

# SB-I8O8(SensorBlock-Input8,Output8)

I2C 接続用 汎用 I/O デバイス ソフトウェア

## ユーザーマニュアル

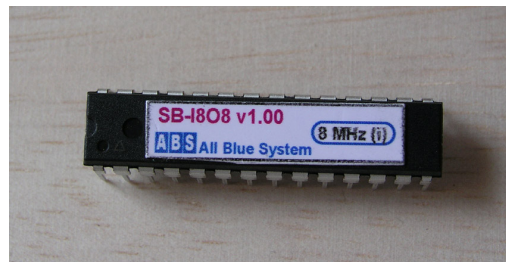
---

ABS-9000 SB-I8O8

SB-I8O8 User Manual Rev A.1.03

2012/10/6

Firmware version: “1.00”



オールブルーシステム (All Blue System)

ウェブページ: [www.allbluesystem.com](http://www.allbluesystem.com)

コンタクト: [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

<b>1</b>	<b>このマニュアルについて</b> .....	<b>3</b>
1.1	著作権および登録商標 .....	3
1.2	連絡先 .....	3
<b>2</b>	<b>使用条件およびライセンス</b> .....	<b>3</b>
<b>3</b>	<b>イントロダクション</b> .....	<b>4</b>
<b>4</b>	<b>SB-I808 動作仕様</b> .....	<b>4</b>
4.1	マイクロコントローラ(Atmega168P).....	4
4.2	SB-I808 ファームウェア書き込み時の Fuse Bits 設定値.....	5
4.3	タイミングと電気特性 .....	5
<b>5</b>	<b>SB-I808 I2Cスレーブコマンドリファレンス</b> .....	<b>6</b>
5.1	自身のスレーブアドレスの設定・取得(0x01) .....	7
5.2	イベント送信先スレーブアドレスの設定・取得(0x02).....	7
5.3	入力ポート(PORTB) のプルアップ設定・取得(0x03).....	7
5.4	入力ポート変化 検出マスク値設定・取得(0x04).....	8
5.5	内部タスクインターバル値設定・取得(0x05) .....	8
5.6	ファームウェアバージョン番号取得(0x0F) .....	9
5.7	ポート出力(PORTD)(0x10).....	9
5.8	ビット出力(PORTD)(0x11).....	9
5.9	ポート入力(PORTB,PORTD)(0x20).....	9
<b>6</b>	<b>イベントリファレンス</b> .....	<b>10</b>
6.1	CHANGE_DETECTイベント(0x01).....	10
<b>7</b>	<b>本製品に使用したソフトウェアライセンス表記</b> .....	<b>11</b>
<b>8</b>	<b>サポートについて</b> .....	<b>14</b>
<b>9</b>	<b>更新履歴</b> .....	<b>14</b>

## 1 このマニュアルについて

---

### 1.1 著作権および登録商標

---

Copyright© 2011 オールブルーシステム

このマニュアルの権利はすべてオールブルーシステムにあります。無断でこのマニュアルの一部を複製、もしくは再利用することを禁じます。

Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

### 1.2 連絡先

---

オールブルーシステム (All Blue System)

ウェブページ <http://www.allbluesystem.com>

メール [contact@allbluesystem.com](mailto:contact@allbluesystem.com)

## 2 使用条件およびライセンス

---

本ソフトウェア(ファームウェア)はオールブルーシステムの ABS-9000 DeviceServer のライセンスを購入されて、その DeviceServer と組み合わせて使用する場合には、複数のプロセッサにインストールして使用することができます。この場合には本ソフトウェアを他の製品に組み込んだり、サポート業務を行うこともできます。

前述の ABS-9000 DeviceServer のライセンスを購入された場合の他に、オールブルーシステムから本ソフトウェアのライセンスを購入された場合に、本ソフトウェア(ファームウェア)の使用ライセンスをお客様に提供いたします。ハードウェアと組み合わせたアプリケーション全体についてのライセンスや、ハードウェアと組み合わせたアプリケーションのサポートは、オールブルーシステムは提供致しません。

本ソフトウェアをオールブルーシステムの ABS-9000 DeviceServer と組み合わせずに単体で使用する場合や、本ソフトウェアライセンスを別途購入していない場合には、個人目的でのみこのソフトウェアを使用することができます。この場合は、業務用途や商用目的で本ソフトウェアを使用したり、他の製品に組み合わせて使用したり、サポート業務をおこなうことはできません。

本ソフトウェアはハイリスクな目的に使用することはできません。ハイリスクな目的とは、原子力、航空、直接的または間接的に人体に死傷を及ぼす可能性のある装置等に使用することを指します。オールブルーシステムは、本ソフトウェアにエラー、バグ等の不具合がないこと、若しくは中断なく稼動すること又は本ソフトウェアの使用がお客様及び第三者に損害を与えないことを保証しません。この事に同意していただけない場合は使用することはできません。

### 3 イン트로ダクション

---

センサーブロック SB-1808 ファームウェア(以降“SB-1808”と略す)の機能について説明します。

SB-1808 は、Atmel Atmega168P<sup>1</sup> マイクロプロセッサ用のファームウェアです。主に、I2C スレーブデバイスとして動作して、I2C マスターデバイスから指定されたデータをポートに出力したり、マスターデバイスから現在のポート値を取得することができます。

SB-1808 は全ての操作を I2C バスを経由したコマンドバイトデータを受信することで実行します。また、SB-1808 でイベント検出を行う様に検出マスクを設定した場合には、SB-1808 が I2C マスターデバイスとして、イベントデータを TDCP デバイスの I2C スレーブデバイスに送信することができます。

SB-1808 の主な機能には下記のものがあります

- I2C マスターデバイスから指定した値を I/O ポートに出力することができます。
- I/O ポート入力値を I2C マスターデバイスから取得することができます。
- I/O ポート入力変化時にイベントデータを I2C スレーブデバイス(TDCP デバイス)に送信することができます。
- 各種設定値をマイクロプロセッサ内の EEPROM に保存、リセット時に自動的に設定値をロードすることができます。

### 4 SB-1808 動作仕様

---

SB-1808 デバイスのファームウェアは下記の動作環境で機能するように設計されています。

- CPU Atmel社製 Atmega168P マイクロコントローラ
- CPU クロック 内部クロック 8MHz
- 電源 5Vまたは3.3V

#### 4.1 マイクロコントローラ(Atmega168P)

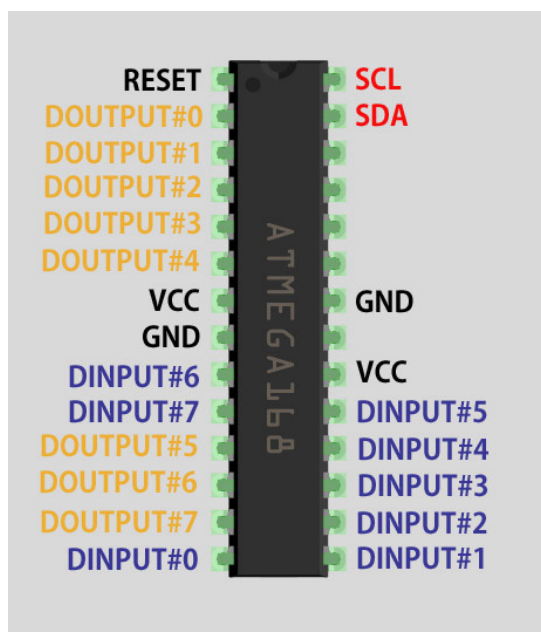
---

	設定値
CPU クロック	8MHz 内部クロック
電源電圧	5V または 3.3V
Flash memory	ファームウェア格納用
EEPROM	コンフィギュレーション保存用

<sup>1</sup> Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

PORTB	bit 0-7	入力ポートとして使用
PORTC	bit 0	リセット直後のSB-1808 起動時に LOW しておくことで、プロセッサ内部のEEPROM コンフィギュレーションデータを初期化することができます。
	bit 1-3	未使用
	bit 4	I2C バスの SDA ピンとして使用（内部プルアップは無効に設定されています）
	bit 5	I2C バスの SCL ピンとして使用（内部プルアップは無効に設定されています）
PORTD	bit 0-7	出力ポートとして使用

ピン配置図



## 4.2 SB-1808 ファームウェア書き込み時の Fuse Bits 設定値

Fuse Bits	設定値
Low Byte	0b11100010
High Byte (*1)	0b11011100 電源電圧 5V で使用する場合
	0b11011101 電源電圧 3.3V で使用する場合
Extended Byte	0b001

(\*1) EEPROM 動作に必要な電源電圧を保障するために、必ず Extended Byte の“BODLEVEL2:0”を上記の値にセットして下さい。

## 4.3 タイミングと電気特性

SB-1808 の電氣的な特性は、ファームウェアを搭載するマイクロプロセッサ Atmega168p の仕様に依存しています。

詳しくは、Atmel Atmega168P<sup>2</sup> マイクロプロセッサのドキュメントを参照してください。

I2C 通信時の、SB-1808 デバイスからイベント送信を行う場合にI2C マスタとして動作するときの I2Cバスのスピードは 100KHz になります。

## 5 SB-1808 I2C スレーブコマンドリファレンス

SB-1808 の I2C スレーブデバイス機能は、I2C マスターデバイスからのコマンドデータを受信してコマンド処理を行った後、コマンドの種類によってはリターン値をマスターデバイスに送信します。

SB-1808 の I2C スレーブデバイスは必ず最初にマスターデバイスから1 byte のコマンドデータと必要に応じてコマンドパラメータを受信します。コマンドデータとパラメータを合わせて最大 32 バイト分のデータをSB-1808 の I2C スレーブデバイスに送信することができます。

```
<command>, <param#0>, <param#1> ... , <param#30>
```

送信されたコマンドデータごとに予め決められた処理を SB-1808 内で実行して、必要に応じてI2C マスターデバイスに最大 32 バイト分のリターンデータを返します。

```
<return#0>, <return#1>, <return#2> ... <return#31>
```

コマンドデータ送信後にリターンデータをマスターデバイスが受信する場合には、I2C バスを “Master Transmitter” から “Master Receiver” に切り替えて操作して下さい。このとき、先に送信済みのコマンドデータの実行結果をリターン値として取得することができます。

I2C バス上にある1つのSB-1808 スレーブデバイスに対して、複数のマスターデバイスがアクセスする場合には、それぞれのマスターデバイスは、“Repeated start mode” を使用して上記のモード切り替え操作を行って、コマンド送信からリターン値取得まで、同一のI2C マスターデバイスが I2C バスを占有するようにして下さい。

SB-1808 のコマンド処理は、同一のコマンドバイト値でもコマンドデータ（とコマンドパラメータ）送信後に受信操作を行う場合と、受信操作を行わない場合では動作が異なります。この区別は、最初に I2C マスターから送信されるコマンドデータとコマンドパラメータの内容で判断しています。

次項から、各スレーブコマンドの説明を行います。

<sup>2</sup> Atmel®, logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries.

## 5.1 自身のスレーブアドレスの設定・取得(0x01)

---

SB-1808 デバイス自身の現在のスレーブアドレスを取得します。

初期設定値は、0x31 です。

(write) 0x01	(read) <slave_addr> 1 byte
--------------	----------------------------

SB-1808デバイス自身のスレーブアドレスを設定します。

(write) 0x01	(write) <slave_addr> 1 byte
--------------	-----------------------------

スレーブアドレス設定直後に、新しいスレーブアドレスで SB-1808 デバイスが動作します。

ここで設定したスレーブアドレスは SB-1808 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

---

### I2C スレーブアドレスの初期化

---

I2C スレーブアドレスは上記のコマンドで任意の値に設定できますが、間違えて意図しないアドレスに設定された場合に、設定値を初期化することができます。リセット直後のファームウェア起動時に PORTC Bit#0 をLOW にしておくことで、プロセッサ内部のEEPROM コンフィギュレーションデータが全て初期化されます。これによって、スレーブアドレスが初期設定値に戻ります。

---

## 5.2 イベント送信先スレーブアドレスの設定・取得(0x02)

---

SB-1808 デバイスがイベント情報を、TDCP デバイスに送信するときのスレーブアドレスを取得します。

初期設定値は、0x30 です。

(write) 0x02	(read) <TDCP_slave_addr> 1 byte
--------------	---------------------------------

SB-1808 デバイスがイベント情報を、TDCP デバイスに送信するときのスレーブアドレスを設定します。

(write) 0x02	(write) <TDCP_slave_addr> 1 byte
--------------	----------------------------------

スレーブアドレス設定直後に、新しいスレーブアドレスで SB-1808 デバイスが動作します。

ここで設定したTDCP スレーブアドレスは SB-1808 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.3 入力ポート(PORTB) のプルアップ設定・取得(0x03)

---

入力ポートの内部プルアップ設定値を取得します。対応するビット位置の値が 1 の場合に内部プルアップが有効に

なります。値が 0 の場合には内部プルアップは無効です。

初期設定値は、0xFF です。

(write) 0x03	(read) <pullup> 1 byte
--------------	------------------------

入力ポートの内部プルアップ設定値を設定します。

(write) 0x03	(write) <pullup> 1 byte
--------------	-------------------------

プルアップ設定変更直後に、新しいプルアップが有効になります。

ここで設定したプルアップ設定は SB-1808 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.4 入力ポート変化 検出マスク値設定・取得(0x04)

入力ポートの値が変化した場合に、SB-1808 デバイスはイベントを送信します。この時のポート変化を検出する範囲のマスク値を取得します。対応するビット位置の値が 1 の場合にそのビットの変化を監視します。値が 0 の場合には監視しません。

初期設定値は、0x00 です。

(write) 0x04	(read) <change_detect> 1 byte
--------------	-------------------------------

入力ポートの内部プルアップ設定値を設定します。

(write) 0x04	(write) <change_detect> 1 byte
--------------	--------------------------------

マスク設定変更直後に、新しいマスク値が有効になります。

ここで設定した変化マスク設定は SB-1808 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.5 内部タスクインターバル値設定・取得(0x05)

デバイス内で実行されているタスク(ポート監視、イベント条件検出)等の実行間隔を取得します。

設定値の単位は ms で通常はデフォルト値 10ms から変更する必要はありません。

初期設定値は、0x0A (10ms) です。

(write) 0x05	(read) <interval> 1 byte
--------------	--------------------------

インターバル値を設定します。(単位は ms)



インターバル値を少なくすると、ポート値の変化等を早く検出することが出来ますが、入力のチャタリング等の影響を受けやすくなりますので注意してください。

(write) 0x05	(write) <interval> 1 byte
--------------	---------------------------

インターバル値は変更直後に有効になります。

ここで設定したインターバル設定は SB-1808 デバイス内部の EEPROM に保存されます。そのため、デバイスリセット後も最後に設定した値が使用されます。

## 5.6 ファームウェアバージョン番号取得(0x0F)

---

ファームウェアバージョン番号を取得します。

(write) 0x0F	(read) <version> 1 byte
--------------	-------------------------

<version> は 0x64 から 0xFF までの値で、整数を実数に変換して100 で割った値がバージョン番号となります。

例: <version> の値が 0x65 の場合には ver 1.01

## 5.7 ポート出力(PORTD)(0x10)

---

出力ポート(PORTD) に<port\_val> で指定したバイトデータを出力します。

出力ポート初期値は、0x00 です。

(write) 0x10	(write) <port_val> 1 byte
--------------	---------------------------

## 5.8 ビット出力(PORTD)(0x11)

---

出力ポート(PORTD) の <bit\_number> で指定したビット位置のデータを <bit\_value> で指定した値に更新します。

(write) 0x11	(write) <bit_number> 1 byte	(write) <bit_value> 1 byte
--------------	-----------------------------	----------------------------

<bit\_number> 0 から 7 までの値

<bit\_value> 0 または 1

## 5.9 ポート入力(PORTB,PORTD)(0x20)

---

現在の入力ポート(PORTB) の値と出力ポート(PORTD) の値を取得します。

(write) 0x20	(read) <port_B_val> 1 byte	(read) <port_D_val> 1 byte
--------------	----------------------------	----------------------------

## 6 イベントリファレンス

---

SB-1808 では、I/O ポート値が変化した場合にイベントデータを I2C マスターデバイスに送信する機能があります。これによって、監視システムに応用する時のサーバーとリモートデバイス間のポーリング動作を最低限にすることで、データ通信量を減らすことができます。

イベントデータは、SB-1808 デバイスが I2C マスターデバイスとして動作させて、I2C スレーブデバイス (TDCP デバイス) に対して下記のようなデータを送信します。

- I2C 送信データ (SB-1808 デバイス → I2C イベント送信先アドレスに指定したスレーブデバイス)

```
<TDCP_slave_command><event_data>
```

<TDCP\_slave\_command> には、0x11 が入ります。(1 byte)

これは、I2C スレーブデバイスとして動作している TDCP デバイスが、I2C マスターデバイスからイベントデータを受信して、このイベントデータを XBee 経由でサーバーへ送信させるためのリクエストコマンドです。

詳細は“TDCP ユーザーマニュアル”を参照してください。

<event\_data> イベント毎に決められたイベントデータです。以降で SB-1808 のイベント毎に予め決められているイベントデータの詳細を説明します。

---

### 注意

SB-1808 内部では、内部タスクインターバルコマンド (0x05) で設定した間隔 (default: 10ms) でイベント発生条件をチェックして、条件に一致した場合にイベント送信先スレーブアドレスコマンド (0x02) で設定された TDCP デバイスにイベントデータを送信します。同一 インターバル内で複数のイベントが発生する条件になっている場合には、後のイベントデータ発生条件のチェックを、次の インターバルまで遅延させて実行します。

---

### 6.1 CHANGE\_DETECT イベント (0x01)

---

- イベント発生条件

入力ポート (PORTB) の値が変化した場合に発生します。

監視対象のポートのビット位置は、入力ポート変化検出マスク値コマンド (0x04) で設定します。

- イベントデータフォーマット

```
<slave_addr><device_type><event_type><diff_bits><dio>
```

<slave\_addr> には SB-1808 デバイス自身のスレーブアドレスが設定されます。(1 byte)

<device\_type> には SB-1808 デバイスであることを示す 0x01 が設定されます。(1 byte)

<event\_type> には “CHANGE\_DETECT” イベントであることを示す 0x01 が設定されます。(1 byte)

<diff\_bits> には、変化したビットを 1、変化していないビットを 0にした値が設定されます。(1 byte)

<dio> には 現在の入力ポート値(PORTB)が設定されます。(1 byte)

- **備考**

イベントの発生条件になる毎に、このイベントが発生してイベントデータが送信されます。

ポート値は、内部タスクインターバルで設定した間隔(default: 10ms) で値を常に取得して、データ値が連続して同一の値を示した時に新しいポート値として内部に保存しています。この保存するポート値が前回保存していたポート値と比べて、変化していた場合にこのイベントが発生します。

## 7 本製品に使用したソフトウェアライセンス表記

---

### avr-libc License

avr-libc can be freely used and redistributed, provided the following license conditions are met.

Portions of avr-libc are Copyright (c) 1999-2008

Werner Boellmann,

Dean Camera,

Pieter Conradie,

Brian Dean,

Keith Gudger,

Wouter van Gulik,

Bjoern Haase,

Steinar Haugen,

Peter Jansen,

Reinhard Jessich,

Magnus Johansson,

Harald Kipp,

Carlos Lamas,

Cliff Lawson,

Artur Lipowski,

Marek Michalkiewicz,

Todd C. Miller,

Rich Neswold,

Colin O'Flynn,

Bob Paddock,

Andrey Pashchenko,  
Reiner Patommel,  
Florin-Viorel Petrov,  
Alexander Popov,  
Michael Rickman,  
Theodore A. Roth,  
Juergen Schilling,  
Philip Soeberg,  
Anatoly Sokolov,  
Nils Kristian Strom,  
Michael Stumpf,  
Stefan Swanepoel,  
Helmut Wallner,  
Eric B. Weddington,  
Joerg Wunsch,  
Dmitry Xmelkov,  
Atmel Corporation,  
egnite Software GmbH,

The Regents of the University of California.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 8 サポートについて

---

SB-1808 製品のファームウェアをオールブルーシステムの DeviceServer と組み合わせて使用する場合には、ABS-9000 DeviceServer のライセンスのサポート期間内において、メールにてサポートを行います。

オールブルーシステムの SB-1808 はファームウェアとそのソフト的な不具合に対してのみサポート致します。お客様のハードウェアへの組み込みに関する内容や、マイクロコントローラ自身に関しましてはサポートできません。本ソフトウェアをオールブルーシステムの DeviceServer と組み合わせずに、使用する場合はサポートは行いません。

ABS-9000 DeviceServer のスクリプトライブラリ関数(Luaのライブラリ)とEXCEL VBA から利用するための API ライブラリ関数(XASDLCMD.DLL)を利用する場合の、プログラミング方法やサンプルプログラムのソースコードに関する解説など、お客様のソフトウェア開発自身に関するサポートにつきましては、別途有償対応となります、詳しくはお問い合わせください。

動作環境を満たしている場合でも、お客様の環境やハードウェアによっては正常に動作しない場合があります。これらを含めて、こちらで再現できない問題については十分なサポートができない場合があります。ABS-9000 DeviceServer のライセンスをご購入になる前に、目的の機能がお客様の環境で動作することを、事前にデモライセンスをご利用になって、充分確認されることをお勧めします。

## 9 更新履歴

---

REV A. 1. 03 2012/10/6

ピン配置図 AVcc (VCC) 追加

REV A. 1. 02 2011/12/5

I2C スレーブアドレス初期化の記述追加

イベントデータにデータサイズの記述を追加

REV A. 1. 01 2011/11/2

ピン配置図追加

REV A. 1. 00 2011/6/22

初版作成