EPSON
EXCEED YOUR VISION

S1C17801
RTC アプリケーションノート

ダウンロードした圧縮ファイルに同梱されているご注意.pdf に従って使用してください。

セイコーエプソン株式会社

Rev.1.0

本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告無く変更することがあります。

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。

2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これら起因する第三者の権利（工業所有権を含む）侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の工業所有権の実施権の許諾を行うものではありません。

3. 特性値の数値の大小は、数直線上の大小関係で表しています。

4. 本資料に掲載されている製品のうち「外国為替及び外国貿易法」に定める戦略物資に該当するものについては、輸出する場合、同法に基づく輸出許可が必要です。

5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。

©SEIKO EPSON CORPORATION 2008, All rights reserved.

要旨

本資料は S1C17801 の RTC 機能を使用するための参考資料です。

動作環境

・S5U1C17801T1100（以下 SVT17801:Software eValuation Tool for S1C17801）
SVT17801 CPU BOARD、SVT17 ICD BOARD のセット（必要に応じて 2 セット必要）
・USB miniB ケーブル
・パソコン
GNU17 開発ツールインストール済み
SVT17 ICD BOARD の USB ドライバインストール済み

目　次

1. 仕様 1
2. 使用機能説明 2
3. ソフトウェア説明 3
3.1 ファイル構成 3
3.2 モジュール説明 3
3.3 グローバル変数 3
3.4 構造体 4
3.5 操作手順 5
3.6 サンプルプログラム動作概要 5
3.7 フローチャート 6
3.8 RTC ドライバ詳細説明 9
3.9 ヘッダの定義 21
改定履歴 22

1. 仕様

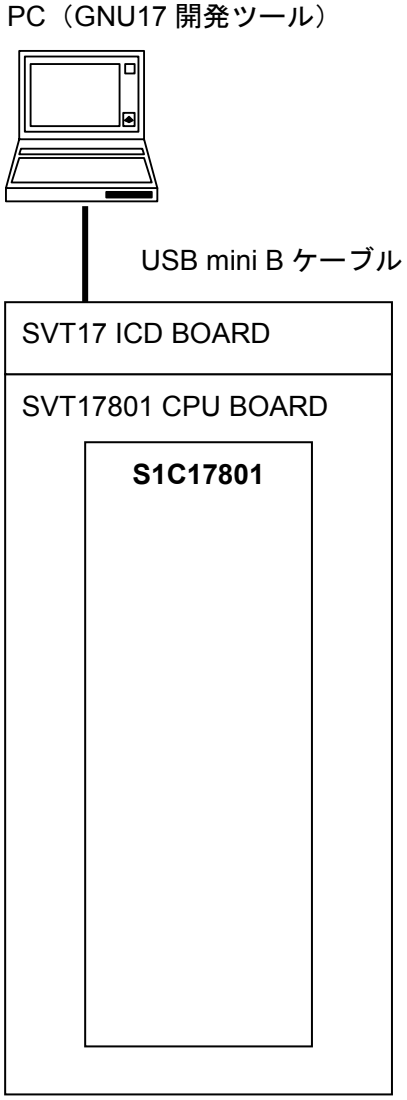
RTC から時刻を取得し表示、指定された日時を RTC へ設定を行います。
また、割込み設定がされ割り込み処理関数が呼び出されているのか確認することができます。

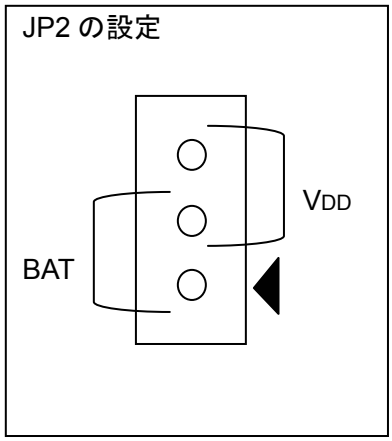
Simulated I/O の「command >」プロンプトから、以下のコマンド入力を行います。

コマンド一覧

コマンド	処理内容
get	RTC の時刻取得
set	RTC への時刻設定
int	割込み発生回数表示

以下に接続方法を記します。



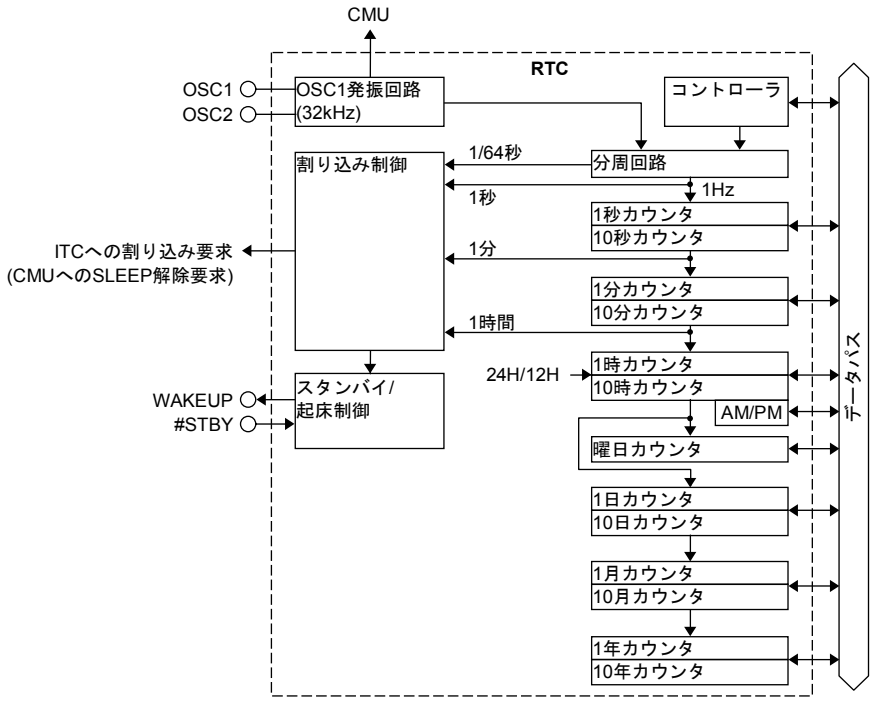


VDD：SVT ボードの電源で RTC が動作
BAT：BAT1 に実装される電池で RTC/IVRAM が動作

2. 使用機能説明

入出力端子　　汎用入出力ポート端子（P43）と#STBY が接続されています。
#STBY 信号が High 入力となるように設定する必要があります。High 入力される事で、CPU から RTC/IVRAM へのアクセスが可能となります。

割り込み　　RTC 割り込みのベクタ番号とベクタアドレスは以下のとおりです。
ベクタ番号：　　20(0x14)
ベクタアドレス：　　0x900050



3. ソフトウェア説明

3.1 ファイル構成

ファイル名	機能
boot.c	スタートアップモジュール
main.c	メイン関数
inthdlr.c	割込み関数
vector.c	ベクターテーブル設定
header/reg_801.h	レジスタ定義
header/vector.h	ベクターテーブル定義
rtc_drv/rtc_drv.c	RTC ドライバ API 群
rtc_drv/rtc_api.h	RTC ドライバ API 定義
cmu_drv/	CMU ドライバ群フォルダ
flashc_drv/	FLASHC ドライバ群フォルダ
gpio_drv/	GPIO ドライバ群フォルダ
itc_drv/	ITC ドライバ群フォルダ
sramc_drv/	SRAMC ドライバ群フォルダ
rtc_gnu17IDE.lds	リンカスクリプトファイル
rtc_gnu17IDE.cmd	GDB コマンドファイル
rtc_gnu17IDE.par	パラメータ設定ファイル
rtc_gnu17IDE.mak	Make ファイル
.cdtproject	プロジェクトファイル
.gnu17project	プロジェクトファイル
.project	プロジェクトファイル
GDB17 Launch for svt17801_rtc.launch	プロジェクト起動ファイル

3.2 モジュール説明

ファイル名：main.c

関数名	機能
main	入出力ポートの設定、ITC への設定、割込みハンドラ登録、Simulated I/O への表示を行う。
rtcInterrupt	RTC 割込み処理関数 本関数が Call された際、Counter の加算を行う。

ITC ドライバ関数については“ITC ドライバ詳細説明”の章を参照してください。
GPIO ドライバ関数については“GPIO ドライバ詳細説明”の章を参照してください。

3.3 グローバル変数

サンプルプログラム内で使用しているグローバル変数について記します。

変数名	型	機能
g_String[50]	unsigned char	時刻読み出し表示用 Buffer
g_RtcCnt	unsigned short	割込み発生回数
inbuff[128]	unsigned char	Simulated I/O 入力用
outbuff[128]	unsigned char	Simulated I/O 出力用

3.4 構造体

サンプルプログラム内で使用している構造体について記します。

定義名		
T_RTC_CFG		
メンバ		
intEnable	unsigned char	RTC 割込み Enable/Disable
intMode	unsigned char	RTC 割込み Edge/Level
intCycle	unsigned char	RTC 割込み発生 Cycle
wakeupPolarity	unsigned char	Wakeup 信号極性
wakeupControl	unsigned char	Wakeup 信号 Active/Inactive
intFunc	void (*)()	Portx5 の割込み処理関数ポインタ
備考		
RTC の初期値設定用構造体。		

定義名		
T_RTC_DATE_INFO		
メンバ		
second	unsigned char	秒の値(BCD 形式)
minute	unsigned char	分の値(BCD 形式)
hour	unsigned char	時の値(BCD 形式)
indicator	unsigned char	午前/午後
day	unsigned char	日の値(BCD 形式)
month	unsigned char	月の値(BCD 形式)
year	unsigned char	年の値(BCD 形式)
week	unsigned char	曜日(0:日,1:月,2:火,3:水,4:木,5:金,6:土)
hourMode	unsigned char	12 時間制/24 時間制
備考		
RTC の日時用構造体。		

3.5 操作手順

プロジェクトのインポート
(1) IDE を起動して、「rtc」プロジェクトをインポートしてください。
※インポートの方法は S5U1C17001C Manual “3.ソフトウェア開発手順”を参照してください。

ビルド
(1) IDE を使用して「rtc」プロジェクトをビルドしてください。

接続、電源投入
(1) SVT17801、USB miniB ケーブル、PC を接続してください。
(2) SVT17 ICD BOARD をリセットしてください。

実行
(1) IDE を使用して「rtc」プロジェクトを実行してください。
(2)「rtc」の実行後、Simulated I/O に「command >」プロンプトが表示されます。
(3) 以下のコマンドの何れかを入力して下さい。

コマンド一覧

コマンド	処理内容
get	RTC の時刻取得
set	RTC への時刻設定
int	割込み発生回数表示

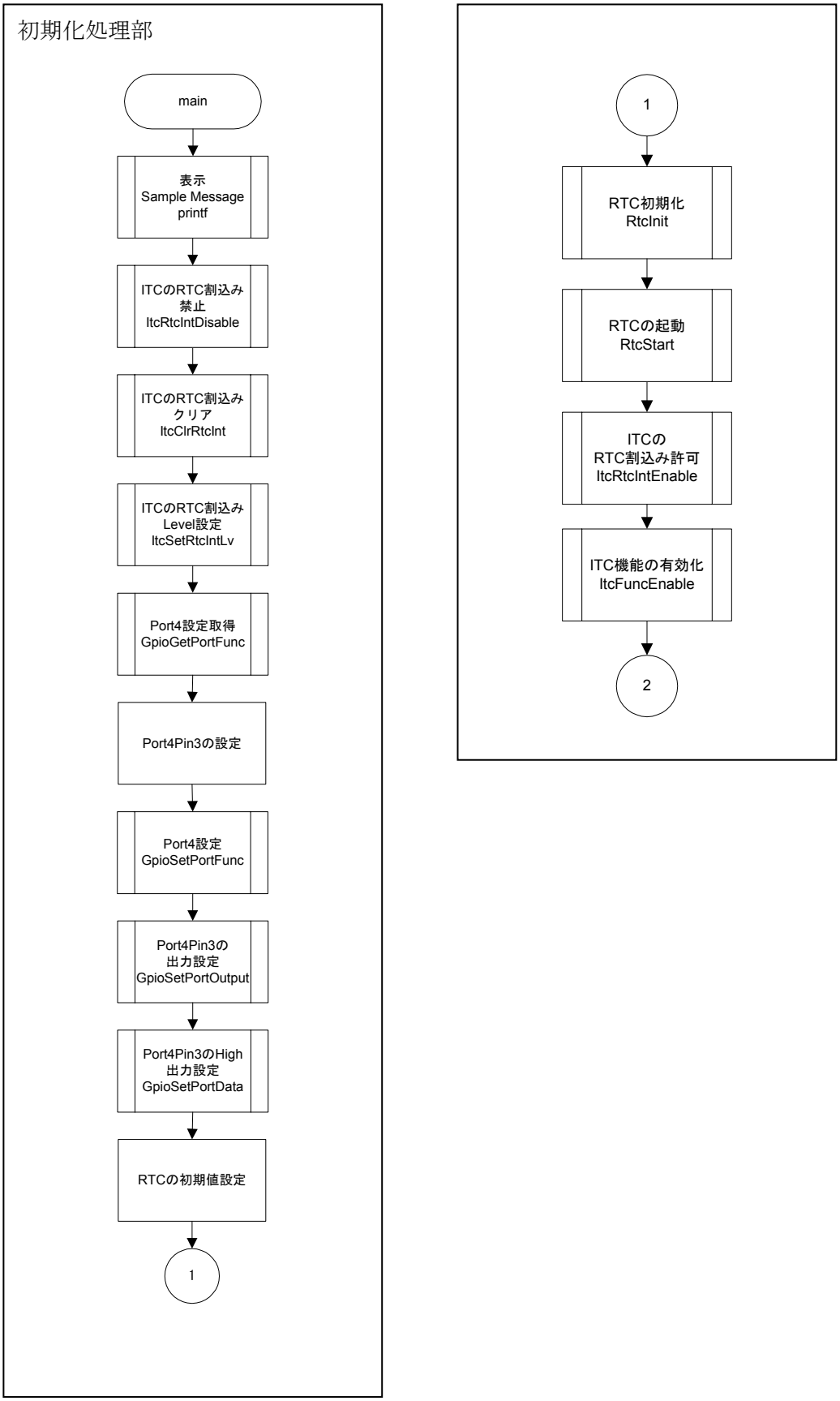
「get」コマンドを入力した場合、現在の RTC の時刻が Simulated I/O に表示されます。
「set」コマンドを入力した場合、Simulated I/O の表示に従って、年、月、日、12/24 時間制、時、分、秒、曜日を入力して下さい。入力した日時が RTC に設定されます。
「int」コマンドを入力した場合、RTC の割込みが発生した回数を Simulated I/O に表示します。

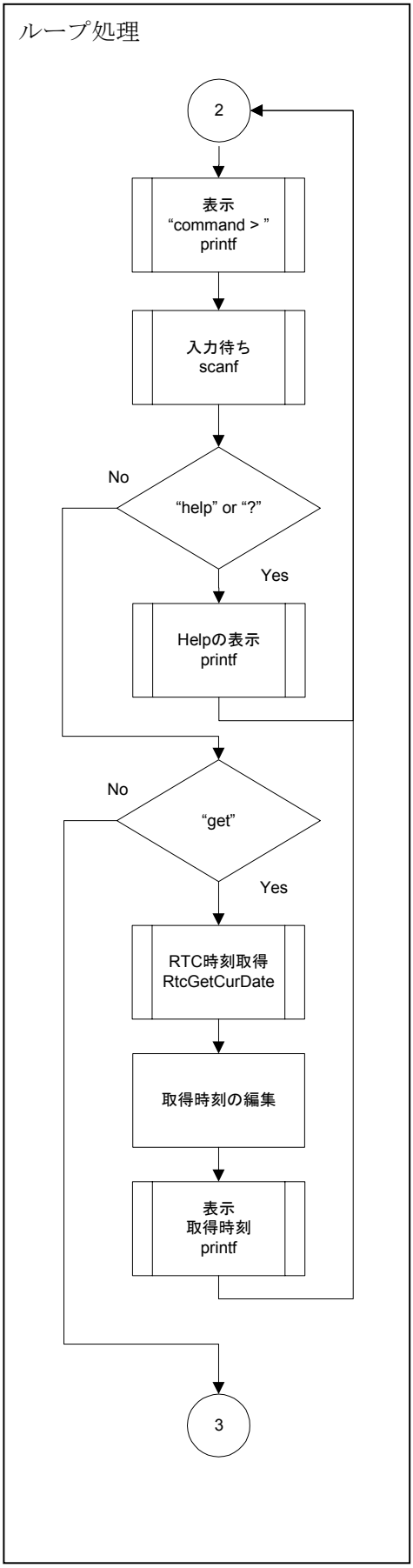
3.6 サンプルプログラム動作概要

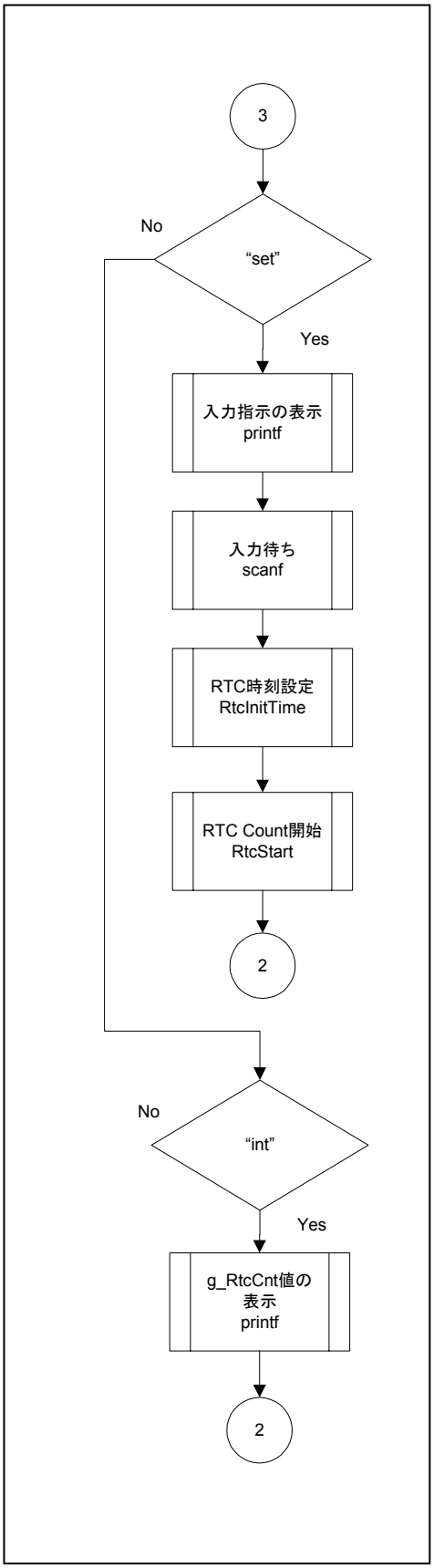
(1) ITC の RTC 割込みの設定を行います。
(2) ポート P43 の出力設定と High 出力を行います。
(2) RTC の割込み設定、割込み処理関数の登録、Wakeup 信号の設定を行います。
(3) RTC を起動します。
(4) ITC の割込みを許可します。
(5) Simulated I/O からの入力コマンドの動作を行います。

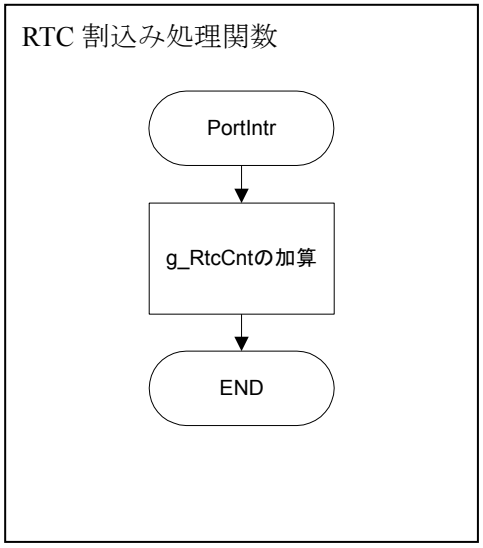
3.7 フローチャート

メインルーチンおよび割込み関数のフローチャートを下記に記します。




ループ処理
2
表示
“command > ”
printf
入力待ち
scanf
No
“help” or “?”
Yes
Helpの表示
printf
No
“get”
Yes
RTC時刻取得
RtcGetCurDate
取得時刻の編集
表示
取得時刻
printf
3


3
No
“set”
Yes
入力指示の表示
printf
入力待ち
scanf
RTC時刻設定
RtcInitTime
RTC Count開始
RtcStart
2
No
“int”
Yes
g_RtcCnt値の
表示
printf
2


RTC 割込み処理関数
PortIntr
g_RtcCntの加算
END

3.8 RTC ドライバ詳細説明

ファイル名 rtc_drv.c と rtc_api.h に記述された関数について記します。

RTC 初期化設定

書式	void RtcInit(T_RTC_CFG *pConfig)		
機能	RTC 初期化設定		
引数	pConfig	-in	RTC 初期化 Parameter
戻り値	なし		

(処理内容)
RTC の初期化を行います。

```
typedef struct {
        unsigned char       intEnable;          // 割込み Enable/Disable
                                                // RTC_INT_DISABLE
                                                // RTC_INT_ENABLE
        unsigned char       intMode;            // 割込みモード
                                                // RTC_INT_MODE_EDGE
                                                // RTC_INT_MODE_LEVEL
        unsigned char       intCycle;           // 割込み間隔
                                                // RTC_INT_CYCLE_1_64S
                                                // RTC_INT_CYCLE_SECOND
                                                // RTC_INT_CYCLE_MINUTE
                                                // RTC_INT_CYCLE_HOUR
        unsigned char       wakeupPolarity;     // Wakeup 極性
                                                // RTC_WAKEUP_HIGH
                                                // RTC_WAKEUP_LOW
        unsigned char       wakeupControl;      // Wakeup 制御
                                                // RTC_WAKEUP_INACTIVE
                                                // RTC_WAKEUP_ACTIVE
        void                (*intFunc)(void);   // 割込み処理関数ポインタ

} T_RTC_CFG;
```

RTC 時刻設定

書式	void RtcInitTime(T_RTC_DATE_INFO *pDate)
機能	RTC 時刻設定
引数	pDate -in RTC 時刻設定 Parameter
戻り値	なし

(処理内容)
RTC の時刻設定を行います。

```
typedef struct {
        unsigned char   second;     // 秒(0x48 = 48 秒)BCD 形式
        unsigned char   minute;     // 分(0x48 = 48 分)BCD 形式
        unsigned char   hour;       // 時(0x12 = 12 時)BCD 形式
        unsigned char   indicator;  // AM/PM
                                    // RTC_INDICATOR_AM
                                    // RTC_INDICATOR_PM
        unsigned char   day;        // 日(0x31 = 31 日)BCD 形式
        unsigned char   month;      // 月(0x11 = 11 月)BCD 形式
        unsigned char   year;       // 年(0x99 = 99 年)BCD 形式
        unsigned char   week;       // 曜日(0:日,1:月,2:火,3:水,4:木,5:金,6:土)
                                    // RTC_WEEK_SUNDAY
                                    // RTC_WEEK_MONDAY
                                    // RTC_WEEK_TUESDAY
                                    // RTC_WEEK_WEDNESDAY
                                    // RTC_WEEK_THURSDAY
                                    // RTC_WEEK_FRIDAY
                                    // RTC_WEEK_SATURDAY
        unsigned char   hourMode;   // 12/24 時間設定
                                    // RTC_HOUR_MODE_12H
                                    // RTC_HOUR_MODE_24H
} T_RTC_DATE_INFO;
```

RTC 開始

書式	void RtcStart(void)
機能	RTC 開始
引数	なし
戻り値	なし

(処理内容)
RTC の Counter を開始させます。

RTC 停止

書式	void RtcStop(void)
機能	RTC 停止
引数	なし
戻り値	なし

(処理内容)
RTC の Counter を停止させます。

RTC 時刻取得

書式	void RtcGetCurDate(T_RTC_DATE_INFO *pDate)
機能	RTC 時刻取得
引数	pDate -in RTC 時刻取得 Parameter
戻り値	なし

(処理内容)
RTC から時刻を取得し、引数へ返します。

```
typedef struct {
        unsigned char   second;     // 秒(0x48 = 48 秒)BCD 形式
        unsigned char   minute;     // 分(0x48 = 48 分)BCD 形式
        unsigned char   hour;       // 時(0x12 = 12 時)BCD 形式
        unsigned char   indicator;  // AM/PM
                                    // RTC_INDICATOR_AM
                                    // RTC_INDICATOR_PM
        unsigned char   day;        // 日(0x31 = 31 日)BCD 形式
        unsigned char   month;      // 月(0x11 = 11 月)BCD 形式
        unsigned char   year;       // 年(0x99 = 99 年)BCD 形式
        unsigned char   week;       // 曜日(0:日,1:月,2:火,3:水,4:木,5:金,6:土)
                                    // RTC_WEEK_SUNDAY
                                    // RTC_WEEK_MONDAY
                                    // RTC_WEEK_TUESDAY
                                    // RTC_WEEK_WEDNESDAY
                                    // RTC_WEEK_THURSDAY
                                    // RTC_WEEK_FRIDAY
                                    // RTC_WEEK_SATURDAY
        unsigned char   hourMode;   // 12/24 時間設定
                                    // RTC_HOUR_MODE_12H
                                    // RTC_HOUR_MODE_24H
} T_RTC_DATE_INFO;
```

RTC 割込みハンドラ

書式	void RtcIntHdlr(void)
機能	RTC 割込みハンドラ
引数	なし
戻り値	なし

(処理内容)
RTC の割込み処理を行います。
RtcInit 関数の intFunc に登録された処理関数を呼び出し、割込み要因のクリアを行います。

RTC 割込み状態取得(Macro 関数)

書式	unsigned char RtcGetIntStatus(void)
機能	RTC 割込み状態取得
引数	なし
戻り値	0 : 要因なし RTC_INTSTAT_RTCIRQ : 要因あり

(処理内容)
RTC 割込み状態を返します。

RTC 割込み状態クリア(Macro 関数)

書式	void RtcClrIntStatus(void)
機能	RTC 割込み状態クリア
引数	なし
戻り値	なし
(処理内容) RTC 割込み状態をクリアします。	

RTC 割込み Disable(Macro 関数)

書式	void RtcDisableInt(void)
機能	RTC 割込み Disable
引数	なし
戻り値	なし
(処理内容) RTC 割込みを Disable 状態にします。	

RTC 割込み Enable(Macro 関数)

書式	void RtcEnableInt(void)
機能	RTC 割込み Enable
引数	なし
戻り値	なし
(処理内容) RTC 割込みを Enable 状態にします。	

RTC 割込み Edge モード(Macro 関数)

書式	void RtcSetIntModeEdge(void)
機能	RTC 割込み Edge モード設定
引数	なし
戻り値	なし
(処理内容) RTC 割込みを Edge モードに設定します。	

RTC 割込み Level モード(Macro 関数)

書式	void RtcSetIntModeLevel(void)
機能	RTC 割込み Level モード設定
引数	なし
戻り値	なし
(処理内容) RTC 割込みを Level モードに設定します。	

RTC 割込み Cycle 設定(Macro 関数)

書式	void RtcSetIntCycle(x)		
機能	RTC 割込み Cycle 設定		
引数	x	-in	割込みサイクル RTC_INT_CYCLE_1_64S RTC_INT_CYCLE_SECOND RTC_INT_CYCLE_MINUTE RTC_INT_CYCLE_HOUR
戻り値	なし		
(処理内容) RTC の割込み Cycle を設定します。			

RTC 割込み Cycle 取得(Macro 関数)

書式	unsigned char RtcGetIntCycle(void)
機能	RTC 割込み Cycle 取得
引数	なし
戻り値	割込みサイクル RTC_INT_CYCLE_1_64S RTC_INT_CYCLE_SECOND RTC_INT_CYCLE_MINUTE RTC_INT_CYCLE_HOUR
(処理内容) RTC の割込み Cycle を取得します。	

RTC Software Reset 開始(Macro 関数)

書式	void RtcSoftwareResetOn(void)
機能	RTC Software Reset 開始
引数	なし
戻り値	なし
(処理内容) RTC に Software Reset を開始します。 ※Reset 開始を行った場合、RtcSoftwareResetOff で Reset 解除する必要があります。	

RTC Software Reset 解除(Macro 関数)

書式	void RtcSoftwareResetOff(void)
機能	RTC Software Reset 解除
引数	なし
戻り値	なし
(処理内容) RTC に Software Reset を解除します。	

RTC Count 開始(Macro 関数)

書式	void RtcCounterStart(void)
機能	RTC Count 開始
引数	なし
戻り値	なし
(処理内容) RTC の Count を開始させます。	

RTC Count 停止(Macro 関数)

書式	void RtcCounterStop(void)
機能	RTC Count 停止
引数	なし
戻り値	なし
(処理内容) RTC の Count を停止させます。	

30 秒補正(Macro 関数)

書式	void Rtc30sAjustment(void)
機能	30 秒補正
引数	なし
戻り値	なし
(処理内容) RTC の 30 秒補正を行います。	

12 時間制設定(Macro 関数)

書式	void RtcSelectMode12H(void)
機能	12 時間制設定
引数	なし
戻り値	なし
(処理内容) RTC の時間を 12 時間制に設定します。	

24 時間制設定(Macro 関数)

書式	void RtcSelectMode24H(void)
機能	24 時間制設定
引数	なし
戻り値	なし
(処理内容) RTC の時間を 24 時間制に設定します。	

時間制取得(Macro 関数)

書式	unsigned char RtcGetSelectMode(void)
機能	時間制取得
引数	なし
戻り値	RTC_HOUR_MODE_12H RTC_HOUR_MODE_24H
(処理内容) RTC に設定されている時間制を取得します。	

Count Hold 設定(Macro 関数)

書式	void RtcCounterHold(void)
機能	Count Hold 設定
引数	なし
戻り値	なし
(処理内容) RTC の Count を Hold 状態にします。 ※本 API で Hold 状態にした場合、RtcCounterRun で Hold 状態を解除する必要があります。	

Count Hold 解除(Macro 関数)

書式	void RtcCounterRun(void)
機能	Count Hold 解除
引数	なし
戻り値	なし
(処理内容) RTC の Count の Hold 状態を解除します。	

Count Busy 状態取得(Macro 関数)

書式	unsigned char RtcGetBusyStatus(void)
機能	CountBusy 状態取得
引数	なし
戻り値	0：リード/ライト可能状態 1：Busy 状態
(処理内容) RTC の Counter の動作状態を返します。	

秒取得(Macro 関数)

書式	unsigned char RtcGetSecond(void)
機能	秒の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の秒の値を返します。	

秒設定(Macro 関数)

書式	void RtcSetSecond(x)		
機能	秒の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の秒の値を設定します。			

分取得(Macro 関数)

書式	unsigned char RtcGetMinute(void)
機能	分の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の分の値を返します。	

分設定(Macro 関数)

書式	void RtcSetMinute(x)		
機能	分の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の分の値を設定します。			

時取得(Macro 関数)

書式	unsigned char RtcGetHour(void)
機能	時の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の時の値を返します。	

時設定(Macro 関数)

書式	void RtcSetHour(x)		
機能	時の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の時の値を設定します。			

午前・午後取得(Macro 関数)

書式	unsigned char RtcGetHourAMPM(void)
機能	午前・午後の値を取得
引数	なし
戻り値	RTC_INDICATOR_AM RTC_INDICATOR_PM
(処理内容) RTC の午前・午後の値を返します。	

午前設定(Macro 関数)

書式	void RtcSetHourAM(void)
機能	午前の値を設定
引数	なし
戻り値	なし
(処理内容) RTC の時間に午前を設定します。	

午後設定(Macro 関数)

書式	void RtcSetHourPM(void)
機能	午後の値を設定
引数	なし
戻り値	なし
(処理内容) RTC の時間に午後を設定します。	

日取得(Macro 関数)

書式	unsigned char RtcGetDay(void)
機能	日の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の日の値を返します。	

日設定(Macro 関数)

書式	void RtcSetDay(x)		
機能	日の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の日の値を設定します。			

月取得(Macro 関数)

書式	unsigned char RtcGetMonth(void)
機能	月の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の月の値を返します。	

月設定(Macro 関数)

書式	void RtcSetMonth(x)		
機能	月の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の月の値を設定します。			

年取得(Macro 関数)

書式	unsigned char RtcGetYear(void)
機能	年の値を取得
引数	なし
戻り値	0xXY(BCD 形式)　X：十の位、Y：一の位
(処理内容) RTC の年の値を返します。	

年設定(Macro 関数)

書式	void RtcSetYear(x)		
機能	秒の値を設定		
引数	x	-in	0xXY(BCD 形式) X：十の位、Y：一の位
戻り値	なし		
(処理内容) RTC の年の値を設定します。			

曜日取得(Macro 関数)

書式	unsigned char RtcGetDayWeek(void)
機能	曜日の値を取得
引数	なし
戻り値	RTC_WEEK_SUNDAY RTC_WEEK_MONDAY RTC_WEEK_TUESDAY RTC_WEEK_WEDNESDAY RTC_WEEK_THURSDAY RTC_WEEK_FRIDAY RTC_WEEK_SATURDAY
(処理内容) RTC の曜日の値を返します。	

曜日設定(Macro 関数)

書式	void RtcSetDayWeek(x)		
機能	曜日の値を設定		
引数	x	-in	曜日 RTC_WEEK_SUNDAY RTC_WEEK_MONDAY RTC_WEEK_TUESDAY RTC_WEEK_WEDNESDAY RTC_WEEK_THURSDAY RTC_WEEK_FRIDAY RTC_WEEK_SATURDAY
戻り値	なし		
(処理内容) RTC の曜日の値を設定します。			

Wakeup 信号極性取得(Macro 関数)

書式	unsigned char RtcGetWakeupPolarity(void)
機能	Wakeup 信号極性を取得
引数	なし
戻り値	RTC_WAKEUP_HIGH RTC_WAKEUP_LOW
(処理内容) RTC に設定されている Wakeup 信号の極性を返します。	

Wakeup 信号極性設定(Macro 関数)

書式	void RtcSetWakeupPolarity(x)		
機能	Wakeup 信号極性を設定		
引数	x	-in	Wakeup 信号極性 RTC_WAKEUP_HIGH RTC_WAKEUP_LOW
戻り値	なし		
(処理内容) RTC の Wakeup 信号の極性を設定します。			

Wakeup 信号制御状態取得(Macro 関数)

書式	unsigned char RtcGetWakeupControl(void)
機能	Wakeup 信号制御状態を取得
引数	なし
戻り値	RTC_WAKEUP_INACTIVE RTC_WAKEUP_ACTIVE
(処理内容) RTC に設定されている Wakeup 信号の制御状態を返します。	

Wakeup 信号 Active 設定(Macro 関数)

書式	void RtcWakeupActive(void)
機能	Wakeup 信号を Active 設定
引数	なし
戻り値	なし
(処理内容) RTC の Wakeup 信号を Active 状態に設定します。	

Wakeup 信号 Inactive 設定(Macro 関数)

書式	void RtcWakeupInactive(void)
機能	Wakeup 信号を Inactive 設定
引数	なし
戻り値	なし
(処理内容) RTC の Wakeup 信号を Inactive 状態に設定します。	

3.9 ヘッダの定義

ドライバ関数内で使用しているデファイン値について記します。

定義名	値	説明
RTC_INT_DISABLE	0	割込み Disable
RTC_INT_ENABLE	1	割込み Enable
RTC_INT_MODE_EDGE	0	割込み Edge モード
RTC_INT_MODE_LEVEL	1	割込み Level モード
RTC_INT_CYCLE_1_64S	0	割込み Cycle 1/64 秒
RTC_INT_CYCLE_SECOND	1	割込み Cycle 1 秒
RTC_INT_CYCLE_MINUTE	2	割込み Cycle 1 分
RTC_INT_CYCLE_HOUR	3	割込み Cycle 1 時間
RTC_HOUR_MODE_12H	0	12 時間制
RTC_HOUR_MODE_24H	1	24 時間制
RTC_INDICATOR_AM	0	午前
RTC_INDICATOR_PM	1	午後
RTC_WEEK_SUNDAY	0	日曜日
RTC_WEEK_MONDAY	1	月曜日
RTC_WEEK_TUESDAY	2	火曜日
RTC_WEEK_WEDNESDAY	3	水曜日
RTC_WEEK_THURSDAY	4	木曜日
RTC_WEEK_FRIDAY	5	金曜日
RTC_WEEK_SATURDAY	6	土曜日
RTC_WAKEUP_HIGH	0x00	Wakeup 信号 High
RTC_WAKEUP_LOW	0x01	Wakeup 信号 Low
RTC_WAKEUP_INACTIVE	0x00	Wakeup 信号 Inactive
RTC_WAKEUP_ACTIVE	0x02	Wakeup 信号 Active

改定履歴

Date	revision	modification	note
08/04/30	1.0	新規作成	

セイコーエプソン株式会社

半導体事業部　IC 営業部

<IC 国内営業グループ>

東京　〒191-8501　東京都日野市日野 421-8

TEL（042）587-5313（直通）　FAX（042）587-5116

大阪　〒541-0059　大阪市中央区博労町 3-5-1　エプソン大阪ビル 15F

TEL（06）6120-6000（代表）　FAX（06）6120-6100

ドキュメントコード：411449700

2008 年 3 月 作成