

情報システム工学実験 I 2023 レポート

2022531033 22 班 関川謙人

2024 年 1 月 25 日

1 プログラムの概要

1.1 概要と目的

課題 8 で制作した LED を使ったラーメンタイマーを改良したプログラム。このタイマーは分単位での表記、操作ができるタイマーである。

- 残り時間を分単位、秒単位に分けて測ることで残り時間をわかりやすくする。
- 一分調節すると、測る時間を 60 秒調節できる

以上の特徴により、秒単位だけで時間を測るタイマーよりわかりやすく、効率的に 3 分以内の時間を測ることを目的としている。6 つの赤 LED が秒単位を、2 つの緑 LED が分単位を表記する。

1.2 使用法

始めに測る時間を指定する。この段階では、以下の操作ができる。

- 上から 2 番目のボタンで 1 秒増やす。
- 上から 3 番目のボタンで 1 秒減らす。
- 下から 3 番目のボタンで 1 分増やす。
- 下から 2 番目のボタンで 1 分減らす。
- 一番下のボタンでタイマーをリセット、時間を 0 にする。
- 一番下のスタートボタンを押し、タイマーを作動させる。

以下の図はボタンと LED の配置、役割、割り当てられたポートについて説明したものである。

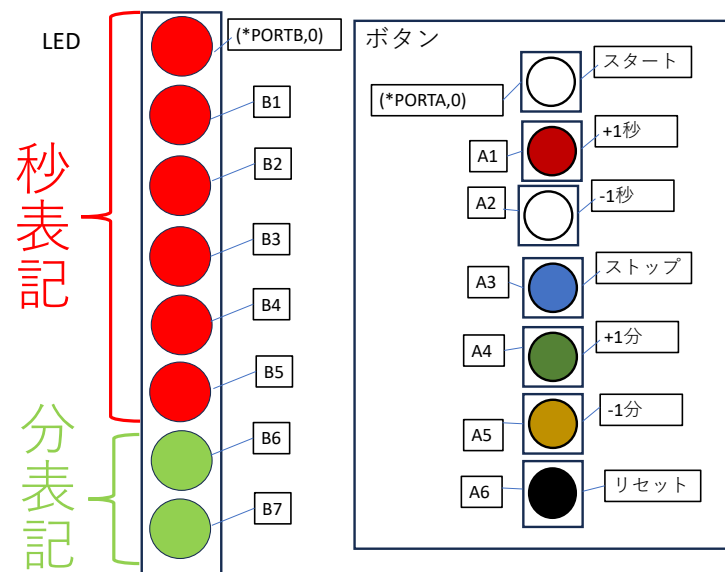


図1 ボタンとLEDの配置、役割

タイマーを作動させた後真ん中のストップボタンを押すと、タイマーが止まり時間指定を行う段階に戻る。

タイマーが時間を測り終わったとき、全てのLEDが1秒間隔で点滅する。時間指定を行う時、タイマーが作動している時の両方において、LEDは残り時間を表す。

2 プログラム作成の上での工夫、苦労

オリジナルプログラムの作成はLEDの色を変え、分と秒を別々に表し、別々に操作できるように工夫した。それにはそこまで苦労しなかった。

課題をこなすのには苦労した。始めは容易に課題を解決できたので、ペアと別々にプログラムを組み一気に課題を進める計画を組んだが課題6で詰まってしまった。具体的には、ボタンを押してもLEDの移り変わる速さが変わらない。その問題が解決してもLEDが一つの方向にしか動かないなどのエラーが起き、その問題がいつまでも解決せず次に進めなかったのである。原因はペアにプログラミングを任せるとき

- 変数名を適当に付けてしまっていた。
- コメントを怠っていた

ことによってプログラムの流れが読みにくくなってしまったことである。ペアと一緒にプログラムを組み、変数名やコメント、インデントなど打ち合わせをきちんとしながらプログラミングするべきであった。

このことから自らの能力を過信することの危険性、適切なコードを書くことの重要性、複数人で協力することでミスや見落としによるエラーを防げることを身に染みて学んだ。自分がプログラミ

ングを担当するときには以下の工夫をした。

- 何を扱う変数かをわかりやすくするために変数名を独自に考える。
- コメントを使用してプログラムに細かく区切りをつける

これによってデバッグが簡単にできるように努めた。

2.1 反省

講義が終了した後に気付いた反省を以下に示す。

- 時間を指定する段階のコードを先に配置すべきであった。
- 変数、コメントは工夫したが、インデントはできていなかった。
- 秒数の変数を count ではなく second にすべきであった
- 分数を減らす条件を count=-1 にする理由を理解せずに count=-1 に設定してしまった。
- 終了条件がなぜ minute=-1 なのかをコメントで説明していなかった。
- ストップボタンでのみタイマーを止める仕様にするつもりであったため、時間を調整するた
びに変数 trig の値を 0 にしてタイマーを止める必要性はなかった。
- ローカル関数を使用するのであれば、細かい処理をまとめてコードを読みやすくする形で使
用すべきだった。

3 ソースコード

3.1 ソースコードの補足説明

終了条件が minute=-1 となっているのは、終了条件を minute=0 にすると minute=1 から minute=0 になるときに、時間を測り終わっていないのにタイマーが終了してしまったからである。しかし一分減らす操作をタイマーを終了させる判定の後に置けば、終了条件を minute=0 に設定しても正常に動かせたのではないかと考えている。

また一分減らす時に秒数の残り時間を 60 秒でなく 59 秒にしている理由は、秒数のカウントを 0-59 秒の範囲で行っており、この時残り時間を 60 秒にしまうと 61 秒カウントすることになってしまうからである。

3.2 ソースコード

制作したプログラムのソースコードを、以下に示す。

Listing 1 original.c

```
1 //  
2 // Seminar :Key & LED Program for 16F84A  
3 // 7/3/2003
```

```

4 // Key & LED Replace Version 11/05/2003
5 #include <16f819.h>
6 #fuses NOWDT,NOPROTECT,NOPUT,INTRC_IO,NOMCLR
7 #use delay(clock=8000000)
8 #include <stdlib.h>
9 #include <stlled.c> // External Subroutine Files, Include
10
11 // 16F84 Definition
12
13 #DEFINE PORTA 0x05 // Define Port A Reg. Address
14 #DEFINE PORTB 0x06 // Define Port B Reg. Address
15 // #DEFINE PORTC 0x07 // Define Port C Reg. Address
16 #DEFINE STATUS 0x03 // Define STATUS Reg. Address
17 #DEFINE TRISA 0x85 // Define TRISA Reg. Address
18 #DEFINE TRISB 0x86 // Define TRISB Reg. Address
19 // #DEFINE TRISC 0x87 // Define TRISC Reg. Address
20
21 //
22 // Hard Ware Definition RA 3-0
23 //
24 #define _KEY0 0 // L:No Push, H:Push(Activate)
25 #define _KEY1 1 // L:No Push, H:Push(Activate)
26 #define _KEY2 2 // L:No Push, H:Push(Activate)
27 #define _KEY3 3 // L:No Push, H:Push(Activate)
28 #define _KEY4 4 // L:No Push, H:Push(Activate)
29 //
30 // Hard Ware Definition RB 6-0
31 //
32 #define _LED0 0 // LED0 (Connect to RB0), H:ON, L:OFF
33 #define _LED1 1 // LED1 (Connect to RB1), H:ON, L:OFF
34 #define _LED2 2 // LED2 (Connect to RB2), H:ON, L:OFF
35 #define _LED3 3 // LED3 (Connect to RB3), H:ON, L:OFF
36 #define _LED4 4 // LED4 (Connect to RB4), H:ON, L:OFF
37 #define _LED5 5 // LED5 (Connect to RB5), H:ON, L:OFF
38 #define _LED6 6 // LED6 (Connect to RB6), H:ON, L:OFF
39 #define _LED7 7 // LED7 (Connect to RB7), H:ON, L:OFF
40 //
41 // Flag Condition Definitions
42 //
43 #define K0 0 //Key 0 Flag Bit
44 #define K1 1 //Key 1 Flag Bit
45 #define K2 2 //Key 2 Flag Bit
46 #define K3 3 //Key 3 Flag Bit
47 #define K4 4 //Key 4 Flag Bit
48 //
49 // IO Init : Subroutines Example
50 //
51 void INT_IO(void){
52     #use fast_io(A)
53     #use fast_io(B)
54     // #use fast_io(C)
55     // RA4, RA3, RA2, RA1, RA0, Push SW Info Obtain. Set to INPUT.
56     // TRISA ***1 1111 0x1F
57
58     set_tris_A(0xFF); // Set Port A Conditions(Mode)
59
60     // RB6-RB0:OUT, RB7:OUT LED Control
61     // TRISB 0000 0000, 0x00
62
63     set_tris_B(0x00); // Set Port B Conditions
64
65     //set_tris_C(0x00);
66
67     setup_oscillator(OSC_8MHZ);
68
69     return;
70 }
71 //
72 // Main Programs Begin from Here.
73 //
74 void main()
75 {
76     INT_IO(); // Call INT_IO() subroutines for INITIALIZE PORTs.

```

```

77  int pre0 = 0 , pre1 = 0 , pre2 = 0 ,pre4 = 0, pre5 = 0,pre6 = 0;
78  int trig = 0; //0:Time setting 1:count time
79  int count = 0 ,minute = 0;
80  //time setting
81  while(1)
82  {
83      OUTPUT_B(count);
84      //minute lighter
85      if(minute == 1){BIT_SET(*PORTB,6);}
86      if(minute == 2){BIT_SET(*PORTB,7);}
87      if(minute == 3)
88      {
89          BIT_SET(*PORTB,6);
90          BIT_SET(*PORTB,7);
91      }
92      while(trig == 1)
93      {
94          //counting
95          OUTPUT_B(count);
96          //minute lighter
97          if(minute == 1){BIT_SET(*PORTB,6);}
98          if(minute == 2){BIT_SET(*PORTB,7);}
99          if(minute == 3)
100         {
101             BIT_SET(*PORTB,6);
102             BIT_SET(*PORTB,7);
103         }
104         //stop button
105         if(BIT_TEST(*PORTA,3) == 0){trig = 0;}
106         //minute -1 or end of timer
107         if(count == -1)//count -1 to avoid timer end first
108         {
109             //end of timer
110             minute = minute - 1;
111             if(minute == -1)
112             {
113                 //minute = -1 to avoid end without second = 0;
114                 OUTPUT_B(255);
115                 delay_ms(1000);
116                 OUTPUT_B(0);
117                 delay_ms(1000);
118                 count = 0;
119             }
120             //1 minute -> 59 second
121             else{count = 59;}
122         }
123         //Prepare of next
124         delay_ms(1000);
125         count = count - 1;
126         //To The Next
127     }
128     //start button
129     if (BIT_TEST(*PORTA,0)== 0)
130     {
131         if (pre0 == 0)
132         {
133             if (trig == 0){trig = 1;}
134             else {trig = 0;}
135         }
136     }
137     else{pre0 = 0;}
138     //time setting
139     //count+1 button(second + 1)
140     if(BIT_TEST(*PORTA,1)== 0)
141     {
142         if(pre1== 0){count = count + 1;}
143         pre1 = 1;
144     }
145     else{pre1 = 0;}
146     //count-1 button (second + 1)
147     if(BIT_TEST(*PORTA,2)==0)
148     {
149         if(pre2 == 0)

```

```

150     {
151         trig = 0;
152         count = count - 1;
153     }
154     pre2 = 1;
155 }
156 else{pre2 = 0;}
157 //minute +1 button
158 if(BIT_TEST(*PORTA,4)==0)
159 {
160     if(pre4 == 0)
161     {
162         trig = 0;
163         minute = minute + 1;
164     }
165     pre4 = 1;
166 }
167 else{pre4 = 0;}
168 //minute -1 button
169 if(BIT_TEST(*PORTA,5)==0)
170 {
171     if(pre5== 0)
172     {
173         trig = 0;
174         minute = minute - 1;
175     }
176     pre5 = 1;
177 }
178 else{pre5 = 0;}
179 //reset button
180 if(BIT_TEST(*PORTA,6)==0)
181 {
182     if(pre6 == 0)
183     {
184         trig = 0;
185         minute = 0;
186         count = 0;
187     }
188     pre6 = 1;
189 }
190 else{pre6 = 0;}
191 //minute + 1
192 if(count == 60)
193 {
194     minute = minute + 1;
195     count = 0;
196 }
197 //regulate minutes
198 if(minute == 4){minute = 3;}
199 }
200 }

```