**Tre distanze differenti con Arduino**

In questo progetto è stato utilizzato il sensore di distanza **HC-SR04**.

Il sensore è costituito da due cilindri vuoti trg ed echo; trg invia ultrasuoni che rimbalzano contro il generico oggetto posto di fronte ad esso ed entrano in echo. Il sensore misura il tempo che ha impiegato il suono e da questo si può ricavare la distanza.

L'HC-SR04 è un **sensore digital**e che misura distanze comprese tra 2 e 400 cm con una precisione di 3 mm.

Caratteristiche tecniche:

* Tensione di lavoro: 5 v;
* Corrente assorbita: 15 mA;
* Frequenza di lavoro: 40 Hz;
* Distanza max: 400 cm;
* Distanza min: 2 cm;
* Angolo di misura: 15°



Dallo schema si nota che esistono tre fasi per la misura delle distanze

1. Invio di un impulso di 10 us al sensore sul pin trig;
2. Il sensore invia 8 impulsi di ultrasuoni ad una frequenza di 40 Hz;
3. Calcolo del tempo da quando si è mandato il primo impulso fino a quando riceve un impulso sul pin echo.

**Distanza = tempo \* velocità del suono / 2 = tempo \* 3,4 \* 10^-2 / 2**

Velocità di propagazione delle onde sonore

La velocità di propagazione di un’onda sonora dipende dal mezzo che deve attraversare e non dalle caratteristiche del suono. La velocità delle onde aumenta all’aumentare della temperatura e della densità del mezzo e per quanto riguarda quelli presi in esame, essa è:

|  |  |
| --- | --- |
| **Mezzo** | **Velocità (m/s)** |
| Aria | 340 |
| Acqua | 1480 |
| Acciaio | 5980 |

Programma

Il programma è stato realizzato per misurare distanze e contare il numero di volte che si verificano tre differenti:

d<50, 50<d<100, d>100

#include <LiquidCrystal.h>

/\* Corrispondenza pin LCD <-> pin digitali di Arduino \*/

#define RS 12

#define EN 11

#define D7 2

#define D6 3

#define D5 4

#define D4 5

LiquidCrystal lcd( RS, EN, D4, D5, D6, D7 );

const int TRIG = 8;

const int ECHO= 9;

 int n=0, m=0, k=0;

void setup() {

 Serial.begin(9600);

 lcd.begin( 2, 16 );

 pinMode(TRIG,OUTPUT);

 pinMode(ECHO,INPUT);

}

void loop()

{

 long durata, distanza;

 digitalWrite(TRIG, LOW);

 delayMicroseconds(2);

 digitalWrite(TRIG, HIGH);

 delayMicroseconds(10);

 digitalWrite(TRIG, LOW);

 durata = pulseIn(ECHO,HIGH);

 distanza = durata / 29.1 / 2 ;

 if (distanza <= 0){

 Serial.println("Out of range");

 }

 else {

 Serial.print(distanza);

 Serial.println("cm");

 Serial.println();

 if(distanza<100){

 if(distanza<50){

 n=n+1;

 }

 else if(distanza>50)m=m+1;

 }

 else if(distanza>100&&distanza<150){k=k+1;

 }

 lcd.clear(); // Pulisce lo schermo

 lcd.setCursor( 0, 0 );

 lcd.print( distanza );

 lcd.print("cm");

 lcd.print(" ");

 lcd.print("n=");

 lcd.print(n);

 lcd.setCursor( 0, 1 );

 lcd.print("m=");

 lcd.print(m);

 lcd.print(" ");

 lcd.print("k=");

 lcd.print(k);

 delay(1000);

 }

}

Collegamento con LCD2X16



Se si vuole considerare che la velocità del suono dipende anche dalla temperatura, si inserisce nel circuito un sensore di temperatura tipo lm35 e si considera la seguente relazione:

cm = 0.0331 + ( 0.000062 \* temp)

#include <LiquidCrystal.h>

/\* Corrispondenza pin LCD <-> pin digitali di Arduino \*/

#define RS 12

#define EN 11

#define D7 2

#define D6 3

#define D5 4

#define D4 5

LiquidCrystal lcd( RS, EN, D4, D5, D6, D7 );

const int TRIG = 8;

const int ECHO= 9;

 int n=0, m=0, k=0;

 float temp;

void setup() {

 Serial.begin(9600);

 lcd.begin( 2, 16 );

 pinMode(TRIG,OUTPUT);

 pinMode(ECHO,INPUT);

}

void loop()

{temp = ( 4.4 \* analogRead(A0) \* 100.0) / 1024.0;

 long durata, distanza;

 digitalWrite(TRIG, LOW);

 delayMicroseconds(2);

 digitalWrite(TRIG, HIGH);

 delayMicroseconds(10);

 digitalWrite(TRIG, LOW);

 float cm = 0.0331 + ( 0.000062 \* temp);

 durata = pulseIn(ECHO,HIGH);

 distanza = (durata\*cm/2.0);

 if (distanza <= 0){

 Serial.println("Out of range");

 }

 else {

 Serial.print(distanza);

 Serial.println("cm");

 Serial.println();

 if(distanza<100){

 if(distanza<50){

 n=n+1;

 }

 else if(distanza>50)m=m+1;

 }

 else if(distanza>100&&distanza<150){k=k+1;

 }

 lcd.clear(); // Pulisce lo schermo

 lcd.setCursor( 0, 0 );

 lcd.print( distanza );

 lcd.print("cm");

 lcd.print(" ");

 lcd.print("n=");

 lcd.print(n);

 lcd.setCursor( 0, 1 );

 lcd.print("m=");

 lcd.print(m);

 lcd.print(" ");

 lcd.print("k=");

 lcd.print(k);

 lcd.print(" ");

 lcd.print("t");

 lcd.print(temp);

 lcd.print("C");

 delay(1000);

 }

}

Controllo di un parcheggio

Si vogliono controllare il numero di auto in ingresso e quelle in uscita di un parcheggio. Il numero di auto presenti sarà dato dalla differenza tra quelle in ingresso e quelle in uscita.

Il numero di posti disponibili sarà la differenza tra Nmax, capienza del parcheggio e auto presenti.

Se il numero di posti disponibili è nullo, si accenderà un led rosso, altrimenti, sarà acceso led verde.



const int TRIG = 8;

const int ECHO= 9;

 int rosso=3;

 int verde=4;

const int TRIG1 = 6;

const int ECHO1= 7;

 int n=0, m=0, k=0;

void setup() {

 Serial.begin(9600);

 pinMode(TRIG,OUTPUT);

 pinMode(ECHO,INPUT);

 pinMode(TRIG1,OUTPUT);

 pinMode(ECHO1,INPUT);

 pinMode(verde,OUTPUT);

 pinMode(rosso,OUTPUT);

}

void loop()

{

 long durata, distanza;

 long durata1, distanza1;

 digitalWrite(TRIG, LOW);

 delayMicroseconds(2);

 digitalWrite(TRIG, HIGH);

 delayMicroseconds(10);

 digitalWrite(TRIG, LOW);

 durata = pulseIn(ECHO,HIGH);

 distanza = durata / 29.1 / 2 ;

 if (distanza <= 0){

 Serial.println("Out of range");

 }

 else {

 Serial.print(distanza);

 Serial.println("cm");

 Serial.println();

 if(distanza<50){

 if(distanza>20)m=m+1;

 }

 digitalWrite(TRIG1, LOW);

 delayMicroseconds(2);

 digitalWrite(TRIG1, HIGH);

 delayMicroseconds(10);

 digitalWrite(TRIG1, LOW);

 durata1 = pulseIn(ECHO1,HIGH);

 distanza1 = durata1 / 29.1 / 2 ;

 if (distanza1 <= 0){

 Serial.println("Out of range");

 }

 else {

 Serial.print(distanza1);

 Serial.println("cm");

 Serial.println();

 if(distanza1<50){

 if(distanza1>20)n=n+1;

 }

 }

 k=m-n;

 if(k>10){digitalWrite(rosso,HIGH);// abbiamo supposto che la capienza max sia 10

 digitalWrite(verde,LOW);}

 else{digitalWrite(verde,HIGH);

 digitalWrite(rosso,LOW);}

}}