

# 16 セグメント LED I2C モジュールキット

説明書 Ver.1.1

2013 年 3 月

株式会社アイキューカンパニ

## 特徴

- 高さ 7cm の超特大 16 セグメント LED を使いやすい I2C モジュールにしました。
- マスターとは I2C 信号線 (SCL、SDA) と電源の合計 4 本の接続だけで制御できます。
- デバイスのアドレスはユーザーが設定できます。同一バス上に最大 64 個の本モジュールをデジ  
ーチェーン接続が可能です。
- 本モジュールは全セグメントをスタティック駆動しますのでチラツキがありません。
- PWM 制御により、明るさを制御することができます。
- 表示切り替え時に視覚的効果をもたせたトランジションを設定できます。
- Arduino で制御した場合のサンプルソースが公開されています。
- 本モジュールのファームウェアも公開されています。ユーザー独自の拡張が可能です。

## 部品表

記号	部品名	型番・規格	数量	備考
LED	16セグメントLED	KA2311-42B-UR91	1	LEDTECH
IC1	AVRマイコン	ATMEGA328P-PU	1	ATMEL
IC2	8回路入り3ステートラッチ	74F373	1	
T1	NPNトランジスタ	2SC1815	1	
R1,R2,R4,R5,R7,R10,R11,R13,R14,R16	カーボン抵抗	51Ω 1/6W	10	緑茶黒金
R3,R6,R8,R12,R15,R17	カーボン抵抗	100Ω 1/6w	6	茶黒茶金
R9,R18,R21	カーボン抵抗	1kΩ 1/6W	3	茶黒赤金
R22,R23,R24,R25,R26,R27	カーボン抵抗	10KΩ 1/6W	6	茶黒橙金
SW1	DIPスイッチ	4P	1	
C1,C2,C3	セラミックコンデンサ	0.1uF 50V	3	104
JP1,JP2	ピンヘッダ/ピンソケット	1x4	2	
PG	ピンヘッダ	2x3	1	AVR書込用
	ICソケット	28ピン	1	
	ICソケット	20ピン	1	
	基板	48x90 t=1.6	1	

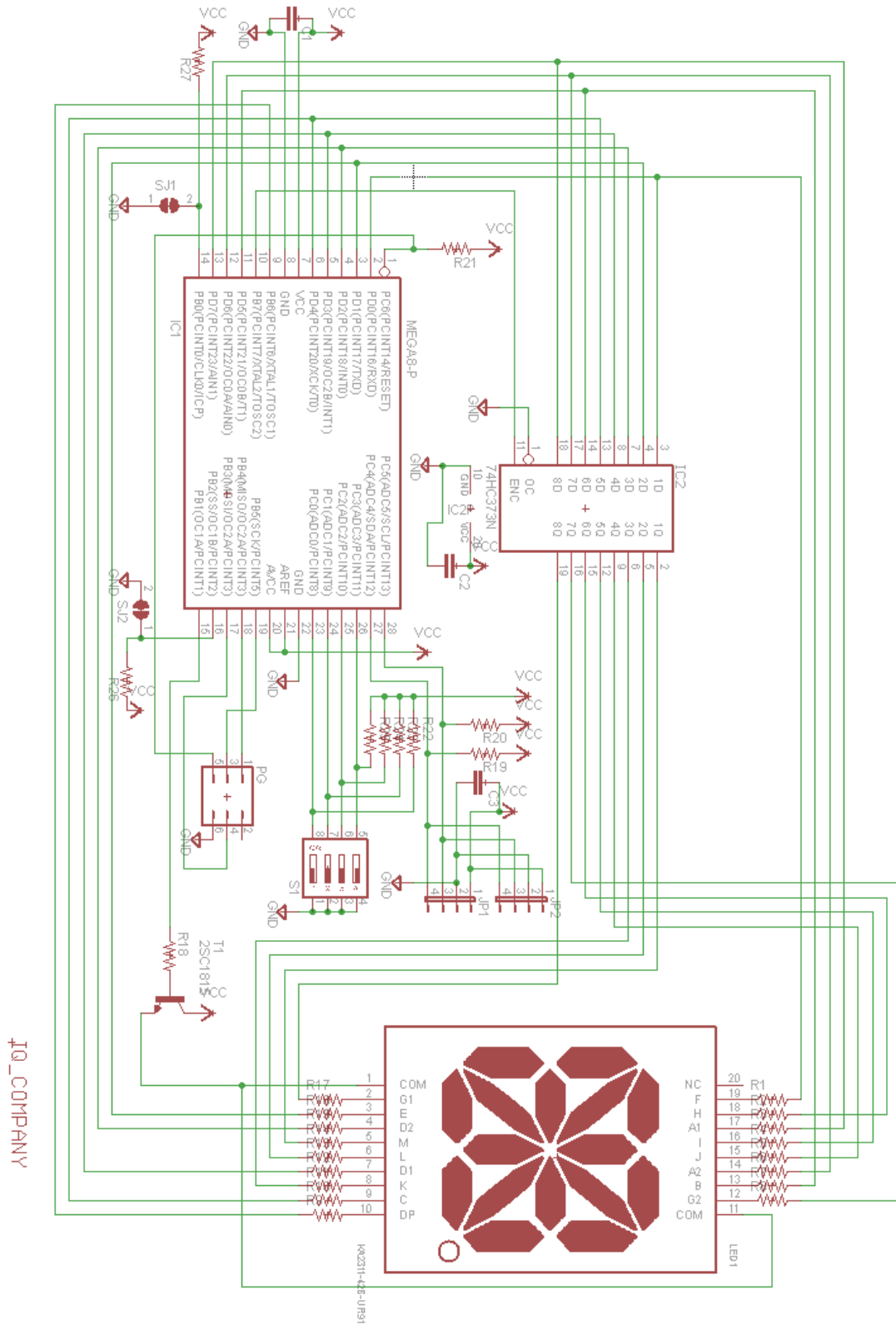
※JP1,JP2 はピンヘッダ/ピンソケットのどちらか好きな方をお使い下さい。

※R19,R20 は I2C のプルアップ用です。通常はマスター側でプルアップするので本モジュールでの実装は不要です。

主要パーツのデータシートは以下ページからダウンロードできます。

<http://iqcompany.jp/products/16-segments-led-module/>

回路図

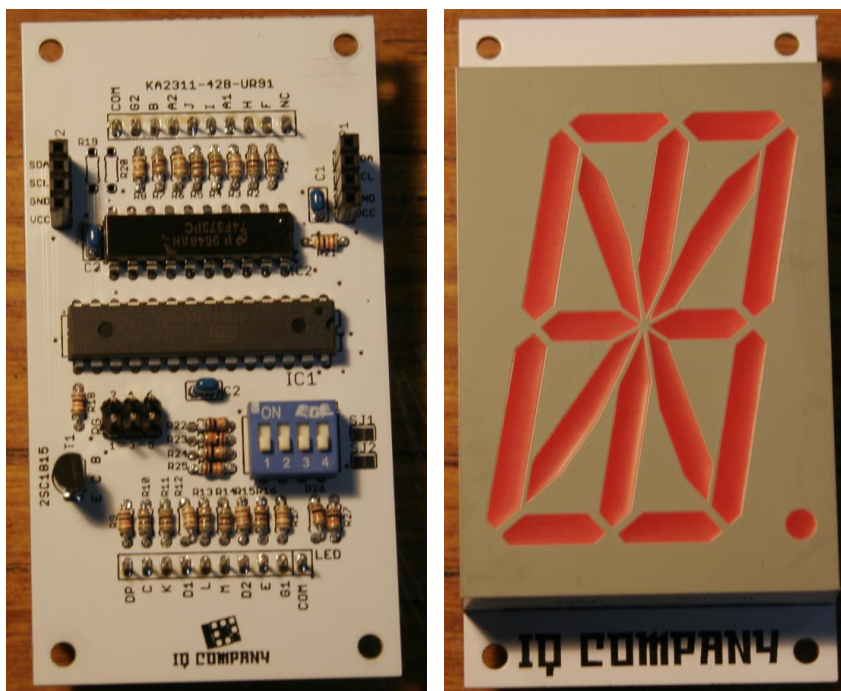


IQ\_COMPANY

## キットの組み立て方

- 部品表と回路図を確認しながら組み立てて下さい。ハンダ付けの順番は背の低い部品から行くと作業しやすいです。一般的な順番は以下のとおりです。参考にしてください。
  1. 抵抗、コンデンサ
  2. IC ソケット、DIP スイッチ、トランジスタ
  3. ピンヘッダ、ピンソケット
  4. LED
- IC(ソケットも)、DIP スイッチ、トランジスタ、LED には極性があります。方向に注意して下さい。
- JP1 および JP2 は 4 ピンのピンヘッダ(オス)またはピンソケット(メス)の好きな方を取り付けて下さい。キットには両方入っています。
- R19とR20はI2Cのプルアップ用です。通常はマスター側でプルアップするので本モジュールでの実装は不要です。
- LED のみ基板の裏面に実装します。
- 本キットのマイコン(ATMEGA328P)には出荷時にファームウェアを書き込んであります。
- **本基板ではパターンミスが一箇所あります。次ページの説明をよく読み、LED 取り付け前に一箇所ジャンパ処理を行なって下さい。**

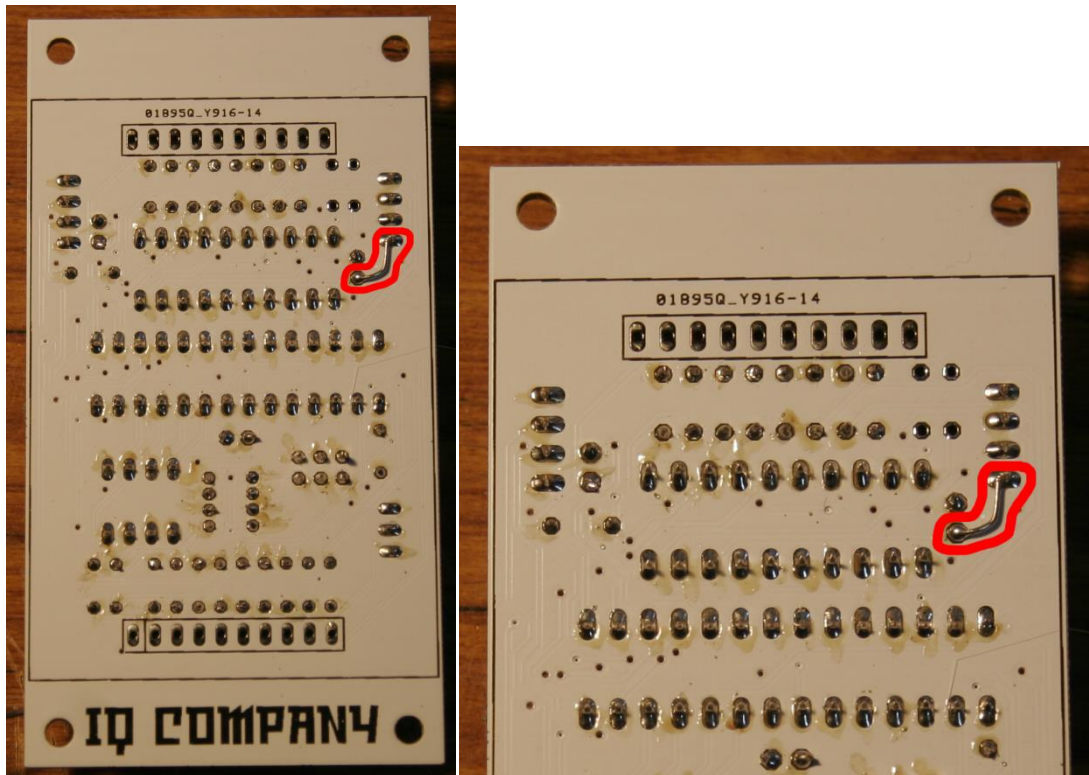
## モジュール完成写真



### ジャンパ処理

本キットで使用している基板には一箇所パターンミス(未配線)があります。お手数をお掛けしますが、**LED 取り付け前** にジャンパ処理をお願いします。

- JP2 の VCC が未接続です。近くの VCC に接続して下さい。一番近いのは C3 の下側のピンになります。写真を参考にしてください。



- C3 の VCC 側リードをカットしないでおいて、そのリードを利用するのが処理しやすいと思います。
- ジャンパ処理後は必ず、JP2 の VCC が他の VCC (例えば JP1 の VCC) と接続されていることをテスター等で確認してください。LED をハンダ付けした後にはリワークできませんのでご注意ください。

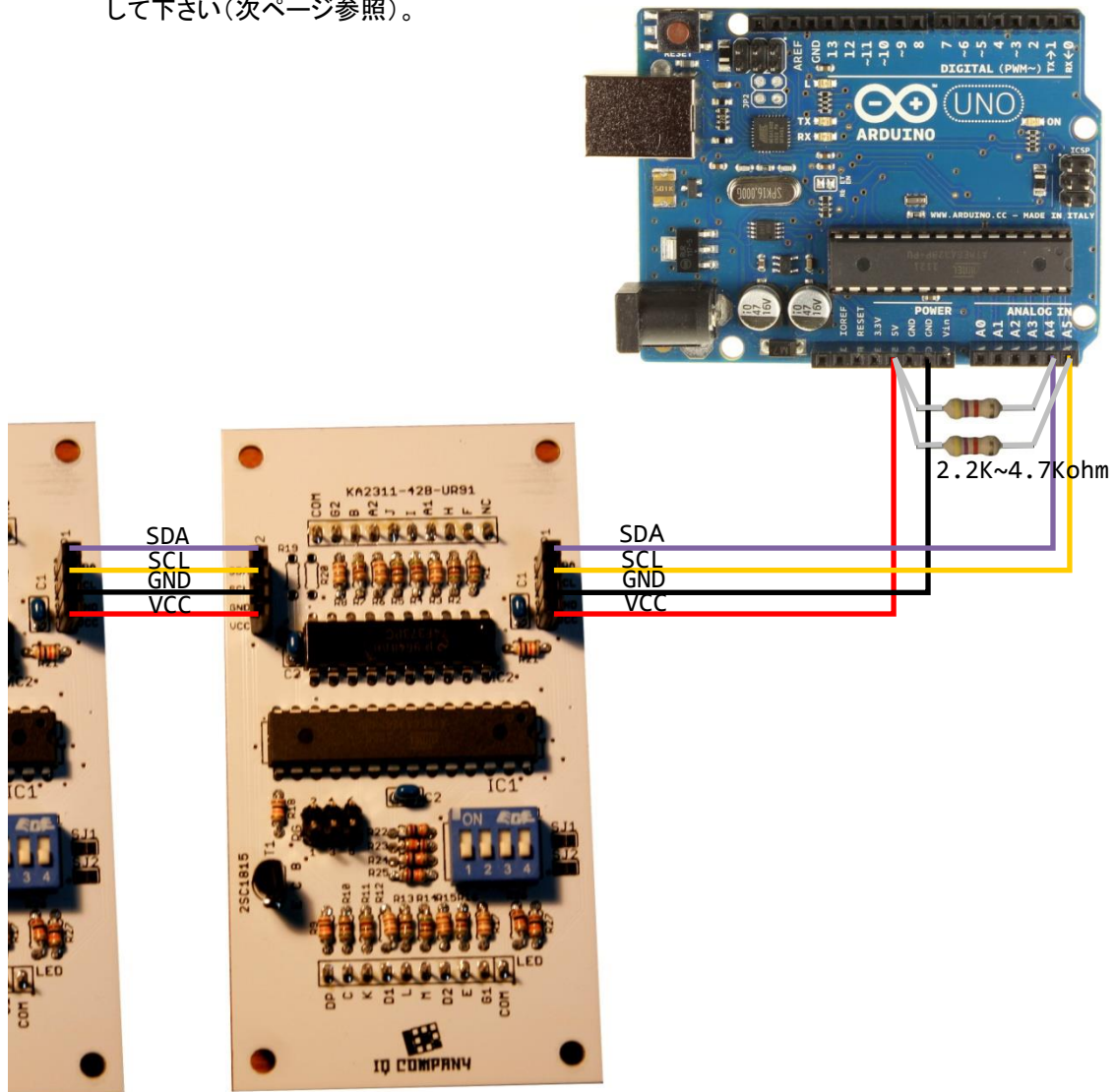
## 本モジュールの使い方

### Arduino で制御する場合

- Arduino と本モジュールを下図の様に接続して下さい。

Arduino	LED モジュール
A5 (SDA)	SDA
A4 (SCL)	SCL
5V	VCC
GND	GND

- I2C 通信の信号線 (SDA と SCL) は 3.3Kohm 程度の抵抗でのプルアップが必要です。図の様に接続して下さい。
- 複数の本モジュールを同一バスに接続する場合、全てのモジュールを異なるアドレスに設定して下さい(次ページ参照)。



## スレーブアドレス

- 本モジュールのスレーブアドレスは DIP スイッチ(S1)と 2 箇所のソルダージャンパ(SJ1,SJ2)で選択できます。

アドレス	ジャンパ		DIP スイッチ(S1)			
	SJ1	SJ2	1	2	3	4
0x40	OFF	OFF	OFF	OFF	OFF	OFF
0x41	OFF	OFF	ON	OFF	OFF	OFF
0x42	OFF	OFF	OFF	ON	OFF	OFF
0x43	OFF	OFF	ON	ON	OFF	OFF
0x44	OFF	OFF	OFF	OFF	ON	OFF
0x45	OFF	OFF	ON	OFF	ON	OFF
0x46	OFF	OFF	OFF	ON	ON	OFF
0x47	OFF	OFF	ON	ON	ON	OFF
0x48	OFF	OFF	OFF	OFF	OFF	ON
0x49	OFF	OFF	ON	OFF	OFF	ON
0x4A	OFF	OFF	OFF	ON	OFF	ON
0x4B	OFF	OFF	ON	ON	OFF	ON
0x4C	OFF	OFF	OFF	OFF	ON	ON
0x4D	OFF	OFF	ON	OFF	ON	ON
0x4E	OFF	OFF	OFF	ON	ON	ON
0x4F	OFF	OFF	ON	ON	ON	ON
0x50	OFF	ON	OFF	OFF	OFF	OFF
0x51~0x5E	OFF	ON	(0x42~0x4F と同様)			
0x5F	OFF	ON	ON	ON	ON	ON
0x60	ON	OFF	OFF	OFF	OFF	OFF
0x61~0x6E	ON	OFF	(0x42~0x4F と同様)			
0x6F	ON	OFF	ON	ON	ON	ON
0x70	ON	ON	OFF	OFF	OFF	OFF
0x71~0x7E	ON	ON	(0x42~0x4F と同様)			
0x7F	ON	ON	ON	ON	ON	ON

## 制御コマンド

---

**表示 ON** **0x01** **formats: { 0x01, c }**

---

表示する文字を設定します。表示できるキャラクタは 0x20~0x7F になります。

Examples:

```
{ 0x01, 'A' } // 'A'を表示
```

---

**表示 OFF** **0x02** **formats:{ 0x02 }**

---

表示を消します。引数はありません。

Examples:

```
{ 0x02 } // 表示を消す
```

---

**ドット ON** **0x03** **formats:{ 0x03 }**

---

ドットを表示します。引数はありません。

Examples:

```
{ 0x03 } // ドットを表示
```

---

**ドット OFF** **0x04** **formats:{ 0x04 }**

---

ドットを消します。引数はありません。

Examples:

```
{ 0x04 } // ドットを消去
```

---

**デューティ設定** **0x05** **formats:{ 0x05, d }**

---

LED 表示のデューティを設定します。引数は 0~255 の範囲で有効で、0 が一番暗く、255 が一番明るく表示します。

Examples:

```
{ 0x05, 255 } // デューティ 100%
```

```
{ 0x05, 127 } // デューティ 50%
```

**トランジション設定** `0x06` `formats:{ 0x06, t }`

表示文字設定時のトランジション効果が選択できます。現在のファームウェアでは3種類のトランジションが用意されています。引数とトランジションは以下のとおりとなっています。なお#define は Arduino サンプルソース内で定義してあります。

値	#define	説明
0	TRANSITION_NONE	トランジションなし、設定した文字を即表示します。
1	TRANSITION_TYPHOON	台風トランジション、渦巻きの様な動きで表示が切り替わります。
2	TRANSITION_FADE	フェードアウト/イン、現在の表示が暗くなり新しい表示が浮かび上がってきます。
3	TRANSITION_CROSS_FADE	クロスフェード、現在の表示と新しい表示がクロスフェードします。

**Examples:**

```
{ 0x06, 3 } // クロスフェード設定
{ 0x06, TRANSITION_TYPHOON } // 台風トランジション設定
```



## Arduino サンプル(I2C マスター)

```

#include <Wire.h>

#define TRANSITION_NONE          0
#define TRANSITION_TYPHOON      1
#define TRANSITION_FADE         2
#define TRANSITION_CROSS_FADE   3

byte dev_addr = 0x41;           // スレーブアドレス

void setup() {
  Wire.begin();                 // I2C マスターの開始
  setDuty(dev_addr, 255);       // DUTY=100%
  dotOff(dev_addr);            // DOT ON
}

void loop() {
  for ( byte trns = TRANSITION_NONE; trns <= TRANSITION_CROSS_FADE; trns++) {
    setTransition(dev_addr, trns); // TRANSITION の設定
    dispOn(dev_addr, 'I');         // 'I'を表示
    delay(1000);
    dispOn(dev_addr, 'Q');         // 'Q'を表示
    delay(1000);
  }
}

// 以下 16segments LED Module 用

void setDuty(byte address, byte duty) {
  Wire.beginTransmission(address);
  byte data[2] = {0x05, duty };
  Wire.write(data, 2);
  Wire.endTransmission();
}

void setTransition(byte address, byte transition) {
  Wire.beginTransmission(address);
  byte data[2] = {0x06, transition };
  Wire.write(data, 2);
  Wire.endTransmission();
}

void dispOn(byte address, byte disp) {
  Wire.beginTransmission(address);
  byte data[2] = { 0x01, disp };
  Wire.write(data, 2);
  Wire.endTransmission();
}

void dispOff(byte address) {
  Wire.beginTransmission(address);
  Wire.write(0x02);
  Wire.endTransmission();
}

void dotOn(byte address) {
  Wire.beginTransmission(address);
  Wire.write(0x03);
  Wire.endTransmission();
}

void dotOff(byte address) {
  Wire.beginTransmission(address);
  Wire.write(0x04);
  Wire.endTransmission();
}

```

## ファームウェア

- 本キットのマイコン(ATMEGA328P)には出荷時に既にファームウェアが書き込まれています。
- ファームウェアのソースコードは HP で公開していますので、ユーザ様が目的に合わせて自由に改変して利用いただいて構いません。
- 最新ファームウェアは以下ページで公開しています。  
<http://iqcompany.jp/products/16-segments-led-module/>
- なお、ファームウェアを改変することによる動作不良については対応できませんのでご了承下さい。

## ヒューズ設定

Device: ATMEGA328P

EXTENDED=0xFD

BODLEVEL=101 (BOD Level=2.7V)

HIGH=0xDF

RSTDISBL=1 (External Reset Enable)

DWEN=1 (debugWIRE Disable)

SPIEN=0 (SPI Programming Enable)

WDTON=1 (Watchdog Timer Off)

EESAVE=1 (EEPROM Erase through the Chip Erase)

BOOTSZ=11 (Boot Size = 256 words)

BOOTRST=1 (Reset Vector=Normal)

LOW=0xE2

CKDIV8=1 (Not Divide clock by 8)

CKOUT=1 (Clock output Disable)

SUT=10 (Start-up time = 14CK + 65ms)

CKSEL=0010 (Internal RC Oscillator 8MHz)

LOCKBIT=0xCF

BLB1=00

BLB0=11

LB2=1

LB1=1

## モジュール仕様

使用マイコン/クロック周波数	ATMEGA328P / 8MHz
サイズ	48 × 90 × 20(mm)
電源電圧	4.5V - 5.5V
消費電流	100mA(最大)
制御方式	I <sup>2</sup> C (100KHz)
I <sup>2</sup> C デバイスアドレス	0x40~0x7F (ユーザー設定可能)

## 外形寸法図

