

## 1. HC-SR04と不具合対策

1.1 超音波センサ.pdf	2 - 5
1.2 FAQ詳細 秋月電子通商 電子部品 ネット通販.pdf	6
1.3.1 超音波センサの回路変更（抵抗除去）.pdf	7
1.3.2 HC-SR04 テスト回路(160308).pdf	8
1.4 超音波センサの不具合対策.pdf	9
1.5 prg-Arduino.pdf	10

## 2. HC-SR04とH8との通信関連

2.1 通信用回路図(17.7.29).pdf	11
2.2 H8マニュアル(h8_3069f,port2).pdf	12 - 15
2.3 実験7(軌道追従制御実験)の一部.pdf	16
2.4 回路図(H8付属).pdf	17
2.5 prg-H8.pdf	18

## 3. 補足事項

3.3 注意・補足事項(1).pdf	19
3.3 注意・補足事項(2).pdf	20
3.3 注意・補足事項(3).pdf	21

# HC-SR04 User Guide

## 1. Ultrasonic Distance Measurement Principles

The transmitter emits a 8 bursts of an directional 40KHz ultrasonic wave when triggered and starts a timer. Ultrasonic pulses travel outward until they encounter an object, The object causes the the wave to be reflected back towards the unit. The ultrasonic receiver would detect the reflected wave and stop the stop timer. The velocity of the ultrasonic burst is 340m/sec. in air. Based on the number of counts by the timer, the distance can be calculated between the object and transmitter The TRD Measurement formula is expressed as:  $D = C \times T$  which is know as the time/rate/distance measurement formula where D is the measured distance, and R is the propagation velocity (Rate) in air (speed of sound) and T represents time. In this application T is devided by 2 as T is double the time value from transmitter to object back to receiver.

## 2. Product Features

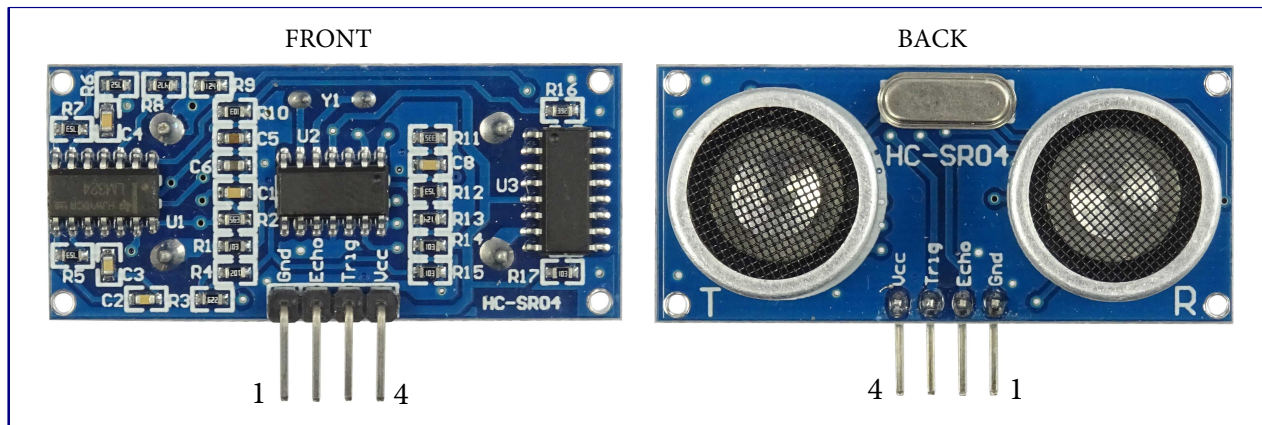
### Features

- Stable performance (Xtal.)
- Accurate distance measurement
- High-density SMD Board
- Close Range (2cm)

### Uses

- Robotics barrier
- Object distance measurement
- Level detection
- Security systems
- Vehicle detection/avoidance

### 3. Product Views



### 4. Module Pin Assignments

	Pin Symbol	Pin Function Description
1	VCC	5V power supply
2	Trig	Trigger Input pin
3	Echo	Receiver Output pin
4	GND	Power ground

### 5. Electrical Specifications

#### **WARNING**

**Do Not connect Module with Power Applied! Always apply power after connecting Connect "GND" Terminal first**

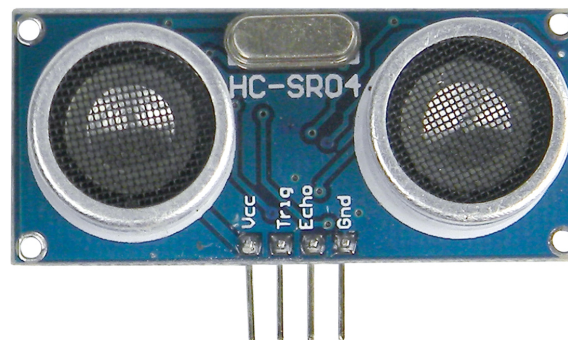
Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	5VDC
Operating Current	15mA
Operating Frequency	40KHz
Max. Range	4m
Nearest Range	2cm
Measuring Angle	15 Degrees
Input Trigger Signal	10us min. TTL pulse
Output Echo Signal	TTL level signal, proportional to distance
Board Dimensions	1-13/16" X 13/16" X 5/8"
Board Connections	4 X 0.1" Pitch Right Angle Header Pins

## 6. Module Operation

Set Trig and Echo Low to initialize module. Place a minimum 10us High level pulse to "Trigger" (module will automatically send eight 40KHz acoustic bursts). At the same time, Gate the microcontroller timer to start timing.

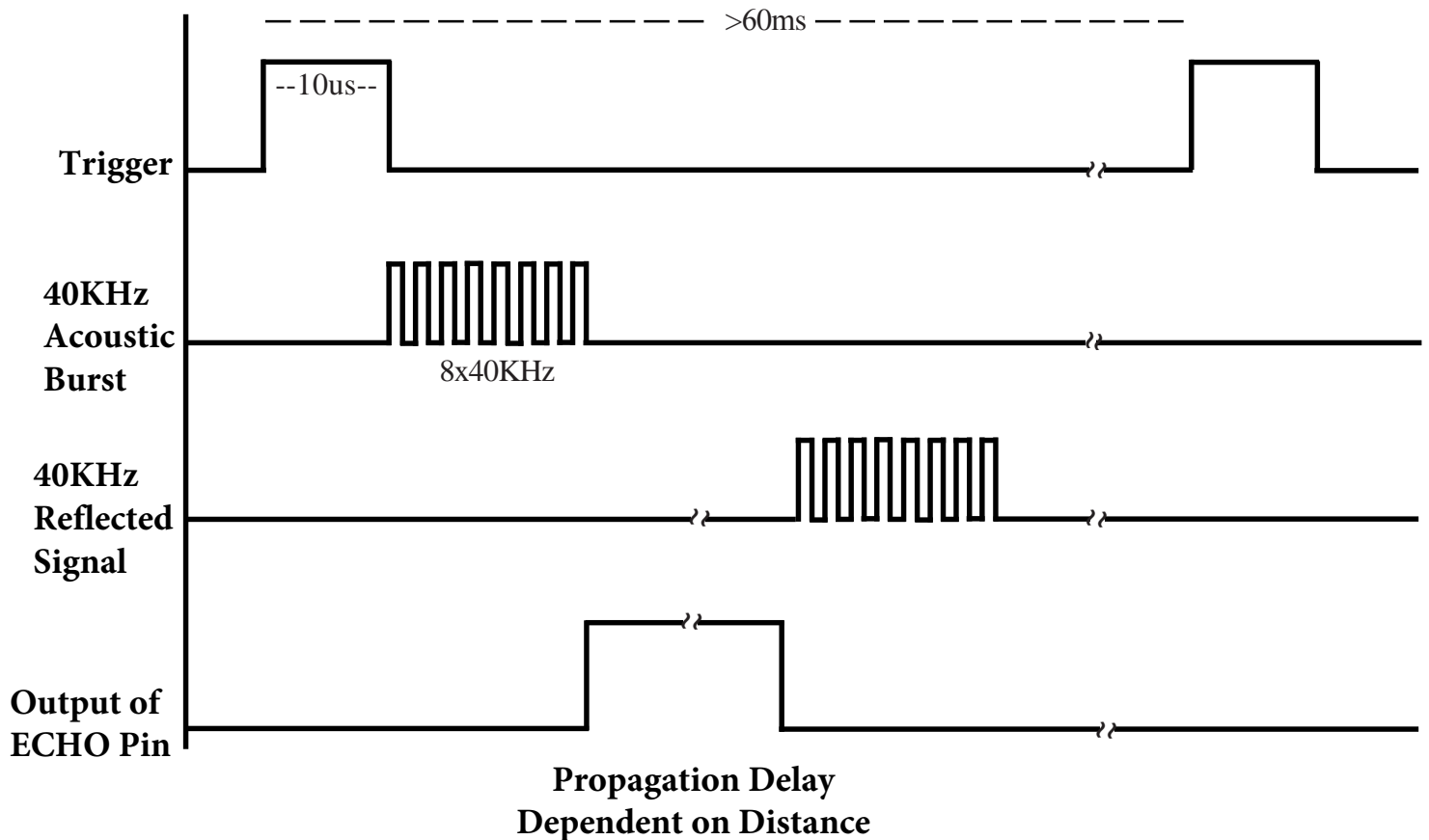
Wait to capture the rising edge output of ECHO port to stop the timer. Now read the time of the counter, which is the ultrasonic propagation time in the air. According to the formula: Distance = (ECHO high level time X ultrasonic velocity (Speed of Sound in air 340m/sec) / 2, you can calculate the distance to the obstacle.

For best results and maximum range, the Object should be larger than 0.5M<sup>2</sup>, the nearer the target object, the smaller it may be.



## 7. ModuleTiming

### HC-SR04 ULTRASONIC MODULE



**Trigger**  $10\mu s$  min. start measurement from microcontroller.

Max Rep. Rate: 60ms

**ECHO** Output pulse to microcontroller, width is the time from last of 8 40KHz bursts to detected reflected signal (microcontroller Timer gate signal)

Distance in cm = echo pulse width in  $\mu s / 58$

Distance in inch = echo pulse width in  $\mu s / 148$

Information obtained from or supplied by Mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.

# 秋月電子通商

マイページ

注文書

お問い合わせ

かごの中身

トラ技広告

回路図集

クイック注文：通販コードを入力  
(アルファベット+数字)  
 -  数量   
[通販コード一括入力フォーム](#)

[商品カタログ](#) | [新商品](#) | [お知らせ](#) | [注文方法](#) | [振込先](#) | [よくある質問](#) | [ダウンロード](#) | [トラ技広告\(PDF\)](#) | [配送状況確認](#) | [ログイン](#)

[トップ](#) > [センサー一般](#) > [超音波センサ](#) > [超音波距離センサー HC-SR04 \(2~180cm\)](#)

## 超音波距離センサー HC-SR04 (2~180cm) の質問と回答



【質問】 測距範囲外になると、どのような動作をしますか？

[2015/07/28 15:13:21]

【回答】 測距範囲外(反射時間約10ms以上)になると、エコー出力がHを保持します。距離センサーの電源を切ってリセットしてください。



[詳細画像4](#)

[この商品を友達に教える](#)

[お気に入り追加する](#)

- この内容は一般的な使用用途・使用環境を想定してお答えしています。商品の使用目的以外、特殊用途でのご使用については必ずしもこの回答通りにならないこともございますので、予めご承知おきください。

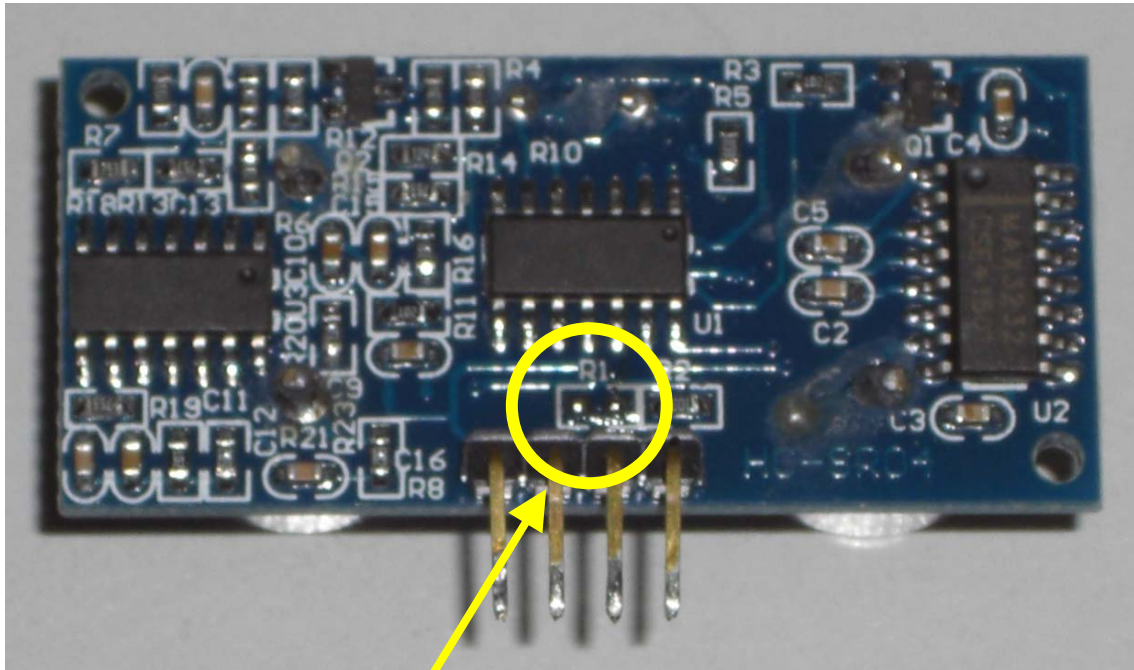
通販コードを入力してください。

-

[新TOP](#) | [旧TOP](#) | [注文方法](#) | [秋葉原店](#) | [八潮店](#) | [通販センター](#) | [川口物流センター](#) | [会社案内](#) | [特定商取法に基づく表示](#)

copyright c 2002-2016 秋月電子通商 AKIZUKI DENSHI TSUSHO CO.,LTD.

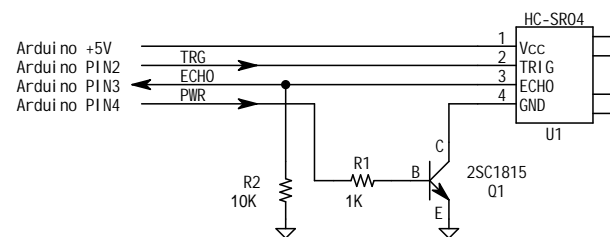
## 超音波センサの回路変更



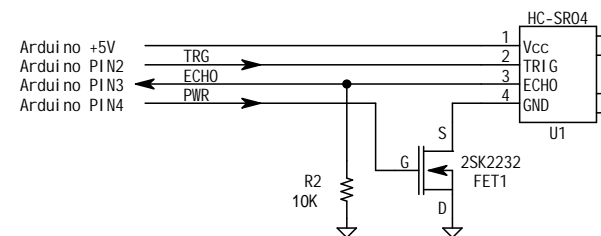
抵抗 R1 を半田ごてを使って取り外す（半田を少し追加し、こてを横にし、2箇所同時に温め、下にスライド）。

うまく行かないときは、こちらでやります。

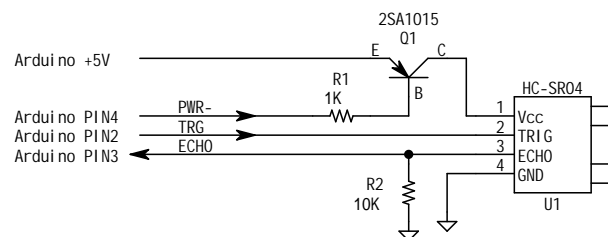
## ローサイドスイッチ NPNトランジスタ



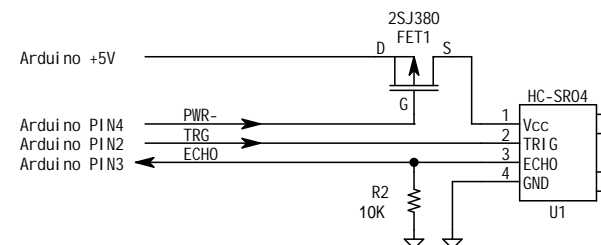
## ローサイドスイッチ NchFET



## ハイサイドスイッチ PNPトランジスタ



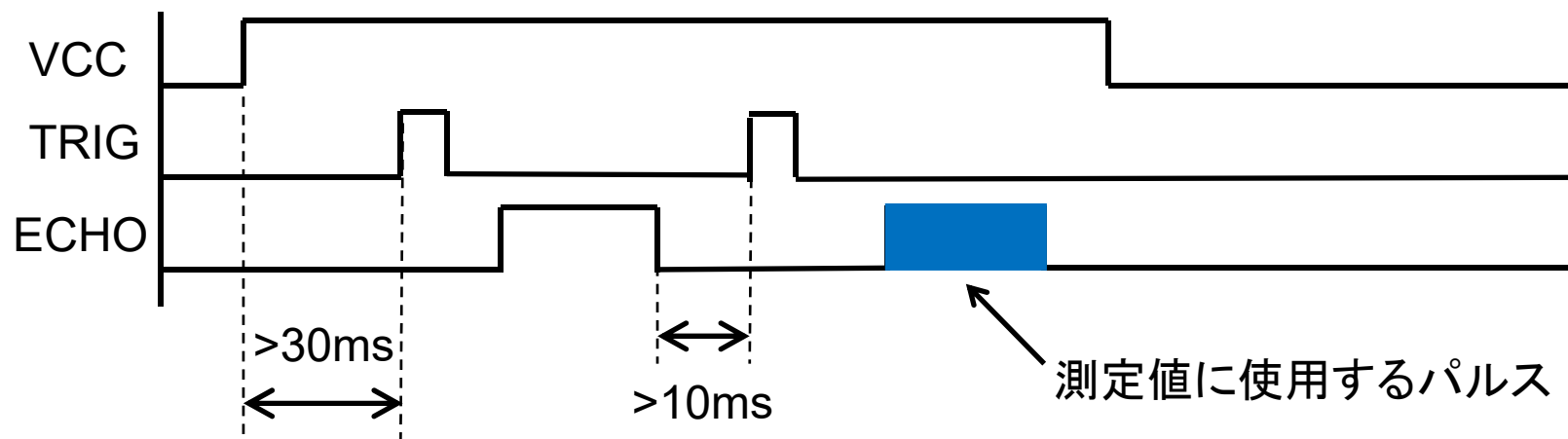
## ハイサイドスイッチ PchFET



TITLE		DRAWING_No.	
HC-SR04 テスト回路			
SHEET	DATE	DESIGN	
1 / 1	16.03.08	M. Tanaka	



- 超音波センサ 不具合対策
- 電源投入から初回トリガ入力まで  
30[ms]以上必要
- ダミーリードが必要？  
初回はエコー出力がほぼ固定になってしまう
- 2回目とのリードの間は10[ms]以上あけないと  
正常なエコー出力が得られない



```

// *****
// *** 超音波センサ(HC-SR04) 読み取りプログラム ***
// *****
#define PIN_TRG 2
#define PIN_ECHO 3
#define PIN_PWR 4

#define LEDPIN 13 // 13番ピンに出力 (LED点灯)
#define T_dist 10.0 // [cm] 閾値の距離

int interval;
double distance;

void setup() {
  // put your setup code here, to run once:
  pinMode(PIN_TRG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  pinMode(PIN_PWR, OUTPUT);
  Serial.begin(115200);

  digitalWrite(PIN_TRG, LOW);
  digitalWrite(PIN_PWR, HIGH);

  // for Output
  // LEDに接続ピンをデジタル出力に設定
  pinMode(LEDPIN, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  // 電源投入
  digitalWrite(PIN_PWR, LOW);

  delay(30); // 時間待ち 30[ms]

  // ダミーリード
  // trigger pulse output (10us)
  digitalWrite(PIN_TRG, HIGH);
  delayMicroseconds(9);
  digitalWrite(PIN_TRG, LOW);
  // echo pulse width measure
  interval = pulseIn(PIN_ECHO, HIGH, 30000);

  delay(10); // 時間待ち 10[ms]

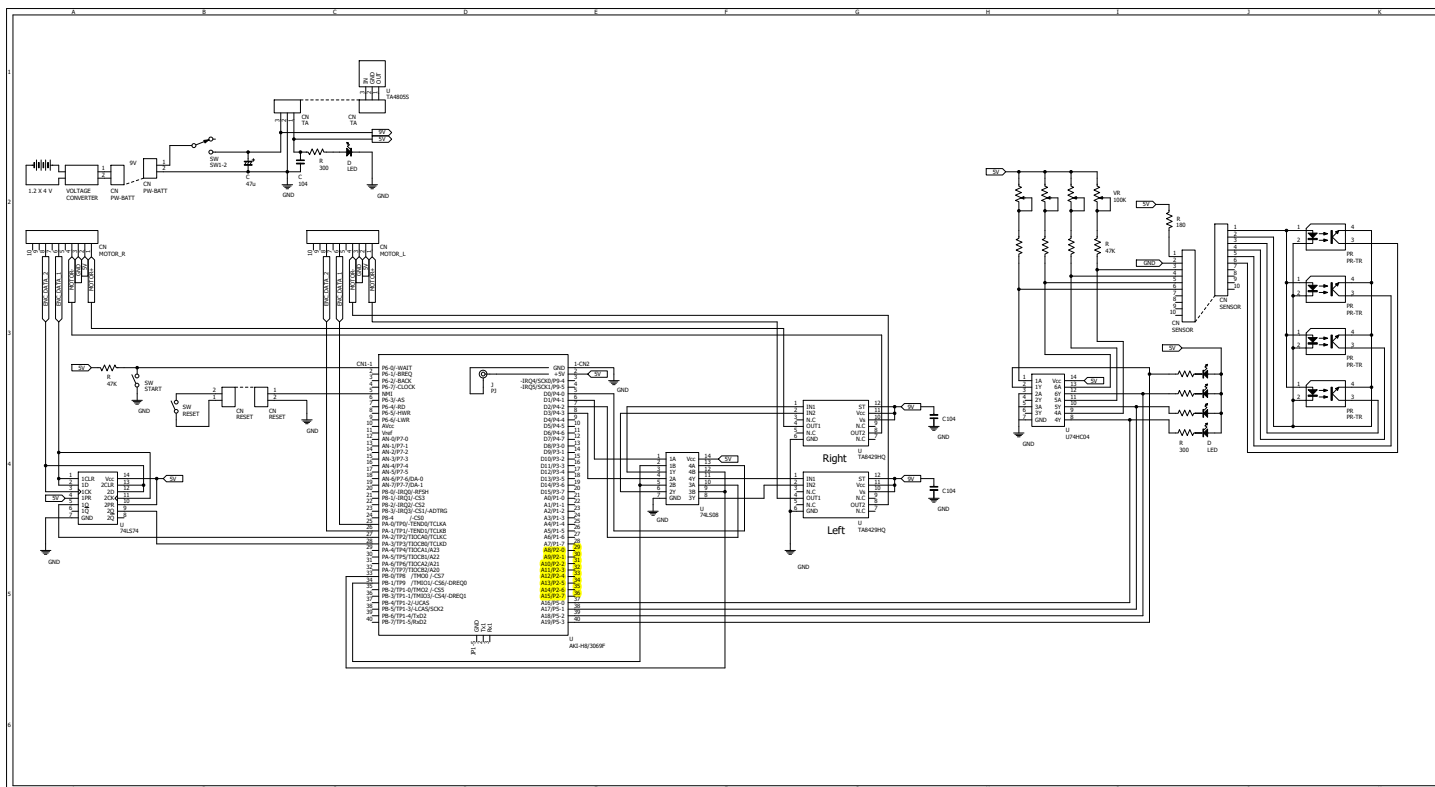
  // 距離計測
  // trigger pulse output (10us)
  digitalWrite(PIN_TRG, HIGH);
  delayMicroseconds(9);
  digitalWrite(PIN_TRG, LOW);
  // echo pulse width measure
  interval = pulseIn(PIN_ECHO, HIGH, 30000); // 時間計測
  distance = interval * 0.017; // 時間=>距離に変換 [cm]
  // 340[m/sec]/2 = 0.017[cm/μs]

  // send result
  Serial.print("pulse width =");
  Serial.print(interval);
  Serial.print("[us] distance =");
  Serial.print(distance);
  Serial.print("[cm]\n");

  // power off & delay
  digitalWrite(PIN_PWR, HIGH);
  delay(200); // adjustable

  // 出力設定 (LED)
  if( distance <= T_dist ){
    digitalWrite( LEDPIN, HIGH );
  }
  else{
    digitalWrite( LEDPIN, LOW );
  }
}

```



## 8.3 ポート 2

### 8.3.1 概要

ポート 2 は、アドレス出力兼用の 8 ビットの入出力ポートです。ポート 2 の各端子は、図 8.2 に示す構成となっており、動作モードにより端子機能が異なります。

モード 1 ~ 4 (内蔵 ROM 無効拡張モード) のときは、アドレスバス (A<sub>15</sub> ~ A<sub>8</sub>) 出力端子となります。モード 5 (内蔵 ROM 有効拡張モード) のときは、ポート 2 データディレクションレジスタ (P2DDR) の設定によりアドレスバス (A<sub>15</sub> ~ A<sub>8</sub>) または入力ポートとなります。

モード 7 (シングルチップモード) のときは、入出力ポートとなります。

エリア 2、3、4、5 に DRAM を接続する場合には、リード / ライトサイクルで A<sub>12</sub> ~ A<sub>8</sub> が ROW / COLUMN アドレス出力となります。詳細は「6.5 DRAM インタフェース」を参照してください。

ポート 2 は、プログラムで制御可能なプルアップ MOS が内蔵されています。また、1 個の TTL 負荷と 90pF の容量を駆動することや、LED、ダーリントントランジスタを駆動することができます。

ポート 2 端子		モード 1 ~ 4	モード 5	モード 7
ポ ー ト 2	↔ P27/A15	A15 (出力)	P27 (入力) / A15 (出力)	P27 (入出力)
	↔ P26/A14	A14 (出力)	P26 (入力) / A14 (出力)	P26 (入出力)
	↔ P25/A13	A13 (出力)	P25 (入力) / A13 (出力)	P25 (入出力)
	↔ P24/A12	A12 (出力)	P24 (入力) / A12 (出力)	P24 (入出力)
	↔ P23/A11	A11 (出力)	P23 (入力) / A11 (出力)	P23 (入出力)
	↔ P22/A10	A10 (出力)	P22 (入力) / A10 (出力)	P22 (入出力)
	↔ P21/A9	A9 (出力)	P21 (入力) / A9 (出力)	P21 (入出力)
	↔ P20/A8	A8 (出力)	P20 (入力) / A8 (出力)	P20 (入出力)

図 8.2 ポート 2 の端子構成

### 8.3.2 レジスタ構成

表 8.3 にポート 2 のレジスタ構成を示します。

表 8.3 ポート 2 レジスタ構成

アドレス*	名 称	略 称	R/W	初期値	
				モード 1~4	モード 5、7
H'EE001	ポート 2 データディレクションレジスタ	P2DDR	W	H'FF	H'00
H'FFFD1	ポート 2 データレジスタ	P2DR	R/W	H'00	
H'EE03C	ポート 2 入力プルアップ MOS コントロールレジスタ	P2PCR	R/W	H'00	

【注】 \* アドバンスモード時のアドレス下位 20 ビットを示しています。

#### (1) ポート 2 データディレクションレジスタ (P2DDR)

P2DDR は、8 ビットのライト専用のレジスタで、ポート 2 の各端子の入出力をビットごとに指定することができます。

ビット:		7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
モード 1~4	初期値:	1	1	1	1	1	1	1	1
	R/W :	—	—	—	—	—	—	—	—
モード 5、7	初期値:	0	0	0	0	0	0	0	0
	R/W :	W	W	W	W	W	W	W	W

ポート 2 データディレクション 7~0  
ポート 2 の各端子の入出力を選択するビットです。

#### (a) モード 1~4 (内蔵 ROM 無効拡張モード)

P2DDR は 1 に固定され、ポート 2 はアドレスバスとして機能します。

#### (b) モード 5 (内蔵 ROM 有効拡張モード)

ポート 2 はリセット直後は入力ポートとなっています。

P2DDR に 1 をセットすると対応するポート 2 の端子はアドレス出力端子となり、0 にクリアすると入力ポートになります。

(c) モード7 (シングルチップモード)

ポート2は入出力ポートとして機能します。P2DDRに1をセットすると対応するポート2の端子は出力ポートとなり、0にクリアすると入力ポートとなります。

モード1～4ではP2DDRは、リードすると常に1が読み出されます。ライトは無効です。  
 モード5、7ではP2DDRは、ライト専用のレジスタで、リードは無効です。リードすると1が読み出されます。

P2DDRは、リセット、またはハードウェアスタンバイモード時にモード1～4の場合はH'FFに、モード5、7の場合はH'00にイニシャライズされます。ソフトウェアスタンバイモード時には、直前の状態を保持します。そのため、ポート2が入出力ポートとして機能しているとき、P2DDRが1にセットされた状態でソフトウェアスタンバイモードに遷移するとその端子は出力状態のままとなっています。

(2) ポート2 データレジスタ (P2DR)

P2DRは、8ビットのリード/ライト可能なレジスタで、ポート2の出力データを格納します。ポート2が出力ポートとして機能する場合、本レジスタの値が出力されます。また、このレジスタをリードすると、P2DDRが0のビットは端子のロジックレベルが読み出され、1のビットはP2DRの値が読み出されます。

ビット:	7	6	5	4	3	2	1	0
	P27	P26	P25	P24	P23	P22	P21	P20
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

——— ポート2データ7～0

ポート2の各端子のデータを格納するビットです。

P2DRは、リセット、またはハードウェアスタンバイモード時に、H'00にイニシャライズされます。ソフトウェアスタンバイモード時には、直前の状態を保持します。

## (3) ポート 2 入力プルアップ MOS コントロールレジスタ (P2PCR)

P2PCR は 8 ビットのリード/ライト可能なレジスタで、ポート 2 に内蔵した入力プルアップ MOS をビットごとに制御します。

ビット:	7	6	5	4	3	2	1	0
	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR
初期値:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ポート2入力プルアップMOSコントロール7~0  
ポート2に内蔵した入力プルアップMOSを制御するビットです。

モード 5、7 のとき、P2DDR を 0 にクリアした (入力ポートの) 状態で P2PCR を 1 にセットすると対応するビットの入力プルアップ MOS は ON します。

P2PCR は、リセット、またはハードウェアスタンバイモード時に、H'00 にイニシャライズされます。ソフトウェアスタンバイモード時には、直前の状態を保持します。

表 8.4 入力プルアップ MOS の状態 (ポート 2)

モード	リセット	ハードウェア スタンバイモード	ソフトウェア スタンバイモード	その他の動作時
1	OFF	OFF	OFF	
2				
3				
4				
5	OFF	OFF	ON / OFF	
7				

## 【記号説明】

OFF : 入力プルアップ MOS は、常に OFF 状態です。

ON / OFF : P2PCR = 1 かつ P2DDR = 0 のとき ON 状態、その他のときは OFF 状態です。

```

// *****
// *** 実験7 のプログラムの一部 ***
// *****

// -----
// --- H8の初期設定関数 ---
// -----

static void
init_settings( void )
{
    // *** NMI 設定 (リセット用) ***
    SYSCR.BIT.NMIEG= 0;
    set_handler( 7, prg_end );

    // *** ITU2 位相係数モード設定 (for Left encoder) ***
    HEX_ITU.TMDR.BIT.MDF= 1; // 位相係数モード
    HEX_ITU.TMDR.BIT.FDIR= 1; // オーバフロー・アンダーフロー検知
    HEX_ITU.TSTR.BIT.STR2= 1; // 1:タイマスタート, 0:ストップ
    HEX_ITU2.TCNT =0;

    // *** ITU0, ITU1 カウンタ (for Right encoder) ***
    // --- ITU0: カウンタ ---
    HEX_ITU0.TCR.BIT.CCLR= 0; // クリア禁止
    HEX_ITU0.TCR.BIT.CKEG= 2; // 0:立ち上がり, 1:立下り, 2:両エッジ
    HEX_ITU0.TCR.BIT.TPSC= 6; // 外部クロックC: TCLKC端子入力でカウント 110
    HEX_ITU0.TCNT= 0;
    HEX_ITU.TSTR.BIT.STR0= 1; // 1:タイマスタート, 0:ストップ
    // --- ITU1: カウンタ ---
    HEX_ITU1.TCR.BIT.CCLR= 0; // クリア禁止
    HEX_ITU1.TCR.BIT.CKEG= 2; // 0:立ち上がり, 1:立下り, 2:両エッジ
    HEX_ITU1.TCR.BIT.TPSC= 7; // 外部クロックD: TCLKD端子入力でカウント 111
    HEX_ITU1.TCNT= 0;
    HEX_ITU.TSTR.BIT.STR1= 1; // 1:タイマスタート, 0:ストップ

    // *** ポート入出力設定 (for motor, photo-sensor) ***
    P4.DDR=0xff; // output (for motor)
    P5.DDR=0x00; // input (for photo-sensor)

    // *** 8bit timer pwm settings (for motor) 3069マニュアル pp.482 参照 ***
    // [ $\phi/8192 \rightarrow 3$ ], [ $\phi/64 \rightarrow 2$ ], [ $\phi/8 \rightarrow 1$ ]
    OCT_ITU0.TCR.BIT.CKS= 2;
    OCT_ITU1.TCR.BIT.CKS= 2;
    //OCT_ITU2.TCR.BIT.CKS= 2; // 予備ポート

    // TCNTをコンペアマッチAでクリア: CCLR0=1, CCLR1=0, pp.462
    OCT_ITU0.TCR.BIT.CCLR= 1;
    OCT_ITU1.TCR.BIT.CCLR= 1;
    //OCT_ITU2.TCR.BIT.CCLR= 1; // 予備ポート

    // コンペアマッチBで1出力: OIS3 と OIS2 を 01 に設定, pp.467
    OCT_ITU0.TCSR.BIT.OIS32= 1;
    OCT_ITU1.TCSR.BIT.OIS32= 1;
    //OCT_ITU2.TCSR.BIT.OIS32= 1; // 予備ポート

    // コンペアマッチAで0出力: OS1 と OS0 を 10 に設定, pp.468
    OCT_ITU0.TCSR.BIT.OS10= 2;
    OCT_ITU1.TCSR.BIT.OS10= 2;
    //OCT_ITU2.TCSR.BIT.OS10= 2; // 予備ポート

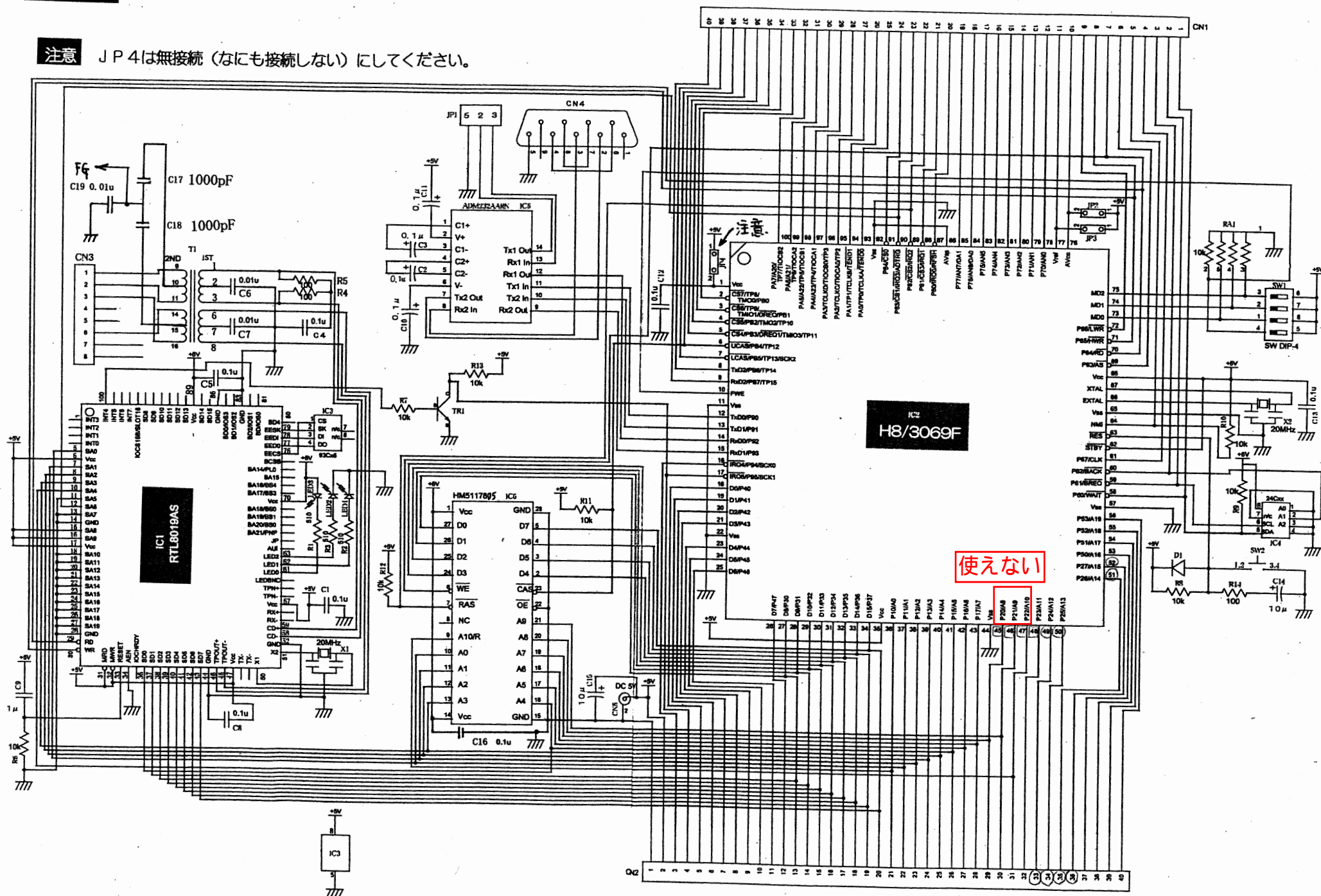
    // Duty
    OCT_ITU0.TCORB= 0;
    OCT_ITU1.TCORB= 0;
    //OCT_ITU2.TCORB= 0; // 予備ポート

    // TCRB: タイムコンスタントレジスタA
    // 周波数: 20MHz (Clock)/64( $\phi$ )/250(TCORA)= 1.25kHz
    OCT_ITU0.TCORA= Max_duty;
    OCT_ITU1.TCORA= Max_duty;
    //OCT_ITU2.TCORA= 250; // 予備ポート

    // *** ITU3 タイマ割り込み 8bit (for timer) ***
    // OCT_ITU3.TCORA: タイムコンスタントレジスタA pp.465
    // 1秒間に呼び出されるfeed関数の回数と同じ.
    // 20000000: システムクロック, 8192:分周値(r3069.hの197行目 CKS:3 (CKS2=0, CKS1=1, CKS0=1))
    // 内部クロック=システムクロック/分周値 pp.463
    OCT_ITU3.TCORA= (20000000/8192)*SampTime/1000;
    OCT_ITU3.TCNT = 0; // カウンタ, TCORAと8TCNTの値は常に比較されている. pp.460

```





```

// *****
// *** Port 読み込みテストプログラム ***
// *** produced by Y. Mori ***
// *** special thanks to A. Ruike, S. Kido ***
// *****
#include "r3069.h"
#include <mes2.h>

// *****
// *** main () ***
// *****
int
main(void) {

    // *** P2 の 5ビット目 (B4, CN2-33ピン) を, 入力ポート (PI) として使う ***
    //
    // 問題点)
    // P2.DR.BIT.B0~B2 は, 出力ポート (P0) の設定でH8ボードに使われている
    // ので, フォトリフレクタと同様の設定: P2.DDR= 0x00; が使えない.
    //
    // 解) DDR を使って, 上位ビットと下位ビットを別々に設定をする.
    // P2.DDR= 0x0f;
    //
    // --- B4~B7: 入力 (PI), B0~B3: 出力 (P0) ---
    //          BBBB BBBB
    //          7654 3210
    //          0x0f: 0000 1111, ただし 0:入力, 1:出力
    //
    // 参考文献: h8_3069f.pdf (pp. 321-324, 本文のページ: pp. 293-296)
    // C言語によるH8マイコンプログラミング入門 pp. 94-97

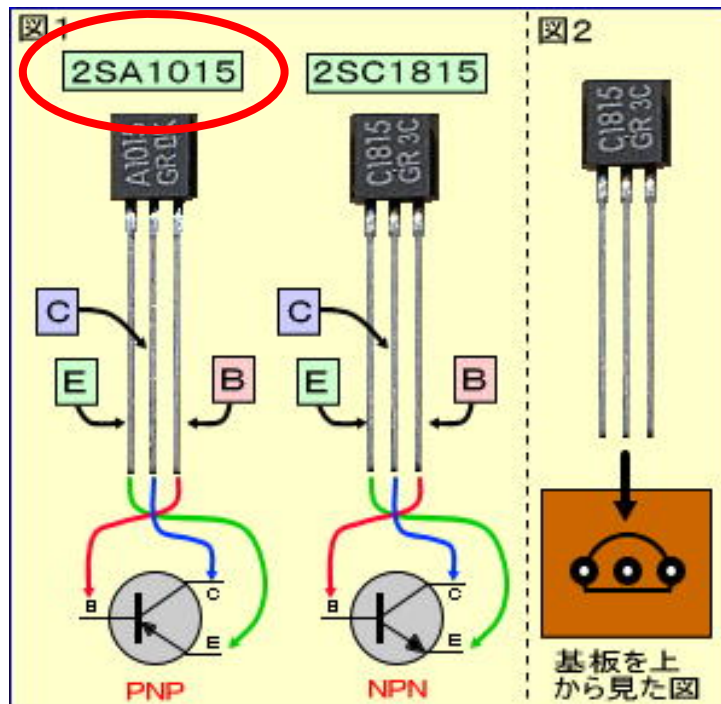
    P2.DDR= 0x0f;

    // B4ポートをチェック
    while(1){
        printf(" P2: 4=%d ¥r", P2.DR.BIT.B4 );
    }
}

```

## 注意・補足事項

- トランジスタのピン配置



エクボ

- ジャンパ線と基板の接続

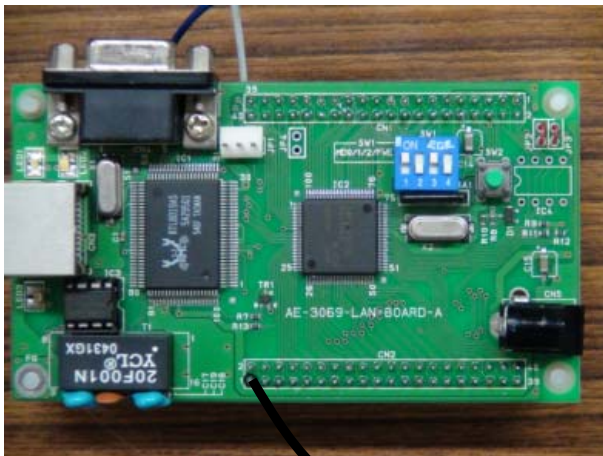


## ● 電源周りの注意事項

- ・ 電源コネクタ： 左が黒（GND）



- ・ アルディーノと H8 のグラウンド（GND, 0V）を共通に（短絡）すること



GND

- ・ サーボモータ，センサ（超音波，PSD）用の電源は，アルディーノから取らない

# 電源周りの配線

