



**Makerspace mobili per promuovere l'educazione dei maker nelle scuole
democratizzare l'istruzione STEAM e lo sviluppo dell'innovazione per tutti gli
studenti**

IO2

Linee guida per gli insegnanti

**Sistema di controllo semaforico con Raspberry Pi e
Scratch**

Livello di difficoltà: basso

Età di riferimento: dai 14 ai 18 anni

Costo: attualmente piuttosto elevato

Realizzato da: IIS Leonardo da Vinci

Autori: Renato Gatti & Loris Ranalli



Erasmus+ GRANT NUMBER: 2020-1-PL01-KA201-081698

Dichiarazione

Questo file è stato preparato nell'ambito del progetto STEAM2GO. Laddove sono stati utilizzati altri materiali originali pubblicati e inediti, questi sono stati riconosciuti.

Diritto d'autore

© Copyright 2020 - 2022 il Consorzio [STEAM2GO](#)



Tutti i diritti riservati.



Questo documento è concesso in licenza al pubblico ai sensi di Attribuzione Creative Commons-Non commerciale - Licenza internazionale No Derivatives 4.0.

Esclusione di responsabilità per il finanziamento

Questo progetto è stato finanziato con il sostegno della Commissione Europea. Questo rapporto riflette solo il punto di vista dell'autore e la Commissione non può essere ritenuta responsabile per qualsiasi uso che possa essere fatto delle informazioni in esso contenute.

Linee guida per gli insegnanti

Sistema di controllo semaforico con Raspberry Pi

Sommario

Linee guida per gli insegnanti.....	1
Note introduttive	4
Studio e realizzazione dei diagrammi temporali.....	5
Configuriamo la scheda Raspberry Pi	7
Costruiamo le basi del nostro programma "Semafori" con scratch	8
Costruiamo il nostro semaforo con Scratch in modalità desktop sul PC	10
Costruiamo il nostro semaforo con dispositivi esterni e i GPIOs	14
La breadboard.....	15
Disegno e realizzazione della planimetria del raccordo a due vie.....	20
Note sui collegamenti di pulsanti e interruttori	24
Note sulle resistenze di pull-up e pull-down.....	25
File di supporto.....	26
Guide Raspberry Pi e Scratch.....	26

Note introduttive

Informazioni sul progetto

Inizieremo con la programmazione di un prototipo realizzato su singola bread board. Successivamente si potrà realizzare un modello dell'incrocio su carta o cartone che verrà appoggiato su un foglio di polistirolo dove saranno praticati dei fori rettangolari, di adeguate dimensioni, per alloggiare le quattro breadboard da 170 contatti. Queste verranno poi collegate alla breadboard con Raspberry Pi, pulsante e interruttore. I vari dispositivi andranno poi collegati ai relativi GPIO della scheda Raspberry Pi e, infine, verrà programmato l'intero sistema.

Obiettivi formativi

Gli studenti coinvolti nel progetto avranno la possibilità di:

- Comprendere e implementare diagrammi di temporizzazione
- Comprendere dettagliatamente un circuito elettronico
- Creare circuiti utilizzando Raspberry Pi e componenti elettronici
- Creare una planimetria di un impianto realizzando un modello in carta utilizzando materiali di uso comune
- Imparare i fondamenti della programmazione per implementare un modello proposto
- Analizzare e identificare i parametri e le funzioni da controllare in un sistema complesso
- Interfacciare, attraverso i GPIO di Raspberry Pi, i segnali di I/O provenienti dai vari dispositivi
- Sviluppare, tramite il software Scratch, un complesso sistema di controllo
- Testare il prototipo
- Ottimizzare e definire la soluzione di lavoro ottimale condividendola tra i partecipanti

Collegamento ad altre aree tematiche e/o concetti STEM specifici

- Matematica
- Tecnologia
- Elettronica
- Informatica
- Realizzazione di manufatti relativi ad un progetto tecnico-scientifico

Hardware e software di cui potresti aver bisogno

Hardware	Software	Materiale artigianale
- 3 x breadboard 170 punti - https://amzn.eu/d/bcFYyKs - 1 x RPi 4 (o 3) - 1 pulsante - 1 interruttore SPDT - Jumper wires kit - Wiring cables kit (M/M and M/F) - 2 x kit singolo semaforo a led preassemblato: https://www.amazon.it/ZHITING-Display-semaforo-Mini-semaforo-elettronici/dp/B0859ZPM47	- Excel - Word - Scratch	- Alcuni fogli di carta bianca o cartoncino formato A4 (o A3). - Un foglio di polistirolo espanso - Un foglio di polipropilene alveolare - 1 taglierino - 1 forbice - 1 colla - 1 nastro biadesivo - 1 matita - 1 temperamatite - Alcuni pennarelli indelebili di vari colori - Attrezzatura di cancelleria come matite, penne, fogli di carta

Le fasi di lavoro

1. Il docente può scegliere, in base al livello degli studenti, una delle opzioni presentate sia per la parte di programmazione che per la realizzazione del piano dell'intersezione stradale.
2. La fase iniziale sarà l'identificazione del corretto funzionamento di un controllo semaforico. Particolare attenzione dovrà essere prestata alla tempistica dei segnali e alle misure di sicurezza da adottare. Questo può essere fatto attraverso una rapida ricerca su Internet guidata dal docente.
3. Successivamente si dovrà studiare il diagramma temporale dell'impianto, inizialmente nel modo più semplice possibile e successivamente tenendo conto di un tempo di sovrapposizione della luce rossa, per garantire che l'incrocio sia libero da eventuali veicoli.
4. Configurazione della scheda Raspberry Pi.
5. Creazione del primo programma operativo utilizzando Scratch.
6. Effettuare tutte le interconnessioni necessarie tra i GPIO della scheda Raspberry Pi e i vari componenti esterni
7. Trasferimento del programma dal PC alla scheda Raspberry Pi.
8. Progettare e realizzare una planimetria dell'intersezione stradale dove verranno posizionate le breadboard e i vari componenti
9. Testare il corretto funzionamento del sistema

Studio e realizzazione dei diagrammi temporali

Il primo passo per realizzare il progetto è analizzare e creare un diagramma con le temporizzazioni dei vari segnali di ogni semaforo. Per una corretta comprensione di ogni passaggio, è necessario partire dal diagramma temporale più semplice. Una volta completato, i pro e i contro di ognuno verranno portati all'attenzione degli studenti.

I diagrammi temporali possono essere realizzati su carta a quadretti oppure utilizzando un foglio di calcolo o un qualsiasi programma di grafica.

a) Diagramma di temporizzazione semplice, senza sovrapposizione delle luci rosse

La sequenza di ogni semaforo sarà, nell'esempio riportato in Fig.1:

- Verde: 4 sec.
- Giallo: 1 sec.
- Rosso: 5 sec. (il tempo del segnale rosso deve essere uguale alla somma del verde e del giallo del semaforo opposto)

In questo caso si assume che si accendano contemporaneamente il semaforo verde del semaforo [1], che chiameremo G1, e il semaforo rosso del semaforo opposto [2], che chiameremo R2.

Trascorso invece il tempo programmato, si accendono contemporaneamente la luce verde del semaforo [2], che chiameremo G2, e la luce rossa del semaforo opposto [1], che chiameremo R1. Fare riferimento alla **Figura 1**

b) Diagramma di temporizzazione con sovrapposizione delle luci rosse

In questo caso abbiamo considerato la necessità di concedere un po' di tempo alla sovrapposizione delle due luci rosse prima di attivare il verde. Questo per essere sicuri che, una volta acceso il verde, non ci siano più veicoli nell'incrocio.

Questo è il principale criterio di sicurezza, richiesto dalle norme stradali e dal codice della strada.

Ora la sequenza di ogni semaforo sarà, nell'esempio riportato in **Figura 2**:

- Verde: 4 sec.
- Giallo: 1 sec.
- Rosso: 7 sec. (il tempo del segnale rosso deve essere uguale alla somma del verde e del giallo del semaforo opposto più il tempo di sovrapposizione che abbiamo assunto in 1 sec per lato)

Figura 1

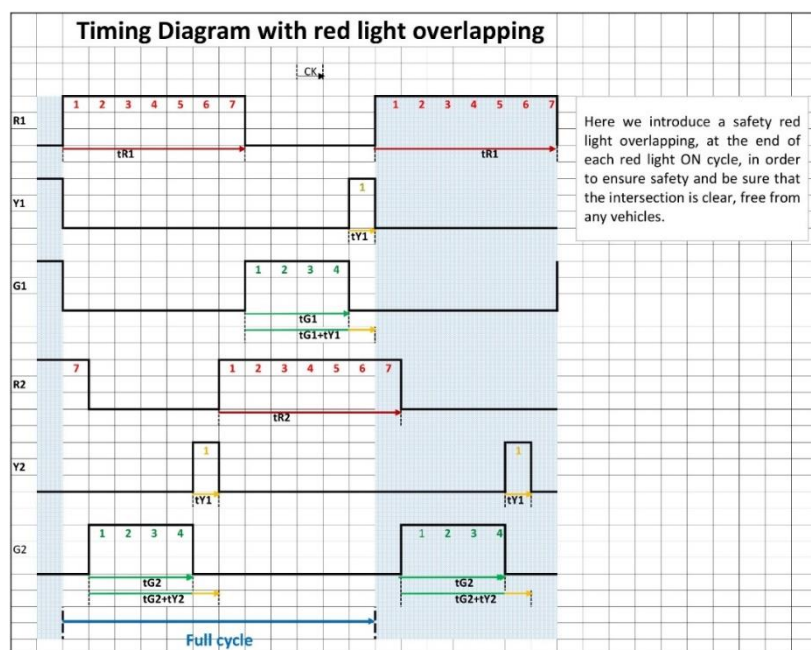
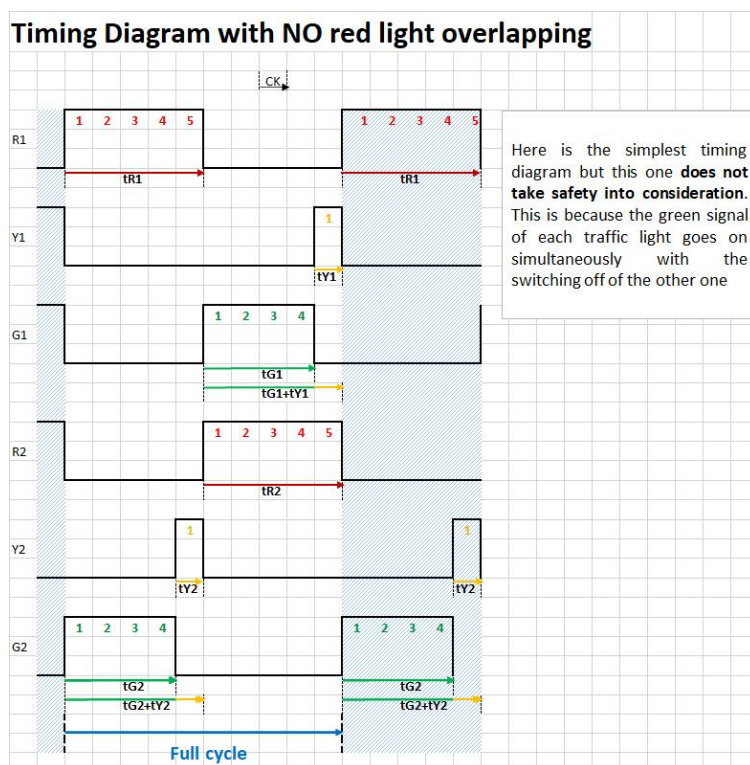


Figura 2

L'insegnante può intervenire sollevando questioni di sicurezza, ad esempio:

- È necessaria una sovrapposizione di luci rosse? Se è così, perché?
- Per quanto tempo le luci rosse dovrebbero sovrapporsi se necessario?
- In base a quali parametri calcolereste il tempo di sovrapposizione dei semafori rossi?
- Nel caso di un semaforo pedonale, come regolereste i vari orari del semaforo?
- Ecc.

Cliccando sul link sottostante è possibile scaricare i Diagrammi di Temporizzazione in vari formati:

Link ai Diagrammi temporali in formato MS Excel:

- [Diagramma di temporizzazione più semplice senza sovrapposizione di luci rosse](#)
- [Diagramma di temporizzazione con sovrapposizione di luci rosse](#)

Link ai diagrammi temporali in formato PDF:

- [Diagramma di temporizzazione più semplice senza sovrapposizione di luci rosse](#)
- [Diagramma di temporizzazione con sovrapposizione di luci rosse](#)

Link ai diagrammi temporali in formato JPG:

- [Diagramma di temporizzazione più semplice senza sovrapposizione di luci rosse](#)
- [Diagramma di temporizzazione con sovrapposizione di luci rosse](#)

Configuriamo la scheda Raspberry Pi

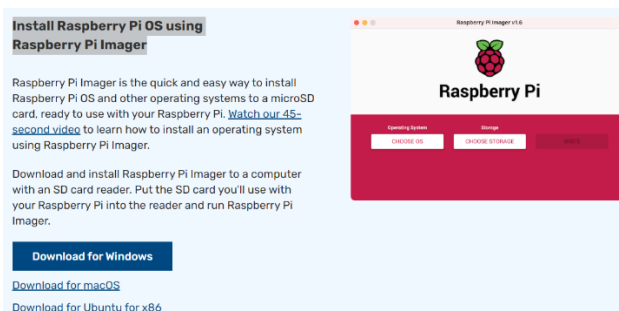
Per far funzionare la scheda Raspberry, è necessario inizialmente trasferire il sistema operativo Raspbian su una scheda SD di almeno 16 GB.

È possibile installare Raspberry Pi OS utilizzando Raspberry Pi Imager, scaricabile dal sito ufficiale di Raspberry al seguente link:

<https://www.raspberrypi.com/software>

e seguendo le istruzioni dettagliate ivi contenute (Figura 3)

Figura 3



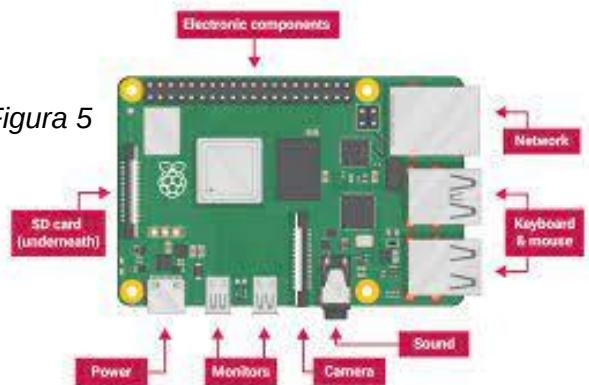
Con lo stesso strumento sarà poi possibile effettuare più copie della scheda SD, una volta configurata completamente, nel caso si vogliano utilizzare più schede.

Una volta installato Raspbian sulla scheda microSD, questa andrà inserita nell'apposito slot della scheda Raspberry (**Figura 4**). La disposizione dei componenti è mostrata nella **Figura 5**.

Figura 4



Figura 5



[Qui](#) troverai un'immagine completa di Raspberry con Scratch 3 preinstallato e i programmi Scratch già integrati (14 GB).

Dopo aver inserito la scheda microSD nell'apposito slot, è possibile accendere la scheda Raspberry e collegarla ad Internet tramite cavo di rete (consigliato) o WiFi. Il sistema operativo si avvierà e potrai installare "Scratch 3" sulla scheda. Il modo più semplice per farlo è utilizzare l'app "Software consigliato":

Apri il menu, fai clic su "Preferenze" > "Software consigliato", quindi seleziona "Scratch 3" e fai clic su OK.

Al seguente link è disponibile il tutorial completo di Scratch per Raspberry Pi:

<https://www.raspberrypi.org/blog/scratch-3-desktop-for-raspbian-on-raspberry-pi>

dove si possono trovare tutte le istruzioni e alcuni esempi di programmazione in modalità desktop.

Costruiamo le basi del nostro programma "Semafori" con scratch

Inizialmente consiglio di creare uno o più programmi in modalità desktop su un PC in modo da prendere confidenza con il sistema.

Il download dell'ambiente Scratch per PC o Mac è disponibile a questo link:

<https://scratch.mit.edu/download>

La prima cosa da fare è creare un nuovo file con Scratch e salvarlo in una cartella specifica dove salveremo tutto lo stato di avanzamento del lavoro.

Una volta creato il nostro file possiamo iniziare importando il disegno precedentemente realizzato su Scratch, che si chiama "Backdrop" e rappresenterà il nostro sfondo su cui posizioneremo i due semafori.

Si può realizzare il Backdrop autonomamente o si può utilizzare questo già fatto:

[Disposizione del semaforo Backdrop.jpg](#)

Come mostrato in **Figure 6 - 7 - 8**

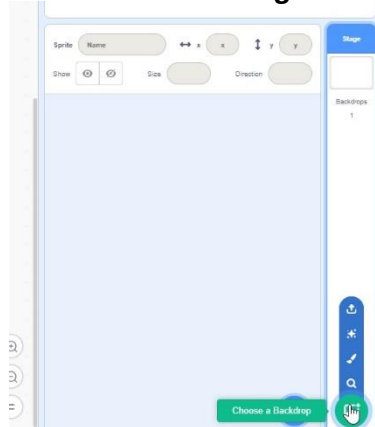


Figura 6

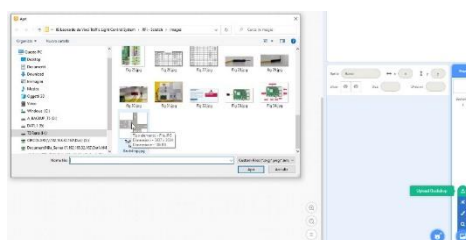


Figura 7



Figura 8

A seguire un video che spiega come importare lo Sfondo:

[Video n. 1: come importare lo sfondo](#)

Dopo aver importato il nostro Backdrop dovremo costruire la forma del semaforo, che in Scratch si chiama "Sprite". Questo sarà duplicato poiché ne avremo bisogno di uno per ogni strada.

Ogni semaforo deve essere rappresentato nelle tre diverse situazioni, quindi abbiamo tre situazioni per ogni Sprite: Rosso, Giallo e Verde. Queste situazioni sono rappresentate da ciò che Scratch chiama "Costumi".

Ora possiamo creare la parte grafica relativa agli Sprite e ai Costumi su Scratch.

Per farlo possiamo utilizzare due metodi:

1. se vogliamo risparmiare tempo nella parte di costruzione grafica per investire più tempo nella parte di programmazione, possiamo utilizzare gli Sprite già creati per questo progetto, come mostrato nelle **Figure 9 - 10 - 11 - 12**.

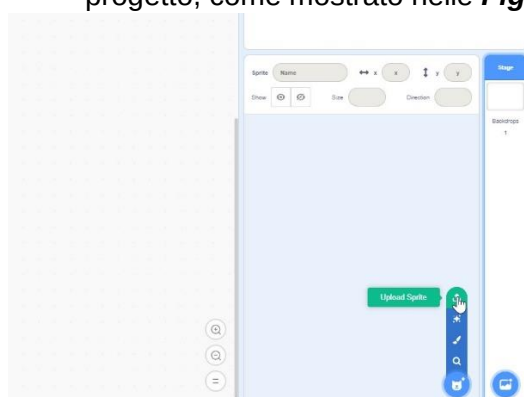


Figura 9

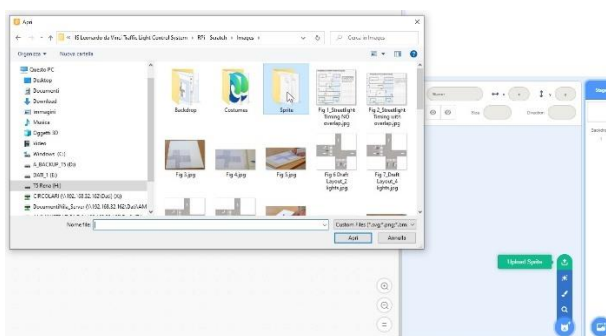


Figura 10

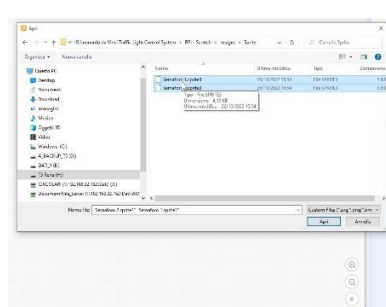


Figura 11



Figura 12

Ecco gli Sprite preassemblati:

[Sprite TL_1](#) e [Sprite TL_2](#)

oppure seguendo il video tutorial qui:

[Video n. 2: come importare sprite e costumi](#)

2. se invece vogliamo dedicare un po' di tempo alla progettazione grafica, possiamo disegnare gli Sprite e Costumi manualmente, all'interno dell'ambiente di sviluppo Scratch. Qui troverai un video tutorial su come realizzarli manualmente:

[Video n. 3: come creare manualmente sprite e costumi](#)

Una volta creata la parte grafica, possiamo iniziare a programmare, in modalità desktop su PC, il nostro semaforo in Scratch.

Costruiamo il nostro semaforo con Scratch in modalità desktop sul PC

Questa fase è utile per chi non ha dimestichezza con la programmazione Scratch, ovvero chi si avvicina a Scratch per la prima volta.

Vari programmi di controllo del semaforo per Scratch in modalità desktop verranno mostrati in ordine crescente di difficoltà:

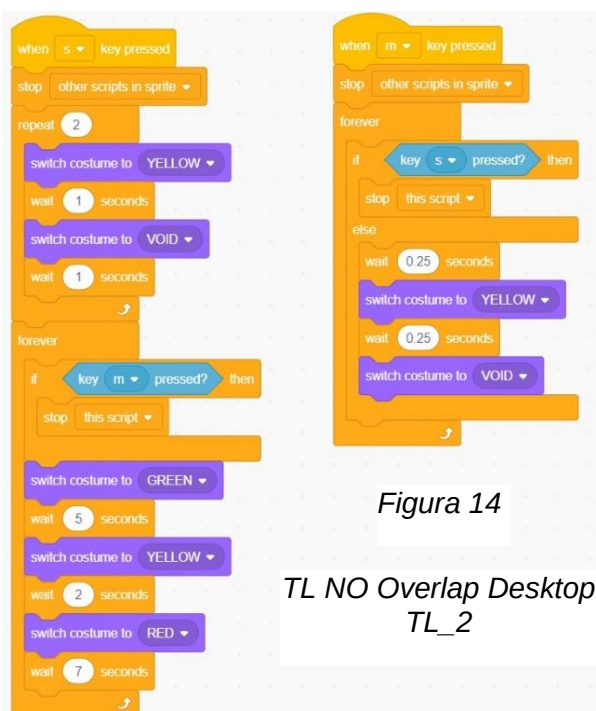
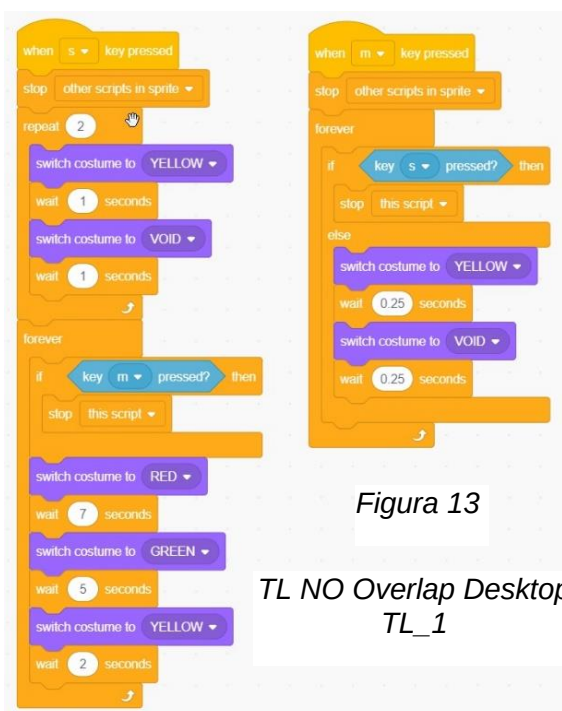
1. TL NO Overlap Desktop
2. TL Overlap Desktop
3. TL Overlap Desktop_Set timing
4. TL Overlap Desktop_Set timing_with Alert

1. TL NO Overlap Desktop

Ecco il primo programma, molto semplice, che controlla i due semafori utilizzando due tasti della tastiera ("s" start e "m" maintenance) per attivare i due script in base al pulsante premuto:

- premendo il tasto "s" si avvia la sequenza (nel nostro esempio 2 lampeggi "solo iniziali" delle luci gialle per 1 secondo poi, continuamente in loop, 7 secondi di rosso, 2 secondi di giallo e 5 secondi di verde). Fai attenzione che il tempo rosso deve essere uguale alla somma del tempo giallo più il tempo verde in ogni Sprite
- premendo il tasto "m" si entra nella modalità "manutenzione" che farà lampeggiare tutte le spie gialle per 0,25 secondi.
- Premendo nuovamente il tasto "s" la sequenza ricomincia dall'inizio e così via.

Per attivare l'una o l'altra funzione è necessario interrompere lo script attualmente in esecuzione. Questo viene fatto usando l'istruzione "ferma gli altri script in sprite", come evidenziato nel programma di esempio mostrato qui:



[Programma: TL NO Overlap Desktop](#)

[Schermata TL 1](#)

[Schermata TL 2](#)

[Video n. 4: TL NO Overlap Desktop](#)

2. TL Overlap Desktop

Questo programma è uguale al precedente ma abbiamo semplicemente inserito per un secondo la sovrapposizione delle due luci rosse.

Ciò si ottiene mantenendo accese le luci rosse in entrambi gli Sprite più a lungo della somma di giallo e verde. Nel nostro esempio abbiamo messo verde=5 secondi, giallo= 2 secondi e rosso=9 secondi, quindi - nell'esempio precedente - 2 secondi più alto del 7. Questo comporta una sovrapposizione di 1 secondo delle luci rosse.

Presta attenzione a queste note:

- *il tempo rosso deve essere uguale alla somma di quelli verde e giallo più il tempo di sovrapposizione*
- *all'inizio della sequenza lo Sprite che partirà con il verde dovrà partire con un ritardo pari al tempo di sovrapposizione (nel nostro caso 1 secondo).*

Qui il codice Scratch e i relativi screenshot:

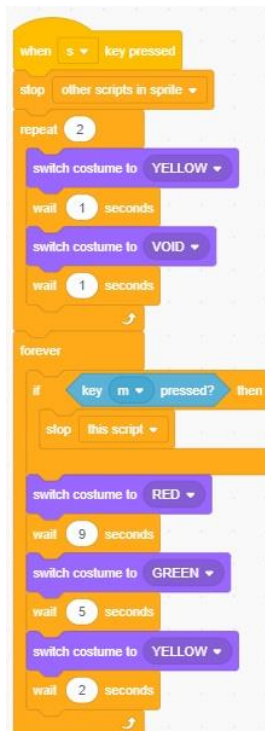


Figura 15
TL Overlap Desktop
TL_1

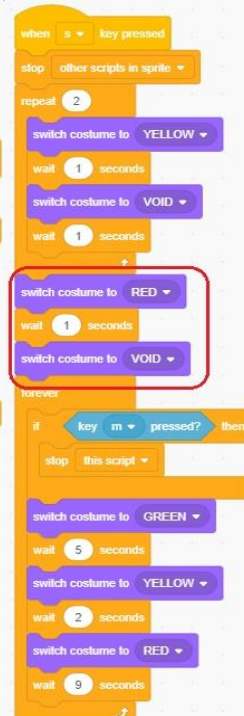


Figura 16
TL Overlap Desktop
TL_2

[Programma: TL CON Sovrapposizione Desktop](#)

[Schermata TL 1](#)

[Schermata TL 2](#)

3. TL Overlap Desktop_Set timing

Introduciamo ora nel programma la possibilità di poter definire, da parte dell'operatore, la durata dei singoli colori. Per farlo è necessario inserire le funzioni "ask" e "answer" che troviamo sotto il gruppo "Sensing".

Dobbiamo creare tante funzioni "ask" quanti sono i colori che vogliamo programmare, cioè tre, ROSSO, GIALLO e VERDE.

Le luci rosse in entrambi gli Sprite devono rimanere "accese" più a lungo della somma di giallo e verde.

Ecco lo Scratch Code, alcuni screenshot con le istruzioni per le nuove funzioni:

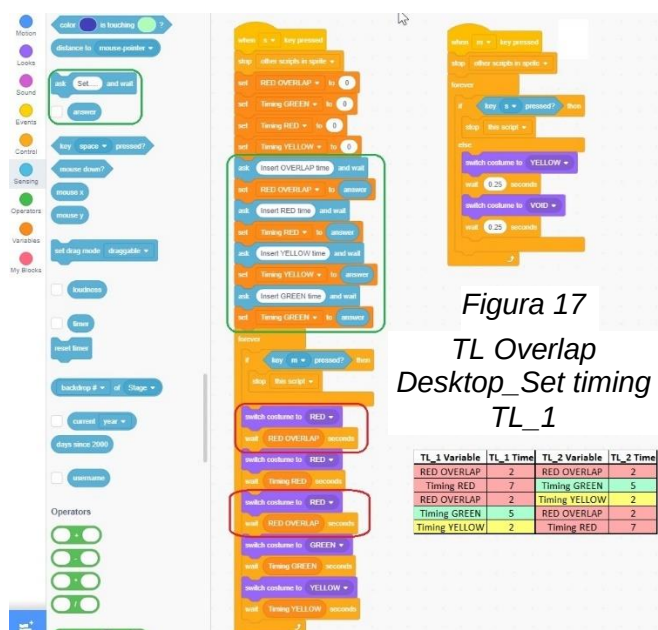


Figura 17
TL Overlap
Desktop_Set timing
TL_1



Figura 18
TL Overlap
Desktop_Set timing
TL_2

Programma: TL Overlap Desktop_Set timing

Schermata TL_1

Schermata TL_2

4. TL Overlap Desktop_Set timing_with Alert

Un altro miglioramento può essere apportato introducendo un "alert" - nel caso in cui la somma dei tempi Verde+Giallo non sia uguale al tempo Rosso. Ciò si ottiene utilizzando la funzione "say" appartenente al gruppo "Looks", come mostrato negli screenshot e nel video seguenti.

Programma: TL Overlap Desktop_Set timing_with Alert

Schermata TL_1

Schermata TL_2

Video n. 5: TL Overlap Desktop_Set timing_with Alert

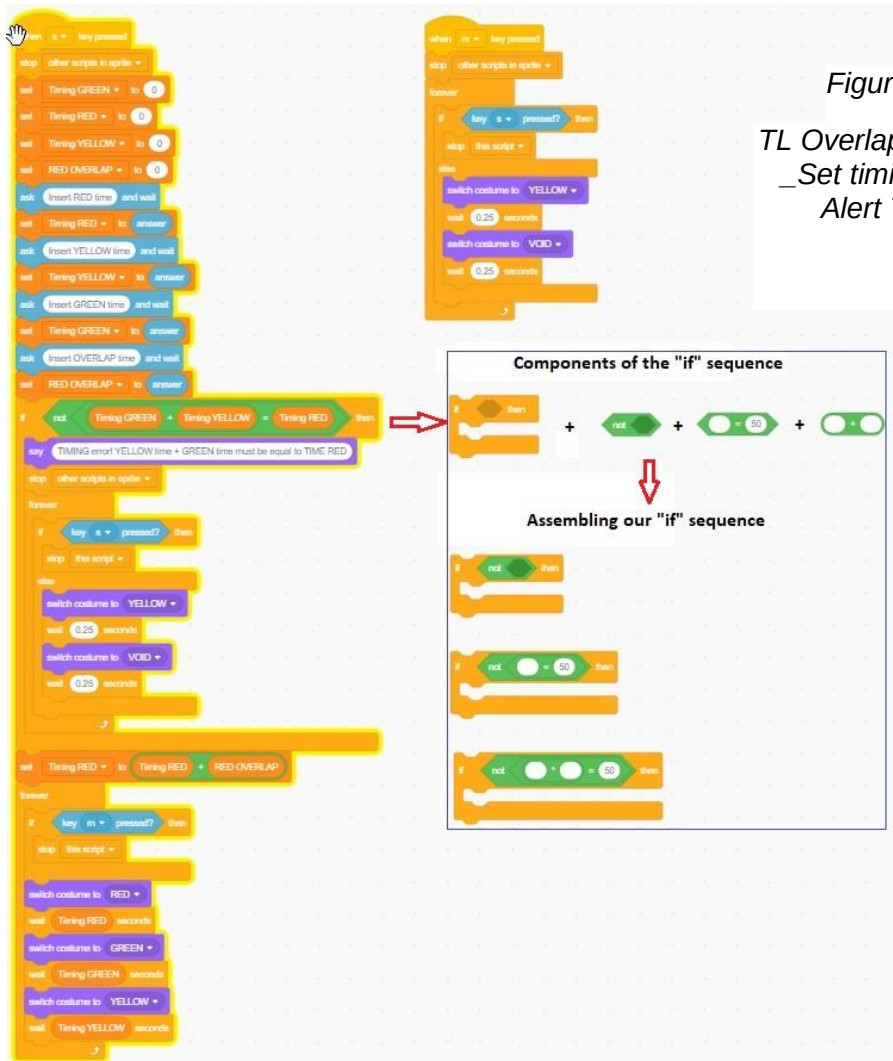
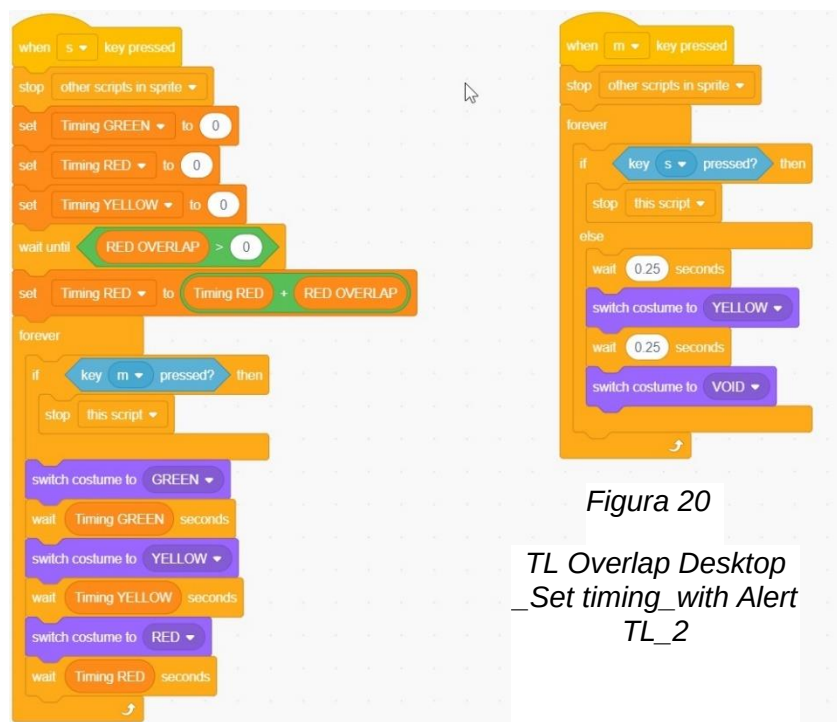


Figura 19

TL Overlap Desktop
_Set timing_with
Alert TL_1



Costruiamo il nostro semaforo con dispositivi esterni e i GPIOs

Adesso andiamo a collegare i GPIO della scheda Raspberry ai dispositivi esterni, nel nostro caso due semafori preassemblati, un pulsante e un interruttore.

Collegamento dei GPIO Raspberry a dispositivi esterni

Il passaggio successivo consiste nel collegare i GPIO Raspberry Pi alle breadboard. Per fare ciò è necessario utilizzare i cavi jumper, inclusi nel Mobile Maker Space. I GPIOs della Raspberry Pi sono mostrati nella seguente **Figura 21**

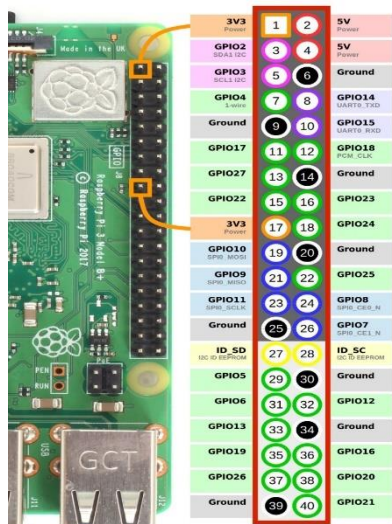


Figura 21 Piedinatura Raspberry Pi

È conveniente realizzare un foglio di calcolo con le interconnessioni tra i GPIO Raspberry Pi e i vari semafori presenti nelle rispettive breadboard. Gli esempi sono in **Figura 23** (due semafori a LED), **Figura 24** (quattro semafori a LED) e **Figura 22** (dispositivi di input). Si noti che nell'esempio "quattro semafori a LED" duplichiamo semplicemente i singoli semafori.

Per il corretto montaggio del pulsante e dell'interruttore è necessario verificare il funzionamento dei pin utilizzando il tester fornito con il Mobile Maker Space. Vedere Note su pulsanti e interruttori a pag. 24.

Raspberry Pi		Input devices
GPIO	PIN	Type
GPIO17	11	Start Push-button
GPIO27	13	Maintenance switch
3V3	1	+Vcc

Figura 22
Pinout dei
dispositivi di
input

Raspberry Pi		Led Traffic Lights		
GPIO	PIN	Traffic Light n.	Colour	Pin
GPIO10	19	1	Green 1	G
GPIO9	21	1	Yellow 1	Y
GPIO11	23	1	Red 1	R
Ground	25	1	---	GND
GPIO2	3	2	Green 2	G
GPIO3	5	2	Yellow 2	Y
GPIO4	7	2	Red 2	R
Ground	9	2	---	GND

Figura 23 Tabella di collegamento con soltanto 2
semafori led

Raspberry Pi		Led Traffic Lights		
GPIO	PIN	Traffic Light n.	Colour	Pin
GPIO10	19	1-1	Green 1-1	G
GPIO9	21	1-1	Yellow 1-1	Y
GPIO11	23	1-1	Red 1-1	R
Ground	25	---	---	GND
Ground	20	1-2	---	GND
GPIO25	22	1-2	Red 1-1	R
GPIO8	24	1-2	Yellow 1-1	Y
GPIO7	26	1-2	Green 1-1	G
GPIO2	3	2-1	Green 2-1	G
GPIO3	5	2-1	Yellow 2-1	Y
GPIO4	7	2-1	Red 2-1	R
Ground	9	2-1	---	GND
Ground	6	2-2	---	GND
GPIO14	8	2-2	Red 2-2	R
GPIO15	10	2-2	Yellow 2-2	Y
GPIO18	12	2-2	Green 2-2	G

Figura 24 Tabella di collegamento con 4
semafori led

Cliccando sui link sottostanti è possibile scaricare la Tabella di Collegamento in vari formati:

Link alla Tabella dei collegamenti in formato MS Excel:

- [Tabella di connessione Semaforo x 2.xlsx](#)
- [Tabella di connessione Semaforo x 4 - 2 x strada.xlsx](#)

Link alla Tabella dei collegamenti in formato PDF:

- [Tabella di connessione Semaforo x 2.pdf](#)
- [Tabella di connessione Semaforo x 4 - 2 x strada.pdf](#)

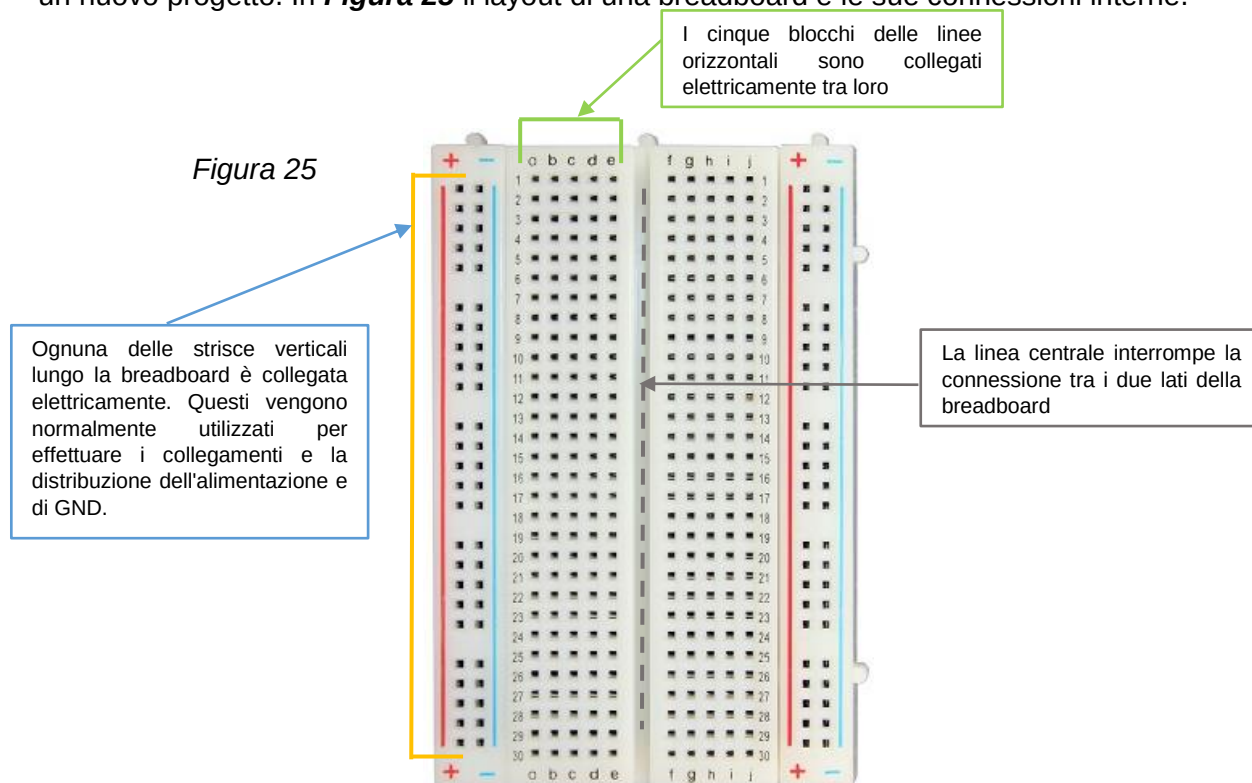
Link alla Tabella dei collegamenti in formato JPG:

- [Tabella di connessione Semaforo x 2.jpg](#)
- [Tabella di connessione Semaforo x 4 - 2 x strada.jpg](#)

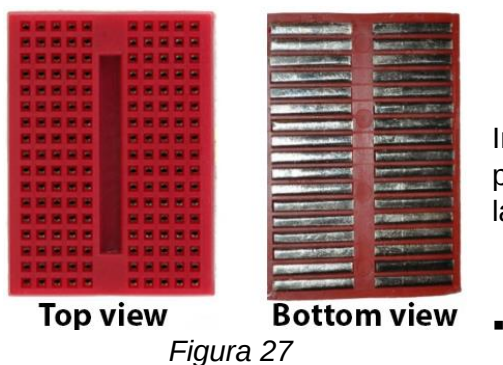
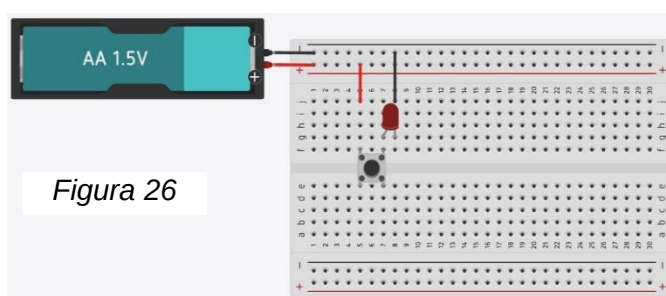
La breadboard

Breadboard è un modo per realizzare dispositivi elettronici senza dover utilizzare un saldatore. I componenti vengono inseriti nelle prese (fori) presenti sulla breadboard e quindi vengono collegati utilizzando fili "jumper" esterni per effettuare le connessioni tra i vari componenti.

Le breadboard sono fatte per effettuare prototipi o esperimenti veloci. Non sono adatte per mantenere in funzione i circuiti per molto tempo. Le connessioni, non essendo permanenti, sono facilmente removibili nel caso si commetta un errore o semplicemente iniziare da capo un nuovo progetto. In **Figura 25** il layout di una breadboard e le sue connessioni interne.



In **Figura 26** un esempio di come collegare un LED con una batteria e un pulsante



In **Figura 27** un altro tipo di breadboard, più semplice e più piccolo, che non ha le strisce verticali per i collegamenti e la distribuzione dell'alimentazione e GND

Il sistema può essere implementato in due modi diversi:

- il più veloce è costruire un prototipo con un'unica breadboard su cui assemblare tutti i componenti. Questo sistema non prevede la realizzazione dei disegni di tracciato stradale e la realizzazione della piattaforma su cui posizionare i vari componenti (**Figura 28**)
- realizzazione dei disegni di tracciato stradale e realizzazione della piattaforma su cui posizionare i vari componenti. Questo sistema richiede più tempo perché comporta anche una parte di realizzazione grafica e costruzione del pannello su cui inserire i vari elementi **Figure 29 - 30**



Figura 28

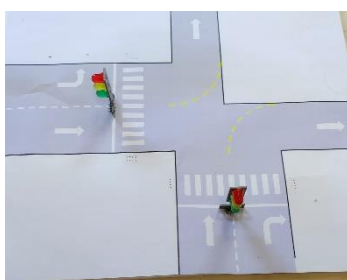


Figura 29

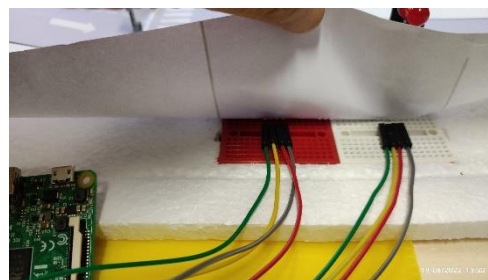


Figura 30

A questo punto è necessario effettuare tutti i collegamenti tra la scheda Raspberry Pi e la/e breadboard/i seguendo il pinout indicato nelle **Figure 21 - 22 - 23**.

Una volta completate le connessioni possiamo iniziare a programmare la Raspberry Pi con i GPIO.

Si noti che per lavorare con i GPIO è necessario utilizzare la scheda Raspberry Pi, in quanto i moduli GPIO non possono essere caricati su un PC.

Di seguito vengono mostrati due esempi:

1. TL NO Overlap GPIO
2. TL Overlap GPIO

1. TL NO Overlap GPIO

Questo programma è simile a quello già fatto nella versione Desktop (TL NO Overlap Desktop) con la differenza che al posto dei "Costumi" dovremo programmare direttamente i vari GPIO che comandano i vari LED dei due semafori.

Prestare attenzione alle seguenti indicazioni:

- all'inizio di ogni sequenza è importante impostare tutte le uscite GPIO a "0", in modo che tutti i LED siano spenti all'avvio del programma
- ogni volta che imposti un GPIO a "1" e programmi il tempo in cui quel led rimarrà acceso, dovrai anche spegnerlo riportandolo a "0" prima di accendere il successivo e viceversa.

Le **Figure 31 e 32** mostrano il programma "TL NO Overlap GPIO" per i due Sprite TL_1 e TL_2 utilizzando i GPIO indicati nelle **Figure 21 - 22 - 23**

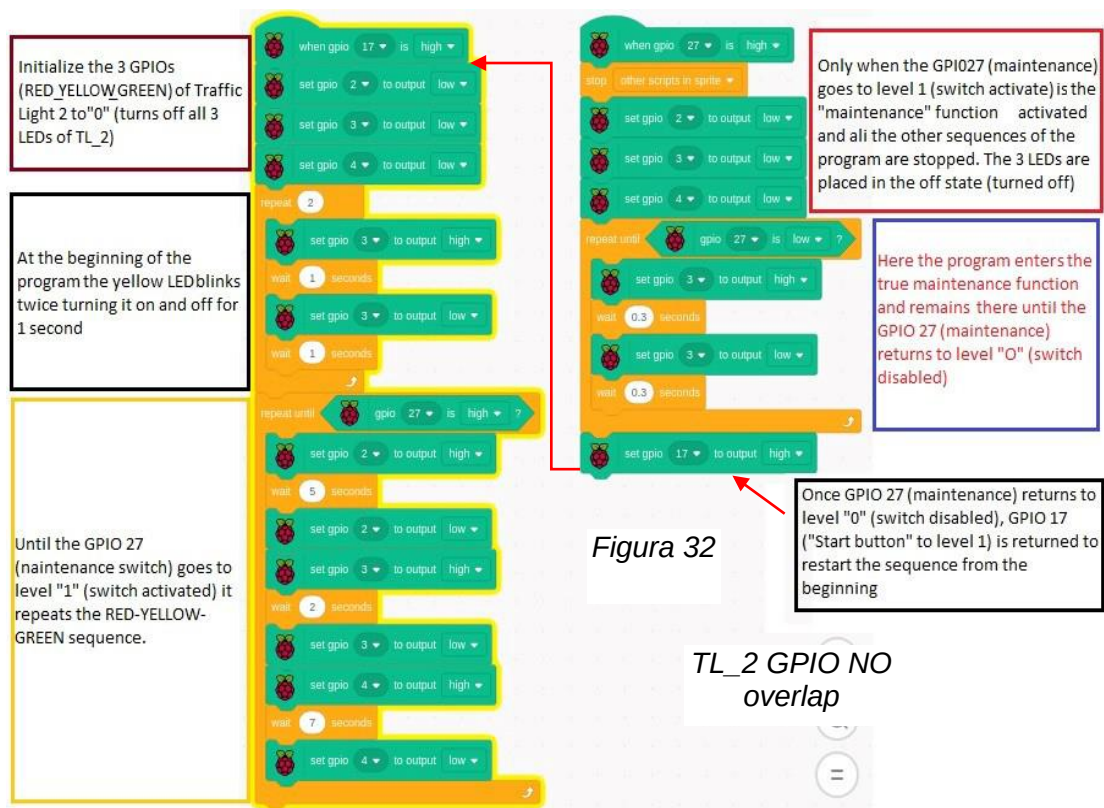
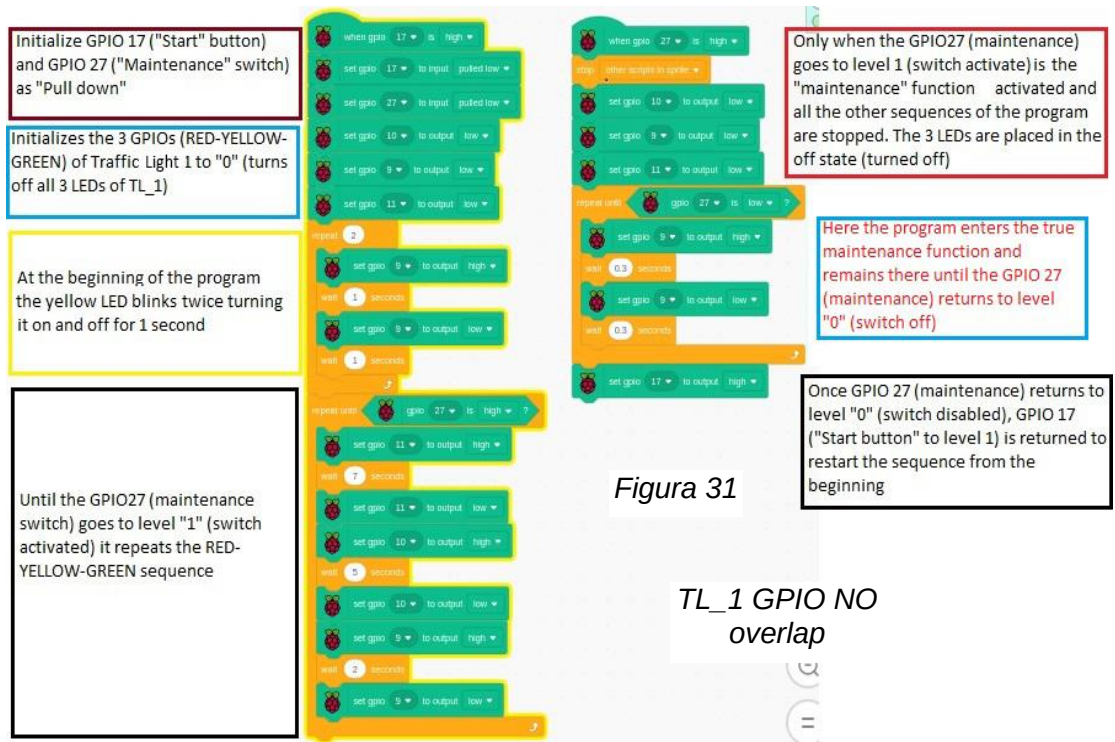
Qui puoi scaricare lo Scratch Code, alcuni screenshot e un video:

[Programma: TL NO Overlap GPIO](#)

[Schermata TL 1](#)

Schermata TL 2

Video n. 6: TL NO Overlap GPIO



2. TL Overlap GPIO

Il programma è molto simile a quello già realizzato (TL SENZA Sovrapposizione GPIO). L'unica differenza è che i tempi delle luci dovranno tenere conto della sovrapposizione delle luci rosse. Nel nostro caso abbiamo impostato:

ROSSO = 7 secondi

GIALLO = 1 secondo

VERDE = 4 secondi

Come si vede la somma VERDE (4 sec.) + GIALLO (1 sec.) è impostata a 5 secondi ma la luce ROSSA rimarrà "accesa" per 7 secondi, quindi 2 secondi in più rispetto alla somma VERDE + GIALLO. Ciò fa sì che le due luci ROSSE si sovrappongano per 2 secondi prima che si attivi la successiva luce VERDE.

Notare che è stata aggiunta un'istruzione "If" ("se") in TL_2 che mantiene accesa la luce rossa di TL_2 per 1 secondo in modo da creare la prima sovrapposizione di luci rosse all'inizio della sequenza, che funziona come segue:

- Se ("if") GPIO 27 (Interruttore di manutenzione) è a livello "0" (non è attivo)
- Allora ("then")
- Impostare ("set") GPIO 4 (ROSSO) a "1" (attivalo)
- Aspetta ("wait") 1 secondo (for 1 sec.)
- impostare ("set") GPIO 4(ROSSO) a "0" (spegnilo)
- aspetta 1 secondo (for 1 sec.)

e quindi il programma passa alle istruzioni successive.

Come indicato anche nell'esempio precedente occorre prestare attenzione alle seguenti indicazioni:

- all'inizio di ogni sequenza è importante impostare tutte le uscite GPIO a "0", in modo che tutti i LED siano spenti all'avvio del programma
- ogni volta che imposti un GPIO a "1" e programmi il tempo in cui quel led rimarrà acceso, dovrai anche spegnerlo riportandolo a "0" prima di accendere il successivo e viceversa.

Le **Figure 33 e 34** mostrano il programma "TL_Overlap GPIO" per i due Sprite TL_1 e TL_2 utilizzando i GPIO indicati nelle **Figure 21 - 22 - 23**.

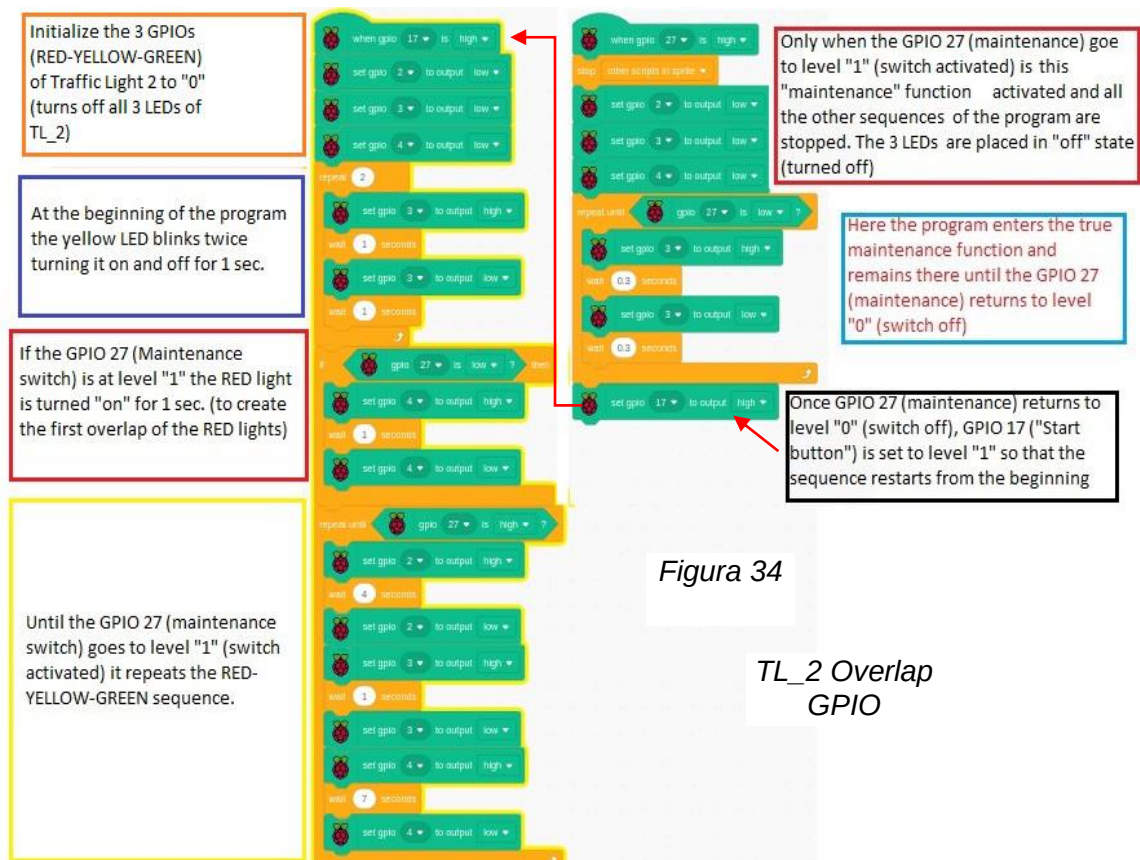
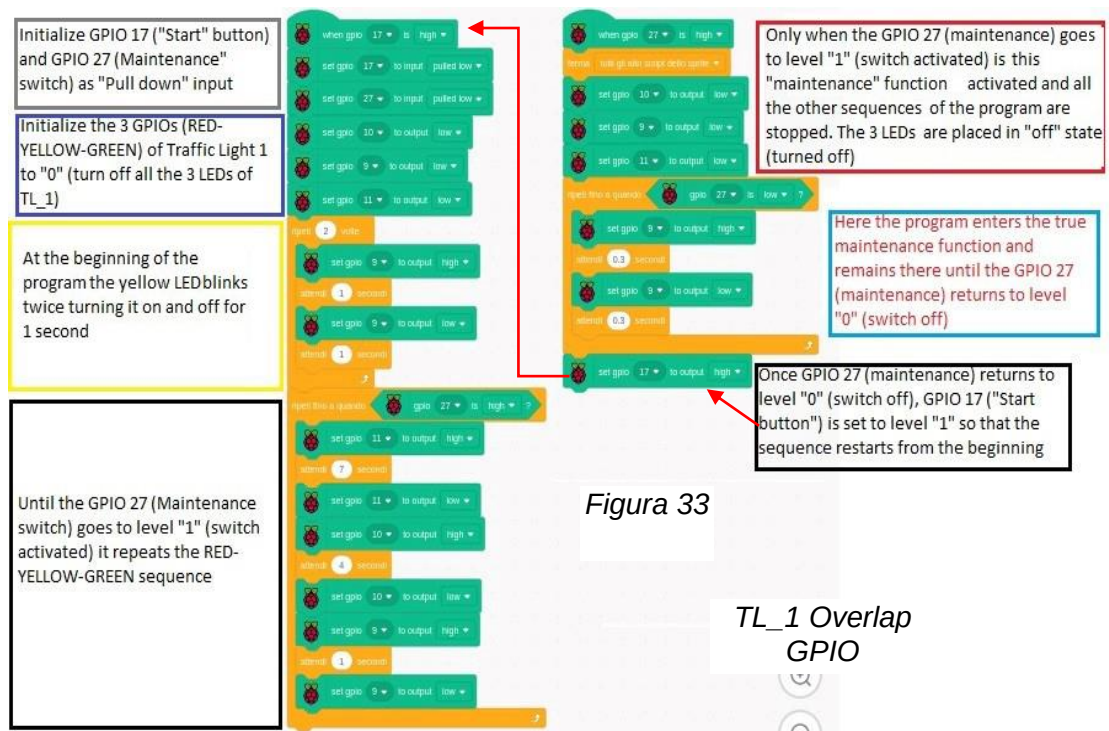
Di seguito i link per scaricare lo Scratch Code, alcuni screenshot e un video:

[Programma: TL_Overlap GPIO](#)

[Schermata TL_1](#)

[Schermata TL_2](#)

[Video n. 7: TL_Overlap GPIO](#)



Terminata la programmazione si può passare alla fase di test, utilizzando il prototipo con un'unica breadboard (**Figura 28**) oppure realizzare i disegni del tracciato stradale e la costruzione della piattaforma (**Figure 29 - 30**) come indicato di seguito.

Disegno e realizzazione della planimetria del raccordo a due vie

La planimetria dell'incrocio semaforico può essere realizzata con un foglio di carta delle dimensioni, come nel nostro caso, 28x34 cm., oppure utilizzando un semplice foglio A4, come indicato nelle **Figure 35 - 36 - 37**.

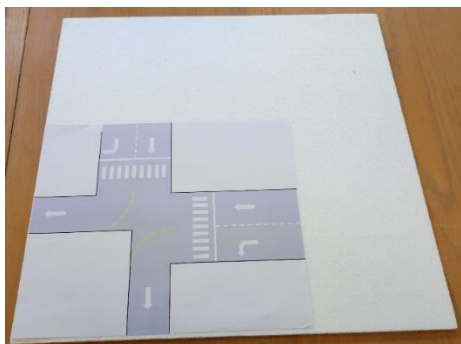


Figura 35

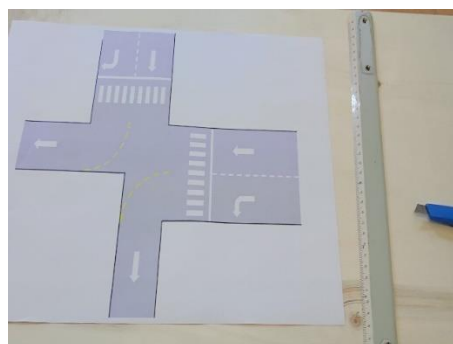


Figura 36



Figura 37

Il progetto può essere realizzato a mano con pennarelli o matite colorate oppure utilizzando un programma di disegno e stampandolo; nel nostro caso abbiamo utilizzato MS Word. Di seguito sono mostrati tre diversi layout:

- “Draft Layout_2 luci” **Figura 38**
- “Draft Layout_4 luci” **Figura 39**
- “Layout a semaforo per il montaggio” **Figura 40**

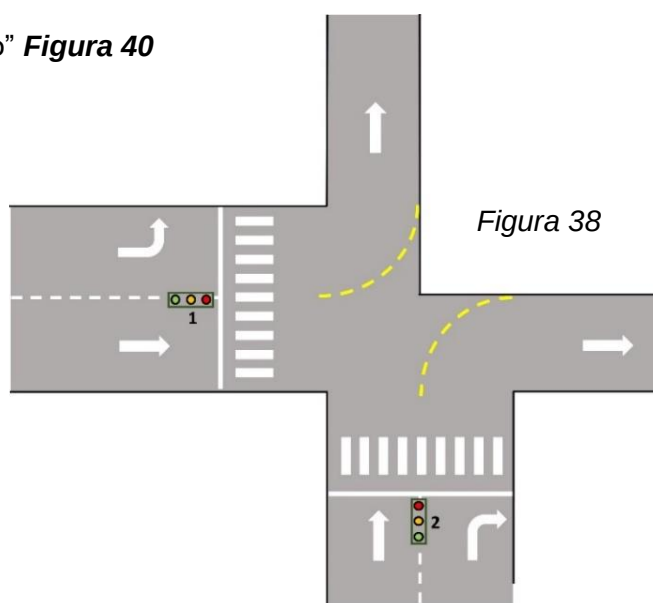


Figura 38

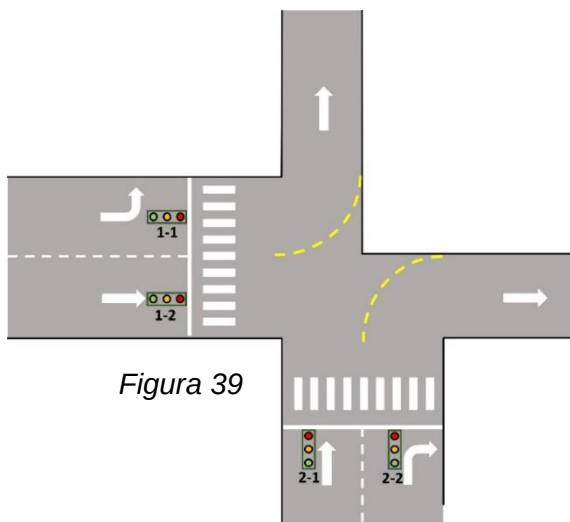


Figura 39

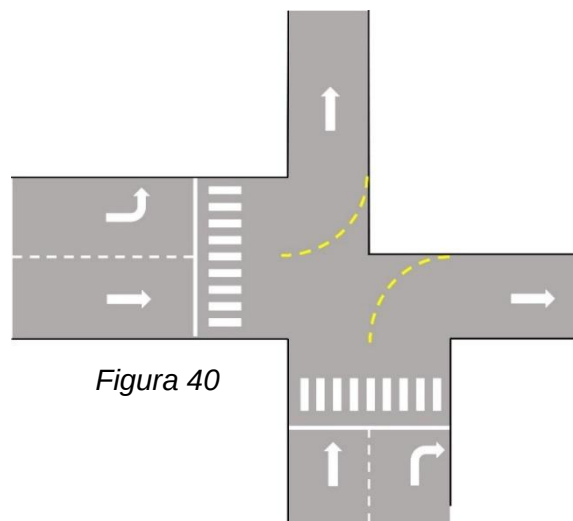


Figura 40

Cliccando sui link sottostanti è possibile scaricare il Layout in vari formati:

Link al Layout realizzato con MS Word:

- [Bozza Layout 2 luci](#)
- [Bozza Layout 4 luci](#)
- [Disposizione del semaforo per il montaggio](#)

Link al Layout realizzato in formato PDF:

- [Bozza Layout 2 luci](#)
- [Bozza Layout 4 luci](#)
- [Disposizione del semaforo per il montaggio](#)

Link al Layout realizzato in formato JPG:

- [Bozza Layout 2 luci](#)
- [Bozza Layout 4 luci](#)
- [Disposizione del semaforo per il montaggio](#)

Dopo aver creato il layout cartaceo sarà necessario realizzare una base in polistirolo alta 1 cm dove andranno inserite le breadboard. Il foglio di polistirolo avrà le dimensioni del foglio di carta A4 usato per creare il layout grafico o leggermente più grande (**Figura 41 – 42**)



Figura 41

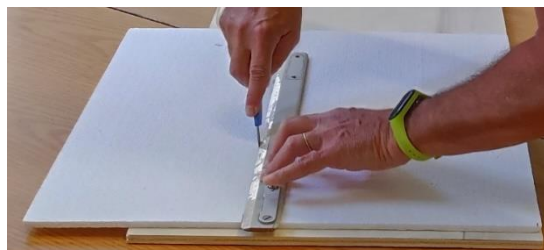


Figura 42

Il layout e i fogli di polistirolo tagliati a misura sono mostrati in **Figura 43**

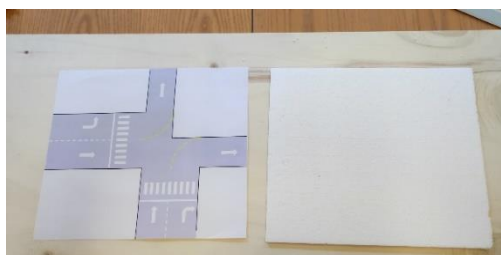


Figura 43

Il prossimo passo sarà segnare la posizione esatta delle breadboard su un disegno dell'intersezione utilizzando un layout creato in precedenza. **Figura 44 - 45**

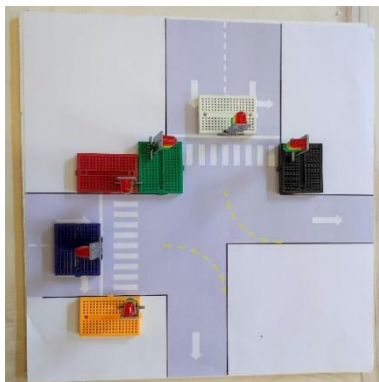


Figura 44

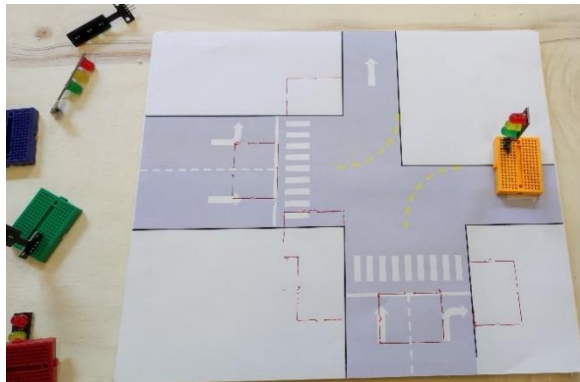


Figura 45

Una volta tracciati sul disegno i contorni esatti delle breadboard, dovremo ritagliare i contorni utilizzando un taglierino **Figura 46**

Fatto ciò, dobbiamo tracciare i contorni sulla lastra di polistirolo usando il disegno che abbiamo appena modellato come riferimento e guida **Figura 47**

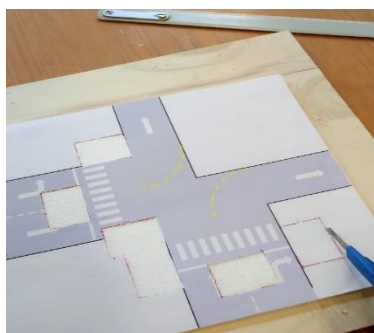


Figura 46

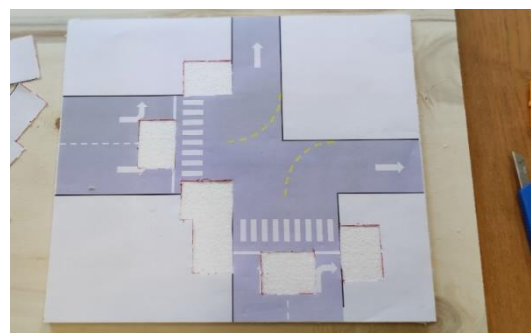


Figura 47

Ora possiamo ritagliare le fessure della breadboard nel foglio di polistirolo **Figura 48 - 49 - 50**

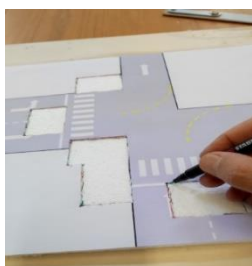


Figura 48

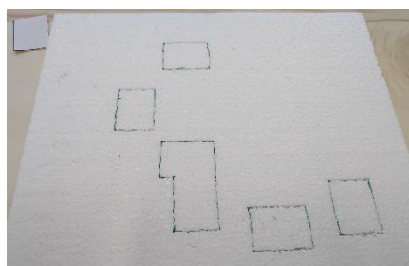


Figura 49

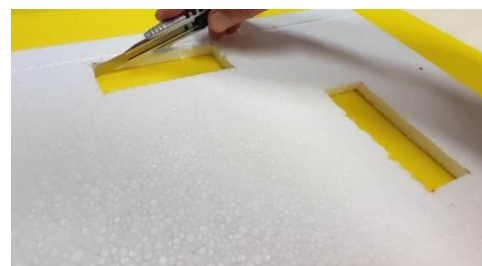


Figura 50

Sarà necessario prestare un po' di attenzione nel sagomare perfettamente le fessure nella lastra di polistirolo, in modo che le breadboard vi si incastrino perfettamente.

Una volta ritagliate le fessure nei fogli di polistirolo, si dovranno inserire le breadboard, ognuna nel proprio alloggiamento, come mostrato in **Figura 51**.

Il risultato di questo passaggio intermedio è mostrato in **Figura 52**.

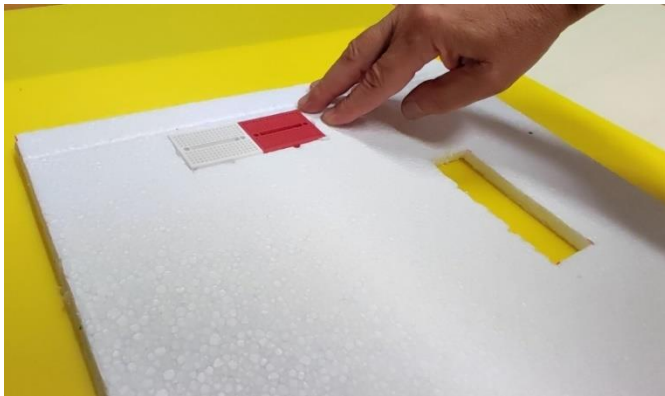


Figura 51

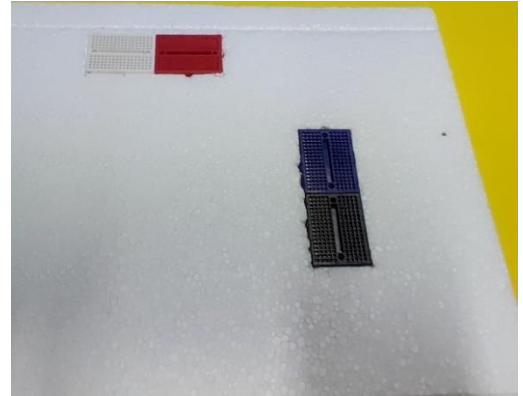


Figura 52

Possiamo ora appoggiare il layout cartaceo precedentemente realizzato sulla base di polistirolo e procedere all'inserimento dei mini semafori a led sulle relative breadboard, seguendo i passaggi mostrati nelle **Figure 53 - 54**

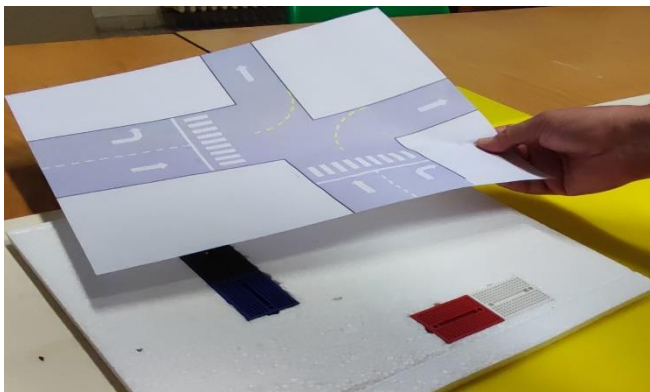


Figura 53

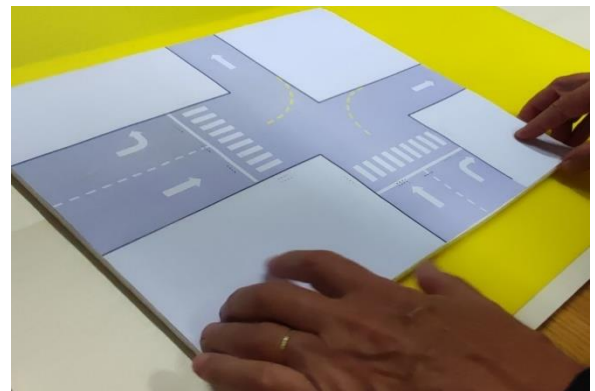


Figura 54

Sotto il layout del semaforo completo con i due semafori a LED inseriti nella breadboard **Figura 55**

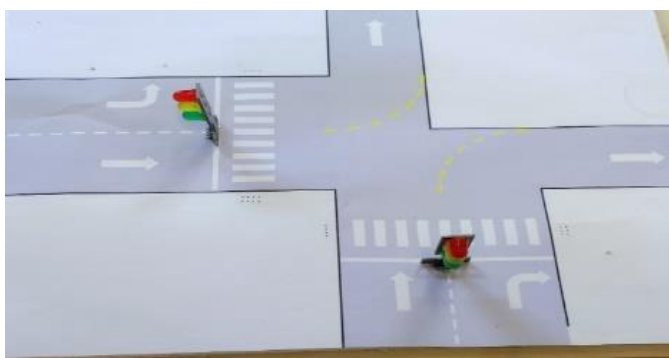


Figura 55

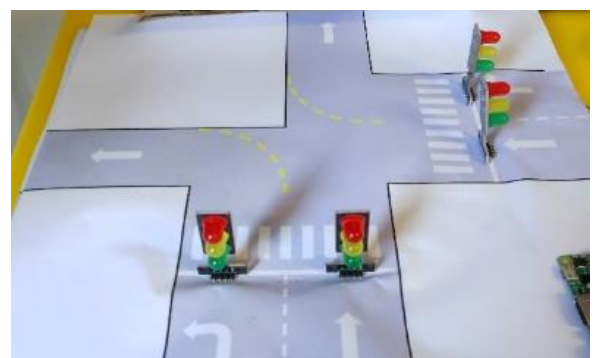


Figura 56

La **Figura 56** mostra un altro esempio dello stesso incrocio ma con due semafori per strada.

Ora possiamo collegare i dispositivi LED ai GPIOs della Raspberry Pi.

È necessario piegare di 90° i piedini dei ponticelli da inserire nella breadboard, in modo da poterli ricoprire con il disegno planimetrico. Vedi **Figure 57 - 58 - 59**



Figura 57



Figura 58

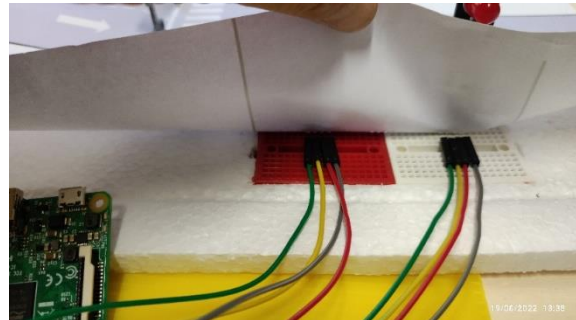


Figura 59

Alla fine otteniamo il sistema completo come mostrato in **Figura 60**

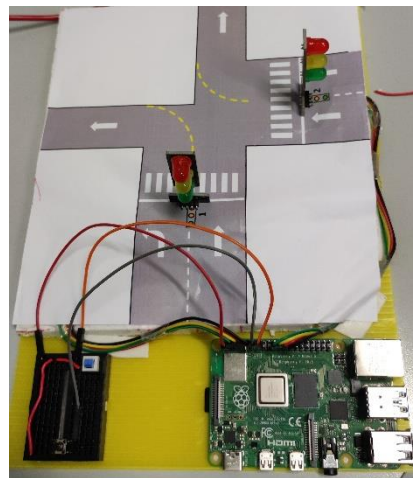


Figura 60

Note sui collegamenti di pulsanti e interruttori

Il PinOut del pulsante utilizzato nel prototipo è mostrato in **Figura 61**.



Figura 61

Si noti che i pin 1-3 (blu nella figura) sono collegati tra loro, così come i pin 2-4 (rossi nella figura). La connessione tra i pin 1-2 è NORMALMENTE APERTA e si chiuderà solo alla pressione del pulsante. Lo stesso vale per i pin 3-4.

Il PinOut dell'interruttore utilizzato nel prototipo è mostrato in **Figura 62**:

Si usino le barre verticali su entrambi i lati come riferimento (su un lato ha solo una barra, l'altro lato ha due barre)

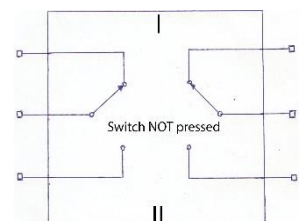


Figura 62

Note sulle resistenze di pull-up e pull-down

I resistori di pull-up (o pull-down) sono resistori a valore fisso, tra l'alimentazione (o massa) e un particolare pin in un circuito logico digitale. Comunemente abbinato agli interruttori, il suo scopo è garantire che la tensione tra massa (GND) e Vcc sia controllata attivamente quando l'interruttore è aperto. Senza il resistore di pull-up lo stato logico dell'interruttore aperto è flottante, quindi potrebbe essere uno '0' o un '1', perché non essendo connesso a un potenziale stabile, si comporta come un'antenna e la tensione ai suoi capi è fluttuante, indeterminata.

Di seguito alcuni esempi di come collegare la resistenza di pull-up (o pull-down).

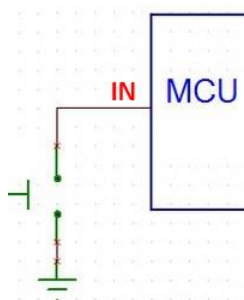


Figure 563
Senza
resistore

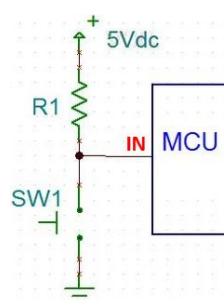


Figure 64
Resistore Pull-Up
Switch non premuto

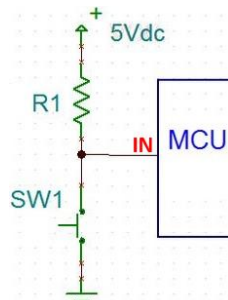


Figure 65
Resistore Pull-Up
Switch premuto

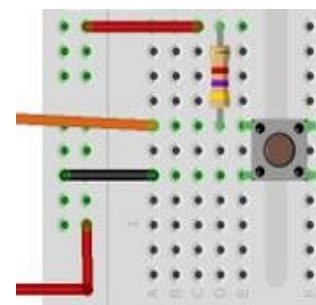


Figure 66
Resistore Pull-Up
sulla breadboard

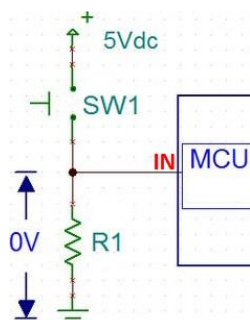


Figure 67
Resistore Pull-Down
Switch non premuto

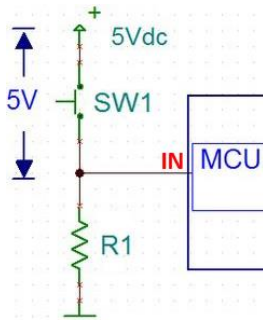


Figure 68
Resistore Pull-Down
Switch premuto

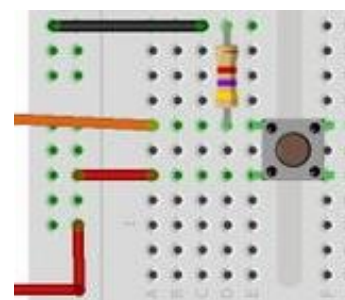


Figure 69
Pull-Down resistor
on the breadboard

Quale dovrebbe essere il valore della resistenza?

Supponiamo di voler limitare la corrente a 1mA. Ad esempio se $V_{cc} = 5V$, utilizzando la legge di Ohm: $R = V/I$ otteniamo $R = 5V/1mA$ (oppure $R = 5000mV/1mA$) quindi $R = 5000\Omega = 5k$. In questo caso possiamo utilizzare una resistenza da 4.7k Ω . Normalmente è possibile utilizzare, in generale, un resistore da 10k Ω , questo richiederà solo 0,5mA

Normalmente in tutti i sistemi a microcontrollore le resistenze di pull-up e pull-down sono già integrate all'interno della scheda stessa. I pin di ingresso possono essere programmati per essere configurati come ingressi pull-up o pull-down con semplici istruzioni.

Si veda la guida di Raspberry Pi per ulteriori informazioni.

Esempio di Pull-Down



La freccia a destra permette di scegliere fra pull-down (pulled low) e pull-up (pulled high).

File di supporto

Foglio di lavoro per gli studenti

[Il foglio di lavoro per gli studenti](#)

Programmi

[La cartella con tutti i programmi](#)

Figure

[La cartella con tutte le figure](#)

Video

[La cartella con tutti i video](#)

Alcuni Video di YouTube con sottotitoli

Titolo	Collegamento YouTube	Sottotitoli
Video n. 1 Come importare lo sfondo	https://youtu.be/50UOzYD03b8	Italiano - Polacco - Greco - Inglese
Video n. 2 Come importare sprite e costumi	https://youtu.be/oZUyCjgZfFc	Italiano - Polacco - Greco - Inglese
Video n. 3 Come realizzare manualmente Sprite e Costumi	https://youtu.be/lb0bh-uLU6Y	Italiano - Polacco - Greco - Inglese
Video n. 6 TL SENZA Sovrapposizione GPIO	https://youtu.be/JE-h8yCibG0	Italiano - Polacco - Greco - Inglese
Video n. 7 TL CON Sovrapposizione GPIO	https://youtu.be/TXQW1D1XOGI	Italiano - Polacco - Greco - Inglese

Guide Raspberry Pi e Scratch

Documentazione Raspberry Pi:

<https://www.raspberrypi.com/documentation>

Iniziare con Raspberry Pi:

<https://projects.raspberrypi.org/en/projects/raspberry-pi-per-iniziare>

Ambiente Scratch per Raspberry Pi:

<https://scratch.mit.edu/download>