

CANopen DS301 当中规定了三种模式用来实现主从之间相互监控对方是否在线，分别是消费者心跳保护报文 consumer heartbeat、生产者心跳保护报文 Producer Heartbeat、以及节点寿命保护报文 life guard。

## 一、简介

在 CANopen 系统中，由于存在总线掉线的可能，规定了 consumer heartbeat, Producer Heartbeat 以及 life guard 三种方式来实现。在 CANopen 系统中，控制器作为 CANopen 的主站，同时也是信息的消费者。

Producer heartbeat 是指从站节点向 CAN 总线上定期发送自身的生产的心跳报文，主站在收到心跳报文之后，刷新从站节点是否在线的状态。

Consumer heartbeat 是指主站向 CAN 总线上定期发送自身的生产的心跳报文，从站节点在收到心跳报文之后，刷新主站是否在线的状态。

Life guard 是指从站节点在收到主站发给自身的 life guard 请求报文之后，向 CAN 总线上发送 life guard 回应报文，从站和主站都可以刷新双方的在线状态。其中，life guard 可以单独使用，Producer heartbeat 和 consumer heartbeat 可以单独使用或者两者组合一起使用。

## 二、Producer Heartbeat

Producer Heartbeat 功能使用对象字典中的 1017h 索引，具体定义如下对象 0x1017: Producer Heartbeat Time(生产者心跳报文时间)这个对象定义了设备心跳报文发出的时间间隔，当该值设为 0 时，不发出心跳报文。

对象描述：

Index	0x1017
Name	Producer Heartbeat Time
ObjectCode	VAR
Data Type	UINT16
Access	RW
PDO	NO
Units	ms

使用的流程如下：

使用 SDO 向 1017 索引中发送从站节点产生心跳报文的时间，例如设置为 1000ms，需要发送的 SDO 数据如下

向索引 1017h 子索引 00h (INT16) 中写入数据 03E8h (1000)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
命令	2Bh	17h	10h	00h	E8h	03h	00h	00h
回答	60h	17h	10h	00h	00h	00h	00h	00h

从站节点的 1017h 索引之内为非 0 的数据之后，会向 CAN 总线上发送 Producer Heartbeat 报文，具体格式如下。

Slave→Master

COB-ID	Byte
0x700+Node_ID (从站 ID)	Bit7: toggle; bit6-0: 状态

位 0 到 6 (bit0~bit6) 表示节点状态，可为下表中数值。

Value	状态
0	Boot-up:(节点首次启动)
4	Stopped:(节点停止)
5	Operational:(节点处于操作状态)
127	Pre-operational:(节点处于预操作状态)

本例中，从站节点会在小于 1000ms 的时间窗口内，发送 CAN\_ID 为 700+从站节点 ID 的心跳报文，主站在收到该报文之后，将从站节点更新状态为在线，如果在 1000ms 的时间窗口内，收到的该从站节点的心跳报文少于 1 条，则认为从站节点掉线。

**注意：该功能主要是用于主站对于从站的监控。如需双向监控，需要与 consumer heartbeat 功能配合使用。**

### 三、 consumer heartbeat

consumer Heartbeat 功能使用从站节点对象字典中的 1016h 索引，具体定义如下对象 0x1016: consumer Heartbeat Time(生产者心跳报文时间)。

这个对象定义了设备心跳报文发出的时间间隔，将根据“消费者心跳时间间隔”参数（索引 1016h）中所设置的周期和 ID 来监控网络中其余站点产生的心跳报文，一般是主站产生的心跳报文，用来确定主站的状态，该周期以 ms 为单位，如果超过该周期，从站节点将会产生通信系统故障报警。

Bits	31-24	23-16	15-0
Value	reserved (value: 00h)	Node-ID	Heartbeat time

对象描述：

Index	0x1016
Name	consumer_heartbeat
Object Code	VAR

目录描述：

Sub-Index	0
Description	Number of entries(条目数量)
Access	RO
Data Type	UINT8
PDO Mapping	NO
Units	--
Value Range	1
Default	1

Sub-Index	1
Name	consumer_heartbeat
Data Type	UINT32
Category	Mandatory
Access	RW
PDO Mapping	NO
Units	--
Value Range	0~4294967295
Default Value	0

使用的流程如下：

在使用 consumer\_heartbeat 功能时，主站必须要有一个非 0 的 ID，例如，在本例中，认为主站的 ID 为 127，主站产生的心跳报文的时间为 1000ms。使用 SDO 向 1016 索引的 01 子索引中发送主站的 ID 和主站

心跳报文的生成时间

向索引 1016h 子索引 01h (UINT32) 中写入数据 03E8h (1000)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte65	Byte6	Byte7
命令	23h	16h	10h	01h	E8h	03h	7Fh	00h
回答	60h	16h	10h	01h	00h	00h	00h	00h

从站节点会在收到 CAN\_ID 为 700h+7Fh 的报文后，将主站刷新为在线状态，在 1000ms 之内，如果没有收到新的主站的心跳报文的情况下，则认为掉线，并发出“系统故障”的报警，同时释放电机。

注意：该功能主要是用于从对于主站的监控。如需双向监控，需要与 Producer Heartbeat 功能配合使用。

#### 四、 life guard

通过使用 life guard 功能，主节点可以检查每个节点当前的状态，当这些节点没有数据传送时尤其有意义。

Master 主节点发送远程帧（无数据）如下：

Master→Slave

COB-ID
0x700+Node_ID (从站 ID)

Slave 从节点发送如下报文应答

Slave→Master

COB-ID	Byte
0x700+Node_ID (从站 ID)	Bit7: toggle; bit6-0: 状态

数据部分包括一个触发位 (bit7)，触发位必须在每次节点保护应答中交替清“0”或者置“1”。触发位在第一次节点保护请求时设置为“0”。位 0 到 6 (bit0~bit6) 表示节点状态，可为下表中数值。

Value	状态
-------	----

0	Boot-up:(节点首次启动)
4	Stopped:(节点停止)
5	Operational:(节点处于操作状态)
127	Pre-operational:(节点处于预操作状态)

对象 0x100C: Guard Time(保护时间)

这个对象描述了主节点发送远程帧（无数据）的周期值。

Index	0x100C
Name	Guard Time
Object Code	VAR
Data Type	UINT16
Access	RW
PDO Mapping	NO
Units	ms
Value Range	0

对象 0x100D: Life Time Factor (生命因子)

这个对象描述了从节点应答主节点发送远程帧的最大时间，如果在 Guard Time\*Life Time Factor 这段时间内，从节点没有应答，则认为设备掉线。

Index	0x100D
Name	Life Time Factor
Object Code	VAR
Data Type	UINT8
Access	RW
PDO Mapping	NO
Units	---
Value Rang	

使用的流程如下：

使用 SDO 向 100Ch 索引中发送节点保护寿命时间，例如为 200ms，从站节点从站地址为 1。

向索引 100Ch 子索引 00h (INT16) 中写入数据 00C8h (200)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
命令	2Bh	0Ch	10h	00h	C8h	00h	00h	00h
回答	60h	0Ch	10h	00h	00h	00h	00h	00h

使用 SDO 向 100Dh 索引中发送节点保护寿命时间因子，例如为 5，  
向索引 100Dh 子索引 00h (INT16) 中写入数据 5h (5)

	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
命令	2Fh	0Dh	10h	00h	05h	00h	00h	00h
回答	60h	0Dh	10h	00h	00h	00h	00h	00h

整个寿命保护周期为  $200\text{ms} \times 5 = 1000\text{ms}$ 。

设置完成后，主站向从站节点发送 ID 为 701 的远程帧，从站节点会回复主站一个 ID 为 701 的本地帧，数据内容为 Bit7: toggle; bit6-0: NMT 状态。

主站在收到从站节点的回复信息后，将该 ID 的从站节点刷新成在线的状态，主站发送了 life guard 远程帧之后，如果收不到从站节点的回复的本地帧，则可以认为从站节点掉线。同样，从站节点在整个生命周期以内，如果收不到主站发送过来的 ID 为  $0x700 + \text{从站节点 ID}$  的远程帧，则认为 CAN 总线以掉线，从站节点报“系统故障”报警，同时释发电机。

注意：life guard 为双向的，主站可以根据从站节点有无回复 life guard 问询帧来判定从站节点是否在线，从站节点也可以根据主站在生命周期时间内是否收到主站的 life guard 问询帧来判定是否与主站通信成功。

在启动了 life guard 之后，只有在从站节点接收到第一帧 life guard 问询帧之后，才会触发生命周期保护，如果在上电以后主站一直没有发送 life guard 问询帧，那么从站节点不会进入生命周期保护模式，并且不会触发报警。

## 五、总结

消费者心跳保护报文 consumer heartbeat，生产者心跳保护报文 Producer Heartbeat，以及节点寿命保护报文 life guard 三种可以同时使用，但是为了节省总线资源，一般会使用一种或其中的两种。

对于三种的优缺点等如下：

生产者心跳保护报文 Producer heartbeat 只是针对于主站查询从站是否在线，而从站不能知道主站是否在线。

消费者心跳保护报文 consumer Heartbeat 只是针对于从站查询主站是否在线，而主站不能知道从站是否在线。并且，在使用消费者心跳保护报文 consumer Heartbeat 的时候，主站的 ID 不能为 0。（在很多 CANopen 系统中主站的 ID 为 0）。所以消费者心跳保护报文 consumer heartbeat 和生产者心跳保护报文 Producer Heartbeat 可以同时使用，并且该种使用为非回复式，程序实现简单，但是实时性不高，并且需要主站一定要拥有与从站不同的 ID。另外，使用消费者心跳保护模式的时候，从站节点可以监控另外一台从站节点的状态，例如，1 号从站节点监控 2 号从站节点的心跳报文，并且，一个从站节点只能设置监控一个其余的站。

life guard 报文同时有针对主站和从站的查询机制，但是由于主站发送给从站的 life guard 询问帧为远程帧，优先级比较低，在系统通信量很大的时候可能会被滞后发送或者丢弃，只能在总线不存拥堵或者完全占满的情况下使用（负载率小于 70%）。