
アナログ信号生成用10ビットDACの使い方

序説

著者: Cristian Sabiuta, Marius Nicolae, Microchip Technology Inc.

AVR® DA MCU系統のマイクロ コントローラはAVR構造に基づくマイクロ コントローラで、10ビット分解能と高駆動能力を装備するDAC周辺機能をもたらし、正確なアナログ電圧を生成してそれらを内部的にまたは物理ピンで外部的に使う使用者に役立ちます。

この技術概説はAVR DAマイクロ コントローラ系統で10ビットDACがどう動くかを記述し、以下の使用事例を網羅します。

- 10ビットDACを使う一定アナログ信号生成

DAC初期化、電圧参照基準設定、特定の定電圧を出力するようにDACを設定

- 10ビットDACを使う正弦波信号生成

DAC初期化、電圧参照基準設定、正弦波の見本を繰り返して出力

- ADCでの内部的DAC読み取り

DACとADCを初期化、電圧参照基準設定、DACを読むようにADCを設定、DAC出力を増して各段階に対してADCでそれを読み取り

- 10ビットDACを使う振幅変調信号生成

外部参照基準でDACを初期して外部参照基準に対して変調されなければならない信号を接続

注: コード例はAVR128DA48 Curiosity Nanoで開発されました。

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
1.1. AVR® DA系概要	3
2. 概要	3
3. 10ビットDACを使う一定アナログ信号生成	3
4. 10ビットDACを使う正弦波信号生成	6
5. ADCでの内部的DAC読み取り	6
6. 10ビットDACを使う振幅変調信号生成	8
7. 参考資料	9
8. 追補	10
9. 改訂履歴	15
Microchipウェブサイト	16
製品変更通知サービス	16
お客様支援	16
Microchipデバイスコード保護機能	16
法的通知	16
商標	17
品質管理システム	17
世界的な販売とサービス	18

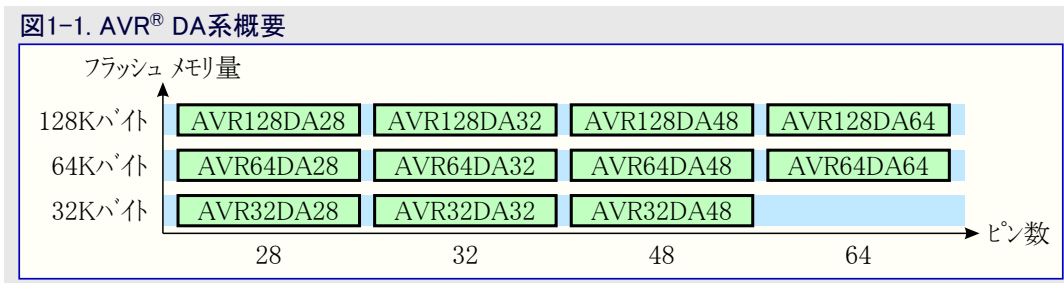
1. 関連デバイス

本章はこの文書に関連するデバイスを一覧にします。

1.1. AVR[®] DA系概要

下図はピン数の変種とメモリ量を展開してAVR[®] DAデバイスを示します。

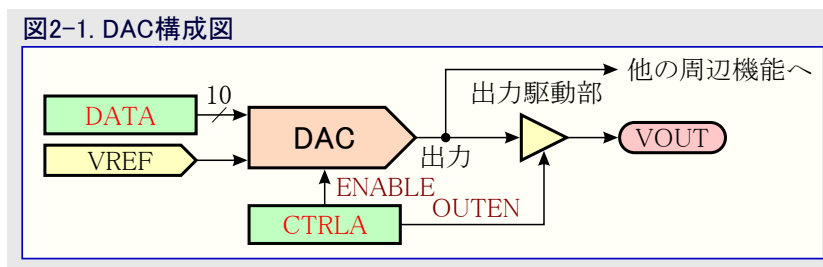
- これらのデバイスが完全にピンと機能が互換のため、垂直方向移植はコード変更なしで可能です。
- 左への水平方向移植はピン数、従って利用可能な機能を減らします。



異なるフラッシュメモリ量を持つデバイスは一般的にSRAM量も異なります。

2. 概要

DACは10ビット分解能が特徴で高駆動能力での1つの継続時間出力を持ちます。DAC変換はデータ(DACn.DATAn)レジスタ対へ書くことによって応用から開始することができます。



信号	形式	説明
VOUT	アナログ出力	D/A変換器出力

3. 10ビットDACを使う一定アナログ信号生成

DACは一定のアナログ信号を生成するのに使うことができます。正の参照基準として基準電圧(VREF)周辺機能の出力を使います。

DAC出力範囲は $0 \sim (1023 \times VREF) / 1024$ (V)の範囲です。

VREFは以下の予め定義された一覧から選ぶことができます。

- 内部1.024V参照基準
- 内部2.048V参照基準
- 内部4.096V参照基準
- 内部2.500V参照基準
- VDD参照基準
- VREFA(PD7)ピンからの外部参照基準

図3-1. VREF.DACOREFレジスタ

ビット	7	6	5	4	3	2	1	0
	ALWAYSON						REFSEL2~0	
アクセス種別	R/W	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

●ビット7 – ALWAYSON : 参照基準常時ON (Reference Always ON)

このビットはDAC0参照基準が常時ONか否かどちらかを制御します。

値	0	1
説明	参照基準は必要とされる時に自動的に許可されます。	参照基準は常にONです。

●ビット2~0 – REFSEL2~0 : 参照基準選択 (Reference Select)

このビット領域はDAC0に対する参照基準電圧水準を制御します。

注: 内部参照基準に対して与えられた値は代表値だけです。更なる詳細については「電気的特性」章を参照してください。

値	000	001	010	011	100	101	110	111
名称	1V024	2V048	4V096	2V500	-	VDD	VREFA	-
説明	内部1.024V参照基準	内部2.048V参照基準	内部4.096V参照基準	内部2.500V参照基準	(予約)	参照基準としてVDD	VREFAピンからの外部参照基準	(予約)

この例の目的に対しては2.048V参照基準電圧が選ばれました。

```
VREF.DACOREF = VREF_REFSEL_2V048_gc;
```

VREF周辺機能許可後に50 μ s遅延が推奨されます。

図3-2. 内部基準電圧(VREF)特性

シンボル	項目	条件	Min	Typ	Max	単位
VVREF_1V024	内部電圧参照基準1.024V	VDD \geq 2.5V, -40~85 $^{\circ}$ C	-4	-	+4	%
VVREF_2V048	内部電圧参照基準2.048V		-4	-	+4	
VVREF_4V096	内部電圧参照基準4.096V	VDD \geq 4.55V, -40~85 $^{\circ}$ C	-4	-	+4	
VVREF_2V500	内部電圧参照基準2.500V		-4	-	+4	
TVREF_ST	VREF始動時間		-	50	-	μ s

```
_delay_us(50);
```

DAC出力は他の周辺機能によって内部的に使われるか、または出力ピンに繋げることができます。AVR128DA48についてDAC出力はPD6ピンに接続されます(下図をご覧ください)。

図3-3. PORT機能多重化

VQFN64/ TQFP64	VQFN48/ TQFP48	VQFN32/ TQFP32	SSIC28/ SSOP28	ピン名 (注1,2)	特殊	ADC0	PTC	ACn	DAC0	ZCDn	USARTn	SPIn	TWIn (注4)	TCA0	TCA1	TCBn	TCDn	EVSYS	CCL-LUTn
30	24	14	10	PD4		AIN4	X20/Y20	1,AINP2 2,AINP1						WO4					
31	25	15	11	PD5		AIN5	X21/Y21	1,AINN0						WO5					
32	26	16	12	PD6		AIN6	X22/Y22	0,AINP3 1,AINP3 2,AINP3	VOUT										2,OUT
33	27	17	13	PD7	VREFA	AIN7	X23/Y23	0,AINN2 1,AINN2 2,AINN0/AINN2											EVOUTD

DAC出力ピンは負荷を減らすために禁止されたデジタル入力緩衝部とプルアップ抵抗を持つことが必要です。

```
PORTD.PIN6CTRL &= ~PORT_ISC_gm;
PORTD.PIN6CTRL |= PORT_ISC_INPUT_DISABLE_gc;
PORTD.PIN6CTRL &= ~PORT_PULLUPEN_bm;
```

データ(DACn.DATA)レジスタは指定したアナログ出力電圧を生成するのに使われます。この出力電圧の値は次式を使って決めることができます。

$$V_{OUT} = \frac{(DACn.DATA \times V_{REF})}{1024}$$

初期化でのDACn.DATAレジスタ書き込みは任意選択ですが、初めからDAC出力を指定の電圧にするのに有用です。DACは10ビット分解能が特徴で、データ下位(DACn.DATAL)とデータ上位(DACn.DATAH)のレジスタ対が10ビット値のDACn.DATAを表します(図3-4をご覧ください)。下位2ビット(DATA1,0)は変位原点でアクセス可能で、上位8ビット(DATA9~2)は変位原点+1でアクセスすることができます。出力はDACn.DATAHが書かれた後で更新されます。

図3-4. DACn.DATALレジスタ

ビット	15	14	13	12	11	10	9	8
	DATA9~2							
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0
ビット	7	6	5	4	3	2	1	0
	DATA1,0							
アクセス種別	R/W	R/W	R	R	R	R	R	R
リセット値	0	0	0	0	0	0	0	0

● ビット15~6 - DATA9~0 : データ (Data)


これらのビットはアナログ電圧に変換されるべきデジタル データを含みます。

この例でDACに対して望む出力は1.2Vです。これを達成するため、次の式が適用されます。

$$DACn.DATA = \frac{(V_{OUT} \times 1024)}{V_{REF}} = \frac{(1.2V \times 1024)}{2.048V} = 600 = 0x258$$

DAC、出力緩衝部、スタンバイで走行動作を許可するため、次のコードを使ってください。

```
DAC0.CTRLA = DAC_ENABLE_bm | DAC_OUTEN_bm | DAC_RUNSTDBY_bm;
```

 **重要:** スタンバイで走行動作が許可された場合、DACはマイクロ コントローラがスタンバイ休止動作の時に走行を続けます。

変換開始

DACが許可される(制御A(DACn.CTRLA)レジスタの許可(ENABLE)=1)と、変換はデータ(DACn.DATA)レジスタが書かれると直ぐに始まります。

DACが禁止される(DACn.CTRLAのENABLE=0)と、データレジスタへの書き込みは変換を起動しません。代わりに、変換はDACn.CTRLAレジスタのENABLEビットに'1'を書くことで始まります。

```
DAC0.DATAL = (value & (0x03)) << 6;
DAC0.DATAH = value >> 2;
```

変換後、出力はDACが動いている限り、次の変換まで(DACn.DATA×VREF)/1024の値を保ちます。VREF選択のどの変更も(許可されて動いていれば、)直ちにDAC出力を変更します。



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。



助言: 完全なコード例は「8. 追補」でも利用可能です。

4. 10ビットDACを使う正弦波信号生成

DACは正弦波信号を生成するのに使うことができます。

この信号を生成するには最初にVREFとDACが初期化され、その後にデータ(ADCn.DATA)レジスタに新しい値を書くことによって出力値を変えることができます。

正弦波が生成される前に、周期に対応する見本が計算されて緩衝部に格納されます。

```
for(i = 0; i < SINE_PERIOD_STEPS; i++)
{
    sineWave[i] = SINE_DC_OFFSET + SINE_AMPLITUDE * sin(2 * M_PI * i / SINE_PERIOD_STEPS);
}
```

正弦状波形は固定の段階数(N_SAMPLES)を使って作成されます。指定周波数(SIGNAL_FREQ)で正弦波信号を作成するには、次の標準化速度(SAMPLE_RATE)に帰着する1周期で全ての段階が実行されます。

$$\text{SAMPLE_RATE} = \frac{1}{\text{STEP_DELAY_TIME}} = \text{SIGNAL_FREQ} \times \text{N_SAMPLES}$$

```
while (1)
{
    DAC0_setVal(sineWave[sineIndex++]);
    if(sineIndex == SINE_PERIOD_STEPS)
        sineIndex = 0;
    _delay_us(STEP_DELAY_TIME);
}
```



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。



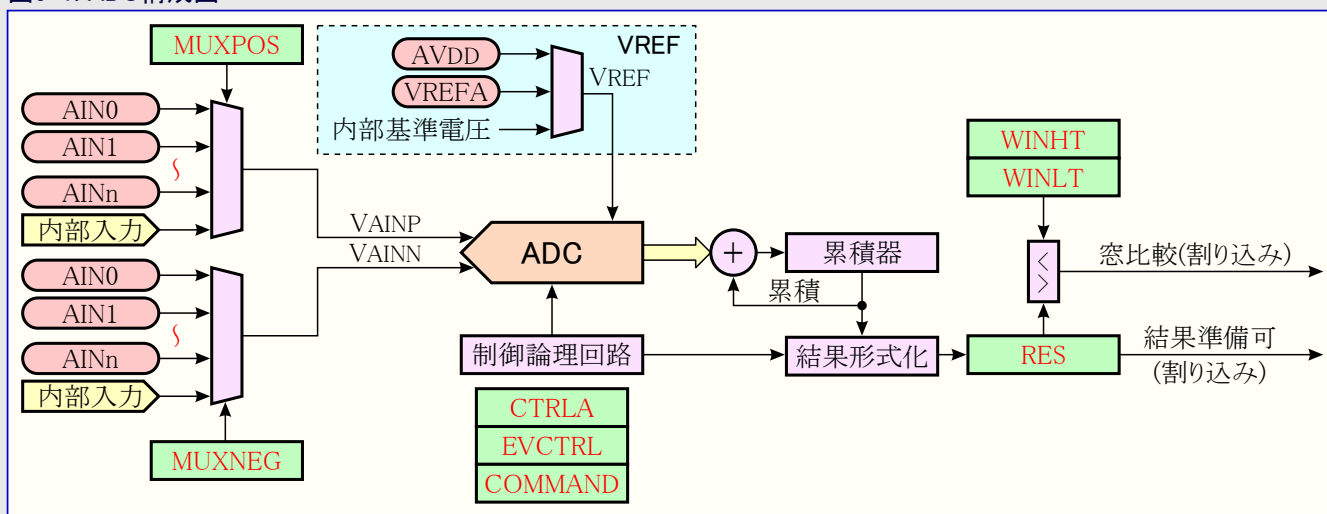
助言: 完全なコード例は「8. 追補」でも利用可能です。

5. ADCでの内部的DAC読み取り

DACのアナログ出力はDACが許可される(制御A(DACn.CTRLA)レジスタの許可(ENABLE)=1)時に他の周辺機能へ内部的に接続することができます。DACアナログ出力が内部的にだけ使われる時はピン出力駆動部を許可する必要がありません(DACn.CTRLAの出力緩衝部許可(OUTEN)=0は受け入れられます)。

下のADC構成図を参照すると、10ビットDACの出力はADCへの内部入力として使うことができます。

図5-1. ADC構成図



DAC電圧参照基準は「3. 10ビットDACを使う一定アナログ信号生成」で言及されたように初期化され、ADC電圧参照基準はVREF周辺機能から同じ様に初期化されます。

図5-2. VREF.DACOREFレジスタ

ビット	7	6	5	4	3	2	1	0
	ALWAYSON						REFSEL2~0	
アクセス種別	R/W	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット7 - ALWAYSON : 参照基準常時ON (Reference Always ON)

このビットはDAC0参照基準が常時ONか否かどちらかを制御します。

値	0	1
説明	参照基準は必要とされる時に自動的に許可されます。	参照基準は常にONです。

● ビット2~0 - REFSEL2~0 : 参照基準選択 (Reference Select)

このビット領域はDAC0に対する参照基準電圧水準を制御します。

注: 内部参照基準に対して与えられた値は代表値だけです。更なる詳細については「電気的特性」章を参照してください。

値	000	001	010	011	100	101	110	111
名称	1V024	2V048	4V096	2V500	-	VDD	VREFA	-
説明	内部1.024V参照基準	内部2.048V参照基準	内部4.096V参照基準	内部2.500V参照基準	(予約)	参照基準としてVDD	VREFAピンからの外部参照基準	(予約)

図5-3. VREF.ADCOREFレジスタ

ビット	7	6	5	4	3	2	1	0
	ALWAYSON						REFSEL2~0	
アクセス種別	R/W	R	R	R	R	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット7 - ALWAYSON : 参照基準常時ON (Reference Always ON)

このビットはADC0参照基準が常時ONか否かどちらかを制御します。

値	0	1
説明	参照基準は必要とされる時に自動的に許可されます。	参照基準は常にONです。

● ビット2~0 - REFSEL2~0 : 参照基準選択 (Reference Select)

このビット領域はADC0に対する参照基準電圧水準を制御します。

注: 内部参照基準に対して与えられた値は代表値だけです。更なる詳細については「電気的特性」章を参照してください。

値	000	001	010	011	100	101	110	111
名称	1V024	2V048	4V096	2V500	-	VDD	VREFA	-
説明	内部1.024V参照基準	内部2.048V参照基準	内部4.096V参照基準	内部2.500V参照基準	(予約)	参照基準としてVDD	VREFAピンからの外部参照基準	(予約)

完全なVREF初期化が下で示されます。

```
VREF.DACOREF = VREF_REFSEL_2V048_gc /* DAC用に2.048V内部電圧参照基準を選択 */
                | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
VREF.ADCOREF = VREF_REFSEL_2V048_gc /* ADC用に2.048V内部電圧参照基準を選択 */
                | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
/* VREF始動時間待ち */
_delay_us (VREF_STARTUP_TIME);
```

その後、ADCが初期化されなければなりません。

```
ADC0.CTRLA = ADC_PRESC_DIV2_gc;
ADC0.CTRLA = ADC_ENABLE_bm | ADC_RESSEL_12BIT_gc;
```

ADCでDACを読むため、ADCの正入力多重器選択(MUXPOS)レジスタがDAC0に対応する\$48に設定されなければなりません。

```
ADC0.MUXPOS = ADC_MUXPOS_DAC0_gc;
```

図5-4. ADC0.MUXPOS選択

ビット	7	6	5	4	3	2	1	0
	MUXPOS6~0							
アクセス種別	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● ビット6~0 - MUXPOS6~0 : 多重器正選択 (MUX Selection for Positive ADC Input)

このビット領域はどのアナログ入力があるADCの正入力に接続されるかを選びます。このビット領域が変換中に変えられる場合、その変更はこの変換が完了するまで無効です。

値	\$00~\$15	\$40	\$42	\$48	\$49	\$4A	\$4B	その他
名称	AIN0~AIN21	GND	TEMPSENSE	DAC0	DACREF0	DACREF1	DACREF2	-
説明	ADC入力0~21ピン	接地	温度感知器	DAC0	DACREF0	DACREF1	DACREF2	(予約)

ADC変換は指令(ADCn.COMMAND)レジスタに対応するビットを書くことによって開始されます。

```
ADC0.COMMAND = ADC_STCONV_bm;
```

変換が終わると、ハードウェアによって割り込み要求フラグ(ADCn.INTFLAGS)レジスタで結果準備可(RESRDY)ビットが設定(1)されます。

```
while (!(ADC0.INTFLAGS & ADC_RESRDY_bm))
{
    ;
}
```

このフラグはRESRDYビット位置に'1'を書くか、または結果(ADCn.RES)レジスタを読むかのどちらかによって解除(0)されます。このビットへの'0'書き込みは無効です。

```
while (1)
{
    adcVal = ADC0_read();
    dacVal++;
    DAC0_setVal(dacVal);
}
```



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。



助言: 完全なコード例は「8. 追補」でも利用可能です。

6. 10ビットDACを使う振幅変調信号生成

DACは振幅変調した信号を生成するのに使うことができます。DACは変調波形(情報信号)としてVREFAピンからの電圧参照基準を使い、同時に搬送信号用の正弦波を生成するように構成設定されます。

```
VREF.DACOREF = VREF_REFSEL_VREFA_gc;
```

$$V_{OUT} = \frac{(DACn.DATA \times VREFA)}{1024}$$

従って、搬送信号の振幅は変調されることが必要な信号に従って変わります。



重要: 搬送信号周波数は適切な結果信号のため情報信号周波数よりも高くなければなりません。

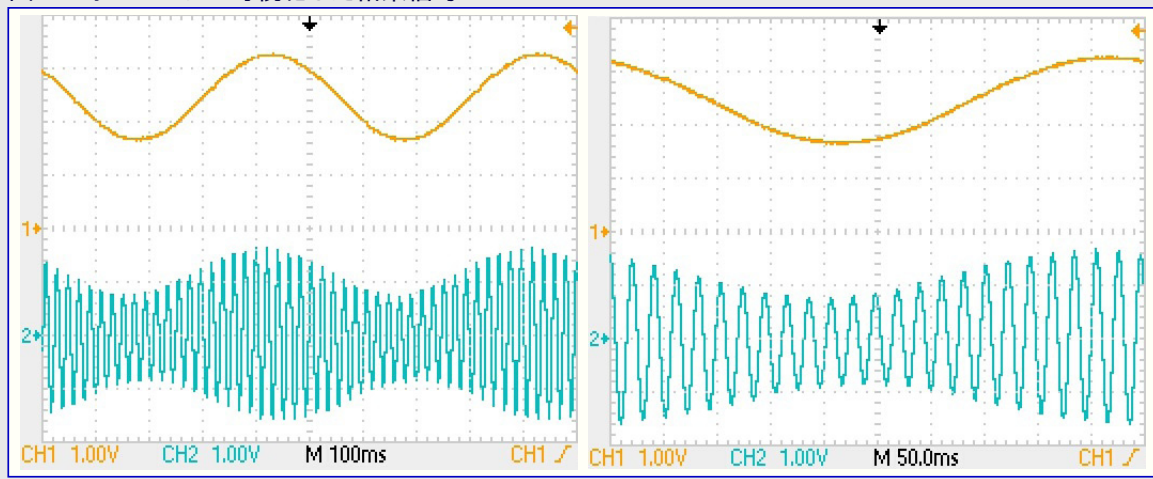
搬送波周波数(SIGNAL_FREQ)に基づき、「4. 10ビットDACを使う正弦波信号生成」で記述されるように信号を構築するために固定の見本数(N_SAMPLES)を使います。結果の標本化速度(SAMPLE_RATE)は次のとおりです。

```
SAMPLE_RATE = N_SAMPLES * SIGNAL_FREQ
```

結果の変調された信号はDAC外部出力(PD6)ピンで利用可能です。

この例では搬送信号として50Hz正弦波が使われ、変調される信号は信号発生器からの2Hz正弦波です。

図6-1. オシロスコープで可視化した結果信号



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。



助言: 完全なコード例は「[8. 追補](#)」でも利用可能です。

7. 参考資料

1. AVR128DA28/32/48/64 暫定データシート
2. AVR128DA48 Curiosity Nano 使用者の手引き

8. 追補

例8-1. 一定アナログ信号生成コード

```

/* 4MHz(遅延関数に必要) */
#define F_CPU          (4000000UL)

#include <avr/io.h>
#include <util/delay.h>

/* DAC値 */
#define DAC_EXAMPLE_VALUE    (0x258)
/* VREF始動時間 */
#define VREF_STARTUP_TIME    (50)
/* DACデータレジスタの2LSBを取得するのに必要とされる遮蔽 */
#define LSB_MASK             (0x03)

static void VREF_init(void);
static void DAC0_init(void);
static void DAC0_setVal(uint16_t value);

static void VREF_init(void)
{
    VREF.DACOREF = VREF_REFSSEL_2V048_gc /* DAC用に2.048V内部電圧参照基準を選択 */
                 | VREF_ALWAYS_ON_bm;   /* 常時ON動作で電圧参照基準を設定 */
    /* VREF始動時間待ち */
    _delay_us(VREF_STARTUP_TIME);
}

static void DAC0_init(void)
{
    /* 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を許可 */
}

static void DAC0_setVal(uint16_t value)
{
    /* DAC0.DATALに2LSBを格納 */
    DAC0.DATAL = (value & LSB_MASK) << 6;
    /* DAC0.DATAHに8MSBを格納 */
    DAC0.DATAH = value >> 2;
}

int main(void)
{
    VREF_init();
    DAC0_init();
    DAC0_setVal(DAC_EXAMPLE_VALUE);

    while (1)
    {
        ;
    }
}

```

例8-2. 正弦波信号生成コード

```

/* 4MHz(遅延関数に必要) */
#define F_CPU          (4000000UL)

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>

/* VREF始動時間 */
#define VREF_STARTUP_TIME    (50)
/* DACデータレジスタの2LSBを取得するのに必要とされる遮蔽 */
#define LSB_MASK            (0x03)
/* 正弦波周期用見本(標本化)数 */
#define SINE_PERIOD_STEPS   (100)
/* 正弦波振幅 */
#define SINE_AMPLITUDE      (511)
/* 正弦波DC変位 */
#define SINE_DC_OFFSET      (512)
/* 正弦波の周波数 */
#define SINE_FREQ           (100)
/* 繰り返し用段階遅延 */
#define STEP_DELAY_TIME     ((1000000 / SINE_FREQ) / SINE_PERIOD_STEPS)

static void sineWaveInit(void);
static void VREF_init(void);
static void DAC0_init(void);
static void DAC0_setVal(uint16_t value);

/* 正弦波見本を格納するための緩衝部 */
uint16_t sineWave[SINE_PERIOD_STEPS];

static void sineWaveInit(void)
{
    uint8_t i;
    for(i = 0; i < SINE_PERIOD_STEPS; i++)
    {
        sineWave[i] = SINE_DC_OFFSET + SINE_AMPLITUDE * sin(2 * M_PI * i / SINE_PERIOD_STEPS);
    }
}

static void VREF_init(void)
{
    VREF.DACOREF = VREF_REFSEL_2V048_gc /* DAC用に2.048V内部電圧参照基準を選択 */
                 | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
    /* VREF始動時間待ち */
    _delay_us(VREF_STARTUP_TIME);
}

static void DAC0_init(void)
{
    /* 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を許可 */
}

```

```

static void DAC0_setVal(uint16_t value)
{
    /* DAC0.DATALに2LSBを格納 */
    DAC0.DATAL = (value & LSB_MASK) << 6;
    /* DAC0.DATAHに8MSBを格納 */
    DAC0.DATAH = value >> 2;
}

int main(void)
{
    uint8_t sineIndex = 0;

    VREF_init();
    DAC0_init();

    sineWaveInit();

    while (1)
    {
        DAC0_setVal(sineWave[sineIndex++]);
        if(sineIndex == SINE_PERIOD_STEPS)
            sineIndex = 0;
        _delay_us(STEP_DELAY_TIME);
    }
}

```

例8-3. ADCで内部的にDAC読み取りコード

```

/* 4MHz(遅延関数に必要) */
#define F_CPU (4000000UL)

#include <avr/io.h>
#include <util/delay.h>

/* VREF始動時間 */
#define VREF_STARTUP_TIME (50)
/* DACデータレジスタの2LSBを取得するのに必要とされる遮蔽 */
#define LSB_MASK (0x03)

static void VREF_init(void);
static void DAC0_init(void);
static void DAC0_setVal(uint16_t val);
static void ADC0_init(void);
static uint16_t ADC0_read(void);

static void VREF_init(void)
{
    VREF.DACOREF = VREF_REFSSEL_2V048_gc /* DAC用に2.048V内部電圧参照基準を選択 */
                | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
    VREF.ADCOREF = VREF_REFSSEL_2V048_gc /* ADC用に2.048V内部電圧参照基準を選択 */
                | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
    /* VREF始動時間待ち */
    _delay_us(VREF_STARTUP_TIME);
}

static void DAC0_init(void)
{
    /* DAC許可 */
    DAC0.CTRLA = DAC_ENABLE_bm;
}

```

```

static void DAC0_setVal(uint16_t value)
{
    /* DAC0.DATALに2LSBを格納 */
    DAC0.DATAL = (value & LSB_MASK) << 6;
    /* DAC0.DATAHに8MSBを格納 */
    DAC0.DATAH = value >> 2;
}

static void ADC0_init(void)
{
    /* 2分周したCLK_PER */
    ADC0.CTRLA = ADC_PRESC_DIV2_gc;
    ADC0.CTRLA = ADC_ENABLE_bm          /* ADC許可 */
                | ADC_RESSEL_12BIT_gc; /* 12ビット分解能を使用 */
    /* ADCチャンネル選択 */
    ADC0.MUXPOS = ADC_MUXPOS_DAC0_gc
}

static uint16_t ADC0_read(void)
{
    /* Start conversion */
    ADC0.COMMAND = ADC_STCONV_bm;
    /* ADC変換終了まで待機 */
    while(!(ADC0.INTFLAGS & ADC_RESRDY_bm))
    {
        ;
    }
    /* 割り込み要求フラグは変換結果が読まれる時に解除(0)されます。 */
    return ADC0.RES;
}

int main(void) {
    uint16_t dacVal = 0;
    volatile uint16_t adcVal = 0;
    VREF_init();
    DAC0_init();
    ADC0_init();

    while (1)
    {
        adcVal = ADC0_read();

        /* adcValで何かをしてください。 */

        dacVal++;
        DAC0_setVal(dacVal);
    }
}

```

例8-4. 振幅変調信号生成コード

```

/* 4MHz(遅延関数に必要) */
#define F_CPU          (4000000UL)

#include <avr/io.h>
#include <util/delay.h>
#include <math.h>

/* VREF始動時間 */
#define VREF_STARTUP_TIME    (50)
/* DACデータレジスタの2LSBを取得するのに必要とされる遮蔽 */
#define LSB_MASK            (0x03)

```

```

/* 正弦波周期用見本(標本化)数 */
#define SINE_PERIOD_STEPS    (100)
/* 正弦波振幅 */
#define SINE_AMPLITUDE      (511)
/* 正弦波DC変位 */
#define SINE_DC_OFFSET      (512)
/* 正弦波の周波数 */
#define SINE_FREQ           (50)
/* 繰り返し用段階遅延 */
#define STEP_DELAY_TIME     ((1000000 / SINE_FREQ) / SINE_PERIOD_STEPS)

static void PORT_init (void);
static void sineWaveInit(void);
static void VREF_init(void);
static void DAC0_init(void);
static void DAC0_setVal(uint16_t val);

/* 正弦波見本を格納するための緩衝部 */
uint16_t sineWave[SINE_PERIOD_STEPS];

static void PORT_init (void)
{
    /* VREFA(PD7)ピンを入力として設定 */
    PORTD.DIRCLR |= PIN7_bm;
}

static void sineWaveInit(void)
{
    uint8_t i;
    for(i = 0; i < SINE_PERIOD_STEPS; i++)
    {
        sineWave[i] = SINE_DC_OFFSET + SINE_AMPLITUDE * sin(2 * M_PI * i / SINE_PERIOD_STEPS);
    }
}

static void VREF_init(void)
{
    VREF.DACOREF = VREF_REFSSEL_VREFA_gc /* VREFAピンからの外部電圧参照基準を選択 */
                 | VREF_ALWAYS_ON_bm; /* 常時ON動作で電圧参照基準を設定 */
    /* VREF始動時間待ち */
    _delay_us(VREF_STARTUP_TIME);
}

static void DAC0_init(void)
{
    /* 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を許可 */
}

static void DAC0_setVal(uint16_t value)
{
    /* DAC0.DATALに2LSBを格納 */
    DAC0.DATAL = (value & LSB_MASK) << 6;
    /* DAC0.DATAHに8MSBを格納 */
    DAC0.DATAH = value >> 2;
}

```

```

}

int main(void)
{
    uint8_t sineIndex = 0;

    PORT_init();
    VREF_init();
    DAC0_init();

    sineWaveInit();

    while (1)
    {
        DAC0_setVal(sineWave[sineIndex++]);
        if(sineIndex == SINE_PERIOD_STEPS)
            sineIndex = 0;
        _delay_us(STEP_DELAY_TIME);
    }
}

```

9. 改訂履歴

文書改訂	日付	注釈
A	2020年2月	初版文書公開
B	2020年3月	貯蔵庫リンク更新 最新の商標により、AVR-DAをAVR® MCU DA (AVR-DA)に更新
C	2020年5月	最新の商標により、AVR® MCU DA (AVR-DA)をAVR® DA MCUに、AVR-DAをAVR DAに更新

Microchipウェブ サイト

Microchipは<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<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<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、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全ては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です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を”破ることができない”として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

デバイス応用などに関してこの刊行物に含まれ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。Microchipはその条件、品質、性能、商品性、目的適合性を含め、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**。Microchipはこの情報とそれの使用から生じる全責任を否認します。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、免責に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、Mmicrochipロゴ、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、PackeTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTracker、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。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、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、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNetロゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、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<http://www.microchip.com/quality>を訪ねてください。

日本語© HERO 2020.

本技術概説はMicrochipのTB3235技術概説(DS90003235C-2020年5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。

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