

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

```
//Sample using LiquidCrystal library  
#include <LiquidCrystal.h>  
#include <Boards.h>  
#include <Firmata.h>  
#include <Servo.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);
```

```
// define some values used by the panel and buttons
```

```
int pushbutton = 0;
```

```
//パターン
```

```
/*I = IN
```

```
    O = OUT
```

```
    cnts = second
```

```
    cntm = minute
```

```
    分秒表示*/
```

```
//時間計算
```

```
//時間
```

```
//時間その他
```

```
//定義
```

```
#define LEFT    0
```

```
#define UP      1
```

```
#define DOWN    2
```

```

#define RIGHT  3
#define SELECT 4
#define NONE   5

#define analogswitch A0

// read the buttons
int button() {
    pushbutton = (analogRead(analogswitch) / 4);
    if (pushbutton > 240) return NONE;
    if (pushbutton < 10) return  RIGHT;
    if (pushbutton < 50) return  UP;
    if (pushbutton < 100) return DOWN;
    if (pushbutton < 150) return LEFT;
    if (pushbutton < 200) return SELECT;
    // return NONE;
}

void timerIN() {

void timerOUT() {

void switcher() {
    pattern = button(); // read the buttons

    switch (pattern) {
        case LEFT:
            Itimemillis = millis();
            Itime1 = Itimemillis;
            Itime2 = Itimemillis - Itime1;
            Icnts = 0;
            Icntm = 0;
            break;

        case UP:
            if (INTimepattern < 1) {
                INTimepattern = 2;

```

```

    }
    /*    if (INTimepattern > 0) {
           INTimepattern = 0;
        }*/
    break;

    case NONE:

        break;
    }
}

```

```

void LCD() {
    lcd.setCursor(5, 1);
    lcd.print(Icntm);
    lcd.print(" ");
    lcd.print(Icnts);
    lcd.print(" ");
    lcd.print(Itime2);
    lcd.print("  ");
    lcd.setCursor(5, 0);
    lcd.print(Ocntm);
    lcd.print(" ");
    lcd.print(Ocnts);
    lcd.print(" ");
    lcd.print(Otime2);
    lcd.print("  ");
}

```

```

void printer() {
    Serial.print("IN");
    Serial.print(" ");
    Serial.print(Icntm);
    Serial.print(" ");
    Serial.print(Icnts);
    Serial.print(" ");
    Serial.print(Itime2);
    Serial.print(" ");
    Serial.print("OUT");
    Serial.print(" ");
    Serial.print(Ocntm);
}

```

```
Serial.print(" ");
Serial.print(Ocnts);
Serial.print(" ");
Serial.print(Otime2);
Serial.print(" ");
Serial.print(INTimepattern);
Serial.print(" ");
Serial.print(pattern);
Serial.print(" ");
Serial.println(IcntIN);
}
```

```
void setup()
{
  Serial.begin(250000);
  lcd.begin(16, 2);           // start the library
  lcd.setCursor(0, 0);
  lcd.print("OUT"); // print a simple message
  lcd.setCursor(0, 1); // move to the beginning of the second line
  lcd.print("IN");
}
```

```
void loop()
{
  timerIN();
  timerOUT();
  switcher();
  LCD();
  printer();
}
```