

# AVR308 : ソフトウェアLIN従装置

## 要点

- 低費用、外部部品なし
- LIN規約仕様1.0版適合
- 効率的な量のコード

## 序説

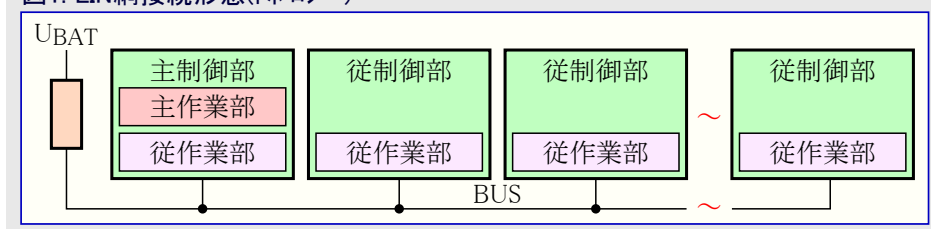
この応用記述は何の外部部品の必要もなしに8ビットRISC AVRマイクロコントローラで局所相互連結網(LIN:Local Interconnect Network)従作業部を実装する方法を示します。

LIN規約は分配された網内でメカトロニクス節点(ノード)の制御を効率的に支援する直列通信規約です。これは車載応用での使用に関して理想的な規約にします。LIN網は単一主装置と一群の従装置節点から成ります。この応用記述はISO/OSI参照モードの2つの最低水準に従って物理層とデータリンク層に関する規約を実装する方法を示します。これは網内の節点間のメッセージ転送に関する基礎を提供します。ISO/OSIモードの物理層は網内の2つの節点間のビット列の流れを提供するだけです。データリンク層は誤り検出と制御を追加することにより、確かな物理層にします。それは接続(リンク)を活性、維持、不活性にすることの意味も追加します。ISO/OSI参照モードのより高いレベル(層)はLIN規約を越え、従ってここでは説明されません。当座、これを定めて実装するのは使用者の責任です。

LIN規約はそれがCANの範囲と性能以下であることの意味に於いてCAN規約と異なります。利点はCAN規約の能力を必要としない応用に対して簡単で安価な解決策を提供することです。LINバスの主な特徴は以下を含みます。

- ・ 単一主装置、複数従装置
- ・ 低費用シリコン実装
- ・ 従節点に於けるクリスタル発振子またはセラミック振動子なしでの自己同期
- ・ 信号送出に対する遅延時間の保証
- ・ 単線実装
- ・ 20kbpsまでの速度

図1. LIN網接続形態(トポロジー)



## LIN規約の概念

### 単一主装置複数従装置

LIN規約はバス調停をしません。1つの主装置が全てのメッセージ転送の開始に関して責任を持ちます。全ての従装置が網内の主装置または他のどの節点にも応答できますが、主装置によってアドレス指定されて許可を与えられた後でだけです。

### 可変長データフレーム

識別子領域に含まれた2つのビットがメッセージ領域の長さを示します(表1をご覧ください)。これは制限されたデータ量だけが必要とされる時に柔軟性と無駄なバイトの低減を付加します。

### 多数割り当て受信

主または従の作業部からメッセージフレームが送信される時に、網内に接続された全ての節点はメッセージを読むことができます。識別子バイトに依存して、受信する節点はどの動作が開始されるべきか否かを決めます。例えば、主装置からの単独"CLOSE ALL"命令は全ての節点で受け入れられ、車の保安システムの場合に全ての窓と扉を閉めることができます。

(訳注) 同梱のソースは若干修正したために元と異なります。また本書のコード量記述も異なりますので注意してください。



8ビット **AVR**<sup>®</sup>  
マイクロコントローラ

## 応用記述

本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 1637B-05/02, 1637BJ3-04/21

## 従装置節点でクリスタル発振子またはセラミック振動子の必要なしでの時間同期

同期中断後、同期領域が主装置から送信され、この領域は主装置クロックへの同期を全ての従装置に対して可能にします。このような同期領域が毎メッセージ フレームの始めに配置されます。受信する従装置の精度はメッセージ フレーム全体を通して同期を維持するのに充分良好なことだけを必要とします。この特徴は内蔵RC発振器での走行、従ってシステム費用全体の低減を従装置に許します。

## データ チェックサムの安全性と誤り検出

メッセージ フレーム内のデータは誤り検出のためにMSBの桁溢れをLSBに加算し、256の剰余を反転したチェック サムを使います。加えて、識別子バイトは誤り検出に関してXOR法を使います。

## 網内の欠損節点(ノード)の検知

主作業部にはメッセージ フレームの送信を開始することに関して責任があり、従って情報を要求して全ての節点が生きていて正しく動いていることを調べる責任を持ちます。

## 最低費用解決策

規約の単純性のため、従節点内の発振器の精度を重い束縛に置かず、最小の外部部品を使って、LIN規格に適合する従作業部を構築することができます。

## 信号送信

LINバスはデータと同期情報の両方を運ぶ単一チャネルから成ります。物理的な媒体はプルアップ抵抗経由でVCCに接続された単一線です(図1をご覧ください)。このバスのアイドル状態はHighまたは“劣性”、そして活性状態はLowまたは“優性”です。車載応用でのVCCは代表的に蓄電池の正節点でしょう。

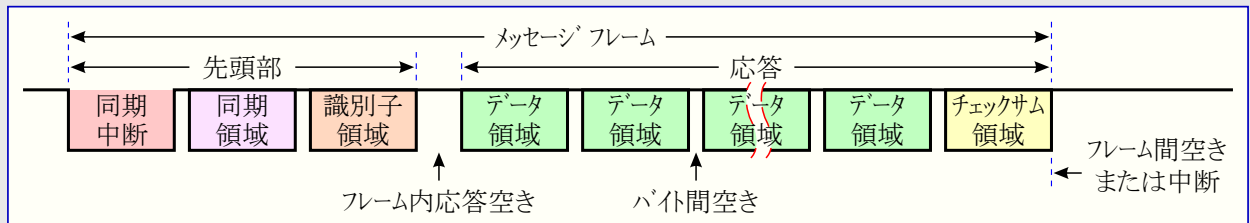
LIN規約は従作業部に関して応答手順を定義していません。主作業部は送ったメッセージ フレームが従作業部によって受信したものと同一ことを検証するのに自身の従作業部を使います。何らかの不一致が検出された場合、メッセージ フレームを再送信することができます。送信データに関するデータ速度は単線送信媒体に対するEMI(電磁妨害)必要条件のため、20kbpsに制限されます。

## メッセージ フレーム

LINバス上に送信される全ての情報はメッセージ フレームとして形式化されます。図2.で示されるように、メッセージ フレームは以下の領域から成ります。

- 同期中断
- 同期領域
- 識別子領域
- データ領域
- チェックサム領域

図2. LINメッセージ フレーム

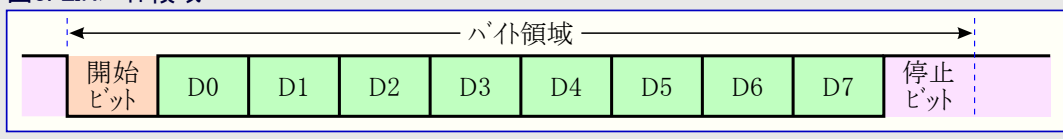


メッセージ フレームは主装置によって送られる“先頭部”と主と従の両作業部によって使うことができる“応答”の2つの部分から成ります。

## バイト領域

図3.で示されるバイト領域形式は一般に使われる8N1符号化でのUART直列データ形式と同じです。これは各バイト領域が8つのデータビット、パリティビットなし、1つの停止ビットを含むことを意味します。各バイト領域は10ビット時間(Tビット)の長さを持ちます。図で示されるように、バイト領域の始めを記す開始ビットは“優性”で、一方停止ビットは“劣性”です。8つのデータビットは“優性”または“劣性”のどちらにもできます。

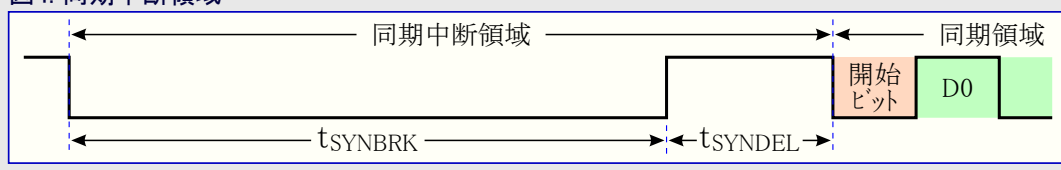
図3. LINバイト領域



## 同期中断

同期中断はメッセージフレームの始まりを記します。この領域は常に主作業部によって送られ、同期領域に対する準備の意味に従作業部に提供します。同期中断は最低13ビット時間(Tビット)であるべき優性(Low)レベルと、後続する1~4Tビット範囲であるべき劣性(High)区間の2つの異なる部分から成ります。2つ目の領域は後続する同期領域の開始ビットの優性(Low)レベルを検出するのに必要です。

図4. 同期中断領域

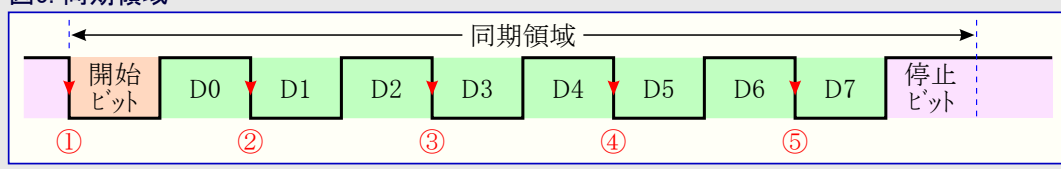


最初の領域の長さは同期中断とデータフレーム内で許された可能な最大連続優性ビット間を区別するように選ばれています。例えば、全てが'0'のデータ領域が同期中断領域に誤ら(誤認)されるべきではありません。

## 同期領域

同期領域は主装置クロックに同期するための従装置に対して必要とされる合図を含みます。同期領域は図5.で示される波形を生成する"\$55"のデータを含むパケット領域です。

図5. 同期領域



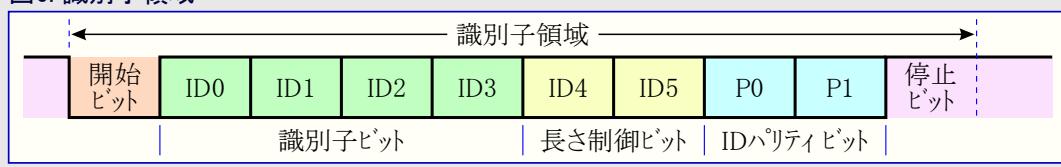
図解されるように、同期領域は5つの下降端(劣性から優性への端)によって特性付けされます。

この[応用記述の後ろで説明](#)されるように、従節点の送受信速度を主節点に合わせるように調整するため、これらの端は同期化中に使われます。

## 識別子領域

識別子領域はメッセージの内容と長さについての情報を含みます。図6.で示されるように、この領域は識別子ビット(4ビット)、長さ制御ビット(2ビット)、パリティビット(2ビット)の3つの分野に分けられます。これは64識別子群を各々16識別子の4つの副群に分けます。

図6. 識別子領域



LIN規約はこれを以下のように定義します。

表1.で示されるように2つのデータ領域を持つ2つの群、4つのデータ領域を持つ1つの群、8つのデータ領域を持つ1つの群があります。識別子領域がメッセージの行き先を示すのではなく、メッセージフレームの内容を記述することに注意してください。受信したメッセージで動くべきか否かを決めるのは受信する全ての従作業部の責務です。識別子領域の最後の2ビットはパリティ情報を含みます。LINは識別子領域が決して全て"**劣性**"または"**優性**"の形から成らないことを保証する混合パリティ法を使います。パ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てください。

パリティ検査ビットは以下の混合パリティ法によって計算されます。

$$P0 = ID0 \text{ XOR } ID1 \text{ XOR } ID2 \text{ XOR } ID4$$

$$\overline{P1} = ID1 \text{ XOR } ID3 \text{ XOR } ID4 \text{ XOR } ID5$$

表1. データフレームに於けるデータ領域数

ID5	ID4	NDA(データ領域数)
0	0	2
0	1	2
1	0	4
1	1	8

## データ領域

データフレームは各々がデータの8ビットを含む2つから8つのデータ領域から成ります。送信はLSB先行で行われます。データ領域は応答する従作業部によって書か(送信)れます。バス調停がないので、各識別子に対して1つの従作業部だけが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。

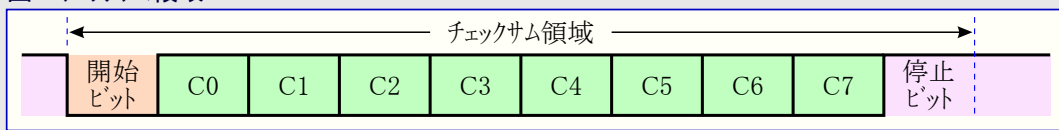
図7. データ領域



## 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(データフレームは識別子を含みません)。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。

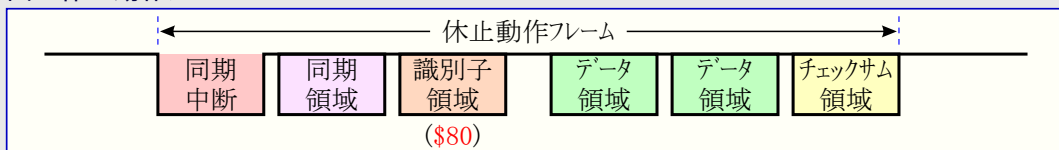
図8. チェックサム領域



## 休止動作フレーム

休止動作フレームに関するフレーム構造は通常のメッセージフレームと同じで、"\$80"の識別子バイトによって区別されます。データ領域の内容は指定されず、休止動作に関するシステムパラメータを配給するのに使うことができます。

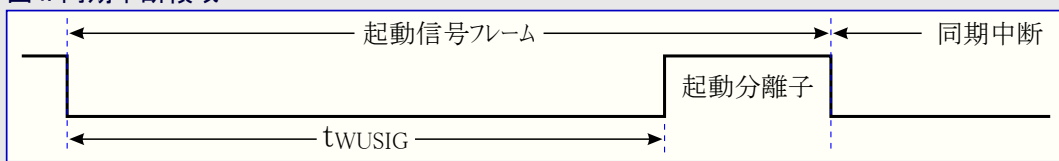
図9. 休止動作フレーム



## 起動信号

休止動作は起動信号を送ることで接続されたどの従作業部によっても終了することができます。起動信号はバスが休止動作中で、節点(ノード)の起動に対する内部要求が未決定の時にだけ許されます。起動信号は"\$80"のキャラクタです。

図4. 同期中断領域



この起動を発行する従作業部が主作業部と同期しているか否かのどちらに依存して、送信されたキャラクタの送信速度は受信する主装置よりも速いまたは遅いかもしれません。この理由のため、"\$80"は"\$C0", "\$80", "\$00"のどれかとして受信され得ます。その全てが所属する従節点によって有効な起動条件と見なされ、そしてそれらは起動して主装置からの同期中断フレームを待ちます。128ビットの時間枠内にこのような領域が検出されない場合、新規の起動信号が発行されます。これは3回繰り返されます。未だ主装置からの応答がない場合、起動要求を保留している従装置は再び試行する前に最低15,000ビット待ちます。起動分離子はLIN規約の現在版に於いて4~64ビットの範囲内であるべきと指定されています。

## 誤り処理

LIN規約は識別子領域とデータ領域の両方で誤り検出を含みます。誤り訂正は全く指定されておらず、従装置に関して不正な送信で自動的に情報を送信する如何なる方法もありません。主装置にはこの情報に関して従装置をポーリングする責任があります。

LIN規約は正しく受信したメッセージに対する応答に関する手続きを定義していません。主制御部は主作業部で送ったメッセージを自身の従作業部で受信したものと比較します。それらが合っていれば正しい送信と仮定されます。不一致の場合、主作業部はメッセージを再送することができます。

従装置によって不一致が検出された場合、この情報は保存され、主装置からの要求で提供されるでしょう。この診断情報は通常のデータフレームの一部として送信することができます。

## 接続

理屈上、節点(ノード)総数は無制限です。現実的にその数は雑音、遅延とバス上の電氣的な負荷によって制限されます。副網内の節点総数は16を越えるべきではありません。原理上、この規約は63節点を支援します。LIN規約は通電線総延長が40mを越えるべきでないことも指定しています。バス終端は主節点に対して1kΩ、従節点に対して20～40kΩとして指定されています。

## メッセージ転送

メッセージ濾過は識別子全体に基きます。全ての従装置はメッセージを読んでそれで動くことができますが、送信された識別子に応答することは1つの節点だけが許されます。

メッセージはメッセージフレーム全体を通して誤りが検出されない場合にだけ有効です。メッセージが不正にされた場合、それは従作業部と主作業部の両方に於いて送信されなかったと見做されます。

## 障害制限

従作業部は以下の異常状態を認識することができるべきです。

- ・自身の送信を読み返している時のデータ領域またはチェックサム領域での“ビット異常”
- ・バスから読んでいる時の“識別子パリティ誤り”または“チェックサム誤り”
- ・バスから読んでいる時の“従装置無応答異常”検出
- ・与えられた許容誤差内で同期領域のエッジが検出されない時の“矛盾同期ビット異常”検出

## 発振器許容誤差

チップ上のRC発振器は製造中の工程要因のため、代表的に大きな許容誤差範囲(例えば、-50%～+100%)を持ちます。LIN規約は、従装置の周波数状態はそれが同期を失った時に主装置と比べて±15%内であるべき、と指定しています。短い時間のフレームでの許容誤差は±2%よりも良好であるべきです。これらの必要条件はデバイスの電圧と温度の全範囲に渡って有効であるべきです。

## ソフトウェア実装

プログラムの各種領域の実装が以降の項で説明されます。プログラムは7つの異なるルーチンを含みます。

表2. 実装された関数

関数	説明	呼び出し元
main	主プログラム	リセット
ext_int0	外部割り込み0用割り込み処理	INT0(PD2)の下降端
check_parity	渡された値内の優性ビット数を計数	ext_int0
tim0_ovf	ビット時間計数時にタイマ/カウンタ0溢れ時に呼び出されます。	タイマ/カウンタ0ハードウェア
putchar	LINバス経由でデータを送信	ext_int0
wakeup	主装置へ起動信号を送出	使用者
(delay1～6)	一般的な遅延(訳注:独立ルーチンではなく各ルーチン内に組み込み)	ext_int0他



## main

本例の主プログラムは何もせずに無限繰り返しを行います。これは使用者応用コードが置かれるべき場所です。AT90S1200がどんなSRAMも持たず、伝統的なSRAMスタックの代わりに3段のハードウェアスタックを用いることに注意してください。3段よりもっと使われる場合、スタックの最初の値が上書きされて失われます。これは予期しないプログラムの動きを引き起こします。(バスの動きのために)外部割り込みが起こると、割り込みルーチンが呼び出されます。データ受信時、割り込みルーチンは`check_parity`ルーチンを呼び、データ送信時に`put_char`ルーチンを呼びます。故に割り込みルーチンによって2つのスタック段が使われます。従って使用者プログラムは1つよりも多くのスタック段を使ってはならず、スタックは合計3つのスタック段を提供します。

図11. 状態ビット

ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
ビット誤り	パリティ誤り						休止

ビット誤りやパリティ検査誤りが検出されると、状態レジスタ(R20)内の対応するビットが設定(1)されます。ビット位置については図11をご覧ください。これらのフラグは読んだ後、手動で削除(0)されなければなりません。主プログラムは継続的に停止フラグをポーリングすべきです。このフラグが1になったとき、以下の命令によってAT90S1200をアイドル動作に置くことができます。

```
LDI    R23, (1<<SE) | (1<<ISC01)    ; 休止命令許可, アイドル休止形態指定値取得
OUT     MCUCR, R23                    ; 休止命令許可, アイドル休止形態指定
SLEEP                                     ; アイドル休止形態へ移行
```

何らかのバスの動きが検出されると、デバイスは起きて`ext_int0`ルーチンを実行します。LINインターフェースによって181語が費やされ、故に使用者プログラム量は最大331語(662バイト)にすることができます。これは合計512語(1Kバイト)を提供します。

## 外部割り込み0 : ext\_int0

このルーチンはLIN従装置インターフェースの全機能を含みます。外部割り込み0はINT0(PD2)ピンでのHigh⇒Low端によって起動されます。割り込みルーチンは同期中断の開始端、同期バイト内の毎下降端、識別子の開始バイトによって起動されます。バス上で何を予期するかを経緯を保つため、"mode"変数によって定義されるように3つの形態が用いられます。

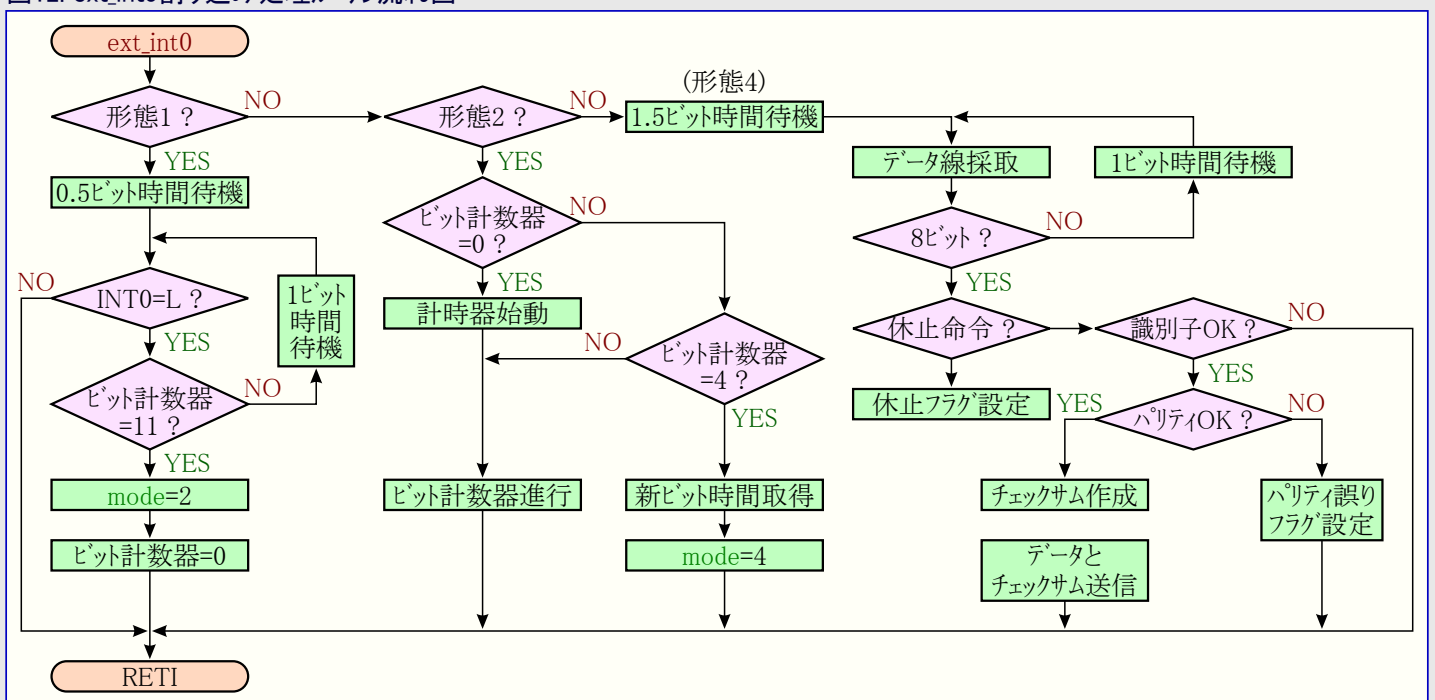
- ・形態1：プログラムは同期中断の受信を予期します。
- ・形態2：プログラムは同期バイトの受信を予期します。
- ・形態4：プログラムは識別子の受信を予期します。

図12は`ext_int0`ルーチンの構成流れ図を示します。

形態2で最初の下降端が検出されると、計時器が開始されます。そしてプログラムは下降端数を数え、5つの下降端が検出された時に計時器が停止されます(図5をご覧ください)。計時器値を8で除算することによってビット幅が得られます。

形態4は送信された識別子を読みます。休止命令があるかを見るため、最初に識別子が調べられます。休止命令が受信されなかったなら、それが応答されるべきかを見るために識別子が内部遮蔽値と比較されます。識別子が遮蔽値と対応する場合、パリティが調べられます。そしてチェックサムが作成され、データとチェックサムが送られます。

図12. ext\_int0割り込み処理ルーチン流れ図



## check\_parity

このルーチンはcounter変数内の劣性ビット数を数えます。偶数なら、bitcount変数のLSBは0です。数が奇数なら、bitcount変数のLSBは1です。これは奇偶パリティに関してcounter変数を調べるのに用いられます。

## タイマ/カウンタ0溢れ: tim0\_ovf

この応用で使われるタイマ/カウンタ0は8ビット計数器です。8ビット時間を計数できるように、このルーチンはタイマ/カウンタ0溢れ時に呼び出され、“counter”変数を増加します。この“counter”変数とタイマ/カウンタ0値の結合により、その結果は8ビットに対する時間を示す16ビット数値を生成します。これを8で除算することがビット時間を与えます。AVRマイクロコントローラは選択した1MHzの周波数で走行し、そして計数器は毎クロック周期で増加します。従ってこのビット時間は直接μs与えられます。

## putchar

putcharルーチンはLINバス経由でデータを送るのに使われます。“0”送出時、PD2ピンはポート出力レジスタでの“0”と共に出力として設定されます。“1”送出時、PD2ピンは入力として設定され、LINバスのプルアップ抵抗がレベルを論理“1”に引っ張ります。これはオープンコレクタ出力のように働きます。送出されるべきデータは昇順で隣接レジスタに配置されるべきです(例えば、データ0～データnはレジスタRm～Rm+nに配置されます)。従装置は識別子領域内のビット4,5によって決められる2,4,8のどれかの数のデータバイトを送ります。アセンブリ言語ファイル内のdata\_addr定数は最初に送出すべきデータのレジスタ番号を保持します。

## delay

delayルーチンは[temp2]±1周期の遅延を生成します。このルーチンはビットの採取間で必要な遅延を生成するのに使われます。(訳注)これは例に於いて独立したルーチンとして存在せず、各々必要な処に展開されています。

## wakeup

バス上で起動呼び出しを生成するために主プログラムからこのルーチンが呼び出されるかもしれません。このルーチンはext\_int0ルーチンによって使われません。

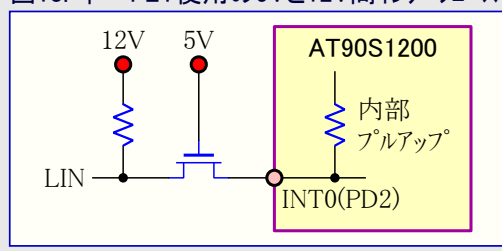
## ハードウェアの考慮

LIN規格は9～18V間での“優性”電圧に対する動作電圧範囲を指定しています(然るに、40Vは動作しません)。この応用は5VのLIN電圧用に構成されています。この応用で使われるAVRマイクロコントローラが安定な5V供給を必要とするので、最も効率的な費用で容易な解決策はLINバスの劣性電圧を5Vにすることです。これが受け入れられない場合、5VのLINを9～18VのLINにインターフェースするいくつかのハードウェア解決策があります。

### 方法1

単一FETを使うことによって5V部と9～18V部を接続できます(図13をご覧ください)。ここで12V部と5V部は単一FETを使って接続されます。FETの5V側か12V側のどちらかがLowに引っ張られると、対応する側が優性ビットを検知するためのLowへ充分に引っ張られます。

図13. 単一FET使用の5Vと12V間インターフェース



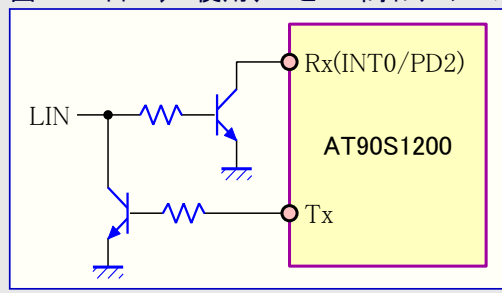
### 方法2

(同じ外圍器内の)2つのバイポーラトランジスタの使用はもっと費用効率的な解決策かもしれません(図14をご覧ください)。これらのトランジスタはベース抵抗とで完全に達成可能でFETトランジスタよりも(費用的に)高くなりません。

この方法が選ばれた場合、プログラムが僅かに書き換えられなければなりません。以下が変えられるべきです。

- putcharルーチンはPD2(Rx)ピンに代わってTxピンへデータを出しなればなりません。
- データは“オープンコレクタ”出力使用時に行われるようにではなく、通常のようにTxピンへ書かなければなりません。
- Txピンは出力(DDRxのDDxn=1)として初期化されなければなりません。
- 送受信されるデータは(Highレベル出力がバス上でLowレベルを生じ、バス上のHighレベルがLowレベル入力を生じる)論理反転されなければなりません。

図14. バイポーラTr使用、5Vと12V間インターフェース



参考: LIN規約仕様1.0版



## 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  
東熱新川ビル 9F  
アトメル ジャパン株式会社  
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 Chanterrie  
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](http://www.atmel.com/literature)

## © Atmel Corporation 2002.

Atmel製品は、ウェブサイト上にあるAtmelの定義、条件による標準保証で明示された内容以外の保証はありません。本製品は改良のため予告なく変更される場合があります。いかなる場合も、特許や知的技術のライセンスを与えるものではありません。Atmel製品は、生命維持装置の重要部品などのような使用を認めておりません。

本書中の®、™はAtmelの登録商標、商標です。

本書中の製品名などは、一般的に商標です。

## © HERO 2021.

本応用記述はAtmelのAVR308応用記述(doc1637.pdf Rev.1637B-05/02)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識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。