

## CAN 通信による自動車内データ収集システム—机上モデルの設計と試作

Design and Trial Manufacture of a CAN-based prototype Vehicular Data Acquisition System

袴田 吉朗\*

小栗 宏和\*\*

Yoshiro HAKAMATA

Hirokazu OGURI

Abstract: The paper summarizes the design and manufacture of a CAN-based prototype Vehicular Data Acquisition System. It consists of PIC micro computers, CAN controllers, CAN transceivers and transmission lines. Voltage and rotation numbers of a DC motor were measured and transmitted via a CAN system correctly to a receiving node. The system hardware configuration and program configuration are precisely taken into account in the paper.

## 1. はじめに

電気電子工学科ではH19年度から学科の象徴物として電気自動車を取り上げ、電気自動車に関するプロジェクトを立ち上げて検討を行ってきた。本論文はその電気自動車プロジェクトの一環として、4年生の卒業研究におけるテーマに取り上げて進めてきた検討結果をまとめたものである。過去3年間は市販のCAN通信ボードを題材として、CAN通信とは何かから初めて、CAN通信の耐雑音特性、伝送の仕組みに関する調査、などの基本的な特性の把握に努めてきた。しかしながら市販のボードを使用している関係上、設計の細部にまで立ち入ることが難しく、なかなか検討のレベルが上がらないと言うのがこれまでの現状であった。

そこで今年度は原点に立ち返り、PICマイコン、CANコントローラ、CANトランシーバ、LCD(液晶ディスプレイ)などを用いて自らCAN通信ノードおよびCAN通信システムを構築し、より踏み込んだ検討を行うことにした。

検討した内容を以下に示す。

- SPI (Serial Peripheral Interface) 通信を用いた PIC マイコン間のデータ通信
- 受信データの LCD への表示方法
- SPI 通信を用いた CAN コントローラ MCP2515 の制御方法
- PIC マイコン、CAN コントローラ MCP2515、CAN トランシーバ MCP2551 を用いた CAN ノードの製作と CAN 通信システムの構築
- PIC マイコンによる電圧およびモータの回転数の取り込み、CAN によるデータ伝送
- 以上を実現するための PIC プログラムの作成

本論文では、以上の検討結果について報告する。

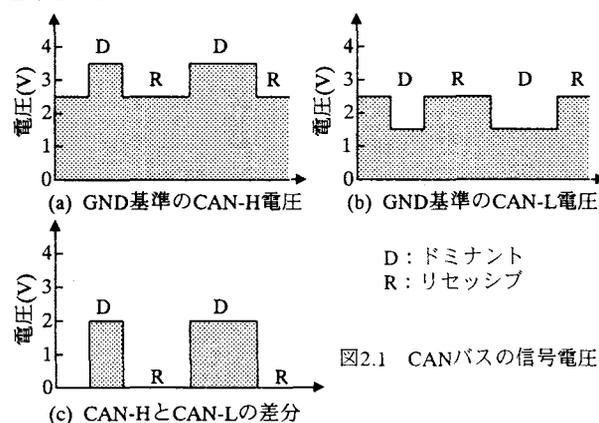
## 2. CAN 通信の概要

## 2.1 伝送路フォーマット

CANバスの信号は、CAN-HおよびCAN-Lと呼ばれる2本の信号線により伝送される。伝送されるデータは、図2.1に

示すように両信号間に電位差があるか、ないかにより決まり

- 電位差がある状態はドミナント (Dominant) ・ データの0
- 電位差がない状態はリセッシブ (Recessive) ・ データの1と呼ばれる。



CANの標準データ・フレームは、以下のようにになっている。

- (1) SOF (Start of Frame), 1ビット, ドミナント
- (2) アービトレーション・フィールド (12ビット)  
このフィールドは11ビットのメッセージIDと1ビットのRTR1ビットからなる。RTRがドミナントのときデータ・フレームを表す。
- (3) 制御フィールド(6ビット)  
標準データフォーマットは最初のIEDビット(1ビット)がドミナントであることにより識別される。DLCフィールド(4ビット)は0~8バイトのデータ長を表す。
- (4) データ・フィールド(最大8バイト)
- (5) CRCフィールド(16ビット)  
CRCフィールドは、15ビットのCRCシーケンス(生成多項式 $x^{15}+x^{14}+x^{10}+x^8+x^7+x^4+x^3+1$ )と、1ビットのCRCデリミタからなる。
- (6) アクノリッジ・フィールド(2ビット)

送信ノードは最初のACKスロットにリセッシブを送信する。エラーのないフレームを受信したノードは、このACKスロットにドミナントを入れて返送する。これにより送信ノードは、フレームが正しく受信されたことを認識する。ACKデリミタはリセッシブである。

2011年3月2日受理

\* 理工学部 電気電子工学科

\*\*理工学部 電気電子情報工学科 4年生

- (7) EOF(End of Frame), 7ビット, リセッシブ  
 (8) IFS (Interframe Space), 3ビット, リセッシブ

他にリモート・フレーム (RTR がリセッシブ), エラー・フレームおよびオーバーロード・フレームがある。

なお, 伝送路符号のフォーマットは NRZ (Non Return to Zero) フォーマットである。このため受信部における PLL がタイミング消失しないようにするために, 同一信号レベルが5ビット以上連続すると, その5ビット後にレベルを反転したビットを強制的に1ビット挿入し, レベルの変化を起こさせるビット・スタッフィングが行われる。したがって実際のフレームにおけるビットの並びは, 送信されるデータの内容に従って以上に述べたフレーム構成とは微妙に異なってくる。

## 2.2 CAN における伝送路アクセス制御

CAN は分散システムであり CSMA/CA 方式によって伝送路にアクセスする。具体的には, CSMA 方式+非破壊のビット単位アービトラージョン (NBA 方式: Non-Destructive Bitwise Arbitration) によって信号衝突を防いでいる。

以下のような手順で伝送路にアクセスする。

- ① バスのキャリアをセンスし, バスが空いていれば, バスにつながっている全てのノードが送信を開始することができる。
- ② 複数ノードが同時に送信を開始した場合の信号衝突を回避するために, 各ノードはアービトラージョン・フィールドを送出する期間, バスのレベルをモニターする。
- ③ ドミナントは文字通りリセッシブを上書きするので, 自ノードが送信した信号と検出したバスのレベルが異なる場合には調停負けと判断し, 受信動作に移るようになっている。
- ④ したがって ID 値が小さい方が高優先度になる。
- ⑤ CAN では, 送信される各メッセージが, メッセージ ID によって一意に識別される。その結果, メッセージは基本的にどのノードにおいても受信でき, メッセージを受け付けるかどうかはコントローラのみによって決定される (受信側選択システム)。例えば電気自動車の場合には, モータの回転数を, データ収集ノードで収集し, 一方で同時に同じデータをメータに表示して使用するような使い方ができるようになる。

## 2.3 CAN 通信におけるビット構成とボーレート

CANバス上の全ノードは同一のボーレートを有することが望ましい。しかし全ノードは必ずしも同一のマスタークロックを持っている訳ではないので, 受信部は微妙に異なるクロック周波数を有するデータの受信に対応する必要がある。このためボーレートは表3.1に示したBRP (ボーレート・プリスケール) およびビットを構成する各セグメントのTQ (Time Quanta) を適切に設定することにより調整できるようになっている。

コントローラのクロック信号 ( $F_{osc}$ ) を分周して作成され, 送信, 受信に使用される時間単位がTQ (Time Quanta) である。CANビットの1ビットは, 図2.3に示すように4つのセグメントに分かれている。データを取り込むサンプリング点はフェーズ1とフェ

ーズ2の間にある (1点サンプリング時)。

同期セグメントSYNCは公称ビット時間NBTにおける最初のセグメントであり, ビットがアイドルからドミナントに変化するビットエッジを検出してデータの受信に移行するために使用される。このセグメントの長さは1TQ (固定) である。

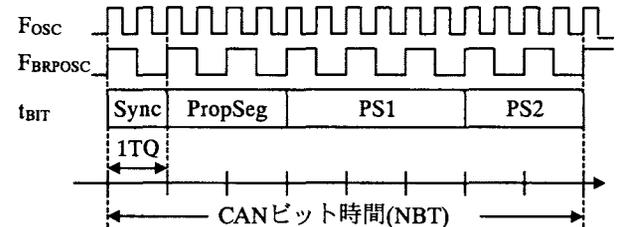


図2.2 CANにおけるビット時間

他の3つのセグメントの長さはプログラム可能である。ボーレートは式 (3.1) により表される。

$$\text{ボーレート} = \frac{F_{osc}}{2 \cdot (BRP + 1) \cdot TQ \text{構成数}} \quad (3.1)$$

$$\begin{aligned} TQ \text{構成数} = & \text{プロパゲーションビット数}(PRSEG) \\ & + \text{フェーズ1ビット数}(PHSEG1) \\ & + \text{フェーズ2ビット数}(PHSEG2) \\ & + 1 \text{ (SYNC)} \end{aligned}$$

今回使用したパラメータは, BRP=4, PRSEG=2, PHSEG1=6, PHSEG2=5であり, TQ構成数=16である。

また受信信号と自身のクロックの位相ずれを補正する値であるSJW (Synchronized Jump Width) の値は1とした。

## 2.4 CAN コントローラ MCP2515 のレジスタ

本検討ではCANコントローラにMCP2515を使用したので, そのレジスタについて説明する。

- (1) コンフィギュレーション・レジスタ (CNF1, CNF2, CNF3)  
表3.1~表3.3に示す。CANビット・タイムの設定, CANフレームのボーレートの設定を行う。

- (2) コントロール・レジスタ (CANCTRL)

MCP2515のオペレーション・モードの設定, クロック出力ピンの設定などを行う (表3.4)。

- (3) ステータス・レジスタ (CANSTAT)

各種レジスタのステータスを示す。

- (4) 割り込み関係レジスタ (CANINTE, CANINTF)

割り込み許可および割り込み要因を示すレジスタである。

- (5) 送信バッファ

TXB0, TXB1 およびTXB2の三つの送信バッファがあり, 各々14バイトの領域を有する。図2.3に示すように送信バッファごとにコントロール・レジスタ (TXBnCTRL), メッセージの格納エリア (TXBnSIDx, TXBnEIDx), データ長レジスタ (TXBnDLC) 及びデータ本体 (TXBnD0~TXBnD7) からなる。レジスタにおける各ビットの内容を表3.6~表3.9に示す。

- (6) 受信バッファ

RXB0 およびRXB1の二つの受信バッファがあり, 各々14バ

イトの領域を有する。図 2.3 に示すように受信バッファごとにコントロール・レジスタ (RXBnCTRL), メッセージの格納エリア (RXBnSIDx, RXBnEIDx), データ長レジスタ (RXBnDLC) およびデータ本体 (RXBnD0~RXBnD7) からなる。バッファの数は一つ少ないが, 送信バッファと対応している。

表 3.10~表 3.12 にビットの詳細を示す。

(7) その他

マスク・フィルタ・レジスタおよびエラーカウンタなどがある。

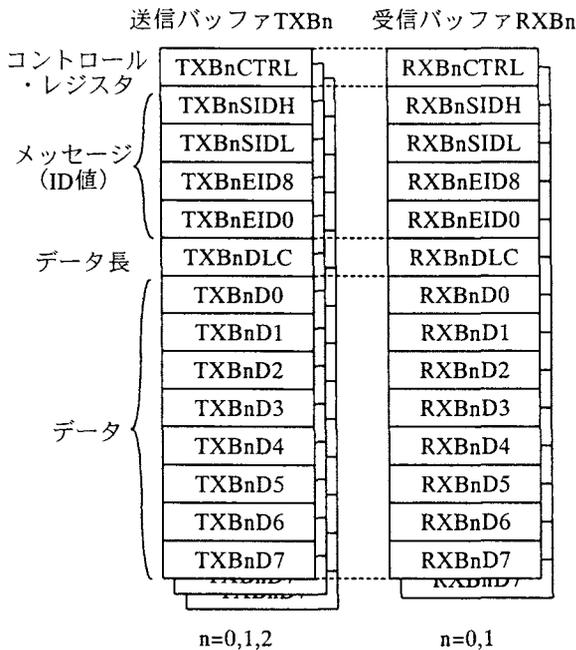


図 2.3 送受信バッファのレジスタ構成

2.5 SPI通信によるMCP2515におけるレジスタの設定

CAN通信を行うには2.4に示したレジスタを適切に設定する必要があるが, そのためにSPI通信を用いる。SPI通信では, /CSB, MOSI, MISOおよびSCLKの4本の信号線とGNDを用いてCPUと通信する。以下に示すコマンドを用いる。

(1) レジスタ・ライト・コマンド

図2.4(a)に示す。スレーブセレクト/CSBをLに下げ, マスターノードから命令0x02に引き続き, レジスタアドレス, データの順番に出力する。データ送信後/CSBをHに戻す。

(2) レジスタ・リード・コマンド

図2.4(b)に示す。スレーブセレクト/CSBをLに下げ, マスターノードから命令0x03に引き続き, レジスタアドレスを出力する。レジスタアドレスを出力後, データを入力する。データ送信後/CSBをHに戻す。

(3) ステータス・リード, 受信ステータス・リード・コマンド

図2.4(c)に示す。命令コード0xA0がステータス・リード・コマンドである。その意味を表3.13に示す。命令コード0xB0は受信ステータス・リード・コマンドである (表3.14)。これは受信メッセージの有無, メッセージ・タイプおよびフィルタ・マッチの状態を調べるコマンドである。

(4) ビット・モディファイ・コマンド

指定レジスタの特定ビットのみを変更するコマンドである。割り込み関係のフラグのように, 8ビットの中で特定のビットだけをセットまたはリセットするとき使用する。

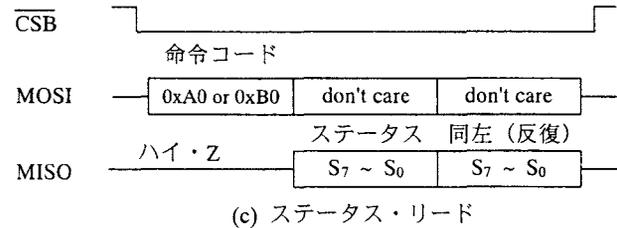
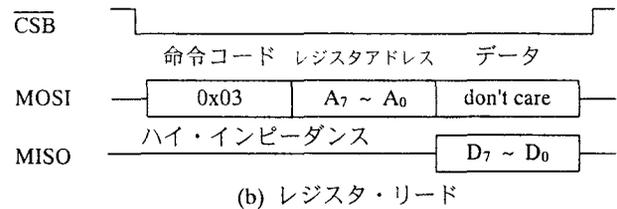
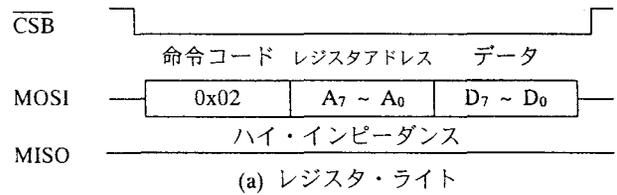


図 2.4 SPIコマンドによるレジスタのリード/ライト

3. 試作した自動車内データ収集システムの概要

試作したシステムの構成を図 3.1 に示す。送信部と受信部各 1 台および CAN 伝送路からなる机上モデルである。送信部では, 電圧を AD 変換して PIC に取り込む回路と, DC モータを回転させ, そのモータから出力される回転数を表すパルス PIC のキャプチャ機能を用いて取り込む回路がある。これらの測定したデータを, CAN 通信を用いて受信部に伝送している。

受信部では CAN 通信によるデータを受信後, 必要な演算を行った結果を LCD に表示させる。また PIC 内の EEPROM にデータを書き込むようにして, 受信データを数値として確認できるようにしている。

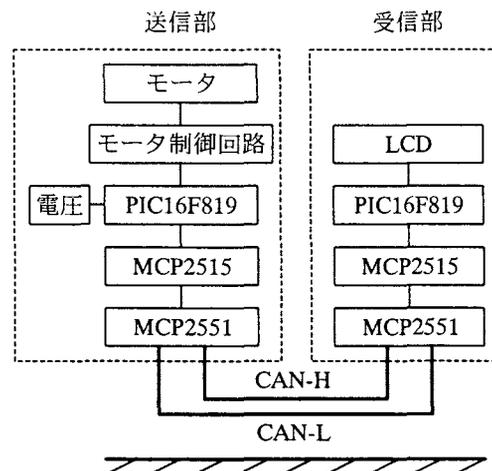


図 3.1 試作システムのブロック構成

システムにおける構成要素の概要を以下に示す。

- ・ PIC16F819・・・18ピンのPICであり、AD変換機能、SPIインタフェースを有するため採用した。測定データをMCP2515の送信レジスタに書き込む、受信データの読み取りを行う、LCDに表示する、などのシステム全体の制御を行う。
- ・ MCP2515・・・CANコントローラ。送信データをCANのメッセージ形式に変換してMCP2551に送る。また、ビット・スタフピングを行って受信部におけるPLLの動作の確実性を高める。受信部では、受信データをMCP2515自身が持つレジスタに格納する、エラーチェックを行うなどの処理を司る。
- ・ MCP2551・・・CANトランシーバ。MCP2515から送信されるデータを差動信号に変換してCAN伝送路に送出する。これにより高いノイズ耐性を得る。受信部では、CANバスの信号をCANのメッセージ形式に戻し、MCP2515に転送する。
- ・ CAN伝送路・・・約20cm長の2本のピニール線であり、両端が100Ωの抵抗により終端されている。

#### 4. 試作システムのハードウェア構成

##### 4.1 PICマイコンにおけるポートの設定

PICマイコンには、SPIインタフェースおよびAD変換機能を有する18ピンのPIC16F819を使用している。

表4.1にPIC16F819におけるポートの割り当てを示す。空白の欄は未使用のポートである。

表4.1 PIC16F819におけるポートの割り当て

PICのポート	送信部	受信部
RA0	電圧入力 (AD変換)	LCDのDB4
RA1		LCDのDB5
RA2	モータのF/R IN	LCDのDB6
RA3	モータのBRAKE	LCDのDB7
RA4	モータのRUN	
RA5	リセット	リセット
RA6		
RA7		
RB0		LCDのRS
RB1	MISO	MOSI
RB2	MOSI	MISO
RB3	モータの回転数入力	LCDのE(STB)
RB4	SCLK	SCLK
RB5	CSB	CSB
RB6		LCDのR/W
RB7		

##### 4.2 PICマイコンにおけるクロックの設定

内部の8MHzクロックを使用している。設定を以下に示す。

$$\text{OSCCON} = 0x74$$

##### 4.3 AD変換による電圧の収集

##### (1) AD変換の設定

1MΩの可変抵抗により電源電圧を分圧してアナログポートに入力し、その電圧の測定を行っている。

またPIC16F819のAD変換機能を使用するために、レジスタADCON0およびADCON1を以下のように設定した。

$$\text{ADCON0} = 0xC1$$

$$\text{ADCON1} = 0x4E$$

内容は以下の通りである。

- ・ AD変換のクロック・・・内部RC発振による
- ・ AD変換の結果を左詰め
- ・ アナログ入力ポートをRA0 (AN0) に設定、電圧の基準V<sub>REF+</sub>を電源電圧V<sub>DD</sub> (5V) に設定

なお、ADCON0レジスタにおけるビット0は、AD変換の開始指令であり、AD変換を開始する度に1に設定する。

AD変換の結果は10ビット長で出力され、レジスタADRESHおよびADRESLの各1バイトのレジスタに格納される。格納方法を左詰めとし、ADRESHの8ビットのみを結果として使用している。8ビット使用の場合、変換結果は0~255となる。今回の設定では基準電圧(最大測定電圧)V<sub>REF+</sub>をV<sub>DD</sub>に設定しており、0~5Vが0~255にAD変換されて出力される。そのため、受信部においてプログラムを用いて変換結果を5で除算し、LCDに表示する時に桁をずらして電圧値に変換した。

##### 4.4 キャプチャによるモータ回転数の取り込み

##### (1) モータの回転の制御

直流モータおよびモータドライバには日本サーボ株式会社製のSERVEX FED 8P30S-D3セットを使用した。

制御回路にPICを接続し、簡単な制御と回転数の取り込みを行った。

- ・ VS・・・10kΩの可変抵抗を接続し手で速度を制御している。
- ・ SPEED OUT・・・モータ1回転あたり48個のパルスが出力される。このパルスをPICのポートRB3に入力し、そのパルス間隔から速度を求めている。
- ・ RUN・・・プログラムでLに設定し、モータの回転開始の指示を出している。

##### (2) モータ回転数の取り込み

SPEED OUTから出力されるパルスは、回転速度が速くなるとパルスの数が増え、パルス間隔は狭まる。逆に回転速度が遅くなるとパルス数は減り、パルス間隔が広くなる。

そこでパルス間隔を測定するのに、PICのCCP (Compare/Capture/PWMの略)におけるキャプチャ機能を用いることにした。キャプチャ機能ではCCPxピン(ここではRB3)の信号をトリガにして、トリガ時点におけるタイマ1の値(TMR1H, TMR1L)をキャプチャレジスタ(CCPRxH, CCPRxL)に記憶するようになっている。

使用したタイマ1制御用レジスタTICONおよびCCP制御レジスタCCP1CONの設定を以下に示す。

$$\text{TICON} = 0x31$$

CCP1CON = 0x07

その意味は以下の通りである。

- ・ タイマ1のプリスケール値 = 1:8
- ・ タイマ1の動作制御を行う
- ・ 16回目の立ち上がりエッジ毎にキャプチャモードを動作  
なお、モータから出力される回転数を表すパルス(48ppr)には大きなジッタが観測された。しかしオシロスコープで観測した結果では、48パルス毎であれば同期がとれていることも確認できた。このためプログラムではPIRレジスタにおけるキャプチャフラグCCP1IFが1になったか否かを確認し、キャプチャを3回行った時(16×3=48パルス目)のキャプチャデータを測定データとして使用するようにした。

(3) 測定データの回転数への変換

キャプチャにより測定したデータを、回転計を使用して実測したモータの回転数と比較することにより、測定データを回転数に変換する方法を検討した。回転計はレーザ光により回転対象物にパルスを照射して、その反射光を検出して回転数を測定するタイプのものである。

図4.1に回転数の実測値と測定データの平均値(20回)との関係を示す。測定範囲はモータの速度制御可能な範囲を考慮して200rpm~1900rpmとした。

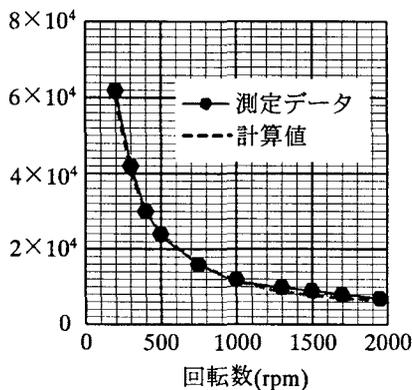


図4.1 測定したデータと回転数の関係

図4.1より測定データyは回転数xに反比例している、すなわち  $xy=a$  なる関係があると定数aを求めることにした。最小自乗法を用いて定数aの値を求め  $a=11998720$  が適していると判断した。波線は理論値である。

回転数が高くなると実測値と理論値の誤差が大きくなっている。これがシステム全体を通して考えたときに大きな誤差を発生する原因の一つになっていると考えられる。

4.5 SPI通信の設定

PIC16F819のSPIインタフェースを用いてMCP2515を制御するためのレジスタの設定を以下に示す。

SSPCON. SSPEN=1・・・シリアルポートを動作させてSCK,  
SDO, SDI, SSをシリアルポートにする  
SSPCON. SSPM0・・・SPIマスターに設定、クロックF<sub>osc</sub>/16  
SSPCON. CKP・・・送信は立ち上がりエッジ、

受信は立下がりエッジ、アイドルは

SSPSTAT. SMP=1・・・データはSCKの立ち上がりで送信

4.6 MCP2515の設定

2.4で示したレジスタを下記のように設定して使用した。

- ・ CNF1 = 0x04
- ・ CNF2 = 0xB1
- ・ CNF3 = 0x05

これより BRP=4, TQ=10/F<sub>osc</sub>, PHSEG1=6, PHSEG2=5, PRSEG=1である。したがってTQ構成数=16になる。F<sub>osc</sub>=20MHzとしたのでボーレートは125 kbaudである。なお1ビット中におけるサンプリングは1回だけである。

- ・ CANCTRL = 0x0E

ノーマル・モードであり、1回だけ送信するワンショット・モードである。またCLKOUTピンを使用する設定である。

- ・ TXB0CTRL = 0x03

転送バッファの優先度は最高レベルである。

- ・ TXB0DLC = 0x02

回転数データは2バイト長のため、データ長を2としている。

AD変換時にはデータ長=1である。

- ・ TXB0SIDH = 0x00
- ・ TXB0SIDL = 0x00
- ・ TXB0EIDH = 0x00
- ・ TXB0EIDL = 0x00

IDを11ビット長の標準IDに設定してある。

4.7 トランシーバMCP2551の設定

MCP2551はCAN信号とCANの伝送路レベルの差動信号を相互変換するインタフェースである。表4.2にピン配置とその機能を示す。R<sub>S</sub>はスロープ(スルーレート)制御であり、この端子とGND間に接続する抵抗値によってスルーレートが決まる。仕様では10kΩ時に23V/μsになっている。今回はR<sub>S</sub>をGNDに接続し、スルーレートが最大となるようにした。

表4.2 MCP2551のピン出力

番号	ピンの名前	ピンの機能
1	TXD	送信データ(入力)
2	V <sub>SS</sub>	グラウンド
3	V <sub>DD</sub>	電源電圧
4	RXD	受信データ(出力)
5	V <sub>REF</sub>	基準出力電圧 (V <sub>DD</sub> /2)
6	CAN-L	CANローレベル電圧 I/O
7	CAN-H	CANハイレベル電圧 I/O
8	R <sub>S</sub>	スロープ制御(入力)

4.8 LCDへの表示

測定したデータを受信部において表示させるために16桁2行のLCD(SC1602B)を用いた。ここではこのLCDの制御方法を示す。ピン番号と信号内容を表4.3に示す。

表 4.4 LCD におけるピン番号と信号

ピン	記号	信号内容	備考
1	Vdd	電源 5V	
2	Vss	グランド	
3	Vo	コントラスト調整	
4	RS	コマンド/データ 選択	H=データ L=コマンド
5	R/W	リード/ライト選択	H=リード L=ライト
6	E(STB)	イネーブル	Hでストロープ
7	DB0	データ下位	8ビットモード で使用
8	DB1	(4ビットモード時は 使用しない、開放にす る)	
9	DB2		
10	DB3		
11	DB4	データ上位	
12	DB5	(4ビットモードで使 用する)	
13	DB6		
14	DB7		

次に、LCD 出力時における制御のタイミングチャートを図 4.3 に示す。リード/ライトをライト(L)、RS が H になっている状態で、E(STB)を H にした時のデータ(DB)の値が LCD に表示される。データは 16 進数で転送され、LCD には ASCII コードで変換されたデータが表示される。

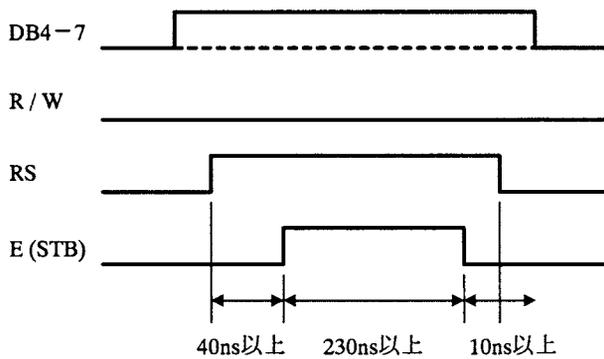


図 4.3 LCD 制御のタイミングチャート

4.7 MCP2515 のクロック回路

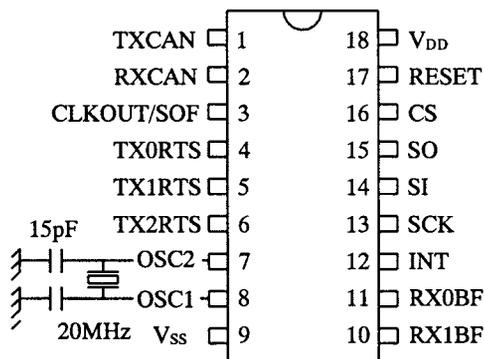


図 4.4 MCP2515 のピン配置

MCP2515 のピン配置を図 4.4 に示す。ピン 8 およびピン 7 間

には内部で CMOS インバータと高抵抗が接続されている。したがってここに水晶振動子とコンデンサを図のように接続し、3 点接続型のコルピッツ形発振器を構成してクロック源とした。この回路は PIC マイコンの場合と全く同様である。

5. 処理プログラムの構成

5.1 概略フローチャート

例として回転数を測定する処理プログラムの送信部における概略フローチャートを図 5.1(a) に示す。

送信部では、SPI の設定、MCP2515 の設定およびキャプチャの設定を行った後に回転数をキャプチャし、SPI コマンドを用いてデータを送信バッファに書き込み、また送信要求を出すことにより CAN バスからデータを送信するようになっている。

受信部における概略フローチャートを図 5.1(b) に示す。受信部では、SPI の設定、MCP2515 の設定および LCD の設定を行った後に、SPI の受信ステータス・リード・コマンドを用いて受信データの有無を確認する。受信データがあるとそのデータを受信バッファに読み込み、データを回転数に変換する演算を行った後、LCD に表示するようになっている。

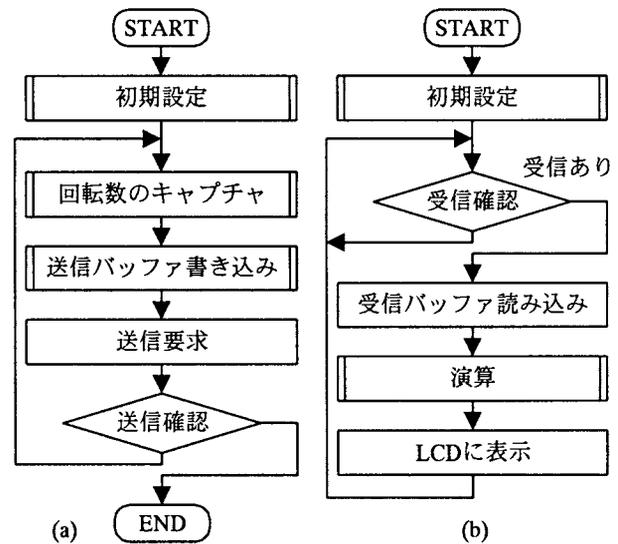


図 5.1 概略フローチャート

5.2 SPI 通信のコマンド

SPI 通信における送受信サブルーチン (送受信に共通) のプログラムを図 5.2 に示す。

W レジスタを介してバッファ SSPBUF にデータが入力され

```

SPI_TX
SPI_RX
MOVWF SSPBUF
BSF STATUS, RP0
BTFSS SSPSTAT, BF
GOTO $-1
BCF STATUS, RP0
MOVF SSPBUF, W
MOVWF RXDATA
RETURN
    
```

図 5.2 SPI 通信のコマンドのサブルーチン

てくるので、SSPBUF がフルになったか否かをバッファ SSPATAT における BF ビットの内容をセンスして監視する。BF=1 になったときに、SSPBUF の内容を W レジスタを経由して RXDATA に取り込む。

### 5.3 AD 変換

図 5.3 にプログラムを示す。4.3 に示した ADCON1 レジスタに値を設定後、図 5.3 のサブルーチンをコールして AD 変換を開始する。その後レジスタ ADCON0 における GO ビット (ビット 0) を監視して、このビットが 0 になってからデータ ADRESH を変数 TXDATA に取り込むようにしている。

```
VOLGET
    MOVLW 0xC1
    MOVWF ADCON0
    CALL TIME_100US
    BSF ADCON0, GO
    BTFSC ADCON0, GO
    GOTO $-1
    MOVF ADRESH, W
    MOVWF TXDATA
    RETURN
```

図 5.3 AD 変換のプログラム

### 5.4 回転数のキャプチャ

図 5.4 にサブルーチンを示す。4.4 に示した CCP1CON レジスタに値を設定後、割り込み処理レジスタ PIR1 におけるフラグ CCP1IF を監視しキャプチャを開始するパルスを検出する。その後タイマ 1 のカウンタをリセットし、入力されるパルスを監視し続けて 16 個のパルスが入力される度にキャプチャし、これを 3 回繰り返した後に CCPR1H および CCPR1L のデータを取り込んで処理を行っている。

```
CAPTURE
    BTFSS PIR1, CCP1F
    GOTO $-1
    BCF PIR1, CCP1F;タイマリセット
    CLRF TMR1H
    CLRF TMR1L
    BTFSS PIR1, CCP1F;最初の16パルス測定
    GOTO $-1
    BCF PIR1, CCP1F
    BTFSS PIR1, CCP1F;次の16パルス測定
    GOTO $-1
    BCF PIR1, CCP1F
    BTFSS PIR1, CCP1F;次の16パルス測定
    GOTO $-1
    BCF PIR1, CCP1F
    RETURN
```

図 5.4 回転数のキャプチャを行うサブルーチン

### 5.5 LCD の制御

文献 [5] を参考にして 4 ビットモードにおけるライブラリを作成し使用した。

### 5.6 除算サブルーチン

16 バイトの整数除算のプログラムは、Microchip 社のサイトからダウンロードした Math ライブラリを使用した。電圧測定において測定データを 5 で割るために使用している。

また CAN 通信で伝送されたデータを受信部において回転数に変換するためには、32 ビットの除算を行う必要がある。このサブルーチンには、インターネットからダウンロードしたライブラリを使用させて頂いた [6]。

### 5.7 その他のプログラム

16 ビットの 16 進数を 5 桁の 10 進数に変換するために、Microchip 社のライブラリを使用した。また PIC の EEPROM にデータを書き込んで確認するためのプログラムを作成した。

## 6. データ収集結果

### 6.1 伝送路波形の測定結果

今回検討したシステムは、図 3.1 に示したように送信部および受信部が各 1 台ずつ対向している構成である。伝送路は約 20cm 長の 2 本のビニール線、終端抵抗は 100Ω である。

伝送路波形の測定例を図 6.1 に示す。CAN-H および CAN-L にはサグが観測されているが、オシロスコープの演算機能を用いて観測した差動出力 TXCAN は理想的な波形になっている。論理振幅は 2V、ボーレートは 125kbaud である。図 6.2 は図 6.1 に対応する MCP2515 出力である。電圧軸は 5V/div、時間軸は 10μs/div である。両者の対応関係は良好である。

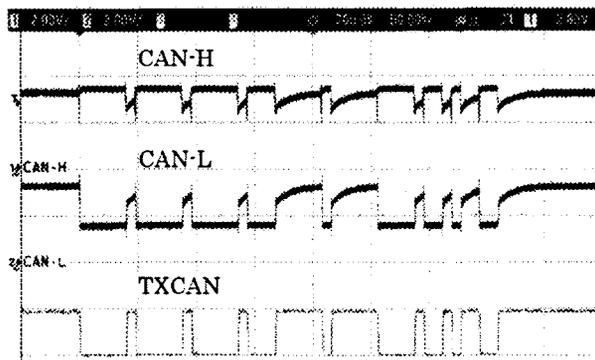


図 6.1 伝送路波形の測定例

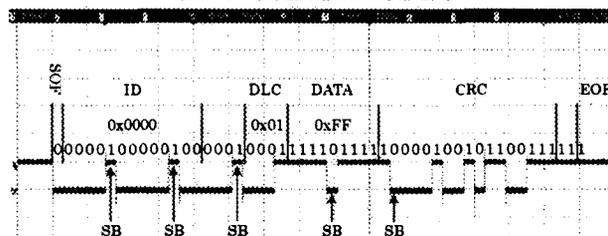


図 6.2 図 6.1 に対応する MCP2515 出力の測定例

### 6.2 CAN 通信を介した電圧の測定結果

PIC のアナログポートに入力される電圧を変化させながら、AD 変換、CAN による伝送、受信部における 5 による除算を介し受信部の LCD に表示させて結果を測定した。図 6.3 に測定結

果を示す。入力電圧、表示電圧の関係から、実用的に問題のないAD変換、CANによる伝送および表示を確認した。

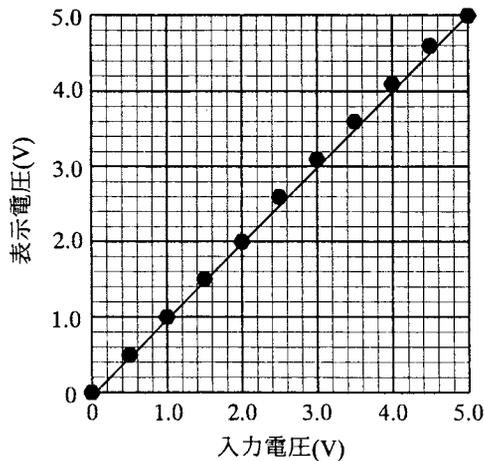


図6.3 CANを介した電圧の測定結果

### 6.3 CAN通信を介したモータ回転数の測定結果

電圧と同様にCANを介して測定したモータにおける回転数の測定結果を図6.4に示す。受信部のPICにおいて、定数aの値をPICでキャプチャした測定データで割ることにより回転数に変換している。破線は理論値であるが、回転数が高くなると誤差が大きくなる傾向が表れている。

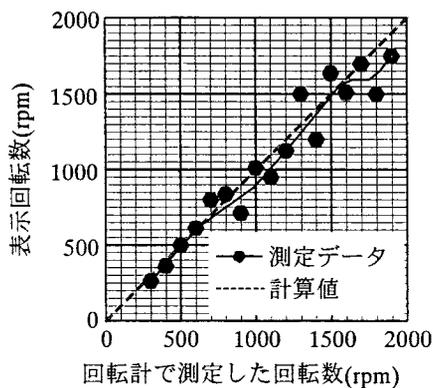


図6.4 CANを介して測定した回転数の測定結果

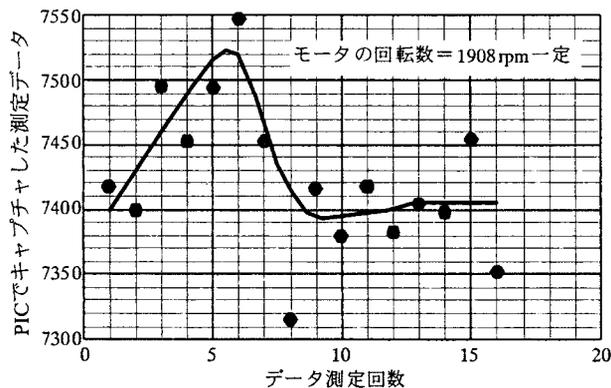


図6.5 EEPROMによるキャプチャデータの測定結果

システム全体として符号誤り率特性を測定し、定量的に評価

を行う必要はあるが、CRCエラーは発生していないので現時点でCAN通信と受信部における計算には大きな誤りがあるようには見えない。したがって誤差の原因はキャプチャ部分にあると考えられる。そこで、同一速度でモータを回転させキャプチャしたデータをEEPROMに書き込み、データの検証を行ってみた。16回連続して測定した結果を図6.5に示す。測定は回転数の最高値(1908rpm)で行った。図6.5から分かる通り、キャプチャ部分で大きな誤差が出ていた。これが図6.4の測定で発生している誤差の大きな要因になっていると思われる。この誤差の軽減が今後の課題である。

## 7. むすび

本論文では、CAN通信システムを構築するために必要となるCAN通信の概要や設計項目を整理し、CANコントローラMCP2515の具体的な設定項目を明らかにした。

またPICマイコンにおけるAD変換器の使用法、パルスのキャプチャの設定方法、SPI通信の設定方法、LCDへの表示方法を示し、これに基づいて送信部と受信部各1台およびCAN伝送路からなるCAN通信を適用したシステムの机上モデルを試作した。また雑音や干渉のない環境下においてシステムの伝送特性を測定し評価した。これにより以下の結果が得られた。

- (1) 送信部において測定した電圧を、AD変換し、CAN通信により受信部に伝送し、その受信データをLCDに表示することにより実用的に問題のない結果が得られた。
- (2) モータの回転数についても同様な測定を行った。しかしながらモータの回転数が高くなると、大きな誤差が発生することが分かった。その原因の一つとしてモータの回転数に対応するパルスの取り込みにバラツキがあり、この改善の必要性が明らかになった。

今回の検討ではシステムを設計・試作し、その特性の測定を行うのが精一杯であり、プログラムの細部にまで立ち入って十分に検討することができなかった。今後未検討の項目についての検討を進め、システムの完成を目指していきたい。

〔謝辞〕

電気電子工学科恩田教授にはDCモータを提供して頂いた。また電気電子工学科服部講師には、モータの回転計を借用させて頂いた。深く感謝いたします。

〔参考文献〕

- (1) MCP2515 SPI スタンドアロン CAN コントローラ, Microchip
- (2) 小川晃, "PICCAN テクニカルガイド", Micro Application Laboratory
- (3) 中尾司, "動かして学ぶCAN通信", CQ出版社, 2010年
- (4) PIC16F818/819 Data Sheet 18/20-Pin Enhanced Flash Micro controllers with nano Watt Technology, Microchip
- (5) 後閑哲也, "改訂版 電子工作のためのPIC16F活用ガイドブック", 技術評論社, 2007年
- (6) "ELECTRONICS SHELVES (趣味の電気工作)" <http://orange.zero.jp/electronics/>

表 3.1 CNF1 (アドレス : 0x2A)

b <sub>7</sub>	SJW1	同期ジャンプ長	11= 4×TQ, 10= 3×TQ
b <sub>6</sub>	SJW0		01= 2×TQ, 00= 1×TQ
b <sub>5</sub>	BRP5	ポーレート・プリスケアラ	ポーレート・プリスケアラ値設定 TQ=(BRP+1)/Fosc
b <sub>4</sub>	BRP4		
b <sub>3</sub>	BRP3		
b <sub>2</sub>	BRP2		
b <sub>1</sub>	BRP1		
b <sub>0</sub>	BRP0		

表 3.2 CNF2 (アドレス : 0x29)

b <sub>7</sub>	BTLMODE	PS2 時間長ビット	1= PS2 の長さは PHSEG22-PHSEG20 で定義 0= PS2 の長さは PS1 と IPT(2TQ) より大きい
b <sub>6</sub>	SAM		1= 3 回サンプリング 0= 1 回だけサンプリング
b <sub>5</sub>	PHSEG12	PH1 ビット長	PH1 のビット長 = (PHSEG1+1)×TQ
b <sub>4</sub>	PHSEG11		
b <sub>3</sub>	PHSEG10		
b <sub>2</sub>	PHSEG2	PropSeg ビット長	プロパゲーション・セグメントのビット長 = (PRSEG+1)×TQ
b <sub>1</sub>	PHSEG1		
b <sub>0</sub>	PHSEG0		

表 3.3 CNF3 (アドレス : 0x28)

b <sub>7</sub>	SOF	SOF ビン設定	1= CLKOUT ビンを SOF ビンに設定 0= CLKOUT ビンを クロック出力に設定
b <sub>6</sub>	WAKFIL	ウェイクアップ・フィルタ	1= フィルタを有効 0= フィルタを無効
b <sub>5</sub> ~ b <sub>3</sub>		未実装	
b <sub>2</sub>	PHSEG22	PH2 ビット長	PH2 のビット長を設定 (最小設定値 2) (PRSEG2+1)×TQ
b <sub>1</sub>	PHSEG21		
b <sub>0</sub>	PHSEG20		

表 3.5 CANINTF (アドレス : 0x2C)

		割込要因	
b <sub>7</sub>	MERR	メッセージ・エラー	1=割込発生中 (ソフトでクリア必要)
b <sub>6</sub>	WAKIF	ウェイクアップ	
b <sub>5</sub>	ERRIF	エラー	1=割込未発生
b <sub>4</sub>	TX2IF	送信バッファ 2 空	
b <sub>3</sub>	TX1IF	送信バッファ 1 空	
b <sub>2</sub>	TX0IF	送信バッファ 0 空	
b <sub>1</sub>	RX1IF	送信バッファ 1 フル	
b <sub>0</sub>	RX0IF	送信バッファ 0 フル	

表 3.4 CANCTRL レジスタ(アドレス : 0x×F)

b <sub>7</sub>	REQOP2	オペレーション・モード設定要求	000= ノーマル・モード
b <sub>6</sub>	REQOP1		001= スリープ・モード
b <sub>5</sub>	REQOP0		010= ループアップ 011= リスン・オンリ 100= コンフィグ・モード
b <sub>4</sub>	ABAT	全送信中断	1= すべて送信を中断 0=送信中断要求を終了
b <sub>3</sub>	OSM	ワンショット・モード	1= ワンショット・モード 0= 非ワンショット・モード
b <sub>2</sub>	CLKEN	CLKOUT ビン許可	1= CLKOUT ビン使用する 0=使用しない(HI-Z)
b <sub>1</sub>	CLKPRE1	CLKOUT ビンプリスケアラ	クロックの分周比を設定 00=Fosc/1, 01=Fosc/2 10=Fosc/4, 11=Fosc/8
b <sub>0</sub>	CLKPRE0		

表 3.6 TXBnCTRL レジスタ (アドレス : 0x30, 0x40, 0x50)

b <sub>7</sub>	-	未実装	
b <sub>6</sub>	ABTF	メッセージ・アボート	1= メッセージ送信は中断 0= メッセージ送信成功
b <sub>5</sub>	MLOA	メッセージ喪失	1= 調停負け, メッセージ喪失 0= 正常に送信
b <sub>4</sub>	TXERR	転送エラー検出	1= 転送エラー発生, メッセージ保留 0= 保留中の送信なし
b <sub>3</sub>	TXREQ	メッセージ送信要求	1= メッセージ送信要求 0= 保留中の送信要求なし
b <sub>2</sub>	-	未実装	
b <sub>1</sub>	TXP1	転送バッファ優先順位	11= 優先度 (最高) ~ 00= 優先度 (最低)
b <sub>0</sub>	TXP0		

表 3.7 TXBnSIDH レジスタ (アドレス : 0x31, 0x41, 0x51)

b <sub>7</sub>	SID10	標準 ID ビット 10
b <sub>6</sub>	SID9	標準 ID ビット 9
b <sub>5</sub>	SID8	標準 ID ビット 8
b <sub>4</sub>	SID7	標準 ID ビット 7
b <sub>3</sub>	SID6	標準 ID ビット 6
b <sub>2</sub>	SID5	標準 ID ビット 5
b <sub>1</sub>	SID4	標準 ID ビット 4
b <sub>0</sub>	SID3	標準 ID ビット 3

表 3.8 TXBnSIDL レジスタ (アドレス : 0x32, 0x42, 0x52)

b <sub>7</sub>	SID2	標準 ID ビット 2
b <sub>6</sub>	SID1	標準 ID ビット 1
b <sub>5</sub>	SID0	標準 ID ビット 0
b <sub>4</sub>		
b <sub>3</sub>	EXIDE	拡張 ID 許可ビット 1=拡張 ID (29 ビット) 0=標準 ID (11 ビット)
b <sub>2</sub>		
b <sub>1</sub>	EID4	拡張 ID ビット 17
b <sub>0</sub>	EID16	拡張 ID ビット 16

表 3.9 TXBnDLC レジスタ (アドレス : n= 0x35, 0x45, 0x55)

b7	-	未使用	
b6	RTR	リモート送信要求	1= リモートフレーム 0= データ・フレーム
b5	-	未実装	
b4	-	未実装	
b3	DLC3	データ長コード(DLC)	データ・フィールドのデータ長(0-8)
b2	DLC2		
b1	DLC1		
b0	DLC0		

表 3.10 RXB0CTRL レジスタ (アドレス : 0x60)

b7	-	未実装	
b6	RXM1	オペレーション・モード	11= マスク, フィルタ未使用 10= 拡張ID, マスク, フィルタを適用 01= 標準ID, マスク, フィルタを適用 00=マスク, フィルタを適用
b5	RXM0		
b4	-	未実装	
b3	RXRTR	リモート送信要求	1=送信要求を受けた 0=送信要求は受けていない
b2	BUKT	ロールオーバー・オプション	1= ロール・オーバーを許可 0= ロール・オーバー許可せず
b1	BUKT1	リード・オンリ・コピー	MCP2515 が内部で使用するフラグ
b0	FILHIT0	フィルタ・ヒット	1= フィルタ 1(RXF1)にヒット 0= フィルタ 0(RXF0)にヒット

表 3.11 RXB1CTRL レジスタ (アドレス : 0x70)

b7	-	未実装	
b6	RXM1	オペレーション・モード	11= マスク, フィルタ未使用 10= 拡張ID, マスク, フィルタを適用 01= 標準ID, マスク, フィルタを適用 00=マスク, フィルタを適用
b5	RXM0		
b4	-	未実装	
b3	RXRTR	フィルタ・ヒット	101= フィルタ 5 にヒット
b2	FILHIT2		100= フィルタ 4)にヒット
b1	FILHIT1		011= フィルタ 3 にヒット
b0	FILHIT0		010= フィルタ 2 にヒット
			001= フィルタ 1 にヒット 000= フィルタ 0 にヒット

表 3.12 RXBnDLC レジスタ (アドレス : 0x65, 0x75)

b7	-	未実装	
b6	RTR	リモート送信要求	1=リモート・フレーム 0=データ・フレーム
b5	RB1	リザーブ1	
b4	RB0	リザーブ2	
b3	DLC3	データ長コード(DLC)	データ・フィールドのデータ長(0-8)
b2	DLC2		
b1	DLC1		
b0	DLC0		

表 3.13 ステータス・リード・コマンドにおけるビット定義

bit	要因	意味
b7	TXB0CNTRL, TXREQ	送信バッファ 0 送信要求
b6	TXB1CNTRL, TXREQ	送信バッファ 1 送信要求
b5	TXB2CNTRL, TXREQ	送信バッファ 2 送信要求
b4	CANINTF, TX2IF	送信バッファ 0 「空」
b3	CANINTF, TX1IF	送信バッファ 1 「空」
b2	CANINTF, TX0IF	送信バッファ 2 「空」
b1	CANINTF, RX1IF	受信バッファ 1 「フル」
b0	CANINTF, RX0IF	受信バッファ 0 「フル」

表 3.14 受信ステータス・リード・コマンド(RX STATUS)

bit	要因	意味
b7	受信メッセージ	00= 受信なし
		01= RXB0 にマッチ
		10= RXB1 にマッチ
b6		11= 両バッファに受信あり
b5	未実装	未実装
b4	メッセージ・タイプ	00= スタンダード・データ・フレーム
		01= スタンダード・リモート・フレーム
b3		10= 拡張データ・フレーム
		11= 拡張リモート・フレーム
b2 ~ b0	フィルタ・マッチ	000= RXF0 にマッチ
		001= RXF1 にマッチ
		010= RXF2 にマッチ
		011= RXF3 にマッチ
		100= RXF4 にマッチ
		101= RXF5 にマッチ
		110= RXF0 にマッチ(RXB1 ヘロール・オーバー)
111= RXF1 にマッチ(RXB1 ヘロール・オーバー)		