```
/*************************************************************
 Target MCU & clock speed: ATtiny13A @ 1.2Mhz internal
 Name    : main.c
 C modules of this project, ISR:
main.c
Custom Headers:
    Nothing
 Author  : Insoo Kim (insoo@hotmail.com)
 Created : May 15, 2015
 Updated : Aug 18, 2018 (On Atmel Studio 7)

 Description:
    ATtiny13A controls power up or down to ESP-01 module by 2n2222 NPN transistor.
    ATtiny13A sleeps in most of operation time and wake up periodically to measure temperature and humidity by DHT22 attached to ESP-01.

 HEX size[Byte]: 308 out of 1024

How to upload to the target MCU
<For Windows Atmel Studio>
Slect Tool - USBtiny (USBtiny memu should be configured in the external tool memu)

<For CMD window or DOS prompt>
cd " C:₩Users₩Insoo₩Documents₩GitHub₩ATmelStudio₩ATtiny13A₩ClockGen₩ISR₩Debug "
avrdude -c usbtiny -P usb -p attiny13 -U flash:w:ISR.hex:i

 Ref:
*************************************************************/

#include <avr/interrupt.h>
#include <avr/sleep.h>

//#include <util/delay.h>
#include <avr/eeprom.h>

//#define UNIT_DELAY_WDT 4 //<=== TEST VALUE in development;
#define UNIT_DELAY_WDT 8 //<=== SELECTED VALUE in production; WDT period in seconds

//#define SET_DELAY_UNIT 2  // <=== TEST VALUE in development
//#define SET_DELAY_UNIT 15  // # of UNIT_DELAY_WDT, Max 253: 2 min when UNIT_DELAY_WDT is 8
//#define SET_DELAY_UNIT 150 // # of UNIT_DELAY_WDT, Max 253: 20 min
#define SET_DELAY_UNIT 225 //<=== SELECTED VALUE in production; # of UNIT_DELAY_WDT, Max 253: 30 min
//#define SET_DELAY_UNIT 253 // # of UNIT_DELAY_WDT, Max 253: 34 min - 8 sec

#define WDTTICK_CTR_ADDR 0
#define MEASURETICK_CTR_ADDR 1

#define NPN_TR_PORT PB4

uint8_t WDTtick = 0;
```

```
50  uint8_t WDTtickMeasure = 0;
51
52  ISR(WDT_vect)
53  {
54      // On every watch dog timer interrupt,
55      //  get the WDTtick counter value every UNIT_DELAY_WDT sec
56      //  from the designated EEPROM address
57      WDTtick = eeprom_read_byte((uint8_t*)WDTTICK_CTR_ADDR);
58
59      // increase WDTtick every UNIT_DELAY_WDT sec
60      //  and update it at the designated EEPROM address
61      eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, ++WDTtick);
62
63      // When the accumulated WDT reaches every SET_DELAY_UNIT, turn on ESP-01
64      if (WDTtick % SET_DELAY_UNIT == 0) //every UNIT_DELAY_WDT * SET_DELAY_UNIT ↵
           sec
65      {
66          // Give logic HIGH to port 4 to turn ON NPN transistor(2n2222),
67          //  so let the GND of ESP-01 module CONNECT to system GND.
68          // This will power ON ESP-01 and measure temperature & humidity via    ↵
              DHT22
69          PORTB = 1<<NPN_TR_PORT; //turn on GND of ESP-01
70
71          // save current WDT counter number(WDTtickMeasure) to EEPROM
72          WDTtickMeasure = WDTtick;
73          eeprom_update_byte((uint8_t*)MEASURETICK_CTR_ADDR, WDTtickMeasure);
74      }//if (WDTtick % SET_DELAY_UNIT == 0)
75
76      // When reaching to the next tick after turning on ESP-01, turn off ESP-01
77      //if (WDTtick == SET_DELAY_UNIT + 1)
78      if (WDTtick == WDTtickMeasure + 1)
79      {
80          // Give logic LOW to port 4 to turn OFF NPN transistor(2n2222),
81          //  so let the GND of ESP-01 module DISCONNECT to system GND.
82          // This will power OFF ESP-01 and don't measure temperature & humidity ↵
               via DHT22
83          PORTB = (0<<NPN_TR_PORT); //turn off GND of ESP-01
84
85          // Reset WDT counter value of the designated address in the EEPROM of  ↵
              ATtiny13A
86          eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);
87
88          // Reset relevant variables
89          WDTtickMeasure = 255;
90          WDTtick = 1;
91      }// if (WDTtick == WDTtickMeasure + 1)
92
93  }//ISR(WDT_vect)
94
95  int main(void) {
96      // Set up NPN_TR_PORT & PNP_TR_PORT mode to output
97      //DDRB = 1<<DDB4;
98      DDRB = (1<<NPN_TR_PORT);
99
100     //DDRB = (1<<NPN_TR_PORT) | (1<<PNP_TR_PORT);
101
```

```
102     //PORTB = 0<<NPN_TR_PORT;
103     // temporarily prescale timer to UNIT_DELAY_WDT seconds so we can measure ↵
          current
104     switch (UNIT_DELAY_WDT)
105     {
106         case 4:
107             WDTCR |= (1<<WDP3); // 4s
108             break;
109         case 8:
110             WDTCR |= (1<<WDP3) | (1<<WDP0); // 8s
111             break;
112         default:
113             WDTCR |= (1<<WDP3) | (1<<WDP0); // 8s
114     }
115     // (1<<WDP2) | (1<<WDP0);
116
117     // Enable watchdog timer interrupts
118     WDTCR |= (1<<WDTIE);
119
120     sei(); // Enable global interrupts
121
122     // Reset the WDTtick at the designated EEPROM address
123     eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);
124
125     // Use the Power Down sleep mode
126     set_sleep_mode(SLEEP_MODE_PWR_DOWN);
127
128     for (;;) {
129         sleep_mode();   // go to sleep and wait for interrupt...
130     }
131 }//main
132
133
```