Progetto Lockit.

# Creazione di un interfaccia grafica UI che comunica con un dispositivo seriale virtuale

## Creatore: Francesco Pugliese

Tempo a disposizione: 1 settimana(11/06/2024-18/06/2024**)**

-winget install OpenJS.NodeJS.LTS.

-restart della power shell.

- mkdir App

-cd App

- npm install electron --save-dev.

-npx electron .

- npm i tableify..

- npm i virtual-serialport.

-npm install serialport.

-Installare Electron Electron Fiddle.

-Installare Virtual Serial Port Drive.

- Aprire Virtual Serial Port Drive e seleziona il loopback COM9.

- Aprire il progetto da Electron Fiddle.

-Clicare su Run.

- Cliccare sui Led.

# Spiegazione architetturale

Il progetto è stato realizzato in javascript e si è utilizzato il framework Electron Fiddle.

Il progetto prevede una cartella con i moduli installati, come serialport che è la libreria utilizzata per creare la comunicazione con la seriale virtuale. Inoltre ci sono i seguenti file:

* Index.html: creazione degli oggetti grafici di tipo led. Hanno una callback sul tag onclick che indirizza l’applicazione su una callback uguale per tutti i led.
* Renderer.js: processo di renderer che comunica con il main.js attraverso l’IPCRenderer, attraverso la callback. La callback prevede la costruzione del comando “LEDxy” e invio del comando attraverso l’IPC al main.js.
* Main.js: File principale dell’applicazione, che prevede la costruzione della finestra principale, il caricamento degli oggetti grafici, la creazione della comunicazione con la porta seriale virtuale e la comunicazione con il processo renderer.js attraverso il processo IPCMain. Una volta che il main ha aperto la comunicazione ascolta attraverso IPCMain ciò che manda il processo renderer. Il processo renderer manda il comando “LEDxy” che poi viene scritto sulla seriale virtuale dal main. La seriale che è un loopback reinvia in formato esadecimale il comando che viene trasformato in ascii. Solitamente se il loopback è andato a buon fine viene inviato in maniera asincrona il comando <comandoRicevuto>+OK al renderer, che verifica il messaggio e cambia il colore del led selezionato.

**DA VERIFICARE:**

* Docker: il processo è compilato correttamente per windows, ma per lanciarlo è necessario l’installazione di un server remoto. Inoltre l’interfaccia GUI su Windows non è applicabile **:**
* Style.css: file per impostare il colore dei led.

Windows:

docker build -t my-node-app **.**

docker run -p 3000:3000 my-node-app

Ho scelto Javascript perché avevo più liberta di gestire le variabili che possono non avere un tipo al contrario di Typescript, che è una estensione di Js.

Istruzioni per il Typescript.

* Entrare nel prompt e selezionare la cartelle di progetto.
* # install typescript globally
* npm install typescript@latest -g
* # generate tsconfig.json file
* npx tsc –init