

分散制御ロボットにおける CANコンポーネント


芝浦工業大学 水川研究室



○ 三浦俊宏 (Toshihiro MIURA)
水川真 (Makoto MIZUKAWA)

目次

1. CANとは
2. 目的
3. CAN Componentsの概要
4. CAN Componentsの仕様
5. 動作確認
6. 使用例
7. 公開資料関連
8. ライセンス・環境について
9. まとめ



©2007 Shibaura Institute of Technology -Mizukawa Lab.-

1. CANとは

ライン型バスポロジーマルチ・マスター方式

耐ノイズ性に優れた物理層
(2線式 差動電圧検出)

CSMA/CA方式バスアクセス

ネットワーク全体でのデータ一貫性

簡潔な情報伝送
(最大8バイトのデータ)

高い通信速度
(最大1Mbps)

優れたエラー検出リカバリー能力

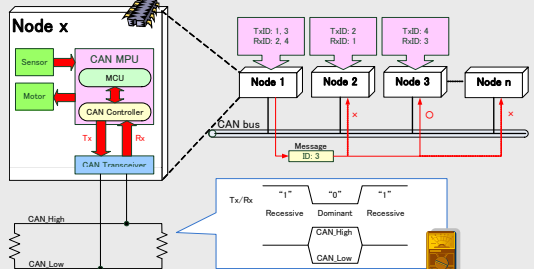
メッセージアドレッシング方式

CANの適用範囲

自動車, 生産オートメーション, ビル内制御, 船舶, 医療機器, ロボットなど

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 1 -

1. CANとは



通信方式採用によるメリット

- ✓ 配線本数の削減
- ✓ コントローラ(Node)の小型化
- ✓ 設計自由度の向上
- ✓ 電氣的信頼性の向上
- ✓ データ通信以外への用途拡大

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 2 -

2. 目的

CAN通信技術の共有・蓄積への貢献を目的とし実装を行った.

ライン型バスポロジーマルチ・マスター方式

耐ノイズ性に優れた物理層
(2線式 差動電圧検出)

CSMA/CA方式バスアクセス

ネットワーク全体でのデータ一貫性

簡潔な情報伝送
(最大8バイトのデータ)

高い通信速度
(最大1Mbps)

優れたエラー検出リカバリー能力

メッセージアドレッシング方式

CANの適用範囲

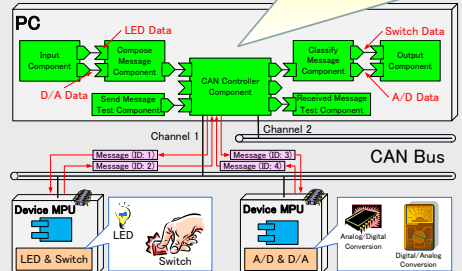
自動車, 生産オートメーション, ビル内制御, 船舶, 医療機器, ロボットなど

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 3 -

3. CAN Componentsの概要

CAN Componentsとは？

CAN通信機能要素をRT-Component化したCAN Controller Componentとこれの動作確認及び評価を行うために作成した他のRT-Componentを含めたもの。



©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 4 -

4.1. CAN Controller Componentの仕様

- 概要 -
2つのCANバスにCANメッセージを送受信することが可能で動的に設定できる受信メッセージのフィルタ機能を実装している。

使用機器
CAN通信機能要素をRT-Component化するにあたり、ベクター・ジャパン社製XLファミリー製品を使用しCAN Controller Componentの実装を行った。

CANcardXL の特徴

- > PCカード (Type II)
- > 64MHzの32ビット マイクロコントローラ
- > 2つの完全に独立したチャンネル
- > CAN 2.0BおよびLIN
- > CANcab/LINcabでのバス トランシーバ
- > プラグ&プレイ

- Port の仕様 -
各InPort, OutPortは、CANバスに送受信するメッセージデータを入力出力しており、このデータ長はCANメッセージ長を示すDLCの値によって変わる。

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 5 -

4.1. CAN Controller Componentの仕様

- 概要 -
2つのCANバスにCANメッセージを送受信することが可能で動的に設定できる受信メッセージのフィルタ機能を実装している。

使用機器
CAN通信機能要素をRT-Component化するにあたり、ベクター・ジャパン社製XLファミリー製品を使用しCAN Controller Componentの実装を行った。

CANcardXL の特徴

- > PCカード (Type II)
- > 64MHzの32ビット マイクロコントローラ
- > 2つの完全に独立したチャンネル
- > CAN 2.0BおよびLIN
- > CANcab/LINcabでのバス トランシーバ
- > プラグ&プレイ

- Port の仕様 -
各InPort, OutPortは、CANバスに送受信するメッセージデータを入力出力しており、このデータ長はCANメッセージ長を示すDLCの値によって変わる。

Port の仕様

ポート種類	ポート名称	データ型	説明
InPort	TxChannel1	TimeOfDateSeca	入力されたデータをCANバスのチャンネル1に送信する
InPort	TxChannel2	TimeOfDateSeca	入力されたデータをCANバスのチャンネル2に送信する
InPort	TxChannelAll	TimeOfDateSeca	入力されたデータをCANバスのチャンネル1とチャンネル2に送信する
OutPort	RxChannel1	TimeOfDateSeca	CANバスのチャンネル1で受信したデータを読み出す
OutPort	RxChannel2	TimeOfDateSeca	CANバスのチャンネル2で受信したデータを読み出す
OutPort	RxChannelAll	TimeOfDateSeca	CANバスのチャンネル1及び2で受信したデータを読み出す

InPort の仕様 (DLC=8 の場合)

データ名称	Bit	Bit	Bit	Bit	Bit	Bit	Bit
DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]
D0	data[0]						
D1	data[1]						
D2	data[2]						
D3	data[3]						
D4	data[4]						
D5	data[5]						
D6	data[6]						
D7	data[7]						

OutPort の仕様 (DLC=8 の場合)

データ名称	Bit	Bit	Bit	Bit	Bit	Bit	Bit
DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]
D0	data[0]						
D1	data[1]						
D2	data[2]						
D3	data[3]						
D4	data[4]						
D5	data[5]						
D6	data[6]						
D7	data[7]						

InPort の仕様 (DLC=4 の場合)

データ名称	Bit	Bit	Bit	Bit
DLC	data[0]	data[1]	data[2]	data[3]
D0	data[0]			
D1	data[1]			
D2	data[2]			
D3	data[3]			

※ InPort の仕様 (DLC=4 の場合) は、OutPort の仕様 (DLC=8 の場合) の一部にのみ適用され、データ長は動的に変化する。

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 6 -

4.1. CAN Controller Componentの仕様

- 受信メッセージのフィルタ機能 -
RTLink上で受信したいCANメッセージのIDを動的に設定することが可能となっている。

Configuration Viewでの設定内容

名称	データ型	説明
ID.From	unsigned long	受信したいCANメッセージの最初のID (10進数)
ID.To	unsigned long	受信したいCANメッセージの終わりのID (10進数)
SetMode	bool	1:フィルタ機能を有効にする 0:フィルタ機能を無効にする

処理負荷を軽減させることが可能

- 概要 -
入力されたデータに対し、あらかじめ設定しておいたID及びDLCの設定を行いメッセージデータの生成を行う。

- プログラム -

```

RTC_ReturnCode_t ComposeMessage::onExecute(RTC_UniqId ec_id)
{
    // ID=1, DLC=1のCANメッセージを生成してTxMessageから出力したい場合
    // (LEDのデータからCANメッセージを生成して出力する場合)
    m_LED.m_read();
    CANTxMessageFrame CANTxMessageTmp;
    memset(&CANTxMessageTmp, 0, sizeof(CANTxMessageTmp)); //クリア
    CANTxMessageTmp.ID = 1; // IDの設定
    CANTxMessageTmp.DLC = 1; // DLCの設定
    CANTxMessageTmp.Data[0].ByteAccess = m_LED.data; // Dataの設定
    m_TxMessage = ComposeTxMessage(CANTxMessageTmp); // 生成
    m_TxMessageOut.write(); // LEDのデータ出力
}
return RTC::RTC_OK;
        
```

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 7 -

4.2. Compose Message Componentsの仕様

TxMessageの仕様 (DLC=8 の場合)

データ名称	Bit	Bit	Bit	Bit	Bit	Bit	Bit
DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]
D0	data[0]						
D1	data[1]						
D2	data[2]						
D3	data[3]						
D4	data[4]						
D5	data[5]						
D6	data[6]						
D7	data[7]						

- 概要 -
入力されたデータに対し、あらかじめ設定しておいたID及びDLCの設定を行いメッセージデータの生成を行う。

- プログラム -

```

RTC_ReturnCode_t ComposeMessage::onExecute(RTC_UniqId ec_id)
{
    // ID=1, DLC=1のCANメッセージを生成してTxMessageから出力したい場合
    // (LEDのデータからCANメッセージを生成して出力する場合)
    m_LED.m_read();
    CANTxMessageFrame CANTxMessageTmp;
    memset(&CANTxMessageTmp, 0, sizeof(CANTxMessageTmp)); //クリア
    CANTxMessageTmp.ID = 1; // IDの設定
    CANTxMessageTmp.DLC = 1; // DLCの設定
    CANTxMessageTmp.Data[0].ByteAccess = m_LED.data; // Dataの設定
    m_TxMessage = ComposeTxMessage(CANTxMessageTmp); // 生成
    m_TxMessageOut.write(); // LEDのデータ出力
}
return RTC::RTC_OK;
        
```

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 8 -

4.3. Classify Message Componentsの仕様

- 概要 -
CAN Controller Componentから出力されたメッセージデータを振り分けて対応するOutPortへ出力する。

- プログラム -

```

RTC_ReturnCode_t ClassifyMessage::onExecute(RTC_UniqId ec_id)
{
    // 受信したCANメッセージを振り分ける
    // (スイッチのデータを振り分けて出力する場合)
    CANRxMessageFrame CANRxMessageTmp;
    memset(&CANRxMessageTmp, 0, sizeof(CANRxMessageTmp));
    m_RxMessageIn.read(); // 受信したデータの読み込み
    CANRxMessageTmp = ClassifyRxMessage(m_RxMessage); // 生成
    switch (CANRxMessageTmp.ID) // IDごとに分ける
    {
        case 2: // ID=2の場合
            m_SW.data = CANRxMessageTmp.Data[0].ByteAccess; // データの設定
            m_SW.Out.write(); // スwitchのデータ出力
            break;
    }
    return RTC::RTC_OK;
}
        
```

RxMessageの仕様 (DLC=8 の場合)

データ名称	Bit	Bit	Bit	Bit	Bit	Bit	Bit
DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]
D0	data[0]						
D1	data[1]						
D2	data[2]						
D3	data[3]						
D4	data[4]						
D5	data[5]						
D6	data[6]						
D7	data[7]						

Channel種類

- 0: 1のチャンネルは実機、0のチャンネルは実機を使用したメッセージ
- 1: 0の場合は「CANcardXL Channel1」からのメッセージ、1の場合は「Virtual Channel1」からのメッセージ
- 2: 0の場合は「CANcardXL Channel2」からのメッセージ、1の場合は「Virtual Channel2」からのメッセージ
- 3: 0の場合は「CANcardXL Channel3」からのメッセージ、1の場合は「Virtual Channel3」からのメッセージ

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 9 -

4.4. その他のRT-Componentの仕様

Input Component
コンソール画面で入力したLED及びD/A変換の指令値を出力する。

- 概要 -
入力されたデータに対し、あらかじめ設定しておいたID及びDLCの設定を行いメッセージデータの生成を行う。

- プログラム -

```

RTC_ReturnCode_t ComposeMessage::onExecute(RTC_UniqId ec_id)
{
    // ID=1, DLC=1のCANメッセージを生成してTxMessageから出力したい場合
    // (LEDのデータからCANメッセージを生成して出力する場合)
    m_LED.m_read();
    CANTxMessageFrame CANTxMessageTmp;
    memset(&CANTxMessageTmp, 0, sizeof(CANTxMessageTmp)); //クリア
    CANTxMessageTmp.ID = 1; // IDの設定
    CANTxMessageTmp.DLC = 1; // DLCの設定
    CANTxMessageTmp.Data[0].ByteAccess = m_LED.data; // Dataの設定
    m_TxMessage = ComposeTxMessage(CANTxMessageTmp); // 生成
    m_TxMessageOut.write(); // LEDのデータ出力
}
return RTC::RTC_OK;
        
```

Output Component
入力されたデータをコンソール画面に出力する。

Send Message Test Component
コンソール画面で入力したID及びDLC、データからCAN Controller Componentに出力するメッセージデータの生成を行い出力する。

Received Message Test Component
CAN Controller Componentで出力されたメッセージデータをコンソール画面に出力する。

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 10 -

5. 動作確認

Bit Shift Component
 PSoCは、A/D変換を12[Bit]でD/A変換を9[Bit]で行っている。そのため、「Bit Shift Component」では入力されたデータを3[Bit]シフトさせ（下位3[Bit]切捨て）出力している。

使用機器
 ◆ ノートPC (W2)
 Pentium M : 900[MHz]
 Memory : 512[MB]
 ◆ CANcardXL
 ◆ PSoC CAN
 PSoC : CY8C29466
 CAN Controller Chip : MCP2515

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 11 -

6. 使用例

PAR-NE07

PAR-NE07 (Physical Agent Robot for Natural Environment)
 屋外環境でロボットを運用するための技術獲得を目的に本研究室で開発を行い『つくばチャレンジ』に参加したロボット。
 同研究室からコンテストに参加している『屋外ナビゲーションにおけるGPSコンポーネント』のRT-Componentと組み合わせることで目的地まで自律でロボットが動く。

クリスマスツリー

クリスマスツリー
 クリスマスツリーをCAN通信で制御している。
 LED1個ずつを周期的に制御し光の強度を変化させている。

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 12 -

7. 公開資料関連

公開資料

- ✓ CAN Componentsのソースコード … Visual Studio 2005
- ✓ ユーザーマニュアル … CAN Componentsの仕様・使用方法等を記載
- ✓ 概要資料 … CAN Componentsの概要について記載

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 13 -

8. ライセンス・環境について

ライセンス等について
 CANインターフェイスを制御するAPIのソースコード等を除き、CAN Componentの著作権は、芝浦工業大学水川研究室に帰属します。
 そして、APIのソースコード等の著作権は開発元の"Vector Informatik"にあります。

開発環境

- Windows XP
- Microsoft Visual Studio 2005 (VC++8.0)
- RT-Middleware (OpenRTM-aist-0.4.0)
- CANcardXL (ベクター・ジャパン社製XLファミリー製品)

動作確認環境

- Windows XP (Visual Studio 2005でコンパイル)
- RT-Middleware (OpenRTM-aist-0.4.0)
- CANcardXL (ベクター・ジャパン社製XLファミリー製品)
- CANboradXL (ベクター・ジャパン社製XLファミリー製品)

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 14 -

9. まとめ

- ◆ CAN通信機能要素をRT-Component化するにあたり、ベクター・ジャパン社製XLファミリー製品を使用LRT-Component化した。
- ◆ CAN Componentsの仕様・構成について述べた。
- ◆ CAN ComponentsのCompose Message ComponentとClassify Message Componentを作りかえることによって容易にDevice MPUの変更・機能拡張などに対応することが可能である。
- ◆ 以下の資料を提供することを述べた。
 - ✓ ソースコード … Visual Studio 2005のプロジェクトで提供
 - ✓ ユーザーマニュアル … 仕様及び使用方法等を記載
 - ✓ 概要資料 … CAN Componentsの概要について記載

©2007 Shibaura Institute of Technology -Mizukawa Lab.- - 15 -

分散制御ロボットにおけるCANコンポーネント

芝浦工業大学 水川研究室

○ 三浦俊宏 (Toshihiro MIURA)
 水川真 (Makoto MIZUKAWA)



PSoC CAN



PSoC CAN

PSoC CANのコンセプト

- ✓回路を学生が簡単に作成することが可能
- ✓部品は低価格であり入手性が良いものを使用
- ✓簡単にCAN通信することが可能
- ✓PSoCを使用しているため容易にセンサを使用したノードを作成することが可能



PSoCとは

PSoCはプログラマブルアナログ・デジタル混載マイコンである。各機能はモジュールの組み合わせにより実現可能。動的に再構成することも可能。

PSoC CANの仕様

PSoC : CY8C29466(他のPSoCにも変更可能)
 CAN Controller Chip : MPC2515
 CAN Transceiver Chip : PCA82C250
 入力電圧 : 5.2~12[V]

PSoC CAN開発!について

芝浦工業大学 水川研究室
 指導教員: 水川 義
 ソフト開発: 丸十九里 洋介
 回路設計: 高木 和貴

©2007 Shibaura Institute of Technology -Mizukawa Lab.-