Journal of Technology and Exploitation in Mechanical Engineering
Vol. 8, no. 1, pp. 32–41, 2022
DOI 10.35784/jteme.3147

ISSN 2451-148X
Available online at: www.jteme.pollub.pl

Research article

REMOTE-CONTROLLED TWO-WHEELED SELF-BALANCING ROBOT

Przemysław Filipek[1]

[1] Lublin University of Technology, Faculty of Mechanical Engineering, Nadbystrzycka 36, 20-618 Lublin, Poland
corresponding author: p.filipek@pollub.pl

Received: 2022-10-19 / Accepted: 2022-11-03 / Published: 2022-12-08

ABSTRACT

The article presents the design and fabrication of a remotely controlled two-wheeled self-balancing robot, along with its mobile control by means of an application on a phone displaying basic parameters in a terminal. A 3D model made in SolidEdge software, the electronic components used to build the robot and electronic schematics and software are presented. Real-world tests were conducted that demonstrate the robot's operation. The robot can be used for educational and commercial purposes, as well as in the human home environment.

KEYWORDS: remote-controlled, self-balancing robot, 3D model

1. Introduction

Thanks to modern and increasingly advanced technologies, there has been a flourishing of electronics that enabled automation of processes and ease of modification of many devices as well as their functions. Over time, to meet the needs of the population, robots have become more and more present in our lives. A big step was the construction of balancing robots of various designs due to their main problem which is to maintain a point of balance. They served as means of transportation, maintained balance according to the terrain and also became similar to humans, e.g. the humanoid robot ASIMO [1]. The increasingly lower cost of electronic components and the intuitiveness and ease of use have made them present in many fields. Balancing robots are becoming an interesting object of research due to the use of various control methods and for their system instability condition [2]. The advantages of robots balancing robots are size, ease of transporting small objects, fast ability to change the direction of movement, communication over a distance and also the possibility of application of modern sensors.

2. Design assumptions

To begin the process of building a self-balancing robot, it is necessary to formulate design assumptions:

- The robot will have two wide rubber wheels to help it maintain its balance and two DC motors with encoders to help control the robot;
- An accelerometer sensor will be used to maintain balance with MPU6050 gyroscope [3];
- The main controller of the robot is an Arduino Uno R3 to which the electronics;
- The robot will be powered by rechargeable batteries in the in the basket which can be recharged in case they are discharged;

- The robot will be remotely controlled by an application on your phone using a Bluetooth module HC-05;
- In the case of attaching an additional holder, it will be able to carry for example, a bottle of water with a maximum weight of up to 0.5 kg or other small objects;
- It can be used for educational purposes or as an aid to everyday use such as a table or shelves;
- Lightweight components will be used for construction, which will give ease of balance and movement;
- Electronics will be mounted as low as possible so that its center of gravity was just above the motors;
- The parameters of the robot will be displayed in the application terminal located on the phone;
- The robot will be small in size with a maximum height up to about 30 cm, width of about 15 cm and weight up to 3 kg.

3. Design

The design of the remote-controlled two-wheeled self-balancing robot includes a 3D model, electronic components including an electronic schematic, and software.

3.1. 3D Model

The design of the remotely controlled two-wheeled self-balancing robot was created in SolidEdge software. It's a program that is easy and very intuitive to use, allowing you to make a variety of parts and then put them together. Thanks to its powerful features, you can use many ready-made parts, such as motors or electronics, found in the program's libraries. It allows you to make a 3D model which makes it easier to make a robot in reality. Allows you to quickly design and size parts and easily modify them (Fig. 1).

After designing all the necessary parts, an assembly was made in the SolidEdge program (Fig. 2).

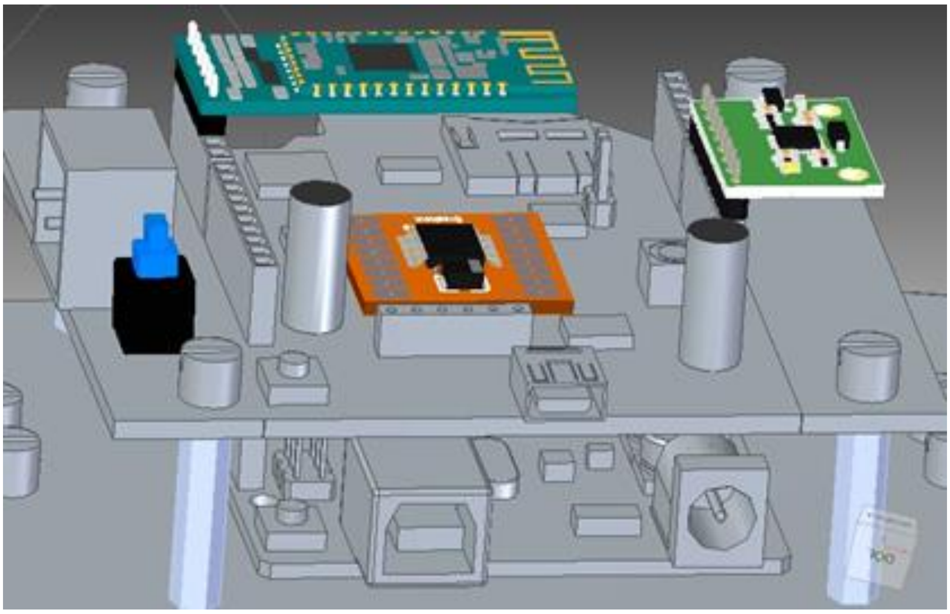

Figure 1. Arduino Uno R3 with expansion board and placed electronics components

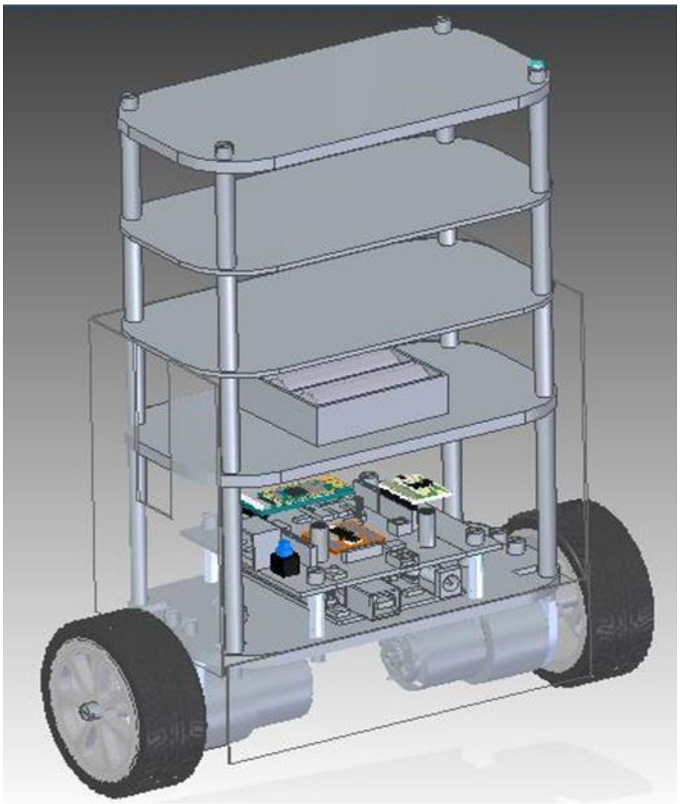

Figure 2. Assembling a remote-controlled two-wheeled self-balancing robot

3.2. Electronics

The electronics include such parts as:

- Arduino Uno R3 - this is the robot controller and at the same time a universal controller controlling balance and movement [4]. Its great advantage is the ability to connect many sensors, easy programmability and many libraries for handling components (Fig. 3);
- Pololu TB6612FNG - is a two-channel DC motor controller, allowing to control DC motors of a robot. It is distinguished by a supply voltage of 4.5 V to 13.5 V and a current per channel of 1 A (Fig. 4);
- Gyroscope and accelerometer MPU6050 - is a 3-axis acceleration sensor characterized by reading motion in three dimensions (Fig. 5);
- GB37 DC motors with encoder and 30:1 gearing - feature a 12-volt power supply, a current consumption of 120 mA without load, and a maximum speed of 333 rpm and a torque of 0.35 Nm. When operating or loading motors their maximum speed is 250 rpm and current consumption is 1 A (Fig. 6);
- Bluetooth module HC - 05 - is a module used to communicate between the robot and the phone. Data is sent and received via the UART serial interface (Rx and Tx pins). It can be powered by 5 V or 3.3 V. Its maximum range is 10 m (Fig. 7).

Figure 3. Arduino Uno R3 [5]

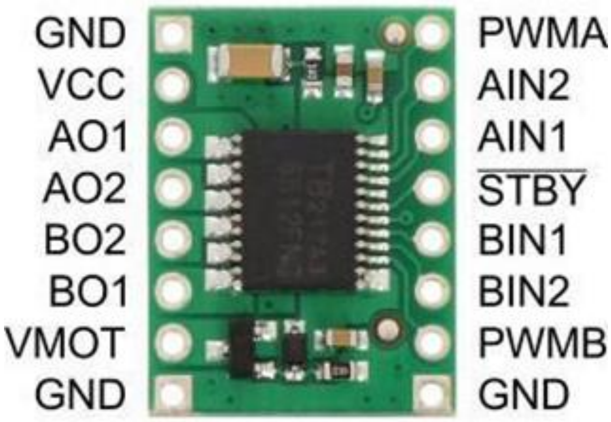

Figure 4. Pololu TB6612FNG motor controller [6]

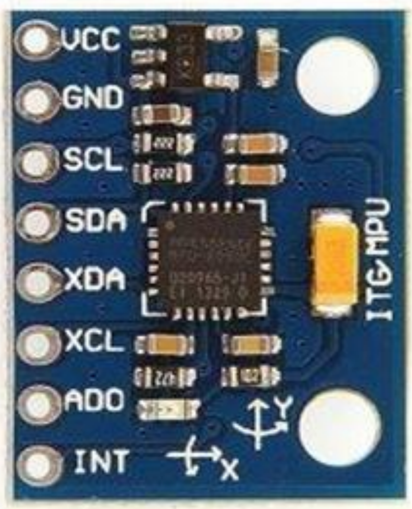

Figure 5. MPU6050 gyroscope and accelerometer [7]

Figure 6. DC motor with encoder [8]

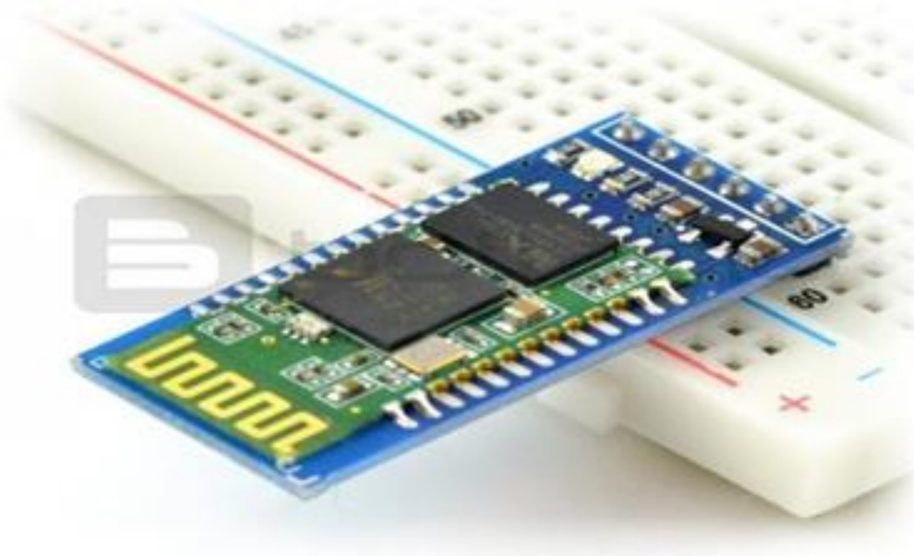

Figure 7. Bluetooth module [9]

The electronic schematic of the remote-controlled two-wheeled self-balancing robot was made in EasyEDA software (Fig. 8).

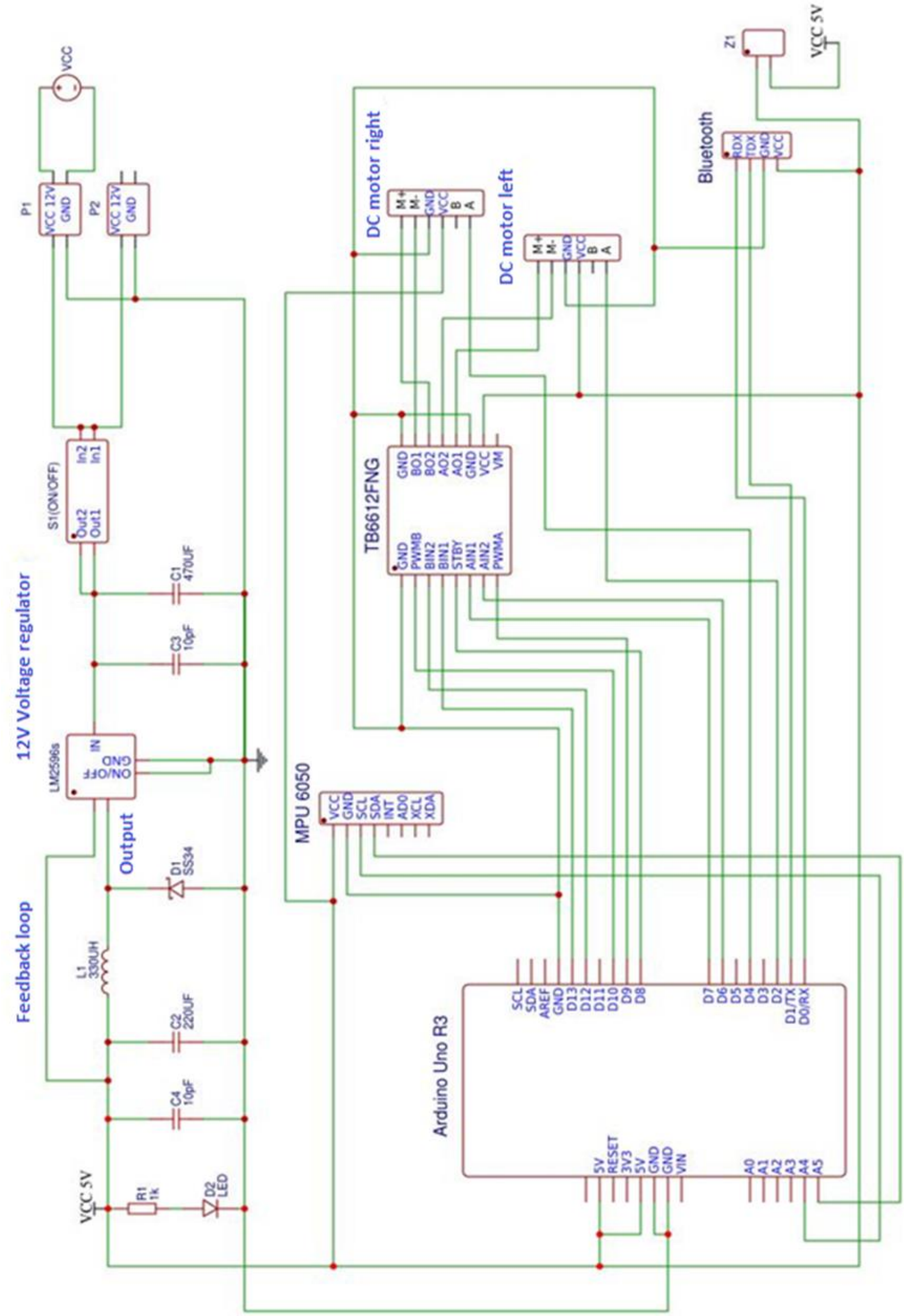

Figure 8. Electronic schematic of the remote-controlled two-wheeled robot self-balancing

3.3. Software

The program for the remote-controlled two-wheeled self-balancing robot was written in the Arduino IDE environment using the C or C++ language. Its purpose is to keep the robot balanced, control it and display parameters. In addition, it should be resistant to external factors such as deliberate knocking out of balance, uneven ground, and driving at different degrees of inclination.

Sample program code with description:

A program snippet about controlling the direction of the robot:
void impulse () // The function responsible for moving the robot in different directions by counting pulses from the encoders and adjusting the PWM signal for the motors. Assign the variables *zl* and *zp* as the corresponding positions for the encoder located at the motor so that it can make the appropriate movement to achieve the set direction.

{ zl = positionenkoderalew;
zp = positionenkoderaldev;
positionenkoderalewo = 0; // Reset encoder position so that it counts from the initial, zero position.
positionencodereft = 0;
if ((pwm11 < 0) && (pwm2p < 0)) // The "if" condition specifies that the following must be such assumptions are met for the PWM signals of both motors. In the given case, the signal PWM on the motors is negative which causes a backward movement and thus both motors rotate in the opposite direction and the robot travels backwards.
{ zp = -zp; // minus means the opposite direction of motor rotation.
zl = -zl; }
else if ((pwm11 > 0) && (pwm2p > 0)) // PWM signal positive, motors rotate forward and the direction of the robot is forward.
{ zp = zp;
zl = zl; }
else if ((pwm11 < 0) && (pwm2p > 0)) // The PWM signal of the first motor is negative, so the motor rotates to the left. The PWM signal of the second motor is positive motor rotates to the right side. The robot rotates clockwise.
{ zp = zp;
zl = -zl; }
else if ((pwm11 > 0) && (pwm2p < 0)) // The PWM signal of the first motor is positive, so the motor rotates clockwise. The PWM signal of the second motor is negative motor rotates to the left side. The robot rotates to the left.
{ zp = -zp;
zl = zl; }
stopright += zp; // Define new variables to stop the robot at a given position read from the encoder.
stoplewo += zl;
impulspravo += zp; // The "+=" sign allows to add to the variable "impulspravo" variable "zp" which is the current position defined with the encoder in order to determine its movement.
impulspravo += zp;
sum = impulsprawo + impullewo; // Determine the sum of the number of pulses needed to move the robot.}

4. The actual performance of the robot

In the construction of a remote-controlled two-wheeled self-balancing robot, many factors play a very important role in order for it to work properly and perform its function. These factors include the proper design of the robot, the use of electronics, the center of gravity and considerations regarding its weight, dimensions and also the materials from which it will be made.

The robot weighs approx. 2.5 kg, is 30 cm high and 15 cm wide, and consists of four floors and a base. It has two wheels with an outside diameter of 65 mm and a width of rubber tires of about 25 mm, which gives a stable ride on various types of ground.

The wheels are attached to the DC motor shaft with a motor axle adapter and the components used to build the robot are lightweight but durable. The robot's floors and its body are made of Plexiglas, while the motor mounting brackets and the base on which the electronics are aluminum plates about 3 mm thick (Fig. 9).

Figure 9. Fabrication of the actual remote-controlled two-wheeled self-balancing robot - front view

5. Screening tests

The robot is powered by 3 rechargeable batteries with a capacity of 2900 mAh (2.9 Ah), while the sum of the currents in the system is 2 A which are included in the calculations. The time of continuous operation, or driving, is about 1 h and 28 min. The maximum working time up to discharging the batteries has been tested, and the results are about 80 minutes. error in the case of actual working time vs. calculated.

The control of the robot is carried out in the directions that have been assumed, i.e. forward forward, backward, left, right and left and right rotation. The control is very dynamic and the robot moves very smoothly and maintains its balance.

The maximum weight the robot can hold was tested by positioning a water bottle. First, a bottle with about 100 ml of water was placed where the robot behaved normally, then the weight was increased and as the weight increased there were more and more oscillations until a bottle with about 500 ml of water where the robot was barely able to maintain its balance. To further test the robot's balance, its height was taken into account with the number of floors. The optimal number of floors was 4 with a base. In the case of 6 floors where it was tested, the robot was driving very fast immediately after turning on and could not find the balance point, human assistance was needed by stabilizing it with hands. One of the next issues when moving real-world tests is the robot's center of gravity, the placement and weight of the components versus the robot's balance and control. To test this, one of the several heaviest components in the construction of the robot and this is a basket with three rechargeable batteries where holes have been bean-shaped holes have been made so that the basket can be moved in a given direction.

The best solution is to place the basket and other components at equal spaced from each edge, which gives the best performance. In the case of changing the position of the basket, the robot continues to balance, but the weight of these elements causes a slight tilting of the entire structure to one side, which is loaded at the moment. The above disadvantages can be very easily neutralized by using sinks with higher torque, selecting even lighter parts, a different design or arranging the components to get a taller and larger robot with additional features and even better performance.

The final stage of testing was to control the robot from a distance via the Bluetooth. The maximum communication distance was about 10 m where the robot responded to the use of the control buttons in the app. In the case of a greater distance it was not possible as well as in the case of floors where the robot was separated by a thick ceiling communication was broken. On the other hand, during a test between rooms where the only obstacle was a wall the robot behaved and drove correctly.

6. Comments and conclusions

The following comments and conclusions were made:

- Selected the appropriate values of the PID controller, that is, the members "P", "I", "D" so that the robot balances at rest and maintains balance while driving;
- Arranged the elements so that they are all in the axis of the robot and at equal preserved distances which results in its stability;
- Used an application that has a terminal that displays the robot's parameters and buttons responsible for movement in a given direction;
- The PWM signal of the motors is set to the right value, if the power is too low, the robot will topple, the robot will fall over;
- Ease of rapid expansion of the robot through a simple design that allows for modification of appearance as well as functions;
- Lightweight components have been used due to their significant impact of weight and preservation of the robot's performance;
- The range of control of the robot by means of an application on the phone and communication with the robot thanks to Bluetooth module HC - 05 is about 10 m, in the case of a test over a longer distance or between floors separating the robot through the ceiling communication and control is not possible;
- Through the use of batteries with a capacity of 2900 mAh, voltage of 3.7 V, current of 8,25 A, the working time of the robot is about 88 min. It is also possible to use more batteries by expanding the robot which will give a better and longer working time;
- The possibility of expanding the robot by using additional sensors such as. ultrasonic distance sensor, WIFI module, weight measurement sensor, GPS module. Each of the sensors is associated with a corresponding function allowing you to achieve control over a greater

distance or to avoid collision with some object. In addition, the weight measurement sensor allows you to determine the maximum value of the weight to be placed on the robot's floors. The GPS module, on the other hand, determines the coordinates geographic coordinates, i.e. the place where the robot is located, works in a similar way as GPS navigation. All this is linked to a change in software, but most importantly to fulfil the function of the balancing robot and also to use it in reality;

- The remote-controlled two-wheeled self-balancing robot can be used for educational, advertising and commercial purposes, as well as further development of the construction and software;
- Control of the robot is carried out using applications on commonly used cell phones which increases the versatility of its application.

7. References

[1] T. Szkodny, *Podstawy robotyki*. Wyd. Politechnika Śląska, Gliwice 2012.
[2] D. Cook, *Budowa robotów dla początkujących – zbuduj własnego pomocnika*. Wyd. Helion, Gliwice 2012.
[3] Gajek A. and Juda Z., *Czujniki*. Wyd. WKŁ, Warszawa 2008.
[4] R. Anderson and D. Cervo, *Arduino dla zaawansowanych*. Wyd. Helion, Gliwice 2014.
[5] "Arduino Uno R3 photo" [Online]. Available: http://akademia.nettigo.pl [Accessed: 15.06.2022].
[6] "Motor controller" [Online]. Available: https://botland.com.pl [Accessed: 15.06.2022].
[7] "Gyroscope and accelerometer MPU – 6050" [Online]. Available: https://abc-rc.pl/mpu-6050 [Accessed: 15.06.2022].
[8] "DC motors with encoder" [Online]. Available: https://www.fruugo.pl [Accessed: 15.06.2022].
[9] "Bluetooth HC – 05" [Online]. Available: https://botland.com.pl [Accessed: 15.06.2022].