Volume 11, Special Issue I, May 2020

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

CONFERENCE PAPER

Available Online at www.ijarcs.info

ARDUINO BASED IN-DOOR AUTONOMOUS ROBOT USING BLUETOOTH SENSORS

Siraj Basha Shaik
Department of CSE,
Reva University Bengaluru,Karnataka
Email:r17tcs16@cit.reva.edu.in

A Anand Shankar
Associate Professor
Department of CSE, Reva University Bengaluru, Karnataka
Email: anandshankar@reva.edu.in

Abstract: In door autonomous robots has wide scope in the current world of Industry 4.0. With the advent of low-cost micro controller platforms like Arduino, it is easier to design and deploy inexpensive robotic solutions in Retail environments like physical stores and supply chain nodes like Distribution/Fulfillment centers. The paper discusses a low-cost Arduino based 4 wheeledrobot which can navigate in the given environment with Bluetooth sensors as position nods and obstacle avoidance built-in. Given the diversity and interconnectivity of Bluetooth sensors, we can implement robots moving with centimeter accuracy which is the need in partially closed environments. The robot should navigate its way to the destination which is fed by a remote server with resides on the premise to achieve lowlatency.

Keywords: Autonomous robot; Bluetooth; sensor; Arduino

INTRODUCTION

Autonomous robots are the future of any industry and retail is no exception. The applications range from retail customer assistance, automatic label update and in stores to sorting packages in Distribution centers. With the advent of cyber-physical systems to be used in manufacturing chains, autonomous robots navigating with coordinating and navigating with centimeter accuracy is the essence. Since there is a requirement for a lot of since there is a requirement for a lot of these machines, the low cost of the platform helps industries evolve them faster.

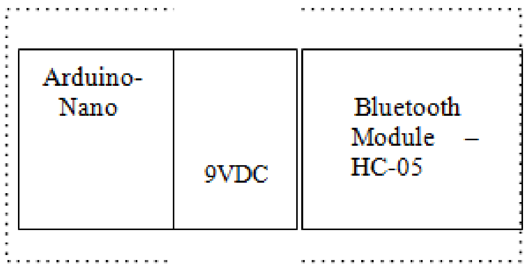
As of now, most robotic systems either use expensive wireless sensor networks or GPS and line following systems to achieve autonomy. The proposed system uses relatively inexpensive Bluetooth sensors to achieve similar capabilities. It is also easy to deploy and configure since the cluster of Bluetooth nodes need not be connected to each other and work like individual nodes broadcasting the address.

HARDWARE DESIGN

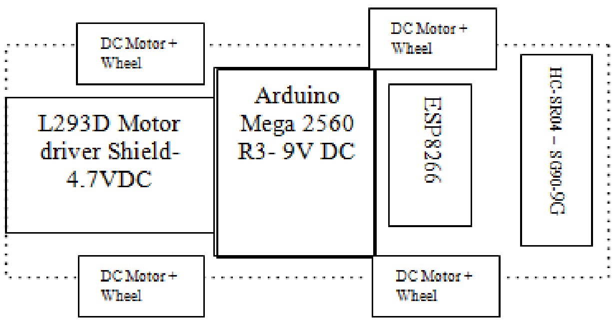
The robotic platform is based on a chassis of four wheeled robot driven by 4 dc motors individually driving each wheel. The motors are connected to the Motor Driver L239D which sits on Arduino Mega 2560 R3 as a shield. Wireless and Bluetooth capabilities are added by ESP8266 module.

Obstacle avoidance functionality is attained by incorporating a SG90 9G micro servo and an ultrasonic distance sensor module HC -SR04.Bluetooth nodes/sensors are made by connecting Bluetooth module HC-05 to Arduino-Nano powered by a 9V battery. These sensor nodes can independently Both the Arduino board and the Bluetooth sensor nodes are powered by a 9V Battery. The motor driver is powered by a 4.7 V power source. The block diagram is shown in Fig. 1. (a) and (b) depicts all the hardware components put on the robot chassis and sensor nodes.

The upper board hosts the Arduino Board and the motor driver shield. The front of the upper board is clamped with a servo motor and an Ultrasonic sensor. The DC motors are fixed in between the two boards of the chassis and connected to the wheels 2 on each side of the length of the chassis. Power supply is split between the middle of the chassis for the 9V battery powering the Arduino board and a 4.7 v battery pack powering the Motor Driver.



1.(a)



1.(b)

Fig. 1. (a) Bluetooth nodes (b) Robot

III. LOCALIZATION

A. Bluetooth node placement

Placement of the Bluetooth nodes is the most critical part of the indoor navigation. They should be placed optimally in order not to interfere with each other's signals and also to be detectable. To the robot's Bluetooth sensor. In a retail environment with a lot of Aisles and huge racks of storage, the Bluetooth nodes can be placed either on the ceiling or attached to the Aisles themselves.

Since the nodes are of low power, they can last for weeks without maintenance/change of battery. A monthly maintenance should be sufficient to get these going. If this solution needs to be set-up at a large scale, there are Bluetooth beacons available commercially which can lasts months without a battery change.

One of the main challenges is the localization method. The localization method considered here is RSS (Received Signal Strength) Technique and can be illustrated as below. Given the limited hardware and considering the absence of external factors like temperature changes, humidity etc. this method is most optimal in the current scenario.

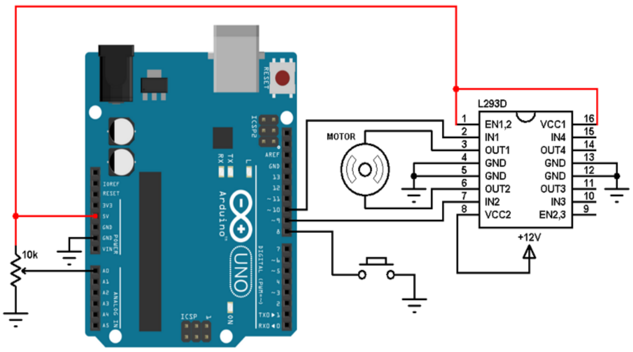


Fig. 2. Arduino interfacing with motor driver L293D

LOCALIZATION TECHNIQUE

The RSS technique [2] is based on the path loss propagation model. With the RSS for reference distance (d0), P0, it is possible to estimate the RSS for the distance d, d > d0, as itfollows:

$$P(d) = P0 - 20\lambda \log 10\, d/d0 + \eta$$

where λ is the path loss exponent and η is a random variable with Normal distribution.

ROBOT INTERFACING

The pictorial representation of circuit interfacing of the Robot is illustrated in Fig .2. The Arduino Megs2560 is connected to the Motor driver shield L239D via header ping which snap in.

Fig. 3. represents the interfacing of Motor driver with the servo and DC motors

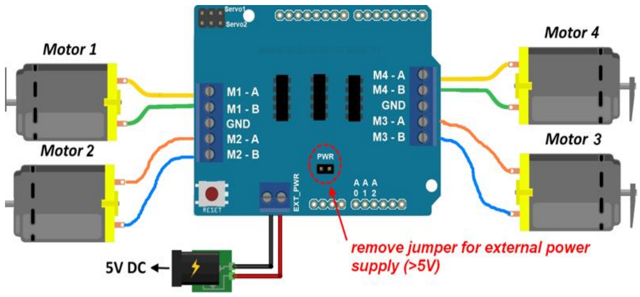


Fig. 3. L293D interfacing with Dc and servo motors

The wireless transceiver ESP8266 is interfaced directly with the Arduino. The micro servo is connected to the corresponding pins of the Motor Driver and the ultrasonic sensor is set-up on the Arduino. Fig. 4. Shows HC-SR04's interfacing with the Arduino.

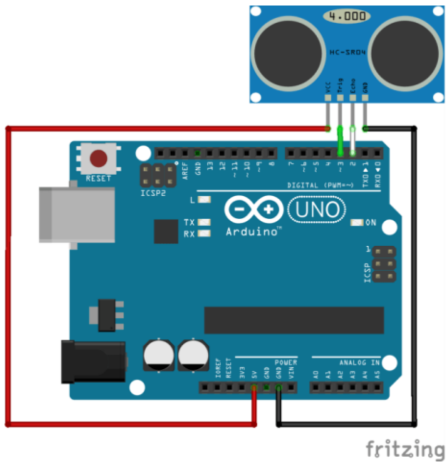


Fig. 4. HC-SR04 ultrasonic sensor with Arduino

CONTROL FLOW

The Brain of the robot is the Arduino and is further controlled by the central server which connects remotely to Arduino to instruct it to reach its destination. ESP8266 enables secure web connection from the Central server to the robot for instructions and also to load new programs. Considerkeystagesrequiredforinitialsystemsetup. The robot interacts with the Bluetooth nodes to estimate the signal strength and also avoiding obstacles if encountered. The Central server could be a simple X86 desktop of a cloud deployed server with access to Wi-Fi network of the environment.

A. Communication protocol

a. I2C communicationprotocol

The communication between Central server and Arduino uses the I2C synchronous protocol. Central server is the "Master" and initiates the data transfer to the "Slave" componentArduino, which provides the response messages. Messages sent on I2C bus are divided in two frames: an address frame representing the address to which the message is to be sent and a data frame that may contain one or more 8-bit fragments.

The software protocol implemented on the robot platform includes two types of messages in the data frame: *Command message* and *Request message*. According to this protocol the first byte of the data frame is the identification code of the message. For a *Command message*, Arduino performs the command received through the protocol, while upon receipt of a *Request message*, Arduino provides the necessary information for the Central server

The Table 1 shows all the messages implemented in the transmission between Raspberry Pi and Arduino Uno and Fig. 6 shows as example the *Command message* sent by the Server. In this message the address where the message should be sent is 0x04, representing the address of the Arduino Interface and the identification code of the message is22. The others 4 byte from the data frame set the duty cycles of the PWM command signals used by Arduino to control the desired speeds of the DC motors. The maximum 8 bit value sets a 100% duty cycle of the PWM command signal given by the Arduino.

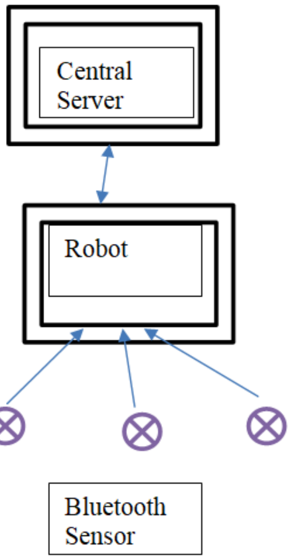


Fig. 5. Represents the Interfacing and Control flow between the components.

Table 1. Messages implemented for I2C serial communication.

*Commandmessages*22, 44, 55, 66, 77, 99, 110,121

Requestmessage 11,33

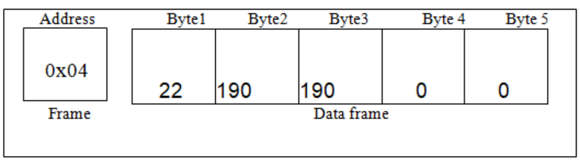


Fig. 6. Command message

The *Command message 22* described in the previous paragraphs sets the PWM signals duty cycle. *Command messages* with codes 44, 55, 66 and 77 allow the Server to control the status of the robot without directly modifying the PWM settings of the DC motors. Depending on the identification code from the message sent, the robot can be stopped (STOP), it can continue (CONT) or it can rotate to the left/right (LEFT / RIGHT). The code 99 can be used to move the robot a little bit forward when it is not close enough to the object that should be picked up. Codes 110 and 121 are used to enable/disable the robot scan mode when the robot does not have any information about the route and needs to do a 2D mapping. Code 110 activates the mapping mode and all ultrasonic sensors, in front and on each side of the chassis needed for the obstacles and walls detection. Code 121 disables this mode of operation and activates the line follower mode when only the front ultrasonic sensor is needed to detect the obstacles for a known route.

In case of *Request messages*, the Arduino responds with the ultrasonic sensors values (for code 11) or the infrared sensor values (for code 22).

The code is realized by Arduino IDE as sketches for the Arduino on the robot and Python running on the central server.

ENVIRONMENT

The environment is a 2D store environment with Aisles setup as nodes either on the ceiling or to the Aisles themselves. Fig 7 indicates the initial position of the robot and the placement of the nodes. The robot is placed at a reference point to start with which is also a Bluetooth node. It has to traverse through the cluster of nodes performing the designated tasks as listed below.

a) canning the bar codes

b) Counting products by RFID tags

c) Performing advanced functions like changing the labels with addition of robotic arms.

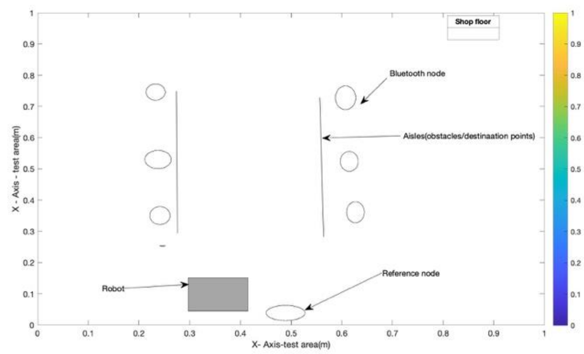


Fig. 7. Proposed 2D environment for the robot to navigate

CONCLUSION AND FUTURE WORK

Since the prototype with hardware and control flow is decided, the algorithm and implementation need to be worked on with multiple scenarios, plotting of the trajectory. Hardware components can also be refined and size minimized to make the robot light weight and powerful. Custom build microcontroller boards should be mass produced with all the required parts on a single chip to save costs and interface better. Multiple algorithms can be explored to refine the path planning, and use advanced functions like simultaneous localization and mapping (SLAM).

REFERENCES

[1] Mobile Robot Platform with Arduino Uno and Raspberry Pi for Autonomous Navigation - - Department of Electrical Engineering and Computers, University of Medicine, Pharmacy, Sciences and Technology of Targu Mures, Nicolae Iorga st., No.1, 540088 Targu Mures, Romania

[2] R. S. Rodrigues, M. Pasin and R. Machado, "Indoor position tracking: An application using the Arduino mobile platform," 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC), Valencia, 2017, pp. 1-8.

[3] Robotic Mapping: A Survey Sebastian Thrun February 2002 CMU-CS-02-111 School of Computer Science,Carnegie Mellon University Pittsburgh, PA 15213

[4] R. Siegwart, I.R. Nourbakhsh, Introduction to Autonomous mobile robots, A Bradford Book, The MIT Press, (2004).

[5] Nabeel Kadim Abid Al-Sahib, Mohammed Zuhair Azeez, Build and Interface Internet Mobile Robot using Raspberry Pi and Arduino, Innovative Systems Design and Engineering, 6:1 (2015) 106-114.

[6] Study of Arduino Controlled Robotic System K. M. Merlin Ruby1 , F. Anne Jenefer2 , D. Vidhya3 Assistant Professor, Department of Electronics and Communication Engineering, Panimalar Engineering College, Chennai, India.

[7] Indoor Localization Using Bluetooth-LE Beacons,Sudarshan S. Chawa the School of Computing and Information Science; University of Maine; Orono, Maine 04469-5711, USAchaw@eip10.org.