AVR322 : ATMEL AVRマイクロ コントローラでの LIN1.3版規約実装

8ビット AVR® マイクロ コントローラ

応用記述

制御器域網(CAN:Controller Area Network)は車体と動力伝達装置制御系に於いて大層成功しました。けれども、簡単なﾃﾞｼﾞﾀﾙ感知器/作動部の車体制御応用に関してCANが強力で高価すぎるように見える自動車産業で起きている重要な変化があります。明らかになっている局所相互連結網(LIN:Local Interconnect Network)仕様1.3規約はこのような応用のために提案されました。LINはUSARTやUSI周辺機能を持つ最下位ﾏｲｸﾛ ｺﾝﾄﾛｰﾗ向け仕様の簡単な規約です。LIN規約はATMELの車載AVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗでの実装と共に、この応用記述で紹介されます。USART実装はLIN主装置と従装置のﾄﾞﾗｲﾊﾞに対して記述され、そのCｿｰｽ ｺｰﾄﾞはATMELから入手できます。

車内網応用

今日、電子部品とそのｼｽﾃﾑは最上級乗客車両の費用の20%以上と思われます。幸運にも車体と動力伝達機構制御系に対して、車内網は既に適用されています。電子制御駆動やｲﾝﾀｰﾈｯﾄへのｱｸｾｽを提供するﾏﾙﾁﾒﾃﾞｨｱ ｼｽﾃﾑのように、もっと複雑な制御系の更なる導入で、この図式は次の10年に渡って重要になることが予測されます。車内制御網技術に関する限り、明らかな成長分野はCANを補完する低価格車体制御補助ﾊﾞｽの採用に於いてです。

多くの車体制御機能は度々、照明、ﾜｲﾊﾟｰ、窓などのように簡単なﾃﾞｼﾞﾀﾙON/OFF操作です。これらはCANで提供され得る応答の必要性が要求されない、緩い必要条件の実時間系と見做せます。加えて、これらはその産業に於いて感知器や作動部内に網節点(ﾉｰﾄﾞ)を統合することも求めます。これがCANを補完する技術であるけれども、よりもっと安価であるため、LINはこれらの応用に関する現在の技術候補です。表1.はCANとLINの規約を簡単に比較します。

表1. CANとLINの主な機能の比較

信号	CAN	LIN
起源/帰属	ﾛﾊﾞｰﾄ ﾎﾞｯｼｭ(Robert Bosch®) GmbH (www.cn.bosch.com) CAN in Automation (www.can-cia.com)	LIN協会 (www.linsubbus.org)
最大ﾋﾞｯﾄ速度	1Mbps – 高速CAN ISO-11898	20kbps 強化ISO-9141(ISO-K)
網形態	ﾊﾞｽ	ﾊﾞｽ
網ｱｸｾｽ法	非破壊ﾋﾞｯﾄ単位調停	主局従局(調停不要)
節点(ﾉｰﾄﾞ)数	通常、物理層での128 または上位層での64	16 (主装置=1,従装置=15)
線数	2	1

LIN応用

LIN規約は以下のような応用に於いてCANを補完するための安価な補助ﾊﾞｽ系として1999年に提案されました。

- 車の天井 (雨感知器、光感知器、照明制御、ｻﾝﾙｰﾌ)
- 車の扉 (扉鏡、室内鏡、鏡ｽｲｯﾁ、窓昇降)
- ｴﾝｼﾞﾝ (感知器、小ﾓｰﾀ、ﾊﾝﾄﾞﾙ、操縦制御ｽｲｯﾁ、ﾜｲﾊﾟｰ、方向指示器、ﾗｼﾞｵ、環境制御)
- 座席 (座席位置ﾓｰﾀ、座席ﾋｰﾀ、占拠感知器)

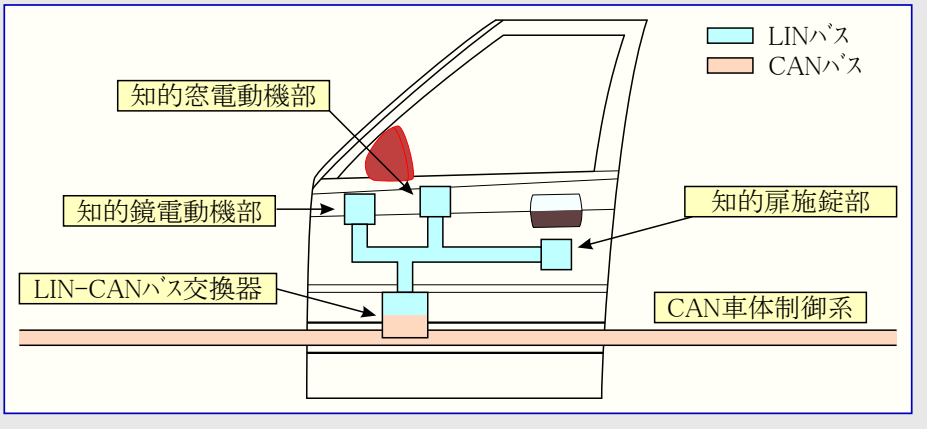
LINの主な機能は以下です。

- USARTまたはUSIに基く
- 物理層、強化ISO-9141
- 20kbpsまでのﾋﾞｯﾄ速度
- 16種のLIN ID、8ﾊﾞｲﾄまでのﾃﾞｰﾀ
- 主装置/従装置通信

Rev. 7548A-12/05, 7548AJ2-02/14

LINの使用は図1.で示されるように車体制御網基本構成数の増加を生じます。図1.は車の扉の電気的な補助系の制御に関するLIN補助ﾊﾞｽ例を示します。LIN-CANﾊﾞｽ交換器経由で扉系とその他(諸々)を持つ主車体制御CAN網ｲﾝﾀｰﾌｪｰｽを見ることができます。知的扉施錠部、知的鏡電動機部、知的窓電動機部に対してLIN-CANﾊﾞｽ交換器へ接続するのに単一LIN信号線が使用されます。これは合計3本の信号線に帰着し、そしてこれは伝統的な実装よりも最低5本少ないのです。従って喜ばしい費用節約は3つの追加最低位ﾏｲｸﾛ ｺﾝﾄﾛｰﾗの追加費用による配線差分に於ける削減から生じます。

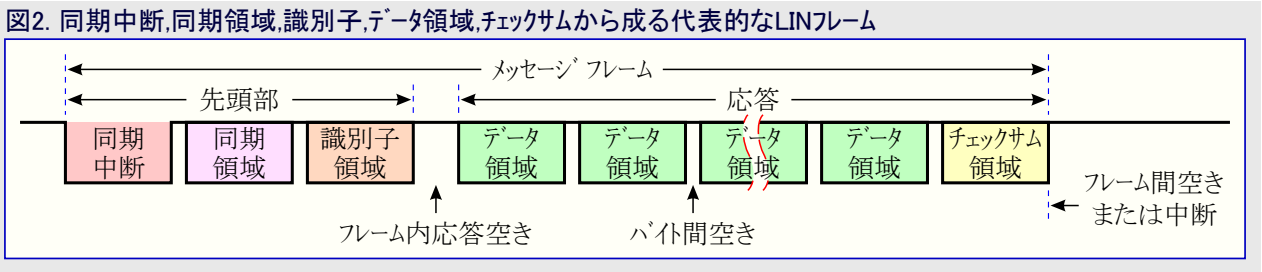
図1. 車扉制御の局所領域でのCANﾊﾞｽとLIN副ﾊﾞｽ組み合わせの代表的な使用



LIN規約1.3版

代表的なLINﾌﾚｰﾑは図2.で示され、以降の構成物から成ります。

図2. 同期中断,同期領域,識別子,ﾃﾞｰﾀ領域,ﾁｪｯｸｻﾑから成る代表的なLINﾌﾚｰﾑ



ﾃﾞｰﾀの各ﾊﾞｲﾄは10ﾋﾞｯﾄの標準USART形式(換言すると、開始ﾋﾞｯﾄ、8ﾃﾞｰﾀ ﾋﾞｯﾄ、停止ﾋﾞｯﾄ)に従います。

同期中断(LIN主作業)

- 最低13ﾋﾞｯﾄ - ﾌﾚｰﾑの開始を示します。

同期領域(LIN主作業)

- 1ﾋﾞｯﾄ - 同期分離子
- $55(開始ﾋﾞｯﾄと停止ﾋﾞｯﾄを伴う8ﾋﾞｯﾄ ﾃﾞｰﾀ) - 従装置での同期用

識別子領域(LIN主作業)

- 開始ﾋﾞｯﾄと停止ﾋﾞｯﾄを伴う8ﾋﾞｯﾄ ﾃﾞｰﾀ
- ﾋﾞｯﾄ0～3 - LIN ID
- ﾋﾞｯﾄ4と5 - 長さ制御
- ﾋﾞｯﾄ6と7 - ﾊﾟﾘﾃｨ

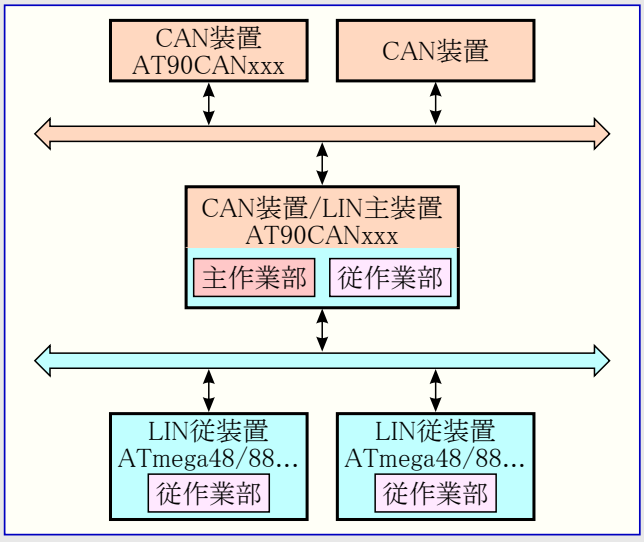
ﾃﾞｰﾀ領域(LIN従作業)

- 2,4,または8ﾊﾞｲﾄ(各々が開始ﾋﾞｯﾄと停止ﾋﾞｯﾄを伴う8ﾋﾞｯﾄ ﾃﾞｰﾀ)

ﾁｪｯｸｻﾑ 8ﾋﾞｯﾄ(LIN従作業)

通信の概念は図2.と図3.で示されるように1つの主作業部と多数の従作業部で成ります。主作業は同期中断、同期領域、識別子領域から成るﾒｯｾｰｼﾞ先頭部を送信します。従作業はﾃﾞｰﾀ領域とﾁｪｯｸｻﾑ領域から成るﾒｯｾｰｼﾞ応答を送信します。主作業は常に主節点(ﾉｰﾄﾞ)によって送信され、一方従作業はLIN副網の節点のどれかによって送信し得ます。主節点はﾃﾞｰﾀを送る必要がある時に従作業部へ送信し、ﾃﾞｰﾀを受信する必要がある時に従作業部経由で受信します。

図3. LIN副ﾊﾞｽ系でのAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗの代表的な応用



ﾒｯｾｰｼﾞ先頭部

同期中断

同期中断はLINﾌﾚｰﾑの始まりを示し、常に主作業部によって生成されます。同期中断は同期領域のための準備を従作業部へ告げます。同期中断は次の2つの要素です。

- 最小13ﾋﾞｯﾄ時間(主装置時間基準に基くTbit)であるべき優性(Low)ﾚﾍﾞﾙで、これは以下が後続します。
- 1～4ﾋﾞｯﾄ時間の範囲であるべき劣性(High)区間(同期分離子)

同期中断の長さは最低13ﾋﾞｯﾄ時間に選ばれ、このためにLIN従作業部は有効な同期中断とﾃﾞｰﾀ ﾌﾚｰﾑ内で許される可能な最大優性ﾋﾞｯﾄの連続(換言すると9ﾋﾞｯﾄ時間)を区別することができます。従装置は11ﾋﾞｯﾄ時間(従装置の局所時間基準に基くTbit)の優性ﾚﾍﾞﾙを受信することによって同期中断を検知します。

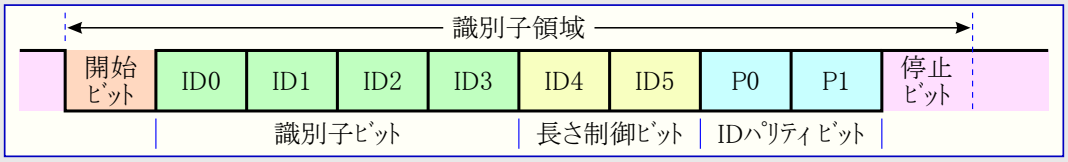
同期領域

同期領域は従装置が主装置と同期するために必要とする信号を含みます。同期領域は5つの上昇端と5つの下降端を持つ波形を生じる"$55"のﾃﾞｰﾀを含みます。これらのｴｯｼﾞは従節点(ﾉｰﾄﾞ)の送受信速度を主節点に合わせるための調整である同期中に使用することができます。従ってﾋﾞｯﾄ時間は5つの上昇端または5つの下降端のどちらかの時間を測定して8で割る(3ﾋﾞｯﾄの論理的移動)ことによって得られます。

識別子領域

識別子領域はﾒｯｾｰｼﾞの内容と長さについての情報を含みます。この領域は下の**図4.**で示されるように、識別子ﾋﾞｯﾄ(4ﾋﾞｯﾄ)、長さ制御ﾋﾞｯﾄ(2ﾋﾞｯﾄ)、ﾊﾟﾘﾃｨ ﾋﾞｯﾄ(2ﾋﾞｯﾄ)の3つの分野に分けられます。

図6. 識別子領域



識別子ﾋﾞｯﾄ(ID0～3)はﾌﾚｰﾑのﾒｯｾｰｼﾞ応答部で応答する節点の識別子を表します。従ってLIN副網では1つの主装置と15までの従装置で最大16節点までにできます。このため、識別子領域はﾒｯｾｰｼﾞの内容を記述し、行き先はありません。

識別子領域内の最後の2ﾋﾞｯﾄは混合ﾊﾟﾘﾃｨ法によって定義されたﾊﾟﾘﾃｨ ﾋﾞｯﾄです。これは識別子領域が全優性または全劣性の様式で決して構成されないことを保証します。このﾊﾟﾘﾃｨ検査が誤り検出だけでそれらを修正しないことに注意しなければなりません。

ﾊﾟﾘﾃｨ検査ﾋﾞｯﾄは以下の混合ﾊﾟﾘﾃｨ法によって計算されます。

```
P0 = ID0 XOR ID1 XOR ID2 XOR ID4
P1 = ID1 XOR ID3 XOR ID4 XOR ID5
```

表2. LIN1.3版の長さ制御による指示でのﾃﾞｰﾀ領域ﾊﾞｲﾄ数

ID5	ID4	N_{DATA}(ﾃﾞｰﾀ領域数)
0	0	2
0	1	2
1	0	4
1	1	8

ﾒｯｾｰｼﾞ応答

ﾃﾞｰﾀ領域

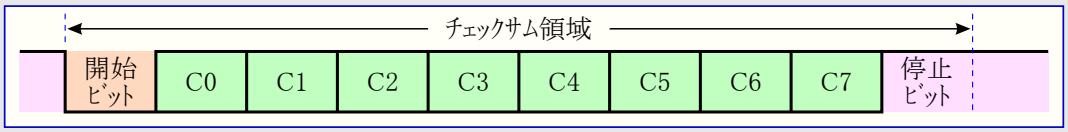
ﾃﾞｰﾀ領域は各々が1つの開始ﾋﾞｯﾄ、8つのﾃﾞｰﾀ ﾋﾞｯﾄ、1つの停止ﾋﾞｯﾄから成る、2、4、または8ﾊﾞｲﾄを含みます。送信はLSB先行で行われます。ﾃﾞｰﾀ領域は従作業部の応答によって書か(送信さ)れます。ﾊﾞｽ調停がないので、1つの従作業部だけが各識別子に対して応答を許されるべきです。他の全ての従作業部は応答を読むこととそれによって動くことに制限されます。ﾃﾞｰﾀ領域内の単一ﾊﾞｲﾄ形式が**図5.**で示されます。

図5. ﾃﾞｰﾀ領域の単一ﾊﾞｲﾄ

チェックサム領域

メッセージ フレームの最後の領域はチェックサム領域です。このバイトはデータ領域内の全データ バイトの総和の(論理)反転した256の剰余を含みます。この総和は全データ バイトに於いて"キャリーを含めた加算"を行い、そして結果を(論理)反転することによって計算されます。この反転した総和の256の剰余の特性は、この数値が全データ バイトの総和に加算された場合の結果が"$FF"のようになることです。チェックサム領域の形式は図6.で示されます。

図8. チェックサム領域



"キャリーを含めた加算"操作での各バイトの加算中に溢れ(キャリー)ビットがLSBに加えられることが重要で、このために次のようになります。

$FF+$01=$01

誤り検出

LIN規約1.3版に於いて以下の誤り検出法が概説されています。

ビット誤り検出はLINバスにビットを送出する何れかの装置に関係します。ビット誤りは監視したビットが送ったビット値と異なる時に発生します。これはデータを送信した主節点(ノード)または従節点のどちらかに責任があります。

直前で記述されたようなデータ領域のチェックサム誤り検出。これはデータ領域内のデータを受信した主節点または従節点のどちらかに責任があります。

先に記述したような識別子領域に対する識別子パリティ誤り検出。これはメッセージ先頭部を受信する従節点に責任があります。

従装置無応答異常検出は従装置のメッセージ応答が主装置のメッセージ先頭部に応答しない時の確認に関係します。これはLIN仕様1.3版に於いてフレーム開始(換言すると、同期中断の開始)後TFRAME_MAX内にその転送が完了しないフレームとして定義されます。これは次のように算定されます。

- 2バイトのデータ領域=91ビット時間
- 4バイトのデータ領域=119ビット時間
- 8バイトのデータ領域=174ビット時間

これは特定のメッセージ応答の受信を期待する主節点または従節点に責任があります。

物理的なバス異常検出は有効なメッセージを全く生成することができない、換言すると、バスがGNDに短絡している場合で、主装置によって処理されます。

同期領域矛盾異常検出は同期領域のエッジが許容外の時の確認に関係します。これは従節点だけに責任があります。

データ領域での不正なデータ バイト数の受信のように、LIN仕様で網羅されていない異常状況もあります。少な過ぎるデータ バイトの受信は従装置無応答異常に帰着するでしょう。多過ぎるデータ バイトはチェックサム誤りで終わるでしょう。

異常の合図と回復時間

主/従の概念のため、直接的な異常の合図は不可能です。異常は局所的に検出され、要求に於いて診断の形式で提供されるべきです。

障害の限度

LIN仕様は異常検出の(全てでない場合の)殆ど、異常回復、診断を主節点が処理すべきと指定しています。障害の限度はシステムの必要条件に依存し、従って基本的な支援だけが指定されています。故に障害の限度はそれらの応用の特定の必要性を使用者によって指定される必要があります。

主制御部の障害限度

主装置は次の異常状況を検知すべきです。

- 主作業部に於けるビット誤りまたは識別子パリティ誤り
- 主制御部内の従作業部⇒従装置無応答異常とチェックサム誤り

主制御部障害限度はLIN1.3版で指定されていません。けれども推奨策が略述されています。推奨策は主作業部のデータ送出は同期または識別子の領域に於いてビット誤りを検出しなければならないと述べてます。主制御部は主送信異常計数器(MasterTransmit ErrorCounter)を増加することによってどの送信異常の経緯も保つべきです。これは同期または識別子の領域内のビットが局所的に不正にされる毎に8増加されます。そして両領域が正しく読み戻せる毎に1減少されます(0未満にはしない)。この計数器がC_MASTER_TRANSMIT_ERROR_THRESHOLD(推奨既定値=64)を越えて増加される場合、LINバスに於いて重大な妨害が起きたと仮定し、応用コード レベルで異常処理手順が取られるべきです。

主制御部送出ﾃﾞｰﾀのﾒｯｾｰｼﾞ応答(換言すると、ﾃﾞｰﾀとﾁｪｯｸｻﾑの領域)はﾋﾞｯﾄ誤りを検知するでしょう。主制御部の従作業部が受信するﾃﾞｰﾀは従装置無応答異常またはﾁｪｯｸｻﾑ誤りのどちらかを検知するでしょう。ﾋﾞｯﾄ誤り発生時、主送信異常計数器が先に記述したように増加されます。予期した時または受信ﾃﾞｰﾀで従装置無応答異常またはﾁｪｯｸｻﾑ誤りが検出されると、主受信異常計数器(MasterReceiveErrorCounter)が8増加されます。これは従装置応答が異常なしで正しく受信される毎に1減少されます(0未満にはしない)。この計数器がC_MASTER_ RECEIVE_ERROR_THRESHOLD(推奨既定値=64)を越えて増加される場合、ｱﾄﾞﾚｽ指定した従装置が機能不全にされたと仮定し、応用ｺｰﾄﾞ ﾚﾍﾞﾙで異常処理手順が行われるべきです。

従装置無応答異常(換言すると、主制御部によって検出された)発生時、LIN仕様1.3版は再送信が行われるべきか否かを指定していません。これは応用ｺｰﾄﾞ ﾚﾍﾞﾙ内で行われるべき判断です。

従制御部の障害限度

従節点は次の異常状況を検知すべきです。

- 従作業部経由のﾃﾞｰﾀ送信時のﾋﾞｯﾄ誤り検出
- 従作業部経由のﾃﾞｰﾀ受信時のﾁｪｯｸｻﾑ誤り
- 主節点からのﾒｯｾｰｼﾞ先頭部受信時の識別子ﾊﾟﾘﾃｨ誤り

これらの誤りに加えて、他の2つの異常が検出されるべきですが、これらは特定LINﾄﾞﾗｲﾊﾞの応用に依存します。

- 従節点がﾃﾞｰﾀは他の従節点から生じると予想する時に従装置無応答異常が検出されるべきです。
- 受信した同期領域が許容範囲外の時に同期領域矛盾異常が検出されるべきです。

従制御部障害限度はLIN1.3版で指定されていません。けれども主制御部の方法と同様の推奨策が略述されています。これは主制御部と同じ範囲で略述されていません。けれども主制御部のそれに基いて示唆される方法がここで提案されます。

提唱策は従制御部からのﾒｯｾｰｼﾞ応答送出ﾃﾞｰﾀはﾃﾞｰﾀとﾁｪｯｸｻﾑの領域に於いてﾋﾞｯﾄ誤りを検出しなければならないことです。従制御部は従送信異常計数器(SlaveTransmitErrorCounter)を増加することによってどの送信異常の経緯も保つべきです。これはﾃﾞｰﾀまたはﾁｪｯｸｻﾑの領域内のﾋﾞｯﾄが局所的に不正にされる毎に8増加されます。そして両領域が正しく読み戻せる毎に1減少されます(0未満にはしない)。この計数器がC_SLAVE_ TRANSMIT_ERROR_THRESHOLD(推奨既定値=64)を越えて増加される場合、LINﾊﾞｽに於いて重大な妨害が起きたと仮定し、応用ｺｰﾄﾞ ﾚﾍﾞﾙで異常処理手順が取られるべきです。

従制御部が受信するﾒｯｾｰｼﾞ応答ﾃﾞｰﾀは識別子ﾊﾟﾘﾃｨ誤り、従装置無応答異常、またはﾁｪｯｸｻﾑ誤りを検知するでしょう。予期した時または受信ﾃﾞｰﾀで従装置無応答異常またはﾁｪｯｸｻﾑ誤りが検出されると、従受信異常計数器(SlaveReceiveErrorCounter)が8増加されます。これは従装置応答が異常なしで正しく受信される毎に1減少されます(0未満にはしない)。従受信異常計数器がC_SLAVE_ RECEIVE_ERROR_THRESHOLD(推奨既定値=64)を越えて増加される場合、ｱﾄﾞﾚｽ指定した従装置が機能不全にされたと仮定し、応用ｺｰﾄﾞ ﾚﾍﾞﾙで異常処理手順が行われるべきです。

予約識別子

予約識別子は以下で略述されるような特定機能用に予約されています。

命令ﾌﾚｰﾑ識別子。主装置から従装置への役務目的のための広域一般命令要求に対して、2つの命令ﾌﾚｰﾑ識別子が予約されています。8ﾊﾞｲﾄを持つ各ﾌﾚｰﾑは、'$3C'の主装置要求ﾌﾚｰﾑと'$D7'の従装置要求ﾌﾚｰﾑでの予約識別子によって区別されます。主装置要求ﾌﾚｰﾑは命令とﾃﾞｰﾀを従節点へ送るのに使用されます。従装置要求ﾌﾚｰﾑは主節点へﾃﾞｰﾀを送るために(ﾀﾞｳﾝﾛｰﾄﾞ ﾌﾚｰﾑ先行によってｱﾄﾞﾚｽ指定された)従節点を起動します。

ﾃﾞｰﾀ領域先頭ﾊﾞｲﾄのMSB(D7)が0の場合、そのﾌﾚｰﾑの使用はLIN協会による定義用に予約され、さもなければその使用は独自定義用に開放されています。

休止形態命令。休止形態命令は全ての従節点を休止状態に置くための広域ﾒｯｾｰｼﾞです。休止形態命令は$00とするﾃﾞｰﾀ領域先頭ﾊﾞｲﾄを持つ主装置からの命令ﾌﾚｰﾑです。

拡張ﾌﾚｰﾑ。'$FE'とする識別子領域を持つﾌﾚｰﾑは使用者定義ﾒｯｾｰｼﾞ形式用に予約されています。'$BF'とする識別子領域を持つﾌﾚｰﾑはLIN協会による将来の定義用に予約されています。

ATMEL AVRマイクロ コントローラ

本資料の以降の部分はATMELのAVRマイクロ コントローラ用の現在のLIN1.3版実装を記述します。主と従の両節点が支援されます。下は現在支援されているMCUの一覧です。

MCU	使用周辺機能		備考
ATmega48/88/168	USART0		
AT90CAN128	USART0	USART1	2系統LIN利用可能
ATtiny45/85	USI		LIN従装置のみ

更に、最低1つの空きUSARTまたはUSIを持つどのAVRデバイスも、ソース コードの少しの変更(インクルード ファイル、レジスタ定義、タイマ/カウンタ使用法など)の後で支援されるかもしれません。この応用記述で掲載されるドライバはAVRマイクロ コントローラのUSART装置での実装だけにかかわります。

発振器の考慮

LIN応用に対しては内蔵RC発振器が重要な特徴です。これは外部クリスタルが必要とされず、従って費用節約に通じることを意味します。AVRマイクロ コントローラの内蔵発振器の精度はUSART LIN従装置実装に充分な±1%です(LIN仕様が要求する精度は再同期後で±1.5%です)。けれども、USART LIN主装置に関しては最低±0.5%の精度が必要とされるため、外部クリスタルでの発振器が使用されなければなりません。

更なる詳細はATMELのウェブサイトで入手できるAVRのデータシートで得られます。全てのソフトウェア ドライバ例は8MHz発振器に対するタイミング用の値を使用します。内蔵RC発振器の走行時校正のより多くの情報はATMELのウェブサイト(AVR504)で得られます。

ファイル一覧概要

以下は代表的な供給ソフトウェア1式に関する主なファイル一覧です。

- lin_drv_usart.c ：USART実装用ドライバ ソース ファイル
- lin_drv_usart.h ：USART実装用ドライバ ヘッダ ファイル
- lin_lib.c ：ライブラリ ソース ファイル
- lin_lib.h ：ライブラリ ヘッダ ファイル
- config.h ：一般形態設定ファイル (ボーレートとシステム クロック選択)

従装置LIN実装 (フレーム先頭部送信なし節点(ノード))

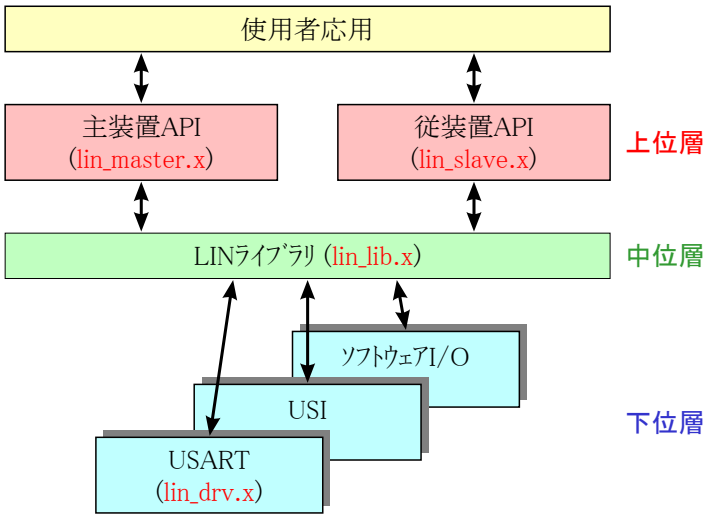
- slave_lin.c ：従装置管理部ファイル
- slave_lin.h ：従装置管理部ヘッダ ファイルと任意選択

主装置LIN実装 (フレーム先頭部送信能力あり)

- master_lin.c ：主装置管理部ファイル
- master_lin.h ：主装置管理部ヘッダ ファイル

ソフトウェア実装

配布された1式では3つのソフトウェア層が定義されています。



低位ﾄﾞﾗｲﾊﾞ "LIN_DRV.x" (必須)

通常、これらは特定周辺機能、換言すると、USART,USI,などに対して専用です。これらは使用者応用からの直接呼び出しを予定されていません。その代わりにこれらはLINﾗｲﾌﾞﾗﾘ(中位層)によって呼ばれます。ﾄﾞﾗｲﾊﾞ層は選択された"LIN"周辺機能の制御変更に於いて内部有限状態機構を通してLIN制御器状態を更新し、より上位の層へ異常/ﾊﾞｽ事象を報告します。主な作業は以下です。

- 同期中断検出
- 同期ﾊﾞｲﾄ復号/ｸﾛｯｸ同期化(従装置)
- 識別子復号、ﾃﾞｰﾀ長抽出、ﾊﾟﾘﾃｨ誤り検出
- ﾌﾚｰﾑ/ﾋﾞｯﾄ誤り検出
- ﾃﾞｰﾀの受信と送信
- ﾁｪｯｸｻﾑの生成/照合
- LIN制御器状態更新
- 時間超過検出

LINﾗｲﾌﾞﾗﾘ "LIN_LIB.x" (必須)

LINﾗｲﾌﾞﾗﾘ関数と変数は使用者応用から直接的に呼び出し/扱うことができます。代わりに提唱APIの使用も可能です。ﾗｲﾌﾞﾗﾘは独立した実装、換言すると、それはUSART,USI,などに対して同じです。主な作業は以下です。

- LIN制御器初期化(ﾎﾞｰﾚｰﾄ,など)
- 起動/休止形態処理
- 使用者命令"解釈部"
- 制御器状態

LIN API "SLAVE/MASTER_LIN.x" (任意)

更に、使用者の作業をより容易にするため、ｿﾌﾄｳｪｱ1式内にLIN APIが追加されています。この層は任意で、削除、または他のどんな上位層によっても置換することができます。LINが時間起動規約(ｽｹｼﾞｭｰﾗ)として実行される時間の殆どなので、関連する時間起動関数が追加されています。

- ﾃﾞｰﾀ緩衝処理(応用緩衝部間でのﾃﾞｰﾀの取得、複写)
- LIN識別子一致判定
- 異常計数器
- 休止形態制御(ﾊﾞｽ活動なし、休止命令)
- LIN従制御器が常に新規ﾌﾚｰﾑ受け入れ準備可なことを保証

LINﾌﾚｰﾑ形式

2種類のﾌﾚｰﾑが考慮されています。

- "REMOTE_DATA_FRAME"は固有の網主節点(ﾉｰﾄﾞ)によって先頭部が送られ、そして現在の従節点によって応答が送信されるﾌﾚｰﾑに対する名称です。これはﾃﾞｰﾀ要求ﾌﾚｰﾑです。その場合、主節点は網に接続された特定従装置からの応答を期待します。固有のｱﾄﾞﾚｽで指定された従節点は用意されている応答で返答するでしょう。
- "STANDARD_DATA_FRAME"は主装置によって先頭が送信され、そして他の外部節点(主節点または他の従節点)によって応答が送信されるﾌﾚｰﾑの名称です。現在の従節点はその応答ﾃﾞｰﾀをただ捕獲するだけでしょう。

"REMOTE_DATA_FRAME"が現在の転送によってｱﾄﾞﾚｽ指定されない、従節点に対する"STANDARD_DATA_FRAME"として見ることができることに留意してください。

LINﾗｲﾌﾞﾗﾘ応用

基板宣言

選択したMCUに対応して異なるﾌｧｲﾙがｲﾝｸﾙｰﾄﾞされます。"board_example_xxxx.h"ﾌｧｲﾙは使用者によって変更または置換されなければなりません。

ﾗｲﾌﾞﾗﾘ任意選択

ｺｰﾄﾞ量を減らすために"master_lin.h"と"slave_lin.h"のﾍｯﾀﾞ ﾌｧｲﾙで以下の任意選択を許可/禁止することができます。

- #define _TIMOUT_DETECTION

定義されたなら、ﾌﾚｰﾑの先頭部と応答に対して時間超過検出が活動(有効)です。時間超過検出機能は1つの計時器が必要です。供給される1式ではﾀｲﾏ/ｶｳﾝﾀ2が使用されます。"timer2_lib.c"と"timer2ovf_isr.c"ﾌｧｲﾙがﾌﾟﾛｼﾞｪｸﾄに追加されるべきです。

- #define _SLEEP_TIMOUT_DETECTION

定義されたなら、休止ﾒｯｾｰｼﾞ ﾌﾚｰﾑが送信された後で休止形態が活性にされます。起動ﾌﾚｰﾑは送受信することができます。現在この機能を使用するには、利用可能な計時器を通して計時関数がﾋﾞｯﾄ時間間隔で呼び出されなければなりません。供給される1式では"timer1ovf_isr.c"ﾌｧｲﾙが例を示します。休止時間超過検出前の時間は"master_lin.h"と"slave_lin.h"ﾌｧｲﾙ内の"SLEEP_TIMOUT"値によって調整されます。

休止ﾒｯｾｰｼﾞ フレームが送られる、またはLIN網でLIN活動が全く検知されない時に、対応するLINSleepFlagまたはNoBusActivityFlagのﾌﾗｸﾞが'1'に切り替えられます。そして使用者が主関数で適切な休止形態を選択することは自由です。

AVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗでは5つの休止形態が利用可能です(より多くの情報についてはﾃﾞｰﾀｼｰﾄをご覧ください)。

- ｱｲﾄﾞﾙ動作
- A/D変換雑音低減動作
- ﾊﾟﾜｰﾀﾞｳﾝ動作
- ﾊﾟﾜｰｾｰﾌﾞ動作
- ｽﾀﾝﾊﾞｲ動作

以下の各種休止形態で活動するｸﾛｯｸ領域と起動元をご覧ください。

休止形態	動作ｸﾛｯｸ範囲					動作発振器		復帰起動要因 (割り込み)						
	clk CPU	clk FLASH	clk IO	clk ADC	clk ASY	主ｸﾛｯｸ供給元	ﾀｲﾏ用発振器	INT1 INT0 ﾋﾟﾝ変化	TWI ｱﾄﾞﾚｽ一致	ﾀｲﾏ/ｶｳﾝﾀ2	SPM EEPROM 操作可	A/D 変換完了	ｳｫｯﾁﾄﾞｯｸﾞ	その他 I/O
ｱｲﾄﾞﾙ			○	○	○	○	②	○	○	○	○	○	○	○
A/D変換雑音低減				○	○	○	②	③	○	②	○	○	○	
ﾊﾟﾜｰﾀﾞｳﾝ								③	○				○	
ﾊﾟﾜｰｾｰﾌﾞ					○		②	③	○	○			○	
ｽﾀﾝﾊﾞｲ(注1)						○		③	○				○	

注1: ｸﾛｯｸ元として外部ｸﾘｽﾀﾙ発振子またはｾﾗﾐｯｸ振動子が選択された場合です。
② ﾀｲﾏ/ｶｳﾝﾀ2非同期状態ﾚｼﾞｽﾀ(ASSR)の非同期ｸﾛｯｸ(AS2)ﾋﾞｯﾄが設定(1)された場合です。
③ INT1とINT0についてはﾚﾍﾞﾙ割り込みだけです。

- #define _COUNTING_ERRORS_ENABLE

定義されたなら、LIN異常が検出される毎に異常計数が更新されます。

"slave_lin.h"でだけ利用可能なもう1つの任意選択が以下です。

- #define _RUN_TIME_RC_CALIBRATION_ENABLE

定義されたなら、内蔵RC発振器の走行時校正が許可されます。この機能のために1つの捕獲入力が必要です(ATMELのｳｪﾌﾞｻｲﾄで「**AVR054:内蔵RC発振器の走行時校正**」を参照してください)。

ﾗｲﾌﾞﾗﾘ形態設定

最も重要なLIN任意選択は"config.h"ﾌｧｲﾙで形態設定されます。

- MCU選択は次のどれかにすることができます。
  - AT90CAN128 (2つのUSART)
  - ATmega48
  - ATmega88
  - ATmega168
- "FOSC"は公称ｼｽﾃﾑ周波数です。
  - これはkHzで定義されなければなりません。現在"8000"だけが支援されます。2つの供給元が利用可能です。
    - 内蔵RC発振器

      従節点だけが内蔵RC発振器(8MHz)で走行できます。温度/電源電圧の変化を補償するために走行時RC発振器校正ﾙｰﾁﾝが利用可能です(より多くの情報についてはAVR054応用記述を参照してください)。
    - 外部ｸﾘｽﾀﾙ用発振器

      主と従の両節点は安定なｸﾛｯｸのために外部ｸﾘｽﾀﾙを使用することができます。±0.5%の最大周波数変動を保証するために主節点に対して8MHz外部ｸﾘｽﾀﾙが推奨されます。
- "LIN_BAUDRATE"
  - LINﾋﾞｯﾄ速度(bps)です。最も一般的なLINﾎﾞｰﾚｰﾄ(19200/9600/4800/2400)だけが支援されます。他のﾎﾞｰﾚｰﾄについては、それを特定する必要があるので、"lin_bdr.h"に於いて#define値が使用者によって手動で入力されなければなりません。
- "LIN_CONFIG"
  - LIN制御器動作種別
    - 1：従制御器　　- 0：主制御器
- USART選択
  - "USE_UART0"はATmega48/88/168,AT90CAN128のUSART0がLINﾎﾟｰﾄとして使用される場合に定義されなければなりません。
  - "USE_UART1"はAT90CAN128のUSART1がLINﾎﾟｰﾄとして使用される場合に定義されなければなりません。

自身のLIN主装置構成法

LIN主装置を動かすために以下の各種段階に従わなければなりません。

- LINﾗｲﾌﾞﾗﾘ任意選択定義。"master_lin.h"ﾌｧｲﾙを調べてください。
- LINﾗｲﾌﾞﾗﾘ形態設定。"config.h"ﾌｧｲﾙを設定してください。
- ﾃﾞｰﾀ緩衝部宣言と初期化

  LIN応用で必要とする全ての緩衝部を設定してください。

```
U8 Buf_SET_SLAVE[8];
～
```

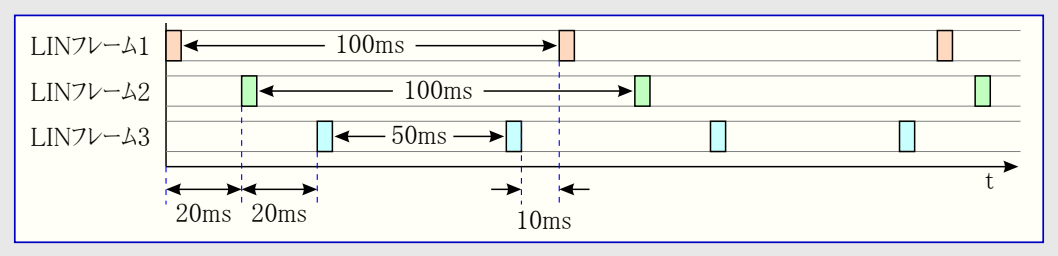
- LINﾌﾚｰﾑ宣言と初期化

```
t_frame MESS_SET_SLAVE;

MESS_SET_SLAVE.frame_id    = SET_SLAVE_FRAME_ID ;
MESS_SET_SLAVE.frame_size  = SET_SLAVE_FRAME_DLC ;
MESS_SET_SLAVE.frame_type  = SET_SLAVE_FRAME_TYPE ;
MESS_SET_SLAVE.frame_delay = 1500 ;                    // 計時での遅延
MESS_SET_SLAVE.frame_data  = Buf_SET_SLAVE;
```

- ｽｹｼﾞｭｰﾗ ﾀｲﾐﾝｸﾞ定義

  全てのﾀｲｲﾝｸﾞを良く知らなければなりません。例えば3つのLINﾌﾚｰﾑが次のように送られなければならない場合は以下です。



  LINﾌﾚｰﾑは次のように初期化されなければなりません。

  - LINﾌﾚｰﾑ1遅延=20ms (delay_tick=20ms×19200bps=384)
  - LINﾌﾚｰﾑ2遅延=20ms
  - LINﾌﾚｰﾑ3遅延=50ms
  - LINﾌﾚｰﾑ3遅延=10ms

- "my_schedule"項目初期化

  "my_schedule"項目は使用される全てのLINﾌﾚｰﾑとLINﾌﾚｰﾑ数が必要です。

```
my_schedule.frame_message[0] = MESS_SET_SLAVE;
my_schedule.number_of_frame = number_of_frame;
```

  ｿﾌﾄｳｪｱで許される最大ﾌﾚｰﾑ数を変更するには"master_lin.h"のNB_FRAME値を変更してください。

- LIN初期化

  I/Oﾎﾟｰﾄ形態設定(CONFIG_IO_PORTS())後に主関数でLIN_init()関数が実行されなければなりません。

- 計時(tick)関数

  計時(tick)関数は通常、主関数、または例えば利用可能な計時器を通してTBIT速度で呼ばれなければなりません。現在、LIN_TIMER_init関数が溢れ毎に再設定するようにﾀｲﾏ/ｶｳﾝﾀ1を初期化します。"timer1ovf_isr.c"は計時(tick)関数呼び出しに使用可能なﾀｲﾏ/ｶｳﾝﾀ1を示します。

- 使用者応用によるﾃﾞｰﾀ処理

  全ての形態設定と初期化は終りです。LIN主装置は使用準備が整っています。

自身のLIN従装置構成法

LIN従装置を動かすために以下の各種段階に従わなければなりません。

- LINﾗｲﾌﾞﾗﾘ任意選択定義。"slave_lin.h"ﾌｧｲﾙを調べてください。
- LINﾗｲﾌﾞﾗﾘ形態設定。"config.h"ﾌｧｲﾙを設定してください。
- ﾃﾞｰﾀ緩衝部宣言と初期化

  LIN応用で必要とする全ての緩衝部を設定してください。

・LINﾌﾚｰﾑ宣言と初期化

```
t_frame MESS_SET_SLAVE;

MESS_SET_SLAVE.frame_id     = SET_SLAVE_FRAME_ID ;
MESS_SET_SLAVE.frame_size   = SET_SLAVE_FRAME_DLC ;
MESS_SET_SLAVE.frame_type   = SET_SLAVE_FRAME_TYPE ;
MESS_SET_SLAVE.frame_data   = Buf_SET_SLAVE;
```

・"my_schedule"項目初期化

"my_schedule"項目は使用される全てのLINﾌﾚｰﾑとLINﾌﾚｰﾑ数が必要です。

```
my_schedule.frame_message[0] = MESS_SET_SLAVE;
my_schedule.number_of_frame = number_of_frame;
```

ｿﾌﾄｳｪｱで許される最大ﾌﾚｰﾑ数を変更するには"slave_lin.h"のNB_FRAME値を変更してください。

・LIN初期化

I/Oﾎﾟｰﾄ形態設定(CONFIG_IO_PORTS())後に主関数でLIN_init()関数が実行されなければなりません。割り込み許可を完了するには、SREG|=0x80です。

・計時(tick)関数

計時(tick)関数は通常、主関数、または例えば利用可能な計時器を通してTBIT速度で呼ばれなければなりません。現在、LIN_TIMER_init関数が溢れ毎に再設定するようにﾀｲﾏ/ｶｳﾝﾀ1を初期化します。"timer1ovf_isr.c"は計時(tick)関数呼び出しに使用可能なﾀｲﾏ/ｶｳﾝﾀ1を示します。

・使用者応用によるﾃﾞｰﾀ処理

全ての形態設定と初期化は終りです。LIN従装置は使用準備が整っています。

LINﾗｲﾌﾞﾗﾘの性能

LIN主装置

・使用資源
- 1つのｼﾘｱﾙ周辺機能(USART)
- 時間超過検出用に予約された1つのﾀｲﾏ/ｶｳﾝﾀ
- 計時(tick)関数を周期的に呼び出すための1つのﾀｲﾏ/ｶｳﾝﾀ (任意選択)

・ｺｰﾄﾞ量と使用SRAM

全ての結果は全ての任意選択(休止時間超過検出、時間超過検出、異常計数)許可で与えられます。IAR®ｺﾝﾊﾟｲﾗでの最適化なしと量最適化最大の2つ形式の結果が与えられます。

- 最適化なし

  総ｺｰﾄﾞ量 ：4Kﾊﾞｲﾄ
  総ﾃﾞｰﾀ量 ：648ﾊﾞｲﾄ(退避とｽﾀｯｸを含む)

ﾗｲﾌﾞﾗﾘ	ｺｰﾄﾞ (ﾊﾞｲﾄ)	ﾃﾞｰﾀ (ﾊﾞｲﾄ)
lin_drv_usart	1246	18
lin_lib	974	0
master_lin	862	46
main	154	8

- 量最適化最大

  総ｺｰﾄﾞ量 ：3.1Kﾊﾞｲﾄ
  総ﾃﾞｰﾀ量 ：648ﾊﾞｲﾄ(退避とｽﾀｯｸを含む)

ﾗｲﾌﾞﾗﾘ	ｺｰﾄﾞ (ﾊﾞｲﾄ)	ﾃﾞｰﾀ (ﾊﾞｲﾄ)
lin_drv_usart	810	18
lin_lib	734	0
master_lin	692	46
main	154	8

・CPU負荷

最悪状態(2つのﾌﾚｰﾑ間で遅延なし)に於いて、CPU負荷は19200bpsで約5%です。2400bpsのLINﾎﾞｰﾚｰﾄに対してはCPU負荷が0.6%に低下します。

LIN従装置

・使用資源
- 1つのｼﾘｱﾙ周辺機能(USART)
- 時間超過検出用に予約された1つのﾀｲﾏ/ｶｳﾝﾀ
- 計時(tick)関数を周期的に呼び出すための1つのﾀｲﾏ/ｶｳﾝﾀ
- 走行時RC発振器校正許可なら、1つの捕獲入力部(AVR054応用記述を参照してください。)

・ｺｰﾄﾞ量と使用SRAM

全ての結果は全ての任意選択(休止時間超過検出、時間超過検出、異常計数)許可で与えられます。IAR®ｺﾝﾊﾟｲﾗでの最適化なしと量最適化最大の2つ形式の結果が与えられます。

－ 最適化なし

総ｺｰﾄﾞ量 ：4Kﾊﾞｲﾄ
総ﾃﾞｰﾀ量 ：648ﾊﾞｲﾄ(退避とｽﾀｯｸを含む)

ﾗｲﾌﾞﾗﾘ	ｺｰﾄﾞ (ﾊﾞｲﾄ)	ﾃﾞｰﾀ (ﾊﾞｲﾄ)
lin_drv_usart	1246	18
lin_lib	974	0
slave_lin	862	46
main	154	8

－ 量最適化最大

総ｺｰﾄﾞ量 ：3.1Kﾊﾞｲﾄ
総ﾃﾞｰﾀ量 ：648ﾊﾞｲﾄ(退避とｽﾀｯｸを含む)

ﾗｲﾌﾞﾗﾘ	ｺｰﾄﾞ (ﾊﾞｲﾄ)	ﾃﾞｰﾀ (ﾊﾞｲﾄ)
lin_drv_usart	810	18
lin_lib	734	0
slave_lin	692	46
main	154	8

・CPU負荷

最悪状態(2つのﾌﾚｰﾑ間で遅延なし)に於いて、CPU負荷は19200bpsで約5%です。2400bpsのLINﾎﾞｰﾚｰﾄに対してはCPU負荷が0.6%に低下します。

他のAVR目的対象用へのｺｰﾄﾞ変更法

他のAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗを使用するにはいくつかのｺｰﾄﾞ変更が必要です。それらは以下です。

・config.h
・lin_drv_usart.h

"config.h"ﾌｧｲﾙで使用するAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗを定義してください。

```
#define NAME_UC_USED
```

次に連携するﾏｲｸﾛ ｺﾝﾄﾛｰﾗと基板のﾍｯﾀﾞをｲﾝｸﾙｰﾄﾞし、そして使用するUARTを定義してください。

```
#ifdef NAME_UC_USED
#include <iomNAME_UC.h>
#include "..¥board.h"
#define USE_UART0/1
#endif
```

"lin_drv_usart.c"で(1つよりも多くのUSARTを持つﾏｲｸﾛ ｺﾝﾄﾛｰﾗに関して)使用するUSARTに応じて使用される割り込みﾍﾞｸﾀを定義してください。

```
#ifdef NAME_UC_USED
#pragma vector = USART0/1_RX_vect
#endif
```

"lin_hw_init()"関数でUSARTの送信と受信を許可してください。

```
#ifdef NAME_UC_USED
#define UCSR0/1B = (1<<RXEN0/1 ) | (1<<TXEN0/1) | (1<<RXCIE0/1);
```

結び

この応用記述はATMELのAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗにLIN規約1.3版を実装するためのｿﾌﾄｳｪｱ解決策を提供します。全てのｿｰｽ ｺｰﾄﾞがATMELのｳｪﾌﾞｻｲﾄで入手できます。

必要な資源は殆どのAVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗに適合します。使用ﾒﾓﾘとCPU負荷は他の機能を実装するための開発に対して充分低く保たれています。

本社

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan
104-0033 東京都中央区
新川1-24-8
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求
www.atmel.com/literature

本応用記述はATMELのAVR322応用記述(doc7548.pdf Rev.7548A-12/05)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。