International Journal of Robotics and Automation (IJRA)
Vol. 4, No. 2, June 2015, pp. 143~149
ISSN: 2089-4856

Wall-E Surveyor Robot Using Wireless Networks

Aatish Chandak, Arjun A, Nithin Kamath
Department of Electronics and Communication Engineering,
National Institute of Technology Karnataka, Surathkal, India

Article Info

Article history:

Received Feb 9, 2015
Revised Apr 25, 2015
Accepted May 12, 2015

Keyword:

Autonomous Navigation
Dropbox API
Microcontrollers
Raspberry Pi
Wall Follower
Wireless Networks

ABSTRACT

The methods for autonomous navigation of a robot in a real world environment are an area of interest for current researchers. Although there have been a variety of models developed, there are problems with regards to the integration of sensors for navigation in an outdoor environment like moving obstacles, sensor and component accuracy. This paper details an attempt to develop an autonomous robot prototype using only ultrasonic sensors for sensing the environment and GPS/ GSM and a digital compass for position and localization. An algorithm for the navigation based on reactive behaviour is presented. Once the robot has navigated to its final location based on remote access by the owner, it surveys the geographical region and uploads the real time images to the owner using an API that is developed for the Raspberry PI's kernel.

Corresponding Author:

Arjun A,
Department of Electronics and Communication Engineering,
National Institute of Technology Karnataka, Surathkal,
D-403, Mega-Tower I, NITK Hostels, Srinivasanagar, Surathkal, Mangalore – 575025, Karnataka, India.
Email: arj.11EC14@nitk.edu.in

1. INTRODUCTION

Autonomous robots are machines that perform tasks without human intervention. For these robots to interact with their surroundings, they must be programmed to respond to outside stimuli. Often, these stimuli can contain chaotic and unpredictable variables which make them difficult to model. This is the reason why many kinds of robots have some degree of autonomy, while complete autonomy is hard to achieve.

Still, a high degree of autonomy is particularly desirable in fields considered dangerous for human intervention and are increasingly performed by autonomous systems. These include wastewater treatment, poisonous gas leakage revelation, mine detection and general surveillance, all of which present unique challenges in navigation and mobilization. Outdoor navigation currently presents the biggest challenge to designers of robotic systems and even though a robot that can maneuver autonomously in an urban environment is extremely desirable, it is still a long time from occurring due to several factors. Problems encountered outdoors by a robot will be different than those encountered indoors. Types of outdoor obstacles cannot be easily categorized because of their wide variety. Therefore, outdoor autonomy is most easily achieved in the air and sea, since obstacles are less common. On ground, several issues arise, including terrain, weather, random moving obstacles and thus increase the weaknesses of the sensing devices themselves. To overcome these sensor limitations, robots combine the use of high-end sensors such as stereo cameras, laser rangefinders, wireless networks like GPS/ GSM, inertial measurement units (IMU) and sonar. These sensors, along with the hardware and software that have to accompany them, constitute a great cost in investment for the robot. Due to large budget constraints and component non-availabilities, we were only able to come up with, what we thought is, an economically feasible prototype. If in case we were equipped with highly accurate electrical and mechanical components, we can assure all that this would have been a wonder bot.

2. MOBILE ROBOT PROTOTYPE

2.1 The Plan

The aim of this publication is to present a simple, low cost method of autonomous navigation for a mobile robot that was developed using the subsumption architecture. This allows the robot to divide its tasks into simple behaviours, which are layered on top of each other. Simple control systems are implemented at each layer of the program to verify the functionality of the system. The design we have made tries to demonstrate how a commercially available sensor can be integrated into one package in order to provide a solution to the outdoor navigation problem. The robot will therefore have to navigate autonomously using only GPS/ GSM module and a digital compass as its only means of localization. To achieve this, ultrasonic sensors help the robot to interact with its environment while the global positioning system module guides it to its destination. We have used a highly accurate GPS module for continuous monitoring of the robot's current location with respect to itself so that it keeps a track of its current location with respect to its destination.

The subsumption model that we have incorporated deals with three particular behaviours which allow it to interact with its surroundings. These behaviours are unobstructed navigation, perimeter following and obstacle avoidance. These are part of the behavioural logic of the robot which uses information from its sensors to respond to the changes in its environment. This means that the robot will not have to be provided with mapping or localization algorithms, but instead only uses an ultrasonic emitter and receiver for sensing and reacting. Each kind of behaviour gave rise to a complex control system, which by itself would not solve the outdoor navigation problem but when layered with other behaviours provides a feasible solution.

The application level development involved the use of the robot once it reached its final destination. As soon as it reaches its destination, a call is made to the Raspberry Pi from the mBED controller to capture images of the surroundings at various angles along the azimuthal plane. The captured images are relayed back to the owner using his Dropbox account for which a specific and developed API was used.

2.2 Vehicle Description

The autonomous vehicle's navigation algorithm is based on the a philosophy that robots should be like insects, equipped with simple control mechanisms tuned to their environments, instead of complex architectures that require complete models of the environment for decision making. The figure shows the hardware structure of the robot that houses the electronic components used to interface the embedded microprocessor LPC1768 ARM-Cortex M3 digital signal controller board (developed by ARM) with motors and sensors. The housing supports a highly accurate GPS module, a digital compass module, an LCD screen used for testing and the mountings that are needed for the connections to the motors and the power supply through a rechargeable battery.

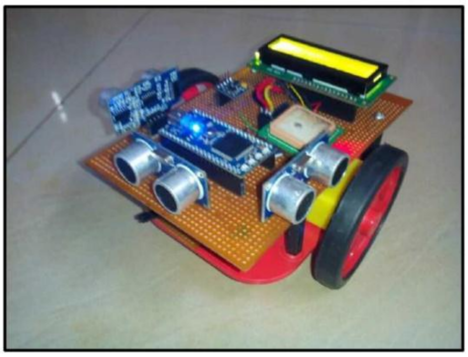


Figure 1. Hardware structure of the robot

Underneath the top layer of the robot is an L298 motor controller that is incorporated to handle higher currents demanded by the motors. Finally, supports and metal contacts are soldered onto the GPCB platform in order to mount the ultrasonic sensors used for obstacle detection. The GSM module is initially placed alongside the robot but the connecting jumpers are removed as soon as the final location is retrieved by the robot from the remotely present owner as we were unable to build a more rigid mechanical structure.

3. MICROCONTROLLER AND COMPONENT INTERFACES

The MBED NXP LPC1768 microcontroller that is used as the brain of our robot, is packaged as a small DIP form-factor for prototyping and uses a built-in USB FLASH programmer. Hence, it is easily programmable. The controller is based on a 32-bit ARM Cortex-M3 core running at 96MHz and includes a multiple interfaces including built-in Ethernet, USB Host and Device, CAN, SPI, I2C, ADC, DAC, PWM, etc. The pin-out shows the commonly used interfaces and their locations. All the numbered pins (p5-p30) can also be used as General Purpose Input-Output (GPIO) pins which gives us the flexibility of placing our components anywhere on the main framework. Hence, we had chosen a controller with a very powerful set of interfaces with the necessary communication protocols as well as a well-established debugging online environment to compile our logical thinking, translated from the algorithm that was developed.

3.1 MBED to GPS and GSM modules interfacing through UART

The UART serial function uses two eTPU channels to provide a 3-wire (TX, RX, and GND) asynchronous serial interface as Figure 2. It can be used to add serial capability to a device without a serial port, or to add additional serial I/O to a device which already has a hardware UART. The function requires no host CPU intervention during data frame transmission or reception. One eTPU channel is configured to function as the serial transmitter (TX), and the other channel is configured to function as a serial receiver (RX).

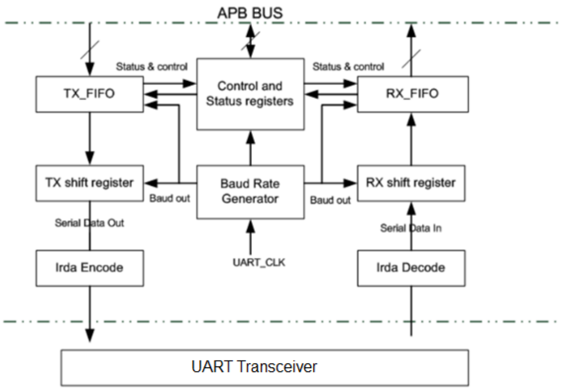


Figure 2. The UART serial function

Both the transmitter and receiver channels have the same attributes in terms of baud-rate, data-size, parity etc. In this regard, the GPS modem has its TX pin connected to the RX pin of the MBED while its RX is connected to the TX pin of the MBED. The GND and VCC connections are made appropriately. The MBED houses 3 UART modules, which allow it to interact with three different devices serially. The GSM modem responds to special prompts called AT (Attention) commands. The microcontroller forces these commands to the GSM through a program written in the software debugging environment to which the GSM responds. The AT commands used here are AT+CMGS (sending messages to a known end-user) and AT+CMGR (reading the SMS sent by the end user and parsing the final latitude and longitude needed for the purpose). In this regard, the GSM modem sends and receives characters as packets based on the UART protocol such that the RX pin of the mBED is connected to the TX pin of the GSM modem through a Max-232 board. The additional purpose of this board is to reduce the 12V adapter DC operating voltage of the GSM to the microcontroller working voltage of 3.3V.

3.2 MBED to Digital Compass interface – I2C

I²C (Inter-Integrated Circuit), a multi-master serial single-ended computer bus for attaching low-speed peripherals to embedded systems, defines basic types of messages, each of which begins with a START and ends with a STOP message: single message where a master writes data to a slave; single message where a master reads data from a slave; combined messages, where a master issues at least two reads and/or writes to one or more slaves, as shown in Figure 3. With only a few exceptions, I²C does not define message semantics, such as the meaning of data bytes in messages. Message semantics are product-specific, in this case, the HMC5883L digital compass module. These exceptions include messages addressed to the I²C general call address (0x00) or to the SMBus Alert Response Address; and messages involved in the SMBus Address Resolution Protocol (ARP) for dynamic address allocation and management.

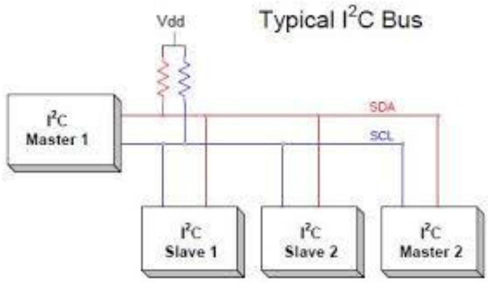


Figure 3. I²C (Inter-Integrated Circuit)

We have made the compass adopt the request/response control models, wherein one or more bytes following a write command are treated as a command or an address. Those bytes determine how subsequent written bytes are treated and/or how the compass responds on subsequent reads. Most of these operations involve single byte commands. Thus, the digital compass exchanges data of the magnetic field magnitudes along all three dimensions with the microcontroller. The magnetic fields are different at different points on the earth and this point is exploited in the construction of the digital compass. The magnetic fields along the azimuthal plane are necessary for the calculation of the orientation and thus helpful in the navigation process. We have tested the robot on a flat and levelled surface, hence, there is no need for the magnitude along the direction of elevation, since the robot does not traverse along the third dimension.

3.3 MBED to LCD panel (Verification and Debugging) and Ultrasonic Sensor Interface – GPIO

General-purpose input/output (GPIO) are generic pins on a microcontroller whose behaviour can be controlled (programmed) by the user at run time. GPIO pins have no special purpose defined, and go unused by default. The idea is that sometimes the system integrator building a full system that uses the controller might find it useful to have a handful of additional digital control lines, and having these available from the controller can save the hassle of having to arrange additional circuitry to provide them. So, the data pins D4-D7, Register Select pin and the control pin of the LCD are connected to six well configured pins of the MBED board. The other pins of the LCD that need either digital high or low voltage are soldered accordingly. Also, the three ultrasonic sensors used for obstacle avoidance have their trigger and echo pins connected to configured pins of the MBED through the GPIO protocol. As is evident from the pin diagram of the MBED, any of the numbered pins can act as a GPIO pin, and so, we had the flexibility to choose the appropriate pins at random locations for configuring the ultrasonic sensors.

3.4 MBED to DC Motor Interface through L293D – PWM

Pulse-width modulation is a modulation technique that conform the width of the pulse, formally the pulse duration, based on modulator signal information, as shown in Figure 4. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors. The main advantage of PWM is that power loss in the switching devices is very low. When a switch is off there is practically no current, and when it is on, there is almost no voltage drop across the switch. Power loss, being the product of voltage and current, is thus in both cases close to zero. In this regard, the PWM signals are used to control the speed of the robot by controlling the motors through the L293D motor driver IC.

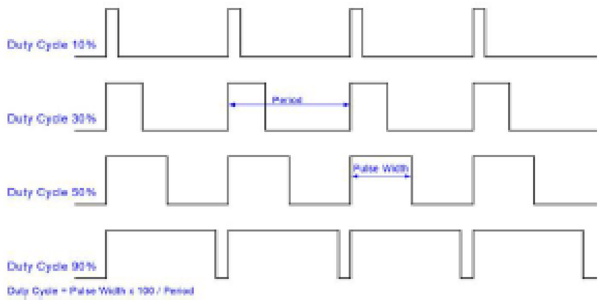


Figure 4. Pulse-width modulation

3.5 MBED to Raspberry Pi Interface - USB

Once the robot reaches the destination intended, it prompts a signal to the Raspberry PI to start capturing images of its surroundings for which a dedicated algorithm was built. After all the images are captured at the destination, they are uploaded on to the Dropbox server using the API that was already developed. The MBED and Raspberry communicate with each other through the USB protocol. USB is an industry standard that defines the cables, connectors and communications protocols used in a bus to standardize the connection of computer peripherals and microcontrollers to communicate and supply electric power.

4. TECHNICAL STRUCTURE AND PCB LAYOUT

As written earlier in section 3.1 to 3.5 on the component interfaces, the various components have been connected through different interfaces to the MBED as shown in Figure 5. The robot has been developed to house the Pi for capturing images through a WebCam and uploading the captured image onto a Dropbox account using the API that was developed. The PCB Layout (as shown in Figure 6) was a single-layered one on a GPCB, so, the soldering of the components had to be done with high amounts of accuracy and precision since we wanted to avoid the use of wires and jumpers as much as possible.

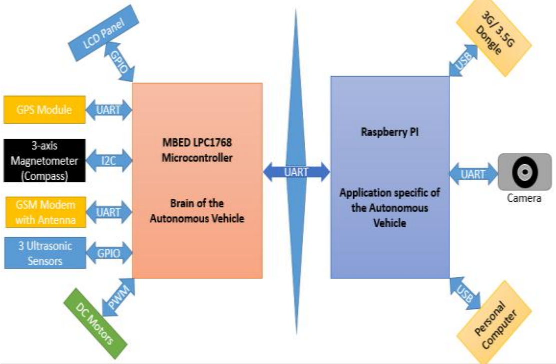


Figure 5. Interfaces to the MBED

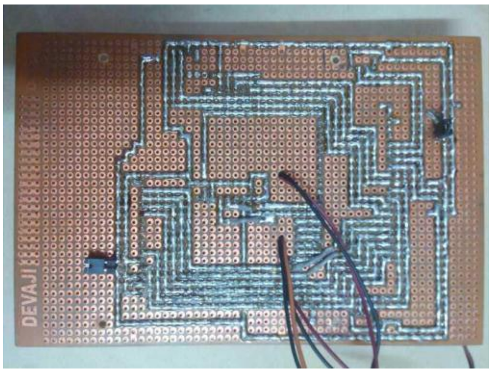


Figure 6. PCB Layout

5. FLOWCHART/ ALGORITHM DEVELOPED

The toughest part in the construction of the algorithm (as shown in Figure 7) was to ensure that the robot acted like an insect that only responded to changes in the environment using only the ultrasonic sensors as its means to detect objects.

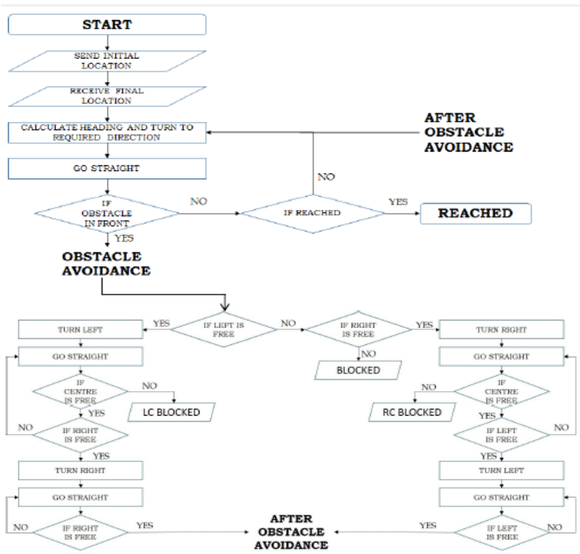


Figure 7. Algorithm for (un) obstructed navigation

The problems of localization and shortest path navigation were taken care by the wireless networks and the digital 3-axis magnetometer that acted as a digital compass. Initially, the master sends a message to the robot asking it to send its current geographical location (latitude and longitude) and other features for the master to tabulate. On receiving the initial co-ordinates, the master sends an SMS to the robot that showcases the latitude and longitude of the destination that the robot has to traverse to in order to survey that particular location. Once the robot receives the co-ordinates, it parses the data received into only what is required for it to calculate the actual heading with regards to its current magnetometer values it reads from the digital compass. The robot turns swiftly to the direction of intended path and upon comparison with the heading values provided by the digital compass, it stops. This ensures that the robot tries to reach the destination in the shortest path possible, that is, it “displaces” towards the destination from its current location. Once the robot turns to the required angle, it starts to navigate forward towards its destination. It monitors the external dynamic environment through ultrasonic sensors that detect any obstacle that may affect in the free running of the robot. A comprehensive wall-follower algorithm was used to detect the presence of obstacles and to check if it is able to navigate in its free state. If obstacles are detected, it follows another set of customarily designed algorithm as shown in the above figure. Once it tackles the obstacle, it tries to once again calculate the required heading to which it has to turn to in order to reach the particular destination. Once it reaches its destination, a request is sent from the MBED to the Raspberry Pi to enable it to capture a set of images. After every image is captured, the Pi prompts the MBED to rotate the robot by a certain degree to capture the next image. This process continues until the robot captures a 360 degree view of the surrounding. All the images are captured in the Pi’s embedded RAM. The captured images are then uploaded onto a Dropbox account that can be accessed by the user using an already developed API.

6. RESULTS

The robot successfully navigated to its destination using the algorithm developed while avoiding obstacles in its path. It ensured that the navigation algorithm worked fine and was independent of any other parameters that we were tracking. We found it difficult to configure and test the Raspberry Pi with the robot’s controller but after a series of experiments, we got that working too. The configuration of the Raspberry Pi was the toughest job at hand. The steps have been elaborated here. We downloaded PUTTY, a software to create a Secure Shell Host in the Raspberry Pi and XMING for configuring PUTTY. We then changed a few parameters in the file cmdline.txt in the boot folder to set the IP address of the PI to be static. This was needed for hosting other peripherals. The IP address of the PI should be set as the IP address of the master. We then updated the Pi to download further packages like fswebcam, usb-modeswitch, python-serial etc. At every instant of time, fswebcam is used to take pictures and an internal motion detection program 'motion' is stopped only after which the camera takes pictures. The package usb-modeswitch was downloaded to allow a dongle or a USB tethered phone to be recognized as a USB modem and access the internet using it. We then changed the file /etc/network/interfaces to allow network access using the USB modem. We then added a default gateway every time we want to access the net in order to upload the images on Dropbox. A video demonstration of the same is available on the Youtube page of one of the authors.

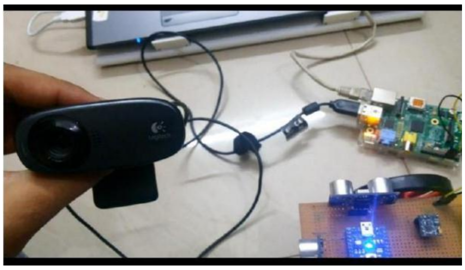


Figure 8. Debugging and Verification

7. CONCLUSION

When looking back at the difficulties encountered in the process of development, it can be said that the unreliable nature of the GPS receivers makes it undesirable for this type of application. It is therefore important to first develop a robust algorithm that can be implemented with other currently available sensors like stereovision and radar before the physical design of the robot is made. Particularly important is the combination of different type of sensors which when properly integrated can give an accurate representation of the robots surroundings. Only with this, will the problem of motion planning in an outdoor setting be solved. Having said this, the next step of investigation should focus on how to integrate these different sensors and use already existent techniques like Kalman filtering that have been shown to filter out bad data from multiple sensor inputs. Also, it is still necessary to continue with more research to prove if reactive behaviour can be a feasible solution for navigating in the outdoor environment when the unstructured settings play a role on sensor accuracy.

ACKNOWLEDGEMENTS

We would like to thank the Department of Electronics and Communication Engineering, National Institute of Technology Karnataka for providing us with the necessary resources to complete this project. We also thank them in encouraging us to publish the content of the undertaken project in the International Journal of Robotics and Automation. We would also like to specially thank Dr. Aparna P who had supervised us in our work. A part of the funds was sponsored by the NITK Chapter of IEEE and for that, we would like to thank them. We would also like to thank each other, our fellow classmates' and our juniors without whom this project would remain incomplete. We would also like to thank our parents, relatives and well wishers for their ever-encouraging support.

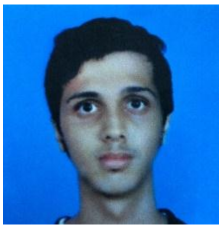
REFERENCES

[1] C. Shi, Y. Wang and J. Yang, "A local obstacle avoidance method for mobile robots in partially known environment", Robot. Auton. Syst. 58, 425–434, 2010
[2] S. Jacobs, A. Ferrein, S. Schiffer, D. Beck and G. Lakemeyer, Robust collision avoidance in unknown domestic environments, in: RoboCup 2009: Robot Soccer World Cup XIII. Springer, Berlin, pp. 116–127, 2010
[3] B. Hamner, S. Singh, S. Roth and T. Takahashi, An efficient system for combined route traversal and collision avoidance, Auton. *Robots* 24, 365–385 (2008).
[4] Control of Mobile Robots, Coursera Online Course conducted by Prof. Magnus Egerstedt of Georgia Institute of Technology
[5] Project URL on Youtube: https://www.youtube.com/watch?v=RKPB__AaQyw

BIBLIOGRAPHIES OF AUTHORS

Aatish Chandak is a Final Year Student Undergraduate currently studying in the Department of Electronics and Communication Engineering at National Institute of Technology Karnataka Surathkal. He is interested in working in the fields of Analog Electronics and Mechatronics. He wants to gain more experience in these fields and contribute to these fields in the best possible ways he can.

Arjun A is a Final Year Student Undergraduate currently studying in the Department of Electronics and Communication Engineering at National Institute of Technology Karnataka Surathkal. He is interested in working and researching in the fields of Digital Signal Processing, Computational Sciences, VLSI System Design, Wireless Sensor Networks and Computer Architecture. He wants to gain experience and skills through effective and efficient practice of his studies while appropriately building strategic approaches and realize their subsequent applications. He wishes to pursue an MS/ PhD in VLSI for Digital Signal Processing in an institute of high learning.



Nithin Kamath is a Final Year Undergraduate student currently studying in the Department of Electronics and Communication Engineering at National Institute of Technology Karnataka Surathkal. His areas of interest include VLSI System Design, Computer Architecture and Image Processing. He intends to gain knowledge in these areas and apply them so as to make a significant contribution in the development of new and efficient technologies. He wishes to pursue an MS degree in Computer Hardware Engineering in an institute of high learning.