階層化構造

1. 階層化構造

これまで設計してきた回路は 1 つの verilog ファイルに記述していきました。これまで設計してきたような小規模回路なら 1 つの verilog ファイルに記述しても問題ありません。しかし、大規模回路を設計するような場合 1 つの verilog ファイルに記述していくと非常に見づらいものになってしまいます。そこで、回路を機能ごとにモジュール化して複数の verilog ファイルに分けることで回路の可読性もよくなり、また新しく回路を設計するときに以前設計した回路を再利用しやすくなります。このように、回路を機能ごとにモジュール化し、複数のモジュールを利用して 1 つの回路を設計する方法を階層化と呼びます。

課題 13

STEP12-2 と全く同じ7セグカウンタ回路を階層化します。

2. プロジェクトの作成

STEP 13 用のプロジェクトを作成してください。

3. プログラム（回路）の記述

では、実際に課題を実現するプログラムを記述していきます。階層化例として、step13-a.v（Counter_7seg）、step13-b.v（UpDown_Counter）、step13-c.v（Decoder）の 3 つの verilog ファイルに分割して、プロジェクトに追加します。

step13-a.v　弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module Counter_7seg(CLK, RESET, TSW, SEG);
2
3    input CLK, RESET;
4    input [1:0] TSW;
5    output [7:0] SEG;
6
7    wire [3:0] Q;
8
9    UpDown_Counter updown_counter1(.CLK(CLK), .RESET(RESET), .TSW(TSW), .Q(Q));
10   Decoder decoder1(.Q(Q), .SEG(SEG));
11
12 endmodule
```

❶ インスタンス構文

階層化構造

❶ インスタンス構文

```
モジュール名 インスタンス名 ( ポート接続 );

例） Decoder decoder1(.Q(Q), .SEG(SEG));
```

モジュールをインスタンス化する際の構文です。まずインスタンス化したいモジュール名を宣言したあとに、インスタンス化した際の名前を指定します。ポート接続は下位モジュールで宣言しているポート名をピリオド記号の後に指定して、括弧内に上位のネット名を指定します。

step13-b.v　step11-3.v と全く同じ

```
1  module UpDown_Counter(CLK, RESET, TSW, Q);
2
3    input CLK, RESET;
4    input [1:0] TSW;
5    output [3:0] Q;
6
7    reg [3:0] Q = 0;
8    reg [18:0] temp_count = 0;
9    reg [1:0] out, buffer;
10
11   always @(posedge CLK or negedge RESET)
12   begin
13     if (!RESET) temp_count <= 0;
14     else temp_count <= temp_count + 1;
15   end
16
17   always @(posedge CLK)
18     if(temp_count==0) out <= TSW;
19
20   always @(posedge CLK or negedge RESET)
21   begin
22     if (!RESET) buffer <= 0;
23     else buffer <= out;
24   end
25
26   assign inc = out[0] & ~buffer[0];
27   assign dec = out[1] & ~buffer[1];
28
29   always @(posedge CLK or negedge RESET)
30   begin
31     if (!RESET) Q <= 0;
32     else if (inc == 1)
33     begin
```

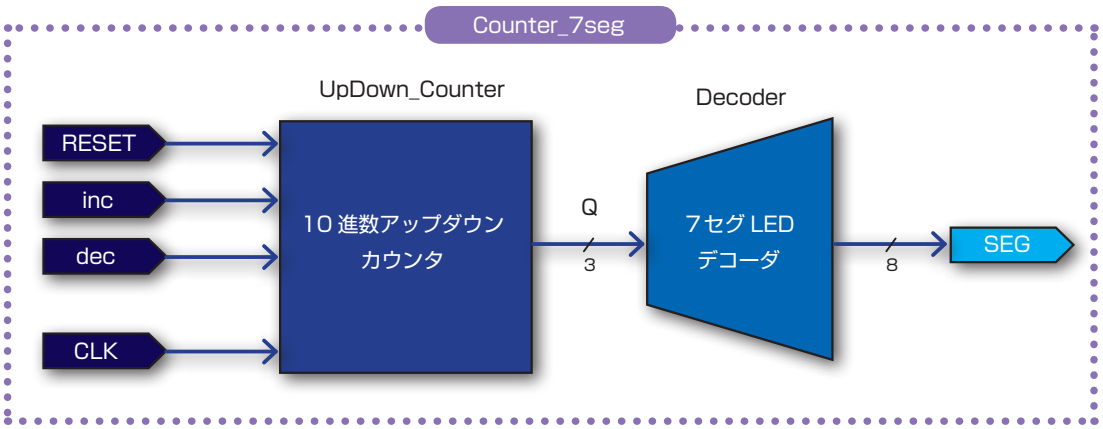
```
34        if (Q == 9) Q = 0;
35        else Q <= Q + 1;
36      end
37      else if (dec == 1)
38      begin
39        if (Q == 0) Q <= 9;
40        else Q <= Q - 1;
41      end
42    end
43
44  endmodule
```

step13-c.v　step12-1.v とほぼ同じ

```
1   module Decoder(Q, SEG);
2
3     input [3:0] Q;
4     output [7:0] SEG;
5
6     reg [7:0] SEG;
7
8     always @(Q)
9     begin
10      case(Q)
11        4'b0000:SEG <= 8'b0001_0000; //0
12        4'b0001:SEG <= 8'b0001_0001; //1
13        4'b0010:SEG <= 8'b0001_0010; //2
14        4'b0011:SEG <= 8'b0001_0011; //3
15        4'b0100:SEG <= 8'b0001_0100; //4
16        4'b0101:SEG <= 8'b0001_0101; //5
17        4'b0110:SEG <= 8'b0001_0110; //6
18        4'b0111:SEG <= 8'b0001_0111; //7
19        4'b1000:SEG <= 8'b0001_1000; //8
20        4'b1001:SEG <= 8'b0001_1001; //9
21        default:SEG <= 8'b0001_1111;
22      endcase
23    end
24
25  endmodule
```

階層化構造

課題 12-2 で設計した回路を階層化に習い、モジュール化したものが前ページのサンプルとなります。
階層化した回路は下図のようなイメージとなります。



10 進数アップダウンカウンタと 7 セグ LED デコーダーを組合せて Counter_7seg 回路を構成しています。ちなみに、課題 12-2 の階層化しない場合のイメージが下図となります。

階層化を有効に使い機能モジュールを用意しておくと、機能モジュールを再利用し組合せながら設計を進めていくことができます。

4. コンパイル（論理合成）

文法チェック

Analysis & Synthesis を行い文法チェックを行ってください。

ピン配置（配置結線）

ピン配置を行ってください。ピン配置例は課題 13-2 と同じです。

コンパイル（コンフィギュレーションファイルの生成）

ピン配置が終わったら、コンパイルを行ってください。

5. コンフィギュレーションファイルの転送

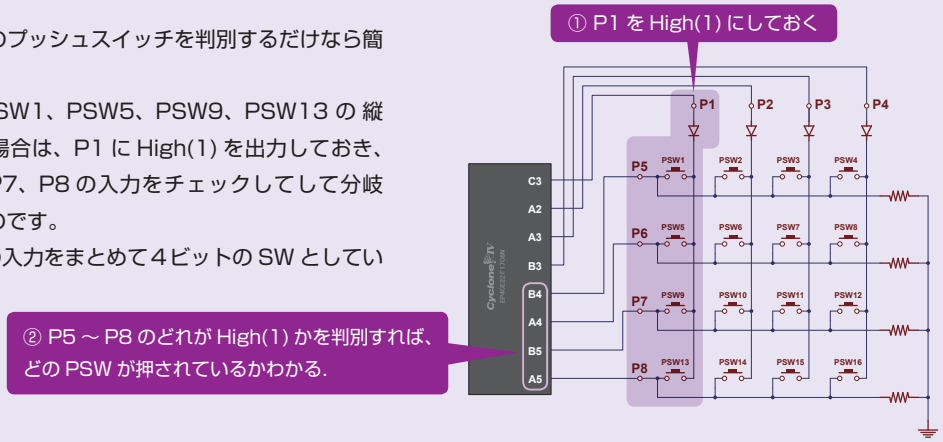
FPGA に .sof を転送して動作を確認してみましょう。
トグルスイッチ 1 を何回か操作してみましょう。
STEP12 と同様に TSW1 を操作するとカウントアップして、TSW2 を操作するとカウントダウンします。結果は 7 セグ LED に数字が表示されます。

プッシュスイッチマトリクスのキースキャン

プッシュスイッチマトリクスの使い方については巻末の「エレモ取扱説明書」に掲載しています。
ここでは、Velilog で「キースキャン」をどのように行うのか具体的に使用例を紹介します。

縦１列だけ（４個の PSW）を使う

縦１列４個のプッシュスイッチを判別するだけなら簡単です。
例えば、PSW1、PSW5、PSW9、PSW13 の縦１列を使う場合は、P1 に High(1) を出力しておき、P5、P6、P7、P8 の入力をチェックしてして分岐すればいいのです。
P5 ～ P8 の入力をまとめて４ビットの SW としています。



sw_matrix4.v

弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module SW_Matrix(CLK, RESET, P, SW, LED);
2
3    input CLK, RESET;
4    input [3:0] SW;
5
6    output [4:0] LED;
7    output P;
8
9    reg [3:0] LED;
10
11   assign P = 1;
12
13   always @(posedge CLK or negedge RESET)
14   begin
15     if (!RESET) LED <= 0;
16     else
17       begin
18       case(SW)
19         4'b1000:LED <= 1;       // PSW1
20         4'b0100:LED <= 5;       // PSW5
21         4'b0010:LED <= 9;       // PSW9
22         4'b0001:LED <= 13;      // PSW13
23       endcase
24       end
25   end
26
27 endmodule
```

ノード名	ピン番号	パーツ名
CLK	PIN_R8	50MHz オシレータ
LED[3]	PIN_A11	LED[3]
LED[2]	PIN_B13	LED[2]
LED[1]	PIN_A13	LED[1]
LED[0]	PIN_A15	LED[0]
P	PIN_C3	P1（エレモ 102）
RESET	PIN_J15	プッシュキー [0]
SW[3]	PIN_B4	P5（エレモ 102）
SW[2]	PIN_A4	P6（エレモ 102）
SW[1]	PIN_B5	P7（エレモ 102）
SW[0]	PIN_A5	P8（エレモ 102）

ピン配置（配置結線）例

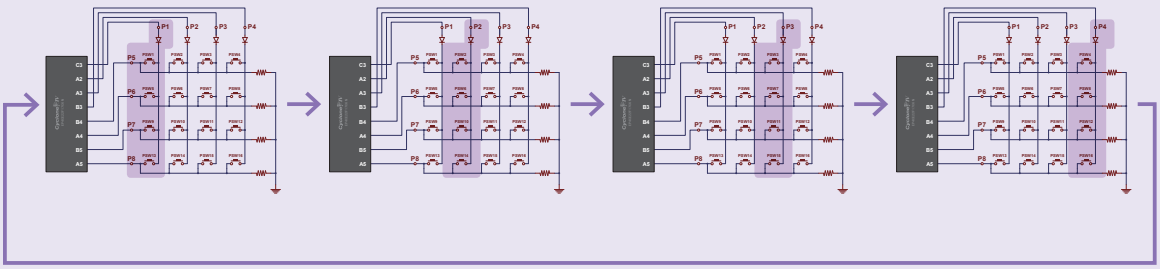
プッシュスイッチマトリクスのキースキャン

4列すべて（16 個の PSW）を使う

では次に 16 個のプッシュスイッチを判別する方法を紹介します。
縦列4個のプッシュスイッチが横4組に増えたので4パターンに分けて考えます。
4パターンは、2ビットの count を 00、01、10、11 で繰り返しカウントし作成します。

count が 00 のとき P1 だけを High(1) にします。
count が 01 のとき P2 だけを High(1) にします。
count が 10 のとき P3 だけを High(1) にします。
count が 11 のとき P4 だけを High(1) にします。

P1 ～ P4 の出力をまとめて4ビットの COLMN とします。
COLMN が 1000、0100、0010、0001 のとき、それぞれ P5、P6、P7、P8 の入力をチェックして分岐します。
例えば COLMN が 0100（P2 が High）のとき SW が 0010（P7 が High）なら PSW10 が押されていると判断できます。



構造は以上なのですが、これだけでは正しく動作しません。
1クロックで count のカウントアップ、COLMN への出力、SW の判別を行うことはできません。
分周して count をカウントアップさせると正しく動作するようになります。

プッシュスイッチマトリクスのキースキャン

sw_matrix16.v

弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module SW_Matrix(CLK, RESET, COLMN, SW, LED);
2
3    input CLK, RESET;
4    input [3:0] SW;
5
6    output [3:0] COLMN;
7    output [4:0] LED;
8
9    reg [3:0] COLMN;
10   reg [4:0] LED;
11   reg [1:0] count = 0;
12
13   reg [1:0]temp_count = 0;
14
15   always @(posedge CLK)
16     if (temp_count == 2) temp_count <= 0;
17     else temp_count <= temp_count + 1;
18
19   assign Enable = (temp_count == 0) ? 1 : 0;
20
21   always @(posedge CLK)
22     if (Enable == 1) count <= count + 1;
23
24   always @(posedge CLK)
25   begin
26     if (Enable == 1)
27     begin
28       case(count)
29         2'b00:COLMN <= 4'b1000;  // P1 High
30         2'b01:COLMN <= 4'b0100;  // P2 High
31         2'b10:COLMN <= 4'b0010;  // P3 High
32         2'b11:COLMN <= 4'b0001;  // P4 High
33       endcase
34     end
35   end
36
37   always @(posedge CLK or negedge RESET)
38   begin
39     if (!RESET) LED <= 0;
40     else if (Enable == 1)
41       if(COLMN == 4'b1000)
42       begin
43         case(SW)
44           4'b1000:LED <= 1;    // PSW1
45           4'b0100:LED <= 5;    // PSW5
46           4'b0010:LED <= 9;    // PSW9
47           4'b0001:LED <= 13;   // PSW13
48         endcase
49       end
```

分周回路の追加

プッシュスイッチマトリクスのキースキャン

```
50        else if(COLMN == 4'b0100)
51        begin
52          case(SW)
53            4'b1000:LED <= 2;    // PSW2
54            4'b0100:LED <= 6;    // PSW6
55            4'b0010:LED <= 10;   // PSW10
56            4'b0001:LED <= 14;   // PSW14
57          endcase
58        end
59        else if(COLMN == 4'b0010)
60        begin
61          case(SW)
62            4'b1000:LED <= 3;    // PSW3
63            4'b0100:LED <= 7;    // PSW7
64            4'b0010:LED <= 11;   // PSW11
65            4'b0001:LED <= 15;   // PSW15
66          endcase
67        end
68        else if(COLMN == 4'b0001)
69        begin
70          case(SW)
71            4'b1000:LED <= 4;    // PSW4
72            4'b0100:LED <= 8;    // PSW8
73            4'b0010:LED <= 12;   // PSW12
74            4'b0001:LED <= 16;   // PSW16
75          endcase
76        end
77    end
78
79  endmodule
```

ノード名	ピン番号	パーツ名
CLK	PIN_R8	50MHz オシレータ
COLMN[3]	PIN_C3	P1（エレモ 102）
COLMN[2]	PIN_A2	P2（エレモ 102）
COLMN[1]	PIN_A3	P3（エレモ 102）
COLMN[0]	PIN_B3	P4（エレモ 102）
LED[4]	PIN_D1	LED[4]
LED[3]	PIN_A11	LED[3]
LED[2]	PIN_B13	LED[2]
LED[1]	PIN_A13	LED[1]
LED[0]	PIN_A15	LED[0]
RESET	PIN_J15	プッシュキー [0]
SW[3]	PIN_B4	P5（エレモ 102）
SW[2]	PIN_A4	P6（エレモ 102）
SW[1]	PIN_B5	P7（エレモ 102）
SW[0]	PIN_A5	P8（エレモ 102）

ピン配置（配置結線）例

7セグ LED のダイナミック点灯

7セグメント LED の使い方については巻末の「エレモ取扱説明書」に掲載しています。
ここでは、Velilog で「ダイナミック点灯」をどのように行うのか具体的に使用例を紹介します。

最上位階層。
SW_Matrix 回路で取得したスイッチの状態から、スイッチの ON/OFF を検知してカウンタの値を増加させる。

dynamic_drive.v　弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module Dynamic_Drive(CLK, RESET, SW, COLUMN, SEG, DIGIT);
2
3      input CLK, RESET;
4      input [3:0] SW;
5
6      output [3:0] COLUMN, SEG, DIGIT;
7
8      wire [15:0] sw_number;
9
10     reg [15:0] counter = 0;
11     reg [15:0] buffer, out;
12
13
14     SW_Matrix sw_matrix1(.CLK(CLK), .RESET(RESET), .SW(SW), .COLUMN(COLUMN),
       .SW_STATE(sw_number));                          ……SW_Matrix のインスタンス
15     Decoder decoder1(.CLK(CLK), .RESET(RESET), .Q(counter), .SEG(SEG),
       .DIGIT(DIGIT));                                 ……Decoder のインスタンス
16
17     always @(posedge CLK or negedge RESET)
18     begin
19         if (!RESET) buffer <= 0;
20         else buffer <= sw_number;
21     end
22
23     assign inc1000 = sw_number[0] & ~buffer[0];
24     assign inc100 = sw_number[1] & ~buffer[1];
25     assign inc10 = sw_number[2] & ~buffer[2];
26     assign inc1 = sw_number[3] & ~buffer[3];
27
28     always @(posedge CLK)
29     begin
30         if (inc1 == 1)
31             counter <= counter + 1;
32         else if (inc10 == 1)
33             counter <= counter + 10;
34         else if (inc100 == 1)
35             counter <= counter + 100;
36         else if (inc1000)
37             counter <= counter + 1000;
38     end
39
40 endmodule
```

各スイッチのエッジ検出

押されたスイッチによって、
カウンタの値を +1、+10、
+100、+1000 する。

7セグ LED のダイナミック点灯

入力された数値を 4 桁の 7 セグ LED に 10 進数で表示する。表示方式はダイナミック点灯。

decoder.v

弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module Decoder(CLK, RESET, Q, SEG, DIGIT);
2
3      input CLK, RESET;
4      input [15:0] Q;
5      output [3:0] SEG, DIGIT;
6
7      reg [3:0] SEG, DIGIT, count;
8      reg [14:0] temp_count = 0;
9
10     always @(posedge CLK or negedge RESET)
11     begin
12         if (!RESET) temp_count <= 0;
13         else temp_count <= temp_count + 1;
14     end
15
16     assign enable = (temp_count == 0) ? 1 : 0;
17
18     always @(posedge CLK or negedge RESET)
19     begin
20         if (!RESET) count <= 0;
21         else if (enable == 1) count <= count + 1;
22     end
23
24     always @(posedge CLK)
25     begin
26         if (!RESET)
27         begin
28             SEG <= 0;
29             DIGIT <= 0;
30         end
31         else if(enable == 1)
32             case (count)
33                 0:
34                 begin
35                     SEG <= (Q % 10000) / 1000;
36                     DIGIT <= 4'b0001;
37                 end
38                 2:
39                 begin
40                     SEG <= (Q % 1000) / 100;
41                     DIGIT <= 4'b0010;
42                 end
43                 4:
44                 begin
45                     SEG <= (Q % 100) / 10;
46                     DIGIT <= 4'b0100;
47                 end
48                 6:
49                 begin
50                     SEG <= Q % 10;
51                     DIGIT <= 4'b1000;
52                 end
53                 default:
54                 begin
55                     SEG <= 0;
56                     DIGIT <= 0;
57                 end
58             endcase
59     end
60
61 endmodule
```

ダイナミック点灯周期決定用カウンタ

ダイナミック点灯パターン用カウンタのカウントアップ

パターンカウンタの値でダイナミック点灯を制御する。入力された数値の各桁の値を 10 進数に変換して対応した 7 セグ LED に出力している。

7セグ LED のダイナミック点灯

16 個マトリックススイッチの各スイッチの状態をスキャンする

sw_matrix.v

弊社サイトに解答例ソースをご用意しています。http://www.adwin.com/product/AKE-1104.html

```
1  module SW_Matrix(CLK, RESET, SW, COLUMN, SW_STATE);
2
3      input CLK, RESET;
4      input [3:0] SW;
5
6      output [3:0] COLUMN;
7      output [15:0]SW_STATE;
8
9      reg [3:0] COLUMN;
10     reg [1:0] count;
11     reg [18:0]temp_count = 0;
12     reg [15:0] SW_STATE;
13
14     always @(posedge CLK or negedge RESET)
15     begin
16         if (!RESET) temp_count <=0;
17         else temp_count <= temp_count + 1;
18     end
19
20     assign enable = (temp_count == 0) ? 1 : 0;
21
22     always @(posedge CLK or negedge RESET)
23     begin
24         if (!RESET) count <= 0;
25         else if (enable == 1) count <= count + 1;
26     end
27
28     always @(posedge CLK or negedge RESET)
29     begin
30         if (!RESET) COLUMN <= 0;
31         else if (enable == 1)
32         begin
33             case(count)
34                 2'b00:COLUMN <= 4'b1000;
35                 2'b01:COLUMN <= 4'b0100;
36                 2'b10:COLUMN <= 4'b0010;
37                 2'b11:COLUMN <= 4'b0001;
38             endcase
39         end
40     end
41
42     always @(posedge CLK or negedge RESET)
43     begin
44         if (!RESET) SW_STATE <= 0;
45         else if (enable == 1)
46             if(COLUMN == 4'b1000)
47             begin
48                 SW_STATE[0] <= SW[3];   // PSW1
49                 SW_STATE[4] <= SW[2];   // PSW5
50                 SW_STATE[8] <= SW[1];   // PSW9
51                 SW_STATE[12] <= SW[0];  // PSW13
```

マトリックススイッチのスキャンタイミング決定用カウンタ

スキャンカウンタのカウント

COLUMN の出力パターンのセット

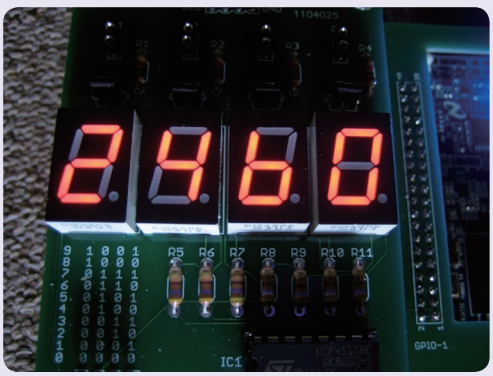
6 個のスイッチの状態を 16 ビットの情報として記憶する。

7セグ LED のダイナミック点灯

```
52              end
53              else if(COLUMN == 4'b0100)
54              begin
55                  SW_STATE[1] <= SW[3];   // PSW2
56                  SW_STATE[5] <= SW[2];   // PSW6
57                  SW_STATE[9] <= SW[1];   // PSW10
58                  SW_STATE[13] <= SW[0];  // PSW14
59              end
60              else if(COLUMN == 4'b0010)
61              begin
62                  SW_STATE[2] <= SW[3];   // PSW3
63                  SW_STATE[6] <= SW[2];   // PSW7
64                  SW_STATE[10] <= SW[1];  // PSW11
65                  SW_STATE[14] <= SW[0];  // PSW15
66              end
67              else if(COLUMN == 4'b0001)
68              begin
69                  SW_STATE[3] <= SW[3];   // PSW4
70                  SW_STATE[7] <= SW[2];   // PSW8
71                  SW_STATE[11] <= SW[1];  // PSW12
72                  SW_STATE[15] <= SW[0];  // PSW16
73              end
74      end
75
76  endmodule
```

6個のスイッチの状態を16ビットの情報として記憶する。

PSW1 ～ PSW4 を押すと、対応した桁の 7 セグ LED がカウントアップします。
カウント値は桁上がりするようになっています。



ノード名	ピン番号	パーツ名
CLK	PIN_R8	50MHz オシレータ
COLMN[3]	PIN_C3	P1（エレモ 102）
COLMN[2]	PIN_A2	P2（エレモ 102）
COLMN[1]	PIN_A3	P3（エレモ 102）
COLMN[0]	PIN_B3	P4（エレモ 102）
DIGIT[3]	PIN_R13	P4（エレモ 203）
DIGIT[2]	PIN_T13	P3（エレモ 203）
DIGIT[1]	PIN_T14	P2（エレモ 203）
DIGIT[0]	PIN_T15	P1（エレモ 203）
RESET	PIN_J15	プッシュキー [0]
SEG[3]	PIN_T12	P5（エレモ 203）
SEG[2]	PIN_R12	P6（エレモ 203）
SEG[1]	PIN_T11	P7（エレモ 203）
SEG[0]	PIN_T10	P8（エレモ 203）
SW[3]	PIN_B4	P5（エレモ 102）
SW[2]	PIN_A4	P6（エレモ 102）
SW[1]	PIN_B5	P7（エレモ 102）
SW[0]	PIN_A5	P8（エレモ 102）

ピン配置（配置結線）例