目次

1 実験機材

マイコンボードの通し番号: 72
使用したパソコン: 実験室の PC(Lenovo Thickpad)
OS: Windows
カラーセンサの机からの高さ: 1cm 程

2 理論

3 実験内容

3.1 【課題 6】colorsensor の改造

color_sensor_main.c にて，割り込みハンドラを CSPinIntHandler に設定して割り込みの初期化を行い，main() 関数内でカラーセンサの割り込みを許可した．そして，下敷きの黒色を読み込ませたときに割り込みが発生せず，その他の色を読み込ませたときだけ割り込みが発生するようにした．このとき，事前に色の読み込みデータの値をメモしておき，それらを用いて閾値の設定を行った．また，得られたカラーセンサの値を利用して，物体の色情報を LCD に表示させた．

3.2 【課題 7】ラーメンタイマーの作成

rotary_main.c にて，割り込みハンドラを SW1PinIntHandler と REPinIntHandler に設定して割り込みの初期化を行った．そして，main() 関数内で SW1 とロータリーエンコーダの割り込みを許可した．REPinIntHandler で，左へ回すと数値が減少し，右へ回すと数値が増加し，下限と上限がそれぞれ 0，600 となるようにコードを書いてその結果を LCD へ表示するようにした．また，SW1PinIntHadler で割り込みが発生したときに Systick 割り込みを許可して数値を 1 秒ごとに減らしていき，0 になったらブザーを鳴らすようにした．そして，その状態でロータリーエンコーダを回すとブザーが停止するようにした．

3.3 【課題 8】色判定器の作成

color_sensor_main.c にて，ロータリーエンコーダの回転に応じて，LCD の 1 行目の左側に「Red」「Yellow」「Blue」「Green」「None」と表示させ，2 行目左側にはカラーセンサが検知している色を同じように表示させ，1 行目右側と 2 行目右側に各色をカラーセンサが検知した回数を 99 回まで表示させるように割り込みハンドラを改造した．また，1 行目左側で選択した色と同じ色を検知したら，ブザーから「ラ (O4A)」の音を出し，その他の 3 色のいずれかであればブザーから「ミ (O4E)」の音を出させるようにコードを書いた．音は，一定時間が経過すると鳴り止むように

した．SW1 を押下することで，色検知カウント回数を 0 に戻すようにした．

4 実験結果

4.1 【課題 6】colorsensor の改造

以下に，CSPinIntHandler のソースコードを示す．

```
1  void CSPinIntHandler(void) {
2      disableCSPinInt();
3      clearCSPinInt();
4      clearIntColorSensor();
5
6      setAddressLCD(0, 0);
7      writeTextLCD("R:     B:       ", 16);
8      setAddressLCD(0, 1);
9      writeTextLCD("G:     C:       ", 16);
10
11     uint16_t _red = read16ColorSensor(RDATAL_REG);
12     uint16_t _blue = read16ColorSensor(BDATAL_REG);
13     uint16_t _green = read16ColorSensor(GDATAL_REG);
14     uint16_t _clear = read16ColorSensor(CDATAL_REG);
15
16     uint16_t colors[] = {_red, _blue, _green, _clear};
17     int i;
18     for(i = 0; i < 4; i++){
19         switch(i){
20             case 0:
21                 setAddressLCD(2, 0);
22                 break;
23             case 1:
24                 setAddressLCD(9, 0);
25                 break;
26             case 2:
27                 setAddressLCD(2, 1);
28                 break;
29             case 3:
30                 setAddressLCD(9, 1);
31                 break;
32         }
33         writeTextLCD(itoh(colors[i], 4), 4);
34     }
35
36     delay_ms(100);
37
38     enableCSPinInt();
39 }
```

ソースコード 1: 【課題 6】colorsensor における CSPinIntHandler 関数

この関数では，2 行目から 4 行目にかけて新たなカラーセンサの割り込みを禁止し，カラーセンサ

側とマイコンボードのカラーセンサ用のピンにおける割り込み処理をクリアした．6 行目から 9 行目で一旦 RGBC の値を空白として出力し，11 行目から 14 行目で受け取った RGBC の値を 16 行目から 34 行目にかけて LCD に出力させた．36 行目で 100ms だけ処理を遅らせることで連続割り込みを防いだ．そして，38 行目でカラーセンサの割り込みを許可した．

4.2 【課題 7】ラーメンタイマーの作成

以下に SystickIntHandler，SW1PinIntHandler，REPinIntHandler のソースコードをそれぞれ示す．

```
1  void SysTickIntHandler(void) {
2      if(cnt == 0){
3          toneBuzzer(O4C);
4          SysTickIntDisable();
5          enableREPinInt();
6      }
7
8      static uint8_t spaces[] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
           ' ', ' ', ' ', ' '};
9
10     setAddressLCD(0, 0);
11     writeTextLCD(spaces,16);
12     setAddressLCD(0, 1);
13     writeTextLCD(spaces,16);
14
15     int len = 1, temp = cnt;
16     while(temp / 10 > 0){
17         len++;
18         temp /= 10;
19     }
20     setAddressLCD(16-len, 0);
21     writeTextLCD(itoa(cnt, len), len);
22     delay_ms(1000);
23
24     cnt--;
25 }
```

ソースコード 2: 【課題 7】における SystickIntHandler 関数

```
1  void SW1PinIntHandler(void) {
2      disableSW1PinInt();
3      clearSW1PinInt();
4      disableREPinInt();
5
6      SysTickIntEnable();
7
8      enableSW1PinInt();
9  }
```

ソースコード 3: 【課題 7】における SW1PinIntHandler 関数

```
1  void REPinIntHandler(void) {
2      disableREPinInt();
3      clearREPinInt();
4
5      restBuzzer();
6
7      static uint8_t spaces[] = {' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
           ' ', ' ', ' ', ' '};
8
9      setAddressLCD(0, 0);
10     writeTextLCD(spaces,16);
11     setAddressLCD(0, 1);
12     writeTextLCD(spaces,16);
13
14     int str_len = 1;
15     int32_t d = getDirectionRotaryEncoder();
16     cnt += d;
17     if(cnt > 600)
18         cnt = 600;
19     else if(cnt < 0)
20         cnt = 0;
21
22     int temp = cnt;
23     while(temp / 10 > 0){
24         str_len++;
25         temp /= 10;
26     }
27
28     setAddressLCD(16 - str_len, 0);
29     writeTextLCD(itoa(cnt, str_len), str_len);
30
31     enableREPinInt();
32 }
```

ソースコード 4: 【課題 7】における REPinIntHandler 関数

ソースコード 2 は，SW1 が押下されてからのカウントタイマーを実装したものである．具体的には，2 行目から 6 行目でカウントが 0 になったときにブザー音を鳴らし，Systick 割り込みを禁止してロータリーエンコーダの割り込みを許可した．また，8 行目から 31 行目にかけて現在のカウントを LCD に表示させ，32 行目は 1 秒ごとのタイマーとして扱うために実装した．これらの処理の後，34 行目でカウントダウンさせた．

ソースコード 3 は，SW1 が押下されたときの割り込みの呼び出しや停止を実装したものである．具体的には，

4.3 【課題 8】色判定器の作成

5　考察

【考察 1】
【考察 2】
【考察 3】
【考察 4】

6　問題回答

提出期限超過のため割愛