

# Bio Sensing Software Platform

## ECG, PPG, SpO<sub>2</sub> の同時測定制御例

### 要旨

本アプリケーションノートは、プログラマブル計装アンプ(以下、PGA と略します。)および $\Delta \Sigma$  A/D コンバータの生体モニタ(ECG(心電図)、PPG(光電式容積脈波)、SpO<sub>2</sub>(経皮的動脈血酸素飽和度))への応用として、RL78/H1D への実装例を説明します。

アナログ差動入力 2 チャンネルの入力機能を持つ PGA 付 24 ビット $\Delta \Sigma$  A/D コンバータを利用して、ECG、PPG、SpO<sub>2</sub>を同時測定することができます。

サンプルコードには、ECG、PPG、SpO<sub>2</sub>それぞれの測定のためのハードウェア設定と測定制御およびデジタルフィルタなどのデータ処理のデモソフトが含まれます。

**本サンプルコードを組み込んだシステムは、医療機器ではありません。**

### 動作確認デバイス

RL78/H1D (R5F11PLG)

動作条件：アナログ差動入力 2 チャンネルの入力機能を持つこと

### 参考ドキュメント

- 心電の R ピーク検出アルゴリズム参考文献  
A Real-Time QRS Detection Algorithm. JIAPU PAN AND WILLIS J. TOMPKINS, 1985  
  
Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database. PATRICK S. HAMILTON AND WILLIS J. TOMPKINS, 1986
- 脈波のピーク検出アルゴリズム参考文献  
A Real-Time Pulse Peak Detection Algorithm for the Photoplethysmogram. Dae-Geun Jang, Sangjun Park, and Minsoo Hahn, 2014
- パルスオキシメータ参考文献  
パルスオキシメータの原理. 日本集中治療医学会雑誌 2016;23:625-31

## 目次

1. 概要	4
1.1 機能	4
1.1.1 ECG 測定機能	4
1.1.2 PPG 測定機能	4
1.1.3 SpO <sub>2</sub> 測定機能	4
1.2 用語	4
1.3 使用上の注意事項／制限事項	5
1.4 動作確認条件	6
1.4.1 環境	6
1.4.2 動作クロック周波数	6
1.4.3 必要メモリサイズ	6
1.5 ファイル構造	6
2. 測定原理	8
2.1 ECG 測定原理と心拍検出方法	8
2.2 PPG 測定原理と脈拍検出方法	9
2.3 SpO <sub>2</sub> 測定原理と算出方法	11
3. システム構成	13
3.1 概要	13
3.1.1 システム構成	13
3.1.2 ソフトウェア構成	13
3.2 ECG 測定回路	14
3.3 PPG/SpO <sub>2</sub> 測定回路	14
3.4 使用するリソース	15
4. ECG, PPG, SpO <sub>2</sub> 同時測定処理	16
4.1 動作概要	16
4.2 同時測定	17
4.2.1 サンプリングパターン	17
4.2.2 サンプリングパターンの詳細	18
4.2.3 $\Delta\Sigma$ A/D コンバータの割り込み処理	20
4.3 ECG 測定	21
4.3.1 ECG 測定フロー	21
4.3.2 ECG 測定ソフトウェア	21
4.3.3 ECG 用フィルタ処理	24
4.3.4 ECG 用心拍計算処理	25
4.4 PPG/SpO <sub>2</sub> 測定	26
4.4.1 PPG/SpO <sub>2</sub> 測定フロー	26
4.4.2 PPG/SpO <sub>2</sub> 測定ソフトウェア	26
4.4.3 PPG/SpO <sub>2</sub> 用フィルタ処理	30
4.4.4 PPG/SpO <sub>2</sub> 用脈拍計算処理	31
5. デジタルフィルタと閾値判定	32
5.1 ローパスフィルタ(LPF)	33

5.1.1	IIR/FIR フィルタ .....	33
5.2	ハイパスフィルタ(HPF) .....	36
5.2.1	IIR/FIR フィルタの遅延値取得 .....	37
5.2.1	移動平均フィルタ .....	38
5.3	微分フィルタ .....	39
5.4	SSF フィルタ .....	40
5.5	移動平均フィルタ .....	40
5.6	閾値判定 .....	41
6.	応用例.....	42
6.1	動作概要 .....	42
6.2	SW 入力による状態遷移 .....	42
6.3	使用するリソース .....	43
6.4	測定アプリケーション.....	43
6.4.1	追加の初期化処理.....	43
6.4.2	ユーザ設定可能パラメータの反映処理 .....	44
6.4.3	測定制御処理とスタンバイ機能処理.....	44
6.5	SW (スイッチ) 入力制御.....	50
6.6	アプリケーション用設定ファイル .....	53
6.6.1	マクロ一覧.....	53
6.6.2	ユーザ設定可能パラメータ用ヘッダファイル .....	57
6.6.3	ユーザ設定可能パラメータ用ソースファイル .....	58
6.7	測定シーケンスとサンプリングパターンテーブルの設定方法.....	61
7.	CG の設定 .....	62
7.1	共通/クロック発生回路 .....	62
7.2	ポート機能.....	63
7.3	タイマ・アレイ・ユニット .....	64
7.4	PGA+ $\Delta\Sigma$ A/D コンバータ .....	67
7.5	アンプ・ユニット .....	69
7.6	D/A コンバータ .....	72
7.7	イベントリンクコントローラ.....	72
7.8	割り込み機能 .....	73
	改訂記録 .....	74

## 1. 概要

### 1.1 機能

#### 1.1.1 ECG 測定機能

一定時間(可変)ごとに ECG 用の電極からの信号を 24 ビット  $\Delta \Sigma$  A/D コンバータで A/D 変換し、ECG を測定できます。

#### 1.1.2 PPG 測定機能

一定時間(可変)ごとに PPG/SpO<sub>2</sub> 用センサの IR LED からの信号を 24 ビット  $\Delta \Sigma$  A/D コンバータで A/D 変換し、PPG を測定できます。

#### 1.1.3 SpO<sub>2</sub> 測定機能

一定時間(可変)ごとに PPG/SpO<sub>2</sub> 用センサの IR LED と RED LED からの信号を 24 ビット  $\Delta \Sigma$  A/D コンバータで A/D 変換し、SpO<sub>2</sub> を測定できます。

### 1.2 用語

以下に用語および略語の定義を示します。

表 1-1 用語および略語の定義

用語/略語	概要
CG	コード生成(Code generator)
ECG	心電図(Electrocardiogram)
IIR	無限インパルス応答(Infinite impulse response)
IR LED	赤外色(Infrared light)発光ダイオード
OSR	オーバーサンプリングレート(Over sampling rate)
PGA	プログラマブル計装アンプ
PPG	光電式容積脈波記録法(Photoplethysmography)
RED LED	赤色(Red light)発光ダイオード
SpO <sub>2</sub>	経皮的動脈血酸素飽和度(Percutaneous oxygen saturation level)

### 1.3 使用上の注意事項／制限事項

#### (1) 必要測定時間と測定サイクル時間について

本サンプルコードでは、ECG, PPG, SpO<sub>2</sub> の測定(サンプリング)パターンを作成できます。初期設定として7パターンを提供しています。また、サンプリングパターンをサンプリングパターンテーブルに登録し、測定順番を設定できます。

サンプリングパターン測定時間に対して、十分大きな測定サイクル時間を設定する必要があります。「4.2.2 サンプリングパターンの詳細」も参照してください。

必要測定時間を以下に示します。

$$\begin{aligned} \text{必要測定時間[us]} &= (\text{PPG/SpO}_2\text{測定時間}) + (\text{ECG 測定時間}) \\ &= (3 \times [\text{SpO}_2\text{の OSR}] \div \text{fin} + 128) + (3 \times [\text{ECG の OSR}] \div \text{fin} + 128) \end{aligned}$$

※ : OSR : オーバーサンプリングレート、fin = 1 (単位 : MHz)

注 : DSADCTL レジスタの DSADSWM ビットが 0 の場合。

DSADSWM ビットが 1 の場合など、詳細は RL78/H1D のハードウェアマニュアルの「16.5.4 24 ビット ΔΣ A/D コンバータを制御するレジスタ」を参照してください。

RL78/H1D のハードウェアマニュアルの「16.5.1 24 ビット ΔΣ A/D コンバータの概要」も参照してください。

測定サイクル周波数を設定する際は、上記の式から算出される測定時間を超える測定サイクル時間になる周波数を設定してください。

$$\text{必要測定時間} < \text{測定サイクル時間} = \left( \frac{1}{\text{測定サイクル周波数[Hz]}} \right)$$

サンプリングパターン毎の必要測定時間の詳細は「4.2.2 サンプリングパターンの詳細」を参照してください。

#### (2) バッファ数について

本サンプルコードで設定する以下の移動平均回数はデータの測定サイクル時間によって依存します。

測定サイクル時間を変更した場合は移動平均回数を見直してください。

移動平均回数 (本サンプルコードでのマクロ名)

- MA\_IR\_TAPSIZE\_BASE、
- MA\_RED\_TAPSIZE\_BASE、
- MA\_OFFSET\_TAPSIZE\_BASE、
- MA\_ECG\_TAPSIZE\_BASE、
- MA\_ECG\_TAPSIZE\_DIFF、
- MA\_IR\_BUFF\_NUM\_DIFF

#### (3) サンプリングパターンテーブルへの登録について

心拍数と脈拍数を正しく計算するため、ECG 測定と PPG/SpO<sub>2</sub> 測定を周期的に繰り返す必要があります。そのため、サンプリングパターンを考慮し、測定シーケンスを作成する必要があります。

「4.2.2 サンプリングパターンの詳細」、「6.7 測定シーケンスとサンプリングパターンテーブルの設定方法」を参照し、正しく設定してください。

## 1.4 動作確認条件

### 1.4.1 環境

表 1-2 環境条件

項目	内容
統合開発環境	ルネサス エレクトロニクス製 CS+ for CC V8.01.00
C コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.08.00

### 1.4.2 動作クロック周波数

表 1-3 動作クロック周波数

項目	内容
メイン・システム・クロック	高速オンチップ・オシレータ 24MHz
内蔵フラッシュメモリの動作モード設定	HS (高速メイン) モード

### 1.4.3 必要メモリサイズ

表 1-4 メモリサイズ

使用メモリ	サイズ
ROM	15.5K バイト
RAM	2.5K バイト
最大使用ユーザスタック	80 バイト
最大使用割り込みスタック	18 バイト

注：本サンプルコード全体でのサイズを示します。コンパイルオプションがデフォルト設定時の値です。

## 1.5 ファイル構造

本サンプルコードのファイル構造を以下に示します。

表 1-5 ファイル構造

```

r11an0390jj0100-bsspf-apl
├─r11an0390jj0100-bsspf-apl.pdf
├─workspace
│   └─Project_Source
│       ├──app
│       │   ├──r_bsse_app_ecg.c      ECG 測定用ソースファイル
│       │   ├──r_bsse_app_ecg.h      ECG 測定用ヘッダファイル
│       │   ├──r_bsse_app_spo2.c     PPG/SpO2測定用ソースファイル
│       │   ├──r_bsse_app_spo2.h     PPG/SpO2測定用ヘッダファイル
│       │   ├──r_bsse_ctrl_measure.c 測定制御用ソースファイル
│       │   └─r_bsse_ctrl_measure.h  測定制御用ヘッダファイル
│       ├──auto_define.h
│       ├──Bio_Sensing_Platform.mtpj
│       ├──Bio_Sensing_Platform.rcpe
│       ├──cg_src                    Code Generation 生成ファイル格納フォルダ
│       │   ├──r_cg_amp.c
│       │   ├──r_cg_amp.h
│       │   ├──r_cg_amp_user.c
│       │   └─r_cg_cg.c

```

		└─r_cg_cgc.h		
		└─r_cg_cgc_user.c		
		└─r_cg_dac.c		
		└─r_cg_dac.h		
		└─r_cg_dac_user.c		
		└─r_cg_elc.c		
		└─r_cg_elc.h		
		└─r_cg_elc_user.c		
		└─r_cg_intp.c		
		└─r_cg_intp.h		
		└─r_cg_intp_user.c		
		└─r_cg_macrodriver.h		
		└─r_cg_main.c		
		└─r_cg_pga_dsad.c		
		└─r_cg_pga_dsad.h		
		└─r_cg_pga_dsad_user.c		
		└─r_cg_port.c		
		└─r_cg_port.h		
		└─r_cg_port_user.c		
		└─r_cg_systeminit.c		
		└─r_cg_tau.c		
		└─r_cg_tau.h		
		└─r_cg_tau_user.c		
		└─r_cg_userdefine.h		
		└─config.c	ユーザ設定可能パラメータ用ソースファイル	
		└─config.h	ユーザ設定可能パラメータ用ヘッダファイル	
		└─config_parameters.h		
		└─cstart.asm		
		└─hdwinit.asm		
		└─interface		
			└─filter.c	フィルタ処理用ヘッダファイル
			└─filter.h	フィルタ処理用ソースファイル
			└─if_dsad.c	24ビット $\Delta\Sigma$ A/Dコンバータ制御用ソースファイル
			└─if_dsad.h	24ビット $\Delta\Sigma$ A/Dコンバータ制御用ヘッダファイル
			└─if_ecg.c	ECG演算用ソースファイル
			└─if_ecg.h	ECG演算用ヘッダファイル
			└─if_spo2.c	SpO <sub>2</sub> 演算用ソースファイル
			└─if_spo2.h	SpO <sub>2</sub> 演算用ヘッダファイル
			└─if_sw.c	SW入力制御用ソースファイル
			└─if_sw.h	SW入力制御用ヘッダファイル
			└─if_userdefine.h	
		└─iodefine.h		
		└─stkinit.asm		

## 2. 測定原理

### 2.1 ECG 測定原理と心拍検出方法

ECG とは心臓が動く際の心筋電位をグラフとして記録するものです。本サンプルコードでは、PGA 付き 24 ビット  $\Delta \Sigma$  A/D コンバータを使用し、心筋電位を測定します。

図 2-1 に ECG 波形を示します。ECG には P 波、Q 波、R 波、S 波、T 波という複数の波があります。

ECG 波形のピーク(R 波)を検出し、ピーク間隔時間より心拍を算出します。

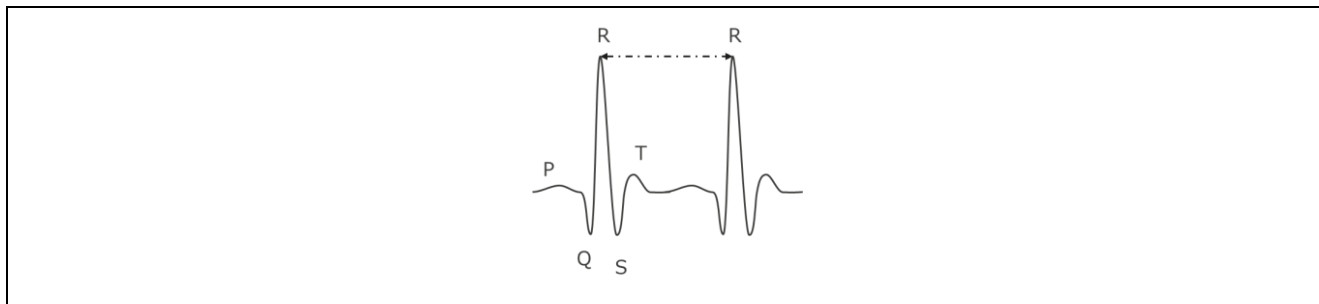


図 2-1 ECG 波形の例

ECG のピーク検出手順は、Jiapu Pan と Willis J. Tompkins による「A Real-Time QRS Detection Algorithm」を基にしています。図 2-2 に ECG 波形のピーク検出処理手順を示します。

測定したデータに対して、ローパスフィルタ(LPF) -> ハイパスフィルタ(HPF) -> 微分フィルタ(Derivative Filter) -> 二乗フィルタ(Squared Filter) -> 移動平均フィルタ(MA Filter)の順で処理し、その結果に対して閾値判定します。閾値を超えた時点をピークと判断します。

本サンプルコードでは、2 秒ごとの移動平均フィルタ処理後の最大値の 0.5 倍(※)の値を閾値とします。

※ECG 波形のピーク検出に用いる閾値の倍率は ECG\_THREXCROSS\_RATE で設定可能です。

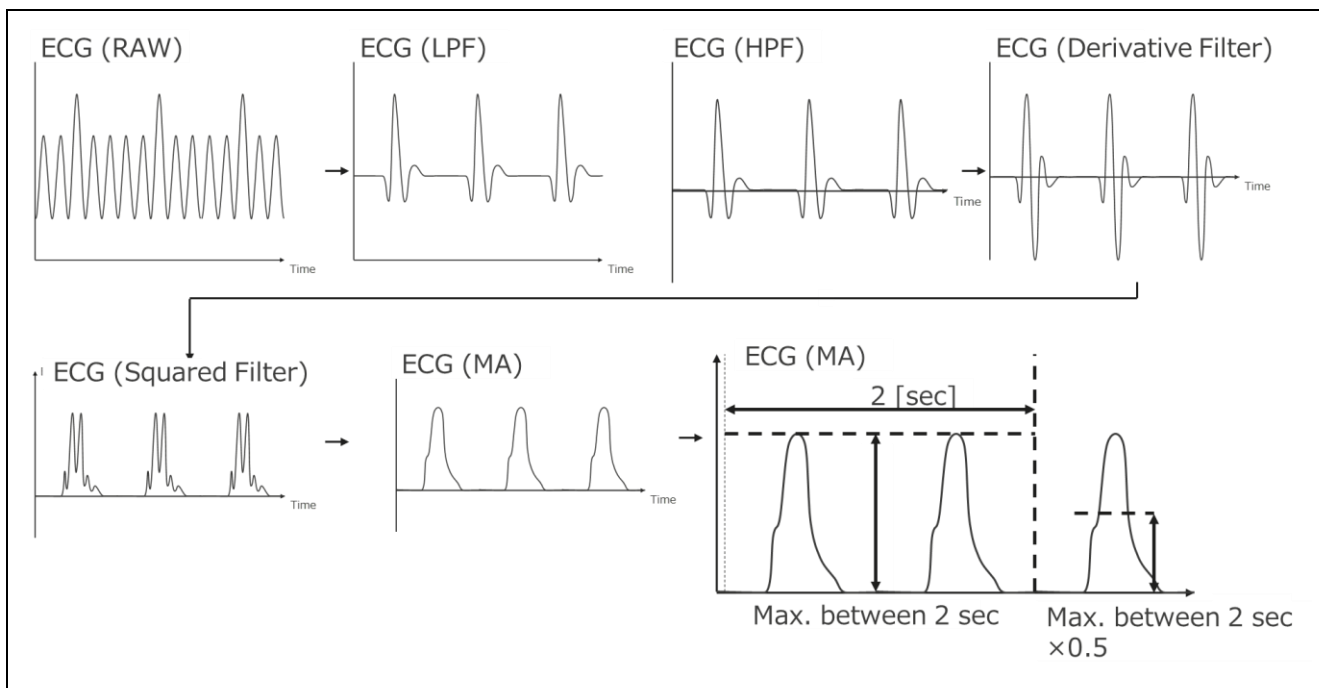


図 2-2 ECG 波形のピーク検出処理手順



ピーク検出後から次のピーク検出までの心拍一拍分の期間内の ECG 測定回数から、心拍一拍分の時間を求めます。

## 2.2 PPG 測定原理と脈拍検出方法

PPG とは Photoplethysmography (光電式容積脈波記録法) の略で、光学的技法により心拍数の変化に対応する動脈および毛細血管の血液量の変化を測定することによって心拍に伴う脈波の情報を記録するものです。

PPG の検出方法として透過式と反射式があります。赤外発光ダイオード(IR LED)を光源とし、フォトトランジスタで検出する方法が一般的です。

赤外色光が赤血球のヘモグロビンに選択的に吸収されその他の組織によって反射されることを利用しており、検出される光の量は血液量に依存します。脈拍ごとに計測部を通過する血液量(血管の容積変化)をセンシングする手法を容積脈波測定といいます。

PPG は、経皮的動脈血酸素飽和度(SpO<sub>2</sub>)の測定に利用できます。

以下では反射式検出方法を説明します。図 2-3 は検出方法の概要を示します。

本サンプルコードでは、PGA 付き 24 ビット  $\Delta \Sigma$  A/D コンバータを使用し、反射光電流を測定します。

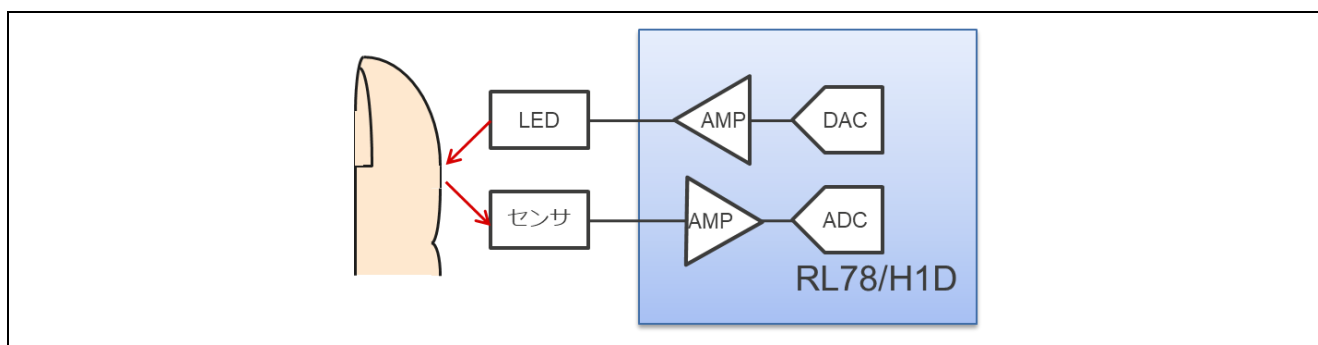


図 2-3 検出方法の概要

反射式では脈波と逆転した値となって ADC によって取得されます。図 2-4 に示すように反射光のピーク(脈波のボトム)を検出し脈拍を算出します。

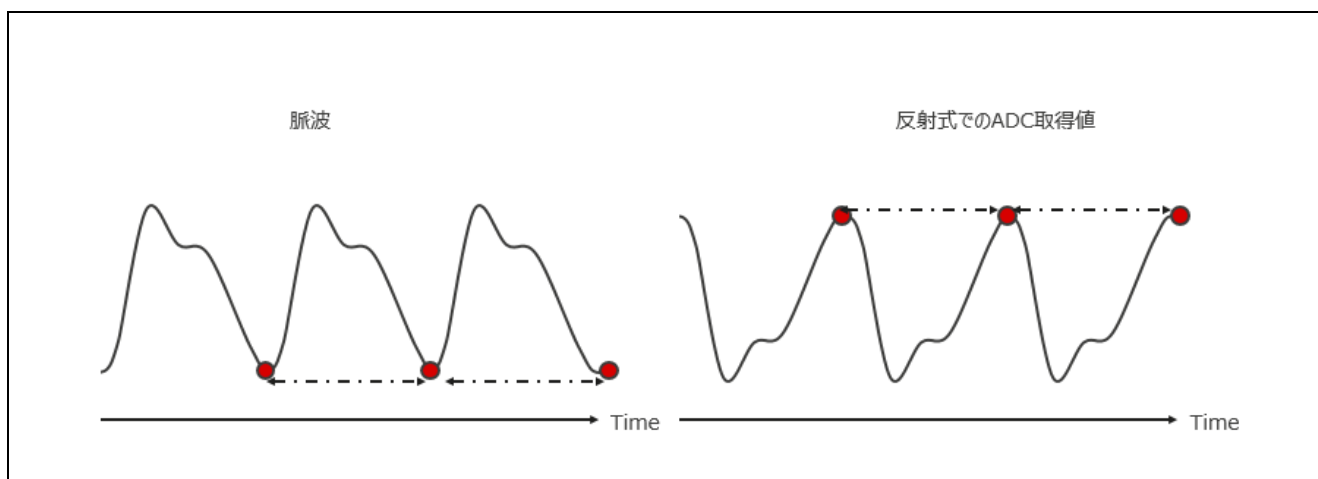


図 2-4 反射式での脈波測定

PPG のピーク検出手順は、Dae-Geun Jang らによる「A Real-Time Pulse Peak Detection Algorithm for the Photoplethysmogram」を基にしています。図 2-5 に PPG 波形のピーク検出処理手順を示します。

測定した PPG 波形(IR LED の反射光)のデータに対して、ローパスフィルタ(LPF) -> ハイパスフィルタ (HPF) -> 微分フィルタ(Derivative Filter) -> SSF フィルタ(Slope Sum Function Filter) -> 移動平均フィルタ (MA Filter)の順に処理し、その結果に対して閾値判定します。閾値を超えた時点をピークと判断します。

本サンプルコードでは、2 秒ごとの移動平均フィルタ処理後の最小値の 0.6 倍 (※) の値を閾値とします。

※PPG のピーク検出に用いる閾値の倍率は SPO2\_THREXCROSS\_RATE で設定可能です。

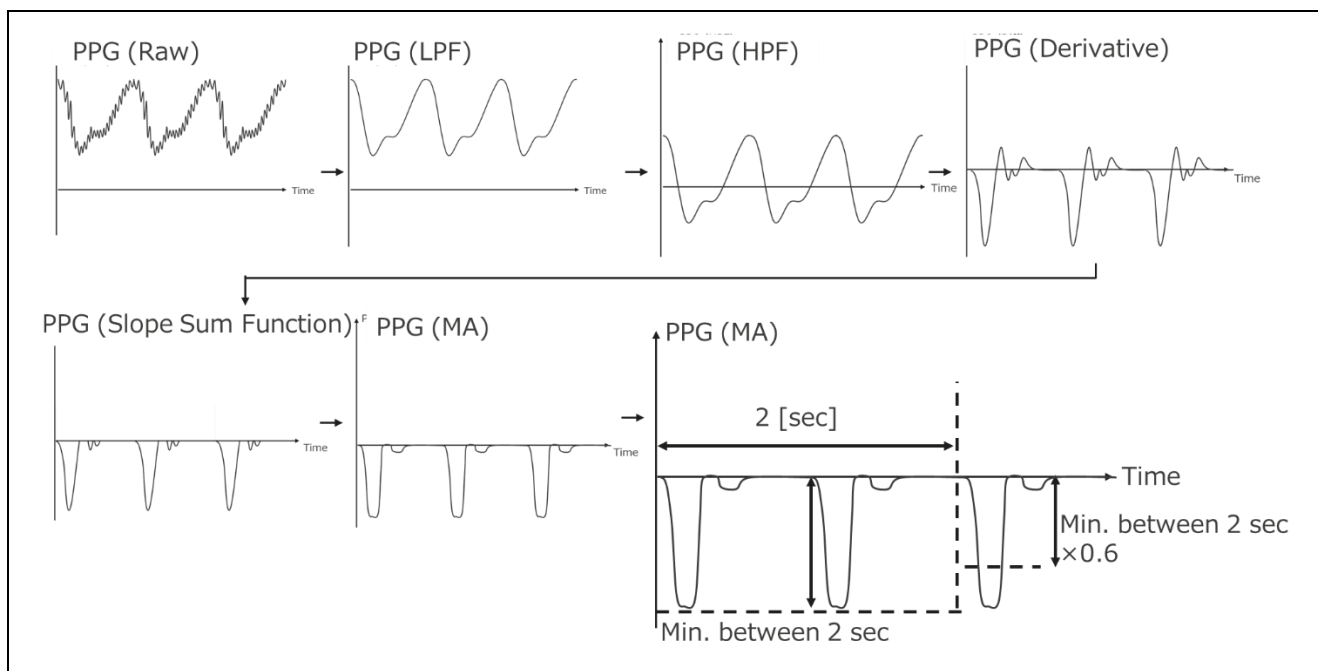


図 2-5 PPG 波形のピーク検出処理手順

ピーク検出後から次のピーク検出までの脈拍一拍分の期間内の IR LED 測定回数から、脈拍一拍分の時間を求めます。

### 2.3 SpO<sub>2</sub> 測定原理と算出方法

SpO<sub>2</sub>は経皮的動脈血酸素飽和度を示します。SpO<sub>2</sub>測定のために、赤色発光ダイオード(RED LED)と赤外線発光ダイオード(IR LED)を利用します。

図 2-6 に示すように、酸素と結合したヘモグロビン(HbO<sub>2</sub>)と酸素と結合していないヘモグロビン(Hb)において、赤色光での吸光度はHbよりHbO<sub>2</sub>の方が著しく低いのに対して、赤外線光での吸光度はHbとHbO<sub>2</sub>ではさほど変わりません。

参考文献：パルスオキシメータの原理. 日本集中治療医学会雑誌 2016;23:625-31

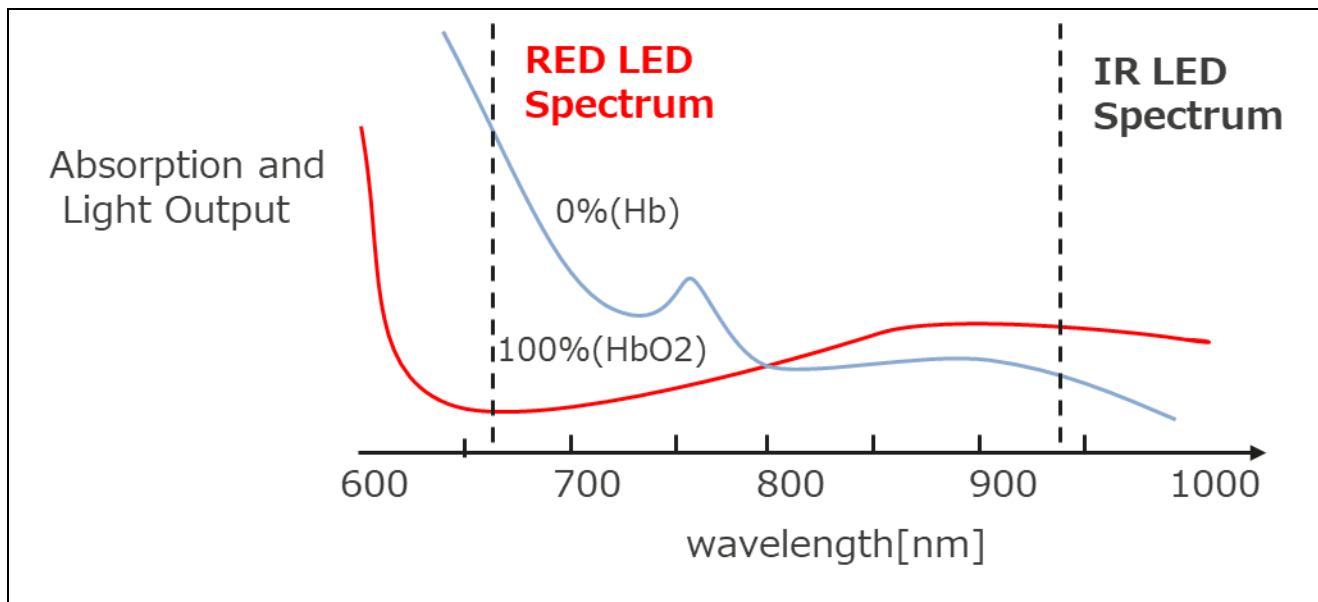


図 2-6 波長と Hb,HbO<sub>2</sub> の吸光度のイメージ

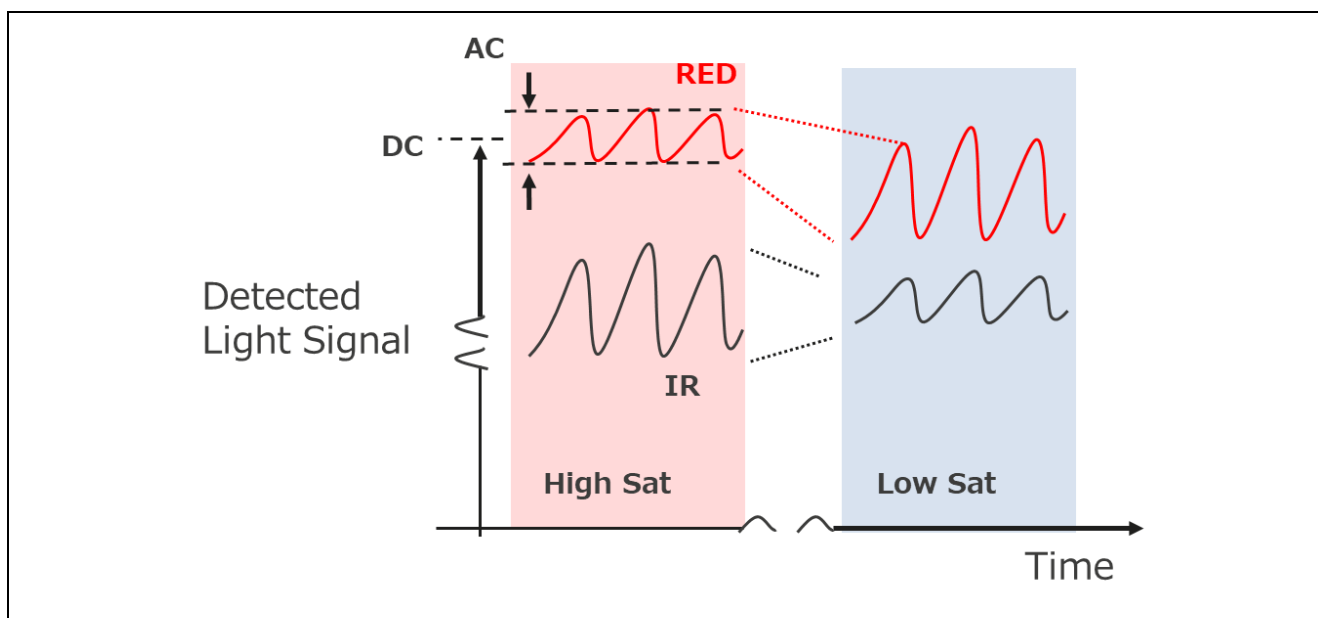


図 2-7 SpO<sub>2</sub>の変化に伴う赤色光、赤外線光の変化のイメージ

上記のことから、光センサが受け取る赤色光と赤外色光の比率が分かれば、SpO<sub>2</sub>を算出することができます。SpO<sub>2</sub>の算出式を以下に記します。

IR\_AC = IR LED の HPF 後の最大値 - IR LED の HPF 後の最小値

RED\_AC = RED LED の HPF 後の最大値 - RED LED の HPF 後の最小値

IR\_DC = IR LED のピーク値検出時の移動平均フィルタ結果

RED\_DC = RED LED のピーク値検出時の移動平均フィルタ結果

$$R = \frac{RED\_AC}{RED\_DC} / \frac{IR\_AC}{IR\_DC}$$

$$SpO_2(\%) = (A \times R + B) \times 100$$

※A, B : センサキャリブレーション用の係数。SPO2\_CALC\_COEF\_A、SPO2\_CALC\_COEF\_B で設定可能

### 3. システム構成

#### 3.1 概要

##### 3.1.1 システム構成

図 3-1 に本サンプルコードのシステム構成を示します。

電極を接続した ECG センサと、RED LED/IR LED/光センサで構成される SpO<sub>2</sub> センサと、測定開始/終了用制御用 SW（スイッチ）を制御します。

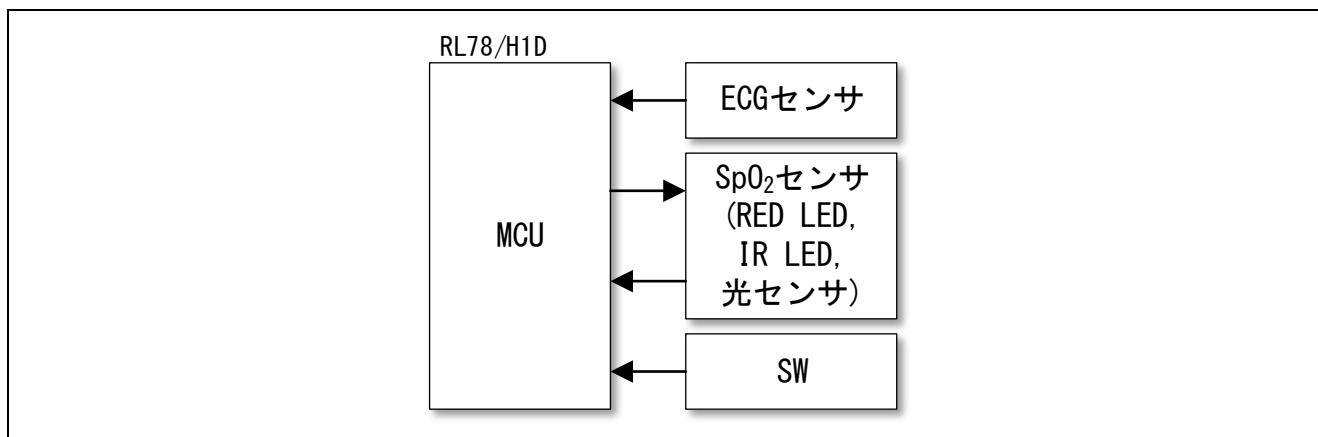


図 3-1 システム構成

##### 3.1.2 ソフトウェア構成

図 3-2 に本サンプルコードのソフトウェア構成を示します。

ECG 測定用 API、PPG/SpO<sub>2</sub> 測定用 API、ECG/PPG/SpO<sub>2</sub> 測定用フィルタ処理ソフトウェアを提供します。

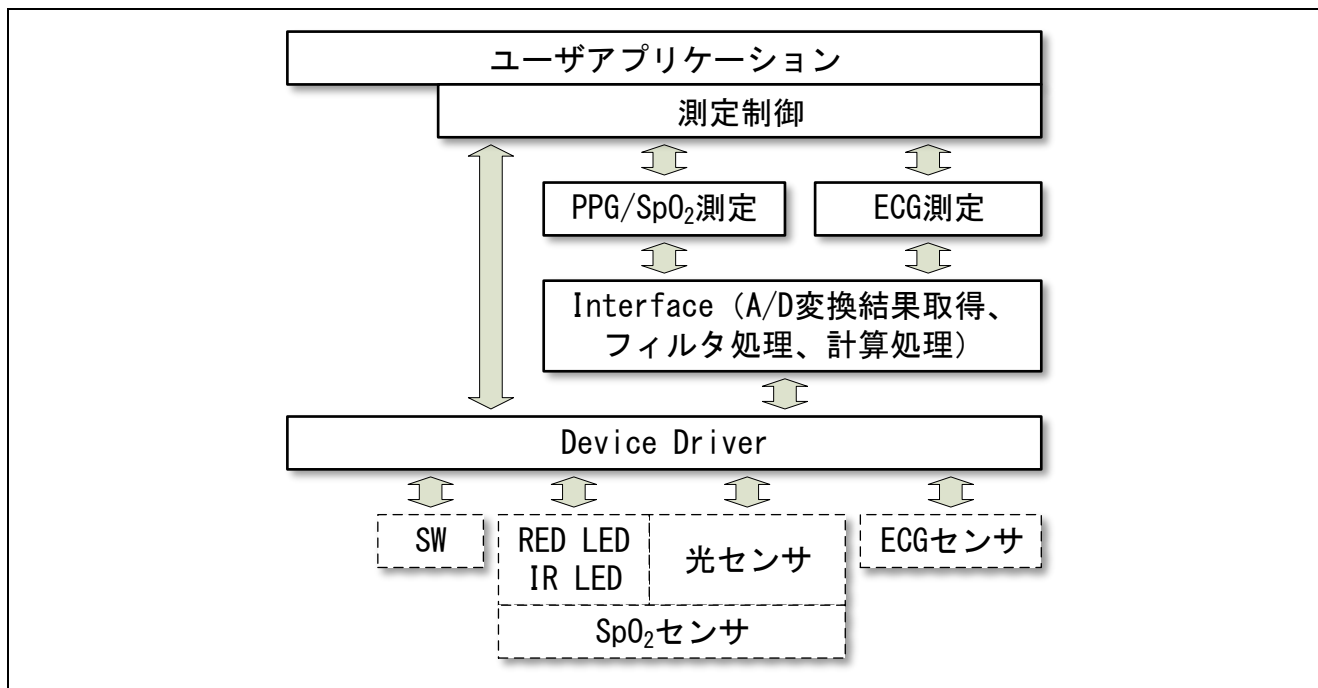


図 3-2 ソフトウェア構成

### 3.2 ECG 測定回路

図 3-3 に ECG 測定で使用する外付け回路例と RL78/H1D の内部接続を示します。

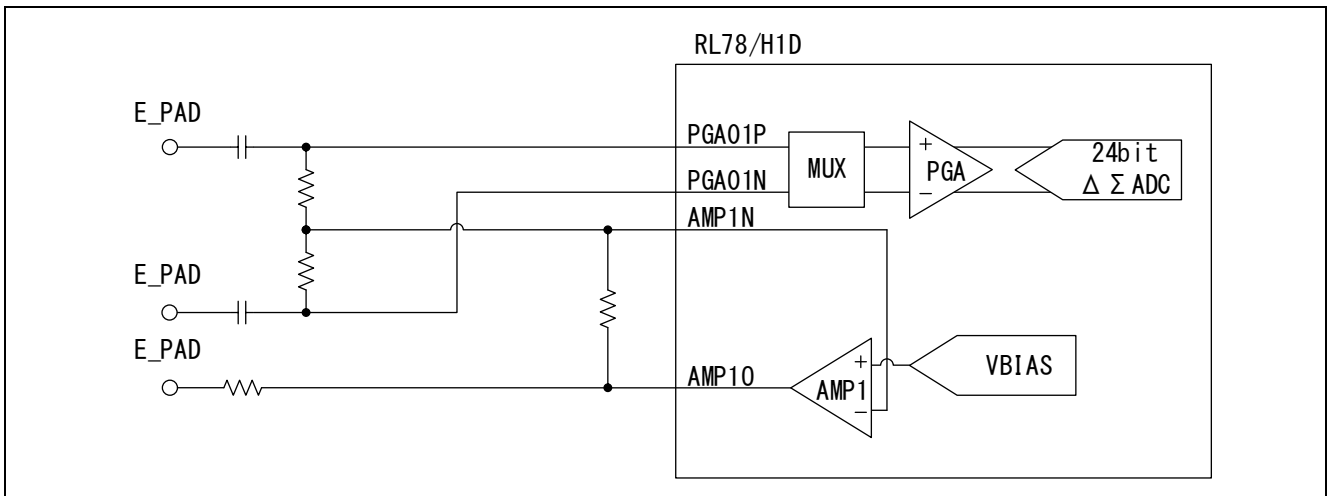


図 3-3 ECG 測定回路例

### 3.3 PPG/SpO<sub>2</sub> 測定回路

図 3-4 に PPG/SpO<sub>2</sub> 測定で使用する外付け回路例と RL78/H1D の内部接続を示します。

SpO<sub>2</sub> センサとして新日本無線社製 NJL5501R を使用した場合の回路例です。

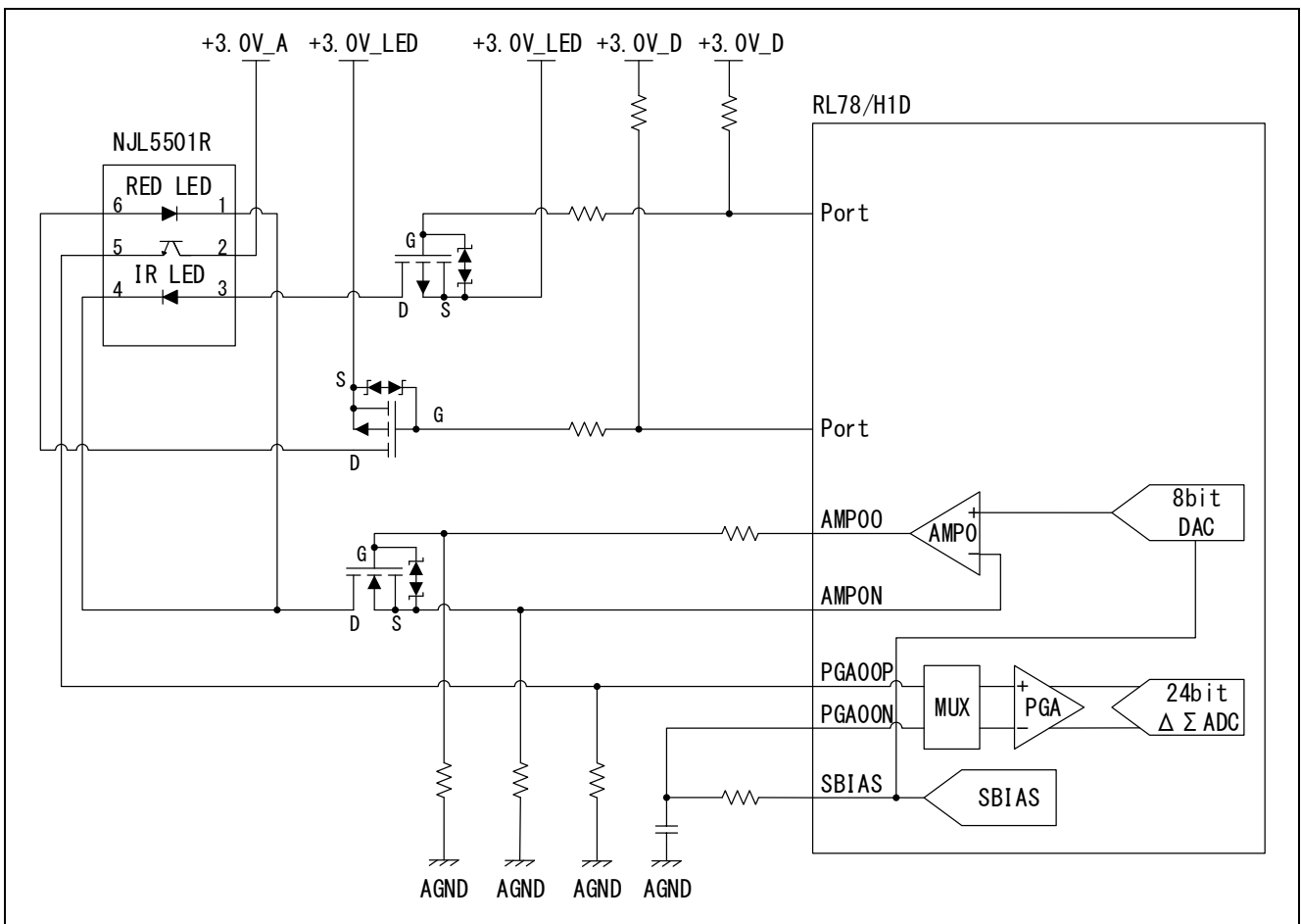


図 3-4 PPG/SpO<sub>2</sub> 測定回路例

### 3.4 使用するリソース

以下に、ECG, PPG, SpO<sub>2</sub> 同時測定制御で使用するリソースを示します。

同時測定制御以外の制御で使用するリソースについては、「6.3 使用するリソース」を参照してください。

表 3-1 使用するリソース

リソース	内容
AMP0N	LED 制御端子
AMP0O	LED 制御端子
AMP1N	ECG 測定用電極端子
AMP1O	ECG 測定用電極端子
PGA00P/PGA00N	PPG/SpO <sub>2</sub> 測定用電極端子
PGA01P/PGA01N	ECG 測定用電極端子
Port(注 1)	RED LED 制御端子
Port(注 2)	IR LED 制御端子
8bit DAC	LED 制御用に使用
ELC(注 3)	$\Delta \Sigma$ ADC の起動用に使用 (TAU チャネルカウント完了イベント)、AMP の起動用に使用 (TAU チャネルカウント完了イベント)
TAU(注 3)	2つのチャネルを使用

注 1：本サンプルコードでは、P01 を割り当てます。

注 3：本サンプルコードでは、P02 を割り当てます。

注 3：本サンプルコードでは、TAU00 と TAU04 を割り当てます。使用方法については、「4.2.2 サンプリングパターンの詳細」を参照してください。

## 4. ECG, PPG, SpO2 同時測定処理

### 4.1 動作概要

以下に本サンプルコード全体の動作の概要を示すフローを示します。

1 測定サイクル内で、PPG/SpO<sub>2</sub>->ECG の順に測定します。

下図の赤枠示す部分が、ECG と PPG/SpO<sub>2</sub> の測定処理を示します。1 回の AUTOSCAN で実施するセンシングの組み合わせをサンプリングパターンとして設定できます。複数のサンプリングパターンをサンプリングパターンテーブルに登録することで、複雑な測定シーケンスを実現できます。

サンプリングパターンテーブルへの登録方法は、「6.6.3(1) サンプリングパターンテーブル設定」を参照してください。

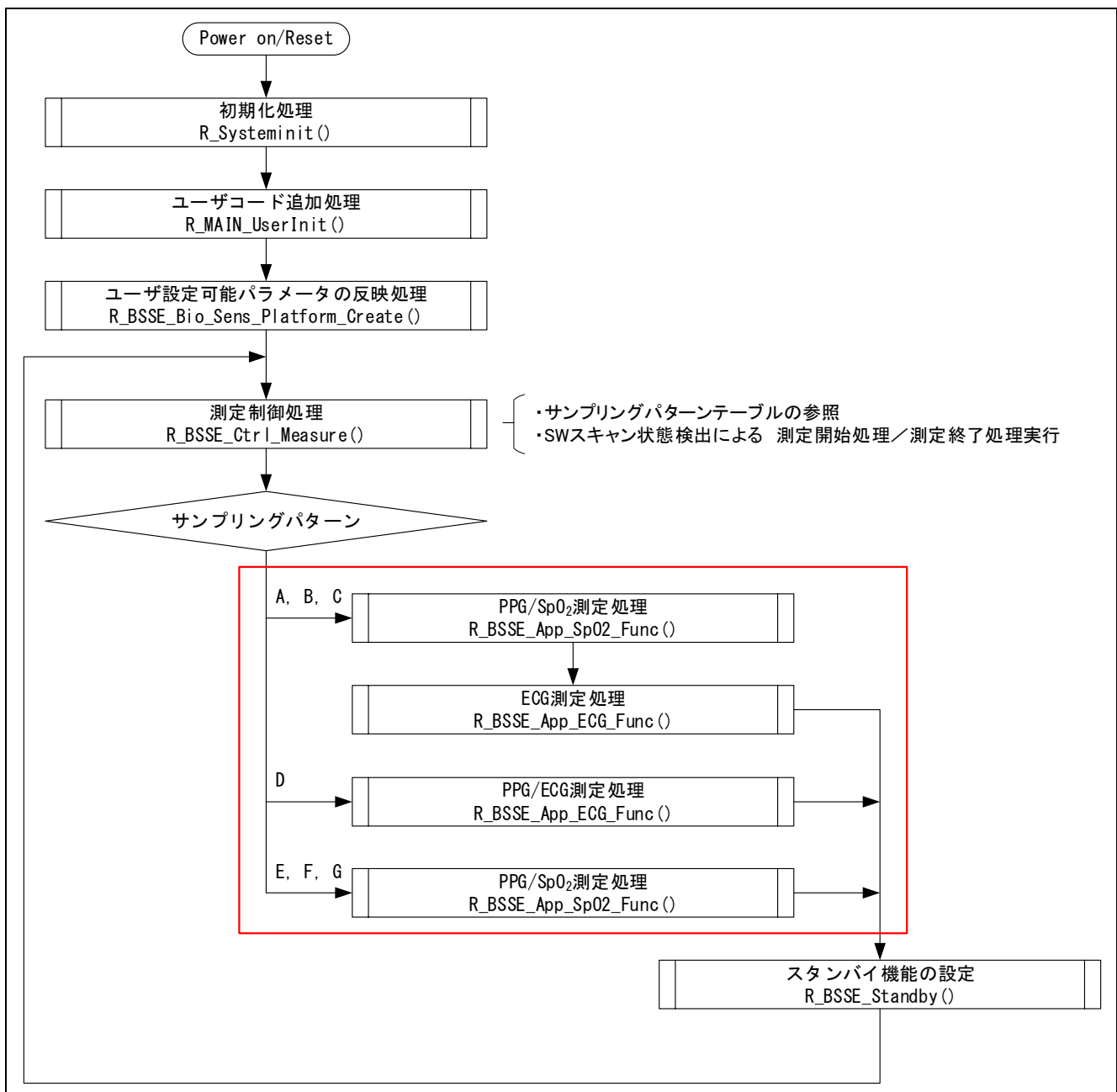


図 4-1 本サンプルコード全体の動作フロー



## 4.2 同時測定

### 4.2.1 サンプルングパターン

本サンプルコードでは AUTOSCAN による複数のセンサによる測定(サンプルング)が可能で、それぞれの測定は1回のAD変換サイクル毎に測定の種別を変更することができます。また、複数を組み合わせて指定することも可能です。

表 4-1 に本サンプルコードで設定済のサンプルングパターンを示します。

サンプルングパターンの設定方法については、「6.7 測定シーケンスとサンプルングパターンテーブルの設定方法」を参照してください。

表 4-1 サンプルングパターン

サンプルングパターン	測定内容
A	IR LED + ECG
B	RED LED + ECG
C	Offset (注) + ECG
D	ECG
E	IR LED
F	RED LED
G	Offset (注)

注：Offset は LED 非点灯時での PPG/SpO<sub>2</sub> 測定回路での測定を示します。

## 4.2.2 サンプリングパターンの詳細

図 4-2 に本サンプルコードで提供するセンシングのサンプリングパターンを示します。

なお、測定サイクル時間は測定サイクル周波数(マクロ名 : SAMPLING\_FREQ)によって決定されます。サンプリングパターンで必要な測定時間の合計に対して、大きな測定サイクル時間を設定してください。

「1.3(1) 必要測定時間と測定サイクル時間」を参照してください。

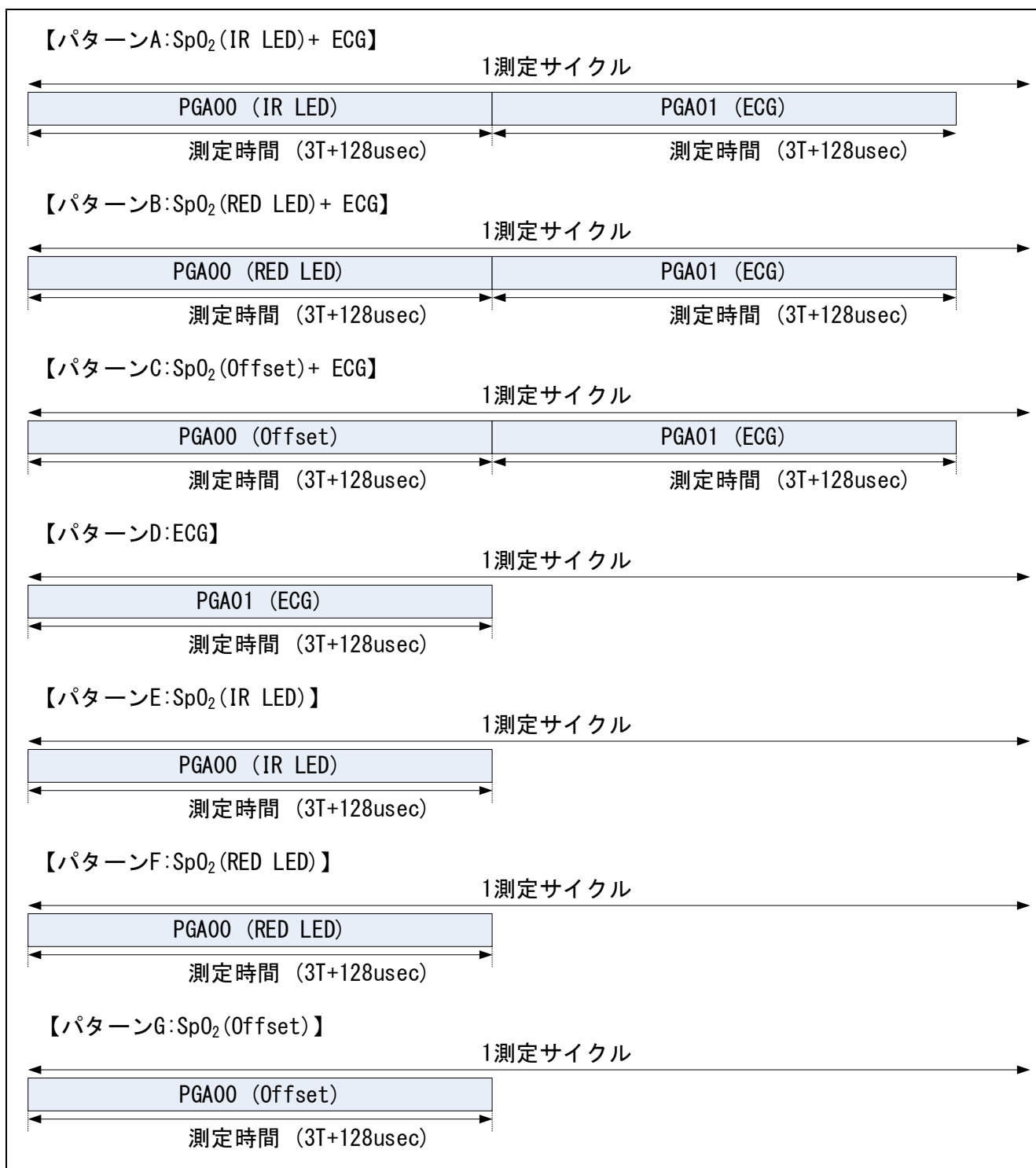


図 4-2 センシングのサンプリングパターン (DSADCTL レジスタの DSADSWM ビットが 0 の場合)

図 4-3 にサンプリングパターンテーブルに A→C→B→C の順で登録されている場合の SpO<sub>2</sub> と ECG の測定タイミングチャート例を示します。図中の TAU00 と TAU04 は本サンプルコードでの設定例です。TAU00 を  $\Delta \Sigma$  ADC 起動のための ELC トリガに、TAU04 を AMP 起動のための ELC トリガに使用します。AMP 起動トリガから  $\Delta \Sigma$  ADC 起動トリガまでの遅延時間（マクロ名：AMP\_DELTA\_SIGMA\_DELAY）を AMP 安定待ち時間に利用します。ハイスピード・モード設定のため、AMP\_DELTA\_SIGMA\_DELAY の初期値として 50us に設定しています。RL78/H1D のハードウェアマニュアルの「17.4.2 アンプ・ユニット制御動作」も参照してください。

図 4-3 の測定パターンの切り替えは 24 ビット  $\Delta \Sigma$  A/D コンバータの A/D スキャン完了割り込み処理で行われます。また、測定データは A/D 変換終了割り込み処理内で RAM に保存されます。

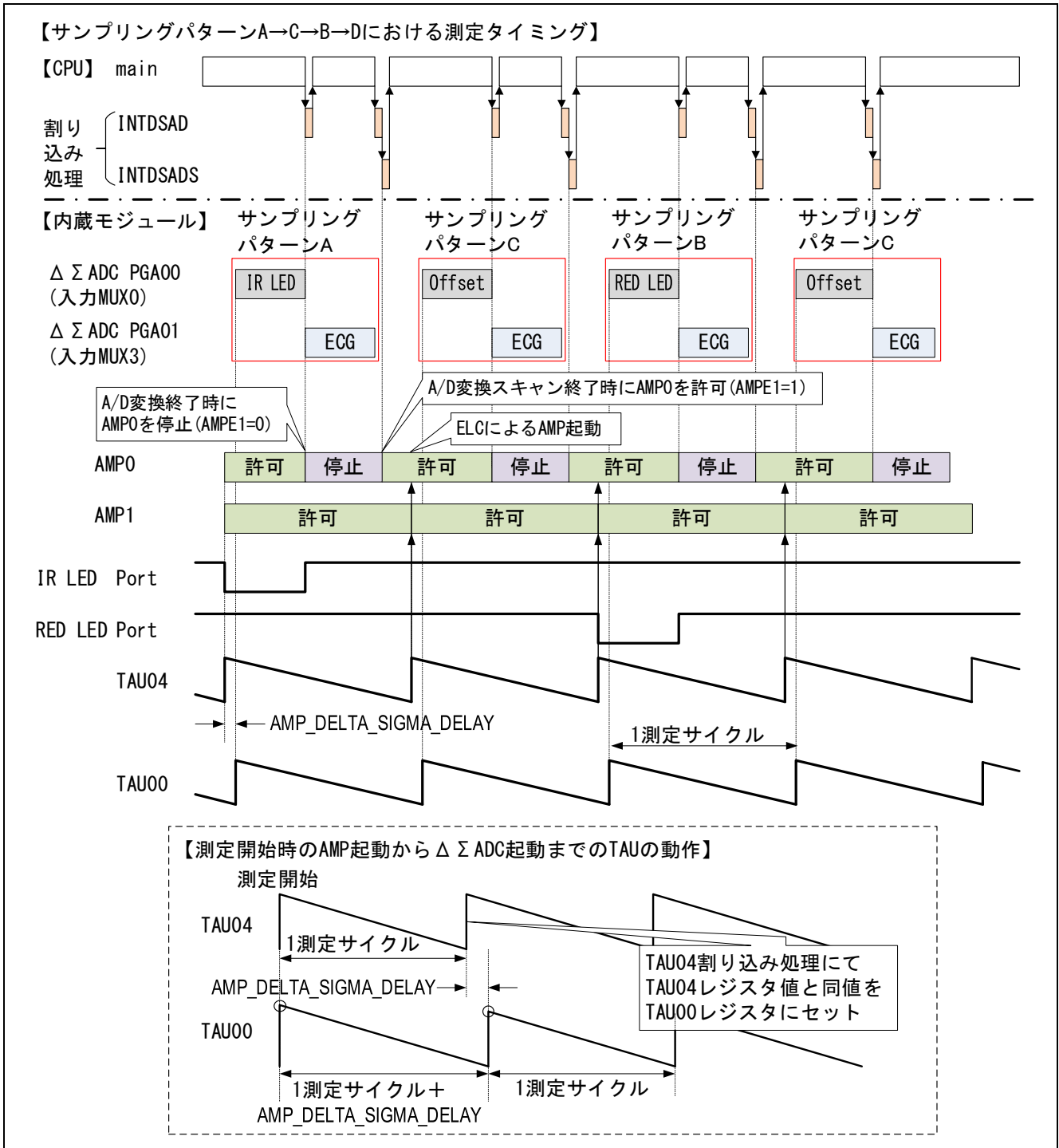


図 4-3 SpO<sub>2</sub> と ECG の測定タイミングチャート例

4.2.3 ΔΣA/Dコンバータの割り込み処理

図 4-3 の測定パターンの切り替えは 24 ビットΔΣA/D コンバータの A/D スキャン完了割り込み処理で行われます。また、測定データは A/D 変換終了割り込み処理内で RAM に保存されます。

図 4-4 に 24 ビットΔΣA/D コンバータの割り込み処理を示します。

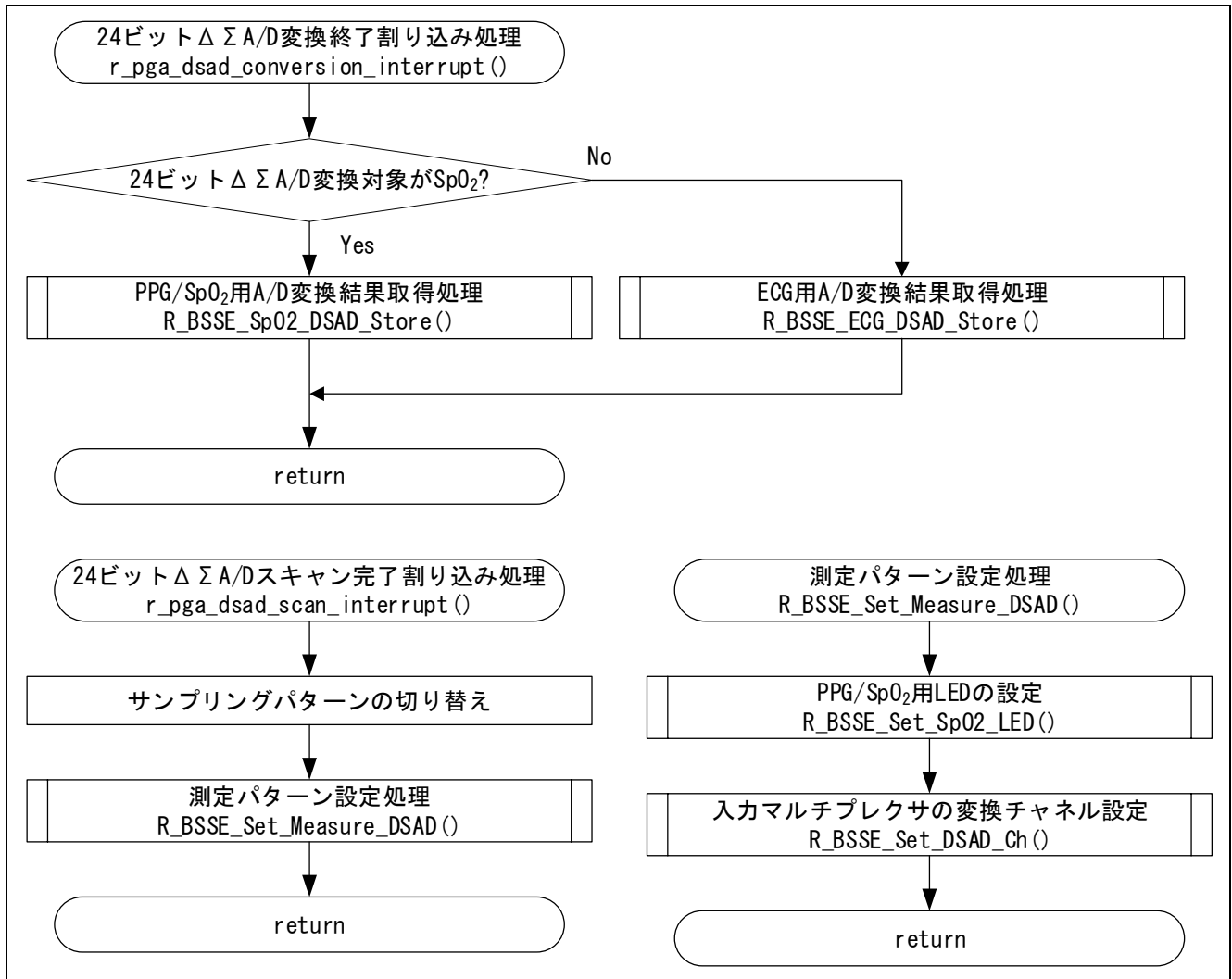


図 4-4 24 ビットΔΣA/D コンバータの割り込み処理

## 4.3 ECG 測定

### 4.3.1 ECG 測定フロー

本サンプルコードでは、PGA 付き 24 ビット  $\Delta \Sigma$  A/D コンバータの入力マルチプレクサ 3 を使用し、簡易的に 2 電極+1 フィードバック電極として両手でボードの電極をつまんで心筋電位を測定します。図 4-5 に ECG 制御フローを示します。

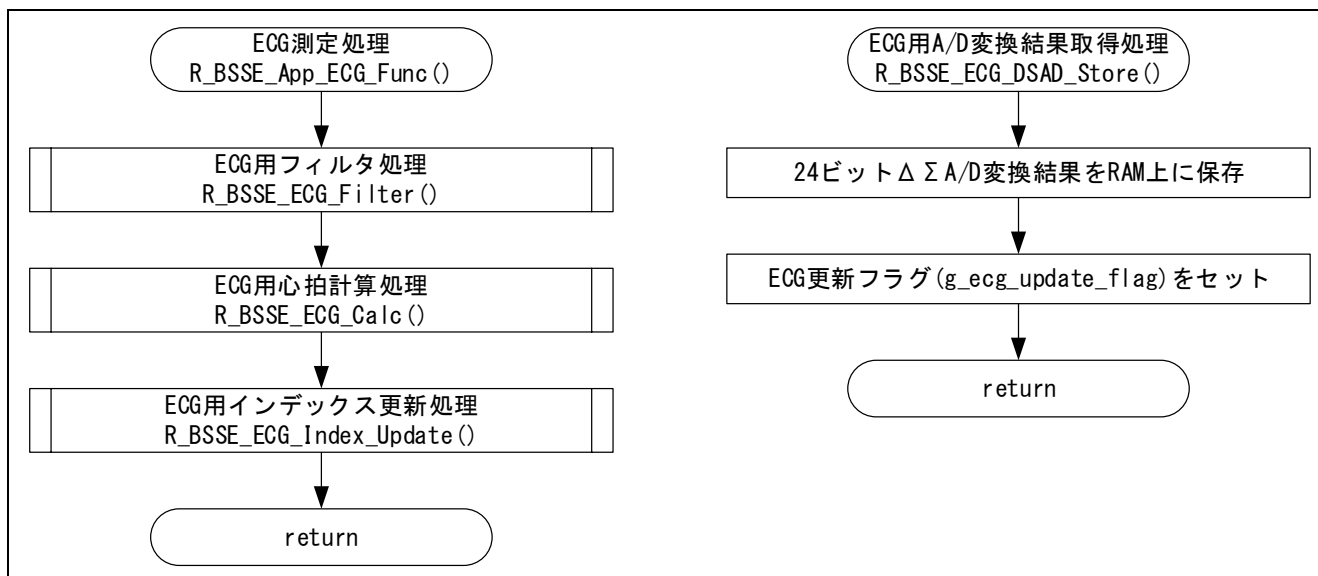


図 4-5 ECG 測定フロー

### 4.3.2 ECG 測定ソフトウェア

#### (1) 列挙子一覧

表 4-2 列挙子一覧

列挙子名	値	備考
e_ecg_status_t	ECG_UPDATE = 0	ECG の更新
	ECG_RESULT_UPDATE = 1	ECG 結果の更新あり

#### (2) 構造体一覧

表 4-3 構造体一覧

構造体名	データ型	変数名	概要
st_thre_cross_t	float *	diff_buf_f	閾値判定の入力値バッファ X[n]のポインタ
	uint16_t *	samp_cnt	測定回数値のポインタ
	float	threshold_m	ピーク検出用の閾値
	e_heart_rate_dir_t *	cross_type	微分データの方向ポインタ

## (3) 変数一覧

表 4-4 変数一覧

データ型	変数名	概要
float	g_ecg_heart_rate	一拍分の時間データ
float	g_ecg_iir_filter [MA_ECG_TAPSIZE_BASE+1]	IIR/FIR フィルタ通過後の ECG データ
float	g_ecg_hpf [HPF_ECG_BUFF_NUM]	ECG 用 HPF 結果
float	g_ecg_iir_coef_kb [IIR_ECG_DEGREE_X]	ECG の係数 $k \times b$ 算出結果を格納 ※オーバフロー対策で事前に算出
float	g_ecg_iir_y_buff [IIR_ECG_DEGREE_Y]	IIR/FIR フィルタ用 Y バッファ (ECG)
float	g_ecg_5diff_squa [MA_ECG_TAPSIZE_DIFF+1]	微分値二乗結果
float	g_ecg_peak	ECG の peak-to-peak のカウント値
float	g_ecg_delay_value	IIR/FIR フィルタ結果の遅延値
int32_t	g_ecg_data	ECG $\Delta \Sigma$ A/D 変換結果
int32_t	g_ecg_iir_x_buff [IIR_ECG_DEGREE_X]	IIR/FIR フィルタ用 X バッファ (ECG)
uint16_t	g_ecg_samp_cnt	ECG サンプリングカウンタ
uint8_t	g_ecg_update_flg	ECG 更新フラグ
uint8_t	g_ecg_init_flg	ECG の測定開始フラグ
uint8_t	g_ecg_meas_cnt	測定パターンテーブル内の ECG 測定パターンの数
st_iir_data_t	g_ecg_iir_data	ECG 用 IIR/FIR フィルタ設定
st_filter_buff_t	g_ecg_iir_result	ECG 用 IIR/FIR フィルタ結果
st_filter_buff_t	g_ecg_ma_data_base	移動平均フィルタ設定 ※IIR/FIR フィルタ結果が対象
st_filter_buff_t	g_ecg_ma_data_diff	移動平均フィルタ設定 ※微分値が対象
st_filter_buff_t	g_ecg_hpf_data	HPF 設定
st_filter_buff_t	g_ecg_5diff_data	5 点微分設定
e_heart_rate_dir_t	g_ecg_dir	ECG の微分値方向
st_thre_cross_t	g_ecg_thre_cross	ECG の閾値判定用構造体

## (4) 関数一覧

表 4-5 関数一覧

ファイル名	関数名	概要
r_bsse_app_ecg.c	R_BSSE_App_ECG_Func	ECG 測定処理
if_ecg.c	R_BSSE_ECG_DSAD_Store	ECG 用 A/D 変換結果取得処理
	R_BSSE_ECG_Filter	ECG 用フィルタ処理
	R_BSSE_ECG_Calc	ECG 心拍計算処理

## (5) 関数仕様

## (a) R\_BSSE\_App\_ECG\_Func

概要	ECG 測定処理
宣言	void R_BSSE_App_ECG_Func (void)
説明	ECG 測定処理を実行します。
引数	なし
戻り値	なし
備考	なし

## (b) R\_BSSE\_ECG\_DSAD\_Store

概要	ECG 用 A/D 変換結果取得処理
宣言	void R_BSSE_ECG_DSAD_Store (int32_t)
説明	24 ビット $\Delta \Sigma$ A/D コンバータの結果を RAM へ格納します。
引数	24 ビット $\Delta \Sigma$ A/D コンバータの結果
戻り値	なし
備考	なし

## (c) R\_BSSE\_ECG\_Filter

概要	ECG 用フィルタ処理
宣言	void R_BSSE_ECG_Filter (void)
説明	ECG 用のフィルタ処理(LPF, HPF 処理)を行います。
引数	なし
戻り値	なし
備考	フィルタ適用後の測定データは g_ecg_iir_result に格納されます。

## (d) R\_BSSE\_ECG\_Calc

概要	ECG 用心拍計算処理
宣言	void R_BSSE_ECG_Calc (void)
説明	5 点微分算出関数、および移動平均フィルタ関数を実行し、閾値判定を行います。閾値を超えた場合は心電の極値と判断し心拍数を算出します。
引数	なし
戻り値	なし
備考	ユーザ設定可能パラメータ ECG_THRE_UPDATE_TIME で閾値の更新時間を変更できます。 算出された心拍数は g_ecg_heart_rate に格納されます。

## 4.3.3 ECG 用フィルタ処理

以下に ECG 用フィルタ処理フローを示します。

HPF 以降のフィルタ処理は、心拍計算処理に含まれます。

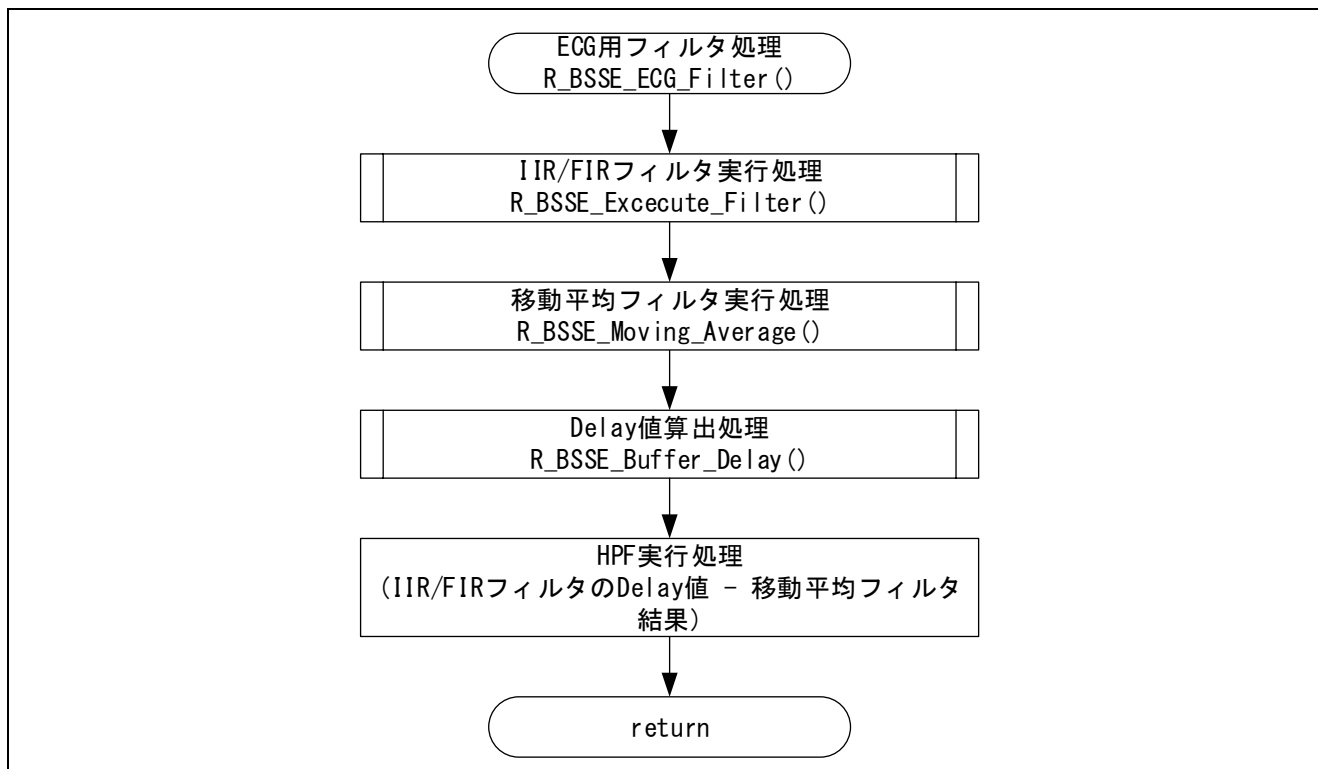


図 4-6 ECG 用フィルタ処理フロー



## 4.3.4 ECG 用心拍計算処理

図 4-7 に心拍計算処理フローを示します。以下に、心拍時間の計算方法を示します。

$$\text{心拍時間} = 1 \text{ 拍中の ECG サンプルカウント} \times \text{ECG 測定間隔}$$

※ ECG 測定間隔については、「6.7 測定シーケンスとサンプリングパターンテーブルの設定方法」を参照してください。

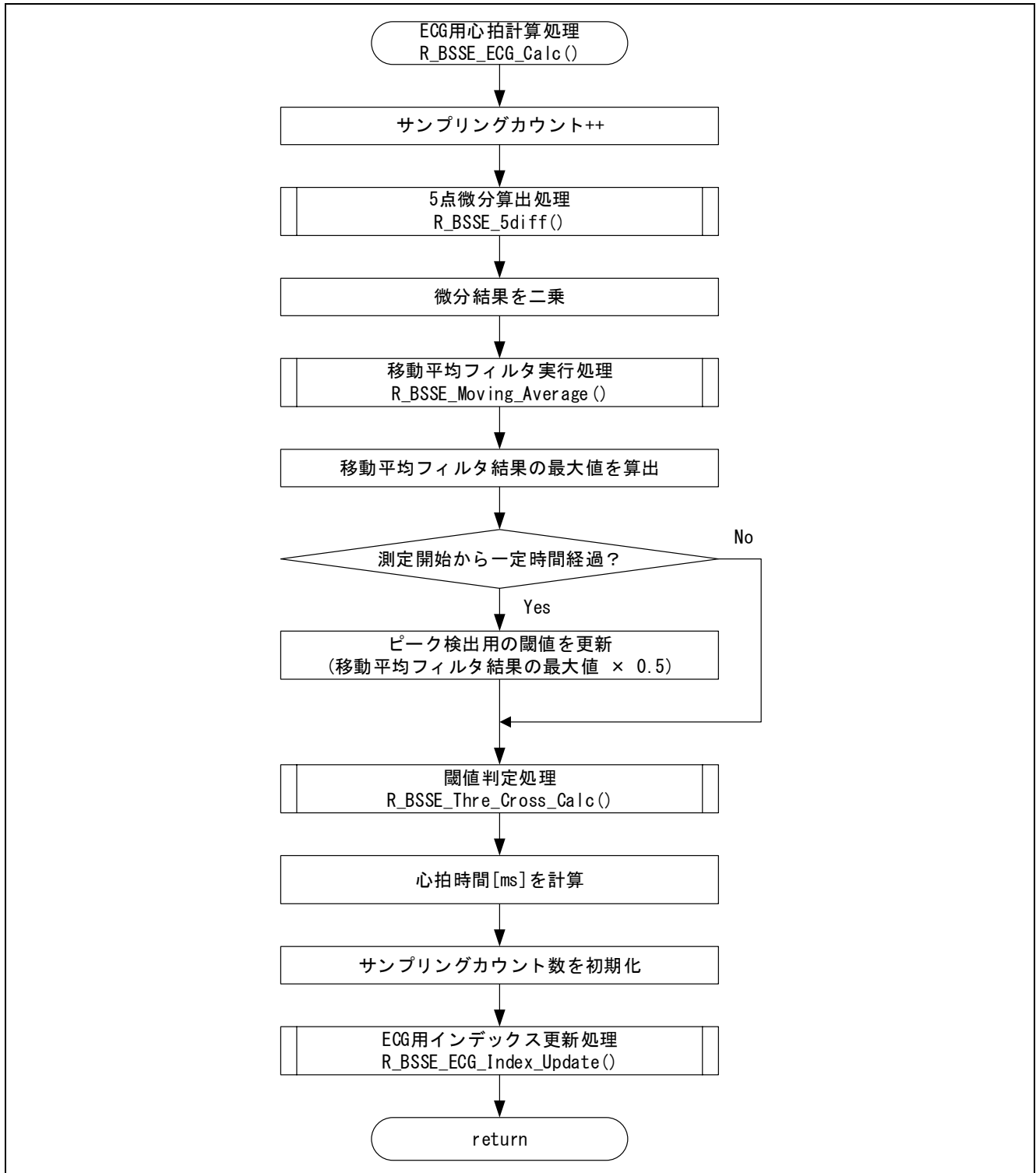


図 4-7 心拍計算処理フロー

## 4.4 PPG/SpO<sub>2</sub> 測定

### 4.4.1 PPG/SpO<sub>2</sub> 測定フロー

本サンプルコードでは、PGA 付き 24 ビット ΔΣ A/D コンバータの入力マルチプレクサ 0 を使用し、RED LED(SpO<sub>2</sub> 測定)と IR LED(PPG/SpO<sub>2</sub> 測定)を制御し、光センサの出力電流を測定します。

24 ビット ΔΣ A/D コンバータの A/D 変換終了割り込み(1 回の A/D 変換ごとの割り込み)が発生した場合、割り込み処理内で A/D 変換結果を RAM に保存します。この A/D 変換終了割り込み処理後、PPG/SpO<sub>2</sub> 状態 (g\_spo2\_status)が更新ありの場合、main 関数内で PPG/SpO<sub>2</sub> 制御処理を開始します。

以下に PPG/SpO<sub>2</sub> 制御フローを示します。

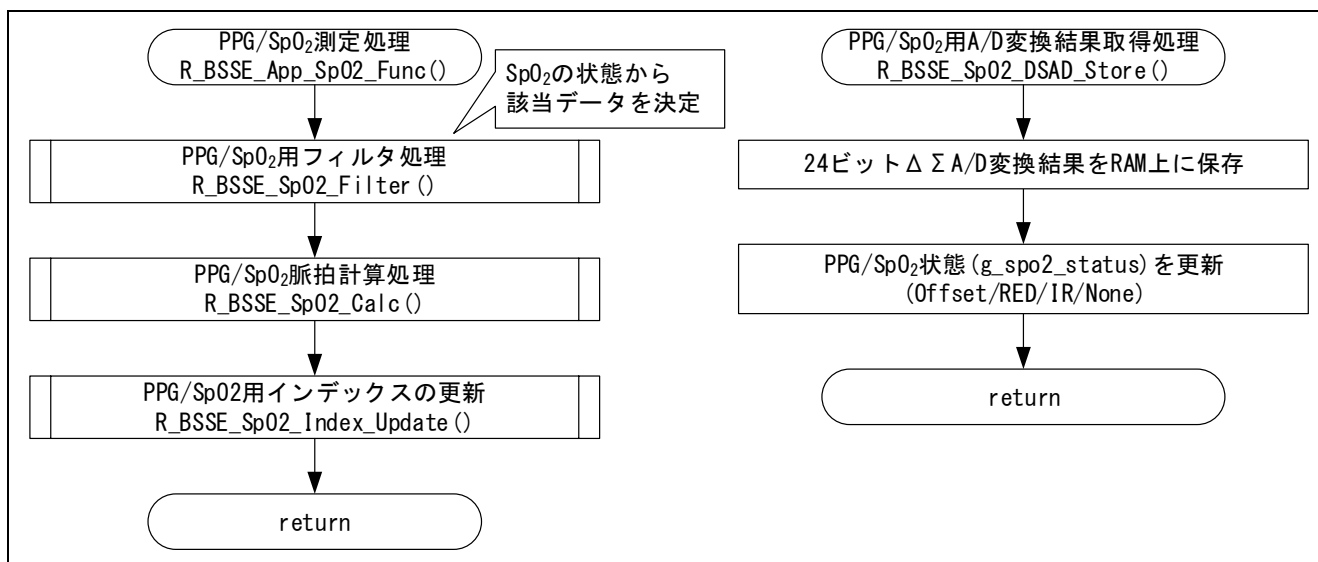


図 4-8 PPG/SpO<sub>2</sub> 測定フロー

### 4.4.2 PPG/SpO<sub>2</sub> 測定ソフトウェア

#### (1) 列挙子一覧

表 4-6 列挙子一覧

列挙子名	値	備考
e_spo2_status_t	SPO2_NONE = 0	SpO <sub>2</sub> の更新なし
	SPO2_OFFSET_UPDATE = 1	SpO <sub>2</sub> のオフセット更新あり
	SPO2_RED_LED_UPDATE = 2	SpO <sub>2</sub> の RED LED 更新あり
	SPO2_IR_LED_UPDATE = 3	SpO <sub>2</sub> の IR LED 更新あり
	SPO2_RESULT_UPDATE = 4	SpO <sub>2</sub> 結果の更新あり

#### (2) 構造体一覧

表 4-7 構造体一覧

構造体名	データ型	変数名	概要
st_thre_cross_t	float *	diff_buf_f	閾値判定の入力値バッファ X[n]のポインタ
	uint16_t *	samp_cnt	測定回数値のポインタ
	float	threshold_m	ピーク検出用の閾値
	e_heart_rate_dir_t *	cross_type	微分データの方向ポインタ

## (3) 変数一覧

表 4-8 変数一覧(1/2)

データ型	変数名	概要
float	g_ppg_heart_rate	PPG 脈拍の結果
float	g_spo2_result	SpO <sub>2</sub> の結果
float	g_offset_hpf [HPF_OFFSET_BUFF_NUM]	HPF 通過後のオフセットデータ
float	g_red_led_hpf [HPF_RED_LED_BUFF_NUM]	HPF 通過後の RED LED データ
float	g_ir_led_hpf [HPF_IR_LED_BUFF_NUM]	HPF 通過後の IR LED データ
float	g_offset_iir_coef_kb [IIR_OFFSET_DEGREE_X]	PPG/SpO <sub>2</sub> オフセット係数 $k \times b_x$ の算出結果を格納 ※オーバフロー対策で事前に算出
float	g_red_led_iir_coef_kb [IIR_RED_LED_DEGREE_X]	PPG/SpO <sub>2</sub> RED LED 係数 $k \times b_x$ の算出結果を格納 ※オーバフロー対策で事前に算出
float	g_ir_led_iir_coef_kb [IIR_IR_LED_DEGREE_X]	PPG/SpO <sub>2</sub> IR LED 係数 $k \times b_x$ の算出結果を格納 ※オーバフロー対策で事前に算出
float	g_offset_iir_y_buff [IIR_SPO2_DEGREE_Y]	IIR/FIR フィルタ用 Y バッファ(オフセット)
float	g_red_led_iir_y_buff [IIR_SPO2_DEGREE_Y]	IIR/FIR フィルタ用 Y バッファ(RED LED)
float	g_ir_led_iir_y_buff [IIR_SPO2_DEGREE_Y]	IIR/FIR フィルタ用 Y バッファ(IR LED)
float	g_offset_iir_filter [MA_OFFSET_TAPSIZE_BASE+1]	IIR/FIR フィルタ通過後のオフセットデータ
float	g_red_led_iir_filter [MA_RED_TAPSIZE_BASE+1]	IIR/FIR フィルタ通過後の RED LED データ
float	g_ir_led_iir_filter [MA_IR_TAPSIZE_BASE+1]	IIR/FIR フィルタ通過後の IR LED データ
float	g_ir_led_peak	IR LED の peak-to-peak のカウント値
float	g_spo2_delay_value	IIR/FIR フィルタ結果の遅延値
float	gs_red_led_hpf_max	HPF 結果の最大値(RED LED)
float	gs_ir_led_hpf_max	HPF 結果の最大値(IR LED)
float	gs_ir_led_hpf_min	HPF 結果の最小値(IR LED)
float	gs_red_led_hpf_min	HPF 結果の最小値(RED LED)

表 4-9 変数一覧(2/2)

データ型	変数名	概要
int32_t	g_offset_data	オフセット $\Delta \Sigma$ A/D 変換結果
int32_t	g_red_led_data	PPG/RED LED $\Delta \Sigma$ A/D 変換結果
int32_t	g_ir_led_data	PPG/SpO <sub>2</sub> IR LED $\Delta \Sigma$ A/D 変換結果
int32_t	g_offset_iir_x_buff [IIR_SPO2_DEGREE_X]	IIR/FIR フィルタ用 X バッファ(オフセット)
int32_t	g_red_led_iir_x_buff [IIR_SPO2_DEGREE_X]	IIR/FIR フィルタ用 X バッファ(RED LED)
int32_t	g_ir_led_iir_x_buff [IIR_SPO2_DEGREE_X]	IIR/FIR フィルタ用 X バッファ(IR LED)
uint16_t	g_spo2_samp_cnt	PPG/SpO <sub>2</sub> のサンプリングカウンタ
e_spo2_status_t	g_spo2_status	PPG/SpO <sub>2</sub> 状態
e_heart_rate_dir_t	g_spo2_dir	PPG/SpO <sub>2</sub> の微分値方向
st_iir_data_t	g_offset_iir_data	PPG/SpO <sub>2</sub> のオフセット用フィルタ設定
st_iir_data_t	g_red_led_iir_data	PPG/SpO <sub>2</sub> の RED LED 用フィルタ設定
st_iir_data_t	g_ir_led_iir_data	PPG/SpO <sub>2</sub> の IR LED 用フィルタ設定
st_filter_buff_t	g_offset_iir_result	PPG/SpO <sub>2</sub> のオフセット用 IIR/FIR フィルタ結果
st_filter_buff_t	g_red_led_iir_result	PPG/SpO <sub>2</sub> の RED LED 用 IIR/FIR フィルタ結果
st_filter_buff_t	g_ir_led_iir_result	PPG/SpO <sub>2</sub> の IR LED 用 IIR/FIR フィルタ結果
st_filter_buff_t	g_offset_ma_data	PPG/SpO <sub>2</sub> のオフセット用移動平均フィルタ設定
st_filter_buff_t	g_red_led_ma_data	PPG/SpO <sub>2</sub> の RED LED 用移動平均フィルタ設定
st_filter_buff_t	g_ir_led_ma_data	PPG/SpO <sub>2</sub> の IR LED 用移動平均フィルタ設定
st_filter_buff_t	g_offset_hpf_data	PPG/SpO <sub>2</sub> のオフセット用 HPF 設定
st_filter_buff_t	g_red_led_hpf_data	PPG/SpO <sub>2</sub> の RED LED 用 HPF 設定
st_filter_buff_t	g_ir_led_hpf_data	PPG/SpO <sub>2</sub> の IR LED 用 HPF 設定
st_filter_buff_t	g_ir_led_5diff_data	PPG/SpO <sub>2</sub> の IR LED 用 HPF 設定
st_thre_cross_t	g_spo2_thre_cross	PPG/SpO <sub>2</sub> の閾値判定用構造体
uint8_t	g_spo2_init_flg	PPG/SpO <sub>2</sub> の測定開始フラグ
uint8_t	g_ir_meas_cnt	測定パターンテーブル内の IR LED 測定パターンの数

## (4) 関数一覧

表 4-10 関数一覧

ファイル名	関数名	概要
r_bsse_app_spo2.c	R_BSSE_App_SpO2_Func	PPG/SpO <sub>2</sub> 測定処理
if_spo2.c	R_BSSE_SpO2_DSAD_Store	PPG/SpO <sub>2</sub> 用 A/D 変換結果取得処理
	R_BSSE_SpO2_Filter	PPG/SpO <sub>2</sub> 用フィルタ処理
	R_BSSE_SpO2_Calc	PPG/SpO <sub>2</sub> 脈拍計算処理

## (5) 関数仕様

(a) R\_BSSE\_App\_SpO<sub>2</sub>\_Func

概要	PPG/SpO <sub>2</sub> 測定処理
宣言	void R_BSSE_App_SpO <sub>2</sub> _Func (void)
説明	PPG/SpO <sub>2</sub> 測定処理を実行します。
引数	なし
戻り値	なし
備考	なし

(b) R\_BSSE\_SpO<sub>2</sub>\_DSAD\_Store

概要	PPG/SpO <sub>2</sub> 用 A/D 変換結果取得処理
宣言	void R_BSSE_SpO <sub>2</sub> _DSAD_Store (int32_t)
説明	24 ビット ΔΣ A/D コンバータの結果を RAM に格納します。
引数	24 ビット ΔΣ A/D コンバータの結果
戻り値	なし
備考	なし

(c) R\_BSSE\_SpO<sub>2</sub>\_Filter

概要	PPG/SpO <sub>2</sub> 用フィルタ処理
宣言	void R_BSSE_SpO <sub>2</sub> _Filter (e_spo2_status_t)
説明	SpO <sub>2</sub> 用のフィルタ処理(LPF, HPF 処理)を行います。
引数	SpO <sub>2</sub> の状態
戻り値	なし
備考	フィルタ適用後の測定データは、以下に格納されます。 g_offset_iir_result (オフセット) g_red_led_iir_result (RED LED) g_ir_led_iir_result (IR LED)

(d) R\_BSSE\_SpO<sub>2</sub>\_Calc

概要	PPG/SpO <sub>2</sub> 脈拍計算処理
宣言	void R_BSSE_SpO <sub>2</sub> _Calc (e_spo2_status_t)
説明	微分フィルタ処理、SSF フィルタ処理、移動平均フィルタ処理を実行し、閾値判定を行います。閾値を超えた場合は脈波の極値と判断し脈拍数を算出します。 また IR LED、RED LED の AC 成分 DC 成分を求めて SpO <sub>2</sub> 値を算出します。
引数	SpO <sub>2</sub> の状態
戻り値	なし
備考	ユーザ設定可能パラメータ SPO2_THRE_UPDATE_TIME で閾値の更新時間を変更できます。算出された脈拍数は g_ppg_heart_rate に格納されます。

4.4.3 PPG/SpO<sub>2</sub>用フィルタ処理

以下に PPG/SpO<sub>2</sub>用フィルタ処理フローを示します。

HPF 以降のフィルタ処理は、脈拍計算処理に含まれます。

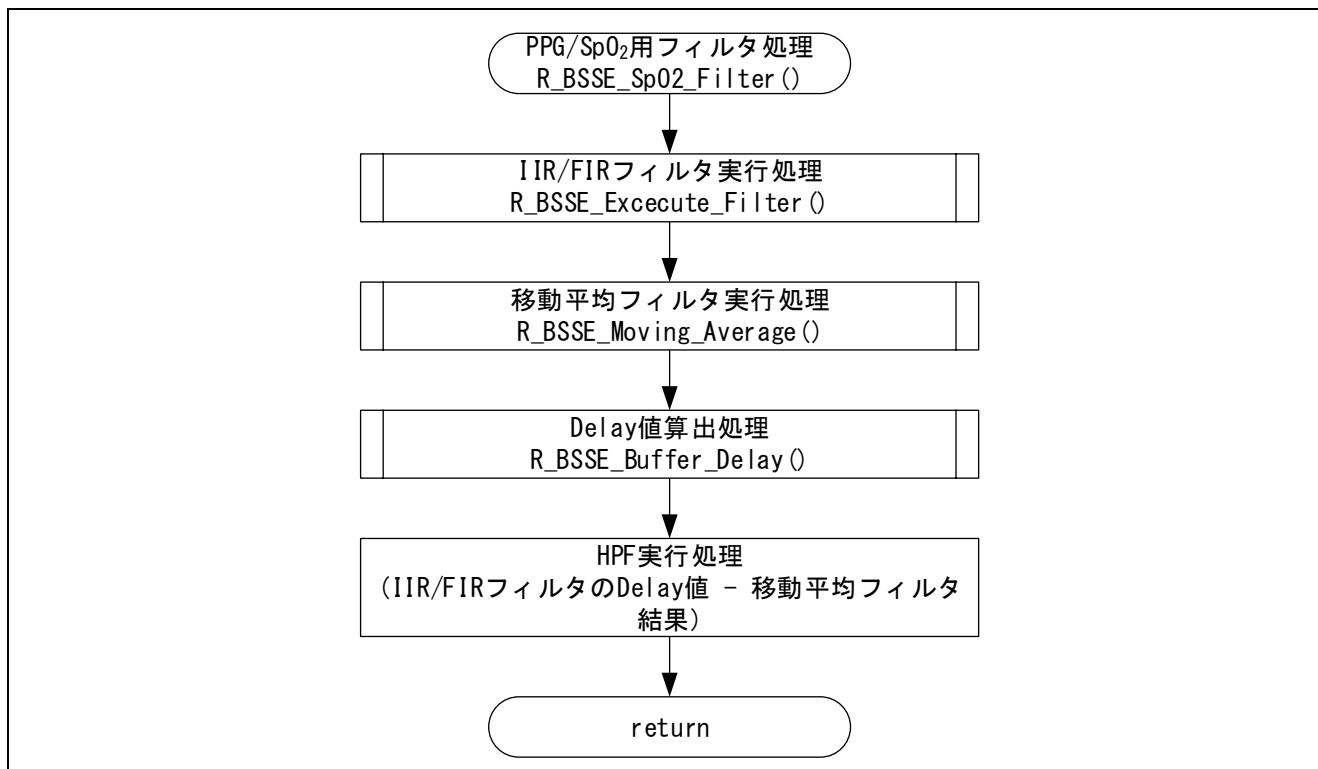


図 4-9 PPG/SpO<sub>2</sub>用フィルタ処理フロー

4.4.4 PPG/SpO<sub>2</sub>用脈拍計算処理

図 4-10 に PPG/SpO<sub>2</sub>用脈拍計算処理フローを示します。

IR LED の 24 ビット ΔΣ A/D 変換結果から脈拍数を算出し、さらに SpO<sub>2</sub> 値を算出します。

以下に、脈拍時間の計算方法を示します。

$$\text{脈拍時間} = 1 \text{ 拍中の IR LED サンプルカウント} \times \text{IR LED 測定間隔}$$

※ IR LED 測定間隔については、「6.7 測定シーケンスとサンプリングパターンテーブルの設定方法」を参照してください。

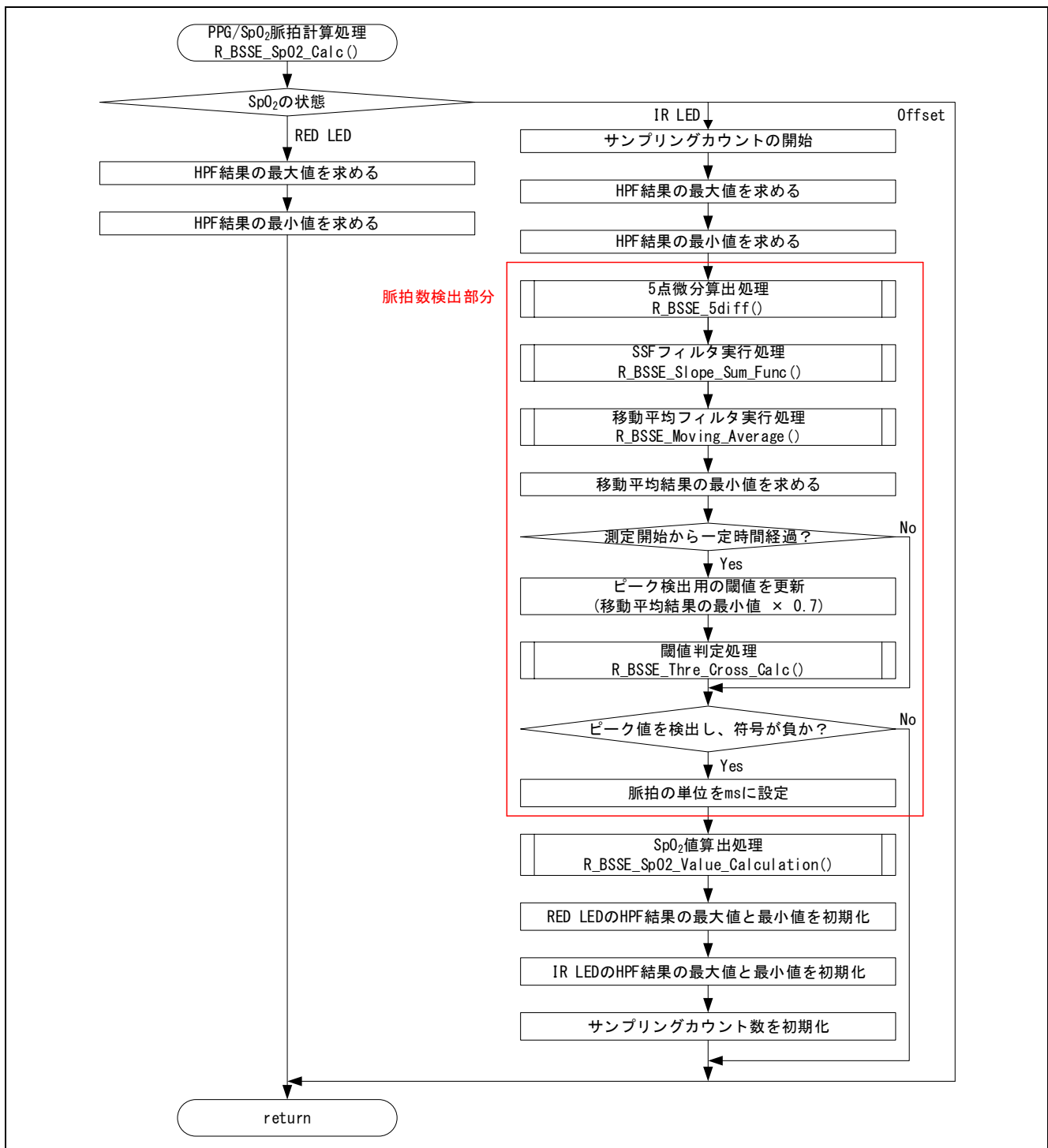


図 4-10 PPG/SpO<sub>2</sub>用脈拍計算処理フロー

## 5. デジタルフィルタと閾値判定

以下に本サンプルコードで使用するフィルタと対応関数一覧を示します。

使用するフィルタは、用途により異なります。「2 測定原理」を参照してください。

表 5-1 フィルタと対応関数一覧

フィルタ	フィルタ関数名
ローパスフィルタ(LPF)	R_BSSE_Execute_Filter (IIR/FIR フィルタ実行処理) R_BSSE_Filter_floating_Calc
ハイパスフィルタ(HPF)	R_BSSE_Buffer_Delay R_BSSE_Moving_Average
微分フィルタ	R_BSSE_5diff
二乗フィルタ	(専用関数なし)
SSF フィルタ	R_BSSE_Slope_Sum_Func
移動平均フィルタ	R_BSSE_Moving_Average
閾値判定	R_BSSE_Thre_Cross_Calc



5.1 ローパスフィルタ(LPF)

5.1.1 IIR/FIR フィルタ

IIR/FIR フィルタを LPF として使用します。図 5-1 に IIR/FIR フィルタの伝達関数を示します。

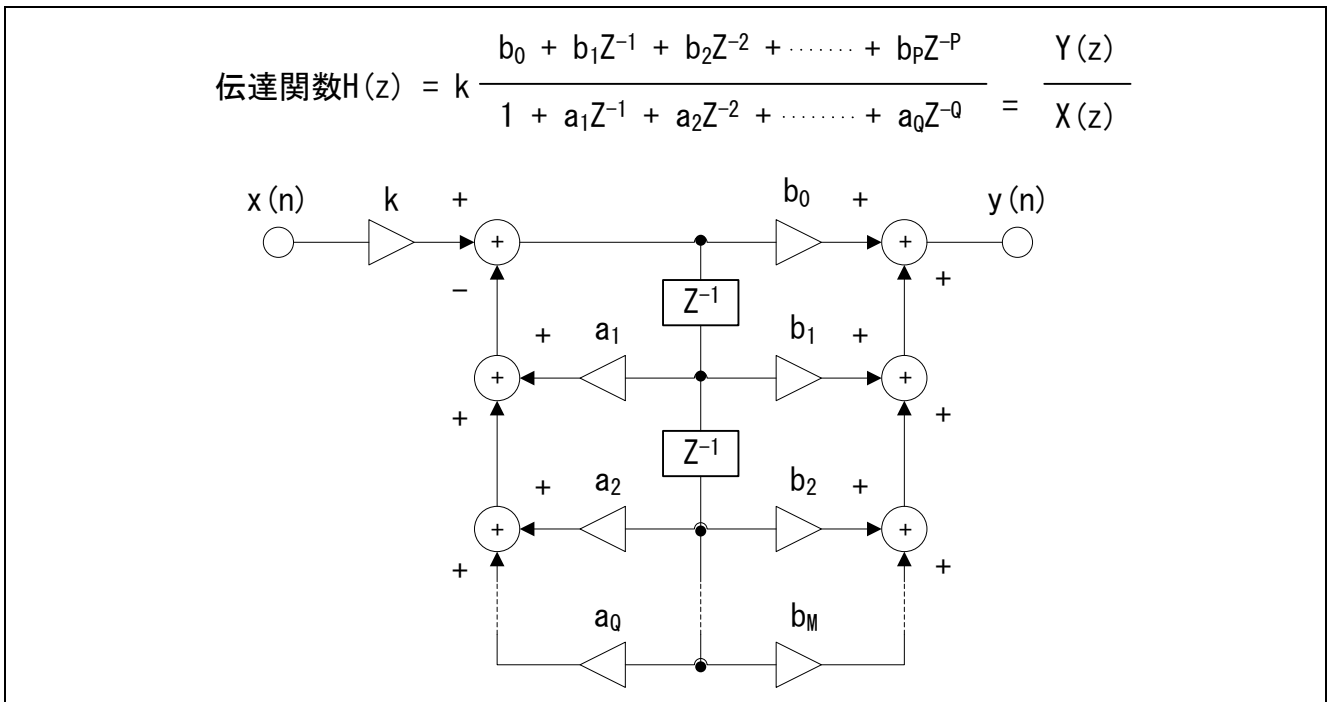


図 5-1 IIR/FIR フィルタの伝達関数

この伝達関数 H(z)をプログラムに実装するため、伝達関数 H(z)から Y(z)を求め、逆 Z 変換を行った式を出力値 y(n)について解きます(図 5-2 を参照)。

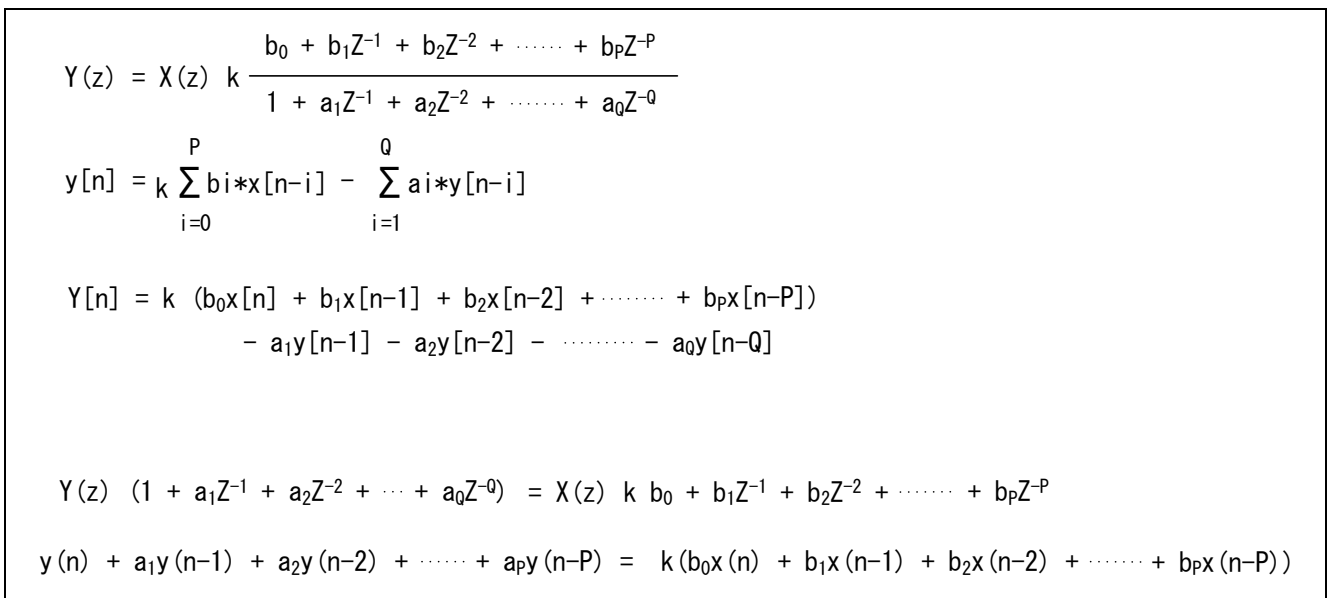


図 5-2 出力値 y(n)の求め方

図 5-3 に IIR/FIR フィルタ制御フローを示します。

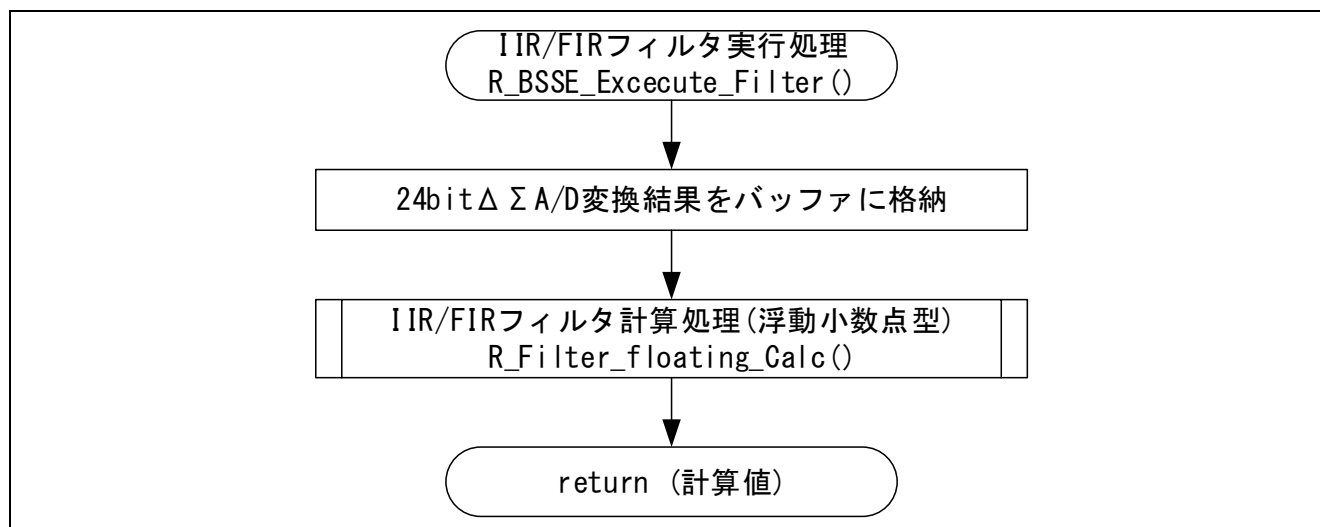


図 5-3 IIR/FIR フィルタ制御フロー

### (1) 列挙子一覧

表 5-2 列挙子一覧

列挙子名	値	備考
e_data_type_t	DATA_TYPE_UINT8 = 0	符号なし 8bit 型データ
	DATA_TYPE_INT8 = 1	符号付き 8bit 型データ
	DATA_TYPE_UINT16 = 2	符号なし 16bit 型データ
	DATA_TYPE_INT16 = 3	符号付き 16bit 型データ
	DATA_TYPE_UINT32 = 4	符号なし 32bit 型データ
	DATA_TYPE_INT32 = 5	符号付き 32bit 型データ
	DATA_TYPE_FLOAT = 6	float 型データ

### (2) 構造体一覧

表 5-3 構造体一覧

構造体名	データ型	変数名	概要
st_iir_data_t	float *	coefficient_kb	フィルタ用 k×b 係数(Forward 側)データ
	float *	coefficient_a	フィルタ用 a 係数(Feedback 側)データ
	uint8_t	b_order	b 係数の次数
	uint8_t	a_order	a 係数の次数
	uint8_t	x_buff_size	X バッファサイズ(注 1)
	uint8_t	y_buff_size	Y バッファサイズ(注 2)
	uint8_t	x_index	X バッファ用インデックス
	uint8_t	y_index	Y バッファ用インデックス
	e_data_type_t	x_data_type	X バッファのデータ型
	e_data_type_t	y_data_type	Y バッファのデータ型
	void *	buf_x	X 用バッファポインタ
	void *	buf_y	Y 用バッファポインタ

注 1 : x\_buff\_size に b\_order +1 以上を設定してください。

注 2 : y\_buff\_size に a\_order +1 以上を設定してください。

## (3) 関数一覧

表 5-4 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_Execute_Filter	IIR/FIR フィルタ実行処理
	R_BSSE_Filter_floating_Calc	IIR/FIR フィルタの浮動小数点演算処理

## (4) 関数仕様

## (a) R\_BSSE\_Execute\_Filter

概要	IIR/FIR フィルタ実行処理
宣言	float_t R_BSSE_Execute_Filter (int32_t, st_iir_data_t *)
説明	IIR/FIR フィルタを実行します。
引数 1	測定データ
引数 2	IIR/FIR フィルタ用データポインタ
戻り値	フィルタ結果
備考	なし

## (b) R\_BSSE\_Filter\_floating\_Calc

概要	IIR/FIR フィルタの浮動小数点演算処理
宣言	float R_BSSE_Filter_floating_Calc (st_iir_data_t *)
説明	浮動小数点の IIR/FIR フィルタを処理します。
引数	フィルタ用データポインタ
戻り値	浮動小数点の IIR/FIR フィルタ結果
備考	係数が 0 の部分は計算しない

## 5.2 ハイパスフィルタ (HPF)

HPF の結果を求めるため、IIR/FIR フィルタの結果と移動平均フィルタを用います。

HPF の計算式を以下に示します。

$$\text{HPF の算出値} = \text{IIR/FIR フィルタ結果}[n - \text{delay}] - \text{移動平均フィルタの結果}[n]$$

※IIR/FIR フィルタ結果[n - delay] : IIR/FIR フィルタの(n - delay)回目の算出結果

※移動平均フィルタの結果[n] : 移動平均フィルタの n 回目の算出結果

IIR/FIR フィルタで指定する遅延値(delay)は以下の式で求めます。

$$\text{IIR/FIR フィルタの遅延値}(\text{delay}) = \frac{\text{tap\_size} - 1}{2}$$

※tap\_size : 移動平均回数

## 5.2.1 IIR/FIR フィルタの遅延値取得

## (1) 構造体一覧

表 5-5 構造体一覧

構造体名	データ型	変数名	概要
st_filter_buff_t	float *	re_data	フィルタ結果格納バッファアドレス
	int16_t	index	フィルタ結果格納バッファ用インデックス
	uint16_t	re_buff_size	フィルタ結果格納バッファサイズ

## (2) 関数一覧

表 5-6 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_Buffer_Delay	指定した遅延値のバッファの値取得処理
	R_BSSE_Buffer_Get	指定インデックスのバッファ値取得処理

## (3) 関数仕様

## (a) R\_BSSE\_Buffer\_Delay

概要	指定した遅延値のバッファの値取得処理
宣言	float R_BSSE_Buffer_Delay (st_filter_buff_t, int16_t)
説明	指定した遅延値のバッファの値を取得します。
引数 1	値を取得したいバッファのデータポインタ
引数 2	遅延値
戻り値	指定した遅延値のバッファの値
備考	R_BSSE_Buffer_Get 関数を使用してバッファの値を取得します。

## (b) R\_BSSE\_Buffer\_Get

概要	指定インデックスのバッファ値取得処理
宣言	float R_BSSE_Buffer_Get (float *, int16_t, uint16_t)
説明	指定したインデックスのバッファの値を取得します。
引数 1	値を取得したいバッファのポインタ
引数 2	指定のインデックス
引数 3	値を取得したいバッファのサイズ
戻り値	指定したインデックスのバッファの値
備考	なし

### 5.2.1 移動平均フィルタ

移動平均回数は config.h で指定(変更)することができます。

図 5-4 に移動平均フィルタ制御フローを示します。

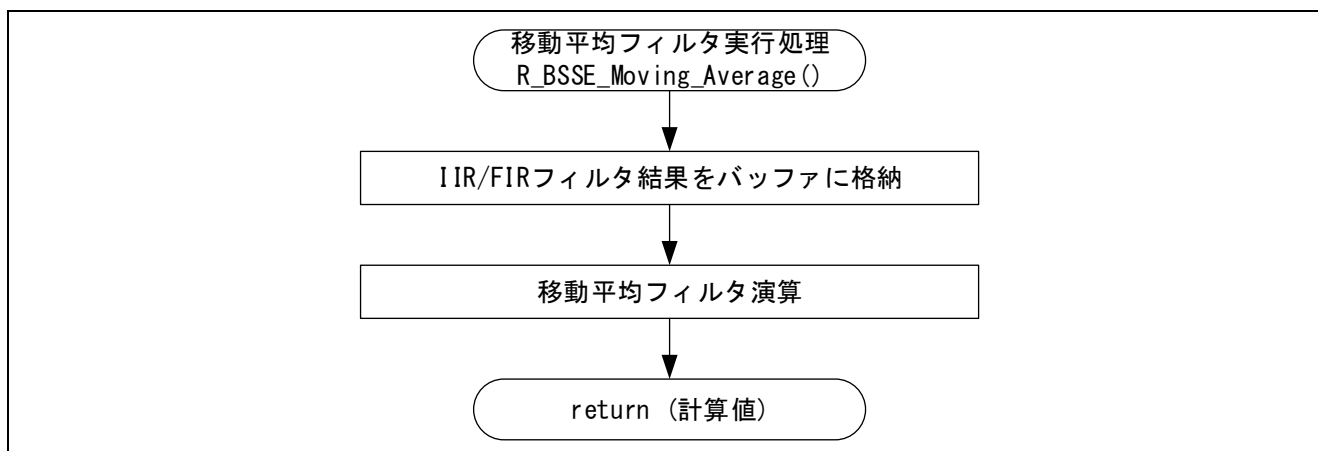


図 5-4 移動平均フィルタ制御フロー

移動平均フィルタの計算式を以下に記します。

$$Y[n] = Y[n - 1] + \frac{X[n]}{\text{tap\_size}} - \frac{X[n - \text{tap\_size}]}{\text{tap\_size}}$$

※Y[n] : n 個目の移動平均フィルタ結果、X[n] : n 個目の IIR/FIR フィルタ結果、tap\_size : 移動平均回数

#### (1) 構造体一覧

表 5-5 を参照してください。

#### (2) 関数一覧

表 5-7 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_Moving_Average	移動平均フィルタ実行処理

#### (3) 関数仕様

##### (a) R\_BSSE\_Moving\_Average

概要 移動平均フィルタ実行処理

宣言 float R\_BSSE\_Moving\_Average (st\_filter\_buff\_t \*, float, uint16\_t)

説明 移動平均フィルタを実行します。

引数 1 移動平均フィルタ対象データポインタ

引数 2 移動平均フィルタ結果データポインタ

引数 3 移動平均回数(tap\_size)

戻り値 移動平均フィルタ計算結果

備考 移動平均フィルタ対象データのバッファサイズとして tap\_size+1 以上を設定してください。

### 5.3 微分フィルタ

入力値(x[n])に対して5点微分を行い、その結果を返します。5点微分の計算式を以下に示します。

$$\text{微分値}[n] = \frac{1}{8} \times (2 \times x[n] + x[n-1] - x[n-3] - 2 \times x[n-4])$$

#### (1) 構造体一覧

表 5-5 を参照してください。

#### (2) 関数一覧

表 5-8 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_5diff	微分算出処理

#### (3) 関数仕様

##### (a) R\_BSSE\_5diff

概要	微分算出処理
宣言	float R_BSSE_5diff (st_filter_buff_t *)
説明	5点微分を行います。
引数 1	微分対象データポインタ
戻り値	5点微分フィルタ計算結果
備考	R_BSSE_Buffer_Get 関数を使用してバッファの値を取得します。 微分対象データのバッファサイズを5以上にしてください。

## 5.4 SSF フィルタ

入力値に対して正負を判定します。処理内容を以下に示します。

表 5-9 SSF フィルタの処理内容

入力条件	処理内容
取得したい値が正の場合	入力値が正ならば入力値を返し、負ならば 0 を返す
取得したい値が負の場合	入力値が正ならば 0 を返し、負ならば入力値を返す。

### (1) 構造体一覧

表 5-5 を参照してください。

### (2) 関数一覧

表 5-10 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_Slope_Sum_Func	SSF フィルタ実行処理

### (3) 関数仕様

#### (a) R\_BSSE\_Slope\_Sum\_Func

概要	SSF フィルタ実行処理
宣言	float R_BSSE_Slope_Sum_Func (float, uint8_t)
説明	入力値の正負を判定します。 取得したい値が正の場合、入力値が正の場合は入力値を返し、負の場合は 0 を返します。 取得したい値が負の場合、入力値が正の場合は 0 を返し、負の場合は入力値を返します。
引数 1	SSF フィルタ対象データポインタ
引数 2	取得したい値の正負 (0 = 負、1 = 正)
戻り値	SSF フィルタ計算結果
備考	なし

## 5.5 移動平均フィルタ

「5.2.1 移動平均フィルタ」を参照してください。



## 5.6 閾値判定

前回の入力値に対して現在の入力値が閾値(threshold\_m)を跨いだかどうかを判定します。

閾値を跨いだと判定した場合、測定回数(samp\_cnt)を返します。

閾値を跨いでいないと判定した場合、0を返します。

### (1) 列挙子一覧

表 5-11 列挙子一覧

e_heart_rate_dir_t	HEART_RATE_DIR_POS = 0	入力値が閾値より大きい
	HEART_RATE_DIR_NEG = 1	入力値が閾値より小さい
	HEART_RATE_DIR_NONE = 2	初期状態

### (2) 構造体一覧

表 5-12 構造体一覧

構造体名	データ型	変数名	概要
st_thre_cross_t	float *	diff_buf_f	閾値判定の入力値バッファ X[n]のポインタ
	uint16_t *	samp_cnt	測定回数値のポインタ
	float	threshold_m	ピーク検出用の閾値
	e_heart_rate_dir_t *	cross_type	微分データの方向ポインタ

### (3) 関数一覧

表 5-13 関数一覧

ファイル名	関数名	概要
filter.c	R_BSSE_Thre_Cross_Calc	閾値判定処理

### (4) 関数仕様

#### (a) R\_BSSE\_Thre\_Cross\_Calc

概要	閾値判定処理
宣言	float R_BSSE_Thre_Cross_Calc (float, st_thre_cross_t *)
説明	閾値判定を実行します。
引数 1	現在の入力値のデータポインタ
引数 2	閾値判定用の構造体ポインタ
戻り値	閾値をクロスしたと判定した場合、戻り値：測定回数(samp_cnt) 閾値をクロスしていないと判定した場合、戻り値：0
備考	なし

## 6. 応用例

コード生成(CG)を使用し、ECG, PPG, SpO<sub>2</sub> 同時測定制御ソフトウェアと SW (スイッチ) による測定開始/停止処理を組み込んだアプリケーション例を示します。

### 6.1 動作概要

動作概要については、「図 4-1 本サンプルコード全体の動作フロー」を参照してください。

SW の押下を割り込み端子で検出し、測定開始と測定終了を制御します。SW 操作の詳細については、「6.5 SW (スイッチ) 入力制御」を参照してください。

図 6-1 に SW 入力制御のタイミングチャートを示します。

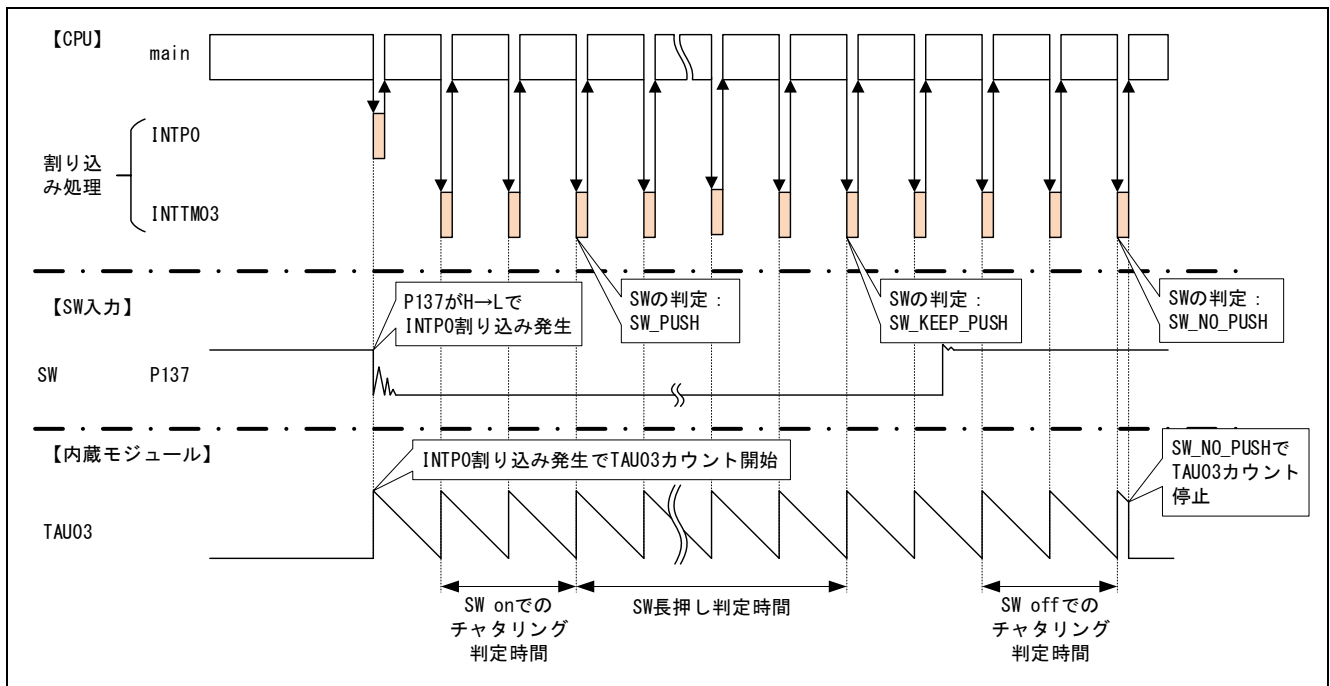


図 6-1 SW 入力制御のタイミングチャート

### 6.2 SW 入力による状態遷移

以下に SW 入力操作による測定状態の遷移図を示します。SW の状態検出については図 6-1 を参照してください。SW 操作の詳細については、「6.5 SW (スイッチ) 入力制御」を参照してください。

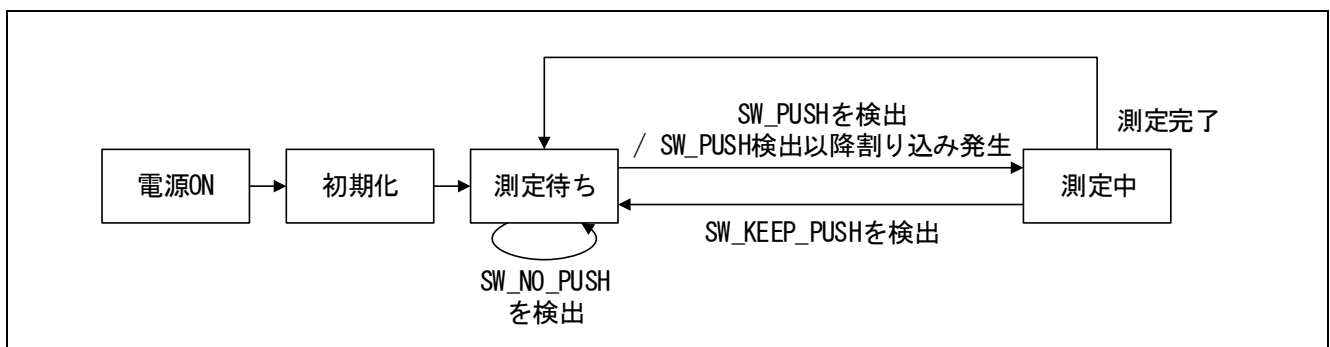


図 6-2 状態遷移図

### 6.3 使用するリソース

以下に測定制御以外で使用するリソースを示します。

測定制御で使用するリソースについては、「3.4 使用するリソース」を参照してください。

表 6-1 使用するリソース

リソース	内容
INTP	SW 制御端子
TAU	SW のチャタリング検出に使用

### 6.4 測定アプリケーション

「図 4-1 本サンプルコード全体の動作フロー」を参照してください。

#### 6.4.1 追加の初期化処理

追加の初期化処理を以下に示します。

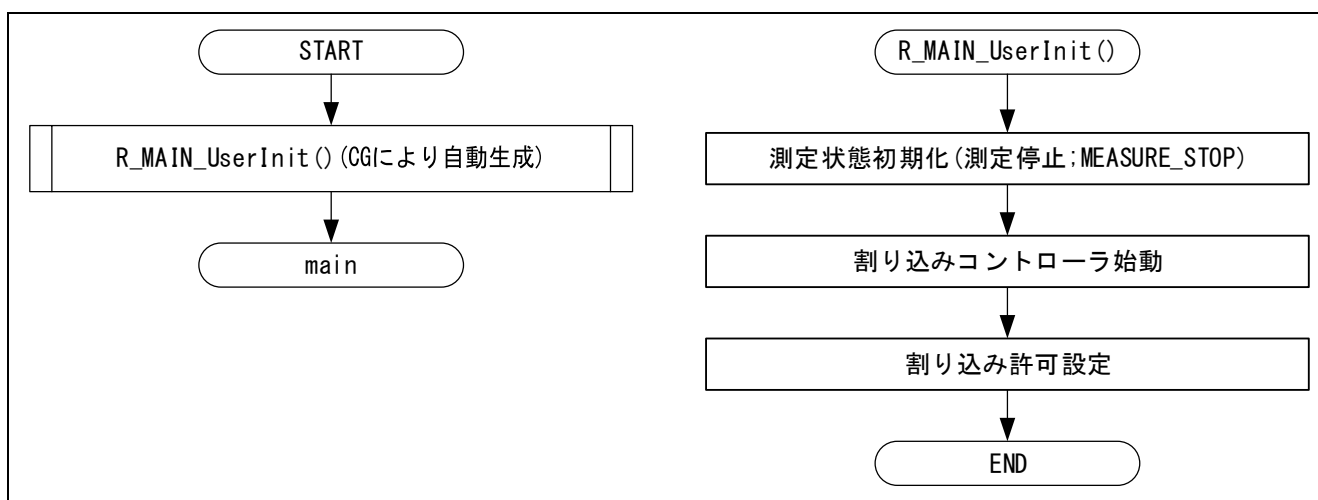


図 6-3 追加の初期化処理

#### (1) 関数一覧

表 6-2 関数一覧

ファイル名	関数名	概要
r_cg_main.c	R_MAIN_UserInit	測定状態初期化および割り込みコントローラ(キースキャン用)始動

#### (2) 関数仕様

##### (a) R\_MAIN\_UserInit

概要	初期化処理
宣言	void R_MAIN_UserInit (void)
説明	測定状態を初期化(停止状態 : MEASURE_STOP)し、割り込みコントローラ動作(キースキャン)を開始し、割り込みを許可に設定します。
引数	なし
戻り値	なし
備考	本関数は CG により出力されたものを編集しています。

6.4.2 ユーザ設定可能パラメータの反映処理

(1) 関数一覧

表 6-3 関数一覧

ファイル名	関数名	概要
r_bsse_ctrl_measure.c	R_BSSE_Bio_Sens_Platform_Create	測定用センサ関連の設定

(2) 関数仕様

(a) R\_BSSE\_Bio\_Sens\_Platform\_Create

概要 測定用センサ関連の設定(タイマ/LED/DSAD)  
 宣言 void R\_BSSE\_Bio\_Sens\_Platform\_Create (void)  
 説明 タイマ設定関数、LED 設定関数および DSAD 設定関数を呼び出します。  
 引数 なし  
 戻り値 なし  
 備考 なし

6.4.3 測定制御処理とスタンバイ機能処理

図 6-4 に測定制御処理を示します。

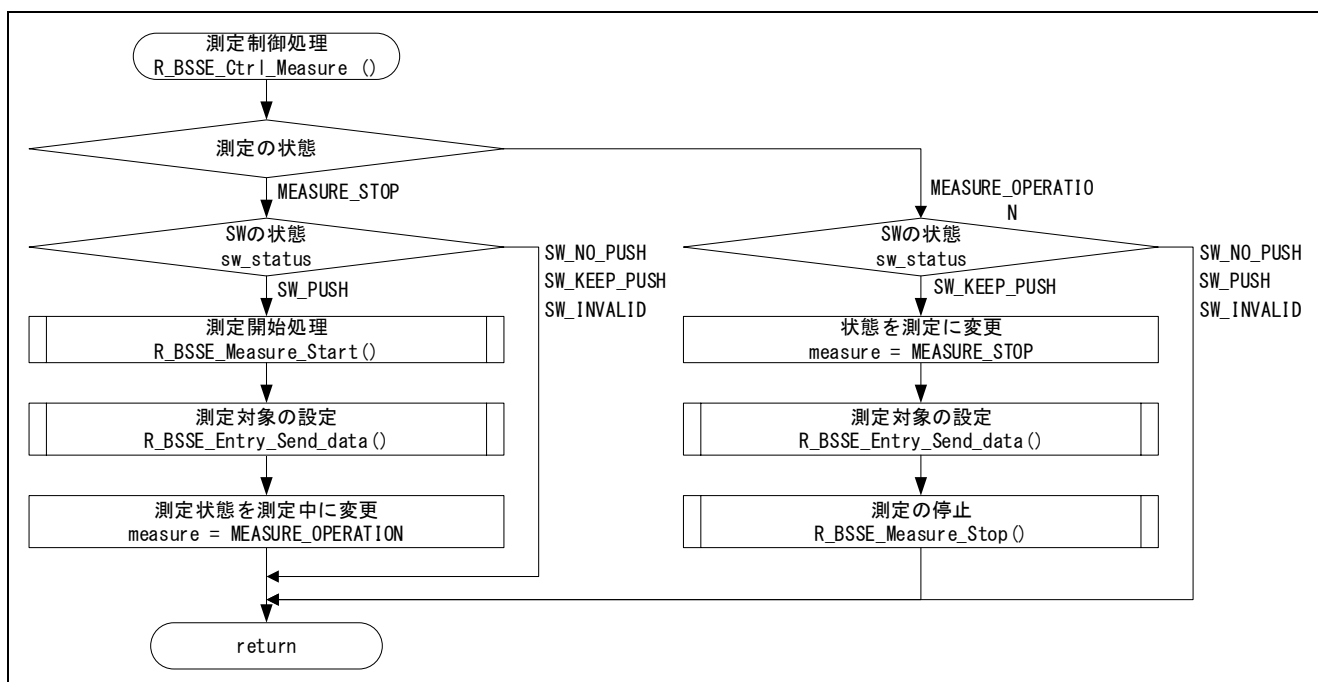


図 6-4 測定制御処理

図 6-5 に測定制御処理を示します。

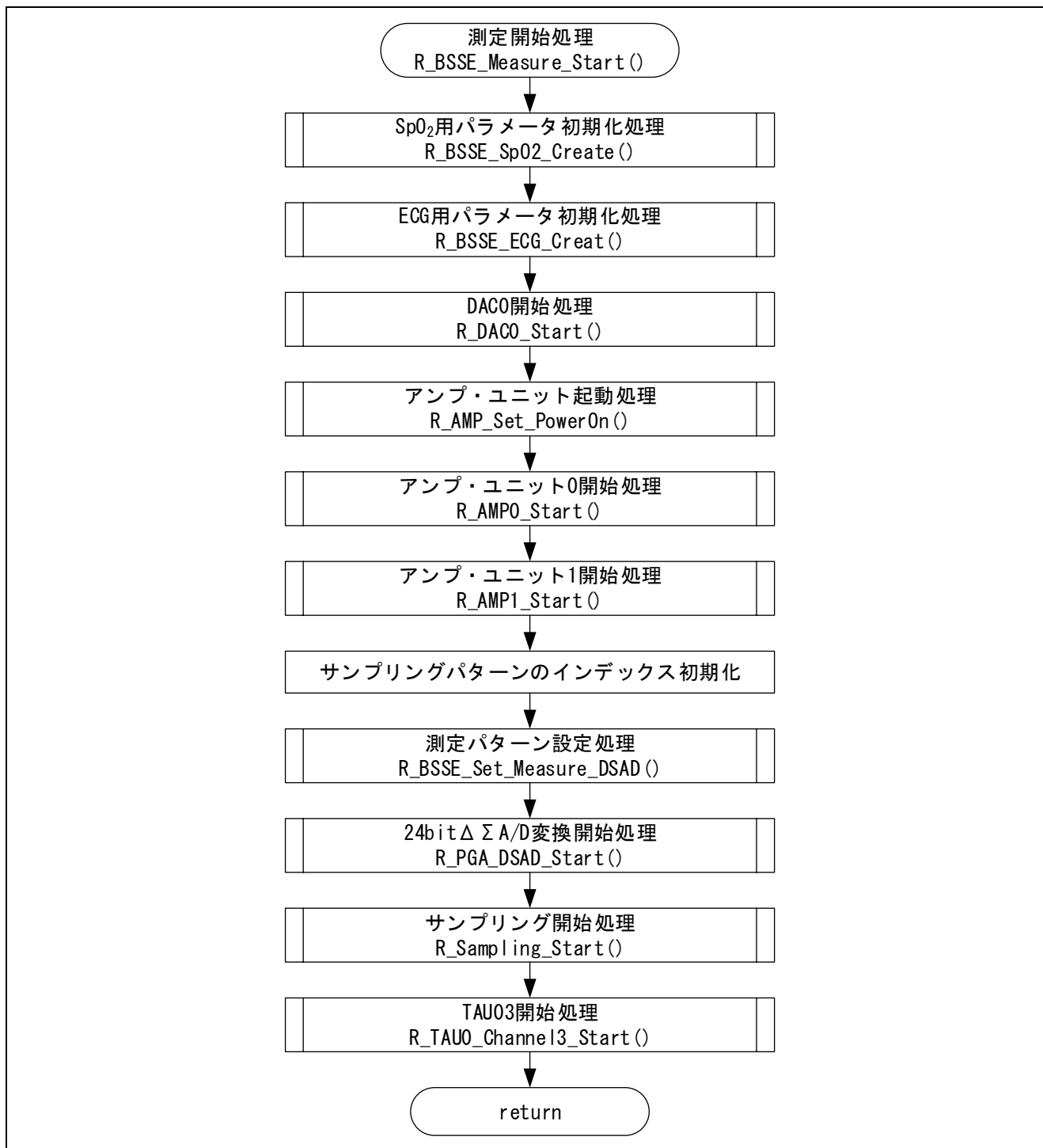


図 6-5 測定開始フロー

図 6-6 に測定停止フローを示します。

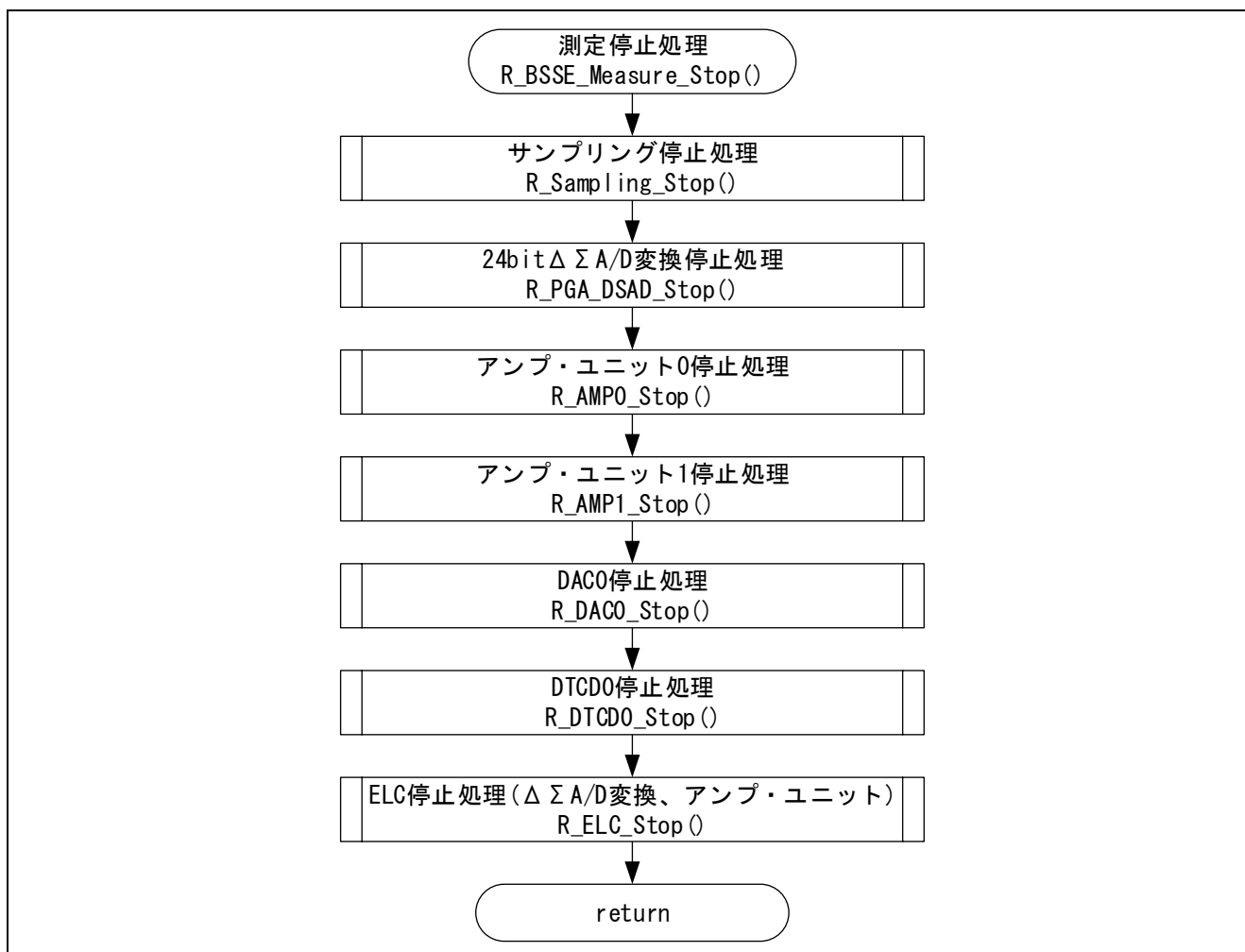


図 6-6 測定停止フロー

(1) 列挙子一覧

表 6-4 列挙子一覧

列挙子名	値	備考
e_measure_status_t	MEASURE_STOP = 0	測定停止中状態
	MEASURE_OPERATION = 1	測定動作中状態

(2) 変数一覧

表 6-5 変数一覧

データ型	変数名	概要
e_measure_status_t	g_measure_status	測定状態

## (3) 関数一覧

CG で生成された関数は「コード生成ツール ユーザーズマニュアル RL78 API リファレンス編 (R20UT4322JJ)」を参照ください。

表 6-6 関数一覧

ファイル名	関数名	概要
r_bsse_ctrl_measure.c	R_BSSE_Ctrl_Measure	測定制御処理
	R_BSSE_Standby	スタンバイ機能処理
	R_BSSE_Measure_Start	測定開始処理
	R_BSSE_Measure_Stop	測定停止処理
r_bsse_app_spo2.c	R_BSSE_SpO2_Create	SpO <sub>2</sub> 測定用変数初期化
r_bsse_app_ecg.c	R_BSSE_ECG_Create	ECG 測定用変数初期化
r_cg_dac.c	R_DAC0_Start	CG により出力された関数
r_cg_amp.c	R_AMP_Set_PowerOn	CG により出力された関数
	R_AMP0_Start	CG により出力された関数
	R_AMP1_Start	CG により出力された関数
if_dsad.c	R_BSSE_Set_Measure_DSAD	LED/DSAD 設定
r_cg_pga_dsad.c	R_PGA_DSAD_Start	CG により出力された関数
r_cg_tau.c	R_Sampling_Start	測定開始(タイマ動作開始)
	R_TAU0_Channel3_Start	CG により出力された関数
	R_Sampling_Stop	測定終了(タイマ動作停止)
r_cg_pga_dsad.c	R_PGA_DSAD_Stop	CG により出力された関数
r_cg_amp.c	R_AMP0_Stop	CG により出力された関数
	R_AMP1_Stop	CG により出力された関数
r_cg_dac.c	R_DAC0_Stop	CG により出力された関数
r_cg_elc.c	R_ELC_Stop	CG により出力された関数

## (4) 関数仕様

CG 出力関数は省略しています。

## (a) R\_BSSE\_Ctrl\_Measure

概要	測定制御処理
宣言	e_measure_status_t R_BSSE_Ctrl_Measure (e_measure_status_t, e_sw_status_t)
説明	測定の開始/停止状態と SW 入力状態から測定を制御します。
引数 1	現在の測定状態
引数 2	SW の入力状態
戻り値	更新する測定状態
備考	なし

## (b) R\_BSSE\_Measure\_Start

概要	測定開始処理
宣言	void R_BSSE_Measure_Start (void)
説明	各種測定を開始します。
引数	なし
戻り値	なし
備考	なし

## (c) R\_BSSE\_Measure\_Stop

概要	測定停止処理
宣言	void R_BSSE_Measure_Stop (void)
説明	サンプリング用タイマを停止後、送信データがないことを確認し、TAU0 のチャンネル 3 以外のチャンネルを停止します。
引数	なし
戻り値	なし
備考	なし

## (d) R\_BSSE\_Standby

概要	スタンバイ機能処理
宣言	void R_BSSE_Standby (e_measure_status_t)
説明	測定停止中状態、かつ SW(INTPO 割り込み許可状態)が押されていない状態の場合、STOP モードへ移行します。その他の場合は、HALT モードへ移行します。
引数	status : 測定状態
戻り値	なし
備考	なし

## (e) R\_BSSE\_SpO2\_Create

概要	SpO <sub>2</sub> 測定用変数初期化
宣言	void R_BSSE_SpO2_Create (void)
説明	各種変数およびバッファを初期化します。
引数	なし
戻り値	なし
備考	なし

## (f) R\_BSSE\_ECG\_Create

概要	ECG 測定用変数初期化
宣言	void R_BSSE_ECG_Create (void)
説明	各種変数およびバッファを初期化します。
引数	なし
戻り値	なし
備考	なし

## (g) R\_BSSE\_Set\_Measure\_DSAD

概要	LED/DSAD 設定
宣言	void R_BSSE_Set_Measure_DSAD (uint8_t)
説明	LED 設定関数および DSAD 設定関数を呼び出します。
引数	サンプリングパターン
戻り値	なし
備考	なし

## (h) R\_Sampling\_Start

概要	測定開始
宣言	void R_Sampling_Start (void)
説明	DSAD 起動用タイマおよび AMP0 起動用タイマの動作を開始します。
引数	なし
戻り値	なし



備考           なし

(i) R\_Sampling\_Stop

概要           測定終了

宣言           void R\_Sampling\_Stop (void)

説明           DSAD 起動用タイマおよび AMP0 起動用タイマの動作を停止します。

引数           なし

戻り値        なし

備考           なし

### 6.5 SW（スイッチ）入力制御

本サンプルコードの SW 入力について説明します。

SW 入力検出のために、INTP0 とタイマ・アレイ・ユニットの ch3(TAU03)のインターバル・タイマを使用しています。SW 押下(ON 状態)により INTP0 割り込みが発生し、TAU03 のカウントをスタートさせます。その後、TAU03 の周期ごとにキースキャンを実行し、チャタリング除去、ON/OFF 判定、長押し判定をします。OFF 状態を検出した場合、TAU03 を停止し、INTP0 割り込み待ち状態になります。

表 6-7 に SW 割り付けを示します。

表 6-7 SW 割り付け

制御	Port の割り当て
測定開始 / 終了	P137 / INTP0

リセット後、SW を押すことにより、測定を開始します。測定中、または測定待ちの間に SW を 1 秒以上長押しすることで測定を終了します。以下に SW 入力制御フローを示します。

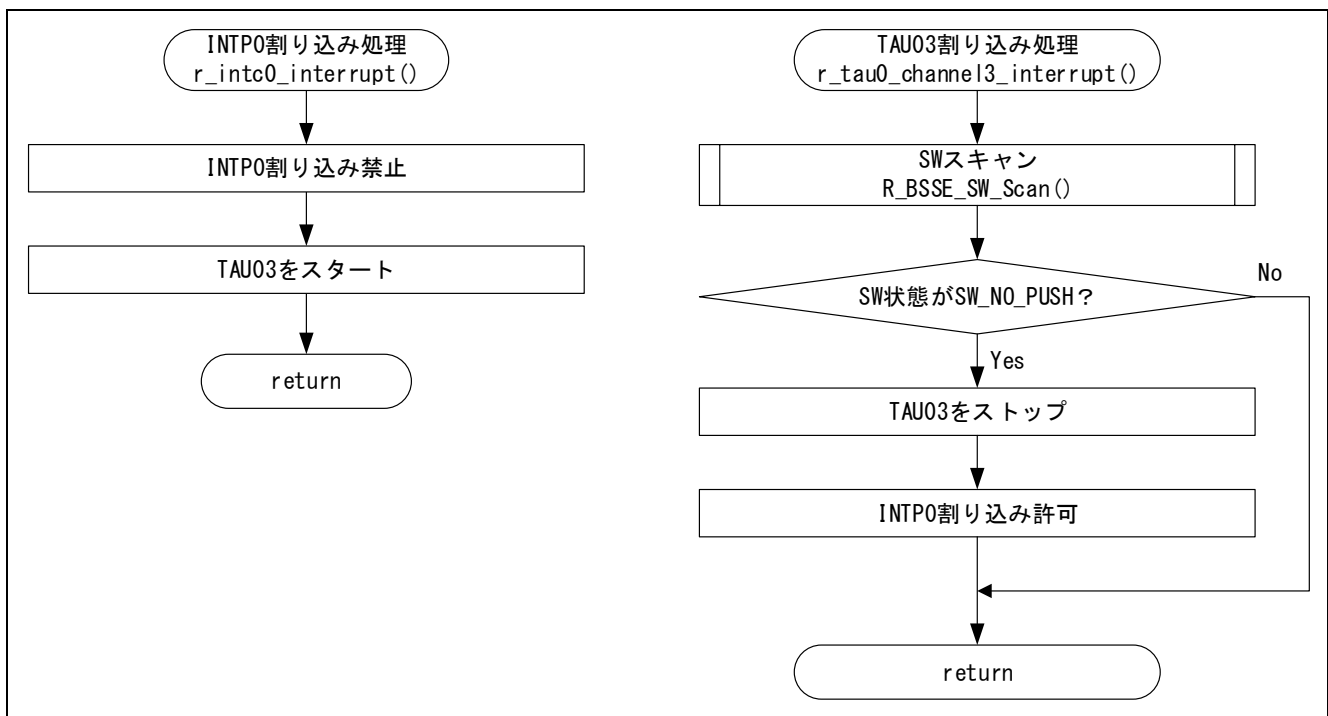


図 6-7 SW 入力制御フロー

#### (1) マクロ一覧

表 6-8 マクロ一覧

マクロ名	値	備考
SW_VALID_TIMES	3U	SW 押しが有効となる一致回数 (注)
SW_KEEP_TIMES	300U	SW 長押しが有効となる一致回数 (注)

注：10 ミリ秒タイマを使用

## (2) 列挙子一覧

表 6-9 列挙子一覧

列挙子名	値	備考
e_sw_status_t	SW_NO_PUSH = 0	SW 押し無し
	SW_PUSH = 1	SW 押し
	SW_KEEP_PUSH = 2	SW 長押し
	SW_INVALID = 3	SW 不定状態

## (3) 変数一覧

表 6-10 変数一覧

データ型	変数名	概要
e_sw_status_t	g_sw_status	SW 入力状態
uint16_t	g_sw_keep_count	長押し判定カウンタ
uint8_t	g_sw_count	チャタリング判定カウンタ
uint8_t	g_sw_data_old	前回 SW 入力情報

## (4) 関数一覧

表 6-11 関数一覧

ファイル名	関数名	概要
r_cg_intp_user.c	r_intc0_interrupt	INTP0 の割り込み処理
r_cg_tau_user.c	r_tau0_channel3_interrupt	TAU03 の割り込み処理
if_sw.c	R_BSSE_SW_Scan	SW スキャン処理

## (5) 関数仕様

## (a) r\_intc0\_interrupt

概要	INTP0 の割り込み処理
宣言	void r_intc0_interrupt (void)
説明	SW を不定状態として、SW 押し判定用の TAU03 をスタートし、INTP0 割り込みを禁止します。
引数	なし
戻り値	なし
備考	なし

## (b) r\_tau0\_channel3\_interrupt

概要	TAU03 の割り込み処理
宣言	void r_tau0_channel3_interrupt (void)
説明	SW スキャン処理を実行し、SW 状態を取得します。 SW が押されていない場合は、TAU03 を停止し、INTP0 割り込みを許可します。
引数	なし
戻り値	なし
備考	なし

## (c) R\_BSSE\_SW\_Scan

概要	SW スキャン処理
宣言	e_sw_status_t_t R_BSSE_SW_Scan (void)

---

説明	SW スキャンを制御します。
引数	なし
戻り値	e_sw_status_t : SW 入力状態 00H (SW_NO_PUSH) : SW 押しなし 01H (SW_PUSH) : SW 押し 02H (SW_KEEP_PUSH) : SW 長押し
備考	なし

## 6.6 アプリケーション用設定ファイル

## 6.6.1 マクロ一覧

対象ファイルは、以下です。

- config.h
- r\_cg\_userdefine.h

## (1) config.h

測定条件やフィルタ係数の変更時に合わせ込みが必要なマクロの一覧を示します。

表 6-12 config.h マクロ一覧

マクロ名	値	備考
SAMPLING_FREQ	200U	測定サイクル周波数 (200Hz)
CYCLE_PTN_TBL_SIZE	4U	サンプリングパターンテーブルサイズ
SPO2_FILTER_TYPE	FILTER_FLOAT	PPG/SpO <sub>2</sub> 用のフィルタタイプ
OFFSET_DAC	0U	PPG/SpO <sub>2</sub> オフセット測定時の DAC 出力
RED_LED_DAC	20U	PPG/SpO <sub>2</sub> RED LED 測定時の DAC 出力
IR_LED_DAC	169U	PPG/SpO <sub>2</sub> IR LED 測定時の DAC 出力
HPF_OFFSET_BUFF_NUM	5U	PPG/SpO <sub>2</sub> オフセット測定時の HPF 用バッファサイズ
HPF_RED_LED_BUFF_NUM	5U	PPG/SpO <sub>2</sub> RED LED 測定時の HPF 用バッファサイズ
HPF_IR_LED_BUFF_NUM	5U	PPG/SpO <sub>2</sub> IR LED 測定時の HPF 用バッファサイズ
SPO2_CALC_COEF_A	1.F	PPG/SpO <sub>2</sub> 計算式のセンサキャリブレーション用の係数 a
SPO2_CALC_COEF_B	0.F	PPG/SpO <sub>2</sub> 計算式の計算式のセンサキャリブレーション用の係数 b
SPO2_THREXCROSS_THRE	0.F	PPG/SpO <sub>2</sub> 用ピーク検出用の閾値の初期値
SPO2_THREXCROSS_RATE	0.6F	PPG/SpO <sub>2</sub> 用ピーク検出用の閾値の倍率
SPO2_THRE_UPDATE_TIME	2U	PPG/SpO <sub>2</sub> 用閾値更新時間[秒]
IIR_OFFSET_DEGREE_B	12U	PPG/SpO <sub>2</sub> のオフセット用 IIR/FIR フィルタの係数 b の次数
IIR_OFFSET_DEGREE_A	2U	PPG/SpO <sub>2</sub> のオフセット用 IIR/FIR フィルタの係数 a の次数
IIR_OFFSET_DEGREE_X	13U	PPG/SpO <sub>2</sub> のオフセット用 IIR/FIR フィルタの X バッファサイズ ※IIR_OFFSET_DEGREE_B + 1 以上を設定してください。
IIR_OFFSET_DEGREE_Y	3U	PPG/SpO <sub>2</sub> のオフセット用 IIR/FIR フィルタの Y バッファサイズ ※IIR_OFFSET_DEGREE_A + 1 以上を設定してください。
IIR_RED_LED_DEGREE_B	12U	PPG/SpO <sub>2</sub> の RED LED 用 IIR/FIR フィルタの係数 b の次数
IIR_RED_LED_DEGREE_A	2U	PPG/SpO <sub>2</sub> の RED LED 用 IIR/FIR フィルタの係数 a の次数
IIR_RED_LED_DEGREE_X	13U	PPG/SpO <sub>2</sub> の RED LED 用 IIR/FIR フィルタの X バッファサイズ

		※IIR_RED_LED_DEGREE_B +1 以上を設定してください。
IIR_RED_LED_DEGREE_Y	3U	PPG/SpO <sub>2</sub> の RED LED 用 IIR/FIR フィルタの Y バッファサイズ ※IIR_RED_LED_DEGREE_A +1 以上を設定してください。
IIR_IR_LED_DEGREE_B	12U	PPG/SpO <sub>2</sub> の IR LED 用 IIR/FIR フィルタの係数 b の次数
IIR_IR_LED_DEGREE_A	2U	PPG/SpO <sub>2</sub> の IR LED 用 IIR/FIR フィルタの係数 a の次数
IIR_IR_LED_DEGREE_X	13U	PPG/SpO <sub>2</sub> の IR LED 用 IIR/FIR フィルタの X バッファサイズ ※IIR_IR_LED_DEGREE_B +1 以上を設定してください。
IIR_IR_LED_DEGREE_Y	3U	PPG/SpO <sub>2</sub> の IR LED 用 IIR/FIR フィルタの Y バッファサイズ ※IIR_IR_LED_DEGREE_A +1 以上を設定してください。
MA_IR_TAPSIZE_BASE	50U	PPG/SpO <sub>2</sub> の IR LED 用 HPF に用いる移動平均フィルタの tap_size (移動平均回数)
MA_RED_TAPSIZE_BASE	50U	PPG/SpO <sub>2</sub> の RED LED 用 HPF に用いる移動平均フィルタの tap_size (移動平均回数)
MA_OFFSET_TAPSIZE_BASE	50U	PPG/SpO <sub>2</sub> のオフセット用 HPF に用いる移動平均フィルタの tap_size (移動平均回数)
SPO2_SSF_DIR	SSF_DIR_NEG	SSF で取得したい値の正負
ECG_FILTER_TYPE	FILTER_FLOAT	ECG 用のフィルタタイプ
MA_ECG_TAPSIZE_BASE	200U	ECG 用 HPF に用いる移動平均フィルタの tap_size (移動平均回数)
MA_ECG_TAPSIZE_DIFF	31U	ECG 用二乗フィルタ後に行う移動平均フィルタの tap_size (移動平均回数)
HPF_ECG_BUFF_NUM	5U	ECG 測定時の HPF 用バッファサイズ
ECG_THRE_CROSS_THRE	0.F	ECG 用ピーク検出用の閾値の初期値
ECG_THRE_CROSS_RATE	0.5F	ECG 用ピーク検出用の閾値の倍率
ECG_THRE_UPDATE_TIME	2U	ECG 用閾値更新時間[秒]
IIR_ECG_DEGREE_B	12U	ECG 用 IIR/FIR フィルタの係数 b の次数
IIR_ECG_DEGREE_A	2U	ECG 用 IIR/FIR フィルタの係数 a の次数
IIR_ECG_DEGREE_X	13U	ECG 用 IIR/FIR フィルタの X バッファサイズ ※IIR_ECG_DEGREE_B +1 以上で設定してください。
IIR_ECG_DEGREE_Y	3U	ECG 用 IIR/FIR フィルタの Y バッファサイズ ※IIR_ECG_DEGREE_B +1 以上で設定してください。
GET_EXTERNAL	ECG_DIRECTION_UP	ピーク値の判定符号
THRE_BUFF_NUM_DIFF	2U	閾値判定用バッファサイズ
MA_IR_BUFF_NUM_DIFF	25U	PPG/SpO <sub>2</sub> の IR LED 用 SSF フィルタ後に行う移動平均フィルタの tap_size (移動平均回数)

## (2) r\_cg\_userdefine.h

測定回路や $\Delta \Sigma$  A/D コンバータの設定値の変更時に合わせ込みが必要なマクロの一覧を示します。

表 6-13 r\_cg\_userdefine.h マクロ一覧

マクロ名	値	備考
MAX_16BIT_TIM_VAL	10000UL	16ビットタイマの最大値
TAU0_CH0	0x0001U	TAU0 のチャンネル 0
TAU0_CH1	0x0002U	TAU0 のチャンネル 1
TAU0_CH2	0x0004U	TAU0 のチャンネル 2
TAU0_CH3	0x0008U	TAU0 のチャンネル 3
TAU0_CH4	0x0010U	TAU0 のチャンネル 4
TAU0_CH5	0x0020U	TAU0 のチャンネル 5
TAU0_CH6	0x0040U	TAU0 のチャンネル 6
TAU0_CH7	0x0080U	TAU0 のチャンネル 7
MUX_CH0	0x01U	入力マルチプレクサのチャンネル 0
MUX_CH1	0x02U	入力マルチプレクサのチャンネル 1
MUX_CH2	0x04U	入力マルチプレクサのチャンネル 2
MUX_CH3	0x08U	入力マルチプレクサのチャンネル 3
MUX_CHECK_CH0	0x20U	DSADCRC での選択チャンネル確認(チャンネル 0)
MUX_CHECK_CH1	0x40U	DSADCRC での選択チャンネル確認(チャンネル 1)
MUX_CHECK_CH2	0x60U	DSADCRC での選択チャンネル確認(チャンネル 2)
MUX_CHECK_CH3	0x80U	DSADCRC での選択チャンネル確認(チャンネル 3)
PGA_OSR_64	0x00U	PGA の OSR=64
PGA_OSR_128	0x01U	PGA の OSR=128
PGA_OSR_256	0x02U	PGA の OSR=256
PGA_OSR_512	0x03U	PGA の OSR=512
PGA_OSR_1024	0x04U	PGA の OSR=1024
PGA_OSR_2048	0x05U	PGA の OSR=2048
PGA_GSET01_1	0x00U	GSET01 のゲイン=1
PGA_GSET01_2	0x01U	GSET01 のゲイン=2
PGA_GSET01_3	0x02U	GSET01 のゲイン=3
PGA_GSET01_4	0x03U	GSET01 のゲイン=4
PGA_GSET01_8	0x04U	GSET01 のゲイン=8
PGA_GSET02_1	0x00U	GSET02 のゲイン=1
PGA_GSET02_2	0x01U	GSET02 のゲイン=2
PGA_GSET02_4	0x02U	GSET02 のゲイン=4
PGA_GSET02_8	0x03U	GSET02 のゲイン=8
DSAD_TRG_TIMER	TAU0_CH0	$\Delta \Sigma$ A/D コンバータの起動トリガ
AMP0_TRG_TIMER	TAU0_CH4	AMP0 の起動トリガ
SW_SCAN_TIMER	TAU0_CH3	SW スキャン用タイマ
SPO2_CH	MUX_CH0	PPG/SpO <sub>2</sub> 測定に使用する入力マルチプレクサのチャンネル
SPO2_OSR	PGA_OSR_256	PPG/SpO <sub>2</sub> 測定時の OSR
OFFSET_GSET01	PGA_GSET01_1	PPG/SpO <sub>2</sub> オフセット測定時の GSET01 のゲイン
OFFSET_GSET02	PGA_GSET02_1	PPG/SpO <sub>2</sub> オフセット測定時の GSET02 のゲイン
RED_LED_GSET01	PGA_GSET01_1	PPG/SpO <sub>2</sub> RED LED 測定時の GSET01 のゲイン
RED_LED_GSET02	PGA_GSET02_1	PPG/SpO <sub>2</sub> RED LED 測定時の GSET02 のゲイン

IR_LED_GSET01	PGA_GSET01_1	PPG/SpO <sub>2</sub> IR LED 測定時の GSET01 のゲイン
IR_LED_GSET02	PGA_GSET02_1	PPG/SpO <sub>2</sub> IR LED 測定時の GSET02 のゲイン
ECG_CH	MUX_CH3	ECG 測定に使用する入力マルチプレクサのチャンネル
ECG_OSR	PGA_OSR_256	ECG 測定時の OSR
ECG_GSET01	PGA_GSET01_1	ECG 測定時の GSET01 のゲイン
ECG_GSET02	PGA_GSET02_1	ECG 測定時の GSET02 のゲイン



## 6.6.2 ユーザ設定可能パラメータ用ヘッダファイル

対象ファイルは、以下です。

- config\_parameters.h

表 6-14 config\_parameters.h マクロ一覧

マクロ名	値	備考
SAMP_PTN_A	0x00U	サンプリングパターン A
SAMP_PTN_B	0x01U	サンプリングパターン B
SAMP_PTN_C	0x02U	サンプリングパターン C
SAMP_PTN_D	0x03U	サンプリングパターン D
SAMP_PTN_E	0x04U	サンプリングパターン E
SAMP_PTN_F	0x05U	サンプリングパターン F
SAMP_PTN_G	0x06U	サンプリングパターン G
FLOAT_DATA_SIZE	sizeof(float)	float 型のデータサイズ
FILTER_OFF	0U	フィルタ OFF
FILTER_FLOAT	1U	浮動小数点型のフィルタタイプ
FILTER_FIXED	2U	固定小数点型のフィルタタイプ
INIT_FLG_FALSE	0U	初期化フラグ : False
INIT_FLG_TRUE	1U	初期化フラグ : True
SSF_DIR_NEG	0U	負入力値取得
SSF_DIR_POS	1U	正入力値取得
AMP_DELTA_SIGMA_DELAY	20000UL	アンプ起動から $\Delta \Sigma$ A/D 変換開始までのディレイ設定 [Hz] (20,000Hz -> 50us)
ECG_DIRECTION_U	0U	プラスの方向で閾値をクロスした時ピークと判定
ECG_DIRECTION_D	1U	マイナスの方向で閾値をクロスした時ピークと判定

### 6.6.3 ユーザ設定可能パラメータ用ソースファイル

対象ファイルは、以下です。

- config.c

設定方法を以下に示します。

#### (1) サンプリングパターンテーブル設定

CYCLE\_PTN\_TBL\_SIZE で定義したパターン数を登録してください。

「6.7 測定シーケンスとサンプリングパターンテーブルの設定方法」も参照してください。

```
/* table of sampling pattern */
const uint8_t g_cycle_ptn_tbl[CYCLE_PTN_TBL_SIZE] = {
    SAMP_PTN_A,          /* IR LED + ECG */
    SAMP_PTN_C,          /* offset + ECG */
    SAMP_PTN_B,          /* RED LED + ECG */
    SAMP_PTN_C,          /* offset + ECG */
};
```

#### (2) IIR/FIR フィルタ関連の設定

IIR/FIR フィルタの係数を設定してください。

##### (a) 'Offset'の設定

```
/****** IIR filter of coefficient for offset *****/
/* coefficient k of IIR for offset */
const float g_offset_iir_coefficient_k = 0.027777777;

/* coefficient b0, b1, b2, ... of IIR for offset */
const float g_offset_iir_coefficient_b[IIR_OFFSET_DEGREE_X] = {
    1,          /* b0 */
    0,          /* b1 */
    0,          /* b2 */
    0,          /* b3 */
    0,          /* b4 */
    0,          /* b5 */
    -2,        /* b6 */
    0,          /* b7 */
    0,          /* b8 */
    0,          /* b9 */
    0,          /* b10 */
    0,          /* b11 */
    1           /* b12 */
};

/* coefficient a0, a1, a2, ... of IIR for offset */
const float g_offset_iir_coefficient_a[IIR_OFFSET_DEGREE_Y] = {
    1,          /* a0 */
    -2,         /* a1 */
    1           /* a2 */
};
```

## (b) RED LED の設定

```
/****** IIR filter of coefficient for RED LED *****/
/* coefficient k of IIR for RED LED */
const float g_red_led_iir_coefficient_k = 0.0277777777;

/* coefficient b0, b1, b2, ... of IIR for RED LED */
const float g_red_led_iir_coefficient_b[IIR_RED_LED_DEGREE_X] = {
    1,          /* b0 */
    0,          /* b1 */
    0,          /* b2 */
    0,          /* b3 */
    0,          /* b4 */
    0,          /* b5 */
    -2,         /* b6 */
    0,          /* b7 */
    0,          /* b8 */
    0,          /* b9 */
    0,          /* b10 */
    0,          /* b11 */
    1           /* b12 */
};

/* coefficient a0, a1, a2, ... of IIR for RED LED */
const float g_red_led_iir_coefficient_a[IIR_RED_LED_DEGREE_Y] = {
    1,          /* a0 */
    -2,         /* a1 */
    1           /* a2 */
};
```

## (c) IR LED の設定

```
/****** IIR filter of coefficient for IR LED *****/
/* coefficient k of IIR for IR LED */
const float g_ir_led_iir_coefficient_k = 0.0277777777;

/* coefficient b0, b1, b2, ... of IIR for IR LED */
const float g_ir_led_iir_coefficient_b[IIR_IR_LED_DEGREE_X] = {
    1,          /* b0 */
    0,          /* b1 */
    0,          /* b2 */
    0,          /* b3 */
    0,          /* b4 */
    0,          /* b5 */
    -2,         /* b6 */
    0,          /* b7 */
    0,          /* b8 */
    0,          /* b9 */
    0,          /* b10 */
    0,          /* b11 */
    1           /* b12 */
};

/* coefficient a0, a1, a2, ... of IIR for IR LED */
const float g_ir_led_iir_coefficient_a[IIR_IR_LED_DEGREE_Y] = {
    1,          /* a0 */
    -2,         /* a1 */
    1           /* a2 */
};
```

## (d) ECG の設定

```
/****** IIR filter of coefficient for ECG *****/
/* coefficient k of IIR for ECG */
const float g_ecg_iir_coefficient_k = 0.0277777777;

/* coefficient b0, b1, b2, ... of IIR for ECG */
const float g_ecg_iir_coefficient_b[IIR_ECG_DEGREE_X] = {
    1,                /* b0 */
    0,                /* b1 */
    0,                /* b2 */
    0,                /* b3 */
    0,                /* b4 */
    0,                /* b5 */
    -2,              /* b6 */
    0,                /* b7 */
    0,                /* b8 */
    0,                /* b9 */
    0,                /* b10 */
    0,                /* b11 */
    1                 /* b12 */
};

/* coefficient a0, a1, a2, ... of IIR for ECG */
const float g_ecg_iir_coefficient_a[IIR_ECG_DEGREE_Y] = {
    1,                /* a0 */
    -2,              /* a1 */
    1                 /* a2 */
};
```

### 6.7 測定シーケンスとサンプリングパターンテーブルの設定方法

心拍数と脈拍数を正しく計算するため、ECG 測定と PPG/SpO<sub>2</sub> 測定を周期的に繰り返す必要があります。以下に設定方法を示します。

#### 1. 各測定を周期的に繰り返すこと

心拍数と脈拍数を正しく計算するため、ECG 測定と PPG/SpO<sub>2</sub> 測定を周期的に繰り返すようにサンプリングパターンテーブル設定してください。以下に例を示します。

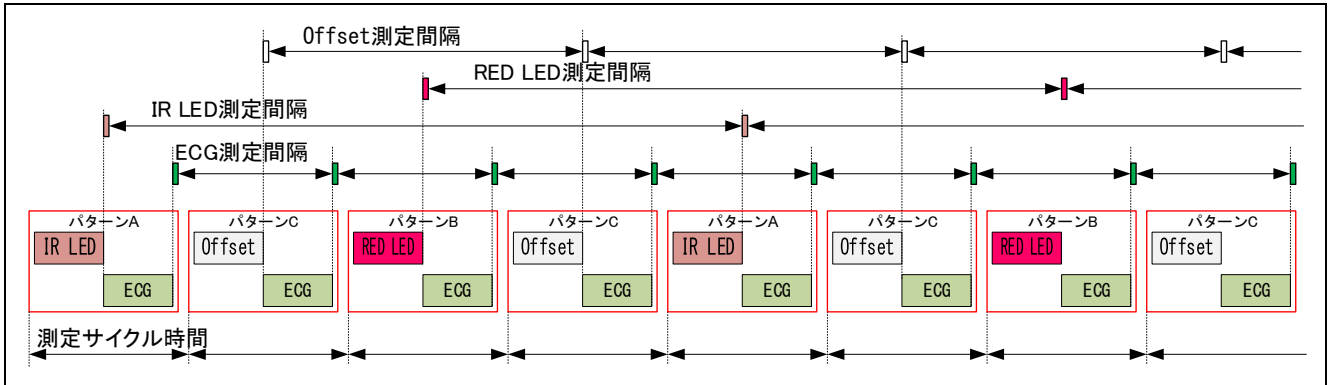


図 6-8 周期的な測定シーケンス例

#### 2. ECG と PPG/SpO<sub>2</sub> を同時測定する場合、SpO<sub>2</sub>(Offset)+ECG 測定のパターンを使用すること

ECG と PPG/SpO<sub>2</sub> を同時測定する場合、AUTOSACN によるチャンネルスキップを考慮する必要があります。

以下のシーケンスを繰り返す設定の場合、ECG 測定を毎回実行しますが、4 回目の ECG 測定開始タイミングが異なります。ECG を測定する場合、ECG 単独測定パターンを使用せず、SpO<sub>2</sub>(Offset)+ECG 測定のパターンを使用してください。

表 6-15 ECG 測定が周期的にならないシーケンス例

シーケンスパターン例	内容
(1)サンプリングパターン A	AUTOSCAN でのチャンネル 0(PPG/SpO <sub>2</sub> 測定) -> チャンネル 3(ECG 測定) の順で測定
(2)サンプリングパターン C	
(3)サンプリングパターン B	
(4)サンプリングパターン D	AUTOSCAN でのチャンネル 0 スキップにより、チャンネル 3(ECG 測定) を実施。そのため、ECG 測定開始タイミングが、他に比べて Settling time 分早い。

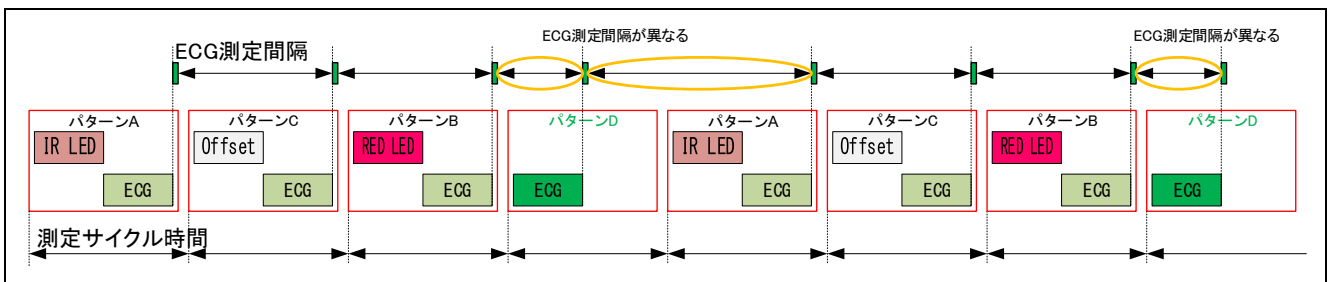


図 6-9 ECG 測定が周期的にならない例

## 7. CG の設定

CG(Code Generator : コード生成)の設定を CS+のコード生成設定画面のキャプチャにて以下に掲載します。以下に掲載されていない機能に関しては、すべて「使用しない」設定です。

### 7.1 共通/クロック発生回路

#### (1) 端子割り当て設定

端子割り当て設定
クロック設定
ブロック図
オンチップ・デバッグ設定
リセット要因確認
安全機能設定
データ・フラッシュ

端子割り当て設定

はじめに必ず設定してください。また、この設定は1度行くと変更できません。

確定する

PIORレジスタ	機能	ポート設定
PIOR02	TxD1/SO10	P50
PIOR02	RxD1/SDA10/SI10	P51
PIOR02	SCL10/SCK10	P52
PIOR10	INTP0	P137
PIOR11	INTP1	P121
PIOR13	INTP3	P30
PIOR14	INTP4	P32
PIOR15	INTP5	P122
PIOR31, PIOR30	TI05/TO05	P07
PIOR33, PIOR32	TI06/TO06	P76

#### (2) クロック設定

端子割り当て設定
クロック設定
ブロック図
オンチップ・デバッグ設定
リセット要因確認
安全機能設定
データ・フラッシュ

端子割り当て設定

動作モード設定

高速メイン・モード 4.0(V) ≤ VDD ≤ 5.5(V)
  高速メイン・モード 3.6(V) ≤ VDD ≤ 5.5(V)
  高速メイン・モード 2.7(V) ≤ VDD ≤ 5.5(V)

高速メイン・モード 2.4(V) ≤ VDD ≤ 5.5(V)

メイン・システム・クロック(fMAIN)設定

高速オンチップ・オシレータクロック(fIH)
  高速システム・クロック(fMX)

高速オンチップ・オシレータクロック設定

動作
 周波数  (MHz)

高速システム・クロック設定

動作

XT1発振(fX)
  外部クロック入力(fEX)

周波数  (MHz)

発振安定時間  52428.8 (μs)

サブシステム・クロック(fSUB)設定

動作

XT1発振(fXT)
  外部クロック入力(fEXS)

周波数  (kHz)

XT1発振回路の発振モード選択

STOP, HALTモード時のクロック供給設定

低速内蔵発振クロック(fIL)設定

周波数  (kHz)

RTC2,12ビット・インターバル・タイマ, 8ビット・インターバル・タイマ, LCD動作クロック設定

動作クロック   (kHz)

CPUと周辺クロック設定

CPUと周辺クロック(fCLK)   (kHz)

## 7.2 ポート機能

### (1) Port0

Port0	Port1	Port3	Port4	Port5	Port6	Port7	Port12	Port13
P01	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ				<input type="checkbox"/> 1を出力
P02	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ			<input type="checkbox"/> N-ch	<input type="checkbox"/> 1を出力
P03	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P04	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P05	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ				<input type="checkbox"/> 1を出力
P06	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ			<input type="checkbox"/> N-ch	<input type="checkbox"/> 1を出力
P07	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力

### (2) Port3

Port0	Port1	Port3	Port4	Port5	Port6	Port7	Port12	Port13
P30	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input checked="" type="checkbox"/> 内蔵プルアップ				<input type="checkbox"/> 1を出力
P32	<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input checked="" type="checkbox"/> 内蔵プルアップ				<input type="checkbox"/> 1を出力
P35	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P36	<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input checked="" type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P37	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ		<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力

### (3) Port5

Port0	Port1	Port3	Port4	Port5	Port6	Port7	Port12	Port13
P50	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ			<input type="checkbox"/> N-ch	<input type="checkbox"/> 1を出力
P51	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P52	<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ	<input type="checkbox"/> TTLバッファ	<input type="checkbox"/> N-ch		<input type="checkbox"/> 1を出力
P53	<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ				<input checked="" type="checkbox"/> 1を出力

## (4) Port7

Port0	Port1	Port3	Port4	Port5	Port6	Port7	Port12	Port13
P70								
<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ					<input type="checkbox"/> 1を出力
P71								
<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ					<input checked="" type="checkbox"/> 1を出力
P76								
<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ					<input checked="" type="checkbox"/> 1を出力
P77								
<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ					<input checked="" type="checkbox"/> 1を出力

## 7.3 タイマ・アレイ・ユニット

内部での設定の都合上、すべてのチャンネルを有効にしていますが、本サンプルコードで使用しているのはチャンネル0、チャンネル3およびチャンネル4のみです。

## (1) 一般設定

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
機能								
チャンネル0	インターバル・タイマ							
チャンネル1	インターバル・タイマ							
チャンネル2	インターバル・タイマ							
チャンネル3	インターバル・タイマ							
チャンネル4	インターバル・タイマ							
チャンネル5	インターバル・タイマ							
チャンネル6	インターバル・タイマ							
チャンネル7	インターバル・タイマ							

## (2) チャンネル0

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
インターバル・タイマ設定								
インターバル時間(16ビット)		2050		μs		(実際の値: 2050)		
<input type="checkbox"/> カウント開始時にINTTM00割り込みを発生する								
割り込み設定								
<input checked="" type="checkbox"/> タイマ・チャンネル0のカウント完了で割り込み発生(INTTM00)								
優先順位		レベル3(低優先順位)						



## (3) チャンネル 1

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-動作モード設定-								
<input checked="" type="radio"/> 16ビット <input type="radio"/> 上位8ビット <input type="radio"/> 下位8ビット <input type="radio"/> 上位と下位8ビット								
-インターバル・タイマ設定-								
インターバル時間(16ビット)			1000	ms	(実際の値: 1000)			
インターバル時間(上位8ビット)			100	μs				
インターバル時間(下位8ビット)			100	μs				
<input type="checkbox"/> カウント開始時にINTTM01割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル1のカウント完了で割り込み発生(INTTM01)								
優先順位			レベル2					
<input checked="" type="checkbox"/> タイマ・チャンネル1のカウント完了で割り込み発生(INTTM01H)								
優先順位			レベル3(低優先順位)					

## (4) チャンネル 2

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-インターバル・タイマ設定-								
インターバル時間(16ビット)			10	ms	(実際の値: 10)			
<input type="checkbox"/> カウント開始時にINTTM02割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル2のカウント完了で割り込み発生(INTTM02)								
優先順位			レベル2					

## (5) チャンネル 3

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-動作モード設定-								
<input checked="" type="radio"/> 16ビット <input type="radio"/> 上位8ビット <input type="radio"/> 下位8ビット <input type="radio"/> 上位と下位8ビット								
-インターバル・タイマ設定-								
インターバル時間(16ビット)			10	ms	(実際の値: 10)			
インターバル時間(上位8ビット)			100	μs				
インターバル時間(下位8ビット)			100	μs				
<input type="checkbox"/> カウント開始時にINTTM03割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル3のカウント完了で割り込み発生(INTTM03)								
優先順位			レベル3(低優先順位)					
<input checked="" type="checkbox"/> タイマ・チャンネル3のカウント完了で割り込み発生(INTTM03H)								
優先順位			レベル3(低優先順位)					

## (6) チャンネル 4

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-インターバル・タイム設定-								
インターバル時間(16ビット)		2		ms		(実際の値: 2)		
<input type="checkbox"/> カウント開始時にINTTM04割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル4のカウント完了で割り込み発生(INTTM04)								
優先順位		レベル3(低優先順位)						

## (7) チャンネル 5

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-インターバル・タイム設定-								
インターバル時間(16ビット)		10		ms		(実際の値: 10)		
<input type="checkbox"/> カウント開始時にINTTM05割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル5のカウント完了で割り込み発生(INTTM05)								
優先順位		レベル2						

## (8) チャンネル 6

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-インターバル・タイム設定-								
インターバル時間(16ビット)		2		ms		(実際の値: 2)		
<input type="checkbox"/> カウント開始時にINTTM06割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル6のカウント完了で割り込み発生(INTTM06)								
優先順位		レベル3(低優先順位)						

## (9) チャンネル 7

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3	チャンネル4	チャンネル5	チャンネル6	チャンネル7
-インターバル・タイム設定-								
インターバル時間(16ビット)		2		ms		(実際の値: 2)		
<input type="checkbox"/> カウント開始時にINTTM07割り込みを発生する								
-割り込み設定-								
<input checked="" type="checkbox"/> タイマ・チャンネル7のカウント完了で割り込み発生(INTTM07)								
優先順位		レベル3(低優先順位)						

7.4 PGA+ $\Delta \Sigma$  A/D コンバータ

## (1) 一般

一般	入力マルチプレクサ0	入力マルチプレクサ1	入力マルチプレクサ2	入力マルチプレクサ3(内部)	入力マルチプレクサ3(外部)
-動作設定-					
<input type="radio"/> 使用しない		<input checked="" type="radio"/> 使用する			
-使用入力マルチプレクサ設定-					
<input checked="" type="checkbox"/> 入力マルチプレクサ0	入力モード	差動			
<input type="checkbox"/> 入力マルチプレクサ1	入力モード	シングルエンド			
<input type="checkbox"/> 入力マルチプレクサ2	入力モード	シングルエンド			
<input type="checkbox"/> 入力マルチプレクサ3(内部)	入力モード	シングルエンド			
<input checked="" type="checkbox"/> 入力マルチプレクサ3(外部)	入力モード	差動			
-電力設定-					
SBIAS出力電圧	1.0 (V)				
-動作クロック設定-					
動作クロック	4(fCLK/6) (MHz)				
-切断検出設定-					
<input checked="" type="radio"/> 使用しない		<input type="radio"/> 使用する			
- $\Delta \Sigma$ A/Dコンバータの動作モード設定-					
<input checked="" type="radio"/> ノーマルモード		<input type="radio"/> 低電力モード			
- $\Delta \Sigma$ A/Dコンバータ起動トリガ設定-					
<input type="radio"/> ソフトウェアトリガ		<input checked="" type="radio"/> ハードウェアトリガ			
-オートスキャンモードの設定-					
<input type="radio"/> 連続スキャン		<input checked="" type="radio"/> シングルスキャン			
-安定待ち時間-					
<input checked="" type="radio"/> 128 usec + 3T		<input type="radio"/> 4T (OSR = 128, 256, 512, 1024, or 2048) or 5T (OSR = 64)			
-割り込み設定-					
<input checked="" type="checkbox"/> $\Delta \Sigma$ A/D変換完了割り込みを使用する(INTDSAD)					
優先順位	レベル3(低優先順位)				
<input checked="" type="checkbox"/> $\Delta \Sigma$ A/D変換スキャン完了割り込みを使用する(INTDSADS)					
優先順位	レベル3(低優先順位)				

(2) マルチプレクサ 0

一般 **入力マルチプレクサ0** 入力マルチプレクサ1 入力マルチプレクサ2 入力マルチプレクサ3(内部) 入力マルチプレクサ3(外部)

-PAG0設定

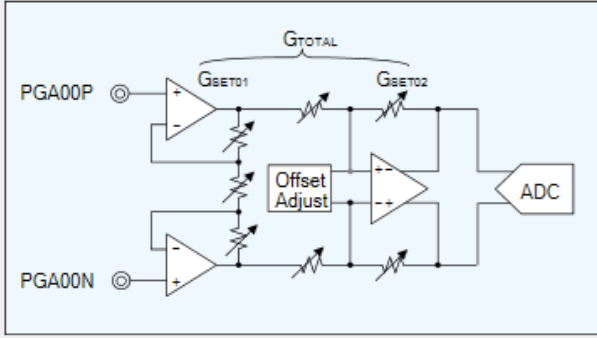
-ゲイン設定

GSET01 1 (db)

GSET02 1 (db)

-オフセット調整電圧設定

16 0 (mV)



-ΔΣA/Dコンバータ設定

オーバーサンプリング比 256 データレート 3906.250 (sps)

A/D変換回数 PGA0CTL2レジスタの設定値で1~8032回を指定

1 1 (回)

平均化処理 平均化処理を行わない

平均化データ数 8

(3) マルチプレクサ 3(外部)

一般 入力マルチプレクサ0 入力マルチプレクサ1 入力マルチプレクサ2 入力マルチプレクサ3(内部) **入力マルチプレクサ3(外部)**

-PAG0設定

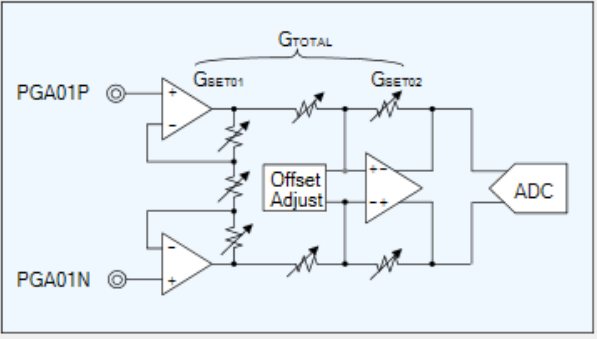
-ゲイン設定

GSET01 1 (db)

GSET02 1 (db)

-オフセット調整電圧設定

16 0 (mV)



-ΔΣA/Dコンバータ設定

オーバーサンプリング比 256 データレート 3906.250 (sps)

A/D変換回数 PGA3CTL2レジスタの設定値で1~8032回を指定

1 1 (回)

平均化処理 平均化処理を行わない

平均化データ数 8

## 7.5 アンプ・ユニット

## (1) 共通設定

共通設定	PGA1	AMP0	AMP1	AMP2
-動作設定-				
<input type="radio"/> 使用しない		<input checked="" type="radio"/> 使用する		
-使用アンプ・ユニット設定-				
<input type="checkbox"/> PGA1 	<input checked="" type="checkbox"/> AMP0	<input checked="" type="checkbox"/> AMP1	<input type="checkbox"/> AMP2	
-PGA動作モード設定-				
<input checked="" type="radio"/> ロウ・パワー・モード		<input type="radio"/> ハイスピード・モード		
-AMP0動作モード設定-				
<input type="radio"/> ロウ・パワー・モード		<input checked="" type="radio"/> ハイスピード・モード		
-AMP1およびAMP2動作モード設定-				
<input type="radio"/> ロウ・パワー・モード		<input checked="" type="radio"/> ハイスピード・モード		
-ELCトリガ設定-				
<input type="radio"/> アンプ・ユニット0:アンプELCトリガ0 アンプ・ユニット1:アンプELCトリガ1 アンプ・ユニット2:アンプELCトリガ2 アンプ・ユニット3:アンプELCトリガ3				
<input type="radio"/> アンプ・ユニット0:アンプELCトリガ0 アンプ・ユニット1:アンプELCトリガ0 アンプ・ユニット2:アンプELCトリガ1 アンプ・ユニット3:アンプELCトリガ1				
<input checked="" type="radio"/> アンプ・ユニット0:アンプELCトリガ0 アンプ・ユニット1:アンプELCトリガ0 アンプ・ユニット2:アンプELCトリガ0 アンプ・ユニット3:アンプELCトリガ0				

(2) AMP0

共通設定
PGA1
AMP0
AMP1
AMP2

-アンプ・トリガ・モード制御設定-

- ソフトウェア・トリガ・モード
- ELCトリガ・モード
- ELCおよび24ビットA/Dトリガ・モード

-スイッチ設定-

-+入力設定-

- AMP0P(AFE兼用入力4) !
- AMP0P(AFE兼用入力5) !
- 8ビットD/Aコンバータ

- -入力設定-

- AMP0N(AFE兼用入力4)
- AMP00(フィードバック)

(3) AMP1

共通設定   PGA1   AMP0   **AMP1**   AMP2

-アンプ・トリガ・モード制御設定-

- ソフトウェア・トリガ・モード
- ELCトリガ・モード
- ELCおよび24ビットA/Dトリガ・モード

-スイッチ設定-

+入力設定

- AMP1P (AFE兼用入力0) ⚠
- AMP1P (AFE兼用入力5) ⚠
- AMP1P (AFE兼用入力6)
- プラス入力には内部バイアス電圧(VBIAS)
- 12ビットD/Aコンバータ

-入力設定

- AMP1N (AFE兼用入力5)
- AMP10 (フィードバック)

## 7.6 D/A コンバータ

## (1) DA0

DA0	DA1
-D/Aコンバータ動作設定-	
<input type="radio"/> 使用しない <input checked="" type="radio"/> 使用する SBIASの出力電圧は1.8V以上に設定してください。(SBIAS出力電圧設定: 1V)	
-D/Aコンバータ動作モード設定-	
<input checked="" type="radio"/> 通常モード <input type="radio"/> リアルタイム出力モード	
-変換値設定-	
変換値	<input type="text" value="0"/> <input type="text" value="0"/> (V)
8ビット分解能: DAC0出力電圧 = SBIAS出力電圧 × (DAC0DR)/256	

## 7.7 イベントリンクコントローラ

-出力先設定-		
<input type="checkbox"/> A/Dコンバータ	イベント発生元	TAUチャネル01カウント完了/キャプチャ完了
<input type="checkbox"/> タイマ・アレイ・ユニット0チャネル0のタイマ入力	イベント発生元	TAUチャネル06カウント完了/キャプチャ完了
<input type="checkbox"/> タイマ・アレイ・ユニット0チャネル1のタイマ入力	イベント発生元	TAUチャネル02カウント完了/キャプチャ完了
<input checked="" type="checkbox"/> アンプELCトリガ0	イベント発生元	TAUチャネル04カウント完了/キャプチャ完了
<input type="checkbox"/> アンプELCトリガ1	イベント発生元	外部割り込みエッジ検出0
<input type="checkbox"/> アンプELCトリガ2	イベント発生元	外部割り込みエッジ検出1
<input type="checkbox"/> D/Aコンバータ0	イベント発生元	外部割り込みエッジ検出0
<input type="checkbox"/> D/Aコンバータ1	イベント発生元	外部割り込みエッジ検出0
<input checked="" type="checkbox"/> 24ビットΔΣ A/Dコンバータ	イベント発生元	TAUチャネル00カウント完了/キャプチャ完了
<input type="checkbox"/> アンプELCトリガ3	イベント発生元	外部割り込みエッジ検出0



## 7.8 割り込み機能

- INTP0 設定			
<input checked="" type="checkbox"/> INTP0	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP1 設定			
<input type="checkbox"/> INTP1	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP2 設定			
<input type="checkbox"/> INTP2	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP3 設定			
<input type="checkbox"/> INTP3	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP4 設定			
<input type="checkbox"/> INTP4 	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP5 設定			
<input type="checkbox"/> INTP5	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)
- INTP6 設定			
<input type="checkbox"/> INTP6	有効エッジ	立下りエッジ	優先順位 レベル3(低優先順位)

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	May.31.2019	—	新規作成

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

### 1. 静電気対策

CMOS製品の取り扱いの際は静電気防止を心がけてください。CMOS製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS製品を実装したボードについても同様の扱いをしてください。

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}(\text{Max.})$  から  $V_{IH}(\text{Min.})$  までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違えば、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

- 当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
  11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](http://www.renesas.com)

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

[www.renesas.com/contact/](http://www.renesas.com/contact/)

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。