

簡単なロボットプログラムを作ろう

目次

簡単なロボットプログラムを作ろう

目次

はじめに

対象

作って欲しいもの

これからやること

プログラム構造

ステート変数

LEDチカチカ

スイッチによりチカチカ

実際のロボット

やること

関数の作成

やること

それぞれの処理を作成する

通信処理

入力バッファ

モータ動作

PWM

ライントレース

はじめに

対象

C言語の

- for
- if else 文
- int float bool変数
- リスト
- 関数

がわかってる人向けです。

作って欲しいもの

```
int state=0;
void setup() {
  // 通信用設定(Serial)
}

void loop() {

  // 通信処理
  // ロボット付属のボタン処理
  if(state==0){
  // 停止処理
  }else if (state==1){
  // ライントレースで駆動
  }else if (state==2){
  // 段差超え
  }else if (state==3){
  // ボール確保
  }
}

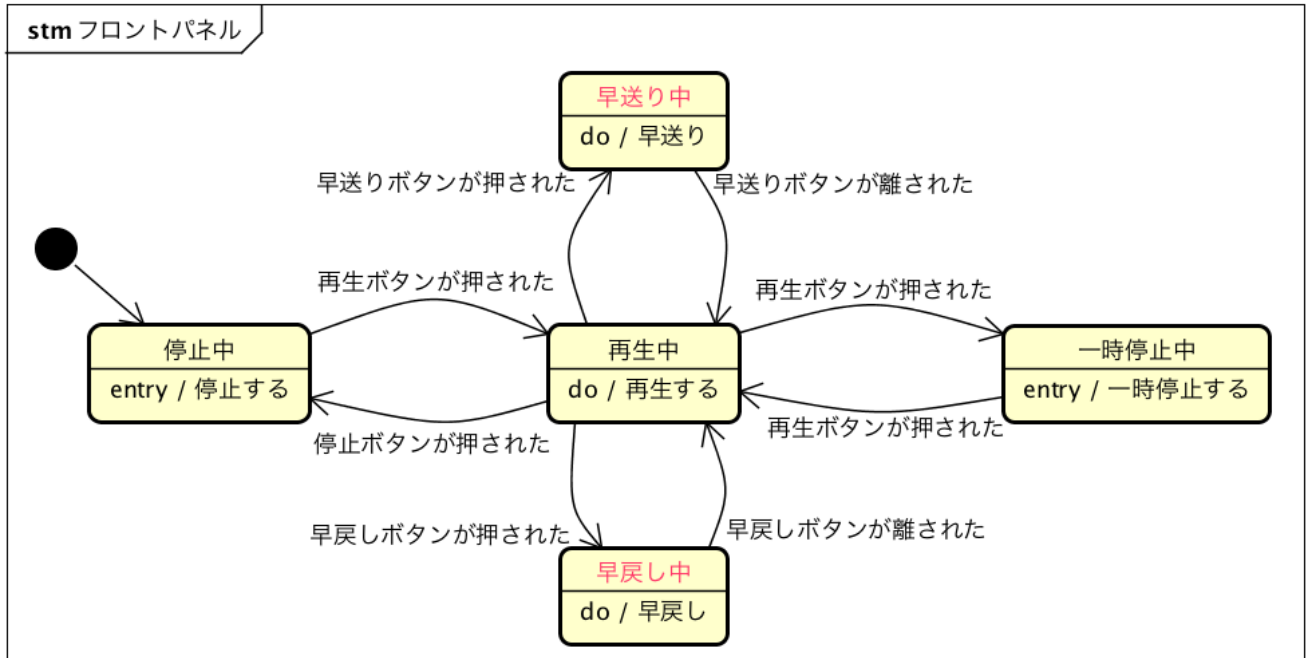
void drive_motor(duty,port){
  // モータ処理 duty:1~0 のfloat。パワーの指定。 port:制御するモータの指定
}

void drive_under(rad,speed_max){
  // 4輪処理 モータ処理を利用する。角度指定で動作させる。(足回りの型によって
  計算式が異なる)
}

void input_lt(){
  // ライントレースセンサ処理
```

```
//return どのセンサが反応したかを返す。  
}
```

あと、いいと思ったらステートマシン(状態遷移図)を作成してみるといい設計ができると思います。



これからやること

1. プログラム構造を学ぶ & 設計する
2. それぞれの関数を制作する

ここでは、モータとライントレースとコントローラを紹介します。めんどくさくてリンクだけにすることも

3. 実際に動作させ、調整する

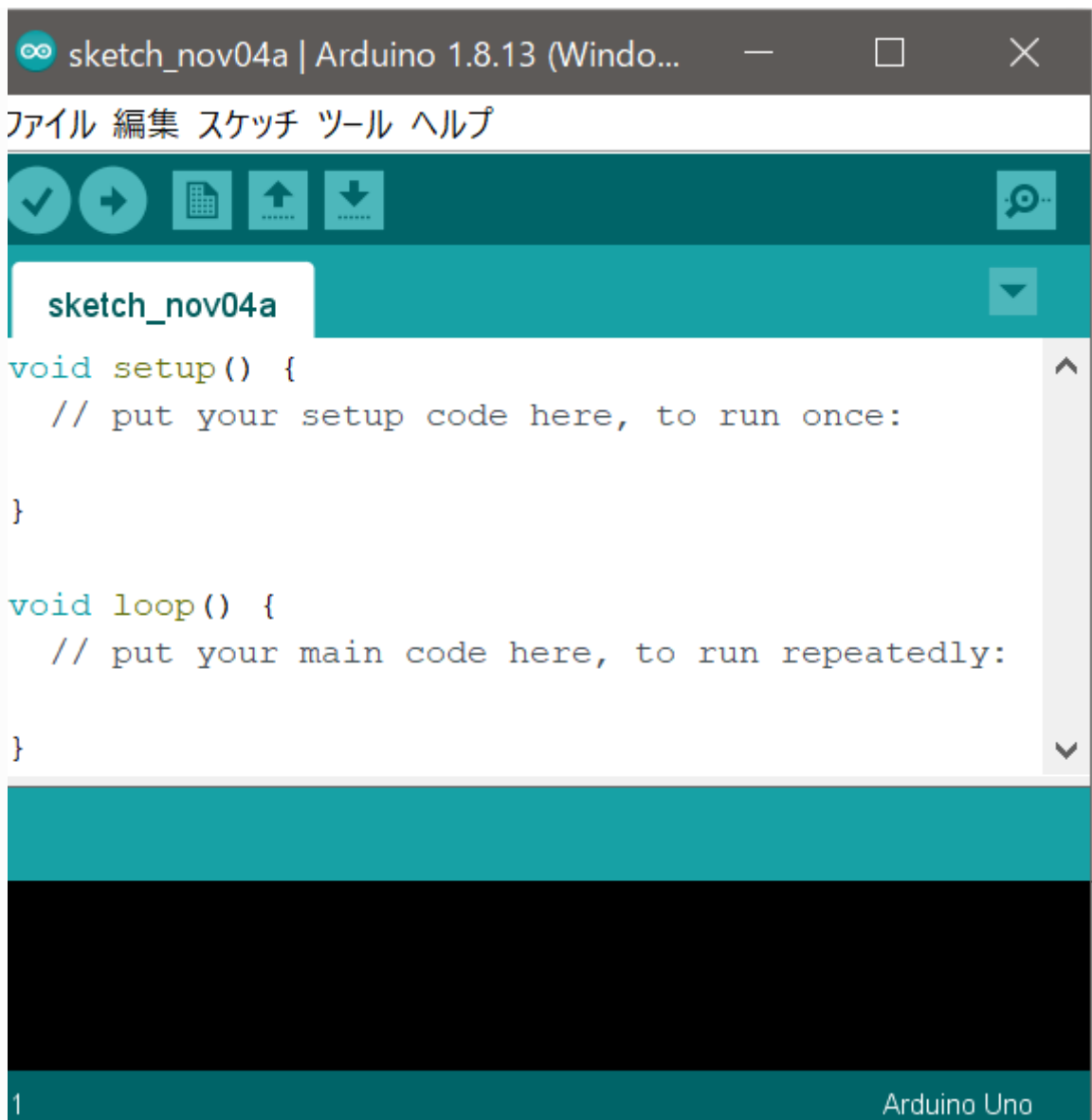
プログラム構造

プログラムを書く上で、考えることです

ステート変数

ステートというのは状態という意味です、状態を表す変数を置くように心がけましょう。

Arduino IDE では、ループ関数が設けられていますよね。ロボットプログラムは常に何かしらの処理をループさせています。



LEDチカチカ

LEDチカチカならば、光る/消す を繰り返すわけです。

その場合は、LEDの状態を変数を用意して記憶させ、条件分岐させると思います。

スイッチによりチカチカ

この場合は、スイッチの状態をループで毎回読み取って、条件分岐させますね。

実際のロボット

スイッチによりチカチカをもっともっと複雑で、いろんな処理が入ることになります。 そうなると、プログラムがとても見にくくなるのです。同じ処理がしたい場合も出てくるでしょう。

やること

こんなかんじに、やること別に処理を分けて、わかりやすくします。

```
int state=0;
void setup() {

}

void loop() {
// 通信処理
// ロボット付属のボタン処理 手動でstateを切り替え テスト走行や、リトライ
// に役立つ。
    if(state==0){
// 停止処理
    }else if (state==1){
// ライントレースで駆動
// 目標に到達したらstate=2,エラーが生じたらstate=0
    }else if (state==2){
// 段差超え
    }else if (state==3){
// ボール確保
    }
}
```

`int state` 変数 をグローバル化しています。通信処理,ボタン処理,センサ処理,などで、このstateを変えて、次の動作に進んだり、停止したりできます。

関数の作成

やること

各処理の中で、繰り返すものを関数としておいてください。考えられるものをいかに挙げておきます。

```

void drive_motor(duty,port){
// モータ処理 duty:1~0 のfloat。パワーの指定。 port:制御するモータの指定
}

void drive_under(rad,speed_max){
// 4輪処理 モータ処理を利用する。角度指定で動作させる。(足回りの型によって
計算式が異なる)
}

void input_lt(){
// ライントレースセンサ処理
//return どのセンサが反応したかを返す。
}

```

それぞれの処理を作成する

通信処理

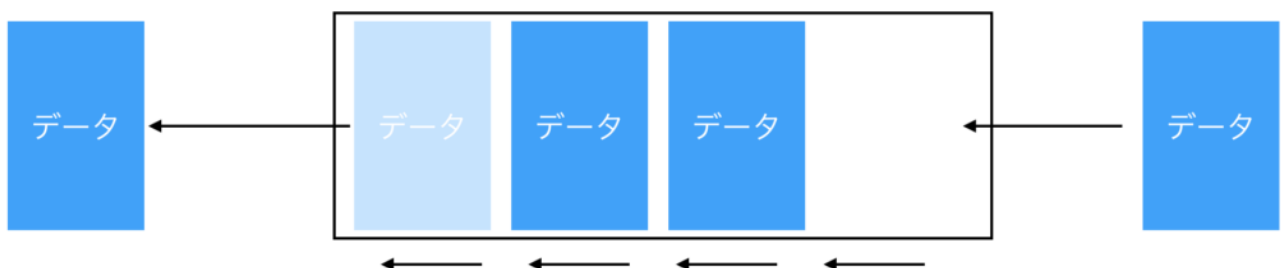
必要なことは2つです

- 初期設定を セットアップ関数(void setup)に書き込む
- ループ関数で入力バッファにあるものを読み込む

入力バッファ

こんな感じです

キュー(Queue)



モータ動作

モータの動作について、表のところを参考にします。

Hブリッジドライバの仕組み

Arudinoから2ピン出力して、モータを制御するわけであります。

制御信号線の省略

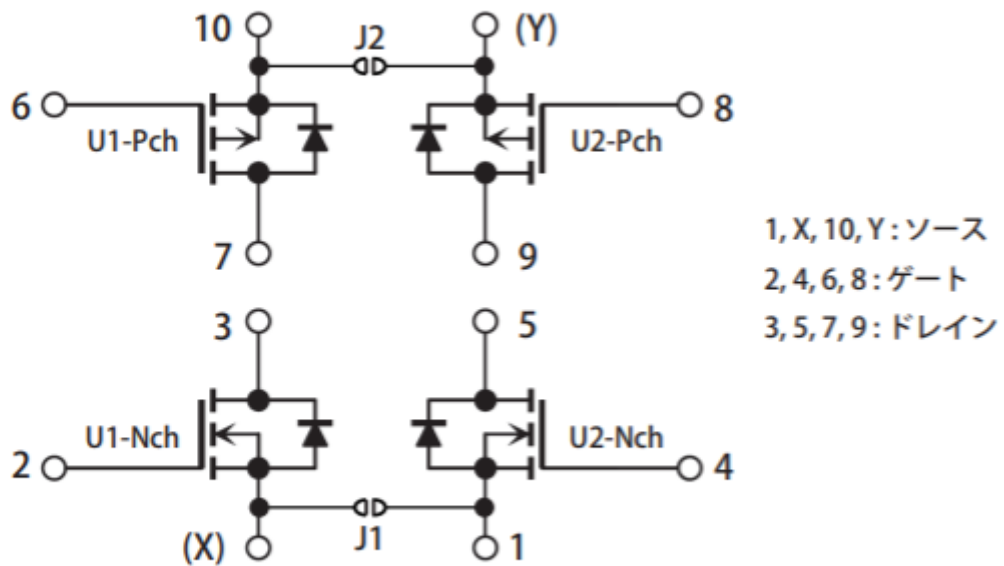
これで、2本の信号線のHIGH/LOWの切り替えで回転方向を制御する事が出来る様になります。
※ 図中では素子の名称をモーターの上側のPNP型トランジスタを「Tr_P1」「Tr_P2」、モーター下側のNPN型トランジスタを「Tr_N1」「Tr_N2」と置き換えてあります。
この置き換えた名称を使って、信号線2つの組み合わせとトランジスタの動作を確認すると次の様になります。

Arduino_PIn1	Arduino_PIn2	Tr_P1	Tr_P2	Tr_N1	Tr_N2	モーター動作
LOW	LOW	ON	ON	OFF	OFF	停止
HIGH	LOW	OFF	ON	ON	OFF	1→2方向
LOW	HIGH	ON	OFF	OFF	ON	2→1方向
HIGH	HIGH	OFF	OFF	ON	ON	停止

回路の子に、以下のICを用いて同じ回路を作ってもらいます。

利用するIC

◆回路図（等価回路）



※MP4242のピンを交換（コンパチブル）して使用する場合

PWM

モータの出力の調整を行います。LEDの調光と同じ原理。

LEDの調光

ライントレース

赤外線を照射できるLEDと、赤外線を検知できるセンサを組み合わせます。

ライントレース

使う赤外線センサ