International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2023 | Issue 57 | Volume 15 | Number 4 | Pages 60-66

PERFORMANCE COMPARISION BETWEEN TWO MICROCONTORLLERS BASED GPS FOR PRECISION IN DETERMING VEHICLES LOCATIONS

Z.K. Farej A.N. Ismail

Department of Computer Technology Engineering, Technical Engineering College of Mosul, Northern Technical University, Mosul, Iraq, drziyad.farej@ntu.edu.iq, ahmadna1985@ntu.edu.iq

Abstract- Innovative technology known as a global satellite system (GPS) tracker allows for the monitoring of a vehicle's whereabouts. The GPS receiver receives position data in the form of latitude and longitude, number of satellites, altitude, direction, velocity, and time from the satellites. This data is analyzed by the microcontroller, which then displays the results on the display. The proposed technique can be employed in several situations where the required information is only sometimes and irregularly requested. This research proposed a vehicle tracking system that uses the Arduino Uno board and Raspberry-pi-Pico as microcontrollers and global positioning system sensor. Under the suggested two microcontrollers, the performance of the proposed GPS system in determining the precision of vehicle location, is investigated for different speeds at different locations. For Arduino microcontroller, the minimum and maximum location accuracy results are 5.3 and 48.8 meters, while for Raspberry Pi Pico are 4.3 and 25.9 meters at speeds of 0 and 100 Km/h respectively. The results show that the proposed system can operate in any weather and offer more precise real-time object position.

Keywords: Global Positioning System, GPS Accuracy, Real-Time Systems, Better Geographic Coordinate Raspberry Pi Pico.

1. INTRODUCTION

GPS is a satellite navigation system used to locate an object on the ground in an outdoor setting. Today, positioning systems are a crucial component of daily life for people. The advancement of positioning systems is crucial for the growth of human civilizations since they are applicable to a variety of fields, including security, military, logistics, monitoring, tracking, health care, sports, and entertainment [1]. We may characterize GPS as a useful technology that has led to the development of one of the best all-purpose navigational aids to date [2]. The GPS project required the creation and deployment of 18 satellites. Each orbital plane, which consists of six segments separated by 120 degrees, cost about $12 billion at the time. These satellites used to collect and show exact geographic coordinates for places on a map by utilizing GPS, among other things [3]. The GPS receiver, which always has an antenna connected, is a tool used to obtain position data [4].

In addition, the GPS system's multipath error may be decreased by the GPS antenna's high gain and low axial ratio, which is less than 3dB. In addition to its other uses, GPS locates objects in real time and is impervious to bad weather. Figure 1 [7] illustrates some of the various applications for GPS sensors, containing location-based services, scientific research, and government use [5]. The getting of accurate position with saving time and speed information and navigation all these done by using GPS [6].

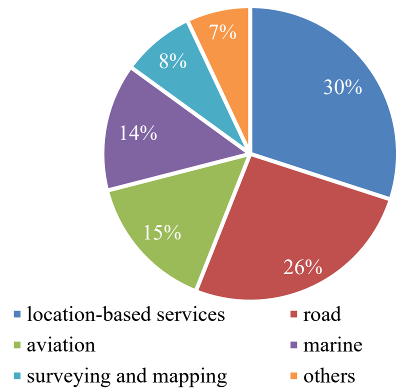


Figure 1. Global GPS market share [7]

In this research, a GPS-based vehicle tracker that uses Arduino and Raspberry Pi Pico microcontrollers to track moving objects remotely through a GPS network, is designed and implemented. Earlier research utilized the internet to track automobiles. With widely accessible materials, this design is affordable. This project is crucial for security and cost savings [5]. This research uses the GPS NEO-6M sensor and two microcontrollers separately to evaluate the performance and accuracy of a GPS system in an automobile at various speeds. The block design and flowchart of the proposed GPS-based vehicle tracking system are shown in Figures 2 and 3.

Both of the angles that are specifically specified on the sphere are latitude and longitude. The angle together with the coordinate system makes up the ability to locate a geographic point on the globe. Latitude is defined in relation to the equatorial reference; as one goes from north to south, the value changes from positive to negative. When referencing the prime meridian, longitude is positive to the east and negative to the west. Without the driver's interference or the use of a technology for continually gathering the GPS coordinates of moving vehicles, tracking enables the base station to follow the vehicle in real time [8].

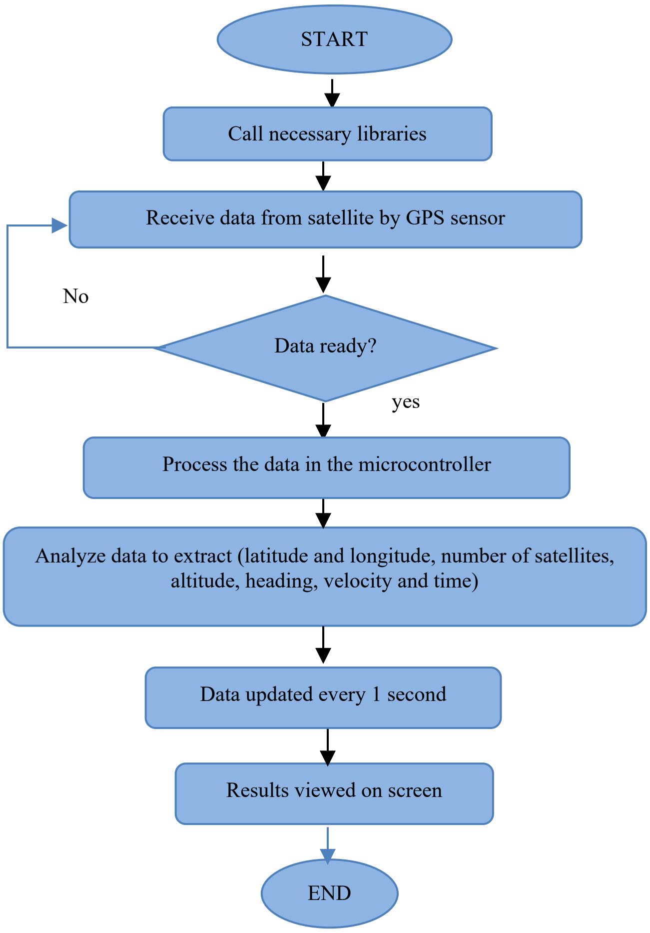


Figure 2. Flowchart of vehicle tracking system based on GPS

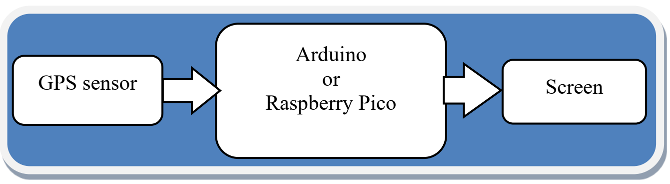


Figure 3. Block diagram of vehicle tracking system based on GPS

Khalid A. utilized a GPS receiver to test system precision in determining eight ground speeds starts from (five to fifty) km/h from the receiver every 10 seconds, the accuracy of GPS ground speed was examined. Less than 7% of the GPS ground speed was inaccurate [9].

Jasper S. et al. evaluated the GPS device's accuracy utilizing three data collecting intervals, four different types of transportation, and real-world environmental conditions. To evaluate the BT-Q1000XT GPS device's accuracy, they chose four routes on various bearings, travelling through various climatic conditions. There were lanes for vehicles, bicycles, and pedestrians on each route. According to their findings, about 50% of the approximately 68,000 GPS points were in 250 cm and 78.7% were within 10 m of the predicted position. The median inaccuracy was also 290 cm.

The median error for travels by foot was 3.9, 2.0 for trips by bicycle, 1.5 for trips by bus, and 0.5 for trips by car. The median inaccuracy varied significantly across the various area types: it was 70 cm in open areas, 260 cm in half-open areas, and 520 cm in urban canyons [10]. To display a car's position on Google Maps, Mohd H. and Mohd F. conceived and created a GPS-based vehicle tracking system. This system utilized an Arduino and NEO6m sensor for coordinate routing and a SIM900A for user connectivity. The system worked well outside but had some problems inside since a roof or other impediment prevents the GPS module from receiving a direct signal from the satellite, which is necessary to derive exact coordinates, the error between the coordinates taken from the application coordinate and real coordinate was 7 meters for outdoor and 30 meters in indoor [11].

Xiaoyun Z. et al. evaluated the accuracy of GPS-based traffic data on the location, speed, and altitude of bicycles, cars, and buses. They found the geographic placement to be accurate, although with a larger inaccuracy than the manufacturer had predicted and a non-zero likelihood of aberrant placing. Altitude readings are unreliable, whereas velocity is somewhat underestimated [12].

Elsyea Adia and et al., created a kid safety gadget with a button that, when hit, instantly sends the scene's position and a photo to a mobile through Telegram. The GPS module and 5MP camera on the Raspberry Pi are used to pinpoint the location and record the incident, respectively. According to the findings, if both the device and mobile are connected to same network, the average communication time for positions is 0.91 seconds. During this time, the expected error in position reading depends on the speed of the car. In addition, if the device and the receiving smartphone are on separate networks, time required to communicate positions is 0.96 seconds. Both circumstances have a 100% picture accuracy rate and a 97.5% location accuracy rate [13].

2. METHODOLOGY

The research method has two parts: hardware part and software part, as follows:

2.1 Hardware Part: Includes

2.1.1 Arduino Uno

An ATmega328P-based microcontroller board. It is an open-source hardware and software platform. Processing the Arduino Uno requires the Arduino Software (IDE) and programming language. It includes six analog inputs, fourteen digital input/output port. Figure 4 [4] showing the Arduino microcontroller [14].

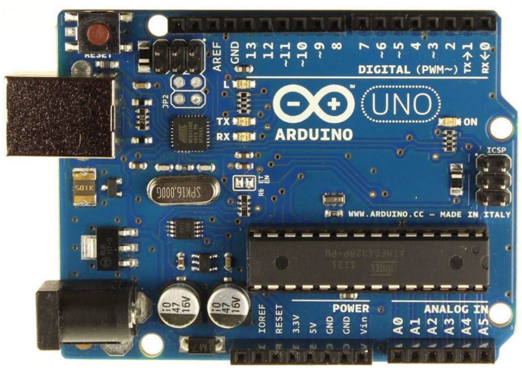


Figure 4. Arduino uno microcontroller [14]

2.1.2. Raspberry Pi Pico

This versatile development microcontroller is affordable with many mimic interfaces like VGA and SD Card, Figure 5 [15].

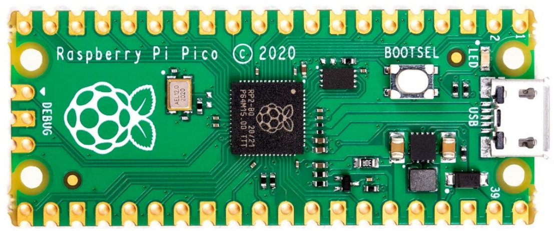


Figure 5. Raspberry Pi Pico [15]

2.1.3. GPS NEO-6M Sensor

The NEO6M GPS chip from uBlox serves as the module's brain. The NEO6M GPS chip has a lot of functions. The chip contains four pins VCC, RX, TX, and GND packed into an area that is smaller than a stamp but large enough to frame as Figure 6 [16].

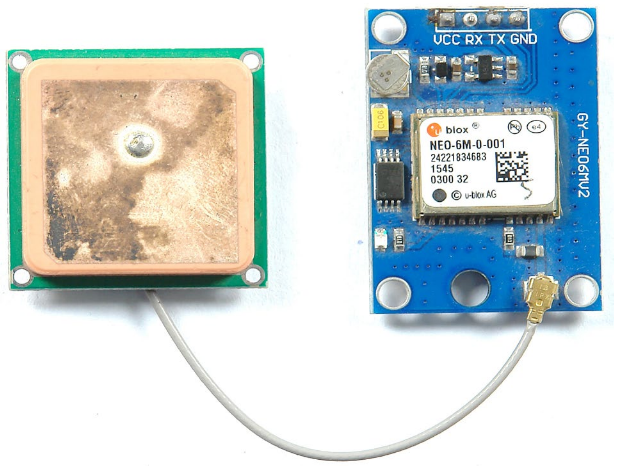


Figure 6. GPS NEO-6M sensor [16]

2.1.4. Battery 9V

2.1.5. I2C OLED Display

It is a single-chip using light-emitting diodes. There are 64 commons and 128 segments in it. As a result, it consumes less power and has fewer external components, such as oscillator, display RAM, and contrast control [17].

2.2. Software Part

2.2.1. Arduino Environment

The Arduino environment has some libraries and software tools that are used in the proposed system and as follows:

1) U8glib.h: A graphics library that supports a variety of monochrome monitors and Arduino boards. It is a font collection featuring a large selection of monospaced and proportional fonts for graphic LCDs and OLEDs. Landscape and portrait modes are supported, and there is a well-defined interface to the device subsystem. [18]

2) Software Serial: On the digital pins of an Arduino board, the Software Serial library enables serial connection by simulating the capabilities with software. Multiple software serial ports with transfer rates of up to 115200 bps are feasible [19].

3) TinyGPS++: is an Arduino library that allows you to analyze data streams that GPS units give. This library offers straightforward and convenient ways to retrieve information from consumer GPS devices, including location, date, time, altitude, speed [20].

4) Displaying Gauge: The Gauge control uses a speedometer-like indicator to display a numerical number. Typically, orders from a linked device set its value. Up to four labels that may be used to show units and status data can be attached to the gauge. Colored bands can be used to mark certain areas of the gauge scale to show target or out-of-range values [21].

5) Software Development: Some of the software that was utilized to create the application. The code will be developed on the open-source Arduino software, which will then be uploaded to an Arduino board. C++ is the programming language used by Arduino.

2.2.2. Raspberry Environment

Raspberry Pi Pico uses python as programming language. It's environment, some libraries and software tools used in the proposed system are as follows:

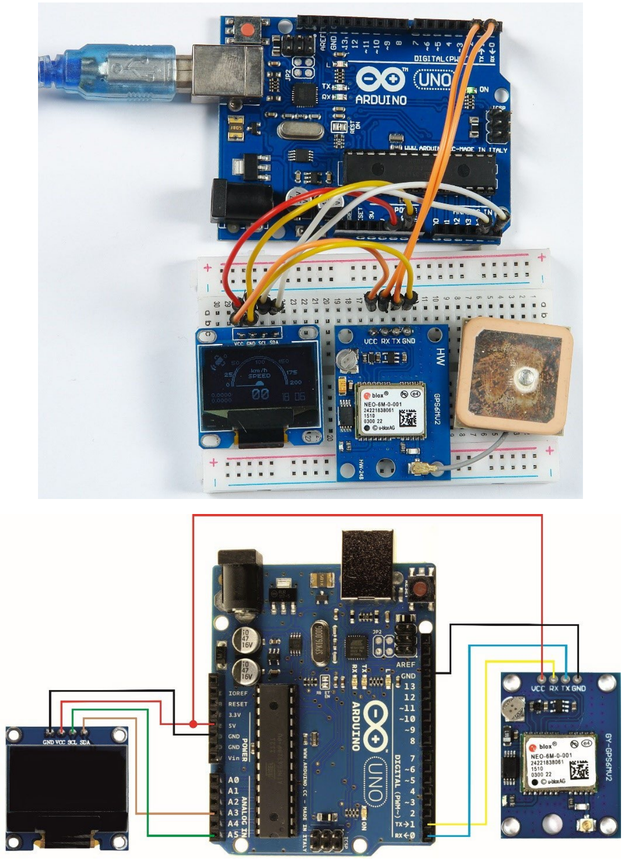
1. SSD1306: Library for Monochrome OLED based on SSD1306 drivers.
2. MicropyGPS: is an advanced GPS NMEA phrase parser designed to work with MicroPython and the PyBoard embedded microcontroller.

Also, Python 3.x is completely compatible with it. It offers helpful techniques for analyzing, displaying, logging, and manipulating GPS data [22].

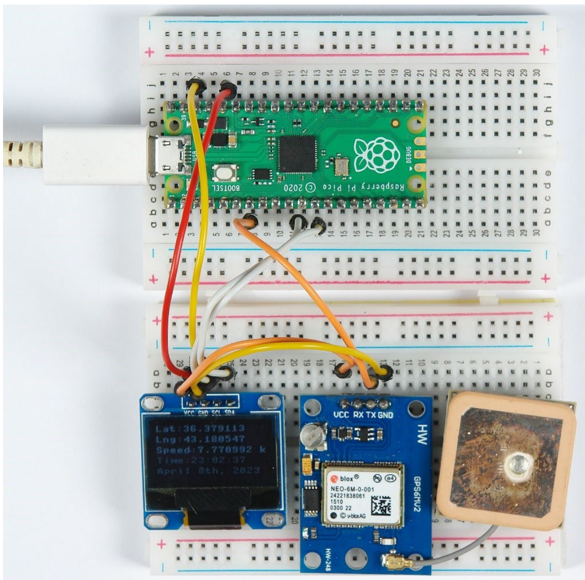
3. CIRCUIT WIRING DESIGN OF THE PROPOSED LOCATION SYSTEM

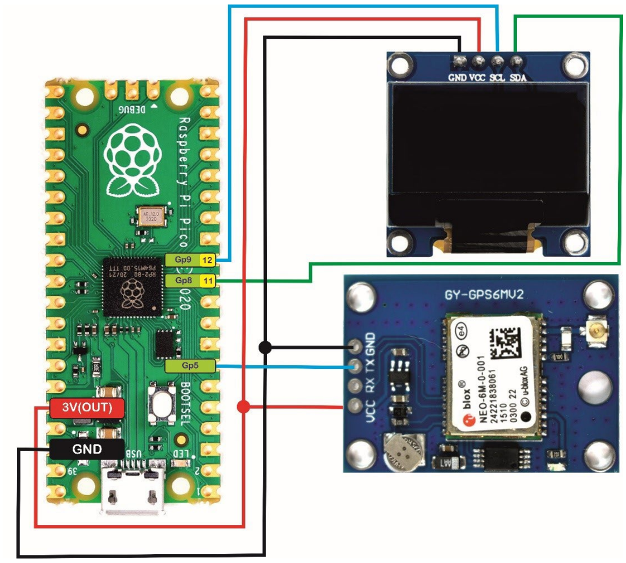
The microcontroller which is both the system's primary component and its brain or controller, is crucial to the locator tracking system. At this time, Arduino and Raspberry are each linked to all supporting components as shown in Figure 7a and b, respectively. NEO-6M connected with Arduino, where VCC of the sensor connected with Arduino 5V, sensor GND connected to Arduino ground, sensor TX connected to Arduino RX. While NEO-6M connected with Raspberry Pi Pico, the sensor VCC pin connected to Pico 3.3v, GND pin sensor to Pico GND, GPS TX pin to Arduino GPS. Wiring OLED with Arduino, OLED VCC connected to 5v Arduino, OLED SCL connected to A5 Arduino, OLED SDA connected to Arduino A4, OLED GND connected to Arduino GND.

Wiring OLED with Raspberry, OLED VCC connected to 3v3 Raspberry pin, OLED SCL pin connected to Raspberry pin 12, OLED SDA pin connected to the Raspberry pin 11, OLED GND pin connected to Raspberry GND.



(a)





(b)

Figure 7. Circuit connection for (a) Arduino microcontroller (b) Raspberry microcontroller

Choosing a microcontroller development board improves the system's efficiency. It prevents on-the-job mistakes and raises research standards. The differences between them are displayed in Table 1. In terms of processing speed and RAM, the comparison reveals that the Raspberry is superior to the Arduino and requires less voltage assumption. Raspberry is smaller, lighter, and costs half of Arduino [15] [23].

Table 1. Comparison between the two microcontrollers [15] [23]

	Arduino	Raspberry Pi Pico
Processor	ATmega 16MHz	Dual core 133 MHz
Flash memory	32 KB	2 MB
Connection	USB	USB c
I/O pins	14 I/O	30 general purposes
Voltage	5v	3.3v
Weight	25g	15g
Dimensions	(53.4x68.6) mm	51 mm x 21 mm
LAN	No	yes
Price	26.95$	13.53$

4. ANALYTICAL PERFORMANCE EVALUATION

A GPS receiver calculates its position by measuring the distance to multiple GPS satellites using time taken for a signal to move from receiver to satellite. At least four satellite signals are used by the receiver to estimate the position, and by using the time of arrival of the signals, it can calculate the distance to each satellite. These distances are then used to determine the receiver's location using a process called trilateration [24]. The basic equation for finding the distance is given by Equation (1) [24]:

$$Distance = [(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]^2 \quad (1)$$

where, x_1, y_1, z_1 and x_2, y_2, z_2 are receiver and satellite coordinates respectively.

To estimate the velocity, the receiver uses the Doppler shift of the GPS signals. The Doppler shift is a change in a wave's frequency brought on by the velocity of the source and the observer. By measuring the Doppler shift of the GPS signals, the receiver can calculate its velocity [24].

The basic equation for the Doppler shift is Equation (2) [24]:

$$f_d = (c+v)^{-1}(vf_0 - cf_0) \quad (2)$$

where, f_d is the Doppler shift, v is the velocity of the receiver, f_0 is the original frequency of the signal, and c is the speed of light.

To estimate the latitude, the receiver can use information from the GPS satellites in combination with data from other sensors such as gyroscopes, accelerometers, and magnetometers. The receiver can use the information from these sensors to calculate its orientation, or attitude, in relation to the Earth's surface. Kalman filter which is a mathematical algorithm used to estimate the attitude of the GPS receiver. It uses a combination of measurements from the GPS receiver, such as satellite signals, and data from other sensors, such as accelerometers and gyroscopes, to estimate the receiver's orientation. The Kalman filter utilizes a set of recursive equations that are used for estimating system state based on a set of measurements. It is important to note that these equations are part of a complex mathematical algorithm that is used to estimate the position, velocity and attitude of a GPS receiver, along with other calculations such as time-sync between satellite and receiver clock, error correction, and more [24].

- Distance Calculation: The Haversine formula was adopted to calculate the distance between google coordinate and the acquired GPS sensor coordinate for any of the considered locations. The haversine is based on the great-circle distance equation, which is utilized for calculating the shortest distance between 2 points on the earth. The distance was calculated between two points using the latitudes and longitudes obtained from a GPS receiver by using the Tiny GPS++ library. The following Equation (3) can be used to calculate the distance [25].

$$\begin{aligned} D = {} & 2r\arcsin[\sin^2((lat_2 - lat_1)/2)]^2 + \\ & + \cos(lat_2)\cos(lat_1)\sin^2((lng_2 - lng_1)/2) \end{aligned} \quad (3)$$

where,

D: 2 points distance difference in meters

r: Earth radius (Mean radius = 6,371 Km)

lat_1, lng_1: point one latitude/longitude in degrees

lat_2, lng_2: point two latitude/longitude in degrees

5. RESULTS AND DISCUSSION

The proposed GPS system performance for both (Arduino and Raspberry Pico) microcontrollers is investigated in terms of location precision. The tests as Figure 8 are carried out at five locations. The locations at which the precision of the determined position is tested, are chosen in Mosul, Iraq. The carried tests are implemented at five speeds approximately 0, 40, 60, 80 and 100 Km/h.

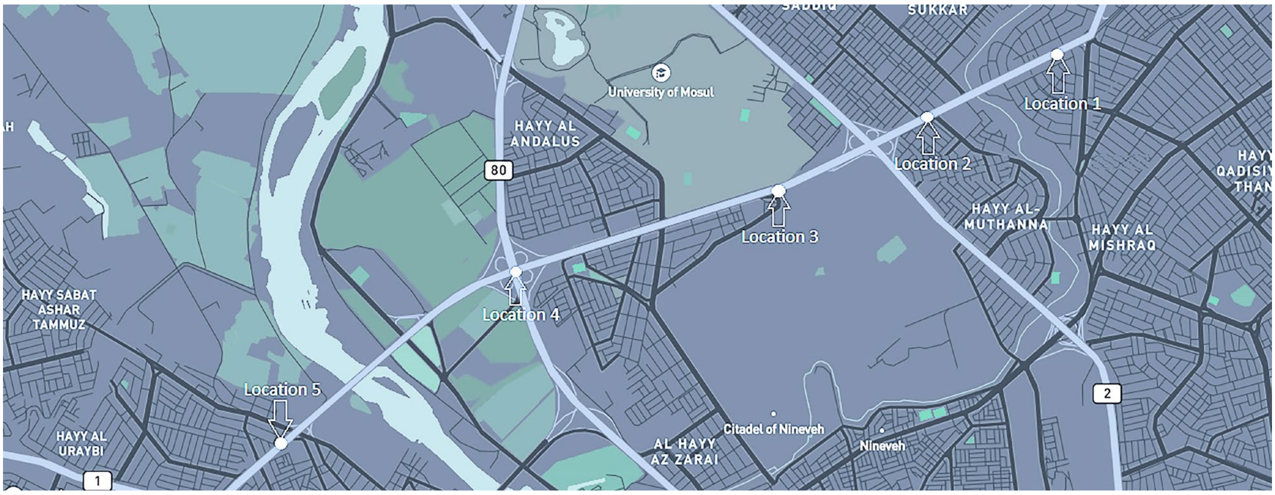


Figure 8. Five locations in Mosul, Iraq

The proposed (Arduino based) GPS has been tested. The testing process is started by uploading the software to Arduino. Wait for the GPS to acquire the coordinates. When a coordinate is obtained, the GPS will blink once per second. The coordinate data may also be displayed on screen and checked through the IDE program's serial monitor. It will display latitude and longitude, number of satellites, altitude, direction, velocity, and time as shown (for location 4 at speed of 80 km) in Figure 9.

SAT:	10	Lat:	36.3699264526	Lon:	43.1297378540	Speed:	81
SAT:	10	Lat:	36.3698692321	Lon:	43.1294975280	Speed:	81
SAT:	10	Lat:	36.3697471618	Lon:	43.1290130615	Speed:	81

Figure 9. System performance test on Arduino for location 4 at 80 km speed

The testing process is repeated for the proposed system under raspberry pi microcontroller. The results of this testing are shown in Figure 10 for location 4 at speed of 100 km.

Lat:	36.3697242737	Lon:	43.1288671494	Speed:	100.7025Km/h	SAT:	9
Lat:	36.3697242737	Lon:	43.1288671494	Speed:	100.7025Km/h	SAT:	9
Lat:	36.3697242737	Lon:	43.1288671494	Speed:	100.7025Km/h	SAT:	9

Figure 10. System performance test on Raspberry for location 4 at 100 km speed

The readings in the two figures demonstrate that using the same sensors, the Raspberry reads more than 14 data coordinates in less than one second compared to one data coordinates every two seconds for the Arduino.

This illustrates how crucial it is to choose the right microcontroller. The results for different at location for are shown in Table 2. The results shown that for Arduino microcontroller the number of satellites is changed from (9 to 11) satellites. the degrees in latitude are changed from (0.00003 to 0.0001) and degrees in longitude are changed from (0.00002 to 0.0005) for GPS sensor. The error in distance between the google coordinates and sensor coordinates is calculated using Haversine formula. These errors changing from (5.3 to 48.8) meters for speed changing from (0 to 100) km respectively. These results show that the accuracy of the sensor decreased as the velocity increased.

The results for different speed at location 4 are shown in Table 3 shows that the number of satellites is changed from (9 to 11) satellites. the degrees in latitude are changes from (0.00004 to 0.00009) and degrees in longitude is changed from (-0.000005 to 0.0002) for GPS sensor. The error in distance between the google coordinates and sensor coordinates is calculated using Haversine formula. These errors changing from (4.3 to 25.9) meters for speed changing from (0 to 100) km respectively. These results also shows that the accuracy of the sensor decreased with velocity increasing. The over all results from the two tables shows that Raspberry better than Arduino due to higher processing rate of the raspberry.

Table 2. Location 4 for Arduino microcontroller

Coordinates from Google Maps			Latitude		Longitude	
			36.3696310000		43.1286010000	
Arduino uno						
Speed km/h	*sat*	*lat*	*lng*	*Error lat*	*Error lng*	*Error*/m
0	9	36.3696632385	43.1286277770	0.0000322385	0.0000267770	5.3
41	11	36.3697242736	43.1289100646	0.0000932736	0.0003090646	29.5
61	10	36.3697013854	43.1290168762	0.0000703854	0.0004158762	38.04
81	10	36.3697471618	43.1290130615	0.0001161618	0.0004120615	39.08
100	9	36.3697357177	43.1291313171	0.0001047177	0.0005303171	48.8

Table 3. Location 4 for Raspberry Pi Pico

Coordinates from Google Maps			Latitude		Longitude	
			36.3696310000		43.1286010000	
Raspberry Pi Pico						
Speed km/h	*Sat*	*latitude*	*Longitude*	*Error lat.*	*Error lng.*	*Error*/m
0	11	36.3696789742	43.1285953522	0.0000479742	-0.0000056478	4.3
41	12	36.3697123528	43.1287145615	0.0000813528	0.0001135615	13.6
60	8	36.3697600365	43.1288623810	0.0001290365	0.0002613810	18.4
80	11	36.3696724540	43.1287381106	0.0000414540	0.0001371106	21.11
100	9	36.3697242737	43.1288671494	0.0000932737	0.0002661494	25.9

In this study, two microcontrollers, Arduino and Raspberry Pi Pico, were used in the proposed GPS system. System performance was verified under these microcontrollers. The findings show that the Raspberry Pi was better than the Arduino in positioning accuracy at different speeds and locations. The Raspberry reads more than 14 data coordinates in less than one second compared to one data coordinates every two seconds for the Arduino, as a result it is concluded that Doppler shift has less effect on the accuracy of the results when the Raspberry Pi Pico microcontroller is tested. For speed changing from 0 to 100 Km/h, the location accuracy changed in the Arduino from (5.3 to 48.8) meters, while that of the Raspberry Pi Pico changed from (4.3 to 25.9) meters at speeds of (0 to 100) km/h. This is due to the high processing speed of the Raspberry Pi Pico. Which results higher Doppler effect reduction in comparison with the Arduino microcontroller.

REFERENCES

[1] K. Suseenthiran, A. Shukur, K. Wei, M. Zoinol, A. Awang, S. Huzaimah, N. Mohd, "Indoor Positioning Utilizing Bluetooth Low Energy RSSI on LoRa System", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 23, No. 2, pp. 927-937, Melaka, Malaysia, August 2021.

[2] T. Khalifa, N. Sahar, N. Ramli, M. Islam, "Circularly Polarized Microstrip Patch Antenna Array for GPS Application", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 15, No. 2, pp. 920-926, Malaysia, August 2019.

[3] N. Islam, M. Alomgir, M. Mazharul, M. Nur, Khairunnaher, "Cost-Effective and User-Friendly Vehicle Tracking System Using GPS and GSM Technology Based on IoT", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 28, No. 3, pp. 1826-1833, Dhaka, Bangladesh, December 2022.

[4] M. Khalil, N. Mahmood, S. Najim, "Smart System for Monitoring Ammonium Nitrate Storage Warehouse", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 23, No. 1, pp. 583-589, Baghdad, Iraq, July 2021.

[5] T. Morallo, "Vehicle Tracker System Design Based on GSM and GPS Interface Using Arduino as Platform", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 23, No. 1, pp. 258-264, Dumaguete, Philippines, July 2021.

[6] M.K. Sameer, A.M. Hamid, "Evaluation of Accuracy GPS and Automatic Level Instruments in Topographic Surveying", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 53, Vol. 14, No. 4, pp. 75-79, December 2022.

[7] www.kbvresearch.com/global-positioning-systems-market/

[8] S. Pooja, "Vehicle Tracking System Using GPS", International Journal of Science and Research, Issue 9, Vol. 2, pp. 128-130, Chhattisgarh, India, September 2013.
[9] K. Al Gaadi, "Testing the Accuracy of Autonomous GPS in Ground Speed Measurement", Journal of Applied Sciences, Vol. 5, pp. 1518-1522, No. 9, Riyadh, Saudi Arabia, 2005.
[10] J. Schipperijn, J. Kerr, S. Duncan, Th. Madsen, Ch. Demant, J. Troelsen, "Dynamic Accuracy of GPS Receivers for Use in Health Research: A Novel Method to Assess GPS Accuracy in Real-World Settings", Frontiers in Public Health, Vol. 2, pp. 1-8, No. 21, March 2014.
[11] M. Hakimi, M. Fiqri, "GPS Based Vehicle Tracking System", International Journal of Scientific and Technology Research, Issue 04, Vol. 10, pp. 278-282, Malaysia, April 2021.
[12] X. Zhao, K. Carling, J. Hakansson, "Reliability of GPS Based Traffic Data: An Experimental Evaluation", Working Papers in Transport, Tourism, Information Technology and Microdata Analysis, pp. 1-23, No. 17, Sweden, 2014.
[13] E. Adia, E. Inaiyah, R. Tri, "A Smart Wearable Device Based on Internet of Things for the Safety of Children in Online Transportation", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 22, No. 2, pp. 708-716, Indonesia, May 2021.
[14] N. Genc, S.J.M. Shareef, "Design Dual Axis Sun Power Tracking System Using Arduino", International Journal on Technical and Physical Problems of Engineering, Issue 29, Vol. 8, No. 4, pp. 34-38, December 2016.
[15] https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf, pp. 17-23, 2020-2023.
[16] O.B. Kharisma, A.A. Dzikra, Mustakim, R. Vebrianto, R. Novita, Hasbullah, Irawati, Y. Novita, Zaitun, A. Nazir, I. Iskandar, Y. Vitriani, R. Rehayati, T. Andriani, "Development of Location Tracking System via Short Message Service (SMS) Based on GPS Unblox Neo-6m and Sim 8001 Module", The 1st Workshop on Environmental Science, Society, and Technology, No. 1363, pp. 1-7, Indonesia, November 2019.
[17] https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf, pp. 6-12.
[18] https://github.com/olikraus/u8glib
[19] https://docs.arduino.cc/learn/built-in-libraries/software-serial
[20] http://arduiniana.org/libraries/tinygpsplus/
[21] www.megunolink.com/documentation/interface-panel/gauge/
[22] https://github.com/inmcm/micropyGPS
[23] S. Mohol, A. Pavanikar, G. Dhage, "GPS Vehicle Tracking System", International Journal of Emerging Engineering Research and Technology, Issue 7, Vol. 2, pp. 71-75, Pune, India, October 2014.
[24] E. D. Kaplan, Ch. J. Hegarty, "Understanding GPS Principles and Applications", Artech house, Norwood, pp. 50-150, London, UK, 2006.
[25] N. Kahtan, "Design and Implementation of a Real-Time Vehicles Monitoring System Based on IoT", Master thesis, Northern Technical University, Engineering Technical Collage, Mosul, Iraq, p. 46, 2022.

BIOGRAPHIES

First Name: **Ziyad**
Middle Name: **Khalaf**
Surname: **Farej**
Birthday: 17.02.1964
Birthplace: Mosul, Iraq
Bachelor: Electronic System Engineering, Cranfield University, Cranfield, UK, 1989
Master: Electronic and Communication Engineering (Spread Spectrum System/FH), University of Mosul, Mosul, Iraq, 2003
Doctorate: Computer and Communication Networks Eng. Department of Electric Engineering, University of Mosul, Mosul, Iraq, 2012
The Last Scientific Position: Asst. Prof., Computer Technology Engineering department, Technical Engineering College/ Mosul, Northern Technical University, Mosul, Iraq, Since 2022
Research Interests: Wireless Computer and Communication Networks, 802.11 WIFI, Internet of Things
Scientific Publications: 30 Papers, 4 Projects, 11 Theses
Scientific Memberships: Iraqi Engineering Union

First Name: **Ahmed**
Middle Name: **Nadeem**
Surname: **Ismail**
Birthday: 01.06.1985
Birthplace: Mosul, Iraq
Bachelor: Computer Technology Engineering, Al Hadba University College, Mosul, Iraq, 2014
Master: Student, Computer Technology Engineering, Computer Technology Engineering Department, Northern Technical University, Mosul, Iraq, Since 2022
Research Interests: IoT System, System Monitoring
Scientific Publications: 1 Paper
Scientific Memberships: Iraqi Engineering Union