

LK2400A 开发板使用手册

法律声明

若接收常州千米电子科技有限公司（以下简称为“千米电子”）的此份文档，即表示您已经同意以下条款。若不同意以下条款，请停止使用本文档。

本文档版权归千米电子所有，保留任何未在本文档中明示授予的权利。文档中涉及千米电子的专有信息。未经千米电子事先书面许可，任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下，对此手册中描述的产品进行修改和改进的权利；同时保留随时修订或收回本手册的权利。

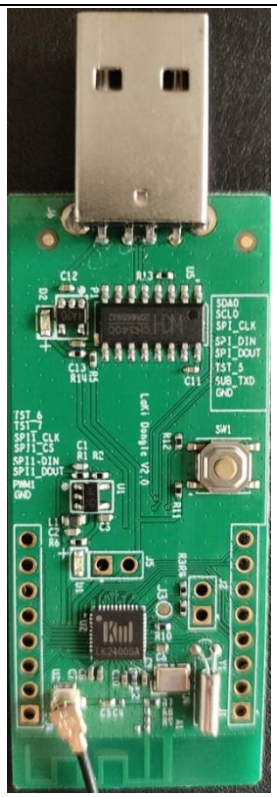
目录

1. 开发板简介 4
1.1 开发板展示 4
1.2 功能描述 4
1.3 操作流程和注意点 4
2. 简单评测 6
2.1. 初始操作 6
2.2. 更新固件 6
2.3. TX（21）/RX（22）通讯测试 6
2.4. 先收后发（“24”）和先发后收（“23”）的联合工作 8
2.5. 休眠-苏醒 9
2.6. 功耗测量 10
3. 常用 AT 命令 12
3.1. 注意点 12
3.2. 常用操作命令 12
4. 附注 15
2 PA Table 15
3 Channel Table 15
4 LaKi Module 16
4.3.1. KMS-2400A 16
4.3.2. KMS-2400B 17

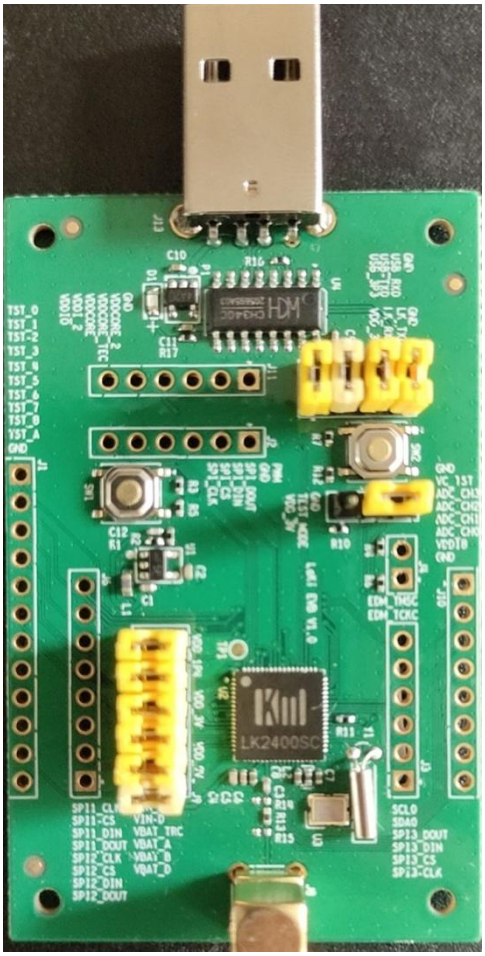
1. 开发板简介

1.1 开发板展示

千米提供 2 种不同类型的开发板提供给客户用于评测，如下图所示：



KMS 开发板



EVB 开发板

1.2 功能描述

- KMS 开发板，提供 USB 接口用于调试，同时除电源 LED 灯外，提供一个 LED 指示灯用于调试使用。 天线使用 IPEX 接口或者 PCB 天线。 同时提供各种外设接口用于客户调试
- EVB 开发板，提供 USB 接口。不提供用于调试的 LED 指示灯。提供各种电源插线，方便用户评测功耗

1.3 操作流程和注意点

- **操作方法：**
  - **有条件的话， 首先使用衰减器界定 RX sensitivity，做链路预算**
  - **强烈建议使用 2+米 USB 延长线连接开发板和电脑，使得板子稍稍远离电脑，测试时设置电脑为飞行模式**
  - **临近（1 米内）各种电子设备关闭 2.4Ghz 的无线，如电脑，手机，PAD 等，关闭 WIFI 和蓝牙**
  - **切勿使用天线延长线拉开距离， 除非用户对所用天线延长线质量有 100%的把握**

- 评测流程
  - 先进行远距离测试和穿墙测试，借此验证用户所使用的天线，各延长线及开发板的可靠性。 请以千米的测试作为参照，确保套件的可靠性
    - 远距测试：
      - ✓ 场地： 空旷可视无遮挡，测试时请检测一下空中信号，确保无超强的 2.4Ghz 信号，尤其是接收端附近无强信号干扰
      - ✓ 天线， 可靠 2dbi 胶棒天线。 若用户的天线增益更高或者更低，请参照下面的距离适当调整目标
      - ✓ 500 米， 收发两端高度约 1 米，基本不会出现掉包现象
      - ✓ 1000 米， 发射端约高度约 2 米， 接收端高度约 2 米， 基本不会丢包
      - ✓ 1500 米， 发射端高端约 3.5 米，接收端高度约 2.5 米， 有丢包现象，但能正常通讯
    - 穿墙测试：楼房上下楼穿楼板，确保是“穿”，而不是“钻缝”“饶射”等通讯
      - ✓ 穿办公楼（层高约 4.5 米，楼板厚度不详）： 3 层楼板可较高接通率通讯， 4 层楼板极低的接通率， 2 层楼板，极少的丢包
      - ✓ 住宅楼， 4 层楼板，有较高接通率， 5 层楼板接通率很低
  - 在完成远距和穿墙测试后， 用户可根据实际的应用场景进行评测

2. 简单评测

2.1. 初始操作

检测开办板是否正常

- 安装驱动程序（uart 转 usb）
- 电脑插上开发板，使用串口调试工具，**选中十六进制输入和显示**，打开串口。输入“00 01 00 90”回显如下
  - 01 00 47 54 软件版本号， FA EB DC 91， ID 号的第一个 word

```
[20:34:57.100]发→◇00 01 00 90 □
[20:34:57.103]收←◆01 00 47 54 FA EB DC 91 00 00 00 C0 08 00 00 00
```

- 或者输入“02 01 00 90”，在 KMS 开发板，LED 指示灯会闪烁 3 秒，后继续回显

```
|发→◇02 01 00 90 □
|收←◆01 00 47 54
|收←◆FA EB DC 91 00 00 00 C0 08 00 00 00
```

注意点：

- 通过 UART 进行 SerialDebug 工具操作，基于 word（32Bits）进行
- **使用串口工具时，每一个 word 的 4 个 Bytes 按照 LSB First 进行**，如上面的命令“02 01 00 90”实际是“90 00 01 02”。 后面所有的基于串口操作均为 LSB first

2.2. 更新固件

使用串口调试工具连接开发板之后，将千米提供的新的固件（文本文档），全部粘贴复制到串口调试工具窗口即可，写入完成后回显如下（回显不对，则写入没有成功）。刷新固件完成后，开发板会自动重启，等待 2 秒即可操作

```
[14:28:44.342]收←◆54 41 44 53
[14:28:44.553]收←◆00 00 00 00
```

2.3. TX（21）/RX（22）通讯测试

1. 准备
   - 准备两块开发板与电脑连接，并打开 uart 接口连接
   - 设定 A 板为 TX， B 板 RX：
     - TX 侧，脚本如下。先将要发送的数据通过 UART 接口写入到 BB_BUF. 输入第二个命令，等待操作。第二个命令为执行命令，点击一次，执行一次 TX

```
// Bit31~Bit25, cmd_word
  // 1st Step:  write TX data to bb_buf through uart. Cmd_word “91”
  // 2nd Step:  Start to TX。 {21 , frm_gap(4) , frm_cnt(4) , tx_pa(8) , tx_chn(8)

  // 将“1f2e3d084c5b6a79aabbccdd”写入 TX BUF
  // 帧间隙为“f”/自动重发 16 帧/pa 为 0c/2f 信道
  00000091 1f2e3d084c5b6a79aabbccdd
  2f0cf421
```

     - RX 侧，脚本如下. 工具窗口输入命令，等待操作。点击一次，执行一次 RX。完成一次接收的定义： 1.在设定的时间窗口期内接收到有效数据。2. 超过设定时间仍然未接收到数据。接收到有效数据通过 uart 接口打印。超时退出，系统将通过 uart 口打印“0x00000000”

```
//  {0x22 , pkt_size(8) , i_cnt_lit(4) , to_cnt_lit(3) , agc(1) , rx_chn(8)}
// summary：时间窗接收。时间长度为 to_cnt_lit(单位： 1.3S)， 若时间限制内未接收到任何数据，退出
//                                                          时间窗内接收到有效数据，退出

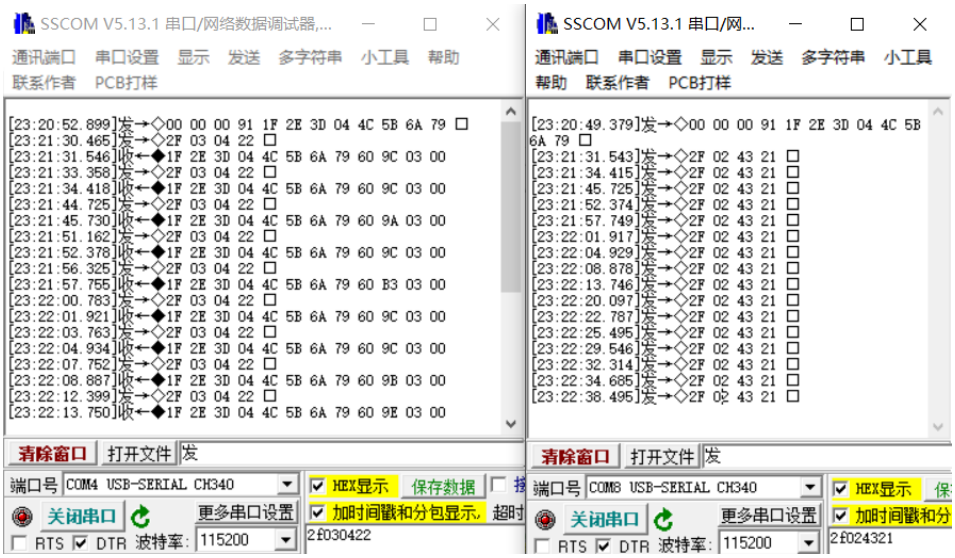
// "22", RX. Pkt_size="04", 预估 package 长度为 4 个 words,
// 注： 此长度必须至少比实际的接收的 package 长度大一个 word。
```

```
// RX 过程不仅仅存贮了 RX DATA，同时还存贮了 RSSI 的数值。
// UART 打印会同时将 RX DATA 和 RSSI+Gain 的数值打印
//  [ i_cnt_lit(4) , to_cnt_lit(3) ,agc(1)] = 03。i_cnt_lit=0, to_cnt_lit=1, agc=1.
// i_cnt_lit = 0, 单帧接收退出。 i_cnt_lit=1, 接收到两帧完全相同的帧退出…
//  rx_chn, 2f: 2.496.
// to_cnt_lit=1, 开窗时间约 1.3s

// 下面例子：预计 packet 长度为 4 个 word/单帧接收/时间窗为“1”（1.3s）/AGC 打开/ 2.496 信道
2f030422

// 下面例子：预计 packet 长度为 4 个 word/接收两帧同样数据结束/时间窗为“7”（9.1s）/AGC 关闭/ 2.496 信道
2f1e0422
```

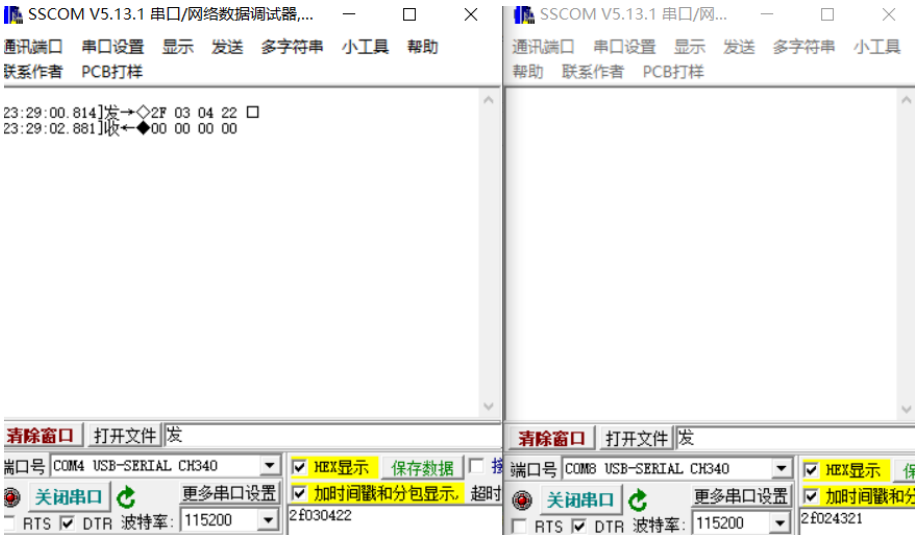
2. 一发一收
   - 先点击 RX 开发板让 RX 进入接收模式，然后点击 TX 板子发送一次数据， 查看接收端是否收到数据
   - 注意点：
     - TX/RX 设定在同一信道
     - 由于手动操作， 接收侧时间窗稍微开大点
     - 如果接收侧选择多帧判决， 则发射侧自动重发的次数必须大于等于接收侧设定的 i_cnt_lit
     - 根据距离，选择合理的 PA。测试最远距离，TX 中 PA 设定为最大值， RX 侧关闭 AGC
     - **执行 TX 前， 必须先在 TX Buffer 中填入需要发送的数据**
3. 图解如下：
   - TX. 首先将要传输的数据写入 TX Buffer（00000091 1f2e3d084c5b6a79aabbccdd）。然后将 TX 的命令（36043321）写入窗口，等待，如左图。 “21”为命令， 其余为配置信息
   - RX。连接好 uart，窗口写入接收命令（2f030c22），“22”为命令， 其余为配置信息. 等待.



TX。先写 TX buffer，准备好 TX 命令，等待

RX。备好 RX 命令，等待

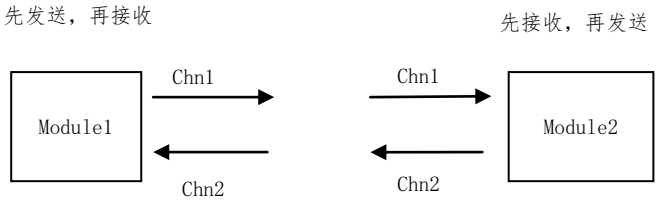
- 先在 RX 窗口点击开始接收，启动接收。在 TX 窗口点击发送数据。 RX 将接收到数据送到 uart 口并打印 RX 接收到 TX 发出的数据“1F2E3D044C5B6A79”和 RSSI+GAIN 的数值 - “609E0300 (00 03 9E 60)” RSSI[16:8], GAIN[7:0]. RSSI - “1 9E”, GAIN - “60”
- RX 等待接收，TX 不发射，RX 超时退出，并在 uart 打印“00 00 00 00”



2.4. 先收后发（“24”）和先发后收（“23”）的联合工作

2.4.1. 测试方法和流程

- 使用两个开发板， M1 先 TX 后 RX， M2 先 RX 后 TX



- 流程
  - 预先配置 M1/M2 的各自的 RX/TX 的参数/准备 M1/M2 需要发送的数据
  - 将各自的命令输入窗口等待
  - 先启动 M2， 然后启动 M1
    1. M2 启动后进入等待接收的状态
    2. 启动 M1 后先发送数据给 M2， M1 发送完成之后转入等待接收状态
    3. M2 收到数据后转入 TX 发送数据
    4. M1 接收到 M2 的数据
    5. 完成

2.4.2. 脚本

```
//////////////////////////////////////////////////////////
// rxAFTtx comunicat with txAFTrx at different channel
//////////////////////////////////////////////////////////

// Module1 acts rxAFTtx
// config module1
// module1 RX after TX
// RX {gap_t2r=3 (48us), pkt_size=04, (i_cnt_lit=1, to_cnt_lit=3(3.9s), agc=1)=17, rx_chn=36}
// TX {gap_r2t=3 (48us), (frm_gap=4, frm_cnt=4)=44, pa=0c, tx_chn=32}
001f509e 36170403
041f509e 320c4403

00000091 1a2b3c045b6d7e8f

 // cmd
001f5023

// module2 acts txAFTrx
// config module2
// RX {gap_t2r=255 (4ms), pkt_size=04, (i_cnt_lit=1, to_cnt_lit=3(3.9s), agc=1)=17, rx_chn=32}
// TX {gap_r2t=255 (4ms), (frm_gap=4(32us), frm_cnt=8)=44, pa=0c, tx_chn=36}

001f509e 321704ff
041f509e 360c44ff

00000091  1f2e3d045c6b7a89
```

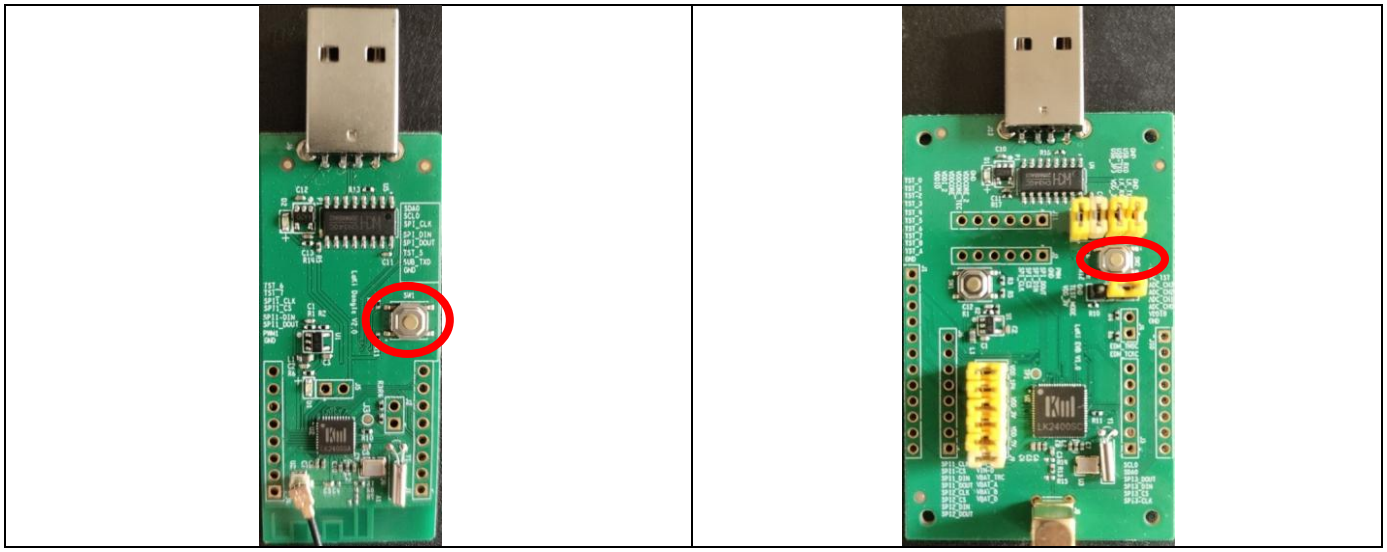
```
// cmd
001f5024

// Step:
// Start Module2 and then module1
```

2.5. 休眠-苏醒

休眠苏醒工作方式简介

- 提供 3 种休眠苏醒工作模式
  - 定时唤醒
  - 外中断唤醒
  - 定时或外中断唤醒
- 开发板上面提供外中断按钮，下图红圈处



- AT 命令（详见后面附解）：
  - “0x93，{sleep_time(24)}”，定时苏醒，时间为 sleep_time（unit 31.25us）或外中断唤醒。 在睡眠时间内如有外中断触发，直接苏醒，否则等睡眠时间到期苏醒

- 睡眠苏醒操作。可通过输入“90000000”查看是否有响应，或者通过观测电流来判断是否进入睡眠或者工作



Active， 芯片正常响应



进入睡眠，外部中断唤醒，命令字 94



进入睡眠，再次输入“90000000” 系统无响应

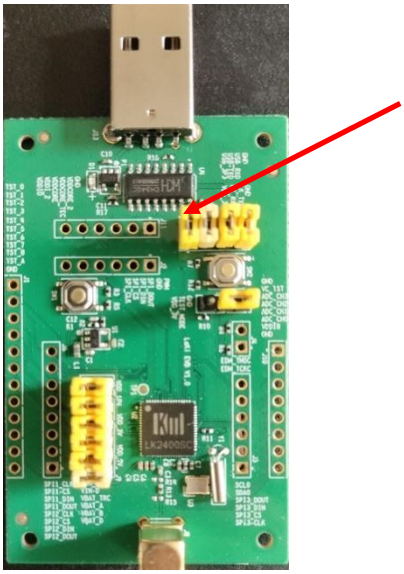


外部唤醒，输入“90000000”系统正常响应

2.6. 功耗测量

电流测量方式

- 在 EVB 开发板上进行功耗测量



上排跳线，左起第一个跳线为系统电源，测量功耗时， 拔去此跳线，电流表串入两根插针即可测量电流

3. 常用 AT 命令

3.1. 注意点

- 所有的数据和操作均基于 32Bits 操作
- 使用串口调试工具时，输入和输出均会将 1 个 word 中的 4 个 Bytes 按照 LSB First 的方式进行

3.2. 常用操作命令

- 命令格式

Bit31 ~ Bit24	Bit23 ~ Bit0
Command word	Parameters

- UART Command List

Command	Function Description
10	进入 UART R/W buffer debug mode
40	退出 UART R/W buffer debug mode
21	完成一次 TX
22	完成一次 RX
23	完成一次： RX after TX
24	完成一次： TX after RX
90	None， 查看 CPU 是否工作
91	将要发送的数据通过 uart 写入 TX Buffer
93	进入睡眠模式， timer 唤醒或外中断唤醒
9d	从一个 buffer 地址读取一个 word 的 data 并打印到 uart --- one word only
9e	将 uart 输入的一个 word data 写入一个 buffer 地址 --- one word only

- UART R/W buffer，主要包含下面三种：
  - Normal R/W， 3steps：｛（进入 R/W 模式），（R/W），（退出）｝,如下例

```
// 1. enterring uart mode. Bit31~Bit25, -- cmd word "10 00 00 00"
// 2. Start uart R/W buffer mode.
//            Bit31~Bit30 --- cmd word: "10" read, "11" write
//            Bit29~Bit24 --- Data length (unit: word, from "0")
//            Bit23~Bit0  --- Buffer Address (word align)
//            "c0", write a word to "301f00"
//            "80", read a word from "301f00"
//            uart closed after each operation

Step1:
/// enter uart dbg mode
00000010

Step2:
// W/R buffer
// 将 uart 口上的数据”0024102f” 写入到地址“0x301f00”
// 读入地址“0x301f00”的数据并打印到 uart 口
001f30c0 2f102400
001f3080

// write 3 words to "401000"。 将 uart 口上的数据（连续 3 个 word）“ffeedd08ccbbaa9911335577”写入到起始地址“401000”的 buffer 中去
// 从起始地址 "401000"连续读取 3 个 word 并打印到 uart 口
001040c2 ffeedd08ccbbaa9911335577
00104082

Step3:
// MUST Quit after ending R/W
// cmd word: 40 00 00 00
```

```
00000040
```

- One-word-only R/W. Cmd word: “9e” write, “9d” read
  {9e,{buffer_address}} {Data}; 通过 uart 将数据“Data”(one word)写入“buffer address”
  {9d,{buffer_address}} ; 通过 uart，将{buffer_address}的数据(one word)读出并送到 uart

```
////////////////////////////////////////////////////////
// R/W buffer through uart --- one word only in normal mode

// {9e, addr(24)} AABBCCDD
// put uart data “aabbccdd” to buffer "401000"
0010409e aabbccdd

// read buffer "40 10 00" and sent to uart port。 Cmd word “9d”
0010409d
```

- Special Write： 通过 uart 将数据写入 TX Buffer. Cmd word:” 91 00 00 00”

```
// 1st Step: prepare data to bb_buf.
// 此数据必须严格遵循 LAKI 数据帧格式，如下
//  Data format: Header1=1f, Header2=2e, Header3=3d, Length=08. Length 仅仅指 payload 的长度，单位（Byte），不包含 Header 的第一个 word
// payload "4c 5b 6a 79",  DATA in BB buffer:
//                         Word0: "08 3d 2e 1f"
//                         Word1: "79 6a 5b 4c "
//                         Word2: “dd cc bb aa”
00000091   1f2e3d084c5b6a79aabbccdd
```

■ RX/TX 命令

- TX， cmword ”21”

	Cmd word（8）	Frm_gap(4)	Frm_cnt(4)	Tx_pa(8)	Tx_chn（8）
Comment	21	自动重发时的帧间隙（单位： 8us）	自动重发的帧数（单位： 1 帧）	发射功率	发射信道
功能	发射一次 TX buffer 中的数据， 可选择自动重发/信道/功率				

- RX， Cmdword “22”

	Cmd word（8）	Packet_size(8)	I_cnt_lit(4)	To_cnt_lit（3）	AGC（1）	Rx_chn（8）
Comment	22	预估接收的包长度（单位: word）	接收多帧并进行多数判决时， 接收多少同样的数据认定为有效数据， “0”，单帧接收即退出	Timeout 的时间长度（单位， 1.3s）	0， off 1， on	RX channel 选择
功能	接收一次数据。在设定的时间窗内接收到数据立即退出，并将接收的数据打印到 uart。 若在时间窗内未接收到任何有效数据则超时退出，在 uart 口打印“00 00 00 00”. 若 i_cnt_lit!=0, 则自动连续接收，并对接收的数据采用多刷判决的原则， 当接收到 i_cnt_lit 个同样数据的时候，接收结束，并将此数据在 uart 打印. **注： 如 i_cnt_lit!=0, TX 侧必须自动重发，且重发次数 tx_frm_cnt >= i_cnt_lit**					

■ “90 00 00 {led flickertime}”,cmd_word: “90” 用于查看板子是否正常

```
// check board
00000090
// check board and led flick 2.6s
02000090
// check board and led flick 2.6s， and print one word chipid info
02010090
```

■ 休眠-苏醒

- {0x93, sleep_time(24)}, 进入休眠，设定休眠时间为 sleep_time（单位： 31.25us)。 到期苏醒或外中断触发唤醒
- 94 xx xx xx, 进入休眠，仅仅会被外中断触发唤醒

■ 复合操作： TX 后自动 RX， RX 后自动 TX

- {23, rf_para_addr(24)}. 在 TX channel 上发射一次数据，然后转到 RX channel 进行一次接收

1 RX 和 TX 的参数各使用一个 word 的 Buffer。 Rx_para_addr = rf_para_addr, tx_para_addr = rf_para_addr + 4

2 RX parameter format

Gap_t2r(8)	Packet_size(8)	I_cnt_lit(4)	To_cnt_lit（3）	AGC（1）	Rx_chn（8）
TX 转 RX 的时间间隔，unit：16us	预估接收的包长度（单位: word）	接收多帧并进行多数判决时，接收多少同样的数据认定为有效数据， “0”，单帧接收	Timeout 的时间长度（单位， 1.3s）	0 off 1 on	RX channel

3 TX parameter format

Gap_r2t（8）	Frm_gap(4)	Frm_cnt(4)	Tx_pa(8)	Tx_chn（8）

RX 转 TX 时间间隔，unit：16us	连续 TX 的帧间隙（单位： 32us）	自动重发的帧数（单位： 4 帧）	发射功率	TX channel

- {24, rf_para_addr（24)}.在 RX channel 完成一次接收，再在 TX channel 上完成一次发送
  1 Config 信息与“23”同
- 注：
  1 以上两种工作方式下，均须预先在 buffer 中写入 TX/RX 的配置信息
  2 Buffer 地址请 follow 例程中的地址： “0x501f00”
  3 Gap_r2t, gap_t2r 最小可以设置为“04”

4. 附注

2 PA Table

Reg(0x0)	Out (dBm)
01	-22
02	-16
03	-12
04	-10
05	-8

Reg(0x0)	Out (dBm)
06	-7
07	-6
08	-5
09	-4
0a	-3

Reg(0x0)	Out (dBm)
0b	-2
0c	-1
0e	0
10	1
12	1.7

Reg(0x0)	Out (dBm)
14	2.3
18	3.1
20	4
30	4.5
3f	4.8

3 Channel Table

RegValue (0x0)	Band (GHz)
00	2.402
01	2.404
02	2.406
03	2.408
04	2.410
05	2.412
06	2.414
07	2.416
08	2.418
09	2.420
0a	2.422
0b	2.424
0c	2.426
0d	2.428
0e	2.430
0f	2.432

RegValue (0x0)	Band (GHz)
10	2.434
11	2.436
12	2.438
13	2.440
14	2.442
15	2.444
16	2.446
17	2.448
18	2.450
19	2.452
1a	2.454
1b	2.456
1c	2.458
1d	2.460
1e	2.462
1f	2.464

RegValue (0x0)	Band (GHz)
20	2.466
21	2.468
22	2.470
23	2.472
24	2.474
25	2.476
26	2.478
27	2.480
28	2.482
29	2.484
2a	2.486
2b	2.488
2c	2.490
2d	2.492
2e	2.494
2f	2.496

RegValue (0x0)	Band (GHz)
30	2.498
31	2.500
32	2.502
33	2.504
34	2.506
35	2.508
36	2.510
37	2.512
38	2.514
39	2.516
3a	2.518
3b	2.520
3c	2.522
3d	2.524
3e	2.526
3f	2.528

4 LaKi Module

4.3.1. KMS-2400A



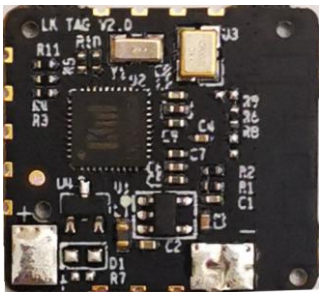
Description

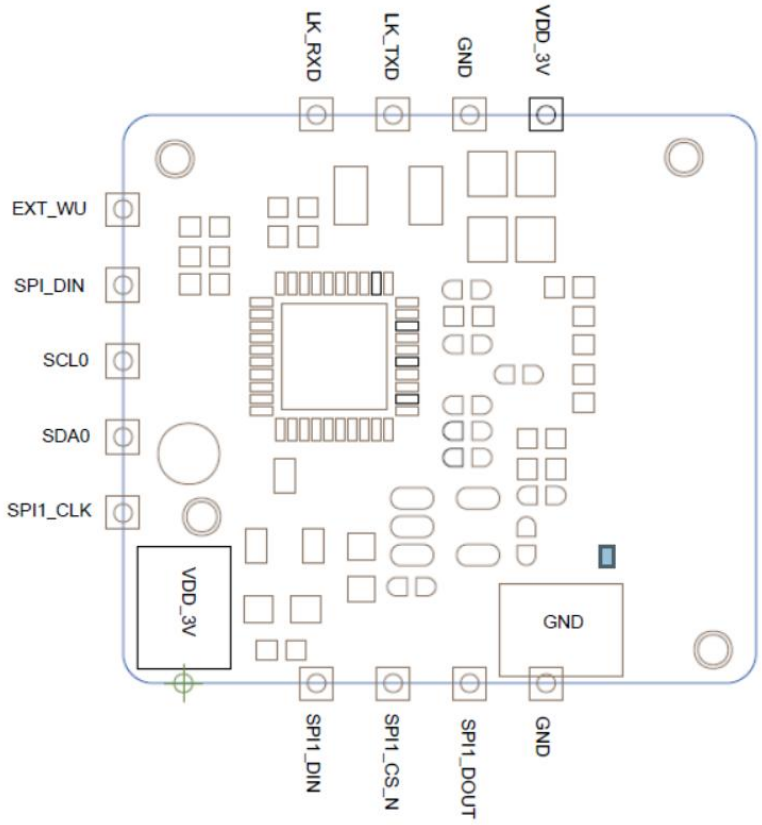
- Circle, Diameter ~2.5cm
- Function
  - Simple hardware only provide a LED for indication
  - PCB antenna, ~ -1 dBi

PIN Description

Pin Name	Function	Description
VDD3V	Power Supply	2.2V ~ 3.6V
GND	Ground	0V
LK_TXD	TXD of UART	Reference Pin28 of datasheet
LK_RXD	RXD of UART	Reference Pin29 of datasheet
J1	SPI_din	Reference Pin38 of datasheet
J3	EDM_TMSC	Reference Pin31 of datasheet
J4	EDM_TCKC	Reference Pin30 of datasheet

4.3.2. KMS-2400B





Description

- Size: 1.9cmX1.9cm
- Function
  - Simple hardware with multi IO resource
  - PCB antenna, 0~2dbi