

SEMAFORO STRADALE

Lonzi Martina e Traino Sabrina
4BM
A.S 2023/2024

Sommario

| | |
|---|----|
| Descrizione (scopo) | 3 |
| Apparecchiatura e componenti utilizzati | 3 |
| Raspberry | 3 |
| Breadboard | 4 |
| Bottone | 4 |
| Led RGB | 5 |
| Resistenza | 6 |
| Procedimento | 6 |
| Rappresentazioni dei collegamenti realizzati | 9 |
| Codice | 11 |
| Osservazioni, conclusioni, possibili sviluppi | 13 |

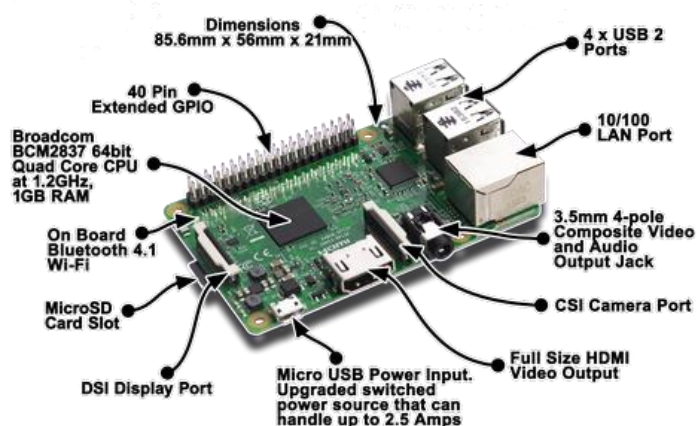
Descrizione (scopo)

Si pensi di realizzare due semafori composti ognuno da un led rosso, giallo e verde. Uno avrà il normale funzionamento in quanto sarà per le macchine e invece l'altro sarà dedicato ai pedoni. Quello dei pedoni non avrà il normale funzionamento, ma sarà sempre rosso finché non lo si prenota, cioè si clicca un pulsante. Allora l'altro diventerà rosso e ci resterà per più tempo, mentre l'altro diventerà verde. Questo meccanismo dovrà essere implementato in un incrocio con 4 semafori: due per i pedoni e due per le macchine.

Apparecchiatura e componenti utilizzati

Per la realizzazione di questo progetto è stato utilizzato: una scheda Raspberry, una breadboard, 4 led RGB, due bottoni, resistenze e fili di collegamento.

Raspberry



La Raspberry Pi è un microcontrollore a scheda singola che offre una piattaforma di calcolo completa in un formato estremamente compatto. È dotata di tutte le componenti essenziali di un computer tradizionale, tra cui CPU, RAM, connettività di rete, porte USB, uscite video e GPIO (General Purpose Input/Output) per l'interfacciamento con dispositivi esterni.

Può essere utilizzato per una vasta gamma di scopi, come l'elaborazione dati, la programmazione, l'automazione domestica, la creazione di progetti di robotica, l'Internet delle cose (IoT), l'insegnamento della programmazione e molto altro ancora. In

sostanza, il Raspberry Pi è una piattaforma flessibile e accessibile che consente agli utenti di realizzare una varietà di progetti e applicazioni informatiche.

Di solito, una volta completate le impostazioni, si collegano mouse e tastiera alle porte USB del Pi e il monitor via HDMI. Ciò provoca generalmente un avvio con il sistema operativo selezionato e la corrispondente GUI, interfaccia grafica utente. Alcuni progetti, però, non necessitano che il Raspberry Pi utilizzi una normale GUI. In questi casi, è possibile collegare il Pi alla rete e accedervi da remoto tramite il protocollo SSH (Secure Shell).

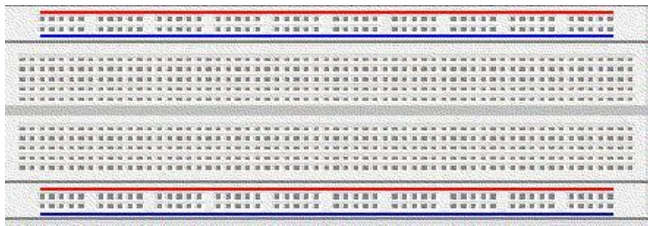
Inoltre bisogna prestare attenzione ai pin quando si scrive il codice perché il numero del GPIO non corrisponde al pin associato.



Raspberry Pi 3 B+ GPIO Header

| Pin# | NAME | NAME | Pin# |
|------|------------------------------------|------------------------------------|------|
| 01 | 3.3v DC Power | DC Power 5v | 02 |
| 03 | GPIO02 (SDA1, I ² C) | DC Power 5v | 04 |
| 05 | GPIO03 (SCL1, I ² C) | Ground | 06 |
| 07 | GPIO04 (GPIO_GCLK) | (TXD0) GPIO14 | 08 |
| 09 | Ground | (RXD0) GPIO15 | 10 |
| 11 | GPIO17 (GPIO_GEN0) | (GPIO_GEN1) GPIO18 | 12 |
| 13 | GPIO27 (GPIO_GEN2) | Ground | 14 |
| 15 | GPIO22 (GPIO_GEN3) | (GPIO_GEN4) GPIO23 | 16 |
| 17 | 3.3v DC Power | (GPIO_GEN5) GPIO24 | 18 |
| 19 | GPIO10 (SPI_MOSI) | Ground | 20 |
| 21 | GPIO09 (SPI_MISO) | (GPIO_GEN6) GPIO25 | 22 |
| 23 | GPIO11 (SPI_CLK) | (SPI_CE0_N) GPIO08 | 24 |
| 25 | Ground | (SPI_CE1_N) GPIO07 | 26 |
| 27 | ID_SD (I ² C ID EEPROM) | (I ² C ID EEPROM) ID_SC | 28 |
| 29 | GPIO05 | Ground | 30 |
| 31 | GPIO06 | GPIO12 | 32 |
| 33 | GPIO13 | Ground | 34 |
| 35 | GPIO16 | GPIO16 | 36 |
| 37 | GPIO26 | GPIO20 | 38 |
| 39 | Ground | GPIO21 | 40 |

Breadboard

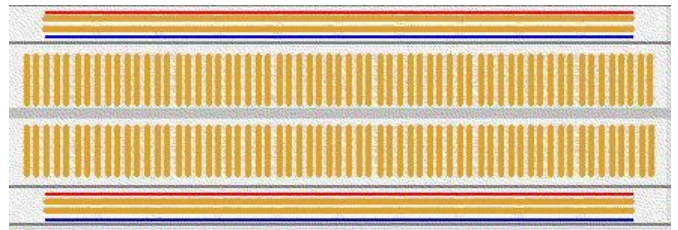


La breadboard o basetta è uno strumento ampiamente usato nell'elettronica per assemblare e testare circuiti temporanei senza la necessità di saldature. È composta da una serie di fori in cui si possono inserire i componenti elettronici e i cavi di collegamento. Le sue parti principali sono:

1. **Striscia di alimentazione:** Le strisce di alimentazione corrono lungo i bordi e sono utilizzate per distribuire l'alimentazione elettrica ai vari componenti inseriti sulla breadboard. Generalmente, queste strisce sono colorate con una linea rossa per indicare il positivo (+) e una linea blu o nera per il negativo o terra (GND).

2. **Area centrale:** La zona centrale è divisa in due parti principali, separate da un canale centrale. Ogni lato è costituito da colonne di fori connessi tra di loro elettricamente in fila, ideali per inserire i piedini dei componenti.

3. **Connessioni interne:** All'interno della breadboard, i fori sono collegati tramite conduttori metallici. Nelle strisce di alimentazione, i fori sono connessi lungo tutta la lunghezza della striscia, mentre nell'area centrale, ogni fila di 5 fori è elettricamente connessa tra loro, ma isolata dalle altre file. Questo permette di inserire i componenti e di collegarli facilmente senza cortocircuiti.



Bottone



Il pulsante è un dispositivo elettronico a due stati (ON, OFF) con una sola posizione monostabile. Nel caso specifico i pulsanti permettono di aprire o chiudere un circuito e pertanto collegare a GND (0V) o a VCC (5V) una specifica uscita.

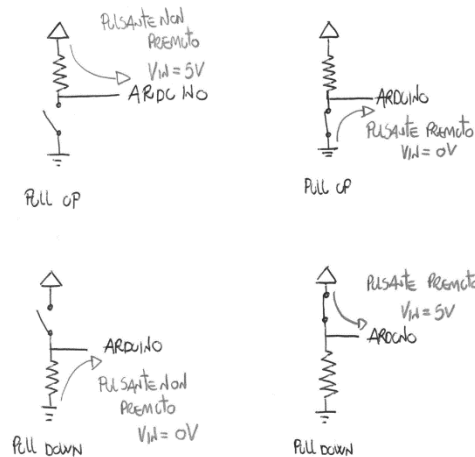
L'utilizzo di una resistenza è indispensabile al fine di collegare correttamente il pulsante evitando cortocircuiti. A seconda del collegamento realizzato, la resistenza prende il nome di:

1. Resistenza di **Pull Up**: la resistenza viene collegata direttamente all'alimentazione.
2. Resistenza di **Pull Down**: la resistenza viene collegata a massa.

Il comportamento del circuito e la tensione letta dal microcontrollore dipendono dalla tipologia di collegamento circuitale utilizzato. In particolare:

- Resistenza di **Pull Up**:
 - se il pulsante viene premuto la tensione in ingresso è pari a 0.
 - se il pulsante non viene premuto la tensione in ingresso è pari a Vcc (5V)
- Resistenza di **Pull Down**:
 - se il pulsante viene premuto la tensione in ingresso è pari a Vcc (5V).

- se il pulsante non viene premuto la tensione in ingresso è pari a 0.

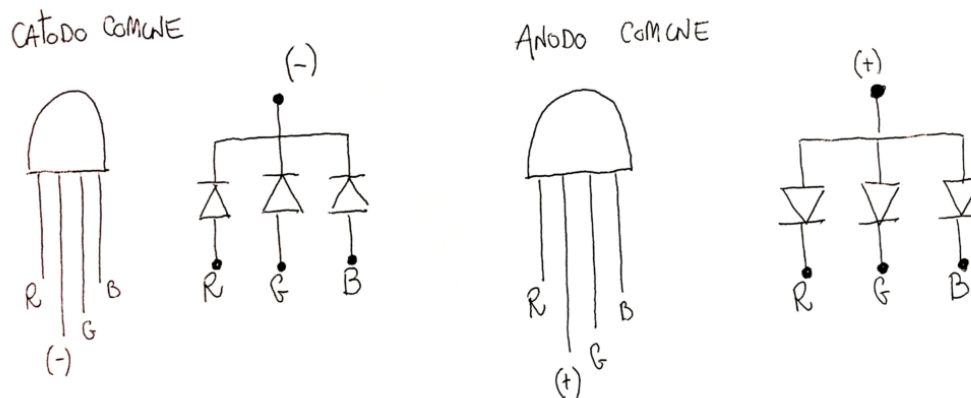


Led RGB

Gli RGB rappresentano una particolare sottocategoria di LED capaci di riprodurre differenti colori attraverso la combinazione dei colori fondamentali Rosso Verde e Blu. Nel dettaglio, questi dispositivi sono costituiti da tre differenti led azionati in funzione del colore da riprodurre.

I LED RGB sono dotati di 4 differenti morsetti e sono così classificati:

- **RGB ad Anodo Comune:** il pin più lungo del LED deve essere collegato a massa (GND) mentre gli altri 3 differenti PIN sono utilizzati per comandare i 3 LED (rosso, verde, blue). Nel caso in questione i singoli LED sono accesi attraverso l'istruzione Arduino `digitalWrite(pinLed, HIGH);`
- **RGB a Catodo Comune:** il pin più lungo del LED deve essere collegato a VCC (5V) mentre gli altri 3 differenti PIN sono utilizzati per comandare i 3 LED (rosso, verde, blue). Nel caso in questione i singoli LED sono accesi attraverso l'istruzione Arduino `digitalWrite(pinLed, LOW);`



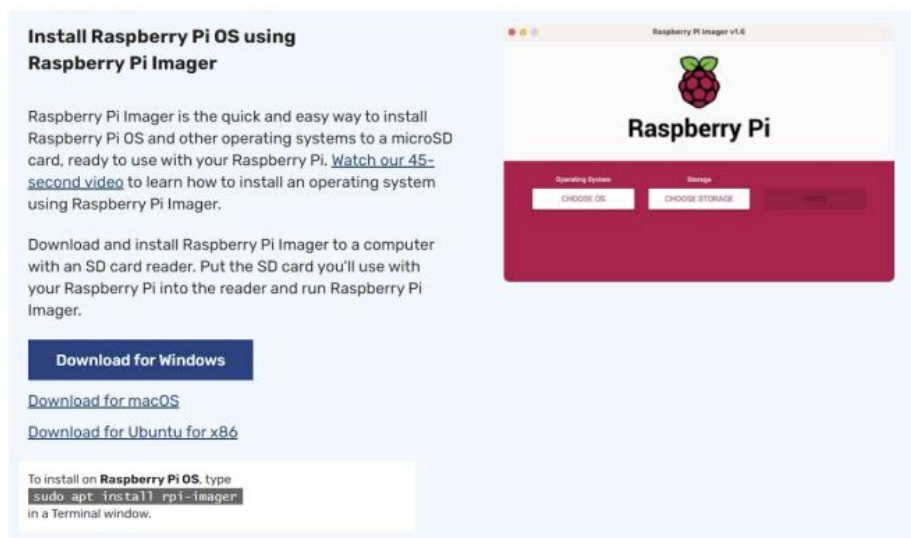
Resistenza



Le resistenze sono componenti elettronici fondamentali che si presentano sotto forma di piccoli cilindri con due terminali utilizzati per limitare il flusso di corrente in un circuito elettrico. Il principio di base delle resistenze è che, quando una corrente elettrica attraversa il componente, il materiale dalla quale è fatta impedisce il libero flusso di elettroni, causando una "resistenza" al passaggio della corrente. Questa resistenza è misurata in Ohm e determina quanto il flusso di corrente è ridotto mentre attraversa la resistenza.

Procedimento

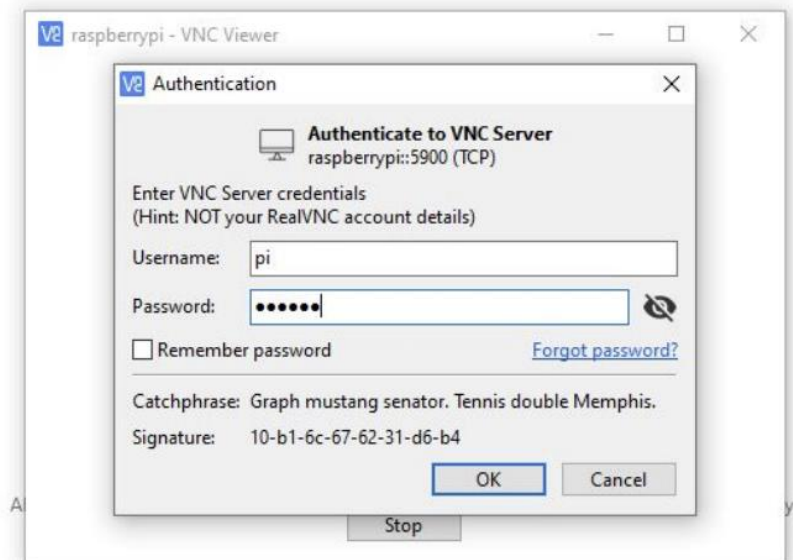
1. Recupero di una scheda micro-SD per lo scaricamento e l'installazione di Raspberry tramite Raspberry Pi imager.
 1. Scaricare ed installare il software "Raspberry Pi imager" dal sito ufficiale Raspberry <https://www.raspberrypi.com/software/>



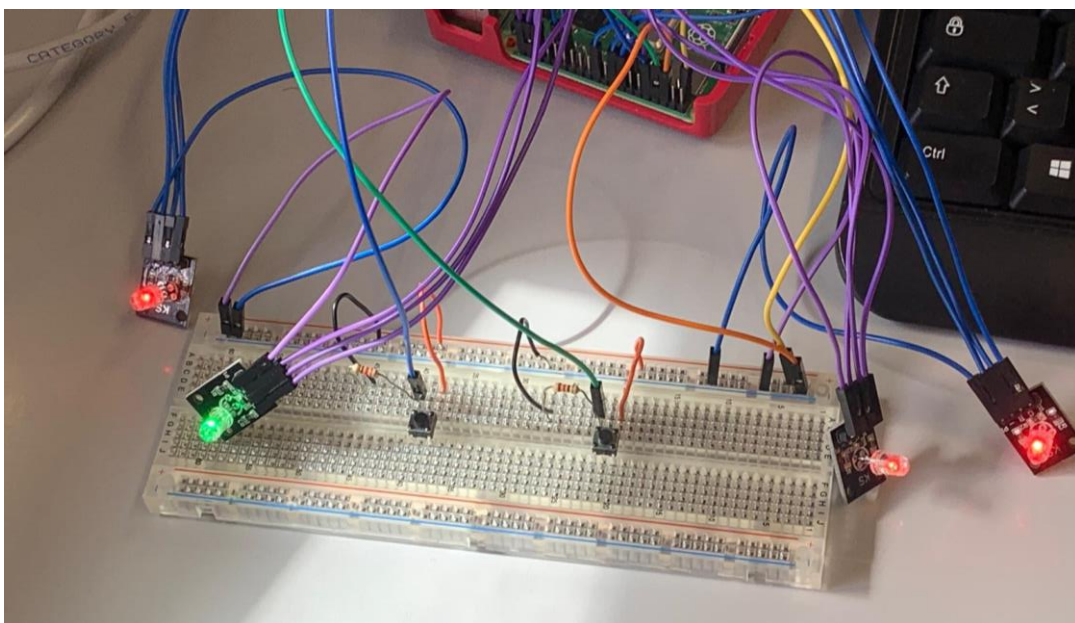
2. Dalla schermata di Raspberry Pi imager selezionare il sistema operativo Raspberry Pi OS (32 bit) e la scheda SD opportunamente inserita nella quale si vuole installare il SO



2. Inserimento della scheda micro-SD con al suo interno il sistema operativo precedentemente installato nel microcontrollore.
3. Connessione del cavo Ethernet alla scheda Raspberry Pi per avere la connessione internet e successivamente si collega anche il filo per l'alimentazione.
4. Configurazione dell'applicazione REALVNC in modo da poter utilizzare il microcontrollore da remoto senza collegare tutte le periferiche.
5. Connessione tra REALVNC con la scheda Raspberry tramite username e password utilizzate per scaricare Raspberry.



6. Assemblaggio dei componenti sulla breadboard e dei relativi collegamenti con la scheda.



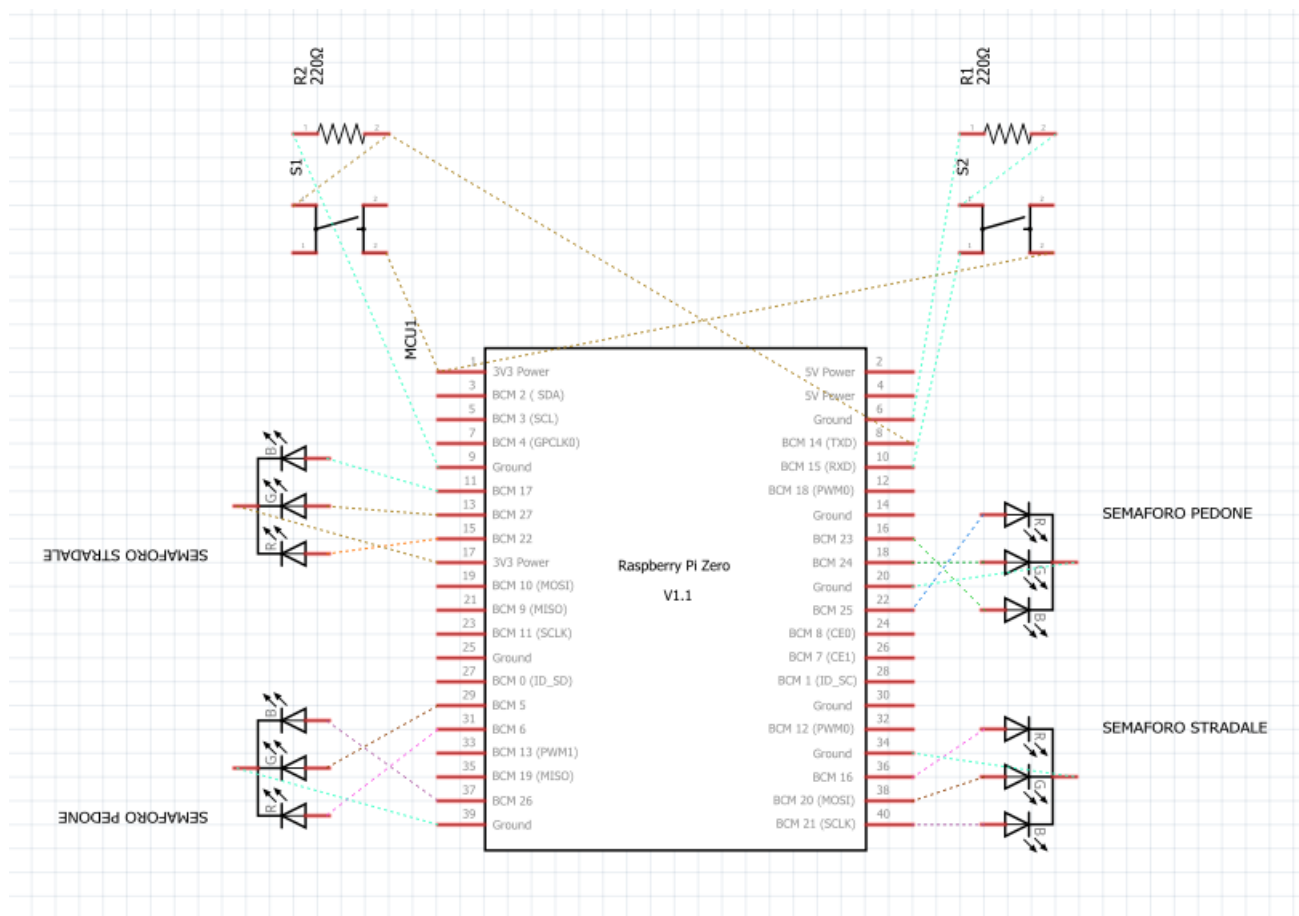
7. Apertura da remoto del programma Thonny che permette la realizzazione e la simulazione di programmi in linguaggio Python.
8. Simulazione e verifica del funzionamento del gioco da realizzare.

VIDEO DEL FUNZIONAMENTO COMPLETO DEL SEMAFORO

<https://youtu.be/tb5rYnQGjn8>

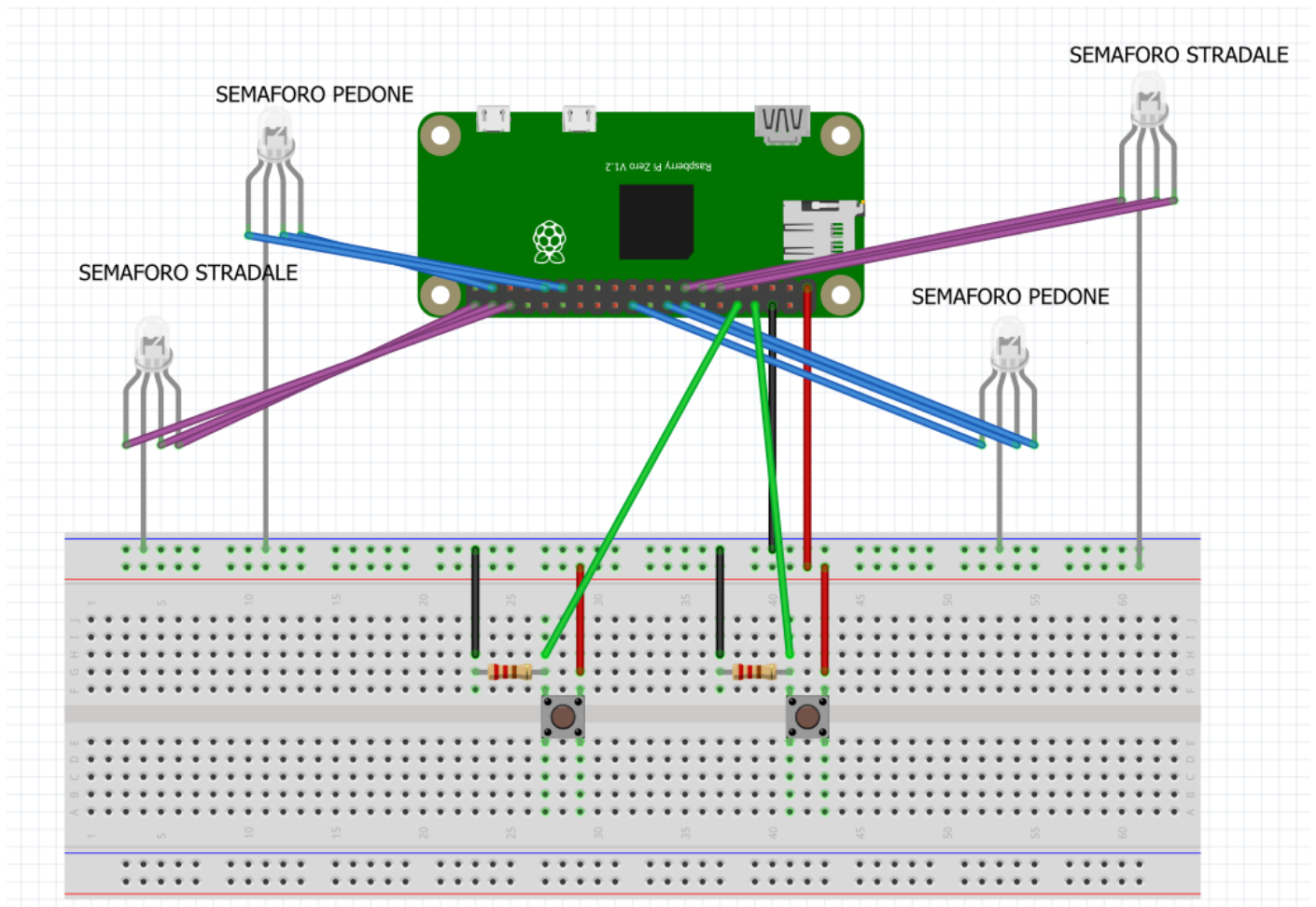
Rappresentazioni dei collegamenti realizzati

Circuito simbolico



In questo circuito notiamo che al led è stata inserita una resistenza che serve a non farlo fondere. Inoltre, possiamo vedere che il collegamento utilizzato per i bottoni è la Resistenza pull down: in quanto se il pulsante viene premuto la tensione in ingresso alla scheda è pari a V_{cc} (5V) altrimenti se il pulsante non viene premuto la tensione in ingresso è pari a 0. Successivamente possiamo osservare che la tipologia di led RGB utilizzato è di tipo catodo comune

Circuito su basetta



Codice

```
import RPi.GPIO as GPIO
import time

#setto la modalità dei pin con il loro effettivo nome
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
# setto i pin in input e output
#bottoni
#bottone sinistro
GPIO.setup(15, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#bottone destro
GPIO.setup(14, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# led destro strada
#blue
semaforoDestroStradaBlue=17
GPIO.setup(17, GPIO.OUT)
#green
semaforoDestroStradaGreen=27
GPIO.setup(27, GPIO.OUT)
#red
semaforoDestroStradaRed=22
GPIO.setup(22, GPIO.OUT)

#led sinistro pedonale
#blue
semaforoSinistroPedoneBlue=26
GPIO.setup(26,
GPIO.OUT)
#green
semaforoSinistroPedoneGreen=5
GPIO.setup(5, GPIO.OUT)
#red
semaforoSinistroPedoneRed=6
GPIO.setup(6, GPIO.OUT)

#led sinistro strada
#blue
semaforoSinistroStradaBlue=21
GPIO.setup(21,
GPIO.OUT)
#green
semaforoSinistroStradaGreen=20
GPIO.setup(20,
GPIO.OUT)
#red
semaforoSinistroStradaRed=16
GPIO.setup(16,
GPIO.OUT)

# led destro pedonale
#blue
semaforoDestroPedoneBlue=23
GPIO.setup(23,
GPIO.OUT)
#green
semaforoDestroPedoneGreen=24
GPIO.setup(24,
GPIO.OUT)
#red
semaforoDestroPedoneRed=25
GPIO.setup(25,
GPIO.OUT)

# sottoprogrammi
def
semaforoVerde(pinRosso,
pinVerde, pinBlu):
    #verde

GPIO.output(pinVerde,
GPIO.HIGH)

GPIO.output(pinRosso,
GPIO.LOW)
    GPIO.output(pinBlu,
GPIO.LOW)

#codice main

def
semaforoRosso(pinRosso,
pinVerde, pinBlu):
    #rosso

GPIO.output(pinRosso,
GPIO.HIGH)

GPIO.output(pinVerde,
GPIO.LOW)
    GPIO.output(pinBlu,
GPIO.LOW)

def
semaforoGiallo(pinRosso,
pinVerde, pinBlu):
    #verde

GPIO.output(pinVerde,
GPIO.HIGH)

GPIO.output(pinRosso,
GPIO.HIGH)
    GPIO.output(pinBlu,
GPIO.LOW)
```

```
while (True):
    #schiacciato il bottone sinistro prima che si riaccenda il rosso
    if GPIO.input(15)==GPIO.HIGH:
        #semaforo sinistra della strada diventa rosso, mentre quello destro resta
verde
        semaforoRosso(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        semaforoVerde(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        #il semaforo pedonale si attiva
        semaforoVerde(semaforoSinistroPedoneRed, semaforoSinistroPedoneGreen,
semaforoSinistroPedoneBlue)
        time.sleep(10)
        semaforoGiallo(semaforoSinistroPedoneRed, semaforoSinistroPedoneGreen,
semaforoSinistroPedoneBlue)
        time.sleep(3)
        semaforoRosso(semaforoSinistroPedoneRed, semaforoSinistroPedoneGreen,
semaforoSinistroPedoneBlue)
        semaforoVerde(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        semaforoRosso(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        #se viene schiacciato il tasto destro prima che si riaccenda il rosso
    elif GPIO.input(14)==GPIO.HIGH:
        #semaforo destro della strada diventa rosso, mentre quello sinistro resta
verde
        semaforoRosso(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        semaforoVerde(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)

        #il semaforo pedonale si attiva
        semaforoVerde(semaforoDestroPedoneRed, semaforoDestroPedoneGreen,
semaforoDestroPedoneBlue)

        time.sleep(10)
        semaforoGiallo(semaforoDestroPedoneRed, semaforoDestroPedoneGreen,
semaforoDestroPedoneBlue)
        time.sleep(3)
        semaforoRosso(semaforoDestroPedoneRed, semaforoDestroPedoneGreen,
semaforoDestroPedoneBlue)
        semaforoRosso(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        semaforoVerde(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
    else:
        #semaforo destro strada Rosso
        semaforoRosso(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        #semaforo sinistro strada verde
        semaforoVerde(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        time.sleep(7)
        #semaforo destro strada Giallo
        semaforoGiallo(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        #semaforo sinistro strada Giallo
        semaforoGiallo(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
```

```
        time.sleep(7)
        #semaforo destro strada verde
        semaforoVerde(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        #semaforo sinistro strada rosso
        semaforoRosso(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        time.sleep(7)
        semaforoGiallo(semaforoDestroStradaRed, semaforoDestroStradaGreen,
semaforoDestroStradaBlue)
        semaforoGiallo(semaforoSinistroStradaRed, semaforoSinistroStradaGreen,
semaforoSinistroStradaBlue)
        time.sleep(7)
```

Osservazioni, conclusioni, possibili sviluppi

Non si hanno riscontrati molti problemi a parte capire il meccanismo di funzionamento dei led RGB e nell'implementazione del bottone all'interno del codice.

Possibili sviluppi possono essere l'implementazione di altoparlanti con diversi suoni in base al colore del semaforo per i non vedenti e implementare questo meccanismo con un incrocio più complesso con ipoteticamente 4 semafori per le macchine e 4 per i pedoni