iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** Issue I **Month of publication:** May 2015

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Shortest Path Finding Based Architecture for Vehicle Navigation

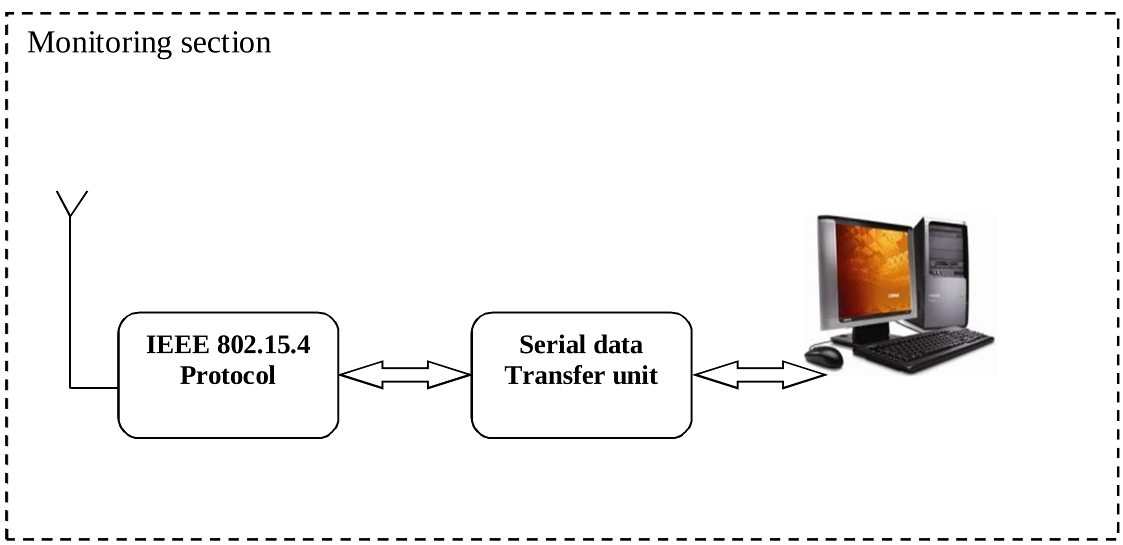
S Pavani[1], M. Kezia Aruna Jyothi[2]
[1]PG Student, Department of ECE, ASCET, Gudur, Andhra Pradesh, India
[2]Associate Professor, Department of ECE, ASCET, Gudur, Andhra Pradesh, India

Abstract *: This paper is a DIJKSTRA'S algorithm based architecture designed for the purpose of finding the shortest path between the nodes in vehicle navigation. Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex*
Keywords: ROBOTICS, RTOS, GSM, ARM.

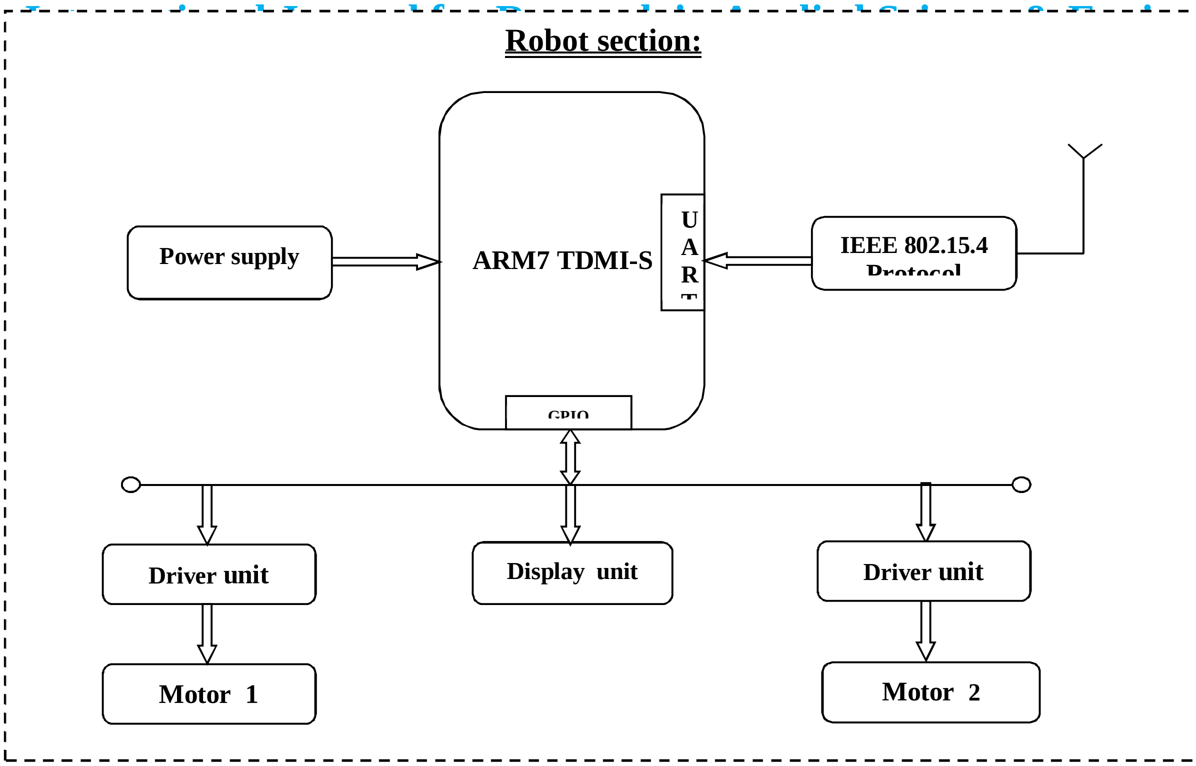
I. INTRODUCTION

The project deals with the vehicle navigation using DIJKSTRA'S algorithm. The data transmission time is increased with the protocol standard. One of the section runs with driver unit and LPC2929 with display unit and another PC as server section runs on a Matlab platform. Communications between two nodes (hardware and application) are accomplished through IEEE 802.15.4. The user can give the source and destination node address to the server section. Using DIJKSTRA'S algorithm the shortest path will be find out and graph plot will be displayed on the server section. Using IEEE standard communication protocol the shortest path will be feeded into the robotic module. Using the path as a reference, the robot moves in the ordered direction. After reaching the destination node, the display unit displays the name (particular place) of the particular node.

A. Block diagram



Volume 3, Special Issue-1, May 2015
ISSN: 2321-9653



ARM core:

II. LPC2148 PROCESSOR

LPC2148Microcontroller Architecture. The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of micro programmed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective processor core. Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory. The ARM7TDMI-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM7TDMI-S processor has two instruction sets:

The standard 32-bit ARM set.

A 16-bit Thumb set.

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code. Thumb code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system

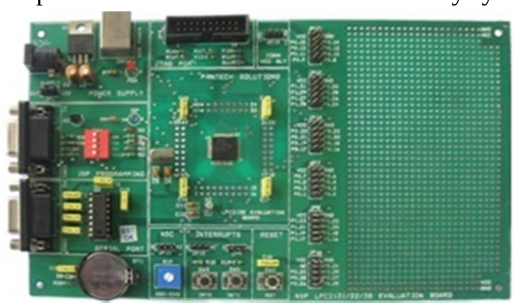


Figure 2 : ARM7TDMI PCB board

A. LCD Display Unit

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal. An LCD consists of two glass panels, with the liquid crystal material sand witched in between them. The inner surface of the glass plates are coated with transparent electrodes which define the character, symbols or patterns to be displayed polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle. One each polarisers are pasted outside the two glass panels. These polarisers would rotate the light rays passing through them to a definite angle, in a particular direction. When the LCD is in the off state, light rays are rotated by the two polarisers and the liquid crystal, such that the light rays come out of the LCD without any orientation, and hence the LCD appears transparent. When sufficient voltage is applied to the electrodes, the liquid crystal molecules would be aligned in a specific direction. The light rays passing through the LCD would be rotated by the polarisers, which would result in activating / highlighting the desired characters. The LCD's are lightweight with only a few millimeters thickness. Since the LCD's consume less power, they are compatible with low power electronic circuits, and can be powered for long durations.

B. Dijkstra's Algoritham

This algorithm finds the shortest path from a source vertex to all other vertices in a weighted directed graph without negative edge weights.Here is the algorithm for a graph G with vertices $V = \{v_1, \ldots v_n\}$ and edge weights w_{ij} for an edge connecting vertex v_i with vertex v_j. Let the source be v_1.Initialize a set $S = \emptyset$. This set will keep track of all vertices that we have already computed the shortest distance to from the source. Initialize an array D of estimates of shortest distances. $D[1] = 0$, while $D[i] = \infty$, for all other i. (This says that our estimate from v_1 to v_1 is 0, and all of our other estimates from v_1 are infinity.)Essentially, what the algorithm is doing is this :Imagine that you want to figure out the shortest route from the source to all other vertices. Since there are no negative edge weights, we know that the shortest edge from the source to another vertex must be a shortest path. (Any other path to the same vertex must go through another, but that edge would be more costly than the original edge based on how it was chosen.) Now, for each iteration, we try to see if going through that new vertex can improve our distance estimates. We know that all shortest paths contain subpaths that are also shortest paths. (Try to convince yourself of this.) Thus, if a path is to be a shortest path, it must build off another shortest path. That's essentially what we are doing through each iteration, is building another shortest path. When we add in a vertex, we know the cost of the path from the source to that vertex. Adding that to an edge from that vertex to another, we get a new estimate for the weight of a path from the source to the new vertex. This algorithm is greedy because we assume we have a shortest distance to a vertex before we ever examine all the edges that even lead into that vertex. In general, this works because we assume no negative edge weights. The formal proof is a bit drawn out, but the intuition behind it is as follows: If the shortest edge from the source to any vertex is weight w, then any other path to that vertex must go somewhere else, incurring a cost greater than w. But, from that point, there's no way to get a path from that point with a smaller cost, because any edges added to the path must be non-negative.By the end, we will have determined all the shortest paths, since we have added a new vertex into our set for each iteration.This algorithm is easiest to follow in a tabular format.The adjacency matrix of an example graph is included below. Let a be the source vertex

C. L293D Driver

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high current/high-voltage loads in positive-supply applications. All inputs are TTL compatible.

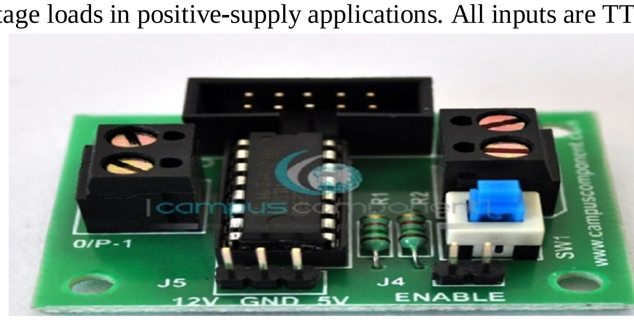


Figure 3: L293D driver unit

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression.

D. ZIGBEE

The ZigBee normal has adopted IEEE 802.15.4 as its Physical Layer (PHY) and Medium Access management (MAC)protocols. Therefore, a ZigBee compliant device is compliant with the IEEE 802.15.4 protocal normal. Here 2 zigbee modules are using one at Robot section side and other one at monitoring section. This can be used to transfer the data between Robot and monitoring section.

Figure 4 : ZigBee module

E. DC Motor

In this project 2 DC motors can be used to drive the Robot . Here the motor requires 12v but ARM7 processor will not provide that much of voltage so driver unit can be used.

Figure 5 : DC motor

F. Ultrasonic Sensors

Whenever Robot is moving then if there is any obstacles then there may be a chance to collision so to avoid this in this project ultrasonic sensor can be used. If there is any obstacle then robot will take another direction.

Figure 6 : Ultrasonic sensor

III. RESULTS

he base design of the project is showing bellow. This is a robot which is used to move in different directions based on the input.

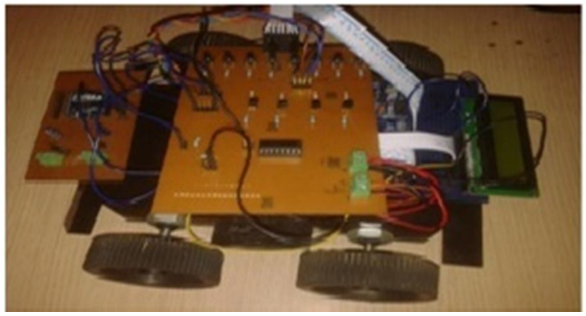


Figure 7: Robotic section of the project

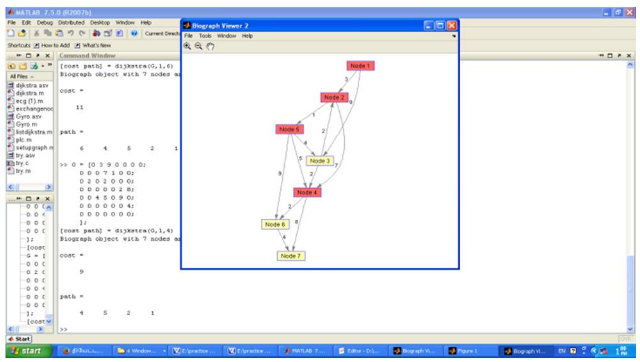


Figure 8: command window of MATLAB

Above figure shows the how we are giving input to the Robot using MATLAB command window and how the shortest path can be calculated using MATLAB.

IV. CONCLUSION

The user can give the source and destination node address to the server section. Using DIJKSTRA'S algorithm the shortest path will be found out and graph plot will be displayed on the server section. Using IEEE standard communication protocol the shortest path will be feeded into the robotic module. Using the path as a reference, the robot moves in the ordered direction. After reaching the destination node, the display unit displays the name (particular place) of the particular node.

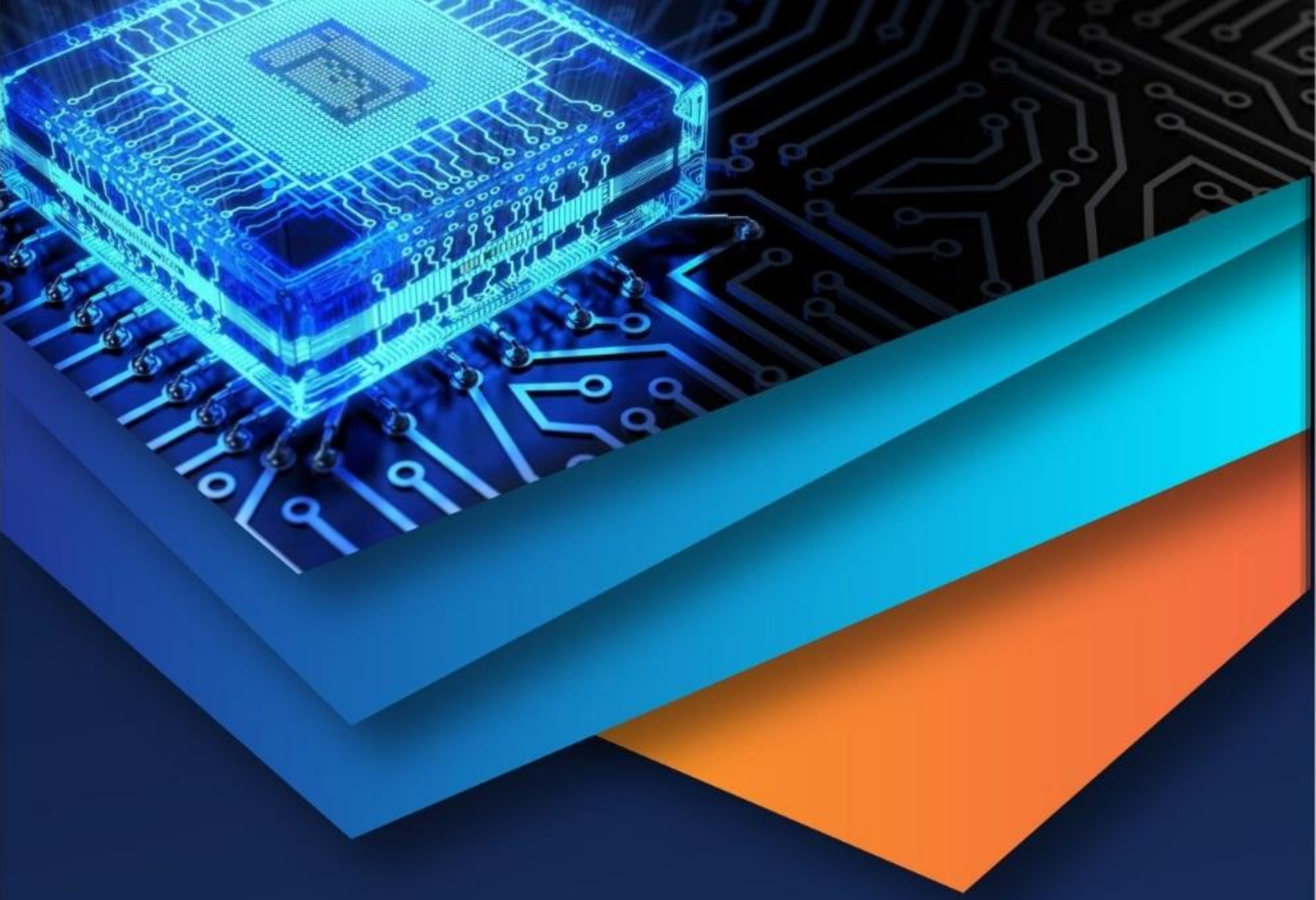
REFERENCES

[1] National Highway Traffic Safety Administration," Automotive Collision Avoidance Systems (ACAS) Final Report". August 2000.
[2] "Google Cars drive themselves in Traffic", J. Markoff - The New York Times, 2010 - bngumassd.org
[3] Tyler Folsom, "Social Ramifications of Autonomous Urban Land Vehicles" IEEE International Symposium on Technology and Society, May 2011, Chicago
[4] "A Method for Shortest Path Search using Extended Dijkstra's Algorithm", IEEE International Conference on System, Man and Cybernetics, 2000
[5] "Proof of a modified Dijkstra's algorithm for computing shortest bundle delay in networks with deterministically time-varying links", Lee, Fraser.S, IEEE Xplore 2011
[6] T.Imielinski and J.C. Navas, "GPS-Based Geographic Addressing, Routing, and Resource Discovery," Commun. ACM, vol. 42, no. 4 Apr. 1999
[7]Q.Xu, R.Sengupta and D.Jiang, "Design and Analysis of Highway Safety Communication Protocol in 5.9Ghz DSRC," Proc. IEEE VTC vol.57, no.4, 2003, pp. 2451-2455

AUTHORS

S.Pavani received her B.TECH degree in Electronics and Communication Engineering from Priyadarshini college of Engineering and Technology, Kanuparthipadu, Nellore (Dist), affiliated to JNTU Anantpur. She is currently pursuing M.Tech Embedded systems in Audisankara college of Engineering and Technology (Autonomous), Gudur, Nellore (Dist), affiliated to JNTU Anantpur

M.Kezia Aruna Jyothi received her B.Tech Degree in Electronics and Instrumentations Engineering from N.B.K.R.IST,Vidhyanagar, Nellore (Dist), AP in 2003. She received her M.Tech in ECE from R.G.E.M.C.E.T Nandhyal in 2006. She has 9 years teaching experience, presently working as Assoc. Professor in the the department of ECE, Audisankara College of Engineering and Technology (Autonomous), Affiliated to JNTU, Anantpur.



10.22214/IJRASET

INDEX COPERNICUS
45.98

ISRA JIF
IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY