

Proceeding paper

Using Ultrawideband Technology to Control a Car to Reach Its Destination [†]

Septimiu Mischie^{1,*} and Ionela Alina Munteanu ²

¹ Politehnica University Timisoara; septimiu.mischie@upt.ro

² Politehnica University Timisoara; ionela.munteanu@student.upt.ro

* Correspondence: septimiu.mischie@upt.ro

Abstract: This paper presents a method that enables a small car to reach its destination in an indoor environment using ultrawideband technology. For this purpose, we used four anchors and one tag. The tag can move because it is fixed on the car. The anchors and the tag are DWM1001 modules. The configuration of the five DW1001 modules is performed through a smartphone application provided by the manufacturer. The car has been built by us. The controller of the car is a Raspberry Pi microcomputer which controls the driver of the car DC motors by generating PWM signals. In addition, Raspberry Pi communicates with the tag by UART interface. Thus, our goal was to control the car to reach a certain destination (x_d,y_d) requested by the user via Wi-Fi. Because the car hosts the tag, it continuously updates its position according to the information received from the four anchors. We implemented two scenarios, considering the initial (x,y) position of the car. In the first scenario, which is simpler, the car moved to its destination either along the x-axis or along the y-axis. In the second scenario, the car moved along one of the axes and then turns 90 degrees to reach its destination. Experimental results that show the precision of positioning system are presented.

Keywords: ultrawideband; indoor localization; anchor; tag; Raspberry Pi; H-bridge

1. Introduction

Indoor localization is an important topic today [1], nonetheless GPS technology cannot be used in indoor locations. Therefore, other methods are used such as inertial sensors [2]. They have the advantage that no infrastructure is required yet the results are not reliable due to different errors of these sensors. Other methods are based on wireless technologies such as Wi-Fi [3], BLE [4] and ultrawideband [5–8]. Ultrawideband means transmission of very short impulses that have different forms, for instance cosine rising pulse. The frequency of these impulses is 499.9 MHz. This signal is then modulated by a frequency that corresponds to the desired channel, for instance 3494.4 MHz [5]. In this way systems with low energy consumption and immune to interferences can be created. Such a system uses fixed devices called anchors and at least one mobile device called a tag. The tag computes its position coordinates according to the information from anchors. There are two methods to achieve this goal. In the first method, called two-way ranging, the tag is active by sending and receiving data to and from the anchors. In this way the distance from the tag to each anchor is computed and then the tag position coordinates can be determined using trilateration [5]. This method is accurate, but it cannot be used for more tags [5]. In the second method the anchors are synchronized, and they send data at the same time. Only the tag receives these data messages. For instance, if two anchors send data, the tag uses the difference between the two times of arrival to compute the difference between the distances to the two anchors. Then its position coordinates can be determined. This method can be used for a larger number of tags [5].

Citation: Mischie, S.; Munteanu, I.A.

Using ultrawideband technology to control a car to reach its destination.

2023, 5, x.

<https://doi.org/10.3390/xxxxx>

Published: date



Copyright: © 2023 by the authors.

Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

There are more commercially available systems that use UWB for indoor localization purposes. Among the most well-known ones are those produced by Ubisense, BeSpooon and DecaWave [7]. We used the latest modules produced by DecaWave [9] to implement a system where a mobile device reaches a location requested by the user.

2. The structure of the proposed system

2.1. The hardware structure

Figure 1 presents the structure of the proposed system. There are four fixed nodes (anchors) and one mobile node (tag). These five nodes are based on DWM 1001 modules from DecaWave.

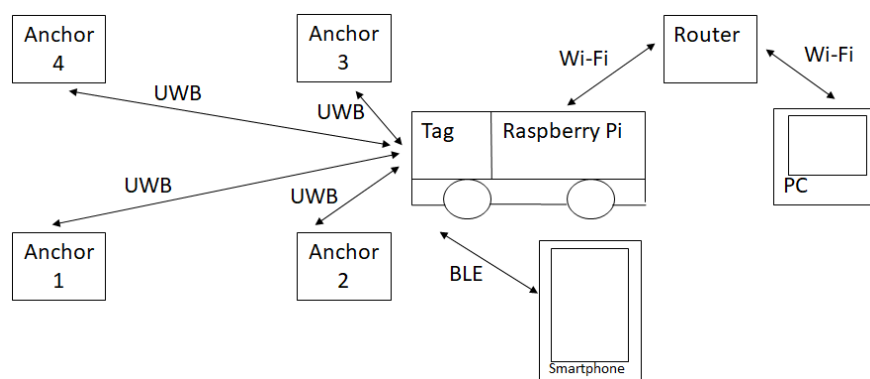


Figure 1. The structure of the proposed system.

The power of each of the four anchors is taken from a 3.7 V DC rechargeable battery. The tag is placed on a small car controlled by a Raspberry Pi microcomputer to implement the mobile node. The tag receives in real time the information from the four anchors and continuously computes its position. The position is sent to Raspberry Pi by an UART interface. The user connects to the Raspberry Pi from a Personal Computer (PC) via Wi-Fi using putty application. Then it introduces the destination position. As a consequence, the Raspberry Pi controls the electronic driver of the car to move it to the destination. The user will receive the positioning result on the PC after the car stops.

The hardware structure of the car, including the tag is presented in Figure 2.

Raspberry Pi is powered by a 5 V DC voltage that is generated by the DC-DC converter. This receives the DC voltage from an ensemble of two rechargeable batteries of 3.7 V DC which are connected in series at its input.

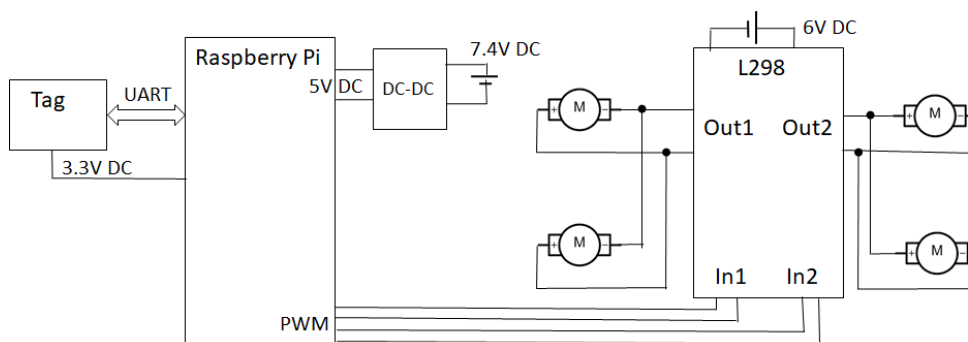


Figure 2. The hardware structure of the car.

Raspberry Pi is connected by UART interface to the DWM1001 module having the function of tag. The power of DWM1001 module of 3.3 VDC is taken from Raspberry Pi.

Raspberry Pi also controls the movement of the car. For this purpose, it is connected by Pulse Width Modulation (PWM) signals to an L298 module which in turn is connected

to the four DC motors of the car. L298 module contains an H-bridge circuit and is powered to a 6V DC from four 1.5 V batteries. The DC motors on one side of the car are connected to the same output of the L298 module and this means that they move in the same way. There are four ways to move the car: forward, back, turn left and turn right. To turn, the motors on the two sides rotate in opposite directions. To get a 90-degree rotation, the rotation of the motors must last a certain time. If the batteries start to run out, the velocity of the motors decreases, and the rotation angle can be lower than 90-degree. This is a problem that can be overcome by increasing the rotation time of the motors.

2.2. The software structure

The embedded firmware from the manufacturer [9] is first loaded in each of the five DWM 1001 modules. The configuration of the five modules as tag or anchor is implemented by the Decawave DRTL Manager smartphone application [9]. It communicates with the modules via Bluetooth Low Energy (BLE), as in Figure 1 is presented. In this way for instance the x, y, z coordinate of each anchor can be established and which of the four anchors has the function of the initiator [9].

The tag contains embedded software that computes its position using data received from the four tags according to the two-way ranging method [5]. Also, the position of the tag is sent to the UART port. The manufacturer provides a C program for Raspberry Pi that accesses the UART port to get the tag position. This program was adopted by the authors and modified to save the current position on a file on the SD card.

The goal of the application is to move the car to a destination position chosen by the user. Thus, the implemented algorithm reads the destination position, compares it with the current position and sends commands to the L298 module to move the car. When the current position exceeds the destination position, the algorithm stops the car.

The user can see the current position of the car (x_c, y_c) and can introduce the destination position (x_d, y_d) . There are two scenarios for moving the car. In the first scenario, either $x_d = x_c$ or $y_d = y_c$, so the car will move to the destination either along the y -axis or along the x -axis. In the second scenario, x_d is different to x_c and y_d is different to y_c , so the car moves along one of the axes and then turns 90 degrees to reach its destination.

This algorithm is implemented in Python. It continuously reads the current position that the C program writes on SD Card. However, the moment when the car will be stopped depends on the rate of update the current position. Thus, the successive positions of the car are established by the resolution of $v\Delta t$, where v is the car velocity and Δt is the update rate. Thus $\Delta t = 0.5$ s and for a velocity of $v = 20$ cm/sec it follows a resolution of 10 cm that also represent the maximum value of the error of positioning. As we said before, if the batteries start to run out, the velocity decreases. Thus, the resolution decreases too, that is an advantage.

3. Results

The setup of Figure 1 is used to obtain the experimental results. The anchors are placed in the following (x, y, z) positions, in cm: A1(0,0,65), A2(116,0,65), A3(116,265,65), A4(0,265,65). There are no obstacles in this rectangular setup, thus the Line-of-Sight is considered. Figure 3 presents a photo of the setup with the car and one of the anchors.

3.1. Static test

The purpose of the static test was to verify the stability and the precision of the DWM1001 modules when the tag is fixed. For this purpose, the car was kept still for 50 seconds in 12 distinct positions. Then, the corresponding 100 position coordinates generated by the DWM1001 tag for each position, in millimeters (mm) are acquired. In each case the position coordinates were measured with a tape measure to obtain the true values.

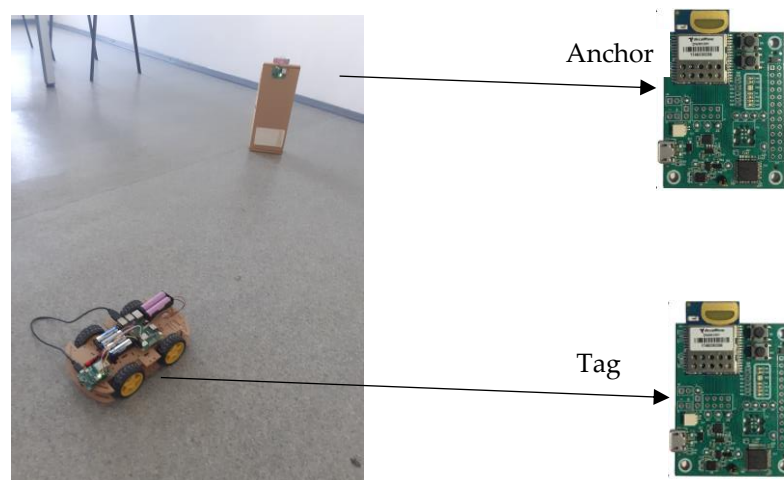


Figure 3. The experimental setup.

Figure 4 presents the variations of position coordinates generated by the tag for two of the positions. The true values were $x = 1160$ mm and $y = 800$ mm in left panel, and respectively, $x = 600$ mm and $y = 2000$ mm in right panel. It can be seen that there was a variation of about 100 mm for x coordinate and about 50 mm for y coordinate. The quality factor, qf , which represents an indicator of quality of estimation [6] is around 75 percent in both examples.

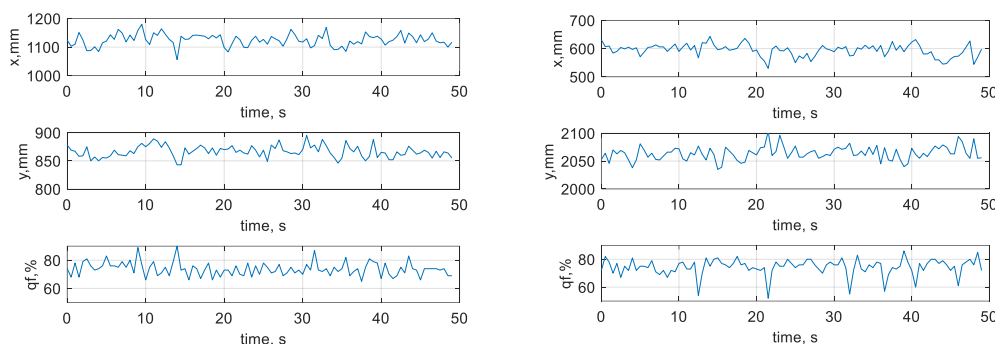


Figure 4. Variations of coordinates during stability test.

Table 1 presents the following results for each of the true values of x and y coordinate: the mean of the 100 position coordinates, the error Δx (Δy) as the difference between the true value and the mean, and the standard deviation σx (σy) of the 100 position coordinates. To evaluate the precision, we considered the error. Except in a few cases the error agrees with the range of ± 100 mm guaranteed by the manufacturer. To evaluate the stability, we computed the standard deviation. Generally, the standard deviation was lower for y coordinate.

3.2. Dynamic test

The purpose of the dynamic test was to verify the functionality of the two scenarios that allow movement of the car to the destination chosen by the user. We start with the first scenario. Thus, for movement along x axis, Table 2 presents the positioning result coordinate, the destination coordinate and the positioning error computed as the difference between them. The initial coordinate is presented too. Table 3 presents the same results for movement along the y axis. For both axes, the first three rows contain the results during forward movement, and the last three rows represent backward movement. It follows that the positioning result coordinate is higher than the destination coordinate for

forward movement while this coordinate is lower than the destination coordinate for a backward movement. These are expected as it was stated in the previous section. With an exception, the absolute value of the error is at most 100 mm so in section 2 we anticipated. On the other hand, by inspecting the values of Table 2 and 3, some of the initial and destination coordinates are outside of the rectangle of the four anchors. Thus, the movement of the mobile node is not limited to this rectangle.

Table 1. The results of the static test, in mm.

x	mean_x	Δx	σx	y	mean_y	Δy	σy
0	-402	402	42	300	271	29	20
600	468	132	17	300	257	43	19
1160	1232	-72	27	300	310	-10	12
0	76	-76	17	800	552	248	16
600	558	42	31	800	801	-1	16
1160	1124	36	21	800	866	-66	10
0	102	-102	59	1400	1230	170	24
600	483	117	21	1400	1240	160	11
1160	1072	88	36	1400	1350	50	16
0	73	-73	28	2000	1970	30	16
600	594	60	21	2000	2060	-60	12
1160	1140	2	22	2000	2010	-10	13

Table 2. The results of the dynamic test for moving along x axis. All values are in mm.

Initial coordinate	Destination coordinate	Positioning result coordinate	Error
x = -1923	x = 800	x = 841	41
x = -1147	x = 1000	x = 1077	77
x = 62	x = 2200	x = 2292	92
x = 3103	x = 1000	x = 953	-47
x = 1083	x = -1000	x = -1036	-36
x = 1167	x = -1800	x = -1830	-30

Table 3. The results of the dynamic test for moving along y axis. All values are in mm.

Initial coordinate	Destination coordinate	Positioning result coordinate	Error
y = 205	y = 1000	y = 1107	7
y = -177	y = 2300	y = 2302	2
y = 1181	y = 2900	y = 2943	43
y = 2505	y = 500	y = 367	-133
y = 3014	y = 1000	y = 953	-47
y = 2211	y = -100	y = -115	-15

Figure 5 presents two trajectories obtained by the second scenario. Thus, in the left panel the car has the initial position (2359, 864). The destination position is (600, 2200). It moves back along x-axis until x coordinate is 540. Then it turns left by 90 degrees and finally moves forward to the positioning result which was (405, 2288). In the right panel, the car has the initial position (2187, 1602). The destination position is (-200, 0). The car starts moving back along the x-axis until x is -279. Then it turns right by 90 degrees and finally goes forward to the positioning result which was (-208, -4). Even if the value of x coordinate is practically constant after the 90 degrees rotation, the value generated by the tag is changed similarly to the static test. This can be seen in Figure 5.

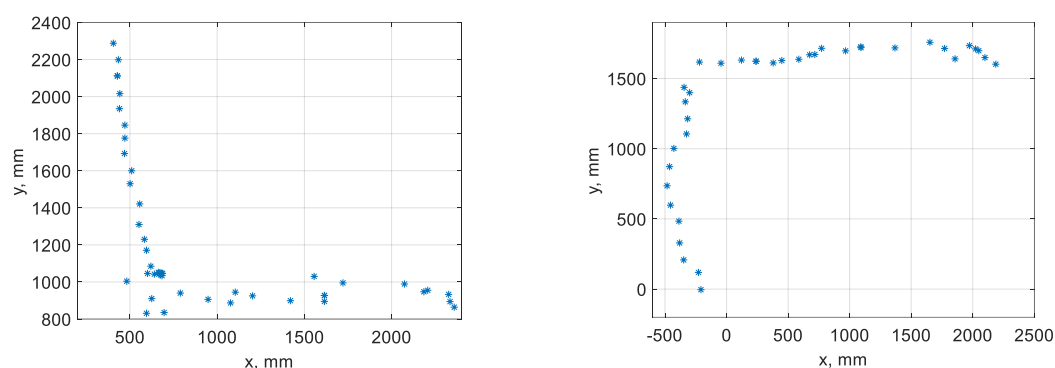


Figure 5. Two trajectories of the tag during the second scenario.

4. Conclusions

This paper has presented the structure of a small car that can be controlled by Wi-Fi to move to a position chosen by the user. The Raspberry Pi microcomputer is the brain of this system because it reads the position chosen by the user, compares it with the current position received from the tag and commands the electronic part of the car until the current position is close to the chosen position.

The stability and the precision of the UWB indoor localization has been studied. It was found that the precision of ± 10 cm claimed by the manufacturer is exceeded in some cases and the results vary even if one of the coordinates are fixed. However, these disadvantages do not affect the goal of the paper as the car was able to arrive at its destination.

For future work, a gyroscope that measures the value of the rotation angle should be introduced. In this way the 90-degree rotation can be precisely controlled, and, in addition other rotation angles can be executed to allow movement between the two points in straight line.

Author Contributions: Conceptualization, S.M. and I.A.M.; methodology, S.M.; software, I.A.M.; validation, S.M. and I.A.M.; investigation, S.M. and I.A.M.; writing—original draft preparation, S.M. and I.A.M.; writing—review and editing, S.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

References

1. Xiaomin Zhu; Jianjun Yi; Junyi Cheng; Liang He. Adapted Error Map Based Mobile Robot UWB Indoor Positioning. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 6336–6350.
2. Xinyu Hou; Jeroen Bergmann. Pedestrian Dead Reckoning with Wearable Sensors: A Systematic Review. *IEEE SENSORS JOURNAL* **2021**, *21*, 143–152.
3. Weixing Xue; Weining Qiu; Xianghong Hua; Kegen Yu. Improved Wi-Fi RSSI Measurement for Indoor Localization, *IEEE SENSORS JOURNAL* **2017**, *17*, 2224–2230.
4. Faheem Zafari; Athanasios Gkelias; Kin K. Leung. A Survey of Indoor Localization Systems and Technologies, *IEEE COMMUNICATIONS SURVEYS & TUTORIALS* **2019**, *21*, 2568–2599.
5. Michael David Hamer. Scalable Localization and Coordination of Robot Swarms. Ph.D thesis, Institute for Dynamic Systems and Control, ETH Zurich, Zürich, Switzerland, 2019.
6. Antonio R. Jimenez; Fernando Seco. Improving the Accuracy of Decawave's UWB MDEK1001. *Sensors MDPI* **2021**, *21*, 1787.
7. Antonio Ramón Jiménez Ruiz; Fernando Seco Granja. Comparing Ubisense, BeSpooon, and DecaWave UWB Location Systems: Indoor Performance Analysis, *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2106–2117.
8. Juri Sidorenko; Volker Schaltz, Norbert Scherer-Negenborn, Michael Arens and Urs Hugentolber. Error Correction for Ultrawide Ranging. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9037–9047.
9. DWM1001-DEV - Qorvo. Available online: www.qorvo.com/products/p/DWM1001-DEV (accessed on 3 September 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.