

マイクロ波センサー 波形モニター通信仕様
(エラー対策+UARTコマンド対応版)

2017年5月8日

Rev 0.35

セミコンダクター事業部

第六開発部

変更履歴

Revision	日付	内容
0.10	2015/06/16	エラー対策版＋確度・新規作成
0.20	2015/07/14	UARTコマンド追加
0.30	2016/05/25	Type9&10追加
0.31	2016/07/15	UARTコマンドと統合
0.32	2016/08/04	UARTコマンドの記述を見直し
0.33	2016/09/16	UARTコマンドの改行コードによる終端を追記
0.34	2017/03/08	calコマンドの詳細解説追加
0.35	2017/05/08	リリース用(バージョン 0.73系列)に合わせて見直し、部門名更新

UARTデータ送信＋UARTコマンド仕様

マイクロ波センサー（本機）をUART経由でPCと接続し、本機が取り込んだデータ等をPCに送信することが可能である。

また、本機への制御コマンドをUARTコマンドとして受信することが可能である。

UARTの接続条件は以下のとおり。

- ・ 通信速度 : 115200baud
 - ・ Data長 : 8bit
 - ・ パリティ : なし
 - ・ ストップビット : 1bit
 - ・ フロー制御 : なし
-
- ・ 本機からPCへの送信データには、通信エラー対策としてプリアンプル、シーケンス番号、チェックサムを追加する
 - ・ PCから本機へのUARTコマンドでは、通信エラー対策は入れない。（プリアンプル、シーケンス番号、チェックサムなし）

データ形式（送信データ）

- データパケットは、Preamble, Payload, Sequence, Checksumで構成する。

8byte	可変長	1byte	1byte
Preamble	Payload	Sequence	Checksum

- Preamble : プリアンブル
- Payload : 実際のデータをType、Length、Valueでエンコードして送る。
- Sequence: シーケンス番号(0x00 ~ 0x7fの任意の値)
- Checksum: チェックサム(crc32で計算した値のうち、最下位バイト)

Preamble

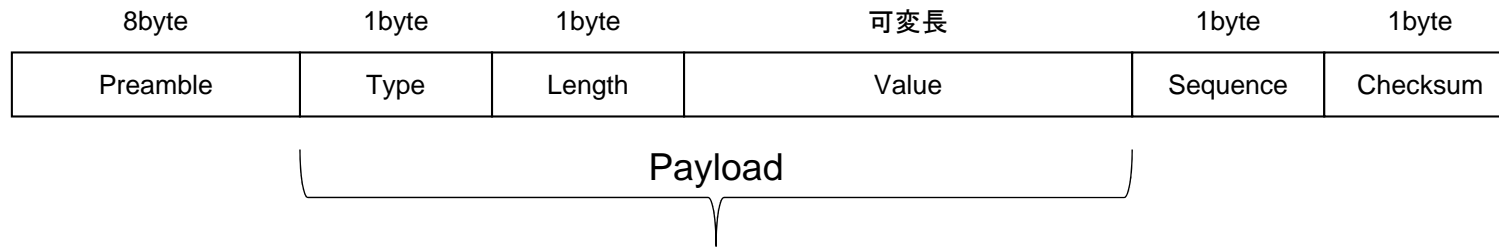
8byte

0x80	0x00	0x80	0x00	0x80	0x00	0x80	0x00
------	------	------	------	------	------	------	------

- Preamble : プリアンブル (Length = 8bytes)
固定値 = 0x80, 0x00, 0x80, 0x00, 0x80, 0x00, 0x80, 0x00

Payload

- 波形データ（心拍、呼吸、体動）、心拍数、呼吸数など複数種のデータを扱うために、Type, Length, Value でエンコードしてデータを送る。



- Type = 1 : 波形（心拍・呼吸・体動）データ（Length = 6bytes）
- Type = 2 : 心拍数+確度（Length = 2bytes）
- Type = 3 : 呼吸数+確度（Length = 2bytes）
- Type = 4 : ACK文字列（Length = 可変長）
- Type = 7 : ACK（dipswコマンド）（Length = 2bytes）
- Type = 10 : BBLレシオ（Body/Breath Ratio）（Length = 2bytes）

上記以外 : Reserved

Type = 1 : 波形データ

Type	Length	Value					
0x01	0x06	心拍(U)	心拍(L)	呼吸(U)	呼吸(L)	体動(U)	体動(L)

- Type = 1 : 波形データ (Length = 6bytes)
心拍、呼吸、体動 の3チャンネル分 (各 2bytes、合計6bytes)を同時に送る。
1チャンネル分のデータは、16bit (signed short) で表現し、0x0がセンター(電圧値で0V)、最大値は、0x7fff、最小値は、0x8000となる。
16bitのデータは、上位8bits(U)、下位8bits(L) の順に送る。

Type = 1の波形データは、100Hzサンプリング(周期10ms)で送信される。

Type = 2, 3 : 心拍数、呼吸数 (+ 確度)

Type	Length	Value	
0x02	0x02	心拍数	確度(心拍数)

Type	Length	Value	
0x03	0x02	呼吸数	確度(呼吸数)

- Type = 2 : 心拍数 + 確度 (Length = 2bytes)
 - Type = 3 : 呼吸数 + 確度 (Length = 2bytes)
- いずれも、unsigned char とする。

確度とは、「値の確からしさの目安となるパラメーター」である。

0,1,2,3を出力し、3が最も確度が高い事を表す。

尚、本確度はあくまでも参考値であり、本確度を用いて、心拍数・呼吸数との演算等は推奨致しません。

Type = 4 : コマンドACK

Type	Length	Value
0x04	0x01 ~ 0xff	デバッグ文字列

- Type = 4 : コマンドACK (Length = 可変)
UARTコマンドを受信した場合に、ACK文字列をType4 として送信する。
Type4 で送信される文字列は、
 OK
 Error
 UARTコマンドACK (バージョン文字列=可変長、DIPスイッチ状態)
 のいずれかである。(ターミネート文字なし)
UARTコマンドversionではバージョン文字列、それ以外は、ACK文字列として、
OKまたはErrorを返す。(dipsw?のみDIP スイッチ状態をType7で返す)
バージョン文字列は可変長とし、送信側(センサー側)では、strlen() にて送信文
字列の長さを取得してLength に設定する。
文字列はASCIIエンコードで、最大長は0xff(255文字)、最小長は1とする。

Type = 7 : ACK (dipswコマンド)

Type	Length	Value	
0x07	0x02	指定値	エラーコード

- Type = 7 : 指定値 + エラーコード (Length = 2bytes)
UARTコマンドdipswを受信した場合に、指定値 + エラーコードを返信する。
指定値、エラーコード、いずれも、unsigned char とする。
エラーコードは、0 : No Error 、1 : Error

コマンドが、

dipsw c

で、正しく受信できて、正常に設定できたときは、指定値 = c、エラーコード = 0

Type7 c 0

をACKとして返す。

dipswコマンドの場合は、Type4のACK文字列は返さない。

Type = 10 : BBレシオ

Type	Length	Value	
0x0a	0x02	体動量(U)	体動量(L)

- Type = 10 (= 0x0a) : BBレシオ (Body/Breath レシオ) (Length = 2bytes)
体動信号を元に、どの程度体が動いたかの目安となる値を計算し、16bit (signed short) で表現する。

BBレシオは、本マイクロ波センサーで新たに導入した概念の値であり、原理的には1.0以上の実数で表現される。体の動きが少なければ、1.0に近づき、体の動きが大きいほど、数字は大きくなる。

元は、実数型の値であるが、1000倍した整数値を使用する。1.0 → 1000、2.0 → 2000に対応させて変換し、上限値を8.0未満で制限をかけることにより、1000 ~ 8000の値を取る。

呼吸数が0のときは、1000に固定する。(対象なしの場合)

Sequence

- Payloadの後ろにSequence(シーケンス番号)をつける。

8byte	1byte	1byte	可変長	1byte	1byte
Preamble	Type	Length	Value	Sequence	Checksum

- Sequence: シーケンス番号(0x00 ~ 0x7fの任意の値)
0x00からスタートし、送信するごとに、1インクリメントし、0x7fを超えたら、0x00に戻る。
- Sequenceは、受信側で波形データの連続性を確認可能とするために、Type1のみでインクリメントし、それ以外(Type2, 3他)では、0固定値とする。
受信側でSequenceが連続していればType1の波形データは欠落しておらず、Sequenceが不連続であればデータが欠落したことを検出可能である。
不連続が発生した場合でも、データを再送しない。

Checksum

- データの妥当性確認のためにChecksumをつける。

8byte	1byte	1byte	可変長	1byte	1byte
Preamble	Type	Length	Value	Sequence	Checksum

- Checksum: チェックサム(crc32で計算した値のうち、最下位バイト)
crc32のアルゴリズムは次ページのとおり。
Valueに対して、crc32を計算し、その結果(unsigned long=4bytes)のうち最下位の1バイト(unsigned char=1byte)をChecksumとして使用する。

次ページの関数(calc_crc32_msb_first())の戻り値を使用する。
なお、その関数はlittle endianのプロセッサを前提としている。

crc32

```
#define CRC_COMPUTE 0x04C11DB7
// calc_crc32_core
// Input
// int len : length of buffer
// unsigned char *buf : pointer of buffer
// Return
// (unsigned long) crc32 value (reverse order, one's compliment)
unsigned long calc_crc32_msb_first(unsigned char *buf, int len)
{
    int i, j;
    unsigned char c;
    unsigned long tmp;
    unsigned long result;
    result = 0xffffffff;
    for(i = 0; i < len; i++)
    {
        c = buf[i];
        for(j = 0; j < 8; j++)
        {
            tmp = ((result >> 31) ^ ((c & 0x80) >> 7)) ? CRC_COMPUTE : 0;
            result = (result << 1) ^ tmp;
            c <<= 1;
        }
    }
    return result;
}
```

UARTコマンド

コマンド書式

コマンド (空白) 引数 (¥n)

の書式による文字列とする。(改行コード: 0x0a=¥nで終端する)

コマンド、引数には英数字(小文字)のASCII文字列を使用する。

コマンド、引数の長さは可変長とするが、コマンド+空白+引数全部合わせて80文字までとする。

引数は1個とする。(2個以上引数を指定した場合の動作は保証されない。)

マイクロ波センサーは、ACK文字列送信(Type4)またはType7でACKを返信する。

コマンド実行の場合は、Type4にて文字列「OK」を送信する。

コマンドエラーの場合は、Type4にて文字列「Error」を送信する。

特殊な例として、versionコマンドでは、Type4にてバージョン文字列を送信する。

また、dipsw?コマンドでは、Type4にてDIPスイッチ状態を送信する。

dipswコマンド(後述)の場合のみ、Type7によるACKを返す。

UARTコマンド

コマンド一覧

- umode : UARTコマンド有効・無効の選択
- version : バージョン表示
- cal : キャリブレーションon/off/start
- dipsw : DIPスイッチ設定
- dipsw? : DIPスイッチ設定内容の表示

UARTコマンド

【注意事項】

本資料では、評価キット(デモ機)をお使い頂いている場合で説明しています。
量産タイプ(UARTモジュール)をお使いの場合は、マイクロ波センサーの入出力ポート(9pin)をご使用頂く事になりますので、以下の読み替えをお願いします。

- ・CALボタン → Pin1
- ・SW1 → Pin4
- ・SW2 → Pin5
- ・SW3 → Pin6
- ・SW4 → Pin7

* 評価キット(デモ機)は、お客様ですぐご評価頂けるようにするため、量産タイプ(UARTモジュール)に以下を追加したものとなっています。

＜CALボタン、SW1～4用Dipスイッチ、USBポート＞

umode (UARTモード切換)

umode com

書式

umode com

UARTコマンドの有効・無効を切り替える。デフォルトはUARTコマンド無効。
umode com 実行後UARTコマンドを受付。umode pin実行後はUARTコマンド無効。

引数

com: UARTコマンドが有効。(UARTモード)

pin: UARTコマンド無効。(デフォルト)

上記以外 : エラー

なお、versionコマンド、dipsw?コマンドは、umode pinの状態でも受け付ける。

umodeコマンドによる動作切換

コマンドumodeにより、本機の動作モードが切り替わる。

- umode pin : スイッチ設定が有効（デフォルト）

umode pinの状態では、ハードウェアのスイッチ設定が有効。（デフォルト状態）

例えば、SW2をONにするとキャリブレーション機能が有効になり、CALボタンを押すとキャリブレーション動作を開始する。

- umode com : UARTコマンドによる設定が有効

UARTコマンドによるスイッチ設定切換が有効。この時は、ハードウェアのスイッチ状態やCALボタンON/OFFは無視される。

例えば、SW2がOFFになっている状態であっても、dipsw 2(24ページ参照)を実行することにより、キャリブレーション機能が有効になり、cal startコマンドによりキャリブレーション動作を開始する。なお、この時は、ボタンをONしても、キャリブレーション動作は開始しない。

version（バージョン表示）

version

書式

version

バージョン表示

引数

なし ソフトウェアのバージョンをType4でエンコードして返信する。

※ UARTコマンド無効の状態でもversionコマンドを受け付ける。

cal（キャリブレーション）

cal on

書式

cal on

しきい値をキャリブレーション結果の値にするか、デフォルト値（固定値）にするか切り替える。

引数

off : デフォルト値を使う。（固定値、デフォルト）

on : キャリブレーションによるしきい値を使う。

start : キャリブレーション動作を実行する。cal on のときだけ有効。

キャリブレーション実行中は、UARTコマンドは受け付けない。（エラーを返す）

上記以外 : エラー

cal（キャリブレーション2）

◆ キャリブレーション実行中の動作

本機はcal startコマンドを受信し、キャリブレーション実行可能と判断した場合は、Type4でOKを返し、キャリブレーション動作を実行する。

キャリブレーション実行時間は、約1分(60秒)である。

キャリブレーション実行中は、UARTコマンドは受け付けず、Type4でErrorと返す。

キャリブレーション動作が完了したことは、UARTコマンドを正常に受け付けるようになったことで確認可能である。例えば、cal on コマンドを実行して、OKと返れば、キャリブレーション動作は完了していると判断できる。

（キャリブレーションの実行には「cal on」の状態になっている必要があり、この状態でさらにcal onコマンドを受け取っても動作上の変化はない）

キャリブレーション実行中、Type2とType3で送信する値は、通常の心拍数・呼吸数とは異なり、内部で使用するパラメータが送信される場合がある。また、Type4についても、内部のパラメータが送信される場合があり、これらの値は無視すること。

cal (キャリブレーション 3)

◆ 実行例

```
> cal on
OK
> cal start
OK
num = 12, H threshold = 0.010025
num = 12, B threshold = 0.008470
<略>
> cal on
Error      ← キャリブレーション実行中なので、Errorが返った
<略>
num = 1, H threshold = 0.012202
num = 1, B threshold = 0.008917
H mean = 0.009892 min = 0.007959 max = 0.012202
B mean = 0.010480 min = 0.008917 max = 0.012472
> cal on
OK      ← キャリブレーションが完了したので、OKが返った
```

dipsw (DIPスイッチ設定)

dipsw c

書式

dipsw 4

DIPスイッチの設定を指定する。SW1 ~ SW4のそれぞれをbit0 ~ bit3に割り当て、エンコードした値を引数とする。 bit4 ~ bit7は0固定とする。

SW1 ON、SW2 OFF、SW3 ON、SW4 OFFの場合は、dipsw 5 とする。

引数

bit0	:	SW1に対応 (ON = 1、OFF = 0)
bit1	:	SW2に対応 (ON = 1、OFF = 0)
bit2	:	SW3に対応 (ON = 1、OFF = 0)
bit3	:	SW4に対応 (ON = 1、OFF = 0)

dipsw (DIPスイッチ設定)

※ 注意事項

umode comを実行し、UARTモードに切り替わった後、dipswコマンドが有効になる。
umode comを実行した時点で、DIPスイッチの状態を読み込み、初期値とする。

dipswコマンドにより、ソフトウェア内部で保持するDIPスイッチ設定値を上書きし、
この設定値に従って動作を継続する。

ハードウェアのスイッチ設定は変更されず、単に無視されるだけになる。

umode pinを実行したら、ハードウェアのスイッチ設定通りに戻る。

dipsw に対するACK

◆ Type7によるACK返信

dipswコマンドについては、マイクロ波センサーにて正しく設定されたことを相互に確認するために、マイクロ波センサーからType7によるACKを返す。

プロトコル

```
dipsw 4  
Type7 4 0
```

Type7では、受け取ったスイッチ設定のパラメータと、エラーコードを返す。

エラーコード

0 : No Error
1 : Error

dipsw? (DIPスイッチ設定内容の表示)

dipsw?

書式

dipsw?

DIPスイッチ設定内容をType4で返信する。

返信する値は1バイトで、SW1 ~ SW4のそれぞれをbit0 ~ bit3に割り当て、bit4 ~ bit7は0固定とする。

引数

なし

※ UARTコマンド無効の状態でもdipsw?コマンドを受け付ける。

dipsw? (DIPスイッチ設定内容の表示)

◆ Type4によるDIPスイッチ設定返信

Type4で返信する値は1バイトで、bit0 ~ bit3はSW1 ~ SW4に対応し、bit4 ~ bit7は0固定とする。

bit0 : SW1に対応 (ON = 1、OFF = 0)

bit1 : SW2に対応 (ON = 1、OFF = 0)

bit2 : SW3に対応 (ON = 1、OFF = 0)

bit3 : SW4に対応 (ON = 1、OFF = 0)

プロトコル (SW3のみON、それ以外はOFFの場合)

dipsw?

dipsw = 0x04 (type4 で文字列をエンコード)