

GUIDE

使用说明书

MODBUS-RTU 串行总线选件

GDHF-AMBX1 通讯卡

武汉港迪技术股份有限公司

Wuhan Guide Technology Co.,Ltd.

目录

安全须知	1
概述	1
安全须知总则	1
序言	2
面向的读者	2
准备工作	2
本使用说明书内容	2
概述	4
概述	4
MODBUS 标准	4
GDHF-AMBX1 通讯卡	5
兼容性	6
交货检查	6
保质期	6
机械安装	7
安装	7
电气安装	9
概述	9
电缆布线	9

总线设置	9
总线终端器	9
MODBUS 总线连接	10
编程	11
概述	11
系统配置	11
MODBUS 通讯配置	11
通讯	13
概述	13
串行链路层	13
MODBUS 协议传输模式	13
CRC 校验	13
MODBUS 协议	17
MODBUS 通讯卡协议功能码和帧结构	19
寄存器数据值和精度	25
寄存器地址	26
故障诊断	29
LED 显示	29
技术数据	30
串行链路	31

安全须知

概述

本章介绍了在安装和操作 GDHF-AMBX1 通讯卡时必须遵守的安全规则。在操作和使用变频器之前，务必阅读本章的内容。

除了仔细阅读下面的安全须知外，您还须阅读所使用型号的变频器完整的安全须知。

安全须知总则

警告!

所有关于变频器的电气安装和维护工作只能由具备资格的电气工程师来完成。

变频器和其相邻设备必须正确接地。

不要带电拆装变频器，在切断主电源之后，应该至少等待五分钟，待中间回路电容放电完毕后再拆装变频器、电机或电机电缆。

最好在进行工作之前检查变频器是否放电完毕（使用万用表）。

在接通主电源时，无论电机是否运行，电机电缆端子都处于危险高电压状态。

即使变频器的主电源被切断，其内部仍会存在由外部控制电路引入的危险电压，因此操作时应该倍加小心。忽视这些安全规则，将会引起人身伤害或死亡。

序言

面向的读者

本使用说明书面向的读者是那些负责调试和使用 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡的用户。读者需要具备基本的电气知识、电气接线经验以及变频器操作方面的知识。

准备工作

在开始安装 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡之前，变频器应该已经安装完毕并且可以准备投入使用。

除了备齐常规的安装工具之外，还应准备变频器使用说明书，这些使用说明书含有本使用说明书所没有的许多重要信息，因此需要在安装过程中进行查阅。

本使用说明书内容

本使用说明书介绍了关于 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡的配线、配置和使用方面的信息。

安全须知 位于本使用说明书的前几页。

概述 简要介绍了 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡，以及交货检查和产品保质期方面的信息。

机械安装 包含放置和安装通讯卡方面的信息。

电气安装 包含配线、总线终端器和接地方面的信息。

编程	介绍了在 GDHF-AMBX1 通讯卡进行通讯之前，如何对主机和变频器进行编程。
通讯	介绍了数据如何通过 GDHF-AMBX1 通讯卡进行传输。
故障跟踪	介绍了如何使用 GDHF-AMBX1 通讯卡上的状态显示 LED 进行故障跟踪。
定义和缩略语	解释了 MODBUS 协议中的定义和缩略语。

概述

概述

本章简要介绍了 MODBUS 标准，以及交货清单和保质期方面的信息。

MODBUS 标准

MODBUS 网络是一个工业通信系统，由带智能终端的可编程序控制器和计算机通过公用线路或局部专用线路连接而成。其系统结构既包括硬件、亦包括软件。它可应用于各种数据采集和过程监控。

MODBUS 网络只有一个主机，所有通信都由它发出。网络可支持 247 个之多的从机，但实际所支持的从机数要由所用通信设备决定。采用这个系统，各从机可以和主机交换信息而不影响各从机执行本身的控制任务。

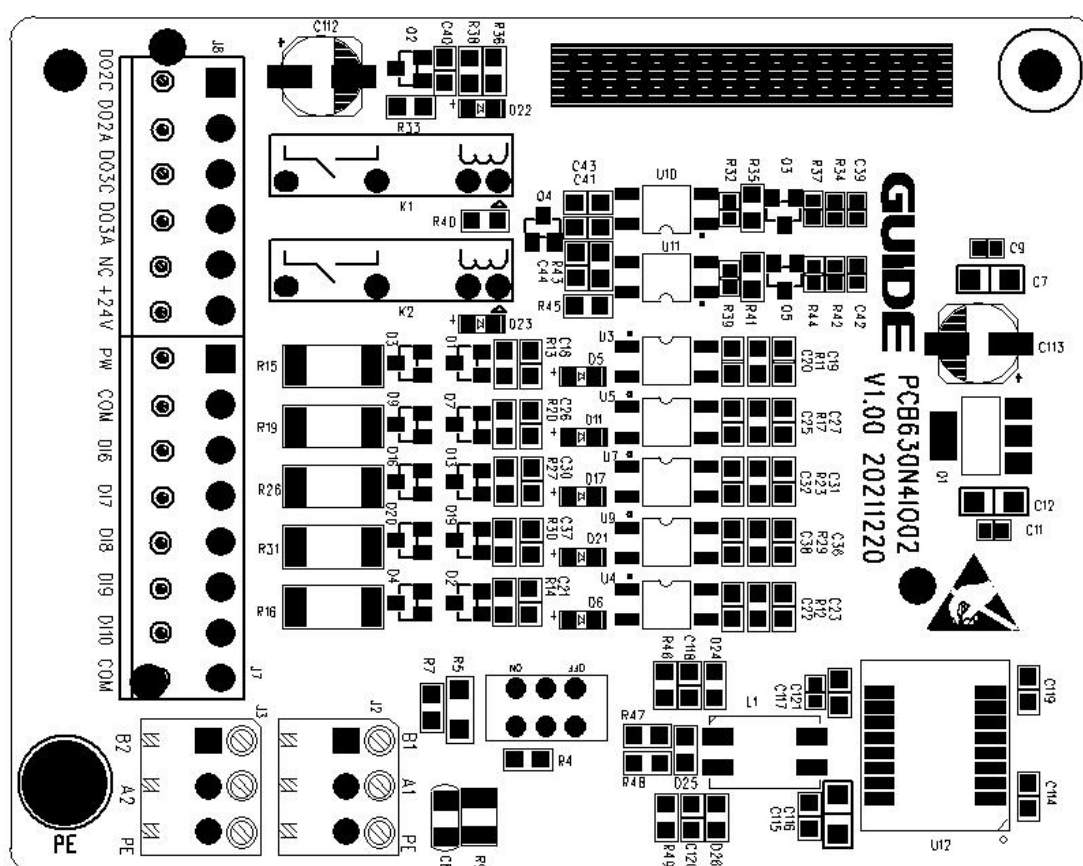
许多工业设备，包括 PLC，DCS，智能仪表等都在使用 MODBUS 协议作为他们之间的通讯标准。

MODBUS 具有以下几个特点：

1. 标准开放，用户可以免费、放心地使用 MODBUS 协议。目前，支持 MODBUS 的厂家超过 400 家，支持 MODBUS 的产品超过 600 种。
2. MODBUS 可以支持多种电气接口，如 RS-232、RS-485 等，还可以在各种介质上传送，如双绞线、光纤、无线等。

3. MODBUS 的帧格式简单、紧凑，通俗易懂，用户使用容易。控制器通信使用主—从技术，即仅一设备（主机）能初始化传输（查询）。其它设备（从机）根据主机查询提供的数据作出相应反应。典型的主机：可编程逻辑控制器（PLC）。典型的从机：设备或仪器仪表。

GDHF-AMBX1 通讯卡



GDHF-AMBX1 通讯卡布局图

MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡是港迪和 HF630N 系列变频器的一个可选设备，它可以将变频器连接到一个 MODBUS 网络。在 MODBUS 网络上，变频器当成从机。通过 GDHF-AMBX1 通讯卡，可以：

- 向变频器发出控制命令（启动、停止、允许运行等等）。
- 给变频器发送速度或转矩等给定信号。
- 向变频器的 PID 调节器发送一个过程实际值或过程给定信号。
- 从变频器中读取状态信号和实际值。
- 对变频器进行故障复位。

MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡与变频器内控制板的 J25 插孔相连。

兼容性

港迪 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡与下列产品兼容：

- 港迪 HF630N 系列变频器。
- 所有支持 RS485 总线的 MODBUS RTU 协议主机站。

交货检查

MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡的包装箱内包括：

- MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡
- 固定螺钉
- 本使用说明书

保质期

制造商承诺凡是设备在设计、材料和工艺上存在缺陷时，都会给予质量保证。保质期为从制造日期起 36 个月。

机械安装

警告!

安装前，务必切断变频器的电源。并至少等待 10 分钟，以确保变频器的电容器组放电完毕。切断从外部控制电路到变频器输入端的危险电压。

安装

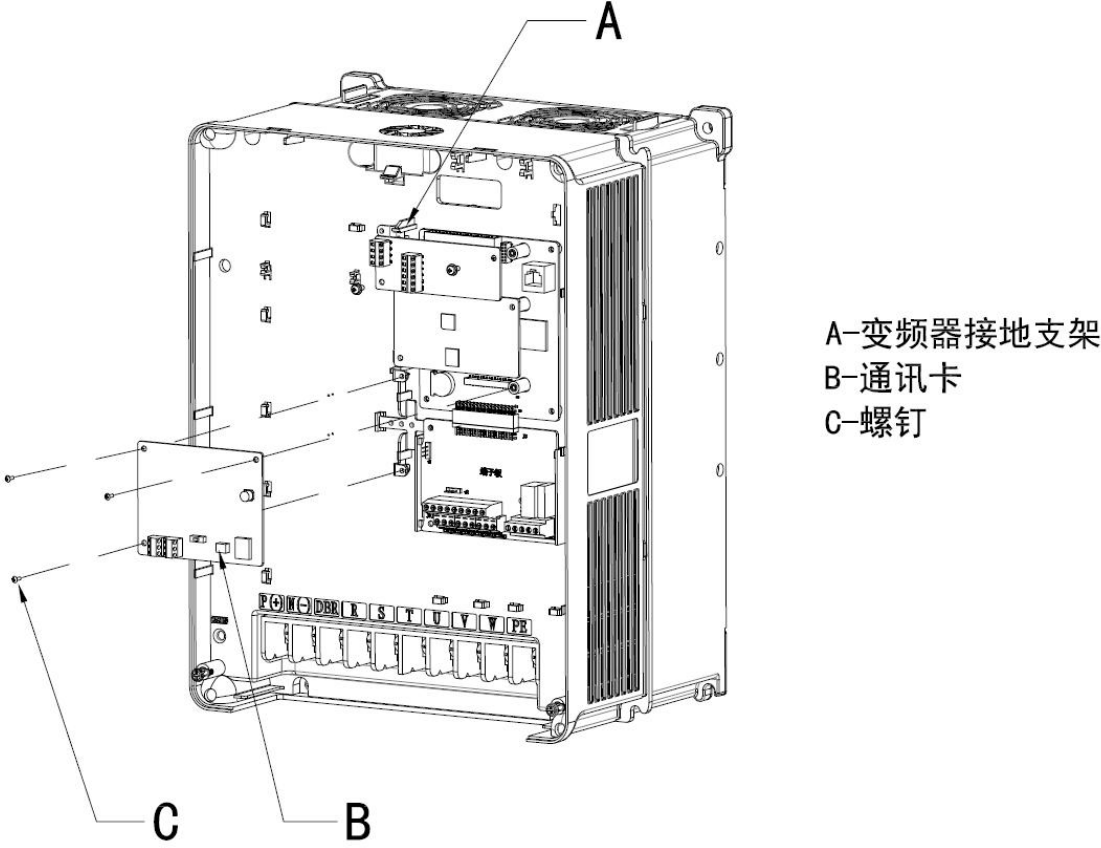
将 GDHF-AMBX1 通讯卡对齐两个固定螺丝孔和控制板 J25 的信号插孔，将 GDHF-AMBX1 通讯卡插入控制板信号插孔中，用螺钉将 GDHF-AMBX1 通讯卡固定。

安装步骤:

- 将 GDHF-AMBX1 通讯卡小心地插入相应插孔。
- 用固定螺钉，牢固的固定住 GDHF-AMBX1 通讯卡。
- 设置通讯卡的总线终端开关至所需位置。

注意: 为了履行 EMC 的要求以及保证 GDHF-AMBX1 通讯卡能正常工作，应将通讯线的屏蔽层与 GDHF-AMBX1 通讯卡连接端子的 PE 端口相连。确保通讯线的屏蔽层良好的连接到变频器保护接地上。

通讯卡的安装如下图所示。



通讯卡的安装

电气安装

概述

本章包含：

- 电缆布线方面的指导信息。
- 设置总线终端器方面的指导信息。
- 模块和 MODBUS 网络连接方面的指导信息。

警告！ 安装前，务必切断变频器的电源。并至少等待 10 分钟，以确保变频器的电容器组放电完毕。切断从外部控制电路到变频器输入端的危险电压。

电缆布线

尽可能使总线远离电机电缆。避免平行布线。在电缆入口处使用套管。

总线设置

在变频器上位机软件或操作面板上，根据实际应用来设置 RS485 端口。

总线终端器

MODBUS 通讯卡底板印刷电路板上的 J1 开关用于接通 RS485 总线终端器和 A2、B2 信号。总线终端器给 RS485 总线提供一个终结电阻，可以防止 RS485 总线电缆端的信号反射。A2、B2 信号用于连接其他的 MODBUS 设备。如果模块是网络中的最后一个模块或是第一个模块，总线终端器必须设置为 ON。

MODBUS 总线连接

总线电缆接至 GDHF-AMBX1 通讯卡的端子 J2、J3 上。

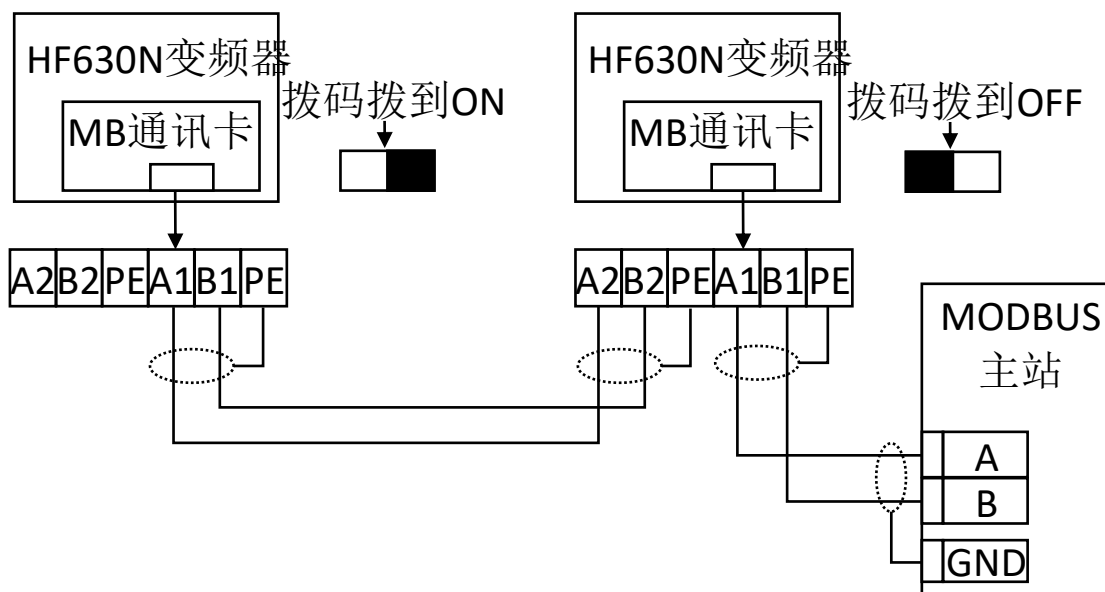
连接端子分配如下：

端子	说明
A2	RS485_A2
B2	RS485_B2
PE	保护地
A1	RS485_A1
B1	RS485_B1
PE	保护地

注意：如已采用变频器上的 M 型电缆支架进行屏蔽层接地连接，则 PE 端子可以不接 MODBUS 电缆屏蔽层。

MODBUS 接线实例：

MODBUS 电缆屏蔽层在所有节点上都直接连接到通讯卡相应信号端口的 GND 端子。



标准 MODBUS RS485 电缆接线图

编程

概述

本章介绍了 MODBUS 主机站配置方面的信息，以及变频器如何通过 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡实现通讯方面的信息。

系统配置

在按照前两章内容安装好 MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡之后，必须要对主机站和变频器进行配置，以使它们之间建立通讯。

MODBUS 通讯配置

建立 GDHF-AMBX1 通讯卡与变频器之间的通讯，首先需要在变频器中对 MODBUS 通讯参数进行配置。下面对这些参数的选项进行了详细的介绍。

注意：新设置的参数立即生效。

表 1 MODBUS 参数配置

参数号	名称	说明	设定范围	缺省值
P2.5	MODBUS ID号	根据实际应用配置设置	1~247	1
P2.6	MODBUS波特率	[0] 9600BPS [1] 14400BPS [2] 19200BPS [3] 38400BPS [4] 56000BPS [5] 57600BPS [6] 115200BPS [7] 128000BPS [8] 256000BPS	0~8	3
P2.7	MODBUS通讯类型	[0] RS232C [1] RS485	0~1	1

1. MODBUS 从站 ID

在 MODBUS 网络中，每台设备都对应一个唯一的节点地址，根据实际应用设置该从站地址。

2. 波特率选择

选择总线端口的波特率，根据实际应用中，与 MODBUS 主站设置的波特率必须保持一致。

3. MODBUS 通讯类型

此处选择。[1]RS485。

通讯

概述

本章介绍了变频器通讯时所使用的 MODBUS RTU 模式通信数据内容。

串行链路层

MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡支持 RS485 串行接口。

符合 EIA/TIA 485-A 规范。

MODBUS 协议传输模式

MODBUS 协议有两种传输模式：ASC II 模式和 RTU 模式。

在同一个 MODBUS 网络上的所有设备都必须选择相同的传输模式。

MODBUS-RTU 串行总线选件 GDHF-AMBX1 通讯卡仅支持 RTU 模式。

CRC 校验

CRC 即循环冗余校验码 (Cyclic Redundancy Check)：是数据通信领域中最常用的一种差错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

循环冗余校验码 (CRC) 的基本原理是：在 K 位信息码后再拼接 R

位的校验码，整个编码长度为 N 位，因此，这种编码也叫 (N, K) 码。对于一个给定的 (N, K) 码，可以证明存在一个最高次幂为 $N-K=R$ 的多项式 $G(x)$ 。根据 $G(x)$ 可以生成 K 位信息的校验码，而 $G(x)$ 叫做这个 CRC 码的生成多项式。校验码的具体生成过程为：假设要发送的信息用多项式 $C(x)$ 表示，将 $C(x)$ 左移 R 位（可表示成 $C(x) \times 2^R$ ），这样 $C(x)$ 的右边就会空出 R 位，这就是校验码的位置。用 $C(x) \times 2^R$ 除以生成多项式 $G(x)$ 得到的余数就是校验码。

在实际应用中，对数据逐位计算，或查表来获得 CRC 校验码。

对数据进行逐位计算，求得 CRC 校验码，该方法，需要较多的 CPU 计算量，在其他任务的 CPU 处理负荷较高或任务执行时间要求较短的情况下，通过逐位计算 CRC 校验码的方法并不适用。但计算过程的指令代码较少，对 CPU 代码存储空间紧张的应用中具有很强的实用性。

查表法，计算过程简单，处理时间快，在任务执行时间要求短，代码存储空间充裕的应用中，使用查表法获得 CRC 校验码是最好的选择。

以下代码为 C 语言实现的 CRC16 位校验码查表法，仅供用户参考。

```
INT16U CRC16 (const INT8U *nData, INT16U wLength)
{
    static const INT16U wCRCTable[] = {
        0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,
        0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,
        0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,
        0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,
```

0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81, 0X1A40,
 0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,
 0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,
 0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,
 0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,
 0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,
 0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,
 0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,
 0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
 0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,
 0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,
 0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,
 0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,
 0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,
 0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,
 0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,
 0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
 0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40,
 0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,
 0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
 0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
 0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
 0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
 0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
 0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
 0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
 0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
 0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040 };

INT8U nTemp;

```
INT16U wCRCWord = 0xFFFF;

while (wLength--)
{
    nTemp = *pData++ ^ wCRCWord;
    wCRCWord >>= 8;
    wCRCWord ^= wCRCTable[nTemp];
}

return wCRCWord;
} // End: CRC16
```

MODBUS 协议

此协议定义了一个控制器能认识使用的消息结构，而不管它们是经过何种网络进行通信的。它描述了一个控制器请求访问其它设备的过程，如何回应来自其它设备的请求，以及怎样侦测错误并记录。它制定了消息域格局和内容的公共格式。

当在一个 MODBUS 网络上通信时，此协议决定了每个控制器需要知道它们的设备地址，识别按地址发来的消息，决定要产生何种行动。如果需要回应，控制器将生成反馈信息并用 MODBUS 协议发出。在同一个 MODBUS 网络上的所有设备都必须选择相同的传输模式。

MODBUS 主从设备，都以 MODBUS 消息帧来查询和回应。MODBUS 典型的消息帧包含地址域、功能域、数据域、帧校验域。RTU 模式中，消息帧发送至少要以 3.5 个字符时间的停顿间隔开始。在网络波特率下多样的字符时间，这是最容易实现的（如下表的 T3.5 所示）。传输的第一个域是设备地址。可以使用的传输字符是十六进制的 0...9, A...F。网络设备不断侦测网络总线，包括停顿间隔时间内。当第一个域（地址域）接收到，每个设备都进行解码以判断是否发往自己的。在最后一个传输字符之后，一个至少 3.5 个字符时间的停顿标定了消息的结束。一个新的消息可在此停顿后开始。

整个消息帧必须作为一连续的流传输。如果在帧完成之前有超过 1.5 个字符时间的停顿时间，接收设备将刷新不完整的消息并假定

下一字节是一个新消息的地址域。同样地，如果一个新消息在小于 3.5 个字符时间内接着前个消息开始，接收的设备将认为它是前一消息的延续。这将导致一个错误，因为在最后的 CRC 域的值不可能是正确的。

下表展示了一个 MODBUS 典型消息帧的详细组成结构：

表 2 MODBUS 典型消息帧

消息帧起始	地址域	功能域	数据域	帧 CRC 校验域	消息帧结束
T3.5	1Byte	1Byte	NBytes	2Bytes	T3.5

地址域：指示一个 MODBUS 设备的地址，有效范围 1 ~ 247。0 地址为广播地址，247 以后的地址为协议保留地址，GDHF-AMBX1 通讯卡该地址有效范围为 1 ~ 247；

功能域：指示该消息帧的功能码，即该消息的具体功能，GDHF-AMBX1 通讯卡支持功能码 F03、F06、F16。其他具体功能码的含义，请查看 MODBUS 官方协议文档。

数据域：不同的功能码，数据域相应字节位的定义不经相同，详细定义参见 MODBUS 官方协议文档。

帧 CRC 校验域：包含一个 16 位的 CRC 校验码，该校验码是从消息帧地址域开始，一直到数据域结束多有的数据做 CRC16 计算而得到一个帧校验码，用于确保整个消息传输后的消息错误检测。

MODBUS 通讯卡协议功能码和帧结构

功能码：0x01(HEX)，读线圈(Read Coils)：

主机询问的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	0x01, 读线圈
2	线圈起始地址 [高位]	读取的线圈起始地址
3	线圈起始地址 [低位]	
4	读取的线圈数量 [高位]	线圈数量<N>
5	读取的线圈数量 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	0x01
2	数据字节数	N / 8
3	线圈[8-0]	线圈值
4	线圈[16-9]	
.....	CRC 校验码 [低位]	CRC16 校验码
.....	CRC 校验码 [高位]	
消息帧总长度：(N / 8 + 5) 个字节		

功能码：0x03(HEX)，读保持寄存器(Read Holding Registers)：

主机询问的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	3, 读保持寄存器
2	寄存器起始地址 [高位]	读取的寄存器起始地址
3	寄存器起始地址 [低位]	
4	读取的寄存器数量 [高位]	寄存器数量<N>
5	读取的寄存器数量 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	3
2	数据字节数	$2 \times N$
3	寄存器 1 [高位]	第一个寄存器值
4	寄存器 1 [低位]	
5	寄存器 2 [高位]	第二个寄存器值
6	寄存器 2 [低位]	
.....
$2 \times N + 1$	寄存器 N [高位]	第 N 个寄存器值
$2 \times N + 2$	寄存器 N [低位]	
$2 \times N + 3$	CRC 校验码 [低位]	CRC16 校验码
$2 \times N + 4$	CRC 校验码 [高位]	
消息帧总长度：(2×N + 5) 个字节		

功能码：0x02(HEX)，读离散输入 (Read Discrete Inputs):

主机询问的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	0x02, 读离散输入
2	离散输入起始地址 [高位]	读取的线圈起始地址
3	离散输入起始地址 [低位]	
4	读取离散输入的数量 [高位]	Input 数量<N>
5	读取离散输入的数量 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	0x02
2	数据字节数	N / 8
3	Input [8-0]	Input 值
4	Input [16-9]	
.....	CRC 校验码 [低位]	CRC16 校验码
.....	CRC 校验码 [高位]	
消息帧总长度：(N / 8 + 5) 个字节		

功能码：0x06(HEX)，写单个寄存器(Write Single Register)：

主机询问的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	6, 写单个寄存器
2	寄存器地址 [高位]	寄存器地址
3	寄存器地址 [低位]	
4	寄存器 1 [高位]	寄存器值
5	寄存器 1 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	6
2	寄存器地址 [高位]	寄存器地址
3	寄存器地址 [低位]	
4	寄存器 1 [高位]	寄存器值
5	寄存器 1 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

功能码：0x10(HEX)，写多个寄存器(Write Multiple Registers)：

主机询问的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	16, 写多个寄存器
2	寄存器起始地址 [高位]	寄存器起始地址
3	寄存器起始地址 [低位]	
4	寄存器数量 [高位]	寄存器数量<N>
5	寄存器数量 [低位]	
6	数据字节数	2×N
7	寄存器 1 [高位]	第一个寄存器值
8	寄存器 1 [低位]	
9	寄存器 2 [高位]	第二个寄存器值
10	寄存器 2 [低位]	
.....
2×N + 5	寄存器 N [高位]	第 N 个寄存器值
2×N + 6	寄存器 N [低位]	
2×N + 7	CRC 校验码 [低位]	CRC16 校验码
2×N + 8	CRC 校验码 [高位]	
消息帧总长度：(2×N + 9) 个字节		

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	16
2	寄存器起始地址 [高位]	寄存器起始地址
3	寄存器起始地址 [低位]	
4	寄存器数量 [高位]	寄存器数量<N>
5	寄存器数量 [低位]	
6	CRC 校验码 [低位]	CRC16 校验码
7	CRC 校验码 [高位]	
消息帧总长度：8 个字节		

错误消息帧 (Protocol Exceptions):

主机在发起一个读或写操作时，难免会在地址范围、读写权限、寄存器数据范围等情况下出现错误。例如：读取的地址超出了通讯卡所支持的地址范围；或向一个只读的地址发起了一个写操作；或写入的寄存器值，超出了该寄存器所支持的数据范围等。此时，GDHF-AMBX1 通讯卡会拒绝主机发送来的读写操作，并回复一个错误的消息帧。

从机响应的消息帧格式：

帧字节顺序	数据定义	描述
0	从机地址	1 ~ 247
1	功能码	0x80 + 操作功能码
2	错误码	错误码解释，见下表
3	CRC 校验码 [低位]	CRC16 校验码
4	CRC 校验码 [高位]	
消息帧总长度：5 个字节		

其中错误码有：

错误码	错误名称	描述
0x01	违法的功能码	本通讯卡不支持的功能码
0x02	违法的数据地址	本通讯卡不支持的寄存器地址
0x03	违法的数据	写寄存器数据值超出寄存器定义的数据范围；或向一个只读寄存器进行写操作
0x06	从机忙	控制板正在进行相应寄存器的操作；或通讯卡正在进行通讯操作，不能回复主机的请求

寄存器数据值和精度

GDHF-AMBX1 通讯卡访问的控制板寄存器数据中，部分参数数据包含一定小数位，而 MODBUS 寄存器数值是一个长度为 2 个字节的整数，并不能代表包含小数点的浮点数，所以，我们在定义寄存器数据有效值时，引入了数据精度这个计算量。即：**寄存器值 = 实际参数值 × 精度**。

数据精度，在下文中，用以下格式表示：

精度	含义
×1	实际参数没有小数位
×10	实际参数有 1 位小数位
×100	实际参数有 2 位小数位
×1000	实际参数有 3 位小数位

寄存器值，实际参数值和精度的换算过程，如下表：

MODBUS 寄存器值	实际参数值	精度
1	1	×1
1	0.1	×10
1	0.01	×100
1	0.001	×1000

例如：给定频率这个寄存器，单位为 Hz，精度×10，实际给定频率的值应为 49.9Hz，那么通过 GDHF-AMBX1 通讯卡向该寄存器写入的寄存器值应为， $49.9 \times 10 = 499$ 。向该寄存器地址写入 499 这个数值，实际给定频率这个参数的值即修改为 49.9Hz。

寄存器地址

GDHF-AMBX1 通讯卡访问的寄存器都有相应的功能定义、精度、单位。

每个寄存器都包含了一个 HF630N 系列变频器的特定操作功能或实时参数，具体寄存器的功能定义、数据精度、数值单位，请查看如下寄存器地址分配表：

功能码 0x01, Coils		
地址	名称	描述
0	运行停止	0: 停止 1: 运行
1	方向	0: 正转 1: 反转
2	本地远程切换	写 1: 切换
3	故障复位	写 1: 故障复位
4	外部故障	0: 无 1: 外部故障
5	外部告警	0: 无 1: 外部告警
6	电机选择	00: 电机 1
7		01: 电机 2 10: 电机 3 11: 电机 4
8	零转矩使能	0: 禁止 1: 使能

功能码 0x02, Discrete Input		
地址	名称	描述
0	系统准备好	0: 忙 1: 系统准备好
1	运行停止	0: 停止 1: 运行
2	故障	0: 正常 1: 故障
3	电机制动器	0: 抱闸 1: 未抱闸

4	告警	0: 无 1: 故障
5	方向	0: 正转 1: 反转
10	本地/远程	0: 远程 1: 本地
11	电机选择	00: 电机 1 01: 电机 2
12		10: 电机 3 11: 电机 4
13	驱动器使能	0: 不使能 1: 使能

功能码 0x03, Holding Regs		
地址	名称	描述
0	D0 输出端子	[读/写] : D0
50	A01 模拟量输出	[读/写] : 10 => 1.0[%]
51	A02 模拟量输出	[读/写] : 10 => 1.0[%]
60	故障代码	[07..00] : 故障代码 [15..08] : 警告代码
62	参数错误码	[07..00] : 故障代码 [15..08] : 警告代码
100	频率给定 [Hz]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [Hz]
101	转速给定 [rpm]@Modbus 接口	[读/写] : $\pm 1 \Rightarrow \pm 1$ [rpm]
102	频率给定 [%]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [%]
103	转矩给定 [%]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [%]
104	转矩限制值给定 [%]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [%]
105	有功电流设定 [%]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [%]
106	无功电流设定 [%]@Modbus 接口	[读/写] : $\pm 10 \Rightarrow \pm 1.0$ [%]
107	加速时间倍数	[读/写] : 1000 => 1.0, [0.05 ~ 10.0]
108	减速时间倍数	[读/写] : 1000 => 1.0, [0.05 ~ 10.0]
4500	当前频率 [Hz]	[读] : $\pm 10 \Rightarrow \pm 1.0$ [Hz]
4501	当前转速 [rpm]	[读] : $\pm 1 \Rightarrow \pm 1.0$ [rpm], @100ms
4502	当前频率 [%]	[读] : $\pm 10 \Rightarrow \pm 1.0$ [%]
4503	输出电流	[读] : $\pm 10 \Rightarrow \pm 1.0$ [A], @100ms
4504	输出电压	[读] : $\pm 10 \Rightarrow \pm 1.0$ [V], @100ms

4505	直流母线电压	[读] : $\pm 10 \Rightarrow \pm 1.0[V]$, @100ms
4506	输出转矩	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$, @100ms
4507	负载转矩	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$, @100ms
4508	输出功率	[读] : $\pm 10 \Rightarrow \pm 1.0[kW]$, @250ms
4509	温度	[读] : $10 \Rightarrow 1.0[^\circ C]$
4510	估算频率[Hz]	[读] : $\pm 10 \Rightarrow \pm 1.0[Hz]$
4511	估算转速[rpm]	[读] : $\pm 1 \Rightarrow \pm 1.0[rpm]$
4512	估算频率[%]	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$
4513	频率给定[Hz]	[读] : $\pm 10 \Rightarrow \pm 1.0[Hz]$
4514	转速给定[rpm]	[读] : $\pm 1 \Rightarrow \pm 1.0[rpm]$
4515	频率给定[%]	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$
4516	频率给定[Hz]@(斜坡值)	[读] : $\pm 10 \Rightarrow \pm 1.0[Hz]$
4517	转速给定[rpm]@(斜坡值)	[读] : $\pm 1 \Rightarrow \pm 1.0[rpm]$
4518	频率给定[%]@(斜坡值)	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$
4519	转矩给定[%]	[读] : $\pm 10 \Rightarrow \pm 1.0[\%]$

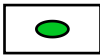


功能码 0x04, Input Regs		
地址	名称	描述
0	DI 输入端子	[读]
50	AI1 输入端子[V]	[读] : $100 \Rightarrow 1.0[V]$, $-10 \sim +10[V]$
51	AI1 输入端子[I]	[读] : $100 \Rightarrow 1[mA]$, $0 \sim 20[mA]$
52	AI2 输入端子[V]	[读] : $100 \Rightarrow 1.0[V]$, $-10 \sim +10[V]$
53	AI2 输入端子[I]	[读] : $100 \Rightarrow 1[mA]$, $0 \sim 20[mA]$

故障诊断

LED 显示

GDHF-AMBX1 通讯卡有 3 个总线状态显示 LED。这些 LED 的分布位置，请参看通讯卡布局图。

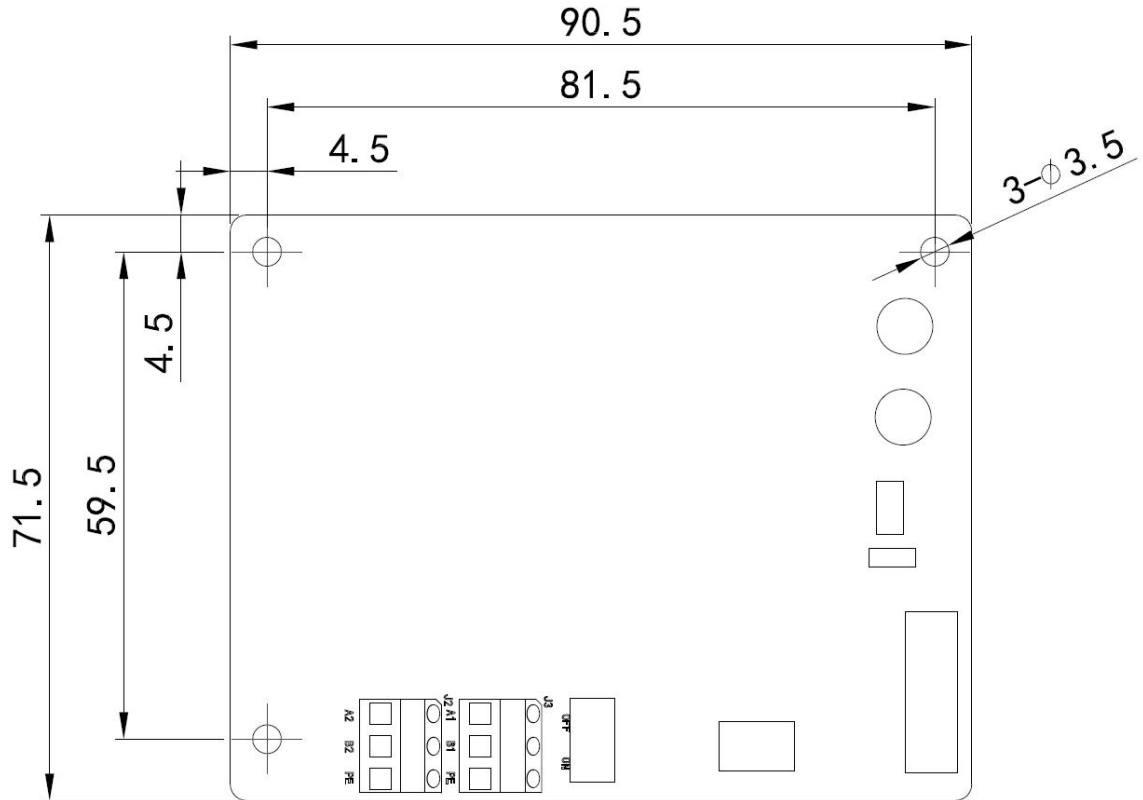
这些 LED 的作用如下：

LED 状态指示灯		
		
RUN	RX	TX

名称	颜色	功能描述
运行状态 (RUN)	绿	快闪： MODBUS 总线通讯异常（闪烁频率 10Hz） 慢闪： 工作正常（闪烁频率 1Hz） 常灭： 通讯卡与控制板连接异常
接收指示 (RX)	绿	亮： 通讯卡正在接收一个消息帧 灭： 总线无接收操作
发送指示 (TX)	绿	亮： 通讯卡正在发送一个消息帧 灭： 总线无发送操作

技术数据

机械尺寸：



GDHF-AMBX1 通讯卡尺寸（单位 mm）

安装： 插入变频器控制板的相应插孔中。

环境条件： 见《港迪 HF630N 系列变频器使用说明书》中关于环境条件方面的相关内容。

硬件设置： 用于总线终端选择的 DIP 开关。

软件设置：

- 模块使能
- 总线接口参数设置
- 站点号设置
- 故障检测时间设置

串行链路

兼容设备： 所有遵循 MODBUS 协议的设备，接口遵循 EIA/TIA 485-A 规范。

连接数量： **RS485：** 127 个站点，包括中继器（每段有 31 站和 1 个中继器）

介质： 带屏蔽层的双绞线 RS-485 电缆

终端器： 模块内部。

技术规格：

RS485 接口			
参数	LINE A	LINE B	单位
感抗	135~165(3~20MHz)	100~130(f>100KHz)	W
电容	<30	<60	PF/m
阻抗	<110	--	Ω/Km
电线号码	>0.64	>0.53	mm
导体芯横截面	>0.34	>0.22	mm ²
最大总线长度			
传输率 Kbit/s	<93.75	115.2	
LINE A (米)	1200	1000	
LINE B (米)	1200	600	

拓扑结构： **RS485：** 多点

串型通讯类型： **RS485：** 异步、半双工

传输率： 9.6kbit/s、14.4kbit/s、19.2kbit/s、38.4kbit/s、56kbit/s、57.6kbit/s、115.2kbit/s

协议： MODBUS RTU

武汉港迪技术股份有限公司

Wuhan Guide Technology Co.,Ltd.

地址：武汉东湖新技术开发区理工大科技园理工园路 6 号

邮编：430223

电话：86-027-87927230

邮箱：shfw@gdetec.com

网址：www.gdetec.com

售后服务专线：400-0077-570