

## 第 9 章 “柳暗，花明”

### 生产者 - 消费者模型

#### 9.1 “柳暗花明”

“山重水复疑无路，柳暗花明又一村”<sup>[1]</sup>。

在前面初级篇中，我们花了大量的篇幅介绍 TTCANopen 应用层协议中传统的“主从模式”和具有一定 CAN 特性的“全触发模式”。无论是上述哪种模式，系统中总是由一个默认“主机”和若干设备（从机）组成，信息的流向是主机到设备，或是设备到主机，并不存在设备间的通讯关联，是典型的“中心监控”系统。

然而，CAN 总线的又一重要特性是能够完美的支持“生产者 - 消费者模型”，在该模型下，总线上的设备可以是信息的生产者，也可以是信息的消费者，系统模糊了主机与从机的概念，信息是在设备之间进行传递的，并且同一信息可以有多个消费者。

如何在应用层协议中实现“生产者 - 消费者模型”，则是考验协议设计者智慧的时候了，也就是在这个时候，我们荣幸的请到了邓先生的加盟。邓先生建议我们将寄存器空间分为两个部分，一部分为“系统寄存器空间”；一部分为“设备寄存器空间”。在设备寄存器空间使用面向节点的通讯协议，实现传统的“中心监控”系统设计；在系统寄存器空间使用面向报文的通讯协议，利用“指令符”重载实现“生产者 - 消费者模型”。为此我们重新划定了寄存器空间的设计，这正如您在初级篇中“寄存器分配”一节所看到的那样。邓先生的一席话，如同黑夜里的明灯，为我们指明了前进的方向。

#### 9.2 生产者 - 消费者指令

##### 9.2.1 信息定位与流向

在 TTCANopen 协议中，所有要传递的信息是存放在寄存器中的，在初级篇中我们所涉及的信息都是存放在“设备寄存器空间”的，即“设备信息空间”（0x7000~0x7FFF）和“配置参数空间”（0x8000~0xFFFF）。信息是倚靠两个元素被唯一定位的，一个是“寄存器地址”，另一个是“设备地址”，信息是在主机和设备之间进行传递，这对于设计中心监控系统的指令集是非常适合的，但要实现生产者—消费者模型指令集的设计则有一定的困难，为此邓先生提出了“系统寄存器空间”的概念，即“系统信息空间”（0x0000~0x6FFF）。在“系统信息空间”，信息是被“系统信息寄存器地址”唯一标识定位的，它与设备地址无关，针对同一“系统信息寄存器地址”所指定的信息是同一个，而不像“设备信息空间”中的寄存器地址指向的信息是根据不同的设备而有所不同的。这样在操作“系统信息寄存器”空间时，TTCANopen 协议指令中的“设备 ID”子段，就从“定位信息”的角色中解脱出来，转换成为信息的“生产者”或“消费者”的指示子段，信息的流向也不再局限于主机和设备之间，而是面向整个网络中的所有设备，网络中任何设备都可以是信息的生产者或消费者。（在这里系统主机和设备管理器也同设备一样需要有自己的地址。我们特别将地址 0x04 和 0x05 分配给设备管理器，0x06 和 0x07 分配给设备主机使用。）

## 9.2.2 系统信息空间

我们定义 0x0000~0x6FFF 为“系统信息寄存器空间”，在设计系统时该寄存器地址对信息内容唯一定位，这些信息，我们称之为“全局信息”（相对于“设备寄存器空间”的设备私有“数据”而言）。那么这些“全局信息”具体存放在哪里呢？是不是有一个“全局寄存器”呢？回答是没有，这些信息实际上是分布在网络中各个设备上，在设备上将这些信息标识在系统寄存器空间，而这些分布在各个设备上的被标识在系统寄存器空间内容的集合，就构成了这些“全局信息”，即“系统信息”。

对“系统信息寄存器”内容的存储和操作，最终还是要归结到每一个具体的设备之上。举例来讲，某设备上有一个过程变量，如果我们将其定义在“设备信息空间”，它是由“设备信息空间”寄存器地址和设备地址共同定位的，它只能与主机进行通讯操作；如果我们将其定义在“系统信息空间”，它便由标识的“系统信息空间”寄存器地址唯一定位，网络上任何设备都可以对该标识的信息“敏感”或“操作”，而该过程变量主体还是在该设备上，并不存在一个超越设备的“系统信息寄存器”空间。

## 9.2.3 系统信息空间的划分

在“系统信息空间”（0x0000~0x6FFF）一共容纳了七个段，其中 0x0000 段分配给系统共有的通用过程变量使用，如：“系统心跳”，该段寄存器空间的分配由 TTCANopen 组织统一制定，详见 TC0001 子协议。

这里我们用 I/O 变量的特性更好说明问题。一般 I/O 变量分为 3 类，一种是“**I**”特性变量，如：典型的 A/D 采集。设备将自身获取和采集的信息发布到网络上，供其它设备分享，这些过程变量我们规定将它分配在 0x1000 段；第二种是“**O**”特性变量，如：典型的 D/A 输出。设备将带有输出控制特性的信息发送到网络上，控制其它设备，这些过程变量我们规定将它分配到 0x2000 段。另外，还有一种变量，其对应的寄存器可以被网络上的 CAN 指令改写，体现为 O 特性变量；也可以被设备本地应用程序改写，体现为 I 特性变量，我们称之为 I/O 特性变量，并将它分配到 0x3000 段。

对 0x4000、0x5000 和 0x6000 我们有同样的规定，只是这些空间属于专业领域使用。

这样划分的目的是使“系统信息空间”的格局更加清晰，更加便于各种指令的运用。

## 9.2.4 TC0040A000 子协议（有应答的生产者消费者指令）

### 指令重载与通讯模式拓展:

在初级篇中我们接触的各通讯子协议定义的“指令”都是针对“设备寄存器空间”的，其完成了 TTCANopen 协议中最基本的主机与设备（从机）间通讯模型的指令实现。本通讯子协议将“指令”作用“域”指向了“系统信息空间”，并对指令“功能码”进行“重载”定义，完成了设备间通讯的指令实现，本通讯子协议操作使用的寄存器地址空间是“系统信息空间”（0x0000~0x6FFF）。

### 1 信息发送指令

功能码：0x06

设备（信息生产者）将信息通过指定“系统信息空间”寄存器基地址发送到“网络”上，一次可发送 1~8 个字节的信息内容，第 2 个字节往后为基地址的顺延。

信息发送指令格式：

优先级	寄存器基地址	生产者地址	0x06	DLC	1~8 个字节信息
-----	--------	-------	------	-----	-----------

例：0x1 0x1100 0x35 0x06 0x4 0x11 0x22 0x33 0x44

本例表述，地址为 0x35 的设备（信息生产者）通过基地址为 0x1100 的系统信息空间寄存器向网络发送连续 4 个字节的信息，其为 0x11 0x22 0x33 0x44。

该指令由信息生产者设备触发（如告警、计时、数据变化等条件），“信息消费者”（可以是多个）在其原“指令功能码”上加 0x10，用“消费者地址”替换“生产者地址”，并使用收到信息内容进行回复。

信息发送指令应答格式：

原优先级	原寄存器基地址	消费者地址	0x16	原 DLC	原 1~8 个字节信息
------	---------	-------	------	-------	-------------

例：0x1 0x1100 0x38 0x16 0x4 0x11 0x22 0x33 0x44

0x1 0x1100 0x39 0x16 0x4 0x11 0x22 0x33 0x44。

## 2 控制信息请求指令

功能码：0x07

设备（控制信息请求者）向“网络”请求信息的原始“控制者”（只能是一个）向设备指定的系统信息寄存器基地址中填充控制信息，指令带一个字节的数据，该数据表述将要连续填充控制信息的寄存器个数  $n$ ，一般  $n=1\sim64$  个，当  $n$  大于 8 时，“网络”中控制信息的原始“控制者”需要使用多条指令响应。

控制信息请求指令格式：

优先级	寄存器基地址	请求者地址	0x07	0x1	请求信息的寄存器个数
-----	--------	-------	------	-----	------------

例：0x1 0x2200 0x35 0x07 0x1 0x04

本例表述，地址为 0x35 的设备（请求者）向“网络”请求信息的原始控制者向基地址为 0x2200 的系统信息寄存器中连续填充 4 个字节的控制信息。

网络上信息的“控制者”（只能是一个）在其原“指令功能码”上加 0x10，用“控制者地址”替换“请求者地址”，并在数据场添充相关控制信息进行回复。

控制信息请求指令应答格式：

原优先级	原寄存器基地址	控制者地址	0x17	DLC	1~8 个字节控制信息
------	---------	-------	------	-----	-------------

例：0x1 0x2200 0x26 0x17 0x4 0x11 0x22 0x33 0x44

上例表述，网络上信息的原始控制者（0x26）响应请求，向基地址为 0x2200 的系统信息寄存器连续写入 4 个字节的控制信息 0x11 0x22 0x33 0x44。

又例： 0x1 0x2200 0x35 0x07 0x1 0x0E

响应 1： 0x1 0x2200 0x26 0x17 0x8 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

响应 2： 0x1 0x2208 0x26 0x17 0x6 0x99 0xAA 0xBB 0xCC 0xDD 0xEE

### 3 控制信息写入指令

功能码：0x08

设备（控制者）向指定系统信息寄存器基地址写入控制信息，一次可写入 1~8 个字节的控制信息，第 2 个字节往后为基地址的顺延。

控制信息写入指令格式：

优先级	寄存器基地址	控制者地址	0x08	DLC	1~8 个字节控制信息
-----	--------	-------	------	-----	-------------

例： 0x1 0x2100 0x35 0x08 0x4 0x11 0x22 0x33 0x44

本例表述，地址为 0x35 的设备（控制者）向基地址为 0x2100 的系统信息寄存器中连续写入 4 个字节的控制信息 0x11 0x22 0x33 0x44。

设备（接受者）（只能是一个）在其原“指令功能码”上加 0x10，用“接受者地址”替换“控制者地址”，并使用接收信息内容进行回复。

控制信息写入指令应答格式：

原优先级	原寄存器基地址	0x18	接受者地址	原 DLC	原 1~8 个字节控制信息
------	---------	------	-------	-------	---------------

例： 0x1 0x2100 0x33 0x18 0x4 0x11 0x22 0x33 0x44

### 4 信息读取指令

功能码：0x09

设备（读取者）请求读取指定的系统信息寄存器基地址中的信息，指令带一个字节的数，该数据表述将要连续读取的信息寄存器个数  $n$ ，一般  $n=1\sim64$  个，当  $n$  大于 8 时，信息的“持有者”需要使用多条指令回复。

信息读取指令格式：

优先级	寄存器基地址	读取者地址	0x09	0x1	请求读取的寄存器个数
-----	--------	-------	------	-----	------------

例：0x1 0x1200 0x35 0x09 0x1 0x04

本例表述，“读取者”编号为 0x35 的设备请求读取基地址是 0x1200 的系统信息寄存器连续 4 个字节的信息。

信息的持有者（只能是一个）在其原“指令功能码”上加 0x10，用“持有者地址”替换“读取者地址”，并在数据场添加指令基地址指向的信息内容进行回复。

信息读取指令应答格式：

原优先级	原寄存器基地址	持有者地址	0x19	DLC	1~8 个字节信息
------	---------	-------	------	-----	-----------

假设：信息持有者（0x31）的系统全局寄存器 0x1200 0x1201 0x1202 0x1203 中分别存储的信息为 0x11 0x22 0x33 0x44。

响应为：0x1 0x1200 0x31 0x19 0x4 0x11 0x22 0x33 0x44

又例：0x1 0x2200 0x35 0x09 0x1 0x0E

回复 1：0x1 0x2200 0x31 0x19 0x8 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88

回复 2：0x1 0x2208 0x31 0x19 0x6 0x99 0xAA 0xBB 0xCC 0xDD 0xEE

## 5 信息“位与”指令

功能码：0x01

设备（控制者）用指令自带的信息与指定系统信息寄存器基地址中的信息进行“位与”操作，结果仍放在该指定系统信息寄存器中，一次可进行 1~8 个字节的操作，第 2 个字节往后为基地址的顺延。

信息“位与”指令格式：

优先级	寄存器基地址	控制者地址	0x01	DLC	1~8 个字节信息
-----	--------	-------	------	-----	-----------

例：0x1 0x2100 0x35 0x01 0x4 0x11 0x22 0x33 0x44

本例表述，控制者用 0x11 0x22 0x33 0x44 与基地址为 0x2100 的指定系统信息寄存器中的连续 4 个字节信息进行“位与”操作，结果仍放在该指定系统信息寄存器中。

假设：0x2100 0x2101 0x2102 0x2103 寄存器中分别存储的信息为 0x44 0x55 0x66 0x77，“位与”操作后的结果为 0x00 0x00 0x22 0x44。

“接受者”（只能是一个）在其原“指令功能码”上加 0x10，用“接受者地址”替换

“控制者地址”，用“位与”操作后的结果信息进行回复。

信息“位与”指令应答格式：

原优先级	原寄存器基地址	接受者地址	0x11	原 DLC	位与操作后结果信息
------	---------	-------	------	-------	-----------

返回响应为：0x1 0x2100 0x33 0x11 0x4 0x00 0x00 0x22 0x44

## 6 信息“位或”指令

功能码：0x02

设备（控制者）用指令自带的信息与指定系统信息寄存器基地址中的信息进行“位或”操作，结果仍放在该指定系统信息寄存器中，一次可进行1~8个字节的操作，第2个字节往后为基地址的顺延。

信息“位或”指令格式：

优先级	寄存器基地址	控制者地址	0x02	DLC	1~8个字节信息
-----	--------	-------	------	-----	----------

例：0x1 0x2100 0x35 0x02 0x4 0x11 0x22 0x33 0x44

本例表述，控制者用0x11 0x22 0x33 0x44与基地址为0x2100的指定系统信息寄存器中的连续4个字节信息进行“位或”操作，结果仍放在该指定系统信息寄存器中。

假设：0x2100 0x2101 0x2102 0x2103 寄存器中分别存储的信息为0x44 0x55 0x66 0x77，“位或”操作后的结果为0x55 0x77 0x77 0x77。

“接受者”（只能是一个）在其原“指令功能码”上加0x10，用“接受者地址”替换“控制者地址”，用“位或”操作后的结果信息进行回复。

信息“位或”指令应答格式：

原优先级	原寄存器基地址	接受者地址	0x12	原 DLC	位或操作后结果信息
------	---------	-------	------	-------	-----------

返回响应为：0x1 0x2100 0x33 0x12 0x4 0x55 0x77 0x77 0x77

## 7 信息“位异或”指令

功能码：0x03

设备（控制者）用指令自带的信息与指定系统信息寄存器基地址中的信息进行“位异或”操作，结果仍放在该指定系统信息寄存器中，一次可进行1~8个字节的操作，第2个字

节往后为基地址的顺延。

信息“位异或”指令格式：

优先级	寄存器基地址	控制者地址	0x03	DLC	1~8 个字节信息
-----	--------	-------	------	-----	-----------

例：0x1 0x2100 0x35 0x03 0x4 0x11 0x22 0x33 0x44

本例表述，控制者用 0x11 0x22 0x33 0x44 与基地址为 0x2100 的指定系统信息寄存器中的连续 4 个字节信息进行“位异或”操作，结果仍放在该指定系统信息寄存器中。

假设：0x2100 0x2101 0x2102 0x2103 寄存器中分别存储的信息为 0x44 0x55 0x66 0x77，“位异或”操作后的结果为 0x55 0x77 0x55 0x33。

“接受者”（只能是一个）在其原“指令功能码”上加 0x10，用“接受者地址”替换“控制者地址”，用“位异或”操作后的结果信息进行回复。

信息“位异或”指令应答格式：

原优先级	原寄存器基地址	接受者地址	0x13	原 DLC	位异或操作后结果信息
------	---------	-------	------	-------	------------

返回响应为：0x1 0x2100 0x33 0x13 0x4 0x55 0x77 0x55 0x33

## 9 信息“位移”指令

功能码：0x04

“控制者”指令携带 3 个字节的数据，第一个数据表述操作目标基地址寄存器字节个数，只能选取 1、2、4、8 中之一；第二个数据表述位移方式，0 为左移，1 为右移，2 为循环左移，3 为循环右移；第三个数据表述位移的位数（0~64）。指令将指定系统基地址寄存器中的变量，按这 3 个字节数据指定的位移要求，进行位移操作，结果仍放在该系统寄存器中，并将“位移”后的数据结果信息进行回复。

位移指令格式：

优先级	寄存器基地址	控制者地址	0x04	0x3	字节数	位移方式	移动位数
-----	--------	-------	------	-----	-----	------	------

例：0x1 0x2100 0x35 0x04 0x3 0x02 0x01 0x01

本例表述，“控制者”操作系统寄存器基地址为 0x2100 的连续 2 个寄存器中的变量进行“右移 1 位”操作，结果仍放在原寄存器中。

假设：0x2100 0x2101 寄存器中分别存储的数据为 0x12 0x31，“右移 1 位”操作后的结果



为 0x89 0x18。（请注意：寄存器的排列顺序，通常我们会将 2 字节看成 U16，4 字节看成 U32，8 字节看成 U64 处理。本例中，0x12 和 0x34，U16 的表述为 0x3112，右移一位的结果位 0x1889。）

“接受者”（只能是一个）在其原“指令功能码”上加 0x10，用“接受者地址”替换“控制者地址”，用“位移”操作后的结果信息进行回复。

位移指令应答格式：

原优先级	原寄存器基地址	接受者地址	0x14	DLC	位移操作后结果数据
------	---------	-------	------	-----	-----------

返回响应为：0x1 0x7100 0x35 0x14 0x2 0x89 0x18

注：位移指令比较特殊，很多设备可以不支持，对此只需返回 0x05 错误指令即可。

## 10 错误提示指令

功能码：0x1A

这里只针对“控制信息写入”、“位操作”控制指令提供错误检查和提示功能，当“接受者”发现对应寄存器为只读或不能接受该指令，或写入控制信息出现错误时（如：越限），发送一个带 2 个字节的错误提示指令，在“错误提示指令”中用“接受者地址”替换“控制者地址”，附录 A：列出了部分错误类型。

错误提示指令格式：

原优先级	原寄存器基地址	接受者地址	0x1A	0x2	原指令	错误类型
------	---------	-------	------	-----	-----	------

例：“控制者”发送 0x1 0x2300 0x35 0x03 0x1 0x54

而 0x2300 寄存器不接受“位异或”操作，则

“接受者”回复 0x1 0x2300 0x56 0x1A 0x2 0x03 0x05（0x05 为错误类型）

例：“控制者”发送 0x1 0x2300 0x35 0x08 0x1 0x54

假设 0x2300 寄存器的上限值为 0x50，则

“接受者”回复 0x1 0x2300 0x56 0x1A 0x2 0x08 0xFF（0xFF 为数据越界）

或用数据上限回复：0x1 0x2300 0x56 0x18 0x1 0x50

例：“控制者”发送 0x1 0x1300 0x35 0x03 0x1 0x54

而 0x1300 寄存器为只读，则

“接受者” 回复    0x1 0x1300 0x56 0x1A 0x2 0x03 0x03 （0x03 为错误类型）

11 对 CAN FD 的支持

当系统和设备需要支持 CAN FD 时，本子协议支持 CAN FD 对传输数据场的扩展。（注：大于 8 个字节的“位”操作是很少出现的。）

9.2.5 TC0041A000 子协议（无应答的生产者消费者指令）

真正的生产者 – 消费者指令:

在 TC00040A000 “指令集” 中，指令是“成对”设计的，这起源于最初的“主从模式”的需要，然而随着系统效率的需求和系统工作模式的演化发展，这种“成对”出现的指令使用形式被打破了，设备可以直接使用“应答格式”发送信息，如：在初级篇中我们见到的设备直接使用 0x19 指令向主机发送上报数据，进而演化为独立的指令，该方法最大的优点是不需进行应答，提高了系统效率。这里我们也可如法炮制，直接使用 0x19 指令“播发信息”到整个网络上去，这比使用 0x06 指令“发送信息”更简洁，而且实现了真实意义上的“生产者 – 消费者模型”。在通常意义上讲生产者只管播发“信息”，并不知道有多少消费者，如何使用该“信息”，这是一个一对多的单向传输，而对该信息“敏感”的消费者只管使用“信息”，并不作接收确认应答。而用 0x06 指令“发送信息”，并不是通常意义的“生产者 – 消费者模型”的指令。

显然使用 0x19 指令比使用 0x06 指令更加高效，而使用 0x06 指令则更加可靠，因为在对“应答”有所预期的情况下，如果没有收到预期的接收“应答”，信息的生产者可以在“应用层”界面上启动“重发”机制或做相应的处理，以提高系统执行的可靠性和变通性。

1 无应答信息播发指令

功能码：0x19

设备（信息生产者）将信息通过指定“系统信息空间”寄存器基地址播发到“网络”上，一次可传送 1~8 个字节的信息内容，第 2 个字节往后为基地址的顺延。

无应答信息播发指令格式：

优先级	寄存器基地址	生产者地址	0x19	DLC	1~8 个字节信息
-----	--------	-------	------	-----	-----------

例：0x1 0x1100 0x35 0x19 0x4 0x11 0x22 0x33 0x44

本例表述，地址为 0x35 的设备（信息生产者）通过基地址为 0x1100 的系统信息空间寄存器向网络中播发连续 4 个字节的信息，其为 0x11 0x22 0x33 0x44。

该指令由信息生产者设备触发（如告警、计时、数据变化等条件），对该信息敏感的“信息消费者”（可以是多个）使用该信息，并不做任何应答。

## 2 无应答控制信息发送指令

功能码：0x17

设备（控制者）将控制信息通过指定“系统信息空间”寄存器基地址发送到“网络”上，一次可传送 1~8 个字节的信息内容，第 2 个字节往后为基地址的顺延。

无应答控制信息发送指令格式：

优先级	寄存器基地址	控制者地址	0x17	DLC	1~8 个字节信息
-----	--------	-------	------	-----	-----------

例：0x1 0x2100 0x35 0x17 0x4 0x11 0x22 0x33 0x44

本例表述，地址为 0x35 的设备（控制者）通过基地址为 0x2100 的系统信息空间寄存器向网络发送连续 4 个字节的控制信息，其为 0x11 0x22 0x33 0x44。

该指令由控制者发送，该信息的“接受者”（只能是一个）使用该信息，并不做任何应答。

理论上“接受者”可以分时接受多个“控制者”的控制信息，可以是一个分时多对一的控制过程，但在实践中为了避免发生混乱，多数应用都是采用一对一的控制过程。

（注：在 TC00040 控制指令中同样存在可能出现的分时多对一的情况。）

## 3 错误提示指令

功能码：0x1A

这里只针对“无应答控制信息发送指令”提供错误检查和提示功能，当“接受者”发现对应寄存器不能接受该控制指令，或写入控制信息出现错误时（如：越限），发送一个带 2 个字节数据的错误提示指令，在“错误提示指令”中用“接受者地址”替换“控制者地址”，附录 A：列出了部分错误类型。

错误提示指令格式：

原优先级	原寄存器基地址	接受者地址	0x1A	0x2	原指令	错误类型
------	---------	-------	------	-----	-----	------

例：“控制者”发送 0x1 0x2300 0x35 0x17 0x1 0x54

而 0x2300 寄存器不接受该指令操作，则

“接受者”回复 0x1 0x2300 0x56 0x1A 0x2 0x17 0x05（0x5 为错误类型）

例：“控制者”发送 0x1 0x2300 0x35 0x17 0x1 0x54

假设 0x2300 寄存器的上限值为 0x50，则

“接受者”回复 0x1 0x2300 0x56 0x1A 0x2 0x17 0xFF（数据越界）

或用数据上限回复：0x1 0x2300 0x56 0x19 0x1 0x50

例：“控制者”发送 0x1 0x1300 0x35 0x17 0x1 0x54

而 0x1300 寄存器为只读，则

“接受者”回复 0x1 0x1300 0x56 0x1A 0x2 0x17 0x03（0x03 为错误类型）

## 4 对 CAN FD 的支持

当系统和设备需要支持 CAN FD 时，本子协议支持 CAN FD 对传输数据场的扩展。

## 9.3 生产者 - 消费者模型

### 1 生产者 - 消费者方式

所谓“生产者-消费者”模型，是网络中设备与设备间的一种通讯方式，信息的生产者将信息发送到网络上，网络上可以有多个对该信息敏感的消费者，是一个一对多的单向通讯过程。

而在 TTCANopen 协议中，更强调的是实现设备间通讯的多样性，其中包括“生产者-消费者模型”，也包含了一些从严格意义上讲不属于该模型的因素，这正如 TTCANopen 包含了有应答的广播指令一样，也包含了有应答的“生产者-消费者”指令，和某些一对一的控制类型的指令，并将它们一并归为“生产者-消费者模型”中，TTCANopen 是一个包容性极强的协议。

在 TTCANopen “生产者-消费者模型”中包含了多种通讯过程。

其一是对具有“I”变量特性的信息，信息的持有者用 0x19 指令将其播发到网络上，供其他设备使用。网络上的设备可根据自身需求，决定使用或不使用该信息，且不对该信息的

接收做出回复应答，见图 9-1。

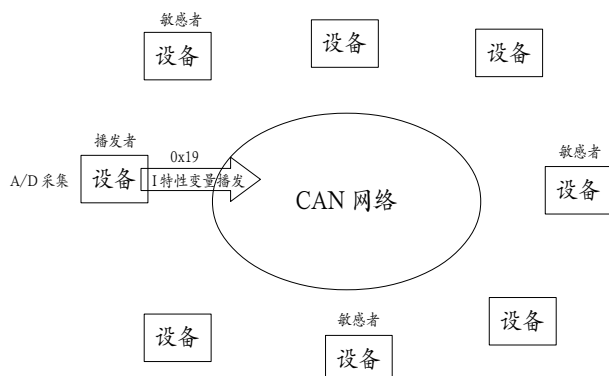


图 9-1 无接收应答的生产者-消费者通讯

其二是对具有“1”变量特性的信息，信息的持有者用 0x06 指令将其发送到网络上，供其他设备使用。网络上的设备可根据自身需求，决定使用或不使用该信息。凡是使用该信息的设备，需用 0x16 指令做出接收应答，见图 9-2。这对那些信息的生产者对其消费者有一定“预期”的系统非常有用，生产者可以根据消费者的应答情况做后续处理。

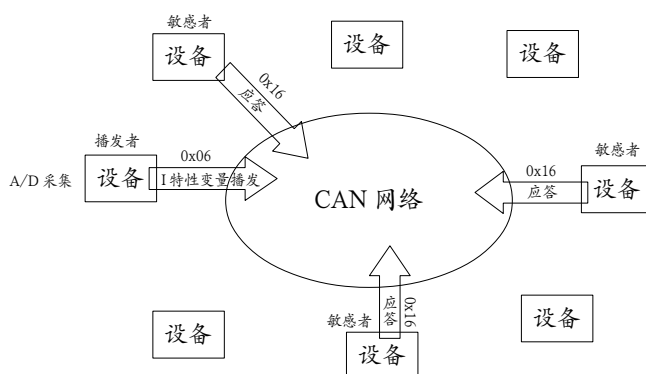
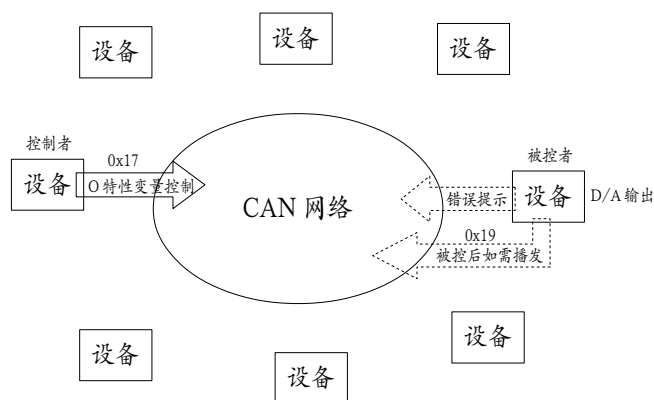


图 9-2 有接收应答的生产者-消费者通讯

其三是对具有“O”变量特性的信息，控制者对接受者的直接控制通讯过程（一对一），控制者将控制信息内容用 0x17 指令发送到网络上，接受者必须接受并使用该信息，不做接收

应答，当该信息内容越界时，发出错误提示指令，见图 9-3。当控制者企图向“I”变量特性的系统寄存器地址空间发送 0x17 指令时，对应设备应给出错误提示指令。如果系统设计者认为控制信息有必要对网络上其它设备产生影响，信息的接受者在正确受理该控制信息后，以信息持有者的身份使用 0x19 指令将信息内容播发到网络上，供其它设备使用。



9-3 控制者对接受者的无应答控制

其四是对具有“O”变量特性的信息，由接受者提出控制请求，发送 0x07 请求控制指令，控制者接收到该指令后，将控制信息内容用 0x17 指令发送到网络上（一对一），接受者必须接受并使用该信息，当该信息内容越界时，发出错误提示指令，见图 9-4。如果系统设计者认为控制信息有必要对网络上其它设备产生影响，信息的接受者在正确受理该控制信息后，以信息持有者的身份使用 0x19 指令将信息内容播发到网络上，供其它设备使用。

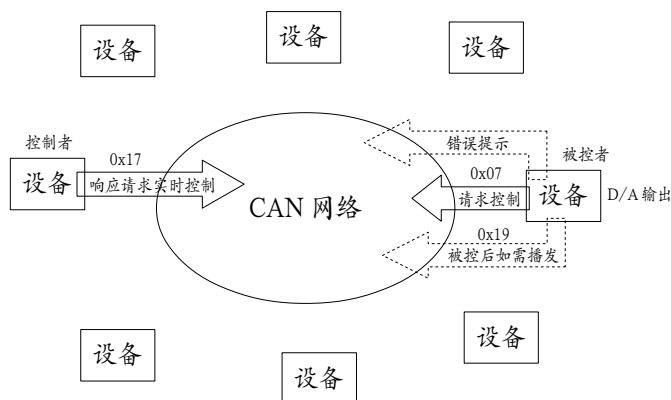


图 9-4 接受者发出控制请求的通讯

其五是对具有“O”变量特性的信息，由控制者使用 0x08、0x01~0x04 指令，将控制信息内容传递给接受者（一对一），接受者正确接受了该指令，用 0x18、0x11~0x14 指令返回接收应答，如果控制信息不被接受或越界则返回错误提示指令，见图 9-5。网络上对该信息敏感的设备可将其 0x18、0x11~0x14 应答指令，当做 0x19 播发指令对待。

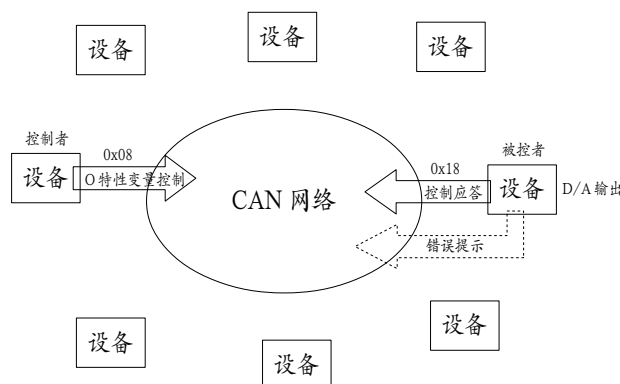


图 9-5 有应答的控制指令过程

其六，网络上某设备需要获知某系统寄存器的信息，可以通过发送 0x09 信息读取指令，信息的持有者使用 0x19 指令进行应答，见图 9-6。网络上对该信息敏感的设备可将其 0x19 应答指令当做独立播发的 0x19 指令对待。（提示：该信息不能是增量信息和切换信息等，以及重发会产生不良后果的信息，因为网络上可能会出现有多个设备用 0x09 指令读取该信息。）

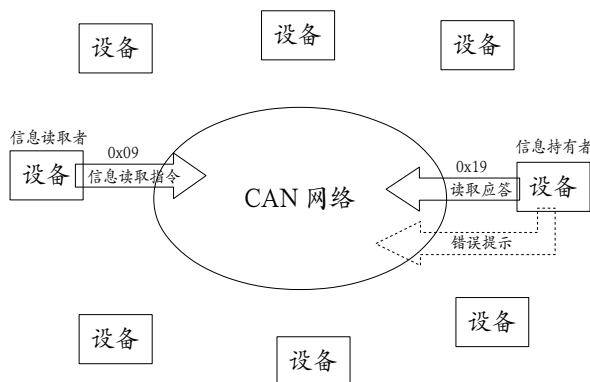


图 9-6 请求读取最新信息的指令过程

(特别提示：当网络上要传递增量信息、切换信息等，以及其它因为重发会产生不良后果的信息时，应当使用 TC0060 机制。)

## 2 设备事务处理主要内容

生产者-消费者模型，针对系统信息空间的操控，已经没有主机和从机的概念了，系统原有的主机此时也演变为一个普通设备，有自己的设备地址。下面我们只列出与系统信息相关的设备事务处理主要内容。系统设计师需要建立系统信息寄存器地址分配表，或使用行业标准的系统信息寄存器地址分配表。

- ① 将相关本地资源标识到系统信息寄存器空间；
- ② 标识设备“敏感”的系统资源变量，和相关的处理应用；
- ③ 条件触发（含接收指令触发）将本地系统资源信息上报，和处理回复指令；
- ④ 接受系统控制信息的设置和应答；
- ⑤ 截获“敏感”系统信息，运行相应的处理应用；

## 9.4 讨论与提示

### 9.4.1 高效的一对多跨设备指令

在“生产者-消费者模型”中，所说的“一对多”，一般是指“信息”层面的一对多，比如：对于一条信息播发指令，可以有多个设备对该信息敏感，而我们下面要讲的是“指令”层面的一对多。有读者马上就会联想到我们已经讲过的“广播指令”和“组广播指令”，是的，这的确是属于“指令”层面的一对多，不过，这只是针对“设备寄存器空间”才有的“广播指令”和“组广播指令”。请读者注意：针对“系统寄存器空间”是不存在“广播指令”和“组广播指令”，因为“系统信息”的定位与“设备地址”已经无关了。

在“系统信息空间”，信息是由“系统寄存器地址”唯一定位的，两个相邻的“系统寄存器地址”其信息可能会来自两个不同的设备上。例如：有两个温度传感器，其温度值分别被定义在系统信息空间的 0x1511 和 0x1512 上。那么，当我们要读取两个温度值求均时，执行信息读取指令 0x1 0x1511 0x32 0x09 0x1 0x02 将是什么结果呢？如果 0x1511 和 0x1512 来自同



一个设备，答案是明确的：0x1 0x1511 0x51 0x19 0x02 0xm 0xn。现在 0x1511 和 0x1512 来自两个设备（0x51 和 0x52），最简单的办法是将信息读取指令拆成两条指令发送，0x1 0x1511 0x32 0x09 0x1 0x1 和 0x1 0x1512 0x32 0x09 0x1 0x1。显然，这样做的效率是不高的。类似的跨设备指令还包括：0x08、0x17 和位控制指令等。

因此，为了提高效率，我们要求设备对“系统信息空间”跨设备指令提供支持。具体到每个设备，要求其对接收的跨设备指令进行“域”分析，只对和自己关联的系统寄存器空间内容进行相应的操作和应答。针对上例来讲，0x51 设备通过“域”分析，只针对 0x1511 的内容进行应答，如：0x1 0x1511 0x51 0x19 0x01 0xm；同理，0x52 设备只应答 0x1512 的内容，如：0x1 0x1512 0x52 0x19 0x01 0xn。而这对“无应答控制信息发送指令 0x17”效率就更高了。因此我们在制定系统信息排布标准时，应尽量把“关联信息”排布在相邻的系统寄存器地址上，以提高跨设备指令对“系统信息空间”的访问效率。

上述针对“系统信息寄存器空间”的指令层面的“一对多”现象，是其它 CAN 应用层协议不曾有过的。这是 TTCANopen 更具效率的表现之一。

相关跨设备一对多指令详见：TC0049 子协议。

## 9.4.2 指令比对

一种很自然的习惯使得我们自觉不自觉的就有一种相互比较的冲动，当我们定义了一个新的规则的时候，新旧比对就很自然的备受关注，很显然两套指令集都遵守了 TTCANopen ID 分配方案，所不同的是其指令作用寄存器地址空间不同。在初级篇中指令使用的是“设备寄存器”地址空间，即 0x7000~0xFFFF。这个空间是设备“私有的”，即每一个设备都可能拥有这样一个空间，对这个空间的“访问”必须和“设备地址”一起才能对信息内容具体定位，信息的流向是在主机与设备之间进行的，而设备之间是没有信息交互的，也就是说一个设备不知道另一个设备的私有寄存器空间都储藏着什么样的信息，也就无从产生信息交互。而新的协议则将指令作用域定义在了“系统寄存器空间”，即 0x0000~0x6FFF。在这个空间中信息内容是由“系统定义的”公共信息，并且是“唯一的”，任何设备如果愿意都可以根据这个公共定义知道相应的“系统寄存器”中的信息，也就是说任何设备都可根据“系统公共定义寄存器信息”选择接收使用，当然也可以发布这样的公共信息给各其他设备使用，这种方式信息是在设备之间传递，并且可以是一对多的，这就是所谓“生产者—消费者”通讯模型，在新指令集中原来的“设备 ID”也从“指定设备”用于“信息定位”的角色中解脱出

来，而用于表述信息的“生产者或消费者”。

新的指令集中延用了原来的“功能码”标识，指令根据寄存器作用域不同，其“功能码”代表的含义有所不同，这有点象 C++ 中的“运算符重载”，而“系统寄存器”与“设备寄存器”则更具有“全局变量”与“私有变量”的味道。

“生产者-消费者模型”其信息定义在“系统寄存器空间”，信息是在设备间传递的，且可以是“一对多的”，不需要中心监控主机“转手”，系统的“实时性”和“总线效率”进一步提高。

“生产者-消费者模型”是一种正在成长的模式，它被 CAN 总线完美的支持，如果所设计的“应用层协议”不能够支持“生产者-消费者模型”，将会失去其生命力。

### 9.4.3 面向节点与面向报文协议的讨论

现场总线通讯协议大致可分为两类，一是“面向节点的协议”，二是“面向报文的协议”。

面向节点的协议，数据交换是基于设备节点寻址的，帧格式中明确包含了目的地址和源地址，通讯的源宿是明确的，信息格式内容是确定的，大多数传统的现场总线通讯协议都是基于节点的。在初级篇中我们所涉及的“主从模式”、“上位机下位机模式”和“全触发模式”都是基于节点的，通讯默认在主节点与从节点之间进行，因此在协议中只需要标明从机节点地址即可，“设备寄存器地址”则是每个从节点上数据存放的标识。

面向报文的协议，数据交换是建立在报文标识基础上的，帧格式中包含唯一的报文标识符，该标识只是对报文“信息”的一种标识，而不标明通讯的目的地址，由总线上的各节点单独决定是否接受传输报文，由此建立了“生产者与消费者”模型。在本章中我们新制定的子协议就是典型面向报文的通讯协议，通讯帧中“系统寄存器地址”实际上就是一种“报文标识符”，它定义了该帧的“信息内容”，而“设备 ID”字段仅仅表示该帧是“谁”发送到总线上的，根据通讯内容性质的不同，该字段可以是“信息”的“生产者”或“消费者”。

从面向节点到面向报文，是从传统通讯理念到现代通讯理念的发展过程，这两种通讯协议并不互相排斥，在同一系统或同一总线上两者是可以并存互补的。面向节点的协议一般需要接收应答，是一对一的，可靠性高；面向报文的协议一般无需接收应答，是一对多的，通讯效率高。

TTCANopen 协议在不改变指令格式结构的状况下，巧妙的利用了不同“寄存器空间”和

“功能码”重载，在应用层上实现了面向节点和面向报文两种协议的无缝衔接，为用户设计可靠而高效的系统提供了便利。在这里再一次感谢邓先生用他的智慧帮助我们闯过了浅滩，同时也为协议的使用者们祈福，邓先生加盟的第一条件就是“协议必须免费使用<sup>[2]</sup>”，我们热切的盼望着有更多的志愿者加入我们的事业，为 TTCANopen 协议的不断完善提供宝贵的意见。

#### 9.4.4 分布式处理

面向报文的生产者-消费者模型，提供了设备间的通讯途径，和可闭环的控制回路，在不需要系统主机参与下，将各种控制功能分散到各种设备中，实现了真正意义上的分布式控制，提高了系统效率和配置重组的灵活性，是现场总线控制系统的发展方向。

#### 9.4.5 生产者 - 消费者模型的推广

对过程变量信息的传递，使用生产者-消费者模型是一个非常好的选择，它不但提高了 CAN 总线效率，还能够加强系统设计者的全局观。我们希望读者在读完本章后，尽量使用生产者-消费者模型去设计自己的系统。

我们在系统信息空间保留了 0x4000、0x5000 和 0x6000 段，作为行业标准使用，各行业如汽车、医疗仪器、纺织机械，具体到各种类型的设备，如：吊车、SUV、卡车、CT、B 超、印花机、滚梯、直升梯、机器人等，都可以在保留段中将行业或设备涉及的过程变量进行固化定义，形成行业和相关产品的应用标准，为设备的兼容、互换以及维护维修带来最大的便利，最大限度的降低用户端的使用成本。

然而，一般的设备厂商却更加喜爱隐匿、独占和不兼容，标准与非标的斗争还会一直持续下去。

#### 9.4.7 一个迫切需要解决的问题

一条指令在网络上的竞争能力，首先由它的优先级位确定，然后就是寄存器地址，寄存器地址越低其竞争能力就越强，当我们将 16 位寄存器地址空间分为 16 个段，并将不同性质的过程变量分配到不同的段（0x0000 段~0x7000 段）中时，处在前面的段则具有了先天的竞争优势，这显得有些不公。举例来讲，处于 0x1000 段最后端地址的播发变量，也比处于 0x2000 段前端地址的控制变量竞争力强，由此处于 0x7000 段的设备信息过程变量竞争力则最

弱（这里不考虑配置参数空间），这是我們不想看到的，我們希望传递过程变量的各段，具有相同的竞争力。

现在合上书，想一想，您有什么“巧妙”的方法帮我们解决这一问题吗？

注：〔1〕出自南宋陆游的《游山西村》。

〔2〕通过艰苦的谈判，邓先生终于同意我们对大批量商用进行收费，并将收费起点定在年产一万件以上部分，邓先生的理想是将 TTCANopen 做成国家标准，助力中国 2025。