

導入マニュアル

CAN マスタモジュール

目次

第1章 ソフトウェア開発環境

- 1-1 RT共有ライブラリ (RSL) 環境 1-1
- 1-2 ProConOS環境 1-1

第2章 RT共有ライブラリ (RSL) 環境

- 2-1 CANMst RSL 動作環境 2-2
- 2-2 ユーザーにて作成するもの 2-2

第3章 ProConOS環境

- 3-1 MULTIPROG動作環境 3-2
- 3-2 ProConOS CAN動作環境 3-2
- 3-3 ユーザーにて作成するもの 3-2

第4章 その他

システム概要

CAN (Controller Area Network)

CAN は 1986 年、ドイツ Robert Bosch により仕様公開されたプロトコルです。後に、CAN プロトコルは国際標準化機構である ISO (International Organization for Standardization) により標準規格化 (ISO11898/ISO11519) されました。現在ではほぼすべての自動車に採用され、さらに FA (Factory Automation)、産業機器など広く利用されています。

・マルチマスタ

CAN プロトコルではライン型構造で接続される各ノードに平等なバスアクセスが可能な「マルチマスタ方式」を採用しており、バスが空いているときに最初に送信を開始したノードが送信権を得ます。送信したいノードが複数ある場合はメッセージの衝突などが考えられるため、衝突回避策が必要となります。また、ネットワーク管理は各 ECU が個別に行っており、動的なノードの追加や削除が可能です。

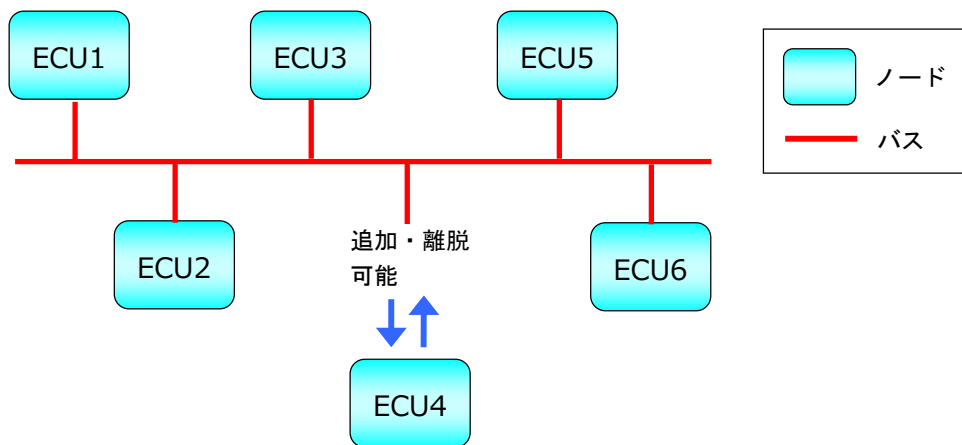


図. CAN通信構造図

CAN は本来マスタ、スレーブの関係はありません。しかし、弊社の CAN マスタモジュールを弊社の産業用 PC/産業用パネル PC に搭載することで、CAN を使ったマスタコントローラとして使用することができます。弊社で販売している CAN 接続の DI032 モジュールを最大 15 台まで接続でき、IN480 点、OUT480 点の制御が可能です。

また、CAN コントローラチップとして NXP (PHILIPS) 製 SJA1000 を採用しており、通常の CAN としても制御することが可能です。

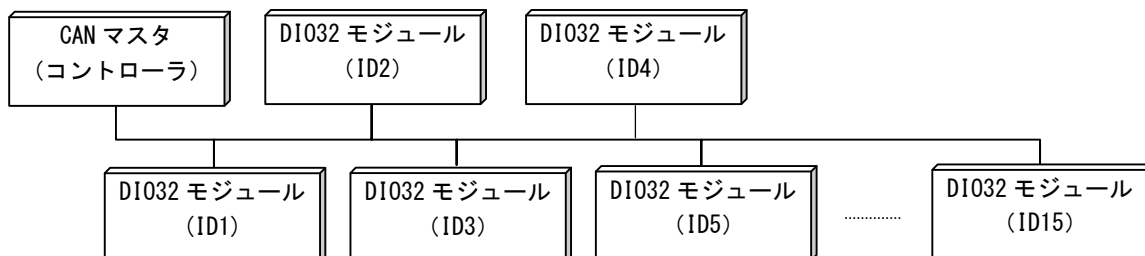
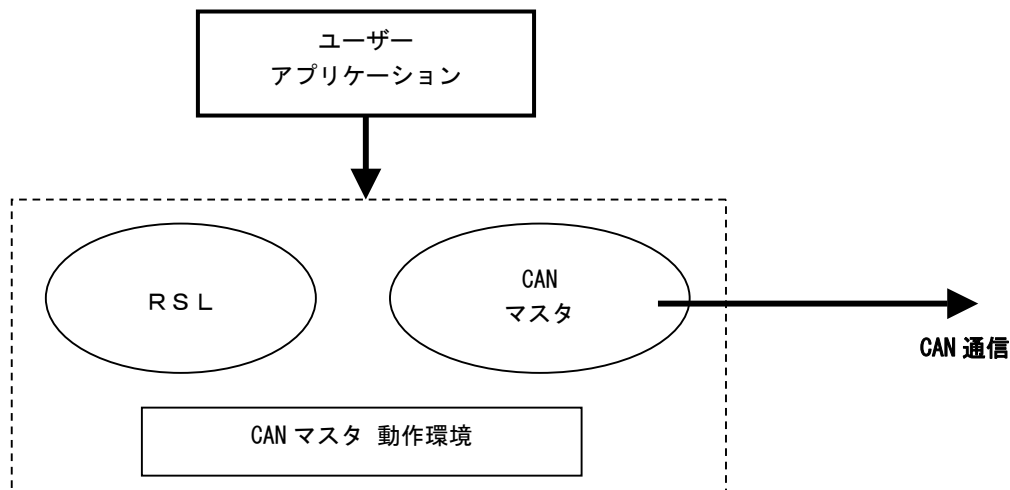


図. CAN接続図

第 1 章 ソフトウェア開発環境

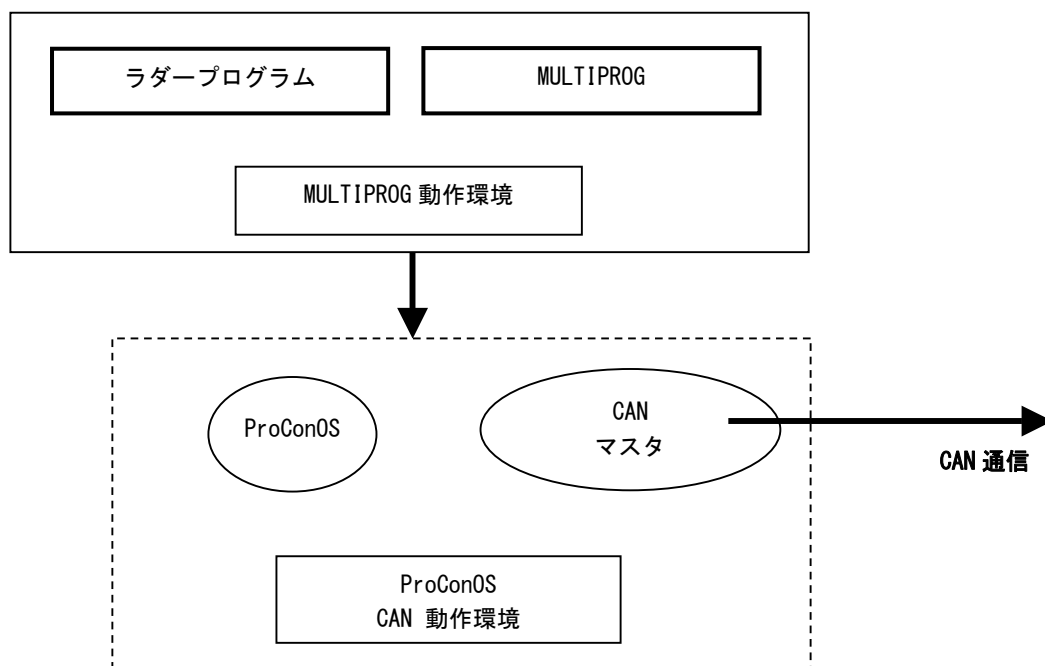
ユーザーは制御ソフトウェアを 2 つの違ったスタイルで開発することができます。

1-1 RT共有ライブラリ (RSL) 環境



C、C++などのプログラミング言語を使用した Intime アプリケーションを作成することにより、CAN 通信を行うことができます。

1-2 ProConOS環境



MULTIPROG を用いたラダー言語による入出力制御を行うことができます。ラダープログラムは ProConOS 上で実行されます。

第 2 章 RT共有ライブラリ (RSL) 環境

RT 共有ライブラリ (以下、RSL とする) 環境では、ユーザーは作成するアプリケーション内で RSL 「CANMst.RSL」 の関数をコールするだけで、CAN 通信、及び各スレーブへの入出力を全て制御できます。

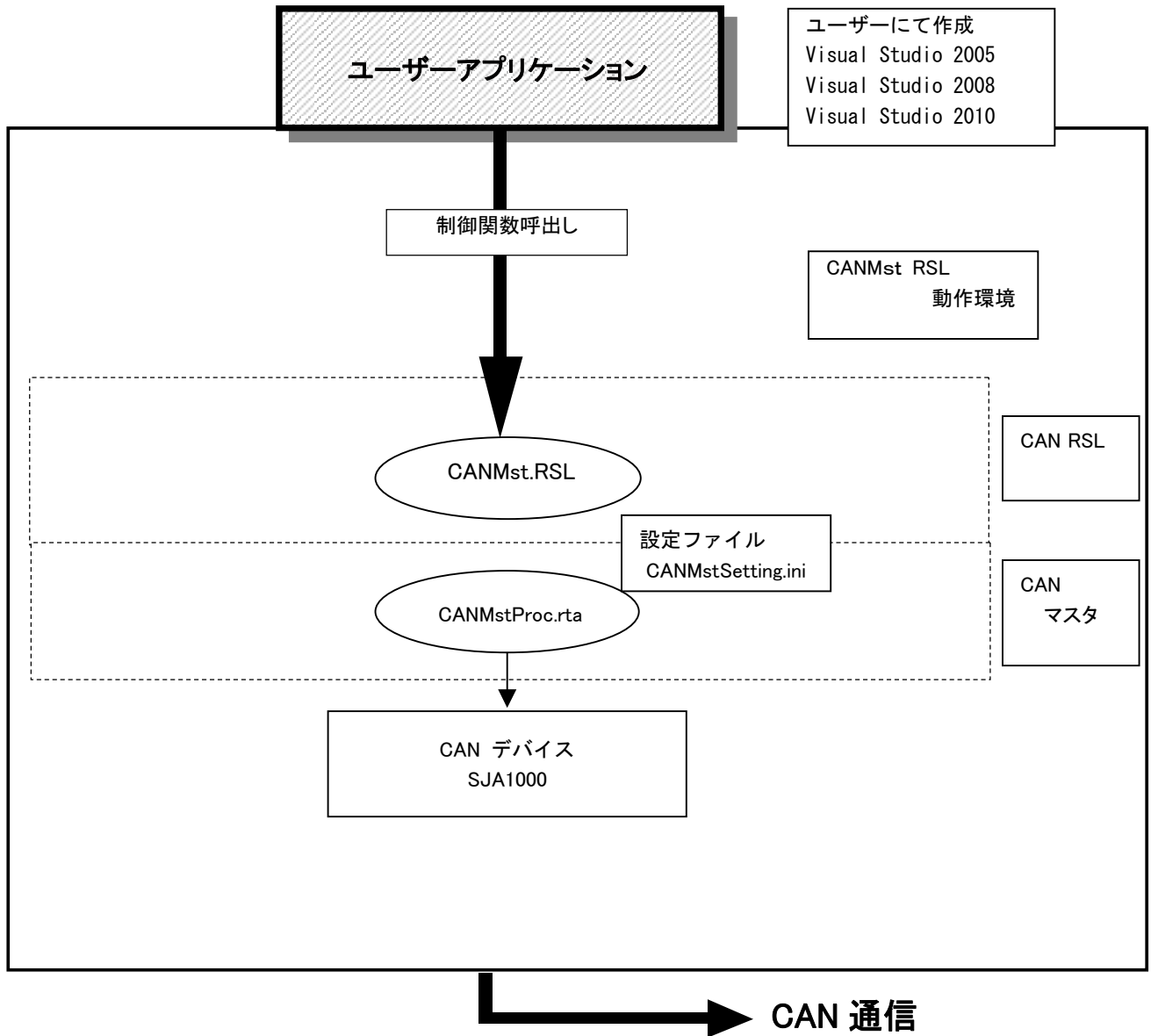


図 2-1. CANアプリケーション構成図

2-1 CANMst RSL 動作環境

- ・ RSL (CANMst.RSL)
- ・ CAN マスタ (CANMstProc.RTA)

ユーザーは CANMst.RSL の制御関数を使用することで、CAN 通信の設定および CAN メッセージの送受信を行うことができます。

SJA1000 のコントロール は CAN マスタで行っているため、ユーザーは SJA1000 の制御を意識する必要はありません。

- ※ CAN 通信の詳細設定を行うには、CANMstSetting.ini を設定しなければなりません。
CANMstSetting.ini の設定方法については「CANMstSetting.ini 設定マニュアル」を参照してください。

2-2 ユーザーにて作成するもの

- ・ アプリケーションソフト
C、C++などのプログラミング言語を使用して作成されたソフトウェア。
CANMst.RSL の関数をコールし、CAN 通信を制御します。
- ・ CANMstSetting.ini (設定ファイル)
使用する CAN の通信方法等を記述します。

- ※ CANMstSetting.ini の設定方法については「CANMstSetting.ini 設定マニュアル」を参照してください。

第3章 ProConOS環境

ProConOS 環境では PLC 開発ツールである KW-Software 社製 MULTIPROG を用いてラダープログラムから CAN 通信を制御します。

この環境では、使用する CAN DI032 モジュール入出力データをラダープログラムで使用するデータ領域への割り当て設定を行います。

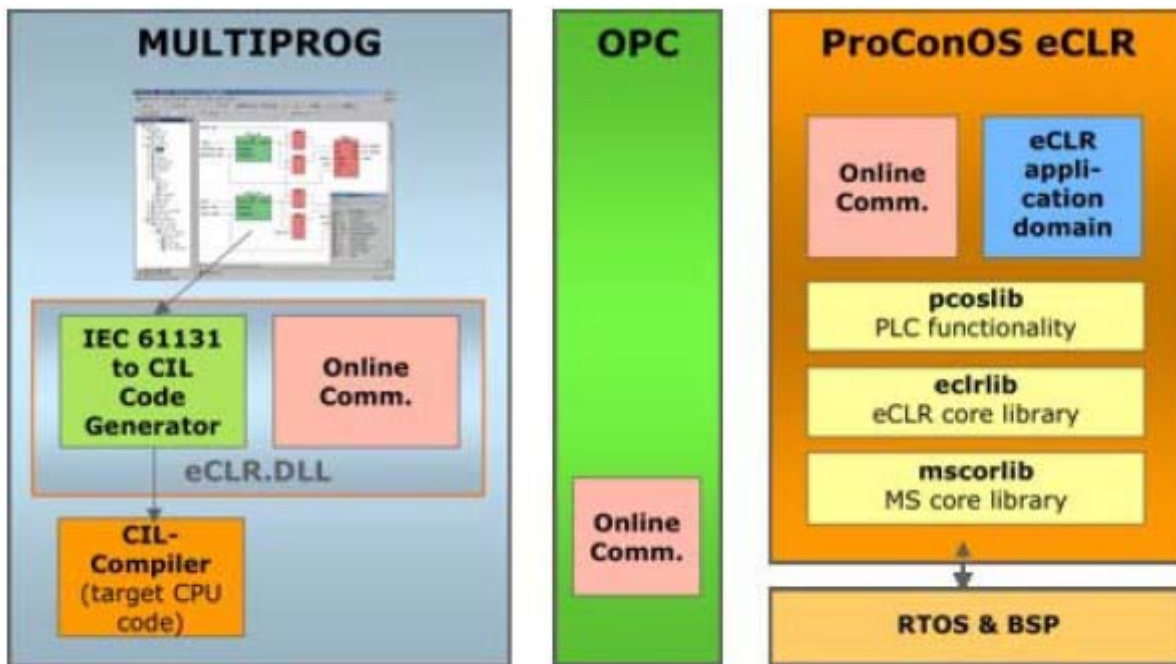


図 3-1. ラダー開発環境構成図

3-1 ProConOS CAN開発環境

- ・ MULTIPROG
ラダープログラムの開発環境を提供します。
- ・ MULTIPROG 用ファームウェアライブラリ
ProConOS で CAN を制御するラダープログラムを作成するために必要となるファームウェアライブラリです。

3-2 ProConOS CAN動作環境

- ・ ProConOS eCLR (eCLRIntime)
ソフトウェア PLC の制御を行います。
- ・ CAN 動作環境
CAN 入出力を制御します。

ユーザーは ProConOS eCLR (eCLRIntime) を動作させることにより、ラダープログラムから CAN DI032 モジュールの入出力データを取り扱うことができます。

ProConOS eCLR の動作には CAN バス通信開始のファンクションブロックと DI032 のリード・ライトを行うファンクションブロックを使って入出力の制御を行うことができます。

ユーザーは、ProConOS eCLR、CAN 通信を意識することはありません。

3-3 ユーザーにて作成するもの

- ・ ラダープログラム
KW-Software 社製 MULTIPROG を使用して作成します。
- ・ CANMstSetting.ini (設定ファイル)
使用する CAN の通信方法等を記述します。

※ CANMstSetting.ini の設定方法については「CANMstSetting.ini 設定マニュアル」を参照してください。

第4章 その他

ここではマニュアルの体系を示します。

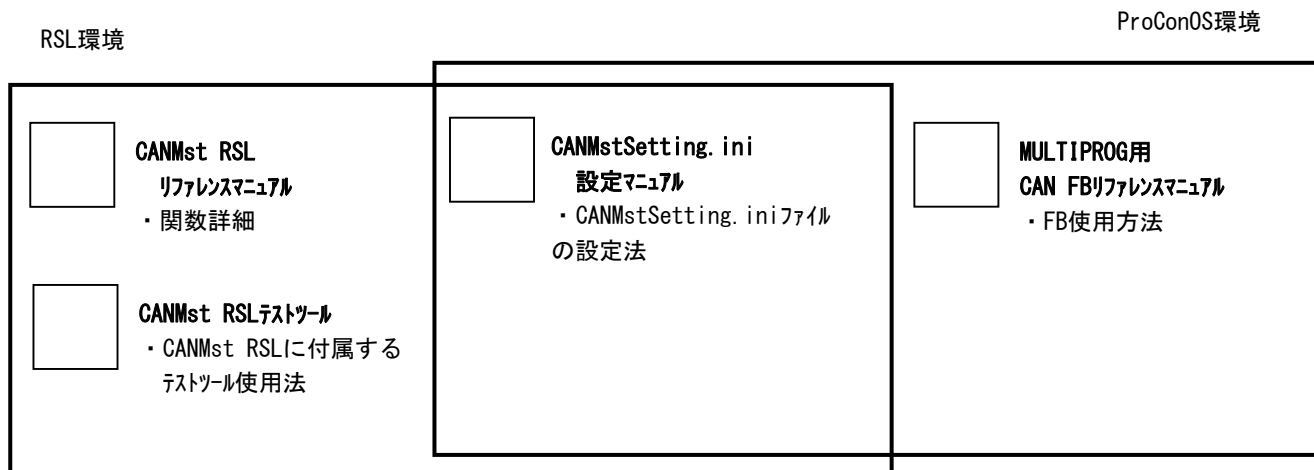


図 4-1. CANマニュアル体系図

このユーザーズマニュアルについて

- (1)本書の内容の一部または全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2)本書の内容に関しては、製品改良のためお断りなく、仕様などを変更することがありますのでご了承下さい。
- (3)本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがございましたらお手数ですが巻末記載の弊社もしくは、営業所までご連絡下さい。その際、巻末記載の書籍番号も併せてお知らせ下さい。

75SSH0027B
75SSH0027A

2014年 10月 第2版
2012年 10月 初版

 株式会社アルゴシステム

本社
〒587-0021 大阪府堺市美原区小平尾656番地

TEL (072) 362-5067
FAX (072) 362-4856

ホームページ <http://www.algosystem.co.jp/>