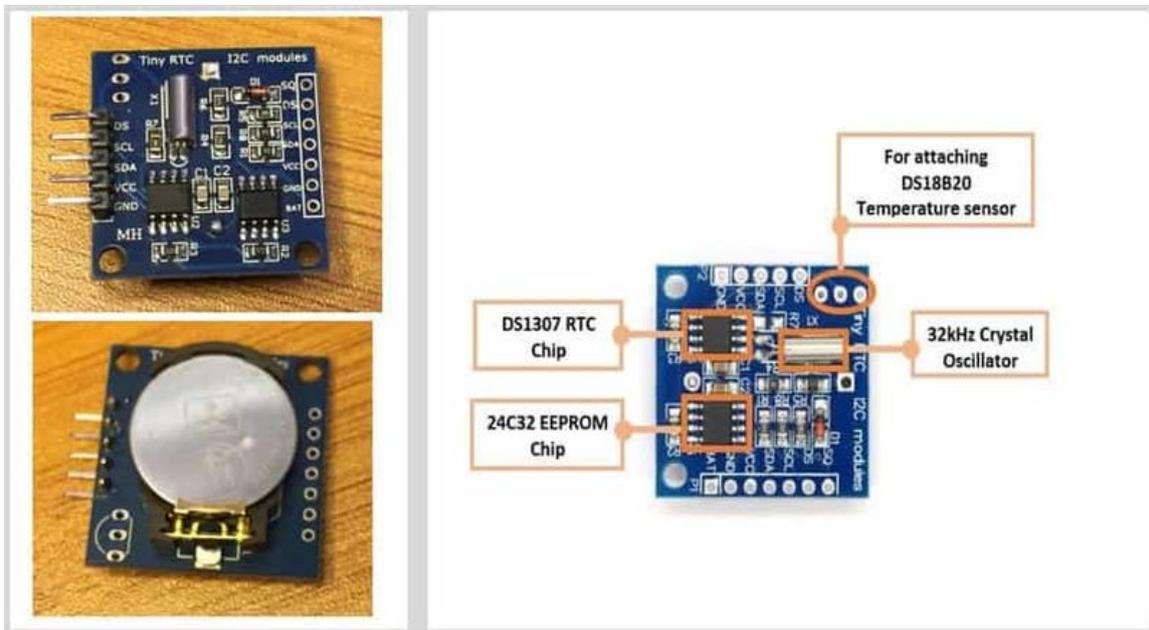


OROLOGIO CON DS 1307 RTC TINY I2C & LCD I2C 16X2

Questo progetto, permette la visualizzazione della data e dell'ora su un display LCD 16x2 I2C; si basa sull'economico modulo DS1307 RTC (Tiny I2C). Questo modulo è dotato di pin I/O su entrambi i lati. La configurazione di questi pin è simile e si può utilizzare il lato che si preferisce a seconda dell'applicazione.



La piedinatura del **DS1307 RTC** è la seguente:

- **DS** -Device Select e viene utilizzato quando un sensore di temperatura è collegato all' RTC per emettere le letture della temperatura.
- **SCL** – Ingresso orologio seriale per l'interfaccia I2C
- **SDA** – Ingresso/uscita dati seriali per l'interfaccia seriale I2C.
- **VCC** – Alimentazione da 3,3 V a 5 V.
- **GND** – Massa circuito.
- **BAT** – Alimentazione di backup 3V per tenere traccia del tempo quando l'alimentazione principale viene interrotta.
- **SQW** – Per emettere frequenze a onda quadra di 1Hz, 4kHz, 8kHz o 32kHz.

C'è anche la predisposizione per un sensore di temperatura digitale DS18B20 e, con l'aggiunta di opportuno codice, sarà possibile monitorare anche la temperatura ambientale. Nei moduli RTC è sempre presente un quarzo che preserva la precisione e l'accuratezza della misura del tempo. I componenti principali di questo modulo RTC includono:

- **Chip DS1307 RTC** : questo è il circuito integrato principale per il dispositivo e tiene traccia dell'ora sia nel formato 12 ore che 24 ore, data, giorni della settimana, mesi e anni con una precisione di un secondo. Esegue anche la regolazione automatica del numero di giorni in un mese e degli anni bisestili.
- **Oscillatore a cristallo di quarzo** da 32 kHz per mantenere il tempo.
- **Chip EEPROM 24C32** con 32 byte di spazio di archiviazione. Questo circuito integrato utilizza anche l'interfaccia I2C e condivide il **bus I2C** con il circuito integrato **DS1307 RTC**.
- È inoltre previsto il collegamento di un sensore di temperatura **DS18B20**.

Questo modulo **DS1307 RTC** ha una batteria a bottone al litio **CR2032** da **3 V** collegata a un circuito di rilevamento dell'alimentazione integrato che rileva lo spegnimento dell'alimentazione principale e passa automaticamente ***all'alimentazione di backup per non interrompere il processo di cronometraggio.***

Poiché l'oscillatore di cristallo è esterno, la sua frequenza di oscillazione è influenzata dalla temperatura esterna che a lungo andare comporta la perdita di alcuni minuti nel processo di cronometraggio. Questo è uno dei principali svantaggi dell'utilizzo del **DS1307 RTC** in quanto, potrebbe essere necessario reimpostare il tempo dopo un certo periodo, per eseguire la correzione dei minuti persi.

Nella sezione di configurazione, inizializziamo il modulo RTC utilizzando il metodo **begin ()** e il metodo **isrunning ()** servono per preparare il **chip DS1307** per azzerare l'ora. La funzione **adjust ()** serve per impostare la data e l'ora. Per impostare la data e l'ora dell'RTC, in modo che corrispondano all'ora in cui è stato compilato, nello sketch del codice, utilizziamo la dichiarazione seguente:

```
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
```

La data e l'orario impostate utilizzando il metodo sopra, corrisponderanno a quelle sul tuo computer.

Per poter impostare una data e un'ora specifiche puoi usare:

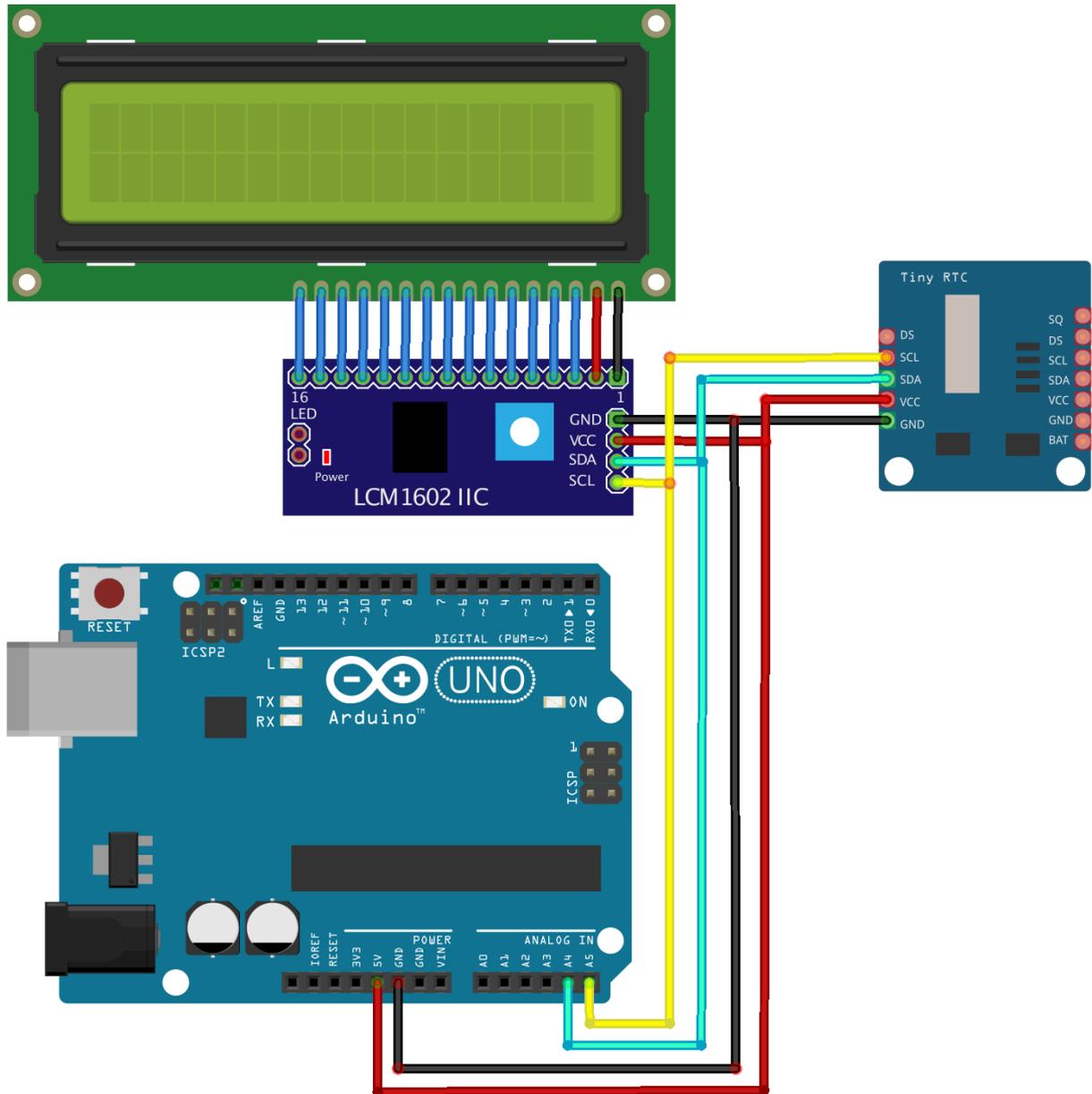
```
rtc.adjust(DateTime(YYYY, M, D, H, M, s));
```

Ad esempio, per impostare il 23 Marzo 2022 alle 9:15 devi usare la sintassi seguente:

```
rtc.adjust(DateTime(2022, 3, 23, 9, 15, 0));
```

N.B. *Dopo aver impostato l'ora desiderata, queste funzioni devono essere commentate fuori dallo sketch del codice quindi rimettere le // per evitare il ripristino ripetitivo dell'ora e della data ogni volta che si desidera caricare un nuovo programma.*

Schema di montaggio:



fritzing

Codice di esempio:

```

/*Simple Digital Clock con Arduino
(c) Marzo 2022 Prof. Francesco Benignetti
con auto update quando programmato da IDE
-Tiny I2C Real Time Clock
-LCD16x2 with I2C interface
-Arduino
*/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "RTCLib.h"
RTC_DS1307 rtc;
LiquidCrystal_I2C lcd(0x27, 16, 2);
char daysOfTheWeek[7][12] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
void setup ()
{
  Serial.begin(9600);
  lcd.begin (); // initialize the lcd
  lcd.backlight();//To Power ON the back light
  if (! rtc.begin())
  {
    lcd.print("NON TROVO RTC");
    while (1);
  }
  if (! rtc.isrunning())
  {
    lcd.print("RTC is NOT running!");
  }
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); //auto update from computer time
  /*ricordarsi una volta programmato l'orario di mettere a commento(//)questa
  linea di codice. In maniera da evitare l'autoreset dell'impostazione orario*/
  //rtc.adjust(DateTime(2022, 3, 23, 9, 5, 0)); // to set the time manually
}
void loop ()
{
  DateTime now = rtc.now();
  lcd.setCursor(0, 1);
  lcd.print("TIME:");
  lcd.print(" ");
  lcd.print(now.hour());
  lcd.print(':');
  lcd.print(now.minute());
  lcd.print(':');
  lcd.print(now.second());
  lcd.print(" ");

  lcd.setCursor(0, 0);
  lcd.print("DATE:");
  lcd.print(" ");
  //lcd.print(daysOfTheWeek[now.dayOfTheWeek()]);
  //lcd.print(" ");
  lcd.print(now.day());
  lcd.print('/');
  lcd.print(now.month());
  lcd.print('/');
  lcd.print(now.year());
  lcd.print(" ");
}

```