第 10 章　“雪中，送碳”
根协议升级

10.1　“雪中送炭”

我们最初制定根协议，并将 16 位寄存器寻址空间分为 16 个段时，各段寄存器的竞争不平等就已经存在了，即：低端段天生优先于高端段。只是在初级篇中，我们只使用了 0x7000 一个段完成设备运行时的过程变量传递，这种不平等并没有显现出来。当我们将协议演绎到“生产者－消费者模型”时，我们使用了更多的段分配不同特性的过程变量，这种不平等便凸显出来。

其实，是一种强大的惯性思维影响了我们。造成上述结果的原因，是我们习惯性的在 29

位 ID 中直接截取了 16 位作为寄存器基地址，越是专业人员就越容易被惯性思维所束缚，我们需要变通。至此，亲爱的读者朋友，您是否想到了办法？而且非常“巧妙”！是的就是这样，能够挣脱惯性思维的束缚是一件非常不容易的事，恭喜您！我们将重新排布划分根协议中 29 位 ID 的排序，即在非连续位上抽取 16 位组成寄存器基地址。

10.2 根协议升级和系统信息空间的分配

如果我们定义 16 位寄存器寻址空间地址的高 4 位为“段地址”，那么正如您所想到的一样，我们将它移动到 29 位 ID 的最后发送，这样各段的竞争优势就均衡了。这是一个好主意但不完美，因为我们并不希望“配置参数空间”的竞争力也就此得到提升，从而与“过程变量”竞争。

于是我们保留“段地址”的最高位不动，将后 3 位移动到 29 位 ID 的最后发送，这样问题就圆满解决了，因为“过程变量”段地址的最高位为 0，而“配置参数”段地址的最高位为 1。

为了兼容老旧 CAN 收发器，要求 29 位 ID 的前 7 位不能连续为 1，这样原来集中在 0xF000 段尾部的不可用空间 0xFC00～0xFFFF（优先级为 1 时），由于“段地址”的移动，就被均匀分配到了 0x8000～0xF000 各段的尾部 128 个字节上 0xF80～0xFFF（优先级为 1 时），我们在使用和分配寄存器时需特别注意。

另外，为了协议的后续需求，我们将 0x0000～0x7000 各段的后 512 个字节 0xE00～0xFFF 也分割出来，作为“系统保留”，用户在使用和分配寄存器时需特别注意。

10.2.1 TC0000A001 根协议（ID

划分）

本协议是对 TTCANopen 之 TC0000A000 的升级，主要包括 ID 的划分、寄存器空间和设备编号的划分等。

1　TTCANopen 指令格式定义

TTCANopen 采用 CAN2.0b 协议扩展帧格式，其中 29 位 ID 分配见图 10－1，

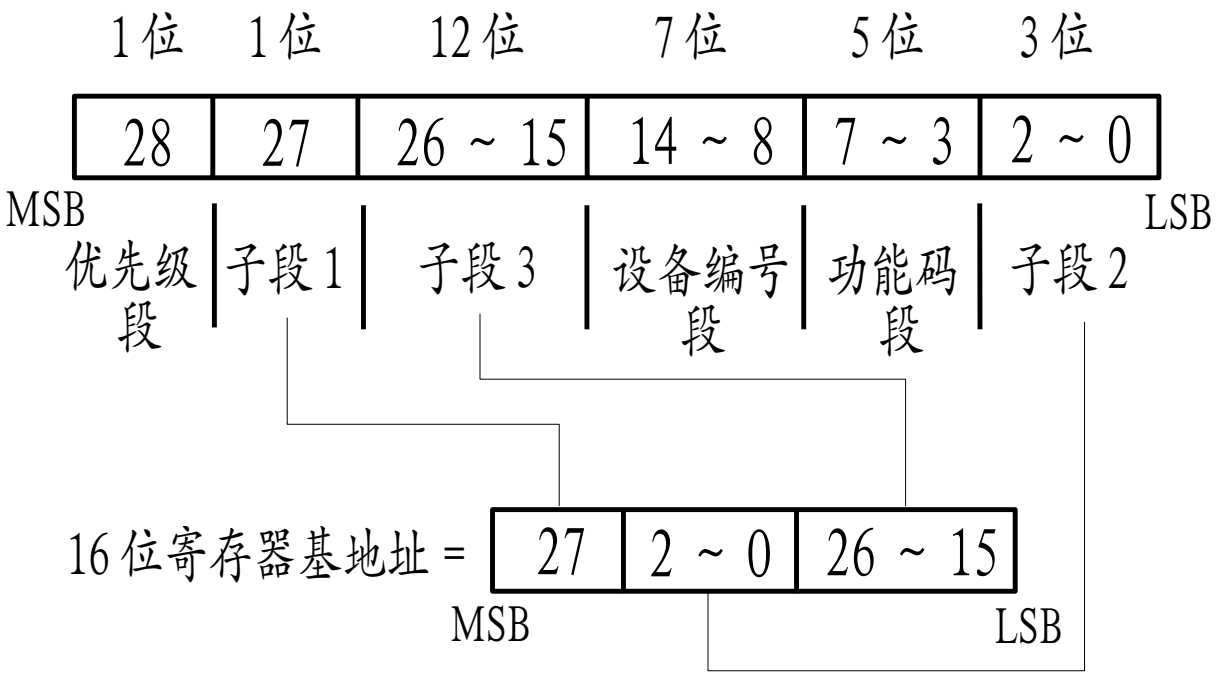

图 10－1： TTCANopen ID 划分方案

优先级段：最高位 28 位，为 1 位指令“优先级”，取值 0x0 或 0x1，值越小优先级越高，协议将 CAN 指令按紧急程度分为两类，一类是普通指令；一类是紧急指令。应用层协议使用优先级为“0x1”发送普通指令，使用优先级为“0x0”发送紧急指令。

寄存器基地址段：由第 27 位 、 2～0 位 和 26～15 位构成一个 16 位寄存器寻址参数，取值范围 0x0000~0xFFFF，每个地址对应一个字节寄存器，应用层协议将数据存放在对应寄存器中；一条 CAN 指令其数据场可携带不多于 8 个字节的数据（对于 CAN FD 协议为不多于 64 个字节的数据)，寄存器基地址标识的是 CAN 指令数据场中第 1 个数据字节的地址，其它字节的地址依序递增。

设备编号段：14～8 位，为 7 位设备在总线上的编号（或称设备地址），取值 0x00~0x7F。可提供 128 个编号，其中：0x00 为广播地址，0x01～0x03 分配给心跳器，0x04 和 0x05

分配给设备管理器，0x06 和 0x07 分配给系统主机，0x60～0x6F 分配给组地址，0x70～0x7F 系统保留。

功能码段：7～3 位，为 5 位“功能编码”，取值 0x00~0x1F，可提供 32 条指令。

基本指令格式表述：使用 CAN2.0b 协议扩展帧格式的数据帧

优先级	寄存器基地址	设备地址	功能码	DLC	0～8 个字节数据

指令格式中的优先级对应优先级段 (28 位)，寄存器基地址对应寄存器基地址段(27、2～0、26～15 位)，设备地址对应设备编号段(14～8 位)，功能码对应功能码段(7~3 位)，DLC 表示指令数据场携带的数据字节数，最后是 0～8 字节的数据场。

指令的优先级最终由 29 位 ID 值决定，值

越小优先级越高（显性电平为0）。

注：对于CAN FD标准，其对数据长度编码进行了扩展，当数据长度大于8时，使用了非线性编码，见表10－1。其指令传输数据的字节数也相应非线性增加。

表10－1 CAN FD中DLC编码对应数据长度

DLC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
数据长度值	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

2 寄存器空间的划分

协议为系统和设备定义了一个16位寄存器寻址空间，每一个地址对应一个8位寄存器，凡是需要参与或可能参与CAN总线通讯的参数和变量，都要使用或映射到该寄存器空间，寄存器空间分配见图10－2，其中：0x0000~0x6FFF定义为“系统信息空间”；0x7000~

0x7FFF 定义为“设备信息空间”；0x8000~0xFFFF 定义为“配置参数空间”。

“系统信息空间”是 TTCANopen 网络系统的全局信息映射的寄存器空间；“设备信息空间”是各设备自身信息存放的寄存器空间；“配置参数空间”是各设备自身配置参数存放的寄存器空间。

TTCANopen 在“系统信息空间”使用面向报文的通讯协议；在“设备信息空间”和“配置参数空间”使用面向节点的通讯协议。

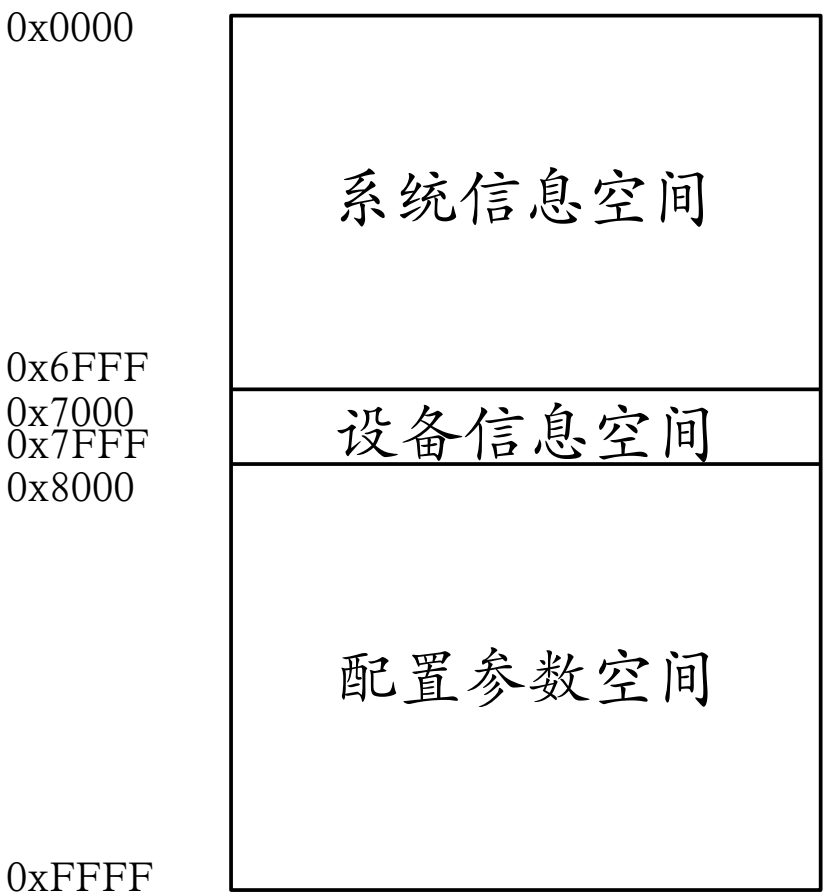

图 10－2 寄存器空间分配

进一步明确各段寄存器空间的使用规则：

0x0000 段：由 TTCANopen 组织规划，分配给共性系统过程变量使用，见 TC0001；

0x1000 段：分配给 I 特性系统信息过程变量使用，见 TC0002；

0x2000 段：分配给 O 特性系统信息过程变量使用，见 TC0003；

0x3000 段：分配给 I/O 特性系统信息过程变量使用，见 TC0004；

0x4000 段：由专业标准化组织规划，分配给 I 特性系统信息过程变量见 TC0005；

0x5000 段：由专业标准化组织规划，分配给 O 特性系统信息过程变量见 TC0006；

0x6000 段：由专业标准化组织规划，分配给 I/O 特性系统信息过程变量见 TC0007；

0x7000 段：分配给设备信息过程变量使用，

见 TC0008；

0x8000 段：分配给设备特征配置参数使用，见 TC0009。

0x9000 段：由标准化组织规划，对标准化设备特征配置参数进行统一分配，TC0010；

0xA000 段：保留；

0xB000 段：保留；

0xC000 段：保留；

0xD000 段：可编程类对象分配空间，见 TC0014；

0xE000 段：可编程过程变量资源配置和映射区，见 TC0015；

0xF000 段：由 TTCANopen 组织规划，分配给设备共性配置参数使用，见 TC0016。

其中：第 0x0000 段至第 0x7000 段为“过程变量”所使用；第 0x8000 段至第 0xF000 段为“配置参数”所使用。各段具体定义见相关“基本子协议”。

将 0x0000～0x7000 各段的后 512 个字节分

割出来，作为“系统保留”。

当优先级位为 1 时，0x0008～0xF000 各段的后 128 个字节不可用。

3　字节传送顺序

在指令中寄存器基地址，是从 29 位 ID 非连续位置抽取组合成的一个 16 位地址，用两个字节表述，如：0x1234 表示为两个字节 0x12　0x34，即：高字节在前，低字节在后。

指令携带的数据按寄存器地址先后顺序传送，由低到高，无论数据表述的是单字节数据还是多字节数据。

4　兼容性要求

为了兼容早期的协议，指令的前 7 位不能出现连续隐性电平（1）。

5　升级后对用户应用程序的影响

由于在用户应用程序中，一般都会将 CAN

指令映射成一个字节流来处理，如：
1 字节优先级、2 字节寄存器地址、1 字节设备地址、1 字节功能码、1 字节 DLC、数个字节的数据，因此根协议的本次升级对用户应用程序的影响很小，只是在收发 CAN 帧的时候，将字节流中的信息对应 29 位 ID 的位置关系进行相关的顺序变更，对应用程序的主体不产生影响。

为了方便，在表述 CAN 帧时，我们还将继续延用 TC0000A000 中的表述方法。即：

优先级	寄存器基地址	设备地址	功能码	DLC	0～8 个字节数据

由于指令中没有使用不同版本根协议的标识符，因此，网络中不能同时支持 TC0000A000 和 TC0000A001 设备的混用，两者不兼容。不过用户不用为此担心，因为这个世界上还没有一款支持 TC0000A000 协议的实用设备，它

只是为了更加清晰描述系统发展演绎过程而建立的。而实际的设备可能会是从支持 TC0000A001 开始，或更后期的协议 002 开始。需要提醒读者的是，升级后系统保留了 0x70～0x7F 地址，因此这些地址不能用于设备地址或组地址，在设备处理接收指令时需对 0x70～0x7F 地址进行甄别。

10.2.2 TC0001A000 子协议（0x0000 段的分配使用）

设备 0x0000 段寄存器分配给系统共有的通用过程变量使用，目前，我们只在该段定义了系统心跳预报帧的信息（关于“系统心跳”请参看第六章内容），该段系统信息空间的分配由 TTCANopen 组织统一制定，见下表[注]。

寄存器地址	参数类型	读写特性	参数描述	值域
0x0010	U32	周期	TTCANopen	0

～ 0x0013		发送	系统时：一天中 0.1ms 的计数	～ 863999999
0x0014	U8	周期发送	CAN 接收错误计数	0～255
0x0015	U8	周期发送	CAN 发送错误计数	0～255
0x0016 ～ 0x0017	U16	周期发送	心跳周期（单位 ms）	1～60000（可被 86400000 整除）

读写特性为“周期发送”，网络上任何设备都不能对该寄存器信息进行“读”或“写”操作。它是由指定设备周期发送到网络上的信息。

[注]：在 TTCANopen 引入双 ID 后，主机心跳预报帧寄存器地址改为 0x0A02；主机

心跳心跳帧寄存器地址改为 0x0A03。

10.2.3 TC0002、TC0003 和 TC0004 子协议（0x1000、0x2000 和 0x3000 段的分配使用）

系统 0x1000 段寄存器分配给具有“I”变量特征的系统信息过程变量使用；

系统 0x2000 段寄存器分配给具有“O”变量特征的系统信息过程变量使用；

系统 0x3000 段寄存器分配给具有“I/O”变量特征的系统信息过程变量使用。

以上三段系统信息空间分配和定义由系统设计师根据系统需求进行制定。

10.2.4 TC0005、TC0006 和 TC0007 子协议

（0x4000、0x5000 和 0x6000 段的分配使用）

系统 0x4000 段寄存器分配给具有“I”变量特征的系统信息过程变量使用；

系统 0x5000 段寄存器分配给具有“O”变量特征的系统信息过程变量使用；

系统 0x6000 段寄存器分配给具有“I/O”变量特征的系统信息过程变量使用。

以上三段系统信息空间由行业规划或专业标准的制定者进行固化定义，不同的行业对上述三段有不同的固化定义。

10.2.5 TC0010（0x9000 段的分配使用）

对于一些标准化的设备，我们希望能够对其设备特征配置参数也进行统一的标准化分配以使设备具有更好的互换性，因此 TTCAN

open 特地将 0x9000 段划分出来，供相关的标准化组织使用。为了区分不同领域不同设备的不同标准化定义内容，在 0xF000 段特地分配出 0xFBC0～0xFBC7 对定义进行说明。

另外，TTCANopen 也支持设备厂商对 0x9000 的企业标准化定义，并在 0xF000 段 0xFBC8～0xFBCF 对定义进行说明，但其不能和标准化组织的定义同时使用。

和 0x8000 段一样，TTCANopen 推荐将设备状态信息放置在 0x9A00 以后的寄存器空间。

另外，随着设备 0x9000 的标准化定义，设备对应的 0x7000 空间可随同一起被标准化。

10.3 讨论与提示

10.3.1 地址过滤

我们希望读者在学习了“生产者－消费者模型”后，尽量的去应用这个模型，将过程变量定义到“系统信息空间”，这样可以有效的

提高系统指令效率，节省总线带宽。

对于某些具有明显主从关系的系统，强制将其应用到“生产者－消费者模型”上后，一些只有主机才能控制的过程变量，被定义到系统信息空间后，网络上任何设备都可以对其进行控制，有人觉得不安全，这需要系统设计者进行全面的安全规划。此外，我们还可以利用指令中的“控制者地址”，对指令进行过滤，如那些只能接受主机控制的变量，其接受者对“控制者地址”进行过滤，凡是“控制者地址”不是 6 或 7 的，将不接受该指令的控制。如此推广，我们对“控制变量”，都可以在“接受者”一方对“控制者地址”进行选择性过滤，以提高系统的安全性。在 TC0016 中我们定义 0xF050～0xF05F 为对应“系统信息过程变量写指令的地址筛选”，每个被控设备都可以最多选择 16 个设备对其过程变量进行写操作，详见 TC00048。

当一个设备不接受某设备的控制时，会给

出错误提示指令，如：控制设备发出控制指令0x1 0x2000 0x32 0x17 0x4 0x11 0x22 0x33 0x44，被控设备认为其不具备这个权利，不接受该设备的控制，返回错误提示指令， 0x1 0x2000 0x35 0x1A 0x2 0x17 0x08 （0x8 是错误代码）。

10.3.2 新的 YTC 应用子协议的出现

随着系统模型的演绎发展，新的 YTC 应用子协议会不断的出现，在各 YTC 中支持的 TC 基本子协议和版本会有所不同。用户和厂商需要特别注意所用设备支持的 YTC 的版本。