

かにロボプログラミング 簡易版

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

「かにロボ1v2」ブロック

練習 1：モーターを動かす



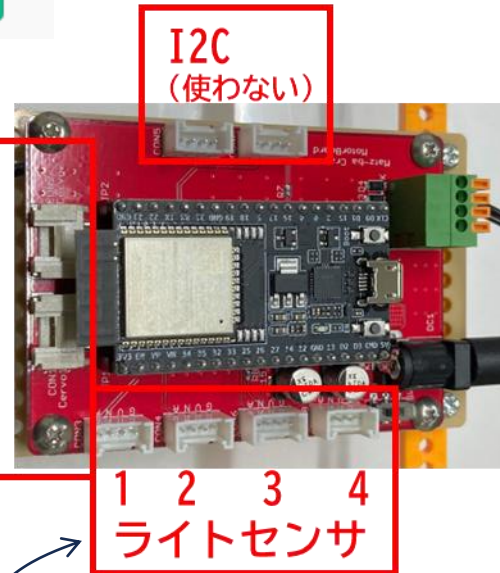
全速力で5秒動かす
ロボを床に置いたら
進むかな？

練習 2：サーボモータを動かす



番号を確認

2
1
サーボモータ



練習 3：ライトセンサの値の確認



白色と黒色の時の値を確認しよう。
この例では、右手にセンサー3、
左手にセンサー4を使ってます



本番：ライントレース

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



Scratch-like block diagram for a line-following robot:

- サーボモーター 1 を 0 度にする
- ループ: ライトセンサー 3 の値 > [] かつ ライトセンサー 4 の値 > [] まで繰り返す
 - もし ライトセンサー 3 の値 > [] なら
 - モーター 右 を 前 方向にセット. スピード 0 %
 - でなければ
 - モーター 右 を 前 方向にセット. スピード 100 %
 - もし ライトセンサー 4 の値 > [] なら
 - モーター 左 を 前 方向にセット. スピード 0 %
 - でなければ
 - モーター 左 を 前 方向にセット. スピード 100 %
 - [] 秒待つ (必ず待ち時間を入れる. 0.1秒くらい?)
- モーター 右 を 前 方向にセット. スピード 100 %
- モーター 左 を 前 方向にセット. スピード 100 %
- 2 秒待つ
- モーター 右 を 前 方向にセット. スピード 0 %
- モーター 左 を 前 方向にセット. スピード 0 %
- サーボモーター 1 を 90 度にする
- 1 秒待つ
- サーボモーター 1 を 0 度にする

センサーに合わせて
数字を調整しよう！

必ず待ち時間を入れる. 0.1秒くらい?

サーボモータを動かした後は、
必ず角度を元に戻しておく

さらにUターンしてスタート地点まで帰って
来るには、どのように書けば良いだろう？

プログラミングツールの使い方

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

[1] 左下の拡張機能  から必要な機能を選択

- かにロボ1v2：必要最低限のブロック
- かにロボ2：細かな調整をするためのブロック。

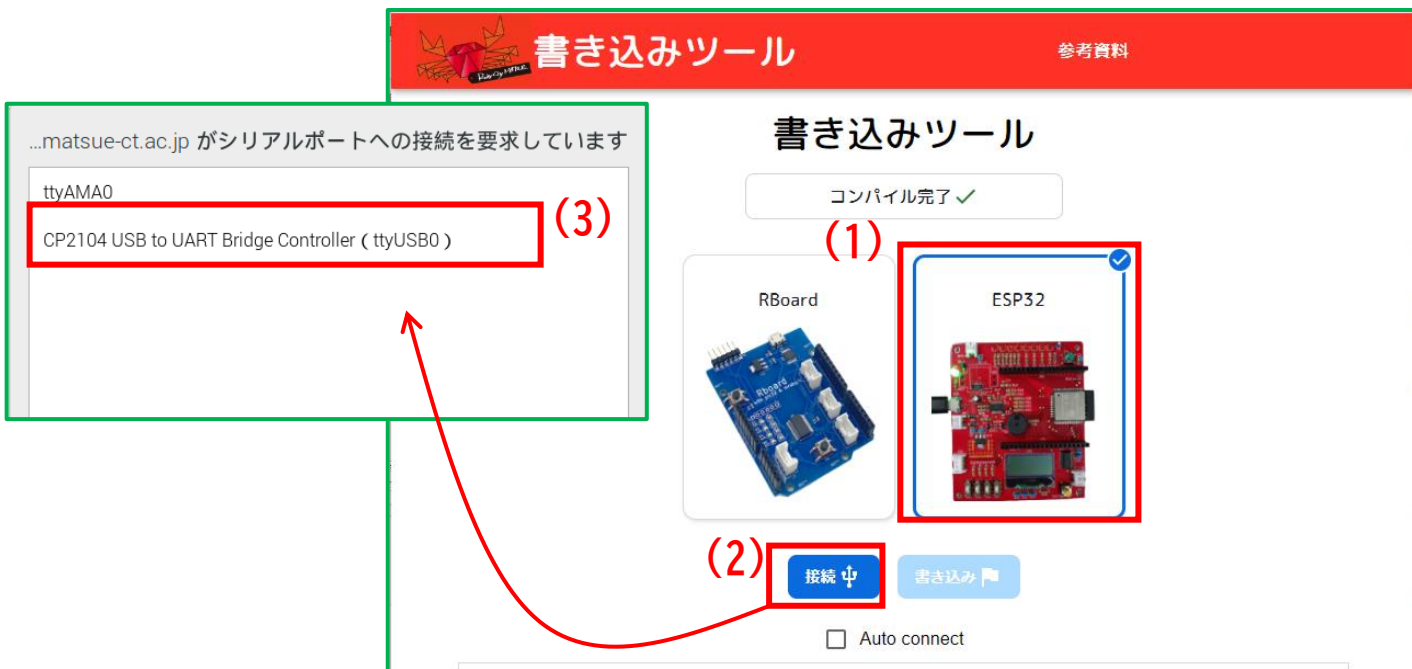
テキストプログラミング（Ruby）への移行を意識

- Tools：センサの値の確認に利用するブロック

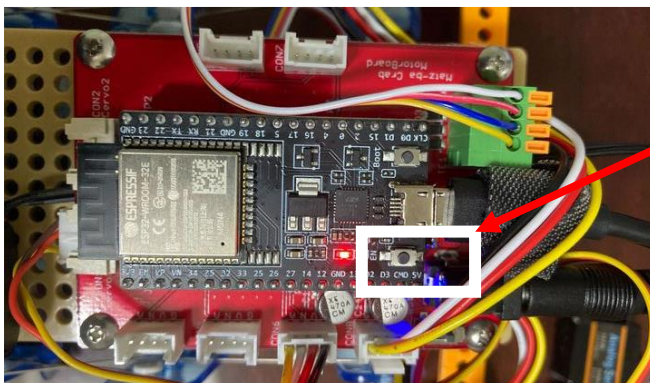
注) かにロボ1v2 とかにロボ2 のブロックを混ぜて使わないこと

[2] ブロックでプログラムを作ったら，緑の旗  をクリック

[3] 別タブで「書き込みツール」が立ち上がるので，以下の手順でロボット（マイコン）をパソコンに接続する



[4] ロボット（マイコン）の「EN ボタン」を押す



「EN」ボタン

プログラミングツールの使い方

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

- [5] 書き込みツールの「書き込み」ボタンの色が変わって押せるようになるので、それをクリックする。



- [6] ロボットが動き出す！

★ 作ったプログラムの保存を忘れずに！

プログラミングツールの動作環境

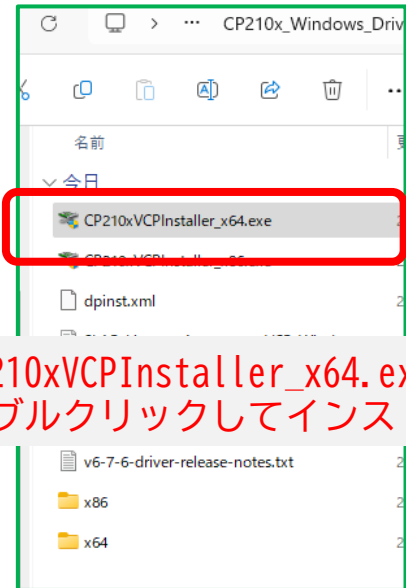
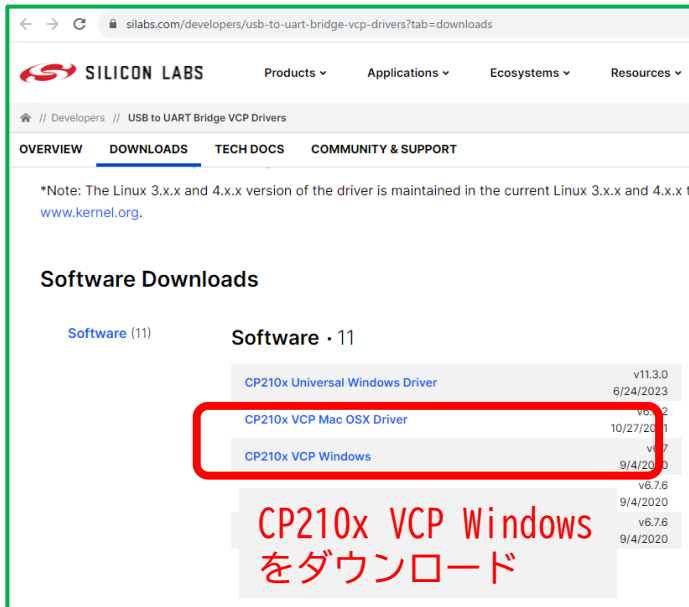
動作確認済 (2023/12/01 現在)

OS : Windows 10, 11, Raspberry Pi OS (Bookworm)

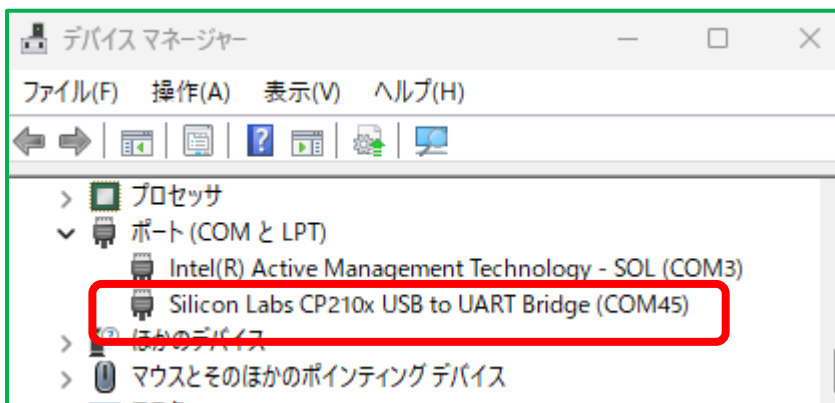
ブラウザ : Google Chrome, Chromium

Windows利用時にはデバイスドライバのインストールが必要です

- デバイスドライバのインストールが必要
- 「silicon labs CP210x」で検索して、Webよりダウンロード
<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>
- ダウンロードした zip ファイルを解凍し (右クリックして「すべて展開」), その中に入っているインストーラを実行



- ロボット (マイコン) をパソコンに接続した後、デバイスマネージャを起動する。ポート (COMとLPT) に CP210x が表示されるか確認する



かにロボプログラミング 補足(1)

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

「かにロボ」ブロック

練習 1：モーターを動かす



練習 2：サーボモータを動かす



かにロボプログラミング 補足(1)

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

練習 3：ライトセンサの値の確認



かにロボプログラミング 補足(2)

<https://ceres.epi.it.matsue-ct.ac.jp/smt/>

Ruby コード

練習 1：モーターを動かす

```
motor25 = GPIO.new(25, GPIO::OUT)
motor32 = GPIO.new(32, GPIO::OUT)
motor26_pwm = PWM.new(26, ch=0)
motor33_pwm = PWM.new(33, ch=1)
motor25.write(1)
motor32.write(1)
motor26_pwm.duty( 0 )
motor33_pwm.duty( 0 )
sleep(5)
motor26_pwm.duty(1023)
motor33_pwm.duty(1023)
```

} 初期化 (1回だけ行う)

スピード 100%

スピード 0%

練習 2：サーボモータを動かす

```
servo27 = PWM.new(27, ch=3)
servo27.freq(80)
servo27.duty(50)
sleep(1)
servo27.duty(150)
sleep(1)
servo27.duty(50)
```

} 初期化 (1回だけ行う)

0 度に

90 度に

0 度に

練習 3：ライトセンサの値の確認

```
lux35 = ADC.new(35, ADC::ATTEN_11DB, ADC::WIDTH_12BIT)
lux2 = ADC.new(2, ADC::ATTEN_11DB, ADC::WIDTH_12BIT)
loop do
  puts( lux35.rawread )
  puts( lux2.rawread )
  sleep(1)
end
```

} 初期化 (1回だけ行う)