The control system for a mobile robot using Arduino mega 2560 with GPS and obstacle detection systems

P Y Cherepanov, P A Romanov, S V Zharikov, T R Mazhitov

Ural State University (National Research University), 76, Lenin prospekt, Chelyabinsk, 454080, Russia

E-mail: pavelphone@gmail.com

Abstract. This research shows the implemented software and hardware solutions of the remote control systems for a robotic vehicle. This paper partially solves the problem of rapid prototyping and debugging of motion algorithms of the robotic vehicle. The composition and brief characteristics of the main components used in this system are also given in this paper. The methods (the programming languages that were used to create the software of this vehicle) are also described in this research. The results and the basic goals for the further modernization of this vehicle are also given in this paper. Main conclusions also described in this paper.

1. Introduction
According to the statistics on the roads of Russia more than 25 thousand human die in an accident every year [1]. The cause of 80% of all accidents, with a fatal outcome, is the human factor. While the technical factor took place to be only in 2% of all accidents, with lethal outcome. Automated vehicle is fully capable to eliminate the human factor, and thus save lives of more than 20 thousands citizens every year.

The main goals:
• to develop the simplest remote control system for a robotic vehicle,
• to develop the simplest navigation systems for a robotic vehicle.

2. Relevance
The robotic vehicle is able to significantly simplify the workflow in a production. There is no need for a driver for electric cars with an autopilot, which can significantly save costs. The robotic trolley is capable of delivering cargo from point A to point B without a driver. One dispatcher is sufficient to control the operation of several trolleys [2]. Also, it is completely impossible to receive various injuries while transporting goods. And in harmful production, the robotic trolley is simply irreplaceable [1, 3, 4]. As a rule, such systems need to be adjusted individually for each task, and this is a long and time-consuming process [5, 6]. Large corporations like Google [7], BMW, Apple, Ssang Young, Kamaz are actively engaged in the development, related to the automation of a motion a vehicle [8]. Another reason for such developments is the economic component. Indeed, companies that deal with cargo transportation do not need to pay truck drivers [9].

To date it has not shown a car that could freely move around without a driver. The research presented in this article greatly simplifies the process of debugging the algorithms for the motion of an automated vehicle, which can significantly accelerate the development process of a robotic car.

The remote control system of a robotic vehicle consists of two separate modules: remote control and
robotic vehicle (RV).

3. Hardware solutions
The control panel is a portable Rasberry Pi computer (Figure 1) with a Bluetooth USB adapter connected via a USB port. The Rasberry Pi computer is equipped with a Raspbian operating system [10].

The robotic vehicle includes the following equipment: a support platform, two traction motor-reducers, a 24V battery, a 24V / 220V semiconductor inverter, a 220V / 5V power supply, a microprocessor module (Figure 2), two microprocessor module protection modules from high currents, relay module, Bluetooth module [11], connecting wires, encoders [12].

![Figure 1. Microcomputer Raspberry Pi.](image1)

![Figure 2. Appearance of the robotic vehicle.](image2)

The use of encoders allows to move the robot with great accuracy. Encoders are attached to the wheel with a special clutch (Figure 3). The relay module is used in the circuit for controlling electromagnetic brakes [2].

Using in this work a computing platform for fast prototyping Arduino2560 with two drivers (Figure 4), it is possible to compile various algorithms for the operation of an automated trolleys, write on these program algorithms, and write these programs, without dismantling main board, via USB directly from PC [13].

![Figure 3. Encoder mounting location.](image3)

![Figure 4. Control system scheme.](image4)

4. Software solutions
Two programs were developed for the remote control system of the RV. The first program for the control panel was written in Processing, an open programming language based on Java [14].

The second program for RV was written in the free software environment Arduino Software, based on Processing and other open software products [15]. The environment compiles and programs the microprocessor control board of the RV. The work algorithm of the control system shown in Figure 5.

![Figure 5. Algorithm of the control system.](image)

4.1. GPS System
The basic principle of the system's operation lies in the use of satellites that are constantly in orbit around the earth, transmitting the anchor signals. Such satellites are not less than 24. The locations of the GPS receiver (Figure 6) are calculated on the basis of measuring the delay in the passage of a radio signal from several satellites and the calculation of geographic coordinates and altitude based on these measurements. The signal of each satellite contains a pseudo-random Number code (PRN code), ephemeris and almanach. The PRN code serves to identify the satellite-source of the signal [4].

Ephemerides - coordinates of this satellite in near-Earth space. They are transmitted to the satellite from the control center.

Almanac contains data about where the satellites should be at the moment and their state - working or not.

This set of data is transmitted by the satellite with a high frequency. To determine its location, the GPS receiver electronics calculates the difference in the time of sending the signal from the satellite with the time it was received on Earth. By this difference, the distance to a specific satellite is calculated. The basis for the idea of determining the coordinates of the GPS receiver is the calculation of the distance from it to several satellites, the location of which is considered known (these data are contained in the almanac received from the satellite). Obviously, the signal of one satellite is not enough to determine its location - it is possible to find somewhere on the surface of a sphere with a calculated radius. For two satellites, the place is already defined as the intersection line of two spheres - in general it is a certain circle. Theoretically, three satellites are already enough - calculations will give two possible points, one of which will be located high above sea level and will therefore be discarded. However, in fact, the distance is calculated with some error, the effect of which is greater, the smaller the angles between the directions from the successor to the satellites. The error in determining the coordinates is corrected by calculating the signals of other satellites. The more satellites the receiver uses, the more accurate its location can be determined. Figure 7 shows the location of the intelligent robotic vehicle (green dot).

In the upper right corner, there are coordinates of the vehicle. On the left in the window is the scale of the map, the position of the mouse cursor, the map provider. This program is written in Java [16].
4.2. Obstacle detection system

The base of the obstacle detection system is Ultrasonic sensors (Figure 8). In the basic mode of operation, the signal from the sensor will be continuously fed to the microcontroller, which converts it to a certain numerical value. With smooth motion without any obstacles, the signal will change slightly, this is due to the fact that it rotates horizontally from 0 to 180 degrees in front of you and sees everything that appears on its way through 1.5 meters, which means if we are move on a relatively flat surface, the signal from the range finder will detect a "static object" that does not move.

If an obstacle such as "break" occurs, the signal from the sensor will dramatically change its value, indicating a sharp increase in the distance, for this case, stopping or detouring the obstacle is provided.

The ultrasonic sensor will recognize the usual approaching surfaces and objects so that there is no collision with our tool. This is achieved by the usual setting: experimentally determined the minimum allowable distance to the object so that there is no collision (about 1.5 m), while recording data from the sensor, a specific number corresponding to this distance is determined, and the microcontroller will simply stop the cart in the future when this value is reached.

To connect to the measurement circuit, the sensor is equipped with 4 leads. Two of them serve to connect the power. Trig input is triggered by a trigger pulse, and a signal is output from the Echo output, the duration of which is proportional to the measured distance [17].

The algorithm for working with the sensor shown in Figure 9.

To start the measurement, a trigger pulse of 10 microseconds is applied to the Trig input.

- After detecting the trigger pulse, the sensor emits a burst of 8 pulses at a frequency of 40kHz.
By detecting the reflected signal, HC-SR04 sets the high level at the Echo output. The duration of this state in microseconds will be proportional to the measured distance in meters.

trigger pulses are recommended to be generated once every 50ms.

All that is required from the controlling microcontroller is to generate a trigger pulse and measure the value of the echo. In practice, it is sufficient to use a timer incremented at a frequency of 1 MHz, (1 microsecond). If an echo is detected, the timer should be started and the counter value read through the negative edge. Given the values of the values, a 16-bit timer is required.

To obtain the exact distance in acceptable units, an additional calculation is necessary. It must take into account that the measured value includes the forward and reverse propagation paths of the signal. Accordingly, to calculate the distance, the time should be divided by 2. It should also take into account the speed of ultrasound in the air. As a result, the general formula will be as follows:

\[ L = \frac{t \times V}{2}, \]

where \( L \) is the distance in meters; \( t \) - echo-pulse time in seconds; \( V = 340 \text{ m/s} \) - the speed of ultrasound.

To simplify the calculations, it is much more convenient to apply a formula of the form:

\[ l = \frac{t_{us}}{58.2}, \]

where \( l \) is the distance in centimeters, \( t_{us} \) is the time in microseconds.

In the process of research, a program was written that tracks the distance for obstacles in real time. Due to the servo drive, the sensor scans the space. When an obstacle appears the program signals this to the robot. There is an interface of the program when there is no obstacles in Figure 10, and with an obstacle in front in Figure 11.

![Figure 10](image.png)

**Figure 10.** Interface of the obstacle detection system (no obstructions).

![Figure 11](image.png)

**Figure 11.** Interface of the obstacle detection system (an obstacle appeared).

5. Conclusion

As a result of the work done, a simple system for remote control of a robotic vehicle, the GPS system and Obstacle Detection system were developed. The use of the platform Arduino Mega 2560 allowed the rapid prototyping and debugging of motion algorithms of the robotic vehicle.

The simplest modes of motion of a robotic vehicle have been developed. Based on these modes, decision algorithms will be built for further development of the automated system. In the future it is planned to equip the robot with a Laser Scanner Lidar for more accurate construction of the map of the space.
References

[1] Statistics of road accidents. URL: stat.gibdd.ru

[2] Lobov S A, Mironov V I, Kastalskiy I A and Kazantsev V B 2015 Combined use of command-proportional control of external robotic devices based on electromyography signals Sovremennye Tehnologii v Medicine 7 (4) 30-37

[3] Sew-eurodrive. Maxolution hybrid AGV. Retrieved from: https://share.seweurodrive.de/media/index.php?module=mediaarchive_view_detail&id=180.html/

[4] Principles of satellite navigation Retrieved from URL: https://www.glonass-iac.ru/guide

[5] Cherepanov P Y and Romanov P A 2017 Development of an automated system for the transport of light goods Science, technology and education 1 (31) 42-47

[6] Garcia E, Jimenez M A, de Santos P G and Armada M 2007 The Evolution of Robotics Research From Industrial Robotics to Field and Service Robotics IEEE Robotics & Automation Magazine MARCH 90-103

[7] Google self-driving car project. Retrieved from: https://waymo.com/

[8] Cherepanov P Y and Romanov P A 2017 The main directions of development of robotics Modern innovations 1 (15) 29-38

[9] 24space.ru. The mission of Restore-L will show new service technologies. Retrieved from: http://24space.ru/1488-missiya-restore-l-pokazhet-novyetehnologii-obsluzhivaniya.html/

[10] Petin V A 2015 Mikrocomputers Raspberry Pi. Practical guide (Saint Petersburg: BHV-Petersburg)

[11] Asmakov S 2013 Bluetooth interface: we will understand with nuances ComputerPress 3 (279) 34-36

[12] Vorobyev A A, Petrukhin A V, Krivonozhkina P S and Pozdnyakov A M 2015 Exoskeleton as a new means in habilitation and rehabilitation of invalids (Review) Sovremennye Tehnologii v Medicine 7 (2) 185-192

[13] Blum J 2015 Exploring Arduino Tools and Techniques for Engineering Wizardry (Saint Petersburg: BHV-Petersburg)

[14] Vasilyev A N 2013 Java with programs and examples (Saint Petersburg: Nauka i Tehnika)

[15] Site for programming in Arduino. Retrieved from: https://www.arduino.cc/

[16] Schildt H 2014 Java The Complete Reference 8th ed.

[17] Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino. URL: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/