



# CAN マルチネットワークボード 取扱説明書 通信編

---

ルネサス エレクトロニクス社 RX231, RL78/F15, RA2L1 搭載  
HSB シリーズ応用キット

-本書を必ずよく読み、ご理解された上でご利用ください

株式会社 **北斗電子**  
REV.1.0.0.0

—目次—

注意事項 .....	1
安全上のご注意 .....	2
概要 .....	4
<b>1. CAN 通信の概要 .....</b>	<b>6</b>
1.1. CAN の接続形態 .....	6
1.2. CAN の物理層 .....	6
1.3. CAN トランシーバとマイコンの接続 .....	7
1.4. CAN のデータパケット .....	8
1.5. ビットスタッフィング .....	10
1.6. 1 ビットと TQ .....	11
1.7. CAN の ID .....	12
1.8. 通信時の実際の波形例 .....	13
<b>2. LIN 通信の概要 .....</b>	<b>14</b>
2.1. 接続形態 .....	14
2.2. LIN の物理層 .....	14
2.3. LIN のデータパケット .....	15
<b>3. UART 通信の概要 .....</b>	<b>17</b>
3.1. 接続形態 .....	17
3.2. UART のデータ波形 .....	17
3.3. UART の通信速度 .....	19
<b>4. マイコンボード上で実行されるデモプログラム(CAN 通信) .....</b>	<b>20</b>
4.1. CAN-ID(1)で出力するデータ .....	21
4.2. CAN-ID(2)でやり取りされるデータ .....	22
4.2.1. データフレーム(外部からの指示) .....	22
4.2.2. リモートフレーム(外部からデータを出力する様に要求) .....	25
<b>5. マイコンボード上で実行されるデモプログラム(LIN 通信) .....</b>	<b>27</b>
<b>6. マイコンボード上で実行されるデモプログラム(UART 通信) .....</b>	<b>30</b>
<b>7. PC 向けデモプログラム .....</b>	<b>36</b>
7.1. HSB_CAN_MULTI_DEMO.EXE .....	36
7.2. HSB_CAN_MULTI_DEMO2.EXE .....	41
取扱説明書改定記録 .....	54
お問合せ窓口 .....	54

## 注意事項

本書を必ずよく読み、ご理解された上でご利用ください

### 【ご利用にあたって】

1. 本製品をご利用になる前には必ず取扱説明書をよく読んで下さい。また、本書は必ず保管し、使用上不明な点がある場合は再読し、よく理解して使用して下さい。
2. 本書は株式会社北斗電子製マイコンボードの使用方法について説明するものであり、ユーザシステムは対象ではありません。
3. 本書及び製品は著作権及び工業所有権によって保護されており、全ての権利は弊社に帰属します。本書の無断複写・複製・転載はできません。
4. 弊社のマイコンボードの仕様は全て使用しているマイコンの仕様に準じております。マイコンの仕様に関しましては製造元にお問い合わせ下さい。弊社製品のデザイン・機能・仕様は性能や安全性の向上を目的に、予告無しに変更することがあります。また価格を変更する場合や本書の図は実物と異なる場合もありますので、御了承下さい。
5. 本製品のご使用にあたっては、十分に評価の上ご使用下さい。
6. 未実装の部品に関してはサポート対象外です。お客様の責任においてご使用下さい。

### 【限定保証】

1. 弊社は本製品が頒布されているご利用条件に従って製造されたもので、本書に記載された動作を保証致します。
2. 本製品の保証期間は購入戴いた日から1年間です。

### 【保証規定】

**保証期間内でも次のような場合は保証対象外となり有料修理となります**

1. 火災・地震・第三者による行為その他の事故により本製品に不具合が生じた場合
2. お客様の故意・過失・誤用・異常な条件でのご利用で本製品に不具合が生じた場合
3. 本製品及び付属品のご利用方法に起因した損害が発生した場合
4. お客様によって本製品及び付属品へ改造・修理がなされた場合

### 【免責事項】

弊社は特定の目的・用途に関する保証や特許権侵害に対する保証等、本保証条件以外のものは明示・黙示に拘わらず一切の保証は致し兼ねます。また、直接的・間接的損害金もしくは欠陥製品や製品の使用方法に起因する損失金・費用には一切責任を負いません。損害の発生についてあらかじめ知らされていた場合でも保証は致し兼ねます。

ただし、明示的に保証責任または担保責任を負う場合でも、その理由のいかんを問わず、累積的な損害賠償責任は、弊社が受領した対価を上限とします。本製品は「現状」で販売されているものであり、使用に際してはお客様がその結果に一切の責任を負うものとします。弊社は使用または使用不能から生ずる損害に関して一切責任を負いません。

保証は最初の購入者であるお客様ご本人にのみ適用され、お客様が転売された第三者には適用されません。よって転売による第三者またはその為になすお客様からのいかなる請求についても責任を負いません。

本製品を使った二次製品の保証は致し兼ねます。

## 安全上のご注意

製品を安全にお使いいただくための項目を次のように記載しています。絵表示の意味をよく理解した上でお読み下さい。

### 表記の意味



取扱を誤った場合、人が死亡または重傷を負う危険が切迫して生じる可能性がある事が想定される



取扱を誤った場合、人が軽傷を負う可能性又は、物的損害のみを引き起こす可能性がある事が想定される

## 絵記号の意味

	<b>一般指示</b> 使用者に対して指示に基づく行為を強制するものを示します		<b>一般禁止</b> 一般的な禁止事項を示します
	<b>電源プラグを抜く</b> 使用者に対して電源プラグをコンセントから抜くように指示します		<b>一般注意</b> 一般的な注意を示しています

## 警告



以下の警告に反する操作をされた場合、本製品及びユーザシステムの破壊・発煙・発火の危険があります。マイコン内蔵プログラムを破壊する場合があります。

1. 本製品及びユーザシステムに電源が入ったままケーブルの抜き差しを行わないでください。
2. 本製品及びユーザシステムに電源が入ったままで、ユーザシステム上に実装されたマイコンまたはIC等の抜き差しを行わないでください。
3. 本製品及びユーザシステムは規定の電圧範囲でご利用ください。
4. 本製品及びユーザシステムは、コネクタのピン番号及びユーザシステム上のマイコンとの接続を確認の上正しく扱ってください。



発煙・異音・異臭にお気づきの際はすぐに使用を中止してください。

電源がある場合は電源を切って、コンセントから電源プラグを抜いてください。そのままご使用すると火災や感電の原因になります。

# 注意



以下のことをされると故障の原因となる場合があります。

1. 静電気が流れ、部品が破壊される恐れがありますので、ボード製品のコネクタ部分や部品面には直接手を触れないでください。
2. 次の様な場所での使用、保管をしないでください。  
ホコリが多い場所、長時間直射日光が当たる場所、不安定な場所、衝撃や振動が加わる場所、落下の可能性がある場所、水分や湿気の多い場所、磁気を発するものの近く
3. 落としたり、衝撃を与えたり、重いものを乗せないでください。
4. 製品の上に水などの液体や、クリップなどの金属を置かないでください。
5. 製品の傍で飲食や喫煙をしないでください。



ボード製品では、裏面にハンダ付けの跡があり、尖っている場合があります。

取り付け、取り外しの際は製品の両端を持ってください。裏面のハンダ付け跡で、誤って手など怪我をする場合があります。



CD メディア、フロッピーディスク付属の製品では、故障に備えてバックアップ（複製）をお取りください。

製品をご使用中にデータなどが消失した場合、データなどの保証は一切致しかねます。



アクセスランプがある製品では、アクセスランプ点灯中に電源の切断を行わないでください。

製品の故障や、データの消失の原因となります。



本製品は、医療、航空宇宙、原子力、輸送などの人命に関わる機器やシステム及び高度な信頼性を必要とする設備や機器などに用いられる事を目的として、設計及び製造されておりません。

医療、航空宇宙、原子力、輸送などの設備や機器、システムなどに本製品を使用され、本製品の故障により、人身や火災事故、社会的な損害などが生じても、弊社では責任を負いかねます。お客様ご自身にて対策を期されるようご注意ください。

## 概要

本書では、CAN マルチネットワークボードで行われている通信に関して説明します。

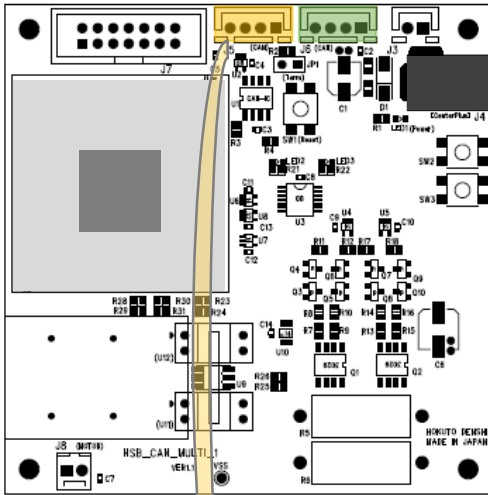
CAN マルチネットワークボードでは、ボード間の通信として、

- ・CAN  
ボード間の通信で使用
  
- ・LIN  
HSB\_CAN\_MULTI\_4 と HSB\_LIN\_COMM ボード間の通信で使用
  
- ・UART(SCI)  
HSB\_CAN\_MULTI\_4 と PC 間の通信で使用

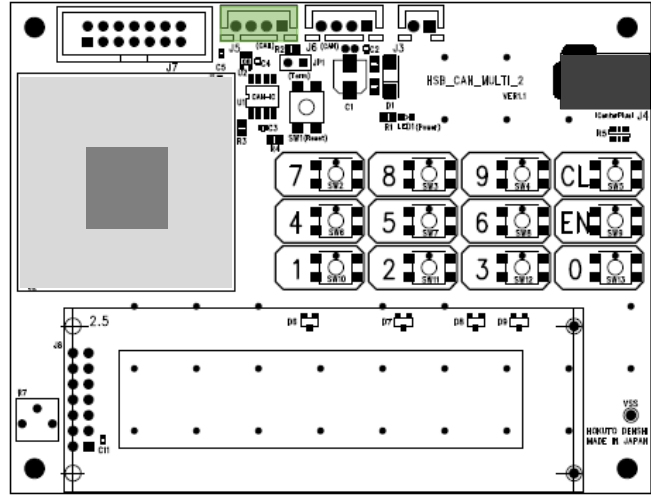
の 3 種類の通信を使用しています。

HSB\_CAN\_MULTI\_1 ~ HSB\_CAN\_MULTI\_4 と、HSB\_LIN\_COMM、PC を接続した際に、それぞれの通信路でどのような信号がやり取りされるかを示しています。

CAN 通信



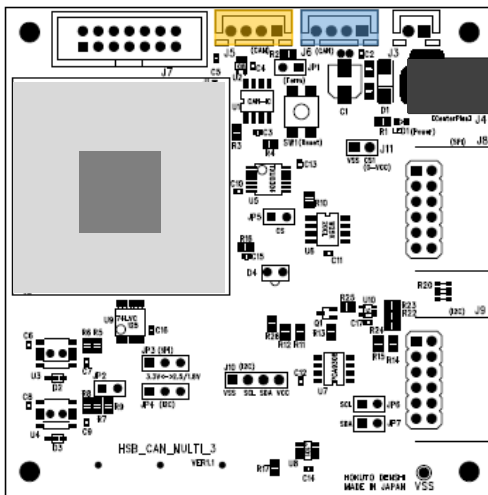
HSB\_CAN\_MULTI\_1



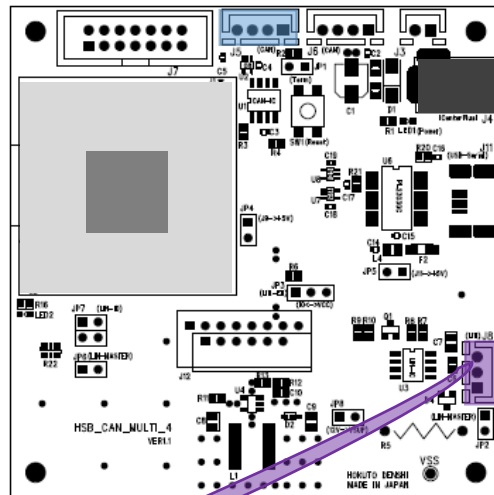
HSB\_CAN\_MULTI\_2

CAN 通信

CAN 通信

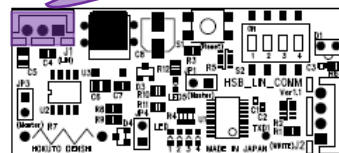


HSB\_CAN\_MULTI\_3



HSB\_CAN\_MULTI\_4

LIN 通信



HSB\_LIN\_COMM

USB-miniB

UART 通信

USB-miniB ケーブル  
[PC に接続]

本キットでは、ボード間の通信として CAN, LIN, UART の 3 種類を使用しています。

# 1. CAN 通信の概要

## 1.1. CAN の接続形態

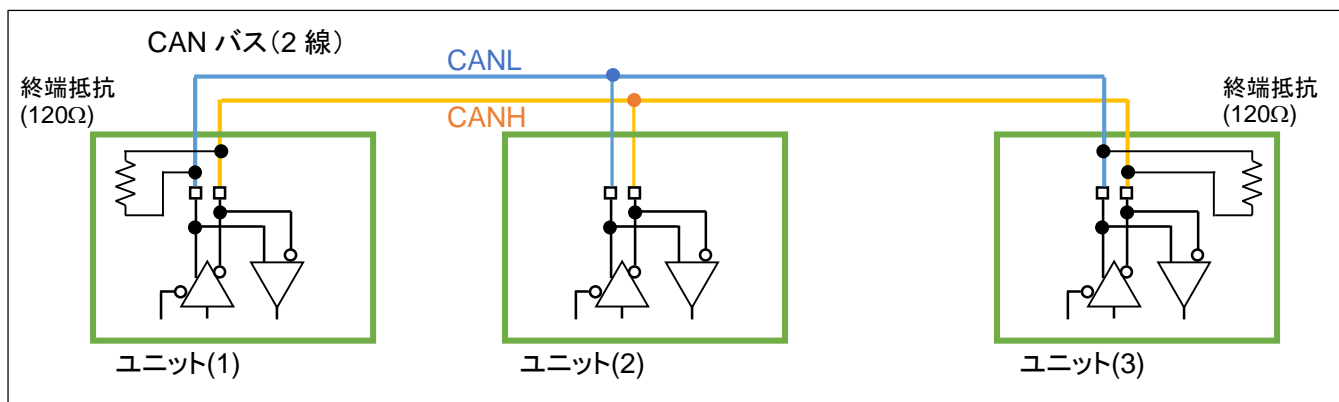


図 1-1 CAN 接続形態

CAN は、2 線の CAN バスのラインに複数のユニットがぶら下がる形となり、以下のような形態を取ります。

- ・CAN バスは 2 線(CANH,CANL)の信号ラインとなる
- ・2 本の信号線で、データの送受信を行う
- ・データパケットに含まれる ID により優先度が変わるが、各ユニットは基本的には対等
- ・バスの終端となるユニットは、終端抵抗を持つ(バスの両端で終端された形となる)

## 1.2. CAN の物理層

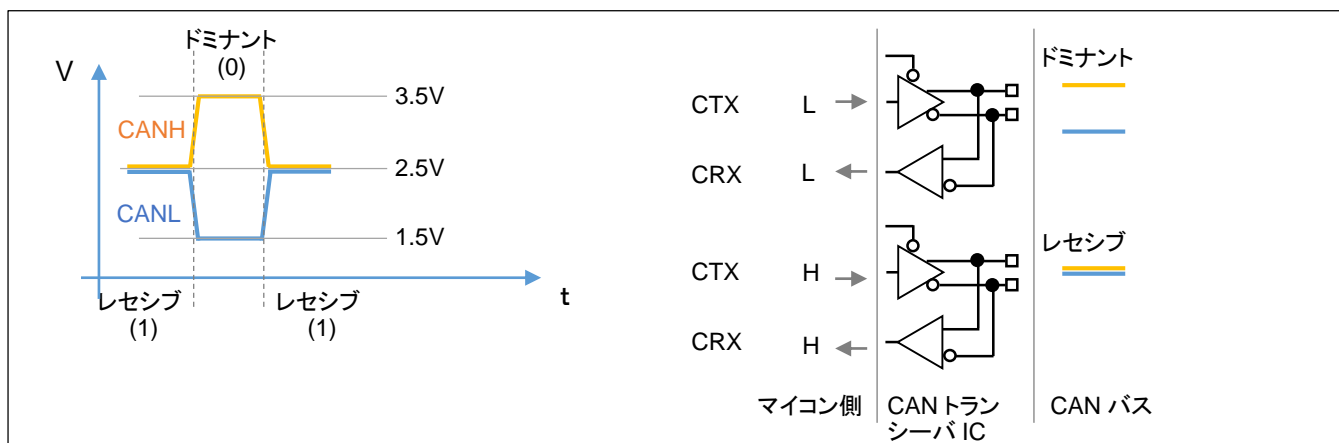


図 1-2 CAN 物理層

CAN の物理層としては、2 本のライン(CANH, CANL)が、同電位に近い状態(レセシブ、デジタル的な 1)と、電位が開いた状態(ドミナント、デジタル的な 0)の、2 状態で、データを送ります。

マイコン側は、L(=VSS 電位,0V)と H(=マイコン VCC 電位,5V または 3.3V)の信号をやりとりしますが、マイコンボード上に搭載されている、CAN トランシーバ IC が、マイコンの L/H と CAN のドミナント / レセシブの変換を行います。

※3.5V/2.5V/1.5V の電圧は代表的な値を示しています



CAN は、LIN や I2C の様にバスに自分のタイミングでデータを流す事のできるマスタが決まっている訳ではないため、バス上でのデータ衝突(2 つ以上のモジュールが同時に出力する)が起こりえる構成となっています。

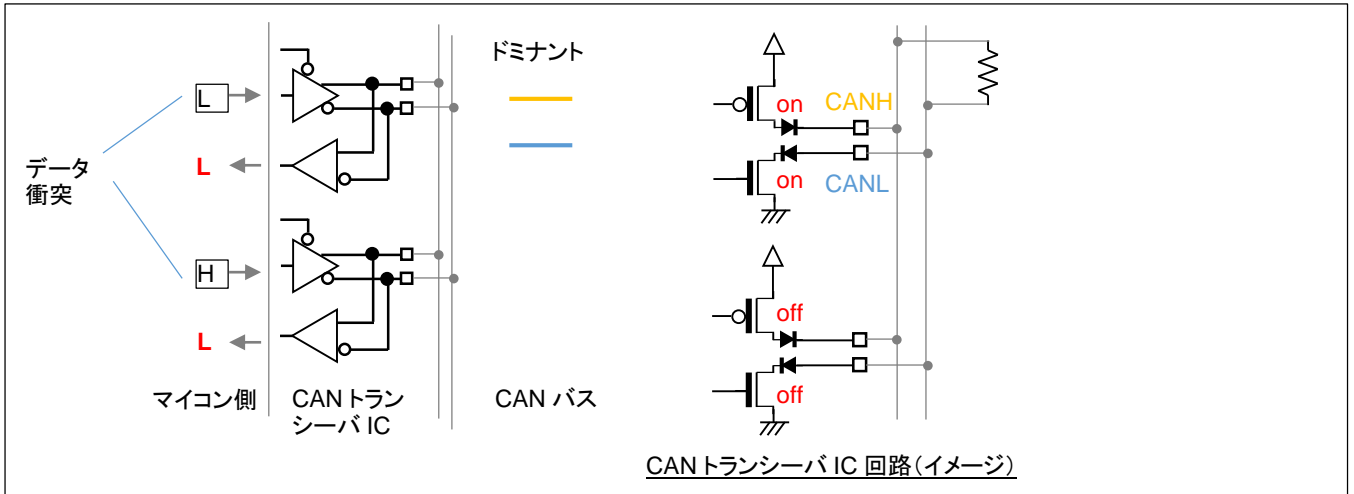


図 1-3 データ衝突時

CAN バスで、データの衝突が起こった際は、バスに接続されているユニットで、1 つでも L(ドミナント)を出力するユニットがあれば、CAN バスはドミナントとなり、CAN バスに接続されているユニットは L を受信します。(規格としてデータの衝突が許されており、ドミナントが優先となります。)

CAN モジュールは、自局が送信したデータと CAN バス上のデータ(トランシーバ回路を経由して受信したデータ)を常に比較しており、データの衝突(自局が送信したデータとバス上のデータの不一致)が起こった際は、CAN の規格に定められた動作(送信をストップし、後ほど再送する等)を行います。

### 1.3. CANトランシーバとマイコンの接続

マイコンと CANトランシーバの接続は図 1-4 に示すような形態となります。

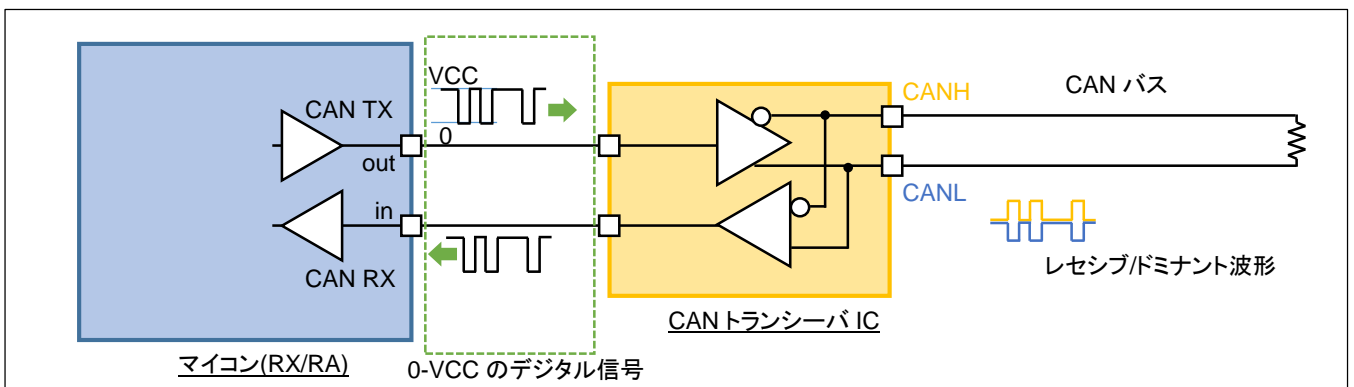


図 1-4 CANトランシーバとマイコンの接続

マイコン側は、デジタルの出力ポートと入力ポートで CAN の信号のやり取りを行います。CANトランシーバ IC は、入力されたデジタル信号を CAN の物理層の波形に変換して CAN バスに流します。また、CAN バスの波形を、デジタル信号に変換してマイコンに渡します。

## 1.4. CAN のデータパケット

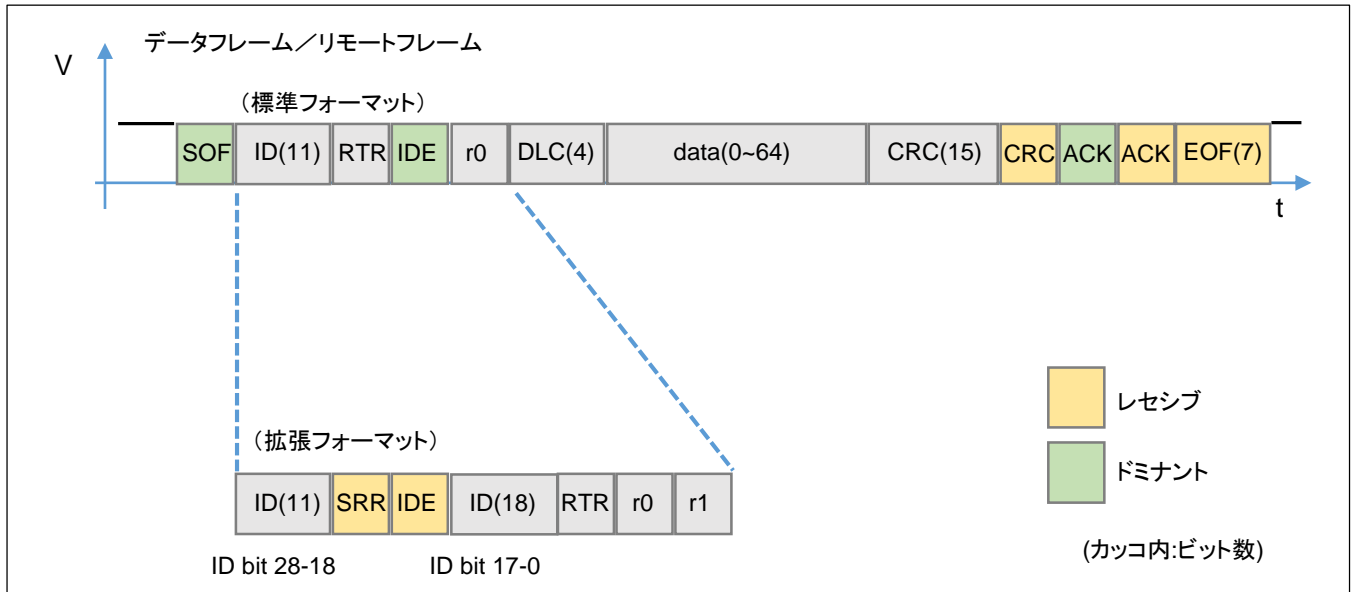


図 1-5 CAN データパケット

CAN のデータパケットは、ID や各種制御情報、CRC コード、ACK(受信側応答)等から構成されています。

CAN には、種々のフレームがありますが、代表的なものはデータを送信する「データフレーム」と、相手にデータの送信を要求する「リモートフレーム」です。

### ・標準フォーマット(先頭部分)

フィールド	ビット長	役割	備考
SOF	1	フレーム送信の開始	ドミナント(0)
ID	11	CAN-ID	
RTR	1	データフレーム(0), リモートフレーム(1)区分	
IDE	1	標準フォーマット(0), 拡張フォーマット(1)区分	標準フォーマットではドミナント(0)
r0	1	予約	

### ・拡張フォーマット(先頭部分)

フィールド	ビット長	役割	備考
SOF	1	フレーム送信の開始	ドミナント(0)
ID	11	CAN-ID(前半)	29 ビットの ID の前半部分
SRR	1	RTR の代替	レセシブ(1)
IDE	1	標準フォーマット(0), 拡張フォーマット(1)区分	拡張フォーマットではレセシブ(1)
ID	18	CAN-ID(後半)	29 ビットの ID の後半部分
RTR	1	データフレーム(0), リモートフレーム(1)区分	拡張フォーマットでは位置が変わる
r0	1	予約	
r1	1	予約	

### ・後半部分(標準フォーマットと拡張フォーマットで共通)

フィールド	ビット長	役割	備考
DLC	4	データバイト数	0~8
data	0~64	送信データ(0~8 バイト)	リモートフレームの場合 0 ビット
CRC	15	データ化けの検証に使用される CRC コード	
CRC	1	CRC のデリミタ(境界)	レセシブ(1)
ACK	1	ACK スロット	送信側レセシブ(1), 受信側ドミナント(0)

フィールド	ビット長	役割	備考
ACK	1	ACK デリミタ(境界)	レセシブ(1)
EOF	7	フレームの終了コード	レセシブ(1)

標準フォーマットの場合、ID は 11bit で構成されます。拡張フォーマットの場合、ID は 29bit となりますが、図 2-5 の様に連続して 29bit 送信される訳ではなく、2 つに分割される形となります。

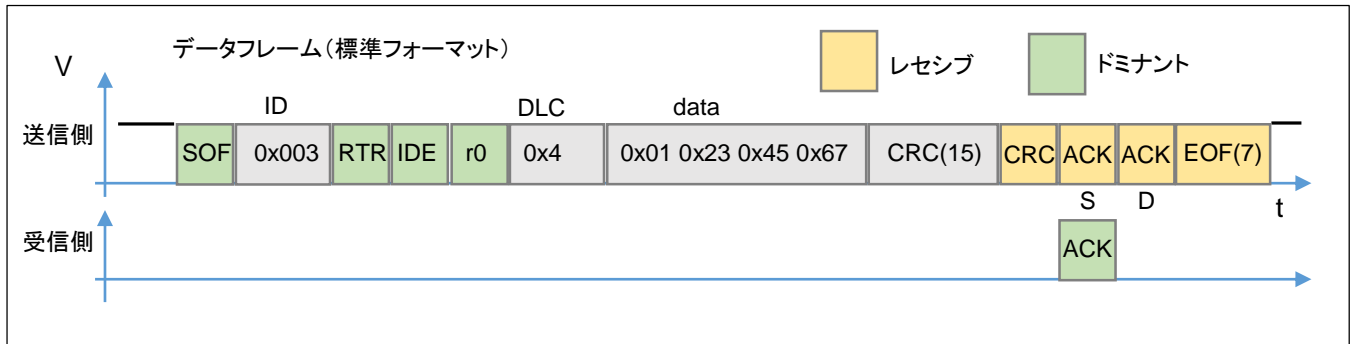


図 1-6 データフレーム送信時のパケット

CAN では、標準フォーマット、データフレームで 4 バイト送信した場合、図 2-6 の様なデータ列となります。送信側は、CRC 15 ビット送信後に、

- CRC デリミタ(レセシブ 1)
- ACK スロット(レセシブ 1)
- ACK デリミタ(レセシブ 1)

を送信します。

このとき、受信側となるモジュールは、SOF から CRC までで、データ列に矛盾がない場合、ACK スロット(ACK, S)のタイミングで、ドミナント(0)を出力(ACK を返す)します。

送信側は、ACK スロットのタイミングで、自分が送信しているのはレセシブ(1)だが、バスの状態がドミナント(0)を検出すると、データが相手に届いたと認識し、データ送信完了(フラグや割り込み)となります。

CAN バス上に、ACK スロットのタイミングでドミナント(0)を返すモジュールがなければ、(送信の設定によっても変わりますが)送信側は、データの再送を試みます(データ送信が繰り返されます)。

よって、CAN の動作を見る場合は、2 つ以上のモジュールを接続し、送信側に対して ACK を返すモジュールが存在する事が非常に望ましいと考えます。(本キットでは、通信の相手となる対向機とペアで使用する事を推奨します。)

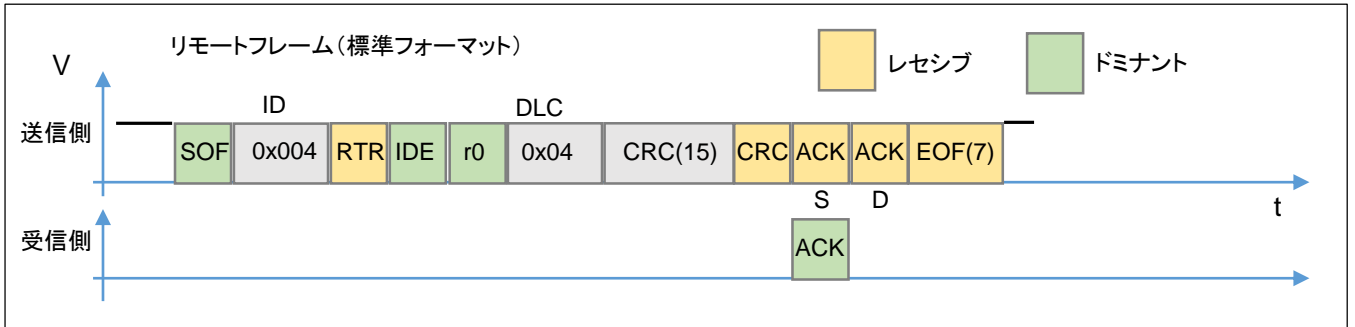


図 1-7 リモートフレーム送信時のパケット

リモートフレーム(相手にデータの送信要求を行う)では、ID=返送を要求する相手の ID, RTR=1, DLC=相手に送って欲しいバイト数, data=空(0 バイト)となります。

## 1.5. ビットスタッフィング

CAN のフレームは、同一信号が 6 ビット以上続く場合、直前の 5 ビットの反転信号(スタッフビット)を挿入する仕様です。図では、レセシブ(1)のスタッフビットとなっていますが、レセシブ(1)のデータが続いた際は、ドミナント(0)のスタッフビットが挿入されます。

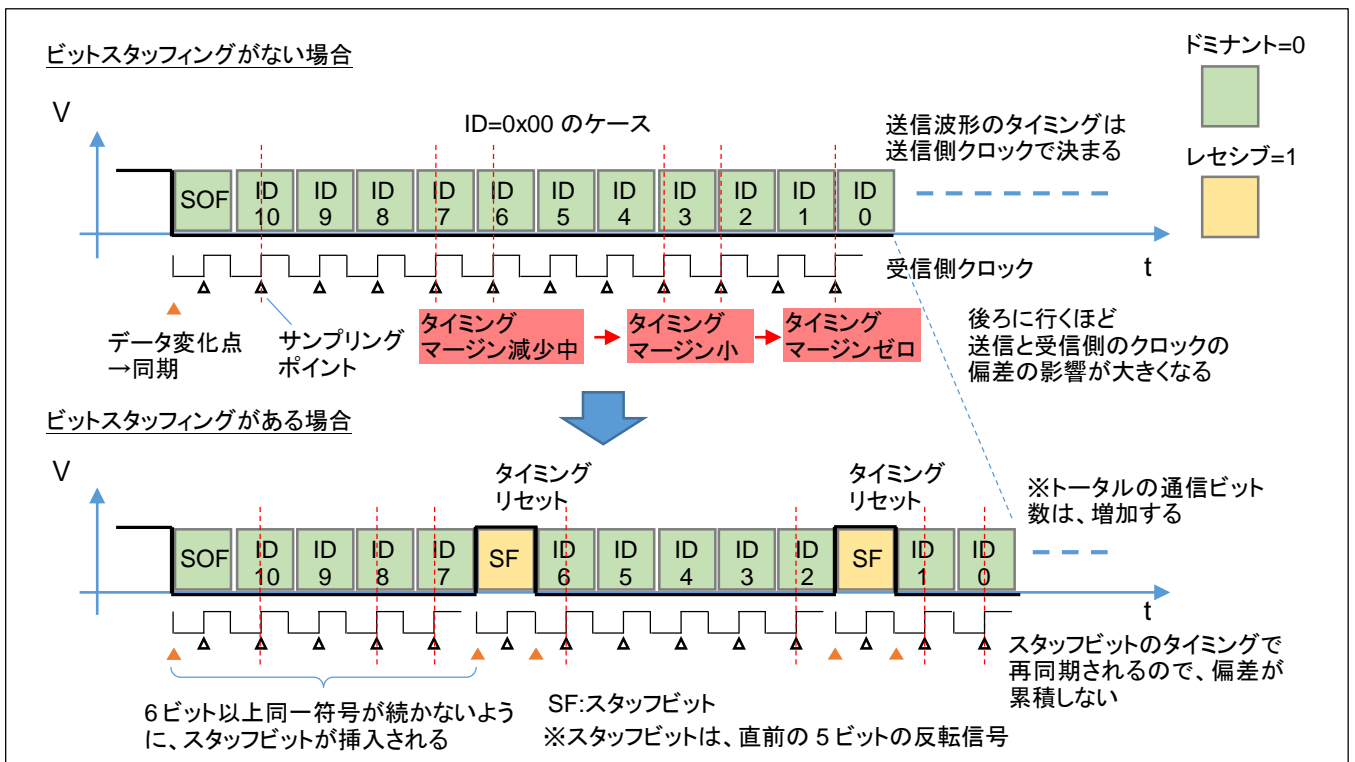


図 4-8 ビットスタッフィング

スタッフビットが挿入された場合は、全体的なデータビット数が増えます。

※ビットスタッフィングの意図

同一符号が続いた場合、データの変化点が存在しません。

CAN は、クロック並走(データ信号とクロック信号を使用してデータ送信を行う)の系ではなく、データ線のみでデータを送る方式です。また、1ビット送信毎に0に戻る RTZ(Return to Zero)符号でもなく、電圧レベルの変化点=データの変化点となる系です。送信側は送信側のクロック信号を使って波形を送出し、受信側は受信側のクロック信号(送信側とは独立)を使ってデータを取り込みます。受信側はデータ変化点を基準にタイミングをリセットする事で、受信側と送信側のクロックの偏差が累積する事はありません(クロック偏差が5ビット時間のタイミングでリセットされます)。

※図ではサンプリングポイントをビットの中央付近(50%)に描いていますが、通常は中央より後ろに設定します

### 1.6. 1ビットとTq

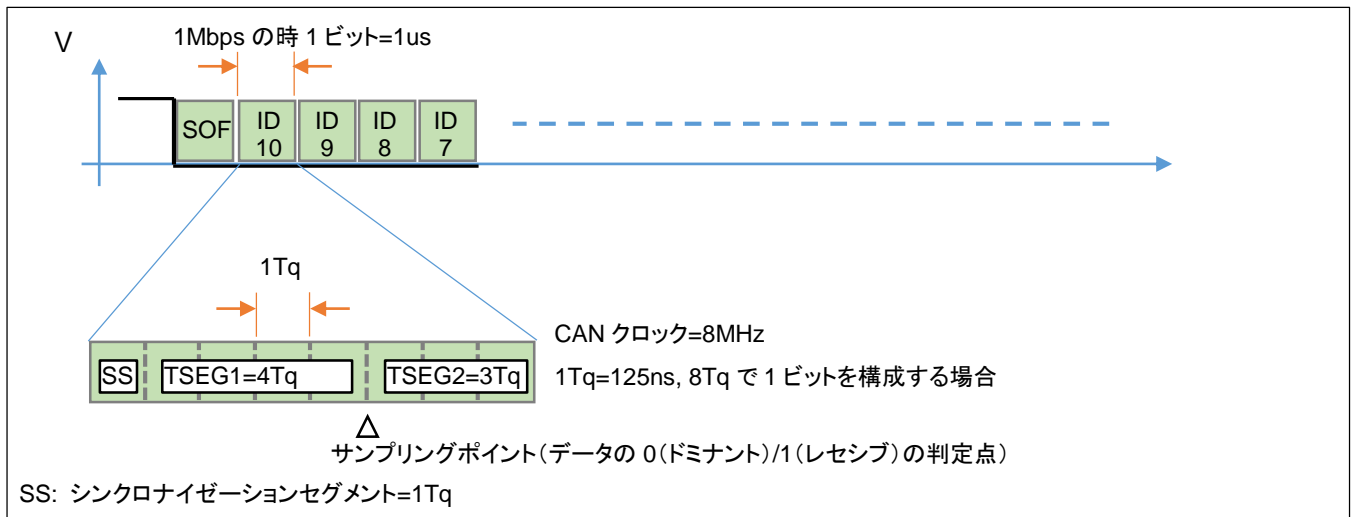


図 1-9 1ビットとTqの関係

CAN の1ビットは複数のTq(最小時間単位)を組み合わせて構成されます。CAN クロックが8MHzの場合、8Tqで1ビットを構成すれば、1ビットが125ns(=Tq)×8=1usとなり、データレートは1Mbpsとなります。

(例えば、CANのクロックを12.5MHzとした場合、12Tqで1ビットを構成した場合、1.04Mbpsとなり、13Tqで1ビットを構成した場合、0.962Mbpsとなり、1Mbpsに設定する事はできません。12.5MHz, 25Tqで1ビットを構成すれば、500kbpsには設定可能です。)CANのクロックと何Tqで1ビットを構成するかで、ビットレートが決まります。なお、ビットレートは、通信相手と合わせる必要があります。

実際の設定では、

- (1)クロックの分周比を決めて1Tqの値を決める
- (2)TSEG1とTSEG2の値を何Tqとするかを決める

(1),(2)の設定値をマイコンのレジスタに設定します。

## 1.7. CAN の ID

CAN は、データの送信時、ID コードをデータに含める規格となっています。ID の設定は、システムの設計者が決める事ですので、

- ・各ユニット毎に 1 つの ID を重複しないように割り振る
- ・各ユニット毎に、送信用 ID と受信用 ID を別々に割り振る
- ・データの種類によって ID を割り振る(例えば、ライトは 0x001, ワイパーは 0x002, パワーウィンドウは 0x003 等)等、色々な手法が考えられます。

送信データパケットには ID が含まれますので、

- ・データ送信を行う際は何らかの ID を付与して送信する必要がある
  - ・データを受信する側はデータパケットに含まれる ID を見て「受信しない」「受信する」を決められる(ID でフィルタリングができる)
- という事となります。

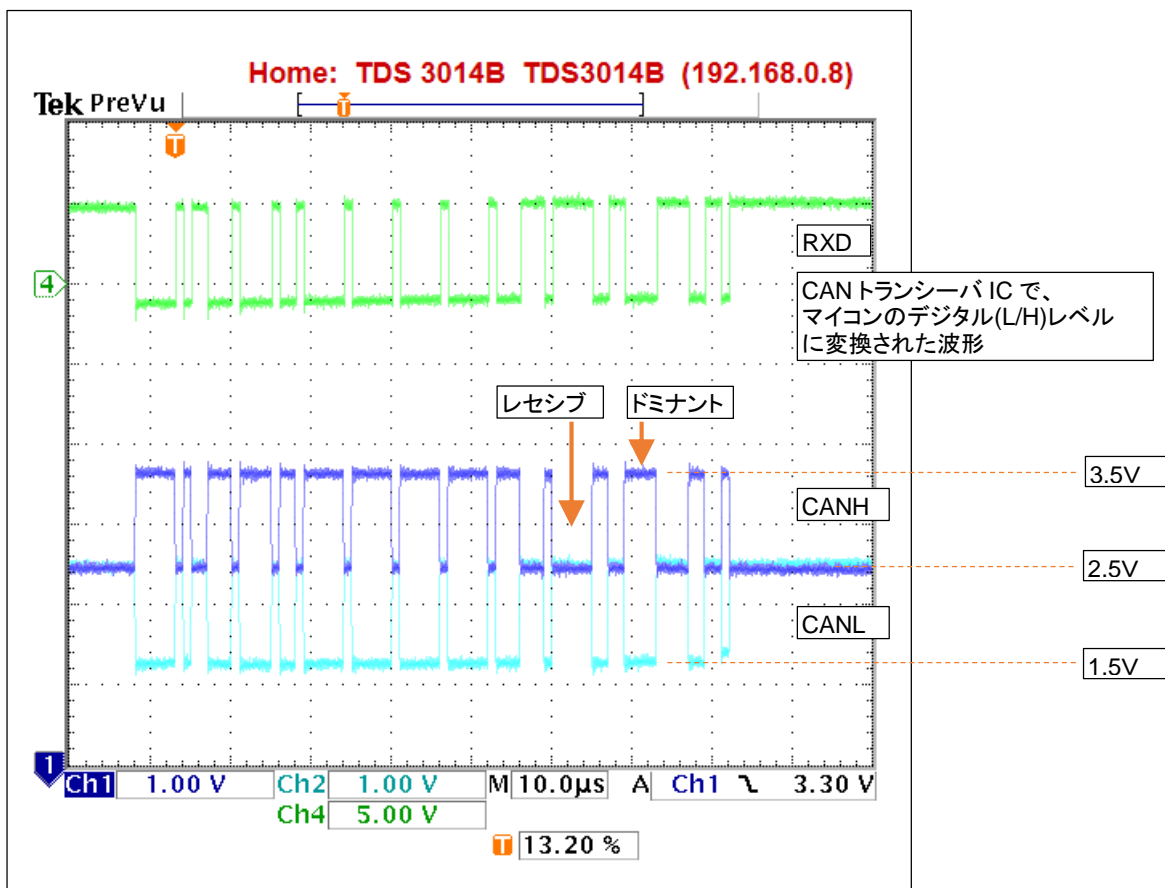
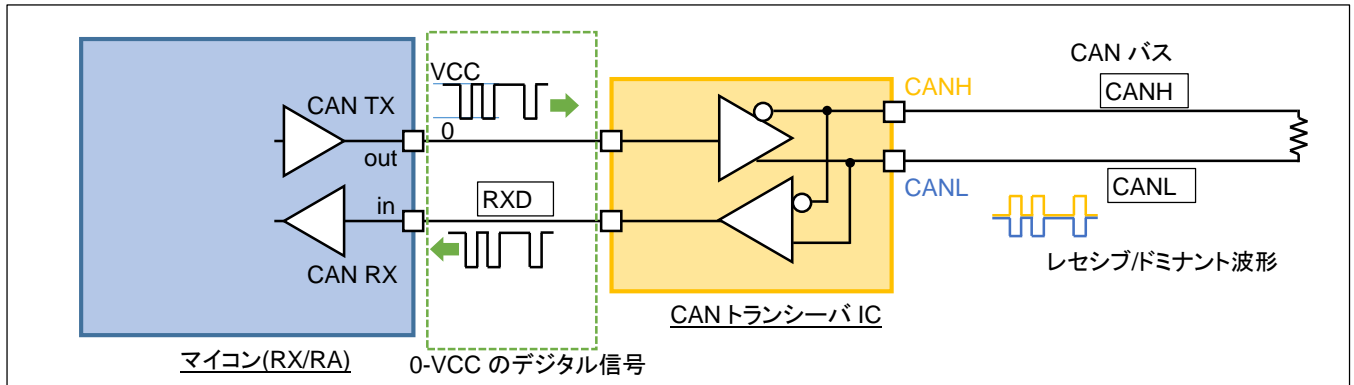
データフレームでは、送信ユニットの(自局の)ID。リモートフレームでは、データを返信して欲しいユニットの(相手先の)ID を CAN のフレームに埋め込んで送信します。

ルネサスのマイコン内蔵 CAN モジュールどのタイプのモジュールでも、送信時に任意の ID を設定する事や、特定の ID のデータのみ受信を行う設定ができます。

ID は、標準フォーマットでは、11bit, 拡張フォーマットでは 29bit となっています。  
(本キットのサンプルプログラムでは、デフォルトで拡張フォーマットに設定していますが、標準フォーマットへの切り替えができる様にしています。)

## 1.8. 通信時の実際の波形例

下記に、CANで通信を行った際の、CANバスと、CANトランシーバを通した後の波形を示します。(標準フォーマット、データフレーム)



CANバスでは、レセプブ(CANHとCANLが同電位に近い)とドミナント(CANHとCANLの電位差が付く)の波形となりますが、CANトランシーバの受信回路を通すとマイコンが受けられるデジタル波形に変換されます。

## 2. LIN 通信の概要

### 2.1. 接続形態

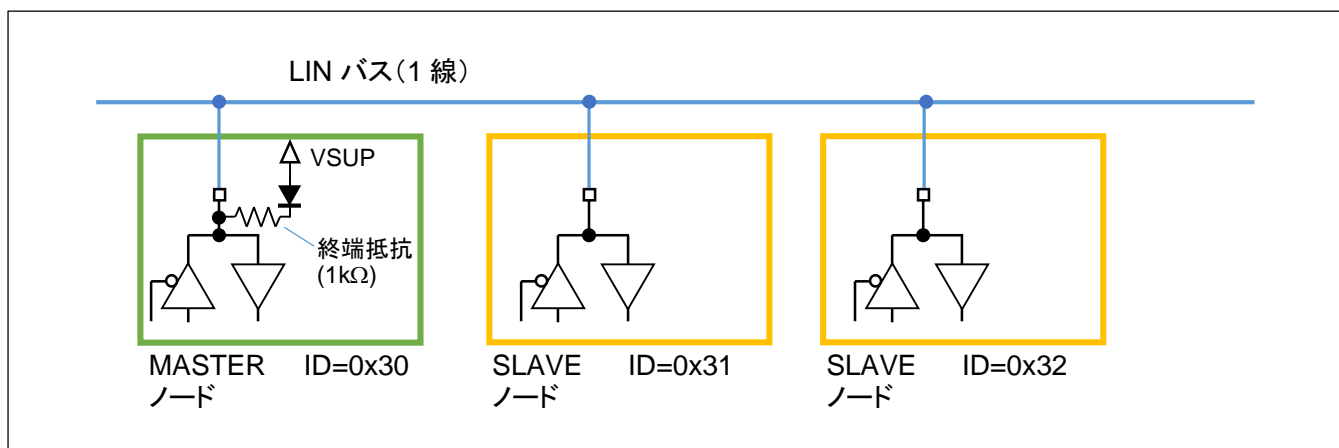


図 2-1 LIN 接続形態

VSUP は基本的には自動車のバッテリーの電源(12~14V)となります

LIN は、1 線の LIN バスのラインに複数のノードがぶら下がる形となり、以下のような形態を取ります。

- ・LIN バスは 1 線の信号ラインとなる
- ・1 本の信号線で、データの送受信を行う
- ・MASTER ノードは、1 つのバスに 1 つのみ
- ・SLAVE ノードは、1 つのバスに複数存在してよい
- ・バスの終端は MASTER ノードで行う
- ・各ノードは重複しない ID を有する

### 2.2. LIN の物理層

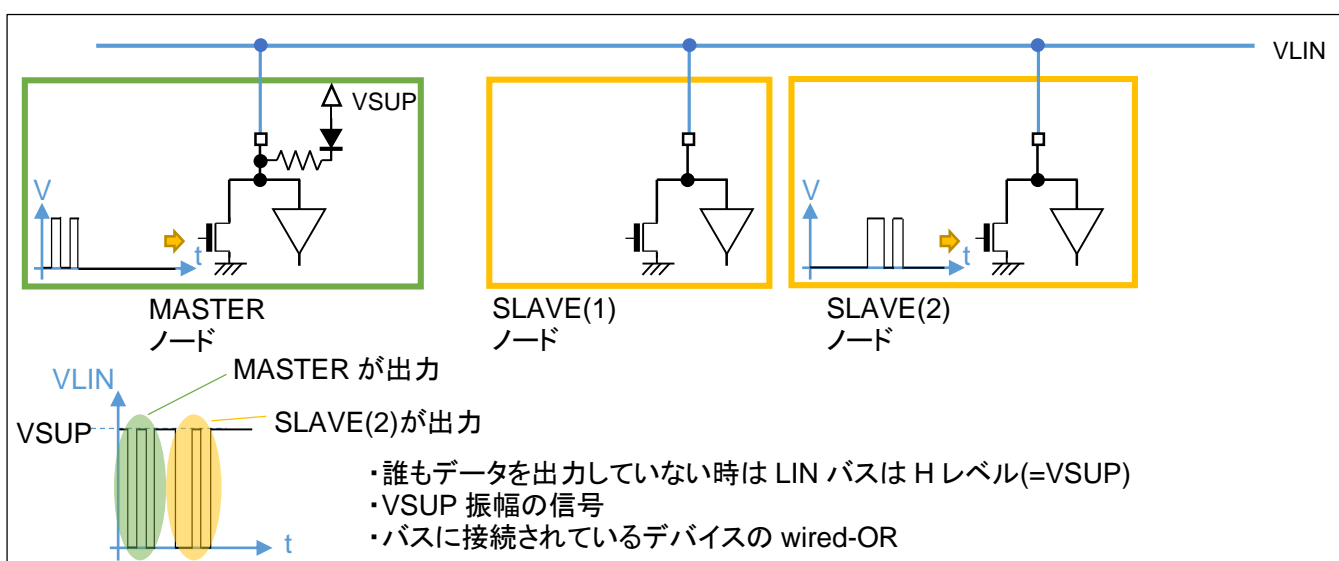


図 2-2 LIN 物理層



LIN バスの信号は、各ノードの出力の wired-OR(いずれかのノードが L を出力すると、LIN バスは L)となります。そのため、各ノードが自由なタイミングで出力を始めると、信号衝突が起こるため、出力を行うタイミングは MASTER が制御する事となります。CAN と異なり、データの衝突は許されていません。また、LIN バスの振幅は、VSUP 振幅となります。

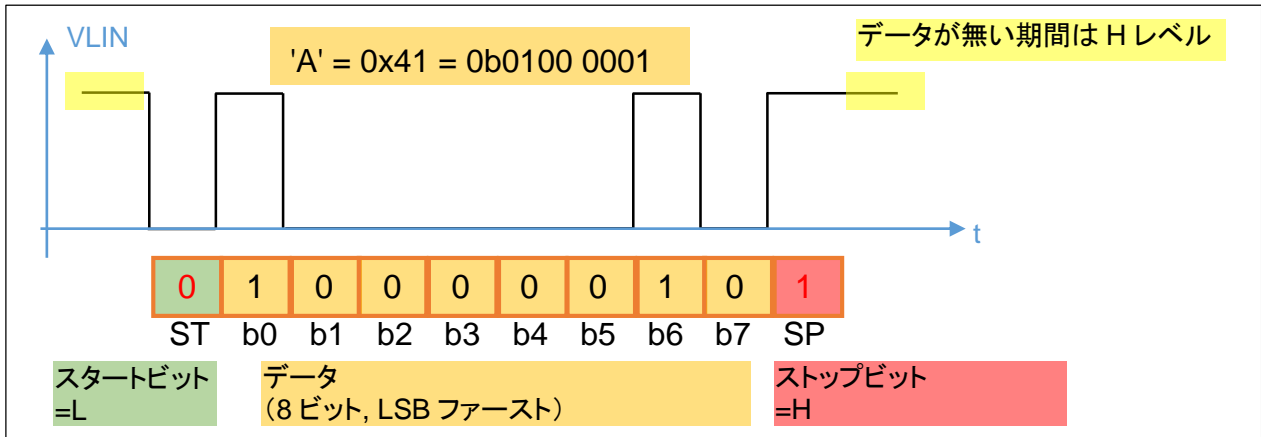


図 2-3 LIN の 1 バイト

LIN で 1 バイト(A=0x41)を送信する場合の波形は上記の様になります。スタートビット=0、ストップビット=1 が付与され、データは LSB(下位ビット)から順に送信されます。このフォーマットは、UART(調歩同期式通信)と同様です。また、LIN の通信速度は一般的には、~20kbps です。

### 2.3. LIN のデータパケット

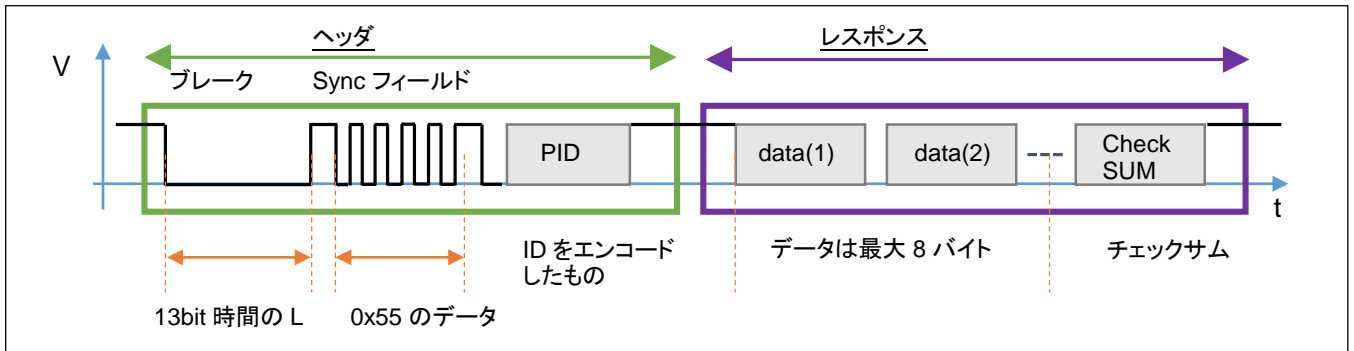


図 2-4 LIN データパケット

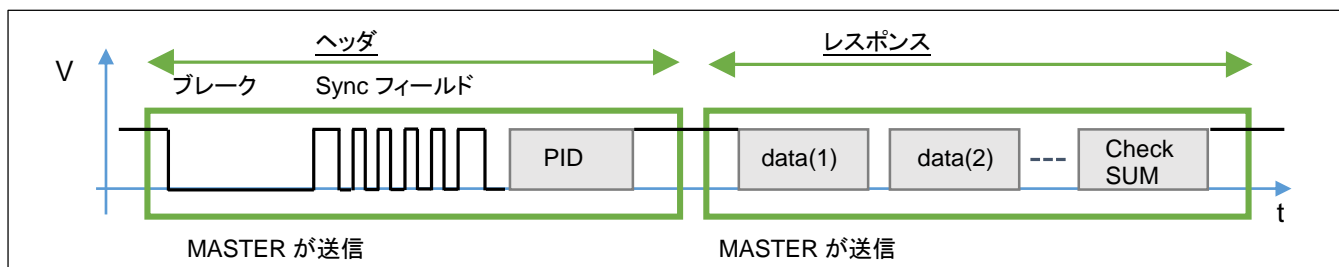
LIN のデータパケットは、ヘッダとレスポンスから構成され、以下の特徴を持ちます。

- ・ヘッダを送出するのは MASTER ノードのみ
- ・レスポンスは、MASTER か SLAVE ノードが送出する
- ・SLAVE ノードは、ヘッダに含まれる ID が自ノードの ID と一致した場合に、レスポンスを返す事ができる
- ・レスポンスデータは、データとチェックサムで構成される
- ・バスに流れているデータは、どのノードでも受信可能

ヘッダは、ブレーク(13bit 時間(以上)の L)、Sync フィールド(0x55 のデータ)、PID(Protected ID)から構成されます。

ブレイクは信号開始の合図。Sync フィールドは、受信側に通信速度を同期させる役割(受信側は Sync フィールドのパルス幅を測定して、送信側の通信速度に合わせる事ができる)があり、PID は ID をエンコード(ID にチェックサムを付与したデータ)となっています。

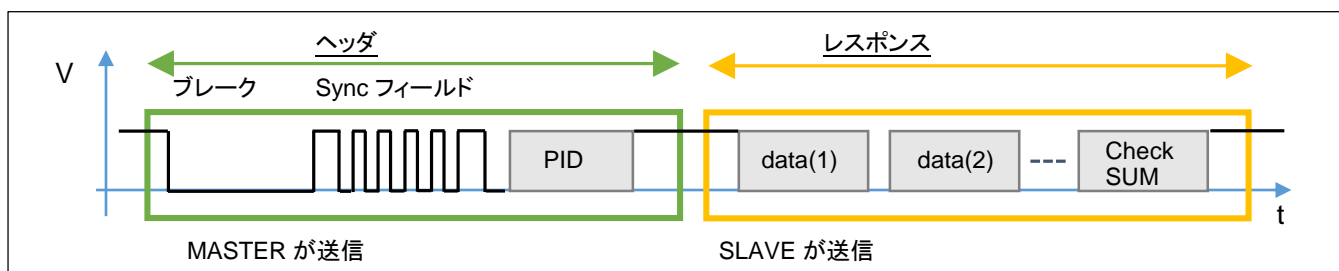
・MASTER レスポンス送信



**図 2-5 MASTER レスポンス送信**

MASTER レスポンス送信では、ヘッダ、レスポンスとも MASTER ノードが送信を行います。SLAVE ノードでは、このデータを受信可能です (SLAVE 側は受信動作のみ)。

・SLAVE レスポンス送信



**図 2-6 SLAVE レスポンス送信**

SLAVE レスポンス送信では、ヘッダは MASTER ノードが出力しますが、レスポンスは SLAVE ノードが送信を行います。SLAVE ノードでは、ヘッダに含まれる PID (ID コードをエンコードしたもの) が自分自身の ID である場合に、レスポンス送信を行います。MASTER 側はレスポンスデータを受信し、SLAVE がどのようなデータを送信したのかを知ることができます。

### 3. UART 通信の概要

#### 3.1. 接続形態

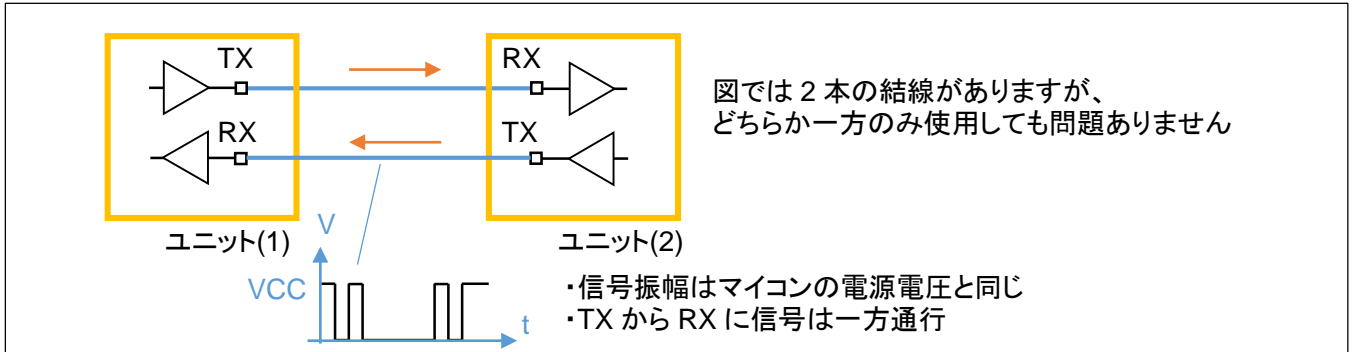


図 3-1 UART 接続形態

UART は、TX と RX が基本的には 1:1 で接続される形態を取ります。(1 本の TX に複数の RX が接続される形でも問題ありません)

- ・TX(送信端子)とRX(受信端子)が接続
- ・データの流は、TX から RX の方向のみ
- ・全二重通信(TX からデータ送信中に RX に来たデータを受信)が可能
- ・通信を行うユニット間で通信速度、通信条件を合わせる必要がある

#### 3.2. UART のデータ波形

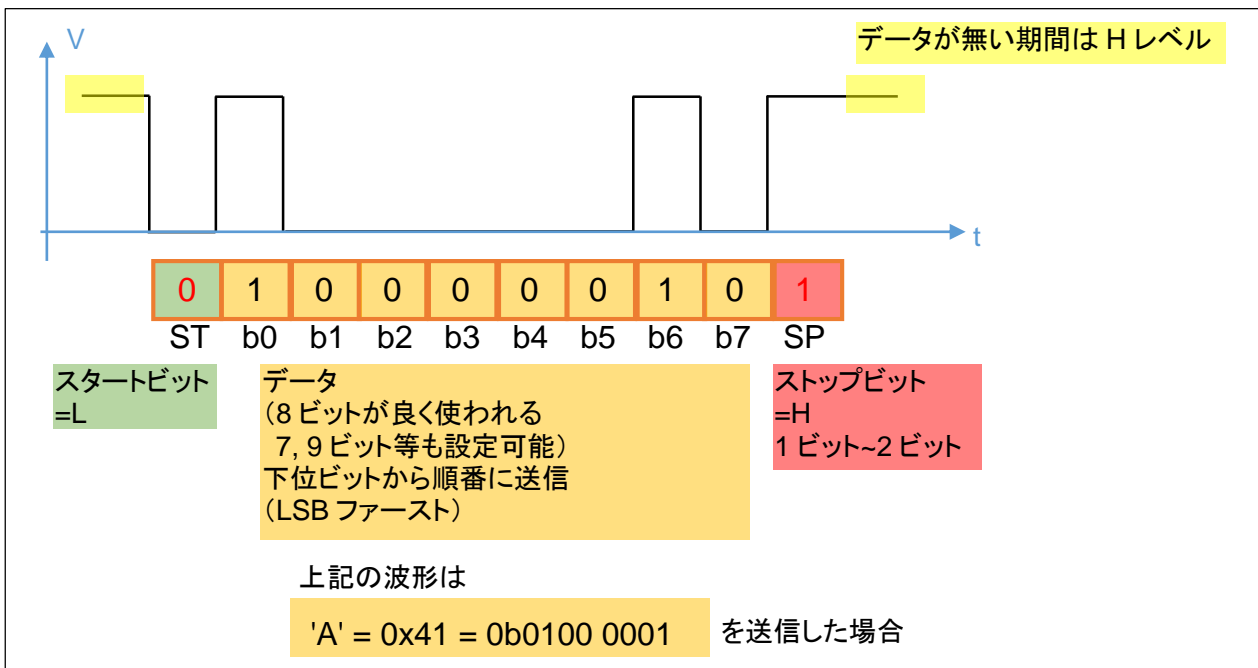


図 3-2 UART の 1 データ

データ 8 ビット、パリティなし、ストップビット=1 でデータを構成した場合の波形例を示します。スタートビットは必ず 0。ストップビットは必ず 1 となり、10 ビットで 1 データ(1 バイト)を送信します。データは、下位ビット(b0)から上位ビット(b7)の順番で送信されます(これを LSB ファーストといいます)。

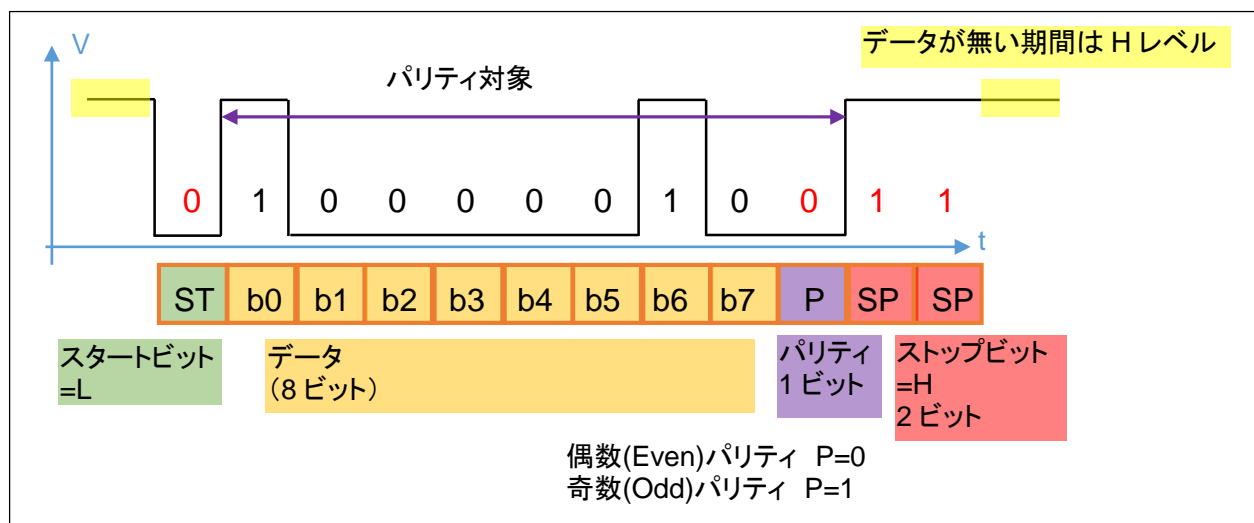


図 3-3 UART の 1 データ(別バージョン)

データ 8 ビット、偶数パリティ、ストップビット=2 でデータを構成した場合の波形例を示します。偶数パリティは、パリティ対象期間の 1 の数が偶数になる様にパリティビットが付与されます(上例では、パリティ=0)。奇数パリティの場合は、b0~b7 までの 1 の数が 2(偶数)なので、パリティビットは 1 となります。(その場合、パリティ対象期間中の 1 の数が奇数(3)となる。)

パリティはエラー検出に使用され、有効にした場合はパリティビットが合っていないと受信側で受信エラーの検出が行えます。上記例では、1 データ(1 バイト)送信するのに、12 ビットとなります。

### 3.3. UART の通信速度

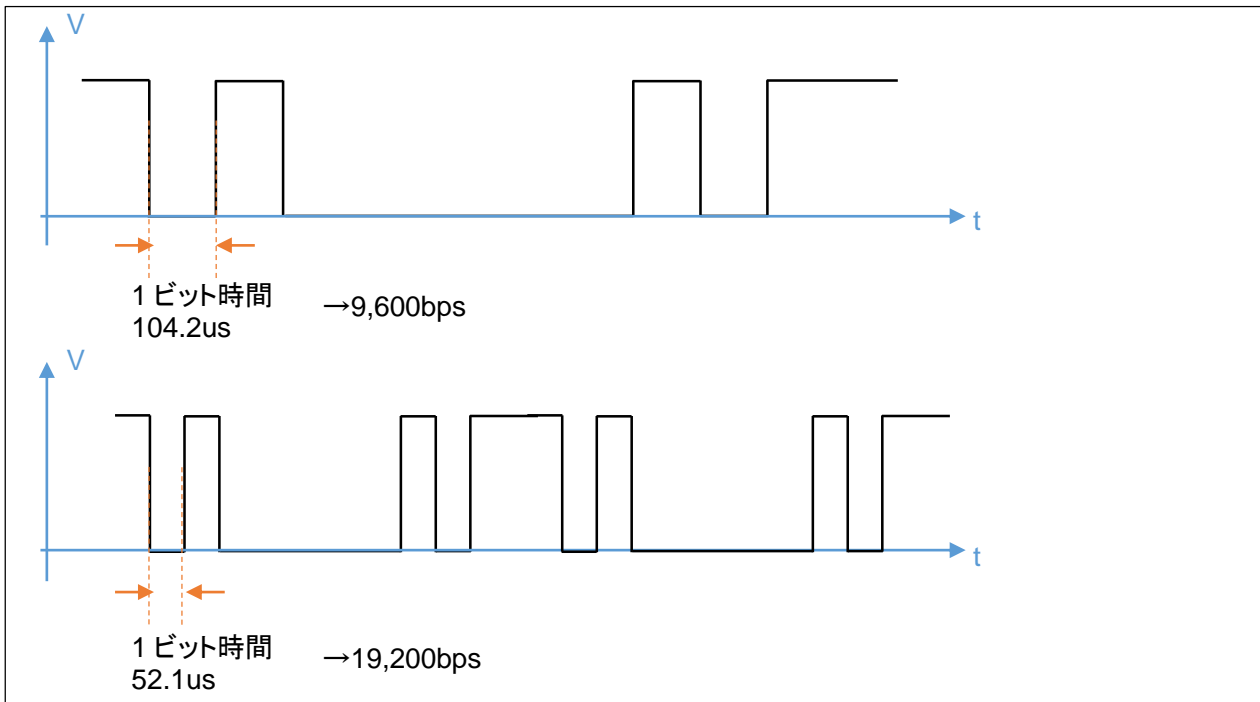


図 3-4 通信速度

通信速度は、1 秒間に何ビット送信するかで表します。1 秒間に 9,600 ビット送信する速度は、9,600bps。9,600bps の場合、1 ビットあたり 104.2us。10 ビット(1 文字)送信で、約 1.04ms 掛ります。

9,600bps、19,200bps、38,400bps、57,600bps、115,200bps 等が良く使用される通信速度です。通信速度は、送信側と受信側で同じ速度に設定を行う必要があります。

UART では、

- (1)通信速度(9,600bps や 115,200bps 等)
- (2)データ 8 ビット
- (3)パリティ(なし、偶数パリティ、奇数パリティ)
- (4)ストップビット(1 または 2 ビット)

の 4 つのパラメータを決めて、通信を行う 2 者で同じ値に設定する必要があり、設定値が異なると文字(データ)化けが起こります。

## 4. マイコンボード上で実行されるデモプログラム (CAN 通信)

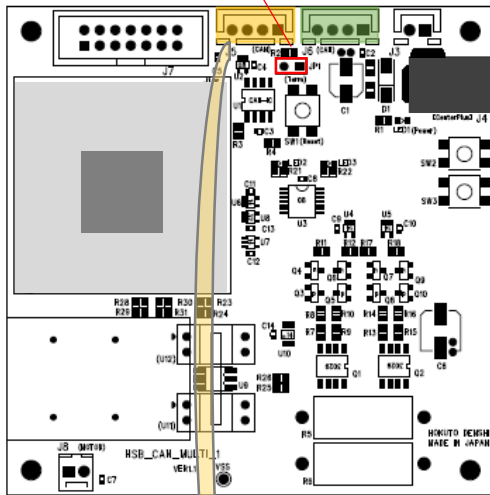
出荷時、マイコンボードにはデモプログラムが書き込まれており、電源を投入すると所定の動作を行います。

### 接続例

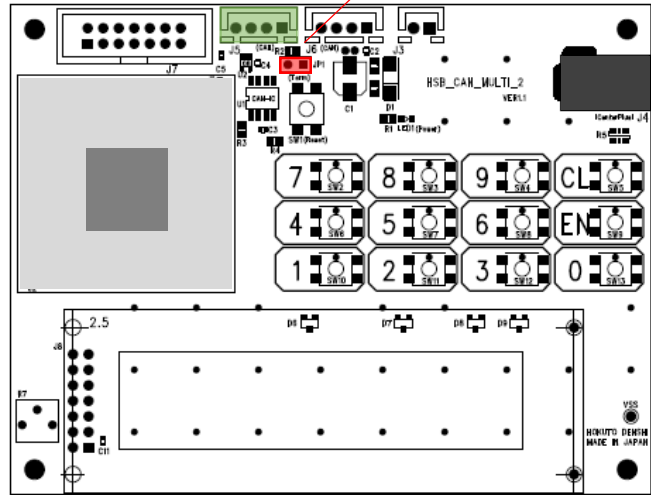
JP1 のジャンパを抜く

CAN 通信ケーブル

JP1 にジャンパを挿す



HSB\_CAN\_MULTI\_1

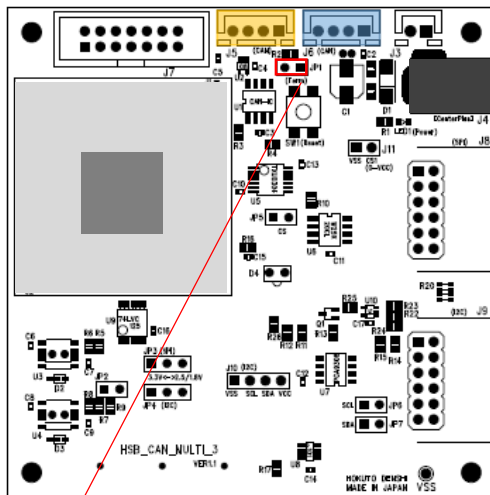


HSB\_CAN\_MULTI\_2

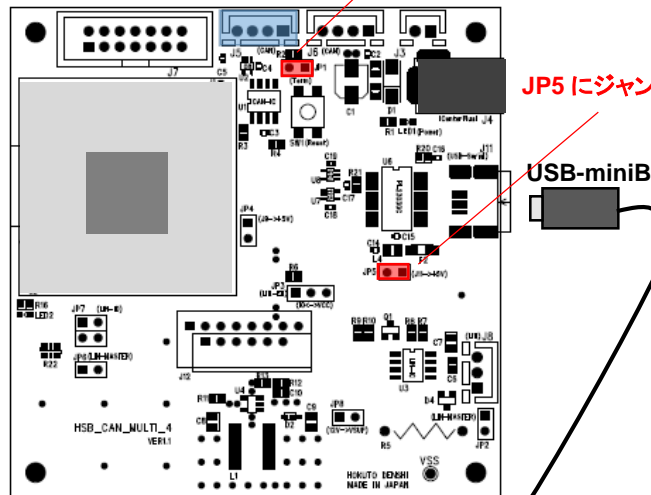
CAN 通信ケーブル

CAN 通信ケーブル

JP1 にジャンパを挿す



HSB\_CAN\_MULTI\_3



HSB\_CAN\_MULTI\_4

JP1 のジャンパを抜く

JP5 にジャンパを挿す

USB-miniB

USB-miniB ケーブル  
[PC に接続]

CAN のデータは、

- ・拡張 ID フォーマット
- ・データ長 8 バイト固定(8 バイトに満たないデータは 0x00 を付与)
- ・通信速度 1Mbps

でやり取りを行っています。それぞれ、定義ファイルで上記値を定義しているので、プログラムを再ビルドすれば設定を変える事は可能です。

各ボードには、それぞれ 2 種類の CAN の ID が割り振られています。

ボード	CAN-ID(1)	CAN-ID(2)
HSB_CAN_MULTI_1	0x101	0x201
HSB_CAN_MULTI_2	0x102	0x202
HSB_CAN_MULTI_3	0x103	0x203
HSB_CAN_MULTI_4	0x104	0x204

CAN-ID(1)は、各ボードが自発的に情報を外部に送信する用途で使用しています。

CAN-ID(2)は、各ボードに対する指示、各ボードにデータを出力させる用途で使用しています。

## 4.1. CAN-ID(1)で出力するデータ

電源を投入すると、各ボードは 500ms に 1 回、CAN バスに対してデータを出力します。

### —送信データ—

ボード	ID	RTR	IDE	DLC	data(0x)								
HSB_CAN_MULTI_1	0x101	0b0	0b1	0b1000	XX	YY	YY	00	00	00	00	00	00
HSB_CAN_MULTI_2	0x102	0b0	0b1	0b1000	XX	YY	YY	00	00	00	00	00	00
HSB_CAN_MULTI_3	0x103	0b0	0b1	0b1000	XX	XX	YY	YY	00	00	00	00	00
HSB_CAN_MULTI_4	0x104	0b0	0b1	0b1000	00	00	00	00	00	00	00	00	00

RTR はデータフレーム／リモートフレーム区分で、ここで出力するのはデータフレーム(0)としています。IDE は標準 ID フォーマット／拡張 ID フォーマット区分で、ここでは拡張 ID フォーマット(1)としています。DLC はデータバイト数で、本プログラムでは、どのデータも 8 バイト固定値としてるので、常に 8(=0b1000)を送信します。data は、8 バイトで上表で 00 となっている部分は 0x00 固定。XX, YY となっている部分は、ボードの情報を出力しています。

### —送信データのデータパケット部分—

ボード	バイト位置	送信内容	備考
HSB_CAN_MULTI_1	1 バイト目	モータ duty 値[%]	-100~100[%]: 符号付き 1 バイト 50%→0x32 100%→0x64 -50%→0xCE -100%→0x9E
	2~3 バイト目	モータ回転数[rpm]	モータの回転数: 符号付き 2 バイト 5000rpm→0x1388 -5000rpm(逆回転)→0xEC78

ボード	バイト位置	送信内容	備考
HSB_CAN_MULTI_2	1 バイト目	LCD の表示内容	0x00:HSB_CAN_MULTI_1~4 の情報を表示 (5 秒毎にページ切り替え) 0x01: HSB_CAN_MULTI_1~2 の情報を表示 0x02: HSB_CAN_MULTI_3~4 の情報を表示 0x03: 外部から送信されたメッセージを表示
HSB_CAN_MULTI_2	2~3 バイト目	マトリックスキーの情報	0~9, CL, EN のキー(12 キー)を、12bit+ 上位 4bit=0 の 16bit(2 バイト)で送信 押している場合は 1 (*1)
HSB_CAN_MULTI_3	1~2 バイト目	明るさセンサの値	0~4095 の A/D 変換結果を 2 バイトで送信 (0x0000~0x0FFF) ※RL78/F15 の場合は 0~1023(0x0000~0x03FF)
	2~3 バイト目	温度センサの温度値	温度(摂氏)値の 100 倍を 2 バイトで送信 23.5℃の場合、100 倍した 2350→0x092E を送信
HSB_CAN_MULTI_4	1~8 バイト目	0x00 × 8	存在確認のための送信

HSB\_CAN\_MULTI\_4 は、送信データは 0x00 × 8 となりますが、これはボードがネットワーク上に存在している存在確認のために送っています。(他のボードからは、ネットワーク上の HSB\_CAN\_MULTI\_4 が接続されている事が認識できます。)

(\*1)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY					CL	EN	9	8	7	6	5	4	3	2	1	0
data(0b)	0	0	0	0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

例えば、キーの 0 とキーの 2 を押している時は、0x0005 を送信する

## 4.2. CAN-ID(2)でやり取りされるデータ

CAN-ID(2)は、ボードに対する指示とボードに対するデータ出力要求に使用されます。

各ボードは、CAN-ID(2)でデータフレームを受信した場合、自分自身のボードに対する指示であると認識します。データパケットに含まれる内容に従い動作します。

CAN-ID(2)でリモートフレームを受信した場合は、リモートフレームに応答する形で、各種データを出力します。リモートフレームは、外部から当該ボードに対し、データを出力する様に要求するコマンドとして作用します。

### 4.2.1. データフレーム(外部からの指示)

—受信データ—

ボード	ID	RTR	IDE	DLC	data(0x)								
HSB_CAN_MULTI_1	0x201	0b0	0b1	0b1000	XX	YY	00	00	00	00	00	00	00
HSB_CAN_MULTI_2	0x202	0b0	0b1	0b1000	XX	YY	YY	YY	YY	YY	YY	00	00
HSB_CAN_MULTI_3	0x203	0b0	0b1	0b1000	XX	YY	YY	YY	YY	YY	YY	YY	YY

HSB\_CAN\_MULTI\_4 は、CAN-ID(2)=0x204 のデータフレームを受け取っても無反応です。



XX の部分はコマンド(1 バイト)、YY の部分にはデータが入ります。

—受信データのデータパケット部分—

ボード	1 バイト目 (コマンド)	2 バイト目~	備考
HSB_CAN_MULTI_1	0x01	YY -----	モータの duty 変更コマンド 2 バイト目の数値を現在の duty に反映させる 3~8 バイト目は未使用(どのようなデータを受け取っても無関係)

ボード	1 バイト目 (コマンド)	2 バイト目~	備考
HSB_CAN_MULTI_2	0x02	YY -----	LCD の情報表示内容変更コマンド 2 バイト目の値で、表示内容を変更 YY=0: HSB_CAN_MULTI1 ~4 の情報を 5 秒毎に 2 ページに分けて表示 YY=1: HSB_CAN_MULTI1~2 の情報を表示 YY=2: HSB_CAN_MULTI3~4 の情報を表示 YY=3: 外部から送信されたデータを表示 3~8 バイト目は未使用
	0x03	YY ZZ ZZ ZZ ZZ ---	LCD に表示するデータを受信 16 桁×2 行のデータを 8 分割して、4 バイトずつ受信 YY=0~7: LCD のページ番号 ZZ ZZ ZZ ZZ: 表示させるデータ(1 回の CAN パケットで 4 バイト) 7~8 バイト目は未使用
HSB_CAN_MULTI_3	0x04	-----	センサ値読み出しコマンド リモートフレーム応答(後述)で、センサ値を送信する 2~8 バイト目は未使用
	0x05	YY YY YY -----	SPI フラッシュ読み出しコマンド adr=YYYYYY のデータをリモートフレーム応答で送信 5~8 バイト目は未使用
	0x06	YY YY YY ZZ ZZ ZZ ZZ	SPI フラッシュ書き込みコマンド addr=YYYYYY に ZZ ZZ ZZ ZZ(4 バイト)のデータを書き込む
	0x07	-----	SPI フラッシュ消去コマンド SPI フラッシュの全消去を行う 2~8 バイト目は未使用
	0x08	YY -----	SPI フラッシュ記録モード設定 YY=0: センサデータを SPI フラッシュに記録停止 YY=1: センサデータを SPI フラッシュに記録開始 3~8 バイト目は未使用

例えば、ID=0x202、data=0x02 0x01 0x00 0x00 0x00 0x00 0x00 0x00 というパケットを CAN バスに流した場合、HSB\_CAN\_MULTI\_2 のボードが自分自身への指令であると認識して、LCD の表示内容を変更します。その他、上表のコマンドに対して、各ボードは指令を受け付けます。

操作例として、HSB\_CAN\_MULTI\_2 の LCD ディスプレイに

"ABCDEFGH IJKLMN OP"

"abcdefghijkl mnop"

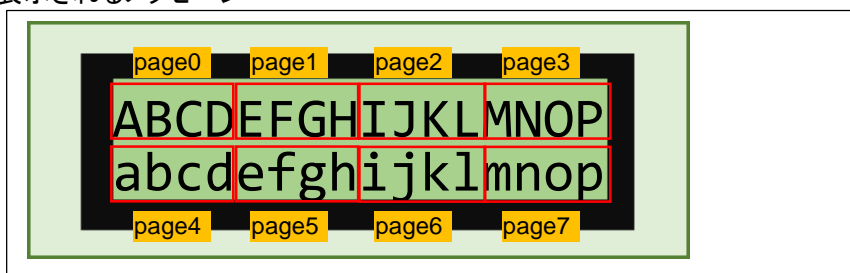
と表示させる場合の、CAN バスに流すデータを示します。

ID	RTR	IDE	DLC	data(0x)								動作
0x202	0b0	0b1	0b1000	02	03	00	00	00	00	00	00	LCD を外部データ表示に切り替え
0x202	0b0	0b1	0b1000	03	00	42	42	43	44	00	00	"ABCD"を送信(ページ 0)
0x202	0b0	0b1	0b1000	03	01	45	46	47	48	00	00	"EFGH"を送信(ページ 1)
0x202	0b0	0b1	0b1000	03	02	49	4A	4B	4C	00	00	"IJKL"を送信(ページ 2)
0x202	0b0	0b1	0b1000	03	03	4D	4E	4F	50	00	00	"MNOP"を送信(ページ 3)
0x202	0b0	0b1	0b1000	03	04	61	62	63	64	00	00	"abcd"を送信(ページ 4)
0x202	0b0	0b1	0b1000	03	05	65	66	67	68	00	00	"efgh"を送信(ページ 5)
0x202	0b0	0b1	0b1000	03	06	69	6A	6B	6C	00	00	"ijkl"を送信(ページ 6)
0x202	0b0	0b1	0b1000	03	07	6D	6E	6F	70	00	00	"mnop"を送信(ページ 7)

16 桁 × 2 行の全データを送信、表示させるためには、合計 9 パケットの送信が必要です。

(16 行 × 2 桁を 8 つのページに区切っていますので、表示を変えたい部分のデータのみ送信する事も可能。)

LCD に表示されるメッセージ



#### 4.2.2. リモートフレーム(外部からデータを出力する様に要求)

##### —受信データ—

ボード	ID	RTR	IDE	DLC
HSB_CAN_MULTI_1	0x201	0b1	0b1	0b1000
HSB_CAN_MULTI_2	0x202	0b1	0b1	0b1000
HSB_CAN_MULTI_3	0x203	0b1	0b1	0b1000
HSB_CAN_MULTI_4	0x204	0b1	0b1	0b1000

リモートフレームの場合、CANのパケットのRTRの部分が1となり、データフィールドはなくなります。DLCの値(0b1000=8)が、送って欲しいデータバイト数になります(DLCの値は定義値で8バイト固定です)。HSB\_CAN\_MULTI\_1~HSB\_CAN\_MULTI\_4の各ボードは、リモートフレーム要求を受け取った際、データの返送(応答)を行います。

##### —返送データ—

ボード	ID	RTR	IDE	DLC	data(0x)								
HSB_CAN_MULTI_1	0x201	0b0	0b1	0b1000	XX	YY	YY	00	00	00	00	00	00
HSB_CAN_MULTI_2	0x202	0b0	0b1	0b1000	XX	YY	YY	00	00	00	00	00	00
HSB_CAN_MULTI_3	0x203	0b0	0b1	0b1000	YY	YY	YY	YY	YY	YY	YY	YY	YY
HSB_CAN_MULTI_4	0x204	0b0	0b1	0b1000	00	00	00	00	00	00	00	00	00

IDはリモートフレーム要求と同じIDを使用します。リモートフレーム要求に対する応答は、データフレームとなるので、RTR=0です。

HSB\_CAN\_MULTI\_1, HSB\_CAN\_MULTI\_2, HSB\_CAN\_MULTI\_4に関しては、4.1記載のデータと同じものを送信します。

HSB\_CAN\_MULTI\_3は、電源投入後や、「センサ値読み出しコマンド」の後は、4.1記載のデータと同じものを送信します。「SPIフラッシュ読み出しコマンド」の後は、SPIフラッシュに書き込まれているデータを送信します。

例えばSPIフラッシュに下記のデータが書き込まれているとします。

addr	data															
0x0	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x10	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
0x20	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F

ID	RTR	IDE	DLC	data(0x)											動作	
<b>0x203</b>	<b>0b0</b>	<b>0b1</b>	<b>0b1000</b>	<b>05</b>	<b>00</b>	<b>00</b>	<b>10</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	SPIフラッシュ読み出しコマンド addr=0x000010
<b>0x203</b>	<b>0b1</b>	<b>0b1</b>	<b>0b1000</b>													SPIフラッシュ読み出しデータ要求
0x203	0b0	0b1	0b1000	10	11	12	13	14	15	16	17					0x10~17のデータを送信
<b>0x203</b>	<b>0b1</b>	<b>0b1</b>	<b>0b1000</b>													SPIフラッシュ読み出しデータ要求
0x203	0b0	0b1	0b1000	18	19	1A	1B	1C	1D	1E	1F					0x18~1Fのデータを送信
<b>0x203</b>	<b>0b1</b>	<b>0b1</b>	<b>0b1000</b>													SPIフラッシュ読み出しデータ要求
0x203	0b0	0b1	0b1000	20	21	22	23	24	25	26	27					0x20~27のデータを送信

※太字はHSB\_CAN\_MULTI\_3以外からのパケット、太字以外はHSB\_CAN\_MULTI\_3からの送信

SPI フラッシュの読み出しは、SPI フラッシュ読み出しコマンドでアドレスを指定し読み出しを行った場合、読み出しアドレスは自動的にインクリメントされる仕様です。(連続したアドレスの読み出しを行う場合は、2 回目以降は、SPI フラッシュ読み出しコマンドを送る必要はありません。)

SPI フラッシュに対しての書き込みコマンドがあり、書き込みも可能ですが、書き込みは消去後未書き込みの領域に対してのみ有効です。(データの上書きはできません)

SPI フラッシュの消去コマンドを送った場合、2Mbit(256kB)の全領域が消去(データとしては 0xFF が読める)状態となります。

SPI フラッシュの記録開始コマンドを送った場合、最初に SPI フラッシュの全消去を行った後、SPI フラッシュの 0 番地から、センサデータの書き込みを開始します。

addr	data															
0x0	XX	XX	YY	YY	XX	XX	YY	YY	XX	XX	YY	YY	XX	XX	YY	YY
0x10	XX	XX	YY	YY	XX	XX	YY	YY	FF	FF	FF	FF	FF	FF	FF	FF
0x20	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

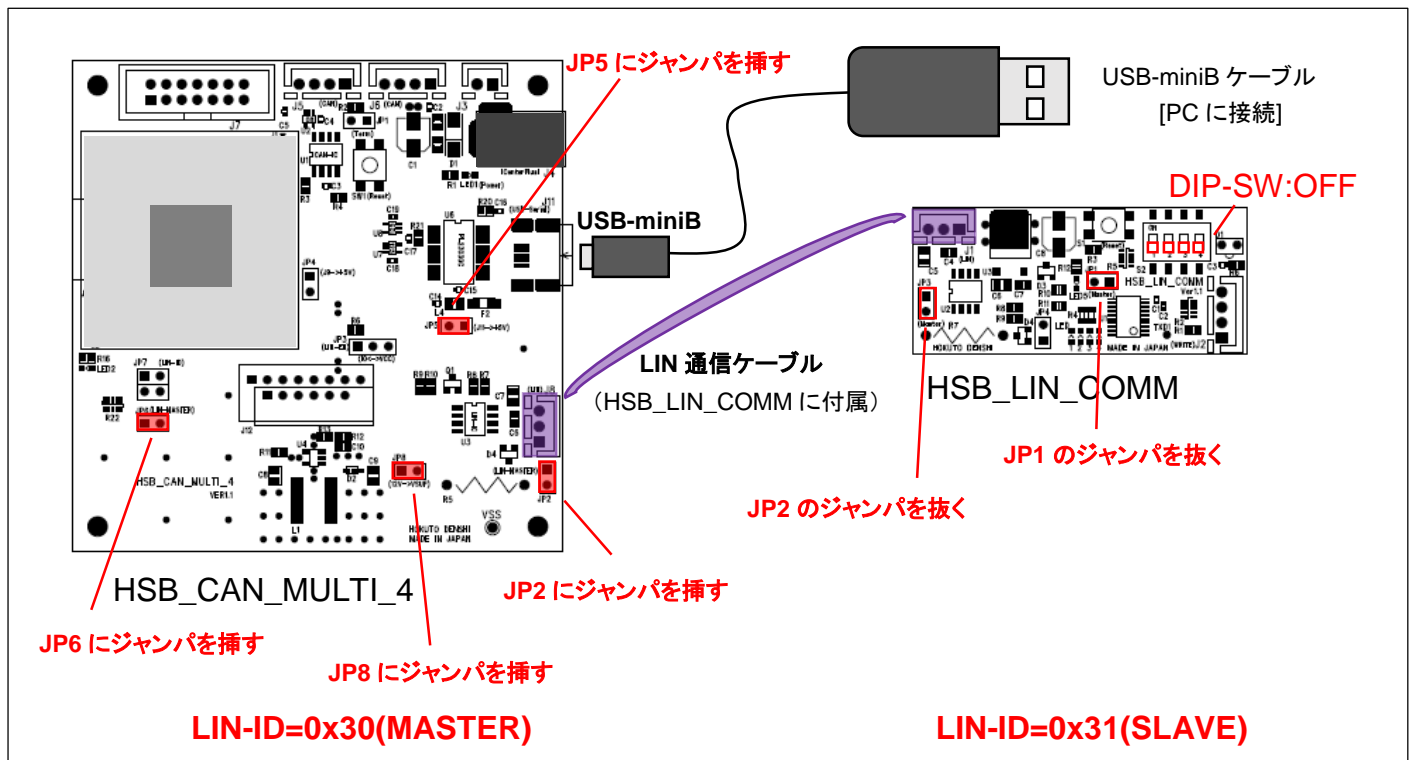
センサデータ書き込み後の SPI フラッシュは上記の様になり、1 回の記録で

XX XX: 明るさセンサの A/D 変換値(0~4095)

YY YY: 温度センサの温度値を 100 倍した値

の 4 バイトの書き込みが行われます。書き込みは、1 秒毎(初期値)に行われます。

## 5. マイコンボード上で実行されるデモプログラム (LIN 通信)



LIN のデータは、

- ・データ 8 バイト (8 バイトに満たないデータは 0x00 を付与)
- ・通信速度 9,600bps

でやり取りを行っています。それぞれ、定義ファイルで上記値を定義しているので、プログラムを再ビルドすれば設定を変える事は可能です。

LIN の ID は、ボードを MASTER に指定した場合は、0x30。SLAVE に設定した場合は、0x31~0x34 の ID が振られる様になっています。

HSB\_CAN\_MULTI\_4 (JP6 をショートに設定し、MASTER モードで動作) : ID=0x30

HSB\_LIN\_COMM (JP1 をオープンに設定し、SLAVE モードで動作)

DIP-SW1	DIP-SW2	LIN-ID
OFF	OFF	0x31
OFF	ON	0x32
ON	OFF	0x33
ON	ON	0x34

DIP-SW は起動時に読み込み。起動後は、設定を変えても LIN-ID は変わりません。

※HSB\_LIN\_COMM を複数台接続する場合は、ID が重複しない様にしてください

HSB\_CAN\_MULTI\_4 を MASTER モードで動作させた場合、LIN の動作はタイマモードとなり、一定時間(100ms)毎に、下記(1)~(5)を繰り返す動作となります。(100ms 毎にヘッダ送信、500ms に1回 MASTER レスポンス送信)

#### —LIN パケット—

	ヘッダ		レスポンス								
	ヘッダ送信元	ID	レスポンス送信元	data(0x)							
(1)	HSB_CAN_MULTI_4	0x30	HSB_CAN_MULTI_4	00	00	00	00	00	00	00	00
(2)	HSB_CAN_MULTI_4	0x31	HSB_LIN_COMM (ID=0x31)	SS	SS	XX	YY	ZZ	ZZ	00	00
(3)	HSB_CAN_MULTI_4	0x32									
(4)	HSB_CAN_MULTI_4	0x33									
(5)	HSB_CAN_MULTI_4	0x34									

100ms 毎に(1)→(2)→(3)→(4)→(5)→(1)…を繰り返す動作です。(上表は、ID=0x32~0x34 のデバイスが存在しない場合。)

HSB\_CAN\_MULTI\_4(MASTER)が MASTER レスポンス送信で送信するデータは、初期値は 0x00 × 8。後述の UART 経由でのコマンド操作により、任意のデータが送信可能です。

HSB\_LIN\_COMM(SLAVE)は、ヘッダに含まれる ID が自分自身の ID と一致した場合、レスポンス送信を行います。

#### —HSB\_LIN\_COMM が送信するデータ—

		内容
1~2 バイト目	SS SS	100ms 毎にインクリメントされるシリアル番号を送信
3 バイト目	XX	DIP-SW の状態(下位 4 ビットが有効データ、上位 4bit は 0) (ON の時 1, OFF の時 0) (SW が OFF, OFF, ON, ON の時、0x03)
4 バイト目	YY	現在の LED の状態(下位 4 ビットが有効データ、上位 4bit は LED 状態保持変数の値) (点灯時 1, 消灯時 0) (LED が OFF, OFF, ON, ON の時、0x?3)
5~6 バイト目	ZZ ZZ	明るさセンサーの A/D 変換値(0~4095)
7~8 バイト目	00 00	0x00 固定

ID=0x32 のデバイスが、LIN バス上に存在しない場合は、ヘッダ送信は行われますが、レスポンスを返すデバイスが存在しない(ヘッダ送信のみ行われる)状態となります。

※CAN はタイムスタンプを見れば、(送信されるデータが同一でも)新しいパケットを受信した事が判りますが、LIN の場合は、同一データが送信されている場合、新しいデータかどうかを見分ける手段がないため、データが送られ続けている事が判る様に、データパケットにシリアル番号を埋め込んでいます

HSB\_LIN\_COMM は、SLAVE 動作時 MASTER レスポンスデータ(MASTER が送信したデータ)を、LED の状態に反映させるようにしています。

(※HSB\_LIN\_COMM が MASTER モードで動作している時は、SLAVE レスポンスデータを LED に状態に反映)

— HSB\_LIN\_COMM が受信して動作に影響を与えるデータ—

		内容
3 バイト目	XX	現在の LED の状態に反映(下位 4 ビットが有効データ、上位 4bit は無視)

例えば、MASTER 側から送信するデータの、3 バイト目を 0x0F に設定した場合、

ヘッダ			レスポンス										
	ヘッダ送信元	ID	レスポンス送信元		data(0x)								
(1)	HSB_CAN_MULTI_4	0x30	HSB_CAN_MULTI_4	00	00	0F	00	00	00	00	00	00	00

HSB\_LIN\_COMM の LED1~LED4 は全点灯となります。3 バイト目を 0x03 とした場合、LED3~LED4(下位 2 ビット)が点灯となります。

※なお、デフォルトはタイマモードで動作しますが、後述の UART のコマンドによりタイマ動作を止める事が可能です

[参考]3 バイト目が LED に反映される理由

HSB\_LIN\_COMM ボードを MASTER 設定にした場合、

レスポンス									
レスポンス送信元		data(0x)							
HSB_LIN_COMM (ID=0x31)		SS	SS	XX	YY	ZZ	ZZ	00	00

SS SS: シリアル NO

XX: DIP-SW の状態

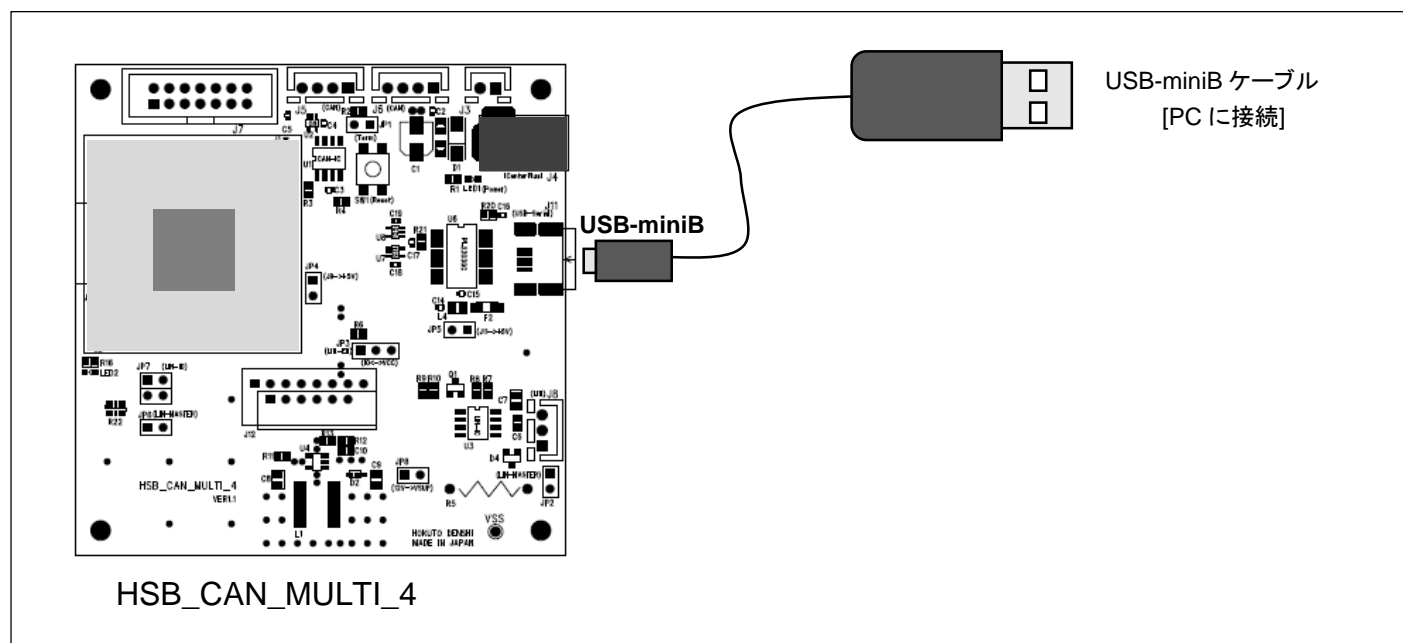
YY: LED の状態

ZZ ZZ: 明るさセンサの情報

3 バイト目に送られるデータは、DIP-SW の状態となります。HSB\_LIN\_COMM を 2 台、MASTER と SLAVE で接続した場合、ボードの LED は接続相手先の DIP-SW の状態と同じになります。

(HSB\_LIN\_COMM を 2 台接続する場合、かつ HSB\_CAN\_MULTI\_4 を接続しない場合は、別途 LIN 電源(VSUP)の印加が必要です。)

## 6. マイコンボード上で実行されるデモプログラム (UART 通信)



HSB\_CAN\_MULTI\_4 には、PC と通信する USB-Serial ポートがあります。PC と接続し PC 上でデモプログラムを実行した場合、PC と UART(SCI)で通信を行い、ボード(HSB\_CAN\_MULTI\_4)に対する指示を出したり、CAN バスのデータを UART 経由で PC に送信する事ができます。

UART(SCI)は、通信速度 115,200bps, データ 8bit、パリティなし、ストップビット 1 ビットのフォーマットです。

UART 通信は、大きく分けると、「(1)PC から HSB\_CAN\_MULTI\_4」への通信、「(2)HSB\_CAN\_MULTI\_4 から PC」への通信に 2 分されます。

(1)PC から HSB\_CAN\_MULTI\_4 に対する指令

- (1-1)HSB\_CAN\_MULTI\_4 が保持しているデータを PC に送信させる指令
- (1-2)HSB\_CAN\_MULTI\_4 を経由して CAN バスにデータを流す指令
- (1-3)HSB\_CAN\_MULTI\_4 の動作を指示する指令

(2)HSB\_CAN\_MULTI\_4 から PC に対してデータを送信

- (2-1)HSB\_CAN\_MULTI\_4 が保持しているデータを PC に送信
- (2-2)CAN バスに流れているデータを PC に転送
- (2-3)LIN バスに流れているデータを PC に転送(LIN SCI モードの場合)



(1)の PC から HSB\_CAN\_MULTI\_4 に対する指令は、3 種類の指令(コマンド)で構成されます。

### (1-1)HSB\_CAN\_MULTI\_4 が保持しているデータを PC に送信させる指令

コマンド ヘッダ	data	終端 コード	動作
&	なし	なし	データ出力コマンド

PC から、HSB\_CAN\_MULTI\_4 に対して、"&"を送ると、HSB\_CAN\_MULTI\_4 は自身が保持しているデータを PC に返送します。返送するデータのフォーマットは、(2-1)で示します。

### (1-2)HSB\_CAN\_MULTI\_4 を経由して CAN バスにデータを流す指令

コマンド ヘッダ	data	終端 コード	動作		
\$	CAN-ID (8bytes)	RTR (2bytes)	CAN-DATA (16bytes)	¥r¥n	CAN バスに指定されたデータを流す

本コマンドを用いると、PC から HSB\_CAN\_MULTI\_4 を経由して CAN バスに対して任意のデータ(データフレーム、またはリモートフレーム)を送信する事が可能。

#### —実際のパケット例—

コマンド ヘッダ	data	終端 コード	動作		
\$	00000201	00	0102030405060708	¥r¥n	CAN-ID 0x201 に対して、0x0102030405060708 のデータフレームを送信する

コマンド ヘッダ	data	終端 コード	動作		
\$	00000201	01		¥r¥n	CAN-ID 0x201 に対して、リモートフレームを送信する

- ・PC から送信するデータはバイナリではなく ASCII(1byte を 2 文字で表す)
- ・コマンドの終端は¥r¥n(0x0d 0x0a)(¥r のみ, ¥n のみでも可)
- ・有効なコマンドは全部で 80 文字まで

### (1-3)HSB\_CAN\_MULTI\_4 の動作を指示する指令

コマンド ヘッダ	data	終端 コード	動作	
%	COM (2bytes)	data	¥r¥n	

HSB\_CAN\_MULTI\_4 のボードの動作を PC から指示、COM はコマンド種。

—実際のパケット例—

コマンド ヘッダ	data		終端 コード	動作
%	01	0102030405060708	¥r¥n	LIN のレスポンスデータ設定

LIN の MASTER/SLAVE モードで送信するデータを設定するコマンドです。

コマンド ヘッダ	data		終端 コード	動作
%	02	00	¥r¥n	LIN をタイマ動作モードに設定

コマンド ヘッダ	data		終端 コード	動作
%	02	01	¥r¥n	LIN を SCI 動作モードに設定

LIN をタイマ動作モード(起動時のデフォルト)に設定した場合(かつ、HSB\_CAN\_MULTI\_4 を LIN MASTER モードに設定している場合)、100ms 毎に LIN ID=0x30~0x34 でヘッダ送信を行います。

LIN 動作モードを、SCI 動作モードにした場合、ヘッダ送信は、UART(SCI)経由でコマンドが来た(下記の COM=03, COM=04 のコマンド)タイミングで行われます。

コマンド ヘッダ	data		終端 コード	動作
%	03		¥r¥n	MASTER ヘッダ・レスポンス送信

HSB\_CAN\_MULTI\_4 を LIN MASTER モードに設定している場合、MASTER ヘッダ/レスポンス送信(ID=0x30 で、ヘッダと前出の(COM=01 の)レスポンスデータ設定コマンドで設定したデータを送信)します。

コマンド ヘッダ	data		終端 コード	動作
%	04	31 (LIN-ID)	¥r¥n	SLAVE ヘッダ送信

HSB\_CAN\_MULTI\_4 を LIN MASTER モードに設定している場合、SLAVE ヘッダ送信(ID=0x31 でヘッダ送信)を行います。(ID=0x31 の SLAVE デバイスに対してレスポンスデータを要求する動作)

ID は、コマンドパケットに含まれるので、任意の LIN-ID のデバイスに対して、ヘッダ送信(レスポンスデータの要求)が行えます。

- ・PC から送信するデータはバイナリではなく ASCII(1byte を 2 文字で表す)
- ・コマンドの終端は¥r¥n(0x0d 0x0a)(¥r のみ, ¥n のみでも可)
- ・有効なコマンドは全部で 80 文字まで

は、(1-3)でも共通です。

## (2-1)HSB\_CAN\_MULTI\_4 が保持しているデータを PC に送信

PC から HSB\_CAN\_MULTI\_4 に"&"コマンドを送ると、HSB\_CAN\_MULTI\_4 は以下のデータを返送します。

データ ヘッダ	data		終端 コード	内容
:	START		¥r¥n	一連のデータの開始

一連のデータの開始を示す":START"が最初に送信されます。その後、データが続きます。

データ ヘッダ	data		終端 コード	内容
#	TYPE	data	¥r¥n	

#の後、データ種(TYPE)を2文字で示し、その後データが続く。

### —実際のパケット例—

データ ヘッダ	data		終端 コード	内容
#	01	1111	¥r¥n	ボード接続状態

HSB\_CAN\_MULTI\_1~HSB\_CAN\_MULTI\_4 の4台のボードが接続されている時は、1111をデータとして返す。HSB\_CAN\_MULTI\_1, HSB\_CAN\_MULTI\_4 の2台のボードが接続されている時は、1001をデータとして返す。(0:ボード未接続, 1:ボード接続)

データ ヘッダ	data		終端 コード	内容
#	02	32	¥r¥n	HSB_CAN_MULTI_1 のモータ duty 値

HSB\_CAN\_MULTI\_1 の現在のモータ duty 値(HSB\_CAN\_MULTI\_1 が CAN ID=0x101 で発信している情報)。duty=50%の時、50=0x32→ASCII で"32"を返す。

データは、ASCII コード(0~F)で、1byte が2文字で送られてきます。

データ ヘッダ	data		終端 コード	内容
#	03	1388	¥r¥n	HSB_CAN_MULTI_1 のモータ duty 値

HSB\_CAN\_MULTI\_1 の現在のモータの回転数(HSB\_CAN\_MULTI\_1 が CAN ID=0x101 で発信している情報)。回転数 5000[rpm]のとき、5000=0x1388→ASCII で"1388"を返す。

データ ヘッダ	data		終端 コード	内容
#	04	0001	¥r¥n	HSB_CAN_MULTI_2 のマトリックスキー情報

HSB\_CAN\_MULTI\_2 の現在押されているマトリックスキーの情報(HSB\_CAN\_MULTI\_2 が CAN ID=0x102 で発信している情報)。0 のキーを押している際、0x0001→ASCII で"0001"を返す。

データ ヘッダ	data		終端 コード	内容
#	05	0123	¥r¥n	HSB_CAN_MULTI_3 の明るさセンサの情報

HSB\_CAN\_MULTI\_3 の明るさセンサの情報(HSB\_CAN\_MULTI\_3 が CAN ID=0x103 で発信している情報)。12 ビットの A/D 変換値(0~0xFFF)、0x0123 の時→ASCII で"0123"を返す。

データ ヘッダ	data		終端 コード	内容
#	06	092E	¥r¥n	HSB_CAN_MULTI_3 の温度センサの情報

HSB\_CAN\_MULTI\_3 の温度センサの情報(HSB\_CAN\_MULTI\_3 が CAN ID=0x103 で発信している情報)。摂氏温度を 100 倍した値、23.5°C の時 2350→0x092E の時→ASCII で"092E"を返す。

データ TYPE=02~06 は、HSB\_CAN\_MULTI\_1~ HSB\_CAN\_MULTI\_3 が、CAN-ID=0x101~0x103 で出力しているデータで、データ内容は 4.1 「CAN-ID(1)で出力するデータ」と同じものが UART で送られてきます。

データ ヘッダ	data		終端 コード	内容	
#	07	31 (LIN-ID)	0102030405060708 (data)	¥r¥n	HSB_CAN_MULTI_4 が受信している LIN のデータ(ID 1byte, データ 8bytes)

TYPE=07 は HSB\_CAN\_MULTI\_4 が受信している LIN データを返します。

31 : LIN-ID(1 バイト)

0102030405060708 : LIN データ(8 バイト)

トータル 9 バイトのデータが ASCII で 18 文字のデータとして送られてきます。

LIN データは、HSB\_CAN\_MULTI\_4 がデータを受信した際に、リングバッファに保存します。HSB\_CAN\_MULTI\_4 が、UART で&コマンドを受け取った際、リングバッファに保存されているデータを送信します。複数のデータが保存されている場合は、保存されているすべてのデータを複数行に分けて送信します。(※リングバッファは、デフォルトでは 15 個までのデータを保存可能。読み出しされなかったデータは、古いものから上書きされます。)

データ ヘッダ	data		終端 コード	内容
:	END		¥r¥n	一連のデータの終了

一連のデータの終了を示す":END"が最後に送信されます。

## (2-2)CAN バスに流れているデータを PC に転送

CAN バスに流れているデータで、ID=0x201~0x203 のデータ(\*1)を UART に転送します。

データ ヘッダ	data						終端 コード	内容
*	CAN-ID	RTR	IDE	DLC	data	TS	¥r¥n	CAN のデータパケット

—実際のパケット例—

データ ヘッダ	data						終端 コード	内容
*	00000201	01	01	08	0102030405060708	A123	¥r¥n	CAN のデータパケット

CAN の ID, RTR(データフレーム/リモートフレームを示す), IDE(標準フォーマット/拡張フォーマットを示す), DLC(データサイズを示す), データ(プログラム定義で 8 バイト), TS(タイムスタンプ、2 バイト)を、ASCII で送信します。

(\*1)UART 送信を行うボードを HSB\_CAN\_MULTI\_4 に設定(デフォルト)した場合

## (2-3)LIN バスに流れているデータを PC に転送(LIN SCI モードの場合)

LIN バスに流れているデータを UART に転送します。

データ ヘッダ	data			終端 コード	内容
+	LIN-ID	data	sum	¥r¥n	LIN のデータパケット

—実際のパケット例—

データ ヘッダ	data			終端 コード	内容
+	31	01020304050607	2A	¥r¥n	LIN のデータパケット

LIN の ID, データ(プログラム定義で 8 バイト), チェックサムを ASCII で送信します。

(※HSB\_CAN\_MULTI\_4 の LIN 動作モードが、SCI 通信モードの場合)

(LIN 動作モードがタイマモードの場合、"&"コマンドで LIN データの読み出しを行います。SCI 通信モード場合、データ受信を行ったタイミングで転送します。LIN の動作モードは、HSB\_CAN\_MULTI\_DEMO2.exe で行います。)

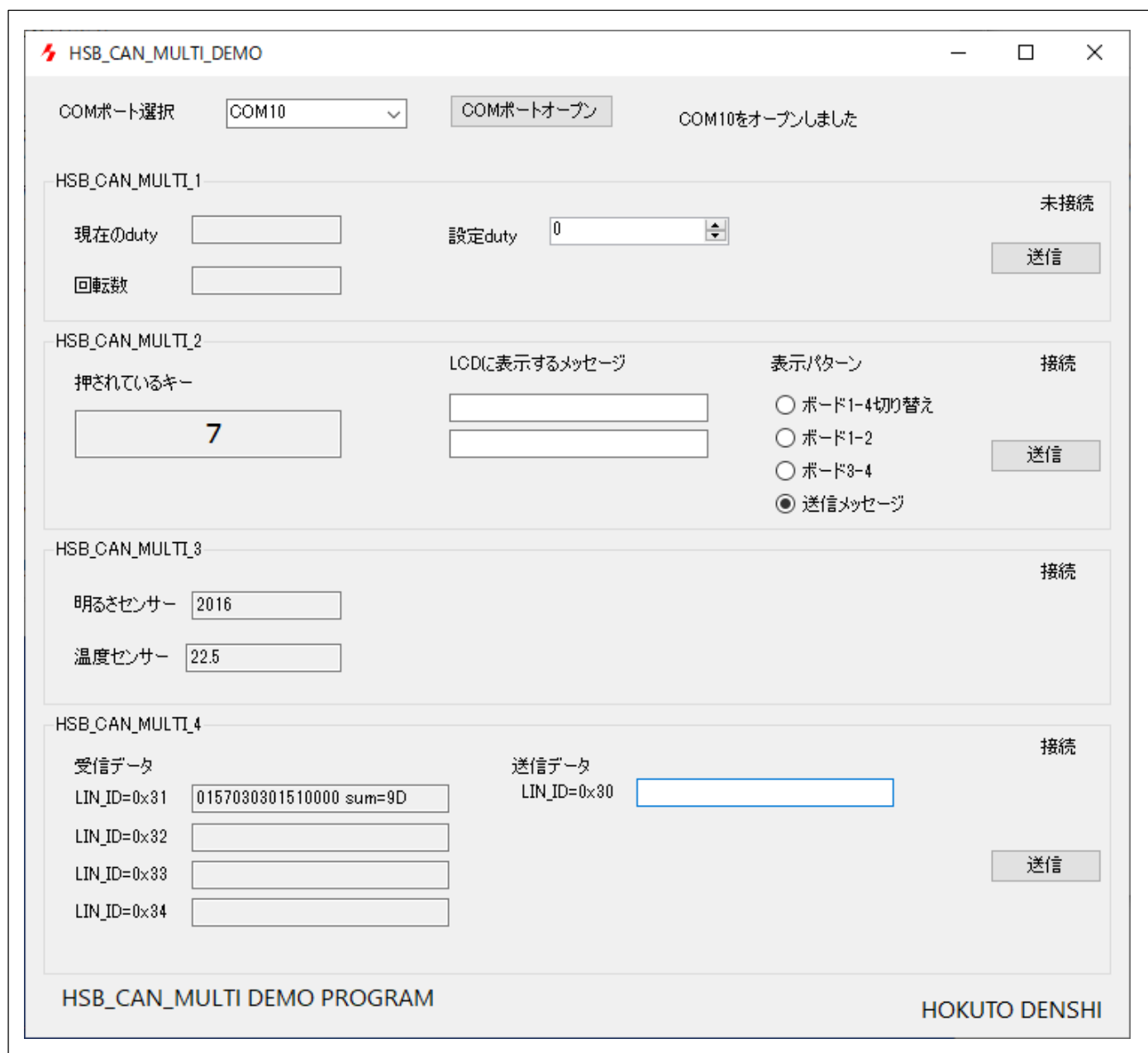
UART 通信に関しては、基本的には HSB\_CAN\_MULTI\_4 のボードが行います。但し、ネットワーク構成に HSB\_CAN\_MULTI\_4 が含まれない場合は、他のボードに UART 通信機能を担わせる事が可能です。

(いくつか制約条件が付くのと、プログラムの再ビルド、ボードへの書き込みが必要です。)

## 7. PC 向けデモプログラム

PC 向けのデモプログラムは 2 種類用意してあります。

### 7.1. HSB\_CAN\_MULTI\_DEMO.exe



HSB\_CAN\_MULTI\_DEMO

COMポート選択: COM10 (COMポートオープン) COM10をオープンしました

HSB\_CAN\_MULTI\_1: 未接続  
 現在のduty: [ ] 設定duty: 0 [ ] 送信

HSB\_CAN\_MULTI\_2: 接続  
 押されているキー: [ 7 ] LCDに表示するメッセージ: [ ] [ ] 表示パターン:  
 ボード1-4切り替え  
 ボード1-2  
 ボード3-4  
 送信メッセージ 送信

HSB\_CAN\_MULTI\_3: 接続  
 明るさセンサー: [ 2016 ]  
 温度センサー: [ 22.5 ]

HSB\_CAN\_MULTI\_4: 接続  
 受信データ: LIN\_ID=0x31 [ 0157030301510000 sum=9D ]  
 LIN\_ID=0x32 [ ]  
 LIN\_ID=0x33 [ ]  
 LIN\_ID=0x34 [ ] 送信データ: LIN\_ID=0x30 [ ] 送信

HSB\_CAN\_MULTI DEMO PROGRAM HOKUTO DENSHI

HSB\_CAN\_MULTI\_DEMO.exe は、

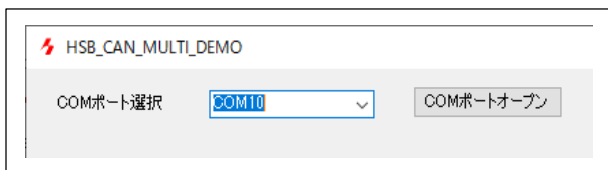
- ・ボードの接続状況の表示
- ・CAN バスに流れているデータを表示 (HSB\_CAN\_MULTI\_1~HSB\_CAN\_MULTI\_3 が出力しているデータ)(\*1)
- ・LIN バス上に流れているデータを表示 (ID=0x31~ID=0x34 のデータ)
- ・HSB\_CAN\_MULTI\_1 のモータ回転デューティ比の設定
- ・HSB\_CAN\_MULTI\_2 の LCD 表示設定
- ・LIN バス上に送信するデータを設定 (ID=0x30)

が行えます。

(\*1)は、ID=0x10X でボードが 500ms 毎に送信したデータを、HSB\_CAN\_MULTI\_4 が受信。PC から"&"コマンドを送り、HSB\_CAN\_MULTI\_4 に UART で情報を出力させてモニタリングを行っています。

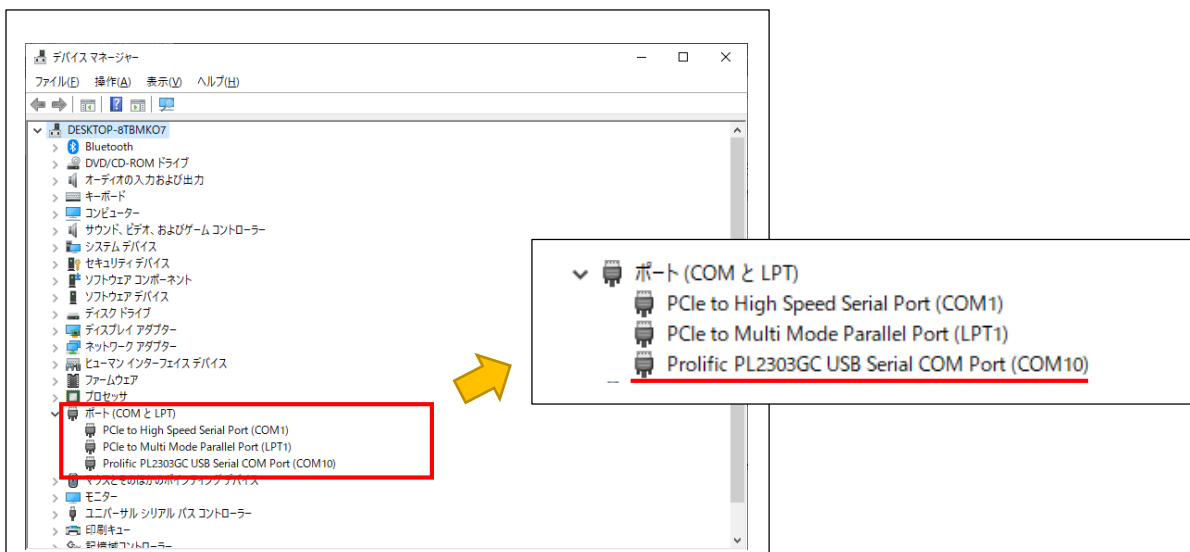
ボード同士が自律的に交換しているデータを拾ってきて表示させる。また、ボードに対して指示を出せる内容になっています。(情報をモニタリングするためのツールです。)

起動すると、



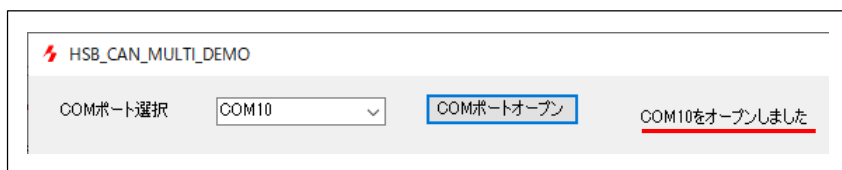
COM ポート選択のプルダウンメニューが左上にありますので、HSB\_CAN\_MULTI\_4 の USB-Serial ポート番号を選択して、「COM ポートオープン」のボタンを押してください。

HSB\_CAN\_MULTI\_4 の USB-Serial ポート番号が不明な場合は、デバイスマネージャーを起動して、



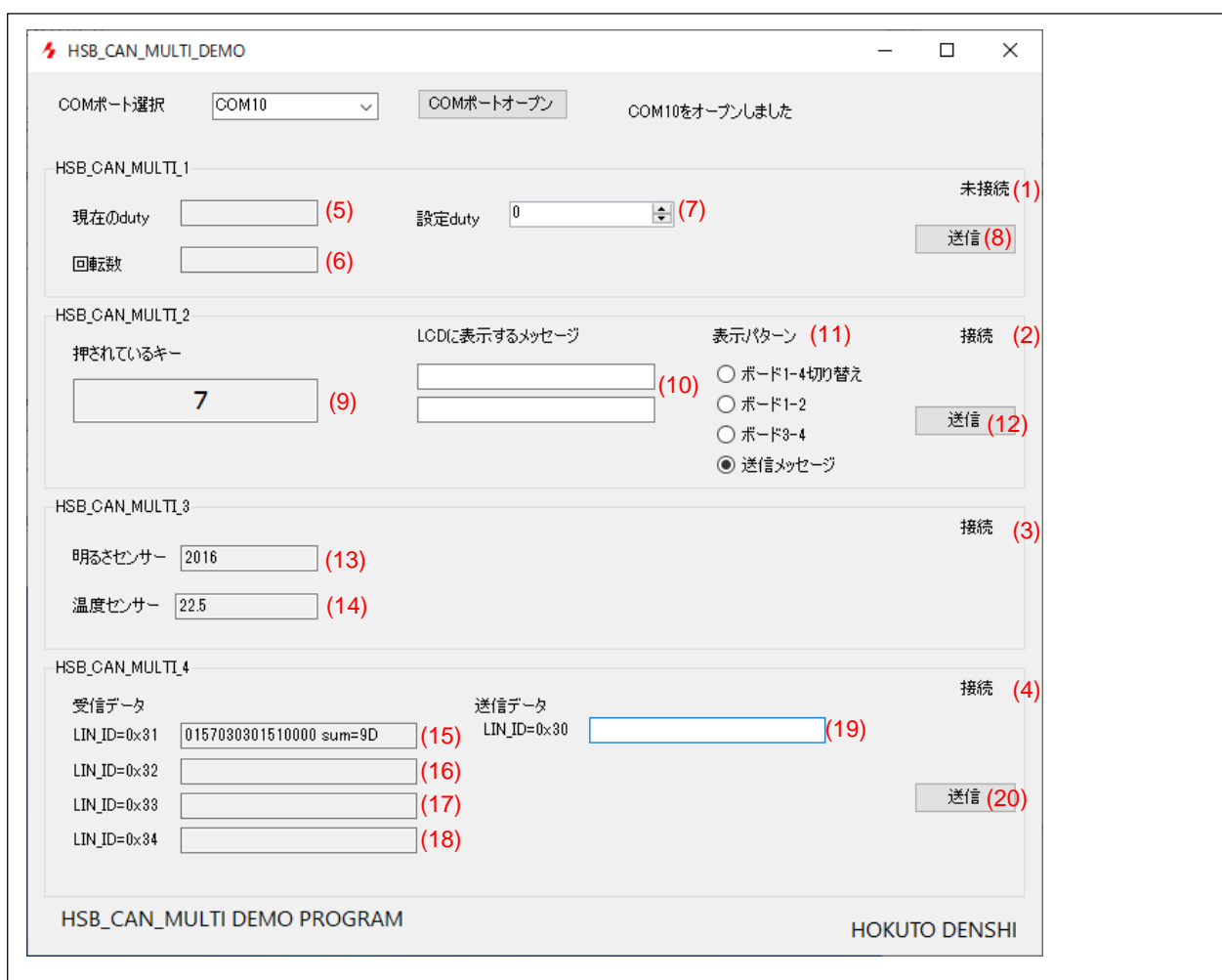
ポート(COMとLPT)の欄の、Prolific PL2303GC USB Serial COM Port の COM?の番号を調べてください。

COMポートオープンのボタンを押して、COMポートが開けた場合、



COM?をオープンしました、という表示が出ます。

—本アプリケーションソフトの表示、ボタンの役割—



(1)~(4)ネットワーク上にボードが存在しているかどうかの表示

(5)HSB\_CAN\_MULTI\_1 の現在のモータ duty 設定値

(6)HSB\_CAN\_MULTI\_1 の現在のモータ回転数

(7)HSB\_CAN\_MULTI\_1 の duty 値を変える場合、数値を入力するボックス(-100~100 の値が入力可能)

(8)(7)の設定値をボードに送信するボタン

(9)HSB\_CAN\_MULTI\_2 のマトリックスキーの状態

(10)HSB\_CAN\_MULTI\_2 の LCD に表示するメッセージ(16×2 行)を入力するボックス

(11)HSB\_CAN\_MULTI\_2 の表示パターンを設定するラジオボタン

(12)HSB\_CAN\_MULTI\_2 にデータを送信するボタン



- (13) HSB\_CAN\_MULTI\_3 の明るさセンサの値
- (14) HSB\_CAN\_MULTI\_3 の温度センサの値
- (15) LIN ID=0x31 で受信したデータ
- (16) LIN ID=0x32 で受信したデータ
- (17) LIN ID=0x33 で受信したデータ
- (18) LIN ID=0x34 で受信したデータ
- (19) HSB\_CAN\_MULTI\_4(LIN MASTER, ID=0x30)が送信するデータを設定するボックス  
0~9,A~F(a~f)の文字を 16 文字まで(8 バイトデータ)の入力が可能
- (20) (19)で入力したデータを送信するボタン

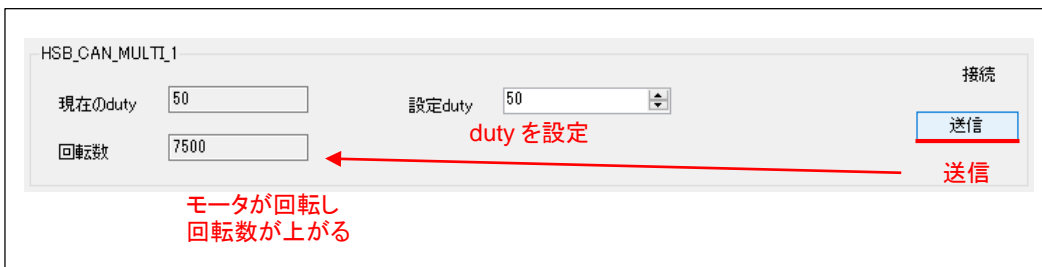
本アプリケーションは、モニタリングツールの位置付けなので、HSB\_CAN\_MULTI\_2 のキーマトリックスのキーを押したり、HSB\_CAN\_MULTI\_3 の明るさセンサに手をかざしたりした場合、表示されている値がリアルタイムに(多少タイムラグがありますが)変わります。

(15)~(18)の LIN データは、HSB\_LIN\_COMM がつながっていれば、表示データが更新されていくはずですが。

送信ボタンを押した際は、HSB\_CAN\_MULTI\_4 経由で各ボードに対してデータが送られますので、HSB\_CAN\_MULTI\_1 のモータ回転数(duty)や、HSB\_CAN\_MULTI\_2 の LCD 表示メッセージ等を変更する事ができます。

#### — 操作例 —

#### ・HSB\_CAN\_MULTI\_1 に対する指示



duty 値を設定して、送信ボタンを押すと、

- ・HSB\_CAN\_MULTI\_1 のモータが回転
- ・回転数が取得され、表示に反映

される動作となります。

・HSB\_CAN\_MULTI\_2 に対する指示

HSB\_CAN\_MULTI\_2
接続

押されているキー

LCDに表示するメッセージ

LCD MESSAGE

123456789ABCDE

表示文字列を入力

表示パターン

ボード1-4切り替え

ボード1-2

ボード3-4

送信メッセージ

送信メッセージを選択

送信

---

送信

LCD に表示されるメッセージ



・HSB\_CAN\_MULTI\_4 に対する指示

HSB\_CAN\_MULTI\_4
接続

受信データ

LIN\_ID=0x31 8E7F030500DB0000 sum=5C

LIN\_ID=0x32

LIN\_ID=0x33

LIN\_ID=0x34

受信データの3バイト目に反映

送信データ

LIN\_ID=0x30 0000050000000000

3バイト目に 05 を設定

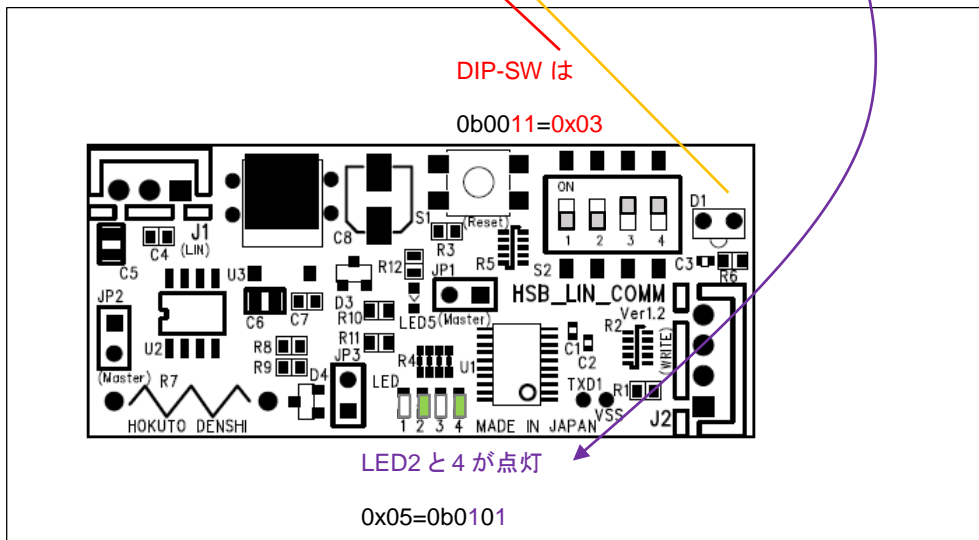
明るさセンサの値

送信

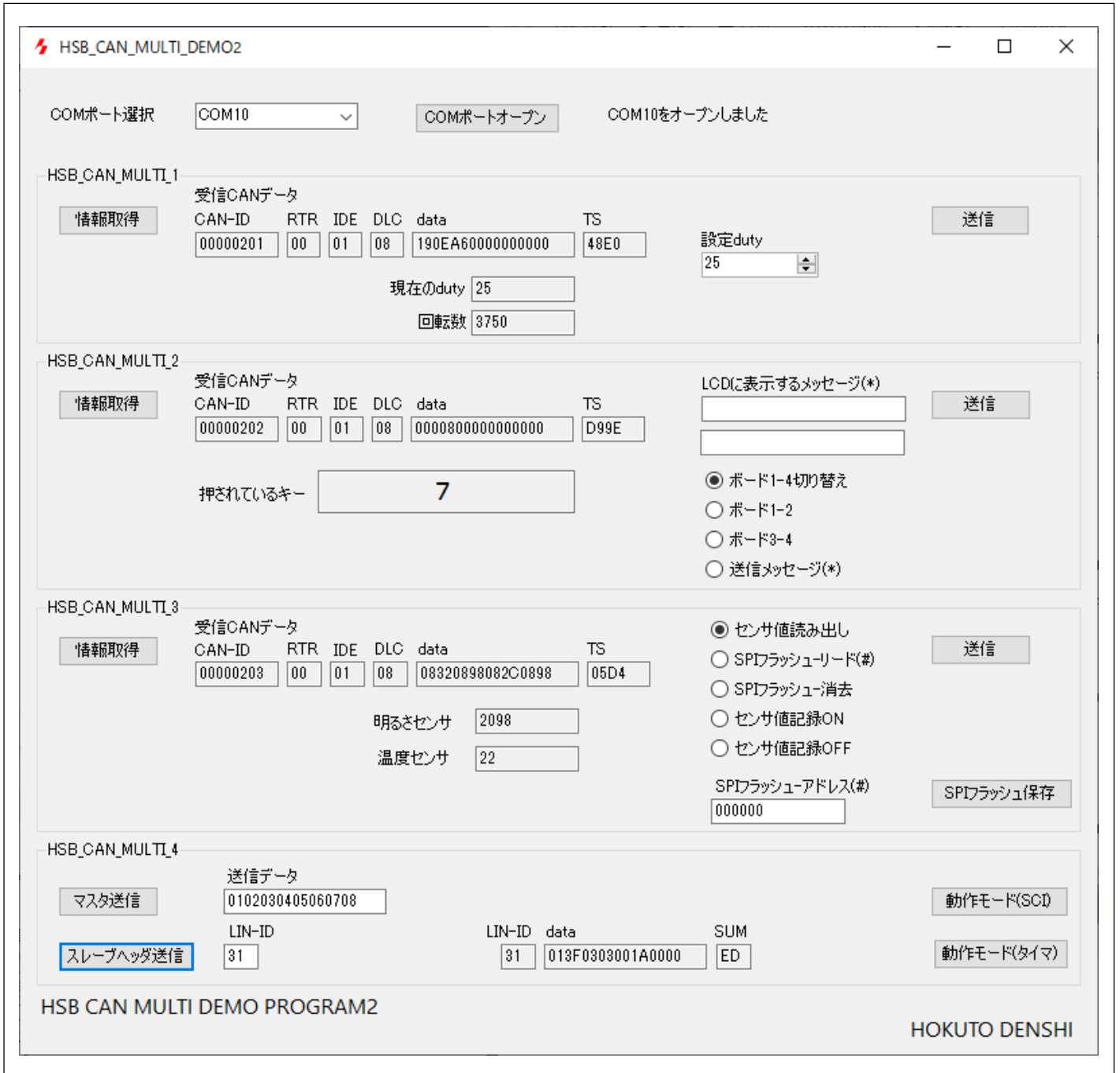
---

送信

接続先の HSB\_LIN\_COMM ボード



## 7.2. HSB\_CAN\_MULTI\_DEMO2.exe



HSB\_CAN\_MULTI\_DEMO2.exe は、

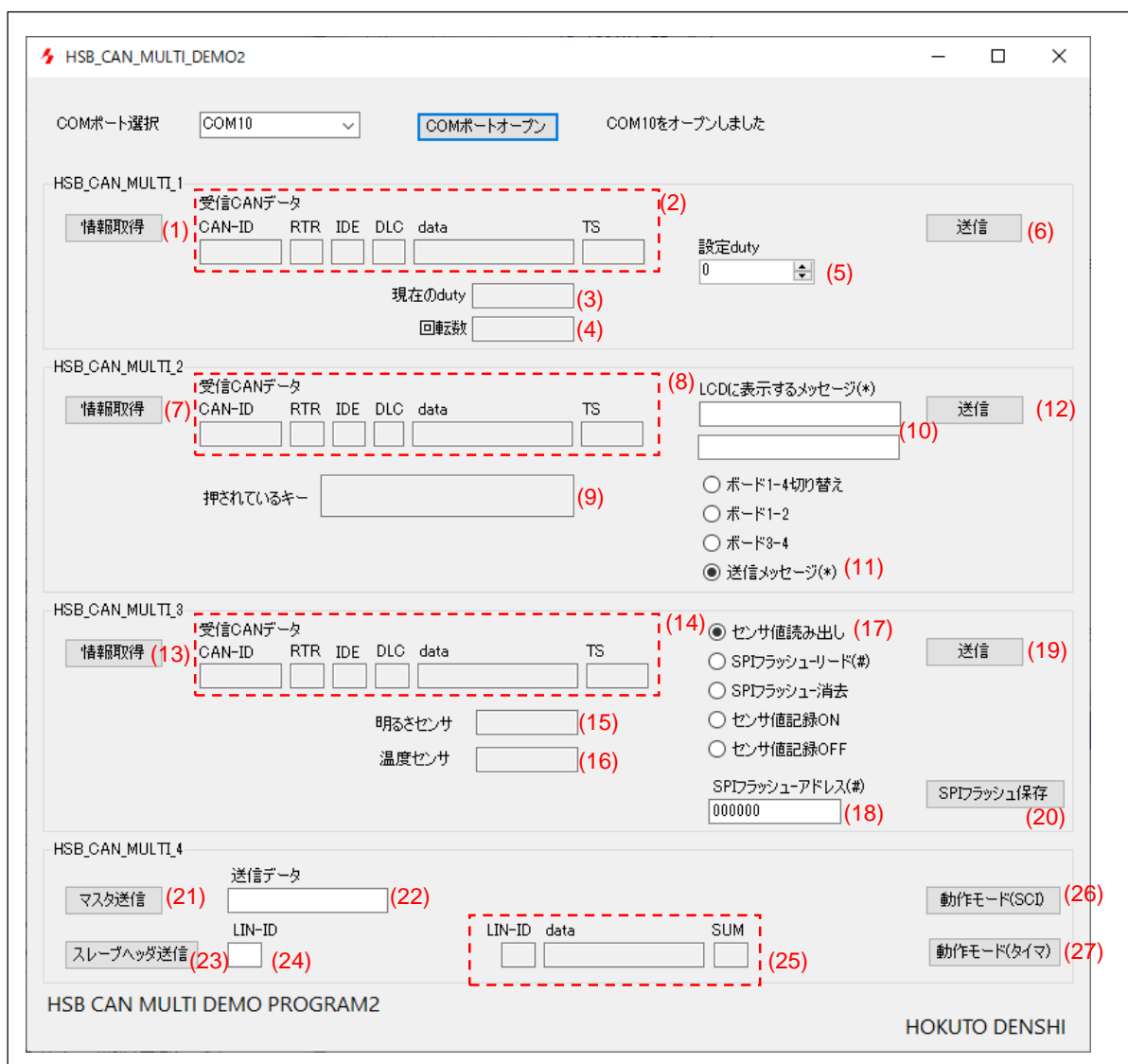
- ・各ボードに対する情報取得の指示 (CAN のリモートフレームを使用したデータ返送指示)
- ・HSB\_CAN\_MULTI\_1 のモータ回転デューティ比の設定
- ・HSB\_CAN\_MULTI\_2 の LCD 表示設定
- ・HSB\_CAN\_MULTI\_3 の SPI フラッシュに対する操作

- ・LIN バス上に MASTER レスポンス送信(ID=0x30)
- ・LIN バスにヘッダ送信(レスポンスデータ要求)

が行えます。

HSB\_CAN\_MULTI\_DEMO.exe が受動的な動作を行うアプリなのに対して、こちらは能動的な動作をするアプリです。ボードに対してデータを送り、その応答を表示に反映させる動作となります。

起動後、COMポートをオープンするところは、HSB\_CAN\_MULTI\_DEMO.exe と同じです。



COMポートを開いても、何も表示更新は行われません。

(1)の情報更新を押すと、HSB\_CAN\_MULTI\_4 から CAN-ID=0x0201 のリモートフレームが送信されます。ネットワーク上に、HSB\_CAN\_MULTI\_1(ID=0x201)が存在していれば、HSB\_CAN\_MULTI\_1 が ID=0x201 でデータフレーム応答を返します。そのデータは、HSB\_CAN\_MULTI\_1 の(2)の欄に表示されます。

また、(2)の data 部分のデータを変換して(3)の duty 値と(4)の回転数が表示されます。

CAN のデータは、以下の項目が表示されます。

CAN-ID 4 バイトデータを ASCII 8 文字で表示

RTR 00:データフレーム, 01:リモートフレーム

IDE 00:標準フォーマット, 01:拡張フォーマット

DLC 08:データ 8 バイト

data CAN のデータパケット(8 バイトデータを ASCII 16 文字で表示)

TS タイムスタンプ 受信時に受信側のマイコンで付与されるタイム値(2 バイトを ASCII 4 文字で表示)

RTR は、HSB\_CAN\_MULTI\_4 からリモートフレームを送信し、それに対する HSB\_CAN\_MULTI\_1 からの応答を HSB\_CAN\_MULTI\_4 が受信して、この欄に反映させますので、必ず 00(データフレーム)となります。

IDE は、マイコンボード側のプログラムで拡張フォーマットを指定しているため、01(拡張フォーマット)となります。

DLC は、マイコンボード側のプログラムで、8 バイトとしているため、08(8 バイト)となります。

data は、その都度変わります。

TS は、「情報更新」ボタンを押すたびに別な値となります。(CAN データを受信したタイミングに起因する値)

(7)の情報取得を押すと、HSB\_CAN\_MULTI\_4 から CAN-ID=0x0202 のリモートフレームが送信されます。ネットワーク上に、HSB\_CAN\_MULTI\_2(ID=0x202)が存在していれば、HSB\_CAN\_MULTI\_2 が ID=0x202 でデータフレーム応答を返します。そのデータは、HSB\_CAN\_MULTI\_2 の(8)の欄に表示されます。

また、(8)の data 部分のデータを変換して(9)の押されているキーの値と、現在のボード側の LCD 表示設定が(11)のラジオボタンの選択に反映されます。

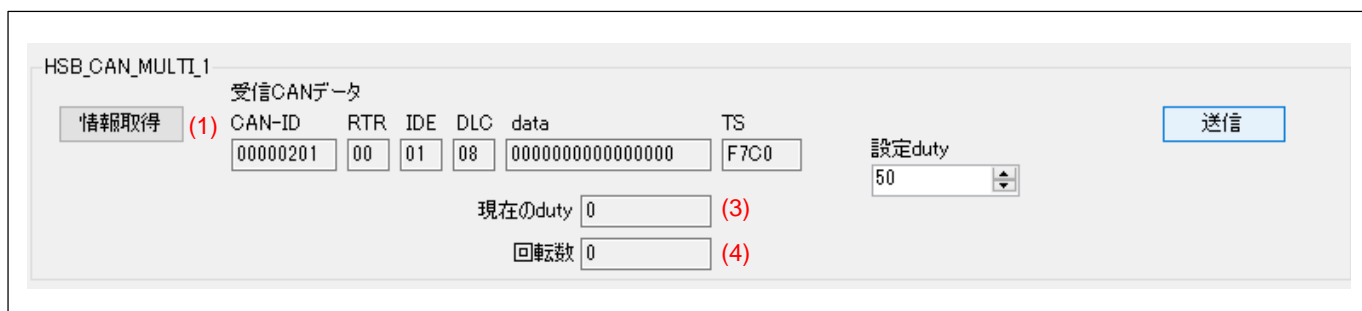
(13)の情報取得を押すと、HSB\_CAN\_MULTI\_4 から CAN-ID=0x0203 のリモートフレームが送信されます。ネットワーク上に、HSB\_CAN\_MULTI\_3(ID=0x203)が存在していれば、HSB\_CAN\_MULTI\_3 が ID=0x203 でデータフレーム応答を返します。そのデータは、HSB\_CAN\_MULTI\_3 の(14)の欄に表示されます。

また、(14)の data 部分のデータを変換して明るさセンサの値が(15)に、温度センサの値が(16)に反映されます。

次にボードに対する指示ですが、  
HSB\_CAN\_MULTI\_1 の操作

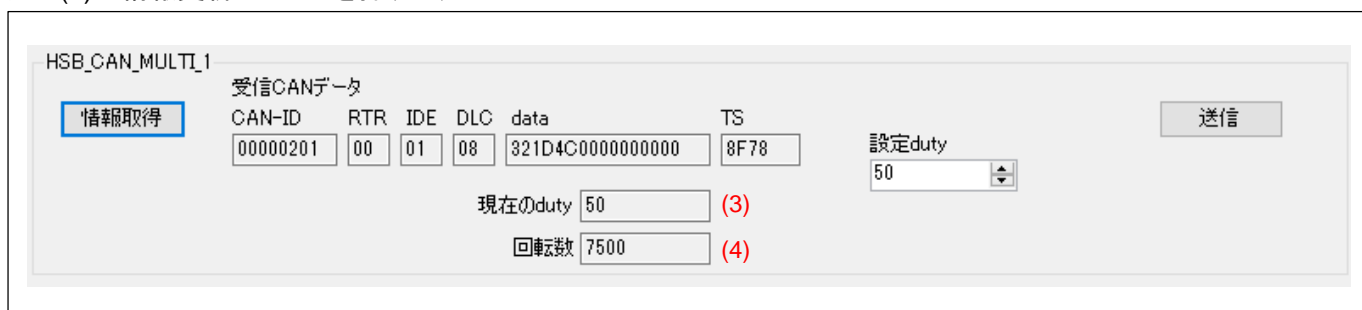


(5)に値(-100~100)を設定して、(6)の送信ボタンを押します。



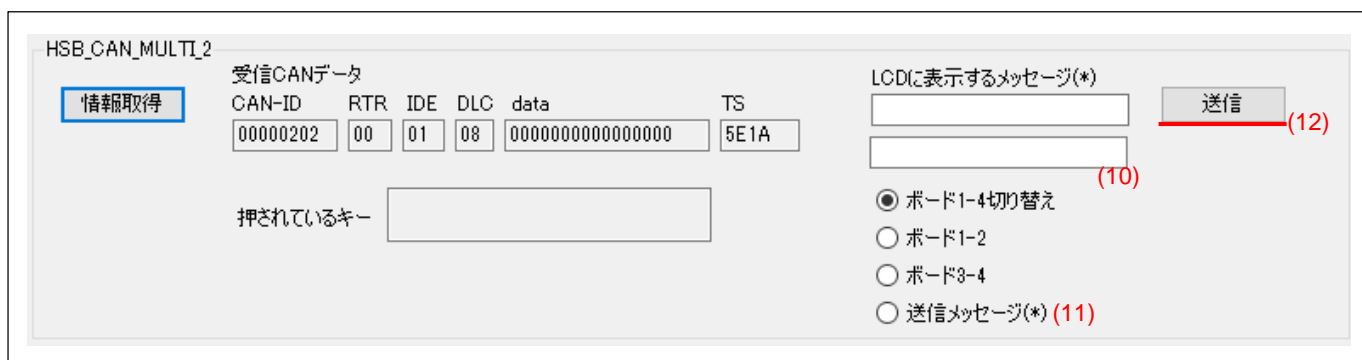
この時点では、CAN データ、(3)(4)の値が更新される事はありません。

次に(1)の情報更新のボタンを押すと、



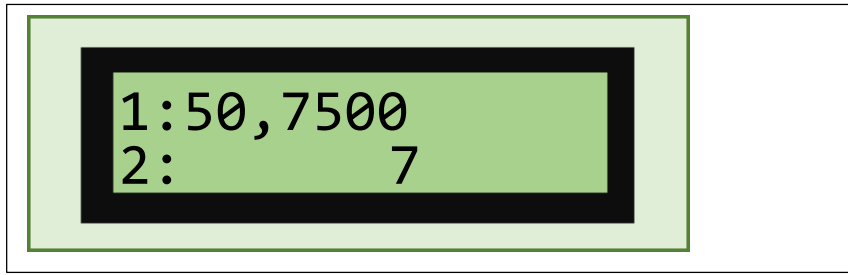
CAN のデータフレームを受信して、(3)(4)のデータが更新されます。

HSB\_CAN\_MULTI\_2 の操作



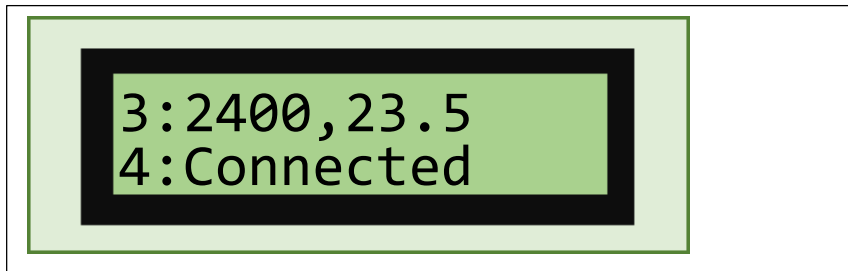
(11)の「ボード 1-2」を選択して、(12)の送信を押すと、

LCDに表示されるメッセージ



LCDには、HSB\_CAN\_MULTI\_1とHSB\_CAN\_MULTI\_2の状態が表示される様になります。

「ボード3-4」を選択して、送信ボタンを押すと、

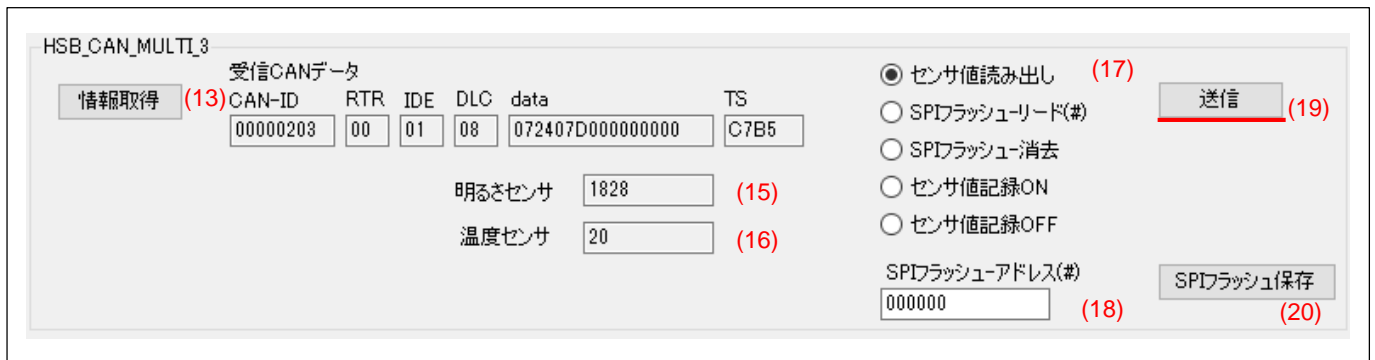


LCDには、HSB\_CAN\_MULTI\_3とHSB\_CAN\_MULTI\_4の状態が表示される様になります。

起動時のデフォルトは、「ボード1-4切り替え」で、上記の2種類の表示が5秒毎に切り替わる動作です。

(10)のボックスにメッセージ(文字列)を入力して、「送信メッセージ」を選択して、送信ボタンを押すとLCDに任意のメッセージを表示させる事ができます。

### HSB\_CAN\_MULTI\_3の操作



(17)のラジオボタンで、ボードに対してコマンドを送ることができます。

「センサ値読み出し」「送信」「情報取得」を行うと、「情報取得」で受信するデータは明るさセンサと温度センサの値となります。(起動時のデフォルト)

HSB\_CAN\_MULTI\_3

受信CANデータ

情報取得 (13)

CAN-ID	RTR	IDE	DLC	data	TS
00000203	00	01	08	072407D000000000	C7B5

明るさセンサ 1828 (15)

温度センサ 20 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(18) 000000

SPIフラッシュ保存 (20)

送信 (19)

(17)のラジオボタンで、「センサ値記録 ON」「送信」を行うと、明るさセンサと温度センサの値を SPI フラッシュメモリに記録する動作となります。

データは、1 秒間に 1 回記録されますので、多少時間を置いた後で、今度は「センサ値記録 OFF」「送信」を押します。

HSB\_CAN\_MULTI\_3

受信CANデータ

情報取得 (13)

CAN-ID	RTR	IDE	DLC	data	TS
00000203	00	01	08	072407D000000000	C7B5

明るさセンサ 1828 (15)

温度センサ 20 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(18) 000000

SPIフラッシュ保存 (20)

送信 (19)

次に、

HSB\_CAN\_MULTI\_3

受信CANデータ

情報取得 (13)

CAN-ID	RTR	IDE	DLC	data	TS
00000203	00	01	08	094E083409380834	7BEA

明るさセンサ 2382 (15)

温度センサ 21 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(18) 000000

SPIフラッシュ保存 (20)

送信 (19)

「SPI フラッシュリード」を選択して「送信」を押します。この状態で、SPI フラッシュの読み出しモードになりましたので、「情報取得」を押します。

HSB\_CAN\_MULTI\_3 が送信するデータは、SPI フラッシュのアドレス 0x0~0x7 のデータです。(センサデータを送ってきた場合は、data の後半 4 バイトは 0 ですが、この時に送ってきたデータは SPI フラッシュ内のデータなので、後半もデータが入っています。)



その後、情報取得のボタンを押していくと、

HSB\_CAN\_MULTI\_3 受信CANデータ

情報取得 (13)	CAN-ID	RTR	IDE	DLC	data	TS
	00000203	00	01	08	0945083409280834	3A0F

明るさセンサ 2373 (15)  
温度センサ 21 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(#) (19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(#) 000000 (18) SPIフラッシュ保存 (20)

data の部分が変化していくはずですが、現在、SPI フラッシュの読み出しモードになっているので、情報取得のボタンを押す度に、

アドレス 0x8~0xf

アドレス 0x10~0x17

アドレス 0x18~0x1f

...

と、アドレスが 8 バイトインクリメントされたデータを順次取得可能です。

HSB\_CAN\_MULTI\_3 受信CANデータ

情報取得 (13)	CAN-ID	RTR	IDE	DLC	data	TS
	00000203	00	01	08	090E0834090D0834	38AB

明るさセンサ 2318 (15)  
温度センサ 21 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(#) (19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(#) 100 (18) SPIフラッシュ保存 (20)

(18)の SP フラッシュアドレス 100 (0x100 を表す)

(17)は、「SPI フラッシュリード」を選択して、「送信」ボタン。その後、「情報取得」を押すと、

アドレス 0x100~0x107

のデータを取得可能です。

同様に、

HSB\_CAN\_MULTI\_3 受信CANデータ

情報取得 (13)	CAN-ID	RTR	IDE	DLC	data	TS
	00000203	00	01	08	FFFFFFFFFFFFFFFF	FE83

明るさセンサ 65535 (15)  
温度センサ -0.01 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(#) (19)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

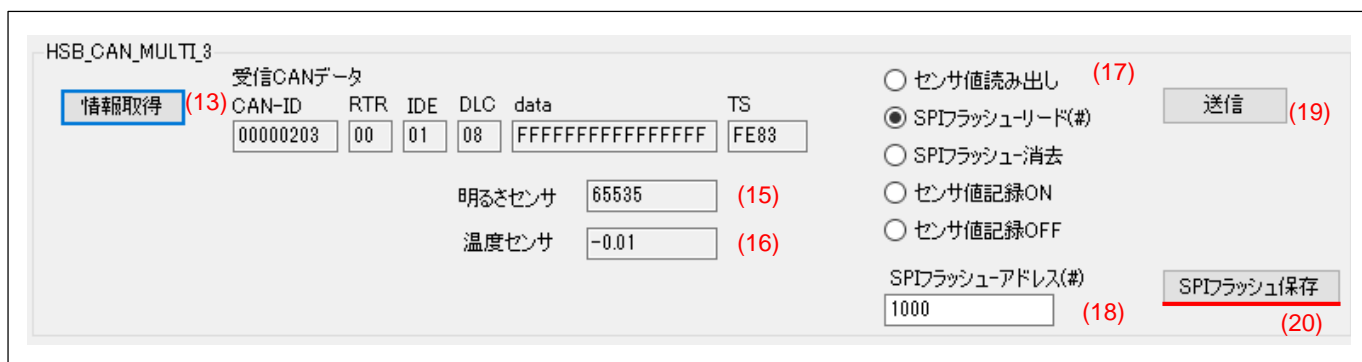
SPIフラッシュアドレス(#) 1000 (18) SPIフラッシュ保存 (20)

SPI フラッシュアドレス 1000(0x1000)を入力して「送信」「情報取得」の順に操作すると、アドレス 0x1000~0x1007 のデータを取得可能です。取得されたデータは、0xFF x8 となっているので、これは SPI フラッシュ消去後の値です。

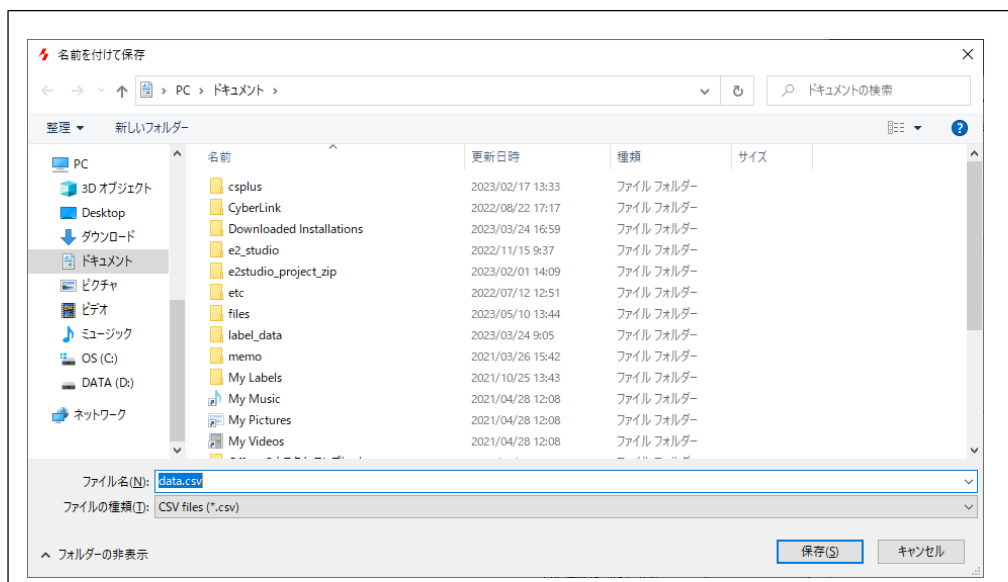
センサ値記録モードにすると、1 秒に 1 データ(8 バイト)ずつ記録されます。(512 秒間記録すれば、4096 バイト=0x1000 まで記録されるはずです。)

このケースでは、0x100 以降 0x1000 未満のどこかのアドレスまで、データが記録されています。

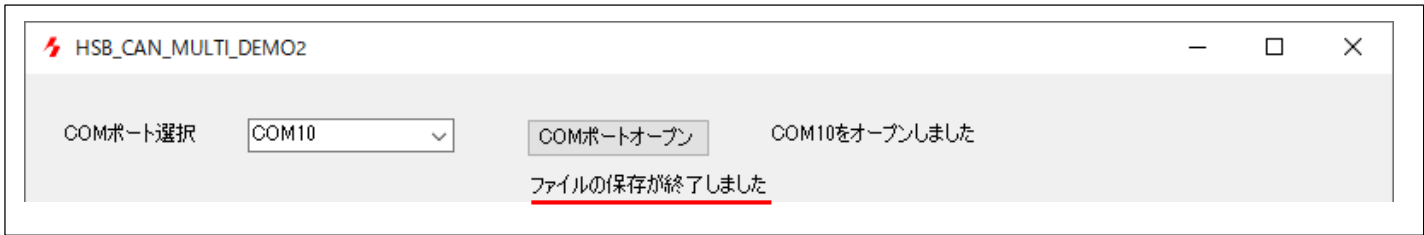
SPI フラッシュに記録されている全データを取得する場合は、



(20)の「SPI フラッシュ保存」ボタンを押します。



ファイル保存のダイアログが出ますので、ファイル名や保存先を選択して、「保存」してください。



ファイルの保存が終了すると、アプリケーションの上の方に、メッセージが出ます。

※処理時間に関して

PC→HSB\_CAN\_MULTI\_4 に UART でデータ取得指示

HSB\_CAN\_MULTI\_4 から HSB\_CAN\_MULTI\_3 にリモートフレーム送信

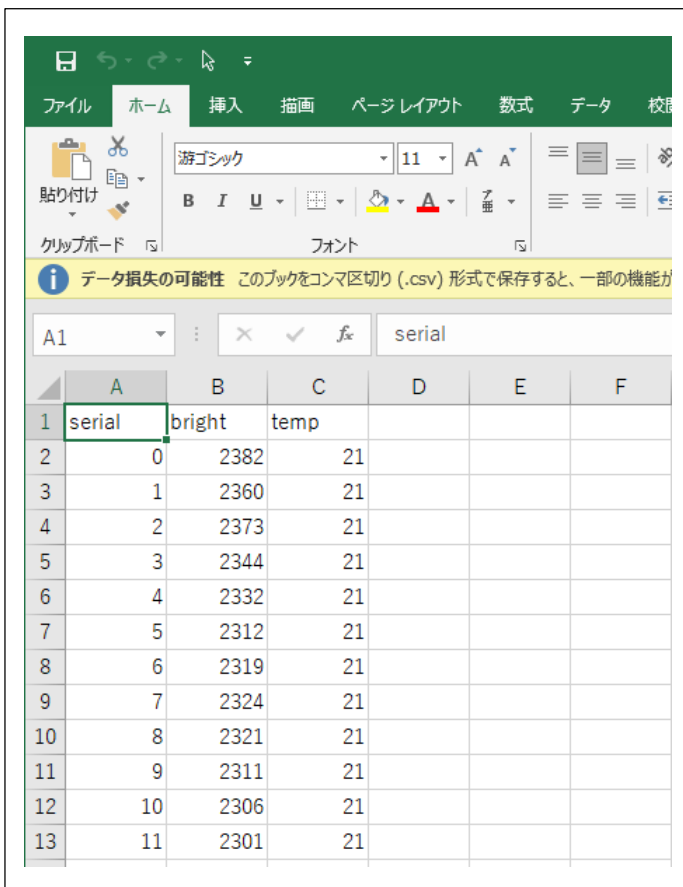
HSB\_CAN\_MULTI\_3 がデータフレーム(データフレーム内に SPI フラッシュメモリの情報)送信

HSB\_CAN\_MULTI\_4 から PC に UART でデータ転送

という、伝言ゲームみたいなやり取りがあるので、ファイルの保存には、ある程度時間が掛かります

(1 時間記録した 3600 点のデータ取得に、約 18 秒掛かります)

出力されたファイルは、CSV 形式になっており、



	A	B	C	D	E	F
1	serial	bright	temp			
2	0	2382	21			
3	1	2360	21			
4	2	2373	21			
5	3	2344	21			
6	4	2332	21			
7	5	2312	21			
8	6	2319	21			
9	7	2324	21			
10	8	2321	21			
11	9	2311	21			
12	10	2306	21			
13	11	2301	21			

データのシリアル番号、明るさセンサの値、温度センサの値が記録されたものになります。

HSB\_CAN\_MULTI\_3

受信CANデータ

情報取得 (13)

CAN-ID	RTR	IDE	DLC	data	TS
00000203	00	01	08	0A4A086600000000	EC0E

明るさセンサ 2634 (15)

温度センサ 21.5 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(#) (17)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(#) 0 (18)

送信 (19)

SPIフラッシュ保存 (20)

「SPIフラッシュ消去」「送信」

を行うと、SPIフラッシュの全消去が行えます。

HSB\_CAN\_MULTI\_3

受信CANデータ

情報取得 (13)

CAN-ID	RTR	IDE	DLC	data	TS
00000203	00	01	08	0A4A086600000000	EC0E

明るさセンサ 2634 (15)

温度センサ 21.5 (16)

センサ値読み出し (17)  
 SPIフラッシュリード(#) (17)  
 SPIフラッシュ消去  
 センサ値記録ON  
 センサ値記録OFF

SPIフラッシュアドレス(#) 0 (18)

送信 (19)

SPIフラッシュ保存 (20)

「SPIフラッシュリード」SPフラッシュアドレス「0」「送信」

「情報取得」

を行うと、

アドレス 0x0~0x7

のデータが、0xFFとなっており、SPIフラッシュが消去されていることが確認できます。

(「センサ値記録 ON」のコマンドを送ると、消去後に書き込みを始めますので、記録前に消去する必要はありません。)

## HSB\_CAN\_MULTI\_4 に対する操作

HSB\_CAN\_MULTI\_4

送信データ

マスタ送信 (21)  (22)

スレーブヘッダ送信(23)  (24)

LIN-ID

LIN-ID data SUM

(25)

動作モード(SCI) (26)

動作モード(タイマ) (27)

最初に、(26)の「動作モード(SCI)」を押してください。

HSB\_CAN\_MULTI\_4 は、起動後 LIN はタイマモードで動作していますので、本アプリケーションとは無関係に LIN のヘッダ送信を行っています。「動作モード(SCI)」のボタンを押すことにより、HSB\_CAN\_MULTI\_4 が本アプリケーションで制御される様になります。

(「動作モード(タイマ)」は、起動後のタイマモードに戻すボタンです。)

HSB\_CAN\_MULTI\_4

送信データ

マスタ送信 (21)  (22)

スレーブヘッダ送信(23)  (24)

LIN-ID

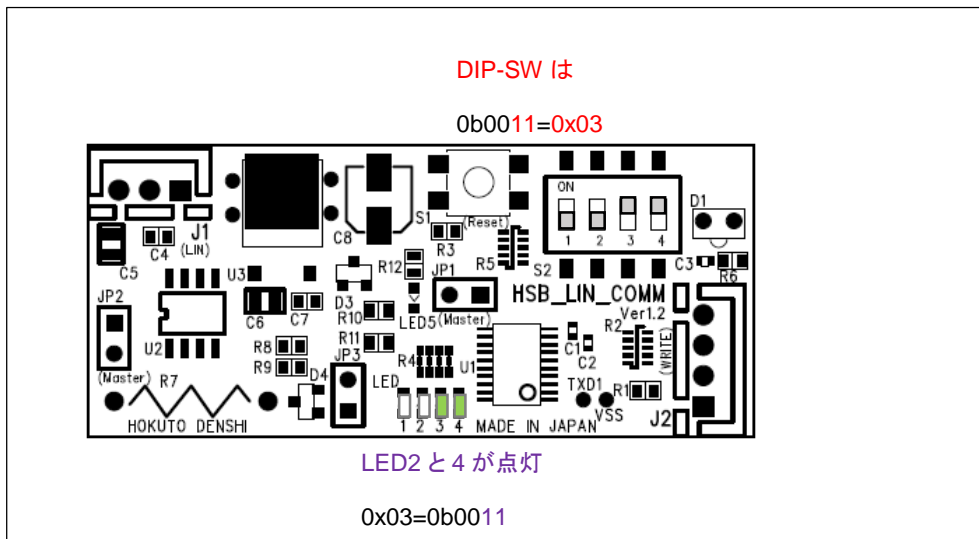
LIN-ID data SUM

(25)

動作モード(SCI) (26)

動作モード(タイマ) (27)

(22)の送信データ(0~9,A-F の文字を 16 桁で入力…8 バイトデータ)を設定し、(21)の「マスタ送信」のボタンを押すと、HSB\_CAN\_MULTI\_4 から LIN バスに、ID=0x30 で MASTER レスポンス送信を行います。送信されるのは、(22)に設定したデータです。



HSB\_CAN\_MULTI\_4

送信データ

マスタ送信 (21) 0102030405060708 (22)

スレーブヘッダ送信 (23) 31 (24)

LIN-ID data SUM

31 0F70030302930000 33 (25)

動作モード(SCI) (26)

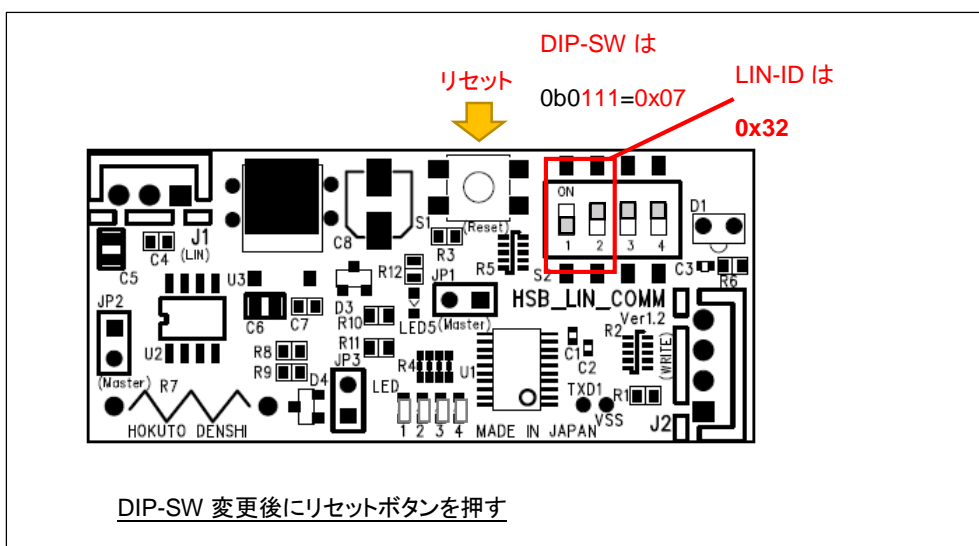
動作モード(タイマ) (27)

次に(24)の LIN-ID に「31」(0x31 の意)を入力し、(23)の「スレーブヘッダ送信」を押します。この場合、(25)の受信 LIN データの部分が更新されるはずですが。

(data の 3 バイト目は、LIN-ID=0x31 の HSB\_LIN\_COMM の DIP-SW の状態が反映され、4 バイト目は LED の点灯状況が反映されるはずですが。)

LIN の動作モードを「動作モード(SCI)」にした場合は、(21)(23)のボタンを押したタイミングで、ヘッダ送信が実行されます。なお、LIN バスに、LIN-ID=0x31 以外の ID を持つデバイスが存在する場合は、(24)の LIN-ID に ID 値を入れ、「スレーブヘッダ送信」を行うと、当該デバイスが応答を返すはずですが。

HSB\_LIN\_COMM の LIN-ID を 0x32 に変更。



HSB\_CAN\_MULTI\_4

送信データ

マスタ送信 (21) 0102030405060708 (22)

スレーブヘッダ送信 (23) 32 (24)

LIN-ID data SUM

32 0005070001170000 A9 (25)

動作モード(SCI) (26)

動作モード(タイマ) (27)

LIN-ID「32」「スレーブヘッダ送信」を押すと、HSB\_LIN\_COMM(LIN-ID=0x32)が応答を返す。(受信データの 3 バイト目は 0x7 で DIP-SW の状態を返している。HSB\_LIN\_COMM は簡単に LIN-ID を変更できます。)

以上、PC で動作するデモプログラムの説明です。

本キットでは、CAN, LIN の通信を用いて、PC から CAN/LIN データのモニタを行ったり、ボードに指示を出す事が可能で、CAN や LIN の動作を実際に体験できる様になっています。

ご自身でマイコンボードで動いているプログラムや、PC 向けのデモプログラムに手を加えてカスタマイズしたいという場合は、別売となっている「CAN マルチネットワーク ソースコード CD」をご購入頂き、プログラムの動作をカスタマイズする事が可能です。（「CAN マルチネットワーク ソースコード CD」には、マイコンボード上で動作するプログラム、PC 上で動作するプログラムの両方のソースコードが含まれます）

## 取扱説明書改定記録

バージョン	発行日	ページ	改定内容
REV.1.0.0.0	2023.6.6	—	初版発行

## お問い合わせ窓口

最新情報については弊社ホームページをご活用ください。

ご不明点は弊社サポート窓口までお問い合わせください。

株式会社 **北斗電子**

〒060-0042 札幌市中央区大通西 16 丁目 3 番地 7

TEL 011-640-8800 FAX 011-640-8801

e-mail: support@hokutodenshi.co.jp (サポート用)、order@hokutodenshi.co.jp (ご注文用)

URL: <https://www.hokutodenshi.co.jp>

商標等の表記について

- ・ 全ての商標及び登録商標はそれぞれの所有者に帰属します。
- ・ パーソナルコンピュータを PC と称します。



---

ルネサス エレクトロニクス社 RX231, RL78/F15, RA2L1 搭載  
HSB シリーズ応用キット

# CAN マルチネットワークボード取扱説明書 通信編

株式会社 **北斗電子**

©2023 北斗電子 Printed in Japan 2023 年 6 月 6 日改訂 REV.1.0.0.0 (230606)

---