

System and method of automatic collection of objects in the room

Mariia Yu. Tiahunova, Halyna H. Kyrychek, Tetiana O. Bohatyrova and Daryna D. Moshynets

National University "Zaporizhzhia Polytechnic", 64 Zhukovsky Str., Zaporizhia, 69063, Ukraine

Abstract

A system and method of automatic collection of objects in a room focused on the minimum energy consumption is proposed in this paper. This result is achieved by the implementation of an improved method of automatic collection of items in the room and mathematical method for calculating the desired motion trajectory; development and implementation of algorithms that implement the proposed method; software and hardware implementation of the system for automatic collection of items with minimal energy consumption, including the small number of system components and an improved movement trajectory. The system's main component is the Arduino Uno, which acts as a controller. The developed software makes it possible to evaluate the implemented method's effectiveness in a real-life system. An application example of the proposed method is given.

Keywords

robotplatform, Alhabet, Arduino, optimization, evolutionary method

1. Introduction

Humanity is on the verge of a new era of technological development. Already now, robotics is rapidly being introduced into people's daily lives, helping them to interact more effectively with automated systems, improve existing jobs and, in general, give people more time to focus on what interests them, it is important and fun. We are talking about cyber-physical systems for modeling complex sociotechnical systems that largely control themselves.

The principle of operation of such systems is to combine physical production processes or any other processes that require continuous control in real-time using software [1] and electronic systems.

The robot differs from a conventional automatic system in its multipurpose purpose, versatility, and ability to be reconfigured to perform various functions. Automation of robotic systems

CS&SE@SW 2021: 4th Workshop for Young Scientists in Computer Science & Software Engineering, December 18, 2021, Kryvyi Rih, Ukraine

✉ mary.tyagunova@gmail.com (M. Yu. Tiahunova); kirgal08@gmail.com (H. H. Kyrychek);

t.bohatyrova.un@gmail.com (T. O. Bohatyrova); reif785@gmail.com (D. D. Moshynets)

🌐 <https://www.facebook.com/mary.tyagunova.7> (M. Yu. Tiahunova);

<http://csn.zntu.edu.ua/galina-grigorivna-kirichek> (H. H. Kyrychek);

<https://www.facebook.com/profile.php?id=100074077952151> (T. O. Bohatyrova);

<https://www.facebook.com/profile.php?id=100012258404028> (D. D. Moshynets)

🆔 0000-0002-9166-5897 (M. Yu. Tiahunova); 0000-0002-0405-7122 (H. H. Kyrychek); 0000-0002-3430-2117

(T. O. Bohatyrova); 0000-0002-7743-3985 (D. D. Moshynets)

© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

requires the development of robots capable of intellectual activity, using a pre-prepared program or artificial intelligence technologies [2, 3, 4].

Today, the use of robots covers almost all industries and tasks. With the rapid development of robotics, human-machine interaction will soon become a common daily practice. Moreover, today technological advances are increasing the adaptability and flexibility of robots. Most people in developed countries prefer home automation, allowing control of lighting, air conditioning, audio content, security systems, and home appliances. Much attention is paid to the creation of mobile robots that are used in home conditions or in situations where the presence of a person is dangerous and optional. Modern robotics focuses on the complete automation and autonomy of robots and robotic systems. A characteristic feature of work is the ability to partially or completely fulfill the mechanically motor and intellectual functions of a person.

Additional support is provided by connected jobs that apply to all types of services, such as robotic vacuum cleaners. Modern companies are actively developing this area, developing new assistant robots. The number and popularity of home robotic assistants in homes are growing at a high rate, due to the increase in the functions that provide work and the decrease in the price of them.

Most robotic home assistants are entertaining in nature, such as a voice assistant built into a music speaker. Another group of robots is designed to free humans from doing household chores, they can guardhouses, wash windows, clean pools and do small household chores. Work greatly improves the quality of our life at home, at work and during leisure. They are adapted to recognize objects, distinguish objects of the environment, repeat simple human movements, coordinate with other works. Such functionality has become possible thanks to the use of innovations in the development of robotics, as well as due to the improvement of algorithms that control perception, thinking, control and coordination of work.

The purpose of the study is to organize the automatic collection of items in the room by developing a system and collection method that allow assessing not only the results of the work, but also implementing the collection process with minimal energy consumption.

The objective of the research is to develop a robotic autonomous system for collecting objects in a room with increased requirements for energy saving due to the developed method.

2. System structure

The basis of the collection system is represented by the AlphasBot double-deck robotic platform, which allows you to remotely control the development via Bluetooth or WiFi. The robotic platform is equipped with two motors, contains mounting holes for mounting various sensors, controllers and power supplies. The main component of the aforementioned system is the Arduino Uno hardware and software complex, which acts as a control controller. It is a board that houses the microcontroller and other electronic components.

The MG995R servo motor is used for automatic remote control of the claws, which assists in the opening (closing) functions. The operation of the wheels is provided by DC motors with a gearbox, the rotation speed of the motors is helped to determine the photo-interrupting sensors. To determine the distance, the HC-SR04 ultrasonic sensor is connected to the robot platform [5, 6].

The robotic claw is made of light metal. The claw opens approximately 55 mm and, depending on the servo used, can lift heavy objects.

The main connections to the Arduino board are shown in figure 1. In this case, the Board is connected in a standard way to the Alfabot platform, according to the documentation [5].

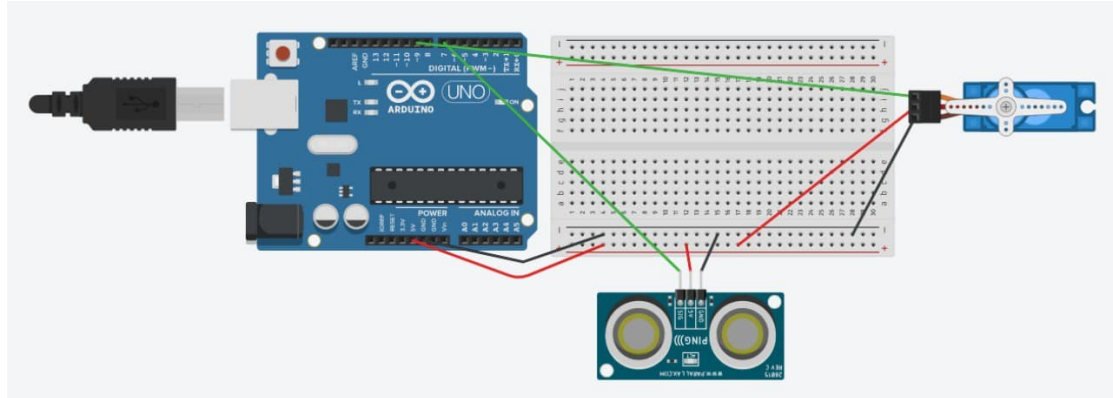


Figure 1: Connections to the Arduino board.

Optimization of energy consumption in this work is considered from the point of view of minimizing the trajectory of the robot's movement. Changing the component base, in the future, is planned as the next stage of research in terms of energy consumption optimization.

3. System operation

For the effective functioning of the system, a method for collecting items from the floor of the room was developed, which allows collecting with minimal energy consumption. In order for the energy consumption to be minimal, it is necessary that the time of the task being performed is also minimal. This result can be achieved by moving the robot along the minimum path along the total distance.

Thus, the developed method of collecting objects from the floor is as follows. The room where the robot moves to collect items is a matrix of points, each of which is equal to the length of the robot platform with a claw. The robot is installed in those points where it can turn to the left. With the help of an ultrasonic sensor, it is determined how far it can travel forward, that is, to a wall or an object [7, 8].

As soon as the system reaches the obstacle and it turns out to be a wall, then this point becomes a "basket" where objects will be collected. Then the robot makes a 180-degree turn and continues to go to the next obstacle. If there is an object on the way, he tries to grab it and, if successful, determines the shortest path to the "basket" from the point at which he stopped, and begins to move in this direction.

Having reached the point of collecting objects, he leaves the object and returns to the point where he picked it up, and continues to search for the following objects. If the robot was unable to grip the object, because the dimensions differ from the permissible parameters, it makes a detour around the object. When the robot reaches the other end of the room, it looks for the

shortest route back to the starting point. As soon as the robot is at the starting point of the room, it turns off.

The algorithm for the functioning of the system that implements the developed method of collecting items, based on the above, is as follows:

1. The robot scans the distance to the obstacle using an ultrasonic sensor.
2. The robot moves towards an obstacle.
3. Having stopped at an obstacle, the object is gripped: the claw opens, the robotic platform travels 4 cm and the claw is closed.
4. The robot moves backward 8 cm and checks the distance:
 - if the distance does not change and the “basket” is indicated, then go to step 8;
 - if the distance does not change and the “basket” is not indicated, then the robot turns in place by 180 degrees, moves to the wall with the help of obstacle sensors and makes a reverse turn;
 - if the distance changes and the “basket” is indicated, then go to step 6;
 - if the distance changes and the “basket” is not indicated: the robot defines this point as a collection point. Then step 6 is performed.
5. The robot opens the claw, moves forward 4 cm, closes the claw. Then step 9 is performed.
6. Turn left.
7. If the distance allows movement, a 90-degree turn is performed, then step 1 is performed, otherwise, the robot is at the end of the room: step 10 is performed.
8. The shortest path to the “basket” is determined. A movement is made towards it and step 5 is carried out.
9. The shortest path to the capture point of the object is determined. When the robot arrives at this point, step 1 is performed.
10. The shortest path to the starting point is determined.
11. The movement to the starting point is carried out, and the robot finishes its work.

4. Mathematical justification

To prove that energy consumption will be minimal, with the smallest distance traveled by the robot and with the minimum number of devices, we will construct the following mathematical model.

First, let's prove that a device with fewer connected devices should use less power.

Let S – be the distance that the robot will cover when moving around the room, while collecting all possible objects. Then we will designate the energy that he will spend as E_{np} . Value N – the number of connected devices powered from one source.

The energy of each i -th device is denoted as E_i , where $i = (1, N)$. Then the energy that will be spent by the robot for the entire path traveled is determined as follows: $E_{np} = \sum_1^N E_i$.

Considering that the value E_i cannot be less than zero, then according to the mathematical law of additivity, the lower the value of the energy of the i devices E_i , the lower the value of

the energy that is spent by the robot while covering the entire distance E_{np} . This means that the fewer devices connected to the robotic platform, the less energy they will spend during a certain operating time. Thus, it has been proved that the device developed in this scientific work meets the requirements of energy saving.

Secondly, we will prove that the energy consumption will be the least, with the minimum distance traveled by the robot. Suppose that the robot moves uniformly, because the time for gripping the object does not depend on the path of movement of the robot, but remains unchanged regardless of which path was traversed by the robot. Then the following formula for determining the distance is valid: $S = vt$, where t – the time during which the movement is carried out at a given speed of robot v .

Then $v = \frac{S}{t}$. Taking into account the energy of motion $E = \frac{mv^2}{2}$, we get that $E = \frac{m(\frac{S}{t})^2}{2}$. This means that the less the path traveled by the robot, the less energy will be spent by the device. Which it was necessary to prove. Thus, at this point of work, it was proved that the developed system meets the requirements of energy saving and has significant advantages in this.

5. Analysis of similar systems

Minimizing energy costs is always an urgent problem. Many scientists are dealing with this issue, and among robotics, not least. Naturally, there are many studies on similar topics, some of which can be considered as distant analogs of the developed system.

So, for example, Taniguchi et al. [9] considers autonomous planning based on spatial concepts for guidance in the house using service robots. To complete the tidying task, the robot must recognize the environment, evaluate objects to clean while moving, and determine where to place objects in three-dimensional (3D) space.

The proposed tidy up planning method is formulated to first select the object whose tidied place is the most defined. In the spatial concept model learned in the tidied environment, the likelihood is the highest when each object position is tidied. Therefore, when tidying up one object from scattered objects, the object with the highest likelihood is selected first. In this study, tidying up implies moving the positions of the observed objects to increase the likelihood of the spatial concept mode.

First, the robot detects cluttered objects while moving. Next, if multiple scattered objects are detected, the robot decides on an object and a place to be tidied up. The robot estimates simultaneously the order and positions of the objects to tidy up from the multiple objects it observed in the cluttered environment. Moreover, the robot can determine if the tidied position of the object is unknown from Equation. If unknown, the robot can ask where to move the object.

Finally, the robot performs motion planning. After both estimating the tidied positions and planning the object order by Equation, the robot is required to accurately manipulate the objects to move them. The developed system uses the MoveIt! framework for the motion planning of the robot arm when grasping an object. Furthermore, the robot moves to the appropriate search positions while performing self-localization based on its map and observations.

When the robot finishes tidying an object, the state of the other objects may have changed

owing to external factors. To deal with such scenarios, it is possible to sequentially plan the tidy up task by redoing the object detection after tidying each object.

Approach is divided into learning and planning phases. In the learning phase, the robot observes the tidied environment. Spatial concepts are formed from observed data regarding objects and their 3D positions by multimodal learning. In the planning phase, the robot observes scattered objects in the environment, and estimates the order and positions of objects to be tidied up based on the spatial concepts formed in the learning phase.

According to the above flow, the robot tidies up a selected object to the target position repeatedly. In the case of an unknown object, the robot tidies it up by asking the user about its place name.

In contrast to the developed system, the reduced one is much more advanced, a robot that has undergone machine learning scans the surrounding space using a camera and builds the shortest paths based on the data obtained and built-in algorithms. On the one hand, such a system is more profitable in terms of finding goals and shortest paths in advance, on the other hand, it requires much higher development costs, in contrast to the one given in the article.

In a large number of works, the construction of trajectories for robots with obstacle detection is considered. So, for example, in work [10] for the study of space, an algorithm is used, based on the algorithms of the robot's movement "Spiral" and "Movement along the wall".

Algorithm "Spiral" is a movement along a circle with increasing radius. "Movement along the wall" is an algorithm in which the robot moves along the perimeter of space. Unlike the "Spiral" algorithm, in this algorithm the robot can go around obstacles. The proposed algorithm for learning a mobile robot in order to detect obstacles is a modified combination of the two algorithms described above. The learning algorithm consists of three parts. The first part can be called the main one – it is an algorithm for passing one circle (figure 2). The robot begins its exploration of space from a possible perimeter, and with each circle it narrows it down to a radius less by one. In order for the robot to narrow the radius with each passed circle, the second part of the algorithm is needed – the algorithm for starting the study of a new circle (figure 3).

After the end of the movement according to this algorithm, the robot needs to check the space for unexplored areas (figure 4). In the case when there are few obstacles and their shape is simple, the robot explores the space in one pass. Otherwise, there will be dark areas unknown to the robot. To prevent the robot from spending a lot of time on the way to such a site, it is necessary to use the algorithm for finding the shortest path. There are many obstacle avoidance algorithms. The most efficient is Dijkstra's algorithm if we consider the classical algorithms for solving the problem.

In [11], to solve the trajectory planning problem, an approach is used based on artificial neural networks of Hopfield, which belongs to intelligent algorithms – fuzzy and genetic algorithms, the main advantage of which is the speed of calculation with a moderate load on onboard computing complexes. However, the use of the specified mathematical apparatus imposes significant restrictions on the minimum system requirements of the MR computing platform, which excludes the possibility of implementing the method on devices with limited computing resources. For example, AVR or STM microcontrollers with a processor capacity of less than 32 bits, which are also a significant drawback.

Based on everything discussed above, we can conclude that the developed system is really

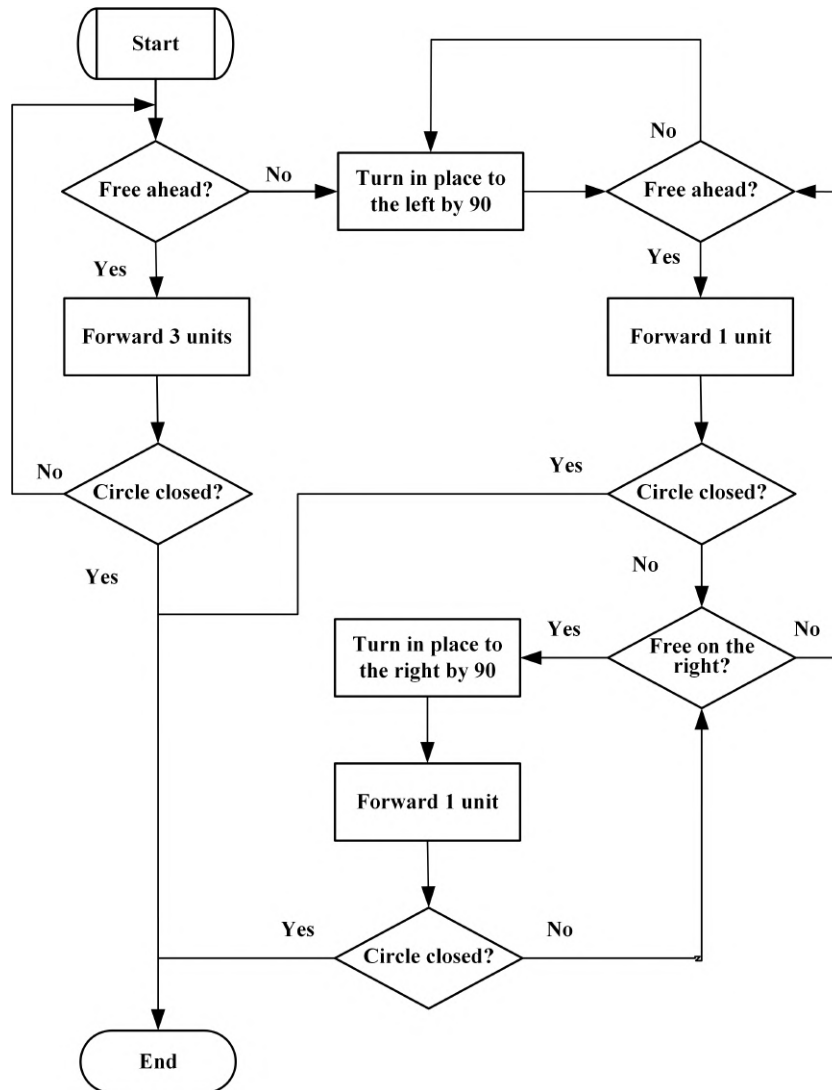


Figure 2: Algorithm for passing one circle.

relevant, having no exact analogs and showing an advantage over others for the set goals.

6. System implementation

By connecting the Arduino board to power, the firmware starts to actively execute. The microcontroller is configured so that the bootloader is responsible for the control functions at system startup. First, the loader checks for 1-2 seconds, the user will not start sending a new program. If the reprogramming process is started, the sketch is loaded into memory and control is transferred to it. If there are no new programs, the loader executes the previously saved program [12].

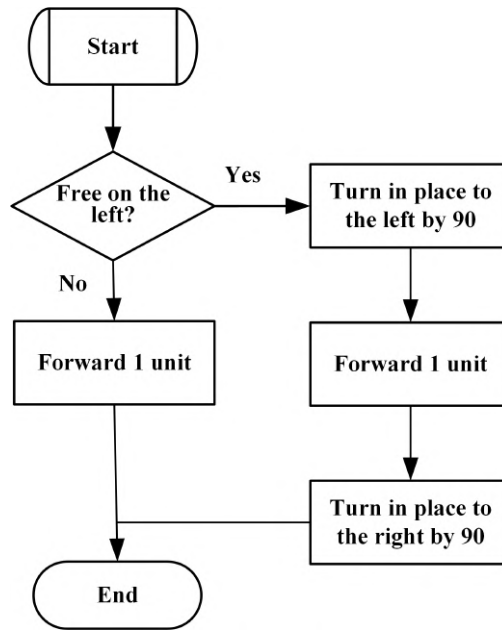


Figure 3: Algorithm for starting the study of a new circle.

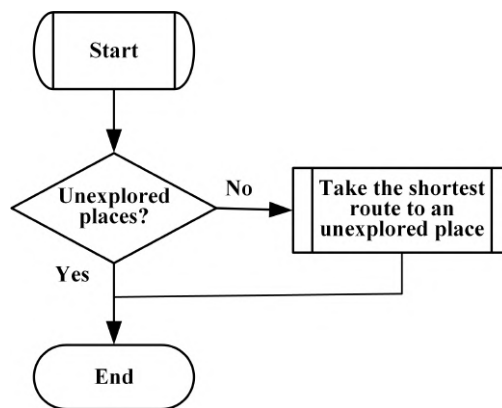


Figure 4: Algorithm for check the space for unexplored areas.

After starting the execution of the program, the Arduino performs a number of routine operations of initializing and setting up the environment, and only then proceeds to execute the same code that is contained in the sketches. Thus, Arduino eliminates the need to remember all the details of the microprocessor architecture and concentrate on the tasks at hand.

The robotic platform contains a large number of sensors and actuators, which can be controlled by the AlphaBot.h library, which can be downloaded from the official AlphaBot website [13]. The instance used to send commands from the sketch is Car1, and the method implemented by the sketch is: SetSpeed(), Forward(), Brake(), Backward(), Left(), Right(), LeftCircle(), RightCircle(), MotorRun().

A set of functions for controlling a servo drive is provided by the Servo.h library:

- `servo.attach()` – indicates the pin to which the servo control wire is connected;
- `servo.write()` – set the turning angle;
- `servo.writeMicroseconds(uS)` – a value is passed to control the servowrite in microseconds (uS), setting the rotation angle to this value;
- `servo.read()` – reads the current angle of rotation of the servo;
- `servo.attached()` – determines if there is a binding to the servo via pin;
- `servo.detach()` – disconnect a pin from the Servo library.

By downloading the program, the Arduino allows the code to participate in the initialization of the system. To do this, the microcontroller is given commands that it will execute at the time of the boot and will no longer return to them (that is, these commands are executed only once at system startup). And it is for this purpose that a block is allocated in the program in which such commands are stored – `void setup()` [14, 15]. In the function, the necessary properties are set for connecting the configuration of the ultrasonic sensor, setting the speed of the robot, specifying the pin for connecting to the servo drive.

```
void setup()
{
  ProximityConfig();
  UltrasonicConfig();
  myservo.attach(9);
  Serial.begin(9600);
  Car1.SetSpeed(150);
}
```

Variables that are used in the program for the functioning of the robot:

- `run` – to set the start of the robot's functioning;
- `side` – to determine the side to which the robot is returning;
- `distance` – to determine the distance to the obstacle at the present time;
- `spacing` – to save the old value of the distance to the obstacle;
- `cart` – to determine the presence of a "basket";
- `pos_last_point` – to save the distance to the point where the object was taken;
- `time_turn` – to save the time for which the reversal is carried out;
- `pos_cart` – to save the distance to the "basket".

Be sure to after `setup()` there is a `loop()` function – the entry point to the program, which ensures the execution of commands while the Arduino board is turned on. This function contains all commands that will be executed cyclically. Starting from the first command, the microcontroller performs all subsequent steps to the very end and immediately returns to the beginning to repeat the same instruction. Checking the distance in the program is performed by the `check distance()` function. It allows you to check the driving distance in the range from 2 to 30 cm.

```

int check_distance()
{
    if((1 < Distance) && (Distance < 38))
    {
        return false;
    } else {
        return true;
    }
}

```

The capture of objects is carried out by the Servo() function.

```

void Servo()
{
    for (pos = 0; pos <= 180; pos += 1) {
        myservo.write(pos);
        delay(15);
    }
    delay(1000);
    Distance_test();
    spacing = Distance;
    Car1.MotorRun(10,10);
    Car1.Brake();
    delay(1000);
    Car1.MotorRun(0,0);
    Car1.Brake();
    for (pos = 180; pos >= 0; pos -= 1) { myservo.write(pos);
    delay(15);
    }
}

```

A cycle is started that opens the claw 180 degrees. Next, a delay is called to start the motors. The robot moves forward 4 cm. The next cycle is responsible for closing the claw.

The object_check() function checks the delight of the subject.

```

int object_check()
{
    Distance_test();
    if (spacing > Distance)
    {
        return true;
    } else {
        return false
    }
}

```

If the distance to the object is less than the value of the distance that was before the capture, then it is considered that the object was taken by the robot. Otherwise, the object defines the wall.

To leave an item in the “basket”, the program calls the leave_object() method. Obstacle sensors are controlled by the sensor_check() function. The movement of the robot backward is set and a cycle is performed until the sensors notify about the object. Upon completion, the system performs a 180-degree turn in place.

```

void sensor_check()
{
  Car1.Motor(-150,-150);
  Car1.Brake();
  while(true)
  {
    RSensor = digitalRead(RSensorPin);  LSensor = digitalRead(LSensorPin);
    if (LSensor == HIGH && RSensor == HIGH)
    { Car1.Motor(0,0);
      Car1.Brake();
      delay(1000);
      Car1.RightCircle(2000);
      Car1.Brake();
      break;
    }
  }
}

```

The Distance_test() function writes the distance to the obstacle in the Distance variable.

```

void Distance_test()
{
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  float Fdistance = pulseIn(ECHO, HIGH);
  Fdistance = Fdistance / 58;
  Distance = Fdistance;
}

```

The definition of the shortest path to the “basket” is provided by the road_to_busket() function. Changing side determines the side to return to (1 – left, 2 – right).

The road_to_last_point() function returns the work to the point along the shortest path where the item was taken.

To return to the starting position of the work, the road_to_start_point() function is called.

The turn() function determines the side in which to turn and carries out movement in a certain direction.

```

int turn(){
  if (side == 1) {
    Car1.RightCircle(1000);
    Car1.Brake();
    Distance_test();
    if (check_distance() == true) {
      Car1.RightCircle(1000);
    }
  }
}

```

```

Car1.Brake();
side = 2;
return true;
} else {
return false;
}} else {
Car1.LeftCircle(1000);
Car1.Brake();
Distance_test();
if (check_distance() == true) {
Car1.LeftCircle(1000);
Car1.Brake();
side = 1;
return true;
} else {
return false; }}}

```

7. Conclusion

On the basis of the Arduino family platform, an object collection system was created that is able to: estimate the distance to an obstacle using an ultrasonic sensor; perform the movement to a given point; grab an object no larger than 5.5 cm; deliver the item to the item collection point; determine the shortest path to a certain point. The movement to the right is performed in the same way as the algorithm of movement to the left.

There are limitations associated with the use of an ultrasonic distance sensor:

1. Partial displays can distort the measurement results (this can be caused by curved or inclined surfaces with respect to the direction of emission of the signal).
2. Objects made of sound-absorbing, insulating materials, with a woolen surface can weaken the signal.
3. The size of the object affects the signal quality. The smaller it is, the weaker the reflected signal.
4. High humidity (rain, snow) contributes to signal distortion.

The system can be improved with a movable ultrasonic sensor stand, which will help determine the width of the object in order to calculate the distance to avoid an obstacle.

The scientific novelty of the work lies in the fact that a new method of collecting objects in the room and a system have been developed that allows organizing the collection process with minimal energy consumption.

The practical significance of the work lies in the fact that the software has been developed, which makes it possible to assess the adequacy and effectiveness of the developed method in the real work of the system.

References

- [1] G. Kirichek, S. Skrupsky, M. Tiahunova, A. Timenko, Implementation of web system optimization method, CEUR Workshop Proceedings 2608 (2020) 199–210.
- [2] M. Tiahunova, O. Tronkina, G. Kirichek, S. Skrupsky, The neural network for emotions recognition under special conditions, CEUR Workshop Proceedings 2864 (2021) 121–134.
- [3] S. Semerikov, I. Teplytskyi, Y. Yechkalo, A. Kiv, Computer simulation of neural networks using spreadsheets: The dawn of the Age of Camelot, CEUR Workshop Proceedings 2257 (2018) 122–147.
- [4] G. Kirichek, V. Harkusha, A. Timenko, N. Kulykovska, System for detecting network anomalies using a hybrid of an uncontrolled and controlled neural network, CEUR Workshop Proceedings 2546 (2020) 138–148.
- [5] Waveshare-Electronics, AlphaBot, 2021. URL: <http://www.waveshare.com/wiki/AlphaBot>.
- [6] T. Pan, Y. Zhu, Designing Embedded Systems with Arduino: A Fundamental Technology for Makers, Springer, Singapore, 2018. doi:10.1007/978-981-10-4418-2.
- [7] M. Boranbaev, Development of a robot for transporting small-sized objects based on a AVR microcontroller (Razrabotka robota dlya transportirovki malogabaritnykh obyektov na baze mikrokontrollera AVR), Molodoy uchenyy (2016) 277–286.
- [8] R. Chase, A. Pandya, A review of active mechanical driving principles of spherical robots, Robotics 1 (2012) 3–23. URL: <https://www.mdpi.com/2218-6581/1/1/3>. doi:10.3390/robotics1010003.
- [9] A. Taniguchi, S. Isobe, L. E. Hafi, Y. Hagiwara, T. Taniguchi, Autonomous planning based on spatial concepts to tidy up home environments with service robots, Advanced Robotics 35 (2021) 471–489. doi:10.1080/01691864.2021.1890212.
- [10] O. V. Avseeva, M. V. Larina, Development of a learning algorithm for a mobile robot in order to detect obstacles in a confined space (Razrabotka algoritma obucheniya mobil'nogo robota v tselyakh obnaruzheniya prepyatstviy v zamknutom prostranstve), Voronezh State University of Engineering Technologies 79 (2017) 65–67. doi:10.20914/2310-1202-2017-3-65-67.
- [11] B. S. Yudintsev, Synthesis of a neural network path planning system for a group of mobile robots (Sintez neyrosetevoy sistemy planirovaniya trayektorii dlya grupp mobilnykh robotov), Systems of Control, Communication and Security (2019) 163–186. doi:10.24411/2410-9916-2019-10406.
- [12] J. Blum, Exploring Arduino®: Tools and Techniques for Engineering Wizardry, Wiley, 2013.
- [13] O. V. Antipov, Robot platform RedBoard (Robot-platforma RedBoard), 2015. URL: <https://cxem.net/uprav/uprav66.php>.
- [14] V. Petin, Designing with an Arduino controller (Proyekty s ispolzovaniyem kontrollera Arduino), BHV-Peterburg, 2019.
- [15] V. Bista, Understanding and design of an arduino-based pid controller, 2016. doi:10.25772/790F-JP22.