Designing a intelligent robot used for explore unknown zones

Paul Ciprian Patic, Lucia Pascale, Gabriela Mantescu and Marin Mainea

***Abstract*—**An intelligent mobile robot is an automatic machine capable of moving freely into the environment, not fixed to a fixed point.

A mobile robot is a very complex system that may be able to perform different tasks in a variety of real-world situations. It is a mix of devices, equipped with sensors and servomotors, controlled by a calculation system. This system operates in a real space, under the action of several physical properties and must organize the movements, so the robot can perform its activity, depending on the initial conditions of the system and the information previously received, related to work environment.

Generally, the problems that appear with mobile robots are: determining the initial position and its orientation in space, plotting an optimal trajectory of motion, and bypassing fixed or mobile obstacles.

***Keywords*—**Arduino Board, Bluetooth module, Intelligent Robot, Mobile Robot, Sensory Systems, Track Chassis.

I. Introduction

CONSIDERING that technology has become more and more a part of human life in the last century, intelligent mobile robots are nowadays highly valued and used in many areas.

Such mobile robots can be used in many areas to replace humans in dangerous situations: Most often used in the military field, they can be sent to conflict areas to gather information, send them to mined land, detect and mark where the explosive devices are, they can be equipped with robotic arms used to defuse bombs, or they can have different equipment in their structure that can help them accomplish certain tasks, thus replacing the man and reducing the risk of loss of life.

They can be used in the chemical industry, such robots can handle toxic substances, dangerous to humans. Can be used in nuclear power plants for the handling of radioactive substances. The use of these mobile robots is primarily aimed at reducing the risk of loss of human life, replacing man in high-risk activities [6].

In general, the system operates in a real space, under the action of several physical properties and must organize the movements, so the robot can perform its activity, depending on the initial conditions of the system and the information previously received, related to work environment. If data from the working environment are more complete, the robot has more success in accomplishing tasks.

Generally, the problems that appear with mobile robots are: determining the initial position and its orientation in space, plotting an optimal trajectory of motion, and bypassing fixed or mobile obstacles.

It is very important for an automated robot to know its spatial positions in order to be able to plan its subsequent movements so that it can perform certain tasks depending on the moment location of the objects in the workspace. The robot's plan of motion is a complex problem, consisting of several problems, similar to each other.

To avoid hitting with fixed or mobile obstacles in the environment in which they operate, several methods are used:

1) Using proximity sensors;
2) Using the information gathered from several types of sensors;
3) Using sensors that determine the distance to the objects encountered in its direction.

Obstacle localization can also be done through direct contact, but this means holding down the robot's speed. As a result of this direct contact, the status of the robot may change as reaction forces occur, making the use of the robot at high speeds dangerous, resulting robot damage and damage to surrounding objects.

Movement of the robot can also be done without knowing its initial position and orientation in space. This operation can be done by using gyroscopes and accelerometers, optical or magnetic signaling and electromagnetic buoys installed in the environment where we use it [6]. Having in view that technology has become a part of human life in the last century, mobile robots are nowadays highly valued and used in many areas.

II. Operating, Controlling and Communicating With INTELLIGENT ROBOT

For each mobile robot we have a drive system with which the robot is set in motion, a control system that controls the robot, and a communication system used to send information to the robot and vice versa. In addition to these, the mobile robots also have a sensory system that helps us gather information from the environment to control in good condition the robots.

Regardless of the type of robot, complex problems arise in achieving its mechanical structure, because we must be very careful about the types of engines we use to act, taking into account its volume and weight, and the cost of production.

Mobile robots can be operated using DC motors, step-wise motors or servomotors, and hydraulic or pneumatic systems can be used to drive robotic arms.

A. Sensory systems at robots

A robot must be able to adapt to environmental conditions to perform different tasks. It has to change its functional characteristics, with the modification of certain external or internal factors in the space where it operates.

Thus, in the composition of the robots there are different special devices for measuring certain parameters. The assembly of these equipment is also referred to as the sensory system and helps the robot to better understand the space in which it evolves and allows it to perform as best as possible, adapting to environmental changes [3], [4].

The complexity of sensory equipment is closely related to the technological functions the robot has to perform. For example, for a robot that is used in painting operations, a correct realization of its motion can be produced by measuring the trajectory with sensors and displacement transducers, but for a robot performing assembling operations, in addition to the motion sensors, they will you also need tactile or optical sensors. In the first case, the sensor system provides the robot with data about its intrinsic parameters, such as speed, acceleration, or displacement, while the second case also requires a definition of the object's workspace characteristics. This gives a very important classification of sensory systems for finding internal or external parameters [3].

Every robot executes movements after a given trajectory. In order to obtain this trajectory it is necessary to know the position at any moment, and to know the speed and acceleration of the mechanical element [4].

Position, speed and acceleration data can be obtained with specific measurement systems, which can also be called position, speed or acceleration transducers [3].

Measurement of the position is achieved by means of transducers that convert the movement of the moving elements into an electrical signal that is compatible with previous numerical processing [4].

In order to drive a robot properly, we need to know the speed of its mobile elements. Speed adjustment is a very important part of a robot's adjustment system.

Speed measurement can be obtained by knowing that this magnitude is a variation in distance, measured at equal time intervals. Thus, speed measurement can be determined from position measurement.

Two methods of speeding are usually used, both starting from the incremental measurement of the position: it is possible to calculate the number of displacement quanta or the number of pulses traveled at a certain time, or to measure the time at which it occurs an impulse [3], [4].

For a robot to be able to carry out the tasks correctly, as it comes into contact with certain external factors, it must be equipped with a special sensor system that can identify certain parameters, such as: the presence of an object on its travel path, map an enclosure or determine the point of contact with certain objects to calculate the force or moment at that point.

Figure 1 shows a GT clock generator that emits certain periodic T-signals. The speed is determined from the number of pulses emitted by the photocell in that range [3], [4].

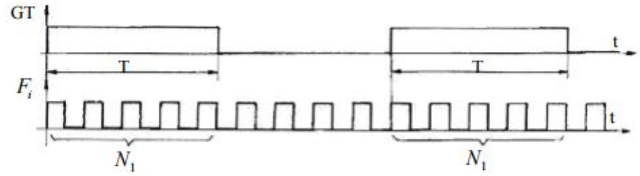

Fig. 1 The GT clock generator

Figure 2 shows the second method. Here we calculate the number of pulses that are generated by the tact generator, over a certain time T, comprised over the duration of the photocell signal F.

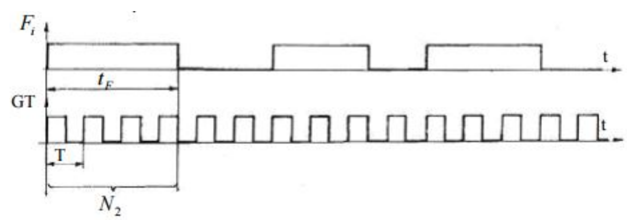

Fig. 2 The second method to calculate the number of pulses

Various sensors can be used to measure external parameters.

The touch sensor. It's a sensor that gives the robot the ability to tap an object. The operating principle of this sensor is to convert the deformation produced on a contact surface into an electrical signal.

Sonar sensors. These sensors are used for obstacle detection or mapping. The sonar operation system is based on the reflection phenomenon of ultra-short waves. Sonar sends a wave, which when it hits an object, is reflected back. The return time of that wave is calculated, and it is possible to determine with the help of some transducers the exact distance of that object. Sonar sensors can be used for mapping.

Optical proximity sensors. They are used to detect an object nearby. They work on the principle of reflection of the bright wave [3], [4].

B. Infrared communication systems

Infrared communications are based on the propagation of light waves emitted in the infrared frequency band.

Infrared communication may be between two mobile devices or, in the case of mobile robots, between a mobile device on the robot and a fixed point, also called an access point or base point, as we present in Fig. 3.

Communication can also be done in other light spectra, not only in the infrared, so these communications can also be called optical communications [5].

This type of communication has a relatively short range of action, but the speed of information transfer can be done at very high speeds.

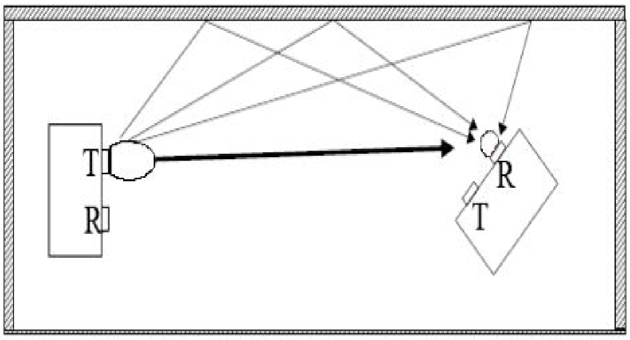

Fig. 3 The infrared communication

C. Bluetooth Communication

Bluetooth communications are mobile-to-mobile communications, and have a range beyond infrared communication.

Bluetooth devices can best be used for applications that want the distance between transmitter and receiver to be small, high data transfer rate and less power consumed.

Each Bluetooth device can connect up to a maximum of 7 devices, but only one connection can be active at a certain time. Each device has a fixed address number, but they do not transmit, unless they have received a master request from the communication.

A device can be both master and slave, and it can change its status depending on the situation.

A pico-net is a network made up of a master and seven slave. In this architecture the master generates the synchronization sequence between devices, and he also initiates the transmission.

A device may be part of several pico-nets. By linking multiple pico-nets, more complex networks has created, that we called scattered networks. Because there are multiple devices connected to multiple pico-nets, communication bridges will be activated between them. A device that is part of a scattered network may be a master in the case of a pico-net of that network, and may be a slave within another pico-net [2].

In Fig. 4 we can see different models of Bluetooth networks: a) a direct network between a master and a slave, b) a pico-net, c) a scattered network in which a device can be both master and slave [5].

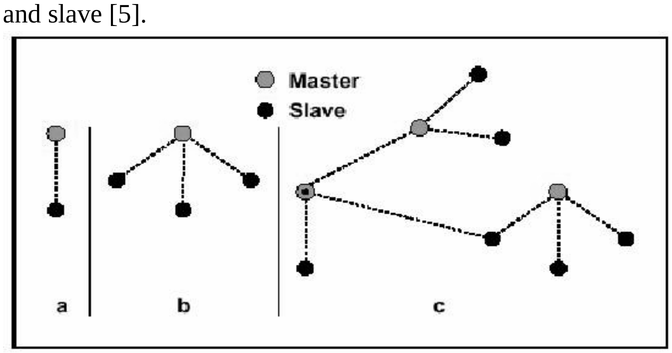

Fig. 4 Bluetooth Networks

III. Designing The Mobile Platform to Explore the Unknown Dangerous Zones

The realization of this robot required two parts, a mechanical part, assembly and assembly of all the components, and a programming part, in which we created the program necessary for controlling this robot and the command interface.

This mobile platform is made up of a track chassis with two robotic arms. They are able to catch and cut. The robot is remotely controlled by a Bluetooth connection and is designed specifically for operating in hazardous areas.

A. Mobile Hardware Executions

To create this robot, the following parts were required: a track chassis, two DC motors and one Arduino due plate, one engine driver, two batteries, two Bluetooth modules and two robotic arms, also known as manipulators [8].

These parts were connected according to block diagram in Fig. 5.

The robot is controlled by an Arduino Due Development Plate. It receives commands from the human operator via a Bluetooth module on a serial port. Arduino plate, interprets these commands, and sends signals to DC motors used for chassis movement via an engine driver or robotic arm actuators to control them. DC motors and servomotors from handlers are separately connected to two different accumulators [10].

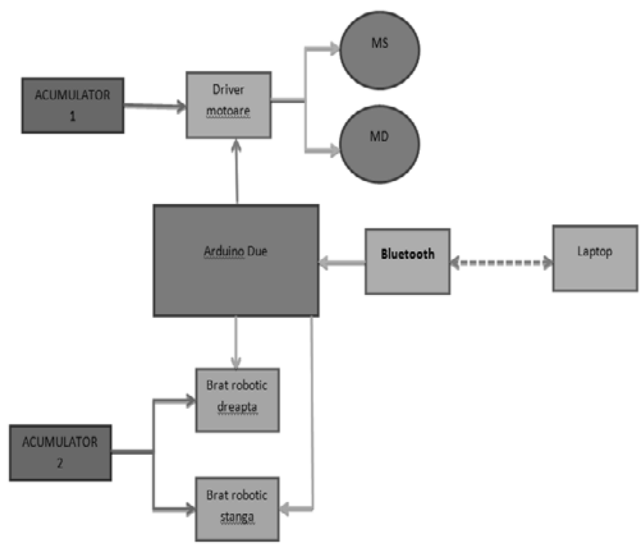

Fig. 5 Robot block diagram

Track chassis

One chose a track chassis because it allows a bigger adhesion on rugged tern, and it can climb up a vertical plane up to 45 degrees. Also, the track-based locomotion system gives a great deal of mobility, because if you spin a track in the same direction and the opposite way, the robot can turn 180 degrees on the spot without having a bigger space maneuver.

The chassis has the following dimensions: 270 mm in length, 220 mm in width and 60 mm in height [7].

The following components are part of this track chassis:
- An engine wheel
- A stretching wheel
- Two carrier wheels
- Track.

These are shown in Fig. 6, from below.

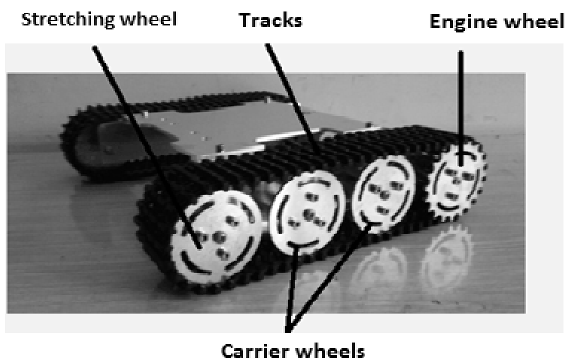

Fig. 6 Track chassis

The drive wheel is a leading toothed wheel that engages an articulated chain from which the track is formed. The tilting wheel is a toothed wheel, which is used to stretch the track, but also serves as a guide. The carrier wheels are very important because they are the support points of the mobile robot. The number of carrier wheels may vary depending on the length of the track, the total weight of the robot and the load it can carry.

The track is made up of several rubber (used only for experimental model) tabs attached to the bolt joints until it reaches the required length. It has on the inside the grooves in which the teeth enter the motor wheel, and on the outside, to have the best adhesion, it has wrinkles in V or X [5].

The drive wheel is driven by an electric DC motor powered by batteries or batteries.

The engines chosen to drive the chassis of the chassis have the operating voltage between 6-12 volts. These engines are powered by a DRV8833 engine driver. This driver has the role of changing the direction of the motors and limiting the supply voltage [7].

The motor shaft is fixed directly to the drive wheel by small screws. The fastening of the chassis motors is shown in figure 7.

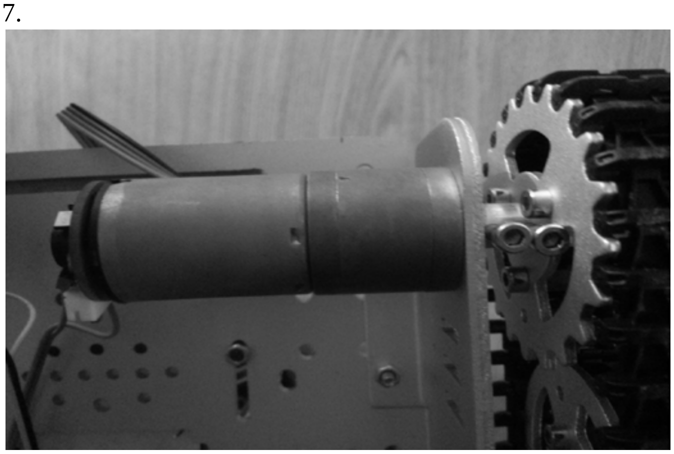

Fig. 7 Gripping the chassis by the DC motor

Engine driver

The DRV8833 engine driver is created by Texas Instruments and can be used for bidirectional control of two DC motors at a voltage between 2.7 and 10.8 V. This driver can also operate with 3 and 5V inputs.

Continuously, it is designed to output 1.2 A currents per channel, but can also produce currents of up to 2 A for a few seconds. Outputs can be connected in parallel to supply 2.4 A of a single motor [7].

The engine driver is used to change the direction of the currents so as to change the direction of movement of the motors. The driver receives current from the battery by connecting to the GND and VIN pins, and receives command from the microcontroller via the AIN1, AIN2 and BIN1, BIN2 pins. It sends the signal to the motors via the AOUT1, AOUT2 and BOUT1, BOUT2 pins. The driver linking to the circuit is made according to the diagram in Fig. 8.

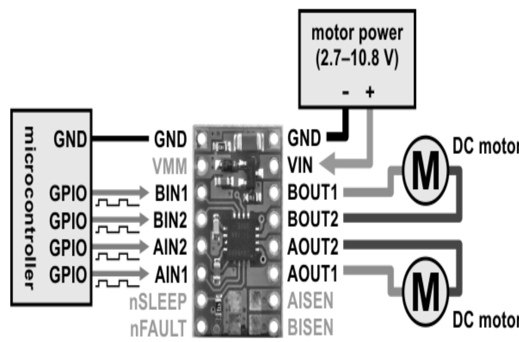

Fig. 8 Schematic of driver linking in circuit

Arduino Due development board

For control, we used an Arduino Due development board (fig. 9) because it has an internal clock frequency of 84 MHz, which gives a very low response time for commands, so it can make commands fast and fine very big. Also, this board has 12 output pins with PWM, which I need for controlling the robotic arms. The disadvantage of this board is that it only works at 3.3 volts.

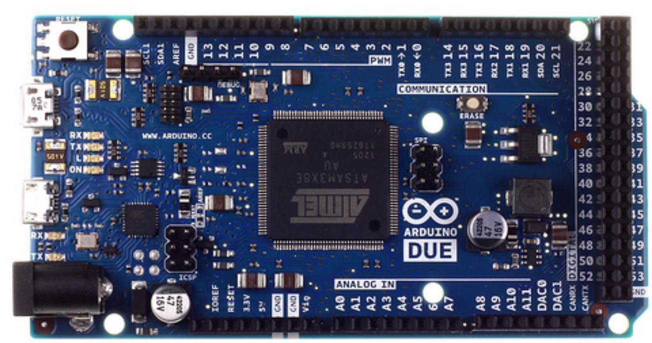

Fig. 9 Arduino Due development board

This Arduino board has an internal security that can protect computer's USB ports from any short-circuit or overcurrent. Although each computer has its own internal security system, this board provides an extra layer of protection. If more than

500 mA is applied to the USB ports, the fuse immediately interrupts your connection until the short circuit is removed.

Arduino Due is a microcontroller based on the Atmel SAM3X8E ARM Cortex-M3 processor. It is the first Arduino board based on a 32-bit ARM processor [10].

The Arduino Due plate has the following features:

- Operating voltage of 3.3 V
- Input voltage 7-12 V
- It has 54 digital input / output pins, of which 12 pins can be used as PWM.
- 12 analog input pins
- 2 analog output pins (DAC) that can convert between digital and analog
- A clock frequency of 84 MHz
- 4 UART transmission and reception ports.

The Arduino Due card can be powered by a USB connector or external power source (accumulator or battery). The external power supplies can be used by an AC adapter - direct current that can be directly plugged into the power jack on the board, or the battery wires can be inserted into the GND and VIN pins on the board [8].

Bluetooth modules

In order to send the remote controls one used two Bluetooth modules. In Figure 10, a Bluetooth module is connected to the Arduino card from the robot and the other module is connected to an Arduino card that is connected to a laptop to transmit the commands on it.

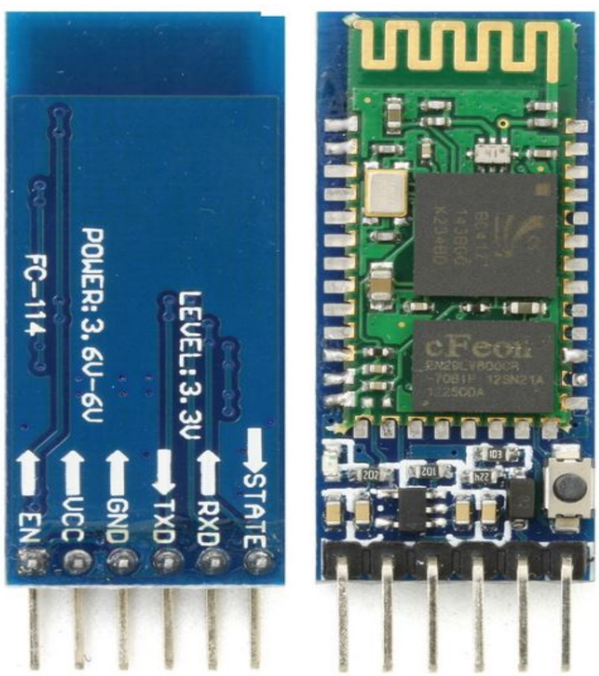

Fig. 10 Bluetooth module

These modules can be master or slave, the laptop module uses it in master mode to send commands to the robot, and the robot is slave.

Robotic arms

On this platform, they use two robotic arms that have the ability to catch or cut objects.

Each of these arms has 6 degrees of freedom and is controlled by 6 MGR 995 servo motors (Fig. 11).

Fig. 11 Servomotor

These servomotors can rotate between 0 and 180 degrees. The robotic arms used by me are built by assembling the 6 actuators with several metal parts of different sizes and shapes.

Fig. 12 Robotic Arm

Accumulators

To feed the robot we used 2 batteries, one to power DC motors used to move the mobile platform, and one to power the robotic arm actuators.

The batteries used are Li-Po type with two cells. They have a voltage of 7.4 V with a capacity of 1300mAh and a discharge rate of 25C.

Wiring diagram

In order to better understand how we connected all the electronic components of the robot, we made an electrical diagram (fig. 13) [1].

On the Arduino board one connected the twelve PWM D2: D13 pins to the twelve servo motors used for robotic arm movement. In pins D14, D15 (Tx3, Rx3) I connected the Bluetooth module to Rx and Tx, respectively. Pin D22, D23, D24, D25 one connected to the DRV8833 driver and use them to control DC motors. From the 3.3 V pin on the Arduino board, we plug the Bluetooth module, and from the GND pin we connected two wires, one for the Bluetooth module and the other for the motor driver [9].

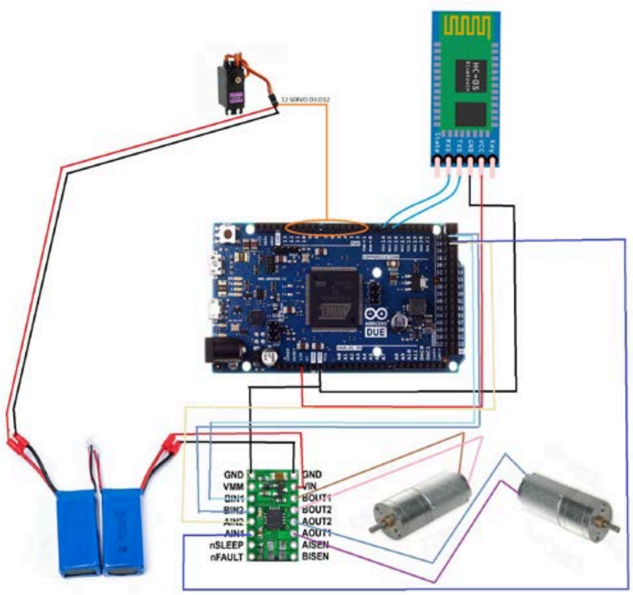

Fig. 13 Electrical scheme

DC motors are connected to the driver in AOUT1, AOUT2, BOUT1, BOUT2.

From a battery, one feed the twelve servomotors from the manipulators through a common wire [7].

The other battery is connected to the engine driver to power the DC motors [5].

B. *Software implementation*

The program for the robot control is made up of two parts: one part in which we made an interface in the programming language C #, with which one read some commands, given by the keyboard, and then we send them to the Arduino plate, and a part where we programmed Arduino board to execute certain commands.

To control the platform to move forward, back, left, right, we used the arrow keys on the keyboard. The program checks if they are pressed, and sends some values to the Arduino plate. In Arduino, those values will be interpreted and commands will be sent to your engines [2].

For example, to move the mobile platform forward, one wrote the following code sequence:

```
private void Form2_KeyDown(object sender,
KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Up)
        {
            arduino.Open();
            arduino.Write("2");
            arduino.Close();
            label1.Text = "Inainte";
        }
    }
    private void Form2_KeyUp(object sender, KeyEventArgs
e)
    {
        if (e.KeyCode == Keys.Up)
        {
            arduino.Open();
            arduino.Write("3");
            arduino.Close();
            label1.Text = "Stop";
        }
}
```

This code sequence shows that when you press the "forward" key, it sends the character "2" to the arduino and will display the text "Forward" on the screen and the robot will move forward due to the commands received from the arduino.

When the key is no longer pressed, the value "3" will be sent to the arduino and "Stop" will be written on the screen and the robot will turn off.

To control the robotic arm actuators we only use KeyDown events. When pressing a certain key, a character will be sent to the arduino. It will be interpreted and arduino will know what servomotor to control.

A portion of this code is shown below:

```
if(e.KeyCode == Keys.W)
{
    arduino.Open();
    arduino.Write("6");
    arduino.Close();
}
```

Writing the program in arduino

In the arduino program we added the servo.h library and declared the servo motors for the two robotic arms.

```
#include<Servo.h>
Servo
ma1,ma2,ma3,ma4,ma5,ma6,mb1,mb2,mb3,mb4,mb5,mb6;
```

In void setup one has declared each PWM pin that the servomotor controls.

```
ma1.attach(2);
ma2.attach(3);
ma3.attach(4);
ma4.attach(5);
ma5.attach(6);
ma6.attach(7);
mb1.attach(8);
mb2.attach(9);
```

```
mb3.attach(10);
mb4.attach(11);
mb5.attach(12);
mb6.attach(13);
```

We have declared the digital pins 22,23,24,25 as outputs for DC motors.

```
pinMode(22,OUTPUT);
pinMode(23,OUTPUT);
pinMode(24,OUTPUT);
pinMode(25,OUTPUT);
```

We have wrote the starting positions of each servomotor.

```
ma1.write(170);
ma2.write(120);
ma3.write(90);
ma4.write(120);
ma5.write(90);
ma6.write(150);

mb1.write(70);
mb2.write(90);
mb3.write(90);
mb4.write(130);
mb5.write(90);
mb6.write(160);
```

We opened the serial port.

```
Serial3.begin(115200);
Serial.begin(115200);
```

In void loop one checked if they get characters on the serial port, and if we get, the plaque executes some commands. For example, to move the robot forward, the following sequence has been written.

```
while(!Serial3.available()){}
int z = Serial3.parseInt();
Serial.println(z);
if(z==2){
digitalWrite(23,HIGH);
digitalWrite(22,LOW);
digitalWrite(25,HIGH);
digitalWrite(24,LOW);
}

if(z==3){
digitalWrite(23,LOW);
digitalWrite(22,LOW);
digitalWrite(25,LOW);
digitalWrite(24,LOW);
}
```

To move the robotic arms one proceed the same. Arduino gets a character and interprets it by knowing what servomotor to act on and in what direction.

```
if(z==6){

  mb1.write(mb1.read()-1);

}
if(z==7){
  mb1.write(mb1.read()+1);
}
```

In the code sequence from above, the value of a servomotor decreases or increases by 1, depending on the characters received on the serial.

IV. Conclusion

The mobile robot that we had presented was designed for use in unknown areas. He is equipped with trackers, which gives him the opportunity to more easily access on rugged terrain. On the mobile platform one installed two robotic arms that have the ability to catch or cut different objects encountered.

This robot can be adapted for use in different situations by adding equipment specializing in the task we want to do. The robot also has drawbacks.

Robot control is remotely powered by a Bluetooth module, so human is not part of risk areas.

The biggest disadvantage is the short range of action due to the use of a Bluetooth module, which is limited to several tens of meters.

For future work we intend to change Bluetooth communication with radio or GPS communication, which would allow to control the robot from far greater distances.

We also want to add a video camera to transmit live video signal or to record and archive images. For better performances we need to add a lot of sensors, to improve its ability to behave as an intelligent mobile robot.

In order to improve the performance of the robotic arms, a multifunctional robotic arm could be used, with the ability to change its drive head and perform multiple operations.

References

[1] Buiu C., Advanced systems for driving autonomous robots, Electra (ICPE) Publishing House, 2003.

[2] Borangiu T., Dumitrache, A., Anton, F. D., Programming robots, A.G.I.R. Publishing House, 2010.

[3] Dumitriu A., Bucsan C., Demian T., Sensor systems for robots, Medro Publishing House, Bucuresti, 1996.

[4] Ignea A., Stoiciu D., Electronic measurements, sensors and transducers, Politehnica Timişoara Publishing House, 2007.

[5] Nedelcu A. M., Mobile Platform for difficult areas, Journal of Electrical Engineering, Electronics, Control and Computer Science JEEECCS, Volume 2, Issue 5, 2016.

[6] Nitulescu M., Robotic systems with navigation capability, Universitaria Craiova Publishing House, 2002.

[7] Patic P. C., Pascale L., Mainea M., Mantescu G., Designing a Mobile Robot used for Access to Dangerous Areas, International Conference on Control, Artificial Intelligence, Robotics and Optimization, ICCAIRO 2017: Prague, Czech Republic, May 20-22, 2017, Proceeding of the Conference.

[8] Zemouri R., Patic P. C., Mobile Robot used to Collect Data from a Difficult Access Area, Journal of New Advances in Mechanisms, Mechanical Transmissions and Robotics, Volume Mechanism and Machine Science 46, 2016, pp. 287 –298, Proceedings of the Joint International Conference of the XII International Conference on Mechanisms and Mechanical Transmissions (MTM) and the XXIII International Conference on Robotics (Robotics'16), Aachen, Germany, October 26-27, 2016.

[9] https://www.pololu.com/product/2130.

[10] https://www.arduino.cc/en/Main/ArduinoBoardDue.

Paul Ciprian Patic is from Valahia University of Targoviste, Electrical Engineering Faculty, Automatics, Informatics and Electrical Engineering Department. The domain of interest are Mechatronics, Industrial Robotics and Educational Robotics, Manufacturing Flexible Systems. He is president of the Romanian Robotics Society, Targoviste Subsidiary, and, also, a member of International Federation of Robotics, or Romanian Automatics and Informatics Society. He wrote many papers presented in different national and international Conferences and published in journals and conferences proceedings.