

# Report on the Experiment

No. 4

Subject 電子回路設計製作と特性試験

Date 2021. 04. 14

Weather 晴れ Temp °C Wet %

Class E5  
Group 4  
Chief  
Partner

No 14  
Name 小畠 一泰

Kure National College of Technology

## 1 実験テーマ

Arduino と LCD を用いた簡易時計の作成

## 2 目的

Arduino の基本的な使い方や割り込み処理についての理解を深めるとともに, LCD を用いて外部出力を行う.

## 3 実験機器

- マイコン: Arduino Uno
- LCD: 3.3 [V] 動作 LCD キャラクタディスプレイモジュール 16x2 行バックライト付白抜き (SC1602BBWB-XA-LB-G)
- 抵抗: 1 [k $\Omega$ ]
- タクトスイッチ

## 4 要件定義

- 1 秒ごとに画面が更新され hh:mm:ss のフォーマットで時刻が表示されること.
  - 12 時 34 分 56 秒の場合は 12:34:56 となること.
- 2 つのタクトスイッチがあること.
  - 時刻表示モード
    - \* NEXT\_CURSOR: 設定モードへ移行すること.
    - \* ADD: 動作未割り当て
  - 時刻設定モード
    - \* NEXT\_CURSOR: hh, mm, ss それぞれの下位もしくは決定, キャンセル (図 2 の下線部) へ移動すること.
    - \* ADD
      - ・ hh, mm, ss: それぞれ 1 時間, 1 分, 1 秒加算すること.
      - ・ 決定: 時刻を設定した時間へ変更し, 時刻表示モードへ移行すること.
      - ・ キャンセル: 時刻を変更せずに, 時刻表示モードへ移行すること.



hh:mm:ss

図 1: 時刻表示モード



図 2: 時刻設定モード

## 5 実験方法

### 5.1 回路

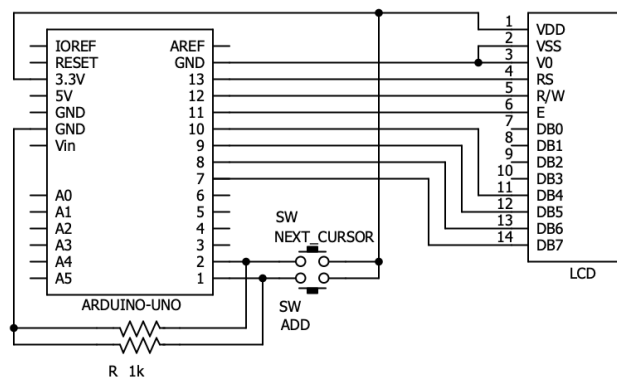


図 3: 回路図

図 3 のような回路をブレッドボード上などに組む。

実際に作成したブレッドボードが 図 4 である。

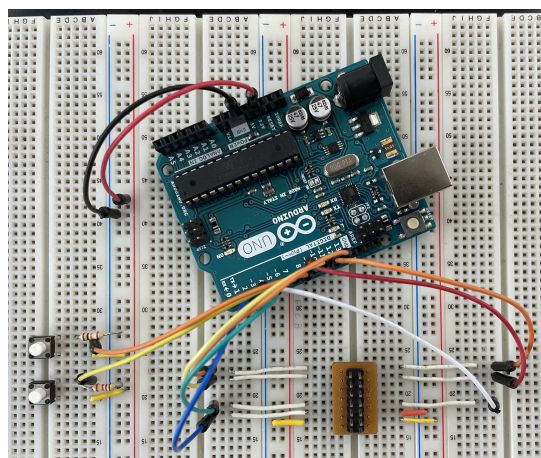


図 4: 実際の回路

## 5.2 ソースコード

---

### コード 1

---

```
1  #include <LiquidCrystal.h> // LCD 制御ライブラリ
2  #include <MsTimer2.h>      // 割り込み処理ライブラリ
3
4  // LiquidCrystal(RS, R/W, Enable, d4, d5, d6, d7)
5  LiquidCrystal lcd(13, 12, 11, 10, 9, 8, 7);
6
7  #define SELECT_NEXT_PIN 2
8  #define ADD_TIME_PIN 1
9
10 boolean isConfig = false;
11 unsigned long seconds = 0;
12 unsigned long tmp_seconds = 0;
13 int cursor = 0;
14
15 // col, row
16 char cursorPostions[5][2] = {
17     { 1, 0}, // hour
18     { 4, 0}, // minute
19     { 7, 0}, // second
20     {10, 0}, // OK
21     {10, 1}, // Cancel
22 };
23
24 void setup() {
25     pinMode(SELECT_NEXT_PIN, INPUT);
26     pinMode(ADD_TIME_PIN, INPUT);
27
28     MsTimer2::set(1000 /*[ms]*/, countUp);
29     MsTimer2::start();
30
31     lcd.begin(16, 2);
32     lcd.clear();
33     lcd.setCursor(0, 0);
34     lcd.print(formatTime(seconds));
35 }
```

---

---

## コード 2

---

```
1 void countUp() {
2     seconds++;
3     if (isConfig == false) {
4         lcd.clear();
5         lcd.print(formatTime(seconds));
6     }
7 }
8
9 // 秒数を受け取って hh:mm:ss にフォーマットする
10 String formatTime(unsigned long raw_s) {
11     int h = raw_s / 3600;
12     int m = raw_s / 60 % 60;
13     int s = raw_s % 60;
14
15     char time[20];
16     sprintf(time, "%02d:%02d:%02d", h, m, s);
17     return time;
18 }
19
20 // 時刻の加算をする、各桁がそれぞれの最大 (hh:23, mm:ss: 59) を超えたら 0 に戻す
21 void addTime() {
22     if (cursor == 0) {
23         if (((tmp_seconds + 3600) / 3600) < 24) tmp_seconds += 3600;
24         else tmp_seconds -= tmp_seconds / 3600 * 3600;
25     }
26     if (cursor == 1) {
27         if (((tmp_seconds + 60) / 60 % 60) != 0) tmp_seconds += 60;
28         else tmp_seconds -= (tmp_seconds / 60 % 60) * 60;
29     }
30     if (cursor == 2) {
31         if (((tmp_seconds + 1) % 60) != 0) tmp_seconds += 1;
32         else tmp_seconds -= tmp_seconds % 60;
33     }
34
35     lcd.clear();
36     lcd.print(formatTime(tmp_seconds) + " OK");
37     lcd.setCursor(0, 1);
38     lcd.print("Cancel");
39     lcd.setCursor(cursorPostions[cursor][0], cursorPostions[cursor][1]);
40 }
```

---

---

### コード 3

---

```
1 void ok() {
2     seconds = tmp_seconds;
3     isConfig = false;
4 }
5
6 void cancel() {
7     isConfig = false;
8 }
9
10 void next_cursor() {
11     cursor = ++cursor % 5;
12     lcd.setCursor(cursorPostions[cursor][0], cursorPostions[cursor][1]);
13 }
14
15 void loop() {
16     // 時刻表示モード
17     if (isConfig == false) {
18         if (digitalRead(SELECT_NEXT_PIN) == HIGH) {
19             // 時刻設定モードへ移行
20             isConfig = true;
21             tmp_seconds = seconds;
22             lcd.clear();
23             lcd.print(formatTime(tmp_seconds) + " OK");
24             lcd.cursor();
25             lcd.setCursor(0, 1);
26             lcd.print("          Cancel");
27             lcd.setCursor(cursorPostions[cursor][0], cursorPostions[cursor][1]);
28         }
29         lcd.noCursor();
30     }
31
32     // 時刻設定モード
33     if (isConfig == true) {
34         lcd.cursor();
35
36         if (digitalRead(SELECT_NEXT_PIN) == HIGH) {
37             next_cursor();
38             delay(300);
39         }
40
41         if (digitalRead(ADD_TIME_PIN) == HIGH) {
42             if (cursor <= 2) addTime();
43             if (cursor == 3) ok();
44             if (cursor == 4) cancel();
45             delay(300);
46         }
47     }
48 }
```

## 6 成果物



図 5: 時刻表示モード



図 6: 時刻設定モード

## 7 まとめ

要件通りに設計, 実験することができ成果物として申し分ないものができたと考える。また普段あまり触る機会のない LCD や割り込み処理を使うことで, Arduino の基本的な使い方だけでなくより深い知識, 知見を得ることができ非常に理解が高まった。

## 8 参考文献

- SC1602BBWB-XA-LB-G データシート
  - <https://akizukidenshi.com/download/ds/sunlike/SC1602BBWB-XA-LB-G.pdf>
- LiquidCrystal Library の各ファンクション | ArduinoRef | 電子工作 (MAKE)
  - <https://www.denshi.club/make/arduinoorg/arduino-ref01c.html>
- Arduino を用いて LCD に文字を表示 - Arduino - 基礎からの IoT 入門
  - <https://iot.keicode.com/arduino/arduino-project-lcd.php>