

Multi Robot Exploration and Navigation using Wireless Communication

Mudasar Basha¹, Seetha Ramakrishna M², Lakshmi Vaishnavi K³ and Tukaram K⁴

¹⁻⁴B V Raju Institute of Technology, Narsapur

Email: mudasar4u@gmail.com, seetharamakrishna239@gmail.com, vaishnavi.karri6026@gmail.com, kotaritukaram8686@gmail.com

Abstract—Autonomous service robots are one of the most demanding and emerging fields of robots with the advancement of technology in real-time scenarios. This paper proposes wireless communication-based autonomous navigation of robots under the direction of a master robot. The interaction between the robots is accomplished by the use of a Bluetooth module, as well as the master robot's self-exploration and self-navigation mechanisms. At every instant of time, the slave robots check the position of the master and travel in accordance with the direction of the master robot. Ultrasonic sensors, FPGA and Stepper motors were used to create the robots. Verilog is being used to create the design, which is then deployed on a Field Programmable Gate Arrays (FPGA), Zedboard Zynq®-7000 SoC for robot control. It has been successfully implemented using the Xilinx Vivado 2017.4 simulator. This design can be utilized for shifting commodities as well as certain domestic uses.

Index Terms— Bluetooth, FPGA, UART, Ultrasonic sensor.

I. INTRODUCTION

With of evaluation technology, Autonomous service navigation robots have become one the most challenging and burgeoning fields in the context of Human Assistive Machines which can serve the human kind assist them in their daily lives to fulfil their needs [5]. In order to serve human kind, a robot must become self-aware, clever enough and behave exactly like a person in order to effectively and efficiently aid elderly or disabled people. Single robots are frequently equipped with several sensors, which necessitate a complicated mechanism and a sophisticated intelligent control system. Although a single robot can perform well some jobs, such as spatially independent activities, are inherently too complicated or perhaps difficult for it to do. Multi robots have the potential to improve overall performance. Robots with diverse skills may be paired to handle complicated tasks, and if one or more robots fail, the task will still be performed. The spatial dispersion of Multi robots is superior. Multi robots may also carry out multiple tasks simultaneously [3]. Using the Bluetooth communication channel, connection between the robots will be established [13], and the essential information will be transmitted between the robots [16].

Finding the optimum route, either for the shortest path or for spending less time to perform a task is important. As a result, the environment was segmented into hexagonal cells, and the robots were given the task to explore and collect the information of the environment [8]. A* algorithm is utilized to enhance the robot's mobility by allowing it to identify the best path by avoiding confined paths and increasing efficiency [4]. The architecture which was designed and implemented in FPGA, which works in the specified environment [2].

The work space is vital for a robotic application. As the environment is divided into hexagonal cells, a robot must know the information of each and every cell to develop self-exploration. The worst-case scenario is also addressed while determining the best approach. The robot's navigation is self-contained, and it is attained by environment mapping, path planning, exploration and decision making. The navigation is developed in such a way that the robot can identify the optimal approach to reach its goal based on the map of the environment and obstacles in the work space.

Our overall goal is to build multirobot with one master robot and three slave robots. Through the unique function of the Bluetooth module PICONET, the slave robot's Bluetooth is paired with the master robot. With the help of Bluetooth module, the three slave robots communicate with the master robot one after the other. An environment with both static and dynamic obstacles was developed. All robots explore the area, record the data every cell in the environment using different algorithms. Our environment suits A* algorithm which makes the shortest path with minimal cost. This data stores in every robot, but only master will command the slave robots accordingly at the time of navigation. The robots navigate using ultrasonic sensors data, slave robot's and master robot's positions. The data is sent at a 9600 bits per second. As the Bluetooth parameters are adjusted by the AT commands using Verilog, the FPGA handles the connection and disconnections tasks in between the master robot's Bluetooth and the slave robot's Bluetooth. The AT commands are used to make feasible Bluetooth connection through FPGA. Here we are using Zed board (FPGA board) where the data of the obstacles is detected by the HC-SR05 (Ultrasonic sensor) and movement of the robots are taken place with the help of NEMA 23 (Stepper Motors) and finally HC-05 is used for communication between the robots.

II. RELATED WORK

Algorithms, Path Planning, Navigation and Bluetooth communication are all tied to the multi robot idea. Chunyu Ju et al [1] have mentioned that the best optimal path is generated by improved A* algorithm than A* algorithm which are resulted at simulations level. Chih-Min Yu et al [6] have proposed that the Hybrid Mesh-Ring algorithm is effective for Bluetooth networks. Piconet feature is utilized in this paper for communication between multi robots through Bluetooth. Guner Tatar et al [9] have mentioned that a land vehicle will be controlled by Bluetooth using an android application in a cell phone and limited to one robot and decisions are controlled by human. In this regard our paper's approach is to make multi robots to take self-decisions, Bluetooth communication using piconet and implementation to be useful in industrial application.

III. ENVIRONMENTAL MODEL

The environment is organised into hexagonal clusters with static and dynamic obstacles. The environment is a hexagonal- shaped two-dimensional map with a 10*10 matrix cell size. The environment is separated into discrete cells of uniform length, each with dimensions of 1 meter between flats Equation (1) where F indicates the distance between the flats, 0.577 meters between edges Equation (2) where A indicates the length of the side and 1.154 meter between corners Equation (3) where the D indicates the distance between the two diagonals. The node where the robot explores and navigates the environment is reflected as the block midpoints.

$$F = 1.732 * A \quad (1)$$

$$A = 0.577 * F \quad (2)$$

$$D = 2 * A \quad (3)$$

The source (0,0) of the environment will be used as the starting node, and the goal node will be used as the ending block (9,9). For the certain excursions, the obstacles in the environment are chosen at random and are static in nature. The robot will determine the obstacles and record the information in the data frame format. Dynamic obstacles are included in the same exploration as well as static obstacles, and information about the dynamic obstacles is also stored in a data frame format.

Every cell is examined and when an obstacle is detected in a cell, an ultrasonic sensor measures the obstacle's behaviour and distance. This information will be recorded for subsequent navigations. This exercise will be done several times so that the robot may better evaluate the information of the static obstructions and the action of the dynamic obstacles, allowing the robot to get more comfortable with the environment. This exploration is done for different environments to validate information about the cells.

IV. PATH PLANNING ALGORITHM

Here we are using all four robots to explore the environment using different algorithms. We consider BFS, DFS, D*, A* algorithms in each robot to explore the total area. After getting the results, we concluded that our environment suits A* algorithm. Our goal is to find quickest route through the environment. Using exploration data of dynamic and static hurdles the A* algorithm is used to find the shortest distance among all other algorithms. Quickest way to the destination is chosen by the robots and information is passed between robots through Bluetooth. We were using the A* algorithm in each scenario to determine the shortest path therefore for the navigation of the robots that will take to complete the various tasks. It has the following formula in Equation (4):

$$F(n) = G(n) + H(n) \quad (4)$$

The projected cost of the shortest path from source to destination is represented by the formula $F(n)$. There will be distinct $F(n)$ for each cell, as well as different approaches to obtaining the path to the endpoint from the origin. We assess the costs of each cell and the cheapest option for mobility. The cost of getting to node “n” from the originating cluster is $G(n)$. Here, we find the best path that reduces the overall cost of $G(n)$, which lowers the overall cost. The cost from node “n” to the target point is $H(n)$. The robots drive less distance when they locate the quickest way from their current location to their destination.

The training data are allocated to each cell as weights according to the A* method and the shortest path between two end points is obtained using the aforementioned formula. As illustrated in the diagram, the optimal route is designated with coloured blocks.

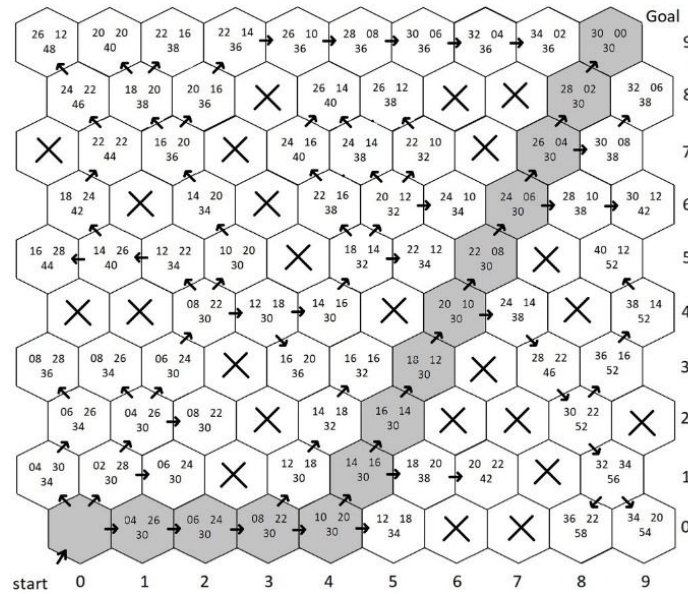


Figure 1. Shortest path from start (0,0) to goal (9,9) using A* algorithm by avoiding obstacles

Figure 1 depicts the environment as being split into hexagonal structure. Each cell's total cost and other factors are unique. Using the statistical information of the cell, we may move the robot in the shortest way possible, resulting in the best possible navigational path. The whole cost to acquire it is 30 units. It will vary depending on the location as well as any impediments that may be there.

V. BLUETOOTH AT COMMANDS

The Bluetooth AT commands function is the module that stores and transmits all of the command's data in response to the input's request. The directives are stored in the register and then transferred as byte data to the Bluetooth module through UART protocol. The Bluetooth AT commands module sends this data to the Bluetooth device one by one. For data validation, the received acknowledgment will be entered in the registers. The HC-05 Bluetooth module is intended for wireless data transmission. This component may be used as either a master or a slave. The message is sent at various baud rates. For interaction between both the vehicles, we set the

Bluetooth baud rate to 9600 bits per seconds including one start bit initially then one stop bit and no parity bit. The Rx pin of the Bluetooth is used to retrieve the information. It will gather the information as a start bit initially then 8 bits of data and then finally one stop bit with the same frequency as the transmitted data frequency. The messages will be received bit by bit. The last pin is named as STATE which performs the function whether the module is connected or not.

Traditionally, any embedded board has been used to modify the Bluetooth specifications. Additional boards are necessary here, and changing the Bluetooth configurations takes too long. Bluetooth AT commands are designed in Verilog to modify Bluetooth parameters and send AT commands through the FPGA to address this issue. Each AT instruction will be performed one after the other, after the acknowledgement has been received. As a result, the executed commands will be delivered to the FPGA one by one via UART. This will be more beneficial when the robots are navigating and creating connections and disconnections between multiple robots. When the master desires, it executes the functionality and establishes the necessary connections. This function is called PICONET which is one of the features of Bluetooth. The lengths of the most often used AT commands are shown below.

TABLE I. AT COMMANDS OF HC-05

S. No	AT Command	Hexadecimal value	Command Length*
1	AT	41 54	2
2	AT+RESET	41 54 2B 52 45 53 45 54	8
3	AT+ADDR?	41 54 2B 41 44 44 52 3F	8
4	AT+ROLE?	41 54 2B 52 4F 4C 45 3F	8
5	AT+UART?	41 54 2B 55 41 52 54 3F	8
6	AT+BIND?	41 54 2B 42 49 4E 44 3F	8
7	AT+CMODE?	41 54 2B 43 4D 4F 44 45 3F	9
8	AT+DISC	41 54 2B 44 49 53 43	7

*The Command Length is measured in Bytes and main commands are mentioned here which are mostly used.

The piconet is the basic organizational unit in the Bluetooth design. A piconet is made up of one master Bluetooth device and up to seven active slave devices that are within 10 meters of the master. All interaction is between the master and one slave, and the master alternates between slaves instantaneously, hence time division multiplexing is needed. Bluetooth is unique among networking technologies in that it defines the applications for which it may be used. It achieves it by defining profiles, which define the application as well as the subset of Bluetooth characteristics required to implement it. One vision is to build a multi-robot system by attaching three slave robots to a master robot and coupling them one by one with the master robot. The piconet versatility made it feasible to collaborate with several robots over Bluetooth.

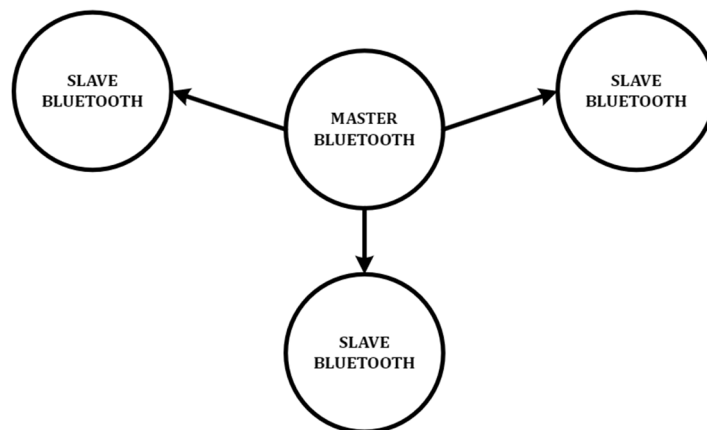


Figure 2. Piconet network with single Master and maximum connection of 3 Slaves

VI. IMPLEMENTATION

Each robot will explore the area utilizing search algorithms such as DFS, BFS, D*, A* during the exploration phase. Four robots are taken and placed in the environment and allowed to increase knowledge and understanding to see which algorithm suits best in order to acquire the optimal route. Robot-1 follows the BFS approach, Robot-2 is permitted to follow the DFS principle, Robot -4 follows the D* algorithm and Robot-4

follows the A* algorithm to save the data of each and every cell and completes the 10*10 cell environment exploration.

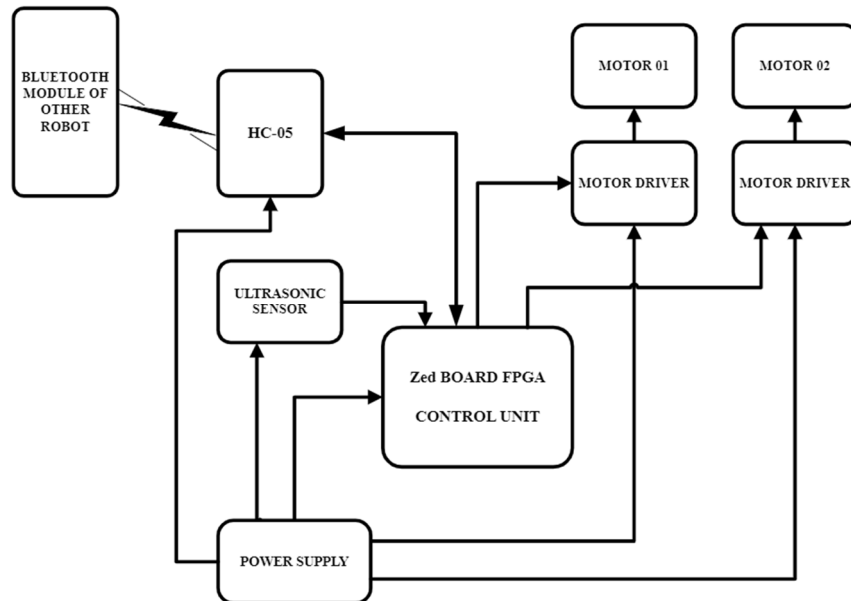


Figure 3. Block diagram of the Master/ Slave Robot

When compared to other techniques, the A* algorithm consumes less energy, takes less time required to reach the objective point, and covers fewer cells to achieve the ideal path. As the robot-4's most optimum path, the A* algorithm is picked and matches the terrain optimal. Now that each robot has been programmed using the A* algorithm, the master robot M will state supervision to slave robots 1 and 2. Slave robot-3 completes each task independently. As a result, two distinct jobs can be executed.

TABLE II. COMPARISON BETWEEN ALGORITHM

	BFS	DFS	D*	A*
Cost	96	82	53	30
Time taken^	44	42	27	12
Power^	90%	80%	40%	25%
Rotations	48	34	15	2

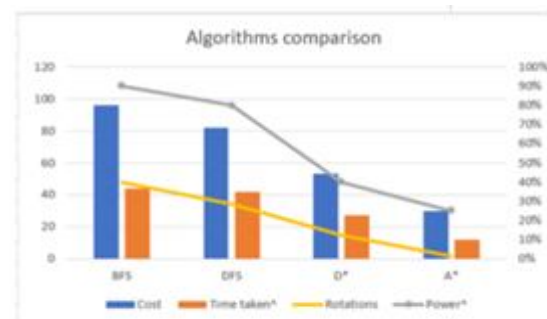


Figure 4. Graph showing comparison between the Algorithms

Here the time taken and power consumption of the different algorithms are taken approximately. Time considered in minutes and the powers are taken in percentage according to the consumption.

The master robot will negotiate the presence of static and dynamic barriers as well as the best way to reach the destination location. Master robot M is positioned at node-0 (0,0), moves forward in the best way possible and uses an ultrasonic sensor to detect any dynamic obstacles in its path. The master will allow the slave robot-1 to move if there is no dynamic impediment. Slave robot-1 is also positioned at node-0 (0,0) and moves once the master robot has reached the next node. The Slave robot-2 is likewise positioned at node-0 (0,0) and moves after the slave robot-1 confirms that it is safe to go forward. This process repeats until all of the robots have arrived at their destination.

If master robot M notices a dynamic impediment, it waits for the transient obstacle to move before following its right course of action. If the slave robot detects dynamic obstacles between the master and itself (slave robot), the slave robot will immediately notify the master robot that dynamic obstacles have been monitored and the master robot will instruct the slave robot to wait (for how long) or deviate to a substitute optimal line to prevent congestion. Communication between master and slave robots will be possible in this manner.

All robots maintain exploratory and navigation information, but that content may be retrieved by any robot if and only if the robot status is master. If the master robot is encircled in all aspects by five hurdles, which can be happens with a dynamic obstacle, the master robot will shift responsibilities to slave and some of the other slave robot's position to master. The new master can now have access to sophisticated pathway and it will direct all other slaves to their objective.

The robot-4 will complete its assigned task using the A* algorithm and will determine the best way to reach the destination on its own. The robot-4 identifies the both static and dynamic impediments and moves ahead in such a manner that it waits patiently for a dynamic obstacle to avoid colliding with it.

Multi Robots may be used in any field to reduce the production costs. The proposed work in this research may be used in an indoor environment where robots can carry loads from source to destination without having to operate manually. This application may be built in which the robots shift the commodities from one spot to another. This work will be valuable in the online order service industry's products warehouses.

The position of the cell and the orientation of the robot in that cell are communicated to the navigation during the excursion. As a result, after the robot is turned, the acknowledgement is given to navigation, which leads the robot towards the next cell and places it in the appropriate location. The navigation provides acknowledgement to exploration when this operation is accomplished. The following cell's data frame is sent by the exploration.

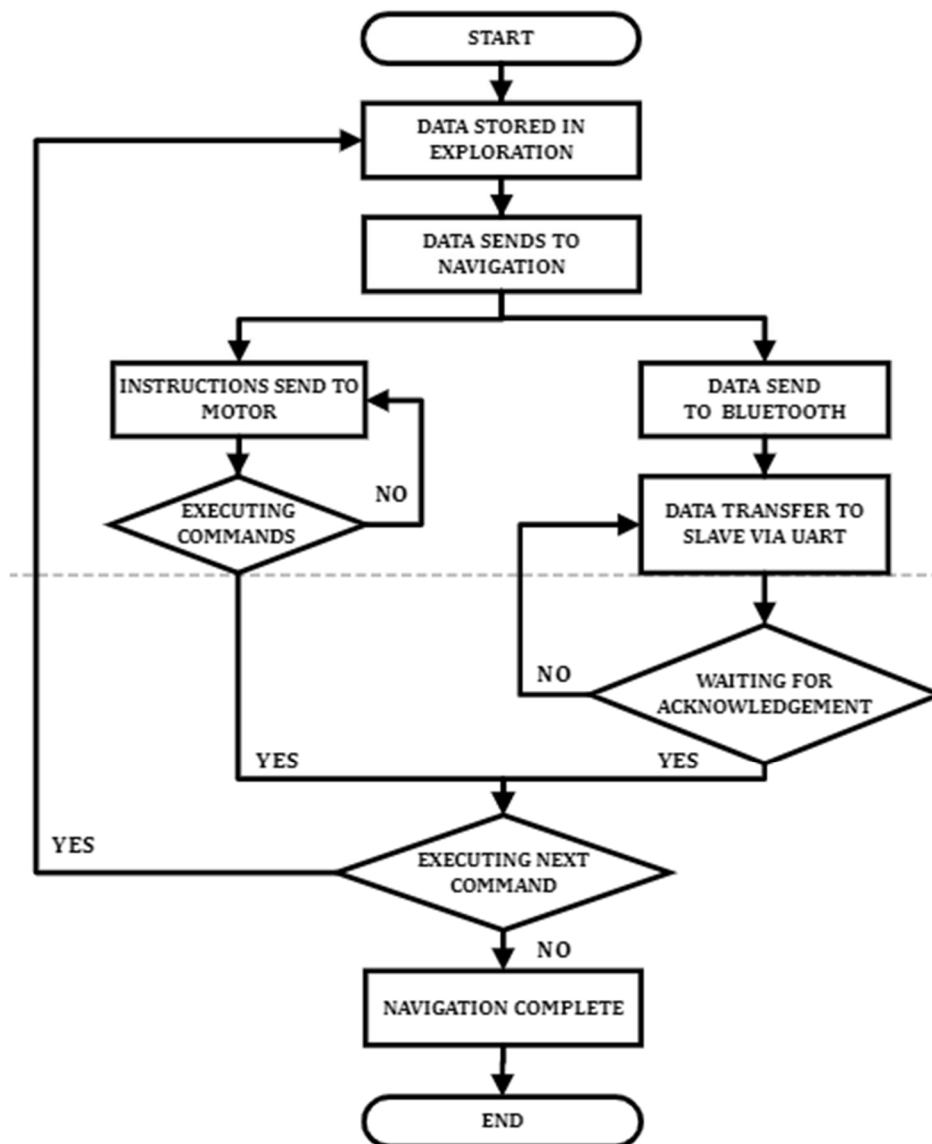


Figure 5. Flow chart of the Implemented design

Relevant information is sent to the data_send_module via the navigation. When the master robot is recognized first, the data_send_module uses Bluetooth connection to send data to the slave. If the role_module functions are enabled on the robot, a comparison between comparison_module, which includes slave robot data and the data_send_module will be performed. When slave robot-1 receives data from the master robot, the slave_connection_module receives acknowledgement. Slave robot-1 will be separated from master robot and slave robot-2 will be attached, with data_send_module transmitting it over Bluetooth connection. The data is delivered to slave robots in this manner and the acknowledgement is sent via the receiver module. The acknowledgement will be received in the data_comparison_module and the next data frame will be delivered constantly. The role_module and data_comparison modules as well as the data_send_module and slave_connection modules will be active if the robot is a slave. When the robot is slave, the data comparison module is initiated and stepper motor and sensor data are transferred to it. When navigational information is captured from the master robot via Bluetooth, it is delivered to the data_comparison_module. The slave robot will acknowledge to the master robot that it is changing to the next cell when the stepper_motor_control data, sensor data and received data are all accurate. If the received data does not match, the robot assumes it has discovered an obstruction and acknowledges that it has encountered one. The slave robots receive additional instructions from the master robot in order to move in the most efficient manner. To signify that it is connected to slave robots, a master LED blinks.

VII. SIMULATION

The receiver acknowledgement is maintained in simulation so that we can see when the Bluetooth specifications have been modified and then the next instruction can be issued as shown in Figure. 6. and Figure. 7 shows the data frame structure. Navigation module as shown in Figure. 8.

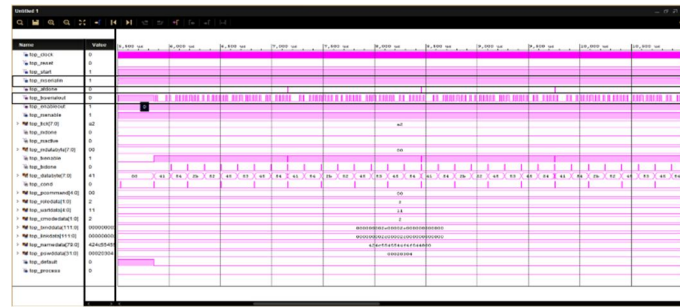


Figure 6. Simulation results of the AT commands executed in ZedBoard FPGA

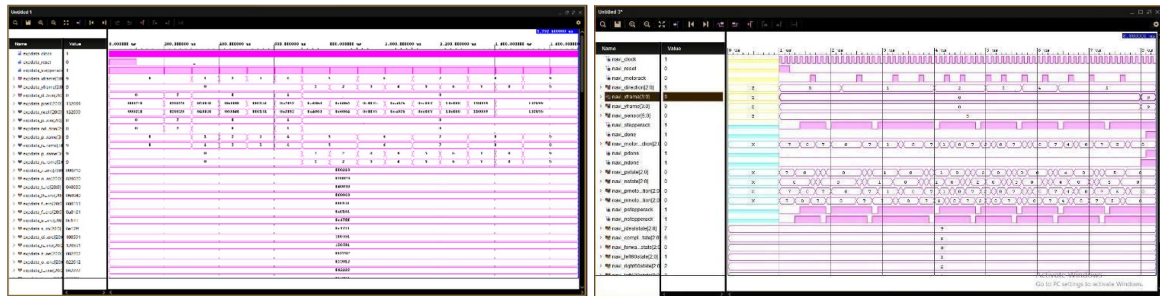


Figure 7. Simulation results of the Exploration where it sends the next cell frame which contains cell information

Figure 8. Simulation results of the Navigation module where it will decode the frame and send the direction to the motors

TABLE III. UTILIZATION OF THE FPGA DEVICE

Logic Utilization	Zed board FPGA	
	Resources	Percentage
No. of Slices	3826	3.31
No. of LUTs	5941	3.82
Slice Register	7356	2.45
No. of Bonded IOBs	44	3

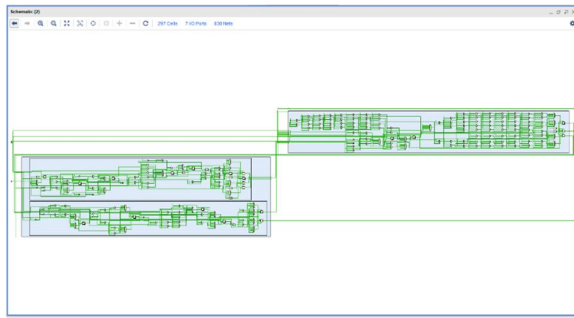


Figure 9. RTL Schematic of the Bluetooth AT command logic

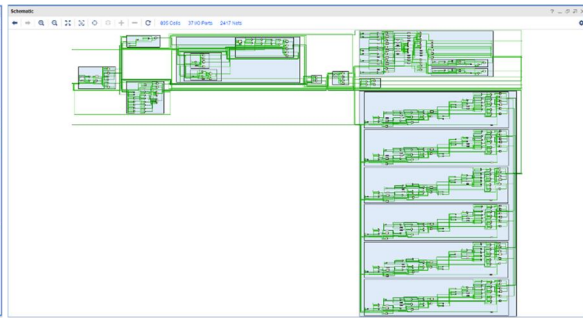


Figure 10. RTL Schematic of the proposed logic

VIII. RESULTS

The Fig. 11 represents the explorations of the robots in different algorithms in the environment. The navigation of the robots is shown in Fig. 12 where robots are using A* algorithm and here Master is guiding the Slaves.

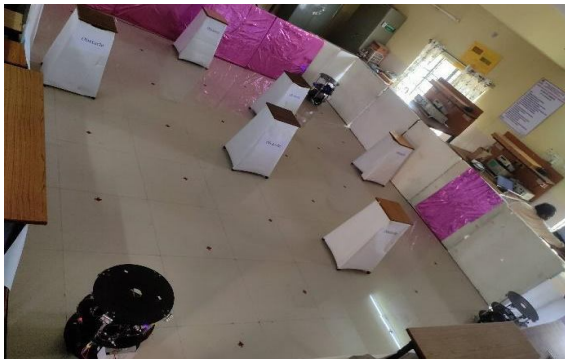


Figure 11. Exploring the environment using different searching Algorithms



Figure 12. Navigating slaves by the Master in environment

IX. CONCLUSION

The primary goal of this project is to design and create a viable multi-robot system with a Bluetooth module. Exploration, Navigation, Path Planning and robot communication are all being developed. The AT commands are programmed in Verilog and the FPGA controls the Bluetooth connections and disconnections. This research eliminates the need for an additional microcontroller to instruct AT instruction to Bluetooth, calculates the precise location of slave robots and allows the robot to do the task in the shortest possible time. The number of robots may be increased to 7 slave robots using the Bluetooth piconet function. This design may be employed in retail industry products warehouses. This technique might be expanded to include SCATTERNET navigation for multirobot.

ACKNOWLEDGEMENT

The authors thank and acknowledge the Management, Principal of B. V. Raju Institute of Technology, Narsapur, Telangana, India. For their constant encouragement in all phases of this work. The author and co-authors also thanks Mr. Kishore Vennela, Assistant Professor for his support, encouragement to stand by with this work.

REFERENCES

- [1] C. Ju, Q. Luo and X. Yan, "Path Planning Using an Improved A-star Algorithm," 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), 2020, pp. 23-26, doi: 10.1109/PHM-Jinan48558.2020.00012.
- [2] Y. Zhou, X. Jin and T. Wang, "FPGA implementation of a* algorithm for real-time path planning", *Int. J. Reconfigurable Comput.*, vol. 2020, Aug. 2020.

- [3] Y. Li, D. Dong and X. Guo, "Mobile Robot Path Planning based on Improved Genetic Algorithm With A-star Heuristic Method," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 1306-1311, doi: 10.1109/ITAIC49862.2020.9338968.
- [4] R. Yang and L. Cheng, "Path Planning of Restaurant Service Robot Based on A-star Algorithms with Updated Weights," 2019 12th International Symposium on Computational Intelligence and Design (ISCID), 2019, pp. 292-295, doi: 10.1109/ISCID.2019.00074.
- [5] M. Basha, S. K. M and M. C. Chinnaiah, "Deliberation of Robotic Services to Human kind using FPGA based Robot," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), 2019, pp. 1079-1083, doi: 10.1109/ICICT46008.2019.8993409.
- [6] C. Yu, M. Ku and H. Lin, "A Hybrid Mesh-Ring Topology for Bluetooth Networks," 2018 IEEE 5G World Forum (5GWF), 2018, pp. 520-523, doi: 10.1109/5GWF.2018.8517076.
- [7] A. N. Sabri, N. H. M. Radzi and A. A. Samah, "A study on Bee algorithm and A algorithm for pathfinding in games," 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2018, pp. 224-229, doi: 10.1109/ISCAIE.2018.8405474.
- [8] Azpúrua, H., Freitas, G., Macharet, D., & Campos, M. (2018). Multi-robot coverage path planning using hexagonal segmentation for geophysical surveys. *Robotica*, 36(8), 1144-1166. doi:10.1017/S0263574718000292
- [9] G. Tatar and S. Bayar, "FPGA Based Bluetooth Controlled Land Vehicle," 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), 2018, pp. 1-6, doi: 10.1109/ISAECT.2018.8618830.
- [10] C. -C. Hsu, R. -Y. Hou, W. -C. Kao and S. -A. Li, "FPGA-based path planning using improved Ant Colony Optimization Algorithm," 2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2015, pp. 443-444, doi: 10.1109/ICCE-Berlin.2015.7391304.
- [11] Size Xiao, A. Postula and N. Bergmann, "Optimal random sampling based path planning on FPGAs," 2016 26th International Conference on Field Programmable Logic and Applications (FPL), 2016, pp. 1-2, doi: 10.1109/FPL.2016.7577392.
- [12] Garima, N. Agarwal and S. R. N. Reddy, "Design & development of daughter board for Raspberry Pi to support Bluetooth communication using UART," International Conference on Computing, Communication & Automation, 2015, pp. 949-954, doi: 10.1109/CCAA.2015.7148533.
- [13] A. H. Ismail et al., "Investigation of homogeneous multi robots communication via Bluetooth," 2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE), 2012, pp. 124-129, doi: 10.1109/ISCAIE.2012.6482082.
- [14] R. Aneesh, B. Sreekumari. and K. Jiju., "Design and implementation of Bluetooth MAC core with RFCOMM on FPGA," 2012 Annual IEEE India Conference (INDICON), 2012, pp. 422-426, doi: 10.1109/INDCON.2012.6420655.
- [15] V. B. Misic and J. Misic, "On Bluetooth piconet traffic performance," The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002, pp. 501-505 vol.1, doi: 10.1109/PIMRC.2002.1046751.
- [16] D. F. Wong and K. -. The, "An algorithm for hierarchical floorplan design," 1989 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers, 1989, pp. 484-487, doi: 10.1109/ICCAD.1989.76996.