CX2500 Series CODESYS Edition Sample Program Manual

文書番号 CCOT-25-027			
発行年月日 2025年6月27日			
担当 審査 承認			
九山	藤本	餐 本	

東京計器株式会社 油圧制御システムカンパニー 電子機器部 技術課



目次

1. はじめに	3
2. 使用前に	3
2.1. 安全上の注意	3
2.2. ソフトウェア使用許諾	5
3. ファイル構成・ソースファイル一覧	6
4. サンプルプログラムのインポート	8
5. プログラム概要	12
5.1. 概要	12
6. 各機能のプログラム	13
6.1. アナログ入力、周波数入力、デジタル入力	13
6.2. カウンタ入力制御(CNTIN_CTRL)	14
6.3. デジタル出力制御(DO_CTRL)	15
6.3.1. ビジュアライゼーション	17
6.4. PWM 出力制御(PWM_CTRL)	18
6.5. CAN 通信制御(CAN_CTRL)	19
6.6. RS232C 通信制御(RS232C_CTRL)	22
6.7. リアルタイムクロック制御(RTC_CTRL)	24
6.8. リテイン変数制御(PLC_PRG)	25
6.9. タイマカウンタ制御(TIMERCNT_CTRL)	26
6.10. CAN フィールドバス—EXM シリーズ制御(EXM1_CTRL、EXM2_CTRL)	27
7. 改訂履歷	30



1. <u>はじめに</u>

本書は、汎用コントローラ CX2500 シリーズ(CODESYS 版)を用いて作成したサンプルプログラム (Cx2500TestProject_Codesys_v***、***はバージョンを表す数字)の簡易仕様を示した文書です。CODESYS 版の詳細仕様については別文書「CCOT-24-016 CX2500 Series CODESYS Edition User Manual(For Programming)」を参照して下さい。

2. 使用前に

2.1. 安全上の注意

本書およびサンプルプログラム使用等における安全に関する注意事項を以下に示します。本書に記載した注意 事項は安全上重大な内容を記載していますので、必ず順守して下さい。

▲ 警告		この表示を無視して誤った取扱いをすると、人が死亡、又は重傷を負う可能性が想定される内
45	言口	容が記載されています。
↑ 沙音		この表示を無視して誤った取扱いをすると、人が傷害を負ったり、物的損害の発生が想定される
<u> </u>	任思	内容やその他留意すべき内容が記載されています。

0	禁止	この表示は <mark>実施してはいけない</mark> 内容が記載されています。
•	必須	この表示は本製品を使用する上で <u>必ず実施する必要がある</u> 内容が記載されています。







必須

デバッグ中におこなう動作開始/停止や値の強制、並びに出力の強制等は、安全に動作するこ とを操作前の確認で必ずおこなった上実施して下さい。確認不十分の場合、操作により事故に つながる可能性があります。



注意



弊社が配布するデバイス定義・機能ドライバファイル及びランタイムファイルを編集し使わないで 下さい。CX2500 並びに CODESYS の機能が損なわれたり、誤動作につながる可能性がありま



本製品に関する物品の分解や改造等はおこなわないで下さい。破損や誤動作、事故につなが る可能性があります。



禁止





必須

本製品の使用に際し、万が一本製品に故障・不具合が発生した場合でもフェールセーフなどの 保全機能を機器外部などにユーザー側で必ず持たせて下さい。



注意

本製品の仕様内で必ず使用して下さい。製品仕様外での使用は事故につながる可能性があり ますので絶対におこなわないでください。



禁止



注意

通電中に端子に触れたり、配線変更等をしないでください。製品の破損や感電などの事故につ ながる可能性があります。



禁止



注意

CX2500 や CODESYS 等のソフトウェアアップデートにより、製品ソフトの表示・表記が実際のも のと本書記載のものが一部異なる場合があります。その際は、実際の表示・表記内容に従って 下さい。



2.2. ソフトウェア使用許諾

本書および CX2500 の CODESYS 版サンプルプログラムの使用・参照等をおこなった場合、安全上の注意ならびに下記項目について同意したものとみなします。

【厳守項目】

- 本書を熟読して下さい。
 - ▶ 本書は本製品を正しく使うための文書のため、必ず最後まで熟読の上で製品を使用して下さい。
- 本書を大切に保管して下さい。
 - ➤ CX2500 を取り扱う場合、本書は重要であるため、いつでも参照できるように身近に且つ大切に保管して下さい。
- 本書を取扱者(ユーザー)の手元に届けて下さい。
 - ➤ 代理店など、CX2500 の仲介になる方は、必ず本書(URL 情報を含む)を実際に取り扱う方の手元に届けて下さい。
- 本書を紛失した場合は直ぐに補充して下さい。
 - ▶ 万が一、本書を紛失した場合は、弊社営業所または購入先までご連絡をお願い致します。
- 本書を断りなく転載・編集、配布や複製等をすることを固く禁じます。

【注意項目】

- 本書およびサンプルプログラムの仕様は予告なく変更する場合があります。
- 本書に記載されている CODESYS の画面表示等はバージョンアップによって一部異なる場合があります。 その際は、実際の画面表示で確認下さい。
- CODESYS は 3S-Smart Software Solutions GmbH の登録商標です。
- 上記以外で、本書に記載されている会社名、製品名等はそれぞれ弊社もしくは第三者の商標や登録商標です。
- 本書及び関連文書、並びに弊社が提供するサンプルプログラムの著作権は弊社が所有します。

【免責事項】

- 弊社はサンプルプログラムにエラーやバグ等の不具合が無いこと、ユーザーの環境下で正常に動作することを保証しません。
- 本書およびサンプルプログラム、CX2500 仕様の内容を逸脱して製品を使用したことによって生じた不具合 故障やその他損害について、弊社は一切責任を負いません。
- 弊社はサンプルプログラムについて、ソフトの使用やバグ、誤動作や不具合、その他、サンプルプログラムにより生じた損害についても一切責任を負いません。
- 弊社はサンプルプログラムの全部もしくは一部を使用し、または改変して使用した場合の動作や性能に関して一切の責任を負わず、これに起因して発生した損害についても一切の責任を負いません。
- 弊社はサンプルプログラム又は改変後プログラムを使用することにより、第三者との間で著作権、特許権その他知的財産権の侵害を理由として紛争が生じたときは、ユーザーが自らの費用で解決するものとします。
- 弊社は本書に記載されている情報に誤りが無いことを保証するものではありません。本書に記載された情報の誤りにより、ユーザーまたは第三者に損害が生じた場合においても弊社は一切責任を負いません。



3. ファイル構成・ソースファイル一覧

サンプルプログラムのファイル構成は Figure 1 の通りです。これは 4 章にある IDE へのサンプルプログラム用プロジェクトのインポートが完了した後、IDE 上で確認することができます。なお、各構成品については Table 1 を参照して下さい。

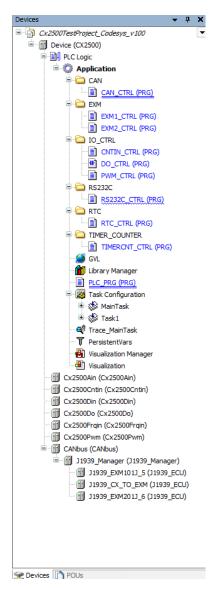


Figure 1 サンプルプログラム ファイル構成(CODESYS-IDE Device ウィンドウ)



Table 1 サンプルプログラム 構成品一覧

名称			摘要	
PLC Logic/Application			-	
	CAN	CAN_CTRL	CANbus ライブラリを用いた CAN 通信制御プログラム	
			(CAN-ch.0 を使用)	
	EXM	EXM1_CTRL	EXM101J シリーズの制御プログラム	
		EXM2_CTRL	EXM201J シリーズの制御プログラム	
	IO_CTRL	CNTIN_CTRL	カウンタ入力の制御プログラム	
		DO_CTRL	デジタル出力の制御プログラム	
		PWM_CTRL	PWM 出力の制御プログラム	
	RS232C	RS232C_CTRL	RS232C 通信の制御プログラム	
	RTC	RTC_CTRL	RTC の制御プログラム	
	TIMER_COUNTER	TIMERCNT_CTRL	タイマカウンタの制御プログラム	
	GVL		グローバル変数群	
	Library Manager		ライブラリマネージャー	
	PLC_PRG		リテイン変数等の制御プログラム	
	Task Configuration		タスク設定	
	Trace_MainTask		トレース	
	PersistentVars		グローバルリテイン変数群	
	Visualization Manager		ビジュアライゼーションマネージャー	
	Visualization		ビジュアライゼーション(6.3.1 項参照)	
Cx2500Ain			アナログ入力機能ドライバ	
Cx25000	Ontin		カウンタ入力機能ドライバ	
Cx2500E	Din		デジタル入力機能ドライバ	
Cx2500Do			デジタル出力機能ドライバ	
Cx2500Frqin			周波数入力機能ドライバ	
Cx2500Pwm			PWM 出力機能ドライバ	
CANbus/J1939_Manager			CAN(J1939)フィールドバス設定(CAN-Ch.1 を使用)	
	J1939_EXM101J_5		EXM101J シリーズから受信するメッセージ設定	
	J1939_CX_TO_EXM		CX2500 から EXM シリーズへ送信するメッセージ設定	
	J1939_EXM201J_6		EXM201J シリーズから受信するメッセージ設定	



4. サンプルプログラムのインポート

本章ではサンプルプログラムの IDE(CODESYS)へのインポート手順を示します。インポートには、弊社の HP からダウンロードできるファイル(ファイル名: Cx2500TestProject_Codesys_v***.projectarchive)を使用します。

なお、IDE との接続等他のセットアップについては別文書「CCOT-24-015 CX2500 Series CODESYS Edition User Manual (For Setup)」の 4、6 章等を必要に応じて参照して下さい。

【手順】

① IDE を起動し、タブメニュー「File」から「Project Archive」→「Extract Archive...」を選択して下さい。

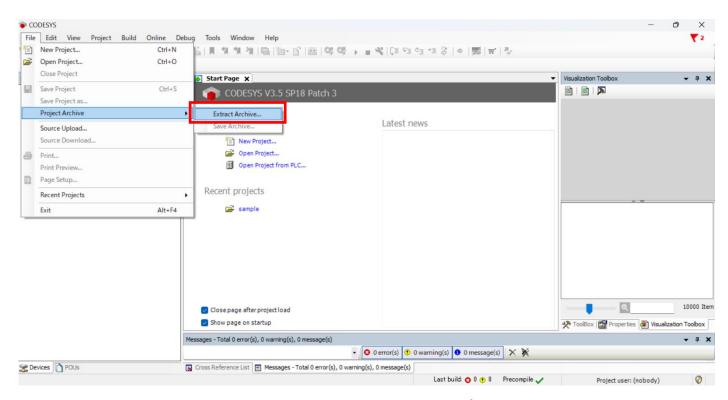
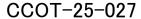


Figure 2 IDE Extract Archive...の選択





② 「Extract Archive」ウィンドウが表示されるので、弊社 HP からダウンロードしたファイル(ファイル名: Cx2500TestProject_Codesys_v***.projectarchive)を選択して「開く(O)」ボタンを押して下さい。すると、このウィンドウが閉じて IDE 上で選択したファイルのインポートが自動でおこなわれます。

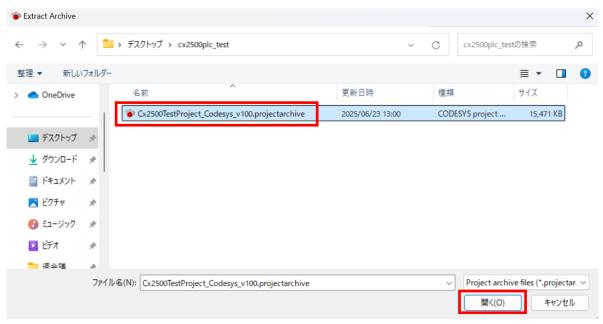


Figure 3 Extract Archive ウィンドウ



③ インポート中に「Extract Project Archive」ウィンドウが表示されます。以下 2 点の設定・確認をおこなった後「Extract」ボタンを押して下さい。

【設定・確認項目】

Locations

➤ 「Extract into the following folder」を選択した後、ファイル (Cx2500TestProject_Codesys_v***.projectarchive)からインポート(抽出)されたプロジェクトファイル の保存先をユーザー任意の場所に設定して下さい。保存先の設定は、「...」ボタンを押すことで保存場 所を選択することができます。次回以降サンプルプログラムを IDE で開く場合には、ここで設定した 保存先に保存されているファイル拡張子(.project)のファイルを IDE 上でオープンして使用して下さい。

Contents

➤ 下図のように、「Items」タブのアイテムすべてに✓印が入っていることを確認して下さい。✓印が入っていない場合はチェックボックスをクリックし✓印を付けて下さい。

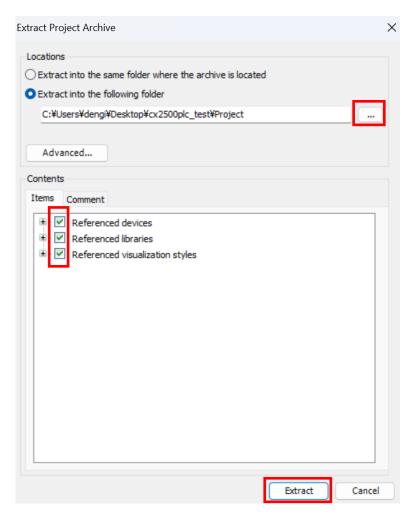


Figure 4 Extract Project Archive ウィンドウ



④ 「Extract Project Archive」ウィンドウが閉じ、下図のようにサンプルプログラムのファイル群が表示されればインポート完了です。ビルドや CX2500 と接続(接続設定要)してプログラムの書き込み・デバッグをすることができます。

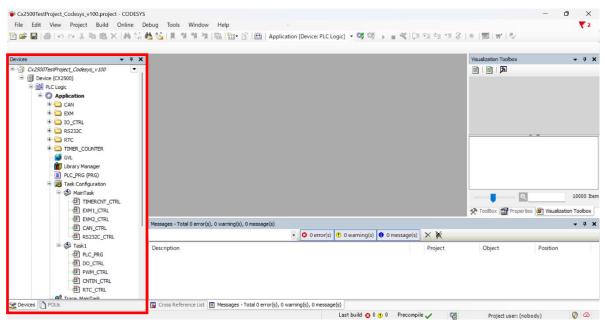


Figure 5 IDE インポート完了後画面例



5. プログラム概要

5.1. 概要

サンプルプログラム(Cx2500TestProject_Codesys.project)は CX2500 が有する I/O や通信機能を簡易的に制御することができるプログラムです。

このサンプルプログラムでは、 β スクを 2つ(MainTask、Task1)設定しています。各 β スクで呼び出される機能については下記を参照して下さい。

なお、各機能の詳細については6章を参照して下さい。

Table 2 タスク設定

項目	タスク名		
	MainTask	Task1	
種類	Cyclic	Cyclic	
呼び出し周期[ms]	50	100	
優先度	1	2	
ソフトウェアウォッチドッグ	無効	無効	
制御プログラム	TIMERCNT_CTRL	PLC_PRG	
(呼び出し順で記述)	EXM1_CTRL	DO_CTRL	
	EXM2_CTRL	PWM_CTRL	
	CAN_CTRL	CNTIN_CTRL	
	RS232C_CTRL	RTC_CTRL	
機能ドライバ	アナログ入力(Cx2500Ain)	_	
	デジタル入力(Cx2500Din)		
	デジタル出力(Cx2500Do)		
	周波数入力(Cx2500Frqin)		
	カウンタ入力(Cx2500Cntin)		
	PWM 出力(Cx2500Pwm)		
CAN フィールドバス	_	CANbus(EXM シリーズとの通信)	



6. 各機能のプログラム

6.1. アナログ入力、周波数入力、デジタル入力

サンプルプログラムにおいて、アナログ入力と周波数入力については、機能ドライバ上で各種入力値を取得する処理のみおこなっています。デジタル入力についても入力値を機能ドライバ上で取得していますが、他機能の制御で一部デジタル入力チャネルの入力状態の値を使用しています。それを使用している機能については関連する節で言及しています。

なお、アナログ入力の入力形式(Input selection)設定値は Table 3、デジタル入力の入力形式(Input selection) 設定値は Table 4 をそれぞれ参照して下さい。

Table 3 アナログ入力 機能ドライバ 設定値(Internal Parameters)

Input selection(チャネル番号)	設定値	備考
Ch0-7、9-15、17-23、25-27、29	FALSE	VOLTAGE INPUT(0-5V 電圧入力)
Ch8	TRUE	ON-OFF INPUT(デジタル入力)
Ch16	TRUE	RESISTANCE INPUT(1kΩ、抵抗プルアップ)
Ch24	TRUE	RESISITANCE INPUT(100 Ω、抵抗プルアップ)
Ch28	TRUE	4-20mA INPUT(4-20mA 入力)

Table 4 デジタル入力 機能ドライバ 設定値(Internal Parameters)

Input selection(チャネル番号)	設定値	備考
Ch0、2、4-22	TRUE	ハイサイド(BAT)入力
Ch1, 3	FALSE	ローサイド(GND)入力



6.2. カウンタ入力制御(CNTIN_CTRL)

サンプルプログラムにおいて、カウンタ入力機能は、以下処理をおこなっています。本節に記載のないチャネルについては、サンプルプログラムでは使用していません。

- 各チャネルのカウンタ値を取得
 - ▶ 機能ドライバ上で取得します。
- 各チャネルのカウンタ値(Ch.0/1:100、Ch.2/3:-100)を設定
 - ▶ ソースファイル CNTIN_CTRL での処理です。処理フローは Figure 6 を参照して下さい。

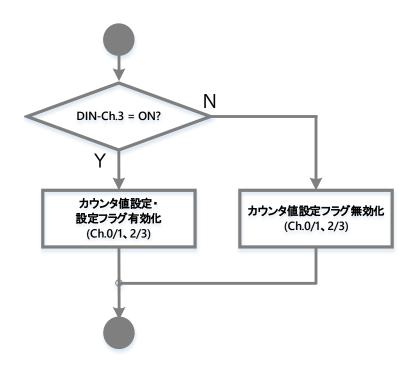


Figure 6 カウンタ入力(CNTIN_CTRL) カウンタ値設定処理フロー ※図上の「DIN」表記はデジタル入力を表す。以降も特記なき限り同様。



6.3. デジタル出力制御(DO_CTRL)

サンプルプログラムにおいて、デジタル出力機能は、以下処理をおこなっています。本節に記載のないチャネルについては、サンプルプログラムでは使用していません。なお、各チャネルの出力形式(Output selection)設定値は Table 5 を参照して下さい。

- 各チャネルのデジタル出力 FB 値を取得
 - ▶ 機能ドライバ上で取得します。
- 各チャネルのデジタル出力指令を設定
 - ▶ ソースファイル DO_CTRL 上での処理です。処理フローは Figure 7 を参照して下さい。

Table 5 デジタル出力 機能ドライバ 設定値(Internal Parameters)

Output selection(チャネル番号)	設定値	備考
Ch0、2、4-15	TRUE	ハイサイド出力
Ch1、3	FALSE	ローサイド出力



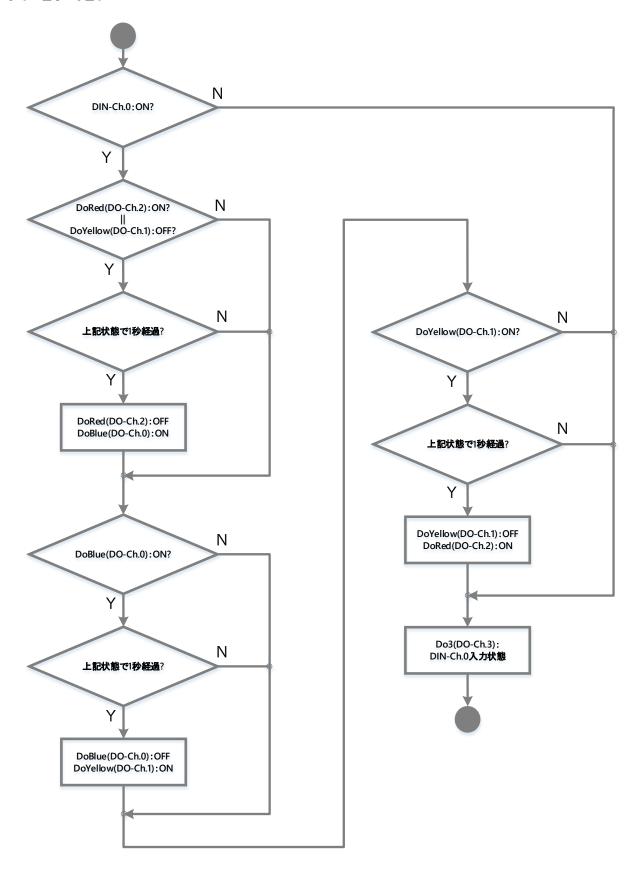


Figure 7 デジタル出力(DO_CTRL) 出力指令設定処理フロー



6.3.1. ビジュアライゼーション

サンプルプログラムでは、DO_CTRL ファイルの処理をビジュアライゼーションでもデバッグできるように設定しています。Visualization の表示画面は下図の通りです。プログラム実行中は DO_CTRL の処理フローに合わせて信号機が順番に点消灯するように設定しています。

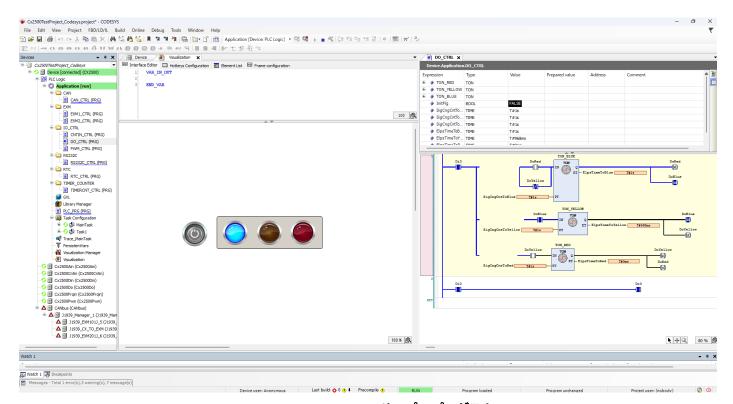


Figure 8 Visualization サンプルプログラム

※信号機の左にあるボタンはデジタル入力-Ch.0 の入力状態を表すためのアイテム



6.4. PWM 出力制御(PWM_CTRL)

サンプルプログラムにおいて、PWM 出力機能は、以下処理をおこなっています。本節に記載のないチャネルについては、サンプルプログラムでは使用していません。

● 各チャネルの電流 FB やエラー状態を取得

PID parameters(Kd)

- ▶ 機能ドライバ上で取得します。
- 各チャネルの緊急停止状態解除および出力電流値を設定
 - ▶ ソースファイル PWM_CTRL での処理です。処理フローは Figure 9 を参照して下さい。

項目	設定値	備考
PWM frequency[Hz]	100	_
PID parameters(Kp)	200	全チャネル共通
PID parameters(Ki)	50	同上

500 同上

Table 6 PWM 出力 機能ドライバ 設定値(Internal Parameters)

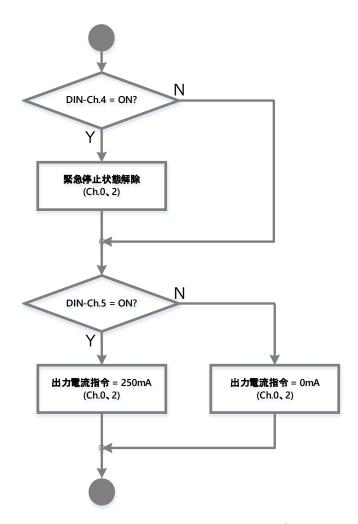


Figure 9 PWM 出力(PWM_CTRL) 出力設定フロー



6.5. CAN 通信制御(CAN_CTRL)

ソースファイル CAN_CTRL における CAN 通信制御は、CANbus ライブラリを使用したプログラムです。 本プログラムでは、以下処理をおこなっています。通信パラメータの設定値は Table 7 の通りです。Table 7 で設定するチャネル以外は本プログラムでは使用していません。

- 通信ポートのオープン(設定値は Table 7 参照)
- データの送受信 (データ(DLC:8バイト)を受信した時、CAN-ID:0x110 としてそのデータを返信する)
 - ▶ ソースファイル CAN_CTRL 上での処理です。処理フローは Figure 10 を参照して下さい。

Table 7 CAN_CTRL(CANbus ライブラリ) 設定値

項目	設定値	
チャネル	Ch.0	
規格	Classic CAN	
ビットレート	250kbps	
受信設定	MaskReceiver を使用	
	全標準 ID メッセージを受信(ID フィルタ・マスク無し)	



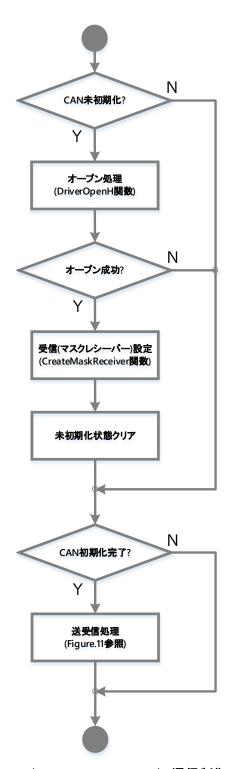


Figure 10 CAN(CAN_CTRL、CANbus) 通信制御フロー(全体)



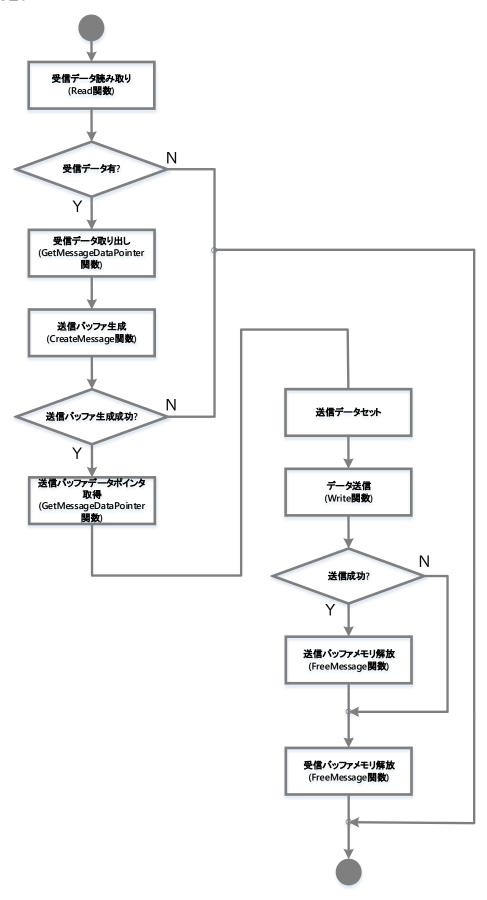


Figure 11 CAN(CAN_CTRL、CANbus) 通信制御フロー(送受信部)



6.6. RS232C 通信制御(RS232C_CTRL)

サンプルプログラムにおいて、RS232C 通信機能は、以下処理をおこなっています。RS232C-Ch.1 はサンプルプログラムでは使用していません。なお、各チャネルの通信パラメータ設定値は Table 8 を参照して下さい。

- 通信ポートのオープン(設定値は Table 8 参照)
- データの送受信 (データを受信した時、同じデータを送信する)
 - ▶ ソースファイル RS232C_CTRL 上での処理です。処理フローは Figure 12 を参照して下さい。

Table 8 RS232C 設定値

項目	設定値
チャネル	Ch.0
ボーレート	19200bps
パリティ	無効
ストップビット	1 ビット
ビット数	8ビット



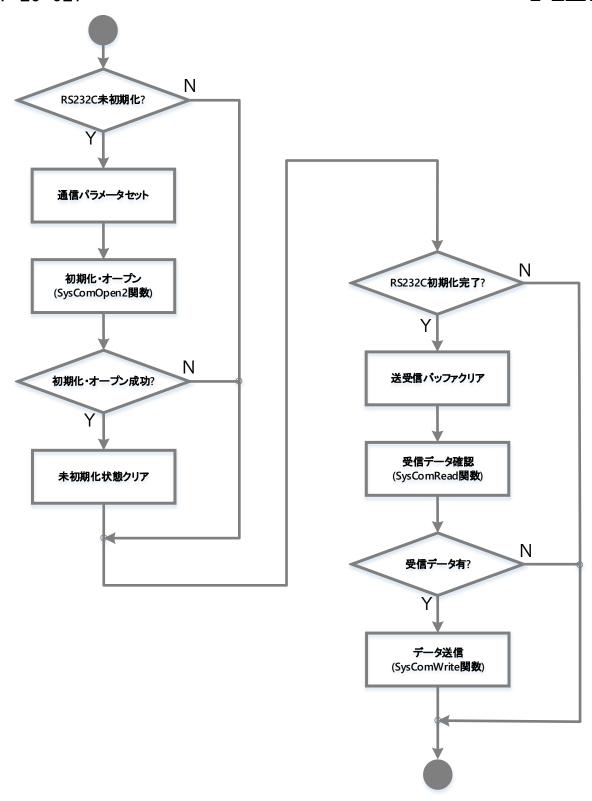


Figure 12 RS232C 通信(RS232C_CTRL) 通信制御フロー



6.7. リアルタイムクロック制御(RTC_CTRL)

サンプルプログラムにおいて、RTC 機能は、以下処理をおこなっています。

- RTC の時刻取得および設定
 - ▶ ソースファイル RTC_CTRL での処理です。処理フローは Figure 13 を参照して下さい。

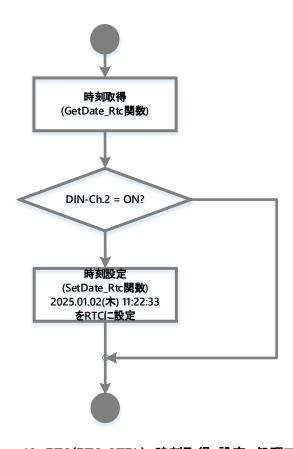


Figure 13 RTC(RTC_CTRL) 時刻取得・設定 処理フロー



6.8. リテイン変数制御(PLC_PRG)

サンプルプログラムでは、リテイン変数(Table 9)を定義しています。これらはソースファイル PLC_PRG で各変数値の加算制御に使われています。

Table 9 サンプルプログラム(PLC_PRG) 通常・リテイン変数定義一覧(初期値は全て 0)

#	変数名	型名	種類
1	g_ulVal	UDINT	グローバル変数・通常変数
2	g_ulRetVal	UDINT	グローバル変数・保持変数
3	g_ucPssVal	USINT	グローバル変数・持続変数
4	ucTmpVal	USINT	ローカル変数・通常変数
5	ucRet	USINT	ローカル変数・保持変数

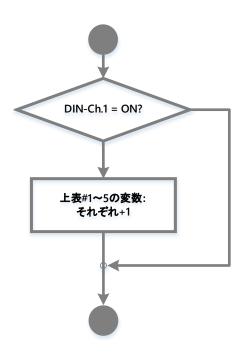


Figure 14 リテイン変数類制御(PLC_PRG) 変数値加算制御フロー



6.9. タイマカウンタ制御(TIMERCNT_CTRL)

サンプルプログラムにおいて、タイマカウンタ機能は、以下処理をおこなっています。

- 現在のタイマカウンタ値の取得
 - ▶ ソースファイル TIMERCNT_CTRL での処理です。処理フローは Figure 15 を参照して下さい。

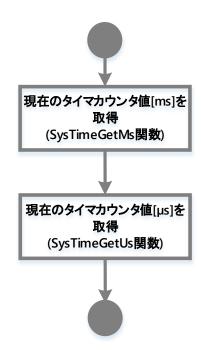


Figure 15 タイマカウンタ タイマカウンタ値(CX2500 起動からの経過時間)取得処理フロー



6.10. CAN フィールドバス—EXM シリーズ制御(EXM1_CTRL、EXM2_CTRL)

サンプルプログラムでは、CAN フィールドバス(J1939)を使用して当社 I/O 拡張コントローラ EXM シリーズ (EXM101J、EXM201J)との通信制御をおこなっています。EXM シリーズのハードウェア仕様や通信プロトコル等については、各種機能仕様書及びソフトウェア仕様書等を参照して下さい。

通信パラメータの設定値は Table 10 の通りです。送受信の設定は CAN デバイス(Table 11)でそれぞれおこなっています。CAN デバイスで送受信の設定をするために、EXM シリーズのデータベースファイル(拡張子.dbc)を IDE にインストールしておく必要があります。弊社 HP からダウンロードできる EXM101J.dbc、EXM201J.dbc がそれぞれ対象のデータベースファイルです。インストール方法については、「CCOT-24-016 CX2500 Series CODESYS Edition User Manual(For Programming)」の 8.3.2 項を参照して下さい。

ソースファイル EXM1_CTRL(EXM101J 用)、EXM2_CTRL(EXM201J 用)では、CAN デバイス(Table 11)で 設定されている送信メッセージのデータを設定しています。設定フローはそれぞれ Figure 16、Figure 17 を参照して下さい。

Table 10 CAN フィールドバス(EXM シリーズ制御) 通信パラメータ設定値

項目	設定値
チャネル	Ch.1
規格	Classic CAN
ビットレート	250kbps

Table 11 サンプルプログラム CAN デバイス一覧

名称	摘要
J1939_EXM101J_5* EXM101J シリーズから受信するメッセージ設定	
J1939_CX_TO_EXM CX2500 から EXM シリーズへ送信するメッセージ設定	
J1939_EXM201J_6**	EXM201J シリーズから受信するメッセージ設定

※各デバイス名称の末尾の数字は、EXM の ID スイッチの値を表します。実際に機器を接続する際は、各 EXM の ID スイッチを表記の値と同じになるように設定(EXM101J は ID:5、EXM201J は ID:6)して下さい。



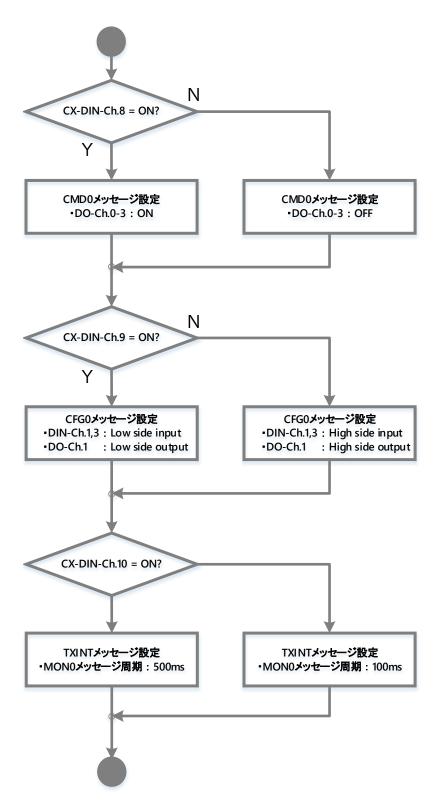


Figure 16 EXM101J(EXM1_CTRL) 送信メッセージ設定フロー

※図上の DIN や DO は EXM101J シリーズのデジタル入出力を表す。



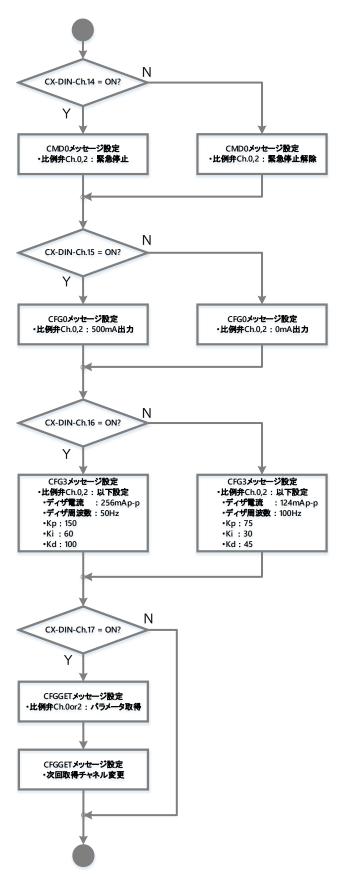


Figure 17 EXM201J(EXM2_CTRL) 送信メッセージ設定フロー



7. 改訂履歴

Rev	改訂日	改訂内容
0	2025.06.27	新規作成