Instantly share code, notes, and snippets.

boseji / wdt_sleep.ino

Created a year ago

Arduino Pro Mini Low power Sleep Example

<> wdt_sleep.ino

```
/**
 * Arduino Pro Mini Low power Sleep Example
 *
 * @license BSD Open Source License
 * Copyright (c) 2004-2021 Abhijit Bose &lt; salearj [at] hotmail [dot] com &gt;
 * If you require a license, see
 *    http://www.opensource.org/licenses/bsd-license.php
 *
 * @note
 * This example show how to properly enter sleep mode and wake up
 * the Arduino Pro Mini board using Watchdog Timer.
 * Additionally this example reads a LDR sensor connected to A6 Pin
 * and a DHT22 sensor connected to the D8 pin of the Pro Mini board.
 * In case you would like to remove the DHT Portion you can remove
 * the necessary code.
 * The design is such that the code within the WDT flag guarded `if`
 * is only executed ones after which the board sleeps for the desired
 * time and then wakes up back to continue.
 * Good part of using Watchdog Timer is that it remains enabled even
 * in the lowest of the power modes.
 * Here are the Connected items to the Pro Mini Board:
 * A6 = LDR sensor
 * D8 = DHT22 (Dout pin)
 * D9 = Indicator LED (Active Low)
 *
 * @warning At the end of all serial sequences it needs some delay else
 * even before the transmission could complete the board might get reset
 * or garbage data starts flowing out due to misalignments in the serial
 * shift register in the UART during sleep entry.
 *
 * Attributions:
 * This work extends the examples provided by Donal Morrissey.
 * His article: https://donalmorrissey.blogspot.com/2010/04/sleeping-arduino-part-5-wake-up-via.html
 */

#include <Arduino.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <avr/power.h>
#include <avr/wdt.h>

// DHT Library
#include <dht.h>
//////////////////////////////////////////////////////////////////////

// DHT Sensor Pin Declaration
#define DHT22_PIN 8
// Instance of SHT Sensor connected to Pin 8 of Pro Mini
dht DHT;
// WDT entry Flag
volatile uint8_t f_wdt=1;


//////////////////////////////////////////////////////////////////////
```

```
55   // WDT Interrupt
56
57   ISR(WDT_vect)
58   {
59           if(f_wdt == 0)
60           {
61                   f_wdt=1; // Reset the Flag
62           }
63           else
64           {
65                   Serial.println(F("WDT Overrun!!!"));
66           }
67   }
68
69   /////////////////////////////////////////////////////////////////////
70   // Sleep Configuration Function
71   //    Also wake-up after
72
73   void enterSleep(void)
74   {
75           WDTCSR |= _BV(WDIE); // Enable the WatchDog before we initiate sleep
76
77           //set_sleep_mode(SLEEP_MODE_PWR_SAVE);          /* Some power Saving */
78           set_sleep_mode(SLEEP_MODE_PWR_DOWN);           /* Even more Power Savings */
79           sleep_enable();
80
81           /* Now enter sleep mode. */
82           sleep_mode();
83           sleep_bod_disable();    // Additionally disable the Brown out detector
84
85           /* The program will continue from here after the WDT timeout*/
86           sleep_disable(); /* First thing to do is disable sleep. */
87
88           /* Re-enable the peripherals. */
89           power_all_enable();
90   }
91
92   /////////////////////////////////////////////////////////////////////
93   // SETUP FUNCTION
94
95   void setup() {
96
97           /*** Setup the WDT ***/
98           cli();
99           /* Clear the reset flag. */
100          MCUSR &= ~(1<<WDRF);
101
102          /* In order to change WDE or the prescaler, we need to
103           * set WDCE (This will allow updates for 4 clock cycles).
104           */
105          WDTCSR |= (1<<WDCE) | (1<<WDE);
106
107          /* set new watchdog timeout prescaler value */
108          WDTCSR = 1<<WDP1 | 1<<WDP2;            /* 1.0 seconds */
109    //WDTCSR = 1<<WDP0 | 1<<WDP1 | 1<<WDP2; /* 2.0 seconds */
110    //WDTCSR = 1<<WDP3;                     /* 4.0 seconds */
111    //WDTCSR = 1<<WDP0 | 1<<WDP3;           /* 8.0 seconds */
112
113          /* Enable the WD interrupt (note no reset). */
114          //WDTCSR |= _BV(WDIE); // Not here but when we go to Sleep
115          sei();
116
117          pinMode(9, OUTPUT);
118          digitalWrite(9,LOW);
119          Serial.begin(9600);
120          Serial.println(F("Initialization complete."));
121          delay(10); //Allow for serial print to complete.
```

```
122    }
123
124    /////////////////////////////////////////////////////////////////
125    // LOOP FUNCTION
126
127    void loop() {
128
129            // Only Execute this part One time
130            if(f_wdt == 1) {
131
132                    /////////////////////////////
133                    // PROCESSING BEGIN
134
135                    digitalWrite(9,LOW);                              // LED Indication ON
136                    Serial.print(analogRead(A6));           // Read and Print the Analog Input Pin used for LDR
137
138                    // Read and Print the DHT pin
139                    if (DHT.read22(DHT22_PIN) == 0) {
140                            Serial.print(F(" Humidity: "));
141                            Serial.print(DHT.humidity);
142                            Serial.print(F(" Temp: "));
143                            Serial.print(DHT.temperature);
144                    }
145
146                    Serial.println();                                // Line Separator
147                    digitalWrite(9,HIGH);                            // LED Indication OFF
148                    delay(20); //Allow for serial print to complete.
149
150                    // PROCESSING END
151                    /////////////////////////////
152
153                    /* Don't forget to clear the flag. */
154                    f_wdt = 0;
155
156                    /* Re-enter sleep mode. */
157                    enterSleep();
158            }
159
160    }
161
162    /////////////////////////////////////////////////////////////////
163    // END OF FILE
```