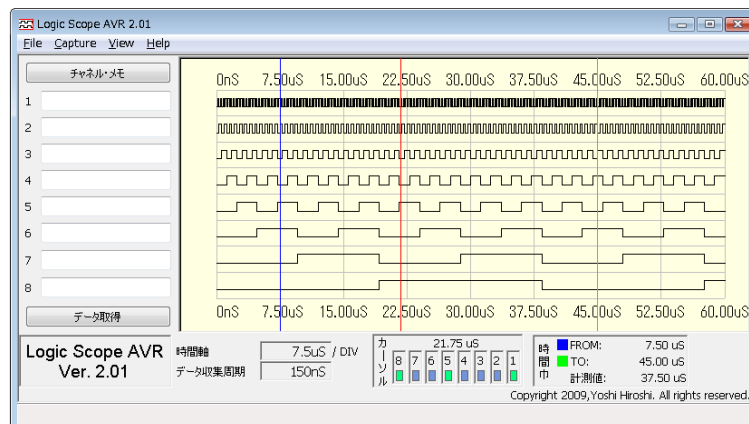
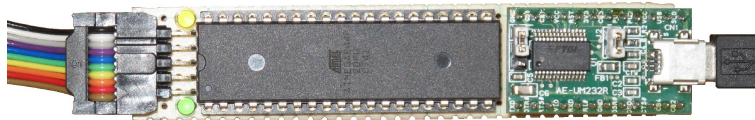


# 簡単ロジック・スコープ ユーザーズ・ガイド

改定第1版



2009年9月

よし ひろし

## 謝辞

簡単ロジック・スコープのパソコン側アプリケーションならびにファームウェアを開発するにあたり、次のソフトウェアを活用させていただきました。  
開発にかかわられた多くの方々に、この場を借りて御礼申し上げます。

wxWidgets  
Code:Blocks  
wxSmith  
Borland Free Command Line Tools  
WinAVR と関係するソフトウェア群

## 目次

1.はじめに.....	4
2.画面概要.....	5
2.1.表示.....	6
2.1.1.カーソル・ラインを表示する.....	6
2.1.2.カーソル位置のデータ値を表示する.....	7
2.1.3.時間巾を計測・表示する.....	7
2.1.4.目盛.....	7
2.1.5.画面サイズ.....	7
2.1.6.画面サイズの横方向最大化.....	7
2.2.ズーム.....	8
2.3.スクロール.....	8
2.4.時間巾の計測・表示.....	9
2.4.1.マウスの左・右ボタンで開始・終了を設定.....	9
2.4.2.マウスの右ボタン・ドラッグで開始・終了を設定.....	9
3.データ取得.....	10
3.1.通信設定.....	10
3.2.サンプリング周期設定.....	11
3.3.データ収集設定.....	11
3.4.トリガー設定.....	11
3.5.設定パターン.....	11
4.ハードウェア.....	12
4.1.回路図.....	12
4.1.1.ATmega644P 版.....	12
4.1.2.ATmega128 版.....	12
4.2.LED.....	13
4.2.1.緑色 LED.....	13
4.2.2.黄色 LED.....	13
4.3.ボーレート設定.....	13
5.通信プロトコル.....	14
5.1.通信メッセージの書式.....	14
5.2.コマンド詳細.....	14

# 1. はじめに

この簡単ロジック・スコープは、ハードウェアの超初心者のためのツールとして開発しました。ワンチップ・マイコンなどのファームウェアを開発しているときに、特別な計測器などを持たない初心者にとって、ファームウェアの動作が正しく行われているかを確認することは、かなり困難です。このようなときに、この簡単ロジック・スコープを用いて、ワンチップ・マイコンのピンにプローブをあてて、パソコンで観測することにより、大まかな動作の確認を行います。

簡単ロジック・スコープでは、8ビット・マイコンとして定評のあるアトメル社の AVR ファミリーの一員である ATmega644P / ATmega128 を使用しました。

AVR マイコンの場合、IO ポートの特定のビットをオン/オフを行うには、少なくとも 4 クロックが必要です。一般的には判定やフロー制御などが入りますので、もっと低速で動作します。

この簡単ロジック・スコープでは ATmega644P 使用時には最短 3 クロック、ATmega128 使用時には最短 4 クロックでデータ・サンプリングを行いますので、AVR ファミリーを使用するファームウェアでは、ほとんどの動作検証が可能になると考えています。

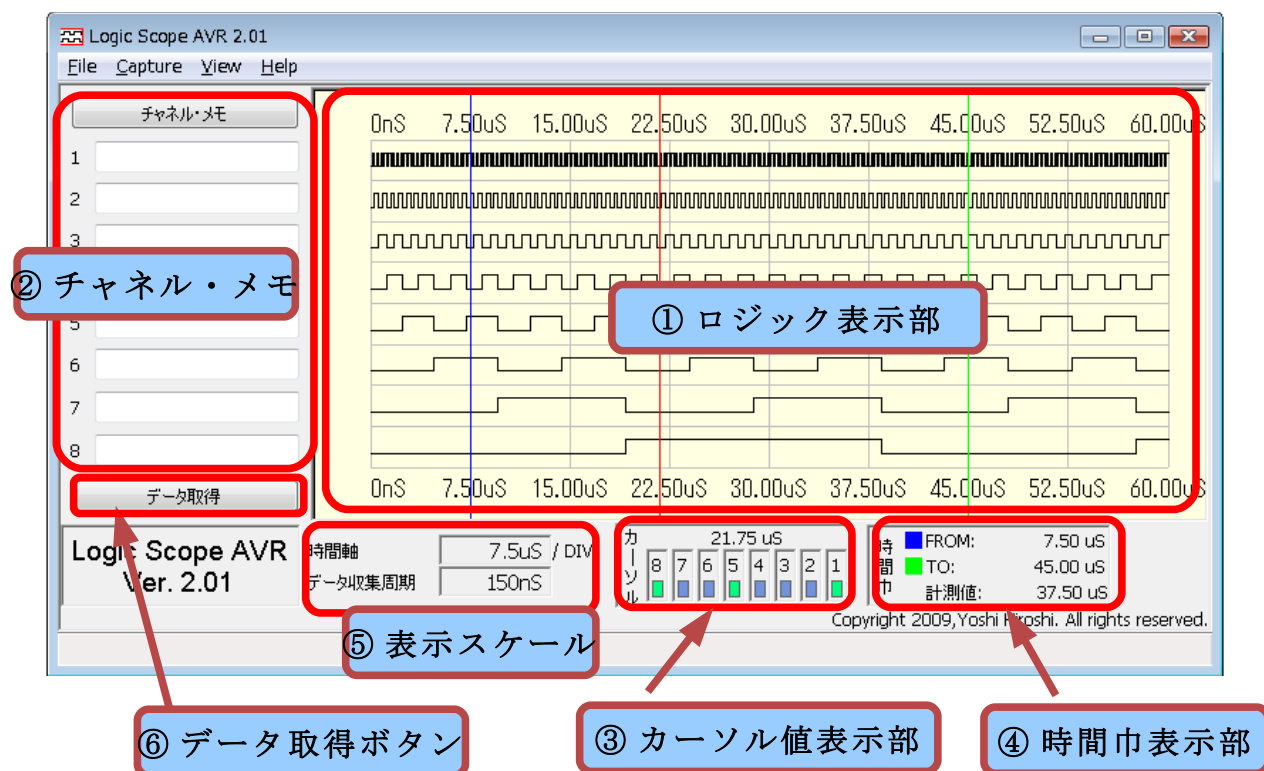
残念ながら低価格ワンチップ・マイコンでは、内蔵 SRAM がそれ程大きくないので、記録メモリが大きく取れません。ご容赦ください。

項目	仕様
チャンネル数	8 チャンネル
サンプリング周期	150nS, 200nS, (ATmega644P だけ) 250nS, (ATmega128 だけ) 500nS, 1uS, 2uS, 5uS, 10uS, 20uS, 50uS, 100uS, 200uS, 500uS, 1mS
記録メモリ	3840 ステップ(ATmega644P) / 30720 ステップ(ATmega128)
入力レベル	本体のみで、 VIH=3.0V、VIL=0.8V レベルシフター使用時 (74HCT544) VIH=2.0V、VIL=0.8V
トリガー	8 チャンネルまでの値一致 サンプリング周期：500nS これより短いパルスは検出できないことがある。

## 2. 画面概要

アプリケーション起動後の画面を次に示します。

ファームウェアが無くて機能概要を把握できるように初期状態では、サンプル・データが表示されています。

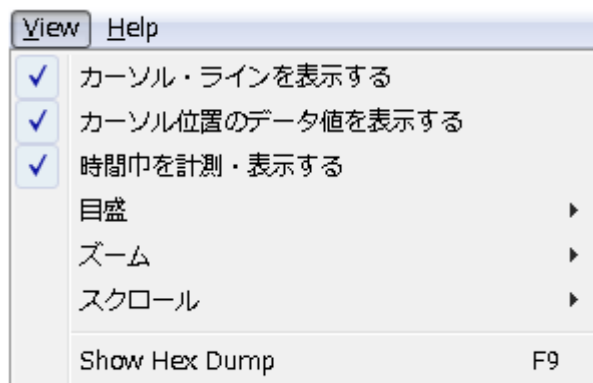


項番	項目	説明
①	ロジック表示部	この領域に、計測した信号を表示します。この領域にフォーカスが当たっているとき、マウス操作あるいはキー操作で表示時間巾や表示位置の変更ができます。 赤ライン：現在カーソル位置 青ライン：時間巾計測開始位置、 緑ライン：時間巾計測終了位置
②	チャンネル・メモ	データ取得時に、各信号線の意味を記録する目的でチャンネルごとにメモをつけられます。ここに、その内容が表示されます。 上部にあるチャンネル・メモ・ボタンをクリックして変更できます。
③	カーソル値表示部	カーソル位置のロジック値を表示します。
④	時間巾表示部	カーソル間の時間間隔を計測して、表示します。
⑤	表示スケール	表示している時間のスケールを示します。 ズーム機能により時間軸を帰ることができます。データ収集周期は、データ取得時の値です。
⑥	データ取得ボタン	ファームウェアからデータを取得するときにクリックします。

## 2.1. 表示

好みに合わせて表示する項目の設定を行います。

View メニューからの設定になります。

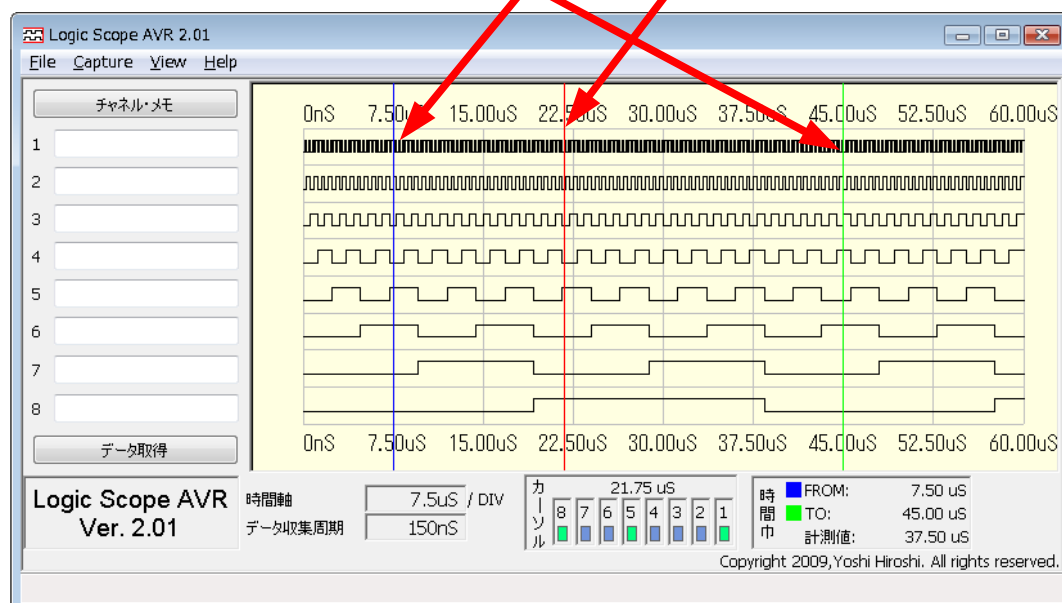


### 2.1.1. カーソル・ラインを表示する

チェックすると、マウス・カーソルがロジック表示部に入ったとき、自動的にカーソル・ラインを表示します。

チェックが無いときは、マウスのボタン操作時だけ表示されます。

赤ライン：現在カーソル位置  
青ライン：時間巾計測用開始位置  
緑ライン：時間巾計測用終了位置



### 2.1.2. カーソル位置のデータ値を表示する

チェックすると、マウス・カーソルがロジック表示部に入ったとき、カーソル位置のデータ値をカーソル値表示部に表示します。

カーソルの移動にあわせて自動で更新されます。

### 2.1.3. 時間巾を計測・表示する

チェックすると、時間巾表示部を表示し続けます。

チェックがない場合には、マウス・ボタン操作時のみ時間巾表示部を表示します。

### 2.1.4. 目盛

目盛は、ロジック表示部に表示される DIV 数を変更します。

デフォルトでは 8DIV ですが、最大 20DIV まで表示可能です。

DIV あたりの表示データ数は変わらないので、目盛を増やすと一度に表示できるデータ数、すなわち、時間巾を増やせます。

ただし、極端にデータ数と画面サイズが大きくなると、動作が遅くなります。

<input checked="" type="radio"/>	8 DIVs	Ctrl+8
<input type="radio"/>	10 DIVs	Ctrl+0
<input type="radio"/>	13 DIVs	Ctrl+3
<input type="radio"/>	16 DIVs	Ctrl+6
<input type="radio"/>	20 DIVs	Ctrl+2

### 2.1.5. 画面サイズ

起動時の画面サイズは、本アプリケーションが表示できる一番小さいサイズの画面サイズです。他の多くのアプリケーション同様の方法で、画面サイズを任意の大きさに変更できます。

### 2.1.6. 画面サイズの横方向最大化

画面下部の、アプリケーションタイトルや、コピーライト表示部分などをダブルクリックすると、横方向に最大化します。

## 2.2. ズーム

5 サンプル/DIV～5000 サンプル/DIV まで変更できます。

動作	キー操作	マウス操作
ズーム・イン	↑ 矢印	ホイール 前回し
ズーム・アウト	↓ 矢印	ホイール 後ろ回し

## 2.3. スクロール

画面一面に収集したデータが収まりきらないときは、スクロール操作で表示範囲を移動することができます。

動作	操作	表示
マウス操作	左ボタンによるドラッグ	ドラッグ分だけ移動
キー操作	→ 矢印 または ← 矢印	2 DIV 移動
	→/← + Shift キー (同時押し)	4 DIV 移動
	→/← + Ctrl キー (同時押し)	1 DIV 移動
	→/← + Shift + Ctrl (同時押し)	1 データ移動



## 2.4. 時間巾の計測・表示

カーソルで開始・終了の範囲を指定して時間巾を計測・表示します。

範囲指定には次の2つの方法があります。

- (1) マウスの左・右ボタンで設定
- (2) マウスの右ボタンでドラッグ

### 2.4.1.           マウスの左・右ボタンで開始・終了を設定

コントロール・キーを押下しながら、マウスボタンを操作します。

機能	操作
開始点の設定	コントロールキーを押しながら、マウスの左ボタンをクリック。 このときドラッグした場合、開始点が変更になります。
終了点の設定	コントロールキーを押しながら、マウスの右ボタンをクリック。 このときドラッグした場合、終了点が変更になります。

### 2.4.2.           マウスの右ボタン・ドラッグで開始・終了を設定

マウスの右ボタンを押したとき、その場所が開始点になります。

右ボタンを押下したままでドラッグすると、終了点を設定できます。

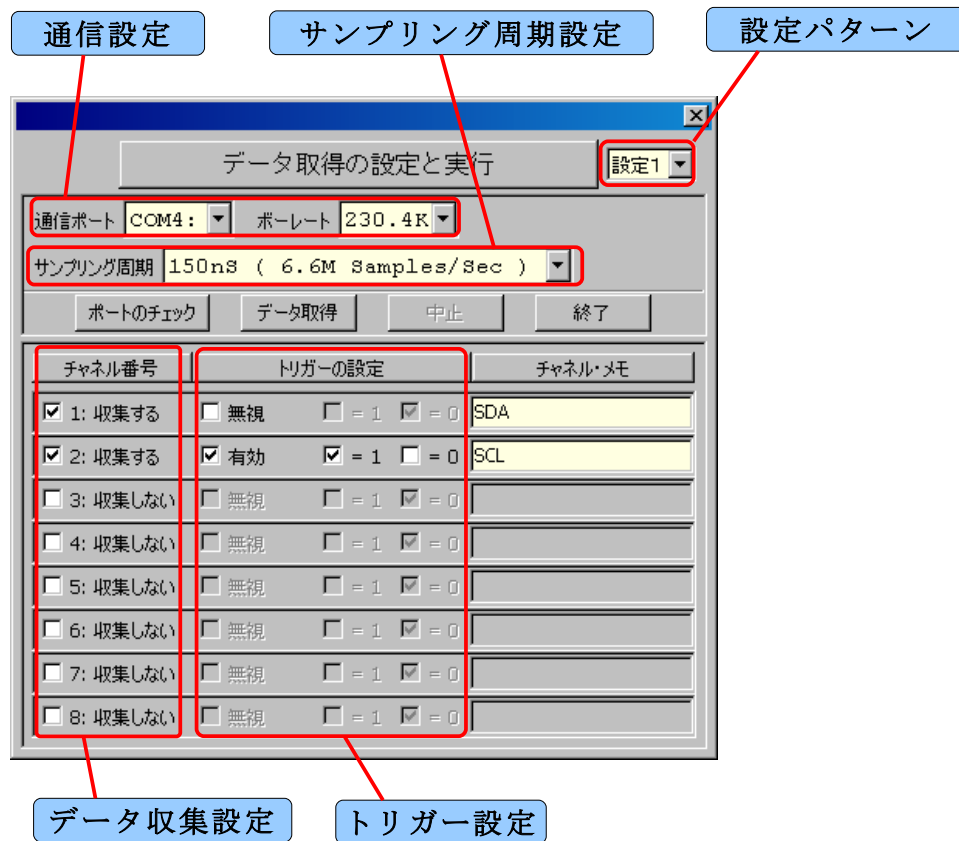
ドラッグにあわせて、カーソルが移動し、時間巾の計測値が更新されます。

マウス右ボタンを離すと、計測値の更新が止まります。

### 3. データ取得

接続されているファームウェアからデータを収集します。

ファームウェアでは、この設定画面でデータ取得ボタンをクリックしたときだけ動作してデータを収集し、パソコンに収集データを転送します。



#### 3.1. 通信設定

通信ポートとボーレートを選択します。

通信ポートは、ユーザー各位の環境によってかなり異なります。

また、場合によってはケーブルを挿す USB ポートの位置によっても変わりますので、必要であればデバイスマネージャを参照してポートを選択してください。

一応 COM24 まで用意していますが、足りないときはご勘弁ください。

設定の確認は、【ポートのチェック】ボタンをクリックすると、ファームウェアと簡単な通信を行い結果を表示します。

なお、データ取得時のみ通信ポートをオープンします。

複数の簡単ロジック・スコープを起動して、異なるタイミングの結果を切替えて表示することもできます。

ATmega644P 使用時のボーレートは 230.4 K です。

ATmega128 使用時のボーレートは 115.2 K です。

### 3.2. サンプリング周期設定

サンプリング周期を 150nS から 1mS の間で選択できます。

最高速以外は、1-2-5 ステップになっています。

データのサンプリング数が 3840 または 30720 固定となっているので、計測時間を長くとり場合には、サンプリング周期を長くします。このとき、時間の分解能が下がりますから、高速に変化するデータを取得することはできません。

サンプリング周期と収集期間のトレードオフになります。

### 3.3. データ収集設定

本ロジック・スコープは 8 チャンネルのデータを収集できますが、もっと少ないデータで十分な場合があります。

このようなときに、不要な信号線にノイズが載ったりすると、画面が見にくくなります。

これをさけるために、各チャンネルごとにデータの収集を許可・禁止する機能があります。

収集するチャンネルは、チャンネル番号のチェック欄にチェックを入れます。

チェック欄にチェックのないチャンネルのデータは収集・表示されません。

### 3.4. トリガー設定

トリガーは、ビットパターンを設定して、当該ビットが設定した値となったときにデータの収集を開始します。

トリガーに使用するチャンネルは、トリガーの設定で【無視】をチェックして【有効】とします。

さらに、そのビットが ON あるいは OFF のどちらのときに収集を開始するかをチェックします。例えば、【=1】をチェックした場合、当該ビットが 0 から 1 に変化したときに収集を開始します。

複数ビットにトリガを設定した場合は、AND 条件、すなわち、すべてのビットが指定した値になったときに収集を開始します。

トリガー条件が不適切で、いつまでも収集を開始しない場合は、【中止】ボタンで収集を中止させてから次の設定を行います。

### 3.5. 設定パターン

設定した値は、8 種類まで記録できます。

設定パターンごとに、データメモリを持っているので、パターンごとに切替えての比較も可能です。

プログラム終了時には、設定パターンを初期化ファイルに格納し、次回起動時にその設定値を利用することが出来ます。

初期化ファイルを削除すると、初期状態に戻ります。

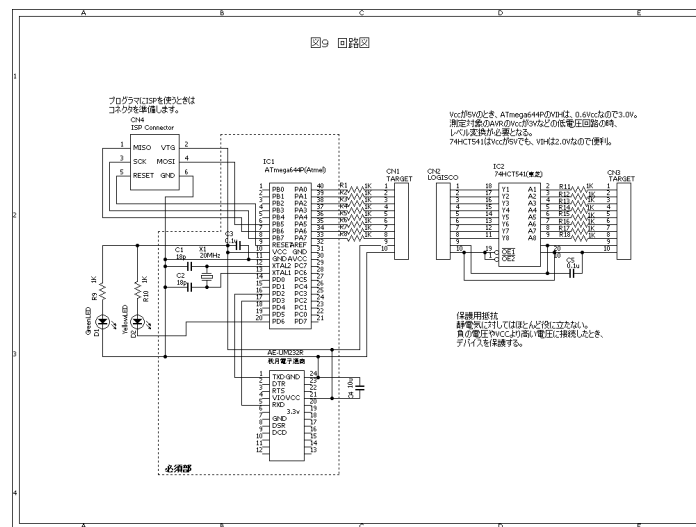
格納場所：プログラムと同じディレクトリ

ファイル名：LogicScope.ini

## 4. ハードウェア

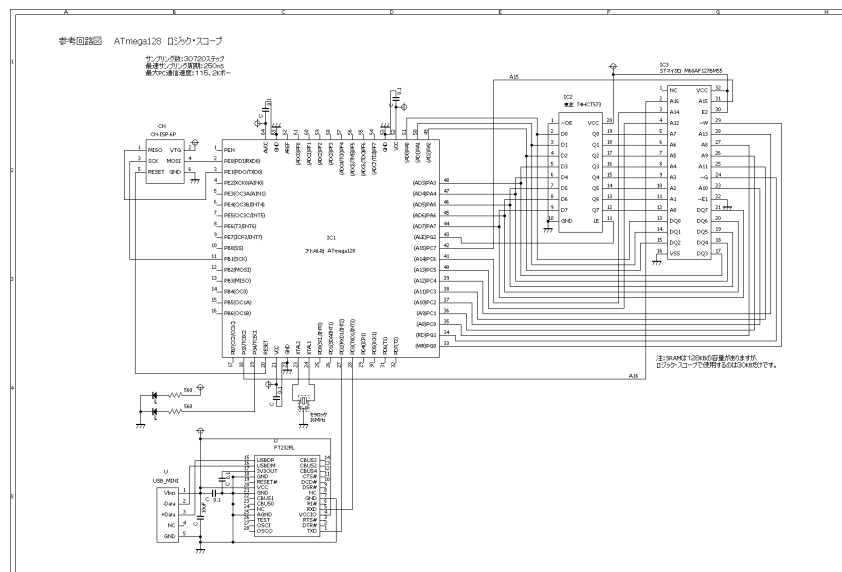
### 4.1. 回路図

### 4.1.1. ATmega644P 版



### 4.1.2. ATmega128 版

参考回路図を示します。ATmega128はフラット・パッケージなので、初心者にはちょっと半田付けが大変です。市販の32KM SRAM搭載マイコンボードを利用する方法もあります。



## 4.2. LED

### 4.2.1. 緑色 LED

電源表示です。

### 4.2.2. 黄色 LED

動作状態を表示します。

項番	状態	意味
1	黄色点滅	電源投入時あるいはリセット時の、初期化处理完了
2	黄色点灯	トリガ待ち
3	黄色中速点滅	データ収集中（注：高速サンプリングでは早すぎて確認できない）
4	黄色高速点滅	チェック・コマンドの受信

## 4.3. ボーレート設定

回路図には描いてありませんが、ポート B の設定でボーレートを変更できます。

項番	PB2	PB1	PB0	ボーレート
1	0	0	0	4. 8 K
2	0	0	1	9. 6 K
3	0	1	0	1 4. 4 K
4	0	1	1	1 9. 2 K
5	1	0	0	3 8. 4 K
6	1	0	1	5 7. 6 K
7	1	1	0	1 1 5. 2 K
8	1	1	1	2 3 0. 4 K (ATmega644P) 1 1 5. 2 K (ATmega128)

各ピンがオープン的时候は、1 です。よって、全ピンがオープン時は、項番 8 の設定となります。

## 5. 通信プロトコル

パソコンと AVR（ファームウェア）をシリアル通信で接続し、ファームウェアではパソコンからの指示に従って動作します。

### 5.1. 通信メッセージの書式

パソコンからのコマンドとそれに対するレスポンスの形式を次に示します。

一般型：

コマンド	データ長	データ
項目	長さ	記事
コマンド	1	ファームウェアに対する指示の種類、あるいは、ファームウェアからの応答の種類を示します。
データ長	2	データ部に含まれるデータの長さを示します。 コマンド部分ならびにデータ兆部分は含みません。
データ	任意	コマンドに必要なパラメータや取得したデータ列などを格納します。

### 5.2. コマンド詳細

メッセージ種類	コマンド	通信データ列（16進数）	備考
チェック	0x20	20000101	
同 レスポンス	0x21	2100064Fmmsssseeee	mm：ビット：マスク(1Byte) ssss：収集データバッファサイズ(2Byte) eeee：拡張メモリサイズ(2Byte)
収集開始	0x22	220006xxxxxxxxmmpp	xxxxxxxx：サンプリング周期(4Byte) mm：トリガー・マスク(1Byte) pp：トリガー・パターン(1Byte)
同 レスポンス	0x23	23000345ssss	データ収集開始 ssss：データ収集数(2Byte)
同 レスポンス	0x23	2300034Fssss	データ収集終了 ssss：データ収集数(2Byte)
収集条件送信要求	0x24	24000101	
同 レスポンス	0x25	250009bbmmppxxxxxxxxss	bb：ビット・マスク(1Byte) mm：トリガー・マスク(1Byte) pp：トリガー・パターン(1Byte) xxxxxxxx：サンプリング周期(4Byte) sss：データ収集数(2Byte)
データ送信要求	0x26	26000501aaaassss	aaaa：送信要求先頭アドレス(2Byte)

			ssss : 送信要求データ数(2Byte)
同 レスポンス	0x27	27llll01aaaassssyyyyy....yy	llll : 記録データ長+5 (2Byte) aaaa : 先頭アドレス(2Byte) ssss : データ数(2Byte) yy...yyy : 記録データ (llll - 5 Byte)
エラー	0xFF	FF00024Exx	xx : エラーコード(1 Byte) F1 : コードエラー (チェック) F2 : 長さエラー (チェック) F3 : サンプリング周期エラー (収集開始) F4 : 長さエラー (収集開始) F5 : コードエラー (データ送信要求) F6 : 長さエラー (データ送信要求) F7 : コードエラー (データ送信要求) F8 : 長さエラー (データ送信要求)

データ送信要求は、1024 バイト単位で行っています。

総データ数が要求データ数にたない場合は、最後の要求ブロックに対するレスポンスでデータ数を要求以下のデータ数にして応答します。

例えば、ファームウェアで収集したデータが 3840 データのとき、1024 データの要求が 4 回送られます。このときは、最後のブロックで、1024 データの要求に対して 768 データの応答を返します。