Hardware Implementation of Compact Genetic algorithm for Robot path planning in Globally static Environment in 8-bit Microcontroller.

Mian Mujtaba Ali
Electrical Engineering Department
Bharia University, Islamabad, Pakistan
mujtabaali.buic@bahria.edu.pk
Omer Farooq
Computer Science Department
Bahria Univesity, Islamabad

Omer Farooq
Computer Science Department
omerfarooq.buic@bahria.edu.pk

Bahria Univesity, Islamabad
e-mail: omerfarooq.buic@bahria.edu.pk

Dr Shyqyri Haxha
Department of Electronic Engineering
School of Engineering, Physical and Mathematical Sciences
Royal Holloway, University of London, Egham, Surrey, TW20 0EX, United Kingdom
Tel: +44 (0)178444 3183
Email: Shyqyri.Haxha@rhul.ac.uk

***Abstract*—In this paper, hardware implementation of genetic algorithm for Robot path planning in Globally static Environment is presented. Genetic algorithm is modified and implemented in 8-bit Microcontroller (MCU) PIC18F452. The genetic algorithm is designed to decrease number of iteration and processing power by using predefined priorities for parent/ initial path generation rather than creating parent paths randomly. The unmanned ground vehicle (UGV) is designed which receives starting node, final destination and obstacles wirelessly, it then create multiple parent paths by using different priorities, cross over to create new child paths, uses distance as fitness function for determining Optimal or shortest path while avoiding the obstacles and uses stepper motors with three- dimensional movements to reach its destination. The environment is 5x5 static Grid Map (25 nodes) in which obstacles are known before path planning. The MCU determines optimal path in which it faces no obstacles and require minimum distance to reach its destination.**

Keywords—Genetic Algorithm, Artificial Intelligence, Heuristic, Path planning.

I. INTRODUCTION

During the past few years, automation has become has become a very important phenomenon influencing our day to day life. Autonomous robots have now become a major part of this trend. Therefore, navigation of the robots from their source to destination has attained a great deal of attention. Motion planning is a crucial part of intelligent autonomous mobile robots which help robot in navigating the environment, different algorithm is implemented and a lot of research is carried out for robot path planning [1-3]. Path planning allows a robot to navigate and identify a feasible (a collision free) path from its source to its destination [4]. Path planning research covers a wide range to techniques implemented in either static or dynamic environments. The path planning for robots can mainly be divided into two components, mapping and path planning algorithms [4]. Initially the environment will be mapped and using these maps it becomes easier for the implementation of path planning algorithms. Given the entire description and environment, the path planning between the initial and the final point must be a collision free and must fulfill all the criteria of optimization. In other words, path planning is the creation of a collision free path of in a given environment keeping in mind the criteria for optimization [5]. The criteria for optimization can be time, power, distance etc. So therefore the criteria for optimization depend on the desired application. Suppose that we want to make our optimization with respect to time, and then the algorithm implemented should consider the path planning for the minimum time irrespective of the power consumed and distance covered. Similarly, if the criteria for the path planning are distance then the path planning algorithm should implement it for shortest distance from is source to its final position irrespective time taken and power consumed and an approach is presented in this paper for path planning of the shortest distance between two points [1].

In global path planning the environment is completely known and static. In this method the algorithm generates an optimal collision free path from the start point to its destination even before the unmanned ground vehicle (UGV) starts its motion. In local path planning, optimal path is found while the UGV in motion and it is capable of producing new path in response to the changing environment i.e. obstacle avoiding robot using Infrared or ultrasonic sensors.

Genetic algorithm is a method that has been used to find the optimal collision free feasible path. Since the appearance of this algorithm, it has been used to in the optimal path planning of mobile robots by generating different paths and using a crossover operator. This algorithm has been inspired from evolutionary biology meaning survival of the fittest. This method generates the all possible combinations of the feasible paths and selects the optimal one depending upon the selection criteria i.e. distance, time, safety [6]. The genetic Algorithm uses the cross over operator to generate new solution by crossing over two or more initial solutions to obtain new solutions and then uses selection criteria to select the best possible solution [7].

An Implementation of Genetic Algorithm is presented which uses on board microcontroller to calculate the path while avoiding obstacles [8]. The presented implementation uses Genetic Algorithm to generate paths by using randomly while avoiding obstacle. Similarly, ant colony optimization (ACO) and Genetic algorithm (GA) is implementation for robotic path planning are presented and compared for different complexities [9, 10]. ACO has mainly advantage in environment where multiple robots are used and the robot updates the feasible paths to server or to other robots but Genetic algorithm is used to calculate the feasible path without the knowledge from the other robots rather it uses the known environment and selection criteria to select best possible solution.

This paper presents an implementation of Genetic Algorithm which is modified to generate initial paths while keeping certain selection priorities so it converges to the solution more quickly. The Genetic algorithm uses a globally static environment i.e. with stationary obstacles. The environment can be representation of certain factory where obstacles like machines are to be avoided while planning optimal path.

An autonomous UGV is designed to find optimal path in a globally static environment using Genetic Algorithm.The UGV designed is capable enough to find the collision free, optimal, time and shortest path from its source to destination. The UGV is designed using stepper motors for the better accuracy in navigation through the environment.

The goal was to develop unmanned ground vehicle that is able to find the optimal path from its source to destination by using genetic algorithm in a globally static environment and for situations where it's dangerous and unreachable for human beings. This UGV will also be very handy for surveillance and security purposes. Although unmanned ground vehicles have already been built before by using pre embedded maps and charts which aid in finding the optimal path but this approach presents a designed UGV which find feasible path by using modified genetic algorithm. The genetic algorithm is designed and modified to be implemented in 8-bit microcontroller. The Genetic Algorithm is repetitive & complex algorithm which normally requires a lot of computing power. A globally static environment of 5x5 or 25 nodes grid map is selected and GA is implemented using PIC18F452 Microcontroller. PIC18F452 has a 32KB program memory with 256 bytes of EEPROM and max crystal speed of 40MHz [11]. The microcontroller supports serial communication without the need of any serial adapter. MPLAB ide with c18 compiler is used to program the microcontroller and maximum code optimization is used to compress the code to the microcontroller program memory of 32KB [12].

II. ALGORITHM DEVELOPMENT

Genetic algorithm is a method that is used to generate and find the optimal collision free path from a pool of feasible paths. Since the appearance of this algorithm, it has been used to in the optimal path planning of mobile robots. This algorithm has been inspired from evolutionary biology. This method generates the all possible combinations of the feasible paths and selects the optimal one.

The following are our specification for GA based project.

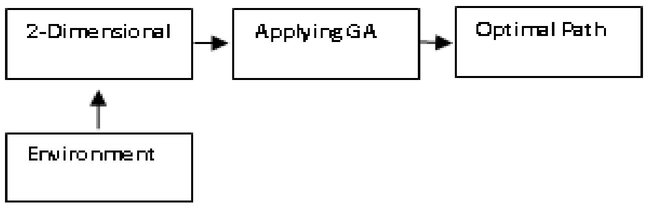


Fig. 1. G.A Implementation

i) A grid map of 5x5 will be known to our UGV.

ii) The start, destination points and the position of the obstacles are also known.

iii) The UGV will be free to move on path that doesn't contain any obstacles.

The Environment is encoded as 2- dimensional (2-D) map where obstacles are identified so the UGV can use the information while planning the paths. This method makes it easy for a search algorithm and is much easier to calculate.

An array is used for mapping the environment in Microcontroller as 2-D environment. It contains the starting position of UGV, destination and obstacles but first the nodes are number from 1 to 25 as shown in the figure 2 below.

21	**22**	**23**	**24**	**25**
16	**17**	**18**	**19**	**20**
11	**12**	**13**	**14**	**15**
6	**7**	**8**	**9**	**10**

1	**2**	**3**	**4**	**5**

Fig. 2. Mapping of Environment

The nodes are combined to form paths. This map represents a 2-Dimenstional Environment. The map is shown below in figure 3 has horizontal and vertical axis which is used for the movement and selecting the next node for the path in environment. The UGV is bound to select the node adjacent to the nodes to forms paths at the same time avoiding any node in selection process that has obstacle in it.

5	**(1,5)**	**(2,5)**	**(3,5)**	**(4,4)**	**(5,5)**
4	**(1,4)**	**(2,4)**	**(3,4)**	**(4,4)**	**(5,4)**
3	**(1,3)**	**(2,3)**	**(3,3)**	**(4,4)**	**(5,3)**
2	**(1,2)**	**(2,2)**	**(3,2)**	**(4,2)**	**(5,2)**
1	**(1,1)**	**(2,1)**	**(3,1)**	**(4,1)**	**(5,1)**
Rows/Columns	***1***	***2***	***3***	***4***	***5***

Fig. 3. Columns and Rows of Environment

An array of created to mimic this 2-D environment in the microcontroller the array will have 26 index, and each index denotes a gene/node which the UGV will use to form parent paths.

```
                   0,     0,    0,     0,    0};
                   0,     0,    0,     0,    0,
                   0,     0,    0,     0,    0,
                   0,     0,    0,     0,    0,
Char Map [26]={1,0,       0,    0,     0,    0,
```

Fig. 4. 2-D mapping using array

The obstacle is denoted by node having value of 1 which means UGV and node having a value of 0 means the UGV can visit that node and use path containing that node i.e. the index 0 (Map [0]) of array Map is out of bond and here contains a value of 1 & index 1 (Map [1]) value is 0 meaning UGV can visit that node.

First, starting position and ending position are used to determine their rows and columns. These value are used by Genetic Algorithm to form parent paths. A modified approach is used to form parent paths which forms parent paths using nodes adjacent to the nodes and at the same time keeping track of optimal nodes and obstacles. This approach decreases the processing power and converge to solution quickly. Microcontroller uses different priorities rather than selecting randomly to generate parent paths using nodes i.e. starting position of UGV is 2 and ending position is 24. There are obstacles at position 12 and position 13. The microcontroller first determines row and column numbers of starting and ending nodes. It is done by function getrownumberandcolum() designed having different mathematical operations i.e. for obtaining column number node index is divided by 5 which is total number of columns and the remainder here is the column number in case of starting position 2%5=2, for row number 2/5+1 but the floating value is ignored i.e. 2/5=0 and adding 1 to that results in row number equal to 1. Next the microcontroller generates two different parent paths using different priorities while keeping final destination column, row number and obstacles in mind. Each time different adjacent nodes are chosen while choosing the parent path using pre-defined priorities which using current and final path to select different priorities each time a parent path is generated.

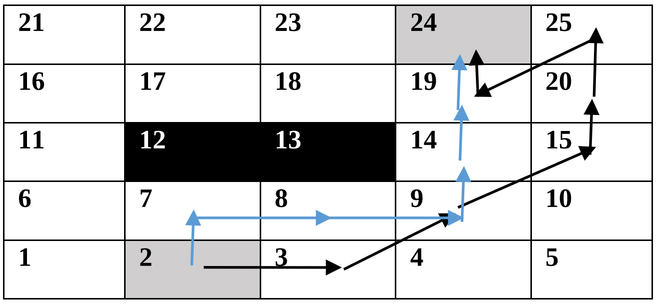


Fig. 5. 5x5 Parent Paths

Two parent paths/ chromosomes are created using predefined priorities and obstacles.

Parent 1= 2, 3 9, 15, 20, 25, 19, 24
Parent 2= 2, 7, 8, 9, 14, 19, 24

The next part is to create child paths using common nodes in parent paths using crossover () function that is created which using parent paths, find common node and exchange them i.e. for the above parent path, the common node is 9 so the cross over path will exchange them to form new child paths. The exchange of the information about the direction and distance of a path is called a crossover. In this method the two parent chromosomes exchange a portion of their information producing two off springs out of which one off spring will exhibit the solution for an optimal path.

Offspring 1= 2, 3, 9, 14, 19, 24
Offspring 2= 2, 7, 8, 9, 15, 20, 25, 19, 24

The two parent chromosomes represent two feasible paths which are cross over by using the common node producing two new paths off spring 1 and off spring 2. The feasibility function checks the distance of each path and save the smallest while discarding the remaining non feasible paths. The same process of generating parent path using different priorities is repeated again and again while doing crossover and checking feasibility. The maximum number of paths generated by microcontroller is 60 in this case.

If our UGV is to move from the lower right corner to the upper left corner of the grid as shown in figure 2.1. In order for the UGV to complete its task there will be two types of movements [1]. Time is used as their optimization criteria, and taking too much turns would increase their delay towards their destination [1]. But our UGV is equipped with three types of movements as our selection criteria is shortest distance. The three type of motions are mentioned below.

i) Row wise movement

In the row wise movement, the UGV starts moving row by row from its start point to its final position. In other words, any horizontal line in search space followed by the UGV but there will be no movement in upwards or back direction

ii) Column wise movement

In a column wise movement, the UGV starts moving in towards its destination in a column by column movement. Thus the UGV will exhibit only the vertical movement

iii) Row/column or 45-degree movement.

In this type of movement, the UGV is able to move at an angle of 45 degrees. Thus the UGV will an executing a row and a column wise movement simultaneously.

Eight different types of movement are possible during navigating the map as shown in the fig. 6 below.

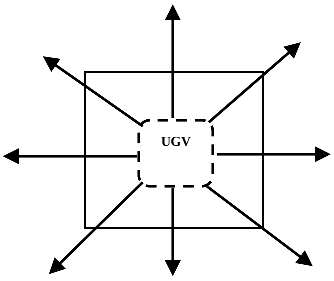
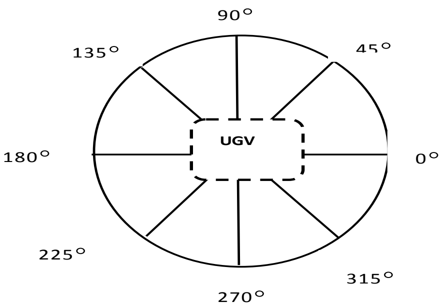


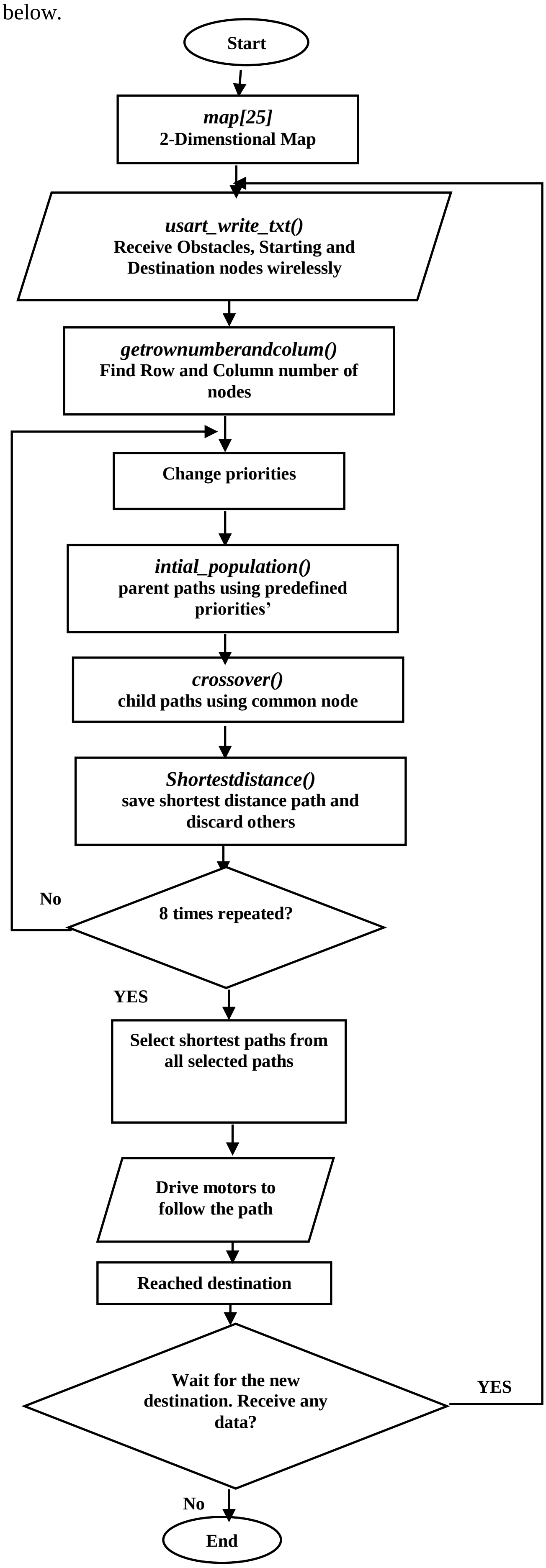
Fig. 6. Possible UGV movements

The microcontroller finds the columns and rows of starting and final destination using mathematical operations and use this information to decide which node to move next. If only the column is increased i.e. If starting position is 7 and UGV has to move to 12 that means a simple forward movement will be suffice except if there is in its path similarly if starting position is 7 and UGV has to move to 13 t hen the UGV can go to 12 using straight motion and then move right but it can also move diagonally to the right from position 7 which will be more feasible as UGV has to cover less distance but path selection depends upon the obstacles in its path and displacement.

Another important aspect of the movement is movement along UGV axis which is required during turning left, right diagonal movements, forward and backwards. The microcontroller saves and update the current position of UGV as shown in the figure below. The microcontroller keeps records of the current movement and uses a function setdegree (int degree) to change its angle by specified degrees which in turn uses left() and right() functions to turn left or turn right respectively.



The flow chart of Compact Genetic Algorithm is shown below.



(Field Effect Transistor). The overall Block Diagram of the project is shown in fig. 9.

III. SIMULATION

The genetic algorithm was tested in proteus on PIC18Fmicrcontroller microcontroller in 5x5 grid. Two obstacles were placed on the grid and the starting and destination points were also given to the microcontroller like shown in figure 3. The UGV was able to find the optimal find the optimal path from its source to the destination. The figure 8(a) below shows the microcontroller finding the possible feasible paths for the UGV. The optimal path selected by the microcontroller is shown below and after that the microcontroller drives the motor by calling left, right, forward functions as shown in figure 8(b) below.

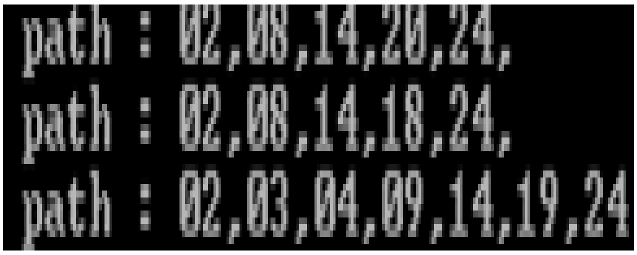


Fig. 8(a). Different paths Generated

Fig. 8(b). Final path Selected and started movement algorithm

The advantage of using predefined pirorities is that its minimize the computational power of Genetic algorithm and converge to solution quickly. If no predefined priorities are used then next node is generated randomly which will result in a more iteration. The predefined prirorites are selected using mathematical operations i.e. if the destination row and coulumn number both is greater than the current node than it will try to move diagonally right as the first pirority because UGV has to travel less distance this way. Similarly othe pirorities are selected such way that it would converge to solution quickly that feature is why it is called compact or optimized genetic algorithm.

IV. HARDWARE

The Hardware is an 3 wheel UGV with free forward wheel and two rear wheels controlled by six pole stepper motors for turning and moving. A 5x5 Grid map is made to test te designed Path implementation. The intial and final nodes alog with obstacles is transmitted wirelessly to the UGV where PIC18F452 microcontroller receives the nodes position and use Genetic algorithm to find optimal path and drive the UGV using stepper motors controller circuit which is built using Bipolar Juntion transistors (BJTs) and FET

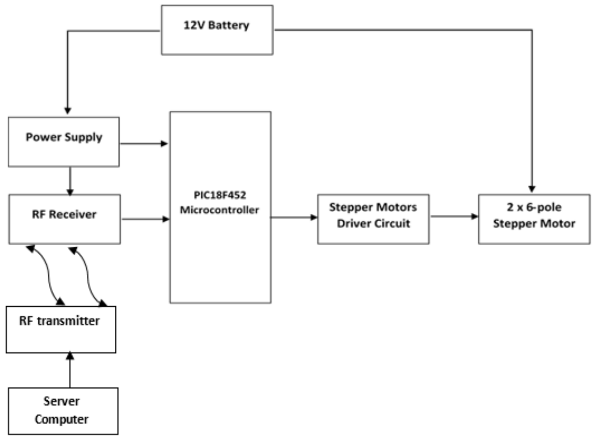


Fig. 9. Block diagram

The stepper motor control is designed using a combination of PNP BJTs and NMOSFETs because of high current that will flow throw stepper motor. When PIC microcontroller pin is low i.e. 0V, the PNP transistor is on which means the NMOSFET gate output is pulls to ground which turns off the NMOSFET and no current will follow through the coil of motor and alternatively the PNP transistor is off when PIC microcontroller pin is high i.e. 5V, the input at NMOSFET gate is 12V which turns on the FET and current will follow through the stepper motor coil. The stepper motor used is Unipolar and the code used to move forward the motor are Pic Microcontroller pins are 1010, 0110, 0101 and 1001. The circuit Diagram is shown in the fig. 10 below.

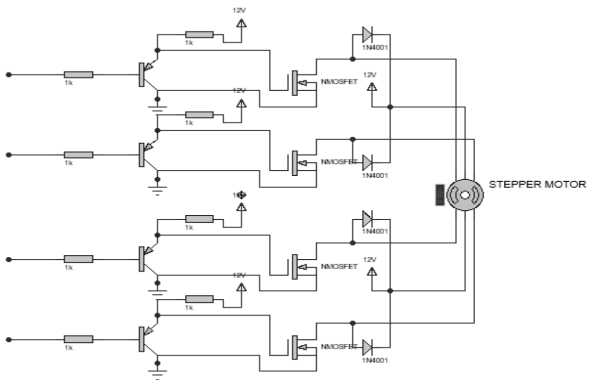


Fig. 10. Stepper Motor Control

UGV requires a constant source of battery to power motors and circuitry. A 12-volt rechargeable battery is used to power the stepper motors. For the microcontroller and other circuitry that requires a constant supply of 5-volts, the 7805 voltage regulator is used [14]. A diode and capacitors are added so the current will flow in forward direction and removing any ripples in output.

The positioning of the UGV plays a very important part for the implementation of the genetic algorithm. It is important for the UGV to know its current location so that is can plan it next step towards the optimal path & the human controller for the UGV must also know the exact location of the UGV so that he or she can map the starting, final positions as well as obstacles on the grid. The UGV receives the current and final position via RF module and the user also updates the obstacles positions using server (in this case a laptop). The microcontroller receives the starting and ending positions and use the Genetic Algorithm to obtain and navigate the environment. The interfacing and communication of different modules is an important feature in our project. The RF transceiver RF7020 [13] is used with a baud rate of 9600bps, the microcontroller is connected to RF module via TX and RX pins (25, 26). The Server or PC is connected to via UXART to USB interface because the RF7020 only supports serial communication which requires Serial port i.e. DB9 [16]. The laptop communicates with microcontroller via RF communication and microcontroller will interface with the motor drive circuit electrically.
The 5x5 Grid map is built to test the system and the results are found to be exactly same as of the simulation. Fig. 11 shows a proteus simulation of the UGV calculation feasible paths using Genetic Algorithm which are shown using virtual terminal communicating same as RF transceivers. The UGV find optimal paths and then drives the motors represented here by right left or forward motions. As seen in the fig. 12 the UGV avoids the nodes with obstacles on it which in this case was 11.

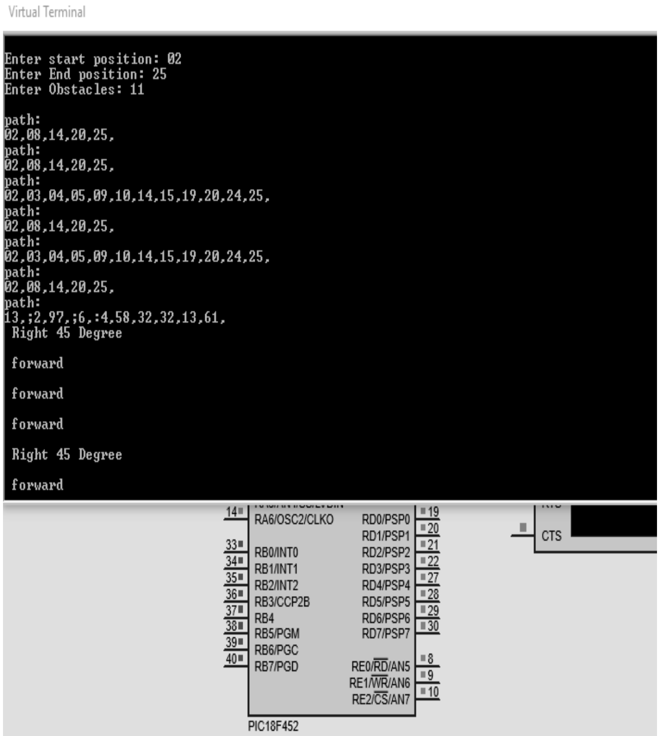


Fig. 11. Proteus Simulation

The final system setup is shown in the fig. 12 below:

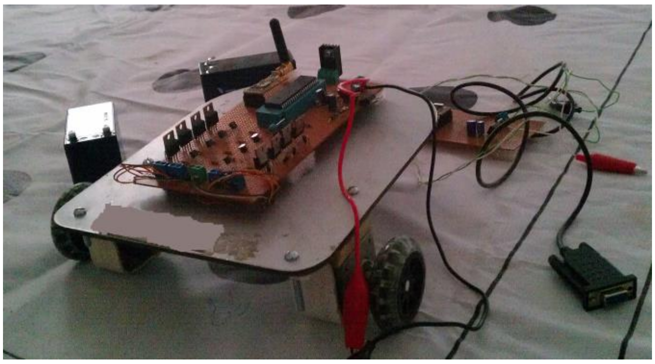


Fig. 12. Hardware

V. CONCLUSION

The Genetic Algorithm is successfully modified and designed to be implemented in 8-bit microcontroller PIC18f452. The unmanned ground vehicles received the coordinates using RF modules from server and then used on board Genetic Algorithm to successfully calculate, avoid obstacles and navigate the path. The Genetic Algorithm is modified to converge to the optimal solution and its successful implantation on such low cost and low computational architecture proves the modified approach of using different set of predefined navigation priorities. The

Genetic Algorithm can be modified using the same approach while increasing the predefined priorities using more advanced microcontroller which will improve the navigation. The UGV is able to self-navigate in globally static environment i.e. factories while avoiding the obstacles. The UGV performs Real time optimal path planning and uses the optimal path for navigating.

REFERENCES

[1] Sedighi, Kamran H., et al. "Autonomous local path planning for a mobile robot using a genetic algorithm." *Evolutionary Computation, 2004. CEC2004. Congress on*. Vol. 2. IEEE, 2004.

[2] Raja, P., and S. Pugazhenthi. "Path planning for mobile robots in dynamic environments using particle swarm optimization." *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on*. IEEE, 2009.

[3] En.wikipedia.org. (2015). *Robotic mapping*. [Online] Available at: http://en.wikipedia.org/wiki/Robotic_mapping [Accessed 24 Oct. 2015].

[4] Correll, N. (2017). *Introduction to Robotics #4: Path-Planning*. [online] Correll Lab. Available at: http://correll.cs.colorado.edu/?p=965 [Accessed 24 Oct. 2017].

[5] Konda, Divya, Keerthana Bhoopanam, and Saravanakumar Subramanian. "A Review of Global Path Planning Algorithms for Planar Navigation of Autonomous Underwater Robots." *Soft Computing Techniques in Engineering Applications*. Springer International Publishing, 2014. 99-113.

[6] Sivanandam, S. N., and S. N. Deepa. *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.

[7] Ortiz-Boyer, Domingo, César Hervás-Martínez, and Nicolás García-Pedrajas. "CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features." *J. Artif. Intell. Res.(JAIR)* 24 (2005): 1-48.

[8] Burchardt, Hagen, and Ralf Salomon. "Implementation of path planning using genetic algorithms on mobile robots." *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006.

[9] Sariff, Nohaidda Binti, and Norlida Buniyamin. "Comparative study of genetic algorithm and ant colony optimization algorithm performances for robot path planning in global static environments of different complexities." *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on*. IEEE, 2009.

[10] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." *IEEE computational intelligence magazine* 1.4 (2006): 28-39.

[11] Microchip.com. (2017). *PIC18F452 - Microcontrollers and Processors*. [online] Available at: http://www.microchip.com/wwwproducts/en/PIC18F452 [Accessed 24 Oct. 2017].

[12] Microchip.com. (2017). MPLAB® C Compiler for PIC18 MCUs (C18) - SW006011 | Microchip Technology Inc. [online] Available at: http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=SW006011 [Accessed 24 Jan. 2017].

[13] Appconwireless.com. (2017). RF7020-27. [online] Available at: http://www.appconwireless.com/ProShow.asp?id=386 [Accessed 24 Feb.

[14] Corner, L. (2017). All About 7805 IC | Voltage Regulator Pin Diagram & Schematics. [online] Electronics for You. Available at: http://electronicsforu.com/electronics-projects/7805-ic-voltage-regulator [Accessed 24 Feb. 2017].