

# Embedded systems for smart cities: intelligent traffic lights

*Progetto per il corso di Computer Systems Design - Prof. Nicola Mazzocca*

M63001494 Abdel Majid Zaira

M63001490 Cipollaro Daiana

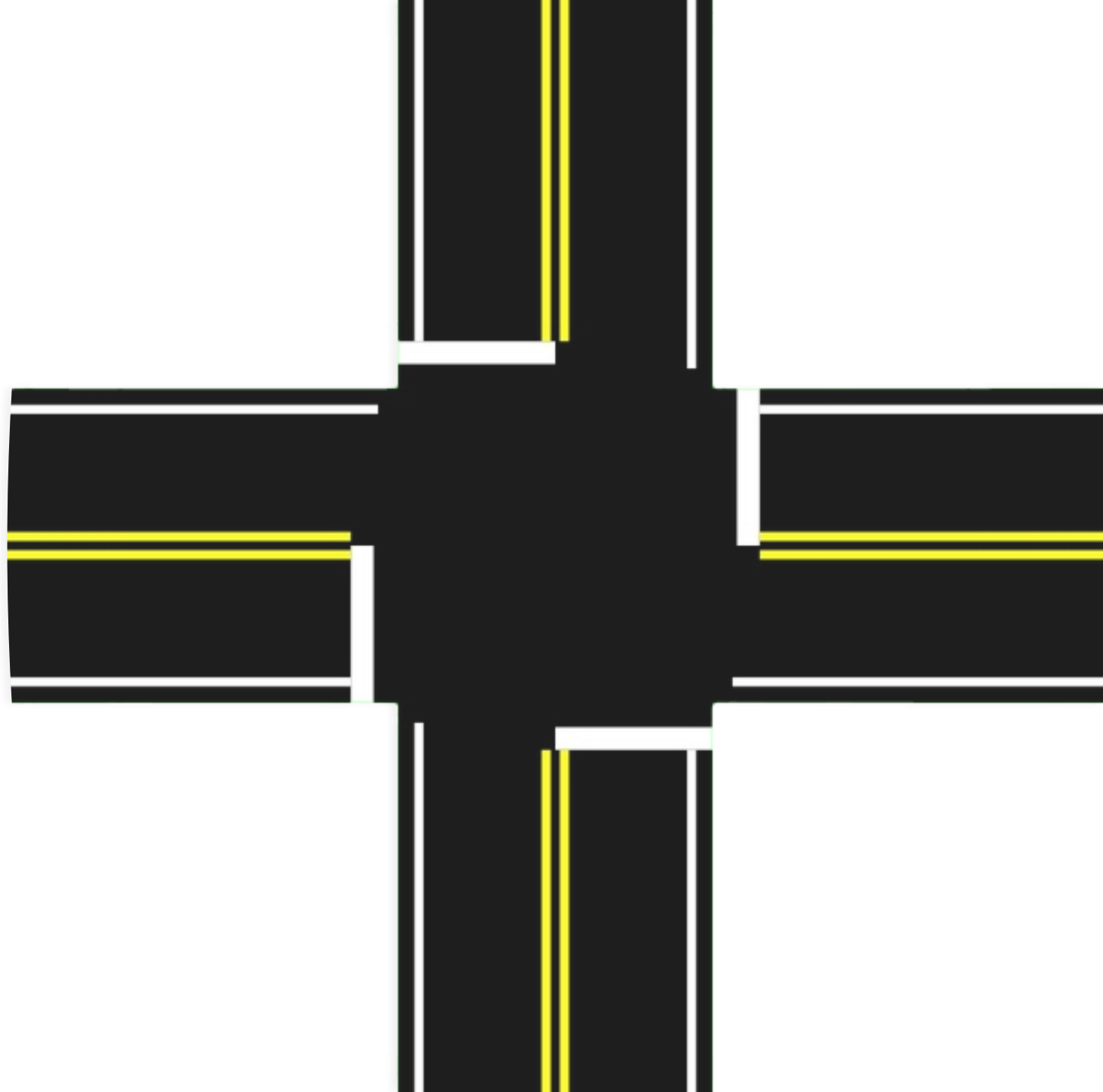
M63001442 Di Serio Francesco

N000122476 Manco Lorenzo



# Schematizzazione del problema

- Si vuole progettare un sistema costituito da due semafori e un giudice per regolamentare il traffico in un incrocio come quello schematizzato in figura
- Si assume che la strada a nord e quella ad est siano a senso unico e sia sempre consentito alle macchine, supposto che esse abbiano il verde, di svoltarvici
- Alla pressione di un bottone, sarà consentito l'attraversamento pedonale e bloccato il flusso di traffico in tutte le direzioni che entrano in conflitto e mettono in pericolo l'attraversamento dei pedoni

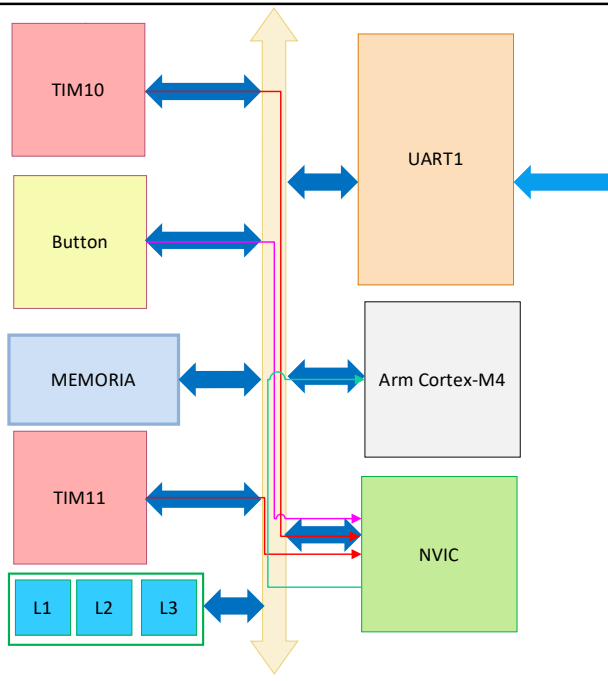


# Architettura complessiva

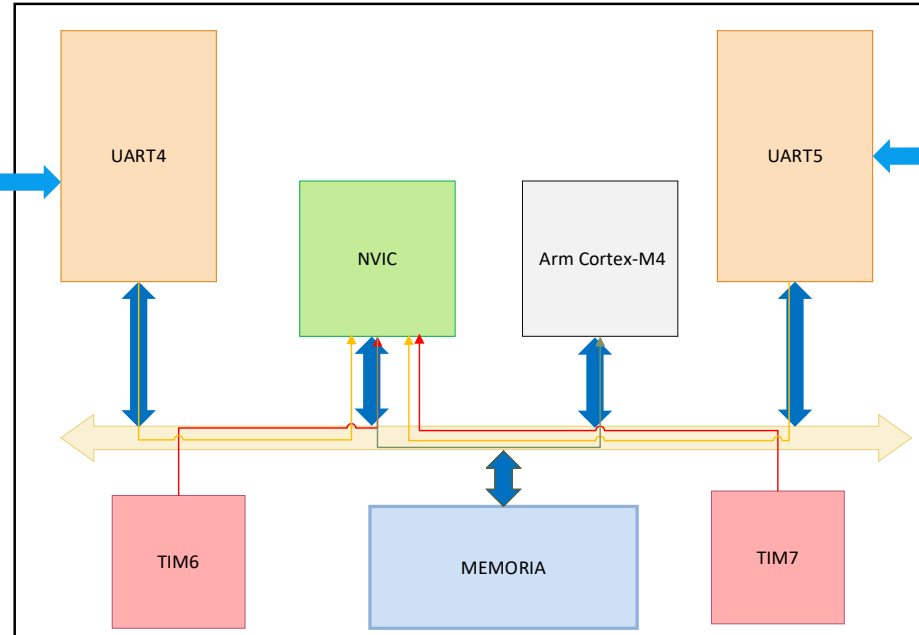
Abbiamo realizzato un sistema **multi-computer** costituito da tre nodi:

- uno schedino (*STM32F3DISCOVERY*) che ricopre il ruolo di **giudice**: si occupa di comunicare ai semafori quali luci accendere;
- due schedini (*NUCELO-F401RE*) che rappresentano i **semafori**: inviano le richieste di prenotazione da parte dei pedoni al giudice e accendono i led indicati.

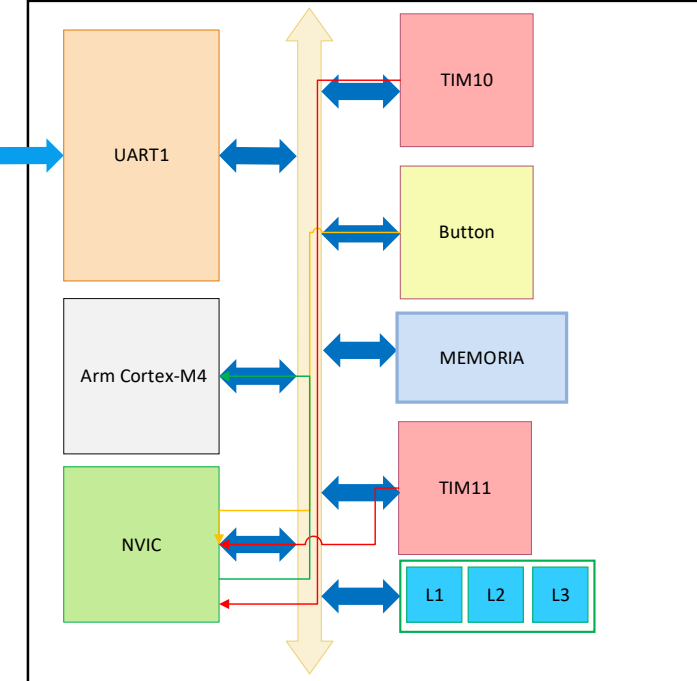
SEMAFORO



GIUDICE



SEMAFORO



# Convenzione

- Con 'Semaforo1' intendiamo **il complesso di semafori** della strada in basso
- Con 'Semaforo2' intendiamo **il complesso di semafori** della strada a sinistra
- Con 'SemDritto' intendiamo **il semaforo specifico** che regola il traffico per andare dritto
- Con 'SemSvolta' intendiamo **il semaforo specifico** che regola il traffico per svoltare nella strada adiacente
- Con 'R1' intendiamo **la richiesta di attraversamento** effettuata da un pedone a semaforo1
- Con 'R2' intendiamo **la richiesta di attraversamento** effettuata da un pedone a semaforo2

# **Automi semafori**



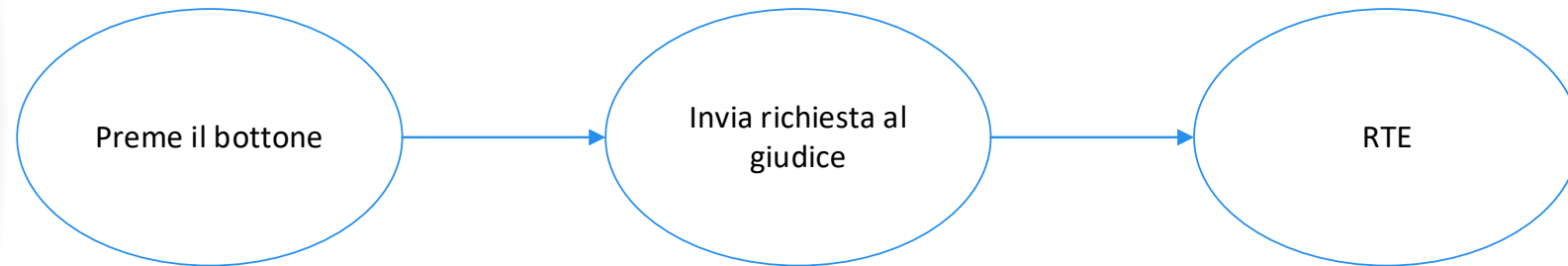
# Main

- Il main sarà dato esclusivamente dall'inizializzazione delle periferiche e dall'avvio del timer per il watchdog. Dopodiché, il sistema compirà ulteriori azioni solo se interrotto da eventi esterni e per eseguire le relative ISR.

# Automa ISR pressione bottone

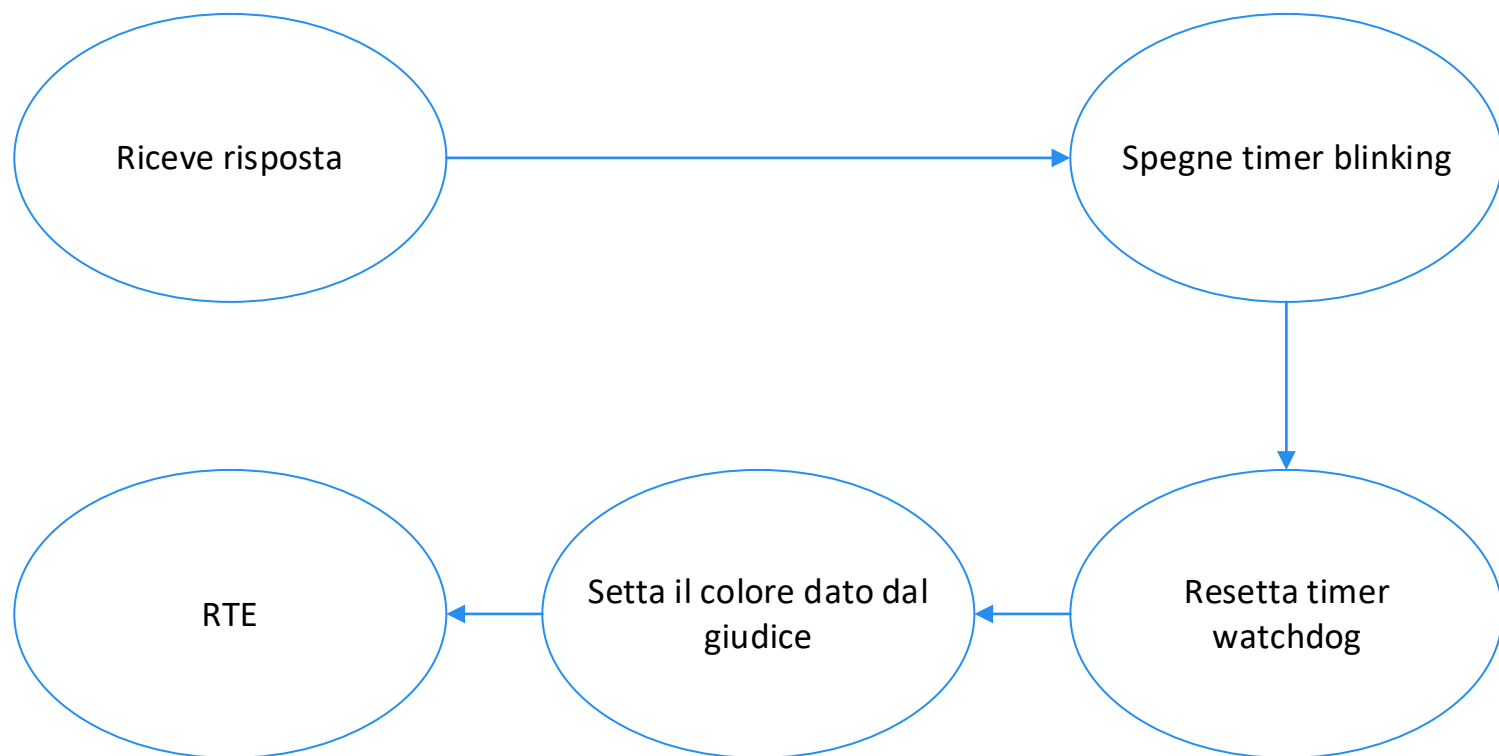
---

- L'automa descrive il flusso di esecuzione per la gestione dell'evento "pressione del bottone", con cui un pedone esprime la volontà di attraversare



## Automa ISR ricezione messaggio

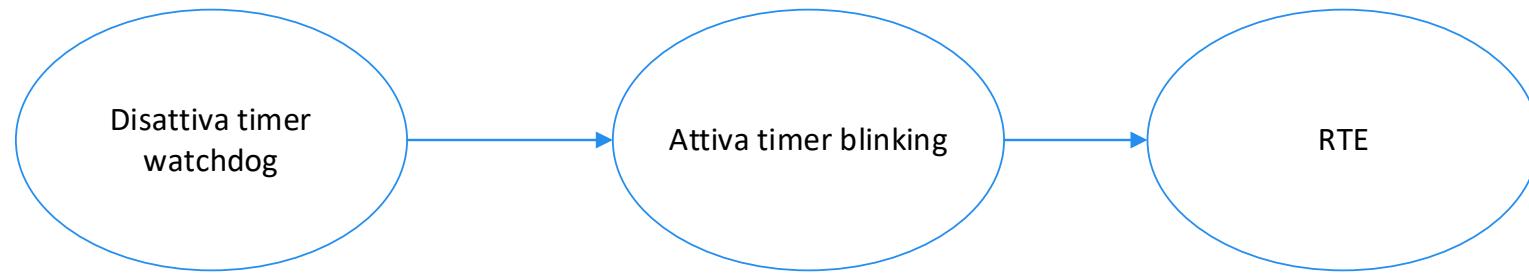
- L'automa descrive il flusso di esecuzione per la gestione dell'evento di ricezione di un messaggio.
- Un messaggio dal Giudice rappresenta, infatti, un messaggio di alive, che assicura che esso è ancora in funzione, e resetta il watchdog





# Automa ISR TimerWatchdog

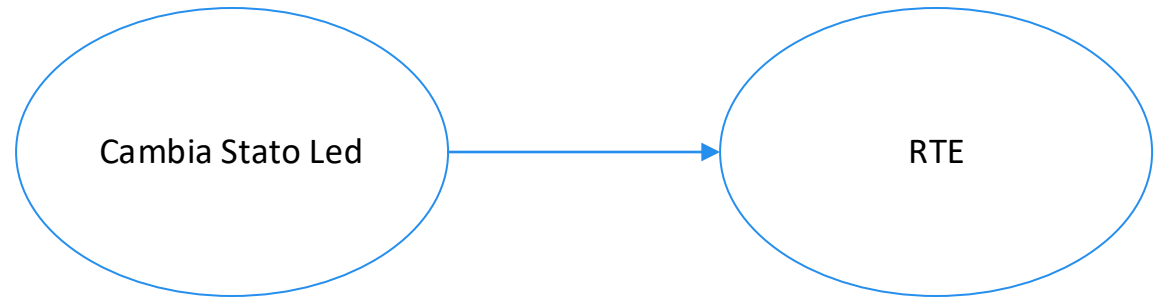
- L'automa descrive il flusso di esecuzione per la gestione dell'evento di scadenza del TimerWatchdog, che indica un failure del Giudice.
- La disattivazione viene effettuata per fare in modo che il timer non continui a scattare; esso sarà infatti riattivato alla ricezione di un nuovo messaggio/al restore del giudice



# Automa ISR TimerBlinking

---

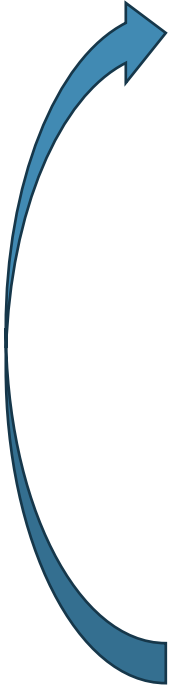
- L'automa descrive il flusso di esecuzione per la gestione dell'evento dello scadere del TimerBlinking, che mette in idle il semaforo qualora il Giudice dovesse fallire
- Ad ogni scadere del timer, si cambierà lo stato del led giallo, che passerà da acceso a spento e viceversa generando quindi un blinking
- Si uscirà dallo stato di idle alla ricezione di un nuovo messaggio dal giudice




# **Automati giudice**



# Pattern di attivazione

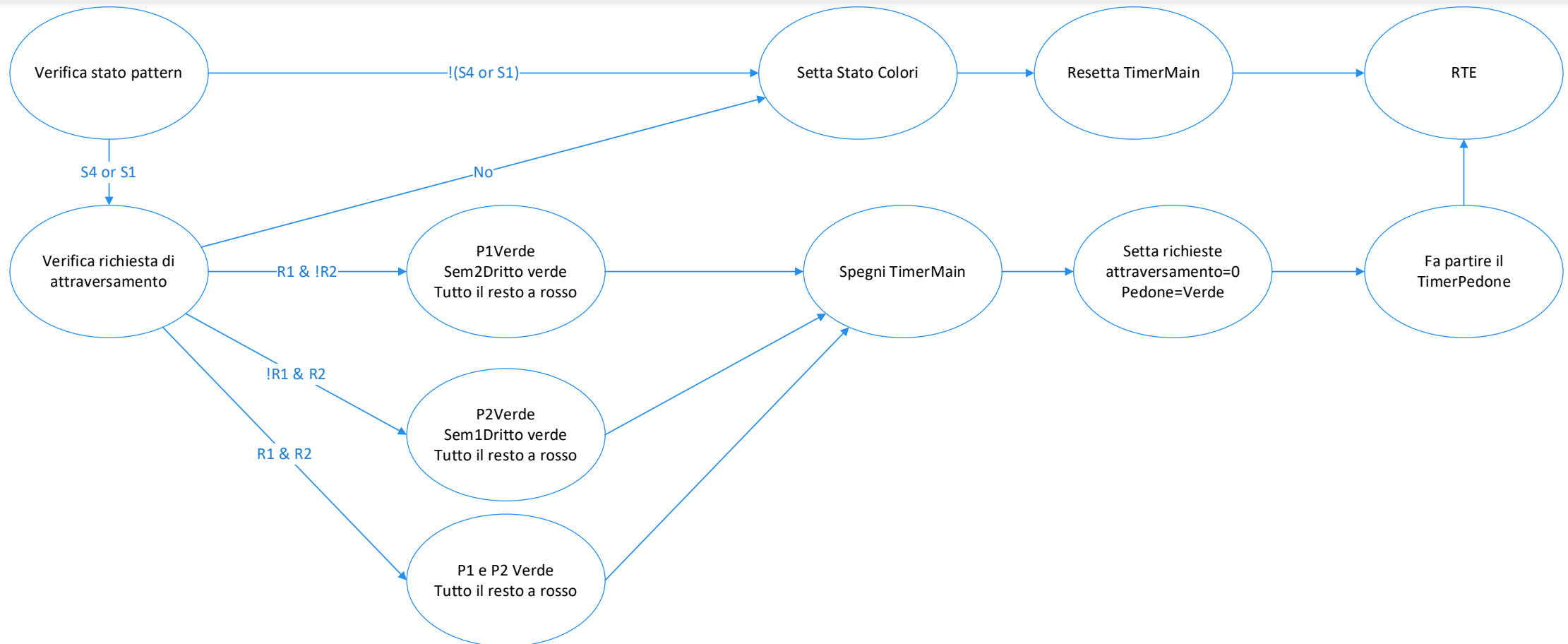
- 
- S0 Mettiamo tutti a rosso per 3 secondi
  - S1 Mettiamo a verde il semaforo1 per 3 secondi
  - S2 Mettiamo a giallo il semaforo1 per 2 secondi
  - S3 Mettiamo tutti a rosso per 3 secondi
  - S4 Mettiamo a verde il semaforo2 per 3 secondi
  - S5 Mettiamo a giallo il semaforo2 per 2 secondi

# Tabella codici di comando

Dritto	Svolta	Pedone	Simbolo
			A
			B
			C
			D
			E
			F
			G

# Automa TimerMain Giudice

- Il Main consisterà nel mandare un pattern di accensione ai diversi semafori, che si ripeterà ciclicamente.
- Si passerà da uno stato all'altro allo scadere del TimerMain, il quale prima di mettere uno dei due semafori a verde controllerà se vi è stata una richiesta di attraversamento
  - In questo caso, si interrompe il normale pattern di attivazione

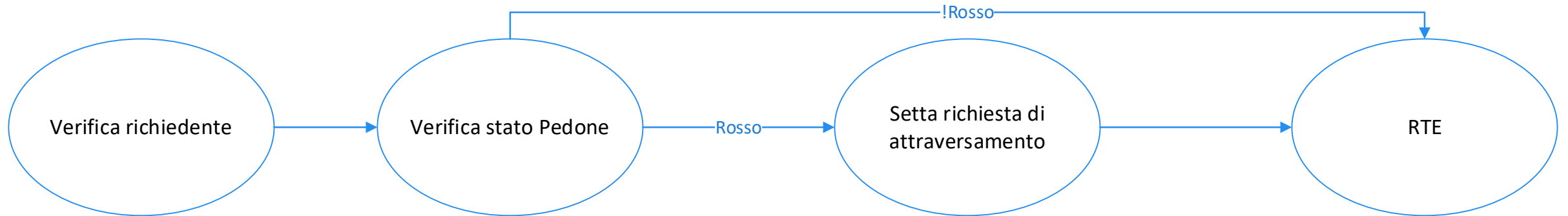


# Automa ISR

## Richiesta

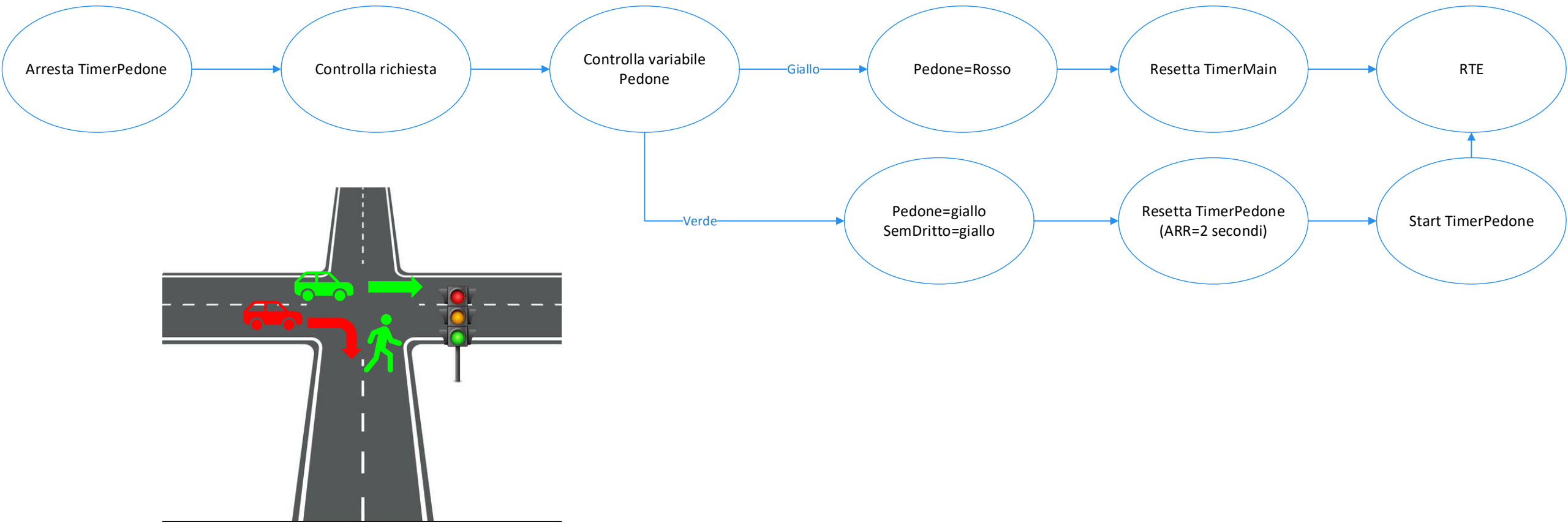
## Attraversamento

- L'automa descrive il flusso di esecuzione per la gestione di ricezione di una richiesta di attraversamento
- Prima di intraprendere alcuna azione, si controlla se si è già in una condizione in cui il pedone sta attraversando
  - Se stiamo già attraversando, non faremo niente, altrimenti setteremo la variabile di richiesta del relativo semaforo di provenienza



# Automa ISR TimerPedone

- L'automa descrive il flusso di esecuzione per la gestione dello scadere del TimerPedone
- A seconda della provenienza della richiesta (sem1 e/o sem2) setteremo il corrispondente pedone (P1 e/o P2) e il semDritto dell'altra strada
  - Se abbiamo una richiesta da entrambe le strade, tutti i semafori relativi ai veicoli saranno rossi e non ci sarà alcuna istruzione del tipo SemDritto=giallo





# Periferiche e componenti utilizzati

- Componenti aggiuntivi:
  - Led RGB
    - 1 per ogni semaforo, 6 in totale
  - BreadBoard
  - Cavi MM
  - Resistori
    - 300  $\Omega$  e 4.7 k $\Omega$
- Periferiche adoperate:
  - USART
  - User Button
  - **Timer**

*Nota: si presuppone l'utilizzo dell'NVIC in quanto opereremo in tutto il progetto utilizzando il meccanismo delle interruzioni*

# Timer

- Sia nel giudice che nei semafori, avremo bisogno di 2 timer
  - TimerWatchdog e TimerBlinking per i semafori
  - TimerMain e TimerPedone per il giudice
- In tutti e 4 i casi, i timer saranno settati in modalità base dei tempi
  - è definito l'intervallo di tempo allo scadere del quale vogliamo che avvenga un determinato evento
- Si utilizzeranno i Timer Basic e il prescaler per ottenere un conteggio dell'ordine dei secondi

# Mapping dei pin

---

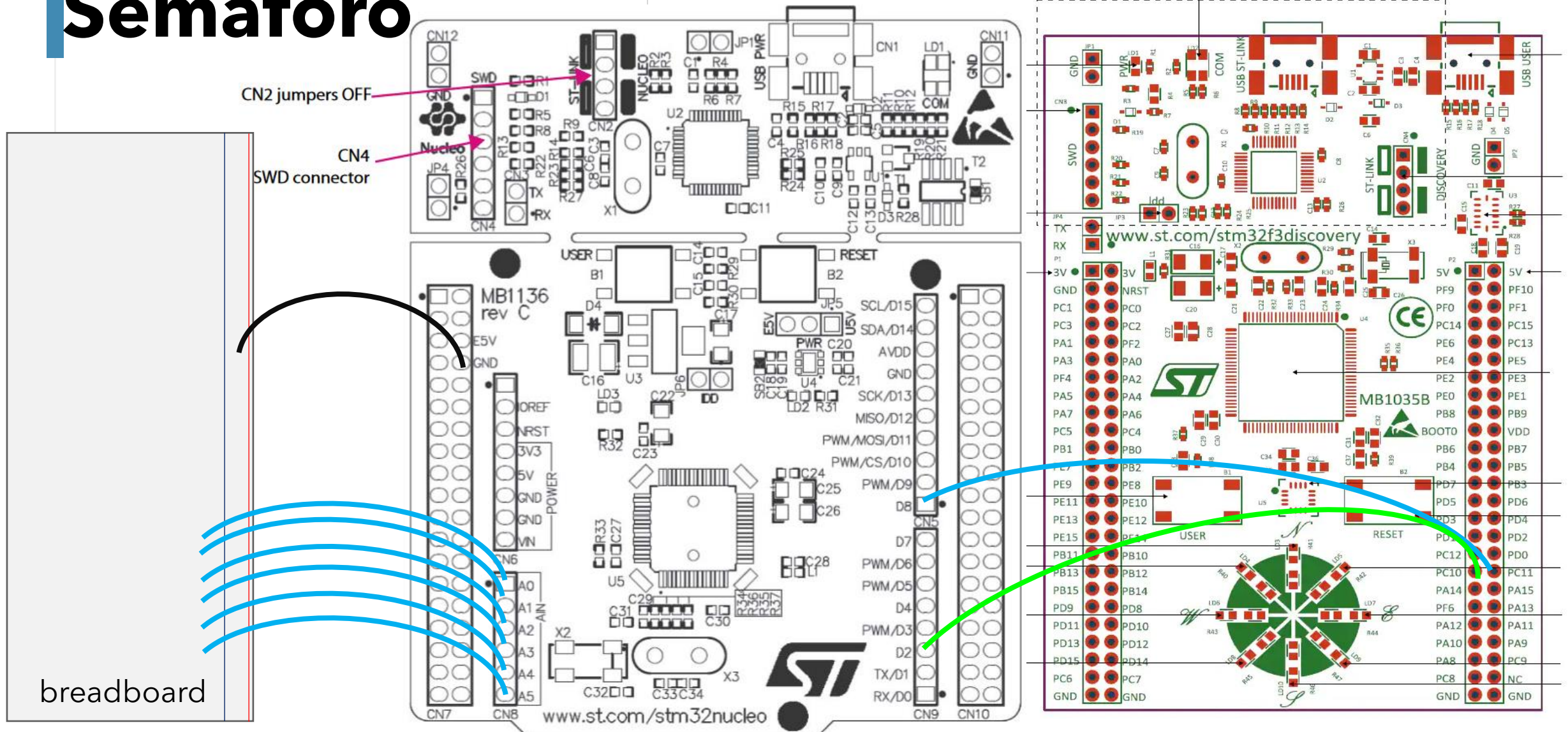
# Mapping Pin Semafori

- Led
  - Semaforo dritto: rosso sul pin PA0 e verde sul pin PA1
  - Semaforo svolta: rosso sul pin PA4 e verde sul pin PB0
  - Semaforo pedone: rosso sul pin PC1 e verde sul pin PC0
- Uart:
  - USART1\_RX mappato su PA10
  - USART1\_TX mappato su PA9
- Button:
  - User button mappato su PC13, interruzioni abilitate sul fronte di discesa
- Timer:
  - TimerBlinking sarà implementato utilizzando il Timer10
  - TimerWatchdog sarà implementato utilizzando il Timer11

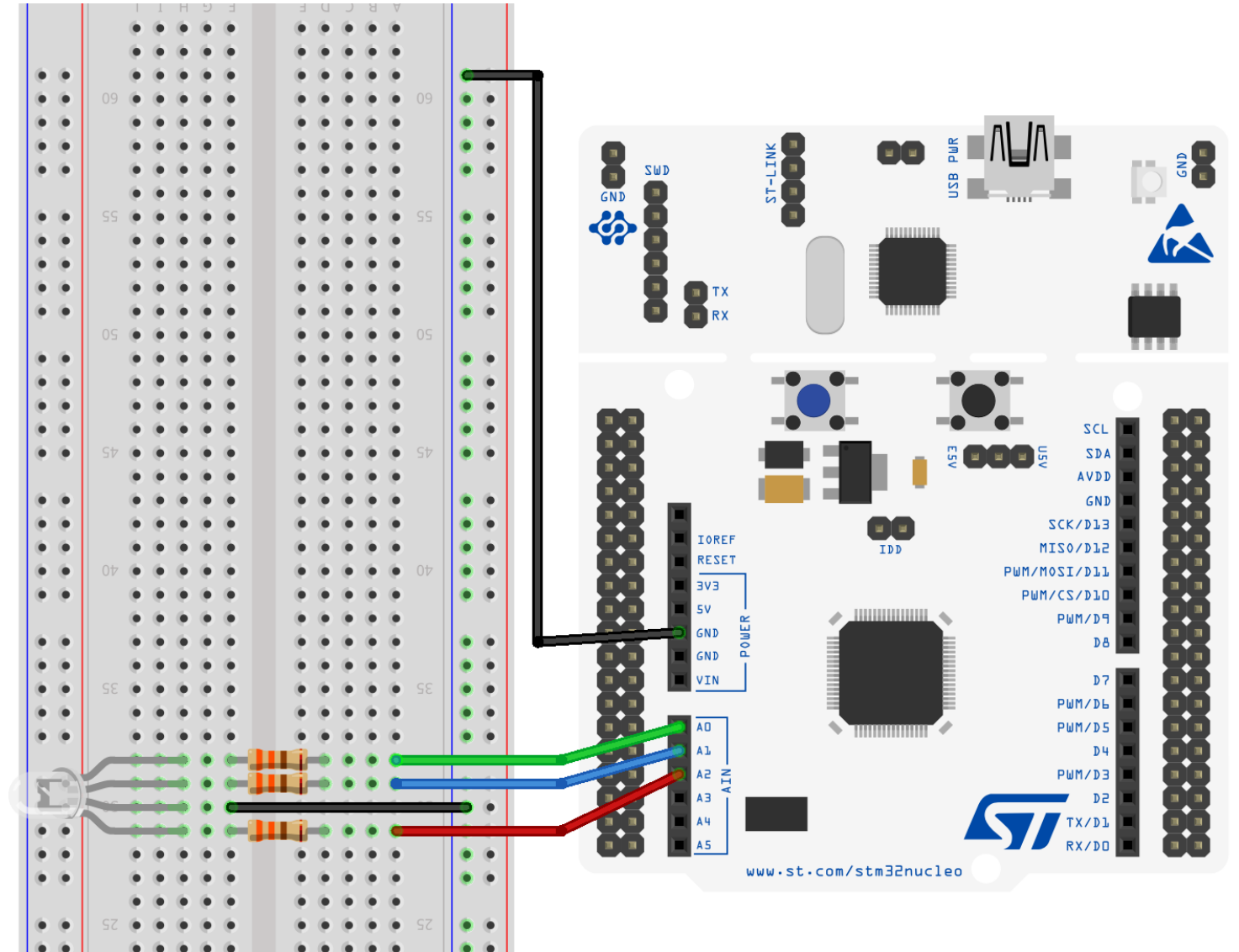
# Mapping Pin Giudice

- Giudice:
  - UART4\_RX mappato su PC11, UART4\_TX mappato su PC10, baud rate 9600
  - UART5\_RX mappato su PD2, UART5\_TX mappato su PC12, baud rate 9600
- Timer:
  - TimerMain sarà implementato utilizzando il Timer6
  - TimerPedone sarà implementato utilizzando il Timer7

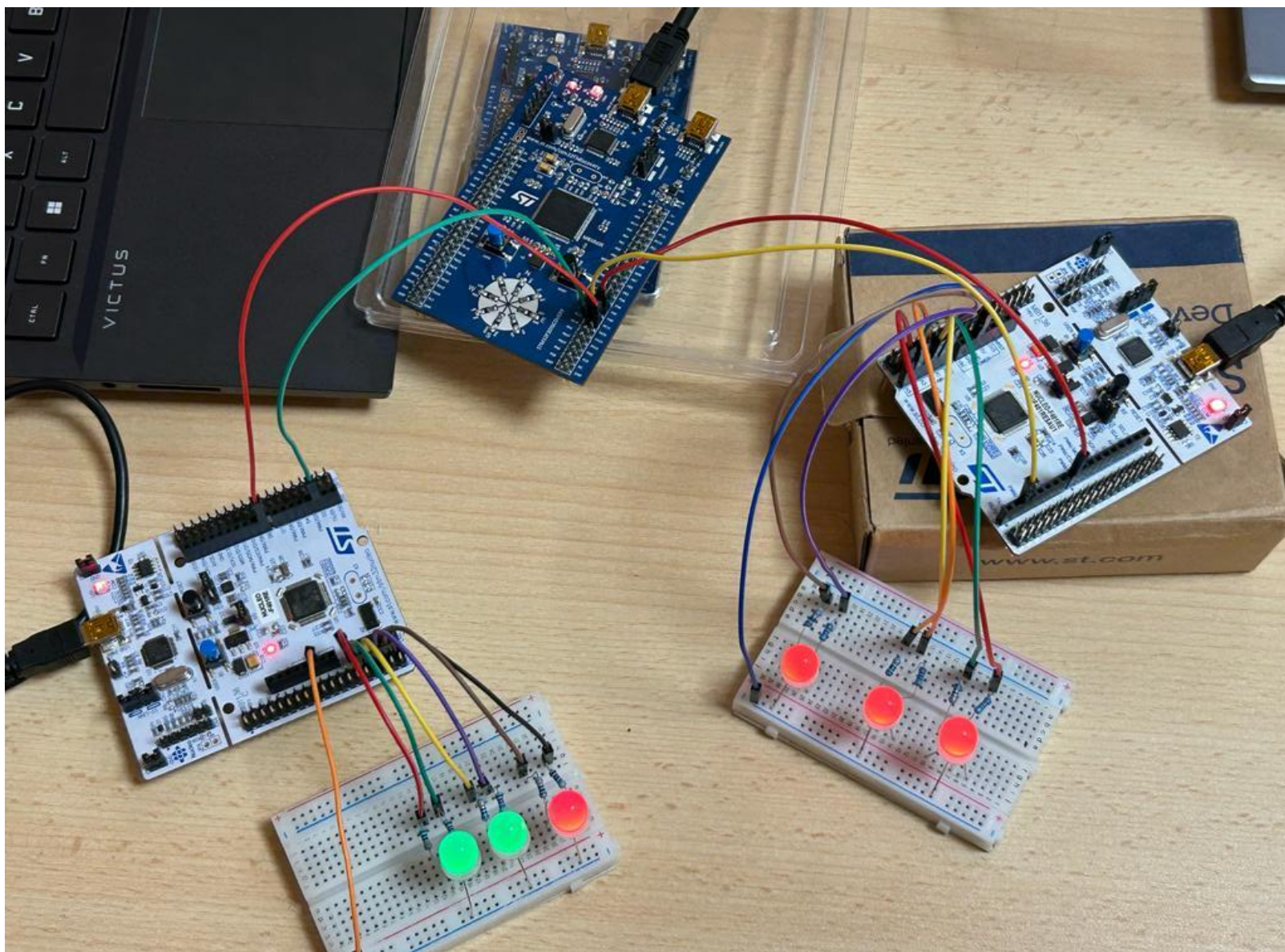
# Schema di collegamento di Giudice e Semaforo



# Schema di collegamento dei led RGB







**Collegamento  
effettivo del  
sistema**



# Video dimostrativo

