



**JO-ZERO&STD-  
ZEROで  
「さがしてポン」にチャ  
レンジ**

**姫路ソフトウェアス  
第一版**

# 姫路ロボチャレンジの自律競技 「さがしてポン」とは



- **専用リング**上に配置されたボール(大中小、各紙パイプ状の台にのせて固定)を、ロボットが自律探査(クラスCロボットは電源及び制御支援用パソコンを有線等で接続可能)してリング上から落とす競技です。

持ち時間:3分間

スタート場所:赤コーナー側スタートライン(姫路ロボチャレンジロゴシール上)からスタート。

ボールの配置:オペレーターがサイコロを振って、出た目によって玉の置き場が決まります。

ボール:中心の高さは台(紙パイプ製)にのせられた場合の中心の位置です。

大:直径30cm(中心の高さ17cm)

中:直径20cm(中心の高さ12cm)

小:直径10cm(中心の高さ7cm)

ボールの材質:3個とも発泡スチロール樹脂

ボールの色:3個とも白色

障害物:リング上には障害物が置かれます。

場所は中央付近で、サイコロによって決めます。

障害物の数は1個

大きさは横30cm、幅10cm、高さ4cmの直方体

障害物の材質:発泡スチロール樹脂

障害物の色:白色

## 注意事項

競技中にロボットがリングより落下した場合は競技終了。

スタート時及び終了時以外はロボットに触れる事を禁じます。

競技中にロボットに操縦者から指示を与える事はできません。

ロボットの体当たりでボールが落ちた場合は得点は無効(ジャッジはレフェリーが行う)

落したボールが他のボールを落した時も得点は有効。

持ち時間が終了した時は競技終了。

得点:ボールを3個落すのに要した時間が短い方から得点を得る。

1位:5点 / 2位:4点 / 3位:3点 / 4位:2点 / 5位:1点

また、各ボールをパンチ・キック等胴体部以外で落とした時は加点されます。

大:1点 / 中:2点 / 小:3点

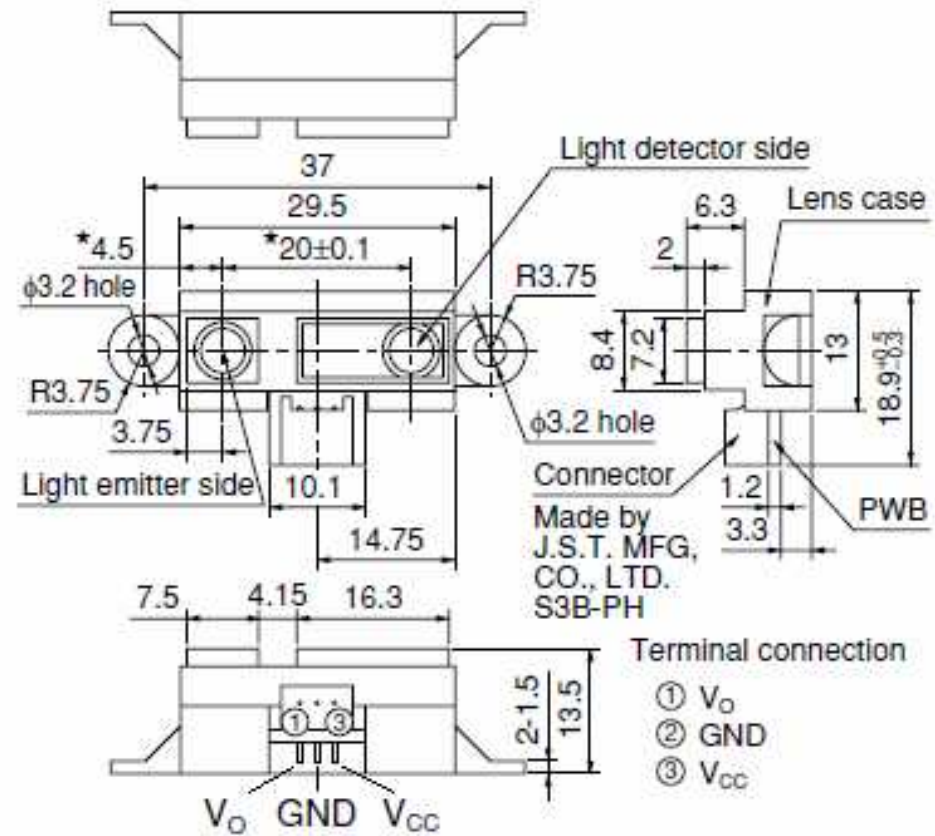
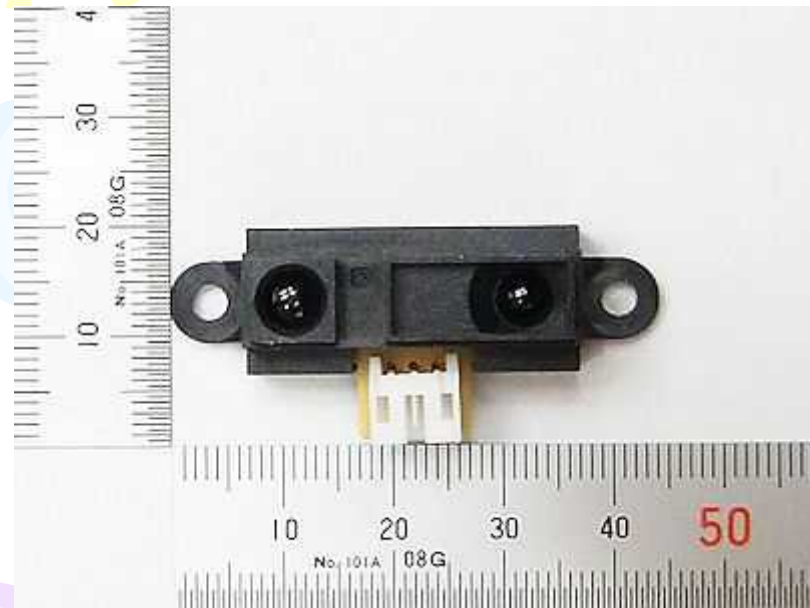
すべてのボールを落としたとき、その経過で障害物に触れていなければボーナスポイント3点を追加。

3個すべてのボールを落とせなかった場合は、1個目を落とすまでの時間で順位を決めます。

# JO-ZERO & STD-ZEROにセンサーを搭載する。

- 対物センサーとしてシャープ測距モジュール GP2Y0A21YKを搭載する例を示します。
  - 秋月電子通商などで購入できます。
  - シャープの測距モジュールです。赤外線LEDとPSD (position sensitive detector) を使用して、非接触で距離を検出することができます。
    - 測距範囲: 10 ~ 80cm
    - 出力: アナログ電圧出力
    - 寸法: 29.5 × 13 × 13.5mm
    - 用途: ロボット・アミューズメント等
    - 電源: 4.5 ~ 5.5V

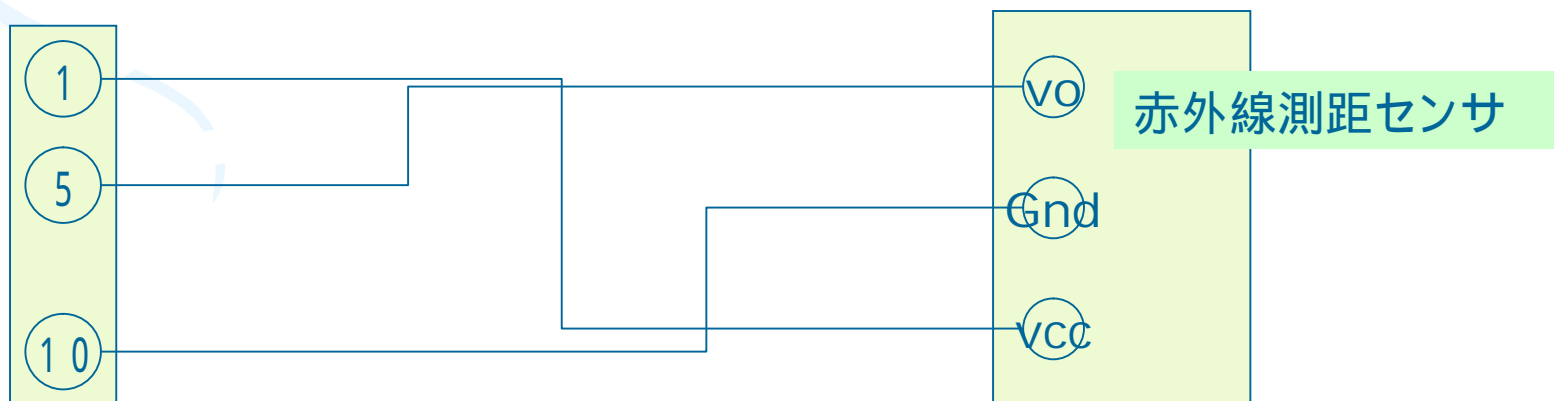
# センサーの外観(寸法図)



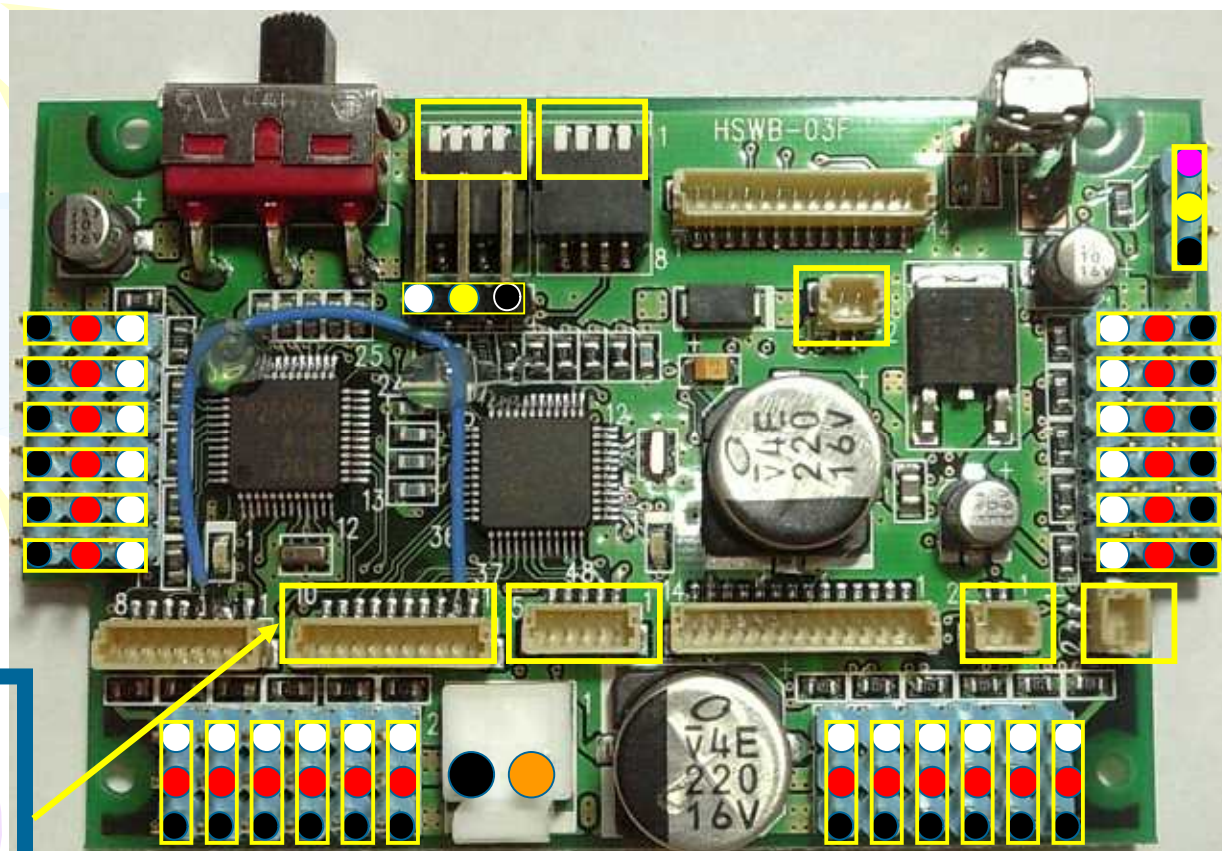
# センサーとHSWB-03Fの接続

- センサーのアナログ出力をHSWB-03Fの汎用アナログ入力6に接続し、電圧変化をプログラムで確認できるようにします。
- 接続回路図は下記を参照してください。
  - HSWB-03Fのアナログ入力端子のコネクタハウジングはRSコンポーネンツなどで入手可能です。
    - アナログ用ポート JST SHR-10V-S-B
    - 圧着端子つきハーネス SH3-SS5-28150

HSWB-03F  
アナログ入  
力端子



# 端子配置図(表面)



アナログ  
入力端  
子  
デジタル  
入力端  
子

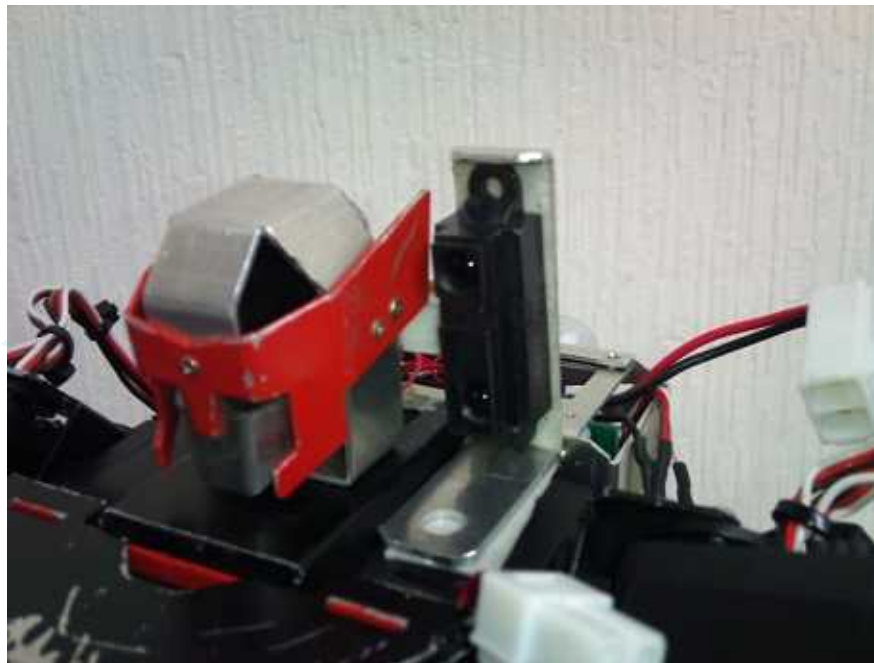
アナログ入力端子デジタル入力端子  
8ビット分解能低速アナログ入力端子  
です。  
入力電圧は+5Vから0Vの範囲です。  
端子割り当ては下記のとおりです。

- 1・+5V
- 2・使用不可
- 3・AD4IN
- 4・AD5IN
- 5・AD6IN
- 6・DI5
- 7・DI6
- 8・DI7
- 9・GND
- 10・GND

-  +7.4V電源供給端子
-  サーボへの電源供給端子
-  +5V安定化電源出力
-  信号端子(送信)
-  信号端子(受信)
-  グランド

# センサーの搭載方法

- JO-ZEROの肩や胸などにセンサーを取り付けます。
- 両面テープでの取り付けやオリジナルのフレームを用意してください。
- 参考例を画像でご確認ください。



# プログラムサンプル

- 第10回姫路ロボチャレンジスタンダードクラスで総合優勝したSTD-ZEROベースのロボット「スタンド」の実際のプログラムです。
- 床判定を行い、床があれば前進、なければ少し後退して方向転換を行う処理を繰り返しています。

```
: START
Move(0,0,0,0,0,0,-38,418,928,540,-38,0,0,38,-540,-928,-418,38,0,0,0,0,0,0,700)
: SENSORCHWECK
Move(0,0,0,0,0,0,-38,901,938,467,-38,0,0,38,-467,-938,-901,38,0,0,0,0,0,0,400)
WAIT(50)
V00=INPUTADC(6)
JUMPIF(V00,<,.40,TURN)
:
Move(0,0,0,0,0,0,-38,418,928,540,-38,0,0,38,-540,-928,-418,38,0,0,0,0,0,0,700)
:
: MOTGENESTART216
MOVE(60,X,X,X,X,X,-149,550,893,373,-149,X,X,-76,-385,-921,-566,-76,0,X,X,X,X,-12,63)
MOVE(-261,X,X,X,X,X,-180,657,1077,450,-180,X,X,-76,-385,-921,-566,-16,0,X,X,X,X,-12,63)
V119 = 2
: MOTGENELOOP216
MOVE(-429,X,X,X,X,X,-147,713,1166,483,-147,X,X,-36,-448,-938,-519,54,0,X,X,X,X,-153,63)
MOVE(-600,X,X,X,X,X,-126,773,1041,299,-126,X,X,-36,-504,-936,-463,24,-150,X,X,X,X,-321,63)
MOVE(-282,X,X,X,X,X,-116,664,816,182,-116,X,X,-51,-552,-922,-401,-51,-260,X,X,X,X,-507,63)
MOVE(-243,X,X,X,X,X,-48,651,879,258,-48,X,X,25,-596,-904,-337,25,-300,X,X,X,X,-699,63)
MOVE(-135,X,X,X,X,X,32,615,910,325,32,X,X,102,-619,-853,-264,102,-260,X,X,X,X,-918,63)
MOVE(12,X,X,X,X,X,14,566,922,386,74,X,X,174,-671,-1063,-422,174,-150,X,X,X,X,-444,63)
MOVE(165,X,X,X,X,X,-15,515,930,445,75,X,X,195,-477,-1147,-701,195,0,X,X,X,X,393,63)
MOVE(333,X,X,X,X,X,15,459,929,500,75,X,X,167,-294,-1024,-760,167,150,X,X,X,X,570,63)
MOVE(507,X,X,X,X,X,49,401,923,552,49,X,X,115,-182,-817,-665,115,260,X,X,X,X,285,63)
MOVE(699,X,X,X,X,X,-25,337,904,596,-25,X,X,47,-258,-879,-651,47,300,X,X,X,X,243,63)
MOVE(918,X,X,X,X,X,-103,264,853,619,-103,X,X,-34,-325,-910,-615,-34,260,X,X,X,X,135,63)
MOVE(444,X,X,X,X,X,-177,422,1062,671,-177,X,X,-76,-385,-921,-566,-16,150,X,X,X,X,-12,63)
: MOTGENELOOPCHECK216
V119 = V119 - 1
JUMPIF(V119,>,.0,MOTGENELOOP216)
: MOTGENELOOPCHECKEND216
MOVE(-261,X,X,X,X,X,-180,657,1077,450,-180,X,X,-76,-385,-921,-566,-16,0,X,X,X,X,-12,63)
MOVE(0,X,X,X,X,X,-38,570,928,388,-38,X,X,38,-388,-928,-570,38,0,X,X,X,X,0,63)
JUMP(SENSORCHWECK)
```

```
:
: TURN
Move(0,0,0,0,0,0,-38,418,928,540,-38,0,0,38,-540,-928,-418,38,0,0,0,0,0,0,700)
: MOTGENESTART217
MOVE(X,X,X,X,X,X,-149,403,893,520,-149,X,X,-76,-536,-921,-415,-76,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-180,480,1077,627,-180,X,X,-76,-536,-921,-415,-16,X,X,X,X,X,25)
V119 = 2
: MOTGENELOOP217
MOVE(X,X,X,X,X,X,-147,513,1166,683,-147,X,X,-36,-506,-936,-460,54,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-129,380,1064,714,-129,X,X,-36,-467,-938,-501,24,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-120,277,854,607,-120,X,X,-51,-429,-934,-535,-51,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-48,330,898,598,-48,X,X,26,-389,-930,-571,26,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,33,376,917,571,33,X,X,106,-339,-899,-591,106,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,14,416,922,536,74,X,X,179,-473,-1083,-640,179,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-15,457,929,502,75,X,X,195,-671,-1147,-507,195,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,15,497,931,464,75,X,X,172,-702,-1046,-375,172,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,50,535,934,429,50,X,X,118,-608,-855,-277,118,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-26,571,930,389,-26,X,X,48,-598,-898,-330,48,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-108,591,899,338,-108,X,X,-34,-571,-917,-376,-34,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-181,639,1082,472,-181,X,X,-76,-536,-921,-415,-16,X,X,X,X,X,25)
: MOTGENELOOPCHECK217
V119 = V119 - 1
JUMPIF(V119,>,.0,MOTGENELOOP217)
: MOTGENELOOPCHECKEND217
MOVE(X,X,X,X,X,X,-180,480,1077,627,-180,X,X,-76,-536,-921,-415,-16,X,X,X,X,X,25)
MOVE(X,X,X,X,X,X,-38,418,928,540,-38,X,X,38,-540,-928,-418,38,X,X,X,X,X,25)
V119 = 5
: MOTGENESTART215
MOVE(X,X,X,X,X,X,-382,770,986,246,-292,X,X,292,-740,-986,-276,382,X,X,X,X,X,8)
WAIT(1)
MOVE(X,X,X,X,X,X,-38,501,938,467,-38,X,X,38,-467,-938,-501,38,X,X,X,X,X,380)
V119=V119-1
JUMPIF(V119,>,.0,MOTGENESTART215)
JUMP(SENSORCHWECK)
```