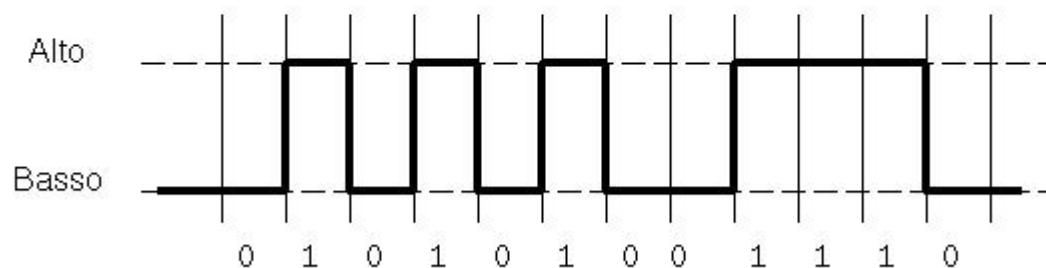


Come funziona una trasmissione ad infrarossi ?

Il meccanismo di funzionamento di una trasmissione ad infrarossi è abbastanza “semplice”; un fascio luminoso, ad elevata frequenza 38 KHz (per i telecomandi tipici per la TV) viene proiettato contro un ricevente. Questo fascio luminoso non è altro che una sequenza di bit (0 e 1) che vengono codificati dal ricevente, attraverso la polarizzazione di photo-transistor. Facciamo questo semplice caso; nel nostro telecomando, vengono codificati i vari pulsanti con varie sequenze bit e per esempio il tasto on/off vale:

010101001110



Il ricevente avrà un segnale in ricezione molto simile, se non uguale a quello inviato e in base alla corrispondenza che ha in memoria, sarà in grado di svolgere la richiesta effettiva, come ad esempio accendere la TV ! Il fascio luminoso che viene inviato dal telecomando, non è possibile vederlo ad occhio nudo perché ha una frequenza troppo veloce per noi !

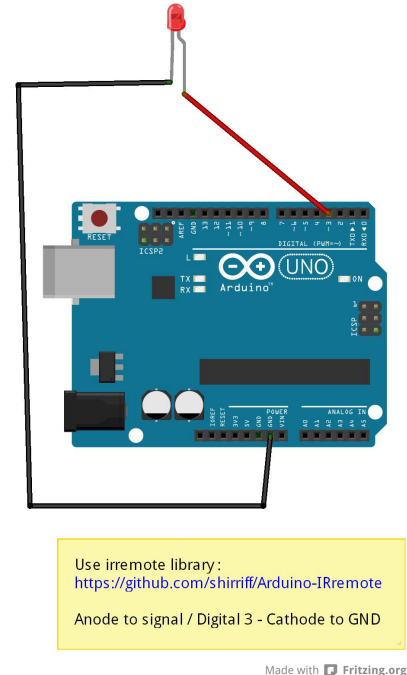
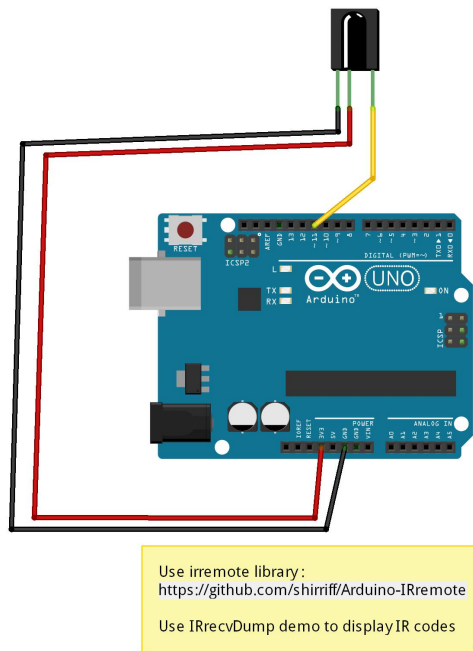
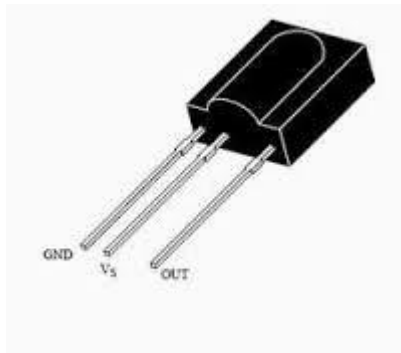
Tuttavia, ci sono diverse codifiche; tipicamente ogni produttore di dispositivi tecnologici, ha una propria codifica, come ad esempio Sony, NEC, Samsung etc..

Tuttavia, è possibile stabilire la sequenza di bit di qualsiasi segnale ad infrarosso, calcolando il tempo in cui il segnale è alto e il tempo in cui il segnale è basso. Questo tipo di codifica si chiama RAW.

Passiamo ai codici per Arduino

Come per ogni libreria nuova in Arduino, dobbiamo installarla, copiandola semplicemente nella cartella Libraries, che tipicamente si trova nei propri documenti.

Qualora utilizzassimo il dispositivo TSOP1738, il collegamento è davvero semplice; basterà alimentare il segnale con i 5 V di Arduino, collegarlo a massa e collegare il pin di output al pin 11.



[TSOP1738 Datasheet](#)

Per quanto riguardo altri dispositivi presenti su Ebay, basterà collegare il pin di output sempre al pin 11 di Arduino.

1° Programma decodifichiamo un generico segnale

Nome programma Decode_generic_IR

```

1 /*
2
3  * Modified by Chris Targett
4  * Now includes more protocols
5  * Novemeber 2011
6
7  * IRremote: IRrecvDump - dump details of IR codes with IRrecv
8  * An IR detector/demodulator must be connected to the input RECV_PIN.
9  * Version 0.1 July, 2009
10 * Copyright 2009 Ken Shirriff

```

```

11 * http://arcfn.com
12 *
13 * Modified by Chris Targett to speed up the process of collecting
14 * IR (HEX and DEC) codes from a remote (to put into and .h file)
15 *
16 */
17
18#include <IRremote.h>
19
20int RECV_PIN = 11;
21
22IRrecv irrecv(RECV_PIN);
23
24decode_results results;
25
26void setup()
27{
28    Serial.begin(9600);
29    irrecv.enableIRIn(); // Start the receiver
30}
31
32// Dumps out the decode_results structure.
33// Call this after IRrecv::decode()
34// void * to work around compiler issue
35//void dump(void *v) {
36//    decode_results *results = (decode_results *)v
37void dump(decode_results *results) {
38    int count = results->rawlen;
39    if (results->decode_type == UNKNOWN) {
40        Serial.print("Unknown encoding: ");
41    }
42    else if (results->decode_type == NEC) {
43        Serial.print("Decoded NEC: ");
44    }
45    else if (results->decode_type == SONY) {
46        Serial.print("Decoded SONY: ");
47    }
48    else if (results->decode_type == RC5) {
49        Serial.print("Decoded RC5: ");
50    }
51    else if (results->decode_type == RC6) {
52        Serial.print("Decoded RC6: ");
53    }
54    else if (results->decode_type == SAMSUNG) {

```

```

55     Serial.print("Decoded SAMSUNG: ");
56 }
57 else if (results->decode_type == JVC) {
58     Serial.print("Decoded JVC: ");
59 }
60 else if (results->decode_type == PANASONIC) {
61     Serial.print("Decoded Panasonic: ");
62 }
63 Serial.print(results->value, HEX);
64 Serial.print("");
65 Serial.print(results->bits, DEC);
66 Serial.println(" bits");
67 Serial.print("#define Something_DEC ");
68 Serial.println(results->value, DEC);
69 Serial.print("#define Something_HEX ");
70 Serial.println(results->value, HEX);
71 Serial.print("Raw ");
72 Serial.print(count, DEC);
73 Serial.print("): ");
74 for (int i = 0; i < count; i++) {
75     if ((i % 2) == 1) {
76         Serial.print(results->rawbuf[i]*USECPERTICK, DEC);
77     }
78     else {
79         Serial.print(-(int)results->rawbuf[i]*USECPERTICK,
80DEC);
81     }
82     Serial.print(" ");
83 }
84 Serial.println("");
85}
86
87void loop() {
88    if (irrecv.decode(&results)) {
89        dump(&results);
90        irrecv.resume(); // Receive the next value
91    }
92 }

```

Il programma qui presente mostra sul seriale la codifica del segnale ad infrarosso che viene ricevuto dal ricevitore, collegato al pin 11 di Arduino. Il meccanismo è davvero abbastanza chiara, grazie ai commenti presenti nel codice, tratto dalla libreria ufficiale.

2° Programma per Arduino: Accendiamo una lampada con qualsiasi segnale ad Infrarosso

```
1  /**
2  Questo programme permette di accendere una lampada, attraverso un
3  relay, con un
4  telecomando ad infrarossi. Questa versione non permette di modificare
5  il codice
6  a run time
7  Autore Giacomo Bellazzi
8  */
9
10 #include <IRremote.h>
11
12 int RECV_PIN = 11;
13 int RELAY_PIN = 13;
14 int codice = 16738455; // codice dec del telecomando
15
16 IRrecv irrecv(RECV_PIN);
17 decode_results results;
18
19 void setup()
20 {
21     pinMode(RELAY_PIN, OUTPUT);
22
23     Serial.begin(9600);
24     irrecv.enableIRIn(); // Start the receiver
25 }
26
27 int on = 0;
28 unsigned long last = millis();
29
30 void loop() {
31     if (irrecv.decode(&results)){
32         // If it's been at least 1/4 second since the last
33         // IR received, toggle the relay
34         if (millis() - last > 250 && results.value==codice ) {
35             on = !on;
36             digitalWrite(RELAY_PIN, on ? HIGH : LOW);
37             //dump(&results);
38         }
39         last = millis();
40         irrecv.resume(); // Receive the next value
41     }
```

}

Questo codice permette di accendere una lampada connessa ad un relè, premendo un telecomando ad infrarossi. La codifica del telecomando che si vuole usare, può essere ottenuta con il codice del primo programma e inserita all'inizio di questo codice. Tuttavia, il codice non è modificabile a run time.