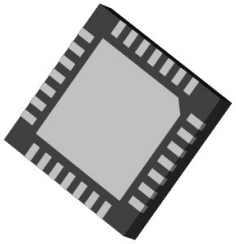
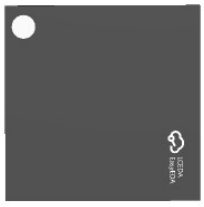
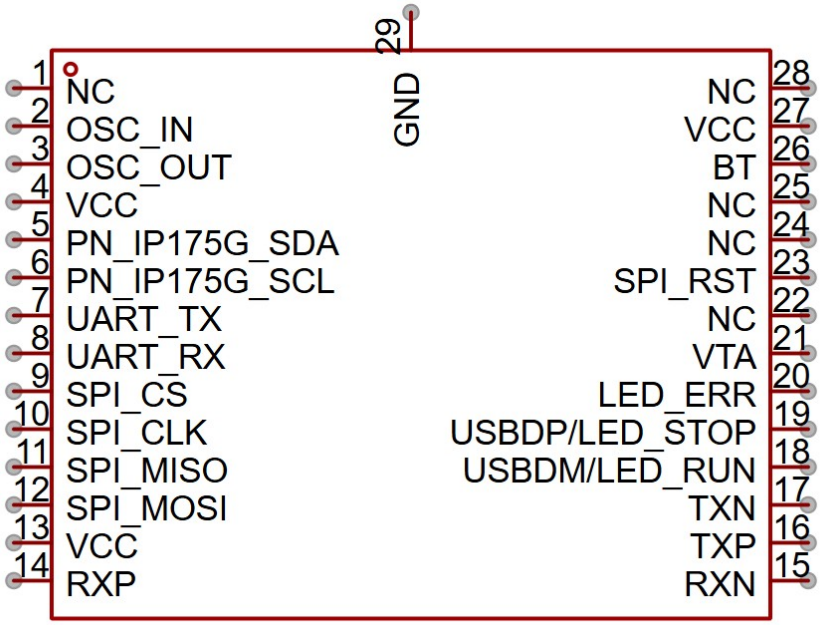
Profinet 透传 SPI 从站协议芯片

2025.3.27

原理图







(4x4mm QFN28 封装）

- TXP,TXN,RXP,RXN 管脚接 IP175G。
- 2 根 PN_IP175G_XX 管脚接 IP175G 交换机芯片。
- 5 根 SPI_XX 管脚接 MCU 的 SPI 接口，SPI 从站模式为模式 0，最高 30M，高位（MSB）在前，CPOL=0，CPHA=0；

(1)SPI_RST 管脚为协议芯片 SPI 从站的复位接口，此管脚为上拉输入，用户 SPI 主站在第一次读取前或者主站重新初始化后进行复位，复位前根据串口请求返回的状态信息来确定是否具备复位功能（或等待 PN 正常通信后），当具备复位功能时需将此管脚拉低 200ms 以上再持续拉高进行复位 SPI 从站，复位完成后方可进行对协议栈的 I 区与 Q 区进行读写操作，每次 MCU 读写长度为 PLC 硬件组态时配置的 Q 区或 I 区的最大字节长度（Q 区或 I 区哪个大以哪个为准）；

(2)SPI_CS 管脚需用户的 SPI 主站在收发每包数据前拉低，收发结束后拉高；

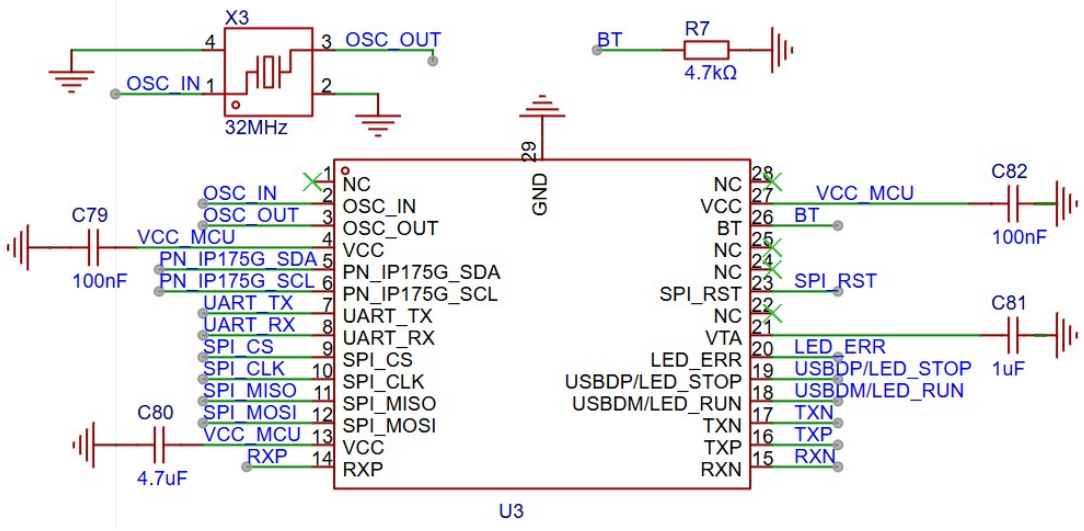
(3)SPI_CLK 管脚为 PN 芯片的 SPI 时钟输入管脚，接用户 SPI 主站的 CLK；

(4)SPI_MISO 管脚为 PN 芯片的 SPI 数据发送管脚，接用户 SPI 主站的 MISO；

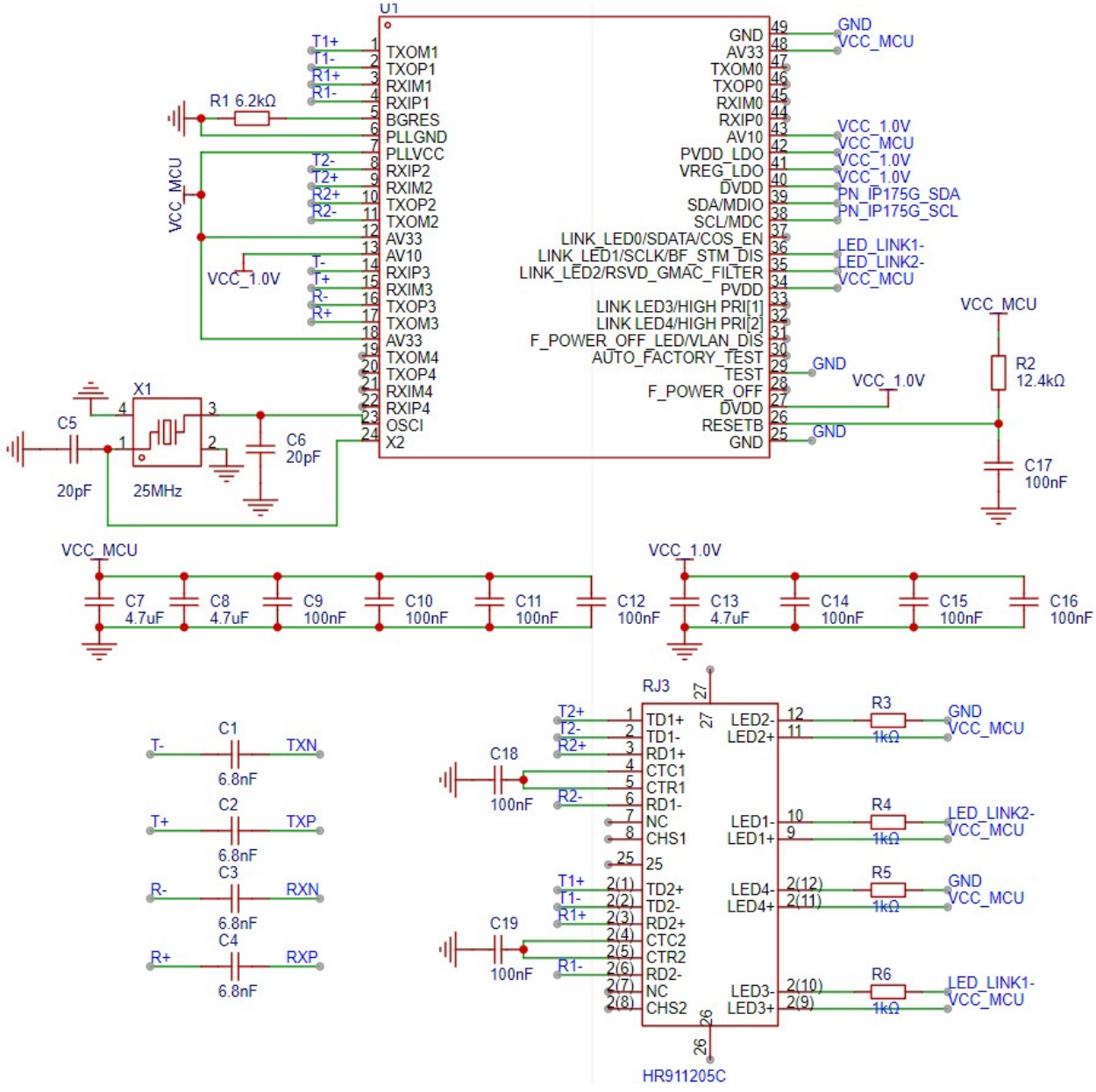
(5)SPI_MOSI 管脚为 PN 芯片的 SPI 数据接收管脚，接用户 SPI 主站的 MOSI；

- UART 的 2 根管脚接 MCU，用于配置参数等作用，默认波特率为 2M。
- VTA 脚接 1uf 电容到 GND。
- BT 脚接 4.7K 电阻到 GND。
- OSC_IN 与 OSC_OUT 需外接 32M 高速晶振（不需要加匹配电容）。
- USBDM/LED_RUN 管脚为升级与 LED 指示复用引脚，默认为通信连接指示灯，当模块正常启动并连接 Profinet 网络正常后常亮，当软件点击闪烁 LED 后 LED_RUN 与 LED_STOP 交替以 200ms 频率闪烁，当通信异常但物理连接正常时以 150ms 闪烁，当物理连接异常时熄灭（正常输出低电平，需接 1K 限流电阻后接 LED 灯到 3.3VCC），当升级时为 USBDM 脚。
- USBDP/LED_STOP 管脚为升级与 LED 指示复用引脚，默认为通信异常指示灯，当模块与 PLC\交换机物理连接异常时或与 PLC 的 Profinet 通信未建立连接时常亮，当软件点击闪烁 LED 后 LED_RUN 与 LED_STOP 交替以 200ms 频率闪烁（正常输出低电平，需接 1K 限流电阻后接 LED 灯到 3.3VCC），当升级时为 USBDP 脚。
- LED_ERR 管脚为系统异常指示灯，当模块 Profinet 通信未连接时以 250ms 周期闪烁，当 Profinet 正常通信时 Modbus 侧通信有异常时以错误的从站位置进行亮灭（正常输出低电平，需接 1K 限流电阻后接 LED 灯到 3.3VCC）。
- VCC 管脚接 3.3V，GND 接地。

芯片电路参考图：

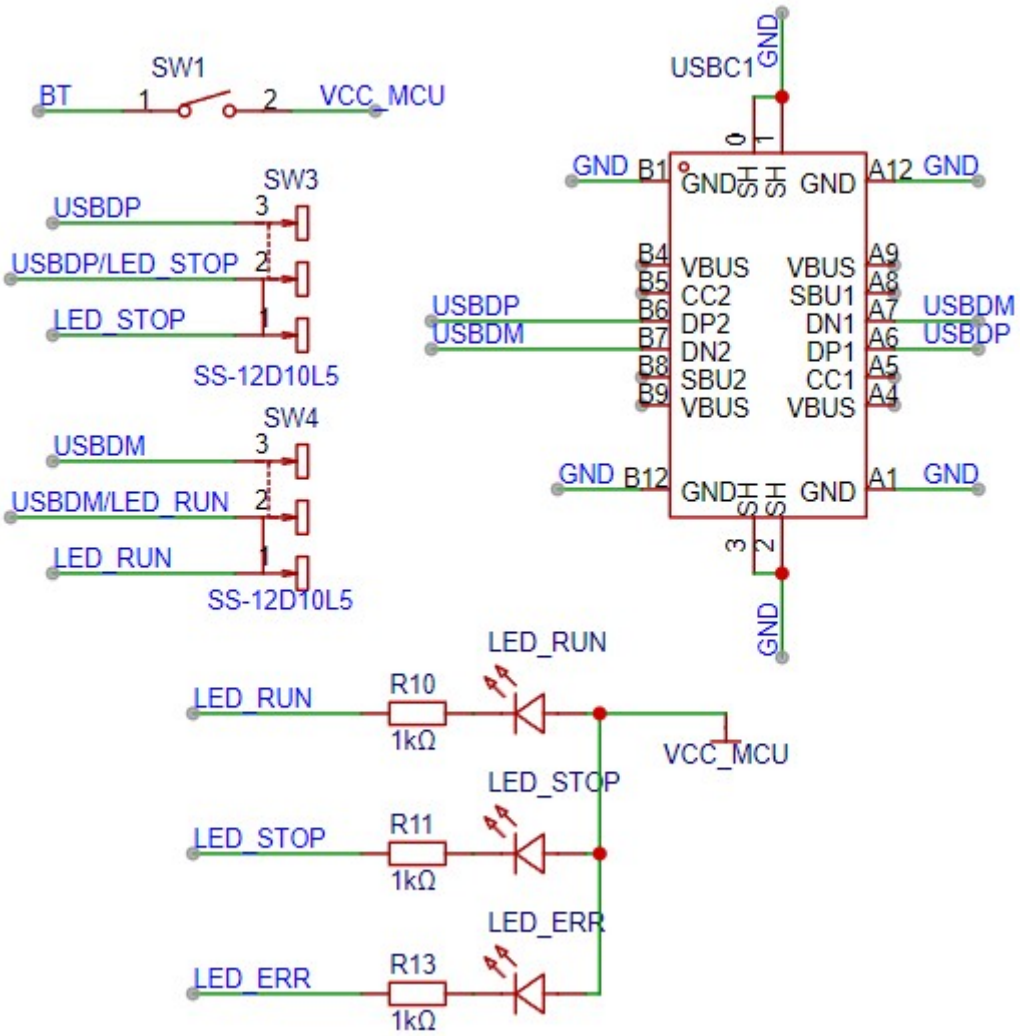


接 IP175G 电路参考图：



注：IP175G 芯片的 3.3V 电容和 1.0V 电容需尽量靠近芯片的供电管脚分布放置。

USB 升级电路参考图：



芯片升级电路建议加装一组三位开关，用于切换 LED 灯显示模式与 USB 升级模式，升级时需按住 SW1 给芯片送电，此时将进入 USB 模式，利用我们提供的升级软件进行固件升级。

周期帧结构：

上(I 区)、下(Q 区)行帧的长度在 GSDML 文件中设定，最大各 1440 字节(包含 Profinet IO 协议占用的，具体长度当超过协议规定时在导入博图后编译时博图软件会有相应提示)，MCU 需配置为 SPI 主站模式 0 去读写数据。

用户 MCU 的 SPI 主站在上电后读写数据前需要先复位 SPI 从站，具备复位条件可以用串口查询芯片信息，具体详见下面非周期帧结构的请求信息帧。

读写长度为 PLC 硬件组态时配置的 Q 区或 I 区的最大字节长度加 0x5A（帧头）+功能码 0x10+帧长度（2 字节）+净载荷数据+0xA5（帧尾）其中 Q 区或 I 区哪个大以哪个为准，超出长度用 0 填充。

下行帧：PN 主站 Q 区数据（PN 主站向 MCU 发出的数据）

- 0x5A（帧头）+功能码 0x10+帧长度（2 字节）+净载荷数据+0xA5（帧尾）

例 1：Q 区数据为 0x11, 0x12, 0x13

下行帧：0x5A, 0x10, 0x00, 0x03, 0x11, 0x12, 0x13, 0xA5

例 2：Q 区无数据时

下行帧：0x5A, 0x10, 0x00, 0x00, 0xA5

上行帧：PN 主站 I 区数据（MCU 发往 PN 主站的数据）

发送方（MCU）必须发送硬件组态时配置的 Q 区或 I 区的最大字节长度加 0x5A（帧头）+功能码 0x11+帧长度（2 字节）+净载荷数据+0xA5（帧尾）其中 Q 区或 I 区哪个大以哪个为准，超出长度用 0 填充。

- 0x5A（帧头）+功能码 0x11+帧长度（2 字节）+净载荷数据+0xA5（帧尾）

例 1：MCU 要发往 I 区的数据为 0x11, 0x12, 0x13

上行帧：0x5A, 0x11，0x00, 0x03, 0x11, 0x12, 0x13, 0xA5

例 2：MCU 无发往 I 区数据时

上行帧：0x5A, 0x11, 0x00, 0x00, 0xA5

注：因 SPI 每次读写时是以组态时的最大字节长度，建议用户将读写的数据存放在 MCU 的缓存中，方便使用时去缓存中提取数据。

非周期帧结构：（标准应用可以忽略）

1. 芯片启动后，UART 串口会一直处于等待状态，用户需要配置时需向芯片发送请求/配置信息帧：

请求信息帧结构为：0x5A（帧头）+0x01（功能码）+0xA5（帧尾）；当发送正确的请求信息帧报文后芯片将回复 26+N 字节帧，内容为 0x5A（帧头）+0x01（功能码）+数据长度(2 字节)+SPI 复位功能状态（1 字节，回复 0xFF 表示不具备复位条件，回复 0xAB 表示具备复位条件）+本机 MAC 地址（6 字节）+Vendor ID(2 字节)+Device ID(2 字节)+当前芯片的设备名称（最大 128 字节）+当前芯片的设备类别（固定 10 字节）+0xA5（帧尾）。

2. 配置信息帧结构为：0x5A（帧头）+0x02（功能码）+要配置的功能码(0x03/0x04/0x05/0x06/0x20)+0xA5（帧尾）；

当发送正确的配置帧报文后芯片将回复 0x5A（帧头）+0x02（功能码）+要配置的功能码(0x03/0x04/0x05/0x06/0x20)+(0x4F, 0x4B, 0x0D, 0x0A)表示 OK+0xA5（帧尾）；

每次发送完对应功能配置后将等待一直等待该功能码的实际数据配置，如果不需要配置则 MCU 需要回复 0x5A+功能码(0xFE)+(0x4F, 0x4B, 0x0D, 0x0A)+0xA5 后将停止 UART 串口对该功能码数据配置的等待，此时才可配置其他功能。

对于配置信息提供了 5 种功能码：

- 0x03：用于配置 Vendor ID+Device ID，芯片出厂预设的 Vendor ID 为 0x574D，Device ID 为 0x00A5；这两个参数可以改变，但必须同步修改 GSDML 文件以保持一致。

注：0x03 功能码的帧结构为 0x5A（帧头）+0x03（功能码）+Vendor ID 与 Device ID 的数据长度(2 字节)+Vendor ID 数据(2 字节)+Device ID 数据(2 字节)+0xA5（帧尾）。

例：用户需要修改的 Vendor ID 为 0x1011，Device ID 为 0x1213

MCU_TX(PN_RX)：0x5A 0x02 0x03 0xA5

MCU_RX(PN_TX)：0x5A 0x02 0x03 0x4F 0x4B 0x0D 0x0A 0xA5

MCU_TX(PN_RX)：0x5A 0x03 0x00 0x04 0x10 0x11 0x12 0x13 0xA5

MCU_RX(PN_TX)：0x5A 0x03 0x4F 0x4B 0x0D 0x0A 0xA5

- 0x04：用于配置芯片的设备类别（固定 10 字节），芯片出厂预设的设备类别为(0x50, 0x4E, 0x2D, 0x55, 0x41, 0x52, 0x54, 0x20, 0x20, 0x20)表示为 PN-UART;这个参数可以改变，最大 10 字节，无字符用 0x20 填充。

注：0x04 功能码的帧结构为 0x5A（帧头）+0x04（功能码）+设备类别数据长度(2 字节)+设备类别数据(10 字节)+0xA5（帧尾）。

例：用户需要修改的设备类别为(0x50, 0x4E, 0x2D, 0x49, 0x4F, 0x20, 0x20, 0x20, 0x20, 0x20)表示 PN-IO

MCU_TX(PN_RX)：0x5A 0x02 0x04 0xA5

MCU_RX(PN_TX)：0x5A 0x02 0x04 0x4F 0x4B 0x0D 0x0A 0xA5

MCU_TX(PN_RX)：0x5A 0x04 0x00 0x0A 0x50 0x4E 0x2D 0x49 0x4F 0x20 0x20 0x20 0x20 0x20 0xA5

MCU_RX(PN_TX)：0x5A 0x04 0x4F 0x4B 0x0D 0x0A 0xA5

- 0x05：用于请求 SlotNumber(2 字节) + ModuleIdentNumber(2 字节)，该请求帧用于获取 PLC 硬件组态时刀片式模块信息，SlotNumber 为当前组态的模块插槽位置号，此号与 ModuleIdentNumber 一一对应，ModuleIdentNumber 与硬件组态时的子模块的 GSDML 文件中 ModuleIdentNumber 一一对应，当请求了该信息后，UART 串口会将当前从站硬件组态的刀片式模块信息发出；用户 MCU 收到信息后需保存在 MCU 中用于反复核实背板总线传递过来的实际刀片式模块信息，如果有实际模块不同与组态模块，用户可将此信息通过故障帧报文通知 PN 芯片进行相应故障反馈，以此达到可在 PLC 诊断缓冲区查看具体信息以及报错位置。

注：0x05 功能码的帧结构为 0x5A（帧头）+0x05（功能码）+数据长度(2 字节)+SlotNumber 数据(2 字节)+ModuleIdentNumber 数据(2 字节)+...(最大 64 子槽的信息)...+0xA5（帧尾）；用户需注意此帧收发方式与其他功能码略有不同，详情见下面例子。

例：用户组态：

插槽 1 为 8 路 DI 模块，模块 ModuleIdentNumber 为 0x1A01；

插槽 2 为 8 路 DO 模块，模块 ModuleIdentNumber 为 0x1A02；

插槽 3 为 8 路 AI 模块，模块 ModuleIdentNumber 为 0x1A03；

插槽 4 为 8 路 AO 模块，模块 ModuleIdentNumber 为 0x1A04；

则请求报文如下：

MCU_TX(PN_RX)：0x5A 0x02 0x05 0xA5

MCU_RX(PN_TX)：0x5A 0x02 0x05 0x4F 0x4B 0x0D 0x0A 0xA5

MCU_TX(PN_RX)：0x5A 0x05 0xA5

MCU_RX(PN_TX)：0x5A 0x05 0x00 0x10 0x00 0x01 0x1A 0x01 0x00 0x02 0x1A 0x02 0x00 0x03 0x1A 0x03 0x00 0x04 0x1A 0x04 0xA5

模块参数帧（0x06）：

芯片和主站通信配置时，UART 串口会发送一个或多个模块参数帧(自定义参数)该帧在芯片与 PN 协商完成后用户请求 0x06 功能码后发出，若在未协商完成或没有模块参数帧时请求，则 PN 芯片会发出 0x5A（帧头）+0x06（功能码）+(0x45, 0x52, 0x52）表示 ERR+0xA5（帧尾）（注：该帧只有 GSD 文件中有用户参数的时候才会发出）。

每个参数帧的最大长度为 37 字节，去掉固定的帧信息，有效的净载荷为 28 字节，最多可配置 66 条参数帧；

用户请求帧为：0x5A（帧头）+0x06（功能码）+0xA5（帧尾）；

PN 芯片回复帧结构为：0x5A（帧头）+0x06（功能码）+模块参数帧总字节数量(2 字节)+输入 I 区的总数据长度（2 字节）+输出 Q 区的总数据长度（2 字节）+用户自定义模块数据帧数量(1 字节)+N*(0x5A+子功能码 0xA6+SlotNumber(2 字节)+模块参数 Index（2 字节）+对应模块自定义参数长度（2 字节）+对应模块参数（净载荷）+0xA5)+0xA5（帧尾）；

例：用户自定义模块数据帧数量为 2，I 区总长度为 10 字节，Q 区总长度为 20 字节，第一帧为 SlotNumber:0x0000，Index:0x1A02，对应模块参数:0x01,0x00,0x02,0xF1,0xF2; 第二帧为 SlotNumber:0x0001，Index:0x1A03，对应模块参数:0x02,0x02,0x03,0xF3,0xF3;

则模块参数请求帧为：

MCU_TX(PN_RX)：0x5A 0x02 0x06 0xA5

MCU_RX(PN_TX)：0x5A 0x02 0x06 0x4F 0x4B 0x0D 0x0A 0xA5

MCU_TX(PN_RX)：0x5A 0x06 0xA5

芯片正常协商完成后 MCU_RX(PN_TX)：0x5A 0x06 0x00 0x1D 0x00 0x0A 0x00 0x14 0x02 0x5A 0xA6 0x00 0x00 0x1A 0x02 0x00 0x05 0x01 0x00 0x02 0xF1 0xF2 0xA5 0x5A 0xA6 0x00 0x01 0x1A 0x03 0x00 0x05 0x02 0x02 0x03 0xF3 0xF3 0xA5 0xA5

芯片未协商完成 MCU_RX(PN_TX)：0x5A 0x06 0x45 0x52 0x52 0xA5

故障报警帧（0x20）：

芯片和主站通信时，用户 MCU 可通过 UART 串口给芯片发送故障代码，该故障需与用户自定义 GSDML 文件中相同，发完故障代码后博图/step7 的诊断缓冲区会上报相关故障信息（GSD 中设置的 SubmoduleIdentNumber 必须为 0x00000001）。

帧结构为：0x5A（帧头）+0x20（功能码）+数据长度（2 字节）+要设置故障的通道号 SlotNumber（2 字节）+要设置的通道故障号（1 字节，0x00 为恢复正常，0x0F-0x1F 为要设置的通道故障号，该值必须要与 GSD 中 ChannelDiagItem ErrorType 设置的相同）+0xA5（帧尾）,当 PN 芯片收到正确的故障帧报文后回复 0x5A（帧头）+0x20（功能码）+通道的状态（0x01 为错误的故障设置，0x02 为故障报警设置成功，0x03 为恢复故障成功）+(0x4F, 0x4B, 0x0D, 0x0A)表示 OK+0xA5（帧尾）；

例：要报故障的 SlotNumber:0x0001，要设置的通道故障号为 0x0F；

则故障帧为：

MCU_TX(PN_RX)：0x5A 0x02 0x20 0xA5

MCU_RX(PN_TX)：0x5A 0x02 0x20 0x4F 0x4B 0x0D 0x0A 0xA5

MCU_TX(PN_RX)：0x5A 0x20 0x00 0x03 0x00 0x01 0x0F 0xA5

MCU_RX(PN_TX)：0x5A 0x20 0x02 0x4F 0x4B 0x0D 0x0A 0xA5

则恢复帧为：

MCU_TX(PN_RX)：0x5A 0x20 0x00 0x03 0x00 0x01 0x00 0xA5

MCU_RX(PN_TX)：0x5A 0x20 0x03 0x4F 0x4B 0x0D 0x0A 0xA5

注：当进入故障报警帧后，串口将一直处于故障报警或恢复的查询等待窗口，如需退出要发送停止 UART 串口对该功能码数据配置的等待的报文帧。

循环周期：

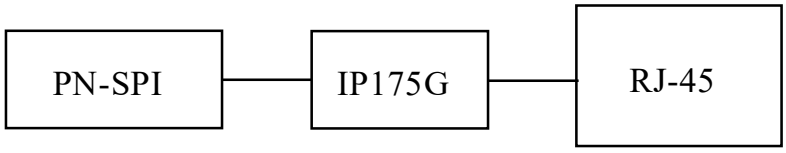
SPI 的发送周期和 Profinet 的发送周期无关，只与用户 MCU 的 SPI 主站速度有关。Profinet 侧请根据实际系统能力，修改 GSDML 文件的[MinDeviceInterval]，确保 Profinet 传输能力适配。对于 1、2、4ms 等快速循环周期，当传送数据较长时，建议用户 MCU 使用高频率 SPI 进行读写数据，最高不得超过 30M。

产品技术参数

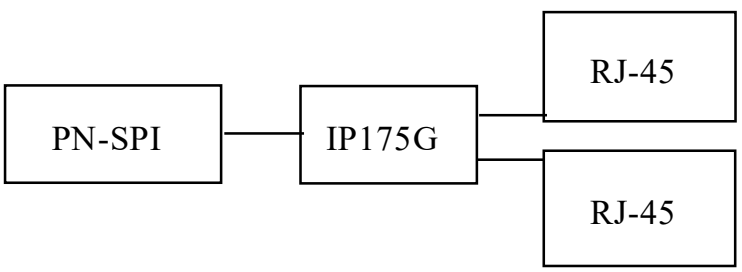
封装	QFN28
Profinet	RT_Class_1，不支持 IRT
	最快循环周期：1ms
	发送字节：最大 1440 字节
	接收字节：最大 1440 字节
	上行、下行最多 64 个 subslot
	设备名、IP 地址可利用 PN-DCP 协议设置
	诊断：支持用户自定义诊断，通过串口发送对应故障
TTL 串口与 SPI	UART 波特率：115200，1M，2M，6M（用户可选择，默认为 2M）
	SPI 最高频率：30M
	SPI 发送字节：Profinet 上行载荷（I 区数据）
	SPI 接收字节：Profinet 下行载荷（Q 区数据）
透传方式	缓存转发
工作电压	3.3V
工作温度	-40 到 85°C

电路原理框图

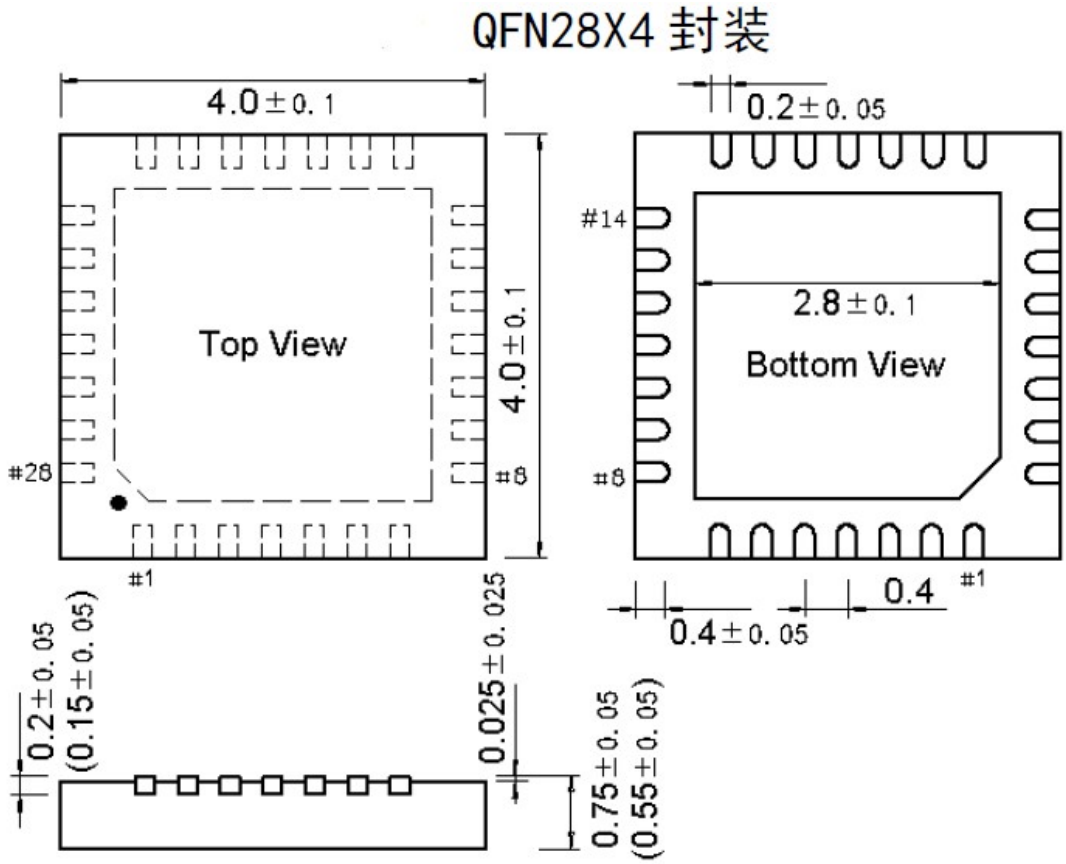
➢ 单 RJ45 配置



➢ 双 RJ45 配置



焊盘尺寸（**mm**）


QFN28X4 封装
4.0±0.1
0.2±0.05
Top View
4.0±0.1
#28
#8
#1
#14
2.8±0.1
Bottom View
#8
#1
0.4
0.4±0.05
0.2±0.05
(0.15±0.05)
0.025±0.025
0.75±0.05
(0.55±0.05)