

電子ペーパー表示書き換えモジュール

2.0インチ電子ペーパー付き 電子ペーパー制御基板

Y-Con P020 With2.0inchEPD Ver 1.0

かんたんマニュアル

「Y-Con P020（ワイコンピー020） With2.0inchEPD」は、2 インチ電子ペーパーと制御基板をセットにしたモジュールです。
ホスト（マイコンやPC等）から電源供給と画像データ転送を行うことで電子ペーパーに簡単に描画できます。
※画像データは、**2 0 0 × 9 6** 画素の白黒（二値）ビットマップファイルです。

はじめに

このたびは「2.0インチ電子ペーパー付き電子ペーパー制御基板 Y-Con P020 With2.0inchEPD」（以降「本機」と記載）をお買い上げいただき、誠にありがとうございます。
ご使用前に本書を必ずお読みいただき、正しくお使いください。お読みになった後は、いつでも見られるようにお手元に大切に保管してください。

- ・本書の内容の一部、または全部を無断転載することは禁止されています。
- ・本書の内容に関して、将来予告なしに変更することがあります。
- ・本書の内容については万全を期してはありますが、万一ご不審な点や記載漏れなどお気づきの点がありましたら、ご連絡くださいますようお願いいたします。

お問い合わせ先

株式会社ワイズ・ラブ 営業部
メールでのお問い合わせ info@yslab.co.jp
ホームページからの問い合わせ http://www.yslab.co.jp/

本機のコセプト

部品として使用できることを主眼に設計しました！
薄さ／省電力／接続のしやすさ



＜開いた状態＞



＜閉じた状態＞

開いたままでも、基板に貼り合わせて閉じた状態でも使用できます。

動作モードの種類

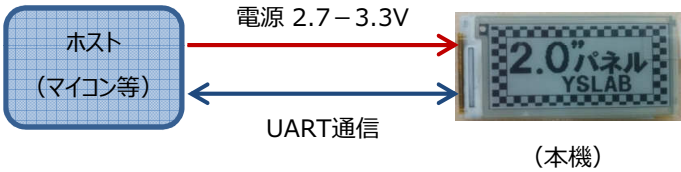
本機には 3 つの動作モードがあります。

動作モード	動 作
デモ動作モード	デモ用の 6 画面を逐次描画します
通常動作モード	ホストから画像データが転送されると描画します
コマンド動作モード	ホストと対話方式でコマンドが実行できます

本書では「デモ動作」と「通常動作」の接続／操作方法を記載します。
「コマンド動作モード」に関してはホームページより詳細マニュアルをダウンロードしていただきますようお願いいたします。

http://www.yslab.co.jp/epd_ycon_20.html

ホストとの接続方法

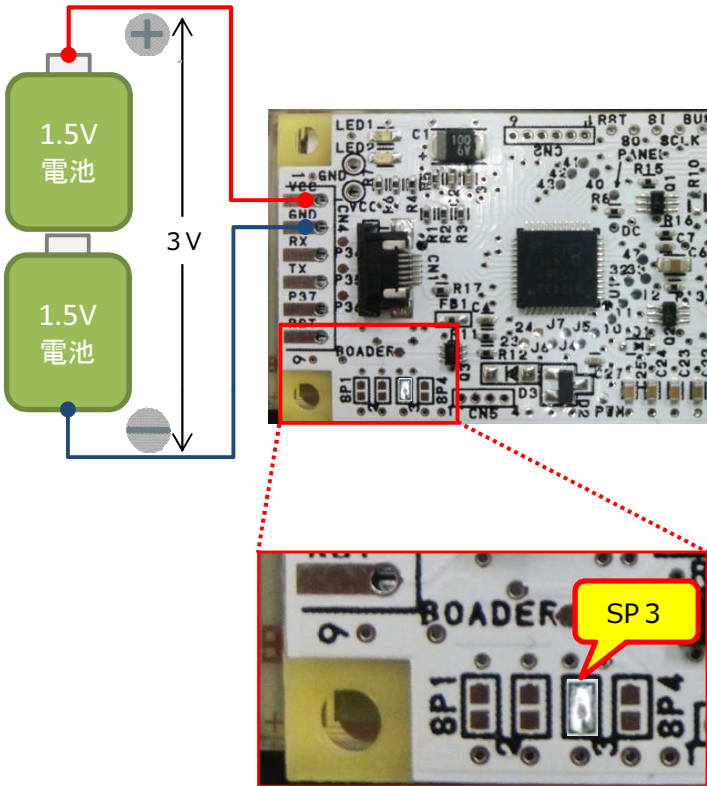


※ホストはお客様側でご用意ください

デモ動作接続方法

下図の様に SP 3 というジャンパーをハンダで短絡（ショート）させて、電池ボックス（単 4 × 2）の（+）側を 1 番端子に、（-）側を 2 番端子に接続して電源供給を開始するとデモが実行されます。

デモ動作時の接続方法



【ご注意！】

デモ動作時は画像データ転送は行えません。
SP 3 をオープンにして電源を再投入すれば通常動作に戻せます。

主な仕様

項 目	仕 様	備 考
外形寸法	57mm (H) × 29mm (V) × 4.1mm (T)	基板に貼合せ状態
重 量	7グラム	2.0" EPD+基板
有効表示エリア	46mm (H) × 22mm (V)	2.0" EPD
電源電圧	2.7 ～ 3.3V	
通信方式	調歩同期方式 115200bps、8N1	19200bps 設定可
画像形式	200 × 96 モノクロビットマップファイル	2750バイト
動作温度	0 ～ 50℃	
保存温度	-20 ～ 60℃	
描画時間	50℃時：約 2.3 秒 40℃時：約 2.3 秒 30℃時：約 2.8 秒 20℃時：約 2.8 秒 15℃時：約 4.8 秒 10℃時：約 6.7 秒 5℃時：約 8.6 秒 0℃時：約16.3 秒	通信時間を除く

本機の特長

- ・電源が落ちても表示を保持できます。
- ・低消費電力です。
- ・視野角が大変広いです。
- ・基板サイズを電子ペーパーと同一サイズにしました。
- ・強度を増すために電子ペーパーを基板に貼り付けて使用できます。
- ・部品として組込み易いように薄型としました。
- ・制御は電源とUART通信だけで行えます。

ハードウェア仕様

コネクタ（CN #）

コネクタ	内 容	部品名
CN3	電子ペーパーの接続コネクタ	62684-401100ALF
CN4	ホストとの接続用 端子は 2.54mm ピッチ	

ホスト接続用コネクタ

端子番号	信号名	入出力	内 容
1	VCC	－	電源入力（2.7 ～ 3.3 V）
2	GND	－	電源グランド
3	RX	入力	UART受信
4	TX	出力	UART送信
5	P37	出力	画像転送Ready信号 ※Highで転送を開始可能
6	RESET	入力	Low で MCU リセット

ジャンパー（SP #）

名称	機 能	初期値
SP1	通信速度設定（ショート時：19200bps）	115200bps
SP2	LED動作設定（ショート時：無効）	LED動作有効
SP3	デモモード設定（ショート時：デモ動作）	通常動作
SP4	P37 出力設定（ショート時：無効）	P37 出力有効

※ショートピンの参照は、パワーオン/リセット時に行われます

LED

名称	機 能	備 考
LED1	LED動作有効時：電子ペーパー描画中に点灯	赤色
LED2	LED動作有効時：ホストからの受信待ち時に点灯	青色

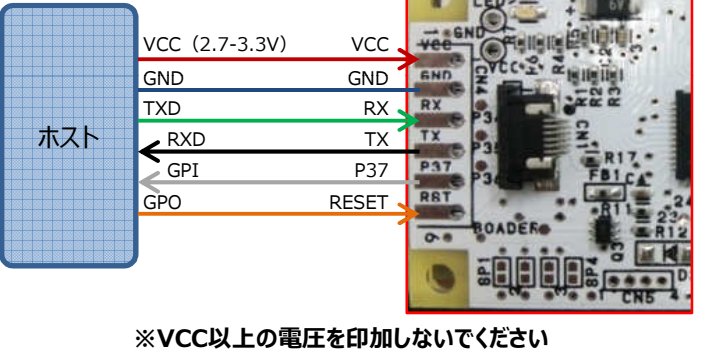
※LED動作を無効にすると点灯しくくなります
（消費電力低減になります）

通信条件

項 目	内 容	備 考
通信方式	調歩同期方式	
通信速度	115200bps または 19200bps	SP1で設定
キャラクタ長	8 ビット	
ストップビット	1 ビット	
フロー制御	なし	

通常接続方法

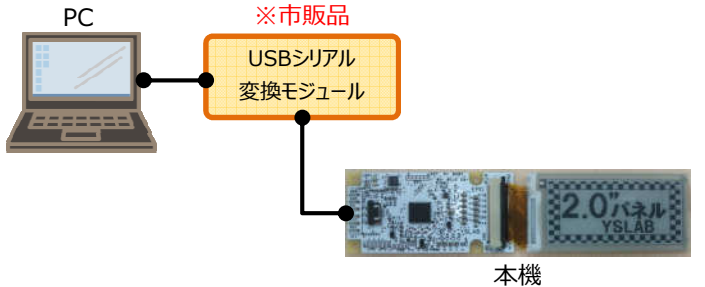
- ・TX端子への接続は、Y-Conからの応答が不要な場合は未接続でも構いません
- ・P37端子は、画像転送が可能な状態の時に High が出力されます
- ・利用しない場合は未接続でも構いません
- ・RESET端子は Low を入力すると Y-Conがリセットされます
- ・ホストからリセットする必要がある場合は未接続でも構いません



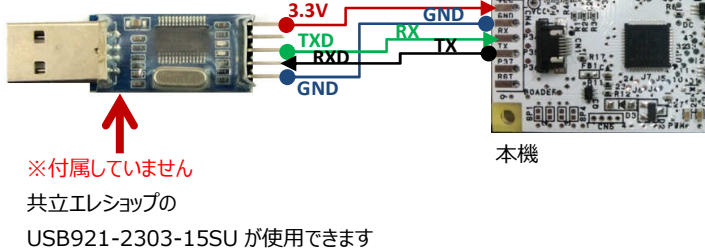
※VCC以上の電圧を印加しないでください

パソコンとの接続方法

本機をパソコンと接続するためには、USB－シリアル変換モジュールが必要です。
またパソコンから画像データを転送するためにTera Tarmなどのターミナルソフトが必要となります。



小型USB-シリアル変換モジュール
USB921-2303-15SU



※付属していません
共立エレクトロニクスの
USB921-2303-15SU が使用できます

マイコンとの接続方法例

詳細な接続方法例は当社ホームページでご確認ください。

画像データ形式

本モジュールでホストとの通信で用いる画像データは、白黒ビットマップファイルの構造（**2 0 0 × 9 6 × 1 b p p**）です。

BMP形式

ファイルヘッダ BITMAPFILEHEADER構造体	BMPファイルのサイズは2750バイトです。 ＜内訳＞ ヘッダー部＝62バイト ピクセルデータ＝2688バイト ※（25＋3）×96ライン ※※＋3は4バイトの倍数にするために「00 00 00」が挿入されるため
情報ヘッダ BITMAPINFOHEADER構造体	
カラーデータ 複数個のRGBQUAD構造体	
ピクセルデータ	

※RGBQUADの個数はbiBitCount＝1（2 値）なので 2 個となります。
適応するBMP形式の詳細は裏面の表でご確認ください。

画像データ転送方法

本書裏面をご覧ください。

さいごに

アンケートご協力のお願い

Y-Con P020with2.0inchEPD にお客様の声を反映するためアンケートを実施させていただきます。つきましては弊社ホームページよりぜひともアンケートにご協力いただきたく、よろしくお願い申し上げます。

アンケート用ページ

<http://goo.gl/forms/I9GFuQt2QI>

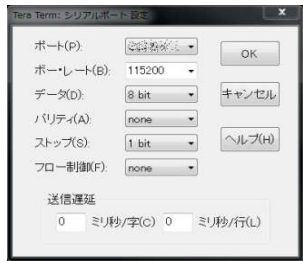


画像データ転送方法

パソコンとの接続時

<Tera Term での画像データ転送例>

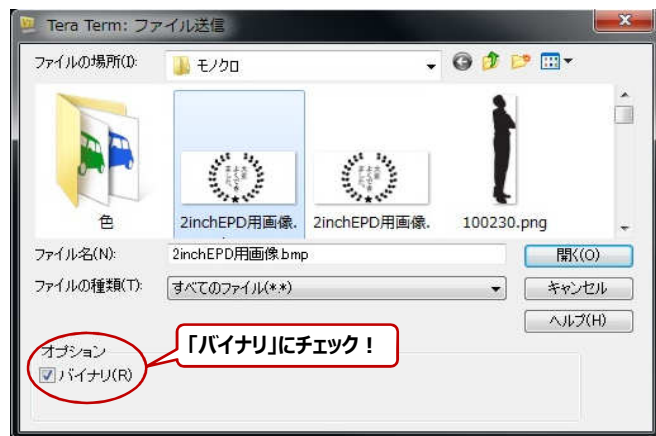
1. 「設定」メニューの「シリアルポート」を選択し、
「Tera Term:シリアルポート設定」の項目をY-Conの通信設定と合わせます。



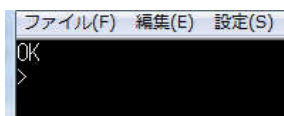
2. 「ファイル」メニューの「ファイル送信」を選択します。



3. 「Tera Term: ファイル送信」ダイアログで送信する
Bmpファイルを選択し、「開く」ボタンを押します。
この時**オプション**の**「バイナリ」**にチェックしてください。



4. 「OK」と表示され、プロンプト (>) が表示されたら完了です。



表示された画像の一例



* Memo *

適応するBMP形式の詳細

offset		00	01	02	03	04	05		
0000:	BITMAPFILEHEADER	42	4D					※fType	"BM"
0002:		BE	0A	00	00			※fSize	2 7 5 0バイト固定
0006:		00	00					※fReserved1	0固定
0008:		00	00					※fReserved2	0固定
000A:		3E	00	00	00			※fOffBits	6 2バイト目固定
000E:	BITMAPINFOHEADER	28	00	00	00			※iSize	4 0バイト固定
0012:		C8	00	00	00			※iWidth	2 0 0画素固定
0016:		60	00	00	00			※iHeight	9 6ライン固定
001A:		01	00					※iPlanes	1固定
001C:		01	00					※iBitCount	1 (2値) 固定
001E:		00	00	00	00			※iCompression	0 (BI_RGB: 無圧縮) 固定
0022:		80	0A	00	00			※iSizeImage	2 6 8 8バイト固定
0026:		00	00	00	00			※iXPelsPerMeter	0固定
002A:		00	00	00	00			※iYPelsPerMeter	0固定
002E:		00	00	00	00			※iClrUsed	0固定
0032:	RGBQUAD	00	00	00	00			※iClrImportant	0固定
0036:		00						※gbBlue	0固定 or 2 5 5固定
0037:		00						※gbGreen	0固定 or 2 5 5固定
0038:		00						※gbRed	0固定 or 2 5 5固定
0039:		00						※gbReserved	0固定
003A:	RGBQUAD	FF						※gbBlue	2 5 5固定 or 0固定
003B:		FF						※gbGreen	2 5 5固定 or 0固定
003C:		FF						※gbRed	2 5 5固定 or 0固定
003D:		00						※gbReserved	0固定
003E:	ピクセルデータ	9 6ライン目 (2 5バイト+3バイト)				00	00	00	
005A:		9 5ライン目 (2 5バイト+3バイト)				00	00	00	
0076:		9 4ライン目 (2 5バイト+3バイト)				00	00	00	
		...							
0A6A:		3ライン目 (2 5バイト+3バイト)				00	00	00	
0A86:		2ライン目 (2 5バイト+3バイト)				00	00	00	
0AA2:		1ライン目 (2 5バイト+3バイト)				00	00	00	
0ABE:									

初期設定されているデモ画像一覧

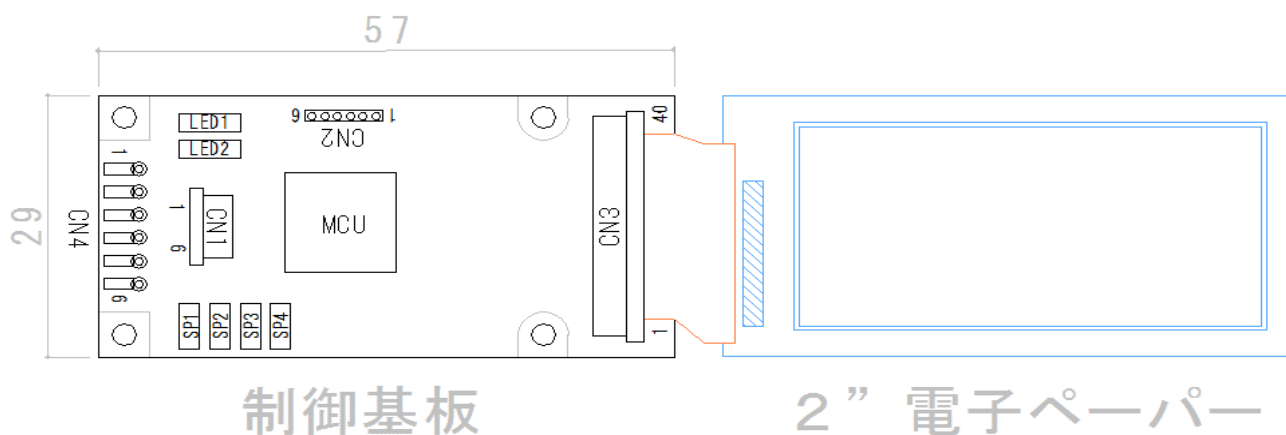
ページ番号	用途	初期画像
ページ0	待機画面	
ページ1	デモ用画面－1	
ページ2	デモ用画面－2	
ページ3	デモ用画面－3	
ページ4	デモ用画面－4	
ページ5	デモ用画面－5	
ページ6	デモ用画面－6	

Y-Con P020 With2.0inchEPD 操作説明書

Rev 1.0

本製品は、2インチ電子ペーパーと制御基板をセットにしたモジュールです。
 ホスト(マイコンやPC等)から電源供給と画像データ転送を行うことで電子ペーパーに簡単に描画できます。

※画像データは、200×96画素のモノクロ(二値)ビットマップファイルです。



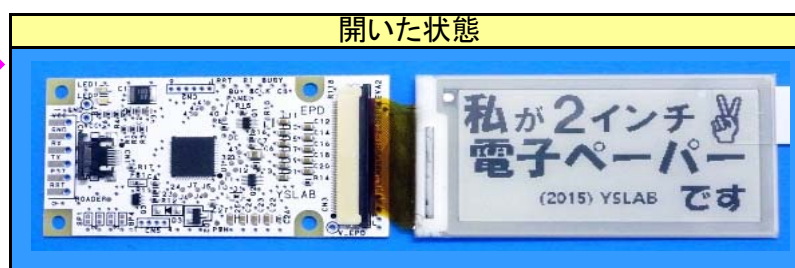
部品として使用できることを主眼に設計しました！

- ・薄さ
- ・省電力
- ・接続

開くも

閉じるも

自由です



接続は

ホスト
(マイコン等)

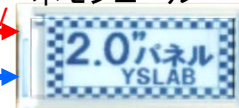
ホストは
お客様の方で
ご用意ください。

電源

2.7-3.3V

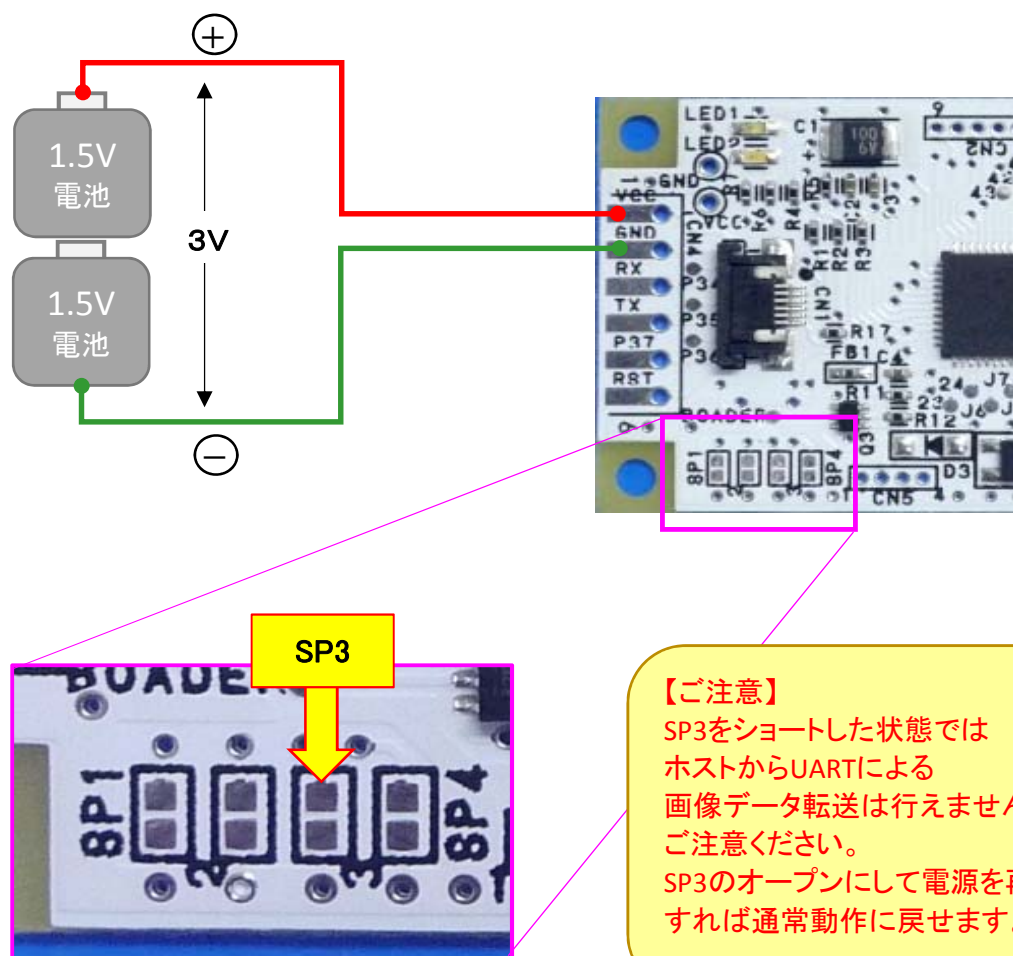
UART通信

本モジュール



■とりあえず動かしてみる……デモモード

写真のように SP3というジャンパーをハンダで短絡(ショート)させて
電池ボックス(単4×2)の(+)側を1番端子に、(-)側を2番端子に接続し
電力供給を開始するとデモが実行されます。



■主な仕様

項番	項目	仕様	備考
(1)	外形寸法	57mm(H) × 29mm(V) × 4.1mm(T)	基板に貼合せ状態
(2)	重量	7グラム	2.0"EPD+基板
(3)	有効表示エリア	46mm(H) × 22mm(V)	2.0"EPD
(4)	電源電圧	2.7～3.3V	
(5)	消費電流	約10mA(突入電流を除く)、待機時約0.3mA	LED出力なし時
(6)	通信方式	調歩同期方式 115200bps, 8N1	19200bps設定可
(7)	画像形式	200x96 モノクロビットマップファイル	2750バイト
(8)	動作温度	0～50℃	
(9)	保存温度	－20～60℃	
(10)	描画時間	50℃時:約2.3秒 40℃時:約2.3秒 30℃時:約2.8秒 20℃時:約2.8秒 15℃時:約4.8秒 10℃時:約6.7秒 5℃時:約8.6秒 0℃時:約16.3秒	通信時間を除く

■特長

項番	内容
(1)	電源が落ちても表示を保持できます。
(2)	低消費電力です。
(3)	視野角が大変広いです。
(4)	基板サイズを電子ペーパーと同一サイズにしました。
(5)	強度を増すために電子ペーパーを基板に貼り付けて使用できます。
(6)	部品として組込み易いように薄型としました。
(7)	制御は電源とUART通信だけで行えます。
(8)	
(9)	
(10)	

■ハードウェア

■コネクタ(CN#)

コネクタ	内容	部品名
CN3	電子ペーパーの接続コネクタです。	62684-401100ALF
CN4	ホストとの接続用。端子は2.54mmピッチです。	

■CN1 & CN4: ホスト接続用コネクタ

端子番号	信号名	入出力	内容
1	VCC	—	電源入力(2.7~3.3V)
2	GND	—	電源グランド
3	RX	入力	UART受信
4	TX	出力	UART送信
5	P37	出力	画像転送Ready信号 ※Highであれば転送を開始できます。
6	RESET	入力	LowでMCUリセット

■ジャンパー(SP#)

名称	機能	初期値
SP1	通信速度設定(ショート時:19200bps)	115200bps
SP2	LED動作設定(ショート時:無効)	LED動作有効
SP3	デモモード設定(ショート時:デモ動作)	通常動作
SP4	P37出力設定(ショート時:無効)	P37出力有効

ショートピンの参照は、パワーオン・リセット時に行われます。

■LED

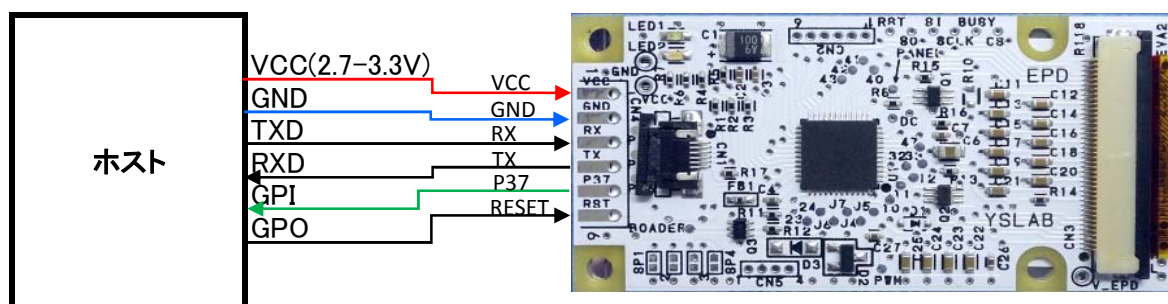
名称	機能	備考
LED1	LED動作有効時:電子ペーパー描画中に点灯	赤色
LED2	LED動作有効時:ホストからの受信待ち時に点灯	青色

■通信条件

項目	内容	備考
通信方式	調歩同期方式	
通信速度	115200bps または 19200bps	SP1で設定
キャラクタ長	8ビット	
ストップビット	1ビット	
フロー制御	なし	

■接続方法

CN4コネクタに接続する場合



【備考】

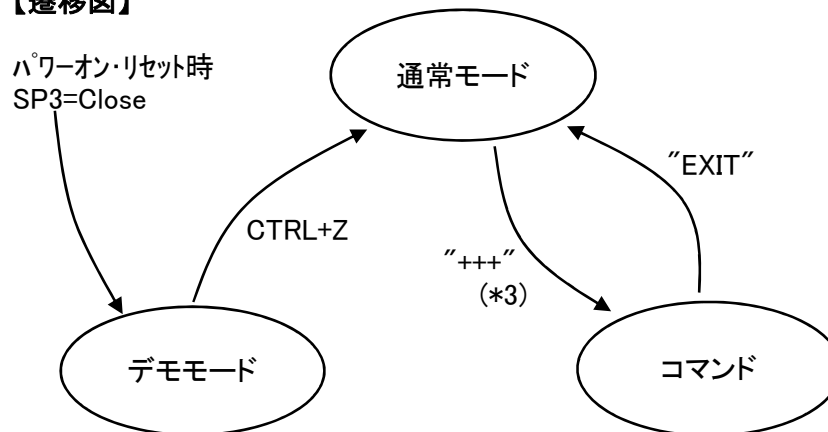
- ◆1. TX端子への接続は、Y-Conからの応答が不要な場合は未接続でも構いません。
- ◆2. P37端子は、画像転送が可能な状態のときにHighが出力されます。
利用しない場合は未接続でも構いません。（SP4で出力を禁止できます）
- ◆3. RESET端子は、Lowを入力するとY-Conがリセットされます。
ホストからリセットする必要がある場合は未接続でも構いません。

■動作モード

このモジュールには、3つの動作モードがあります。

項番	動作モード	動作
(1)	通常モード	ホストから画像データが転送されると描画します。
(2)	コマンドモード	ホストと対話形式でコマンドが実行できます。
(3)	デモモード	デモ用です。

【遷移図】



(1)通常モード …… プロンプトは '>(3Eh)

ホストからビットマップファイル(*1)を送信することで描画が行えるモードです。
 パワーオン・リセット後、待機画面を描画します。
 待機画面の描画完了後にプロンプトを返します。
 ホストはプロンプトが返された時に画像データ(*1)を送ることができます。
 転送が完了すると、描画を開始します。
 描画が終了すると完了通知(*2)を返し、プロンプトに戻ります。
 ※プロンプトが返された時に'+++'(*3)を入力すると「コマンドモード」に移行できます。

(2)コマンドモード …… プロンプトは '!(21h)

プロンプト('!)が表示されたらコマンド入力が行えます。
 '?'コマンドで、コマンド一覧が表示されます。
 EXITコマンドで、「通常モード」に移行できます。

(3)デモモード

モジュールは、デモ用に6画面のデータを内蔵しています。
 デモモードで起動した場合は、ホストに"DEMO MODE"と返した後、
 デモ画面を逐次描画します。
 CTRL+Z(1Ah)を受信すると「通常モード」に移行します。

*1: 200x96のモノクロビットマップファイル(2750byte)

*2: 完了通知は、正常時は"OK"、異常時は"NG"

*3: '+' (wait) '+' (wait) '+' (CR)と入力する。waitは 100-500ms の範囲内で。

■コマンドモードで使用できるコマンド

コマンドモード時は、ホストと対話形式で任意のコマンドを実行できます。

使用できるコマンドは、

コマンド文字列 + (CR)

※(CR)は、0Dh

で実行できます。

※コマンド文字列は、大文字／小文字の区別はありません。

コマンドモードのプロンプトは、'!' (21h)です。

プロンプトがTX出力されると、コマンド受付ができる状態です。

コマンドの実行が完了すると、プロンプトがTX出力されます。

コマンド入力を間違えた場合は、BS (バックスペース) で1文字削除できます。

入力されたコマンド文字列が適切でない場合は、"??" がTX出力されます。

LED出力が許可されている場合は、LED2 (青) 点灯中はコマンド入力ができる状態を表します。

■コマンド一覧

項番	コマンド文字列	機能	備考
(1)	?	コマンド一覧の表示	
(2)	W	白画面の描画	
(3)	B	黒画面の描画	
(4)	N	現在の画面の白黒反転	
(5)	D	指定ページの描画	
(6)	R	指定ページへの画像転送と描画	
(7)	I	システム情報のTX出力	
(8)	DEMO	デモ実行	
(9)	INTERVAL	デモのページ間のウェイト時間表示・設定	
(10)	LASTWAIT	デモの最終ページのウェイト時間表示・設定	
(11)	STANDBY	パワーオン・リセット時の待機画面描画の許可／禁止	
(12)	LED	LED出力の許可／禁止	
(13)	P37	P37端子出力の許可／禁止	
(14)	EXIT	通常モードに遷移	
(15)	RESET	MCUのリセット	
(16)	YSLAB	弊社の案内をTX出力	

■(1)コマンド: ?

コマンド一覧をTX出力します。

書式: ?(CR)

!?

[Usage]

? : Command list(this menu)	?
W: White screen drawing	W
B: Black screen drawing	B
N: Nega screen drawing	N
D: Draw Image Page	D [page]
R: Regist Image Page	R [page]
I: System Information	I
DEMO: Run Demo	DEMO
INTERVAL: Demo Interval	INTERVAL [ms]
LASTWAIT: Demo Last page wait	LASTWAIT [ms]
STANDBY: Enable Standby screen	STANDBY [0/1]
LED: Enable LED Contorol	LED [0/1]
P37: Enable Ready Signal	P37 [0/1]
EXIT: Exit Command mode	EXIT
RESET: Reset MCU	RESET
YSLAB: Yslab Information	YSLAB

■(2)コマンド: W

白画面を描画します。

書式: W(CR)

コマンド実行時

!W

コマンド終了時

OK
!

画面クリアなどに使用できます。

実行後



■(3)コマンド: B

黒画面を描画します。

書式: B(CR)

コマンド実行時

!B

コマンド終了時

OK
!

画面クリアなどに使用できます。

実行後



■(4)コマンド: N

現在の画面の白黒反転

書式: N(CR)

コマンド実行時

!N

コマンド終了時

OK
!

実行前(例)



実行後(例)



■(5)コマンド: D

指定されたイメージ格納ページの画像を描画します。

書式: D [page] (CR)

page の範囲は、0～6。省略時は 0 と見なします。 ※参照”■イメージ格納領域”

例.

コマンド実行時

!D 1

コマンド終了時

OK
!

例. "D 1"実行後



■(6)コマンド: R

指定されたイメージ格納ページ領域に、任意の画像データを格納します。

書式: R [page] (CR)

page の範囲は、0～6。省略時は 0 と見なします。 ※参照”■イメージ格納領域”

ホストから送出する画像データは、200×96ピクセルのモノクロビットマップファイルです。

コマンド実行時

```
!R 1
Please send the BMP data(binary).
```

画像データを送信してください。

コマンド終了時

```
Registered the page 1
!
```

【重要】
画像転送は、必ずバイナリで
行ってください。

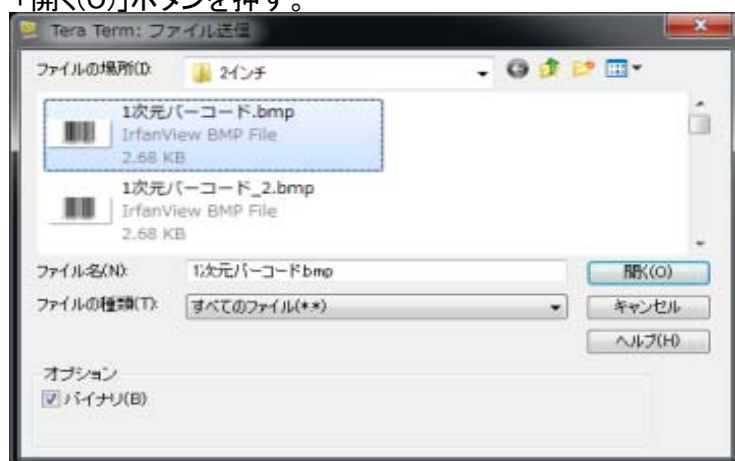
備考: Dコマンドで同じページを描画すれば画像データが確認できます。

■参考: Tera TermでBmpデータを送信するには

Rコマンドを実行し、
“Please send the BMP data(binary).”と表示されたら以下を行います。

(1)「ファイル」メニューの「ファイル送信(S)」を選択 (右図)

(2)「Tera Term: ファイル送信」ダイアログで
オプションの「バイナリ(B)」をチェックする。
送信するBmpファイルを選択する。
※下図では“1次元バーコード.bmp”としています。
「開く(O)」ボタンを押す。



(4) 下図のように“Registered the page #”と表示され、プロンプト(!)が表示されたら完了です。

```
!R 1
Please send the BMP data(binary).
Registered the page 1
!
```

■(7) I

システム情報をTX出力します。

書式: I (CR)

```
!i
/**** System Information ****/
UID: 69813030080A050792081B003200F8FE
H/W: YSPCB-YL_NBEPD01-2_OEPD_EVA2
S/W: EVA2_EPD_2_0_V110_R025
VCC: 3.426
Temperature: 24.4
Elapsed time: 0:48:08
EPD Drew Count : 391
EPD Size : 1.44 inch
EPD Resolution: 0 x 0
EPD Driver : COG G1 Driver
EPD FPL : V110
SP: SP1 (Open), SP2 (Open), SP3 (Open), SP4 (Open)
UART: 115200bps 8N1
!
```

UID: モジュールの固有ID
H/W: 制御基板の名称
S/W: ソフトウェアのバージョン
VCC: 現在の電源電圧[Volt]
Temperature: 現在の温度[°C]
Elapsed time: モジュールの動作時間(=通電時間)
EPD Drew Count: 描画回数(通算)
EPD Size: 電子ペーパーのサイズ
EPD Resolution: 電子ペーパーの画素数
EPD Driver: 電子ペーパーのドライバ名称
EPD FPL: 電子ペーパーのパネル仕様
SP: ショートピンの状態
UART: 通信設定

■(8) DEMO

デモを実行します。

書式: DEMO (CR)

デモは、デモ用画像(6ページ分)の描画を繰り返し行います。

ホストからCTRL+Z(1Ah)を送出すると終了できます。
描画中は、CTRL+Zが効かないので注意してください。
デモを終了すると、プロンプト(!)がTX出力され、コマンド受付状態になります。
デモモードとの違いは、終了後もコマンドモードのままという点です。

DEMOコマンド実行時

```
!demo
Quit CTRL+Z
```

CTRL+Zで終了すると

```
!
```

備考:

ページ間のインターバルはINTREVALコマンドで設定できます。
最終の6ページ目については、LASTWAITコマンドで更にウエイトさせることができます。
※最終ページのみインターバル+ウエイト時間になります。
これらのウエイトは、描画完了から次の描画開始までの時間です。

■(9) INTERVAL

デモのページ切換え時間について表示／設定が行えます。

書式1: **INTERVAL (CR)** [表示]

```
!INTERVAL
Page Interval 3000 ms
!
```

書式2: **INTERVAL #### (CR)** [設定] ####は時間[ミリ秒]

```
!INTERVAL 1234
Page Interval 1234 ms
!
```

※最大 5000 ms です。

■(10) LASTWAIT

デモの最終ページ表示後のウェイト時間について表示／設定が行えます。

書式1: **LASTWAIT (CR)** [表示]

```
!LASTWAIT
Last Page Wait 0 ms
!
```

書式2: **LASTWAIT #### (CR)** [設定] ####は時間[ミリ秒]

```
!LASTWAIT 1234
Last Page Wait 1234 ms
!
```

※最大 5000 ms です。

■(11) STANDBY

R003

パワーオンリセット時に待機画面表示有無を表示／設定できます。

書式1: **STANDBY (CR)** [表示]

```
!STANDBY
Standby Screen Enable
!
```

左例は、待機画面表示が有効なとき

書式2: **STANDBY [0/1] (CR)** [設定] 0は無効、1は有効

```
!STANDBY 0
Standby Screen Disable
!
```

左例は、待機画面表示を無効にする場合

■(12) LED

R003

LED出力の有効／無効を表示／設定できます。

書式1: LED (CR) [表示]

```
!LED  
LED Control Enable  
!
```

左例は、LED出力が有効なとき

書式2: LED [0/1] (CR) [設定] 0は無効、1は有効

```
!LED 0  
LED Control Disable  
!
```

左例は、LED出力を無効にする場合

■(13) P37

R003

P37端子出力の有効／無効を表示／設定できます。

書式1: P37 (CR) [表示]

```
!P37  
Ready Signal Enable!
```

左例は、P37端子出力が有効なとき

書式2: P37 [0/1] (CR) [設定] 0は無効、1は有効

```
!P37 0  
Ready Signal Disable  
!
```

左例は、P37端子出力を無効にする場合

■(14) EXIT

「コマンドモード」を終了し、「通常モード」に遷移します。

書式: EXIT (CR)

```
!exit
>
```

「通常モード」への遷移が完了すると、プロンプト(>)がTX出力されます。

■(15) RESET

制御基板のマイコンをリセットします。

書式: RESET (CR)

```
!reset
>
```

■(16) YSLAB

弊社への連絡先をTX出力します。

書式: YSLAB (CR)

```
!yslab
/=====/
YYY  YYYYYY          YSLAB INFORMATION  (^-^)/
YYY  YYYY            =====
YY   YYYY            COMPANY: Y' s Lab INC.
YYY  YY  SSSSS       URL: http://www.yslab.co.jp/
YYYYY SS   S        EMAIL: info@yslab.co.jp
YYYYY SSSS          TEL: 072-254-5109 FAX: 072-254-5209
YYYYY SS   SSS      3079-13 Nagasone-cho, Kita-ku, Sakai
YYYYY SSSSS         Osaka, Japan      Zip code(591-8025)
/=====/
!
```

■イメージ格納領域

本モジュールは、画像データを格納(保持)するためにイメージ格納領域を持っています。
イメージ格納領域は、7つのページで構成されます。

項番	ページ番号	用途	初期画像	備考
(1)	ページ0	待機画面		ImageBuff[2] 開始アドレス=B6C0h サイズ=2400バイト
(2)	ページ1	デモ用画面-1		ImageBuff[3] 開始アドレス=C020h サイズ=2400バイト
(3)	ページ2	デモ用画面-2		ImageBuff[4] 開始アドレス=C980h サイズ=2400バイト
(4)	ページ3	デモ用画面-3		ImageBuff[5] 開始アドレス=D2E0h サイズ=2400バイト
(5)	ページ4	デモ用画面-4		ImageBuff[6] 開始アドレス=DC40h サイズ=2400バイト
(6)	ページ5	デモ用画面-5		ImageBuff[7] 開始アドレス=E5A0h サイズ=2400バイト
(7)	ページ6	デモ用画面-6		ImageBuff[8] 開始アドレス=EF00h サイズ=2400バイト

■画像データ形式

本モジュールでホストとの通信で用いる画像データは、モノクロビットマップファイルの構造(200×96×1bpp)です。

■BMP形式

ファイルヘッダ BITMAPFILEHEADER構造体
情報ヘッダ BITMAPINFOHEADER構造体
カラーデータ 複数のRGBQUAD構造体
ピクセルデータ

※RGBQUADの個数は
biBitCount=1(2値)
なので2個となる

BMPファイルのサイズは

2750バイト

内訳は、

ヘッダー部=62バイト

ピクセルデータ=2688バイト

※(25+3)×96ライン

※※+3は4バイトの倍数にするため

```
typedef struct tagBITMAPFILEHEADER {
    WORD    bfType;        // 常に"BM"
    DWORD   bfSize;        // ファイルサイズ
    WORD    bfReserved1;   // 0に設定
    WORD    bfReserved2;   // 0に設定
    DWORD   bfOffBits;     // ファイル先頭からピクセルデータ先頭までのオフセット
} BITMAPFILEHEADER;

typedef struct tagBITMAPINFOHEADER {
    DWORD   biSize;        // この構造体のサイズ
    DWORD   biWidth;       // 幅 (ピクセル単位)
    DWORD   biHeight;      // 高さ (ピクセル単位)
    WORD    biPlanes;      // 常に1
    WORD    biBitCount;     // 1ピクセルあたりのカラービットの数
    DWORD   biCompression; // BI_RGB, BI_RLE8, BI_RLE4のいずれか
    DWORD   biSizeImage;   // イメージの全バイト数
    DWORD   biXPelsPerMeter; // 0または水平解像度
    DWORD   biYPelsPerMeter; // 0または垂直解像度
    DWORD   biClrUsed;     // 通常は0、biBitCount以下のカラー数に設定可能
    DWORD   biClrImportant; // 通常は0
} BITMAPINFOHEADER;

typedef struct tagRGBQUAD {
    BYTE    rgbBlue;       // 青の輝度 (0~255)
    BYTE    rgbGreen;      // 緑の輝度 (0~255)
    BYTE    rgbRed;        // 赤の輝度 (0~255)
    BYTE    rgbReserved;   // 予約値。0にする
} RGBQUAD
```

■適応するBMP形式の詳細

offset	00	01	02	03	04	05
0000:	42	4D				※fType "BM"
0002:	BE	0A	00	00		※fSize 2 7 5 0 バイト固定
0006:	00	00				※fReserved1 0 固定
0008:	00	00				※fReserved2 0 固定
000A:	3E	00	00	00		※fOffBits 6 2 バイト目固定
000E:	28	00	00	00		※iSize 4 0 バイト固定
0012:	C8	00	00	00		※iWidth 2 0 0 画素固定
0016:	60	00	00	00		※iHeight 9 6 ライン固定
001A:	01	00				※iPlanes 1 固定
001C:	01	00				※iBitCount 1 (2 値) 固定
001E:	00	00	00	00		※iCompression 0 (BI_RGB: 無圧縮) 固定
0022:	80	0A	00	00		※iSizeImage 2 6 8 8 バイト固定
0026:	00	00	00	00		※iXPelsPerMeter 0 固定
002A:	00	00	00	00		※iYPelsPerMeter 0 固定
002E:	00	00	00	00		※iClrUsed 0 固定
0032:	00	00	00	00		※iClrImportant 0 固定
0036:	00					※gbBlue 0 固定 or 2 5 5 固定
0037:	00					※gbGreen 0 固定 or 2 5 5 固定
0038:	00					※gbRed 0 固定 or 2 5 5 固定
0039:	00					※gbReserved 0 固定
003A:	FF					※gbBlue 2 5 5 固定 or 0 固定
003B:	FF					※gbGreen 2 5 5 固定 or 0 固定
003C:	FF					※gbRed 2 5 5 固定 or 0 固定
003D:	00					※gbReserved 0 固定
003E:	9 6 ライン目 (2 5 バイト + 3 バイト)				00	00 00
005A:	9 5 ライン目 (2 5 バイト + 3 バイト)				00	00 00
0076:	9 4 ライン目 (2 5 バイト + 3 バイト)				00	00 00
	⋮					
0A6A:	3 ライン目 (2 5 バイト + 3 バイト)				00	00 00
0A86:	2 ライン目 (2 5 バイト + 3 バイト)				00	00 00
0AA2:	1 ライン目 (2 5 バイト + 3 バイト)				00	00 00
0ABE:						

部は、全て固定値になるはずです。

部は、ピクセルデータなので画像により変わります。