|  |  |
| --- | --- |
|  | Simulazione impianto termico |
|  |  |
| 03/04/2017 | Realizzazione di un automa con Arduino |
|  | *Con l’ausilio di Arduino e di alcuni componenti elettronici basilari è possibile simulare il funzionamento di una caldaia, regolata da un sensore di temperatura LM35.* |

Simulazione impianto termico

Flow - Chart

Temperatura < 18

Temperatura (O)

Valore\_Sensore (I)

Voltaggio = Valore\_Sensore / 1024 \* 5

Temperatura = (Voltaggio - 0,5) \* 100

Vero

Falso

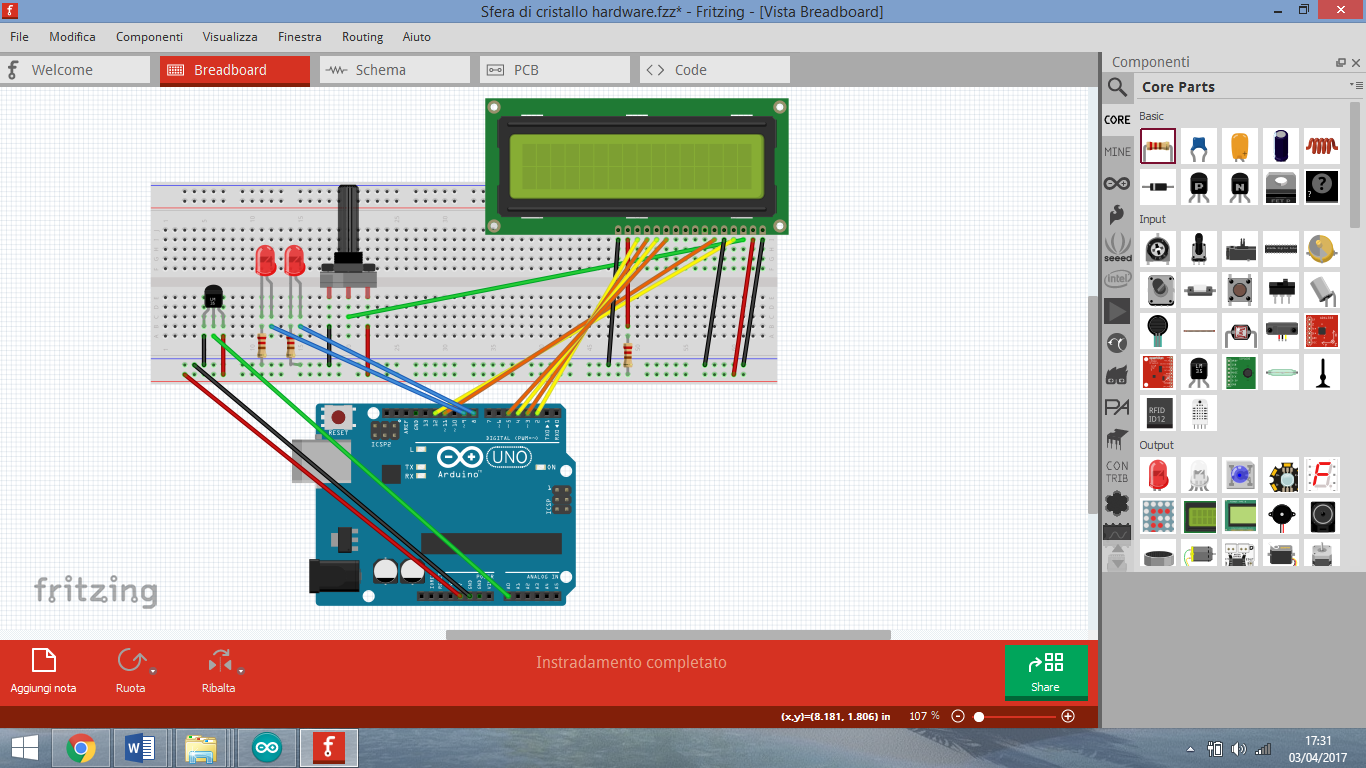
Caldaia ON (O)

Caldaia OFF (O)

Tabella delle variabili e delle costanti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Nome* | *Tipo* | *Variabile o Costante* | *Utilizzo* | *Descrizione* |
| *Valore\_Sensore* | *Reale* | *Variabile* | *Input* | *Valore ADC letto dal sensore.* |
| *Voltaggio* | *Reale* | *Variabile* | *Lavoro* | *Tensione in uscita dal sensore.* |
| *Temperatura* | *Reale* | *Variabile* | *Output* | *Temperatura rilevata dal sensore.* |

Hardware di riferimento



Analisi complessiva

Questo progetto simula il funzionamento di un impianto termico automatizzato. Il funzionamento è regolato da un sensore di temperatura che rileva la temperatura esterna. Questo valore viene confrontato con il valore preimpostato dall’utente: se minore del valore standard la caldaia si accende, in caso contrario smette di funzionare.

Analisi hardware

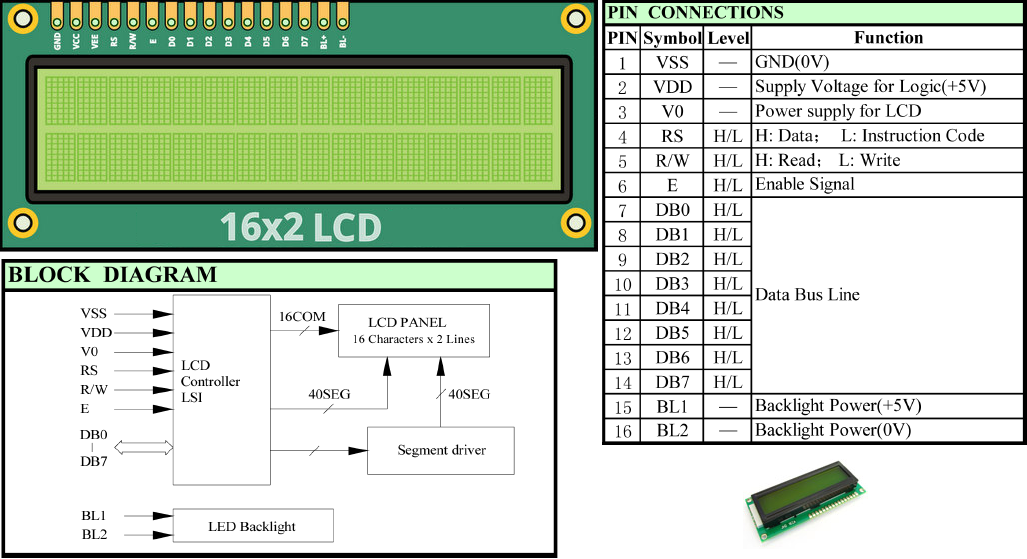
Per la realizzazione di questo automa sono necessari:

* una scheda open source Arduino UNO;
* una breadboard;
* un sensore di temperatura LM35;
* un potenziometro da 10 KΩ;
* display HD44780 LCD 162 caratteri;
* un LED rosso;
* un LED verde;
* 3 resistenze da 220Ω;
* ponticelli rigidi e flessibili.

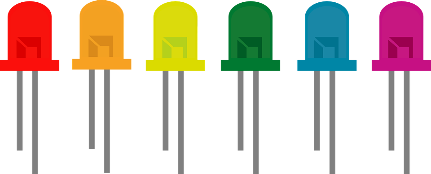
La realizzazione del circuito è abbastanza semplice.

Abbiamo collegato l’alimentazione (5V) e la massa (GND) a un lato della breadboard e dopo abbiamo iniziato a collegare i piedini del display LCD ai rispettivi pin di Arduino.

Il piedino Register Select (RS) controlla dove i caratteri compaiono sullo schermo. Il piedino read/write (R/W) mette lo schermo in modaltà lettura o scrittura (in questo caso utilizzeremo solo la modalità scrittura). L’abilitazione (E) dice all’LCD che riceverà un comando. I piedini dei dati (D0 – D7) sono usati per mandare i dati dei caratteri allo schermo (ne abbiamo usati solo 4 D4 – D7). Infine c’è una connessione per regolare il contrasto del display (abbiamo usato un potenziometro per controllarlo).

Dunque i due piedini esterni all’LCD Vss e LED- devono essere collegati a massa. Poi, abbiamo collegato il piedino R/W a massa. Questo mette lo schermo in modalità scrittura. L’alimentazione dell’LCD Vcc deve collegarsi direttamente ai 5V. Il piedino LED+ sullo schermo si collega all’alimentazione tramite una resistenza da 220Ω. In seguito abbiamo collegato il pin 2 di Arduino a LCD D7, il pin 3 a LCD D6, il pin 4 a LCD D5, il pin 5 a LCD D4. Dopo abbiamo collegato E dello schermo al pin 11 di Arduino, RS al pin 12.

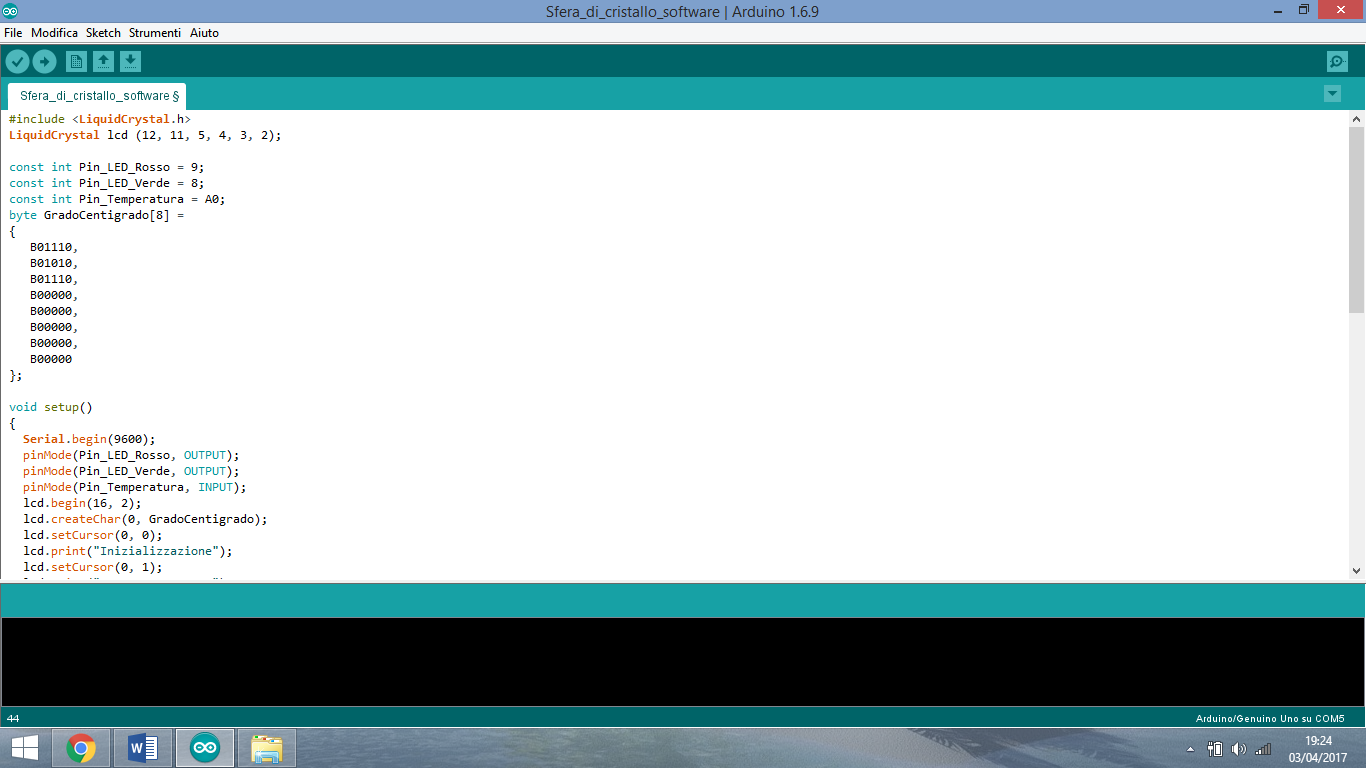
Abbiamo posizionato un potenziometro da 10 KΩ sulla breadboard, collegando i piedini esterni all’alimentazione e alla massa e quello centrale al V0 dell’LCD (per regolare il contrasto del display).



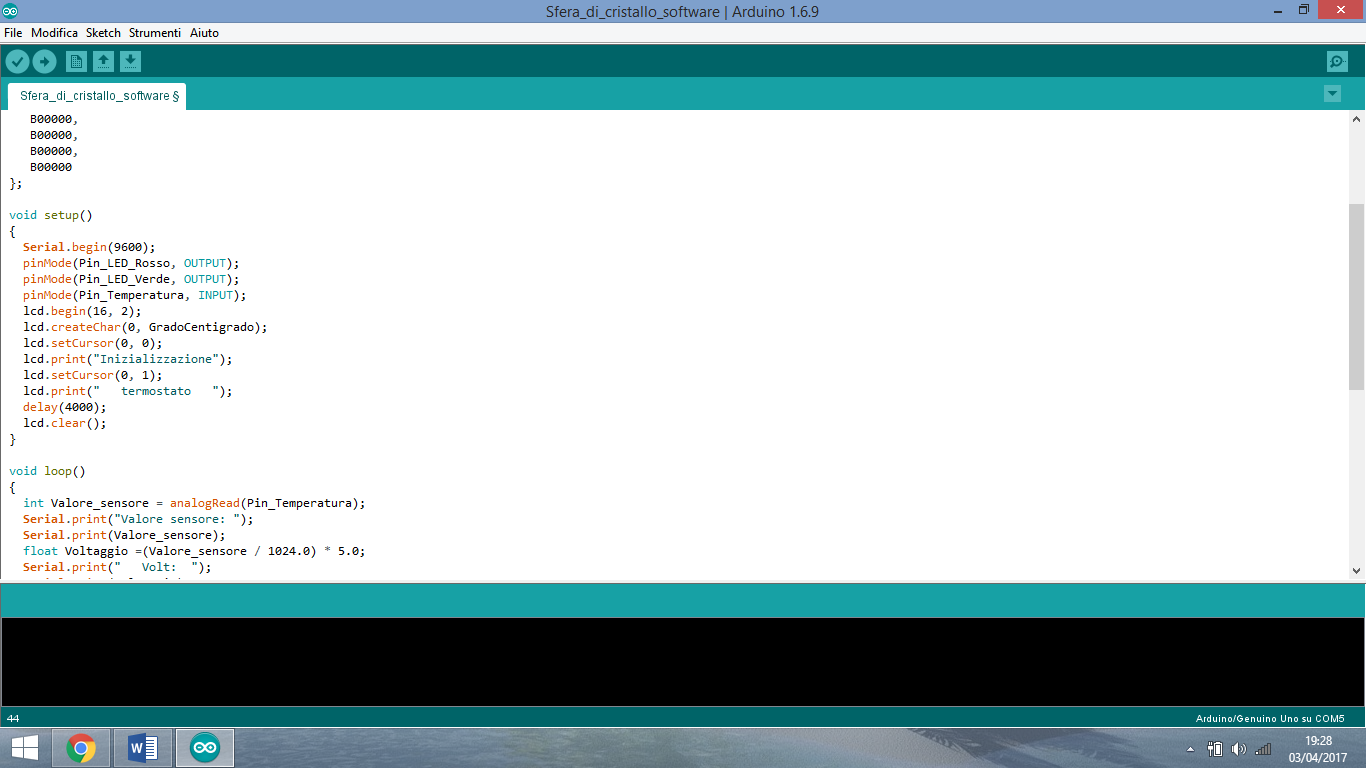
Abbiamo posizionato i LED verde e rosso sulla breadboard collegando a massa l’anodo con una resistenza da 220Ω e il catodo con i rispettivi piedini 8 e 9.

Infine abbiamo posizionato il cuore del circuito, il sensore LM35 che presenta un contenitore di tipo TO92 analogo a quello di un normale transistor, e necessita solo di un collegamento ad una porta ADC per la lettura del valore di tensione fornito dal sensore. Il piedino a sinistra (dalla parte piatta) va collegato all’alimentazione e il piedino a destra alla massa. Il piedino centrale va collegato al pin A0 di Arduino.

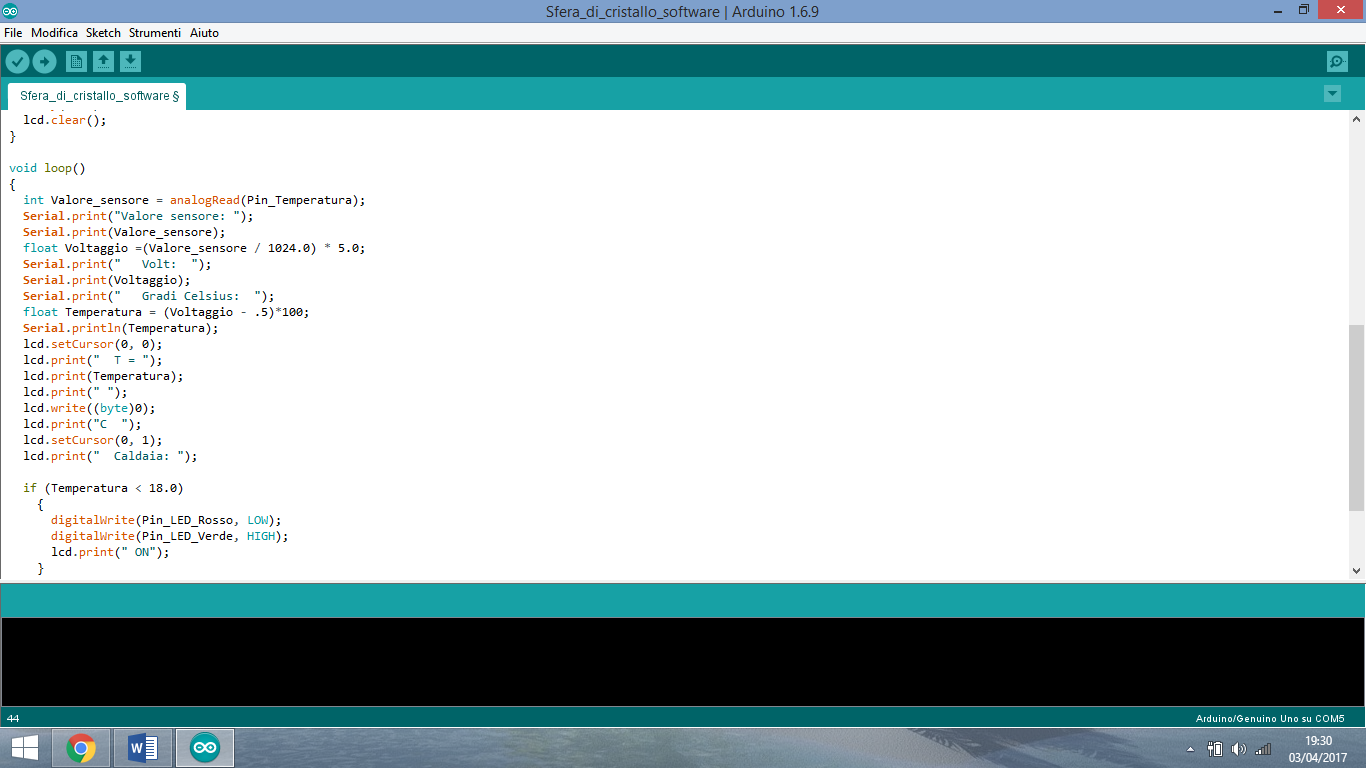
Analisi software

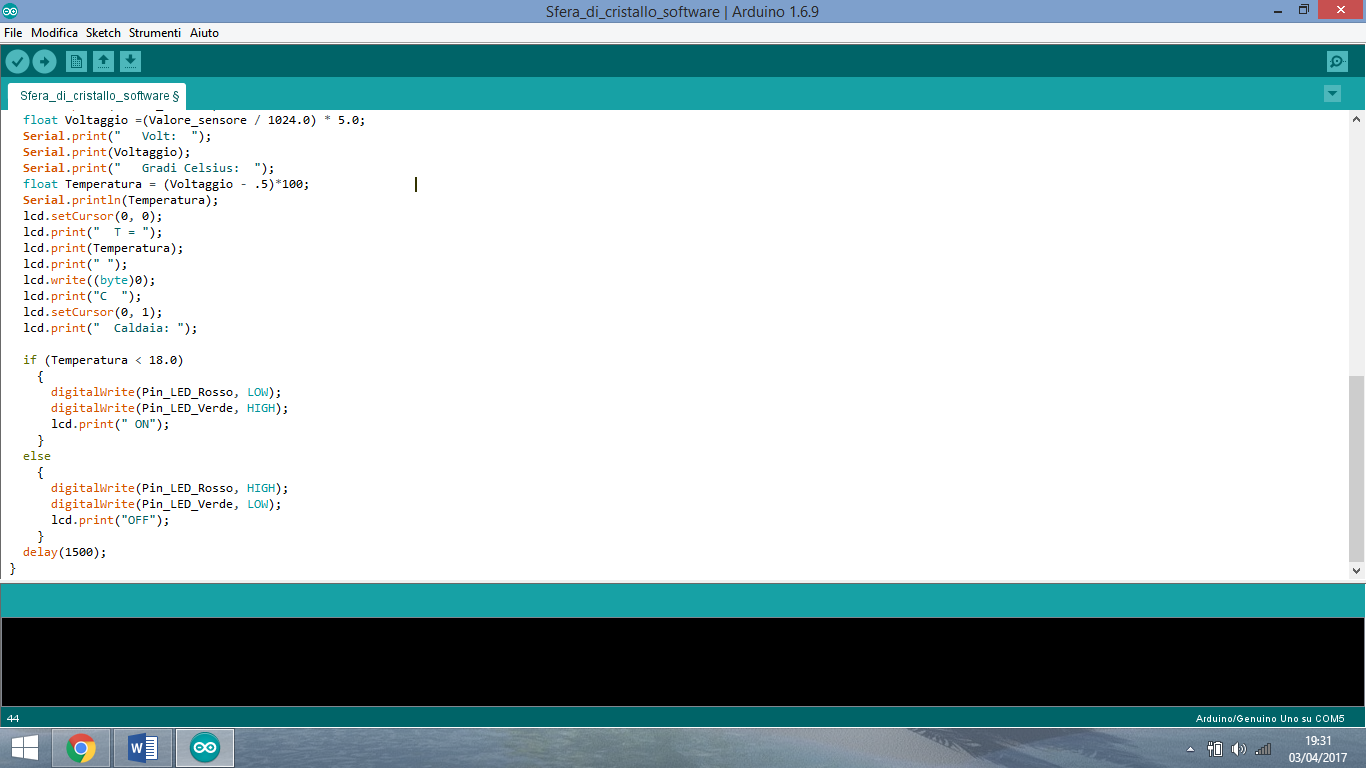
Per prima cosa è necessario importare la libreria LiquidCrystal.h, necessaria per l’importazione nel programma delle funzioni relative al display. Per avviare la libreria è necessario specificare i piedini da utilizzare con l’apposita funzione LiquidCrystal.lcd (…).

Prima della realizzazione vera e propria del programma è necessario dichiarare alcune costanti per memorizzare i piedini relativi ai LED e al sensore. Una costante è necessaria per la realizzazione del carattere ° sul display, definendo i pixel accesi e spenti: questo grazie ad un array di valori binari.

Si prosegue con il setup() che viene eseguito una sola volta soltanto all’accensione di Arduino. In questa porzione di programma è presente:

* la configurazione dei piedini digitali per renderli degli ingressi o delle uscite usando la funzione pinMode;
* l’apertura della comunicazione seriale con il computer a 9600 baud (bit al secondo);
* la definizione della larghezza dello schermo usando la funzione lcd.begin.

Segue il loop() che viene eseguito continuamente dopo che è stato eseguito il setup(). Nel loop() va inserito:

* la lettura del valore ADC del sensore, trasformandolo prima in una tensione e dopo in una temperatura;
* la scrittura sul monitor seriale del valore ADC del sensore, i volt e la temperatura corrispondenti;
* la scrittura sul display del valore della temperatura;
* segue la condizione:
  + se la temperatura è minore di 18 °C la caldaia si mette in funzione, di conseguenza sul display va scritto Caldaia ON e il rispettivo LED verde si accende;
  + Se la temperatura non è minore di 18 °C la caldaia smette di funzionare, di conseguenza sul display va scritto Caldaia OFF e il rispettivo LED rosso si accende.
* Infine è presente un ritardo (delay) di 1500 millisecondi prima della seconda lettura del sensore.