

光源探索をする準受動四脚移動ロボット

松下光次郎

東京大学大学院 工学系研究科 精密機械工学専攻

matsushita@robot.t.u-tokyo.ac.jp

1	ものづくり講座の概要	2
1.1	準受動四脚移動ロボット	2
1.2	制御器概要	3
1.2.1	マイクロチップ「秋月通商製 H8/3664」	3
1.2.2	制御器	3
2	制御器の使い方	4
2.1	USB シリアルケーブル	4
2.2	プログラム開発環境	7
2.3	電子回路と H8 プログラム記述方式	8
2.4	H8 における C プログラミングの注意事項	9
2.5	H8 (ハードウェア) の書き込み準備	9
2.6	H8 プログラムのコンパイル	9
2.7	H8 プログラムの書き込み方法	11
2.8	H8 (ハードウェア) のプログラム実行準備	11
2.9	ハイパーターミナル	12
3	H8 プログラミング	14
3.1	センサ読み取り方法	14
(a)	抵抗値の計測	15
(b)	制御器による読み取り方法	15
(c)	H8 プログラミング : AD 変換	17
3.2	RC サーボモータの制御方法	17
(a)	概要	17
(b)	RC サーボモータのモータの制御方法 : PWM (パルス変調)	18
(c)	モータ制御のための H8 プログラム	19
4	準受動四脚移動ロボットの製作法	20
4.1	概要	20
4.2	光源探索機能の概要	

1 ものづくり講座の概要

時間	内容
13:00 - 13:15	- はじめに -
13:15 - 14:15	マイクロチップ制御器の使い方 1. H8/3664 の使い方 (プログラミング方法, 書き込み方法) 2. シリアル通信方法 3. センサ値の読取方法 4. モータ制御の方法 5. 準受動四脚ロボットのプログラム 【コメント】一通りの制御器の使用方法を理解してください。
14:15 - 16:30	ロボットの製作 1. ペットボトルを使用したロボット製作「準受動四脚移動ロボット」 2. センサフィードバック制御による自律移動 3. 光源探索を行う自律移動ロボットの完成へ 【コメント】試行錯誤による機構&制御パラメータ調整を楽しんで下さい
16:30 - 17:00	- コンテスト & 総括 -

準受動四脚移動ロボット

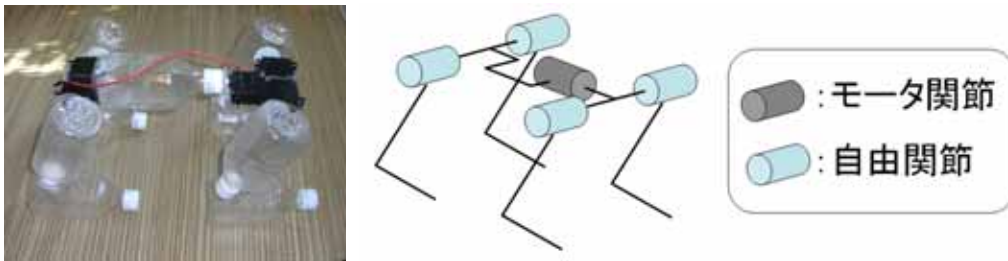


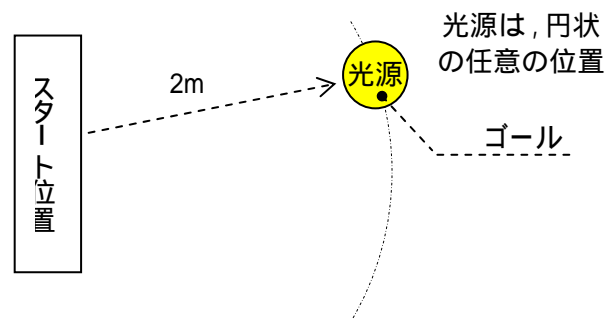
図 準受動四脚移動ロボット

コンテスト：光源探索レース

光センサ 2 個を利用して，光源探索する自律移動ロボットを製作してください。



(a) 実環境



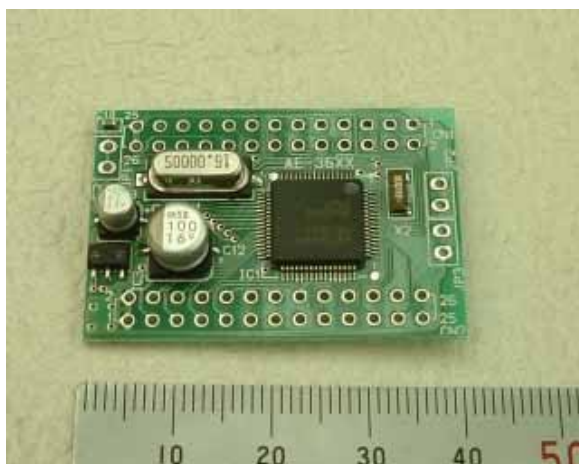
(b) 概念図

図 コンテスト環境

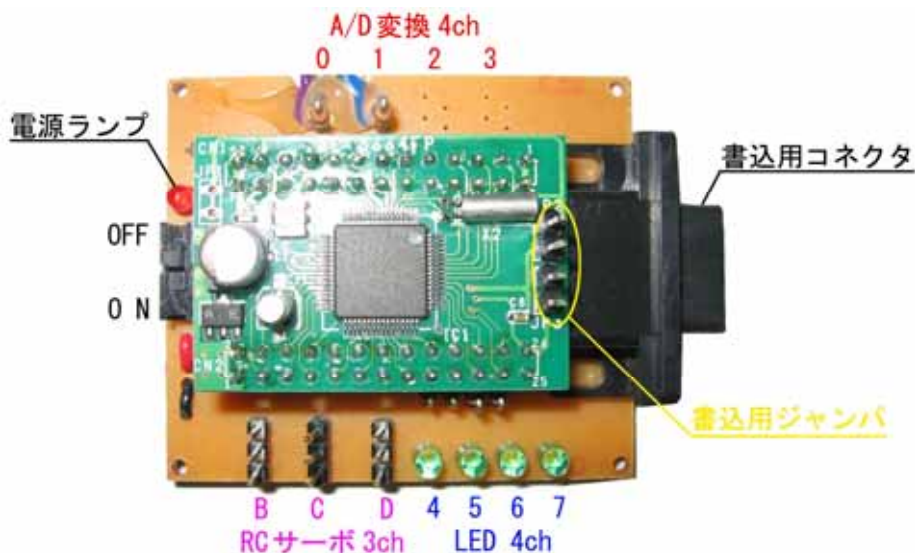
1.2 制御器概要

1.2.1 マイクロチップ「秋月通商製 H8/3664」

制御器は、日本国内のロボット系研究室に人気のある秋月通商製「H8/3664」を使用しています。このマイクロチップは、秋月通商の通信販売で簡単に購入することができ、1600円と安価です（リンク：<http://akizukidenshi.com/>）。また簡単な設定（C言語）で、サーボモータ3個の制御、8個のセンサ値を読み取り、デジタルインプット/アウトプットの制御、シリアル通信によるパソコンとのデータ送受信を可能とします。



1.2.2 制御器



電子回路ポート	内容
PWM 制御信号 3ch	RC サーボモータ制御
A/D 変換入力 2ch	光センサ状態監視
デジタル出力 4ch	LED 点灯・消灯

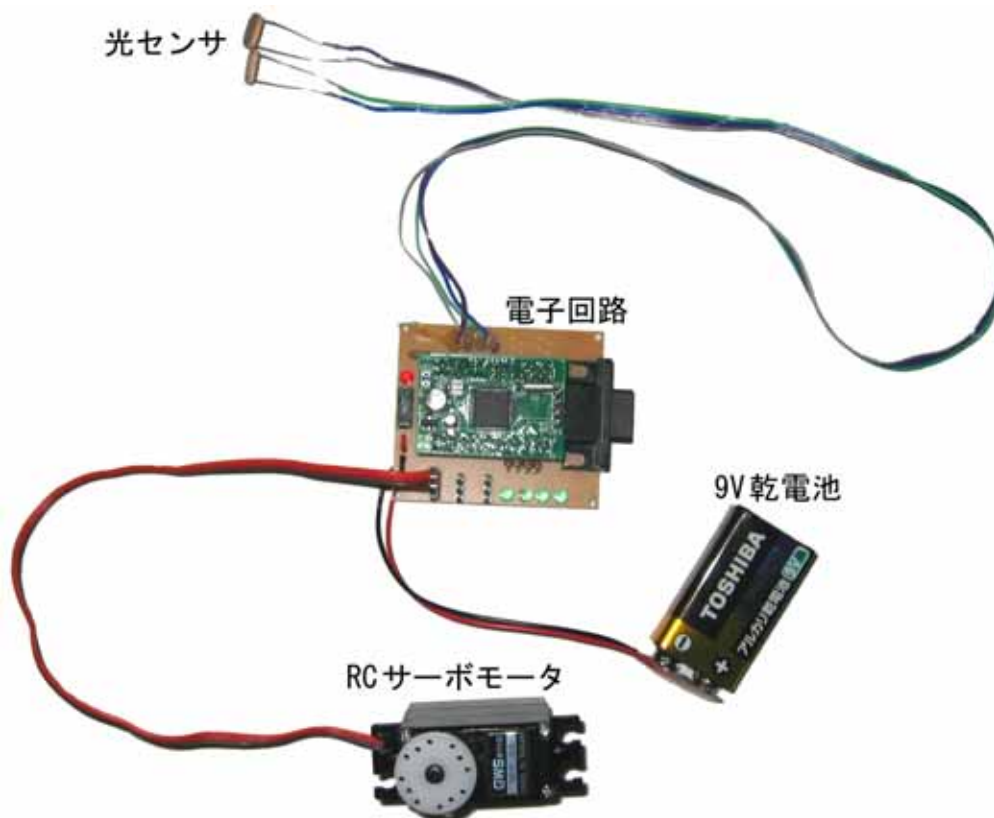


図 H8/3664 を用いた制御器の概観図

2. 制御器の使い方

2.1 USB シリアルケーブルのインストール

最近のノートパソコンには、図のようなシリアル通信ポート [Dsub 9 ピン (オス)] が使用されなくなってきています。ノートパソコンに使用されていない場合は、USB シリアルケーブルを使用してもらうこととなります。このケーブルは USB を利用して仮想的にシリアル通信ポートとする変換器となっています。



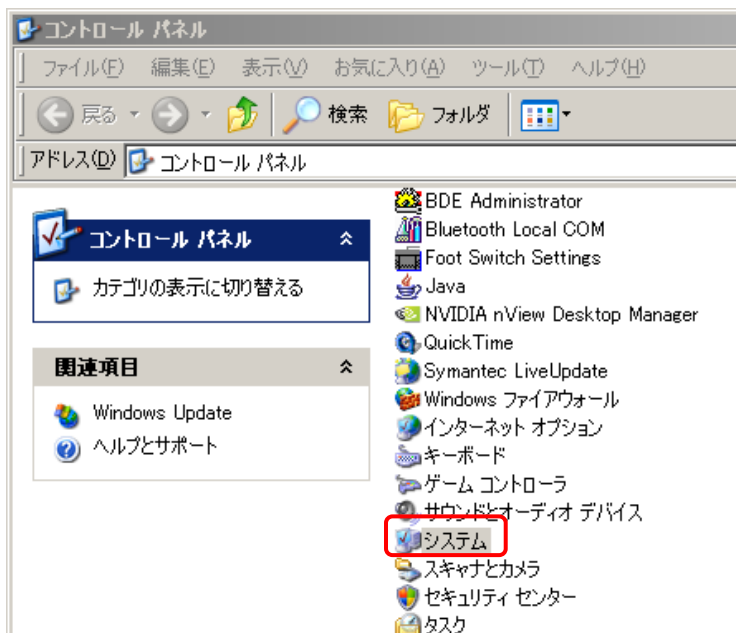
シリアル通信ポート
Dsub9 ピン (オス)



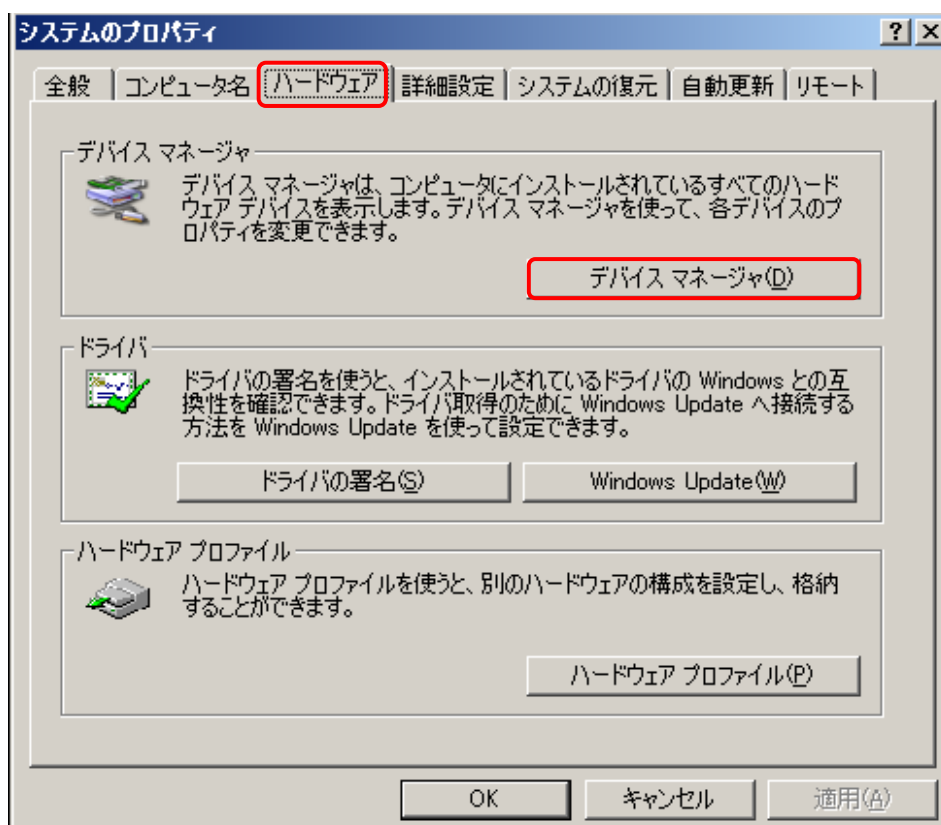
USB シリアルケーブル

次に、USB シリアル通信ケーブルを「COM1」に設定する必要があります。COM1 とは、第一番目のシリアル通信ポートに指定するということです。どのメーカーの USB シリアル通信ケーブルを使用する場合でもこの手順を行い、「COM1」であるかどうか確認してください。

まず、「コントロールパネル」を開き、「システム」のアイコンをダブルクリックしてください。

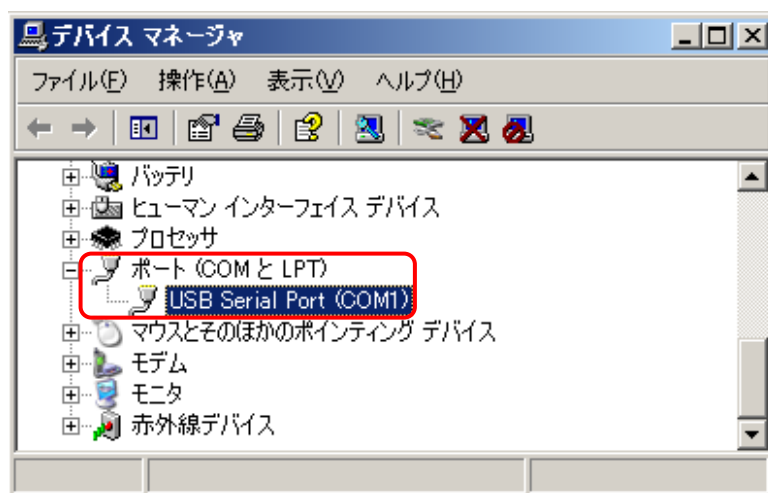


「システムのプロパティ」ウィンドウのタブである「ハードウェア」を選択肢、「デバイスマネージャ」を指定してください。

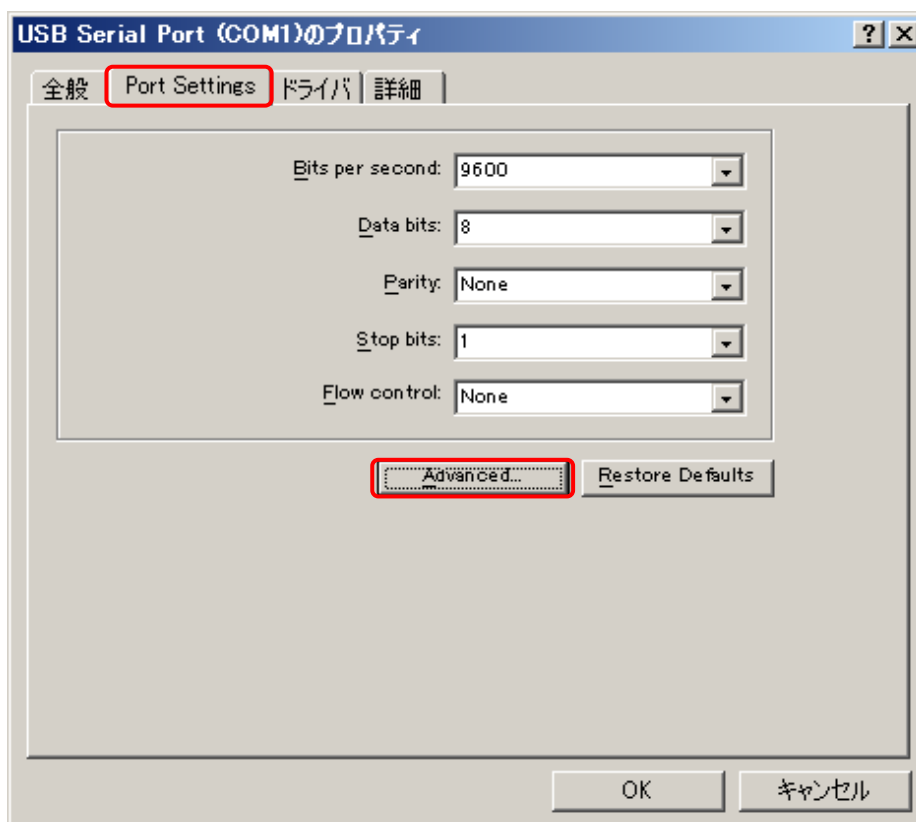


「デバイスマネージャ」のツリー構造中にある「ポート (COM と LPT)」を選択すると、「USB Serial Port (COM1)」と確認することができます。ここで「COM1」と表示されている場合は、問題ないのでこの手順を終了してもらって問題ありません。

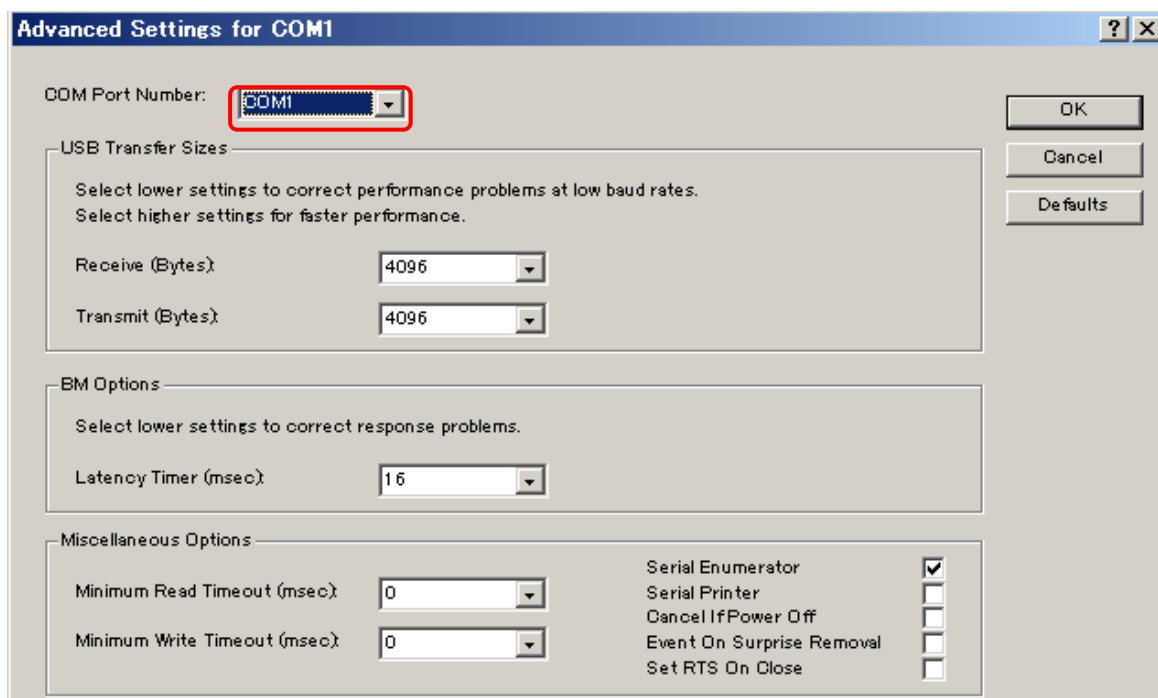
ここで「COM2」や「COM5」と表示されている場合は、「COM1」に直す必要があります。「USB Serial Port (COM2)」をダブルクリックしてください。



「USB Serial Port (COM2)のプロパティ」ウィンドウを開き、「Port Settings」のタブを選択し、「Advanced...」をクリックしてください。



「Advanced Settings for COM2」ウィンドウの最上段に「COM Port Number:」という項目があります。ここの項目で「COM1」をして、「OK」ボタンを押し、この手順を終了してください。



プログラム開発環境

「Cドライブ」直下には「h8」フォルダがあり、ここにプログラムファイルが保存されることとなります。

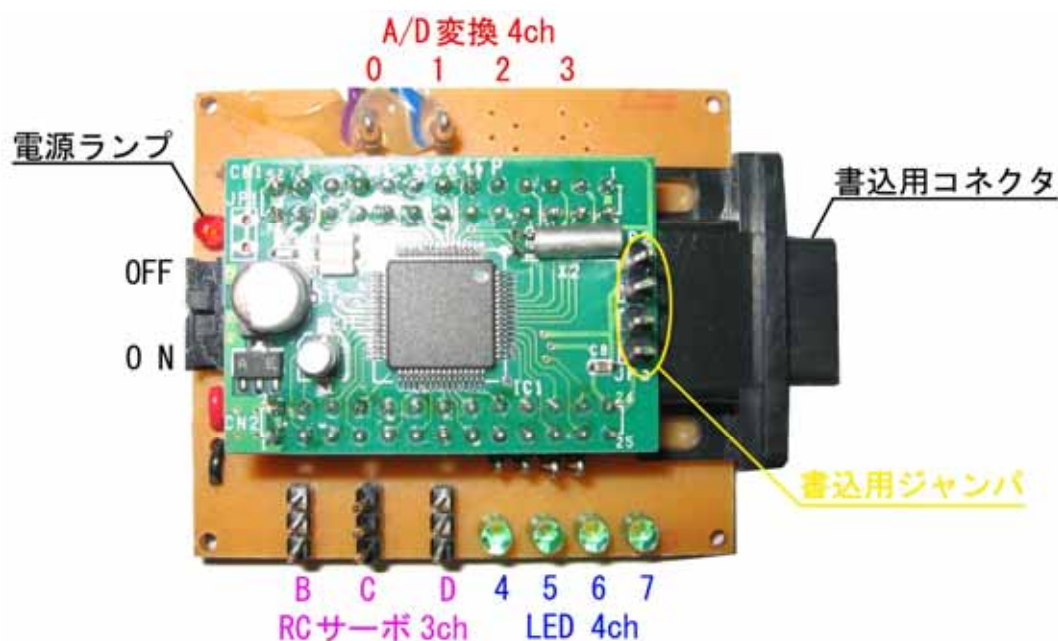


本講座では、サンプルプログラムを利用して授業を進めます。H8 プログラムの実行には、C ファイルと Sub ファイルが必要となります。なお、サンプルプログラムを利用するときは、色付部のみを変更することで簡単に利用できます。

	ファイル	内容
サンプル 1	01sensor.c 01sensor.sub	センサ読み取り
サンプル 2	02motor.c 02motor.sub	モータ制御
サンプル 3	03ppdw.c 03ppdw.sub	準受動四脚移動ロボットの基本プログラム
サンプル 4	04adjust.c 04adjust.sub	準受動四脚移動ロボットによる自律移動を実現するためのパラメータ調整用プログラム

電子回路と H8 プログラム記述方式

電子回路の各ポートの H8 プログラムにおける記述方式を下表に示します。



電子回路		プログラム		
RC サーボ 3ch	B	TW.GRB	RC サーボの軸位置	
	C	TW.GRC	左端	中央
	D	TW.GRD	1200	2600 3800
A/D 変換入力 2ch	0	x0	光の強弱	
	1	x1	明るい(0)	暗い(255)
LED 4ch	7	IO.PDR1.BIT.B7	LED 点灯・消灯	
	6	IO.PDR1.BIT.B6	点灯 : IO.PDR1.BIT.B7 = 1	
	5	IO.PDR1.BIT.B5	消灯 : IO.PDR1.BIT.B7 = 0	
	4	IO.PDR1.BIT.B4		




H8 における C プログラミングの注意事項

コメントは「//」ではなく、「/* */」で記入すること。

二つの条件を含む if 文の場合、「if(x0<10 && x0>50)」とはせず、二つの条件必ず括弧で囲「if((x0<10)&&(x0>50))」とすること

このコンパイラは、「if(x0=20)」の記述間違いにおいてエラーが表示されないので、気をつけること。(if文が「if(x0==20)」というように「==」で記述されているのを確認すること。)

H8 (ハードウェア) の書き込み準備

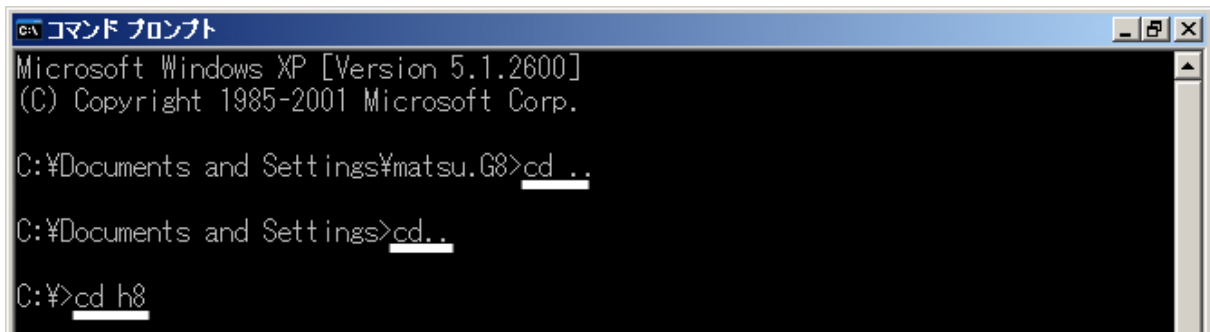
内容	写真
<p><u>書込手順 1</u> 書き込み用ジャンパを用意します。</p>	
<p><u>書込手順 2</u> 電源を OFF にして、ジャンパを書き込み用ジャンパピンに取り付け、また Dsub 9 ピンをシリアル通信ポートに接続してください。</p>	
<p><u>書込手順 3</u> ジャンパを取り付けた状態で、電源を ON にして、書き込み準備ができました。「コマンドプロンプト」を立ち上げ、作成したプログラムのコンパイルおよび「hterm」による H8 への書き込みを行ってください。</p>	

H8 プログラムのコンパイル

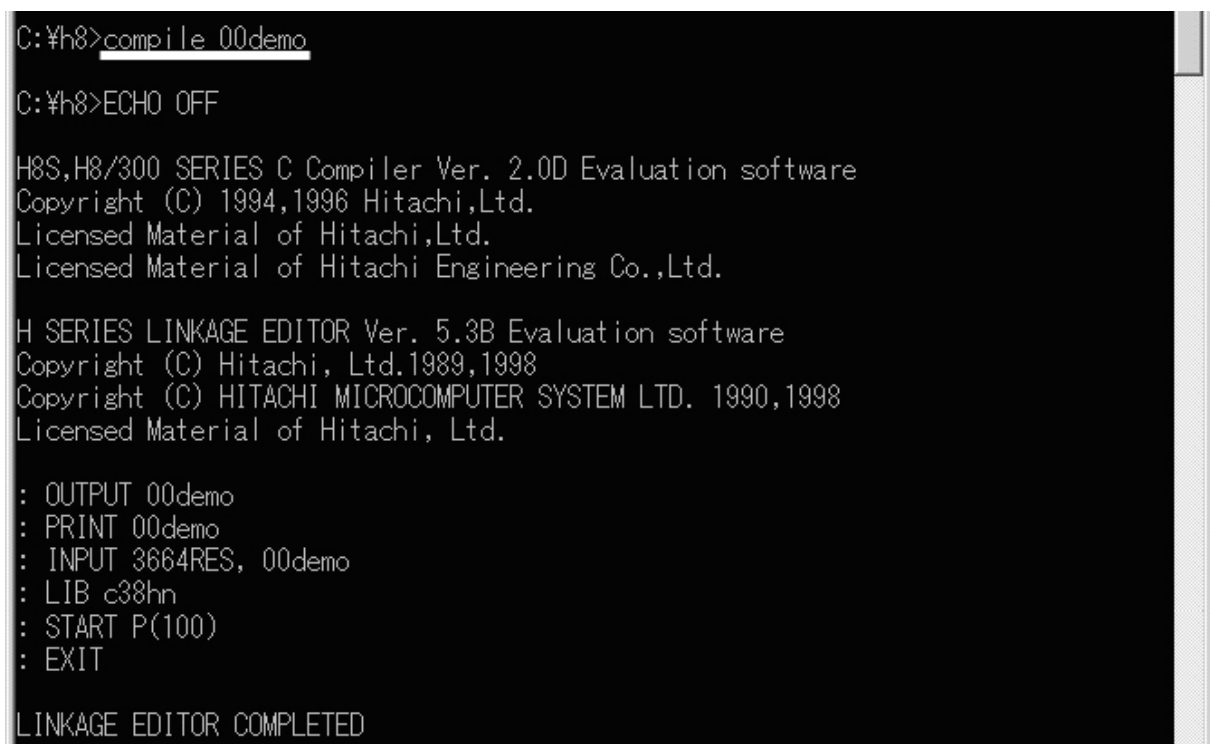
次に、ノートコンピュータから H8 のプログラムの書き込みをします。「スタート」>「すべてのプログラム」>「アクセサリ」>「コマンドプロンプト」を起動してください。



コマンドプロンプト起動後、まず「C:\h8」フォルダに移動してください。「cd ..」で上層のフォルダに移動することができます。



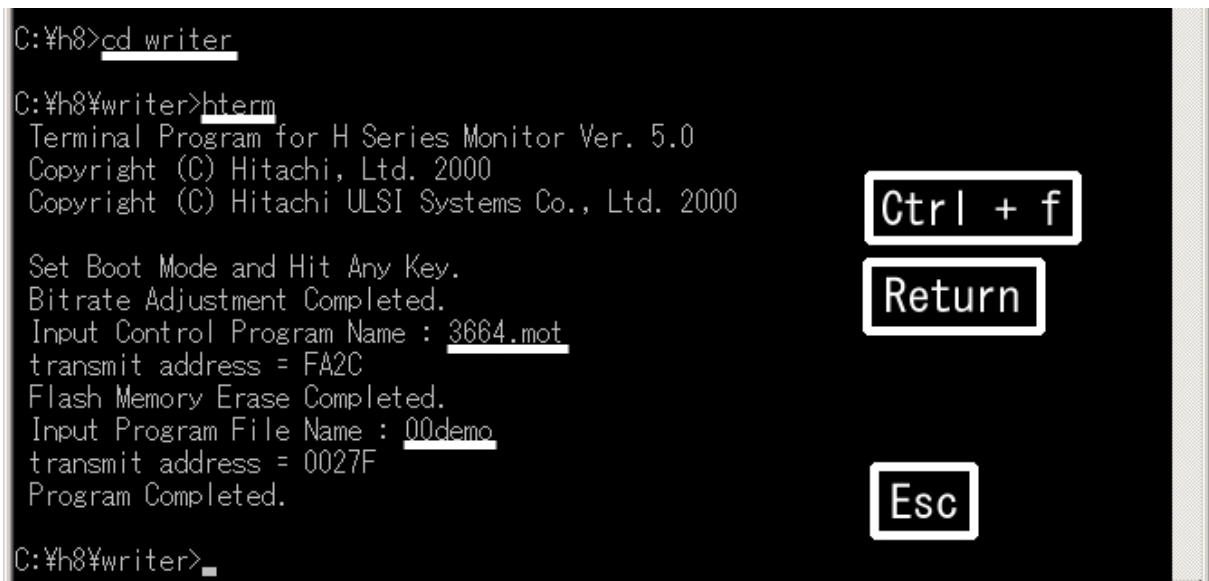
「C:\h8」に移動できたならば、プログラムのコンパイルを行います。ここでは「00demo.c」と「00demo.sub」のコンパイルを行うこととします。コンパイルは「compile 00demo」というように、「compile」をタイプし、スペース後にCファイルおよびSubファイルの名前のみをタイプし、実行することです。無事に「LINKAGE EDITOR COMPLETED」が表示されれば、コンパイル終了です。表示されない場合、間違いのあるプログラムの行番号が表示されますので、改善してください。





H8 プログラムの書き込み方法

次に「C:\h8」フォルダ内にある「writer」フォルダに移動するため、「cd writer」とタイプし、リターンボタンを押してください。そして書き込み用ソフト「hterm」を起動するため、「hterm」タイプし、リターンボタンを押してください。

「Ctrl」ボタンを押しながら「f」を押して「Set Boot Mode and Hit Any key」が表示された後、「Return」ボタンを押してください。そして、「Input Control Program Name:」が表示されたら、「3664.mot」と入力し、リターンを押してください。「Input Program Name:」の表示がありましたら、作成したプログラム名、ここでは「00demo」を入力し、リターンを押してください。プログラムが無事に書き込まれると「Program Completed.」と表示されますので、「Esc」ボタンを押して、シリアル通信を終了し、プログラムの書き込み終了となります。



H8（ハードウェア）のプログラム実行準備

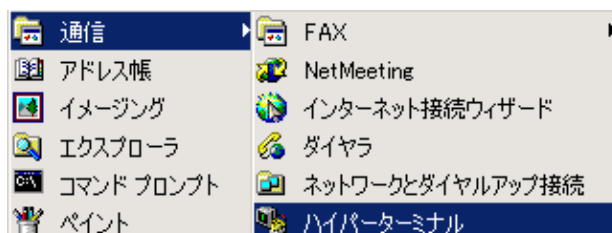
内容	写真
<p>実行手順 1 書き込み完了後、まず電源を OFF して、取り付けられている書き込み用ジャンパを取り外し、かつ、シリアル通信用ポートからはずして下さい。</p>	
<p>実行手順 2 全て取り外した後、電源を ON にすれば、書き込みしたプログラムが実行されます。</p>	

2.9 ハイパーターミナル

本鋼材では制御器の利用を利用して、「センサ値をパソコンに表示」と「キーボードからモータを制御」を行います。そのため本節では、パソコンと制御器のデータ送受信に関する設定方法を紹介します。

Windows 側の設定

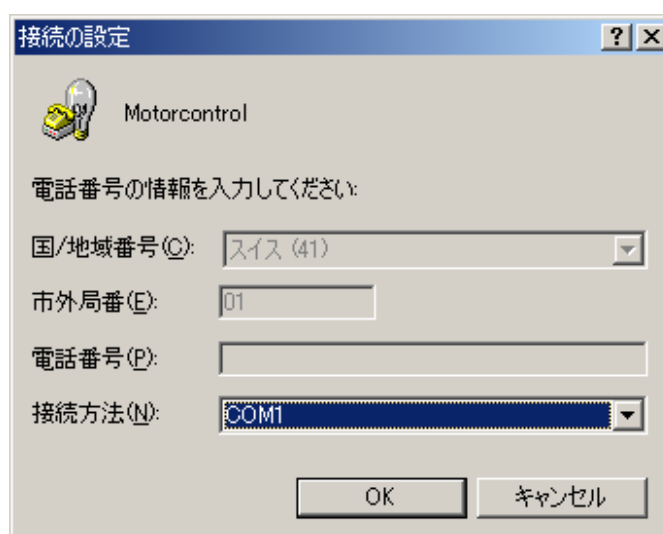
1. Windows に付属しているハイパーターミナル (“プログラム > アクセサリ > 通信 > ハイパーターミナル”) というシリアルコミュニケーション用ソフトウェアを起動しましょう。



2. 名前を入力、アイコンを選んでください。



3. 接続方法を “COM1” にしましょう。



4 . H8 プログラムと同じ設定にしましょう。ここでは「ビット/秒」を 38400 としてください .



5 . 最後に、左から三番目のボタンをして、COM1 を接続しましょう .



H8 側の設定

シリアル通信において、ハイパーターミナル側と H8 側の設定が共通である必要があります。通信速度を設定するのは SCI3.BRR レジスタです。通信速度を変更する場合には、プログラムの SCI3.BRR 値を 12 に設定してください (すなわち、38400 ビット/秒)。

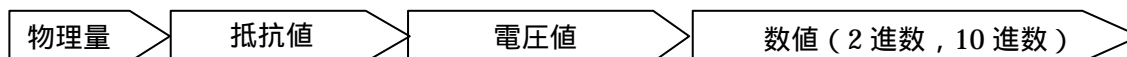
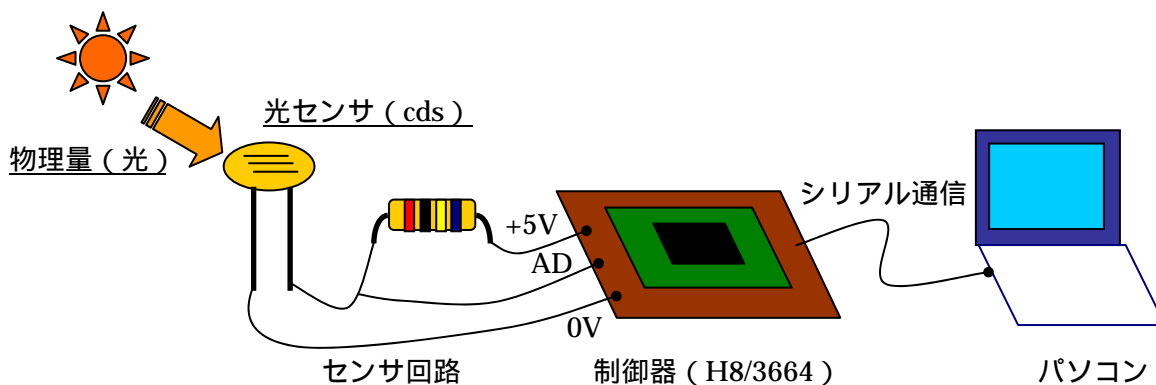
```
void init_serial(void){
    int i;
    IO.PMR1.BIT.TXD = 1;    /* P22 を TXD 端子として使う設定*/
    SCI3.SCR3.BYTE = 0;    /* */
    SCI3.SMR.BYTE = 0;    /* Ascnc, 8bit, NoParity, (Even), stop1 */
    SCI3.BRR = 12;    /* ボーレート設定 38400[bit/s] */
    for (i=0;i<10;i++) ;    /* ダミーウェイト */
    SCI3.SCR3.BIT.RIE = 0; /* 受信割り込み禁止 */
    SCI3.SCR3.BIT.TE = 1; /* 送信許可 */
    SCI3.SCR3.BIT.RE = 1; /* 受信許可 */
    SCI3.SSR.BYTE &= 0x80; /* ステータスクリア (TDRE=1)*/
}
```

3. H8 プログラミング

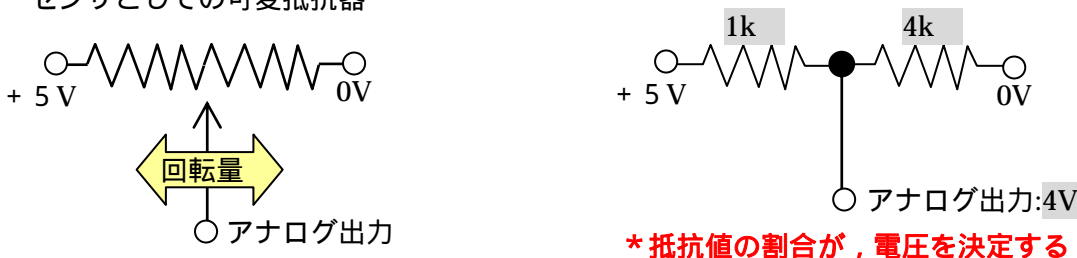
3.1 センサ読み取り方法

分類	説明
可変抵抗型センサ	基本的には、可変抵抗器と同様の方式でセンサとして利用。 例：ポテンシオメータ（角度）、Cds センサ（光）、曲げセンサ、圧力センサ
IC 型アナログ出力型センサ	IC パッケージ化されており、電圧を印加することで物理量に対応したアナログ出力が得られる。 例：加速度センサ、温度センサ、
IC 型デジタル出力型センサ	IC パッケージ化されており、電圧を印加することで物理量に対応したデジタル出力が得られる。 例：エンコーダ、デジタルコンパス

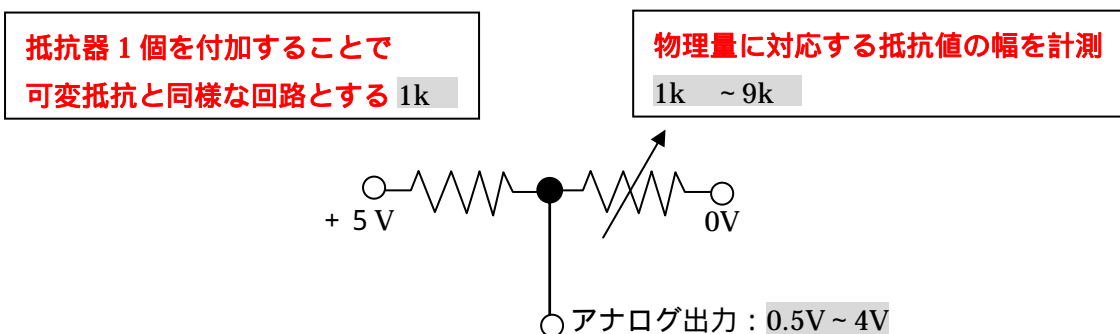
可変抵抗型



センサとしての可変抵抗器



可変抵抗型センサ（曲げセンサ、圧力センサ、光センサ）の場合



(a) 抵抗値の計測

センサは、特定の物理量の変化を電圧に変換するものです。角度センサ、光センサ、曲げセンサ、圧力センサ、加速度センサ、距離センサなど多くの種類があります。これらセンサは、一般的に可変抵抗型と IC 型に分類することができ、可変抵抗型は抵抗での値調整が必要となりますが、共に「入力電圧線」と「グランド線」と「アナログ入力線」を制御器に接続することで、物理状態の変化を読み取ることができます。

ここではまず、マルチメータを利用して可変抵抗型センサの物理量（光センサ：光、曲げセンサ：角度、圧力センサ：力）の変化を読み取ってみましょう。可変抵抗型センサは、物理量の変化により変化する抵抗値を利用するセンサであるので、まずは抵抗値を読み取ります。以下の写真のように接続し、各状態の抵抗値を下表に書き込んでください。



図 マルチメータによる抵抗の計測



図 計測する可変抵抗タイプのセンサ

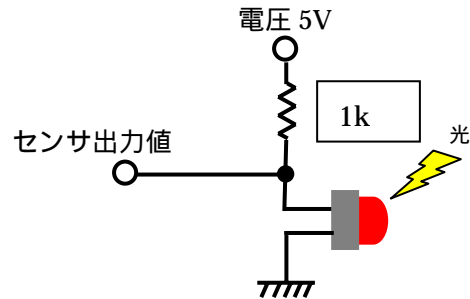
(左：光センサ，中央：曲げセンサ，右：圧力センサ)

センサの種類	状態 1	状態 2	状態 3
光センサ	(暗い)	(通常)	(明るい)
曲げセンサ	(0度)	(45度)	(90度)
圧力センサ	(無負荷)	(普通に押す)	(最大圧力)

(b) 制御器による読み取り方法

ここでは光センサに注目し、制御器による値の読み取り方法を説明します。光センサは、光の強弱、物体の有無や距離を検出するセンサとして使用されています。光の強弱を測る場合、受光素子は光が当たると電流が流れるようになる素子なので、光の強弱がアナログ量に比例して出力されます。物体の有無や、物体の距離を検出する場合、このセンサは発光素子と受光素子で構成され、発光素子から照射した光が物体に当たって反射することで反射量に比例した電流が流れて物体の距離を検出することになります。

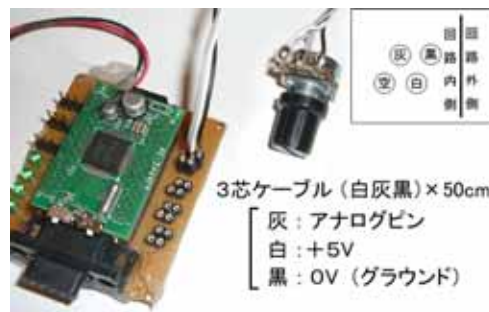
そのため、変化した抵抗値を制御器で読み取るためには、下図のように抵抗（例：1k Ω ）1個を付加し、抵抗値の変化により電圧が変化する回路を作成する必要があります。



【センサの読取における対応関係】			
物理量	抵抗値	電圧値	数値
状態	光センサ抵抗値 [Ω]	センサ出力値 [V]	8bit 値 (0-255) x0, x1, x2, x3
光源に向いている状態	0 - 400	0.00 - 1.45	0 - 74
室内環境	400 - 1000	1.45 - 2.50	74 - 128
やや暗い環境	1k - 5k	2.50 - 4.20	128 - 214
真っ暗な環境	5k -	4.20 - 5.00	214 - 255

上表は、制御器でセンサ値を読み取るための相関表となっております。すなわち、センサの抵抗値が回路により電圧に変換され、また制御器のアナログ入力ピンにより電圧が読み取られ、制御器内では数値として扱われることとなります。

それでは、実際に制御器によるセンサ読み取りを行います。センサの接続に際し、以下の図を参考にして、抵抗とセンサを接続してください。

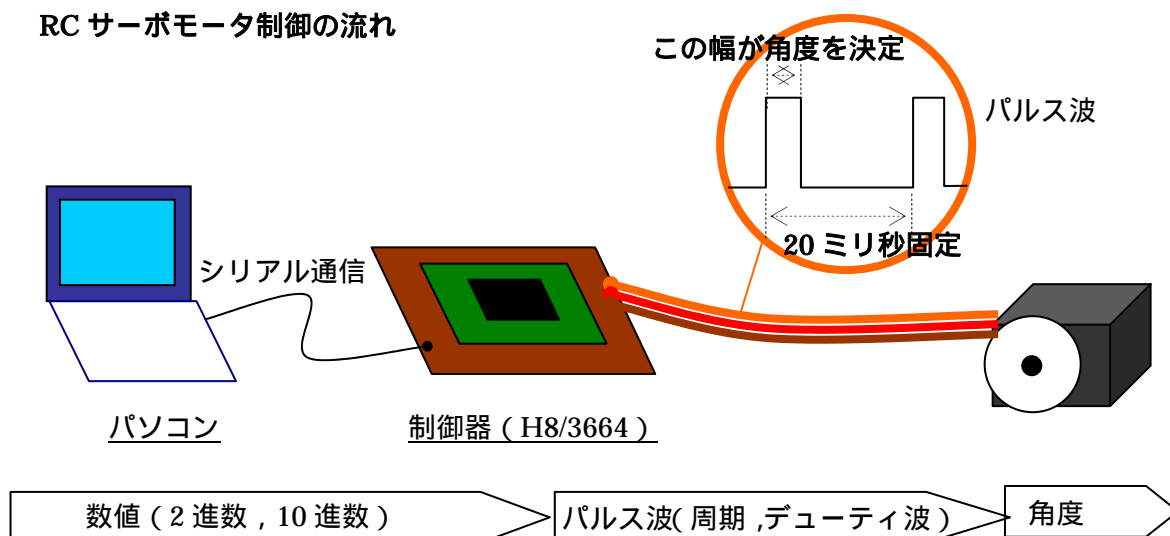


(c) H8 プログラミング : AD 変換

```
main(){
    AD.CSR.BIT.ADST=0; /* AD 変換機能停止 */
    AD.CSR.BIT.SCAN=1; /* 複数チャンネル読み取り */
    AD.CSR.BIT.CKS=1; /* 高速読み取り */
    AD.CSR.BIT.CH=1; /* 読み取りチャンネル指定 */
    while(1){
        AD.CSR.BIT.ADST=1; /* AD 変換機能開始 */
        while(AD.CSR.BIT.ADF==0){}; /* AD 変換中 */
        AD.CSR.BIT.ADF=0; /* AD 変換が終了 */
        AD.CSR.BIT.ADST=0; /* AD 変換機能停止 */
        x0=(AD.DRA>>8); /* AD 0ch の読み取り */
        x1=(AD.DRB>>8); /* AD 1ch の読み取り */
    }
}
```


3.2 RC サーボモータの制御方法

RC サーボモータ制御の流れ



(a) 概要

ラジコン用サーボモータ (RC サーボ) は、DC モータ、減速機構 (ギヤボックス)、サーボアンプ (モータを駆動するための回路) が一体になっていて簡単に扱うことのできるアクチュエータといえます。RC サーボには、アナログサーボ (プラスチックギア・メタルギア)・デジタルサーボなどがあります。デジタルサーボはアナログサーボに比べて出力トルクも高く。ラジコンショップや模型店での入手が可能であり、価格も安く、制御が簡単でとても使いやすいため、様々なロボットで RC サーボが使用されています。



標準型 RC サーボ GWS S03T/2B			
性能		トルク	速度
		4.8V	7.2kg-cm
	6.0V	8.0kg-cm	0.27sec/60°
重量	46 g		
寸法	39.5 × 20.0 × 39.6mm		

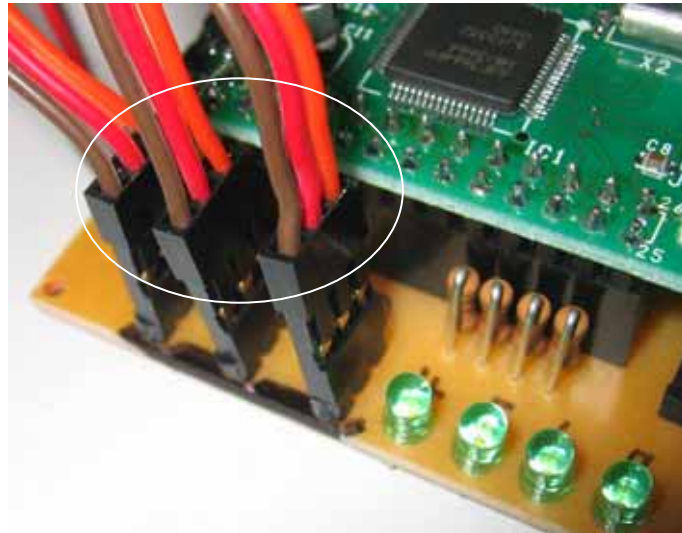
図 RC サーボモータ

RC サーボは、図のケーブル 3 本を制御信号・電源電圧・グラウンド接続して初めて動かすことができます。制御信号とは幅の違う矩形波 (PWM) のことであり、幅によりモータの位置を制御しています。PWM については、次の節で説明します。



橙色: PWM 入力端子
 赤色: RC サーボ用電源
 茶色: グラウンド

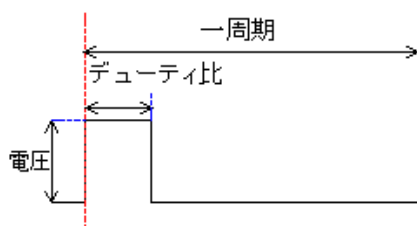
図 RC サーボのコネクタ



**RC サーボを接続するときはコネクタの向きに気をつけること！
茶色の Ground 線が電子回路の外側に！**

(b) RC サーボモータのモータの制御方法 - PWM (パルス幅変調)

パルス変調幅 (PWM) は簡単にいうと、高速でモータ電源の ON/OFF を繰り返すことです。この ON/OFF の繰り返しにはルールがあり、ある一定時間での ON にする時間と OFF にする時間の比を変えてあげるのです。この比をデューティ比といいます。デューティ比が高いほど ON になっている期間が長く、デューティ比がゼロではずっと OFF のままである様子を示しています。ここで重要なのは、先ほど述べた“ある一定時間”、つまり、パルスを与える周期です。何秒おきに ON/OFF を繰り返すか最適に設定しなければなりません。



パルス変調幅	
周期	20msec など。
デューティ比	一周期中の High の占める割合 0 ~ 100%。
電圧	一般的に 5V。

RC サーボは、マイクロチップから送られる PWM 制御信号のパルス幅を読み取り、モータを決められた角度に動かします。つまり、デューティ比がモータの角度に対応しているのです。一般の RC サーボモータにおいて、パルス：3 ~ 5V、周期：20ms、幅：0.9ms ~ 2.1ms (中心 1.5ms) です。RC サーボは一般的に 180° 回転が限界ですので、0.1ms のパルス幅変化が 15° の角度変化に対応しています。

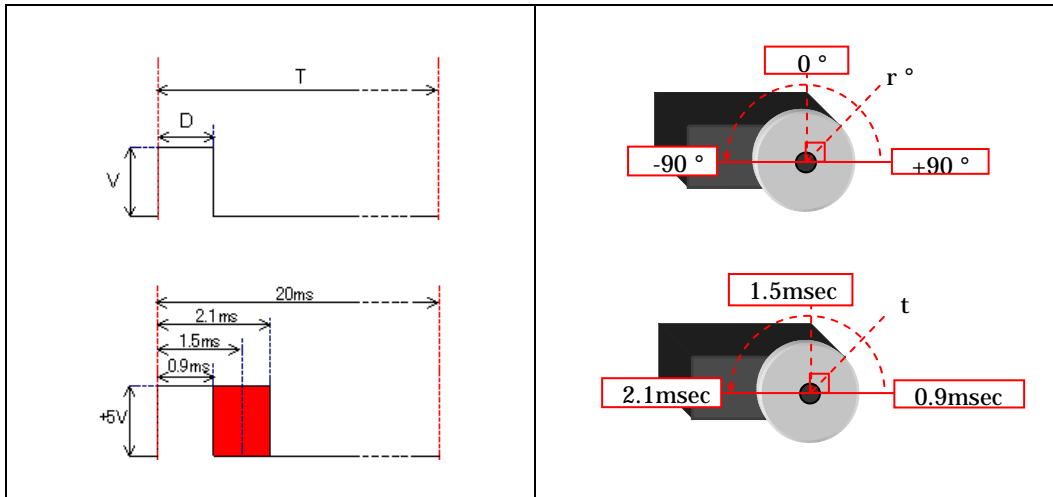


表 RC サーボモータの制御信号

PWM 制御信号					
周期 T	20msec(50Hz) PWM レジスタ GRA = 40000 (16MHz8 周分)				
制御	流れ	始点	中央	終点	
	数値	PWM レジスタ GRB, GRC, GRD	2600-1650	2600	2600+1650
	パルス波	デューティ比 d	4.5%	7.5%	10.5%
		時間 t	0.9ms	1.5ms	2.1ms
物理量	角度 r	-90 °	0 °	90 °	

(c) モータ制御のための H8 プログラム

```

TW.TMRW.BIT.PWMB=1; /* PWMB を設定 */
TW.TMRW.BIT.PWMC=1; /* PWMC を設定 */
TW.TMRW.BIT.PWMD=1; /* PWMD を設定 */
TW.TCRW.BIT.CCLR=1; /* TCNT が GRA でクリア */
TW.TCRW.BIT.CKS=3; /* クロック /8 */
TW.TCRW.BIT.TOB=1; /* PWMB:最初の出力値 */
TW.TCRW.BIT.TOC=1; /* PWMC:最初の出力値 */
TW.TCRW.BIT.TOD=1; /* PWMD:最初の出力値 */

TW.GRA = 40000; /* パルスの周期:20 ミリ秒 c */
TW.GRB = 2600; /* PWMB: モータ Bch の初期角度 */
TW.GRC = 2600; /* PWMC: モータ Cch の初期角度 */
TW.GRD = 2600; /* PWMD: モータ Dch の初期角度 */
TW.TMRW.BIT.CTS=1; /* PWM 開始 */

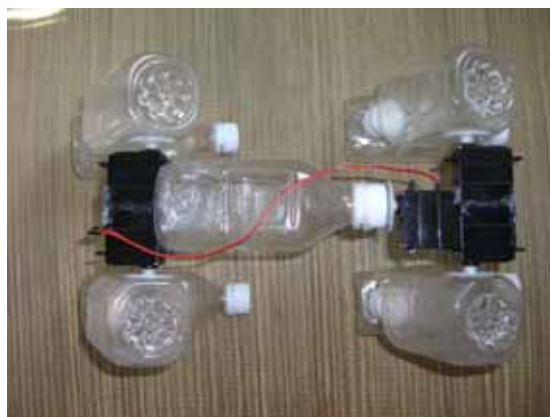
while(1){
    if(TW.TSRW.BIT.IMFA==1){
        TW.TSRW.BIT.IMFA=0;
        TW.GRB=1200; /* モータ Bch の軸角度指定 */
        TW.GRC=1300; /* モータ Cch の軸角度指定 */
        TW.GRD=1400; /* モータ Dch の軸角度指定 */
    }
}
    
```

4. 準受動四脚歩行ロボットの製作法

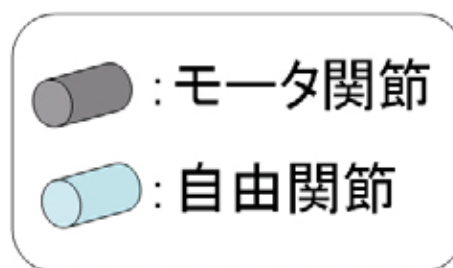
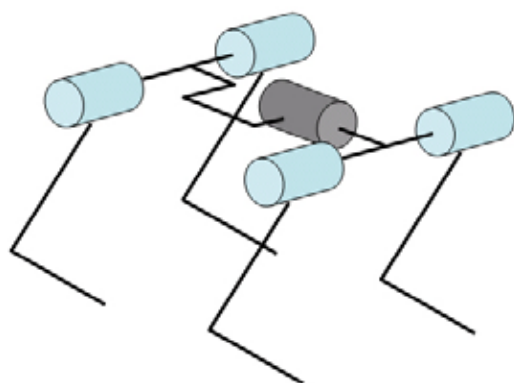
4.1 概要



(a) 側面の概観



(b) 上部からの概観



(c) 自由度図

1. 材料 (1人分)

材料名	数量	備考
四角い 350ml のペットボトル	9	がんばって集めてください。
標準型 RC サーボ GWS S03T/2BB	1	モータ関節となります。 できれば、メタルギアに交換してください
標準型 RC サーボ GWS S03T/2BB	4	自由関節となります。下記の手順で中身を抜いて、自由関節にしてください

2. 自由度関節の作り方.

手順	図解
標準型 RC サーボ GWS S03T/2BB を 4 個準備してください。	A close-up photograph of a standard RC servo motor (GWS S03T/2BB). The servo is black with a white gear housing and red and black wires extending from it.

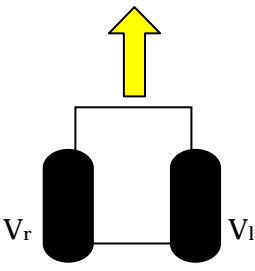
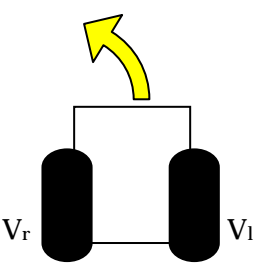
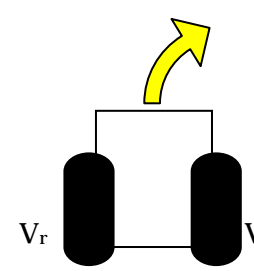
<p>分解してください。図に示した赤領域の部品を取り除き、保存してください。なお、決して無くさないでください。</p>	 <p>点線枠(赤領域)の部品は、無くさないよう保存。</p>
<p>モータ軸となるギアを加工します。角度制限となっている突起部をカッターで削ってください。なお、削り方が甘いと、モータ組立後、自由度関節に引っかかりを感じます。その場合は、再び分解し、削りなおしとなります。</p>	 <p>突起部(点線部) 不要 カッターで突起部削除 完成</p>
<p>右の部品を組み立てます。なお、ギアヘッド(プラスチックの円盤)は、大きいものに交換しておいてください。</p>	 <p>ギアヘッド</p>
<p>ねじは二本のみで構いません。また、緩めに二本のねじを締めてください。きつめに締めますと、スムーズに回転しません。また、引っ掛かりを感じる場合は、再び分解して、モータ軸の突起部を削りなおしてください。</p>	

3. 各自、写真を参考にして四脚受動歩行ロボットを組み立ててください。

あとは、TA と共に光センサ 2 個を使用して、光源探索自律移動ロボットを製作してください。

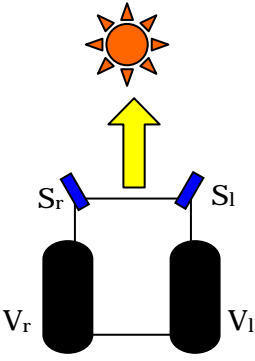
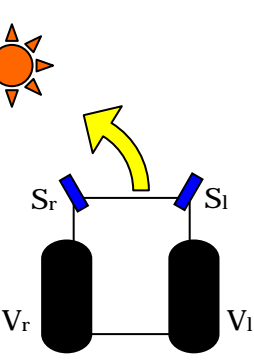
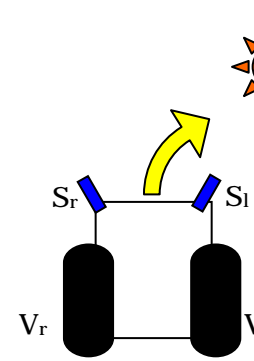
4.2 光源探索の概要

人工生命 (Artificial Life) の分野におけるブライテンバーグビークル .

前進	左旋廻	右旋回
		
$V_r = V_l$	$V_r < V_l$	$V_r > V_l$

【光源探索】

基本システムは，右側のセンサ値の大きさにより右側のタイヤの回転速度が決定され，左側のセンサ値の大きさにより左側のタイヤの回転速度の大きさが決定される．

前進	左旋廻	右旋回
		
If $S_r = S_l$, Then, $V_r = V_l$	If $S_r > S_l$, Then, $V_r < V_l$	If $S_r < S_l$, Then, $V_r < V_l$

【障害物回避】

光センサの代わりに距離センサを使うことで，障害物回避も可能となる．皆さんは室内掃除機ロボットルンバをしていますか？ルンバは，同様な方式で壁との衝突回避を行っています．結果，統計的に部屋全体を掃除できるというわけなのです．このように単純なシステムでも，外見からは複雑な動作を実現しているのです．

<http://www.koj-m.sakura.ne.jp/edutainment/>

最新のものづくり講座の情報は、以下のサイトにアクセスしてください。

<http://www.koj-m.sakura.ne.jp/edutainment/>