

LKT2102U 用户手册

仅授权给北京圆志科信电子科技有限公司开发使用，严禁复制

凌科芯安科技（北京）有限公司

版本记录

当前版本		V2.0	2016.5.25
原始版本		V1.0	2015.12
升级说明			
升级日期	版本号	新增内容	修改内容
2016.5.25		<p>1、第2章增加了6个开发步骤的说明。</p> <p>2、第4章与第5章更换位置，然后将第4章更名为算法移植编程指南，丰富了算法移植介绍和注意事项；第5章更名为LCS SAM 软件的使用及算法下载，5.2节中的方法二增加了在线下载的简要描述。</p>	<p>1、开发板、LCS SAMV2.0 软件以及P2000 等图片，进行了更新</p> <p>2、图 2-1 中的每个框图内容做了适当修改，把步骤六的批量下载改为了稳定性老化测试。</p> <p>3、删除 3.2.4 节中关于LCS SAM 软件提速的说明，改在LKT-K100 使用说明手册中进行描述，避免产生歧义</p>

联系凌科芯安

公司名称：凌科芯安科技（北京）有限公司

办公地点：北京市石景山区阜石路 166 号泽洋大厦 1601 室

电话：010 - 6886 4300

传真：010-68864300-604

仅授权给北京圆志科信电子科技有限公司开发使用，严禁复制

目 录

第 1 章 LKT2102U 简介.....	- 1 -
1.1 概述.....	- 1 -
1.2 LKT2102U 产品特性.....	- 2 -
1.2.1 硬件特性.....	- 2 -
1.2.2 系统软件特性.....	- 2 -
1.2.3 安全特性.....	- 2 -
1.3 应用领域.....	- 3 -
第 2 章 LKT2102U 开发流程.....	- 4 -
第 3 章 通讯协议说明.....	- 7 -
3.1 指令协议.....	- 7 -
3.2 A3 协议说明.....	- 7 -
3.2.1 指令格式.....	- 7 -
3.2.2 算法调用指令举例说明.....	- 7 -
3.3 ISO7816 T=0 协议说明.....	- 8 -
3.3.1 缩略语.....	- 8 -
3.3.2 指令格式.....	- 9 -
3.3.3 指令处理流程说明.....	- 10 -
3.3.4 算法调用指令举例说明.....	- 10 -
3.3.5 提速指令说明.....	- 11 -
第 4 章 算法移植编程指南.....	- 13 -
4.1 编译环境.....	- 13 -
4.2 算法移植.....	- 13 -
4.2.1 函数调用说明.....	- 14 -
4.2.2 算法例程中其他函数的功能简介.....	- 15 -
4.3 算法移植注意事项.....	- 16 -
4.3.1 全局变量的使用.....	- 16 -
4.3.2 局部变量的使用.....	- 16 -
4.4 算法编译、下载.....	- 16 -

4.5 算法调试.....	- 17 -
第 5 章 LCS SAM 软件使用.....	- 18 -
5.1 连接开发板.....	- 18 -
5.2 下载算法.....	- 19 -
5.3 修改下载保护口令.....	- 20 -
5.4 发送算法指令.....	- 20 -
5.5 批量测试算法指令.....	- 21 -
附录 A：系统函数说明.....	- 23 -
附录 B：批量生产工具.....	- 29 -

第 1 章 LKT2102U 简介

1.1 概述

LKT2102U 采用 32 位 EAL5+高安全等级智能卡芯片内核，芯片内部嵌入凌科芯安公司的 LKCOS 智能操作系统，用户可以把 MCU 中程序一部分关键算法函数移植到 LKT2102U 芯片中运行。采用标准 C 语言编写代码，可以用 ARM-MDK 编译器编译程序。在实际运行过程中，通过调用函数方式运行智能卡芯片内的程序段，并获得运行结果，并以此结果作为用户程序进一步运行的输入数据。LKT2102U 成为了产品的一部分，而算法在 LKT2102U 内部运算，盗版商无法破解，从根本上杜绝了程序被破解的可能。

MCU 程序分为两个部分：一部分是在 MCU 中，另一部分在 LKT2102U 中。当需要用到 LKT2102U 中的算法时 MCU 向 LKT2102U 发送指令，LKT2102U 根据指令在内部运行算法程序并返回结果给 MCU。

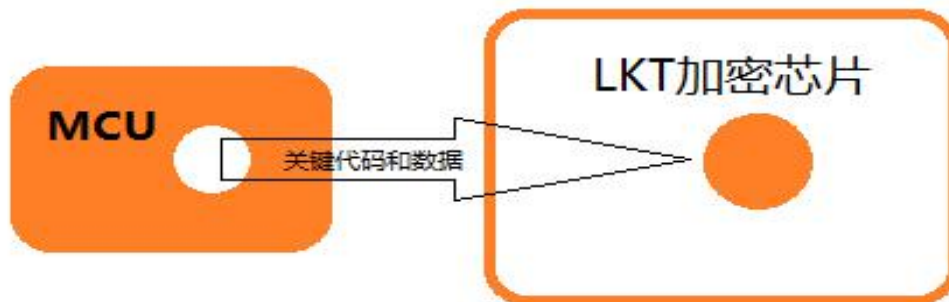


图 1-1：加密原理

1.2 LKT2102U 产品特性

- 使用 EAL5+最高安全等级的 32 位智能卡芯片为基础，具有极高的软硬件安全性
- 实现算法下载，用户可灵活实现自有知识产权的保护
- 标准 DIP8 或 SOP8 封装形式，另外还可为用户定制其他封装形式

1.2.1 硬件特性

- 采用 32 位智能卡芯片内核，内置 32 位保密操作系统
- 全球唯一硬件 ID 与管理编码
- 具有兼容 uart 的串口
- 支持 ISO7816 T=0 和自定义 A3 通讯协议
- 具有 20K 字节用户程序下载空间（可为客户定制容量）
- 4K 字节可定义安全性 NVM 数据存储区
- 1K 字节用户程序运行 RAM
- 编译环境具有丰富的系统调用和开发接口
- 3DES 协处理器

1.2.2 系统软件特性

- 自主知识产权的 COS 系统--LKCOS
- 片上操作系统(COS)进行通信、文件、存储、安全管理
- LKCOS 提供 16 级安全控制等级
- 支持用户程序下载
- 支持用户自己定义参数的输入，输出

1.2.3 安全特性

(1) 硬件防护措施

- 传感器（电压，时钟，温度，光照）
- 过滤器（防止尖峰/毛刺）
- 独立的内部时钟（独立CLK）
- （SFI）的检测机制

- 被动和主动盾牌
- 胶合逻辑（难以逆转工程师电路）
- 握手电路
- 高密度多层技术
- 具有金属屏蔽防护层，探测到外部攻击后内部数据自毁
- 总线和内存加密
- 虚拟地址（SW != 硬件地址）
- 芯片防篡改设计，唯一序列号
- 硬件错误检测
- 随机数发生器
- 噪音的产生（对边信道攻击）
- 预硅功率分析

(2) 软件 - 操作系统防护措施

- 内部数据不可读取、拷贝
- 敏感信息进行加密（钥匙，别针）
- 双重执行的（如加密解密核查）
- 校验
- 验证程序流
- 不能直接访问硬件平台
- 防止缓冲区溢出
- 防止错误的偏移
- 防火墙机制
- 异常计数器
- 执行验证码
- 归零的键和引脚

1.3 应用领域

控制器、安防监控、游戏机、汽车电子、平板电脑、机顶盒、DVR、路由器、交换机、仪器仪表等各种电子产品终端。

第 2 章 LKT2102U 开发流程

MCU 的源代码应被分成两部分，分别存储到 MCU 和 LKT2102U 内部。当程序运行时 MCU 通过发送算法命令调用 LKT2102U 的算法。开发流程分为 6 个步骤如图 2-1 所示：

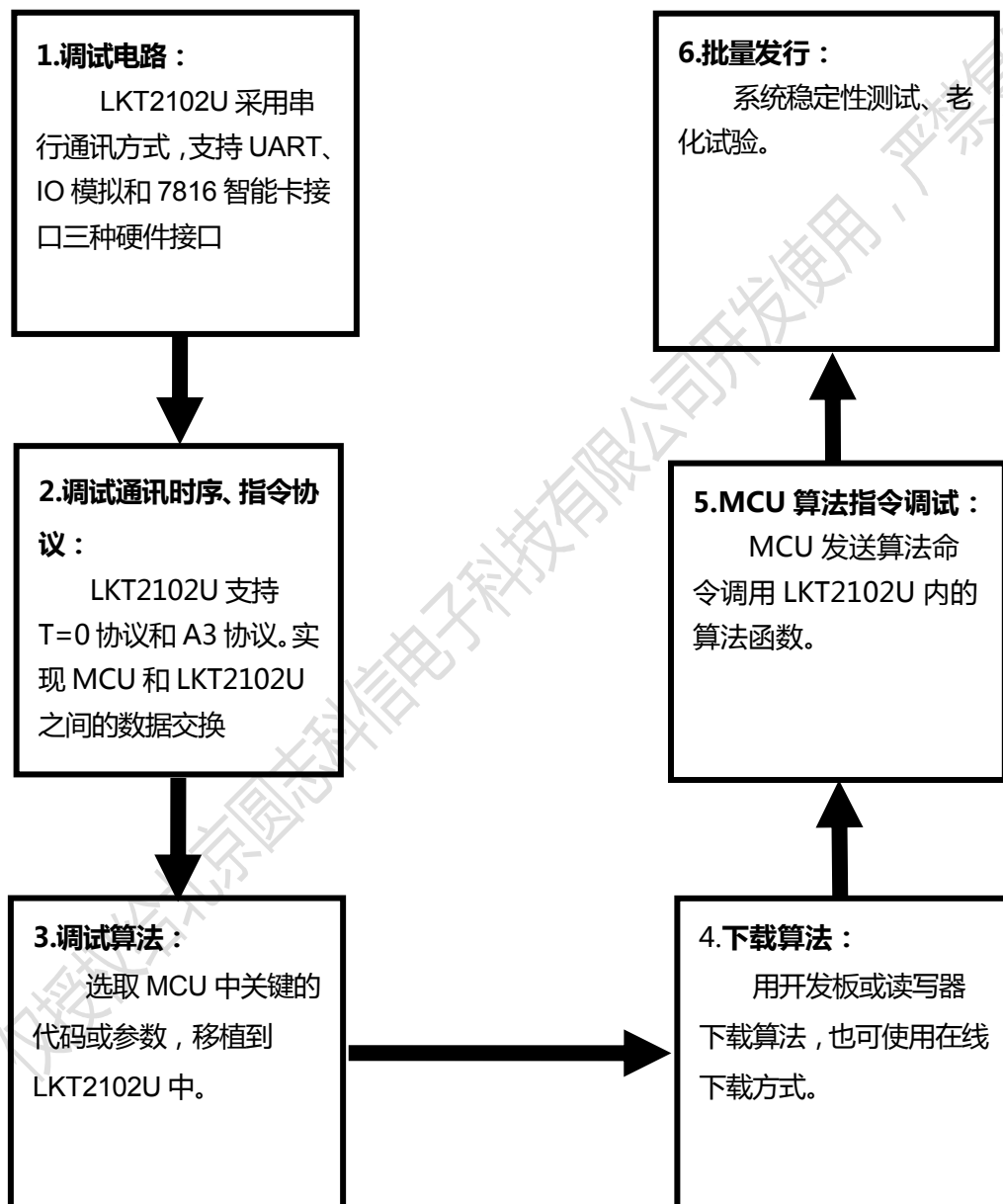


图 2-1：开发流程

步骤 1: 搭建通讯电路

LKT2102U 采用智能卡接口通讯，为单线半双工串行通道。LKT2102U 电气参数详见文档“LKT2102U DATASHEET”第 1、2 章。

用户有三种通讯电路选择。

a、通过外围电路，将智能卡接口转换为 UART 接口。有两种 UART 外围电路供参考设计。详见文档“LKT2102U DATA SHEET”中第 3 章参考电路 B、C。B 电路的优点是节约成本，减少板上空间占用，但使用该电路的用户请注意，MCU 的 TX 在发送数据时，RX 会同步接收，应对方法可有多种，可以忽略接收到的指令，也可在发送的时候，禁用 RX。C 电路利用与非门等器件，对 MCU 的 TX、RX 进行了隔离，所以收发互不影响，板上空间足够，成本压力不大的客户，建议使用该电路。

b、利用 MCU 的 IO 口模拟 UART 通讯。该方法适用于不带操作系统，或者中断源较少的 MCU，如果用户只在开机阶段与 LKT2102U 做少量交互，也可考虑使用该通讯接口。详见文档“LKT2102U DATA SHEET”中第 3 章参考电路 A。

c、可直接使用 MCU 的智能卡接口与 LKT2102U 的 IO 口进行对接。

LKT2102U 需要合适的外部才可正常运行。标准为 1~10Mhz，占空比为 40%~60%的方波。时钟的频率决定了通讯速率，通讯速率的计算方法详见第 3.3.5 节。时钟电路有三个参考，分别为 MCU 的 PWM 波、无源晶振+起震电路、有源晶振，详见文档“LKT2102U DATA SHEET”第 3 章。

步骤 2: 通讯调试

上电后至少做一次复位操作。复位和通讯时序请参考文档“LKT2102U DATA SHEET”第 4 章。另外，开发套件中还提供了串行通讯接口例程供参考。LKT2102U 支持 T=0 和 A3 两种通讯协议，协议内容可参考本手册第 3 章。

步骤 3: 算法移植

选取 MCU 中适合移植的关键代码，用 C 语言编程移植到 LKT2102U 算法例程中。详见本手册第 4 章。

步骤 4: 算法下载

详见本手册第 5 章

步骤 5: 算法调试

详见本手册第 5.5 节

步骤 6: 稳定性、老化测试

建议客户充分测试稳定性，并进行环境测试、老化测试。

仅授权给北京圆志科信电子科技有限公司开发使用，严禁复制

第 3 章 通讯协议说明

3.1 指令协议

通讯时序本章不做描述,详见“LKT2102U DATA SHEET” 文档第 4 章 Communication Debugging 部分。

LKT2102U 支持两种指令协议：A3 协议和 T=0 协议。区别在于命令格式不同。

- A3 协议命令头是“A3”，用 MCU 调试算法时，同时支持 T=0 和 A3 协议。
- T=0 协议命令头是“8008 0000”，LCS SAM 软件配合 LKT-K100 开发板调试算法时，同时支持 T=0 和 A3 协议。若使用智能卡读写器调试时，不支持 A3 协议。

注：指令全部是 16 进制。

3.2 A3 协议说明

A3 协议是我公司自定义的一种指令格式协议，其特点是交互流程简单。

3.2.1 指令格式

命令由命令头和命令体两部分构成如表3-1所示。其中命令体为 LV 格式，即命令体长度 L+命令体内容 V，L 的长度为1字节，范围0x01~0xFF，命令体内容最多为255字节。

命令头	命令体	
A3	Lc	DATA

表 3-1：A3 指令结构说明

3.2.2 算法调用指令举例说明

在测试之前 LKT2102U 下载好算法。现在以“LKT2102U 算法例程\mdk4\AppDemo”中的“fun_1”函数为例子。

使用 MCU 或 LCS SAM 软件向 LKT2102U 发送调用算法命令的流程如图 3-1 所示。



图 3-1：算法命令操作流程

指令结构说明如表 3-2 所示。

命令头	LC	算法函数序号	传入算法函数中的参数
A3	09	01	0102030405060708

表 3-2 : 算法指令结构说明

返回数据结构如表 3-3 所示。

命令头	后续数据长度	算法返回的数据	状态码 (SW)
A3	0A	FEFD FCFB FAF9 F8F7	9000

表 3-3 : 返回数据结构

3.3 ISO7816 T=0 协议说明

T=0 协议基于智能卡 7816 标准，该协议交互流程复杂。

3.3.1 缩略语

APDU	应用协议数据单元
ATR	复位应答
CLA	类字节
CLK	时钟信号
GND	地，基准电压。
INS	指令字节
I/O	串行数据的输入/输出
Lc	在指令中发送的字节长度
Le	接收响应数据的字节长度
P1	参数1
P2	参数2
SW1	状态字节1
SW2	状态字节2
VCC	电源输入

表 3-4 : 缩略语

3.3.2 指令格式

命令由命令头和命令体两部分构成，如表3-5所示：

命令头				命令体		
CLA	INS	P1	P2	Lc	DATA	Le

表 3-5：指令结构

常见的指令构成形式，如表3-6所示：

形式	指令类型
CASE 1	CLA INS P1 P2
CASE 2	CLA INS P1 P2 Le
CASE 3	CLA INS P1 P2 Lc Data
CASE 4	CLA INS P1 P2 Lc Data Le

表 3-6：指令类型

LKT2102U 常用指令，如表3-7所示：

命令头				命令体			描述
CLA	INS	P1	P2	Lc	DATA	Le	
00	84	00	00	无	无	01-10	取随机数
80	08	00	00	XX	XX...XX	无	算法调用指令
00	C0	00	00	无	无	XX	取响应数

表 3-7：常用指令

返回状态码的具体意义，如表3-8所示：

SW1	SW2	意义
90	00	正确执行
61	XX	有 XX 字节数据返回
67	00	长度错误
69	85	使用条件不满足
6A	80	数据参数错误
6A	86	参数 P1,P2 错误
6D	00	INS 错误
6E	00	无效的 CLA
6F	00	数据无效

表 3-8：SW

3.3.3 指令处理流程说明

◆ 大于 5 字节的 APDU:

先发送前 5 字节：如果收到 INS，继续发送后续数据，并接收 SW；如果收到 6X/9X，那么只需再收一字节（X 表示 1~F 之间任意值）；如果收到 60，则继续接收，直至接收到 SW 为止。

◆ 等于 5 字节的 APDU:

发送指令后：如果收到 INS，则继续接收数据 + SW；如果收到 6X/9X（X 表示 1~F 之间任意值），那么只需再接收一字节数据；如果收到 60，则继续接收，直至接收到 SW 为止。

3.3.4 算法调用指令举例说明

使用 LCS SAM 软件和用 MCU 测试 T=0 协议的命令过程略有差别。LCS SAM 软件调用的是私有接口函数，接口函数对 INS 内部已经做了处理，所以不需要将指令分步发送，但是 MCU 必须按照标准格式分步发送。在测试之前先保证 LKT2102U 已经下载算法，下载方法详见第四章。现在以“LKT2102U 算法例程\LKT2102U_AppDemo”中的“fun_1”函数为例子进行说明。

使用 LCS SAM 向 LKT2102U 发送调用算法命令的流程如图 3-2 所示。

```

-> 80080000 09 01 0102030405060708  /**发送调用算法命令**/
<- 6108                                /**有 8 字节应答数据等待读出**/
-> 00C0000008                          /**发送获取数据命令**/
<< FEFD FCFB FAF9 F8F7 9000          /**取出返回的数据以及 SW 值**/
    
```

图 3-2：LCS SAM 执行过程

使用 MCU 向 LKT2102U 发送调用算法命令的流程如图 3-3 所示。

```

-> 80080000 09          /**发送命令头+LC**/
<- 08                  /**返回过程字节 INS**/
-> 01 0102030405060708  /**发送后续数据**/
<- 6108                /**有 8 字节应答数据等待读出**/
-> 00C0000008          /**发送获取数据命令**/
<- C0 FEFD FCFB FAF9 F8F7 9000 /**取出数据 (INS+算法返回数据 +SW )**/
    
```

图 3-3 : MCU 执行过程

指令结构说明如表 3-9 所示。

命令头	LC	算法函数序号	传入算法函数中的参数
8008 0000	09	01	0102030405060708

表 3-9 : 算法指令结构说明

3.3.5 提速指令说明

LKT2102U 支持通讯提速。MCU 对 LKT2102U 进行复位操作并接收完复位信息后，如果不向其发送任何指令，则 LKT2102U 使用缺省通信速率 $S = \text{外部时钟频率} / \text{默认分频系数}$ ，每位有效数据时间 $1\text{etu} = 1 / S$ 。其中，外部时钟频率由用户决定，支持 1M~10MHz，默认分频系数为 372。

如果给 LKT2102U 提供的外部时钟频率是 3.579MHz。则 $S = 3.579\text{MHz} / 372 \approx 9600\text{bps}$ 。 $1\text{etu} = 372 / 3.579(\text{M}) \approx 104\mu\text{s}$ 。

在外部时钟不变的前提下，用户可以通过修改分频系数来提速。提速指令见表 3-10。

提速指令	PPS 值 (修改后的分频系数)
FF10947B	64
FF10957A	32
FF109679	16

表 3-10 : PPS 值

操作流程：先对 LKT2102U 进行复位操作，使用缺省通讯速率接收完整复位信息，然后发送提速指令。当 LKT2102U 返回与提速指令一样的数据后，提速操作完成。用户必须采用提速后波特率进行通讯。注意：一旦进行复位操作或重新上电，分频系数重新恢复为

372，通讯速率又恢复为缺省速率，需要重新进行提速。下面实际举例说明完整提速流程。

假设外部时钟频率为 3.579MHz，需要将通讯速率提升到 115200。交互流程如下图所示。

复位操作	/**复位操作**/
使用 9600bps 进行通讯	
<- 3B6D00004C4B917002201212170008021D	/**返回 17 字节复位信息,最后 8 字节为芯片唯一 ID 号,不会出现重复**/
-> FF109679	/**将分频系数修改为 16**/
<- FF109679	/**修改成功, 通讯速率提升为 115200bps**/
-> 使用 115200bps 进行通讯	

图 3-4：提速操作

第 4 章 算法移植编程指南

完整的加密方案是由 MCU 程序和 LKT2102U 程序共同组成的。算法移植就是在 LKT2102U 内部用 C 语言编程实现 MCU 中被移植代码的功能，编译成功后将生成的 HEX 文件下载到 LKT2102U 中，当 MCU 需要运行被移植的程序时，给 LKT2102U 发送指令来调用，然后接收运行结果。指令收发协议详见第 3 章。

注：HEX 文件就是被移植到 LKT2102U 内部的算法，关系到整个系统的安全，一定要妥善保管。

4.1 编译环境

LKT2102U采用采用32位 EAL5+高安全等级智能卡芯片内核。可使用美国Keil Software公司出品ARMMDK编译器开发编译。

编译器安装成功后，请直接打开我们提供的“**LKT2102U 算法例程**”工程进行算法移植，如图 4-1 所示,工程文件路径为\LKT2102U 算法例程\LKT2102U_AppDemo\mdk4\。工程文件中有详细注释，请按照注释要求进行调试。

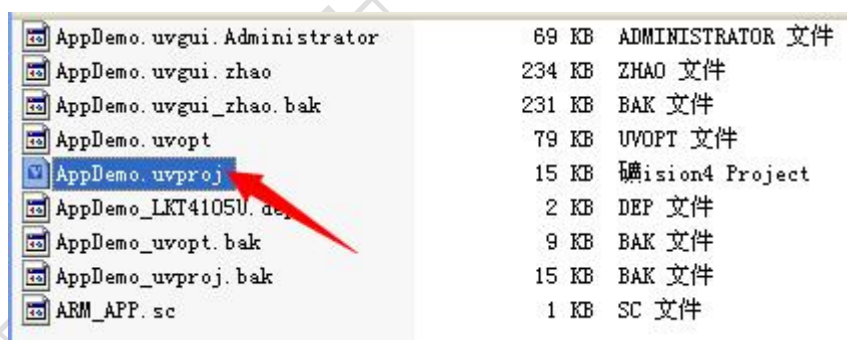


图 4-1 : AppDemo 工程文件

4.2 算法移植

算法工程中基础配置已经完成，不需要进行 device 型号选择。在“**APP_Main.c**”和“**APP_Fun.c**”这两个文件及其头文件中进行算法移植，其它文件禁止修改，否则可能会造成芯片工作异常。

4.2.1 函数调用说明

当芯片接收到指令后，先由底层系统对指令进行解析。若指令为 0084 0000 08 (随机数生成指令)，则直接生成 8 字节随机数后返回结果。若指令前四字节为 8008 0000，则调用算法例程中的函数 App_Command，同时将参数传入该函数中。下面以调用算法 1 对输入数据取反的例子来进行详细说明。(注：算法调用过程中输入输出数据均为十六进制)

举例：8008 0000 09 01 1122334455667788

该指令会调用 01 号算法，对输入的 8 字节数据 “1122334455667788” 取反后输出。
“8008 0000 09” 总共 5 字节是指令头 (T=0 协议)，其中第五字节 “09” 代表后续指令长度。

数据 “09” 传入 MainFile.c 文件 App_Command 函数的 LenOfIn 中，“01 1122334455667788” 会传入 MainFile.c 文件 App_Command 函数的 pInBuf 数组中，如图 4-2 所示。

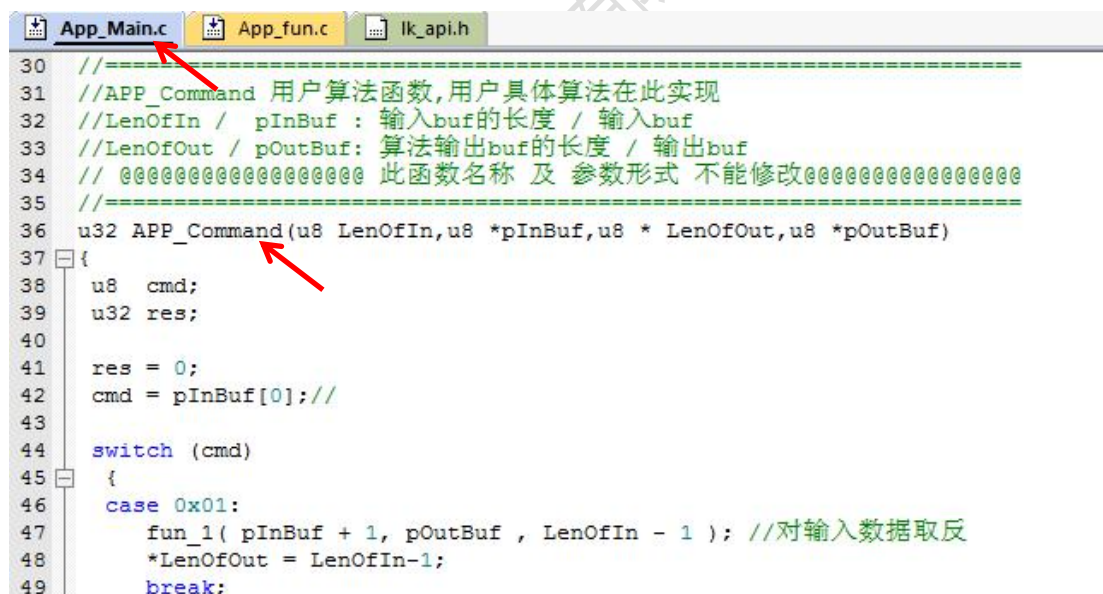


图 4-2：函数传参

输入参数 (由 MCU 发送给加密芯片)

LenOfIn：后续指令长度，即 0x09，最大值为 0xFF。

pInBuf[0]：调用函数的标识，即 0x01 号算法。

pInBuf[1..8]：8 字节输入数据，即 0x11~0x88。

输出结果（加密芯片调用内部算法后，将运算结果返回给 MCU）

LenofOut：输出数据长度，可由用户自行设置，最大值为 0xFF。

pOutBuf[0]~pOutBuf[8]：取反操作后输出的数据。

0x01 号算法 fun_1 的函数实现在 “App_Fun.c” 文件中，如图 4-3 所示。用户可在该文件中移植算法，或直接在 APP_FUNCTION 函数中进行移植。



```

109 //=====
110 //fun_1 对输入的若干字节取反
111 //=====
112 void fun_1(u8 *in,u8 *out,u8 len)
113 {
114     u32 i;
115
116     for(i=0;i < len;i++)
117         out[i]= ~in[i];
118 }
119
120

```

图 4-3：函数实现

综上所述，pInBuf 是输入缓冲区，LenofIn 是输入数据长度，pInBuf[0~N]对应输入数据。输入缓冲区内的数据都是通过指令被动接收。pOutBuf 是输出缓冲区，LenofOut 是输出数据长度，pOutBuf[0~N]是输出数据。输出缓冲区内的数据由用户根据移植的算法自行设定。当被调用的算法执行完毕后，LKT2102U 会根据输出缓冲区的内容，自动将数据通过 IO 口发送给 MCU。

4.2.2 算法例程中其他函数的功能简介

fun_2 功能：获取随机数（长度 0x00~0xFF）

fun_4 功能：写数据到 NVM 区（掉电不擦除）

fun_5 功能：使用存入到 NVM 区的 3DES 密钥对数据加密

fun_6 功能：使用存入到 NVM 区的 3DES 密钥对数据解密

fun_7 功能：使用存入到 NVM 区的 AES 密钥对数据加密

fun_8 功能：使用存入到 NVM 区的 AES 密钥对数据解密

fun_9 功能：对输入数据做 HASH 运算

4.3 算法移植注意事项

4.3.1 全局变量的使用

(1) 定义全局变量：要在全局变量初始化函数里进行初始化。

(2) 如果觉得定义全局变量都要到初始化函数里进行初始化，而数据量很大，可以进行数据搬运，即定义一个 const 类型数组 1，再定义一个非 const 类型数组 2，并在初始化函数里用 memcpy 函数把数组 1 拷贝到数组 2 里进而实现数据搬运。

搬运举例：

```
unsigned char const temp_key[0x09]={0x08,0x11,0x11,0x11,0x11,0x11,0x11,0x11,0x11};
unsigned char key[0x09];
//APP_INIT

// 初始化函数，如果有全局变量的话，请在此函数里做初始化

//=====================================================
void APP_INIT(void)
{
    //此处演示如何将全局变量初始化
    memcpy(key,temp_key,0x09);
}
```

4.3.2 局部变量的使用

定义局部变量：不能直接赋值，用 for 循环逐个定义或者逐个定义数组元素。

4.4 算法编译、下载

1. 在 Keil 编译环境中打开 “Project->Options for Target” 在对话框中的 “Output” 选项页勾选 “Creat HEX File” 项，如图 4-5 所示。

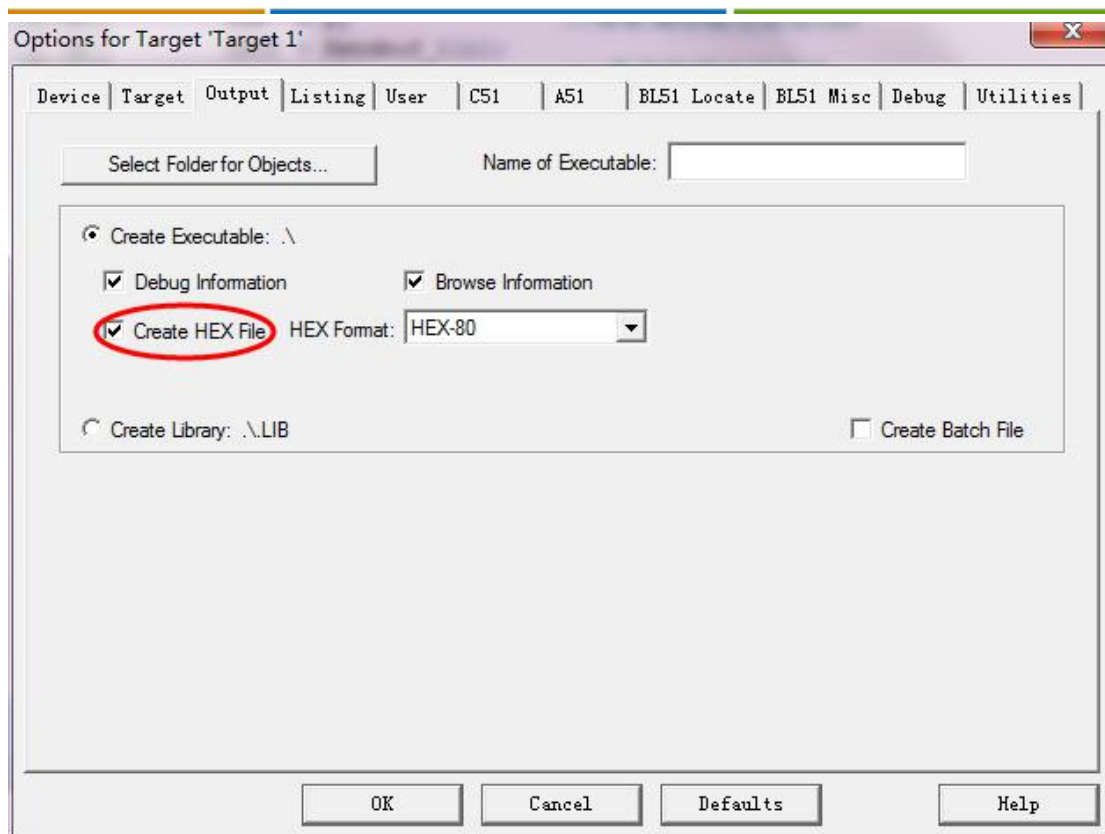



图 4-5 : 生成 hex 文件选项

2. 编译算法工程文件。点击 “Project->Build target files” 或直接点击编译按钮  , 编译算法例程，编译无误后即可在 “\LKT2102U 算法例程\out” 路径下生成 hex 文件。
3. 下载算法。（参照 5.2 节进行操作）

4.5 算法调试

LKT2102U 无法使用在线调试。因此，客户调试移植到 LKT2102U 内部的算法时，只能先将算法下载到 LKT2102U 中，然后通过指令调用分析输出结果。调试复杂函数时，建议用户将其拆分成多个函数，将每个函数的运算结果输出分析，定位问题。

第 5 章 LCS SAM 软件使用

5.1 连接开发板

LKT2102U 芯片放入 SOP8 的转接座（芯片的凹点或白点与图 5-1 中红圈对应）。将开发板与 PC 连接。

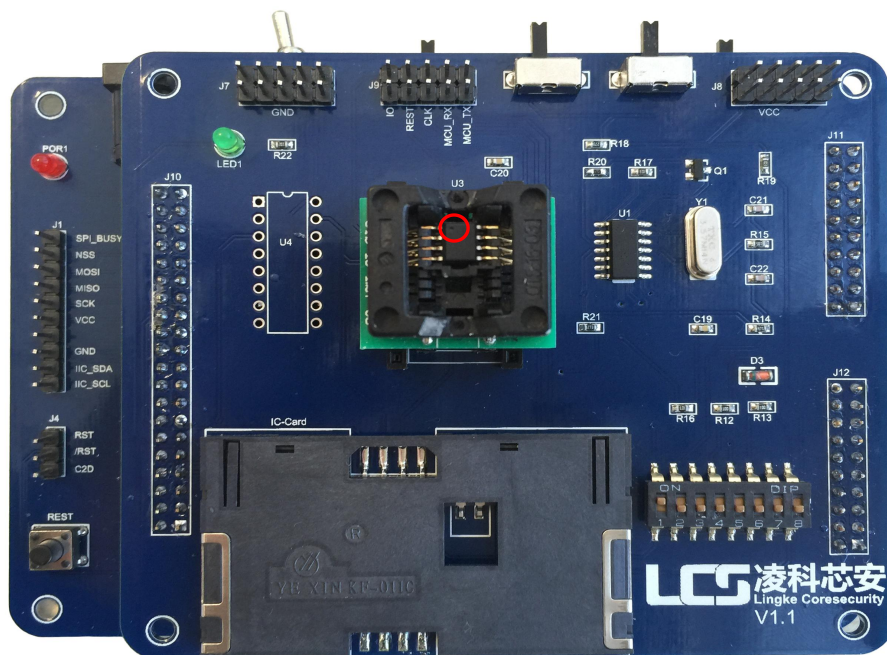


图 5-1 连接芯片

打开 LCS SAM 软件，如图 5-2 所示。

1. 点击“设备通信”选项页，选择“HID”通信方式（默认）。
2. 在通信时钟输入框内，输入提供加密芯片的时钟频率（范围：1~5Mhz，默认 3.579MHz）。
3. 点击“连接”按钮，会显示当前的连接状态、时钟频率和波特率。



图 5-2：连接开发板

5.2 下载算法

方法一：通过 LCS SAM 软件配合 LKT-K100 开发板对加密芯片进行算法下载，然后贴片测试。

方法二：先贴片，然后通过 MCU 进行在线下载算法。在线下载又分为明文下载和密文下载两种方式。用户可以使用该方式灵活升级 LKT2102U 芯片中的算法，保证产品的不断更新完善。客户若想进行远程更新算法，可以采用密文下载方式，将算法 hex 文件转换成密文格式在线传输给 MCU，后者将密文指令转发给 LKT2102U 完成算法升级，该方法可有效防止线路跟踪，避免截获算法。

下面仅对方法一进行说明。如需在线下载算法，请与凌科芯安技术支持人员联系，获取在线下载算法说明文档。

1. 点击“算法下载”选项页。
2. 在“旧口令”中填写下载口令，默认下载口令为“0000000000000000”（口令长度必须为 8 字节）。
3. 点击“打开文件”按钮，选择目标 hex 文件。
4. 点击“算法下载”按钮下载算法，如图 5-3 所示。

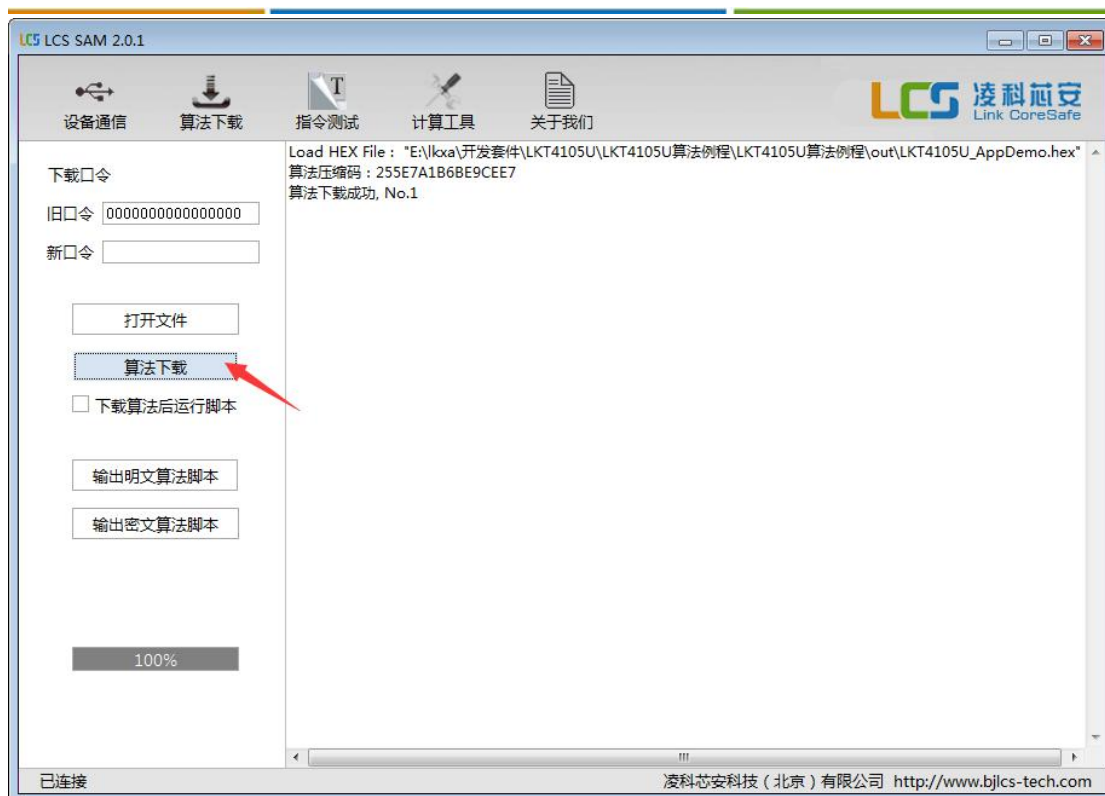


图 5-3 : 下载算法

5.3 修改下载保护口令

1. 在“算法下载”选项页中的“旧口令”输入框内，填入当前使用的下载口令，在“新口令”中，填入修改后的下载口令(口令长度必须为8字节)。

2. 点击“算法下载”，算法下载成功后完成修改。

修改下载口令后，该芯片只能用新口令下载算法，新口令与其它芯片无关（其他芯片默认口令仍为0000000000000000）。

5.4 发送算法指令

1. 点击“指令测试”选项页。
2. 在“测试指令”中输入算法指令。
3. 点击“单步运行”，如图 5-4 所示。

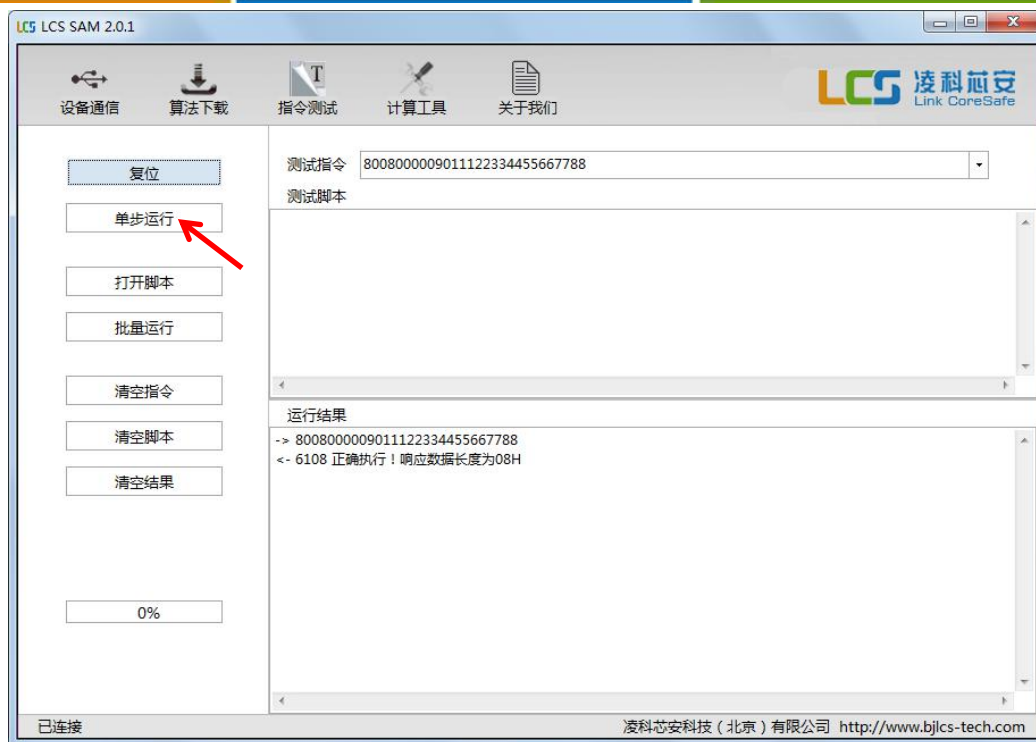


图 5-4：发送指令

5.5 批量测试算法指令

批量测试例程中的几个算法指令步骤如下：

- 1.在“指令测试”选项页中，点击“打开脚本”，选择脚本文件。
- 2.点击“批量运行”按钮，如图 5-5 所示。

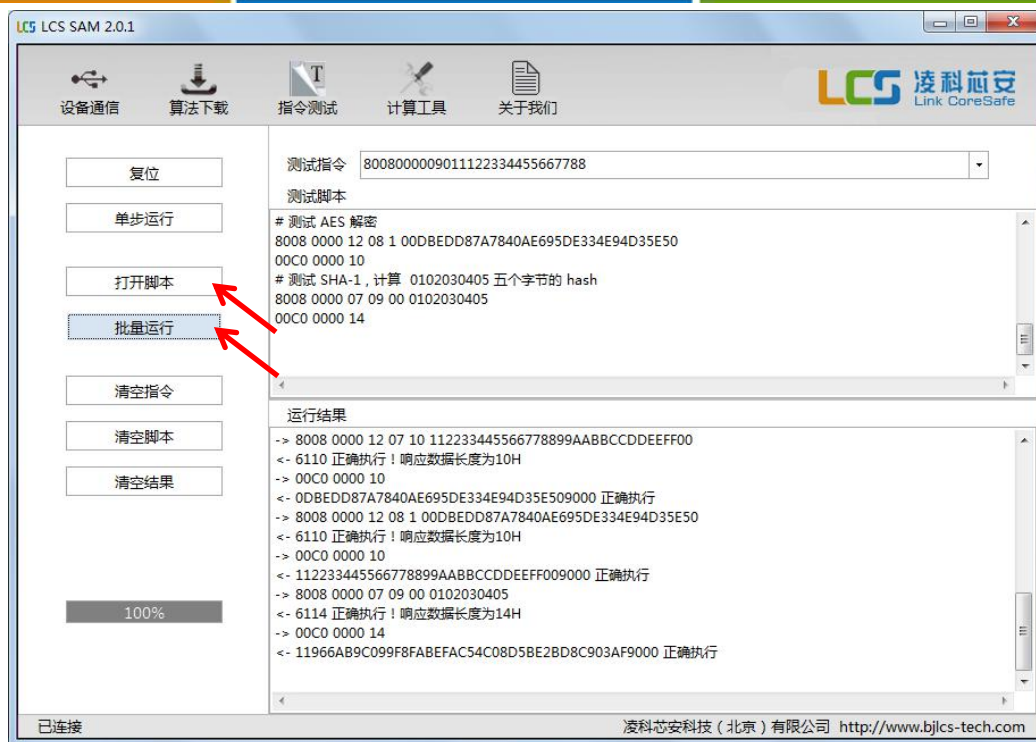


图 5-5 : 运行脚本

附录 A：系统函数说明

LKT2102U 提供 4K 字节的 NVM 数据存储区，从地址“0x0000”到“0x1000”。

写 NVM 区函数如表 A-1 所示。

函数描述	说明
函数形式	extern void LK_WriteNvm (unsigned short addr, unsigned char * buf , unsigned char len);
参数 1	NVM 区地址
参数 2	写入的数据
参数 3	写入数据的长度

表 A-1：写 NVM 区

读 NVM 区函数如表 A-2 所示。

函数描述	说明
函数形式	extern void LK_ReadNvm (unsigned short addr, unsigned char * buf , unsigned char len);
参数 1	NVM 区地址
参数 2	存放读出的数据
参数 3	读出数据的长度

表 A-2：读 NVM 区

DES/3DES 加密函数。注意这三个参数都是 LV 结构(数据长度+数据，如加密数据时 08 (长度) 1122334455667788(数据内容)) 如表 A-3 所示。

函数描述	说明
函数形式	extern void LK_DESEncrypt(unsigned char *plain, unsigned char *key,unsigned char *cipher);
参数 1	明文长度+明文内容
参数 2	密钥长度+密钥值
参数 3	输出的密文长度+密文值

表 A-3 : DES/3DES 加密

DES/3DES 解密函数。注意这三个参数都是 LV 结构(数据长度+数据 , 如解密数据时 08 (长度) 1122334455667788(数据内容)) 如表 A-4 所示。

函数描述	说明
函数形式	extern void LK_DESDecrypt(unsigned char *plain, unsigned char *key,unsigned char *cipher);
参数 1	需解密的密文长度+密文值
参数 2	密钥长度+密钥值
参数 3	解密后的明文长度+明文值

表 A-4 : DES/3DES 解密

获取随机数函数见表 A-5。

函数描述	说明
函数形式	extern void LK_GetRandom (unsigned char * buf, unsigned char len);
参数1	存放随机数据
参数 2	获取随机数的位数

表 A-5 : 获取随机数

获取芯片 ID 号函数见表 A-6。

函数描述	说明
函数形式	extern void LK_GetChipID(unsigned char *sn);
参数1	存放芯片 ID 号

表 A-6 : 获取芯片 ID 号

AES 加密函数。注意这三个参数都是 LV 结构(数据长度+数据 , 如加密数据时 08 (长度) 1122334455667788(数据内容)) 如表 A-7 所示。

函数描述	说明
函数形式	extern void LK_AESEncrypt(unsigned char *plain,

	unsigned char *key,unsigned char *cipher);
参数 1	明文长度+明文内容
参数 2	密钥长度+密钥值
参数 3	输出的密文长度+密文值

表 A-7 : AES 加密

AES 解密函数。注意这三个参数都是 LV 结构(数据长度+数据 ,如解密数据时 08(长度) 1122334455667788(数据内容)) 如表 A-8 所示。

函数描述	说明
函数形式	extern void LK_AESDecrypt(unsigned char *plain, unsigned char *key,unsigned char *cipher);
参数 1	需解密的密文长度+密文值
参数 2	密钥长度+密钥值
参数 3	解密后的明文长度+明文值

表 A-8 : AES 解密

AES 写入加密密钥函数如表 A-9 所示。

函数描述	说明
函数形式	extern void LK_AesSetKeyEnc(unsigned char *ByteLenOfKey, unsigned char *pKey, unsigned char *pRoundKey);
参数 1	密钥长度 (只能为 16、24、32 字节)
参数 2	密钥值
参数 3	用于运算的空间 (240 字节)

表 A-9 : AES 写入加密密钥

AES 写入解密密钥函数如表 A-10 所示。

函数描述	说明
------	----

函数形式	extern void LK_AesSetKeyDec(unsigned char *ByteLenOfKey, unsigned char *pKey, unsigned char *pRoundKey);
参数 1	密钥长度 (只能为 16、24、32 字节)
参数 2	密钥值
参数 3	用于运算的空间 (240 字节)

表 A-10 : AES 写入解密密钥

AES 加密函数。如表 A-11 所示。

函数描述	说明
函数形式	extern void LK_AesEncode(unsigned char *pIn, unsigned char *pOut);
参数 1	明文内容
参数 2	密文值

表 A-11 : AES 加密

AES 解密函数。如表 A-12 所示。

函数描述	说明
函数形式	extern void LK_AesDecode(unsigned char *pIn, unsigned char *pOut);
参数 1	密文值
参数 2	明文值

表 A-12 : AES 解密

DES 写入解密密钥函数如表 A-13 所示。

函数描述	说明
函数形式	extern void LK_DesSetKey(unsigned char ByteLenOfKey, unsigned char *pKey);
参数 1	密钥长度 (只能为 8、16 字节)

参数 2	密钥值
------	-----

表 A-13 : DES 写入解密密钥

DES/3DES 加密函数。如表 A-14 所示。

函数描述	说明
函数形式	extern void LK_DesEncode(unsigned char *pIn, unsigned char *pOut);
参数 1	明文值
参数 2	密文值

表 A-14 : DES 加密

DES/3DES 解密函数。如表 A-15 所示。

函数描述	说明
函数形式	extern void LK_DesDecode(unsigned char *pIn, unsigned char *pOut);
参数 1	密文值
参数 2	明文值

表 A-15 : DES 解密

HASH 摘要 初始化函数如表 A-16 所示。

函数描述	说明
函数形式	extern void LK_HashInit(unsigned char hashType);
参数 1	选择的算法 (0 表示 SHA-1, 1 表示 SHA-256)

表 A-16 : HASH 摘要 初始化

HASH 摘要 过程数据块输入函数如表 A-17 所示。

函数描述	说明
------	----

函数形式	extern void LK_HashUpdate(unsigned char hashType,unsigned char *buf,unsigned char len);
参数 1	选择的算法 (0 表示 SHA-1,1 表示 SHA-256)
参数 2	需要摘要的中间数据块
参数 3	数据块长度

表 A-17 : HASH 摘要 过程数据块输入

HASH 摘要 最后数据块输入函数如表 A-18 所示。

函数描述	说明
函数形式	extern void LK_HashLastUpdate(unsigned char hashType, unsigned char *buf, unsigned char len);
参数 1	选择的算法 (0 表示 SHA-1,1 表示 SHA-256)
参数 2	需要摘要的最后一个数据块
参数 3	数据块长度

表 A-18 : HASH 摘要 最后数据块输入

HASH 摘要 生成摘要值函数如表 A-19 所示。

函数描述	说明
函数形式	extern void LK_HashGetDigest(unsigned char hashType, unsigned char *digest);
参数 1	选择的算法 (0 表示 SHA-1,1 表示 SHA-256)
参数 2	摘要返回的结果

表 A-19 : HASH 摘要 生成摘要值

附录 B：批量生产工具

凌科芯安科技（北京）有限公司提供三款批量生产工具。

使用 LKT-K100 开发板下载算法如图 B-1 所示。

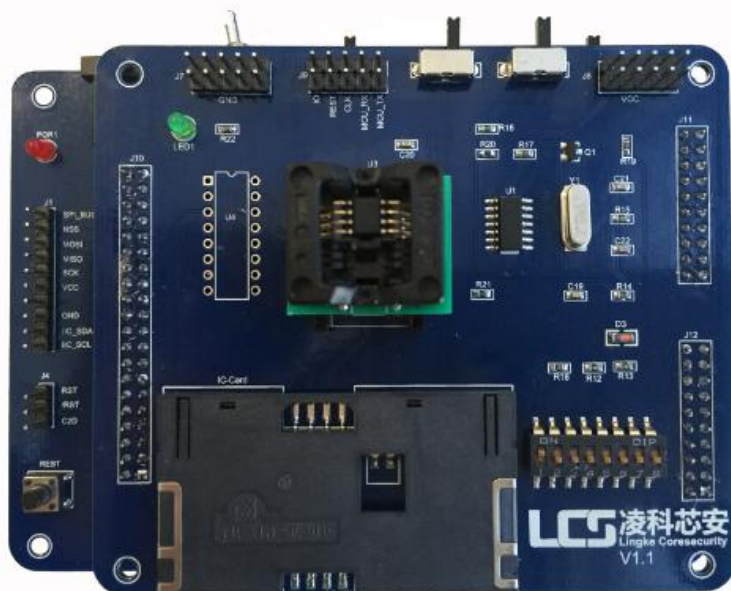


图 B-1：LKT-K100 开发板

脱机烧写器批量下载算法如图 B-2 所示。

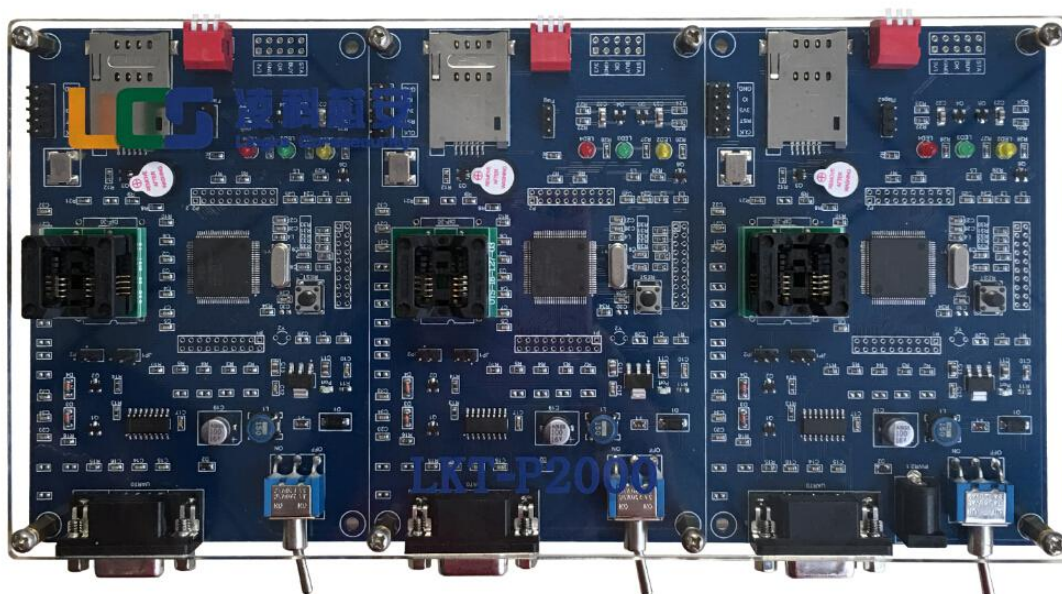
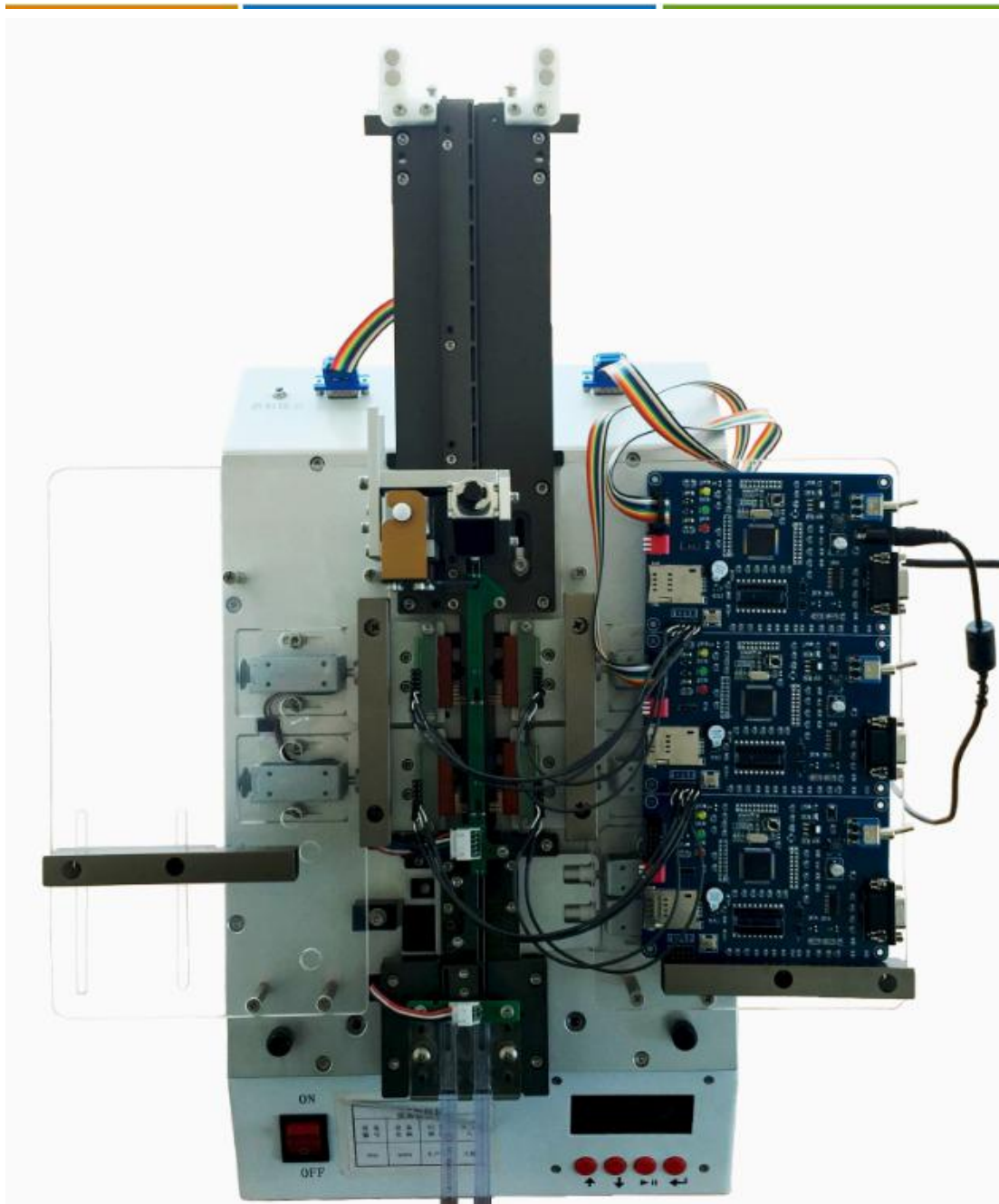


图 B-2：P2000 下载器

自动机械手烧录，烧录器和机械手相连后通过自动机械手自动烧录。见图 B-3。



图B-3 : 机械手