

Programmi da scrivere per la lezione 5

### **Sketch numero 1**

**salvare il seguente programma con il nome: 18-hc-sr04-dht11**

```
/* sensore ultrasuoni HC-SR04
 * programma di test funzionamento
 * distanza= (tempo segnale echo high/2) *velocità del suono ((331,45 + (0,62 * T)))
 * echo collegato al pin 9
 * trig collegato al pin 11
 * Aggiunto sensore DHT11 - temperatura ed umidità
 * Signal collegato al pin 2
 * Per usare questo sensore in modo semplice dobbiamo includere
 * nel nostro programma una libreria con il codice preparato da altri per questo scopo
 * il nome di questa libreria è dht.h
 */
#include <dht.h>
// ora creiamo l'oggetto DHT cioè un oggetto che eredita tutte le
// proprietà (metodi e funzioni) scritte nella libreria dht
dht DHT;

//definiamo i pin a cui siamo collegati
const int pindht = 2;
const int trigPin = 11;
const int echoPin = 9;

//===== spazio variabili =====
// variabili usate per durata impulso echo e calcolo centimetri
float durata, cm;
// definisco una variabile per memorizzare il passare del tempo
unsigned long tempo;
//definisco l'intervallo minimo tra due letture del sensore di temperatura
int intervallo = 10000;
//variabile temperatura e umidità
float miaTemp = 0.0;
float miaUmid = 0.0;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);
  pinMode(pindht, INPUT);
  // tempo è uguale ad adesso!
  tempo = millis();
  Serial.println("HT-SR04 + DHT11 \t Inizializzazione");
  delay(2000);
}

void loop() {
  //controllo se è ora di interrogare il sensore temperatura
  if (( millis() - tempo )>= intervallo) {
    int chk = DHT.read11(pindht);
```

```

switch (chk){
case DHTLIB_OK:
    break;
case DHTLIB_ERROR_CHECKSUM:
    Serial.print("Checksum error,\t");
    break;
case DHTLIB_ERROR_TIMEOUT:
    Serial.print("Time out error,\t");
    break;
default:
    Serial.print("Unknown error,\t");
    break;
}
miaTemp = DHT.temperature;
miaUmid = DHT.humidity;
Serial.println("");
Serial.print("umidità = ");
Serial.print(miaUmid);
Serial.print("%\t");
Serial.print("temperatura = ");
Serial.print(miaTemp);
Serial.println(" °C");
Serial.println("");
// aggiorno il tempo con adesso!
tempo = millis();
}
//richiesta invio impulsi al sensore
digitalWrite(trigPin,LOW);
delayMicroseconds(2);
digitalWrite(trigPin,HIGH);
delayMicroseconds(10);
digitalWrite(trigPin,LOW);

//restituisce il tempo di echo
durata=pulseIn(echoPin,HIGH);

//trasformo in cm
cm = durata / 20000.0 *(331.45 + (0.62 * miaTemp));
//scrivo sul monitor seriale
Serial.print("Ostacolo rilevato a ");
Serial.print(cm);
Serial.println(" cm di distanza");
delay(1000);
}

```

---

## Sketch numero 2

salvare il seguente programma con il nome: 19-hc-sr04-dht11Buz

```
/* sensore ultrasuoni HC-SR04
 * programma di test funzionamento
 * distanza= (tempo segnale echo high/2) *velocità del suono ((331,45 + (0,62 * T)))
 * echo collegato al pin 9
 * trig collegato al pin 11
 * Aggiunto sensore DHT11 - temperatura ed umidità
 * Signal collegato al pin 2
 * Per usare questo sensore in modo semplice dobbiamo includere
 * nel nostro programma una libreria con il codice preparato da altri per questo scopo
 * il nome di questa libreria è dht.h
 * Aggiungiamo un buzzer attivo al pin 12
 */

//aggiungiamo le librerie che ci servono
#include <dht.h>

// ora creiamo l'oggetto DHT cioè un oggetto che eredita tutte le
// proprietà (metodi e funzioni) scritte nella libreria
dht DHT;

//definiamo i pin a cui siamo collegati
const int pindht = 2;
const int trigPin = 11;
const int echoPin = 9;
const int buzzer = 12;
//===== spazio variabili =====
int buzzerOn = 0;      // tempo di suono buzzer
int buzzerOff = 0;    // tempo di pausa buzzer

float durata;         // variabile per durata impulso echo
float cm = 200.0;     // variabile per calcolo centimetri
unsigned long tempoD; // tempo e intervallo per il sensore distanza
int intervalloD = 1000;

unsigned long tempoT; // definisco una variabile per memorizzare il passare del tempo
int intervalloT = 10000; //definisco l'intervallo minimo tra due letture del sensore di temperatura
float miaTemp = 0.0; //variabili temperatura e umidità
float miaUmid = 0.0;

// Creo una funzione che mi permette di far suonare con frequenze diverse il buzzer
// suonaBuzzer(pin, tempo_on, tempo_off)
void suonaBuzzer (int myPin, int myTOn, int myTOff) {
    digitalWrite(myPin,HIGH);
    delay(myTOn);
    digitalWrite(myPin, LOW);
    delay(myTOff);
}

void setup() {
```

```

//inializzazione lcd
lcd.begin(16, 2);
//inializzazione porta seriale
Serial.begin(9600);
pinMode(trigPin,OUTPUT);
pinMode(echoPin,INPUT);
pinMode(pindht, INPUT);
pinMode(buzzer, OUTPUT);

// tempo è uguale ad adesso!
tempoT = millis();
tempoD = millis();
Serial.println("HT-SR04 + DHT11 \t Inizializzazione");
delay(2000);
}

void loop() {
//controllo se è ora di interrogare il sensore temperatura
if (( millis() - tempoT )>= intervalloT) {
    int chk = DHT.read11(pindht);
    switch (chk){
    case DHTLIB_OK:
        break;
    default:
        Serial.print("Unknown error,\t");
        break;
    }
    miaTemp = DHT.temperature;
    miaUmid = DHT.humidity;

    //scrivo i valori sul monitor
    Serial.print("Temp: ");
    Serial.print(miaTemp,1);
    Serial.print(" C ");
    Serial.print(" Umidita': ");
    Serial.print(miaUmid,1);
    Serial.println(" %");

    // aggiorno il tempo con adesso!
    tempoT = millis();
}
//controllo se è ora di interrogare il sensore distanza
if ((millis() - tempoD) >= intervalloD) {
    //richiesta invio impulsi al sensore
    digitalWrite(trigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);

    //restituisce il tempo di echo
    durata=pulseIn(echoPin,HIGH);
}
}

```

```

//trasformo in cm
cm = durata / 20000.0 *(331.45 + (0.62 * miaTemp));

//scrivo sul monitor
Serial.print("Ostacolo rilevato a: ");
Serial.print(cm);
Serial.print(" cm di distanza");

// aggiorno il tempo con adesso!
tempoD = millis();
}
if (cm < 100 && cm >90) {
  buzzerOn = 200;
  buzzerOff = 200;
}
else if (cm < 91 && cm > 80) {
  buzzerOn = 180;
  buzzerOff = 180;
}
else if (cm < 81 && cm > 70) {
  buzzerOn = 170;
  buzzerOff = 170;
}
else if (cm < 71 && cm > 60) {
  buzzerOn = 150;
  buzzerOff = 150;
}
else if (cm < 61 && cm > 50) {
  buzzerOn = 130;
  buzzerOff = 130;
}
else if (cm < 51 && cm > 40) {
  buzzerOn = 90;
  buzzerOff = 90;
}
else if (cm < 41 && cm > 30) {
  buzzerOn = 60;
  buzzerOff = 60;
}
else if (cm < 31 && cm > 20) {
  buzzerOn = 30;
  buzzerOff = 30;
}
else if (cm < 21 && cm > 10) {
  buzzerOn = 10;
  buzzerOff = 10;
}
else if (cm < 11 ) {
  buzzerOn = 5;
  buzzerOff = 5;
}
}

```

```

    if (cm < 101) {
        suonaBuzzer (buzzer, buzzerOn, buzzerOff);
    }
}

```

---

### **Sketch numero 3**

**salvare il seguente programma con il nome: 20-hc-sr04-dht11BuzLcd**

```

/* sensore ultrasuoni HC-SR04
 * programma di test funzionamento
 * distanza= (tempo segnale echo high/2) *velocità del suono ((331,45 + (0,62 * T)))
 * echo collegato al pin 9
 * trig collegato al pin 11
 * Aggiunto sensore DHT11 - temperatura ed umidità
 * Signal collegato al pin 2
 * Per usare questo sensore in modo semplice dobbiamo includere
 * nel nostro programma una libreria con il codice preparato da altri per questo scopo
 * il nome di questa libreria è dht.h
 * Aggiungiamo un buzzer attivo al pin 12
 * Aggiungiamo lo schermo LCD
 */
//aggiungiamo le librerie che ci servono
#include <dht.h>
#include <LiquidCrystal.h>

//creiamo l'oggetto lcd e gli assegnamo i pin collegati
//          BS E  D4 D5 D6 D7
LiquidCrystal lcd(3, 4, 5, 6, 7, 8);

// ora creiamo l'oggetto DHT cioè un oggetto che eredita tutte le
// proprietà (metodi e funzioni) scritte nella libreria
dht DHT;

//definiamo i pin a cui siamo collegati
const int pindht = 2;
const int trigPin = 11;
const int echoPin = 9;
const int buzzer = 12;
//===== spazio variabili =====
int buzzerOn = 0;      // tempo di suono buzzer
int buzzerOff = 0;    // tempo di pausa buzzer

unsigned long tempoD;  // tempo e intervallo per il sensore distanza
int intervalloD = 1000;
float durata;         // variabile per durata impulso echo
float cm = 200.0;     // variabile per calcolo centimetri

unsigned long tempoT; // tempo e intervallo per il sensore temperatura
int intervalloT ;    // attenzione intervalloT definito nel setup()
float miaTemp = 0.0; // variabili temperatura e umidità
float miaUmid = 0.0;

// Creo una funzione che mi permette di far suonare con frequenze diverse il buzzer
// suonaBuzzer(pin, tempo_on, tempo_off)
void suonaBuzzer (int myPin, int myTOn, int myTOff) {
    digitalWrite(myPin,HIGH);
    delay(myTOn);
    digitalWrite(myPin, LOW);
    delay(myTOff);
}

```

```

}

void checkTemp() {
  if (( millis() - tempoT )>= intervalloT) {
    int chk = DHT.read11(pindht);

    miaTemp = DHT.temperature;
    miaUmid = DHT.humidity;
    //scrivo i valori sulla prima riga dell'lcd
    lcd.setCursor(0, 0);
    lcd.print("T:  C U:  %");
    lcd.setCursor(2, 0);
    lcd.print(miaTemp,1);
    lcd.setCursor(11, 0);
    lcd.print(miaUmid,1);

    tempoT = millis();          //aggiorno il tempo con adesso!
  }
}

void checkDist() {
  if ((millis() - tempoD) >= intervalloD) {
    digitalWrite(trigPin,LOW);          //richiesta invio impulsi al sensore
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);
    durata=pulseIn(echoPin,HIGH);      //restituisce il tempo di echo

    cm = durata / 20000.0 *(331.45 + (0.62 * miaTemp)); //trasformo in cm

    lcd.setCursor(0, 1);          //scrivo il valore sulla seconda riga dell'lcd
    lcd.print("Dist.:  cm");
    lcd.setCursor(6, 1);
    lcd.print(cm);

    tempoD = millis();          // aggiorno il tempo con adesso!
  }
}

void checkBuzzer() {
  if (cm < 100 && cm >90) {
    buzzerOn = 200;
    buzzerOff = 200;
  }
  else if (cm < 91 && cm > 80) {
    buzzerOn = 180;
    buzzerOff = 180;
  }
  else if (cm < 81 && cm > 70) {
    buzzerOn = 170;
    buzzerOff = 170;
  }
  else if (cm < 71 && cm > 60) {
    buzzerOn = 150;
    buzzerOff = 150;
  }
  else if (cm < 61 && cm > 50) {
    buzzerOn = 130;
    buzzerOff = 130;
  }
  else if (cm < 51 && cm > 40) {
    buzzerOn = 90;
    buzzerOff = 90;
  }
}

```

```

else if (cm < 41 && cm > 30) {
    buzzerOn = 60;
    buzzerOff = 60;
}
else if (cm < 31 && cm > 20) {
    buzzerOn = 30;
    buzzerOff = 30;
}
else if (cm < 21 && cm > 10) {
    buzzerOn = 10;
    buzzerOff = 10;
}
else if (cm < 11 ) {
    buzzerOn = 5;
    buzzerOff = 5;
}
if (cm < 101) {
    suonaBuzzer (buzzer, buzzerOn, buzzerOff);
}
}

void setup() {
    //inializzazione lcd
    lcd.begin(16, 2);
    pinMode(trigPin,OUTPUT);
    pinMode(echoPin,INPUT);
    pinMode(pindht, INPUT);
    pinMode(buzzer, OUTPUT);

    // tempo è uguale ad adesso!
    tempoT = millis();
    tempoD = millis();
    delay(2000);

    intervalloT = 1000; //imposto un intervallo corto per eseguire la prima lettura temperatura
    checkTemp();
    intervalloT = 10000; //rimetto intervallo originale
}

void loop() {
    checkTemp(); //controllo se è ora di interrogare il sensore temperatura
    checkDist(); //controllo se è ora di interrogare il sensore distanza
    checkBuzzer(); //controllo la frequenza del buzzer
}

```