



Istituto di Istruzione Secondaria Superiore

Michelangelo Bartolo

Viale A. Moro - Pachino - tel. 0931 593596 - sris01400@istruzione.it
Dirigente Prof. Vincenzo Pappalardo - Sito a cura del Prof. S. Giannitto



Alunna:

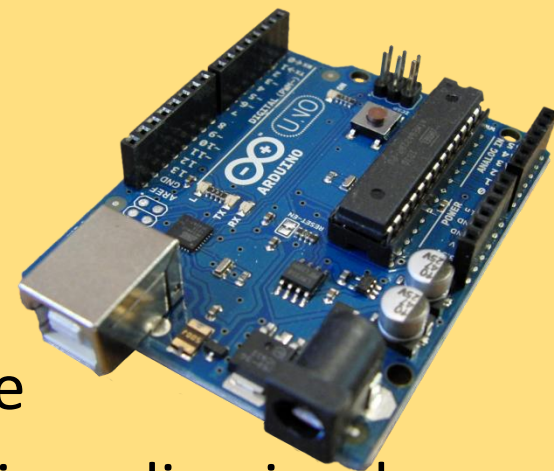
Gennuso Ylenia
classe VB
a.s. 2014/2015



T.D.P: arduino



Arduino è una piattaforma di sviluppo open-source basata su una semplice scheda I / O con un μ C e un ambiente di sviluppo che implementa hardware e software facile da usare. E' una scheda elettronica di piccole dimensioni con un microcontrollore e circuiteria di contorno, utile per creare rapidamente prototipi e per scopi didattici.



Con Arduino si possono realizzare in maniera relativamente rapida e semplice piccoli dispositivi come controllori di luci, di velocità per motori, sensori di luce, temperatura e umidità e molti altri progetti che utilizzano sensori, attuatori e comunicazione con altri dispositivi.

Il Microcontrollore possiede su un unico chip:
Una **CPU** RISC (Reduced Instruction Set Computer);
Una piccola memoria di programma (**EPROM-EEPROM**);
Una piccola memoria di lavoro **RAM** (alcuni kB);
Porte di ingresso/uscita;

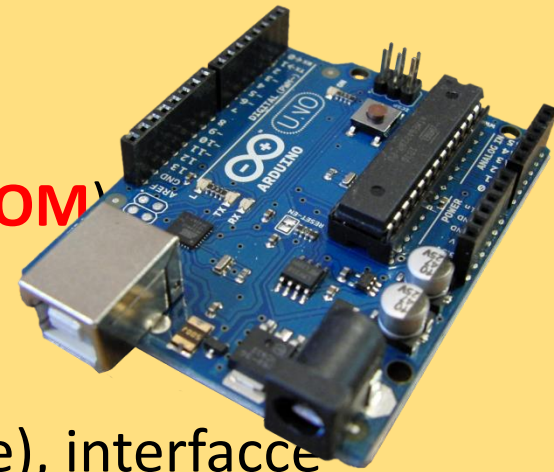
Contatori, timer, convertitore A/D;

UART (ricevitore-trasmittitore asincrono universale), interfacce di comunicazione di vari tipi.

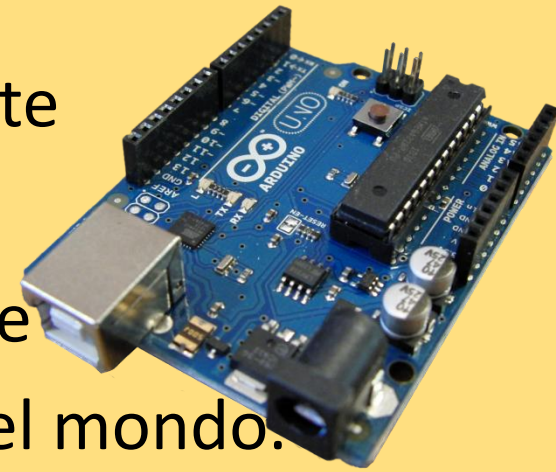
Il microcontrollore ha ingombro minimo e richiede poca potenza di alimentazione. Sono inoltre semplici istruzioni che richiedono circa il medesimo tempo di elaborazione.

Le istruzioni perciò:

- sono poche decine
- sono eseguite molto velocemente
- non serve un clock molto elevato per un efficiente funzionamento (dai 4 agli **8 MHz** per i tipi più semplici, fino a **33-50 MHz** per i tipi più evoluti).

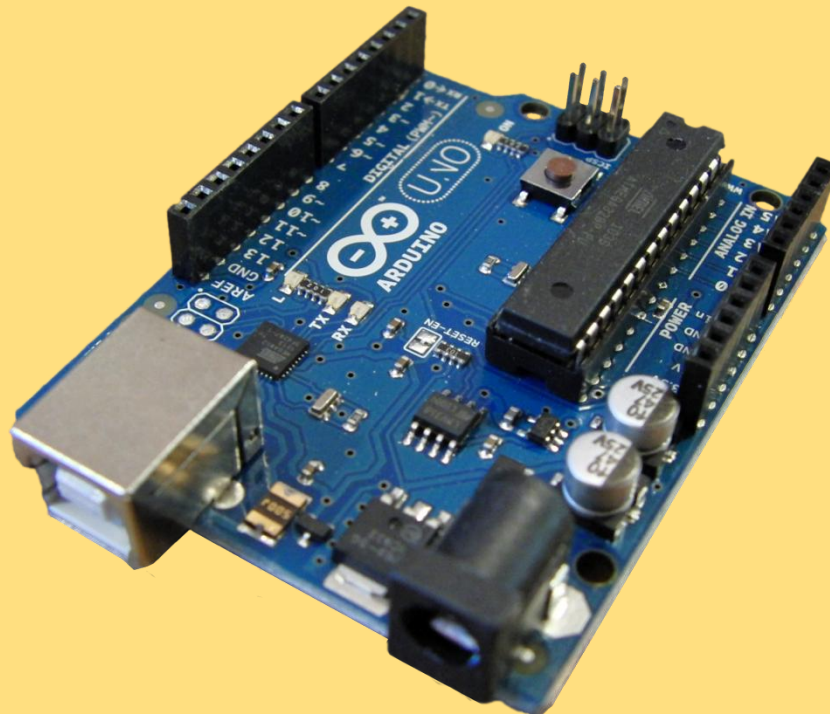


L'hardware originale Arduino è interamente realizzato in Italia dalla Smart Projects, mentre i cloni della scheda possono essere realizzati da chiunque in qualsiasi parte del mondo.



Una scheda Arduino tipica consiste in un microcontrollore a 8-bit AVR prodotto dalla Atmel, con l'aggiunta di componenti complementari per facilitarne l'incorporazione in altri circuiti. In queste schede sono usati chip della serie megaAVR - nello specifico i modelli ATmega8, ATmega168, ATmega328, ATmega1280 e ATmega2560.

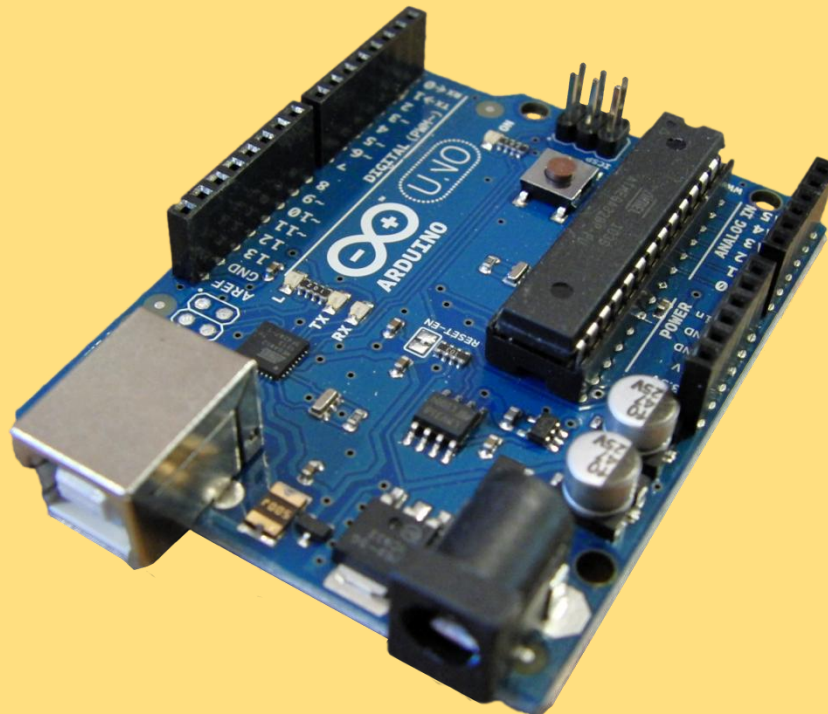
Arduino è dotato di molti dei **connettori di input/output** per i microcontrollori in uso su altri circuiti. Tutti i pin di I/O sono collocati sulla parte superiore della scheda mediante connettori femmina da 0,1". Arduino Uno offre **14 connettori per l'I/O digitale** (numerati da 0 a 13). La direzione di funzionamento, input o output, è decisa dallo sketch programmato sull'IDE.



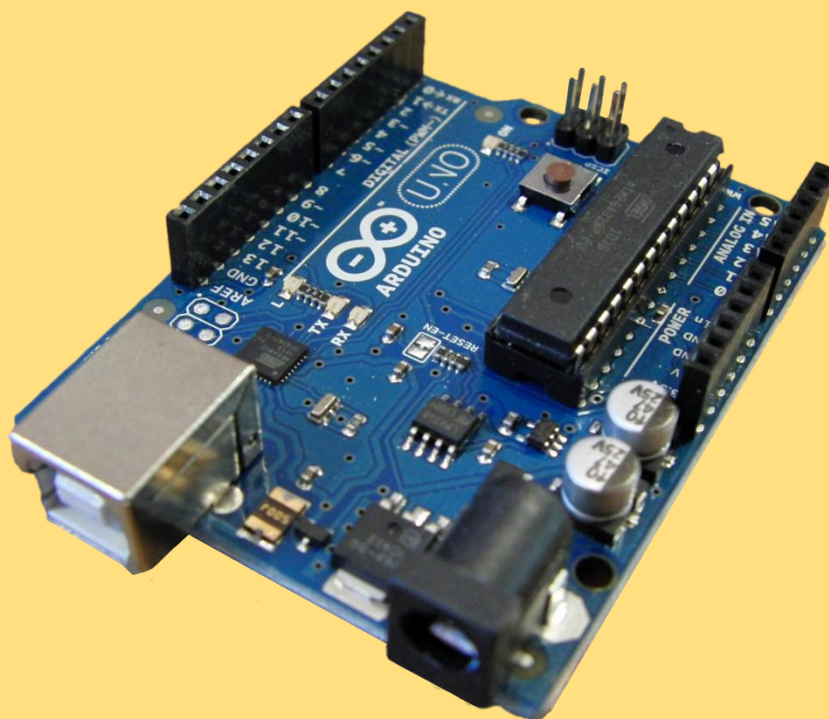
Sei dei canali I/O possono produrre segnali Pulse-width modulation (PWM). Attraverso i segnali **PWM** è possibile, ad esempio, regolare l'intensità di luminosità di un LED o la velocità di rotazione di un motorino elettrico. L'hardware di tre dei pin di I/O (9, 10 e 11) implementa la possibilità di gestirli direttamente attraverso la funzione analogWrite, che permette di controllare la PWM del segnale in uscita in maniera efficiente, senza dover eseguire linee di codice appositamente predisposte.



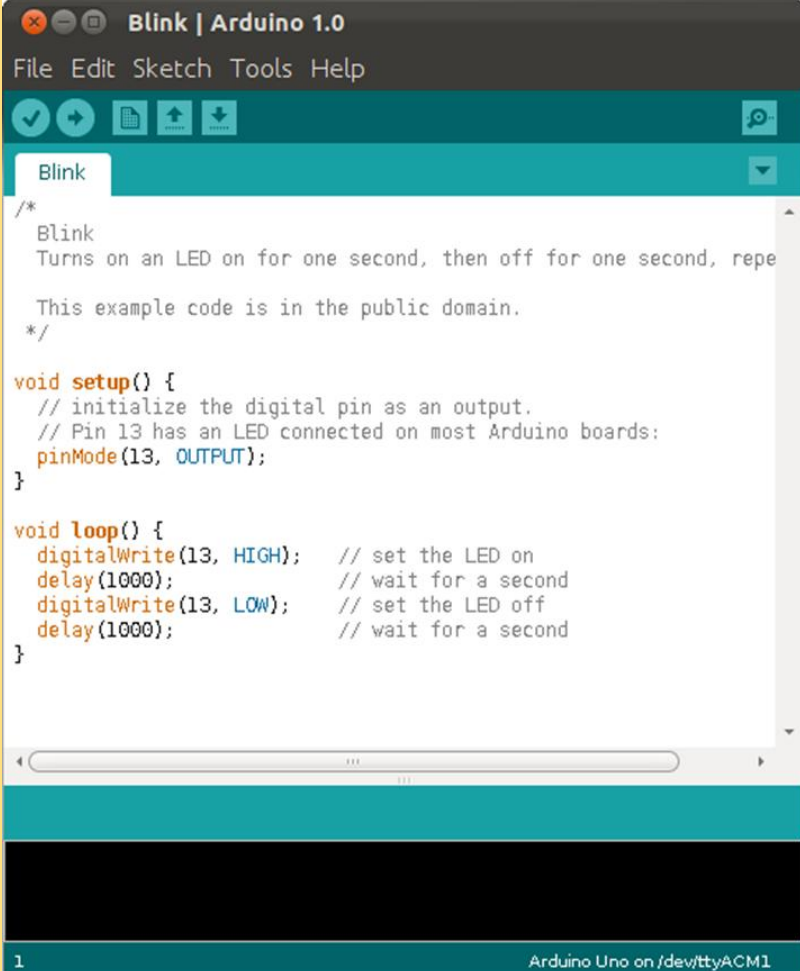
Sempre sulla Uno, sono presenti altri **6 connettori specificamente dedicati a ingressi di segnali analogici** (collegati quindi ad una ADC), cioè valori di tensione letti da sensori esterni i cui valori, fino a un massimo di 5 Volt, sono convertiti in 1024 livelli discreti (da 0 a 1023). Questi 6 connettori possono essere riprogrammati (sempre dal codice dello sketch sull'IDE) per funzionare come normali entrate/uscite digitali.



L'alimentazione della scheda può avvenire attraverso la porta USB del computer, o attraverso la maggior parte degli alimentatori **USB**, oppure attraverso un **adattatore in corrente continua a 9 volt**, con connettore cilindrico (diametro 2,1 mm e positivo centrale). I programmi di Arduino si chiamano **sketch** o bozzetti e collegando gli opportuni sensori arduino disporrà di tatto, udito, olfatto e vista ai quali si potrà aggiungere l'orientamento, il calcolo delle distanze, la vista agli infrarossi e altro ancora.



Se un progetto richiede molti collegamenti che superano le porte di I/O disponibili in una sola scheda, allora si potrà intraprendere la strada a più schede Arduino UNO che colloquiano tra di loro e si dividono i compiti. Arduino dispone di soli 32KB e in questo spazio dovrà starci tutto il nostro programma. L'ambiente di sviluppo integrato (IDE) di Arduino è un'applicazione multiplatforma scritta in Java, ed è derivata dall'IDE creato per il linguaggio di programmazione Processing e per il progetto Wiring.

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, running, uploading, and downloading. The main editor area shows the code for the "Blink" sketch. The code includes a comment block describing the sketch's function and a public domain notice. It defines a setup function to initialize pin 13 as an output and a loop function that toggles the LED on and off with 1000ms delays.

```
Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeating.
 *
 * This example code is in the public domain.
 */
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000); // wait for a second
}
1 Arduino Uno on /dev/ttyACM1
```

Il programma inizia con la dichiarazione delle variabili.

Seguono due blocchi distinti:

-Void setup(); contenente le istruzioni di inizializzazione che impostano la scheda settando i pin di input e di output

-Void loop(); contenete il programma principale , che si ripete ciclicamente

- /* e */ delimitano i commenti che possono occupare più righe

// segna l'inizio di un commento che però deve essere contenuto in una sola riga

Ogni istruzione deve terminare con il **;**

IL PROGETTO

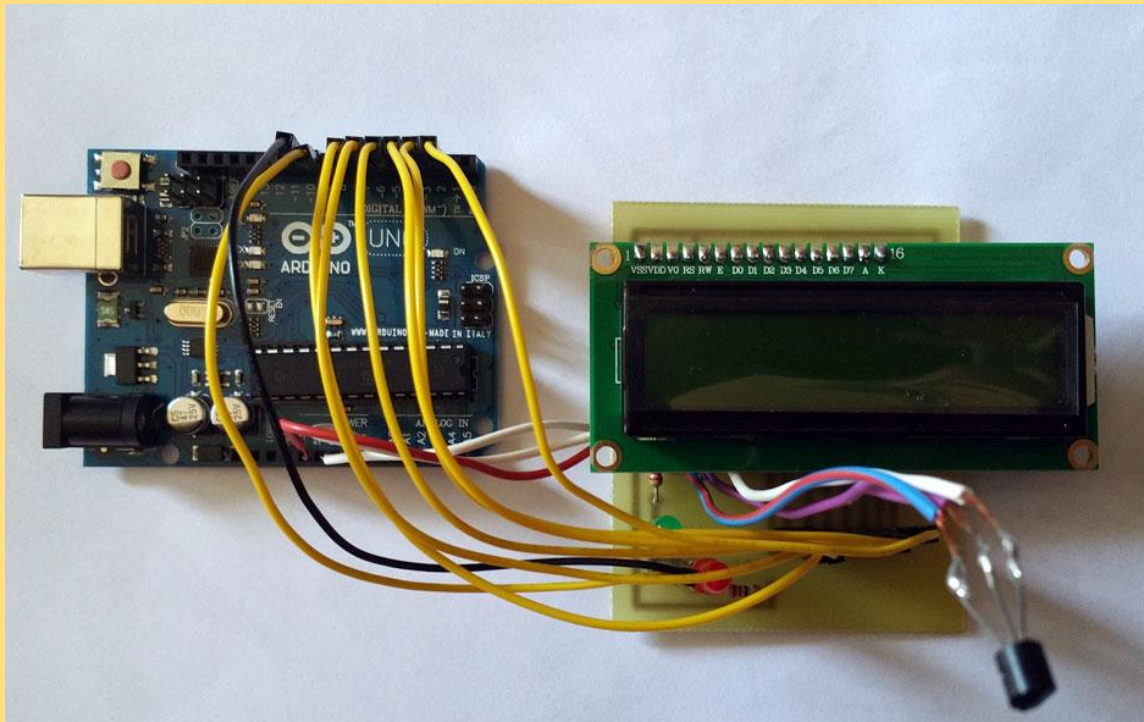
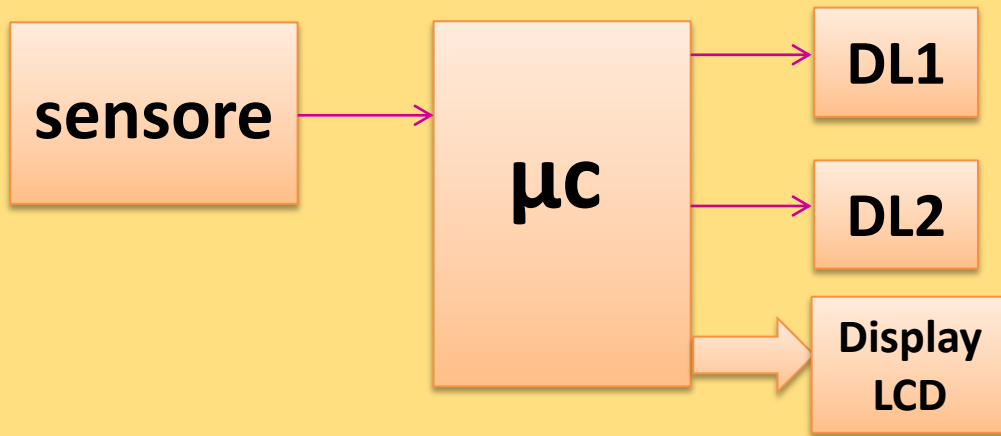
In laboratorio abbiamo realizzato un'esperienza con arduino: acquisizione della temperatura con LM35 e visualizzazione in un display LCD.

Il progetto consiste nel realizzare un termometro digitale con Arduino.

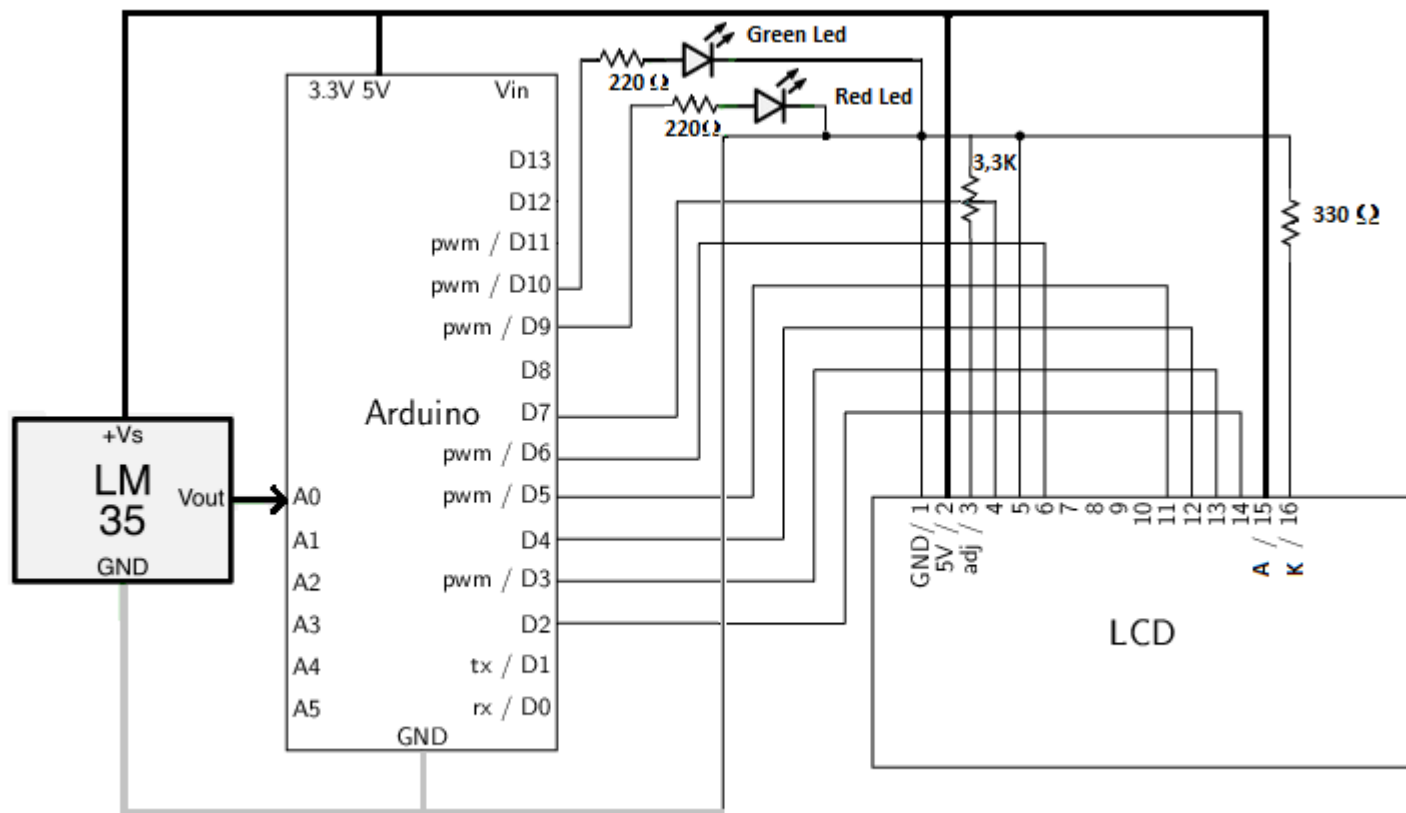
La temperatura verrà rilevata con il sensore di temperatura LM35 e dovrà essere visualizzata su un display LCD, aggiornata ogni secondo.

Per temperature minori di 20° C si accende il led verde mentre per temperature maggiori di 20° C si accende il led rosso.

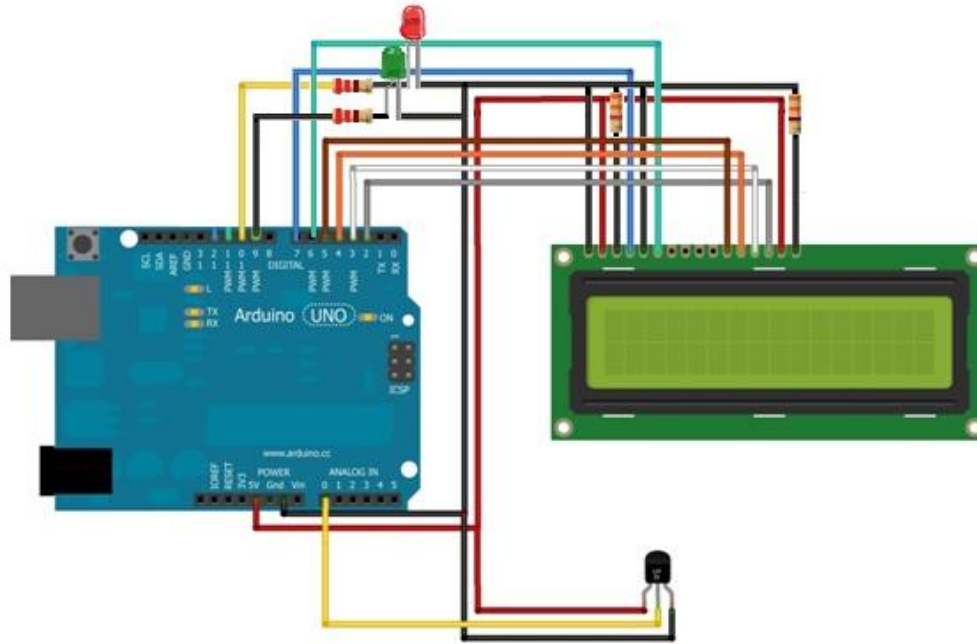
Schema a blocchi dell'esperienza



Schema elettrico

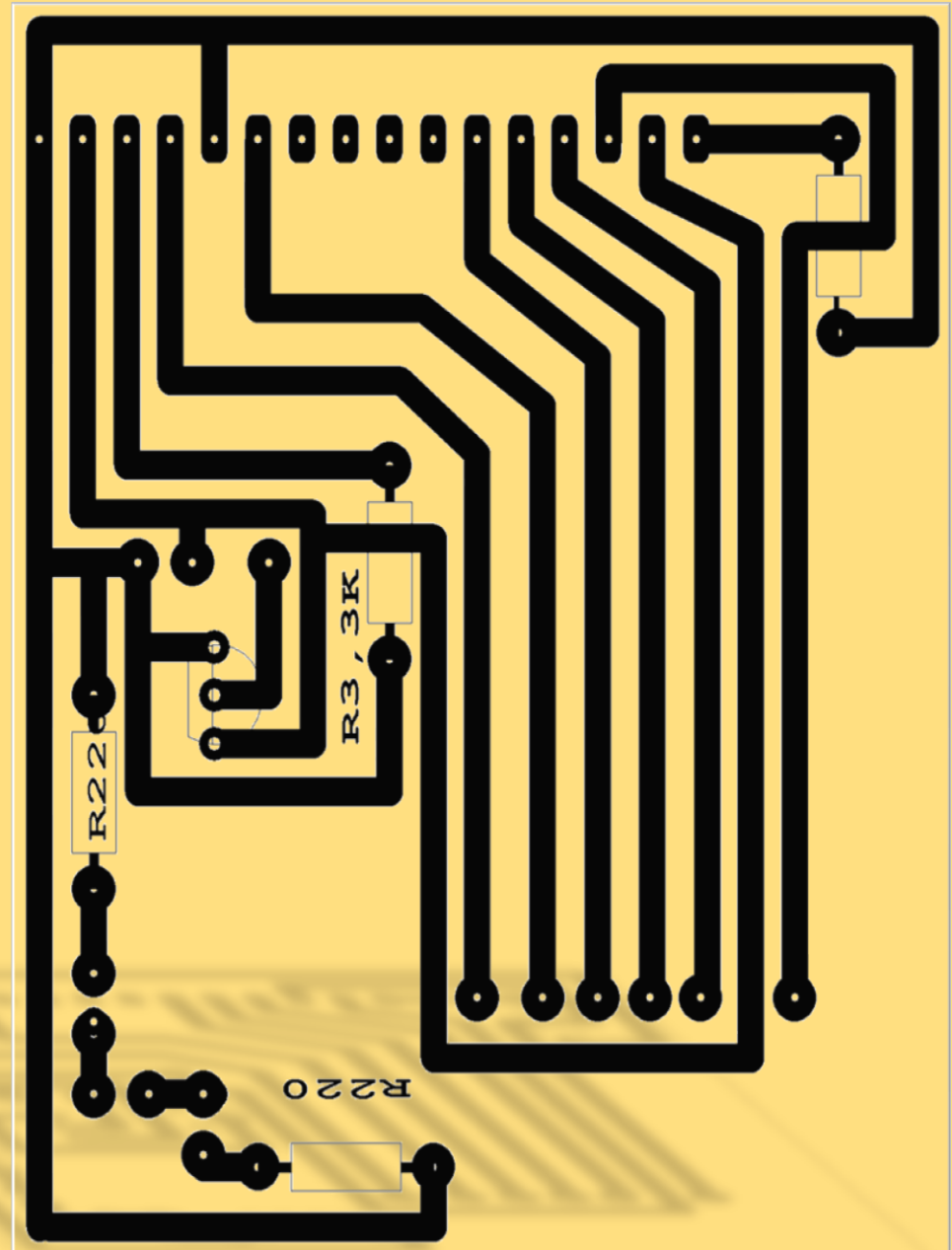


Collegamenti elettrici



Pin n°	Nome pin	Collegato a
1	Vss	GND
2	Vdd	5V
3	V0	resistenza da 3,3 kohm collegata a GND (vedi nota sotto in rosso)
4	RS	pin 7 di Arduino
5	RW	GND
6	E	pin 6 di Arduino
11	D4	pin 5 di Arduino
12	D5	pin 4 di Arduino
13	D6	pin 3 di Arduino
14	D7	pin 2 di Arduino
15	A	5V
16	K	resistenza da 330ohm collegata a GND

MASTER SCHEDA



Vista del master dal lato componenti

IL CODICE

```
#include <LiquidCrystal.h> //Libreria per pilotare il display LCD
const byte pinTemp = A0; //Pin a cui collegare il piedino Vout del sensore di temperatura
float temp = 0; //Variabile in cui verrà memorizzata la temperatura
//Specifico i pin da utilizzare per pilotare il display LCD
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
const int pin = 0;
int sensor;
const int G_LED = 9;
const int Y_LED = 10;
void setup()
{ Serial.begin(9600); // inizializza la comunicazione seriale
  pinMode(G_LED, OUTPUT);
  pinMode(Y_LED, OUTPUT);
  lcd.begin(16, 2); //Impostazione del numero di colonne e righe del display LCD
  lcd.setCursor(0, 0); //Sposto il cursore sulla prima riga (riga 0) e sulla prima colonna
  lcd.print("Temperatura:"); //Stampo il messaggio 'Temperatura:' sulla prima riga
  /*Imposto Vref dell'ADC a 1,1V
  (per una maggiore precisione nel calcolo della temperatura)
  IMPORTANTE: Se utilizziamo Arduino Mega sostituisci INTERNAL
  con INTERNAL1V1 */
  analogReference(INTERNAL); }
```

IL CODICE

```
void loop() {  
  
  /*Calcolo la temperatura*/  
  temp = 0;  
  for (byte i = 0; i < 5; i++) { //Esegue l'istruzione successiva 5 volte  
    temp += (analogRead(pinTemp) / 9.31); //Calcola la temperatura e la somma alla variabile  
'temp'  
  }  
  temp /= 5; //Calcola la media matematica dei valori di temperatura  
  /*Visualizzo la temperatura sul display LCD*/  
  lcd.setCursor(0, 1); //Sposto il cursore sulla prima colonna e sulla seconda riga del display LCD  
  lcd.print(temp); //Stampo sul display LCD la temperatura  
  lcd.print(" "); //Stampo uno spazio  
  lcd.print((char)(223)); //Stampo il carattere °  
  lcd.print("C"); //Stampo il carattere 'C'  
  delay(1000); //Ritardo di un secondo (può essere modificato)  
  if (temp<25) { // 41 corrisponde a 20°C  
    analogWrite(G_LED, 255); analogWrite(Y_LED, 0); }  
  else {analogWrite(G_LED, 0); analogWrite(Y_LED, 255); }  
  delay(1000); }
```