iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47141

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Multi-level Architecture for a Swarm of Surveillance Rovers

Lakshya Jain[1], Upendra Bhanushali[2], Rajashree Daryapurkar[3]
[1, 2, 3]*Department of Electronics Engineering, KJ Somaiya College of Engineering*

Abstract: ***This paper depicts the design and implementation of a multi-level, multi-faceted software and hardware architecture that we have used to create a swarm of intelligent robots that can communicate basic instructions to each other. We have several layers in the architecture that can control a particular part of the system, with humans as the major drivers for the whole system. The individual rovers can navigate autonomously and avoid obstacles in their way. Our system utilizes several microcontrollers and companion computers such as Raspberry Pi, Pixhawk Flight Controller, and Arduino Mega Microcontroller. We have also done the circuit design and PCB implementation for the rovers. We have conducted this project as a proof of concept for the feasibility of Surveillance Robots. A swarm of intelligent robots would undoubtedly be helpful in situations such as disaster management and search and rescue missions.***
Keywords: Multi-level architecture, Swarm Intelligence, Robotics, Embedded Systems, PCB Design, System integration

I. INTRODUCTION

The main aim of our project is to design a swarm of cost-effective, reliable, and intelligent surveillance robots. The individual robots in our system are autonomous rovers equipped with a control/actuation system and can take footage of their surroundings for recording and processing functions. It follows a master-slave architecture where raspberry pi act as the master and Arduino is the subordinate. Arduino takes commands from raspberry pi, and most of the sensors are integrated into the Arduino, and a raspberry pi handles commands and image processing.

Surveillance and monitoring systems aim to produce high-quality, reliable data, giving measurement, reporting, and verification functions on captured data. One use-case of our surveillance system is monitoring forest fires, which can be detected early and precisely. Building a surveillance and monitoring system using stationary cameras for such areas can be challenging as the structures need to be designed considering the surroundings to be monitored. Also, if we use Surveillance Cameras, they can leave blind spots since they are fixed to the position. So, we propose a swarm of mobile autonomous surveillance robots that can effectively cover a large area.

A swarm of robots is a cluster that works in unison towards a particular objective. Such systems can be used for numerous systems such as light shows, disaster management, surveillance, mapping, and search and rescue missions.

II. LITERATURE REVIEW

Krauss[1] developed a cost-efficient autonomous vehicle platform that combines an Arduino microcontroller, Zumo, and Raspberry Pi-driven robot chassis. Raspberry pi provides a web interface, control for debugging, and wireless data streaming while also providing improved computing power. Arduino executes the control law in accurate time intervals. The reason to use Arduino and raspberry pi is to overcome the limitation of the Arduino related to streaming the data back. It is adding Raspberry Pi, which works as a mini-computer, to solve these issues by providing Wi-Fi via VNC or SSH. The synergistic combination of an Arduino and raspberry pi allows reading the sensor signals and gives the motors PWM signals accordingly in real-time execution by Arduino.

Rohith B N[2] uses the concepts of IoT and computer vision. This system is to be used for surveillance activity on large farms or forests. As we know Mobile Surveillance Camera gives much more flexibility than a fixed or static one, it is an optimal solution. The author gave brief information on the hardware that was used to develop, which is ESP32 with a sensor array. ESP32 is connected to Wi-Fi, and an antenna is an alternative for long-range communication. The author has also planned on implementing a fire alert system that depends on the readings from the temperature, humidity, and gas sensor. Readings are taken from the sensors, and a notification is sent in case a fire is detected. There is a video feed that can be viewed to catch false alarms. If an ultrasonic sensor detects any object, an obstacle alert is launched, which helps the robot to avoid obstacles. A microphone detects uncommon noises, and a notification is sent to the user if noises like cutting trees, vehicle passing, or other high decibels sounds are sensed. Thus the author has created a mobile surveillance system using IoT and computer vision.

Arabi et al. [3] have designed a GPS-Based Navigation system for robot path planning. A control station feeds the GPS coordinates of target points to the rover. With the help of a GPS signal, the rover creates a path to the target location. A PID control algorithm is used for a course correction of the position and direction of the robot to compensate for any deviation or error, follow the path, and reach the destination. A Raspberry Pi is used to drive the rover after making the necessary calculations. The features of this project are the calculations between destination and current locations and resolving any complications that may occur, such as path formation and localization. Haversine formula has a high degree of approximation, ensuring high accuracy, and they used longitude and latitude along with the formula for distance measurement.

G. Anandravisekar et al. [4] have proposed an IOT-based surveillance robot where they have interfaced a Wi-Fi module with Arduino to achieve an unlimited range of operations. The surveillance robot can be operated in both automatic and manual modes. The complexity and cost of this project are reduced by using the Arduino microcontroller, and the communication with the robot is carried out more securely, which has several advantages over the existing systems as they use robots with a limited range of communication and use short-range wireless cameras. Cayenne software is used in this project. This software is an object-relational mapping (ORM) framework. It allows us to work with objects abstracted from databases. It is used to design prototypes and IoT-based applications as it is a drag-and-drop project builder which allows devices to connect to the internet easily. Both manual and automatic modes can be performed with this software.

III. PROJECT DESIGN

This section covers the design and planning stages of our project. In our brainstorming sessions, we formulated a problem statement for the project, followed by the proposed solution that we would work on further. We have explained the working principles with a generalized block diagram.

A. Problem Statement

1) To make ground-based robot platforms that will be able to navigate autonomously in their designated regions.
2) These robots must be able to communicate, and a central device can be used to monitor, control and instruct the robots to work together.
3) The individual robots must have locally onboard processing, communication, and sensing capabilities for low-level control to react to their environments autonomously.
4) Individual robots should be able to act upon instructions sent to them by a higher authority system or device.

B. Proposed Solution

1) We start with a swarm of robots that may or may not be similar in terms of capabilities.
2) The swarm consists of one leader robot, which can take instructions from the ground station and broadcast them to the rest of the swarm.
3) The ground station will be able to receive the status of the swarm from the leader robot, and there will be an interface to send instructions to the swarm.
4) We will use MATLAB to create an application and communicate with the Raspberry Pi on the leader robot.
5) Each robot will have two levels of control – a low-level controller and a high-level controller.
6) A Raspberry Pi 3B+ is used as the higher-level controller in the individual robots. This device will be able to communicate with the raspberry pi in the leader robot and take instructions from it. It will also be able to relay these instructions to the Arduino.
7) An Arduino will be used as a low-level controller, running the state machine for switching states, taking inputs from sensors, and processing motors' control algorithms.
8) Each robot will have a Li-Po battery pack, power bank, and L298P motor driver shield to power the motors and the hardware. The sensor array will consist of ultrasonic sensors, a GPS module, an accelerometer, and a gyroscope.

IV. METHODOLOGY

A. Multi-Layered Architecture For Swarm Of Robots

Since we have a swarm of platforms that need to accomplish a task by coordinating, it is essential to have a definite structure and protocols to operate every component in the system. The architecture that we have proposed is as follows:

1) *Low-level Control Layer:* The lower-level control layer is responsible for the control of motors, error correction, predefined tasks or maneuvers we have planned, switching between states, taking in and processing inputs from the sensor array, and reading instructions from higher levels. The Arduino MEGA, the motor driver shield, and the sensor array come under the low-level control layer. The state machine, which runs on the Arduino, changes state based on commands from the higher-level control layer. The state machine is responsible for switching between different modes of operation on an individual robot. We have implemented this using State Machine Library in Arduino, which uses Linked Lists to traverse between different states. The robot can decide which function to activate and switch between different functions based on the input from the higher layer.

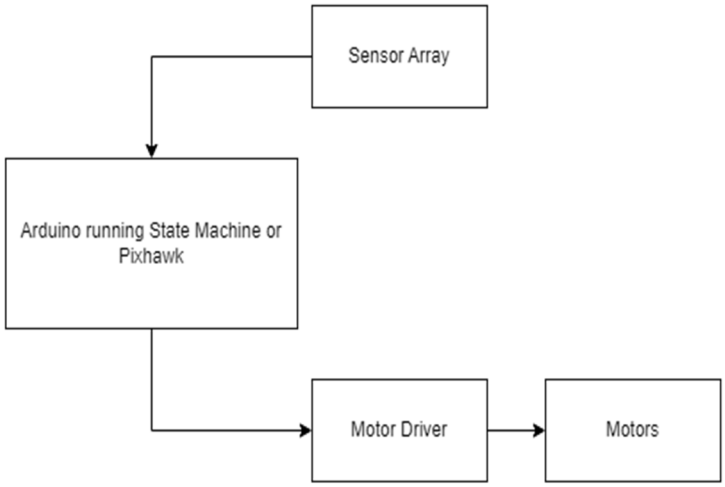


Fig. 1 Block Diagram showing Flow of signals in low-level control layer

2) *High Level Control Layer:* The high-level control layer is responsible for visual perception, generating instructions for the low-level control layer, communicating with higher layers, and taking instructions from them. The higher-level control layer consists of Raspberry Pi with a Pi Cam as the camera device and a zig-bee antenna for communication. The relationship between low-level and high-level control layers can be described as a unidirectional master-slave system in which Raspberry Pi can communicate with the Arduino but not the other way around. To establish this method of communication, we connected four GPIO output pins from the RPi to the Arduino digital input pins. This way, we can send a 4-bit binary signal by a very simplified approach, and we can send up to 16 individual instructions.

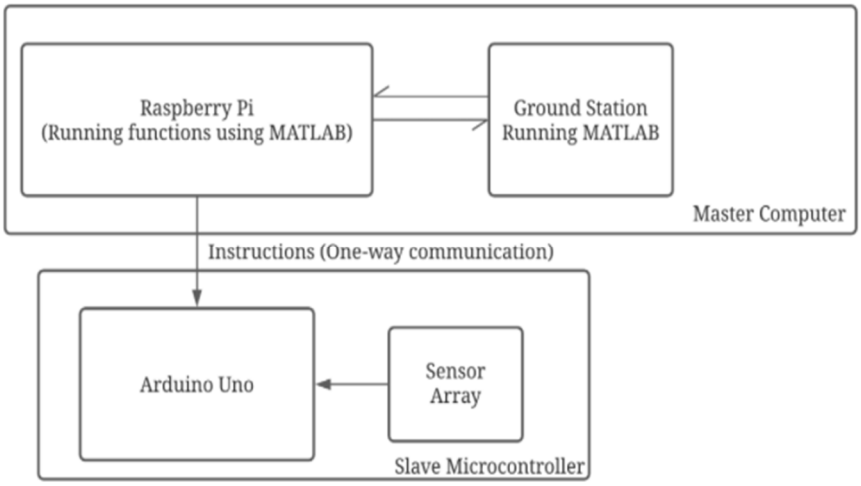


Fig. 1 Block Diagram showing the interaction between low-level and high-level control layer

Decimal Input to Arduino Digital Pins	Command
0	Stop
1	Go Left
2	Go Right
3	Go Reverse
4	Go Straight
5	Traveller Mode
6	Scanning Mode
7	Follow Me Mode
8	Fire Fighter Mode
9	Return to Home
10	Manual
11	Rest
12	Obstacle Avoidance

Fig. 3 List of instructions for communication between Raspberry Pi and Arduino.

3) *Communication Layer:* The communication layer is responsible for communication between the ground station, leader robot, and little robots. Depending on the mission profile, this communication layer is switchable between Wi-Fi, ad-hoc network, or ZigBee-based communication. The leader robot is responsible for all interactions with the ground station, relaying communications to minor robots and coordinating between minor robots. The raspberry pi on the leader robot can broadcast instructions to all minor robots in its group.
4) *Human Interaction Layer:* The human interaction layer consists of the ground station and the interface present on the ground station to communicate with leader robots and receive any communication from them. In our current implementation, a laptop running an app designed using MATLAB is being used as the ground station. Using this app, we can seamlessly navigate different functions and send and receive communication from the leader robots. This layer acts as a point of contact between the human (the operator) and the rest of the system.

B. Block Diagram of Entire System

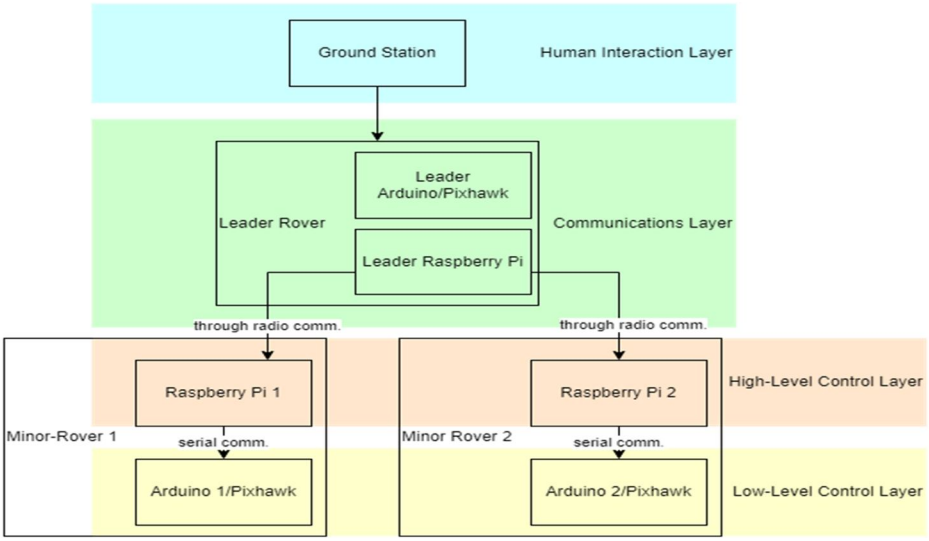


Fig. 4 Block Diagram showing the layers in the architecture and the hardware components in play

C. *Hardware Used*

TABLE I
SENSOR ARRAY

Hardware name	Description
U-blox Neo 6M GPS module	GPS module will be used to track the robots' position and navigation
MPU-9250 accelerometer/ gyro	Accelerometer/gyro will be used to get heading information
SR04 ultrasonic sensor	Ultrasonic sensors are used to map close by objects and for obstacle avoidance
DHT11 temperature and humidity sensor	DHT11 will be used to get readings for the temperature and humidity.
Pi-Cam (5mp)	Pi cam is our primary camera which will be used to take input for all the computer vision and identification functions

D. PCB Design and Implementation

Since we are using several sensors and devices, it was essential to develop a PCB to make the system streamlined and reliable. For this purpose, we designed a double-sided PCB that fits onto the Arduino as a shield, and this PCB can house/provide connections to all of our components.

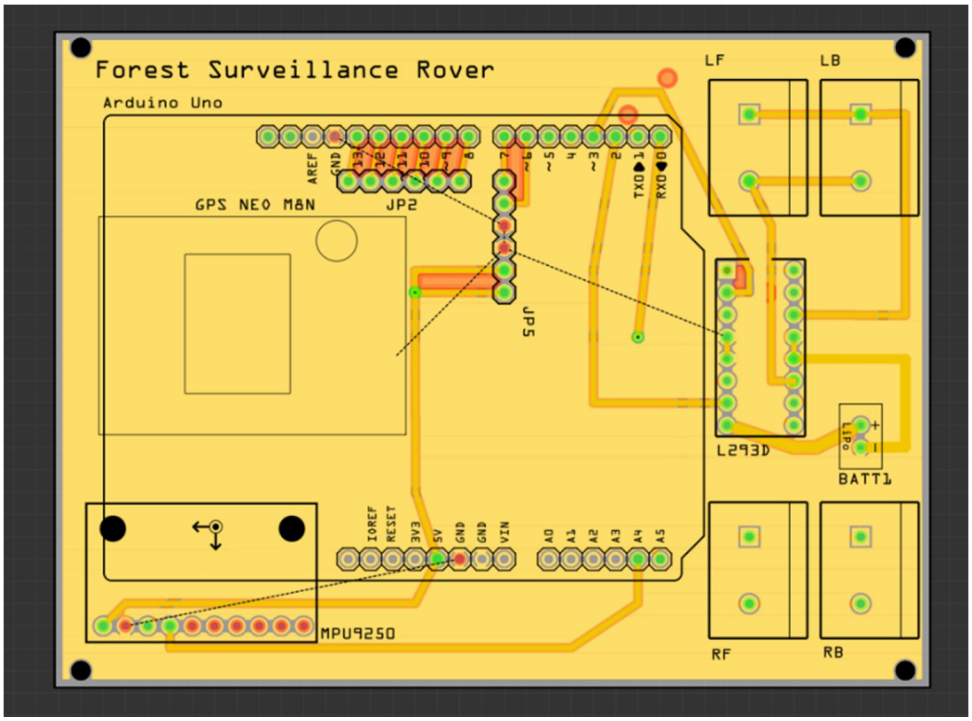


Fig. 5 Custom PCB Layout for Surveillance Rover

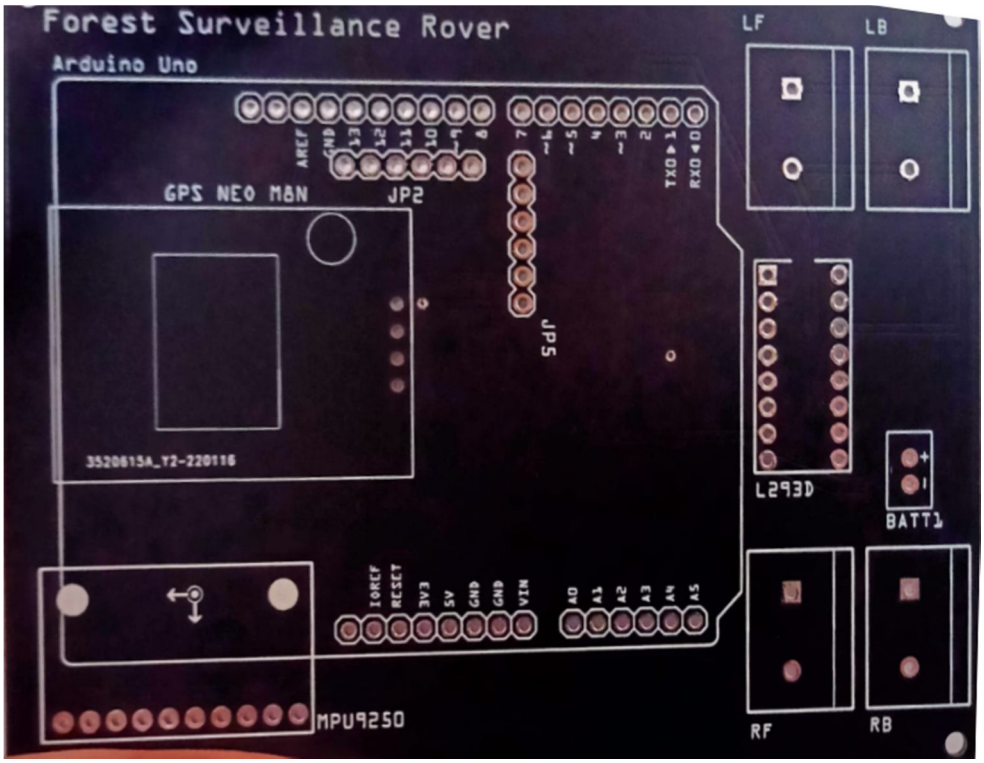


Fig. 6 PCB Layout for Surveillance Rover

E. State Machine Implementation

When a command is sent from Raspberry-pi to Arduino, the Arduino will change its state based on the entry and exit conditions of the states as prescribed in the program. The transition condition used to transition between states is as follows.

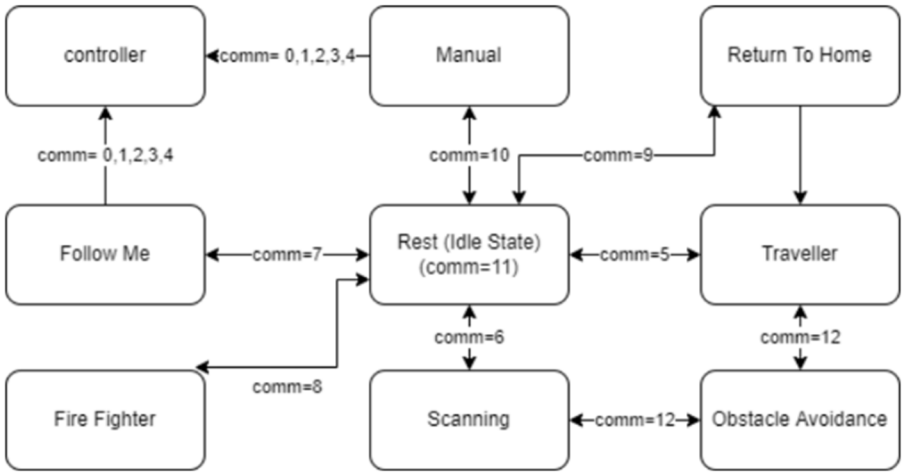


Fig. 7 State Machine Algorithm implemented on Arduino Mega

In figure 7, the rest state is the default/idle state of the robot; the traveler enables autonomous waypoint navigations, scanning feature executes a preplanned maneuver. The entry and exit conditions of every state are according to figure 7.

F. MATLAB app GUI

The MATLAB app we designed allows us to switch between different image modes and robot states. It makes it very easy for the user to send commands to the rover(s) and display image data. We can also control the robot manually using the app.

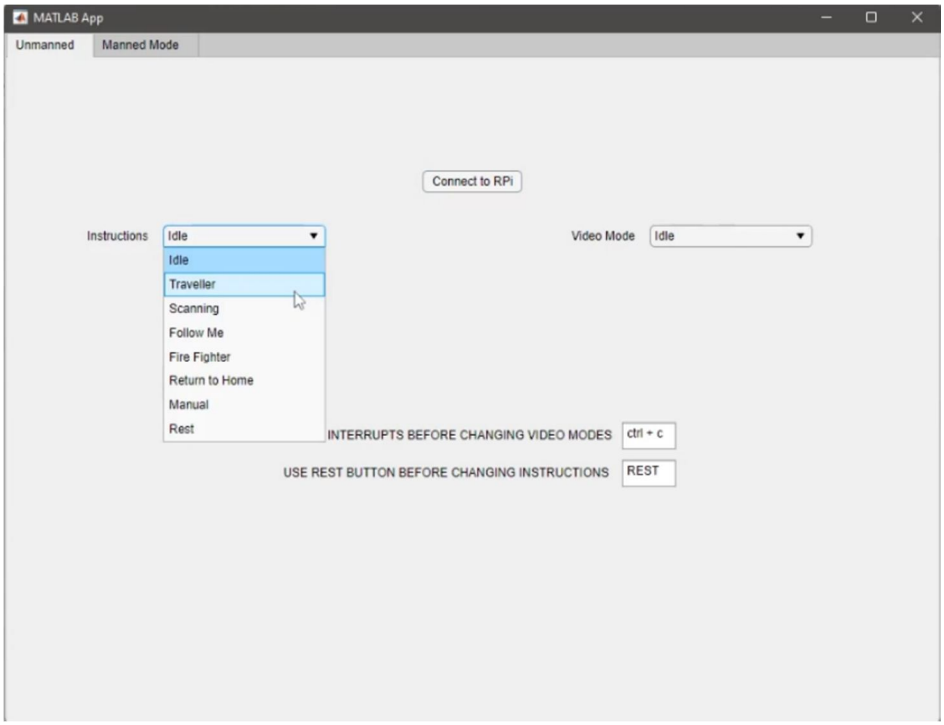


Fig. 8 Unmanned Commands on MATLAB GUI App

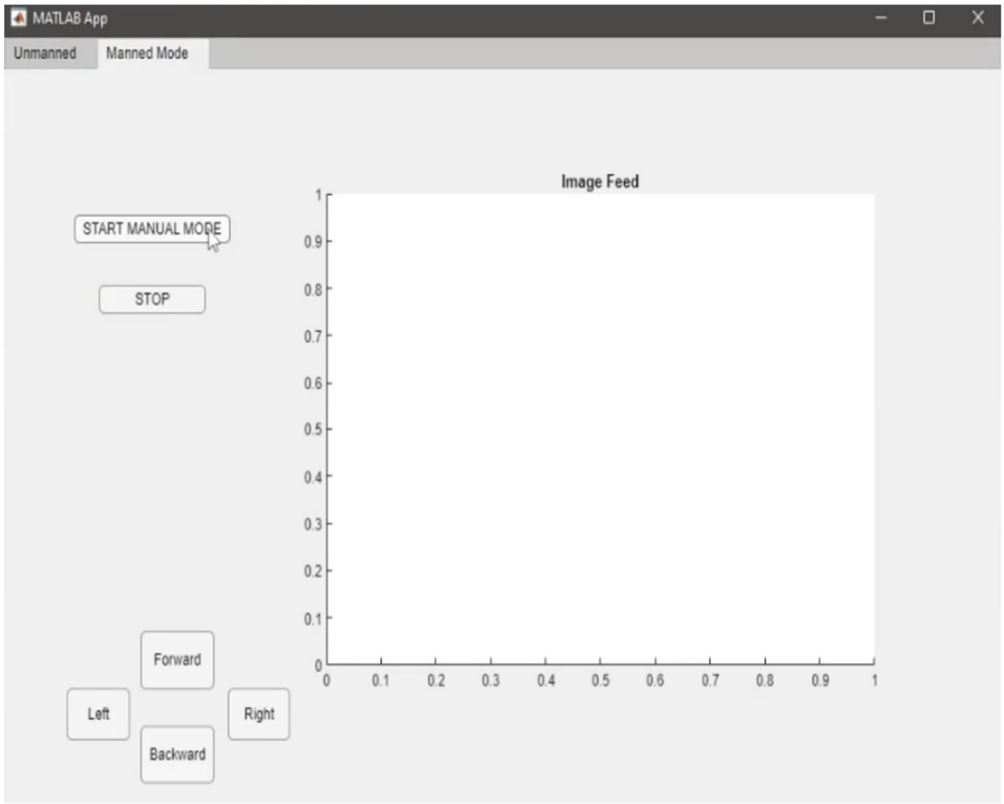


Fig. 9 Manned input commands on MATLAB GUI App

G. Prototyping and Practical Implementation

After testing out all the individual components and systems, we started integrating them into our prototype, combining hardware and software architectures and testing them. In the hardware department, we started with placing and mounting all our components in the chassis. Our chassis has two levels capable of housing components, so we placed our components as shown below.

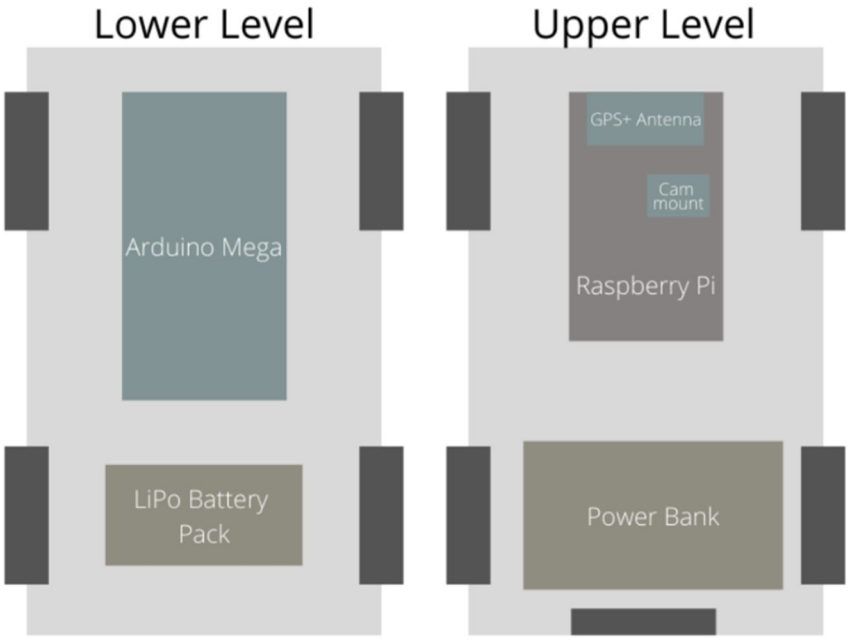


Fig. 10 Component Placement for Hardware in Rover

When the system integration and debugging were done, we tested out all of our proposed functionalities in the individual robot prototype. After this, we tested the system with the communication layer by sending commands and checking if the lower layer is receiving them and can execute them.

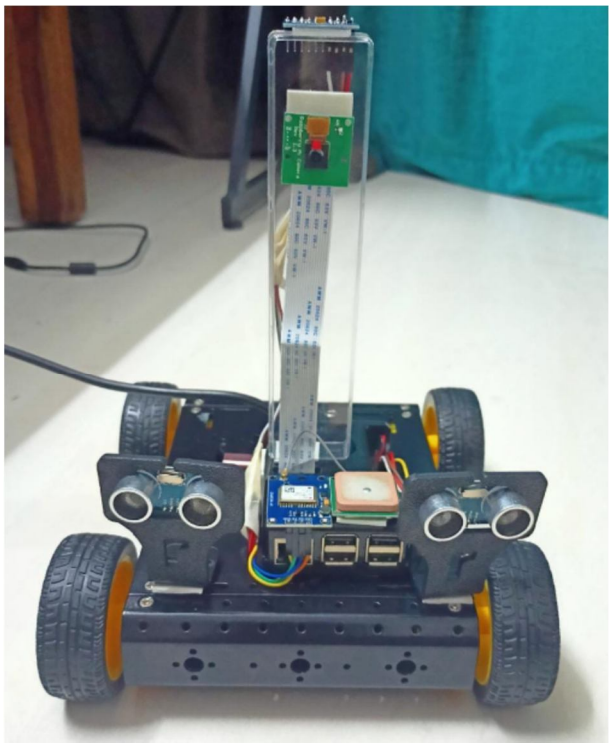


Fig. 11 Initial Prototype of an individual rover

H. Simulation Setup with Pixhawk

Once we had the proof of concept that the system we implemented could work, we decided to simulate autonomous missions with our robots. For this, we used the Ardupilot Mission Planner Software in SITL Mode and simulated the test in an area that might need surveillance.

We will use a Pixhawk Flight Controller Board in place of the Arduino to run the ArduRover Firmware. So, a Pixhawk can be used as an alternative to Arduino for low-level control.

We used the ArduRover Firmware to simulate rovers, but we can include drones and gliders in this swarm.

1) Swarm Autonomous Movement Setup

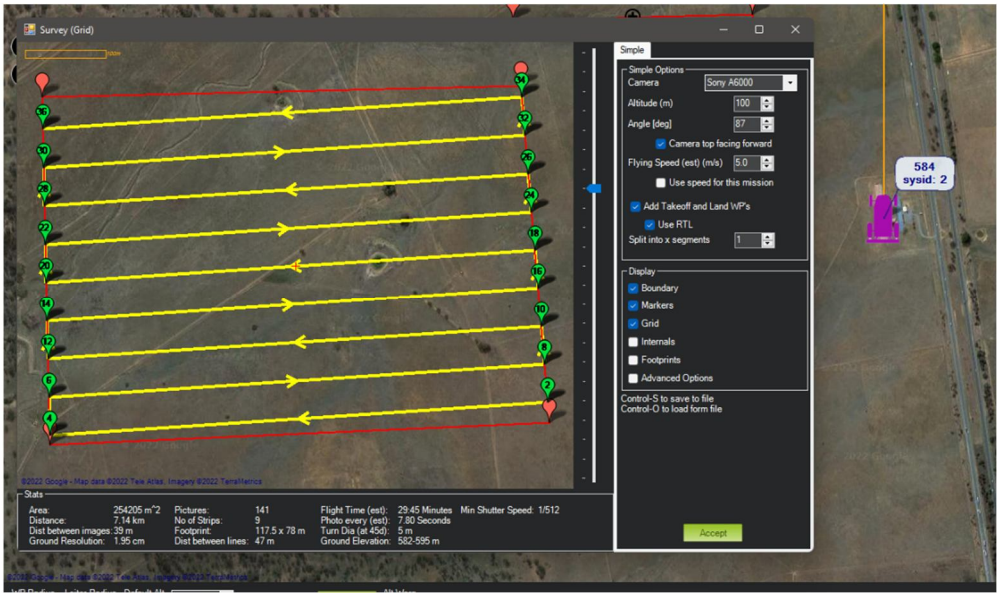


Fig. 12 Autonomous swarm movement setup with telemetry

2) Swarm of Robots Performing tasks Simultaneously:

As we can see in the figures below, the two robots are working on surveillance of two different areas simultaneously. These instructions can be given from the Ground Station to our rovers easily using the 'Plan' functionality.

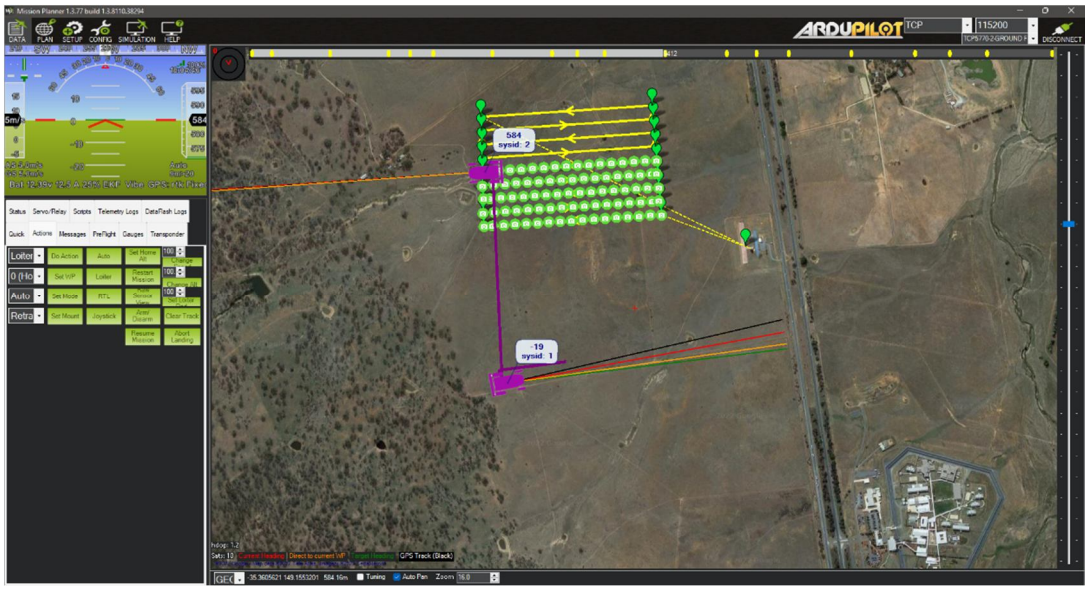


Fig. 13 Waypoint-based navigation for rover number 2 (scanning pattern)

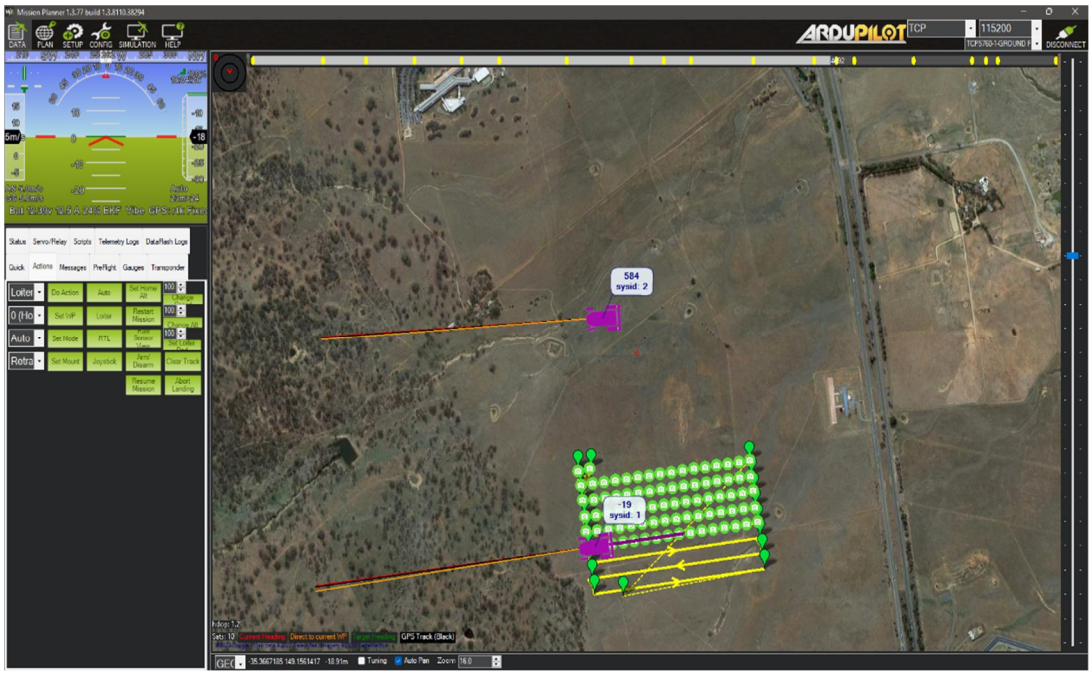


Fig. 14 Waypoint-based navigation for rover number 1 (scanning pattern)

V. FUTURE SCOPE

This swarm of robots can be used for various applications, such as

1) Surveying
2) Mining Monitoring
3) Construction Monitoring
4) Agriculture
5) Photography and Videography/Film Making
6) Public Safety
7) Disaster Management
8) Inspection
9) Telecommunications
10) Deep-Sea/Forest (Remote) Exploration
11) Light Shows

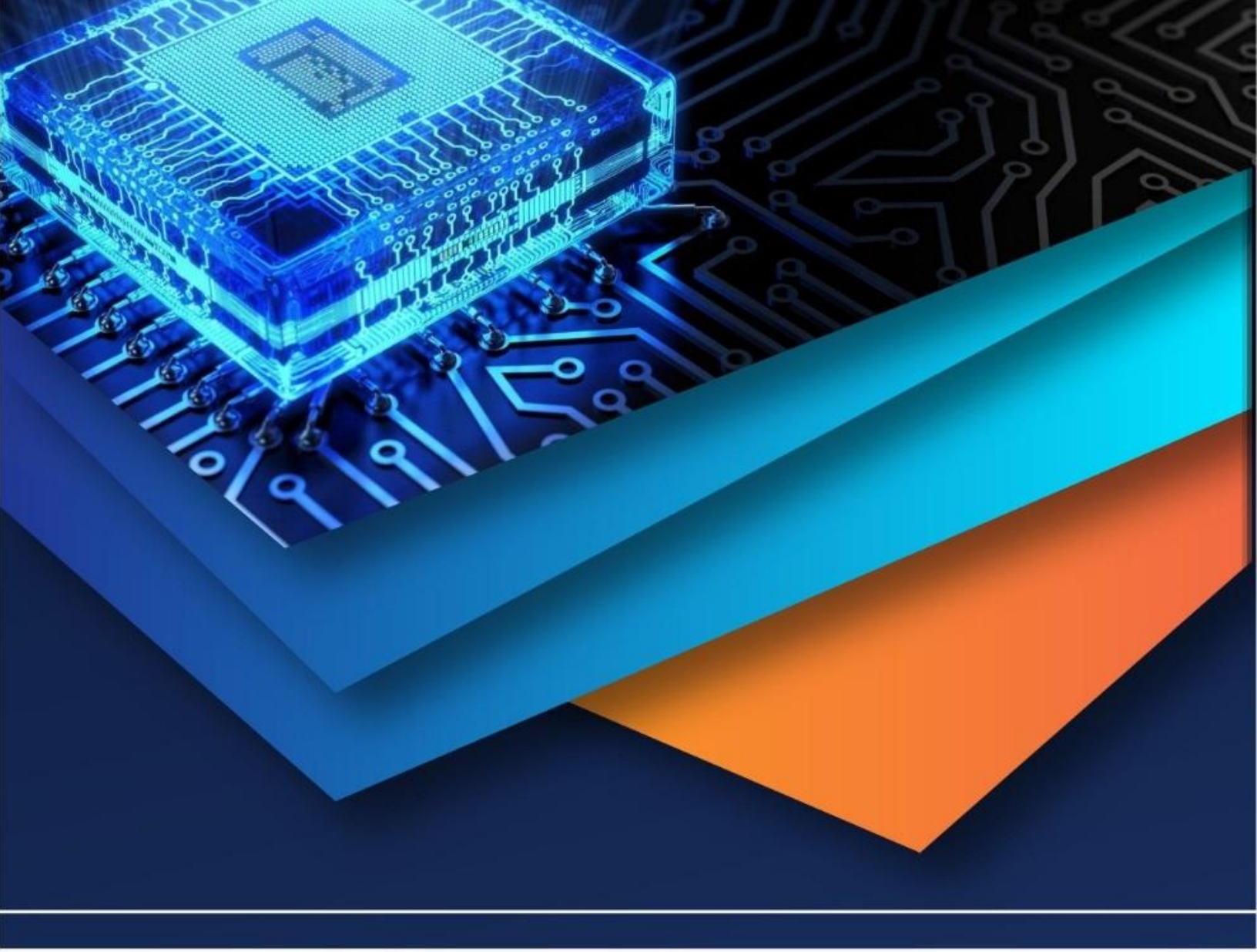
These robots can use swarm architecture and communication methods to actively communicate with each other and create a network of robots that can function autonomously and with human guidance.

VI. CONCLUSIONS

A. Thus, we have created a small-scale project to be used as swarm surveillance and intelligence solution. This project uses a swarm of Raspberry Pi and Arduino-powered robots for surveillance.
B. This system illustrates a low-cost robotic surveillance solution that can be made using components found at most electronics stores.
C. An automated surveillance system gives us convenience and a higher field of view and makes the surveillance task autonomous.
D. We have implemented a multi-layered architecture in the form of the master-slave system helps the system to divide complex tasks into practical resources and helps us optimize our system to get the best performance out of it without overloading it.
E. We also created a simulation to test our system using ArduRover Firmware on a Pixhawk Flight Controller Board and to make two rovers survey two different areas autonomously and simultaneously.

REFERENCES

[1] R. Krauss," Combining Raspberry Pi and Arduino to form a low-cost, real-time autonomous vehicle platform," 2016 American Control Conference ACC), 2016, pp. 6628-6633, doi: 10.1109/ACC.2016.7526714.
[2] B. N. Rohith," Computer Vision and IoT Enabled Bot for Surveillance and Monitoring of Forest and Large Farms," 2021 2nd International Conference for Emerging Technology (INCET), 2021, pp. 1-8, doi: 10.1109/IN-CET51464.2021.9456180.
[3] A. Al Arabi, H. Ul Sakib, P. Sarkar, T. P. Proma, J. Anowar and M. A. Amin," Autonomous Rover Navigation Using GPS Based Path Planning," 2017 Asia Modelling Symposium (AMS), 2017, pp. 89-94, doi: 10.1109/AMS.2017.22.
[4] Anandravisekar, G., A. Anto Clinton, T. Mukesh Raj, L. Naveen, and M. Mahendran." IOT based surveillance robot." International Journal of Engineering Research Technology (IJERT) 7, no. 03 (2018): 84-87.
[5] B. N. Rao, R. Sudheer, M. A. Sadhanala, V. Tibirisettti and S. Muggulla," Movable Surveillance Camera using IoT and Raspberry Pi," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225491.
[6] Sakali, Nagaraju, and G. Nagendra," Design and implementation of Web Surveillance Robot for Video Monitoring and Motion Detection." International Journal of Engineering Science and Computing (2017): 4298-4301.
[7] R. Washington, K. Golden, J. Bresina, D. E. Smith, C. Anderson and T. Smith," Autonomous rovers for Mars exploration," 1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403), 1999, pp. 237-251 vol.1, doi: 10.1109/AERO.1999.794236



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY