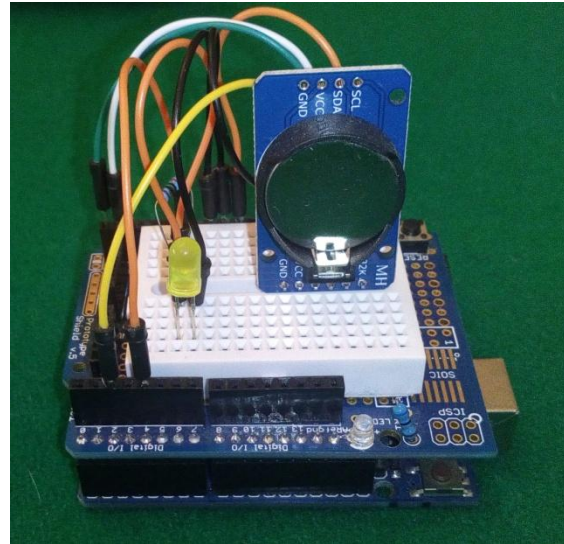
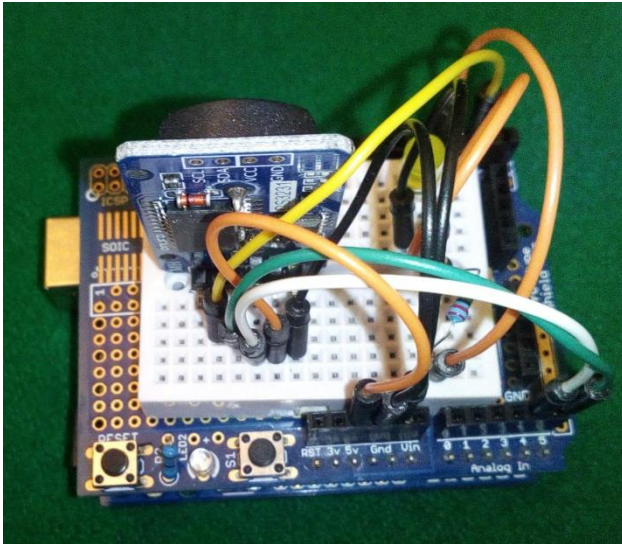


45 bis - Sleep e wake su interrupts da un orologio digitale

Sleep and wake on interrupts by RTC (some notes at the end of this section)



Nella scheda dell'esercizio 44 e' stato trattato l'orologio digitale DS3231 mentre in quella dell'esercizio 45 sono presenti alcune indicazioni sul significato e sull'utilizzo dell'interrupt e della funzione sleep. Qui sotto i link alle due schede, la cui lettura consente di meglio comprendere questo esercizio.

- [44 - modulo RTC DS3231- orologio digitale](#)
- [45 – sleep e wake da interrupt esterno](#)

Ora vengono unite le due esperienze per ottenere un sistema in grado di entrare in stato di sleep quando non e' richiesto il suo intervento (e ridurre quindi il consumo di energia) e di essere risvegliato mediante un allarme lanciato da un orologio esterno. L'utilita' di un dispositivo di questo tipo e' apprezzabile nel momento in cui Arduino deve lavorare per brevi periodi in un ambiente privo di fonti di energia. In questo caso Arduino deve attingere energia da una batteria la cui

COM1 (Arduino Uno)

```
dormo - sleep.... 30/09/2016 11:30:00
sveglio - working.. 30/09/2016 11:30:10
dormo - sleep.... 30/09/2016 11:30:15
sveglio - working.. 30/09/2016 11:31:10
dormo - sleep.... 30/09/2016 11:31:15
```

durata deve essere per ovvi motivi massimizzata.

Esempi di impianti del genere possono essere le centraline di rilevazione meteorologica che devono rilevare e registrare la situazione ogni due o tre ore situate, qualche volta, in ambienti privi di energia oppure ancora un impianto di irrigazione da balcone

che, per sicurezza, si vuole mantenere scollegato sia dalla rete elettrica che dalla rete idrica. In quest'ultimo caso Arduino po' essere mantenuto in stato di sleep per molte ore ed essere svegliato una volta al giorno, al fine di controllare la situazione di umidita' del terreno ed eventualmente azionare una pompa che, prelevando acqua da un serbatoio, ne ripristina il livello ottimale.

Sotto l'aspetto pratico in questo esercizio si pone Arduino in stato di sleep e lo si risveglia al decimo secondo di ogni minuto. Lo si mantiene sveglio per cinque secondi durante i quali lo si fa "lavorare" (facendogli mantenere acceso un led) per poi riportarlo in stato di sleep, in attesa del successivo risveglio. In pratica Arduino dorme, si risveglia, lavora e quindi torna a dormire senza alcun intervento dell'operatore.

Prima di procedere alla compilazione e' necessario installare, se non gia' fatto, la libreria: DS3231.h, reperibile [qui](#)

Per le modalita' di installazione si rimanda all'iter gia' illustrato nei precedenti esercizi e comunque riassumibile in:

- download della libreria in forma compressa
- installazione via IDE->sketch->include library->add .zip library
- verifica di avvenuta installazione (la libreria deve essere nell'elenco presente in IDE->sketch->include library->contributed library)

Nota: Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube.

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a giocarduino@libero.it

Here some notes about this project, translated by google translator



Arduino falls asleep and is awakened by interrupts generated by a digital clock, DS3231 type. Notes about DS3231, sleeping, awakening and interrupts are present in sheets:

- [44 - modulo RTC DS3231- orologio digitale - digital clock](#)
- [45 – sleep e wake da interrupt esterno](#)

Under the practical aspect, this project keeps Arduino in sleep state and, at tenth second of every minute, wakes it up. Arduino is kept awake for five seconds during which it works (yellow LED is on) and then is brought back in a state of sleep, waiting for the next awakening. In practice Arduino sleeps, wakes up, works and then goes back to sleep without any operator intervention.

Before proceeding to program compilation must be installed, if not already done, the library:

- DS3231.h found [here](#).

For library installation, see process shown in previous examples, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation need verification: the library must be present in IDE-> sketch-> includes Library-> Contributed library

Note: This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

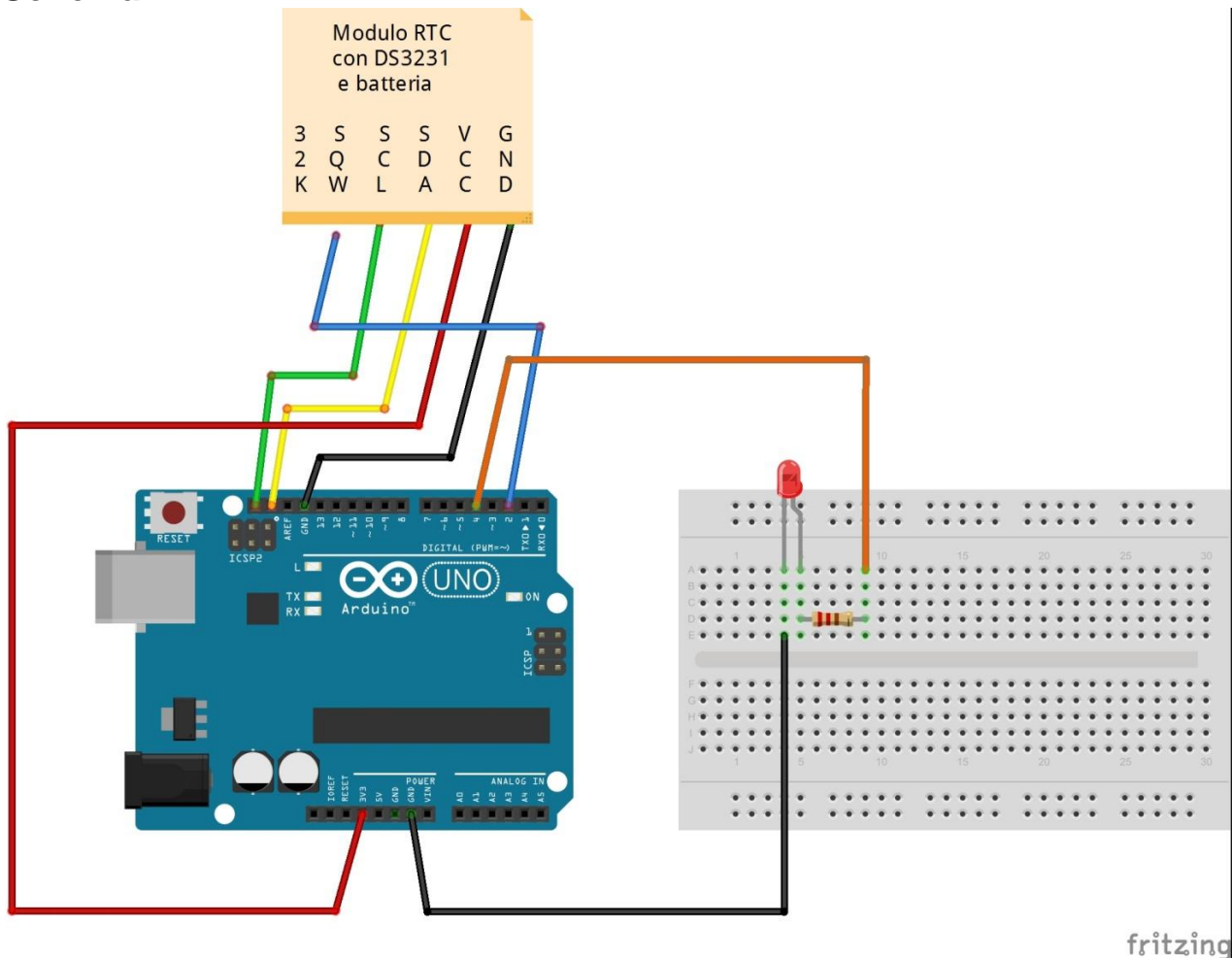
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to giocarduino@libero.it (simple words and short sentences, please)

Materiali

- Un modulo RTC DS3231
- Un led
- Una resistenza da 220 ohm
- Un po' di cavetteria

Schema



Programma

[Download del programma tramite Dropbox](#)
[Program download via Dropbox](#)

```
/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo.
 * Per rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T.
 * Questo programma e' derivato da un analogo programma reperito in rete
 * Arduino entra in stato di sleep e viene risvegliato, al decimo secondo di ogni
 * minuto, da un impulso lanciato da un orologio digitale (DS3231).
 * Una volta sveglio accende e mantiene acceso un led per 5 secondi dopo di che entra
 * di nuovo in stato di sleep, in attesa del successivo impulso proveniente dall'orologio
 *
 * per ottenere questo risultato e' stato necessario modificare il funzionamento del pin
 * SQW del modulo DS3231 intervenendo su di un byte di controllo interno al modulo.
 * Senza questa modifica il sistema non riesce ad addormentarsi
 *
 * -----
 * Warning: cut and paste from PDF to IDE loses formatting. To restore it press CTRL + T
 *
 * Tis program is based on program found on line
 *
 * Arduino wakes (from a sleep state) on interrupts launched from an external digital clock
 * (DS3231 RTC). To rise this goal must be change a control byte on DS3231.
```

Arduino: sleep e wake su interrupts da RTC DS3231

```
* Without that change Arduino doesn't go to sleep mode even the first time
* -----
*/
#include <Wire.h> //libreria wire, presente di default nell'IDE
#include <DS3231.h> // Libreria ds3231, https://github.com/jarzebski/Arduino-DS3231
#include <avr/sleep.h> // libreria avr/sleep, presente di default nell'IDE
DS3231 clock;
RTCDatetime dt;
const byte wakePin = 2; //Porta per il segnale di risveglio - pin used for alarm clock (interrupt 0)
byte temp_buffer = 0b11110111; // struttura del byte che disabilita il pin SQW sul modulo DS3231.
// Sembra inutile, ma senza di questo arduino non si addormenta. - Byte to disable SQW pin
// of DS3231, not logical yes, but without that the Arduino doesn't go to sleep mode even the first
// time
char print_date[16]; // memorizzazione della data ed ora fornita da ds3231 - Actual time repository
//
//
// ***** routine di modifica del byte di controllo DS3231 *****
// ***** Set DS3231 SQW control bytes *****
//
void modificabyte (byte control, bool which) // aggiorna il byte di controllo su ds3231 - Set
// DS3231 RTC control bytes
{
    Wire.beginTransaction(0x68);
    if (which) // which=false -> 0x0e, true->0x0f.
        Wire.write(0x0f);
    else
        Wire.write(0x0e);
    Wire.write(control);
    Wire.endTransmission();
}
//
//
// ***** routine di sveglia, attivata dall'interrupt (su allarme DS3231) *****
// *****wake routine, activated on interrupt from ds3231 *****
//
void svegliatoria()
{
    sleep_disable(); // disabilita la modalita' sleep (sveglia il sistema)
}
//
//
// ***** routine di "addormentamento" *****
// ***** asleep routine *****
//
void dormiora()
{
    Serial.print("dormo - sleep... ");
    dt = clock.getDateTime(); // chiede l'ora dall'orologio digitale DS3231 - get time form DS3231
    sprintf(print_date, "%02d/%02d/%d %02d:%02d:%02d", dt.day, dt.month, dt.year, dt.hour, dt.minute,
dt.second);
    Serial.println(print_date); // visualizza sul monitor seriale il momento dell'addormentamento -
//display sleeping time on serial monitor
    sleep_enable(); // abilita l'addormentamento del sistema
    delay(100);
    digitalWrite(4, LOW); // spegne il led 4, per segnalare che il sistema e' dormiente
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // Seleziona la modalita' "power down", che massimizza
// il risparmio di energia - Select "power down mode" to maximize energy saving
    cli(); // disabilita l'interrupt - disable interrupts
    sleep_bod_disable(); // disabilita il controllo interno di alimentazione (brown out detection)
    sei(); // riabilita l'interrupt - enable interrupts
    sleep_cpu(); // "addormenta" la cpu - sleep mode on cpu
    //
    // **questo e' il punto in cui Arduino riprende a lavorare, subito dopo la routine svegliatoria
    // *****arduino wake up here *****
    //
    clock.clearAlarm1(); // elimina l'allarme (per evitare ulteriori indesiderati interrupt)
    Serial.print("sveglia - working.. ");
    dt = clock.getDateTime(); // chiede l'ora dall'orologio digitale DS3231 - get time form DS3231
    sprintf(print_date, "%02d/%02d/%d %02d:%02d:%02d", dt.day, dt.month, dt.year, dt.hour, dt.minute,
dt.second);
    Serial.println(print_date); // visualizza sul monitor seriale il momento della sveglia
    //display wake time on serial monitor
}
//
//
void setup()
{
    Serial.begin(9600);
```

Arduino: sleep e wake su interrupts da RTC DS3231

```
clock.begin();
modificabyte(temp_buffer, 0); //lancia la routine di modifica del byte su ds3231 - call DS3231
byte modify routine
clock.armAlarm1(false);
clock.armAlarm2(false);
clock.clearAlarm1();
clock.clearAlarm2();
pinMode (4, OUTPUT); // alla porta 4 e' collegato il led che segnala lo stato di "sveglia"
pinMode(wakePin, INPUT); //la porta 2 riceve l'allarme da DS3232 - pin 2 for alarm from ds3232
attachInterrupt(0, svegliatoria, FALLING); // attiva l'interrupts - enable interrupts
clock.setDateTime(2016, 9, 30, 11, 30, 00); // inizializza l'orologio con data ed ora prefissata,
// utile al solo fine del test. In caso di uso effettivo l'orologio deve essere inizializzato
// una sola volta, con data ed ora effettiva
// initializes ds3231 with a dummy date and hour.
}
//
//
void loop()
{
  clock.setAlarm1(0, 0, 0, 10, DS3231_MATCH_S); // attiva la sveglia al 10 secondo di ogni minuto -
  // set alarm on the tenth second of every minute
  delay (100);
  dormiora(); // richiama la routine di addormentamento - call sleep routine
  //
  // **** punto in cui il sistema rientra nella sezione di loop dopo il risveglio
  // ***** Arduino return in loop section here, after woke
  //
  digitalWrite (4, HIGH); // segnala lo stato di "sveglia" - woke status signal
  delay (5000); // rimane sveglio per 5 secondi - arduino work for 5 seconds
}
```