```
/*************************************************************
 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 20, 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]: 376 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

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


#define WAKEUP_PERIOD   2 // <=== TEST VALUE in development
//#define WAKEUP_PERIOD 2 // 2:one hour, 4:two hours,
                          // when SET_DELAY_UNIT is 225 of ATtiny13a at 1.2Mhz

#define USE_NPN
```

```
51  //#define USE_PNP
52
53  #ifdef USE_NPN
54      #define NPN_TR_PORT PB4 // when using NPN TR
55  #endif
56  #ifdef USE_PNP
57      #define PNP_TR_PORT PB4 // when using PNP TR
58  #endif
59
60  uint8_t WDTtick;
61  uint8_t WDTtick30min;
62  uint8_t i;
63
64  ISR(WDT_vect)
65  {
66      /*
67      for(i=0; i<WDTtick30min; i++)
68      {
69          PORTB = _BV(NPN_TR_PORT);
70          _delay_ms(200);
71          PORTB = ~_BV(NPN_TR_PORT);
72          _delay_ms(200);
73      }
74      */
75
76      // --------- HOW MANY WDT HAS OCCURED ? ---------------------------------
77      // On every watch dog timer interrupt,
78      //  get the WDTtick counter value every UNIT_DELAY_WDT sec
79      //  from the designated EEPROM address
80      //WDTtick30min = eeprom_read_byte((uint8_t*)WDTTICK_30MIN_ADDR);
81      //WDTtick = eeprom_read_byte((uint8_t*)WDTTICK_CTR_ADDR);
82
83      // On every one hour or from the 1st beginning of the system
84
85      if (WDTtick30min >= WAKEUP_PERIOD)
86      {
87          // --------- DO TO PROPER ACTION TO WDT TICK COUNT                    ↵
              ----------------------
88          // When the accumulated WDT reaches every SET_DELAY_UNIT, turn on    ↵
              ESP-01
89          if (WDTtick == 0)
90          {
91              // Give logic HIGH to port 4 to turn ON NPN transistor(2n2222),
92              //  so let the GND of ESP-01 module CONNECT to system GND.
93              // This will power ON ESP-01 and measure temperature & humidity  ↵
                  via DHT22
94              #ifdef USE_NPN
95                  PORTB = 1<<NPN_TR_PORT; //turn on GND of MOSFET or ESP-01
96              #endif
97              #ifdef USE_PNP
98                  PORTB = 0<<PNP_TR_PORT; // turn on MOSFET Vin
99              #endif
100         }//if (WDTtick == 0)
101         else if (WDTtick == 1)
102         {
103             // Give logic LOW to port 4 to turn OFF NPN transistor(2n2222),
```

```
104             //  so let the GND of ESP-01 module DISCONNECT to system GND.
105             // This will power OFF ESP-01 and don't measure temperature &
                  humidity via DHT22
106             #ifdef USE_NPN
107                 PORTB = (0<<NPN_TR_PORT); //turn off GND of MOSFET or ESP-01
108             #endif
109             #ifdef USE_PNP
110                 PORTB = 1<<PNP_TR_PORT; // turn off MOSFET Vin
111             #endif
112
113             // Reset WDT counter value of the designated address in the EEPROM
                   of ATtiny13A
114             //eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);
115             WDTtick = 0;
116             //Reset WDT Half-hour counter value of the designated address in
                  the EEPROM of ATtiny13A
117             //eeprom_update_byte((uint8_t*)WDTTICK_30MIN_ADDR, 0);
118             WDTtick30min=0;
119         }//else if (WDTtick == 1)
120     }
121
122     // On every half-hour except last half-hour
123     if ((WDTtick >= SET_DELAY_UNIT) && (WDTtick30min < WAKEUP_PERIOD))
124     //if (WDTtick >= SET_DELAY_UNIT)
125     {
126         // Reset WDT counter value of the designated address in the EEPROM of
              ATtiny13A
127         //eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, 0);
128         WDTtick = 0;
129
130         //Increase WDT Half-hour counter value of the designated address in
              the EEPROM of ATtiny13A
131         //eeprom_update_byte((uint8_t*)WDTTICK_30MIN_ADDR, ++WDTtick30min);
132         ++WDTtick30min;
133     }//if (WDTtick >= SET_DELAY_UNIT)  && (WDTtick30min < WAKEUP_PERIOD)
134
135     // --------- INCREASE WDT TICK COUNT ----------------------
136     // increase WDTtick every UNIT_DELAY_WDT sec
137     //  and update it at the designated EEPROM address
138     if (WDTtick < SET_DELAY_UNIT)
139         //eeprom_update_byte((uint8_t*)WDTTICK_CTR_ADDR, ++WDTtick);
140         ++WDTtick;
141
142
143
144 }//ISR(WDT_vect)
145
146 int main(void) {
147
148     WDTtick=0;
149     WDTtick30min=0;
150     // Set up NPN_TR_PORT & PNP_TR_PORT mode to output
151     #ifdef USE_NPN
152         DDRB = (1<<NPN_TR_PORT);
153     #endif
154     #ifdef USE_PNP
```

```
155         DDRB = (1<<PNP_TR_PORT);
156     #endif
157
158     #ifdef USE_NPN
159         PORTB = 1<<NPN_TR_PORT; //turn on MOSFET
160     #endif
161     #ifdef USE_PNP
162         PORTB = 0<<PNP_TR_PORT; //turn on MOSFET
163     #endif
164
165     _delay_ms(1000);
166
167     #ifdef USE_NPN
168         PORTB = 0<<NPN_TR_PORT; //turn off MOSFET
169     #endif
170     #ifdef USE_PNP
171         PORTB = 1<<PNP_TR_PORT; //turn off MOSFET
172     #endif
173     // temporarily prescale timer to UNIT_DELAY_WDT seconds so we can measure ↩
        current
174     switch (UNIT_DELAY_WDT)
175     {
176         case 4:
177             WDTCR |= (1<<WDP3); // 4s
178             break;
179         case 8:
180             WDTCR |= (1<<WDP3) | (1<<WDP0); // 8s
181             break;
182         default:
183             WDTCR |= (1<<WDP3) | (1<<WDP0); // 8s
184     }
185
186     // Enable watchdog timer interrupts
187     WDTCR |= (1<<WDTIE);
188
189     sei(); // Enable global interrupts
190
191     // Use the Power Down sleep mode
192     set_sleep_mode(SLEEP_MODE_PWR_DOWN);
193     //set_sleep_mode(SLEEP_MODE_IDLE);
194
195     for (;;) {
196         sleep_mode();   // go to sleep and wait for interrupt...
197     }
198 }//main
199
200
```