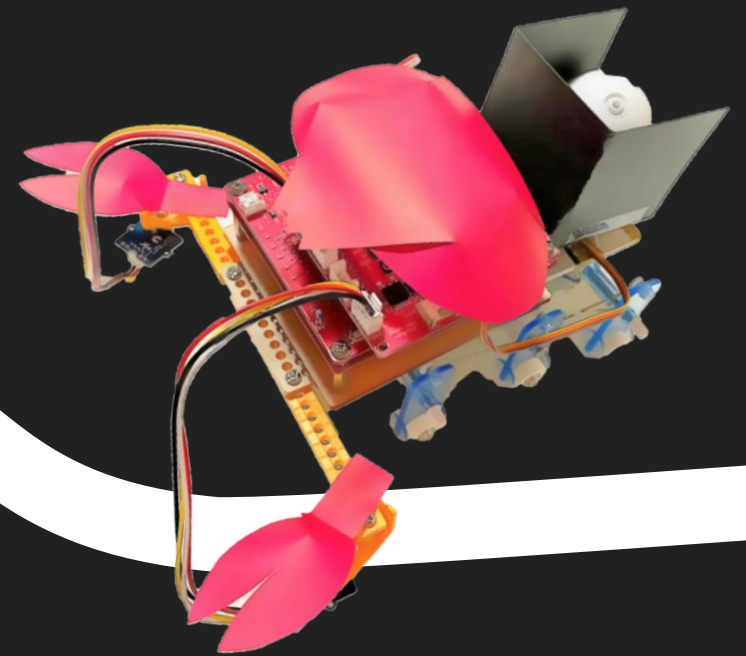


Matz葉がにロボコン プレ大会 事前講習会

2023/1/5,6 @オープンソースラボ



プレ大会

Matz葉がに ロボコン



2023.1.15 (sun)

会場：松江テルサ 大会議室



主催：松江工業高等専門学校
共催：しまねOSS協議会、松江市
後援：島根県
協力：福井県こどもプログラミング協議会
かにロボ連盟：越前がにロボコン
JAXA 宇宙科学研究所 Planet-C プロジェクト

Matz葉がにロボコンって？

スモウルビーや軽量Rubyを活用した、

こども向けのプログラミングコンテスト

かにをモチーフにしたロボットを作成して、
コースで走行させて競争しよう！







ミッション

金星に探査ロボットを送り込み、
プローブ（観測装置）を投入せよ
さらに、金星の雲粒子を取得して
地球に帰還せよ

得点

項目	点数
松江エリアを出た（ロボット全体）	1
中間点を越えた（ロボット全体）	1
金星エリアに入った（片道ゴール）	1
金星エリアにプローブを投入した ※投入後にプローブがエリアから転がり出ても OK ※プローブはロボットに「載せた状態」で運ぶ	1
松江エリアに戻った（ロボット全体）	2
雲粒子を松江エリアに持ち帰った ※最大 3 点まで	1(x3)
[本選のみ]相手より先に「ゴール」と言った	1



今日のスケジュール

- 8:30 受付開始
- 9:00 講習会開始
 - ・ロボットの組み立て
 - ・プログラミング
- 12:00 お昼休憩
- 13:00 講習会再開
 - ・プログラミング
 - ・試走
- 16:00 終了

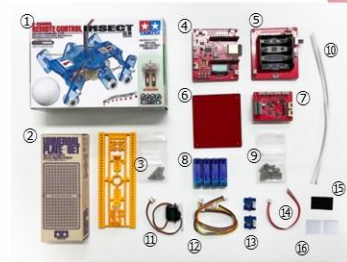
カニロボを作ろう

KANIROBOT 組み立てマニュアルを見てね

KANIROBOT

組み立てマニュアル

セット内容



- | | | |
|-------------------|-----------|-------------|
| ①タミヤインセクト | ⑥プレート | ⑫GROVEケーブル |
| ②ユニバーサルプレート | ⑦モータードライバ | ⑬ライトセンサー x2 |
| ③ユニバーサルアーム
セット | ⑧単3電池 x4 | ⑭電源ケーブル |
| ④マイコンボード | ⑨ネジセット | ⑮ゴムクッション |
| ⑤電池ボックス | ⑩結束ワイヤー | ⑯マジックテープ |
| | ⑪サーボモーター | |

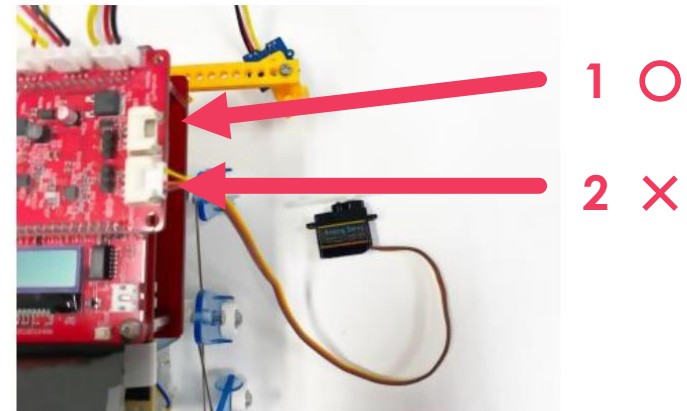
道具

- ニッパー
- ワイヤーストリッパー
- ラジオペンチ
- はさみ
- カッター
- セロハンテープ or マスキングテープ
- 工作用紙
- 両面テープ

カニロボを作ろう

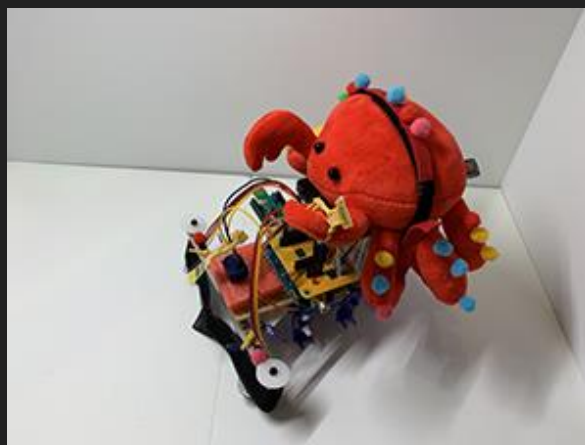
- マニュアルを見て作ってね
 - 道具は、自分たちの机に用意されているものを使ってね
 - 仲良く交代で使ってね
-
- サーボモーターは、2じゃなくて1にさしてね

③サーボモーターをつなぐ



デコレーションも工夫しよう！

ライバルの越前がにロボット



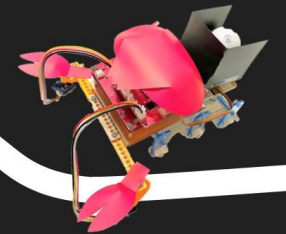
大会当日にはデザイン賞があります
優秀者にはプレゼントあり！？

プログラミングを
やってみよう

KANIROBOT 講習会資料を見てね

Matz葉がにロボコン
プレ大会
事前講習会

2023/1/5 @オープンソースラボ



プログラミングの準備

まず、プログラミング環境を用意する

1. パソコンの電源を入れる
2. 「Chromeブラウザ」を開く
 -  をクリック
3. URLにアクセス

- <http://kanicon.epi.it.matsue-ct.jp/>

完了！

プログラミング画面

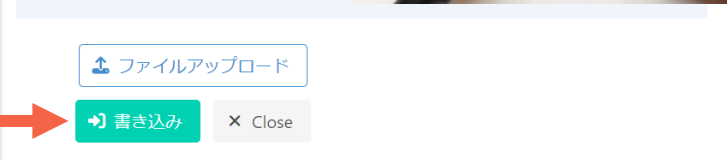
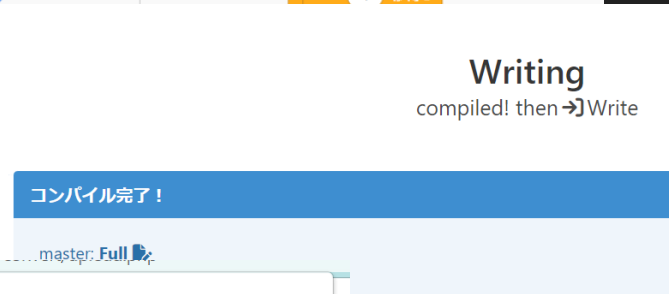
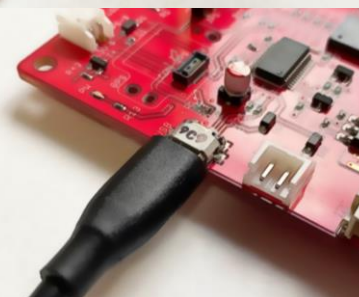
バックバック

はじめてのプログラミング

The screenshot shows the Scratch 3.0 IDE interface. The top menu bar includes 'ファイル', '編集', 'チュートリアル', 'スモウルビーのプロジェクト...', '共有する', 'プロジェクトページを見る', '意見を送る', and the user name 'smalruby-hatti'. The left sidebar shows various code blocks categorized by function: Control, Math, Variables, Projects, Mechanisms, and Pin, PWM, ADC (ふつう), Pin, PWM, ADC (かんたん), Pin, PWM, ADC (むずい), and I2C, UART. The main workspace contains a script for a Raspberry Pi project. The script starts with an initialization block: '初期化: LED1 を出力モードで使う'. This is followed by a 'ずっと' (forever) loop containing three blocks: 'LED1 を ON にする', '1 秒待つ', and 'LED1 を OFF にする'. The right sidebar shows a Raspberry Pi hardware model with four switches (SW1-SW4) set to 0. Below the hardware model is the 'スプライト' (Sprite) area with a 'master' sprite and a 'slave' sprite. The 'ステージ' (Stage) area shows '背景 1'. At the bottom of the workspace, there is a 'バックパッ' (Back) button.

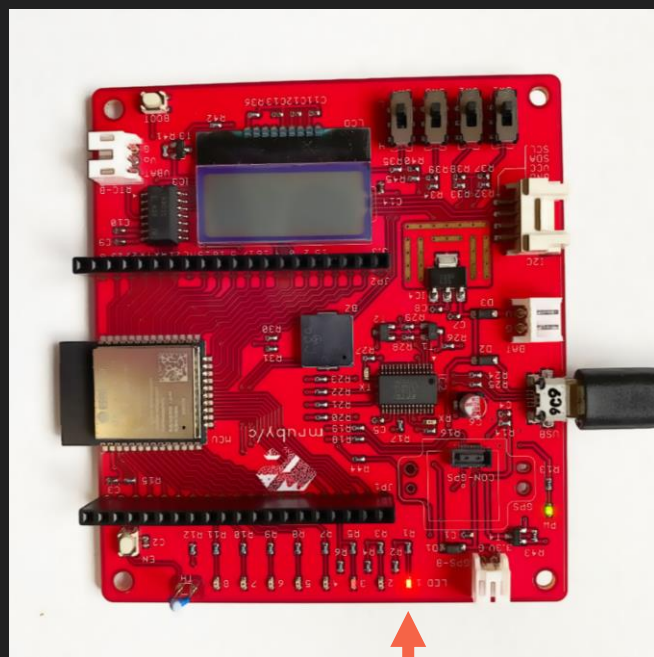
マイコンを動かす

1. ファイル > マイコンに書き込む
2. ケーブルでマイコンとパソコンをつなぐ
3. 書き込みボタン
4. USB Serial Port を選ぶ
5. 「Serial Close」と表示されたら成功



マイコンを動かす

LED1がチカチカ！



書き込みの注意

- 書き込みページは複数開かない
- もう 1 度書き込むときは、ケーブルをさし直す
 - （失敗したときは、「Disconnect」ボタンを押してからケーブルを抜くと良い）
- 書き込みページはリロード（再読み込み）しないで
- 「書き込み」ボタンのくるくるが止まらないときは、ページを消して、ケーブルをさし直す

書き込みページ

Full : プログラムあり
Empty : プログラムなし

Writing
compiled! then → Write

コンパイル完了!

master: Full
slave: Empty

ファイルアップロードでも
書き込みできる

書き込み

ファイルアップロード
→ 書き込み × Close

書き込みページを消す

OUTPUT

OUTPUT

Debug

OUTPUTを消す

Clear

Connect

Disconnect

マイコンの接続を切る

カニロボを動かすプログラミング

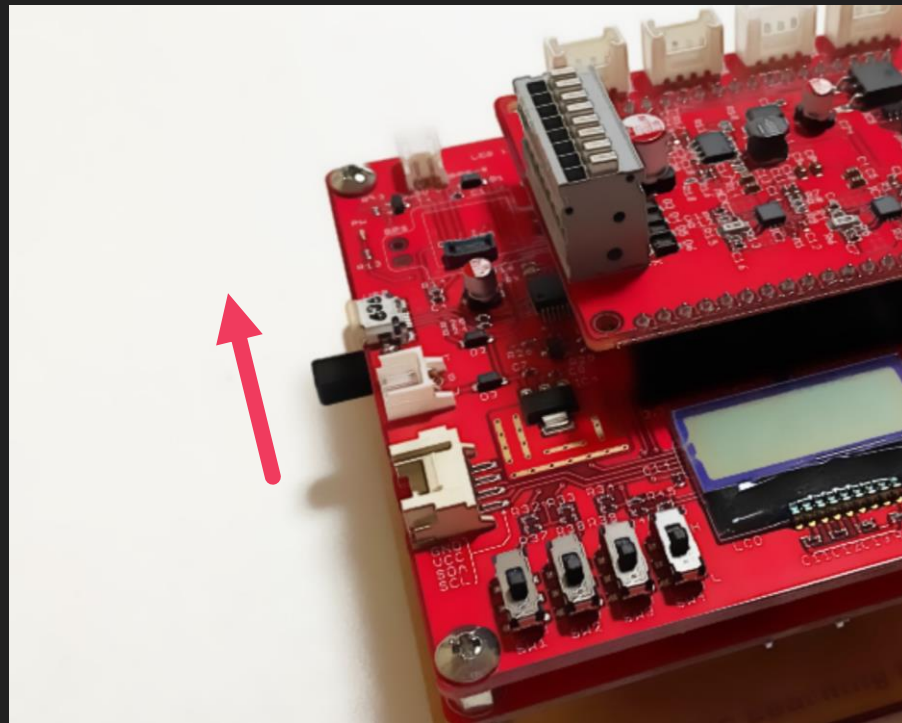
The screenshot displays the Scratch programming environment with a script for a crab robot. The script is as follows:

- 初期化：モーターを使う
- 右 のモーターを 前 に動かす
- 右 のモーターを止める
- 初期化：ライトセンサー 1 を使う
- ライトセンサー 1 の明るさ
- 初期化：サーボモーター 1 を使う
- サーボモーター 1 を 0 度にする
- 20 秒待つ
- 右 のモーターを止める
- 左 のモーターを 前 に動かす
- 左 のモーターを止める

The interface also shows a hardware monitor for switches SW1, SW2, SW3, and SW4, all set to 0. A sprite monitor shows a 'master' sprite, and a stage monitor shows a background of 1.

モーターを動かす

- マイコン側のケーブルを外す
- マイコンのスイッチを入れる



サーボモーターを動かすプログラミング

The image displays a Scratch-like programming environment with a blue header bar. The main workspace contains a sequence of code blocks for controlling a servo motor:

- Initial block: 初期化：サーボモーター 1 を使う (Initialize: Use servo motor 1)
- Block 1: サーボモーター 1 を 0 度にする (Set servo motor 1 to 0 degrees)
- Block 2: 3 秒待つ (Wait 3 seconds)
- Block 3: サーボモーター 1 を 90 度にする (Set servo motor 1 to 90 degrees)

A green callout box with a white border points to the '0' and '90' degree values in the code blocks, containing the text: 数字は 0 ~ 120まで (Numbers are 0 to 120).

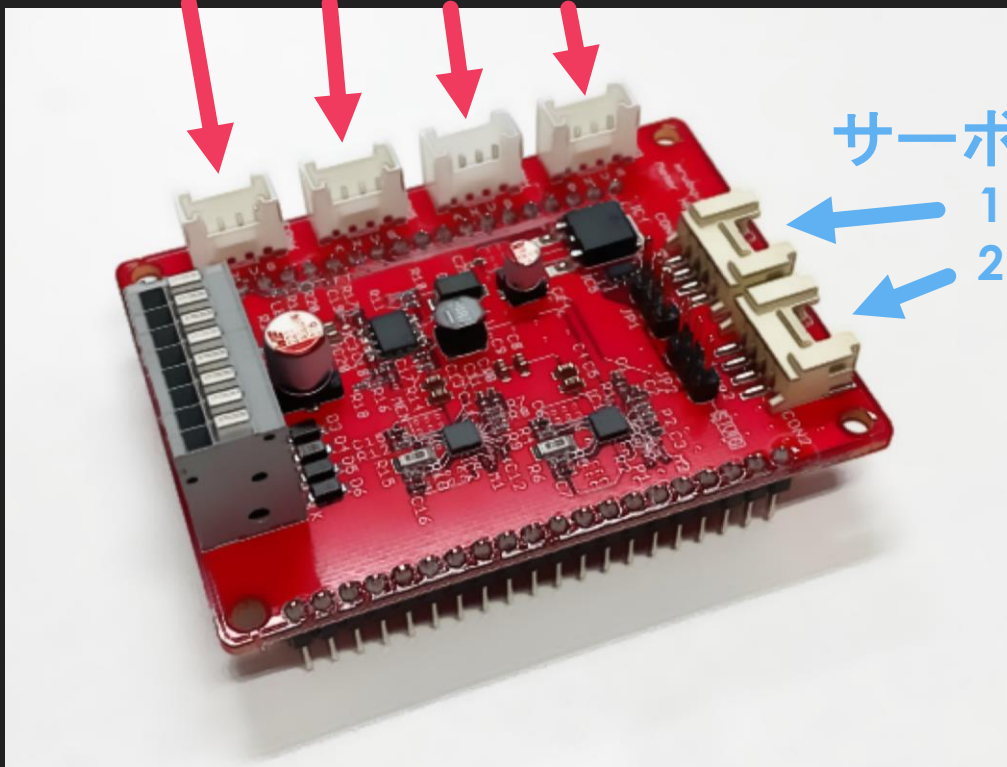
The interface includes a left sidebar with a 'パーツパレット' (Parts Palette) showing various sensors and actuators. The right sidebar features a 'モーター' (Motor) control panel with four switches (SW1-SW4) and a 'スプライト' (Sprite) panel with 'master' and 'slave' options. The bottom status bar shows 'バックバック' (Back Back).

配置

モーター
1: 右
2: 左

ライトセンサー

4 3 2 1



サーボモーター

1
2

ライトセンサーを使うプログラミング

0 500~

スモルビーのプロジェクト... 共有する プロジェクトページを見る 意見を送る smalruby-hatti

初期化: ライトセンサー 1 を使う
ずっと
モニターの 1 行目に ライトセンサー 1 の明るさ と書く
1 秒待つ

モニター

バックバック

SW1 0
SW2 0
SW3 0
SW4 0

スプライト
master
表示する 向き 90
大きさ 100
slave

ステージ
背景 1

ライトセンサーとモーターを使うプログラミング

The screenshot displays the Scratch programming environment. The main workspace is a grid where a script is being built for a 'カニロボ' (Crab Robot). The script includes the following blocks:

- Initial block: 初期化: モーターを使う (Initialize: Use Motor)
- Event block: 右 のモーターを 前 に動かす (Move the right motor forward)
- Event block: 右 のモーターを止める (Stop the right motor)
- Initial block: 初期化: ライトセンサー 1 を使う (Initialize: Use Light Sensor 1)
- Event block: ライトセンサー 1 の明るさ (Light Sensor 1 brightness)
- Initial block: 初期化: サーボモーター 1 を使う (Initialize: Use Servo Motor 1)
- Event block: サーボモーター 1 を 0 度にする (Set Servo Motor 1 to 0 degrees)

The left sidebar shows various categories of blocks, including 'カニロボ' (Crab Robot) with blocks for motor pin initialization, enabling the motor, and setting speed. The right sidebar shows a visual representation of the robot with four switches (SW1-SW4) and a 'Sprite' panel with a 'master' sprite.

At the bottom of the workspace, there is a 'バックバック' (Back) button.



線をたどる

白線の両脇にライトセンサーがくるように置く

まず、まっすぐ進む

もし、右足のセンサが白を感知したら、右に方向を変える

もし、左足のセンサが白を感知したら、左に方向を変える

金星到着

右足も左足も白を感知したら、到着！

サーボモーターを動かして、プローブを落とす

地球に帰る

Uターンして、松江に戻ろう！



ライトセンサーとモーターを使うプログラミング

The screenshot shows the Scratch programming environment with a code block for a robot's movement logic. The code is written in Japanese and includes the following steps:

- 初期化 (Initialization)
- ずっと (Forever loop):
 - もし 右足が白を探知 なら (If right foot detects white, then):
 - 右足を止める (Stop right foot)
 - でなければ (Otherwise):
 - 右足を動かす (Move right foot)
 - もし 左足が白を探知 なら (If left foot detects white, then):
 - 左足を止める (Stop left foot)
 - でなければ (Otherwise):
 - 左足を動かす (Move left foot)
 - 0.1 秒待つ (Wait 0.1 seconds)

Red and blue arrows are overlaid on the code to indicate the flow of execution. A red arrow points from the '右足を止める' block to the 'もし 右足が白を探知 なら' block, and a blue arrow points from the '左足を止める' block to the 'もし 左足が白を探知 なら' block. The text '右向きに進む' (Move right) is written in red, and '左向きに進む' (Move left) is written in blue.

ライントレース (かんたん)

The screenshot shows the Scratch IDE interface for a project titled "スモウルビーのプロジェクト...". The main workspace contains a Scratch script for a line-following robot. The script is written in Japanese and uses blocks from the "カニロボ" (CaniBot) library.

Script Details:

- Initialization:** Three "初期化" (Initialize) blocks are used to set up the motors and light sensors.
 - 初期化: モーターを使う (Initialize: Use Motor)
 - 初期化: ライトセンサー 1 を使う (Initialize: Use Light Sensor 1)
 - 初期化: ライトセンサー 2 を使う (Initialize: Use Light Sensor 2)
- Main Loop:** A "ずっと" (Forever) loop block contains the following logic:
 - Condition:** "もし ライトセンサー 1 の明るさ > 100 なら" (If Light Sensor 1 brightness > 100 then)
 - Action:** "右 のモーターを止める" (Stop the right motor)
 - Else:** "でなければ" (Otherwise) block containing "右 のモーターを 前 に動かす" (Move the right motor forward).
 - Condition:** "もし ライトセンサー 2 の明るさ > 100 なら" (If Light Sensor 2 brightness > 100 then)
 - Action:** "左 のモーターを止める" (Stop the left motor)
 - Else:** "でなければ" (Otherwise) block containing "左 のモーターを 前 に動かす" (Move the left motor forward).

Left Sidebar (Component Categories):

- 演算 (Math)
- 変数 (Variables)
- 作ったブロック (Created Blocks)
- 機器 (Hardware)
 - 初期化: ライトセンサー 1 を使う (Initialize: Use Light Sensor 1)
 - ライトセンサー 1 の明るさ (Light Sensor 1 brightness)
 - 初期化: サーボモーター 1 を使う (Initialize: Use Servo Motor 1)
 - サーボモーター 1 を 0 度にする (Set Servo Motor 1 to 0 degrees)
- Pin, PWM, ADC (かんたん) (Pin, PWM, ADC (Easy))
- Pin, PWM, ADC (ふつう) (Pin, PWM, ADC (Normal))
- Pin, PWM, ADC (むずい) (Pin, PWM, ADC (Hard))
- I2C, UART
- カニロボ (CaniBot)
 - モーター有効化ピンを初期化 (Initialize motor enable pin)
 - モーターを有効化ピンを 有効 化 (Enable motor with enable pin)
 - モーター 1 を初期化 (Initialize motor 1)
 - モーター 1 のスピードを初期化 (Initialize motor 1 speed)

Right Sidebar:

- SW1, SW2, SW3, SW4 (Switches)
- Sprite (スプライト) section: "master" sprite, size 100, direction 90.
- Stage (ステージ) section: "背景 1" (Background 1).

At the bottom of the workspace, there is a "バックバック" (Back) button.

ライトレース (普通)

初期化：モーターを使う
初期化：ライトセンサー 1 を使う
初期化：ライトセンサー 2 を使う

ずっと

もし **ライトセンサー 1 の明るさ > 100** なら

右 のモーターを止める

でなければ

右 のモーターを 前 に動かす

もし **ライトセンサー 2 の明るさ > 100** なら

左 のモーターを止める

でなければ

左 のモーターを 前 に動かす

0.1 秒待つ

同じ意味

モーター有効化ピンを初期化
モーターを有効化ピンを 有効 化
モーター 1 を初期化
モーター 1 のスピードを初期化
モーター 2 を初期化
モーター 2 のスピードを初期化
ライトセンサー 1 を初期化
ライトセンサー 2 を初期化
モーター 1 を 前 方向にセット
モーター 2 を 前 方向にセット

ずっと

もし **ライトセンサー 1 の値 > 100** なら

モーター 1 のスピードを 0 にする

でなければ

モーター 1 のスピードを 1000 にする

もし **ライトセンサー 2 の値 > 100** なら

モーター 2 のスピードを 0 にする

でなければ

モーター 2 のスピードを 1000 にする

0.1 秒待つ

ライントレース (テキスト)

```
1 motorEn = GPIO.new(12, GPIO::OUT)-
2 motorEn.on-
3 motor25 = GPIO.new(25, GPIO::OUT)-
4 motor26_pwm = PWM.new(26, ch=0)-
5 motor32 = GPIO.new(32, GPIO::OUT)-
6 motor33_pwm = PWM.new(33, ch=1)-
7 lux36 = ADC.new(36, ADC::ATTEN_11DB, ADC::WIDTH_12BIT)-
8 lux34 = ADC.new(34, ADC::ATTEN_11DB, ADC::WIDTH_12BIT)-
9 motor25.on-
10 motor32.on-
11 loop do-
12   if lux36.rawread > 100-
13     motor26_pwm.duty(0)-
14   else-
15     motor26_pwm.duty(1000)-
16   end-
17   if lux34.rawread > 100-
18     motor33_pwm.duty(0)-
19   else-
20     motor33_pwm.duty(1000)-
21   end-
22   sleep(0.1)-
23 end-
24
```

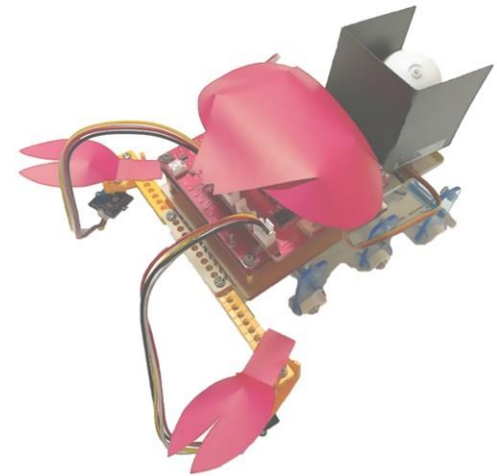
バックバック

コースを走る
プログラムを作ろう

KANIROBOT サンプルプログラムを見てね

KANIROBOT

サンプルプログラム



Let's Run !



大会スケジュール

1/15(日)

- 10:00 開場 受付開始 テスト走行開始
- 12:30 受付・試走会終了 ロボット展示
- 13:00 開会式
- 13:10 デザイン賞投票
- 13:30 予選開始
- 14:45 予選終了 本選出場者発表
- 15:00 本選（4チーム）
- 15:30 本選終了
- 15:40 閉会式
- 16:00 記念写真撮影

前日試走

1/14(土)

13:30~17:00

@オープンソースラボ

高専の学生がいます
アドバイスするよ！



お知らせ

1/7 (土) 8 (日) は、メンテナンスのため

<http://kanicon.epi.it.matsue-ct.jp/>

は使えません。

<http://pluto.epi.it.matsue-ct.jp:8601/>

を使ってください。

※会場のバックアップサーバは今日 (1/5) しか使えません