

CBCAN

CANopen 通讯模板使用说明书

V1.4

2012/10/3

目錄

目錄.....	2
表目錄.....	4
圖目錄.....	5
1. 模板介绍	7
2. 规格	7
3. 硬件安装及配线	8
4. PLC 应用界面	9
4.1 通讯接口区.....	9
4.2 应用参数区(Parameter data).....	9
4.3 程序数据区(Process data).....	9
4.4 模板状态区.....	10
5. LED 状态指示灯	11
6. EasyCANopener 软件操作	12
6.1 PLC 连接.....	13
6.2 组态设定.....	13
6.2.1 建立新组态.....	13
6.2.2 储存组态至档案.....	18
6.2.3 开启已建立的组态档.....	18
6.2.4 读取 CBCAN 模板的组态内容.....	18
6.2.5 将组态资料写入 CBCAN 模板.....	19
6.2.6 产生电子数据文件.....	19
6.2.7 结束组态设定.....	20
6.3 SDO 数据存取操作.....	20
6.3.1 SDO 数据读取.....	20
6.3.2 SDO 资料写入.....	20
6.3.3 SDO 资料批处理.....	21

6.4	PLC 远程监控服务	23
6.4.1	开启服务	23
6.4.2	使用服务	24
6.5	NMT(网络管理)操作	24
6.6	CBCAN 模板韧体更新操作	25
6.6.1	选取韧体档案	25
6.6.2	开始韧体更新	25
6.7	结束操作.....	25
7.	CBCAN SDO 和 NMT 服务的 PLC 缓存器界面.....	26
7.1	SDO 服务.....	26
7.2	NMT 服务.....	27
7.3	范例 – NMT 服务 (Start Remote).....	28
附录一	30
1.1	应用手册.....	30
1.1.1	范例一：将 CBCAN 连接至一个 CANopen 主站装置	30
1.1.2	范例二：将 CBCAN 连接至一个 CANopen 从站装置	35
1.1.3	范例三：两台 FBs PLC 透过 CBCAN 模块互相连接.....	41
附录二	45
	CBCAN 模板的对象字典(Object Dictionary).....	45

表目錄

表 1: CBCAN 规格.....	7
表 2: CBCAN 端子台讯号.....	8
表 3: 程序数据区.....	9
表 4: 模板状态区.....	10
表 5: RUN 灯模式.....	11
表 6: ERR 灯模式.....	11
表 7: RUN 灯模式 - Single status indicator.....	12
表 8: PLC Registers for SDO Service.....	26
表 9: Registers Status after SDO Service Execution.....	26
表 10: SDO Error Code.....	27
表 11: 缓存器設定值 - NMT 服务:Start Remote.....	28
表 12: 范例 - 网络设定.....	30
表 13: 范例 - 网络组态.....	36
表 14: 主站及从站组态表.....	41

圖目錄

图 1: CBCAN 上视图.....	8
圖 2: FBs 上視圖	8
图 3: CANopen line termination	8
图 4: Term jumper 位置.....	9
图 5: 灯号闪烁时序.....	11
图 6: EazyCANOpener 主画面	12
图 7: 通信设置.....	13
图 8: 组态设定	13
图 9: RPDO 设定.....	14
图 10: RPDO 突出式选单.....	14
图 11: 编辑 RXPDO	15
图 12: 传输模式.....	16
图 13: TXPDO 设定页面	16
图 14: 杂项设定页面	17
图 15: 读取组态的通信设置.....	18
图 16: 节点扫描.....	19
图 17: 搜寻结果窗口	19
图 18: 产生 EDS 档案.....	19
图 19: SDO 服务页面	20
图 20: SDO Read 页面	20
图 21: SDO Write 页面.....	21
图 22: SDO 批处理	21
图 23: SDO 批处理设定页面	21
图 24: 批处理突出式选单.....	22
图 25: 增加 SDO 命令	22
图 26: PLC 远程监控服务.....	23
图 27: Gateway 状态窗口	23
图 28: WinProladder 联机设定页面.....	24
图 29: NMT 服务	24
图 30: 韧体更新.....	25
图 31: 韧体信息.....	25
图 32: NMT Start Remote command - 阶梯程序范例.....	29
图 33: 范例网络.....	30
图 34: 主页面.....	31
图 35: 模块设定画面.....	31
图 36: 节点 ID 设定	31
图 37: 速率设定	31

图 38: 删除 RXPDO	32
图 39: RXPDO 设定页面	32
图 40: 此应用的 RXPDO1 设定	33
图 41: 移除 TXPDO	33
图 42: TXPDO 设定	33
图 43: TXPDO 传输参数设定	34
图 44: 将组态写入 CBCAN.....	35
图 45: 范例网络	35
图 46: 主页面	36
图 47: 模块组态页面	36
图 48: 节点 ID 设定	37
图 49: 速率设定	37
图 50: 移除 RXPDO	37
图 51: RXPDO 设定页面	38
图 52: 此范例中的 RXPDO1 设定	38
图 53: 移除 TXPDO	39
图 54: TXPDO 设定	39
图 55: TXPDO 传输参数设定	40
图 56: 将组态写入到 CBCAN.....	40
图 57: FBs-24MC 和 FBs-20MC 点对点通讯.....	41
图 58: FBs-24MC RXPDO 设定	42
图 59: FBs-24MC TXPDO 设定	42
图 60: FBs-24MC Misc. 设定	43
图 61: FBs-20MC RXPDO 设定	43
图 62: FBs-20MC TXPDO 设定	43
图 63: FBs-20MC Misc. 设定	44
图 64: FBs-24MC 及 FBs-20MC 的 Ladder 程序	44

CBCAN 模板使用说明书

1. 模板介绍

CBCAN 模板是永宏 FBs PLC 系列的通讯模板.使用时可直接装在 CPU 模块的扩充板位置即可不占用额外空间.透过此模板永宏 PLC 即能与 CANopen 网络上的其它装置直接沟通.

CANopen 是一种架构在 CAN bus 网络上的通讯协议,已广泛的应用在各种控制系统中,例如,工业机械控制、车辆控制系统、工厂自动化、医疗设备控制、大楼自动化、远程数据收集和监控、环境监测... 等等.具有安全可靠及反应迅速的优点.

2. 规格

表 1: CBCAN 规格

名称	特性	
依循标准	CAN 2.0A , DS301 V4.02	
PDO 数目	RPDO	最大 10 个
	TPDO	最大 10 个
SDO 数目	Server	1 个
	Client	1 个
应用参数对象	最多 1000 个缓存器	
同步 Master	可规画	
NMT Master	可操作	
Time Stamp	Consumer	
错误控制	Heartbeat	
通讯速率	20K, 50K, 125K, 250K, 500K, 750K, 1M 可设定	
组态设定	远程操作	可
	工具	PC 软件 EasyCANopener
	方法	透过 PLC 通讯端口进行修改
远程 PLC 程序修改	可	
Vendor ID	2EFH	
信号端子	3 Pin 免螺丝端子	
电气隔离	是	
电源电压及耗电流	5V, 150mA	
工作温度	0 ~ 60 °C	
储存温度	-20 ~ 80 °C	

3. 硬件安装及配线

FBs PLC 主机左侧有通讯板专用的扩展槽，CBCAN 通讯板可直接安装于此扩展槽。

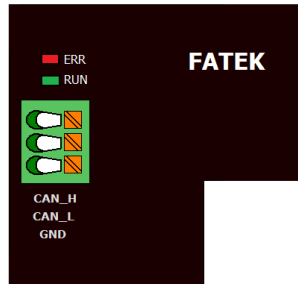


图 1: CBCAN 上视图

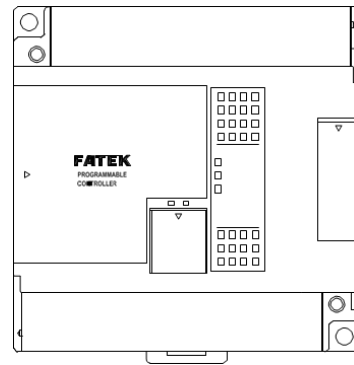


圖 2: FBs 上視圖

CBCAN 模块使用的是三线式免螺丝的端子台，其三条讯号线定义如下：

表 2: CBCAN 端子台讯号

Pin	Signal	Description
1	CAN_H	CAN_H bus line (dominant high)
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_GND	Ground / 0V / V-

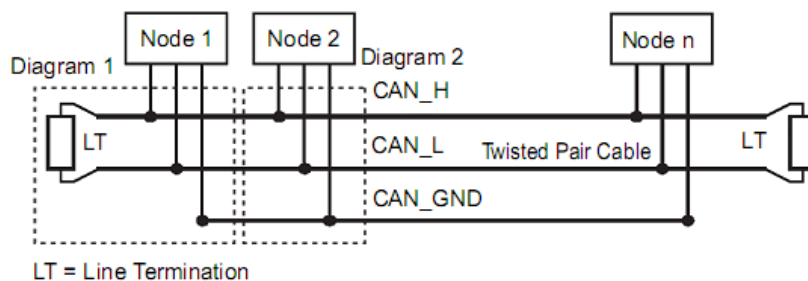


图 3: CANopen line termination

如上图所示，通常在网络的最左侧及最右侧的节点必须在 CAN_H 和 CAN_L 讯号线加上 120Ω 1/4W 终端电阻以确保讯号质量，但 CBCAN 模块的用户只要将下图位于 CBCAN 模块内标示“Term”的 jumper(JP4)短路，就具备了终端电阻的功能。

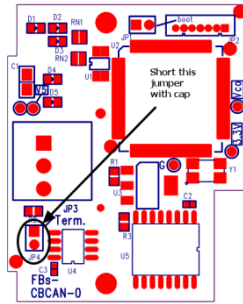


图 4: Term jumper 位置

4. PLC 应用界面

PLC 与 CBCAN 模板间的沟通系利用 PLC 的缓存器来达成.此用于沟通之缓存器基本上区分为以下几区.

4.1 通讯接口区

范围为 R3700~R3769 共 70 个缓存器. PLC 应用程序请勿重复使用此区的缓存器.组态规画工具 EasyCANopener 利用此缓存器区来与 CBCAN 沟通.

4.2 应用参数区(Parameter data)

范围可规画.最大可使用 1000 个缓存器.此区的缓存器从网络端看来相当该控制节点下的对象.参数区内的缓存器均有对应之索引(Index)及副索引(SubIndex).应用时系透过 SDO 通讯来进行存取.控制器内比较不常变化的数据可放置于此区.如此当机器于安装调整阶段时可轻松的透过网络来加以设定.

4.3 程序数据区(Process data)

范围从 R3600~R3679 共 80 个缓存器. PLC 的实时控制数据是透过此缓存器区与其它节点的控制器交换.

表 3: 程序数据区

Sequence	Item	Function	Register	
1	TPDO1	Word #1	R3600	
2		Word #2	R3601	
3		Word #3	R3602	
4		Word #4	R3603	
5	TPDO2 ~ TPDO10		R3604	
~			~	
40			R3639	
41		RPDO1	Word #1	R3640
42			Word #2	R3641
43			Word #3	R3643
44	Word #4		R3644	
45	RPDO2 ~ RPDO10		R3645	
~			~	
80			R3679	

每个 PDO(程序数据对象)可个别设定其对应的数据长度(0~4,以一个缓存器为最小规画单位).当长度小于 4 时会先使用前面的缓存器.举例来说

TPDO1 若长度设为 2 则 R3600,R3601 会被使用而 R3602,R3603 不会被使用。
每个 PDO 对应的起始缓存器号码为固定,不受前面的 PDO 数据长度影响。
此范围没使用到的缓存器,应用程序可任意应用。

4.4 模板状态区

表 4: 模板状态区

Sequence	Register	Function	
1	R3680	Module Status	Low Byte: Bit 0 : =0, Normal =1, Stopped when excessive RX error occur while startup. Bit 1: Sync signal time-out, valid only if at least one TXPDO is configured in sync. mode. Bit 2: Reserved Bit 3: =1, CAN Rx error Bit 4: =1, CAN Tx error High Byte: Bit[15:8] CBCAN state. =0, init. =5, OPERATIONAL. =4, STOPPED. =127 PRE-OPERATIONAL
2	R3681	RPDO Status	每一位代表一个 RPDO 的接收状态。 当为 1 时代表有正常更新。 Bit #0 表 RPDO1. Bit #9 表 RPDO10
3	R3682	Heart beat status	Node 1 ~ 15 Bit #1 为 1 时表有侦测到 Node #1 的 heartbeat. 侦测以 ConsumerHeartbeatTime 为周期。 当 ProducerHeartbeatTime 设为零时不产生也不侦测 heartbeat.
4	R3683		Node 16-31
5	R3684		Node 32-47
6	R3685		Node 48-63
7	R3686		Node 64-79
8	R3687		Node 80-95
9	R3688		Node 96-101
10	R3689		Node 102-127
11	R3690	Time Stamp	Second (0-59)
12	R3691		Minute (0-59)
13	R3692		Hour (0-23)
14	R3693		Day (1-31)
15	R3694		Month (1-12)
16	R3695		Year (2000-2099)
17	R3696	Time Stamp receiving indication	每收到新的 Time Stamp,此值都会加 1,直到 65535 后会从 0 重新累计.
18	R3697	Ladder software version	设定十进制值,CBCAN 会自动转换成 ASCII 字符到 Index 4000H sub-Index 02H 的 Ladder software version. 十进制值 0215 => ASCII 字符'0' '2' '1' '5'

5. LED 状态指示灯

如果 " Single status indicator " (参阅錯誤! 找不到參照來源。)选项没有被开启, 则 RUN LED 和 ERR LED 会按照下面两表运作:

RUN LED (绿色) -

表 5: RUN 灯模式

Indicator State	Node Operation State
Single flash	STOPPED
On	OPERATIONAL
Blinking	PRE-OPERATIONAL

ERR LED (红色) -

表 6: ERR 灯模式

Indicator State	Error State
Off	No error
Single flash	Warning limit reach
Triple flash	Rcv. SYNC signal is time-out
Quadruple flash	Any one of expected RXPDOs is time-out
On	Bus off

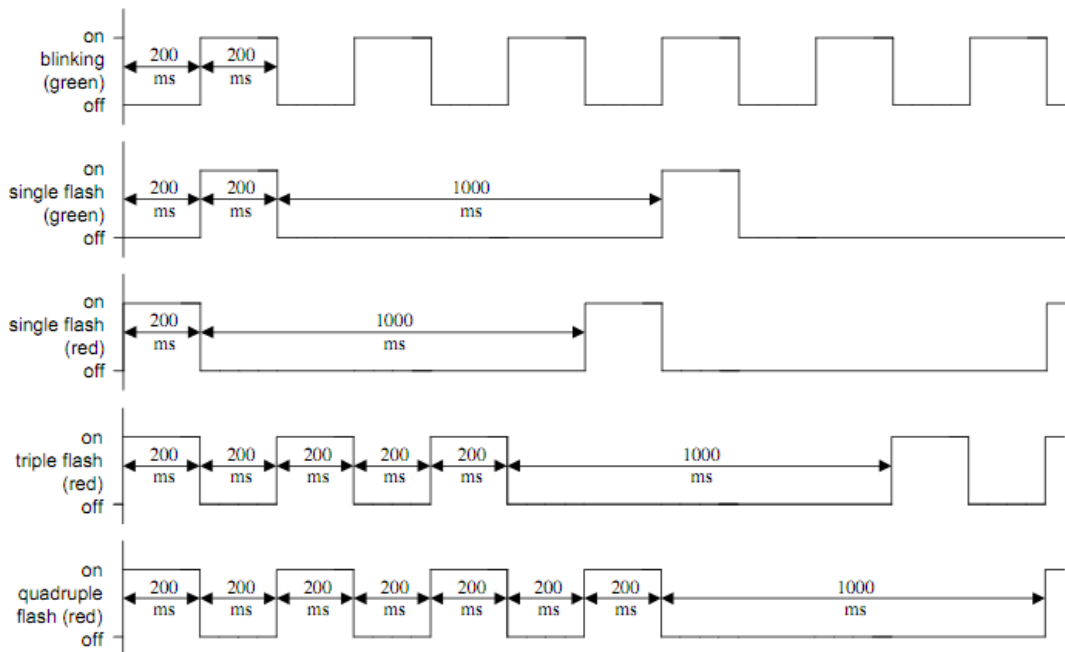


图 5: 灯号闪烁时序

当 " Single status indicator " 被开启, 则 RUN LED 和 ERR LED 会按照下列方式运作:

RUN LED-

表 7: RUN 灯模式 - Single status indicator

Indicator State	Node State
On	Normal network operation(OPERATIONAL)
Blink (period = 1Sec.)	Warning limit reach
Blink (period = 5Sec.)	Bus OFF
Off	Not OPERATIONAL state

ERR LED- 恒暗

6. EasyCANOpener 软件操作

执行 EasyCANOpener.exe 后出现以下画面

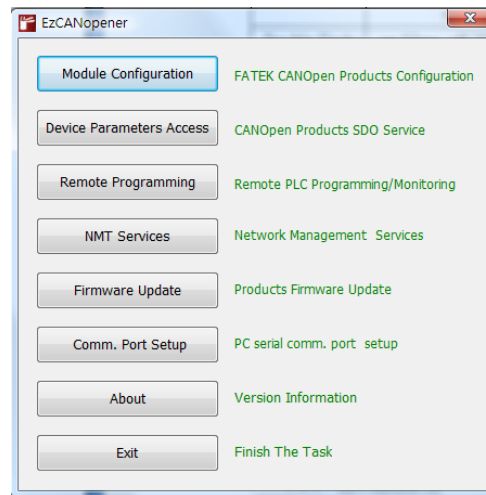


图 6: EasyCANOpener 主画面

本软件主要具备以下功能

- CBCAN 模板组态的建立,修改与设定.组态内容可以存成档案以方便复制与储存或修改.
- SDO 服务.可以直接操作对单一控制器的对象作读取或写入的动作或以批次的方式进行连续多笔的存取.此批次内容也可以存成档案以方便下次重复使用.
- PLC 远程访问桥接功能.透过此功能 PLC 规画软件 Winproladder 可以利用串行端口与中介 PLC 连接再透过中介 PLC 上的 CBCAN 模板与网络上另一端的 PLC 进行远程的存取.
- 网络管理(NMT)操作. 单一操作可对网络上的单一节点或所有节点发出
- Enter OPERATIONAL, Enter PRE-OPERATIONAL, Reset Node,
- Reset Communication, Stop 等命令.
- CBCAN 韧体更新. 可以利用串行埠对连接 PLC 上的 CBCAN 模板进行韧体更新.

6.1 PLC 连接

本软件操作时除非是建立或修改组态档,SDO 批处理文件等脱机操作外必须透过串行埠与 PLC 连接才能进行联机操作.当通讯线连接好后可透过按下

“Comm. Port Setup”功能键对 PC 端连接串行埠进行设定. 操作画面如下

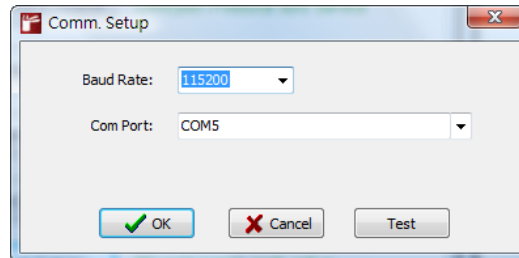


图 7: 通信设置

设好通讯参数时可按“Test”钮进行联机测试.

6.2 组态设定

在最上层点选“Module Configuration” 钮可以进入组态设定操作画面.如下图所示.

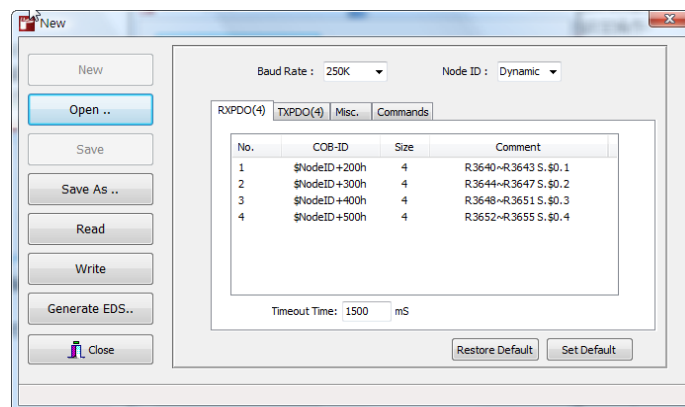


图 8: 组态设定

6.2.1 建立新组态

当进入组态设定操作画面时会自动产生一个新的组态.此时的组态内容为内定值.

6.2.1.1 基本设定

Baud Rate: 为网络的运作速度.内定值为 250K bps.设定值可从 10K~1M bps.每一个网络节点装置必须设相同的值否则无法正常运作.

Node ID: 为对应节点(Node)的站号.设定范围为 1~127 或动态(Dynamic).当设为动态时表示对应站号与 CBCAN 模板安装在一起的 PLC 站号相同.设定为动态的好处是不同的 CBCAN 可设相同

的设定值而不会有冲突.每一节点的站号须为唯一否则会运作不正常.此项设定内定值为动态.

6.2.1.2 RPDO 设定

点选 RPDO 页签即可进行接收程控对象(Receive Process Data Object)的设定.画面如下

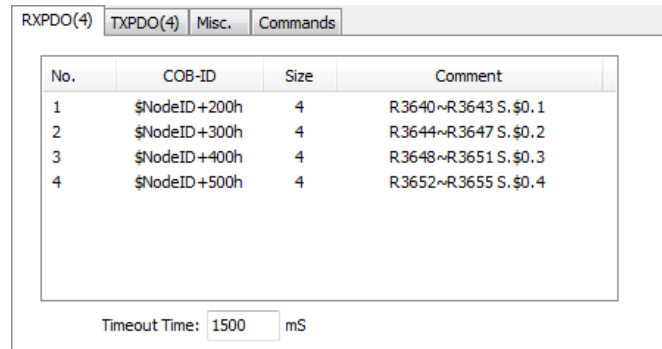


图 9: RPDO 设定

页签上括号内的数字表目前已定义的 PDO 数量.上图表示有 4 个 RXPDO 已定义.

RPDO 系利用突出式选单来进行操作.选单如下

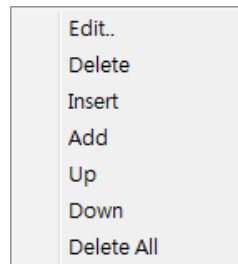


图 10: RPDO 突出式选单

兹分述各项的操作于后 -

- 编辑: 点选欲编辑的 PDO 后按鼠标右键叫出选单点选(Edit..)或双击欲编辑的 PDO 项后出现下面的画面.

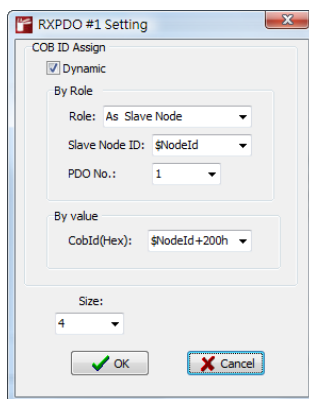


图 11: 编辑 RXPDO

此画面主要是作对应 RXPDO 的通讯对象标识符(COB-ID)及数据长度设定.画面内容说明于下:

- **Dynamic** 选项: 通讯对象标识符可设为固定值或与站号相关.此选项只有当前面的节点站号(Node ID)设为 **Dynamic** 才有作用.

COB-ID 的设定可以用两种观点来设定-

- **节点角色(By Role):** 基本上 CANOpen 针对主站及各节点的从站 PDO 有一默认的通讯对象标识符分配.每个节点均分配到 4 个 TXPDO COB-ID 及 4 个 RXPDO COB-ID.此种设定法即是从此角度来设定通讯对象标识符.
- **直接设值(By Value):** 输入 181H~5FFH 的数值.
- **数据长度(Size)设定:** 对应 RXPDO 数据长度可设 1~4 个 16 位字组(Word).

- **删除:** 点选欲删除的 PDO 后按鼠标右键叫出选单点选"Delete"即可将其删除.
- **插入:** 点选欲插入位置的 PDO 后按鼠标右键叫出选单点选"Insert"即可叫出一 PDO 编辑画面.输入设定值后即会在元位置插入一新的 PDO 定义.
- **新增:** 按鼠标右键叫出选单点选"Add"即可叫出一 PDO 编辑画面.输入设定值后即会在最后面加入一笔新的 PDO 定义.
- **上移:** 点选欲上移的 PDO 后按鼠标右键叫出选单点选"Up"即可将选取的 PDO 往上移动一个位置.
- **下移:** 点选欲下移的 PDO 后按鼠标右键叫出选单点选"Down"即可将选取的 PDO 往下移动一个位置.
- **全部删除:** 按鼠标右键叫出选单点选"Delete All"即可将全部 PDO 定义删除.

最后在页面下方还有一单独字段如下

Timeout Time: 配合 RPDO 的接收状态侦测,在模板状态区有一缓存器 RPDO Status 记录各 RPDO 的接收状态.此时间代表 RPDO 信号的最长接收间隔时间.当 RPDO 有收到信号时其对应状态位被设为 1.此值将一直维持直至超出此时间时若仍未再次收到对应 RPDO 的信号才清除为 0.内定值为 1.5 秒.

6.2.1.3 TPDO 设定

TPDO 的设定类似 RPDO.除以下说明为不同处其余操作请参考**錯誤! 找不到參照來源。**节说明.

在 TXPDO 编辑画页中增加了一个 Transmission type 的设定字段

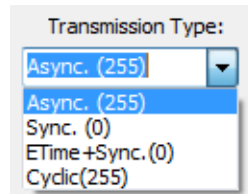


图 12: 传输模式

Transmission Type: 代表对应 TPDO 的传输模式.

- **Async** 表异步传输.传送的时机为状态变化时或已超出一固定时间(Event Time)都没发生变化.当上一状态变化已造成数据传送时,须等待禁止传送时间(Inhibit Time)后才能继续侦测状态变化.利用此功能可避免异常频繁变化信号占据太大的通讯带宽.
- **Sync.** 表同步传输.其传送时机和异步一样.但不同的是会再延后一直等到下一个同步信号(SYNC)收到时才会开始传送.同步传送须配合对方的同步接收才能发挥功用.
- **Cyclic:** 表固定传输速度.其传送间隔时间由 Cyclic Time 的设定值决定.

No.	COB-ID	Size	Comment
1	\$NodeID+180h	4	R.3600~R.3603 S.\$0.1
2	\$NodeID+280h	4	R.3604~R.3607 S.\$0.2
3	\$NodeID+380h	4	R.3608~R.3611 S.\$0.3
4	\$NodeID+480h	4	R.3612~R.3615 S.\$0.4

Inhibit Time: 100 mS Event Time: 1000 mS Cyclic Time: 2000 mS

图 13: TXPDO 设定页面

Inhibit Time: 最小传送间隔时间.当对应信号传送后须至少延迟此时间才能发出下一笔信号.默认值为 100 毫秒

Event Time: 最大传送间隔时间.当对应信号传送后,下笔信号的间隔时间不大于此时间. 默认值为 1 秒

Cycle Time: Cyclic 模式下的传送间隔时间. 默认值为 2 秒

6.2.1.4 杂项设定

点选“Misc.”页签即可进行杂项设定.画面如下

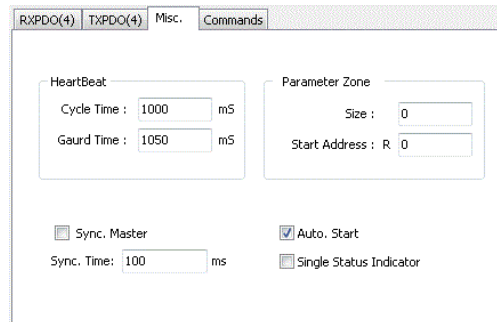


图 14: 杂项设定页面

Heartbeat(脉搏信号): 用来代表节点存活的固定间隔时间信号. 正常状况时节点每隔一固定间隔时间即会发出一笔 Heartbeat 信号.其它节点依据此信号是否及时出现来判定它节点是否有在运作.

- **Cycle Time:** 代表本节点的固定 Heartbeat 发送间隔时间.若此值设为零时表示不使用 heartbeat 功能.
- **Guard Time:** 代表本节点用来侦测它节点 heartbeat 是否正常的时间.理论来说每个节点可设不同的 cycle time.因此 cycle time 应针对不同节点进行个别设定.但为避免设定过于繁复因此本系统假设每个节点的 cycle time 是相同的.设定时此值应比 cycle time 设定值大.

参数变数区: 通常应用时会有一些不常变化的系统参数.此些参数可以映像至 CNAOpen 的对象然后透过 SDO(Service Data Object)服务进行网络远程访问.

- **Size:** 表示映像的缓存器数量.最大值为 999.
- **Start Address:** 表示被映像的缓存器起始地址.只有 R 缓存器才可以被映像为参数缓存器.

Sync. Master 选项: 当勾选时表本节点为同步主节点负责同步信号的发送.当此选项没有勾选时且任一 TXPDO 运作在 Sync 模式,则 CBCAN 会侦测网络上的同步主节点的同步讯号.若是其同步讯号没有在 timeout 的设定值内回传,则系统会发出同步讯号 timeout 的事件 (R3680 Bit1).

Sync. Time: 当 Sync. Master 被勾选时,此值代表同步信号的间隔发送时间.当 Sync. Master 没有被勾选时,则此值代表同步讯号

timeout 的时间.一般来说,timeout 时间必须比同步主节点设定的同步间隔时间还要长,才能确保同步讯号侦测机制运作正常.

Auto Start 选项: 选取"Auto Start"功能后,CBCAN 模板会在上电开机后直接由 PRE-OPERATIONAL 状态自动进入到 OPERATIONAL 状态.

Single Status Indicator: CBCAN 模板提供了两种 LED 显示方式,当选取 Single Status Indicator,CBCAN 模板只会使用 RUN LED 来展示网络状态,其详细的灯号显示请参阅第**錯誤! 找不到參照來源。**章内容.


6.2.1.5 字符串命令(Commands)

必要时可利用此窗口,输入字符串命令给 CBCAN 模板以达到额外的组态设定效果.

6.2.2 储存组态至档案

点选"Save .."或"Save As" 按钮即出现组态文件储存窗口.设定好文件名后按"存盘" 按钮即完成存入动作.

6.2.3 开启已建立的组态档

点选"Open .."按钮即出现组态文件开启窗口.点  取按钮选取先前储存的档案. 按"OK" 按钮即完成读取.

6.2.4 读取 CBCAN 模板的组态内容

点选"Read"按钮即出现通讯参数设定窗口.如下所示

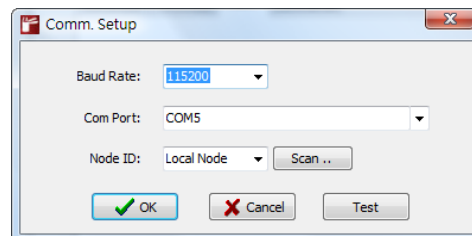


图 15: 读取组态的通信设置

其中 Baudrate 及 Comm. Port 为 PC 的通讯端口参数.

Node ID: 此表欲读取的 CBCAN 节点站号(1~127).若设为 Local Node 时表读取与 PC 连接的 PLC 上的 CBCAN 节点内容.

SCAN 按钮: 按此键可利用扫描的方式了解网络上有哪些节点上线.其操作画面如下:

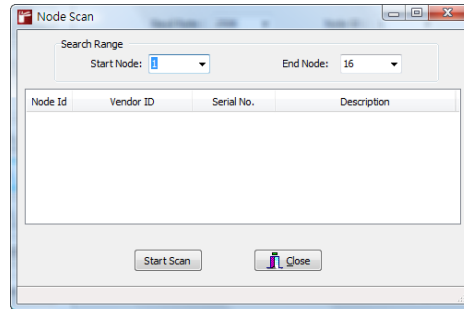


图 16: 节点扫描

- Search Range: 利用此设定可缩小扫描范围以节省扫描时间.
- Start Node: 扫描从此节点站号开始
- End Node: 扫描至此节点站号后结束

按“Start Scan”按钮后开始扫描. 扫描进行时可按“Stop Scan”按钮以提前结束扫描.扫描到的模板信息会出现在中间的窗口.如下所示

Node Id	Vendor ID	Serial No.	Description
1	02EF (Fatek)	0	Fatek CBCAN
2	02EF (Fatek)	0	Fatek CBCAN

图 17: 搜寻结果窗口

由 Vendor ID, CBCAN Serial number 提供的信息可用来辅助模板的选取.

执行读取: 完成以上相关栏设定后按“OK”键即完成读入.

6.2.5 将组态资料写入 CBCAN 模板

与**錯誤! 找不到参照来源**。节的操作类似不同的仅是点选“Write”按钮.详细操作请参阅**錯誤! 找不到参照来源**。节的说明.

6.2.6 产生电子数据文件

点取“Generate EDS..”即出现产生电子数据文件(Generate EDS File)窗口.

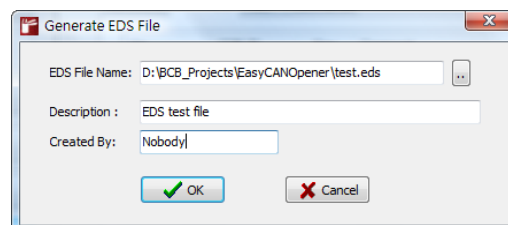


图 18: 产生 EDS 档案

Description: EDS 檔的附加描述.

Create By: 建立此电子文件的作者名.

以上二输入字符串会出现在电子档案中.

按“OK”钮完成电子文件建立工作.

6.2.7 结束组态设定

按“Close”钮离开组态设定相关功能

6.3 SDO 数据存取操作

在最上层点选“Device Parameter Access” 钮可以进入 SDO 数据存取操作操作画面.如下图所示.共通的设定字段说明于下

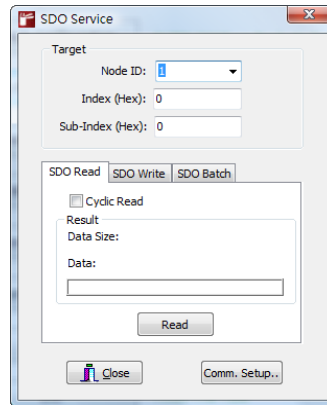


图 19: SDO 服务页面

Node ID: 表欲存取的节点站号

Index: 对象主索引号

Sub-Index: 对象次索引号

6.3.1 SDO 数据读取

点选 SDO Read 页签即可进入 SDO 对象读取服务画面.如上所示.

Cyclic Read 选项: 勾选时代表重复执行读取动作.勾选此栏适于观察动态对象的变化.

Read 按钮: 点选后开始执行读取动作.读取的结果会出现在中间的窗口.如下所示.

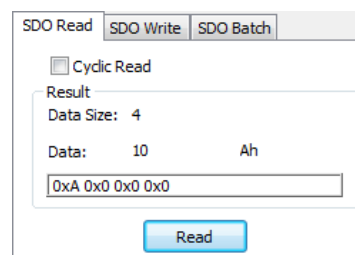


图 20: SDO Read 页面

上面内容表示读取的对象大小为 4 个 byte.内容是 10(十进制) AH(十六进制)或字符串 0xA 0x0 0x0 0x0.

6.3.2 SDO 资料写入

点选 SDO Write 页签即可进入 SDO 对象写入服务画面.如下所示

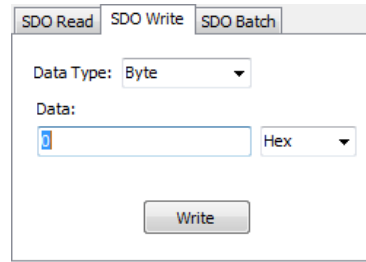


图 21: SDO Write 页面

Data Type: 表对象的数据种类.目前仅提供 Byte, Word, Dword 等三种类型数据.

Data: 欲写入的值.配合后方的 Hex 或 Decimal 格式选取其输入可为十六进数或十进数.

Write 按钮: 点选后开始执行写入动作. 执行若有错误时会出现错误窗口.

6.3.3 SDO 资料批处理

对于重复又固定性的 SDO 写入动作可利用此功能来执行. 首先点选 SDO Batch 页签进入下面画面.

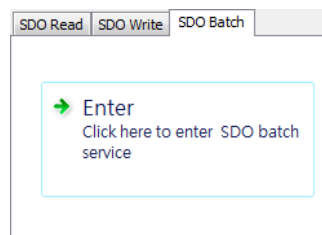


图 22: SDO 批处理

此时须再单击上图的 Enter 键后才会真正进入 SDO 对象批次服务画面.如下所示

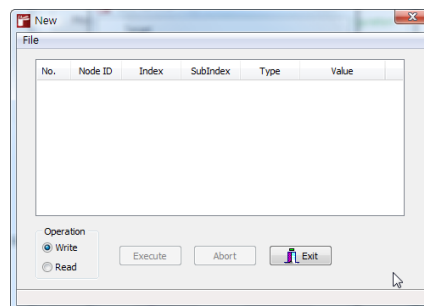


图 23: SDO 批处理设定页面

6.3.3.1 建立批次内容

进入 SDO 对象批次服务画面时批次执行内容是空的.此时可自行建立执行内容.操作方法采突出示的选单操作.可执行的选单如下所示.

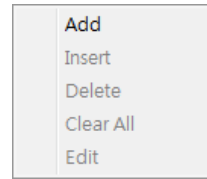


图 24: 批处理突出式选单

增加: 在批次内容窗口内按鼠标右键点选“Add”选单项即会出现以下编辑窗口

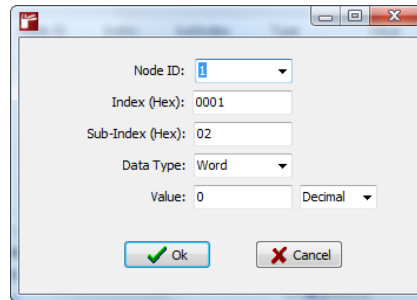


图 25: 增加 SDO 命令

- **Node ID:** 欲写入的物件节点站号
- **Index:** 对象的主索引值
- **Sub-Index:** 对象的次索引值
- **Data Type:** 对象的数据型态.分 Byte, Word, Dword 三种型态
- **Vaule:** 写入值

按“OK”钮完成建立一笔新批次内容的操作

插入: 在欲插入点上按鼠标右键点选“Insert”选单项即可进行插入内容的设定. 按“OK”钮即会在插入点新增一笔批次命令.

删除: 在欲删除的批次项上按鼠标右键点选“Delete”选单项即可将对应项内容删除.

删除全部: 在批次内容窗口内按鼠标右键点选“Delete All”选单项即会将全部批次命令内容清除.

编辑: 在欲进行编辑的批次项上按鼠标右键点选“Edit”选单项或双击该批次项即可对该项进行编辑.按“OK”钮完成内容的修改.

6.3.3.2 批次内容存档

点选窗口上方的主选单“FILE”再点选“Save”或“Save As”选单即可将目前的批次内容存入档案.

6.3.3.3 读此批处理文件

点选窗口上方的主选单“FILE”再点选“Open”选单即可将批处理文件的内容读入。

6.3.3.4 执行批次写入

先点选 operation 群组内的“Write”选项后按“Execute”钮即会开始进行批次写入动作.写入的动作是依据目前在批次窗口内的内容来进行.

6.3.3.5 执行批次读取

先点选 operation 群组内的“Read”选项后按“Execute”钮即会开始进行批次读取动作.读取的动作是依据目前在批次窗口内的内容来进行.与写入不同的是读取的值会覆盖掉原来的批次内容值.

6.4 PLC 远程监控服务

透过此服务,永宏 PLC 阶梯程序规划软件 Winproladder 可以透过 PC 的串行埠与带有 CBCAN 模板的 PLC 联机.藉由 CBCAN 连上网络与远程带有 CBCAN 模板的 PLC 进行阶梯程序的修改与监视.

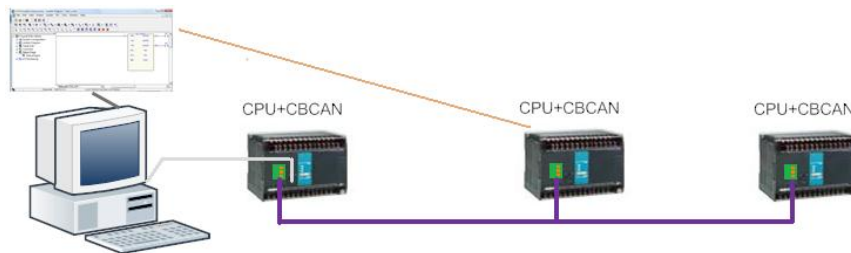


图 26: PLC 远程监控服务

6.4.1 开启服务

在最上层点选“Remote Programming” 钮可以进入远程监控服务操作画面.如下图所示

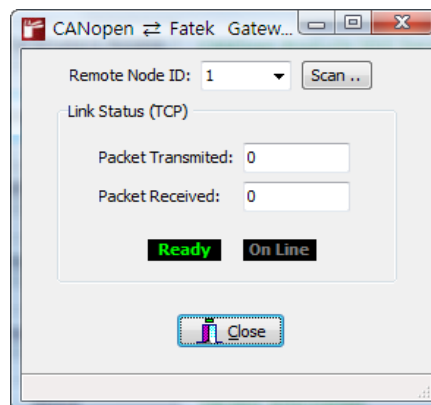


图 27: Gateway 状态窗口

字段说明如下

Remote Node ID: 输入欲联机的节点站号.

Link Status: 輔助狀態

Packet Transmitted: 傳回給應用程式的數據封包數

Packet Received: 接收到的應用程式數據封包數

本服務是採所謂的閘信道(Gate Way)運作方式.此服務對外的服務窗口是採 TCP 網絡,服務端口號為 500

Ready: 已准备好接受服務時顯示為綠色

On Line: TCP 網絡已與應用程式鏈接時顯示為綠色字.

6.4.2 使用服務

要使用本服務時須透過 TCP 联机至本机(IP: 127.0.0.1)的 500 埠号.因此 Winproladder 的設定例如下.

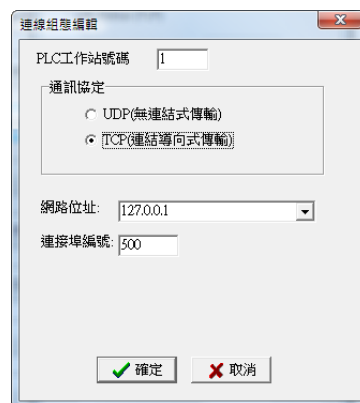


圖 28: WinProladder 联机設定頁面

由於透過 TCP 網絡中介,應用時甚至可由遠程透過以太網上 CANopen 網絡.

6.5 NMT(網絡管理)操作

在最上層點選“NMT Service” 鈕可以進入網絡管理操作畫面,如下圖所示

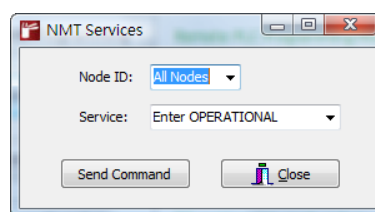


圖 29: NMT 服務

Node ID: 輸入欲管理的節點站號.可以單獨選擇一站(1~127)或選所有站 (All Nodes)

Service: 服務選擇.有以下項目

Start Remote, Enter PRE_OPERATIONAL, Reset Node

Reset Communication, Stop

設定好後按“Send Command”開始執行.

6.6 CBCAN 模板固件更新操作

在最上层点选“Firmware Update” 钮可以进入固件更新操作画面,如下图所示.

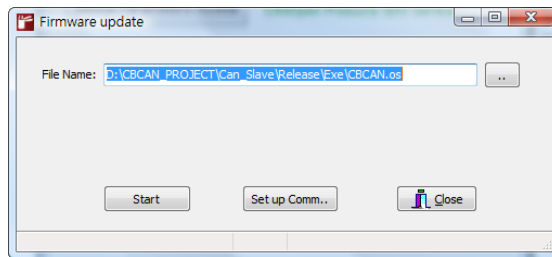



图 30: 固件更新

File Name: 欲更新的固件档案

6.6.1 选取固件档案

点取  打开档案选择窗口确定后按“OK”出现以下画面

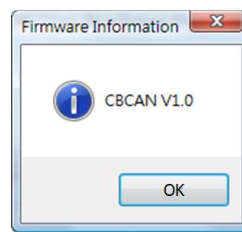


图 31: 固件信息

画面中会显示对应档案内的固件版本信息.

6.6.2 开始固件更新

点取“Start” 钮开始固件更新.

6.7 结束操作

在最上层点选“Exit” 钮结束 EasyCANOpener 操作

7. CBCAN SDO 和 NMT 服务的 PLC 缓存器界面

目前的 Winproladder 设计并没有提供 SDO 和 NMT 服务的指令，因此在某些特定应用必需使用 SDO 或 NMT 就必须根据此章节提供的 PLC 缓存器界面来设计 Ladder 程序。

7.1 SDO 服务

当 CBCAN 模块在 FBs PLC 使用时，PLC 会经由缓存器 R3700~R3769 来控制 CBCAN 模块。完成一个 SDO 服务的主要信息包含有 SDO 从站的 node ID (R3702)、从站对象字典的 index 及 sub-index (R3703 and R3704)、要存取的数据大小(R3705)以及数据本身(R3706~R3769)；在上述数据都准备妥当且写到相对应的缓存器了之后，就可以将“CA0Eh”写入到缓存器 R3700 来通知 CBCAN 模块执行 SDO 服务。

而执行结果会在指令执行结束后显示在同样范围的缓存器 R3700~R3769，请参阅下方的表格来进一步了解 SDO 服务的详细使用方法。

表 8: PLC Registers for SDO Service

序号	缓存器	功能
1	R3700	Execute Flag = CA0EH
2	R3701	Command. SDO Read=5055 SDO Write = 5066
3	R3702	Node ID(station number, 0~127. If =0, this node)
4	R3703	Object Index (0~65535)
5	R3704	Object sub-index (0~255)
6	R3705	Object data size: 1 ~ 128 Byte. Data size to Write or max. data buffer size for read
7	R3706	Data word #0
8	R3707	Data word #1
.	.	.
70	R3769	Data word #63

表 9: Registers Status after SDO Service Execution

序号	缓存器	功能	
		Read	Write
1	R3700	Result Code* ₁	
2	R3701	No Change	
3	R3702		

4	R3703		
5	R3704		
6	R3705	Actual read data size	No Change
7	R3706	Data word #0* ₂	No Change* ₂
8	R3707	Data word #1* ₃	No Change* ₃
.	.	.	No Change
70	R3769	Data word #63	No Change

*₁: Result Code: OK =0, CMD_CODE_ERR = 1, NODE_ID_ERR = 2, SDO_EXEC_ERR = 4.

*₂: 当执行失败时的 SDO error code low word (Result Code= 4)

*₃: 当执行失败时的 SDO error code high word (Result Code= 4)

表 10: SDO Error Code

Error Code Name	Error Value	Description
ABORT_TIME_OUT	0x05040000L	SDO service Time out
ABORT_NO_OBJ	0x06020000L	No such object
ABORT_RO	0x06010002L	Attempt to write a read-only object
ABORT_SYS_LENGTH	0x06040047L	Data length exceed system allow
ABORT_NO_SEGEMNT	0x06010000L	Not support segment transfer
ABORT_OBJ_LENGTH	0x06070010L	Not match object length
ABORT_SYNC	0x05040001L	Command specifier not valid
ABORT_TOGGLE_BIT	0x05030000L	Toggle bit not alternated
ABORT_PARM_LENGTH	0x06070012L	Length of service parameter too high
ABORT_WO	0x06010001L	Attempt to read a write-only object
ABORT_READ_LENGTH	0x05040005L	Object length too big to read

7.2 NMT 服务

NMT 服务同样的也是使用和 SDO 服务一样的方式来完成，不同点在于使用 NMT 服务时是对特定的对象字典地址做存取“0x1F82” (write service)以及“0x300C” (read service)。

NMT Management 服务 – 发送 NMT 服务指令到 NMT 从站

SDO write (R3701): With Object index (R3703) = 0x1f82

R3704: Sub-Index n	Description
n = 1 ~ 127	Node with node ID n to be requested
n = 128	Request all nodes

R3702: Do not care.

R3706: Command Code	NMT Service Description
5 or 1	Start Remote Node
4 or 2	Stop Remote Node
127 or 128	Enter Pre-Operational State
6 or 129	Reset Node
7 or 130	Reset Communication

从站 **NMT** 状态服务 – 读取特定从站的 **NMT** 状态

SDO read (R3701): With Object index (R3703) = 0x300c

Sub-index (R3704) = 0

Node ID (R3702): Requested node ID

Execution Result (R3700 and R3706):

寄存器	值	描述
R3700	0	OK
	Not 0	Not OK
R3706 (Only valid when R3700 is 0)	4	Stopped
	127	Pre-operational
	5	Operational
	6	Not detected

7.3 范例 – NMT 服务 (Start Remote)

下方阶梯程序提供了如何使用 **NMT** 服务来致能网络上的远程节点的范例；根据章节 7.2 的描述，将各寄存器设定如下表所示便可送出 **NMT** 的命令。

表 11: 寄存器设定值 - NMT 服务:Start Remote

寄存器	值	描述
-----	---	----

R3701	5066	Write command
R3703	1F82H	NMT service index
R3704	128	Request all nodes
R3706	5	Start Remote Node

在完成上述的缓存器后,最后需将 R3700 填入 CA0Eh 以触发 NMT 命令。
下方范例的第一个 function “17.CMP” 只是来作为 NMT 服务的触发,使用 NMT 服务并不是必要步骤。

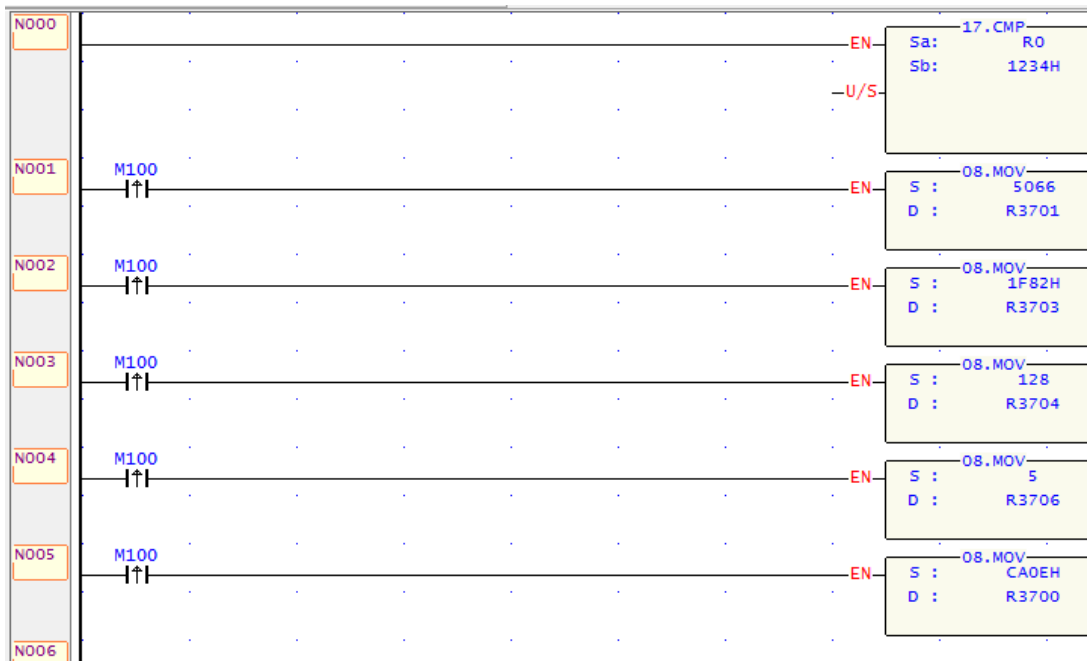


图 32: NMT Start Remote command - 阶梯程序范例

附录一

1.1 应用手册

CBCAN 是一个支持 CANopen 协议的通讯模块，它可安装在 FBs 系列 PLC 便可使 PLC 与 CANopen 装置通讯。CBCAN 里大部份的对象都可以透过 EzCANopener 软件来设定，而且 CBCAN 的对象必需要根据应用网络需求妥善设定才可成功与外界通讯。这个章节提供了两个简单的应用，让用户可以快速了解对象设定及上手使用 CBCAN 模块。

1.1.1 范例一：将 CBCAN 连接至一个 CANopen 主站装置

如下图所示，这个范例将安装了 CBCAN 的 FBs PLC 当做 CANopen 的从站装置连接到一个 CANopen 主站装置。

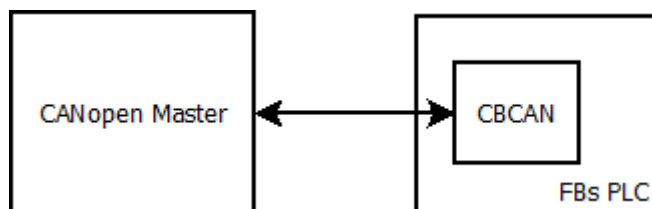


图 33: 范例网络

此范例中，主站装置使用了下列的 CANopen 网络参数。

表 12: 范例 - 网络设定

Items	Values
Node ID	2
Baudrate	125K bps
Sync. master	Disabled
Error control	Heartbeat producer (1000 ms) Heartbeat consumer (1050 ms)
Number of TxPDO	1 (3 words) Transmission type: Async. Mode (255)
Number of RxPDO	1 (3 words)

在 CBCAN 的对象参数里，都可以透过 EasyCANopener 软件来设定，通常 node id 和 baud rate 是第一个需要设定好的参数。要开始设定参数，首先，點選主页面的 Module Configuration 按钮开始 CBCAN 的 CANopen 参数设定画面。

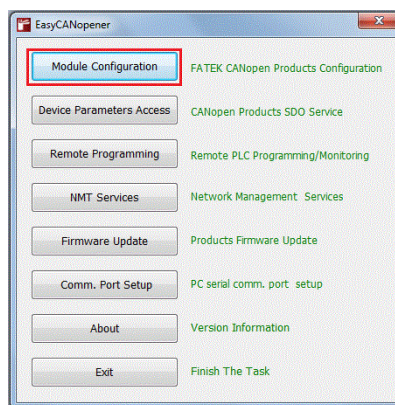


图 34: 主页面

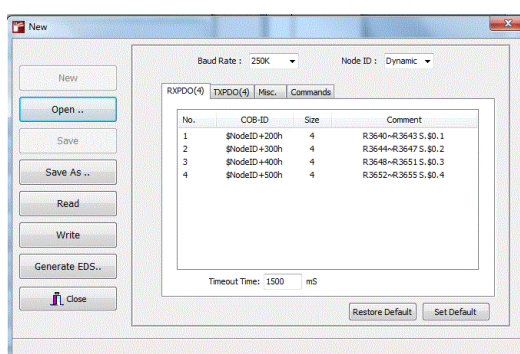


图 35: 模块设定画面

因为主站 node id 是 2，所以所有 1~127 的 id 里，除了 2 之外的 node id 都可指定给 CBCAN 模块，假设此范例 FBs PLC 所使用的 node id 是 1，那 CBCAN 便可直接指派使用 FBs PLC 的 node id。若要使用 FBs PLC 的 node id 只要将” Node ID” 的参数设定成” Dynamic”。



图 36: 节点 ID 设定

在” Node ID” 设定旁便是” Baud Rate” 的设定，根据主站需求将 CBCAN 的波特率设定成 1M bps.



图 37: 鲍率设定

设定 RxPDO:

此应用中，主站装置使用异步 Async. 255 的传输模式，此传输模式中 PDO 封包只会在事件产生时才传送，所以两个连续封包间的时间是无法预测

的，因此” Timeout Time” 在这个范例中并不重要且可忽略。主站装置在此应用中只和 CBCAN 交换 6 个 byte 的数据，所以只需要一个 RxPDO，RxPDO2~RxPDO4 便可移除掉不需使用，要移除它们只要圈选它们，然后再按鼠标右键呼叫操作选单，再点选” Delete” 选项即可。

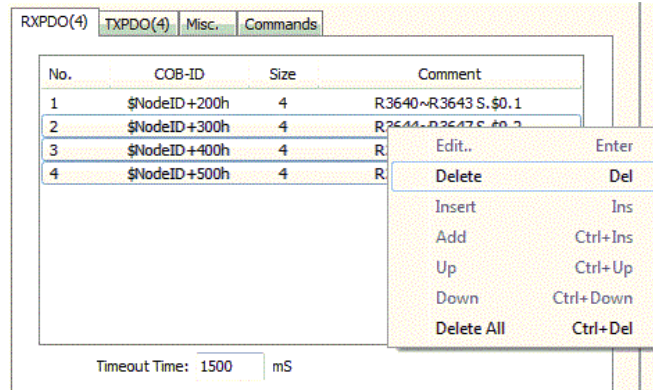


图 38: 删除 RXPDO

删除之后，画面上应该只剩下 RxPDO1，只要用鼠标左键双击 RxPDO1 就可进行进一步的 RxPDO 参数设定。

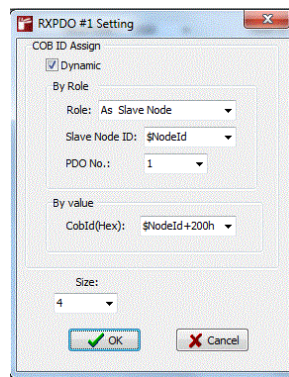


图 39: RXPDO 设定页面

在” RXPDO #1 Setting” 的画面里，因为 CBCAN 已被设定使用 FBs PLC 主机的 node id，所以在这里可以保留” Dynamic” 选项勾选的状态使 CBCAN 使用 FBs PLC 的 node id 当做 slave node id。此范例 CBCAN 是从站的角色，选择参数” As Slave Node” 来设定” Role” 即可，而” Slave Node ID” 参数可选择” \$NodeId ” 使 CBCAN 使用 node id 来当做 CANopen 从站的 node id。通常 CANopen 主站和从站都会使用” slave node id + 200h” 来当做从站 RxPDO1 的 COB-ID，因此若没有特别的需求，” CobId(Hex) ” 设定应该选择 ” \$NodeId+200h ”。CBCAN 的设计是使用内部缓存器来当做计算 PDO 数据长度的基本计算单位，所以当 PDO 长度设定成 4 时，其实际数据长度是 8 个 bytes。在此应用，CBCAN 和主站间只有传输 6 个 bytes 的数据，因此其数据长度 ” Size ” 应该设定成 3。

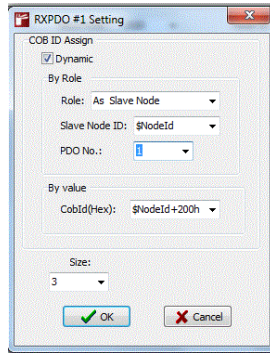


图 40: 此应用的 RXPDO1 设定

设定 TxPDO:

根据此范例的网络需求，CANopen 主站只需要一个 TxPDO，因此 TxPDO 2~4 也可以移除不使用。

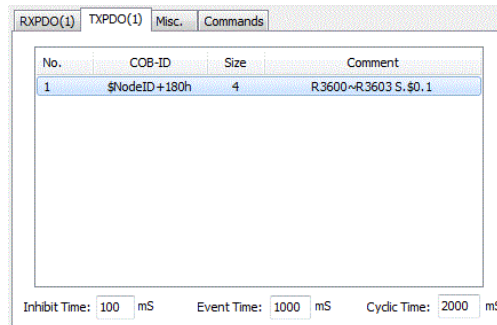


图 41: 移除 TXPDO

就有如 RxPDO 设定，鼠标左键双击 TxPDO1 来进行进一步 PDO 的设定。

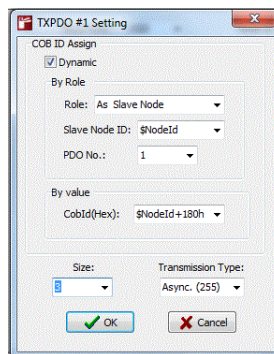


图 42: TxPDO 设定

和 RxPDO 的设定一样，此范例里 " Role " 、" Slave Node ID " 、" PDO No " 及 " CobId(Hex) " 皆可直接使用出厂默认值，其设定可以参考上图。根据此应用所需，分别将 " Size " 及 " Transmission Type " 设定成 3 及 " Async.(255) " 使 CBCAN 使用异步传输来与主站交换 6 个 bytes 数据。除此之外，在 TxPDO 表列画面下方有三个与 TxPDO 传输有关的参数，他们

分别是 " Cyclic Time " 、 " Inhibit Time " 和 " Event Time " ；因为 " Cyclic Time " 只使用在 Cyclic 的传输模式，所以在此范例并不需要特别设定，而 " Inhibit Time " 和 " Event Time " 是和异步传输 Async. (255)有关的参数，是此范例是需要设定的。但是此范例并没有对此二参数有特别要求，所以可以直接使用默认值 1000ms 和 100ms，它们的意义如下，不管有没有事件产生，" Event Time " 1000ms 代表连续封包的间隔最长不可超过 1000ms，而 "Inhibit Time" 100ms 代表最短时间不可少于 100ms。

Inhibit Time: 100 mS Event Time: 1000 mS Cyclic Time: 2000 mS

图 43: TXPDO 传输参数设定

设定 Misc.:

在 Misc.设定页面中,有三个参数可设定,分别是 " HeartBeat " 、" Parameter Zone " 及 " Sync. Master " ； " Parameter Zone " 和 CANopen 通讯没有关系因此可以将其参数设定成 0 以关闭此功能,因为此范例使用 " Async.(255) " 传输,所以 " Sync. Master " 也可以不勾选将它关闭,除非网络需求此 CBCAN 从站需扮演 Sync. Master 的角色。

因为主站参数为每 1000ms 会送出一个 heartbeat 讯号,所以 CBCAN 必需要将 " Guard Time" 指定一个较大的参数值,譬如 1050ms; 同样的,身为从站的 CBCAN 模块必须将 " Cycle Time" 的参数值设定一个相较主站 Producer heartbeat time 参数较小的值。

下载组态到 CBCAN:

当上述设定都完成后,便可以将组态下载到 CBCAN 了; 首先,先点选主页面的 " Write " 按钮呼叫 CBCAN 的串行通信设置页面,在此页面,将 " Baud Rate " 参数设定成 Auto 让 EasyCANopener 软件自行侦测串行传输的速度设定,然后点选 OK 按钮来下载组态到 CBCAN 模块。

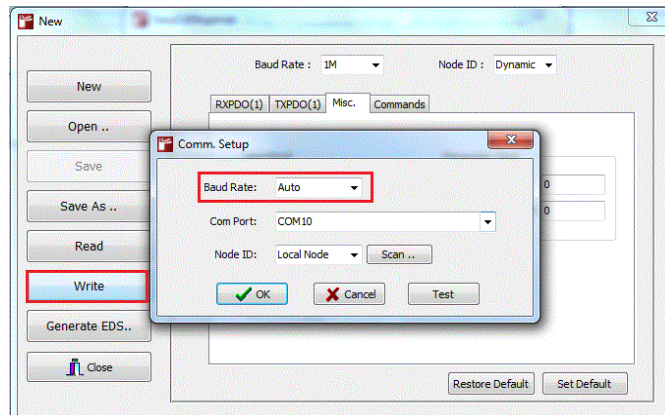


图 44: 将组态写入 CBCAN

输入及输出数据映像:

在下载新组态到 CBCAN 模块后, CBCAN 模块应已自行重置且和主站开始通讯; CBCAN 最多可支持 10 个 RxPDO 和 10 个 TxPDO, 而且每个 PDO 都可包含 8 个 byte 的数据, 这些 PDO 的数据都被映像到 FBs PLC 的内部缓存器, 当 RxPDO 封包被 CBCAN 模块接收时, CBCAN 首先会将所有 RxPDO 的数据整合, 然后再映像到内部缓存器区 R3640~R3679, 相同的, 当 CBCAN 模块要传送 TxPDO 时, 会将从内部缓存器区 R3600~R3639 将数据分别发送到个别的 TxPDO。

而此范例只有传输 6 个 byte 的数据, 因此 RxPDO1 的数据被映像到 R3640~R3642 而 TxPDO1 的资料将从 R3600~R3602 读出。

1.1.2 范例二: 将 CBCAN 连接至一个 CANopen 从站装置

如下图所示, 这个范例将安装了 CBCAN 的 FBs PLC 当做 CANopen 网络上的主站装置且连接到一个 CANopen 从站装置。

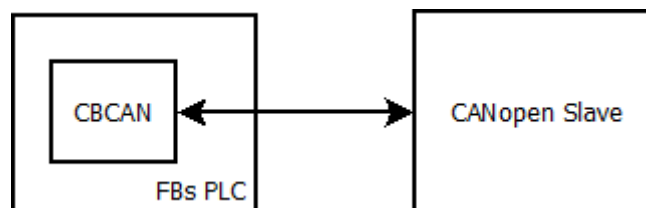


图 45: 范例网络

此范例中, 假设主站 CBCAN 模块使用了下列的 CANopen 网络参数来与从站装置通讯。

表 13: 范例 - 网络组态

items	Values
Node ID	1
Slave Node ID	2
Baudrate	125K bps
Sync. master	Disabled
Error control	Heartbeat producer (1000 ms) Heartbeat consumer (1050 ms)
Number of TxPDO	1 (3 Words) Transmission type: Async. Mode (255)
Number of RxPDO	1 (3 Words)

在 CBCAN 的对象参数里，都可以透过 EasyCANopener 软件来设定，通常 node id 和 baud rate 是第一个需要设定好的参数。要开始设定参数，首先，点选主页面的 Module Configuration 按钮开始 CBCAN 的 CANopen 参数设定画面。

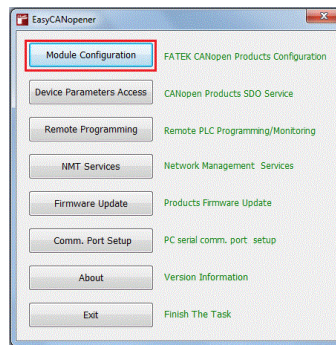


图 46: 主页面

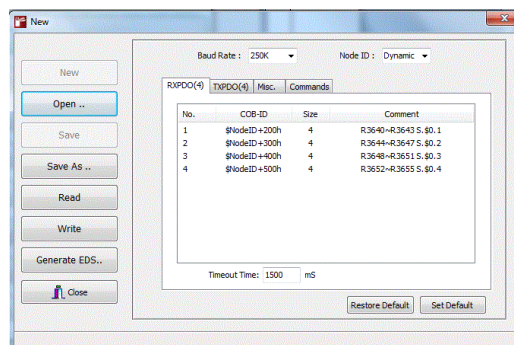


图 47: 模块组态页面

因为从站 node id 是 2，所以所有 1~127 的 id 里，除了 2 之外的 node id 都可指定给 CBCAN 模块，假设此范例 FBs PLC 所使用的 node id 是 1，那

CBCAN 便可直接指派使用 FBs PLC 的 node id。若要使用 FBs PLC 的 node id 只要将” Node ID” 的参数设定成” Dynamic” 。



图 48: 节点 ID 设定

在” Node ID” 设定旁便是” Baud Rate” 的设定，根据此范例网络需求将 CBCAN 的波特率设定成 1M bps.



图 49: 鲍率设定

设定 RxPDO:

此范例的网络中使用异步 Async. 255 的传输模式，在此传输模式中 PDO 封包只会在事件产生时才传送，所以两个连续封包间的时间是无法预测的，因此” Timeout Time” 在这个范例中并不重要且可忽略。此应用中 CBCAN 模块只和从站装置交换 6 个 byte 的数据，所以只需要一个 RxPDO，RxPDO2~RxPDO4 便可移除掉不需使用，要移除它们只要圈选它们，然后再按鼠标右键呼叫操作选单，再点选” Delete” 选项即可。

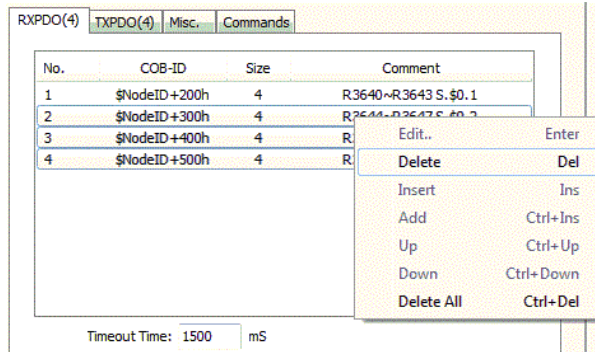


图 50: 移除 RxPDO

删除之后，画面上应该只剩下 RxPDO1，只要用鼠标左键双击 RxPDO1 就可进行进一步的 RxPDO 参数设定。

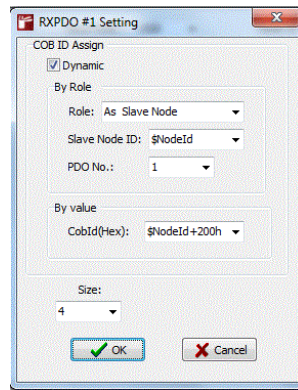


图 51: RXPDO 设定页面

因为从站装置不能使用和 CBCAN 模块一样的 node id, 所以不需要勾选 "Dynamic" 选项把 FBs PLC 主机的 id 当做从站的 node id。此范例 CBCAN 模块是一个主站模块, 选择 "As Master Node" 来设定 RxPDO 为主站模式; 范例中, 从站装置使用 node id 2, 在 "Slave Node ID" 设定中要设定成 2。一般来说, 主站装置会使用 "slave node id + 180h" 来当做 RxPDO1 的 COB-ID, 因为从站的 node id 为 2, 此 COB-ID 可设定成 "182h"。(请同时检查从站是否也是使用相同规则来设定 PDO)

CBCAN 的设计是使用内部缓存器来当做计算 PDO 数据长度的基本计算单位, 所以当 PDO 长度设定成 4 时, 其实际数据长度是 8 个 bytes。在此应用, CBCAN 和主站间只有传输 6 个 bytes 的数据, 因此其数据长度 "Size" 应该设定成 3。

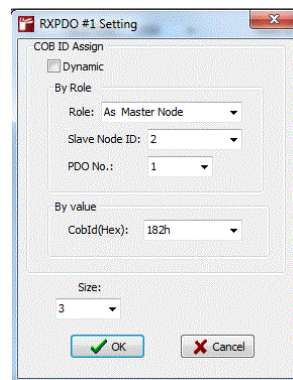


图 52: 此范例中的 RXPDO1 设定

设定 TxPDO:

根据此范例的网络需求, CBCAN 只需要一个 TxPDO, 因此 TxPDO 2~4 也可以移除不使用。

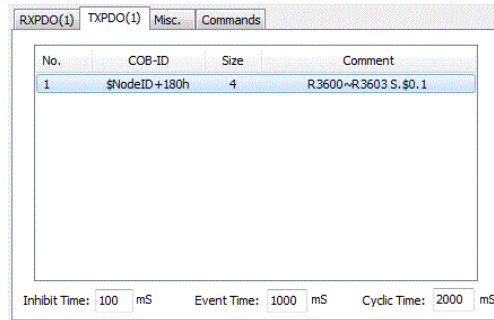


图 53: 移除 TXPDO

就有如 RxPDO 设定，鼠标左键双击 TxPDO1 来进行进一步 PDO 的设定。

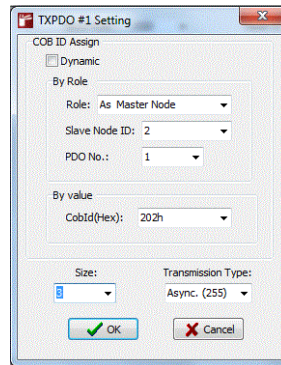


图 54: TXPDO 设定

和 RxPDO 的设定一样，" Role " 应该设定成 " As Master Mode "，此时 " CobId(Hex) " 应该会自动设定成 " \$NodeId+200h "。从站装置并不使用和 FBs PLC 主机的 id，所以 " Dynamic " 应该要关闭不勾选；从站的 node id 为 2，因此将 " Slave Node ID " 设定成 2，此时 " CobId(Hex) " 会自动变成 " 202h "，这个设定也符合一般规则使用 " Slave node ID+200h " 来当做主站 TxPDO 的 COB-ID 设定。根据此应用所需，分别将 " Size " 及 " Transmission Type " 设定成 3 及 " Async.(255)" 使 CBCAN 使用异步传输来与从站交换 6 个 bytes 数据。除此之外，在 TxPDO 表列画面下方有三个与 TxPDO 传输有关的参数，他们分别是 " Cyclic Time "、" Inhibit Time " 和 " Event Time "；因为 " Cyclic Time " 只使用在 Cyclic 的传输模式，所以在此范例并不需要特别设定，而 " Inhibit Time " 和 " Event Time " 是和异步传输 Async. (255)有关的参数，是此范例是需要设定的。但是此范例并没有对此二参数有特别要求，所以可以直接使用默认值 1000ms 和 100ms，它们的意义如下，不管有没有事件产生，" Event Time " 1000ms 代表连续封包的间隔最长不可超过 1000ms，而 " Inhibit Time " 100ms 代表最短时间不可少于 100ms。

Inhibit Time: 100 mS Event Time: 1000 mS Cyclic Time: 2000 mS

图 55: TXPDO 传输参数设定

设定 Misc.:

在 Misc. 设定页面中, 有三个参数可设定, 分别是 " HeartBeat "、" Parameter Zone " 及 " Sync. Master "; " Parameter Zone " 和 CANopen 通讯没有关系因此可以将其参数设定成 0 以关闭此功能, 因为此范例使用 " Async.(255) " 传输, 所以 " Sync. Master " 也可以不勾选将它关闭, 除非网络需求此 CBCAN 主站需扮演 Sync. Master 的角色。

有如此范例网络所需, CBCAN 每 1000ms 会送出一个 heartbeat 讯号, 所以 " Cycle Time " 要设定成 1000ms; 通常从站的 heartbeat 讯号也会被设定为每 1000ms 发送一次, 因此主站 CBCAN 模块的 " Guard Time " 要被设定一个比 1000ms 大的值 (譬如 1050ms) 来监测从站的 heartbeat 讯号。

下载组态到 CBCAN:

当上述设定都完成后, 便可以将组态下载到 CBCAN 了; 首先, 先点选主页面的 " Write " 按钮呼叫 CBCAN 的串行通信设置页面, 在此页面, 将 " Baud Rate " 参数设定成 Auto 让 EasyCANopener 软件自行侦测串行传输的速度设定, 然后点选 OK 按钮来下载组态到 CBCAN 模块。

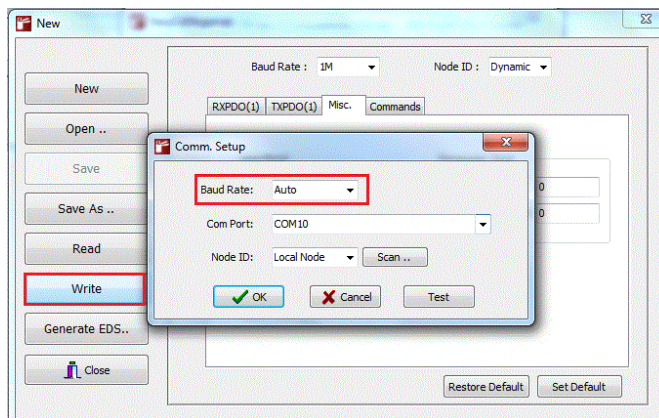


图 56: 将组态写入到 CBCAN

输入及输出数据映像:

在下载新组态到 CBCAN 模块后, CBCAN 模块应已自行重置且和从站开始通讯; CBCAN 最多可支持 10 个 RxPDO 和 10 个 TxPDO, 而且每个 PDO 都可包含 8 个 byte 的数据, 这些 PDO 的数据都被映像到 FBs PLC 的内部缓存器, 当 RxPDO 封包被 CBCAN 模块接收时, CBCAN 首先会将所有 RxPDO 的数据整合, 然后再映像到内部缓存器区 R3640~R3679, 相同的,

当 CBCAN 模块要传送 TxPDO 时，会将从内部缓存器区 R3600~R3639 将数据分别发送到个别的 TxPDO。

而此范例只有传输 6 个 byte 的数据，因此 RxPDO1 的数据被映像到 R3640~R3642 而 TxPDO1 的资料将从 R3600~R3602 读出。

1.1.3 范例三：两台 FBs PLC 透过 CBCAN 模块互相连接

此章节提供了一个实际的范例将两台 FBs PLC 透过 CBCAN 模块互连接，同时也提供了相对应的 EasyCanOpener 详细组态及 Ladder 程序让用户可以实机操作来了解 CBCAN 使用方式。此范例中的两台 PLC，一台是 FBs-24MC 在此范例是 CANopen 主站的角色，另外一台是 FBs-20MC 则是从站的角色，主站运作时将本机的数字输入状态(X0~X7)透过 CANopen 通讯传送到从站，而从站收到讯息后将状态写到从站的数字输出(Y0~Y7)；同时，从站也将本机的数字输入状态(X0~X7)传送到主站做为主站数字输出的状态值。

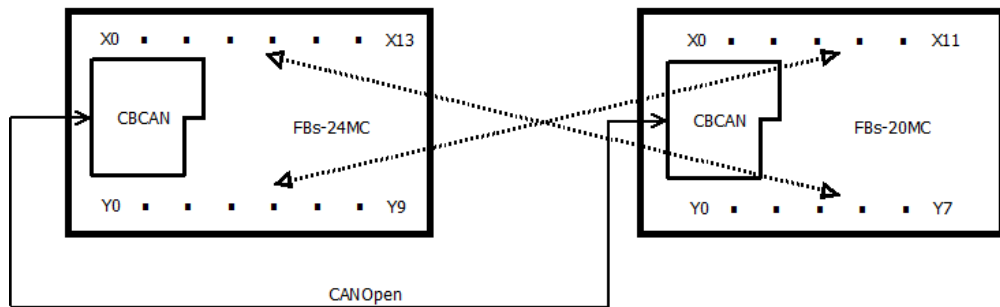


图 57: FBs-24MC 和 FBs-20MC 点对点通讯

在设定主站及从站的组态前，建议可以先行将所有组态先用表格方式列出以帮助系统设定。

表 14: 主站及从站组态表

模块名称	FBs-24MC	FBs-20MC
网络角色	主站	从站
速率	250 Kbps	
站台 ID	1	8
RXPDO #	1	1
RXPDO COB-ID	188h	208h
RXPDO 长度	1 (代表从 20MC 送来的数据长度不会超过一个 word)	1 (代表从 24MC 送来的数据长度不会超过一个 word)
TXPDO #	1 (代表要送到 20MC 的数据长度不会超过一个 word)	1 (代表要送到 24MC 的数据长度不会超过一个 word)
TXPDO COB-ID	208h	188h
TXPDO 长度	1	1
TXPDO 传输模式	Async. (255)	Async. (255)

Heart Beat 设定	此范例没有特殊要求, 因此使用默认值 Cycle Time: 1000 ms Guard Time: 1050 ms	此范例没有特殊要求, 因此使用默认值 Cycle Time: 1000 ms Guard Time: 1050 ms
Sync. Master 设定	OFF (Async. transmission)	OFF (Async. transmission)
Auto. Start 设定	ON (代表 PDO 传输会自动在开机后开始运作)	ON (代表 PDO 传输会自动在开机后开始运作)

接下来就要根据上述表格来设定主站及从站的组态, 其详细设定请参考图 58~图 63。

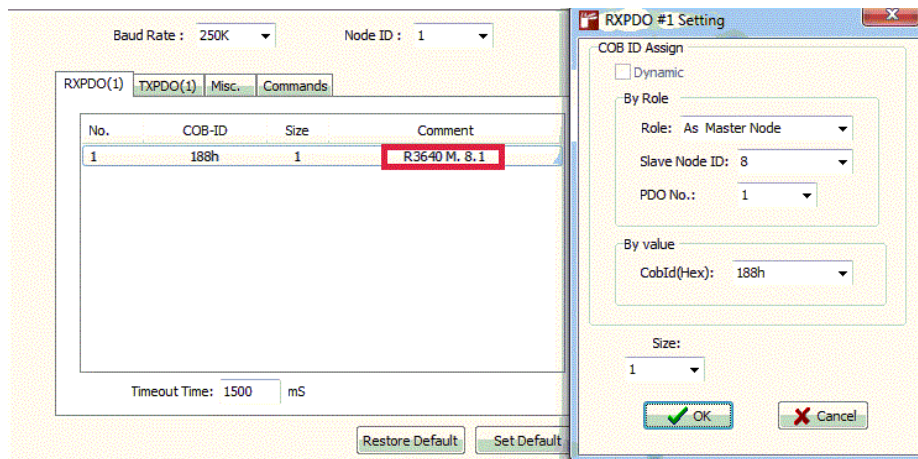


图 58: FBs-24MC RXPDO 设定

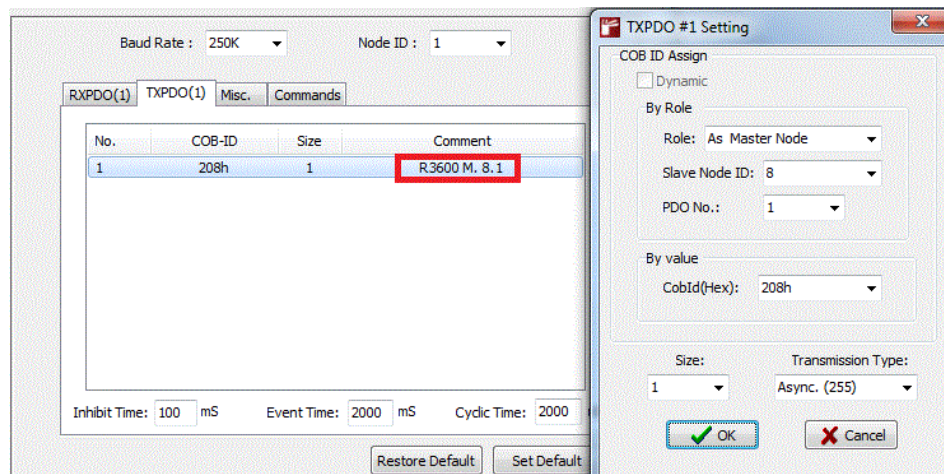


图 59: FBs-24MC TXPDO 设定

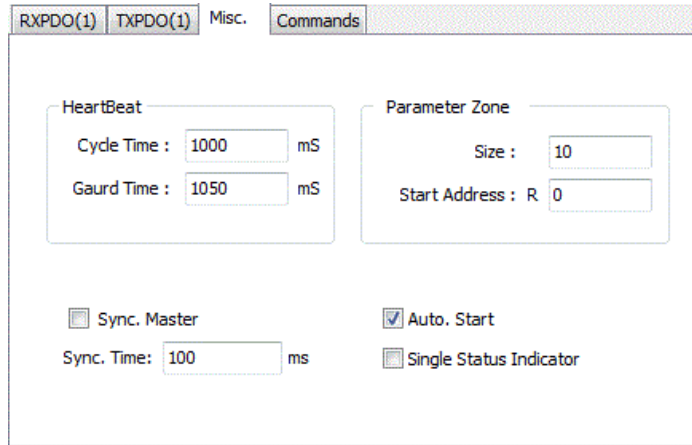


图 60: FBs-24MC Misc. 设定

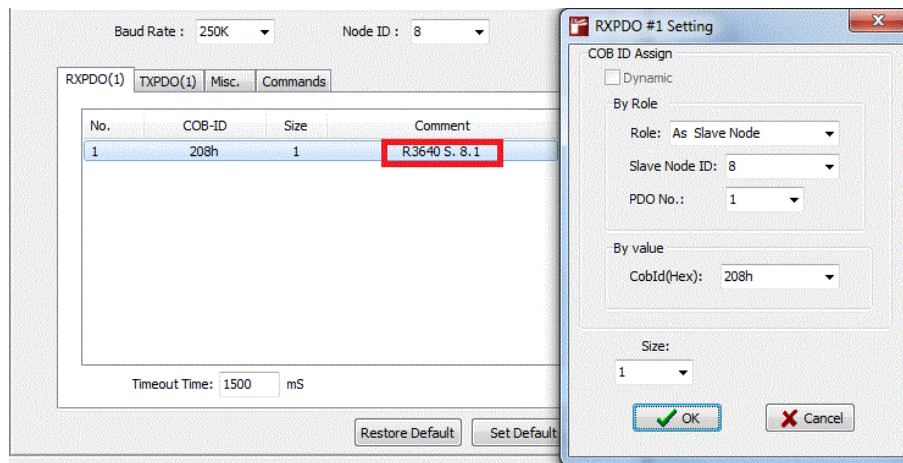


图 61: FBs-20MC RXPDO 设定

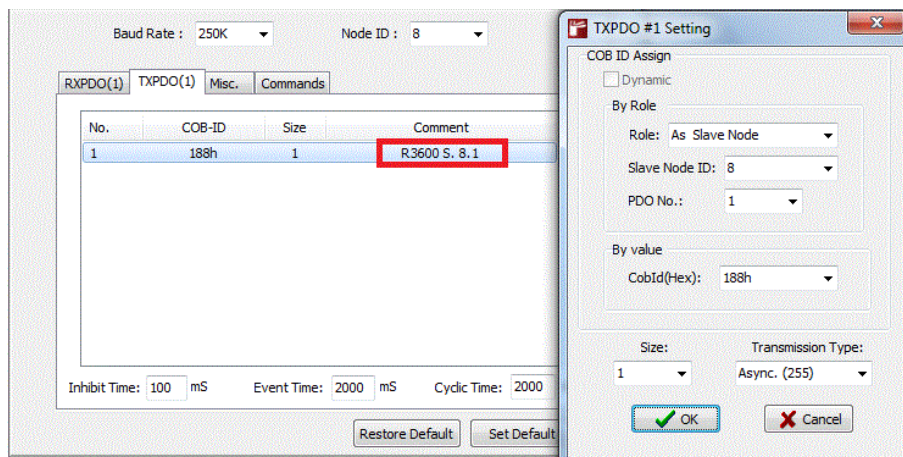


图 62: FBs-20MC TXPDO 设定

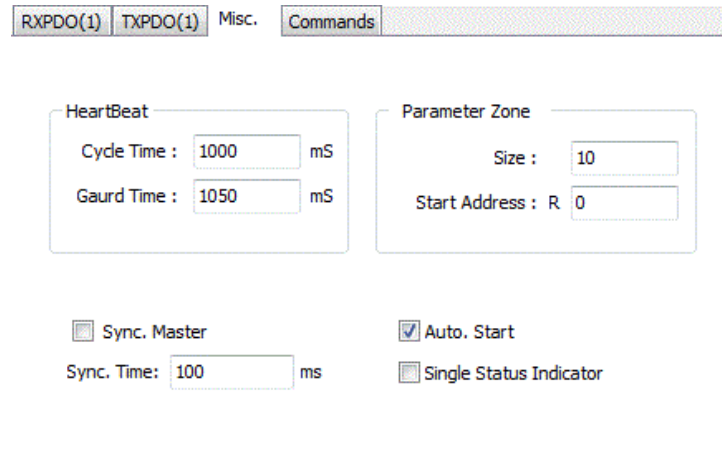


图 63: FBs-20MC Misc. 设定

在主站及从站都妥当设定了之后，主站及从站之间的 CANopen 通讯就会在系统重新启动后自行启动；但是因为尚未设计 Ladder 程序将两站各自本机的数字输入状态写入到 TXPDO 的，所以此时的 PDO 通讯封包里的信息尚未带有各站的数字输入状态，因此，最后一个步骤便是设计 ladder 程序将数字输入状态和 TXPDO 的传输连结在一起，实际作法是将本机的数字输入状态写入到 TXPDO 专用的 PLC 缓存器(TXPDO1: R3600~R3603)，在此范例除了发送端站台要将本机的数字输入状态传送给接收端站台外，接收端站台也需要在收到讯息后将状态写入到本机的数字输出，实际作法是将 RXPDO 的专用缓存器(RXPDO1: R3640~R3643)的值以 ladder move 的指令移动到本机的数字输出。

注：因为 24MC 及 20MC 的本机数字输入点数都没有超过 16 点，因此上述不管在 TXPDO 缓存器写入或 RXPDO 缓存器读出时，都只需要存取一个缓存器；在 EasyCanOpener 软件设定 RXPDO 及 TXPDO 的页面也有相对应的提示，请参阅图 58 及图 59。图 64 为详细的 Ladder 程序。

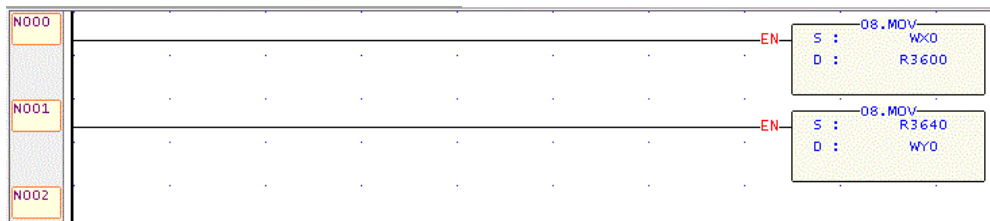


图 64: FBs-24MC 及 FBs-20MC 的 Ladder 程序

附录二

CBCAN 模板的对象字典(Object Dictionary)

. Communication Object

Index	SubIndex	Name	Data Type	Access	Default
1000H	0	Equipment Type	I32U	R	0
1001H	0	Error Register	I8U	R	0
1005H	0	COB-ID of SYNC	I32U	R	80H
100CH	0	Guard Time	I16U	R	0
100DH	0	Life Time Factor	I8U	R	0
Consumer Heartbeat Time					
1016H	0	Item Count	I8U	R	1
	1	Consumer Heartbeat Time	I32U	R	1020
1017H	0	Producer Heartbeat Time	I32U	R	1000
Identity Object					
1018H	0	Item Count	I8U	R	4
	1	Vendor code	I32U	R	2EFH
	2	Product Code	I32U	R	0
	3	Revision No.	I32U	R	0
	4	Serial No.	I32U	R	0
RXPDO1 Communication Parameters					
1400H	0	Item Count	I8U	R	2
	1	RXPDO1 COB-ID	I32U	R	CFG
	2	Transmission type	I8U	R	0xff
1401H~ 1409H	RXPDO2~RXPDO10 Communication Parameters				
RXPDO1 Mapped Objects					
1600H	0	Item Count	I8U	R	4
	1	RXPDO1 Mapped Object #1	I32U	R	CFG
	2	RXPDO1 Mapped Object #2	I32U	R	CFG
	3	RXPDO1 Mapped Object #3	I32U	R	CFG
	4	RXPDO1 Mapped Object #4	I32U	R	CFG
1601H~ 1609H	RXPDO2 ~ RXPDO10 Mapped Objects				

Communication Object (Cont.)

Index	SubIndex	Name	Data type	Access	Default
TXPDO1 Communication Parameters					
1800H	0	Item Count	I8U	R	5
	1	TXPDO1 COB-ID	I32U	R	CFG
	2	TXPDO1 Transmission Type	I32U	R	0xff
	3	TXPDO1 Inhibit Time	I32U	R	CFG
	4	Reserved	-	-	-
	5	Event Time	I16U	R	CFG
1801H ~ 1809H	TXPDO2 ~ TXPDO10 Communication Parameters				
TXPDO1 Mapped Objects					
1A00H	0	Item Count	I8U	R	4
	1	TXPDO1 Mapped Object #1	I32U	R	CFG
	2	TXPDO1 Mapped Object #2	I32U	R	CFG
	3	TXPDO1 Mapped Object #3	I32U	R	CFG
	4	TXPDO1 Mapped Object #4	I32U	R	CFG
1A01H ~ 1A09H	TXPDO2 ~ TXPDO10 Mapped Objects				

. PLC Parameter Zone Objects (Max. 1000 R register)

Index	SubIndex	Name	Data type	Access	Default
2000H	1	P Zone+0	WORD	RW	
2000H	2	P Zone+1	WORD	RW	
2000H	n	P Zone+n-1	.		
2000H	100	P Zone+99	WORD	RW	
2001H	1	P Zone+100	WORD	RW	
2001H	2	P Zone+101	WORD	RW	
~	~	~	WORD	RW	
2009H	100	P Zone+999	WORD	RW	

. PLC Process Data Objects

Index	SubIndex	Name	Data type	Access	Map
TXPDO process data					
2010H	1	R3600	I16U	R	Yes
	2	R3601		R	
	3	R3602		R	
	4	R3603		R	

2011H ~ 2019H	R3604~R3639				
RXPDO process data					
201AH	1	R3640	116U	R	Yes
	2	R3641		R	
	3	R3642		R	
	4	R3642		R	
201BH ~ 2023H	R3640~R3679				

.Program Version Objects

Index	SubIndex	Name	Data type	Access	Default
4000H	0	Item Count	I8U	R	2
	1	CBCAN Firmware version	I32U	R	
	2	Ladder software version	I32U	R	

Ladder software version 由 PLC 的 R3697 决定

. Electron board Version Objects

Index	SubIndex	Name	Data type	Access	Default
4001H	0	Item Count	I8U	R	1
	1	Electron board version	I32U	R	