

```

/*****
   スタートゲート制御基板用スケッチ
   LCD Keypad Shield 使用
   *****/

//インクルード
#include <Servo.h>
#include <LiquidCrystal.h>

/*select the pins used on the LCD panel
   lcd の使っているピン番号
   LiquidCrystal(rs, enable, d4, d5, d6, d7)
   rs: LCD の RS ピンに接続する Arduino 側のピン番号
   rw: LCD の RW ピンに接続する Arduino 側のピン番号
   enable: LCD の enable ピンに接続する Arduino 側のピン番号
   d0~d7: LCD の data ピンに接続する Arduino 側のピン番号

   d0~d3 はオプションで、省略すると 4 本のデータライン(d4~d7)だけで制御します。 */
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

//サーボ
Servo myservo1;
Servo myservo2;

//定義
#define analogswitch A0
#define LEFT    0
#define UP      1
#define DOWN    2
#define RIGHT   3
#define SELECT  4
#define NONE    5
#define sensor1 2 //ゲート 1 回路が少ないほう(出口) 41 OUT
#define sensor2 A3 //ゲート 1 回路が多いほう(入口) 41D
#define sensor3 3 //ゲート 2 回路が少ないほう(出口) 41 IN
#define sensor4 A4 //ゲート 2 回路が多いほう(入口) 41D

//時間計算
unsigned long Itimemillis = 0;
unsigned long Itimemicros = 0;
unsigned long Otimemillis = 0;

```

```
unsigned long Otimemicros = 0;
unsigned long Itime1 = 0;
int Itime2 = 0;
unsigned long Itime3 = 0;
unsigned long Otime1 = 0;
int Otime2 = 0;
unsigned long Otime3 = 0;
```

```
// 分秒表示
```

```
int Icnts = 0;
int Icntm = 0;
int Ocnts = 0;
int Ocntm = 0;
```

```
//タイマー表示 2 周目 3 周目
```

```
int secondOTIME = 0;
int thirdOTIME = 0;
int secondITIME = 0;
int thirdITIME = 0;
int secondOcntM = 0;
int secondOcntS = 0;
int thirdOcntM = 0;
int thirdOcntS = 0;
int secondIcntM = 0;
int secondIcntS = 0;
int thirdIcntM = 0;
int thirdIcntS = 0;
```

```
//時間その他
```

```
int IcntIN = 0;
int OcntOUT = 0;
```

```
//パターン
```

```
int INTimepattern = 4;
int OUTTimepattern = 4;
int INsensorpattern = 0;
int OUTsensorpattern = 0;
int gate1pattern = 0;
int gate2pattern = 0;
int pattern;
```

```

//センサー
int sensor30;
int sensor10;

//センサーカウント
int sensorINcnt = 0;
int sensorOUTcnt = 0;

//processing 変数宣言
int outTime_m = 0; // OUT コース:分
int outTime_s = 0; // OUT コース:秒
int outTime_c = 0; // OUT コース:センチ秒
int inTime_m = 0; // IN コース:分
int inTime_s = 0; // IN コース:秒
int inTime_c = 0; // IN コース:センチ秒

int button() {
    int pushbutton;
    pushbutton = (analogRead(analogswitch) / 4);
    if (pushbutton > 240) return NONE;
    if (pushbutton < 10) return RIGHT;
    if (pushbutton < 60) return UP;
    if (pushbutton < 110) return DOWN;
    if (pushbutton < 170) return LEFT;
    if (pushbutton < 210) return SELECT;
    // return NONE;
}

void timerOUT() {
    pattern = button();
    switch (OUTTimepattern) {
        case 0:
            //ミリ秒基準
            Otimemillis = millis();
            Otime2 = Otimemillis - Otime1;
            if (Otime2 >= 1000) {
                Otime1 = Otimemillis;
                Ocnts += 1;
            }
    }
}

```

```
if (Ocnts >= 60) {
    Ocnts = 0;
    Ocntm += 1;
}
if (OUTsensorpattern == 31) {
    secondOTIME = Otime2;
    secondOcntS = Ocnts;
    secondOcntM = Ocntm;
    outTime_c = secondOTIME;
    outTime_s = secondOcntS;
    outTime_m = secondOcntM;
}
if (OUTsensorpattern == 60) {

    OUTTimepattern = 2;
}
break;
```

case 2:

```
break;
```

case 4:

```
Otimemillis = millis();
Otime1 = Otimemillis;
Otime2 = Otimemillis - Otime1;
Ocnts = 0;
Ocntm = 0;
secondOTIME = 0;
secondOcntS = 0;
secondOcntM = 0;
outTime_c = secondOTIME;
outTime_s = secondOcntS;
outTime_m = secondOcntM;
break;
```

```
}
```

```
}
```

```
void sensorOUTpattern() {
    sensoro();
```

```
pattern = button();
switch (OUTsensorpattern) {
  case 0:
    sensorOUTcnt = 0;
    gate2pattern = 1;
    if ( pattern == 1) {
      gate2pattern = 0;
      OUTsensorpattern = 1;
    }
    break;

  case 1:
    sensorOUTcnt = 0;
    if ( pattern == 4 ) {
      gate2pattern = 1;
      OUTsensorpattern = 20;
    }
    break;

  case 10:
    sensorOUTcnt = 0;
    if (sensor10 == 1) {
      OUTsensorpattern = 20;
    }
    break;

  case 20://1 回目通過
    sensorOUTcnt++;
    OUTTimepattern = 0;
    if (sensorOUTcnt >= 1700) {
      sensorOUTcnt = 0;
      OUTsensorpattern = 30;
    }
    break;

  case 30://
    if (pattern == 0) {
      sensorOUTcnt = 0;
      OUTsensorpattern = 70;
    }
}
```

```
if (sensor10 == 1 ) {
    sensorOUTcnt = 0;
    OUTsensorpattern = 31;//31 にすると 2 回目通過のタイムを PC に表示
}

break;

case 31:
    sensorOUTcnt++;
    if (sensorOUTcnt >= 1) {
        sensorOUTcnt = 0;
        OUTsensorpattern = 40;
    }
    break;

case 40://
    sensorOUTcnt++;
    if (sensorOUTcnt >= 1700 ) {
        sensorOUTcnt = 0;
        OUTsensorpattern = 30;
    }
    break;

case 50://ゴール後
    if (sensor10 == 1 ) {
        OUTTimepattern = 2;
        sensorOUTcnt = 0;
        OUTsensorpattern = 60;
    }
    break;

case 60:
    sensorOUTcnt++;
    if (sensorOUTcnt >= 1) {
        sensorOUTcnt = 0;
        OUTsensorpattern = 61;
    }
    break;

case 61:
```

```

    if (pattern == 0) {
        sensorOUTcnt = 0;

        OUTsensorpattern = 70;
    }
    break;

case 70:
    OUTTimepattern = 2;
    sensorOUTcnt++;
    if (pattern == 4) {
        sensorOUTcnt = 0;
        OUTsensorpattern = 71;
    }
    break;

case 71:
    OUTTimepattern = 4;
    sensorOUTcnt++;
    if (sensorOUTcnt > 300) {
        gate2pattern = 0;
        sensorOUTcnt = 0;
        OUTsensorpattern = 1;
    }
    break;
}
}

void timerIN() {
    pattern = button();
    switch (INTimepattern) {
    case 0:
        Itimemillis = millis();
        Itime2 = Itimemillis - Itime1;
        if (Itime2 >= 1000) {
            Itime1 = Itimemillis;
            Icnts += 1;
        }
        if (Icnts >= 60) {
            Icnts = 0;

```

```
    Icntm += 1;
}
if (INsensorpattern == 31) {
    secondITIME = Itime2;
    secondIcntS = Icnts;
    secondIcntM = Icntm;
    inTime_c = secondITIME;
    inTime_s = secondIcntS;
    inTime_m = secondIcntM;
}
if (INsensorpattern == 60) {

    INTimepattern = 2;
}
break;
```

case 2:

```
break;
```

case 4:

```
Itimemillis = millis();
Itime1 = Itimemillis;
Itime2 = Itimemillis - Itime1;
Icnts = 0;
Icntm = 0;
secondITIME = 0;
secondIcntS = 0;
secondIcntM = 0;
inTime_c = secondITIME;
inTime_s = secondIcntS;
inTime_m = secondIcntM;
```

```
break;
```

```
}
}
```

```
void sensorINpattern() {
    sensori();
    pattern = button();
```

```
switch (INsensorpattern) {
  case 0:
    sensorINcnt = 0;
    gate1pattern = 1;
    if ( pattern == 1) {
      gate1pattern = 0;
      INsensorpattern = 1;
    }
    break;

  case 1:
    sensorINcnt = 0;
    if ( pattern == 4 ) {
      gate1pattern = 1;
      INsensorpattern = 20;
    }
    break;

  case 10:
    sensorINcnt = 0;
    if (sensor30 == 1) {
      INsensorpattern = 20;
    }
    break;

  case 20://1 回目通過
    sensorINcnt++;
    INTimepattern = 0;
    if (sensorINcnt >= 1700) {
      sensorINcnt = 0;
      INsensorpattern = 30;
    }
    break;

  case 30://
    if (pattern == 0) {
      sensorINcnt = 0;
      INsensorpattern = 70;
    }
    if (sensor30 == 1 ) {
```

```
    sensorINcnt = 0;
    INsensorpattern = 31;//31 にすると 2 回目通過のタイムを PC に表示
}

break;

case 31:
    sensorINcnt++;
    if (sensorINcnt >= 1) {
        sensorINcnt = 0;
        INsensorpattern = 40;
    }
    break;

case 40://
    sensorINcnt++;
    if (sensorINcnt >= 1700 ) {
        sensorINcnt = 0;
        INsensorpattern = 30;
    }
    break;

case 50://ゴール後
    if (sensor30 == 1 ) {
        INTimepattern = 2;
        sensorINcnt = 0;
        INsensorpattern = 60;
    }
    break;

case 60:
    sensorINcnt++;
    if (sensorINcnt >= 1) {
        sensorINcnt = 0;
        INsensorpattern = 61;
    }
    break;

case 61:
    if (pattern == 0) {
```

```
    sensorINcnt = 0;
    INsensorpattern = 70;
}
break;
```

case 70:

```
    INTimepattern = 2;
    sensorINcnt++;
    if (pattern == 4) {
        sensorINcnt = 0;
        INsensorpattern = 71;
    }
    break;
```

case 71:

```
    INTimepattern = 4;
    sensorINcnt++;
    if (sensorINcnt > 300) {
        gate1pattern = 0;
        sensorINcnt = 0;
        INsensorpattern = 1;
    }
    break;
```

```
}
}
```

```
void sensori() {
    if (sensor3 == LOW) {
        sensor30 = 1;
    }
    else {
        sensor30 = 0;
    }
}
```

```
void sensoro() {
    if (sensor1 == LOW) {
        sensor10 = 1;
    }
    else {
        sensor10 = 0;
    }
}
```

```

    }
}
void setup() {
  Serial.begin(250000);
  pinMode(sensor1, INPUT_PULLUP);
  pinMode(sensor3, INPUT_PULLUP);
  attachInterrupt(1, sensori, FALLING );
  attachInterrupt(0, sensoro, FALLING );
  myservo1.attach(12);
  myservo2.attach(13);
  lcd.begin(16, 2);           // start the library
  lcd.setCursor(0, 0);
  lcd.print("1OUT"); // print a simple message
  lcd.setCursor(0, 1); // move to the beginning of the second line
  lcd.print("2IN");
}

void loop() {
  switcher();
}

void switcher() {
  int pushbutton;
  int button1 = button();
  pushbutton = (analogRead(analogswitch) / 4);
  Serial.println(button1);
  Serial.println(pushbutton);
}

```