

```
1  /*****
2  Target MCU & clock speed: ATtiny13A @ 1.2Mhz internal
3  Name      : main.c
4  C modules of this project, ISR:
5  main.c
6  Custom Headers:
7      Nothing
8  Author   : Insoo Kim (insoo@hotmail.com)
9  Created  : May 15, 2015
10 Updated  : Sep 03, 2018 (On Atmel Studio 7)
11
12 Description:
13     ATtiny13A controls 220V power relay for multiple purposes using three  ↗
14     slide switches.
15     In this particular application for managing submersed water pump in fish  ↗
16     tank,
17     I don't care slide switches, but only works in a single mode of operation. ↗
18
19 HEX size[Byte]: 218 out of 1024
20
21 How to upload to the target MCU
22 <For Windows Atmel Studio>
23 Slect Tool ? USBtiny (USBtiny memu should be configured in the external tool  ↗
24 memu)
25
26 <For CMD window or DOS prompt>
27 cd " C:\Users\Winsoo\Documents\GitHub\ATmelStudio\ATtiny13A\ClockGenWISR\Debug  ↗
28 "
29
30 avrdude -c usbtiny -P usb -p attiny13 -U flash:w:ISR.hex:i
31
32 Ref:
33 *****/
34
35 #include <avr/interrupt.h>
36 #include <avr/sleep.h>
37 #include <avr/io.h>
38 #include <util/delay.h>
39
40 // #define TEST_MODE
41 #define PRODUCTION_MODE
42
43 #ifdef TEST_MODE
44     #define UNIT_DELAY_WDT 4 //<=== TEST VALUE in development; 4 sec WDT
45     #define INITPUMP_DURATION 5000 // 5 sec for initial pump period
46 #endif
47 #ifdef PRODUCTION_MODE
48     #define UNIT_DELAY_WDT 8 //<=== SELECTED VALUE in production; WDT period  ↗
49     in seconds; 8 sec WDT
50     #define INITPUMP_DURATION 10000 // 60 sec for initial pump period
51 #endif
52
53 //select accumulated WDT period as DELAY_UNIT = UNIT_DELAY_WDT *
54 SET_DELAY_UNIT
55
56 #ifdef TEST_MODE
57     #define SET_DELAY_UNIT 2 //<=== TEST VALUE in development; 4*3=12 [sec]
```

```

50 #endif
51 #ifdef PRODUCTION_MODE
52     #define SET_DELAY_UNIT 225 //<=== SELECTED VALUE in production; 8*225=1800 ↗
        [sec]=30 minutes
53 #endif
54 // # of UNIT_DELAY_WDT, Max 253
55 // #define SET_DELAY_UNIT 2 // <=== TEST VALUE in development
56 // #define SET_DELAY_UNIT 15 // 2 min (15*8=120[sec]) when UNIT_DELAY_WDT is 8
57 // #define SET_DELAY_UNIT 150 // 20 min when UNIT_DELAY_WDT is 8
58 // #define SET_DELAY_UNIT 225 //<=== SELECTED VALUE in production; 30 min when ↗
        UNIT_DELAY_WDT is 8
59 // #define SET_DELAY_UNIT 253 // (34 min - 8 sec) when UNIT_DELAY_WDT is 8
60
61 //select accumulated DELAY_UNIT as WAKEUP period = SET_DELAY_UNIT * ↗
        WAKEUP_PERIOD
62 #ifdef TEST_MODE
63     //4*3*1=12 [sec], every 12 sec system will wakeup from sleep and do ISR
64     // #define WAKEUP_PERIOD 1 //<=== TEST VALUE in development;
65     uint8_t wakeupPeriod = 1;
66 #endif
67 #ifdef PRODUCTION_MODE
68     //8*225*2=3600[sec]=1 [hour], every hour system will wakeup from sleep and ↗
        do ISR
69     // #define WAKEUP_PERIOD 2 //<=== SELECTED VALUE in production;
70         // 2:one hour, 4:two hours, etc
71         // when SET_DELAY_UNIT is 225 of ATtiny13a at ↗
            1.2Mhz
72     uint8_t wakeupPeriod = 2;
73 #endif
74
75 //select WAKEUP duration,i.e. how long ISR function will persist.
76
77 #ifdef TEST_MODE
78     // #define wakeupDuration 75 // 10 min operation, Refer the comments of ↗
        SET_DELAY_UNIT
79     // #define wakeupDuration 1 // 4 sec operation, Refer the comments of ↗
        SET_DELAY_UNIT
80     uint8_t wakeupDuration = 1; // 4 sec operation, Refer the comments of ↗
        SET_DELAY_UNIT
81 #endif
82 #ifdef PRODUCTION_MODE
83     // #define wakeupDuration 75 // 10 min operation, Refer the comments of ↗
        SET_DELAY_UNIT
84     // #define wakeupDuration 150 // 20 min operation, Refer the comments of ↗
        SET_DELAY_UNIT
85     uint8_t wakeupDuration = 150; // 4 sec operation, Refer the comments of ↗
        SET_DELAY_UNIT
86 #endif
87
88 //port for the transistor to control relay on/off
89 #define NPN_TR_PORT PB0 // when using NPN TR
90
91 uint8_t WDTtick;
92 uint8_t WDTtick30min;
93 //uint8_t readBit;
94 ISR(WDT_vect)

```

```

95 {
96     //readBit = PINB & PB3 ;
97     //if (readBit == 1)
98     //if(bit_it_set(PINB, PB3))
99     if (PINB & (1<<PB3))
100         //wakeupDuration = 2;
101         wakeupPeriod = 1; // every 30min in production mode
102     else
103         //wakeupDuration = 1;
104         wakeupPeriod = 2; // every 60min in production mode
105
106     // ----- INCREASE WDT TICK COUNT -----
107     // increase WDTtick every UNIT_DELAY_WDT sec
108     ++WDTtick;
109
110     // On every SET_DELAY_UNIT (half-hour) except last half-hour
111     // reset WDTtick count and increase half-hour count
112     if ((WDTtick >= SET_DELAY_UNIT) && (WDTtick30min < wakeupPeriod))
113     {
114         // Reset WDT counter value of the designated address in the EEPROM of ATtiny13A
115         //eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);
116         WDTtick = 0;
117
118         //Increase WDT Half-hour counter value of the designated address in the EEPROM of ATtiny13A
119         //eeprom_update_byte((uint8_t*)WDTTICK_30MIN_ADDR, ++WDTtick30min);
120         ++WDTtick30min;
121     } //if (WDTtick >= SET_DELAY_UNIT) && (WDTtick30min < WAKEUP_PERIOD)
122
123     // On every wakeupPeriod (i.e. one hour in production mode)
124     if (WDTtick30min >= wakeupPeriod)
125     {
126         // ----- DO TO PROPER ACTION TO WDT TICK COUNT -----
127
128         // When the accumulated WDT reaches every SET_DELAY_UNIT, turn on ESP-01
129         if (WDTtick == 0)
130         {
131             // Give logic HIGH to port 4 to turn ON NPN transistor(2n2222),
132             // so let the GND of ESP-01 module CONNECT to system GND.
133             // This will power ON ESP-01 and measure temperature & humidity via DHT22
134             PORTB = 1<<NPN_TR_PORT; //turn on GND of MOSFET or ESP-01
135         } //if (WDTtick == 0)
136         else if (WDTtick >= wakeupDuration)
137         {
138             // Give logic LOW to port 4 to turn OFF NPN transistor(2n2222),
139             // so let the GND of ESP-01 module DISCONNECT to system GND.
140             // This will power OFF ESP-01 and don't measure temperature & humidity via DHT22
141             PORTB = (0<<NPN_TR_PORT); //turn off GND of MOSFET or ESP-01
142
143             // Reset WDT counter value of the designated address in the EEPROM of ATtiny13A
144             //eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);

```

```

144         WDTtick = 0;
145         //Reset WDT Half-hour counter value of the designated address in the EEPROM of ATtiny13A
146         //eeprom_update_byte((uint8_t*)WDTTICK_30MIN_ADDR, 0);
147         WDTtick30min=0;
148         }//else if (WDTtick == 1)
149     }//if (WDTtick30min >= wakeupPeriod)
150
151
152 }//ISR(WDT_vect)
153
154 int main(void) {
155
156     WDTtick=0;
157     WDTtick30min=0;
158     // Set NPN_TR_PORT mode to output
159     DDRB = (1<<NPN_TR_PORT);
160     // Set PB1-3 mode to input
161     DDRB &= ~(1<<PB1);
162     DDRB &= ~(1<<PB2);
163     DDRB &= ~(1<<PB3);
164
165     PORTB = 1<<NPN_TR_PORT; //turn on MOSFET
166
167     _delay_ms(INITPUMP_DURATION); // initial measurement of DHT22 via ESP-01 for 10 sec
168
169     PORTB = 0<<NPN_TR_PORT; //turn off MOSFET
170
171     // temporarily prescale timer to UNIT_DELAY_WDT seconds so we can measure current
172     switch (UNIT_DELAY_WDT)
173     {
174         case 4:
175             WDTCR |= (1<<WDP3); // 4s
176             break;
177         case 8:
178             WDTCR |= (1<<WDP3) | (1<<WDPO); // 8s
179             break;
180         default:
181             WDTCR |= (1<<WDP3) | (1<<WDPO); // 8s
182     }
183
184     // Enable watchdog timer interrupts
185     WDTCR |= (1<<WDTIE);
186
187     sei(); // Enable global interrupts
188
189     // Use the Power Down sleep mode
190     set_sleep_mode(SLEEP_MODE_PWR_DOWN);
191     //set_sleep_mode(SLEEP_MODE_IDLE);
192
193     for (;;) {
194         sleep_mode(); // go to sleep and wait for interrupt...
195     }
196 }//main

```

197

198