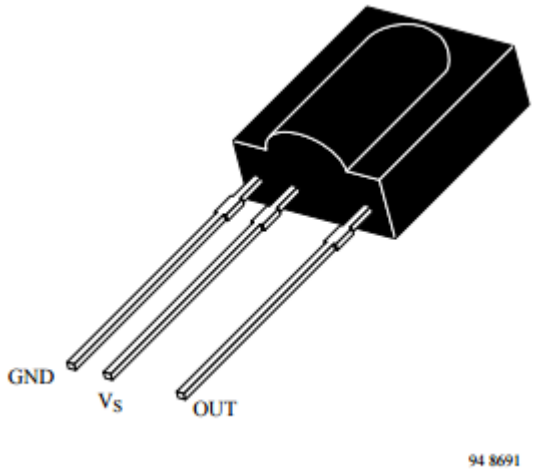
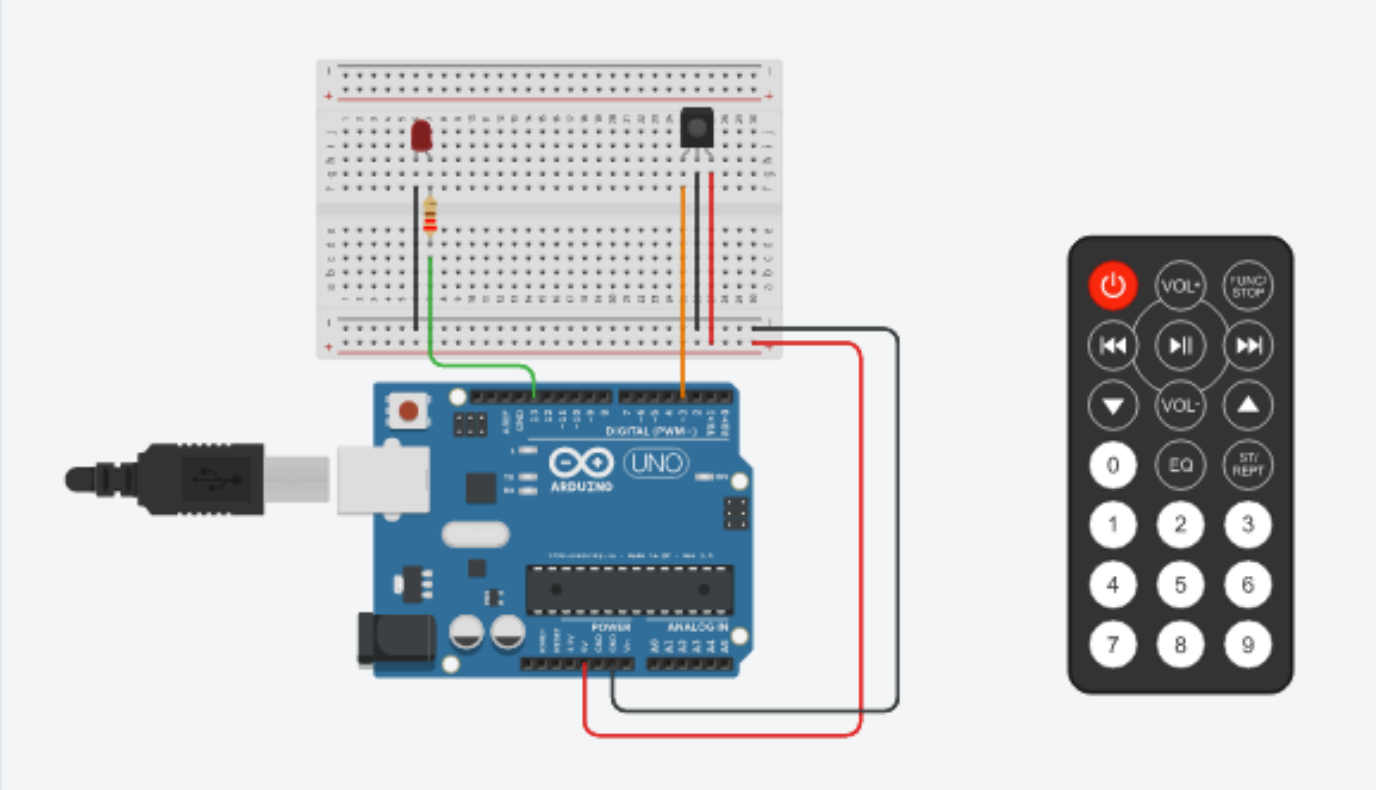
Arduino lettore e trasmettitore di codici IR

In questo tutorial decodificheremo i segnali provenienti da un telecomando qualsiasi e utilizzeremo gli stessi segnali per comandare Arduino a distanza per esempio pilotare i relè o accendere un led.

Utilizzeremo un TOSP1738 e la libreria IR.remote.h che nella cartella "Libraries" di Arduino.



Arduino lettore e trasmettitore di codici IR



SKETCH

Per prima cosa decodificheremo i segnali provenienti dal telecomando utilizzando questo sketch:

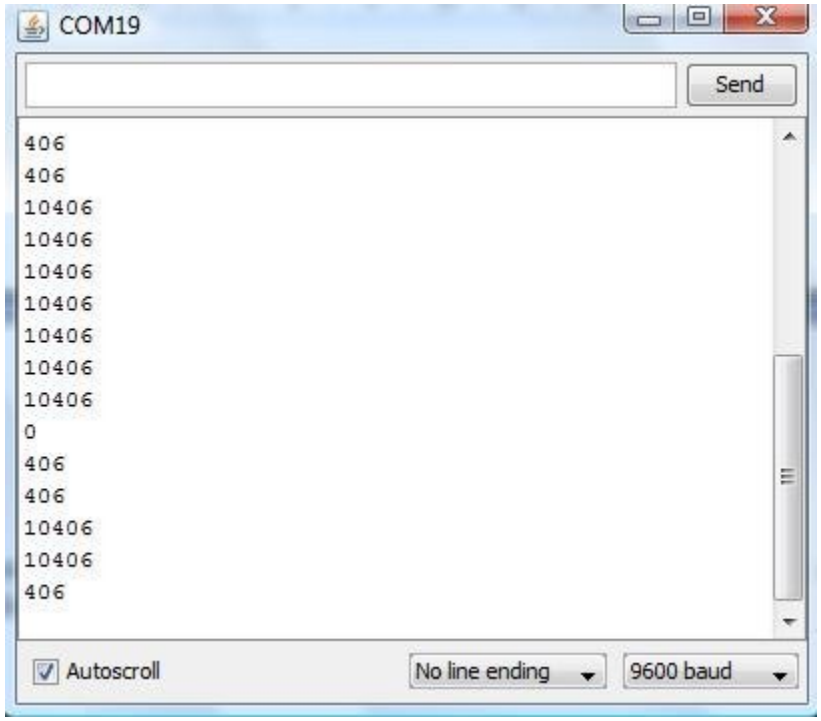
```
// #include <IRremoteInt.h>
#include <IRremote.h>  // use the library
int receiver = 11;           // pin 1 of IR receiver to Arduino digital pin 11
IRrecv irrecv(receiver);  // create instance of 'irrecv'
decode_results results;

void setup()
{  Serial.begin(9600);       // for serial monitor output
   irrecv.enableIRIn();       // Start the receiver
    pinMode(9, OUTPUT); // Pin 9 output
}

void loop()
{  if (irrecv.decode(&results)) // have we received an IR signal?
    {Serial.println(results.value, HEX); // display it on serial monitor in hexadecimal
      irrecv.resume(); // receive the next value
    }
 }
```

Primo passo)

Carichiamo e apriamo il serial monitor;

Secondo passo)

Premiamo un pulsante sul telecomando e leggiamo il valore che appare sulla finestra del serial monitor (schiacciando il tasto 6 sulla finestra appare 406 o 10406):



Per utilizzare i valori scritti nel serial monitor basta aggiungere **0x** prima della sigla del valore nello sketch. In questo caso faremo svolgere ad Arduino una funzione, come per esempio fare accendere un led.

```
#include <IRremoteInt.h>
#include <IRremote.h> // use the library
int receiver = 11;          // pin 1 of IR receiver to Arduino digital pin 11
IRrecv irrecv(receiver);  // create instance of 'irrecv'
decode_results results;   //decodifico il risultato che Arduino riceve dal sensore

void setup()
{  Serial.begin(9600);       // for serial monitor output
   irrecv.enableIRIn();       // Start the receiver
   pinMode(9, OUTPUT); // Pin 9 output
}
void loop(){
if (irrecv.decode(&results)) // have we received an IR signal?
{    Serial.println(results.value, HEX);  // display it on serial monitor in hexadecimal
      irrecv.resume();             // receive the next value
 }
if ( results.value == 0x406 ||  results.value == 0x10406 )
   { //tasto 6 sul telecomando
     digitalWrite(9, HIGH);   // set the LED on
   }
if ( results.value == 0x404 ||  results.value == 0x10404 )
   { //tasto 4 sul telecomando
     digitalWrite(9, LOW);   // set the LED 0ff
    }
 }
```

SPIEGAZIONE

linea 1: includo la libreria **IRemote** nel mio sketch

linea2: dichiaro che il pin 11 è il “receiver” ovvero il pin da cui la libreria legge il segnale dal sensore, il pin 11 viene collegato al piedino OUT del sensore TSOP1738:

linea3: inizializzo la libreria IRemote

linea 4: decodifico il risultato che Arduino riceve dal sensore per ottenere un valore numerico utilizzabile nel confronto.

linea6: costruisco la funzione setup() che, come sai, è fondamentale per Arduino.

linea7: attivo la comunicazione seriale per vedere i valori letti dal sensore sulla Serial Monitor di Arduino quando premo un tasto sul telecomando

linea8: utilizzo il metodo **irrecv.enableIRIn();** della libreria IRemote perchè legga i valori provenineti dal TSOP1738

linea9: imposto, mediante il comendo pinMode(9, OUTPUT), l’utilizzo del pin 9 di arduino come output per pilotare il led

linee13-17: quando il sensore riceve un segnale proveninete dal mio telecomando scrive sul monitor seriale il valore rilevato **Serial.println(results.value, HEX);** e prosegue mettendo il sensore nuovamente n modalità di ascolto: **irrecv.resume();**

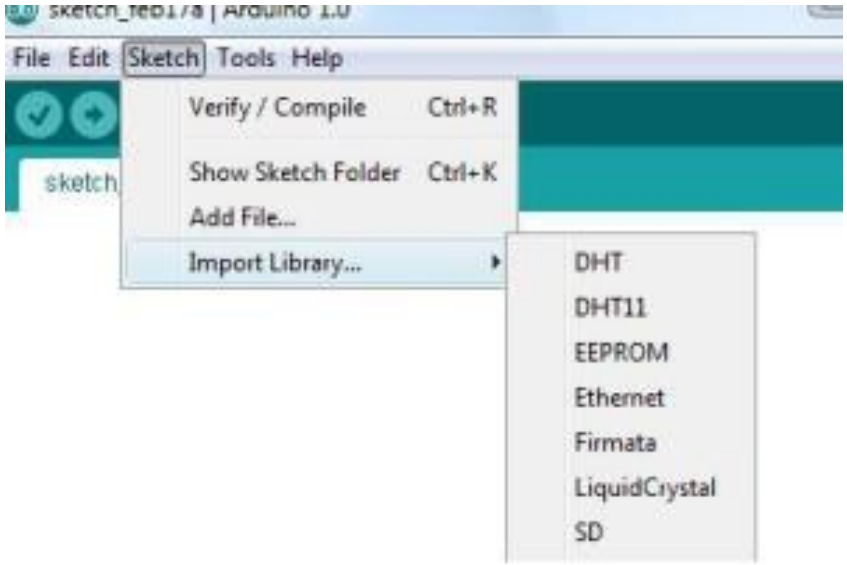
linee19-21: premo il **tasto 6,** il valore letto del sensore è uguale a 406 o 10406, quindi porto l’uscita del pin 9 ad HIGH (1) ed il LED si accend

linee22-24: premo il **tasto 4** , il valore letto del sensore è ugua ,
quindi porto l’uscita del pin 9 a LOW (0) ed il LED si spegne.

TSOP1738

Installare una nuova libreria

Le librerie di Arduino sono un insieme di funzioni predisposte per essere richiamate all'interno di un programma.
La libreria "LiquidCrystal" ad esempio, consente ad Arduino di controllare un display basato su chip Hitachi HD44780 (o compatibile) con appena 6 righe di codice.



Esempio

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

 void setup()
   { lcd.print("hello, world!"); }
 void loop() {}
```

Installare una nuova libreria

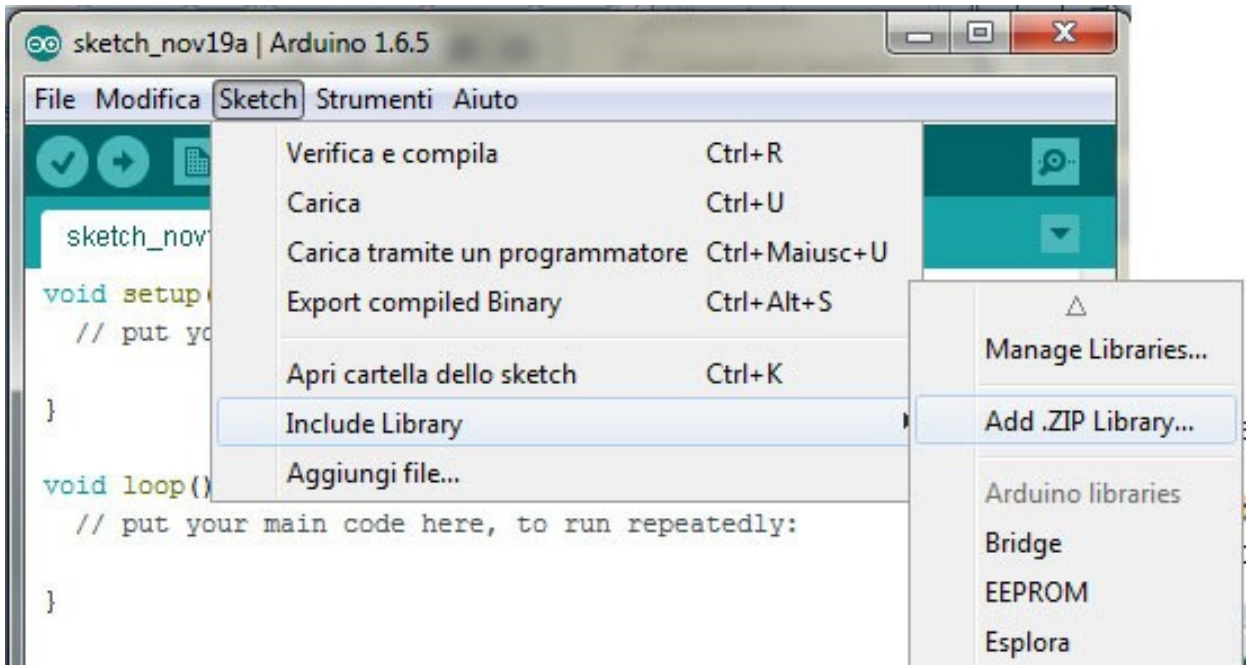
Una nuova libreria ha bisogno di essere installata per funzionare correttamente, e per fare cio è sufficiente copiare la cartella contenente i due file **.cpp** e **.h** all'interno della cartella "**libraries**" del compilatore Arduino. Una volta riavviato il compilatore siamo pronti ad utilizzare tutte le funzionalità della nuova libreria dopo averla caricata il comando

#include <nome_libreria.h>

Andiamo quindi su internet a cercare la libreria che ci interessa, nel nostro caso si trova qui .Nella sezione release possiamo scegliere l'ultima versione stabile. La struttura di una libreria è una cartella che contiene almeno un file .cpp e un file .h per ogni libreria.
In genere contiene anche una cartella "examples", che contiene degli esempi.

Installare una nuova libreria

Per aggiungere la libreria Arduino-IRremote-master.zip, non dobbiamo far altro che che aggiungerla nella cartella libraries, che si trova nella stessa cartella dove abbiamo salvato Arduino selezionando il comando **Sketch->Include Library->Add.Zip Library**



Riavviamo Arduino e lui la riconoscerà, e ne aggiungerà gli esempi alla lista degli esempi.