International Journal of Advanced Culture Technology Vol.7 No.4 313-320 (2019)
DOI 10.17703/IJACT.2019.7.4.313

IJACT 19-12-39

An Implementation of the path-finding algorithm for TurtleBot 2 based on low-cost embedded hardware

[1]Onesphore Ingabire, [2]Minyoung Kim, [3]Jaeung Lee, [4]Jong-wook Jang*

[1]Graduate student, Department of Computer Engineering, Dong-Eui University, Korea
[2]Researcher, Research Institute of ICT Fusion and Convergence, Dong-Eui University, Korea
[3]Graduate student, Department of Computer Engineering, Dong-Eui University, Korea
[4]Professor, Department of Computer Engineering, Dong-Eui University, Korea
ingabireonesphore10@gmail.com, kmyco@deu.ac.kr, marin1356@naver.com, jwjang@deu.ac.kr

Abstract

Nowadays, as the availability of tiny, low-cost microcomputer increases at a high level, mobile robots are experiencing remarkable enhancements in hardware design, software performance, and connectivity advancements. In order to control Turtlebot 2, several algorithms have been developed using the Robot Operating System(ROS). However, ROS requires to be run on a high-cost computer which increases the hardware cost and the power consumption to the robot. Therefore, design an algorithm based on low-cost hardware is the most innovative way to reduce the unnecessary costs of the hardware, to increase the performance, and to decrease the power consumed by the computer on the robot. In this paper, we present a path-finding algorithm for TurtleBot 2 based on low-cost hardware. We implemented the algorithm using Raspberry pi, Windows 10 IoT core, and RPLIDAR A2. Firstly, we used Raspberry pi as the alternative to the computer employed to handle ROS and to control the robot. Raspberry pi has the advantages of reducing the hardware cost and the energy consumed by the computer on the robot. Secondly, using RPLIDAR A2 and Windows 10 IoT core which is running on Raspberry pi, we implemented the path-finding algorithm which allows TurtleBot 2 to navigate from the starting point to the destination using the map of the area. In addition, we used C# and Universal Windows Platform to implement the proposed algorithm.

Keywords: *Windows 10 IoT core, UWP, Raspberry pi, RPLIDAR A2M8, TurtleBot 2*

1. INTRODUCTION

As the use of mobile robots in everyday life and in natural environments increases, so does the requirement to build affordable robots that can perform multiple tasks and that can be widely used in different areas at low-cost [1]. TurtleBot 2 is the world's most popular mobile robot made to navigate widely in indoor environments such as offices, homes, restaurants, and so forth. With the Robot Operating System (ROS) open-source software available for TurtleBot 2, various researches have been conducted and TurtleBot 2 has been proved able to accomplish various tasks such as mapping, localization and path planning [2-4].

However, in order to control TurtleBot 2, a high-sized and expensive computer with high power in terms of memory and speed is used to handle ROS. Robot operating system is a set of libraries and tools mainly running

Manuscript received: October 17, 2019/ revised: October 30, 2019 / Accepted: November 11, 2019
Corresponding Author: jwjang@deu.ac.kr
Tel: +82-51-890-1709, Fax: +82-51-890-1724
Professor, Department of Computer Engineering, Dong-Eui University, Korea

on traditional OS (operating system) such as Linux. ROS is thus not made to run on compact OS, whose constrained resources (such as storage and speed) cannot match the requirements of traditional OS [5]. Therefore, using cheap, tiny microcomputer and newer and compact operating system is the most novel way to decrease the cost of the TurtleBot 2 hardware and software while maintaining the performance of the robot to perform various tasks.

In this paper, we present a path-finding algorithm for TurtleBot 2 developed using Raspberry pi [6], Windows 10 IoT core [7], and RPLIDAR A2 [8]. Raspberry pi is preferred for its scalability, flexibility, and reliability. Raspberry pi is used as the replacement of the laptop that is employed to control the TurtleBot 2. It collects the data returned by the RPLIDAR A2 and helps the TurtleBot 2 to make decisions such as move forward, backward, turn clockwise, and turn counterclockwise. RPLIDAR A2 is used to scan its surrounding environment and to generate an outline map of the environment. Using RPLIDAR A2, the Raspberry pi will detect the objects around the TurtleBot 2 and will help it to avoid them. Such a design will not only reduce the hardware cost and power consumption for the robot but will also increase performance as Windows 10 IoT core will be used as a light-weight operating system running on Raspberry pi [7].

Here is the outline of the rest of this paper. The section II focuses on the requirements of hardware and software used in this implementation and in section III, we describe the system design of the proposed algorithm. In section IV, we explain the system implementation and finally, the conclusion and the future work are presented in section VI.

2. REQUIREMENTS

2.1 Hardware requirements

The proposed system consists of a couple of components that work together to allow the robot navigation.

a. RPLIDAR A2: The RPLIDAR A2 is the low cost 360-degree 2D laser scanner engineered by SLAMTEC. The core of RPLIDAR A2 runs clockwise and performs a 360-degree omnidirectional laser scanning for its surrounding environment and then generates an outline map for the environment [9].
b. Raspberry pi: Raspberry pi is a powerful, portable minicomputer. With its cheapest and its built-in WI-FI, Raspberry pi is capable of many things such as building tablets, laptops, phones, robots and so forth [6].
c. TurtleBot 2: The TurtleBot 2 is the low-cost mobile robot developed for research purposes [10]. The process of controlling TurtleBot 2 consists of sending commands and control the wheel motor which makes the robot moving in any direction.
d. Computer: Inter® Core ™ i7-6700 CPU@3.40GHZ, RAM: 16GB; 64 bit of operating system type. The computer is used to implement the UWP (Universal Windows Platform) program that will be deployed on the Raspberry pi and control the RPLIDAR and the TurtleBot 2.
e. USB Cable A-B Male/Male type peripheral which is compatible with most of Raspberry Pi boards. The USB cable is used to connect the Raspberry Pi and both the RPLIDAR A2 and The TurtleBot 2.

2.1 Software requirements

In order to allow communication between the computer and the Raspberry pi, two main software have been used in this implementation.

a. Windows 10: Windows 10 is the latest version of the Microsoft Windows operating system released in 2015 and follows Windows 8 [11]. In this project, windows 10 is running on the computer and allows the implementation and the deployment of the program controller to the Raspberry Pi.
b. Windows 10 IoT core: Windows 10 IoT Core is a light-weight version of Windows 10 optimized to run on small devices that have no display and it is compatible with the following embedded boards: Raspberry Pi 2, Raspberry Pi 3, DragonBoard 410c and MinnowBoard MAX [12]. In this study, windows 10 IoT core is used on Raspberry pi as an operating system.

3. SYSTEM DESIGN

In this section, we explain the system architecture with detailed description of how specified requirements work together to perform the results we expect. Below is the system design and the description of each part as marked (1), (2), (3), and (4).

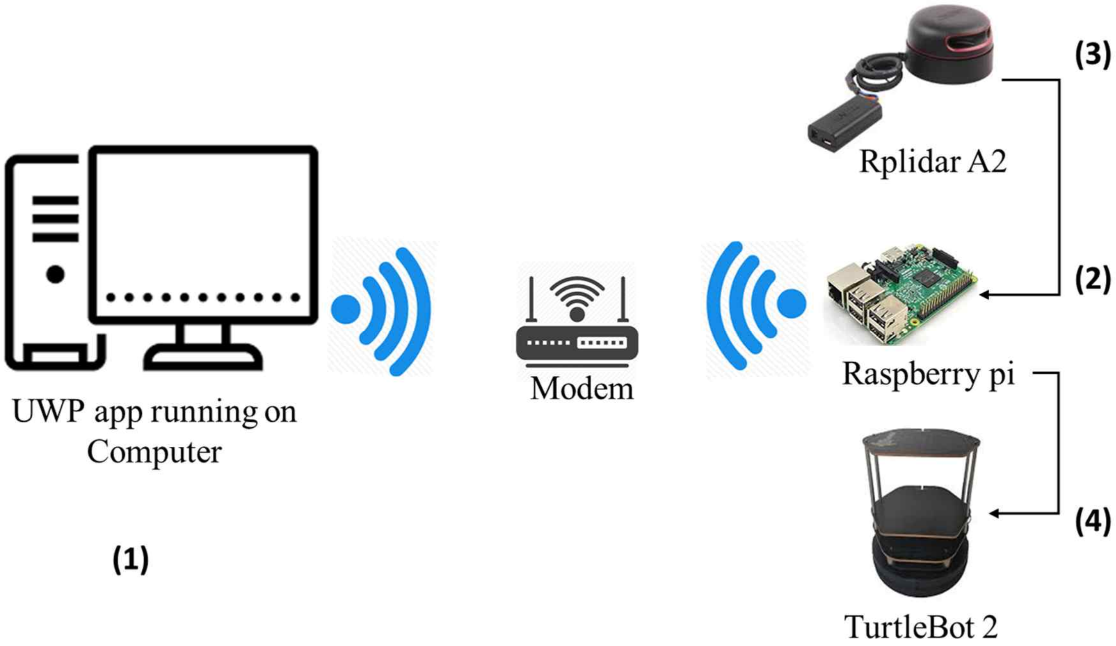


Figure 1. System design of the proposed system

(1) According to Figure 1, the program controller is implemented on a computer. Microsoft Visual Studio has been used as the integrated development environment (IDE) to develop the program. In addition, Universal Windows Platform (UWP) supported by Windows 10 IoT core has been employed to implement and to deploy the program on Raspberry pi via TCP/IP protocol.

(2) Raspberry Pi mounted on TurtleBot 2 is considered as the heart of the system. It is a cheap credit-card sized computer with 1.2 GHZ quad-core ARM CortexA53 CPU and a Broadcom BCM2837 SoC. It has 4 USB 3.0 ports and 802.11n wireless LAN. Instead of using a high-cost computer on the robot, Raspberry pi will take care of all the communication protocols between devices. Raspberry pi is communicating with the computer using TCP/IP protocol and exchange data with both the RPLIDAR A2 and TurtleBot 2 via Serial communication. In addition, Raspberry pi and the computer must be on the same wireless connection.

(3) RPLIDAR A2 sensor scanner is a low-cost 2D laser scanner solution developed to work in all kinds of indoor and outdoor environments without direct sunlight exposure. It uses the triangulation ranging principle and is based on high-speed acquisition and processing hardware [13]. RPLIDAR A2 communicates with the Raspberry pi via serial communication. Using C# and UWP program, RPLIDAR A2 returns a tuple type measurement containing the quality (intensity of the reflected laser), the angle (the angle of measurement) and the distance (distance between the object and the center of rotation of the RPLIDAR A2). With these measurements, the Raspberry pi can help the robot to make decisions when is facing the obstacles.

(4) TurtleBot 2 is controlled by receiving the frame from the Raspberry pi in an asynchronous serial communication at a speed of 115200bit/s. The frame is divided into four parts: the header (2 bytes with fixed values 0xAA and 0x55), the length (1byte which is the size of the payload), the payload (divided into sub-payload with a different number of bytes), and the Checksum to check the transmission (XOR'ed byte stream). To tell the robot to move, one frame is sent at time and allows the robot to move forward, backward, turn clockwise, or counterclockwise. In addition, TurtleBot 2 is made with a high capacity lithium-Ion battery

(4400mAh) so that it can provide the necessary power to the devices mounted on it and can last longer without power from a fixed point.

4. SYSTEM IMPLEMENTATION

The purpose of the implementation is to allow the TurtleBot 2 to navigate in an indoor environment from the starting point to the destination avoiding obstacles on his path. To do that, the computer and the Raspberry pi must be on the same WIFI. The Windows 10 IoT core which is a windows version made for smaller devices will be running on Raspberry pi and will allow communication between the computer and Raspberry pi. The Universal Windows Platform (UWP) app is implemented using the C# programming language and will be deployed to the Raspberry pi.

In fact, when the program is deployed to the Raspberry pi, it initializes the map, the RPLIDAR A2, and the TurtleBot 2 program. The RPLIDAR A2 will start scanning and will return the distance between the center of rotation of the RPLIDAR A2 and the objects. The data will be used to avoid obstacles and will be sent to the client computer for localization of the TurtleBot 2. Then, using the map coordinates, the program will set the starting point and the destination location. Using the library RogueSharp for pathfinding algorithm provided by Microsoft Visual Studio, the program will get the direction from the starting point to the destination by taking the shortest path and will give the instructions of navigation according to the map. For instance, the program will recommend moving 4 meters from the starting point, turn left 45 degrees, move 1.5 meters, and so forth until the destination. The robot will know in real-time how long to move and will stop after reaching the destination. The RPLIDAR A2 will keep tracking the distance between the robot and the object and will give the recommendation to adjust or to stop the robot at a given distance.

As mentioned above, the TurtleBot 2 will be navigating in an indoor area (laboratory) from one point to another by avoiding the objects on his path. Below is the grid map of the laboratory area, the possible nodes and the connection (distance) between connecting nodes.

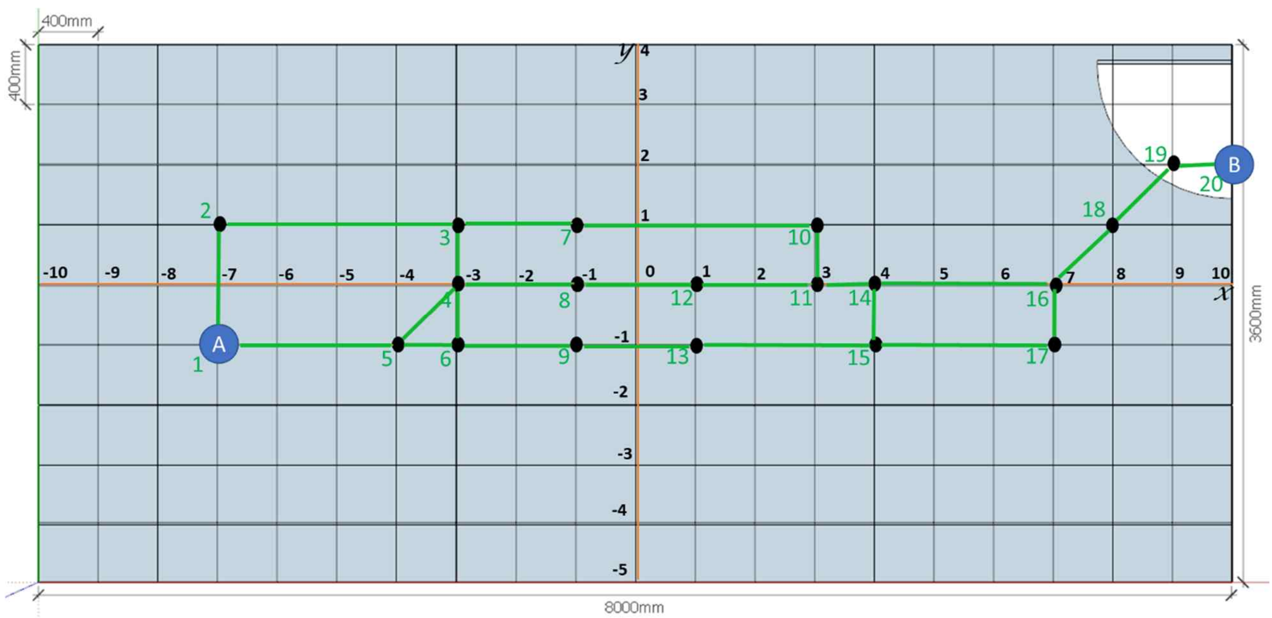


Figure 2. Grid map of the laboratory area

Figure 2 presents the laboratory area where the experimentations will take place. On the map, point A is the starting point and point B is the destination. The points in black are nodes and the points in green are connections between nodes. Each connection represents the cost (distance in meter) from one node to another. The coordinates of the nodes and the cost will be used to find the shortest path from point A to point B using the path-finding algorithm. In the result section, the shortest path according to the algorithm has been presented.

Below is the flowchart showing the process from when the program controller is deployed to the Raspberry pi.

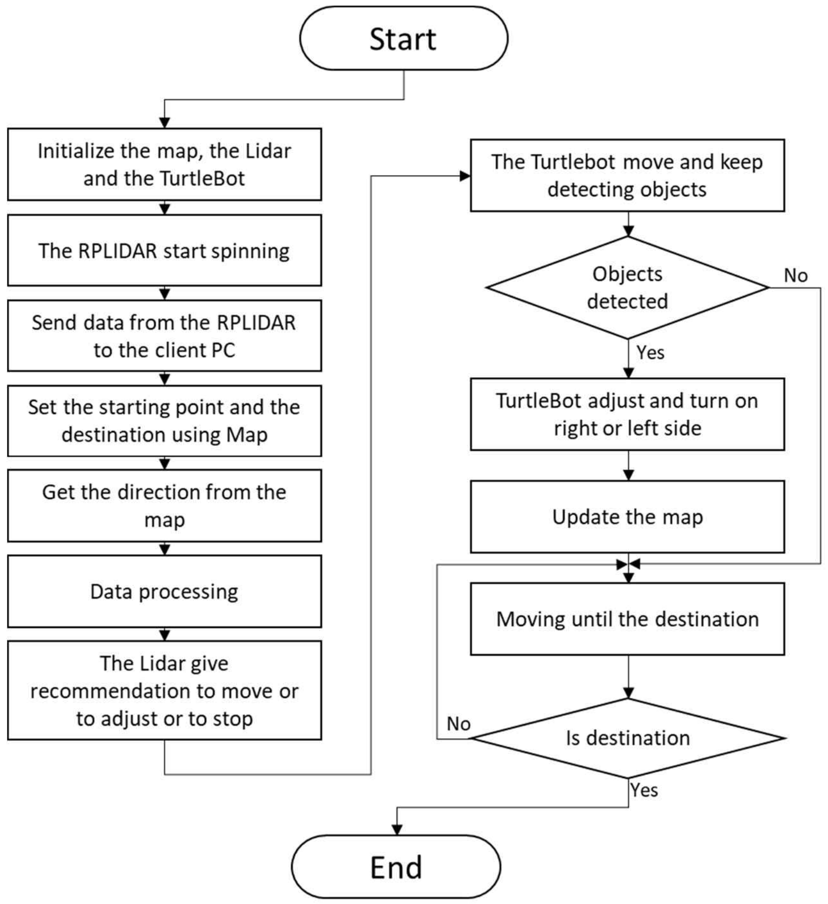


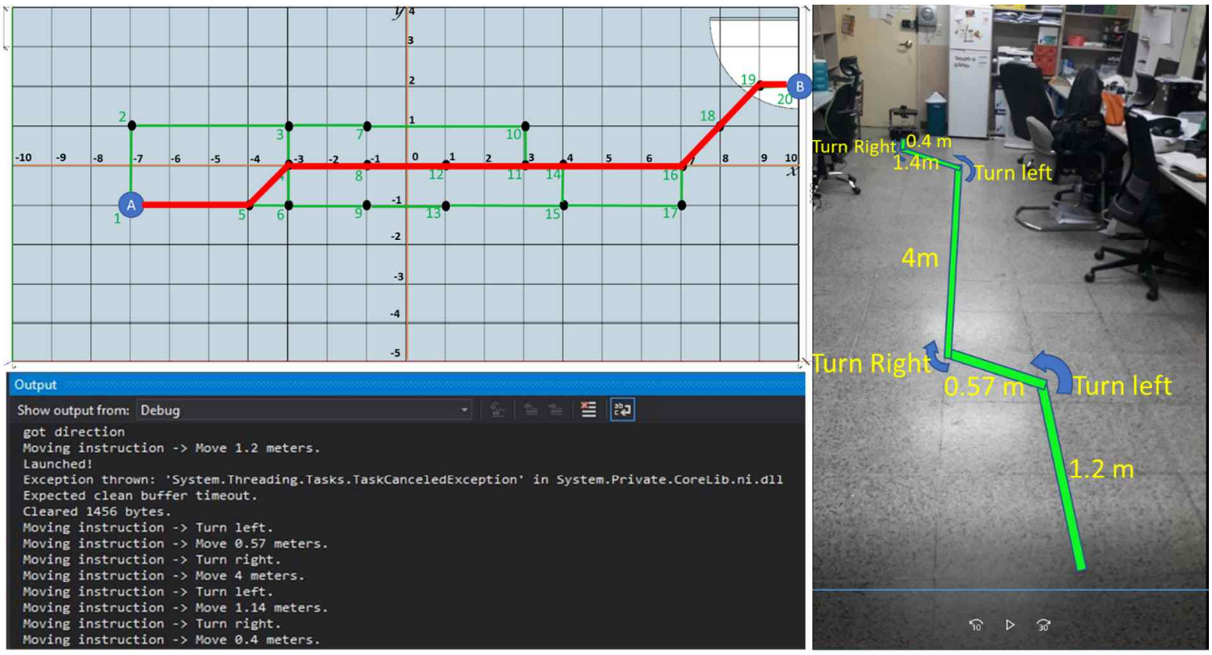
Figure 3. Flowchart of the program controller

This flowchart (Figure 3) explains the process that will take place in the program from when the program is launched until the robot get the destination. In order to start the process, the program controller which is implemented on computer will be deployed to the Raspberry pi. It will initialize the map, the RPLIDAR A2, and the TurtleBot 2. The RPLIDAR A2 will start scanning and the program will send asynchronously the sensor data to the Computer for visualization. Using the map, the program will set the starting point and the destination and will get the direction (shortest path). After analyzing the data from the RPLIDAR A2, the program will give recommendation to move, to adjust or to stop. The TurtleBot 2 will move according to the shortest path and keep detecting the object. If the obstacle is detected, the TurtleBot 2 will adjust to avoid it and the map is updated. Then the TurtleBot will navigate until the destination.

5. RESULTS

The main result of this implementation is that a low-cost microcomputer Raspberry pi has been proved able to allow the TurtleBot 2 to navigate from the starting point to the destination using the path-finding algorithm. In addition, a computer (laptop) consumes about 20 times energy more than Raspberry pi [1]. Therefore, we can confirm that with Raspberry pi, a low-cost hardware has been designed and the energy consumption has been decreased 20 times compared to the energy consumed by the computer (laptop mounted on the robot) on the robot. Below are the results of the navigation of the robot and the graphical view implemented using the data from the RPLIDAR A2.

(1)



(2) (3)

Figure 4. Test of TurtleBot 2 moving from point A to point B in Laboratory area

According to Figure 4, the point (1) is the map of the area made to allow the path-finding algorithm to determine the shortest path and help the TurtleBot 2 execute the instructions when moving. On the map, the point A is the starting point and point B is the destination. The points in black are possible nodes and the distance between nodes are cost (distance in meters) from one node to another. The red line is the shortest path according to the path-finding algorithm. When the program will be launched, the robot must be placed at the starting point. Then, the program will give navigation instructions according to the direction and the distance between the nodes of the shortest path as can be seen in (2) of Figure 4. As shown in (3), that is the laboratory area where the experimentation took place. Using the instructions provided by the program controller, the TurtleBot 2 will know the distance to move and at which angle to turn and will execute the navigation instructions following the path found from the starting point to the destination. After each instruction, Raspberry will help the TurtleBot 2 to make decision by sending the frame which will control the motor according to the instructions.

Below is the graphical view where the user can localize the navigation of the TurtleBot 2. In fact, this program includes the client server architecture. The raspberry pi is considered as the server and the computer as the client. After the program controller is deployed on Raspberry pi, it will listen for the client connection and as soon as the client is connected, it will send the data from the RPLIDAR A2 to the client computer. Then, the data will be used asynchronously to plot the points in the graphic where the user can see the objects around the robot. Below is the result:

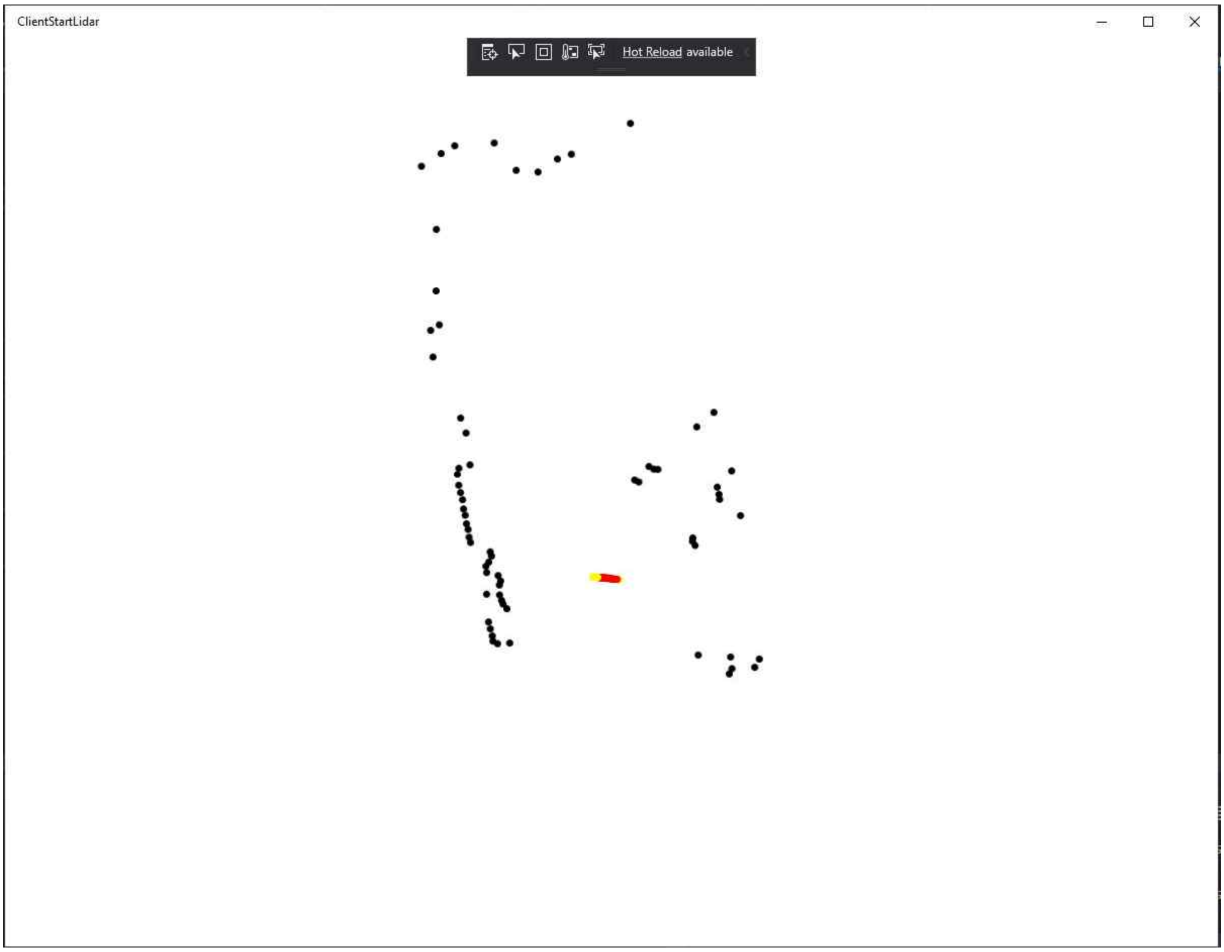


Figure 5. Graphical view showing objects around the robot

According to Figure 5, when the robot is facing an object at a certain distance, the graphical view will show the object in red color. In that case, the robot will be recommended to adjusted or to stop.

6. CONCLUSION

In this paper, we have explained the implementation of the path-finding algorithm for TurtleBot 2 based on low-cost embedded hardware. The presented algorithm has been implemented using Windows 10 IoT core which is running on Raspberry pi. The Raspberry pi is used as a low-cost small device that acts as the heart of the implementation. It allows communications between the computer, the RPLIDAR A2, and the TurtleBot 2. Raspberry pi has been proved able to control the TurtleBot 2 and to decrease the power consumption on the robot. The UWP app has been used to implement the program controller that will hold the communications between the devices and will allow the robot to navigate from the starting point to the destination. In future work, we will allow the robot to navigate and adapt to a new environment. In addition, TurtleBot 2 also can be designed in such a way it can navigate outdoor using Google map and low-cost devices.

ACKNOWLEDGEMENT

The research was supported by the BB21+ project in 2019.

REFERENCES

[1] Hamza Aagela IEEE studet Member, Maha Al-Nesf, and Violeta Holmes, "An Asus_xtion_probased indoor MAPPING using a Raspberry Pi with Turtlebot Robot Turtlebot Robot," *2017 23rd International Conference on Automation and Computing (ICAC),* Huddersfield, UK, DOI: 10.23919/IConAC.2017.8082023, 7-8 September 2017.

[2] Chuantang Xiong and Xu Zhang, "An exclusive human-robot interaction method on the TurtleBot platform," *International Conference on Robotics and Biomimetics (ROBIO),* Shenzhen, China, DOI:10.1109/ROBIO.2013.6739662, 12-14 Dec. 2013

[3] Arbnor Pajaziti and Petrit Avdullahu, "SLAM – Map Building and Navigation via ROS#," *International Conference advanced Technology & Sciences (ICAT*), Antalya, Turkey, pp. 71-75, August 12-15, 2014

[4] TurtleBot great for driving food around the house. https://www.cnet.com/news/turtlebot-great-for-driving-food-around-the-house/

[5] Hauke Petersen, Cedric Adjih, Oliver Hahm, and Emmanuel Baccelli, "Demo: IoT Meets Robotics - First Steps, RIOT Car, and Perspectives" *International Conference on Embedded Wireless Systems and Networks (EWSN),* Graz, Austria, 15-17 February 2016.

[6] What is the Raspberry Pi 3? Everything you need to know about the tiny, low-cost computer. https://www.zdnet.com/article/what-is-the-raspberry-pi-3-everything-you-need-to-know-about-the-tiny-low-cost-computer/

[7] Windows for Internet of Things. https://developer.microsoft.com/en-us/windows/iot

[8] RPLIDAR A2. https://www.slamtec.com/en/Lidar/A2

[9] RPLIDAR Laser range scanner. https://www.slamtec.com/en/Lidar/A2

[10] TurtleBot Tutorials. http://www.clearpathrobotics.com/assets/guides/turtlebot/

[11] Windows 10. https://techterms.com/definition/windows_10

[12] Ioana-Maria CULIC and Alexandru RADOVICI, "Extending the Wyliodrin Platform for Windows 10 Core," *2016 15th RoEduNet Conference: Networking in Education and Research,* DOI: 10.1109/RoEduNet.2016.7753246, 7-9 Sept. 2016

[13] RPLIDAR datasheet. https://cdn.sparkfun.com/assets/e/a/f/9/8/LD208_SLAMTEC_rplidar_datasheet_A2M8_v1.0_en.pdf