ANALELE UNIVERSITĂŢII
"EFTIMIE MURGU" REŞIŢA
ANUL XXI, NR. 2, 2014, ISSN 1453 - 7397

Eugen Răduca, Adrian-Florin Păun, Mihaela Răduca, Cornel Haţiegan, Silviu Drăghici, Cristian Rudolf, Lenuţa Candea

Command and Running of an Experimental Model Vehicle for Covering a Labyrinth Type Runway

The paper presents the command and running of an autonomous vehicle - experimental model – for covering a labyrinth type runway. The command is realized with a Micro Arduino type platform and the running with two mini engines of continuous power, each coupled to a tire. The program has been written in C++.

Keywords: command, running, vehicle, experimental model, C++

1. Introduction

The auto field, due to its major involvement in the social and economic life is a dynamic one, continuously developing; the progress is ensured by applying the appropriate technologies, namely the most advanced ones. One constantly targets the increase of the transport system security, in general, and of the participants in traffic, in particular, but also the increase in the comfort level for the passengers. One of the ways for achieving these objectives is the development of autonomous vehicles. In the paper there is presented such an experimental electric model, designed and realized by the authors, regarding the aspects of its command and running.

2. General presentation of the vehicle

The authors set out to design and realize an experimental model of an autonomous vehicle with electric running that can cover a labyrinth type runway.

After studying the theme, the experimental model, presented in figure 1 has been designed and realized.

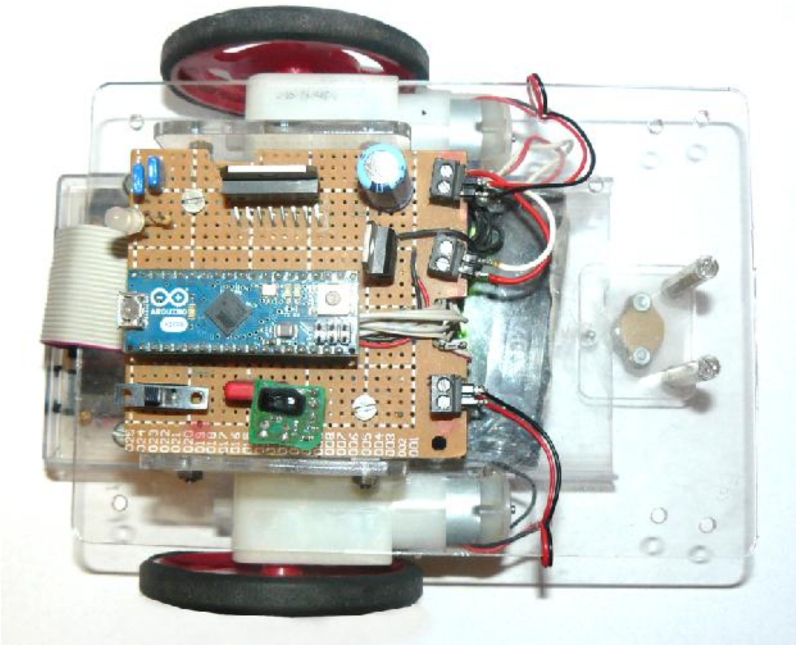

Figure 1. The realized experimental vehicle

The main elements of the realized model are:

1. The plastic frame that is positioned all components: source electrically-Powered, sensors, control equipment and action
2. The unit for command, control and drive, represented by a platform type. This is equipped with Micro Arduino platform [1] with controller Atmega32u4 type [2] (Figure 2).

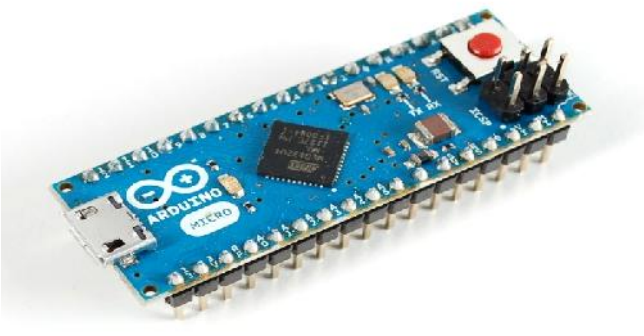

Figure 2. The Micro Arduino Platform

3. Two-wheel drive (Figure 1) each independently driven by direct coupling by one minimotor DC that is embedded and one reducer 1: 120

4. A wheel with two degrees of freedom for ensuring the third point of the tread support, and to carry out a movement lighter

5. A driver unit 5 for driving the motor

6. A power supply system of the whole system consists of a battery pack and a voltage stabilizer

7. A strap IR sensors, each consisting of an eight-coupler-type [3] device through which it is possible to identify the track (Figure 3)

8. Connecting elements: cables, sockets, connectors, double plated cabling

Figure 3. IR sensors ribbon

3. The command principle of the vehicle for covering the runway

The vehicle will go through a labyrinth path where the road surface is disposed by a dark band guiding role (Figure 4) for the vehicle. The trajectory followed by the band may be a rectilinear or curvilinear.

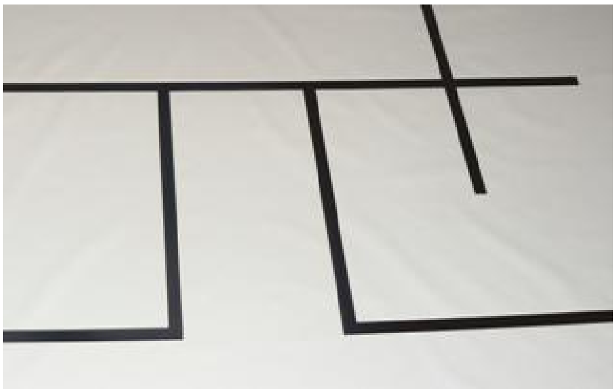

Figure 4. Portion from a labyrinth

The principle on which the coordinating of the vehicle is realized can be observed in figure 5.

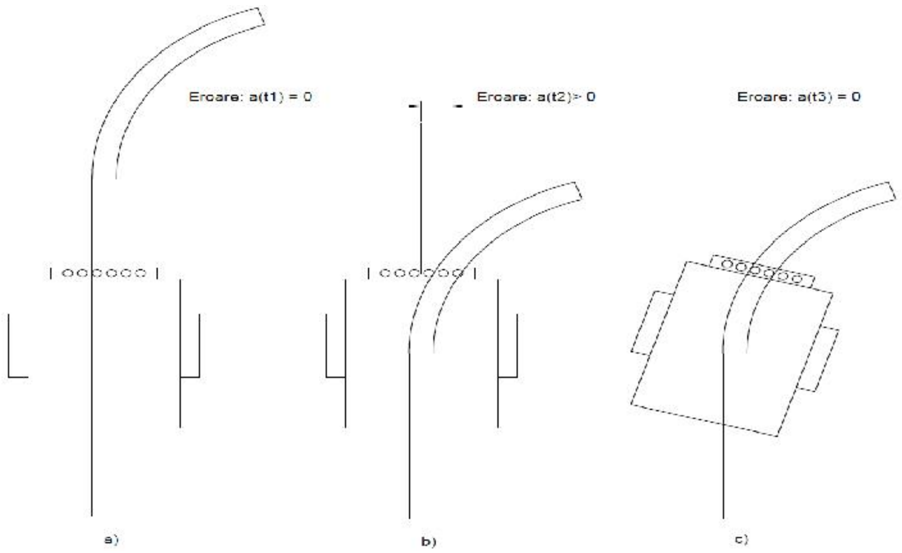

Figure 5. The guiding principle of the autonomous model

It is believed that the vehicle is on path if the two sensors on the central strap infrared sensors are positioned perpendicular to the band on the road (figure 5) (the observation should be noted that the present case are used only 6 of the 8 sensors strap on sensors extremities being inactive).

If at any time at least one of the two sensors (sensor defined in a program written in C ++[4] as S3 and S4) emerge outside the band on the route, whether the current route is straight (figure 5) or curved (figure 5b), information taken from sensors will be transmitted to the micro controller that will send orders by two PWM outputs by the two actuators to drive the wheels of the vehicle for the purposes of repositioning optimal route (figure 5c) (e.g. for a curvilinear trajectory).

Connecting the sensor ribbon [3] to the UC is indicated in figure 6.

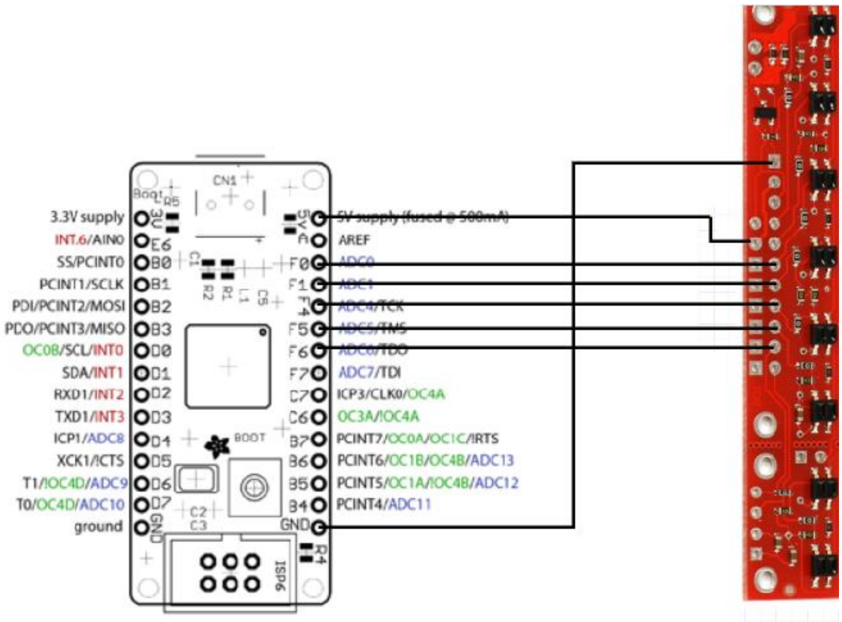

Figure 6. The interface of the QTR8A sensor ribbon with Micro Arduino

Systemically speaking, the whole system behaves like an automatic control system (RAS) discrete, one P-type controller implemented in the controller Atmega course. Actuators are represented by two DC motors. The size of a response is obtained from the optical pickup, the eight coupler type: LED infrared emitter, infrared receiver photodiode, the perceived beam reflected by the surface of the road, and the road lane black light (Figure 7). The deviation can be seen as the difference in position between the tape route and current position where the vehicle is to it at the time.

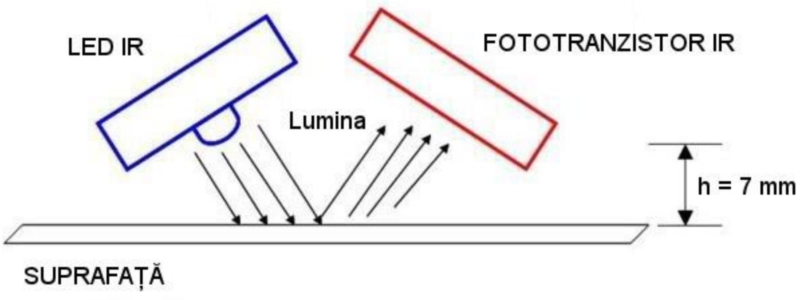

Figure 7. One of the six eight coupling type sensors

The runway resolution - which we define as the maximum space that you route through the vehicle’s top speed between two sets of consecutive PWM control signals [5] - is 0.13 mm, more than enough to not cause any problems (figure 8).

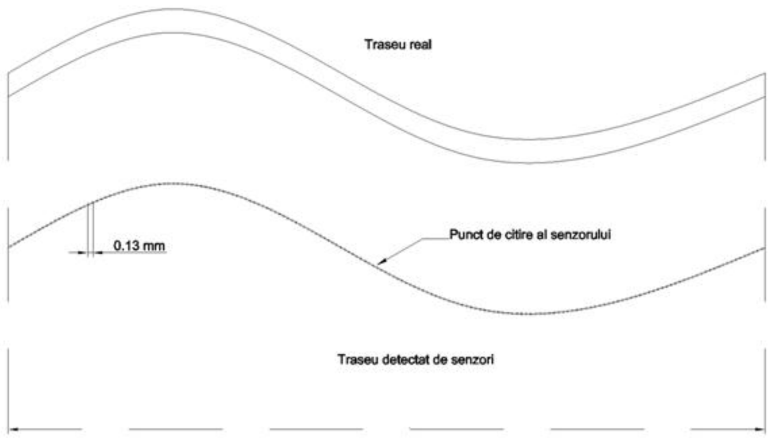

Figure 8. Defining the “runway resolution”

Regarding the control module of the vehicle in terms of solving specific intersections, based on Figure 9 shows that a standard labyrinth path can distinguish four types of intersections.

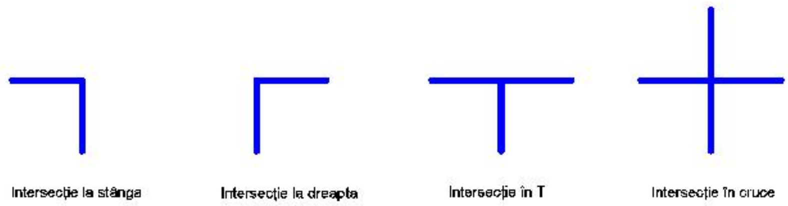

Figure 9. Types of crossings in a labyrinth

The algorithm based on the "left hand rule" provides a solution by setting strict scroll. This algorithm requires by four simple rules, valid for any type of intersection encountered:

- If the vehicle can do just left, it will turn left
- If the vehicle can do and left and right, it will turn left
- If the vehicle can do right or to go forward, it will go forward
- If the vehicle can do strict right, it will turn right

Many subroutines have been implemented in C ++ [4], so that this problem was solved.

As an example is given in the vehicle code sequence found in an intersection "make right" in the end.

```
Void right ()
{
Forward ();
Analog Write (pwm1, 0); // Engine stop
Analog Write (pwm2, 0);
Delay (300); // Delay between readings
Analog Read (A3); // Secondary reading central sensors
Delay (1);
s3= analog Read (A2);
Delay (1);
Analog Read (A3);
Delay (1);
s4=analog Read (A6);
Delay (1);
If (((s3 >=150) || (s4 >=150)) && (covered== false)) // If the runway continues forward
{
Runway += 'S'; // Record the decision
Return;   // Exits the function
}
Righty back (); // Otherwise the vehicle continues the function right ()
```

4. Running the autonomous vehicle- experimental model

A specialized integrated L298N [6] driver was used to drive the two DC motors. The need to supply the engine with a driver occurs due to the current limitations of the microcontroller used. Its output cannot overcome the value 40mA per pin.

The L298N driver circuit is a 2-channel H bridge made completely of 4 transistors and logic gates 8 of "AND" [6]. It is designed to supply such minimotor. It can feed two separate motors individually and can reverse their direction of rotation. The circuit can provide a maximum current of 2 amps on each channel, enough for the proposed application.

The electric wiring diagram for connecting electrical circuit mini engines L298N to drive is shown in Figure 10.

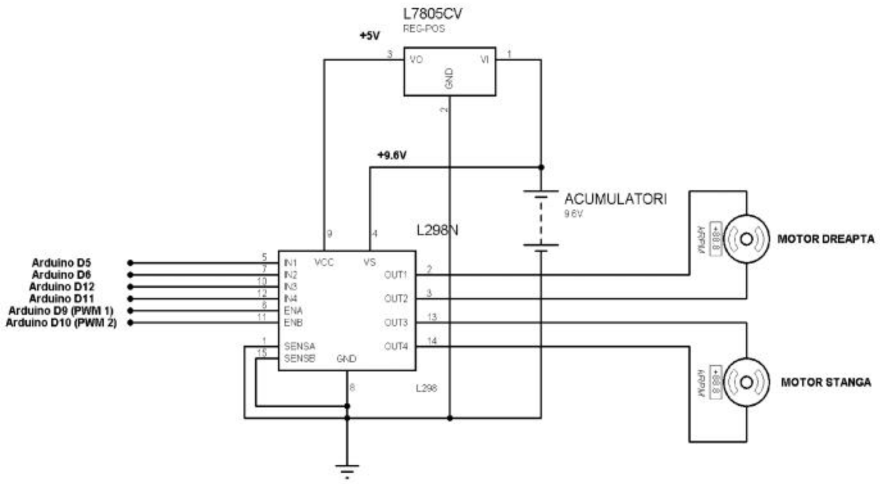

Figure 10. The electric chart for running the mini- engines of continuous power

From the diagram it is noted that driver control inputs IN 1, IN 2, IN 3 IN 4 digital outputs are connected to D5, D6, D12, D11 Micro Arduino development platform; Entries activation ENABLE A driver ENABLE B are connected to digital outputs D9, D10 platform. The L298N connecting the motor to the two groups of outputs OUT 1, OUT 2, a deck or OUT 3 OUT 4 to the bridge B, allows the engine to operate reversible variable speed depending on the combination of signals from the input driver, but also sensitive and responsive SENS A and SENS B signals according to the table. Thus the vehicle can move "forward" or "backward".

Table 1. Settling the runway direction of the vehicle

Bridge A (right engine a)		Bridge B (left engine)		Rotation direction
D5	D6	D12	D11	
1	0	1	0	Forward
0	1	0	1	Backward
0	0	0	0	**Forbidden combination**
1	1	1	1	**Forbidden combination**

The input ENA and ENB activation axle is controlled independently of the speed of rotation of the second motor on the principle of pulse width of the PWM, by varying the ON period and the OFF time to the duty cycle parameter. Thus, the rotational speed of the motor is directly proportional to the average output voltage of OUT.

An example of the motor control signals via PWM current [5] is used in the application shown in Figure 11.

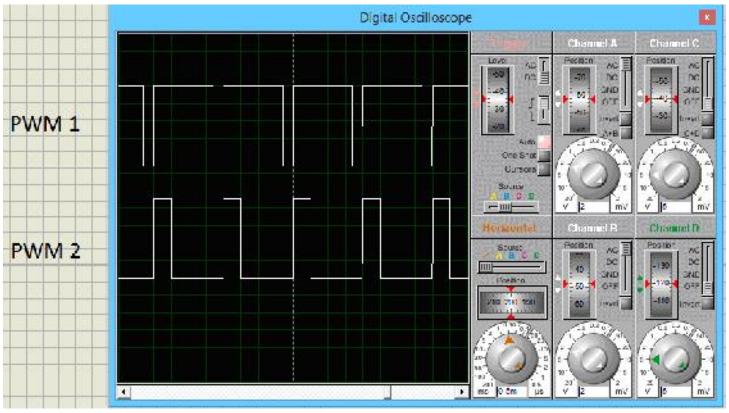

Figure 11. PWM signals with a 500 Hz. frequency

5. Experiments

The practical model was designed and developed extensively tested both components and as a whole. Proper functioning of the experimental testing was performed on the test labyrinth runway overall size of 200 x 140 cm; width of the black band was 1.5 cm (Figure 12).

Because the minimum read/conversion time is 660 us from UC when the run-way resolution is 0,13 mm was established that maximum speed of vehicle is 20 cm/s. At tests was determinate that maximum programming speed of 10 cm/s the autonomous vehicle- experimental model have a inertia moment at stopping of many millimeters, because the vehicle do not have the mechanical brake.

If the model has started from its original position completing the maze, the principle of the scroll presented in Section 3 will be on the blue trail (Figure 13); the black segments have been covered by the vehicle in this test.

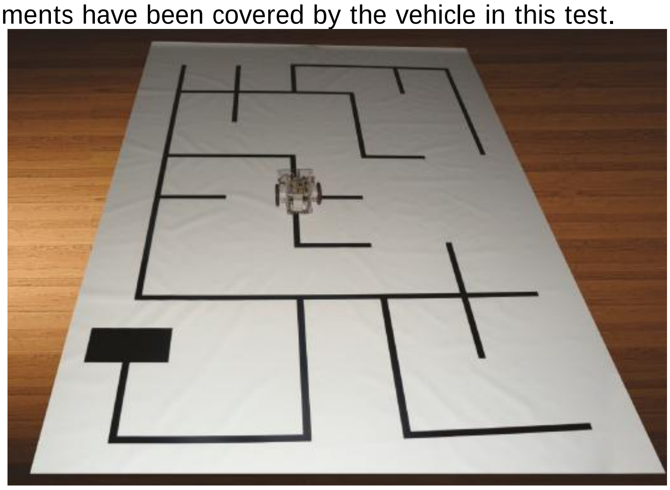
Figure 12. The labyrinth type runway

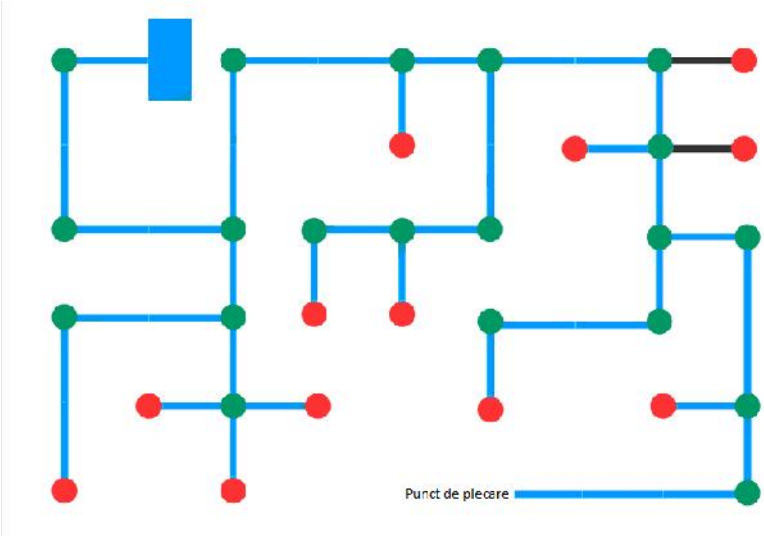

Figure 13. The labyrinth in an initial coverage mode

It was verified that if the vehicle is initially positioned at any point, random route, it will always reach the maze exit following a suitable route according to the general algorithm designed and implemented; red nodes represent points back of the vehicle (changing the direction of rolling, once in such a point)

6. Conclusions

The paper tackles a current theme, namely the one of electric vehicles. Within the paper, concrete and practical solutions, have been presented for realizing a command and running module for an autonomous and functional vehicle. The used technical solutions for the command and running of this auto vehicle miniature model can be totally used for vehicle models on the scale 1:1, that is, the command is naturally realized by a platform with a micro controller programmed in C++, whereas the running is done by continuous power engines. The runway the vehicle covers on a path, when the said item moves between two points can be assimilated to a labyrinth type runway, presented in the paper, and, thus, the principle used for covering a labyrinth type runway in the said paper, can be a starting basis for covering a path between two points, by a real vehicle. The carried out experiments have proven that the used solutions were correct, the presented experimental model being viable and completely functional, according to the set objectives.

References

[1] http://arduino.cc/en/Tutorial/HomePage
[2] http://www.atmel.com/devices/atmega32u4.aspx
[3] http://www.pololu.com/docs/pdf/0J12/QTR-8x.pdf
[4] Jamsa, L. Klander - "Totul despre C si C++". Manualul fundamental de programare in C si C++, Ed.Teora 1999.
[5] http://arduino.cc/en/Tutorial/SecretsOfArduinoPWM
[6] http://ro.farnell.com/jsp/search/productdetail.jsp?sku=403295&CMP=KNC-GRO-FRO-GEN-SKU-MEDI-CAL&mckv=sZtN6KtK4|pcrid|13698417640|kword|l298n|match|p|plid|

Addresses:

- Prof. Dr. Eng. Eugen Raduca, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, e.raduca@uem.ro
- Eng. Adrian-Florin Păun, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, padrian88@gmail.com
- Assoc. Prof. Dr. Eng. Mihaela Raduca, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, m.raduca@uem.ro
- Lect. Dr. Phys. Cornel Hatiegan, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, c.hatiegan@uem.ro
- Eng. Silviu Draghici, enrolled PhD student No.2188/oct.2011 Politehnica Timisoara University, Automatic Systems domain, and Assoc.Eng "Eftimie Murgu" University of Reşiţa, s.draghici@uem.ro
- Lect. Dr. Cristian Rudolf, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, c.rudolf@uem.ro
- Lect. Dr. Eng. Lenuţa Candea, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, l.candea@uem.ro