

Relazione verifica Prandi

- **Obbiettivo**

Realizzare un circuito elettrico attraverso dell'uso di un microprocessore Raspberry Pico nel quale accendiamo 4 led in successione tramite l'utilizzo di un pulsante e di un potenziometro

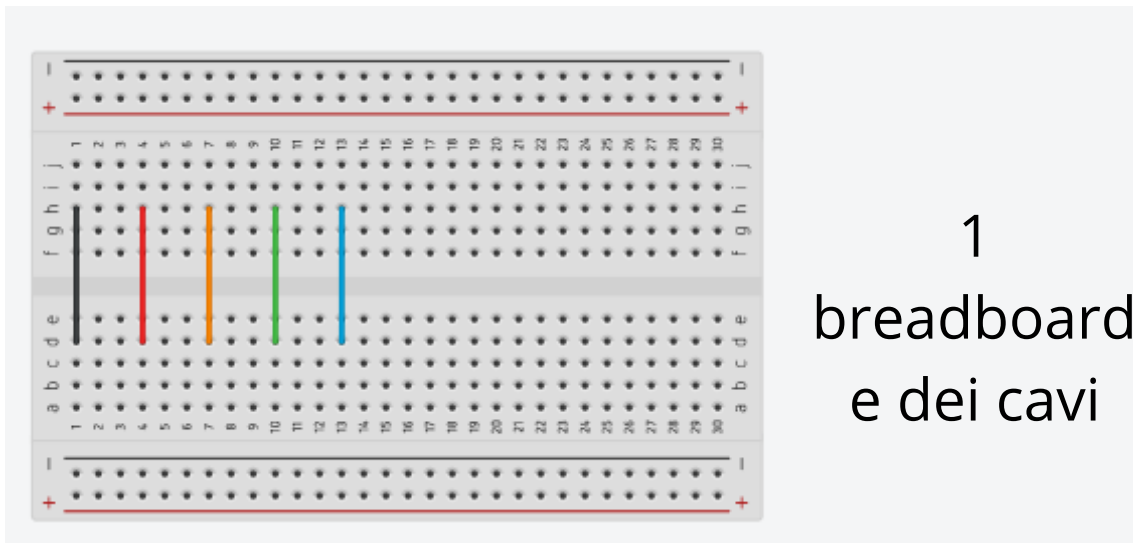
- **Cenni teorici**

Il Raspberry Pi Pico è una scheda di sviluppo basata su un microcontrollore ARM a basso costo e a basso consumo energetico, lanciato da Raspberry Pi Foundation. La scheda è dotata di una serie di periferiche di input/output, tra cui 26 pin GPIO, una porta micro USB per la programmazione e l'alimentazione, e una serie di connettori per l'interfacciamento con dispositivi esterni come sensori, attuatori e display. Può essere programmato utilizzando il linguaggio di programmazione Python, ma supporta anche C e C++.

Le resistenze sono componenti elettronici utilizzati per limitare la corrente in un circuito elettrico. La resistenza viene espressa in ohm (simbolo Ω) e rappresenta la misura della capacità di un materiale di opporsi al passaggio della corrente elettrica.

Una breadboard, o scheda sperimentale, è un dispositivo utilizzato per costruire e testare prototipi di circuiti elettronici. Consiste in una base, solitamente di plastica, sulla quale sono presenti una serie di fori, organizzati in righe e colonne, in cui possono essere inseriti i componenti elettronici.

- Materiale



4 led rossi

4 resistenze da 330 Ω per
i led

1 raspberry Pi Pico

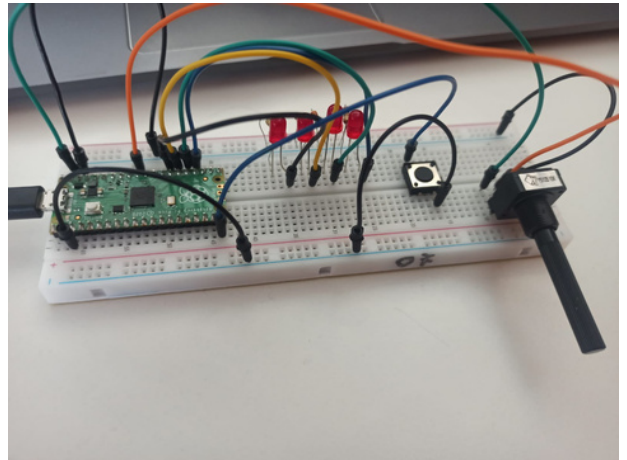
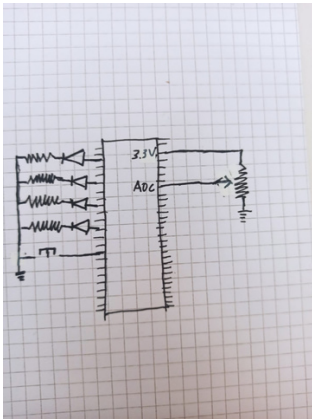
1 resistenza da 10 K Ω per il
pulsante

1 portatile

1 potenziometro

• Procedimento

- Si crea un collegamento con il pico creando il seguente schema:



- Successivamente si passa alla creazione del codice del Pico:

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/adc.h"
#define LED1 21
#define LED2 20
#define LED3 19
#define LED4 18
#define BUTTON 15
#define T 1000 //tempo di attesa tra i led
#define TUTTI 2000 //durata accensione di tutti i led

int main()
{
    float valore;
    float tempo;
    float risultato;
    gpio_init(LED1);
    gpio_init(LED2);
    gpio_init(LED3);
    gpio_init(LED4);
    gpio_init(BUTTON);
    gpio_set_dir(LED1, GPIO_OUT);
    gpio_set_dir(LED2, GPIO_OUT);
    gpio_set_dir(LED3, GPIO_OUT);
    gpio_set_dir(LED4, GPIO_OUT);
    gpio_set_dir(BUTTON, GPIO_IN);
    gpio_pull_up(BUTTON);
    adc_init();
    adc_select_input(0);
    const float conversion_factor = 3.3f / (1<<12);
```

```
while(true){
    valore = adc_read();
    risultato = valore * conversion_factor;
    tempo = 3500 - risultato*1000 + 200; //utilizzata per il tempo con il potenziometro
    //inizializzo tutto a 0 prima di iniziare
    gpio_put(LED1, 0);
    gpio_put(LED2, 0);
    gpio_put(LED3, 0);
    gpio_put(LED4, 0);

    if(!gpio_get(BUTTON)) {
        sleep_ms(TUTTI); //conto alla rovescia iniziale
        gpio_put(LED1, 1);
        sleep_ms(tempo);
        gpio_put(LED1, 0);
        gpio_put(LED2, 1);
        sleep_ms(tempo);
        gpio_put(LED2, 0);
        gpio_put(LED3, 1);
        sleep_ms(tempo);
        gpio_put(LED3, 0);
        gpio_put(LED4, 1);
        sleep_ms(tempo);
        gpio_put(LED4, 0);
        sleep_ms(TUTTI);

        gpio_put(LED3, 0);
        gpio_put(LED4, 1);
        sleep_ms(tempo);
        gpio_put(LED4, 0);
        sleep_ms(tempo);
        gpio_put(LED1, 1); //accensione di tutti i led per 2 secondi
        gpio_put(LED2, 1);
        gpio_put(LED3, 1);
        gpio_put(LED4, 1);
        sleep_ms(TUTTI);
    }

    return 0;
}
```

- Finito il seguente programma si esce dal programma e si entra sul terminale per compilare il codice dando le seguenti istruzioni:

-cd pico/esercizi/Esercizio4/build

)all'interno di build:

-cmake .. (da eseguire solo la prima volta che si compila).

-make -j2 (eseguire ogni volta che si modifica il codice).

- Per caricare il programma sul pico bisogna collegarlo al PC tenendo premuto il pulsante **RESET** presente sul Pico, se verrà riconosciuto aprirà una finestra del gestore file in cui andremo a inserire il nostro programma in questa modo:
- Da interfaccia grafica entrare nella directory **Pico** e poi in **esercizi**, si sceglie la directory dell'esercizio che si vuole caricare e si entra all'interno della directory **build** all'interno di essa.
- Successivamente si seleziona il file con denominazione **".uf2"** e si copia all'interno della scheda della directory aperta all'inserimento del Pico.
- Se eseguito correttamente il pico verrà riconosciuto dal Pc come dispositivo di archiviazione e inizierà ad eseguire il programma.

• Osservazioni

Al successivo inserimento del Pico nel PC, se si vuole solamente eseguire il codice caricato precedentemente non serve premere il pulsante RESET, ma bensì bisogna solamente inserire la USB nel PC.

• Conclusioni

La realizzazione del circuito e l'esecuzione del codice sono riuscite senza troppe complicazioni.