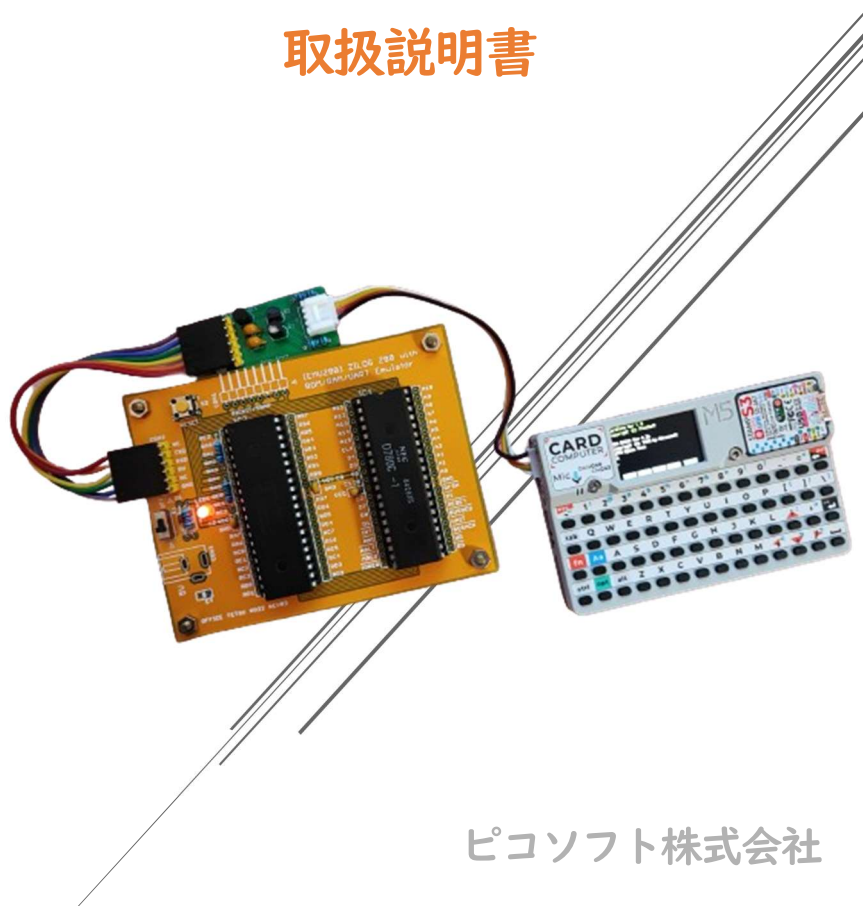


Cardputer

シリアルコンソール化キット

取扱説明書



ピコソフト株式会社

はじめに

電腦伝説(*1)さんが設計された EMUZ80(*2)というレトロでモダンなマイコンボードがあります。このようなマイコンボードはUSB シリアルケーブルでPC と接続して、PC のシリアルコンソールで使用することを前提としています。

M5Stack Cardputer の実用的な使い方を考えたときに、「Cardputer と EMUZ80 を簡単に接続できるようにすること」を最初に思いつきました。(いや、最初に思いついたのはCardputer をポケコン化することでした。順序が変わりますが、ポケコン化キットも頒布予定です。)

「Cardputer シリアルコンソール化キット」は、1本のケーブルでCardputer とマイコンボードを接続しCardputer 上にシリアルコンソールを構築します。PC 不要で、EMUZ80 のようなマイコンボードが試せます。

また、Cardputer 上のシリアルコンソール「CardTerm」は全ソースコードを添付しています。Cardputer 上のソフトウェア開発に大いに参考になるはずです。ぜひご活用ください。

*1 https://vintagechips.wordpress.com/2022/03/05/emuz80_reference/

*2 http://www.amy.hi-ho.ne.jp/officetetsu/storage/emuz80_techdata.pdf

目次

1. 組立て/配線	3
1.1. 組立て.....	3
1.2. 配線.....	5
2. CardTerm	6
2.1. VSCode のダウンロードとインストール.....	6
2.2. PlatformIO のインストール.....	7
2.3. ビルド&書込み.....	9
3. 操作方法	11
3.1. 起動.....	11
3.2. ターミナル.....	13
3.3. スクリーンエディター	14
付録1 エスケープシーケンス	16
付録2 キーコード	18
付録3 表示キャラクター	18
付録4 配布ファイルの内容	19

1. 組立て/配線

1.1. 組立て

組立てにははんだ付けが必要です。基板の印刷のある面を上にして各部品（部品リスト参照）を差し込み、下側からはんだ付けします。はんだ付けする順番は特にありません。部品によっては、紙粘着テープ等で部品を固定しておくとうりやすいです。

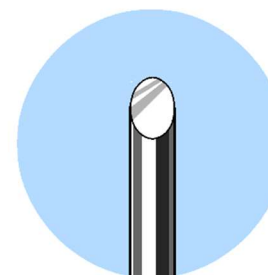
はんだ付けには以下のものをご用意ください。

- ・はんだこて
- ・こて台
- ・はんだ(0.8mm)
- ・精密ニッパー
- ・紙粘着テープ

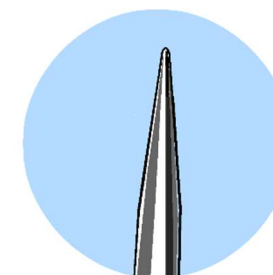
【参考】

はんだこては温度制御機能付きのもの(HAKKO FX-600、FX-951 など)をおすすめします。特に鉛フリーはんだを使用する場合は必須です。鉛フリーはんだを使用する場合は350℃くらいに設定しておくとういでしょう。

また、標準のこて先はB型と呼ばれる万能タイプのものが付属していますが、BC型(円すいを斜めにカットした型)やC型(円柱を斜めにカットした型)に交換すると、さらに使いやすくなります。(HAKKO FX-600にはC型のT18-C2が適合します。)



斜めにカットした型



標準型

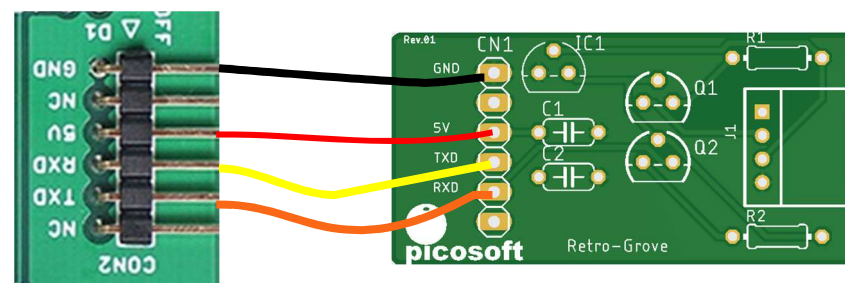
部品リスト

No.	名称	個数	場所	説明
001	シリアル信号コンバーター専用プリント基板	1		印刷のある面を上にして各部品を差し込み、下側から半田付けします。部品によっては、紙粘着テープ等で部品を固定しておくともやりやすいです。
002	10kΩ	2	R1、R2	抵抗のカラー帯が 茶黒黒赤茶 です。極性はありませんので向きはどちらでもかまいません。
003	10μF	2	C1、C2	部品に 106 と印刷してあるものです。極性はありませんので向きはどちらでもかまいません。
004	2N7000	2	Q1、Q2	3 端子の FET です。 7333-A と全く同じ形状ですので、注意してください。基板上的印刷に合わせてください。
005	7333-A	1	IC1	3 端子の LDO です。 2N700 と全く同じ形状ですので、注意してください。基板上的印刷に合わせてください。
006	L 型ピンヘッダー6P	1	CN1	基板上的印刷に合わせてください。
007	Grove 用 4P コネクタ-L 型	1	J1	基板上的印刷に合わせてください。 この部品はしっかりとんだ付けして確実に固定してください。
008	6P コネクター付きケーブル	1		L 型ピンヘッダー6P に差し込みます。
009	Grove ケーブル	1		Grove 用 4P コネクター-L 型に差し込みます。

1.2. 配線

信号レベル 5V 側端子(CN1)はよく使われている「FTDI USB シリアル変換ケーブル」の信号と同じ並びになっています。(ただし、CTS と RTS は未接続です)

ターゲットのマイコンボードとは TXD と RXD がクロスするように接続します。(一般にマイコンボードの信号の並びは CPU 側から見た信号が書いてあります。**6P コネクター付きケーブルで接続した場合は、線は平行になりクロスすることはありません。**また、付属のケーブルの色はランダムですのでご注意ください。)



Grove ケーブルは逆差しできない構造になっています。GND 用の黒い線を合わせて接続します。本製品と Cardputer を Grove ケーブルで接続するとケーブルが捻れた状態になるのが正解です。

2. CardTerm

Cardputer 本体に CardTerm というファームウェアを書き込む必要があります。また、CardTerm はソースコードでのみのご提供ですので、ご自身でビルドする必要があります。

CardTerm は Arduino IDE ではなく、Visual Studio Code(以降 VSCode)とその拡張機能の Platform IO を用いて開発しています。

2.1. VSCode のダウンロードとインストール

[VSCode ダウンロードページ](#)で、OS を指定して VSCode のインストーラーをダウンロードします。

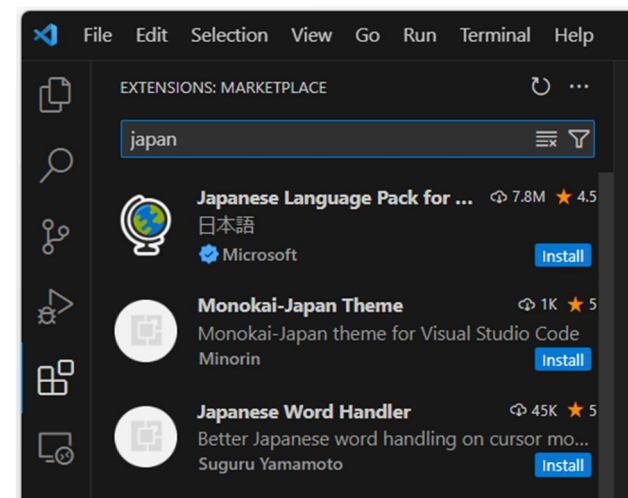
Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.

OS	Installation Method	Supported Architectures
Windows (Windows 10, 11)	User Installer	x64, Arm64
	System Installer	x64, Arm64
	.zip	x64, Arm64
	CLI	x64, Arm64
Linux (Debian, Ubuntu, Red Hat, Fedora, SUSE)	.deb	x64, Arm32, Arm64
	.rpm	x64, Arm32, Arm64
	.tar.gz	x64, Arm32, Arm64
	Snap	Snap Store
	CLI	x64, Arm32, Arm64
Mac (macOS 10.15+)	.zip	Intel chip, Apple silicon, Universal
	CLI	Intel chip, Apple silicon

ダウンロードが完了したら、インストーラーを実行して VSCode をインストールしてください。


2.2. PlatformIO のインストール

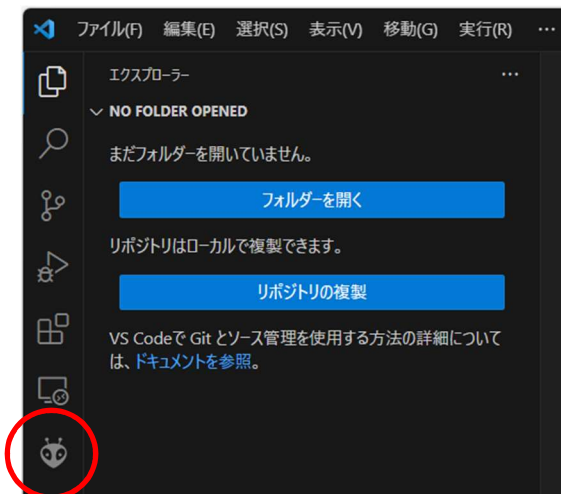
VSCode を起動したら、Japanese Language Pack for VS Code をインストールしましょう。Ctrl+Shift+X キーを押して検索入力フィールドに japan 入力し、Japanese Language Pack for VS Code の下の Install ボタンをクリックします。インストール後、Restart ボタンが現れたらクリックし再起動します。



同様に Ctrl+Shift+X キーを押して検索入力フィールドに plat と入力し、拡張機能から PlatformIO IDE をインストールします。



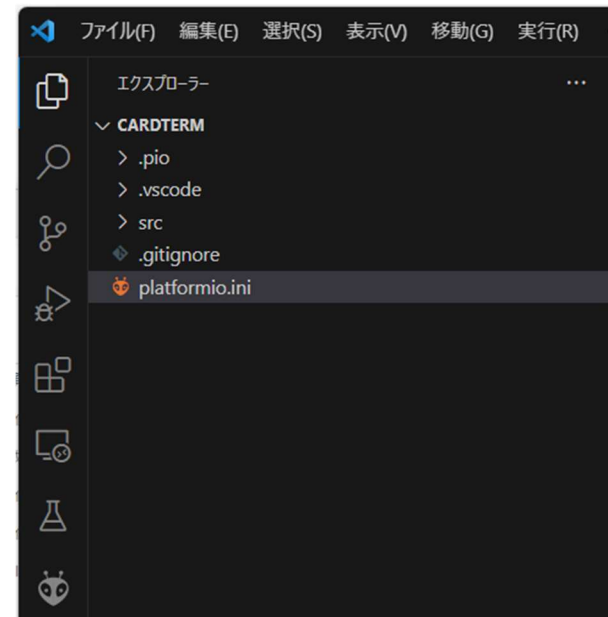
VSCode を終了し、再起動すると左側にアイコンが現れます。



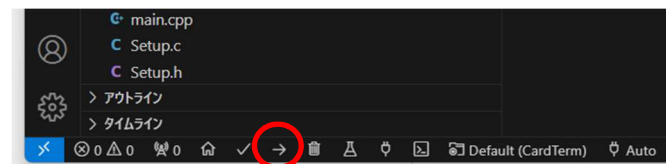
2.3. ビルド&書込み

VSCode を起動したら、ファイル(F)→フォルダーを開くを選択してください。USB メモリーのルート直下の **CardTerm フォルダ**を指定します。(実際には USB メモリーの内容をハードディスクにコピーしていると思いますので、ハードディスク上の該当フォルダを指定してください。)

しばらくすると、次のような画面になります。



この状態で、画面下のをクリックします。



SUCCESS と表示されたら書き込み完了です。

```
問題 出力 デバッグコンソール ターミナル ポート
Writing at 0x0009aacd... (90 %)
Writing at 0x000a03cd... (95 %)
Writing at 0x000a649f... (100 %)
Wrote 616832 bytes (344978 compressed) at 0x00010000 in 4.4 seconds (effective
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 55.30 seconds
* ターミナルはタスクで再利用されます、閉じるには任意のキーを押してください。
It (CardTerm)  Auto
```

失敗した場合は、USB ケーブルを抜き差ししてから再度実行してみてください。

Arduino IDE でのビルド方法

普段、Arduino IDE で開発されている方も多いかと思いますが、この機会に VSCode に移行することをおすすめします。

VSCode ではなく Arduino IDE(2.3.0)でもビルド&書き込みができることを確認しています。USB メモリーのルート直下に Arduino フォルダがあります。その直下の CardTerm というフォルダの中に CardTerm.ino というファイルがあります。Arduino IDE から CardTerm.ino を開いてビルド&書き込みを実行してください。

最初は失敗するかと思います。失敗した場合は以下の点を確認しましょう。

- ・ボード名は M5Cardputer になっているか？
- ・必要なライブラリーをインストールしているか？
- ・COM ポートは認識されているか？

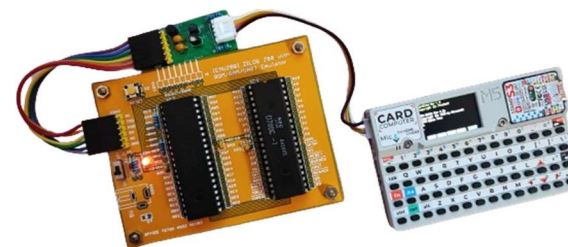
3. 操作方法

3.1. 起動

CardTerm の起動時には、Cardputer 内の microSD カードをチェックします。microSD カードは FAT32 でフォーマットし、ルート直下に `setup.sys` をコピーしておいてください。(setup.sys は USB メモリー内にあります。)

setup.sys が読み込めなかった場合はデフォルト起動します。(9600bps、送信遅延 30 ミリ秒/字、100 ミリ秒/行)

Cardputer に EMUZ80 ボード等のマイコンボードを接続し、電源を入れます。EMUZ80 はバスパワーでも動作しますので、Cardputer の左側にあるスライドスイッチを 5VOUT 側にスライドし、Cardputer から電源を供給することも可能です。



起動するとロゴを表示しますので、適当なキーを押します。



ターミナル画面になります。この状態でキー入力すると送信し、受信した文字があれば画面表示します。



EMUZ80のリセットボタンを押すと、EMUZ80の起動メッセージが表示されます。

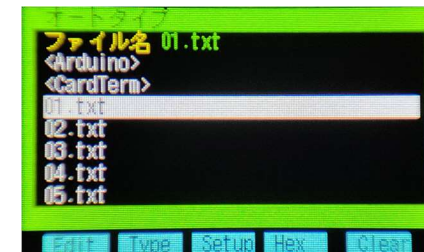


3.2. ターミナル

ターミナル画面では、以下のファンクションキー(fnキーと数字キーの同時押し)が使用できます。

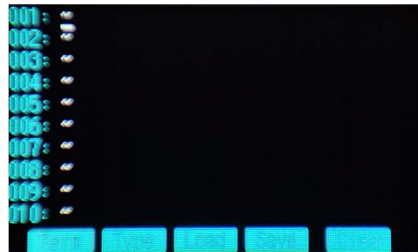
キー	機能	説明
fn + 1	Edit	エディター画面に切り替わります。
fn + 2	Type	microSD カード内のファイルを送信します。ファイルを指定してください。
fn + 3	Setup	エディター画面に切り替わり、setup.sysを読み込みます。エディター画面で編集後、ファイルをセーブしてからリセットボタンを押してください。
fn + 4	Hex	ファンクションキーが黄色になり、16進表示モードになります。再度押すと解除されます。
fn + 5	Clear	画面をクリアします。

オートタイプ時のファイル名指定は、fn + ▼またはfn + ▲で移動します。ファイル名で^{ok}を押すとオートタイプが開始されます。フォルダー名で^{ok}を押すとそのフォルダーに移動します。fn + ESCでキャンセルできます。



3.3. スクリーンエディター



ターミナル画面で **fn + I** を押すと、エディター画面に切り替わります。

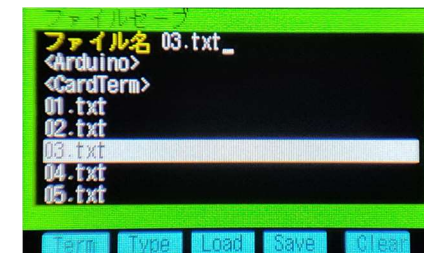


エディター画面では、以下の編集キーやファンクションキー（fn キーと数字キーのを同時押し）が使用できます。

キー	機能	説明
fn + ◀	LEFT	カーソルを左へ移動します。
fn + ▶	RIGHT	カーソルを右へ移動します。
fn + ▲	UP	カーソルを上へ移動します。
fn + ▼	DOWN	カーソルを下へ移動します。
 BS	BS	カーソルの左の文字を削除します。
fn + 	DEL	カーソル上の文字を削除します。
Opt + ▲	PAGE UP	前ページに移動します。
opt + ▼	PAGE DOWN	次ページに移動します。
opt + H	HOME	カーソルを行頭に移動します。
opt + E	END	カーソルを行末に移動します。

キー	機能	説明
fn + I	Term	ターミナル画面に戻ります。
fn + 2	Type	ターミナル画面に戻り、編集中のテキストを送信します。
fn + 3	Load	microSD カード内のテキストファイルをロードします。
fn + 4	Save	編集テキストを microSD カードにセーブします。
fn + 5	Clear	編集テキストをすべてクリアします。

ファイルロード時のキー操作はオートタイプのキー操作と同じです。ファイルセーブ時のキー操作はファイルロード時とほぼ同じですが、最後にファイル名入力欄を編集してから  を押してください。編集には  のみを使用できます。



付録1 エスケープシーケンス

シーケンス	機能	備考						
ESC [K	カーソル位置から行末までを消去する。							
ESC [0 K	カーソル位置から行末までを消去する。							
ESC [I K	行頭からカーソル位置までを消去する。							
ESC [2 K	行全体を消去する。							
ESC [J	カーソル位置から画面末尾までを消去する。							
ESC [0 J	カーソル位置から画面末尾までを消去する。							
ESC [I J	画面先頭からカーソル位置までを消去する。							
ESC [Ps A	カーソルを Ps 行上に移動する。Ps の省略時の値は 1。							
ESC [Ps B	カーソルを Ps 行下に移動する。Ps の省略時の値は 1。							
ESC [Ps C	カーソルを Ps 行右に移動する。Ps の省略時の値は 1。							
ESC [Ps D	カーソルを Ps 行左に移動する。Ps の省略時の値は 1。							
ESC [Ps l ; Ps2 H	カーソルを Ps l 行目の Ps2 桁目に移動する。Ps l、Ps2 の省略時は 1。							
ESC [Ps l ; Ps2 f	カーソルを Ps l 行目の Ps2 桁目に移動する。Ps l、Ps2 の省略時は 1。							
ESC [H	カーソルをホーム位置に移動する。							
ESC 7	カーソル位置を保存する。							
ESC 8	保存したカーソル位置を復元する。							
ESC D	カーソルを一行下に移動する。							
ESC M	カーソルを一行上に移動する。							
ESC E	カーソルを次の行の先頭に移動する。							
ESC [Ps M	カーソルのある行から Ps 行を削除する。Ps の省略時の値は 1。	未実装						
ESC [Ps L	カーソルのある行の前に Ps 行空行を挿入する。Ps の省略時の値は 1。	未実装						
ESC Pm m	文字属性を設定する。省略時の Pm の値は 0。 <table border="1" data-bbox="347 1273 645 1386"> <thead> <tr> <th>Pm</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>すべての属性を解除する。</td> </tr> <tr> <td>4</td> <td>下線属性を設定する。</td> </tr> </tbody> </table>	Pm	意味	0	すべての属性を解除する。	4	下線属性を設定する。	下線属性は未実装
Pm	意味							
0	すべての属性を解除する。							
4	下線属性を設定する。							

	7	反転属性を設定する。	
	30	文字色 BLACK	
	31	文字色 RED	
	32	文字色 GREEN	
	33	文字色 YELLOW	
	34	文字色 BLUE	
	35	文字色 MAGENTA	
	36	文字色 CYAN	
	37	文字色 WHITE	
	40	背景色 BLACK	
	41	背景色 RED	
	42	背景色 GREEN	
	43	背景色 YELLOW	
	44	背景色 BLUE	
	45	背景色 MAGENTA	
	46	背景色 CYAN	
	47	背景色 WHITE	
ESC [? 2 5 l		カーソルを非表示にする。	
ESC [? 2 5 h		カーソルを表示する。	
ESC [4 l		上書きモードにする。	未実装
ESC [4 h		挿入モードにする。	未実装
ESC [Ps l ; Ps2 r		上下マージン(スクロールリージョン)の設定は無視する。	

付録2 キーコード

キー	シーケンス
↑	ESC [A
↓	ESC [B
←	ESC [D
→	ESC [C

付録3 表示キャラクター

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p								
1			!	1	A	Q	a	q								
2			”	2	B	R	b	r								
3			#	3	C	S	c	s								
4			\$	4	D	T	d	t								
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7			'	7	G	W	g	w								
8			(8	H	X	h	x								
9)	9	I	Y	i	y								
A			*	:	J	Z	j	z								
B			+	;	K	[k	{								
C			,	<	L	¥	l									
D			-	=	M]	m	}								
E			.	>	N	^	n	~								
F			/	?	O	_	o									

付録4 配布ファイルの内容

<CardTerm>	<src>	<editor>	TextEditor.c TextEditor.h	TextEditor 部品	
		<iocs>	Fio.cpp Fio.h	SD カード処理ラッパー	
			Iocs.c Iocs.h	I/O 処理	
			Kb.c Kb.h	キー入力処理	
			Lcd.cpp Lcd.h	LCD 表示ラッパー	
			Uart.cpp Uart.h	UART 処理ラッパー	
			<gtext>	Font.c Font.h	東雲フォント
				Gtext.c Gtext.h	テキスト表示処理
		<vt100>	Vt100.c Vt100.h	VT100 サブセット表示処理	
		main.cpp		CardTerm メイン処理	
		Edit.c Edit.h		TextEditor 部品を使ったスクリーンエディターの実装	
		Error.c Error.h		エラーメッセージ表示処理	
		Field.c Field.h		フィールド入力処理	
		Filer.c Filer.h		ファイラー	
		Fnkey.c Fnkey.h		ファンクションキー表示処理	
Setup.c Setup.h		セットアップファイル読み込み処理			
Splash.c Splash.h		起動画面表示処理			
Term.c Term.h		シリアルコンソール			
	platformio.ini		PlatformIO 設定ファイル		
	setup.sys		CardTerm 設定ファイル		
	Cardputer シリアルコンソール化キット取扱説明書.pdf		このファイル		
	Retro-Grove-sch.pdf		信号レベルコンバーター回路図		
<Arduino>			Arduino 用ソース一式		

CardTerm のソースコードを無変更でビルドしたもの、あるいは修正を加えてビルドしたものを配布してかまいません。バイナリーファイルは非商用/商用に関わらず自由に配布することができます。ただし、著作権表示は削除しないでください。また、ソースコードそのものの配布はご遠慮ください。ご不明な場合は、お問い合わせ、ご相談ください。

Cardputer シリアルコンソール化キット取扱説明書

2024 年 2 月 14 日 1.0 版発行

著者 宇喜多裕一

発行 ピコソフト株式会社

<http://www.picosoft.co.jp>

本書の無断複写は、著作権法上の例外を除き、禁じられています。