，点击“验证RF收发”按钮，下框中会收到Target节点回送的测试信息，即可验证通信良好。

1. 温度采集

点击“开始采集”，即可采集target的温度值，textbox内显示的是当前的温度值，chart记录温度值的变化曲线，每隔3秒采集一次；点击“停止”，则停止采集温度数据。

1. 小灯控制

PORTB|（1）的gpio口接上小灯，启动时，小灯默认打开，点击“关闭”，可以关闭小灯，点击“打开”，可以打开小灯并且显示小灯目前状态

（5）.flash读写测试

1）设置扇区号，偏移量和读写字节数，点击“target Flash读测试”，可以读到对应扇区、对应偏移位置的数据，数据量为输入的读写字节数；设置扇区号、偏移量和写入数据，点击“target Flash写测试”，可以将所输入的数据写入到指定位置；

2）绝对地址的写入和读取：在对应文本框输入要读取flash位置的绝对地址（十六进制方式），点击“读取绝对地址”按钮，即可读取相应位置的数据。

在绝对地址文本框输入要写入的地址以及数据，点击“写入绝对地址”，则可以将数据写入到对应的位置。

1. 图片读写

点击“打开图片”，选取要发送的图片，左侧图片框和文本框会显示待发送的图片、图片转为字符的结果和图片大小（即索要发送的字节数）。注：所选图片大小不超过5000字节。点击“发送图片”即可将图片发送至terget节点并存入flash；点击“读取图片”，可以从target端的flash中将图片字符串数据读回，显示在右侧文本框，点击“还原图片”，可以将字符串转换成为图片显示出来。

（1/2项是保留的样例工程原有的功能，第5项只添加了绝对地址的转换）

## 3.2 Target节点方软件设计

本节将详细介绍Target节点端的主程序和中断程序，利用各个模块完成本实验的各个功能，其中包括无线通信数据输入输出、DA转换、FLASH读写。

程序基于C语言编写，使用面向硬件对象模块封装的思想。面向硬件对象模块的封装是对系统中的每个硬件对象建立相对应的头文件和硬件驱动程序文件。头文件中主要包含该硬件的接口和相应的寄存器设置参数，硬件驱动程序文件主要是对该硬件的驱动，同时标注出各个功能模块的功能，入口，出口和堆栈深度。所以编程人员可以不用了解模块的编程方法，即可利用这些模块程序。于是剩下的编程工作只是面向总体各模块配合以及PC显示与控制的主程序和中断程序的编程。硬件模块封装屏蔽了具体硬件对象的差异性，提高了程序的移植性和复用性。

各个模块的驱动程序见随书光盘。

Target端程序执行流程：

Target节点相关任务包括：运行指示灯任务task\_light，接收PC节点数据的任务task\_rf\_recv,将数据发回PC节点的任务task\_rf\_send,采集温度的任务task\_adc,小灯控制任务task\_lightctrl

简要流程如下：

1）开机，启动操作系统，完成初始化工作，小灯开始闪烁，温度数据开始采集。

2）Target节点接受到PC节点发来的数据时产生中断，当接收到一个完整的数据帧时，置接收成功事件位并唤醒帧数据接收任务函数

3）帧数据接收任务获接收成功事件位，得对数据进行校验。

校验成功则对帧命令进行解析与处理，并对需要回发数据的帧进行组帧、置RF发送事件位。然后清除RF接收事件位，继续等待RF接收事件位置1

1. 帧数据发送任务函数获得RF发送事件位，将数据回发，然后清除RF发送事件位，继续等待RF发送事件位置1。

每个功能具体流程：

1. 温度采集
2. 在PC端点击“开始采集”按钮，则PC端通过串口将数据帧传给PCNode，PCNode将数据帧再转发给TargetNode，发送的据帧如上表，数据赋值了2个0；
3. Target节点接收到该数据帧，产生接收中断，置lwevent\_group事件组的EVENT\_RF\_RECV事件位
4. 接收任务task\_rf\_recv获得事件位，进行数据校验以及命令解析
5. 采集温度的任务是每隔1秒，执行一次，将采集到的26通道的物理量回归变成摄氏温度数据，赋值给全局变量g\_temperature。
6. 接受任务判断得出接收缓冲区g\_rf\_recvBuf[7]是RF命令字0x02，并且g\_rf\_recvBuf[8]温度采集子命令，则执行RFTest()子函数的相关代码。获取全局变量g\_temperature的值，整数部分放在发送缓冲区的g\_rf\_sentBuf[9]，小数部分乘以10，四舍五入，放在g\_rf\_sentBuf[10]。发送缓冲区除去帧头帧尾，其余均与接收缓冲区值一致。返回要发回的数据长度。接下来设置发送时间位。清接收事件位
7. 数据发送函数获得数据发送事件位，将发送缓冲区的数据发送出去，然后清除数据发送事件位。
8. PC端接收到MCU端发来的数据，将接收到的数据recvData[9]+recvData[10]/10.0,获得温度值，在温度界面的文本框中进行显示，并且在chart控件上增加一个数据。
9. 小灯控制
10. 在PC端发送小灯的控制命令，格式如上，数据位的0表示打开，1表示关闭。
11. Target节点接收到数据帧之后，进行解帧，命令处理函数command\_proc()判断得出接收缓冲区g\_rf\_recvBuf[7]为0x03（即小灯控制命令），则执行LIGHTstate()子函数。
12. 将接收缓冲区g\_rf\_recvBuf赋值给发送缓冲区g\_rf\_sendBuf。将小灯状态赋值给全局变量light\_temp，设置小灯控制事件位，让小灯控制任务被唤醒，设置数据发送事件位，让数据发送任务被唤醒。然后清除数据接收事件位。
13. 小灯控制任务获取小灯控制事件位，将全局变量light\_temp转换成小灯状态，控制小灯状态变成命令要求的状态。清除小灯控制事件位。
14. 数据发送函数获取数据发送事件位，将数据发送缓冲区的数据发送出去，清除数据发送事件位。
15. PC端获取target发回的数据，判断命令字，为小灯控制命令字，然后获取数据位的状态，在小灯控制界面的而文本框显示小灯状态，当数据位为0，显示：小灯已经打开，数据位为1，显示：小灯已经关闭。
16. Flash图片的读写
17. 在PC端发先送一个信息帧，帧格式如上。将图片的字节数和帧数发送出去。再将图片的数据 部分发出去，最后一帧的长度不定，单独用一个’F’来标注。
18. Target端接收到PC端传来的数据，产生接收中断。数据接收任务对数据进行处理。命令处理子 函数command\_proc()判断出g\_rf\_recvBuf[7]为0x06（即写入Flash图片控制命令），则到FlashWritePic() 子函数进行处理。
19. 当接收到信息帧，则将记录已经接收到的帧数的全局变量RevZhenCount清0，将图片字节数赋 值给Pic\_Arr前两位，将即将发送的帧数赋值给全局变量PicZhenNum，以便之后判断帧数是否收全。

当接收到数据帧和最后一帧，则将数据依次赋值给Pic\_Arr。

1. 判断，当帧数齐全，或者受到最后一帧，则设置图片写入事件位，唤醒图片写入任务。
2. 图片写入任务通过图片缓冲区Pic\_Arr的前两位计算出所需要存储的字节数，从而计算出所需 要擦除的扇区数。然后擦除对应扇区，将表示图片字节数的两个字节写入第99扇区的偏移量为1001 和1002的两个字节。之后，Pic\_Arr每1000字节写入到99扇区开始的偏移量为0开始的1000个字 节位置。

写入完毕，清除图片写入事件位。

至此，图片写入过程完毕。

1. 在PC端送图片读取命令。
2. 第一帧，读取第99扇区的偏移量为1000开始的两个字节。帧格式如表格所示。

Target端接收到数据帧，判断为图片读取命令，则获取扇区，偏移量，以及读取字节数，将对应位置 的数据读取出来。

1. 之后PC端依次发送读取后面40字节的命令帧，直到最后一帧。
2. 读取数据完毕，在Flash图片存取界面，显示图片的字符表示在文本框中。点击图片还原，如 果没有丢帧和数据损坏，可以还原出原来的图片。

相关函数具体代码见工程；

各个模块的命令发送被包装成数据帧，以帧头+帧长+命令类型+数据+帧尾为格式，Target节点端接收到会以命令类型予以区分。接收到来自Target节点的数据，同样也是包装成的数据帧，串口通信以字节为单位传输，先将接收到的数据组帧，根据命令类型判断是哪个模块，在相应区域加以显示。

## 3.3 PC方软件设计

PC端程序以串行SCI为核心，用户界面上划分多个区域，实现不同控制功能。PC端用户界面如图

（1）PC节点检测



点击“检测PC节点”，即可检测PC节点和Target节点的无线通信状态。

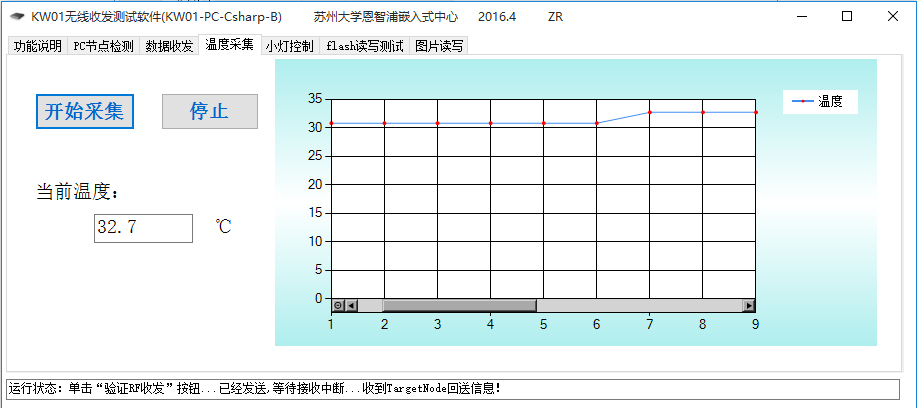
状态包括：没有找到USB串口，有USB串口但无PC节点，串口号：PCNode（表示检测到PCNode） 三种；

（2）数据收发



输入测试数据内容，点击“验证RF收发”按钮，下框中会收到Target节点回送的测试信息，即可验证通信良好。

（3）温度采集



点击“开始采集”，即可采集target的温度值，textbox内显示的是当前的温度值，chart记录温度值的变化曲线，每隔3秒采集一次；点击“停止”，则停止采集温度数据。

（4）小灯控制



PORTB|（1）的gpio口接上小灯，启动时，小灯默认打开，点击“关闭”，可以关闭小灯，点击“打开”，可以打开小灯并且显示小灯目前状态

（5）flash读写测试





设置扇区号，偏移量和读写字节数，点击“target Flash读测试”，可以读到对应扇区、对应偏移位置的数据，数据量为输入的读写字节数；设置扇区号、偏移量和写入数据，点击“target Flash写测试”，可以将所输入的数据写入到指定位置；

绝对地址的写入和读取：在对应文本框输入要读取flash位置的绝对地址（十六进制方式），点击“读取绝对地址”按钮，即可读取相应位置的数据。

（6）图片读写







点击“打开图片”，选取要发送的图片，左侧图片框和文本框会显示待发送的图片、图片转为字符的结果和图片大小（即索要发送的字节数）。注：所选图片大小不超过5000字节。点击“发送图片”即可将图片发送至terget节点并存入flash；点击“读取图片”，可以从target端的flash中将图片字符串数据读回，显示在右侧文本框，点击“还原图片”，可以将字符串转换成为图片显示出来。

## 3.4 设计过程中技术难点分析

### 3.4.1 ADC转换

KW01芯片自身就有温度传感器，可以通过引脚配置，采集传感器监测到的温度，但是此时监测到的温度并不能用摄氏度进行显示，得到的数据只是原始数据，无法与真实的温度联系在一起，因此我们需要采用回归的方法，将采集到的数据转换为我们所广泛认知的摄氏温度，从而显示在PC界面中，因此，我们采用了ADC转换，

### 3.4.2组帧解帧

#### 1．帧协议的概念

在该工程搭建过程中，其中一个难点就是组帧与解帧。各个任务分别对应于各自的模块，PC发送给某个模块的数据以及Target节点会发给PC 的数据，均可通过增加命令字和子命令字段来区分。同时为了确保数据的完整性，可增加帧头、帧尾来组成一个完整的帧数据，供串口组帧、解帧时使用。

帧协议是Target节点方与PC方数据双向传输的通信约定，主要包括帧格式和帧命令两个部分。帧格式给出了帧数据的各个字段含义以及各个字段的长度。帧命令给出了各个模块苏实现功能的命令编码。

#### 2．帧格式

约定的帧格式如下所示

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 帧头 | 帧长 | 硬件地址 | 软件地址 | 命令 | 子命令 | 数据 | 帧尾 |
| 字节数 | 1B | （n+7）B | 1B | 4B | 1B | 1B | nB | 1B |
| 字节数 | 1B | （n+6）B | 1B | 4B | 1B | 0B | nB | 1B |

具体各模块数据帧如下：

1. 温度采集

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 帧头 | 帧长 | 硬件地址 | 软件地址 | 命令 | 子命令 | 数据 | 帧尾 |
| 字节数 | 1B | 1B | 1B | 4B | 1B | 1B | 2B | 1B |
| PC端 | P | 9 | 0xff |  | 0x02 | T | 0,0 | C |
| Target | T | 9 | 0xff |  | 0x02 | T | 温度整数，小数 | D |

PC端和Target节点所发数据帧格式如上：0x02是RF命令字，T是温度采集子命令

1. 小灯控制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 帧头 | 帧长 | 硬件地址 | 软件地址 | 命令 | 子命令 | 数据 | 帧尾 |
| 字节数 | 1B | 1B | 1B | 4B | 1B | 0B | 1B | 1B |
| PC端 | P | 7 | 0xff |  | 0x03 |  | 0-打开,1-关闭 | C |
| Target | T | 7 | 0xff |  | 0x03 |  | 0-打开,1-关闭 | D |

PC端和Target节点所发数据帧格式如上：0x03是小灯控制命令字

1. Flash图片的读写

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 帧头 | 帧长 | 硬件地址 | 软件地址 | 命令 | 子命令 | 数据 | 帧尾 |
| 字节数 | 1B | 1B | 1B | 4B | 1B | 1B | 2B | 1B |
| PC端 | P | 12 | 0xff |  | 0x06 | I | 0,0 | C |
|  |  |  |  |  | 0x06 | D |  |  |
|  |  |  |  |  | 0x06 | F |  |  |
|  |  |  |  |  | 0x04 |  |  |  |
| Target | T | 12 | 0xff |  | 0x02 |  | 整数，小数 | D |

PC端和Target节点所发数据帧格式如上：0x04. 读取FLASH指定页内容

0x06. 写入FLASH图片，I-信息帧，D-数据帧，F-最后一帧数据

# 第4章 综合测试

步骤一、将PCNode的串口引脚与电脑连接，给芯片通电，将targetNode也通电。

步骤二、PC节点检测：点击“检测PC节点”，即可检测PC节点和Target节点的无线通信状态。

状态包括：没有找到USB串口，有USB串口但无PC节点，串口号：PCNode（表示检测到PCNode）三种；

步骤三、数据收发：输入测试数据内容，点击“验证RF收发”按钮，下框中会收到Target节点回送的测试信息，即可验证通信良好。

步骤四、温度采集：点击“开始采集”，即可采集target的温度值，textbox内显示的是当前的温度值，chart记录温度值的变化曲线，每隔3秒采集一次；点击“停止”，则停止采集温度数据。

步骤五、小灯控制： PORTB|（1）的gpio口接上小灯，启动时，小灯默认打开，点击“关闭”，可以关闭小灯，点击“打开”，可以打开小灯并且显示小灯目前状态

步骤六、flash读写测试：设置扇区号，偏移量和读写字节数，点击“target Flash读测试”，可以读到对应扇区、对应偏移位置的数据，数据量为输入的读写字节数；设置扇区号、偏移量和写入数据，点击“target Flash写测试”，可以将所输入的数据写入到指定位置；

步骤七、图片读写：点击“打开图片”，选取要发送的图片，左侧图片框和文本框会显示待发送的图片、图片转为字符的结果和图片大小（即索要发送的字节数）。注：所选图片大小不超过5000字节。点击“发送图片”即可将图片发送至terget节点并存入flash；点击“读取图片”，可以从target端的flash中将图片字符串数据读回，显示在右侧文本框，点击“还原图片”，可以将字符串转换成为图片显示出来。



模块较多，采用了Tabcontrol分成几页，需要哪个模块，点击相应标签即可显示。

# 第5章 心得体会

通过本次综合实验，让我了解了嵌入式工程的任务设计过程、PC与Target节点通信帧协议的设计、以及PC界面的优化等方面的知识。

编写PC界面，用到了C#语言，让我体会到C#语言的方便简洁，能够让设计界面的工作更加便捷。各种控件的使用，让界面设计更加快速，入门变得简单。使用之前没用过的控件，通过网上查询，资料很多，需要加以筛选和尝试。

Target节点 方面，让我对这搭建一个工程有更加深刻的理解。从工程框架到芯片的传感器，都有了更深一步的掌握。一个工程需要规范的工程框架。使用样例工程，每个文件夹都有特定的文件，很规范，编写修改程序都很方便，让我体会到，标准的工程框架会使以后的编程更加便捷。修改文件时，在原先文件的复制版本上改，标明时间，让编写修改工作很有条理。当修改出现了问题，可以返回到修改之前的日期重新修改，这样就可以将损失降到最低，且是一个十分有效率的方法。

将多个功能组合在一起时需要综合考虑Target节点的资源利用问题，必须合理分配资源。例如各个模块的数据传输都使用串口，因此要区分是哪个模块的数据，做相应处理。

设计与实现过程让自己学到许多知识，特别是遇到问题的时候学会的查找资料，网络上大量的资源都可以使用，只需要认真检索一定可以找到需要的内容。设计系统需要足够的耐心。

# 参考文献

[1] 王宜怀, 朱仕浪, 郭芸. 嵌入式技术基础与实践（第三版）[ M ]. 北京: 清华大学出版社, 2013:

# 附录 使用说明

系统使用KW01节点为核心，其最小原理图见教材所示。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表2-3 KW01硬件最小系统引脚表 | | | | | |
| 引脚名 | | 引脚号 | 典型值 | 功能描述 |
| LGA |
| 电源  输入 | VDD | 19、55 | 3.3V | 电源 |
| VSS1 | 4、20 | 0V | 地 |
| VDDA、VSSA | 56 | 3.3V | A/D模块的输入电源 |
| VSSA | 3 | 0V | A/D模块的输入电源 |
| VREFH、VREFL | 1 | 3.3V | A/D模块的参考电压 |
| VREFL | 2 | 0V | A/D模块的参考电压 |
| VBAT1 | 43 |  | RF收发器电池供电 |
| VBAT2 | 35 |  | RF收发器电池供电 |
| RESETB，  RESET（RF） | | 33，48 | 复位引脚（双向），作为输入，拉低可使芯片复位；作为输出，上电复位期间有低脉冲输出，输出脉宽最小约100ns，芯片完成复位。常态下，引脚拉高，可通过复位按键拉低复位。 | |
| EXTAL、XTAL | | 15、16 | 分别为无源晶振输入、输出引脚 | |
| XTA（RF）、XTB（RF） | | 46、47 | 收发器参考晶振 | |
| SWD\_CLK | | 10 | SWD时钟信号线 | |
| SWD\_DIO | | 11 | SWD数据信号线 | |
| 引脚个数统计 | | LGA封装18个 | | | |

引脚图：

