

Costruiamo il Semaforo Intelligente!

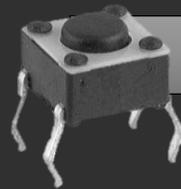


2) DOPPIO SEMAFORO CON
PULSANTE

Semaforo con Chiamata Pedoni



Abbiamo aggiunto al circuito realizzato in precedenza un micro-pulsante e 3 nuovi led per gestire un secondo semaforo rivolto ai pedoni. In questo modo i pedoni premendo il pulsante possono richiedere il verde del 2° semaforo e il rosso del 1° rivolto alle auto per poter attraversare in sicurezza! Precede l'accensione del rosso per le auto e del verde per i pedoni una fase di transito di durata 1-3sec in cui verrà acceso il led giallo per entrambi i semafori per segnalare la presenza di pedoni e l'accensione del rosso per le auto.

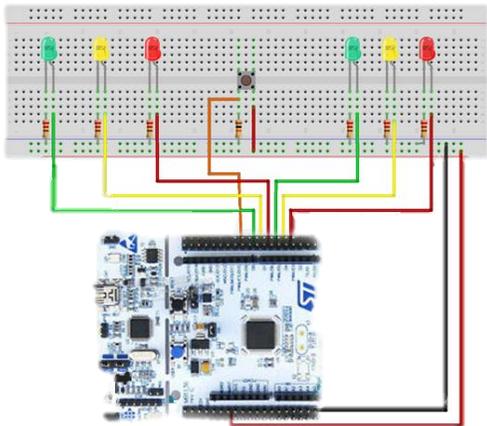


MICROPULSANTE

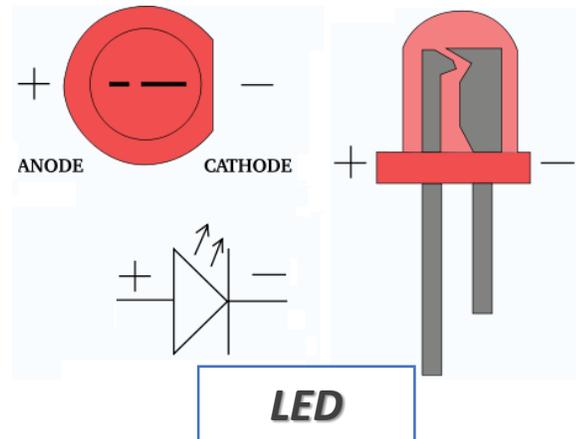


Circuito e Schema circuitale

Come nel circuito precedente disponiamo i 6 led che andranno a costituire i 2 semafori (auto e pedoni). Inseriamo successivamente nella breadboard il micropulsante collegando un'estremità del pulsante al 5V della scheda e l'altra estremità a un pin digitale di output della scheda. Inseriamo infine una resistenza da 1kohm tra la messa a terra della scheda e l'estremità del pulsante che avevamo precedentemente collegato ad un pin digitale di out, in questo modo permettiamo la scarica di eventuale energia accumulata dopo aver premuto il pulsante.



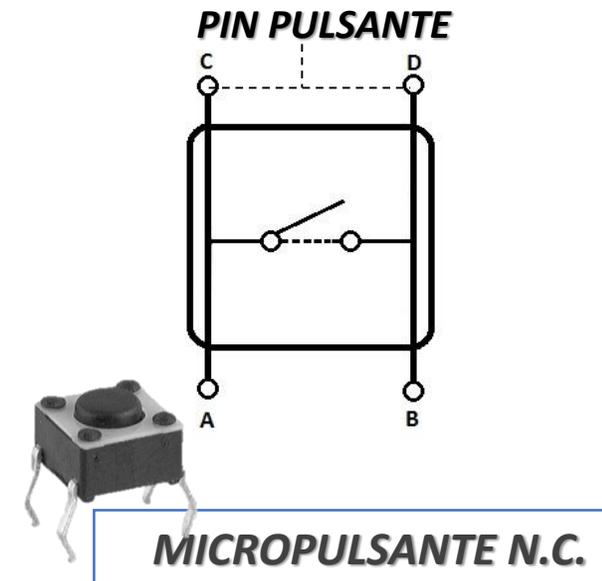
SCHEMA CIRCUITALE



LED

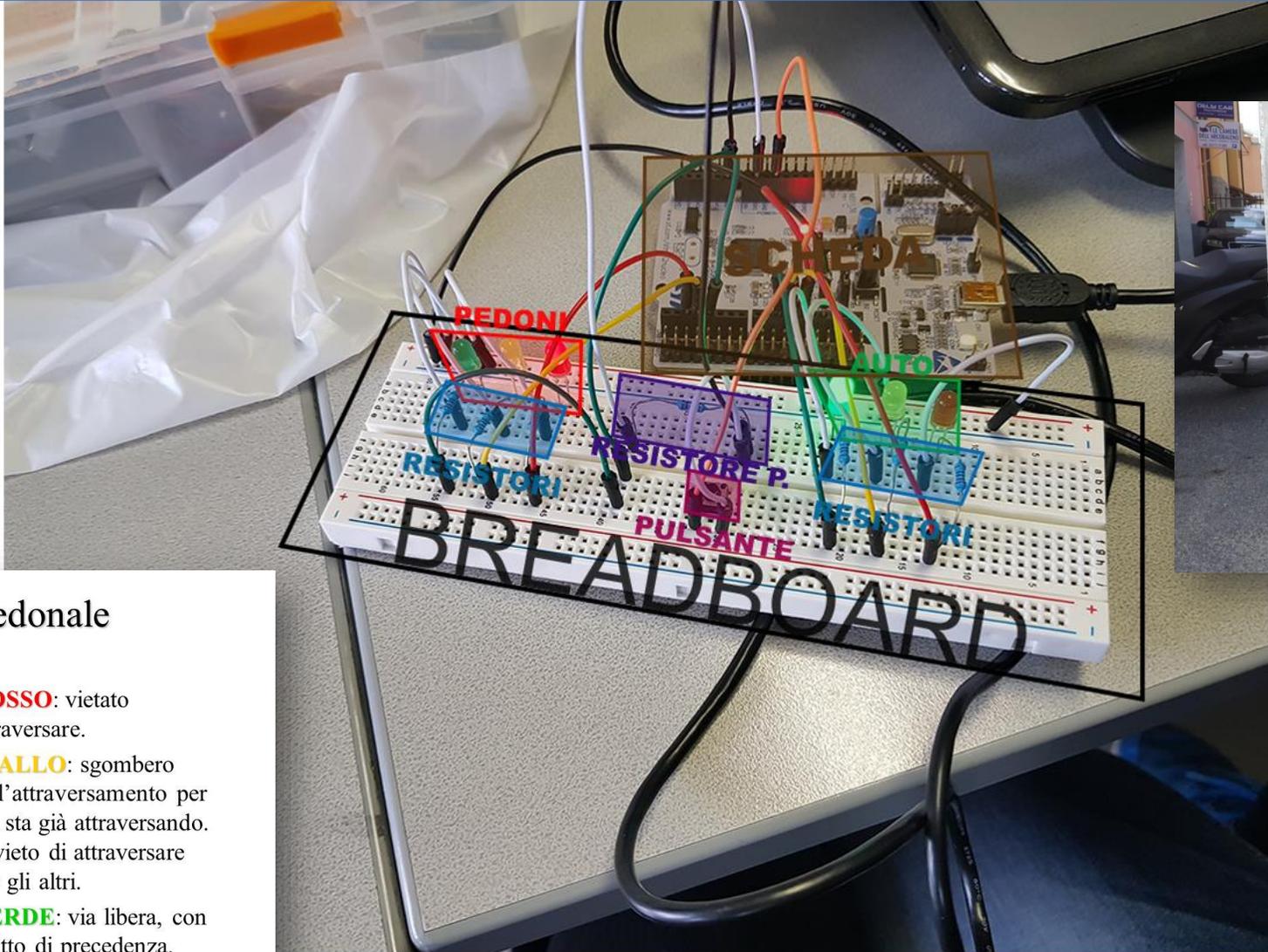


RESISTORE

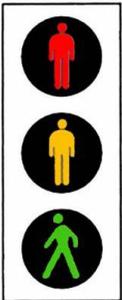


MICROPULSANTE N.C.

Collegamento alla Scheda!



Semaforo pedonale



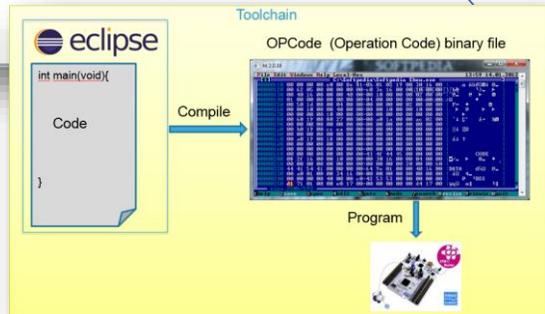
- **ROSSO**: vietato attraversare.
- **GIALLO**: sgombero dell'attraversamento per chi sta già attraversando. Divieto di attraversare per gli altri.
- **VERDE**: via libera, con diritto di precedenza.



Programmiamo la Scheda!

```
main.cpp x main.cpp x
1          /*DOPPIO SEMAFORO CON PULSANTE*/
2          //CAIAZZO ROMANO e SQUILLACE 3Ai
3
4 #include "mbed.h"          //include le librerie mbed
5
6 DigitalOut r_m(D12);      //LED ROSSO MACCHINE
7 DigitalOut g_m(D10);      //LED GIALLO MACCHINE
8 DigitalOut v_m(D9);       //LED VERDE MACCHINE
9
10 DigitalOut r_p(D2);       //LED ROSSO PEDONI
11 DigitalOut g_p(D4);       //LED GIALLO PEDONI
12 DigitalOut v_p(D6);       //LED VERDE PEDONI
13
14 DigitalIn button(D3);     //BOTTONE PEDONI  \\(USER_BUTTON)=>BOTTONE di DEFAULT
15
16 int main() {              //ESEGUI:
17 while(1){ //loop:
18     //ACCENDI VERDE MACCHINE
19     r_m=0;
20     g_m=0;
21     v_m=1;
22     //ACCENDI ROSSO PEDONI
23     r_p=1;
```

1



```
main.cpp x main.cpp x
24 g_p=0;
25 v_p=0;
26     if(button==1){ //QUANDO IL BOTTONE VIENE PREMUTO:
27
28         v_m=0; //SPEGNI VERDE MACCHINA
29         r_p=0; //SPEGNI ROSSO PEDONE
30
31         wait(1.0); //ASPETTA
32
33         g_m=1; //ACCENDI GIALLO MACCHINA
34
35         wait(2); //ASPETTA
36
37         r_m=1; //ACCENDI ROSSO MACCHINA
38         g_m=0; //SPEGNI GIALLO MACCHINA
39         g_p=1; //ACCENDI GIALLO PEDONI
40
41         wait(1.5); //ASPETTA
42
43         v_p=1; //ACCENDI VERDE PEDONE
44         g_p=0; //SPEGNI GIALLO PEDONE
45
46         wait(7); //ASPETTA
47     }
48 }
49 }
50 }
```

2