

LARK-1 Modbus 通信协议

本公司生产的 LARK-1 系列传感器通过 RS232 串行通讯接口提供命令供外部设备操作。为了加快客户使用 LARK-1 的开发速度，本公司开发了 RS232 转 RS485 Modbus 转换板，该转换板 RS232 端接 LARK-1，Modbus 端供外部设备使用。这篇笔记将描述如何用该转换板通过 Modbus 协议操作 LARK-1，包括信息查询、数据采集、ZERO 和 SPAN 标定等功能。

该转换板在 Modbus 网络以 RTU 模式通信，且仅作为从设备，设备地址可见转换板反面的标签，如图 1 所示。

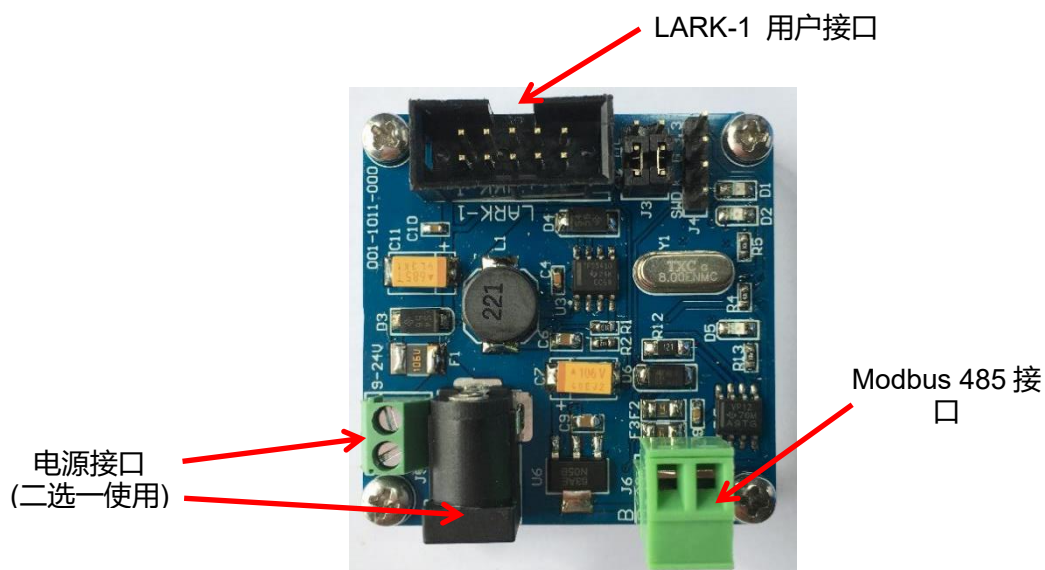


图 1. Modbus 转换板

- Modbus 说明

- Modbus 接口说明

表 1. Modbus 接口说明

序号	技术规格	规定
1	物理接口	RS485 半双工
2	波特率	9600
3	传输方式	RTU(远程终端单元) 格式
4	数据格式	见下文
5	数据数量	< 255
6	校验方式	CRC-16/Modbus (多项式为 $x^{16}+x^{15}+x^2+1$)
7	字节格式	1 起始位 + 8 数据位 + 1 停止位, 无校验
8	广播地址	0
9	接口定义	A(+), B(-)

➤ Modbus RTU 指令说明

Modbus RTU 指令由起始符、设备地址(1 字节)、功能码(1 字节)、数据区、CRC 校验码(2 字节)和结束符组成。

表 2. Modbus RTU 指令格式

起始符	设备地址	功能码	数据区	CRC 校验码	结束符
T1-T2-T3-T4	8Bit	8Bit	N 个 8Bit	16Bit	T1-T2-T3-T4

起始符和结束符的 T1-T2-T3-T4 表示 4 个字符时间的停顿间隔，使用 RTU 模式，消息发送要以至少 3.5 个字符时间的停顿间隔开始和结束。

设备地址可以为 0 ~ 247，0 为广播地址，Modbus 协议可以接 247 个从机，地址范围为 1 ~ 247，但是 RS485 接口在没有中继情况下，最多可以接 32 个从机。

该转换板支持 0x04, 0x06, 0x10 三种功能码，当用户使用其他功能码操作时会返回异常码：非法功能码。

➤ 功能码说明

表 3. 支持的功能码说明

功能码	功能	寄存器地址	
0x04	读寄存器数据	0x2000~0x22FF	
0x06	写单个寄存器的数据	0x1000~0x100D	
0x10	写多个寄存器的数据	0x1000~0x100D	

功能码 0x04：读寄存器数据，对应地址范围为：0x2000~0x22FF

请求的指令格式为：

设备地址	功能码	起始地址 高字节	起始地址 低字节	寄存器个数 高字节	寄存器个数 低字节	CRC 校验码 低字节	CRC 校验码 高字节
0x01	0x04	0x20	0x00	0x00	0x02	0x7A	0x0B

响应的指令格式为：

设备地址	功能码	字节个数	数据 1	数据 2	数据 3	数据 4	CRC 校验码 低字节	CRC 校验码 高字节
0x01	0x04	0x04	0x00	0x00	0xB1	0x7E	0x0F	0xF4

功能码 0x06：写单个寄存器的数据，对应地址范围为：0x1000~0x100D

请求和响应的指令格式都为：

设备地址	功能码	起始地址 高字节	起始地址 低字节	写入数据 高字节	写入数据 低字节	CRC 校验码 低字节	CRC 校验码 高字节
0x01	0x06	0x10	0x0B	0xFF	0xFF	0xFD	0x78

功能码 0x10: 写多个寄存器的数据，对应地址范围为：0x1000~0x100D

请求的指令格式为：

设备地址	功能码	起始地址 高字节	起始地址 低字节	寄存器 个数高 字节	寄存器 个数低 字节	字节 数	数据 1	数据 2	数据 3	数据 4	CRC 校验码 低字节	CRC 校验码 高字节
0x01	0x10	0x10	0x01	0x00	0x02	0x04	0x00	0x00	0xC3	0x50	0xAF	0x6F

响应的指令格式为：

设备地址	功能码	起始地址 高字节	起始地址 低字节	寄存器个数 高字节	寄存器个数 低字节	CRC 校验码 低字节	CRC 校验码 高字节
0x01	0x10	0x10	0x01	0x00	0x02	0x14	0xC8

非法功能码：

Modbus 操作异常返回异常码格式为：

设备地址	功能码	错误码	CRC 校验码低字节	CRC 校验码高字节
0x01	请求功能码+0x80: 0x84 / 0x86 / 0x90	0x01: 不支持的功能码 0x02: 寄存器地址错误 0x03: 寄存器数量错误 0x04: 数据错误	0xXX	0xXX

● 寄存器信息

表 4. LARK-1 传感器信息寄存器(只读)

地址	长度 (字节)	名称	描述	读/写 R/W	类型
0x2100	2	Bit Map 版本 Bit Map Version	A, B, C, ... '00 01' for the 'A'	R	HEX UINT16 Big Endian
0x2101	2	传感器类别 Sensor Type ID	见附录 1 '00 01' for the 'NDIR'	R	HEX UINT16 Big Endian
0x2102	4	传感器气种 Sensor Sub ID	见附录 2 '00 00 00 01' for the 'CH4'	R	HEX UINT32 Big Endian
0x2104	10	气种名称 Gas Name	字符串, 比如'C2H5OH, C6H5CH3' '20 20 20 20 20 20 20 43 48 34' for the 'CH4'	R	ASCII
0x2109	12	传感器序列号 S.N.	第 1 位: 硬件版本; 第 2~3 位: 固件版本; 第 4~8 位: 流水码; 第 9~12 位: 年月 YYYYMM '34 30 32 30 30 30 34 35 31 37 30 39' for the '402000451709'	R	ASCII
0x210F	8	出厂日期 Produce Date	YYYYMMDD '32 30 31 37 30 39 33 30' for the '2017.09.30'	R	ASCII
0x2113	8	有效日期 Warranty Expire Data	YYYYMMDD '32 30 31 39 30 33 33 30' for the '2019.03.30'	R	ASCII

0x2117	2	读数单位 Reading Unit Code	见附录 3 '04' for the 'ppm'	R	HEX UINT16 Big Endian
0x2118	6	单位名称 Reading Unit Name	字符串, 比如'PPM', '%vol', '%mbar' '20 20 20 50 50 4D' for the ' PPM'	R	ASCII
0x211B	4	高精度量程 Range 1	高精度读数范围 '00 00 C3 50' for the '50000'	R	HEX UINT32 Big Endian
0x211D	4	低精度量程 Range 2	低精度读数范围 '00 00 C3 50' for the '50000'	R	HEX UINT32 Big Endian
0x211F	4	报警下限 Alarm Limit 1	报警下限值 '00 00 01 F4' for the '500'	R	HEX UINT32 Big Endian
0x2121	4	报警上限 Alarm Limit 2	报警上限值 '00 00 C3 50' for the '50000'	R	HEX UINT32 Big Endian
0x2123	4	零点漂移阈值 Drift Limit	零点漂移超过该值则不允许执行零点标定 '00 00 27 10' for the '10000'	R	HEX UINT32 Big Endian
0x2125	4	最小标定点 Minimum Calibration Value	做 SPAN 标定时允许标定的最小浓度, 默认取量程的 25% '00 00 30 D4' for the '12500'	R	HEX UINT32 Big Endian

表 5. LARK-1 传感器数据采集寄存器(只读)

地址	长度 (字节)	名称	描述	读/写 R/W	类型
0x2000	4	读数 Reading	传感器测得的浓度值 '00 00 C3 50' for the '50000'	R	HEX INT32 Big Endian
0x200A	4	负漂补偿后的读数 Negative Drift Compensation Reading	传感器测得的经过负漂补偿的浓度值 '00 00 C3 50' for the '50000'	R	HEX INT32 Big Endian
0x2002	2	探测器温度 Det_Temp	探测器温度(0.01K) '72 74' for the '29300'	R	HEX UINT16 Big Endian
0x2003	2	光源温度 Lamp_Temp	光源温度(0.01K) '72 74' for the '29300'	R	HEX UINT16 Big Endian
0x2004	4	Sig_CTS	探测器 Sig 信号峰峰值 '00 03 22 BC' for the '205500'	R	HEX UINT32 Big Endian
0x2006	4	Ref_CTS	探测器 Ref 信号峰峰值 '00 02 5F 6C' for the '155500'	R	HEX UINT32 Big Endian
0x2008	2	空气气压 Air Pressure	环境大气压	R	HEX UINT16 Big Endian
0x2009	2	待测气压 Gas Pressure	通入传感器的待测气压	R	HEX UINT16 Big Endian

表 6. LARK-1 传感器操作状态寄存器(只读)

地址	长度 (字节)	名称	描述	读/写 R/W	类型
0x2200	2	Zero 标定记录 操作状态	0x0000: Zero 点标定记录成功 0x0001: LARK-1 未稳定(Ref 信号为 0) 0x0002: Zero 点漂移超出阈值(Drift Limit) 0x0005: 数据帧被破坏 0x00F0: 等待 Zero 点标定记录完成 0x00F1: 写入数据错误 0x00F2: 通信超时	R	HEX UINT16 Big Endian
0x2201	2	Span 标定记 录操作状态	0x0000: Span 点标定记录成功 0x0001: LARK-1 未稳定(Ref 信号为 0) 0x0002: Span 气浓度小于四分之一量程或 超出量程 0x0004: Span 点数据不合理(应检查通入的 Span 点气种和浓度) 0x0005: 数据帧被破坏 0x00F0: 等待 Span 点记录完成 0x00F1: 写入数据错误 0x00F2: 通信超时	R	HEX UINT16 Big Endian
0x2202	2	激活标定操作 状态	0x0000: 激活标定成功 0x0001: 激活标定失败 0x00F0: 等待标定激活完成 0x00F1: 写入数据错误 0x00F2: 通信超时	R	HEX UINT16 Big Endian
0x2203	2	加热/不加热操 作状态	0x0000: 操作成功 0x0001: 操作失败 0x00F0: 等待操作完成 0x00F1: 写入数据错误 0x00F2: 通信超时	R	HEX UINT16 Big Endian
0x2204	2	恢复出厂标定 操作状态	0x0000: 操作成功 0x0001: 操作失败 0x00F0: 等待操作完成 0x00F1: 写入数据错误 0x00F2: 通信超时	R	HEX UINT16 Big Endian

表 7. LARK-1 传感器可写寄存器(可写)

地址	长度 (字节)	名称	描述	读/写 R/W	类型
0x1000	2	零点标定记录 Zero Record	写入 0xFFFF 可记录当前采集数据为零点 注意：此时记录的零点并未生效，写用户标定激活寄存器(地址是 0x100B)可使其生效，并存储以供下次上电时使用	W	HEX UINT16 Big Endian
0x1001	4	Span1 标定点浓度 Span1 Value	Span1 标定时的标气浓度 '00 00 30 D4' for the '12500' 注意：此时记录的 SPAN 点并未生效，写用户标定激活寄存器(地址是 0x100B)可使其生效，并存储以供下次上电时使用	W	HEX UINT32 Big Endian
0x1003	4	Span2 标定点浓度 Span2 Value	Span2 标定时的标准气浓度 目前未使用	W	HEX UINT32 Big Endian
0x1005	4	Span3 标定点浓度 Span3 Value	Span3 标定时的标准气浓度 目前未使用	W	HEX UINT32 Big Endian
0x1007	4	Span4 标定点浓度 Span4 Value	Span4 标定时的标准气浓度 目前未使用	W	HEX UINT32 Big Endian
0x1009	4	Span5 标定点浓度 Span5 Value	Span5 标定时的标准气浓度 目前未使用	W	HEX UINT32 Big Endian
0x100B	2	用户标定激活	写入 0xFFFF 可激活标定	W	HEX UINT16 Big Endian
0x100C	2	加热 ON/OFF 控制	写入 0x00FF, 启动光路加热功能 写入 0x0000, 停止光路加热功能	W	HEX UINT16 Big Endian
0x100D	2	恢复出厂标定	写入 0xFFFF 可恢复出厂标定	W	HEX UINT16 Big Endian

- 使用 Modbus 操作 LARK-1 传感器

- LARK-1 传感器信息获取

使用功能码 0x04 根据表 4 中读 0x2100~0x21FF 地址段的寄存器数据。

比如要获取设备地址为 0x01 的 LARK-1 的 S.N.号,根据表 4 传感器序列号对应的寄存器地址为 0x2109, 长度为 12 个字节, 数据格式为 ASCII 码格式。

获取指令为: 01 04 21 09 00 06 AA 36

应答指令为: 01 04 0C 34 30 32 30 31 30 30 36 31 37 30 38 BB C3

由应答指令可知 S.N.号为: 402010061708

➤ LARK-1 数据采集

使用功能码 0x04 根据表 5 读 0x2000~0x20FF 地址段的寄存器数据。

比如要获取设备地址为 0x01 的 LARK-1 采集的待测气浓度值，根据表 5 待测气浓度值即 LARK-1 的 Reading 值，寄存器地址为 0x2000，长度为 4 个字节，数据格式为 HEX INT32 格式。

获取指令为：01 04 20 00 00 02 7A 0B

应答指令为：01 04 04 00 00 02 73 BB 01

由应答指令可知待测气体浓度为 0x00000273，即十进制的 627

➤ LARK-1 标定

1、零点标定(Zero Calibration)需要分**两步**进行，第一步通入零点气，记录零点，第二步标定点激活。

零点记录(Zero Record): 使用功能码 0x06 根据表 7 向 0x1000 寄存器写入 0xFFFF。

写入指令：01 06 10 00 FF FF 8C BA

应答指令：01 06 10 00 FF FF 8C BA

标定点激活(Activate): 使用功能码 0x06 根据表 7 向 0x100B 寄存器写入 0xFFFF。

2、SPAN 标定(Span Calibration)需要分**两步**进行，第一步通入 SPAN 点标准气，向 0x1001 写入标准气的浓度记录 SPAN1 标定点，第二步标定点激活。

SPAN1 标定点记录，比如通入 50000ppm 的标准气进行 SPAN1 点标定。

写入指令：01 10 10 01 00 02 04 00 00 C3 50 AF 6F

响应指令：01 10 10 01 00 02 14 C8

标定点激活(Activate): 使用功能码 0x06 根据表 7 向 0x100B 寄存器写入 0xFFFF。

3、注：每次执行写操作且从机返回应答后，主机应通过只读指令(功能码 0x04)根据表 6 读取对应操作状态以确保每次写操作都是有效的。

例如读取零点记录操作状态，根据表 6，零点记录操作状态对应的寄存器地址为 0x2200，长度为 2 个字节，数据格式为 HEX UINT16 格式。

获取指令为：01 04 22 00 00 01 3B B2

应答指令为：01 04 02 00 00 B9 30

由应答指令可知记录零点操作成功。

● 附录

附录 1 Sensor Type ID Definition

参见 AN-011 附录 1

附录 2. Sensor sub ID Definition

参见 AN-011 附录 2

附录 3. Reading unit ID

参见 AN-011 附录 3

附录 4. 主机常用操作指令 (以从机地址号为 0x01 为例, 实例所用数据都为 16 进制)

序号	功能	操作指令(HEX)
1	采集待测气浓度	01 04 20 00 00 02 7A 0B
2	查询 LARK-1 的序列号 S.N.	01 04 21 09 00 06 AA 36
3	查询 LARK-1 可测气种	01 04 21 04 00 05 7B F4
4	查询生产日期	01 04 21 0F 00 04 CB F6
5	查询有效日期	01 04 21 13 00 04 0A 30
6	查询 LARK-1 量程	01 04 21 1B 00 02 0B F0
7	查询 SPAN 点最小标定浓度	01 04 21 25 00 02 6A 3C
8	采集探测器和光源温度	01 04 20 02 00 02 DB CB
9	采集环境大气压	01 04 20 08 00 01 BB C8
10	Zero 点标定记录	01 06 10 00 FF FF 8C BA
11	Span1 点标定记录(以 50000ppm 作为 Span1 标定点气体浓度)	01 10 10 01 00 02 04 00 00 C3 50 AF 6F
12	用户标定激活	01 06 10 0B FF FF FD 78
13	恢复出厂标定(清除用户标定数据)	01 06 10 0D FF FF 1D 79
14	查询 Zero 点标定记录、Span 点标定记录、用户标定激活、加热操作、恢复出厂标定的操作状态	01 04 22 00 00 05 3A 71
15	启动加热	01 06 10 0C 00 FF 0D 49
16	停止加热	01 06 10 0C 00 00 4D 09

附录 5. CRC16 计算方法(C 语言版)

```
const uint8_t CRCTABL[] = //CRCL Value Table
```

```
{
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
```



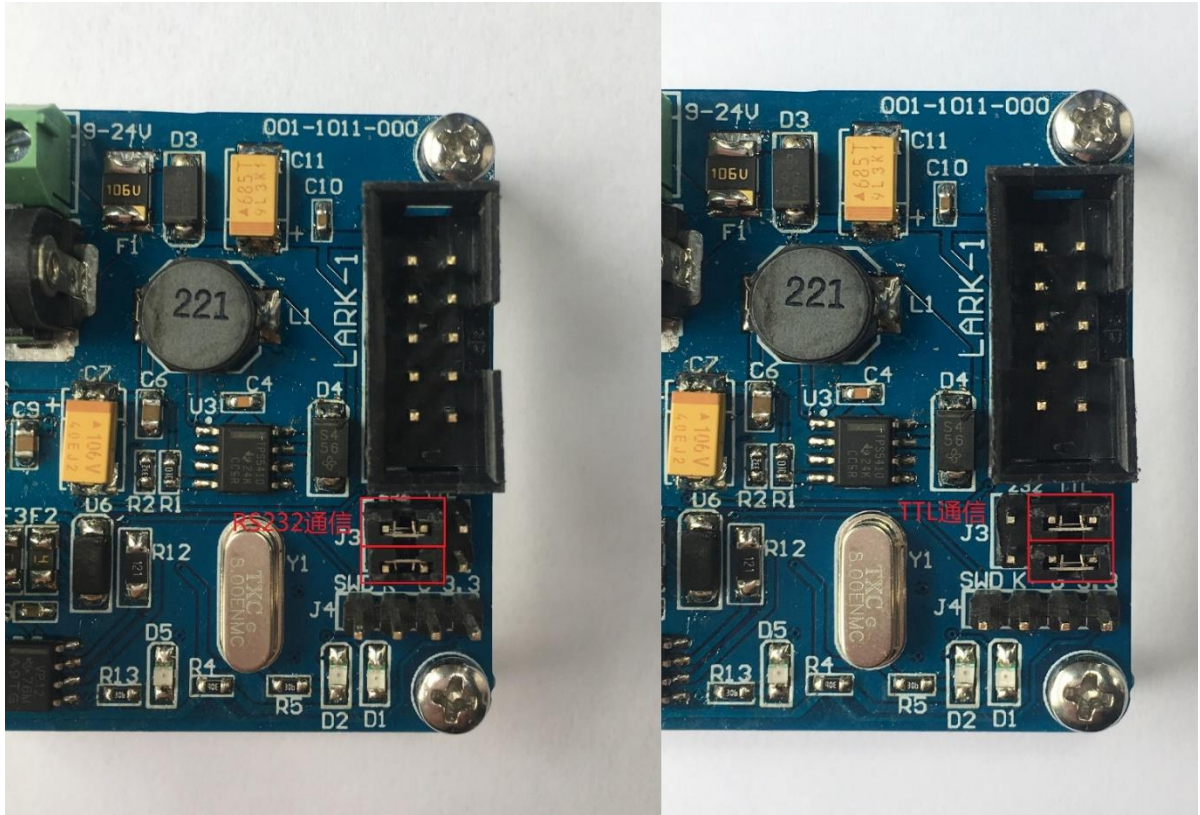
```

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40
};
const uint8_t CRCTABH[] = //CRCH Value Table
{
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7,
    0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E,
    0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9,
    0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32,
    0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D,
    0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF,
    0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1,
    0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB,
    0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA,
    0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97,
    0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E,
    0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89,
    0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83,
    0x41, 0x81, 0x80, 0x40
};
typedef struct _sCRC16Data
{
    uint8_t crch;
    uint8_t crcl;
}sCRC16Data;
sCRC16Data CRC16_8005Calc(uint8_t *p, int len)//x16+x15+x2+x1
{
    sCRC16Data crc = {0xFF, 0xFF};
    if (len > 0)
    {
        int i, index;
        for (i = 0; i<len; i++)
        {
            index = crc.crcl ^ (*(p + i));
            crc.crcl = crc.crch^CRCTABL[index];
            crc.crch = CRCTABH[index];
        }
    }
    return crc;
}

```

}

附录 6. 跳线帽安装方式



附录 7. 指示灯状态

D5 – 电源指示灯：

亮 – 电源正常

灭 – 电源异常

D2 – Modbus 通信指示灯：

亮 – 接收到主机的错误命令

灭 – 完成一次正常的通信或者主机未发送命令

闪 – 主机与此转换板正在通信

D1 – LARK-1 通信指示灯：

亮 – 转换板与 LARK-1 通信异常（常见原因 1.跳线帽安装错误，安装方式见附录 6，2.LARK-1 未连接）

灭 – 转换板故障

闪 – 转换板与 LARK-1 通信正常