

```
1 // Ribiani & Graziadei Stampante 3D con arduino Uno
2 // 1 potenziometro
3 // 6 pulsanti
4 // 1 motori passo passo
5
6 // INPUT vengono dati dei nomi ai valori dei pulsanti letti dall'arduino
7 int Pb_fw = digitalRead(11); // pulsante che fa girare in avanti il motore
8 int Pb_back = digitalRead(13); // pulsante che fa girare indietro il motore
9 int Pb_enable = digitalRead(7); // pulsante che abilita il driver
10 int Pb_stop = digitalRead(12); // pulsante che arresta il funzionamento del
    programma e disabilita l'integrato
11
12 // OUTPUT vengono assegnati i numeri dei pin a delle variabili che ne rappresenta
    la funzione
13 int dir = 8; // pin connesso alla direzione dell'integrato
14 int enable = 6; // pin connesso all'ingresso enable dell'integrato
15 int ledR = 2; // pin che alimenta il led rosso
16 int ledY = 3; // pin che alimenta il led giallo
17 int ledV = 4; // pin che alimenta il led verde
18 int Step = 10; // pin connesso all'ingresso step dell'integrato, pwm,
    ogni volta che è alto il motore compie uno step
19 int State_enable = HIGH; // impostiamo l'enable alto (quindi non attivo) in modo
    che il motore non parta in maniera accidentale
20 int State_stop = LOW; // impostiamo lo stop basso (quindi attivo) in modo che il
    motore non parta in maniera accidentale
21 // le variabili chiamate State_xxxx corrispondono all'ultimo valore utilizzato e
    non a quello del pulsante
22 int potentiometer; // dichiariamo la variabile alla quale verrà assegnato il valore
    letto dal potenziometro
23 float Speed; // variabile che va da 18000 a 25000 micro secondi
24
25 void setup()//nel setup impostiamo le variabili con i valori desiderati
    all'accensione del circuito
26 //e dichiariamo se sono uscite e ingressi (specificandone il tipo)
27 {
28
29
30 // pullup tensione bassa = true
31 digitalWrite(enable, HIGH); // inizializziamo il valore del enable
    sull'integrato alto (quindi spento)
32 digitalWrite(State_stop, HIGH);
33 digitalWrite(State_enable, HIGH); //inizializziamo gli stati in modo che appena
    acceso il circuito sia in idle
34 pinMode(Pb_fw, INPUT_PULLUP); // impostiamo come input il pin al quale sono
    collegati i due pulsanti
35 pinMode(Pb_back, INPUT_PULLUP); // per la direzione
36 pinMode(dir, OUTPUT); // impostiamo il pin collegato al pin direzione
    del driver come output
37
38 pinMode(Pb_enable, INPUT_PULLUP); // impostiamo il pin collegato al pulsante che
    fungerà da enable come input
39
40 pinMode(Pb_stop, INPUT_PULLUP); // impostiamo il pin collegato al pulsante di
    stop come input
41
42 // LED
43 pinMode(ledR, OUTPUT); // impostiamo il pin collegato al led rosso come output
```

```

44     pinMode(ledY, OUTPUT); // impostiamo il pin collegato al led giallo come output
45     pinMode(ledV, OUTPUT); // impostiamo il pin collegato al led verde come output
46
47 }
48
49 void loop()
50 {
51     ////////////////////////////////////////////////// GESTIONE VELOCITÀ
52     ////////////////////////////////////////
53     potentiometer = analogRead(A0); //leggiamo la tensione ai capi del potenziometro
54     e la assegnamo alla variabile
55     Speed=map(potentiometer,0,1023,18000,25000); //cambiamo il valore minimo della
56     tensione al delay minimo e 1023 al valore massimo di delay
57     ////////////////////////////////// GESTIONE START & STOP
58     ////////////////////////////////////////
59     if (State_enable == HIGH && Pb_enable == LOW) // se viene premuto il pulsante di
60     enable per la prima volta dopo lo stop
61     {
62         digitalWrite(State_enable, LOW); // si imposta il valore dello stato a vero
63         digitalWrite(State_stop, HIGH); // si elimina il vecchio valore di stop
64         resettandone il valore
65     }
66
67     if (State_stop == HIGH && Pb_stop == LOW) // se viene premuto lo stop
68     {
69         digitalWrite(State_enable, HIGH); // si mette lo stato come vero
70         digitalWrite(State_stop, LOW); // si elimina il vecchio stato dell'enable
71     }
72     ////////////////////////////////// FUNZIONAMENTO MOTORE
73     ////////////////////////////////////////
74     if (State_enable == LOW && State_stop == HIGH) // se è stato premuto il pulsante
75     di enable e non lo stop (fattore di sicurezza)
76     {
77         digitalWrite(ledV, HIGH); // si spengono tutti i led tranne quello verde poichè
78         il circuito sta funzionando bene
79         digitalWrite(ledY, LOW);
80         digitalWrite(ledR, LOW);
81         if (Pb_fw == LOW && Pb_back == LOW){ //se vengono premuti entrambi i pulsanti
82         di direzione
83         digitalWrite(Step, LOW); // si tiene fermo il motore
84         delay(10); //per evitare malfunzionamenti
85         digitalWrite(ledV, LOW); // si spengono tutti i led tranne quello rosso
86         poichè
87         digitalWrite(ledY, LOW); // si è verificato un errore
88         digitalWrite(ledR, HIGH);
89     }
90     else if (Pb_fw == LOW) // se viene premuto il pulsante avanti
91     {
92         digitalWrite(dir, LOW); // si imposta la direzione del motore su avanti
93         (segnale basso)
94         digitalWrite(Step, HIGH); // si alza il segnale sullo step
95         delayMicroseconds(Speed); // si fa rimanere il segnale alto per un
96         determinato tempo
97         digitalWrite(Step, LOW); // si abbassa di nuovo
98         delayMicroseconds(Speed); // si attende un altro intervallo di tempo prima di
99         un'altra interazione
100         // meno tempo si aspetterà più veloce sarà il motore
101     }

```

```
88
89     else if (Pb_back == LOW)// se viene premuto il pulsante indietro
90     {
91         digitalWrite(dir, HIGH);// si imposta la direzione del motore su indietro
(segnale alto)
92         digitalWrite(Step, HIGH);//si alza il segnale sullo step
93         delayMicroseconds(Speed);// si fa rimanere il segnale alto per un determinato
tempo
94         digitalWrite(Step, LOW); // si abbassa di nuovo
95         delayMicroseconds(Speed);//la procedura è uguale in entrambe le direzioni,
cambia solo la direzione
96     }
97 }
98
99     ////////////////////////////////////////////////// GESTIONE LED IDLE E STOP
/////////////////////////////////////////////////
100     else if (State_enable == HIGH && State_stop == LOW)//se è stato premuto lo stop
101     {
102         digitalWrite(ledV, LOW);//si spengono tutti i led
103         digitalWrite(ledY, LOW);
104         digitalWrite(ledR, HIGH);//tranne il rosso
105         digitalWrite(Step, LOW);//si ferma il motore impostando il segnale di step
costantemente basso
106     }
107     else if ((State_enable == LOW && State_stop == LOW) || (State_enable == HIGH &&
State_stop == HIGH)) //se start e stop dovessero essere attivi o inattivi
contemporaneamente
108     {
109         digitalWrite(ledV, LOW);
110         digitalWrite(ledY, HIGH);
111         digitalWrite(ledR, LOW);
112         //viene acceso solo il led giallo per segnalare un'anomalia, il circuito
infatti non prevede questa situazione
113         digitalWrite(Step, LOW);//si ferma il motore impostando il segnale di step
costantemente basso per ragioni di sicurezza
114
115     }
116 } // fine del LOOP
```