

# PMA71xx / PMA51xx

SmartLEWIS™ MCU

射频发射器 FSK/ASK 315/434/868/915 MHz

内置 8051 微控制器，配 10 位 ADC

内置 125 kHz ASK LF 接收器

## 应用说明

PMAfob 软件示例

修订版 1.2, 2010-10-11

## 感应与控制

**版本 2010-10-11**

由 **Infineon Technologies AG** 出版

**81726 Munich, Germany**

**© 2010 Infineon Technologies AG 保留所有权利。**

#### **法律免责声明**

在任何情况下均不得将本文件所提供的信息视为对条件或特征的担保。英飞凌科技公司特此声明，对本文中所提及的任何示例或提示、任何典型数值和/或任何与设备应用相关的信息，不作任何及所有形式的担保或承担责任及所有形式的责任（包括但不限于对不侵犯任何第三方知识产权的担保）。

为方便客户浏览，英飞凌以下所提供的将是有关英飞凌产品及服务资料的中文翻译版本。该中文翻译版本仅供参考，并不可作为任何论点之依据。虽然我们尽力提供与英文版本含义一样清楚的中文翻译版本，但因语言翻译和转换过程中的差异，可能存在不尽相同之处。因此，我们同时提供该中文翻译版本的英文版本供您阅读，请参见 [【PMA71xx\\_PMA51xx\\_PMAfobSoftwareExample】](#)。并且，我们在此提醒客户，针对同样的英飞凌产品及服务，我们提供更加丰富和详细的英文资料可供客户参考使用。请详见 [【Transmitter + Microcontroller】](#)

客户理解并且同意，英飞凌毋须为任何人士由于其在翻译原来的英文版本成为该等中文翻译版本的过程中可能存在的任何不完整或者不准确而产生的全部或者部分、任何直接或者间接损失或损害负责。英飞凌对于中文翻译版本之完整与正确性不担负任何责任。英文版本与中文翻译版本之间若有任何歧异，以英文版本为准，且仅认可英文版本为正式文件。

您如果使用以下提供的资料，则说明您同意并将遵循上述说明。如果您不同意上述说明，请不要使用本资料。

#### **信息**

有关技术、交货条款及条件和价格的更多信息，请与您最近的英飞凌科技公司办事处 ([www.infineon.com](http://www.infineon.com)) 联系。

#### **警告**

由于技术要求，元件可能含有危险物质。如需相关型号的信息，请与距离您最近的英飞凌科技公司办事处联系。如果可能合理地预期此类元件的故障会导致生命支持设备或系统发生故障或影响该设备或系统的安全性或有效性，则英飞凌科技公司提供的元件仅可用于获得英飞凌科技公司明确书面批准的生命支持设备或系统。生命支持设备或系统的目的是植入人体或支持和/或保持并维持和/或保护生命。如果出现故障，则可能危及使用者或他人的健康。

## 目录

1	引言	6
2	文件结构	6
2.1	头文件	7
2.1.1	Reg_PMA71xx_PMA51xx.h	7
2.1.2	PMA71xx_PMA51xx_Library.h	7
2.1.3	defines.h	7
2.1.4	RF_Functions.h	8
2.1.5	Misc_Functions.h	8
2.2	源文件	8
2.2.1	STARTUP_PMA71xx_PMA51xx.A51	8
2.2.2	InitEEPROM.A51	8
2.2.3	main.c	8
2.2.4	Misc_Functions.c	9
2.2.4.1	Check_SampleArray()	9
2.2.4.2	Calc_RCode()	9
2.2.4.3	ProcessButtonPress()	9
2.2.4.4	XTEA_encipher()	9
2.2.5	RF_Functions.c	9
2.2.5.1	RFInit()	9
2.2.5.2	RFTransmit()	9
2.2.5.3	TransmitCmdFrame_AES()	10
2.2.5.4	TransmitCmdFrame_XTEA()	10
2.3	函数库文件 (PMA71xx_PMA51xx_Library.lib)	10
3	程序流程	10
4	模拟 EEPROM 初始化	12
5	端口采样	12
6	射频协议	13
6.1	AES 成帧信息净负荷	13
6.2	XTEA 成帧信息净负荷	14
6.2.1	半字节交换	15
7	灵活的软件	15
7.1	复位配置	16
7.2	软件配置和 GPIO/按钮分配	16
	参考资料	18

修订历史: 2010-10-11, 修订版 1.2

先前版本: 1.1

页码	变更内容 (自上次修订后的主要变更)
6	修改了源代码的下载路径
6, 8	可以使用唯一的 ID 作为 PMA 的唯一 ID 或用户自定义的唯一 ID
12	纠正了图 3 中的错误: 0..按下按钮, 1..松开按钮
	概括介绍无线遥控应用, 不仅是 RKE
1,8-17	通过长按按钮进行软件配置的说明

### Infineon Technologies AG 的商标

ABM™、BlueMoon™、CONVERGATE™、COSIC™、C166™、FALC™、GEMINAX™、GOLDMOS™、ISAC™、OMNITUNE™、OMNIVIA™、PROSOC™、SEROCO™、SICOFI™、SIEGET™、SMARTi™、SmartLEWIS™、SMINT™、SOCRATES™、VINAX™、VINETIC™、VOIPRO™、X-GOLD™、XMM™、X-PMU™、XWAY™

### 其它商标

Microsoft Corporation 的 Microsoft®、Visio®、Windows®、Windows Vista®、Visual Studio®、Win32®。Linus Torvalds 的 Linux®。Adobe Systems Incorporated 的 FrameMaker®、Adobe® Reader™、Adobe Audition®。Comneon GmbH & Co OHG 的 APOXI®、COMNEON™。ARM Limited 的 PrimeCell®、RealView®、ARM®、ARM® Developer Suite™ (ADS)、Multi-ICE™、ARM1176JZ-S™、CoreSight™、Embedded Trace Macrocell™ (ETM)、Thumb®、ETM9™、AMBA™、ARM7™、ARM9™、ARM7TDMI-S™、ARM926EJ-S™。ParthusCeva Inc. 的 OakDSPCore®、TeakLite®、DSP Core, OCEM®。Global Locate 的 IndoorGPS™、GL-20000™、GL-LN-22™。MIPI Alliance 的 mipi™。DECT Forum 的 CAT-iq™。MIPS Technologies, Inc. 的 MIPS™、MIPS II™、24KEc™、MIPS32®、24KEc™。Texas Instruments Incorporated 的 Texas Instruments®、PowerPAD™、C62x™、C55x™、VLYNQ™、Telogy Software™、TMS320C62x™、Code Composer Studio™、SSI™。Bluetooth SIG, Inc. 的 Bluetooth®。The Infrared Data Association 的 IrDA®。Sun Microsystems, Inc. 的 Java™、SunOS™、Solaris™。Koninklijke Philips Electronics N.V. 的 Philips®、I2C-Bus®。Seiko Epson Corporation 的 Epson®。Kabushiki Kaisha Hattori Seiko Corporation 的 Seiko®。Matsushita Electric Industrial Co., Ltd. 的 Panasonic®。Murata Manufacturing Company 的 Murata®。Taiyo Yuden Co., Ltd. 的 Taiyo Yuden™。TDK Electronics Company, Ltd. 的 TDK®。Motorola, Inc. 的 Motorola®。National Semiconductor Corporation 的 National Semiconductor®、MICROWIRE™。The Institute of Electrical and Electronics Engineers, Inc. 的 IEEE®。Samsung Corporation 的 Samsung®、OneNAND®、UtRAM®。Toshiba Corporation 的 Toshiba®。Dallas Semiconductor Corp. 的 Dallas Semiconductor®、1-Wire®。The International Organization for Standardization 的 ISO®。The International Engineering Consortium 的 IEC™。EMVCo, LLC 的 EMV™。Zetex Semiconductors 的 Zetex®。Microtec Research, Inc. 的 Microtec®。Cadence Design Systems, Inc. 的 Verilog®。The American National Standards Institute, Inc. 的 ANSI®。Wind River Systems, Inc. 的 WindRiver® 和 VxWorks®。Mentor Graphics Corporation 的 Nucleus™。OmniVision Technologies, Inc. 的 OmniVision®。Sharp Corporation 的 Sharp®。SSymbian Software Ltd. 的 Symbian OS®。Openwave Systems, Inc. 的 Openwave®。Maxim Integrated Products, Inc. 的 Maxim®。Spansion LLC 的 Spansion®。Micron Technology, Inc. 的 Micron®、CellularRAM®。RF Micro Devices, Inc. 的 RFMD®。EPCOS AG 的 EPCOS®。The Open Group 的 UNIX®。Tektronix, Inc. 的 Tektronix®。Intel Corporation 的 Intel®。Qimonda AG 的 Qimonda®。Samsung Corporation 的 1GOneNAND®。Hilgraeve, Inc. 的 HyperTerminal®。The MathWorks, Inc. 的 MATLAB®。Red Hat, Inc. 的 Red Hat®。Cadence Design Systems, Inc. 的 Palladium®。SIRIUS Satellite Radio Inc. 的 SIRIUS Satellite Radio®。TOKO Inc. 的 TOKO®。以及 KEIL™。

本文件包含内容如有变更, 恕不另行通知。

商标最后更新于 2009-03-10

## 插图目录

图 1	PMAfob 软件示例的文件结构 .....	7
图 2	PMAfob 软件示例程序流程 .....	11
图 3	按钮按下和间隔定时器的时间设置 .....	12
图 4	模拟 EEPROM 写访问期间的端口采样 .....	13
图 5	射频帧, 带 AES 加密的信息净负荷 .....	14
图 6	射频帧, 带 XTEA 加密的信息净负荷 .....	14
图 7	根据滚动码 LSD 进行半字节交换 .....	15

## 1 引言

**PMAfob** (单片遥控解决方案) 软件示例介绍了一个在使用 PMA51xx 或 PMA71xx 进行无线远程遥控应用 (例如远程无钥匙进入 (RKE) 或家庭自动化) 上具有可行性的软件解决方案。可用的软件示例文件包括内联文档, 这些文档专为帮助程序员轻松入门和缩短软件面世时间而编制。您可以通过 [http://www.infineon.com/PMA\\_tooling](http://www.infineon.com/PMA_tooling) 下载软件示例文件。PMAfob 软件示例源代码安装程序是 *PMAfob\_DEMO\_Tx\_Sources\_Vx.y.msi*。该安装程序在下述下载包中可以找到:

- PMAfob - 软件示例
- PMAfob - RKE 演示软件
- PMAfob – 家庭自动化演示软件

此外, 还可以在下载包中找到更多用 Doxygen 生成的源代码文件, 在运行 *PMAfob\_DEMO\_Tx\_Sources\_Vx.y.msi* 安装程序后, 打开文件 *./PMAfobSoftwareExample/html/index.html* 就可以在标准浏览器中显示这些文件。

本软件示例支持下述特性:

- 五个按钮 (处理五个外部唤醒功能)
- 按钮卡住检测
- 弹回按钮
- 安全通信
  - AES<sup>1)</sup> 或 XTEA<sup>2)</sup> 加密
  - 生成滚动码
- 电池电压测量
- 使用掉电模式达到节能效果
- 唯一 ID (PMA 或用户定义)
- 通过长按按钮进行软件配置
  - 修改射频成帧
  - 切换加密开/关, 修改波特率
  - 修改唯一密钥号 (PMA 或用户定义)

本文件介绍了该软件大致的文件结构、程序流程、模拟 EEPROM 初始化、端口采样和在 PMAfob 软件示例中使用的射频协议。此外, 还介绍了如何通过长按按钮可实施哪些配置。本文件与 1.1 修订版源代码 (*PMAfob\_DEMO\_Tx\_Sources\_V1.1.msi*) 内容相容。

## 2 文件结构

本章节概括介绍了 PMAfob 软件示例的文件结构, 描述了通过源文件实现的功能。图 1 显示各文件之间的关联。

1) 高级加密标准

2) 扩展的微型加密算法

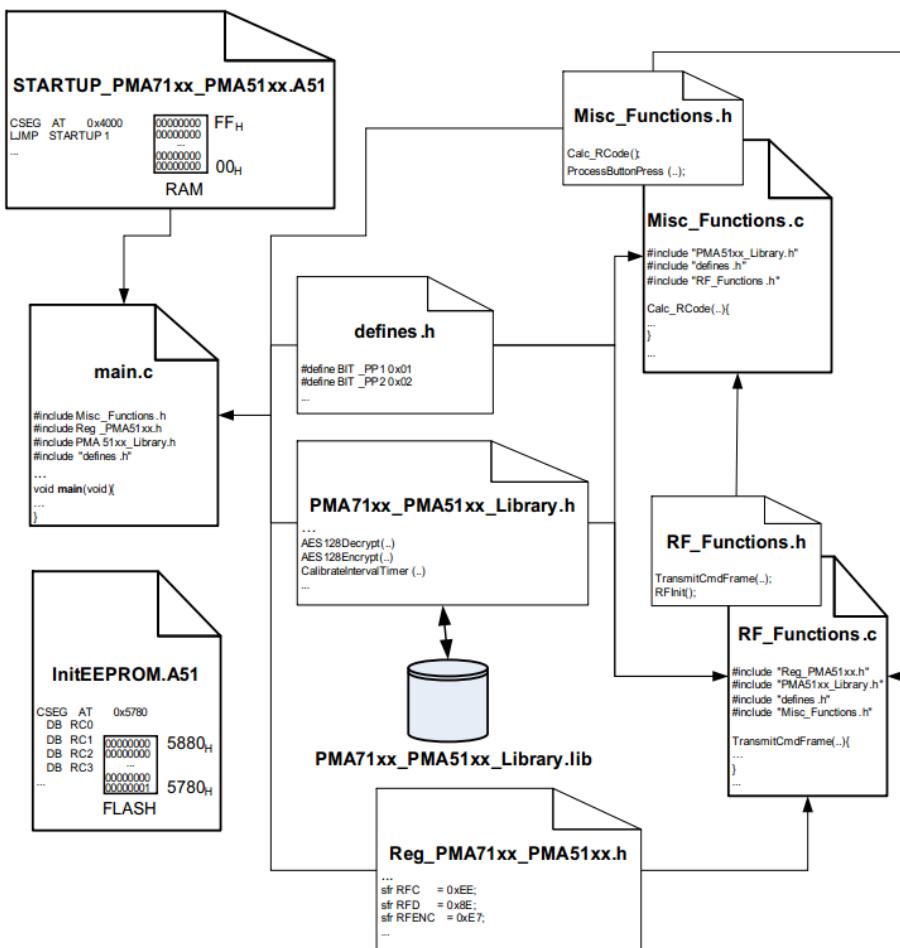


图1 PMAfob 软件示例的文件结构

## 2.1 头文件

头文件中定义了不同模块接口。

### 2.1.1 Reg\_PMA71xx\_PMA51xx.h

这是 PMA71xx / PMA51xx 寄存器定义文件。文件中定义了 PMA71xx / PMA51xx 所有的 SFR (特殊功能寄存器)。如果需要直接访问 SFR，则必须将该文件添加到项目中。

### 2.1.2 PMA71xx\_PMA51xx\_Library.h

`PMA71xx_PMA51xx_Library.h` 是 PMA71xx / PMA51xx 函数库接口。该文件中定义了函数库的原型和关于射频传输的一些声明。如果要使用 PMA71xx / PMA51xx 函数库，则必须将该文件和 `PMA71xx_PMA51xx_Library.lib` 一起添加到项目中。函数库中所有函数详见 [1]。

### 2.1.3 defines.h

`Defines.h` 含有能增加代码可读性的比特位定义。枚举型 `Encryption_Type`、`Encryption_Status` 和 `Unique_Key_Nr` 也于此定义。使用结构体 `ActualButtonPresses_s` 识别已按下的按钮，对实际按钮按下时间计时。

## 2.1.4 RF\_Functions.h

*RF\_Functions.h* 中定义了函数 *TransmitCmdFrame\_AES()* 和 *TransmitCmdFrame\_XTEA()* 的接口。

## 2.1.5 Misc\_Functions.h

*Misc\_Functions.h* 中定义了函数 *Calc\_RCode()*、*ProcessButtonPress()* 和 *XTEA\_encipher()* 的接口。

## 2.2 源文件

源文件包括了启动文件、EEPROM 初始化文件、按钮按下检测函数，滚动码生成、射频成帧和传输的实现。

### 2.2.1 STARTUP\_PMA71xx\_PMA51xx.A51

本文件是标准 8051 启动文件 *STARTUP.A51*（由 KEIL™ 提供）的修订副本，必须添加到项目中。如果不添加，链接器就会自动添加标准 *STARTUP.A51*。如果是因复位而启动 PMA71xx / PMA51xx，整个 **idata** 存储区  $00_{\text{H}} - FF_{\text{H}}$  都将被初始化为  $00_{\text{H}}$ 。将 SFR 比特位 **CFG2.4 [PDLMB]** 设为 0，就可以在掉电或热关机模式下为 **idata** 存储区低位  $00_{\text{H}} - 7F_{\text{H}}$  供电。如果 PMA71xx/PMA51xx 从掉电或热关机状态醒来，则检查 **PDLMB** 比特位。如果在掉电或热关机状态时对低位 **idata** 存储区供电，只有 **idata** 高位存储块  $80_{\text{H}} - FF_{\text{H}}$  置 0。其它情况下所有 **idata** 存储区  $00_{\text{H}} - FF_{\text{H}}$ ，都被初始化为  $00_{\text{H}}$ 。

最终，设置堆栈指针，执行 **ljmp** 跳转至 *main()* 程序。

### 2.2.2 InitEEPROM.A51

本文件中定义了滚动码在闪存中的起始值位置。程序向下加载到设备时，滚动码被写入闪存用户数据扇区 I 中。

### 2.2.3 main.c

本文件包括 PMAfob 软件示例的 *main()* 函数，在 *STARTUP\_PMA71xx\_PMA51xx.A51* 之后执行。

长按按钮，使用下列全局变量来修改射频帧、加密和唯一密钥号。

- *My\_Encryption\_Type*: 可以使用 AES 或 XTEA 成帧。RF 帧的定义参见第 6 章。
- *My\_Encryption\_Status*: 可以对 AES 或 XTEA（取决于 *My\_Encryption\_Type*）加密进行开/关切换。
- *My\_Uneque\_Key\_Nr*: 可以使用 4 个字节的 PMA 唯一 ID 或用户自定义的唯一 ID。使用 *defines.h* 中的 **#define USER\_KEY\_NR** 来设置用户自定义的唯一 ID。

**xdata** 也是全局变量。*main()* 函数会检查唤醒位 **DSR.1 [WUP]** 以判断器件是因复位还是因掉电模式时的唤醒事件而启动。

如果是复位，则将 **PP1-PP4** 和 **PP6** 配置为外部唤醒。因此端口方向被设为输入，内部上拉电阻和外部唤醒 **WU0-WU4** 启用（未屏蔽）。**PP8** 也被设为输入，启用 **PP8** 上拉电阻。使用 **PP8** 来检查是否已使用端口采样功能（在擦除模拟 EEPROM 的一个扇区后）。最后，会对一些变量进行初始化并将间隔定时器设为最大唤醒间隔，大约 524 s。

唤醒例行程序包括对看门狗定时器的操作。使用外部唤醒和间隔定时器唤醒以检测按钮是否弹回、是否被按下、是否被卡住。

## 2.2.4 Misc\_Functions.c

*Check\_SampleArray()*、*Calc\_RCode()*、*ProcessButtonPress()* 和 *XTEA\_encipher()* 函数在本文件内实现。每个函数的功能描述参见下文。

### 2.2.4.1 Check\_SampleArray()

*Check\_SampleArray()* 是对样本阵列检查的基本实现，样本阵列被用于在执行 *Calc\_RCode()* 中的 EEPROM 写入功能过程中对端口进行检测。每进行 4 次写访问就对 WU 端口取一次样。使用 PP8 来检查端口是否被采样。若端口已被采样，因为内部上拉电阻的关系，样本列阵在 PP8 位置上为  $1_B$ ，否则为  $0_B$ 。

### 2.2.4.2 Calc\_RCode()

从模拟 EEPROM 中加载先前的滚动码，根据 *main.c* 文件中全局变量 *My\_Encryption\_Type* 的设置，计算出新的滚动码。若选择了 AES 成帧，则将先前的滚动码乘以长型常量（32 位）值，然后添加整型常量（16 位）值；若选择了 XTEA 成帧，则将滚动码作为简单型 32 位计数器加以实现。因此，先前的滚动码会增加 1，所得的实际滚动码会被存储到模拟 EEPROM 中。执行 EEPROM 写入功能时进行端口检测，确保其间按下按钮能够被检测到。

### 2.2.4.3 ProcessButtonPress()

按钮按下之后，按钮 ID 被存储到电池缓冲 *xdata* 中。*ProcessButtonPress()* 检查按下按钮的 ID 与存储在 *xdata* 中的 ID（之前按下的按钮）是否相同，计数器增加一个数。若该计数器达到预先定义值，则调用 *TransmitCmdFrame\_AES()* 或 *TransmitCmdFrame\_XTEA()* 将按钮 ID 设为 0，从而令计数器在按钮卡住时不再增数。在 *TransmitCmdFrame\_AES()* 或 *TransmitCmdFrame\_XTEA()* 之内，通过 *Calc\_RCode()* 计算出一个新的滚动码。在 *Calc\_RCode()* 调用期间若发现按钮被按下，则会计算一个新的滚动码，发送相关射频帧。

### 2.2.4.4 XTEA\_encipher()

XTEA 加密数据算法是 David Wheeler 和 Roger Needham 发布的公开算法。这个算法是通过 C 语言来实现，至今未优化成可与 8051 微控制器相容。采用 32 轮加密。

## 2.2.5 RF\_Functions.c

函数 *RFInit()*、*RFTransmit()*、*TransmitCmdFrame\_AES()* 和 *TransmitCmdFrame\_XTEA()* 在 *RF\_Functions.c* 中实现。

### 2.2.5.1 RFInit()

首先，定义和初始化结构体 *RF\_Config* 的一个变量，参见 *PMA71xx\_PMA51xx\_Library.h*。然后，调用库函数 *InitRF()* 为射频传输所需的所有 SFR 设置合适的值。关于结构体 *RF\_Config* 的每一个元素的详细说明可以参见 [1]。

### 2.2.5.2 RFTransmit()

使用这个函数来生成 TDA523x 接收器所需的射频成帧。射频帧详见第 6 章。

### 2.2.5.3 `TransmitCmdFrame_AES()`

射频帧的信息净负荷就是在这个函数内生成的。本信息净负荷的结构参见[第 6.1 章](#)。滚动码和电池电压是信息净负荷的组成部分，通过库函数 `MeasureSupplyVoltage()` 测得。整个 128 位的信息净负荷可以进行 AES 加密，然后嵌入射频帧中。

### 2.2.5.4 `TransmitCmdFrame_XTEA()`

射频帧的信息净负荷就是在这个函数内生成的。本信息净负荷的结构参见[第 6.2 章](#)。滚动码和电池电压是信息净负荷的组成部分，通过库函数 `MeasureSupplyVoltage()` 测量。如第 6.2.1 章节所示，若射频帧加密，根据滚动码的 LSB，信息净负荷中的一些比特位会进行半字节交换。88 位的信息净负荷中有 64 位可以通过 XTEA 加密并嵌入射频帧中。

## 2.3 函数库文件 (`PMA71xx_PMA51xx_Library.lib`)

这是实现 `PMA71xx_PMA51xx_Library.h` 中定义的所有函数的函数库。函数库中所有函数详见[\[1\]](#)。

## 3 程序流程

[第 11 页图 2](#) 显示了 PMAfob 软件示例的程序流程。

器件在启动文件内开始执行程序。RAM 初始化后，检查 SFR DSR.1 [WUP] 中的唤醒标志，判断器件是因复位还是因唤醒事件而启动。

若 PMA71xx / PMA51xx 是因复位而启动，则 PP1-PP4 和 PP6 (WU0-WU4) 被配置成外部唤醒引脚。因此，端口方向必须被设为输入，内部上拉电阻必须被激活（假设已按下的按钮在引脚上产生了一个 LOW）。PP8 也被设为输入，内部上拉电阻被激活，以供分析端口采样阵列使用。最后，间隔定时器被设为可能的最长唤醒间隔，大约 524 s，PMA71xx / PMA51xx 被设为掉电模式以降低能耗，等待唤醒事件。

如果 PMA71xx / PMA51xx 是因唤醒事件从掉电模式中启动，则检查唤醒源。PMAfob 软件示例能够处理七种唤醒源：看门狗定时器、外部唤醒 WU0-WU4 和间隔定时器。看门狗定时器唤醒的优先级别最高。若发生看门狗定时器唤醒，则触发软件复位以确保所有 SFR 处于预设状态。

如果检查发现不是看门狗定时器唤醒，接下来检查外部唤醒。检测到外部唤醒之后，相关的唤醒源被禁用（屏蔽），唤醒 ID 被存入 `xdata` 中，间隔定时器被设为 50 ms。

在间隔定时器唤醒例行程序中，检查 PP1-PP4 和 PP6 引脚以检测按钮是否仍被按住。若先前存放在 `xdata` 中的外部唤醒 ID 等于当前按下的按钮，计数器会增加数值。若计数器达到预设值，则视为按钮被按下。然后，计算新的滚动码，测量电池电压，向射频帧插入相关指令，对射频帧进行 AES 或 XTEA 加密（若开启了加密功能），然后进行传输。向 EEPROM 写入这个新的滚动码时，可对端口进行任一采样操作。在向 EEPROM 写入期间若发现按钮被按下，则会计算一个新的滚动码，发送相关射频帧。每个在间隔定时器唤醒例行程序中被确定为未被按下的按钮，相应的唤醒会被（重新）使能。如果没有按钮被按下，间隔定时器会被设成可能的最长唤醒间隔时间（524 s 左右）。如果按钮被按下至少 1 小时，在这一期间又没有其他按钮被按下，间隔定时器唤醒例行程序会检测到键被卡住。软件的部分配置，可以通过长按按钮加以修改（参见[第 7 章](#)）。因此，可以使用间隔定时器来检查按钮按下的时间是否过长，然后通过 `#define SWITCH_DUR(defines.h)` 来设置。

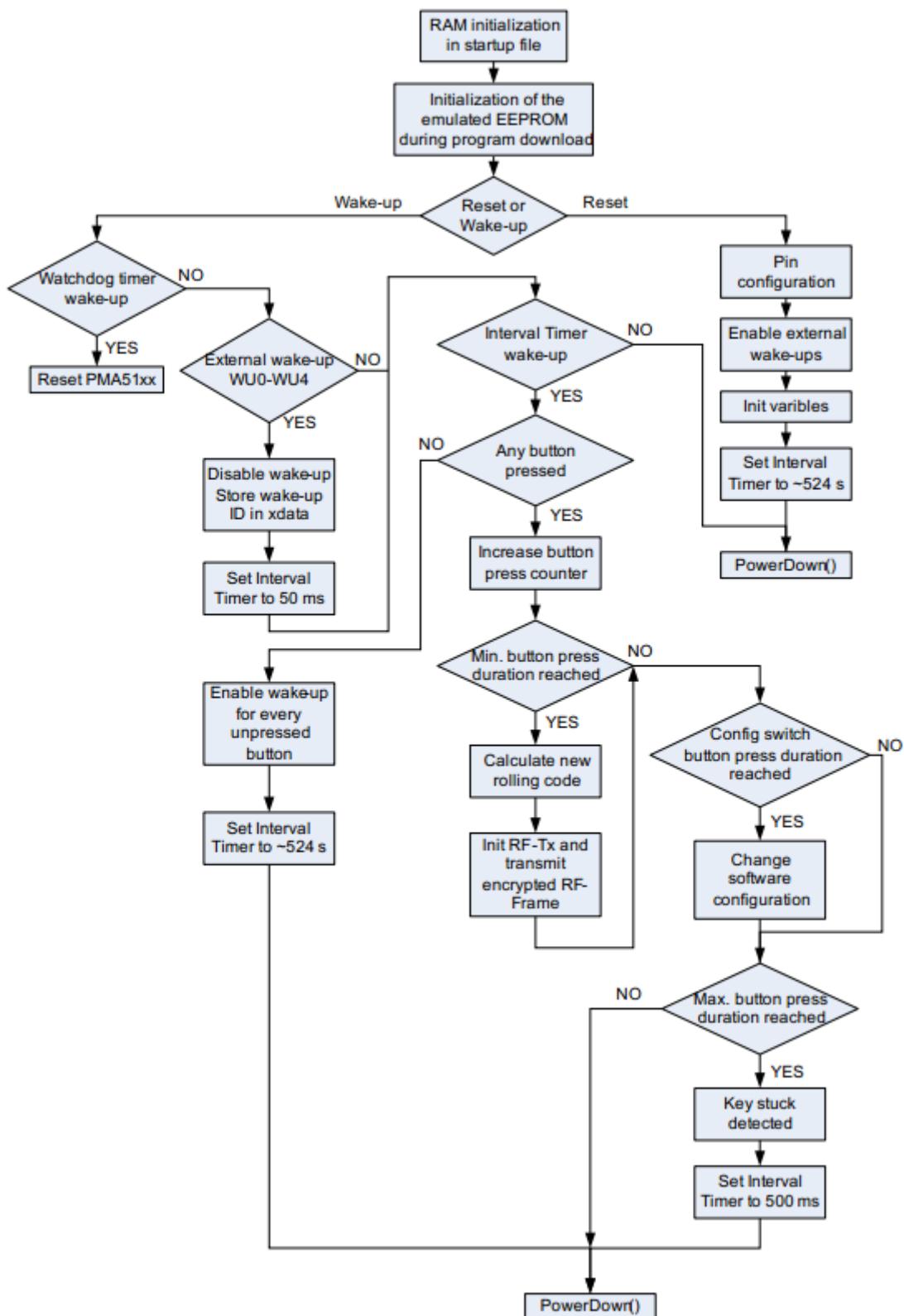


图 2 PMAfob 软件示例程序流程

第 12 页图 3 介绍了按钮按下的时间设置和间隔定时器唤醒间隔的变化方式。如果没有按下按钮，为了降低能耗，间隔定时器将按照可能的最长间隔时间（524 s 左右）来唤醒 PMA71xx / PMA51xx。若按下按钮，则检测到外部唤醒，并将间隔定时器设为 50 ms。间隔定时器唤醒服务例行程序每 50 ms 检查一次按钮是否仍被按住不放，每次测量之间将器件设为掉电模式。通过这种方式来防止按钮误检测。

如果按钮被长按至少 150 ms (间隔定时器唤醒服务例行程序中三次按钮检查的结果都是按钮被按下)，识别为按钮被按下，启动射频传输。可以通过修改定义在 *defines.h* 中的 *BUTTON\_PRESS\_DUR* 的值和/或修改间隔定时器设置，轻松修改防抖动（按压检测）时间（原为 150 ms）。只要按钮处于按下状态，间隔定时器每 50 ms 就会“醒来”检查按钮按下情况。这种情况最长持续 1 小时。

如果按钮被按下超过 3 秒，则会根据按按钮的情况俩修改软件配置（参见第 7 章）。可以通过修改定义在 *defines.h* 中的 *SWITCH\_DUR* 的值和/或修改间隔定时器的设置，轻松修改配置切换时间。

若按钮被按下 1 个小时左右，在这期间亦无任何其他按钮被按下，则被视为按钮卡住，为了降低能耗，将间隔定时器设为 500 ms。若释放按钮，又没有其他按钮被按下，间隔定时器会再被设为 524 s 左右。

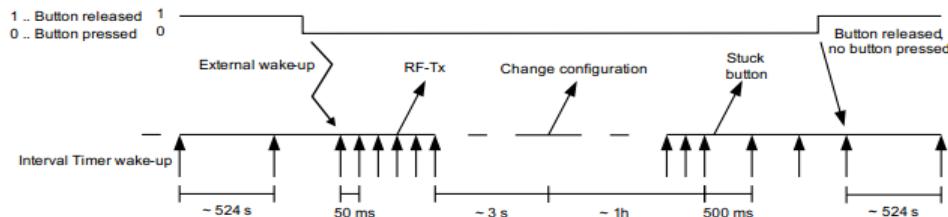


图 3 按钮按下和间隔定时器的时间设置

## 4 模拟 EEPROM 初始化

PMAfob 软件示例将滚动码存储在模拟 EEPROM 中。这样做是为了在将 PMA71xx / PMA51xx 设置为掉电模式时，保证滚动码不丢失。但是，模拟 EEPROM 应该如何初始化？

PMA71xx / PMA51xx 模拟 EEPROM 有几种不同的初始化方法：

- 使用 PMA71xx / PMA51xx 函数库函数 *EEPROM\_Init()* 进行初始化
- 在程序下载过程中进行初始化

一种方法是通过调用库函数 *EEPROM\_Init()*，在复位例行程序中初始化模拟 EEPROM。在使用滚动码来提高安全等级的遥控应用中，这种方法有一个很大的缺点：更换电池之后，必须对遥控器和接收器进行同步。问题在于遥控器的滚动码会被重置，而接收器的滚动码却保持不变。

解决这个问题的办法是在程序下载过程中，初始化模拟 EEPROM（包括滚动码起始值）。更换电池后，PMAfob 的滚动码会保持不变，无需同步。

注：在程序下载过程中初始化模拟 EEPROM（包括滚动码起始值）的注意事项：

1. 将 *InitEEPROM.A51* 添加至 Keil 项目
2. 确保用户数据扇区 I 和 II 在程序下载之前已被擦除

## 5 端口采样

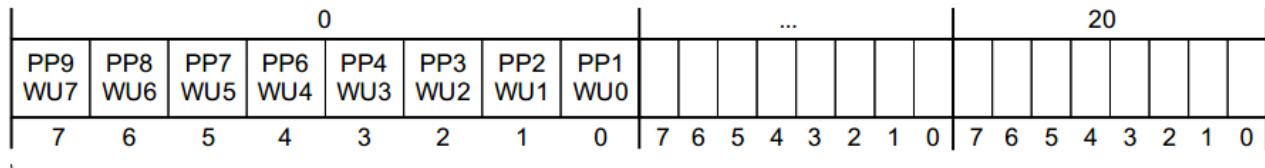
检测到按钮被按下之后，一个新的滚动码将被计算出来并存储到模拟 EEPROM 中。每 4 次对模拟 EEPROM 进行写访问时，

必须擦除一个用户数据扇区，用时约 102 ms。为保证在这期间能正常使用按钮按下检测功能，可以提供唤醒端口采样服务。PMAfob 软件示例介绍了如何实现端口采样功能。图 4 显示端口采样列阵的形状和阵列操纵方法。在 *Calc\_RCode(..)* 中调用库函数 *Wr\_EELong(..)*。如果在 *Wr\_EELong(..)* 期间，用户数据扇区被擦除，则每 5ms 对唤醒端口采样一次，然后插入采样阵列中，如图 4 所示。

*Check\_SampleArray(..)* 函数分析采样阵列，判断按钮是否被按下。若检测到按钮被按下，则发送相应的射频帧。

```
Calc_RCode (...) {
```

```
    Wr_EELong (...) {
```



```
}
```

```
}
```

```
Check_SampleArray (...)
```

Command to  
transmit

图 4 模拟 EEPROM 写访问期间的端口采样

## 6 射频协议

PMAfob 软件示例的设计与 TDA523x 接收器相容。因此必须使用特殊的射频协议。

下述设置供射频传输使用（在 *RFInit()* 中设置，参见 *RF\_Functions.c*）：

- **编码：** 曼切斯特
- **调制：** FSK
- **波特率：** 9600 bps (开启加密)； 4800 bps (关闭加密)
- **频率：** 434 MHz

射频帧开头是一个由 8 个曼切斯特编码数字位 (16 码片) 构成的RUNIN 序列，TDA523x 接收器使用这些比特位进行内部滤波器设置和频率调节。后面接着 16 码片长的 TSI (电报开始标识符)，用于同步帧和检测数字帧的确切起始位置 (信息净负荷)。为了检测 EOM (消息结束)，会发送一个曼切斯特违例 (2 个  $1_B$  码片)。

信息净负荷取决于 *main.c* 文件中定义的全局变量 *My\_Encryption\_Type* 的设置。

### 6.1 AES 成帧信息净负荷

AES 成帧信息净负荷由 128 个比特位构成，包括滚动码、唯一密钥号、命令代码、电池电压和按钮按下总次数。关于包括 (AES 加密) 信息净负荷的整个射频帧，参见第 14 页图 5。

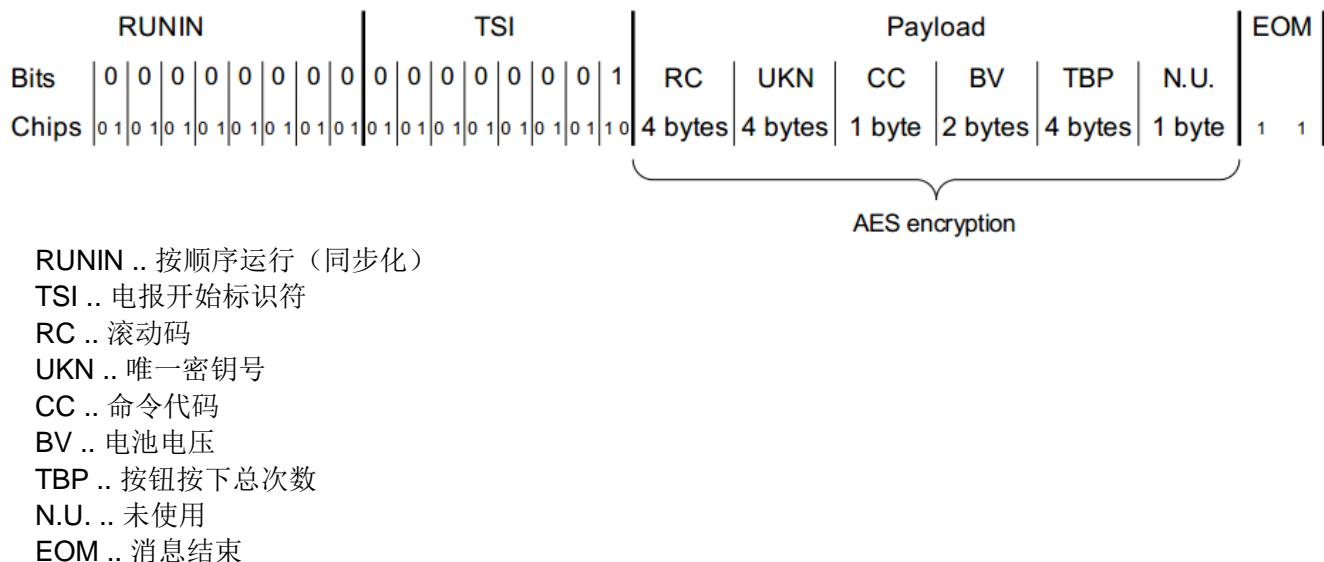
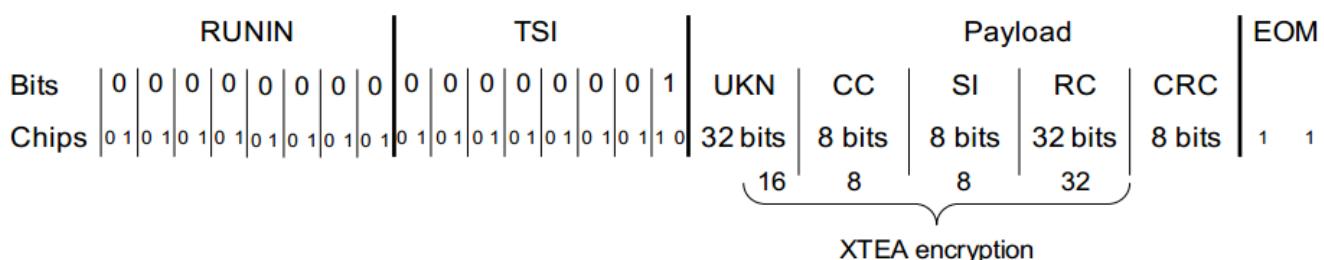


图 5 射频帧带 AES 加密信息净负荷

## 6.2 XTEA 成帧信息净负荷

88 位信息净负荷中有 64 位可以进行 XTEA 加密。信息净负荷中包括唯一密钥号、命令代码、状态信息（电池电压）、滚动码和 CRC 校验和。展示了包括信息净负荷在内的整个射频帧，介绍了信息净负荷中哪个部分可以用 XTEA 加密。

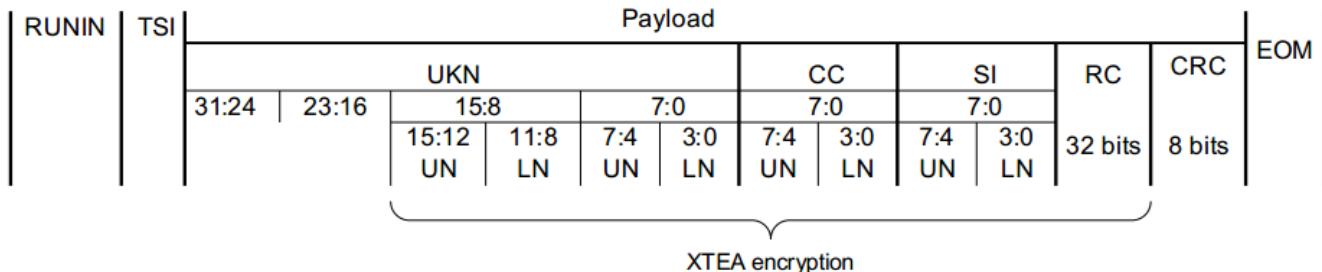


RUNIN .. 按顺序运行 (同步化)  
TSI .. 电报开始标识符  
UKN .. 唯一密钥号  
CC .. 命令代码  
SI .. 状态信息 (如电池电压)  
RC .. 滚动码  
CRC .. UKN、CC、SI、RC 校验和  
EOM .. 消息结束

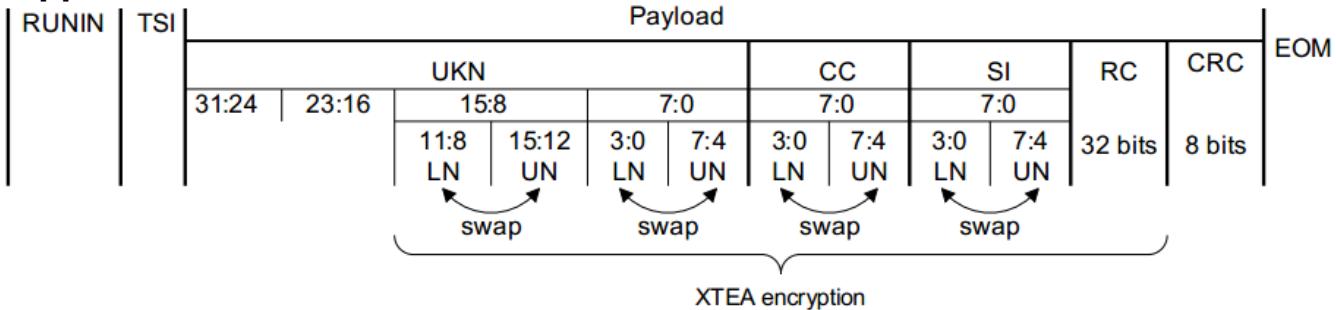
### 6.2.1 半字节交换

为了提高安全性, 若加密功能开启, 则将根据滚动码的 LSB, 对唯一密钥号的 2 个低字节、命令代码和状态信息字节进行半字节交换。若 LSB 为  $1_B$ , 则进行半字节交换。半字节交换参见第 15 页图 7。

RC[0] == 0 (偶: 不进行半字节交换)



RC[0] == 1 (奇: 进行半字节交换)



RUNIN .. 按顺序运行 (同步化)

TSI .. 电报开始标识符

RC .. 滚动码

UKN .. 唯一密钥号

CC .. 命令代码

SI .. 状态信息 (如电池电压)

CRC .. UKN、CC、SI、RC 校验和

EOM .. 消息结束

LN .. 低位半字节

UN .. 高位半字节

图 7 根据滚动码 LSB 进行半字节交换

## 7 灵活的软件

长按按钮即可配置 PMAfob 软件示例中的软件, 无需对器件进行刷机操作。因此, PP1-PP4 通用输入输出端口 (GPIO) 中的一个或 PP6 必须连接至接地 (GND)。提供 5 个按钮 (GPIO) 供用户更换射频帧、对开/关加密功能进行切换、修改数据率、使用 PMA 唯一 ID 的 4 个字节或用户自定义的唯一 ID。

使用 `#define SWITCH_DUR (defines.h)` 设置用于软件配置的按钮所按时长。默认切换时间为 3 s。PMAfob Home Automation 和 PMAfob RKE Demo 中使用的 PMAfob 有不同的配置和按钮分配, 如本章所示。

## 7.1 复位配置

复位后 — 例如更换电池后 — 软件配置如下：

- 成帧: AES
- 加密: 开启
- 唯一 ID: 用户自定义的唯一 ID (defines.h: #define USER\_KEY\_NR)
- 波特率: 9600 bps

## 7.2 软件配置和 GPIO/按钮分配

是已实现的软件配置和该配置的 GPIO/按钮分配。

表 1 软件配置和 GPIO/按钮分配

PMAfob 按钮	GPIO	软件配置
	PP1	<p><b>对开/关加密进行切换:</b> 开启/关闭加密功能, 根据所选的成帧, 决定采用 AES 还是 XTEA 加密。 下述配置将发生变化, 如果开启了加密功能:</p> <ul style="list-style-type: none"><li>• <b>开启加密时</b><ul style="list-style-type: none"><li>– 波特率: 9600 bps</li><li>– <b>XTEA 成帧:</b> 使用 XTEA 加密, 开启半字节交换</li><li>– <b>AES 成帧:</b> 使用 AES 加密</li></ul></li><li>• <b>关闭加密</b><ul style="list-style-type: none"><li>– 波特率: 4800 bps</li><li>– <b>XTEA 成帧:</b> 未使用 XTEA 加密, 关闭半字节交换</li><li>– <b>AES 成帧:</b> 未使用 AES 加密</li></ul></li></ul>
	PP2	<p><b>选择 AES 成帧:</b> 使用<a href="#">第 6.1 章</a>所述的 AES 成帧。</p>

Table 1 软件配置和 GPIO/按钮分配

PMAfob 按钮	GPIO	软件配置
	PP4	选择 XTEA 成帧： 使用 <a href="#">第 6.2 章</a> 所述的 XTEA 成帧。
	PP3	选择用户自定义的唯一 ID： 使用用户自定义的唯一 ID (defines.h: #define USER_KEY_NR) 作为唯一密钥号。
	PP6	选择 PMA 唯一 ID 使用 PMA 唯一4个字节的ID 作为唯一密钥号。

## 参考资料

- [1] 《PMA51xx 函数库指南》

[www.infineon.com](http://www.infineon.com)

由 Infineon Technologies AG 出版