[Misuratore distanza ad ultrasuoni](http://www.ipsiaplcarduinolab.altervista.org/index.php/arduino/a-s-2016-2017/50-misuratore-distanza-ad-ultrasuoni)

<http://www.ipsiaplcarduinolab.altervista.org/index.php/arduino/a-s-2016-2017/50-misuratore-distanza-ad-ultrasuoni>

AS 2016\_2017

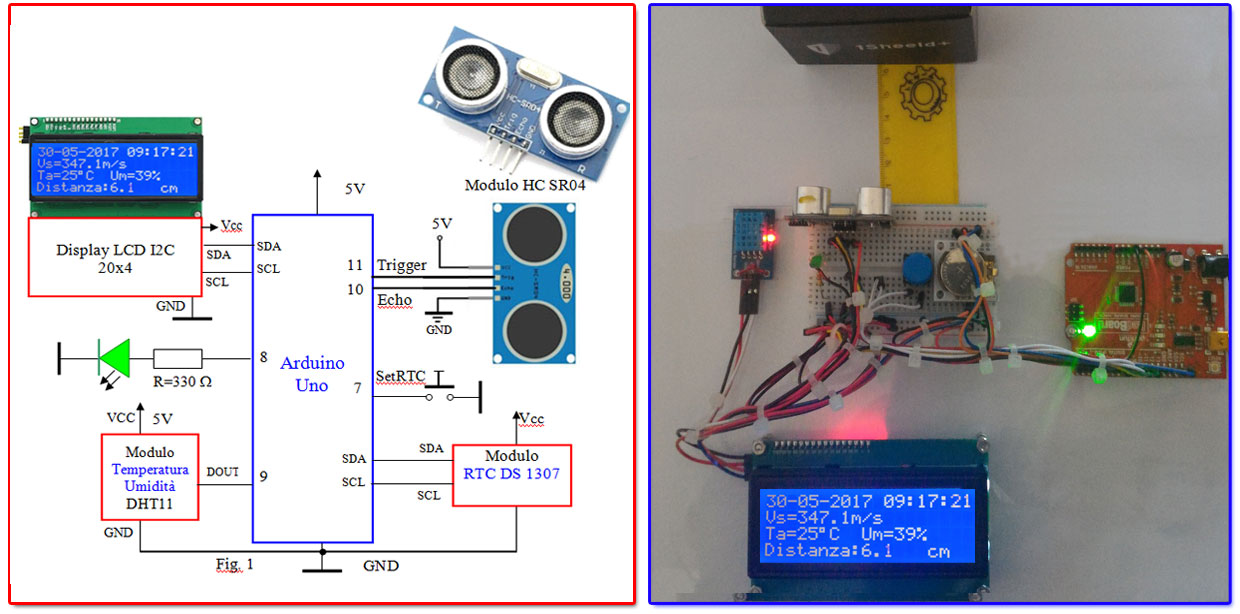
*Pubblicato: 07 Febbraio 2017*

*Scritto da De Marco Luca - Eusebi Valerio - Pop Raul*

*Categoria:*[*Tesine-Arduino\_2016\_2017*](http://www.ipsiaplcarduinolab.altervista.org/index.php/arduino/a-s-2016-2017)

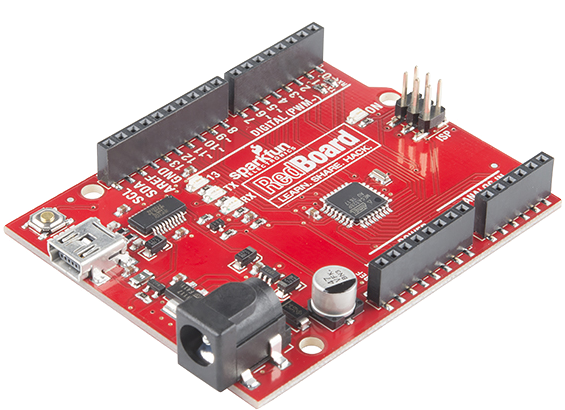
**Eusebi Valerio - De Marco Luca - Pop Raul - Classe 5A-IPAI -  Esame di Stato: 2016-2017         [](http://www.ipsiaplcarduinolab.altervista.org/index.php/arduino/a-s-2016-2017/50-misuratore-distanza-ad-ultrasuoni#uk-pop)**  
**Misuratore distanza ad ultrasuoni**  
***Applicazione con Arduino e Modulo HC SR04***

**Descrizione**  
Il circuito, mediante un modulo ad ultrasuoni HC SR04, misura la distanza in cm tra il sistema ed un oggetto.   
La misura viene effettuata correggendo la velocità del suono in funzione della temperatura e dell'umidità ambientale. Lo stato di un Led Verde segnala se la distanza supera 10 cm. I dati acquisiti ed elaborati insieme alla data e all'ora corrente vengono visualizzati su un display LCD.

[](http://www.ipsiaplcarduinolab.altervista.org/images/2016_2017/De_Marco_Luca/schema-misura-distanza.jpg)

Lo schema è costituito dalle seguenti parti/moduli  
**1) Arduino UNO/RedBoard  
2) Modulo ad ultrasuoni HC SR04  
3) Display LCD I2C 20x4  
4) Modulo RTC DS 1307  
5) Modulo Temperatura-Umidità DHT11  
6) LED Verde**

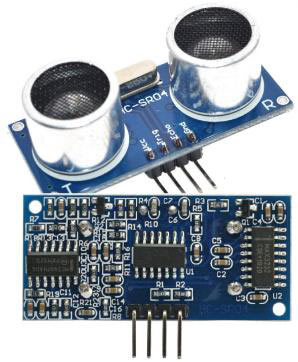
**1) Arduino UNO/RedBoard**

[](http://www.ipsiaplcarduinolab.altervista.org/images/2016_2017/Pop_Raul_Petru/redboard1.png)UKThe arduino UNO is a board based on the ATmega328 microcontroller.  
The arduino UNO board is equipped with 14 digital input/output pins, 6 analogic inputs, a 16MHz quartz, a USB connector, a power jack, a ICSP programming connector and a reset button.  
The board also provides anything you may need to support the microcontroller's operation.  
To use the arduino Uno you need to connect the board to a PC by using a USB cable or power the card  by using a AC/DC adapter or by a simple battery.

ita La Arduino UNO è una scheda basata sul microcontrollore ATmega328 .  
La scheda Arduino UNO è dotata di 14 pin di input/output digitali,6 input analogici, un quarzo a 16MHz, un connettore USB, un jack per l'alimentazione, un connettore per la programmazione ICSP ed un pulsantino per il reset della scheda. La scheda inoltre fornisce tutto ciò che è necessario per supportare il funzionamento del microcontrollore.  
Per utilizzare la Arduino UNO è necessario connettere la scheda ad un PC tramite uncavo USB oppure alimentare la scheda tramite un adattatore AC/DC o tramite una semplice batteria.

|  |  |  |
| --- | --- | --- |
| **Specifiche tecniche:** |  |  |
| ● Microcontrollore: Atmel ATmega328 | ● DC Current per I/O Pin: 40 mA | ● SRAM: 2 KB |
| ● Tensione operativa: 5V | ● DC Current per Pin alimentati a 3.3V: 50mA | ● EEPROM: 1KB |
| ● Input Voltage: (consigliata) 7-12V | ● Flash Memory: 32 KB (di cui 0.5KB utilizzati dal bootloader) | ● Pin di I/O Digitali: 14 (6 PWM) |
| ● Input Voltage: (limiti) 6-20V | ● Pin di Input Analogici: 6 | ● Frequenza di Clock: 16MHz |

**2) Modulo ad ultrasuoni HC SR04**

Questo modulo, con tecnologia ad ultrasuoni, è capace di misurare una distanza compresa in un range tra 2-400 cm con una precisione di 3 mm.  
E' solitamente usato in applicazioni di robotica, sistemi di sicurezza, misuratore di livello nei serbatoi o in sostituzione di unità ad infrarossi.  
Il sensore HC-SR04 è costituito da una scheda, che presenta nella sua parte posteriore un circuito elettronico basato su un microcontrollore della ST, e nella parte anteriore sono presenti un quarzo e due cilindri metallici, i trasduttori ad ultrasuoni.

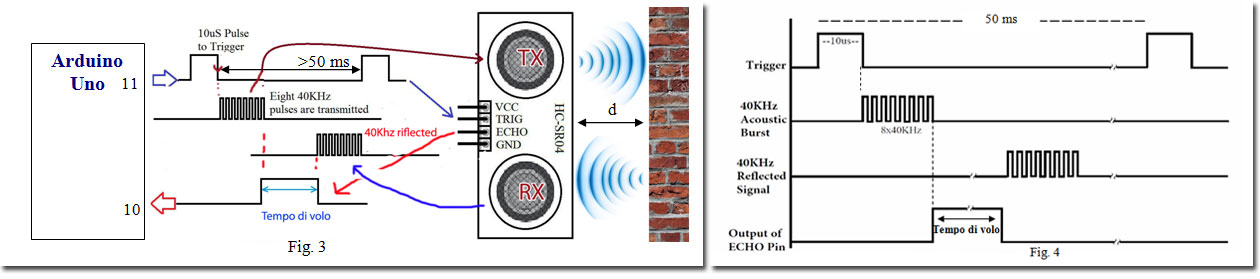
Uno di questi invia ultrasuoni (TX) che rimbalzano contro ad un qualunque oggetto posto di fronte ad esso, ed entrano di ritorno nell'altro cilindro (RX).

Il modulo dispone di quattro terminali di collegamento, due di alimentazione (Vcc, GND) e due di controllo (Trigger, Echo).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Caratteristiche tecniche** | | | Tensione di lavoro | 5 Vdc | | Corrente assorbita | >2 mA | | Frequenza di lavoro | 40 Khz | | Distanza max | 400 cm | | Distanza min | 2 cm | | Risoluzione | 3 mm | | Angolo di misura | 15° | | Ingresso Trigger 10µs | Impulso TTL | | Uscita Echo | segnale PWM TTL | | Dimensioni | 45,5 x 20,5 x 15,3 mm | | Tabella hcsr04 | [Schema bello2](http://www.ipsiaplcarduinolab.altervista.org/images/2016_2017/Pop_Raul_Petru/Schema-bello.jpg) |

**Funzionamento: procedura di misura - Sequenza di funzionamento:**Sequenza di funzionamento (Fig. 3 e Fig. 4):

**1.** Si invia tramite il Pin 11 di Arduino un impulso di Trigger della durata di 10 µs.  
**2.** Sul fronte di discesa dell'impulso di Trigger il Modulo tramite la capsula TX invia un pacchetto di 8 impulsi alla frequenza di 40 Khz (ultrasuoni) (invia un ping).  
**3.** Sul fronte di discesa dell'ultimo impulso il terminale Echo, collegato al pin 10 di Arduino, si porta a livello alto.  
**4.** Se il segnale emesso da TX incontra un ostacolo torna indietro e viene captato dalla capsula RX.  
**5.**Quando la capsula RX capta il segnale riflesso il segnale Echo torna a livello basso.  
**6.**La durata dell'impulso di Echo si chiama tempo di volo (µs) e rappresenta il tempo di viaggio (andata e ritorno) del segnale (alla velocità del suono) impiegato per percorrere lo spazio tra le capsule ed l'ostacolo.   
**7.** Dopo 30 ms si considera che non sia stato incontrato nessun ostacolo, per sicurezza si aspettano 50-60 ms per far si che non vi siano interferenze con la misura successiva.

[](http://www.ipsiaplcarduinolab.altervista.org/images/2016_2017/Pop_Raul_Petru/fig-3-4-qa.jpg)

Conoscendo la velocità del suono (343,8 m/s a 20°C) e sapendo che il “viaggio” dell’onda sonora è stato il doppio della distanza tra il modulo e l’ostacolo  la distanza è derivata dalla seguente formula:

d

**d**=   Distanza in metri tra il modulo HC-SR04 e l’ostacolo  
***Tv***= Tempo di volo in secondi  
***Vs***= Velocità del suono (343,8 m/s a 20°C)

|  |  |
| --- | --- |
| Il modulo restituisce il tempo di volo in µs, per una semplicità di lettura conviene avere  la distanza in cm, si trasforma la velocità del suono da m/s a cm/µs.  L'espressione per il calcolo della distanza in cm diviene: | dCM |

La velocità del suono in aria varia con la temperatura e l' umidità.  
Pertanto, al fine di calcolare con precisione la distanza, occorre prendere in considerazione la temperatura ambiente e l'umidità locale.  
La formula per la velocità del suono in aria in funzione della temperatura e l'umidità è:

**Vs= 331,45 + (0,606 x T) + (0,0124 x H)**

**Vs** = velocità del suono in metri al secondo (m /s)  
**331,45** = velocità del suono (in m/s) a 0 °C e 0% di umidità  
**T** = temperatura in °C  
**H** =% di umidità (umidità relativa)  
**0,606** e **0,0124** = costanti

**Ad esempio**, a 20 °C e 50% di umidità il suono viaggia ad una velocità di: Vs = 331,45 + (0,606 x 20) + (0,0124 x 50) Vs= 344,19 m/s.  
Per la conversione da m/s in a cm/µs si divide per 104    -Vs= 344,19 m/s. Vs=0,034419 cm/µs  
Nell'equazione sopra la temperatura ha un'influenza maggiore rispetto l'umidità.

**Nota:** il sensore intercetta i segnali di ritorno con un “angolo di visuale” di circa 15 gradi per cui intercetta anche l’eventuale segnale di ritorno proveniente dal pavimento, dal soffitto o da eventuali pareti laterali. Per ottenere una valida misurazione bisogna quindi tenere il modulo ad una altezza da terra o da ostacoli laterali sufficiente ad evitare  interferenze. In linea di massima puo’ essere utilizzata la seguente formula per calcolare la posizione del sensore in funzione della distanza massima misurabile.  
Puo’ essere utilizzata la seguente formula per calcolare la posizione del sensore in funzione della distanza massima misurabile:  
**Posizione del sensore = Distanza massima misurabile \* 0,26**(0,26 = tan 15°)  
Se ad esempio vogliamo individuare ostacoli a 1 metro di distanza, dobbiamo solevare il sensore ad almeno 26 cm da terra.

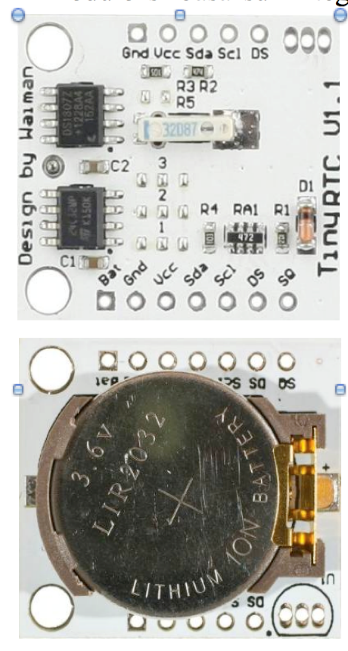
**3) Display LCD I2C 20x4**



Il display LCD I2C dispone di 4 righe e 20 colonne e viene utilizzato per visualizzare i dati quali, in questo caso: data, ora, temperatura, umidità, velocità del suono e distanza.Il display, dul retro della basetta, è fornito di un trimmer per la regolazione del contrasto e 4 piedini: vcc, gnd, SDA e SCL, questa dua caratteristi rende la comunicazione display-arduino molto più semplice rispetto ad altri modeli.Lo schermo a cristalli liquidi (LCD) è un display a schermo piatto, una visualizzazione elettronica o un video che utilizza le proprietà modulanti della luce dei cristalli liquidi. I cristalli liquidi non emettono direttamente la luce.

Specifiche:  
● Alimentazione: 5V  
● Interfaccia: I2C  
● Indirizzo I2C: 0x3F  
● Retroilluminazione blu e caratteri bianchi  
● Retroilluminazione: selezionabile tramite ponticello  
● Compatibile: Arduino e Raspberry Pi  
● Dimensioni: 60 x 90 mm  
● Peso: 75,5g

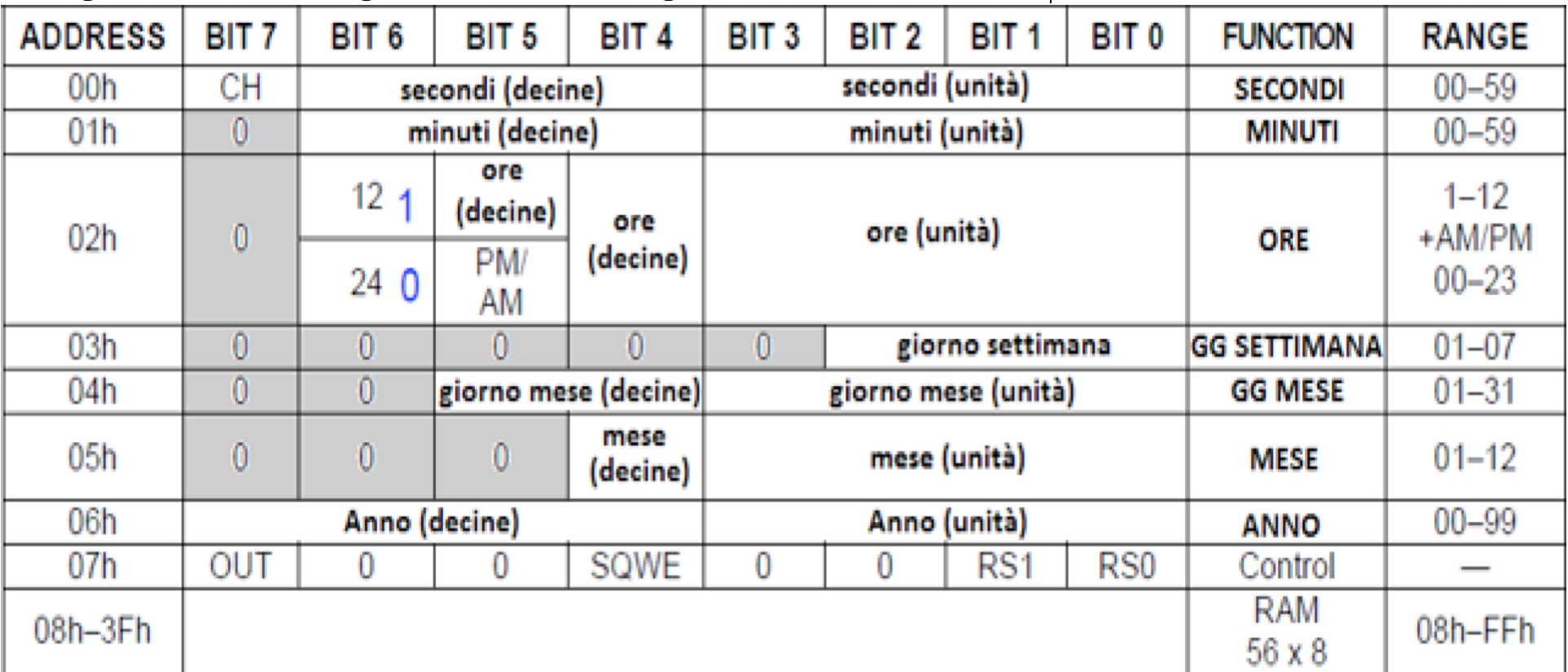
**4) Modulo RTC DS 1307**



Il modulo  RTC DS 1307 è dotato di EEPROM con 32kbyte memoria. Il modello in esame integra il circuito per l’istallazione del sensore di temperatura DS18B20. È presente una batteria tampone ricaricabile che consente di alimentare il dispositivo e ne garantisce l'automantenuta per un lungo periodo anche quando è disconnesso da Arduino. Il circuito risulta essere così totalmente indipendente: incrementa l’ora senza aver bisogno di un microcontrollore e con la batteria completamente carica è in grado di funzionare per 1 anno intero quando il sensore di temperatura risulta spento oppure non presente. L'orario e il calendario sono salvati nei registri in formato BCD. In questo formato ogni cifra di un numero è rappresentata con un codice binario a quattro bit, il cui valore è compreso tra 0000 (0) e 1001 (9). Ad esempio il numero 127 in formato BCD viene registrato in questo modo:

0001 0010 0111.

Nella seguente figura viene mostrata la tabella degli indirizzi dei registri interni dell'RTC.



I registri dell'RTC vanno dall'indirizzo 00h a 07h.

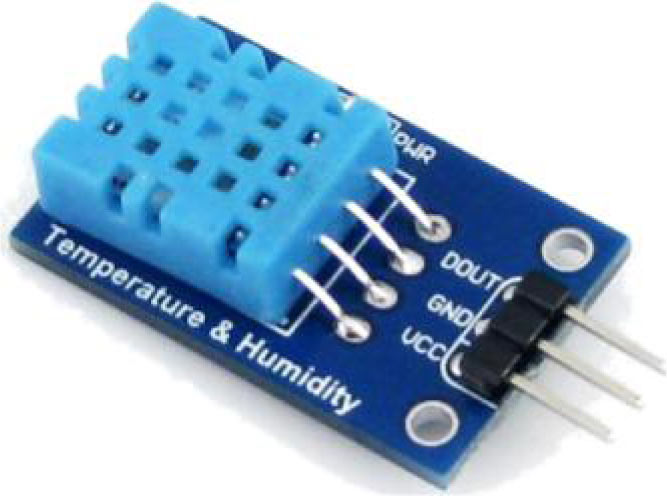
I registri utilizzabili come RAM vanno da 08h a 3Fh.

- Il Bit 7 del registro dei secondi (00h) è il bit di halt (CH) dell'orologio. Quando questo bit è posto a 1 l'oscillatore è spento mentre se posto a 0 l'oscillatore viene riattivato.

- I valori che corrispondono al giorno della settimana vengono definiti dall'utente (solitamente 1 = domenica, 2 Lunedì etc.). Allo scadere della mezzanotte questi vengono incrementati di uno (se vale 7 viene riportato a 1).  
- Il Bit 6 del registro delle ore (02h) indica se l'orario è definito con modalità 12-ore (quando vale 1) oppure 24-ore (quando vale 0). Nel caso si utilizzi la modalità a 12 ore il bit 5 (bit AM/PM) viene posto a 1 se è PM, 0 se è AM. Nel caso sia abilitata la modalità 24-ore il 5 bit è usato per la codifica delle decine delle ore. Se si cambia da 12 a 24 ore l'orario deve essere resettato.

- il registro di controllo (08h) è utilizzato per pilotare l'uscita SQ.

**5) Modulo Temperatura-Umidità DHT11**



Il modulo utilizza il sensore DHT11, sensore di temperatura e umidità con uscita dei dati in

formato digitale. Esso rileva i valori di umidità e temperatura e, attraverso un  
microcontrollore ad 8 bit in esso racchiuso, li trasforma in segnali digitali.  
Si tratta di quindi di un componente evoluto, composto da un sensore di umidità di tipo  
resistivo, un sensore di temperatura di tipo NTC (Negative Temperature Coefficient) e da un microcontrollore.

Questo sensore, su sollecitazione di Arduino, trasmette sul pin DOUT (Uscita digitale) un treno di quaranta bit:  
-8 bit per indicare la parte intera del valore di umidità  
-8 bit per indicare la parte decimale dell’umidità   
-8 bit per indicare la parte intera della temperatura  
-8 bit per indicare la parte decimale della temperatura

-8 bit per indicare il numero di controllo



**Software**Di seguito è riportato il software (Sketch) di gestione

/\* ----------------------------------------------------------------------------------------  
- Misuratore distanza con il modulo HC SR04  
- Applicazione con Arduino e Modulo HC SR04  
- Gestione modulo tramite la libreria NewPing  
- correzione della velocità del suono in base alla temperatura ed all'umidità.  
- Visualizzazione sul display 20x4: Distanza in cm, Velocità del suono, Temperatura - Umidità  
- Eusebi Valerio - De Marco Luca - Pop Raul - Esame di Stato A.S.2016-2017 - CL:5AIPAI  
-----------------------------------------------------------------------------------------  
\*/  
#include <NewPing.h>  
#include <Wire.h>  
#include <RTClib.h>  
RTC\_DS1307 RTC;  
#define DS1307\_I2C\_ADDRESS 0x68  
#include <LiquidCrystal\_I2C.h>  
LiquidCrystal\_I2C lcd(0x27,20,4);  
#include <dht11.h>  
#define TRIGGER\_PIN 11 // Pin di Arduino collegato al pin "TRIGGER" del sensore a ultrasuoni  
#define ECHO\_PIN 10 // Pin di Arduino collegato al pin "ECHO" del sensore a ultrasuoni  
#define MAX\_DISTANZA 400 // Massima distanza del ping (cm) - Massima distanza=400 cm  
dht11 DHT11;  
#define DHT11PIN 9  
NewPing sonar(TRIGGER\_PIN, ECHO\_PIN, MAX\_DISTANZA); // Setup del sensore (pin e massima distanza)  
char ora[9];  
char data[12];  
int pulsanteset = 7;  
byte buzzerled=8;

void setup()   
{  
pinMode (pulsanteset, INPUT\_PULLUP);  
pinMode (buzzerled, OUTPUT);  
Wire.begin();  
RTC.begin();  
Wire.beginTransmission(DS1307\_I2C\_ADDRESS);  
Wire.write(7);  
Wire.write(B10010000); // sends 0x10 (hex) 00010000 (binary) to control register - turns on square wave  
Wire.endTransmission();  
lcd.init();   
lcd.backlight();  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Misura Distanza");  
lcd.setCursor(0,1);  
lcd.print("Ultrasuoni: SR04");  
delay (2000);  
lcd.clear();  
  
}

void loop()   
{  
boolean set = digitalRead (pulsanteset);  
if (set == LOW)  
{  
RTC.adjust(DateTime(\_\_DATE\_\_, \_\_TIME\_\_)); // Imposta Data e Ora del Computer  
}  
DateTime now = RTC.now(); // Legge i dati dal modulo RTC DS1307  
sprintf(data, "%02d-%02d-%d", now.day(),now.month(), now.year()); // Composizione stringa data  
sprintf(ora, "%02d:%02d:%02d", now.hour(), now.minute(), now.second()); // Composizione stringa ora  
int chk = DHT11.read(DHT11PIN);  
int h = DHT11.humidity;  
//int t = DHT11.temperature;  
int t=18;  
float vs = 331.45 + (0.606 \* t) + (0.0124 \* h); //Velocità del suono in m/s - Correzione con Temp. e Umidità  
unsigned int tempo= sonar.ping\_median(40); // Invia un ping multiplo (40 ping),restituisce il tempo medio di echo in microsecondi  
String vstu="Vs="+String(vs,1)+"m/s";// Composizione stringa dati (Velocità del suono e Temp.)  
float d=(vs\*tempo/2.0)/10000.0;

lcd.setCursor(0,0);  
lcd.print(data);  
lcd.setCursor(11,0);  
lcd.print(ora);  
lcd.setCursor(0,1);  
lcd.print(vstu);  
lcd.setCursor(0,2);  
lcd.print("Ta=");  
lcd.setCursor(3,2);  
lcd.print(t,DEC);  
lcd.print(char(223));  
lcd.print("C");  
lcd.setCursor(9,2);  
lcd.print("Um=");  
lcd.setCursor(12,2);  
lcd.print(h,DEC);  
lcd.print("%");  
lcd.setCursor(0,3);  
lcd.print("Distanza: ");  
lcd.setCursor(9,3);  
lcd.print(d,1);  
lcd.setCursor(15,3);  
lcd.print("cm");  
lcd.setCursor(19,3);  
lcd.print(" ");

if (d>=10.0)  
{  
digitalWrite(buzzerled,HIGH);  
lcd.setCursor(19,3);  
lcd.print("!");  
}  
else  
{  
digitalWrite(buzzerled,LOW);  
lcd.setCursor(19,3);  
lcd.print("%");  
}  
delay(50);// Ritardo 50ms tra 2 ping (20 ping al secondo). 29ms è il ritardo minimo ammesso tra 2 ping.  
}

**Bibliografia:**  
1) Appunti scolastici  
2)

|  |  |  |
| --- | --- | --- |
| **Allegati:** | | |
| **[Scarica questo file (DHT11_Datasheet.pdf)](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/DHT11_Datasheet.pdf)**[**DHT11\_Datasheet.pdf**](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/DHT11_Datasheet.pdf) | **[DataSheet DHT11]** | **842 kB** |
| **[Scarica questo file (DS1307 datasheet.pdf)](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/DS1307%20datasheet.pdf)**[**DS1307 datasheet.pdf**](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/DS1307%20datasheet.pdf) | **[DataSheet RTC DS 1307]** | **276 kB** |
| **[Scarica questo file (HCSR04-Datasheet.pdf)](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/HCSR04-Datasheet.pdf)**[**HCSR04-Datasheet.pdf**](http://www.ipsiaplcarduinolab.altervista.org/attachments/article/50/HCSR04-Datasheet.pdf) | **[DataSheet HC SR04]** |  |