

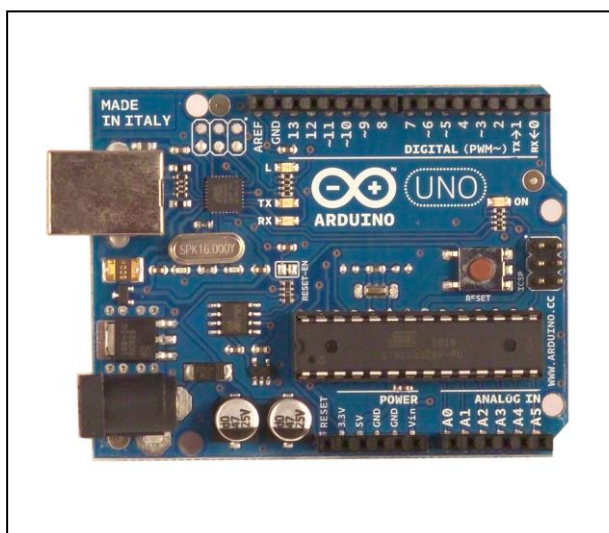


TESINA di Prisco Fabio

Applicazione pratica del microcontroller

Arduino Uno:

Dimostrazione di acquisizione e distribuzione dati della temperatura ed umidità





INDICE

GENERALITÀ	3
1 DESCRIZIONE Arduino	4
2 DESCRIZIONE Cablaggio	5
3 DESCRIZIONE Software	6
4 DESCRIZIONE Acquisizione	7
5 DESCRIZIONE Pubblicazione	8
6 IMPIEGO La Domotica	9
7 Appendice A Programma Arduino	10
8 Appendice B Mappa del Sito	13



GENERALITÀ

Questo progetto consiste nella dimostrazione di acquisizione e distribuzione dati della temperatura ed umidità attraverso l'utilizzo di un hardware, sviluppato nell'ultimo decennio, che possiede le stesse funzioni pratiche di un semplice compilatore elettronico programmabile. Utilizzando una base hardware chiamata "Arduino" e un linguaggio di programmazione C#, verrà dimostrato come è possibile creare un semplice termometro digitale che misura umidità e temperatura. Una volta creato il circuito e testato, si implementeranno i dati raccolti, tramite il salvataggio su file Excel, in un database di Access per la creazione di pagine web che prenderanno parte al sito da me creato. Ogni computer svolge calcoli aritmetici molto complessi, per questo il pc utilizza il Processore (CPU) con un sistema binario, che consiste nei soli valori logici 0 e 1; perché l'hardware lavora tramite tensioni associate che possono essere alte (1) o quasi nulle (0).

L'interfaccia in INPUT al pc consiste in piccoli dispositivi elettronici chiamati Trasduttori, i quali permettono la trasmissione di una grandezza fisica al compilatore trasformando il segnale in grandezza elettrica.

Nel mio progetto ho acquisito la temperatura e l'umidità tramite un Trasduttore chiamato DHT11. Il sensore utilizzato non è molto preciso, infatti ha un margine di errore di $\pm 2C^{\circ}$, ma è stato comunque utile al funzionamento dell'impianto.

Una volta ricevuti i dati, Arduino elabora i passaggi successivi da me ordinati. Infatti ho programmato Arduino in modo da ricevere i dati del sensore, inviarli via USB al mio pc, salvarli su un foglio Excel e parallelamente visualizzare i dati su uno schermo LCD collegato all'Arduino.

Lo schermo LCD fungerà da OUTPUT al mio Arduino per la visualizzazione dei dati della temperatura e umidità raccolti. Lo schermo in questione opera su 2 righe e 16 colonne quindi visualizza al massimo 24 caratteri.

Ovviamente per la creazione del progetto ho seguito vari passaggi sia per la creazione del circuito, sia per la programmazione dell'Arduino. Infatti se non fossi stato attento all'inserimento delle resistenze giuste nel circuito, avrei rischiato di rovinare qualche componente oppure se non avessi programmato l'Arduino in modo giusto, non avrebbe funzionato l'impianto.



PARTE - 1

Arduino

Arduino è una piattaforma hardware programmabile, con la quale è possibile creare circuiti elettronici, soprattutto in ambito di robotica ed automazione. Nasce a Ivrea, nel 2005, da un'idea di un professore universitario, Massimo Banzi, che decise di creare una piattaforma per i propri studenti, così da facilitare lo studio della programmazione e dell'elettronica .

Fu un gruppo di studenti della facoltà di Ingegneria Informatica a scrivere la libreria, l'IDE (libreria portabile su ogni sistema operativo); grazie a questi pre-ingegneri, Arduino tutt'oggi programma in modo fluido, semplice e molto intuitivo. Su Internet si possono trovare librerie già scritte in base al nostro bisogno.

Per esempio se vogliamo fare qualche applicazione e ci serve qualche funzione in particolare o qualche supporto per sensori possiamo ricorrere, appunto, alla navigazione in Internet.

C'è da aggiungere un particolare molto importante, ogni programma che si scrive su Arduino sarà naturalmente avviato a loop() finché non si toglie l'alimentazione dal dispositivo. Quando lo colleghiamo ad una fonte di alimentazione (ad esempio la USB del PC o anche una comunissima Batteria da 9V) si accende e avvia il programma caricato dall'IDE a loop infinito.

La scheda Arduino è in grado di interagire con l'ambiente in cui si trova ricevendo informazioni da una grande varietà di sensori. Ma non si parla solo di sensori, Arduino può comandare luci, LED, motori e altri attuatori. Il linguaggio di programmazione è basato su un C "specializzato".

Caratteristiche Arduino Uno R3

Microcontroller	ATmega 328P
Pin digitali I/O	14
Pin analogici	6
SRAM	2KB
EEPROM	1KB
Clock	16MHz



PARTE – 2

Cablaggio

Il cablaggio del sistema prevede l'utilizzo di un sensore di temperatura ed umidità <<DHT11>>, e di uno schermo LCD 16x2.

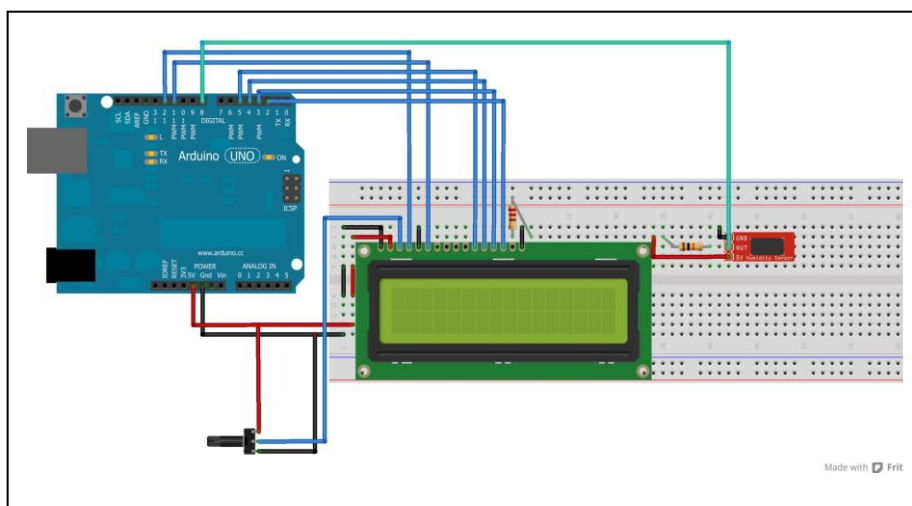
Per configurare i componenti in modo corretto con Arduino bisogna conoscere i datasheet.

Si monta lo schermo LCD disponendo i pin questo modo:

- pin 1 connesso a GND;
- pin 2 connesso a 5v;
- pin 3 connesso al potenziometro da 10Kohm (per il contrasto dei caratteri);
- pin 5 connesso a GND;
- pin 4,5,10,11,12,13 connessi ai pin digitali dell'Arduino;
- pin 14 connesso in serie ad una resistenza da 220ohm (per la retroilluminazione);
- pin 15 connesso a massa (per la retroilluminazione).

Per connettere il sensore invece bisogna fare questi 3 collegamenti:

- pin 1 connesso a 5v;
- pin 2 connesso al MC e in serie ad una resistenza da 5Kohm;
- pin 3 connesso a massa.





PARTE – 3

Software

Il programma dell'Arduino si compone principalmente in 4 parti.

La prima parte riguarda le dichiarazioni delle librerie utilizzate per usufruire del sensore e dello schermo LCD, oltre alle dichiarazioni delle variabili utilizzate dal programma e la definizione dei pin occupati dai sensori sull'Arduino.

La seconda parte riguarda il settaggio delle impostazioni dell'Arduino e l'interfacciamento con il pc `<<void setup() {..}>>`.

La terza parte, `<<void loop(){..}>>`, è il corpo centrale dell'intero programma. Eseguo vari passaggi, tutti in ordine logico per il giusto funzionamento, tra cui:

1. Lettura dei dati;
2. Funzione "Inizio"
3. Istruzione IF se le variabili che contengono i dati sono vuote allora visualizzo un messaggio di errore ALTRIMENTI visualizzo i dati su LCD e pc;

La quarta parte riguarda la programmazione della funzione "inizio".

Questa funzione ha il semplice scopo di arricchire visualmente lo schermo LCD.

Infatti la funzione non è altro che il caricamento del conto alla rovescia all'avvio del programma vero e proprio che visualizzerà i dati.

Una volta "verificato" il programma, si può caricare su Arduino.

Allego il programma nell'Appendice A .



PARTE – 4

Acquisizione

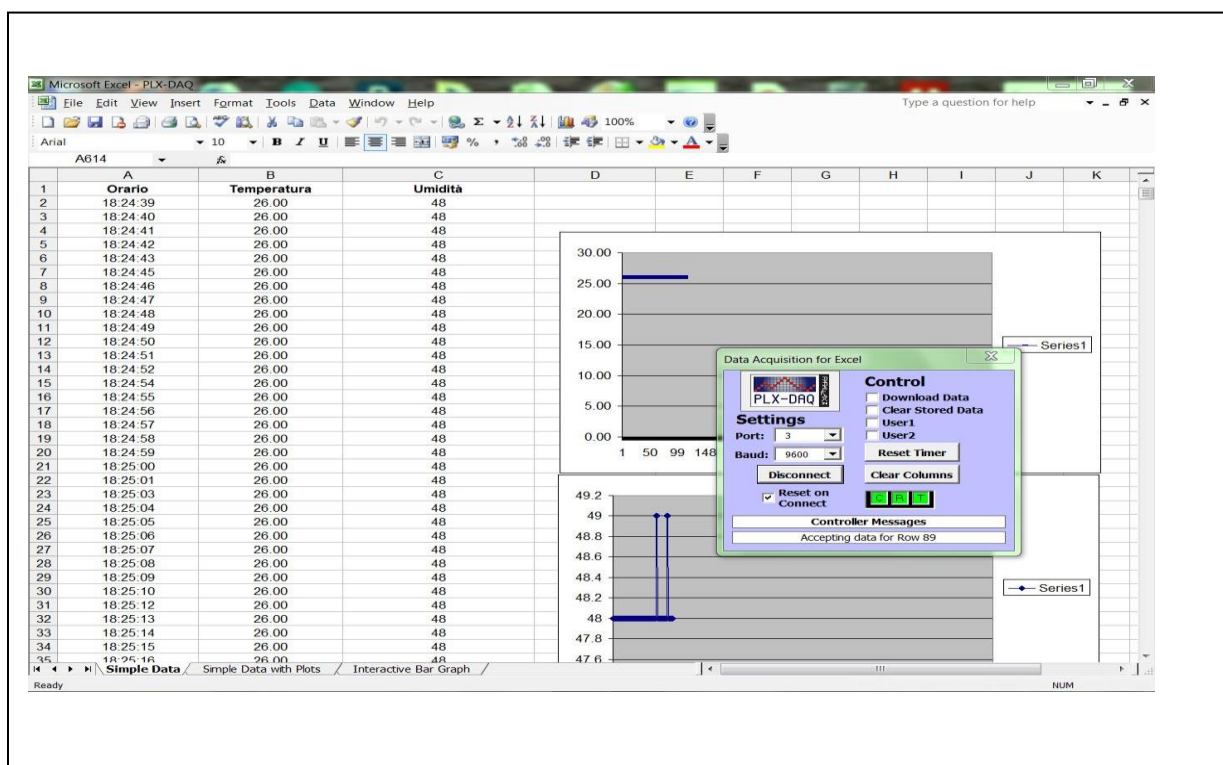
L'acquisizione dei dati avviene in via seriale tramite l'Arduino e un programma chiamato PLX-DAQ.

PLX-DAQ (Parallax Data Acquisition tool) è un software add-in per Microsoft Excel che acquisisce dati da qualsiasi microcontroller e scrive i numeri nelle colonne così come li riceve. PLX-DAQ provvede ad una facile analisi, con grafici, dei dati raccolti dai sensori con un monitoraggio in tempo reale.

PLX usa i macro di Excel, ovvero utilizza un piccolo programma che automatizza il processo di acquisizione e visualizzazione dati.

Questo programma è fornito gratuitamente dal sito inglese www.Parallax.com.

L'acquisizione della temperatura in questo progetto avviene proprio grazie a questo programma che, una volta connesso l'Arduino al pc, popolerà le tabelle di un file Excel con i dati.





PARTE – 5

Publicazione

La parte della pubblicazione riguarda la creazione di un sito web personale per poter monitorare a distanza quelle informazioni che si archiviano tramite i passaggi precedenti. Tramite i linguaggi ASP e HTML, ho creato un sito che riporta i dati delle temperature raccolte in diversi giorni per poterli confrontare ed analizzare.

Inizialmente ho creato una pagina di login e di registrazione per controllare il numero di visitatori del sito; una volta che ci si è loggati si visualizza la homepage.

Dalla homepage si può passare, tramite un menù, alle pagine secondarie del sito che riguardano la progettazione del circuito e la programmazione di Arduino; oltre ad altre pagine che trattano di Arduino, termometro e future implementazioni della piattaforma hardware utilizzata nel progetto.

Allego all'Appendice B la mappa del sito.

Per quanto riguarda lo schema delle pagine ASP che utilizzano i dati raccolti, ne riporto lo schema qua sotto:

- Si creano gli oggetti connessione e recordset;
- Si dà uno stile alla pagina in html;
- Apro la connessione e creo il recordset;
- Eseguo la Query contenente i comandi per estrarre i dati e visualizzarli;
- Creo un ciclo che visualizza tutti i dati contenuti nel recordset;
- Setto a "nullo" la connessione e il recordset;
- Chiudo la connessione e il recordset;
- Chiudo la pagina html.

Una volta fatto questo aggiungo al sito la pagina ASP.



PARTE – 6

IMPIEGO : La domotica

La domotica nasce nella seconda metà del novecento con lo scopo di migliorare la qualità della vita nella casa e più in generale negli ambienti antropizzati.

Questa area fortemente interdisciplinare richiede l'intervento di diverse scienze, tra cui elettronica ed informatica.

La domotica svolge un ruolo importante nel rendere “intelligenti” apparecchiature, impianti e sistemi. Grazie allo sviluppo tecnologico, già oggi è possibile avere una casa limitatamente automatizzata. Infatti attraverso il proprio smartphone si può controllare l'illuminazione della casa e/o la temperatura. Questo progetto può fare da base per l'aggiunta di altri sensori che possono controllare altri mezzi elettronici della casa.

Con l'aggiunta della scheda Ethernet all'Arduino, per esempio, si potrebbe creare una connessione in tempo reale con il nostro sito e fare in modo di aggiornare automaticamente, e ad intervalli prestabiliti, i dati della temperatura e dell'umidità presente in casa. Inoltre con l'utilizzo di un PWM (pulse wide modulation) si potrebbe controllare l'apertura delle finestre, porte o altro ancora, con dei motori elettrici connessi all'Arduino.

In sintesi Arduino è una tecnologia che ha molto da dare, se ben sviluppato, inoltre è già stato adottato dal sistema operativo per dispositivi mobili Android che ha sviluppato diverse applicazioni e progetti.



Appendice A

Programma Arduino

```
#include "DHT.h" //includo la libreria DHT.h//
#include <LiquidCrystal.h> //includo la libreria LiquidCrystal.h//
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //imposto lo schermo//

#define DHTPIN 8 //dichiaro al pin 8 il sensore//
#define DHTTYPE DHT11 //definisco il tipo di sensore//

DHT dht(DHTPIN, DHTTYPE);

void setup() { //avvio il setup//
  lcd.begin(16, 2); //lo schermo ha 16 colonne e 2 righe//
  Serial.begin(9600); //indico la velocità di trasferimento dati//
  Serial.print("CLEARDATA"); //“pulisco” lo schermo//
  inizio(); //avvio della funzione “inizio”//
  dht.begin(); //avvio il sensore//
}
void loop() { //avvio il programma//
  int h = dht.readHumidity(); //inserisco in h l'umidità//
  int t = dht.readTemperature(); //inserisco in t la temperatura//

  if (isnan(t) || isnan(h)) { //se le variabili sono vuote...//
    lcd.setCursor(0,0);
    lcd.print("Errore."); //visualizzo il messaggio “errore”//
  } else { //altrimenti visualizzo i dati//
    lcd.setCursor(0,0);
    lcd.print("Umidita': ");
    lcd.setCursor(11,0);
    lcd.print(h);
    lcd.print(" %");
    lcd.setCursor(0,1);
    lcd.print("Temp.: ");
    lcd.setCursor(11,1);
    lcd.print(t);
    lcd.print(" C");
    Serial.print("DATA,TIME,"); //invio i dati al pc su Excel//
    Serial.print(t);
    Serial.print(",");
    Serial.print(h);
    int row=0;
    row++;
    if (row>360){
      row=0;
    }
  }
}
```



```
        Serial.println("ROW,SET,2");
    }
    delay(100);

    Serial.println();
    delay(500);
}
void inizio(){ //avvio della funzione "inizio"//
    lcd.setCursor(0,0);
    lcd.print("Avvio in corso");
    lcd.setCursor(6,1);
    lcd.print("Loading");
    lcd.setCursor(15,1);
    for (int k=0;k<1;k++) {
        lcd.print("0");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Avvio in corso");
        lcd.setCursor(6,1);
        lcd.print("Loading");
        lcd.setCursor(15,1);
        lcd.print("1");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Avvio in corso");
        lcd.setCursor(6,1);
        lcd.print("Loading");
        lcd.setCursor(15,1);
        lcd.print("2");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Avvio in corso");
        lcd.setCursor(6,1);
        lcd.print("Loading");
        lcd.setCursor(15,1);
        lcd.print("3");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Avvio in corso");
        lcd.setCursor(6,1);
        lcd.print("Loading");
        lcd.setCursor(15,1);
        lcd.print("4");
        delay(1000);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Avvio in corso");
```



```
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(15,1);  
lcd.print("5");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Avvio in corso");  
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(15,1);  
lcd.print("6");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Avvio in corso");  
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(15,1);  
lcd.print("7");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Avvio in corso");  
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(15,1);  
lcd.print("8");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Avvio in corso");  
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(15,1);  
lcd.print("9");  
delay(1000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Avvio in corso");  
lcd.setCursor(6,1);  
lcd.print("Loading");  
lcd.setCursor(14,1);  
lcd.print("00");  
delay(1000);  
lcd.clear();  
}  
lcd.setCursor(2,0);  
lcd.print("Starting...");  
delay(1000);  
lcd.clear();}
```



Appendice B
Mappa del Sito

