

RX113 タッチキー評価キット R RX130 タッチキー評価キット RX130-100 タッチキー評価キット RX231 タッチキー評価キット RX230 タッチキー評価キット

[ソフトウェア編] マニュアル

-本書を必ずよく読み、ご理解された上でご利用ください



- Hokuto Electronic

注意事項	1
概要	2
1. 全体構成	3
1.1. CD フォルダ構成	
2. WORKBENCH6 を使用したチューニング	4
2.1. 自己容量タッチキー基板(S16A)	4
2.1.1. ボードのセットアップ	
2.1.2. Workbench6 の起動	4
2.1.3. CS+への接続	4
2.1.1. 感度の確認	7
2.1.2. 再チューニング	
2.2. 相互容量タッチキー基板(D55A)	
2.2.1. ボードのセットアップ	
2.2.2. 感度の確認	
2.2.3. 再チューニング	
2.3. シリアルポート接続に関して	
3. サンプルプログラム	21
3.1. サンプルプログラムで使用している定義値・変数等の説明	
4. 自己容量キー読み出しサンプル	23
	23
4.1. ク ハンプ 備成	
	25
4.4. プログラム説明	
4.4.1. フローチャート	
4.4.2. 実際の測定値の変化例	
5. 相互容量キー読み出しサンプル	
	31
5.1. ノーヘノリー構成	
5.2. 旧立谷里、 ハハ	32
5.5. 周奴	32
541 フローチャート	
5.4.2. 実際の測定値の変化例	
6 アプリケーション例サンプル	34
0.1. ノーヘノリー 悟风	• • •
らつ 問料	
6.2. 関数	

RX タッチキー評価キット[ソフトウェア編] 株式会社 オレチ電子

	6.3.	3.1. フローチャート	
7.	LCE	D 表示サンプルプログラム[セグメント LCD 向け]	37
	7.1.	ソースツリー構成	
	7.2.	関数	
8.	LCE	D 表示サンプルプログラム[キャラクタ LCD 向け]	41
	8.1.	ソースツリー構成	41
	8.2.	関数	41
	8.3.	操作例	
9.	付鐞	録	44
	取扱訪	説明書改定記録	
	お問合	合せ窓口	44



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】

- 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。
- 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。

RX タッチキー評価キット[ソフトウェア編] 株式会社



概要

本書は、

「RX113 タッチキー評価キット R」 「RX130 タッチキー評価キット」 「RX130-100 タッチキー評価キット」 「RX231 タッチキー評価キット」 「RX230 タッチキー評価キット」 付属 CD 内のサンプルプログラムについて解説を行うものです。

本書ではルネサスエレクトロニクス社製ツールである、Workbench6 を使用したプロジェクトのサンプル及び、「自己 容量タイプタッチキー基板(S16A)」及び「相互容量タイプタッチキー基板(D55A)」のキー読み取りのサンプルプログラ ムを示しています。

本書で解説するサンプルプログラムはキーパッドに触れた場合に触れているキーを検出する部分のエッセンスを抜き出したものとなっています。そのため、単純で短いコードとなっており、キーにタッチしたときの測定値の変化をモニ タするのに適しています。

実際のタッチキーアプリケーションを組む際には、ノイズや特性の経時変化等を考慮する必要がある場合がありま すので、Workbench6 ベースでプログラムを作成するか、タッチキーの仕組みを理解した上で、お客様のシステムに 応じたプログラム設計をお願い致します。



1. 全体構成

1.1. CD フォルダ構成

・RX113 タッチキー評価キット R

WB6/	RX113_TouchAPI_S16A/	Workbench6 を使用した自己容量サンプル
	RX113_TouchAPI_D55A/	Workbench6 を使用した相互容量サンプル
SOURCE/	RX113_Self_cap_S16A_sample/	自己容量サンプルプログラム
	RX113_Mutual_cap_D55A_sample	相互容量サンプルプログラム
	RX113_Self_cap_S16A_appli/	自己容量アプリケーションサンプル
DOCUMENT/	RX_TouchKey_Software_REV_x_s.pdf	本マニュアル

・RX130 タッチキー評価キット

·RX130-100 タッチキー評価キット

WB6/	RX130-80_TouchAPI_S16A/ RX130-100_TouchAPI_S16A/	Workbench6を使用した自己容量サンプル
	RX130-80_TouchAPI_D55A/ RX130-100_TouchAPI_D55A/	Workbench6 を使用した相互容量サンプル
SOURCE/	RX130-80_Self_cap_S16A_sample/ RX130-100_Self_cap_S16A_sample/	自己容量サンプルプログラム
	RX130-80_Mutual_cap_D55A_sample RX130-100_Mutual_cap_D55A_sample	相互容量サンプルプログラム
	RX130-80_Self_cap_S16A_appli/ RX130-100_Self_cap_S16A_appli/	自己容量アプリケーションサンプル
DOCUMENT/	RX_TouchKey_Software_REV_x_s.pdf	本マニュアル

※-80 が付くものは、80 ピンのマイコンボード(HSBRX130F80)向けのサンプル、-100 が付くものは、100 ピンのマイ コンボード(HSBRX130F100)向けのサンプルとなります

・RX231 タッチキー評価キット

・RX230 タッチキー評価キット

WB6/	RX231_TouchAPI_S16A/	Workbench6を使用した自己容量サンプル
	RX230_TouchAPI_S16A/	
	RX231_TouchAPI_D55A/	Workbench6を使用した相互容量サンプル
	RX230_TouchAPI_D55A/	
SOURCE/	RX231_Self_cap_S16A_sample/	自己容量サンプルプログラム
	RX231_Mutual_cap_D55A_sample	相互容量サンプルプログラム
	RX231_Self_cap_S16A_appli/	自己容量アプリケーションサンプル
DOCUMENT/	RX_TouchKey_Software_REV_x_s.pdf	本マニュアル

本 CD に格納されているサンプルプログラムソースは、ルネサスエレクトロニクス社 CS+向けのプロジェクトとなっております。(x は、ファイルのバージョンが入ります)



lokuto Sectronic Workbench6 を使用したチューニング

Workbench6は、ルネサスエレクトロニクス社の新しいタッチキーに対応したツールで、タッチキーのプログラムの自 動生成及びタッチ感度の最適化を行うためのツールです。

Workbench6を使用する事により、タッチキーアプリケーションの開発を容易に行う事ができます。

Workbench6を使用する際は、別途 E1 をご用意ください。

2.1. 自己容量タッチキー基板(S16A)

2.1.1. ボードのセットアップ

マイコンボードと、自己容量タッチキー基板(S16A)、LCD 及び E1 を接続。ボードには、電源を供給してください。

2.1.2. Workbench6 の起動

Workbench6は、ルネサスエレクトロニクス社のWebより最新版をダウンロードしてインストールを予め行ってくだ

さい。

ルネサスエレクトロニクス Web ツールダウンロード検索 https://www.renesas.com/ja-jp/search/keyword-search.html 「Workbench6」で検索してください

※古いバージョンのものは、RX130(512KB版)やRX231等の新しいマイコンに対応していない場合がありますの で、最新版を使用する事を推奨致します

2.1.3. CS+への接続



接続-CS+接続を選択





静電容量タッチセンサ・プロ	ジェクト・セットアップ for CS+
静電容量タッチセ 静電容量タッチセンナ	アンサ・プロジェクト・セットアップ for CS+ ナ・プロジェクト・ファイルとロード・モジュール・ファイルを選択してください。
プロジェクト・ファイル(E):	参照(B) (1)
ロード・モジュール・ファイル(∟)	参照(民)
	ОК キャンセル

付属 CD 内の

WB6/RXnnn_TouchAPI_S16A/ (nnn にはマイコン名が入ります)

を、PC 内の適当なフォルダにコピーし、

上記の画面で、

(1)参照ボタンを押し、Workbench6向けの自己容量タッチキー基板のプロジェクトを選択する

隆理 ▼ 新しいフォル	ダー			!≡ ▼ 🚺
★ お気に入り ダウンロード	ドキュメント ライブラリ RX130_TouchAPI_S16A		<u>غَلَد</u> ُ	べ替え: フォルダー、
🔜 デスクトップ	名前	更新日時	種類	サイズ
週 最近表示した場所	🎍 DefaultBuild	2016/10/14 15:05	ファイル フォル…	
💪 OneDrive	\mu Include	2016/10/14 14:27	ファイル フォル…	
=	퉬 sci	2016/10/14 15:49	ファイル フォル	
フライブラリ	퉬 Source	2016/10/19 10:25	ファイル フォル…	
Subversion	퉬 wbSetting	2016/10/14 14:40	ファイル フォル…	
ドキュメント	TouchApiBase.mtpj	2016/10/14 18:55	MTPJ ファイル	5,898 KB
📔 ピクチャ				
📑 ビデオ				
ー シュージック				
💂 コンピューター 🖕				

プロジェクトファイルのファイル名は、TouchApiBase.mptj となっています。

ロード・モジュール・ファイルの方は自動的に選択されますので、変更の必要はありません。

RX タッチキー評価キット[ソフトウェア編] 株式会社 **北手電子**



静電容量タッチセンサ・プロジェクト・セットアップ for CS+	×
静電容量タッチセンサ・プロジェクト・セットアップ for CS+ 静電容量タッチセンサ・プロジェクト・ファイルとロード・モジュール・ファイルを選択してください。	
プロジェクト・ファイル(巴):	
C:\Users\win64-5\Documents\cubesuite\RX130-100_TouchAPI_S16A\TouchApiBas	参照(巴)
ロード・モジュール・ファイル(生):	
C:\Users\win64-5\Documents\cubesuite\RX130-100_TouchAPI_S16A\DefaultBuild\`	参照(<u>R</u>)
OK (2) キャンセル	

(2)OK を押す

Г

CS+が起動して、プロジェクトファイルが開かれます。

※CS+の設定でスタートアップメッセージが表示される場合は閉じてください

Navigator	Renesas	Wor	kbench <mark>6</mark>
		VV OI	KDEIICH 🚺
First Step Guide	Tuning Window	User's Manual	Document Manager
	(3)		

左下の表示が CS+接続になっていれば、接続は成功です。 TUNING WINDOW のボタンを押してください。

RX タッチキー評価キット[ソフトウェア編] 株式会社



2.1.1. 感度の確認

Workbench6					
接続(<u>C</u>) 表示(⊻) 調整ツール(<u>U</u>) ツール(<u>I</u>) '	カインドウ(<u>W</u>) ヘルプ(<u>H</u>)				
Board Monitor(B)	- വ്വ്വ് ഡ്				
9_TS09 - TS: 09 Uni Multi Monitor(U)		A_TS10 - TS: 10 Unit: 0	Button11 - TS: 11 Unit: 0		
20000 Difference Monitor(D)	ヌッナレF: ホタン ユニット: 0 ナヤネル: 9 11	9000			
Measurements(<u>M</u>)					
Setup Parameters(P)	1	8200			
・ モニタ開始(工)					
19000					
	1	7400			
					基準任
	11	6600			タッチしきい
					差分1
18000	カウント1道:				
	11	5800			
	基準値:				
	タッチしきい値: 1	5000			
17000	差分値:	(THE) (THE)			
	メテータス 🗠	Difference Board Mor	litor	12 A7 M	
		ボタン: 14 マトリックスボタン: 0	スライダ:0 ホイール:0	オーバフロー:	
				TSCAP電圧異常:	
10000		ターゲットボード			
		0_TS01 1_TS03	2_TS05 3_TS06	4_TS08 A_TS10	O_TS14 D_TS1
		5 7500	6 TS02 7 TS04	8 7507 0 7500	F TS13
15000		0_1300	0_1002 7_1004	0_1007 8_1008	121313

TUNING WINDOW を押すと上記画面が開きます。

調整ツールーモニタ開始

で、マイコンボードの動作を PC 上からモニタします。

Workbench6 (位) 表示(公) 調整ツール(U) ツール(I)	ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)	TS00	(キーパッ	ド 5) のク	ブラフ表	7				
5 TS00 - TS:00 Upit: 0	Max • <mark>of at w</mark> */	Multi Monitor					Measurements			
	タッチ1/F:ポタン ユニット:0 チャネル:0	22000					ラベル 5_TS00 (00 - 00)	カウント値 15182	プライマリ セカン 	· 例 基準 - 15195
		20400				5, T800: 15187 0_T801: 15249	6 TS01 (00 - 01) 6 TS02 (00 - 02) 1 TS03 (00 - 03) 7 TS04 (00 - 04)	15234 15155 15234 15210		- 15594 - 15594 - 15281 - 1532
		18800				6_T\$02: 15162 1_T\$08: 15241	2_TS05 (00 - 05) 3_TS06 (00 - 06) 8_TS07 (00 - 07) 4_TS08 (00 - 09)	15125 15202 15248 15240		- 1515 - 1524 - 1540 - 1520
		17200				7_T604: 15241 2_T805: 15130	9_TS09 (00 - 09) A_TS10 (00 - 10) F_TS13 (00 - 13)	15170 15280 15327		- 1538 - 1529 - 1538
		15600]	3_T\$06: 15211 8_T\$07: 15235	C_TS14 (00 - 14) D_TS15 (00 - 15)	15207 15133		- 1521 - 1516
	カウント値 15187	14000								
	基準値 15195	Difference Board N	Ionitor			• X	-			
	差分値 00008 ステータス: 列:デ・オブ	- タッチレア定義 ボタン: 14 マトリックスボタン	: 0 スライダ: 0 ホイール: 0	エラーステ・ オーバフロ TSCAPT	-92. 0 : ■ 配圧異常: ■	キーパッド表	示			
		ターゲットボード						※本ī	画面は	_
		0_TS01 1_TS0	8 2_TS05 8_	rso6 4_trso8	A_TS 10	O_TS14 D_TS15	1	RX13 のもの	30 (100 りです	ピン
	- L						1	その=	キットで	使用
		(2	6_TS02 7_	8_TS07	9_TS09	F_TS18		能な	モーか す	衣
							•			





(1)Status Monitor 画面を開く(アイコンまたは、調整ツール-Status Monitor) (2)TS00の画面を見ながら、タッチキー基板の5(TS00)のキーに触れる

青いラインが非タッチ時の基準値 赤いグラフが、現在の測定値 緑のラインが、タッチ・非タッチを判断する閾値 となっています。

本ツールを使用することにより、各キーのタッチ判定に対するマージン等を確認する事ができます。

終了させる際は、 調整ツールーモニタ停止

接続一切断

で、CS+接続を切った後、Workbench6を終了させてください。

2.1.2. 再チューニング

付属の CS+プロジェクトは、マイコンボードとタッチキー基板の組み合わせで一度チューニングされたものとなります が、誤作動やタッチ感度に問題があれば、再度 Workbench6 を用いてチューニングを行う事ができます。

※チューニング時は LCD がノイズ源となる可能性がありますので、LCD 画面を外した状態で行う事を推奨します。



Workbench6 を起動後、

START ボタンを押す。





 First Step Guide 静雷容量タッラ 	チセンサ・プロジェクトの新規作成/再調整								
Workbench 6	 ●静電容量タッチセンサ・プロジェクトの新規作成 								
	 ・ ・ ・								
	win64-5¥Documents¥cubesuite¥RX130-100_TouchAPI_S16A 参照… 静電容量タッチセンサ・プロジェクトフォルダは以下の形式で命名されます。 TouchAPI_YYYYMMDDhhmmss YYYY:年 MM:月 DD:日 hh:時 mm:分								
	ss:秒 ← 戻る → 次へ 図 キャンセル								

静電容量タッチセンサ・プロジェクトの再調整。

RX*nnn*_TouchAPI_S16A/ (nnn にはマイコン名が入ります)

フォルダを選択

G First Step Guide	
再調整プロセ	スの選択
Workbench 6	First Step Guideの再調整プロセスを選択してください。
	静電容量タッチセンサ・インタフェースの定義
	静電容量タッチセンサ・インタフェースの定義
	寄生容量計測
	センサオフセット調整 感度調整
A STATE OF BEILD	
	← 戻る → 次へ × キャンセル

再調整プロセスの選択





・タッチインタフェースの定義

キーパッドの配置や数等を変更します。

キット付属の自己容量タッチキー基板(S16A)を使用する限りでは、変更の必要はないと考えます。

・タッチ電極とTS 端子間の抵抗の選択

マイコンとキーパッド間に実装されている抵抗値を変更します。

キット付属の自己容量タッチキー基板(S16A)を使用する限りでは、変更の必要はないと考えます。

寄生容量の計測

キーパッドの容量値を計測します。

基板による個体差が大きく出るものではありませんので、キット付属の自己容量タッチキー基板(S16A)を使用する 限りでは、実行の必要はないと考えますが、実行しても問題ありません。

・センサーオフセット調整

マイコン側のパラメータのチューニングを行います。マイコンチップ毎に、オフセットの最適値は異なりますので、本 調整により、動作マージンは向上すると考えます。

センサーオフセット調整を選択。次へ進むと、以下の画面となります。

。First Step Guide ターゲットボー	 ド起動
Workbench 6	
	 ✓ フロクラムタウンロード完了 ✓ プログラム実行完了 リトライ
	← 戻る → 次へ ≥ キャンセル

次へ。





G First Step Guide						
調整中 - セン	サオフセット調整	<u>k</u>				
Workbench 6	ターゲットボードのセン	>サオフセットを	調整し ⁻	ています : 5_TS	S00 (0/14)	
	リファレンスカウンタ値: 15360					
	リ・ノント1道:	18527				
	zット調整中は さい。					
	+	戻る	\rightarrow	次へ	▶ キャンセル	

Workbench6 が最適なオフセット値を算出しますので、しばらくお待ちください。オフセット調整結果が表示されますので、次へ。

·感度調整

タッチキーの、タッチ・非タッチの閾値を決定します。

ボタンの <mark>最大</mark> の変化量を計測します がら、キーボードの"Y"キーを押して	。ハイライト表示されているボタ ください。	タンを鉄製の伝導体などで触れな
ターゲット: 5_TS00 (0/14) 0.7501 1.7503 2.7505 3.7506 4.7500 7501 7503 0.7506 0.7506 7506 7506 7506 7507 7507 7507 7507	タッチ回数: 1 ATS10 7510 7509 TS13 TS13	カウント値: 15400 本画面は、RX130(100 ピン)の表示例です RX130(80 ピン)では、16 キーが使用できま RX113, RX231/RX230 ではこの画面で表示
		ているキーが使用できます

画面に従い、キーに触れた状態(1回目は金属等で触れた場合の最大値、2~4回目は通常の指でのタッチの値を 拾う)で、PC側のキーボードで"y"を押して、タッチ時の値の変化に基づく最適な閾値を算出します。

RX タッチキー評価キット[ソフトウェア編] 株式会社



調整後の表示は、以下の様になります。



ここで赤線(タッチしきい値)に対し、青のグラフ(非タッチ状態の静電容量)までのマージンが極端に少ないキーがあ るようでしたら、再度調整を行ってください。

次へを押すと、調整値がプログラムソースに反映され、プロジェクトがビルドされます。

。First Step Guide ターゲットボー	-ド起動			
Workbench 6	🔗 静電容量タッチャ	2ンサ・プロジェ	クトオープン	
	●ビルド完了			
	✓ プログラムダウン	ロード完了		
2 - 1 - 1 - 1 - 1 - 1	🕑 ブログラム実行学	己了 参考文献 -		
	リトライ	CTSU CTSU	<u>J API リファレンスフ</u> を制御する各種APIを	<u>ヴィド</u> 解説します。
	() [完了]を押してこのウィ	ザードを閉じてく	ださい。	
	← ∄	 ₹る →	完了	≥ キャンセル

完了を押して、再調整は終了です。





2.2. 相互容量タッチキー基板(D55A)

2.2.1. ボードのセットアップ

マイコンボードと、相互容量タッチキー基板(D55A)、LCD 及び E1 を接続。ボードには、電源を供給してください。

WB6/RX*nnn*_TouchAPI_D55A/ (nnn にはマイコン名が入ります) を、PC 内の適当なフォルダにコピーし、2.1.2, 2.1.3 同様に CS+から接続を行ってください。

2.2.2. 感度の確認

TUNING WINDOW を押すと上記画面が開きます。

調整ツールーモニタ開始

で、マイコンボードの動作を PC 上からモニタします。



~ タッチしていると判断されている期間

(1)Status Monitor 画面を開く(アイコンまたは、調整ツール-Status Monitor)

(2)MB00-00の画面を見ながら、タッチキー基板の0(TS00-05)のキーに触れる

赤いラインが、現在の測定値

緑のラインが、タッチ・非タッチを判断する閾値

となっています。相互容量の場合は、自己容量タイプとは逆に、タッチした際に、計測値(容量値)が下がります。 上記画面では、右側に Measurements ウィンドウを表示させており、各種測定値がリアルタイムで表示されます。





プロジェクトを開くところは、2.1.2と同様です。

※チューニング時は LCD がノイズ源となる可能性がありますので、LCD 画面を外した状態で行う事を推奨します。

・静電容量タッチ・インタフェースの定義(D55A 基板を使用する場合変更不要)

・タッチ電極とタッチ端子間の総抵抗の選択(D55A 基板を使用する場合変更不要)

・寄生容量の計測(必要に応じて)

上記3項目は、基板による個体差が大きく出るものではありませんので、キット付属の相互容量タッチキー基板 (D55A)を使用する限りでは、変更の必要はないと考えます。

・センサオフセット調整

マイコン側のパラメータのチューニングを行います。マイコンチップ毎に、オフセットの最適値は異なりますので、本 調整により、動作マージンは向上すると考えます。

🧕 First Step Guide				×
センサオフセッ	ト調整			
Workbench 6	[次へ]を押すとセン	ンサオフセット	調整を開始します	- o
	÷	戻る	→ 次へ	≥ キャンセル

センサオフセット調整を選択し、次へで進みます。

RX タッチキー評価キット[ソフトウェア編] 株式会社



First Step Guide	
調整中 - セン	サオフセット調整
Workbench 6	ターゲットボードのセンサオフセットを調整しています : MB00-00 (0/25)
	プライマリ・リファレンスカウンタ値: 10240 プライマリ・センサカウンタ値: 10524
	Workbench6によるセンサオフセット調整中は タッチセンサに触れないでください。
	← 戻る → 次へ × キャンセル

Workbench6 がセンサのカウンタ値を基に、オフセット値の算出を行いますので、タッチキーボードには触れずにし ばらくお待ちください。調整が終わると調整結果が表示されます、次へで感度調整に進みます。

·感度調整

タッチキーの、タッチ・非タッチの閾値を決定します。

16.7	F	ードの	۲″+-	ーを押してく	ください	۸. ۱	, 43 Y C 4 U	C V 1 Q/1		2× V7 144	
マーゲ	ምሥ:	MB00	-00 ((0/25)		タッチ	回数:	1	カウ	シト値	: 11968
TS00	TS01	TS02	TS03	TS04							
MB00-00	MB00-01	MB00-02	MB00-03	MB00-04 TS05							
MB00-05	MB00-06	MB00-07	MB00-08	MB00-09 TS06							
MB00-10	MB00-11	MB00-12	MB00-13	MB00-14 TS07							
MB00-15	MB00-16	MB00-17	MB00-18	MB00-19 TS08							
MB00-20	MB00-21	MB00-22	MB00-23	MB00-24 TS09							
		Matrix00									





※相互容量タッチキー基板(D55A)基板の配置は下記の様になっており、Workbench6の行列とは配置が異なります

4	9	14	19	24
3	8	13	18	23
2	7	12	17	22
1	6	11	16	21
0	5	10	15	20

	TS00-05	TS01-05	TS02-05	TS03-05	TS04-05	₽=₽≈ ● ₽=₽=
D55A	1500-06	TS01-06	1502–06 7	TS03-06 8	TS04-06	₩ <u>₩</u> ₩ ₩ ₩ ₩
HOKUT	TS00-07	TS01-07	1502-07 12	13111111111111111111111111111111111111	TS04-07	
DENSHI	1500-08	TS01-08 16	TS02-08 17	18 18	TS04-08	<u>Lij</u>
HATC IN JAPAN	TS00-09	TS01- 0 21	TS02-09	1S03-09 23	TS04-09 24	

タッチキー基板を90度右回転した場合Workbench6の表示と一致します。

お手数ですが、Workbench6を使用したチューニング時は、キーの位置を読み替えるか、90度基板を回転させてく ださい。

画面の表示に従い、1つのキーに付き、1回目(金属等で最大の変化量を取得)、2~4回目(通常の指でのタッチ) の合計4回タッチして測定を行ってください。

タッチキーの、タッチ・非タッチの閾値を決定します。







ここで赤線(タッチしきい値)に対し、青のグラフ(非タッチ状態の静電容量)までのマージンが極端に少ないキーがあ るようでしたら、再度調整を行ってください。

次へを押すと、調整値がプログラムソースに反映され、プロジェクトがビルドされます。

G First Step Guide			×					
ターゲットボード起動								
Workbarreh								
静電容量タッチセンサ	ナ・ブロジェク	フトオーブン						
● ビルド完了	● ビルド完了							
✓ プログラムダウンロー	ド完了							
● プログラム実行完了	→ プログラム実行完了							
	参考文献							
リトライ	<u>СТSU</u> стsuź	API リファレンス7 制御する各種APIを	<u> ヴィド</u> 解説します。					
	ドを閉じてくた	ざい。						
← 戻る	→	完了	≥ キャンセル					

完了を押して、再調整は終了です。





RX130、RX231,230 のタッチキー評価キットの場合、シリアルポート経由で、Workbench6 でのモニタリングが可能です。

E1をお持ちの場合は、E1 接続の場合プログラムの転送等を含め一括で行えるので、E1 での接続を推奨致します。E1 を使用できない環境の場合は、本節のシリアルポート経由での接続で、測定値のモニタリングという用途では、E1 接続と同様の事が行えます。



-PCとタッチキー評価キットの接続-

タッチキー評価キットを組み立て(タッチキーボードは、S16A, D55A のどちらでも構いません)、LCD 基板について いる J5(5P コネクタ)に、USB-Serial 機器(北斗電子の製品では USB-1S(JST)、市販の機器の場合 TXD, RXD, GND の 3 ピンを結線してください)経由で、PC と接続してください。

プログラムを予めマイコンボードに書き込んでください。 書き込むプログラムは、プロジェクトフォルダ以下にある、motファイルです。

WB6/RX*nnn*_TouchAPI_S16A/DefaultBuild/TouchApiBase.mot (nnn にはマイコン名が入ります, S16A→マトリックスタイプのキーボードの場合は D55A)





ε続(C) 表示(V) 調整ツール(U) ツー.	ル(T) ウィンドウ(W) へル	_^ プ(H)		
》CS+接続(C)				▼
e2 studio接続(E) シリアルポート接続(S)		PENESAS	W	L.L
 パースト・モニタ接続(B)… 切断(D) 		I (ENES/15	wor	kbench 🔵
全て閉じる(A)				
終了(X)		Tuning Window	User's Manual	Document Manager
* START				

接続ーシリアルポート接続

シリアルポート接続		+3.00 M	CS-AS	×
シリアルポート(<u>S</u>):				
COM5 : USB Seria	al Port (COM5)			-
シリアルポート・ボーレ	·			
38400				-
	ОК	*#>20	L	

シリアルポートは、マイコンボードと接続している、COM ポート番号を選択してください。ボーレートはデフォルトのままで問題ありません。

Workbench6			
接続(<u>C</u>) 表示(⊻) 調整ツ-	-ル(<u>U)</u> ツール(I) ウィンドウ(<u>W</u>)	ヘルプ(<u>H</u>)	
Navigator			• >
		Renesas	Workbench 6
First Step STA	Guide RT	Tuning Window U	ser's Manual Document Manager
シリアルボート接続 8 ログ	停止 🔤 モニタ停止		

RX タッチキー評価キット[ソフトウェア編] 株式会社



左下の表示が「シリアルポート接続」に変われば、接続は成功しています。

- 0 **- X** Workbenche 接続(C) 表示(V) 調整ツール(U) ツール(T) ウィンドウ(W) ヘルプ(H) A 1500 - TS: 00 Unit: 0 × Multi Mo - വ്വ്വ് 🕯 5 TS00 - TS: 00 Unit: 0 ₽ × Measurements Ψ× Measurements 5-XJU 5_T500 (00 - 01) 0_T501 (00 - 01) 6_T502 (00 - 02) 1_T503 (00 - 03) 7_T504 (00 - 04) 2_T505 (00 - 05) 8_T507 (00 - 07) 4_T508 (00 - 08) 9_T509 (00 - 09) A_T513 (00 - 13) C_T514 (00 - 14) D_T515 (00 - 15) カウント値 プライマリ セカンダリ 基準値 差分値 しきい値 20514 15093 15545 15397 15579 15546 15578 15446 15585 15446 15585 15503 15260 15260 15260 15375 14946 117 201 72 132 69 51 72 24 61 34 31 13 23 14976 15344 15325 15447 15477 15527 15374 15561 15442 15266 15229 15388 14969 3109 3054 2469 3049 2511 2551 2989 2619 3003 2643 3098 2761 3313 15 12 12 12 14 13 15 13 15 13 6_TS02: 15534 800 7200 2_TS05: 15541 3_TS06: 15567 プライマリ・カウンタ: 00000 セカンダリ・カウンタ: 00000 Board Monit 差分値: 05425 エラーステータス タッチIF定義 ボタン:14 マトリックスボタン:0 スライダ:0 ホイール:0 0 オーバフロー: TSCAP電圧異常: ターゲットボード 2 TS0 TS0 2 TSC TS1 TS1-5 TS0 _TS13 🛓 シリアルボート接続 👫 ログ停止 📷 モニタ中

Tuning Window のボタンを押し、ウィンドを開いてください。

調整ツールー接続

で、モニタを開始してください。

E1 接続時と同様、タッチキーのモニタが可能です。展示会等、プログラムの変更(閾値のチューニング)、再ダウン ロードを行う必要がないケースでは、シリアルでの接続も、ご利用頂けます。

Radia Maria State State



3. サンプルプログラム

CD内には、Workbench6を使用していないサンプルプログラムも格納されています。サンプルプログラムは、タッチ キーの動作に最低限必要な構文のみで構成されています。実際にタッチキーを製品等に組み込み場合は、時間経過 による特性変動の影響を補正する処理が行われている Workbench6 ベースで開発を行う事を推奨致します。動作中 にレジスタ値(測定値)がどのように変化するか等の単純に動作を確認する場合は、複雑な処理を行っていない本サ ンプルプログラムを参考にしてください。

3.1. サンプルプログラムで使用している定義値・変数等の説明

プログラムで共通で使用している変数について。

(1)#define 定義変数

CTSU_CH_NUM

測定チャネル数。自己容量タイプではキーの数を定義。

相互容量タイプでは、キーが 5x5 の 25 キーとなるが、1 キーで同相と逆相の 2 回測定するので、"50"を定義しています。

CTSU_KEY_NUM

相互容量のキーの数。5x5の"25"を定義しています。

CTSU_COLUMN_NUM

相互容量の列数。相互容量キーパッドでは5列となっているので"5"を定義しています。

CTSU_CTSUSNUM

|測定回数。サンプルでは、1 回としているので、"0"を定義している。マイコンレジスタの CTSUSNUM[5:0]に対応。

CTSU_CTSUSO_TS_x [x=0~CTSU_CH_NUM]

センサ ICO 入力電流オフセット値。0-1023 の値、初期値は"0"としています。デバイスばらつきにより最適値は異なり、タッチキー初期化ルーチン内で最適なオフセット値を算出すします。マイコンレジスタの CTSUSO[9:0]に対応。

CTSU_CTSUICOG

ゲイン設定値。"0"でゲイン 100%。マイコンレジスタの CTSUICOG[1:0]に対応。

CTSU_CTSUSDPA

ベースクロック分周比。"0"で、2分周となる。マイコンレジスタの CTSUSDPA[4:0]に対応。

CTSU_CTSURICOA_TS_ x [x=0~CTSU_CH_NUM]

リファレンス ICO 電流調整。オフセット電流調整用のリファレンス ICO に与える電流値を決める。初期値は"63"としている。マイコンレジスタの CTSURICOA[7:0]に対応。

RX タッチキー評価キット[ソフトウェア編] 株式会社 **北井電子**



CTSU_CTSUSO_TS_SEARCH_DELTA

入力電流オフセットの最適値を検索する刻み。初期値は10としています。

(2)グローバル変数

unsigned short sens[CTSU_CH_NUM]

測定値格納変数。キーの容量によって変わる測定値を格納しています。

unsigned short ref[CTSU_CH_NUM], initial_ref[CTSU_CH_NUM]

リファレンス測定値格納変数。

long initial_sens[CTSU_CH_NUM]

初期値格納変数。プログラム起動時に、キーに触っていないときの初期値を拾い、後にキーに触れた場合との差分を見るために使用。

unsigned short base_diff[CTSU_KEY_NUM];

相互容量タイプで、2回の測定値の差分の初期値(パッドに触れていないときの値)

unsigned short diff[CTSU_KEY_NUM];

相互容量タイプで、2回の測定値の差分

unsigned char status[CTSU_CH_NUM]

測定時のステータス格納変数。

unsigned short ctsu_ctsuso0[CTSU_CH_NUM]

マイコンレジスタ CTSUSOO 設定変数。

unsigned short ctsu_ctsuso1[CTSU_CH_NUM]

マイコンレジスタ CTSUSO1 設定変数。

unsigned short data_index

測定チャネルを示す変数。

unsigned short err

測定エラーを格納する変数。マイコンレジスタ CTSUERRS の結果が格納される。

unsigned char key

そのキーが押されているかを示す変数。

unsigned char in_measure, result_update, initialize_finished

フラグ変数。

22





4. 自己容量キー読み出しサンプル

4.1. ソースツリー構成

[MCU_NAME]_Self_cap_S16A_sample/ 以下

cg_src	r_cg_main.c	メイン関数
ctsu	ctsu.h	CH 数等の定義ファイル
	ctsu_selfcap.c	自己容量キー読み出し
	ctsu_intr.h	割込みプログラムヘッダ
	ctsu_intr.c	割込みプログラム
	ctsu_offset_value.h	電流オフセット値定義ファイル

ソースで主要なものを上表に示す。

4.2. 自己容量キーパッド



自己容量タイプはキーパッドが合計 16 個並んでおり、それぞれ、マイコンの TS0~TS15 に接続されています。検 出は、パッドに手で触れた際、それぞれの PAD に見える容量値が増加することから、どのパッドに触れたかを判断す る方式です。





組み合わせるマイコンボードにより、使用可能なキーは変わります。RX130(80 ピン)では、16 キー全て使用可能で す。その他のマイコンでは、一部のキーが使用できません。

キーパッド	接続ノード	RX113	RX130 (80ピン)	RX130 (100ピン)	RX231/ RX230
0	TS01	0	0	0	0
1	TS03	0	0	0	0
2	TS05	0	0	0	0
3	TS06	0	0	0	0
4	TS08	0	0	0	0
5	TS00	0	0	0	0
6	TS02	0	0	0	0
7	TS04	0	0	0	0
8	TS07	0	0	0	0
9	TS09	0	0	0	0
A	TS10	0	0	0	—
В	TS13	—	0	—	0
С	TS14	—	0	0	_
D	TS15	_	0	0	_
E	TS12	_	0	_	0
F	TS11	0	0	O[TS13]	_

※RX130(100 ピン)と S16A キーボードとの接続の場合、F のキーが TS13 となります

RX タッチキー評価キット[ソフトウェア編] 株式会社 **北手電子**



4.3. 関数

(1)ctsu_selfcap.c 内で定義

void ctsu_init(void)

引数:なし 戻り値:なし

タッチセンサ初期化。各種マイコンレジスタの設定。

void ctsu_initial_ref(void)

引数:なし

戻り値:なし

センサ測定値の初期値(仮値)を求める関数。キーにタッチしていないときに実行する必要がある。 キーの測定を256回行い、測定値の平均をinitial_ref[]変数に格納する。

void ctsu_offset_value(void)

引数:なし 戻り値:なし

オフセット値を求める関数。キーにタッチしていないときに実行する必要がある。

void ctsu_initial_value(void)

引数:なし

戻り値:なし

センサ測定値の初期値(オフセット電流調整後)を求める関数。キーにタッチしていないときに実行する必要がある。 キーの測定を256回行い、測定値の平均をinitial_sens[]変数に格納する。



void ctsu_set_param(void)

引数:なし

戻り値:なし

マイコンのCTSUSSC, CTSUSO0, CTSUSO1レジスタにそれぞれ、0, ctsu_ctsuso0[],ctsu_ctsuso1[]に格納されている値を転送する。

void ctsu_measure(void)

引数:なし 戻り値:なし

測定を開始する。

void ctsu_read(void)

引数:なし 戻り値:なし

センサ測定結果をsens[]、リファレンス測定値をref[]、測定ステータスをstatus[]、エラーをerr変数に格納する。

unsigned char ctsu_result(void)

引数:なし 戻り値:押されているキー番号

(2)ctsu_intr.c 内で定義

void int_ctsuwr(void)

引数:なし 戻り値:なし

チャネル測定前に入る割込み関数。

void int_ctsurd(void)

引数:なし 戻り値:なし

チャネル測定後に入る割込み関数。



RX タッチキー評価キット[ソフトウェア編]

HALA



void int_ctsufn(void)

引数:なし 戻り値:なし

全チャネルの測定終了時に入る割込み関数。



27



4.4. プログラム説明

4.4.1. フローチャート

・全体フロー



タッチセンサの測定値は、パッドの容量によって変化します。パッド毎にパッドに触れていないときの初期容量値は 微妙に異なるため、「タッチセンサ初期測定値取得」時に、パッドに触れていない時の測定値(256 回測定の平均値) を取得します。

「タッチセンサ測定」時の測定値を見て、

・一定の変化量があるか(本プログラムでは、20%以上値が増加したか)

※判定閾値を変える場合は、 ctsu_selfcap.c 内の const float diff_threshf=1.2;//20%変化以内を閾値とする 変数の値を変更してください

・一番変化の大きなキーはどれか(タッチの位置が微妙にずれている場合触れているパッド以外の測定値も増加するため、一番大きく変化した測定値のキーが押されていると判断する)

という基準で触れているパッドを判定する。





・タッチセンサオフセット最適化



電流オフセット値の調整は、マイコンの電流オフセット CTSUSO0 レジスタを加算(本プログラムでは 10 刻み)して、リファレンス値を越えた際、リファレンス値を超える 1 つ前の値を、調整されたオフセット値として保存する。

電流オフセット値は、マイコンチップ毎(チップの製造ばらつきに依存)に最適値が異なる値で、マイコンチップ毎に最 適値を求める必要があります。

プログラム実行中にエラー(err=0x8000)となった場合、電流オフセットの設定値が大きすぎてオーバフローしていますので、タッチキーパッドとの接続が正しいか等確認してください。





4.4.2. 実際の測定値の変化例

initial_sens	sens(1)	sens(2)
1212	1234	1222
1611	1528	1626
1562	1529	1617
1458	1450	1428
1712	1749	2819
1531	1532	1608
1664	1622	1773
1382	1430	1484
1746	1711	1743
1348	1315	1374
1537	1491	1516
1526	1546	1652
1686	1667	1728
1402	1338	1456
1609	1545	1551
1700	1688	1670
	initial_sens 1212 1611 1562 1458 1712 1531 1664 1382 1746 1348 1537 1526 1686 1402 1609 1700	initial_sens sens(1) 1212 1234 1611 1528 1562 1529 1458 1450 1712 1749 1531 1532 1664 1622 1382 1430 1746 1711 1348 1315 1537 1491 1526 1546 1686 1667 1402 1338 1609 1545 1700 1688

initial_sens が初期に取得した値。sens(1)がパッドに触れていないときの測定値。sens(2)が TS4 のパッドに触れたときの測定値。パッドに触れた際測定値が大きく変化するため、プログラムでの判別は容易である。

Radia Maria State State



5. 相互容量キー読み出しサンプル

5.1. ソースツリー構成

[MCU_NAME]_Mutual_cap_D55A_sample/ 以下

コード生成	r_cg_main.c	メイン関数
ctsu	ctsu.h	CH 数等の定義ファイル
	ctsu_mutual.c	相互容量キー読み出し
	ctsu_intr.h	割込みプログラムヘッダ
	ctsu_intr.c	割込みプログラム
	ctsu_offset_value.h	電流オフセット値定義ファイル

ソースで主要なものを上表に示す。

5.2. 相互容量キーパッド



相互容量キーパッドは、5x5のマトリックスで構成されており、行側がマイコン TS0~TS4 端子、列側が TS5~TS9 に 接続されている。プログラムでは、TS0~TS4 を出力、TS5~TS9 を入力に設定する必要がある。(なお、相互容量測 定モードでは、入力に設定した TS5~TS9 端子にも測定時に信号が出力される)



5.3. 関数

関数の構成に関しては、基本的に自己容量タイプと同一構成となっている。相違のあるもののみ示す。

(1)ctsu_mutualcap.c 内で定義

void ctsu_row_column(unsigned char num, unsigned char *result)

引数:

num diffのインデックス値

result インデックス値をrow(列番号), column(行番号)に変換した値(関数内で書き換え) 戻り値:なし

キーのインデックス値を、列番号、行番号に変換する関数。

5.4. プログラム説明

5.4.1. フローチャート

全体フローチャートは 2.4 自己容量タイプと同一である。

5.4.2. 実際の測定値の変化例

	初期値	sens(1)	sens(2)
diff_0	7178	7208	6949
diff_1	7671	7560	7275
diff_2	8003	8116	7962
diff_3	9576	9435	9577
diff_4	10240	10194	10091
diff_5	6627	6618	6249
diff_6	7351	7378	5963
diff_7	7490	7451	7209
diff_24	7194	7240	7199

測定は 1 キーパッド 2 回(同相、逆相)ずつ、合計 48 回行われる。上記表は、同相-逆相の差分の値(パッドの容 量値相当の値)である。手でキーにタッチすると、信号線間の容量値が減少するため、初期値に対して値が減少する。

初期値が初期に取得した値。sens(1)がパッドに触れていないときの測定値。sens(2)がパッド 6(TS1-TS6 交点)に 触れたときの測定値となっている。

ここで、diff_6に触れた sens(2)の、測定値が他の値に関して減少している事が見て取れる。

この値の減少幅が一番大きなキーをプログラムで拾い、LCD 画面に表示している。(全てのキーの減少幅が 5%未 満の場合は、どのキーもタッチされていないと判断)





測定値の減少率を見て、減少率が一定以内のときはタッチしていないとみなします。サンプルでは、5%減の値を設 定しています。

※判定閾値を変える場合は、 ctsu_mutualcap.c 内の const float diff_threshf=0.95;//95%を閾値とする 変数の値を変更してください





6.1. ソースツリー構成

[MCU_NAME]_Self_cap_S16A_sample/ 以下

以下 …自己容量タイプアプリケーションサンプル

コード生成	r_cg_main.c	メイン関数
	r_cg_rtc_user.c	割込みでキー読み込みを処理するメインルーチン
ctsu	ctsu.h	CH 数等の定義ファイル
	ctsu_selfcap.c	測定プログラム
	ctsu_intr.h	割込みプログラムヘッダ
	ctsu_intr.c	割込みプログラム
	ctsu_offset_value.h	電流オフセット値定義ファイル

ソースで主要なものを上表に示す。

6.2. 関数

関数定義で初出のものを示す。

(1) r_cg_rtc_user.c 内で定義

void r_rtc_prd_interrupt (void)

引数:なし

戻り値:なし

タイマ割込み(1/64秒毎)に呼び出され、タッチキーの測定及びメインの処理を行う。



6.3. アプリケーション動作

加算を行う電卓のアプリケーションとなっている。 自己容量キーパッドを接続し、以下のキーで操作を行う。 0-9:0-9の数値入力 A, B: 加算入力(入力数値を確定し、加算を行う) E, F: オールクリア

※RX113では、加算入力にAを、オールクリアにFを使用してください ※RX231/230では、加算入力にBを、オールクリアにEを使用してください ※RX130(80ピン)では、A及びBを加算入力に、E及びFをオールクリアに使用してください ※RX130(100ピン)では、Aを加算入力に、Fをオールクリアに使用してください

[RX113] セグメントLCD

加算入力を押した時点で、入力中の数値を確定させ、入力値と前回までの合計値を LCD に表示します。 表示で最上位の'(シングルクオート)が点滅した場合は、計算結果がオーバフローした事を示す。 オールクリアで、合計値を0にします。

[RX130/RX231/230] キャラクタLCD

2 行表示で

1 行目:input> [入力中の数値を表示]

2 行目:sum > [合計値を表示]

となります。

加算入力を押した時点で、入力中の数値を確定させ、入力値と前回までの合計値を2行目に表示します。 オールクリアで、合計値を0にします

※扱える数値は、0~2³¹-1 まで(unsigned long)の範囲です





6.3.1. フローチャート







7. LCD 表示サンプルプログラム[セグメント LCD 向け]

本プログラムは、「RX113 タッチキー評価キット R」に含まれる、セグメントタイプ LCD 向けのプログラムです。 「RX130(-100)タッチキー評価キット」及び「RX231/230 タッチキー評価キット」には含まれません。

7.1. ソースツリー構成

lcd	lcd.c	ソースファイル
	lcd.h	ヘッダファイル

7.2. 関数

void lcd_init (void)

引数:なし

戻り値:なし

LCDの初期化を行う。

unsigned char lcd_ctrl(unsigned char column, unsigned char chara, unsigned char flag)

引数:

column:表示する桁を指定 1-8(1:最上位桁,8:最下位桁) chara: 表示するセグメントを指定 'A'-'N'のセグメントと'q'(':クオート), 'd'(.:ドット) flag: フラグを指定 0:指定されたセグメントを消灯, 1:点灯, 2:点滅

戻り値:

- 0:引数OK
- 1: 引数NG







DIGIT 1- 8

LCDのセグメント単位での、点灯、消灯、点滅を制御する。

使用例:

Icd_ctrl(1, 'A', 1);

lcd_ctrl(1, 'B', 1);

Icd_ctrl(1, 'C', 1);

1 桁目(左詰)の、A, B, C のセグメントを点灯。(数値の7を表示)

unsigned char lcd_clear_disp(unsigned char flag)

引数:

flag: フラグを指定 0:全セグメントを消灯, 1:点灯, 2:点滅

戻り値:

0:引数OK

1:引数 NG

LCD のクリアを行う。

使用例:

lcd_clear_disp(0); //LCD 消灯 lcd_clear_disp(1); //LCD 全点灯





unsigned char lcd_disp_val(unsigned char column, unsigned char num)

引数:

column: 表示する桁を指定 1-8(1:最上位桁, 8:最下位桁)

num: 表示する文字を指定

0-9:0~9を表示(引数は1桁)

'A', 'B', 'C', 'D', 'E', 'F': A~Fを表示(B, Dは、表示上は小文字での表示)

- '.', 'd': . を表示
- '", 'q': ' を表示
- '+':+ を表示
- '-':-を表示
- '*':* を表示
- '/' 's': / を表示

'b' '¥': ¥(バックスラッシュ)を表示

- 'x':x を表示
- '<':< を表示
- '>':> を表示
- ||:| を表示
- 戻り値:
 - 0:引数OK
 - 1: 引数NG

1桁の数値、記号を表示する。

使用例:

lcd_disp_val(1,5); //1 桁目(左詰)に5を表示する lcd_disp_val(1, '+'); //1 桁目(左詰)に+を表示する





unsigned char lcd_disp(unsigned long num, unsigned char fillzero)

引数:

num: 表示する数値(符号なし) fillzero: 数値の0埋めフラグ 0:0埋めを行わない, 1:先頭の0埋めを行う

戻り値:

- 0:引数OK
- 1:引数NG

数値を表示する。

使用例:

lcd_disp(123,0) //数値 123 を表示



8. LCD 表示サンプルプログラム[キャラクタ LCD 向け]

「RX130(-100)タッチキー評価キット」及び「RX231/230 タッチキー評価キット」に付属のキャラクタ LCD(SC1602)の 表示方法を示します。

「RX113 タッチキー評価キット R」付属 LCD ボードの表示に関しては、「LCD 表示サンプルプログラム」の章を参照 ください。

「RX130(-100)タッチキー評価キット」及び「RX231/230 タッチキー評価キット」に付属のキャラクタ LCD は、変換基板を介して接続されます。本マニュアルでは、ソフトウェアの部分に関して解説しています。

8.1. ソースツリー構成

sc1602_lcd	lcd_1602.c	ソースファイル
	lcd_1602.h	ヘッダファイル
	lcd_1602_[BOARD].h	ヘッダファイル(ボード固有定義)

[BOARD]には、ボード名が入ります。

8.2. 関数

void lcd_init (void)

引数:なし 戻り値:なし

LCDの初期化を行う。

void lcd_cmd(unsigned char c)

引数:

c:LCDに送るコマンド

戻り値:なし

LCDにコマンドを送信する

使用例:

lcd_cmd(0x01);

LCD画面の全消去コマンドを送る





void lcd_hs1(void)

引数:なし 戻り値:なし

1行目にカーソルを移動する

void lcd_hs2(void)

引数:なし 戻り値:なし

2行目にカーソルを移動する

void lcd_write_char(unsigned char c)

引数:

c :LCDに表示する文字

戻り値:なし

LCDに1文字表示させる

void lcd_write_hex(unsigned char c)

引数:

c:LCDに表示する数値

戻り値:なし

LCDに、与えられた引数を、16進数2桁(00~FF)で表示する。

void lcd_write_byte_int(unsigned char num)

引数:

num :LCDに表示する数値(0~255)

戻り値:なし

LCDに、与えられた引数を、10進数(0~255)で表示する。

42





void lcd_write_short_int(unsigned short num)

引数:

num :LCDに表示する数値(0~65535)

戻り値:なし

LCDに、与えられた引数を、10進数(0~65535)で表示する。

void lcd_write_str(unsigned char *str)

引数:

*str:LCDに表示する文字列('¥0'終端)

戻り値:なし

LCDに、与えられた文字列を表示する。

8.3. 操作例

lcd_hs1(); lcd_write_str("HSBRX130F80-S16A");

1 行目に"HSBRX130F80-S16A"を表示

HSBRX130F80-S16A

lcd_hs2(); lcd_write_str(">"); lcd_write_short_int(12345);

2 行目に">[数値]"を表示

HSBRX130F80-S16A >12345







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	2016.11.17	—	初版発行
REV.1.1.0.0	2018.4.9	P2,3,24,35,37,41 P4-17 P18-20	RX130-100ピン向けに追加、修正 Workbench6(v1.07)の画面に差し替え シリアル接続の手順を追記

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} 北手電子

〒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

商標等の表記について

- ・ 全ての商標及び登録商標はそれぞれの所有者に帰属します。
- ・ パーソナルコンピュータを PC と称します。



ルネサス エレクトロニクス RX グループマイコン搭載 HSB シリーズマイコンボード

RX113 タッチキー評価キット R RX130 タッチキー評価キット RX130-100 タッチキー評価キット RX231 タッチキー評価キット RX230 タッチキー評価キット [ソフトウェア編] マニュアル

©2016-2018 北斗電子 Printed in Japan 2018 年 4 月 9 日改訂 REV.1.1.0.0 (180409)