

組み込みシステム設計者のための LIN 2.0マイコン実装術



システムの記事

スレーブ・ノード用ドライバ・ソフトウェア開発のポイント

館 伸幸

前編(本誌 2005年 11月号, pp.92-97)と中編(同 12月号, pp.96-105)でLIN 2.0のプロトコル仕様について説明しました。今回は、このLIN 2.0プロトコルを実際にマイコンに実装するためのドライバ・ソフトウェアの設計について、事例を交えて解説します。(筆者)

前編で解説したように、LINは大きく分けて「マスタ・タスク」と「スレーブ・タスク」という二つの機能で動作します。マスタ・タスクは基本的にスケジュールに従ってヘッダを送信するのみであるのに対し、スレーブ・タスクはヘッダを受信し、その内容に従ってレスポンスを送信または受信しなければなりません(図1)。さらに、スレーブ・タスクが単体で搭載されるスレーブ・ノードは、コストを低く抑えるために非力なマイコンで実現することがほとんどです。また、RC発振器など、精度の低いクロックによ

る動作を補正する必要が生じる場合もあります。したがって、ドライバ・ソフトウェアの設計にあたっては、マイコンの機能をうまく使いこなす工夫を施さなければなりません。

一方、マスタ・タスクは、LINプロトコルの実装よりは、むしろゲートウェイとしての動作や、OS(またはスケジューラ)との連携といったところに設計のポイントがあります。実際の動作環境ではLINクラスタが単独で動作することは少なく、マスタ・ノードは別の基幹回線(例えば、CANなど)とのゲートウェイとしての動作が主であるためです(図2)。

本稿では、LINプロトコルを実装するうえで、マイコンのハードウェア制御に関する技術的ポイントが多く、また実際にノードを設計する機会がより多いスレーブ・ノード(タスク)について解説します。

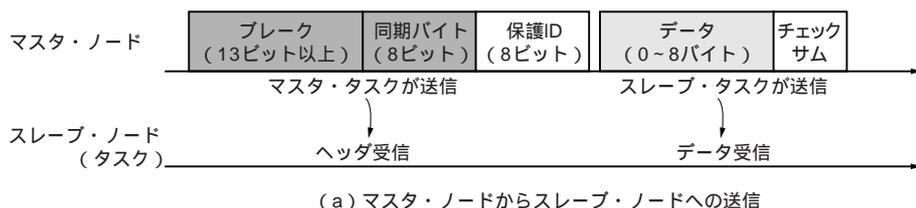


図1 マスタ・スレーブ間通信

マスタ・タスクがヘッダを送信して、スレーブ・タスクがそれを受信することで、後に続くデータの送受信を行う。

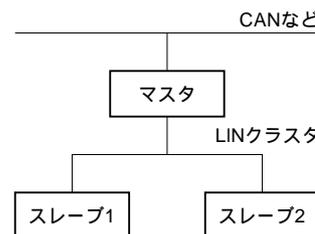


図2 実際の使用環境の例

とくに車載用途では、LINクラスタはCANのノードとして動作することがほとんどである。

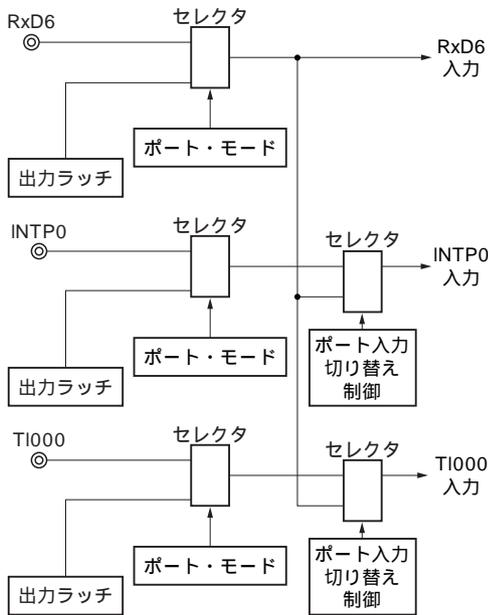


図3 LIN 受信操作作用ポートの構成(概略図)

ポート入力切り替え制御によって、マイコンの端子を結線することなく、受信ポート(RxD6)の入力データを外部割り込み(INTP0)や16ビット・タイマイベント・カウンタ(TI000)へ接続できる。

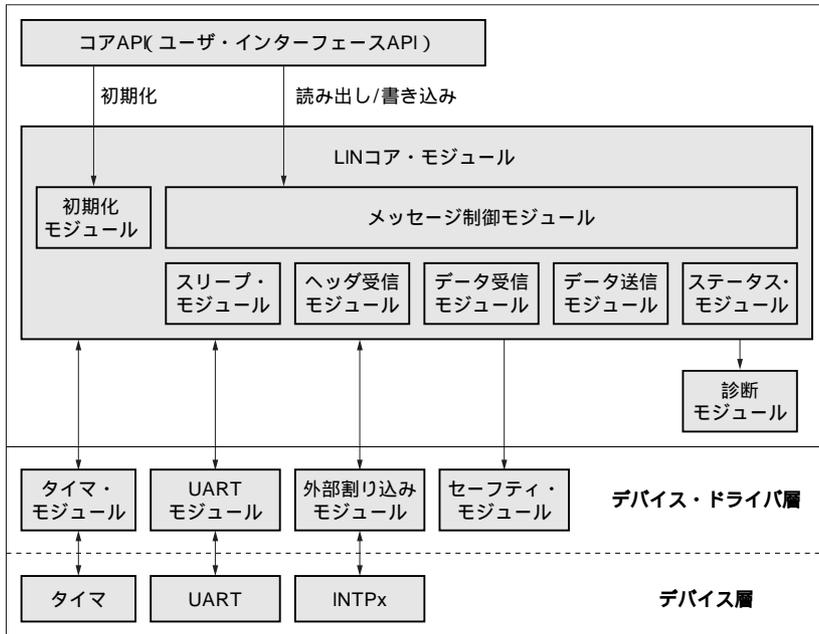


図4 プログラムの構造

移植性、拡張性を考慮して、階層化と部品化を行っている。

1. ノードの構成と状態遷移

今回、ノードの開発に使用したのは、NECエレクトロニクスの8ビット・マイコン「 μ PD78F8006 (78K0/Kx1, Kx2シリーズ)」です。UARTにLINのブレーク送受信を行う回路が含まれています(図3)。また、LIN仕様に適合した回線ドライバ回路も内蔵しているため、写真1のように小さなノード基板を作ることができます。

本稿では、ブレーク受信について、専用回路を使わない方法も記載します。ほかのマイコンを利用する場合は機能名などを随時読み替えて参照してください。

● 機能ごとにモジュール化する

図4に、LINドライバ(ソフトウェア)の構成例を示します。最下層はマイコンのハードウェアです。タイマ、UART、外部割り込み(INTPx)を使います。用途はそれぞれ、次のとおりです。

- タイマ：各種時間規定の実現
- UART：ヘッダの一部の受信とレスポンスの送受信
- INTPx：ブレークの検出、同期バイトの計測、ウェイクアップの検出

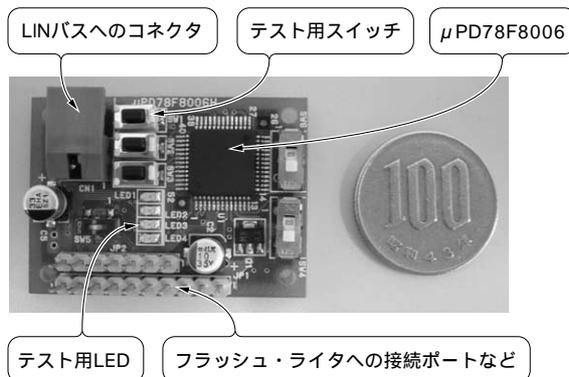


写真1 LINスリープ・ノード基板の例

下から2層目には物理リソースのドライバを配置しています。この層でマイコンごとのハードウェア仕様の差異を吸収することにより、移植性を確保しています。

LINプロトコルそのものはLINコア・モジュールという部品で実現しています。このモジュールでは、LINスリープ・タスクの基本機能であるヘッダ受信、データ(レスポンス)送受信、スリープ制御などを行います。メッセージ制御モジュールが、これらのモジュールを制御して実際の通信を行います。

そのほか、初期化および全体の状態管理を行うステータス・モジュールを備えています。診断モジュールは、クラ