

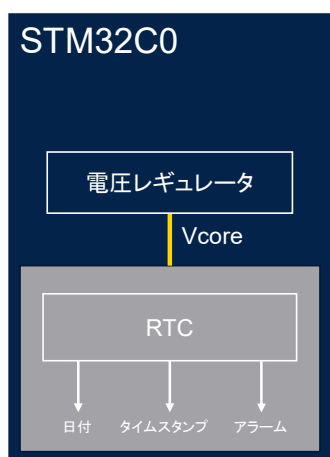


STM32C0 - RTC

リアルタイム・クロック
レビジョン1.0

STM32C0リアルタイム・クロックのプレゼンテーションへようこそ。ここでは、非常に正確なタイムベースを提供するために使用されているこのペリフェラルの主な機能について説明します。

.



- RTCは、アラーム付きの超低消費電力ハードウェアカレンダーを搭載
- 供給電圧が動作範囲内にある限り、デバイスのステータスに関係なく、RTCが停止しない
- RTCイベント(アラーム、タイムスタンプ)は割込みを生成し、デバイスを低電力モードからウェイクアップさせる

アプリケーション側の利点

- 超低消費電力: 1.13 μ A/MHz
- STOPモード: 300nA
- ハードウェアBCDカレンダーによりソフトウェアの負荷が低減

RTCペリフェラルは、アラーム付きの超低消費電力カレンダーを備えており、SLEEPモードとSTOPモードで動作します。供給電圧が動作範囲内にある限り、デバイスの状態に関係なく、RTCが停止することはありません。RTCの消費電流は1MHzあたりわずか1.13 μ Aです。ハードウェアカレンダーは、特に日付と時刻を表示する必要がある場合のソフトウェア負荷を軽減するために、2進化10進数(BCD)フォーマットで提供されます。

主な機能

- BCDフォーマットのサブセカンド、秒、分、時、曜日、日、月、年
- 動作中にプログラム可能なサマータイム補正
- ウェイクアップ割込み機能を備えたプログラム可能なアラーム
- カレンダの精度を向上させるためにリファレンスクロックソース(50Hzまたは60Hz)を使用可能
- デジタル較正回路により0.954ppmの精度を達成
- サブセカンドの精度でのカレンダの内容の保存に使用できるタイムスタンプ機能(1イベント)
- LSEによってクロックされると、リセットソースがパワーオンリセットとは異なる場合、RTCはシステムリセット下で動作し続ける



RTCの主な機能は次の通りです。

秒、分、時、曜日、日付、月、年を2進化10進数形式で提供します。サブ秒は2進化10進数形式で提供されます。

サマータイムを管理するために、動作中にカレンダに1時間を追加または削除します。

SLEEP、STOPモードからマイクロプロセッサをウェイクアップできる、1つのプログラム可能なアラーム。

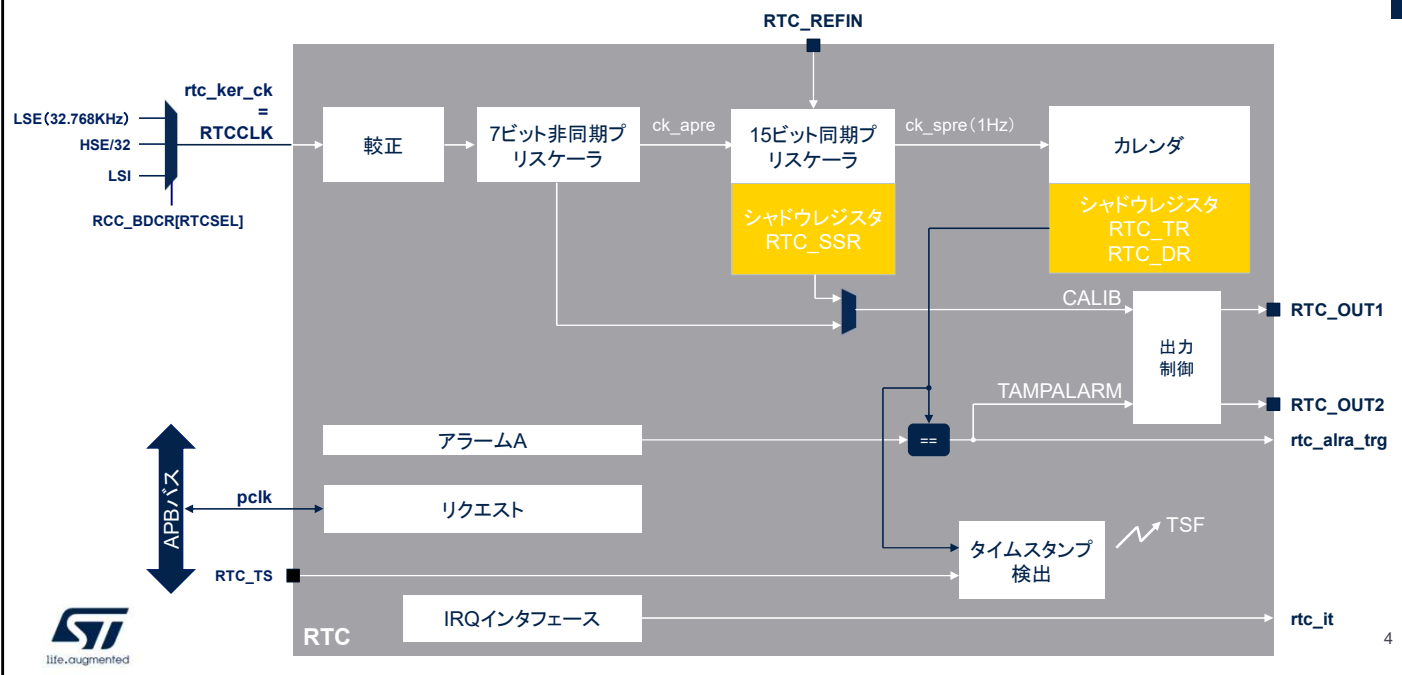
カレンダは、50Hzまたは60Hz電源のリファレンスクロックソースにより較正できます。

0.954ppm分解能でクリスタル精度の補正が可能なデジタル較正回路。外部イベントに応じて、タイムスタンプレジスタにカレンダの内容を保存するためのタイムスタンプ機能。

リセットイベントに関しては、カレンダシャドウレジスタとRTCステータスレジスタの一部のビットは、利用可能なすべてのシステムリセットソースによってデフォルト値にリセットされます。

現在のカレンダレジスタを含む他のレジスタは、パワーオンリセットによってデフォルト値にリセットされ、システムリセットの影響は受けません。さらに、LSEによってクロックされる場合、リセットソースが電源投入時と異なる場合、RTCはシステムリセット中でも動作し続けます。

ブロック図



これは、RTCのブロック図です。

RTCには2個のクロックソースがあります。RTCクロック(RTCCLK)はRTCタイマカウンタに使用され、APBクロックはRTCレジスタの読み出しおよび書き込みアクセスに使用されます。RTCクロックは、32分周された高速外部オシレータ(HSE)、低速外部オシレータ(LSE)、または低速内部オシレータ(LSI)のいずれかを使用できます。

RTCクロックは、最初に、ck_apreクロックを供給するプログラム可能な7ビット非同期プリスケラにより分周されます。ほとんどのRTCにはck_apre周波数のクロックが供給されるので、消費電力を減らすために、高い非同期分周値をセットすることをお勧めします。

デフォルト値は128です。

次に、15ビットのプログラム可能な同期プリスケラがck_spreクロックを提供します。

時刻と日付のBCDレジスタを1秒単位で更新するには、ck_spreクロックは1Hzである必要があります。

1秒未満のレジスタ分解能はck_apre周波数によって定義されます。デフォルトでは、RTCクロック周波数が32768Hzの場合、これは256Hzです。SSRレジスタの分解能は、非同期プリスケラ値を減らすことによって増加します。

非同期プリスケラもバイパスできます。この場合、1秒未満のレジスタ分解能はRTCクロック周波数によって定義されます。

RTC_TS入力でプログラム可能なエッジが検出されると、現在のカレンダーがタイムスタンプレジスタに保存されます。

現在のデータがアラームAにプログラムされた時間と等しくなると、アラームイベントが発生し、RTC_OUTピンに送信できます。

512Hzまたは1Hzクロックは、キャリブレーションの目的でRTC_OUTピンの1つで駆動できます。

この図では、シャドウレジスタはAPBクロックドメインに属しています。これについては、このプレゼンテーションの後半で説明します。

RTC_OUT1とRTC_TSは同じピンにマッピングされていることに注意してください。

RTCレジスタ書込み保護

セキュアなRTC初期化

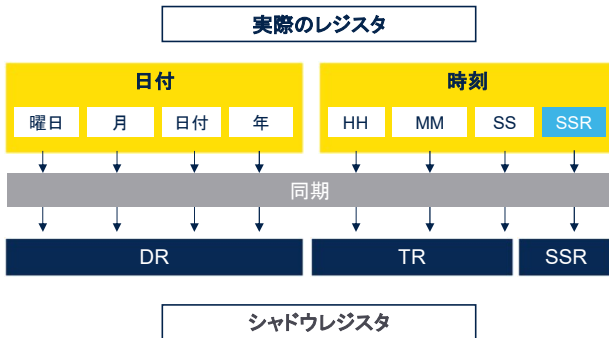
- パワーオンリセット後、一部のRTCレジスタは書込み保護される
 - 保護されたRTCレジスタへの書込みは、書込み保護レジスタRTC_WPRにキーを書き込むことで有効になる
- RTC初期化モードに入るための特定のソフトウェアシーケンス
 - カレンダーレジスタとプリスケアラの初期化に使用
 - RTC_ICSRレジスタのINITビットを1に設定すると、初期化モードに移行する
 - このモードでは、カレンダーカウンタが停止し、その値を更新できるようになる



RTCは、安全な方法を使用して初期化されます。
RTCレジスタは不正な書込みアクセスを避けるために書込み保護されています。
RTC書込み保護レジスタに特定のシーケンスを書き込む必要があります。
クロックプリスケアラ値またはカレンダー値を変更するには、初期化モードに入る必要があります。

すべての低消費電力モード、VBAT、リセットでアクティブ

- シャドウレジスタ(時間レジスタと日付レジスタ)を通じて初期化が行われる



life.augmented

- カレンダーの読出し:

- BYPSHAD=0: シャドウレジスタを読み出す
 - STOP/STANDBY/SHUTDOWNモードの終了時にシャドウレジスタを更新するために最大1RTCCLKサイクルの遅延が発生
 - RTC_SSRまたはRTC_TRのどちらかを読み出すと、高次カレンダーシャドウレジスタの値はRTC_DRが読み出されるまでロックされる
- BYPSHAD=1: シャドウレジスタをバイパス
 - カレンダーの読出しによりカレンダーカウンタに直接アクセスする
 - ソフトウェアはすべてのカレンダーレジスタを2回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要がある

6

RTCカレンダーは、低電力モードのSLEEP、STOP中、およびリセット中も実行され続けます。

時刻レジスタと日付レジスタの初期化は、APBクロックドメインに属するシャドウレジスタを介して実行されます。サブセカンドレジスタは初期化できません。

RTCCLK周期ごとに、現在のカレンダー値がシャドウレジスタにコピーされます。

カレンダーのサブ秒、時刻、および日付レジスタの内容は、2つの異なるモードで読み取ることができます。

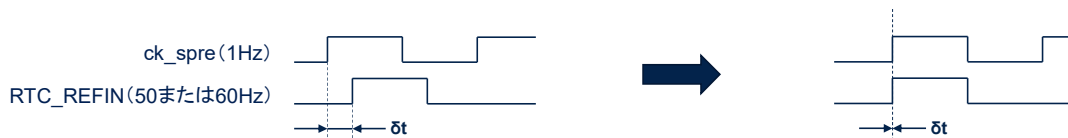
バイパスシャドウレジスタ制御ビットがクリアされると、シャドウレジスタが読み取られます。このモードの利点は、3つのレジスタすべての一貫性が保証されることです。RTC_SSRまたはRTC_TRのいずれかを読み取ると、RTC_DRが読み取られるまで上位カレンダーシャドウレジスタの値がロックされます。

このモードの欠点は、SLEEPモードまたはSTOPモードを終了するとき、シャドウレジスタが最後のカレンダーレジスタ値で更新されるようにするために、ソフトウェアが同期遅延を待つ必要があることです。この同期遅延は、最大4RTCクロック周期になる可能性があります。また、このシャドウモードでRTCカレンダーレジスタを適切に読み取るには、APB1クロック周波数がRTCクロック周波数の7倍以上である必要があります。

バイパスシャドウレジスタ制御ビットがセットされている場合は、実際のカレンダーレジスタが直接読み出されます。このモードの利点は、同期遅延を待つ必要がないことです。欠点は、同期の問題が原因で読み出された値が誤っていたり一貫性がない可能性があるため、2回読み出して前回の読出し値と比較して、それらが正しく整合性があることを確認する必要があるという点です。

RTCカレンダー機能

- 「サマータイム」は、自動的に1時間を加算または減算することによって管理される
- サブセカンドの分解能でオフセットを加算／減算することで最大1秒のカレンダー同期
 - リモートクロックによる同期が可能
- リファレンスクロック検出: より正確な秒のクロックソース (50または60Hzの電源) を使用して、カレンダーの長期的な精度を向上させることができる
 - リファレンスクロックが自動的に検出され、カレンダー精度の向上に使用される
 - リファレンスクロックが使用できなくなった場合には、自動的にLSEクロックがカレンダーの更新に使用される



このスライドでは、カレンダーの主な機能について説明します。

サマータイムは、ソフトウェアで自動的に1時間を加算または減算することによって管理できます。

RTCはリモートクロックと高精度で同期できます。

動作中にオフセットをサブセカンドレジスタに加減算することにより、ck_spreクロック分解能で、RTCクロックをリモートクロックに同期させることができます。この機能は、RFアプリケーションで一般的に使用されます。

長期的なカレンダーの精度を向上させるためにリファレンスクロック (50Hzまたは60Hzの電源) を使用できます。

RTC_REFINピンに接続された基準クロックは自動的に検出されます。

LSEクロックが不正確なために1Hzのクロックがずれた場合、RTCは1Hzのクロックを少しシフトさせ、その後の1Hzのクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダーはリファレンスクロックと同様に正確になります。

リファレンスクロックが使用できない場合、自動的にLSEクロックがカレンダーの更新に使用されます。

高精度デジタル較正

クリスタルの精度を補償

- 設定可能なウィンドウ内に適切に分散される、N個(設定可能)の32KHzクロックパルスのマスキング/追加から構成される
- 動作中に較正值を変更できる
- クリスタル周波数と較正結果を測定するために、1Hz出力を提供

較正ウィンドウ	精度	全範囲
8秒	±1.908ppm	[-487.1ppm,+488.5ppm]
16秒	±0.954ppm	[-487.1ppm,+488.5ppm]
32秒	±0.477ppm	[-487.1ppm,+488.5ppm]



8

デジタル較正は、クリスタル精度と、温度および経年変化による精度のばらつきの補償に使用されます。

設定可能なウィンドウ内に適切に分散される、プログラム可能な数のRTCクロックサイクルのマスキングまたは追加から構成されます。較正值は、検出された温度変化などに応じて動作中に変更できます。較正值を適用する前後のクリスタル周波数を測定するために、1Hzの較正出力信号が提供されます。

ここに示した精度は、デジタル較正の分解能です。較正ウィンドウサイズは、8秒、16秒、32秒の中から設定可能です。32秒の較正ウィンドウでの精度は±0.477ppmとなります。

全補正範囲は、-487~488ppmです。精度分解能は、較正ウィンドウのサイズに比例します。

アプリケーションの最終的な精度は、クリスタルパラメータ精度、温度検出精度、ソフトウェア較正手順が開始された頻度などに依存します。

較正ウィンドウの精度に到達するためには、測定ウィンドウは較正ウィンドウの倍数である必要があります。

RTCカレンダー機能

- タイムスタンプ
 - RTC_TSEピンでタイムスタンプイベントが検出されると、カレンダーがタイムスタンプレジスタに保存される
 - RTC_CRLレジスタのTSIEビットをセットすることで、タイムスタンプイベント発生時に割込みを生成
- タイムスタンプフラグ (TSF) がすでにセットされているときに新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) フラグがセットされる

タイムスタンプ機能が使用できます。タイムスタンプI/Oでイベントが発生すると、サブセカンド、時刻、日付の各値がタイムスタンプレジスタに保存されます。

タイムスタンプフラグが設定されているときにタイムスタンプイベントが発生すると、タイムスタンプオーバーフローフラグが設定されます。

この場合、タイムスタンプレジスタは前のイベントのタイムスタンプを維持します。

RTCのプログラム可能なアラーム

カレンダー値に基づく柔軟性の高いアラーム

- カレンダーのサブセカンド、秒、分、時、日付が、アラームレジスタにプログラムされた値と一致した場合に、アラームフラグがセットされる
 - アラームによりスリープが解除され、低電力モードが停止
- アラームイベントを、極性が設定可能な特定の出カピンRTC_OUTに送ることもできる
- カレンダーのサブセカンド、秒、分、時、日付の各フィールドは、(マスクの有無を)個別に選択できる
 - マスクすると、周期的なアラーム割込みの設定が可能となる



10

RTCには、カレンダーの値との比較に基づく、柔軟性の高い2つのアラームが搭載されています。カレンダーのサブセカンド、秒、分、時、日付が、アラームレジスタにプログラムされた値と一致した場合に、アラームフラグがセットされます。

アラームイベントは、SLEEPモード、STOPモードからデバイスをウェイクアップできます。

アラームイベントを、極性が設定可能な特定の出カピンRTC_OUTに送ることもできます。

カレンダーアラームのサブセカンド、秒、分、時、日付の各フィールドを、比較のために個別にマスクしたりしなかったりできます。マスクを使用する場合、周期的なアラームが生成されます。

割込みイベント	説明
アラームA	カレンダー値がアラームA値に一致した場合にセット
タイムスタンプ	タイムスタンプイベントの発生時にセット

- イベントフラグはRTC_SRレジスタにある
- 割込みマスクされたフラグはRTC_MISRレジスタにある

2つのRTCイベントにより割込みが生成される場合があります。これらの割込みイベントは、マイクロコントローラをSLEEPモードおよびSTOPモードからウェイクアップすることができます。カレンダー値がアラームA値と一致すると、アラームA割込みが設定されます。タイムスタンプ割込みは、タイムスタンプイベントが発生すると設定されます。

低消費電力モード

モード	説明
RUN	アクティブ
SLEEP	アクティブ ・ RTC割込みによって、デバイスはSLEEPモードを終了する
STOP	クロックがLSEまたはLSIによって供給される場合、アクティブ ・ RTC割込みによって、デバイスはSTOPモードを終了する
STANDBY	RTCの電源がオフになっているため、STANDBYモードを終了した後に再初期化する必要がある
SHUTDOWN	RTCの電源がオフになっているため、SHUTDOWNモードを終了した後に再初期化する必要がある

RTCペリフェラルは、RUN、SLEEP、およびSTOPモードでアクティブです。

RTC割込みにより、デバイスはSLEEPモードとSTOPモードを終了します。

STOPモードでは、RTCのクロック供給にLSEまたはLSIクロックのみを使用できます。

STANDBYおよびSHUTDOWNモードでは、RTCの電源はオフになります。

STM32G0との違い

- STM32C0に実装されたRTCは、STM32G0に存在するRTCのサブセット:
 - STM32G0には2つのアラームAとBがある
 - STM32C0はアラームAのみサポート
 - STM32G0には、RTCモジュールにタンパ機能が統合されてる
 - STM32C0にはタンパ機能は統合されていない
 - RTCは、STM32G0のSTANDBYモードおよびSHUTDOWNモードでもアクティブな状態を維持する
 - STM32C0には当てはまらない



13

STM32C0のRTCは、STM32G0に存在するRTCのサブセットです。

コストと消費電力の理由から削除された機能は次のとおりです。

：

- 2つ目のアラームB
- タンパ機能
- バックアップドメインと、STANDBYモードおよびSHUTDOWNモード時にRTCをアクティブで維持する機能

- DBG_RTC_STOPビット:コア停止時にRTCカウンタは停止する

デバッグを目的としてMCUデバッグインタフェースで1ビットを使用できます。これにより、コアが停止したときにRTCカウンタを停止できます。

関連ペリフェラル

- 次のRTCにリンクされているペリフェラルトレーニングを参照
 - リセットおよびクロック制御(RCC)
 - 電源制御(PWR)
 - 拡張割込みコントローラ(EXTI)



これは、リアルタイム・クロックに関連するペリフェラルのリストです。詳細については、必要に応じてこれらのペリフェラルトレーニングを参照してください。

- リセットおよびクロック制御
- 電源制御
- 拡張割込みコントローラ

Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



ありがとうございました。