



---

---

## megaAVR®からAVR® Dxマイクロ コントローラ系への移植

---

---

### 序説

著者: Cristian Pop, Microchip Technology Inc.

この文書は違いと類似性の両方を網羅して、AVR® Dx MCU系へ移植する、megaAVR®系に習熟した応用設計者を手助けします。比較はmegaAVR対AVR Dx製品の殆どに適用できますが、この文書での中心は64ピン外圍器で利用可能な128Kバイトのフラッシュ メモリのMCUの2つの世代であるATmega128とAVR128DA64です。

AVR Dxの周辺機能の殆どは(WDT、RTC、AC、ADC、SPI、USART、TWI、それと計時器を含み)megaAVRの周辺機能と機能的に互換ですが、移植時にはソースコードに対する更新が必要とされます。以降の項はいくつかの更新での詳細を含みますが、移植されるコードは目的対象応用の意図する動きが同じことを保証するため、完全に検査されなければなりません。megaAVRとAVR Dxの系統はピン互換ではありません。

AVR Dx系に対してピンの名前が同じですが、それらの位置はmegaAVR系から変えられています。より多くの詳細については各デバイスのデータシートで「ピン構成設定」と「ピン配置」の各々の部分をご覧ください。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

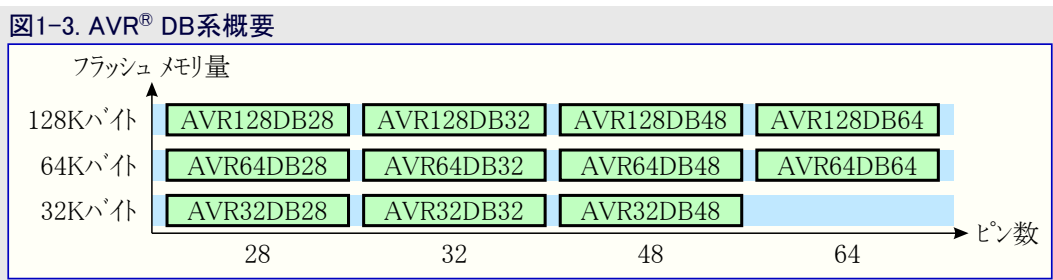
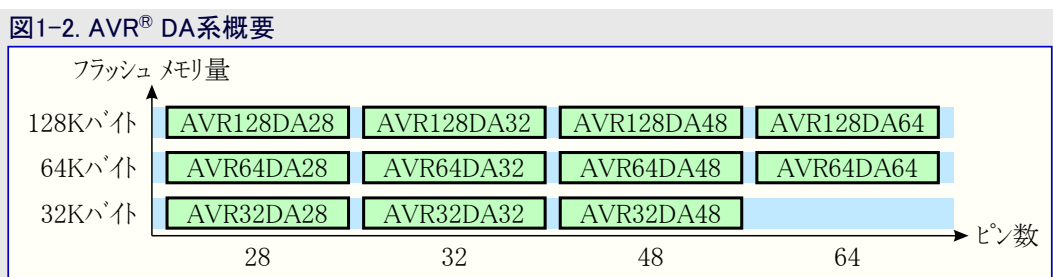
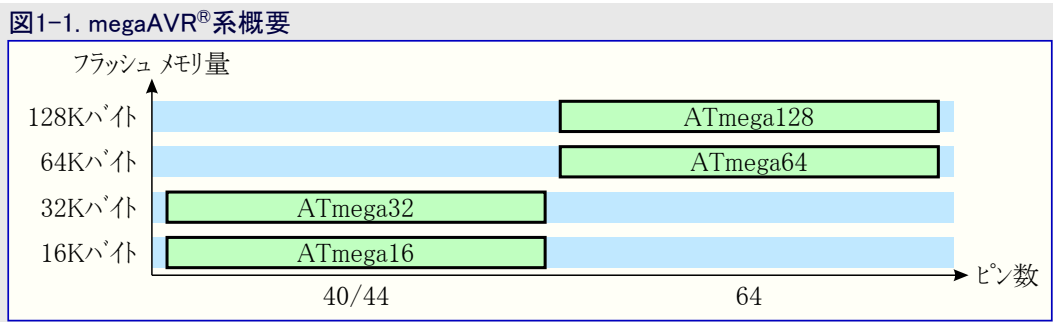
## 目次

序説	1
1. 関連デバイス	3
2. 共通周辺機能	4
2.1. 共通周辺機能	4
2.2. システム	4
2.3. メモリ	6
2.4. 入出力ポートとピン配列	7
2.5. 計時器	9
2.6. RTC - 実時間計数器	14
2.7. SPI	15
2.8. USART	16
2.9. TWI - 2線直列インターフェース	17
2.10. AC - アナログ比較器	20
2.11. ADC - A/D変換器	21
2.12. WDT - ウォッチドッグ タイマ	23
3. AVR <sup>®</sup> Dx - 追加周辺機能	25
3.1. 概要	25
3.2. DAC - D/A変換器	25
3.3. CCL - 構成設定可能な注文論理回路	26
3.4. EVSYS - 事象システム	26
3.5. CRCSCAN - 冗長巡回検査メモリ走査	26
3.6. ZCD - 0交差検出器	26
3.7. PTC - 周辺機能接触制御器	26
3.8. MVIO - 複数電圧入出力	27
3.9. OPAMP - アナログ信号調整	27
4. 参考文献	28
5. 改訂履歴	28
Microchipウェブサイト	29
製品変更通知サービス	29
お客様支援	29
Microchipデバイスコード保護機能	29
法的通知	29
商標	30
品質管理システム	30
世界的な販売とサービス	31

## 1. 関連デバイス

本章はこの文書に関連するデバイスを一覧にします。下図はピン数の変種とメモリ量を展開して各種系列デバイスを示します。

- これらのデバイスがピン互換で同じまたはより多くの機能を提供するため、垂直上方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。
- 異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMも持ちます。



(訳補) megaAVR系は機能が微妙に異なる周辺機能があるため、垂直方向の移植は変更が必要な場合もあります。

## 2. 共通周辺機能

### 2.1. 共通周辺機能

次表は両系統に存在する周辺機能の一覧を示します。

周辺機能/機能	ATmega16	ATmega32	ATmega64	ATmega128	AVR <sup>®</sup> DA系	AVR <sup>®</sup> DB系
最大周波数	16MHz (VDD ≥ 4.5V)				24MHz	
供給電圧 (VDD)	1.8~5.5V					
外圍器 (ピン数)	40、44		64		28、32、48、64	
フラッシュメモリ (読み-変更-書き領域)	16Kバイト (14Kバイト)	32Kバイト (28Kバイト)	64Kバイト (56Kバイト)	128Kバイト (120Kバイト)	32~128Kバイト (-)	
SRAM	1Kバイト	2Kバイト	4Kバイト		4~16Kバイト	
EEPROM	512バイト	1Kバイト	2Kバイト	4Kバイト	512バイト	
汎用入出力 (GPIO)	32		53		21~54	
USART	1		2		3~5	
SPI			1		2	
TWI (I <sup>2</sup> C)			1		1~2	
8ビットタイマ/カウンタ			2		- (注)	
12ビットタイマ/カウンタ			-		1	
16ビットタイマ/カウンタ			1		4~7 (注)	
実時間計数器 (RTC)			1			
アナログ比較器 (AC)			1		3	
A/D変換器 (ADC)			1			
ウォッチドッグタイマ (WDT)			1			

注: AVR Dxについては各TCAが2つの分離された8ビット計時器として動き、それらの各々がPWM生成用に3つの比較チャネルを持ちます。

## 2.2. システム

### 2.2.1. CPU

megaAVRとAVR Dxの系統は良く認識されているAVR CPUに基づきますが、AVR Dxは旧CPUの更新版から利益を得ます。megaAVR命令一式はAVR Dxコアによって完全に支援され、実行時間が改善されます。違いはC/C++コンパイラによって処理されますが、時間が重要または周期が正確なコードを移植する使用者は、実行時間が受入れ可能な範囲なことを確認するためにAVR<sup>®</sup>命令一式手引書を参照しなければなりません。

注: 命令一式と実行時間についての詳細に関してはAVR<sup>®</sup>命令一式手引書を参照してください。AVR Dxデバイスに関する詳細についてはAVRxtと呼ばれるCPU版を、megaAVRデバイスについてはAVRと呼ばれる版を参照してください。

### 2.2.2. 割り込み

megaAVRとAVR Dxの両系統は構成設定可能な割り込みベクタ表で包括的な割り込みシステムを提供しますが、基本構造と機能は系統間で異なります。割り込みのベクタとフラグは例えそれらが同じ周辺機能によって使われるとしても両系統で同じではありません。

megaAVRデバイスについて、リセットベクタはアドレス\$0000またはブートリセットアドレスのどちらかに置くことができ、ブートリセット(BOOTRST)ヒューズと割り込みベクタ選択(IVSEL)ビットを使って選択可能です。それはソフトウェアに於いてそれら間で交換する柔軟性で、ブートローダと応用に対して違う割り込みベクタ位置の使用を許します。

AVR Dxデバイスの割り込みベクタ表は例えブートローダが使われても、常にリセット後、アドレス\$0000に置かれます。それはブートローダ領域がmegaAVRデバイスのようにフラッシュメモリ領域の最後ではなく、先頭に置かれるために可能です。割り込みベクタ表は制御A(CPUINT.CTRLA)レジスタのIVSELビットを使ってソフトウェアで応用領域の先頭に再割り当て可能です。

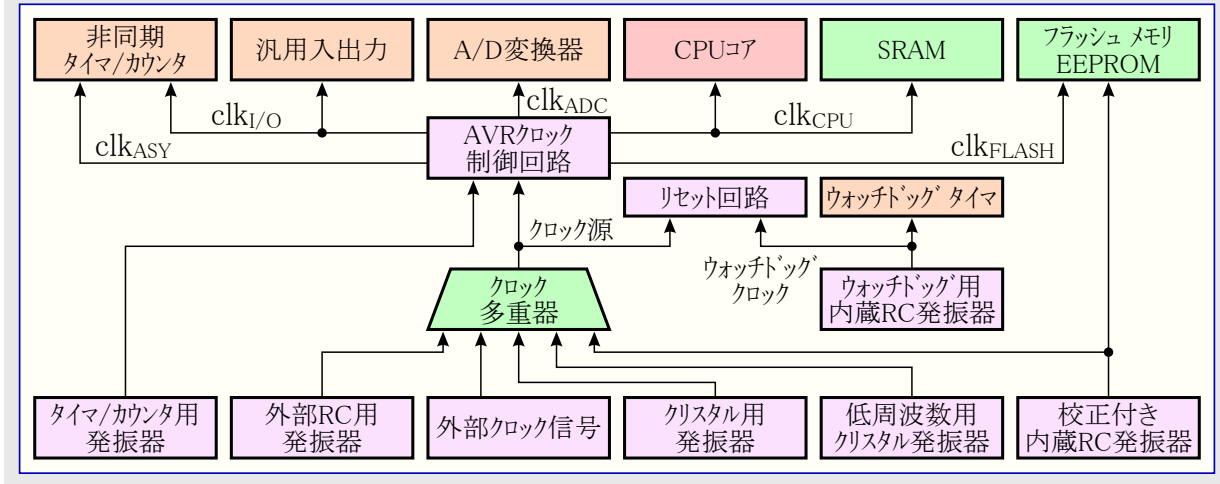
割り込みベクタ表は例え制限された割り込み数だけが使われても、常に生成されたコードで配置されます。簡潔なコード書きを許すため、AVR Dxデバイスの割り込み制御器は選択可能な簡潔ベクタ表(CVT: Compact Vector Table)を提供します。CVT機能が使われと、全ての割り込みが同じ割り込みベクタ番号/割り込み処理ルーチン(ISR: Interrupt Service Routine)を共用します。

CVT機能に加え、AVR Dxの割り込み制御器は重要機能用の遮蔽不可割り込み、1つの選択可能な高優先権割り込み、標準優先権割り込み用の任意選択ラウンドロビン計画機構を提供します。

2.2.3. クロック元

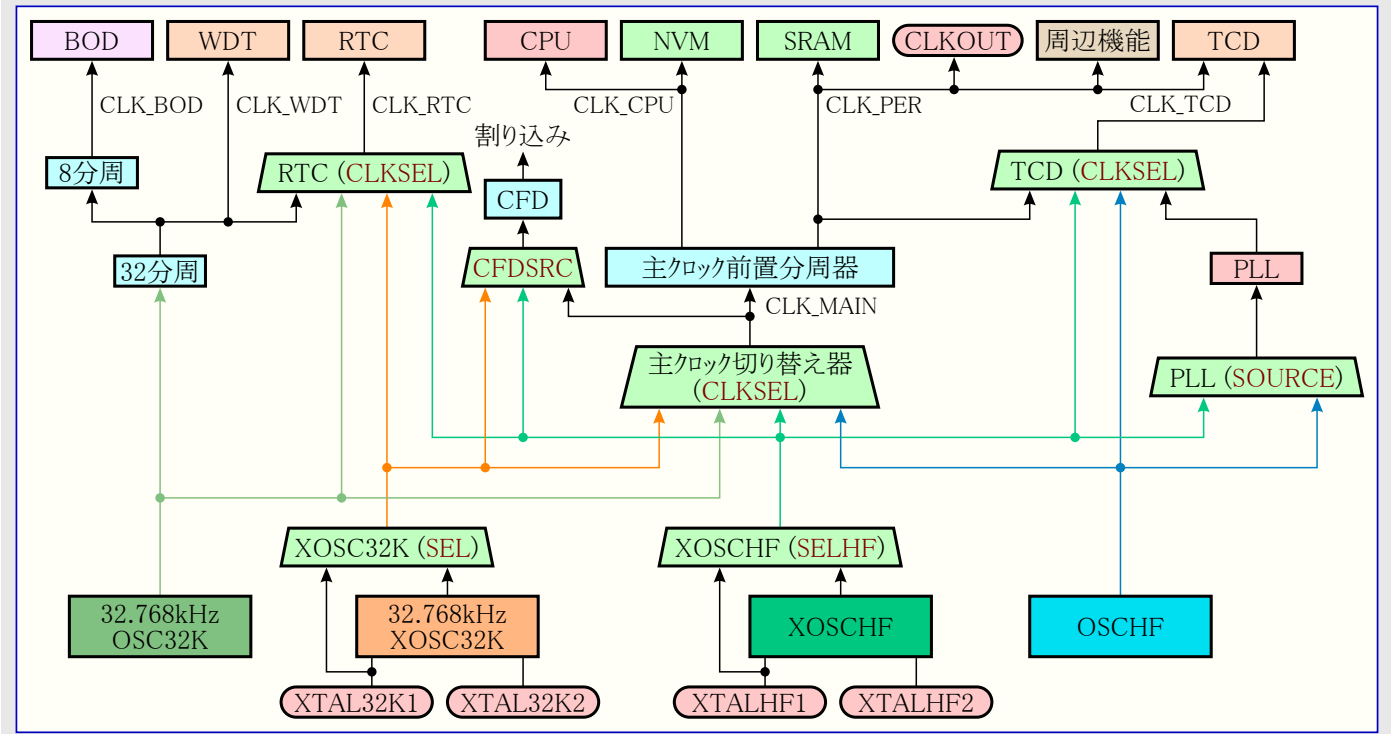
megaAVRマイクロ コントローラは外部RC発振器、クリスタル用発振器、外部クロック、校正付き内部RC発振器から選択可能な複数のシステムクロック用クロック任意選択を許します。クロック元はCKOPTとCKSEL3~0のヒューズによって選択可能で、ソフトウェアによって変更することはできません。

図2-1. ATmega128 クロック - 構成図



AVR Dx系はmegaAVR系と比べて異なるクロックシステムを使います。例えヒューズを使って既定発振器が選ばれていても、これは標準動作中にソフトウェアによって安全に変更することができます。内部高周波数発振器(OSCHF)の周波数もソフトウェアによって1,2,3,4MHzまたは4の倍数で24MHzまで選択可能です。より多くの詳細についてはデバイスのデータシートで「CLKCTRL - クロック制御器」章をご覧ください。

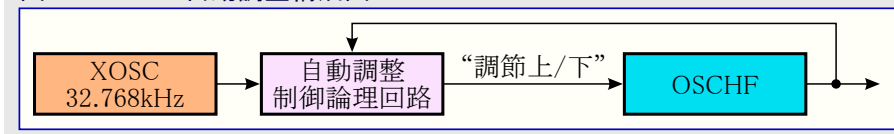
図2-2. AVR Dx クロック - 構成図



加えて、AVR Dxデバイスは2倍または3倍でのクロック通倍(最大周波数48MHz)を許す位相固定化閉路(PLL:Phase-Locked Loop)を提供し、タイマ/カウンタD型(TCD)用の入力として扱うことができます。

全てのAVR Dxデバイスが高周波数クリスタル用発振器(XOSCHF)用任意選択を提供するとは限りませんが、32.768kHzクリスタル用発振器と組み合わせた組み込み自動調整論理回路を使ってAVR Dxの内部高周波数発振器(OSCHF)の精度を改善することができます。これは32.768kHzクリスタルから派生した参照基準で内部1MHzを比較することによって行われます。

図2-3. OSCHF自動調整構成図



注: 自動調整機能の使い方についての詳細に関しては「TB3234 - 自動調整機能を使う高周波数発振器校正」を参照してください。

## 2.3. メモリ

AVR基本構造はデータメモリとプログラムメモリの2つの主なメモリ空間を持ちます。CPUデータメモリ空間はより速いアクセスを許し、アセンブリ言語でLD/ST系命令を使ってアクセス可能な一方で、プログラムメモリ空間(コード空間)はLPM/SPM系命令を通してアクセス可能です。

megaAVRデバイスについて、データメモリ空間の最初の4352バイトはレジスタファイル、I/Oメモリ、拡張I/Oメモリ、内部データSRAM用に予約されています。最初の32位置はレジスタファイルにアクセスするのに使われ、次の64位置は標準I/Oメモリにアクセスし、その後拡張I/Oメモリの160位置、そして次の4096位置は内部データSRAMにアクセスします。

AVR Dxデバイスについて、周辺機能、SRAM、EEPROM、I/Oレジスタは全てデータメモリ空間に配置される一方で、フラッシュメモリはプログラム空間に配置されます。フラッシュメモリは32Kバイトの塊でデータメモリ空間に割り当てられることもできます。

megaAVRデバイスと異なり、AVR Dxデバイスは外部メモリインターフェース用支援を持ちません。

### 2.3.1. 実装書き込み可能なフラッシュメモリとブートローダ支援

両AVR系はソフトウェアを使ってプログラムメモリの更新を許します。チップ上で実装再書き込み可能なフラッシュメモリは主にプログラム記憶に使われます。ソフトウェアの安全性のため、フラッシュプログラムメモリ空間は異なるアクセス権を持つ部分に分けられ、megaAVRデバイスは2つの領域(ブート領域と応用プログラム領域)を持つ一方で、AVR Dxデバイスは構成設定可能な3つ部分(ブートコード領域、応用コード領域、応用データ領域)を提供します。

megaAVRデバイスについて、ブート領域の大きさはBOTSZ1~0ヒューズを使って次の値、512、1024、2048、4096語からのどれかに構成設定することができます。この領域はフラッシュメモリの最後に置かれます。

AVR Dxデバイスについて、ブート領域の大きさはブートの大きさ(BOOTSZ)ヒューズを使って512バイトの粒度でどの値にも構成設定することができ、これはフラッシュメモリ領域の始めに置かれます。残りのフラッシュメモリはCDDSIZEヒューズを使って応用プログラム領域と応用データ領域に分けられます。安全性の理由のため、ブート領域からのコード実行は応用領域またはデータ領域を書くことができ、応用領域からのコード実行はデータ領域だけを書くことができ、データ領域からのコード実行はどのフラッシュメモリ領域も書くことができません。この機構は領域の意図せぬ変更を防ぎます。このハードウェア機構に加え、デバイスは他の領域からの望まれないブートコード読み込みを防ぐことができるソフトウェア任意選択を実装します。

### 2.3.2. SRAMとEEPROMのメモリ

両システムについて、SRAMメモリはデータ空間に配置されます。SRAMメモリのアドレス変位(オフセット)と大きさは(同じシステムでさえ)デバイス毎に異なり得ます。利用可能な大きさとアドレス変位についての詳細に関してはデバイスのデータシートを参照してください。

megaAVRデバイスについて、EEPROMは独立したデータ空間に置かれ、I/O空間を通してアクセス可能です。以下のコード例をご覧ください。

#### 例2-1. megaAVR® - EEPROMバイト書き込み

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    /* 直前の書き込み完了待ち */
    while (EECR & (1<<EWE))
    ;
    /* アドレスとデータのレジスタを設定 */
    EEAR = uiAddress;
    EEDR = ucData;
    /* EEMWEに論理1書き込み */
    EECR |= (1<<EEMWE);
    /* EWE設定(1)によってEEPROM書き込み開始 */
    EECR |= (1<<EWE);
}
```

AVR Dxデバイスに対してEEPROMはデータ空間に配置されます。それは領域全体の直線的なアクセスとLD/ST系命令を使ってより高速なアクセスを許します。

#### 例2-2. AVR® Dx - EEPROMバイト読み書き

```
uint8_t FLASH_0_read_eeprom_byte(eeprom_adr_t eeprom_adr)
{
    /* 読み込み操作はどれかの書き込み進行中の場合、ハードウェアによって引き延ばされます。 */
    return *(uint8_t *) (EEPROM_START + eeprom_adr);
}

nvmctrl_status_t FLASH_0_write_eeprom_byte(eeprom_adr_t eeprom_adr, uint8_t data)
{
    /* 直前の書き込み完了待ち */
    while (NVMCTRL.STATUS & (NVMCTRL.EEBUSY_bm | NVMCTRL.FBUSY_bm))
    ;
}
```

```

/* 望む値でEEPROM書き込み指定 */
ccp_write_spm((void *)&NVMCTRL.CTRLA, NVMCTRL_CMD_EEERWR_gc);

/* EEPROMにバイト書き込み */
*(uint8_t *) (EEPROM_START + eeprom_adr) = data;

/* 現在の指令を解消 */
ccp_write_spm((void *)&NVMCTRL.CTRLA, NVMCTRL_CMD_NONE_gc);

return NVM_OK;
}

```

## 2.4. 入出力ポートとピン配列

### 2.4.1. 共通機能

#### 2.4.1.1. GPIO基本機能

megaAVRとAVR Dxの系統はピン互換ではありません。けれども、GPIO基本機能は同じで両系統について3つのレジスタを使って構成設定可能です。

megaAVR <sup>®</sup>	AVR <sup>®</sup> Dx	説明
DDRx	PORTx.DIR	データ方向 - データ方向(出力駆動部)を制御します。
PORTx	PORTx.OUT	データ出力 - 各PORTxピン用出力駆動部水準を制御します。
PINx	PORTx.IN	データ入力 - PORTxピンの状態を示します。

以下のコード断片は各系統に対してPORTAの7番ピンをHighに駆動された出力として構成設定する方法を示します。

#### 例2-3. megaAVR<sup>®</sup> - 出力でHigh駆動として構成設定されたポートAの7番ピン(PA7)

```

{
  DDRA = 0x80;
  PORTA = 0x80;
}

```

#### 例2-4. AVR<sup>®</sup> Dx - 出力でHigh駆動として構成設定されたポートAの7番ピン(PA7)

```

{
  PORTA.DIR = 0x80;
  PORTA.OUT = 0x80;
}

```

#### 2.4.1.2. ピン上書き

AVRマイクロコントローラは基本操作(I/Oアクセス)と周辺機能に対してI/O PORTレジスタを使います。megaAVRマイクロコントローラは2つの機能間を切り替えるのに代替機能レジスタを使います。入出力ピンが周辺機能によって使われる時に代替ポート機能レジスタを正しく構成設定するのは使用者の責任です。

AVR Dxデバイスについて、周辺機能構成設定は周辺機能が許可されて入出力ピンを使う時にGPIOピンの基本機能を上書きします。それらが使われる時に強制的に入出力ピンを必要としないいくつかの周辺機能(即ち、計時器、CCL、SPI)があり、これらの周辺機能の機能は他の周辺機能(即ち、事象システム、割り込み、CCL)に対する入力として、またはソフトウェアアクセスに対して内部的に利用可能です。この場合、その周辺機能構成設定レジスタ内に追加出力許可任意選択が提供されます。

#### 2.4.1.3. プルアップ任意選択

megaAVRマイクロコントローラについて、プルアップは既定で通電に於いて許可されます。ソフトウェアは特殊I/O機能レジスタ(SFIOR)のプルアップ禁止(PUD)ビットを使って同時に全ポートを許可または禁止することができます。

megaAVRマイクロコントローラとは異なり、AVR Dx系は各ピンに対して個別のプルアップ任意選択を提供します。プルアップはリセットで禁止され、ピン制御(PORTx.PINnCTRL)レジスタのプルアップ許可(PULLUPEN)ビットを使ってソフトウェアによって許可することができます。

## 2.4.2. AVR<sup>®</sup> Dx追加機能

### 2.4.2.1. 直接ピン構成設定

GPIO基本機能は拡張I/Oレジスタ空間に属する3つのレジスタを使って制御されます。この空間はビット操作命令を許さず、1つのピンに対する構成変更は読み-変更-書き命令を使って行われます。ハードウェア読み-変更-書き機能は駆動値や入力と感知の構成設定の安全で正しい変更を保証しますが、これは3つのアセンブリ言語命令に変換されます。以下のコード断片は出力としてのPA7ピン構成設定を示します。

#### 例2-5. C言語コード例

```
PORTA.DIR |= 0x80;
```

#### 例2-6. アセンブリ言語コード

```
; ZポインタにPORTA.DIRアドレスを取得
LDI R30, 0x00;
LDI R31, 0x04;
LDI R24, Z    ; PORTA.DIRレジスタの内容を読み込み
ORI R24, 0x80 ; 0x80で論理和
ST Z, R24    ; PORTA.DIRレジスタに結果格納
```

アクセスを高速化してより多くの柔軟性を提供するため、新しいAVR Dxマイクロコントローラは1命令で複数ピンに対するGPIO構成設定の更新を許す新しい一部のレジスタを提供します。

レジスタ (注)	説明
PORTx.DIRSET	このビット領域のどのビットへの'1'書き込みもPORTx.DIRの対応するビットを設定(1)し、対応するピンを出力ピンとして構成設定して出力駆動部を許可します。
PORTx.DIRCLR	このビット領域のどのビットへの'1'書き込みもPORTx.DIRの対応するビットを解除(0)し、対応するピンを入力専用ピンとして構成設定して出力駆動部を禁止します。
PORTx.DIRTGL	このビット領域のどのビットへの'1'書き込みもPORTx.DIRの対応するビットを反転します。
PORTx.OUTSET	このビット領域のどのビットへの'1'書き込みもPORTx.OUTの対応するビットを設定(1)し、対応するピンをHigh駆動されるように構成設定します。
PORTx.OUTCLR	このビット領域のどのビットへの'1'書き込みもPORTx.OUTの対応するビットを解除(0)し、対応するピンをLow駆動されるように構成設定します。
PORTx.OUTTGL	このビット領域のどのビットへの'1'書き込みもPORTx.OUTの対応するビットを反転します。

注: レジスタ内のどのビット位置への'0'書き込みも無効です。

直接ピン構成設定機能を使うと、出力とするPA7ピンの構成設定用コードは次のようになります。

#### 例2-7. C言語コード

```
PORTA.DIRSET = 0x80;
```

#### 例2-8. アセンブリ言語コード

```
; ZポインタにPORTA.DIRSETアドレスを取得
LDI R30, 0x01;
LDI R31, 0x04;
LDI R24, 0x80 ; 適用ビット値取得
ST Z, R24    ; PORTA.DIRSETレジスタに書き込み
```

### 2.4.2.2. 仮想ポート

頻繁に使われるPORTレジスタもビットアクセス可能なI/Oメモリ空間に割り当てられます。仮想PORTレジスタへのアクセスは通常のレジスタへのアクセスと同じ結果を持ちますが、ビット操作命令のようなメモリ特有命令を許します。これらの命令は通常のPORTレジスタが属す拡張I/Oレジスタ空間で使うことはできません。

通常ポートレジスタ	仮想ポートレジスタ	説明
PORTx.DIR	VPORTx.DIR	データ方向 - データ方向(出力駆動部)を制御します。
PORTx.OUT	VPORTx.OUT	データ出力 - 各PORTxピン用出力駆動部水準を制御します。
PORTx.IN	VPORTx.IN	データ入力 - PORTxピンの状態を示します。
PORTx.INTFLAGS	VPORTx.INTFLAGS	ピン割り込み要求フラグ - これはPORTxピンの変更または状態がPORTx.PINCTRLでピンの入力/感知構成設定(ISC)と一致する時に設定(1)されます。



### 2.4.2.3. 複数ピン構成設定

1操作で複数のPORTピンを構成設定するのに複数ピン構成設定機能が使われます。望むピン構成設定が最初に複数ピン構成設定(PORTx.PINCONFIG)レジスタに書かれ、変更するために選んだピンでのレジスタ書き込みが後続します。これは1回の書き込みで8つまでのピンに対する(PORTx.PINnCTRL)構成設定変更を許します。以下のコード断片はPORTAに対するピン構成設定を示します。

#### 通常の方法による設定

```

; ZポインタにPORTA.DIRSETアドレスを取得
/* ピン構成設定 - 反転入出力許可、プルアップ許可、上昇端で割り込み感知 */
PORTA.PIN7CTRL = PORT_INVEN_bm | PORT_PULLUPEN_bm | PORT_ISC1_bm;
PORTA.PIN5CTRL = PORT_INVEN_bm | PORT_PULLUPEN_bm | PORT_ISC1_bm;
PORTA.PIN2CTRL = PORT_INVEN_bm | PORT_PULLUPEN_bm | PORT_ISC1_bm;
PORTA.PIN0CTRL = PORT_INVEN_bm | PORT_PULLUPEN_bm | PORT_ISC1_bm;

```

前のコードは複数ピン構成設定機能を使って次のように置き換えることができます。

#### 複数ピン構成設定機能を使う方法による設定

```

; ZポインタにPORTA.DIRSETアドレスを取得
/* ピン構成設定 - 反転入出力許可、プルアップ許可、上昇端で割り込み感知 */
PORTA.PINCONFIG = PORT_INVEN_bm | PORT_PULLUPEN_bm | PORT_ISC1_bm;
/* PA7,PA5,PA2,PA0ピンに対してPINxCTRLを更新 */
PORTA.PINCTRLUPD = PIN7_bm | PIN5_bm | PIN2_bm | PIN0_bm;

```

### 2.4.2.4. PORTMUX

AVR Dxデバイスは周辺機能入出力用の複数の任意選択を提供します。これらの任意選択はPORTMUXレジスタを使って使用者選択可能で、実装設計中の柔軟性を提供します(使用者はPCB配線に対して最良の任意選択を選ぶことができます)。利用可能な入出力についてのより多くの詳細に関しては指定AVR Dxデータシートを参照してください。

## 2.5. 計時器

AVR Dx系で利用可能な計時器はATmega128に存在する計時器の改良版です。加えて、AVR Dxデバイスは増された計時器数とそれらの出力に対する複数の任意選択を提供し、回路図設計を簡単化します。AVR Dxデバイスには以下の型の計時器が存在します。

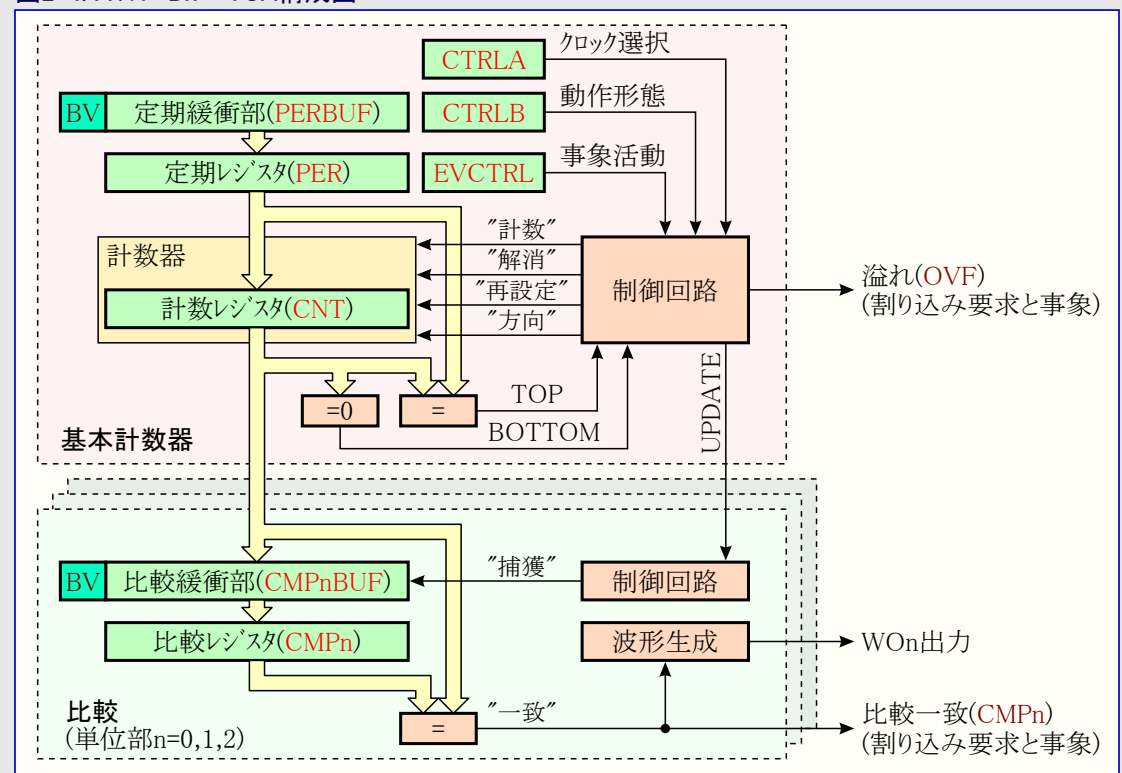
- ・ タイマ/カウンタA型 (TCA)
- ・ タイマ/カウンタB型 (TCB)
- ・ タイマ/カウンタD型 (TCD)

計時器はコード互換ではありませんが(違うレジスタ式)、ATmega128に存在する全ての機能がAVR Dxの計時器を使って置き換え可能です。従って、以降の項はAVR Dxの計時器を使って各種機能を得る方法を記述します。

### 2.5.1. TCA - タイマ/カウンタA型

タイマ/カウンタA型(TCA)はmegaAVRデバイスのタイマ/カウンタ0の改良版です。TCAは基本計数器と比較チャネルの組から成ります。比較チャネルは比較一致制御、周波数生成、パルス幅波形変調を行うために基本計数器と共に使うことができます。

8ビット計時器であるmegaAVRデバイスのタイマ/カウンタ0と異なり、TCAは3つ比較チャネルを持つ16ビットタイマ/カウンタです。任意選択の前置分周と共に周辺機能クロックまたは事象システムからクロック駆動、計時することができます。事象システムは方向制御や同期動作に使うこともできます。

図2-4. AVR<sup>®</sup> Dx - TCA構成図

加えて、TCAは16ビット計時器を各々がPWM生成用に3つの比較チャネルを持つ2つの独立した8ビット計時器に分割する分割動作機能を持ちます。分割動作は単一傾斜下降計数でだけ動きます。分割動作で事象制御された動作は支援されません。

以下の副項はTCAに対して可能ないくつかの使用事例を示します。

#### 2.5.1.1. 標準動作(計数器動作)

この動作ではTCAがクロック元として周辺機能クロックまたは事象を持つ上昇/下降計数器として使われます。使われる動作形態に応じて、内部計数器は各計時器クロックまたは事象で解消、増加、減少されます。計数器(TCAn.CNT)レジスタ値は何時でもCPUによって読むことができます。TCAn.CNTへの書き込みアクセスは計数、解消、再設定よりも高い優先権を持ち、即時に行われます。計数器の方向はTCAn.CTRLBの計数方向(DIR)に書くことによって標準動作中に変更することもできます。

標準動作ではピンでのTCA溢れ出力が存在しませんが、TCAは事象の計数数が比較n(TCAn.CMPn)値に達した時に割り込みを生成するのに使うことができます。以下のコードは周期的な割り込みを生成するためのTCAの構成設定を示します(割り込みコードは含まれません)。

#### 例2-9. AVR<sup>®</sup> Dx - TCA初期化、周期的割り込み許可

```
void TCA0_init(void)
{
    /* 溢れ割り込み許可 */
    TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;

    /* 標準動作設定 */
    TCA0.SINGLE.CTRLB = TCA_SINGLE_WGMODE_NORMAL_gc;

    /* 事象計数禁止 */
    TCA0.SINGLE.EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);

    /* 周期設定 */
    TCA0.SINGLE.PER = PERIOD_EXAMPLE_VALUE;

    TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV256_gc      /* クロック元(sys_clk/256)設定 */
                      | TCA_SINGLE_ENABLE_bm;         /* 計数器開始 */
}
```

2.5.1.2. 波形生成動作

対応するポートピンでの波形生成に比較チャネルを使うことができます。各比較nチャネルは計数器値(TCAn.CNT)を比較n(TCAn.CMPn)レジスタと継続的に比較します。TCAn.CNTがTCAn.CMPnと等しい場合に比較器n信号が一致します。この信号はその後に制御B(TCAn.CTRLB)レジスタの波形生成動作(WGMODE)ビット領域選択によって機能する波形生成に使われます。

注: WGMODEビット領域が全てのチャネルに対して共通なため、全ての比較チャネルに対して1つの単一波形動作が利用可能です。

例2-10. AVR® Dx - TCA初期化、単一傾斜PWM動作

```
void TCA0_init(void)
{
    /* ポートAで波形出力設定 */
    PORTMUX.TCAROUTEA = PORTMUX.TCA0_PORTA_gc;

    TCA0.SINGLE.CTRLB = TCA_SINGLE_CMP0EN_bm /* 比較0チャネル許可 */
                      | TCA_SINGLE_WGMODE_SINGLESLOPE_gc; /* 単一傾斜PWM動作設定 */

    /* 事象計数禁止 */
    TCA0.SINGLE.EVCTRL &= ~(TCA_SINGLE_CNTEI_bm);

    /* PWMの周波数とデューティサイクル(50%)を設定 */
    TCA0.SINGLE.PERBUF = PERIOD_EXAMPLE_VALUE;
    TCA0.SINGLE.CMPOBUF = DUTY_CYCLE_EXAMPLE_VALUE;

    TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV4_gc /* クロック元(sys_clk/4)設定 */
                      | TCA_SINGLE_ENABLE_bm; /* 計時器開始 */
}
```

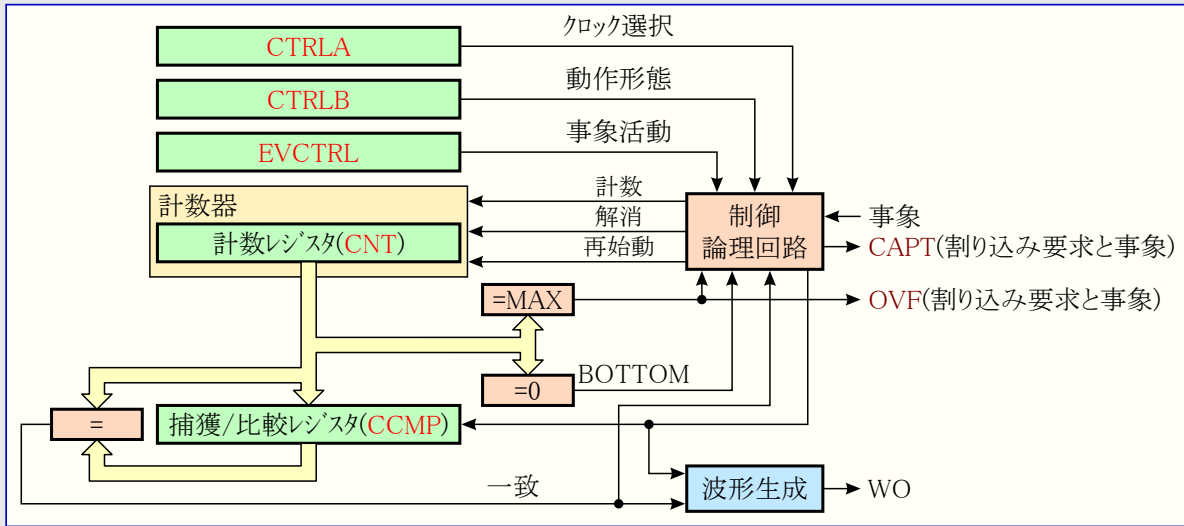
注: より多くの詳細とコード例については「TB3217 - TCAでの開始に際して」を参照してください。

2.5.2. TCB - タイマ/カウンタB型

AVR Dxデバイスのタイマ/カウンタB型(TCB)はmegaAVRデバイスのタイマ/カウンタ1の全ての機能を置き換えることができます。

TCBの能力は周波数と波形の生成、時間と共に事象での捕獲入力、デジタル信号の周波数測定を含みます。TCBは基本計数器と各動作形態が独特の機能を提供する8つの異なる動作形態の1つに設定することができる制御論理回路から成ります。

図2-5. AVR® Dx - TCB構成図

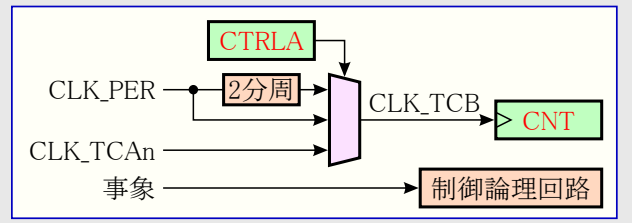


TCBは制御A(TCBn.CTRLA)レジスタのクロック選択(CLKSEL)ビット領域を通して選択可能な周辺機能クロック(CLK\_PER)、16ビットタイマ/カウンタA型(CLK\_TCAn)、事象システム(EVSYS)からクロック駆動することができます。

ピンで利用可能な波形出力(WO)が必要とされる場合、制御B(TCBn.CTRLB)レジスタの比較/捕獲出力許可(CCMPEN)が'1'を書かれなければなりません。WOの位置もTCBnピン位置(PORTMUX.TCBROUTEA)レジスタを使って選択可能ですが、WOピンの方向は出力として構成設定する必要があります(TCBは出力ピンの機能を上書きしますが、ピン方向を上書きしません)。

以降のコード断片はいくつかの使用事例に対するTCB初期化例を示します。

図2-6. AVR® Dx - TCBクロック論理回路



### 2.5.2.1. 事象での計数捕獲動作のTCB

この動作では計数器がBOTTOMからMAXまで継続的に計数します。事象が検出されると、計数器(TCBn.CNT)値が比較/捕獲(TCBn.CCMP)レジスタに転送され、捕獲(CAPT)の割り込みと事象が生成されます。この動作は連続する事象間の間隔を測定することなどに使うことができます。

計数器が継続的に動くため、それを合図するのに割り込み要求フラグ(TCBn.INTFLAGS)レジスタの溢れ(OVF)ビットが使われます。溢れが連続する事象間で現れる時を考慮に入れるのは使用者の責任です。

#### 例2-11. AVR<sup>®</sup> Dx - 事象での計数捕獲動作TCB初期化

```
void TCB2_Init(void)
{
    TCB2.CTRLB = 0 << TCB_ASYNC_bp           /* 非同期許可: 禁止 */
                | 0 << TCB_CCMPEN_bp         /* 比較/捕獲出力許可: 禁止 */
                | 0 << TCB_CCMPINIT_bp       /* 比較/捕獲ピン初期値: 禁止(Low) */
                | TCB_CNTMODE_CAPT_gc;       /* 事象での計数捕獲動作 */

    TCB2.EVCTRL = 1 << TCB_CAPTEI_bp         /* 捕獲事象入力許可: 許可 */
                | 0 << TCB_EDGE_bp          /* 事象端: 正端 */
                | 0 << TCB_FILTER_bp;       /* 捕獲入力雑音除去濾波器許可: 禁止 */

    TCB2.INTCTRL = 1 << TCB_CAPT_bp          /* 捕獲割り込み許可: 許可 */
                | 0 << TCB_OVF_bp;         /* 溢れ割り込み許可: 禁止 */

    TCB2.CTRLA = TCB_CLKSEL_DIV1_gc         /* CLK_PER */
                | 1 << TCB_ENABLE_bp       /* 許可: 許可 */
                | 0 << TCB_RUNSTDBY_bp     /* スタンバイ時走行: 禁止 */
                | 0 << TCB_SYNCUPD_bp      /* 同期更新: 禁止 */
                | 0 << TCB_CASCADE_bp;     /* 2つのタイマ/カウンタ連結: 禁止 */
}
```

### 2.5.2.2. 計数捕獲周波数測定動作のTCB

計数捕獲周波数測定動作ではTCBが事象入力信号の正端または負端のどちらかで計数器値を捕獲して再始動します(有効端は事象制御(TCBn.EVCTRL)レジスタの事象端(EDGE)ビットを使って選択可能です)。この動作を使うと、入力信号の周期は各信号周期に於いてソフトウェアに対して利用可能で、信号の周期/周波数を監視するソフトウェアを助けます。

#### 例2-12. AVR<sup>®</sup> Dx - 計数捕獲周波数測定動作TCB初期化

```
void TCB2_Init(void)
{
    TCB2.CTRLB = 0 << TCB_ASYNC_bp           /* 非同期許可: 禁止 */
                | 0 << TCB_CCMPEN_bp         /* 比較/捕獲出力許可: 禁止 */
                | 0 << TCB_CCMPINIT_bp       /* 比較/捕獲ピン初期値: 禁止(Low) */
                | TCB_CNTMODE_FRQ_gc;       /* 計数捕獲周波数測定動作 */

    TCB2.EVCTRL = 1 << TCB_CAPTEI_bp         /* 捕獲事象入力許可: 許可 */
                | 0 << TCB_EDGE_bp          /* 事象端: 正端 */
                | 0 << TCB_FILTER_bp;       /* 捕獲入力雑音除去濾波器許可: 禁止 */

    TCB2.INTCTRL = 1 << TCB_CAPT_bp          /* 捕獲割り込み許可: 許可 */
                | 0 << TCB_OVF_bp;         /* 溢れ割り込み許可: 禁止 */

    TCB2.CTRLA = TCB_CLKSEL_DIV1_gc         /* CLK_PER */
                | 1 << TCB_ENABLE_bp       /* 許可: 許可 */
                | 0 << TCB_RUNSTDBY_bp     /* スタンバイ時走行: 禁止 */
                | 0 << TCB_SYNCUPD_bp      /* 同期更新: 禁止 */
                | 0 << TCB_CASCADE_bp;     /* 2つのタイマ/カウンタ連結: 禁止 */
}
```

### 2.5.2.3. 単発動作のTCB

単発動作は接続された事象チャンネルで上昇端または下降端が観測される毎に比較/捕獲(TCBn.CCMP)レジスタによって定義された持続時間を持つパルスを生成するのに使うことができます。

#### 例2-13. AVR<sup>®</sup> Dx - 単発動作でのTCB初期化

```
void TCB2_Init(void)
{
    TCB2.CCMP = 0x400; /* パルス幅: 1000 */

    TCB2.CTRLB = 0 << TCB_ASYNC_bp /* 非同期許可: 禁止 */
                | 0 << TCB_CCMPEN_bp /* 比較/捕獲出力許可: 禁止 */
                | 0 << TCB_CCMPINIT_bp /* 比較/捕獲ピン初期値: 禁止(Low) */
                | TCB_CNTMODE_SINGLE_gc; /* 単発動作 */

    TCB2.EVCTRL = 1 << TCB_CAPTEI_bp /* 捕獲事象入力許可: 許可 */
                 | 0 << TCB_EDGE_bp /* 事象端: 正端 */
                 | 0 << TCB_FILTER_bp; /* 捕獲入力雑音消去濾波器許可: 禁止 */

    TCB2.INTCTRL = 1 << TCB_CAPT_bp /* 捕獲割り込み許可: 許可 */
                  | 0 << TCB_OVF_bp; /* 溢れ割り込み許可: 禁止 */

    TCB2.CTRLA = TCB_CLKSEL_DIV1_gc /* CLK_PER */
                 | 1 << TCB_ENABLE_bp /* 許可: 許可 */
                 | 0 << TCB_RUNSTDBY_bp /* スタンバイ時走行: 禁止 */
                 | 0 << TCB_SYNCUPD_bp /* 同期更新: 禁止 */
                 | 0 << TCB_CASCADE_bp; /* 2つのタイマ/カウンタ連結: 禁止 */
}
```

前の全ての使用事例に対して、TCBは事象システムとの相互関係で使われなければなりません。以下のコード見本は事象システムの構成設定を示します(TCB2用事象入力としてPA6ピンが使われます)。

#### 例2-14. AVR<sup>®</sup> Dx - TCB用事象システム初期化

```
int8_t EVENT_SYSTEM_0_init()
{
    EVSYS.CHANNEL0 = EVSYS_CHANNEL0_PORTA_PIN6_gc; /* ポートAの6番ピン */
    EVSYS.USERTCB2CAPT = EVSYS_USER_CHANNEL0_gc; /* TCB2を事象チャンネル0に接続 */
}
```

提示された使用事例に加えて、TCBは周期的割り込み、制限時間検査、PWM生成、または別のTCB周辺機能と併せて32ビット計時器として使うこともできます。動作形態の完全な一覧についてはAVR Dxのデータシートを参照してください。

**注:** より多くの詳細とコード例については「TB3214 - TCBでの開始に際して」を参照してください。

### 2.5.3. TCD - タイマ/カウンタ型

タイマ/カウンタ型(TCD)は同期計数器、前置分周器、比較、捕獲、制御論理回路から成る高性能波形生成器です。TCDの計数器は周辺機能クロックと非同期のクロックで動くことができます。任意選択の沈黙時間を持つ2つの独立した出力を生成する比較論理回路を含みます。捕獲と決定論的障害制御のために事象システムに接続されます。タイマ/カウンタは比較一致と溢れで割り込みと事象を生成します。

AVR Dxの他の計時器と同様に、TCDはPWM波形を生成するのに使うことができます。例えば波形生成動作(1傾斜、2傾斜、4傾斜、2重勾配)がTCAやTCBと同様でも、TCDの比較論理回路はハードウェアによる障害処理、入力無効化、過負荷保護、高速緊急停止のような外部事象での条件付き波形生成を許します。沈黙時間支援と共にこれらの機能はを半ブリッジまたは全ブリッジの信号のハードウェア支援を必要とする応用に対してTCDを理想的な選択にします。

TCDの波形出力信号は障害制御(TCDn.FAULTCTRL)レジスタの比較x許可(CMPxEN)ビットによって制御され、内部的またはピンで利用可能です。波形出力がピンで必要とされる場合、ソフトウェアはそれらのピンを出力として構成設定しなければなりません。

以下のコードは沈黙時間付き補完駆動信号の生成用TCD初期化を示します。

#### 例2-15. AVR<sup>®</sup> Dx - 半ブリッジ動作でのTCD初期化

```
void TCD0_init(void)
{
    /* 波形動作設定 */
    TCD0.CTRLB = TCD_WGMODE_DS_gc;

    /* 信号周期設定 */
}
```

```

TCD0.CMPBCLR = SIGNAL_PERIOD_EXAMPLE_VALUE;

/* 信号は交互に活性で小さな対称沈黙時間が必要とされます。*/
TCD0.CMPBSET = SIGNAL_DUTY_CYCLE_EXAMPLE_VALUE + 1;
TCD0.CMPASET = SIGNAL_DUTY_CYCLE_EXAMPLE_VALUE - 1;

/* 許可準備可(ENRDY)ビットの設定(1)を保証 */
while (!(TCD0.STATUS & TCD_ENRDY_bm))
{
;
}

TCD0.CTRLA = TCD_CLKSEL_20MHZ_gc          /* 計時器のクロックを選択 */
| TCD_CNTPRES_DIV1_gc                    /* 前置分周器を選択 */
| TCD_ENABLE_bm;                         /* 計時器を許可 */

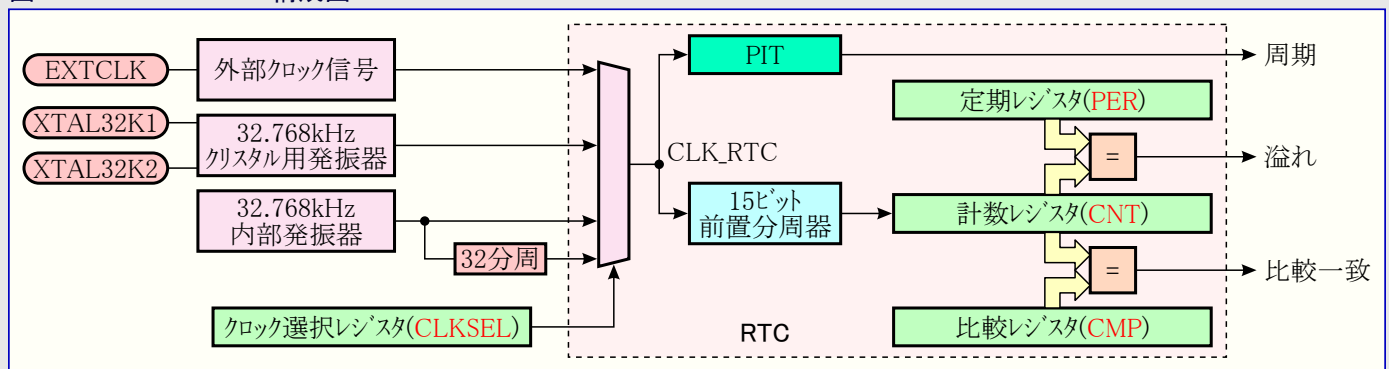
```

注: より多くの詳細とコード例については「TB3212 - TCDでの開始に際して」を参照してください。

## 2.6. RTC - 実時間計数器

実時間計数器(RTC)は一般的に時間の経緯を保つために低電力休止動作を含み、継続的に動く周辺機能です。実時間計数器(RTC:Real-Time Counter)と周期的割り込み計時器(PIT:Periodic Interrupt Timer)の2つのタイミング機能を提供します。デバイスを休止動作から起こしたり、規則的な間隔でデバイスに割り込むことができます。

図2-7. AVR<sup>®</sup> Dx - RTC構成図



RTC周辺機能はアイドルとスタンバイの休止動作で動きます。広範囲の分解能と計数可能期間を構成設定することができます。32.768 kHzのクロック元で最大分解能は30.5μsで、計時可能期間は2秒までにすることができます。1秒の分解能では最大計時可能期間が18時間(65536秒)を超えます。

以下のコード断片は1秒周期で割り込みを生成するためのRTCの初期化を示します(割り込みコードは含まれません)。

### 例2-16. AVR<sup>®</sup> Dx - 1秒周期的割り込み用RTC初期化

```

void RTC_0_init()
{
    while (RTC.STATUS > 0) {          /* 全レジスタ同期待ち */
    }
    RTC.PER = 0x3FF;                  /* 周期: 0x3FF - 1024 RTCクロック周期 */
    RTC.CLKSEL = RTC_CLKSEL_OSC1K_gc; /* 32kHzの32分周 */
    RTC.INTCTRL = 0 << RTC_CMP_bp;   /* 比較一致割り込み許可: 禁止 */
    | 1 << RTC_OVF_bp;               /* 溢れ割り込み許可: 許可 */
}

```

RTC機能に加えて、AVR DxデバイスのRTCはパワーダウン動作でも動くことができる周期的割り込み計時器(PIT)を含みます。許可されたなら、2<sup>n</sup>クロック周期毎に割り込み要求を生成することができ、nはPIT制御A(RTC.PITCTRLA)レジスタ経由で2~15のどの値も持つことができます。この動作は周期的な割り込み/デバイスがパワーダウン動作の時に休止からの起き上がりを許します。

注: より多くの詳細とコード例については「TB3213 - RTCでの開始に際して」を参照してください。

## 2.7. SPI

### 2.7.1. 共通機能

主装置/従装置動作に於けるAVR DxのSPI周辺機能の機能は標準動作が使われる時にATmega128と同じですが、レジスタ名とビット順が異なることがあります。右表はAVR Dx SPIレジスタ構造に対するmegaAVR SPI構成設定ビットの割り当てを示します。

系統間での別の違いはSPI周辺機能で使うための入出力ピン構成設定です。megaAVRは使われるピンの正しい構成設定が必要な一方で、AVR DxデバイスではピンがSPI周辺機能によって使われる時に入出力ピンの機能を上書きします。詳細については「2.4.1.2. ピン上書き」項を参照してください。

以下のコード断片は主装置動作でのSPI周辺機能の初期化を示します。

megaAVR®	AVR® Dx	説明
SPCR.DORD	SPIn.CTRLA.DORD	SPIデータ順
SPCR.MSTR	SPIn.CTRLA.MASTER	主装置/従装置選択
SPCR.CPOL	SPIn.CTRLB.MODE1,0	SPI動作形態選択
SPCR.CPHA	SPIn.CTRLB.MODE1,0	SPI動作形態選択
SPCR.SPR1	SPIn.CTRLA.PRESC1,0	SPIクロック前置分周器(分周数)
SPCR.SPR0	SPIn.CTRLA.PRESC1,0	SPIクロック前置分周器(分周数)
SPSR.SPI2X	SPIn.CTRLA.CLK2X	倍速クロック
SPCR.SPE	SPIn.CTRLA.ENABLE	SPI許可
SPCR.SPIE	SPIn.INTCTRL.IE	SPI割り込み許可
SPSR.SPIF	SPIn.INTFLAGS.IF	SPI割り込み要求フラグ
SPSR.WCOL	SPIn.INTFLAGS.WRCOL	書き込み衝突フラグ
SPDR	SPIn.DATA	SPIデータレジスタ

#### 例2-17. megaAVR® - 主装置動作でのSPI初期化

```
void SPI_MasterInit(void)
{
    /* MOSIとSCKを出力、他の全てを入力に設定 */
    DDR_SPI = (1<<DD_MOSI) | (1<<DD_SCK);
    /* SPI許可、主装置、クロック速度をfck/16に設定 */
    SPCR = (1<<SPE) | (1<<MSTR) | (1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
    /* 送信開始 */
    SPDR = cData;
    /* 送信完了待ち */
    while (!(SPSR & (1<<SPIF)))
        ;
}
```

#### 例2-18. AVR® Dx - 主装置動作でのSPI初期化

```
void SPI0_init(void)
{
    PORTA.DIR |= ~PIN4_bm; /* MOSIピンの方向を出力に設定 */
    PORTA.DIR &= PIN5_bm; /* MISOピンの方向を入力に設定 */
    PORTA.DIR |= ~PIN6_bm; /* SCKピンの方向を出力に設定 */
    PORTA.DIR |= ~PIN7_bm; /* SSピンの方向を出力に設定 */
    SPI0.CTRLA = SPI_DORD_bm /* 下位ビット先行送信 */
                | SPI_ENABLE_bm /* 単位部許可 */
                | SPI_MASTER_bm; /* SPI単位部主装置動作 */
}

uint8_t SPI0_exchangeData(uint8_t data)
{
    SPI0.DATA = data;
    while (!(SPI0.INTFLAGS & SPI_IF_bm)) /* データが交換されるまで待機 */
    {
        ;
    }
    return SPI0.DATA;
}
```

## 2.7.2. AVR® Dx追加機能

AVR Dx系のSPI周辺機能はデータ緩衝支援を持ち、従ってデータ処理に影響を与えます。緩衝動作は制御B(SPIIn.CTRLB)レジスタの緩衝動作許可(BUFEN)ビットに'1'を書くことによって許可されます。緩衝動作を支援するために割り込み制御(SPIIn.INTCTRL)と割り込み要求フラグ(SPIIn.INTFLAGS)のレジスタに追加ビットがあります。詳細については指定AVR Dxデータシートを参照してください。

**注:** より多くの詳細とコード例については「TB3215 - SPIでの開始に際して」を参照してください。

## 2.8. USART

### 2.8.1. 共通機能

USARTの基本操作は両系統に対して同様ですが、レジスタとビットの名称が違います。AVR Dxデバイスは構成設定/状態のビットに対するより効率的なアクセスを許す改良されたレジスタ構造を持ちます。

以降の項はAVR Dxレジスタ構造へのATmega128レジスタの割り当てを詳述します(megaAVRレジスタとAVR Dx対応だけが表されます)。

megaAVR®	AVR® Dx	説明
RXBn (UDRn読み込み)	USARTn.RXDATAL	受信データレジスタ下位バイト
TXBn (UDRn書き込み)	USARTn.TXDATAL	送信データレジスタ下位バイト
UCSRnB.RXB8n	USARTn.RXDATALH.DATA8	受信データビット8
UCSRnB.TXB8n	USARTn.TXDATALH.DATA8	送信データビット8
UCSRnB.UCSZn2	USARTn.CTRL.CHSIZE2~0 (注1)	USART文字ビット数
UCSRnC.UCSZn1,0		
UCSRnB.RXENn	USARTn.CTRL.RXEN	受信部許可
UCSRnB.TXENn	USARTn.CTRL.TXEN	送信部許可
UCSRnB.RXCIEEn	USARTn.CTRLA.RXCIE	受信完了割り込み許可
UCSRnB.TXCIEEn	USARTn.CTRLA.TXCIE	送信完了割り込み許可
UCSRnB.UDRIENn	USARTn.CTRLA.DREIE	USARTデータレジスタ空割り込み許可
UCSRnA.RXCn	USART.STATUS.RXCIF	USART受信完了フラグ
UCSRnA.TXCn	USART.STATUS.TXCIF	USART送信完了フラグ
UCSRnA.UDREN	USART.STATUS.DREIF	USARTデータレジスタ空フラグ
UCSRnA.FEn	USARTn.RXDATALH.FERR	フレーム異常
UCSRnA.DORn	USARTn.RXDATALH.BUFOVF	データオーバーラン(緩衝部溢れ)
UCSRnA.UPEEn	USARTn.RXDATALH.PERR	パリティ誤り
UCSRnA.U2Xn	USARTn.CTRLB.RXMODE1,0 = CLK2X	USART動作倍速
UCSRnA.MPCMn	USARTn.CTRLB.MPCM	複数プロセッサ通信動作
UCSRnC.UMSELn	USARTn.CTRL.CMODE1,0 (注2)	USART動作形態選択
UCSRnC.UPMn1,0	USARTn.CTRL.PMODE1,0	パリティ動作
UCSRnC.UBSn	USARTn.CTRL.SBMODE	停止ビット動作
UCSRnC.UCPOLn	USARTn.CTRL.UCPHA (注3)	クロック極性(同期動作)
UBRRn.UBRRn11~0	USARTn.BAUD.BAUD15~0 (注4)	ボーレート

**注1:** この設定は9ビット動作に対して違いがあります。

**注2:** ASYNCHとSYNCHの動作のみ

**注3:** 同期動作のみ

**注4:** ボーレート式が異なります。詳細についてはデバイスのデータシートを参照してください。

以下のコード断片は両系統に対するUSART周辺機能の初期化を示します。

#### 例2-19. megaAVR® - USART初期化 (9600ボー、パリティなし、1停止ビット)

```
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

void USART_Init(void)
{
    /* ボーレート設定 */
    UBRRL = BAUD_PRESCALE; /* UBRRレジスタ下位バイトに下位8ビット設定 */
    UBRRH = (BAUD_PRESCALE >> 8); /* UBRRレジスタ上位バイトに上位8ビット設定 */

    UCSRB = (1<<RXEN) | (1<<TXEN); /* 受信部と送信部を許可 */
    UCSRC = (1<<USBS) | (3<<UCSZ0); /* フレーム形式設定: 8データビット、1停止ビット */
}
```



}

例2-20. AVR<sup>®</sup> Dx - USART初期化 (9600ボー、パリティなし、1停止ビット)

```
#define USART1_BAUD_RATE(BAUD_RATE) ((float)(64 * 4000000 / (16 * (float)BAUD_RATE)) + 0.5)

void USART1_init(void)
{
    USART1.BAUD = (uint16_t)(USART1_BAUD_RATE(9600)); /* ボーレート設定 */

    USART1.CTRLA = USART_CHSIZE0_bm
                  | USART_CHSIZE1_bm; /* データ形式を8ビットに設定 */

    USART1.CTRLB |= USART_RXEN_bm | USART_TXEN_bm; /* 受信部と送信部を許可 */
}
```

2.8.2. AVR<sup>®</sup> Dx追加機能

AVR Dxデバイスは次のような追加機能を持つUSART周辺機能の改良版を含みます。

- 半二重動作 (単線動作とRS-485動作)
- 主装置SPI動作支援
- 自動ボーレート機能 (標準とLIN)
- IrDA<sup>®</sup>準拠通信用IRCOM単位部

注: より多くの詳細とコード例については「TB3216 - USARTでの開始に際して」を参照してください。

## 2.9. TWI - 2線直列インターフェース

TWI周辺機能はクロック(SCL)用の1本とデータ(SDA)用の1本で2本の双方向バス線だけを使って128個までの個別アドレス指定可能な装置の相互接続をシステム設計者に許します。主装置/従装置でのAVR DxのTWI周辺機能の機能は基本操作に対してATmega128と同様ですが、この周辺機能の構造、レジスタ名、ビット順が異なります。これらの違いのため、TWI周辺機能で相互通信するソフトウェア製品は統合段階中に更新されて検査されなければなりません。

以下のコード断片は主装置でのTWI周辺機能の使い方を示します。

例2-21. megaAVR<sup>®</sup> - 主装置でのTWI

```
/* 主装置初期化関数 */
void TWI_init_master(void)
{
    /* SCL周波数= F_CPU/(16+2(TWBR)×4^TWPS) */
    TWBR = 0x01; /* ビット速度 */
    TWSR = (0<<TWPS1) | (0<<TWPS0); /* 前置分周器ビット設定 */
}

void TWI_start(void)
{
    /* TWI割り込み要求フラグ解除(0)、SDAで開始条件、TWI許可 */
    TWCR = (1<<TWINT) | (1<<TWSTA) | (1<<TWEN);
    while (!(TWCR & (1<<TWINT))); /* 開始条件送信待ち */
    while ((TWSR & 0xF8) != 0x08); /* 応答調査 */
}

void TWI_write_address(unsigned char data)
{
    TWDR = data; /* アドレスと書き込みを指令 */
    TWCR=(1<<TWINT) | (1<<TWEN); /* TWI割り込み要求フラグ解除(0)、TWI許可 */
    while (!(TWCR & (1<<TWINT))); /* TWDRバイト送信完了まで待機 */
    while ((TWSR & 0xF8) != 0x18); /* 応答調査 */
}

void TWI_write_data(unsigned char data)
{
    TWDR = data; /* データをTWDRに配置 */
    TWCR=(1<<TWINT) | (1<<TWEN); /* TWI割り込み要求フラグ解除(0)、TWI許可 */
}
```

```

while (!(TCCR & (1<<TWINT)));          /* TWDRバイト送信完了まで待機 */
while ((TWSR & 0xF8) != 0x28);        /* 応答調査 */
}

unsigned char TWI_read_data(void)
{
    unsigned char recv_data;
    TCCR = (1<<TWINT) | (1<<TWEN);      /* TWI割り込み要求フラグ解除(0)、TWI許可 */
    while (!(TCCR & (1<<TWINT)));      /* TWDRバイト送信完了まで待機 */
    while ((TWSR & 0xF8) != 0x58);    /* 応答調査 */
    recv_data = TWDR;
    return recv_data;
}

void TWI_stop(void)
{
    /* TWI割り込み要求フラグ解除(0)、SDAで停止条件、TWI許可 */
    TCCR = (1<<TWINT) | (1<<TWEN) | (1<<TWSTO);
    while(!(TCCR & (1<<TWSTO)));      /* 停止条件送信待ち */
}

```

## 例2-22. AVR<sup>®</sup> Dx - 主装置でのTWI

```

void TWIO_init()
{
    TWIO.MBAUD = (uint8_t)TWIO_BAUD(100000, 0); /* ビット速度設定 */
    TWIO.MCTRLA = 1 << TWI_ENABLE_bp          /* TWI主装置許可: 許可 */
    | 0 << TWI_QCEN_bp                        /* 迅速指令許可: 禁止 */
    | 0 << TWI_RIEN_bp                        /* 読み込み割り込み許可: 禁止 */
    | 0 << TWI_SMEN_bp                        /* 簡便動作許可: 禁止 */
    | TWI_TIMEOUT_DISABLED_gc                /* バス制限時間禁止 */
    | 0 << TWI_WIEN_bp;                       /* 書き込み割り込み許可: 禁止 */
}

void TWIO_start(void)
{
    /* 開始条件はハードウェアによって生成されます。互換性のため維持 */
}

static uint8_t TWIO_WaitW(void)
{
    uint8_t state = 0;
    do
    {
        if(TWIO.MSTATUS & (TWI_WIF_bm | TWI_RIF_bm))
        {
            if(!(TWIO.MSTATUS & TWI_RXACK_bm))
            {
                /* 従装置が肯定で応答 - TWIはM1状態へ */
                state = I2C_ACKED;
            }
            else
            {
                /* アドレスを送出したが、肯定応答なし - TWIはM3状態へ */
                state = I2C_NACKED;
            }
        }
        else if(TWIO.MSTATUS & (TWI_BUSERR_bm | TWI_ARBLOST_bm))
        {
            /* バス異常または調停敗退の場合にだけここに到着 - M4状態 */
            state = I2C_ERROR;
        }
    } while(1);
}

```

```
    } while(!state);
    return state;
}

uint8_t TWIO_write_address(uint8_t address)
{
    uint8_t state = 0;
    TWIO.MADDR = address;          /* 従装置アドレス送信 */
    state = TWIO_WaitW();         /* 状態符号待ち */
    return state;
}

uint8_t TWIO_write_data(uint8_t data)
{
    uint8_t state = 0;
    TWIO.DATA = data;             /* データ送信 */
    state = TWIO_WaitW();         /* 状態符号待ち */
    return state;
}

static uint8_t TWIO_WaitR(void)
{
    uint8_t state = I2C_INIT;
    do
    {
        if(TWIO.MSTATUS & (TWI_WIF_bm | TWI_RIF_bm))
        {
            state = I2C_READY;
        }
        else if(TWIO.MSTATUS & (TWI_BUSERR_bm | TWI_ARBLOST_bm))
        {
            /* バス異常または調停敗退の場合にだけここに到着 - M4状態 */
            state = I2C_ERROR;
        }
    } while(!state);
    return state;
}

uint8_t TWIO_read_data(uint8_t* data, bool last_byte)
{
    uint8_t state;
    state = TWIO_WaitR();
    if(state == I2C_READY)
    {
        *data = TWIO.DATA;
        if (last_byte) TWIO.MCTRLB = TWI_ACKACT_bm | TWI_MCMD_STOP_gc;
        else TWIO.MCTRLB = TWI_MCMD_RECVTRANS_gc;
    }
    return state;
}

void TWIO_stop(void)
{
    TWIO.MCTRLB |= TWI_MCMD_STOP_gc;
}
```

2.9.1. AVR® Dx追加機能

AVR Dx系のTWI周辺機能は次のような新しい機能一式の支援を提供するTWI周辺機能の改良版が装備されます。

- SMBus
- 複数主装置
- 簡便動作
- 2元動作
- 10ビット アドレス
- 迅速指令動作

2.10. AC - アナログ比較器

megaAVRデバイスのアナログ比較器はAVR Dxデバイスと比べて異なるレジスタ式と限定された機能を持ちます。megaAVRのアナログ比較器の正入力(AIN0)と内部参照基準(1.23V代表固定)から選択可能な一方で、負入力(AIN1)は特殊I/O機能レジスタ(SFIOR)のアナログ比較器多重器許可(ACME)ビットを使って専用外部ピン(AIN1)とADC入力ピンから選ぶことができます。ACMEビットが設定(1)され、ADCがOFFに切り替えられると(A/D変換制御/状態レジスタ(ADCSRA)のADC許可(ADEN)ビットが'0')、ADC多重器(ADMUX)レジスタの多重器(MUX2~0)はアナログ比較器へ負入力を置き換えるための入力ピンを選びます。

AVR Dxデバイスは多重器制御(ACn.MUXCTRL)レジスタを使って正と負の入力に対してより多くの選択を提供します。各アナログ比較器は専用の多重器が装備され、従って利用可能なアナログ入力の制限されない使用を許します。アナログ比較器には専用の内部参照基準電圧(VREF)が提供されます。加えて、各比較器は精密な参照基準電圧選択用にチップ上8ビットDAC(DACREF)が装備されます。

図2-8. megaAVR®デバイス用アナログ比較器構成図

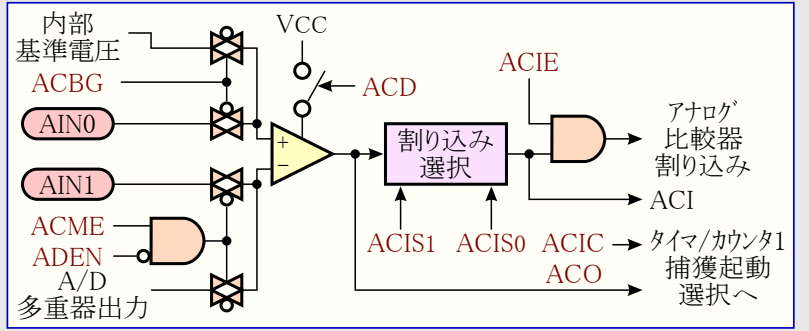
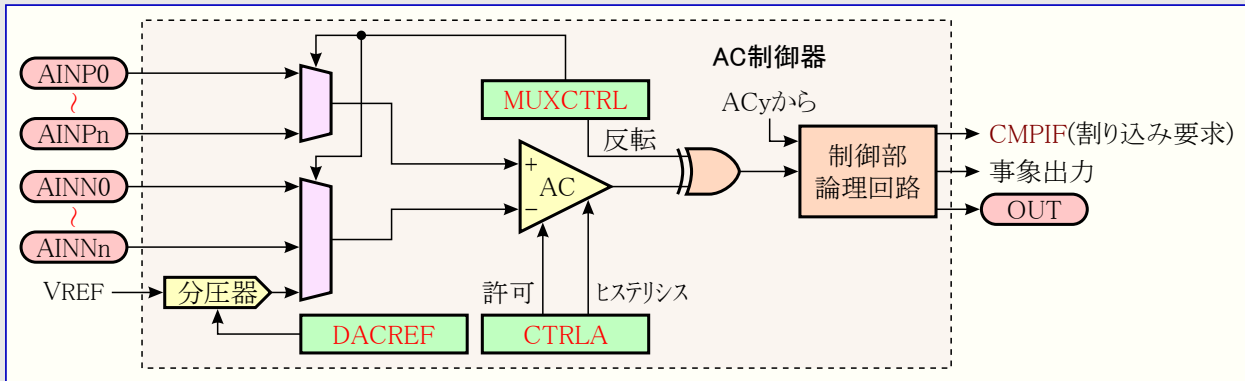


図2-9. AVR® Dxデバイス用アナログ比較器構成図



これらの違いにも関わらず、megaAVRのACに対する機能はAVR DxのAC周辺機能の正しい構成設定によって得ることができます。以下のコード例は両システムに対するAC周辺機能の初期化(固定参照基準電圧での入力比較)を示します。

例2-23. megaAVR® - ACの基本的な初期化

```
#define AINpin PA3
void ACInit(void)
{
    DDRA  &=~(1<<AINpin);          /* ピンを入力として設定 */
    PORTA &=~(1<<AINpin);          /* プルアップなし */
    SFIOR |= (1<<ACME);           /* 多重器許可 */
    ADCSRA &= ~(1<<ADEN);         /* ADCがOFFなことを保証*/
    ADMUX |= (0<<MUX2) | (1<<MUX1) | (1<<MUX0); /* 負AINとしてADC3を選択 */
    ACSR |= (0<<ACD) |           /* 比較器ON */
           (1<<ACBG) |           /* AIN0に1.23V参照基準を接続 */
           (0<<ACIE) |           /* 比較器割り込み禁止 */
           (0<<ACIC) |           /* 捕獲入力禁止 */
           (0<<ACIS1) |
           (0<<ACIS0);
}
```

例2-24. AVR<sup>®</sup> Dx - ACの基本的な初期化

```

/* VREF=1.5VのためDACREFを1.23Vに設定 */
#define DACREF_VALUE (1.23 * 256 / 1.5)

void AC0_init(void)
{
    /* 正入力 - デジタル入力緩衝部禁止 */
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /* 負入力は内部参照基準を使用 - 参照基準電圧が許可されなければなりません。 */
    VREF.CTRLA = VREF_ACOREFSEL_1V5_gc; /* 1.5V参照基準電圧 */
    VREF.CTRLB = VREF_ACOREFEN_bm; /* AC0 DACREF参照基準許可: 許可 */

    AC0.DACREF = DACREF_VALUE; /* DAC参照基準電圧設定 */
    /* 比較器に対して正しい入力を選択 */
    AC0.MUXCTRLA = AC_MUXPOS_PINO_gc /* 正入力 - アナログ正入力ピン0 */
                  | AC_MUXNEG_DACREF_gc; /* 負入力 - DAC参照基準電圧 */

    AC0.CTRLA = AC_ENABLE_bm /* アナログ比較器許可 */
               | AC_OUTEN_bm; /* 出力緩衝部許可: 許可 */
}

```

2.10.1. AVR<sup>®</sup> Dx追加機能

AVR Dxデバイスはそれらの各々2つを窓動作で構成設定可能な3つまでのアナログ比較器を提供します。窓動作では電圧窓を定義することができ、選ばれた比較器は入力信号がこの範囲内か否かのどちらかを示します。

デバイスは基準電圧(VREF)周辺機能経由で選択可能なアナログ比較器用参照基準電圧が装備されます(同じ値がデバイス内の全てのACに適用されます)。各比較器は参照基準電圧水準の微調整を許す追加8ビットDACが装備されます。

雑音耐性を改善するため、制御A(ACn.CTRLA)レジスタのヒステリシス動作選択(HYSMODE1,0)ビット領域を使ってヒステリシス機能が利用可能です。この機能は選択可能な水準を持ち、雑音に悩まされる入力信号がお互いに近い時に定常的な出力の交互切り替わりを防ぐのに役立ちます。

**注:** より多くの詳細とコード例については「TB3211 - ACでの開始に際して」を参照してください。

## 2.11. ADC - A/D変換器

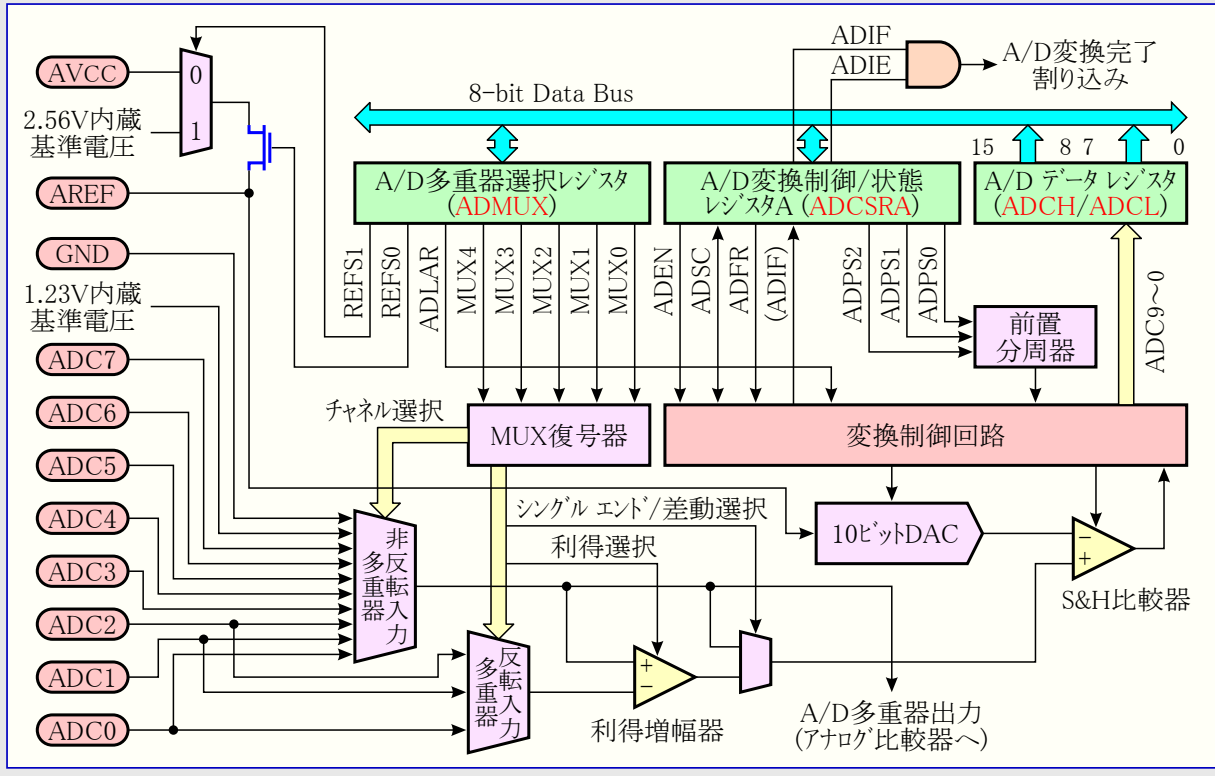
2.11.1. megaAVR<sup>®</sup>デバイス

megaAVRデバイスのA/D変換器(ADC)はAVR Dxデバイスに比べて異なるレジスタ一式と限定された機能を持ちます。これらの違いのため、移植されるソフトウェアは両系統で同じ機能を保証するために完全に検査されて確認されなければなりません。

megaAVRデバイスについて、ADCは10ビット分解能と15kspsまでの採取速度を持ちます。ADCはポートAのピンを使って構成された8つのシングルエンド電圧入力を許す8チャンネルのアナログ多重器に接続されます。このデバイスは、それらの2つが設定可能な利得段を備える16種の差動電圧入力の組み合わせも支援します。入力構成設定はADC多重器選択(ADMUX)レジスタのアナログチャンネルと利得選択のビットを使って選ばれます。

ADCはADMUXレジスタの基準電圧選択(REFS)ビット領域を使って選択可能な内部または外部の参照基準用の任意選択を持ちます。内部参照基準電圧はチップ上の2.56V固定電圧またはAVCCピンから選ぶことができ、使われる時に雑音分離(デカップ)用AREFピンで利用可能です。従って、外部参照基準電圧がAREFピンに印加している場合、内部参照基準電圧を使うことはできません。

図2-10. megaAVR® - ADC構成図

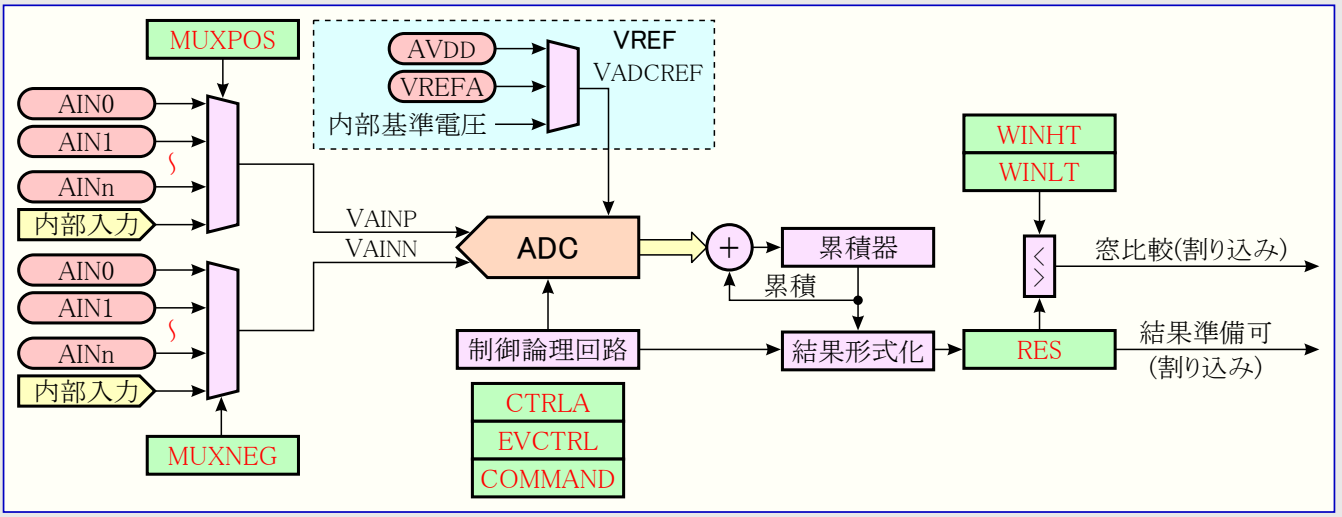


### 2.11.2. AVR® Dxデバイス

AVR DxデバイスのADCは12ビット分解能と130kspsまでの採取速度を持ちます。制御A(ADCn, CTRLA)レジスタの変換動作形態(COMM)ビットを使って選択可能なシングルエンド動作と差動動作で使うことができます。megaAVRデバイスと異なり、差動動作では利用可能な正と負の入力のどの組み合わせも許され、多重器選択(MUXPOSとMUXNEG)のレジスタを使って構成設定することができます。

ADC参照基準電圧も改善され、専用の参照基準電圧(VREFADC)が利用可能になっています。この参照基準は基準電圧(VREF)周辺機能のADC0参照基準(ADC0REF)レジスタを使って選択可能です。

図2-11. AVR® Dx - ADC構成図



以下のコードはシングルエンド動作、内部参照基準電圧でのADC初期化を示します。

#### 例2-25. megaAVR® - シングルエンド動作でのADC初期化

```
void ADC_Init(void)
{
    /* ADC前置分周器を128分周に設定 */
    ADCSRA |= (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);

    /* ADC参照基準を2.56V内部参照基準に設定 */
    ADMUX |= (1 << REFS0);
}
```

```

/* 容易な8ビット読み取りを許すためADC結果を左揃え */
ADMUX |= (1 << ADLAR);

// ADC0入力を使うためMUX値は変更される必要がありません。

ADCSRA |= (1 << ADFR);          /* ADCを自由走行(連続変換)動作に設定 */
ADCSRA |= (1 << ADEN);          /* ADC許可 */
ADCSRA |= (1 << ADSC);          /* A/D変換開始 */
}

```

#### 例2-26. AVR<sup>®</sup> Dx - シングル エント 動作でのADC初期化

```

void ADC0_init(void)
{
    /* デジタル入力緩衝部禁止 */
    PORTD.PINCTRL &= ~PORT_ISC_gc;
    PORTD.PINCTRL |= PORT_ISC_INPUT_DISABLE_gc;

    /* プルアップ抵抗禁止 */
    PORTD.PINCTRL &= ~PORT_PULLUPEN_bm;

    ADC0.CTRLB = ADC_PRESC_DIV4_gc          /* CLK_PERを4分周 */
                | ADC_REFSEL_INTREF_gc;    /* 内部参照基準 */
    ADC0.CTRLA = ADC_ENABLE_bm             /* ADC許可: 許可 */
                | ADC_RESSEL_12BIT_gc;    /* 12ビット動作 */
    ADC0.MUXPOS = ADC_MUXPOS_AIN0_gc;     /* ADCチャンネル0選択 */
    ADC0.CTRLA |= ADC_FREERUN_bm;         /* 自由走行動作許可 */
    ADC0.COMMAND = ADC_STCONV_bm;        /* 変換開始 */
}

```

### 2.11.3. AVR<sup>®</sup> Dx追加機能

megaAVRの機能に加え、AVR DxデバイスのADCは次のような他の機能を提供します。

- 変換毎に128採取までの累積
- チップ上の温度感知器チャンネル
- 設定可能な入力採取持続時間
- 構成設定可能な閾値と窓比較器
- 事象起動変換

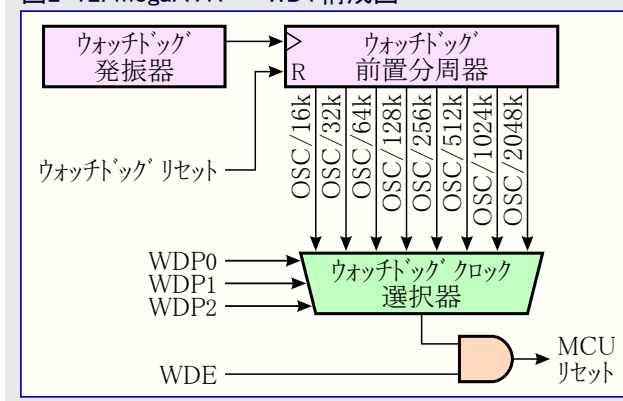
**注:** より多くの詳細とコード例については「TB3209 - ADCでの開始に際して」を参照してください。

### 2.12. WDT - ウォッチドッグ タイマ

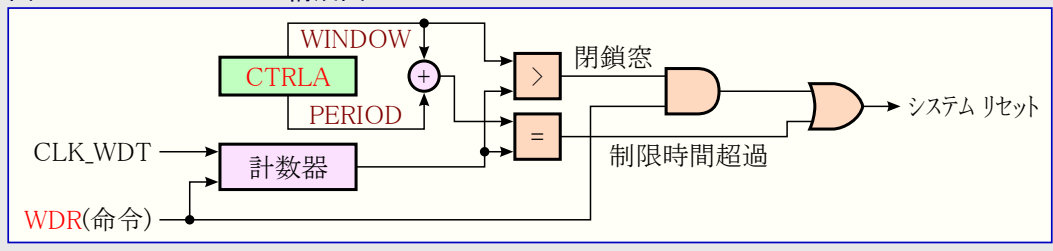
両系統はウォッチドッグ タイマ(WDT)周辺機能が装備されていますが、機能とレジスタ一式は異なります。従って、megaAVRのWDT操作を構成設定するソフトウェア手続きはAVR Dxデバイスを支援するように完全に置き換えられなければなりません。

megaAVRデバイスのウォッチドッグは1MHzで動くチップ上の発振器からクロック駆動されます。ウォッチドッグ リセット間隔はウォッチドッグ タイマ制御(WDTCR)レジスタのウォッチドッグ タイマ前置分周器(WDP2~0)ビットを使って16ms~2.2秒が選択可能です。

図2-12. megaAVR<sup>®</sup> - WDT構成図



AVR Dxデバイスについて、ウォッチドッグ単位部は改良され、追加の機能一式を持ちます。独立した発振器を使って周辺機能クロックから非同期に動きます。改善された消費電力のためにチップ上の超低電力発振器からクロック駆動され、16ms~8秒の11種の選択可能な制限時間間隔を提供します。

図2-13. AVR<sup>®</sup> Dx - WDT構成図

両システム用のWDTはWDT設定が偶然に変更できないことを保証する書き込み保護機構を持ちます。以下のコード例はウォッチドッグ禁止手順を示します。

#### 例2-27. megaAVR<sup>®</sup> - ウォッチドッグ禁止コード例

```
void WDT_off(void)
{
    /* WDTリセット */
    _WDR();
    /* WDTOEとWDEに論理1書き込み */
    WDTCR |= (1<<WDTOE) | (1<<WDE);
    /* WDTをOFFに切り替え */
    WDTCR = 0x00;
}
```

#### 例2-28. AVR<sup>®</sup> Dx - ウォッチドッグ禁止コード例

```
void WDT_0_off(void)
{
    /* WDTリセット */
    asm("WDR");
    /* WDT禁止 */
    ccp_write_io((void *)&(WDT.CTRLA),
                WDT_PERIOD_OFF_gc /* OFF */
                | WDT_WINDOW_OFF_gc /* OFF */);
}
```

### 2.12.1. AVR<sup>®</sup> Dx追加機能

AVR Dxデバイスはウォッチドッグ構成設定(WDTCFG)ヒューズを使ってウォッチドッグ機能を構成設定するための任意選択を持ちます。この任意選択はウォッチドッグリセット(WDR)命令の必要性を無くすという訳ではなく、ソフトウェア支援なしにリセット後直ちにWDTが動くように構成設定することができることを保証します。ソフトウェアはWDTによって起動されるシステムリセットを避けるため、周期的にWDR命令を走行しなければなりません。

**注:** WDTの構成設定はヒューズから許可された場合にソフトウェアによって変更することができません(状態(WDT.STATUS)レジスタの施錠(LOCK)ビットが自動的に設定(1)されます)。詳細についてはAVR Dxデバイスのデータシートを参照してください。

標準動作に加え、AVR Dxのウォッチドッグは窓動作を持ちます。窓動作はWDTがリセットされなければならない間の制限時間期間の内側に時間枠または窓を定義します。WDTが早すぎまたは遅すぎでこの窓の外側でリセットされる場合、システムリセットが発行されます。標準動作に比べ、窓動作はコード異常が定期的にWDR実行を引き起こす状況を捕らえることができます。



### 3. AVR<sup>®</sup> Dx – 追加周辺機能

#### 3.1. 概要

AVR Dx系はいくつかの応用で外部部品の必要性を置き換える追加周辺機能が装備されます。これらの構成部はより簡潔な設計を許し、費用と消費電力を低めます。

表3-1. AVR<sup>®</sup> Dx – 追加周辺機能

周辺機能/機能	AVR <sup>®</sup> DA系	AVR <sup>®</sup> DB系
D/A変換器 (DAC)	1 (10ビット)	
構成設定可能な注文論理回路 (CCL)	6 LUT	
事象システム (EVSYS)	1 (10チャンネル)	
巡回冗長検査 (CRCSCAN)	1	
0交差検出器 (ZCD)	3	
周辺機能接触制御器 (PTC)	1	-
複数電圧入出力 (MVIO)	-	1
アナログ信号調整 (OPAMP)	-	3

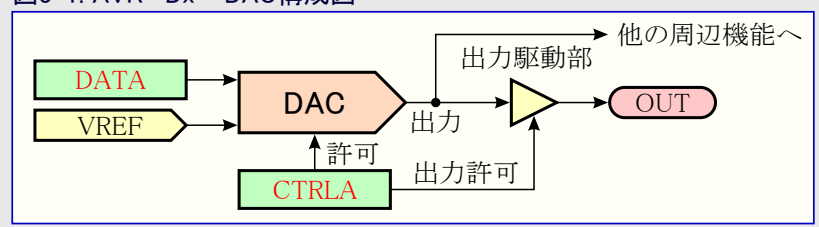
#### 3.2. DAC – D/A変換器

AVR Dxデバイスは140ksp/sまでの変換速度で動くことができる10ビットD/A変換器(DAC)が装備されます。出力範囲はGND～選んだ参照基準電圧間で、外部ピンで利用可能、または(ADCのような)他の内部周辺機能によって使うことができます。

DACのアナログ出力は制御A(DACn.CTRLA)レジスタの出力緩衝部許可(OUTEN)に'1'を書くことによってピンに接続することができます。DACによって使われるピンはPORT周辺機能で禁止された入力を持たなければなりません。DAC出力とピン間にはアナログ値がピン負荷に依存しないのを保証する出力緩衝部があります。出力緩衝部は電流を吐き出すことだけでき、非常に制限された吸い込み能力しか持ちません。

以下のコード断片はDACアナログ出力ピンで固定電圧を生成するためのDAC用初期化コードを示します。

図3-1. AVR<sup>®</sup> Dx – DAC構成図



#### 例3-1. AVR<sup>®</sup> Dx – 固定電圧を生成するためのDAC初期化

```

/* DAC値 */
#define DAC_EXAMPLE_VALUE    (0x258)
/* DACデータレジスタ用2LSbを得るのに必要とされる遮蔽 */
#define LSB_MASK              (0x03)

void DAC0_init(void)
{
    VREF.DACOREF = VREF_REFSEL_2V048_gc    /* DAC用に2.048V内部参照基準電圧を選択 */
                | VREF_ALWAYS_ON_bm;      /* 参照基準電圧を常時ON動作に設定 */
    /* デジタル入力緩衝部禁止 */
    PORTD.PIN6CTRL &= ~PORT_ISC_gc;
    PORTD.PIN6CTRL |= PORT_ISC_INPUT_DISABLE_gc;
    /* プルアップ抵抗禁止 */
    PORTD.PIN6CTRL &= ~PORT_PULLUPEN_bm;
    DAC0.CTRLA = DAC_ENABLE_bm            /* DAC許可 */
                | DAC_OUTEN_bm           /* 出力緩衝部許可 */
                | DAC_RUNSTDBY_bm;       /* スタンバイ動作で走行許可 */
    /* DAC0.DATALに2LSb格納 */
    DAC0.DATAL = (DAC_EXAMPLE_VALUE & LSB_MASK) << 6;
    /* DAC0.DATAHに8MSb格納 */
    DAC0.DATAH = DAC_EXAMPLE_VALUE >> 2;
}

```

**注:** DAC周辺機能を使うことについてのコード例とより多くの詳細に関しては「TB3235 – アナログ信号生成用10ビットDACの使い方」を参照してください。

### 3.3. CCL - 構成設定可能な注文論理回路

構成設定可能な注文論理回路(CCL:Configurable Custom Logic)はデバイスのピン、事象、他の内部周辺機能に接続することができる設定可能な論理回路周辺機能です。CCLはデバイスの周辺機能と外部装置間の”接続論理回路”として扱うことができます。CCLは外部論理回路部品の必要性をなくすことができ、CPUから独立した応用の最も重要な部分を扱うのにコアから独立した周辺機能(CIP:Core Independent Peripherals)を組み合わせることによって実時間制限を克服する設計者を手助けすることもできます。

**注:** CCL周辺機能を使うのとコード例についての詳細に関しては「PIC®とAVR®のマイクロコントローラでの注文論理回路」ウェブページを参照してください。

### 3.4. EVSYS - 事象システム

事象システム(EVSYS:Event System)はCPUの関与なしに周辺機能相互通信を許す配線網です。これは1つの周辺機能(事象生成部)での変化がCPUを使うことなく事象チャネルを通して別の周辺機能(事象使用部)での活動を起動するのを許します。事象は遅れなし、決して失われず、高速で予測可能な合図を提供し、実時間応用に対して理想的な決定論的な系を許します。これは自律的な周辺機能制御と相互作用、いくつかの周辺機能単位部での同期されたタイミングの活動を許します。従って、これはソフトウェアの複雑さ、大きさ、実行時間を減らすための強力な道具です。

**注:** EVSYS周辺機能を使うのとコード例についての詳細に関しては「事象システム(EVSYS)」ウェブページを参照してください。

### 3.5. CRCSCAN - 冗長巡回検査メモリ走査

AVR Dxデバイスは重要な安全機能である巡回冗長検査機構を提供します。コード化けが起きていないことを保証することにより、危険な状況を引き起こし得る応用に於ける潜在的な意図せぬ動きを避けることができます。

CRCSCAN周辺機能は不揮発性メモリ(NVM:Non-Volatile Memory)を走査し、コードが化けていないことを保証します。これは予め計算された物と比較されるチェックサムを生成します。2つのチェックサムが一致した場合、フラッシュメモリはOKで、応用コードは走行を開始することができます。

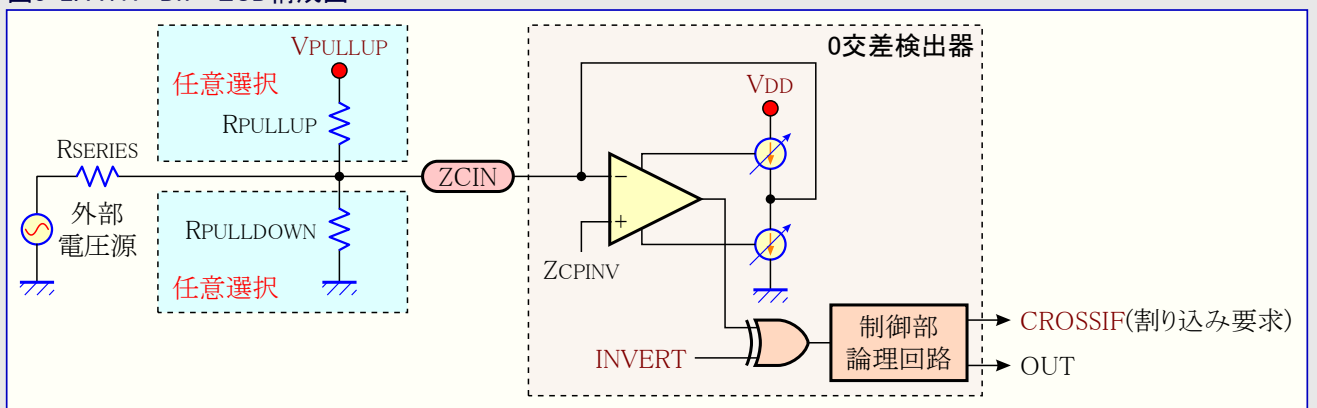
**注:** CRCSCAN周辺機能を使うのとコード例についての詳細に関しては「巡回冗長検査(CRC/SCAN)」ウェブページを参照してください。

### 3.6. ZCD - 0交差検出器

0交差検出器(ZCD:Zero-Cross Detector)は交流電圧が接地電位近くの閾値電圧を通過して横切る時を検出します。ZCDは以下のような目的に限定されず、交流波形を監視する時に使うことができます。

- ・ 周期測定
- ・ 正確な長期時間測定
- ・ 調光位相遅延駆動
- ・ 低EMI周回切り替え

図3-2. AVR® Dx - ZCD構成図



0交差を検出するため、直列の電流制限抵抗(RSERIES)だけを必要とし、従って設計を簡単化します。

**注:** ZCD周辺機能を使うことについてのコード例と詳細に関しては「TB3233 - 特殊機能を実装するためのZCDの使い方」を参照してください。

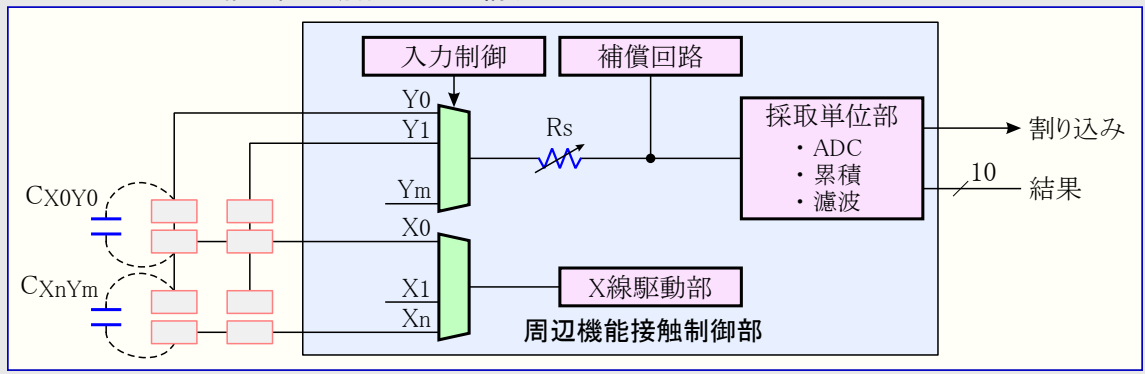
### 3.7. PTC - 周辺機能接触制御器

周辺機能接触制御器(PTC:Peripheral Touch Controller)は容量性感知器での接触を検出するための専用周辺機能です。外部容量性感知器は一般的にPCBまたは、酸化インジウム錫(ITO:Indium Tin Oxide)やPEDOTのような透明または半透明の材料を持つ透明基板で形成されます。PTCは外部部品を使うことなく、様々な感知器(釘、摺動子、輪、2D表面)に対して低電力、高感度での頑強な接触解決策の設計を許します。

感知部電極はデバイスの入出力ピンを通してPTCのアナログ前処理部へ直接的に接続されます。専用ハードウェアとQTouch®ソフトウェアライブラリは低いCPU使用と接触解決策のより速い開発を許します。

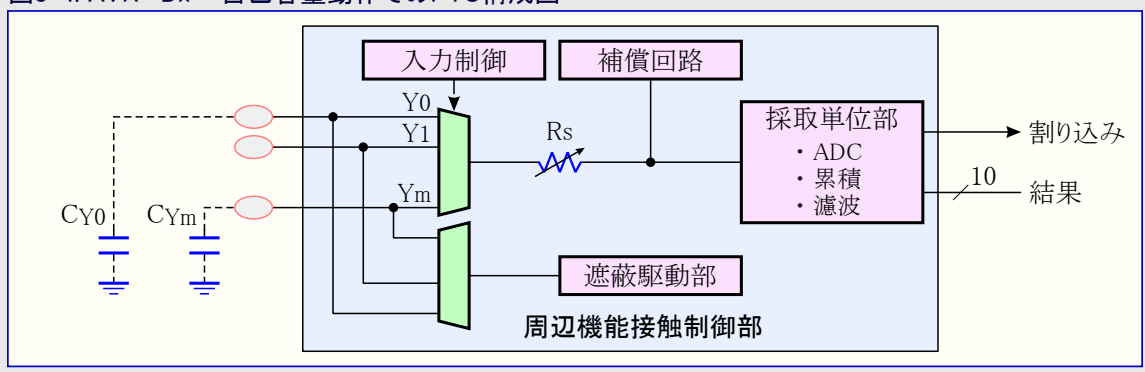
PTCは相互容量と自己容量の両感知器を支援します。相互容量動作では様々なX-Y構成の容量性接触配列を使って感知が行われます。PTCはX線毎に1つのピンとY線毎に1つのピンを必要とします。

図3-3. AVR® Dx - 相互容量動作でのPTC構成図



自己容量動作では各接触感知器に対してPTCが1つのピン(Y線)を必要とします。

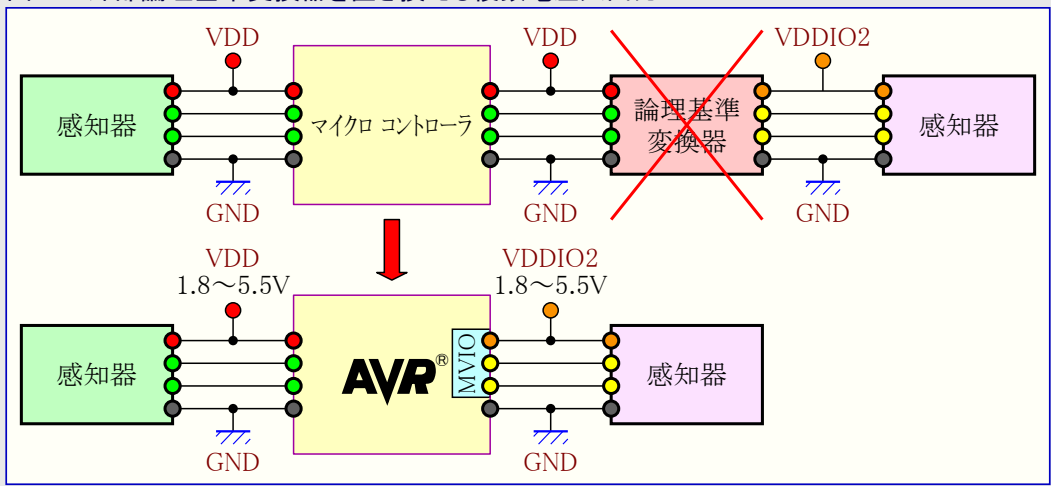
図3-4. AVR® Dx - 自己容量動作でのPTC構成図



### 3.8. MVIO - 複数電圧入出力

MVIO機能は残りの入出力ピンと違う入出力電圧領域によって給電されることを一群の入出力ピンに許します。これは違う電圧基準で動いている外部部品の通信や制御のために外部の基準変換器を持つ必要性をなくします。外部の論理基準変換器省略は更に部品を減らしてPCB空間を空けます。

図3-5. 外部論理基準変換器を置き換える複数電圧入出力

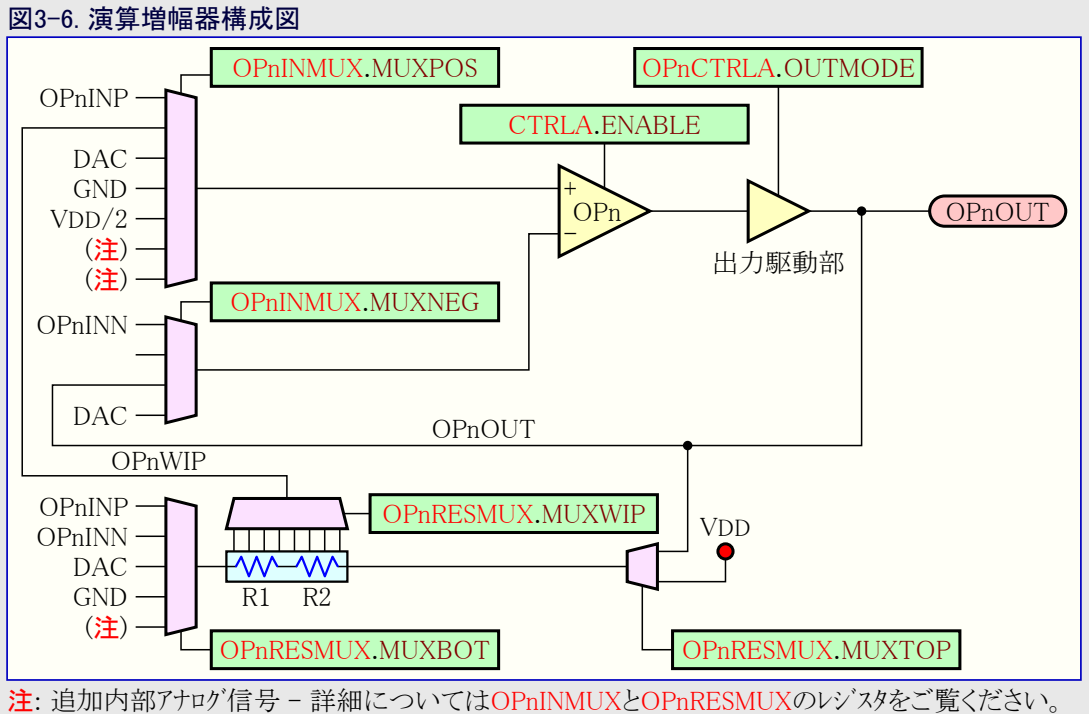


注: より多くの詳細とコード例については「TB3287 - 複数電圧入出力(MVIO)での開始に際して」を参照してください。

### 3.9. OPAMP - アナログ信号調整

アナログ信号調整(OPAMP)周辺機能は3つまでの演算増幅器が特徴です。これらの演算増幅器はアナログ多重器と梯子型抵抗を使う柔軟な接続の仕組みで実装されます。これは達成するための多数のアナログ信号調整構成設定を許し、その多くは外部部品を必要としません。

各演算増幅器の非反転(+)入力での多重器は外部ピン、梯子型抵抗の摺動子位置、DAC出力、接地、VDD/2、別の演算増幅器からの出力のどれかへの接続を許します。各演算増幅器の反転(-)入力での2つ目の多重器は外部ピン、梯子型抵抗の摺動子位置、演算増幅器の出力、DAC出力のどれかへの接続を許します。梯子型抵抗に接続される後3つの多重器は付加的な構成の柔軟性を提供します。これらの多重器の2つは梯子型抵抗への上側と下側の接続を選び、3つ目は摺動子位置を制御します。



注: OPAMP周辺機能を使うことについてコード例と詳細に関しては「TB3286 - アナログ信号調整(OPAMP)での開始に際して」を参照してください。

#### 4. 参考文献

- ATmega128データシート
- AVR128DA64データシート
- AVR128DB64データシート

追加文書は[www.microchip.com](http://www.microchip.com)で見つけることができます。

#### 5. 改訂履歴

資料改訂	日付	注釈
A	2020年11月	初版文書公開

## Microchipウェブ サイト

Microchipは[www.microchip.com/](http://www.microchip.com/)で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- **製品支援** - データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **一般的な技術支援** - 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- **Microshipの事業** - 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

## 製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するには[www.microchip.com/pcn](http://www.microchip.com/pcn)へ行って登録指示に従ってください。

## お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 組み込み解決技術者(ESE:Embedded Solutions Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援は[www.microchip.com/support](http://www.microchip.com/support)でのウェブ サイトを通して利用できます。

## Microchipデバイスコード保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が安全であると考えます。
- Microchipデバイスのコード保護機能を破ろうとする試みに使われる不正でおそらく違法な方法があります。当社はこれらの方法がMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要とされると確信しています。これらのコード保護機能を破ろうとする試みは、おそらく、Microchipの知的財産権に違反することなく達成することはできません。
- Microchipはそのコードの完全性について心配されている何れのお客様とも共に働きたいと思えます。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が”破ることができない”ことを当社が保証すると言うことを意味しません。コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

## 法的通知

この刊行物含まれる情報はMicrochip製品を使って設計する唯一の目的のために提供されます。デバイス応用などに関する情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責することに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

---

## 商標

---

Microchipの名前とロゴ、Microchip、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、Hyper Light Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、Inter-Chip Connectivity、JitterBlocker、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICKtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect、and ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adaptecロゴ、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

---

## 品質管理システム

---

Microchipの品質管理システムに関する情報については[www.microchip.com/quality](http://www.microchip.com/quality)を訪ねてください。

---

日本語© HERO 2020.

本応用記述はMicrochipのAN3731応用記述(DS00003731A-2020年11月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。



**MICROCHIP**

## 世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
<b>本社</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: <a href="http://www.microchip.com/support">www.microchip.com/support</a> ウェブアドレス: <a href="http://www.microchip.com">www.microchip.com</a> <b>アトランタ</b> Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 <b>オースチン TX</b> Tel: 512-257-3370 <b>ボストン</b> Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 <b>シカゴ</b> Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 <b>ダラス</b> Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 <b>デトロイト</b> Novi, MI Tel: 248-848-4000 <b>ヒューストン TX</b> Tel: 281-894-5983 <b>インディアナポリス</b> Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 <b>ロサンゼルス</b> Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 <b>ローリー NC</b> Tel: 919-844-7510 <b>ニューヨーク NY</b> Tel: 631-435-6000 <b>サンホセ CA</b> Tel: 408-735-9110 Tel: 408-436-4270 <b>カナダ - トロント</b> Tel: 905-695-1980 Fax: 905-695-2078	<b>オーストラリア - シドニー</b> Tel: 61-2-9868-6733 <b>中国 - 北京</b> Tel: 86-10-8569-7000 <b>中国 - 成都</b> Tel: 86-28-8665-5511 <b>中国 - 重慶</b> Tel: 86-23-8980-9588 <b>中国 - 東莞</b> Tel: 86-769-8702-9880 <b>中国 - 広州</b> Tel: 86-20-8755-8029 <b>中国 - 杭州</b> Tel: 86-571-8792-8115 <b>中国 - 香港特別行政区</b> Tel: 852-2943-5100 <b>中国 - 南京</b> Tel: 86-25-8473-2460 <b>中国 - 青島</b> Tel: 86-532-8502-7355 <b>中国 - 上海</b> Tel: 86-21-3326-8000 <b>中国 - 瀋陽</b> Tel: 86-24-2334-2829 <b>中国 - 深圳</b> Tel: 86-755-8864-2200 <b>中国 - 蘇州</b> Tel: 86-186-6233-1526 <b>中国 - 武漢</b> Tel: 86-27-5980-5300 <b>中国 - 西安</b> Tel: 86-29-8833-7252 <b>中国 - 廈門</b> Tel: 86-592-2388138 <b>中国 - 珠海</b> Tel: 86-756-3210040	<b>インド - ハンガロール</b> Tel: 91-80-3090-4444 <b>インド - ニューデリー</b> Tel: 91-11-4160-8631 <b>インド - フネー</b> Tel: 91-20-4121-0141 <b>日本 - 大阪</b> Tel: 81-6-6152-7160 <b>日本 - 東京</b> Tel: 81-3-6880-3770 <b>韓国 - 大邱</b> Tel: 82-53-744-4301 <b>韓国 - ソウル</b> Tel: 82-2-554-7200 <b>マレーシア - クアラルンプール</b> Tel: 60-3-7651-7906 <b>マレーシア - ペナン</b> Tel: 60-4-227-8870 <b>フィリピン - マニラ</b> Tel: 63-2-634-9065 <b>シンガポール</b> Tel: 65-6334-8870 <b>台湾 - 新竹</b> Tel: 886-3-577-8366 <b>台湾 - 高雄</b> Tel: 886-7-213-7830 <b>台湾 - 台北</b> Tel: 886-2-2508-8600 <b>タイ - バンコク</b> Tel: 66-2-694-1351 <b>ベトナム - ホーチミン</b> Tel: 84-28-5448-2100	<b>オーストラリア - ウェルズ</b> Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 <b>デンマーク - コペンハーゲン</b> Tel: 45-4485-5910 Fax: 45-4485-2829 <b>フィンランド - エスポー</b> Tel: 358-9-4520-820 <b>フランス - パリ</b> Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 <b>ドイツ - ガルヒング</b> Tel: 49-8931-9700 <b>ドイツ - ハーン</b> Tel: 49-2129-3766400 <b>ドイツ - ハイムブロン</b> Tel: 49-7131-72400 <b>ドイツ - カールスルーエ</b> Tel: 49-721-625370 <b>ドイツ - ミュンヘン</b> Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 <b>ドイツ - ローゼンハイム</b> Tel: 49-8031-354-560 <b>イスラエル - ラーナナ</b> Tel: 972-9-744-7705 <b>イタリア - ミラノ</b> Tel: 39-0331-742611 Fax: 39-0331-466781 <b>イタリア - ハドバ</b> Tel: 39-049-7625286 <b>オランダ - デルフト</b> Tel: 31-416-690399 Fax: 31-416-690340 <b>ノルウェー - トロンハイム</b> Tel: 47-72884388 <b>ポーランド - ワルシャワ</b> Tel: 48-22-3325737 <b>ルーマニア - ブカレスト</b> Tel: 40-21-407-87-50 <b>スペイン - マドリッド</b> Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 <b>スウェーデン - イェテボリ</b> Tel: 46-31-704-60-40 <b>スウェーデン - ストックホルム</b> Tel: 46-8-5090-4654 <b>イギリス - ウォーキングム</b> Tel: 44-118-921-5800 Fax: 44-118-921-5820