WEB BASED SERVICE ROBOT

Karim Hany, Eman Siraj, noha ahmed, Hagar Khaled, Mariam Metwally, Rodina Ayman, Heba Ahmed, Basma Farid

Faculty of Engineering, Canadian International College
Department of Communication and Electronics

Abstract **-** Open-source software and low-cost sensors bring advantages to the robotics community, facilitating the access to a wide range of robotic applications, which play an important role in recent advances. In this project, a web-based robot control system is designed in order to help in decreasing the infected people in hospitals thus designing an autonomous service robot that has the ability to serve the patients with high efficiency can decrease the interaction between the patient and doctors. Our contribution is to design a full 3D indoor mapping system that can reach high performance and high efficiency to track and control the movement of the robot through a web based interface. The indoor mapping system is mainly consists of robot operating system (ROS), MQTT, slam, low-cost sensors, such as the LIDAR, zed camera. In addition, the web application is designed with Html, CSS and JavaScript.

I. INTRODUCTION

Recently, new products and designs of robots is investigated in the market with innovation, better efficiency and accuracy. Most robots are designed to assist humans in different fields as dangerous, medical services, factories and repetitive tasks. Those robots work semi-autonomously to perform services useful to humans, while also being capable of performing primary goals. The term autonomous means 'having the freedom to act independently' [1]. Autonomous robots do not rely on an operator, but they make decisions on their own and perform tasks, such as transporting material while navigating in uncertain terrain and in a constantly changing environment. They even have their own computer, which is the "microcontroller" [1]. Any autonomous robot should be capable of navigating through a room thus our robot is being controlled via a web application user interface and can performs those main functions:

1. Generation a map of surround environment

2. Localization

3. Path planning

4. Obstacle detection

These specifications are correspond to some of the expected behaviors that are programmed into the robot. The main goal of this project is to make an autonomous robot capable of moving in the environment around it and controlling it from the web based interface with efficient and accurate positioning indoor system.

II. HARDWARE SETUP

A. ZED Camera

ZED Camera is a passive stereo camera that reproduces the way human vision works. Using its two lenses ("eyes") to create a three-dimensional map (3D MAP) of the scene by comparing the displacement of pixels between the left and right images. [2]

B. Lidar (light detection and ranging)

Lidar measures the time of flight of a pulse of light to be able to tell the distance between the sensor and an object.

C. Encoder

Encoder is a sensing device that provides feedback. Encoders convert motion to an electrical signal that can be read by some type of control device in a motion control system.

D. NVidia Jetson Xavier

NVidia Jetson is a series of embedded computing boards from NVidia. Jetson is a low-power system and is designed for accelerating machine learning applications.

E. DC – DC Converter

A DC-to-DC converter is an electronic circuit or electromechanical device that converts a source of direct current (DC) from one voltage level to another. It Reduce 24 volts to 12 volts to be suitable with the microcontroller.

F. DC motor

A DC motor is converts electrical energy into mechanical energy to allow the movement of the wheels of the robot.

III. USER INTERFACE

Figure 1: Home Page

We built and designed a user friendly web application so that the users can control and monitor the robot easily and efficiently. The languages used to build and connect the web pages: HTML &CSS, Java Script and Python language (Flask framework). We built our web pages using HTML and designed it using CSS. We also used Java Script for the some functions to create the web pages. In the Home page as shown in figure 1, the user can use from 2 modes for controlling the robot which are remote mode and automatic mode.

A. *Remote*

As shown in figure 2, the manual page contains arrows for the user to move the robot and see the response of the command that given if it has been delivered and executed correctly or wrong The video stream is shown at the right side of the page.

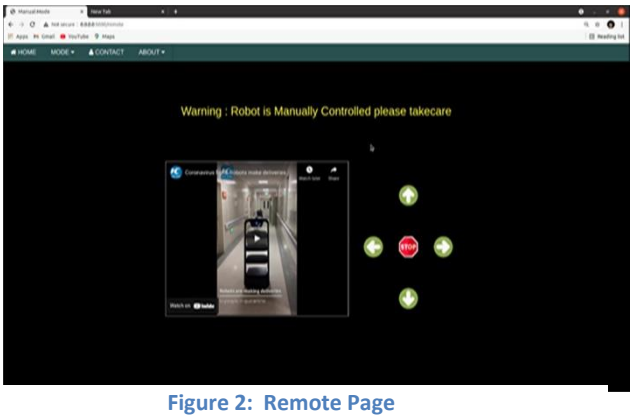

Figure 2: Remote Page

B. *Automatic*

As shown in figure 3, the page has a map where the user can set the goal he wants the robot to go to by clicking on that point, then click go. The robot will automatically determine the best path to the goal you set and start moving to follow that path.

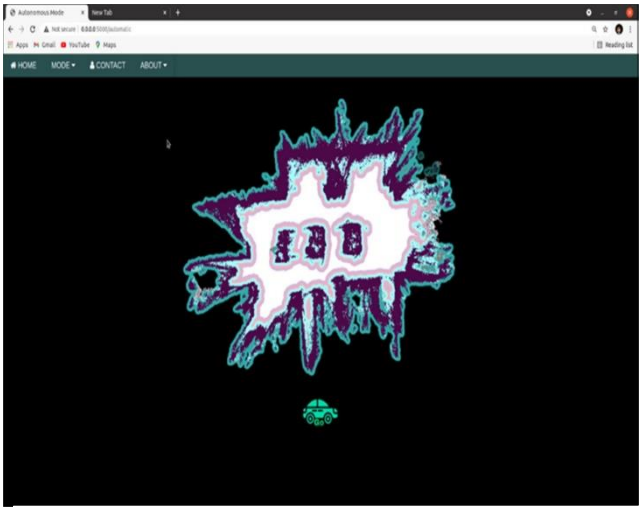

Figure 3: Auto Page

At this point each of the 3 pages are separate and not connected, so we need to connect them to turn them from 3 separate web pages to a web application. Here comes the function of flask.

IV. BACK-END

A. *Flask*

Flask is known as a micro-framework written in python because it is lightweight and only provides components that are essential, such as routing, request handling, sessions, and so on. The "micro" in micro framework means Flask aims to keep the core simple but extensible. Some features which make Flask an ideal framework for web application development are:

- Flask provides a development server as show in figure 4.

- Many extensions are available for Flask, which can be used to enhance its functionalities.

We used flask to be provided with tools, libraries and technologies that allow us to build a web application connecting the web pages and generating a server.[3]

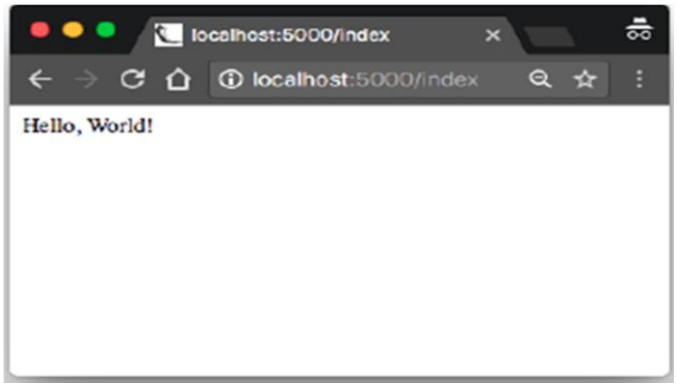

Figure 4: Local Server

B. *MQTT*

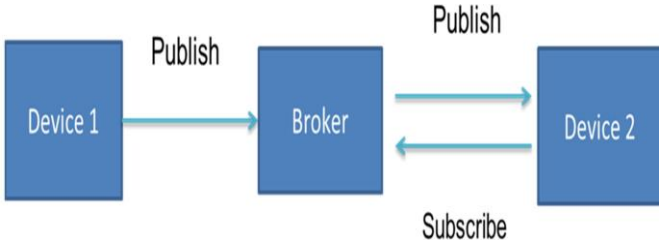

Figure 5: MQTT Block Diagram

Message queuing telemetry transport is an IOT machine to machine communication protocol. After connecting the web pages using flask to develop the web application we needed a communication protocol to connect the flask with the robot operating system. The MQTT protocol allows multiple devices to communicate with each other by subscribing to an agreed upon topic on a broker (server).

Also it is easy for us to implement as MQTT already has library in the python language which we are using. In our project the web application poses as the publisher (device 1 in

figure 5) which send the commands as a message to the robot which poses as the subscriber (device 2 in figure 5) via the broker, we used the hive MQ's broker. The messages being communicated in our project using MQTT are: command of motion from manual page for the robot to move in a certain direction or way as shown in Figure 6, reading of the sensors from the robot and response of the robot to the commands

With MQTT we have managed to successfully implement remote sensing and control of the robot and to efficiently delivering the commands to the robot operating system and receiving the response and reading of the sensors.

After connecting the web pages with flask then connecting the flask with the robot operating system we have managed to successfully to develop a web application that can automatically control and monitor the robot.

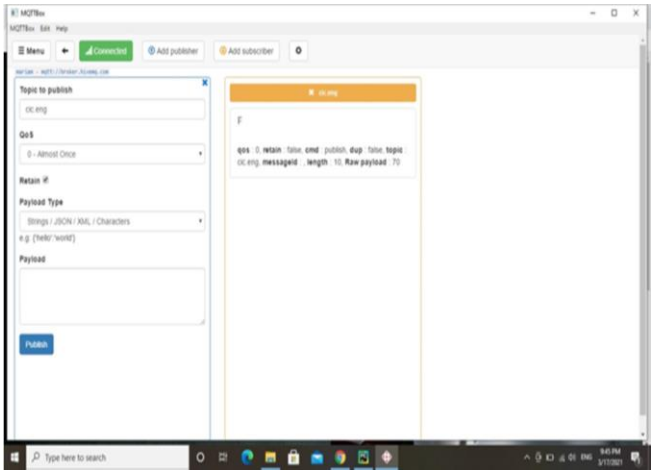

Figure 6: MQTT Box

C. Robot Operating System (ROS 2)

ROS is an open source programming framework that provides a library of reusable code packages for robot development, provide a run time environment and connect nodes together.[4] Ros is the operating system of the robot that is responsible for the synchronization and communicationbetween all the nodes of the robot. ROS's built-in messaging system saves you time by managing the communication between nodes via the anonymous publish/subscribemechanism such as shown in the figure 7.

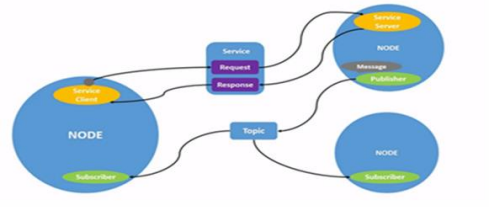

Figure 7: Ros Message Passing

D. SLAM Algorithm

Simultaneous localization and mapping (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it and it is done in four steps which are:

1. MAPPING
2. LOCALIZATION
3. PLANNING
4. CONTROL

We used R-TABMAP as SLAM package with stereo camera (ZED2) And this is the block diagram that describe the overall system

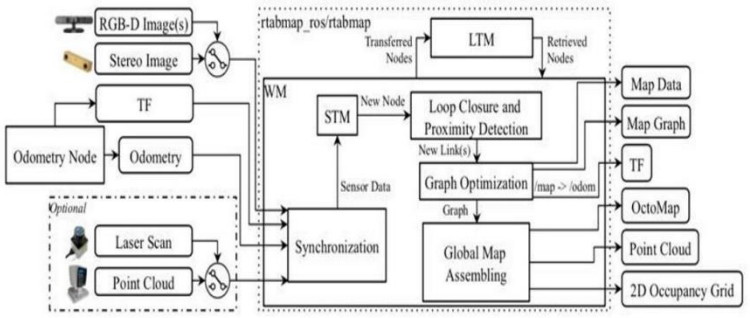

Figure 8 RTAB-MAP OVERVIEW

E. Ros Visualization

The RVIZ is a 3D visualization tool for ROS applications that: capture sensor information from robot sensors, display data from camera, lasers, from 3D and 2D devices including pictures and point clouds. Shown in figures 8 a map we have generated of the ITI Library.

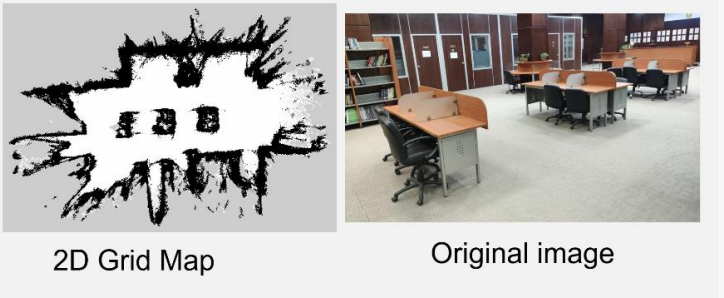

Figure 8: Map Generated of ITI Library

I. RESULTS

We have managed to develop a web-based Robot that can be controlled through web application access by any device like android mobiles or pcs that have access to the internet by doing following steps:

1. Generating a reliable map that illustrate the surrounding environment accurately using SLAM algorithm.
2. Designing a web application to control Robot
3. Using FLASK to generate a local server
4. Using MQTT protocol as our IOT protocol

II. CONCLUSION

We have successfully managed to design and implement a web-based service robot controlled via a web application interface built and designed with HTML, CSS and JavaScript connected by flask and capable of communicating with ROS by the MQTT protocol.

The robot purpose is to:
- Decrease medical staff interaction with patients.
- Decrease the severity and risk of infection.
- Autonomous task performing without direct control
- Enhancing Hospital Performance

III. REFERENCES

i. Ben-Ari, Mordechai, and Francesco Mondada. Elements of robotics. Springer Nature, 2017.
ii. https://www.stereolabs.com/assets/datasheets/zed2- camera-datasheet.pdf.
iii. https://flask.palletsprojects.com/en/2.0.x/
iv. https://docs.ros.org/en/foxy/index.html#
v. https://www.hivemq.com/public-mqtt-broker/
vi. https://flask.palletsprojects.com/en/2.0.x/