

CAN-LINゲートウェイのソフトウェア・モジュールの実装

—高い信頼性を確保して異なるプロトコル通信間をつなぐ

笹 伸幸, 木下秀昭

自動車に搭載されるネットワークの規格としてCAN (Controller Area Network)が普及してきた。また、CANほどの高速性を求められないミラー制御やウィンドウ制御といったボディ系のネットワークには、低コストで通信できるLIN (Local Interconnect Network)が採用されつつある。この2種類のネットワークはプロトコルなどが異なるため、1台の自動車に組み込むにはゲートウェイ装置が必要となる。ここでは、実際に開発された試作機を例に挙げながら、ゲートウェイの設計について解説する。(編集部)

今から10年ほど前、パソコン上でTCP/IP (transmission control protocol/internet protocol)が動くようになり、「ネットワーク」ということばが市民権を得はじめました(そのころ、パソコンの世界はまだDOSの時代だった)。MosaicやNetscapeが登場したのもこのころです。

一方、組み込み機器の分野では、開発にC言語を導入することすら珍しい状況で、ましてOSやネットワークなどを利用するなんてまだまだ遠い世界のお話でした。

そのような中、自動車に搭載される電子機器は、市場や自動車メーカーからの信頼性や多機能性に対する高い要求に対応するべく驚異的な進歩をとげました。現在、さらに高度化しつつあるのは周知のことです^{(1),(2)}。今や車載機器の開発においてネットワークはあたりまえの技術であり、CAN (Controller Area Network)プロトコルは車載ネットワークのデファクト・スタンダードとなっています。

1999年にはLIN (Local Interconnect Network)という新しい通信プロトコルが公開されました。CANほどの速度を必要としない通信を、低コストで実現することを目的としています。昨年(2003年)秋にはバージョン2.0が公開され、日本でも開発案件が急増しています。

● CANは「マルチマスタ」、LINは「シングル・マスタ」

CANとLINについて、ここで簡単におさらいしておきましょう。

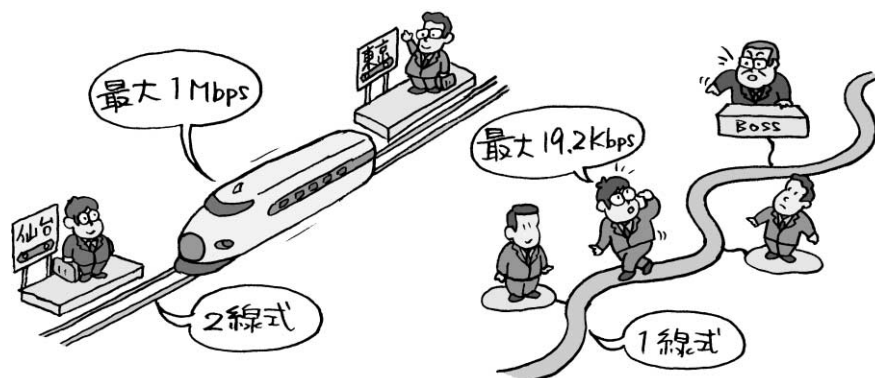
CANはデータ転送速度が最大1Mbpsと規定されている高速な通信プロトコルです(図1)。物理層(実際の電線)は2線式^{注1}で(グラウンドを含めると3線)、通信距離は最大

注1: 低速領域では1線式の仕様もある。

図1

CANとLINの違い

CANはデータ転送速度が最大1Mbpsで、バスが開放されていれば、どのノードも平等にデータをネットワークに送信できる(マルチマスタ)。LINはCANと比べて低速で、一つのマスタが複数のスレーブを管理する。マスタの許可なしにスレーブは通信できない。



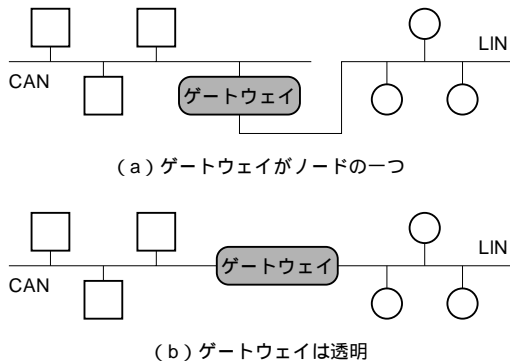


図2 ゲートウェイのタイプ

(a)では、ゲートウェイとそれにつながるLINクラスタが一つのCANノードとして機能する。(b)の場合、ゲートウェイにつながるLINノードがCANノードであるかのように見える。

1kmです。すべてのノードは平等の立場であり、どのノードもマスタになることができます。

一方、LINのデータ転送速度は最大19.2kbpsです。基本的には、UART通信をもとにしたシンプルなプロトコルです。物理層は1線で半二重通信です。一つの決まったマスタが複数のスレーブを管理します。

CANとLINはそれぞれ独立に存在するのではなく、一つの車の中で共存します。しかし、上述のように両者の通信プロトコルはかなり異なります。まるで、ことばも生活習慣も違う二人が一つ屋根の下に住んでいるようなイメージです。なんとかこの二人の仲を取り持ってやる必要があります。

本稿では、その役割を果たす「ゲートウェイ」について、設計事例を交えながら解説していきます。なお、今回のゲートウェイでは、LIN側はLIN 1.3に準拠して設計しています。

● 信頼性や設計難易度はゲートウェイの接続方法に依存

ゲートウェイを設計するにあたって、まずネットワークの中でどのような位置づけにするか、つまりCANとLINからゲートウェイ装置がどう見えるようにするかを検討する必要があります。

CANでは基本的にすべてのノードは平等なので、ゲートウェイを接続する場合、図2のように、

- 1) ゲートウェイをCANノードの一つにする
- 2) ゲートウェイの存在を透明(透過的)にする

の2通りの方法が考えられます。

1)の場合、ゲートウェイは、そこにつながるLINクラスタ(ノードの集合)が実現している機能を持った一つの装置

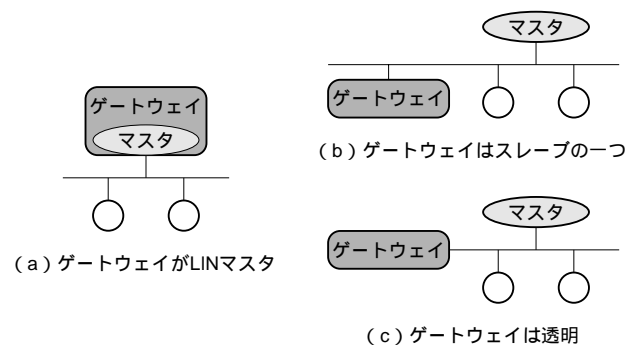


図3 ゲートウェイ内のLINの扱い

(a)ではゲートウェイがLINマスタとして、また、(b)ではLINのスレーブとして機能する。(c)の場合、ゲートウェイにつながるCANノードがLINのスレーブに見える。

として動作するように見えます。会社の代表電話のような感じでしょうか。一方、2)の方法ではゲートウェイの存在は消え失せ、LINノードがあたかもCANノードであるかのように見えることとなります。代表電話に対して内線電話へのダイレクト・インに似ています。

CANは欠陥ノードを切り離す機能を備えているので、ゲートウェイが故障したときなどの信頼性については、どちらの方法でも差はありません。

一方、LINでは1マスタ+複数スレーブのプロトコルになっているので、ゲートウェイについて図3のような三つの接続方法が考えられます。

- 1) ゲートウェイがLINマスタとして動作する
- 2) ゲートウェイがスレーブとして動作する
- 3) ゲートウェイの存在を透明にする

もっとも設計が簡単でコストがかからないのが、1)の方法です。使用するマイコンに必要な数の通信チャンネルとCPU性能があれば、マルチマスタ、すなわち一つのゲートウェイに複数のLINバスをつなげるような設計も容易です。

ただし、この方法ではゲートウェイが故障すると、そこにつながるLINクラスタ全体がダウンしてしまう危険性があります。もし、対象のLINクラスタがそれだけで機能的に閉じているような場合(例えば、スレーブ・ノードにスイッチとランプがあり、スイッチを入れればランプが点灯するといった機能)、信頼性について検討する必要があります。LINノードの機能がすべてCANからの指令でのみ従属的に動作するような場合、ゲートウェイが故障すればスレーブ・ノードが生きていても意味がないので、ほかの方法と信頼性の差はありません。