Auto

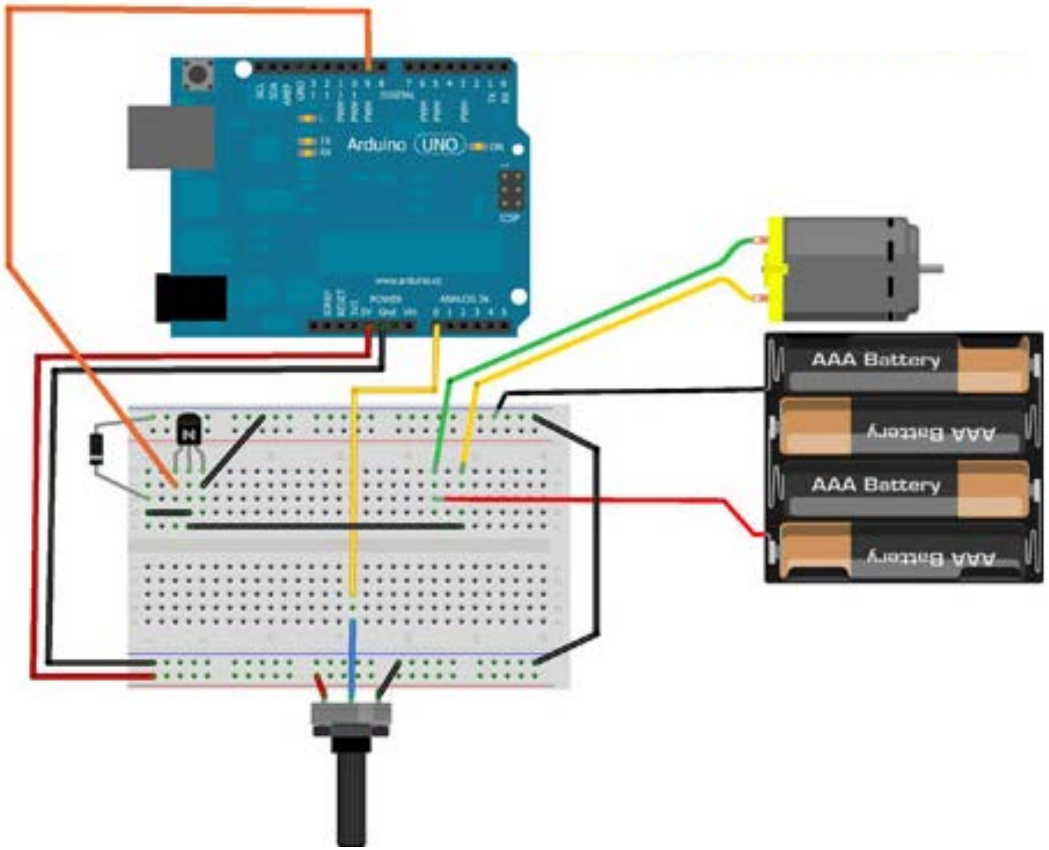
Arduino

STM32

Motori

- I motori utilizzati sono a corrente continua apolari nel senso che si possono invertire i poli
- Per pilotare i motori ci vorrebbe una tensione di mini 6 V
- I pin di arduino o della STM32 nucleo erogano una tensione di 5V e corrente massima 40 mA
- Affinchè un motore funzioni su una delle due schede c'è bisogno di una alimentazione esterna
- Sarebbe possibile tutto ciò attraverso un transistor che funzioni da interruttore tramite la configurazione illustrata nella slide successiva

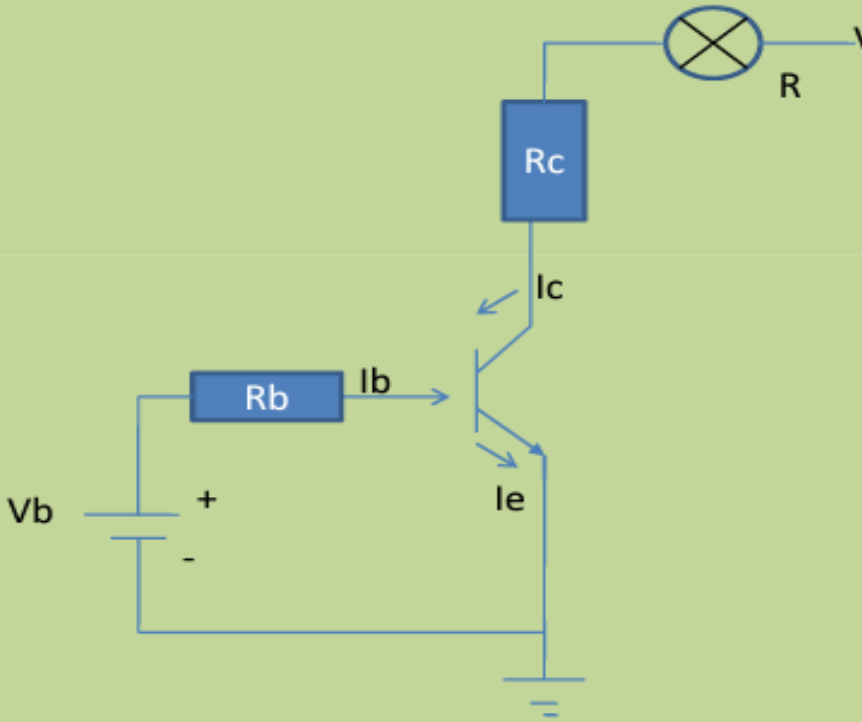
Motorino cc con la scheda Arduino uno



Alimentazione esterna

Quando il transistor viene utilizzato come interruttore, la configurazione circuitale è quella ad emettitore comune.

Una configurazione ad emettitore comun sta ad indicare che l'emettitore è comune sia al circuito di ingresso che a quello in uscita come in figura

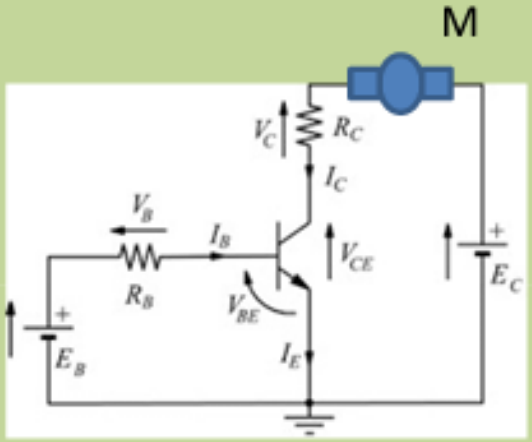


Applichiamo il principio di Kirchoff alla maglia in ingresso
$V_b = R_b * I_b + V_{be}$
Per la maglia in uscita, il principio di kirchoff:
$V_{cc} = (R_c + R) * I_c - V_{ce(sat)}$

Per i transistor Ib e Ic sono legate dalla seguente formula:
$I_c = \beta * I_b$ dove β, $V_{ce(sat)}$, V_{be} sono caratteristiche del transistor

Se $I_b = 0 \Rightarrow I_c = 0$ e il circuito risulta aperto;
Se $I_b \neq 0 \Rightarrow I_c \neq 0$ e il circuito è chiuso

Esempio di dimensionamento per il motore



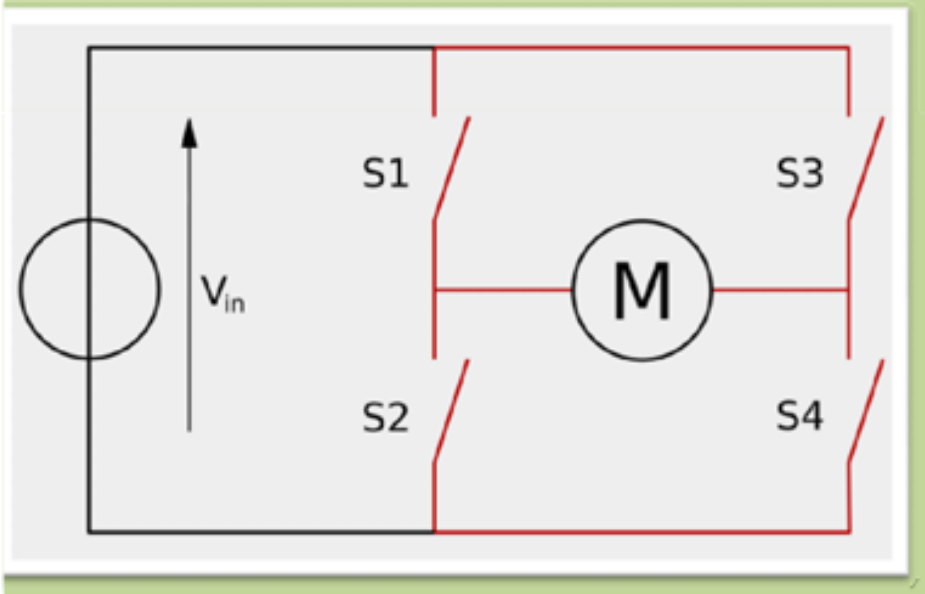
V_{CEsat}=0.3V

V_{BEsat}=0.6V
h_{fe}=100
$I_{c(max)}$=600 mA
I_c=250mA

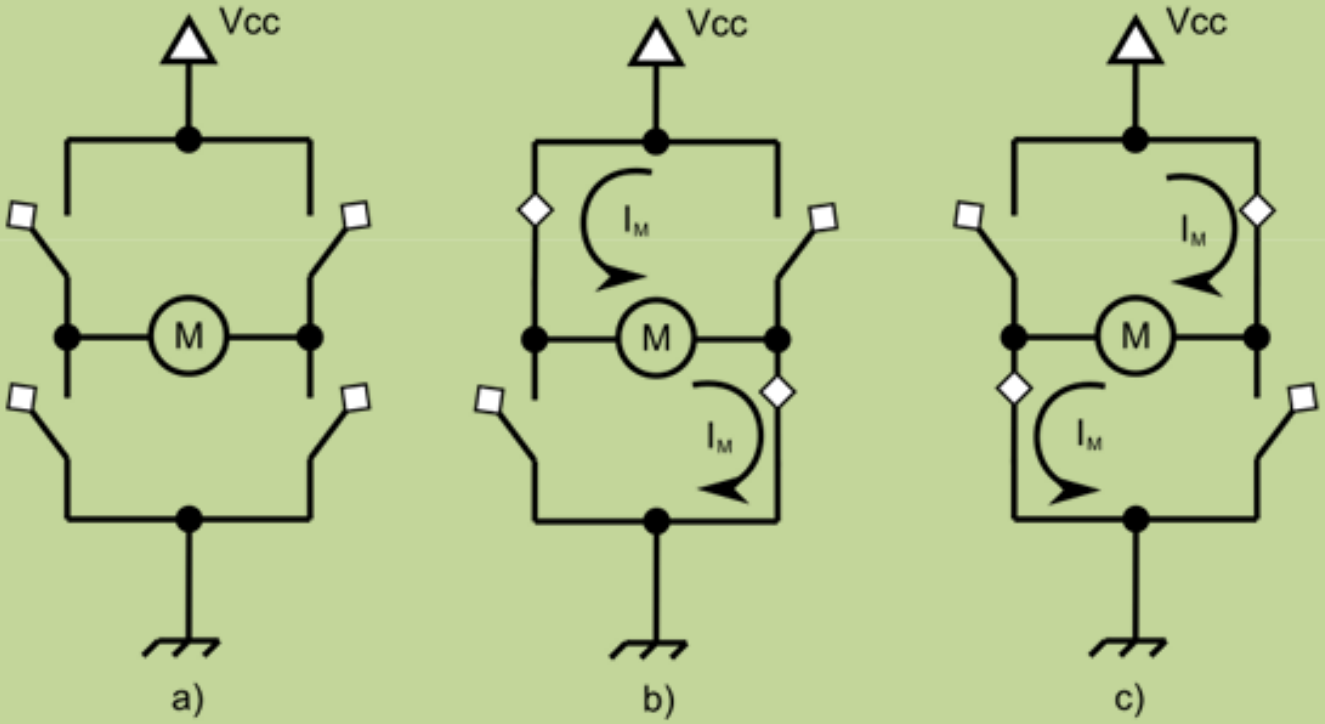
$E_B = V_{PIN}$=5V
$I_B = I_c / h_{fe}$= 250/100=2.5mA
$R_B = (V_{PIN} - V_{Besat}) / I_B$ =(5-0.6)/2.5=1760 Ω

Rc dipende dalla resistenza del motore M

Il circuito precedente permette la
rotazione del motore solo in un verso
Il seguente circuito con interruttori
fa ruotare il motore in entrambe i versi

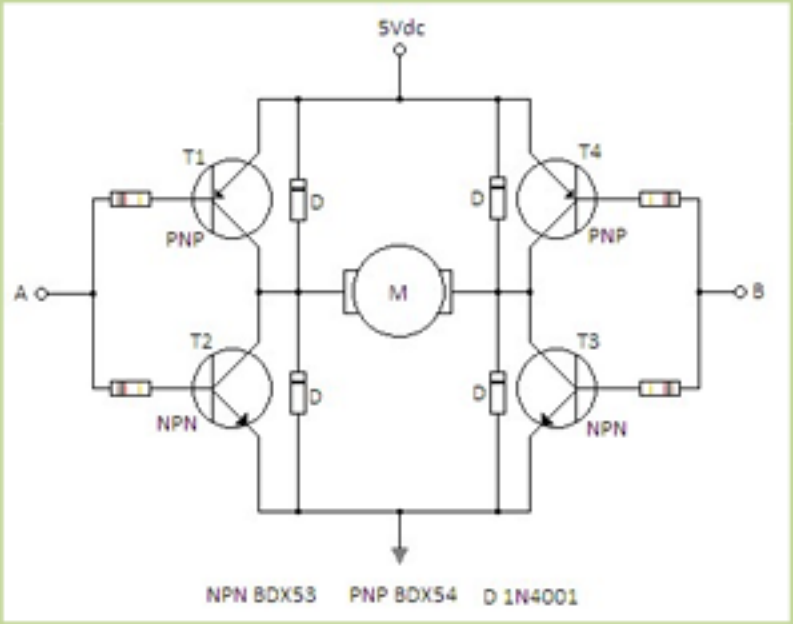


Configurazione degli interruttori per la rotazione dei motorini in uno dei due versi

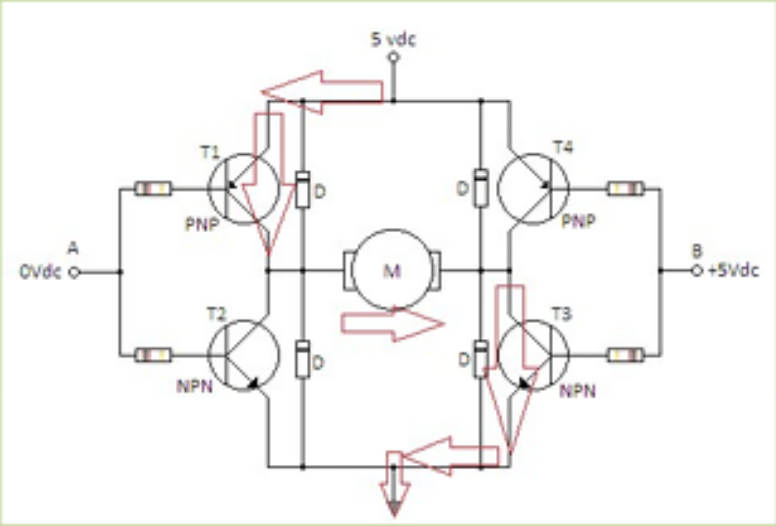


Motore bidirezionale

Per poter pilotare il motorino in due versi differenti si utilizza il ponte H formato da quattro transistor due NPN e due PNP

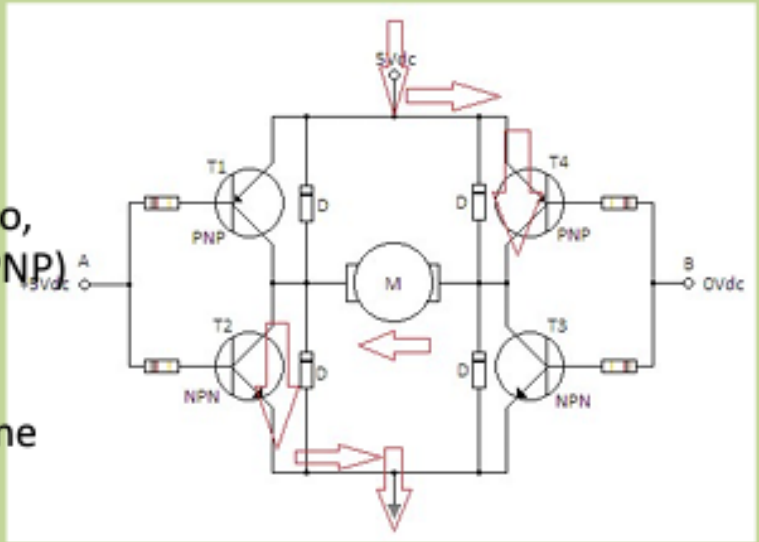


I diodi sono posizionati in antiparallelo e sono di protezione. I transitor NPN sono 3904, i PNP sono 3906 e le resistenze sono 1 KΩ

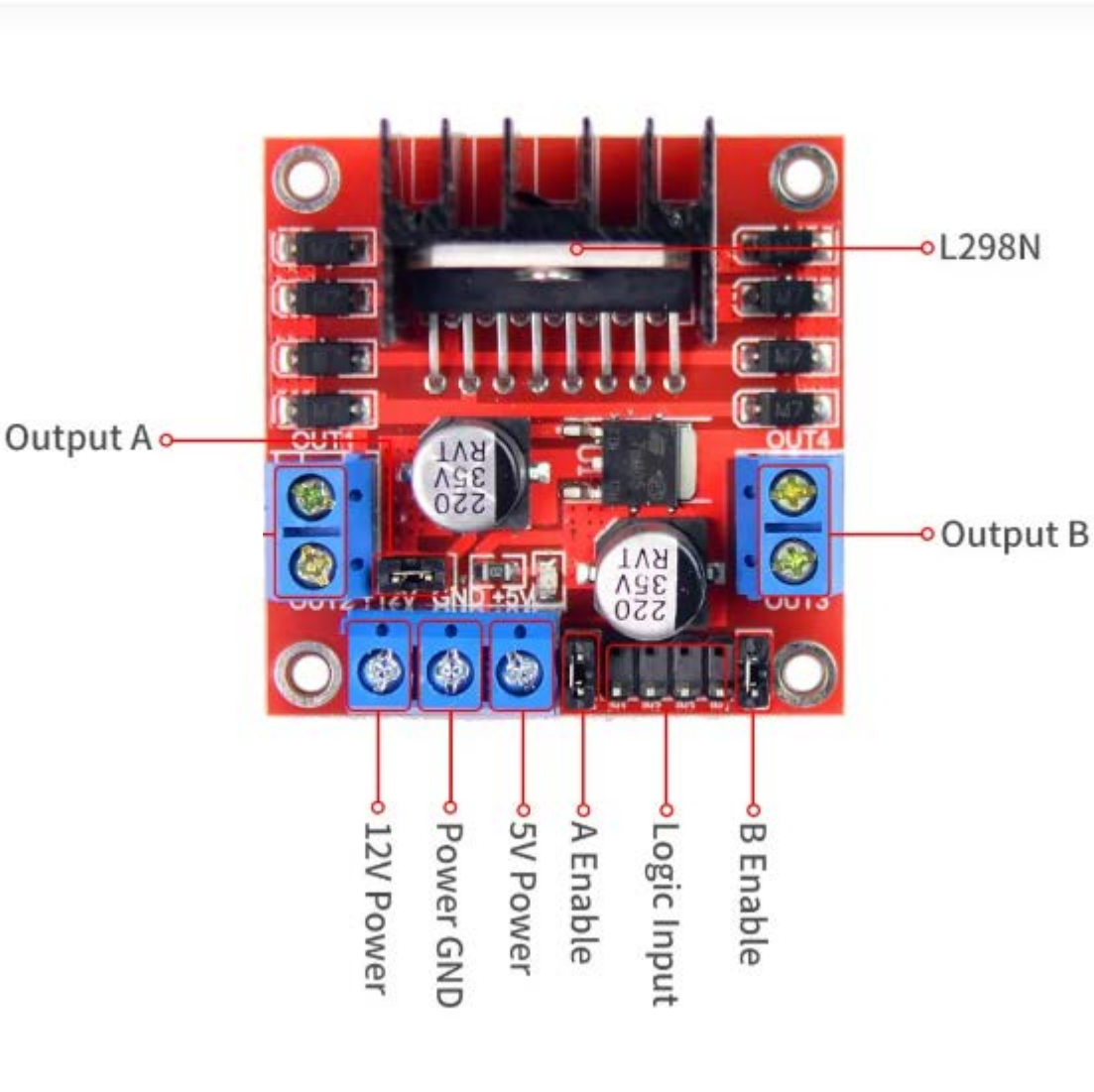


È chiaro che NPN e PNP funzionano diversamente

Nella configurazione in alto
Quando la base di T1 è a livello logico basso, T1 funge da interruttore chiuso (essendo PNP) così come T3 è interruttore chiuso se la base è a livello logico alto (essendo NPN)
Simmetricamente funziona la configurazione a destra.



Doppio ponte H integrato L298N



Programma Arduino: auto che va avanti indietro e si ferma

auto.ino

```
1  int in1 = 3;
2  int in2 = 6;
3  int in3 = 10;
4  int in4 = 11;
5  void setup() {
6    pinMode(in1, OUTPUT);
7    pinMode(in2, OUTPUT);
8  pinMode(in3, OUTPUT);
9    pinMode(in4, OUTPUT);
10 }
11 void AvantiA(){
12   digitalWrite(in1, HIGH);
13   digitalWrite(in2, LOW);
14 }
15 void AvantiB(){
16   digitalWrite(in3, HIGH);
17   digitalWrite(in4, LOW);
18 }
19 void IndietroA(){
20   digitalWrite(in1, LOW);
21   digitalWrite(in2, HIGH);
22 }
```

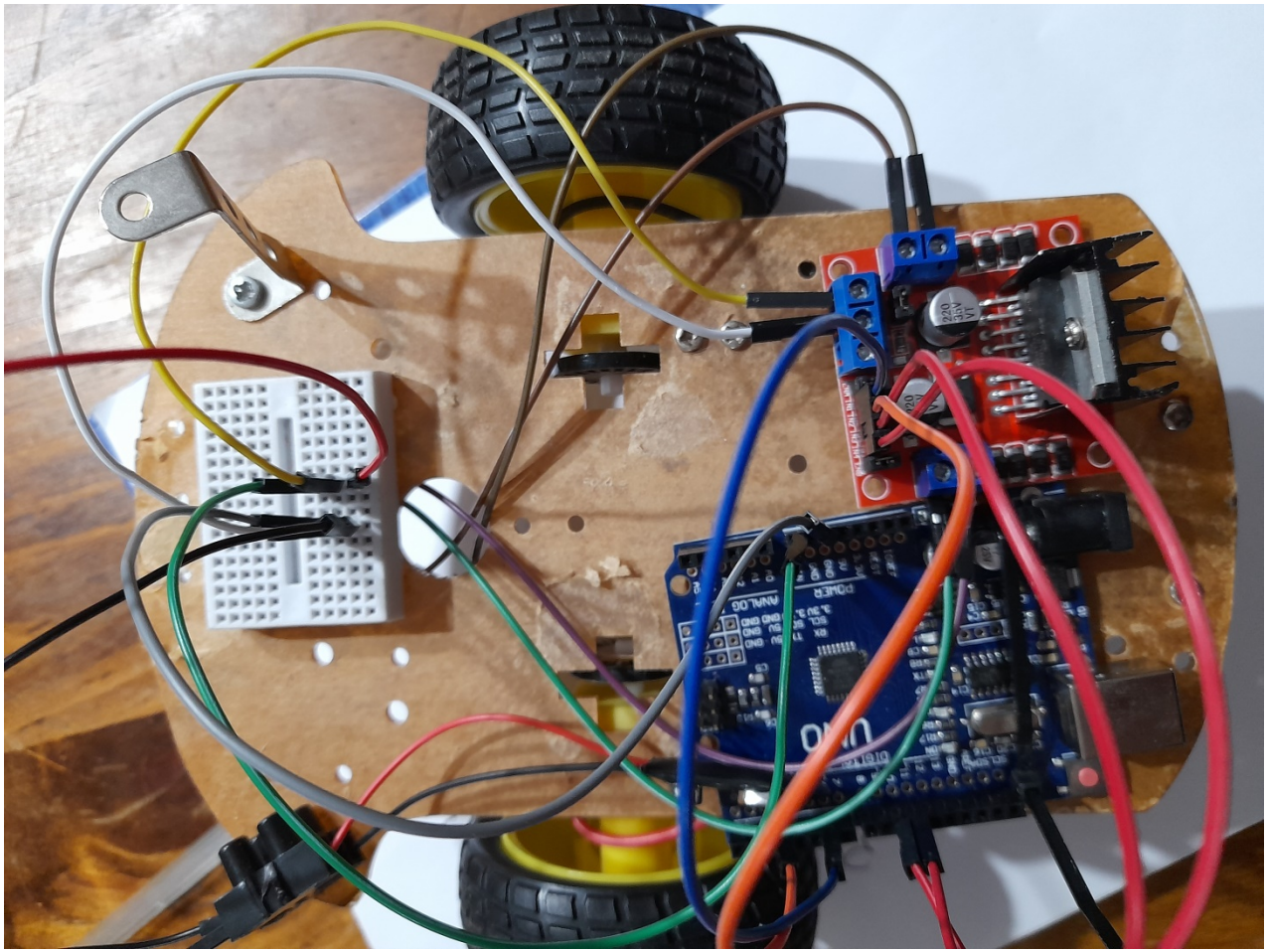
Output

Programma Arduino: auto che va avanti indietro e si ferma

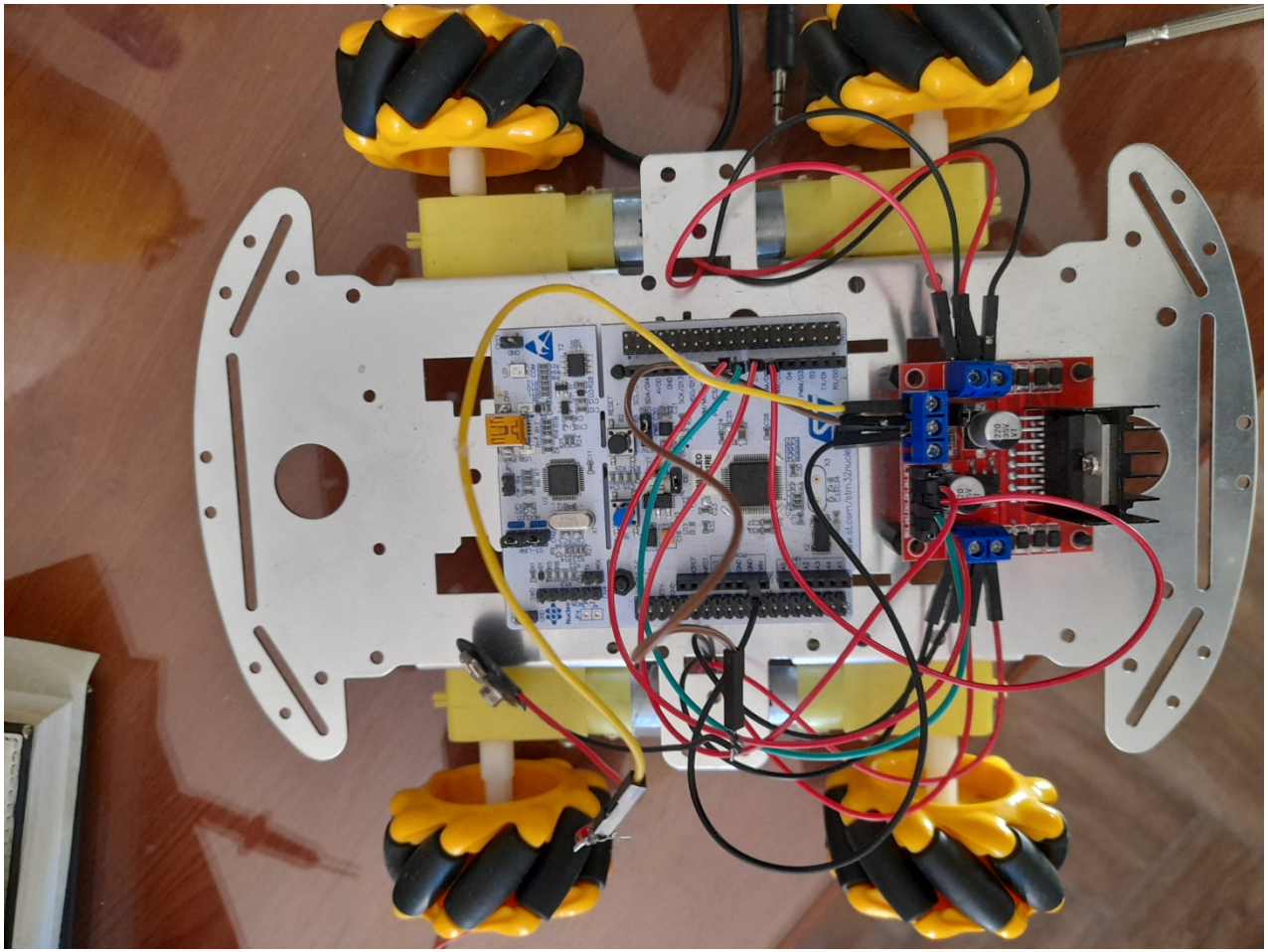
auto.ino

```
23  void IndietroB(){
24    digitalWrite(in3, LOW);
25    digitalWrite(in4, HIGH);
26  }
27  void OFFA(){
28    digitalWrite(in1, LOW);
29    digitalWrite(in2, LOW);
30  }
31  void OFFB(){
32    digitalWrite(in3, LOW);
33    digitalWrite(in4, LOW);
34  }
35  void loop() {
36    AvantiA();
37    AvantiB();
38    delay(3000);
39    OFFA();
40    OFFB();
41    delay(2000);
42  IndietroA();
43  IndietroB();
44  delay(3000);
```

Auto con Arduino UNO



Auto con stm32



main.cpp

```
1  //automobilina pilotata dai comandi bluet
2  #include "mbed.h"
3  #include "SoftSerial.h"
4  Serial pc(USBTX,USBRX);
5  DigitalOut aa(D5);
6  DigitalOut ab(D6);
7  DigitalOut ba(D9);
8  DigitalOut bb(D10);
9  SoftSerial bt(D2,D3);  //TX RX
10 char receive;
11 float vel;
12 int main()
13 {
14    while(1)
15    {
16         if (bt.readable()>0)
17         {
18
19              if(receive=='1'){
20                   bt.printf("ciao");
21                   aa=0;
22                   ab=1;
```

main.cpp

```
26                   if(receive=='2'){
27                   aa=1;
28                   ab=0;
29                   ba=1;
30                   bb=0;
31                   }
32                    if(receive=='0'){
33                   aa=0;
34                   ab=0;
35                   ba=0;
36                   bb=0;
37                   }
38                   if(receive=='3'){
39                   aa=0;
40                   ab=0;
41                   ba=0;
42                   bb=1;
43                   }
44                   if(receive=='4'){
45                   aa=0;
46                   ab=0;
47                   ba=1;
```