[第1章]

信頼性の高い通信を行う

中距離マイコン間インターフェースCANとは

● CANとは

CANとは，「Controller Area Network」の略で，BOSCH社が開発した車載ネットワークのことです．元々は自動車内の機器間の通信で使用されるためのシリアル・バスですが，信頼性の高さや優れた故障検出機能により，オートメーション機器の制御などにも利用されています．

CANは仕様が公開されているため，ほかの自動車メーカなどもこの方式を採用しているところがあります．

最近のカー・エレクトロニクスではコンピュータ化が進み，いたるところに多数のマイコンが使われています．これらの機器を個別に配線していたのでは，膨大な量の配線が必要になり，重量もコストも増大します．この問題を解消する手段として開発されたのがCANです．ネットワーク上で短い情報(コマンドやステータスなど)をやり取りすることで各マイコンを制御できるため，車内配線の大幅な省力化が可能です．

CANの特徴として，比較的長距離の伝送が可能，通信レートが高い，差動通信路のためコモン・モード(同相)ノイズに強い，エラー検出機能が充実していてリカバリが容易などが挙げられます．また，マルチ・マスタ，というより，すべてのデバイスがマスタでありスレーブであり，優先順位もメッセージの内容に依存するというのも大きな特徴です．

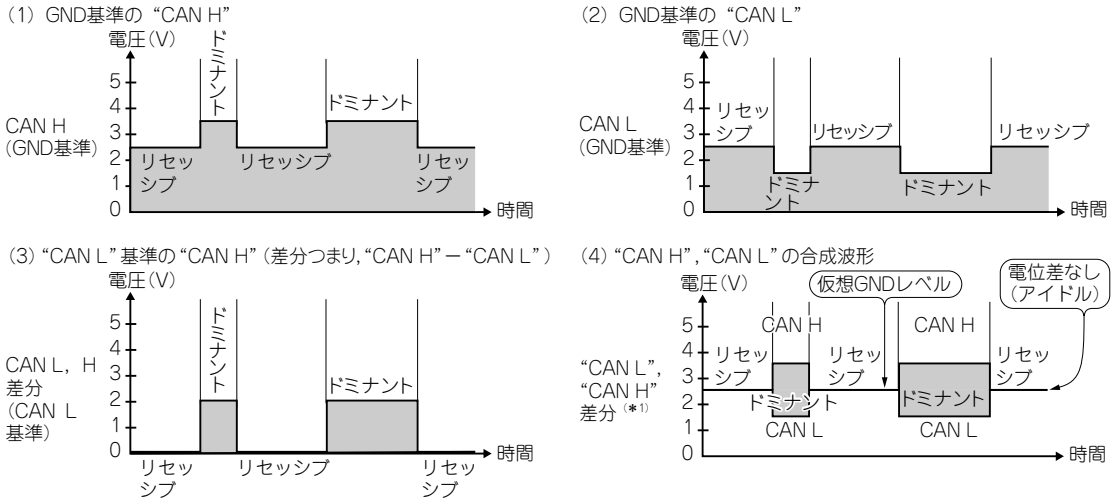
ただし，一度に送受信できるデータは最大8バイトと短く，大量のデータをやり取りするのには向いていません．

本書では，比較的安価で手軽に実現できるマイコン機器間の中距離通信の通信手段としてCANを取り上げ，マイコン機器を制御する具体的な方法を解説していきます．

● 転送レート

CANバスは比較的長い距離でも通信できるのが特徴の一つですが，当然ながら，伝送路が長いと通信速度も落ちます．規格上の最高レートは伝送路が40mのときで1Mbps，1000mのときでは50kbpsとなっています．

CANは通信レートの違いにより250kbps以上のものはハイ・スピードCAN，それ未満で，125kbps程度のものはロー・スピードCANと分類されています．実際の自動車では，エンジンやブレーキなどの制御系にはハイ・スピードCAN，パワーウィンドやワイパなどの制御系にはロー・スピードCANやLIN(これもクルマ車載ネットワークの一つの標準規格)というように数種類のバスが



(*1) CANバスの信号ラインにはGNDは存在しないが，2.5Vを仮想的なGNDとしたときの“CAN L”，“CAN H”両信号の波形を示したもの.

図1-1　CANバスの信号
これらのタイムチャートはCANバスの信号の電圧レベルを基準電圧を変えて表現したもの.

混在しています.

同一のCANバスに接続されているすべてのCANデバイスは，同じ転送レートで動作している必要があります.

● CANバスの信号――ドミナントとリセッシブ

CANバスの信号は，2本の信号ラインで伝達されます．コモン線(GND)は不要です．この二つの信号は，“CAN H”，“CAN L”と呼ばれ，両信号間の電位差(電圧差があるか，ないか)で信号を伝達します．I^2CやTTLレベルの信号のようにGNDに対する電位レベルで信号を伝達するのとは異なります.

実際の自動車内の配線では，CANラインと一緒に電源もケーブルで分配されていて，電源を共用している関係でGNDがシャーシなどで接続されていますが，本来は電源を共用する必要もないため，2本の信号線だけで通信できます.

このように電位差で動作するものを差動(ディファレンシャル)式といいます．差動式は伝送路などで拾う同相のノイズが打ち消されるため，外来ノイズの影響を受けにくく，自動車などのノイズの発生源の多いところではとくに有用な方式です．差動式は，RS-422などの長距離伝送にも利用されています.

CANバスの信号状態には**図1-1**のように2通りあり，“CAN L”，“CAN H”両ラインの，

- 電位差がある状態をドミナント(Dominant)
- 電位差がない状態をリセッシブ(Recessive)

といいます．ドミナントは信号がアクティブである，リセッシブは非アクティブ(アイドル)である，と言い換えることもできます．このように，ドミナントもしくはリセッシブの状態と長さの組み合わせにより，データを伝達します.

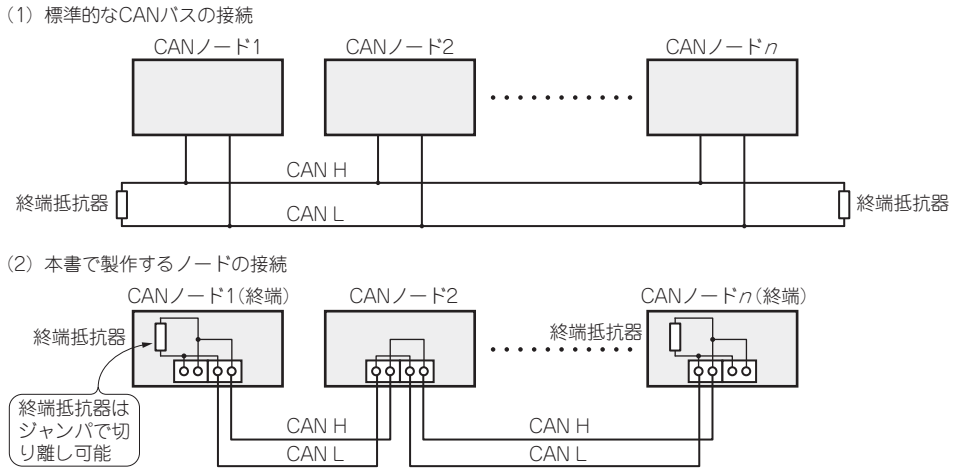


図1-2　CANノードの接続
一般的なCANノードの接続方法と，本書で製作するCANノードの接続形態を示す．本書で製作するものはCANの端子を二つ持ち，内部で並列に結線されているため，ノードを数珠つなぎにできる．

CANは，同期用のクロック信号がなく，同時に双方向の通信ができないことから，非同期の半二重通信方式といえます．

同一バス上で複数のノード(接続されているCANデバイス)が信号を出力する際，あるノードがリセッシブ状態にある場合でも，ほかのノードがドミナントを出力すると，バスの状態はドミナントに上書きされます．リセッシブはバスがアイドル状態と考えればわかりやすいでしょう．この仕組みはバスのアービトレーション(調停)やACKの応答に利用されます(後述)．

CANバスの両端には終端抵抗が必要です．**図1-2**はバスにつながるCANデバイス(ノード)間の配線例を示したものです．

● 通信波形の確認

図1-3はCANバス・トランシーバを二つ接続して，片側にオシレータ(発振器)をつなぎ，CANバスの“CAN H”と“CAN L”の2本の信号をオシロスコープで実際に測定したものです．オシレータの周波数はロー・スピードCANの代表値である125kHzに設定してあります．

トランシーバのGNDをオシロスコープのGNDレベルにして，“CAN H”，“CAN L”の両信号をオシロスコープのCH_1とCH_2に接続し，2チャネル同時に表示させています．

約2.5Vを中心として約±1Vで信号が伝達されている様子が確認できます．なお，CANバス・トランシーバについては第2章以降で説明しています．

● CANノード

CANバスに接続される各デバイスは「ノード」と呼ばれます．同じくシリアル・バスの規格であるI²C(シングル・マスタ・モードの場合)やSPIのように一つのマスタに対して複数のスレーブが接続されているというのではなく，各ノードがマスタであり，スレーブでもあります．通信を始めるノー

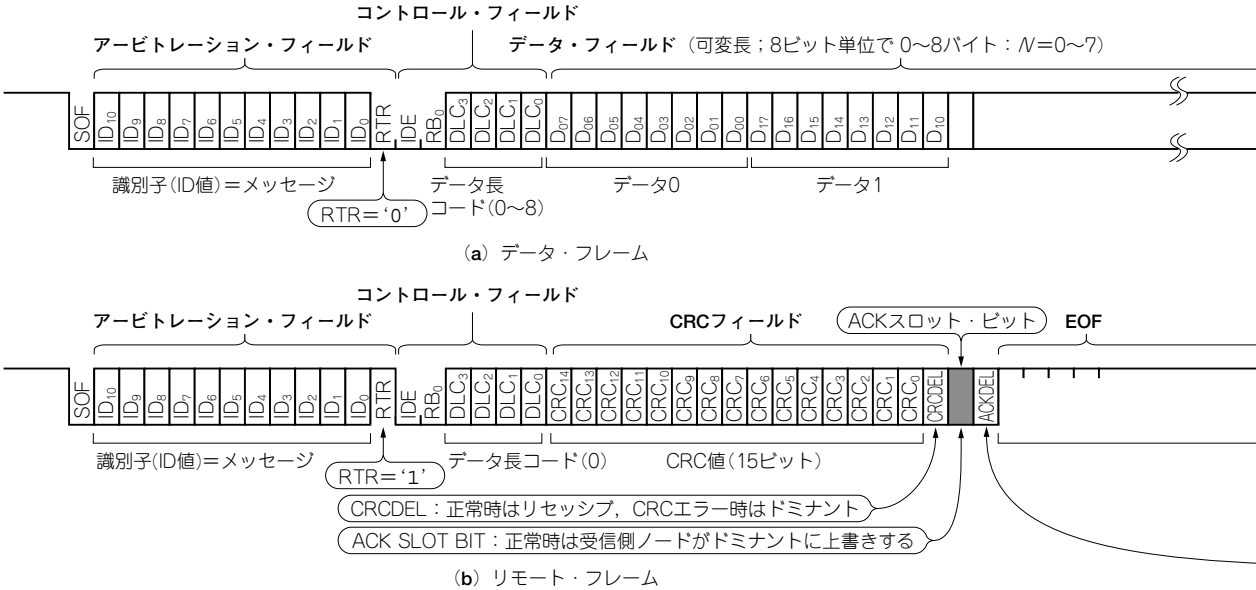


図1-4　データ・フレーム/リモート・フレーム
標準モードのデータ・フレーム，リモート・フレームのフォーマットを示す．データ・フレームはRTR=‘0’で，DLCビットで指定されたバイト数（0～8バイト）のデータ・フィールドをもつ．リモート・フレームはRTR=‘1’でデータ・フレームはない．ACKスロット・ビットは送信ノード側はリセッシブを出力し，正常受信時には受信側ノードがドミナントに上書きする．

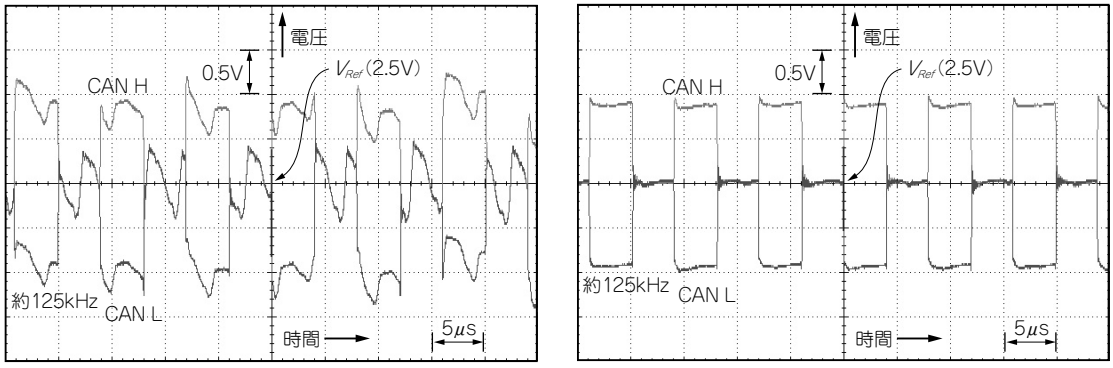
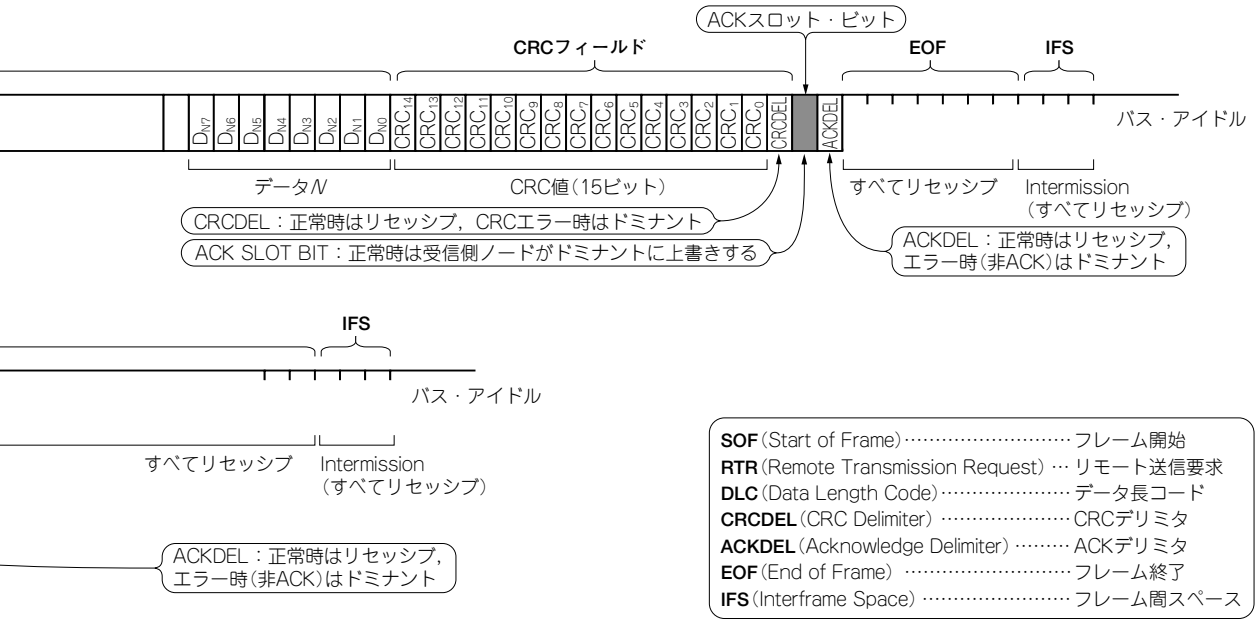


図1-3　CANバスのオシロ測定波形
125kHzの矩形波をトランシーバに入力した際のCAN出力をオシロスコープで測定した波形．(a)は大きな同相ノイズが混入して，波形が大きくひずんでいるが，差動の原理により，ノイズが打ち消されて通信には問題ない．

ドがマスタとなり，受信する側がスレーブということになります．

各ノードは**図1-2**のように接続され，バスの終端には終端抵抗器が付いています．

各ノードは，データを送信する際はデータ・フレームまたはリモート・フレームにID値（メッセージを含めて送信します．このID値はフレームの優先順位も表していて，同時に複数のノードが送信を始めようとするときのアービトレーション（調停）に利用されます．



ノードの数は仕様上の制限はありませんが，ハイ・スピードCANの場合は最大30と規定されています．また，トランシーバや伝送路の電気的な特性や制限により，ノード数は制限されます．

● フレームのフォーマット

図1-4にフレームのフォーマットを示します．フレームとは通信情報として意味をもつフィールドの集まりのことで，CAN通信の基本単位です．この図では，“L”レベルがドミナント，“H”レベルがリセッシブを表しています(リセッシブがアイドル状態)．

CANの転送データには4種類のフレームが規定されています．各フレームはいくつかのフィールドから構成されています．

データ・フレームまたはリモート・フレームの場合，フレームはSOF(Start of Frame)で始まり，アービトレーション・フィールド，コントロール・フィールド，データ・フィールド(データ・フレームの場合のみ)，CRCフィールド，ACKスロット，EOF(End of Frame)と続き，最後にIFS(Interframe Space)フィールドで終わります．

SOFとEOFは文字通り，フレームの最初と最後を示すもので，それぞれ1ビットのドミナントが送信されます．

アービトレーション・フィールドには識別子(以下ID値)とRTRビット(データ・フィールドかリモート・フィールドかを区別するためのフラグ)が含まれています．このID値はメッセージとも呼ばれ，値の大きさにより，優先順位も決まります(値が小さいほど高優先)．

データ・フィールドは可変長(0～8バイト)のデータ領域で，その直前にあるコントロール・フィールド内の4ビットのDLCフィールドで長さが指定されています．CRCフィールドは15ビットのCRC(巡回冗長検査)の値が設定されます．CRCDELはCRCフィールドを区切るためのデリミタです．CRCエラーがない場合にはドミナントですが，エラーがある場合は，リセッシブに設定されます．ACKスロットでは，送信側のノードはリセッシブを出力しますが，受信側ノードが正常にフレームを受信できたときに受信側ノードがドミナントに上書きします．ACKDELはACKビットのデリミタです．

IFSは次のフレームが送信できるようになるまでの時間稼ぎのフィールドで，フレームの区切りをつけるのと，内部処理のための時間を確保するために挿入されています．

次に四つのフレームの機能を簡単に説明しますが，各フィールドの内容については**図1-4**も参照してください．なお，アービトレーション・フィールドの長さにより，標準フォーマットと拡張フォーマットの2種類が規定されていますが，本書ではビット長が11ビットの標準フォーマットについて説明します．拡張フォーマットの場合でもデータ長が増えるだけで動作は同じです．

(1) データ・フレーム

相手にデータを送信するフレームです．データ長は0～8バイトの可変長で，データ長は，コントロール・フィールドにある4ビットのDLCフィールドで指定できます．データ・フレームではRTRビットがドミナントに設定されます．なお，データ長が'0'の場合は，RTRビットがドミナントである以外はリモート・フレームと同じフォーマットになります．

(2) リモート・フレーム

相手へデータを要求するフレームです．データ・フレームのデータ・フィールドがないようなフォーマットになっています．リモート・フレームはRTRビットがリセッシブに設定されます．このフレームを受け取ったノードは(1)のデータ・フレームでデータを送り返します．

(3) エラー・フレーム

エラーを検出したノードがエラー状態を通知するために送信するフレームです．

エラーが発生すると，ほかのフレーム送信の途中であってもエラー・フレームが割り込んで出力されます．エラー・フレームは連続6ビットのドミナントをビット・スタッフィング(後述)なしで送信したあと，個別のエラー・フラグとエラー・デリミタを送信します．エラーについては「CANのエラー」の項を参照してください．

(4) オーバロード・フレーム

受信した側の処理が追いつかないときに，このフレームを出力することで，次のメッセージが出力されるのを待機させて，処理時間を稼ぎます．

最近ではあまり使われていないようですが，なんらかの理由で相手ノードを待たせたいときに使用できます．

● メッセージ

データ・フレームとリモート・フレームにはアービトレーション・フィールドが含まれていますが，このフィールドの値(ID値)がメッセージを表しています．

アービトレーション・フィールドは標準フォーマットの場合は11ビットなので，2^{11}種類（実際は

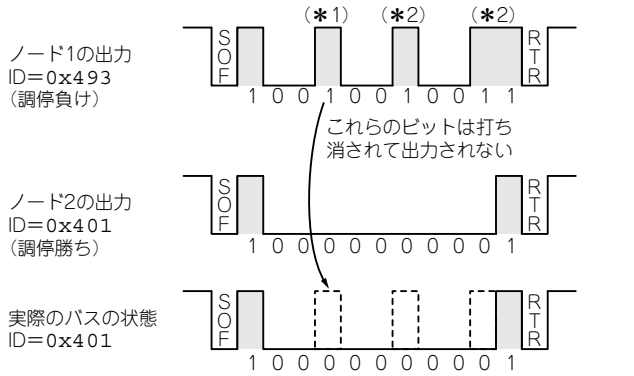


図1-5　CANバスのアービトレーション(調停)
二つのノードが同時に送信を始めた場合の調停の仕組みを示す．この図はID値が小さいノード2が調停勝ちする場合の例．チャートの“H”レベルはリセッシブ，“L”レベルはドミナントを示す．なお，この図はビット・スタッフィングを考慮していないので注意(p.18参照)．

それより少し少なくなる)のメッセージが作成できます．また，拡張フォーマットの場合は，アービトレーション・フィールドは29ビットあるので，2^{29}種類のメッセージが作成できます．

本書では標準フォーマットのID値(メッセージ)を単にSIDと表記することがあります．

● フィルタとマスク

受信したメッセージを振り分けるための仕組みで，特定のメッセージだけを受信したり(フィルタ機能)，ある範囲の複数のメッセージを受信したい(マスク機能)ときに使用します．マスクとは，特定のビットを隠して，そのビットは何でもよいという状態にすることを意味しています．詳細はMCP2515の「フィルタとマスク」の項で説明します．

● CANノードの優先順位とアービトレーション(調停)

CANネットワークは半二重(送受信の交互，片側通行)通信なので，送信と受信は同時にはできません．また複数のノードが一斉に送信することもできません．

そこで，複数のノードがバスをシェアしながら通信する仕組みが用意されています．

CAN通信では，ノードに関係なく，送信されるメッセージごとに優先順位が決まっていますが，基本的にはメッセージの優先順位にも関係なく，早い者勝ちで送信されます．すでに通信が始まってバスが使用中の場合は，たとえ優先順位の高いメッセージを送信しようとしても，バスが空くのを待たなければなりません．

CANでは1回の通信データの量が少ないため，高優先順位のメッセージが少し待たされてもそれほど問題はありません(と考える)．

もし，まったく同時に複数のノードが通信を始めた場合は，アービトレーションの機能が働きます．この場合は，優先順位の高いメッセージを出力しているノードが使用権を獲得し，そのまま送信を継

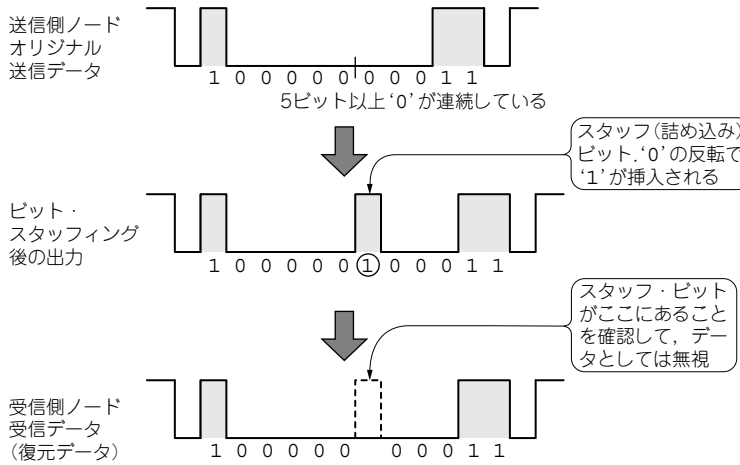


図1-6　ビット・スタッフィング
ビット・スタッフィングが発生する場合の通信データの例．5ビット以上同一レベルの信号が連続した場合に1ビットの反転ビットが挿入される．受信側ではビット・スタッフィングが発生する状況で，反転ビットがあることを確認して，ビット・データとしては無視することで，元のデータ・ビット列を復元する．

続しますが，優先順位の低いほうのノードは送信を中断して，バスが空いてからリトライします．

優先順位は，データ・フレームまたはリモート・フレームのアービトレーション・フィールドのメッセージ値(ID値)が小さいほど高順位になります．**図1-5**はアービトレーションの仕組みを簡略化して示したものです．

複数のノードが同時に通信を始めた場合，一斉にフレームを出力するわけですが，バスの信号はID値の小さいID値で上書きされてしまう(ドミナントはリセッシブを上書きする)ため，結果的にID値の小さいものの値がバス上に出力されることになります．これが，ID値が小さいほど優先順位が高いという理由です．

ID値の大きいほう(優先順位の低いほう)のID値は打ち消されてしまいます．このようにして調停が成立します．

調停負けは，自分が出力したID値と実際にバスに出力されたID値を比較して，結果が異なっていれば負けたと判断できます．通常，このような判定はCANコントローラのハードウェアが自動でやってくれます．

同じID値のデータ・フレームとリモート・フレームが同時に送信された場合，フレーム中のRTRビット(リモート送信要求ビット)がドミナントであるデータ・フレームのほうが調停勝ちします．

● ビット・スタッフィング

スタッフとは詰め込むという意味ですが，ビット・スタッフィングとは，同一の信号レベルが5ビット以上連続した場合，その5ビットの後に，レベルを反転したビットを1ビット挿入して(詰め込んで)出力することです．つまり，6ビット以上，同一レベルの状態が連続しないようにします．

図1-6は‘0’が8ビット連続した場合に‘1’のスタッフ・ビットが挿入される場合の例を示していま

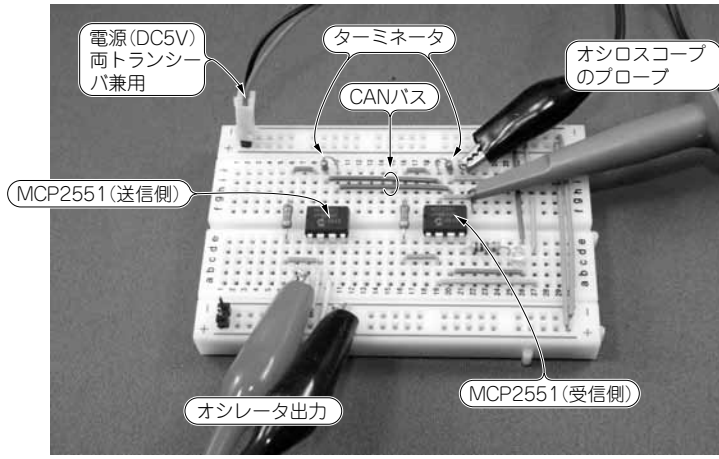


写真1-1　CANトランシーバ通信実験
これはブレッドボード上で二つのCANトランシーバ(MCP2551)を接続して通信を確認したようす．オシレータで125kHzを発生させ，オシロスコープで波形を確認した．この写真ではMCP2515の“VRef”を基準として“CAN H”を測定している．

す．‘1’が5ビット以上連続している場合は，‘0’が挿入されます．

スタッフ・ビットの役割は，データ・ビットの同期ずれを補正する際に，補正ができない時間を短くするためのものです(連続した状態が長く続くとその間，同期ずれの補正ができない)．

CANではDPLL(Digital Phase Lock Loop) という位相補正の仕組みがあり，信号の状態がリセッシブからドミナントへ変化した時に位相ずれの補正機構が働き始めるようになっています．したがって，同じ状態のビットが長い間連続すると，この変化点がなかなか検出できないため，補正できない時間も長くなって位相ずれが大きくなってしまい，ついには補正できなくなる恐れがあります．

データの内容によっては‘0’または‘1’が6ビット以上連続するということもあるわけですから，その場合にダミーのビットを挿入して連続個所をなくす，というのがビット・スタッフィングの役目です．DPLLの働きで，同期ずれは頻繁に補正されながら通信しているため，ビット・レートの多少の食い違い(発振子の精度による周波数のばらつき)をカバーでき，信頼性が向上します．

このスタッフ・ビットはコントローラにより自動的に挿入され，受信時に自動的に取り除かれる(データ・ビットとして無視される)ため，通常のアプリケーションでは意識する必要はありません．

図1-7(p.20)は実際にCANトランシーバMCP2515同士で通信した際にロジアナ(ロジック・アナライザ)で測定した波形に，各フレームの説明を書き加えたものです(**写真1-1**)．

この例では，2か所で‘0’のビットが5ビット以上連続しているため，自動的にビット・スタッフィングが発生しています．

もし，6ビット以上同一レベルの信号を受け取ると，本来あるべきはずのスタッフ・ビットが検出できないということで，スタッフ・エラーとなります．

● CANのバス・エラー

CANでは，運用中に発生するエラーに次のようなものがあります．

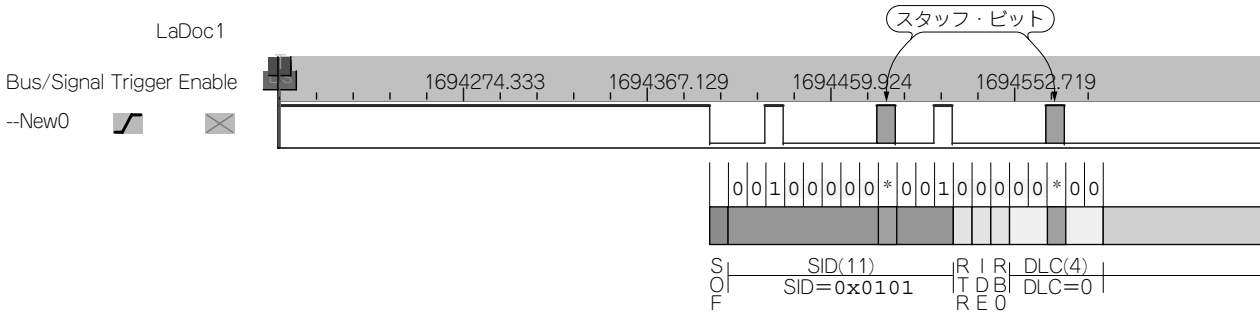


図1-7　ロジアナ実測波形
データ・フィールドがない(DLC＝‘0’)場合のデータ・フレームの測定波形．2か所でビット・スタッフィングが発生している．括弧内の数字はフィールドのビット数．ACKが“L”にドライブされているため，受信側がメッセージを正常に受信したことが確認できる．

(1) CRCエラー(受信側)

受信したCRC値と受信データから再計算したCRC値が異なる場合に発生します．エラーを検出すると，エラー・フレームが送信されるので，送信側はフレームを再送しなくてはなりません．

(2) ACKエラー(送信側)

受信側ノードがACKを返さない(ノードが存在しない場合も含む)ときに発生します．エラーを検出すると，エラー・フレームが送信されるので，送信側はフレームを再送しなくてはなりません．

(3) フォーム・エラー(受信側)

EOF，IFS，CRCDEL，ACKDELでドミナントを検出した場合に発生します．エラーを検出すると，エラー・フレームが送信されるので，送信側はフレームを再送しなくてはなりません．

(4) ビット・エラー(送信側)

アービトレーション・フィールドとACKスロットを除き，自分が出力したビット状態と実際のバスの状態が異なるときに発生します(ドミナントを出力しているのに実際のバス状態がリセッシブになっているとき，またはその逆のとき)．

ノイズの混入やバスが断線しているとき，短絡しているときなどに起こる可能性があります．

(5) スタッフ・エラー(受信側)

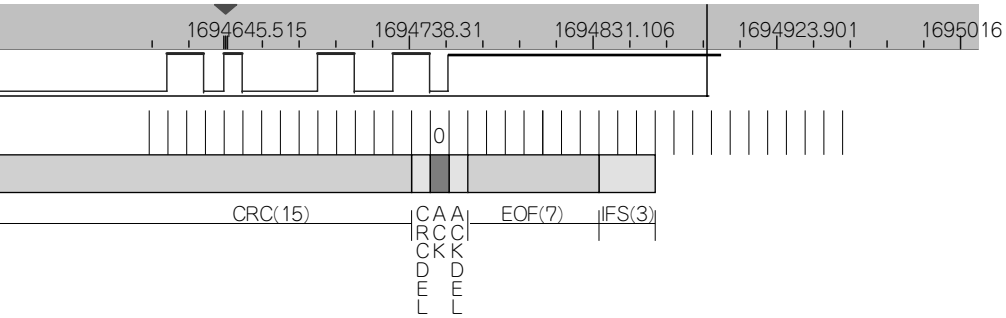
SOFからCRCDELの間で5ビット以上連続して同一のデータ・ビットが送信されているにも関わらず，スタッフ・ビットがない場合に発生します．エラーを検出すると，エラー・フレームが送信されるので，送信側はフレームを再送しなくてはなりません．

● エラー・ステート

CANでは運用中に発生するエラーの程度を累積して評価する仕組みがあり，エラーの程度を数値としてカウントするためのカウンタが用意されています．このカウント値により，いくつかのエラー・ステート(状態)に分類されます．このステートはエラー発生状況の深刻度を表すものです．

PICマイコンの18Fxx8や本書で利用するCANトランシーバMCP2515などでは，

- 送信エラーをカウントするTEC(Transmit Error Counter)



●受信エラーをカウントするREC(Receive Ereor Counter)

の二つの8ビット・カウンタが用意されていて，それぞれ，通常は，エラーが発生するたび(エラー・フレームが送信されるたび)に+8されます．ただし，送信しようとしても受信ノードが存在しない場合は，いきなり+128されるというようなこともあります(**リスト2-1** 参照)．

正常な運用状態では，エラー・アクティブというステートにありますが，エラー・カウンタの値が127を超えるとエラー・パッシブというステートに移行し，さらに255に達するとバス・オフというステートに移行します．なお，正常に送受信できた場合は，それぞれのカウンタが−1され，エラーの重症度が減ります．したがって，正常な通信が続けばカウンタ値はどんどん小さくなります(最小0)．これら三つの状態をまとめると次のようになります．

(1) エラー・アクティブ

REC，TECとも128未満の正常な状態．とくに96以上128未満はエラー・アクティブのウォーニング・ステートに分類される．

(2) エラー・パッシブ

RECまたはTECが128以上で255未満の中〜高程度の異常な状態．復旧可能だが，かなり重症．

(3) バス・オフ

TECが255の最悪な状態．この状態では通信不能で，特定の手順(デバイスのリセットなど)を踏まないと復旧不可能．

[第2章]

CAN専用コントローラを用いて各種マイコンから活用を図る

CANコントローラMCP2515の機能

本章では，本書でのCAN通信の要となるCANコントローラMCP2515について詳しく説明します．このコントローラさえ使えるようになれば，多様なCPU用の制御プログラムにも応用ができるようになります．

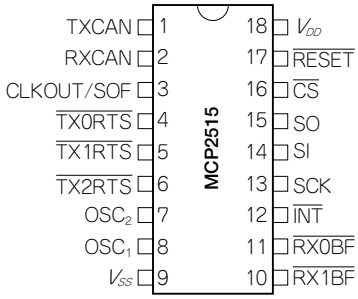
2-1 CANコントローラの概要

● MCP2515 CANコントローラの概要

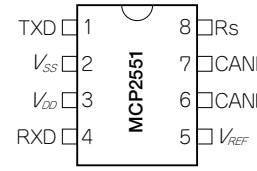
マイクロチップ社のMCP2515はSPI(Serial Peripheral Interface)通信で操作することできるCANコントローラです．CAN V2.0Bに対応しています．CANトランシーバMCP2551と組み合わせて簡単にCANノードを作ることができます．**図2-1**にMCP2515のピン配列を示します．DIPパッケージも用意されているため，ブレッドボードなどでも使用できます．また，フラット・パッケージを含めた外観を**写真2-1**に示します．

おもな仕様

- CAN V2.0B準拠 最高ビットレート 1Mbps
- メッセージ長は11ビット，29ビット両対応



(a) CANコントローラ MCP2515(DIP/SOIC) ピンアウト(*1)



(b) CANトランシーバ MCP2551(DIP/SOIC) ピンアウト(*2)

図2-1(1),(2) CANデバイスのピン・アウト
本書で使用するCANコントローラ(MCP2515)とCANトランシーバ(MCP2551)のピン配列を示す．MCP2515のTSSOPパッケージは20ピンで一部ピン配置が異なるので注意．

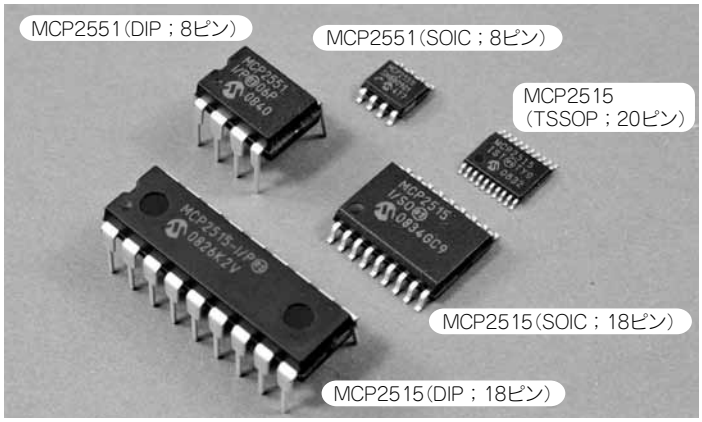


写真2-1　CANコントローラIC
マイクロチップ社のCANコントローラ(MCP2515)，CANトランシーバ(MCP2551)の外観を示す．MCP2515のTSSOPのピン数，ピン配列は，DIPとSOICのものとは異なるので注意．

- ▶二つの受信バッファをもつ
- ▶三つの送信バッファをもつ
- ▶六つの29ビット・フィルタと二つの29ビット・マスクをもつ
- ▶三つのRTS信号入力ピンあり
- ▶二つのRXBUF信号出力ピンあり
- ▶ハイ・スピードSPI（10MHz）
- ▶クロック出力端子あり（ホスト・マイコンのクロックなどに利用可能）
- ▶送受信タイミングなど多様な割り込み発生機能あり
- ▶ロー・パワーCMOSテクノロジで動作電圧は2.7V～5.5V動作時消費電力 5mA（標準）
- ▶動作保証温度－40℃～+85℃（産業用）
- ▶スリープ・モードあり データ受信などのイベントでウェイクアップ可能

このコントローラは，PIC18Fxx8などに内蔵されているECANモジュールのレガシ・モード相当の機能があるものと思われます．

レジスタのアクセスの際は，PICがもつSPI通信でインストラクションやデータを送受信するため，特定の手順が必要ですが，その反面，SPIさえ備えていれば，接続先を選ばないという利点があります．

■ RTS*n*信号（*n*=0～2）

MCP2515には三つのRTS信号入力があります．この三つはそれぞれ三つの送信バッファに対応していて，ハードウェアにより送信を開始させるトリガ信号となっています．この機能を利用しない場合は，切り替えにより単純な汎用の入力ポートとしても使えます．

■ RX*n*BUF信号（*n*=0,1）

MCP2515には二つのRXBUF出力信号があります．この二つはそれぞれ二つの受信バッファに対応していて，メッセージが受信されたタイミングをハードウェアで検出できるようになっています．割り込み信号として利用できます．この機能を利用しない場合は，切り替えにより単純な汎用の出力

表2-1　ビットレート一覧表
CANの1ビットが16*TQ*の場合のBRP設定値とビットレートの一覧表．
TQ = 2 ＊（BRP＋1）/ F_{OSC}の関係がある．

発振子周波数（MHz）	BRP 設定値	*TQ*（ns）	CAN ビットレート（kHz）
20	0	100.0	625.0
	1	200.0	312.5
	2	300.0	208.3
	3	400.0	156.3
	4	500.0	125.0
	5	600.0	104.2
	6	700.0	89.3
16	0	125.0	500.0
	1	250.0	250.0
	2	375.0	166.7
	3	500.0	125.0
	4	625.0	100.0
	5	750.0	83.3
	6	875.0	71.4
12	0	166.7	375.0
	1	333.3	187.5
	2	500.0	125.0
	3	666.7	93.8
	4	833.3	75.0
	5	1000.0	62.5
	6	1166.7	53.6
8	0	250.0	250.0
	1	500.0	125.0
	2	750.0	83.3
	3	1000.0	62.5
	4	1250.0	50.0
	5	1500.0	41.7
	6	1750.0	35.7

ポートとしても使えます．

本書の製作例では，RX0BFにLEDを接続して，SPI通信の動作確認に利用している場合があります．

■ リセット信号

リセット回路は外部に用意する必要があります．また，SPI経由のリセット・コマンド送出によってもリセットすることができます．リセット回路は*CR*による一般的なものですが，実際のリセット回路は，製作例の回路図を参照してください．半導体メーカ各社から発売されているリセットICを使ってもかまいません．

■ クロック

ボーレートが低い場合は，発振子には多少精度が劣るレゾネータ（セラミック発振子）が利用できます．ボーレートが高い場合にはクリスタルを使う必要があります．発振子を使わずにオシレータのクロック信号を直接入力することも可能です．発振子周波数とビットレート，パラメータの設定の組み合わせの一部を**表2-1**に示します．

CLKOUTピンからクロック信号を出力することができます．このクロックは内蔵のプリスケーラで1/1，1/2，1/4，1/8のいずれかに分周して出力することができます．マイコンのクロックとして利用することも可能です．パワー・オン直後，リセット時には1/8に設定されるので，必要に応じてプリスケーラを再設定する必要があります．

● 割り込みとその要因(エラー，非エラー)

MCP2515には大きく分けて8種類の割り込み要因があります．割り込みを使わない場合でも，要因フラグのチェックやクリアが必要なものがあります．

割り込みが許可された状態で割り込みが発生すると，INTピンが“L”レベルに設定されホストへ知らせます．この状態は，ソフトウェアによって割り込み要因がクリアされるまで維持されます．

割り込みを使う場合は，このINT信号をマイコンの割り込み入力ピンに接続しておき，割り込み処理で，MCP2515から要因を読み出して処理を実行したあと，割り込み要因をクリアします．

割り込み要因はSPIコマンドでCANINTFレジスタを読み出すことで特定できます．また，割り込み状態をクリアする場合もSPIコマンドで同レジスタの該当ビットに‘0’を設定することでクリアできます．

なお，割り込みを使う場合は，あらかじめCANINTEレジスタで要因ごとに割り込みを許可しておく必要があります．割り込みを使わない場合は，CANINTFレジスタをポーリングするなどして読み出すことで，割り込み要因を知ることもできます．

なお，言うまでもありませんが，マイコン側でも割り込みを受け付けるような処理が必要です．PICの場合は，通常はINTn(RB_0～)割り込みを使うことになると思います．

次に，割り込み要因をCANINTFレジスタのビット構成にそって簡単に説明します．次のような要因があります．

- MERRE……メッセージ・エラー(メッセージの送受信中に発生するエラー)が発生
- WAKIF……スリープ状態からウェイクアップした
- ERRIF……エラー割り込み発生(複数要因あり，内容はEFLGレジスタに格納)
- TX2IF……送信バッファ2「空」
- TX1IF……送信バッファ1「空」
- TX0IF……送信バッファ0「空」
- RX1IF……受信バッファ1「フル」(受信データあり)
- RX0IF……受信バッファ0「フル」(受信データあり)

RXnIF受信バッファn「フル」は，受信データがバッファにあるときにセットされますが，このフラグは割り込み使用の有無に関わらず，受信データを取り出したあと，必ずクリアしておく必要があります．これを怠ると，次にメッセージを受信しようとしたときにオーバフロー・エラーが発生します．

TXnIF送信バッファn「空」は，送信バッファが利用可能になったことを知るためのものですが，このフラグは割り込みを使用していない場合は，クリアしなくても動作に支障はありません(クリアしておかないといつ空になったかわからないが)．割り込みを許可している場合はもちろん，クリア