# Progetto di Internet of things

* Obiettivi
  + Smart Street Light
  + Cos’è
  + Qual è il suo obiettivo
* Contributi
  + Breve paragrafo sui contributi dati dai singoli componenti del gruppo
* Realizzazione pratica
  + Materiali utilizzati
  + Sistema fisico, come vengono collegati i vari elementi
  + Spiegazione codice parte automatica
  + Sistema web
  + Spiegazione codice parte web
  + Sicurezza e gestione degli errori

# Smart Street Lights

## Introduzione

Il progetto si basa sulla realizzazione di un sistema IoT il cui scopo è la gestione automatica di lampioni stradali che si accendono e si spengono in risposta a determinati eventi.

Il funzionamento del sistema riguarda due modalità, quella automatica che gestisce il modo autonomo i lampioni, ovvero accende e spegne quest’ultimi in base alla luce esterna rilevata. Inoltre regola l’intensità della luce, quando accesa, che aumenta al passaggio di veicoli o pedoni in prossimità dei lampioni grazie all’attivazione dei sensori a infrarossi. L’altra modalità è quella manuale che sfrutta una pagina web dove un addetto può pilotare i vari lampioni in tre modalità: automatica, accesa alla luminosità massima oppure spenta.

## Obiettivi e contributi del gruppo

L’obiettivo principale di questo sistema è il risparmio energetico, in quanto riduce i consumi nelle ore in cui le strade sono meno trafficate, mantenendo sempre una buona visibilità sulla strada. Inoltre le luci possono essere comandate singolarmente e quindi è possibile aumentare la loro intensità in caso di necessità, ad esempio per lavori in corso, attraversamenti pedonali, emergenze ecc.

Il progetto è stato eseguito da Leonardo Scandino e Marco Tateo per lo più in collaborazione, ma concentrandosi alle volte singolarmente su vari aspetti del progetto, ovvero il primo sulla parte automatica del sistema e le sue parti fisiche (raspberry, sensori ecc.), e il secondo sulla parte web e la comunicazione tra quest’ultima e la parte fisica.

## Materiali e strumenti utilizzati

* Raspberry pi 4
* Sensori
  + Due sensori a infrarossi
  + Fotoresistenza
* Due luci dimmerabili a 12v
* Trasformatore da 230v a 12v
* Breadboard, jumper e condensatore
* Due moduli mosfet per comandare le luci
* Flask

### Collegamenti dei componenti

Il cuore del sistema è un raspberry pi 4 utilizzato per gestire tutti gli elementi del progetto, alle varie GPIO disponibili sono collegati tutti i sensori e tutte le luci.

I sensori a infrarossi presentano tre pin ovvero, alimentazione (collegata alla GPIO 3.3v ), neutro(collegato al GND) e il segnale collegati rispettivamente alle due GPIO 22 e 23.

La fotoresistenza invece ha bisogno di un condensatore da inserire condensatore per calcolare il livello di luce e come per i precedenti ha bisogno di alimentazione e GND, il segnale passa per la GPIO 4.

La parte più complicata dei collegamenti sono le luci. Le nostre luci devono essere alimentate a 12v quindi abbiamo bisogno di un trasformatore e di un modulo mosfet per riuscire a comandarle a nostro piacimento. Le due luci sono collegate ai due canali pwm presenti su Raspberry ovvero GPIO 13 e 18.

Per semplicità tutto quanto è stato collegato ad una breadboard tramite dei jumper.

## Parte web

In questa fase si è fatto uso di un framework di nome Flask

* Cos’ è flask
* Come funziona
* Cosa fa la parte web

------scrivi tu marco-------

## Sviluppo del codice

Il codice è suddiviso in due file ovvero mainApp.py che è il nostro codice principale scritto in python e il main.html che è il codice html per la parte web.

In mainApp.py vengono importate le librerie che serviranno nel corso del progetto ovvero RPi.GPIO, time, multiprocessing e Flask.

Per fare interagire la parte web e la parte automatica viene utilizzato un dizionario denominato lights dove inseriamo le informazioni più importanti ovvero: i pin delle luci, il pulsante selezionato nella pagina web, il pin del sensore a infrarossi, il pwm di riferimento e il processo.

A questo punto vengono settati tutti i sensori e tutte le luci tramite dei cicli e salvati nel dizionario.

Il funzionamento della parte automatica è essenzialmente sviluppata in due definizioni principale, ovvero dimmerLuce e valueFoto. La prima si occupa di leggere valueFoto e se il valore è alto accende le luci al 40%, dal momento che le luci sono accese e un veicolo passa in prossimità di un sensore a infrasrossi la luce di riferimento si accenderà per due secondi al 100% per poi tornare al 40%. La seconda si occupa di leggere la fotoresistenza e ricavarne il valore.

In seguito c’è la parte di codice che si interfaccia anche alla parte web descritta in precedenza.

Come prima cosa viene passato il dizionario iniziale alla nostra pagina web che farà visualizzare lo stato iniziale, a questo punto viene inserita una definizione click che cambia lo stato del nostro dizionario e lo aggiorna.

Se si clicca su Auto viene modificato il pulsante in AUTO, viene spenta qualsiasi luce, in quanto nella modalità automatica se è giorno la luce non deve essere accesa, e infine attiva il processo che sfrutta dimmerLuce e valueFoto (inserendo nel dizionario che la luce di riferimento ha quel processo in esecuzione).

Se si clicca su OFF viene modificato il pulsante in OFF, il codice verifica se il processo è vivo e in caso affermativo lo termina. Il funzionamento di ON è lo stesso solo che accende la luce al 100%

Ad ogni click il dizionario viene aggiornato.

Nella parte html si costruisce la pagina web e grazie al dizionario sempre aggiornato tramite delle condizione modifichiamo i pulsanti e i colori in modo da visualizzare sullo schermo lo stato di ogni luce in quel momento.

## Sicurezza

* Idee per rendere più sicuro il codice