

「お父さんとロボットカーを作ろう」

2010年8月22日(日)10:00~13:00

ミュウロボ・PIC16F630・2軸タイプ

キーワード

ハードウェア、ソフトウェア、自律、プログラム、C言語、PICマイコン、センサー、モーター、ブレッドボード、リミットスイッチ、タッチセンサー、CdSセル、迷路脱出、光追跡、ライントレース

はじめに

みなさんは、ロボットと聞いてまずどんなものを想像しますか？

アトムやガンダムのように、空を飛んだり二本足で走ったりするものでしょうか。それとも宇宙開発で活躍するロボットアームのようなものでしょうか。

残念ながら(?) 今日みなさんに作ってもらうのは、それよりもうんと構造の単純な、2輪車のロボットです。だからといってでも「簡単」というわけではありませんので、ぜひ頑張って作ってください！

さてロボットと聞くと、おそらくは頑丈なボディでできた、コツゴツしたものを想像するのではないかと思います。しかしそのような外見、もっといえば手や足や作業する一連のしくみといった機械的な構造は、実はロボットらしさの根本ではありません。

たとえば「左に10メートル直進する」という動きをさせるのに、10個の部品が必要だったとします。しかしやっとできあがったところで、急に「右に10メートルジグザグ走行」をさせたくなったときには、どうしたらいいでしょうか。その場合は、もっと部品の数を増やさなければなりません。そうして前の部品の使えるところそのまま使い、線をつなぎ変えたり、違う部品と取りかえて、ようやく20個の部品でできあがったとします。しかし次にまた別のことがしたくなったら、どうしたらいいのでしょうか？ その時もやはり一から作り直さなくてはなりません。

このようにその都度工夫しながら目的にあったものに作りかえることは、楽しい作業だともいえるのですが、しかし多くの時間と部品(つまりお金)が必要になることはまちがいありません。何より、せっかく前に作ったものをまたなくしてしまうことは、もったいないと思えるでしょう。

そこで私たち人間の体を見てみましょう。子供から大人まで成長によって大きさが変わることはありますが、基本的な形はほとんど同じです。それなのに同じ「モノ」を使って、走ろうと思えばすぐにでも走れるし、ジャンプしようと思えばジャンプできます。一つ一つ違う動作をするために、いちいちそれ専用で体のどこかを取りかえたり、作り直すということはしません。それなのに色々なことができてしまうのは、なぜでしょうか。

そうです。それは脳の命令によってなしとげているからです。右手でボールをつかみかかったらそのような命令を脳から送れば、体はそのようにとっさに動いてくれます。これは決して体のどこかを「機械的に」変更して実現しているわけではありません。さっきまでと同じ体を使いながら、その「中身の命令」を変えることによってさまざまな動きを実現しているのです。

この人間(生物)と同じ「脳からの命令によって体を動かす」ということをまねるのが、「ロボットらしい」と言えることなのです。

そこで、今日のキーワードは「ハードウェア」と「ソフトウェア」です。この言葉はすでに何度か耳にしたことがあるかもしれませんが、上のたとえでいえば、脳の命令や、脳にあらかじめ入っている情報がソフトウェア、あとの全部(手足やそこに脳からの命令を伝える神経など)はみんなハードウェアです。細かく言えば、脳も実はハードウェアの一つで、それはマイコンやパソコンといわれる部分に代表されます。そして脳の命令のことを「プログラム」といいます。

このプログラムを書きかえることで、ハードウェアの変更では実現がなかなか大変だった動作の微調整、あるいは大がかりな変更ができてしまうというのが、ソフトウェアの特徴です。(ソフトとは「柔らかい」という意味ですが、それは「変更しやすい」からです。またハードは「変更しにくい」(それは良い意味でもあります)ことが「固い」と言われるゆえんです)

ぜひそれがどういうことなのかを、今日はロボットを通して体験しましょう。そしてぜひお家に帰ったあとも、各自で研究を続けて、さまざまな発見をしていってください。

今日の段取り

箇条書きにほとんど網羅していますので、順番に読んで、できるところはどんどん進めていってください。ただし途中で全体に向かって説明する時があると思いますので、その時はいったん手を止めてください。

10:00～10:20 はじめに

- 1) 最初はパソコンや部品には触らず、講師からの話をしっかり聞いて下さい。
- 2) ミュウロボでどんなことができるのか、デモンストレーションをします。
- 3) また以下の作業では、ロボット工作は極力自力で解決することを心がけてください。(書いていないことでも試行錯誤で解決できることがほとんどです。その時、わざとこわすのはいけません、線をつなぎ間違えることはよくあることです。そうやってやむをえずに、ある程度はこわしながら、それぞれの部品の特性を覚えていくものです。)
- 4) もしこわれても、ほとんどの場合ははんだ付けや部品の交換で元通り直りますので、あまりこわがりすぎず、でも基本は1つ1つ部品を大事にしながら楽しく作って下さい。

10:20～11:20 ミュウロボの製作 (本体=ハードウェア)

- 1) 付属の説明書をよく見て、各家庭で一台作ります。
- 2) 特にギアボックスが一番時間がかかると思います (小学生だけの場合は30分以上かかるのが普通です。ただし今回はお父さんと協力して作って下さい)。またグリスはほどほどにぬって下さい (ぬりすぎはほこりがつきやすくなり、回転がかえって悪くなります)。
- 3) モーターに、ピニオンという小さなギアを入れる時には、モーターの上に少し出ている軸が飛び出さないように、平行なかたいものでその部分を押さえながら圧入します。この作業に限り、講師に確認を取って下さい。
- 4) 次に時間がかかるのが基板からのギアボックスへの配線です。しかしこれはあまり神経質にならずとも、あとでパソコンからのリモートコントロールで動作を確認・修正ができます。
- 5) 基板の本体への取り付けは、一度声をかけてください。スペーサーと呼ばれる部品を、あとで基板の位置の変更がしやすいものと交換します。
- 6) 基板は力をかけると割れる (中の細かな線が切れる) ことがあります。物の下じきにしないように、十分気をつけて下さい。
- 7) 配線の色は赤が+ (Vcc)、黒が- (Grand) です。+-の配線、電池の向きにはくれぐれも注意して下さい。部品が熱くなったり変な音がしたら、すぐに電源を切って、もう一度配線を確認して下さい。
よくあるのがスイッチ1 (in1) と電源コードの間違いです。
- 8) あとでセンサーを汎用的に取り付けられるように、ブレッドボードというものを取り付けます。
- 9) 触覚センサー、Cds セルというセンサーもついでに作っておきます。ブレッドボードと基板への配線の仕方は、あとでプログラムの時に説明します。

11:20～11:40 動作確認(トラブルシューティング)

- 1) 2家庭で1台のパソコンを使います。myubasic を立ち上げ、remote を選びます。
- 2) ロボットと赤い線をつなぐと、パソコンのパネル上からリモートコントロールができるようになります。前進・後退・左前(左のタイヤが前進)・右前・左回り(左に向きを変える)・右回りができるか動作確認して下さい。
(モーター3は今はないので動きません。)
- 3) 動かない場合は、remote のすぐ下にある白い欄を確認して下さい。そこがもし空欄の場合は、講師を呼んで下さい。COM1～6(ポート番号といいます)が表示されていれば、そのどれかを選び直して再びリモートコントロールを実行して下さい。
- 4) 正確なCOM番号は、パソコンのタスクマネージャーからシリアルポートを確認することわかります。ミュウロボの9ピンのケーブルをパソコンのRS232C端子に、あるいはUSBシリアル変換ケーブルをつないでおかないと表示されないので注意が必要です。
- 5) リモートコントロールで動くけれども指示通りの動きでない場合は、配線を疑って下さい。モーターにつないだ青と白の線を上下、左右に入れかえるとおそらくうまくいきます。
- 6) それでも動きがおかしかったら、たとえば片方のタイヤだけ正確に動く場合は、片側の線がはずれていることがあります。またボックス自体のねじの締めつけが強すぎてその部分でモーターの動きがブロックされていることもあります。その場合はねじを少しゆるめて下さい。またピニオンのかみ合わせがおかしい場合もあり、その場合はモーターを交換して直るかどうか見てみます。

11:40～12:10 プログラムの製作・初級(ソフトウェア)

- 1) 2家庭で1台のパソコンを使います。myubasic を2つ立ち上げます。両家庭で間違わないようにそれぞれ管理してください。(難しいようでしたら2家庭で協力して1つのプログラムを作ってもかまいません。それはお任せします)
- 2) プログラムの書き方とプログラムの転送のしくみ(myubasic の使い方)を説明します。この時はいったん手を止めて下さい。
- 3) 最初のプログラムは、前進・後退など動きの組み合わせで作って下さい。
「前進 10」は「前進を1秒間する」というプログラムです。
- 4) 模造紙にかかれた「1面」(直進してゴール)をクリアしましょう。

12:10~13:00 センサーの実装 (ハードウェア) +プログラムの製作・中級 (ソフトウェア)

- 1) センサーとはロボットの感覚であり、スイッチと同じ役割をします。
- 2) センサー→ブレッドボード→基板の配線の仕方を説明します。(別紙のフリーハンド参照)。
- 3) in(1)、in(2)、in(3)はアナログセンサー用、in(4)はデジタルセンサー用です。
- 4) ミュウロボでのアナログとは、計測値が0~255まで連続的に変化するものことで、Cdsや赤外線センサーがその代表です。デジタルとは、計測値がオン(in() $=$ 0)かオフ(in() $=$ 1)かになるもの(この間があるのがアナログ)のことで、リミットスイッチや触覚センサーがそれにあたります。
両方をうまく使えば、それに見合ったプログラムでロボットを賢くできます。
- 5) do loop文、for next文、if endif文をおさえれば、たいいていのできるようになります(中学生以上)。
細かなテクニックは、別紙のプログラム編を参照して下さい。時間の都合で今回は割愛することになるかと思いますが、時間の許す限りフォローいたします。
- 6) 「1面」を触覚センサーでクリアするプログラムを作りましょう。(今日のメイン)
- 7) 最初なので、とっかかりになるヒント(させたいことをどの命令文を使ったら実現できそうか)は、遠慮なく講師に聞いて下さい。

その他の課題

時間のゆるす限り、センサーとプログラムを駆使して、新しい課題に挑戦しましょう。

- 1) 触覚センサーで、正面からぶつかった時にUターンできるプログラムを作ってみよう。

*ここからはおまけの部分です

- 2) -1 Cdsセルで、光(懐中電灯)を真っ直ぐ追跡する。→2面(直進)
- 2) -2 できたら、今度は光が当たると停止するプログラムを作ってみよう。
- 3) -1 Cdsセル2つで、光のある方向に曲がりながら光を追跡する。→3面(Uターン)
- 3) -2 できたら、今度は光から逃げる(あるいは停止する)プログラムを作ってみよう。

- 4) Cdsセル2つで、黒線をライントレース。
2面(直進)
3面(Uターン)
4面(直角カーブ)
5面(8の字)

…ほかにも、なみ線、ヘアピンカーブ、線を細くするといった課題も難しくて挑戦しがいがあります。

- 5) 3つ目のモーター(3軸)を取り付けたら、どんなことができるかを考えてみよう。

プログラム編

コンパイル

- ・ コンパイルとは、編集画面で作成したプログラムを、ミュウロボの PIC マイコンが理解してくれる言葉に翻訳する作業のことです。
- ・ ボタンを押して、もしエラーが出れば、プログラムのどこかが間違えています。
- ・ プログラムは、必ずその都度コンパイルをしてからロボットに転送してください。

プログラム

- ・ 大文字、小文字でもかまいません。
- ・ C 言語だと行の最後に「;」(セミコロン) がいつも必要ですが、このミュウロボの場合は必要ありません。また命令に全角の日本語があっても大丈夫です。
- ・ 命令のあとに、半角スペースを入れてから、数字を書き込み、その命令を何秒続けるかを指定します。
- ・ 数字は 1 につき、実際の 0.1 秒です。
- ・ たとえば、

```
前進 10  
後退 100
```

とすれば、前進を 1 秒したあと、後退を 10 秒します。for 文で囲めば、それを繰り返させることができます。また if 文で囲めば、その条件のときだけそれを実行させるようにできます。

例 1 2 重の if 文 59 バイト

```
do  
  if in(1)=0 then  
    前進 10  
    if in(2)=0 then  
      モーター 3 左 10  
    else  
      前進 10  
    endif  
  else  
    後退 10  
  endif  
Loop
```

例 2 2 重の for 文 63 バイト

```
byte n m  
for n=1 to 5  
  前進 10  
  for m=1 to 5  
    モーター 3 左 10  
    wait 10  
    buzzer 10  
  next  
後退 10  
next
```

例 3 for 文+if 文 76 バイト

```
byte n  
for n=1 to 5  
  前進 10  
  if in(1)=0 then  
    モーター 3 左 10  
  elseif in(2)=0 then  
    モーター 3 右 10  
  else  
    後退 10  
  endif  
next
```

* ウェブ上にあるテキストファイルをコピーして、そのまま編集ボックスに貼り付けてもコンパイルできます。

1) do loop 文

```
do
  繰り返したい動作
loop
```

左のように書きます。一番単純なプログラムです。
do ではじまり、loop でとじます。その間にあることを無限回繰り返します。

ただし無限回だと使い勝手が悪いときもあります。その場合は、あとに出てくる if 文と breakP という命令を使って、

```
do
  繰り返したい動作
  if 条件 then
    breakP
  endif
loop
```

とすれば、do loop を抜けることができます。

あるいは回数を決めて for 文で作るのが良いでしょう(このあと出てきます)

2) for 文

```
byte n
For n=1 to 5
  繰り返したい動作
next
```

まず文の最初に、byte という命令で、繰り返し回数に使う定数を言っておきます。こうしないと、「n って何？」と聞かれてエラーになってしまいます。PIC16F630 では byte という命令ですが、PIC16F688 や 88 ではかわりに int (インテジャー=整数) という命令が使えます (int は 630 では使えません)。

同じことを 5 回繰り返したいときは、for の横に n=1 to 5 と書きます。これは n=1 から n=5 までという意味です。

そして next で文をとじます。

```
byte n m
for n=1 to 5
  buzzer 1
  for m=1to10
    後退 10
    右回り 30
    前進 20
  next
next
```

for 文の中に for 文を入れると、一番中にある for 文を繰り返したあと、それ自体をさらに外側の for 文が繰り返します。

この場合、定数を 2 つ宣言します。(それぞれ半角あけます)

左のようにすることを「入れ子」といいます。このような書き方をマスターすることが、プログラムができるようになる早道です。

左のプログラムでは、「後退 1 秒・右回り 3 秒・前進 2 秒を 10 回繰り返す」ことを 1 セットとして、そのたびにブザーを鳴らして 5 セット繰り返します。

また中の for 文の next をよく付け忘れるので、気をつけます。

3) If 文

「もしも～なら…する」という命令です。ロボットをインテリジェンス化（かしこくする）ために一番よく使われるプログラムです。これがわかればたいいのことをロボットにさせることができるようになります。
if ではじまり、endif でとじます。

in はセンサー入力のことです。ミュウロボには 1～4 つまで入力があり、ただしアナログ入力は in(1)～in(3)、in(4)だけはデジタル入力（オン/オフだけ感知）です。

A or B（AかBである場合）、A and B（AでありしかもBである場合）ということもできます。

```
if in(1)=1 or in(2)=1 then
  前進 10
endif
```

```
if in(1)=0 or in(2)=0 then
  前進 10
endif
```

注意: If 文は **then** を忘れないようにします。

左: センサー 1 かセンサー 2 がオンなら、前進 1 秒せよ。

右: センサー 1 がオフであり、しかもセンサー 2 もオフなら、前進 1 秒せよ。

```
if AN(1)>100 then
  前進 10
elseif AN(1)>50 then
  後退 10
elseif AN(1)>0 then
  停止 10
endif
```

```
if AN(1)>100 then
  前進 10
elseif AN(1)>50 then
  後退 10
else
  停止 10
endif
```

* AN(1)はアナログ入力(1)の値 (0～255) のことです。

場合わけがたくさんあるときは、左のように **elseif** をいくつも使って書きます。もし条件が「それ以外」という場合は、**elseif…then** を省いて、**else** とします。**endif** を忘れないようにしましょう。

また if 文も for 文のように 2 重の入れ子にすることができます。前のページの例 1 を参考にしてください。

4) 入力文について

If in(1)=0 then…next のかわり、iflo(1) then…next の命令を使うと、容量 (バイト) を節約できます。lo はスイッチが入っていない状態。

hi はスイッチが入っている状態です。

たとえば入力 1 について場合分けしたい時は、

```
iflo(1) then //入力 1 が切れている場合
ifhi(1) then //入力 1 が入っている場合
next
```

と書きます。これを使うと、たとえば先の例 1 のプログラムは 59 バイトでしたが、同じことを 35 バイトでさせることができます。

例 4 例 1 と同じ命令 35 バイト

```
do
  iflo(1) then
    前進 10
  iflo(2) then
    モーター 3 左 10
  else
    前進 10
  endif
else
  後退 10
endif
Loop
```

<補足>

PIC (ピック : ワンチップマイコン)

- 16F630、16F688 は 14本の足があります。16F88 は 20本。
- 足が多いとそれだけ、センサー (入力) やモーター (出力) をコントロールすることができます。
- 630 と 688 はアナログセンサー 3つ、デジタルセンサー 1つ、モーター 3つ。
- 88 はアナログセンサー 5つ、デジタルセンサー 1つ、モーター 4つ。

ファームウェア

- myubasic などで作って (EEPROM に) 転送したプログラムを実行するためのベースプログラムです。
- ミュウロボ基板の PIC16F630 には、すでに PICライターを使ってファームウェアが書き込まれています。
- PIC16F630 自体は 150 円程度です。ただしそのままだと使えず、PICライターを使ってファームウェアを書き込む必要があります。

EEPROM (不揮発メモリ)

- 電源スイッチを切ってもプログラムをおぼえています。
- だいたい 1つの命令が 2 バイト。for 文、if 文は約 30 バイトです。
 - 16F630 は、128 バイト (「2重の for 文・if 文」×2つか、「3重の for 文+if 文」×1つ程度)
 - 16F688 は、256 バイト (「3重の for 文+if 文」×2つ程度)
 - 16F88 は、2048 バイト (まず容量が不足することはありません)

<630 からのアップグレード>

- プログラム容量に不足してきてから、688 に切り替えるのも十分いいと思います。
- スタジオミュウにメールで注文し、「パワーアップキット」というものを購入し (送料込み 800 円)、基板の 630 と取り替えます。慎重に引き抜かないと足が折れたり、黒い部分が割れたりするので注意します。また向きがあるので (丸い切っ欠きがある) 間違えないようにします。
- 688 はセンサーの値を計測できます。
- 88 は基板が専用になるので、基板ごと新しいものを購入してください。
(マルツパーツ、ツクモロボット王国などで扱っています)

<モーターの追加>

- スタジオミュウから直接「モータードライバーIC」というものを購入します (送料別? 315 円)。688 と一緒に買うのがいいと思います。
スタジオミュウ <http://www.studiomyu.com/index.html>

<家で続けてロボットを研究する人のために>

まず、おうちの人と相談して、パソコンを使わせてもらう許可を得てください（パソコンがないとプログラムが作れません）。

myubasic など必要な環境のソフトをまとめた DVD をお渡ししますので、「プログラムの開発環境」のフォルダからコピーしてそれを家のパソコンでも再現して下さい。他のおまけも必要であれば使ってください。（rs232C ドライバーとは、USB シリアル変換ケーブルのドライバーのことです。普通に買えばドライバーの CD がついてきますが、もしなくしたらこれを使って下さい）

プログラムを開発する環境は今日使った myubasic（C 言語）のほかに、myurobo（BASIC 言語）、ドリトル（日本語）、エクセルの計 4 種類があります。myubasic か exel が一番将来的に汎用性が高くて意味があると思うのでおススメです。

私が作ったサンプルのプログラムも入っています。研究の参考にして下さい。ただし myubasic 用（windows）です。myubasic を setup した後、myubasic があったのと同じフォルダから、「LED」「ライントレース」「光センサーによる…」「接触センサーによる…」の 4 つのフォルダを新しく myubasic のプログラムを保存する場所に移動させてください。

パソコンに mac を使う人は、「プログラムの開発環境（mac）」から、以下の手順に従ってください。

- 1 Scratch のダウンロードサイトから、Mac 版の Scratch をダウンロードしてインストールする。
- 2 ScratchMyuRobo.zip をダウンロードして、インストールした Scratch フォルダ内に展開する。
#このとき、インストールした Scratch も同時に使いたい場合は、フォルダごと別名でコピーする。（今回は ScratchMyuRobo とした）
- 3 展開した ScratchMyuRobo.zip 内の「locale」「Projects」フォルダの中身を、インストールした Scratch フォルダ内のそれぞれのフォルダに移動する。
- 4 実行ファイルである「Scratch」を右クリックして「パッケージの内容を表示」を選択し、「Contents」フォルダ内にある「Info.plist」を、Property List Editor で開く。
#ダブルクリックして開けるはず。ダメならテキストエディタで開いてもよい。
- 5 「Root」をクリックすると、リストが表示されるので「SqueakImageName」という項目が「Scratch.image」となっているところを「ScratchMyuRobo.image」に直す。
#「Dump」ボタンを押して、テキストエディタ風の編集画面で編集することもできる。
- 6 上書き保存して Property List Editor を終了する。

（<http://4oc.blogspot.com/2009/10/macscratchmyurobo.html> というサイトを参考にしましたが、私は動作確認にはしていません）

最近のパソコンでは RS232C 端子がなく、USB 端子だけというのがほとんどと思われていますが、その場合は USB シリアル変換ケーブル（パラレルではない）というものを購入してください。（これによってパソコンからミュウロボへプログラムを転送しますので、これも必要です）

秋月電子通商というところで 950 円（おそらく一番安い価格）で売っています。



プログラムは、スタジオミュウのページ <http://www.studiomyu.com/index.html> を丹念に研究してください。

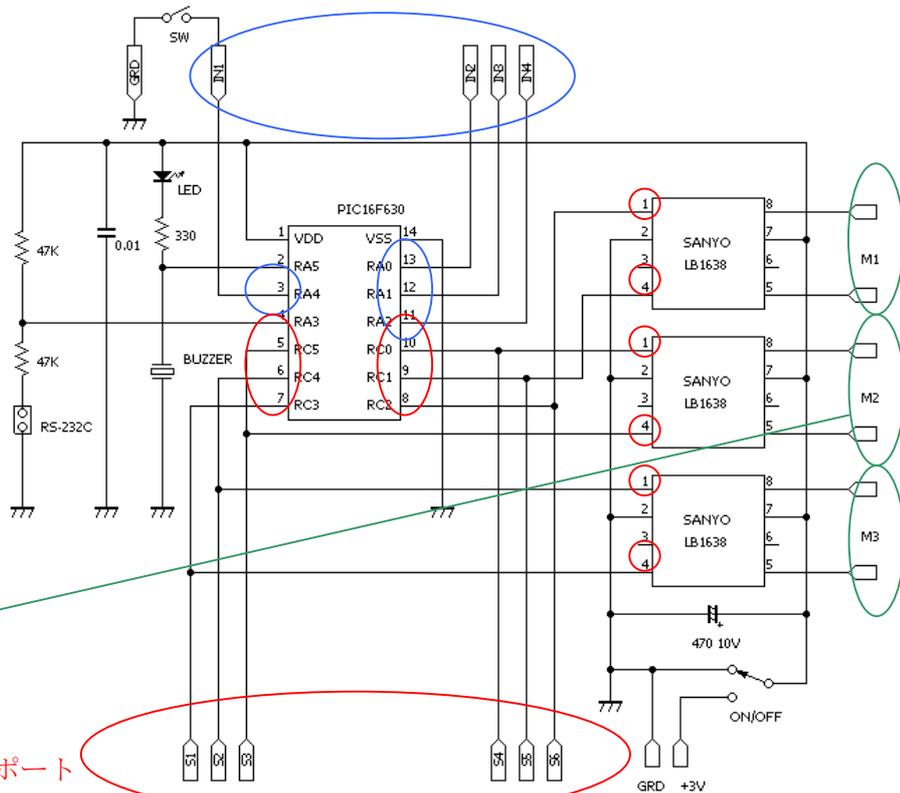
センサーには、Cdsのような in() の入力（内側+）と出力（外側 Grand）を2つだけの端子のものがありますが、赤外線センサーのように3つ端子（普通は赤黒白）のあるものもあります。例で言うと、

- +（赤） ミュウロボの+電源か、PICの左側に並んでいるS6などの真ん中の穴へ
- （黒：Grand） ミュウロボの-電源か in() の外側へ
- 信号（白） in() の内側へ

となります。このあたりは、基板の回路図を理解することが一番早道です。

<PIC16F630>

- 動作電圧 2.0V-5.5V
- 内部クロック 4MHz
- タイマー1 125KHz
- RA ポートA
- RC ポートC
- <入力ポート>**
- RA4 In1(リミットスイッチ)
- RA3 シリアル信号入力用
- RA2 In4
- RA1 In3
- RA0 In2
- RA0-RA2, RA4はプルアップされている
- <出力ポート>**
- RA5 LED と圧電ブザー用
- RC5 S3, M2R
- RC4 S2, M3L
- RC3 S1, M3R
- RC2 S6, M1L
- RC1 S5, M1R
- RC0 S4, M2L



モーター

出力ポート

入力ポート

<上の図のまとめ>

基板のセンサ入力ポート（2連の穴）

（詳しくは上の表の線を追いかけて確認下さい）

in1~4 外側（in1は黒線がつながっている方） grand（共通）

注意：外側の穴は全部つながっています。そのため in1~4 の外側の穴のどれか1つが電源-とつながっていればいいです。

- in1 赤線がつながっている方 信号+→マイコン 3番ピン→起動のリミットスイッチ（アナログスイッチ）1
- in2 内側 信号+→マイコン 13番ピン→アナログスイッチ 2
- in3 内側 信号+→マイコン 12番ピン→アナログスイッチ 3
- in4 内側 信号+→マイコン 11番ピン→デジタルスイッチ（Cdsは使えません）

注意：in0の左右の穴が導通すれば（極端な例では直接線をつなげば）スイッチが入ったことになります。（プログラムではスイッチが入ったことを in0=0 か Lo0と書きます）

また今はマイコンのピンとの対応は関係ありませんが、もし基板の in0の穴が壊れた時でも、その穴に通じている線をたどっていき、つながっているマイコンのピンにじかに半田付けすれば同じです。

基板の（サーボ）モーター出力ポート（3連の穴）

S1~S6（共通）
外側 Grand（=電源-）
真中 電源+

注意：外側、真ん中の穴はそれぞれつながっています。そのため S1~S6 のどれかが電源-、電源+それぞれにつながっていればいいです。

- S1 内側 信号+→モータードライバ M3の4番ピンとマイコン7番ピン→モーター3右駆動
- S2 内側 信号+→モータードライバ M3の1番ピンとマイコン6番ピン→モーター3左駆動
- S3 内側 信号+→モータードライバ M2の4番ピンとマイコン5番ピン→モーター2右駆動
- S4 内側 信号+→モータードライバ M2の1番ピンとマイコン10番ピン→モーター2左駆動
- S5 内側 信号+→モータードライバ M1の4番ピンとマイコン9番ピン→モーター1右駆動
- S6 内側 信号+→モータードライバ M1の1番ピンとマイコン8番ピン→モーター1左駆動

注意：モータードライバとマイコンのピンの数え方は、切り欠き左を1番として反時計回りです。

<パーツ>

- ・ブレッドボード 秋月で 150 円。半田付けがいらず、大変便利です。
- ・光センサー 可視光…Cds 小さいのは秋月で 50 円。大きいのは 100 円～300 円します。
赤外線…フトリフレクタ 他にも名称の違いで、反射型フォトセンサー、フォトカプラ、フォトインタラプタなどがあります。ロームの RPR220 (マルツパーツ) が最も使いやすいです。ミュウロボの 525 円の赤外線センサーもこの RPR220 を使ったモジュールです。
- ・音センサー ECM マイクロフォン 秋月で 4 個 200 円
- ・LED 秋月で 100 本 500 円ぐらい。色や適正電流 (電圧) の違いによって各種あります。赤外線 LED、紫外線 LED もあります。
- ・赤外線焦点センサー 体温のある動体を検知します。やや高価でキットで 800 円ぐらい。
- ・リミットスイッチ 接触センサーとして使えます。壁の検知に。各種あります。100 円ぐらい。

- ・ミュウロボキット…マルツパーツ (ネット or 寺町店)
 - 688 タイプ基板 (2 モーター/3 モーター) 2362 円、2551 円
 - 688 タイプキット (2 モーター/3 モーター) 3213 円、3874 円
 - 赤外線センサー 472 円
 - リミットスイッチ 141 円
 - 触覚センサー 170 円
 - ベースキット 2929 円 (ただし内容はタミヤ製品と同等。うちリモコンが 1000 円相当します。プログラムで制御してリモコンがいない人にはその分がもったいないので、タミヤ製品で揃えることをお勧めします。逆にリモコンがほしい人は、別売り (マルツで 1001 円) で買うよりは、ベースキットを買った方がお得です)

- ・ミュウロボ上位基板・モータードライバーIC などの部品…スタジオミュウ (電子メールで注文)
 - PIC16F688 (ファームウェア書き込み済み) 800 円
 - ミュウロボ 2 基板 (2 モーター/3 モーター/4 モーター) 2835 円、3045 円、3255 円
 - モータードライバーIC 315 円

<お店>

マルツパーツ <https://www.marutsu.co.jp/user/index.php>

寺町通りにもお店 (マルツ電子) があります。しかしインターネットで注文するのが便利です。ミュウロボのキットと工具はここが安いです。電子部品は linkman などの特売品以外は秋月の方が安いです。配達融通 (日時指定) が利くのと、5000 円以上は送料無料です。

秋月電子通商 <http://akizukidenshi.com/catalog/default.aspx>

昔からある有名なお店で、電子部品が最も安いです。センサー、LED、抵抗、コンデンサ、各種マイコンなど、回路の作り方を知っている人にとっては格安です。RS232C 端子につなぐ USB シリアル変換ケーブルもここで買うのが安いです。ただし配達融通 (日時指定) が利かないのが難点です。

ツクモロボット王国 <http://robot.tsukumo.co.jp/>

モーターなどロボットに必要な部品で、他のお店にないものがそろいます。機械部品 (特にタミヤ楽しい工作シリーズ) が安い。ただし目が飛び出るような高価なものも。ミュウロボのキットも (定価ですが) 扱っています。

ジョーシン (J & P) <http://joshinweb.jp/hobby/top.html>

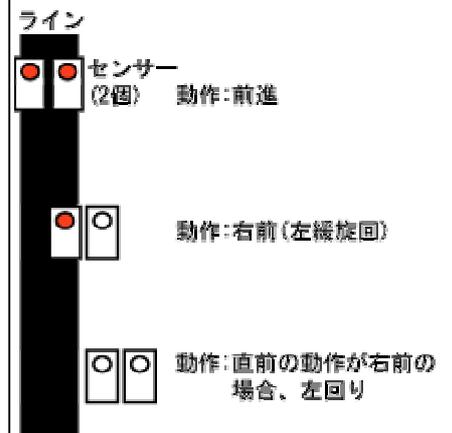
タミヤの製品はツクモと同じく安い。ユニバーサルアーム、ロングユニバーサルアーム、ユニバーサルプレート、ユニバーサル金具、ユニバーサルジョイント、ホイールセット (キャタピラ) がお勧めです。

ラインレース用のプログラム

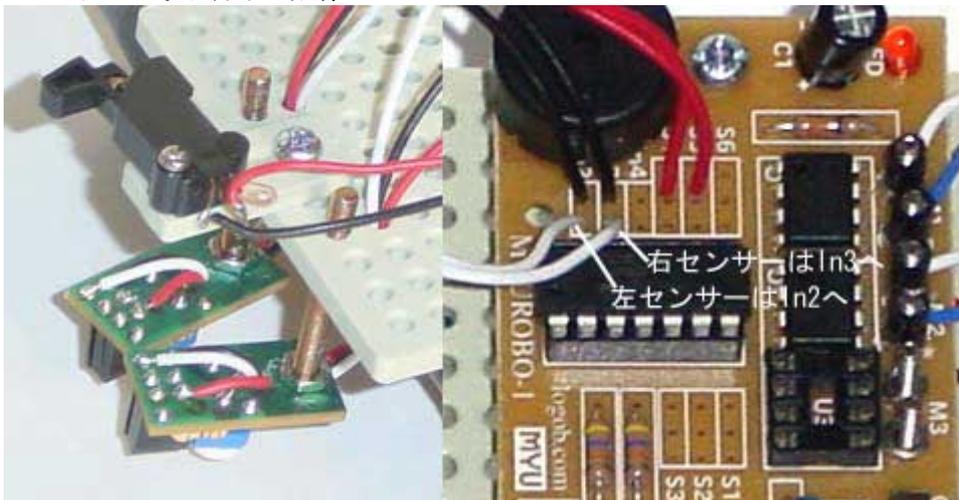
以下のプログラムの意味を自分で考えて、研究用に使ってください。これをそのまま打ち直すだけでも勉強になります。

その1 (赤外線センサー2つ)

```
//ラインレースロボット (MYU BASIC) 71byte
Byte C //直前の動き用変数
C=0
Do
  IfLo(2) Then //左センサーが黒なら
    IfLo(3) Then //左右とも黒
      前進
    Else //左:黒、右:白
      右前
      C=0 //右前を保存
    EndIf
  Else //左右とも白(コースアウト)の場合
    IfHi(3) Then //直前が右前
      左後
    Else //直前が左前
      右後
    EndIf
  Else //左:白、右:黒
    左前
    C=1 //左前を保存
  EndIf
EndIf
Loop
```



<センサーの取り付けと配線>



赤外線センサー3つ 16F630 タイプなら 134 バイト (容量不足)、16F688 タイプなら 124 バイト

```
//ライントレースロボット赤外線センサー3つ

Byte C          //直前の動き用変数

C=0

Do

//コースインの場合

ifLo(2) then //中央センサーが黒なら

    ifLo(1) then          //かつ左センサー黒なら

        ifLo(3) then //かつ右センサー黒、すなわち黒黒黒なら

            前進

        else //右センサー白、すなわち黒黒白なら

            右前

            C=0 //右前を保存

        endif

    else //左センサー白

        ifLo(3) then //かつ右センサーが黒、すなわち白黒黒なら

            左前

            C=1

        else //白黒白なら

            前進

        endif

    endif

else //中央センサー白

    ifHi(1) then //かつ左センサーが白なら

        ifHi(3) then          //かつ右センサーが白、すなわち白白白なら

            if C=0 then

                左後

            else
```

(次ページに続きます)

(続き)

```
        右後
    endif
else //右センサーが黒、すなわち白白黒なら
    左前
    C=1
endif
else //左センサ黒
    ifHi(3) then //かつ右センサーが白、すなわち黒白白なら
        右前
        C=0
    else //黒白白なら
        if C=0 then
            左後
        else
            右後
        endif
    endif
endif
endif
endif
Loop
```

<発展>

基板のマイコンを PIC630 から 688 (または 88) に変更することで、センサーの出力を「計測」することができます。(ただし windows2000～XP に付属のハイパーターミナルか、エクセルが必要です)

特にアナログセンサーの出力 (PIC にとっては入力) が計測ができると、その値をもとにプログラムでオンとオフの境界をより正確に設定することができ、ロボットのセンサーをより感度を上げることができます。

(たとえばAというセンサーで 100 まではオン、100 を超えたらオフというように。その 100 という値でうまくいかなかったら、80 あるいは 120 にしてみるというように、自由にその値 (しきい値といいます) を変えられることで、よりロボットを賢くすることができます。またそれでこそブレッドボードにした意味があります)