

임베디드 시스템 프로젝트

2012253066

김기현

<소스코드(메인)>

```
module watch_project (clk, reset, set_outside, set_inside, display_sec, display_sec10, display_min, display_min10, display_hour, display_hour10, state );
    input clk, reset, set_outside, set_inside;
    output [6:0] display_sec, display_sec10, display_min, display_min10, display_hour, display_hour10;
    output [5:0] state;
    wire en_sec, en_sec10, en_min, en_min10, en_hour, en_hour10, stop_enable_sec, stopwatch_tc, stopwatch, add_hour, add_min;

    count_display_sec #(60,6) u1 (clk, reset, set_outside, set_inside, en_sec, display_sec, display_sec10, tc_sec, stop_enable_sec, stopwatch_tc, stopwatch);
    count_display_min #(60,6) u2 (clk, reset, en_min, display_min, display_min10, tc_min, stop_enable_sec, stopwatch_tc, stopwatch, add_min);
    count_display_hour #(24,5) u3 (clk, reset, en_hour, display_hour, display_hour10, stop_enable_sec, stopwatch, add_hour);

    controller u6 (clk, reset, set_outside, set_inside, tc_sec, tc_min, en_sec, en_min,
    en_hour, state, stop_enable_sec, stop_enable_min, stopwatch_tc, stopwatch, add_hour, add_min);

endmodule
```

⇒ Controller와 clock 동작하는 모듈들을 통괄하는 메인 모듈이다.

<소스코드(컨트롤러)>

```
module controller (clk, reset, b0, b1, tc_sec, tc_min, en_sec, en_min, en_hour,
state, stop_enable_sec, stop_enable_min, stopwatch_tc, stopwatch, add_hour, add_min);

    input clk, reset, b0, b1, tc_sec, tc_min, stopwatch_tc;
    output en_sec, en_min, en_hour, stop_enable_sec, stop_enable_min, stopwatch, add_hour, add_min;
    output [5:0] state;
    reg [5:0] state;
    reg inc_min = 1, inc_hour = 1;
    integer a;
    parameter s0 = 1, s10 = 4, s11 = 6, s12 = 5, s20 = 8, s21 = 10, s22 = 9, s30 = 16, s31 = 18, s32 = 17;

    always @(posedge clk or posedge reset) begin
        if (reset) state <= s0;
        else begin
            if (a == 20000000) begin
                a = 0;
                case (state)
                    s0: if (!b0) state <= s10; //s0 : watch
                        else if (!b1) state <= s30;

                    s10: if (!b0) state <= s0; //s10 : set clock
                        else if (!b1) state <= s11;
                    s11: if (!b1) state <= s12; //s11 : set hour
                    s12: if (!b1) state <= s10; //s12 : set min

                    s30: if (!b0) state <= s31; //s30 : stop watch
                        else if (!b1) state <= s0;
                    s31: if (!b0) state <= s32; //s31 : start stop watch
                    s32: if (!b0) state <= s30; //s32 : stop stop watch
                        else if (!b1) state <= s0;
                    default : state <= s0;
                endcase
            end
        end
    end
```

```

32         endcase
33     end
34     else
35         a = a + 1;
36     end
37 end
38
39 assign en_sec = ((state == 4) | (state == 6) | (state == 5)) ? 0 : 1;
40 assign en_min = (state == 5) ? 0 : tc_sec;
41 assign en_hour = (state == 6) ? 0 : tc_min;
42 assign add_hour = ((state==6)&(!b0)) ? 1:0;
43 assign add_min = ((state==5)&(!b0)) ? 1:0;
44 assign alarm = ((state == 8) | (state == 10) | (state == 9)) ? 0 : 1;
45 assign stopwatch = ((state==16)|(state==17)|(state==18)) ? 1 : 0;
46 assign stop_enable_sec = ((state==18)& (state!=17)) ? 1 : 0;
47 assign stop_enable_min = stopwatch_tc;
48 endmodule

```

⇒ State를 모듈들의 순서를 제어해주는 컨트롤러이다. 기본 형태는 교수님의 자료에서 따왔다. 버튼은 key0, key1 두 개를 사용했으며 clock(일반 시계 기능)상태에서 시작을 하고 거기서 key0를 누르면 시계를 세트(수정)할 수 있는 모드로 들어가며 key1 누를 시, 시 수정 그 다음 key1을 누르면 분 수정으로 옮겨간다. 또한 시 수정, 분 수정에서 key0를 누르면 수가 증가한다. 세트 모드에서 key0를 누르면 수정한 시간이 세트되고, 다시 시계모드로 돌아간다. 여기서 key1을 누를 시 스탑워치 화면이 나타나고 key0를 누르면 시작, 또 누르면 정지이고 key1을 누르면 다시 시계모드로 돌아간다. 안타깝게도 알람기능은 수행하지 못했다. En_sec, en_min, en_hour는 enable신호이며 carry에 따라 시계가 올바르게 돌아갈 수 있게 해주고, add_hour, add_min은 세트 모드에서 시간 증가를 할 때 사용된다. Stopwatch는 스탑워치 모드인지 stop_enable_sec은 시작이 됐는지, stop_enable_min은 sec의 캐리를 받아 스탑워치의 수행 제어를 해준다.

<소스코드(시계의 초와 스탑워치 초)>

```

module count_display_sec(clk, reset, set_outside, set_inside, enable, display, display10, tc, stop_enable_sec, stopwatch_tc, stopwatch)
parameter N=60, M=6;
input clk, reset, enable, set_outside, set_inside, stop_enable_sec, stopwatch;
output [6:0] display, display10;
output tc, stopwatch_tc;
reg [M-1:0] bcd, stop_bcd;
reg [6:0] display, display10;
integer a, sr, srl0, stop_a, stop_sr, stop_srl0, check, check_return;

assign tc = (bcd == N-1) & enable;
assign stopwatch_tc = (stop_bcd == N-1) & stop_enable_sec;

always @(posedge clk or posedge reset) begin
    if (reset) begin
        bcd = 0;
        sr = 0;
        srl0 = 0;
        check=0;
        display = 7'b000_0001;
        display10 = 7'b000_0001;
    end
    else if (stopwatch & (check==0) & (check_return==0)) begin
        display = 7'b000_0001;
        display10 = 7'b000_0001;
        check=1;
    end
    else if (stopwatch & (check==0) & (check_return==1)) begin

```

```

26 | end
27 | else if(stopwatch & (check==0)&(check_return==1))begin
28 |     case (stop_sr)
29 |         0:display = 7'b000_0001;
30 |         1:display = 7'b100_1111;
31 |         2:display = 7'b001_0010;
32 |         3:display = 7'b000_0110;
33 |         4:display = 7'b100_1100;
34 |         5:display= 7'b010_0100;
35 |         6:display = 7'b010_0000;
36 |         7:display = 7'b000_1111;
37 |         8:display = 7'b000_0000;
38 |         9:display = 7'b000_0100;
39 |         default: display = 7'b100_1111;
40 |     endcase
41 |     case (stop_sr10)
42 |         0:display10 = 7'b000_0001;
43 |         1:display10 = 7'b100_1111;
44 |         2:display10 = 7'b001_0010;
45 |         3:display10 = 7'b000_0110;
46 |         4:display10 = 7'b100_1100;
47 |         5:display10 = 7'b010_0100;
48 |         default: display10 = 7'b100_1111;
49 |     endcase
50 |     check=1;
51 | end
52 | else if((check==1) & !stopwatch & !enable)begin
53 |     display = 7'b000_0001;
54 |     display10 = 7'b000_0001;
55 |     check=0;
56 | end
57 | else if(enable) begin
58 |     if(a == 49999999) begin
59 |         ~-0.

```

```

58 | if(a == 49999999) begin
59 |     a=0;
60 |     if(bcd == N-1) begin bcd = 0; sr10 = 0; sr = 0; end
61 |     else begin
62 |         bcd = bcd + 1;
63 |         if(bcd <10) begin sr10 = 0; sr = bcd; end
64 |         else if(bcd <20) begin sr10 = 1; sr = bcd-10; end
65 |         else if(bcd <30) begin sr10 = 2; sr = bcd-20; end
66 |         else if(bcd <40) begin sr10 = 3; sr = bcd-30; end
67 |         else if(bcd <50) begin sr10 = 4; sr = bcd-40; end
68 |         else begin sr10 = 5; sr = bcd-50; end
69 |     end
70 |     if(!stopwatch) begin
71 |         case (sr)
72 |             0:display = 7'b000_0001;
73 |             1:display = 7'b100_1111;
74 |             2:display = 7'b001_0010;
75 |             3:display = 7'b000_0110;
76 |             4:display = 7'b100_1100;
77 |             5:display= 7'b010_0100;
78 |             6:display = 7'b010_0000;
79 |             7:display = 7'b000_1111;
80 |             8:display = 7'b000_0000;
81 |             9:display = 7'b000_0100;
82 |             default: display = 7'b100_1111;
83 |         endcase
84 |         case (sr10)
85 |             0:display10 = 7'b000_0001;
86 |             1:display10 = 7'b100_1111;
87 |             2:display10 = 7'b001_0010;
88 |             3:display10 = 7'b000_0110;
89 |             4:display10 = 7'b100_1100;
90 |             5:display10 = 7'b010_0100;
91 |             default: display10 = 7'b100_1111;

```

```

90 |             5:display10 = 7'b010_0100;
91 |             default: display10 = 7'b100_1111;
92 |         endcase
93 |     end
94 | end
95 | else begin
96 |     a= a+1;
97 | end
98 |
99 | if(stop_enable_sec) begin //stopwatch
100 |     check_return=1;
101 |     if(stop_a == 4999999) begin
102 |         stop_a=0;
103 |         if(stop_bcd == N-1) begin stop_bcd = 0; stop_sr10 = 0; stop_sr = 0; end
104 |         else begin
105 |             stop_bcd = stop_bcd + 1;
106 |             if(stop_bcd <10) begin stop_sr10 = 0; stop_sr = stop_bcd; end
107 |             else if(stop_bcd <20) begin stop_sr10 = 1; stop_sr = stop_bcd-10; end
108 |             else if(stop_bcd <30) begin stop_sr10 = 2; stop_sr = stop_bcd-20; end
109 |             else if(stop_bcd <40) begin stop_sr10 = 3; stop_sr = stop_bcd-30; end
110 |             else if(stop_bcd <50) begin stop_sr10 = 4; stop_sr = stop_bcd-40; end
111 |             else begin stop_sr10 = 5; stop_sr = stop_bcd-50; end
112 |         end
113 |     end
114 |     case (stop_sr)
115 |         0:display = 7'b000_0001;
116 |         1:display = 7'b100_1111;
117 |         2:display = 7'b001_0010;
118 |         3:display = 7'b000_0110;
119 |         4:display = 7'b100_1100;
120 |         5:display= 7'b010_0100;
121 |         6:display = 7'b010_0000;
122 |         7:display = 7'b000_1111;
123 |         8:display = 7'b000_0000;

```

```

118         3:display = 7'b000_0110;
119         4:display = 7'b100_1100;
120         5:display= 7'b010_0100;
121         6:display = 7'b010_0000;
122         7:display = 7'b000_1111;
123         8:display = 7'b000_0000;
124         9:display = 7'b000_0100;
125         default: display = 7'b100_1111;
126     endcase
127     case (stop_sr10)
128         0:display10 = 7'b000_0001;
129         1:display10 = 7'b100_1111;
130         2:display10 = 7'b001_0010;
131         3:display10 = 7'b000_0110;
132         4:display10 = 7'b100_1100;
133         5:display10 = 7'b010_0100;
134         default: display10 = 7'b100_1111;
135     endcase
136     end else stop_a =stop_a + 1;
137 end
138     end
139 end
140 endmodule

```

- ⇒ 시계의 초 부분 디스플레이와 초 시계 동작 시 초시계의 초 부분 디스플레이가 되는 모
 둘이다. 시계와 초시계의 동작은 modulo-n을 통해서 7-segment를 통해 디스플레이가 되
 고 시계는 1초, 스탑워치는 0.01초 단위로 지정하였다. Check와 check_return을 통해 조건
 문을 여러 개 써 놓았는데 이것들은 스탑워치 초반 동작 시 0으로 초기화 동작과 스탑워
 치에서 동작 시작한 후, 시계로 돌아올 때 또 시계에서 스탑워치로 전환할 때 값 처리할
 때 쓰인다.

<소스코드(분 시계와 분 스톱워치 부분)

```
module count_display_min (clk, reset, enable ,display, display10, tc, stop_enable_sec, stop_enable_min, stopwatch, add_min);
parameter N=60, M=6;
input clk, reset, enable;
input stop_enable_min, stop_enable_sec, stopwatch, add_min;
output [6:0] display, display10;
output tc;
reg [M-1:0] bcd, stop_bcd;
reg [6:0] display, display10;
integer a, mr, mrl0, stop_a, stop_mr, stop_mr10, check, check_return;

assign tc = (bcd == N-1) & enable;

always @(posedge clk or posedge reset) begin
    if (reset) begin
        bcd = 0;
        mr = 0;
        mrl0 = 0;
        display = 7'b000_0001;
        display10 = 7'b000_0001;
    end
    else if (add_min) begin
        if (a == 49999999) begin
            a = 0;
            if (bcd == N-1) begin bcd = 0; mrl0 = 0; mr = 0; end
            else begin
                bcd = bcd + 1;
                if (bcd < 10) begin mrl0 = 0; mr = bcd; end
                else if (bcd < 20) begin mrl0 = 1; mr = bcd - 10; end
                else if (bcd < 30) begin mrl0 = 2; mr = bcd - 20; end
                else if (bcd < 40) begin mrl0 = 3; mr = bcd - 30; end
                else if (bcd < 50) begin mrl0 = 4; mr = bcd - 40; end
                else begin mrl0 = 5; mr = bcd - 50; end
            end
        end
        else begin
            check = 0;
            case (mr)
                0: display = 7'b000_0001;
                1: display = 7'b100_1111;
                2: display = 7'b001_0010;
                3: display = 7'b000_0110;
                4: display = 7'b100_1100;
                5: display = 7'b010_0100;
                6: display = 7'b010_0000;
                7: display = 7'b000_1111;
                8: display = 7'b000_0000;
                9: display = 7'b000_0100;
                default: display = 7'b100_1111;
            endcase
            case (mrl0)
                0: display10 = 7'b000_0001;
                1: display10 = 7'b100_1111;
                2: display10 = 7'b001_0010;
                3: display10 = 7'b000_0110;
                4: display10 = 7'b100_1100;
                5: display10 = 7'b010_0100;
                6: display10 = 7'b010_0000;
                7: display10 = 7'b000_1111;
                8: display10 = 7'b000_0000;
                9: display10 = 7'b000_0100;
                default: display10 = 7'b100_1111;
            endcase
        end
        else begin
            a = a + 1;
        end
    end

    else if (stopwatch & (check == 0) & (check_return == 0)) begin
        display = 7'b000_0001;
        display10 = 7'b000_0001;
        check = 1;
    end

    else if (stopwatch & (check == 0) & (check_return == 1)) begin
        case (stop_mr)
            0: display = 7'b000_0001;
            1: display = 7'b100_1111;
            2: display = 7'b001_0010;
            3: display = 7'b000_0110;
            4: display = 7'b100_1100;
            5: display = 7'b010_0100;
            6: display = 7'b010_0000;
            7: display = 7'b000_1111;
            8: display = 7'b000_0000;
            9: display = 7'b000_0100;
            default: display = 7'b100_1111;
        endcase
        case (stop_mr10)
            0: display10 = 7'b000_0001;
            1: display10 = 7'b100_1111;
            2: display10 = 7'b001_0010;
            3: display10 = 7'b000_0110;
            4: display10 = 7'b100_1100;
            5: display10 = 7'b010_0100;
            default: display10 = 7'b100_1111;
        endcase
        check = 1;
    end

    else if ((check == 1) & !stopwatch & !enable) begin
        display = 7'b000_0001;
        display10 = 7'b000_0001;
        check = 0;
    end
end
```

```

end
else if(enable | stop_enable_min) begin
  if(enable) begin
    if(a == 49999999) begin
      a=0;
      if(bcd == N-1) begin bcd = 0; mr10 = 0; mr = 0; end
      else begin
        bcd = bcd + 1;
        if(bcd <10) begin mr10 = 0; mr = bcd; end
        else if(bcd <20) begin mr10 = 1; mr = bcd-10; end
        else if(bcd <30) begin mr10 = 2; mr = bcd-20; end
        else if(bcd <40) begin mr10 = 3; mr = bcd-30; end
        else if(bcd <50) begin mr10 = 4; mr = bcd-40; end
        else begin mr10 = 5; mr = bcd-50; end
      end
      if(!stopwatch) begin
        check=0;
        case (mr)
          0:display = 7'b000_0001;
          1:display = 7'b100_1111;
          2:display = 7'b001_0010;
          3:display = 7'b000_0110;
          4:display = 7'b100_1100;
          5:display= 7'b010_0100;
          6:display = 7'b010_0000;
          7:display = 7'b000_1111;
          8:display = 7'b000_0000;
          9:display = 7'b000_0100;
          default: display = 7'b100_1111;
        endcase
        case (mr10)
          0:display10 = 7'b000_0001;
          1:display10 = 7'b100_1111;

          2:display10 = 7'b001_0010;
          3:display10 = 7'b000_0110;
          4:display10 = 7'b100_1100;
          5:display10 = 7'b010_0100;
          default: display10 = 7'b100_1111;
        endcase
      end
    end
    else begin
      a= a+1;
    end
  end
end

if(stop_enable_min) begin //stopwatch
  check_return=1;
  if(stop_a == 4999999) begin
    stop_a=0;
    if(stop_bcd == N-1) begin stop_bcd = 0; stop_mr10 = 0; stop_mr = 0; end
    else begin
      stop_bcd = stop_bcd + 1;
      if(stop_bcd <10) begin stop_mr10 = 0; stop_mr = stop_bcd; end
      else if(stop_bcd <20) begin stop_mr10 = 1; stop_mr = stop_bcd-10; end
      else if(stop_bcd <30) begin stop_mr10 = 2; stop_mr = stop_bcd-20; end
      else if(stop_bcd <40) begin stop_mr10 = 3; stop_mr = stop_bcd-30; end
      else if(stop_bcd <50) begin stop_mr10 = 4; stop_mr = stop_bcd-40; end
      else begin stop_mr10 = 5; stop_mr = stop_bcd-50; end
    end

    case (stop_mr)
      0:display = 7'b000_0001;
      1:display = 7'b100_1111;
      2:display = 7'b001_0010;
      3:display = 7'b000_0110;

      4:display = 7'b100_1111;
      5:display= 7'b010_0100;
      6:display = 7'b010_0000;
      7:display = 7'b000_1111;
      8:display = 7'b000_0000;
      9:display = 7'b000_0100;
      default: display = 7'b100_1111;
    endcase
    case (stop_mr10)
      0:display10 = 7'b000_0001;
      1:display10 = 7'b100_1111;
      2:display10 = 7'b001_0010;
      3:display10 = 7'b000_0110;
      4:display10 = 7'b100_1100;
      5:display10 = 7'b010_0100;
      default: display10 = 7'b100_1111;
    endcase
  end else stop_a =stop_a + 1;
end
end
endmodule

```

=> 분 시계와 분 스탑워치 동작을 해주는 모듈이며 앞에 보았던 초 시계와 초 스탑워치와 기능은 거의 다를 것이 없다.

<소스코드(시계의 시 부분)

```
module count_display_hour (clk, reset, enable, display, display10, stop_enable_sec, stopwatch, add_hour);
parameter N=24, M=5;
input clk, reset, enable, stop_enable_sec, add_hour, stopwatch;
output [6:0] display, display10;
reg [M-1:0] bcd;
reg [6:0] display, display10;
integer a, hr, hr10, check;

always @(posedge clk or posedge reset) begin
3   if (reset) begin
3       bcd = 0;
        hr=0;
        hr10=0;
        display = 7'b000_0001;
        display10 = 7'b000_0001;
    end
3   else if(stopwatch & (check==0))begin
        display = 7'b000_0001;
        display10 = 7'b000_0001;
        check=1;
    end
3   else if(add_hour) begin
3       if(a == 49999999) begin
            a=0;
            if(bcd == N-1) begin bcd = 0; hr10 = 0; hr = 0; end
3         else begin
            bcd = bcd + 1;
            if(bcd < 10) begin hr10 = 0; hr = bcd; end
            else if(bcd < 20) begin hr10 = 1; hr = bcd - 10; end
            else begin hr10 = 2; hr = bcd - 20; end
        end
3         if(!stop_enable_sec)begin
3             case (hr)
                    0:display = 7'b000_0001;
                    1:display = 7'b100_1111;
                    2:display = 7'b001_0010;
                    3:display = 7'b000_0110;
                    4:display = 7'b100_1100;
                    5:display= 7'b010_0100;
                    6:display = 7'b010_0000;
                    7:display = 7'b000_1111;
                    8:display = 7'b000_0000;
                    9:display = 7'b000_0100;
                    default: display = 7'b100_1111;
                endcase
                case (hr10)
                    0:display10 = 7'b000_0001;
                    1:display10 = 7'b100_1111;
                    2:display10 = 7'b001_0010;
                    default: display10 = 7'b100_1111;
                endcase
            end
        end
        else
            a = a + 1;
    end
3   else if(enable) begin
3       if(a == 49999999) begin
            a=0;
            if(bcd == N-1) begin bcd = 0; hr10 = 0; hr = 0; end
            else begin
                bcd = bcd + 1;
                if(bcd < 10) begin hr10 = 0; hr = bcd; end
                else if(bcd < 20) begin hr10 = 1; hr = bcd - 10; end
                else begin hr10 = 2; hr = bcd - 20; end
            end
        end
        if(!stop_enable_sec)begin
```

```

end
if(!stop_enable_sec)begin
    case (hr)
        0:display = 7'b000_0001;
        1:display = 7'b100_1111;
        2:display = 7'b001_0010;
        3:display = 7'b000_0110;
        4:display = 7'b100_1100;
        5:display= 7'b010_0100;
        6:display = 7'b010_0000;
        7:display = 7'b000_1111;
        8:display = 7'b000_0000;
        9:display = 7'b000_0100;
        default: display = 7'b100_1111;
    endcase
    case (hr10)
        0:display10 = 7'b000_0001;
        1:display10 = 7'b100_1111;
        2:display10 = 7'b001_0010;
        default: display10 = 7'b100_1111;
    endcase
end
else
    a = a + 1;
    if(stop_enable_sec)begin display =7'b000_0001; display10 = 7'b000_0001;end
end
end
endmodule

```

⇒ 시계의 시간 부분을 담당해주는 모듈이다. 스톱워치는 시간까지는 가지 않기 때문에 스톱워치 역할은 하지 않고 있고 그 대신 스톱워치로 돌아갈 때 시간이 0으로 보여지게 구현하려 하였는데 왜 인지 모르게 제 구현을 못하였다.

<디스커션>

처음 계획을 세우고 머리 속으로 구상하였을 때는 그다지 어려울 것이라는 생각은 들지 않았었다. 시계를 만들고 스톱워치 기능, 시간 세트 기능, 알람 기능 이렇게 3기능을 추가할 계획이었지만 스톱워치에만 시간이 엄청 오래걸려 결국 알람 기능은 시도도 못해보았다. 일단 시계는 modulo를 수업시간에 배웠기 때문에 만드는데 큰 시간이 걸리지 않았다. 어려운 부분도 없었고 바로 다음 기능인 스톱워치 기능으로 넘어갔다. 스톱워치는 0.01초 단위로 움직이기 때문에 클럭만 잘 조절해주면 금방 구현할 수 있을 줄 알았는데 스톱워치와 시계와의 연동에서 문제가 많이 발생하였다. 일단 스톱워치가 동작하는 동안에도 시계도 같이 움직이게 하는 것은 별로 어렵지 않았는데 계속해서 스톱워치 값이 시계에 따라오고 시계 값이 스톱워치 값에 따라오는 현상이 일어났다. 따라서 조건문을 주어서 해결하려 하였는데 이것도 만만치가 않았다. 일단 언어자체가 always에서 처음 if(reset)문 else if(enable) 이런 식으로 되어있는데 여기서 나는 조건을 주기 위해서 else if 밑에 따로 if문을 선언하려 하였지만 실행 자체가 안되어서 enable조건을 바꾼 다음에 그 안에다가 조건을 새로 만드는 방식으로 해결하였다. 또한 분 모듈에서 계속 스톱워치의 분이 동작이 제대로 안 하는 것을 발견하였는데 몇 번을 돌려가면서 확인하고 깨달은 결과, enable은 시계의 enable이기 때문에 시계가 분 동작을 할 때에만 동작을 enable이 되기 때문에 그 안에 넣으면 서로 시간 오차가 발생하기 때문이었다. 이것도 위와 같이 enable조건에 새로 조건을 추가하여 해결하였다. 이러한 결과로 스톱워치와 시계 간 전환에서, 시간이 자신의 모드의 시간으로 정확하게

변환이 되게 만들었는데 변환에 딜레이가 생기는 부분도 있었고 시 부분에서는 스톱워치 전환시에 시계의 시간 부분이 남아있는 상태가 되었다. 이 부분은 분과 초에서 시간을 바꾼 것처럼 똑같이 구성해주면 해결할 수 있을거라고 생각한다. 마지막으로 구현하지 못했던 알람 기능에 대한 원래 생각했던 알고리즘을 설명하고 끝마치겠다. 먼저 세트를 구현해놨기 때문에 그 세트 기능에 덧붙여 감소하는 modulo를 만들어주면 알람은 쉽게 구현되지 않았을까 싶다.