

---

---

## USARTでの開始に際して

---

---

### 序説

著者: Alexandru Niculae, Microchip Technology Inc.

この資料の目的はmegaAVR<sup>®</sup> 0系とtinyAVR<sup>®</sup> 0及び1系でUSART周辺機能をどう構成設定するかを段階的に記述することです。これが複雑な周辺機能で様々な動作形態で動くことができるとは言え、この資料はそれを非同期動作で使い、以下の使用事例を記述します。

- 端末へ”Hello World”送出

PCに文字列を送出する方法を実演し、それを端末で見せます。

- 書式化した文字列送出/’printf’を使う文字列雛形送出

USART上で文字列を送るのに’printf’関数を使うことができる能力で最初の使用事例を強化します。

- 制御命令受信

度々、USARTは命令行インターフェースを実装するのに使われます。この方法で、マイクロコントローラはUSART経由で制御命令を受け取ることができます。

**注:** コード例はATmega4809 Xplained Pro (ATMEGA4809-XPRO)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. tinyAVR® 0系統	3
1.2. tinyAVR® 1系統	3
1.3. megaAVR® 0系統	3
2. 概要	4
3. “Hello World”送	4
4. 書式化した文字列送/printfを使う文字列雛形送	7
5. 制御命令受	7
6. 他の実装動作形態	9
6.1. 同期動作	9
6.2. 単線動作	9
7. 参照	10
8. 改訂履歴	10
Microchipウェブ サイト	11
製品変更通知サービス	11
お客様支援	11
Microchipデバイス コード保護機能	11
法的通知	11
商標	12
品質管理システム	12
世界的な販売とサービス	13

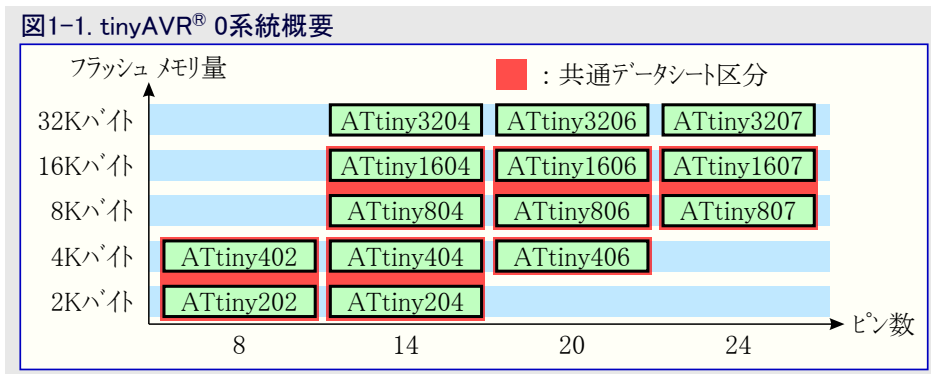
## 1. 関連デバイス

本章はこの資料に関連するデバイスを一覧にします。

### 1.1. tinyAVR<sup>®</sup> 0系統

下図はピン数の変種とメモリ量を展開してtinyAVR<sup>®</sup> 0系統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。

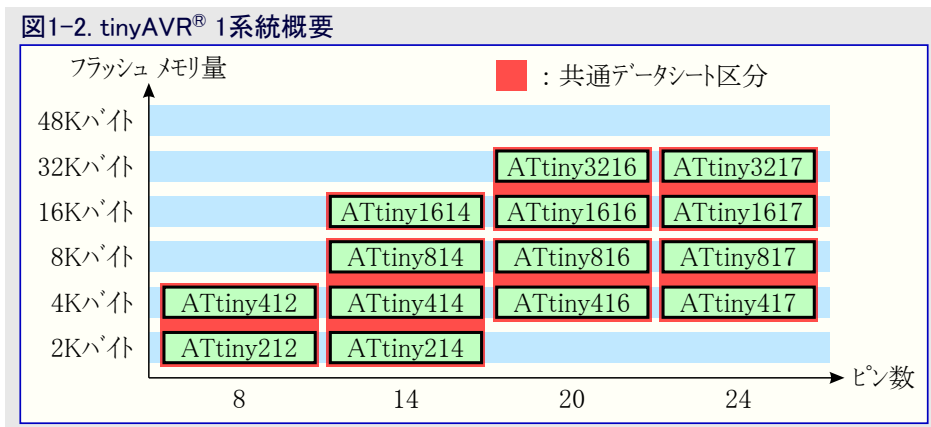


異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMの量を持ちます。

### 1.2. tinyAVR<sup>®</sup> 1系統

下図はピン配置変種とメモリ量を展開してtinyAVR<sup>®</sup> 1系統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に可能です。下方向移植はより少ない利用可能ないくつかの周辺機能の実体のためにコード変更が必要かもしれません。
- 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の量を持ちます。

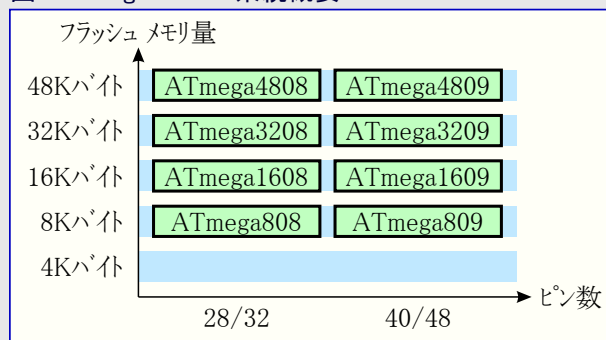
### 1.3. megaAVR<sup>®</sup> 0系統

右図はピン配置変種とメモリ量を展開してmegaAVR<sup>®</sup> 0系統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異なるSRAMとEEPROMの量を持ちます。

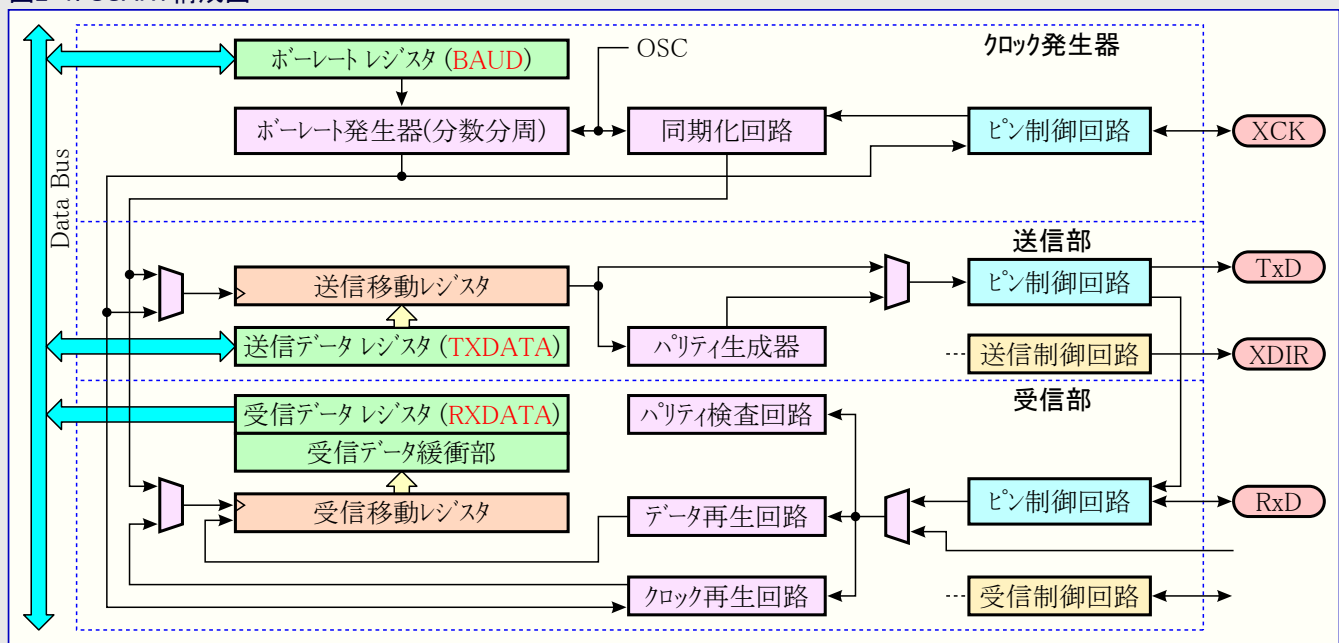
図1-3. megaAVR<sup>®</sup> 0系統概要



## 2. 概要

USART単位部はRX(受信)、TX(送信)、XCK(クロック)、XDIR(方向)と名付けられた4つのピンを持ちます。単線動作だけは、送信と受信の両方にTXピンが使われます。この動作の欠点は半二重通信だけを提供することです。非同期動作ではRXとTXの両ピンが使われ、従って全二重通信を成し遂げます。XCKピンは同期動作でのクロック信号に使われ、XDIRピンはRS485動作に使われます。

図2-1. USART構成図



最も一般的なUSART構成設定は”9600 8N1”として参照され、ボーレート9600、8データビット、パリティなし、1停止ビットを意味します。通常のUSARTフレームは従って10ビット(1開始ビット、8データビット、1停止ビット)を持ち、1つのASCII文字を表すことができます。これは”8N1”構成設定が秒毎にBAUD\_RATE/10 ASCII文字を送ることを意味します。

**注:** この資料で記述される全ての例はボーレート9600と”8N1”フレーム形式を使います。シリアル端末はこの構成設定に設定されなければなりません。

その上、USARTは複雑な周辺機能で以下のような少数の他の規約を達成するのに使うことができます。

- 主装置SPI
- 従装置LIN
- IR通信
- アドレス指定可能なUSART (または複数プロセッサ通信と呼ばれる)
- RS485

## 3. “Hello World”送出

この使用事例はPCに文字列をどう送るかを実演してそれを端末で見せます。(MCP2200のようなUSART-USB変換器とAtmel Studioのデータ可視器(Data Visualizer)のような端末ソフトウェアに対して利用可能な多数の任意選択があります。USARTは非同期動作でTXピンだけが使われるように構成設定されます。

**注:** マイクロコントローラのTXピンはUSART-USB変換器のRXピンに接続されなければなりません。RXピンも使われる場合、それはUSART-USB変換器のTXピンに接続されなければなりません。時に、デバイスは共通接地線も共有しなければなりません。

この使用事例は次のような段階に従います。

- ボーレート設定
- 送信部(TX)許可
- ピン構成設定

### ボーレート構成設定方法

ボーレートは秒毎にどの位のビットが送られるかを示します。より高いボーレートがより速い通信です。一般的なボーレートは、1200、2400、4800、9600、19200、38400、57600、115200で、9600が最も一般的に使われるものです。

megaAVR 0系では最大ボーレートが、非同期動作に於いて最大でUSARTクロックの1/8、同期動作に於いて最大でUSARTクロックの1/2に制限されます。ボーレートを設定するにはUSARTn.BAUDレジスタに書いてください。

```
USART0.BAUD = (uint16_t)USART0_BAUD_RATE(9600);
```

ボーレートからレジスタの値を計算するUSART0\_BAUD\_RATEマクロの使用に注意してください。このマクロは次図の式に基づいて定義されなければなりません。この式はUSARTの設定に依存し、故に他の動作で同じにならないかもしれません。

図3-1. ホーレートレジスタ設定を計算するための等式

動作形態	条件	ホーレート(bps)計算式	USARTn.BAUDレジスタ値計算式
非同期動作	$f_{\text{BAUD}} \leq \frac{f_{\text{CLK\_PER}}}{S}$	$f_{\text{BAUD}} = \frac{64 \times f_{\text{CLK\_PER}}}{S \times \text{BAUD}}$	$\text{BAUD} = \frac{64 \times f_{\text{CLK\_PER}}}{S \times f_{\text{BAUD}}}$
同期動作	$f_{\text{BAUD}} < \frac{f_{\text{CLK\_PER}}}{2}$	$f_{\text{BAUD}} = \frac{f_{\text{CLK\_PER}}}{2 \times \text{BAUD}[15 \sim 6]}$	$\text{BAUD}[15 \sim 6] = \frac{f_{\text{CLK\_PER}}}{2 \times f_{\text{BAUD}}}$

Sはビット当たりの採取数です。非同期動作では16(標準(NORMAL)動作)または8(倍速(CLK2X)動作)です。同期動作形態についてはS=2です。

これはUSART0\_BAUD\_RATEマクロがどう定義されるかです。USARTクロックがCPUクロックと一致するため、F\_CPUを使います。

```
#define F_CPU 3333333
#define USART0_BAUD_RATE(BAUD_RATE) ((float)(F_CPU * 64 / (16 * (float)BAUD_RATE)) + 0.5)
```

### 送信部を許可してデータを送る方法

応用の要求に応じて、使用者はUSART単位部の受信部または送信部だけを許可するように選ぶことができます。この使用事例ではマイクロコントローラがメッセージを送るだけなので、送信部だけ許可されることが必要です。

```
USART0_CTRLB |= USART_TXEN_bm;
```

データを送る前に、使用者はUSARTn.STATUSレジスタを調べることによって以前の送信が完了されているかを調べる必要があります。以下のコード例は送信データ(TXDATA)レジスタが空になるまで待つその後、USARTn.TXDATAレジスタへ文字を書きます。

```
void USART0_sendChar(char c)
{
    while (!(USART0.STATUS & USART_DREIF_bm))
    {
        ;
    }
    USART0.TXDATA = c;
}
```

送出レジスタは9ビット長です。従って、TXDATAと呼ばれる最初の8ビットを保持する下位側部分とTXDATAHと呼ばれる残りの1ビットを保持する上位側部分の2つの部分に分けられました。TXDATAHはUSARTが9ビットデータを使うように構成設定される時にだけ使われます。使われる時に、USARTn.CTRLCのCHSIZEがUSARTn.TXDATAが先に書かれるべき”9ビット-下位バイト先行”に設定される場合を除き、この第9ビットはUSARTn.TXDATAへ書く前に書かれなければなりません。

### ピン構成設定方法

TXピンは出力として構成設定されなければなりません。既定で、各周辺機能はいくつかの関連したピン位置を持ちます。ピンはデバイス特定データシートの「多重化された信号」項で見つけることができます。各USARTは2組のピン位置を持ちます。USART0に対する既定と代替のピン位置が下で示されます。

図3-2. 多重化された信号

QFN48/ TQFP48	ピン名(注1,2)	特殊	ADC0	AC0	UARTn	
44	PA0	EXTCLK			0-TxD	既定ピン位置
45	PA1				0-RxD	
46	PA2	TWI			0-XCK	
47	PA3	TWI			0-XDIR	
48	PA4				0-TxD	代替ピン位置
1	PA5				0-RxD	
2	PA6				0-XCK	
3	PA7	CLKOUT		OUT	0-XDIR	

この使用事例については既定USART0ピン位置が使われ、それはPA0～PA3です。以下のコードはTXピンの方向を出力に設定します。

```
PORTA.DIR |= PIN0_bm;
```

代替ピン位置を使うには以下のようにPORTMUX.USARTROUTEAレジスタを書いてください。

```
PORTMUX.USARTROUTEA |= PORTMUX_USART00_bm;
```

注: この例では代替ピン位置ではなく、既定ピン位置が使われます。

## 実演コード

以下のコードは継続的に”Hello World!”文字列を送ります。文字列は文字単位で送られます。’USART0\_sendString’関数は”Hello World!”文字列内の各文字に対して’USART0\_sendCharacter’関数を呼びます。各文字送出前に、’USART0\_sendCharacter’関数は直前の文字送信完了を待ちます。これはデータレジスタ空フラグのSTATUS.DREIFが設定(1)されるまでステータス(STATUS)レジスタをポーリングすることによって行われます。

```
#define F_CPU 3333333
#define USART0_BAUD_RATE(BAUD_RATE) ((float)(3333333 * 64 / (16 * (float)BAUD_RATE)) + 0.5)

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

void USART0_init(void);
void USART0_sendChar(char c);
void USART0_sendString(char *str);

void USART0_init(void)
{
    PORTA.DIR &= ~PIN1_bm;
    PORTA.DIR |= PIN0_bm;

    USART0.BAUD = (uint16_t)USART0_BAUD_RATE(9600);
    USART0.CTRLB |= USART_TXEN_bm;
}

void USART0_sendChar(char c)
{
    while (!(USART0.STATUS & USART_DREIF_bm))
    {
        ;
    }
    USART0.TXDATAL = c;
}

void USART0_sendString(char *str)
{
    for(size_t i = 0; i < strlen(str); i++)
    {
        USART0_sendChar(str[i]);
    }
}

int main(void)
{
    USART0_init();

    while (1)
    {
        USART0_sendString("Hello World!%r%rn");
        _delay_ms(500);
    }
}
```

**注:** 遅延関数を正しく動かすために<avr/delay.h>ヘッダがインクルードされる前にCPU周波数が定義されなければなりません。

**注:** フレーム構造体は構成設定されず、既定は”8N1”(8データビット、パリティなし、1停止ビット)です。CPUと周辺機能のクロック周波数も既定で3.33MHzです。



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。

## 4. 書式化した文字列送/printfを使う文字列雛形送

これは可変領域を持つ文字列を送るような応用、例えば、応用がその状態や計数器値を報告する時用の一般的な事例です。書式化した文字列の使用は非常に柔軟な手段でコード行数を減らします。これは'printf'関数の出力データ列の流れを変更することによって達成することができます。

この使用事例は次のこれらの手順に従います。

- USART周辺機能を最初の使用事例と同じに構成設定します。
- 使う定義済みデータ列を作成します。
- 使用者定義データ列の流れで標準出力データ列の流れを置き換えます。

通常、'printf'使用時、文字は標準出力データ列と呼ばれるデータ列の流れへ送られます。PCでは標準出力データ列が画面で文字を表示する関数によって処理されます。しかし、データ列の流れは別の関数がそれらのデータを処理するように作成することができます。

以下のコードはUSART0\_printChar関数によって処理される使用者定義データ列の流れを作成します。この関数はUSART\_sendChar関数の覆い部ですが、FDEV\_SETUP\_STREAMがパラメータとして期待するものと一致するように僅かに異なる用法を持ちます。

```
static void USART0_sendChar(char c)
{
    while (!(USART0.STATUS & USART_DREIF_bm))
    {
        ;
    }
    USART0.TXDATA1 = c;
}
static int USART0_printChar(char c, FILE *stream)
{
    USART0_sendChar(c);
    return 0;
}

static FILE USART_stream = FDEV_SETUP_STREAM(USART0_printChar, NULL, _FDEV_SETUP_WRITE);
```

その後にUSART送関数によって処理される使用者定義データ列の流れで標準出力データ列の流れを置き換えてください。

```
stdout = &USART_stream;
```

応用は今や直接USARTレジスタに書く代わりに'printf'を使うことができます。

```
uint8_t count = 0;
while (1)
{
    printf("Counter value is: %d\r\n", count++);
    _delay_ms(500);
}
```

注: 'printf'関数は文字列雛形内に変数挿入する場所を記すのに代用語を使います。いくつかの可変代用語が右表で示されます。

代用語	説明
%d	符号付き整数を挿入
%s	一続きの文字(文字列)を挿入
%c	文字を挿入
%x	16進数形式で符号なし整数を挿入

他の設定は変更せず、従って上のコード断片では飛ばされています。完全なコード例はGitHubでご覧ください。



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。

## 5. 制御命令受信

USARTの1つの重要な使い方は命令行インターフェースの実装を意味します。この方法で、マイクロコントローラはUSART経由で制御命令を受け取ることができます。命令分離子として行終了子を使うのが便利で、故にこの使用事例についてはUSARTが完全な行を読み取ります。

この使用事例は次のような手順に従います。

- USART周辺機能を最初の使用事例と同じに構成設定します。
- 受信部を許可します。
- 行の最後まで到着データを読んで格納します。
- 受信したデータが有効な命令かを調べて、そうならばそれを実行します。

### 受信部を許可してデータを受信する方法

USART0について、RX用の既定ピン位置はポートAの1番(PA1)ピンです。以下の行はPA1の方向を入力に設定します。

```
PORTA.DIR &= ~PIN1_bm;
```

送信部と同じで、受信部はUSARTn.CTRLBレジスタに書くことによって許可されます。

```
USART0.CTRLB |= USART_RXEN_bm;
```

データを読む前に、使用者は受信完了フラグをポーリングすることによって利用可能なデータを待たなければなりません。

```
uint8_t USART0_read()
{
    while (!(USART0.STATUS & USART_RXCIF_bm))
    {
        ;
    }
    return USART0.RXDATA1;
}
```

### 行を読む方法

以下のコード断片は1行のデータを読んでそれを配列に格納します。有効な行がこの配列長よりも短いと仮定します。

より長い受信行の場合に緩衝部溢れ異常を避けるため、配列が最後に達した時に配列の指標が0にリセットされます。‘\n’ (改行)と‘\r’ (復帰)はこれらが終了子の部分なので無視されます。‘\n’が見つかったら、命令に対して文字列終了(NULL)が追加され、‘executeCommand’関数が命令文字列の値に基づいた関数を呼びます。

```
char command[MAX_COMMAND_LEN];
uint8_t index = 0;
char c;

while (1)
{
    c = USART0_readChar();
    if(c != '\n' && c != '\r')
    {
        command[index++] = c;
        if(index > MAX_COMMAND_LEN)
        {
            index = 0;
        }
    }
    if(c == '\n')
    {
        command[index] = '\0';
        index = 0;
        executeCommand(command);
    }
}
```

以下のGitHubでのコード例ではUSARTが”ON”または”OFF”の命令を受信してマイクロコントローラがGPIO出力を制御します。これは、例えば、LEDを切り替えます。



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。



## 6. 他の実装動作形態

この資料で提示された使用事例は非同期操作でUSARTを使います。USARTが動作可能な他の動作形態があります。どれを使うかは主に実装での判定です。

### 6.1. 同期動作

図6-1. CTRLCレジスタのCMODEビット領域

ビット	7	6	5	4	3	2	1	0
	CMODE1,0		PMODE1,0		SBMODE	CHSIZE2~0		
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	1	1

CTRLCレジスタのCMODEビット領域は通信動作形態を制御します。

非同期動作の欠点は受信部チップと送信部チップが同じホーレートを必要とし、正確なタイミングが必要とされることです。非同期規約はクロック信号に対して分離された線を使い、故にクロックを生成するチップが通信速度を規定します。これは正確なタイミングの面でより一層の柔軟性を持ち、通信での2つの役割、クロックを生成する主装置とクロックを受け取る従装置を作ります。

同期USART動作では、追加のクロックピンのXCKが使われます。RXとTXのピンと同じく、XCDは既定ピンを持ち、PORTMUXレジスタの変更はXCKも変更します。XCKの方向変更はデバイスが主装置(クロックを生成)か、または従装置(クロックを受け取り)かを決めます。

同期動作を達成するには、

- XCK(PA2)ピンの方向を出力として構成設定してください。

```
PORTA.DIR |= PIN2_bm;
```

- USARTn.CTRLCレジスタのCMODEビット領域に0x01を書いてください。

図6-2. USART通信動作形態

値	0 0	0 1	1 0	1 1
名称	ASYNCHRONOUS	SYNCHRONOUS	IRCOM	MSPI
説明	非同期USART	同期USART	赤外線通信	主装置SPI

```
USART0.CTRLC = USART_CMODE_SYNCHRONOUS_gc;
```



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。

### 6.2. 単線動作

1つの線だけを使うことはUSART通信に使われるピン数を効率的に1つに減らします。RXとTXは内部的に接続され、TXピンだけが使われます。これはやって来るデータと出て行くデータの両方が同じ線を共用することを意味し、故に送信と受信は同時に起き得ません。これは半二重通信と呼ばれます。

図6-3. CTRLAレジスタのLBMEビット

ビット	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	DREIE	RXSIE	LBME	ABEIE	RS4851,0	
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

CTRLAレジスタのLBMEビットはRXとTX間の内部送り返し接続を許可するのに使われます。RXとTX間の内部接続はUSARTn.CTRLAに書くことによって作成することができます。

```
USART0.CTRLA |= USART_LBME_bm;
```

これはRXとTXを内部的に接続しますが、TXピンだけが使われます。TXピンが送受信の両方に使われるため、ピン方向は各送信の前には出力として構成設定され、送信の最後の時に入力へ切り替え戻されることが必要です。

送信中にRXが内部的にTXへ接続されるため、送られつつあるデータを受信します。これは衝突検出機構として使うことができます。別に起きている送信がある場合、受信したデータは送信したデータと一致しません。高度な単線ドライバはこの方法を利用することができます。



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。

## 7. 参照

1. ATmega4809ウェブページ : <https://www.microchip.com/wwwproducts/en/ATMEGA4809>
2. megaAVR® 0系手引書 (DS40002015)
3. ATmega3209/4809 – 48ピン megaAVR® 0系データシート (DS40002016)
4. ATmega4809 Xplained Proウェブページ : <https://www.microchip.com/developmenttools/ProductDetails/atmega4809-xpro>

## 8. 改訂履歴

資料改訂	日付	注釈
A	2018年12月	初版資料公開
B	2019年6月	3. “Hello World”章と4. “書式化した文字列送出/printfを使う文字列雛形送出”章のコード例を更新。改訂履歴を追加。その他些細な修正。

---

## 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
- **Microchip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”として保証するとい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、Mchip、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、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。

© 2019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 2019.

本技術概説はMicrochipのTB3216技術概説(DS90003216B-2019年6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
<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">http://www.microchip.com/support</a> ウェブアドレス: <a href="http://www.microchip.com">http://www.microchip.com</a>	<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 - 香港特别行政区</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>オーストリア - ウェルス</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: 45-4450-2828 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 - ガルピング</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 - デルネン</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
<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>ホーストン</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			