Remote Automation Control

[1]Rajath J, [2]Rithvik Kumar, [3]Chetan SR
[1,2,3]*Student, Department of Electronics and Communication Engineering*
[1,2,3]*PES University, Bangalore, Karnataka, India*
rajathjnx@gmail.com, rithvik1939@gmail.com, csrajenavar23@gmail.com

***Abstract*—There is dire need to create a safe and stable working environment for people in an industry. Due to the present pandemic. Workflow of many factories is cut by half. This significantly hampers the progress of the working sector and brings down our economy. We wanted to make a robust method which would be able to work with any machinery simple or complex. The method so proposed is not limited to only one type of industry. The concept behind the method is Internet of Robotic Things (IoRT) [7] [8] and is achieved by using the programming tool Node-RED [3].**

***Index Terms*—Remote control, Automation control, Node-RED**

I. Introduction

In the present situation of the pandemic, it is daunting to see industry labourers and workers risk their lives and work on the site. Their safety is a big concern in the working environment [1] [2].

However there are many new technologies which can work towards ensuring this security and IOT is one of them. Robotics has always been an important part of the industries, so our project aims towards combining the power of IOT and robotics to ensure safety in the industrial environment with minimal human interference - which is more popularly known as “Internet of Robot Things” [7]. IoRT [8] always helps a robot function better since it gives the robot a more clear picture of the situation by the sensors or “smart objects” which are external to it.

An important point in looking for the solution of such problems is that it has to satisfy all the industrial norms. It has to even provide security and prevent unauthorised access. Another aspect to be taken into consideration is the safety the project offers. It should have a fail-safe which stops all the present running jobs incase of any accidents. Last but not the least, to have a surveillance system integrated to our project. This further adds a layer of security to our program.

We wanted to build a program taking all this into consideration. At the end of our project we wanted to atleast setup a basic flow.

II. Methodology

We give an account of the present methods and our propsed method below.

A. Present Methods

[12] A lot of general case methods can be found online. Most methods work via MQTT [12]. They offers some layer of protection, but cannot be extended further. The support offered by the present methods to remotely controlling devices is miniscule. Most of the present methods lack end to end encryption which is important in the current setting. This will prevent unauthorised and accidental access. They lack a robust nature, most of them are hard coded because of which they are not versatile. The present methods offer multi device access which is feature incorporated in our project. At last is the perpetuality of the system. All the present methods will keep on working forever in an ideal environment, but most of them don’t have a any fail-safe methods incorporated in them. This makes them less reliable. Taking all this into account, We decided to use some features from the present method into our project [9].

B. Proposed Method

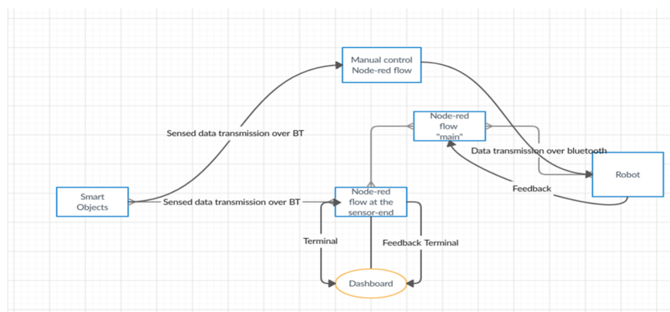

Fig. 1. Program Flow.

The project flow for our porposed method is given above. First we isolate the decision making part and working part from our flow.

The decision making part of our project consists of the Node-RED flows. We used Node-RED [3] [6] because its easy to use [4], flexible and offers both a flow access protection and a end to end user protection. The end to end user protection is achieved using the email node which works on IMAP. This automatically protects the instruction sent to the robots [10]. An internal node is setup to verify the email address from which the message. Node-RED is a visual programming tool in which each decision is performed using blocks and complex tasks can be programmed via the function blocks [4]. It has an active user base which povides support to almost all problems encountered. Most of the decisions in Node-RED can be done via its nodes. This helps to easily reduce any condition to decision making nodes [5]. We also employ a two state decision making model in our project. This helps us in making conditions [4]. Lastly we can deploy multiple decision making flows in Node-RED. A MQTT server can also be established using Node-RED. This helps in the control of several bots and smart objects. Communication can be established for n number of bots and sensors by just running a single flow but this time we need to define a MQTT server where the sensors publish the data to the server, and the robots recieve the data by subscribing to the server.

We made both a Manual control and an Automation control for our project this helps easy debugging, performing specific tasks and provides a final control to the user [9].

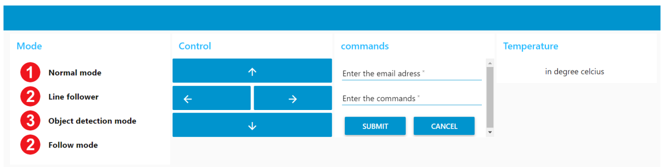

Fig. 2. Manual Control Flow

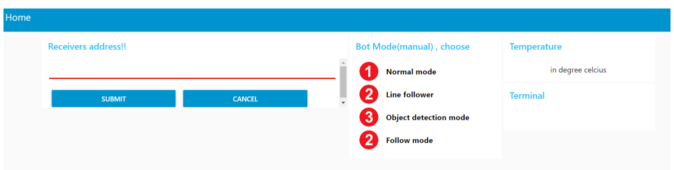

Fig. 3. Automation Control Flow

Node-RED supports multiple platforms. This means it can be configured in any device. Below is a picture of our Node-RED flow running on android.

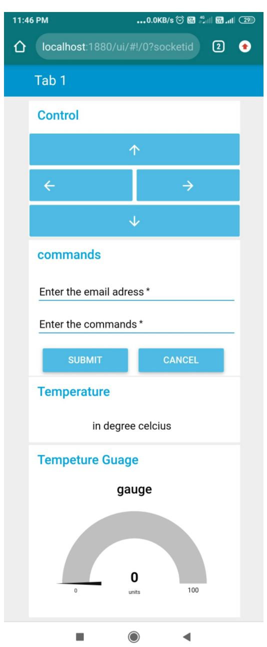

Fig. 4. Node red on Android.

The working part of our flow is mostly composed of the robots and smart-objects. These can be made using any hardware and can be anything. It can be a simple machine which turns on the lights in a factory when it detects an employee. It can be a surveillance bot which goes around the whole factory. It can be a complex bot, which oversees the working of other bots. All the devices in our working part have only one working requirement that it should have a part which is able to accept commands. This part can be anything a simple Bluetooth module or gsm module which communicates based on messages. It can also work with the MQTT server. All the bots should have an inbuilt command which maybe unique to it or all can have the same command to perform a common task. This gives absolute control to the user. He can setup the commands he wants. This command list can later be programmed in Node-RED and can also be programmed around a specific condition [11].

III. Workflow

We have built a working robot prototype and a sensor-machinery called "smart objects" which keeps on feeding the sensed data(in our case a temperature/infrared reading) to the robot [10]. The robot also gives a feedback of its operation to the smart objects. The robot is designed to operate in different modes based on the data received and act to ensure safety in case if the data received is depicting some danger. Node-RED 'flows' monitors the flow of sensory data from the smart objects to the actual bot. The same basic idea can be extended to any number of robots by just taking care that each robot will have its own working email-id and by subscribing to a MQTT server. The idea can be used in any industry and even the safety of the data is ensured since the user has to provide his username, password to login to Node-RED UI and also we use email as a means of sending the data.

A robot, a few smart objects and node-red flows are built. We used bluetooth as the mode of communication between the bots.

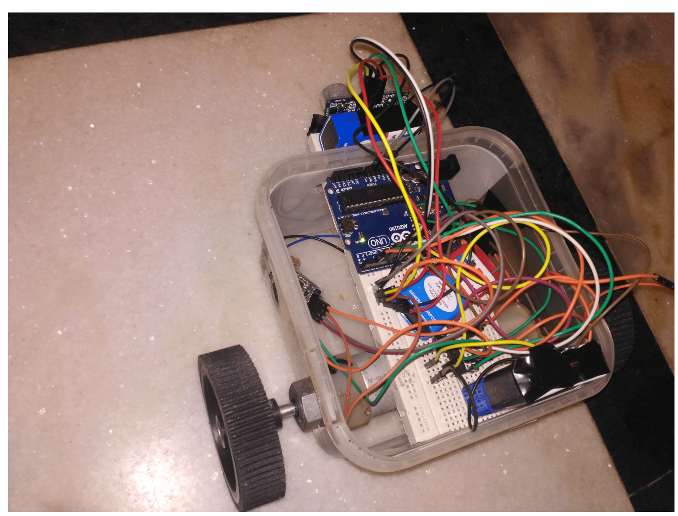

Fig. 5. Our Robot.

Above is a picture of our robot. We programmed it to perform basic tasks. We also programmed it to work in some predefined methods. We programmed an Object detection mode, Object follow mode and a Line following mode. By default it is set to work in the Normal mode. The ciruit diagram of our Robot is attached below.

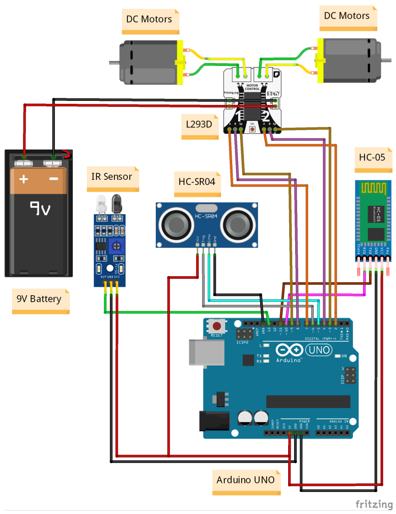

Fig. 6. Circuit diagram of our Robot.

Smart objects consists of temperature sensors, servo based door/lid objects and a infra-red sensor. These take in the readings from the surrounding and send the data they have gathered to the Node-RED server [11]. We attached all our smart objects to an Arduino UNO. This way we were able to calibrate between many smart objects.

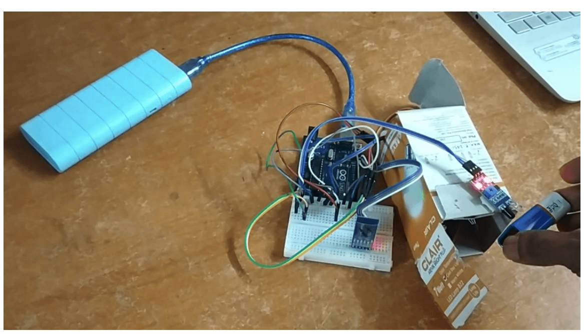

Fig. 7. Our Smart object.

In actuality the smart objects is nothing but a sensor which feeds in data, to the Node-RED [11]. This can be replaced by other devices also. We can also have two robots instead of a smart object. This is the flexibilty we wanted to achieve. This setup can be also mimiced in a production environment where each stage can be setup as a smart object. Each stage can also have many subflows within it. This can help to make each stage work independently, without the actions of other stages infuencing it. Thereby by reducing the complexity and allowing it to be controlled by a single command [5]. A good programming practise can further enhance the working of our program. The circuit diagram of our smart object is given below.

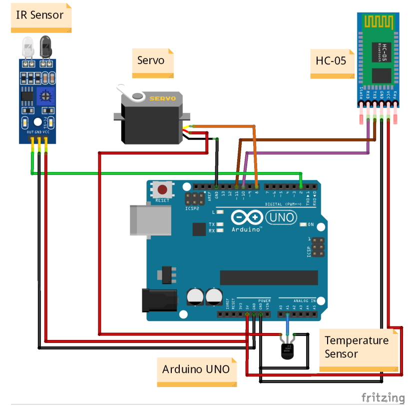

Fig. 8. Circuit diagram of our Smart-Object.

There can be any number of basic structures but we made the three must and important building blocks of the project. The temperature/infra-red readings and even the messages like 'the door is open/closed' is sent via Bluetooth to a Node-RED flow running on a computer/android phone in close proximity to the smart objects-end. The flow then transmits this information to another flow 'main' which is running on a computer in close proximity to the robots. Both the flows have a UI interface associated with them, where the user can enter the email-id to channelise the data flow between the required smart objects and the required robot. The robot also has a flow called 'main' which should be running on a device close to it, so that it could receive messages and readings from the flow via Bluetooth. The robot is designed to operate in different modes like the line-follower mode, object-detection mode and the normal mode.This mode is dictated by the data received by the flow. The robot sends a feedback about its operation from the flow "main" to the smart objects. The user can also manually control the robot from anywhere using a noed-red based UI called "manual control" on his phone or laptop. The basic approach of the design is by configuring Arduino with node-red. Node-red can be used as a tool for wiring together the hardware devices, API's and online services. In our case we use node-red to collect the data from the bot, process the data to a suitable form so that the information sent can be used by the other bot to perform specific operation. It is assumed that every robot and the smart objects involved is provided with a unique email-id using which a separate communication channel is allocated to each of them. A Dashboard is also been provided for the user interface to control the bot manually by selecting the mode of operation and can also be used to get sensor outputs, notification regarding the condition of the bot. The user can access the dashboard by just logging in through the specified email-id. If the user interface is not on, then the bots act as dynamic swarm by performing operation depending on the data received by communicating between them.

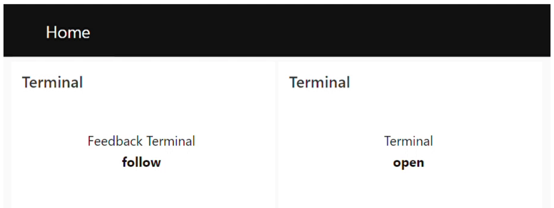

Fig. 9. Receiving data and getting the feedback from the bot.

IV. Results and Discussion

At the end of the project we were able to get two remote equipments i.e. the smart objects and the robot work together by exchanging data with the aid of node-red. This in-fact could work for any distance as 'mails' were used to relay the data. So the distance between bots for communication doesnt matter. The robot continuously kept receiving the temperature readings/other data from the smart objects and based on it, changed its mode. It remained in normal mode if the temperature range is at a safe range and went to line follower mode if it fell in the range of danger. This in a scenario of a real industry like a chemical industry, would mean the same normal robot functioning in a normal mode would also adapt itself for an emergency situation, like moving towards the chemical-container whose temperature falls in a risky range and solving the problem. The sensory-end continuously received a feedback from the robot regarding what it is doing. The same was also displayed on the Dashboard. We were also able to control the robot manually using a UI. A video feedback relayed by the robot helped the operator in controlling it. The project was made scalable to further include any number of bots, by defining a MQTT server where the sensory node has to publish to the server at the same time and number of bots can subscribe to the server, to get information of the action it needs to perform. The flows could in-fact run on any system android/computer.

V. Conclusion

A basic working prototype of our model was successfully made and it worked as expected. Further learning on how to completely use Node-RED can help enhance the working of our program. This method can work successfully if configured properly. The only problem with our persent model is the response time, which can be reduced by using better sensors in communication. It can also be reduced by setting up the response time for reading and sending the emails. This can be achieved by synchronising all the working flows. Our method can also work for heavy machinery but most of them need to be configured to process the data they receive. This is the only drawback which one may feel, As this works as configuring feature. But repeating the same configuration for n number of robots and smart objects can be a daunting task.

VI. Acknowledgment

We would like to express our gratitude to PESU IOT for providing us this project. Also would like to thank our Professor Charanraj B R. His supervision and instructions truly helped us, during the period of conducting the project. Overall, this project would not have been successful without our professors' and peer support.

References

[1] Covid 19 effect on Industries
[2] Factory employees tested positive
[3] nodered.org/docs/getting-started/windows
[4] Node RED Programming Guide
[5] IoT Services and Virtual Objects Management in Hyperconnected Things Network
[6] IoT Application Development with Node-RED
[7] The Internet of Robotic Things
[8] What Does the Future of IoT Robotics Look Like?
[9] IoT-aided robotics applications
[10] Robotic process automation
[11] I Dream of IoT/Chapter 4 : IoT and Cloud Computing
[12] Node-RED controlling Robot