Il telecomando a infrarossi: comandare Arduino dal divano

Un ricevitore a 38 kHz e la libreria IRremote: leggi i codici di qualsiasi telecomando, anche quello della TV, e con i tasti accendi un LED, cambi colore a un LED RGB e muovi un servo.

Guida online, sempre aggiornata

A cura di **Ada Intelisano** · redazione tecnica di informaticait.it

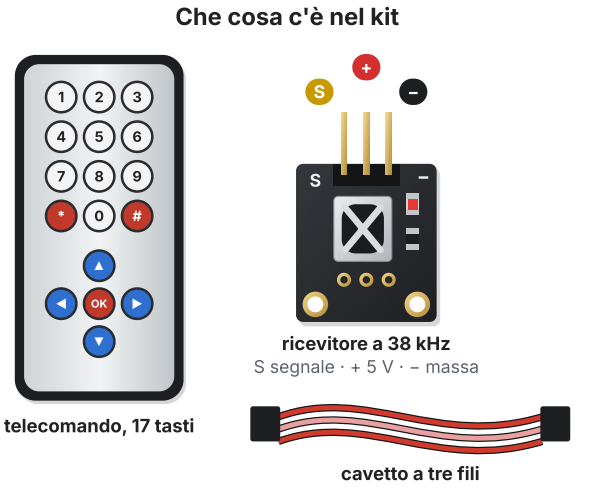
IN BREVE

- Il ricevitore ha tre pin: S al pin 2, il + a 5 V, il – a GND.
- La libreria IRremote trasforma ogni tasto in protocollo, indirizzo e comando.
- Il primo programma legge i comandi di qualsiasi telecomando: li copi negli altri.
- Sulla UNO IRremote occupa il PWM dei pin 3 e 11, la libreria Servo quello dei pin 9 e 10.

Tempo: 50 minuti | Livello: principiante | Prima: guide n. 34, 41 e 46

Ogni telecomando parla con una luce che i nostri occhi non vedono. In questa guida Arduino impara ad ascoltarla. Leggerai i codici dei tasti di un telecomando qualsiasi, anche quello della TV, e li userai per accendere un LED, cambiare colore a un LED RGB e girare il braccetto di un servo.

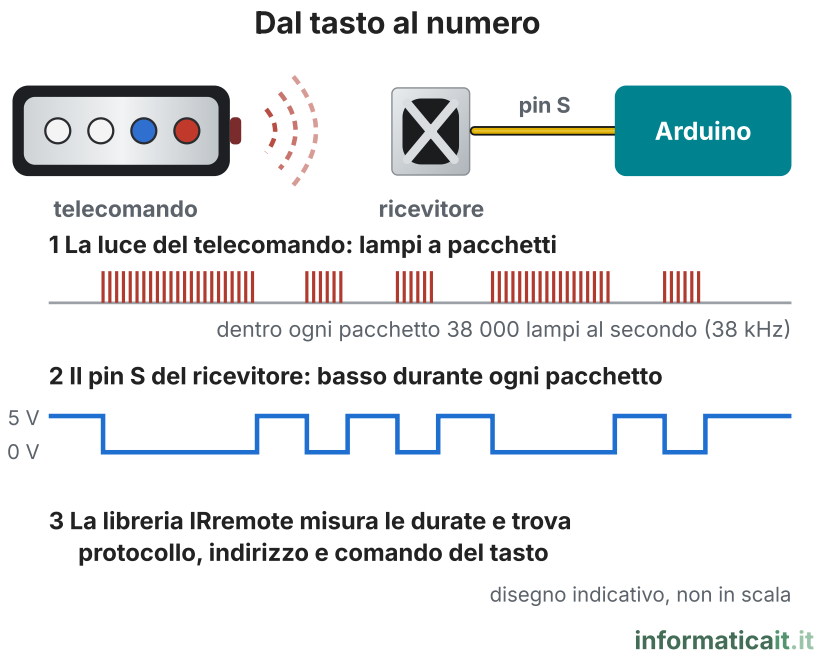
1 Prepara il materiale



- il **kit telecomando con ricevitore** a 38 kHz: telecomando, ricevitore su una schedina e un cavetto a tre fili
- Arduino UNO (va bene la scheda compatibile) con il suo cavo USB
- una breadboard da 400 punti, cavetti maschio-femmina e maschio-maschio
- dal passo 8: il modulo **LED RGB KY-009**, tre resistenze da **330 Ω** e un **micro servo SG90**

Parola nuova: infrarossi
Una luce di un colore che l'occhio umano non vede. Il LED del telecomando la accende e la spegne molto in fretta, a pacchetti: ogni tasto ha la sua sequenza.

2 Come viaggia il segnale



Dentro il ricevitore ci sono un sensore di luce, un amplificatore e un filtro sintonizzato a **38 kHz**: tiene la luce che lampeggia 38 000 volte al secondo e scarta quella fissa del sole o delle lampadine. Sul pin **S** arrivano già i pacchetti puliti: a riposo il pin è alto, durante ogni pacchetto scende a 0 V. Il resto lo fa la libreria **IRremote**: misura quanto durano pacchetti e pause e ne ricava tre informazioni.

> **Parola nuova: protocollo, indirizzo, comando**
> Il **protocollo** è la regola con cui un telecomando trasforma un tasto in lampi: i più diffusi hanno un nome, come NEC, Sony, RC5 o Samsung. L'**indirizzo** spesso è lo stesso per tutti i tasti di un telecomando. Il **comando** cambia da un tasto all'altro: è il numero che useremo.

3 Installa la libreria IRremote

Nell'IDE di Arduino apri il **gestore librerie**, cerca **IRremote** e installa quella di **shirriff, z3t0, ArminJo**. I nostri programmi sono provati con la versione 4.7.1.

Attenzione ai vecchi esempi che trovi in rete: se contengono `irrecv.decode(&results)` sono scritti per la versione 2 della libreria e con quella attuale non funzionano.

4 Collega il ricevitore

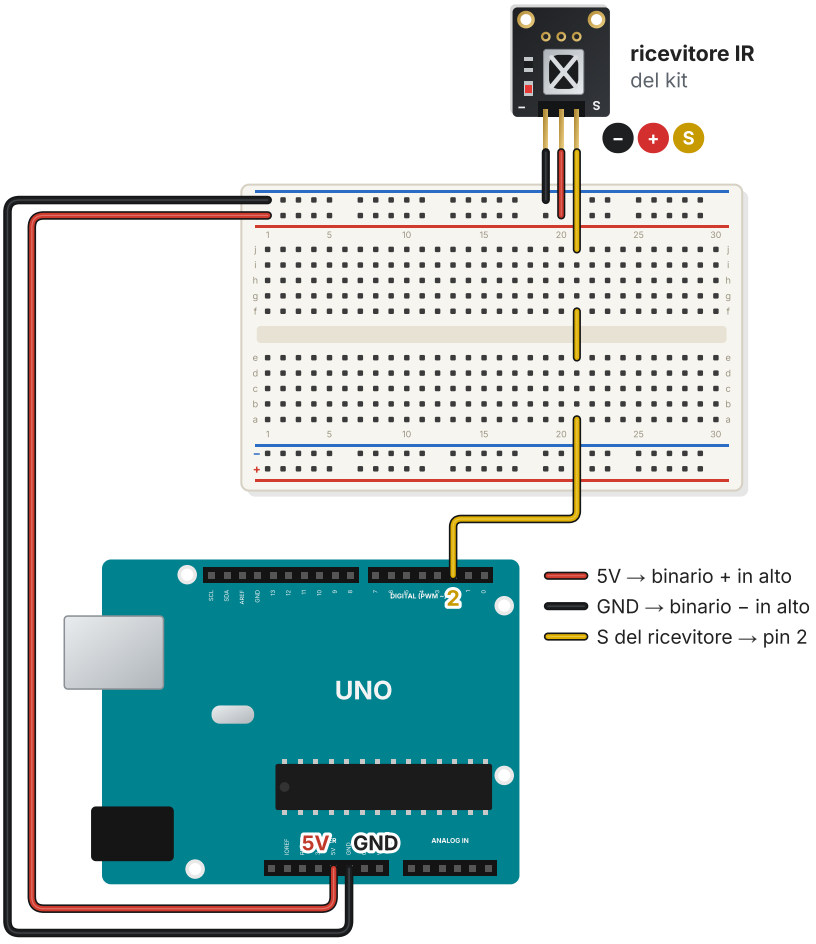
La breadboard serve a distribuire i 5 V e la massa: tra poco li useranno tre pezzi. Segui il disegno qui sotto.

- **5V** della UNO → fila **+** in alto della breadboard (filo rosso), **GND** → fila **–** in alto (filo nero). I due fili girano a sinistra della scheda.

- Il ricevitore resta fuori dalla breadboard, collegato con tre cavetti maschio-femmina: **–** alla fila −, il pin centrale (**+**) alla fila +, **S** nella colonna 21, fila j.
- Un ponticello giallo da **f** a **e** nella colonna 21 porta il segnale nella metà bassa. Dalla fila a della stessa colonna un cavetto va al **pin 2**.

Sulla schedina sono stampati **S** e **–**; il pin in mezzo, senza scritta, è il +. Controlla le scritte del tuo modulo prima di collegare: l'ordine dei pin cambia da un produttore all'altro.

Il ricevitore sulla breadboard



informaticait.it

✓ Fatto?
Tre fili dal ricevitore: S arriva al pin 2, il + alla fila +, il – alla fila −. La fila + prende il 5V, la fila – il GND.

5 Leggi i codici dei tasti

```
// Premi un tasto del telecomando: Arduino scrive il suo codice nel monitor seriale.
#include <Arduino.h>
#include <IRremote.hpp>      // la libreria IRremote

const int PIN_IR = 2;        // il pin S del ricevitore
```

```
void setup() {
  Serial.begin(115200);
  IrReceiver.begin(PIN_IR, ENABLE_LED_FEEDBACK);   // il LED L lampeggia a ogni segnale
  Serial.println("Pronto: premi un tasto del telecomando");
}

void loop() {
  if (IrReceiver.decode()) {                       // è arrivato un segnale?
    IRData &dati = IrReceiver.decodedIRData;       // quello che la libreria ha capito
    if (dati.flags & IRDATA_FLAGS_IS_REPEAT) {
      Serial.println("  ...tasto tenuto premuto");
    } else if (dati.protocol == UNKNOWN) {
      Serial.println("Segnale non riconosciuto: riprova");
    } else {
      Serial.print("Protocollo: ");
      Serial.print(getProtocolString(dati.protocol));
      Serial.print("   indirizzo: 0x");
      Serial.print(dati.address, HEX);
      Serial.print("   comando: 0x");
      Serial.println(dati.command, HEX);
    }
    IrReceiver.resume();                           // pronto per il prossimo tasto
  }
}
```

Carica il programma, apri il monitor seriale a **115200** e punta il telecomando verso il ricevitore. Premi un tasto alla volta, senza fretta.

✓ Cosa devi vedere
A ogni tasto compare una riga, con il nome del protocollo e due numeri: `indirizzo: 0x...` e `comando: 0x...` (i numeri dipendono dal tuo telecomando). Il LED L della scheda lampeggia quando arriva la luce. Se tieni premuto compare `...tasto tenuto premuto`.

Parola nuova: esadecimale
`0x10` è un numero scritto in base 16: lo `0x` davanti lo dice. Non serve fare conti: copialo così com'è, con lo 0x, nella tabella qui sotto.

Scrivi qui i tuoi codici (ti servono nei passi 7, 9 e 10):

Tasto	Comando del tuo telecomando	Mini telecomando Adafruit 389
1		0x10
2		0x11
3		0x12
0		0x0C
freccia a sinistra		0x08
freccia a destra		0x0A
OK (sul 389 si chiama Enter)		0x09

Colonna Adafruit 389: comandi ricavati da noi dai codici che Adafruit pubblica per questo telecomando. Sono i numeri già scritti nei programmi.

Se non funziona

- Nessuna riga: controlla il filo S sul pin 2, il + e il – sulle file giuste e la velocità 115200 in basso nel monitor.
- Il telecomando è nuovo? Guarda se una linguetta di plastica isola la pila: va tolta (il mini telecomando Adafruit ce l'ha).
- «Segnale non riconosciuto» su tutti i tasti: avvicinati, il segnale può arrivare debole; se non cambia, quel telecomando usa un protocollo che la libreria non conosce.

6 Prova il telecomando della TV

Prendi i telecomandi di casa: TV, decoder, impianto audio. La libreria riconosce i protocolli più diffusi (NEC, Sony, RC5, RC6, Samsung, LG, Panasonic, JVC, Denon e altri) e per ognuno scrive il nome. Con quelli che non conosce compare «Segnale non riconosciuto». I condizionatori sono un caso a parte: mandano messaggi molto più lunghi e la libreria può non riconoscerli, oppure scrivere `PulseDistance`, cioè capisce la forma del segnale ma non sa dividerlo in indirizzo e comando. Anche un telecomando che lampeggia a una frequenza diversa dai 38 kHz si sente peggio: il filtro del ricevitore è sintonizzato lì.

Ricorda che la luce arriva a tutti: se usi i tasti del volume della TV per comandare Arduino, anche la TV cambierà il volume.

7 Un tasto accende il LED

```
// Un tasto del telecomando accende e spegne il LED L della scheda (pin 13).
#include <Arduino.h>
#include <IRremote.hpp>

const int PIN_IR = 2;              // il pin S del ricevitore
const int PIN_LED = 13;            // il LED L, già montato sulla scheda
const uint16_t TASTO_LED = 0x10;   // tasto 1 del telecomando Adafruit 389: scrivi il tuo

bool acceso = false;               // all'inizio il LED è spento

void setup() {
  pinMode(PIN_LED, OUTPUT);
  IrReceiver.begin(PIN_IR, DISABLE_LED_FEEDBACK);  // il LED L ora lo comandiamo noi
}

void loop() {
  if (IrReceiver.decode()) {
    bool tenuto = IrReceiver.decodedIRData.flags & IRDATA_FLAGS_IS_REPEAT;
    if (!tenuto && IrReceiver.decodedIRData.command == TASTO_LED) {
      acceso = !acceso;                            // spento diventa acceso e viceversa
      digitalWrite(PIN_LED, acceso ? HIGH : LOW);
    }
    IrReceiver.resume();
  }
}
```

Scegli un tasto e scrivi il suo comando al posto di `0x10` nella riga di `TASTO_LED`. Con il mini telecomando Adafruit 389 lascia 0x10: è il suo tasto 1. Il LED è il **LED L** già montato sulla scheda, sul pin 13: non serve collegare niente. Per questo `DISABLE_LED_FEEDBACK` dice alla libreria di non usarlo più per lampeggiare.

Parola nuova: il contrario
In `acceso = !acceso` il punto esclamativo vuol dire «il contrario»: se `acceso` era `true` diventa `false`, e viceversa. Così lo stesso tasto accende e spegne.

✓ Cosa devi vedere
Premi il tasto: il LED L si accende. Premilo ancora: si spegne. Gli altri tasti non fanno niente. Se tieni premuto, il LED non sfarfalla: le ripetizioni vengono scartate.

Se non funziona
- Non succede niente: il comando scritto nel programma è quello del tasto? Rileggilo nella tabella, lettera per lettera.
- Il LED si accende e si spegne da solo mentre tieni premuto: manca il controllo `!tenuto` nella riga dell'`if`.

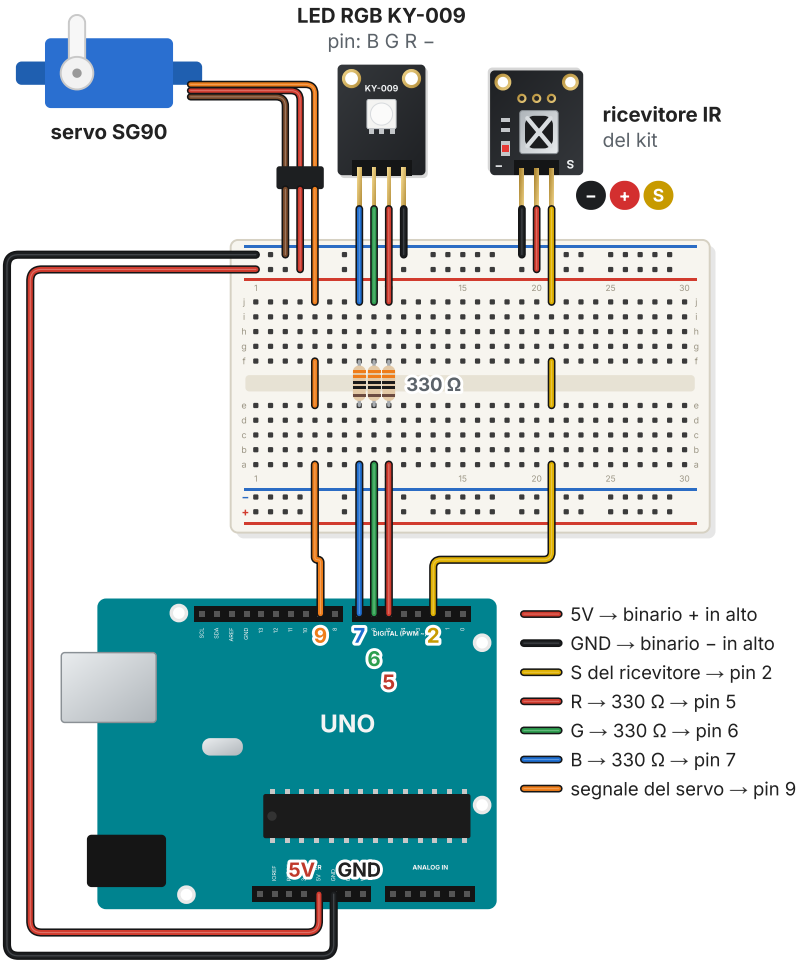
8 Aggiungi il LED RGB e il servo

Il ricevitore resta dov'è. Il modulo KY-009 ha quattro pin: **B**, **G**, **R** e **–** (catodo comune). Le scritte sono sul retro; nel nostro modulo, con il LED verso di te e i pin in basso, l'ordine da sinistra è B, G, R, –. Sulla schedina non ci sono resistenze: ogni colore ne vuole una da **330 Ω** (arancio, arancio, nero, nero, marrone). Segui il disegno qui sotto.

- B, G e R con tre cavetti maschio-femmina nelle colonne **8**, **9** e **10**, fila j; il – alla fila – in alto.
- Le tre resistenze a cavallo della scanalatura, da f a e, nelle stesse colonne.
- Dalla fila a delle colonne 8, 9 e 10 tre cavetti ai pin **7** (blu), **6** (verde) e **5** (rosso).
- Il connettore del servo ha tre fori: infila tre cavetti maschio-maschio. **Marrone** alla fila – in alto, **rosso** alla fila +, **arancione** nella colonna 5, fila j. Un ponticello da f a e nella colonna 5 e un cavetto dalla fila a al **pin 9**.

Parola nuova: perché 330 Ω
La nostra scheda del KY-009 indica al massimo 10 mA per colore, con tensioni da 1,8 a 3,5 V. Nel caso peggiore, con 1,8 V, passano (5 − 1,8) / 330 ≈ 9,7 mA: è un nostro calcolo e resta sotto i 10 mA del LED e sotto i 20 mA che Arduino indica per ogni pin.

Ricevitore, LED RGB e servo



informaticait.it

9 Cambia colore con i tasti

```
// I tasti 1, 2 e 3 accendono e spengono rosso, verde e blu; il tasto 0 spegne tutto.
#include <Arduino.h>
#include <IRremote.hpp>

const int PIN_IR = 2;              // il pin S del ricevitore
const int PIN_ROSSO = 5;           // R del modulo LED RGB (passa dalla sua resistenza)
const int PIN_VERDE = 6;           // G
const int PIN_BLU = 7;             // B

// i comandi dei tasti 1, 2, 3 e 0 del mini telecomando Adafruit 389:
// con un altro telecomando scrivi quelli letti con il primo programma
const uint16_t TASTO_1 = 0x10;
const uint16_t TASTO_2 = 0x11;
const uint16_t TASTO_3 = 0x12;
const uint16_t TASTO_0 = 0x0C;

bool rosso = false;
bool verde = false;
bool blu = false;
```

```
void mostra() {                    // accende i colori che valgono true
  digitalWrite(PIN_ROSSO, rosso ? HIGH : LOW);
  digitalWrite(PIN_VERDE, verde ? HIGH : LOW);
  digitalWrite(PIN_BLU, blu ? HIGH : LOW);
}

void setup() {
  pinMode(PIN_ROSSO, OUTPUT);
  pinMode(PIN_VERDE, OUTPUT);
  pinMode(PIN_BLU, OUTPUT);
  IrReceiver.begin(PIN_IR, ENABLE_LED_FEEDBACK);
}

void loop() {
  if (IrReceiver.decode()) {
    bool tenuto = IrReceiver.decodedIRData.flags & IRDATA_FLAGS_IS_REPEAT;
    uint16_t tasto = IrReceiver.decodedIRData.command;
    if (!tenuto) {                 // un colore cambia solo alla prima pressione
      if (tasto == TASTO_1) rosso = !rosso;
      if (tasto == TASTO_2) verde = !verde;
      if (tasto == TASTO_3) blu = !blu;
      if (tasto == TASTO_0) {
        rosso = false;
        verde = false;
        blu = false;
      }
      mostra();
    }
    IrReceiver.resume();
  }
}
```

Scrivi i comandi dei tuoi tasti 1, 2, 3 e 0 nelle quattro righe `TASTO_`. Ogni tasto accende o spegne un colore; due colori insieme si mescolano come nella guida n. 46: 1 e 2 fanno giallo, tutti e tre bianco. Lo 0 spegne tutto.

Parola nuova: timer
Un contatore interno al microcontrollore: tra le altre cose fa il PWM di due pin. La libreria IRremote usa quello dei pin **3 e 11**, la libreria Servo quello dei pin **9 e 10**. Per questo qui i colori sono accesi o spenti con `digitalWrite`: sette colori più il nero. Se vuoi le sfumature con `analogWrite`, restano i pin 5 e 6. Per lo stesso motivo niente `tone()` per un cicalino: usa il timer di IRremote e, come scrive la sua documentazione, dopo un `tone()` la ricezione si ferma.

✓ Cosa devi vedere
Premi 1: rosso. Premi 2: rosso e verde insieme, cioè giallo. Premi 1 ancora: resta il verde. Lo 0 spegne tutto.

Se non funziona

- Un colore non si accende mai: controlla il suo cavetto e la sua resistenza, che deve stare proprio da f a e nella stessa colonna.
- I colori escono scambiati: il tuo modulo ha i pin in un altro ordine. Segui le scritte sul retro, non il disegno.

10 Muovi il servo con le frecce

```
// Le frecce girano il servo di 10 gradi alla volta, OK lo rimette al centro.
#include <Arduino.h>
#include <IRremote.hpp>
#include <Servo.h>

const int PIN_IR = 2;              // il pin S del ricevitore
const int PIN_SERVO = 9;           // il filo arancione del servo

// i comandi delle frecce e di Enter del mini telecomando Adafruit 389:
// con un altro telecomando scrivi quelli letti con il primo programma
const uint16_t TASTO_SINISTRA = 0x08;
const uint16_t TASTO_DESTRA = 0x0A;
const uint16_t TASTO_OK = 0x09;

Servo braccio;
int angolo = 90;                   // si parte dal centro

void setup() {
  braccio.attach(PIN_SERVO);
  braccio.write(angolo);
  IrReceiver.begin(PIN_IR, ENABLE_LED_FEEDBACK);
}

void loop() {
  if (IrReceiver.decode()) {
    uint16_t tasto = IrReceiver.decodedIRData.command;   // anche tenendo premuto
    if (tasto == TASTO_SINISTRA) angolo = angolo + 10;
    if (tasto == TASTO_DESTRA) angolo = angolo - 10;
    if (tasto == TASTO_OK) angolo = 90;
    angolo = constrain(angolo, 0, 180);                  // mai sotto 0 e mai sopra 180
    braccio.write(angolo);
    IrReceiver.resume();
  }
}
```

Scrivi i comandi delle tue frecce e del tasto OK nelle tre righe `TASTO_`.

Qui le ripetizioni servono: finché tieni premuta una freccia il telecomando continua a mandare il segnale e il braccetto avanza di 10 gradi alla volta. `constrain` tiene l'angolo tra 0 e 180; OK lo riporta a 90.

✓ Cosa devi vedere
Una pressione breve sposta il braccetto di 10 gradi; tenendo premuto continua fino in fondo. OK lo rimette al centro.

Se non funziona

- Il braccetto gira al contrario: scambia `+ 10` e `- 10` nelle righe delle due frecce.
- Il servo si muove a scatti o la scheda si riavvia: Adafruit scrive che può essere un problema di alimentazione e che alcune porte USB non danno abbastanza corrente per la scheda e il servo insieme. Prova un'altra porta o un alimentatore.
- Il ricevitore sbaglia solo mentre il servo si muove: il motore può disturbare i 5 V. Per un'alimentazione disturbata Vishay, che produce il ricevitore, consiglia una resistenza in serie sul + del ricevitore e un condensatore tra il suo + e il suo −.

11 Altri ricevitori



Il ricevitore si trova anche senza schedina, per esempio i nostri 1838 in confezione da tre o il **Vishay TSOP38238**. Si monta direttamente sulla breadboard: con la lente verso di te i piedini sono, da sinistra, **OUT**, **GND** e **VCC**. OUT va al pin 2, GND alla fila −, VCC alla fila +. Nel datasheet del TSOP382 la resistenza di pull-up dell'uscita è già dentro il ricevitore: non serve aggiungerne una.

Il **ricevitore Grove** di Seeed Studio ha lo stesso Vishay e un cavo a quattro fili: giallo il segnale, rosso VCC, nero GND, bianco non collegato.

12 Mettiti alla prova

Prova tu

- Unisci i due programmi: frecce per il servo, 1, 2 e 3 per i colori. Le righe di `setup()` vanno insieme, i controlli dei tasti dentro lo stesso `if`.
- Fai una sbarra del parcheggio: OK alza il braccetto a 90 gradi, aspetta 3 secondi e torna a 0.
- Controlla anche l'indirizzo, con `&& IrReceiver.decodedIRData.address == ...`: così un altro telecomando con gli stessi comandi non disturba.

✓ Cosa hai imparato

- il ricevitore filtra la luce a 38 kHz e sul pin S dà impulsi puliti;
- IRremote trasforma ogni tasto in protocollo, indirizzo e comando: il comando è il numero da usare;
- con il primo programma leggi i comandi di qualsiasi telecomando, poi li scrivi negli altri;
- le ripetizioni di un tasto tenuto premuto si possono scartare o usare.

Trovi i componenti su informaticait.it

- Telecomando con ricevitore ir infrarossi 38 Khz VS183 per arduino e...
- Ricevitori infrarossi IR TL1838 VS1838 VS1838B 38 Khz 3 pezzi
- Mini telecomando a infrarossi 21 tasti, codifica NEC 38 kHz
- Ricevitore a infrarossi 38 kHz TSOP38238 per telecomandi, 3–5 V
- Modulo Grove ricevitore a infrarossi TSOP382 38 kHz, portata 10 m, 3,3–5 V
- Scheda Atmega328p compatibile con IDE Arduino uno r3 100% alta qualità
- Scheda Arduino Uno Rev3 originale
- Modulo led rgb smd 5050 a catodo comune multicolore pwm ky-009 per arduino
- 10pz Resistenze resistore resistenza 1/4w 330r 330 ohm 1% 0,25w film di...
- Micro servo TowerPro SG90 9g 180° servomotore RC con squadrette per Arduino
- Piastra sperimentale breadboard pcb 400 contatti per arduino elettronica
- Cavetti jumper Dupont maschio-femmina 20 cm per breadboard
- 40 ponticelli cavi cavetti flessibili dupont maschio maschio 20 cm per...

Clicca sul nome per aprire la scheda prodotto. Disponibilità e prezzi aggiornati sul sito.

Fonti: Vishay, TSOP382.. TSOP384.., datasheet doc. 82491 rev. 2.1 · Libreria IRremote 4.7.1 (Arduino-IRremote): README ed esempi · Adafruit, prodotti 157 e 389, codici del mini telecomando e guida «Using an Infrared Library on Arduino» · Seeed Studio, wiki Grove - Infrared Receiver e standard Grove · Arduino, UNO R3 Tech Specs, libreria Servo 1.3.0, riferimento tone() e analogWrite() · Schede prodotto informaticait 2867, 1909, 5031 e 2024 · Guide informaticait n. 34, 41 e 46