

Relazione es.2 Pico

- **Obbiettivo**

realizzare un circuito elettrico attraverso dell'uso di un microprocessore Raspberry Pico in cui lampeggi un led esterno.

- **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**

1 Breadboard

1 Resistenza da 330 Ω

1 Raspberry Pico

2 Cavi

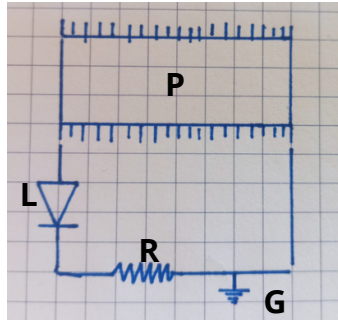
1 Led

1 Collegamento USB

1 Portatile

• Procedimento

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



Nel quale:

- P è il Pico
- L è il Led
- R è la Resistenza
- G è il Ground

- Successivamente si passa alla creazione del codice del Pico:

```
#include <stdio.h>
#include "pico/stdlib.h"

int main()
{
    const uint LED_PIN = 15;
    float T = 1000;
    float D = 0.3;
    float TD = T * (1-D);
    float TU = T*D;
    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);
    while(1){
        gpio_put(LED_PIN, 1);
        sleep_ms(TU);
        gpio_put(LED_PIN, 0);
        sleep_ms(TD);
    }
    return 0;
}
```

- Vengono dichiarate le librerie utilizzate.
- Si dichiara il pin di uscita del Led.
- Si calcola il Duty Cycle.
- Si inizializza il Led e si dichiara come variabile di Output.
- Applicando la formula usata in precedenza si assegna valore logico alto e basso al Led usando la durata in cui rimane acceso e spento per determinare l'intensità del Led.

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

-cd pico/esercizi/Esercizio2/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 questo 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, ovvero durante la scrittura del codice ricordare che l'intensità non è espressa in percentuale quindi bisogna ricordarsi di inserire il valore in decimi in quanto deve essere un valore compreso tra 0 e 1.