

インテグレートドタイマユニット

STEP22 からは、インテグレートドタイマユニット (ITU) を使っていきます。
これは、H8/3052F に内蔵されている機能です。
使い方は A/D 変換器や D/A 変換器と似ているのですぐに理解できると思います。

これまでの課題では、ループを使ったソフトウェアタイマを使ってきました。しかし、このようなプログラムでは、CPUが無駄な処理をしている時間が多く、タスクを効率よく処理することができません。また、音階再現に必要な μs 単位パルスは、ソフトウェアタイマの限界のため正確に計測できませんでした。

22.1 ITU とタイマカウンタ

紹介が遅れてすみません。H8/3052F には、高機能タイマである ITU (Integrated Timer Unit) が内蔵されています。ITU を使えば、正確で、CPU に無駄な処理をさせることなく、時間計測が可能になります。

ITU は 16 ビットのカウンタで、チャンネルが 5 つあります。

本テキストでは、5 つあるチャンネルからチャンネル 0 を使う方法を学習しましょう。

タイマカウンタは、CPU の動作クロックとは別の専用のクロックでカウントされます。

22.2 フリーランニングカウンタ動作について

STEP22 では、ITU の基本的な使い方であるフリーランニングカウンタ動作の使い方を学習します。フリーランニングカウンタ動作は、動作を開始するとひたすらカウントし続けるモードです。ITU チャンネル 0～4 のカウンタはリセット直後は、すべてフリーランニングカウンタの設定となっています。

ITU 関係のレジスタには以下のものがあります。その中で、フリーランニングカウンタ動作させるのに必要なレジスタについて説明していきます。

- 10.2.1 タイマスタートレジスタ 【TSTR】
- 10.2.2 タイマシンクロレジスタ 【TSNC】
- 10.2.3 タイマモードレジスタ 【TMDR】
- 10.2.4 タイマファンクションコントロールレジスタ 【TFCR】
- 10.2.5 タイマアウトプットマスタイネーブルレジスタ 【TOER】
- 10.2.6 タイマアウトプットコントロールレジスタ 【TOCR】
- 10.2.7 タイマカウンタ 【TCNT】
- 10.2.8 ジェネラルレジスタ A、B 【GRA、GRB】 ※ STEP26 で解説
- 10.2.9 バッファレジスタ A、B 【BRA、BRB】
- 10.2.10 タイマコントロールレジスタ 【TCR】
- 10.2.11 タイマ I/O コントロールレジスタ 【TIOR】
- 10.2.12 タイマステータスレジスタ 【TSR】
- 10.2.13 タイマイントラプトイネーブルレジスタ 【TIER】 ※ STEP26 で解説



各レジスタの前についている番号は、ハードウェアマニュアル (ADJ-602-208A) の章番号です。ITU の機能詳細は、マニュアルの「10. 16 ビットインテグレートドタイマユニット (ITU)」の項を参照してください。

22.3 タイマスタートレジスタ【TSTR】

TSTR (タイマスタートレジスタ) は、カウントの開始と停止を設定するレジスタで8ビットありますが、ビット7～5はリザーブビットです。

ビット:	7	6	5	4	3	2	1	0
	—	—	—	STR4	STR3	STR2	STR1	STR0
初期値:	1	1	1	0	0	0	0	0
R/W:	—	—	—	R/W	R/W	R/W	R/W	R/W

図 22-1 TSTR

【bit 4～0 STRn】カウンタスタート4～0

TCNT4～0の動作を設定します。TCNT0はSTR0、TCNT1はSTR1のビットになります。ビットに1を書き込むと、対応するTCNTがカウントを開始し、0を書き込むと停止します。

0	TCNTn のカウント動作停止
1	TCNTn のカウント動作開始

22.4 タイマコントロールレジスタ【TCR 0～4】

TCR (タイマコントロールレジスタ) は、クロック設定を行うレジスタで8ビットあり、ビット7はリザーブビットです。レジスタは各チャンネルごとにあります。

ビット:	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
初期値:	1	0	0	0	0	0	0	0
R/W:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

図 22-2 TCR

【bit 6, 5 CCLR1, 0】カウンタクリア

カウンタのクリア条件を設定します。

ここでは、**コンペアマッチ**によってクリアするか、しないかを設定することができます。フリーランニングカウンタ動作では、タイマカウンタのコンペアマッチによるクリアは禁止にしておき、クリアはオーバーフローした時、タイマカウンタに0を代入することで行います(オーバーフロー、コンペアマッチについてはSTEP26で説明します)

ビット6	ビット5	説明
CCLR1	CCLR0	
0	0	TCNT のクリア禁止
	1	GRA のコンペアマッチ/インプットキャプチャ ^{*1} で TCNT をクリア
1	0	GRB のコンペアマッチ/インプットキャプチャ ^{*1} で TCNT をクリア
	1	同期クリア。 同期動作 ^{*2} をしている他のタイマのカウンタクリアに同期して TCNT をクリア

【bit 4, 3 CKEG1, 0】クロックエッジ

外部クロックを使用するときのエッジの検出方法を設定します。

今回は、内部クロックを使用するのでクロックエッジは関係なく、初期値の0のままで構いません。

ビット4	ビット3	説明
CKEG1	CKEG0	
0	0	立上りエッジでカウント
	1	立下りエッジでカウント
1	—	立上り/立下りの両エッジでカウント

【bit 2 ~ 1 TPSC2 ~ 1】タイマプリスケアラ

カウントクロックを設定します。

TPSC2 に 1 を書き込むと外部クロックを使用するようになり、0 を書き込むと内部クロックを使用するよう設定します。本キットでは内部クロックを使用するので TPSC2 は 0 にします。

内部クロックを使用する場合、TPSC1、TPSC0 に書き込む値によってクロック周波数が分周されます。φは、内部クロックの最大値を表しており、本キットの場合は 25MHz です。

ビット2	ビット1	ビット0	説明
TPSC2	TPSC1	TPSC0	
0	0	0	内部クロック：φでカウント
		1	内部クロック：φ/2でカウント
	1	0	内部クロック：φ/4でカウント
		1	内部クロック：φ/8でカウント
1	0	0	外部クロック A：TCLKA 端子入力でカウント
		1	外部クロック B：TCLKB 端子入力でカウント
	1	0	外部クロック C：TCLKC 端子入力でカウント
		1	外部クロック D：TCLKD 端子入力でカウント

22.5 タイマカウンタ【TCNT 0～4】

TCNT（タイマカウンタ）は、カウントした値が入るレジスタで、16ビットあります。

チャンネル0だとTCNT0、チャンネル1だとTCNT1のレジスタになります。本テキストではチャンネル0を使用するので、TCNT0のレジスタを使います。

このTCNTは、アップカウンタであれば、初期値が0でカウントすることに1ずつ増えていきます。16ビットカウンタなので最大65535までカウントできるということになります。

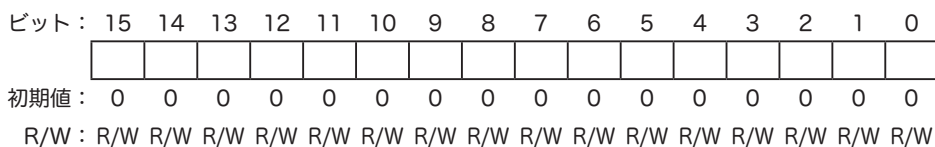


図 22-3 TCNT

チャンネル	レジスタ名	機能
0	TCNT0	アップカウンタ
1	TCNT1	
2	TCNT2	位相計数モード：アップ/ダウンカウンタ 上記以外：アップカウンタ
3	TCNT3	相補PWMモード：アップ/ダウンカウンタ 上記以外：アップカウンタ
4	TCNT4	

22.6 タイマステータスレジスタ【TSR 0～4】

TSR (タイマステータスレジスタ) は、TCNT がオーバーフローしたかどうかを確認するレジスタで8ビットあり、ビット7～3 はリザーブビットです。レジスタは各チャンネルごとにあります。

ビット:	7	6	5	4	3	2	1	0
	—	—	—	—	—	OVF	IMFB	IMFA
初期値:	1	1	1	1	1	0	0	0
R/W:	—	—	—	—	—	R/(W)*	R/(W)*	R/(W)*

※フラグをクリアするための0ライトのみ可能です。

図 22-4 TSR

【bit 2 OVF】オーバーフローフラグ

TCNT がオーバーフロー、またはアンダフローしたときに 1 が書き込まれます。初期値は 0 です。つまり、OVF をチェックすればタイマカウンタがオーバーフロー (アンダフロー) したか確認できるということです。

0	[クリア条件] (初期値) OVF = 1 の状態で、OVF フラグをリードした後、OVF フラグに 0 をライトしたとき
1	[セット条件] TCNT の値がオーバーフロー (H'FFFF → H'0000) またはアンダフロー (H'0000 → H'FFFF) したとき*

【bit 1, 0 IMFB, IMFA】インプットキャプチャ/コンペアマッチフラグ

IMFB、IMFA はジェネラルレジスタ B(GRB)、ジェネラルレジスタ A(GRA) のコンペアマッチを知らせるフラグです。詳しくは STEP26 で説明します。

オーバーフロー と アンダフロー

オーバーフローとは、コンピュータが数値計算などを行った際に決められた桁を超えてしまうことを言います。ここでは、タイマカウンタがアップカウント (カウント数を増やしていくカウント方法) を開始し、カウント数がカウントできる最大値 (本キットでは 65535 0xFFFF) より大きくなってしまったことを言います。

アンダフローとは、オーバーフローの逆でコンピュータが数値計算などで決められた値より小さくなってしまったことを言います。ここでは、タイマカウンタがダウンカウント (カウント数を減らしていくカウント方法) を開始し、カウント数がカウントできる最小値 (本キットでは 0) より小さくなってしまったことを言います。

※ 本テキストではダウンカウントは行いません。