MOKSLO TAIKOMIEJI TYRIMAI LIETUVOS KOLEGIJOSE, 2022 Nr.18 (1), p. 185-189
ISSN 1822-1068 / eISSN 2335-8904, http://ojs.kaunokolegija.lt/index.php/mttlk/index

DRONE STADIUM INTERACTIVE SYSTEM

Petras Zigmantas Barčkus[1], Vidmantas Rimavičius[1], Dangis Rimkus[2], Dalius Makackas[3]

[1]Kaunas University of Applied Sciences, [2]Kaunas University of Technology,
[3]Marijampole University of Applied Sciences

Abstract. The article proposes new access to the use of drones and the development of their technical capabilities. Innovative features include an interactive drone flying track offering cordless sensors. The interactive route, via wireless, is associated with a real-time WEB graphical interface. The result of the project is the Drone stadium interactive system, realised with wireless sensors. According to the selected criteria, requirements are set for the system design. The project aims to create a space without pilots with cordless sensors and link them with a real-time user interface, thus extending the application of the abovementioned stadiums or hangars by improving the technical capabilities of the system or its components.

Keywords: drone racing, ultrasonic sensor, Arduino, wi-fi, wireless network.

Introduction

The market is becoming increasingly innovative with the rapid development of technologies. Nowadays, there is a lot of interest in drones (absence of pilot), their technical capability and their possible use in various industries. Drones are becoming increasingly widespread in various fields of activity, such as film production, when the images are taken from above. Drones were first used in Australia in 2014 (Vėželis, 2017). At that time, drone enthusiasts organised racing indoors. The short video about the use of drones in sports announced and shared on the Internet has received considerable interest (DRL, 2018).

This area is still under development in Lithuania. There is a "Robotics School" in Vilnius, where courses are organised. During the classes, it is explained how to collect the drones and manage them. Real competitions in Lithuania were organised on May 24-26, 2017, in the Cathedral Square of Vilnius. Also, Lithuania started building a drone fly map to display drone flight zones within the country (Dronemade, 2019).

More recently, users who buy these unmanned aircraft to spend time driving the drones are increasingly frequent (Technologijos.lt, 2015). This consumer's goal will be complemented by racing and competition. Thus, consumers' interests become like games that require unmanned aircraft runways (drone runway, i.e., dronodrome) (Dronu Mokykla, 2016).

To achieve this, it is necessary to create a drone trail for the pilot to fly and not depart from the track. Because the route will be interactive, the sensors will be located at a specified distance on the track, which will fix or drone pass through the sensor. The sensor can also be connected to the illumination to allow the pilot to see where to fly. The sensors will be bundled with wi-fi modules that communicate with the microcomputer and routers; later, they will send information to the pilot computer, which will display the route map. The map will show the location of the sensors on the track. Passing the sensor on the map will indicate that the pilot has travelled the required sensor and can fly further. Also, the time of the first and last sensors will be captured so that it is possible to determine the time spent throughout the game so that players can compete with each other.

To create a working and practically usable drone route with low economic costs, a wireless network between sensors, microcomputers, modems and a user's computer is needed. Also, the user has an interface that allows the pilot to see the sensor points on the map, the ones the drone has already passed through and fix the time taken to complete the route.

The objectives set to achieve the aim included:

1. To define the wireless sensor network and analyse it.
2. Create a users' interface for a pilot on a WEB server.
3. Select the technical components and evaluate the project from the economic point of view.
4. Create a user's interface for the pilot on a WEB server.
5. Compile a description of the functioning of the system.
6. To present the results and draw conclusions.

The architecture of the drone stadium system

This project examines the Drone stadium's interactive system with wireless local-lock sensing networks. The system can be adapted to various locations. Besides, the safety standards related to the localities that prohibit unmanned aircraft flights must be considered. This system is based on ultrasonic distance sensors, microcomputers or microcontrollers, wi-fi modules for microcomputers or microcontrollers, routers, and the WEB server. With the help of a sensor, it is possible to detect the flight of a drone in a certain place. The drone should fly about one meter from the sensor to capture it. The data captured by the drone sensor will be sent to the WEB server with the user's interface. Each sensor will have its own point on the map, which will be recorded on the WEB server. With the help of the map, the user will be able to see already scattered sensors and those which will still have to fly through. The sensor will also be able to tell the user on the WEB server the time over which the pilot overcame the route. Figure 1 below shows the logical and physical diagram of a preliminary Drone stadium to demonstrate seven sensors on the map on a WEB server.

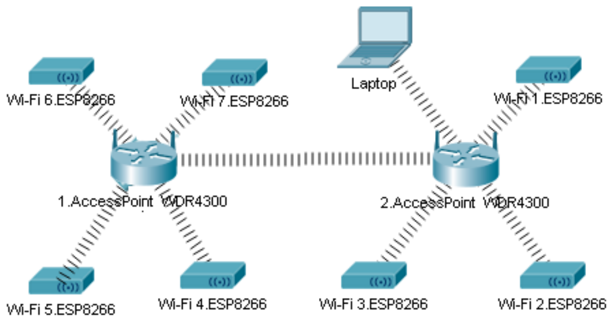

Fig. 1. Components of the wireless network of the drone stadium system

The diagram (Fig. 1) shows how the components are interconnected and can communicate with one another. These components are:
1) microcontroller Arduino Uno R3;
2) ultrasonic sensor SRF05;
3) wi-fi 802.11 b/g/n module ESP8266;
4) wireless router ieee 802.11 b/g/n 150mbps;
5) additional devices and materials (voltage converter from 5V to 3.3V, battery 3300mAh, resistors, wires, hermetic boxes).

The Arduino Uno R3 microcontroller performs the main function in this scheme, combining all the necessary components physically and programmatically (Patterson R. W., Patterson K. M. 2013). Because the esp8266 wi-fi module has a working voltage of 3.3V, a voltage converter that makes 3.3V out of 5V is used, as Arduino Uno R3 has a working voltage of 5V. The wi-fi module is practically plugged into a voltage converter chip and does not need to be used for additional wiring. An ultrasonic distance meter is connected via a pair of resistors because it is necessary to reduce the current.

Distance of drone identification algorithm

When the entire system is initiated, all physical program elements and objects are activated and purposefully configured, the system operation can be described step by step (Fig. 2).

We will divide these stages according to the use of each object consistently. In particular, this is an ultrasonic distance measuring sensor that interacts with a microcomputer. Theoretically, this sensor measures a distance from one centimetre to four meters at an angle of thirty degrees. However, in practice, it is necessary to use less than one and a half meter distances to measure it without any significant errors.

Further, when interacting with a microcomputer, the distance recorded by the sensor will be assigned to the microcomputer variable, which will check the conditions. If the condition is satisfied and the distance recorded is less than one meter, the microcomputer assigns the variable, which must be sent for further processing. When the variable already satisfies the condition, the microcomputer interacts with the connected wi-fi module. The microcomputer sends commands to the wi-fi module and specifies what data to send at what address. When it is sent to the correct address on the router and accesses the connection, it is sent to another nearby router or an associated WEB server. The data received on a WEB server will be indicated in the user's interface, which will contain a map indicating where the sensors are located. If the sensor detects that the distance is less than one meter on the map, the sensor lights up green in that place. If it does not, then the red light remains. Of course, more checkpoints will interact with routers and the WEB server, so the pilot (player) can fly through those checkpoints. Also, all sensors on the map are green, and then the system indicates how long it took to finish the route. Once the whole route has been finished, it will be possible to start it again through the user's interface by re-pressing the button, which will make all the sensors on the map red.

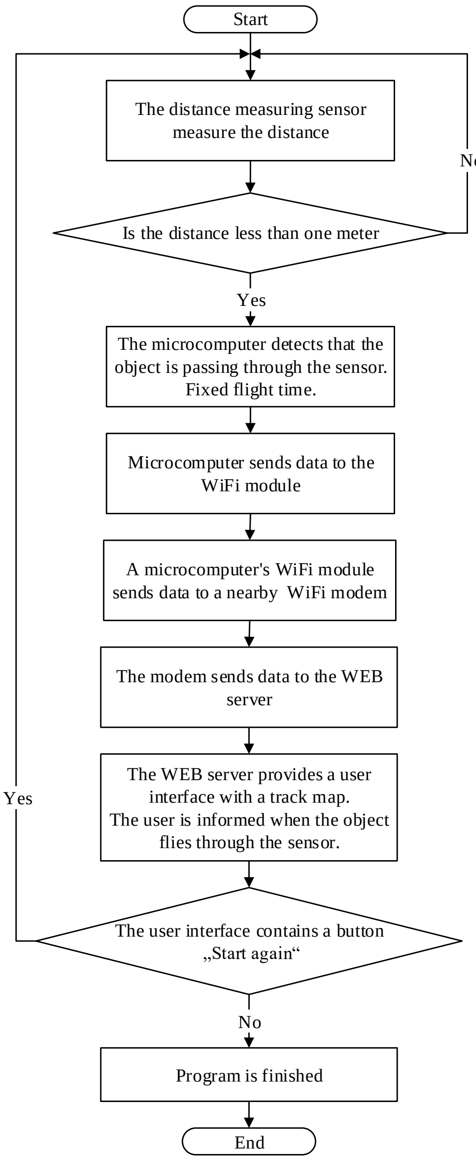

Fig. 2. Algorithm of drone stadium system

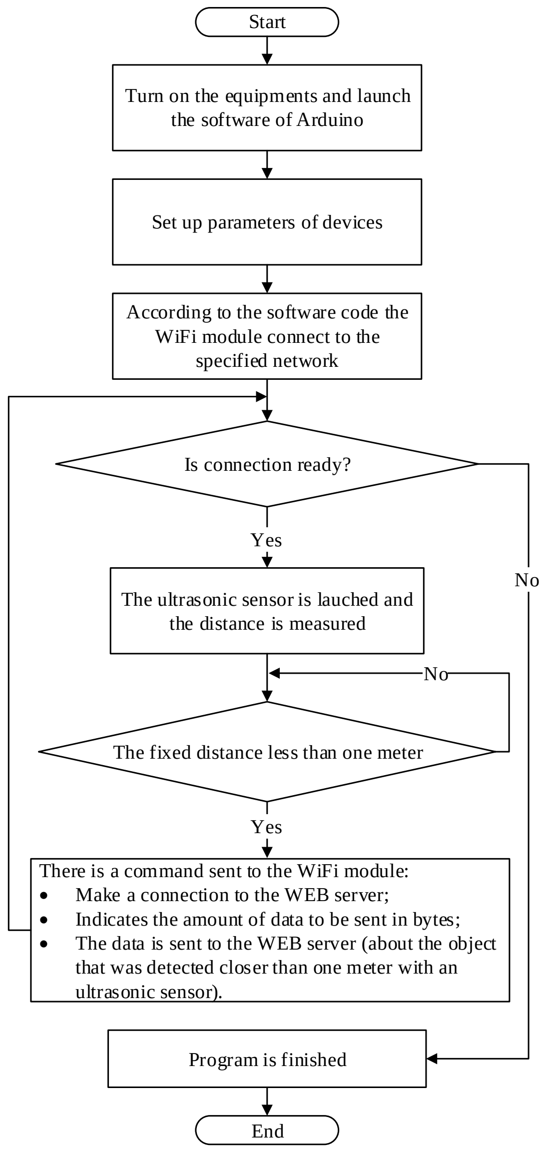

Fig. 3. Arduino program block diagram

The block diagram of the Arduino Uno R3, which this microcontroller guides (Boxall J. 2013), is plotted in Fig. 3. The components are triggered, and the program code initiated is entered into the Arduino.

When the first iteration is executed, the code will first describe the parameters and variables of the components connected to the microcontroller. Then it will send an AT command to the wi-fi module, which will list the wi-fi network desired to connect to, where SSID is „WDR4300“ and password is „dr0n@dr0m“. The command will indicate the wireless connection to which SSID and Password are required. Here’s what the AT command looks like:

("AT+CWJAP=\"WDR4300"\,"\"dr0n@dr0m\"\r\n")

When this command is sent to the Esp8266 wi-fi module, it connects to the specified network and receives its IP address (Espressif 2018). If the connection fails, the program will not continue.

When connected to a network, an infinite loop is initiated, in which the ultrasonic sensor collects data about the distance. If the distance is more than one meter, it does not use this data anywhere and continues to measure the distance every time the loop is cycled. Only when the distance is less than one meter, it will send commands to the wi-fi module that connects the WEB server to the wi-fi module.

("AT+CIPSTART=\"TCP"\,"\"192.168.33.100\",80\r\n")

This command specifies the protocol to be connected to the WEB server (Schwartz 2014), i. e., TCP. The WEB server’s IP address is “192.168.33.100”, and the port 80 is used. The command completion symbols are “\r\n”. When executing this command, a certain amount of time is needed to send the required data.

esp8266.write("AT+CIPSEND=61\r\n")

This command specifies the number of characters to be sent to another command. As we see in the next command, there will be 61 characters. Symbols \ r and \ n are counted as one character.

("GET /receiver.php?first=1 HTTP/1.1\r\nHost: 192.168.33.100\r\n\r\n ")

This command is already sent to the "receiver.php" variable "first" which is equal to the unit specified by the IP address.

After all the data of these three commands are received in the "receiver.php" file in the WEB server, this file will further process the variable. The following steps will again check if the distance recorded is less than one meter, and if less, the commands above will be initiated again and contact the WEB server again.

Testing Drone Stadium System

This preliminary physical topology (Fig. 4) indicates how to arrange and use the equipment. While testing the system, the stands were placed in the stadium of Kaunas University of Applied Sciences. An electric cable was brought to the modems. Modems should be covered or put in boxes that are protected from rain.

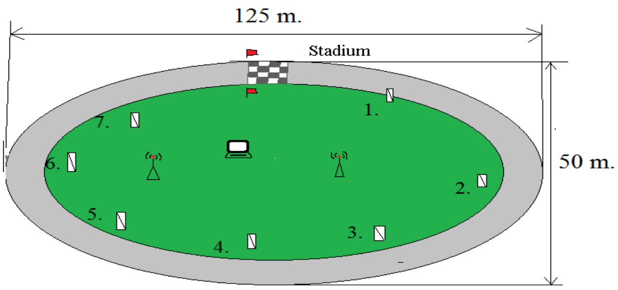

Fig. 4. Preliminary physical topology

This user's interface (Fig. 5) is created using HTML, PHP and CSS programming languages W3SCHOOLS. It works in real time. Google Chrome browser can test this user's interface. We could test only the first "checkpoint" because we used only one sensor. If more sensors are connected, and everything is configured properly in the user's interface, we will test all the checkpoints in the user's interface. It also is possible to modify "checkfile.txt" data received from the sensors so that their operation can be tested.

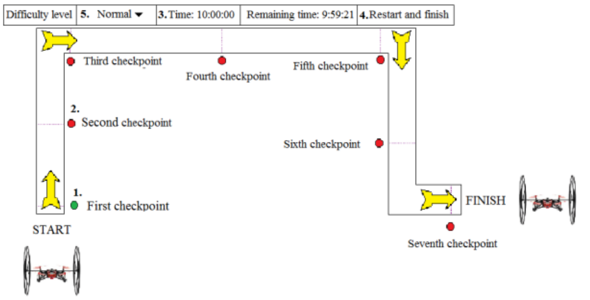

Fig. 5. Graphical user's interface

1) This is where the "First checkpoint" is located. It is green because it received the required information from the sensor, which means that the sensor has captured the object (drone). The arrows indicate which trajectory the pilot needs to fly with the drone to reach the next sensors.
2) There are currently no detected data from the second checkpoint. Sensors from the second sensor have not yet detected any objects.
3) This time is displayed in real time. It is the time when the drone was captured in the first sensor.
4) The "Restart and Finish" button erases the data received from sensors and the time captured according to the first sensor.
5) In this place, you can choose the complexity of how to position the sensors. This feature is for further development.

This interface is a prototype and can be further refined according to desired needs and outcomes. In our opinion, further development would require a better presentation of the database and Javascript programming language for animations in the interface.

Conclusions

1. A comparison of wi-fi and Bluetooth-based wireless sensor networks has proved that wi-fi is superior to Bluetooth in data transfer rates more than a hundred times, wider frequency bandwidth, and a wider action zone.
2. The algorithm of "Drone stadium interactive systems with wireless local area network sensing networks" has been developed and described.
3. Software of the interface of the drone's user has been implemented.
4. During the test, it was found that the technical and software parts meet the requirements set and practically operate in real time.
5. The technical components and software have been selected to evaluate the economic utility and technical features. Components and software have been selected according to

realisable functions, including the advantages and disadvantages.

6. The system provides possibilities for improving the user's interface of the software and the integration of new technical components into the system by extending functionality.

References

1. Vėželis, P. (2017). *Droneracing*. http://droneracing.lt/dronu-sportas-lietuvoje-ir-pasaulyje/
2. DRL. (2018). *The drone racing league*. https://thedroneracingleague.com/technology
3. Dronemade. (2019). *The Ultimate Guide to Lithuania (Vilnius) Drone Laws & Rules.* https://www.drone-made.com/post/lithuania-drone-laws
4. Technologijos.lt. (2015). *Nauja sporto šaka: dronų žaidimai.* http://www.technologijos.lt/n/technologijos/lektuvai_ir_aviacija/S-50197/straipsnis/Nauja-sporto-saka-dronu-zaidimai-Video
5. DronuMokykla. (2016). *Dronų lenktynės*. http://www.dronumokykla.lt/dronu-lenktynes
6. Patterson R. W., Patterson K. M. (2013). Arduino for Teens, Course Technology, p 301.
7. Boxall J. (2013). Arduino Workshop: A Hands-on Introduction with 65 Projects, No Starch Press, p 392.
8. Espressif (2018). ESP8266 AT Instruction Set. [Online]. Available: https://ieee-sensors.org/wp-content/uploads/2018/05/4a-esp8266_at_instruction_set_en.pdf. [Accessed 20 06 2022].
9. Schwartz M. (2014). Arduino Networking. Packt Publishing, p 206.
10. W3SCHOOLS „W3Schools Online Web Tutorials“ [Online]. Available: https://www.w3schools.com/. [Accessed 13 02 2019].

DRONODROMO INTERAKTYVI SISTEMA

Santrauka

Šiame straipsnyje nagrinėjamas kitoks požiūris į bepiločių skraidyklių pritaikymą bei jų techninių ir programinės įrangos galimybių plėtojimą. Projekto tikslas – sukurti erdvę skirtą bepilotėms skraidyklėms, jas susiejant su realaus laiko vartotojo sąsaja ir belaidžio ryšio jutikliais, pagerinti sistemos ar atskirų komponentų technines ir programines galimybes, taip pritaikant stadionus, angarus ar kitas panašias erdves. Reikalavimai sistemos projektavimui suformuoti pagal pasirinktus kriterijus. Tarp inovatyvių funkcijų yra interaktyvus bepiločių skraidyklių (dronų) skraidymo stadionas su belaidžio ryšio jutikliais. Šiame darbe naudojant belaidžio ryšio jutiklius kuriama Dronų stadiono interaktyvi sistema. Naudojant žiniatinklio serveryje esančią grafinę vartotojo sąsają realiu laiku per belaidį ryšį yra pasiekiamas interaktyvus skraidymo kelias.

Reikšminiai žodžiai: dronų lenktynės, ultragarsinis jutiklis, Arduino, Wi-fi, belaidžio ryšio tinklas.

Information about the authors

Petras Zigmantas Barčkus. Student at the Department of Applied Informatics of the Faculty of Technologies, Kaunas University of Applied Sciences. Kaunas, Lithuania.
E-mail address: petraszigmantas.ba4603@go.kauko.lt

PhD Vidmantas Rimavičius. Lecturer at the Department of Applied Informatics of the Faculty of Technologies, Kaunas University of Applied Sciences. Kaunas, Lithuania.
E-mail address: vidmantas.rimavicius@go.kauko.lt

PhD Dangis Rimkus. Lecturer at the Department of Applied Informatics, the Faculty of Informatics, Kaunas University of Technology. Kaunas, Lithuania.
E-mail address: dangis.rimkus@ktu.lt

PhD Dalius Makackas. Lecturer at the Faculty of Business and Technology, Marijampole University of Applied Science. Marijampole, Lithuania.
E-mail address: dalius.makackas@mkolegija.lt