

CODIFICA E RICETRASMISSIONE DI UN SEGNALE A DISTANZA MEDIANTE ONDE LUMINOSE

Obbiettivo

L'obbiettivo dell'esperimento è quello di codificare un segnale alfanumerico in una serie di stringhe di codice binario, quindi trasmetterlo, riceverlo e decodificarlo mediante impulsi luminosi controllati dalla scheda elettronica Arduino.

Esecuzione dell'esperimento

Il primo passo è la costruzione dei circuiti delle due schede, che chiameremo da ora *Trasmittente* e *Ricevitore*. Per fare ciò saranno necessari i seguenti materiali:

- Led a luce infrarossa
- Ricevitore infrarosso
- Laser
- Fotoresistenza
- N. 2 schede Arduino
- Cablaggi
- N. 2 resistenze da 270 Ω e 10 k Ω

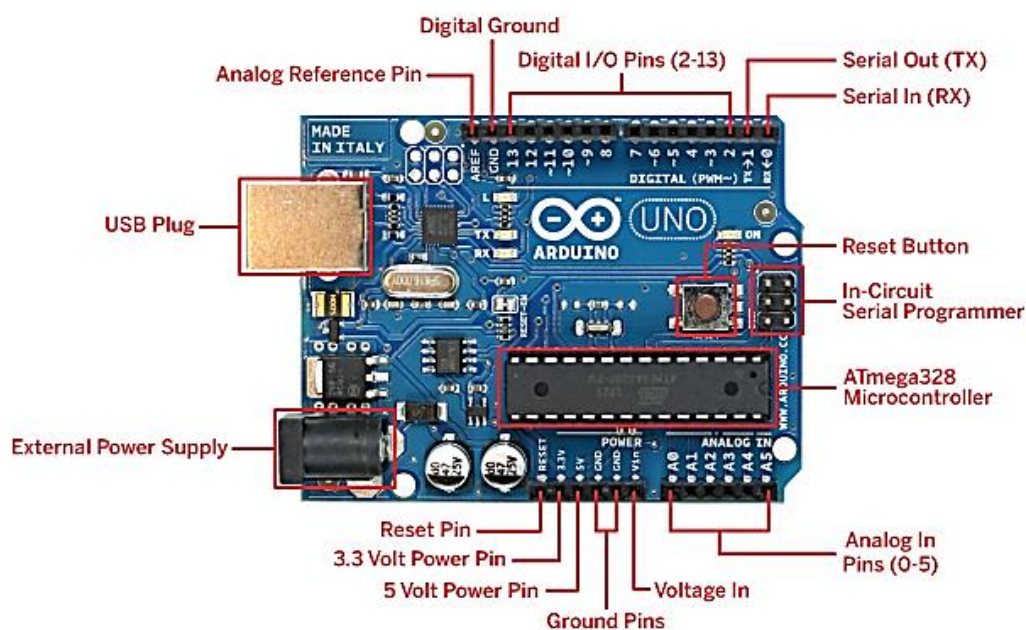


Figura 1. Scheda elettronica Arduino

Costruzione del circuito ricevitore

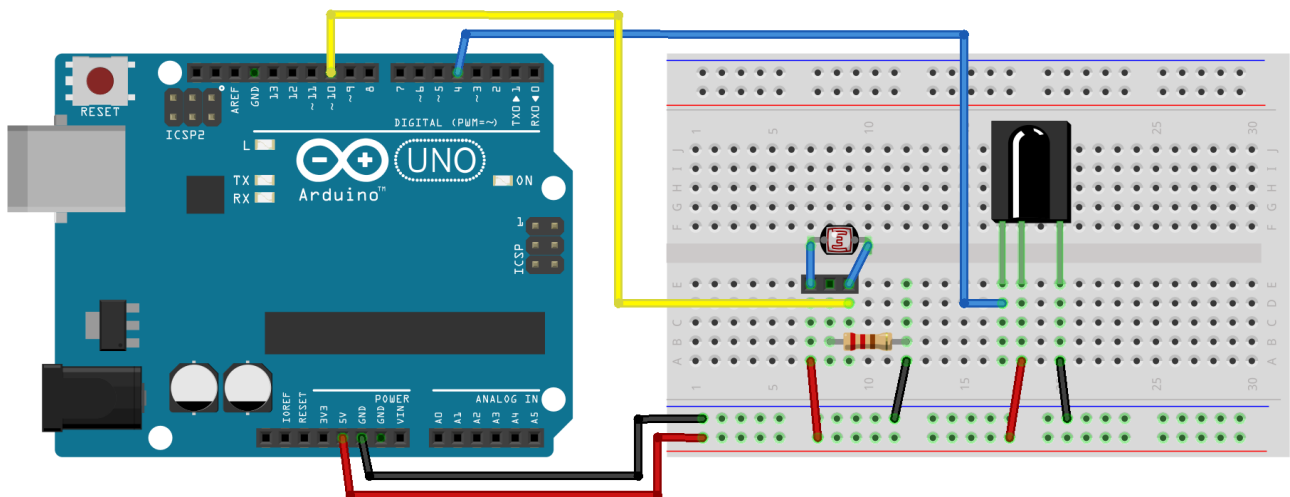


Figura 2. Schema del circuito ricevitore

Il ricevitore è costituito da due principali componenti, una fotoresistenza ed un rilevatore infrarosso. La **fotoresistenza** possiede i seguenti pin: 1 – VCC ; 2 – GND ; 3 – SIGNAL che andranno collegati rispettivamente a 5V, al GND, e ad una porta digitale su arduino (nell'esempio 10), inserendo una resistenza da 270 Ω tra il GND e il componente.

Il rilevatore infrarosso invece possiede le seguenti porte: 1 – GND ; 2 – VCC ; 3 – SIGNAL. Le porte GND e VCC andranno collegati relativamente ai pin GND e 5V sulla scheda Arduino, mentre la porta SIGNAL andrà collegata ad uno dei pin digitali di Arduino (4 nell'esempio).

N.B. Fare attenzione ad effettuare ogni collegamento con la scheda non alimentata, e controllare la correttezza dei collegamenti prima di alimentarla per evitare di bruciare i componenti.

1st digit	2nd digit	Multiplier	Tolerance
0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
8	8	x 0.1	±5%
9	9	x 0.01	±10%

Figura 3. Tabella di lettura del valore di una resistenza

Costruzione del circuito trasmettitore

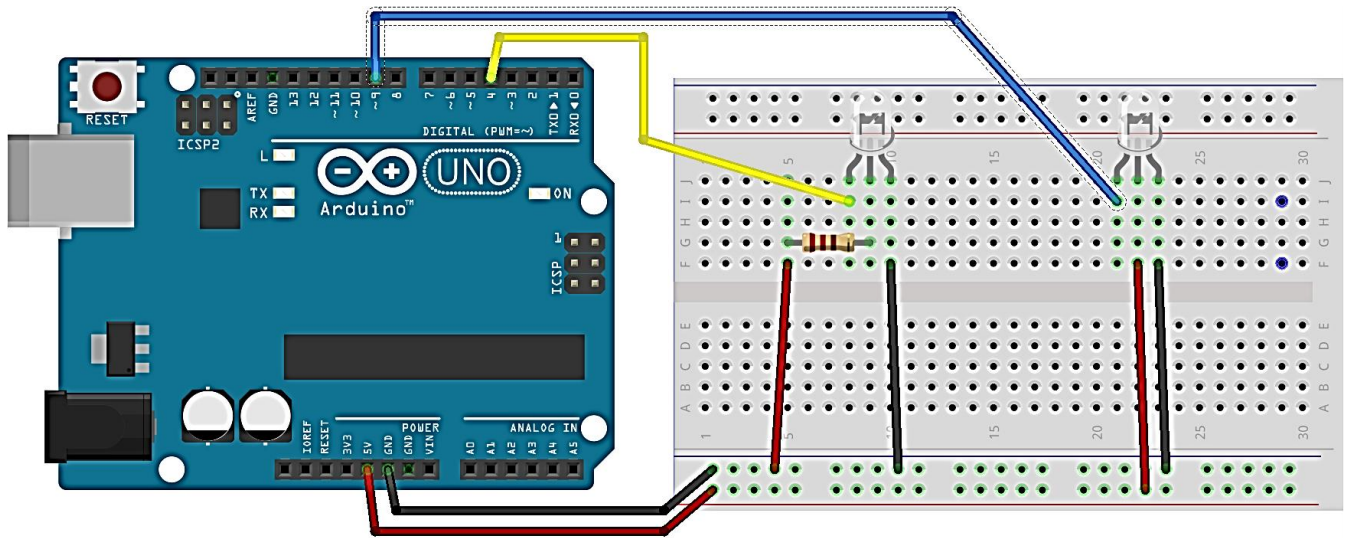


Figura 4. Schema del circuito trasmettitore

Il trasmettitore, analogamente al ricevitore, sarà composto da due principali componenti, un laser rosso e un led infrarosso.

I pin del laser sono nell'ordine 1 – Signal ; 2 – VCC ; 3 – GROUND. Collegare quindi il primo pin ad una porta digitale della scheda Arduino (9 nell'esempio), mentre il VCC e il GROUND ai poli negativo e positivo come in precedenza.

Per il collegamento del diodo infrarosso è necessaria una resistenza da 10 kΩ, che va inserita tra il pin centrale (VCC) e il polo positivo. Il pin 1 come in precedenza è il Signal, che va collegato ad una porta digitale della scheda Arduino (4 nell'esempio).

Programma (sketch) per Arduino – Circuito trasmettitore

Per prima cosa si andrà a definire il “protocollo” attraverso cui i caratteri andranno codificati, creando due Array contenenti rispettivamente il carattere e la stringa corrispondente.

```
char chars[] =
{'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H',
'I','L','M','N','O','P','Q','R','S','T','U','V','Z'};
String codes[] =
{"00000","00001","00010","00011","00100","00101","00110","00111","01000",
"01001","01010","01011","01100","01101","01110","01111","10000","10001",
"10010","10011","10100","10101","10110","10111","11000","11001","11010","1
1011","11100","11101","11110"};
```

Si stabilisce poi a quali pin della scheda le componenti sono collegate.

```
const int laser = 9;    (Pin a cui è collegato il laser)
const int IRwrite = 4; (Pin a cui è collegato il led infrarosso)
```

Viene dichiarata una costante utile per l'esecuzione del programma

```
const int t = 1000;
```

Nel dettaglio, la costante “t” sarà utilizzata per definire gli intervalli di attesa, in questo caso, 1000 millisecondi (1s).

A questo punto inizia la prima parte vera e propria del codice, quella di Setup, cioè la parte che viene eseguita una singola volta all'avvio del programma, per poi lasciare spazio ad infiniti cicli del loop.

```
void setup() {

    Serial.begin(9600); (si stabilisce la frequenza di aggiornamento della porta seriale)

    pinMode(LED_BUILTIN,OUTPUT);
    pinMode(laser,OUTPUT);
    pinMode(IRwrite,OUTPUT);    (si dichiara che i component collegati alle porte in questione
verranno utilizzati come OUTPUT)
```

Il sistema viene poi inizializzato spegnendo tutte le sorgenti luminose e inserendo il codice da comunicare al ricevitore.

```
//Spegnimento di tutte le sorgenti
digitalWrite(laser,LOW);
digitalWrite(IRwrite,LOW);
```

```
//Codifica del codice
String code = "H725";
```

Viene inviata una serie di brevi segnali infrarossi in modo da accertarsi che il ricevitore ne rilevi almeno uno, per iniziare la lettura.

```
//Impulso d'avvio

Serial.println("Impulso IR inviato");
for(int x = 0;x<5;x++){
```

```

digitalWrite(IRwrite,HIGH);
delay(40);
digitalWrite(IRwrite,LOW);
delay(40);
}

```

Inizia l'invio del segnale in forma binaria, l'algoritmo prende in esame ciascuno dei 4 caratteri, stampa nella seriale lo stato di avanzamento della comunicazione mentre invia il segnale tramite la funzione "codice".

```

//Invio segnale
for(int lettera = 0; lettera < 4; lettera++){
  Serial.print("Invio carattere ");
  Serial.println(code.charAt(lettera));

  for(int nBit = 0; nBit < 5; nBit++){
    Serial.print(nBit+1);
    digitalWrite(laser,codice(code.charAt(lettera),nBit));

    Serial.println("");
    delay(t);
  }
}

```

Al termine della comunicazione il laser viene nuovamente spento (potrebbe già esserlo, a seconda dell'ultimo bit inviato).

```

digitalWrite(laser,LOW);

}

```

A questo punto la parte di setup del codice è terminata, e si sfrutta il loop per far lampeggiare il led interno della scheda (LED_BUILTIN), per segnalare il terminato invio.

```

void loop() {

  digitalWrite(LED_BUILTIN,HIGH);
  delay(400);
  digitalWrite(LED_BUILTIN,LOW);
  delay(400);
}

```

Chiuso anche il loop si definisce il funzionamento della funzione "codice", cioè la funzione che si occupa di prendere un singolo carattere alla volta nel codice da inviare, cercarlo all'interno dell'array di caratteri e cercare il codice binario corrispondente nel secondo array.

```

int codice(char val,int index){
  for( int y = 0; y < sizeof(chars); y++ ){
    if( val == chars[y] ){
      Serial.print("° bit Inviato: ");
      Serial.print(codes[y].charAt(index));
      if((codes[y].charAt(index)) == '0'){

```

```

        return 0;
    }
    else{return 255;}
}
}
}

```

Programma (sketch) per Arduino - Circuito ricevitore

Il programma di ricezione è molto più snello in quanto non è presente tutta la parte di decodifica del codice, che sarà svolta manualmente.

Come nello sketch precedente, vanno dichiarati i pin a cui sono connessi i componenti del circuito.

```

//Dichiarazione del pin a cui è connessa la fotoresistenza
const int photor = 11;
const int IRread = 2;

```

Inizia la funzione “setup”, di nuovo con l’inizializzazione della porta seriale e la definizione delle periferiche, questa volta come dispositivi di input.

```

void setup() {
//Avvio della seriale
Serial.begin(9600);

//Fotoresistenza impostata come INPUT
pinMode(phototor,INPUT);
pinMode(IRread,INPUT);
}

```

Il codice del ricevitore è per la maggior parte incluso nella funzione loop in quanto il ricevitore non sa il preciso istante in cui arriverà la comunicazione, deve quindi continuamente aggiornarsi in attesa di un impulso d’avvio.

```

void loop() {

```

Una condizione verifica la ricezione o non di un segnale infrarosso per avviare la lettura.

```

    if(digitalRead(IRread)==LOW){
        Serial.println("IR Ricevuto");
        delay(500);
    }

```

Una volta rilevato l’impulso il programma inizia a leggere lo stato del laser e a stamparlo a schermo ordinato in gruppi di 5 bit.

```

        for(int y = 0; y < 4; y++){
            for(int i = 0;i<5;i++){
                Serial.print((abs(digitalRead(phototor)-1)));
                delay(1000);
            }
        }
        Serial.println();
    }

```

Esecuzione dell'esperimento (parte 1)

Per caricare gli sketch sulla scheda Arduino usare il software Arduino che si trova sul desktop; una volta aperto bisogna copiare lo sketch precedente al suo interno e poi salvare il file su una cartella. Poi si può verificare la corretta scrittura del codice con il pulsante compila. Se la compilazione va a buon fine si può caricare il programma sulla scheda con il pulsante carica (ovviamente la scheda deve essere connessa al computer tramite la porta USB).

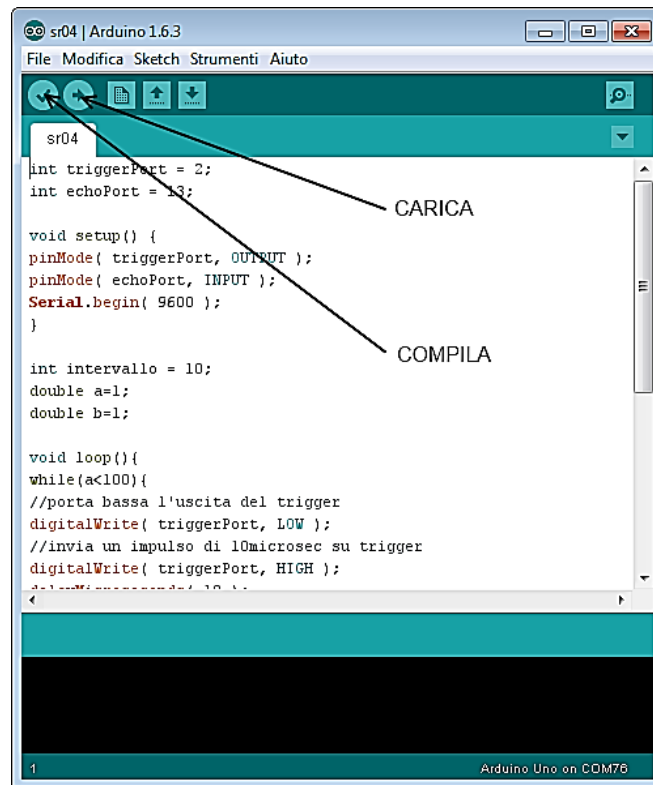


Fig.4 Finestra del software Arduino

Per fare in modo che trasmettitore e ricevitore siano in grado di comunicare, è necessario fissare e allineare il fascio di luce del laser con la fotoresistenza, facendo anche attenzione che non ci siano ostacoli tra la sorgente e il sensore infrarosso. Quest'ultimo ha una portata limitata, è consigliabile quindi non superare distanze di 50-60 cm tra ricevitore e trasmettitore.

Una volta allineati i due dispositivi, aprire il monitor seriale del ricevitore (per primo, in modo da non perdere dati ricevuti) e successivamente aprire il monitor seriale del trasmettitore; facendo ciò l'esecuzione del programma si riavvierà inviando l'impulso d'inizio e facendo iniziare la trasmissione di dati.

A questo punto sul monitor del ricevitore apparirà una serie di segnali traducibile nel codice iniziale mediante la seguente tabella:

00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	
0	1	2	3	4	5	6	7	8	9	
01010	01011	01100	01101	01110	01111	10000	10001	10010	10011	10100
A	B	C	D	E	F	G	H	I	L	M
10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	
N	O	P	Q	R	S	T	U	V	Z	

Controllare la corrispondenza tra la stringa di prova (“*H725*” nell’esempio) trasmessa e quella che appare sul monitor collegato al circuito ricevitore.

Esecuzione dell’esperimento (parte 2)

Ripetere l’esperimento allineando il circuito ricevitore con quello trasmettitore del docente e rilevare il codice trasmesso dal docente.

Conclusioni

Annotare sul report il codice segreto trasmesso dal docente.