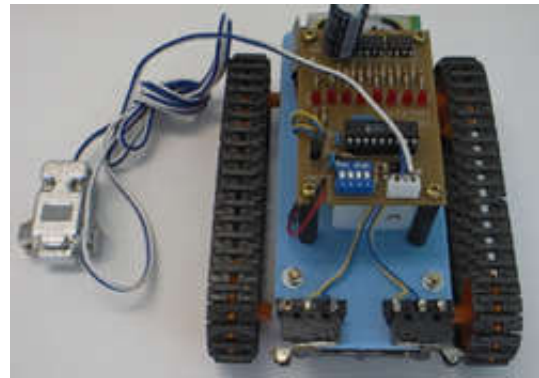


．使用するロボットとプログラム言語

1．ロボットの名称

動く模型の制作と計測・制御で用いる動く模型は、久富電機産業株式会社製RB-03という自律型ロボットです。



2．RB-03の仕様

ハードウェア

- | | |
|-------|--|
| 1) 制御 | ・ マイクロコンピュータを内蔵させた自律型ロボット |
| | ・ RS-232Cインターフェースでコンピュータと接続 |
| | ・ 赤外線センサでTVリモコンを使った操作 |
| 2) 入力 | ・ 赤外線センサ 1 |
| | ・ マイクロスイッチ 2 |
| | ・ デップスイッチ 1 |
| | ・ 簡易型RS-232Cポート 1 |
| 3) 出力 | ・ 小型DCモータ 2 |
| | ・ LED 1 |
| 4) 電源 | ・ モータ 3V (単三2個) |
| | ・ 制御系 (マイクロコンピュータ) 5V (3端子レギュレータ、006P電池) |
| 5) メカ | ・ 2軸ギアモータ |
| | ・ キャタピラ駆動 |

制御ソフトウェア

1) RB-03の内蔵プログラム

- | | |
|--------------|--|
| ・ 2軸ギアモータの制御 | TVリモコンによる制御
コンピュータからプログラムを転送して実行 |
| ・ 動作命令 | 前進、後退
右旋廻 (前進、後退)、左旋廻 (前進、後退)
右方向転換、左方向転換

指定時間待つ
分岐、繰り返し

入力があるまで前進、入力の状態を見る |

2) コンピュータで使用するプログラム言語

- | | |
|----------|--|
| ・ フリーソフト | ロゴ坊 (作者 兼宗 進 氏) |
| ・ 制御方法 | 基本プログラムをStartup.lgに登録し、制御したいプログラムを手順として作成、実行させます |

3．プログラミング言語ロゴとロゴ坊について

ロゴの紹介

ロゴは教育用のプログラミング言語です。親しみやすい命令を使って、画面上のタートル(カメ)を操作します。タートルを中心にした小さな世界(マイクロワールド)を子供たちが構築できます。

ロゴはプログラミングの教育にもたいへん適しています。他の言語にない、次のような特長を持っています。

操作する対象(タートル)が目に見える。

小さい関数が集まってプログラムになる。

強力な再帰がある。

万能データ構造のリストがある。

ロゴ坊の誕生

このようなすばらしい言語を多くの人に体験してもらうためにロゴ坊は生まれました。

ロゴ坊の特徴

カメがかわいい。 あゆみちゃんと言います。(^_^)

親しみやすい色付きのタートルが画面を動きます。

好きなエディタや日本語入力IMEと組み合わせて使えます。つまり、普段のパソコン生活を何ひとつ変える必要がありません。

標準的なロゴの文法を採用しています。

日本語に対応しています。 ひらがなや漢字で命令を入力したり、日本語でプログラムを書くことができます。

得意なこと

ロゴ坊は、小さなプログラムや対話的な処理に向いています。

子供(生徒)用のロゴ入門。

子供(生徒)用のプログラム入門。

大人(先生)用のロゴ体験。

大人(先生)用の簡単な教材作成(シミュレーションなど)。

かんたんモードを使って幼児のお絵かき。

苦手なこと

ロゴ坊は、以下のことが苦手です。

「速く動け!」と言われること。 カメなので、実行速度は速くありません。

(^_^ ;)

巨大なプログラムの実行。

マルチタートル。

4．ロゴ坊のプログラミングのしかた(手順の作成)

てじゅんは、手順は、to

おわり、終わり、終り、end

説明

手順を定義します。ロゴ坊のプロンプトから構文(1)の形式で実行すると、手順名と同じ名前のファイルをエディタで編集する画面になります。新しいファイルを指定した場合には、あらかじめ手順作成に必要な文字列が書き込まれてからエディタが起動します。書き込まれる文字列は、コマンドの文字列(ひらがな、漢字、英語)によって変わります。

コマンド	作成される文字列
to	to,end
てじゅんは	てじゅんは,おわり
手順は	手順は, 終わり

エディタの中では、構文(2)の形式で手順を定義することができます。これは
ロゴ坊のプログラム作成の基本形です。

作成したプログラムは、「ファイルを読み込め(load)」で読み込みます。

ファイル名を指定しないで実行した場合は、直前に編集したファイルが対象
になります。

起動直後のデフォルトのファイル名は「work.lg」です。

【注意】「てじゅんは」と「おわり」は、手順の本体とは別の行に独立して
記述してください。

【注意】手順名は評価されないので、ダブルクォート(")は不要です。

構文

```
(1) てじゅんは 手順名
(2) てじゅんは 手順名 引数の並び
...
( 手順本体)
...
おわり
```

5．ロゴ坊によるI/O制御（RS-232Cの制御）

現在ロゴ坊でRS-232Cを制御する場合以下の仕様になっています。

- ・ポートはcom1～com5を指定する。
- ・通信パラメーター
9600baud、8bits、パリティなし、ストップビット1、フロー制御なし
- ・ポートを開けないときはエラーとする。
- ・出力するデータは数値で渡す。
(「abc」という3文字を出力するには、`sprint 141 sprint 142 sprint 143` を実行する)

(文法)

<code>sopen</code>	ポート番号	RS-232Cポートを開く（使用可能にする）
<code>sout</code>	データ	RS-232Cポートにデータを書き込む
<code>sinp</code>		RS-232Cポートからデータを読み込む
<code>sclose</code>		RS-232Cポートを閉じる（使用不可能にする）

(例)

ロゴ坊の起動時に以下の命令を実行します。（com1を指定）

```
sopen 1
```

ポートにデータを書き出します。（十進数の141）

```
sout 141
```

ロゴ坊を終了時に以下の命令を実行します。

```
sclose
```

ここに紹介した内容と巻末の命令一覧は、作者 兼宗進 氏の了解を得て
ロゴ坊のマニュアルより掲載させて頂きました。

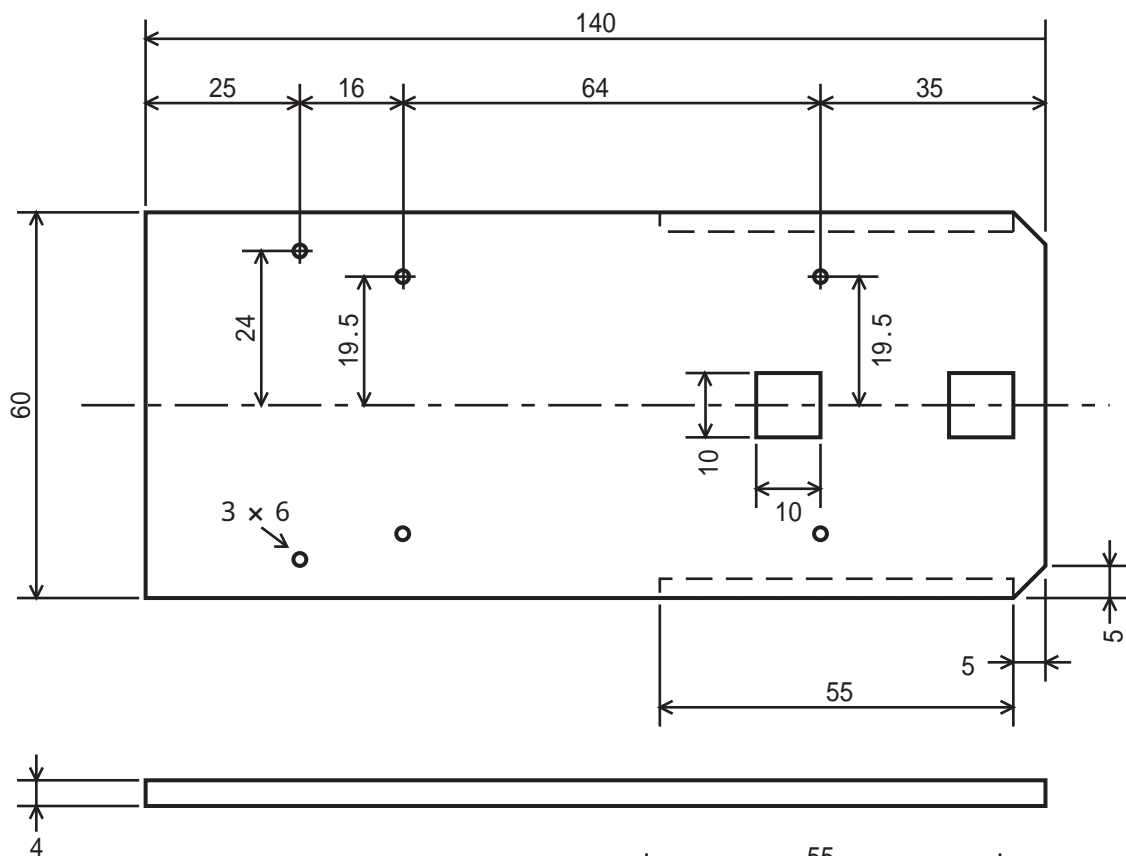
・キャタピラロボットの製作

R B - 0 3 の車体は、自作用教材として 4 mm 厚のエフセル板を以下のようにカットして車体としています。

車体寸法

・メインシャーシ

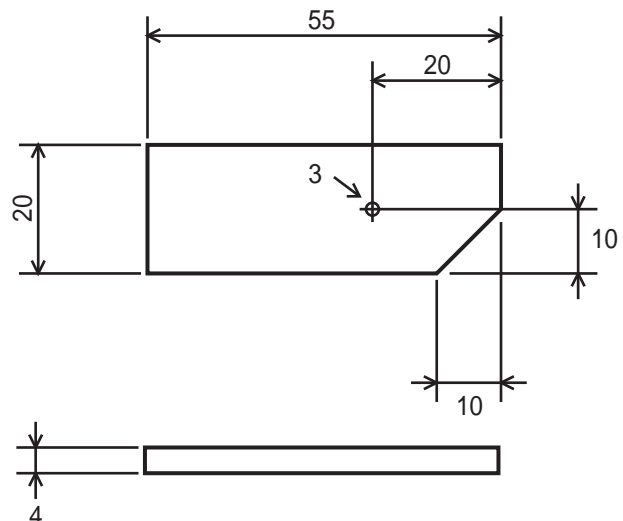
- ・材料のカットは、カッターナイフでカットして下さい。
- ・穴あけは、3 のドリルで穴あけをして下さい。



・サブシャーシ（2 個使用）

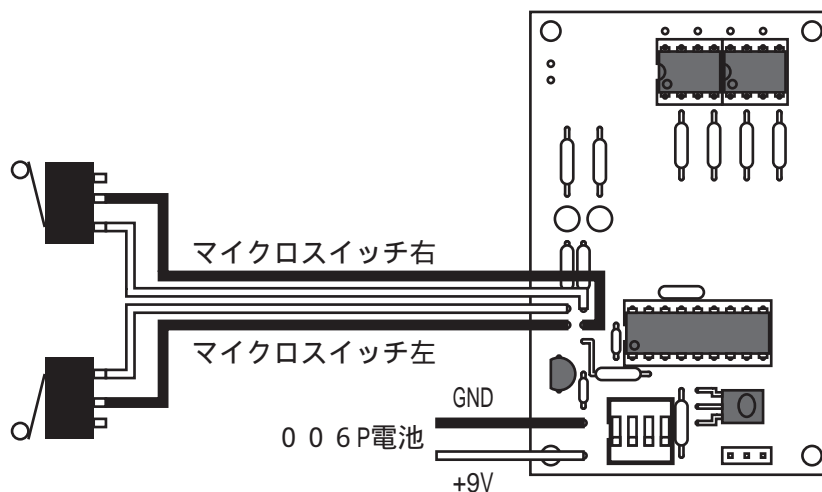
サブシャーシの組み立ては、ホットグルーボンドを使い接着して下さい。

サブシャーシの加工組み立て精度でロボットの直進性が決まってしまうので注意して製作して下さい。



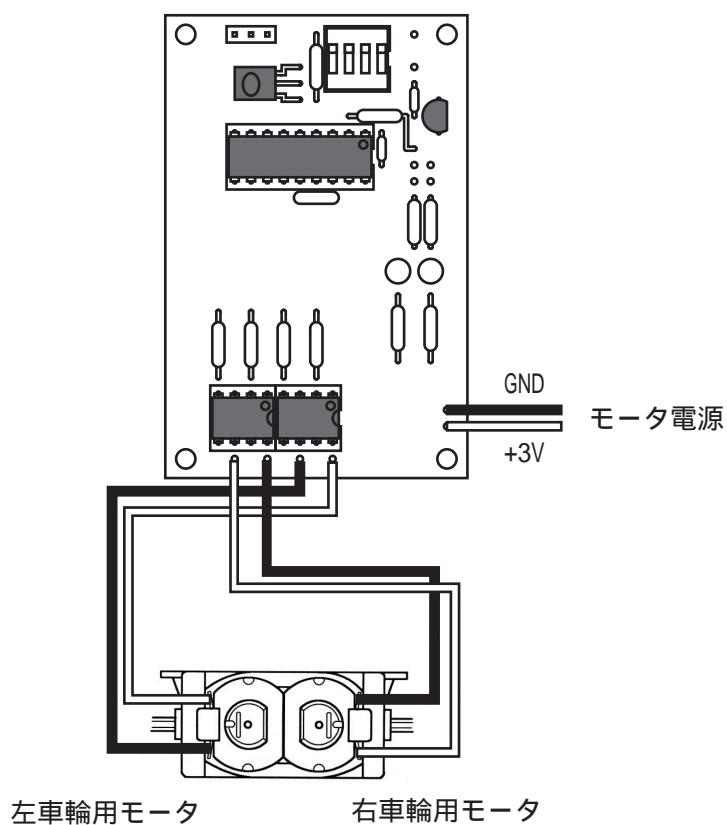
2．マイクロスイッチ配線

マイクロスイッチの配線で半田付けする端子は、COM、NO（スイッチが入ると接点が導通になる端子）間に半田付けをして下さい。
制御基板側の配線はあらかじめ配線されています。

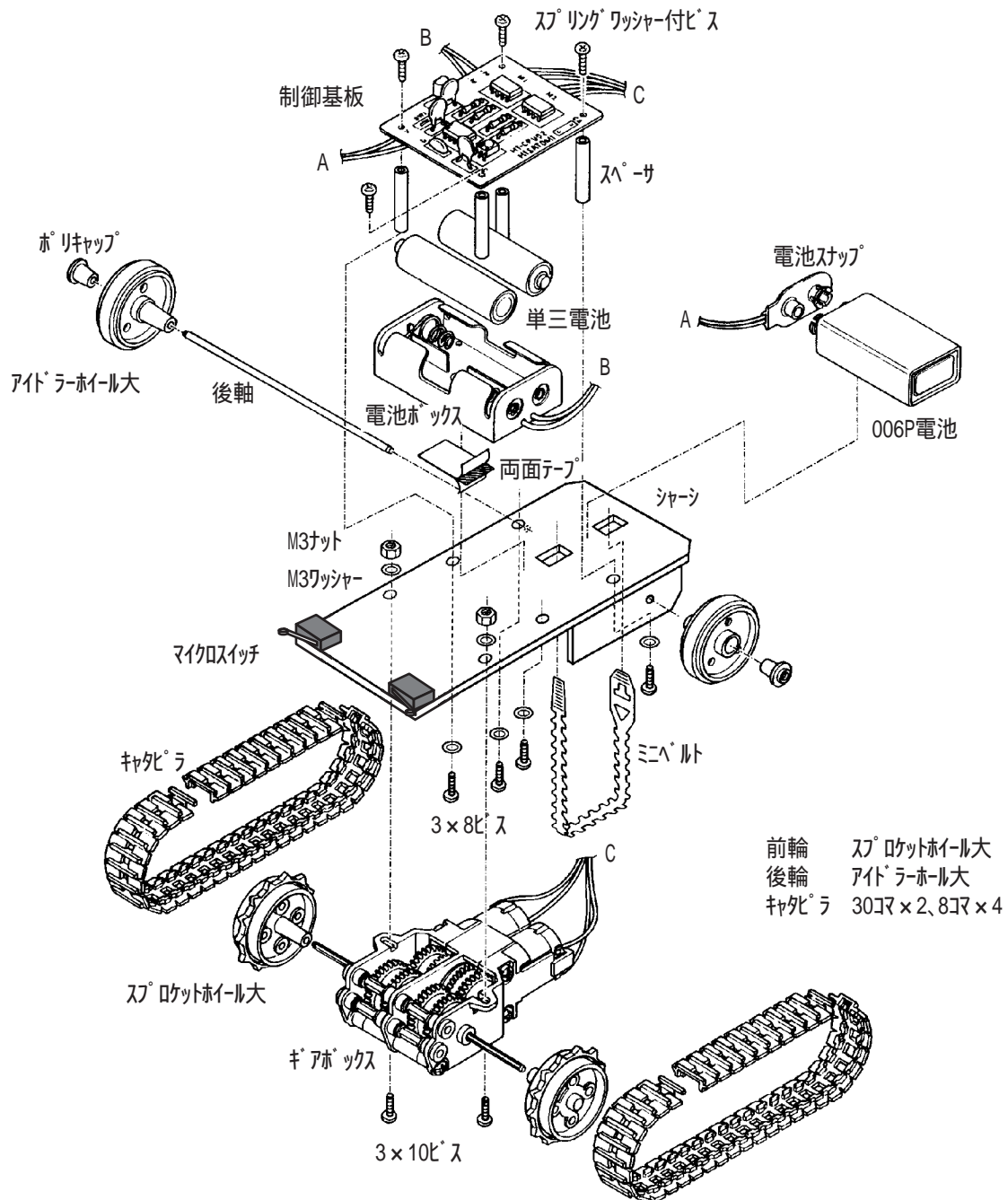


3．モータの配線

モータの配線は、極性に注意して半田付けをして下さい。



- 1) シャーシを図面のように穴あけ、カットをします。
- 2) ギアボックスを組み立てます。(低速仕様Cタイプ)
- 3) シャーシにギアボックス、ホイール、キャタピラを取り付けます。
- 4) モータ、マイクロスイッチの配線をします。
- 5) 電池ボックスの配線をします。
- 6) 制御基板を取り付けます。
- 7) 単三電池、006P電池を取り付けます。

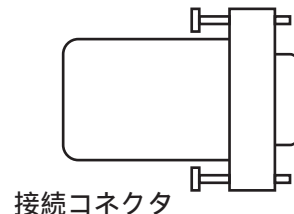


．操作の手順

1．コンピュータ側の準備

- ・ ロゴ坊をインストールします。
- ・ Startup.lgにRB-03用基本制御プログラムを登録します。
- ・ RS-232C(com1)にロボット通信ケーブルを接続します。
- ・ ロゴ坊を起動します。
- ・ ロゴ坊のコンソールに

sopen 1

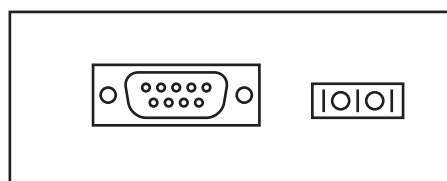


と入力します。（これでエラーがなければRS-232Cが使用可能になります）

RS - 232Cのコネクタは

コンピュータ Dサブ9ピンオス

接続ケーブル Dサブ9ピンメス

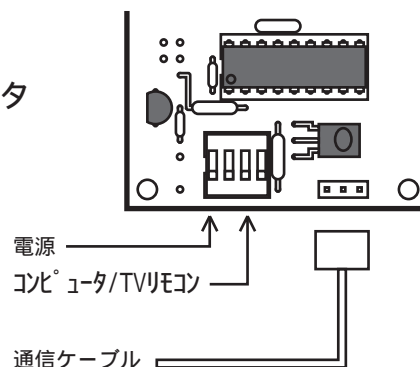


RS - 232Cポート

PC98の場合変換アダプタが必要です。

2．ロボットの準備

- ・ ディップスイッチ-3をOFF側にします。（コンピュータモード）
- ・ 電源スイッチをONします。（ディップスイッチ-3の設定を先にしないと設定は変更されません）
- ・ 正常に起動するとLEDが点滅します。
- ・ ロボット通信ケーブルをロボットの通信コネクタに接続します。（極性があります）

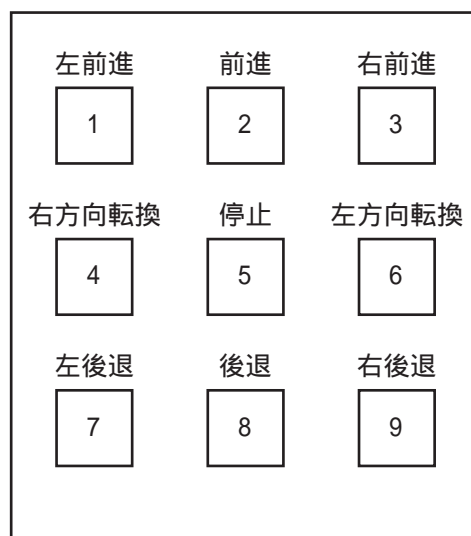


TVリモコンで操作

- ・ ディップスイッチ-3をON側にします。
- ・ 電源スイッチをONします。
- ・ 正常に起動するとLEDが点滅します。

TVリモコンでの操作は、右図のようにチャンネルボタンを押すことにより操作可能となります。

（使用できるリモコンはSONYのTV用、又は汎用リモコンでのSONY設定です）



TVリモコンでの動作

3．ケーブルの接続・動作の確認

まず以下の準備をして下さい。

- ・コンピュータに通信ケーブルを接続（Dサブ9ピンコネクタ）
- ・ロボットに通信ケーブルを接続（白い3ピンコネクタ）
- ・ロゴ坊を起動して、
sopen 1
を実行する。（RS-232Cポートを使用可能にする）
- ・ロボットをコンピュータモードで起動する（ディップスイッチ3をON）
- ・ロゴ坊のコンソールから
frt 10

を実行する。（1秒前進させる命令）

以上でロボットが1秒前進する動作を行えばコンピュータとロボットの接続はOKです。

4．プログラムの作成・転送・実行

プログラムは以下の書式で作成して下さい。

```
pgm  「プログラムのステップ数」      ;プログラムの転送命令
      |
      プログラム本体
      |
rob                                     ;プログラムの実行命令
```

これを手順として作成、実行すれば作成したプログラムを保存し、実行させることができます。

例．

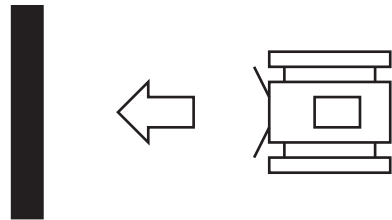
- ・5ステップのプログラムを作り実行させる。
- ・内容は、ロボットの左マイクロスイッチを手動でタッチした0.5秒後前進1秒、右に90度方向転換をするプログラムです。

```
to test1
  pgm 5
  inp1
  tim 5
  frt 10
  mvr 25
  stp
  rob
end
```

- ・load test1「ret」 でロゴ坊にプログラムを読み込みます。
- ・test1「ret」でプログラムの実行（プログラムの転送、実行）をします。

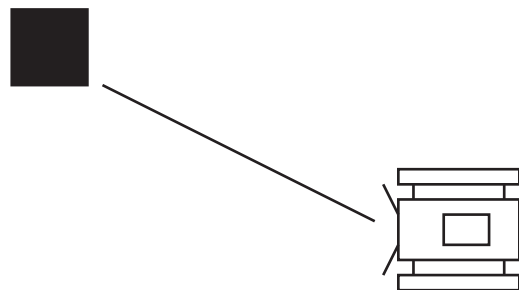
1. 例題：2つのスイッチのどちらかがONになるまで前進する

RB - 03 には、この命令が実装されています。



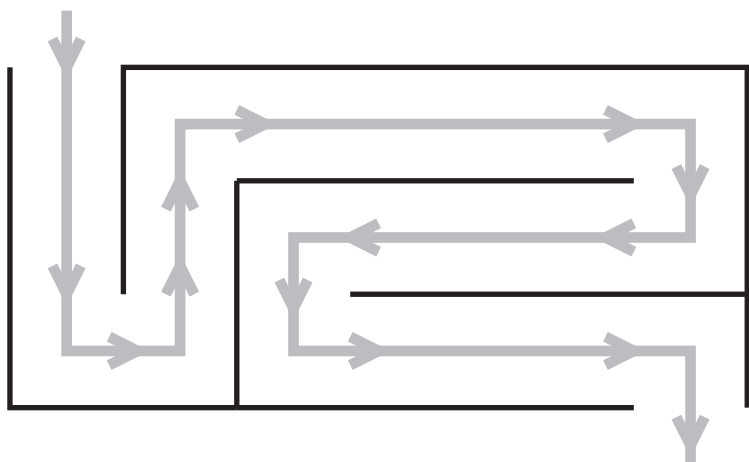
右図のようにロボットと探すものの距離角度が、分からない状態を想定します。

これを探すプログラムを考えて下さい。



右図のような迷路を用意しました。この迷路を抜け出すプログラムを考えて下さい。

プログラムステップ数は最大28ですからその範囲に収まるように工夫して下さい。



．命令一覧

1．R B - 0 3 命令一覧表

R B - 0 3 の命令は、2 バイト命令となっています。最初のバイトが動作の命令
次のバイトが時間又は回数という構成になっています。

・基本的な命令

命令 ftl [時間: × 0.1秒]
意味 左車輪が前進
例 ftl 10 ; 1 秒間左車輪が前進

命令 frt [時間: × 0.1秒]
意味 両車輪が前進
例 frt 10 ; 1 秒間前進

命令 ftr [時間: × 0.1秒]
意味 右車輪が前進
例 ftr 10 ; 1 秒間右車輪が前進

命令 mvr [時間: × 0.1秒]
意味 右に方向転換
例 mvr 10 ; 1 秒間右に方向転換

命令 tim [時間: × 0.1秒]
意味 指定時間待つ
例 tim 10 ; 1 秒間なにもしない

命令 mvl [時間: × 0.1秒]
意味 左に方向転換
例 mvl 10 ; 1 秒間左に方向転換

命令 bkl [時間: × 0.1秒]
意味 左車輪が後退
例 bkl 10 ; 1 秒間左車輪が後退

命令 bck [時間: × 0.1秒]
意味 後退
例 bck 10 ; 1 秒間後退

命令 bkr [時間: × 0.1秒]
意味 右車輪が後退
例 bkr 10 ; 1 秒間右車輪が後退

・プログラム転送、実行命令

命令 pgm [転送するステップ数]
意味 プログラム転送
例 pgm 3 ; 3 ステップのプログラムを転送
 frt 5
 mvl 10
 stp

命令 rob [引数はなし]
意味 プログラム実行
例 rob ; 転送したプログラムの実行

・無条件ジャンプ命令

命令 jmp [飛び先アドレス]
意味 無条件ジャンプ
例 jmp 1 ; プログラムの最初に飛ば

・繰り返し命令

命令 cnt1 [繰り返し回数]
意味 繰り返しループの回数を設定
例 cnt1 10 ; 繰り返し回数を 1 0 回に設定

命令 lop2 [飛び先アドレス]
意味 繰り返しをするループの頭へ設定回数ジャンプ

命令 lop3 [飛び先アドレス]
意味 繰り返しをするループの頭へ設定回数ジャンプ

命令 lop4 [飛び先アドレス]
意味 繰り返しをするループの頭へ設定回数ジャンプ

・マイクロスイッチの入力条件

命令 inp1 [引数はなし]
意味 入力1がONするまで待つ
例 pgm 3 ; プログラムを3ステップ転送
 inp1 ; 左側マイクロスイッチが入るまで待ちます
 frt 10 ; 1秒前進
 stp ; 終了

命令 inp2 [引数はなし]
意味 入力2がONするまで待つ
例 pgm 3 ; プログラムを3ステップ転送
 inp2 ; 右側マイクロスイッチが入るまで待ちます
 frt 10 ; 1秒前進
 stp ; 終了

命令 inpo [引数はなし]
意味 入力1,2のどちらかがONするまで待つ
例 pgm 3 ; プログラムを3ステップ転送
 inpo ; 左右両方のマイクロスイッチが入るまで待ちます
 frt 10 ; 1秒前進
 stp ; 終了

命令 inpa [引数はなし]
意味 入力1,2の両方がONするまで待つ
例 pgm 3 ; プログラムを3ステップ転送
 inpa ; 左右のどちらかのマイクロスイッチが入るまで待ちます
 frt 10 ; 1秒前進
 stp ; 終了

・マイクロスイッチが入るまで前進

命令 mov1 [引数はなし]
意味 入力1がONするまで前進
例 pgm 4 ; プログラムを4ステップ転送
 mov1 ; 左側のマイクロスイッチが入るまで前進
 back 10 ; 1秒後退
 mvl 25 ; 左に90度向きを変える
 stp ; 終了

命令 mov2 [引数はなし]
意味 入力2がONするまで前進
例 pgm 4 ; プログラムを4ステップ転送
 mov2 ; 右側のマイクロスイッチが入るまで前進
 back 10 ; 1秒後退
 mvl 25 ; 左に90度向きを変える
 stp ; 終了

命令 movo [引数はなし]
意味 入力1,2のどちらかがONするまで前進
例 pgm 4 ; プログラムを4ステップ転送
 movo ; 左右のどちらかのマイクロスイッチが入るまで前進
 back 10 ; 1秒後退
 mvl 25 ; 左に90度向きを変える
 stp ; 終了

命令 mova [引数はなし]
意味 入力1,2の両方がONするまで前進
例 pgm 4 ; プログラムを4ステップ転送
 mova ; 左右両方のマイクロスイッチが入るまで前進
 back 10 ; 1秒後退

・ 入力の有り無しで 2 つに条件分岐

命令	skp1 [引数はなし]
意味	入力1がONだと次の命令をスキップ
命令	skp2 [引数はなし]
意味	入力2がONだと次の命令をスキップ
命令	skpo [引数はなし]
意味	入力1,2のどちらかがONだと次の命令をスキップ
命令	skpa [引数はなし]
意味	入力1,2の両方がONだと次の命令をスキップ
命令	skz1 [引数はなし]
意味	入力1がOFFだと次の命令をスキップ
命令	skz2 [引数はなし]
意味	入力2がOFFだと次の命令をスキップ
命令	skz [引数はなし]
意味	入力1,2のどちらかがOFFだと次の命令をスキップ
命令	skza [引数はなし]
意味	入力1,2の両方がOFFだと次の命令をスキップ

・ 前進中に入力が入って停止に、指定時間入力がなければという条件をプラスした命令

命令	mvk1 [時間: x0.1秒]
意味	前進中に入力 1 が入れば停止。(指定時間入力がなければ次の命令をスキップ)
命令	mvk2 [時間: x0.1秒]
意味	前進中に入力 2 が入れば停止。(指定時間入力がなければ次の命令をスキップ)
命令	mvko [時間: x0.1秒]
意味	前進中に入力 1 , 2 のどちらかが入れば停止。(指定時間入力がなければ次の命令をスキップ)
命令	mvka [時間: x0.1秒]
意味	前進中に入力 1 , 2 の両方が入れば停止。(指定時間入力がなければ次の命令をスキップ)
命令	mvk [時間: x0.1秒]
意味	前進中に入力 1 が入れば停止、2 が入れば次の命令をスキップ(指定時間入力がなければ次の命令をスキップ)

・ 前進中に左右の入力の条件により 2 つに分岐する命令

命令	tur
意味	前進中に入力 1 がONだと次の命令をスキップ、入力 2 がONだと次の命令を実行。
例	pgm 11 ; プログラムを 1 1 ステップ転送
	cnt1 10 ; 繰り返し回数を 1 0 回に設定
	tur ;
	jmp 8 ; 右側のマイクロスイッチが入ったら、ステップ 8 ヘジャンプ
	bck 10 ; 左側のマイクロスイッチが入ったら、1 秒後退
	mvr 25 ; 右に 9 0 度向きを変える
	lop1 2 ; 全部で 1 0 回繰り返す
	stp ; 終了
	bck 10 ; 右側のマイクロスイッチが入ったら、1 秒後退
	mvl 25 ; 左に 9 0 度向きを変える
	lop1 2 ; 全部で 1 0 回繰り返す
	stp ; 終了

・ プログラムの終わりを表す命令

命令	stp
意味	プログラム終了
例	pgm 2 ; プログラムを 2 ステップ転送
	frt 10 ; 1 秒前進
	stp ; プログラムの終わりには必ずこの命令が必要です

2 . ロゴ坊命令一覧表

まえへ	前へ	forward	fd	タートルを前に動かす
うしろへ	後ろへ	back	bk	タートルを後ろに動かす
いろは	色は	setcolor	setc	ペンの色を変える
けしごむ	消しゴム	penerase	pe	ペンの色を背景の色にする
いろ	色	color		ペンの色を返す
ぬれ	塗れ	fill		色を塗る
とりけせ	取り消せ	undo		色塗りを取り消す
ペンをあげろ	ペンをあげる(上げる)	penup	pu	ペンを上げる
ペンをおろせ	ペンをおろせ(下ろせ)	pendown	pd	ペンを下げる
みぎへ	右へ	right	rt	タートルを右に回す
ひだりへ	左へ	left	lt	タートルを左に回す
えだけけせ	絵だけ消せ	clean		絵と字を消す
えをけせ	絵を消せ	cg		絵と字を消しタートルを中央に戻す
えをもとへ	絵を元へ	resetgraphics	rg	絵と字を消し画面を初期化する
かえれ	帰れ	home		タートルを中央に戻す
かくれろ	隠れる	hideturtle	ht	タートルを表示しない
でてこい	出てこい	showturtle	st	タートルを表示する
むき	向き	heading		タートルの向きを返す
むきは	向きは	setheading	seth	タートルの向きを変える
よこのいち	横の位置は	setx		タートルを x の位置に動かす
たてのいち	縦の位置は	sety		タートルを y の位置に動かす
たてよこのいち	縦横の位置は	setxy		タートルを x,y の位置に動かす
いち	位置は	setpos		タートルを [x,y] の位置に動かす
よこのいち	横の位置	xcor		タートルの x の位置を返す
たてのいち	縦の位置	ycor		タートルの y の位置を返す
いち	位置	pos		タートルの x,y の位置を返す
えん	円	circle		円を描く
まわりこみ	回り込み	wrap		wrapモードにする
とびだし	飛び出し	window		windowモードにする
しょうとつ	衝突	fence		fenceモードにする
きより	距離	distance		[x,y]への距離を返す
ほうこう	方向	towards		[x,y]への向きを返す
ペンの太さは	ペン(ペン)のふとさは	setpensize		ペンの太さを指定する
ペンの太さ	ペン(ペン)のふとさ	pensize		ペンの太さを返す
おそくなれ	遅くなれ	slowturtle		タートルの速さを遅くする
はやくなれ	速くなれ	fastturtle		タートルの速さを戻す
シェイプは	しえいぶは	setshape	setsh	シェイプを読み込む
シェイプのサイズは	しえいぶのさいずは	setshapsize	setshsize	シェイプのサイズを [x,y] にする
シェイプのサイズ	しえいぶのさいず	shapsize	shsize	シェイプのサイズを返す
画像の大きさ	がぞうのおおきさ	imagesize	imgsize	シェイプの画像サイズを返す
シェイプの大きさは	シェイプ(しえいぶ)のおおきさは	setshapemag	setshmag	シェイプの大きさを標準タートルの n 倍に
シェイプの大きさ	シェイプ(しえいぶ)のおおきさ	shapemag	shmag	シェイプの大きさを返す
あまり	余り	remainder		余りの計算
わ	和	sum		足し算
		+		足し算
さ	差	difference	diff	引き算
		-		引き算
まいなす	マイナス	minus		マイナス
せき	積	product		掛け算
		*		掛け算
しょう	商	quotient		割り算
		/		割り算
せいすう	整数	integer	int	小数を切り捨てる
ししゃごにゅう	四捨五入	round		小数を四捨五入する
へいほうこん	平方根	sqrt		平方根を計算する
らんすう	乱数	random		n未満の整数をランダムに返す
		sin		sinを計算する
		cos		cosを計算する
		tan		tanを計算する
		arcsin	asin	sinの逆関数を計算する
		arccos	acos	cosの逆関数を計算する
		arctan	atan	tanの逆関数を計算する
		abs		絶対値を計算する
		exp		べき乗を計算する
		log		対数を計算する

		power		累乗を計算する
		^		累乗を計算する
かけ	書け	print	pr	文字を表示する
つづけてかけ	続けて書け	type		文字を表示する(改行なし)
みせる	見せる	show		リストを括弧付きで表示する
じをけせ	字を消せ	cleartext	ct	画面の文字を消す
よんだじ	読んだ字	readchar	rc	1文字入力する
よんだきー	よんだキー/読んだキー	readkey		押されているキーを返す
よんだりと	よんだ(読んだ)リスト	readlist	rl	リストを入力する
かーそるのいち	カーソルのいち(位置)	setcpos		カーソルの位置を移動する
じのいろは	字の色は	setccolor	setcc	文字の色を変える
てじゅんは	手順は	to		手順を定義する
おわり	終わり/終り	end		手順の定義を終る
てじゅんのなかみ	手順の中身	printout	po	手順の中身を表示する
てじゅんのりすと	てじゅん(手順)のリスト	proclist		手順の一覧を表示する
てじゅんをなくせ	手順をなくせ	eraseproc		手順を消す
てじゅんをぜんぶなくせ	手順を全部なくせ	eraseprocs		手順をすべて消す
ふぁいるをだせ	ファイルをだせ(出せ)	edit		プログラムを編集する
ふぁいるをよみこめ	ファイルをよみこめ(読み込め)	load		プログラムを読み込む
へんすうは	変数は	make		変数に値を入れる
なかみ	中身	thing		変数の値を取り出す
		:		変数の値を取り出す
へんすうをなくせ	変数をなくせ	erasevar		変数を消す
へんすうをぜんぶなくせ	変数を全部なくせ	erasevars		変数をすべて消す
へんすうをかけ	変数を書け	printvars		変数の一覧を表示する
ぜんぶほんとう	全部本当	and		論理積(かつ)
どれかほんとう	どれか本当	or		論理和(または)
はんたい	反対	not		論理否定(~でない)
りすとか	リストか	list?		リストが調べる
からか	空か	empty?		空か調べる
ようそか	要素か	member?		リストに含まれるか調べる
すうじか	数字か	number?		数字か調べる
わーどか	ワードか	word?		ワードか調べる
へんすうか	変数か	var?		変数か調べる
よりおおきいか	より大きい	greater?		比較(大きい)
		>		比較(大きい)
よりちいさいか	より小さい	less?		比較(小さい)
		<		比較(小さい)
おなじか	同じ	equal?		比較(同じ)
		=		比較(同じ)
くりかえせ	繰り返せ	repeat	repeat n	Aをn回繰り返し実行する
ずっとくりかえせ	ずっと繰り返せ	forever	forever	Aを無限に繰り返し実行する
ほんとうのあいだ	本当の間	while	while [条件]	[条件]が成り立つ間Aを繰り返し実行する
もし		if	if 条件	[条件]が成り立てばAを実行する
もしどちらかを		ifelse	ifelse 条件	[条件]が成り立てばAを、そうでなければBを実行する
じっこう	実行	run	run [A] 	Aを実行する
わたせ	渡せ	output	op	手順から抜けて値を返す
とまれ	止まれ	stop		手順から抜ける
ぜんぶとまれ	全部止まれ	stopall		プログラムの実行を止めてロゴ坊のプロンプトに戻る
さよなら	さようなら	quit/exit/bye		ロゴ坊を終了する
さいしょ	最初	first		リストの最初の要素またはワードの最初の文字を返す
さいご	最後	last		リストの最後の要素またはワードの最後の文字を返す
さいしよがい	最初以外	butfirst	bf	リストの最初の要素またはワードの最初の文字を除いて返す
さいごがい	最後以外	butlast	bl	リストの最後の要素またはワードの最後の文字を除いて返す
さいしよに	最初に	firstput	fput	リストの先頭に要素を追加して返す
さいごに	最後に	lastput	lput	リストの末尾に要素を追加して返す
りすと	リスト	list		リストまたはワードをつないだリストを返す
ぶん	文	sentence	se	リストまたはワードの要素をつないだリストを返す
わーど	ワード	word		複数のリストまたはワードをつないだワードを返す
いくつめ		item		リストのn番目の要素またはワードのn文字目の文字を返す
かず	数	count		リストの要素数またはワードの文字数を返す
		hcopy		画面を印刷する
		rcopy		画面を反転印刷する
ほぞんはじめ	保存始め	logstart		実行したコマンドをファイルに記録する
ほぞんおわり	保存終わり/保存終り	logend		ファイルへの記録を止める
ついせきはじめ	追跡始め	trace		実行されているコマンドを表示する
ついせきおわり	追跡終わり/追跡終り	notrace		コマンドの表示を止める
まで	待て	wait		実行をn秒間休む
		step		コマンドを止めながら実行する
		nostep		コマンドを続けて実行する
やすめ	休め	pause		プログラムの実行を中断する
つづける	続ける	continue	co	プログラムの実行を再開する
ぜんかく	全角	fepon		日本語変換をはじめる
はんかく	半角	fepoff		日本語変換を止める
めもりー	メモリー	free		使用可能なメモリ量を表示する
ばーじょん	バージョン	version		ロゴ坊のバージョンを表示する
ひすとリー	ヒストリー	history		過去に実行した命令を表示する
		!		過去に実行した命令を再実行する