

ALPHABOT A COMANDO WIRELESS

Nome: Alice Fasulo

Classe: 5^A ROB

Descrizione degli obiettivi da raggiungere:

1. Flashare una sim e configurarla con i dati necessari al progetto
2. Connettere l'alphabot al proprio computer
3. Testare e personalizzare, implementando un client e un server, la libreria base fornita dal professore
4. Imparare la gestione dei sensori ad infrarossi dell'alphabot
5. Imparare la gestione della libreria pynput
6. Gestire il movimento dell'alphabot tramite l'implementazione di un database

Prerequisiti	<ul style="list-style-type: none"> - balenaEtcher - python - SQLite - PuTTY - WinSCP - socket - accesso all'interfaccia web di amministrazione del router
Conoscenze	<ul style="list-style-type: none"> - cosa sono e a cosa servono i sensori ad infrarossi - cosa sono e a cosa servono i sensori ad ostacoli - utilità e contenuto della libreria pynput - a cosa serve la libreria SQLite3 - come funzionano i motori dell'alphabot - come si calcola l'avanzamento dell'alphabot - funzionamento della libreria alphabot
Abilità	<ul style="list-style-type: none"> - funzionamento e gestione dei sensori ad infrarossi - funzionamento e gestione dei sensori ad infrarossi - gestione della libreria pynput - gestione della libreria SQLite3 - gestione dei motori dell'alphabot - applicazione della libreria alphabot
Competenze	<ul style="list-style-type: none"> - fare una relazione - comandare un alphabot attraverso applicazione client-server - gestire un database in python - gestione dei comandi di movimento per l'alphabot in modalità controller

Abstract

Questa relazione descrive il percorso svolto a scuola per l'acquisizione di competenze avanzate nella gestione di un alphabot. L'attività è partita dalle basi, come la configurazione di una scheda SIM e l'utilizzo della libreria alphabot, per arrivare allo

sviluppo di competenze complete orientate alla realizzazione di:

- una comunicazione client/server per il controllo dell'alphabot in python;
- un sistema di gestione dell'alphabot tramite database SQLite.

L'intero progetto è stato svolto in 11 ore di lezione, durante le quali la classe, organizzata in coppie, ha lavorato per sperimentare e applicare tutte le fasi del percorso.

Dettaglio delle fasi operative:

lezione 1:

- dopo essersi procurati una sim, un alphabot e un computer, inserire la sim all'interno del pc e flesharla tramite l'ausilio dell'applicazione balenaEtcher
- inserire la sim in una tastiera esterna raspberry, collegare la tastiera a un monitor e configurare la sim
- CODICE SE NON CI STANNO APPENDICE
- inserire la sim nell'alphabot e procedere all'accensione dell'alphabot, possibile tramite l'ausilio di batterie ricaricabili
- accendere il router
- aprire sul computer il sito web archer, in modo tale da verificare che la sim sia stata configurata correttamente

lezione 2:

- adattare il codice fornito dal professore, il quale presentava già le funzioni avanti, indietro, destra e sinistra, aggiungendo le funzioni di rotazione destra di 90 gradi e sinistra di 90 gradi. Il tempo di rotazione può essere misurato sperimentalmente
- implementare anche il controllo dell'alphabot tramite l'inserimento di input da parte dell'utente, come per esempio "avanti" per far avanzare il robot
- accendere l'alphabot e inserire la sim
- terminato il codice, aprire il sito web archer sul pc per visualizzare l'ip dell'alphabot, riconoscibile dal nome dell'alphabot deciso inizialmente
- aprire l'applicazione PuTTY ed avviare la connessione computer-alphabot scrivendo "py" e inserendo la password "raspberrry"
- aprire l'applicazione WinSCP, inserire l'ip e il nome dell'alphabot, avviare la connessione e spostare i file client, server, libreria alphabot presenti sul pc nell'alphabot

- testare il lavoro svolto comandando il client dal pc e avviando il server sull'alphabot tramite l'applicazione PuTTY

lezione 3 e 4:

- adattare i sensori in modo tale da rilevare l'ostacolo alla distanza desiderata
- implementare il vecchio codice aggiungendo la gestione dei sensori, iniziando con un codice che permette la fermata del robot in caso di rilevamento ostacolo
- implementare un secondo codice in cui, anche se l'utente preme il comando di avanzamento, il robot si fermerà se rileva un ostacolo
- per testare il codice svolgere i passaggi elencati nel punto 2

lezione 5:

- modificare il codice sviluppato precedentemente, eliminando la gestione del movimento del robot tramite parole e sostituendola con il controllo WASD. Da qui in avanti l'alphabot si muoverà in base al tasto premuto e si fermerà quando il tasto non è più premuto
- per testare il codice svolgere i passaggi elencati nel punto 2

lezione 6:

- sviluppare un database in SQLite3 contenente tutti i comandi necessari per il movimento dell'alphabot
- sviluppare un codice che legga il database e permetta il movimento del robot in base all'input inserito dall'utente
- per testare il codice svolgere i passaggi elencati nel punto 2

Risultati raggiunti e autovalutazione

il progetto è stato completato nella sua pienezza, senza tralasciare nessun punto di quelli forniti dal professore. Nel complesso siamo molto soddisfatte del risultato raggiunto

Criticità riscontrate e modalità di gestione delle criticità

- sensori: non sempre i sensori misuravano come previsto durante la fase di testing, per questo motivo, sotto consiglio del professore, è stato necessario regolare anche l'altezza del sensore, altrimenti rilevava come ostacolo il pavimento o non rilevava mai nessun ostacolo quando era rivolto verso l'alto. Un secondo problema è stato il sole, in quanto se l'alphabot era esposto ai raggi solari, rilevava sempre un ostacolo per via della luce. Per questo motivo è stato necessario svolgere tutti i test all'ombra.
- collegamento con l'alphabot: nella seconda lezione abbiamo avuto dei problemi a collegare l'alphabot, in quanto non riuscivamo a individuarlo nel sito web archer.

dopo diversi tentativi e controlli anche sulla sim, è stato rilevato che il problema era il robot in sé. Per questo motivo è stato necessario cambiare alphabot.

Prospettive di sviluppo future

Durante questo progetto abbiamo raggiunto un grande obiettivo, ma il lavoro non è ancora terminato. Nelle prossime lezioni si dovrà finire l'implementazione di un database ricco di comandi in modo tale da poter comandare il nostro alphabot con funzioni anche più complesse e terminato ciò grazie all'aiuto dei professori proveremo a implementare il framework Flask così da poter implementare un'applicazione web in grado di comandare l'alphabot come un vero e proprio controller