RX-8025／機能概要

3種類のRTCにて機能は大雑把に同じと思われるので、ここでは RX-8025の機能概要を 説明します。

① 時計機能:
西歴の下2桁の年と、月、日、時、分、秒までのデータの設定／計時／読み出しが 可能です、
西暦の下2桁が 4の倍数のときは、自動的にうるう年と認識し 2099年までを自動認識します。

② 時計精度調整機能:
時計精度を $\pm 3.05 \times 10^{-6}$単位で進ませる、あるいは 遅らせることが出来ます。
この機能を使用することで
- 季節に合わせた時計精度調整をあらかじめ考慮する事で 1年を通しての時計精度の向上が 可能
- 温度検知機能を有するシステムでは、温度変動に合わせて時計精度を補正する事が可能

になり、より高精度の時計機能を実現出来ます。
注意） 調整出来るのは時計精度のみです。
FOUT端子の 32.768KHz出力へは反映されません。

③ 定周期割り込み発生機能:
定周期の割り込み信号を、/INT A端子から出力できます。 その周波数は、2Hz、1Hz、1/60Hz、毎時、毎月の 5通りから選択できます。
定周期割り込みの出力波形は、通常のパルス状の波形（ 2Hz、1Hz ）CPU インタラプトにも対応できるCPUのレベル割り込みを考慮した波形（毎秒、毎分毎時、毎月）の2つから選択できます。

＜元のデータシートに 上記のように 4つの項目を並べて2つ から選択できます、と 書いてあります。？
しばらくして 再度見てみると、パルス状の波形か、レベル割り込みを考慮した波形、という意味の 2つなのではないのかと思われます。紛らわしい。＞

ホストから割り込みをポーリング可能なフラグビット付きです。

④ アラーム機能:

予め設定された時刻にホストに対する割り込み信号を出すアラーム機能(アラームW機能 と アラームD機能の 2 種) を装備しています。

アラームW 機能は 曜日, 時, 分 の設定が可能で、また、割り込み信号は / INTB 端子から出力されます。 曜日設定は (たとえば) 月水金土日のような複数の曜日の選択が可能です。

アラームD 機能は 時, 分 の設定のみが可能で、また、割り込み信号は / INTA 端子から出力されます。 ホストからそれぞれのアラームを ポーリング可能なフラグビット付きです。

⑤ 発振停止検出機能, 電源低下検出機能 (電圧監視機能) と パワーオンリセット検出機能

発振停止検出機能は、発振が停止したことを記憶するレジスタを持った機能です。

電源低下検出機能(電源電圧監視機能) は、電源電圧が ある一定電圧よりも低くなったことを記憶するレジスタを持った機能です。

検出電圧は、2.1 V と1.3 V の2 種のどちらかを レジスタ設定により選択可能です。 電圧サンプリングは低消費電流を考慮した1 秒周期にて行っています。

発振停止検知機能は、計時データが無効になったことを判定するのに対し、電源電圧監視機能では計時データが、無効になる可能性があることを判定するのに有効です。

また、バッテリの電源電圧監視にも使えます。

これらと パワーオンリセットの発生を検出する機能とを合わせて使用することにより、電源が0V から立ち上がったか または バックアップされていたかの判断の際の 計時データの有効無効判定に有効です。

⑥ CPU とのインタフェース：

SCL(クロック)とSDA(データ)の2 つの信号線により、I2C バスインタフェースにてデータのリード,ライトを行います。

SCL,SDA ともにVDD 側に保護ダイオードがありませんので、回路基板上でプルアップ抵抗を付加することで電源電圧の異なるホストとのデータのインタフェースが可能です。

SCLの最大クロック周波数は400 kHz (1.7V <= VDD)で、I2C バス高速モードに対応しています。

⑦ 32.768 kHz クロック出力

内蔵水晶振動子と同精度の 32.768 kHz クロックを、FOUT 端子から出力することができます。

FOEが Highで 且つ /CLEN1、/CLEN2ビットのどちらか一方でも "0" のときに、FOUT端子から 32.768 kHzが 出力されます。

FOE=Low または OPEN のときには FOUT 出力は停止し、このときのFOUT出力は "L" 固定になります。

注意： [時計精度調整機能]を使用しても、FOUT 端子からの 32.768 kHz クロックの精度は調整できません。

RX-8025／レジスタ説明

RX-8025の I2Cでアクセスする内部レジスタです。
0 ～ Fの 16個の 8bitレジスタで構成されます。

Address	機　能	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	Seconds	0	S40	S20	S10	S8	S4	S2	S1
1	Minutes	0	M40	M20	M10	M8	M4	M2	M1
2	Hours	0	0	H20 P,/A	H10	H8	H4	H2	H0
3	Weekdays	0	0	0	0	0	W4	W2	W1
4	Days	0	0	D20	D10	D8	D4	D2	D1
5	Months	C	0	0	Mo10	Mo8	Mo4	M02	Mo1
6	Years	Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
7	Digital Offset	TEST	F6	F5	F4	F3	F2	F1	F0
8	Alarm_W ; Minute	0	Wm40	Wm20	Wm10	Wm8	Wm4	Wm2	Wm1
9	Alarm_W ; Hour	0	0	Wh20 WP, /A	Wh10	Wh8	Wh4	Wh2	Wh1
A	Alarm_W ; Weekday	0	Ww6	Ww5	Ww4	Ww3	Ww2	Ww1	Ww0
B	Alarm_D ; Minute	0	Dm40	Dm20	Dm10	Dm8	Dm4	Dm2	Dm1
C	Alarm_D ; Hour	0	0	Dh20 DP, /A	Dh10	Dh8	Dh4	Dh2	Dh1
D	Reserved	Read: 全bit = 0 、Write:無効							
E	Control 1	WALE	DALE	/12, 24	/CLEN2	TEST	CT2	CT1	CT0
F	Control 2	VDSL	VDET	/XST	PON	/CLEN1	CTFG	WAFG	DAFG

0 は 常時 0の bitです。 時刻に関わるデータは、2桁の BCDデータです。 現在時刻に関わるレジスタは、Address 0 ～ 6 の レジスタです。

Digital Offsetのレジスタは、季節や温度による時刻補正を行ためのレジスタです。

Alarm_Wは、週単位のアラーム設定レジスタです。 Alarm_Dは、時、分単位のアラーム設定レジスタです。

Control 1、2 は、初期設定や、定周期割り込みに関わるフラグ群です。

前のページの追加説明ですが、灰色の四角に 0 が書き込まれた bit は、読み出し時 常時 0 ですが、データ書き込み時も 0 にして下さい。

BCDデータですが、Binary Coded Decimalの略で二進化十進表現とも言われます。 4bit単位で、10進数の各桁に割り当てた数値表現です。 10進数の123であれば BCDでは 0001, 0010, 0011 になります。
温度による時刻精度補正の Digital Offsetは今回は使いません。 初期化時 00h を書き込んで下さい。

Alarm_W、Alarm_Dの アラーム機能も今回使用しません。 Control 1 レジスタの WALE ビット、DALE ビット共に 0 を設定する事で アラーム機能は停止します。
オレンジ色の[TEST]の ビットは、常に 0 を設定して下さい。
オレンジ色の[C]の ビットは、通常 0 ですが、年が99年から 00年になった時 1 になるとの事です。

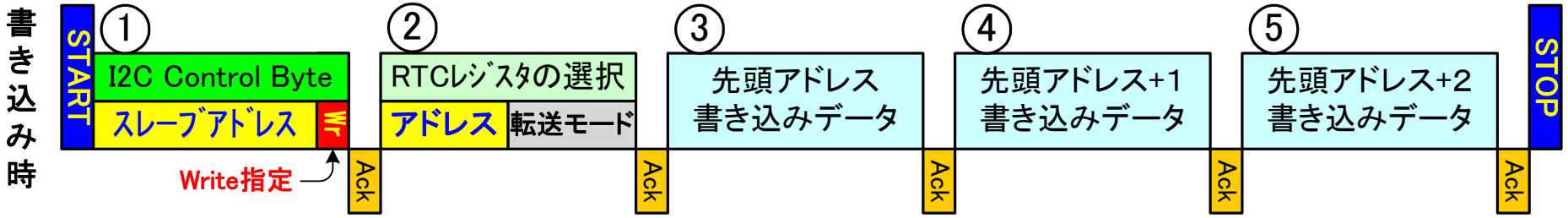
メーカーのデータシートに沿って順次説明を行うと、かなりの情報量になるので、今回使用する機能を実現するコマンドの出し方に絞って説明を行います。
(メーカーのデータシートは、秋月電子のサイトからダウンロード出来ます。)

コマンドの種類としては、以下を想定しています。

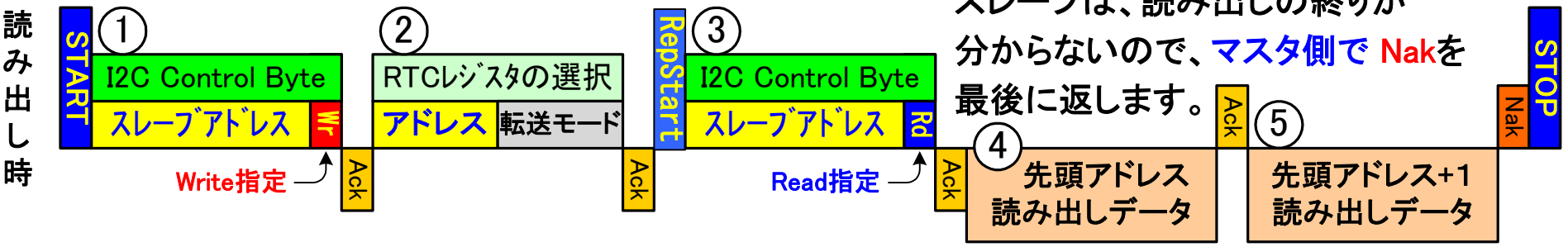
① 初期化コマンド:
(FOUT無し、1秒 割込み無しで初期化)
② バックアップ復帰処理 (電池から主電源に復帰した際、正常に動作しているかの確認処理)
③ 日付時刻設定コマンド:
現在の 年、月、日、時、分、秒の 設定
④ FOUTの出力有無の設定:
⑤ 1秒割り込み有無の設定:
⑥ 日付時刻読み出しコマンド:
現在の 年、月、日、時、分、秒の 読出し
(CTFG=0 : 割り込みフラグのリセット処理も同時に行う)

RX-8025／I2Cプロトコル

RX-8025の I2C電文の出し方ですが、まずは書き込み手順の凡例を示します。
3バイトのデータ書き込み例です。　RTCレジスタの選択の アドレスは、アクセスするレジスタの
先頭アドレスです。　転送モードは通常 0 で 問題ないです。
モード 4 というのもありますが、I2Cの標準的な転送手順では無いので 使わない方が いいと思います。



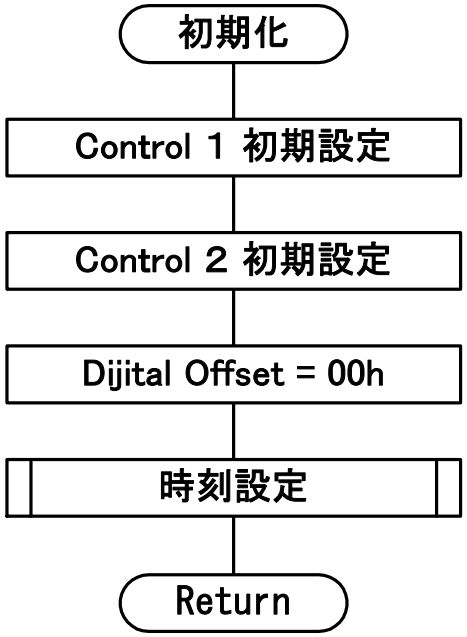
次は、読み出し時の手順の凡例を示します。 読み出しは、読み出す先頭アドレス指定を行った後、
スレーブからデータを出すように、マスタから スレーブに、リピートスタートコンディションを発行します。



スレーブは、読み出しの終りが
分からないので、マスタ側で Nakを
最後に返します。

RX-8025／コマンド処理シーケンス

RTC 型式	I2C スレーブアドレス
RX-8025NB	32 h
RX8900	32 h
RTC-8564NB	51 h



各種制御フラグの
初期設定 (1)

各種制御フラグの
初期設定 (2)

季節や温度による
時刻補正は行わない。

時刻設定のサブプログラムを呼び出す。 時刻データは、一時的であれば、仮のダミー時刻でも良い。
後で正規の時刻に設定し直せば良い。

Control.1の 各フラグ設定：

```
b7 : WALE = 0;    // Alarm_Wは、使用しない。
b6 : DALE = 0;    // Alarm_Dは、使用しない。
b5 : /12,24 = 1; // 24時間制で設定する。
b4 : /CLEN2 = 1; // CLEN1=1と CLEN2=1で
                  // FOUTを出さない。
b3 : TEST = 0;    // TESTは 0 固定
b2 : CT2 = 0;     // CT2=0 & CT1=0 & CT0=0で
b1 : CT1 = 0;     // 定周期タイマー処理は
b0 : CT0 = 0;     // 起動しない。
```

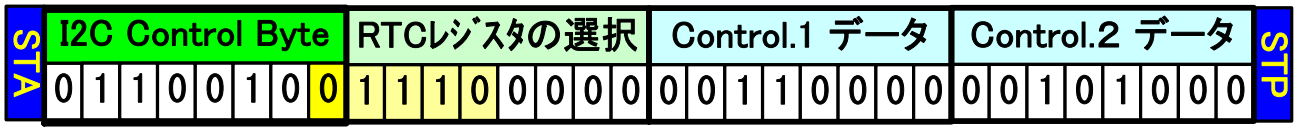
Control.2の 各フラグ設定：

```
b7 : VDSL = 0; // 電源低下検出機能基準値 2.1V
b6 : VDET = 0;  // 電源低下検出ビット クリア
b5 : /XST = 1;  // 発振停止検出ビット クリア
b4 : PON = 0;   // ﾊﾟﾜｰｵﾝﾘｾｯﾄ検出 クリア
b3 : /CLEN1 = 1; // CLEN2参照の事
b2 : CTFG = 0;  // 定周期割込みFlag クリア
b1 : WAFG = 0;  // Alarm_Wの割込みFlag クリア
b0 : DAFG = 0;  // Alarm_Dの割込みFlag クリア
```

よって Ctrl.1 = 30h, Ctrl.2 = 28hに なります。

初期化処理 I2C電文シーケンス

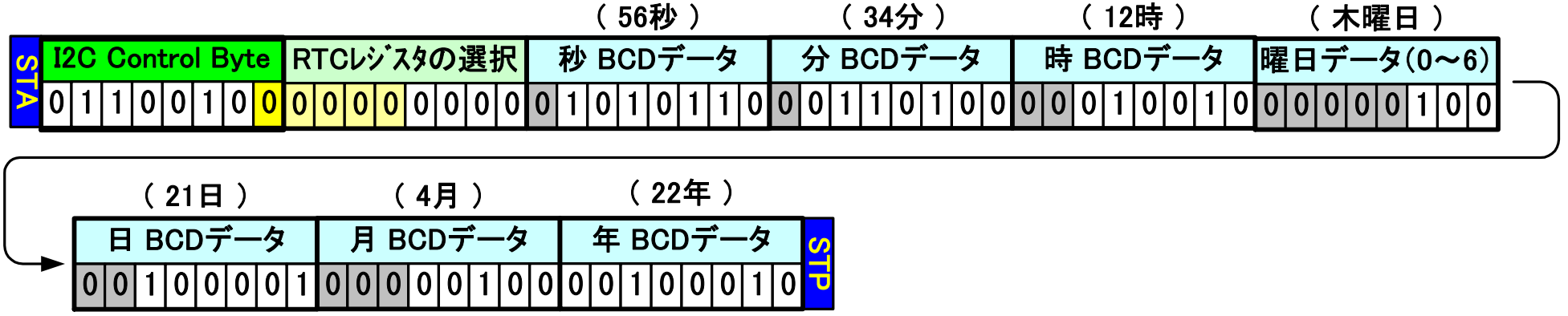
(1) Control.1と Control.2の 初期設定 I2C電文です。



(2) Dijital Offsetの 初期設定 I2C電文です。
レジスタアドレスが、離れているので、個別に初期化しました。

曜日の数値
日曜： 0
月曜： 1
火曜： 2
水曜： 3
木曜： 4
金曜： 5
土曜： 6
（ 7 は設定するな。 ）

(3) 時刻設定の I2C電文の例です。
秒を含む時刻設定後に、Contol.2の VDETを 0 にする必要があります。

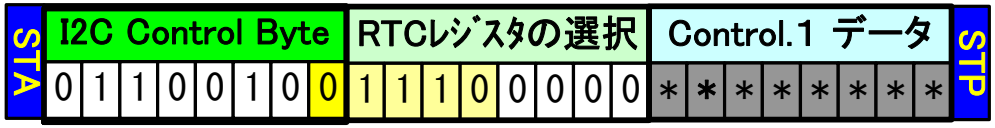


(4)　Control.2の VDETのクリア処理　I2C電文です。

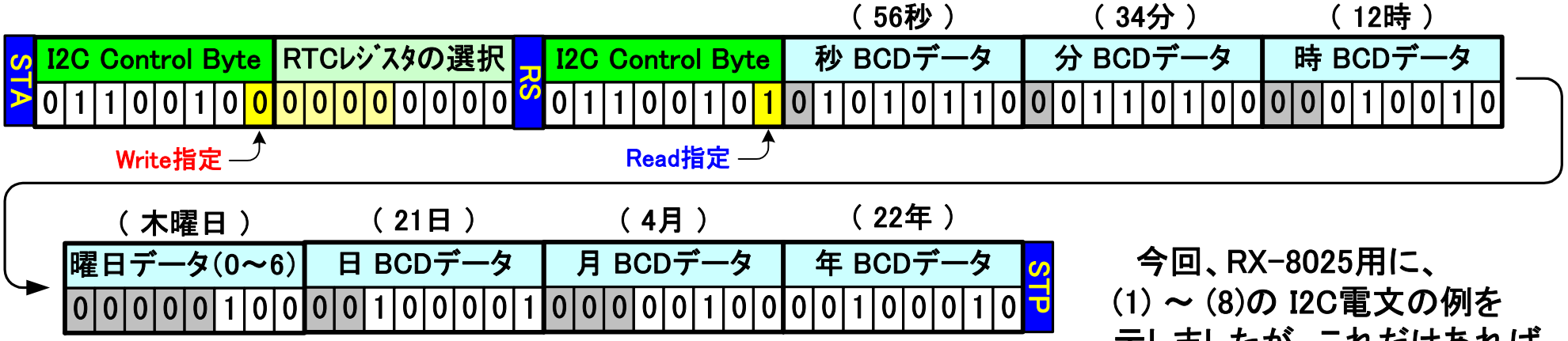
bit単位で、書き込む事は出来ないので、前回 Control.2に書き込んだ
Byte単位の値を マイコンのメモリ上に保存しておく必要があります。
それを、読み出して 変更する bit を AND OR の ビット演算処理で
編集して、Control.2レジスタに書き込む事になります。
もちろん、VDET以外の Control.2の フラグも 変更できます。

(5)　Control.1の bit編集処理　I2C電文です。

Control.1側の フラグも bit単位で、書き込む必要は出て来ると
思われるので、Control.1の フラグ編集用の編集書き込み機能を
用意しておきます。



(6) 時刻読み出しの　I2C電文の例です。
　この場合は、電文の間に リピートスタートを入れて、書き込みから、読み出しモードに切り替えます。



(7) Control.1の 読み出し。 Control.1内の フラグ読み出し確認用です。

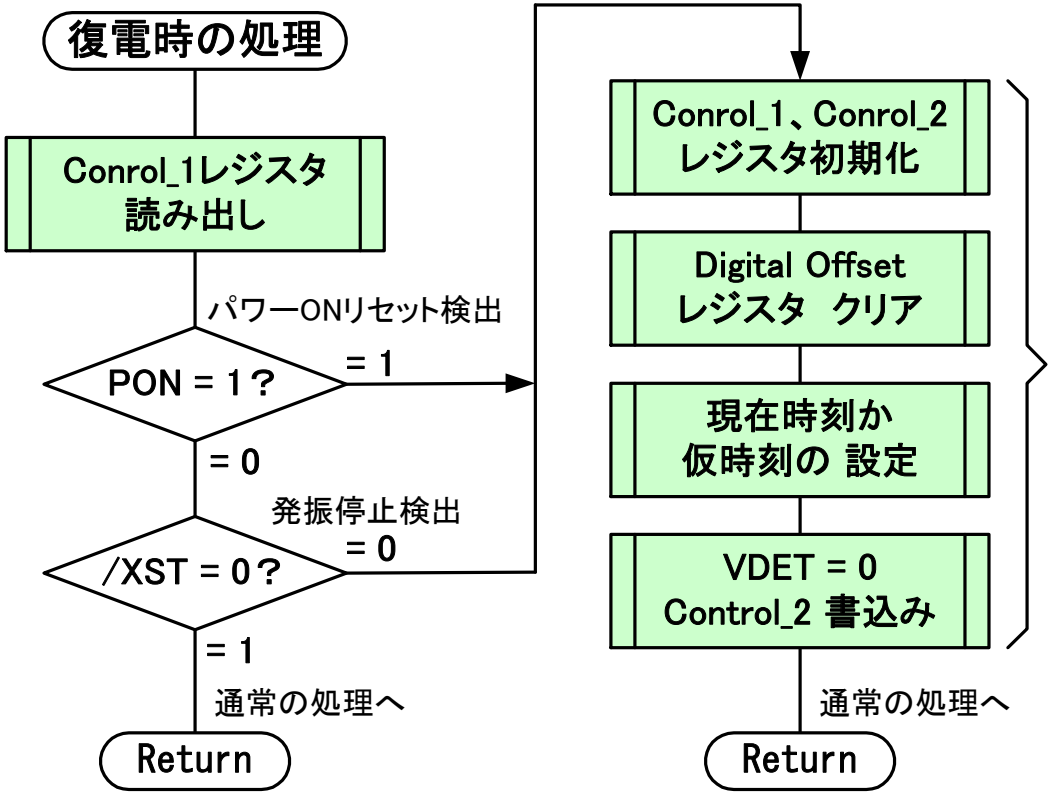
(8) Control.2の 読み出し。 Control.2内の フラグ読み出し確認用です。

今回、RX-8025用に、
(1) ～ (8)の I2C電文の例を示しましたが、これだけあれば、今回の用途には十分です。

RX8900、RTC-8564NBは、レジスタセットの構成、フラグの名称仕様等も少し異なるため、RX-8025用の電文そのままでは 使えません。

でも I2Cの電文の大まかな構成は、同じような構成になると思います。

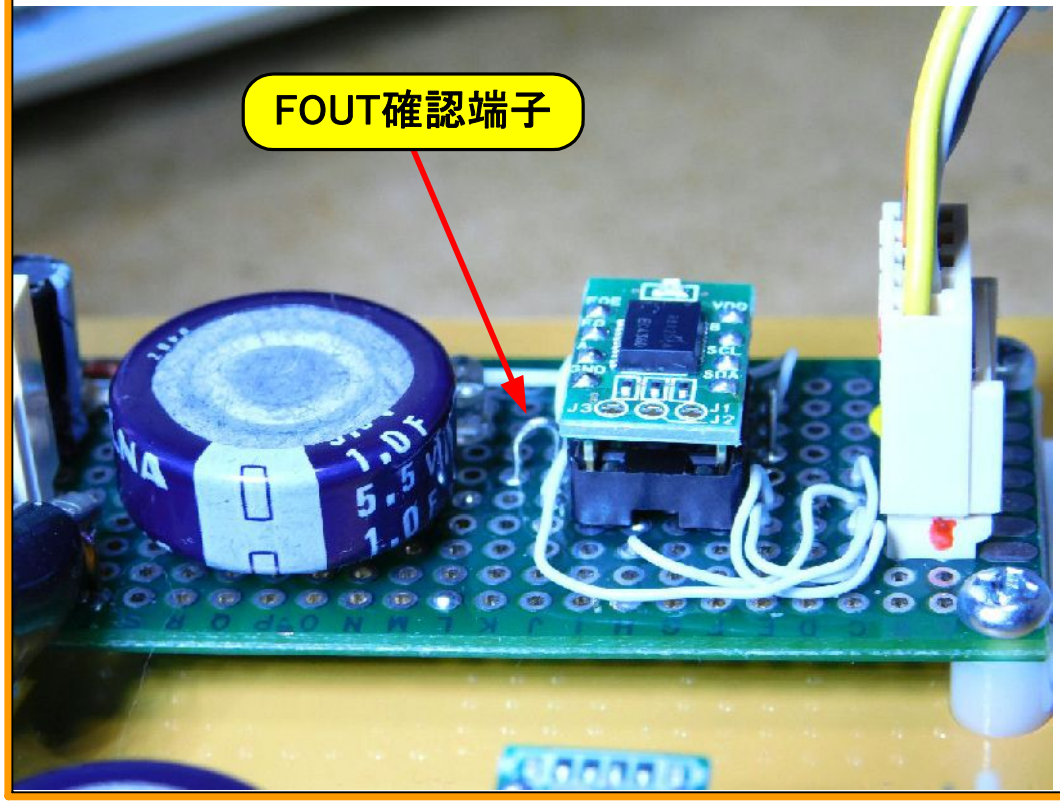
RX-8025／電池から、主電源に復旧した時の処理シーケンス



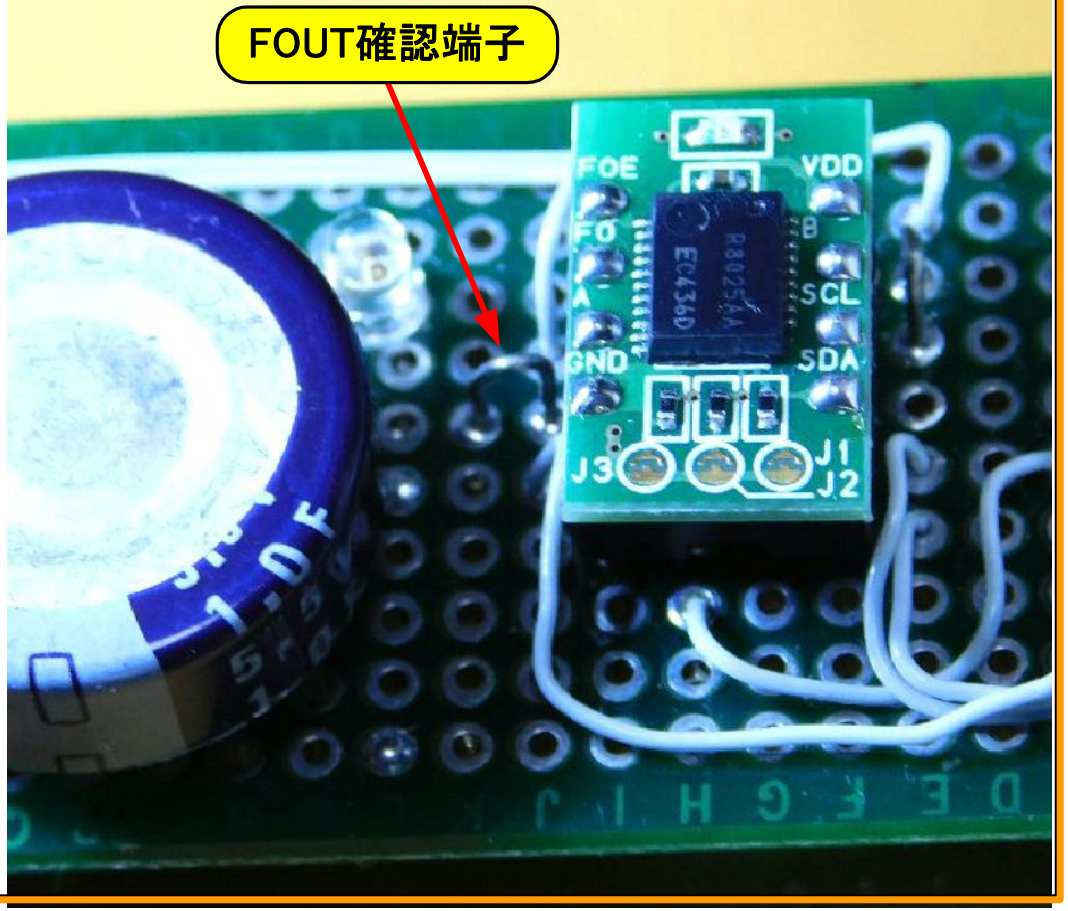
仮時刻 というのは、主電源復旧直後は、システムが 現在時刻を取り込んで 無い場合が 殆どと思われます。
仮に 2020年 1月1日 0時 0分 0秒とか、初期化されたと分かりやすい 年月日時分秒に 仮設定する事です。
そしてオペレーターが 時刻を確認して本来の時刻を、改めて設定する。
という事に なります。

FOUT確認端子の位置

スーパーキャパシタと RTC小基板の間に、FOUT確認端子は あります。 その左上に透明色の LEDが、あります。 これが、RTC小基板 3ピンの /INT Aの Low状態で点灯するLEDです。

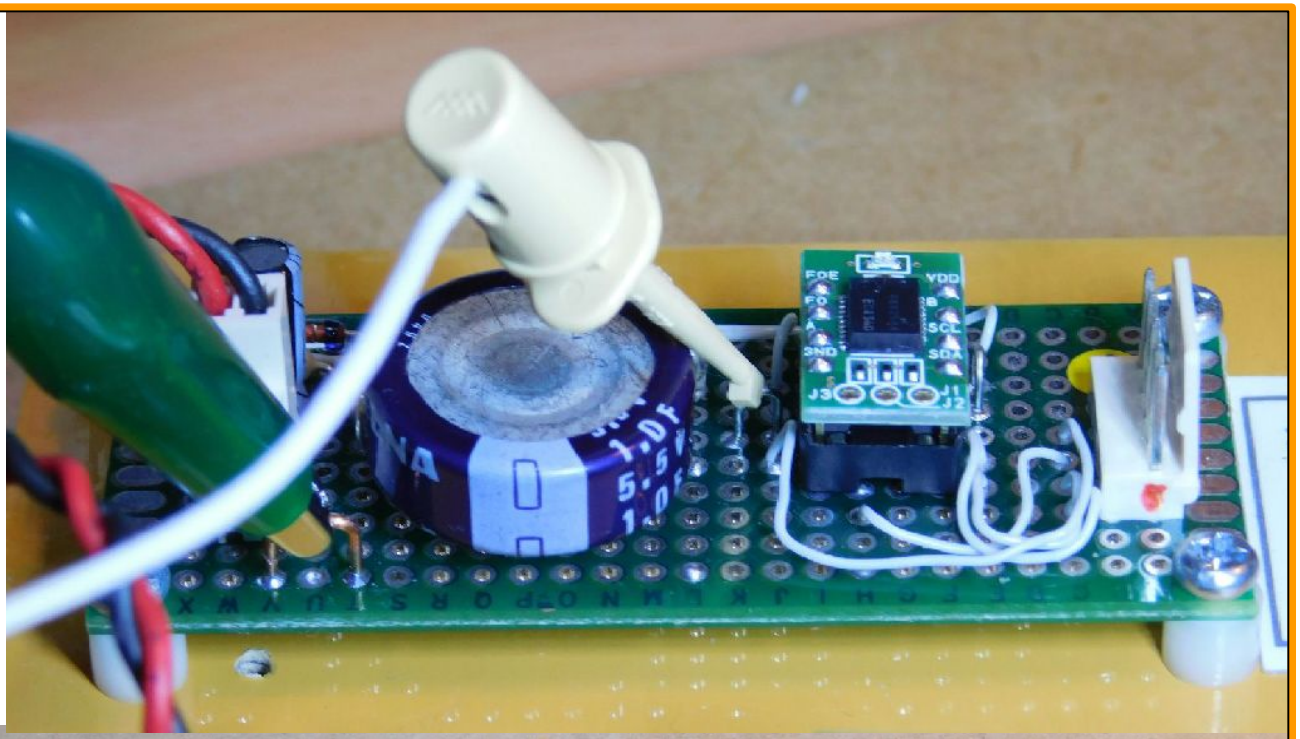


2ピンの FOUT確認端子と、3ピンの /INTA確認 LED、そして 1ピン FOE(主電源VDDと接続)は 接近した位置にあったのです。 多分、主電源VDDと /INTAをミノムシクリップで ショートさせたと思います。



狭い場所の信号測定の対策

今後、狭い箇所での信号測定時に、隣の信号との接触事故を防ぐために、右下の短い信号延長ケーブルを作りました。
片側が、オシロのプローブを小さくしたような ICクリップ（白）と、ミノムシクリップ（緑）で、反対側に ネジ径 3.5mm用の 圧着端子を付けました。 白い方が信号用、緑がGND用です。 圧着端子側に、計測器のプローブや ミノムシクリップを接続します。



結び目から、圧着端子側までの長さを、段違いにしているのも接触事故防止のためです。

次から、この延長ケーブルを使用して測定します。

RX8900用 実験用基板の部分変更

RX8900の 電源端子 VDDと VBATに関する資料を、
データシートで 見つけました。 その回路を見て
私が作ったRX8900用の テスト用基板では、
具合の悪い箇所がある事に 気付きました。

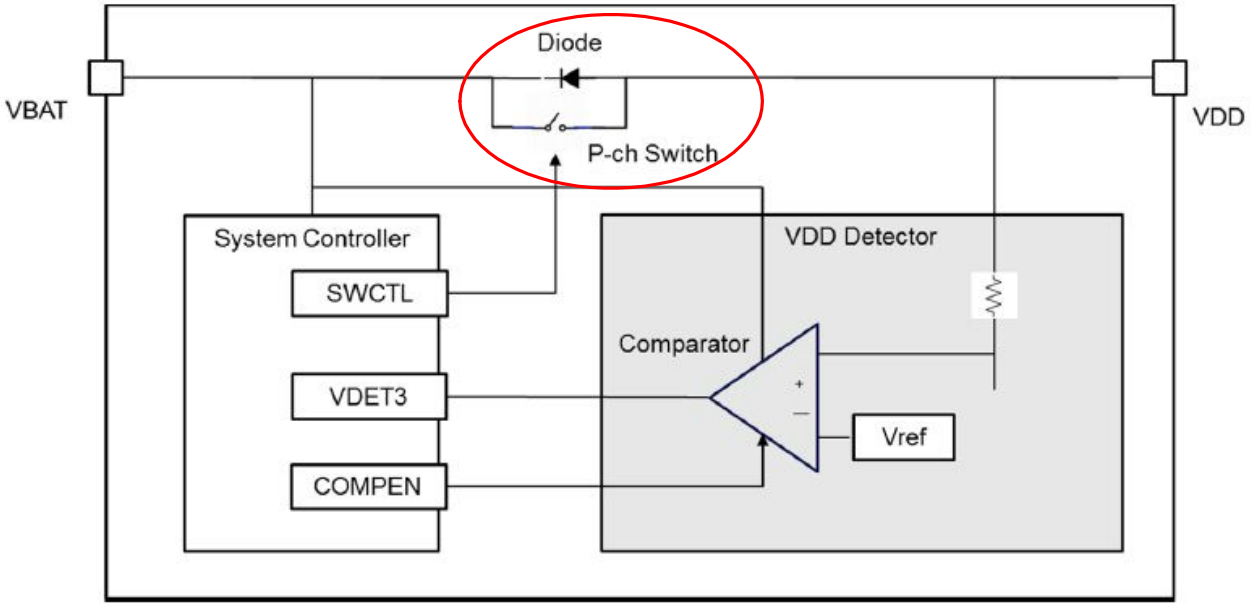


Figure 8-10. バックアップ電源切換機能ブロック図

メーカーのデータシートの説明です。

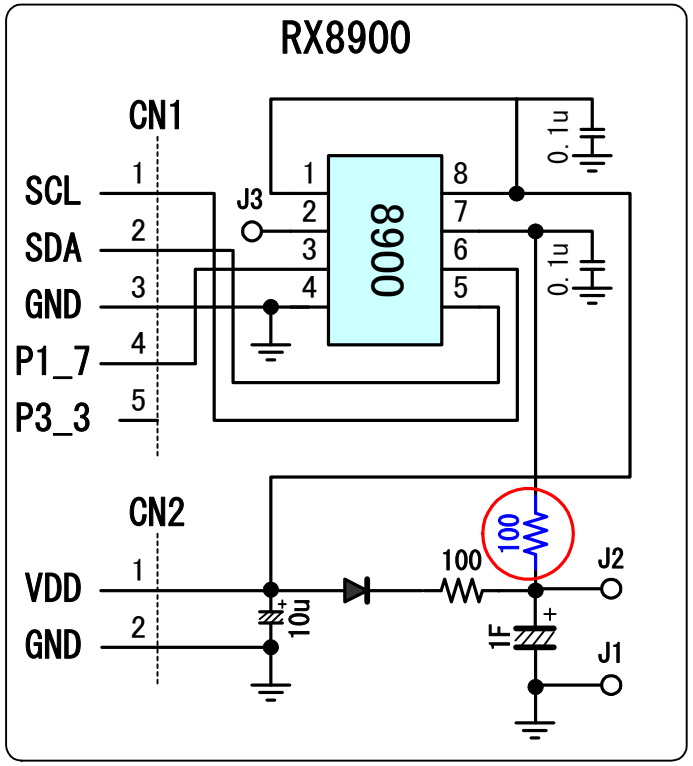
バックアップ電源切換機能の概要
メイン電源VDD の電圧低下を検出する電源電圧検出回路VDD Detector と、メイン電源端子VDD とバックアップ電源端子 VBAT 間に、配置された内蔵Pch-Switch, Diode から構成されます。

Pch-Switch が open 状態でVDD 電圧がスレッショルド電圧(VDET3)以下の場合には、バックアップモードに移行します。

逆にVDD 電圧が VDET3 より大きい場合にはノーマルモードに移行します。

VDD 電圧の測定のため毎秒ごとにPch-Switch を open にして電圧を検出し、その間は diode により VBAT 端子の電池からVDD端子へのリーク電流を阻止します。

まさか、Pch MOSFETによるスイッチで VDDと VBATがショートされるとは 思わなかったので、ショートされた場合 VDD VBAT を通り 1Fのキャパシタを充電するため突入電流で、MOSFETを壊しかねないと思い、電流制限用に、青色の100Ω抵抗を 追加挿入しました。



右側に、間が開いたので
メーカーの うたい文句？を 書いておきます。
RX8900 特長

- 32.768 kHz 温度補償発振器(DTCXO)を搭載
  高精度
- I2C-Bus シリアル・インターフェース
- 曜,日,時,分のアラーム割り込み機能
- 定周期タイマー割り込み機能
- 時刻更新割り込み機能 (毎秒・毎分)
- 電源切り替え機能
- OE 機能付き 32.768 kHz 出力 (FOE, FOUT 端子)
- 自動うるう年補正機能 (2000 ～ 2099 年まで対応)
- 2.5 V ～ 5.5 V の幅広いインターフェース電圧範囲
- 2.0 V ～ 5.5 V の幅広い温度補償電圧範囲
- 低消費電流 0.70 μA / 3 V (Typ.)
- 1.6 V ～ 5.5 V の幅広い計時(保持)電圧範囲

RX8900／レジスタテーブル1

RX8900の I2Cでアクセスする内部レジスタです。
00～0Fと 17～1Aの 20個の レジスタで構成されます。

Address	機　能	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00 or 10	Seconds	0	40	20	10	8	4	2	1
01 or 11	Minutes	0	40	20	10	8	4	2	1
02 or 12	Hours	0	0	20	10	8	4	2	1
03 or 13	Weekdays	0	土	金	木	水	火	月	日
04 or 14	Days	0	0	20	10	8	4	2	1
05 or 15	Months	0	0	0	10	8	4	2	1
06 or 16	Years	80	40	20	10	8	4	2	1
07	RAM	*	*	*	*	*	*	*	*
08	MIN Alarm	AE	40	20	10	8	4	2	1
09	HOUR Alarm	AE	0	20	10	8	4	2	1
0A	WEEK Alarm	AE	土	金	木	水	火	月	日
	DAY Alarm		0	20	10	8	4	2	1
0B or 1B	Timer Counter 0	128	64	32	16	8	4	2	1
0C or 1C	Timer Counter 1	0	0	0	0	2048	1024	512	256
0D or 1D	Extension Register	TEST	WADA	USEL	TE	FSEL1	FSEL0	TSEL1	TSEL0
0E or 1E	Flag Register	0	0	UF	TF	AF	0	VLF	VDET
0F or 1F	Control Register	CSEL1	CSEL0	UIE	TIE	AIE	0	0	RESET

10～16 と 1B～1Fは 00～06 と 0B～0Fの ミラーアドレスとなります。 どちらをアクセスしても同じです。
07～0Aと 17～1Aは 別のレジスタとなります。 17～1Aのレジスタは次のページで示します。
0 は Write不能で Readでは常時 0 です。
TEST は、Write時は、必ず 0 を書き込む事。
07の RAMは 任意のデータを書き込む事が出来ます。読み出しは最後に書いたデータが出てきます。

RX8900／レジスタテーブル2

Address	機　能	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
17	TEMP	128	64	32	16	8	4	2	1
18	Backup Function	0	0	0	0	VDETOFF	SWOFF	BKSMP1	BKSMP0
19	Not use	0	0	0	0	0	0	0	0
1A	Not use	0	0	0	0	0	0	0	0

RX8900に関しても
今回使用する機能だけ
説明します。
よってアラーム関係の
機能は説明しません。

①　まず、00 ～ 06の 時刻設定、読み出しのレジスタは RX-8025と 同じレジスタアドレスになっているので I2Cの書き込み、読み出し関数は、RX-8025と同じ関数が使えそうです。 但し、03 レジスタ　曜日データのフォーマットが異なります。

Address	bit6	bit5	bit4	bit3	bit2	bit1	bit0
03	土	金	木	水	火	月	日

bit0 ～ bit6の 各ビットが、日曜 ～ 土曜に対応しています。 よって2つ以上の ビットを 1にしてはいけません。 bit0 ～ bit6のどれか一つだけビットを 1 にします。 bit7 は、常時 0 にしておきます。西歴下2桁 年、月、日、時、分、秒は、BCD2桁データです。

それと時は、24時間制 固定です。
最初の、初期化は全てのレジスタを初期化して下さい。その際、あり得ない日時は設定しないで下さい。 と、書いてありました。
日時の仮初期化は、月と 日は 01hにして、年と 時、分、秒は 00h で初期化していいと思います。
ちなみに、曜日は 日曜=01h、月曜=02h、火曜=04h、水曜=08h、木曜=10h、金曜=20h、土用=40h になります。 曜日の仮初期値は、上記のどれかの曜日を設定して下さい。 アラームレジスタの初期化は
08 MIN Alarm = 00h 、09 HOUR Alarm = 00h 、
0A Week or Day Alarm = 01h でいいと思います。

RX8900／タイマーカウンタ

TimerCounter 0 は 下位 8bitで、TimerCounter 1は
上位 4bitの 12bit分周カウンタの初期値になります。
カウンターとして使わないならば、RAMとして使える
そうです。

0B or 1B	Timer Counter 0	128	64	32	16	8	4	2	1
0C or 1C	Timer Counter 1	0	0	0	0	2048	1024	512	256

TSEL1	TSEL0	ソースクロック
0	0	4096Hz
0	1	64Hz
1	0	1Hz
1	1	1/60Hz

左の表は、タイマーカウンタの クロックソースで、4つの周波数のクロックを選択できます。

設定出来る周期は、かなり自由度が あります。
しかしマイコン側にも、タイマー周辺回路は、何本かあるので、これを使用する必要性は、あまり無いような気がします。

このタイマーを使用しない時は、制御フラグの
TE = 0 、TIE = 0 に して下さい。

RX8900／各種制御フラグの説明

0D or 1D	Extension Register	TEST	WADA	USEL	TE	FSEL1	FSEL0	TSEL1	TSEL0

TEST： 弊社のテスト用ビットです。必ず 0 を設定して下さい。

WADA： WEEK/DAY アラームの設定ビットです。
0 を設定すると 曜アラーム 、1 を設定すると 日アラームになります。

USEL： 時刻更新割り込みを、発生させるタイミングを 指定します。
0 を設定すると 秒 更新 、 1 を設定すると 分 更新 に なります。

TE： TE = 1 にて、プリセッタブルダウンカウンタが カウントダウンを開始します。
TE = 0 にて、カウンタは、停止します。

FSEL1、FSEL0： FOUT端子 出力周波数を 指定します。
FSEL1 = 0 、FSEL0 = 0 ： 32.768 KHz
FSEL1 = 0 、FSEL0 = 1 ： 1024 Hz
FSEL1 = 1 、FSEL0 = 0 ： 1 Hz
FSEL1 = 1 、FSEL0 = 1 ： 32.768 KHz

TSEL1、TSEL0： タイマーのクロックソース選択ビットで、前ページに 説明があります。

0E or 1E	Flag Register	0	0	UF	TF	AF	0	VLF	VDET

UF: Update Flag 時刻更新終了時にセットされます。0 を書き込む事で リセット します。 1 は、書き込み出来ません。

TF: Timer Flag タイマーカウンタが、ゼロになった時 1 になります。 0 を書き込む事で リセットします。 1 は、書き込み出来ません。

AF: Alarm Flag アラームが発生すると 1 になります。0 を書き込む事で リセット します。 1 は、書き込み出来ません。

VLF: Voltage Low Flag 本フラグは、以下の2要因で セットされます。

1) ICの電源電圧が VLOW電圧を下回った時

2) 水晶発振が 約 10ms以上 停止したとき

VLFビットが、1 を示している場合は、全てのレジスタデータの初期化を行って下さい。 VLF は 0 を書き込むまで、1 を 保持します。

RESETビットには、影響は受けません。 VLOW電圧の検出は常時監視です。

VDET: ICの電源電圧が、VDET電圧よりも低下したことを検出して温度補償回路の停止を検出保持します。 VDET=1 の時は温度補正が停止した履歴を示し 0 を 書き込むまで保持します。 RESETビットには、影響は受けません。

0F or 1F	Control Register	CSEL1	CSEL0	UIE	TIE	AIE	0	0	RESET

CSEL1、CSEL0: 温度補償動作の時間間隔を指定します。
初期電源投入時、2.0sec になっています。

CSEL1	CSEL0	温度補償間隔
0	0	0.5 sec
0	1	2.0 sec
1	0	10 sec
1	1	30 sec

UIE: 時刻更新時に 割り込みを発生させる。
TIE: タイマーカウント ゼロ時に 割り込みを発生させる。
AIE: アラーム発生時に 割り込みを発生させる。
UIE、TIE、AIE は、 1 で 割り込み有効、0 で 割り込み無効

RESET: 読み出し時は、常時 0 です。
書き込み 0 は、無効です。
(ちょっと分かりにくいかもしれませんが. . .)
書き込み 1 で、クロックと カレンダー回路の秒未満の
カウンターが、0Fレジスタの I2C書き込み終了時の
ストップコンディションのタイミングで、クリアされます。

要は RESET=1を I2Cで書き込んだら、秒以下の端数が
0 になるという事です。 ミリ秒オーダーの高精度な
時刻合わせを行う場合に使用します。

RX8900／温度センサのレジスタ

17	TEMP	128	64	32	16	8	4	2	1

TEMP： TEMPの 8bitレジスタは、温度センサーのデータ読み出し用で、書き込みは出来ません。温度 ℃ に変換するには
温度[℃] = (TEMP[7:0] ＊ 2 -187.19) / 3.218　の 式で
変換して下さい。
試しにいくつか計算してみました。

TEMP=　0 であれば（　0 ＊ 2 -187.19 ） / 3.218 = -58.2 ℃
TEMP=　94 であれば（ 94 ＊ 2 -187.19 ） / 3.218 =　 0.3 ℃
TEMP= 134 であれば（134 ＊ 2 -187.19 ） / 3.218 =　25.1 ℃
TEMP= 255 であれば（255 ＊ 2 -187.19 ） / 3.218 = 100.3 ℃

この RX8900は、内蔵温度センサの値を読み取り、発振回路に対する周波数補正値を、工場出荷検査時に、内蔵メモリーに書き込んであるので、温度変動に対する発振周波数が、保障されます。
と、メーカーデータシートに 書いてあります。
±3.4×10^{-6}／ -40℃ ～ +85℃ （ 月差　9 秒相当 ）

RX8900／バックアップ電源機能

制御フラグ関連の説明は、このページで最後です。

18	Backup Function	0	0	0	0	VDETOFF	SWOFF	BKSMP1	BKSMP0

VDET OFF： Voltage Detect OFF ／ メイン電源 VDD の電圧検出回路の ON、OFFを 制御します。

SWOFF： Switch OFF ／ 逆流防止用内蔵 P-MOSスイッチの ON、OFF を制御します。

BKSMP1、BKSMP0： Backup mode Sampling time ／ メイン電源の電圧検出を間欠駆動させるときの動作時間を制御します。

上記のフラグは、密接に影響しているので、表にして関係を示します。

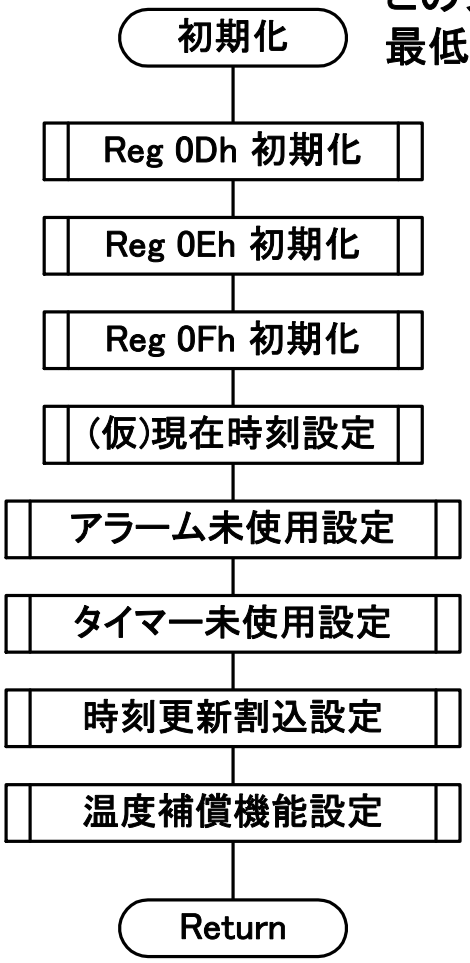
VDD detector	VDET OFF	SWOFF	BKSMP1	BKSMP0	VDET3 検定時間	Pch-Switch ON-OFF	Pch-Switch ON-OFF
ON	0	×	0	0	2ms	2ms	Default
			0	1	16ms	16ms	
			1	0	128ms	128ms	
			1	1	256ms	256ms	
OFF	1	0	×	×	OFF	ON	VDDとVBATは、PMOS SWでショートされます。
		1	×	×	OFF	OFF	VDDは 内蔵ダイオードを介してVBATに接続されます

これらの関係は、表にしても分かりにくいですね。 私も、まだよく理解できてません。

更に詳細を知りたい方は、RX8900のデータシートを ダウンロードして参照して下さい。

RX8900／初期化処理

このフローは、メーカーの初期化例で、
最低限の処理 フローです。



TESTは 0 固定、TE=0は、タイマー停止、
FSEL1=0、FSEL0=0は FOUT端子出力 32768Hz

VLF=0、VDET=0　電源関係の検出フラグをリセット
UF=0、TF=0、AF=0　割り込み通知フラグリセット

UIE=0、TIE=0、AIE=0　割込みは全て禁止

Reg 00 ～ 06 に時刻設定(仮の時刻でも良い)
　曜日が、シフトする ビット設定なので注意する事。

Reg 08 ～ 0A のアラーム設定は 使用しない。 Reg 0F/AIE=0にする。
MIN=0、HOUR=0、Week or Day=1 に仮設定

Reg 0B=01h、Reg 0C=0h　仮に 001Hを設定

時刻更新割込みは使用する予定であるが、最初の初期化段階では 禁止にしておく。
Reg 0Eh、0Fhにて UF=0、UIE=0 に設定しているので、やる事は特にない。

CSEL1=1、CSEL0=0　温度補償動作インターバル　2秒
Reg 0Fh の初期化に含める事が可能。

Reg Adr	機　能	設定値
00 ～ 06	時刻設定	適切な時刻
08 と 09	ｱﾗｰﾑ 時,分	00h, 00h
0A	ｱﾗｰﾑ 曜or日	01h
0B と 0C	TimerCtrl	01h, 00h
0D	Exten	0000 0000 = 00h
0E	Flag Reg	0000 0000 = 00h
0F	Ctrl Reg	0100 0001 = 41h

Reg	値
00 秒	00h
01 分	00h
02 時	00h
03 週	40h
04 日	01h
05 月	01h
06 年	22h

22年 1月 1日(土)
00時 00分 00秒

RX8900／初期化処理 I2C電文シーケンス例

曜日の値
日曜： 01h
月曜： 02h
火曜： 04h
水曜： 08h
木曜： 10h
金曜： 20h
土曜： 40h

(B1) Extension Reg と Flag Reg と Cotrol Regの 初期設定 I2C電文です。

STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Exten Reg データ	Flag Reg データ	Control Regデータ	STP
	0 1 1 0 0 1 0 0	0 0 0 0 1 1 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 1	

(B2) タイマー初期化（ 未使用 ）

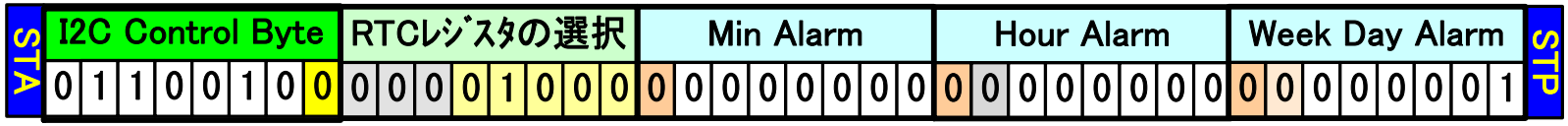
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Timer Counter 0	Timer Counter 1	STP
	0 1 1 0 0 1 0 0	0 0 0 0 1 0 1 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0	

(B3) 時刻設定の I2C電文の例です。

			（ 56秒 ）	（ 34分 ）	（ 12時 ）	（ 木曜日 ）
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	秒 BCDデータ	分 BCDデータ	時 BCDデータ	曜日データ(0～6)
	0 1 1 0 0 1 0 0	0 0 0 0 0 0 0 0	0 1 0 1 0 1 1 0	0 0 1 1 0 1 0 0	0 0 0 1 0 0 1 0	0 0 0 1 0 0 0 0

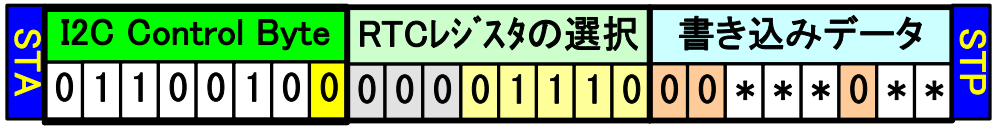
（ 21日 ）	（ 4月 ）	（ 22年 ）		
日 BCDデータ	月 BCDデータ	年 BCDデータ	STP	
0 0 1 0 0 0 0 1	0 0 0 0 0 1 0 0	0 0 1 0 0 0 1 0		

(B4) アラーム初期化（未使用）

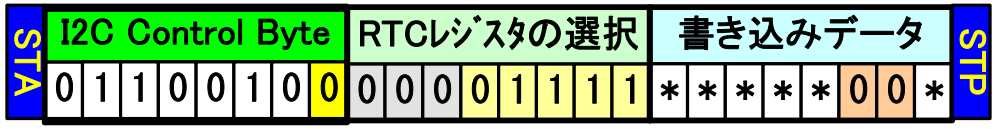


(B5) Extension Regster 設定

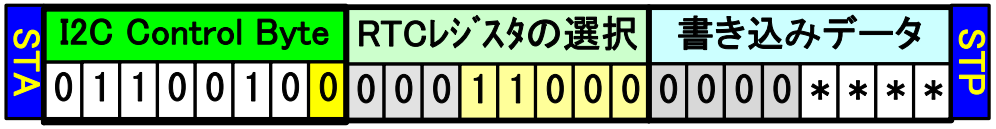
(B6) Flag Register 設定



(B7) Control Register 設定

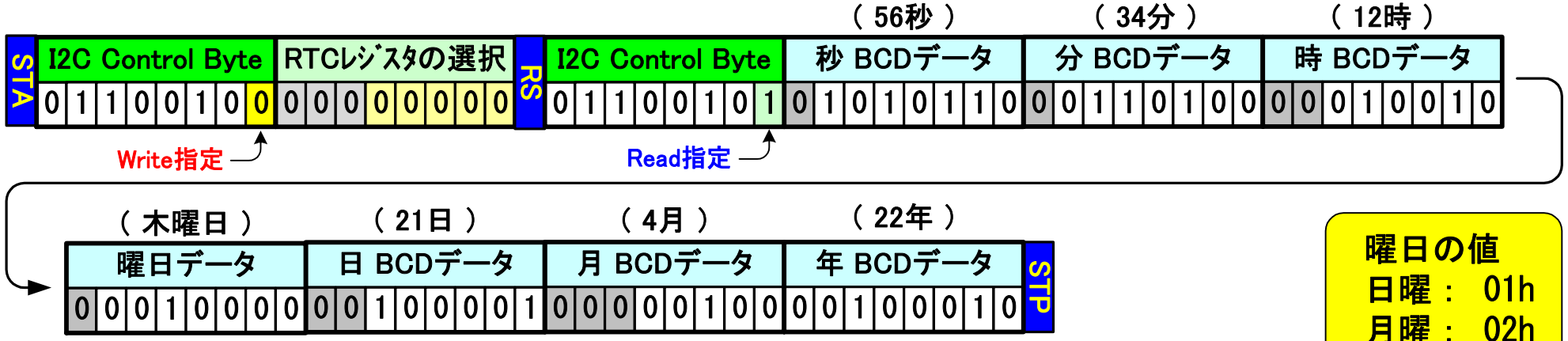


(B8) Backup Function 設定



(B9) 時刻読み出しの　I2C電文の例です。

　この場合は、電文の間に リピートスタートを入れて、書き込みから、読み出しモードに切り替えます。



曜日の値

日曜：　01h
月曜：　02h
火曜：　04h
水曜：　08h
木曜：　10h
金曜：　20h
土曜：　40h

(B10) Flag Register 読み出し （ ステータス読み出しは Flag Regster だけで OKと思います ）

ICの電源電圧が VDET電圧より低下

① ICの電源電圧が VLOW電圧未満
② 水晶発振が 10ms以上停止した

アラーム 割り込みが発生した

タイマー 割り込みが発生した

時刻更新 割り込みが発生した

RTC-8564／機能概要

- 32.768 KHz 水晶振動子(周波数精度 調整済み)を内蔵
- インタフェース方式: I2C BUS シリアル インタフェース
  高速バス規格(400KHz) 対応
- インタフェース電圧範囲: 1.8V ～ 5.5V
- 計時(保持)電圧範囲: 1.0V ～ 5.5V ／ Ta = -20℃ ～ +70℃
- バックアップ時消費電流: 275nA (Typ) ／ VDD = 3.0V
- 出力制御付き 32.768 KHz 出力機能: CLKOUT 端子出力 (C-MOS 出力)
- リアルタイムクロック機能: 時計・カレンダ機能、自動うるう年補正機能
  アラーム割込み機能、定周期タイマ割込み機能 等

入出力端子等の ハードの説明は、前回しているので、次のページから、I2Cでアクセスする
レジスタテーブルの説明から始めます。

RTC-8564／レジスタ説明

RTC-8564の I2Cでアクセスする内部レジスタです。
0 ～ Fの 16個の 8bitレジスタで構成されます。

Address	機　能	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0	Control 1	TEST	0	STOP	0	TEST	0	0	0
1	Control 2	0	x	0	TI/TP	AF	TF	AIE	TIE
2	Seconds	VL	40	20	10	8	4	2	1
3	Minutes	x	40	20	10	8	4	2	1
4	Hours	x	x	20	10	8	4	2	1
5	Days	x	x	20	10	8	4	2	1
6	Weekdays	x	x	x	x	x	4	2	1
7	Months	C	x	x	10	8	4	2	1
8	Years	80	40	20	10	8	4	2	1
9	Minute Alarm	AE	40	20	10	8	4	2	1
A	Hour Alarm	AE	x	20	10	8	4	2	1
B	Day Alarm	AE	x	20	10	8	4	2	1
C	Weekday Alarm	AE	x	x	x	x	4	2	1
D	CLKOUT Frequency	FE	x	x	x	x	x	FD1	FD0
E	Timer Control	TE	x	x	x	x	x	TD1	TD0
F	Timer	128	64	32	16	8	4	2	1

0 は 読み書き共に常時 0の bitです。

x は、書込み不可能で読出し値不定です。

C は、通常 0 で 99年から 00年になった時、1になります。

時刻に関わるデータは、2桁の BCDデータです。 現在時刻に関わるレジスタは、Address 2 ～ 8 の レジスタです。

Control 1、2 は、初期設定や、割り込みに関わるフラグ群です。

TEST は、メーカーテスト用ビットで 必ず 0 を設定して下さい。

Address 0 ／ Control 1 レジスタ:

TEST	0	STOP	0	TEST	0	0	0

TEST: 2つの TESTビットは、弊社テスト用ビットです。 常時 必ず 0 を設定して下さい。

STOP: STOPビットを 1 にすると 時計、カレンダ、アラーム、タイマなどの動作を停止させます。
STOPビットを 0 にすると 動作を再開します。

Address 1 ／ Control 2 レジスタ:

0	x	0	TI/TP	AF	TF	AIE	TIE

TI/TP: 定周期タイマ割込みイベント発生時に、その割込み動作を 1回で終了させるか、または、繰り返し継続させるかを 選択するビットです。
0 = 単発動作 、1 = 繰り返し動作

AF: アラーム割込みイベントを検出して、結果を保持するフラグビットです。 イベントが発生すると 0 → 1 になります。 確認したら 0 にします。

TF: 定周期タイマ割込みイベントを検出して、結果を保持するフラグビットです。イベントが発生すると 0 → 1 になります。 確認したら 0 にします。

AIE: アラーム割込みイベント発生時の /INT割込み信号の動作を設定します。 1 の書き込みにより割込みイベント発生時に /INT端子から L レベルの割り込み信号を出力します。
0 の書き込みでは、/INT端子からの L レベル出力を禁止します。

TIE: 定周期タイマ割込みイベント発生時の /INT割込み信号の動作を設定します。 1 の書き込みにより、割込みイベント発生時に、/INT 端子から L レベルの割り込み信号を出力します。
0 の書き込みでは、/INT端子からの L レベル出力を禁止します。

Address 2 ～ 4 ／ 時計カウンタ:

Seconds	VL	40	20	10	8	4	2	1
Minutes	x	40	20	10	8	4	2	1
Hours	x	x	20	10	8	4	2	1

- 秒、分、時 を 計時します。
- 計時データを置き換える時は、STOPビットを 1 にして、計時動作を停止させた状態で行う事を推薦します。（ データ書き換え中の不用意な桁上げ発生を防止する事で、より適切な時計合わせが出来ます。）

① Seconds レジスタ: 秒 を 計時するカウンタです。

② Minutes レジスタ: 分 を 計時するカウンタです。

③ Hours レジスタ: 時 を 計時するカウンタです。

④ VL ビット:（ Voltage Low Flag ）
電圧低下を検出して、結果を保持するフラグビットです。
電源電圧が、VLOW [V]以下に低下すると、 0 → 1 に 変化します。
読み出し時 1 のときは RTCの内容は無効ですので、その場合は必ず、全てのレジスタを、初期設定してから使用して下さい。

Address 5、7、8 ／ カレンダ カウンタ:

5	Days	x	x	20	10	8	4	2	1
7	Months	C	x	x	10	8	4	2	1
8	Years	80	40	20	10	8	4	2	1

- 2001年 1月 1日 ～ 2099年 12月 31日までの、日 月 年 を オートカレンダ機能によって更新します。
- 存在しないカレンダデータが 書き込まれた場合は 正常な動作が出来ない原因になりますので ご注意ください。

① Days レジスタ
- 日の カウンタです。
- 月によって更新状況が、異なります。
- 年が、4の倍数の時は、うるう年として、2月が 29日まで あります。

② Months レジスタ
- 月の カウンタです。
- Cビット: 99年から 00年に更新された時に 1 になります。

③ Years レジスタ
- 年の カウンタです。
- 年が 4の倍数のときは、うるう年になります。

Address 6 ／ 曜日 カウンタ:

6	Weekdays	x	x	x	x	x	4	2	1

・ 曜日を、bit0 ～ bit2 の 3ビットにて示します。
000 = 日曜 、001 = 月曜 、010 = 火曜 、011 = 水曜 、100 = 木曜 、
101 = 金曜 、110 = 土用　で、RX-8025 と　同じです。

Address 9 ～ C ／ アラーム レジスタ:

9	Minute Alarm	AE	40	20	10	8	4	2	1
A	Hour Alarm	AE	x	20	10	8	4	2	1
B	Day Alarm	AE	x	20	10	8	4	2	1
C	Weekday Alarm	AE	x	x	x	x	4	2	1

・ アラーム割り込み機能を使用して　曜、日、時、分　などに対する割り込みイベントを
得たいときに、AIE 、AF ビットと 共に 設定 使用します。

・ 上記アラームレジスタの設定状況に　現時刻が 一致すると　AF ビット = 1 かつ
/INT 端子 = Low と なるなど、アラーム割り込みのイベントの発生を知る事が
出来ます。

・ アラームレジスタ／AE ビットの説明が、データシートに書いて無い。？　ので 推測ですが、
Alarm Enable の略で 各レジスタを 1 で 有効にする。 という事ではないかと 思います。

Address E ／ 定周期タイマ割込み機能　制御レジスタ

E	Timer Control	TE	x	x	x	x	x	TD1	TD0

- 定周期タイマ割り込み機能を 制御するためのレジスタです。
- 定周期タイマ割り込み機能を 使用するには、TI/TP ビット、Timer レジスタ 及び TF、TIE ビットと　共に　設定 使用します。

① TE ビット:　定周期タイマ割り込み機能の、動作を制御するビットです。
　1 の書き込みで、動作を開始します。 0 の書き込みで、動作を停止します。
② TD1、TD0 は、カウントダウン周期を 選択指定するビットです。（ 右図参照 ）

TD1	TD0	周期
0	0	4096Hz
0	1	64Hz
1	0	1秒更新時
1	1	1分更新時

Address F ／ 定周期タイマ用 ダウン カウンタ

F	Timer	128	64	32	16	8	4	2	1

- 定周期タイマ割り込み機能を 使用するさいの、カウントダウン初期値（プリセット値）を、設定するレジスタです。 カウンタの設定は、1（ 01h ）～ 255（ FFh ）の範囲で設定できます。
- 定周期タイマ割り込み機能を 使用するには、 TE、TI/TP、TF、TIE、TD1、TD0 ビットと　共に設定 使用します。
- 本ダウンカウンタのカウント値が、01h → 00h になると、TF ビット = 1、/INT端子 = Low と　なるなど、定周期タイマ割り込みイベントの発生を知る事が出来ます。

Address D ／ CLKOUT 出力 設定 レジスタ

D	CLKOUT Frequency	FE	x	x	x	x	x	FD1	FD0

- CLKOUT 出力端子の クロック出力を 制御します。
- CLKOE 入力端子 = H の時のみ 本レジスタが有効となり、本レジスタの設定による クロックを出力 、もしくは 出力停止します。
  CLKOE入力端子 = L のときは、本レジスタの設定に関わらず CLKOUT = L と なります。

① FE ビット（ Frequency output enable ）
本ビットが 有効なとき（ CLKOE = H のとき ）に限り、CLKOUT端子の出力状態を 制御します。 本 FE ビットが 1の時、CLKOUT端子を出力状態にします。
この時の出力内容は、FD1、FD0 ビットで指定した周波数になります。
本 FE ビットが 0 のときは、 CLKOUT端子を 出力停止状態 L に します。

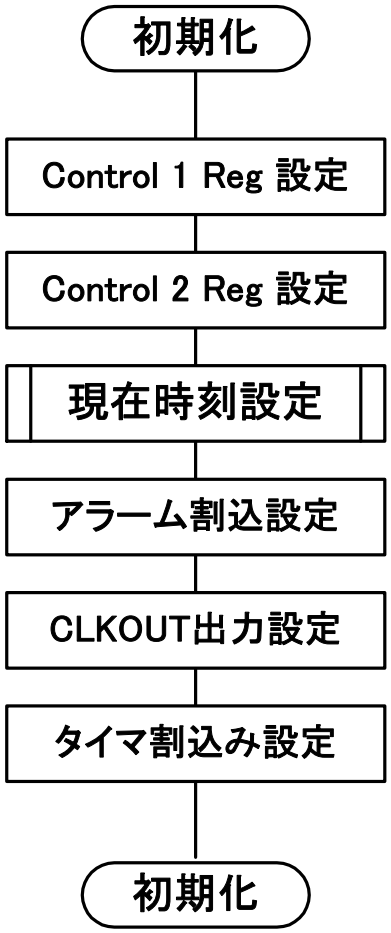
② FD1、FD0 ビット
FD1、FD0 ビットの 組み合わせにより、出力する周波数を選択します。
右の表を参照して下さい。

FE	FD1	FD0	CLKOUT端子
1	0	0	32768Hz 出力
1	0	1	1024Hz 出力
1	1	0	32Hz 出力
1	1	1	1Hz 出力
0	*	*	Low出力

RTC-8564／フローチャート

RTC-8564の I2Cスレーブアドレスは
51H です。

RTC-8564の レジスタアドレス指定は
8bit の 下位 4ビットに設定すると 思われます。



今回のテストプログラム
での設定値

Control 1 = 00h

Control 2 = 11h ／ Timer 繰り返し継続設定 、割り込み出力有り

時刻設定サブ呼び出し（ 仮の初期化時刻でもよい ）

Minute Alarm = 00h、Hour Alarm = 00h、
Day Alarm = 01h、Weekday Alarm = 00h 全て停止状態にする

CLKOUT Frequency = 00h ／ 32.768Hzを出力 停止状態で初期化

Timer Control = 02h 、Timer カウント初期値 = 01h
初期化時は、タイマーを停止状態にする
Timer Control = 82h でタイマは 稼動する

RTC-8564／初期化処理 I2C電文シーケンス

(1) Control.1と Control.2の 初期設定 I2C電文です。(Control.1 = 20h は、一時的に 計時を停止します)

		(00h)	(20h)	(11h)	
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Control.1 データ	Control.2 データ	STP
	1 0 1 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 1 0 0 0 0 0	0 0 0 1 0 0 0 1	

(2) 時刻設定の I2C電文の例です。(他の 2つのRTCと異なり 曜日と 日 が入れ替わってます。)

		(02h)	(56秒)	(34分)	(12時)	(21日)
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	秒 BCDデータ	分 BCDデータ	時 BCDデータ	日 BCDデータ
	1 0 1 0 0 0 1 0	0 0 0 0 0 0 1 0	0 1 0 1 0 1 1 0	0 0 1 1 0 1 0 0	0 0 0 1 0 0 1 0	0 0 1 0 0 0 0 1

	(木曜日)	(4月)	(22年)	
	曜日データ(0～6)	月 BCDデータ	年 BCDデータ	STP
	0 0 0 0 0 1 0 0	0 0 0 0 0 1 0 0	0 0 1 0 0 0 1 0	

(3) 計時を 開始するために Control.1 = 00h を書き込みます。

		(00h)	(00h)	
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Control.1 データ	STP
	1 0 1 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	

時刻を設定する場合は、設定前に
STOP = 1 で 計時を 一時停止させて、
時刻を設定し、
STOP = 0 で 計時を 再開します。

(4) アラーム割り込み設定（ 全て停止 ）

		（09h）	（00h）	（00h）	（01h）	（00h）	
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Minute Alarm	Hour Alarm	Day Alarm	Weekday Alarm	STP
1 0 1 0 0 0 1 0	0 0 0 0 1 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 0		

(5) CLKOUT 出力設定　CLKOUT Frequency = 00h（ 停止状態 ）を書き込みます。

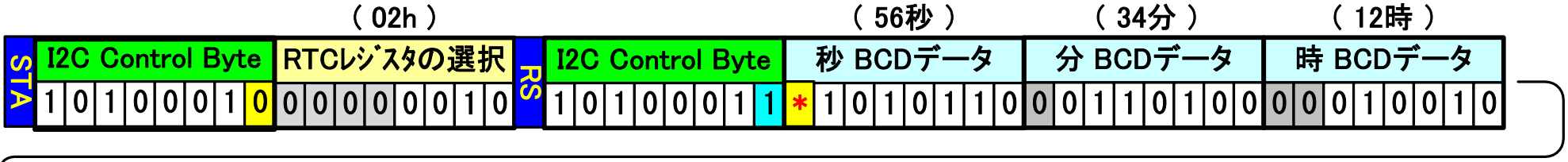
		（0Dh）	（00h）	
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	CLKOUT Freq	STP
1 0 1 0 0 0 1 0	0 0 0 0 1 1 0 1	0 0 0 0 0 0 0 0		

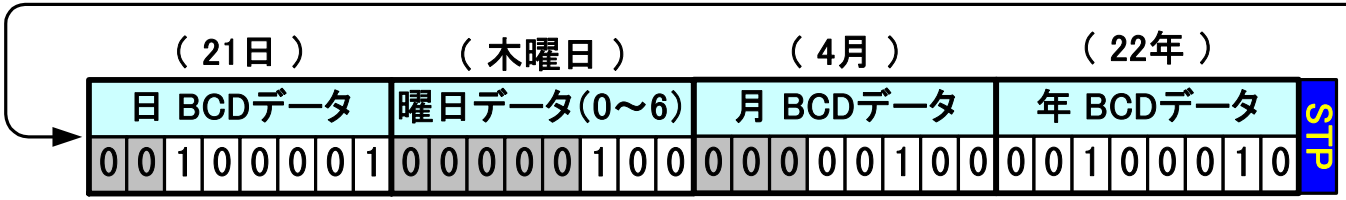
(6) Timer Control、Timer 設定（ Timer Conrol = 停止状態で、秒単位のカウント設定 ）

		（0Eh）	（02h）	（01h）	
STA	I2C Control Byte	RTCﾚｼﾞｽﾀの選択	Timer Control	Timer カウント値	STP
1 0 1 0 0 0 1 0	0 0 0 0 1 1 1 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 1		

RTC-8564／その他処理 I2C電文シーケンス

(8) 時刻読み出しの I2C電文の例です。（ 曜日と 日 が入れ替わってます。 ）
　　読み出し時は、STOP ビットの操作は、必要無いようです。





秒　BCDデータの最上位 bitは
VL(Voltage Low Flag)です。
1 の 時は再初期化が必要です

(9) Timer Start 設定 （ Timer Control 最上位Bit = 1 で 秒単位で カウント開始 ）

(10) Timer Stop 設定 （ Timer Control 最上位Bit = 0 で カウント停止 ）

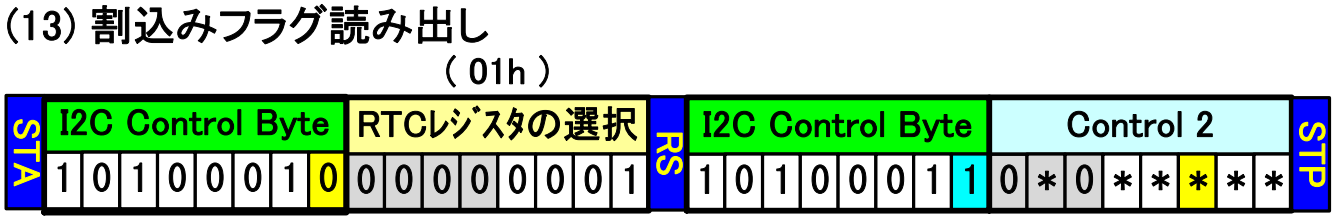
曜日の数値

日曜：	0
月曜：	1
火曜：	2
水曜：	3
木曜：	4
金曜：	5
土曜：	6

(11) CLKOUT 出力設定 （ CLKOUT端子から 32768Hzを 出す ）

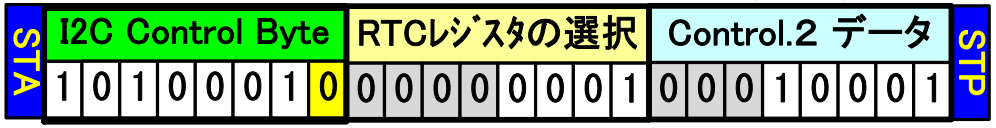
(12) CLKOUT 出力設定 （ CLKOUT端子から 32768Hzを 出さない ）

(13) 割込みフラグ読み出し



TF タイマ割り込み通知フラグ
TF = 1 で、割り込み有り
確認したら、TF = 0 にする

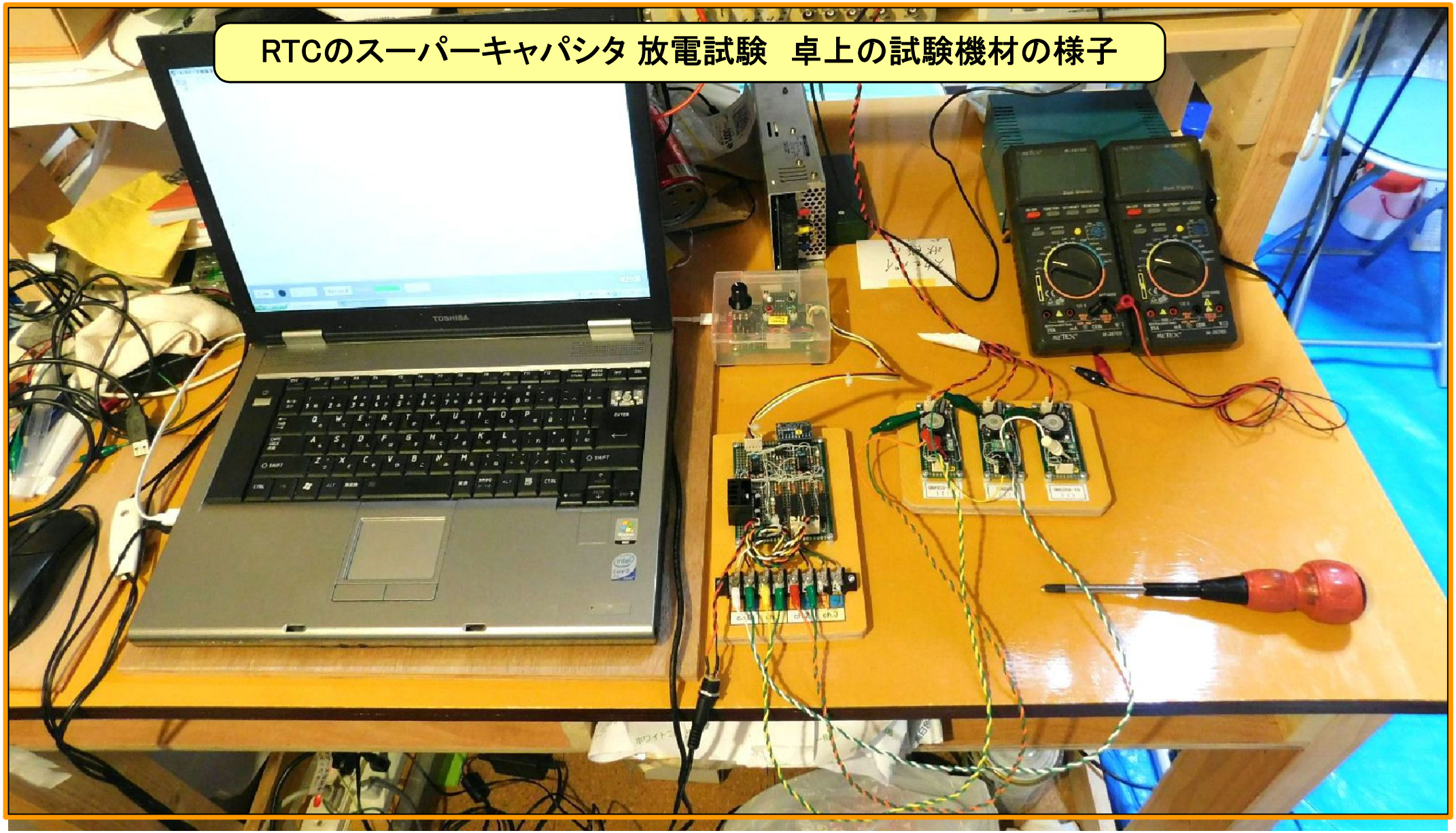
(14) Control 2 書き込み(割込み通知フラグ リセット用)

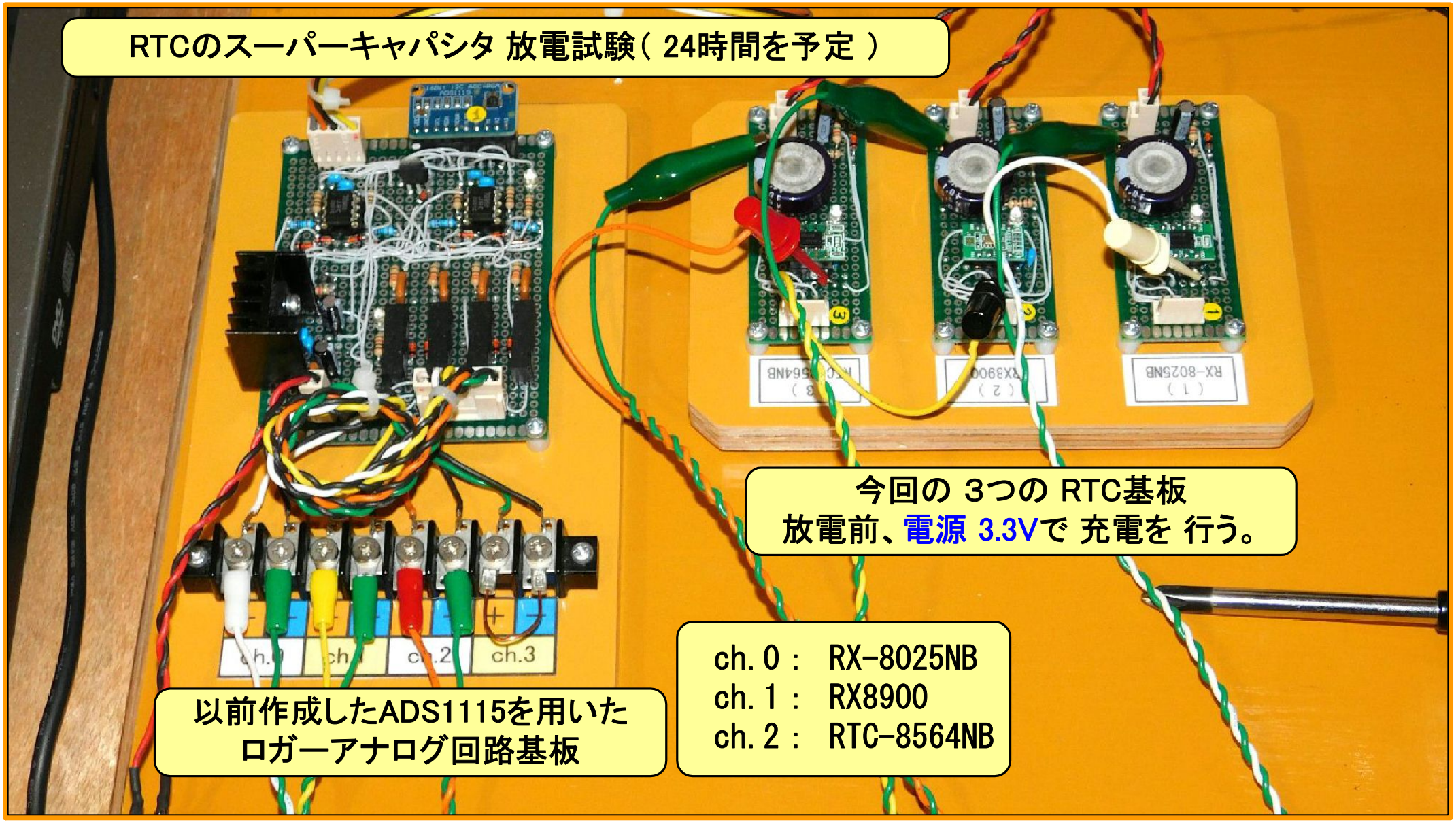


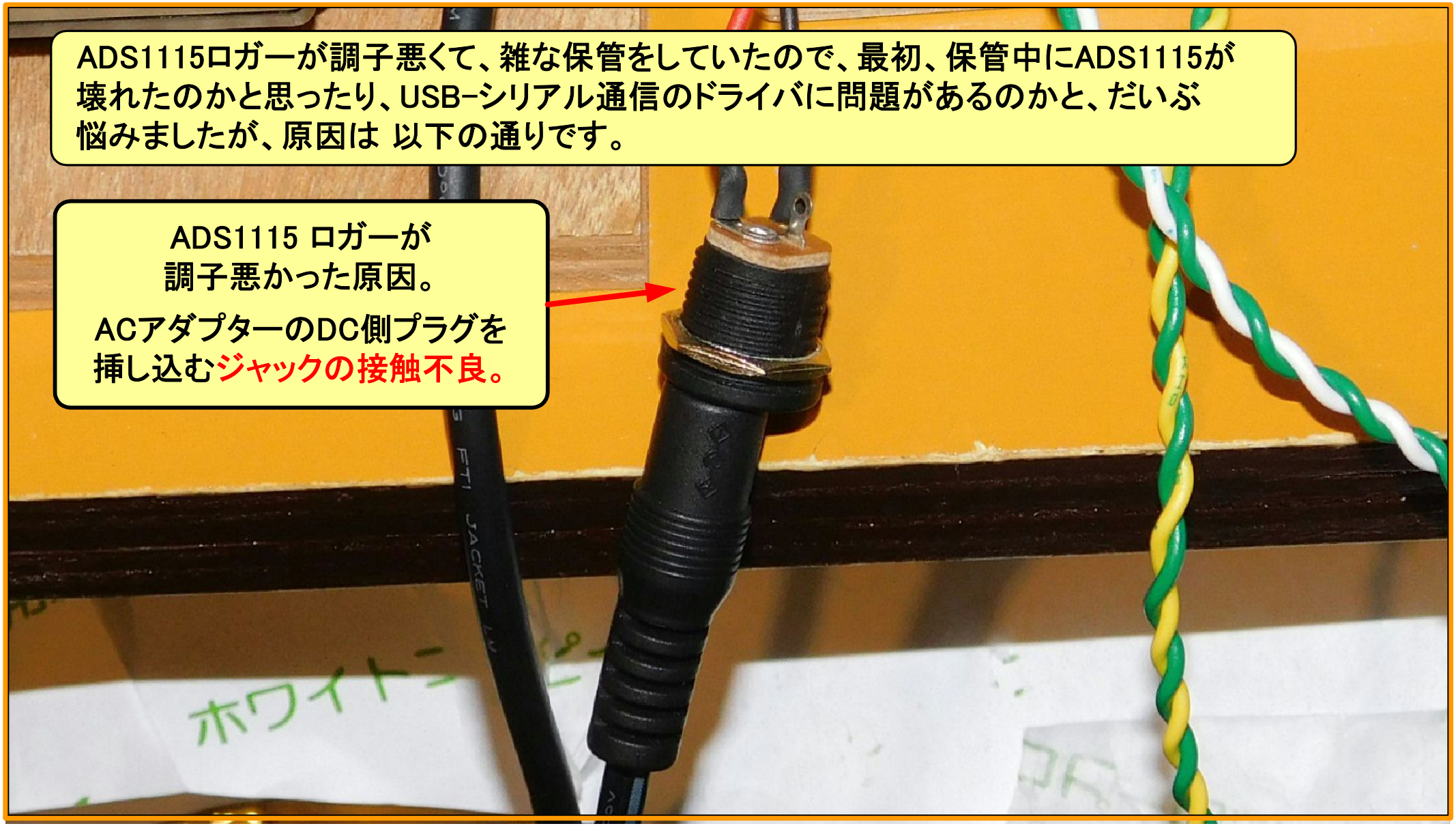
これだけ電文を用意すれば、
今回の用途には 十分です。

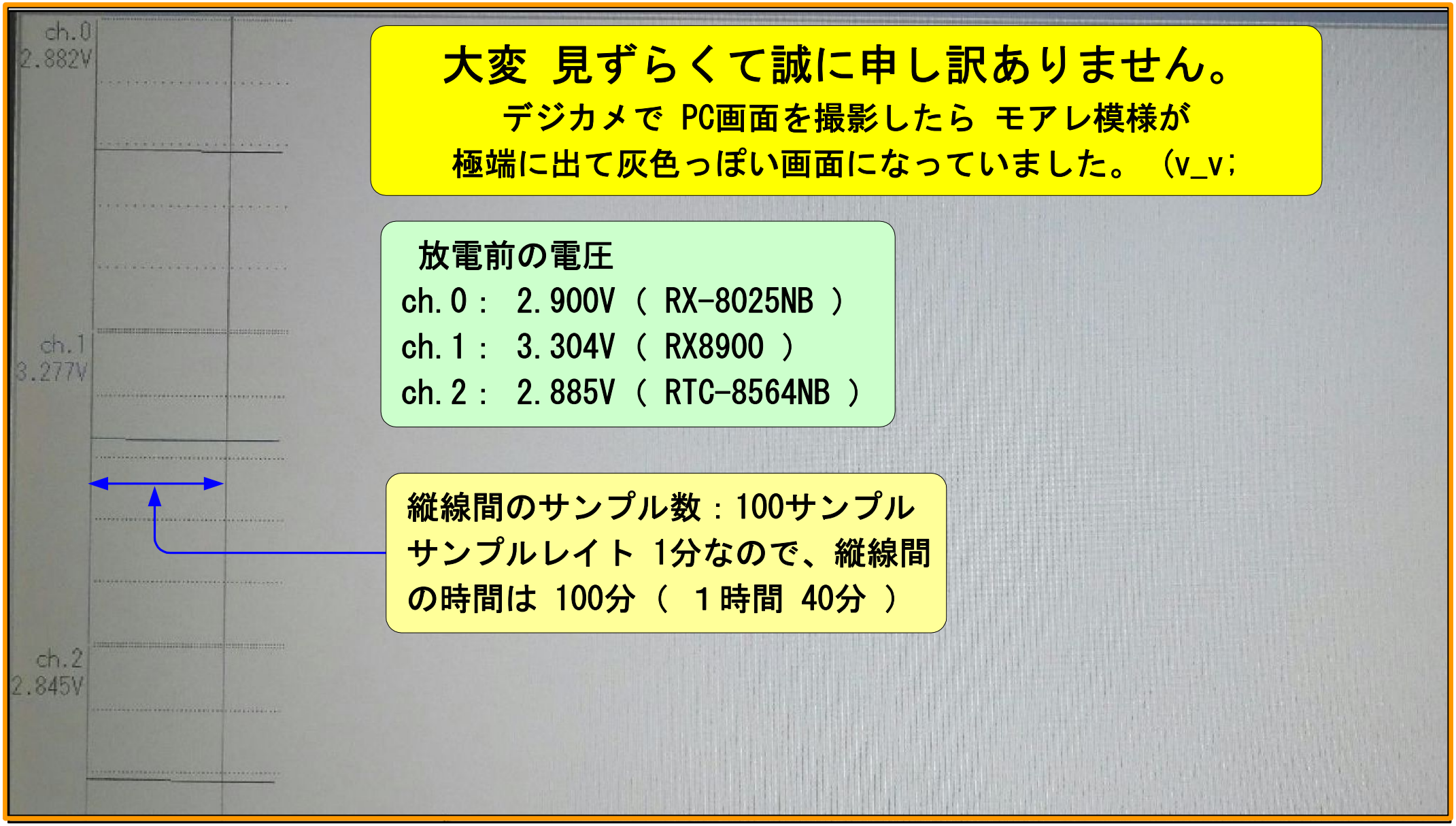
RTC-8564用として
マイコンプログラムに
追加します。

RTCのスーパーキャパシタ 放電試験　卓上の試験機材の様子




RTCのスーパーキャパシタ 放電試験(24時間を予定)
今回の 3つの RTC基板
放電前、電源 3.3Vで 充電を 行う。
以前作成したADS1115を用いた
ロガーアナログ回路基板
ch. 0 : RX-8025NB
ch. 1 : RX8900
ch. 2 : RTC-8564NB


ADS1115ロガーが調子悪くて、雑な保管をしていたので、最初、保管中にADS1115が壊れたのかと思ったり、USB-シリアル通信のドライバに問題があるのかと、だいぶ悩みましたが、原因は 以下の通りです。
ADS1115 ロガーが
調子悪かった原因。
ACアダプターのDC側プラグを
挿し込むジャックの接触不良。

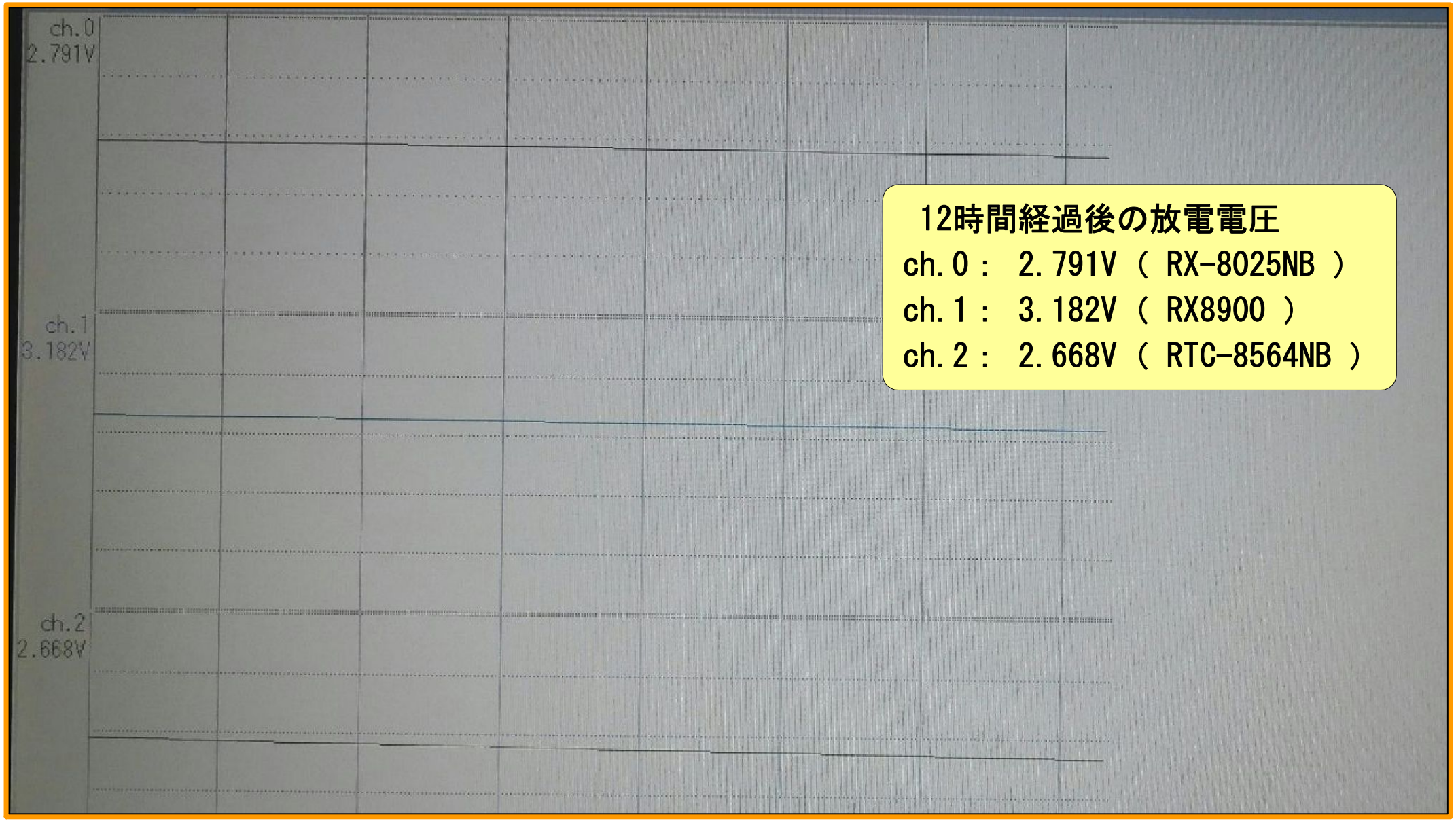


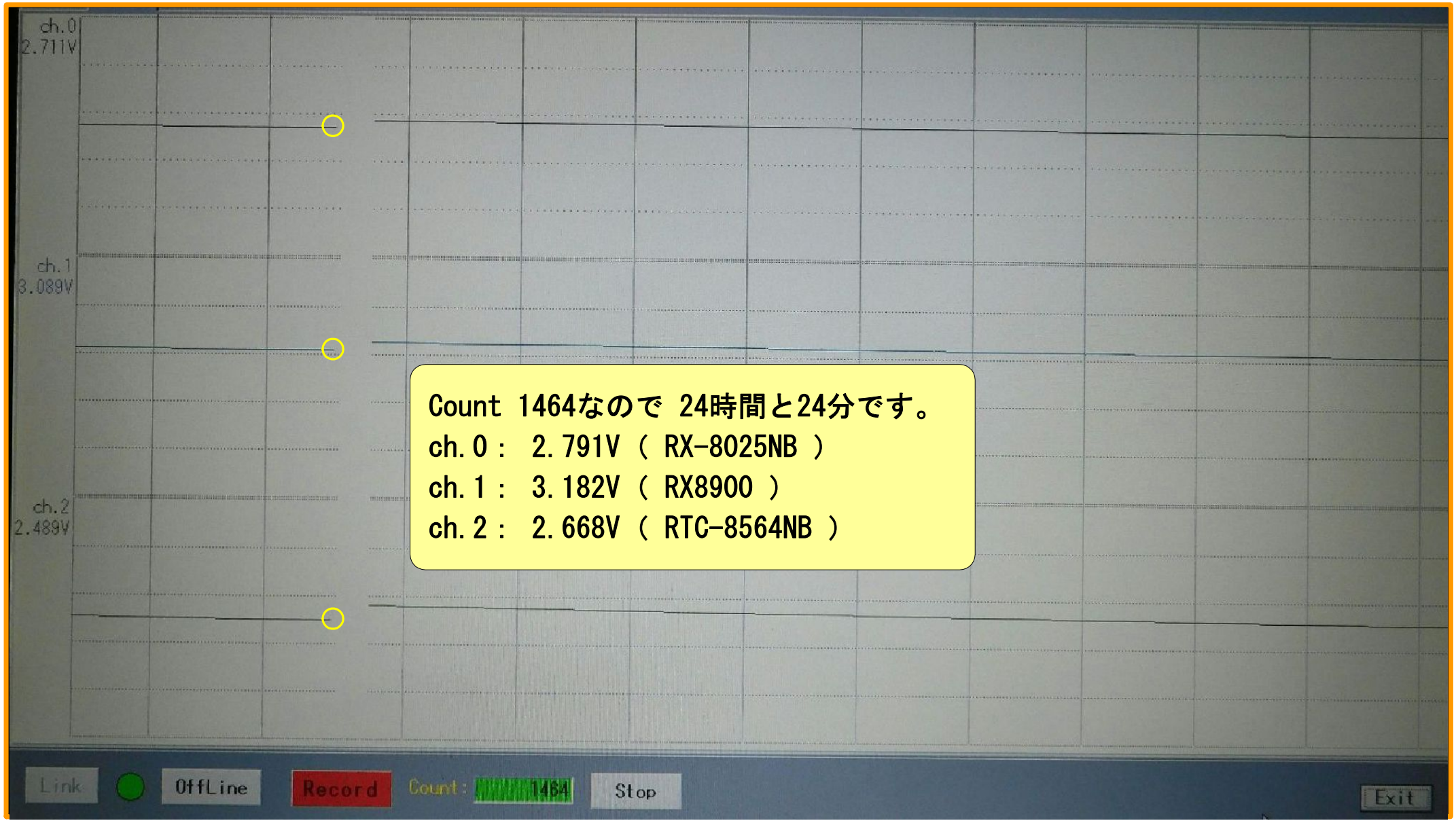
大変 見ずらくて誠に申し訳ありません。

デジカメで PC画面を撮影したら モアレ模様が
極端に出て灰色っぽい画面になっていました。 (v_v;

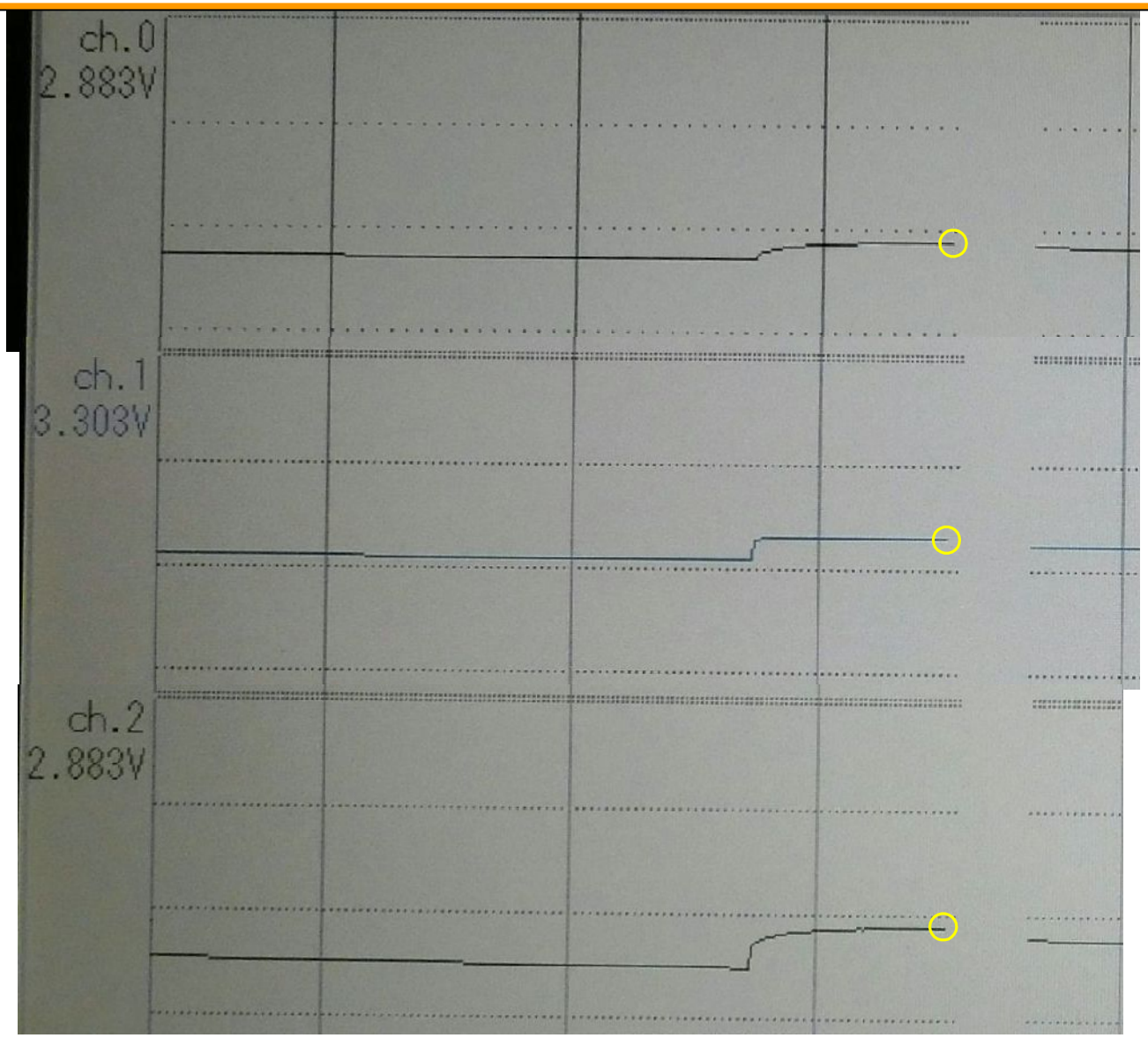
放電前の電圧
ch.0： 2.900V （ RX-8025NB ）
ch.1： 3.304V （ RX8900 ）
ch.2： 2.885V （ RTC-8564NB ）

縦線間のサンプル数：100サンプル
サンプルレイト 1分なので、縦線間
の時間は 100分（ １時間 40分 ）


ch.0
2.791V
ch.1
3.182V
ch.2
2.668V
12時間経過後の放電電圧
ch.0： 2.791V （ RX-8025NB ）
ch.1： 3.182V （ RX8900 ）
ch.2： 2.668V （ RTC-8564NB ）


ch.0
2.711V
ch.1
3.089V
ch.2
2.489V
Count 1464なので 24時間と24分です。
ch.0： 2.791V（ RX-8025NB ）
ch.1： 3.182V（ RX8900 ）
ch.2： 2.668V（ RTC-8564NB ）
Link
OffLine
Record
Count:
1464
Stop
Exit


ch.0
2.883V
ch.1
3.303V
ch.2
2.883V
縦の赤線のタイミングで、3.3Vの電源を
再度、投入しています。
ch.0： 2.883V（ RX-8025NB ）
ch.1： 3.303V（ RX8900 ）
ch.2： 2.883V（ RTC-8564NB ）
Link
OffLine
Record
Count:
1557
Stop
Exit



充電開始した時のスーパーキャパシタの電圧変化のグラフ:

この中で ch.1の RX8900が、他と比べ、非常に早く立ち上がっている。 これは、充電に関わる電流制限抵抗 100Ωが、2個あり充電時は、2パラ接続の 50Ωとなっていたため充電速度が速くなっていたと思われる。

ch.0の RX-8025 は、立ち上がりは緩やかで高さもそう高くならない。

ch.2の RTC-8564は、立ち上がりの高さが、やや高い。 この件については、次で説明します。

3種類のRTC／測定結果

放電開始時と、24時間後の終了時の電圧

単位 [V]

型式	開始時	終了時	電圧降下
RX-8025	2.900	2.791	0.109
RS8900	3.304	3.182	0.122
RTC-8564	2.885	2.668	0.217

充電開始時と、終了時の電圧

単位 [V]

型式	開始時	終了時	電圧差
RX-8025	2.791	2.883	0.092
RS8900	3.182	3.303	0.121
RTC-8564	2.668	2.883	0.215

放電途中のデータを、1460サンプルほど収録しましたが、今回は、最初と最後のデータの比較のみ行います。

当初、想定していた通りになった部分と、実際に実験して、あれっと思う事がありました。

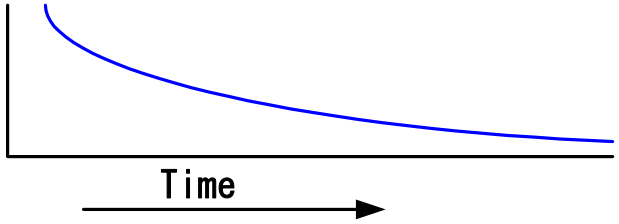
放電に関わる事として、RX-8025の電圧降下 0.109V と、RX8900の電圧降下 0.122Vは、RX8900の方が多少電流を消費するだろうなと思っていたので、カタログスペックで予想した通りでした。 逆に RTC-8564は、カタログスペックで、一番消費電流が少ないはずなのに電圧降下が、一番大きい。？ この電圧降下が他より大きかったので、充電時も、電圧差が一番大きくなっています。 それとは別に、RX-8025は、充電にやや時間がかかっており、電圧降下 0.109Vに対し、充電時の電圧差が0.092Vで、 充電が いまいちという感じです。

これは、スーパーキャパシタの内部抵抗にバラツキがあるのかもしれません。

今回の測定データで 動作保持期間を推測する

メーカーカタログの計時保持電圧範囲と、今回の 24時間の電圧降下値で、今回の3機種において、スーパーキャパシタ1Fの放電で、何日ぐらい持たせる事が出来るかを推測してみます。

本来の放電特性は、電圧が降下するにつれ負荷の消費電流も減るので、カーブの傾斜が、寝てきます。



今回は、仮に リニアに電圧が下がっていくものとして簡易的に計算してみます。実際に持たせる事が出来る時間より、やや短い時間になると思います。

という事で、各RTCの 計時保持電圧範囲の下限値を、確認します。

RX-8025: 1.15V
RX8900: 1.6V（ 温度補償動作下限: 2.0V ）
RTC-8564: 1.0V
で、電源電圧は、3.3Vで、計算してみます。

RX-8025: （ 3.3 − 1.15 ）/ 0.109 = 19.7 日
RX8900: （ 3.3 − 1.6 ）/ 0.122 = 13.9 日
RTC-8564:（ 3.3 − 1.0 ）/ 0.215 = 10.7 日
RX8900: 温度補償動作下限:
（ 3.3 − 2.0 ）/ 0.122 = 10.6 日

最近のミドルレンジ以上のマイコンは、3.3Vが主流ですが、5Vで使用する場合は、3.3 の 所を 5.0 に変えて計算すれば、大雑把な日数は、割り出せると思います。 最近スーパーキャパシタも3.3V仕様で、5Fなどの容量の大きな物も、出回っています。

各RTCを再度比較

① 時計精度:　　誤差

1番　RX8900　　（ 0.000000762 ）

2番　RX-8025　（ 0.000003479 ）

3番　RTC-8564（ 0.000007934 ）

（ FC-7150 ユニバーサルカウンタで計測 ）

② 3.3V 1Fキャパシタによる バックアップ日数

1番　RX-8025　　19.7 日

2番　RX8900　　13.9 日

3番　RTC-8564　　10.7 日

③ RX8900は、逆流防止ダイオードが、内蔵されているので、外部に付ける必要が ない。

電流制限抵抗は必要だが、周辺回路はすっきりする。　残り2つには、逆流防止ダイオードが必要。 0.7V VDDが 下がる。

0.7V電圧が下がる対応策を考えるのが面倒。

上記、対応策の詳細は、前の動画 参照の事。

④ ソフト開発、RTCアクセスの開発に関して

今回のRTCは、それぞれに細かい所に違いがあるが、大雑把な考え方は同じ。

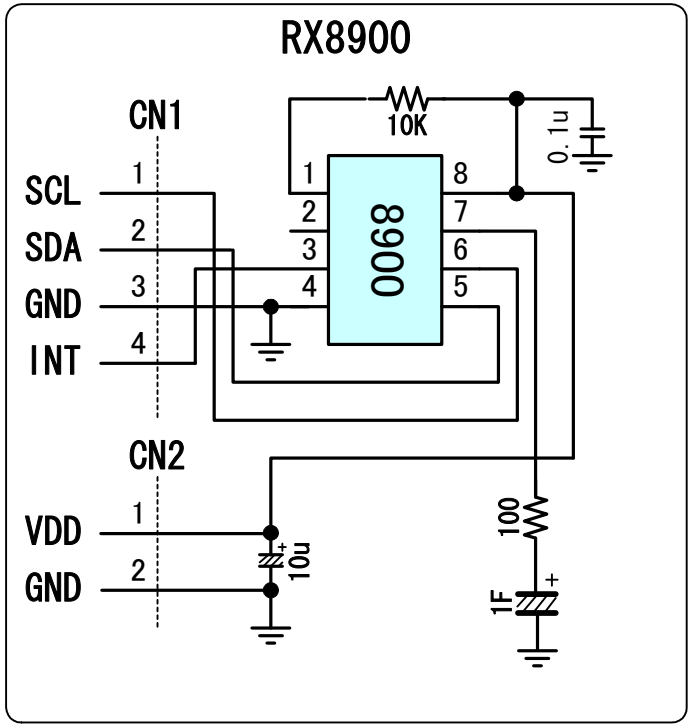
今回は、最低限の使い方しかしてないので3つの RTCに関しては、ソフトは同程度のボリュームと思う。

RX8900は、温度補償回路が入っているので季節による安定した高精度を求める用途では、大きなメリットと考える。 メーカーカタログでは、 $\pm 3.4 \times 10^{-6}$／-40℃ ～ 85℃（月差 9秒相当）

⑤ 価格は、RX-8045は、450円、RX8900は、500円、RTC-8564は、500円であまり価格差がないので、高性能をお安くという事で、私としては　RX8900を お勧めします。

RX8900／周辺回路の再検討

最初に、検討後の 最終版 回路図を示します。



VDD端子と、VBAT端子の機能を生かす事で回路は、スッキリしたと思います。

VDD端子8ピンから、FOE端子1ピンへ、以前は、直接配線してましたが、RX-8025NBにて、ショート事故をやらかしたので、VDDと、FOE間に 10KΩを入れました。

この、RX8900の内部回路で VDD 8ピンと、VBAT 7ピンの間に、逆流防止ダイオードと Pch MOSFETの スイッチが 入ってます。 よって、VDD供給時、スイッチONする事で、逆流防止ダイオードの 両端をショートする事になり、0.7Vの電圧降下を 防ぎます。

これにより、I2Cの SCLと SDAのプルアップ抵抗ホット側を、そのまま マイコンのVDDに接続できるので、回路的にシンプルに出来ます。

そしてスイッチが ONしている間 VDD端子の電圧でスーパーキャパシタを充電できるメリットもあります。これは、3.3V電源のシステムで有利に働きます。

セイコーエプソンさん、Pch スイッチを 内蔵したのがバックアップ回路の ミソですね。 残り2つの RTCに外付けで Pch MOSFETを付ければ 外付け部品は、増えますが、同様の事が、出来るかもしれません。