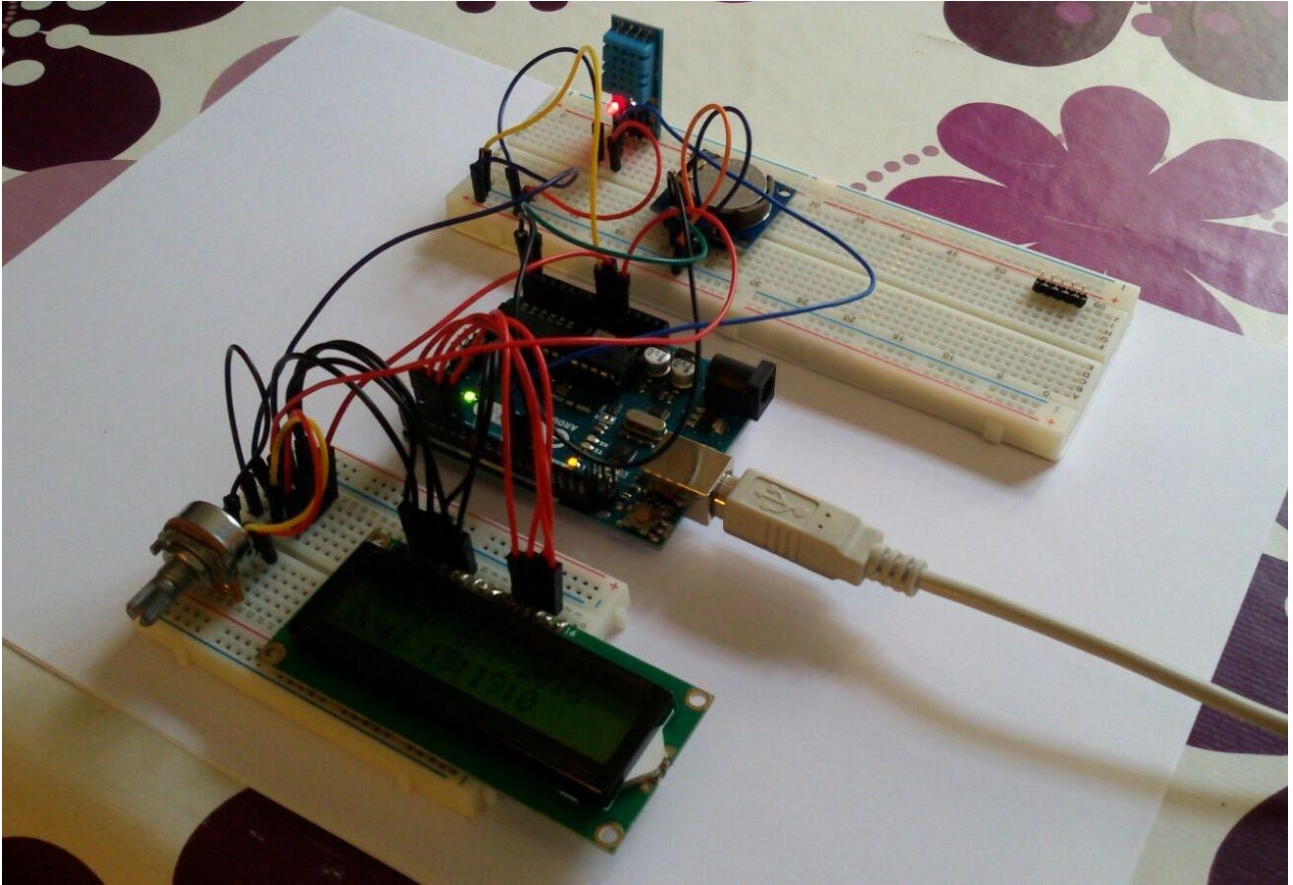


Relazione progetto Arduino

A cura di Francesca Cella e Andrea Marras



Il progetto che abbiamo realizzato è un dispositivo in grado di leggere la temperatura, l'umidità presente nell'aria e di tenere il conto della data e dell'ora correnti, alternando la stampa a video delle due informazioni tramite un ciclo che si aggiorna ogni lasso di tempo definito nel programma.

Il dispositivo si avvale della tecnologia Arduino UNO, famosa per la sua facilità di programmazione e per essere alla portata di chiunque abbia una minima base di conoscenza dei suoi linguaggi, pur non avendo praticamente basi di elettronica.

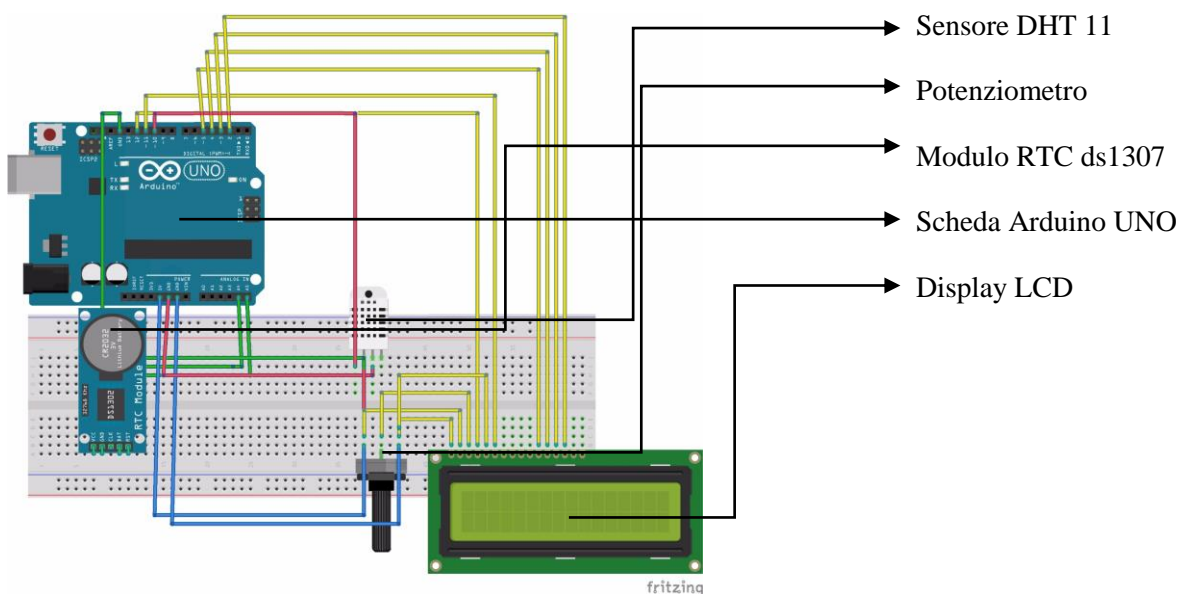
Per la realizzazione del progetto abbiamo utilizzato i seguenti componenti:

- Scheda Arduino UNO
- Display LCD di dimensioni 16x9 mm, in dotazione con la scheda
- Potenziometro
- Sensore di Temperatura e umidità DHT 11
- Modulo RTC ds1307 per la data e l'ora
- Jumper

Fase 1: Progettazione

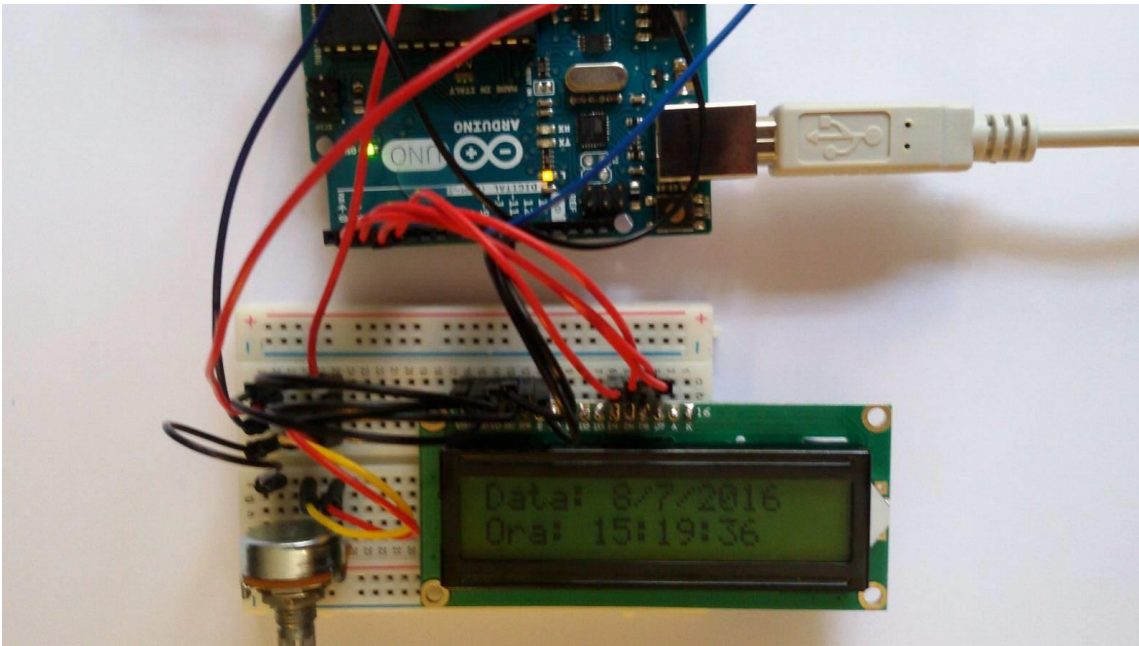
Una volta creata la nostra idea, per la progettazione ci siamo ispirati a vari progetti e dispositivi già ampiamente presenti nel web. Dopo le prime ricerche e documentazioni, grazie al programma gratuito Fritzing, abbiamo creato la base da cui partire per la realizzazione del dispositivo. Il programma, purtroppo, non era dotato di due dei componenti che avevamo scelto per il dispositivo (in particolare il DHT 11 ed il modulo RTC ds1307), tuttavia non è stato un grosso problema, poiché, essendo Arduino e tutti i suoi dispositivi compatibili, molto facili ed intuitivi da utilizzare, siamo riusciti comunque a creare tutti i collegamenti. Per l'alimentazione abbiamo scelto di utilizzare un cavo per collegare la scheda alla porta USB del computer (5V), in questo modo avremmo potuto alimentare il dispositivo e nel frattempo programmarlo, in modo da vedere immediatamente i progressi col codice.

Ecco un'immagine del progetto realizzato con Fritzing:

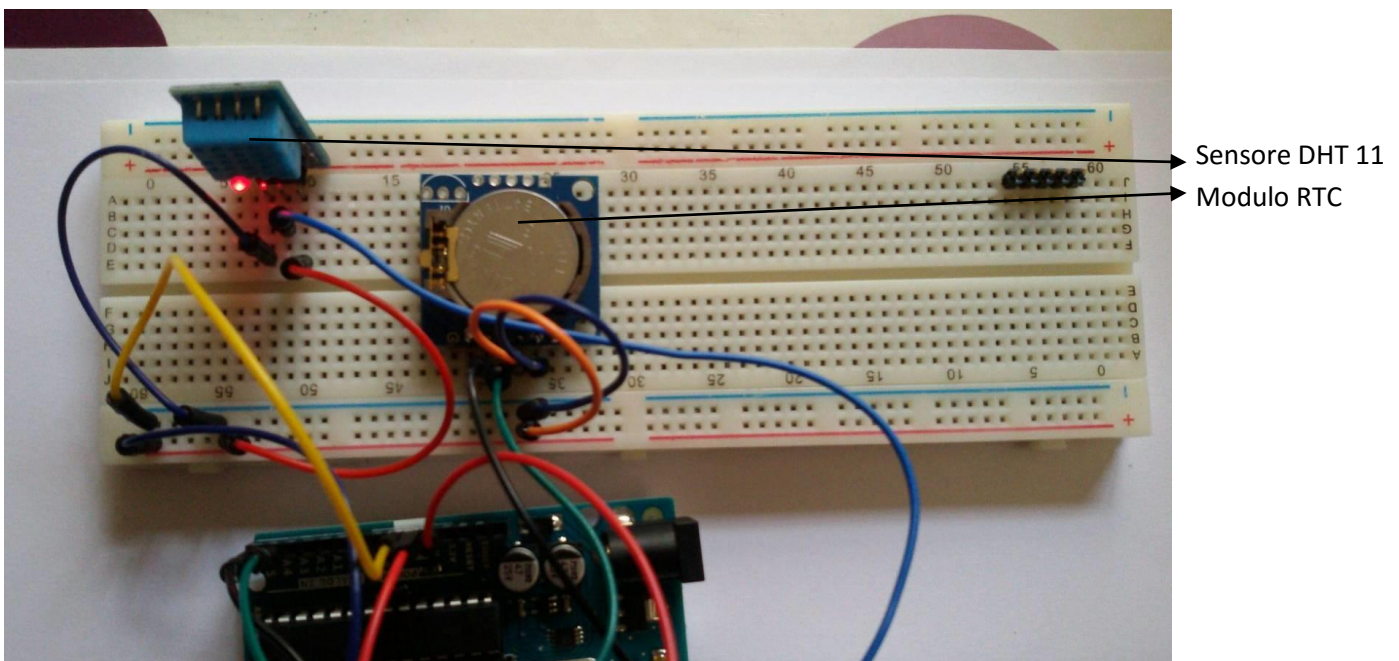


Fase 2: Assemblaggio

Una volta creato il progetto virtuale, abbiamo iniziato a collegare fisicamente i vari componenti, iniziando dal Display LCD, in modo da poter stampare in seguito qualsiasi informazione e verificare che tutti i collegamenti futuri sarebbero stati corretti. Il display in dotazione può stampare due righe di caratteri, perfette per il tipo di informazioni che vogliamo visualizzare. Con l'uso del potenziometro è possibile regolare luminosità e contrasto, evitando inoltre che lo schermo possa bruciarsi con un eccessivo voltaggio.



Una volta appurato che lo schermo fosse correttamente connesso e funzionante, abbiamo collegato i componenti restanti:



-Dal DHT 11 abbiamo un **filo rosso** collegato alla breadboard, collegata a sua volta con l'uscita 5V della scheda Arduino; un **filo blu** collegato alla breadboard che è collegata tramite il **filo giallo** all'uscita GND della scheda Arduino; un **filo azzurro** collegato al rispettivo pin di Arduino, che serve per trasportare i dati.

- L'RTC ds1307 è collegato con un **filo arancione** e un **filo blu**, rispettivamente all'uscita dei 5V e GND di Arduino, mentre il **filo verde** e il **filo nero** portano le informazioni ai rispettivi pin di entrata analogica.

Fase 3: Codice

La programmazione è avvenuta di pari passo con il collegamento di tutte le componenti hardware del dispositivo, questo per evitare di dover ricominciare da capo nel caso in cui avessimo sbagliato qualche collegamento. Possiamo dividere il corpo del programma che abbiamo realizzato in tre parti principali:

1) Inizializzazione della scheda Arduino, con inclusione e dichiarazione delle librerie necessarie al funzionamento dei componenti fisici e delle entrate ad essi associate;

```
#include <RTCLib.h>
#include <dht11.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#define DHT11_PIN 10
#define RTC_PIN 6

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
dht11 DHT;
RTC_DS1307 RTC;
char buffer1[20];
char buffer2[20];
```

2) Controllo all'accensione di tutti i vari componenti;

```
void setup() {
  lcd.begin(16, 2);
  Wire.begin();
  RTC.begin();
  if (! RTC.isrunning()) {
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
}
```

3) Ciclo per la stampa delle informazioni.

```
void loop() {
  int chk=DHT.read(DHT11_PIN);
  //Stampa a video temperatura e umidità
  lcd.setCursor(0, 0);
  lcd.print("Umid: ");
  lcd.print(DHT.humidity,1);
  lcd.print("%");
  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.print(DHT.temperature,1);
  lcd.print(" gradi");
  delay(3000);
  //pulizia "artigianale" del display per evitare sovrapposizioni
  lcd.setCursor(0,0);
  lcd.print(" ");
  lcd.setCursor(0,1);
  lcd.print(" ");
  //Stampa della data e dell'ora
  RTC_DS1307 RTC;
  DateTime now = RTC.now();
```



```
lcd.setCursor(0,0);  
lcd.print("Data: ");  
sprintf(buffer1, "%d/%d/%d",now.day(),now.month(),now.year());  
lcd.print(buffer1);  
lcd.setCursor(0,1);  
lcd.print("Ora: ");  
sprintf(buffer2, "%d:%d:%d", now.hour(), now.minute(), now.second());  
lcd.print(buffer2);  
delay(3000);
```

```
lcd.setCursor(0,0);  
lcd.print(" ");  
lcd.setCursor(0,1);  
lcd.print(" ");
```

```
}
```