

Traveo™ ファミリ ウォッチドッグタイマの設定方法

著者: Kenichi Sunada

関連製品ファミリ: Traveo Family

S6J3110/3120/3200/3310/3320/3330/3340/3350/3360/3370/3400 Series

関連ドキュメント: [4. 関連ドキュメント](#).

本アプリケーションノートは、Traveo™ ファミリ S6J3110/3120/3200/3310/3320/3330/3340/3350/3360/3370/3400 シリーズのソフトウェアウォッチドッグタイマとハードウェアウォッチドッグタイマの設定方法について説明しています。

1 はじめに

本アプリケーションノートは、Traveo™ ファミリ S6J3110/3120/3200/3310/3320/3330/3340/3350/3360/3370/3400 シリーズのユーザを対象としています。本アプリケーションノートでは、ソフトウェアウォッチドッグタイマとハードウェアウォッチドッグタイマの設定方法について説明します。

2 ウォッチドッグタイマの概要

ウォッチドッグタイマは、ユーザプログラムの暴走を検出するために使用する機能です。ウォッチドッグタイマには、ソフトウェアウォッチドッグタイマとハードウェアウォッチドッグタイマがあります。[Table 1](#) に双方の主な相違点を示します。

Table 1. ソフトウェアウォッチドッグタイマとハードウェアウォッチドッグタイマの主な相違点

項目	ソフトウェアウォッチドッグタイマ	ハードウェアウォッチドッグタイマ
ウォッチドッグタイマの起動	ユーザプログラムのレジスタ設定で起動	リセット解除後に自動起動
ウォッチドッグカウンタのソースクロック選択	High-speed CR clock Low-speed CR clock Main clock	High-speed CR clock Low-speed CR clock
レジスタ設定方法	ユーザプログラムによる設定	BootROM マーカによる設定 (BootROM ソフトウェアは、リセット後にユーザプログラムの前に実行される組み込みファームウェアです)
PSS モードでの動作	ユーザプログラムによって有効/無効の選択可能	動作停止

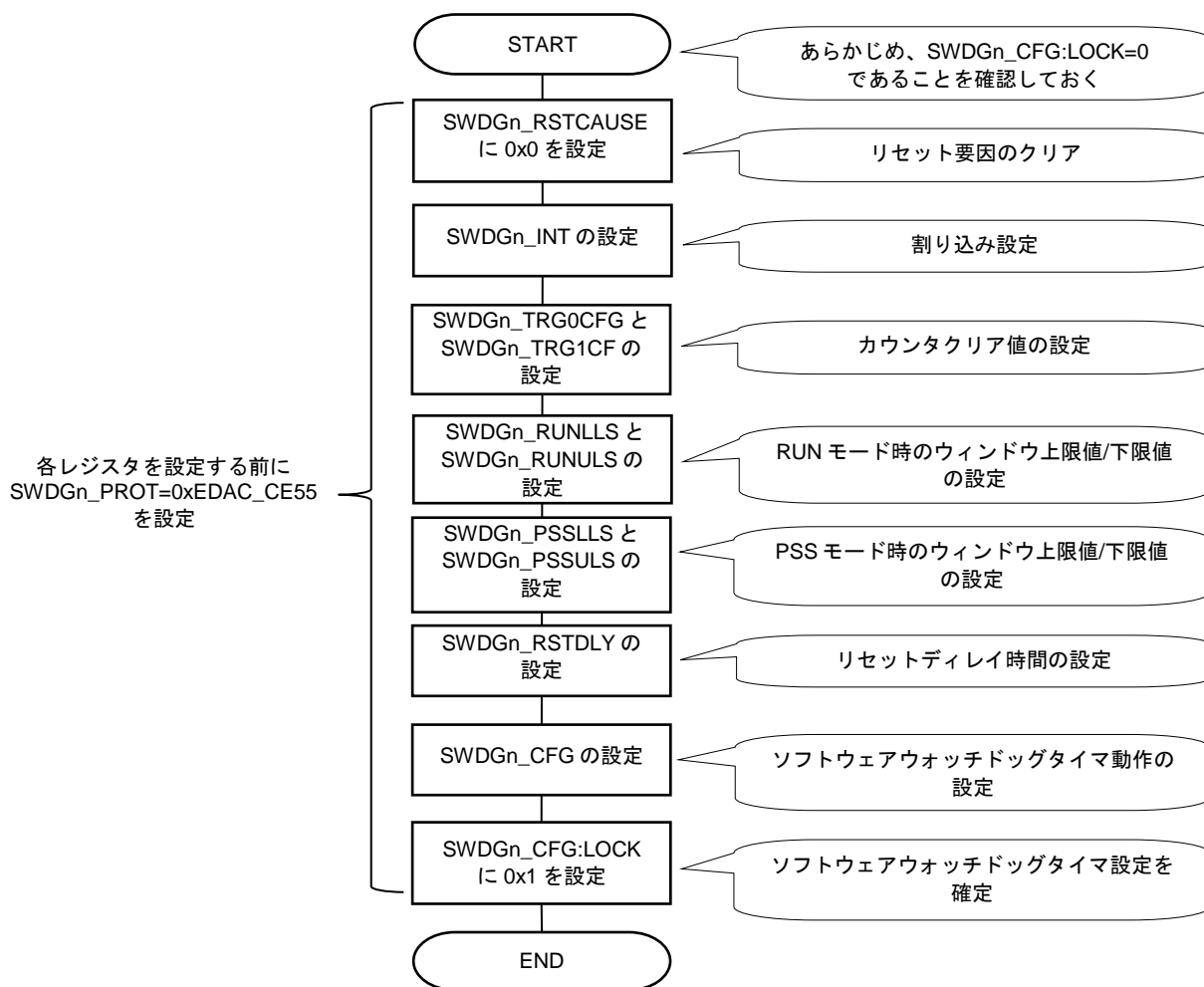
3 ウォッチドッグタイマの設定手順

本章では、ソフトウェアウォッチドッグタイマとハードウェアウォッチドッグタイマの設定方法について説明します。

3.1 ソフトウェアウォッチドッグタイマの設定手順

Figure 1 にソフトウェアウォッチドッグタイマの設定手順例を示します。

Figure 1. ソフトウェアウォッチドッグタイマ設定手順例



3.1.1 サンプルソフトのソフトウェアウォッチドッグタイマ設定手順

Figure 2 にサンプルソフトにおけるソフトウェアウォッチドッグタイマ設定手順を示します。

Figure 2. サンプルソフトのソフトウェアウォッチドッグタイマ設定手順

```

swdg.c  /* SWDG configuration registers can be written only once. */
        /* Before setting, confirm that SWDgn_CFG:LOCK=0 */
        if (SWDG0_CFG_LOCK != 0)
        {
            return Error;
        }

        /* Clear reset cause */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_RSTCAUSE = 0x00000000;

        /* Set the configuration of the interrupt */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_INT = unInt.u32Register;

        /* Set WDG trigger values */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        /* Set the counter clear value */
        SWDG0_TRG0CFG = pstcConfig->u8TriggerKey0;
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        /* Set the counter clear value */
        SWDG0_TRG1CFG = pstcConfig->u8TriggerKey1;

        /* Set window lower limit for RUN mode */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_RUNLLS = pstcConfig->stcRunModeSettings.u32WindowLowerLimit;

        /* Set window upper limit for RUN mode */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_RUNULS = pstcConfig->stcRunModeSettings.u32WindowUpperLimit;

        /* Set window lower limit for PSS mode */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_PSSLIS = pstcConfig->stcPssModeSettings.u32WindowLowerLimit;

        /* Set window upper limit for PSS mode */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_PSSULS = pstcConfig->stcPssModeSettings.u32WindowUpperLimit;

        /* Set delay cycle for Reset or NMI. */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_RSTDLY_WDGRSTDLY = pstcConfig->u16ResetDelay;

        /* Set the software watchdog timer configuration */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_CFG = unCfg.u32Register;

        /* Lock the software watchdog timer configuration */
        SWDG0_PROT = SWDG_PROT_UNLOCK;
        SWDG0_CFG_LOCK = 1;

```

各レジスタを設定
する前に
SWDgn_PROT=0
xEDAC_CE55 を
設定

3.1.2 ソフトウェアウォッチドッグタイマのクリア手順

あらかじめ SWDG0_TRG0CFG と SWDG0_TRG1CFG に設定した値を、各々SWDG0_TRG0 と SWDG0_TRG1 に書き込むことで、ソフトウェアウォッチドッグタイマのカウンタをクリアします。

Figure 3 にサンプルソフトにおけるソフトウェアウォッチドッグタイマのクリア手順を示します。

Figure 3. サンプルソフトのソフトウェアウォッチドッグタイマのクリア手順

```
swdg.c  en_result_t Swdg_Clear(uint8_t u8TrgVal0, uint8_t
        u8TrgVal1)
        {
            uint32_t cnt;

            /* -----Check parameters and conditions ----- */

            cnt = Swdg_GetCounterValue();
            if (cnt < SWDG0_RUNLLS )
            {
                return Error;
            }

            /* ----- Access registers ----- */

            /* Clear SWDG */
            /* Clear counter of software watchdog timer */
            IRQ_DISABLE_LOCAL();
            SWDG0_TRG0 = u8TrgVal0;
            SWDG0_TRG1 = u8TrgVal1;
            IRQ_RESTORE();

            return Ok;
        }
```

3.2 ハードウェアウォッチドッグタイマの設定手順

Table 2 にハードウェアウォッチドッグタイマ設定用の BootROM マーカの一覧表を示します。ハードウェアウォッチドッグタイマの設定のために、これらのマーカ(WDR)を使用します。

WDR_CEM マーカが有効な場合、WDR によって定義された設定に従ってタイマは開始されます。WDR_CEM マーカが無効の場合、ハードウェアマニュアルに記載されている初期設定に従ってタイマは開始されます。

Table 2 ハードウェアウォッチドッグタイマ設定用 BootROM マーカ(WDR)一覧表

マーカ名	概要
WDR_INTM	割り込み設定用マーカ
WDR_TRG0CFGM	カウンタクリア値設定用マーカ
WDR_TRG1CFGM	カウンタクリア値設定用マーカ
WDR_RUNLLM	RUN モード時の下限値設定用マーカ
WDR_RUNULM	RUN モード時の上限値設定用マーカ
WDR_PSSLLM	PSS モード時の下限値設定用マーカ
WDR_PSSULM	PSS モード時の上限値設定用マーカ
WDR_RSTDLYM	リセットディレイ時間設定用マーカ
WDR_CFGM	ハードウェアウォッチドッグタイマ動作設定用マーカ
WDR_CEM	ハードウェアウォッチドッグタイマ設定有効用マーカ

3.2.1 サンプルソフトのハードウェアウォッチドッグタイマ設定手順

Figure 4 にサンプルソフトにおけるハードウェアウォッチドッグタイマ設定手順を示します。

Figure 4. サンプルソフトのハードウェアウォッチドッグタイマ設定手順

flash_marker.asm

```
*****
**
**
; * 5.4 Watchdog Description Record (WDR) *
; *****
**
; * Hardware Watchdog Interrupt Configuration Marker
; * Reset Enable Marker
; * If set to MARKER_ENABLE, a reset is generated when a watchdog error
; * occurs.
; * Otherwise, an NMI interrupt is generated in the same condition.
; * < set to MARKER_DISABLE or MARKER_ENABLE
#define WDR_INTM_RSTENM_ENABLE (MARKER_ENABLE)
; * Prior Warning Interrupt Enable Marker
; * If set to MARKER_ENABLE, prior warning interrupt is enabled.
; * Otherwise, it is disabled.
; * < set to MARKER_DISABLE or MARKER_ENABLE
#define WDR_INTM_IRQENM_ENABLE (MARKER_ENABLE)

; * Watchdog Trigger 0 Configuration Marker
; * Following values are used to clear hardware watch dog timer.
; * Valid range: 0x00...0xFF
#define WDR_TRG0CFGM_WDGTG0CFGM_SETTING (0x00)
#define WDR_TRG1CFGM_WDGTG0CFGM_SETTING (0x00)

; * Hardware Watchdog Lower Limit RUN Setting Marker
; * This value defines the lower border of the Watchdog window for RUN
; * state.
; * When set to "0x00000000", the window function does not work.
#define WDR_WDR_RUNLLM_SETTING (0x00000000)

; * Hardware Watchdog Upper Limit RUN Setting Marker
; * This value defines the upper border of the Watchdog window for RUN
; * state.
#define WDR_WDR_RUNULM_SETTING (0x01000000)

; * Hardware Watchdog Lower Limit PSS Setting Marker
; * This value defines the lower border of the Watchdog window for PSS
; * state.
; * When set to "0x00000000", the window function does not work.
#define WDR_WDR_PSSLML_SETTING (0x00000000)

; * Hardware Watchdog Upper Limit PSS Setting Marker
; * This value defines the upper border of the Watchdog window for PSS
; * state.
#define WDR_WDR_PSSULM_SETTING (0x80000000)

; * Hardware Watchdog Reset Delay Counter Marker
; * This value defines the delay to be inserted
; * in Watchdog reset/NMI generation in case of a Hardware Watchdog
; * error.
; * valid range: 0 to 65535 (0xFFFF)
#define WDR_RSTDLYM_WDGRSTDLYM_SETTING (0x0000)

; * Clock Selection Marker
; * This value is used to select a source clock of the watchdog counter.
; * < set to WDR_CFGM_CLKSELM_LOW_SPEDD_CR or
; * WDR_CFGM_CLKSELM_HIGH_SPEDD_CR
#define WDR_CFGM_CLKSELM_CLK_SELECTION (WDR_CFGM_CLKSELM_LOW_SPEDD_CR)

; * Hardware Watchdog Configuration Enable Marker
; * If set to MARKER_ENABLE, the hardware watchdog is started based
; * on the settings defined with the WDR.
; * Otherwise, the hardware watchdog operates based on the default
; * settings.
; * < set to MARKER_DISABLE or MARKER_ENABLE
// #define WDR_CEM_ENABLE (MARKER_DISABLE)
#define WDR_CEM_ENABLE (MARKER_ENABLE)
```

リセット生成有効

事前警告割り込み有効

カウンタクリア値として 0x00 を設定

RUN モード時の
ウィンドウ下限値
の設定(無効)

RUN モード時のウィ
ンドウ上限値の設定

PSS モード時のウィ
ンドウ下限値の設定
(無効)

PSS モード時のウィ
ンドウ上限値の設定

リセットディレイ時
間の設定

ソースクロックの選
択 (low-speed CR)

ハードウェアウォッ
チドッグタイマの
マーカー設定有効

3.2.2 ハードウェアウォッチドッグタイマのクリア手順

WDR_TRG0CFGM と WDR_TRG1CFGM は、BootROM マーカによって設定されます。あらかじめ HWDG_TRG0CFGM と HWDG_TRG1CFGM に設定した値を、各々HWDG_TRG0 と HWDG_TRG1 に書き込むことで、ハードウェアウォッチドッグタイマのカウンタをクリアします。

Figure 5 にサンプルソフトにおけるハードウェアウォッチドッグタイマのクリア手順を示します。

Figure 5. サンプルソフトのハードウェアウォッチドッグタイマのクリア手順

```
main.c int main(void)
{
    /* Finalize initialization to default settings. */
    /* (this will do IRQ and NMI initialization and global IRQ/NMI enable) */
    Start_Init();

    /* Endless loop */
    for (;;)
    {
        ClearWatchdog();
    }

    static void ClearWatchdog(void)
    {
        /* Clear hardware watchdog */
        /* Clear counter of hardware watchdog timer */
        IRQ_DISABLE_LOCAL();
        HWDG_TRG0 = HWDG_TRG0_VALUE;
        HWDG_TRG1 = HWDG_TRG1_VALUE;
        IRQ_RESTORE();
    }
}
```

4 関連ドキュメント

- S6J311E/D/C/B Series Datasheet (Doc. No.002-05681)
- S6J311A/9/8 Series Datasheet (Doc. No.002-04632)
- S6J3110 Series Hardware Manual (Doc.No.002-10667)
- S6J3120 Series Datasheet (Doc.No.002-04863)
- S6J3120 Series Hardware Manual (Doc.No.002-04855)
- S6J3200 Series Datasheet (Doc.No.002-05682)
- S6J3200 Series Hardware Manual (Doc.No.002-04852)
- S6J32E/F/G Series Datasheet (Doc.No.002-10689)
- S6J32E/F/G Series Hardware Manual (Doc.No.002-12500)
- Traveo Family Hardware Manual Platform Part for S6J3200 Series (Doc.No.002-04854)
- S6J3310/20/30/40 Series Datasheet (Doc.No.002-10635)
- S6J3350 Series Datasheet (Doc.No.002-10634)
- S6J3310/20/30/40/50 Series Hardware Manual (Doc.No.002-10185)
- Traveo Family HardwareManual Platform Part for S6J3310/3320/3330/3340/3350 Series (Doc.No.002-07884)
- S6J3360/70 Series Datasheet (Doc.No.002-03359)
- S6J3360/70 Series Hardware Manual (Doc.No.002-18302)
- Traveo Family HardwareManual Platform Part for S6J3360/3370 Series (Doc.No.002-07884)
- S6J3400 Series Datasheet (Doc.No.001-97829)
- S6J3400 Series Hardware Manual (Doc.No.002-09919)
- Traveo Family Hardware Manual Platform Part for S6J3400 Series (Doc.No.002-07884)

改訂履歴

ドキュメント名: AN204452 - Traveo™ファミリ ウォッチドッグタイマの設定方法

ドキュメント番号: 002-04453

Revision	ECN	変更者	発行日	変更内容
**	—	KHAS	06/24/2015	Initial Release
*A	5435976	HITO	09/14/2016	これは英語版 002-04452 Rev.*C を翻訳した日本語版です。
*B	5835008	HITO	07/27/2017	これは英語版 002-04452 Rev.*D を翻訳した日本語版です。

ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーション ページ](#)をご覧ください。

製品

ARM® Cortex® Microcontrollers	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
照明&電力制御	cypress.com/powerpsoc
メモリ	cypress.com/memory
PSoC	cypress.com/psoc
タッチ センシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス/RF	cypress.com/wireless

PSoC® ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

サイプレス開発者コミュニティ

[コミュニティ](#) | [フォーラム](#) | [ブログ](#) | [ビデオ](#) | [トレーニング](#)
[Components](#)

テクニカルサポート

cypress.com/support



Cypress Semiconductor
 198 Champion Court
 San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2015-2017. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社（以下「Cypress」という。）に帰属する財産である。本書面（本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア（以下「本ソフトウェア」という。）を含む）は、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ、（直接又は再販売者及び販売代理店を介して間接のいずれかで）本ソフトウェアをバイナリコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア（Cypress により提供され、修正がなされていないもの）が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス（サブライセンスの権利を除く）を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示をとわず、いかなる保証（商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない）も行わない。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報（あらゆるサンプルデザイン情報又はプログラムコードを含む）は、参照目的のためのみに提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用（以下「本目的外使用」という。）のためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部をとわず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本来目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任（人身傷害又は死亡に基づく請求を含む）から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, Capsense, EZ-USB, F-RAM, 及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、cypress.com を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。

ARM and Cortex are the registered trademarks of ARM Limited in the EU and other countries.