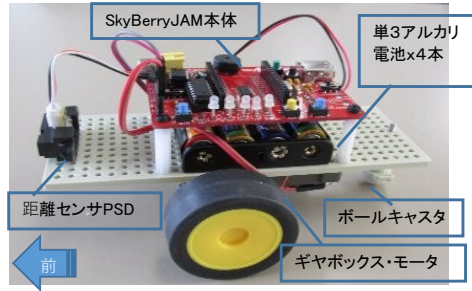


SkyBerryJAM ロボットレシピ 製作編 <初級向> 走行性抜群のセンサ付マイコンロボットカーをつくらう

SkyBerryJAMは、モータドライバIC(L293D)を搭載していますので、小型の模型ロボットをベースに、電池・センサなどを本体に接続すれば、比較的簡単に組み込みプログラミングロボットができます。

ここでは右のような、太く大きいタイヤで安定性・走行性抜群の、アナログ距離センサ(PSD)付のロボットカーを製作してみましょう。



●1 必要なパーツ



ツインモーターギヤボックス タミヤ
測距モジュールGP2Y0A21YK シャープ
電池ボックス 単3×4本 Bスナップ付
電池スナップソケット・Bスナップ(プラスチック型)
タッピングプレート(ボールキャスタ付) MYU002 スタジオミウ
50mmタイヤ(3mm穴角シャフト用) MYU004 スタジオミウ
樹脂6角スペーサM3×25mm長(メス・メス) x4個
単3アルカリ電池 x4個
セラミックコンデンサ 0.01~0.1μF x2個
ブレッドボード用ジャンパ配線コード(オスピン・オスピン20cm長) x4本
(または、通常の配線コード20cmx4本とピンヘッダ4ピン)

(各パーツ入手先例) 一般模型店・電子部品パーツ店・秋月電子通商・千石電商・桜木総業 など 計約3千円弱程度

●2 必要な工具 はんだごて(はんだ/台)/ニッパ/ドライバ/精密ドライバ/両面テープ/ワイヤストリッパ/ビニルテープなど

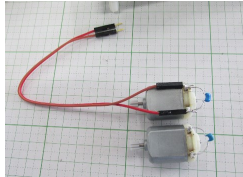
●3 組立

1) ギヤボックスの組立

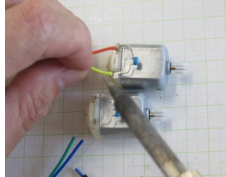
①モータの端子にセラミックコンデンサをつけ、コードをオスピンごとにはんだ付けする

(通常の配線コードの場合は、ワイヤストリッパなどで皮膜をむき、モータ端子にはんだづけする。)

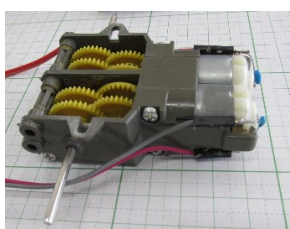
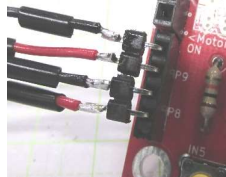
(反対側の先端には、ピンヘッダ4ピンなどをはんだづけする。さらに、収縮チューブやビニルテープなどで巻いておく。)



ジャンパ配線コードの場合



(通常の配線コードの場合は、反対の先端にピンヘッダを接続)

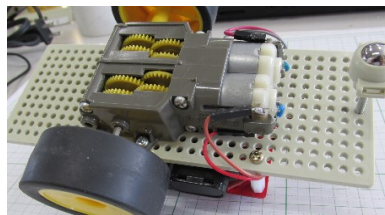


ギヤボックス<C>低速タイプに組立

②説明書にどおりに、ギヤボックスを<タイプC(低速)>に組立

③モータを装着

2) ボールキャスタ・スペーサ・電池BOX、ギヤボックス(タイヤ)をベース板にタッピングビスで取り付ける



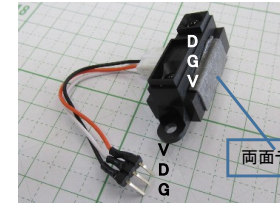
電池ボックスはほぼ中央に、スペーサはJAM本体の間隔で(長めのタッピングビスで)取付 タイヤをつけキャスタの高さも調整しておく

3) 測距(距離)センサPSD の処理

- ①コネクタ付コードの配線色がズレている場合は、赤線と黒線を精密ドライバーで外し入れ替えておくとい。(配線コードが長い場合は、半分程度に切断して先端をワイヤストリッパなどで皮膜をむいておくとい。)
- ②先端には、ヘッダピン3ピンなどをはんだづけする。さらに、収縮チューブやビニルテープなどで巻いておく。
- ③センサ上面に両面テープ(厚手)をはる



赤V/黒G/白Dの順にする



両面テープ(厚手)をはる

精密ドライバーで爪を上げて、赤線Vと黒線Gを取り出し入れ替

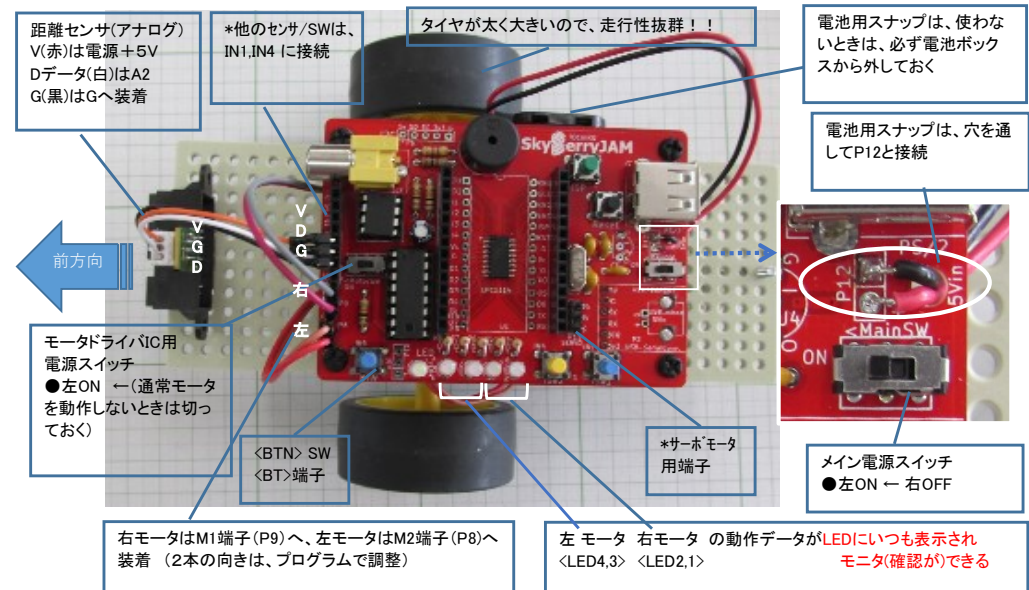
先端には、ピンヘッダ3ピン(別途)などを接続する

4) SkyBerryJAM本体の取付

- ①SkyBerryJAM本体P12ピンの＋ーに、電池スナップソケットの線を基板下から穴を通して、配線する。
- ②スペースに本体を取り付けする

5) モータ配線、センサ配線の装着

- ①右モータはM1端子(P9)へ、左モータはM2端子(P8)へ装着する(2本の方向は後で調整)
- ②データ線(白線D)A2へ、(赤線V)は電源5V、(黒線G)はGへ装着し、両面テープでベース台にはり付ける



- *ロボットの前後方向やセンサ・ギヤボックス・電池などの取付位置は、自由にアレンジ可能
- *プレート・ベース板をカットし、全体の長さを2/3程度に短くコンパクトにしてもOK
- *他のセンサやSW(サーボモータなど)も接続して、ライトレースカーなどにも

●4 プログラミング方法

<モニタ><キーボード>を接続し、動作プログラム制作して、0番に保存

●5 ロボット動作方法 (モニタ・キーボードを外し)

<BTN>を押しながら<メインSW>オンすると、0番のプログラムが自動起動

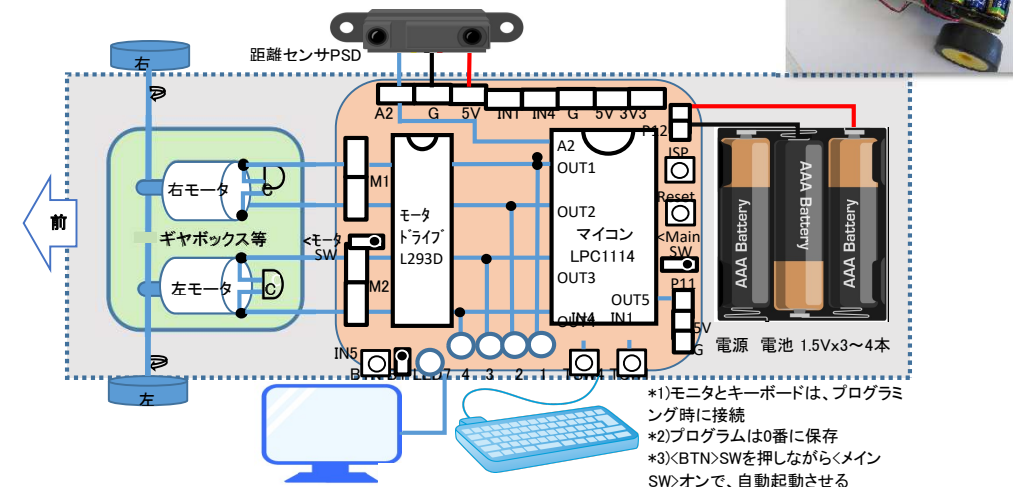
(または、BTブート端子をオンしておくとい。<メインSW>オンで常に0番が自動起動)

OUTポート	4	3	2	1
LED	LED4	LED3	LED2	LED1
L293D	M2左モータ動作データ	M1右モータ動作データ		
INポート	A2	IN4	IN1	
	PSDセンサ	TSW4	TSW1	

SkyBerryJAMロボットレシピ 概要編

SkyBerryJAMは、直流モータドライバIC (L293D)を標準で搭載していますので、組込みプログラミングロボットが比較的容易に構成できます。(右写真参照)

●1)SkyBerryJAMロボット構成



ロボット構成イメージ図

- ①各直流モータの端子には、雑音防止のために、セラミックコンデンサC(0.01~0.1uF)を接続します。
- ②<M1>ピン端子には右モータ、<M2>には左モータを接続します。(各モータの2本の配線向きは、後で調整します。)
- ③電源として、単3アルカリ電池や充電電池等3~4本を<P12>に接続します。
- ④距離センサ(アナログ型)は、A2(アナログ入力2)に接続します。(他のスイッチやセンサはIN1,4などに接続)
- ⑤プログラムは0番に保存、(モニタ・キーボードを外し)<BTN>を押しながら<メインSW>オンして自動起動させます。(または、BTブート端子をオンしておく、<メインSW>オンで常に0番が自動起動します。)

●2)モータ動作確認と調整(JAM本体にはモニタとキーボード接続、MainSWとモータSWをOnにしておく)

OUTポート	4	3	2	1
データ例	1	0	0	1
LED	●	○	○	●
L293D	左モータ	右モータ	動作データ	動作データ

例) **OUT 9** または **OUT '1001**
OUTポートに 9(2進値1001)を出力するとLEDは、●○○● と表示されモータドライバIC-L293Dにより左モータ前進・右モータ後退=全体右回転

動作データ	モータ動作
00	停止(フリー)
10	回転(前進)
01	反転(後退)
11	ブレーキ

ロボット本体は、台などをおいて浮かした状態にして、次のコマンドを入力して、モータ回転とLED表示を確認します。

	<左モータ>	<右モータ>
① OUT 1<enter>	-	回る
② OUT 2<enter>	-	逆に回る
③ OUT 0<enter>	-	停止
④ OUT 4<enter>	回る	-
⑤ OUT 8<enter>	逆に回る	-
⑥ OUT 0<enter>	停止	停止

*前進する向きになるように調整(<M1><M2>それぞれで、2本の配線を入れ替え)し、上の表の動作となるようにします。

●3)ロボット基本動作コマンドの確認

次のコマンドを入力して、**モータ回転**と**LED表示**を確認します。

	<左モータ>	<右モータ>	<全体>
① OUT 8+2<enter> =OUT 10<enter>	↑	↑	=前進
② OUT 4+1<enter> =OUT 5<enter>	↓	↓	=後退
③ OUT 8+1<enter> =OUT 9<enter>	↑	↓	=右回転
④ OUT 4+2<enter> =OUT 6<enter>	↓	↑	=左回転
⑤ OUT 0<enter>	-	-	=停止

LED表示	動作データ 2進値	動作データ 10進値	ロボット動作
●○○●	1010	10	前進
○○●○	0101	5	後退
●○○○	1001	9	右回転
○●○○	0110	6	左回転
○○○○	0000	0	停止

SkyBerryJAMロボットレシピ プログラミング編

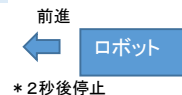
●例1 前進(2秒間)プログラム

```
1' ZENSHIN
10 WAIT 120
20 OUT 10:WAIT 120
30 OUT 0
40 END
SAVE 0 <enter>
```

先頭にはプログラムの内容を 'で書いておくとよい
2秒程度待つ
前進 2秒待つ
停止
プログラム終了
0番に保存

LEDの表示
●○○○ 1010
○○○○ 0000

モニタ・キーボードを外し、<BTN>を押しながら<メインSW>オンして自動起動させます。
ロボット実行動作後は、<メインSW>OFF、モニタ・キーボードをつなぎ、再び<メインSW>ONする
動作確認後、保存しておきたい場合は、0番を一度呼び出して他の番号、例えば1番に保存しておく



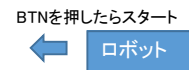
●例2 前進・後退・左右回転 連続動作プログラム

```
NEW
1' RENZOKU
10 WAIT 120
20 OUT 10:WAIT 120
30 OUT 5:WAIT 120
40 OUT 9:WAIT 120
50 OUT 6:WAIT 120
60 OUT 0: GOTO 10
SAVE 0 <enter>
```

2秒程度待つ
前進2秒
後退2秒
右回転2秒
左回転2秒
停止、10行目へ移動 無限繰り返し(ループ)
0番に保存

LEDの表示
○○○○ 0000
●○○○ 1010
○○●○ 0101
●○○○ 1001
○○○○ 0000

(以下同様に0番に保存し、動作後保存しておきたい場合は、他の番号で保存しておく)



●例3 BTN(一番左のボタン)を押したらスタートするように例2を改造する

```
LOAD 0<enter>
15 IF BTN()=0 THEN GOTO 15
SAVE 0 <enter>
```

BTN()は、BTNボタンを調べる命令で、押してない0の間は、15行目(現在行)を無限に繰り返し、ボタンを押すまで待っている
(例題2に、上の1行を追加する)
(以下同様に0番に保存し、動作後保存しておきたい場合は、他の番号で保存しておく)

●例4 <TSW1>を押したら右回転(3秒間)、<TSW4>を押したら左回転(3秒間)するプログラム

```
NEW
1' BTN KAITEN
10 IF IN(1)=0 THEN OUT 9:WAIT 180:OUT 0
20 IF IN(4)=0 THEN OUT 6:WAIT 180:OUT 0
30 GOTO 10
SAVE 0 <enter>
```

IN(1)はTSW1を調べる命令で、押すと0となる
IN(4)はTSW4を調べる命令で、押すと0となる
10行目へ移動 無限繰り返し(ループ)
0番に保存

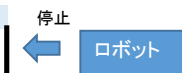
(以下同様に0番に保存し、動作後保存しておきたい場合は、他の番号で保存しておく)

●例5 前進して、前にかべがあれば停止するプログラム(距離センサ利用)

```
NEW
1'ZENSHIN SENSOR
10 WAIT 120
20 D=ANA(2):PRINT D
30 IF D>550 THEN OUT 0:END
40 OUT 10:WAIT 5
50 GOTO 20
SAVE 0 <enter>
```

2秒程度待つ
ANA(2)は距離センサを調べる命令 0~1024でかべまでの距離
壁が近い550(最大650程度)以上なら停止 ENDプログラム終了
そうでないなら 少し前進
20行目へ移動 無限繰り返し(ループ)
0番に保存

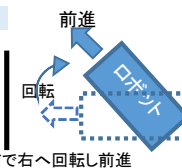
(20行目の PRINT D があるのでモニタ画面をつないでみると、センサの値が確認できる 550の値は調整する必要)
(以下同様に0番に保存し、動作後保存しておきたい場合は、他の番号で保存しておく)



●例6 前進して、かべがあったら向きを変え、再び前進するプログラムに例5を改造する

```
LOAD 0<enter>
30 IF D>550 THEN OUT 9:WAIT 120
SAVE 0 <enter>
```

壁が近い550以上なら右回転2秒間
(550や2秒間の値は、調整して変更する)
(例題5の30行目を上に変更する)
(以下同様に0番に保存し、動作後保存しておきたい場合は、他の番号で保存しておく)



●番外 動作スピードを調整するには、PWM命令を使います

例)前進 PWM 4,1000:PWM 2,1000 OUT 4, 2 に10ms周期で信号出す
停止 PWM 4,0:PWM 2,0 (1000がスピード値となる 最大2000)