

# CANNEXユーザーズマニュアル

株式会社サンプロシステム

改定：2025年3月26日

製品に関するお問い合わせ先

[product@sunprosys.co.jp](mailto:product@sunprosys.co.jp)

TEL：082-507-1067

「CANNEX（キャネックス）に関するお問い合わせ」とお伝えください

## ダウンロードリンク

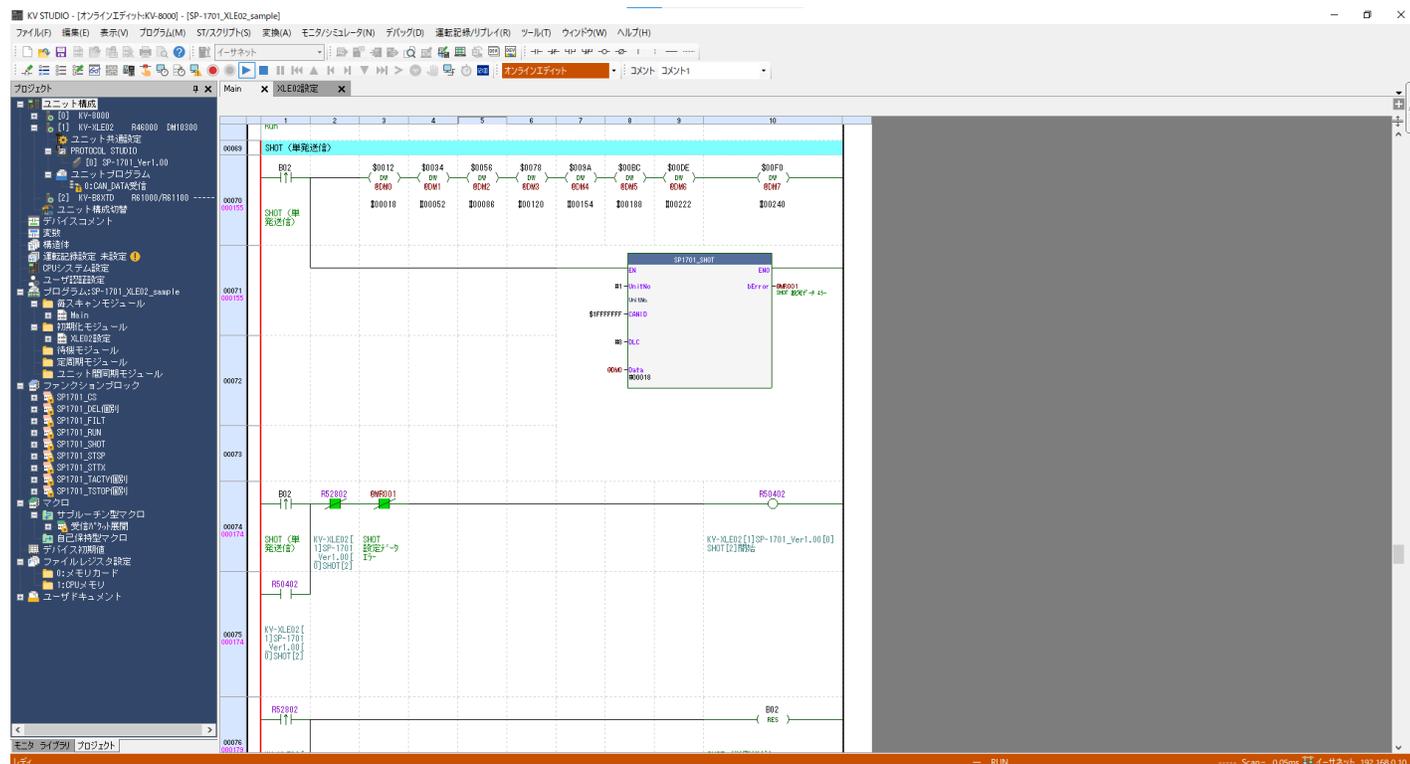
==== KV-Studioサンプルプロジェクト ====

KV-XLE02を使用したKV-Studioのサンプルプロジェクトです。

[CANNEX\\_XLE02\\_sample\\_240712.zip](#)

※ZIPの解凍パスワードはお問い合わせいただいたお客様に無償でお伝えします。

サンプルプロジェクトではKV-XLE02のポート1を使用しています。  
ポート2を使用する場合や1ポートで複数のCANNEXを制御する場合は  
お手数ですがメールにてお問い合わせご連絡ください。



# ==== CANNEX Control ====

Windowsのアプリケーションです。無料です。  
CANNEXへのコマンド発行、CAN通信のモニタができます。

25年03月26日公開 Version 1.2.4 最新版

24年07月10日公開 Version 1.2.2

## インストール手順

※初回起動時にファイアウォールの警告が出ます。

プライベートネットワーク、パブリックネットワークの両方にチェックを入れて  
アクセスを許可してください

The screenshot displays the CANNEX Control software interface, Version 1.0.1.0. The main window is divided into several sections:

- Top Panel:** Includes menu options (設定, 表示, ヘルプ) and control buttons (play, stop, refresh, etc.).
- Central Data Table:** A large table showing real-time CAN bus data. The columns include Index, Time, Dir, ID, and data bytes (D0-D11). The data is color-coded by direction (R for receive, T for transmit).
- Right Panel:** A summary table of statistics. The columns are '項目' (Item) and 'Count'.

項目	Count
CAN data	164634
CAN error	11
Missing data	0
BEF	0
EFW	0
EPF	0
BOEF	0
BORF	0
OVLV	0
BLF	0
ALF	11
SERR	0
FERR	0
AERR	0
CERR	0
BIERR	0
BOERR	0
ADERR	0
Bus占有率	55.67%
負荷OVER	0
- Bottom Panel:** Contains input fields for '表示ID(下限)' (700) and '表示ID(上限)' (1FFFFFFF), an 'Apply' button, and a 'CANNEX操作' button. A status bar at the bottom right shows '5 ms'.

## 概要

CAN/CAN FD通信 - Ethernet変換基板です。単純なレベル変換基板ではありません。伝文を定期送信するようにアプリケーションからCANNEXにコマンドで指令すると、CANNEX側で周期の管理を行い自動で定期送信するといったことができます。

任意のタイミングでの単発送信、複数IDの定期送信、データの一部を変えながら定期送信なども可能です。

相手機器からの受信データはKV-XLE02のユニットプログラムで受信します。そのためスキャンタイムに影響することなく受信することが可能です。

500us間隔で8byteのデータが絶え間なく発生するようなCAN通信や、10us間隔で8byteのデータが幾つか連続で送られてきたあと500us間隔に戻るといった頻度の通信であれば取りこぼしなく受信することが可能です。

高負荷のデータを取り扱う場合は何かしらの工夫が必要になる場合がありますので

取りこぼしに関する正確な見解が必要な場合は、ご相談ください。

CANのボーレート、データサイズ、データの周期が分かれば評価することができます。

UDP通信のできる機器であればCANNEXを使用してCAN通信を行うことができますが、弊社はKV-XLE02

(KEYENCE) を推奨しています。サンプルプログラムやマニュアルの説明はKV-XLE02を使用して記述しています。

**KV-8000 + KV-XLE02をご利用のお客様に関しては、ドライブレコーダなどのログを通じてサポートを提供できます。ただし、Windowsアプリケーションや他メーカーのシーケンサを使用されているお客様に関しては、お問い合わせに対応できない場合がございます。**

OMRON製PLC(CJシリーズ / NJ・NXシリーズ) :

Ethernetの受信割り込みができるユニットが存在しないため

高頻度で発生するデータに対応できません。

スキャンタイムに左右されます。

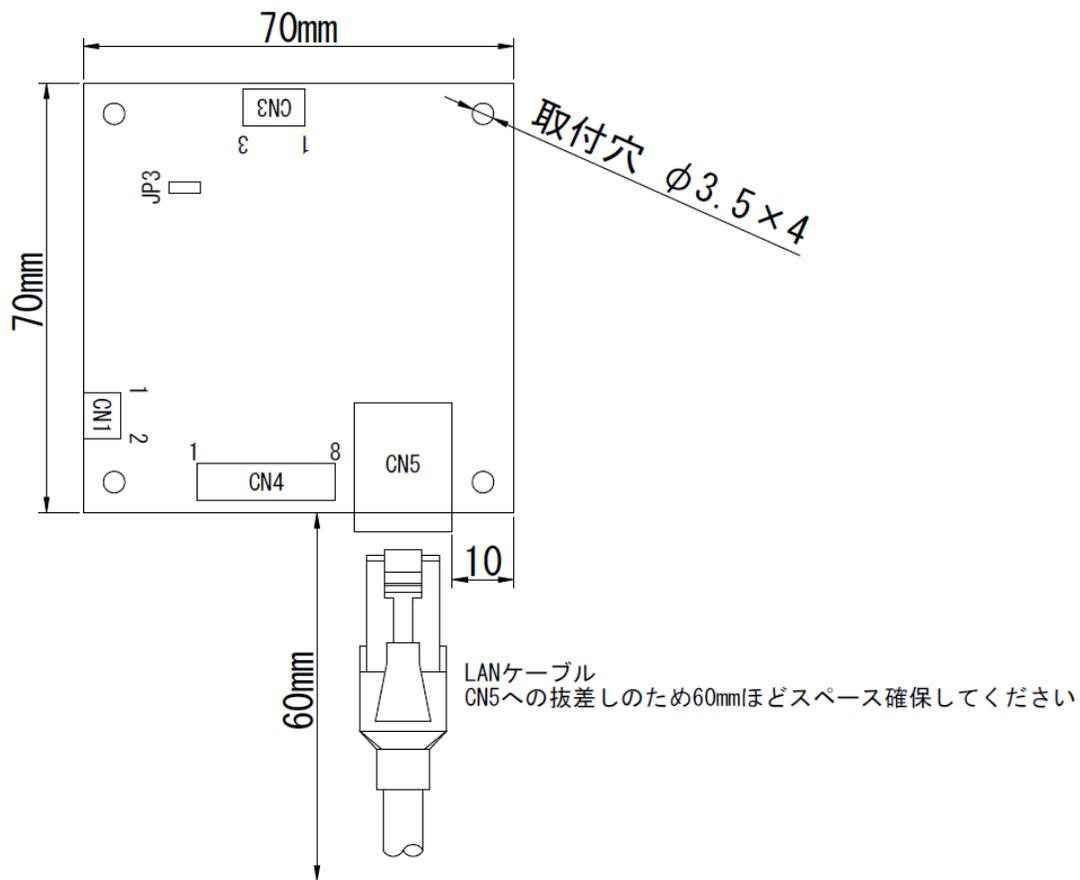
三菱製PLC(Qシリーズ、IQ-Rシリーズ) :

Ethernetの受信割り込みはEthernetユニットでできますが 実際に評価したところ3ms程度の周期で取りこぼしが発生しました。また割り込み処理速度、データ転送速度に関してメーカーが保証していないため非推奨です。

## 基板仕様

- 鉛フリー
- RoHS非対応

## 外形/インタフェース仕様



縦：70mm（LANコネクタ部突き出し部を含むと74mm）

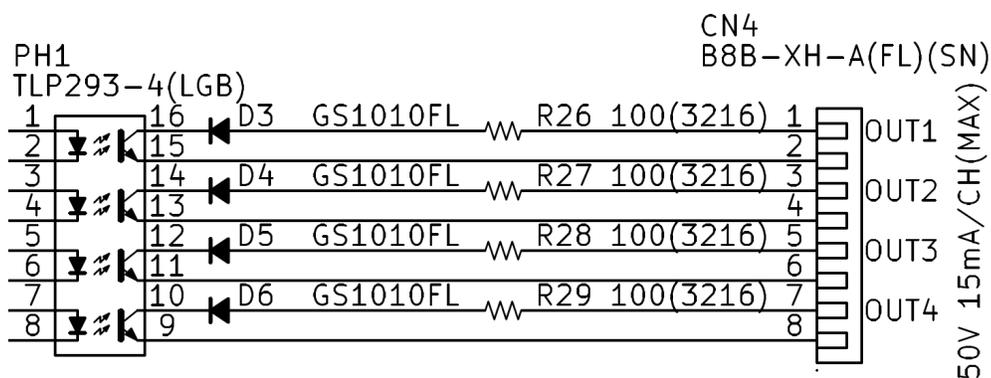
横：70mm

高さ：16mm

取付穴：φ3.5×4（端から5mmの位置）

コネクタ	内容	ピン配列	コネクタ型式
CN1	基板電源	1:DC12-24V 2:0V(※1)	基板側：B2B-XH-A 相手側：XHP-2 3mのコネクタ付きケーブル付属
CN3	CAN	1:CANH 2:CANL 3:GND(※1)	基板側：B3B-XH-A 相手側：XHP-3 3mのコネクタ付きケーブル付属
CN4	出力4点 OUT1： Run(CANNEX動作中) OUT2：Ethernet通信確立 OUT3：CAN Run OUT4：CANNEX 高負荷	下図参照(図： パラレル出力回路)	基板側：B8B-XH-A 相手側：XHP-8 3mのコネクタ付きケーブル付属
CN5	Ethernet CAT5e以上推奨 PLC/PC接続用	-	RJ-45 LANケーブルはお客様にてご用意ください (CAT5e UTPで可)
JP3	CAN終端抵抗有無切替	短絡:終端抵抗あり 解放:終端抵抗なし	出荷時ジャンパー装着(短絡)

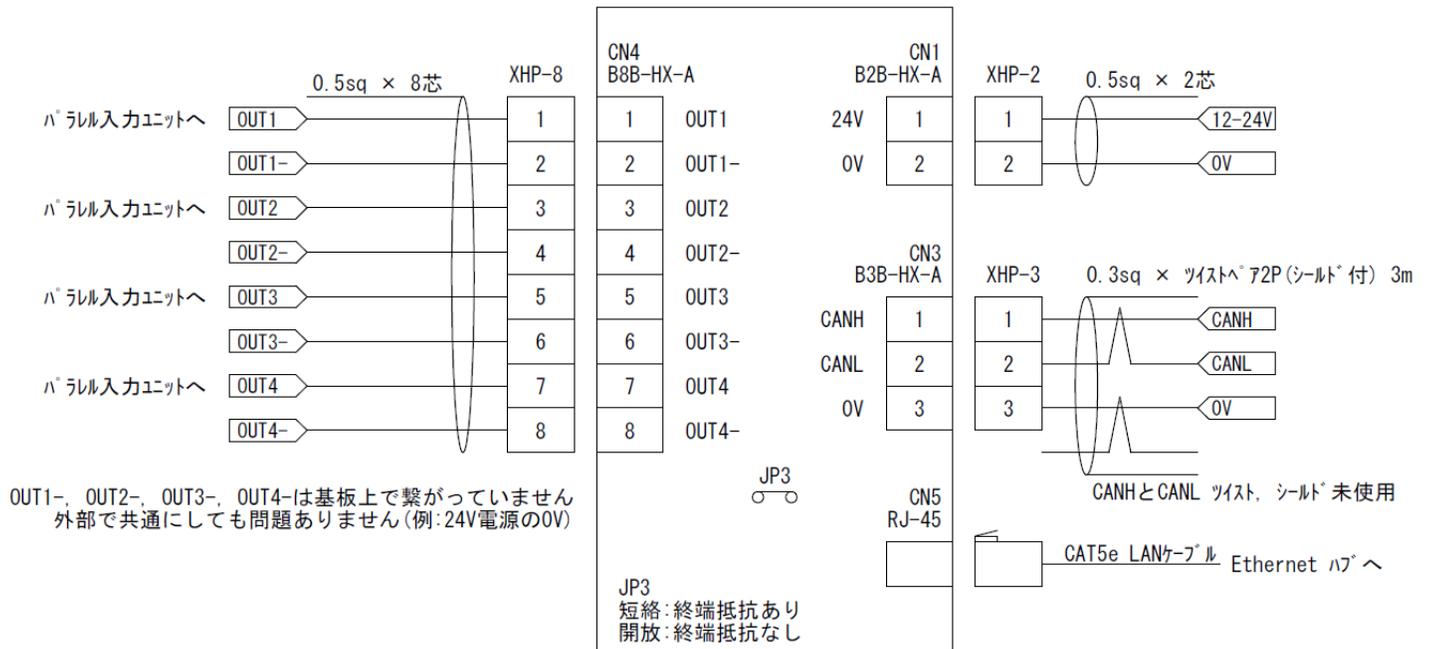
※1 0VとCAN GNDは絶縁されておらず、基板回路内で繋がりますのでご注意ください



図：パラレル出力回路

# 配線図

## PCB1 CAN-Ethernet変換基板 (CANEX)



図：制御配線図

OUT1-, OUT2-, OUT3-, OUT4-, 0Vはそれぞれ絶縁されていますが、共通にして使用しても構いません。  
 CN1の0VとCN3の0Vは基板内で繋がっています  
 CN1の24Vは12Vでも構いません。

# CAN仕様

## 回路

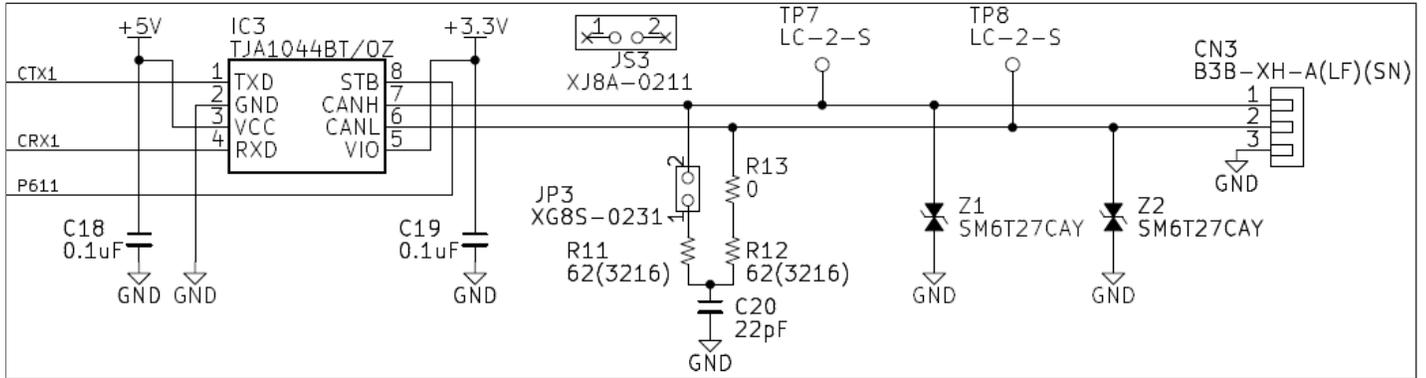


図 : CAN I/F回路

JP3 : 終端抵抗の有無切替

IC3 : CAN FDドライバIC(TJA1044BT/OZ ※NXP社製 または同等品)

## アービトレーションフェーズボーレート

125kbps / 250kbps / 500kbps / 1Mbps

## データフェーズボーレート

125kbps / 250kbps / 500kbps / 1Mbps / 2Mbps / 4Mbps / 5Mbps

## データサイズ

CAN:1-8 byte

CAN FD:0-8,12,16,20,24,32,48,64 byte

# Ethernet通信仕様

## IPアドレス（出荷時）

機器	IPアドレス	受信ポート	送信ポート
CANNEX	192.168.0.70	20000(変更不可)	不定
PLC	192.168.0.20	10000	ユーザ任意
モニタツール用PC	192.168.0.*	39999	不定

上記は初期値です。専用のパソコンツールで変更できます。

IPアドレスは192.168.0.1 - 192.168.9.254の範囲内で変更可能です  
サブネットマスクは255.255.255.0 固定です。

CANNEXのIPアドレスを変更することで

PLCのLANポート1つで複数のCANNEXを制御可能です。

(シーケンサCPUへの受信割り込みの頻度に注意する必要があります)

## CANNEXに対してPLCが送信するコマンド

### ● STOP（CAN通信停止）

Byte index	0	1	2	3
Data	S	T	O	P
内容	コマンド			

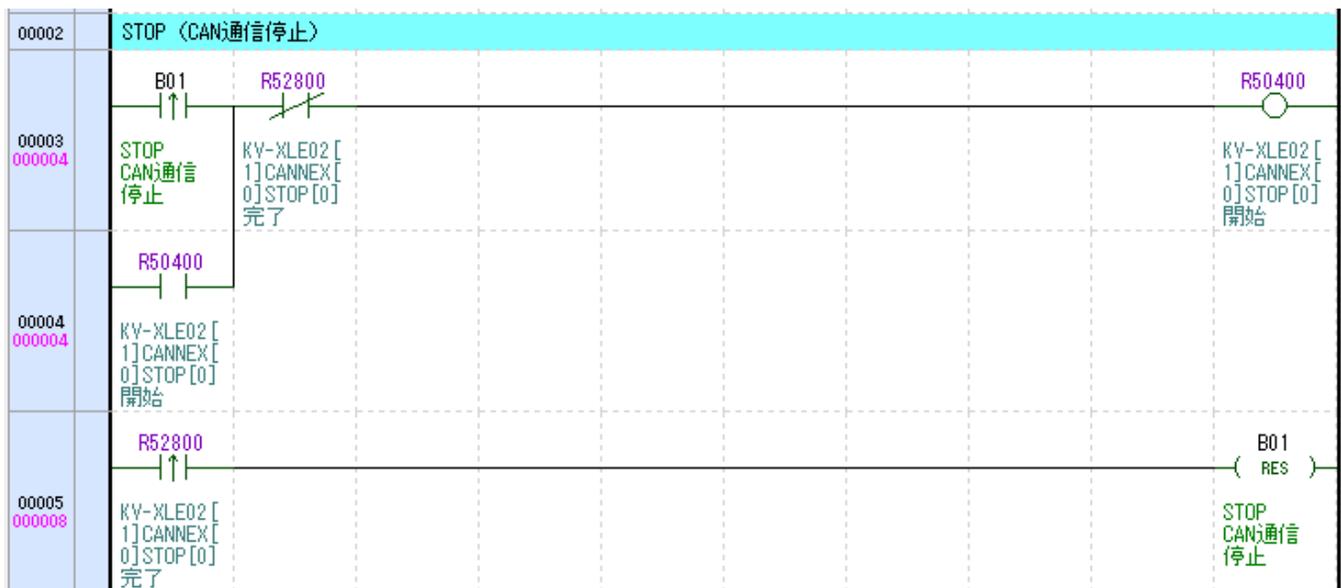
CAN通信を停止します。

停止状態になると、相手機器からCANデータがきてもACKを返さなくなります。

==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー
	名称	形態			
0	STOP	イベント	送信のみ	R50400	R52800

図：Protocol Studio設定



図：ラダープログラム

送信データ(HEX)	53	54	4F	50
送信データ(ASCII)	S	T	O	P

図：Ethernet通信モニタ

## ● RUN (CAN通信開始)

Byte index	0	1	2	3	4	5
Data	R	U	N	0-1	0-3	0-5
内容	コマンド			CAN種別	アービトレーションフェーズ ボーレート	データフェーズ ボーレート

CAN通信を開始します。

相手機器からCANデータが来た場合、ACKを返します。

STTXで定期送信を設定している場合、定期送信が開始されます。

- CAN種別
  - 0:CAN
  - 1:CAN FD
- アービトレーションフェーズボーレート
  - 0:125kbps
  - 1:250kbps
  - 2:500kbps
  - 3:1Mbps
- データフェーズボーレート
  - 0:125kbps
  - 1:250kbps
  - 2:500kbps
  - 3:1Mbps
  - 4:2Mbps
  - 5:5Mbps

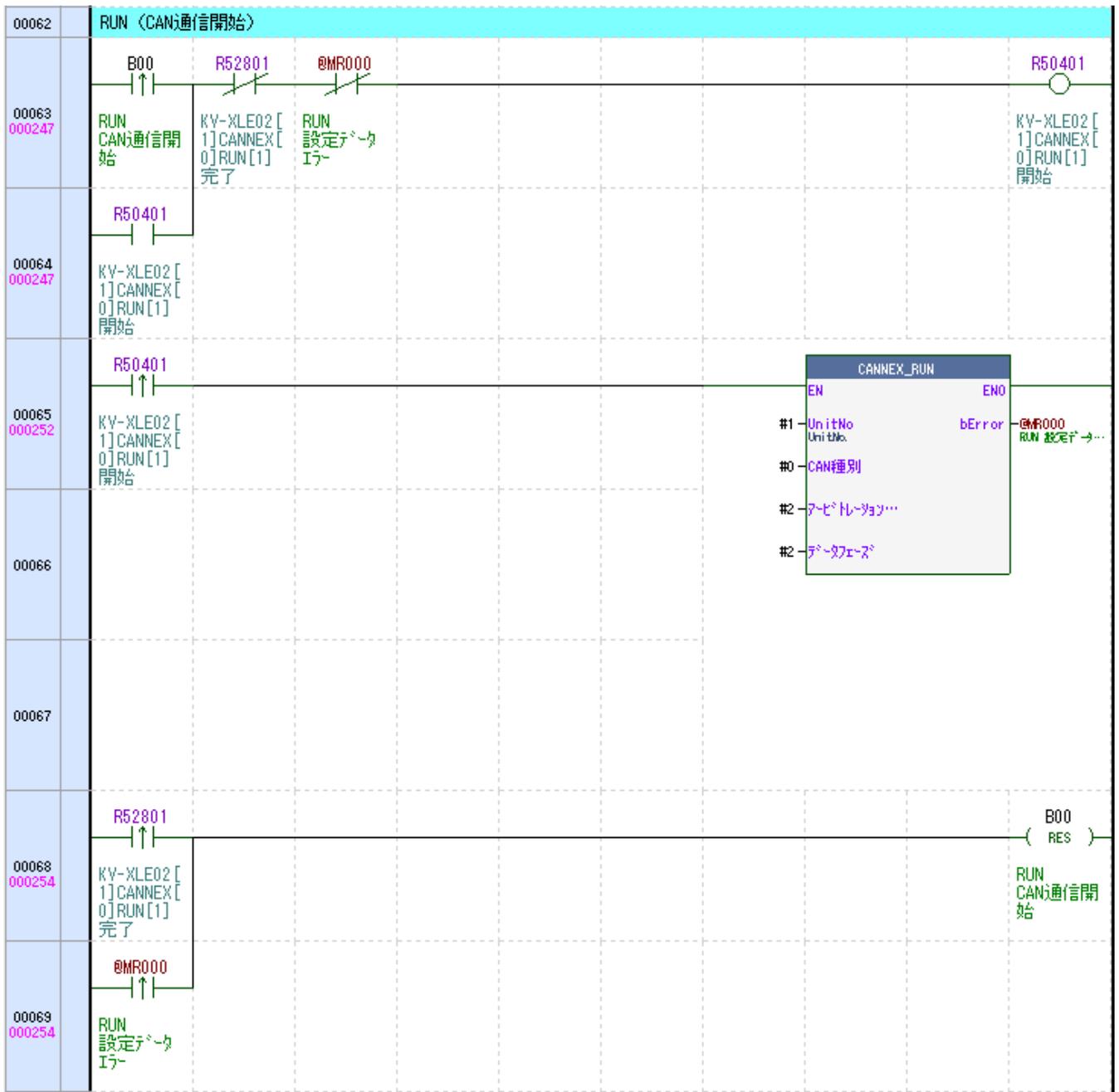
上記以外のボーレートには対応しておりません。カスタム仕様で対応可能な場合があります。ご相談ください

CAN種別を「0:CAN」に設定した状態で、8byteより大きなサイズのデータを送信しようとした場合、CANNEXはCANバスにデータ送出を行いません。

### ==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
1	RUN	イベント	送信のみ	R50401	R52801	UG35000 UG35001 UG35002	0:CAN 1:CANFD アービトレーション ボーレート データフェーズ ボーレート

図 : Protocol Studio設定



図：ラダープログラム

送信パラメータコメント	データ(ASCII定数)	0:CAN 1:CANFD	アービトレーション フェーズレート	データフェーズ フェーズレート
送信データ(HEX)	52 55 4E	01	02	05
送信データ(ASCII)	R U N	SOH	STX	ENQ

図：Ethernet通信の例

CAN種別：1 (CANFD)

アービトレーションフェーズボーレート：2 (500kbps)

データフェーズボーレート：5 (5Mbps)

## ● STTX (定期送信設定)

Byte index	0	1	2	3	4	5	6	7-8	9-12	13	14-77
Data	S	T	T	X	0-29	0-15	0-3	1-65535	0x01-0x1FFFFFFF	1-64	0x00-0xFF
内容	コマンド				送信MBX	Sweep番号	CS_TYPE	送信周期[ms]	CAN ID	DLC	Data0 - Data63

CAN送信の設定

STOP中、RUN中どちらでも設定可

設定済のMBX、Sweep番号に再設定すると上書きされる

### CS\_CNT\_TYPE

定期送信中に送信周期の異なる設定を上書きした場合、変更したタイミングで周期が1回だけ乱れます。ご注意ください。下図は送信周期1000msで定期送信中、Index = 30 送信後に 送信周期500msの設定を上書きした例です。

Index = 30とIndex = 31の間隔が1000 - 1500msの間で不定になり、その後500ms周期で定期送信が行われます

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
15	21383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
16	22383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
17	23383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
18	24383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
19	25383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
20	26383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
21	27383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
22	28383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
23	29383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
24	30383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
25	31383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
26	32383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
27	33383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
28	34383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
29	35383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
30	36383.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
31	37759.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
32	38259.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
33	38759.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
34	39259.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
35	39759.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
36	40259.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
37	40759.55	T	1010	8	12	34	56	78	9A	BC	DE	F0
38	41259.55	T	1010	8	12	34	56	78	9A	BC	DE	F0

図：RUN中に定期送信の周期変更した場合

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
3	STTX	イベント	送信のみ	R50403	R52803	UG35038 UG35039 UG35040 UG35041 UG35042 - UG35043 UG35044 - UG35076	送信 MBX SWEEP 番号 CS_CNT_TYPE 送信周期 CAN ID Data

図 : Protocol Studio設定

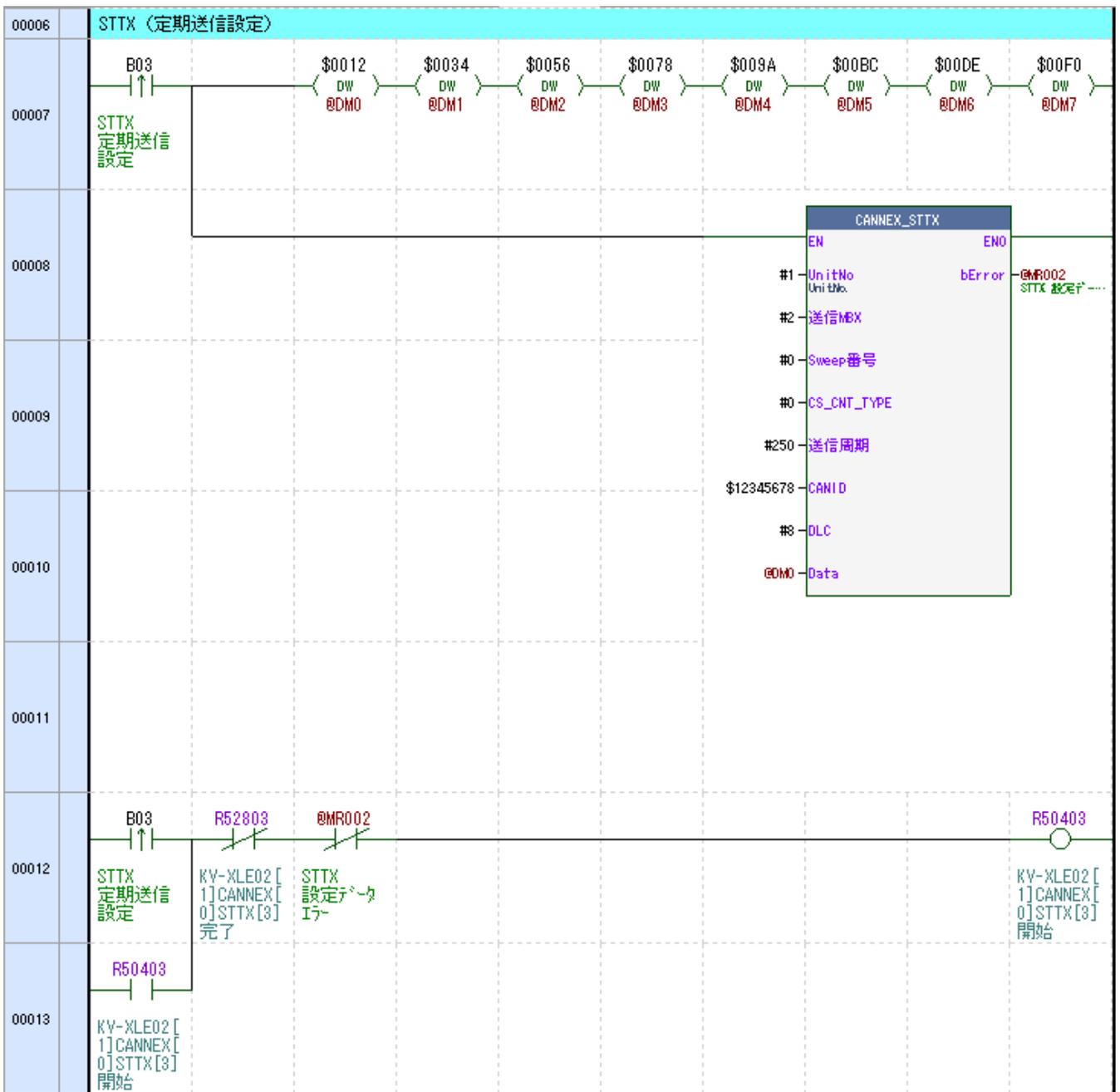
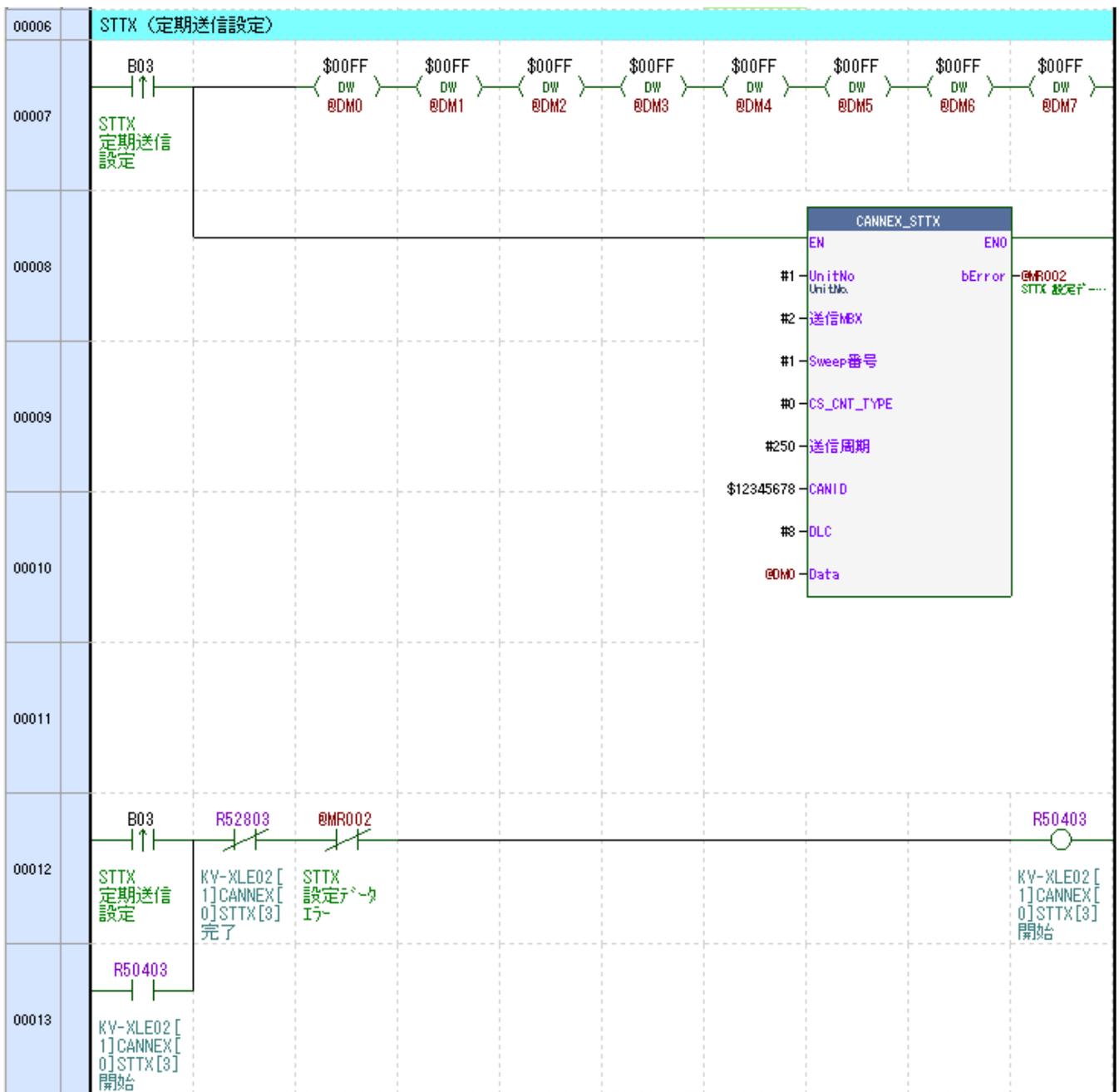


図 : ラダープログラム

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
0	249.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0
1	499.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0
2	749.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0
3	999.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0
4	1249.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0
5	1499.65	T	12345678	8	12	34	56	78	9A	BC	DE	F0

図：CAN通信データ

同じMBX番号にSweep番号1の伝文を設定することで下記のように交互に異なるデータを送信することができます (Sweep送信)



図：ラダープログラム

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
17	4499.57	T	12345678	8	12	34	56	78	9A	BC	DE	F0
18	4749.59	T	12345678	8	FF							
19	4999.57	T	12345678	8	12	34	56	78	9A	BC	DE	F0
20	5249.59	T	12345678	8	FF							
21	5499.57	T	12345678	8	12	34	56	78	9A	BC	DE	F0
22	5749.59	T	12345678	8	FF							
23	5999.57	T	12345678	8	12	34	56	78	9A	BC	DE	F0
24	6249.59	T	12345678	8	FF							
25	6499.57	T	12345678	8	12	34	56	78	9A	BC	DE	F0

図 : CAN通信データ

Sweep送信を使用する場合、Sweep番号0から連番で設定してください。

連番で設定しなかった場合、下記のような振る舞いになります。

MBX	Sweep	CS_CNT_TYPE	CANID	周期	DLC
0	0	0	111	1000	8
0	2	0	333	1000	8
0	3	0	444	1000	8
0	4	0	555	1000	8

図 : Sweep番号に空きがある設定

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
0	1000.36	T	111	8	00	00	00	00	00	00	00	00
1	3000.36	T	333	8	00	00	00	00	00	00	00	00
2	4000.37	T	444	8	00	00	00	00	00	00	00	00
3	5000.36	T	555	8	00	00	00	00	00	00	00	00
4	6000.36	T	111	8	00	00	00	00	00	00	00	00
5	8000.36	T	333	8	00	00	00	00	00	00	00	00
6	9000.37	T	444	8	00	00	00	00	00	00	00	00
7	10000.36	T	555	8	00	00	00	00	00	00	00	00
8	11000.36	T	111	8	00	00	00	00	00	00	00	00
9	13000.36	T	333	8	00	00	00	00	00	00	00	00
10	14000.37	T	444	8	00	00	00	00	00	00	00	00
11	15000.36	T	555	8	00	00	00	00	00	00	00	00

図 : Sweep番号に空きがある場合のCAN通信データ

送信パラメータコメント	データ(ASCII定数)	送信MBX	SWEEP番号	CS	送信周期	CAN ID	データ長	Data												
送信データ(HEX)	53 54 54 58	03	00	00	00 01	0D EF 00 AB	10	21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30												
送信データ(ASCII)	S T T X	ETX	NUL	NU	NUL SD	NU	DLE ! ~ # \$ % & ' ( ) * + , - . / 0													

図 : Ethernet通信の例

送信MBX : 3  
 SWEEP番号 : 0  
 CS\_TYPE : 0  
 送信周期 : 1ms  
 CANID : 0x00ABCDEF  
 データ長 : 0x10 (16byte)  
 送信Data : 0x21, 0x22, 0x23... 0x30

## ● SHOT（単発送信）

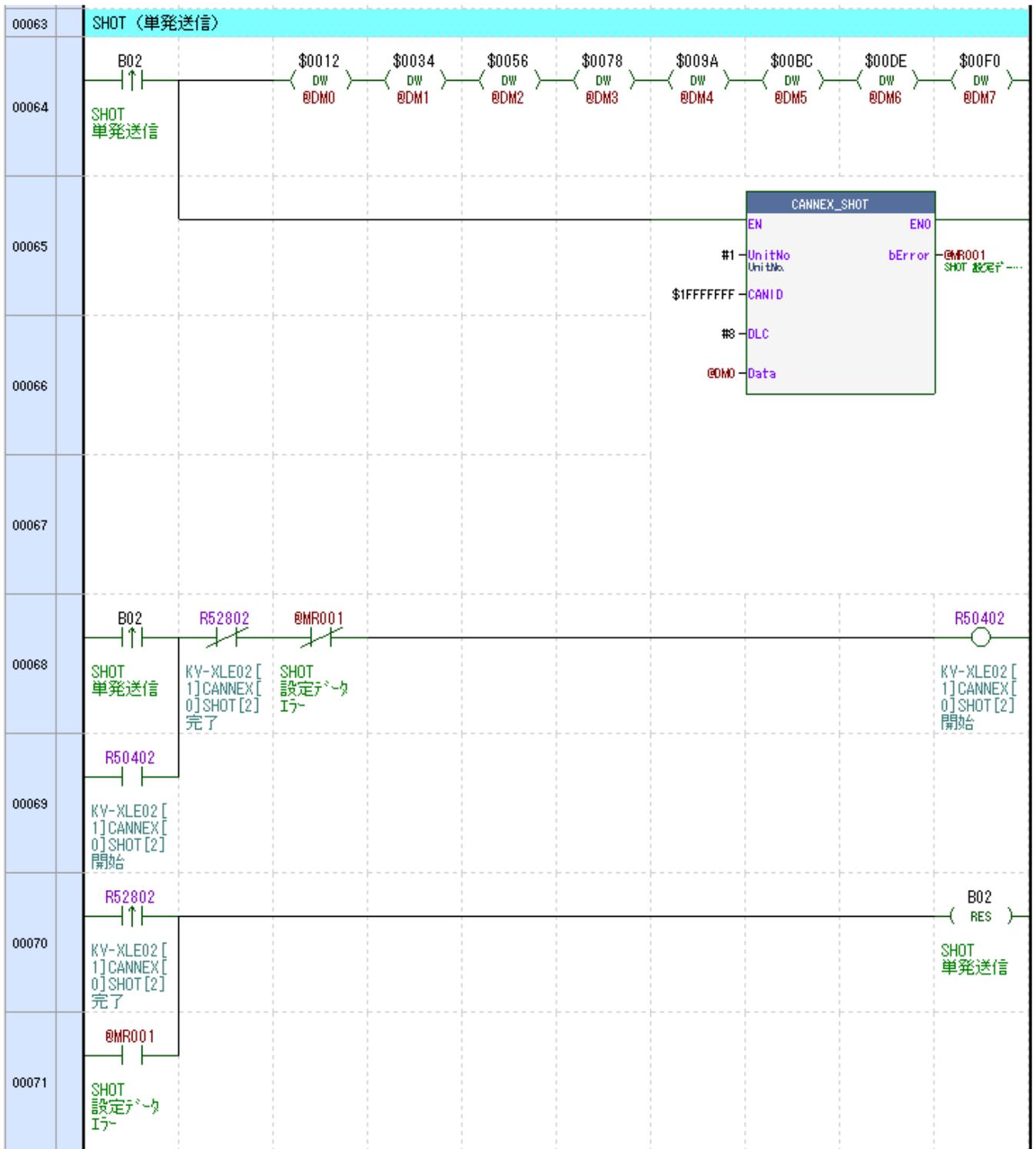
Byte index	0	1	2	3	4-7	8	9-72
Data	S	H	O	T	0x01-0x1FFFFFFF	1-64	0x00-0xFF
内容	コマンド				CAN ID	DLC	Data0 - Data63

RUN中に送信すると、1shot送信する

==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
2	SHOT	イベント	送信のみ	R50402	R52802	UG35003 - UG35004	CANID
						UG35005 - UG35037	DATA

図：Protocol Studio設定



図：ラダープログラム

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
4	13424.28	T	1FFFFFFF	8	12	34	56	78	9A	BC	DE	F0

図：CAN通信データ

送信パラメータコメント	データ(ASCII定数)	CANID	データ長	DATA															
送信データ(HEX)	53 48 4F 54 56 78 12 34	10	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10																
送信データ(ASCII)	S H O T V × DC 4	DLE	S0 S1 ET E0 EN AC BE BS HT LF VT FF CR S0 SI DL																

図：Ethernet通信の例

CANID : 0x12345678

データ長 : 0x10 (16byte)

送信Data : 0x01, 0x02, 0x03... 0x10

## ● DEL（定期送信設定消去）

Byte index	0	1	2	3
Data	D	E	L	0-29, 99
内容	コマンド			送信MBX

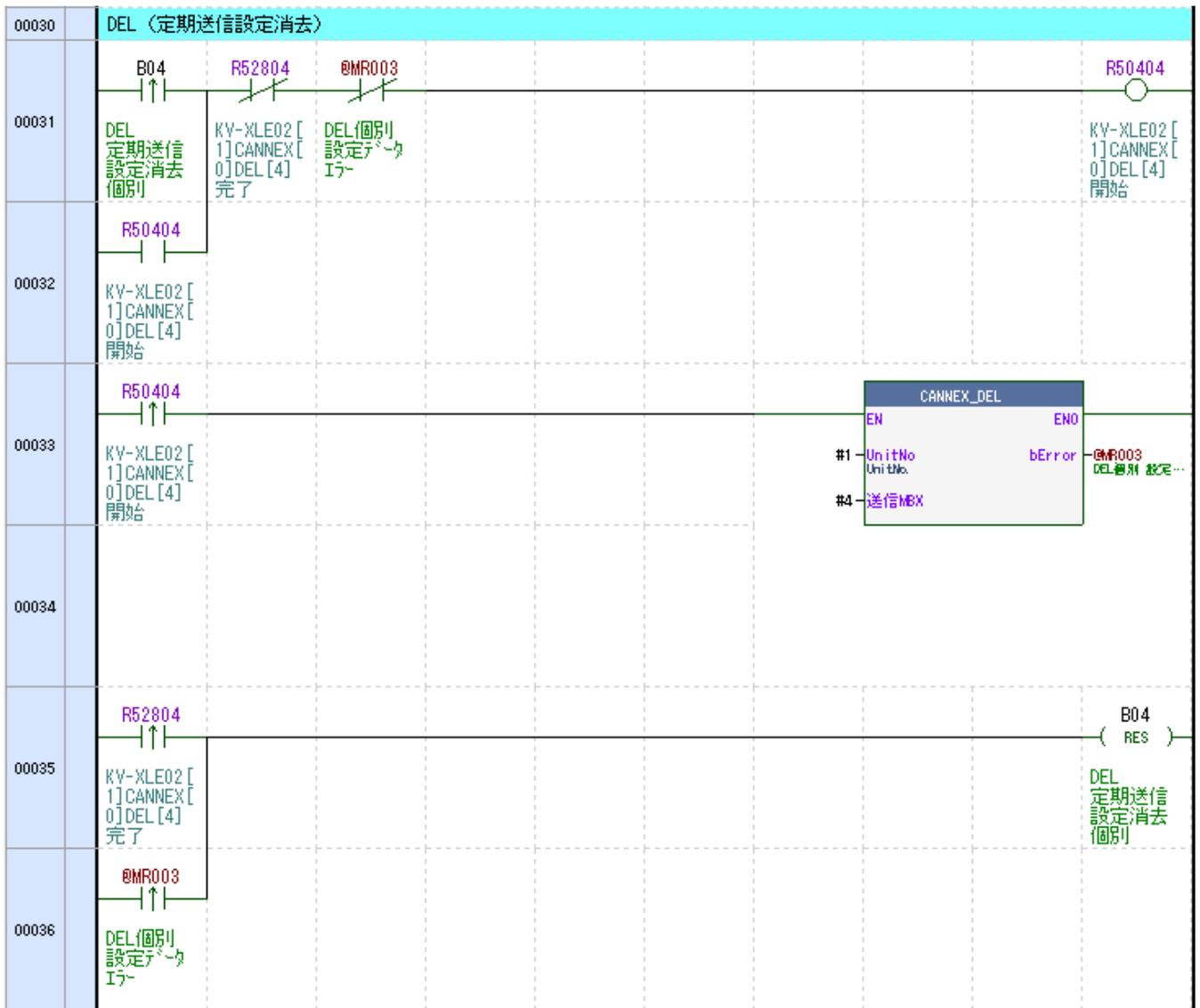
指定した送信MBXの設定を消去します

送信MBXに99を指定するか、「DEL」のみ送信すると全ての設定を消去します

==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
4	DEL	イベント	送信のみ	R50404	R52804	 UG35077	送信MBX

[図：Protocol Studio設定](#)



図：ラダープログラム

送信パラメータコメント	データ(ASCII定数)			送信MBX
送信データ(HEX)	44	45	4C	63
送信データ(ASCII)	D	E	L	c

図：Ethernet通信の例

送信MBX：0x63 (99) ※全ての送信設定を消去

## ● TSTOP（定期送信一時停止）

Byte index	0	1	2	3	4	5
Data	T	S	T	O	P	0-29, 99
内容	コマンド					送信MBX

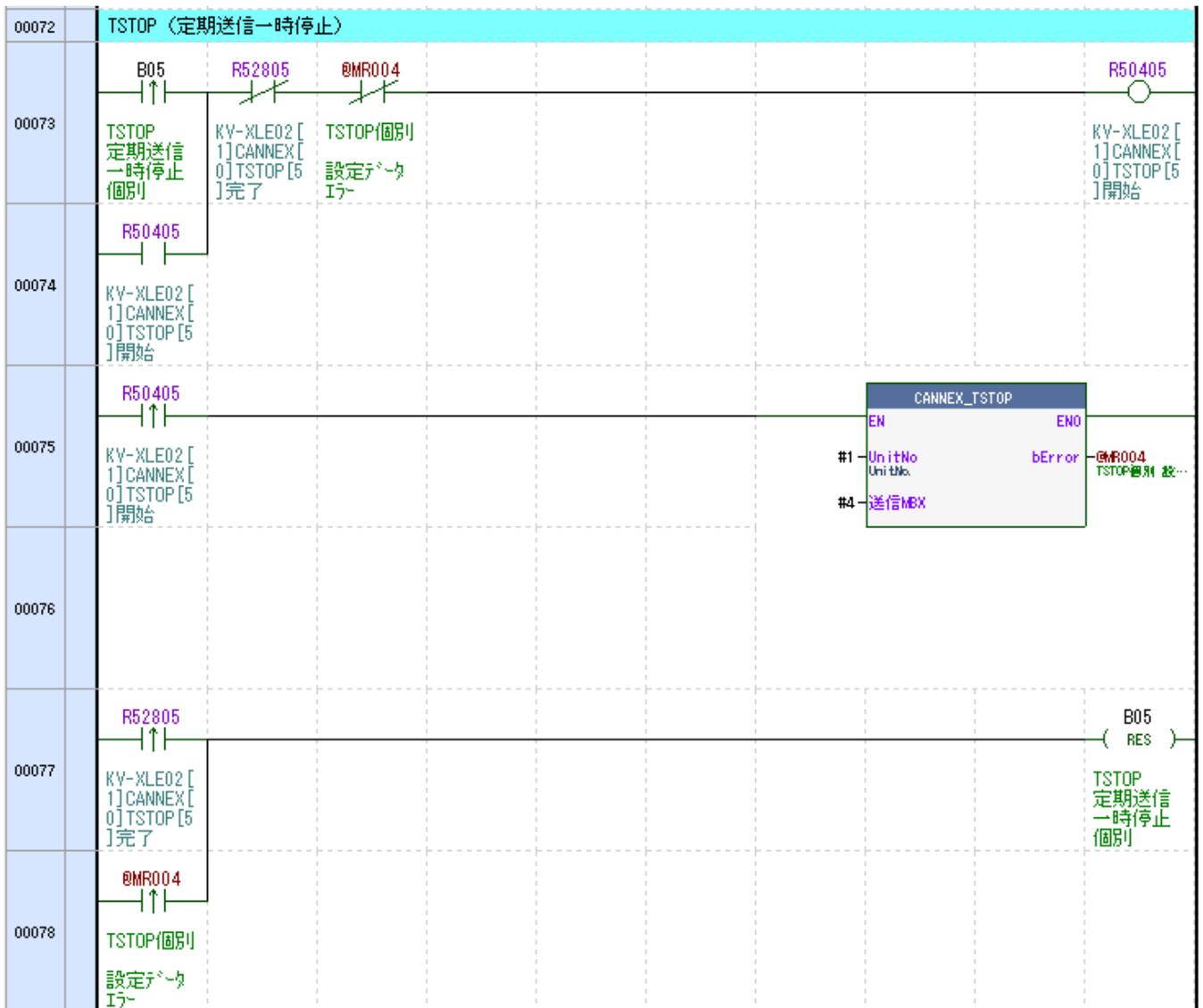
指定した送信MBXの設定を維持したまま、送信を停止します。

送信MBXに99を指定するか、「TSTOP」のみ送信すると全ての送信MBXの送信を停止します

==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
5	TSTOP	イベント	送信のみ	R50405	R52805	 UG35078	送信MBX

[図](#) : Protocol Studio設定



図：ラダープログラム

送信パラメータコメント	データ(ASCII定数)	送信MBX
送信データ(HEX)	54 53 54 4F 50	04
送信データ(ASCII)	T S T O P	EOT

図：Ethernet通信の例

送信MBX：0x04

※MBX04の定期送信を停止

## ● TACTV (定期送信 活性化)

Byte index	0	1	2	3	4	5
Data	T	A	C	T	V	0-29, 99
内容	コマンド					送信MBX

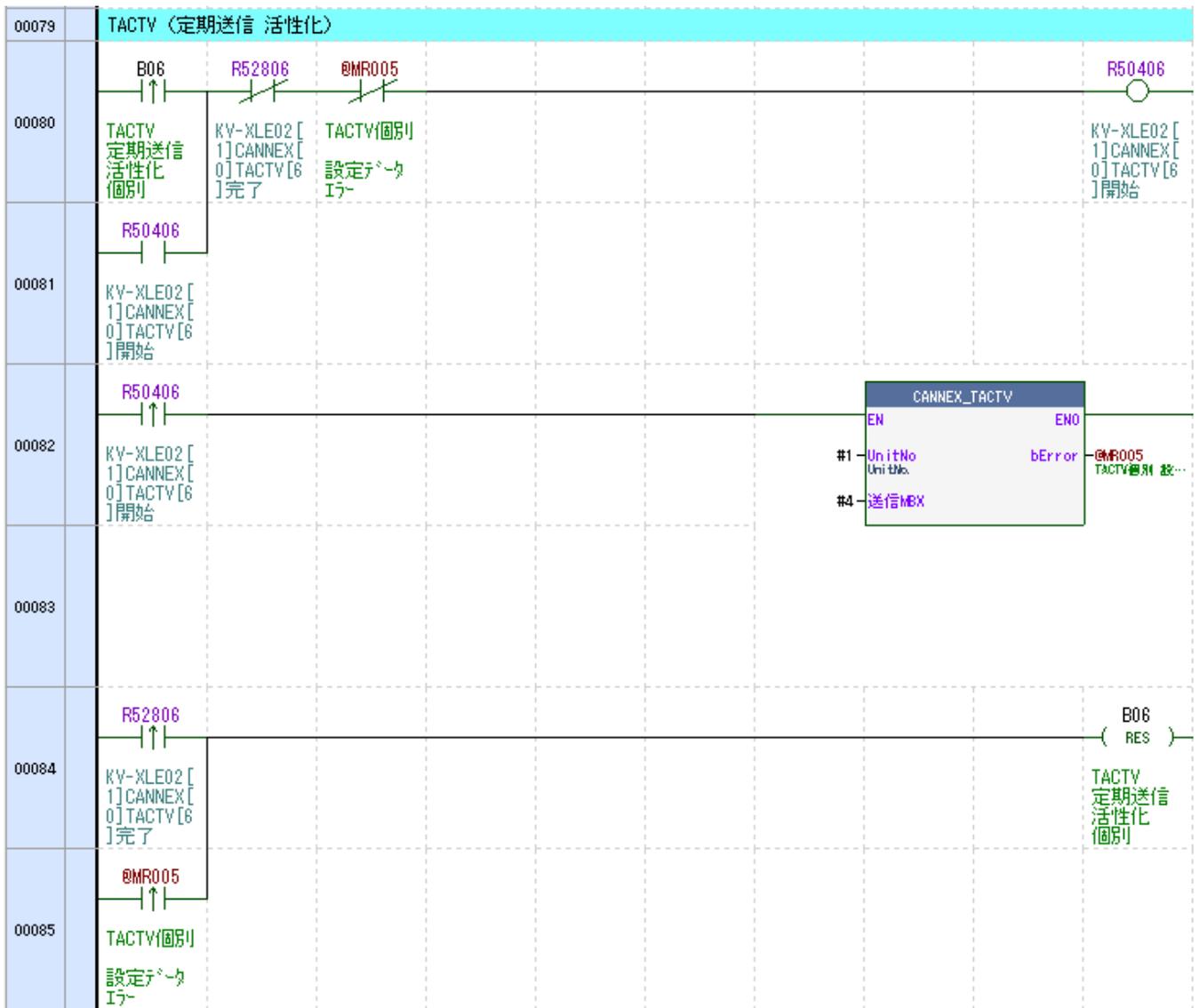
指定した送信MBXの送信を再開する

送信MBXに99を指定するか、「TACTV」のみ送信すると全ての送信MBXの送信を再開します

==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
6	TACTV	イベント	送信のみ	R50406	R52806	 UG95079	送信MBX

[図](#) : Protocol Studio設定



図：ラダープログラム

送信パラメータコメント	データ(ASCII定数)	送信MBX
送信データ(HEX)	54 41 43 54 56	04
送信データ(ASCII)	T A C T V	EOT

図：Ethernet通信の例

送信MBX：0x04

※MBX04の定期送信を再開

## ● FILT (UDPデータ送信フィルタ)

Byte index	0	1	2	3	4
Data	F	I	L	T	0-1
内容	コマンド				フィルタ種別

### フィルタ種別

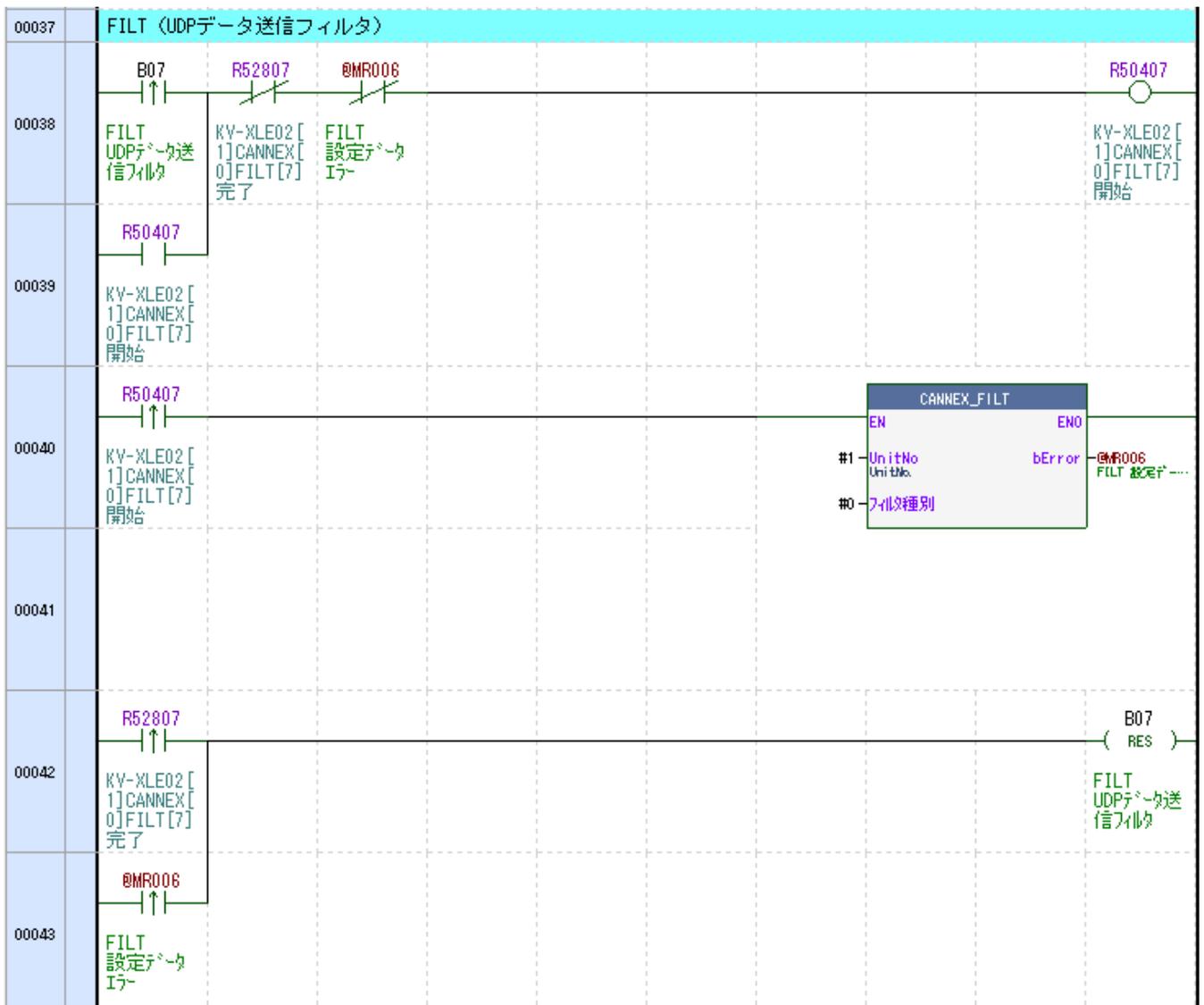
- 0. CANの送受信データを全てPLCに送信する
- 1. 送信MBXに登録されているCAN IDについては  
PLCに送信しない  
(受信したいIDと同じIDが送信MBXに存在する場合  
受信できなくなります)  
1shot(SHTX)についてはフィルタ対象外のため  
PLCにフィードバックがあります

※本コマンドはPLC側のEthernetの受信処理が追い付かず取りこぼしが発生する場合に、  
不要なデータをCANNEXからPLCに送信しないようにするために存在します

### ==== KV-Studio記述例 ====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
7	FILT	イベント	送信のみ	R50407	R52807	 UG35080	種別

[図 : Protocol Studio設定](#)



図：ラダープログラム

送信パラメータコメント	データ(ASCII定数)	種別
送信データ(HEX)	46 49 4D 54	00
送信データ(ASCII)	F I L T	NUL

図：Ethernet通信の例

フィルタ種別：0x00

※CANの送受信データを全てPLCに送信する（フィルタしない）

## ● CS (定期送信 カウント・チェックサム設定)

Byte index	0	1	2	3
Data	C	S	0-29	0-3
内容	コマンド		送信MBX	CS_CNT_TYPE

指定した送信MBXのCANデータの一部分を  
CANNEX側でチェックサムやカウントに置き換えます。  
(STTXコマンド内にも同様の機能があります)

### CS\_CNT\_TYPE

0. チェックサムやカウントなし

1. 下記の計算によるチェックサム、カウント

data6の下位4bit : カウンタ(0-F)

data7 : チェックサム(ID, DLC, Data0-6の加算)

```
p_snd_frame.data[6] += k_snd_can_ary[i].cnt % 0x10;
k_snd_can_ary[i].cnt++;
```

```
p_snd_frame.data[7] = (uint8_t)((((p_snd_frame.id & 0x0700) >> 8) +
    (p_snd_frame.id & 0x00ff) +
    p_snd_frame.data_length_code +
    p_snd_frame.data[0] +
    p_snd_frame.data[1] +
    p_snd_frame.data[2] +
    p_snd_frame.data[3] +
    p_snd_frame.data[4] +
    p_snd_frame.data[5] +
    p_snd_frame.data[6]));
```

CS_CNT_TYPE	CANID	周期	DLC	D0	D1	D2	D3	D4	D5	D6	D7
1	9A	100	8	01	02	03	04	05	06	00	00

図 : CS\_CNT\_TYPE = 1の例

Time	Dir	ID	D0	D1	D2	D3	D4	D5	D6	D7
288214.30	T	9A	01	02	03	04	05	06	00	B7
288314.30	T	9A	01	02	03	04	05	06	01	B8
288414.30	T	9A	01	02	03	04	05	06	02	B9
288514.30	T	9A	01	02	03	04	05	06	03	BA
288614.30	T	9A	01	02	03	04	05	06	04	BB
288714.30	T	9A	01	02	03	04	05	06	05	BC
288814.30	T	9A	01	02	03	04	05	06	06	BD
288914.31	T	9A	01	02	03	04	05	06	07	BE
289014.31	T	9A	01	02	03	04	05	06	08	BF
289114.31	T	9A	01	02	03	04	05	06	09	C0
289214.31	T	9A	01	02	03	04	05	06	0A	C1
289314.31	T	9A	01	02	03	04	05	06	0B	C2
289414.30	T	9A	01	02	03	04	05	06	0C	C3
289514.30	T	9A	01	02	03	04	05	06	0D	C4
289614.30	T	9A	01	02	03	04	05	06	0E	C5
289714.30	T	9A	01	02	03	04	05	06	0F	C6
289814.30	T	9A	01	02	03	04	05	06	00	B7
289914.30	T	9A	01	02	03	04	05	06	01	B8

図 : CS\_CNT\_TYPE = 1のCANデータの様子

## 2. カウント

data7の下位4bit : カウンタ(0-F)

CS_CNT_TYPE	CANID	周期	DLC	D0	D1	D2	D3	D4	D5	D6	D7
2	9A	100	8	01	02	03	04	05	06	00	00

図 : CS\_CNT\_TYPE = 2の例

Time	Dir	ID	D0	D1	D2	D3	D4	D5	D6	D7
657814.31	T	9A	01	02	03	04	05	06	00	00
657914.31	T	9A	01	02	03	04	05	06	00	01
658014.31	T	9A	01	02	03	04	05	06	00	02
658114.31	T	9A	01	02	03	04	05	06	00	03
658214.31	T	9A	01	02	03	04	05	06	00	04
658314.31	T	9A	01	02	03	04	05	06	00	05
658414.30	T	9A	01	02	03	04	05	06	00	06
658514.30	T	9A	01	02	03	04	05	06	00	07
658614.30	T	9A	01	02	03	04	05	06	00	08
658714.30	T	9A	01	02	03	04	05	06	00	09
658814.30	T	9A	01	02	03	04	05	06	00	0A
658914.31	T	9A	01	02	03	04	05	06	00	0B
659014.30	T	9A	01	02	03	04	05	06	00	0C
659114.30	T	9A	01	02	03	04	05	06	00	0D
659214.31	T	9A	01	02	03	04	05	06	00	0E
659314.30	T	9A	01	02	03	04	05	06	00	0F
659414.31	T	9A	01	02	03	04	05	06	00	00
659514.31	T	9A	01	02	03	04	05	06	00	01

図 : CS\_CNT\_TYPE = 2のCANデータの様子

## 3. 64bitカウント (D0-D7を使った64bitカウンタ)

```

p_snd_frame.data[1] = (cnt >> 48) & 0xFF;
p_snd_frame.data[2] = (cnt >> 40) & 0xFF;
p_snd_frame.data[3] = (cnt >> 32) & 0xFF;
p_snd_frame.data[4] = (cnt >> 24) & 0xFF;
p_snd_frame.data[5] = (cnt >> 16) & 0xFF;
p_snd_frame.data[6] = (cnt >> 8) & 0xFF;
p_snd_frame.data[7] = cnt & 0xFF;
cnt++;

```

CS_CNT_TYPE	CANID	周期	DLC	D0	D1	D2	D3	D4	D5	D6	D7
3	9A	1	8	00	00	00	00	00	00	00	00

図 : CS\_CNT\_TYPE = 3の例

D0 - D7のすべてを使用したカウンター動作になります

Time	Dir	ID	D0	D1	D2	D3	D4	D5	D6	D7
932761.31	T	9A	00	00	00	00	00	00	FF	FB
932762.31	T	9A	00	00	00	00	00	00	FF	FC
932763.31	T	9A	00	00	00	00	00	00	FF	FD
932764.31	T	9A	00	00	00	00	00	00	FF	FE
932765.31	T	9A	00	00	00	00	00	00	FF	FF
932766.31	T	9A	00	00	00	00	00	01	00	00
932767.31	T	9A	00	00	00	00	00	01	00	01
932768.31	T	9A	00	00	00	00	00	01	00	02
932769.31	T	9A	00	00	00	00	00	01	00	03
932770.31	T	9A	00	00	00	00	00	01	00	04
932771.31	T	9A	00	00	00	00	00	01	00	05
932772.31	T	9A	00	00	00	00	00	01	00	06
932773.32	T	9A	00	00	00	00	00	01	00	07
932774.31	T	9A	00	00	00	00	00	01	00	08
932775.31	T	9A	00	00	00	00	00	01	00	09

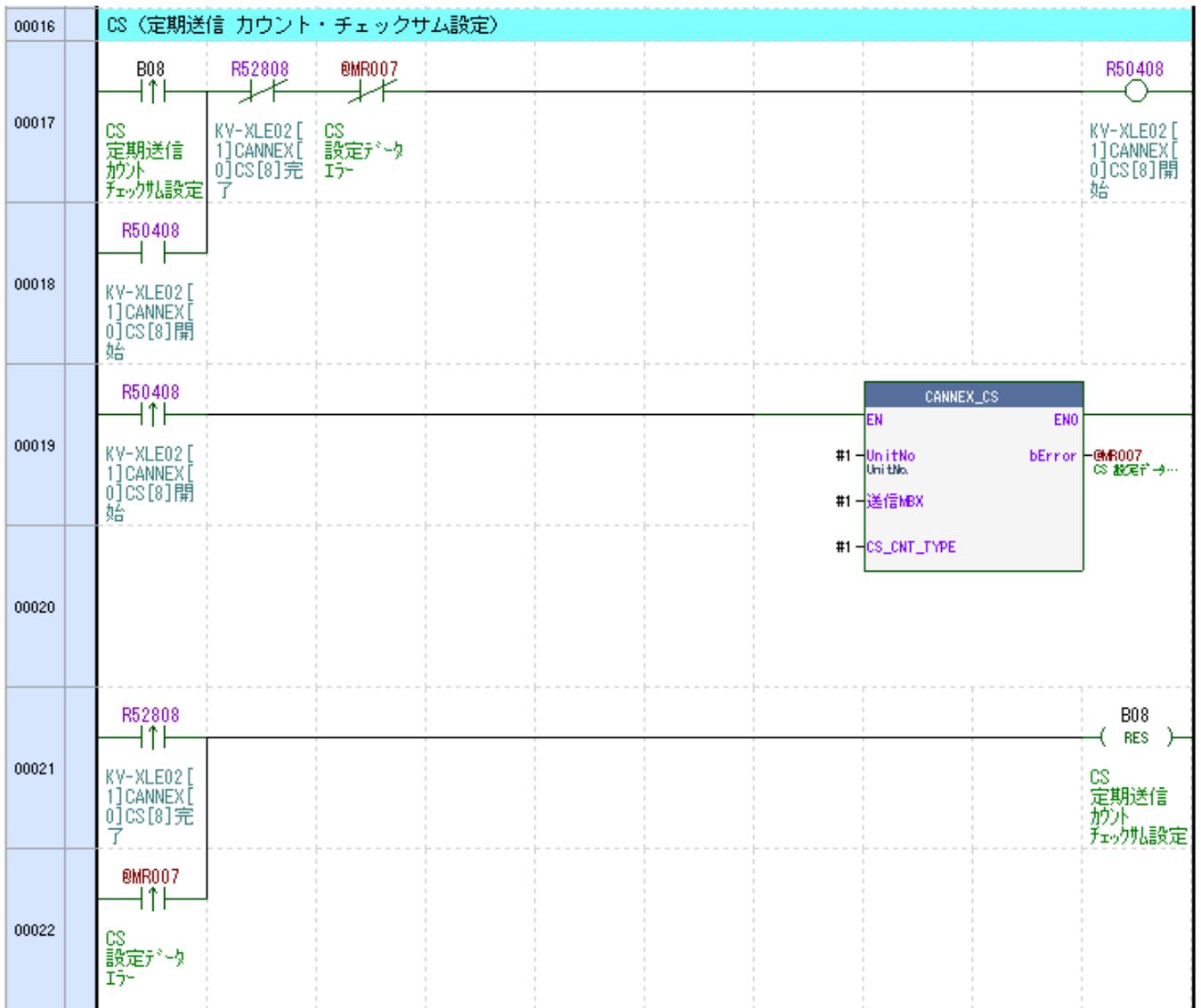
図 : CS\_CNT\_TYPE = 3のCANデータの様子

上記に当てはまらないチェックサム計算やカウンタのデータ位置を変えたい場合、CS\_CNT\_TYPEを追加しますのでご相談ください

==== KV-Studio記述例 =====

No.	通信コマンド		通信パターン	開始リレー	完了リレー	変数デバイス	パラメータコメント
	名称	形態					
8	CS	イベント	送信のみ	R50408	R52808	UG95081 UG95082	送信MBX CS_CNT_TYPE

図 : Protocol Studio設定



図：ラダープログラム

## ● STSP (Step No.設定)

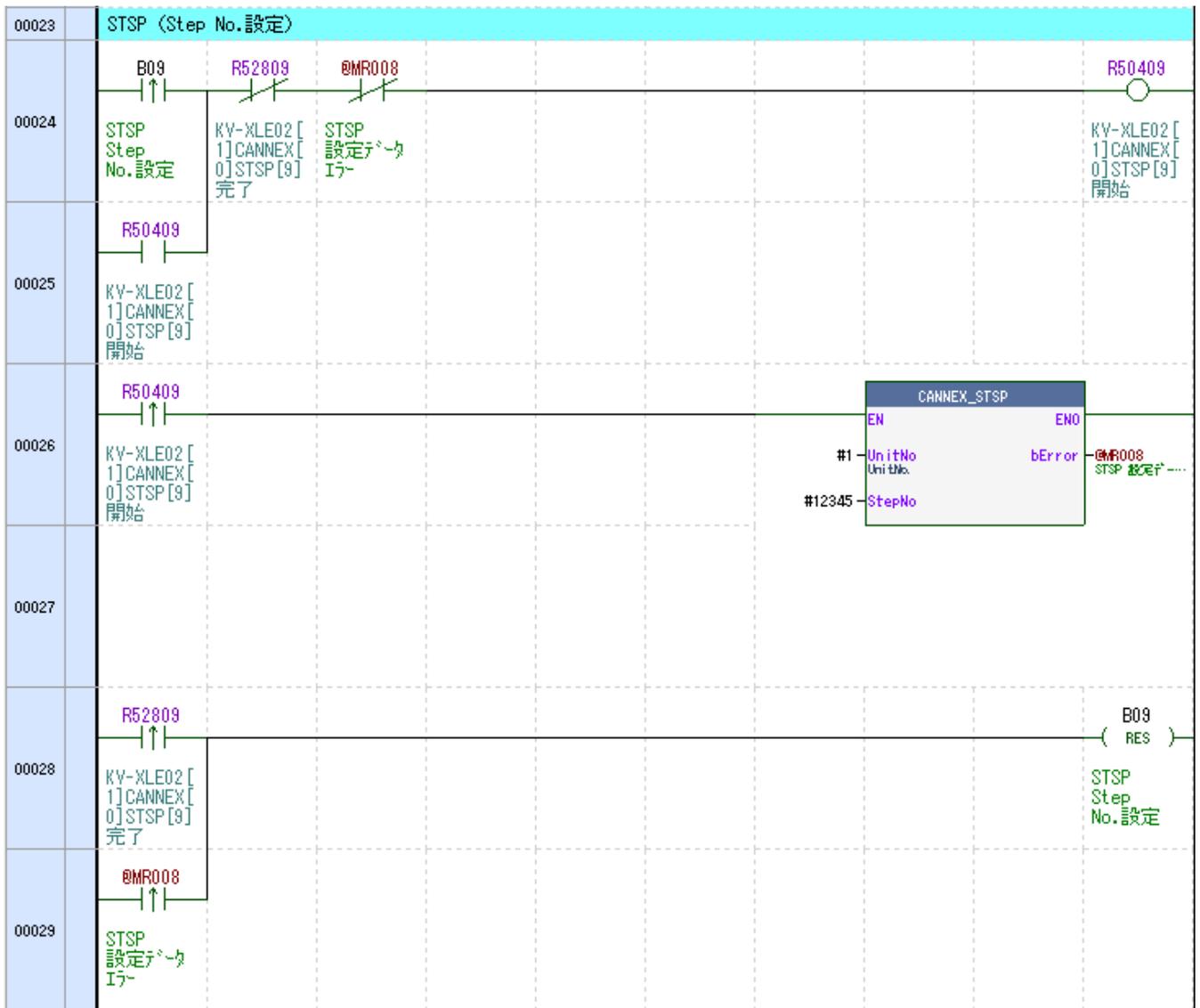
Byte index	0	1	2	3	4-5
Data	S	T	S	P	0-65535
内容	コマンド				Step No.

CANNEXから送られてくるCANデータに  
 任意の数値を付加してデバッグやロギングの際に活用できます。  
 本コマンドでStepNo.を設定するとCANNEXの電源を切るまでは  
 最後に設定したStepNo.が常時付加され続けます

==== KV-Studio記述例 =====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー	変数デバイス	パラメータコメント
	名称	形態					
9	STSP	イベント	送信のみ	R50409	R52809	 UG35083	Step No.

[図 : Protocol Studio設定](#)



図：ラダープログラム

Step No.を9999から12345に変更した例です  
ラダープログラムの検査シーケンスが進むにつれてStep No.を変更しておく  
シーケンスとCANデータの突き合わせをする際に便利です。

Index	Time	StepNo	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
234	118473.18	9999	T	12345678	8	FF							
235	118973.18	12345	T	12345678	8	FF							
236	119473.18	12345	T	12345678	8	FF							
237	119973.18	12345	T	12345678	8	FF							
238	120473.18	12345	T	12345678	8	FF							
239	120973.18	12345	T	12345678	8	FF							

図：CS\_CNT\_TYPE = 3のCANデータの様子

送信パラメータコメント	データ(ASCII定数)	Step No.
送信データ(HEX)	53 54 53 50 30 39	
送信データ(ASCII)	S T S P 0 9	

図：Ethernet通信の例

Step No. : 0x3039 (12345)



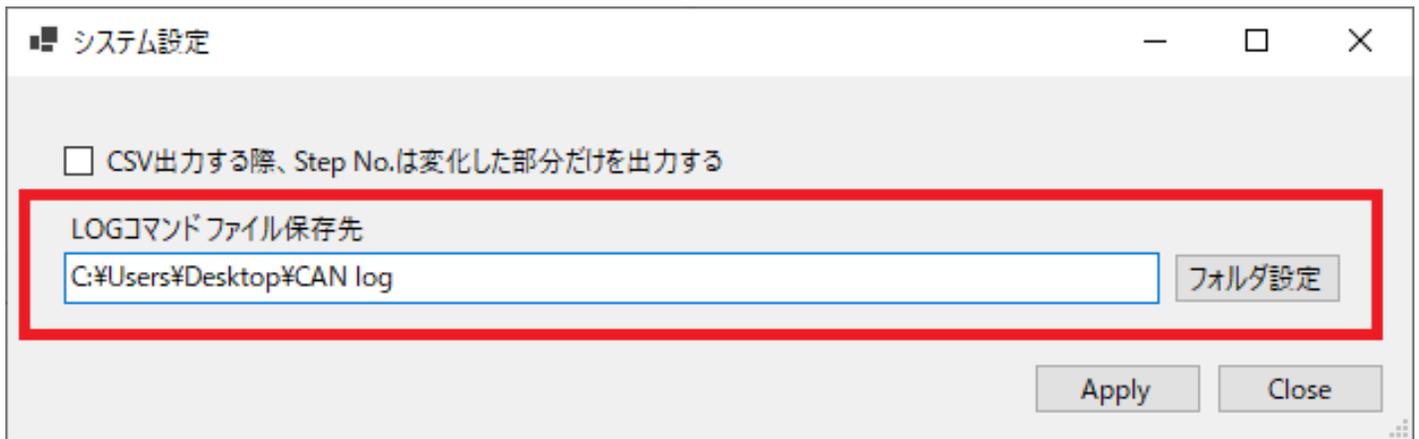
## LOG (パソコンアプリ CSVロギング指令)

Byte index	0	1	2
Data	L	O	G
内容	コマンド		

CANNEXをパソコンに接続して専用のアプリを起動した状態で

PLCから本コマンドをCANNEXに発行すると

パソコンはあらかじめ指定しておいたフォルダにCAN通信ログをCSVファイルで出力します。ファイル名はcan\_log\_YYMMddHHmmss.csvとなります。



図：LOG保存先の指定

デスクトップ > CAN log

名前	更新日時	種類	サイズ
can_log_240118_113715.csv	2024/01/18 11:37	Microsoft Excel CS...	156 KB
can_log_240118_113741.csv	2024/01/18 11:37	Microsoft Excel CS...	182 KB
can_log_240118_113818.csv	2024/01/18 11:38	Microsoft Excel CS...	1,137 KB

図：LOGコマンドで生成されるCSVファイル

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Index	Time	Step No.	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7
2	943	472973.18	12345	T	12345678	8	FF							
3	944	473473.18	12345	T	12345678	8	FF							
4	945	473973.18	12345	T	12345678	8	FF							
5	946	474473.18	12345	T	12345678	8	FF							
6	947	474973.18	12345	T	12345678	8	FF							
7	948	475473.18	12345	T	12345678	8	FF							
8	949	475973.18	12345	T	12345678	8	FF							
9	950	476473.18	12345	T	12345678	8	FF							
10	951	476973.18	12345	T	12345678	8	FF							

図：CSVファイルの内容

出力されるログの項目（列）は、CANNEX Controlの「表示」設定の状態に従います。  
例えば、「DLC表示」のチェックを外すと、CSVファイルにDLCの列は出力されません。

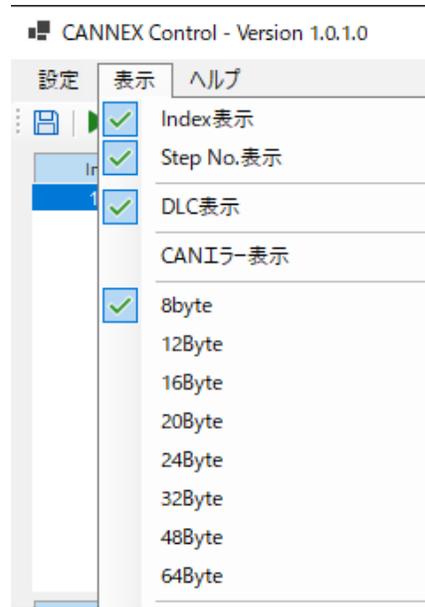


図 : CANNEX Control「表示」設定

CAN通信中にLOGコマンドを複数回送信した場合、  
一度CSV出力したCANデータは以降のLOGコマンドでは出力対象外となります。  
例えば、CANデータを10回分受信後にLOGコマンドを実行し、その後追加でCANデータを35回受信し、LOGコマンドを実行すると、  
1回目のCSVファイルには最初の10回分のデータが記録され、  
2回目のCSVファイルには最新の35回分のデータが記録されます。

==== KV-Studio記述例 =====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー
	名称	形態			
10	LOG	イベント	送信のみ	R50410	R52810

図 : Protocol Studio設定



図：ラダープログラム

## ● CLOG (パソコンアプリ ログクリア指令)

Byte index	0	1	2	3
Data	C	L	O	G
内容	コマンド			

CANNEXをパソコンに接続して専用のアプリを起動した状態で  
 PLCから本コマンドをCANNEXに発行すると  
 パソコン画面上的通信ログをクリアします  
 (ファイルの消去ではなく、画面表示のクリア)

==== KV-Studio記述例 =====

No.	通信コマンド		通信 パターン	開始 リレー	完了 リレー
	名称	形態			
11	CLOG	イベント	送信のみ	R50411	R52811

図 : Protocol Studio設定

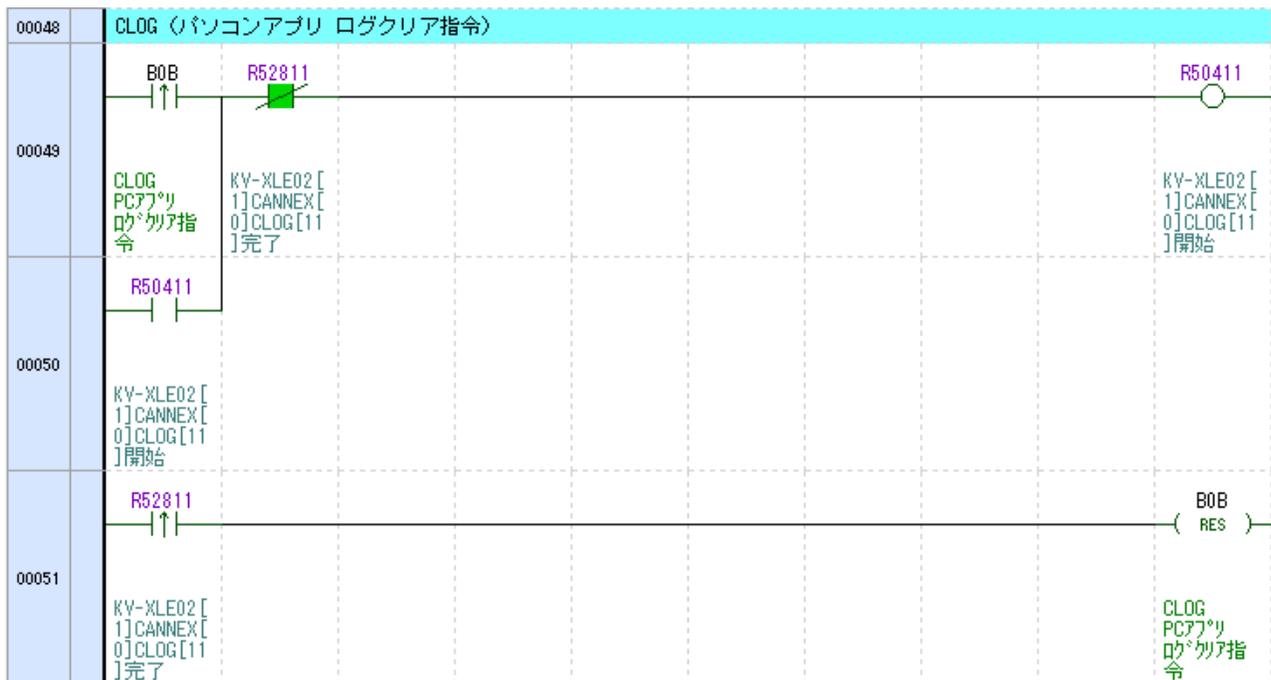


図 : ラダープログラム

# CANNEXがPLCやパソコンに送信するコマンド

## CAN受信データ

CANNEXがRUN中にCANデータを受信すると、下記のフレームをPLCに送信します。

CANNEXは自らが送信したCANデータも受信しているため、CANバス上のデータが全てEthernetで送出されます。

(FILTコマンドの設定による)

CANNEXが送信したCANデータは下記の「送受信種別」が0になります。

Byte index	0-3	4-7	8-9	10-11	12-15	16-17	18-19	20-23	24-25	26-89
Data	0x00000000-0xFFFFFFFF	0x00000000-0xFFFFFFFF	0-1	0-65535	0	0-0xFFFF	0-1	0x0-0xFFFFFFFF	1-64	0x00-0xFF
内容	Index	TimeStamp(10us)	MSG_TYPE	Step No.	Reserved	CAN Error	送受信種別	CAN ID	DLC	Data0 - Data63

※総データ数はDLCに関わらず、90Byte固定で送信します

Index

CANメッセージ毎にインクリメントします。

STOPコマンド → RUNコマンドで0に戻ります。

0xFFFFFFFFの次は0に戻ります。

MSG\_TYPE

0:CANデータ

1:Reserved(機能拡張用の予約。現在は未使用)

送受信種別

0:送信 (CANNEXが送信したCANデータ)

1:受信 (CANNEXが受信したCANデータ)

CAN Error

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADER R	BOER R	B1ER R	CERR	AERR	FERR	SERR	ALF	BLF	OVLf	BORf	BOEF	EPF	EWf	BEF

ビット	シンボル	機能	R/W
0	BEF	バスエラーフラグ 0: チャンネルバスエラー未検出 1: チャンネルバスエラー検出	R/W
1	EWF	エラーワーニングフラグ 0: チャンネルエラーワーニング未検出 1: チャンネルエラーワーニング検出	R/W
2	EPF	エラーパッシブフラグ 0: チャンネルエラーパッシブ未検出 1: チャンネルエラーパッシブ検出	R/W
3	BOEF	バスオフ開始フラグ 0: チャンネルバスオフ開始未検出 1: チャンネルバスオフ開始検出	R/W
4	BORF	バスオフ復帰フラグ 0: チャンネルバスオフ復帰未検出 1: チャンネルバスオフ復帰検出	R/W
5	OVLF	オーバーロードフラグ 0: チャンネルオーバーロード未検出 1: チャンネルオーバーロード検出	R/W
6	BLF	バスロックフラグ 0: チャンネルバスロック未検出 1: チャンネルバスロック検出	R/W
7	ALF	アービトレーションロストフラグ 0: チャンネルアービトレーションロスト未検出 1: チャンネルアービトレーションロスト検出	R/W
8	SERR	スタッフエラー 0: チャンネルスタッフエラー未検出 1: チャンネルスタッフエラー検出	R/W
9	FERR	フォームエラー 0: チャンネルフォームエラー未検出 1: チャンネルフォームエラー検出	R/W
10	AERR	ACK エラー 0: チャンネル ACK エラー未検出 1: チャンネル ACK エラー検出	R/W
11	CERR	CRC エラー 0: チャンネル CRC エラー未検出 1: チャンネル CRC エラー検出	R/W
12	B1ERR	ビット1エラー 0: チャンネルビット1エラー未検出 1: チャンネルビット1エラー検出	R/W
13	B0ERR	ビット0エラー 0: チャンネルビット0エラー未検出 1: チャンネルビット0エラー検出	R/W
14	ADERR	ACK デリミタエラー 0: チャンネル ACK デリミタエラー未検出 1: チャンネル ACK デリミタエラー検出	R/W

```

cString[0] = (uint8_t)((k_can_msg.index >> 8) & 0xFF);
cString[1] = (uint8_t)((k_can_msg.index >> 0) & 0xFF);
cString[2] = (uint8_t)((k_can_msg.index >> 24) & 0xFF);
cString[3] = (uint8_t)((k_can_msg.index >> 16) & 0xFF);

cString[4] = (uint8_t)((k_can_msg.ts_10us >> 8) & 0xFF);
cString[5] = (uint8_t)((k_can_msg.ts_10us >> 0) & 0xFF);
cString[6] = (uint8_t)((k_can_msg.ts_10us >> 24) & 0xFF);
cString[7] = (uint8_t)((k_can_msg.ts_10us >> 16) & 0xFF);

cString[8] = 0; // アライメント調整
cString[9] = (uint8_t)K_MSG_TYPE_CAN_DATA;

cString[10] = (uint8_t)((PLC_step_no >> 8) & 0xFF);
cString[11] = (uint8_t)((PLC_step_no >> 0) & 0xFF);

cString[12] = 0; // 予備
cString[13] = 0; // 予備
cString[14] = 0; // 予備
cString[15] = 0; // 予備

cString[16] = (uint8_t)((k_can_msg.error >> 8) & 0xFF);
cString[17] = (uint8_t)((k_can_msg.error >> 0) & 0xFF);

cString[18] = 0; // アライメント調整
cString[19] = k_can_msg.tx;

cString[20] = (uint8_t)((k_can_msg.can_id >> 8) & 0xFF);
cString[21] = (uint8_t)((k_can_msg.can_id >> 0) & 0xFF);
cString[22] = (uint8_t)((k_can_msg.can_id >> 24) & 0xFF);
cString[23] = (uint8_t)((k_can_msg.can_id >> 16) & 0xFF);

cString[24] = 0; // アライメント調整
cString[25] = (uint8_t)k_can_msg.dlc;

memcpy(&(cString[26]), k_can_msg.data, 64);

```

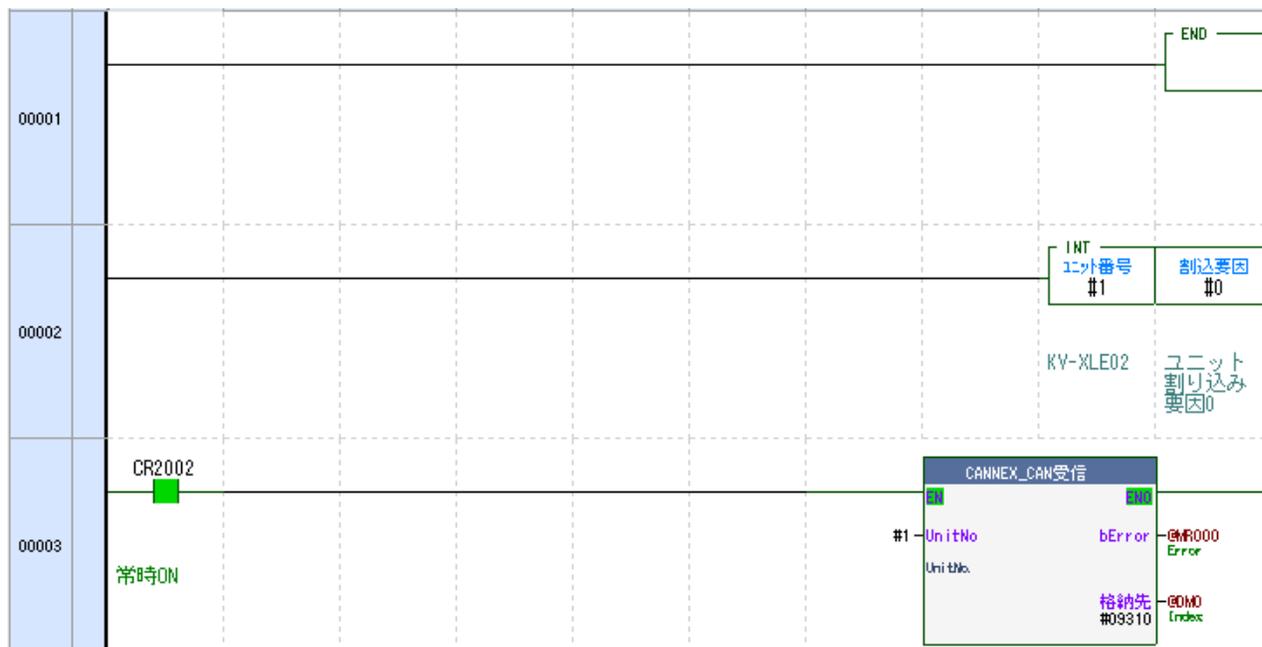
No.	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42							
間隔(秒)																																																	
送信データ(HEX)																																																	
送信データ(ASCII)																																																	
受信データ(HEX)	57	F7	00	00	5C	FA	00	22	00	00	00	00	00	00	00	00	00	00	00	0D	EF	00	AB	00	10	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30								
受信データ(ASCII)	W	NU	NU	¥	NU	~	NU	DL	!	~	#	\$	%	&	'	(	)	*	+	,	-	.	/	0																									

図 : Ethernet通信の例

Index : 0x000057F7 (22519)  
 TimeStamp : 0x00225CFA (22520.26ms)  
 MSG\_TYPE : 0x0000 (0)  
 StepNo. : 0x0000 (0)  
 Reserved : 0x00000000  
 CAN Error : 0x0000  
 送受信種別 : 0x0000 (0)  
 CANID : 0x00ABCDEF  
 DLC : 0x0010 (16) byte  
 受信Data : 0x21, 0x22, 0x23, 0x24... 0x30

※DLCに関わらず、受信Dataは64byte送られてきます。  
 DLCの値に応じて不要なデータは無視してください。  
 0x00以外の値が格納されている場合があります。  
 上記の例だとDLCは16byteなので、17byte以降のデータは無視してください。

KEYENCE KV-XLE02での受信は下図のように割り込みを用いて行います。

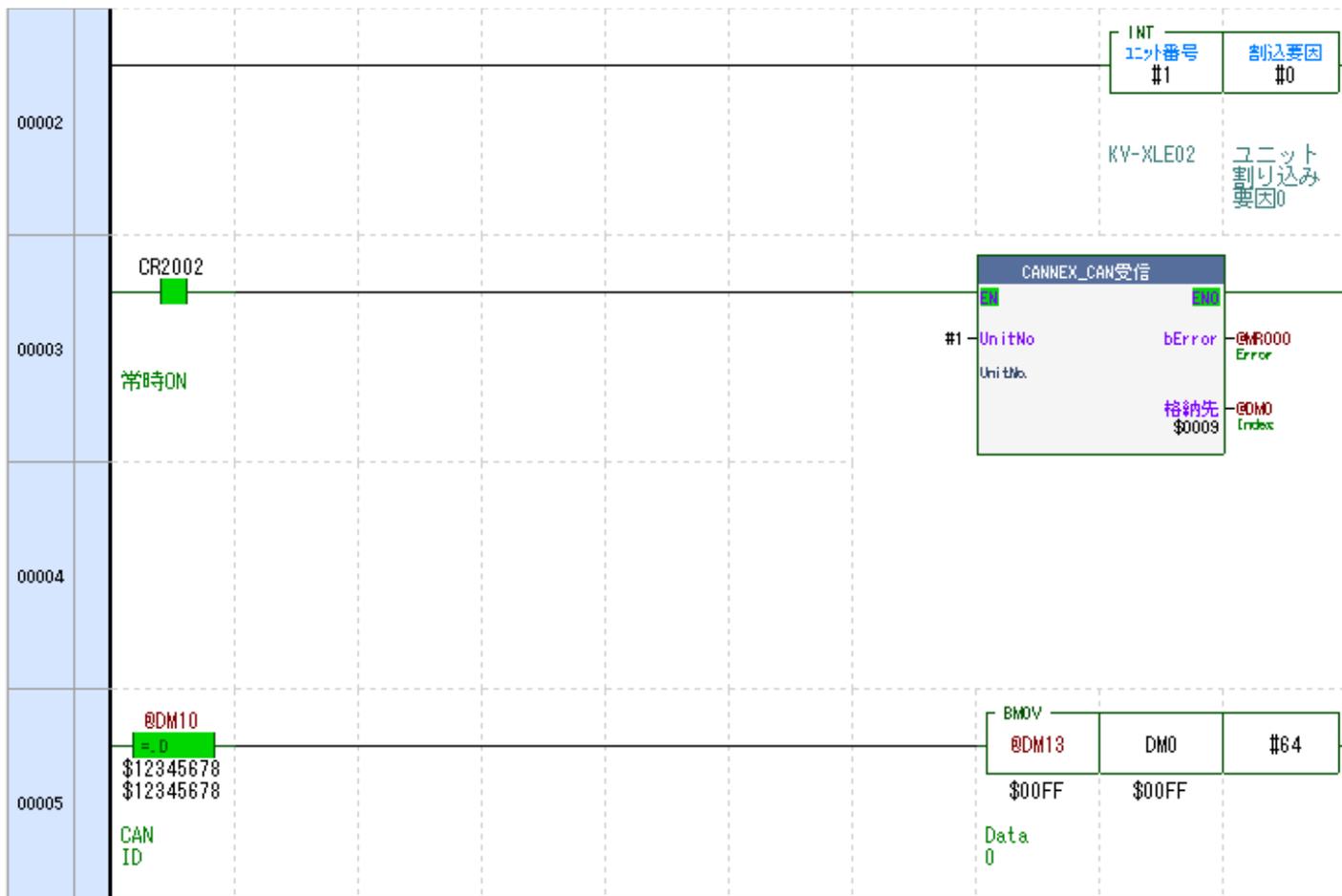


図：受信割り込みプログラム

格納先に@DM0を指定した場合、下図のように値が格納されます。（@DM0 - @DM76）

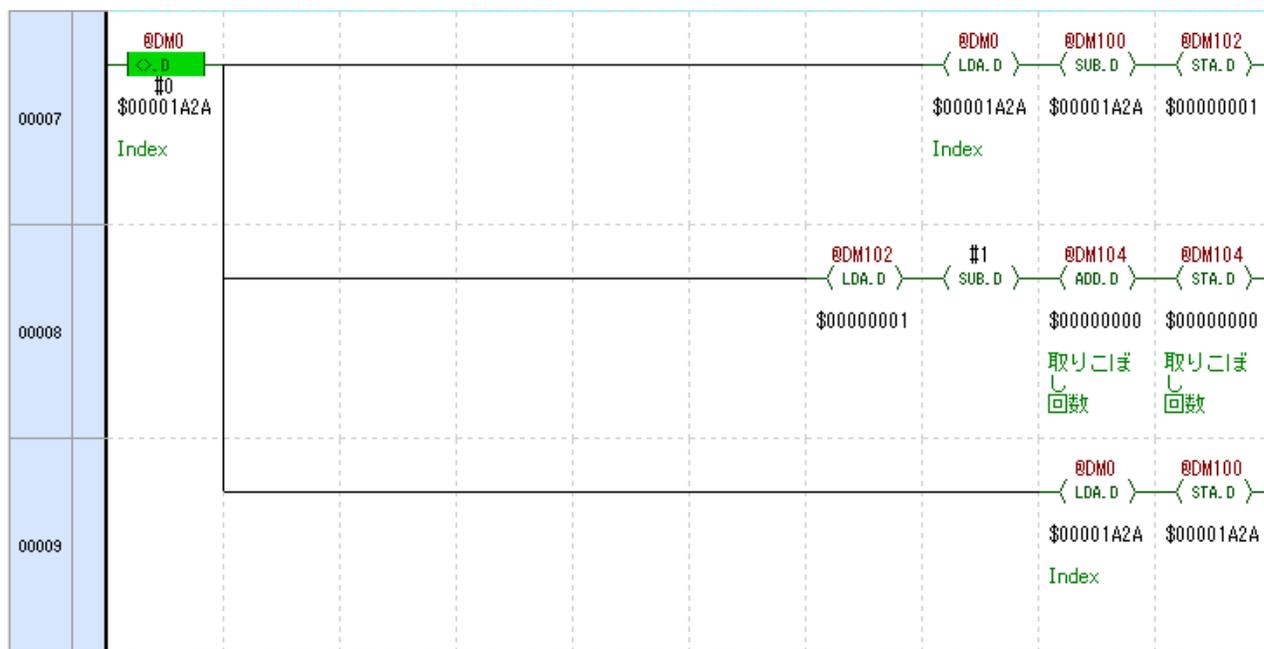
プログラム/ユニット	デバイス	参照先	現在値	表示形式	設定値	接点	コメント
CANNEX_受信	@DM0	-	8	10進数32BIT			Index
CANNEX_受信	@DM2	-	1097624	10進数32BIT			TimeStamp
CANNEX_受信	@DM4	-	0	10進数16BIT			MSG_TYPE
CANNEX_受信	@DM5	-	0	10進数16BIT			Step No.
CANNEX_受信	@DM6	-	0	10進数16BIT			Reserved
CANNEX_受信	@DM7	-	0	10進数16BIT			Reserved
CANNEX_受信	@DM8	-	\$0000	16進数16BIT			CAN Error
CANNEX_受信	@DM9	-	0	10進数16BIT			送受信 種別
CANNEX_受信	@DM10	-	\$00000064	16進数32BIT			CAN ID
CANNEX_受信	@DM12	-	64	10進数16BIT			DLC
CANNEX_受信	@DM13	-	\$0001	16進数16BIT			Data 0
CANNEX_受信	@DM14	-	\$0002	16進数16BIT			Data 1
CANNEX_受信	@DM15	-	\$0003	16進数16BIT			Data 2
CANNEX_受信	@DM16	-	\$0004	16進数16BIT			Data 3
CANNEX_受信	@DM17	-	\$0005	16進数16BIT			Data 4
CANNEX_受信	@DM18	-	\$0006	16進数16BIT			Data 5
CANNEX_受信	@DM19	-	\$0007	16進数16BIT			Data 6
CANNEX_受信	@DM20	-	\$0008	16進数16BIT			Data 7
CANNEX_受信	@DM21	-	\$0009	16進数16BIT			Data 8
CANNEX_受信	@DM22	-	\$0010	16進数16BIT			Data 9
CANNEX_受信	@DM23	-	\$0011	16進数16BIT			Data 10
CANNEX_受信	@DM24	-	\$0012	16進数16BIT			Data 11
CANNEX_受信	@DM25	-	\$0013	16進数16BIT			Data 12
CANNEX_受信	@DM26	-	\$0014	16進数16BIT			Data 13
CANNEX_受信	@DM27	-	\$0015	16進数16BIT			Data 14
CANNEX_受信	@DM28	-	\$0016	16進数16BIT			Data 15
CANNEX_受信	@DM29	-	\$0017	16進数16BIT			Data 16
CANNEX_受信	@DM30	-	\$0018	16進数16BIT			Data 17
CANNEX_受信	@DM31	-	\$0019	16進数16BIT			Data 18
CANNEX_受信	@DM32	-	\$0020	16進数16BIT			Data 19
CANNEX_受信	@DM33	-	\$0021	16進数16BIT			Data 20
CANNEX_受信	@DM34	-	\$0022	16進数16BIT			Data 21
CANNEX_受信	@DM35	-	\$0023	16進数16BIT			Data 22
CANNEX_受信	@DM36	-	\$0024	16進数16BIT			Data 23
CANNEX_受信	@DM37	-	\$0025	16進数16BIT			Data 24
CANNEX_受信	@DM38	-	\$0026	16進数16BIT			Data 25
CANNEX_受信	@DM39	-	\$0027	16進数16BIT			Data 26
CANNEX_受信	@DM40	-	\$0028	16進数16BIT			Data 27
CANNEX_受信	@DM41	-	\$0029	16進数16BIT			Data 28
CANNEX_受信	@DM42	-	\$0030	16進数16BIT			Data 29
CANNEX_受信	@DM43	-	\$0031	16進数16BIT			Data 30
CANNEX_受信	@DM44	-	\$0032	16進数16BIT			Data 31
CANNEX_受信	@DM45	-	\$0033	16進数16BIT			Data 32
CANNEX_受信	@DM46	-	\$0034	16進数16BIT			Data 33
CANNEX_受信	@DM47	-	\$0035	16進数16BIT			Data 34
CANNEX_受信	@DM48	-	\$0036	16進数16BIT			Data 35
CANNEX_受信	@DM49	-	\$0037	16進数16BIT			Data 36
CANNEX_受信	@DM50	-	\$0038	16進数16BIT			Data 37
CANNEX_受信	@DM51	-	\$0039	16進数16BIT			Data 38
CANNEX_受信	@DM52	-	\$0040	16進数16BIT			Data 39
CANNEX_受信	@DM53	-	\$0041	16進数16BIT			Data 40
CANNEX_受信	@DM54	-	\$0042	16進数16BIT			Data 41
CANNEX_受信	@DM55	-	\$0043	16進数16BIT			Data 42

下図のように記述すると、ID = 0x12345678のCANデータをDM0-DM63に格納することができます



割り込みでの処理のため取りこぼしが発生しません。

それでも取りこぼしの確認が必要な場合は、@DM0(Index)が1ずつインクリメントされていることを確認するようなプログラムを作成して確認してください。



図：取りこぼし確認プログラム

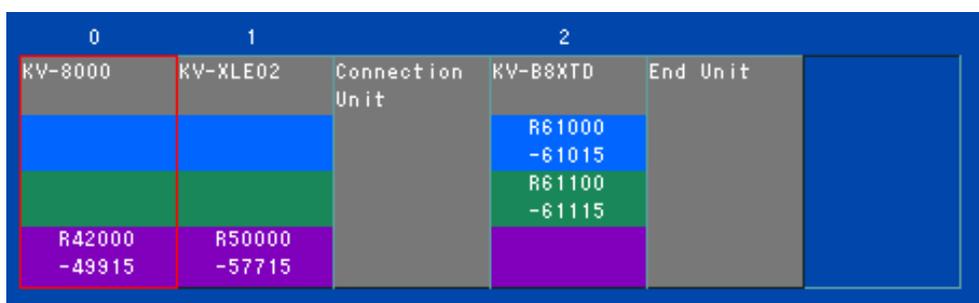
# KV-XLE02での使用方法

## ハードウェア構成

### ● PLCユニット構成

- KV-8000 (KV-7000/8000シリーズならKV-8000でなくても可)
- KV-XLE02
- KV-B8XTD (入力4点あれば他ユニットでも可)

※本マニュアルはKV-8000, KV-XLE02, KV-B8XTDの構成で説明します。



### ● ネットワーク構成

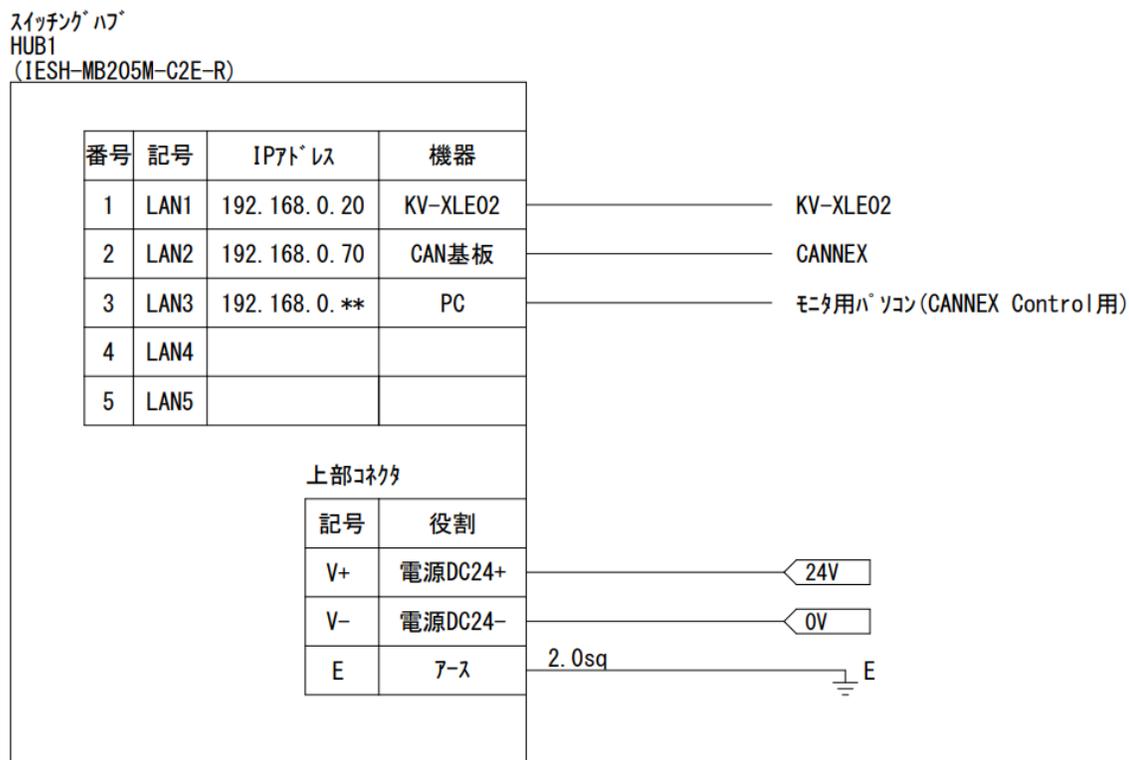


図 : Ethernet 接続図

# KV STUDIO設定

1. ユニットエディタでKV-XLE02を下記のように設定します

ポート1	
IPアドレス設定方法	固定IPアドレス(※)
IPアドレス	192.168.0.20
サブネットマスク	255.255.255.0
通信速度	1000/100/10Mbps自動(※)
産業用イーサネット	使用しない(※)
PLCリンク	使用しない(※)
PROTOCOL STUDIO	使用する
FTPサーバ	使用しない(※)
Modbusサーバ	使用しない(※)
上位リンク通信	有効(※)
MCプロトコル/SLMP通信	有効(※)
KVS, KV COM+接続	有効(※)
VT接続	有効(※)
KVS経由通信	有効(※)

図 : KV-XLE02 ユニットエディタ設定

2. PROTOCOL STUDIOに弊社提供の設定ファイルをインポートします

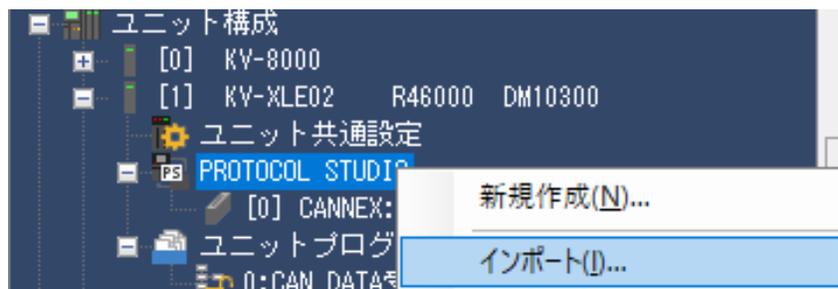


図 : KV-XLE02 PROTOCOL STUDIO設定インポート

※設定ファイルが必要な方は、お問い合わせください

3. PROTOCOL STUDIOの接続機器設定を行います

CANNEXのIPアドレスと、ポート番号:20000(固定)を設定します

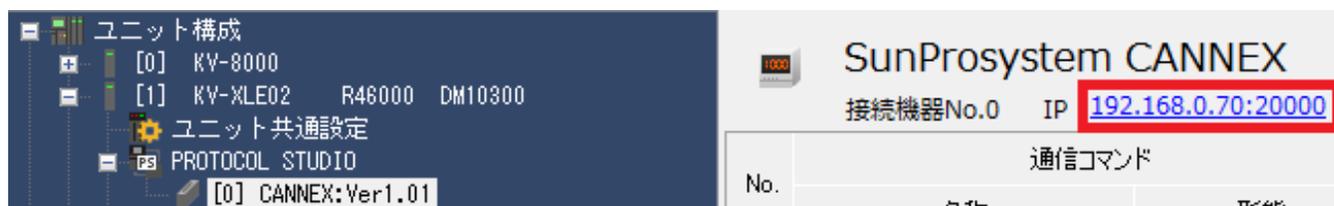


図 : KV-XLE02 PROTOCOL STUDIO接続機器 設定

#### 4. KV-XLE02のユニット割り込み要因0を"使用する"に設定します

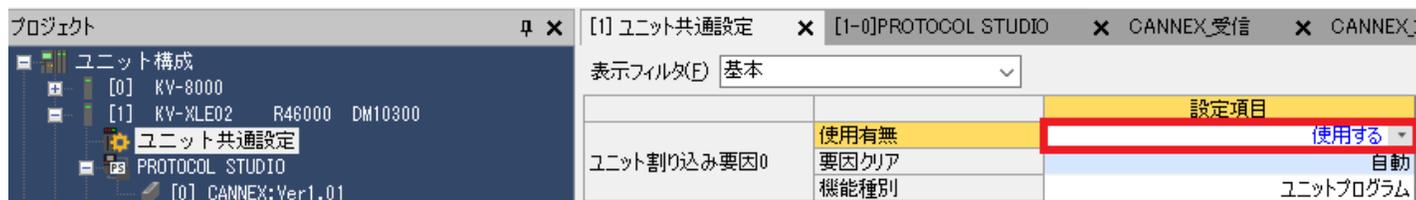
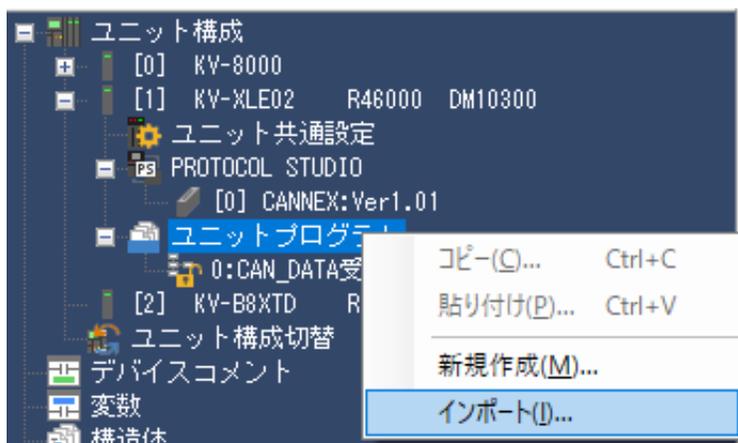
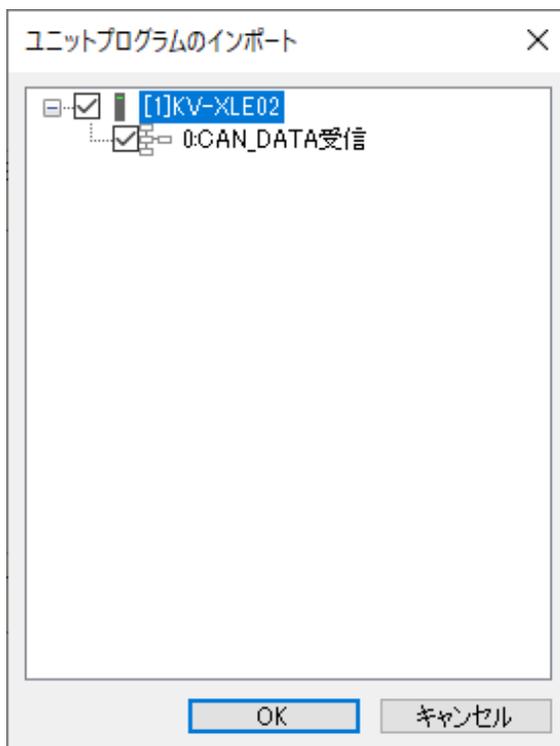


図 : KV-XLE02 ユニット割り込み要因0設定

#### 5. KV-XLE02のユニットプログラムをインポートします



※下記のKV-Studioサンプルプロジェクトからインポートしてください  
[ダウンロードリンク](#)



ここまでの手順で作成したプロジェクトファイルを下記からダウンロードできます。

[ダウンロードリンク](#)

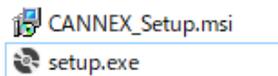
## パラレル出力

番号	名称	内容
OUT1	CANNEX Run	CANNEX動作中 電源を供給し、マイコンプログラムが動作するとONします。 OFFしたままになる場合、基板故障の可能性があります
OUT2	ネットワーク接続済	ネットワークに接続されています。 PLCやPCとの通信が確立を意味するものではありません LANケーブルがハブなどに接続されるとONします
OUT3	CAN Run	RUNコマンドを受け付けるとONします。 STOPコマンドでOFFします
OUT4	CANNEX 高負荷	CANNEXから外部へのEthernet/パケットの送信処理量を CANデータ受信量が上回っている状態です。 この出力がONする場合、サンプロシステムにご相談ください。 CANの仕様がCANNEXの処理能力を超えている可能性があります

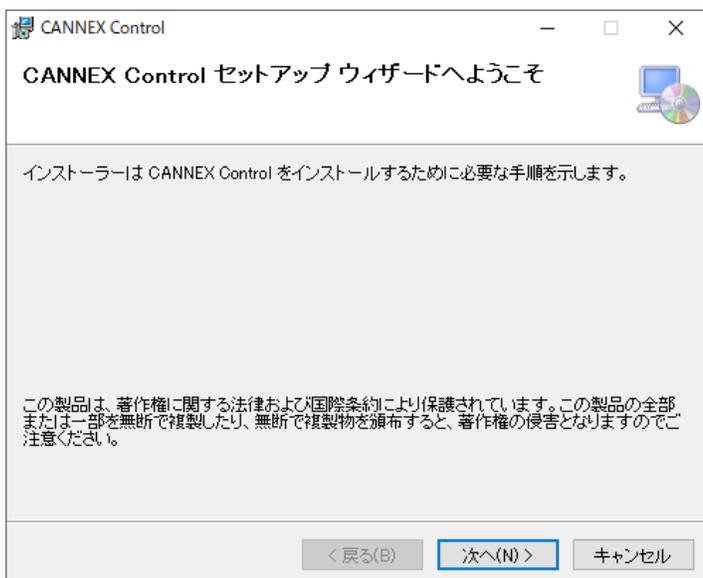
# CANNEX Controlインストール手順

CANNEX Controlのセットアップファイルをダウンロードします。

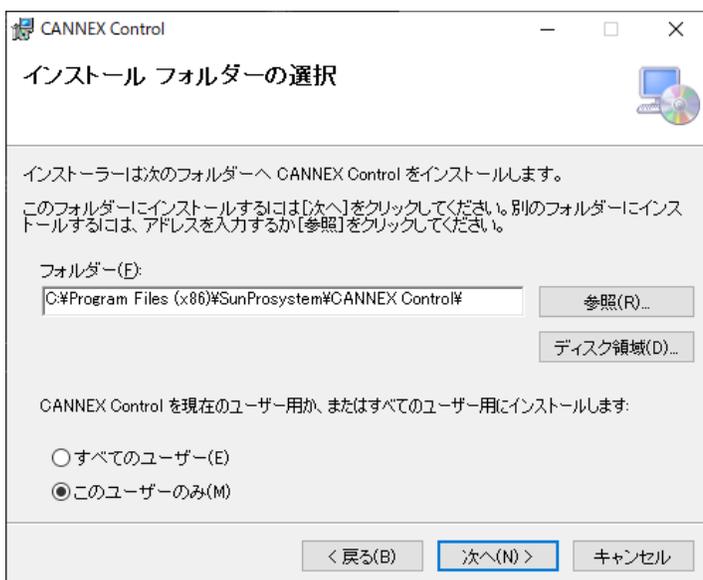
zipファイルを解凍すると下記の2つのファイルがあります。



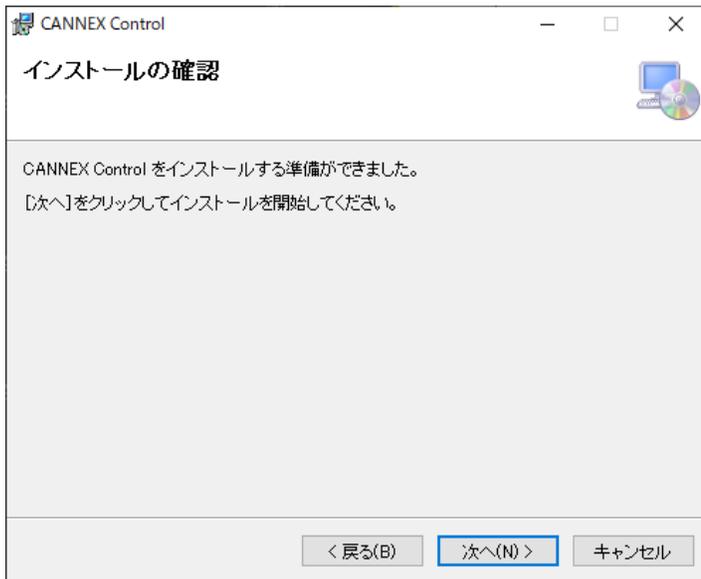
「setup.exe」をダブルクリックします。



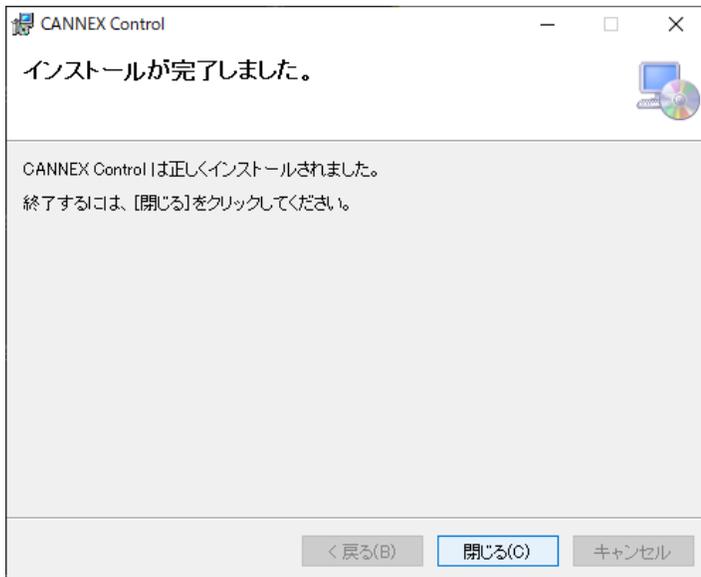
「次へ」をクリック



「次へ」をクリック



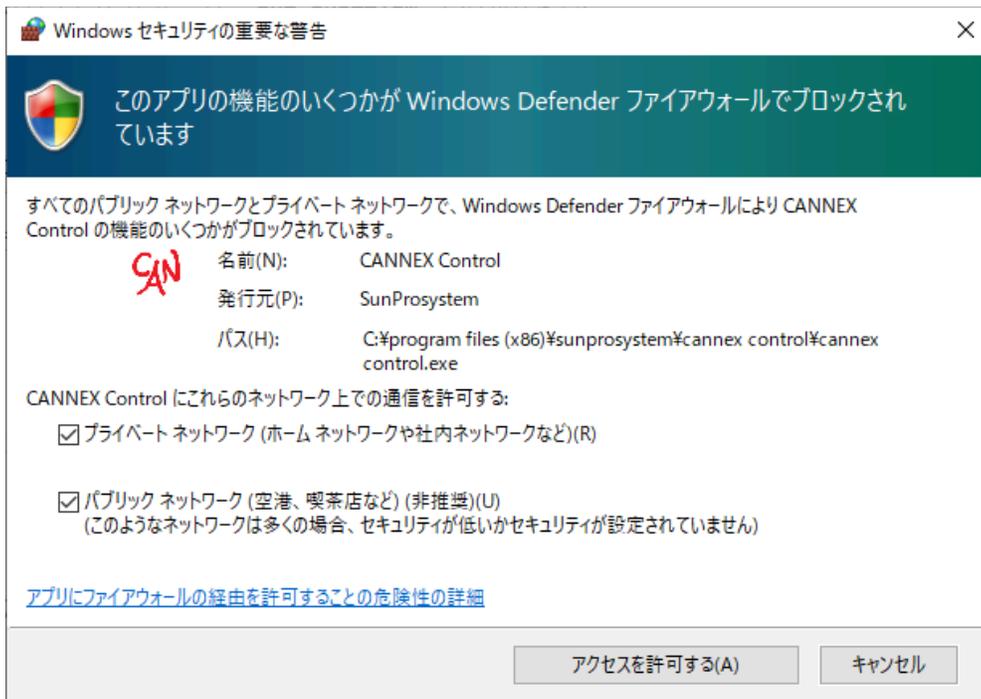
「次へ」をクリック



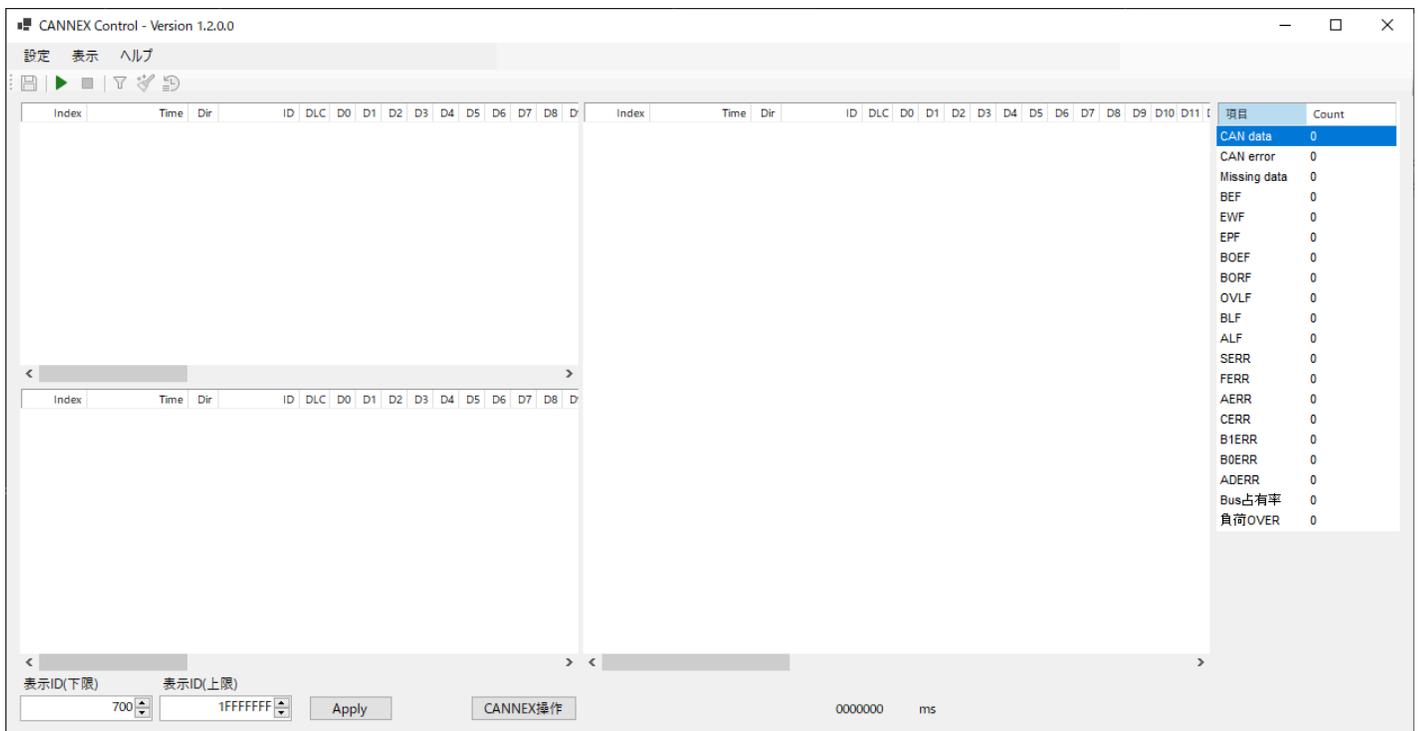
「閉じる」をクリック



デスクトップに生成されるショートカットをクリックしてアプリケーションを起動



「プライベートネットワーク」「パブリックネットワーク」の両方にチェックを入れて、「アクセスを許可する」



CANNEX Controlが起動します。

## モニタツール 通信設定

通信設定

	変更前	変更後
CANNEXのIPアドレス	192.168.0.70	
PLCのIPアドレス	192.168.0.20	
PLCのポート番号	10000	
CANNEXのMACアドレス	02:0c:7b:aa:46:04	
Gatewayアドレス	0.0.0.0	

検索

変更実施

前 次

モニタIPアドレス 192.168.0.70

CANNEXが見つかりました : 1枚

-----

CANNEX IPAddress : 192.168.0.70  
CANEX MAC address : 02:0c:7b:aa:46:04  
PLC IPAddress : 192.168.0.20:10000  
Gateway IPAddress : 0.0.0.0  
問い合わせたPCのIPAddress : 192.168.0.98

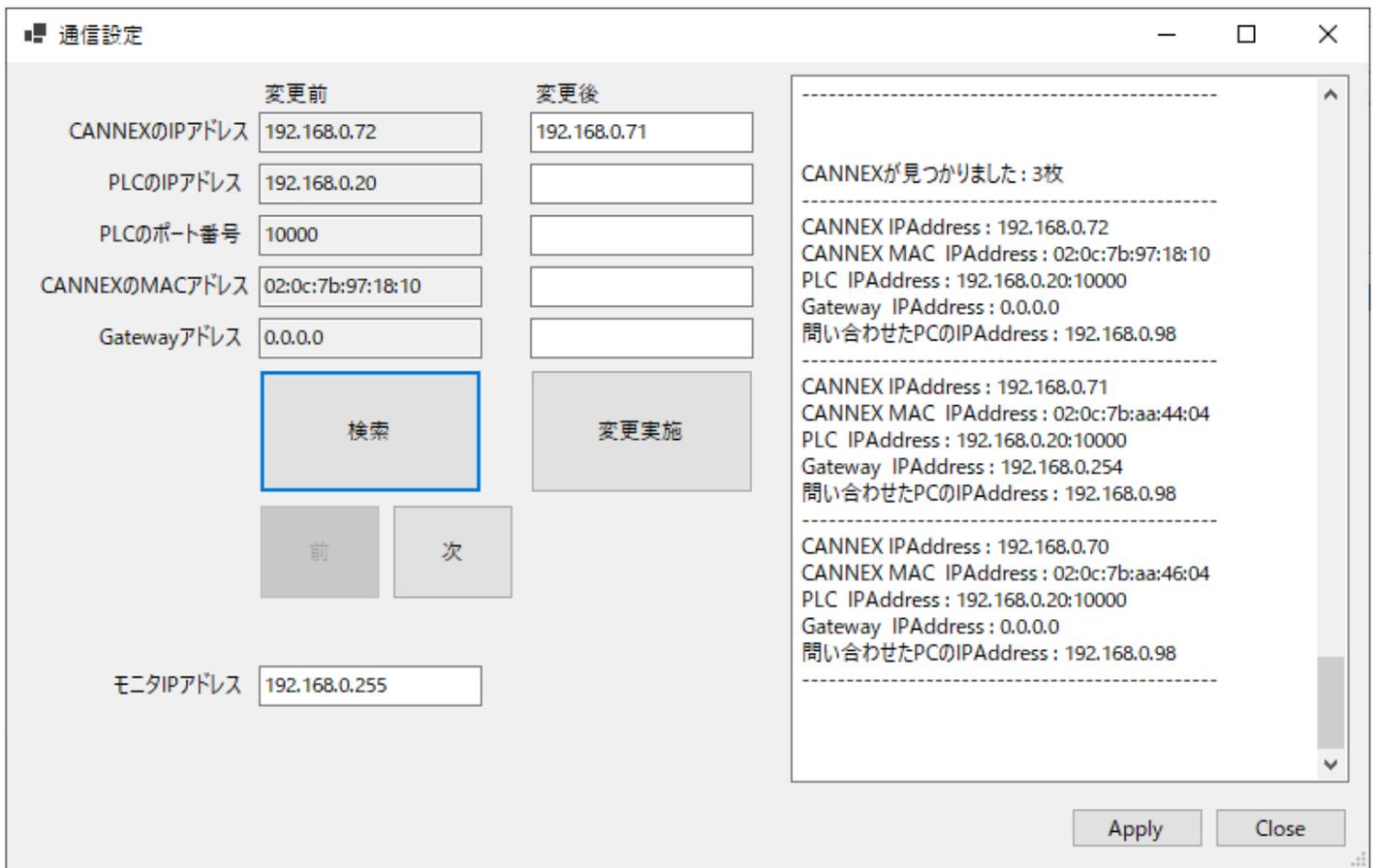
-----

Apply Close

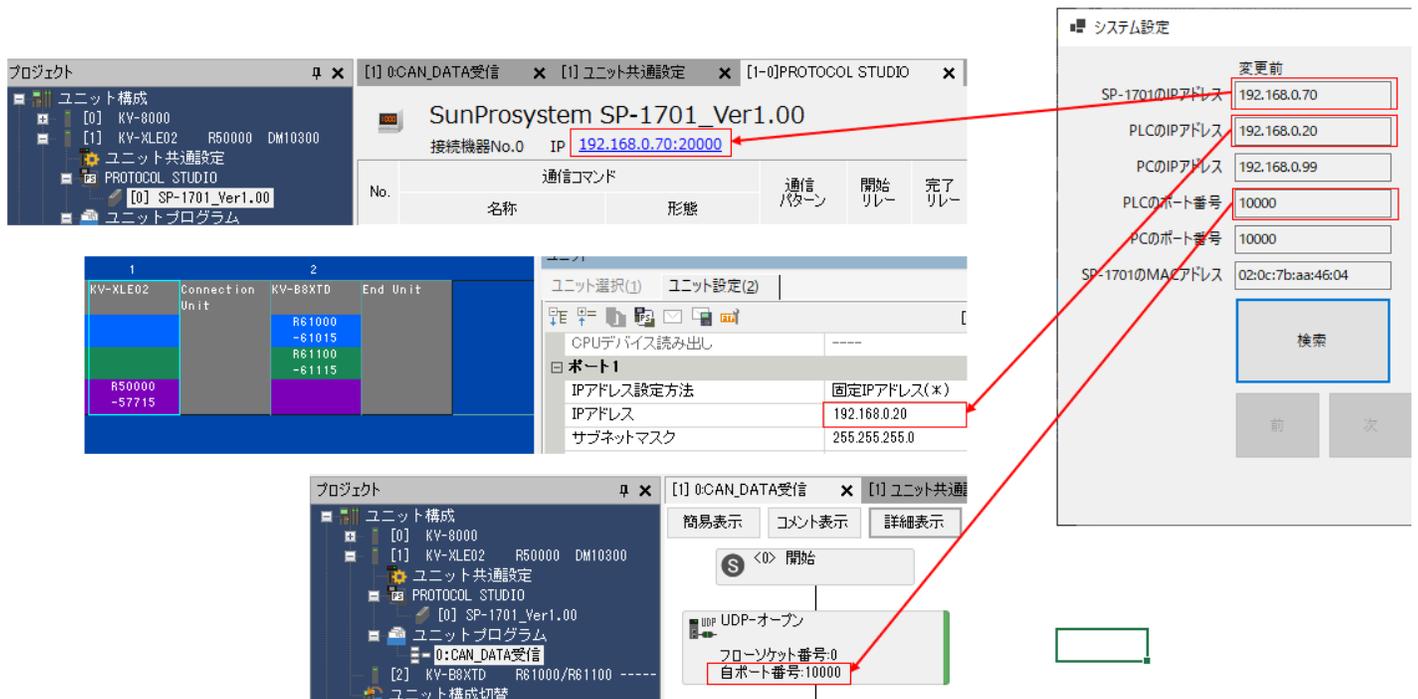
図 : CANEX モニタツール 通信設定画面

- 「検索」ボタンを押すと、「モニタIPアドレス」に設定したIPアドレスのCANEXを検索します。（サブネットマスクは255.255.255.0です）
- 「変更実施」ボタンを押すと、「変更後」のテキストボックスに入力した値がCANEXに書き込まれます。書き込み後はCANEXが自動で再起動します。
- 「Apply」ボタンを押すと、「モニタIPアドレス」の所に表示されているCANEXと通信できるようにモニタツールの通信設定が変更されます

モニタIPアドレスを「192.168.0.255」といった具合にブロードキャストアドレスを設定して検索すると、ネットワーク内に接続されているすべてのCANEXを見つけることができます。複数枚見つかった場合は、「前」「次」ボタンで表示するCANEXを切り替えます。



KV-STUDIOの設定とCANNEXの設定は下記のように対応してします



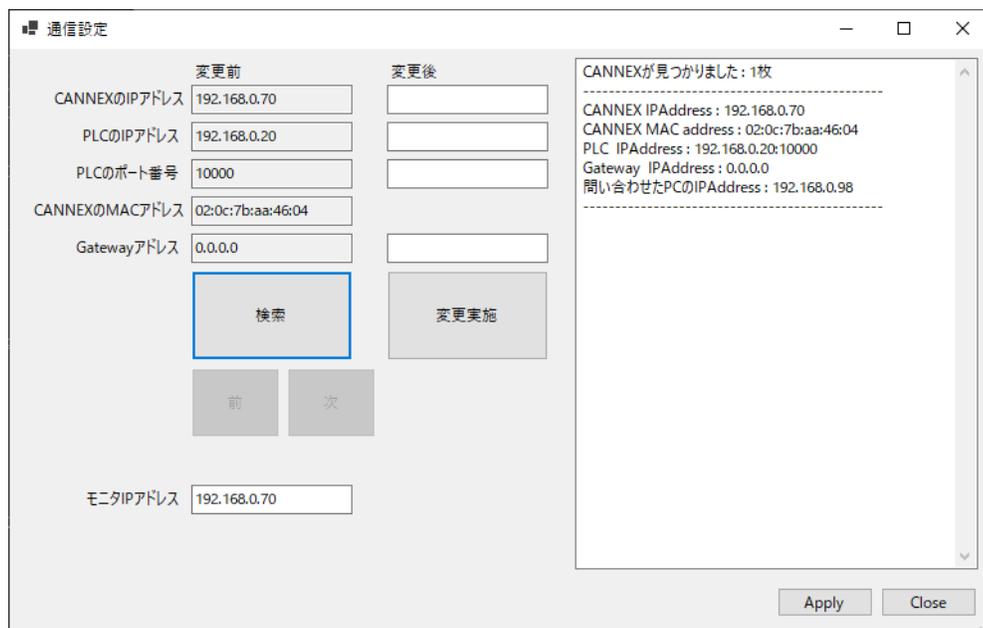
図：KV-STUDIO設定 と モニタツール 通信設定の対応

# IPアドレス変更手順

1. パソコンのIPアドレスをCANNEXと同一ネットワークアドレスに設定してください  
(CANNEXのIPアドレスが192.168.5.70なら、  
パソコンのIPアドレスを192.168.5.\*にする。\*は他の機器と重複しないアドレス)
2. CANNEXとパソコンをLANケーブルで接続。
3. CANNEX Controlアプリケーションを起動し、  
「設定」->「通信設定」



4. 「モニタIPアドレス」をCANNEXの現在のIPアドレスに変更
5. 「検索」ボタンを押すとCANNEXが見つかる



## ※見つからない場合

- パソコンのIPアドレスを192.168.0.\*の3オクテット目(0の部分)を0-9に変更して検索
- サブネットマスクを255.255.255.0にする
- モニタIPアドレスを192.168.0.255にして検索を実施する
- CANNEXとパソコンを1対1で接続する (他の機器を外す)

6. 変更後のIPアドレスを入力して、「変更実施」ボタンを押す

	変更前	変更後
CANNEXのIPアドレス	192.168.0.70	192.168.0.123
PLCのIPアドレス	192.168.0.20	
PLCのポート番号	10000	
CANNEXのMACアドレス	02:0c:7b:aa:46:04	
Gatewayアドレス	0.0.0.0	
モニタIPアドレス	192.168.0.70	

検索

変更実施

前 次

Apply Close

CANNEXが見つかりました: 1枚

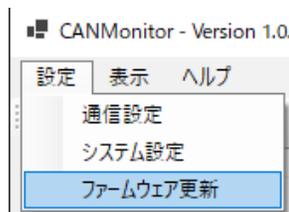
CANNEX IPaddress : 192.168.0.70  
CANNEX MAC address : 02:0c:7b:aa:46:04  
PLC IPAddress : 192.168.0.20:10000  
Gateway IPAddress : 0.0.0.0  
問い合わせたPCのIPAddress : 192.168.0.98

CANNEXのIPAddressを 192.168.0.123 に変更しました

7. CANNEXに変更内容が書き込まれ、自動でソフトウェアリセットがかかります  
(電源再投入と同じ状態)
8. 「モニタIPアドレス」を変更したIPアドレスに設定して「Apply」を押します。

# ファームウェア更新手順

1. CANNEXとパソコンをLANケーブルで接続。
2. CANNEX Controlアプリケーションを起動し、  
「設定」->「ファームウェア更新」



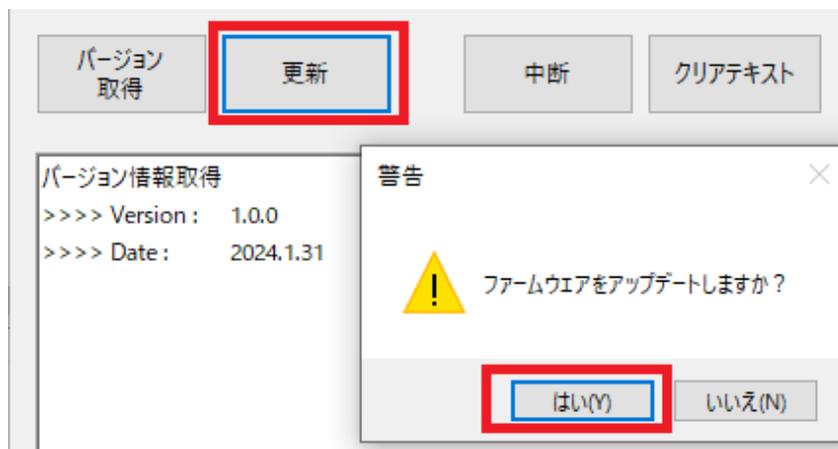
4. 「バージョン取得」ボタンを押して、現在のバージョンを確認  
(表示されない場合は、「設定」->「通信設定」を確認してください)



5. 「選択」ボタンを押して、更新するファームウェアを選択



6. 「更新」ボタンを押して、「はい」を選択



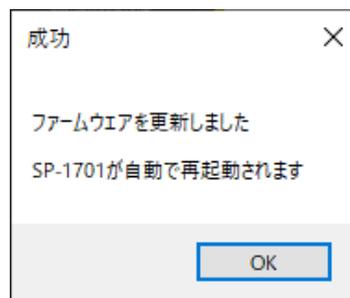
7. ファームウェアの書き込みが開始されます。

```
書き込みファイルサイズ: 86600bytes
ブランクチェック
>>>> Blank check
>>>> Secondary slot : Not blank
>>>> Erase Secondary slot
>>>> Success
>>>> Blank check
>>>> Success
>>>> Secondary slot erased

コードフラッシュ書き込み開始
書き込み開始アドレス: 0x00200000
書き込み終了
```

86600 / 86600 bytes

8. 正常に終了すると下記のダイアログが表示されます。



9. 「OK」を押したあと「バージョン取得」を押すと書き込まれたファームウェアのバージョンが表示されます。  
(CANNEXの再起動が完了するまで応答は返ってきません)



※元のファームウェアに戻す場合は、古いファームウェアを書き込んでください

## Q & A

- 電源投入時はSTOP状態ですか？ 電源切で送信設定は消えるのですか？

電源投入時はSTOP状態です。電源切りで送信設定（送信MBX）は消えます（IPアドレス、MACアドレスは記憶されますが、CANに関する設定は消えます）

- TSTOP中にSTTXコマンドで該当する送信MBXの内容を変更すると送信を再開するのですか？

再開はせず、内容のみ書き換わって、TACTVで再開します

例

1. CANNEXの電源起動
2. STTX→RUN：定期送信開始
3. TSTOP：定期送信停止
4. STTXで上書き：定期送信は停止したまま
5. TACTV：定期送信再開
6. TSTOP：定期送信停止
7. STOP→RUN：定期送信は停止したまま
8. STOP→TACTV→RUN：定期送信開始

送信MBXごとにstopフラグを持っています

初期値：stop = false

TSTOPコマンドでのみ stop = trueとなります

stop = falseに戻すのはTACTVと電源再起動のみ

- LOGコマンドについて、コマンド送信時からログを開始するのではなく、コマンド発行時にバッファに溜まっているデータをはき出すのですね？

はい。常時CSV出力するのではなく、コマンド受信時にバッファに溜まっているデータをまとめてCSV出力します

- CANFD ボーレート5Mbpsで、IDが全部で15種類、データの周期はすべて10ms、データサイズは64byteです。CANNEXは使用できますか？

はい。実際にCANNEXで通信を行い、CANNEX Controlで記録したログ(CSVファイル)が下記です。  
CANの受信処理よりもEthernetの送出处理の方が早いためCANNEX側では取りこぼしなく適切に処理を行うことができます。  
PLC側でも受信割り込みで取りこぼしなくデータを取得できました。  
スキャンプログラムで本データをどのように扱うかはユーザー次第となります。

[can\\_log\\_240417\\_110257.csv](#)

- CANFD ボーレート5Mbpsで、IDが全部で4種類、データの周期はすべて1ms、データサイズは64byteです。CANNEXは使用できますか？

Ethernetの送出处理よりも、CANの受信処理のほうが早いため  
CANNEXの処理能力を超えてしまい、データの取りこぼしが発生してしまいます。

- CANNEXの処理能力を超えているかどうかを判断するにはどうすればよいですか？

弊社にCANの仕様をお伝えいただくか、  
デモ機 + CANNEX Controlを使用して確認することができます。

下記のようにCANNEX ControlにてCANバスにデータを送出する設定を行います。

MBX	Sweep	CS_CNT_TYPE	CANID	周期	DLC
0	0	0	111	5	64
1	0	0	222	5	64
2	0	0	333	5	64
3	0	0	444	5	64
4	0	0	555	5	64
5	0	0	666	5	64
6	0	0	777	5	64
7	0	0	888	5	64
8	0	0	999	5	64
9	0	0	AAA	5	64
10	0	0	BBB	10	64
11	0	0	CCC	10	64
12	0	0	DDD	10	64
13	0	0	EEE	10	64
14	0	0	FFF	10	64

実際にCANバスにデータを送出し、「負荷OVER」の値が0であることを確認します。

Index	Time	Dir	ID	DLC	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	項目	Count
367920	147178.23	T	444	64	00	00	00	00	00	00	00	00	00	00	00	00	CAN data	367951
367921	147178.45	T	999	64	00	00	00	00	00	00	00	00	00	00	00	00	CAN error	0
367922	147179.99	T	555	64	00	00	00	00	00	00	00	00	00	00	00	00	Missing data	0
367923	147180.20	T	AAA	64	00	00	00	00	00	00	00	00	00	00	00	00	BEF	0
367924	147180.37	T	111	64	01	02	03	04	05	06	07	08	09	0A	0B	0C	EWf	0
367925	147180.57	T	666	64	00	00	00	00	00	00	00	00	00	00	00	00	EPF	0
367926	147180.78	T	BBB	64	00	00	00	00	00	00	00	00	00	00	00	00	BOEF	0
367927	147181.25	T	CCC	64	00	00	00	00	00	00	00	00	00	00	00	00	BORF	0
367928	147181.44	T	222	64	FF	FF	OVLf	0										
367929	147181.63	T	777	64	00	00	00	00	00	00	00	00	00	00	00	00	BLf	0
367930	147182.25	T	DDD	64	00	00	00	00	00	00	00	00	00	00	00	00	ALf	0
367931	147182.44	T	333	64	00	00	00	00	00	00	00	00	00	00	00	00	SERR	0
367932	147182.65	T	888	64	00	00	00	00	00	00	00	00	00	00	00	00	FERR	0
367933	147183.25	T	EEE	64	00	00	00	00	00	00	00	00	00	00	00	00	AERR	0
367934	147183.45	T	444	64	00	00	00	00	00	00	00	00	00	00	00	00	CERR	0
367935	147183.65	T	999	64	00	00	00	00	00	00	00	00	00	00	00	00	B1ERR	0
367936	147184.25	T	FFF	64	00	00	00	00	00	00	00	00	00	00	00	00	B0ERR	0
367937	147184.45	T	555	64	00	00	00	00	00	00	00	00	00	00	00	00	ADERR	0
367938	147184.66	T	AAA	64	00	00	00	00	00	00	00	00	00	00	00	00	Bus占有率	50.05%
367939	147185.21	T	111	64	01	02	03	04	05	06	07	08	09	0A	0B	0C	負荷OVER	0
367940	147185.40	T	666	64	00	00	00	00	00	00	00	00	00	00	00	00		
367941	147186.23	T	222	64	FF	FF												
367942	147186.42	T	777	64	00	00	00	00	00	00	00	00	00	00	00	00		
367943	147187.23	T	333	64	00	00	00	00	00	00	00	00	00	00	00	00		
367944	147187.44	T	888	64	00	00	00	00	00	00	00	00	00	00	00	00		
367945	147188.23	T	444	64	00	00	00	00	00	00	00	00	00	00	00	00		
367946	147188.44	T	999	64	00	00	00	00	00	00	00	00	00	00	00	00		
367947	147189.23	T	555	64	00	00	00	00	00	00	00	00	00	00	00	00		
367948	147189.44	T	AAA	64	00	00	00	00	00	00	00	00	00	00	00	00		
367949	147190.25	T	BBB	64	00	00	00	00	00	00	00	00	00	00	00	00		
367950	147190.42	T	111	64	01	02	03	04	05	06	07	08	09	0A	0B	0C		
367951	147190.63	T	666	64	00	00	00	00	00	00	00	00	00	00	00	00		

負荷OVERの値が増加する場合、CANNEXの処理能力を超えています。

- OMRONや三菱製のシーケンサでも使用できますか？

メーカー問わずEthernetのUDP通信の送受信ができる機器であれば使用可能です。  
ただし、CANデータの発生頻度、Ethernet通信ユニットやCPUの処理能力次第でデータを取りこぼします。

- 定期送信で何種類のIDを送れますか？

最大30種類送信できます。それ以上必要な場合はご相談ください。  
カスタム仕様にて対応できます。

- KV-XLE02のPort2を使用したい。どうすればよいですか？

Port2に対応したKV-Studioのプロジェクトをメールにて送付しますので  
ご相談ください。（無償です）

- KV-XLE02のPort1で3つのCANNEXを制御したい。どうすればよいですか？

KV-Studioのプロジェクトをメールにて送付しますので  
ご相談ください。（無償です）