Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 3467–3476

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Autonomous control of a mobile robot to explore areas with difficult terrain

Sebastian Pecolt[a]*, Andrzej Błażejewski[a], Tomasz Królikowski[a], Piotr Zmuda Trzebiatowski[a], Błażej Schulz[a]

[a]*Faculty of Mechanical Engineering, Koszalin University of Technology, Koszalin 75- 453, Poland*

Abstract

The article presents a design solution of a mobile robot dedicated to autonomous movement in hard-to-reach places. The 3D printing technology made it possible to make a mechanical structure adapted to the actuators and electronic elements with which the robot was equipped. To control the mobile robot, a Raspberry Pi 2 microcomputer and an Atmega328P microcontroller were used, which allows the robot to be controlled via WiFi or a game controller for PlayStation3 using Bluetooth communication. In addition to remote control, the rover has an implemented algorithm of autonomous operation with simultaneous image transmission.

Keywords: mobile robot, autonomous control, Raspberry Pi, video camera, Bluetooth.

1. Introduction

Mobile robots are becoming increasingly popular and necessary in everyday life. At the beginning many of them were controlled by wire connection which was easy to create but wasn't convenient to work with. Mobile robot had limited area of motion.

Fast growth of technology gave solution for better communication with robots. Popularity of microcomputer had influence on creation new generation of mobile robots. Currently construction of simple electronic system based on

* Corresponding author. Tel.: +48 606 304 679
E-mail address: sebastian.pecolt@tu.koszalin.pl

10.1016/j.procs.2021.09.120

microcontrollers is very easy. Everyone can buy few electronic elements and microcontroller and build at home different systems that previously were reserved only for company.

One of the most popular microcomputers is Raspberry Pi which is small computer with lots of features for example communication by WiFi or live stream videos. Recently popularity of Raspberry Pi is growing especially in engineering community [1].

In addition to electronic components, crucial for the project was to prepare working solutions for the suspension and drive system. The popularization of affordable 3D printers allowed the use of additive manufacturing methods. This significantly reduced the time needed to work on subsequent iterations of the prototype. The work presents the advanced computer design and production technology applied to create these systems.

2. Main platform construction

Mobile robot (Fig. 1.) was created as project to explore areas with difficult terrain and places without lights. The design of robot is based on Curiosity Rover, which exploring the Gale crater on Mars as part of NASA's Mars Science Laboratory (MSL) mission [2]. It has dedicated construction, designed to withstand specific conditions, which were:

- rocker – bogie suspension,
- differential drive mechanism,
- six wheels with individual wheel drive based on servo motors,
- controlled by microcomputer – Raspberry Pi 2 and microcontroller Atmega,
- movement manipulate by WiFi or autonomous control,
- streaming video by WiFi,
- Night Vision Camera with IR LED illuminators,
- headlights,
- independent power supply.

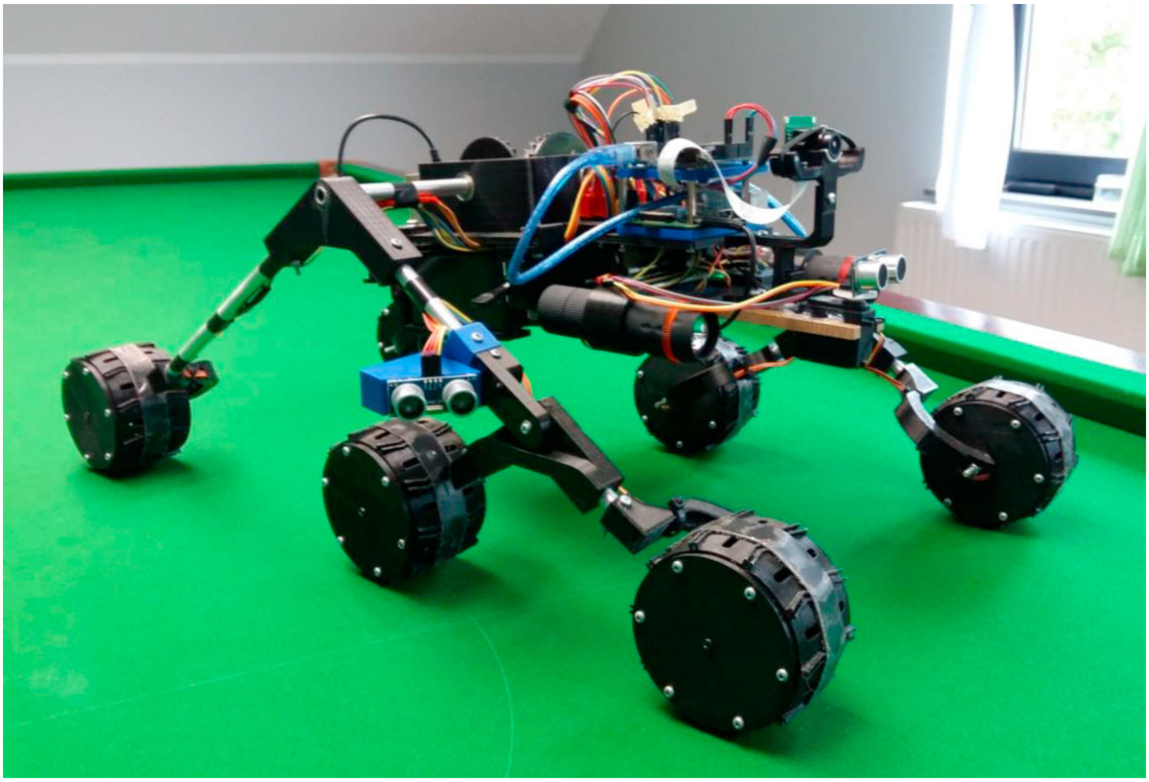

Figure 1. Mobile robot-general view

I Design and manufacturing – additive technology

To find the best solution for mobile robot construction, all parts of the prototype chassis were designed in SolidWorks environment (Fig. 2.) The planned operating conditions for the prototype did not include operating at high temperatures and high loads. Therefore, it was decided to use Polylactic acid (PLA) a thermoplastic polyester with a melting point from 150 to 160 ° C and glass transition temperature between 50 and 80 ° C. It is the most common material used in home FDM technology 3D printers. At temperatures up to 50 ° C, it is a very rigid material with low shrinkage and tight dimensional tolerances. It made it possible to quickly prepare durable elements with high precision, ideal for the needs of autonomous robot prototype, with 3D technology (Fig. 3.).
Using of additive technology allowed to avoid the high costs associated with the preparation of metal or ceramic parts with subtractive manufacturing. By using elements available in hardware stores (tubes, screws, etc.) and components printed on site, it became possible to efficiently and cost-effectively prepare the frame and mechanical construction of the prototype and its subsequent versions [3,4]. This technology is widely and safely applied in different branches [5-7].

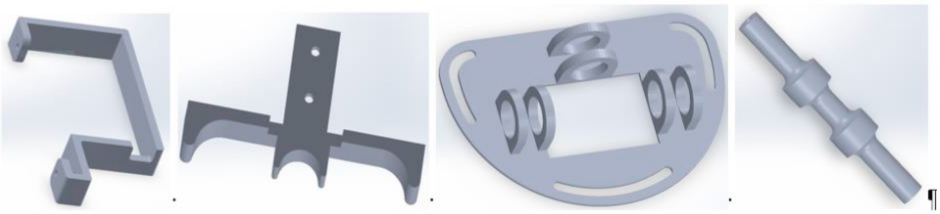

Figure 2 Designed part of robot - Solid Works application

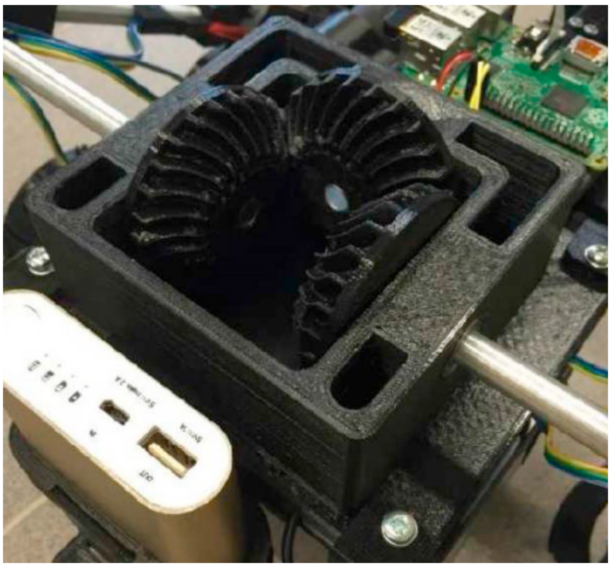

Figure 3 Differential gear printed - 3D printer application

II Functional subsystems and control

Apart of mechanical construction of the mobile robot, it is necessary to provide it with the ability to move or transmit video via a WIFI connection. To control a robot a Raspberry Pi 2 Model B microcomputer was used ,it is the second generation of Raspberry Pi [8]. The microcomputer has 900MHz quad- core AMR Cortex A7 processor, 1 GB

RAM memory, 4 USB ports, 40 GPIO pins, Full HDMI and Ethernet port, Camera (CSI) and Display (DSI) interface, combined 3,5mm audio jack and compose video, micro SD card slot and VideoCore IV 3D graphics core [9]. The Raspberry Pi's operating system is Raspbian, and is based on Debian GNU/Linux distribution. By programing GPIO pins on the Raspberry Pi there is possibility to control external systems connected to microcomputer.

Each of the robot's wheels has its own motor. AR – 3606HB are R/C servo motors with the ability for continuous rotation. This solution allow an easy way to get independent wheel movement. Maximum rotation speed of the servo is 71 RPM and it can produce up to 6.7 kg*cm of torque [10]. The vision system of the mobile robot is based on OV5647 sensor. The sensor itself has a size of ¼ inch, native resolution of 5 megapixels and has a fixed focus lens onboard. Camera supports HD 1080p night vision and it is directly connected to Raspberry Pi 2 via the CSI bus.

Figure 4a shows electrical connections between Raspberry Pi 2 and Arduino UNO with external devices. The servo motors that drive the vehicle where grouped by three for each side of the mobile robot and connected directly to output ports on the Raspberry Pi. This is the simplest way to achieve drive each track with a separate power source To deliver specific voltage for headlights fixed-voltage regulator where used with Mosfet transistors controlled by Raspberry Pi 2. The Fig. 4b shows how the ultrasonic sensors and actuator were connected to Arduino UNO. The logic voltage converter allows you to connect the Arduino UNO with the Raspberry Pi via I2C communication. It is necessary because both systems operate with different output voltages and the inputs of one of them could be damaged during data transfer between them.

a) b)

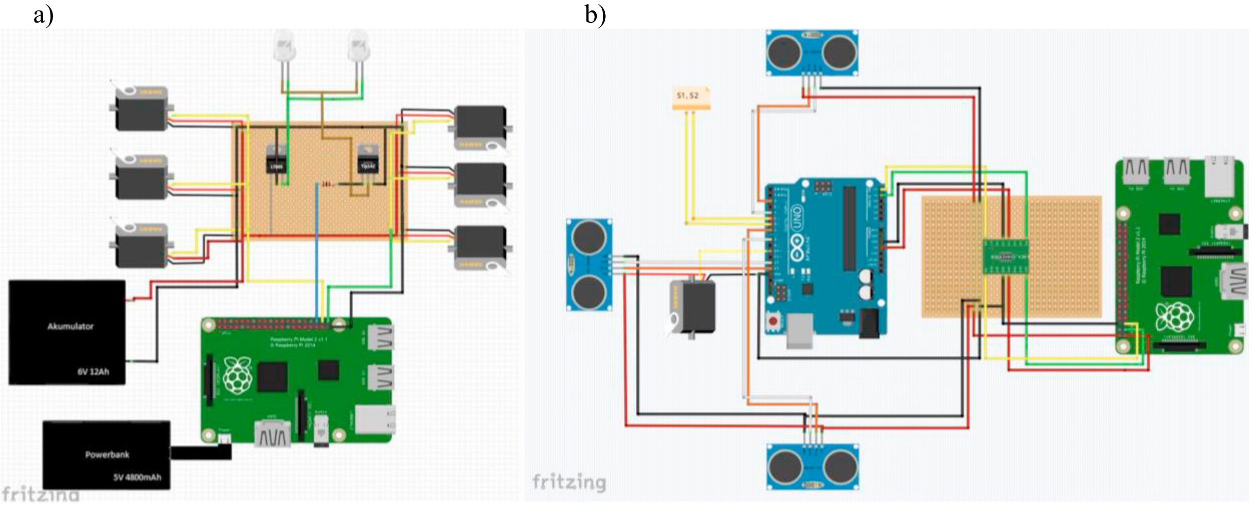

Figure 4 Main electrical circuits of the mobile robot

II Subsystems connections

Mobile robot can be split by four systems (Fig. 5.):

- Power supply system (power bank 5V for Raspberry Pi 2 and VRLA battery 6V for DC motor);
- vision system (camera HD),
- drive system (DC servo motor),
- control system (Raspberry Pi 2 and Atmega32 microcontroller).

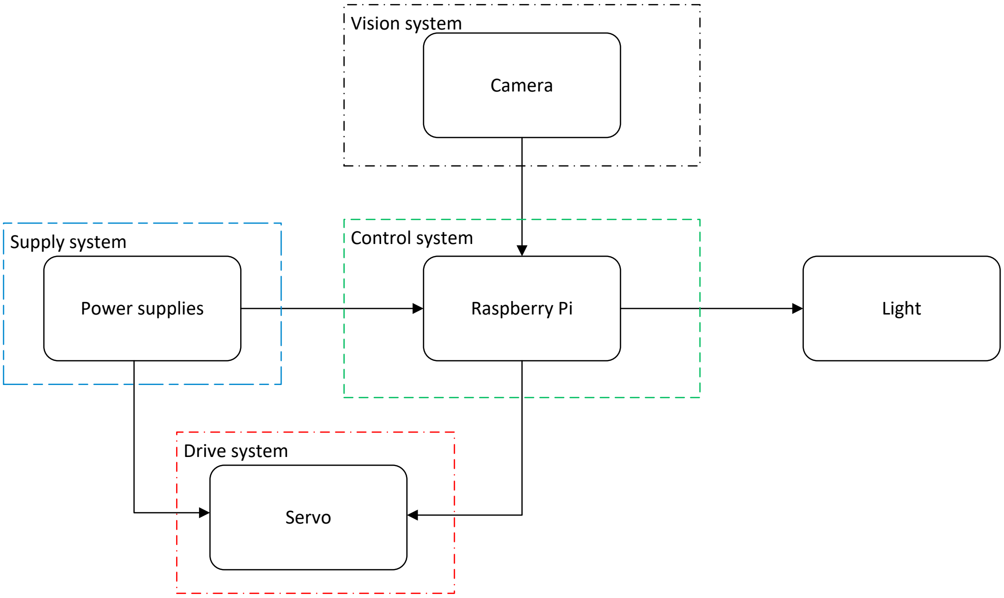

Figure 5 Existing subsystems connections

Servomotors and lights are connected to Raspberry Pi's outputs . The system is powered by 2 independent power sources. Standard Power bank provides 5V power to the Raspberry Pi and microcontroller. While the larger 6V VRLA battery supplies power for the six servomotors. The autonomous control algorithm uses distance sensors (ultrasonic sensors), and the microcontroller acts on their basis.. The task of the algorithm is to control the robot so that it moves in a closed room with the and avoid any obstacles. The algorithm assumes four states that the robot distinguishes while driving: fast driving state, slow driving state, scanning state and making a decision after scanning. In the fast driving mode, the rover's speed is 30% of the maximum power of the drive motors, the distance read by the front ultrasonic sensor is more than 75 cm, and left or right turns are made when one of the side ultrasonic sensors detects an obstacle within the distance of 30 ÷ 40 cm. The fast driving state ends when the distance measured by the front sensor is in the range of 45 ÷ 75 cm. The robot moves at a speed of 10% of the maximum power with the simultaneous measurement of the distance by the side sensors in the range of 30 ÷ 60 cm, detection of an obstacle by the front sensor below 45 cm or by the side sensors, results in the robot entering the scanning state. In this mode, only the front ultrasonic sensor is measured and the servo is turned on, which moves in the angular position 0 ÷ 180 ° to one side and the other in 5 ° increments (which gives a total of 36 measurements). These measurements are then divided every 12 into 3 separate sectors: the left sector, the front sector and the right sector (Fig. 6).

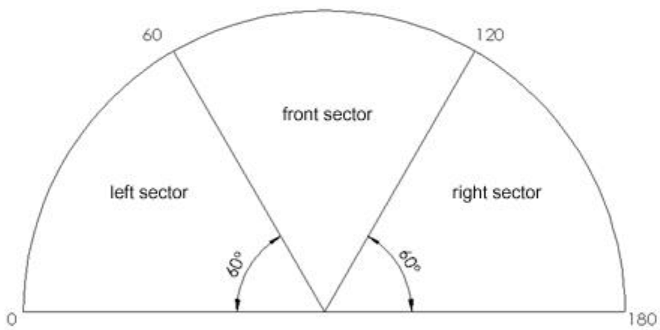

Figure 6 Division of the area in front of the rover into sectors

Measurements over 75 cm are considered successful, measurements below 50 cm are counted by the variable determining the length of obstacles. The comparison of the obtained results allows to determine the direction in which it will move next, i.e. turns left, right, forward or backward with a 180 ° rotation of the robot. It is possible to extend the turn of the robot to 90 °, when the collected lengths below 50 cm give a total result of more than 20, otherwise the robot rotates by 60 °. The diagram of the algorithm is presented in Fig. 7.

The algorithm uses abbreviations referring to the variables used in the program:

- dist1 - distance measured by the front sensor,
- dist2 - distance measured by the left sensor,
- dist3 - distance measured by the right sensor,
- s1, s2, s3 - variables defining the left sector, the front sector and the right sector, respectively,
- d - variable specifying the length of the tested obstacles,
- vel - variable defining the speed of the robot.

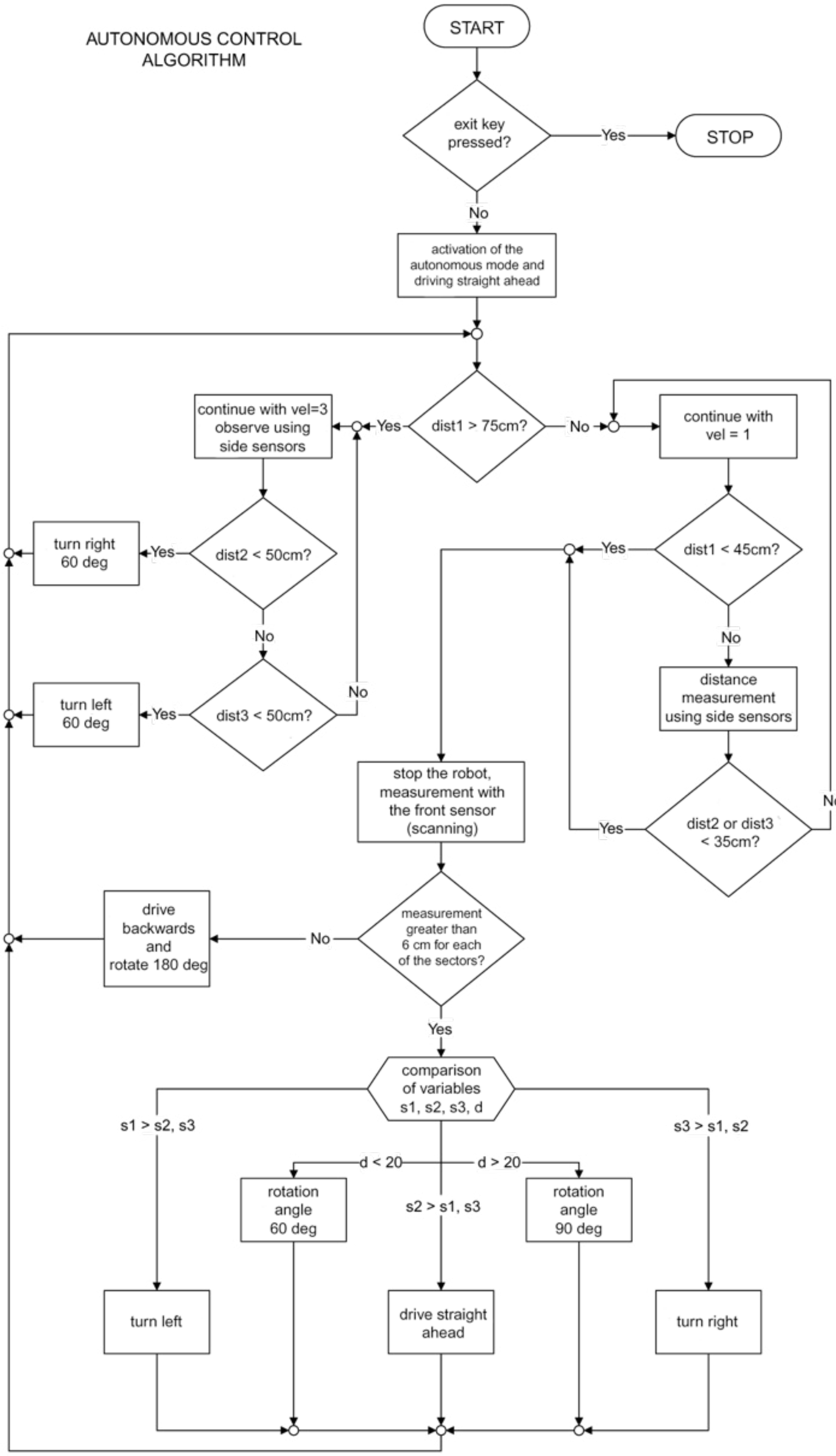

Figure 7 Autonomous control algorithm diagram

3. Controlling and video streaming

To control the robot, programs in the Python programming languages supported by Raspberry Pi and C / C ++ supported by Arduino UNO have been written. The programs are divided in such a way that separate programs on the Raspberry Pi are used for image processing and for autonomous control [11-14].

The autonomous control program was written based on the Pygame library, which allows you to switch between control modes with its functions. On the side of the ATMega328 microcontroller, the control program uses the Wire and Servo libraries, whose task is to communicate with other devices by reading or writing data and the ability to control the servers using the built-in function libraries [15-20].

In autonomous mode, the model recognizes the following ways to control it: driving straight, fast and slow, scanning and making a decision after scanning. The numerical value stored in the flag variable determines which control mode is active at the moment.

In the straight-ahead driving mode, the measurement from three distance sensors is taken into account, the values of which are stored in the variables: *odleglosc, odleglosc2, odleglosc3*, by calling the functions responsible for measuring each sensor separately. The algorithm determines the rover's movement.

The transition to the scanning mode and the way the robot moves are described by the control algorithm. During the scan, *servo-prawo* function is called followed by *servo-lewo*. Part of the program is described below.

```
int servo_prawo(){
for(angle = angle; angle <=180; angle+=5)
{
    myservo.write(angle);
    delay(100);
    odleglosc = zmierzOdleglosc1();
    delay(10);
if(angle ==0){
      impulsy_prawo =0;
}
if(state == false){
break;
}
if(odleglosc >=0&& odleglosc <=50&& zliczanie1 ==true){
      licz +=1;
      impulsy_prawo +=1;
if(licz ==2){
        poczatek = angle;
        zliczanie1 = false;
}
}
else{
      licz =0;
}

if(odleglosc >=80){
if(angle >=0&& angle <=60){
//sektor 1
        licz_sektor1 +=1;
}
if(angle >60&& angle <=120){
//sektor 2
        licz_sektor2 +=1;
}
if(angle >120&& angle <=180){
```

```
//sektor 3
        licz_sektor3 +=1;
}
}
}
}
```

Data on the collected distances are stored in three variables: *sektor1, sektor2, sektor3,* calculated as the average value of the measurements made to the left and right and representing the ranges of angular positions: 0 ÷ 60, 60 ÷ 120 120 ÷ 180. The length of the robot's rotation time is a variable *srodek*, the value of which is calculated as the difference of the two variables *poczatek* and koniec, which in turn describe the angle at which the obstacle was detected and at what angle the obstacle ended.

The decision on how to control the robot after the scanning process is completed results from comparing the values of the above variables in accordance with the algorithm. The choices are: go straight, left, right, left or right, left or straight, right or straight. The necessary condition for choosing one of the above is the value of one of the sector variables greater than 6, otherwise the value of the flag variable will change (the rover is moved back and its rotation by 180 °). The program code is described below.

```
if((sektor1 >6)||(sektor2 >6)||(sektor3 >6)){
        wykonaj = true;
}
else if((sektor1 <6)&&(sektor2 <6)&&(sektor3 <6)){
        wykonaj = false;
        flag =13;
}
```

While the autonomous control program is running, it is possible to transfer the camera image by streaming to another PC via Wi-Fi. To do this, run the streaming.sh file through the terminal, install and start the VLC program on another computer and open the stream, where then you must enter the IP address of the Raspberry computer connected to the network with the port number added. After executing the above instructions, the program code loops until the control mode is switched or the program is turned off by the user.

Alternatively, you can use a wireless controller to control the movement of a mobile robot, via the Raspberry Pi Bluetooth module.Robot movement is controlled by an analog joystick built into the PS3 controller. In this case the range of the connection is limited by Bluetooth adapter up to 10 meters. During the test, the Wi-Fi range outside reached up to 30 meters. Figure 7b shows the mobile robot in the field.

a) b)

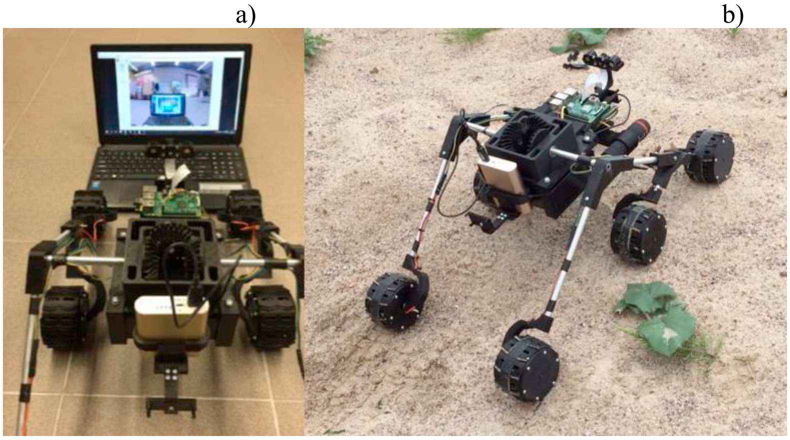

Figure 8 a) Video streaming from mobile robot to computer, b) Mobile robot in action

4. Summary

The Raspberry Pi is extremely popular, portable computer which is used by electronic society to create a various robotic project. Low price, flexibility and technical knowledge are the reason of popularity this electronic board.

Rapid prototyping using 3D printing technology allowed to test the mechanical solutions of the mobile robot in a real environment and allowed for quick modifications In the future some elements which have insufficient strength can be replacement by solid metal parts.

The constructed mobile robot has a remote control functionality with autonomous operation. The rover's control system is based on three component algorithms and four control programs written in Python and C / C ++ supported by control platforms. It was necessary to use I2C communication between platforms to send information about the action taken by the Raspberry Pi computer to a slave device which was the ATMega328P microcontroller. The autonomous program located on the microcontroller is responsible not only for controlling the outputs but is also responsible for collecting information about the distances measured by the sensors. The program responsible for selecting the control on the Raspberry Pi computer was written based on the Pygame library and is its modernization.

References

[1] Ashwin Pajankar, Raspberry Pi Computer Vision Programming -Second Edition: Design and implement computer vision applications with Raspberry Pi, OpenCV, and Python 3, ISBN-10 : 1800207212.

[2] Richard Soare at al. (2021) Mars Geological Enigmas 1st Edition. Elsevier ISBN: 9780128202456

[3] Tomasz Krolikowski at al. (2021) Industry 4.0—Supporting Industry in Design Solutions—All-in-One Computer Cover. Sustainable Design and Manufacturing 2020. Smart Innovation, Systems and Technologies, vol 200. Springer.

[4] T. Krolikowski, at al. Computer modeling and testing of structural metamaterials, Procedia Computer Science, Volume 159, 2019, Pages 2543-2550.

[5] Tomasz Królikowski at. al. Optimization of multilayer rail substrate under moving load, using metamaterials., Procedia Computer Science,Volume 176, 2020, Pages 3399-3406.

[6] Remigiusz Knitter. at al. (2021) Test Stand for Metamaterials Dynamic Properties Examination. Sustainable Design and Manufacturing 2020. Smart Innovation, Systems and Technologies, vol 200. Springer.

[7] Tomasz K. at al. (2019) Emission of Particles and VOCs at 3D Printing in Automotive. In: Ball P., Huaccho Huatuco L., Howlett R., Setchi R. (eds) Sustainable Design and Manufacturing 2019. KES-SDM 2019. Smart Innovation, Systems and Technologies, vol 155. Springer.

[8] J. Kula Piotr, Raspberry Pi 2 Server Essentials, EAN: 9781783985692.

[9] Kai Zhou, Youhong Yuan, A Smart Ammunition Library Management System Based on Raspberry Pie, Procedia Computer Science, Volume 166, 2020, Pages 165-169, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.02.041.

[10] Rob Toulson, Tim Wilmshurst (2012) Fast and Effective Embedded Systems Design 1st Edition Newnes ISBN: 9780080977683

[11] Zhang-Yue, Zhou, An Introduction to Python and Computer Programming, Springer-Verlag GmbH, ISBN: 9812876081.

[12] Mark Lutz, Programming Python: Powerful Object-Oriented Programming, 4th Edition, O'Reilly Media, ISBN: 9781449302757.

[13] Programming Arduino Getting Started with Sketches". McGraw-Hill. Nov 8, 2011. Retrieved 03-28 2013.

[14] Jeremy Blum, Exploring Arduino: Tools and Techniques for Engineering Wizardry, 1st Edition, Wiley 2013, ISBN-10 : 1118549368.

[15] McGugan, Will. (2007). Beginning Game Development with Python and Pygame: From Novice to Professional. 10.1007/978-1-4302-0325-4.

[16] Oxer, J. & Blemings, H.. (2009). Practical Arduino: Cool Projects for Open Source Hardware. 10.1007/978-1-4302-2478-5.

[17] Wagner, N., & Strulak-Wójcikiewicz, R. (2020). Concerns about the technology used by collaborative platforms - a challenge for managers. Procedia Computer Science, 176 2536-2545. doi:10.1016/j.procs.2020.09.319

[18] Strulak-Wójcikiewicz, R. at al. (2020). Applying the business model canvas to design the E-platform for sailing tourism. Paper presented at the Procedia Computer Science, , 176 1643-1651. doi:10.1016/j.procs.2020.09.188 Retrieved from www.scopus.com

[19] Latuszyńska, M., & Strulak-Wójcikieiwcz, R. (2014). Computer simulation of transport impact on environment. Transport Problems, 9(1), 37-47.

[20] Sałabun, W. at al. (2020). How to apply fuzzy miso pid in the industry? an empirical study case on simulation of crane relocating containers. Electronics (Switzerland), 9(12), 1-15. doi:10.3390/electronics9122017