SmartRA 学習キット
チュートリアル 3

ルネサス エレクトロニクス社　RA マイコン搭載
HSB シリーズマイコンボード　評価キット

-本書を必ずよく読み、ご理解された上でご利用ください

株式会社北斗電子

REV.1.0.0.0

ー目　次ー

注意事項

本書を必ずよく読み、ご理解された上でご利用ください

【ご利用にあたって】

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

【限定保証】

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

【保証規定】

保証期間内でも次のような場合は保証対象外となり有料修理となります

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

【免責事項】

弊社は特定の目的･用途に関する保証や特許権侵害に対する保証等、本保証条件以外のものは明示・黙示に拘わらず一切の保証は致し兼ねます。また、直接的・間接的損害金もしくは欠陥製品や製品の使用方法に起因する損失金・費用には一切責任を負いません。損害の発生についてあらかじめ知らされていた場合でも保証は致し兼ねます。
ただし、明示的に保証責任または担保責任を負う場合でも、その理由のいかんを問わず、累積的な損害賠償責任は、弊社が受領した対価を上限とします。本製品は「現状」で販売されているものであり、使用に際してはお客様がその結果に一切の責任を負うものとします。弊社は使用または使用不能から生ずる損害に関して一切責任を負いません。
保証は最初の購入者であるお客様ご本人にのみ適用され、お客様が転売された第三者には適用されません。よって転売による第三者またはその為になすお客様からのいかなる請求についても責任を負いません。
本製品を使った二次製品の保証は致し兼ねます。

安全上のご注意

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

表記の意味

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

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

絵記号の意味

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

警告

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

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

発煙・異音・異臭にお気付きの際はすぐに使用を中止してください。

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

⚠注意

以下のことをされると故障の原因となる場合があります。

1. 静電気が流れ、部品が破壊される恐れがありますので、ボード製品のコネクタ部分や部品面には直接手を触れないでください。
2. 次の様な場所での使用、保管をしないでください。

   ホコリが多い場所、長時間直射日光があたる場所、不安定な場所、衝撃や振動が加わる場所、落下の可能性がある場所、水分や湿気の多い場所、磁気を発するものの近く
3. 落としたり、衝撃を与えたり、重いものを乗せないでください。
4. 製品の上に水などの液体や、クリップなどの金属を置かないでください。
5. 製品の傍で飲食や喫煙をしないでください。

ボード製品では、裏面にハンダ付けの跡があり、尖っている場合があります。

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

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

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

アクセスランプがある製品では、アクセスランプの点灯中に電源を切ったり、パソコンをリセットをしないでください。

製品の故障や、データ消失の原因となります。

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

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

1. RA2L1_LCD

キャラクタ LCD を使うチュートリアルです。

キャラクタ型の LCD はグラフィック型 LCD よりもマイコン側の処理が軽く、簡単な情報表示に適しています。LCD の制御方式としては、I2C や SPI といった通信規格を使うものもありますが、本キットに付属の LCD はパラレル制御のものです。パラレル制御タイプは、I/O 端子を単純に駆動する事で制御できますので、I2C や SPI で制御するタイプより初心者向けです。

本キットに付属の LCD は、SC1602 というタイプで、電子工作の世界では非常にメジャーなキャラクタ型 LCD ですので、このタイプの LCD を使える様になると、マイコンから何か情報を出力したいという際に非常に便利です。

キット付属の CD は 5V のタイプですので、このチュートリアルでは電源電圧(VCC)を、5V（4.0 以上）とする必要があります。（USB 電源で使用する分には問題ありません）

1.1. プログラムの動作

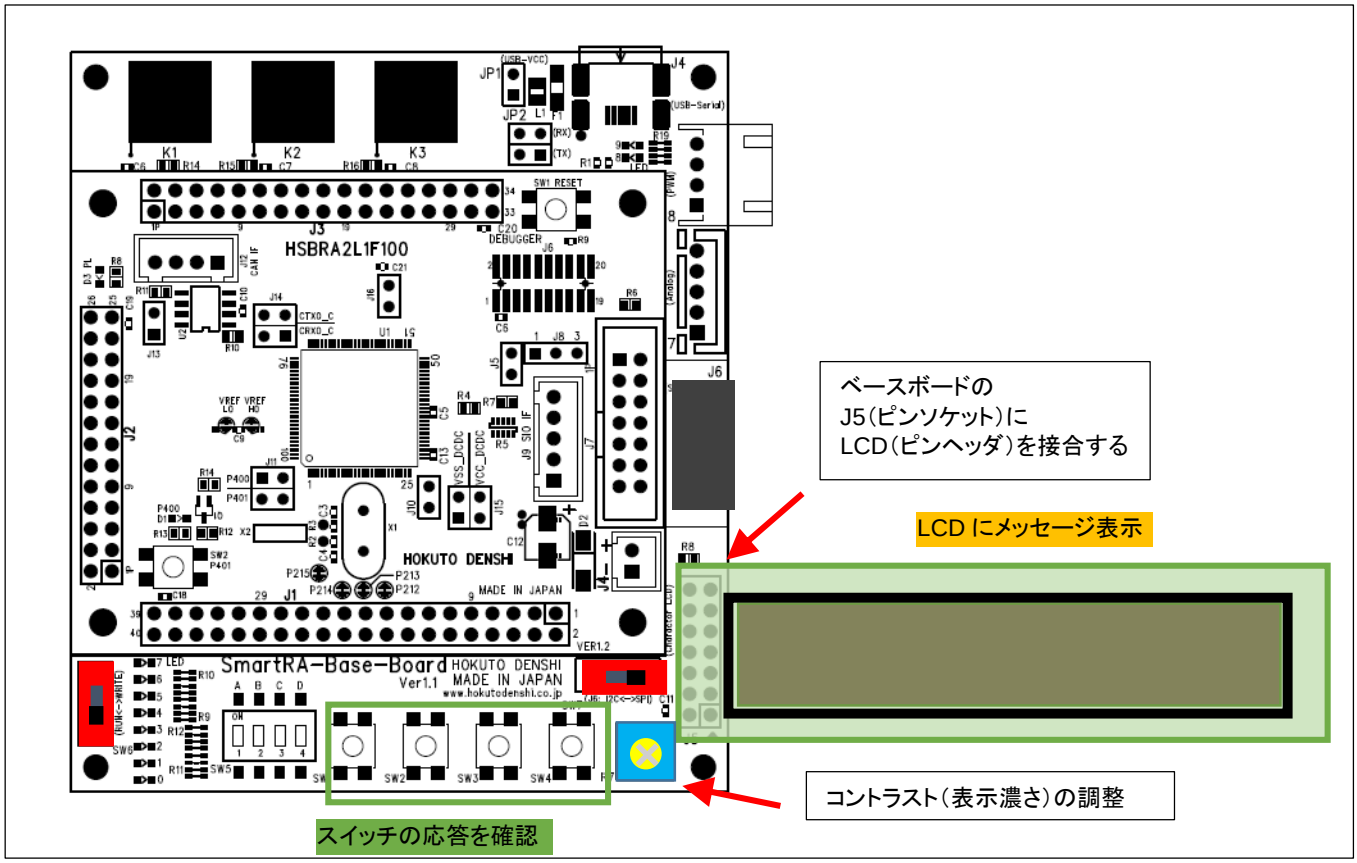
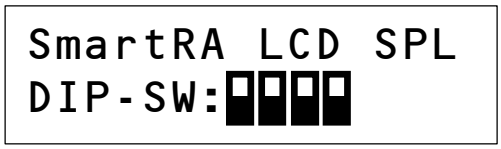


図 1-1 LCD の接続

LCD は、キットに付属致しますので、ベースボードのピンソケット（黒いコネクタ）に上からかぶせるように接合してください。

LCD 使用時は、電源電圧を 5V としてください。(LCD の動作電圧が 4.5~5.5V となっているためです。概ね 4V 以上あれば、LCD の動作は確認できます。)
(※市販の 3.3V の LCD モジュールを使用した場合は、3.3V でも LCD が使用可能です)
LCD 接続、電源投入後、可変抵抗(B-R7)を精密ドライバ等で回して、液晶の画面の濃さを調整してください。

サンプルプログラムを起動すると、LCD 画面に下記表示が出力されます。



2 行目は DIP-SW の ON/OFF に連動しており、ON:, OFF: となります。

※ はユーザ定義文字で、文字コード 0x00, 0x01 にドット単位でパターンを定義しています
(LCD には、8 文字分のユーザ定義パターンを作成可能なので、簡単な図形等の作成、表示が可能です)

DIP-SW(B-SW5-A~B-SW5-D)を全て ON 側に倒すと

の表示に変わります。

B-SW1~B-SW4 を押すと、右側の数値が加算されていきます(スイッチを押した回数を表示)。

はスイッチを押していない時のイメージ、 はスイッチを押している時のイメージです。ユーザ定義パターンで、それぞれ、文字コード 0x02, 0x03 に割り当てています。

ユーザ定義パターンは、CGRAM(後述)を使って定義していますので、具体的な定義例は、サンプルプログラムを参考にしてください。

1.2. LCD の制御方式

LCD は、マイコンと図の様に接続されています。

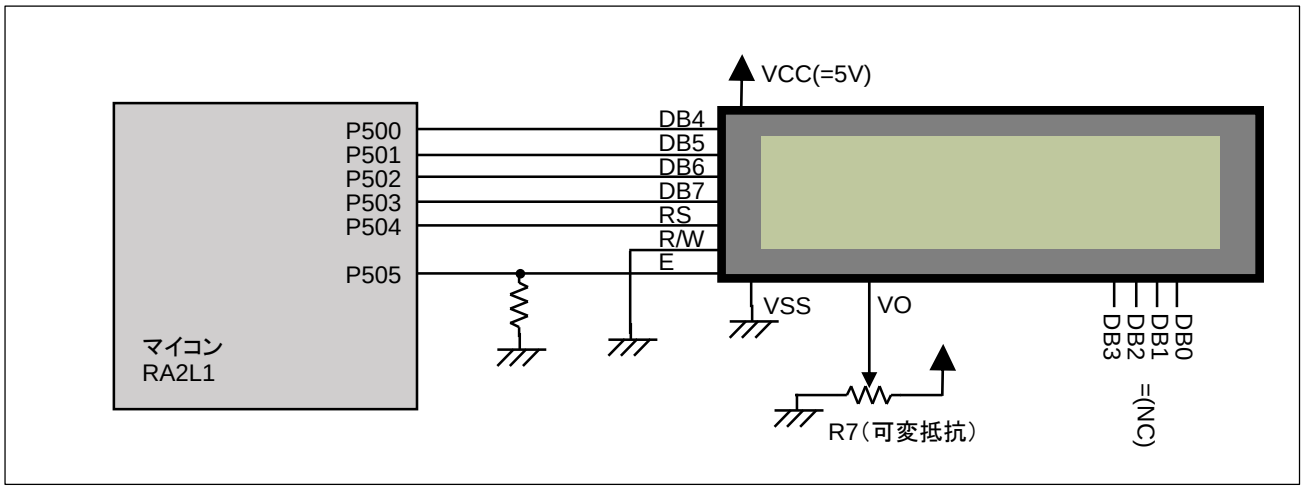


図 1-2 LCD の接続（回路図）

マイコン側は、P500~P505 の汎用 I/O を出力モードに設定して、LCD に信号を送ることで、LCD の制御を行います。

表 1-1 SC1602 LCD インタフェース信号表 (J9)

No	信号名	接続先	備考
1	VDD	VCC	
2	VSS	VSS	
3	VO	コントラスト電位	R7（可変抵抗） 時計回り：表示薄い 反時計回り：表示濃い
4	**RS**	**P504**	コマンド/キャラクタデータ切り替え
5	R/W	VSS	Read/Write 切り替え
6	**E**	**P505**	クロック信号（プルダウン）
7	DB0	(NC)	データ b0（未使用）
8	DB1	(NC)	データ b1（未使用）
9	DB2	(NC)	データ b2（未使用）
10	DB3	(NC)	データ b3（未使用）
11	**DB4**	**P500**	データ b0/b4
12	**DB5**	**P501**	データ b1/b5
13	**DB6**	**P502**	データ b2/b6
14	**DB7**	**P503**	データ b3/b7

(NC)は未接続です。

SC1602 タイプの LCD は、マイコンからコマンドを送る事により、表示や画面のクリアを行います。

DB0~DB7 の 8bit 幅でアクセスする手法と、DB4~DB7 の 4bit 幅でアクセスする手法がありますが、本ボードでは後者の手法としています。（マイコンからの信号は、太字の合計 6 本でのアクセスとなります）

基本的に、キャラクタ(A-Z, a-z, 0-9)は、8bit の文字コードを持っていますので、1 文字を送るのに、4bit で 2 回に分けて送る方式となります。

R/W は、LCD に対してデータの送信(Write)と LCD からのデータ読み出し(Read)を切り替える端子ですが、本ボードでは、LCD からのデータ読み出しはサポートしていません。ボード上で、Write 方向（マイコンから LCD にデータを送る）に固定しています。

・R/W=L のときは、マイコン→LCD へデータを送る
・R/W=H のときは、LCD→マイコンにデータを送る（R/W はボードで L 固定しているのでこのモードは使用できない）

・RS=L のときは、コマンド（画面のクリアやカーソルの移動）を送るモード
・RS=H のときは、キャラクタデータ（画面に表示する文字）を送るモード

・E（イネーブル）が、クロックの役割を行う

・DB4-7 が送信データ（4bit モードなので、8bit のデータを 2 回に分けて送信する）

となります。

LCD は、「コマンド」と「キャラクタデータ」という 2 種類のデータを受け取る仕様である、という点がポイントです。プログラム作成時は、コマンドを送る関数と、キャラクタデータを送る関数を作成すると良いと思います（サンプルプログラムではそうしています）。

LCD の仕様は、巻末に付録として記載します。

1.3. LCD 制御時の波形タイミング

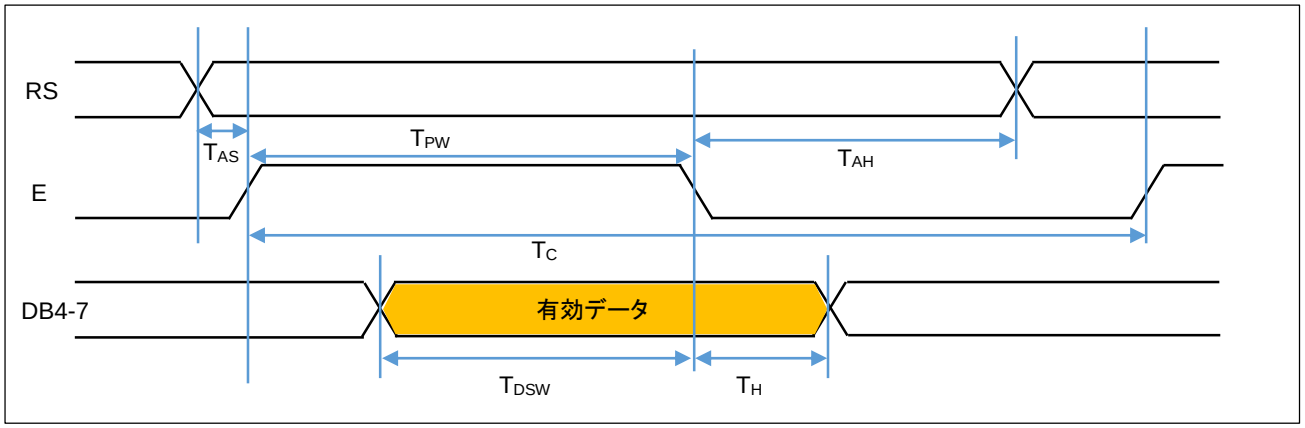


図 1-3　SC1602 LCD ユニット 制御波形

表 1-2 SC1602 LCD ユニット タイミング

シンボル	項目	min	typ	max	Unit
T_{AS}	RS セットアップ時間	0			[ns]
T_{AH}	RS ホールド時間	10			[ns]
T_C	E(イネーブル)周期	1200			[ns]
T_{PW}	E(イネーブル)パルス幅	140			[ns]
T_{DSW}	データセットアップ時間	40			[ns]
T_H	データホールド時間	10			[ns]

規定されている波形と、タイミングを示します。ここで見るべきポイントですが、

・DB4-7 にデータをセット(L/H を確定)させてから、E を H→L に変化させると、LCD ユニットはデータを取り込む
・E の H 期間は、140ns 以上必要(TPW の規定)
・E の周期は、1200ns 以上必要(TC の規定)

・RS を確定させてから、E を L→H に変化させるタイミングは 0ns 以上であればよい(TAS の規定)(*1)
・E を H→L に変化させてから、RS は最低 10ns は変化させてはいけない(TAH の規定)(*2)
・DB4-7 を確定させてから、E を H→L に変化させるタイミングは最低 40ns 以上取らなければならない(TDSW の規定)(*1)
・E を H→L に変化させてから、DB4-7 は最低 10ns は変化させてはいけない(TH の規定)(*2)
・上記タイミング規定は mim 側の規定なので、時間を長く取る分には問題はない

となります。ちょっと約束事が多い気もしますが、機械(電子回路)の動作なので、多少のルールが必要になってきます。

プログラム作成時は、上記のタイミング規定を守るようにプログラムを作成する必要があります。

1.4. LCD 制御手順

実際に、LCD に文字を表示させる手順としては、以下のようなものとなります。

－キャラクタデータの送信手順－
(1)E は初期 L としておく
(2)RS を H にする(*1)
(3)E を H に切り替える（この時点で、LCD は、RS の情報を取り込む（キャラクタデータか、コマンドかを識別）
(4)データ（上位 4bit）を DB4~7 にセットする(*2)
(5)E を L に切り替える（この時点で、LCD はキャラクタデータの上位 4bit を受け取る）
(6)E を H に切り替える
(7)データ（下位 4bit）を DB4~7 にセットする(*3)
(8)E を L に切り替える（この時点で、LCD はキャラクタデータの下位 4bit を受け取る）

(*1)コマンドを送る際は、この部分で RS=L とします

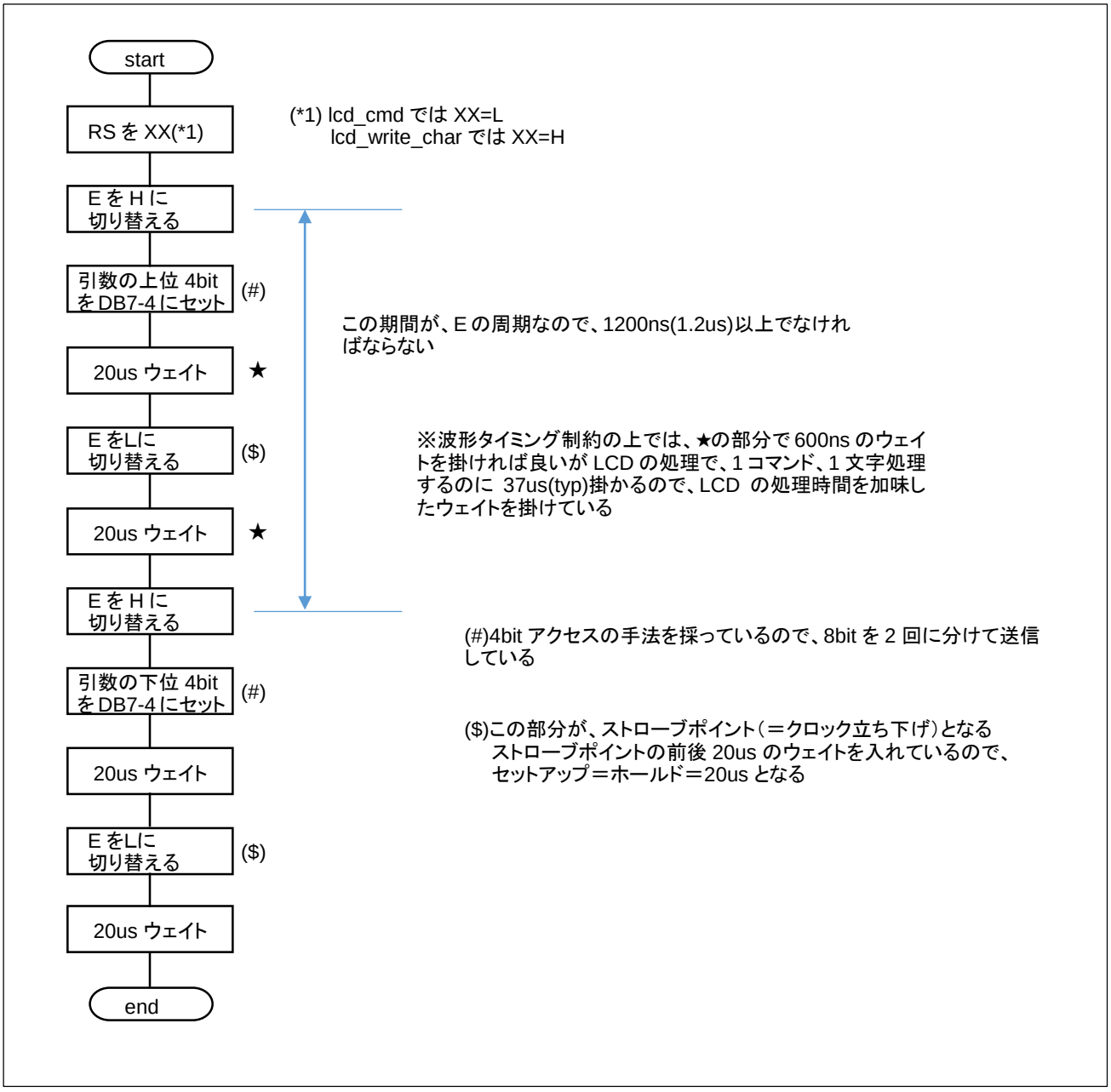
例えば、キャラクタの'1'を LCD に送る際は、'1'は ASCII コードで、0x31 なので、(*2)の時に 0x3=0b0011(DB7=DB6=L, DB5=DB4=H)を(*3)の時に、0x1=0b0001(DB7=DB6=DB5=L, DB4=H)とします。

※LCD が RS 信号を受け取るのは、E が↑（立ち上がり）のポイントで、DB7-4 の信号を受け取るのは、E が↓（立下り）のポイントなので、上記のフローの通りでなくても（例えば、(1)の後に(4)を行っても）問題はありません。
・(3)の時点で(2)が完了している事
・(5)の時点で(4)が完了している事
・(8)の時点で(7)が完了している事
さえ、守られていれば問題無い事となります。

上記手順をフローチャートにすると、以下の様になります。

1.5. LCD 制御関数フローチャート

コマンド送信関数 lcd_cmd(), キャラクタ送信関数 lcd_write_char()



キャラクタデータを送信する関数と、コマンドを送信する関数は、最初の方の RS を H に設定するか L に設定するかが異なりますが、他は同じです。

LCD 側で、文字表示を行ったり、コマンドを処理するのにある程度の処理時間(37us)掛かるので、データ送信時にある程度ウェイトを入れています。

1.6. LCD のコマンドとアドレス

LCD が認識するコマンドとしては、

コマンド	動作	備考
0b 0000 0001	全表示クリア	
0b 0000 001x	カーソルを先頭に戻す	表示内容に変化はなし
0b 0000 01111	カーソルを表示、カーソル位置の表示を点滅	
0b 1100 0000	**DDRAM アドレスを 2 行目/1 桁目に設定**	**以降送信キャラクタデータは 2 行目に表示**

等があります。詳細は、巻末の LCD の仕様を参照してください。

コマンドで最初に使用するのは、アドレスの設定だと思います。

LCD は、DDRAM（キャラクタデータを格納するメモリ）と CGRAM（ドットパターンを格納するメモリ）を持っており、マイコンからこれらの RAM にデータを送信することにより表示を行います。DDRAM に書き込んでデータ＝表示データです。

－DDRAM－

桁	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 行目	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
2 行目	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F

DDRAM は、上記のアドレスを持っています。

DDRAM のアドレスを 0x6 に設定した後に、キャラクタデータの 0x41(='A')を送ると、

桁	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 行目						**A**									
2 行目															

の様に表示されます。続いて、キャラクタデータの 0x31(='1')を送ると、

桁	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 行目						**A**	**1**								
2 行目															

の様に表示されます。キャラクタデータを送った場合、DDRAM のアドレスは自動的にインクリメントされます（1 進みます）。

DDRAM のアドレス設定は、コマンドで行い、コマンドとしては 0b1xxx xxxx の xxxx のところに、上記 DDRAM アドレスを指定する事となります。

※キャラクタデータ送信時に DDRAM のアドレスを自動的にインクリメントするかどうかは、コマンドで決められます。本チュートリアルのサンプルプログラムでは、インクリメントする設定としています。

本チュートリアルのサンプルプログラムでは、LCD にコマンドを送る関数は、

lcd_cmd(コマンド)

キャラクタデータを送る関数は、

lcd_write_char(キャラクタ)

としているので、

・1 行目 6 桁目に移動
　lcd_cmd(0x86);　//0x1xxx xxxx, xxx xxxx のところに 0x06 を設定するので、0b1000 0110=0x86
・A を表示
　lcd_write_char(0x41);
・1 を表示（1 行目 7 桁目）
　lcd_write_char(0x31);
・2 行目 1 桁目に移動
　lcd_cmd(0xc0);　//0x1xxx xxxx, xxx xxxx のところに 0x40 を設定するので、0b1100 0000=0xc0
・abc を表示
　lcd_write_char(0x61);
　lcd_write_char(0x62);
　lcd_write_char(0x63);

桁	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 行目						**A**	**1**								
2 行目	**a**	**b**	**c**												

上記手順で、LCD の任意の場所に任意のデータを表示させる事が可能です。

表示した文字を消す方法ですが、

・オールクリア（コマンドで 0x01 を送る）
・DDRAM の内容を、0x20（ASCII でスペースのコード）で埋める

という方法があります。前者はコマンドを 1 回送信するだけですので、簡単ですが LCD 側の処理時間とし 1.52ms 掛かります。数文字を消したい場合は、0x20 を表示文字に設定するという方法もあります。

また LCD は、DDRAM の他に、CGRAM といって任意のドットパターンを定義できるメモリを持っています。

1 文字は、横 5 ドット、縦 8 ドットで構成されます。

ユーザ定義キャラクタとして、8 文字作成できます。

CGRAM の設定は、コマンドでアドレスを設定して、データを送る方法（DDRAM への書き込みと同じ）です。

CGRAM の使い方は、巻末の資料に記載しています。また、サンプルプログラムでは、DIP スイッチやプッシュスイッチのイメージのパターンを定義して使用していますので、具体的な使い方はサンプルプログラムを参照してください。

1.7. LCD の初期化手順

次に、多少面倒な LCD の初期化に関して記載します。

LCD は、データを 8bit で受け取る 8bit モードと 4bit で受け取る 4bit モードを持っています。本キットでは、4bit モードで LCD を使用するので、4bit モードにコマンドで設定を行う必要があります。

LCD の設定コマンドとしては、

0b0010 10xx （xx のところは任意なので、0x28）
※正確には上位 4bit 0x2?が 4bit モード、下位 4bit 0x?8 は、2 行モードの設定です

を送れば良いのですが、この初期化コマンド送った際、受け取る LCD 側でどのような状態となっているかがポイントです。

ここで、多少考えなければならない事があり、それはコマンドを送っている最中にマイコン側をリセットした場合などで、LCD 側が初期化前にどのモードに設定されているか判らないケースがある（も想定しておく必要がある）という事です。

LCD の状態としては、

(1)8bit モードで動作している（コマンド・キャラクタデータ待ちの状態）
(2)4bit モードで動作している（コマンド・キャラクタデータ待ちの状態…上位 4bit のデータを待っている状態）
(3)4bit モードで動作しており、上位 4bit コマンド・キャラクタデータ受信済みで、下位 4bit のデータを待っている状態

の 3 パターンが考えられます。(1)~(3)のどの状態でも初期化できるように考えなくてはなりません。

面倒なのは、(2)と(3)です。4bit モードでは、今 LCD が(2)の状態なのか(3)の状態なのかを判断する術がありません。

そこで、ちょっとトリッキーですが、最終的には 4bit モードに設定したいのですが、まずは 8bit モードに設定する事とします。8bit モードに設定した後に、4bit モードに設定すると、設定後は(2)の状態になっている事が保証されます。

8bit モードに設定するコマンドは、

0b0011 xxxx （xx のところは 4/8bit とは無関係な設定）

です。

ここでは、8bit モードに設定する上位 4bit のコマンド、0b0011 を 3 回連続で送る事とします。3 回連続で送る意図は下記の様なものです。

(1)初期化時 LCD が 8bit モードだった場合

	LCD が受け取るデータ	LCD が認識するコマンド
1 回目	0b0011XXXX	8bit モード切替コマンド
2 回目	0b0011XXXX	8bit モード切替コマンド
3 回目	0b0011XXXX	8bit モード切替コマンド

※XXXX は、マイコンと未接続の DB3-0 のレベルとなります（未接続の場合 LCD 内蔵のプルアップが有効となり実際は 1 となります）

(2)初期化時 LCD が 4bit モードだった場合　上位 4bit のデータ受信シーケンスからのスタート

	LCD が受け取るデータ	LCD が認識するコマンド
1 回目	0b0011（上位 4bit）	8bit の内半分の上位 4bit のデータの受信完了
2 回目	0b0011（下位 4bit）	8bit で 0b00110011 データの受信→8bit モード切替コマンド
3 回目	0b0011XXXX（8bit モード）	8bit モード切替コマンド

(3)初期化時 LCD が 4bit モードだった場合　下位 4bit のデータ受信シーケンスからのスタート

※LCD の動作モードが 4bit モードで、上位 4bit の受信が終わっており、下位 4bit のデータを待っている状態で、マイコンのリセットが掛かり、LCD 初期化を開始したケース

	LCD が受け取るデータ	LCD が認識するコマンド
1 回目	0b0011（下位 4bit）	受信済みの上位 4bit と組み合わせたデータ 0b????0011 を受信
2 回目	0b0011（上位 4bit）	8bit の内半分の上位 4bit のデータの受信完了
3 回目	0b0011（下位 4bit）	8bit で 0b00110011 データの受信→8bit モード切替コマンド

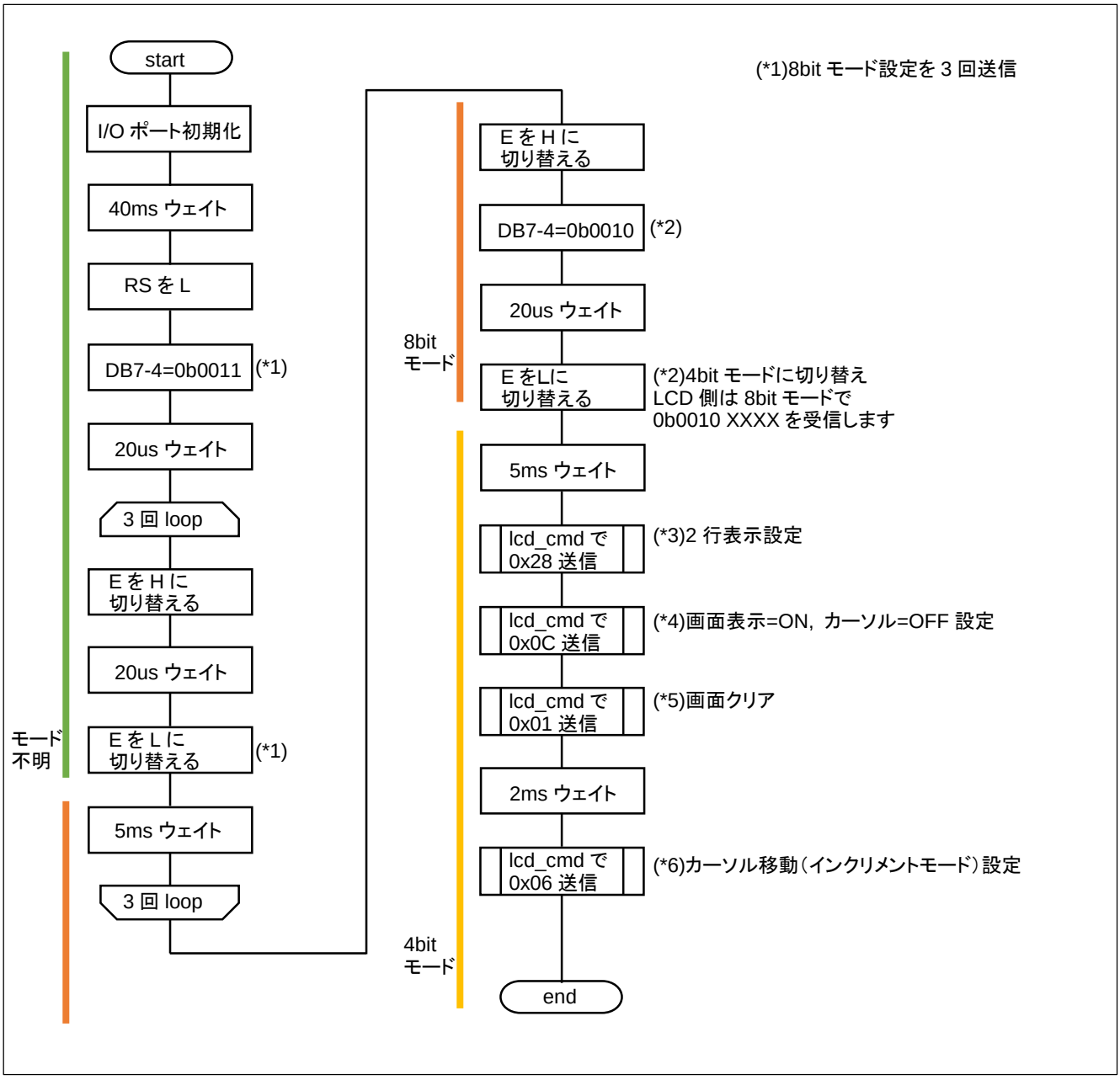
LCD の状態（8bit モード、4bit モード、4bit モードで上位 4bit のデータを受信済みの場合）に拘わらず、0b0011 のコマンドを 3 回送る事により、LCD は 8bit モードへ設定ができます。

一度 8bit モードに設定ができれば、その後 4bit モード設定コマンドを送れば、確実に 4bit モードに入れる事ができます。（一度 8bit モードに入れる事により、4bit の上位・下位のどちらから受信を開始するかという問題をクリアできます）

上記を踏まえた、LCD 初期化のフローチャートは以下になります。

1.8. LCD 初期化関数フローチャート

LCD 初期化関数 lcd_init()



1.9. サンプルプログラムで使用している関数（抜粋）

サンプルプログラム(lcd.c)内で定義している関数です。

lcd_init

概要:LCD 初期化関数
宣言:
　　void lcd_init(void)
説明:
・LCD の初期化を行います

引数:
　　なし
戻り値:
　　なし

lcd_cmd

概要:LCD コマンド送信関数
宣言:
　　void lcd_cmd(unsigned char c)
説明:
・LCD にコマンドを送信します

引数:
　　c: 送信コード(8bit)
戻り値:
　　なし

lcd_hs1, lcd_hs2

概要:1 行目にカーソルを移動させる関数, 2 行目にカーソルを移動させる関数
宣言:
　　void lcd_hs1(void), void lcd_hs2
説明:
・LCD にコマンドを送信します

引数:

なし

戻り値:

なし

lcd_clear

概要:LCD 画面クリア関数

宣言:

void lcd_clear(void)

説明:

・LCD の画面をクリアします

引数:

なし

戻り値:

なし

lcd_write_char

概要:LCD キャラクタ送信関数

宣言:

void lcd_write_char(unsigned char c)

説明:

・LCD にキャラクタ(1 文字)を送信します

引数:

c: キャラクタコード

戻り値:

なし

使用例:

lcd_write_char('A');

現在のカーソル位置に、A を表示させる。

lcd_write_hex, lcd_write_byte_int, lcd_write_short_int

概要:LCD 数値表示関数
宣言:
　　void lcd_write_hex(unsigned char c)
　　void lcd_write_byte_int(unsigned char num)
　　void lcd_write_short_int(unsigned short num)

説明:
　・LCD にキャラクタ(1 文字)を送信します

引数:
　　c, num: 表示する数値
戻り値:
　　なし
使用例:
lcd_write_hex(0x34);
　現在のカーソル位置に、34 を表示させる。
lcd_write_byte_int(120);
　現在のカーソル位置に、120 を表示させる。
lcd_write_short_int(12345);
　現在のカーソル位置に、12345 を表示させる。

※lcd_write_byte_int, lcd_write_short_int は、どちらも符号なしの表示関数です

lcd_write_str

概要:LCD 文字列表示関数
宣言:
　　void lcd_write_str(unsigned char *str)

説明:
　・LCD に文字列を送信します

引数:
　　*str: 表示する文字列
戻り値:
　　なし

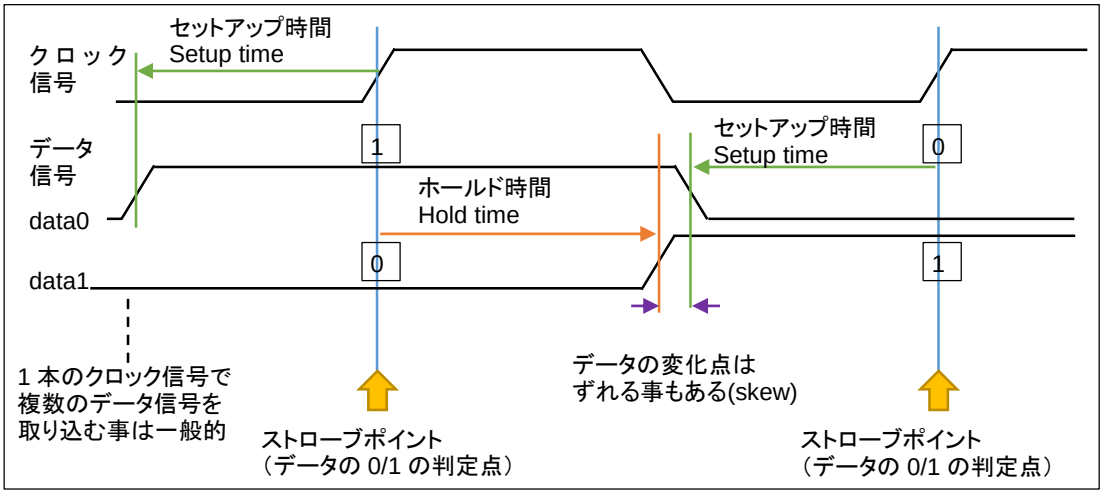
使用例:

lcd_write_str("LCD SAMPLE");

現在のカーソル位置に、LCD SAMPLE を表示させる。

コラム　セットアップ時間とホールド時間

電子回路では、セットアップ時間とホールド時間という概念が良く出てきます。
2 者間でデータ転送を行う際、守らなければならないタイミングの規定です。



この図では、ポジティブエッジトリガ（クロックの立ち上がりのタイミングでデータを取り込む）としています。（クロックの立下りでデータを取り込むネガティブエッジトリガや、両方のエッジでデータを取り込む DDR(Double Data Rate)というシステムもあります。）
※図 1-3 では、E がクロック、DB4-7 がデータですが、E と DB の関係は、E の立下りでデータを取り込むネガティブエッジトリガです（E に対し RS はポジティブエッジトリガで信号を取り込みます）

セットアップ時間は、データ信号が変化してから、データを取り込むポイント（上図のストローブポイント）までの時間です。この時間は、規定時間以上のタイミングを確保する必要があります。規定されているセットアップ時間より短い場合（データが変化してから直ぐにクロックが変化する場合）は、正しいデータを取り込めない可能性があります。

ホールド時間は、クロックが変化してから、一定時間データを変化させてはいけない時間です。データが変化して良いタイミングは、クロック変化後・ホールド時間経過後となる必要があります。

セットアップ時間やホールド時間は最低の時間が規定されていますので、その時間以上のタイミングが確保できるよう波形変化のタイミングを設計する必要があります。

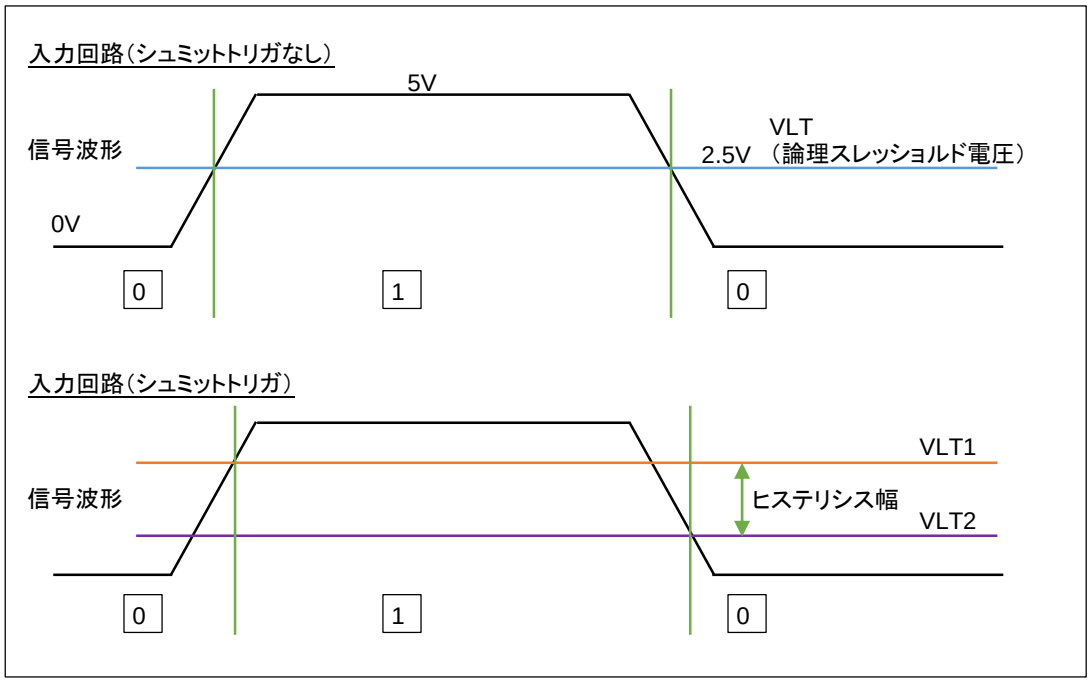
一般的には、1 本のクロック信号で複数のデータ信号を送るというケースが多いですが（本キットの LCD でも 4 本のパラレル信号となっています）、データの変化点に時間的なずれが生じる場合でも、全てのデータ信号が規定のセットアップ時間、ホールド時間を守る必要があります。一般的にはデータの変化点を、ストローブポイントの真ん中とすると、一番マージンが確保できます。（例えばホールド時間を長く取ると（=ストローブポイントを前の方に持ってくると）、セットアップ時間が短くなり、結局どちらかにしわ寄せが来ます）

[参考]通常は、デバイスの規定タイミングは　セットアップ時間>ホールド時間　のスペックとなりますので、ストローブポイントをデータ変化点のセンターではなく、多少後ろ側とした方がマージンが確保できるケースもあります。（セットアップ側に余裕を持たせる）

コラム　デジタル信号の L/H

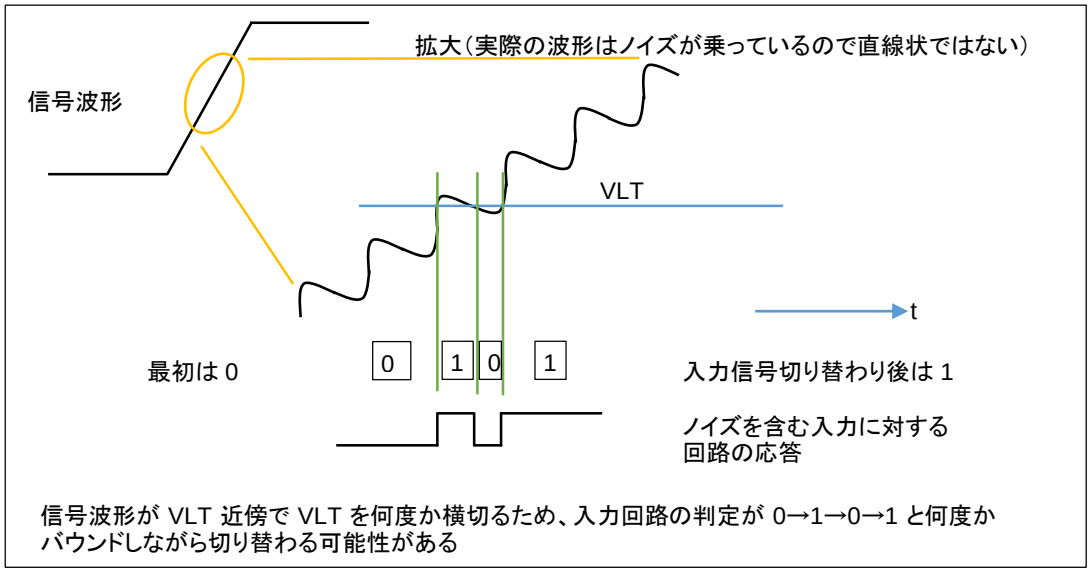
デジタル回路では、L/H のレベルで信号をやり取りしますが、信号を受け取る側の回路は電圧が 0V の時が L レベル。5V の時に、H レベルといった判断をします。

・回路の動作



デジタル回路では、L(0V), H（例えば 5V）の間に、スレッショルド（閾値）があり、閾値より低い場合デジタル的な 0。高い場合、デジタル的な 1 と判断されます。

シュミットトリガといい、閾値を 2 値設け、入力信号が L→H に変化する際は、高い閾値(VLT1)で 0/1 を判定し、入力信号が H→L に変化する際は低い閾値(VLT2)で 0/1 を判定する方式のものもあります。

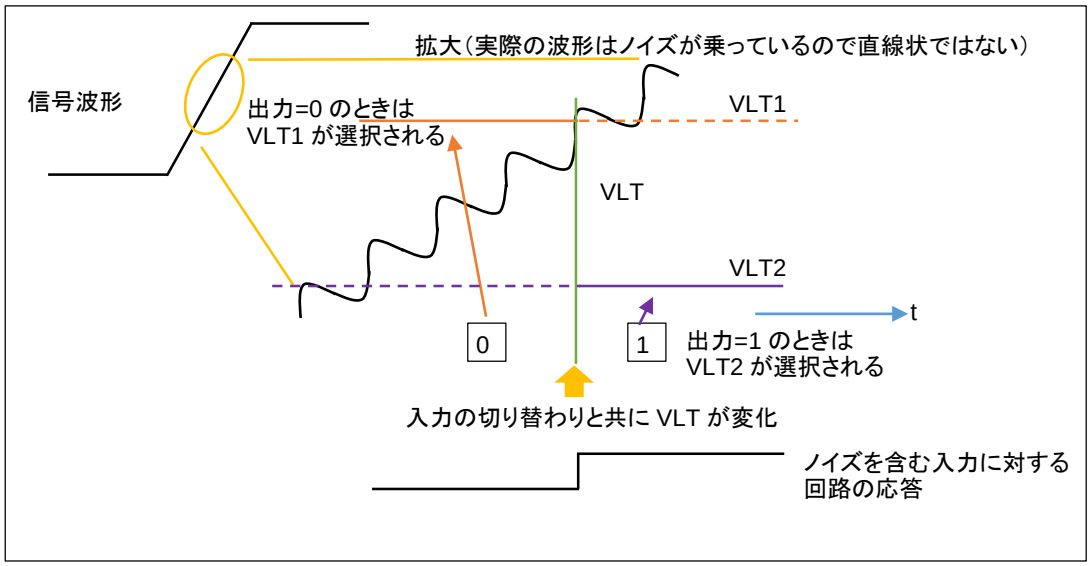


シュミットトリガなしの回路の場合、入力が 0→1 の切り替わりではなく、変化の途中で何度か 0 と 1 の間を行ったり来たりする可能性があります。

コラム　デジタル信号の L/H

入力のバウンドは、データの信号であれば問題ないのですが、クロック信号では問題となります。
※クロック信号と共に使用される、データ信号はあくまでクロックの切り替わりの際にデータがどちらかに確定していれば良いので切り替わりの過程で 0/1 の切り替わりが起こっても（望ましくはないかもしれませんが）問題はありません

シュミットトリガ系の回路を使用した場合は以下となります。



シュミットトリガの回路では、入力が 0→1 と判定された段階で、閾値が下がる（VLT2 に切り替わる）ので、1 に切り替わった後で信号波形が VLT1 を横切っても入力が 0 と判定される事はありません。

そのため、
・信号波形の傾きが鈍っている場合
・ノイズが多い場合
では、シュミットトリガ付きの入力回路が良く使用されます。

コラム　ViH/ViL と VOH/VOL（一見問題なさそうに見えるが…）

今回、マイコンの出力回路とLCD の入力回路を接続して信号のやり取りを行っています。そもそも、マイコンの出力と LCD の入力の電気的特性は整合しているのでしょうか。

データシートを見ると、以下の様になっています。

・LCD 側の入力のスペック

Item	Symbol	min	typ	max	Unit
Input Voltage	V_{IL}	0		0.6	V
	V_{IH}	2.2		VDD	V

これは、0~0.6V までは、入力=0 と見なします。2.2~VDD(=5V)までは、入力=1 と見なします。ということを意味しています。では、0.6~2.2V のときはどうなるのでしょうか？これは、不定（0 か 1 かは保証しないが、0 か 1 のどちらか）となります。

・マイコン側の出力スペック

項目	記号	min	max	単位	測定条件
出力 Low レベル	V_{OL}	-	1.2	V	IOL=20mA, 4.0<=VCC<=5.5V
出力 High レベル	V_{OH}	VCC-0.8	-	V	IOH=-4mA, 4.0<=VCC<=5.5V

ここで、VCC=VDD=5V 時、H レベルは

出力側：4.2V 以上（5-0.8）が保証れており　>　入力側：2.2V 以上であればよい

の関係が満たされています。L レベルは、

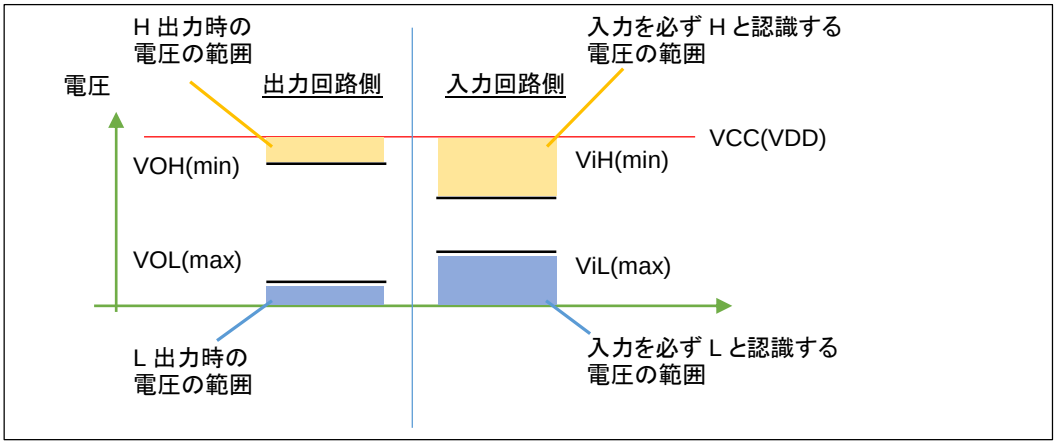
出力側：1.2V 以下を保証　<　入力側：0.6V 以下でなければならない
↑条件を満たしていない！

マイコン側は L レベルとして、1.2V 以下を保証しているので、L=0.7V でもスペックの範囲内です。それに対し、LCD 側は、L レベルとして 0.6V 以下を要求（0.7V の時は、H と見なすかもしれないよ）と言っているので、整合が取れていません。

但し、マイコン VOL の測定条件が IOL=20mA（マイコンが L 出力、外部から 20mA の電流を流し込んだ場合）の条件です。実際はマイコン L 出力時外部からの電流印加はほぼ 0 ですので、その場合の VOL はほぼ 0V（少なくとも、0.6V 以下）となります。そのため、実際は問題ないのですが、単純に VOL, ViL のスペックを見比べると NG となってしまいます。

コラム　ViH/ViL と VOH/VOL（一見問題なさそうに見えるが…）

ViH, ViL と VOH/VOL は、以下の関係となっている事が必要です。



※RA2L1 マイコンと、LCD(SC1602)の組み合わせは、感覚的には全く問題のない使い方ですが、データシートからは、問題のないことが読み取れないので多少厄介です

2. 付録

2.1. SmartRA 学習キット付属 LCD(SC1602)の仕様

＜LCD 資料＞

資料 1　液晶部について　特長

- 5×7 ドットマトリックス+カーソル、16 桁×2 の液晶表示
- 1/16 デューティ
- 192 種のキャラクタジェネレータ ROM
  文字フォント：5×7 ドットマトリクス
- プログラム書込み可能な 8 種のキャラクタジェネレータ RAM
  文字フォント：5×7 ドットマトリクス
- 80×8 ビットの表示データ RAM（最大 80 文字）
- 4ビット及び8ビットの MPU とのインタフェース可能
- 表示データ RAM、キャラクタジェネレータ RAM ともに MPU からの読み出しが可能
- 豊富なインストラクション機能
  表示クリア　他　資料 3　インストラクションについて参照
- 発振回路内蔵
- 5V 単一電源　・　動作温度範囲　0～50℃
- 電源投入時自動リセット回路内蔵
- CMOS プロセス使用

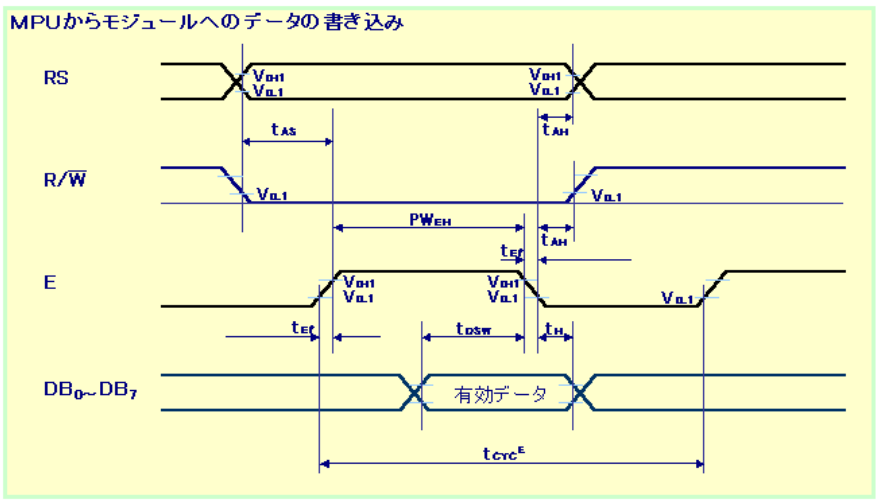
資料 2　タイミング特性について

＜タイミング＞

項目	記号	MIN	MAX
イネーブルサイクル時間	tCYCE	1200	-
イネーブルパルス幅 "High"レベル	PWEH	140	-
イネーブル立上がり・立下り時間	tEr・tEf	-	25
セットアップ時間 RS、R/*W→E	tAS	0	-
アドレスホールド時間	tAH	10	-
データセットアップ時間	tDSW	40	-
データホールド時間	tH	10	-

■書込み動作　単位：ns

VDD=5.0V±10% VSS=0V Ta=0～50



資料 3　インストラクションについて

＜機能コード一覧＞

インストラクション	コード										機能	実行時間(typ)
	RS	R/*W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
表示クリア	0	0	0	0	0	0	0	0	0	1	全表示クリア後、カーソルをホーム位置（0番地）へ戻す	1.52ms
カーソルホーム	0	0	0	0	0	0	0	0	1	*	カーソルをホーム位置へ戻し、シフトしていた表示も元へ戻る（DDRAM の内容は変化無し）	1.52ms
エントリーモード	0	0	0	0	0	0	0	1	I/D	S	カーソルの進む方向、表示をシフトするかどうかの設定（データ書込み及びデータ読み出し時に上記動作が行われます）	37μs
表示 ON/OFF コントロール	0	0	0	0	0	0	1	D	C	B	全表示の ON/OFF[D]、カーソル ON/OFF[C]、カーソル位置の文字のブリンク[B]をセット	37μs
カーソル/表示シフト	0	0	0	0	0	1	S/C	R/L	*	*	DD RAM の内容を変えずカーソルの移動、表示シフト	37μs
ファンクションセット	0	0	0	0	1	DL	N	F	*	*	インタフェースデータ長[DL]、表示行数[N]、文字フォント[F]を設定	37μs
CG RAM アドレスセット	0	0	0	1	ACG						CG RAM のアドレスセット(以後送受するデータは CG RAM データ)	37μs
DD RAM アドレスセット	0	0	1	ADD							DD RAM のアドレスセット(以後送受するデータは DD RAM データ)	37μs
BF/アドレス読出し	0	1	BF	AC							ﾓｼﾞｭｰﾙが内部動作中であることを示す BF 及び AC の内容を読出し(CG RAM/DD RAM 双方可)	37μs
CG RAM/DD RAM データ書込み	1	0	書込みデータ								CG RAM または DD RAM にデータを書込む	37μs tADO=5.6μs
CG RAM/DD RAM データ読出し	1	1	読出しデータ								CG RAM または DD RAM にデータを読出す	37μs tADO=5.6μs

*	:	無効のビット
ACG	:	CGRAM のアドレス
ADD	:	DDRAM のアドレス
AC	:	アドレスカウンタ

	=1	=0
R/L	右シフト	左シフト
S	表示をシフトさせる	表示をシフトしない
N	2 行表示	1 行表示
F	5×10 ドットマトリックス	5×7 ドットマトリックス
BF	内部動作中	インストラクション受付可
S/C	表示のシフト	カーソル移動

	=1	=0
I/D	インクリメント	デクリメント
DL	8ビット	4ビット
D	表示ON	表示OFF
C	カーソルON	カーソルOFF
B	ブリンクON	ブリンクOFF

■クロック発振周波数（fOSK）が変化すると実行時間も変化します
　例　fOSK＝190kHz の場合　37μs×270/190＝53μs
■tADO 時間はクロック発振周波数（fOSK）によって変化します
tADO＝1.5/（fOSK）（s）

資料 4　文字コードと文字パターンについて

文字コードと文字パターンは下記例の通りの関係となっております　（対応一覧は次の資料 5 文字コード一覧をご覧下さい）

＜CG RAM アドレスと文字コード・文字パターン＞

- CGRAM データは"1"が表示上の選択、"0"が非選択に対応します
- 文字コードビット 0-2 と CGRAM アドレスビット 3-5 が対応します（3ビット8種）
- CGRAMアドレスビット 0-2 が文字パターンの行位置を指定します
- 文字パターンの8行目はカーソル位置で、カーソルとCGRAMデータの論理和をとって表示されますので、カーソル表示を行う際は8行目のCGRAMデータを0にして下さい
- 8行目のデータを1にするとカーソルの有無に関係なく1ビットが点灯します
- 文字パターンの列位置はCGRAMデータビット 0-4 に対応し、ビット4が左端になります
- CGRAMデータビット 5-7 は表示されませんが、メモリは存在しているので、一般のデータRAMとして使用できます
- CGRAM の文字パターンを読み出すときは文字コードの 4-7 ビットは全て"0"を選択します
- どのパターンを読み出すかは 0-2 のビットで決定しますが、ビット3は無効なので"00H"と"08H"では同じ文字が選択されます

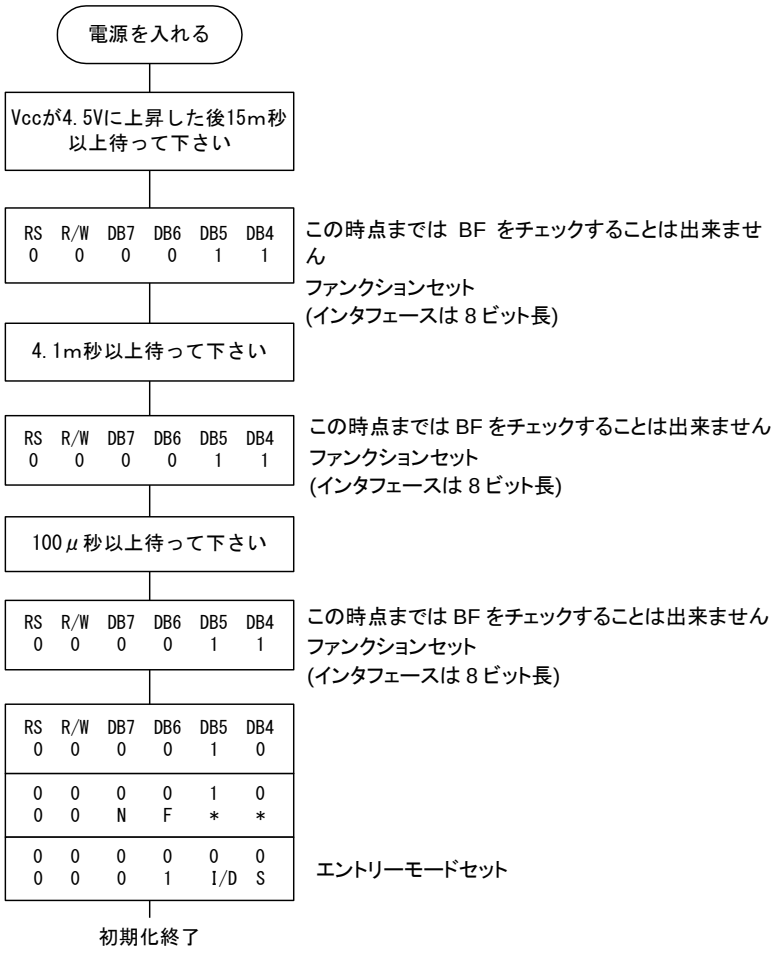
文字コード(DDRAMデータ) 7 6 5 4 3 2 1 0								CG RAMアドレス 5 4 3			文字パターン(CGRAMデータ) 7 6 5						
上位ビット				下位ビット				上位ビット			上位ビット			下位ビット			
										0 0 0	* * *	1 1 1 1 0					文字パターン例「R」
										0 0 1	* * *	1 0 0 0 1					
										0 1 0	* * *	1 0 0 0 1					
0 0 0 0	・	0 0 0						0 0 0		0 1 1	* * *	1 1 1 1 0					
										1 0 0	* * *	1 0 1 0 0					
										1 0 1	* * *	1 0 0 1 0					
										1 1 0	* * *	1 0 0 0 1					
										1 1 1	* * *	0 0 0 0 0					←カーソル位置
										0 0 0	* * *	1 0 0 0 1					文字パターン例「¥」
										0 0 1	* * *	0 1 0 1 0					
										0 1 0	* * *	1 1 1 1 1					
0 0 0 0	・	0 0 1						0 0 1		0 1 1	* * *	0 0 1 0 0					
										1 0 0	* * *	1 1 1 1 1					
										1 0 1	* * *	0 0 1 0 0					
										1 1 0	* * *	0 0 1 0 0					
										1 1 1	* * *	0 0 0 0 0					←カーソル位置
										0 0 0	* * *						
0 0 0 0	・	1 1 1						1 1 1		0 1 1	* * *						
										1 0 0	* * *						
										1 0 1	* * *						
										1 1 0	* * *						
										1 1 1	* * *						←カーソル位置

資料 5　文字コード･文字パターン対応一覧

＜文字コードと文字パターン対応表 ＞

上位4ビット 下位4ビット	xxxx	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx	0000	CG RAM (1)		0	@	P	`	p		−	タ	ミ	α	p
xxxx	0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx	0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	ß	θ
xxxx	0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx	0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
xxxx	0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
xxxx	0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx	0111	(8)		7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
xxxx	1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
xxxx	1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	$^{-1}$	y
xxxx	1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx	1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	^{x}	万
xxxx	1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
xxxx	1101	(6)	−	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx	1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
xxxx	1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	■

資料 6　LCD 初期化フロー



取扱説明書改定記録

バージョン	発行日	ページ	改定内容
REV.1.0.0.0	2021.5.10	—	初版発行

お問合せ窓口

最新情報については弊社ホームページをご活用ください。

ご不明点は弊社サポート窓口までお問合せください。

株式会社 北斗電子

〒060-0042 札幌市中央区大通西 16 丁目 3 番地 7

TEL 011-640-8800　FAX 011-640-8801

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

URL:http://www.hokutodenshi.co.jp

ルネサス エレクトロニクス RA マイコン搭載
HSB シリーズマイコンボード 評価キット

SmartRA 学習キット
チュートリアル 3

株式会社 北斗電子

 2021 年 5 月 10 日改訂 REV.1.0.0.0 (210510)