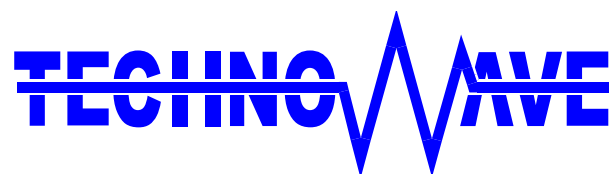


KBM3694 ユーザーズマニュアル



テクノウェーブ株式会社

1. はじめに.....	3
□ 注意事項	3
2. 製品概要.....	4
□ 特徴.....	4
3. 製品仕様.....	4
□ 概略.....	4
□ 端子説明	5
□ ジャンパースイッチ	6
4. 動作原理.....	7
□ キースキャン	7
□ キーコード.....	7
□ バッファ	8
□ タイパマティック・キー.....	8
□ 起動.....	8
5. 接続方法.....	9
□ キーマトリックス.....	9
□ PS/2 インタフェース.....	9
□ シリアルインタフェース.....	10
□ LCD インタフェース.....	11
□ LED.....	11
6. 制御方法.....	12
□ シリアルポートの設定	12
□ 制御コマンド	13
□ 動作オプションの設定	14
□ キーコードの入力.....	15
□ I/O ポートの操作.....	15
PIO.....	15
PO	16
PI.....	16
□ AD 変換	17
□ LCD の制御	18
初期化处理	18
インストラクションレジスタへの書き込み	18

データレジスタへの書き込み	19
□ 通信エラー	20
□ コマンドリファレンス	21
<i>SET_DEFAULT</i>	21
<i>READ_VERSION</i>	21
<i>READ_STATUS</i>	22
<i>ECHO</i>	22
<i>SET_MODE0</i>	23
<i>READ_MODE0</i>	24
<i>ENABLE</i>	25
<i>DISABLE</i>	25
<i>SET_TR_DLY</i>	26
<i>INDICATOR</i>	27
<i>PIO_WRITE</i>	28
<i>PIO_READ</i>	28
<i>PIO_DIR</i>	29
<i>PI_READ</i>	29
<i>PO_WRITE</i>	30
<i>PO_READ</i>	30
<i>LCD_DWRITE</i>	31
<i>LCD_DREAD</i>	31
<i>LCD_RWRITE</i>	32
<i>LCD_RREAD</i>	32
<i>LCD_INIT</i>	33
<i>LCD_CGLOAD</i>	33
<i>AD_READ</i>	34
7. フラッシュメモリ	35
□ 『KBMWriter』使用方法	35
APPENDIX.....	37
□ キーボード配列と走査コード	37
キーボード配列	37
走査コード・セット 2	38
□ フラッシュメモリ書換え用データフォーマット	41
保証期間.....	44
サポート情報.....	44

1. はじめに

このたびはキーエンコーダ・モジュール『KBM3694』をご購入頂き、まことにありがとうございます。以下の注意事項をよくお読みになり、安全にご使用いただけますようお願い申し上げます。

□ 注意事項

- (1) ジャンパースイッチやコネクタなどには尖った部分がありますので、取り扱いの際には十分ご注意ください。
- (2) 本製品は製品の性質上、電源も含めて信号線が露出している部分があります。製品、接続したパソコンやその他の機器などが故障する恐れがありますので、信号線同士がショートしないよう十分ご注意ください。
- (3) 本製品にはお客様のご用途に合わせて設定を変更していただくためのジャンパースイッチがあります。これらの接続を誤りますと、製品、接続したパソコンやその他の機器などが故障する場合がありますので、設定変更の際には本マニュアルの内容に従い、十分ご注意ください。
- (4) お客様の不注意、誤操作により発生した製品、パソコン、その他の故障、及び事故につきましては弊社は一切の責任を負いませんのでご了承ください。
- (5) 本書、および本製品のホームページに掲載されている応用回路、プログラム、使用方法などは、製品の代表的動作・応用例を説明するための参考資料です。これらに起因する第三者の権利(工業所有権を含む)侵害、損害に対し、弊社はいかなる責任も負いません。
- (6) 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
- (7) 本製品は一般民製品です。特別に高い品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある機器(医療機器、輸送機器、交通信号機、航空宇宙機器、海底中継器、燃焼制御、原子力発電制御機器、各種安全装置など)に使用すること(以下“特定用途”)を前提としていません。よって、弊社は本製品をこれらの製品に用いた場合の如何なる責任についても負いかねます。本製品を当該特定用途に使用される場合は、お客様の責任においてなされることになります。
- (8) 本書の内容の一部または全部を無断転載することをお断りします。
- (9) 本書の内容については、将来予告なしに変更することがあります。
- (10) 本書の内容については、万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がございましたらご連絡ください。

2. 製品概要

□ 特徴

『KBM3694』(以下、本モジュール)は多機能キーエンコーダ・モジュールです。パソコンのキーボードコネクタと接続すれば、PS/2 キーボードと同様に使用できます。最大 128 点のスイッチを接続可能で、それぞれのキーコードは内部フラッシュメモリに格納されますので、Windows®用の付属ツールを使用して自由に変更可能です。

また、シリアルインタフェースからは、I/O ポート、AD コンバータをコントロールでき、シリアル I/O モジュールとしても使用可能です。さらに、キャラクタタイプの LCD とのインタフェース機能も備えていますので、1 台のモジュールで様々なアプリケーションに対応可能となっております。

シリアルインタフェースは TTL レベルでの通信も可能ですので、パソコンや産業用コンピュータの他、マイコンなどとの接続も可能です²。

3. 製品仕様

□ 概略

表 1 仕様概略

項目		仕様
基板寸法		54 × 38 [mm]
電源電圧 ³		5[V] ± 5%
消費電流(ボード単体、無負荷時)		50[mA]
動作温度範囲		-20 ~ 70°C
キースキャン間隔		10 [ms]
キーボードインタフェース		PS/2、シリアル通信
制御インタフェース		シリアル通信(バイナリデータによる。表 2 参照)
内蔵フラッシュメモリ書換え可能回数		1000 回
I/O 端子信号		TTL、非絶縁
I/O 端子数	入力専用	最大 7 ピン
	出力専用	最大 4 ピン
	入出力	最大 7 ピン

表 2 シリアル通信仕様

項目	仕様
同期方式	調歩同期
信号レベル	RS-232C、TTL
通信速度	4800 ~ 57600 [bps]
フロー制御	CTS 信号による送信制御
データ長	8 ビット
ストップビット	1
パリティ	偶数パリティ

Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

¹ 端子を兼用しているため、全ての機能を同時に使用することはできません。

² キーコードや動作オプションをフラッシュメモリに格納するには Windows を搭載したパソコンが必要になります。

³ PS/2 接続時は PS/2 ポートより供給可能です。

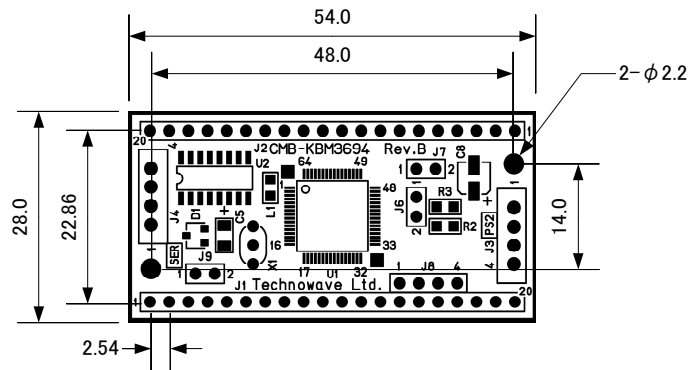


図 1 基板寸法図

□ 端子説明

表 3 J1,J2 端子

コネクタ ピン番	信号名	説明	方向	コネクタ ピン番	信号名	説明	方向
J1-1	COL0#	COL0#入力	I	J2-1	GND	GND	-
J1-2	COL1#	COL1#入力	I	J2-2	CTS#	シリアル CTS 入力 (TTL)	I
J1-3	COL2#	COL2#入力	I	J2-3	RxD	シリアル入力 (TTL レベル)	I
J1-4	COL3#	COL3#入力	I	J2-4	TxD	シリアル出力 (TTL レベル)	O
J1-5	COL4#	COL4#入力	I	J2-5	LCD_RS/PI00	LCD 制御/入出力ポート	I/O
J1-6	COL5#	COL5#入力	I	J2-6	LCD_R/PI01	LCD 制御/入出力ポート	I/O
J1-7	COL6#	COL6#入力	I	J2-7	LCD_E/PI02	LCD 制御/入出力ポート	I/O
J1-8	COL7#	COL7#入力	I	J2-8	LCD_D4/PI04	LCD 制御/入出力ポート	I/O
J1-9	COL8#	COL8#入力	I	J2-9	LCD_D5/PI05	LCD 制御/入出力ポート	I/O
J1-10	ROW0	ROW0 出力	O	J2-10	LCD_D6/PI06	LCD 制御/入出力ポート	I/O
J1-11	ROW1	ROW1 出力	O	J2-11	LCD_D7/PI07	LCD 制御/入出力ポート	I/O
J1-12	ROW2	ROW2 出力	O	J2-12	P16/AD6/COL15#	入力ポート/AD6/COL15#入力	I
J1-13	ROW3	ROW3 出力	O	J2-13	P15/AD5/COL14#	入力ポート/AD5/COL14#入力	I
J1-14	PO4/ROW4	出力ポート/ROW4 出力	O	J2-14	P14/AD4/COL13#	入力ポート/AD4/COL13#入力	I
J1-15	PO5/ROW5	出力ポート/ROW5 出力	O	J2-15	P13/AD3/COL12#	入力ポート/AD3/COL12#入力	I
J1-16	PO6/ROW6	出力ポート/ROW6 出力	O	J2-16	P12/AD2/COL11#	入力ポート/AD2/COL11#入力	I
J1-17	PO7/ROW7	出力ポート/ROW7 出力	O	J2-17	P11/AD1/COL10#	入力ポート/AD1/COL10#入力	I
J1-18	KB_CLK	PS/2 クロック信号	I/O	J2-18	P10/AD0/COL9#	入力ポート/AD0/COL9#入力	I
J1-19	KB_DATA	PS/2 データ信号	I/O	J2-19	RES#	リセット入力	I
J1-20	GND	GND	-	J2-20	VCC	PS/2 と接続の場合オープン	-

表 4 シリアル通信用端子(J4)

コネクタ-ピン番	信号名	説明	方向
J4-1	CTS	CTS 入力	I
J4-2	TxD	シリアル出力	O
J4-3	RxD	シリアル入力	I
J7-4	GND	GND	-

適合コネクタ：XHP-4(日本圧着端子製造株式会社)

表 5 PS/2 インタフェース用端子(J3)

コネクタ-ピン番	信号名	説明	方向
J3-1	VCC	VCC	-
J3-2	KB_DATA	PS/2 データ信号	I/O
J3-3	KB_CLK	PS/2 クロック信号	I/O
J3-4	GND	GND	-

適合コネクタ：EHR-4(日本圧着端子製造株式会社)

表 6 LED 用電源端子(J8)

コネクタ-ピン番	信号名	説明	方向
J8-1	LED_COM0	LED 用の電源端子です。150Ω の抵抗を通して VCC と接続されています。	-
J8-2	LED_COM1		-
J8-3	LED_COM2		-
J8-4	LED_COM3		-

表 7 DC 特性

VCC=5.0V、GND=0.0V、Ta=-20～70℃

項目		記号	Min	Max	単位	測定条件
入力 High レベル電圧	RES#	V_{IH}	4.0		V	
	RES#以外		3.5			
入力 Low レベル電圧	RES#	V_{IL}		1.0	V	
	RES#以外			1.5		
出力 High レベル電圧		V_{OH}	4.0		V	$I_{OH} = -1.5mA$
出力 Low レベル電圧		V_{OL}		0.6	V	$I_{OL} = 1.6mA$
出力 Low レベル 許容電流	PO、KB_DATA、KB_CLK 以外	I_{OL}		2	mA	
	KB_DATA、KB_CLK			6		
	PO			20		
出力 High レベル許容電流		$ I_{OH} $		2	mA	
総和出力 Low レ ベル許容電流	PO 以外	$ \sum I_{OL} $		40	mA	
	PO			60		
総和出力 High レベル電流		$ \sum I_{OH} $		30	mA	

ADC の特性は「AD 変換」をご参照ください。

□ ジャンパースイッチ

ジャンパースイッチの機能を示します。全てのジャンパー設定は必ず電源オフの状態で行ってください。

表 8 ジャンパースイッチ

番号	説明	出荷時の設定
J7	CTS#信号を強制的に"Low"レベルにします。フロー制御が必要ないときに"ON"にします	ON
J9	"ON"にして起動すると、自動的にフラッシュの書換モードに移行します。	OFF
J6	使用しません。"ON"にしないでください。	OFF

"ON":ショートした状態、"OFF":オープンの状態

4. 動作原理

□ キースキャン

キーエンコーダにキーを接続した様子を以下の図に示します。図中の白丸の部分は図 3 のようにスイッチが挿入されているものとします。

ROW0～ROW7 の各端子は、通常ハイインピーダンスになっています。キースキャンを行う際、キーエンコーダはまず、図中の ROW0 端子に“Low”を出力します。そして各 COLx#端子のレベルを読み取ります。もし、ROW0 に接続されているスイッチのどれかが押されている場合、そのスイッチが接続されている COLx#端子は“Low”として読み出されるので、エンコーダはそのスイッチが押されていることを知ることができます。押されていない COLx#端子はプルアップ抵抗により“High”に固定されます。ROW0 行の読み取りが終わると、ROW0 端子をハイインピーダンスに戻し、ROW1 端子に“Low”を出力し、同様に COLx#端子を読み取ります。この作業を ROW7 まで繰り返すことで、最大 128 個のスイッチの状態をスキャンします。

キーエンコーダは 10ms 毎に全てのキーをスキャンし、必要な場合、キーコードをホストに通知します。

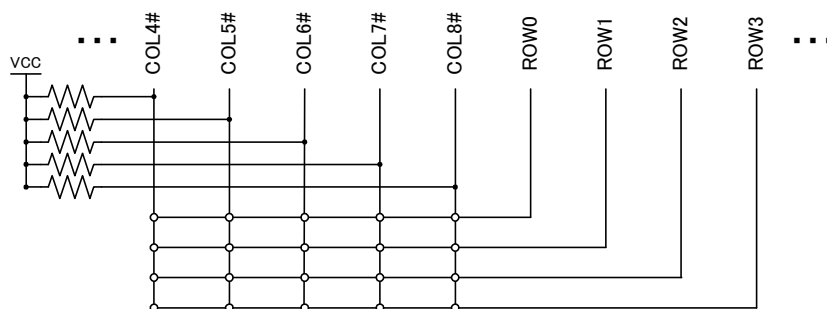


図 2 キーの接続

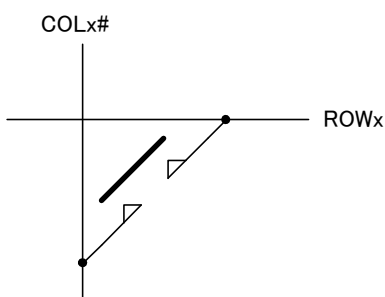


図 3 スイッチ

□ キーコード

各キーとキーコードの対応表は、キーエンコーダ・モジュールのメモリ内に格納されます。パソコン用キーボードのほとんどのキーは、メークコードとブレイクコードを持っています。メークコードはあるキーが押されたときに、ホストに通知されるコードで、ブレイクコードはそのキーが開放されたときに通知されるコードです。例えば、「A」というキーが押された場合、“1C”(HEX)というメークコードがホス

トに送られ、離されると“F0,1C”(HEX)という 2 バイトのブレイクコードが送られます。

キーボードの規格では 3 つの走査コード・セット(キーとキーコードの組み合わせを定義したもの)が定義されていますが、一般には「走査コード・セット 2」が使用されています。本モジュールは走査コード・セットを動的に切り替える機能を持っていません。PS/2 の「代替走査コード選択」コマンドでの切り替えは無視されます。走査コード・セットの問い合わせに対しては常に「走査コード・セット 2」であるかのように応答します。

また、本モジュールは「Num Lock」やシフト状態によってキーコードを入れ替える機能を持っていません。そのため、完全に PS/2 キーボードと同じ動作を再現することはできません。

□ バッファ

キーコードは、一旦キーエンコーダ・モジュール内のバッファに蓄えられ、順次ホストに送信されます。本モジュールでは 128 バイトのバッファを持っています。バッファから溢れたキーコードは廃棄され、代わりにオーバーラン・コード“00”(HEX)が送られます。

□ タイパマティック・キー

押し続けると連続してキーコード(メークコード)を送出するキーのことです。タイパマティック・キーとして登録されているキーは一定時間以上押し続けられると、メークコードをホストに対して連続で送信します。最初にメークコードが送信されてから、タイパマティック動作を開始して次のメークコードを送信するまでの時間をタイパマティック・ディレイ、タイパマティック動作中のメークコードの送信レートをタイパマティック・レートと呼びます。

複数のキーが押された場合、タイパマティック動作によって送られるメークコードは、最後に押されたキーのものです。最後に押されたキーが離されると、他のキーが押されたままでもタイパマティック動作は終了します。

□ 起動

一般のキーボードでは電源投入時に「電源投入リセット(POR)」と、「基本保証テスト(BAT)」が実行されます。本モジュールは以下のステップで起動します。

1. 電源投入リセット(150ms～500ms)。
2. J9 端子のチェック。J9 が“ON”の場合、フラッシュ書換モードに移行します。PS/2 インタフェースには応答しません。J9 が“OFF”の場合、3.に進みます。
3. 初期化作業。
4. ダミーの基本保証テスト。実際にはテストは行いません。定められた時間だけ停止します(約 500ms)
5. BAT 完了コードの送信。PS/2 インタフェースに BAT 完了コード“AA”(HEX)を送信します。
6. キースキャンを開始し、PS/2、及びシリアルコマンドに備えます。

5. 接続方法

□ キーマトリックス

図 4 にキーの番号と ROWx 端子、COLx#端子の関係を示します。COL0#～COL8#端子はモジュール内でプルアップされていますので、外部にプルアップ抵抗を接続する必要はありません。COL9#～COL15#をご使用になる場合は、別途プルアップ抵抗を取り付ける必要があります。

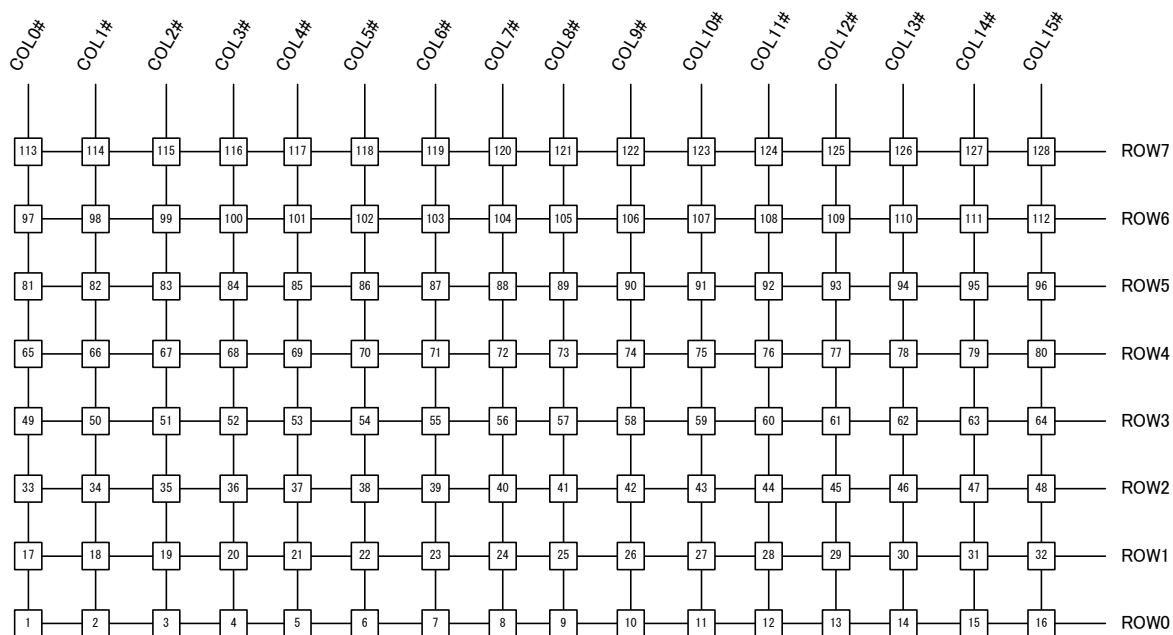


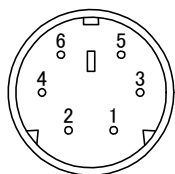
図 4 キー配置

- ・ 出荷時の設定では ROW4～ROW7 は出力ポート(PO4～PO7)として機能します。ROW 信号として使用するためにはオプションを変更してください。
- ・ 出荷時の設定では COL9#～COL15# は入力ポート(PI0～PI7)として機能します。COL#信号として使用するためにはオプションを変更してください。
- ・ COL9#～COL15# はプルアップされていません。COL#信号として使用する場合は外部にプルアップ抵抗(10kΩ程度)を接続してください。

□ PS/2 インタフェース

本モジュールを通常のキーボードのように、パソコンやボードコンピュータ等に接続する場合には PS/2 インタフェースを使用します。

図 5 は Windows パソコンのキーボード用 PS/2 ポートです。パソコンに接続する場合には、図を参照してそれぞれの信号を接続してください。J1 と J3 の同一信号名の端子はモジュール内で接続されています。どちらを使用しても構いません。



パソコン		KBM3694	
1	DATA	J3-2,J1-19	KB_DATA
2	Reserved	—	—
3	GND	J3-4,J1-20	GND
4	5V	J3-1,J2-20	VCC
5	CLOCK	J3-3,J1-18	KB_CLK
6	Reserved	—	—

図 5 PS/2 の接続

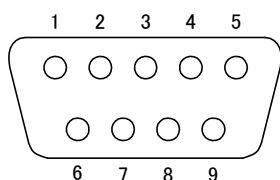
- ・ 本モジュールの J2-20 端子に PS/2 以外から 5V を供給される場合には、PS/2 の 4 番ピンとは接続しないでください。異なる電源が並列に接続されると、電源回路にダメージを与える可能性があります。

□ シリアルインタフェース

本モジュールが出力するキーコードをシリアルポートで受信する場合、I/O ポートや AD コンバータの機能を利用する場合、内蔵フラッシュメモリを書き換える場合にはシリアルポートの接続が必要になります。

図 6 はパソコンのシリアルポートです。パソコンと接続する場合には、図を参照してそれぞれの信号を接続してください。

J4-1 の CTS 信号はフロー制御用入力です。出荷時には無効になっていますので、ご使用になる場合には J7 のジャンプスイッチを“OFF”にしてください。アクティブレベルが入力された場合のみ、データが送信されるようになります。



パソコン		KBM3694	
1	DCD	—	—
2	RxD	J4-2	TxD
3	TxD	J4-3	RxD
4	DTR	—	—
5	GND	J4-4	GND
6	DSR	—	—
7	RTS	J4-1	CTS
8	CTS	—	—
9	RI	—	—

図 6 シリアルポートの接続

TTL レベルのシリアル通信によって本モジュールを制御する場合には、J2-2～J2-4 端子をご使用ください。J2-2(CTS#)と J2-3(RxD)は 2mA 以上の駆動電流が必要ですので、信号元のドライブ能力にご注意ください。また、J2 と J4 のシリアル端子を同時に使用することはできません。

- ・ J2 の TTL シリアルを使用しない場合は、J2-2～J2-4 端子には何も接続せず、オープンとしてください。

□ LCD インタフェース

本モジュールは一般的なキャラクタタイプの LCD モジュールと接続するためのインタフェースを備えています。図 7 に LCD モジュールとの接続例を示します。表 4 は動作確認済みの LCD モジュールです。LCD との接続は 4 ビットモードで行われます。

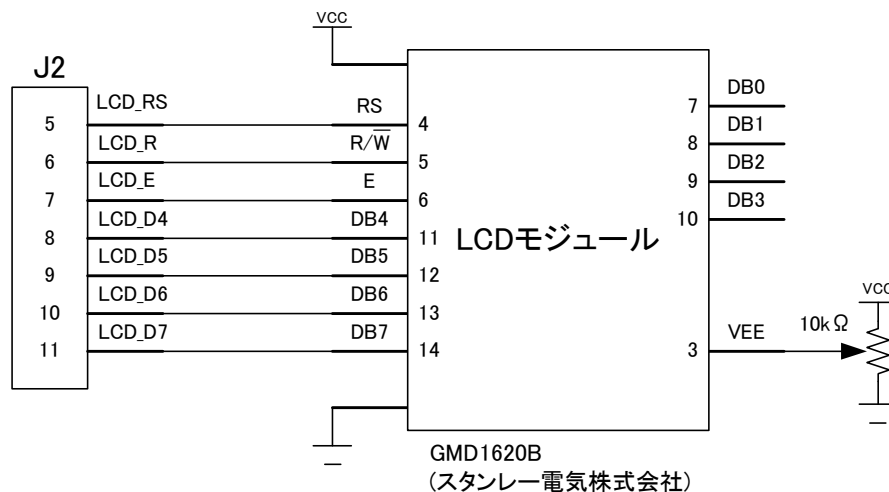


図 7 LCD モジュールとの接続例

表 9 動作確認済み LCD モジュール

メーカー名	型式
スタンレー電気株式会社	GMD1620B
SUNLIKE DISPLAY TECH. CORP.	SC2004C
SUNLIKE DISPLAY TECH. CORP.	SC1602B

□ LED

J8 の各端子はモジュール内で図 8 の左図のように抵抗を通して VCC と接続されています。図 8 の右図のように LED を接続すると、PO 端子出力が“Low”のとき点灯、“High”のとき消灯となり、出力ポートにより点灯/消灯のコントロールが可能です。

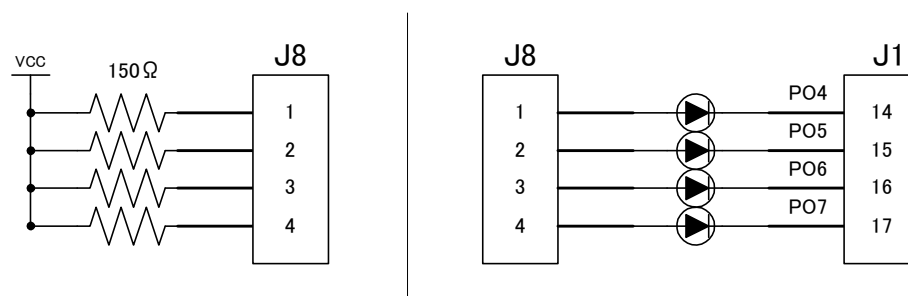


図 8 LED の接続

- LED は順方向電圧(V_f)が 2.0～3.5V のものをご使用ください。

6. 制御方法

本章ではシリアルポートを通して本モジュールを制御する方法を解説します。I/O ポートや AD コンバータ、LCD インタフェースを制御する場合には、本章の内容に従って操作を行ってください。単に PS/2 キーエンコーダとして使用する場合には、ホストコンピュータと接続(「接続方法」を参照)するだけで PS/2 キーボードとして認識されますので特別な制御はありません。

また、本章で解説する制御方法を API 関数化し、弊社ホームページにて公開しておりますので、Visual C++ をご使用のお客様はそちらをダウンロードしていただければ、プログラミングの手間を省くことができます。API 関数はソースコードで公開しておりますので、Visual C++ 以外の開発環境をお使いの方や、Windows 環境以外のアプリケーションを作られる場合にもご参照ください。

□ シリアルポートの設定

表 10 にシリアルポートの設定を示します。通信速度の出荷時の設定は 38400[bps]です。ツールを使用して変更可能ですが、特別必要が無い限り出荷時の設定を変更されないことをお勧めします。

表 10 シリアルポートの設定

項目	設定
同期方式	調歩同期
データ長	8ビット
ストップビット	1
パリティ	偶数パリティ
通信速度	可変(4800,9600,14400,19200, 38400 ,57600 [bps])

CTS 信号を使用する場合には J7 のジャンパースイッチを OFF にしてください。本モジュールは CTS 信号がアクティブレベルのときだけデータを送信します。特にホストコンピュータにマイコンなどをご利用の場合には、処理速度の関係でフロー制御が必要になることがあります。

ホストコンピュータがパソコンやボードコンピュータの場合は、CTS 信号を操作する必要は、あまりありませんが、ご使用になる場合は RTS 信号と接続してください。

送信データがある場合に、CTS 信号で通信が禁止されていると、再び許可されるのを待ち続けます。この間、PS/2 ポートへのキーコード出力、キースキャンも停止してしまいますのでご注意ください。

□ 制御コマンド

本モジュールの制御はホストから 1 バイトのコマンドコードを送ることで開始されます。デバイスはホストからのコマンドやデータを 1 バイト受信する毎に ACK コマンドで応答します。通信は全てバイナリです。表 11 にホストから送信するコマンドの一覧を、表 12 に本モジュールからの応答コマンドを示します。定義名は API 関数のソースファイル中で `#define` によって定義されている名称です。各コマンドの詳細は「コマンドリファレンス」を参照してください。

表 11 コマンド一覧

コマンド名	コード(HEX)	意味	定義名
SET_DEFAULT	f6	初期化	KBM_CMD_SET_DEFAULT
ENABLE	f4	キーボードスキャンをイネーブル	KBM_CMD_ENABLE
DISABLE	c9	キーボードスキャンをディセーブル	KBM_CMD_DISABLE
SET_TR_DLY	f3	タイパマティック・ディレイ/レートを設定	KBM_CMD_SET_TR_DLY
ECHO	ee	エコー	KBM_CMD_ECHO
INDICATOR	ed	インジケータ(LED)の設定	KBM_CMD_INDICATOR
PIO_WRITE	df	PIO ヘライト	KBM_CMD_PIO_WRITE
PIO_READ	de	PIO からリード	KBM_CMD_PIO_READ
PO_WRITE	dd	PO ヘライト	KBM_CMD_PO_WRITE
PO_READ	dc	PO からリード	KBM_CMD_PO_READ
PI_READ	db	PI からリード	KBM_CMD_PI_READ
PIO_DIR	da	PIO の入出力方向変更	KBM_CMD_PIO_DIR
LCD_DWRITE	d9	LCD のデータレジスタへライト	KBM_CMD_LCD_DWRITE
LCD_DREAD	d8	LCD のデータレジスタからリード	KBM_CMD_LCD_DREAD
LCD_RWRITE	d7	LCD のインストラクションレジスタへライト	KBM_CMD_LCD_RWRITE
LCD_RREAD	d6	LCD のインストラクションレジスタからリード	KBM_CMD_LCD_RREAD
LCD_INIT	d5	LCD の初期化	KBM_CMD_LCD_INIT
LCD_CGLOAD	d4	フラッシュから CGRAM データを LCD に転送	KBM_CMD_LCD_CGLOAD
AD_READ	d3	AD のリード	KBM_CMD_AD_READ
SET_MODE0	cf	動作モード 0 の設定	KBM_CMD_SET_MODE0
READ_MODE0	ce	動作モード 0 のリード	KBM_CMD_READ_MODE0
READ_STATUS	cb	ステータスの読み出し	KBM_CMD_READ_STATUS
READ_VERSION	ca	バージョンの読み出し	KBM_CMD_READ_VERSION

表 12 応答コマンド一覧

コマンド名	コード(HEX)	意味	定義名
ACK	fa	正常応答	KBM_CMD_ACK
RESEND	fe	再送要求	KBM_CMD_RESEND
COM_ERR	f1	通信エラーの発生を通知	KBM_CMD_COM_ERR
OVERRUN	00	(キーボードの)オーバーランエラー	KBM_CMD_OVERRUN

□ 動作オプションの設定

[SET_MODE0]コマンドで動作オプションを変更することができます。設定するデータは1バイトで、各ビットの意味は以下の表のようになっています。

表 13 動作オプション

ビット	意味	出荷時の設定
0	1 とすると起動時に BAT 終了コードを PS/2 ポートに送りません。	0
1	1 とすると識別できない PS/2 コマンドにも ACK で応答します。	0
2	1 とすると PIO 端子(J2-5～11)を LCD インタフェースに設定します。	1
3	1 とすると PO4-7(J1-14～17)端子を出力ポートとして使用します。	1
4	1 とすると PIO-6(J2-18～12)端子を COL9-15#入力としてスキャンします。	0
5	1 とするとインジケータの設定値を PIO4-7 に出力します。	0
6	1 とするとキーコードをシリアルに出力します。	0
7	予約(0 としてください)	0

ビット 0、ビット 1 は通常 0 としてください。

ビット 2 を 1 とすると J2-5～11 の各端子は LCD インタフェース用として設定されます。I/O ポートとしての操作やインジケータとしては使用できなくなります。

ビット 3 を 1 とすると J1-14～17 の端子が出力ポート(PO4～PO7)となり、0 とすると ROW4～ROW7 の出力となります。出荷時の設定では J1-14～17 は出力ポートとなっていますので、キーの数が足りない場合には ROW 出力に変更してください。

ビット 4 を 1 とすると J2-18～12 端子が COL9#～COL15#入力となります。0 の場合は、入力ポート(PIO～PI6)、または AD コンバータへのアナログ入力端子(AD0～AD6)として使用できます。出荷時の設定では入力ポートまたは、AD コンバータの入力端子として使用する設定となっていますので、キーの数が足りない場合には、COLx#信号入力に設定してください。

ビット 5 を 1 とするとキーボードのインジケータ(LED)への設定値を PIO4～PIO7 端子に出力します。ただし、PIO4～PIO7 は LED を駆動できませんので、実際に表示する場合には外付けのバッファが必要になります。また、PIO 端子は LCD インタフェースの設定が優先されますので、LCD インタフェースに設定されている場合にはビット 5 の設定は無効です。

表 14 PIO ビットとインジケータの対応

ビット	意味
PIO4	Scroll Lock
PIO5	Num Lock
PIO6	Caps Lock
PIO7	Kana (AX キーボード)

ビット 6 を 1 とするとキーコードをシリアルインタフェースに出力します。この場合、PS/2 インタフェースへは出力を行いません。また、PS/2 経由のキーボードコマンドにも応答しません。

□ キーコードの入力

キーコードは動作オプションを変更することにより、シリアルインタフェースに出力させることができます。設定方法は「動作オプションの設定」を参照してください。

キーコードの送出方法は PS/2 の場合とほぼ同様です。本モジュールはユーザーがキーを押したとき、または離れたときにメークコード、ブレイクコードをシリアルポートへ送信します。また、キーバッファの容量を越えてキーが入力されると、[OVERRUN]コマンド(“00”)を送信します。

キーコード送信のための通信は本モジュールから開始されます。そのため、他の制御コマンドを同時に使用すると、応答コマンドやデータとキーコードが混ざってしまう恐れがあります。シリアルインタフェースからキーコードを受け取る場合には、他の制御コマンドを使用する前に[DISABLE]コマンドでキースキャンを一旦停止して、シリアルを受信バッファをクリアされることをお勧めします。制御コマンドが完了した後、再び[ENABLE]コマンドでキースキャンを再開します。

□ I/O ポートの操作

本モジュールには 3 種類の I/O ポートがあります。

表 15 利用可能なポートの種類

ポート名	方向	ピン数	説明
PIO	入出力	7	LCD インタフェース、インジケータ出力と兼用。
PO	出力	4	ROW 出力と兼用。LED 駆動可能
PI	入力	7	AD 入力、COL#入力和兼用。

PIO

PIO0～PIO7 は入出力どちらにも設定可能な I/O 端子です。方向の設定には[PIO_DIR]コマンドを使用します。コマンドコードに続いて 1 バイトのデータを送ります。データバイトの各ビットは図 9 のようにそれぞれの端子と対応しています。1 としたビットと対応した端子は出力となり、0 としたビットと対応した端子は入力となります。

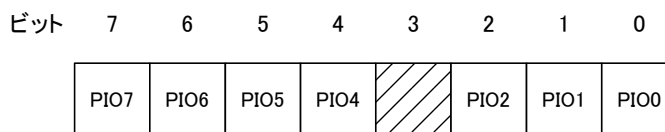


図 9 PIO データと端子の対応

出力に設定されている端子の値を変更するには[PIO_WRITE]コマンドを使用します。続く 1 バイトのデータで各端子の値を指定します。

ポートからの入力には[PIO_READ]コマンドを使用します。モジュールからは応答コマンドに続いて、ポートから読み取った結果が送られます。出力に設定されている端子のビットは現在の出力値が返ります。

PIO ポートは出荷時には LCD インタフェースとして設定されています。I/O ポートとして使用する場

合には動作オプションを変更してください。

PO

PO4～PO7 は出力専用のポートで、シンク電流は 1 端子あたり 20mA まで許容されていますので、直接LEDを接続することが可能です。出力値の変更は[PO_WRITE]コマンドを使用します。コマンドコードに続く 1 バイトのデータで各端子の値を指定します(図 10 参照)。[PO_READ]コマンドで現在の出力値を読み出すことができます。

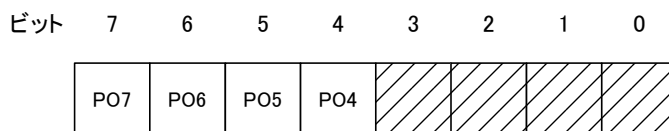


図 10 PO データと端子の対応

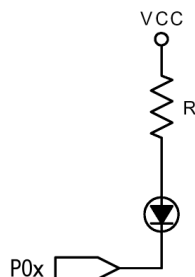


図 11 PO 端子と LED の接続例

- ・ PO 端子のソース電流は最大 2mA です。LED の取り付け方向にご注意ください。
 - ・ PO4～PO7 端子のシンク電流の合計が 60mA を超えないようにしてください。

PI

PI0～PI7 は入力専用のポートです。ポートからの入力には[PI_READ]コマンドを使用します。モジュールからは応答コマンドに続いて、ポートから読み取った 1 バイトのデータが送られます。

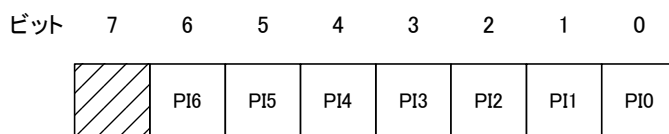


図 12 PI データと端子の対応

□ AD 変換

10 ビットの AD コンバータを最大で 7 チャンネル利用できます。[AD_READ]コマンドを使用して、AD0～AD6 端子に入力されたアナログ電圧を 10 ビットのデジタルデータに変換して読み出すことができます。コマンドコードとチャンネル番号を送信すると、AD 変換の結果が下位バイト、上位バイトの順で 2 バイト返されます。変換結果は図 13 のように 16 ビットデータの上位 10 ビットに納められ、下位 6 ビットは常に 0 となります。アナログ入力電圧と出力コードの関係を図 14 に示します。

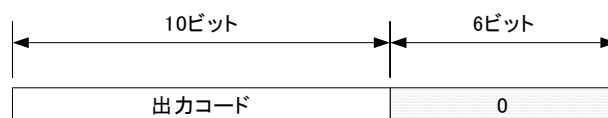


図 13 変換結果の格納方法

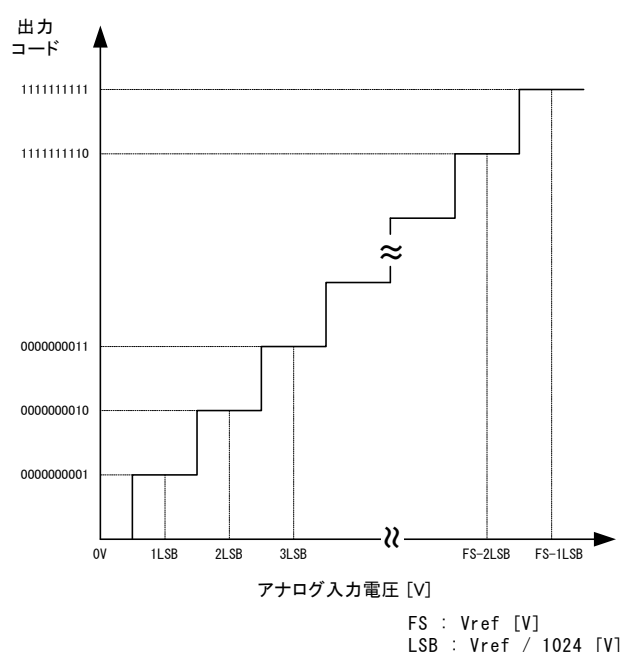


図 14 アナログ入力電圧と出力コードの関係

AD コンバータの特性を表 16 に示します。

表 16 AD 変換特性

項目	min	max	単位
リファレンス電圧(Vref)	4.95	5.05	V
アナログ入力範囲	0	Vref	V
分解能	10	10	bit
変換時間		6.7	us
アナログ入力容量		30	pF
許容信号源インピーダンス		5	kΩ
非直線性誤差		±3.5	LSB
オフセット誤差		±3.5	LSB
フルスケール誤差		±3.5	LSB
量子化誤差		±0.5	LSB
絶対精度		±4.0	LSB

変換時間にシリアル通信時間は含まれません。

□ LCD の制御

本モジュールはキャラクタタイプの LCD モジュールとのインタフェース機能を持っています。初期化処理やレジスタのリード/ライトを簡単なコマンドだけで行うことが可能です。

初期化処理

LCD モジュールは初期化されることで初めて使用可能になります。本モジュールは起動時や PIO が LCD インタフェースに設定されたときに、自動で LCD モジュールの初期化処理を行います。その他に初期化が必要な場合には[LCD_INIT]コマンドを使用してください。

インストラクションレジスタへの書き込み

LCD モジュールはインストラクションレジスタにコードを設定することで、カーソル位置の設定、スクロール、表示/非表示の切り替えなどを行います。インストラクションレジスタにコードを書き込む場合は[LCD_RWRITE]コマンドを使用します。コマンドに続いてインストラクションコードを送信します。主なインストラクションコードを表 17 に示します。

表 17 インストラクションコード

インストラクション	コード		説明
	16 進	2 進	
表示クリア	01	00000001	表示をクリアし、アドレスカウンタを DDRAM の 0 番地に移動
カーソルホーム	02	00000010	アドレスカウンタを DDRAM の 0 番地に設定
表示許可	0C	00001100	画面表示を許可
表示禁止	08	00001000	画面表示を禁止
カーソル表示	0A	00001010	カーソルを表示
文字のブリンク	09	00001001	カーソルのある位置の文字を点滅
カーソルを左移動	10	00010000	カーソルを左に移動
カーソルを右移動	14	00010100	カーソルを右に移動
画面の左スクロール	18	00011000	画面を左にスクロール
画面の右スクロール	1C	00011100	画面を右にスクロール
DDRAM アドレスセット	2 進参照	1xxxxxxx	アドレスカウンタに DDRAM アドレスを設定。アドレスは下位 7 ビットで指定します。
CGRAM アドレスセット	2 進参照	01xxxxxx	アドレスカウンタに CGRAM アドレスを設定。アドレスは下位 6 ビットで指定します。

初期化処理の後には非表示の設定となっていますので、「表示許可」のインストラクションは必ず必要になります。また「表示許可」、「カーソル表示」、「文字のブリンク」は“OR”で結合して一度に指定します。

データレジスタへの書き込み

データレジスタは画面と対応した DDRAM 領域と、独自のキャラクタを設定するための CGRAM 領域の 2 つがあります。LCD 画面に文字を表示するためには「DDRAM アドレスセット」のインストラクションで表示アドレスを設定した後に、[LCD_DWRITE]コマンドで文字コードを送ります。文字コードは一般に「JS8 単位符号」⁴が使用できます。[LCD_DWRITE]コマンドでデータを送ると DDRAM のアドレスは自動的にインクリメントされますので、文字列を連続して送ることができます。表 18 は 16 文字 × 2 行タイプ LCD モジュールの表示画面と DDRAM アドレスの対応です。

表 18 LCD の表示画面と DDRAM アドレスの対応

行 \ 桁	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2	40	41	42	43	44	45	46	47	48	49	5A	5B	5C	5D	5E	5F

CGRAM はユーザーが作成したキャラクタを、8 文字まで登録可能な領域です。この領域のキャラクタは文字コード“00”～“07”(HEX)が割り当てられており、DDRAM に文字コードを書き込むことで、予め登録したオリジナルの文字を LCD 画面に表示することができます。

CGRAM に書き込みを行うには「CGRAM アドレスセット」のインストラクションで CGRAM アドレスを設定した後、キャラクタデータを [LCD_DWRITE]コマンドで順次送ります。

また、モジュール内のフラッシュメモリには CGRAM に書き込むためのデータを予め用意しておくことができます。フラッシュ内のデータを CGRAM に転送するには [LCD_CGLOAD]コマンドを使用します。CGRAM へのデータ設定や、その他 LCD モジュールについての詳細はそれぞれのモジュールのデータシートをご参照ください。

⁴ ASCII コードと半角カナ

□ 通信エラー

シリアル通信でエラーが発生した場合の本モジュールの動作を説明します。表 19 に通信エラーの一覧を示します。ビットは[READ_STATUS]コマンドで読み出されるステータス情報のビット位置を示しています。

表 19 通信エラー一覧

エラー	ビット	説明
タイムアウトエラー	0	コマンドに続くはずのデータが 50ms 以上たっても受信されない場合に発生します。そのコマンドは無視され、通常のコマンドループに戻ります。
オーバーランエラー	1	ホストからの送信データを読み出す前に、次のデータが到着しデータを消失してしまった場合に発生します。モジュールは直ちにエラーループに入ります。
フレーミングエラー	2	ストップビットが検出できなかった場合に発生します。モジュールは[RESEND]コマンドを送って再送要求を行います。再送時にエラーが発生した場合にはエラーループに入ります。
パリティエラー	3	パリティビットによりエラーが検出された場合です。モジュールは[RESEND]コマンドを送って再送要求を行います。再送時にエラーが発生した場合にはエラーループに入ります。

ホストからの送信データには 1 バイトずつ[ACK]コマンドが返りますので、応答を受信してから送信を行う限りオーバーランエラーは発生しません。このエラーが発生した場合は制御方法や、通信設定に問題があるものと思われます。

フレーミングエラーとパリティエラーを検出すると本モジュールは[RESEND]コマンドを送り、再送要求を行います。ホストからは再度データを送信してください。再送時にもエラーが発生した場合には次に説明する「エラーループ」に入ります。

モジュールはオーバーランエラーを検出した場合や、再送データにもエラーを検出した場合には、正しい通信が行えない状態にあるものと判断して、“エラーループ”に入り[COM_ERR]コマンドを送信します。エラーループに入っている間は[SET_DEFAULT]コマンドと[READ_STATUS]コマンドのみ受け付けられます。[SET_DEFAULT]コマンドを受信すると、初期化した後コマンドループに復帰します。

コマンドループに復帰するもう一つの方法は、[READ_STATUS]コマンドを 2 度呼び出すことです。1 度目の[READ_STATUS]コマンドのステータスは、発生した通信エラーと対応したビットが“1”となります。このとき通信エラーが発生しなければ、2 度目の[READ_STATUS]コマンドに対する応答では、エラービットがクリアされたステータスが返ります。2 度続けてコマンドが正しく処理されると、エラーループを抜けて通常のコマンドループに復帰します。

□ コマンドリファレンス

以下に各コマンドの説明とコマンドシーケンスを示します。

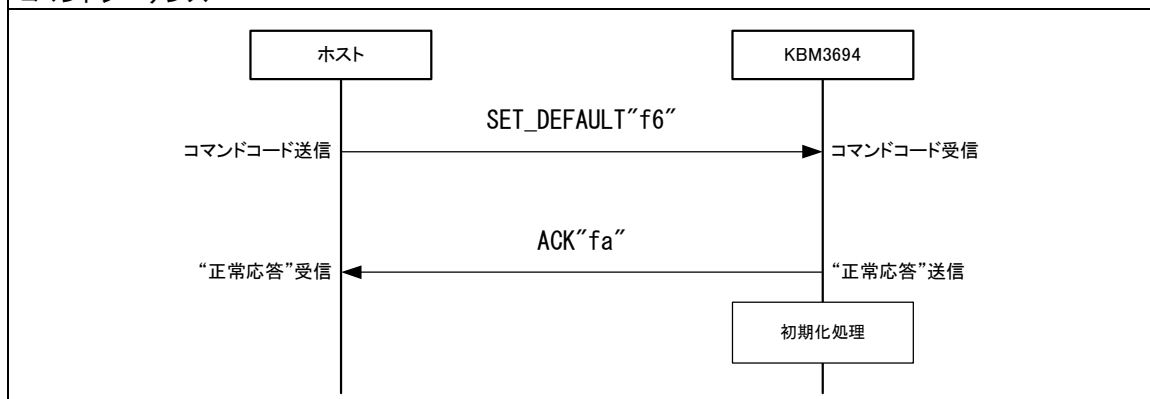
SET_DEFAULT

コマンド名	コード(HEX)
SET_DEFAULT	f6

説明

モジュールを初期状態に戻します。基本保証テスト(BAT)は行われません。

コマンドシーケンス



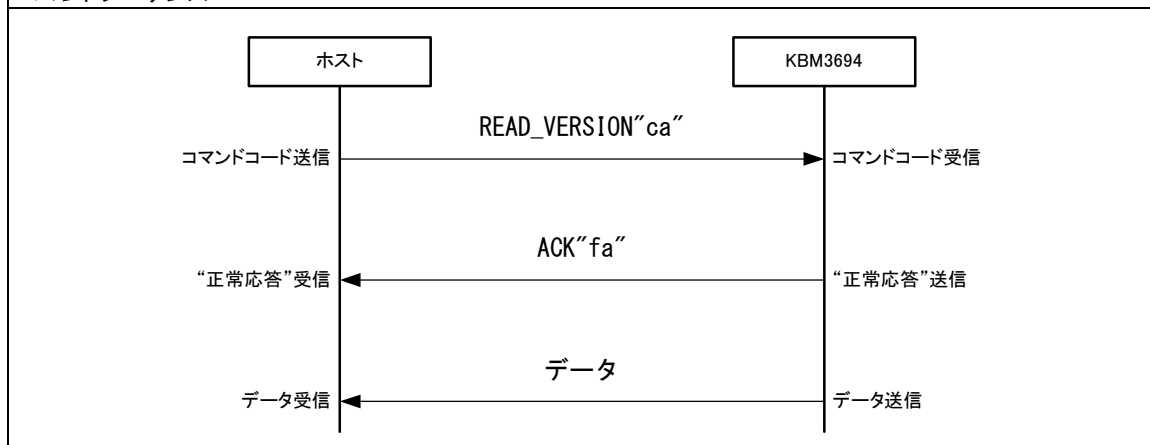
READ_VERSION

コマンド名	コード(HEX)
READ_VERSION	ca

説明

バージョン情報を読み出します。データの低位 4 ビットはマイナーバージョン、上位 4 ビットがメジャーバージョンとなっています。現在は 10(HEX)が返ります。

コマンドシーケンス



READ_STATUS

コマンド名	コード(HEX)
READ_STATUS	cb

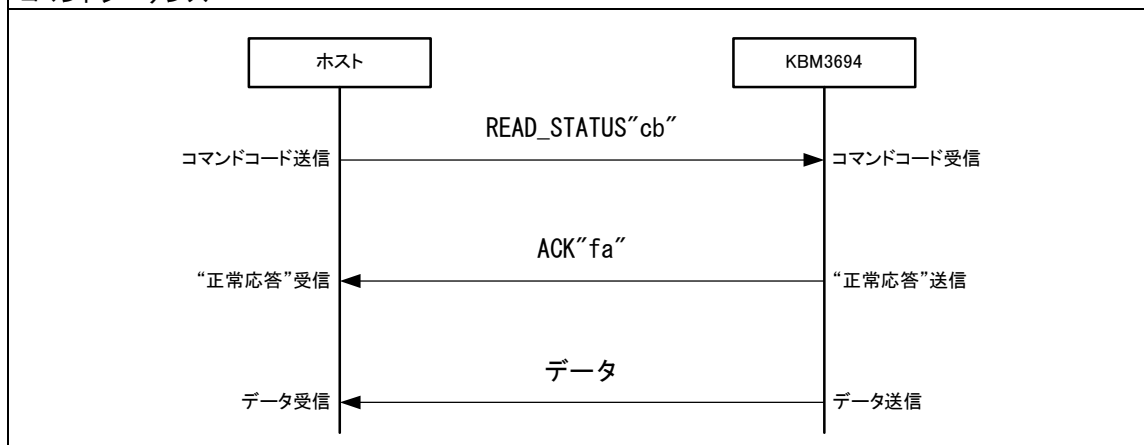
説明

ステータスを読み出します。データの意味は以下のとおりです。

ビット	意味
0	ホストからのデータ待ちでタイムアウトが発生すると1になります。
1	シリアル通信でパリティエラーが発生すると1になります。
2	シリアル通信でフレーミングエラーが発生すると1になります。
3	シリアル通信でオーバーランエラーが発生すると1になります。
4-6	予約(不定)
7	キースキャンがイネーブルされているときは1になります。

ビット 0-3 はステータスを読み出すとクリアされます。

コマンドシーケンス



ECHO

コマンド名	コード(HEX)
ECHO	ee

説明

エコー応答を確認します。

コマンドシーケンス



SET_MODE0

コマンド名	コード(HEX)
SET_MODE0	cf

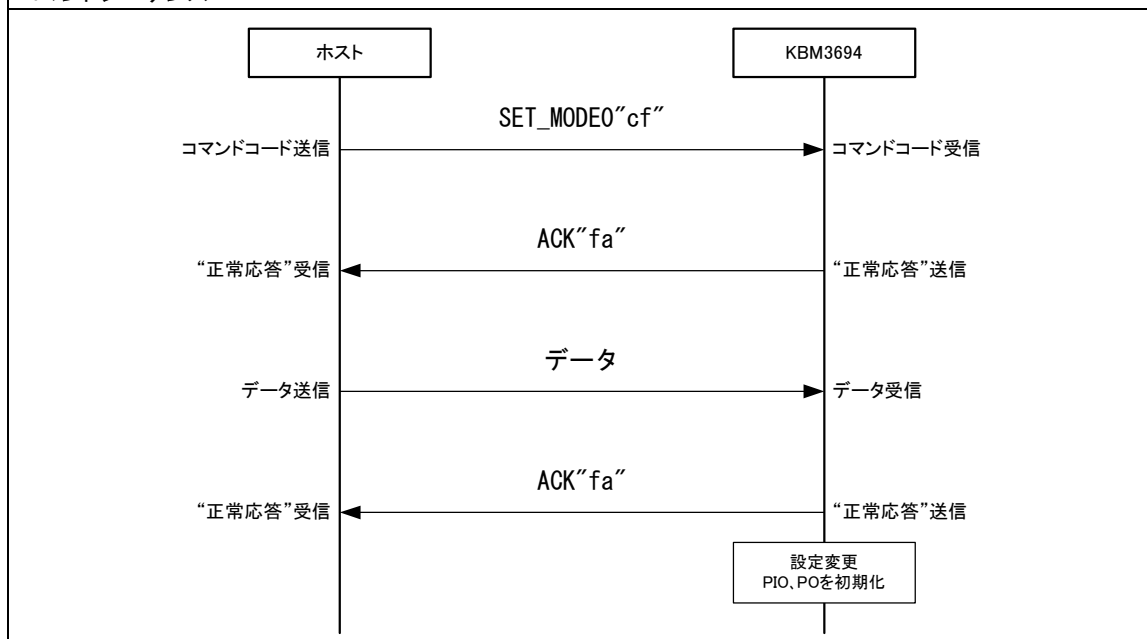
説明

動作オプションを変更します。コマンドに続いて 1 バイトの設定データを送ります。データの意味は以下のとおりです。

ビット	意味	出荷時の設定
0	1 とすると起動時に BAT 終了コードを PS/2 ポートに送りません。	0
1	1 とすると識別できない PS/2 コマンドにも ACK で応答します。	0
2	1 とすると PIO 端子を LCD インタフェースに設定します。	1
3	1 とすると PO4-7 端子を出力ポートとして使用します。	1
4	1 とすると PIO-6 端子を COL9-15#入力としてスキャンします。	0
5	1 とするとインジケータの設定値を PIO4-7 に出力します。	0
6	1 とするとキーコードをシリアルに出力します。	0
7	予約(0 としてください)	0

このコマンドが送られると PIO と PO の各端子は設定に合わせて初期化されます。

コマンドシーケンス



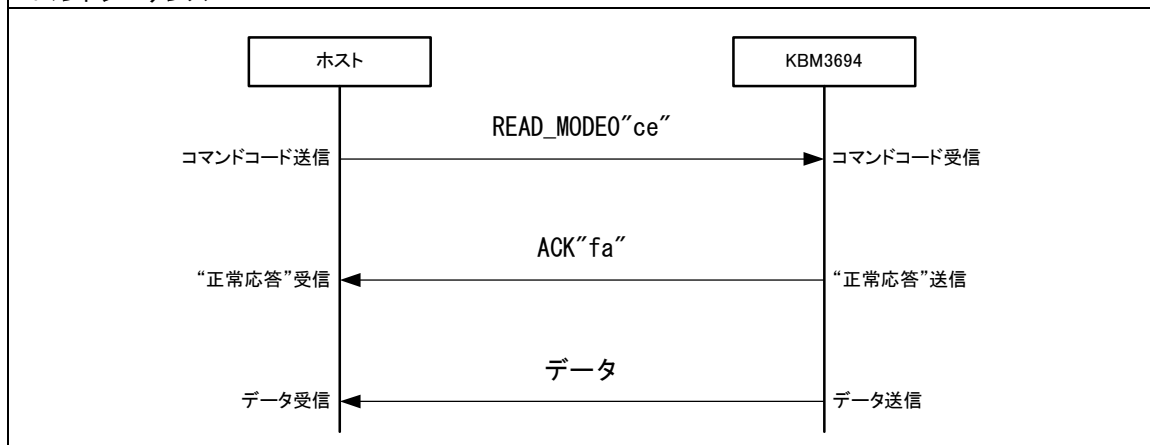
READ_MODE0

コマンド名	コード(HEX)
READ_MODE0	ce

説明

現在の動作オプションを読み出します。読み出したデータの意味は「SET_MODE0」コマンドを参照してください。

コマンドシーケンス



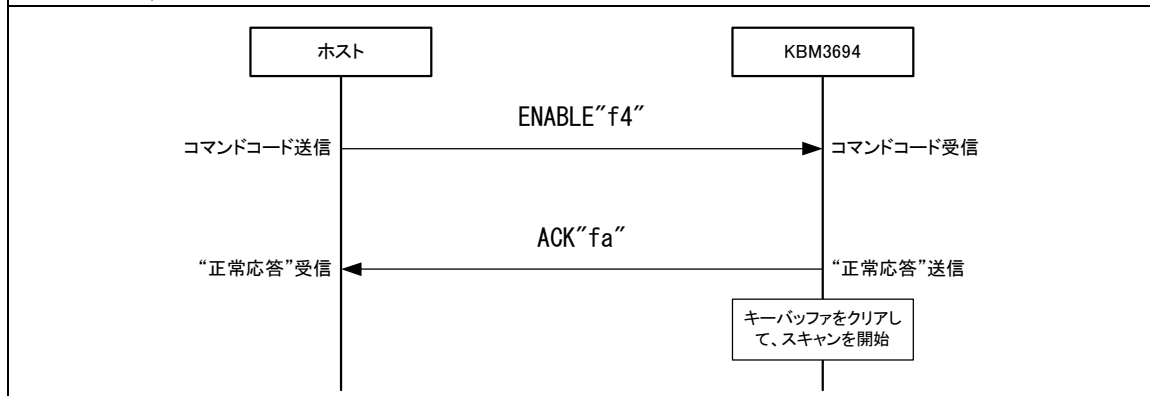
ENABLE

コマンド名	コード(HEX)
ENABLE	f4

説明

キーバッファをクリアして、キースキャンを開始します。電源投入時、は自動的にキースキャンが開始されます。

コマンドシーケンス



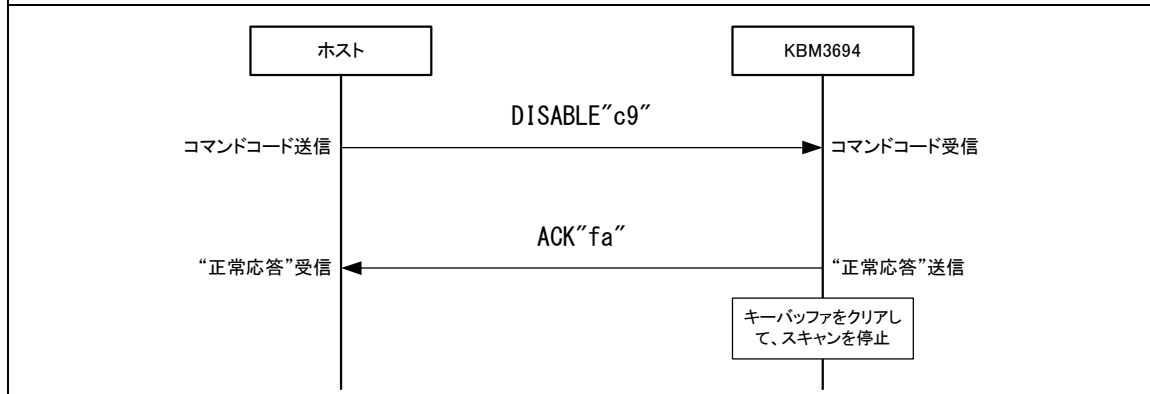
DISABLE

コマンド名	コード(HEX)
DISABLE	c9

説明

キーバッファをクリアして、キースキャンを停止します。

コマンドシーケンス

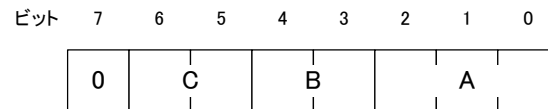


SET_TR_DLY

コマンド名	コード(HEX)
SET_TR_DLY	f3

説明

タイパマティック・ディレイ/レートを設定します。コマンドの後に図に示す 1 バイトのデータを送信します。



タイパマティック動作中の出力間隔 T_r と タイパマティック・ディレイ時間 T_d は次の式で求められます。

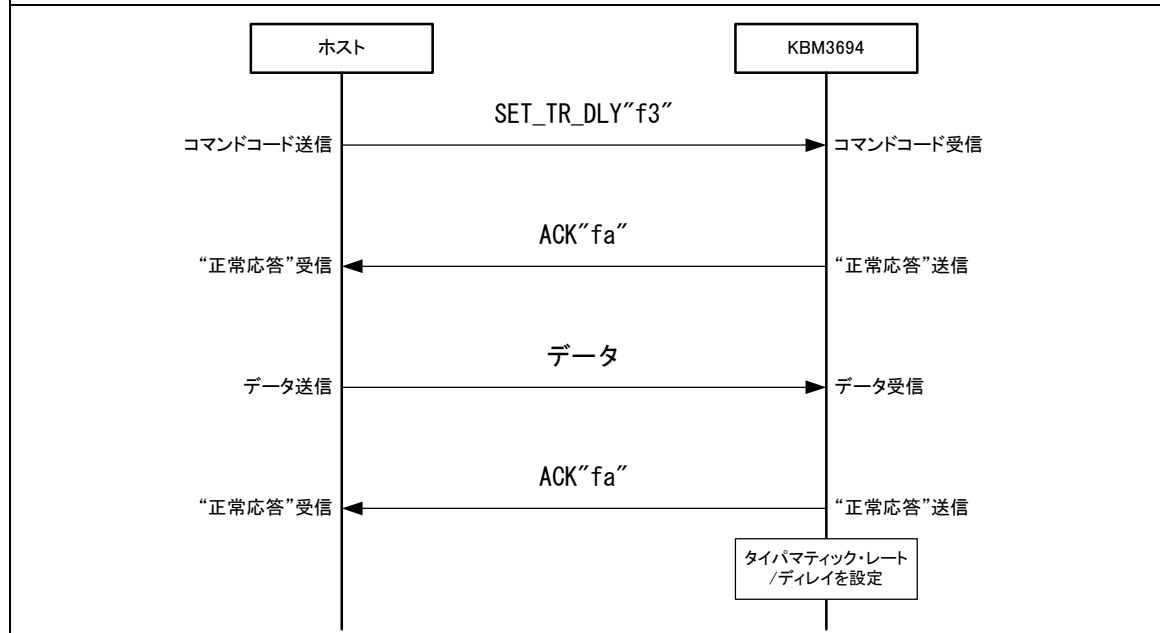
$$T_r = (8 + A) \times 2^B \times 0.00417 \text{ [s]}$$

$$T_d = (1 + C) \times 250 \text{ [ms]}$$

A,B,C: 上の図の各ビットの値

値は設定可能なものにまるめられるため、誤差が生じます。

コマンドシーケンス



INDICATOR

コマンド名	コード(HEX)
INDICATOR	ed

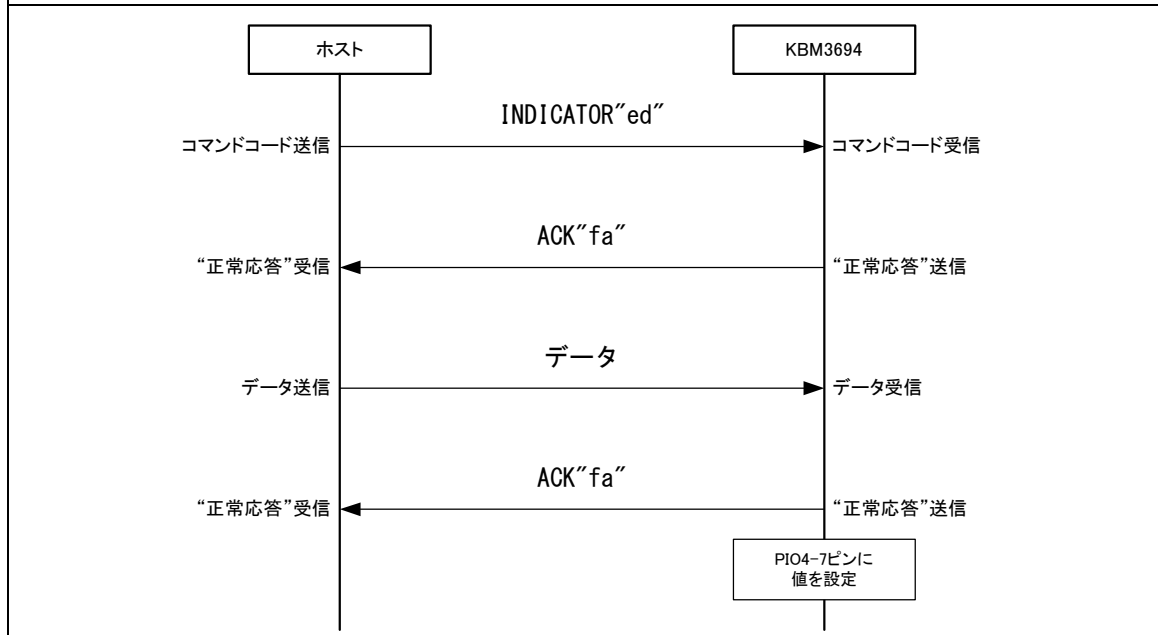
説明

インジケータ(LED)の設定を行います。コマンドに続いて 1 バイトのデータを送信します。データの意味を表に示します。

ビット	意味
0	Scroll Lock
1	Num Lock
2	Caps Lock
3	Kana (AX キーボード)
4-7	0

PIO ポートがLCD インタフェースに割り当てられていない場合で、PIO4-7にインジケータを割り当てるようにオプション設定されている場合のみ、PIO4-7 ピンにデータの 0-3 ビットの値が反映されます。

コマンドシーケンス



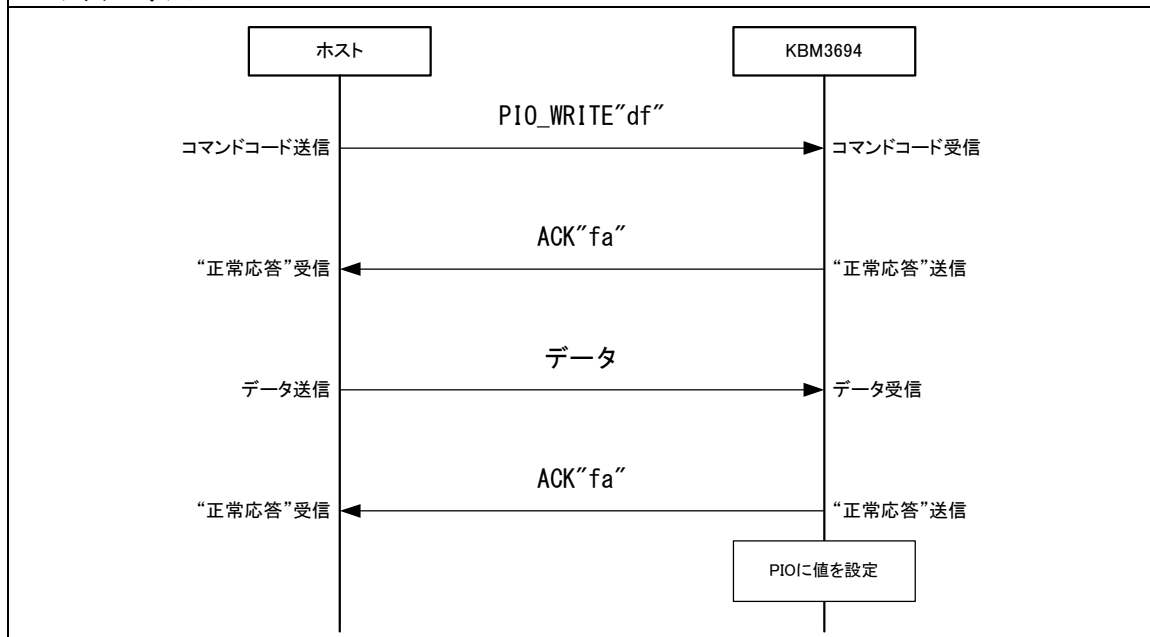
PIO_WRITE

コマンド名	コード(HEX)
PIO_WRITE	df

説明

PIO ポートに値を書き込みます。出力に設定されている端子にのみ影響を与えます。コマンドに続いて PIO ポートに設定するデータを送信します。PIO ポートが LCD インタフェースに設定されている場合は、コマンドは無視されますが、PIO4-7 がインジケータに設定されている場合は、通常どおり書き換えられます。

コマンドシーケンス



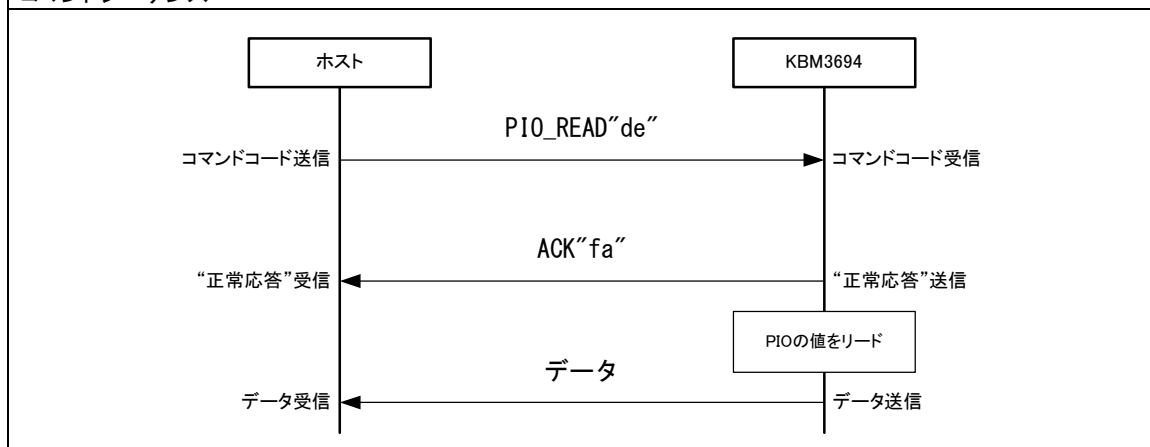
PIO_READ

コマンド名	コード(HEX)
PIO_READ	de

説明

PIO ポートからデータを読みます。データのビット 3 は不定です。

コマンドシーケンス



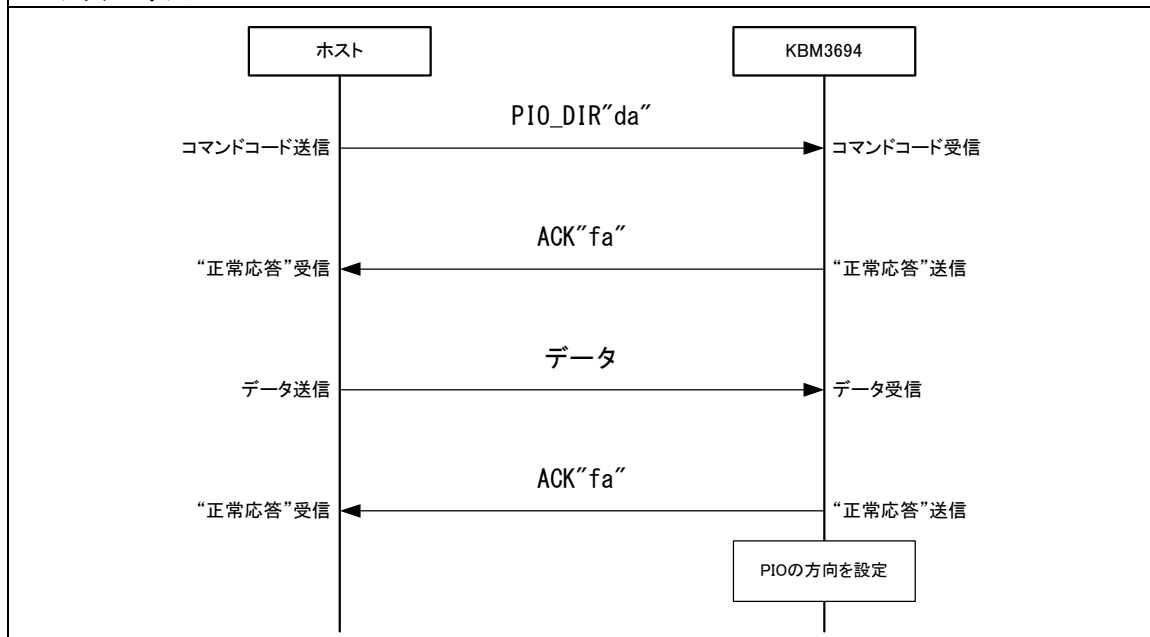
PIO_DIR

コマンド名	コード(HEX)
PIO_DIR	da

説明

PIO ポートの入出力方向を切り替えます。コマンドに続いて PIO の各ビットの方向を示すデータを送信します。データのうち 1 となっているビットは出力に、0 となっているビットは入力に設定されます。ビット 3 は無視されます。

コマンドシーケンス



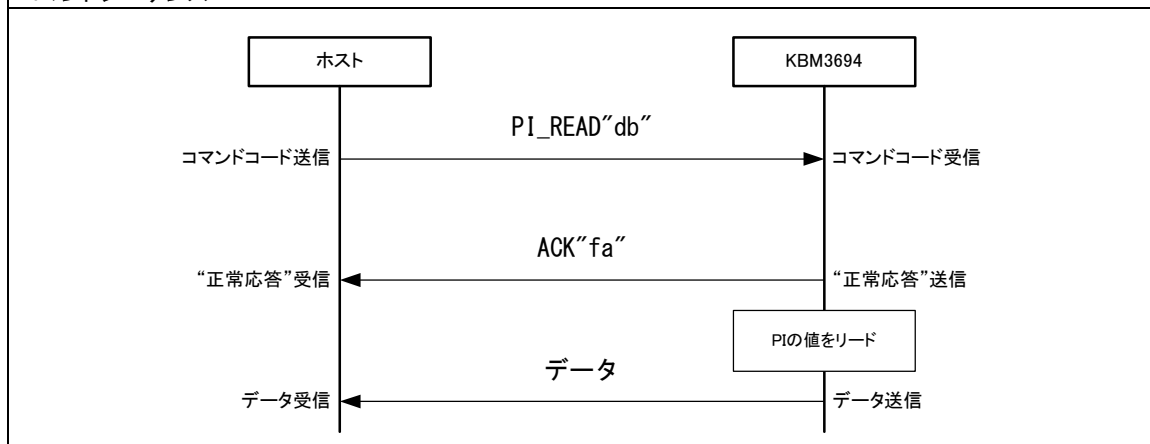
PI_READ

コマンド名	コード(HEX)
PI_READ	db

説明

PI ポートからデータを読みます。ビット 7 は不定です。

コマンドシーケンス



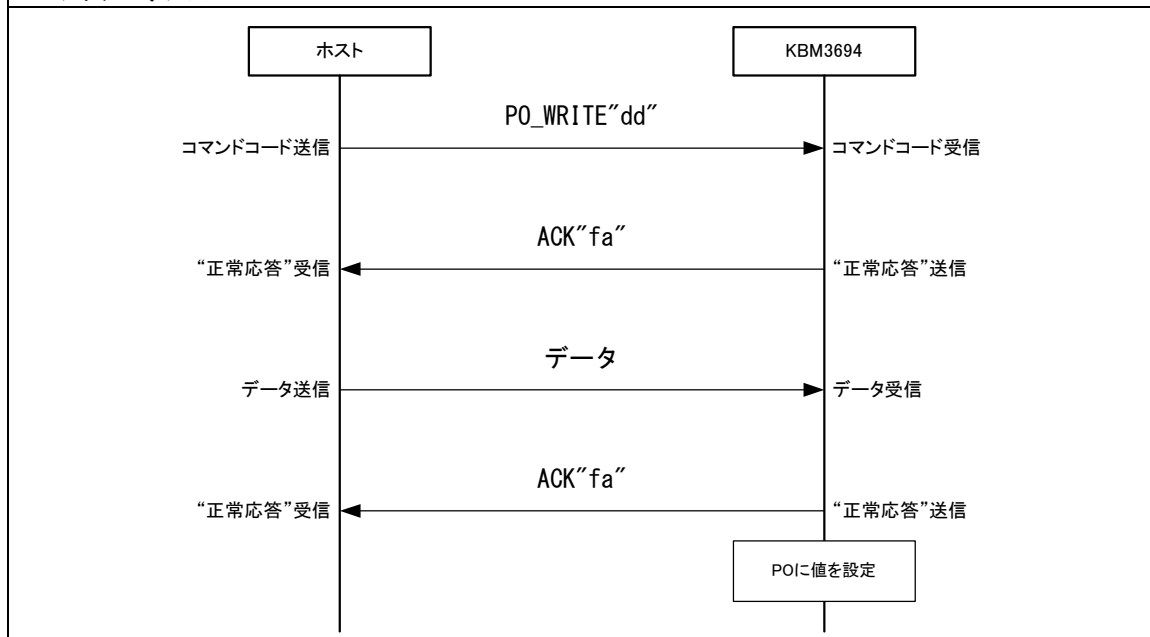
PO_WRITE

コマンド名	コード(HEX)
PO_WRITE	dd

説明

PO ポートに値を書き込みます。コマンドに続いて PO ポートに設定するデータを送信します。データの上位 4 ビットが PO4-7 ピンに反映されます。その他のビットは無視されます。PO4-7 が ROW 出力に設定されている場合は全てのビットが無視されます。

コマンドシーケンス



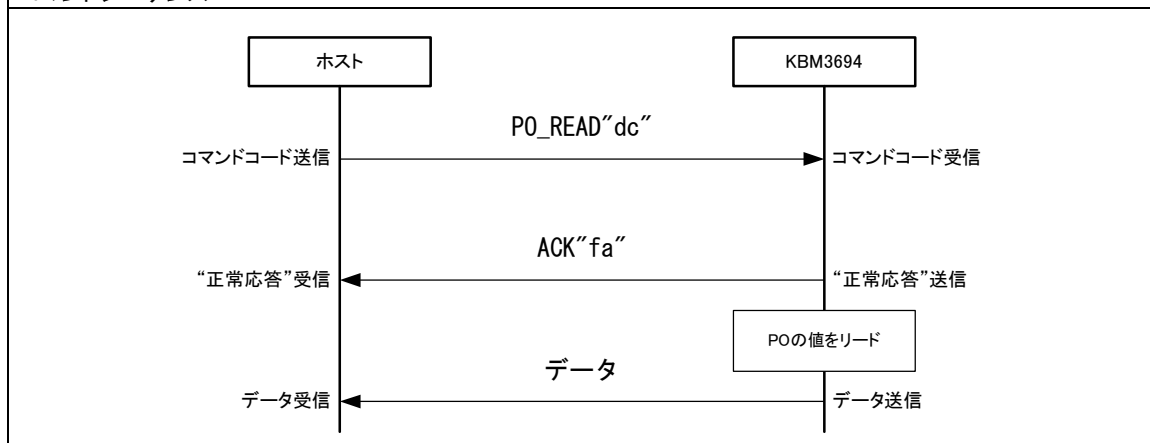
PO_READ

コマンド名	コード(HEX)
PO_READ	dc

説明

PO ポートに設定されている値を読み出します。下位 4 ビットは不定です。PO4-7 が ROW 出力に設定されている場合はデータの全てのビットが不定です。

コマンドシーケンス



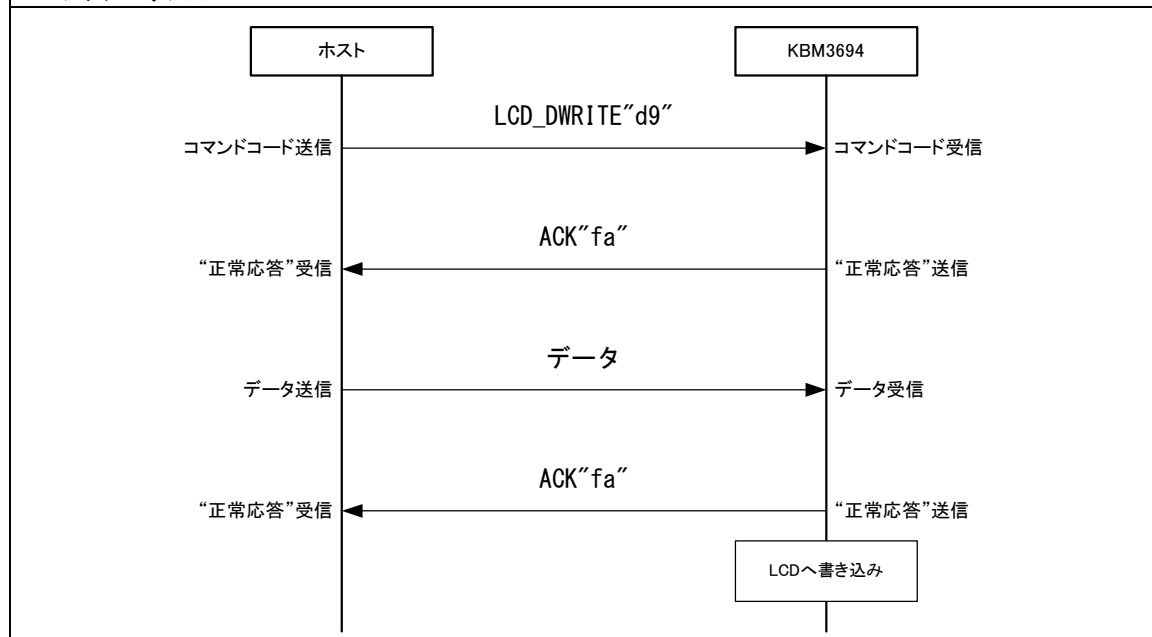
LCD_DWRITE

コマンド名	コード(HEX)
LCD_DWRITE	d9

説明

LCD のデータレジスタへ書き込みを行います。つまり RS 信号を 1 として書き込み動作を行います。コマンドに続いて書き込むデータを送信します。PIO が LCD インタフェースに設定されていない場合は無視されます。

コマンドシーケンス



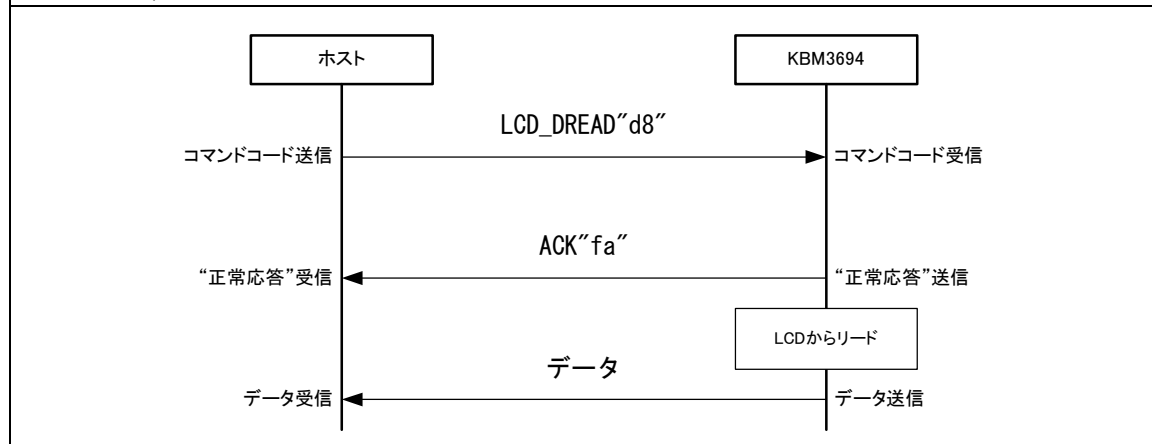
LCD_DREAD

コマンド名	コード(HEX)
LCD_DREAD	d8

説明

LCD のデータレジスタからデータを読み出します。つまり、RS 信号を 1 として読み出し動作を行います。PIO が LCD インタフェースに設定されていない場合、読み出し値は不定です。

コマンドシーケンス



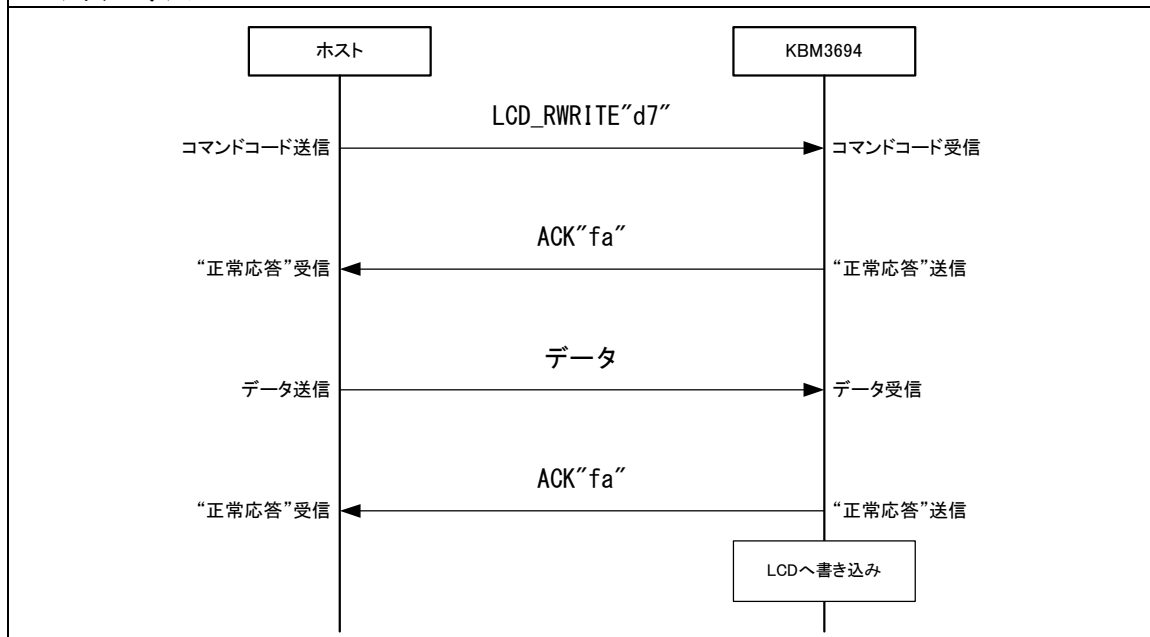
LCD_RWRITE

コマンド名	コード(HEX)
LCD_RWRITE	d7

説明

LCD のインストラクションレジスタへ書き込みを行います。つまり RS 信号を 0 として書き込み動作を行います。コマンドに続いて書き込むデータを送信します。PIO が LCD インタフェースに設定されていない場合は無視されます。

コマンドシーケンス



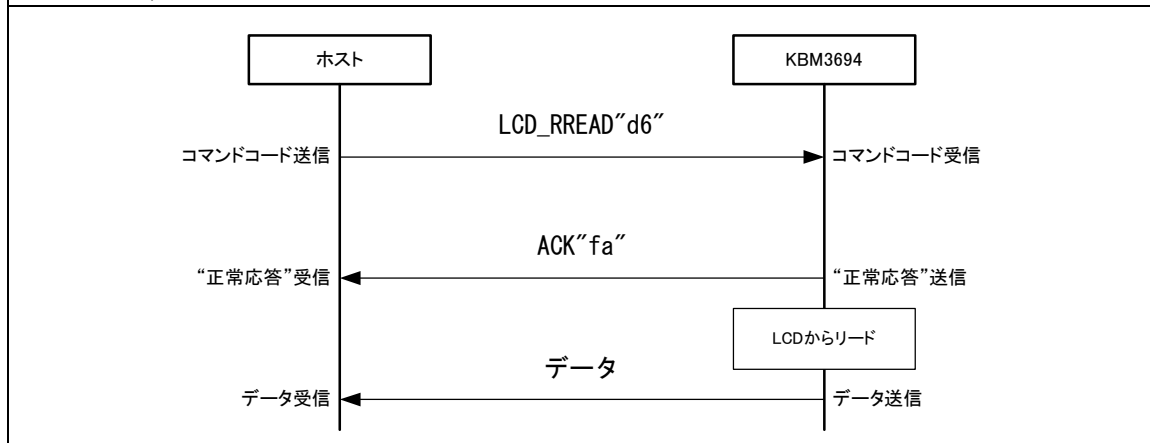
LCD_RREAD

コマンド名	コード(HEX)
LCD_RREAD	d6

説明

LCD のインストラクションレジスタからデータを読み出します。つまり、RS 信号を 0 として読み出し動作を行います。PIO が LCD インタフェースに設定されていない場合、読み出し値は不定です。

コマンドシーケンス



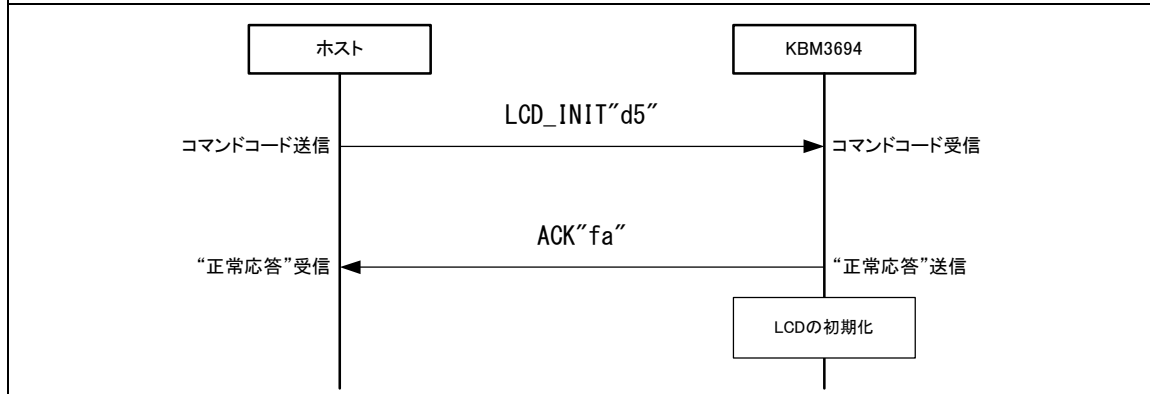
LCD_INIT

コマンド名	コード(HEX)
LCD_INIT	d5

説明

LCD の初期化処理を行います。PIO が LCD インタフェースに設定されていない場合は無視されます。起動時や PIO を LCD インタフェースに設定した場合は自動的に実行されます。

コマンドシーケンス



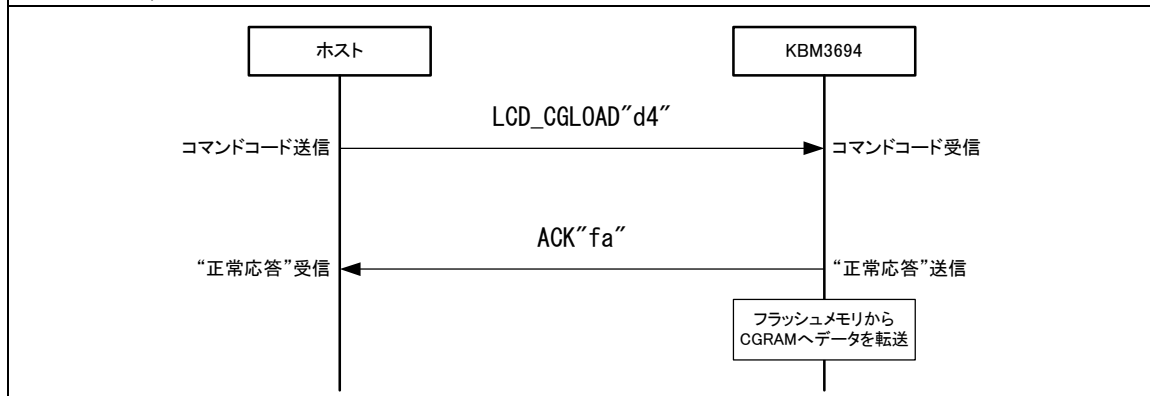
LCD_CGLOAD

コマンド名	コード(HEX)
LCD_CGLOAD	d4

説明

フラッシュメモリの定められた領域から CGRAM にキャラクタデータを転送します。PIO が LCD インタフェースに設定されていない場合は無視されます。

コマンドシーケンス



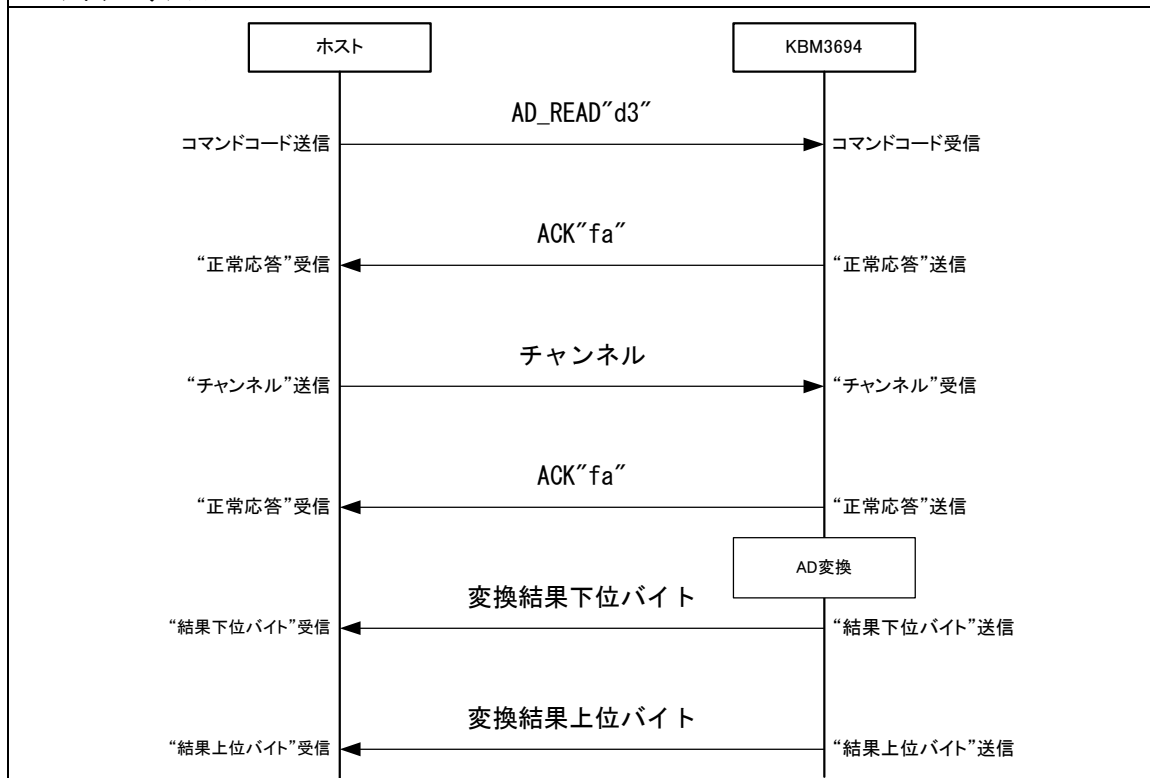
AD_READ

コマンド名	コード(HEX)
AD_READ	d3

説明

AD 変換を実行し、結果を読み出します。コマンドに続いてチャンネルデータ(0~6)を送ります。結果は下位バイト、上位バイトの順で送られます。

コマンドシーケンス



7. フラッシュメモリ

本モジュールにはキーコード、起動時の動作オプション、LCD の CGRAM データなどを保存するためのフラッシュメモリ領域が用意されています。この領域へは専用ライティングソフト『KBMWriter』を利用してデータの書き込みを行うことができます(『KBMWriter』は弊社ホームページよりダウンロードしていただけます)。

□ 『KBMWriter』使用方法

図 15 に『KBMWriter』の動作画面を示します。書き込み操作を行う前に、まずデータファイルを用意します。データファイルはテキスト形式ですので「メモ帳」などのエディタで編集することが可能です。書式は「APPENDIX」をご参照ください。また、『KBM3694 スターターキット』には専用の編集ソフト『KBMEEditor』が付属しますので、GUI で簡単にデータファイルを作成することができます。

1. 本モジュールとパソコンのシリアルポートを接続します。
2. 本モジュールをパソコンの PS/2 ポートと接続する、または VCC と GND 端子に電源を接続するなどして本モジュールを起動します。
3. [参照]ボタンを押して書き込むデータファイルを選択します。
4. 使用する“COM ポート”と通信速度を選択してください。通信速度は出荷時の設定では 38400[bps]です。以前の書換えにより変更した場合にはその値を選択します。
5. [書き込み]ボタンを押してデータを書き換えます。書き込み中は絶対に本モジュールの電源を切らないでください。

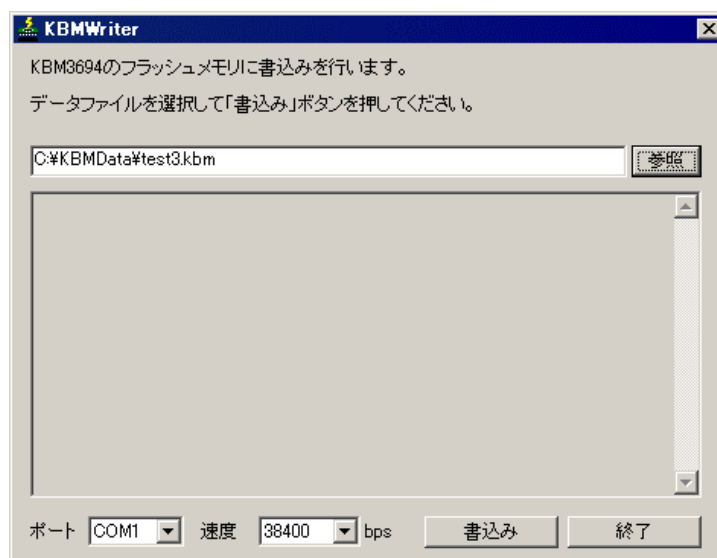


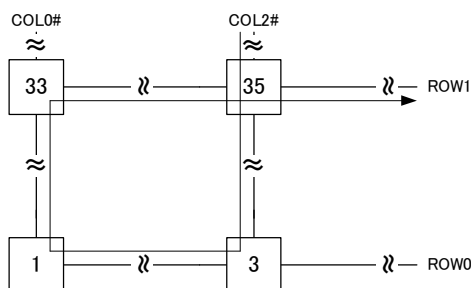
図 15 『KBMWriter』の動作画面

以前に書き込んだ通信速度がわからなくなってしまった場合や、何らかの理由で書き込みに失敗し、正常に書き換えが行えなくなってしまった場合は、「J9」のジャンパースイッチを“ON”にして本モジュ

ールを起動してください。自動的にフラッシュメモリの書き換えモードに移行して立ち上がります(通常の操作はできません)。書き換えは前記の手順に従いますが、通信速度は必ず 38400[bps]に設定します。書き換えが終了したら「J9」は必ず“OFF”にしてください。

“ゴースト”キー

「KBM3694」と同じ原理でキースキャンする全てのキーエンコーダに共通して、“ゴースト”キーと呼ばれる現象が発生します。これは 3 つのキーを同時に押した場合に、特定の位置にある 4 つ目のキーが押されているように見える現象です。例えば図 4 の 1、3、33 番のキーが押されると四角形の残りの頂点となる 35 番のキーも押されているように見えます。下図のように電流が流れ、COL2#の端子が“Low”となってしまうことが原因ですが、これを完全に取り除くには各スイッチにダイオードを入れる必要があり、コストがかかります。



実用的な対処方法として、シフトキーやコントロールキーのように、他のキーと同時に押すことの多いキーには独立した、行か列を割り当てる方法があります。例えば、COL0#の列にシフトキーを配置したとき、その列に他のキーを割り当てなければ、上のような電流経路ができなくなります。

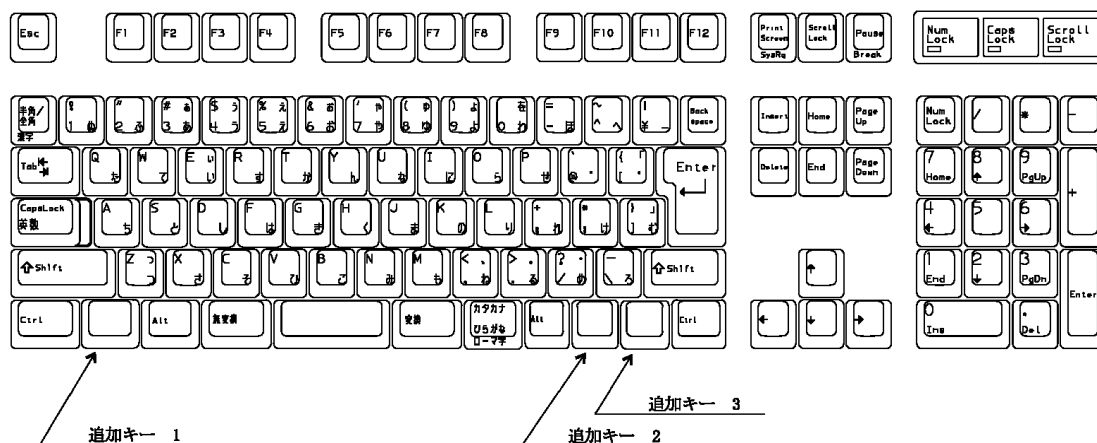
Appendix

□ キーボード配列と走査コード

以下に参考資料として「OADG109A 型キーボード」の刻印とキー番号を掲載します。また、走査コードについては一般的に使用されている「走査コード・セット 2」を掲載します。

出典：「OADG テクニカル・リファレンス（ハードウェア）」⁵
(PC オープン・アーキテクチャー推進協議会 発行)

キーボード配列



注：ピッチ及び刻印は定義しない。

図 16 OADG109A 型キーボードの刻印



注：追加キー 1，追加キー 2，追加キー 3 の
キー番号は割当てない。

図 17 OADG109A 型キーボードのキー番号

⁵ PC オープン・アーキテクチャー推進協議会(OADG)の許可を得て転載。

走査コード・セット 2

各キーに個別の 8 ビットのメーク走査コードが割り当てられ、キーが押されたときにシステムに送られます。またキーが放されたとき、ブレイク・コード・プリフィックス(Breake code prefix F0h)が付加されてシステムに送られます。ブレイク・コードは 2 バイトで構成され、最初のバイトがブレイク・コード・プリフィックス(F0h)で、第 2 バイトがそのキーのメーク走査コードです。タイパマティック走査コードは、そのキーのメーク走査コードです。

38 ページの図 18 は、キーボードのシフト状態やシステムの状態に影響されない、各キーの走査コードを示します。

Key Number	Make Code	Break Code	Key Number	Make Code	Break Code
1	0E	F0 0E	49	2A	F0 2A
2	16	F0 16	50	32	F0 32
3	1E	F0 1E	51	31	F0 31
4	26	F0 26	52	3A	F0 3A
5	25	F0 25	53	41	F0 41
6	2E	F0 2E	54	49	F0 49
7	36	F0 36	55	4A	F0 4A
8	3D	F0 3D	56	51	F0 51
9	3E	F0 3E	57	59	F0 59
10	46	F0 46	58	14	F0 14
11	45	F0 45	60	11	F0 11
12	4E	F0 4E	61	29	F0 29
13	55	F0 55	62	E0 11	E0 F0 11
14	6A	F0 6A	64	E0 14	E0 F0 14
15	66	F0 66	90	77	F0 77
16	0D	F0 0D	91	6C	F0 6C
17	15	F0 15	92	6B	F0 6B
18	1D	F0 1D	93	69	F0 69
19	24	F0 24	96	75	F0 75
20	2D	F0 2D	97	73	F0 73
21	2C	F0 2C	98	72	F0 72
22	35	F0 35	99	70	F0 70
23	3C	F0 3C	100	7C	F0 7C
24	43	F0 43	101	7D	F0 7D
25	44	F0 44	102	74	F0 74
26	4D	F0 4D	103	7A	F0 7A
27	54	F0 54	104	71	F0 71
28	5B	F0 5B	105	7B	F0 7B
29(Reserved)	5D	F0 5D	106	79	F0 79
30	58	F0 58	108	E0 5A	E0 F0 5A
31	1C	F0 1C	110	76	F0 76
32	1B	F0 1B	112	05	F0 05
33	23	F0 23	113	06	F0 06
34	2B	F0 2B	114	04	F0 04
35	34	F0 34	115	0C	F0 0C
36	33	F0 33	116	03	F0 03
37	3B	F0 3B	117	0B	F0 0B
38	42	F0 42	118	83	F0 83
39	4B	F0 4B	119	0A	F0 0A
40	4C	F0 4C	120	01	F0 01
41	52	F0 52	121	09	F0 09
42	5D	F0 5D	122	78	F0 78
43	5A	F0 5A	123	07	F0 07
44	12	F0 12	125	7E	F0 7E
45(Reserved)	61	F0 61	131	67	F0 67
46	1A	F0 1A	132	64	F0 64
47	22	F0 22	133	13	F0 13
48	21	F0 21			

図 18 キーボード走査コード・セット 2 (1/6)

残りのキーは、シフト・キー(Ctrl, Alt, Shift)や Num Lock (オンまたはオフ)の状況に応じて、異なる一連の走査コードをシステムに送ります。基礎走査コードは他のキーと同じなので、基礎走査コードに付加コード(E0h)を追加して個別化しています。

Key No.	Base Case, or Shift+Num Lock Make/Break	Shift Case Make/Break	Num Lock on Make/Break
75	E0 70 /E0 F0 70	E0 F0 12 E0 70 /E0 F0 70 E0 12*	E0 12 E0 70 /E0 F0 70 E0 F0 12
76	E0 71 /E0 F0 71	E0 F0 12 E0 71 /E0 F0 71 E0 12*	E0 12 E0 71 /E0 F0 71 E0 F0 12
79	E0 6B /E0 F0 6B	E0 F0 12 E0 6B /E0 F0 6B E0 12*	E0 12 E0 6B /E0 F0 6B E0 F0 12
80	E0 6C /E0 F0 6C	E0 F0 12 E0 6C /E0 F0 6C E0 12*	E0 12 E0 6C /E0 F0 6C E0 F0 12
81	E0 69 /E0 F0 69	E0 F0 12 E0 69 /E0 F0 69 E0 12*	E0 12 E0 69 /E0 F0 69 E0 F0 12
83	E0 75 /E0 F0 75	E0 F0 12 E0 75 /E0 F0 75 E0 12*	E0 12 E0 75 /E0 F0 75 E0 F0 12
84	E0 72 /E0 F0 72	E0 F0 12 E0 72 /E0 F0 72 E0 12*	E0 12 E0 72 /E0 F0 72 E0 F0 12
85	E0 7D /E0 F0 7D	E0 F0 12 E0 7D /E0 F0 7D E0 12*	E0 12 E0 7D /E0 F0 7D E0 F0 12
86	E0 7A /E0 F0 7A	E0 F0 12 E0 7A /E0 F0 7A E0 12*	E0 12 E0 7A /E0 F0 7A E0 F0 12
89	E0 74 /E0 F0 74	E0 F0 12 E0 74 /E0 F0 74 E0 12*	E0 12 E0 74 /E0 F0 74 E0 F0 12

* If the left Shift key is held down, the F0 12/12 shift break and make is sent with the other scan codes. If the right Shift key is held down, F0 59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

図 19 キーボード走査コード・セット 2 (2/6)

Key No.	Scan Code Make/Break	Shift Case Make/Break
95	E0 4A/E0 F0 4A	E0 F0 12 E0 4A/E0 F0 4A E0 12*

* The left Shift key is held down, the F0 12/12 shift break and make is sent with the other scan codes. If the right Shift key is held down, F0 59/59 is sent. If both Shift keys are down, both sets of code are sent with the other scan code.

図 20 キーボード走査コード・セット 2 (3/6)

Key No.	Scan Code Make/Break	Ctrl Case, Shift Case Make/Break	Alt Case Make/Break
124	E0 12 E0 7C /E0 F0 7C E0 F0 12	E0 7C/E0 F0 7C	84/F0 84

図 21 キーボード走査コード・セット 2 (4/6)

Key No.	Make Code	Ctrl Key Pressed
126*	E1 14 77 E1 F0 14 F0 77	E0 7E E0 F0 7E
* This key is not typematic. All associated scan codes occur on the make of the key.		

図 22 キーボード走査コード・セット 2 (5/6)

	Make Code	Break Code
追加キー 1	E0 1F	E0 F0 1F
追加キー 2	E0 27	E0 F0 27
追加キー 3	E0 2F	E0 F0 2F

図 23 キーボード走査コード・セット 2 (6/6)

□ フラッシュメモリ書換え用データフォーマット

図 24 にフラッシュメモリ書換え用データファイルの例を示します。赤字は説明のために加えたものです。また、“//”以降はコメントです。『KBMWriter』にはサンプルファイルが付属していますので、必要な部分だけを書き換えてご使用ください。

[illegible]

図 24 フラッシュ書換え用データファイル例

- ① ファイル識別用の記述です。削除しないでください。
- ② 起動時の動作オプションです。以降、全てのデータは 16 進値をテキストで記述したものになります。最初のバイトは予約です。“00”としてください。2 バイト目は[SET_MODE0]コマンドで設定するデータと同様の値です。
- ③ シリアル通信の通信速度を設定するためのフィールドです。初期値は“0F”(38400[bps])です。

表 20 通信速度の設定値

通信速度(bps)	設定値(HEX)
4800	81
9600	40
14400	2A
19200	20
38400	0F
57600	0A

- ④ PO ポートの起動時の状態を設定するためのフィールドです。上位 4 ビットが PO4～PO7 に反映されます。
- ⑤ 予約フィールドです。252 バイト分“FF”とします。
- ⑥ LCD モジュールに CGRAM データをロードするデータを保存する領域です。[LCD_CGLOAD]コマンドを使用すると、この領域のデータが LCD モジュールの CGRAM 領域に転送されます。1 つのキャラクタを設定するのに 8 バイトのデータを使用し、8 つまでキャラクタを登録できます。使用しない領域は“00”としてください。図 25 にキャラクタデータの例を示します。

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

図 25 CGRAM データの設定例

- ⑦ キーコードの設定領域です。1 つのキーにつき 64 バイトのデータ領域があります。前半の 32 バイトはメークコード用、後半の 32 バイトはブレイクコード用です。メークコード、ブレイクコードそれぞれの領域の先頭バイトはコードのバイト数を表します。さらにメークコードの先頭バイトの最上位ビ

ットは、そのキーがタイパマティック・キーであることを表すフラグビットです(図 26 参照)。図 24 の例では最初のキーに「109 キーボード」の“0”キー(キーコード:45)をタイパマティック・キーとして登録しています。使用しないキーは先頭バイトも含めて全て“FF”としてください。

ビット	7	6	5	4	3	2	1	0
意味	フラグ	メーカーコードのバイト数						

1のときタイパマティック・キーとなる

図 26 メーカーコード・データの1バイト目

保証期間

本製品の保証期間は、お買い上げ日より 1 年間です。保証期間中の故障につきましては、無償修理または代品との交換で対応させていただきます。ただし、以下の場合は保証期間内であっても有償での対応とさせていただきますのでご了承ください。

- (1) 本マニュアルに記載外の誤った使用方法による故障。
- (2) 火災、震災、風水害、落雷などの天災地変および公害、塩害、ガス害などによる故障。
- (3) お買い上げ後の輸送、落下などによる故障。

サポート情報

製品に関する情報、最新のファームウェア、ユーティリティなどは弊社ホームページにてご案内しております。また、お問い合わせ、ご質問などは下記までご連絡ください。

テクノウェーブ(株)

URL : <http://www.techw.co.jp>

E-mail : support@techw.co.jp

参考資料：

「OADG テクニカル・リファレンス（ハードウェア）」
(PC オープン・アーキテクチャー推進協議会 発行)

改訂記録

年月	版	改訂内容
2006 年 2 月	初	
2009 年 4 月	2	・ サポート情報を修正