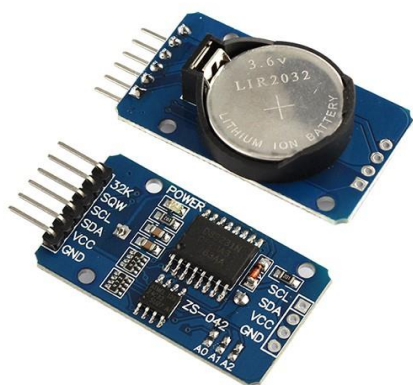


## TUTORIAL PER UN CORRETTO USO DEL MODULO REAL TIME CLOCK DS 3231

L' DS3231 è un modulo orologio in tempo reale (R.T.C.) a basso costo, con interfaccia di comunicazione I2C estremamente preciso con "a bordo" un oscillatore integrato compensato in temperatura a cristallo di quarzo (TCXO). Il dispositivo incorpora anche una batteria al Litio (sostituibile), che mantiene aggiornato l'orologio anche in assenza della alimentazione principale (5V). L'oscillatore integrato migliora l'accuratezza a lungo termine del dispositivo e riduce il numero di componenti esterni. Il DS3231 è disponibile in gamme di temperature commerciali e industriali, l'RTC mantiene secondi, minuti, ore, giorno, data, mese e anno. Per i mesi con meno di 31 giorni, la data di fine viene regolata automaticamente, comprese le correzioni per l'anno bisestile. L'orologio funziona in entrambi i formati: 24 ore oppure / AM / PM per l'indicazione del formato a 12 ore; fornisce due sveglie configurabili, indirizzo e dati sono trasferiti in serie attraverso un bus bidirezionale I2C. Il dispositivo integra un sensore di temperatura digitale (on-chip), interoperabile attraverso l'interfaccia I2C, la precisione del sensore di temperatura è  $\pm 3^{\circ}\text{C}$ ; il circuito di controllo dell'alimentazione può rilevare automaticamente e gestire l'alimentazione principale e di standby. Se si verifica una interruzione dell'alimentazione principale, il dispositivo può continuare a fornire tempi precisi e temperature, detta interruzione non influenzano le prestazioni del dispositivo.



### Parametri dell'unità:

- Dimensioni: 38mm (lunghezza) \* 22mm (larghezza) \* 14mm (altezza)
- Peso: 8 gr
- Tensione di esercizio: 3,3-5 V  $\pm 0,5$  V
- Clock chip: DS3231 circuito integrato dell'orologio di alta precisione
- Precisione Orologio: 0-40  $^{\circ}\text{C}$  gamma, il 2 ppm accuratezza, l'errore è stato di circa 1 minuto
- Sveglie programmabili n° 2
- Uscita onda quadra programmabile
- Tempo reale del generatore di clock secondi, minuti, ore, giorno, data, mese e anno valido fino al 2100 bisestile compensazione anno
- Sensore di temperatura del circuito integrato è dotato di una precisione di  $\pm 3^{\circ}\text{C}$
- Chip di memoria: AT24C32 (32Kb)
- Interfaccia bus I2C ( velocità di trasmissione massima di 400KHz con tensione di lavora pari a 5V)
- Batteria al litio CR 2032 (o LIR 2032) per assicurare che il sistema dopo un'interruzione di corrente continui a tenere aggiornato tempo e data.
- Precisione:  $\pm 1$  sec/mese  $\pm 63$  secondi/anno

### Pin da utilizzare

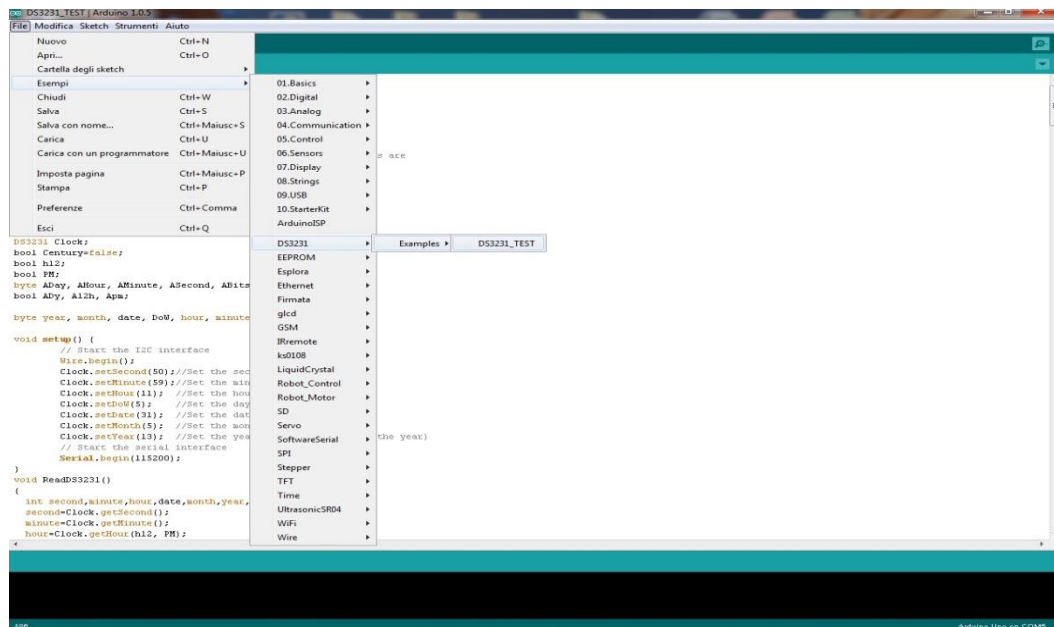
SCL  $\rightarrow$  A5 (for Arduino old version)

SDA  $\rightarrow$  A4 (for Arduino old version)

VCC  $\rightarrow$  5V

GND  $\rightarrow$  GND

Una volta collegati correttamente tutti i pin possiamo scaricare la libreria DS 3231 di Eiric Ayars sul nostro IDE Arduino dal seguente link: <http://m5.img.dxcn.com/CDDriver/CD/sku.222910.rar> e caricare il seguente codice di esempio direttamente dalla sezione esempi dell'IDE Arduino .



Leggeremo i dati direttamente sul serial monitor. Ovviamente dovremo impostare manualmente data e ora agendo sul codice. Purtroppo il modulo non contempla l'ora legale quindi, dovremo aggiornarla quando c'è il cambio (a marzo e a ottobre). Ecco il codice di Eric Ayars:

```
/*
DS3231_test.pde
Eric Ayars
4/11
```

Test/demo of read routines for a DS3231 RTC.

Turn on the serial monitor after loading this to check if things are working as they should.

```
*/
```

```
#include <DS3231.h>
#include <Wire.h>
```

```
DS3231 Clock;
bool Century=false;
bool h12;
bool PM;
byte ADay, AHour, AMinute, ASecond, ABits;
bool ADy, A12h, Apm;
```

```
byte year, month, date, DoW, hour, minute, second;
```

```

void setup() {
    // Start the I2C interface
    Wire.begin();
    Clock.setSecond(50); //Set the second
    Clock.setMinute(59); //Set the minute
    Clock.setHour(11); //Set the hour
    Clock.setDoW(5); //Set the day of the week
    Clock.setDate(31); //Set the date of the month
    Clock.setMonth(5); //Set the month of the year
    Clock.setYear(16); //Set the year (Last two digits of the year)
    // Start the serial interface
    Serial.begin(115200);
}
void ReadDS3231()
{
    int second,minute,hour,date,month,year,temperature;
    second=Clock.getSecond();
    minute=Clock.getMinute();
    hour=Clock.getHour(h12, PM);
    date=Clock.getDate();
    month=Clock.getMonth(Century);
    year=Clock.getYear();

    temperature=Clock.getTemperature();

    Serial.print("20");
    Serial.print(year,DEC);
    Serial.print('-');
    Serial.print(month,DEC);
    Serial.print('-');
    Serial.print(date,DEC);
    Serial.print(' ');
    Serial.print(hour,DEC);
    Serial.print(':');
    Serial.print(minute,DEC);
    Serial.print(':');
    Serial.print(second,DEC);
    Serial.print("\n");
    Serial.print("Temperature=");
    Serial.print(temperature);
    Serial.print("\n");
}
void loop() {ReadDS3231();delay(1000);
    // send what's going on to the serial monitor.
    // Start with the year
    /* Serial.print("2");
    if (Century) { // Won't need this for 89 years.
        Serial.print("1");
    } else {
        Serial.print("0");
    }
    Serial.print(Clock.getYear(), DEC);
    Serial.print('-');

```

```

// then the month
Serial.print(Clock.getMonth(Century), DEC);
Serial.print('-');
// then the date
Serial.print(Clock.getDate(), DEC);
Serial.print(' ');
/* and the day of the week
Serial.print(Clock.getDoW(), DEC);
Serial.print(' ');
// Finally the hour, minute, and second
Serial.print(Clock.getHour(h12, PM), DEC);
Serial.print(':');
Serial.print(Clock.getMinute(), DEC);
Serial.print(':');
Serial.print(Clock.getSecond(), DEC);
// Add AM/PM indicator
if (h12) {
    if (PM) {
        Serial.print(" PM ");
    } else {
        Serial.print(" AM ");
    }
} else {
    Serial.print(" 24h ");
}
// Display the temperature
Serial.print("T=");
Serial.print(Clock.getTemperature(), 2);
// Tell whether the time is (likely to be) valid
if (Clock.oscillatorCheck()) {
    Serial.print(" O+");
} else {
    Serial.print(" O-");
}
*/
// Indicate whether an alarm went off
/*if (Clock.checkIfAlarm(1)) {
    Serial.print(" A1!");
}
if (Clock.checkIfAlarm(2)) {
    Serial.print(" A2!");
}
*/
// New line on display
//Serial.print('\n');
// delay(1000);
// Display Alarm 1 information
/*
Serial.print("Alarm 1: ");
Clock.getA1Time(ADay, AHour, AMinute, ASecond, ABits, ADy, A12h, Apm);
Serial.print(ADay, DEC);
if (ADy) {
    Serial.print(" DoW");
} else {
    Serial.print(" Date");
}
*/

```

```

Serial.print(' ');
Serial.print(AHour, DEC);
Serial.print(' ');
Serial.print(AMinute, DEC);
Serial.print(' ');
Serial.print(ASecond, DEC);
Serial.print(' ');
if (A12h) {
    if (Apm) {
        Serial.print('pm ');
    } else {
        Serial.print('am ');
    }
}
if (Clock.checkAlarmEnabled(1)) {
    Serial.print("enabled");
}
Serial.print('\n');
// Display Alarm 2 information
Serial.print("Alarm 2: ");
Clock.getA2Time(ADay, AHour, AMinute, ABits, ADy, A12h, Apm);
Serial.print(ADay, DEC);
if (ADy) {
    Serial.print(" DoW");
} else {
    Serial.print(" Date");
}
Serial.print(' ');
Serial.print(AHour, DEC);
Serial.print(' ');
Serial.print(AMinute, DEC);
Serial.print(' ');
if (A12h) {
    if (Apm) {
        Serial.print('pm');
    } else {
        Serial.print('am');
    }
}
if (Clock.checkAlarmEnabled(2)) {
    Serial.print("enabled");
}*/
/* display alarm bits
Serial.print('\n');
Serial.print('Alarm bits: ');
Serial.print(ABits, DEC);
*/
/*
Serial.print('\n');
Serial.print('\n');
delay(1000);

// Display the time once more as a test of the getTime() function

```

```
    Clock.getTime(year, month, date, DoW, hour, minute, second);

    Serial.print(year, DEC);
    Serial.print("/");
        Serial.print(month, DEC);
    Serial.print("/");
        Serial.print(date, DEC);
    Serial.print("day of the week :");
        Serial.println(DoW, DEC);
        Serial.print(hour, DEC);
    Serial.print(":");
        Serial.print(minute, DEC);
    Serial.print(":");
        Serial.println(second, DEC);*/
}
```