Volume 58, Number 4, 2017

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

AUTONOMOUS SCALE MODEL CAR WITH ULTRASONIC SENSORS AND ARDUINO BOARD

Cosmin CIUCIU, Daniel BĂRBUȚĂ, Sebastian SĂSĂUJAN, Emilia ŞIPOŞ
Technical University of Cluj-Napoca
George Barițiu nr. 26-28, 400027, Cluj-Napoca, Romania
E-mail: cosmin_ciuciu1996@yahoo.com

Abstract: An autonomous car is a vehicle that is capable to perceive the environment and navigate without human input. The paper presents the construction of an autonomous scale model car, using ultrasonic sensors and Arduino board. The designed scale model car can follow a predefined route with obstacles and turns, with the automatic change of the wheels steering angle. Data collected from the route is displayed in the smartphone application using a Bluetooth connection. The car passes the route correctly and reaches the finish line in about 30 seconds.

Keywords: *Arduino board, automatically angle change, autonomous car.*

I. INTRODUCTION

Scale model cars are the initial stage in developing fully functional cars. The concepts of embedded systems can be tested on these scale models, since they require less financial efforts and offer quick and reliable results, provided that the model is properly built. Also, the autonomous scale model cars can be used as student learning projects, in the engineering technology field.

There are four level of car automation: no automation, simple mechanical automation, smart automation and full autonomous driving [1]. Nowadays it is clear that no automation cars are history, and the goal to full autonomous driving vehicles (zero passenger intervention, multiple sensors, sensor fusion processing, machine learning, positioning technology) is set.

Vehicles all over the world are now equipped with intelligent devices that make the vehicles respond to various factors: climate control, sudden accelerations or braking, or even self-repair of modules [2].

The presence of the autonomous cars is imminent in public and industrial transport: buses, taxis, trucks, couriers [3]. There are many challenges ahead before fully autonomous vehicles can become the main mode of transportation across the globe. Safety, ethics, psychology, and other profound fields will undoubtedly be involved in decision making and policy formulation. In the meantime, huge steps are being made towards that goal [1].

The development of autonomous cars involves first checking the design and functionality of the car, in a smaller scale. For this purpose microcontrollers like Arduino [4] or Raspberry Pi are usually used.

There are previous works regarding the use of Arduino boards to design autonomous cars. In [5] a Macchina M2 board was designed using Arduino platform. The Macchina M2 allows the user to read their vehicle's electronic signals and reverse engineer them.

An autonomous Arduino-based racecar project is presented in [6]. The racecar comprises an Arduino Uno, an Adafruit motor/stepper/servo shield, ultrasonic ranging modules, assorted motors and gears, and a chassis fabricated on a 3-D printer. The purpose of the car is to stimulate students' interest in developing skills as computer programming and industrial design and fabrication.

A series of projects focused on the development of mini cars and car parts is presented in [7]. The majority of them are developed to obtain cars with Wi-Fi or Bluetooth control, obstacle avoidance, and smart parking.

Moreover, many companies from the automotive market, among them Continental and Porsche, are organizing contests dedicated to autonomous scale model cars [8]. The competitions are focused on building smaller scale autonomous cars emphasizing on simplicity, efficiency and safety.

The paper is organized as follow: Section II presents themes and objectives, Section III describes the car modules, while the results are discussed in Section IV. Section V concludes the paper.

II. THEME AND OBJECTIVES

II.1. Theme description

All the above mentioned projects are focused on solving a single aspect, rather than combining multiple functionalities. The autonomous scale model car presented in this paper is subject to five functions: manual and wireless remote control, speed adjustment, autonomous obstacle avoidance, accurate trajectory following and the possibility to stop at a specified point. The trajectory is a predefined route, with obstacles, turns and ramps.

II.2. Objectives

The objectives set for the autonomous car are:

- to be controlled by an electronic speed controller for brushless DC motor (BLDC);
- to indicate the speed, uptime in autonomous mode, travel direction, battery level in the smartphone application;
- to be able to stop in an emergency case;
- to have good cornering stability and a small steering angle;

Manuscript received November 11, 2017; revised December 20, 2017

- to contain ultrasonic sensors that measure the distances around the car;
- to stop the motor if the car is not in contact with the ground;
- to recognize the end of the route and stop.

II.3. Principle of operation

The core of the autonomous scale model car is the Arduino module - *Figure 1*. First, the user connects the smartphone

to the Arduino board using a Bluetooth module. The connection is bidirectional: the Arduino collects data about battery level, speed, direction from the various sensors and sends them to the smartphone which also sends data back.

III. CAR MODULES

III.1. Chassis

The chassis consists of an internal vehicle frame that supports the modules of the car, such as the motor, battery, steering system and electronics. The chassis should be light, strong and enduring.

Requirements regarding three types of angles (approach angle, departure angle, break-over angle) for the chassis are usually specified. The approach angle is the maximum angle of a ramp onto which a vehicle can climb from a horizontal plane without interference. The departure angle is its counterpart at the rear of the vehicle. The break-over angle is the maximum possible angle that a vehicle can drive over without the apex of that angle touching any point of the vehicle other than the wheels. A chassis with good values for the approach angles, departure angle and break-over angle is the off-road chassis.

III.2. Servo

The output shaft of a servo device can be positioned to specific angular positions by sending a coded signal to the servo. If the coded signal exists on the input line, the servo will maintain the angular position of the shaft. As the coded signal changes, the angular position of the shaft changes and the wheel steering.

Servomechanisms are differentiated per their size and our project, we used micro servo which is tiny and lightweight (23 mm x 12 mm x 29 mm), with high output power (4.8 V). This servo can rotate approx. 180° (90° in each direction), and works just like the standard kinds but smaller (operating speed: 0.1s/60°). Any servo code, hardware or library can be used to control these servos [9].

III.3. BLDC motor

A brushless DC electric (BLDC) motor is a DC electric motor where the rotation of the rotor is achieved electronically [10].

The advantages of using a BLDC motor:
- it requires less maintenance due to the absence of the brushes;
- it has higher efficiency. Hence there is no voltage drop a across brushes;
- it generates low electric noise;
- the speed range is higher. Hence no mechanical limitation is imposed by brushes or commutator.

III.4. Electronic speed control (ESC)

An electronic speed control or ESC is a device mounted onboard an electrically-powered R/C model to vary its drive motor's speed, its direction and even to act as a dynamic brake in certain controllers, perhaps even antilock braking.

An ESC can be commanded as a servo in the range [0, 180], the interval [0, 87] is reverse speeds, the [87, 94] interval is dead points and the interval [94, 180] represents the forward speeds.

III.5. Sensors

All sensors detect and respond to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing [11].

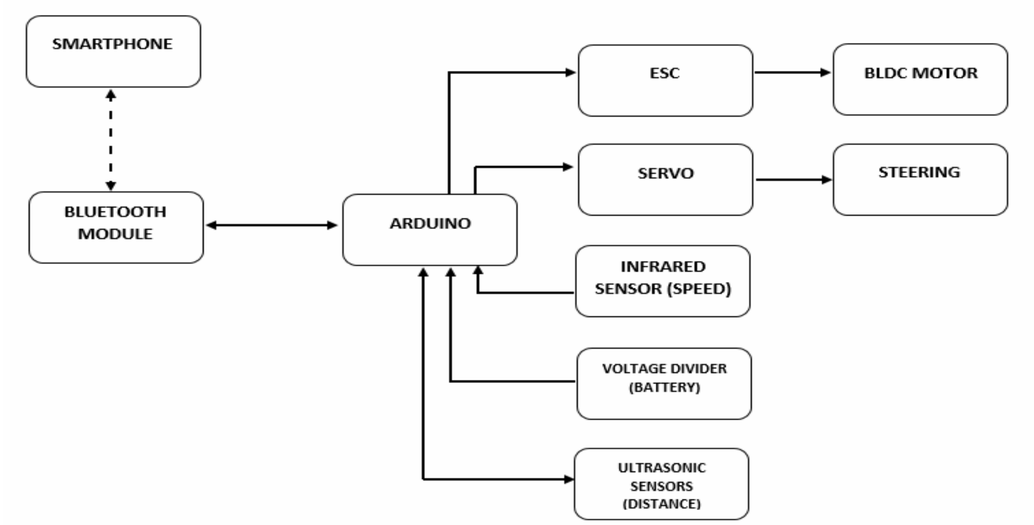

Figure 1. The block diagram

Our car uses four types of sensors: ultrasonic sensors, speed sensors, battery level sensor, and tilt sensor.

III.5.1. Ultrasonic sensors
The HC-SR04 ultrasonic ranging sensor is an economical sensor which provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit [12].

There are only four pins on the HC-SR04 (*Figure 2*): VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground).

Working steps:

-Step 1: Make a “Trig” pin of the sensor high for 10 µs. This indicates a sensor cycle.

-Step 2: 8*40 KHz pulse will be sent from the transduces of the sensor, after a while time the Echo pin of the sensor will go from low to high.

-Step 3: The 40 KHz sound will bounce off the nearest object and return to the sensor.

-Step 4: When the sensor detect the reflected sound wave, the Echo pin will go low.

-Step 5: The distance between the sensor and the detected object can be calculated based on the length of the Echo pin is high.

-Step 6: If no object is detected, the Echo pin will stay high for 38 ms and then go low.

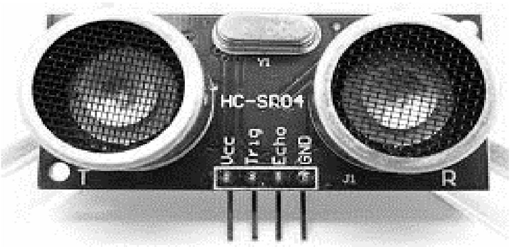

Figure 2. Ultrasonic sensor [12]

III.5.2. Speed sensor
The speed is determined by simply measuring the rate at which some event occurs. Usually this is done by counting the events for a given period (integration interval) and then simply dividing the number of events by the time to get the rate. To measure the time difference IR sensor is used. Aluminum foil is used set a reference point. Also, it reflects light thus enhances sensitivity of IR sensor. It gives HIGH output, when aluminum foil comes in front of the sensor. The command *micros()* is used to measure time which is 10^{-6} of second which can measure time lapse at high speed accurately.

III.5.3. Battery level sensor
For monitoring the battery lever has been performed a voltage divider (*Figure 3*). It has the role to divide the voltage in two parts because the Analog In pin can’t read a voltage bigger than 5V. As an example, in our case the battery has the nominal voltage 7.4 V and for decrees it, we chose R1 equal with R2 and that result the maximum voltage across the Analog Input pin of Arduino in 3.2 V.

III.5.4. Tilt sensor
The automatic speed adjustment is obtained with a tilt sensor helps. The tilt sensor consists of a cylindrical tube in which there is a metal ball (*Figure 4*). At one end of the tube is an infrared sensor: when the car is on the ramp, the ball slides, closes the circuit and the speed increases.

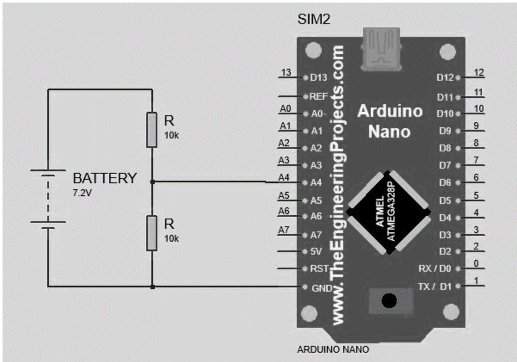

Figure 3. Battery level sensor

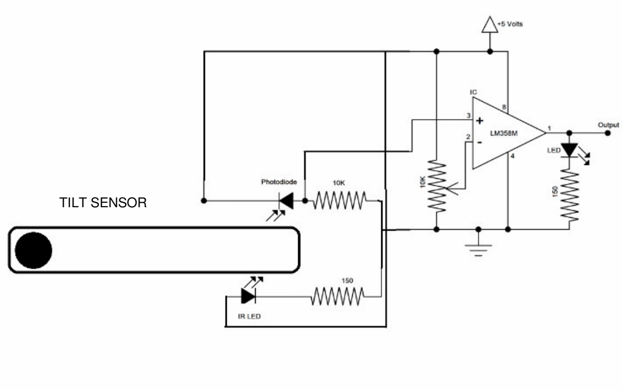

Figure 4. Tilt sensor schematic

III.6. Arduino board
Arduino is an open source electronics platform. Is widely used by teachers and students in chemistry, physics and robotics. Arduino boards are able to read various types of inputs like light on a sensor and turn it into an output like activating a motor, this can be accomplished by sending a set of instructions to the microcontroller on the Arduino board by using the Arduino language and software [4].

The Arduino board used in our scale model car is Arduino Nano. Arduino Nano is a small platform compatible with breadboard and built around a microcontroller called Atmega328P. The Arduino Nano can be powered via the Mini-B USB connection, 6-20 V unregulated external power supply (pin 30), or 5 V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. Each of the 14 digital pins on the Nano can be used as an input or output, using *pinMode()*, *digitalWrite()*, and *digitalRead()* functions. They operate at 5 V. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kΩ.

III.7. Bluetooth module
In our project, the first step is to connect the smartphone to Bluetooth and then, depending on the information sent from the smartphone application, the car executes the command attached to the received character.

HC-06 module is an easy to use Bluetooth SPP (Serial

Port Protocol) module, designed for transparent wireless serial connection setup. The HC-06 module has low power consumption, high performance and a low price. It can be powered between 3.1 V and 4.2 V and the communication current being 8 mA.

III.8. Mobile application

The mobile application makes the connection between the user and the car. The application has two distinct running modes: manual driving and autonomous driving.

In the manual mode, the user can control the car using four buttons: forward, backward, left, and right. If the input values are different from 2 (backward), 4 (left), 6 (right), and 8 (forward), the car stops (*Figure 5*).

In the autonomous mode, the application shows only a single button for starting or stopping the car. The information collected from the car is shown in both modes.

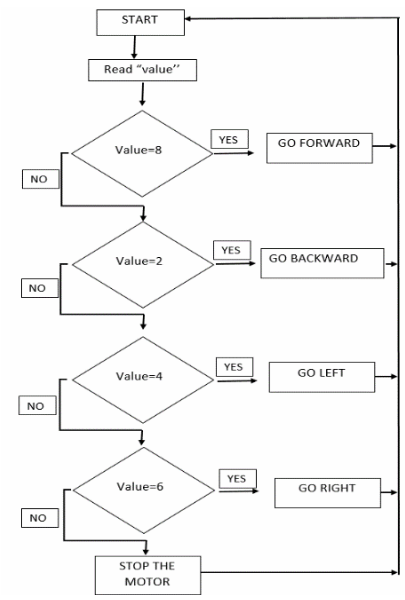

Figure 5. Manual driving diagram

IV. RESULTS

The core of the schematic circuit is the Arduino board (*Figure 6*). The track on which the test were carried out is 11.61 m long, 4.868 m wide, contains the curves, there are two obstacles on the track each with a length of 0.4 m and a height of 0.3m. The route also includes a ramp of 4.42 m long and 18° tilt.

Our car resembles an off-road car having good approach angles, so it can overcome any obstacle on track. The vehicle angles for our ensemble are: approach angle A=64.29°, departure angle D=42.75°, break-over angle B=16.78°. The steering on our car is made using Ackermann steering principles. A vehicle with Ackermann steering produces different steer angles inside-to-outside with the inside tire steered at a greater angle than the outside tire.

The system is designed for 100% Ackermann (the car has a common center of rotation for all tires). For the outside wheel, the Ackermann angle is 19.05° and for the inside wheel, the Ackermann angle is 25.94° (*Figure 7*). The steering ratio is 3:1 (when the servo turns 90°, the front wheel turns 30°). The minimum turning radius of a vehicle is the radius of the smallest circular turn (i.e. U-turn) that the vehicle can make.

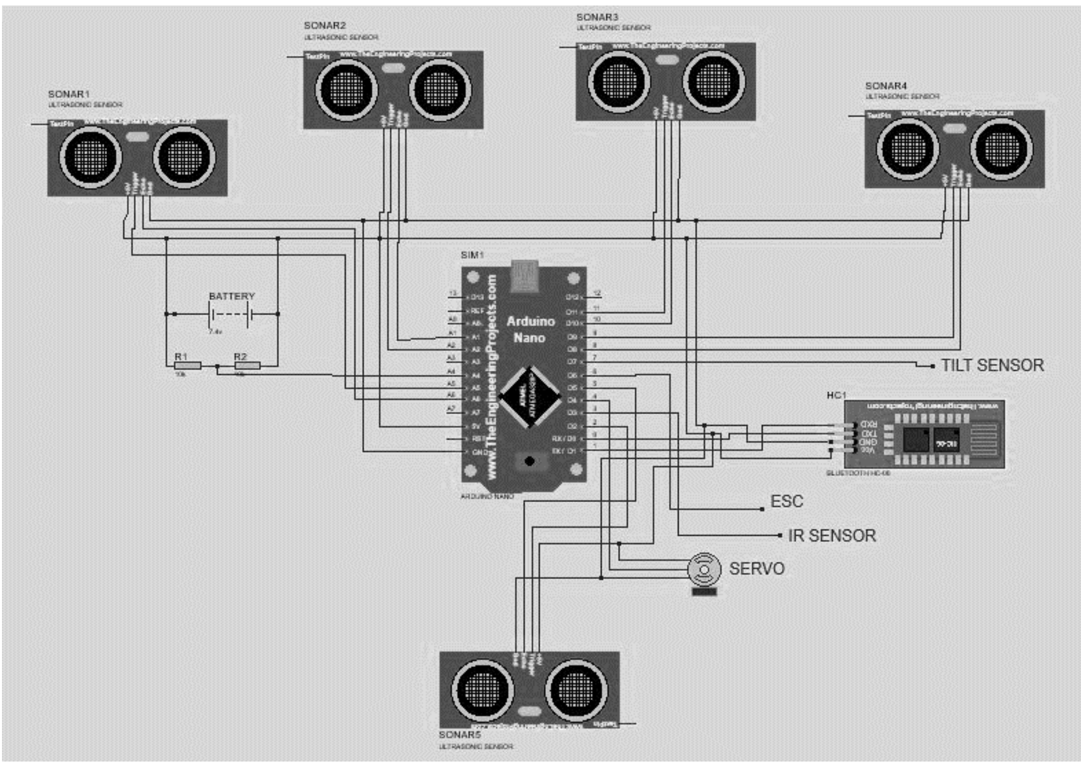

Figure 6. Schematic circuit

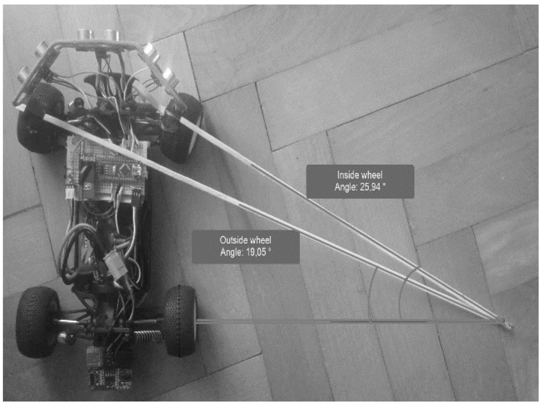

Figure 7. Acherman angle

In our case, the minimum turning radius is calculate with the below formula.

$$\pi \cdot r = 3.14 \cdot 31.79 = 99.82\ [cm] \quad (1)$$

The car is built on a scale of 1/18 and is equipped with an 18 A Brushless DC motor, which helps the car to be very fast, with a maximum speed of 32 km/h.

To collect the distance between the car and the environment we chose to use four ultrasonic sensors (*Figure 8*). Two sensors are located on the side of the car - to sense distances from the side walls, and the other two sensors are placed in front of the car for quick identification of curves. In additional there is another ultrasonic sensor placed behind the car - acts as a safety system and stops the motor if the car is not in contact with the ground.

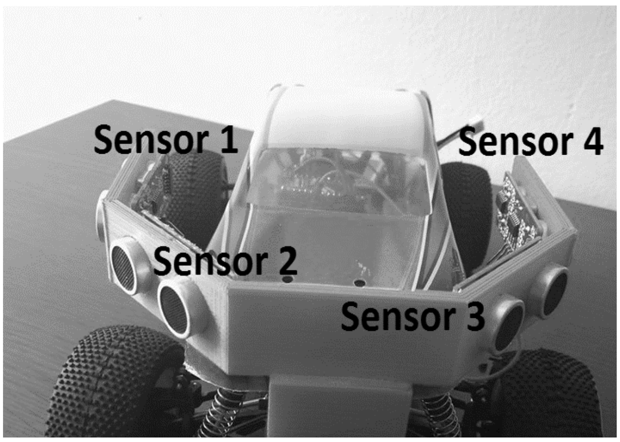

Figure 8. Position of the frontal sensors

The Android application has two running modes: manual driving and autonomous driving. The screen of manual driving is shows in *Figure 7*, it is formed from four buttons needed to drive the car in any direction and in the top of the screen a box with the data receive from the car like: direction of moving, speed, battery lever, the distance traveled in manually mode and time from start.

Manual control function has a single parameter named "value" that represent the command receive from the smartphone (*Figure 9*). If the parameter value is equals with '8' the microcontroller executes the command

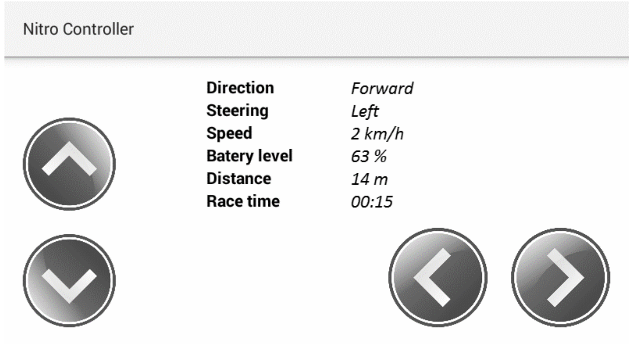

Figure 9. Manual screen and data receive from car

associated to the '8' character by going forward from command *esc.write(115).* We chose that the car to run with the '115' speed because is a fast car and the truck where it is use has small dimension for a car that can run with 32km/h.

If the value of the parameter 'value' is 'x', the car stays in place (*esc.write(89)*), being the value set for the state in place. For the other direction of travel we did the same.

The code for the automation mode of the car is formed from two parts (*Figure 10* and *Figure 11*); in the *Figure 10* is the code used for positioning the car in the middle of the track. The methods *measureRightDistance()* and *measureLeftDistance()* are first called, and they return the distance in centimeters between Sensor1, Sensor4 and the walls or obstacles.

After that, the microcontroller compares the distance measured previous and decides on which side of the *if* condition to continue. As an example, if the distance measured by the left sensor is bigger than the one measured on the right, the Arduino fulfills the conditions required by the first *if* statement. The next set of instructions has the role to maintain the car on a centered path, in the middle of the track. Using the variable *delta* we determine the difference between the values measured by the left and right sensor. If the difference is smaller than 20 cm the wheels will turn at smaller angles in interval [90; 120] determined by the *map* function according to the deviation from the center routh *delta*.

```
int rightSensor= measureRightDistance();
int leftSensor=measureLeftDistance();

 if(leftSensor<rightSensor){

  delta=rightSensor-leftSensor;
   if(delta<20){
   map_left=map(delta,0,20,90,120);
   directie.write(map_left);
   }
   }else{

   delta=leftSensor-rightSensor;

   if(delta<20){
   map_right=map(delta,0,20,90,60);
   directie.write(map_right);
   }
 }
```

Figure 10. Automation code - positioning the car in the middle of the track

For a better approach of the tight turns and curves, we added a set of two sensors in the front of the car placed at an angle of 45° (Sensors2 and Sensors 3 from *Figure 6*). These sensors are meant to analyze the track in front of the car and detect any change in the route to follow. If the distance measured by any frontal sensor is greater than 120 cm, the wheels start turning depending on the distance measured, using another *map* function – *Figure 11*.

```
if(middleLeft>120 && middleRight<40){
  map_left=map(middleLeft,120,200,120,180);
  directie.write(map_left);
  }

if(middleRight>120 && middleLeft<40){
  map_right=map(middleRight,120,200,60,0);
  directie.write(map_right);
  }
```

Figure 11. Automation code - curves code

In order to fulfill the requirements regarding autonomous obstacle avoidance, accurate trajectory following and ramp climbing, the car is designed with automatic speed adjustment (tilt sensor). The car needs high speed to climb the ramp – *Figure 12*, but low/moderate speed to avoid the collisions with the walls or obstacles.

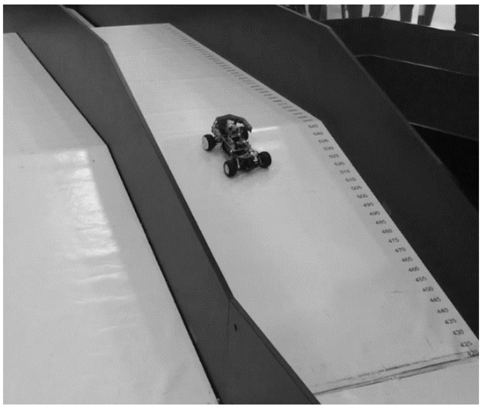

Figure 12. Ramp climbing

V. CONCLUSIONS

Scale model cars are widely used for testing autonomous functionalities, before implementing them on real size cars. The autonomous scale model car proposed in this paper is subject to specific requirements and follows a predefined route with obstacles and turns, with the automatic change of the wheels steering angle.

The track on which the test were carried out is 11.61 m long, 4.868 m wide, contains 10 curves, there are two obstacles on the track each with a length of 0.4 m and a height of 0.3m. The route also includes a ramp of 4.42 m long and 18° tilt. All the established objectives have been met. The car works in both autonomous and manual mode, and can be controlled from the smartphone using the application to which we connect with the Bluetooth module.

The car traverses the route in approximately 30 s, and stops within 3 s after it has no contact with the ground.

REFERENCES

[1] J. VanWashenova, "Challenges to fully autonomous cars", *Embedded*, 2006. [Online].
Available: https://www.embedded.com/electronics-blogs/say-what-/4442830/Challenges-to-fully-autonomous-cars [Accessed: November, 2, 2017].

[2] D. G. Dhyaneswaran, M. Harish, "Embedded systems and autonomous car", *International Journal of Advances in Science Engineering and Technology*, ISSN: 2321-9009, vol. 2, iss. 2, pp. 6-9, 2014

[3] D. Alexander, "How Self-Driving Cars Will Change The World", *Forbes,* 2013. [Online].
Available:https://www.forbes.com/sites/pikeresearch/2013/11/13/how-self-driving-cars-will-change-the-world/#593d5d7c6384 [Accessed: July, 11, 2017].

[4] M. Honrubia, "How is Arduino Technology used in Automated Vehicles?", *Ennomotive*, 2017. [Online].
Available: https://www.ennomotive.com/automated-vehicles-arduino-technology/ [Accessed: September, 27, 2017].

[5] Chantelle Dubois, "Open Source Solutions for Your Automotive Projects and Prototypes", *All about circuits*, 2017. [Online].
Available: https://www.allaboutcircuits.com/news/open-source-solutions-for-your-automotive-projects-and-prototypes/ [Accessed: November, 1, 2017].

[6] W. Rosen, Y. Ertekin, M. E. Carr. "An Autonomous Arduino-based Racecar for First-Year Engineering Technology Students.", *Proceedings of the 121st ASEE Annual Conference & Exposition*, Indianapolis, IN., 2014.

[7] Arduino project hub. [Online].
Available: https://create.arduino.cc/projecthub/projects/tags/car [Accessed: October, 26, 2017].

[8] Continental Automotive Romania, TU Iasi, *Electromobility contest*, 2017. [Online].
Available: http://www.electromobility.ace.tuiasi.ro/ [Accessed: June, 16, 2017].

[9] Micro Servo motor SG90, *ePro Labs*
Available:https://wiki.eprolabs.com/index.php?title=Micro_Servo_ motor_SG90 [Accessed: October, 17, 2017].

[10] "Brushless DC Motor, How it works?", *Learning engineering*, 2014. [Online].
Available: http://www.learnengineering.org/2014/10/Brushless-DC-motor.html [Accessed: October, 26, 2017].

[11] Sensor, *What Is*. [Online].
Available:http://whatis.techtarget.com/definition/sensor [Accessed: October, 3, 2017].

[12] Ultrasonic Ranging Module HC - SR04, *Elec Freaks*, [Online].
Available:http://www.micropik.com/PDF/HCSR04.pdf [Accessed: November, 2, 2017].