ORIGINAL PAPER

TOWARDS NEURAL CONTROL OF THE MOBILE ROBOTS

FLORIN DRAGOMIR[1], MIHAI STEFAN CARAMIDA[1], OTILIA ELENA DRAGOMIR[1], EUGENIA MINCA[1]

Manuscript received: 03.04.2019; Accepted paper: 22.05.2019;
Published online: 30.06.2019.

Abstract. *This paper proposes a control strategy, on each direction, based only on brain impulses of a mobile robot with tracks. In this respect it is used a neural headset able to read user's brain impulses. The system is composed of: tanks chassis, 12V DC motor equipped with drivers, Bluetooth module HC-05, microcontroller Arduino mega 2560, RGB LEDs and neural headset Neurosky mind wave mobile. The added value of the paper consists in building a board using Arduino Mega 2560 for controlling the mobile robot. It integrates hardware devices for the command of 12V motor through driver motor, four RGB LEDs indicating the direction enabled. It receives data from Neurosky neural headset, through Bluetooth module.*

Keywords: *Arduino Mega, tank chassis, Bluetooth, brain impulses, wave, Neurosky, mind wave.*

1. INTRODUCTION

The framework of this paper is represented by practical application integrating neural technologies, precisely the neuronal control of mobile robots. This type of technology can be used in fields like medical, military, entertainment, educational and industrial, practically in any field of activity where they can replace the physical movement of a person for a particular robot or device. In the medical field, it can be used for people with physical disabilities and the brain has activity (i.e. a wheelchair controlled with a neural headset), so it would not require physical action but only neuronal control [1]. Another example may be for people without limbs (a manipulating arm, similar to a human arm controlled with a neural headset), so it will replace a human's limb [2, 3]. In the military field, it can be used to control a robot of any kind without physical movement; the work that was built a "mobile robot controlled with a neural headset" can be an example and can be used to research hard-to-reach places and controlled without physical effort [4]. Another example in the military field may be the control of a rocket or missile without physical intervention [5].

Neurosky mind wave mobile neural headset is a device consisting in a headset and in an ear clip. It uses a single AAA battery, with 8 hours of battery life to measure the safe output power spectrum EEG (alpha waves, beta etc.), neuroSky eSense algorithm (attention and meditation) and blinking with the TGAM1 module.

The thechnology integrated in NeuroSky is ThinkGear and eSense. ThinkGear permits the interference with the brain waves due to the touch sensor head (electrode) contact points and reference ear clip. Both, the brain waves and the eSense counter (attention and meditation) are calculated on the ThinkGear chip. eSense is an algorithm used to characterize

[1] Valahia University of Targoviste, Faculty of Electrical Engineering, Electronics and Information Technology, 130004 Targoviste, Romania. E-mail: drg_florin@yahoo.com; mihai.caramida@yahoo.com.

the mental states. The data flow supposes to amplify the brain wave signal and to eliminate the environmental noise and muscle movement. Then, the result is discretised and interpreted with sense counter. The values of the eSense counter do not describe an exact number, but instead describe the domains of activity [6].

The eSense attention meter indicates the intensity of a user's "concentration" or "attention" level. These have to be intense, directed and stable. The rank values varies from 0 to 100. Distractions, stray thoughts, lack of focus, or anxiety can lower the attention level "attention".

The eSense meditation counter indicates the level of mental "calmness" or "relaxation" of a user. The value ranges from 0 to 100. The meditation is assigned with person's mental states, not physical levels. Thus, a simple relaxation of all muscles of the body cannot immediately lead to an increased level of meditation. However, the majority of people, in normal conditions, relax their body and helps mind to relax.

Meditation is related to reduce activity through active mental processes in the brain. For a long time, it has been noticed that eye closure turns into mental activities that process images from the eye. So, closing eyes is often an effective way to increase the level of meditation measurement. Distractions, stray thoughts, anxiety, agitation, and sensory stimulus can lower the levels of the meditation counter [6, 7].

2. SYSTEM’S HARDWARE ARCHITECTURE

The system acts as follows: the human brain produces small impulses called the cerebral waves in order to control the movement direction of the mobile robot. These ones are detect by neural headset placed on the head of the use and forwarded towards HC-05 Bluetooth module (Fig. 1).

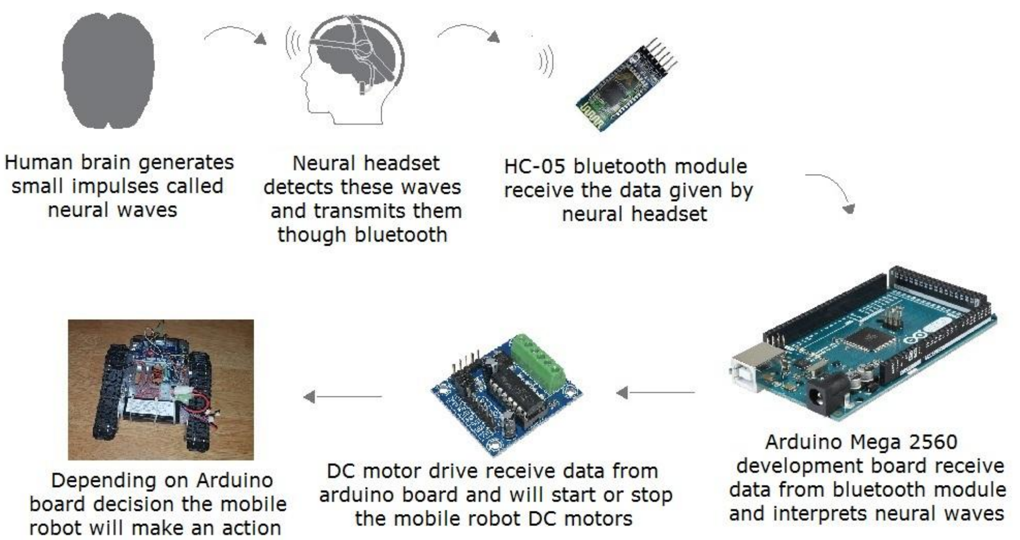

Figure 1. Block diagram.

The Bluetooth device pushes forward neural waves to Arduino Mega 2560 development board to be interpreted. It has a LED indicator that shows whether or not the neural headset is connected (when it is not connected a LED will flash continuously with an interval less than 1 second). The interpretation of the signals is made bit by bit using specific algorithms.

Table 1 provides a general summary of some of the commonly recognized frequencies tend to be generated by different types of brain activity:

Table 1. Brain waves.

Wave type	Frequency range	Mental states and conditions
Delta	0.1Hz to 3Hz	Sleep deep, dreamless, non-REM sleep, unconscious
Theta	4Hz to 7Hz	Intuitive, creative, memories, fantasy, imaginary, dream
Alpha	8Hz to 12Hz	Relaxed (but not sleepy) quiet, conscious
Low Beta	12Hz to 15Hz	Former SMR, relaxed, but focused, integrated
Midrange Beta	16Hz to 20Hz	Thinking, self-aware and surroundings
High Beta	21Hz to 30Hz	Alert, agitation

The electrical scheme of the above system is presented in Fig. 2.

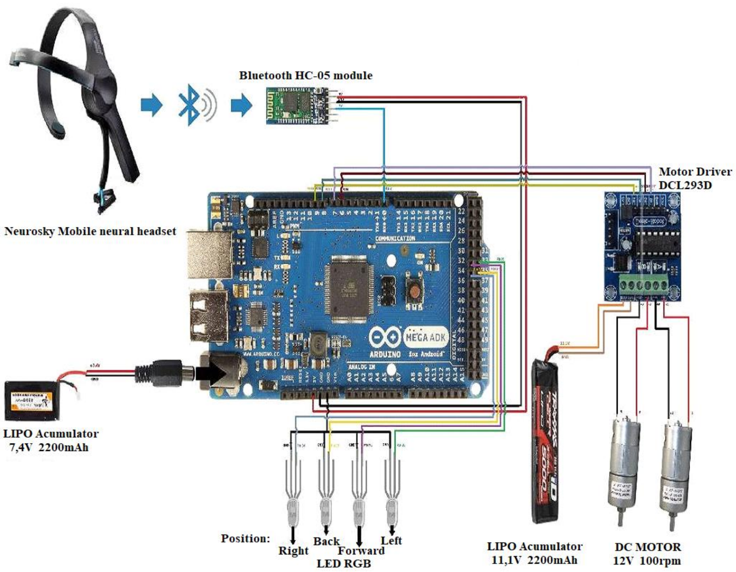

Figure 2. Electrical scheme of the system.

The fourth block is the Arduino mega 2560 microcontroller. It is designed to interpret the information sent by the neural headset via the Bluetooth HC-05 module; this is done by reading each transmitted bit and by specific neural headset algorithms. After the cerebral waves has been interpreted, it will make a decision and give command to the DC motor driver module (block five) and to the direction indicator lights [8-12]. The last block is the mobile robot; it will take action based on what the Arduino mega 2560 microcontroller interpreted through the DC motor driver module.

The hardware structure of the mobile robot is composes of robot chassis build of: an aluminium plate, 2 toothed gears wheels where the 12 V motor axes it's connected, 8 supporting wheels. All 10 wheels are located in the lateral place of the aluminium plate (5 in left side and 5 in right side). Tracks are assembled along the wheels so that the toothed wheel would enter the tracks (Fig. 3).

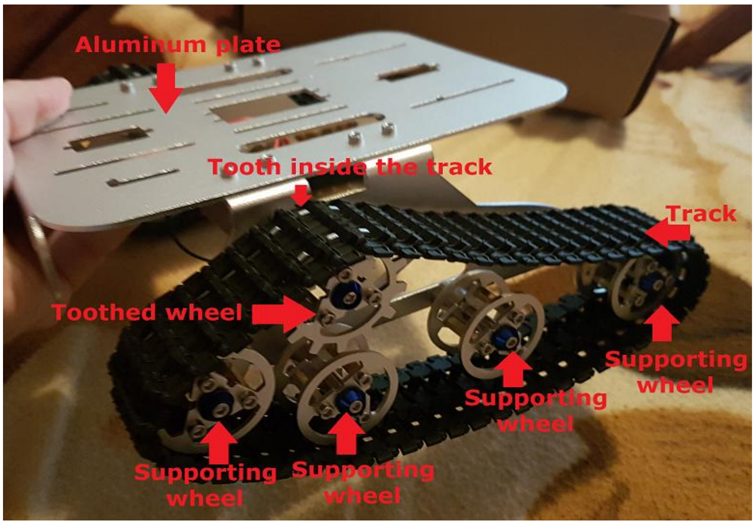

Figure 3. Mechanical subsystems.

The Arduino Mega 2560 controls the robot 12V DC motor through a motor driver and receives data from neural headset through Bluetooth HC-05. This one and 12V DC motors are powered separated by 2 ON/OFF buttons with 2 LiPo battery one of 7.4V for Arduino board and one of 11.1 for 12V DC motors.

The green light of the 4 RGB led was the only one used to indicate the direction to go of the mobile robot.

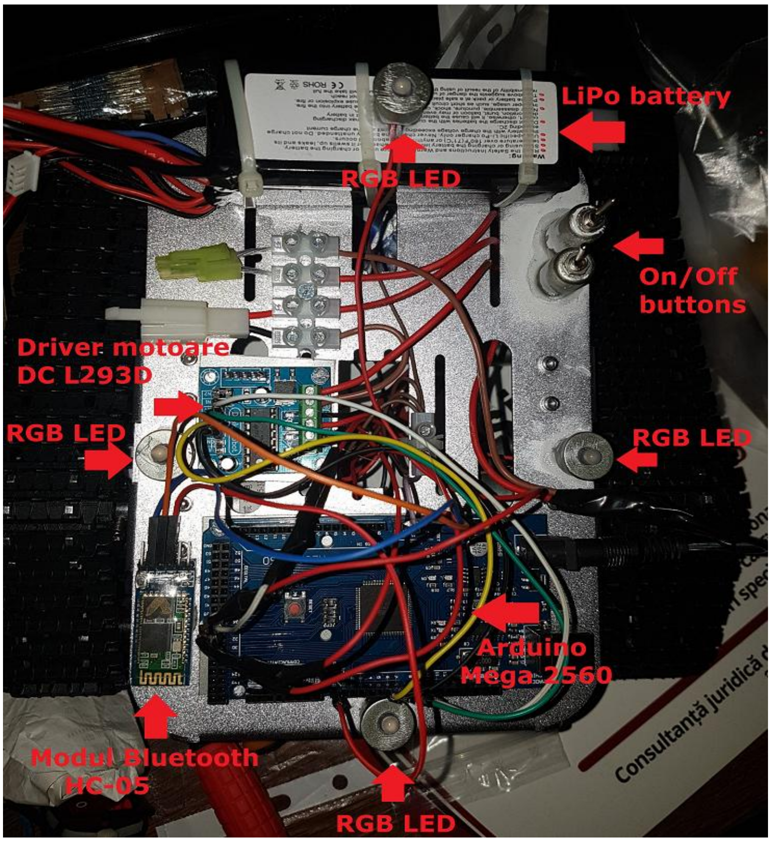

Figure 4. Electric and electronic subsystems.

These subsystems are located on aluminium plate of robot chassis (Fig. 2). The communication between the subsystems id made as follows:

• HC-05 Bluetooth module communicates with Arduino board though TX pin (Bluetooth) and RX pin (Arduino). HC-05 module is power supply it's 5V DC.

• DC robot motors are controlled by Arduino board through the motor drivers (H bridge) through digital pins (6, 7, 8, 9) set as outputs. Power supply is 12V.

• RGB led are controlled by Arduino board through digital pins (36, 37, 38, 39) set as output. Power supply is 5V.

The actions that the mobile robot can do are:

1. Stop - both engines are stopped regardless of direction;
2. Go forward - both engines are turned on in the forward direction and the direction is given by the led in front of the mobile robot (Fig. 5);
3. Go to the right - only the left engine is turned on so the robot will turn right, the direction is given by the right middle light (Fig. 6);
4. To go left - only the right engine is turned on so the robot will turn left, the direction is given by the left middle light (Fig. 7);
5. Go back - both motors are turned on in the reverse direction of point 2 and the direction is given by the LED placed in the back of the mobile robot (Fig. 8).

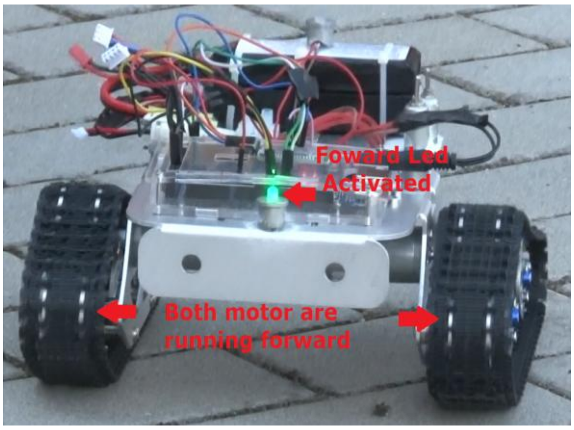

Figure 5. Robot running forward.

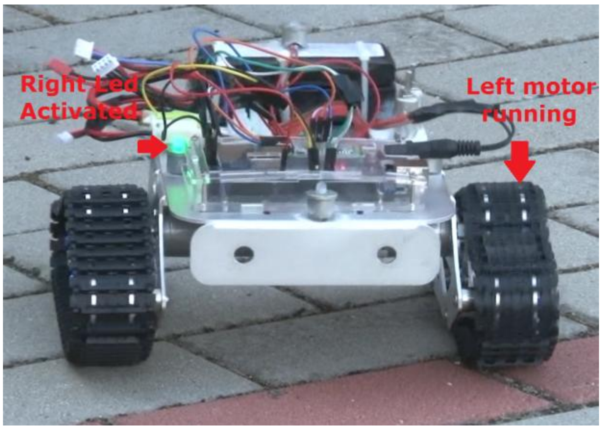

Figure 6. Robot running right.

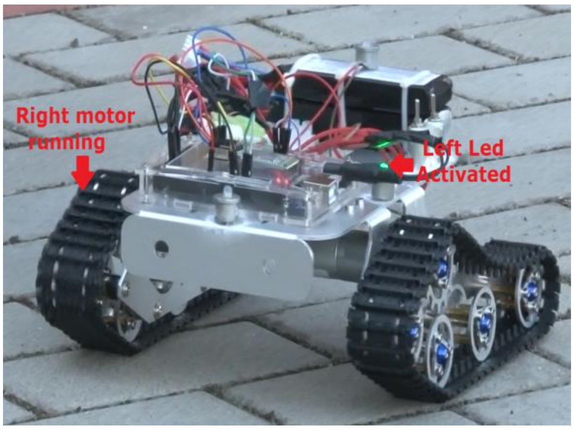

Figure 7. Robot running left.

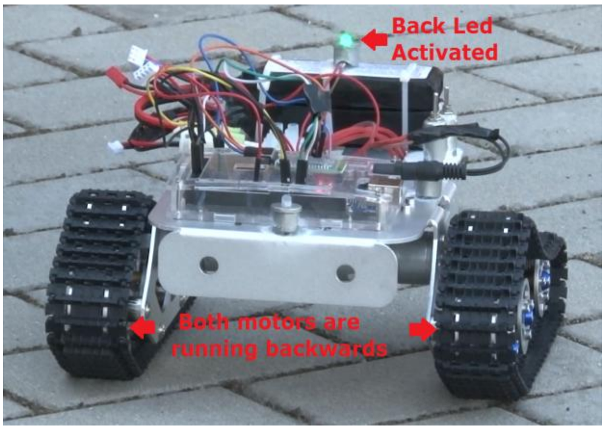

Figure 8. Robot running backwards.

3. PHYSICAL MODELLING

The mobile robot is a two wheeled differential drive robot, where each wheel is driven independently. Forward motion is produced by both wheels driven at the same velocity, turning left is achieved bydriving the right wheel at a higher velocity than the left wheel and turning right is achieved bydriving the left wheel at a higher velocity than the right wheel. This type of mobile robot can turn on the spot by driving one wheel forward and second wheel in the opposite directionat same velocity. Drive wheels are equipped with encoders and their angular velocity readings become available through simple routine calls.

Geometrical dependencies of a differential drive mobile robot are given in Fig. 9, where: v_l and v_r are left and right drive wheel velocities (mm/s), X and Y present mobile robot position in cartesian coordinates (mm), and l is axle length between drivewheels (mm), r is drive wheel radius (mm), R is instantaneous curvature radius of the robot trajectory, relative to the mid-point axis, *ICC* is Instantaneous Center of Curvature

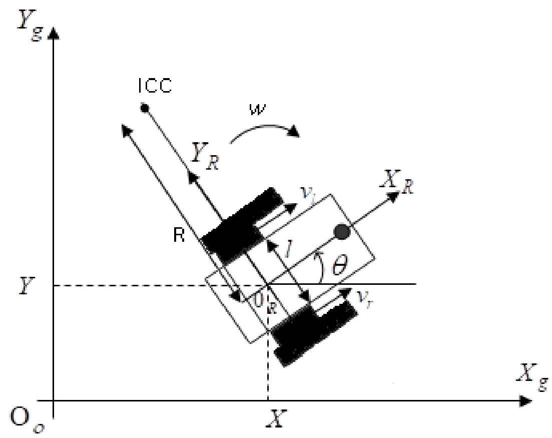

Figure 9. Geometrical dependencies of a differential drive mobile robot.

Kinematic model of a differential drive mobile robot are as follows. With respect to ICC the angular velocity of the robot is given as follows:

$$w(t) = \frac{v_r(t)}{(R + l/2)} \tag{1}$$

$$w(t) = \frac{v_l(t)}{(R - l/2)} \tag{2}$$

where: $v_r(t)$ is linear velocity of right wheel
$v_l(t)$ is linear velocity of left wheel
$\omega_r(t)$ is angular velocity of right wheel
$\omega_l(t)$ is angular velocity of left wheel
$(R-l/2)$ is curvature radius of trajectory described by left wheel
$(R+l/2)$ is curvature radius of trajectory described by right wheel

From (1) and (2) we can write the following equations

$$w(t) = \frac{v_r(t) - v_l(t)}{l} \tag{3}$$

The instantaneous curvature radius of the robot trajectory relative to the mid-point axis is given as:

$$R = \frac{l}{2}\frac{(v_l(t) + v_r(t))}{(v_r(t) - v_l(t))} \tag{4}$$

$$v(t) = w(t)\,R = \frac{1}{2} v_r(t) + v_l(t) \tag{5}$$

Robot’s coordinate and robot’s angle are obtained from the following equations:

$$x(t) = \int v(t)\cos(\theta(t))dt$$

$$y(t) = \int v(t)\sin(\theta(t))dt \tag{6}$$

$$\theta(t) = \int w(t)dt$$

Kinematics model in the global frame is given by:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta(t) \\ v(t)\sin\theta(t) \\ w(t) \end{bmatrix} \tag{7}$$

$$\dot{\xi}_I = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \tag{8}$$

Equation (8) illustrates the relation between the robot’s inputs and the robot’s outputs. That illustrates the robot’s variables and the relation between them, so we could control the robot position by adjusting the two variables that are the average robot speed and the angular velocity.

A particular case in differential robot appears when right wheel linear speed is equal to left wheel linear speed (the robot will move in a straight line). Then: $v_r(t) = v_l(t)$, R = infinity, $w(t) = 0$ and $\theta(t)$ = constant.

Another particular case is when the robot turns around a point in the middle of its axis. Then: $v_r(t) = -v_l(t)$, $R = 0$, $w(t) = (2/l)v_r(t)$.

For two input variables are left motor speed and right motor speed, the kinematics model is:

$$\begin{aligned} \dot{x}(t) &= v_l(t)\cos\theta(t) + v_r(t)\cos\theta(t) \\ \dot{y}(t) &= v_r(t)\sin\theta(t) + v_l(t)\sin\theta(t) \\ \dot{\theta}(t) &= \frac{1}{2l} v_r(t) - \frac{1}{2l} v_l(t) \end{aligned} \tag{9}$$

and we can write the following kinematics model:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta(t) & c\,\mathrm{os}\,\theta(t) \\ \sin\theta(t) & \sin\theta(t) \\ \frac{1}{2l} & -\frac{1}{2l} \end{bmatrix} \begin{bmatrix} v_l(t) \\ v_r(t) \end{bmatrix} \tag{10}$$

Then we could control the robot position by adjusting the two variables that are the right and left wheel linear velocities. The equation 10 illustrates the robot's variables and the relation between them.

4. SYSTEM'S SOFTWARE FRAMEWORK

The mobile robot was programmed using C++ routines and has been done in two steps: firstly was configured the HC-05 Bluetooth module and after Arduino Mega 2560 microcontroller.

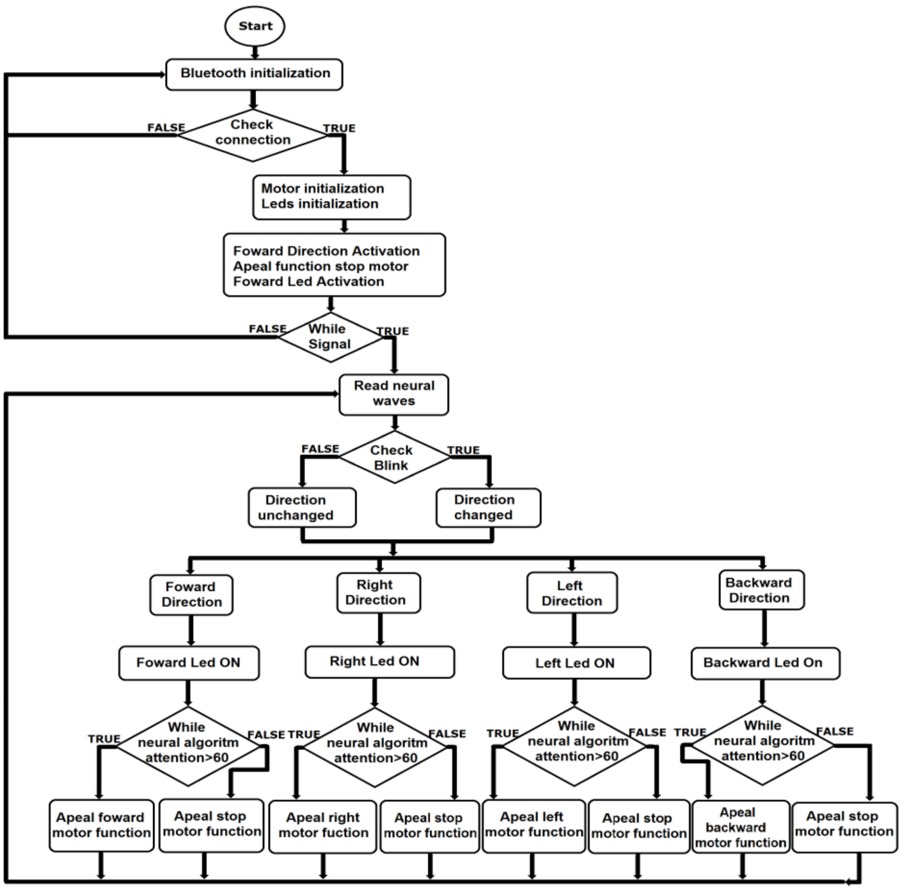

Figure 10. Logical schema of the program used for Arduino Mega 2560 microcontroller programming.

At the beginning of the program, the Bluetooth module HC-05 is initialized via the module's TX pin on the RX-0 pin of the Arduino mega 2560 microcontroller.

After initialization it's verifying if there is a connection with the Bluetooth module, if it's not it will return to the previous block. If there is a connection, pins no 6, 7, 8 and 9 will be initialized for dc 12V motors and pins no 32, 33, 34 and 35 for indicator lights. After initialization of the engines and the LEDs, the default forward function with the stop engine function is called and the front green LED will be ON.

The next step, it will check whether there is a signal between the neural headset and the Bluetooth module HC-05, if there is no signal, it will return to the Bluetooth module initialization block (it will flash continuously within 1 second's time). If there is a signal, the data from the neural headset will be transmitted and read by the Arduino mega 2560 microcontroller. The flowchart from Fig. 10 presents the structure of the program used for Arduino Mega 2560 microcontroller programming.

After reading the neural waves by the microcontroller, it will check if it has blinked. If not blink the direction functions will remain unchanged and direction indicator light will also remain unchanged, and if it blinks and it will change the direction function and the indicator light will also change to that direction selected.

Once a direction has been established, the value of algorithm "attention" will be compared with value 60. If it's below then the both engines will be stopped, in the case if the value is greater than 60 the robot will take action depending on the direction you choose, this can be 4 ways: forward, left, right and back (Fig. 11). After an action is completed, the program will resume reading the neural waves and perform the steps described above

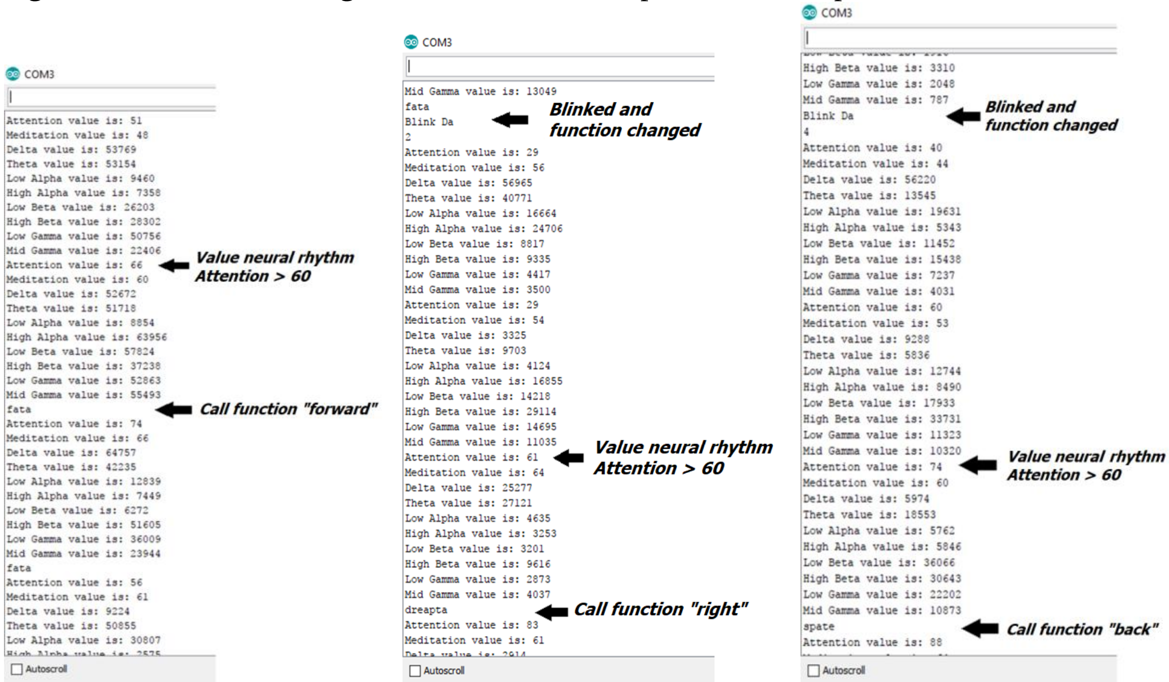

Figure 11. Values displayed on the monitor.

The program sequence for calling function forward (in program is called *fata()*) is:

```
case 1:
 led1();
 if (brain.readAttention() < 60)
 {
   oprit();

 }
 while (brain.readAttention() > 60)
 {
   fata();
   delay(25);
   Serial.print("fata");
   Serial.println();
   brain.update();
```

```
    f_blink();
   }
   break;
```

The program sequence for function right (in program is called *drepta()*), left (in program is called *stanga()*) and back (in program is called *inapoi()*) is similar with forward. The program sequence for function *f_blink()* and function *Eye_Blink()* is:

```
void f_blink()
{ if (ReadOneByte() == 170)
 { if (ReadOneByte() == 170)
  { Plength = ReadOneByte();
   if (Plength == 4)
   { Small_Packet ();
   }
   else if (Plength == 32)
   { Big_Packet ();
   }  }  }

void Eye_Blink ()
{  if (Eye_Enable)
 {   if (On_Flag == 1 && Off_Flag == 0)
  {     if ((Avg_Raw > Theshold_Eyeblink) && (Avg_Raw < 350))
   {      Serial.print("Blink");
    a++;
    Serial.print(a);
   }
   else
   {      if (Avg_Raw > 350)
    {        On_Flag == 0; Off_Flag == 1;
    }    }    }
  else
  {
  }  }
 else
 {   Serial.print("TBLINK");
 }  }
```

The program sequence for *Small_Packet()*) and *Big_Packet()* is:

```
void Small_Packet ()
{  generatedchecksum = 0;
 for (int i = 0; i < Plength; i++)
 {  payloadDataS [i]    = ReadOneByte();
  generatedchecksum  += payloadDataS[i] ;
 }
 generatedchecksum = 255 - generatedchecksum;
 checksum  = ReadOneByte();
 if (checksum == generatedchecksum)
 {   if (j < 512)
  {     Raw_data  = (payloadDataS[2] <<  payloadDataS[3]);
   if (Raw_data & 0xF000)
   {      Raw_data = (((~Raw_data) & 0xFFF) + 1);
   }
   else
   {      Raw_data = (Raw_data & 0xFFF);
   }
   Temp += Raw_data;
   j++;
  }
  else
```

```
  {     Onesec_Rawval_Fun ();
  }   }   }

void Big_Packet()
{  generatedchecksum = 0;
 for (int i = 0; i < Plength; i++)
 {  payloadDataB[i]     = ReadOneByte();
   generatedchecksum  += payloadDataB[i] ;
 }
 generatedchecksum = 255 - generatedchecksum;
 checksum  = ReadOneByte();
 if (checksum == generatedchecksum)
 {   Poorquality = payloadDataB[1];
   if (Poorquality == 0 )
   {     Eye_Enable = 1;
   }
   else
   {     Eye_Enable = 0;
   }   }   }

void Onesec_Rawval_Fun ()
{  Avg_Raw = Temp / 512;
 if (On_Flag == 0 && Off_Flag == 1)
 {    if (n < 3)
   {     Temp_Avg += Avg_Raw;
     n++;
   }
   else
   {     Temp_Avg = Temp_Avg / 3;
     if (Temp_Avg < EEG_AVG)
     {       On_Flag = 1; Off_Flag = 0;
     }
     n = 0; Temp_Avg = 0;
   }   }
 Eye_Blink ();
 j = 0;
 Temp = 0;
}
```

The program sequence for function *Onesec_Rawval_Fun()* is:


```
void Onesec_Rawval_Fun ()
{  Avg_Raw = Temp / 512;
 if (On_Flag == 0 && Off_Flag == 1)
 {    if (n < 3)
   {     Temp_Avg += Avg_Raw;
     n++;
   }
   else
   {     Temp_Avg = Temp_Avg / 3;
     if (Temp_Avg < EEG_AVG)
     {       On_Flag = 1; Off_Flag = 0;
     }
     n = 0; Temp_Avg = 0;
   }   }
 Eye_Blink ();
 j = 0;
 Temp = 0;
}
```

5. CONCLUSIONS

The added value of the paper consists in building a board using Arduino Mega 2560 for controlling the mobile robot. It integrates hardware devices for the command of 12V motor through driver motor, four RGB LEDs indicating the direction enabled. It receives data from Neurosky neural headset, through Bluetooth module.

The work can be improved in many ways: from software point of view, the algorithm supporting the control strategy can be adjusted in order to compensate the reading time of neuronal waves. In present the communication/ reception of neural waves have some interruptions. On the other hand, elements of artificial intelligence can be used for a better recognition of neural wave's classification.

From hardware point of view, the research directions can be oriented towards the the improvement of data transmission and the replacement of HC-05 Bluetooth module with LTE, Wi-Fi or radio technology in order to obtain a greater distance of communication.

Acknowledgements: *This work was supported by the Romanian Executive Unit of Funding Higher Education, Research, Development and Innovation (UEFISCDI), project number: PN-III-P1-1.2-PCCDI-2017-0290, project title: Intelligent and distributed control of 3 complex autonomous systems integrated into emerging technologies for medical-social personal assistance and servicing of precision flexible manufacturing lines.*

REFERENCES

[1] Eid, M.A., Giakoumidis, N., El Saddik, A., *IEEE Access*, **4**, 558, 2016.
[2] Miranda, R.A., Casebeer, W.D., Hein, A.M., Judy, J.W., Krotkov, E.P., Laabs, T.L., Weber, D.J., *Journal of Neuroscience Methods*, **244**, 52, 2015.
[3] Bhattacharyya, S., Basu, D., Konar, A., Tibarewala, D.N., *Robotics and Autonomous Systems*, **68**, 104, 2015.
[4] Bonevski, B., Randell, M., Paul, C., Chapman, K., Twyman, L., Bryant, J., Hughes, C., *BMC Medical Research Methodology*, **14**(1), 42, 2014.
[5] Scharre, P., *Autonomous weapons and operational risk. Washington, DC: Center for a New American Security*, 2016.
[6] NeuroSky Inc., *MindWave Mobile: User Guide*, 2017.
[7] Huo, X., Ghovanloo, M., *IEEE Transactions on Biomedical Engineering*, **56**(6), 1719, 2009.
[8] Gertz, E., Di Justo, P., *Environmental monitoring with Arduino: building simple devices to collect data about the world around us*, O'Reilly Media, Inc., 2012.
[9] Van Erp, J., Lotte, F., Tangermann, M., *Computer*, **45**(4), 26, 2012.
[10] Wheat, D., *Arduino Internals*, Technology in Action, 2011.
[11] Dragomir, F., Dragomir, O.E., *Programarea în limbaj de asamblare a microcontrolerelor*, MatrixRom, Bucharest, 2013
[12] Dragomir, F., *Sisteme cu microprocesoare*, Valahia University Press, Targoviste, 2015