

# PCAN-LIN

RS-232 to LIN/CAN Interface

## Protocol Definitions Documentation



Document version 1.1.0 (2016-01-14)

**PEAK**  
System

## 関連商品

Product Name	Firmware version
PCAN-LIN	01.02.xx

PCAN®は、PEAK-System Technik GmbH の登録商標です。

本書に記載されているその他すべての製品名は、それぞれの会社の商標または登録商標である可能性があります。“TM” および “®” で明示的にマークされていません。

©2016PEAK-System Technik GmbH

この文書の複製（コピー、印刷、またはその他の形式）および電子配布は、PEAK-System Technik GmbH の明示的な許可がある場合にのみ許可されます。PEAK-System Technik GmbH は、事前の発表なしに技術データを変更する権利を留保します。一般的なビジネス条件とライセンス契約の規制が適用されます。すべての権利は留保されています。

PEAK-System Technik GmbH

Otto-Roehm-Strasse 69

64293 Darmstadt

Germany

Phone: +49 (0)6151 8173-20

Fax: +49 (0)6151 8173-29

[www.peak-system.com](http://www.peak-system.com)

[info@peak-system.com](mailto:info@peak-system.com)

Document version 1.1.0 (2016-01-14)

## 目次

<b>1 PCAN-LIN モジュールとの通信</b> .....	<b>5</b>
1.1 シリアルプロトコル (Serial Protocol) .....	5
1.1.1 データ送信ホスト – モジュール (Data Transmission Host – Module) .....	5
1.1.2 データ送信モジュール – ホスト (Data Transmission Module – Host) .....	8
<b>2 インターフェイス (Interfaces)</b> .....	<b>12</b>
2.1 インターフェイスとコマンドに関する注記 .....	12
2.2 RS-232 インターフェイス (RS-232 Interface) .....	13
2.2.1 初期化コマンド (Initialization Commands) .....	13
2.2.2 リード・コマンド (Read Commands) .....	14
2.2.3 設定コマンド (Configuration Commands) .....	17
2.2.4 リセット/消去 コマンド (Reset / Erase Commands) .....	19
2.3 CAN インターフェイス (CAN Interface) .....	21
2.3.1 初期化およびライト・コマンド (Write Commands) .....	21
2.3.2 リード・コマンド (Read Commands) .....	22
2.3.3 コンフィグレーション・コマンド (Configuration Commands) .....	30
2.3.4 リセット/消去 コマンド (Reset / Erase コマンド) .....	36
2.4 LIN インターフェイス (LIN Interface) .....	37
2.4.1 初期化およびライト・コマンド (Write Commands) .....	37
2.4.2 リード・コマンド (Read commands) .....	38
2.4.3 コンフィグレーション・コマンド (Configuration Commands) .....	59
2.4.4 リセット/消去 コマンド (Reset / Erase Commands) .....	81
2.5 モジュール・インターフェイス (Module Interface) .....	88
2.5.1 初期化コマンド (Initialization Commands) .....	88
2.5.2 リード・コマンド (Read Commands) .....	89
2.5.3 コンフィグレーション・コマンド (Configuration Commands) .....	93
2.5.4 リセット/消去 コマンド (Reset / Erase Commands) .....	93
2.6 メッセージの送信と受信 (Message Transmission and Reception) .....	95
2.6.1 CAN .....	95
2.6.2 LIN .....	98

<b>3 エラー・メッセージ (Error Messages)</b> .....	<b>103</b>
3.1 RS-232 リターンコード (RS-232 Return Codes) .....	103
3.2 CAN エラー・コード (CAN Error Codes) .....	104
3.3 LIN エラー・コード (LIN Error Codes) .....	105
<b>4 シリアル・トレース分析 (Serial Trace Analysis)</b> .....	<b>107</b>

## 1 PCAN-LIN モジュールとの通信

PCAN-LIN モジュールとの主な通信は、RS-232 インターフェイスを介して行われます。これは、実行中の操作、影響を受けるモジュールのインターフェイス、必要なパラメーター、CAN および LIN メッセージによって送信されるデータ、またはスケジューラー・エントリーに関する情報を含む構造に分割されています。後者の場合はより多くのデータが必要になるため、“Write Data” および “Scheduler Entry” コマンドはより複雑な構造を使用します。

合計 8 つの異なるストラクチャが使用されます。ホストとモジュール間の通信用に 2 つ、CAN メッセージのリードとライト用に 2 つ、LIN メッセージのリードとライト用に 2 つ、LIN マスター・スケジューラーのエントリーを取得、および設定するためのものに 2 つです。

### 1.1 シリアルプロトコル (Serial Protocol)

#### 1.1.1 データ送信ホスト – モジュール (Data Transmission Host – Module)

Byte	Description	Bit position							
		7	6	5	4	3	2	1	0
1	STX	0	0	0	0	0	0	1	0
2	Sequence code (シーケンス・コード)	X	Sequence no. (シーケンス・ナンバー)			Parameter byte count (パラメーターのバイト数)			
3	Control code (コントロール・コード)	Target interface (ターゲット・インターフェイス)		Command code (コマンド・コード)					
4…n	Parameters (パラメーター)	パラメーターの数は、Sequence code (シーケンス・コード) の Parameter byte count (パラメーターのバイト数) 部分によって異なります。							
n+1	Checksum (チェックサム)	STXバイト (最初のバイト、開始バイト) を除くこのメッセージのすべてのバイト間のビット単位のexclusive-OR (排他的論理和) 演算。							

## STX

これは、モジュール通信のメッセージの“Start Transmission”コードです。このコードの固定値は2 (0x2、0b10) です。この開始バイトのないメッセージは、モジュールメッセージとして認識されません。

## Sequence code (シーケンス・コード)

ホストからモジュールへのメッセージの2番目のバイトは、送信されるデータの Sequence Number (シーケンス・ナンバー) と長さを示します。Sequence code (シーケンス・コード) のフィールド長は7ビットです。このフィールドの最上位ビットは使用されません。将来のファームウェア・バージョンとの互換性を保証するために、0に設定することをお勧めします。

**Sequence Number (シーケンス・ナンバー) :** 0 から 7 までのこのナンバーは、メッセージを送信するための“User Control Number”です。その目的は、セグメント化されたデータとして送信できるメッセージの区別を支援することです。メッセージは 15 バイトを超えることはできません。ホストからのメッセージ (command) がモジュールからの応答メッセージを生成する場合、この応答には、それを引き起こしたメッセージと同じ Sequence Number (シーケンス・ナンバー) が付属します。

**Parameter byte count (パラメーター・バイト数) :** Sequence code (シーケンス・コード) のこれらの4ビットは、メッセージのパラメーター/データ・バイト数を示します。LIN コミュニケーション・メッセージは 15 バイトを超えることはできません (Parameter byte count : パラメーター・バイト・カウントの最大値は 15 = 4 ビットです)。

例えば、Sequence Number (シーケンス・ナンバー) が 3 でパラメーターが 2 のメッセージには、Sequence code (シーケンス・コード) 50 (0x32、0b0110010) になります。

## Control code (コントロール・コード)

このバイトは、処理されるコマンドと、コマンドの送信先となるモジュールのインターフェイスを示します。最上位2ビットはインターフェイスを示し、他のビットは Command code (コマンド・コード) を示します。

**Target interface (ターゲット・インターフェイス)** : 2 ビットの長さのこれらのフィールドは、4 つのインターフェイスのどれが “Command code” フィールドの Command code (コマンド・コード) に関連付けられているかを示します。可能なすべての Command code (コマンド・コード) と説明は、12 ページの第 2 章 インターフェイス で説明されています。

**Command code (コマンド・コード)** : 長さが 6 ビットの場合、このフィールドは 0~63 のコマンドを示すことができます。コマンドは 4 つのグループに分けられます。可能なすべてのコマンドのコードと説明は、12 ページの第 2 章 インターフェイス で説明されています。

例えば、CAN インターフェイスのモジュールのビットレートを設定するメッセージには、制御コード 95 (0x5F、0b01011111) になります。CAN インターフェイスのコード= 1。ビットレートパラメーターのコード= 31。

#### Parameters (パラメーター)

制御コードに続いて、LIN 通信メッセージには、Sequence code (シーケンス・コード) の Parameter byte count (パラメーター・バイト・カウント) フィールドに示されている数のパラメーター/データ・バイトが含まれます。例えば、パラメーターのバイト数が 2 (0x2、0b0010) の場合、モジュールは、制御コードの後の次の 2 つの受信バイトをパラメーター/データとして解釈します。このパラメーター/データのフォーマットは、各コマンドによって異なります。

例えば、CAN インターフェイスの新しいビットレートを含むメッセージには、目的のビットレートを示す 2 つのパラメーター・バイトがあります。パラメーターは、最下位バイト (LSB) から始めて配置する必要があります。500 kbit / s (0x001C) のビットレートを設定するために、メッセージの最初のパラメーターとして値 28 (0x1C、0b11100) とし、2 番目のパラメーターとして値 0 (0x0、0b0) にします。したがって、パラメータービットは 0001110000000000 です。

#### Checksum (チェックサム)

LIN コミュニケーション・メッセージの最後のバイトは Checksum (チェックサム) です。このバイトは、開始バイト (STX) を除くメッセージのすべてのコンポーネントのビット間の Exclusive-OR (排他的論理和演算) の結果です。

例えば、Sequence code (シーケンス・コード) が 50、Control code (コントロール・コード) が 95、2つの Parameter (パラメーター) 28 と 0 のメッセージは、Checksum Byte (チェックサム・バイト) として値 113 (0x71、0b01110001) を持ちます。

要約すると、モジュールの CAN インターフェイスを Sequence Number (シーケンス・ナンバー) 3 を使用して 500 kbit / s のビットレートに設定する LIN コミュニケーション・メッセージは、次のデータを RS-232 ポートに送信することによって実行されます (出現順序) 、 左から右へ) :

- 16 進表記 : 02 – 32 – 5F – 1C – 00 – 71
- 10 進表記 : 2 – 50 – 95 – 28 – 0 – 113
- バイナリ表記 :

1	STX	0	0	0	0	0	0	1	0
2	Sequence code	0	0	1	1	0	0	1	0
3	Control code	0	1	0	1	1	1	1	1
4	Parameter0	0	0	0	1	1	1	0	0
5	Parameter1	0	0	0	0	0	0	0	0
6	Checksum	0	1	1	1	0	0	0	1

### 1.1.2 データ送信モジュール – ホスト (Data Transmission Module – Host)

Byte	Description	Bit position							
		7	6	5	4	3	2	1	0
1	STX	0	0	0	0	0	0	1	0
2	Sequence code (シーケンス・コード)	AR	Sequence no. (シーケンス・ナンバー)			Parameter byte count (パラメーターのバイト数)			
3...n	Parameters (パラメーター)	パラメーターの数は、Sequence code (シーケンス・コード) の Parameter byte count (パラメーターのバイト数) 部分によって異なります。							
n+1	Checksum (チェックサム)	STXバイト (最初のバイト、開始バイト) を除くこのメッセージのすべてのバイト間のビット単位のexclusive-OR (排他的論理和) 演算。							

## STX

これは、モジュール通信のメッセージの“Start Transmission”（送信開始）コードです。このコードの固定値は 2 (0x2、0b10) です。この開始バイトのないメッセージは、モジュールメッセージとして認識されません。

## Sequence code（シーケンス・コード）

モジュールからホストへのメッセージの 2 番目のバイトは、受信するデータの Sequence Number（シーケンス・ナンバー）と長さを示します。このフィールドの最上位ビットは **Auto Reply**（自動応答）ビットと呼ばれ、その使用はモジュール用に予約されています。これは、転送されたメッセージ（**AR** が 1 に設定されている）とホストから送信されたコマンドへの応答を区別するために導入されました。

**Sequence Number（シーケンス・ナンバー）** : 0 から 7 までのナンバーは、メッセージの“User Control Number”（ユーザー管理番号）です。メッセージがホストによって行われた要求への応答である場合、Sequence Number（シーケンス・ナンバー）は要求メッセージで使用されたものと同じ Sequence Number（シーケンス・ナンバー）です。メッセージが **Auto Reply**（自動応答）メッセージの場合、Sequence Number（シーケンス・ナンバー）は次のように定義されます。

Bit	Value	Description
6	0	未使用
5/4	00 = RS-232 01 = CAN 02 = LIN 03 = Module	ソース

**Parameter byte count（パラメーター・バイト）** : Sequence code（シーケンス・コード）の 4 ビットは、Sequence code（シーケンス・コード）の後、Checksum Byte（チェックサム・バイト）の前に着信メッセージでパラメーター/データとして解釈されるバイト数を示します。

例えば、上記のセクション 1.1.1 (CAN ビットレートの設定) の例の応答は 49 (0x31, 0b00110001) です。この値は、Sequence number field (シーケンス・ナンバー・フィールド) で 3 (上記で使用したものと同一) に対応し、Parameter byte count Field (パラメーター・バイト・カウント・フィールド) で 1 (回答として 1 バイト) に対応します。このデータ・バイトは、操作が成功したかどうかを示します。詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

#### Parameter (パラメーター)

モジュールからの着信 LIN コミュニケーション・メッセージには、Sequence code byte (シーケンス・コード・バイト) の後に、Parameter byte count (パラメーター・バイト・カウント) フィールドに示されている数のパラメーター／データ・バイトが含まれます。このデータを解釈する方法は、要求された情報によって異なります。詳細については、このドキュメントで後述する目的のコマンドの可能な戻り値を参照してください。

例えば、CAN ビットレートを設定するコマンドが送信され、この操作が成功した場合、着信メッセージにはパラメーターが 1 つだけあり、これは 0 (エラーなし) です。

#### Checksum (チェックサム)

着信 LIN コミュニケーション・メッセージの最後のバイトは Checksum (チェックサム) です。このバイトは、開始バイト (STX) を除くメッセージのすべてのコンポーネントのビット間の exclusive-OR (排他的論理和) 演算の結果です。

例えば、Sequence code (シーケンス・コード) が 49 でパラメーターが 1 のメッセージの場合、Checksum Byte (チェックサム・バイト) として値は 49 (0x31, 0b00110001) になります。

要約すると、RS-232 ポートの CAN インターフェイスのビットレート (前のセクションの例として、要求メッセージで使用された値に関して) を設定した結果を示す着信 LIN コミュニケーション・メッセージは次のデータを受信します :

- 16 進表記 : 02 – 31 – 00 – 31
- 10 進表記 : 2 – 49 – 0 – 49
- バイナリ表記 :

1	STX	0	0	0	0	0	0	1	0
2	Sequence code	0	0	1	1	0	0	0	1
3	Parameter0	0	0	0	0	0	0	0	0
4	Checksum	0	0	1	1	0	0	0	1

## 2 インターフェイス (Interfaces)

### 2.1 インターフェイスとコマンドに関する注記

PCAN-LIN モジュールには、RS-232、CAN、LIN、およびモジュールの4つのロジカル・インターフェイスがあります。コマンドは、すべてのインターフェイスで有効であるとは限りません。各インターフェイスには、コマンドを送信するためのコードがあります。これらのコードは次のとおりです：

- RS-232: 0
- CAN: 1
- LIN: 2
- Module: 3

#### RS-232 (0)

このインターフェイスは、モジュールの RS-232 インターフェイスを参照する情報をコンフィグレーションおよび、リセット/消去するために使用されます。このインターフェイスは、モジュールに必要なすべてのコンフィグレーションを行うために使用されます。

#### CAN (1)

このインターフェイスは、モジュールの CAN 機能の設定をコンフィグレーションおよび、リセット/消去するために使用されます。

#### LIN (2)

このインターフェイスは、モジュールの LIN 機能の設定をコンフィグレーションおよび、リセット/消去するために使用されます。

#### Module (3)

このインターフェイスは、モジュールに関する一般情報をハードウェアおよびファームウェア・バージョンとしてリード、リセットするために使用されます。

コマンドは、タスクに応じて4つのグループに配置されています：

初期化、リード、コンフィグレーション、リセット/削除。コマンド "Write Data" は、より複雑な構造を使用するため、個別に扱われます。

次のセクションには、説明されている各コマンドメッセージの Sequence code (シーケンス・コード) と Control code (コントロール・コード) の値を示す表が含まれています。Sequence Number (シーケンス・ナンバー) はデフォルトで 0 です。別の Sequence Number (シーケンス・ナンバー) を使用する場合は、このフィールドを再計算する必要があります。

#### 略語

- STX : 開始バイト。常に 0x02 です。
- SC : Sequence code (シーケンス・コード)
- CC : 制御コード
- PX : Parameters (パラメーター)。X は、Parameters (パラメーター) の相関数 (1...n) を表します。
- CHK : Checksum (チェックサム)。Checksum (チェックサム) は、開始バイト (STX) を除くメッセージのすべてのコンポーネントのビット間で exclusive-OR operation (排他的論理和演算) を実行して計算されます。

## 2.2 RS-232 インターフェイス (RS-232 Interface)

### 2.2.1 初期化コマンド (Initialization Commands)

Command	on page
現在のパラメーターによる初期化	13

#### 現在のパラメーターによる初期化

このコマンドは、設定されたパラメーターで RS-232 インターフェイスを初期化します。この場合、コンフィグレーション・コマンドを使用して設定できるパラメーターはボーレートのみです。コンフィグレーション・コマンドが成功した場合、このコマンドが送信されるまで、新しいボーレートは使用されません。

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0, 0x0			0, 0x0		0, 0x0		0, 0x0				
0, 0x0					0, 0x0							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x00) XOR PX(0x00) = 0x0
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x00) – CHK(0x00)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：**使用するRS-232 ボーレートを変更して初期化した後、ホストとの接続を新しい値に調整する必要があります。したがって、モジュールが応答しなくなるため、新しいボーレートを使用したホストとモジュール間の再接続が必要になります。

## 2.2.2 リード・コマンド (Read Commands)

Command	on page
ボーレート (Baud Rate) を取得	14
RCV エラー・カウンター (RCV Error Counter) を取得	15
XMT エラー・カウンター (XMT Error Counter) を取得	16
すべてのエラー・カウンター (All Error Counters) を取得	16

### ボーレート (Baud Rate) を取得

このコマンドは、RS-232 インターフェイスの設定されたボーレートを取得します。

受信するデータは LSB から始まります。例：0x9600 = P1 : 0x00 および P2 : 0x96

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code			
0	0	0	0	0	0	0	0	0	0	1	0	1	1
0	0, 0x0			0, 0x0				0, 0x0		11, 0xB			
0, 0x0					11, 0xB								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x0B) XOR PX(0x00) = 0x0B
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x0B) – CHK(0x0B)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x96) – CHK(0x94)

肯定応答には、構成されたボーレートを持つ 2 つのパラメーターがあります。この例では、デフォルトのボーレートが P1 および P2 : 0x9600 = 38400 Baud で返されます。 ※ Baud は、変調レート



**注：** 設定されたボーレートは、使用されたボーレートとは異なります。使用されたボーレートは、決められた時間内にモジュールとの間でメッセージを送受信するために使用される速度です。構成されたボーレートは、ボーレート構成メッセージを送信してボーレートを変更したが、初期化していない結果です。新しく設定された値をアクティブにするには、セクション 2.2.1 初期化コマンド を参照してください。

### RCV エラー・カウンター (RCV Error Counter) を取得

このコマンドは、RS-232 インターフェイスの受信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code			
0	0	0	0	0	0	0	0	0	1	0	1	1	0
0	0, 0x0			0, 0x0				0, 0x0		22, 0x16			
0, 0x0					22, 0x16								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x16) XOR PX(0x00) = 0x16
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x16) – CHK(0x16)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、RCV エラー・カウンターの値を持つ 2 つのパラメーターがあります。この例では、デフォルト値が P1 および P2 : 0x0 = 0 エラーで返されます。

**XMT エラー・カウンター (XMT Error Counter) を取得**

このコマンドは、RS-232 インターフェイスの送信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります。 例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)							
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
0	0, 0x0			0, 0x0			0, 0x0		23, 0x17					
0, 0x0							23, 0x17							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x17) XOR PX(0x00) = 0x17
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x17) – CHK(0x17)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、XMT エラー・カウンターの値を持つ 2 つのパラメーターがあります。この例では、デフォルト値が P1 および P2 : 0x0 = 0 エラーで返されます。

**すべてのエラー・カウンター (All Error Counters) を取得**

このコマンドは、RS-232 インターフェイスのエラー・カウンターの値を取得します。戻り値は、RCV エラー・カウンターの値、XMT エラー・カウンターの値、および Checksum error counter (チェックサム・エラー・カウンターの値 (この順序で) です。各値の長さは 2 バイトです。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)										
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0, 0x0			0, 0x0			0, 0x0		24, 0x18					
0, 0x0				24, 0x18										

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x18) XOR PX(0x00) = 0x18
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x18) – CHK(0x18)
<b>Positive response</b>	STX(0x02) – SC(0x06) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0x00) – CHK(0x06)

肯定応答には、RCV、XMT、および Checksum error counter (チェックサム・エラー・カウンター) の値を持つ 6 つのパラメーターがあります。この例では、以下が返されます。

- P1 および P2 の場合、RCV error counter (RVC エラー・カウンター) のデフォルト値：0x0 = 0 エラー
- P3 および P4 の場合、XMT error counter (XML エラー・カウンター) のデフォルト値：0x0 = 0 エラー
- P5 および P6 の場合、Checksum error counter (チェックサム・エラー・カウンター) のデフォルト値：0x0 = 0 エラー

### 2.2.3 設定コマンド (Configuration Commands)

Command	on page
ボーレート (Baud Rate) を設定	17

#### ボーレート (Baud Rate) を設定

変更できる RS-232 インターフェイスの唯一のパラメーターはボーレートです。特定の許可されたボーレートがいくつかあります。

許可リストにないボーレートを設定しようとする、モジュールによって否定応答メッセージが開始され、ボーレートは変更されません。

パラメーター：目的のボーレートを指定するには、2つのパラメーター（2バイト）が必要です。次の表に、許可されるボーレートとそのパラメーター表記 P1-P2 を示します。

Baud rate	Hex value	P1	P2
4,800	0x12C0	0xC0	0x12
9,600	0x2580	0x80	0x25
19,200	0x4B00	0x00	0x4B
38,400	0x9600	0x00	0x96
57,600	0xE100	0x00	0xE1

送受信するデータは LSB から始まります。例：0x12C0 = P1：0xC0 および P2：0x12

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	0	0	1	1	1	1	1
0	0, 0x0			2, 0x2		0, 0x0		31, 0x1F					
2, 0x2					31, 0x1F								

<b>Parameters</b>	ボーレート、2つのパラメーター
<b>Checksum</b>	PX = P1(0x00) XOR P2(0x96) = 0x96 SC(0x02) XOR CC(0x1F) XOR PX(0x96) = 0x8B
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x1F) – P1(0x00) – P2(0x96) – CHK(0x0B)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：**構成されたボーレートは、使用されたボーレートとは異なります。使用されるボーレートは、決められた時間内にモジュールとの間でメッセージを送受信するために使用される速度です。構成されたボーレートは、ボーレート・コンフィグレーション・メッセージを送信してボーレートを変更したが、初期化していない結果です。新しく設定された値をアクティブにするには、13ページのセクション 2.2.1 初期化コマンドを参照してください。

## 2.2.4 リセット/消去 コマンド (Reset / Erase Commands)

Command	on page
RVC エラー・カウンター (RCV error counter) をリセット	19
XML エラー・カウンター (XMT error counter) をリセット	20
チェックサム・エラー・カウンター (Checksum error counter) をリセット	20

### RCV エラー・カウンター (RCV Error Counter) をリセット

このコマンドは、RS-232 インターフェイスの受信エラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。 例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
0	0, 0x0			0, 0x0			0, 0x0		52, 0x34					
0, 0x0					52, 0x34									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x34) XOR PX(0x00) = 0x34
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x34) – CHK(0x34)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第3章 エラー・メッセージを参照してください。

**XMT エラー・カウンター (XMT Error Counter) をリセット**

このコマンドは、RS-232 インターフェイスの伝送エラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)							
X	Sequence	Param Byte Count		Interface		Command Code					
0	0 0 0 0	0 0 0 0		0	0	1	1	0	1	0	1
0	0, 0x0	0, 0x0		0, 0x0		53, 0x35					
0, 0x0			53, 0x35								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x35) XOR PX(0x00) = 0x35
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x35) – CHK(0x35)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第3章 エラー・メッセージを参照してください。

**チェックサム・エラー・カウンター (Checksum error counter) をリセット**

このコマンドは、RS-232 インターフェイスのチェックサム・エラー・カウンター (Checksum error counter) をリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)							
X	Sequence	Param Byte Count		Interface		Command Code					
0	0 0 0 0	0 0 0 0		0	0	1	1	0	1	1	1
0	0, 0x0	0, 0x0		0, 0x0		55, 0x37					
0, 0x0			55, 0x37								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x37) XOR PX(0x00) = 0x37
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x37) – CHK(0x37)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

## 2.3 CAN インターフェイス (CAN Interface)

### 2.3.1 初期化およびライト・コマンド (Initialization and Write Commands)

Command	on page
現在のパラメーターに合わせて初期化する	21
バスにデータを書き込む (Write Data on Bus)	22

#### 現在のパラメーターに合わせて初期化

このコマンドは、設定されたパラメーターを使用して CAN インターフェイスを初期化します。

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0, 0x0			0, 0x0		1, 0x1		0, 0x0						
0, 0x0					64, 0x40									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x40) XOR PX(0x00) = 0x40
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x40) – CHK(0x40)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### バスにデータをライト (Write Data on Bus)

コマンド "Write Data" は、より複雑な構造を使用するため、個別に扱われます。セクション メッセージの送信と受信 > 2.6.1 CAN (95 ページ) を参照してください。

### 2.3.2 リード・コマンド (Read Commands)

Command	on page
アクティベーション・ステータス (Activation Status) を取得	22
ビットレート (Bit Rate) を取得	23
フォワード・マスク (Forward Mask) を取得	24
フィルタ・マスク (Filter Mask) を取得	25
フィルタ・コード (Filter Code) を取得	26
LIN メッセージの CAN ID オフセット (CAN ID Offset for LIN Messages) を取得	26
RCV エラー・カウンター (RCV Error Counter) を取得	27
XMT エラー・カウンター (XMT Error Counter) を取得	28
すべてのエラー・カウンター (All Error Counters) を取得	28

### アクティベーション・ステータス (Activation Status) を取得

このコマンドは、モジュールの CAN 機能がアクティブであるかどうかを示す値を取得します。そうでない場合でも、インターフェイスの構成は可能ですが、イベントを処理するために CAN 機能が必要な場合は効果がありません。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	1	0	1	0
0	0, 0x0			0, 0x0		1, 0x1		10, 0xA						
0, 0x0					74, 0x4A									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4A) XOR PX(0x00) = 0x4A
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4A) – CHK(0x4A)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x01) – P2(0x00) – CHK(0x03)

肯定的な応答には 2 つのパラメーターがあります。 P1 と P2（この場合は 0x01）によって形成される値が 0 より大きい場合、これは “Interface is active” であることを意味し、そうでない場合は “Interface is not active” ことを意味します。

### ビットレート (Bit Rate) を取得

このコマンドは、CAN インターフェイスの設定済みビットレートを取得します。ビットレートは Btr0Btr1 フォーマットです。

受信するデータは LSB から始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	1	0	1	1
0	0, 0x0			0, 0x0		1, 0x1		11, 0xB						
0, 0x0					75, 0x4B									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4B) XOR PX(0x00) = 0x4B
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4B) – CHK(0x4B)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x1C) – P2(0x00) – CHK(0x1E)

肯定応答には、構成されたビットレートを持つ 2 つのパラメーターがあります。この例では、デフォルトのビットレートが P1 と P2 で返されます：0x0010 = 500 kbit / s



**注：** 設定されたビットレートは、使用されたビットレートとは異なります。使用ビットレートは、決められた時間内にモジュールとの間でメッセージを送受信するために使用される速度です。構成済みビットレートは、ビットレート構成メッセージを送信してビットレートを変更したが、初期化していない結果です。新しく設定された値をアクティブにするには、21 ページのセクション 2.3.1 初期化およびライト・コマンドを参照してください。

## フォワード・マスク (Forward Mask) を取得

このコマンドは、モジュール内の CAN インターフェイスの設定済みフォワード・マスクを取得します。このマスクは、受信したメッセージを他のインターフェイスに転送するインターフェイスと、エラー情報を監視するかどうかをインターフェイスに通知します。エラーは RS-232 インターフェイスにリダイレクトすることもできます。

フォワード・マスク値は、次の表の値のビットの組み合わせ（ビット単位の OR 演算）です：

Mask value	Meaning
0x80	エラー情報を RS-232 インターフェイスに転送する
0x40	エラー情報を CAN インターフェイスに転送する
0x04	CAN メッセージを RS-232 インターフェイスに転送する
0x02	CAN メッセージを LIN インターフェイスに転送する

例えば、0x86 のフォワード・マスクは、CAN インターフェイスのエラー情報が RS-232 インターフェイスに送信され、受信した CAN メッセージが LIN と RS-232 インターフェイスの両方に転送されることを意味します。

受信するデータは LSB から始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)				Control Code (CC)							
X	Sequence	Param Byte Count		Interface	Command Code						
0	0 0 0 0	0 0 0 0		0 1	0 0 1 1 0 0						
0	0, 0x0	0, 0x0		1, 0x1	12, 0xC						
0, 0x0				76, 0x4C							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4C) XOR PX(0x00) = 0x4C
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4C) – CHK(0x4C)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。この例では、戻り値はデフォルト値です（転送なし）。値が 0 と異なる場合は、マスク・テーブルの値を使用してビット単位の AND 演算を実行することにより、どの種類の転送が構成されているかを判断できます。例えば、戻り値が 0x86 の場合、AND 演算は次のようになります。

- (0x86 AND 0x80 = 0x80) の場合、エラー情報を RS-232 インターフェイスに転送します
- (0x86 AND 0x04 = 0x04) の場合、CAN メッセージを RS-232 インターフェイスに転送します
- (0x86 AND 0x02 = 0x02) の場合、CAN メッセージを LIN インターフェイスに転送します

### フィルタ・マスク (Filter Mask) を取得

このコマンドは、モジュール内の CAN インターフェイスの設定済みアクセプタンス・マスクを取得します。CAN アクセプタンス・マスクは、アクセプタンス・コードとともに、受信したメッセージを CAN インターフェイスに通知するために使用されるフィルタです。このフィルタは、CAN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例：0x221100FF = P1 : 0xFF、P2 : 0x00、P3 : 0x11、および P4 : 0x22

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	1	1	0	1
0	0, 0x0			0, 0x0		1, 0x1		13, 0xD						
0, 0x0					77, 0x4D									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4D) XOR PX(0x00) = 0x4D
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4D) – CHK(0x4D)
<b>Positive response</b>	STX(0x02) – SC(0x04) – P1(0xFF) – P2(0xFF) – P3(0xFF) – P4(0xFF) – CHK(0x04)

肯定的な応答には 4 つのパラメーターがあります。この例では、戻り値は Filter Mask のデフォルト値 (0xFFFFFFFF) です。CAN フィルタの詳細については、CAN コントローラー Philips SJA1000 の Application Note AN97076 を参照してください。

### フィルタ・コード (Filter Code) を取得

このコマンドは、モジュール内の CAN インターフェイスの構成済みアクセプタンス・コードを取得します。CAN アクセプタンス・コードは、アクセプタンス・マスクとともに、受信したメッセージを CAN インターフェイスに通知するために使用されるフィルタです。このフィルタは、CAN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例：0x221100FF = P1 : 0xFF、P2 : 0x00、P3 : 0x11、および P4 : 0x22

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
0	0, 0x0			0, 0x0		1, 0x1		14, 0xE						
0, 0x0					78, 0x4E									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4E) XOR PX(0x00) = 0x4E
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4E) – CHK(0x4E)
<b>Positive response</b>	STX(0x02) – SC(0x04) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – CHK(0x04)

肯定的な応答には 4 つのパラメーターがあります。この例では、戻り値はフィルタ・コードのデフォルト値 (0x00000000) です。CAN フィルタの詳細については、CAN コントローラー Philips SJA1000 の Application Note AN97076 を参照してください。

### LIN メッセージの CAN ID オフセット (CAN ID Offset for LIN Messages) を取得

このコマンドは、LIN メッセージ用に構成された CAN ID オフセットを取得します。LIN メッセージ ID の長さは 1 バイトです。CAN メッセージを LIN インターフェイスに正常に転送するには、ID オフセットが必要です。

受信するデータは LSB から始まります。例：0x00110022 = P1：0x22、P2：0x00、P3：0x11、および P4：0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	1	1	1
0	0, 0x0			0, 0x0			1, 0x1		15, 0xF					
0, 0x0					79, 0x4F									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x4F) XOR PX(0x00) = 0x4F
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x4F) – CHK(0x4F)
<b>Positive response</b>	STX(0x02) – SC(0x04) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – CHK(0x04)

肯定的な反応の説明：

Param(s)	Description
P1	CAN ID オフセットの最下位バイト
P2	CAN ID オフセットの 2 バイト目 (11 ビット CAN ID の場合は 3 ビットのみ)
P3	CAN ID オフセットの 3 バイト目 (29 ビット CAN ID のみ)
P4	Bit 0~4：CAN ID オフセットの最上位部分 (29 ビット CAN ID のみ)
	Bit 5：未使用
	Bit 6：未使用
	Bit 7：extended CAN ID のフラグ (0 = 11 ビット、1 = 29 ビット)

### RCV エラー・カウンター (RCV Error Counter) を取得

このコマンドは、CAN インターフェイスの受信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります。例：0x1 = P1：0x01 および P2：0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	1	0	1	1	0
0	0, 0x0			0, 0x0			1, 0x1		22, 0x16					
0, 0x0					86, 0x56									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x56) XOR PX(0x00) = 0x56
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x56) – CHK(0x56)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、RCV エラー・カウンターの値を持つ 2 つのパラメーターがあります。この例では、デフォルト値が P1 および P2 で返されます：0x0 = 0 エラー

### XMT エラー・カウンター (XMT Error Counter) を取得

このコマンドは、CAN インターフェイスの送信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	1	0	1	1	1
0	0, 0x0			0, 0x0		1, 0x1		23, 0x17						
0, 0x0					87, 0x57									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x57) XOR PX(0x00) = 0x57
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x57) – CHK(0x57)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、XMT エラー・カウンターの値を持つ 2 つのパラメーターがあります。この例では、デフォルト値が P1 および P2 で返されます：0x0 = 0 エラー

### すべてのエラー・カウンター (All Error Counters) を取得

このコマンドは、CAN インターフェイスのエラー・カウンターの値を取得します。戻り値は、RCV エラー・カウンターの値、XMT エラー・カウンターの値、バス・パッシブ状態エラー・カウンター、およびバス・オフ状態エラー・カウンターです。各値の長さは 2 バイトです。

受信するデータは LSB から始まります。例：0x1 = P1：0x01 および P2：0x00

Sequence Code (SC)								Control Code (CC)							
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0
0	0, 0x0			0, 0x0				1, 0x1		24, 0x18					
0, 0x0								88, 0x58							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x58) XOR PX(0x00) = 0x58
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x58) – CHK(0x58)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0x00) – P7(0x00) – P8(0x00) – CHK(0x08)

肯定応答には、RCV、XMT、バス・パッシブ、およびバス・オフ状態のエラー・カウンターの値を持つ 8 つのパラメータがあります。この例では、以下が返されます：

- P1 および P2 の場合、RCV エラー・カウンターのデフォルト値：0x0 = 0 エラー
- P3 および P4 の場合、XMT エラー・カウンターのデフォルト値：0x0 = 0 エラー
- P5 および P6 の場合、バス・パッシブ状態エラー・カウンターのデフォルト値：0x0 = 0 エラー
- P7 および P8 の場合、バス・オフ状態エラー・カウンターのデフォルト値：0x0 = 0 エラー

### 2.3.3 コンフィグレーション・コマンド (Configuration Commands)

Command	on page
アクティベーション・ステータス (Activation Status) を設定	30
ビットレート (Bit Rate) を設定	31
フォワード・マスク (Forward Mask) を設定	32
フィルタ・マスク (Filter Mask) を設定	33
フィルタ・コード (Filter Code) を設定	34
LIN メッセージの CAN ID オフセット (CAN ID Offset for LIN Messages) を設定	35

#### アクティベーション・ステータス (Activation Status) を設定

このコマンドは、CAN 機能がアクティブ化されているかどうかを示す値をモジュールに設定します。

送受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	1	1	1	1	0
0	0, 0x0			1, 0x1		1, 0x1		30, 0x1E						
1, 0x1					94, 0x5E									

<b>Parameters</b>	CAN インターフェイスのアクティベーション・ステータス (起動状態) を指定するには、1つのパラメーター (1バイト) が必要です。このパラメーター・バイトはブール値として機能します。パラメーターが0の場合、インターフェイスは deactivated (非アクティブ化) され、それ以外の場合は active (アクティブ) に設定されます。(この例では、インターフェイスが アクティブ化されています。)
<b>Checksum</b>	P1 = 0x01 SC(0x01) XOR CC(0x5E) XOR PX(0x01) = 0x5E
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0x5E) – P1(0x01) – CHK(0x5E)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：** インターフェイスのアクティブ化には、initialization コマンドを使用した初期化は必要ありません。

### ビットレート (Bit Rate) を設定

このコマンドは、CAN インターフェイス (Btr0Btr1 フォーマット) で使用されるビットレートを変更するようにモジュールに指示します。プリセットビットレートはいくつかありますが、すべての有効な CAN ビットレートを使用できます。

Parameters (パラメーター) : 目的のビットレートを指定するには、2つのパラメーター (2 バイト) が必要です。次の表に、頻繁に使用されるビットレートとそのパラメーター表記 P1-P2 を示します。

Bit rate	Hex value	P1	P2
1 Mbit/s	0x0014	0x14	0x00
500 kbit/s	0x001C	0x1C	0x00
250 kbit/s	0x011C	0x1C	0x01
125 kbit/s	0x031C	0x1C	0x03
100 kbit/s	0x432F	0x2F	0x43
50 kbit/s	0x472F	0x2F	0x47
20 kbit/s	0x532F	0x2F	0x53
10 kbit/s	0x672F	0x2F	0x67
5 kbit/s	0x7F7F	0x7F	0x7F

送受信するデータは LSB から始まります。例 : 0x12C0 = P1 : 0xC0 および P2 : 0x12

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	0	0	1	1	1	1	1
0	0, 0x0			2, 0x2		1, 0x1		31, 0x1F					
2, 0x2					95, 0x5F								

<b>Parameters</b>	この例では、CAN インターフェイスのデフォルトのビットレートが 500 kbit / s (Btr0Btr1 : 0x001C) に設定されています。必要なパラメーターの数は 2 です。
<b>Checksum</b>	PX = P1(0x1C) XOR P2(0x00) = 0x1C SC(0x02) XOR CC(0x5F) XOR PX(0x1C) = 0x41
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x5F) – P1(0x1C) – P2(0x00) – CHK(0x41)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメータがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：**設定されたビットレートは、使用されているビットレートとは異なります。使用されるビットレートは、CAN インターフェイスが決められた時間内に CAN バスと対話するために使用する速度です。コンフィグレーション済みビットレートは、ビットレート・コンフィグレーション・メッセージを送信してビットレートを変更したが、初期化していない結果です。新しく設定された値をアクティブにするには、21ページのセクション 2.3.1 初期化およびライト・コマンド (Write Commands) を参照してください。

### フォワード・マスク (Forward Mask) を設定

このコマンドは、モジュール内の CAN インターフェイスのフォワード・マスクを設定します。このマスクは、受信したメッセージが転送される他のインターフェイスと、エラー情報が監視されているかどうかをインターフェイスに通知します。エラーは RS-232 インターフェイスにリダイレクトできます。

フォワード・マスク値は、次の表の値のビットの組み合わせ (ビット単位の OR 演算) です。

Mask value	Meaning
0x80	エラー情報を RS-232 インターフェイスに転送
0x40	エラー情報を CAN インターフェイスに転送
0x04	CAN メッセージを RS-232 インターフェイスに転送
0x02	CAN メッセージを LIN インターフェイスに転送

例えば、0x86 のフォワード・マスクは、CAN インターフェイスのエラー情報が RS-232 インターフェイスに送信され、受信した CAN メッセージが LIN と RS-232 インターフェイスの両方に転送されることを意味します。



**注：**エラー情報を RS-232 インターフェイス (マスク値 0x80) に転送するには、対応するメッセージ転送 (0x04) もアクティブにする必要があります。

受信するデータは LSB から始まります。例：0x011C = P1：0x1C および P2：0x01

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0, 0x0			1, 0x1		1, 0x1		32, 0x20						
1, 0x1					96, 0x60									

<b>Parameters</b>	1つのパラメーター（1バイト）が必要です。この例では、マスクを使用してエラーを監視し、エラーがある場合はRS-232 インターフェイスに転送します。送信する値は 0x80 です（エラー情報を RS-232 インターフェイスに転送します）。
<b>Checksum</b>	P1 = 0x80 SC(0x01) XOR CC(0x60) XOR P1(0x80) = 0xE1
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0x60) – P1(0x80) – CHK(0xE1)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第3章 エラー・メッセージを参照してください。

### フィルタ・マスク (Filter Mask) を設定

このコマンドは、モジュール内の CAN インターフェイスのアクセプタンス・マスクを設定します。CAN アクセプタンス・マスクは、アクセプタンス・コードとともに、受信したメッセージを CAN インターフェイスに通知するために使用されるフィルタです。このフィルタは、CAN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例：0x221100FF = P1：0xFF、P2：0x00、P3：0x11、および P4：0x22

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	1	0	0	0	0	0	0	1
0	0, 0x0			4, 0x4		1, 0x1		33, 0x21				
4, 0x4					97, 0x61							

<b>Parameters</b>	4つのパラメーター（4バイト）が必要です。アクセプタンス・マスクは、29ビットの識別子を使用するため、4バイトの長さです。CANフィルタの詳細については、CANコントローラー Philips SJA1000のApplication Note AN97076を参照してください。この例では、0x1FFFFFFFのフィルタ・マスクが使用されます。
<b>Checksum</b>	$PX = P1(0xFF) \text{ XOR } P2(0xFF) \text{ XOR } P3(0xFF) \text{ XOR } P4(0x1F) = 0xE0$ $SC(0x04) \text{ XOR } CC(0x61) \text{ XOR } PX(0xE0) = 0x85$
<b>Message to send</b>	STX(0x02) – SC(0x04) – CC(0x61) – P1(0xFF) – P2(0xFF) – P3(0xFF) – P4(0x1F) – CHK(0x85)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には、エラーがないことを意味する値0のパラメーターが1つあります。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### フィルタ・コード (Filter Code) を設定

このコマンドは、モジュール内のCANインターフェイスのアクセプタンス・コードを設定します。CANアクセプタンス・コードは、アクセプタンス・マスクとともに、受信したメッセージをCANインターフェイスに通知するために使用されるフィルタです。このフィルタは、CANメッセージのIDを参照しています。

受信するデータはLSBから始まります。例：0x221100FF = P1 : 0xFF、P2 : 0x00、P3 : 0x11、および P4 : 0x22

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	1	0	0	1	0	0	0	1	0
0	0, 0x0			4, 0x4		1, 0x1		34, 0x22					
4, 0x4					98, 0x62								

<b>Parameters</b>	4つのパラメーター（4バイト）が必要です。アクセプタンス・コードは29ビットの識別子を使用するため、4バイトの長さです。CANフィルタの詳細については、CANコントローラー Philips SJA1000のApplication Note AN97076を参照してください。この例では、0xABBCDEFのフィルタ・コードが使用されます。
<b>Checksum</b>	$PX = P1(0xEF) \text{ XOR } P2(0xCD) \text{ XOR } P3(0xBC) \text{ XOR } P4(0xAB) = 0x35$ $SC(0x04) \text{ XOR } CC(0x62) \text{ XOR } PX(0x35) = 0x53$
<b>Message to send</b>	STX(0x02) – SC(0x04) – CC(0x62) – P1(0xEF) – P2(0xCD) – P3(0xBC) – P4(0xAB) – CHK(0x53)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には、エラーがないことを意味する値0のパラメーターが1つあります。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### LINメッセージのCAN ID オフセット（CAN ID Offset for LIN Messages）を設定

このコマンドは、モジュール内のLINメッセージのCAN ID オフセットを設定します。LINメッセージのIDの長さは1バイトです。CANメッセージをLINインターフェイスに正常に転送するには、IDオフセットが必要です。

受信するデータはLSBから始まります。例：0x80110022 = P1：0x22、P2：0x00、P3：0x11、およびP4：0x80

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	1	0	0	0	1	0	0	0	1	1
0	0, 0x0			4, 0x4		1, 0x1		35, 0x23						
4, 0x4					99, 0x63									

<b>Parameters</b>	4つのパラメーター（4バイト）が必要です。CAN ID オフセットの長さは4バイトです（以下の追加の表を参照）。 この例では、0x00110022のIDマスクが使用されます。
<b>Checksum</b>	$PX = P1(0x22) \text{ XOR } P2(0x00) \text{ XOR } P3(0x11) \text{ XOR } P4(0x80) = 0xB3$ $SC(0x04) \text{ XOR } CC(0x63) \text{ XOR } PX(0xB3) = 0xD4$
<b>Message to send</b>	STX(0x02) – SC(0x04) – CC(0x63) – P1(0x22) – P2(0x00) – P3(0x11) – P4(0x80) – CHK(0xD4)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には、エラーがないことを意味する値0のパラメーターが1つあります。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

Param(s)	Description
P1	CAN ID オフセットの最下位バイト
P2	CAN ID オフセットの2番目のバイト（11ビットCAN IDの場合は3ビットのみ、その他のビット=0）
P3	CAN ID オフセットの3番目のバイト（29ビットのCAN IDのみ、それ以外の場合は0に設定）
P4	Bit 0~4 : CAN ID オフセットの最上位部分（29ビットCAN IDのみ、それ以外は0に設定） Bit 5 : 未使用（0に設定） Bit 6 : 未使用（0に設定） Bit 7 : extended CAN ID のフラグ（0 = 11ビット、1 = 29ビット）

### 2.3.4 リセット/消去 コマンド (Reset / Erase Commands)

CAN インターフェイスには、リセットまたは消去コマンドはありません。

## 2.4 LIN インターフェイス (LIN Interface)

### 2.4.1 初期化およびライト・コマンド (Initialization and Write Commands)

Command	on page
現在のパラメーターに合わせて初期化する	37
バスにデータをライト (Write Data on Bus)	37

#### 現在のパラメーターに合わせて初期化する

このコマンドは、設定されたパラメーターを使用して LIN インターフェイスを初期化します。

Sequence Code (SC)					Control Code (CC)						
X	Sequence			Param Byte Count		Interface		Command Code			
0	0	0	0	0	0	0	0	0	0	0	0
0	0, 0x0			0, 0x0		2, 0x2		0, 0x0			
0, 0x0					128, 0x80						

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x80) XOR PX(0x00) = 0x80
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x80) – CHK(0x80)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には、エラーがないことを意味する値 0 のパラメーターが 1 つあります。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

#### バスにデータをライト (Write Data on Bus)

コマンド “Write Data” は、より複雑な構造を使用するため、個別に扱われます。セクション メッセージの送信と受信 > 2.6.2 LIN (98 ページ) を参照してください。

## 2.4.2 リード・コマンド (Read commands)

Command	on page
アクティベーション・ステータス (Activation Status) を取得	38
ビットレート (Bit Rate) を取得	39
フォワード・マスク (Forward Mask) を取得	40
フィルタ・マスク (Filter Mask) を取得	41
フィルタ・コード (Filter Code) を取得	42
スレーブ・マスク・ステータス (Slave Mask Status) を取得	42
マスター・ステータスを (Master Status) 取得	45
LINバス・ターミネーション (マスター・ターミネーション) を取得	45
LIN Bus Termination (Master Termination)	
メッセージ送信タイムアウト (Message Transmission Timeouts) を取得	46
メッセージの再試行 (Message Retries) を取得	47
スケジューラー・エントリ・カウント数 (Scheduler Entries Count) を取得	47
スケジューラー・エントリ (Scheduler Entry) を取得	48
RCV エラー・カウンター (RCV Error Counter) を取得	49
XMT エラー・カウンター (XMT Error Counter) を取得	50
すべてのエラー・カウンター (All Error Counters) を取得	50
スレーブ ID + データ・コンフィグレーション情報を取得 (Slave ID + Data Configuration Information)	51
同期フィールド・ステータスのビットレート認識を取得 (Bit Rate Recognition for Synchronization Field Status)	53
フレーム長 (Frame Length) を取得	53
チェックサム・タイプ (Checksum Type) を取得	55
スレーブ・アクティベーションの CAN ID (CAN ID for Slave Activation) を取得	57

## アクティベーション・ステータスを取得

このコマンドは、モジュールの LIN 機能がアクティブであるかどうかを示す値を取得します。そうでない場合でも、インターフェイスの構成は可能ですが、イベントを処理するために LIN 機能が必要な場合は効果がありません。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count			Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
0	0, 0x0			0, 0x0			2, 0x2		10, 0xA						
0, 0x0				138, 0x8A											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x8A) XOR PX(0x00) = 0x8A
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x8A) – CHK(0x8A)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x01) – P2(0x00) – CHK(0x03)

肯定的な応答には 2 つのパラメーターがあります。P1 と P2（この場合は 0x01）によって形成される値が 0 より大きい場合、これは “interface is active” であることを意味し、そうでない場合は “interface is not active” ではない” ことを意味します。

### ビットレート (Bit Rate) を取得

このコマンドは、LIN インターフェイスの設定済みビットレートを取得します。

受信するデータは LSB から始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
0	0, 0x0			0, 0x0				2, 0x2		11, 0xB					
0, 0x0				139, 0x8B											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x8B) XOR PX(0x00) = 0x8B
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x8B) – CHK(0x8B)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x4B) – CHK(0x49)

肯定応答には、構成されたビットレートを持つ 2 つのパラメーターがあります。この例では、デフォルトのビットレートが P1 と P2 で返されます：0x4B00 = 19200 ビット/秒



**注：** 設定されたビットレートは、使用されているビットレートとは異なります。使用されるビットレートは、決められた時間内に LIN バスと対話するために LIN インターフェイスによって使用される速度です。コンフィグレーション済みビットレートは、ビットレート・コンフィグレーション・メッセージを送信してビットレートを変更したが、初期化していない結果です。新しく設定された値をアクティブにするには、セクション 2.4.1 初期化およびライト・コマンド (Write Commands) を参照してください。

### フォワード・マスク (Forward Mask) の取得

このコマンドは、モジュール内の LIN インターフェイスのフォワード・マスクを設定します。このマスクは、受信したメッセージが転送される他のインターフェイスと、エラー情報が監視されているかどうかをインターフェイスに通知します。

エラーは RS-232 および CAN インターフェイスにリダイレクトできます。フォワード・マスク値は、次の表の値のビットの組み合わせ（ビット単位の OR 演算）です。

Mask value	Meaning
0x80	エラー情報を RS-232 インターフェイスに転送する
0x40	エラー情報を CAN インターフェイスに転送する
0x04	LIN メッセージを RS-232 インターフェイスに転送する
0x01	LIN メッセージを CAN スレーブ・マスク・ステータス・インターフェイスに転送する

例えば、0xC5 のフォワード・マスクは、エラー情報と LIN インターフェイスのメッセージが CAN インターフェイスと RS-232 インターフェイスの両方に転送されることを意味します。

受信するデータは LSB から始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	0	0	0	0	1	1	0	0
0	0, 0x0			0, 0x0		2, 0x2		12, 0xC				
0, 0x0					140, 0x8C							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x8C) XOR PX(0x00) = 0x8C
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x8C) – CHK(0x8C)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。この例では、戻り値はデフォルト値です（転送なし）。値が 0 と異なる場合は、マスク・テーブルの値を使用してビット単位の AND 演算を実行することにより、どの種類の転送が構成されているかを判断できます。例えば、戻り値が 0xC5 の場合、AND 演算は次のようになります。

- (0xC5 AND 0x80 = 0x80) の場合、エラー情報を RS-232 インターフェイスに転送します
- (0xC5 AND 0x40 = 0x40) の場合、エラー情報を CAN インターフェイスに転送します
- (0xC5 AND 0x04 = 0x04) の場合、LIN メッセージを RS-232 インターフェイスに転送します
- (0xC5 AND 0x02 = 0x02) の場合、LIN メッセージを CAN インターフェイスに転送します

### フィルタ・マスク (Filter Mask) を取得

このコマンドは、モジュール内の LIN インターフェイスの設定済みアクセプタンス・マスクを取得します。LIN アクセプタンス・マスクは、アクセプタンス・コードとともに、受信したメッセージを LIN インターフェイスに通知するために使用されるフィルタです。このフィルタは、LIN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	0	0	0	0	1	1	0	1
0	0, 0x0			0, 0x0		2, 0x2		13, 0xD				
0, 0x0					141, 0x8D							

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x8D) XOR PX(0x00) = 0x8D
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x8D) – CHK(0x8D)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0xFF) – P2(0x00) – CHK(0xFD)

肯定的な応答には 2 つのパラメーターがあります。この例では、戻り値はフィルタ・マスク (0xFF) のデフォルト値です。

### フィルタ・コード (Filter Code) を取得

このコマンドは、モジュール内の LIN インターフェイスの構成済みアクセプタンス・コードを取得します。LIN アクセプタンス・コードは、アクセプタンス・マスクとともに、受信したメッセージを LIN インターフェイスに通知するために使用されるフィルタです。このフィルタは、LIN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例：0x011C = P1：0x1C および P2：0x01

Sequence Code (SC)								Control Code (CC)									
X	Sequence				Param Byte Count				Interface				Command Code				
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
0	0, 0x0				0, 0x0				2, 0x2				14, 0xE				
0, 0x0								142, 0x8E									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x8E) XOR PX(0x00) = 0x8E
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x8E) – CHK(0x8E)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。この例では、戻り値はフィルタ・コード (0x00) のデフォルト値です。

### スレーブ・マスク・ステータス (Slave Mask Status) の取得

このコマンドは、1 つまたはすべての LIN メッセージ ID の現在のスレーブ・マスク・ステータス (Publisher : パブリッシャーまたは Subscriber : サブスクライバー) を取得します。このコマンドには 2 つのパラメーター (2 バイト) が必要です。最初のパラメーター (P1) には、0x00 または 0x01 の 2 つの可能な値があります。0 を使用すると、すべての ID のスレーブ・マスク・ステータスが返されます。この場合、2 番目のパラメーターは重要ではありませんが、送信する必要があります。1 を指定すると、2 番目のパラメーター (P2) が有効な LIN ID を示します。この LIN ID の場合、スレーブ・マスク・ステータスが返されます。

P1	P2	Description
0	無視されます	64 個の ID すべて (0x00~0x3F) のスレーブ・マスク・ステータスを取得します
1	LIN ID (0x00...0x3F)	パラメーターP2 で渡された LIN ID のスレーブ・マスク・ステータスを取得します

受信するデータは LSB から始まります。 例：0x011C = P1：0x1C および P2：0x01

ケース A：すべての LIN ID のスレーブ・マスク・ステータスの要求 (P1 = 0)

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	0	0	0	1	1	1	1
0	0, 0x0			2, 0x2		2, 0x2		15, 0xF					
2, 0x2					143, 0x8F								

<b>Parameters</b>	2つのパラメーター (2 バイト) が必要です。最初のパラメーターは、操作をすべての ID (0x00) に対して実行する必要があることを示し、2番目のパラメーターは使用されませんが、送信する必要もあります。
<b>Checksum</b>	PX = P1(0x00) XOR P2(0x00) = 0x00 SC(0x02) XOR CC(0x8F) XOR PX(0x00) = 0x8D
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x8F) – P1(0x00) – P2(0x00) – CHK(0x8D)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5 (0x00) – P6(0x00) – P7(0x00) – P8(0x00) – CHK(0x02)

肯定応答には 8 つのパラメーターがあります。この例では、戻り値はすべてのスレーブ・マスク ID が Subscriber (サブスクライバー) であるスレーブ・マスクのデフォルト値です。

ケース B：単一 ID のスレーブ・マスク・ステータスの要求 (P1 = 1)

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	0	0	0	1	1	1	1
0	0, 0x0			2, 0x2		2, 0x2		15, 0xF					
2, 0x2					143, 0x8F								

<b>Parameters</b>	2つのパラメーター（2バイト）が必要です。最初のパラメーターは、操作を単一のID（0x01）に対して実行する必要があることを示し、2番目のパラメーターは対応するIDを示します。この例では、これはID 57（0x39）です。
<b>Checksum</b>	PX = P1(0x01) XOR P2(0x39) = 0x38 SC(0x02) XOR CC(0x8F) XOR PX(0x38) = 0xB5
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x8F) – P1(0x01) – P2(0x39) – CHK(0xB5)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x39) – P2(0x00) – CHK(0x3B)

肯定的な応答には2つのパラメーターがあります。この例では、戻り値は、Subscriber（サブスクライバー）であるID 57（0x39）のスレーブ・マスクのデフォルト値です。P1には要求されたIDが含まれ、P2には関連する値が含まれます。

マスク・ステータスはブール値として扱われます。戻り値が1の場合、IDはPublisher（パブリッシャー）であり、0の場合、Subscriber（サブスクライバー）です。すべてのIDのスレーブ・マスク・ステータスは8データ・バイトで返され、各ステータスは1ビットで配信されます。

Data Bytes	ID/Bit Assignment							
	7	6	5	4	3	2	1	0
P1 (IDs 0...7)	0x07	0x06	0x05	0x04	0x03	0x02	0x01	0x00
P2 (IDs 8...15)	0x0F	0x0E	0x0D	0x0C	0x0B	0x0A	0x09	0x08
P3 (IDs 16...23)	0x17	0x16	0x15	0x14	0x13	0x12	0x11	0x10
P4 (IDs 24...31)	0x1F	0x1E	0x1D	0x1C	0x1B	0x1A	0x19	0x18
P5 (IDs 32...39)	0x27	0x26	0x25	0x24	0x23	0x22	0x21	0x20
P6 (IDs 40...47)	0x2F	0x2E	0x2D	0x2C	0x2B	0x2A	0x29	0x28
P7 (IDs 48...55)	0x37	0x36	0x35	0x34	0x33	0x32	0x31	0x30
P8 (IDs 56...63)	0x3F	0x3E	0x3D	0x3C	0x3B	0x3A	0x39	0x38

たとえば、値が0x82（0b10000010）のP3を受信した場合、これはID 16（0x10）、18（0x12）、19（0x13）、20（0x14）、21（0x15）、および22（0x16）のスレーブ・マスク・ステータスがSubscriber（サブスクライバー）であり、ID 17（0x11）および23（0x17）のスレーブ・マスク・ステータスがPublisher（パブリッシャー）であることを意味します。

### マスター・ステータス (Master Status) の取得

このコマンドは、モジュールの LIN インターフェイスのマスター・ステータス (active : アクティブ / inactive : 非アクティブ) を示す値を取得します。ステータス “active” は、マスター・スケジューラーにエントリーがあり、それらが送信されることを意味します。

受信するデータは LSB から始まります。例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
0	0, 0x0			0, 0x0				2, 0x2		16, 0x10					
0, 0x0							144, 0x90								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x90) XOR PX(0x00) = 0x90
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x90) – CHK(0x90)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。P1 と P2 (この場合は 0x00) によって形成される値が 0 に等しい場合、LIN マスターは inactive (非アクティブ) です。それ以外の場合、LIN マスターは active (アクティブ) です。

### LIN バス・ターミネーション (Master Termination) の取得

このコマンドは、モジュールが master (マスター) または slave (スレーブ) として構成されているかどうかを示す値を取得します (いわゆる Termination : ターミネーション)。モジュールのデフォルトのターミネーションは “Slave” です。

受信するデータは LSB から始まります。例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1
0	0, 0x0			0, 0x0				2, 0x2		17, 0x11					
0, 0x0							145, 0x91								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x91) XOR PX(0x00) = 0x91
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x91) – CHK(0x91)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。P1 と P2（この場合は 0x00）によって形成される値が 0 に等しい場合、これは “slave termination” を意味し、そうでない場合は “master termination” を意味します。

**メッセージ送信タイムアウト (Message Transmission Timeouts) の取得**

このコマンドは、メッセージ送信のタイムアウト (ms 単位) を示すいくつかの値を取得します。タイムアウトは、バス・アイドル・メッセージ、フレーム長 8、フレーム長 4、フレーム長 2 のメッセージ用です。このコマンドへの応答は、タイムアウトをこの順序で表します。各タイムアウトの長さは 2 バイトです。

受信するデータは LSB から始まります。例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	0	0	0	1	0	0	1	0
0	0, 0x0			0, 0x0		2, 0x2		18, 0x12					
0, 0x0					146, 0x92								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x92) XOR PX(0x00) = 0x92
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x92) – CHK(0x92)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x16) – P2(0x05) – P3(0x0A) – P4(0x00) – P5(0x07) – P6(0x00) – P7(0x05) – P8(0x00) – CHK(0x13)

肯定的な反応の説明：

Param(s)	Description	Example value	Meaning
P1, P2	バス・アイドル	0x0516 (デフォルト)	1302 ms
P3, P4	フレーム長 8	0x000A (デフォルト)	10 ms
P5, P6	フレーム長 4	0x0007 (デフォルト)	7 ms
P7, P8	フレーム長 2	0x0005 (デフォルト)	5 ms

### メッセージの再試行 (Message Retries) を取得

このコマンドは、“slave not responding” (スレーブが応答していません) タイムアウトを引き起こす LIN 応答メッセージの再試行回数を示す値を取得します (マスター・ステータスが inactive : 非アクティブの場合のみ)。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
0	0, 0x0			0, 0x0				2, 0x2		19, 0x13					
0, 0x0				147, 0x93											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x93) XOR PX(0x00) = 0x93
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x93) – CHK(0x93)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。この例では、戻り値はメッセージ再試行のデフォルト値 0 です。

### スケジューラー・エントリー・カウント数 (Scheduler Entries Count) の取得

このコマンドは、モジュール内のマスター・スケジューラーの設定済みの有効なエントリーの数を示す値を取得します。スケジューラー・エントリーの最大数は 50 です。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)								Control Code (CC)								
X	Sequence				Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0
0	0, 0x0				0, 0x0				2, 0x2		20, 0x14					
0, 0x0								148, 0x94								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x94) XOR PX(0x00) = 0x94
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x94) – CHK(0x94)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定的な応答には 2 つのパラメーターがあります。この例では、デフォルトでスケジューラーが空であるため、返される “count of entries” (エントリー数) はデフォルト値 0 です。

### スケジューラー・エントリー (Scheduler Entry) の取得

このコマンドは、モジュール内のマスター・スケジューラーから有効なエントリーの構成を示すいくつかの値を取得します。スケジューラー・エントリーは、LIN メッセージ ID とそれに対応するタイムスタンプの組み合わせです。タイムスタンプは、開始から測定された 2 つのフレーム間の時間差です (各フレームからの “sync break” 信号のトレーディングエッジ)。スケジューラー・リストから返されるエントリーは、このコマンドのパラメーターとして指定されます。このパラメーターは、スケジューラー・リストのエントリーの索引に対応しています。最初に可能なスケジューラー・エントリー・インデックスは 1 で、最後は 50 です。

受信するデータは LSB から始まります。例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)								Control Code (CC)								
X	Sequence				Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1
0	0, 0x0				1, 0x1				2, 0x2		21, 0x15					
1, 0x1								149, 0x95								

<b>Parameters</b>	返されるスケジューラ・エントリーのインデックスを示すには、1つのパラメータ（1バイト）が必要です。この例では、スケジューラに次の構成のエントリーがあると想定しています。ID：0x10、タイムスタンプ：50 ミリ秒
<b>Checksum</b>	PX = P1(0x01) = 0x01 SC(0x01) XOR CC(0x95) XOR PX(0x01) = 0x95
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0x95) – P1(0x01) – CHK(0x95)
<b>Positive response</b>	STX(0x02) – SC(0x03) – P1(0x32) – P2(0x00) – P3(0x10) – CHK(0x21)

肯定的な反応の説明：

Param(s)	Description	Example value	Meaning
P1, P2	Timestamp	0x0032	50 ms
P3	LIN ID	0x10	LIN ID 0x10

### RCV エラー・カウンター（RCV Error Counter）を取得

このコマンドは、LIN インターフェイスの受信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります 例：0x1 = P1：0x01 および P2：0x00

Sequence Code (SC)				Control Code (CC)										
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	0	1	0	1	1	1	0
0	0, 0x0			0, 0x0			2, 0x2		22, 0x16					
0, 0x0				150, 0x96										

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x96) XOR PX(0x00) = 0x96
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x96) – CHK(0x96)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、XMT エラー・カウンターの値を持つ 2 つのパラメータがあります。この例では、デフォルト値は P1 および P2：0x0 = 0 のエラーで返されます

**XMT エラー・カウンター (XMT Error Counter) を取得**

このコマンドは、LIN インターフェイスの送信エラー・カウンターの値を取得します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1
0	0, 0x0			0, 0x0				2, 0x2		23, 0x17					
0, 0x0				151, 0x97											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x97) XOR PX(0x00) = 0x97
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x97) – CHK(0x97)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x00) – P2(0x00) – CHK(0x02)

肯定応答には、XMT エラー・カウンターの値を持つ 2 つのパラメーターがあります。この例では、デフォルト値は P1 および P2 : 0x0 = 0 のエラーで返されます

**すべてのエラー・カウンター (All Error Counters) を取得**

このコマンドは、LIN インターフェイスのエラー・カウンターの値を取得します。戻り値は、RCV エラー・カウンター、XMT エラー・カウンター、ビットエラー・カウンター、チェックサム・エラー・カウンター、ID チェックサム・エラー・カウンター、“Slave not responding” エラー・カウンター、“Inconsistent synchronization bytes” エラー・カウンターから（この順序で）返されます。各値の長さは 2 バイトです。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)								Control Code (CC)									
X	Sequence			Param Byte Count				Interface		Command Code							
0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0
0	0, 0x0			0, 0x0				2, 0x2		24, 0x18							
0, 0x0								152, 0x98									

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x98) XOR PX(0x00) = 0x98
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x98) – CHK(0x98)
<b>Positive response</b>	STX(0x02) – SC(0x0E) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0x00) – P7(0x00) – P8(0x00) – P9(0x00) – P10(0x00) – P11(0x00) – P12(0x00) – P13(0x00) – P14(0x00) – CHK(0x0E)

肯定的な反応の説明 :

Param(s)	Description	Example value	Meaning
P1, P2	RCV エラー・カウンター (RCV error counter)	0x0 (デフォルト)	0 エラー
P3, P4	XMT エラー・カウンター (XMT error counter)	0x0 (デフォルト)	0 エラー
P5, P6	ビットエラー・カウンター (Bit error counter)	0x0 (デフォルト)	0 エラー
P7, P8	チェックサム・エラー・カウンター (Checksum error counter)	0x0 (デフォルト)	0 エラー
P9, P10	ID チェックサム・エラー・カウンター (ID Checksum error counter)	0x0 (デフォルト)	0 エラー
P11, P12	“スレーブが応答しない”エラー・カウンター (“Slave not responding” error counter)	0x0 (デフォルト)	0 エラー
P13, P14	“一貫性のない同期バイト”エラー・カウンター (“Inconsistent synchronization bytes” error counter)	0x0 (デフォルト)	0 エラー

### スレーブ ID + データ・コンフィグレーション (Slave ID + Data Configuration) 情報の取得

このコマンドは、モジュールのスレーブ構成に関する情報を取得します。

このコマンドへの応答として、次の情報が受信されます :

“Slave ID + Data Configuration” フィールドのステータス (enabled か disabled)、スレーブを識別する ID、およびそのスレーブ ID に関連付けられたコンフィグレーション・データ。データの長さはスレーブ ID によって異なります。

LIN ID の長さは 6 ビット (0x00...0x3F) で、データ長は次のとおりです :

ID range	Data length
0x00 から 0x1F	2 バイト
0x20 から 0x2F	4 バイト
0x30 から 0x3F	8 バイト

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)						Control Code (CC)								
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0, 0x0			0, 0x0			2, 0x2		25, 0x19					
0, 0x0						153, 0x99								

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x99) XOR PX(0x00) = 0x99
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x99) – CHK(0x99)
<b>Positive response</b>	STX(0x02) – SC(0x0A) – P1(0x00) – P2(0x3F) – P3(0xFF) – P4(0xFF) – P5(0xFF) – P6(0xFF) – P7(0xFF) – P8(0xFF) – P9(0xFF) – 10(0xFF) – CHK(0x35)

肯定的な反応の説明：

Param(s)	Description	Example value	Meaning
P1	フィールド・ステータス (0 = disabled, 1 = enabled)	0x00 (デフォルト)	disabled
P2	Slave ID	0x3F	データ長は 8 バイトになります
P3...P10	コンフィグレーション・データ (バイト 1 から 8)	各 0xFF (デフォルト)	

この例では、ID の長さが、上記の “ID range / Data length” の表に記載されているデフォルトの長さに相関していると想定しています。モジュールの新機能により、任意の ID に任意の長さを設定できます。これにより、前述の表の規則に従わずに、さまざまな長さの肯定応答が発生します。

### 同期フィールド・ステータスのビットレート認識 (Bit Rate Recognition for Synchronization Field Status) を取得

このコマンドは、ビットレート認識がアクティブかどうかを示す値を取得します。自動ビットレート認識は、受信したフレームで、同期の中断が正しく認識された場合にのみ機能します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count		Interface		Command Code							
0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0
0	0, 0x0			0, 0x0		2, 0x2		26, 0x1A							
0, 0x0				154, 0x9A											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x00) XOR CC(0x9A) XOR PX(0x00) = 0x9A
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x9A) – CHK(0x9A)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x01) – P2(0x00) – CHK(0x03)

肯定的な応答には 2 つのパラメーターがあります。この例では、ビットレート認識フィールド 0x01 (active) のデフォルト値が返されます。

### フレーム長 (Frame Length) の取得

このコマンドは、64 個の LIN メッセージ ID の 1 つまたはすべてに割り当てられたフレーム長を取得します。コマンド Retrieve Slave ID + Data Configuration Information (51 ページ) に示されているように、LIN メッセージのフレーム長は ID によって異なります。モジュールの機能の拡張により、ID に関係なく、任意の ID のフレーム長を 0~8 バイトに設定できます (フレーム長のデフォルト値は、前述の ID の規則に従います)。このため、すべての ID のフレーム長または単一の ID のフレーム長を取得できます。

このコマンドには 2 つのパラメーターがあります。必要なデータに応じて、パラメーターは次のとおりです。

P1	P2	Description
0	使用されない、無視される	ID 0x00~0x0F のフレーム長を取得します
1	使用されない、無視される	ID 0x10~0x1F のフレーム長を取得します
2	使用されない、無視される	ID 0x20~0x2F のフレーム長を取得します
3	使用されない、無視される	ID 0x30~0x3F のフレーム長を取得します
4	LIN ID (0x00...0x3F)	パラメーターP2 によって渡された ID のフレーム長を取得します

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

**Case A** : 一連のフレーム長の取得 (0 から 3 の範囲内の P1)

Sequence Code (SC)							Control Code (CC)							
X	Sequence			Param Byte Count			Interface		Command Code					
0	0	0	0	0	0	1	0	1	0	1	1	0	1	1
0	0, 0x0			2, 0x2			2, 0x2		27, 0x1B					
0, 0x0							155, 0x9B							

<b>Parameters</b>	このコマンドには2つのパラメーター (2 バイト) が必要です。最初のパラメーターは、ID の範囲を示します。2 番目のパラメーターは使用されませんが、送信する必要があります。この例では、最後の 16 個の ID (0x30 から 0x3F) のフレーム長が取得されます。
<b>Checksum</b>	PX = P1(0x03) XOR P2(0x00) = 0x03 SC(0x02) XOR CC(0x9B) XOR PX(0x03) = 0x9A
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x9B) – P1(0x03) – P2(0x00) – CHK(0x9A)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x88) – P2(0x88) – P3(0x88) – P4(0x88) – P5(0x88) – P6(0x88) – P7(0x88) – P8(0x88) – CHK(0x08)

肯定応答には 8 つのパラメーターがあります。この例では、ID 0x30~0x3F のフレーム長のデフォルト値が返されます。各データ・バイトには、2 つの ID の長さが含まれます (順序 : 最下位 4 ビット、最上位 4 ビット)。たとえば、0x84 は、1 つの ID (0x16) が 4 バイトで、次の ID (0x17) が 8 バイトの長さであることを意味します。

Param(s)	Description	Example value	Meaning
P1...P8	特定の ID のフレームの長さ	0x88	フレーム毎に 8 バイト

**Case B** : 単一の ID のフレーム長を取得する (P1 = 4)

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	1	0	0	1	1	0	1	1
0	0, 0x0			2, 0x2		2, 0x2		27, 0x1B						
0, 0x0					155, 0x9B									

<b>Parameters</b>	このコマンドには 2 つのパラメーター (2 バイト) が必要です。最初のパラメーターは、2 番目のパラメーターに ID が含まれていることを示します。この ID のフレーム長が返されます。この例では、ID 0x07 のフレーム長が取得されます。
<b>Checksum</b>	PX = P1(0x04) XOR P2(0x07) = 0x03 SC(0x02) XOR CC(0x9B) XOR PX(0x03) = 0x9A
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x9B) – P1(0x04) – P2(0x07) – CHK(0x9A)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x07) – P2(0x02) – CHK(0x07)

肯定的な応答には 2 つのパラメーターがあります。この例では、ID 0x07 のフレーム長のデフォルト値が返されます。最初のデータ・バイトには ID が含まれ、2 番目のデータ・バイトにはフレーム長が含まれます。

Param(s)	Description	Example value	Meaning
P1	LIN ID	0x07	ID 0x07
P2	フレームの長さ	0x02 (default)	2 bytes

### チェックサム・タイプ (Checksum Type) の取得

このコマンドは、単一または、すべての 64 個の LIN メッセージ ID の現在のチェックサム・タイプ (Checksum Type) を取得します。これは、クラシック・タイプ (LIN 1.x) または拡張タイプ (LIN 2.0) のいずれかです。このコマンドには 2 つのパラメーターがあります。

必要なデータに応じて、パラメーターは次のようになります。

P1	P2	Description
0	使用されない、無視される	64 個の LIN ID すべて (0x00 から 0x3F) のチェックサム・タイプ (Checksum Type) を取得します。
1	LIN ID (0x0...0x3F)	パラメーターP2 として渡された LIN ID のチェックサム・タイプ (Checksum Type) を取得します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

**Case A : すべてのチェックサム・タイプの取得 (P1 = 0)**

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0, 0x0			2, 0x2				2, 0x2		28, 0x1C					
2, 0x2							156, 0x9C								

<b>Parameters</b>	このコマンドには 2 つのパラメーター (2 バイト) が必要です。最初のパラメーターは、この場合、すべての ID のチェックサム・タイプ (Checksum Type) が返されることを示します。2 番目のパラメーターは使用されませんが、送信する必要があります。
<b>Checksum</b>	$PX = P1(0x00) \text{ XOR } P2(0x00) = 0x00$ $SC(0x02) \text{ XOR } CC(0x9C) \text{ XOR } PX(0x00) = 0x9E$
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x9C) – P1(0x00) – P2(0x00) – CHK(0x9E)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0x00) – P7(0x00) – P8(0x00) – CHK(0x08)

肯定応答には 8 つのパラメーターがあります。この例では、すべての ID ラシック・チェックサム・タイプ (Classic checksum type) のデフォルト値が返されます。各データ・バイトには、データ・バイト 1 の ID 0x00~0x07 からデータ・バイト 8 の ID 0x38~0x3F までの 8 つの ID のフォーマットが含まれます。ID は最下位ビットから昇順で並べられます。0 は、従来のチェックサム・タイプ (Checksum type) を表します。1 は、拡張チェックサム・タイプ (Checksum type) を表します。

Param(s)	Description	Example value	Meaning
P1...P8	指定された ID のチェックサム (Checksum) のフォーマット (各ビット)	0b00000000 (default)	すべての ID のクラシック・チェックサム・タイプ (Classic checksum type)

**Case B : 単一 ID のチェックサム・タイプ (Checksum type) の取得 (P1 = 1)**

Sequence Code (SC)				Control Code (CC)										
X	Sequence			Param Byte Count				Interface		Command Code				
0	0	0	0	0	0	1	0	1	0	1	1	1	0	0
0	0, 0x0			2, 0x2				2, 0x2		28, 0x1C				
2, 0x2				156, 0x9C										

<b>Parameters</b>	このコマンドには2つのパラメーター (2バイト) が必要です。最初のパラメーターは、この場合、特定のIDのチェックサム・タイプ (Checksum type) が返されることを示します。2番目のパラメーターにはLIN IDが含まれます。この例では、ID 0x07 のチェックサム・タイプ (Checksum type) が取得されます。
<b>Checksum</b>	$PX = P1(0x01) \text{ XOR } P2(0x07) = 0x06$ $SC(0x02) \text{ XOR } CC(0x9C) \text{ XOR } PX(0x06) = 0x98$
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x9C) – P1(0x01) – P2(0x07) – CHK(0x98)
<b>Positive response</b>	STX(0x02) – SC(0x02) – P1(0x07) – P2(0x00) – CHK(0x05)

肯定的な反応の説明 :

Param(s)	Description	Example value	Meaning
P1	LIN ID	0x07	LIN ID 0x07
P2	チェックサム・タイプ (Checksum type)	0x00 (default)	クラシック・チェックサム・タイプ (Classic checksum type)

**スレーブ・アクティベーションの CAN ID (CAN ID for Slave Activation) を取得**

このコマンドは、CAN メッセージングを介してモジュールを LIN スレーブとしてアクティブ化するために使用される構成済み CAN ID と、CAN ID が 11 ビットまたは 29 ビット長であり、このフィールドのステータスがアクティブであるかどうかを示します。

受信するデータは LSB から始まります。例 : 0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1
0	0, 0x0			0, 0x0				2, 0x2		29, 0x1D					
0, 0x0				157, 0x9D											

<b>Parameters</b>	none
<b>Checksum</b>	SC(0x02) XOR CC(0x9D) XOR PX(0x00) = 0x9D
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0x9D) – CHK(0x9D)
<b>Positive response</b>	STX(0x02) – SC(0x04) – P1(0xF0) – P2(0x07) – P3(0x00) – P4(0x00) – CHK(0xF3)

肯定的な反応の説明：

Param(s)	Description	Example value	Meaning
P1	CAN ID の最下位バイト	0xF0	CAN ID 0x07F0
P2	CAN ID の 2 バイト目 (11 ビット CAN ID の場合は 3 ビットのみ)	0x07	
P3	CAN ID の 3 バイト目 (29 ビット CAN ID のみ)	0x00	
P4	ビット 0~4 : 29 ビット CAN ID の最も重要な部分 ビット 5 : 未使用 ビット 6 : スレーブがアクティブ ビット 7 : 拡張 CAN ID	0x00	スレーブがアクティブではない標準 CAN ID (11 ビット)

### 2.4.3 コンフィグレーション・コマンド (Configuration Commands)

Command	on page
アクティベーション・ステータス (Activation Status) の設定	59
ビットレート (Bit Rate) を設定	60
フォワード・マスク (Forward Mask) を設定	61
フィルタ・マスク (Filter Mask) を設定	62
フィルタ・コード (Filter Code) を設定	63
マスター・ステータス (Master Status) の設定	64
LIN バス・ターミネーションの設定 (マスター・ターミネーション)	65
LIN Bus Termination (Master Termination)	
メッセージ送信タイムアウト (Message Transmission Timeouts) の設定	66
メッセージの再試行 (Message Retries) を設定	67
スケジューラー・エントリー (Scheduler Entry) の挿入	68
スレーブ ID +データ構成 (Slave ID + Data Configuration) の設定	69
同期フィールドのビットレート認識を構成 (Bit Rate Recognition for Synchronization Field)	70
フレーム長 (Frame Length) を設定	71
チェックサム・タイプ (Checksum Type) の設定	74
スレーブ・アクティベーションの CAN ID (CAN ID for Slave Activation) を設定	77
スレーブ・マスク・ステータス (Slave Mask Status) の設定	78

#### アクティベーション・ステータス (Activation Status) の設定

このコマンドは、CAN 機能がアクティブかどうかを示す値をモジュールに設定します。

送受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	0	1	1	1	1	0
0	0, 0x0			1, 0x1		2, 0x2		30, 0x1E						
1, 0x1					158, 0x9E									

<b>Parameters</b>	LIN インターフェイスのアクティブ化ステータスを指定するには、1つのパラメーター（1バイト）が必要です。このパラメーター・バイトはブール値として機能します。パラメーターが0の場合、インターフェイスは非アクティブ化され、そうでない場合はアクティブ化されます。この例では、インターフェイスがアクティブ化されます。
<b>Checksum</b>	P1 = 0x01 SC(0x01) XOR CC(0x9E) XOR PX(0x01) = 0x9E
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0x9E) – P1(0x01) – CHK(0x9E)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：** LIN インターフェイスのアクティブ化には、初期化コマンドを使用した初期化は必要ありません。

### ビットレート（Bit Rate）を設定

このコマンドは、LIN インターフェイスで使用されるビットレートを変更するようにモジュールに指示します。LIN ビットレートの可能な値は、1000～20000 ビット/秒の範囲です。

送受信するデータは LSB から始まります。例：0x12C0 = P1：0xC0 および P2：0x12

Sequence Code (SC)						Control Code (CC)							
X	Sequence			Param Byte Count			Interface		Command Code				
0	0	0	0	0	0	1	0	0	1	1	1	1	1
0	0, 0x0			2, 0x2			2, 0x2		31, 0x1F				
2, 0x2						159, 0x9F							

<b>Parameters</b>	目的のビットレートを指定するには、2つのパラメーター（2バイト）が必要です。この例では、LIN インターフェイスのデフォルトのビットレートである 19200 ビット/秒（0x4B00）が使用されています。
<b>Checksum</b>	PX = P1(0x00) XOR P2(0x4B) = 0x4B SC(0x02) XOR CC(0x9F) XOR PX(0x4B) = 0xD6
<b>Message to send</b>	STX(0x02) – SC(0x02) – CC(0x9F) – P1(0x00) – P2(0x4B) – CHK(0xD6)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第3章 エラー・メッセージを参照してください。



**注：**設定されたビットレートは、使用されたビットレートとは異なります。使用されるビットレートは、決められた時間内にモジュールとの間でメッセージを送受信するために使用される速度です。構成済みビットレートは、ビットレート構成メッセージを送信してビットレートを変更したが、初期化はしなかった結果です。新しく設定された値をアクティブにするには、37 ページのセクション 2.4.1 初期化およびライト・コマンドを参照してください。

### フォワード・マスク（Forward Mask）を設定

このコマンドは、モジュール内の LIN インターフェイスのフォワード・マスクを設定します。このマスクは、受信したメッセージが転送される他のインターフェイスと、エラー情報を監視する必要があるかどうかをインターフェイスに通知します。エラーは RS-232 および/または CAN インターフェイスにリダイレクトすることもできます。

フォワード・マスク値は、次の表の値のビットの組み合わせ（ビット単位の OR 演算）です。

Mask value	Meaning
0x80	エラー情報を RS-232 インターフェイスに転送する。
0x40	エラー情報を CAN インターフェイスに転送する。
0x04	LIN メッセージを RS-232 インターフェイスに転送する。
0x01	LIN メッセージを CAN インターフェイスに転送する。

たとえば、0xC5 のフォワード・マスクは、エラー情報と LIN インターフェイスのメッセージが CAN インターフェイスと RS-232 インターフェイスの両方に転送されることを意味します。



**注:** エラー情報を RS-232 インターフェイス (マスク値 0x80) に転送するには、対応するメッセージ転送 (0x04) もアクティブにする必要があります。

受信するデータは LSB から始まります。例 : 0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0, 0x0			1, 0x1		2, 0x2		32, 0x20						
1, 0x1					160, 0xA0									

<b>Parameters</b>	1つのパラメーター (1バイト) が必要です。この例では、エラーを監視し、エラーがある場合はそれを CAN インターフェイスに転送するためのマスクが使用されます。送信する値は 0x40 (エラー情報を CAN インターフェイスに転送) です。
<b>Checksum</b>	P1 = 0x40 SC(0x01) XOR CC(0xA0) XOR PX(0x40) = 0xE1
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA0) – P1(0x40) – CHK(0xE1)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には 1つのパラメーターがあり、値 0 はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

### フィルタ・マスク (Filter Mask) を設定

このコマンドは、モジュール内の LIN インターフェイスのアクセプタンス・マスクを設定します。LIN アクセプタンス・マスクは、アクセプタンス・コードとともに、受信したメッセージを LIN インターフェイスに通知するために使用されるフィルタです。このフィルタは、LIN メッセージの ID を参照しています。

受信するデータは LSB から始まります。例 : 0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)								Control Code (CC)									
X	Sequence				Param Byte Count				Interface		Command Code						
0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1
0	0, 0x0				1 <sub>(1)</sub> , 0x1 <sub>(16)</sub>				2, 0x2		33, 0x21 <sub>(6)</sub>						
1, 0x1								161, 0xA1									

<b>Parameters</b>	1つのパラメーター（1バイト）が必要です。アクセプタンス・マスクの長さは1バイトです（LINメッセージIDの長さは1バイトです）。この例では、LINフィルタ・マスクのデフォルト値（0xFF）が使用されています。
<b>Checksum</b>	PX = P1(0xFF) = 0xFF SC(0x01) XOR CC(0xA1) XOR PX(0xFF) = 0x5F
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA1) – P1(0xFF) – CHK(0x5F)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### フィルタ・コード (Filter Code) を設定

このコマンドは、モジュール内のLINインターフェイスのアクセプタンス・コードを設定します。LINアクセプタンス・コードは、アクセプタンス・マスクと組み合わせて、受信したメッセージをLINインターフェイスに通知するために使用されるフィルタです。このフィルタは、LINメッセージのIDを参照しています。

受信するデータはLSBから始まります。例：0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)								Control Code (CC)								
X	Sequence				Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0
0	0, 0x0				1, 0x1				1, 0x1		34, 0x22					
1, 0x1								162, 0xA2								

<b>Parameters</b>	1つのパラメーター（1バイト）が必要です。受け入れコードの長さは1バイトです（LINメッセージIDの長さは1バイトです）。この例では、LINフィルタ・コードのデフォルト値（0x00）が使用されています。
<b>Checksum</b>	PX = P1(0x00) = 0x00 SC(0x01) XOR CC(0xA2) XOR PX(0x00) = 0xA3
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA2) – P1(0x00) – CHK(0xA3)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### マスター・ステータス（Master Status）の設定

このコマンドは、モジュール内のLINマスターのステータスを設定します。この機能のアクティブ化は、マスター・スケジューラーにエントリーがあり、それらが送信されることを意味します。スケジューラーが空の場合、マスターをアクティブ化しようとするエラーが返されます。

受信するデータはLSBから始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0, 0x0			1, 0x1		2, 0x2		36, 0x24				
1, 0x1					164, 0xA4							

<b>Parameters</b>	LIN マスターのアクティブ化ステータスを指定するには、1つのパラメーター（1バイト）が必要です。このパラメーター・バイトはブール値として機能します。パラメーターが0の場合、マスターは非アクティブになり、そうでない場合はアクティブになります。この例では、マスターがアクティブであり、マスター・スケジューラーに少なくとも1つの有効なエントリーがあると想定されています。新しいスケジューラー・エントリーをマスター・スケジューラーに挿入する方法の詳細については、コマンド スケジューラー・エントリーの挿入（68 ページ）を参照してください。
<b>Checksum</b>	PX = P1(0x01) = 0x01 SC(0x01) XOR CC(0xA4) XOR PX(0x01) = 0xA4
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA4) – P1(0x01) – CHK(0xA4)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第3章 エラー・メッセージを参照してください。

### LIN バス・ターミネーション (LIN Bus Termination) の設定

このコマンドは、モジュールの終了をマスターまたはスレーブに設定します。モジュールのデフォルトの終端はスレーブです。

受信するデータは LSB から始まります。 例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count			Interface		Command Code						
0	0	0	0	0	0	0	1	1	0	1	0	0	1	0	1
0	0, 0x0			1, 0x1			2, 0x2		37, 0x25						
1, 0x1							165, 0xA5								

<b>Parameters</b>	モジュールの終了を指定するには、1つのパラメーター（1バイト）が必要です。このパラメーター・バイトはブール値として機能します。パラメーターが0の場合、ターミネーションはスレーブ（デフォルト）に設定され、それ以外の場合はマスターに設定されます。この例では、パラメーターとして値 0x01 を使用して、ターミネーションがマスターに設定されています。
<b>Checksum</b>	PX = P1(0x01) = 0x01 SC(0x01) XOR CC(0xA5) XOR PX(0x01) = 0xA5
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA5) – PX(0x01) – CHK(0xA5)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### メッセージ送信タイムアウト（Message Transmission Timeouts）の設定

このコマンドは、さまざまな種類のメッセージの送信のタイムアウト（ミリ秒単位）を設定します。設定するタイムアウトは、バス・アイドル・メッセージとフレーム長 8、4、2 バイトのメッセージ用です。すべてのタイムアウトの長さは2バイトで、8つの必要なパラメーターがあります。パラメーターの順序は次のとおりです：

- P1 – P2 : バス・アイドル
- P3 – P4 : フレーム長 8
- P5 – P6 : フレーム長 4
- P7 – P8 : フレーム長 2

受信するデータは LSB から始まります。 例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	1	0	0	0	1	0	0	1	1	0
0	0, 0x0			8, 0x8		2, 0x2		38, 0x26					
8, 0x8					166, 0xA6								

<b>Parameters</b>	<p>4つのタイムアウトすべてを指定するには、8つのパラメーター（8バイト）が必要です（LSBから始まるタイムアウトごとに2バイト）。この例では、デフォルトのタイムアウトが設定されています。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th style="background-color: #e0e0e0;">Timeout for</th> <th style="background-color: #e0e0e0;">Period</th> <th style="background-color: #e0e0e0;">Parameter Value</th> </tr> </thead> <tbody> <tr> <td>バス・アイドル</td> <td>1302 ms</td> <td>0x0516</td> </tr> <tr> <td>フレーム長 8</td> <td>10 ms</td> <td>0x000A</td> </tr> <tr> <td>フレーム長 4</td> <td>7 ms</td> <td>0x0007</td> </tr> <tr> <td>フレーム長 2</td> <td>5 ms</td> <td>0x0005</td> </tr> </tbody> </table>	Timeout for	Period	Parameter Value	バス・アイドル	1302 ms	0x0516	フレーム長 8	10 ms	0x000A	フレーム長 4	7 ms	0x0007	フレーム長 2	5 ms	0x0005
Timeout for	Period	Parameter Value														
バス・アイドル	1302 ms	0x0516														
フレーム長 8	10 ms	0x000A														
フレーム長 4	7 ms	0x0007														
フレーム長 2	5 ms	0x0005														
<b>Checksum</b>	<p>PX = P1(0x16) XOR P2(0x05) XOR P3(0x0A) XOR P4(0x00)                  XOR P5 (0x07) XOR P6(0x00) XOR P7(0x05) XOR P8(0x00) = 0x1B                  SC(0x08) XOR CC(0xA6) XOR PX(0x1B) = 0xB5</p>															
<b>Message to send</b>	<p>STX(0x02) – SC(0x08) – CC(0xA6) – P1(0x16) – P2(0x05) – P3(0x0A) –                  P4(0x00) – P5(0x07) – P6(0x00) – P7(0x05) – P8(0x00) – CHK(0x5B)</p>															
<b>Positive response</b>	<p>STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)</p>															

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

**メッセージの再試行（Message Retries）を設定**

このコマンドは、“Slave not responding” タイムアウトを引き起こすLIN応答メッセージの再試行回数を設定します（マスター・ステータスが非アクティブの場合のみ）。デフォルト値は0（繰り返しなし）です。

最大値は255回の再試行です。

受信するデータはLSBから始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence	Param Byte Count			Interface	Command Code							
0	0 0 0 0	0	0	0	1	1	0	1	0	0	1	1	1
0	0, 0x0	1, 0x1			2, 0x2	39, 0x27							
1, 0x1					167, 0xA7								

<b>Parameters</b>	LIN 応答メッセージの再試行回数を指定するには、1つのパラメーター (1バイト) が必要です。この例では、このフィールドは最大値である 255 回の再試行 (0xFF) に設定されます。
<b>Checksum</b>	PX = P1(0xFF) = 0xFF SC(0x01) XOR CC(0xA7) XOR PX(0xFF) = 0x59
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xA7) – P1(0xFF) – CHK(0x59)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### スケジューラー・エントリー (Scheduler Entry) の挿入

このコマンドは、モジュールのマスター・スケジューラーに新しいスケジューラー・エントリーを挿入します。スケジューラー・エントリーは、LIN メッセージ ID とそれに対応するタイムスタンプの組み合わせです。タイムスタンプは、開始から測定された2つのフレーム間の時間差 (フレームの “Sync. break” 信号の立ち下がりエッジ)、つまり、フレームが送信される周波数です。このコマンドを使用すると、新しいエントリーをスケジューラーに含めることができます。最大 50 のエントリーが許可されます。既存のエントリーは変更できません。マスター・スケジューラーは、リスト全体を消去して、変更した値を再度入力する必要があります。LIN メッセージ ID の長さは1バイトで、タイムスタンプの長さは2バイトです。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	1	1	0	1	0	0	0
0	0, 0x0			3, 0x3		2, 0x2		40, 0x28					
3, 0x3					168, 0xA8								

<b>Parameters</b>	<p>有効な LIN スケジューラー・エントリーを示すには、3つのパラメーター（3バイト）が必要です。1つはLIN メッセージ ID 用で、2つはそのタイムスタンプ用です。</p> <p>この例では、ID0x05 とタイムスタンプ 100ms（0x64）に準拠したエントリーがスケジューラーに挿入されます。</p> <p>パラメーターの順序は次のとおりです。</p> <p style="padding-left: 40px;">タイムスタンプ : 100 ms (0x0064)</p> <p style="padding-left: 40px;">LIN ID : 0x05</p>
<b>Checksum</b>	$PX = P1(0x64) \text{ XOR } P2(0x00) \text{ XOR } P3(0x05) = 0x01$ $SC(0x03) \text{ XOR } CC(0xA8) \text{ XOR } PX(0x61) = 0xCA$
<b>Message to send</b>	STX(0x02) – SC(0x03) – CC(0xA8) – P1(0x64) – P2(0x00) – P3(0x05) – CHK(0xCA)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### スレーブ ID + データ構成 (Slave ID + Data Configuration) の設定

このコマンドは、モジュールのスレーブ構成を設定します。この構成は、“Slave ID + Data Configuration”（スレーブ ID + データ構成）フィールドのステータス（有効または無効）によってコンフィグレーションされます。この ID は、スレーブとスレーブ ID に関連付けられたコンフィグレーション済みデータを識別します。データの長さはスレーブ ID によって異なります。使用されるフレームの長さがデフォルトの場合、使用される長さは次の表を使用して見つけることができます。

ID Range	Data Length
0x00 to 0x1F	2 バイト
0x20 to 0x2F	4 バイト
0x30 to 0x3F	8 バイト

使用する LIN ID のフレーム長が別の長さで構成されている場合は、その長さよりも長い長さを使用する必要があります。そうしないと、このコマンドは失敗します。フレーム長のリードの詳細については、53ページの Retrieve Frame Length（フレーム長の取得）コマンドを参照してください。フレーム長の構成については、71ページの Set Frame Length（フレーム長の設定）コマンドを参照してください。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	1	0	0	1	0	1	0	0	1
0	0, 0x0			4, 0x4		2, 0x2		41, 0x29					
4, 0x4					169, 0xA9								

<b>Parameters</b>	<p>この例では、ID 0x010 がスレーブ ID として設定されています。デフォルトの長さの使用が想定されています。“Slave ID + Data Configuration” (スレーブ ID + データ・コンフィグレーション) フィールドがアクティブに設定され、data [0]が 0x1A に、data [1]が 0xBB に設定されます (この場合、2 データ・バイト。上の表を参照)。</p> <p>したがって、このフィールドのアクティブ化ステータス (1 バイト)、設定する ID (1 バイト)、およびデータ・バイト (2 バイト) を示すには、4 つのパラメーター (4 バイト) が必要です。パラメーターの順序は次のとおりです。</p> <p style="padding-left: 40px;">フィールド・ステータス : アクティブ (1)</p> <p style="padding-left: 40px;">LIN ID : 0x10</p> <p style="padding-left: 40px;">データ・バイト 0 : 0x1A</p> <p style="padding-left: 40px;">データ・バイト 1 : 0xBB</p>
<b>Checksum</b>	<p>PX = P1(0x01) XOR P2(0x10) XOR P3(0x1A) XOR P4(0xBB) = 0xB0</p> <p>SC(0x04) XOR CC(0xA9) XOR PX(0xB0) = 0x1D</p>
<b>Message to send</b>	<p>STX(0x02) – SC(0x04) – CC(0xA9) – P1(0x01) – P2(0x10) – P3(0x1A) – P4(0xBB) – CHK(0x1D)</p>
<b>Positive response</b>	<p>STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)</p>

肯定応答には 1 つのパラメーターがあり、値 0 はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

### 同期フィールドのビットレート認識 (Bit Rate Recognition for Synchronization Field) を構成

このコマンドは、ビットレート認識・ステータスを設定します。自動ビットレート認識は、モジュールが受信したフレームに対してのみ機能し、同期ブレークが正しく認識された場合にのみ機能します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	1	0	1	0	1	0
0	0, 0x0			1, 0x1		2, 0x2		42, 0x2A						
1, 0x1					170, 0xAA									

<b>Parameters</b>	<p>モジュールの自動ビットレート認識機能のステータスを指定するには、1つのパラメーター（1バイト）が必要です。パラメーターが1の場合、この機能はアクティブになり（デフォルト）、そうでない場合は非アクティブになります。</p> <p>この例では、ビットレート認識がアクティブになっています。</p>
<b>Checksum</b>	<p>PX = P1(0x01) = 0x01</p> <p>SC(0x01) XOR CC(0xAA) XOR PX(0x01) = 0xAA</p>
<b>Message to send</b>	STX(0x02) – SC(0x01) – CC(0xAA) – P1(0x01) – CHK(0xAA)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### フレーム長 (Frame Length) を設定

このコマンドは、64個のLINメッセージIDの1つまたはすべてのフレーム長を設定します。コマンド Set Slave ID + Data Configuration (69ページ) に示されているように、LINメッセージのフレーム長はIDによって異なります。モジュールの機能の拡張により、任意のIDのフレーム長を0~8バイトに設定できます（フレーム長のデフォルト値は、前述のIDの規則に従います）。

受信するデータはLSBから始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

このコマンドは、2つの方法で実装できます。1つ目は、1回の操作でIDの範囲のフレーム長を設定することであり、2つ目は、単一のIDのフレーム長を設定することです。操作の種類に応じて、パラメーター数は3バイト（単一IDの場合）または9バイト（IDの範囲の場合）のいずれかになります。

## ID の範囲にフレーム長を割り当てる

ID には 4 つの範囲（0 から 3）があります。ID の各範囲は、モジュールに存在する 64 個の ID のうち 16 個の ID のフレーム長を構成します。フレーム長割り当ての最初のパラメーター（P1）は、構成されている ID の範囲を示します。これらの 16 個の ID のフレーム長は、ID のペア（8 バイト、P2 から P9）のパラメーター（1 バイト）を使用して割り当てられます。

次の表に ID 範囲を示します：

Range code (P1)	ID Range	
	from	to
0	ID 00 (0x00)	ID 15 (0x0F)
1	ID 16 (0x10)	ID 31 (0x1F)
2	ID 32 (0x20)	ID 47 (0x2F)
3	ID 48 (0x30)	ID 63 (0x3F)

選択した範囲に応じて、次の 8 つのパラメーター（8 バイト）には、その範囲に関連付けられた 16 個の ID のフレーム長が含まれます。各パラメーターの 4 ビットは、1 つの ID のフレーム長を表します。次の表は、ID のフレーム長を表すビットを示しています：

P1	Bits of P2...P9	P2	P3	P4	P5	P6	P7	P8	P9
0	0...3	ID 00	ID 02	ID 04	ID 06	ID 08	ID 10	ID 12	ID 14
	4...7	ID 01	ID 03	ID 05	ID 07	ID 09	ID 11	ID 13	ID 15
1	0...3	ID 16	ID 18	ID 20	ID 22	ID 24	ID 26	ID 28	ID 30
	4...7	ID 17	ID 19	ID 21	ID 23	ID 25	ID 27	ID 29	ID 31
2	0...3	ID 32	ID 34	ID 36	ID 38	ID 40	ID 42	ID 44	ID 46
	4...7	ID 33	ID 35	ID 37	ID 39	ID 41	ID 43	ID 45	ID 47
3	0...3	ID 48	ID 50	ID 52	ID 54	ID 56	ID 58	ID 60	ID 62
	4...7	ID 49	ID 51	ID 53	ID 55	ID 57	ID 59	ID 61	ID 63

たとえば、選択した範囲は 0 で、ID 10（4 バイト）と ID 11（6 バイト）を除いて、各 ID に必要なフレーム長は 8 である必要があります。その場合のパラメーターは次のとおりです。

P2	P3	P4	P5	P6	P7	P8	P9
0x88	0x88	0x88	0x88	0x88	0x64	0x88	0x88

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	1	0	0	1	0	1	0	1	1
0	0, 0x0			9, 0x9		2, 0x2		43, 0x2B				
9, 0x9					171, 0xAB							

<b>Parameters</b>	<p>このコマンドには9つのパラメーター（9バイト）が必要です。最初のパラメーターは、フレーム長の設定に使用される ID の範囲を示します。残りの8つのパラメーター P2~P9 は、選択した範囲に関連付けられた 16 個の ID のフレーム長を示します。</p> <p>この例では、最後の 16 個の ID（range number [範囲番号] 3）のフレーム長がデフォルト値（8バイト）に設定されています。</p>
<b>Checksum</b>	$PX = P1(0x03) \oplus P2(0x88) \oplus P3(0x88) \oplus P4(0x88) \oplus P5(0x88) \oplus P6(0x88) \oplus P7(0x88) \oplus P8(0x88) \oplus P9(0x88) = 0x03$ $SC(0x09) \oplus CC(0xAB) \oplus PX(0x03) = 0xA1$
<b>Message to send</b>	$STX(0x02) - SC(0x09) - CC(0xAB) - P1(0x03) - P2(0x88) - P3(0x88) - P4(0x88) - P5(0x88) - P6(0x88) - P7(0x88) - P8(0x88) - P9(0x88) - CHK(0xA1)$
<b>Positive response</b>	$STX(0x02) - SC(0x01) - P1(0x00) - CHK(0x01)$

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### 単一の ID にフレーム長を割り当てる

単一の ID のフレーム長を設定するには、パラメーター1（P1）を4に設定する必要があります。この場合、コマンドはさらに2つのパラメーター（P2とP3）のみを受け入れます。P2はIDを識別するために使用され、P3は対応するフレーム長を指定するために使用されます。P3の値は1から8の範囲でなければなりません。そうでない場合、このコマンドはエラーを返します。IDをフレーム長0に設定する場合は、範囲による割り当てを使用する必要があります（上記を参照）。

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	1	1	1	0	1	0	1	1
0	0, 0x0			3, 0x3		2, 0x2		43, 0x2B						
3, 0x3					171, 0xAB									

<b>Parameters</b>	<p>このコマンドには3つのパラメーター（3バイト）が必要です。最初のパラメーターの値は4（0x04）で、単一のIDが構成されていることを示します。2番目のパラメーターは、目的のID（0x00から0x3F）を示します。3番目のパラメーターは、フレームの長さ（0x01から0x08）を示します。</p> <p>この例では、ID 64（0x3F）がフレーム長5に設定されています。</p>
<b>Checksum</b>	$PX = P1(0x04) \text{ XOR } P2(0x3F) \text{ XOR } P3(0x05) = 0x3E$ $SC(0x03) \text{ XOR } CC(0xAB) \text{ XOR } PX(0x3E) = 0x96$
<b>Message to send</b>	$STX(0x02) - SC(0x03) - CC(0xAB) - P1(0x04) - P2(0x3F) - P3(0x05) -$ $CHK(0x96)$
<b>Positive response</b>	$STX(0x02) - SC(0x01) - P1(0x00) - CHK(0x01)$

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。



**注：** コマンド "Slave ID + Data Configuration"（スレーブ ID + データ・コンフィグレーション）でも使用する ID を変更すると、その設定も変更されます。フレーム長が "Slave ID + Data Configuration"（スレーブ ID + データ・コンフィグレーション）で使用される数値よりも小さい数値に設定されている場合、そのデータは切り捨てられます。フレーム長がより大きな数値に設定されている場合、データは値 0xFF で埋められます。

### チェックサム・タイプ（Checksum Type）の設定

このコマンドは、単一またはすべての64個のLINメッセージIDのチェックサム・タイプ（Checksum Type）を設定します。これは、クラシック・フォーマット（LIN 1.x）またはエンハンスド（拡張）・フォーマット（LIN 2.0）のいずれかです。

このコマンドは、2つの異なる方法で実装できます。1つ目は、1回の操作ですべてのIDのチェックサム・タイプ（Checksum Type）を設定することです。2つ目は、単一IDのチェックサム・タイプ（Checksum Type）を設定することです。操作の種類に応じて、パラメーター数は3バイト（単一IDの場合）または9バイト（すべてのIDの場合）のいずれかになります。

### すべての ID にチェックサム・タイプ (Checksum Type) を割り当てる

最初のパラメーター (P1) が 0 の場合、チェックサム・タイプ (Checksum Type) がすべての ID に設定されます。次の 8 つのパラメーター (8 バイト) には、ID のチェックサム・タイプ (Checksum Type) が含まれ、各データ・バイトの各ビットをブール変数として使用します (0 はクラシック・フォーマットを意味し、1 は拡張フォーマットを意味します)。次の表は、8 データ・バイト内の各 ID の位置を示しています。

Param	Bit assignment for LIN IDs							
	7	6	5	4	3	2	1	0
P2	0x07	0x06	0x05	0x04	0x03	0x02	0x01	0x00
P3	0x0F	0x0E	0x0D	0x0C	0x0B	0x0A	0x09	0x08
P4	0x17	0x16	0x15	0x14	0x13	0x12	0x11	0x10
P5	0x1F	0x1E	0x1D	0x1C	0x1B	0x1A	0x19	0x18
P6	0x27	0x26	0x25	0x24	0x23	0x22	0x21	0x20
P7	0x2F	0x2E	0x2D	0x2C	0x2B	0x2A	0x29	0x28
P8	0x37	0x36	0x35	0x34	0x33	0x32	0x31	0x30
P9	0x3F	0x3E	0x3D	0x3C	0x3B	0x3A	0x39	0x38

たとえば、最初の 8 つの ID (P2 を使用) のフォーマットを ID 3 を除いて拡張に設定すると、パラメーター P2 は 0xF7 (0b11110111) である必要があります。0 に設定されたビットは、クラシック・チェックサム・タイプ (Classic checksum type) を使用する ID 3 (0x03) に対応します。

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	1	0	0	1	1	0	1	0	1	1	0	0
0	0, 0x0			9, 0x9				2, 0x2		44, 0x2C					
9, 0x9				172, 0xAC											

<b>Parameters</b>	このコマンドには9つのパラメーター（9バイト）が必要です。最初のパラメーターは、すべてのIDのチェックサム・タイプ（Checksum Type）が設定されていることを示します。残りの8つのパラメーター P2 から P9 は、IDのチェックサム・タイプ（Checksum Type）を示します。 この例では、ID 0~31のチェックサム・タイプ（Checksum Type）がデフォルト値クラシック（Classic）に設定され、ID 32~64 が拡張に設定されています。
<b>Checksum</b>	$PX = P1(0x00) \text{ XOR } P2(0x00) \text{ XOR } P3(0x00) \text{ XOR } P4(0x00) \text{ XOR } P5(0x00) \text{ XOR } P6(0xFF) \text{ XOR } P7(0xFF) \text{ XOR } P8(0xFF) \text{ XOR } P9(0xFF) = 0x00$ $SC(0x09) \text{ XOR } CC(0xAC) \text{ XOR } PX(0x00) = 0xA5$
<b>Message to send</b>	STX(0x02) – SC(0x09) – CC(0xAC) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0xFF) – P7(0xFF) – P8(0xFF) – P9(0xFF) – CHK(0xA5)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### 単一のIDへのチェックサム・タイプ（Checksum Type）の割り当て

単一のIDのフレーム長を設定するには、パラメーター P1 を1に設定する必要があります。この場合、コマンドには2つの追加パラメーター（P2 と P3）が必要です。P2 はIDを識別するために使用されます。P3 は、チェックサム・タイプ（Checksum Type）を指定するために使用されます。このパラメーターの値は、クラシックの場合は0、拡張の場合は1でなければなりません。そうでない場合、このコマンドはエラーを返します。

Sequence Code (SC)				Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	1	1	1	0	1	1	0	0
0	0, 0x0			3, 0x3		2, 0x2		44, 0x2C					
3, 0x3				172, 0xAC									

<b>Parameters</b>	このコマンドには3つのパラメーター（3バイト）が必要です。最初のパラメーターの値は1（0x01）で、単一のIDが構成されていることを示します。2番目のパラメーターには目的のID（0x00から0x3F）があります。3番目のパラメーターは、フレームのフォーマットです（classic [クラシック] の場合は0x00、enhanced [拡張] の場合は0x01）。 この例では、ID 64（0x3F）が拡張チェックサム・タイプ（enhanced checksum type）に設定されています。
<b>Checksum</b>	$PX = P1(0x01) \text{ XOR } P2(0x3F) \text{ XOR } P3(0x01) = 0x3F$ $SC(0x03) \text{ XOR } CC(0xAC) \text{ XOR } PX(0x3F) = 0x90$
<b>Message to send</b>	STX(0x02) – SC(0x03) – CC(0xAC) – P1(0x01) – P2(0x3F) – P3(0x01) – CHK(0x90)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### スレーブ・アクティベーションのCAN ID（CAN ID for Slave Activation）を設定

このコマンドは、CANメッセージングおよびそのステータス（アクティブまたは非アクティブ）を介してモジュールをLINスレーブとしてアクティブ化するために使用されるCAN IDを設定します。

受信するデータはLSBから始まります。例：0x1 = P1：0x01 および P2：0x00

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	1	0	0	1	0	1	1	0	1
0	0, 0x0			4, 0x4		2, 0x2		45, 0x2D					
4, 0x4					173, 0xAD								

<b>Parameters</b>	4つのパラメーター（4バイト）が必要です。4バイトは32ビット値を形成します。P4のビット5～7は、それぞれブール値として使用され、次の構成を指定します。		
	<b>Bit</b>	<b>Set to 1</b>	<b>Set to 0</b>
	7	Extended CAN ID（29ビット）	Standard CAN ID（11ビット）
	6	機能が有効	機能が無効
	5	予約済み	
	この値の残りの29ビットは、CAN IDを識別します。たとえば、CAN ID 0x1AABBCCD（29ビット）を表し、この機能をアクティブにするには、パラメーターをP1（0xCD）、P2（0xBC）、P3（0xAB）、P4（0xDA）にする必要があります。この例では、CAN IDはデフォルト値（ID：0x7F0、長さ：11ビット、ステータス：無効）に設定されています。		
<b>Checksum</b>	$PX = P1(0xF0) \text{ XOR } P2(0x07) \text{ XOR } P3(0x00) \text{ XOR } P4(0x00) = 0xF7$ $SC(0x04) \text{ XOR } CC(0xAD) \text{ XOR } PX(0xF7) = 0x5E$		
<b>Message to send</b>	STX(0x02) – SC(0x04) – CC(0xAD) – P1(0xF0) – P2(0x07) – P3(0x00) – P4(0x00) – CHK(0x5E)		
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)		

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### スレーブ・マスク・ステータス（Slave Mask Status）の設定

このコマンドは、1つまたはすべてのLINメッセージIDのスレーブ・マスクのステータス（Publisher：パブリッシャーまたはSubscriber：サブスクリイバー）を設定します。このコマンドは、2つの異なる方法で実装できます。1つ目は、すべてのIDのスレーブ・マスク・ステータスを設定することです。2つ目は、単一IDのスレーブ・マスク・ステータスを設定することです。操作の種類に応じて、パラメーター数は3バイト（単一のIDの場合）または9バイト（すべてのIDの場合）のいずれかになります。

すべてのIDへのスレーブ・マスク・ステータスの割り当て

最初のパラメーター（P1）が0の場合、すべてのIDにスレーブ・マスク・ステータスが設定されます。

次の8つのパラメーター（8バイト）には、IDのスレーブ・マスク・ステータスが生まれ、各データ・バイトの各ビットをブール変数として使用します（0はSubscriber：サブスクリイバー、1はPublisher：パブリッシャーを意味します）。次の表は、8データ・バイト内の各IDの位置を示しています：

Param	Bit assignment for LIN IDs							
	7	6	5	4	3	2	1	0
P2	0x07	0x06	0x05	0x04	0x03	0x02	0x01	0x00
P3	0x0F	0x0E	0x0D	0x0C	0x0B	0x0A	0x09	0x08
P4	0x17	0x16	0x15	0x14	0x13	0x12	0x11	0x10
P5	0x1F	0x1E	0x1D	0x1C	0x1B	0x1A	0x19	0x18
P6	0x27	0x26	0x25	0x24	0x23	0x22	0x21	0x20
P7	0x2F	0x2E	0x2D	0x2C	0x2B	0x2A	0x29	0x28
P8	0x37	0x36	0x35	0x34	0x33	0x32	0x31	0x30
P9	0x3F	0x3E	0x3D	0x3C	0x3B	0x3A	0x39	0x38

たとえば、値が 0x82 (0b10000010) の P4 を送信する場合、これは ID 16 (0x10) 、 18 (0x12) 、 19 (0x13) 、 20 (0x14) 、 21 (0x15) のスレーブ・マスクを意味します。および 22 (0x16) は Subscriber : サブスクライバーとして設定され、ID 17 (0x11) および 23 (0x17) は Publisher : パブリッシャーとして設定されます。

受信するデータは LSB から始まります。例 : 0x011C = P1 : 0x1C および P2 : 0x01

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	1	0	0	1	0	1	1	1	0
0	0, 0x0			9, 0x9		2, 0x2		46, 0x2E				
9, 0x9					174, 0xAE							

<b>Parameters</b>	このコマンドには9つのパラメーター（9バイト）が必要です。最初のパラメーターは、すべてのIDのスレーブ・マスク・ステータスが設定されていることを示します。残りの8つのパラメーター P2~P9 は、IDのスレーブ・マスク・ステータスを示します。 この例では、ID 0~31のスレーブ・マスク・ステータスがデフォルト値（Subscriber：サブスクライバ）に設定されています。
<b>Checksum</b>	$PX = P1(0x00) \text{ XOR } P2(0x00) \text{ XOR } P3(0x00) \text{ XOR } P4(0x00) \text{ XOR } P5(0x00) \text{ XOR } P6(0x00) \text{ XOR } P7(0x00) \text{ XOR } P8(0x00) \text{ XOR } P9(0x00) = 0x00$ $SC(0x09) \text{ XOR } CC(0xAE) \text{ XOR } PX(0x00) = 0xA7$
<b>Message to send</b>	STX(0x02) – SC(0x09) – CC(0xAE) – P1(0x00) – P2(0x00) – P3(0x00) – P4(0x00) – P5(0x00) – P6(0x00) – P7(0x00) – P8(0x00) – P9(0x00) – CHK(0xA7)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### スレーブ・マスク・ステータスを単一のIDに割り当てる

単一のIDのスレーブ・マスク・ステータスを設定するには、パラメーター P1を1に設定する必要があります。この場合、コマンドには2つの追加パラメータ（P2とP3）が必要です。P2はIDを識別するために使用されます。P3はステータスを指定するために使用されます。このパラメーターの値は、Subscriber：サブスクライバーの場合は0、Publisher：パブリッシャーの場合は1である必要があります。そうでない場合、このコマンドはエラーを返します。

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count			Interface		Command Code						
0	0	0	0	0	0	1	1	1	0	1	0	1	1	1	0
0	0, 0x0			3, 0x3			2, 0x2		46, 0x2E						
3, 0x3				174, 0xAE											

<b>Parameters</b>	<p>このコマンドには3つのパラメーター（3バイト）が必要です。最初のパラメーターの値は1（0x01）で、単一のIDが構成されていることを示します。2番目のパラメーターには目的のID（0x00から0x3F）があります。3番目のパラメーターのステータスはスレーブ・マスクです（Subscriber：サブスクライバーの場合は0x00、Publisher：パブリッシャーの場合は0x01）。</p> <p>この例では、ID 64（0x3F）のスレーブ・マスクが Publisher：パブリッシャーに設定されています。</p>
<b>Checksum</b>	$PX = P1(0x01) \text{ XOR } P2(0x3F) \text{ XOR } P3(0x01) = 0x3F$ $SC(0x03) \text{ XOR } CC(0xAE) \text{ XOR } PX(0x3F) = 0x92$
<b>Message to send</b>	STX(0x02) – SC(0x03) – CC(0xAE) – P1(0x01) – P2(0x3F) – P3(0x01) – CHK(0x92)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

#### 2.4.4 リセット/消去 コマンド (Reset / Erase Commands)

Command	on page
メッセージ送信タイムアウト (Message Transmission Timeouts) のリセット	82
マスター・スケジューラー・リスト (Master Scheduler List) の消去	82
RCV エラー・カウンター (RCV Error Counter) をリセット	83
XMT エラー・カウンター (XMT Error Counter) をリセット	84
ビットエラー・カウンター (Bit Error Counter) のリセット	84
チェックサム・エラー・カウンター (Checksum Error Counter) をリセット	85
ID チェックサム・エラー・カウンター (ID Checksum Error Counter) をリセット	85
”スレーブが応答していません”エラー・カウンターをリセット (“Slave not responding” Error Counter)	86
一貫性のない同期バイト・カウンターをリセット (“Slave not responding” Error Counter)	87
すべてのエラー・カウンター (All Error Counters) をリセット	87

### メッセージ送信タイムアウト (Message Transmission Timeouts) のリセット

このコマンドは、さまざまな LIN 送信メッセージのタイムアウトをリセットするようにモジュールに指示します。バス・アイドル・メッセージとフレーム長 8、4、および 2 バイトのメッセージにはタイムアウトがあります。リセット後、これらのタイミングにはデフォルト値が含まれます。

- Bus idle: 0x516 = 1302 ms
- Frame length 8: 0x0A = 10 ms
- Frame length 4: 0x07 = 7 ms
- Frame length 2: 0x05 = 5 ms

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	0	1	1	0	0	1	0
0	0, 0x0			0, 0x0		2, 0x2		50, 0x32						
0, 0x0					178, 0xB2									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB2) XOR PX(0x00) = 0xB2
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB2) – CHK(0xB2)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には 1 つのパラメーターがあり、値 0 はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

### マスター・スケジューラー・リスト (Master Scheduler List) の消去

このコマンドは、モジュール内のマスター・スケジューラーからすべてのエントリーを削除するようにモジュールに指示します。マスターがアクティブな場合、非アクティブに設定されます。スケジューラー・エントリーは後で回復できません。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
0	0, 0x0			0, 0x0		2, 0x2		51, 0x33						
0, 0x0					179, 0xB3									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB3) XOR PX(0x00) = 0xB3
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB3) – CHK(0xB3)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### RCV エラー・カウンター (RCV Error Counter) をリセット

このコマンドは、LIN インターフェイスの受信エラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。 例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	1	1	0	1	0	0	0
0	0, 0x0			0, 0x0		2, 0x2		52, 0x34						
0, 0x0					180, 0xB4									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB4) XOR PX(0x00) = 0xB4
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB4) – CHK(0xB4)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### XMTエラー・カウンター (XMT Error Counter) をリセット

このコマンドは、LIN インターフェイスの送信エラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
0	0, 0x0			0, 0x0		2, 0x2		53, 0x35						
0, 0x0					181, 0xB5									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB5) XOR PX(0x00) = 0xB5
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB5) – CHK(0xB5)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### ビットエラー・カウンター (Bit Error Counter) のリセット

このコマンドは、LIN インターフェイスのビットエラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	0	1	1	0	1	1	0
0	0, 0x0			0, 0x0		2, 0x2		54, 0x36						
0, 0x0					182, 0xB6									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB6) XOR PX(0x00) = 0xB6
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB6) – CHK(0xB6)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### チェックサム・エラー・カウンター (Checksum Error Counter) をリセット

このコマンドは、LIN インターフェイスの Checksum error (チェックサム・エラー) のカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)				Control Code (CC)											
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1
0	0, 0x0			0, 0x0				2, 0x2		55, 0x37					
0, 0x0				183, 0xB7											

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB7) XOR PX(0x00) = 0xB7
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB7) – CHK(0xB7)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### ID チェックサム・エラー・カウンター (ID Checksum error) をリセット

このコマンドは、LIN インターフェイスの ID Checksum error (ID チェックサム・エラー) のカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
0	0, 0x0			0, 0x0				2, 0x2		56, 0x38					
0, 0x0							184, 0xB8								

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB8) XOR PX(0x00) = 0xB8
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB8) – CHK(0xB8)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

**“スレーブが応答していません” エラー・カウンター (“Slave not responding” Error Counter) をリセット**

このコマンドは、LIN インターフェイスの “Slave not responding” (“スレーブが応答していません”) エラーのカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	1
0	0, 0x0			0, 0x0				2, 0x2		57, 0x39					
0, 0x0							185, 0xB9								

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xB9) XOR PX(0x00) = 0xB9
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xB9) – CHK(0xB9)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### 一貫性のない同期バイト・カウンター (Inconsistent Synchronization Bytes Counter) をリセット

このコマンドは、LIN インターフェイスの同期バイトに一貫性がない場合にカウンターをリセットするようにモジュールに指示します。

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	0	0	0	0	1	1	1	0	1	0
0	0, 0x0			0, 0x0		2, 0x2		58, 0x3A						
0, 0x0					186, 0xBA									

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xBA) XOR PX(0x00) = 0xBA
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xBA) – CHK(0xBA)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメータがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### すべてのエラー・カウンター (All Error Counters) をリセット

このコマンドは、LIN インターフェイスのすべてのエラー・カウンターをリセットするようにモジュールに指示します。それらのカウンターは次のとおりです。

- RCV エラー (RCV errors)
- XMT エラー (XMT errors)
- ビットエラー (Bit errors)
- チェックサム・エラー (Checksum errors)
- ID チェックサム・エラー (ID checksum errors)
- “Slave not responding” エラー (“Slave not responding” errors)
- 一貫性のない同期バイト (Inconsistent synchronization bytes)

受信するデータは LSB から始まります。例：0x1 = P1 : 0x01 および P2 : 0x00

Sequence Code (SC)					Control Code (CC)										
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1
0	0, 0x0			0, 0x0				2, 0x2		59, 0x3B					
0, 0x0					187, 0xBB										

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xBB) XOR PX(0x00) = 0xBB
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xBB) – CHK(0xBB)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

## 2.5 モジュール・インターフェイス (Module Interface)

### 2.5.1 初期化コマンド (Initialization Commands)

Command	on page
モジュールのデフォルト値 (Module Default Values)	88

#### モジュールのデフォルト値 (Module Default Values)

このコマンドは、現在の構成をメモリに保存されているデフォルト値に設定し、すべてを初期化するようにモジュールに指示します。

Sequence Code (SC)					Control Code (CC)										
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0	0, 0x0			0, 0x0				3, 0x3		2, 0x2					
0, 0x0					194, 0xC2										

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xC2) XOR PX(0x00) = 0xC2
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xC2) – CHK(0xC2)
<b>Positive response</b>	正常に実行されると、RS-232 も初期化され、接続が中断されます。したがって、応答は送信されません。これはエラーではありません。

## 2.5.2 リード・コマンド (Read Commands)

Command	on page
ハードウェア・バージョンをリード (Read Hardware Version)	89
ファームウェア・バージョンをリード (Read Firmware Version)	91

### ハードウェア・バージョンをリード (Read Hardware Version)

このコマンドは、モジュールのハードウェア・バージョンを取得します。

Sequence Code (SC)					Control Code (CC)								
X	Sequence			Param Byte Count		Interface		Command Code					
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0, 0x0			0, 0x0		3, 0x3		0, 0x0					
0, 0x0					192, 0xC0								

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xC0) XOR PX(0x00) = 0xC0
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xC0) – CHK(0xC0)
<b>Positive response</b>	STX(0x02) – SC(0x0F) – P1(0x30) – P2(0x32) – P3(0x2E) – P4(0x30) – P5(0x32) – P6(0x2E) – P7(0x30) – P8(0x31) – P9(0x20) – P10(0x48) – P11(0x53) – P12(0x2D) – P13(0x43) – P14(0x41) – P15(0x4E) – CHK(0x37)

肯定応答には 15 個のパラメーターがあります。返されるデータ・バイトは ASCII 文字です。解釈は次のとおりです。

Information	Parameters
ハードウェアメジャー・バージョン (Hardware – major version)	P1, P2
ハードウェアマイナー・バージョン (Hardware – minor version)	P4, P5
ハードウェアリリース・バージョン (Hardware – release version)	P7, P8
CAN トランシーバー (CAN transceiver)	P10...P15
データ・セパレーター (Data separators)	P3, P6, P9

ハードウェアメジャー・バージョン (Hardware – major version)

メジャー・バージョンは ASCII 値、通常は数字として受信されます。たとえば、文字 "0" (P1 = 0x30) および文字 "1" (P2 = 0x31) は、メジャー・バージョン 01 を示します。特殊な種類のハードウェア・バージョンには、特殊なプロパティを識別するための数字以外の文字を含めることができます。

たとえば、S1 は特別なメジャー・バージョン 01 を表すことができます。

ハードウェアマイナー・バージョン (Hardware – minor version)

マイナーバージョンは ASCII 値 (数字) として受信されます。

たとえば、文字 "0" (P4 = 0x30) および文字 "2" (P5 = 0x32) は、マイナーバージョン 02 を示します。

ハードウェアリリース・バージョン (Hardware – release version)

リリース・バージョンは ASCII 値 (数字) として受信されます。

たとえば、文字 "5" (P7 = 0x35) および文字 "1" (P8 = 0x31) は、リリース・バージョン 51 を示します。

CAN トランシーバー (CAN transceiver)

CAN トランシーバーはテキスト (ASCII 文字) として返されます。このテキストは、トランシーバーの種類を示す 2 つの大文字で構成されています。

Parameters Characters	Characters	CAN transceiver
P10 = 0x48, P11 = 0x53	HS	High-speed according to ISO 11898-2 (ISO11898-2 に準拠した High-speed)
P10 = 0x4C, P11 = 0x53	LS	Low-speed according to ISO 11898-3 (ISO11898-3 に準拠した Low-speed)

文字列 "-CAN" (P12 = 0x2D、P13 = 0x43、P14 = 0x41、P15 = 0x4E) がサフィックスとして使用されます。たとえば、高速 CAN トランシーバーがモジュールに実装されている場合、返されるテキストは "HS-CAN" です。

### データ・セパレーター (Data separators)

このコマンドへの応答では、3つの区切り文字が使用されます。ピリオド文字 (P3、P6 = 0x2E = ".") は、ハードウェアのメジャー、マイナー、およびリリースバージョンを区切るために2回使用されます。空白文字 (P9 = 0x20 = " ") は、ハードウェア・バージョンを CAN トランシーバーのタイプから区別するために使用されます。

上記の肯定応答の例を解釈すると、次の文字列連結が生成されます。

- ハードウェア・バージョン= "02.02.01"  
P1 (0x30) P2 (0x32) P3 (0x2E) P4 (0x30) P5 (0x32) P6 (0x2E) P7 (0x30) P8 (0x31)
- 空白の区切り文字= " "  
P9 (0x20)
- CAN トランシーバー= "HS-CAN"  
P10 (0x48) P11 (0x53) P12 (0x2D) P13 (0x43) P14 (0x41) P15 (0x4E)

### ファームウェア・バージョンをリード (Read Firmware Version)

このコマンドは、モジュールのファームウェア・バージョンを取得します。

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1
0	0, 0x0			0, 0x0				3, 0x3		1, 0x1					
0, 0x0							193, 0xC1								

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xC1) XOR PX(0x00) = 0xC1
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xC1) – CHK(0xC1)
<b>Positive response</b>	STX(0x02) – SC(0x08) – P1(0x30) – P2(0x31) – P3(0x2E) – P4(0x30) – P5(0x31) – P6(0x2E) – P7(0x30) – P8(0x31) – CHK(0x0D)

肯定応答には 8 つのパラメータがあります。返されるデータ・バイトは ASCII 文字です。解釈は次のとおりです。

Information	Parameters
ファームウェアメジャー・バージョン (Firmware – major version)	P1, P2
ファームウェアマイナー・バージョン (Firmware – minor version)	P4, P5
ファームウェアリリース・バージョン (Firmware – release version)	P7, P8
データ・セパレーター (Data separators)	P3, P6

ファームウェアメジャー・バージョン (Firmware – major version)

メジャー・バージョンは ASCII 値 (数字) として受信されます。

たとえば、文字 “0” (P1 = 0x30) および文字 “1” (P2 = 0x31) は、メジャー・バージョン 01 を示します。

ファームウェアマイナー・バージョン (Firmware – minor version)

マイナーバージョンは ASCII 値 (数字) として受信されます。

たとえば、文字 “0” (P4 = 0x30) および文字 “2” (P5 = 0x32) は、マイナーバージョン 02 を示します。

ファームウェアリリース・バージョン (Firmware – release version)

リリース・バージョンは ASCII 値 (数字) として受信されます。

たとえば、文字 “5” (P7 = 0x35) および文字 “1” (P8 = 0x31) は、リリース・バージョン 51 を示します。

データ・セパレーター (Data separators)

このコマンドへの応答では、2 つの区切り文字が使用されます。ピリオド文字 (P3、P6 = 0x2E = “.”) は、ファームウェアのメジャー、マイナー、およびリリースバージョンを区切るために 2 回使用されます。

上記の肯定応答の例を解釈すると、次の文字列連結が生成されます。

– ファームウェア・バージョン = “01.01.01”

P1 (0x30) P2 (0x31) P3 (0x2E) P4 (0x30) P5 (0x31) P6 (0x2E) P7 (0x30) P8 (0x31)

### 2.5.3 コンフィグレーション・コマンド (Configuration Commands)

Command	on page
フラッシュの現在の構成 (Flash Current Configuration)	93

#### フラッシュの現在の構成 (Flash Current Configuration)

このコマンドは、モジュールの現在の構成を取得して、モジュールのフラッシュメモリに保存するようにモジュールに指示します。その後、モジュールがリセットまたは再起動されるたびに、構成がフラッシュメモリからリロードされます

Sequence Code (SC)					Control Code (CC)							
X	Sequence			Param Byte Count		Interface		Command Code				
0	0	0	0	0	0	0	0	0	0	0	1	1
0	0, 0x0			0, 0x0		3, 0x3		3, 0x3				
0, 0x0					195, 0xC3							

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xC3) XOR PX(0x00) = 0xC3
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xC3) – CHK(0xC3)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

肯定応答には1つのパラメーターがあり、値0はエラーがないことを意味します。エラー・コードの詳細については、103ページの第3章 エラー・メッセージを参照してください。

### 2.5.4 リセット/消去 コマンド (Reset / Erase Commands)

Command	on page
モジュールのリセット (Reset Module)	94

## モジュールのリセット (Reset Module)

このコマンドは、フラッシュメモリからロードされた値で初期化するようにモジュールに指示します。

Sequence Code (SC)					Control Code (CC)						
X	Sequence			Param Byte Count		Interface		Command Code			
0	0	0	0	0	0	0	0	0	1	0	0
0	0, 0x0			0, 0x0		3, 0x3		4, 0x4			
0, 0x0					196, 0xC4						

<b>Parameters</b>	None
<b>Checksum</b>	SC(0x00) XOR CC(0xC4) XOR PX(0x00) = 0xC4
<b>Message to send</b>	STX(0x02) – SC(0x00) – CC(0xC4) – CHK(0xC4)
<b>Positive response</b>	ASCII フォーマットのテキスト情報

このコマンドに対する肯定的な応答は、説明テキストです。この応答は、他のコマンドの応答と同じルールには従いません。つまり、STX、SC、または CC バイトを使用しません。返されるテキストは、モジュールの名前、ハードウェアとファームウェアのバージョン、および製造元の組み合わせです。このテキストの例：

PCAN-LIN Ver. 02.02.01 HS-CAN/01.01.01 (c) PEAK-System Technik GmbH

応答の終わりは、carriage return (キャリッジ・リターン) + line feed (ライン・フィード) でマークされます。この応答の情報と長さは異なる場合があります。

モジュールをリセットすると、モジュールの電源を入れた後と同じ LED テスト手順が実行されます (ユーザーマニュアルを参照)。リセット時に使用したボーレートがフラッシュメモリに保存されているボーレートと異なる場合、モジュールとの RS-232 接続が失われます。

## 2.6 メッセージの送信と受信 (Message Transmission and Reception)

モジュールは、RS-232 インターフェイスを介して CAN および LIN メッセージを送受信できます。

### 2.6.1 CAN

Command	on page
データのライト (Write Data)	95
データのレシーブ (Receive Data)	97

#### データのライト (Write Data)

このコマンドは、このドキュメントで前述した他のコマンドと同様にメッセージ構造を使用します。モジュールは、埋め込まれたメッセージを CAN インターフェイスに転送します。メッセージ構造のデータ部分は、送信する CAN メッセージの構成によって異なります。

Description	Byte count	Bit position							
		7	6	5	4	3	2	1	0
STX	1	0	0	0	0	0	0	1	0
Sequence code	1	0	Sequence number: 0b000...0b111 (0...7)			Parameter byte count: 1 + ID length (2, 4) + data length n (0...8)			
Control code	1	Target interface: 0b01 (1 = CAN)		Command code: 0b000001 (1 = Write data on bus)					
CAN message info	1	0: STD 1: EXT	0: Data 1: RTR	0	0	CAN data length n			
Parameters	CAN ID	CAN ID STD: 0...0x07FF, 11 bits							
	STD/EXT	STD: 2 EXT: 4	CAN ID EXT: 0...0x1FFFFFFF, 29 bits from LSB to MSB						
CAN data bytes	n (0...8)	Data bytes							
Checksum	1	STXバイトを除くこのメッセージのすべてのバイト間のビット単位 のXOR演算。							

- 送信する CAN メッセージがリモート要求フレーム (RTR) の場合、CAN メッセージ情報の CAN データ長は 0 でなければなりません。それ以外の場合、コマンドはエラーを返し、メッセージは CAN インターフェイスに転送されません。
- 肯定応答には 1 つのパラメーターがあり、値 0 はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

たとえば、標準 ID (2 バイト) と 3 データ・バイトのメッセージを送信するには、Sequence code (シーケンス・コード) は 0x06 である必要があります :

Sequence Code (SC)							Control Code (CC)								
X	Sequence			Param Byte Count				Interface		Command Code					
0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1
0	0, 0x0			6, 0x6				1, 0x1		1, 0x1					
6, 0x6							65, 0x41								

<b>Parameters</b>	この例では、3 データバイト (0x0A、0x0B、0x0C) の標準 CAN メッセージ (ID : 0x123) が送信されます。
<b>Checksum</b>	$PX = P1(0x03) \text{ XOR } P2(0x23) \text{ XOR } P3(0x01) \text{ XOR } P4(0x0A)$ $\text{XOR } P5(0x0B) \text{ XOR } P6(0x0C) = 0x2C$ $SC(0x06) \text{ XOR } CC(0x41) \text{ XOR } PX(0x2C) = 0x6B$
<b>Message to send</b>	STX(0x02) – SC(0x06) – CC(0x41) – P1(0x03) – P2(0x23) – P3(0x01) – P4(0x0A) – P5(0x0B) – P6(0x0C) – CHK(0x6B)
<b>Positive response</b>	STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)

## データのレシーブ (Receive Data)

CAN メッセージが RS-232 インターフェイスに転送されると (24 ページのコマンド **フォワード・マスクを取得する** [Retrieve Forward Mask]を参照)、CAN メッセージはデータ部分としてコマンド・ストラクチャに埋め込まれます。

Description	Byte count	Bit position							
		7	6	5	4	3	2	1	0
STX	1	0	0	0	0	0	0	1	0
Sequence code	1	1 (AR)	Sequence number: 0b001 (1)			Parameter byte count (1 + ID length + data length n)			
CAN message info	1	0: STD 1: EXT	0: Data 1: RTR	ERR	0	CAN data length n			
Parameters CAN ID STD/EXT	STD: 2 EXT: 4	CAN ID STD: 0...0x07FF, 11 bits CAN ID EXT: 0...0x1FFFFFFF, 29 bits from LSB to MSB							
	CAN data bytes	n (0...8)	Data bytes						
Checksum	1	STXバイトを除くこのメッセージのすべてのバイト間のビット単位のXOR演算。							

- CAN メッセージが転送されるため、自動応答ビット (AR) が設定されます。
- Sequence number (シーケンス番号) は 1 です。これは、次のデータが CAN インターフェイスから転送されたメッセージであることを意味します。
- 受信した CAN メッセージがリモート要求フレーム (RTR) の場合、CAN データ長は 0 です。
- エラービット (ERR) が 1 の場合、CAN エラー・フレームが受信されます。エラー・フレームは、CAN インターフェイスのフォワード・マスクが CAN エラーを監視および転送するように設定されている場合にのみ受信されます (詳細については、コマンド Set Forward Mask (32 ページ) を参照してください)。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

たとえば、標準 ID 0x456 (2 バイト) および 3 データバイト (0x0D、0x0E、0x0F) で転送されたメッセージを受信すると、次のようなフォーマットになります。

受信したメッセージ : STX (0x02) – SC (0x96) – P1 (0x03) – P2 (0x56) – P3 (0x04) – P4 (0x0D) – P5 (0x0E) – P6 (0x0F) – CHK (0xCB)

Sequence code (SC = 0x96)								
Bit	7	6	5	4	3	2	1	0
Description	Auto reply	Sequence number			Parameter byte count			
Contents	1	0	0	1	0	1	1	0

CAN message info (P1 = 0x03)								
Bit	7	6	5	4	3	2	1	0
Description	Frame format	RTR	Error	Fixed	CAN data length			
Contents	0 (STD)	0	0 (None)	0	0	0	1	1

フレームフォーマットは標準です : CAN ID は 2 バイト長です (P2 と P3) :

– CAN ID = 0x56 0x04

CAN data 長は 3 です。

– P4: CAN data [0] = 0x0D

– P5: CAN data [1] = 0x0E

– P6: CAN data [2] = 0x0F

## 2.6.2 LIN

Command	on page
データのライト (Write Data)	99
データのレシーブ (Receive Data)	100

## データのライト (Write Data)

このコマンドは、このドキュメントで前述した他のコマンドと同様にメッセージ構造を使用します。モジュールは、埋め込まれたメッセージを LIN インターフェイスに転送します。メッセージ構造のデータ部分は、送信される LIN メッセージの構成によって異なります。

Description	Byte count	Bit position							
		7	6	5	4	3	2	1	0
STX	1	0	0	0	0	0	0	1	0
Sequence code	1	0	Sequence number: 0b000...0b111 (0...7)			Parameter byte count: 1 + data length n			
Control code	1	Target interface: 0b10 (2 = LIN)		Command code: 0b000001 (1 = Write data on bus)					
Parameters	LIN message info	1	0: REQ 1: RESP	0	LIN ID: 0...0x3F				
	LIN data bytes	n (0...8)	Data bytes						
Checksum	1	STXバイトを除くこのメッセージのすべてのバイト間のビット単位の XOR 演算。							

- ー 要求フレーム (REQ) : メッセージはデータ要求を行うために使用されます。
- ー 応答フレーム (RESP) : メッセージは LIN スレーブから送信されます。
- ー LIN ID がデフォルトの長さ (LIN 1.x 仕様) を使用する場合、データ長は指定された LIN ID によって異なります (53 ページの Retrieve Frame Length (フレーム長の取得) コマンドを参照)。それ以外の場合は、LIN ID に設定されているデータ・バイト数を含める必要があります。
- ー 送信されたデータ・バイト数が、使用された LIN ID に設定された長さと一致しない場合、エラーが返され、メッセージは LIN インターフェイスに転送されません。
- ー 肯定応答には 1 つのパラメーターがあり、値 0 はエラーがないことを意味します。エラー・コードの詳細については、103 ページの第 3 章 エラー・メッセージを参照してください。

たとえば、標準 ID (LIN 1.x) と 4 データ・バイト (0x20 から 0x2F の範囲の ID) でメッセージを送信するには、Sequence code (シーケンス・コード) は 0x05 である必要があります :

Sequence Code (SC)					Control Code (CC)									
X	Sequence			Param Byte Count		Interface		Command Code						
0	0	0	0	0	1	0	1	1	0	0	0	0	0	1
0	0, 0x0			5, 0x5		2, 0x2		1, 0x1						
5, 0x5					129, 0x81									

<b>Parameters</b>	<p>必要なパラメーターは、LIN メッセージ情報に 1 バイト、LIN メッセージ自体に必要なバイトです。</p> <p>この例では、4 バイト (ID : 0x20; データ・バイト : 0x0A, 0x0B, 0x0C, 0x0D) の standard LIN 要求が送信されます。</p>
<b>Checksum</b>	<p><math>PX = P1 (0x20) \text{ XOR } P2 (0x0A) \text{ XOR } P3 (0x0B) \text{ XOR } P4 (0x0C) \text{ XOR } P5 (0x0D) = 0x20</math></p> <p><math>SC (0x05) \text{ XOR } CC (0x81) \text{ XOR } PX (0x20) = 0xA4</math></p>
<b>Message to send</b>	<p>STX(0x02) – SC(0x05) – CC(0x81) – P1(0x20) – P2(0x0A) – P3(0x0B) – P4(0x0C) – P5(0x0D) – CHK(0xA4)</p>
<b>Positive response</b>	<p>STX(0x02) – SC(0x01) – P1(0x00) – CHK(0x01)</p>

### データのレシーブ (Receive Data)

LIN メッセージが RS-232 インターフェイスに転送されると (40 ページの **コマンド フォワード・マスクを取得** [ Retrieve Forward Mask ]を参照) 、データ部分としてコマンド構造に埋め込まれます。

Description	Byte count	Bit position							
		7	6	5	4	3	2	1	0
STX	1	0	0	0	0	0	0	1	0
Sequence code	1	1 (AR)	Sequence no.: 0b010 (2)			Parameter byte count (1 + data length n)			
Parameters	LIN message info	1	0: REQ 1: RESP	ERR	LIN ID: 0...0x3F or error code				
	LIN data bytes	n (0...8)	Data bytes (LINエラーの場合は0バイト)						
Checksum	1	STXバイトを除くこのメッセージのすべてのバイト間のビット単位の XOR 演算。							

- LIN メッセージが転送されるため、自動応答ビット (AR) が設定されます。
- Sequence number (シーケンス番号) は 2 です。これは、次のデータが LIN インターフェイスから転送されたメッセージであることを意味します。
- REQ はデータの要求を示し、RESP はメッセージがスレーブからのものであることを示します。
- エラー・ビット (ERR) が 1 の場合、LIN エラーを受信します。エラー・フレームは、LIN インターフェイスのフォワード・マスクが LIN エラーを監視および転送するように設定されている場合にのみ受信されます (詳細については、コマンド **フォワード・マスクを設定** [ Set Forward Mask ] 61 ページを参照してください) 。この場合、LIN メッセージ情報にはエラー・コードが含まれています。エラーが発生した場合、データは送信されません。
- 読み取るデータ長は、LIN Sequence code (シーケンス・コード) の Parameter byte count (パラメーター・バイト数) によって異なります。LIN メッセージ情報にはすでに LIN ID が含まれているため、残りのパラメーターはデータ・バイトです。この場合、データ・バイト数はパラメーターの合計から 1 バイトを引いたものです。

たとえば、standard ID 0x21 (フォーマット LIN 1.x、詳細については、**フレーム長 (Frame Length) の取得** (53 ページ) を参照) および、データ・バイト (0x11、0x22、0x33、0x44) で転送されたメッセージを受信すると、次のフォーマットになります。

受信したメッセージ : STX (0x02) – SC (0xA5) – P1 (0xA1) – P2 (0x11) – P3 (0x22) – P4 (0x33) – P5 (0x44)  
– CHK (0x40)

Sequence code (SC = 0xA4)								
Bit	7	6	5	4	3	2	1	0
<b>Description</b>	Auto reply	Sequence number			Parameter byte count			
<b>Contents</b>	1	0	1	0	0	1	0	1

CAN message info (P1 = 0x03)								
Bit	7	6	5	4	3	2	1	0
<b>Description</b>	Frame format	Error	LIN ID					
<b>Contents</b>	1 (RESP)	0	1	0	0	0	0	1

- LIN data 長 = Parameter byte count (パラメーター・バイト数) -1 バイト = 0b0101 - 1 = 4
- LIN ID = 0b100001 = 0x21
- P2: LIN data [0] = 0x11
- P3: LIN data [1] = 0x22
- P4: LIN data [2] = 0x33
- P5: LIN data [3] = 0x44

### 3 エラー・メッセージ (Error Messages)

#### 3.1 RS-232 リターンコード (RS-232 Return Codes)

各要求メッセージへの応答として、コードが返されます。つまり、1 データ・バイトの有効な RS-232 / LIN メッセージです。応答のストラクチャの長さは 4 バイトで、次のように定義されています：

Byte	Description
1	STX (0x02)
2	Sequence code (0x01)
3	Data (error code)
4	Checksum (0x01 XOR error code)

可能なコードを次の表に示します：

Error code	Label	Description
0x00	OK	エラーなし
0x01	INV_CHECKSUM	Checksum (チェックサム) が無効です
0x02	SER_BUFF_OVERRUN	RS-232 通信でのバッファ・オーバーラン
0x03	INV_MASK	無効なフォワード・マスクが渡されました
0x04	NOT_POSSIBLE	指定されたインターフェイスではコマンドは許可されていません
0x05	OUT_OF_RANGE	有効な値の範囲外のパラメーター
0x06	NO_SCHEDULE_ENTRIES	LIN スケジューラーにはエントリーがありません
0x07	INV_ARGUMENT	無効な引数が渡されました
0x08	INV_GROUP	無効なインターフェイスが渡されました
0x09	NOT_SUPPORTED	サポートされていないコマンド
0x0A	FAILED	有効なコマンドを実行すると、エラー状態が発生しました
0x0B	ACTIVE_MASTER	LIN フレームの循環処理 (LIN マスター操作) は現在アクティブです。現時点では、単一フレームの送信は許可されていません。
0x0C	INV_TIMESTAMP	無効なタイムスタンプが渡されました

### 3.2 CAN エラー・コード (CAN Error Codes)

CAN インターフェイス内でエラーが発生し、CAN フォワード・マスクが CAN エラーを RS-232 インターフェイスに送信するように構成されている場合 (詳細については、24 ページの **フォワード・マスクを取得する (Forward Mask)** コマンドを参照してください)、エラーは RS-232 / CAN として返されます。データ長が 4 バイトで合計 8 バイトのメッセージ構造 :

Byte	Description
1	STX (0x02)
2	Sequence code (シーケンス・コード) (0x94)
3	CAN メッセージ情報 (0x24)
4	一般的なエラー情報を含むビットマスク
5	CAN コントローラーからのエラー・キャプチャ・コード (SJA1000)
6	CAN コントローラーの RX エラー・カウンター (SJA1000)
7	CAN コントローラーの TX エラー・カウンター (SJA1000)
8	Checksum (0x01 XOR [Byte 2~7])

Byte 2: Sequence code

Bit	7	6	5	4	3	2	1	0
<b>Description</b>	AR	CAN			Parameter byte count			
<b>Contents</b>	1	0	0	1	0	1	0	0

Byte 3: CAN message info

Bit	7	6	5	4	3	2	1	0
<b>Description</b>	Frame format	Error	CAN Error			Length		
<b>Contents</b>	0	1	1	0	0	1	0	0

CAN エラーに関する一般的な情報を含むビットマスクは、次のいずれかの値になります :

Error code	Value	Description
0x00	CAN_ERR_BUS_OFF	CAN コントローラーはバス・オフ状態です
0x01	CAN_ERR_BUS_PASSIVE	CAN コントローラーはバス・パッシブ状態です
0x04	CAN_ERR_BUS_ACTIVE	CAN コントローラーはアクティブ状態に戻ります (以前の状態はバス・オフまたはバス・パッシブのいずれかでした)
0x08	CAN_ERR_DATA_OVERRUN	受信キューのリードが遅すぎました (メッセージが失われました)
0x10	CAN_ERR_WARNLIMIT	CAN コントローラーの内部エラー・カウンターの 1 つが事前定義された制限に達しました
0x20	CAN_ERR_EWL_EXCEEDED	事前定義されたエラー警告制限を超えました

### 3.3 LIN エラー・コード (LIN Error Codes)

LIN インターフェイス内でエラーが発生し、LIN フォワード・マスクが LIN エラーを RS-232 インターフェイスに送信するように構成されている場合 (詳細については、40 ページのコマンド **フォワード・マスクの取得 (Forward Mask)** を参照)、エラーは RS-232 / LIN として返されます。データ長が 1 バイトで合計 4 バイトのメッセージ構造 :

Byte	Description
1	STX (0x02)
2	Sequence code (0xA1)
3	Data (error code)
4	Checksum (0xA1 XOR error code)

Byte 2: Sequence code

Bit	7	6	5	4	3	2	1	0
<b>Description</b>	AR	LIN			Parameter byte count			
<b>Contents</b>	1	0	1	0	0	0	0	1

Byte 3: Data (error code)

Bit	7	6	5	4	3	2	1	0
<b>Description</b>	Frame format	Error	Error code					
<b>Contents</b>	0	1	×	×	×	×	×	×

Byte 3 の Bit 6 はオン (1) であり、通常 LIN ID に使用されるバイト内の場所が LIN エラー・コードによって占有されていることを示します。

Error code	Description
0x01	バッファ・オーバーフロー (メッセージ送信中に失われた情報)
0x02	無効なストップ・ビットが見つかりました (ビットレートの偏差が原因である可能性がある)
0x03	プロトコル・スタックで無効なブレークが見つかった
0x04	無効な同期バイトが見つかった (おそらくビットレートの偏差が原因です)
0x05	LINID パリティ・ビット 0 のエラー
0x06	LINID パリティ・ビット 1 のエラー
0x07	受信した無効なデータ (ビットエラー)
0x08	無効なフレーム・チェックサム (Frame checksum) を受信した
0x09	フレーム・タイムアウト (スレーブが応答しない、またはフレーム間のギャップが長すぎる)
0x0A	スレーブが応答しない
0x0B	LIN ヘッダーが完成した (ブレーク、同期、ID)
0x0C	LIN ウェイク・アップが検出された
0x0D	受信 IRQ はありませんが、いくつかの XMT IRQ : LIN ラインが電源に短絡している可能性がある
0x0E	スケジューラーの遅延が短すぎる
0x0F	利用不可
0x12	LIN ラインが GND に短絡している可能性がある

## 4 シリアル・トレース分析 (Serial Trace Analysis)

RS-232 インターフェイスからの着信メッセージのトレースがある場合は、次の表を使用して分析できます。

1バイト目 : STX

Bin	Hex	Description
0000 0010	2	モジュール通信のメッセージには必須の “Start Transmission” (送信開始) コード

2バイト目 : Sequence code (シーケンス・コード)

Bin	Hex	Description
0 NNN PPPP	NP	コマンド自動応答
1 000 PPPP	8P	RS-232 インターフェイスから転送されたメッセージ
1 001 PPPP	9P	CAN インターフェイスから転送されたメッセージ
1 010 PPPP	AP	LIN インターフェイスから転送されたメッセージ
1 011 PPPP	BP	モジュール・インターフェイスから転送されたメッセージ

NNN、N = 任意の Sequence number (シーケンス・ナンバー) (0-7、0x0-0x7)

PPPP、P = パラメーターのバイト数 (1~15、0x1~0xF)

3バイト目 : Control (制御) コード

Bin	Hex	Interface	Command name
00 00 0000	00	RS-232	現在のパラメーターによる初期化
00 00 1011	0B	RS-232	ボーレートを取得する
00 01 0101	16	RS-232	RCV エラー・カウンターを取得する
00 01 0111	17	RS-232	XMT エラー・カウンターを取得する
00 01 1000	18	RS-232	すべてのエラー・カウンターを取得する
00 01 1111	1F	RS-232	ボーレートを設定する
00 11 0100	34	RS-232	RCV エラー・カウンターをリセットする
00 11 0101	35	RS-232	XMT エラー・カウンターをリセットする
00 11 0111	37	RS-232	チェックサム・エラー・カウンターをリセットする (Checksum Error Counter)
01 00 0000	40	CAN	現在のパラメーターによる初期化
01 00 0001	41	CAN	データのライト
01 00 1010	4A	CAN	アクティベーション・ステータスの取得する

Bin	Hex	Interface	Command name
01 00 1011	4B	CAN	ビットレートを取得する
01 00 1100	4C	CAN	フォワード・マスクを取得する
01 00 1101	4D	CAN	フィルタ・マスクを取得する
01 00 1110	4E	CAN	フィルタ・コードを取得する
01 00 1111	4F	CAN	LIN メッセージの CAN ID オフセットを取得する
01 01 0110	56	CAN	RCV エラー・カウンターを取得する
01 01 0111	57	CAN	XMT エラー・カウンターを取得する
01 01 1000	58	CAN	すべてのエラー・カウンターを取得する
01 01 1110	5E	CAN	アクティベーション・ステータスの設定する
01 01 1111	5F	CAN	ビットレートを設定する
01 10 0000	60	CAN	フォワード・マスクを設定する
01 10 0001	61	CAN	フィルタ・マスクを設定する
01 10 0010	62	CAN	フィルタ・コードを設定する
01 10 0011	63	CAN	LIN メッセージの CAN ID オフセットを設定する
10 00 0000	80	LIN	現在のパラメーターによる初期化
10 00 0001	81	LIN	データのライト
10 00 1010	8A	LIN	アクティベーション・ステータスの取得する
10 00 1011	8B	LIN	ビットレートを取得する
10 00 1100	8C	LIN	フォワード・マスクを取得する
10 00 1101	8D	LIN	フィルタ・マスクを取得する
10 00 1110	8E	LIN	フィルタ・コードを取得する
10 00 1111	8F	LIN	スレーブ・マスク・ステータスの取得する
10 01 0000	90	LIN	マスター・ステータスの取得する
10 01 0001	91	LIN	LIN バス・ターミネーションの取得する (マスター・ターミネーション)
10 01 0010	92	LIN	メッセージ送信タイムアウトの取得する
10 01 0011	93	LIN	メッセージの再試行を取得する
10 01 0100	94	LIN	スケジューラー・エントリー数の取得する
10 01 0101	95	LIN	スケジューラー・エントリーの取得する
10 01 0110	96	LIN	RCV エラー・カウンターを取得する
10 01 0111	97	LIN	XMT エラー・カウンターを取得する
10 01 1000	98	LIN	すべてのエラー・カウンターを取得する
10 01 1001	99	LIN	スレーブ ID +データ・コンフィグレーション情報を取得する

Bin	Hex	Interface	Command name
10 01 1010	9A	LIN	同期フィールド・ステータスのビットレート認識を取得する
10 01 1011	9B	LIN	フレーム長を取得する
10 01 1100	9C	LIN	チェックサム・タイプ (Checksum Type) を取得する
10 01 1101	9D	LIN	スレーブ・アクティベーション用の CAN ID を取得する
10 01 1110	9E	LIN	アクティベーション・ステータスを設定する
10 01 1111	9F	LIN	ビットレートを設定する
10 10 0000	A0	LIN	フォワード・マスクを設定する
10 10 0001	A1	LIN	フィルタ・マスクを設定する
10 10 0010	A2	LIN	フィルタ・コードを設定する
10 10 0100	A4	LIN	マスター・ステータスを設定する
10 10 0101	A5	LIN	LIN バス・ターミネーションの設定 (マスター・ターミネーション)
10 10 0110	A6	LIN	メッセージ送信タイムアウトを設定する
10 10 0111	A7	LIN	メッセージの再試行を設定する
10 10 1000	A8	LIN	スケジューラー・エントリーを挿入する
10 10 1001	A9	LIN	スレーブ ID +データ・コンフィグレーションを設定する
10 10 1010	AA	LIN	同期フィールドのビットレート認識を構成する
10 10 1011	AB	LIN	フレーム長を設定する
10 10 1100	AC	LIN	チェックサム・タイプ (Checksum Type) の設定
10 10 1101	AD	LIN	スレーブ・アクティベーションの CAN ID を設定します
10 10 1110	AE	LIN	スレーブ・マスク・ステータスの設定
10 11 0010	B2	LIN	メッセージ送信タイムアウトのリセット
10 11 0011	B3	LIN	マスター・スケジューラー・リストを消去する
10 11 0100	B4	LIN	RCV エラー・カウンターをリセットする
10 11 0101	B5	LIN	XMT エラー・カウンターをリセットする
10 11 0110	B6	LIN	ビットエラー・カウンターをリセットする
10 11 0111	B7	LIN	チェックサム・エラー・カウンターをリセットする (Checksum Error Counter)
10 11 1000	B8	LIN	ID チェックサム・エラー・カウンターをリセットする (ID Checksum Error Counter)
10 11 1001	B9	LIN	“Slave not responding” エラー・カウンターをリセットする
10 11 1010	BA	LIN	一貫性のない同期バイト・カウンターをリセットする
10 11 1011	BB	LIN	すべてのエラー・カウンターをリセットする
11 00 0000	C0	Module	ハードウェア・バージョンを読む
11 00 0001	C1	Module	ファームウェア・バージョンを読む

Bin	Hex	Interface	Command name
11 00 0010	C2	Module	モジュールのデフォルト値
11 00 0011	C3	Module	フラッシュの現在のコンフィグレーション
11 00 0100	C4	Module	モジュールをリセットする