CARduino: A Semi-automated Vehicle Prototype to Stimulate Cognitive Development in a Learning-Teaching Methodology

2

Everton Rafael da Silva and Breno Lisi Romano

Abstract

This paper aims to present the development of an semi-automated vehicle prototype using Arduino and sensors that will be controlled by software developed for Android that can simulate the execution of manual and semi-automatic paths according to the user's needs. It will be presented both the physical and the logical development of the proposed vehicle and will be presented a set of experiments to demonstrate the feasibility of its application to different situations, emphasizing cognitive development in a learning-teaching methodology for children, youth and adults. Finally, we carried out a cost analysis in the market, based on some e-commerces available, to design the physical development of the proposed prototype.

Keywords

Component • Arduino • Android • Driver software • Prototype • Cognitive development

2.1 Introduction

This paper aims to show the feasibility of developing a semi-automated vehicle prototype, entitled CARduino, with low cost using the Arduino microcontroller that is controlled by an Android application using Bluetooth communication.

In CARduino locomotion, DC motors powered by batteries are used, in addition to reduction boxes to increase the torque of the motors and circuits used to reverse the direction of rotation.

It is hoped that this research will contribute to the cognitive development of children, youth and adults, complementing existing research in these areas, as will be presented to the course of this paper [1]. Additionally, as a future complementary research, this project will also assist beginners in programming to understand in a practical way, the operation of programming using blocks (block diagrams) and viewing the operation in practice of the developed program [2].

Another motivation of this work was to unite the practical part of the construction of an automated vehicle prototype with programming for mobile devices with the Android operating system, one of the main market of operating systems. Moreover, as major contributions, will be shown in detail:

- the design of a embedded software in the automated vehicle prototype Arduino;
- the physical construction of this prototype, configuring the Bluetooth communication on at a low level using a specific sensor for this;
- the software design in Android that will communicate with the prototype of the proposed vehicle;
- simulation of the prototype proposed to illustrate its operation to be applied in cognitive development; and
- a cost analysis for design of the proposed prototype.

The following sections will present these outstanding contributions previously.

E.R. da Silva (✉) • B.L. Romano
Federal Institute of Education, Science, and Technology of Sao Paulo, Sao Joao da Boa Vista, SP, Brazil
e-mail: evertonrafael@ifsp.edu.br; blromano@ifsp.edu.br

S. Latifi (ed.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing 558, DOI 10.1007/978-3-319-54978-1_2

2.2 Related Works

In this section, we introduce some papers related to this research that stood out in recent years in Brazil and in the world: Wirsing (2014) shows the sending and receiving data via Bluetooth, used Arduino and a device with Android operating system [3]; Cardoso (2014) had a home automation system controlled Arduino [4]; Moreira (2012) who presented a weather station model for measuring temperature and humidity of the Amazon [5]; Scriptore (2014) shows the use of a distributed database, which stores information of humidity and temperature measurements using Arduino and Android [6]; Steps (2011) shows the operation of vehicle accessories used Arduino [7]; Lim et al. (2014) discloses a system developed for measuring environmental factors Arduino in a farm field [8]; and Watve (2015) has a verification system for plants using Arduino and Android [9].

In working Wirsing (2014), the Bluetooth device configuration was presented, showing how were made the connections enters the Arduino device and the Bluetooth communication module HC-05 (Fig. 2.1):

As is shown in Fig. 2.1, the Bluetooth module requires 4 connections, which are: the communication ports TX and Arduino RX, GND (Ground or earth) and a 3.3 V connection to power supply. The Android device will connect via Bluetooth with the Arduino.

Some settings are also presented in this paper in relation to communication with Android, where Android has to check if your Bluetooth is active or not, and if it is not active, you have to turn it on. Only after made these checks, Android Arduino can send and receive data via Bluetooth.

In Cardoso's work (2014), part of home automation is presented using the Arduino. This paper presents another form of communication between mobile devices and the Arduino - GSM (Global System for Mobile - Global System for Mobile). With the implementation of GSM as a new type of communication it was also used another communication service - SMS (Short Message Service - Short Message Service) - which could be sent texts with up to 160 characters and a low financial cost.

In this work, has presented a data conversion problem of type float to char, which, to solve the problem, a data conversion table was made. The scanned data had to be converted to ASCII standard for the GSM module could send SMS with the obtained data. The source of this work was conducted in order to work with other similar sensors, which may add other or work only with the proposed sensor.

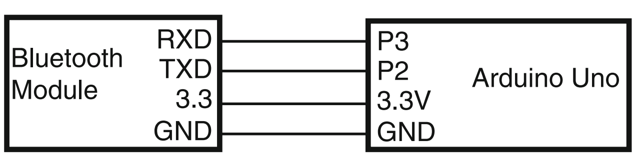

Fig. 2.1 Connections - bluetooth and Arduino module

In working Scriptore (2014) a database technology was presented distributed, where they are stored read data of humidity and temperature sensors connected to the Arduino. We used a WebService as a solution to systems integration and communication between different applications, thus enabling the storage of data in a MySQL database (Database Management System that uses SQL or Structured Query Language). To achieve the objective of this study also used up an Ethernet Shield - module for Arduino that has the connection function with network and data storage SD Card (Secure Digital Card or no Volatile Memory Card).

Communication and transmission of data and Arduino Ethernet Shield was made by a technology called JSON (JavaScript Object Notation), a lightweight format for the exchange of computer data. The synchronism of the data is done by a WebService PHP (Hypertext Preprocessor - programming language for internet) that makes a request to the Ethernet Shield WebService and returns a query that is stored on the SD Card. With this, a list of measurements is generated and displayed on a software on Android.

Passos (2011) shows in his work a prototype to simulate the control of automotive features. The prototype has some servomotors, which are connected to the Arduino and Bluetooth module. In this work, it is also used a software called Amarino developed for Android that is the prototype of control using the cell accelerometer.

As forgetting a key problem in the vehicle is common, the Passos's research (2011) presents a solution to this problem. With Android software, the car user can connect to the prototype and trigger the opening of locks or car windows, not requiring the intervention of a professional to its opening, so without damaging the car.

Watve (2015) presents an implementation of a system for real time data read from a factory using Arduino and Android. The system is to monitor the humidity and temperature sensors. The software developed for Android uses features such as camera and microphone. The camera is used to log into the factory environment and microphone to receive voice commands within the software. The software also has reporting using graphics, thus presenting the performance of sensors of each environment where they are used. All sensors are used together with the Arduino UNO.

2.3 CARduino Development

In this section, you will be presented as we developed the CARduino - automated vehicle prototype using the Arduino to control sensors / motors and Android to control the interaction of the vehicle with the user. First, it will be presented the tools and components used in vehicle development. It will then be shown how to perform the Arduino setup along with Bluetooth HC-05 module and the configuration of the HC-SR04 Ultrasonic Sensor operating in conjunction with the Buzzer (loudspeaker). Finally, development of the complete vehicle prototype will be presented, focusing on both the physical (hardware) and the development of embedded software and the Arduino controller software on Android.

2.3.1 Setting the Bluetooth Module Integrated with Arduino

For a Bluetooth connection to the HC-05 module, there is the need for a hardware-based configuration to enable the administrator mode and a configuration via software to change the name and Bluetooth device password. When Bluetooth modules purchased, they come with a factory default setting, so the configuration is intended to customize the device name, change the password for security measures, and display the MAC address required for connections. The hardware configuration via It is diagrammed in Fig. 2.2.

As shown in Fig. 2.2, a link between 3.3 V Arduino output port directly to pin 34 of the Bluetooth module has been made, aiming to enable its run mode, allowing to amend its factory settings.

Also links to power the Bluetooth module powered directly from the Arduino 5 V output port and GND connected to the VCC and GND of Bluetooth module doors were made.

The TX and RX pins are Bluetooth module communication pins (TX - RX and transmission - reception) and must be configured shape illustrated in Fig. 2.2. The use of resistors was needed to create a voltage divider for the RX pin, thus preventing damage to the component by excessive electric current. For this reason, we used two 330Ω resistors (ohms) in series reduces the electric current of 5 V to 2.5 V.

In order to set up the Bluetooth Module HC-05 changing your ID, recovering your MAC address and changing your password pairing, it was held the command sequence shown below in Serial Monitor the Arduino IDE:

- **AT + NAME** – Returns the device name;
- **AT + NAME = NOVO_NOME**→Return OK - Sets a new name for the Bluetooth device;
- **AT + ADDR** – Returns the MAC address of the device;
- **AT + PSWD** – Returns the Password (password) to connect to the device;
- **AT + PSWD = NOVA_SENHA**→Returns OK - Sets a new password for the Bluetooth device.

This configuration allows the module to be configured according to user needs, to use and integration with Android.

Table 2.1 shows the source code that allows the Bluetooth module configuration as previously reported. There was the need to use a library called SoftwareSerial.h, for the modification of the serial ports to be used by Bluetooth, in order to avoid conflict between the serial port of the computer with the Bluetooth HC-05 module also works with the serial port.

The Bluetooth Module works with a speed different from the normal serial port (value 9600), so to set this value, we used the command BTserial.begin (38400). This setting should always be within the setup () function code, it will be executed only once, without the need for new configuration later.

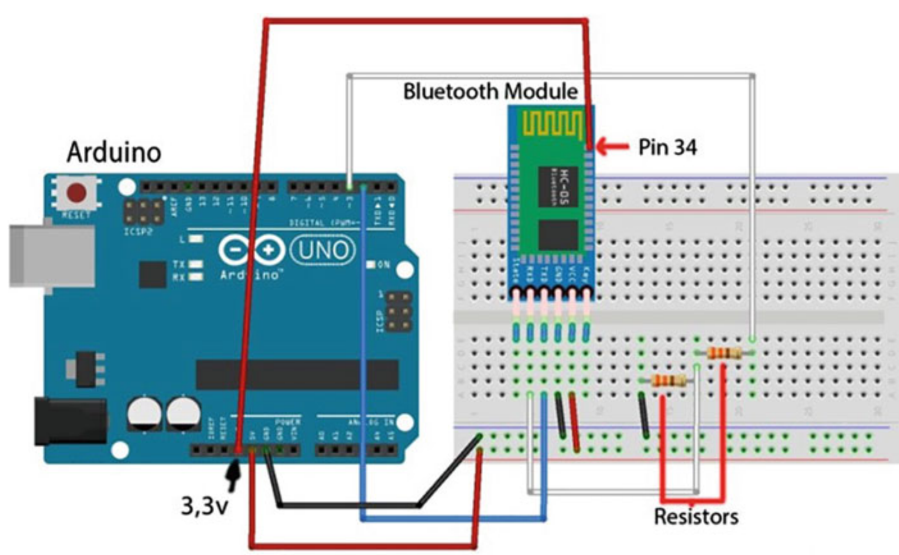

Fig. 2.2 Configuration schema of the Bluetooth module

2.3.2 Physical Prototype (Hardware) Automated Vehicle

Figure 2.3 shows the upper and lower parts of CARduino prototype in this research, containing the following components: a Bluetooth module HC-05, an ultrasonic HC-SR04 sensor, a buzzer, fifteen Jumper cables, two motors with reduction and two modules batteries, one with four rechargeable batteries to power the motors and the other with a high performance battery to power the remaining components.

Table 2.1 Code for the Bluetooth configuration

```
#include<SoftwareSerial.h>
SoftwareSerialBTserial(22,24); // RX | TX
char c = ' ';
void setup(){
    Serial.begin(9600);
    Serial.println("Arduino is ready!");
    BTserial.begin(38400);
}
void loop(){
    if (BTserial.available()){
        c = BTserial.read();
        Serial.write(c);
    }
    if (Serial.available()){
        c = Serial.read();
        BTserial.write(c);
    }
}
```

As can be seen in Fig. 2.3, the top of the breadboard prototype are the HC-SR04 Ultrasonic sensor the buzzer and the batteries. At the bottom of the prototype were coupled Arduino MEGA, the Motor Shield and two engines. All connections necessary to design the CARduino hardware prototype are shown in the schematic of Fig. 2.4.

In this prototype, it could have been eliminated engine power battery but overtax the other battery, having a lot shorter than expected. For this reason, we used four batteries with on / off switch for driving the motors.

Note that we chose to use the Arduino MEGA in CARduino design based on comparative analysis of the different types of Arduino presented on its official website (www.arduino.cc, 2016). This comparison shows that there is a big difference between the amount of communication ports (analog and digital) of the Arduino Uno and Arduino MEGA, most widely used models in the market. The choice for Arduino MEGA justified by this have more analog and digital ports to meet all requirements to be used in this work.

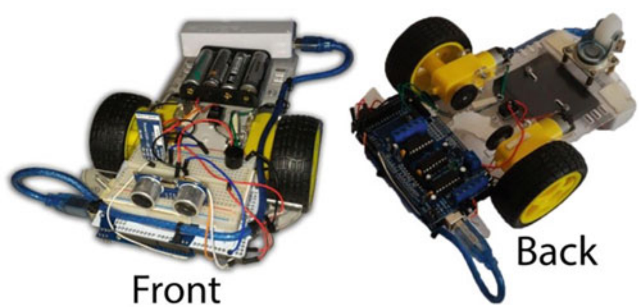

Fig. 2.3 Full prototype of CARduino

2.3.3 Development of Software in Boarded Arduino

In this section, we will present the development of embedded software on the Arduino to control sensors and automated vehicle engines. The software used for the development was the Arduino IDE 1.65.

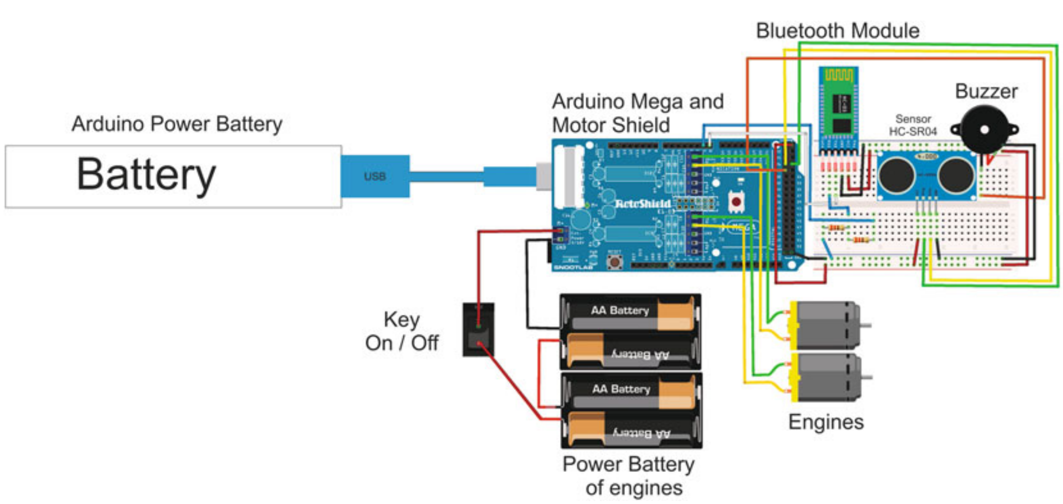

Fig. 2.4 Complete physical schema of CARduino

Because the source code developed and embedded in Arduino was extensive, this section will be presented the operating logic of the embedded software.

The setup () function of the embedded software is the function that must be set all input and output pins of the Arduino and other issues that will be performed only once by the embedded software on the Arduino. The Serial.begin command (9600) means that communication is initiated via the serial port with a 9600 baud rate, which determines the data communication via Bluetooth with the driver software developed on Android.

In the loop () function, it is developed every part of the code that will be repeated forever, following the operation and status changes of embedded software. Initially, defines the instructions for operating the automated vehicle engines. The movement is performed according to the data read by the serial port. This data is received by the Arduino Bluetooth module via the serial communication port. It is worth mentioning that all communication performed by Bluetooth HC-05 module is the serial type.

To illustrate the operation of forward movement of the automated vehicle when the number 8 is read through the serial port, the software will enter a condition, as shown in the last four lines of Table 2.2. First, should stop all motors and then it runs the drive function forward.

Following the same analogy forward movement, when it is read the value 4, the condition executes a piece of code that will move the vehicle to the left. Similarly, when the read value is the number 6, the vehicle will move to the right and, when the value 2 is read, the software performs the backward motion or reverse. Finally, if the value is 5, the software performs the stop command of all motors.

Still on the drive engine, you must determine the maximum speed of each engine so that they have a move in the same speed and in what direction they will turn. For example, in moving forward, the two engines should turn clockwise at the same speed so that the CARduino move straight. The setSpeed () function, available in AFMotor.h library, is responsible for determining the engine speed coupled in Motor Shield used.

It is noteworthy that a function was done to prevent the collision of automated vehicle with obstacles that may be in the path. This collision function works directly with the ultrasonic sensor HC-SR04 and requires the Ultrassonic.h library for its operation. The value of the distance of obstacles, in centimeters, is returned by ultrassonic.convert function ().

Table 2.2 Operation of CARduino engines

```
if(Serial.available() >0)
{
intentrada = Serial.read();
direcaoAT = entrada;
switch (entrada) {
case '8':
pararMotores();
moverParaFrente();
break;
```

The embedded software on the Arduino for CARduino prototype can be found in the link: https://github.com/EvertonRafaeldaSilva/CARduinoArduinoCode.

2.3.4 Controller Software Development in Android CARduino

In this section, you will be presented the source code of Android software, and explained its operation.

First, one must make a setting within the general configuration file Android (AndroidManifest.xml) to allow the use of the Bluetooth mobile device. For the use of Bluetooth, this setting is mandatory and without it, the Android application can not connect to another Bluetooth device or activate the Bluetooth mobile [10].

Figure 2.5 illustrates the interface developed for CARduino driver software on Android. As can be seen, there are two types of user interactions with the vehicle. The first interaction is manual and there are 5 buttons that can be used to move the CARduino, which are: Front, Ré, right, left and stop. The second interaction is designed to work with custom paths, aiming mainly assist in research related to cognitive development, as previously reported.

As can be seen in Fig. 2.5, to work with this second interaction, one must establish the path to be traveled by CARduino defining a sequence of commands that matches the direction of motion (F - Front T - Back, D - Right, E - Left, P - Stop) and the quantity of centimeters (F or T) or degree (D or E) in which the vehicle should move. For example, if the value entered is F30, the vehicle must move forward and run this command by 30 cm.

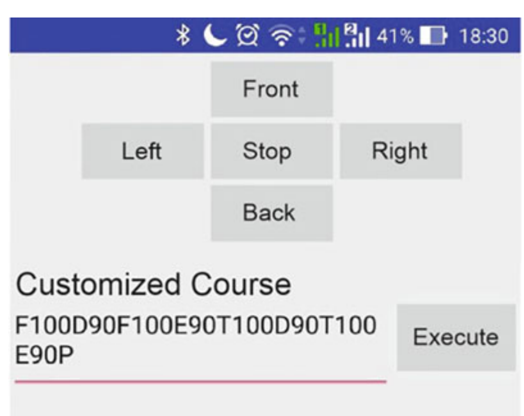

Fig. 2.5 Controller software interface developed on Android

As CARduino was developed using two DC motors, it should be parameterised how long each of the motors is operated by the driver software developed in Android to perform the movements of the vehicle. To meet this goal, we performed a set of ten manual measurements of the prototype developed in this research simulating its motion in a straight line of 100 cm and a 360 °.

In view of the calculations performed, we obtained the result of 0.0218 seconds to travel 1 cm and 0.0051 seconds to go 1. The driver software for Android CARduino can be found on the link:

https://github.com/EvertonRafaeldaSilva/CARduino.

Once finalized the development of Arduino controller software on Android, as reported in this section will be presented in the next section a simulation of its operation.

2.4 Vehicle Operation of Automated Simulation

In this section, we will present the functioning of the automated vehicle in this research through paths sent by the controller software developed on Android. 3 experiments are presented which were used to simulate such an operation, focusing on the functionality of sending a complete path by the controller software to the Arduino, once the manual control has the feature automated direct feedback vehicle.

It is noteworthy that the purpose of this simulation is not only to show the operation of the integrated CARduino with software controlled, but also already provide an initial analysis of the feasibility of using this research to assist in the cognitive development of children, youth and adults. For this reason, the choice of the experiments was based on cognitive exercises ever conducted in other studies, highlighting the research CHIOCHETI (2015).

As already presented in Sect. 2.3.4, the commands to be sent by the controller software on Android for CARduino must be a letter, which identifies the movement (F - Front, T - Back, D - Right, E - Left, P - Stop) and a value in centimeters (front or Rear) or in degrees (Right or Left).

As can be seen in Fig. 2.6, it was defined as Experiment I a route in the shape of a square to treat cognitive issues involving mathematics and logic. The CARduino start the route from the starting point and will go up to 240 cm return to starting position.

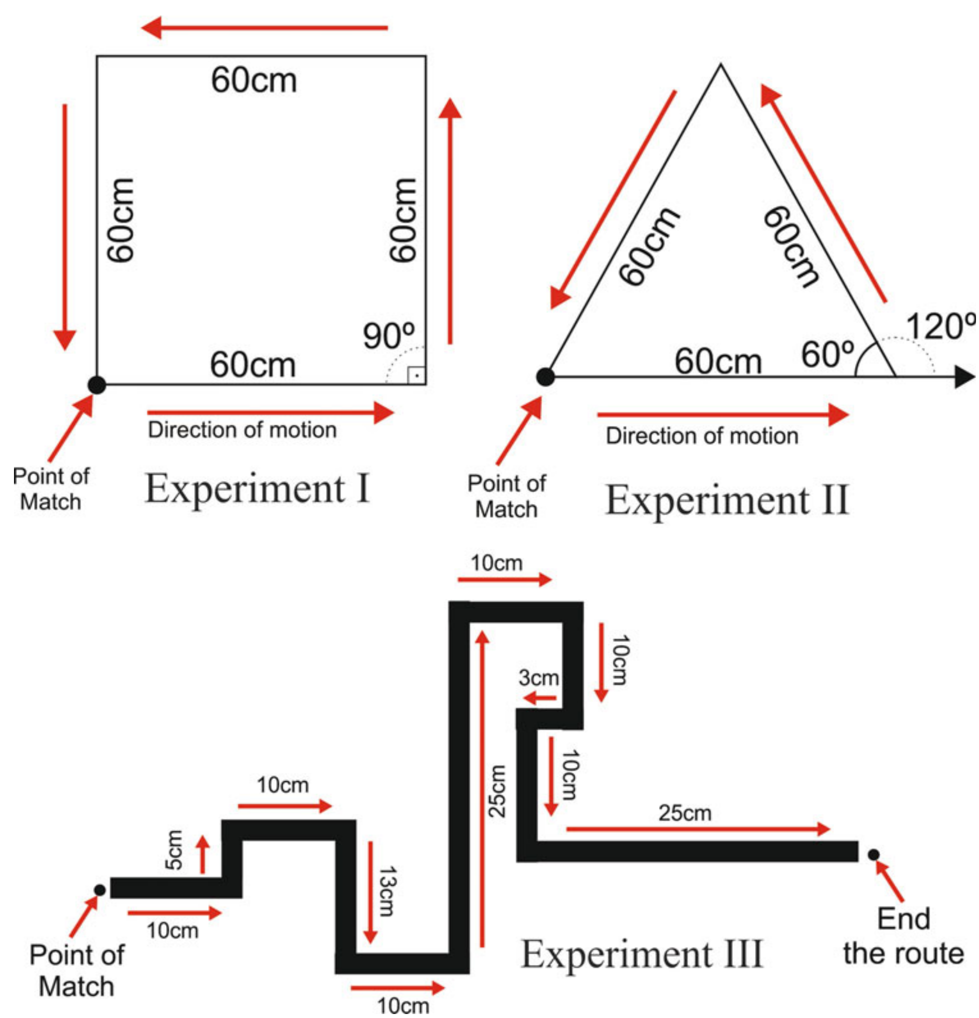

Fig. 2.6 Defined experiments in research: Experiment I - Square Path, Experiment II - Triangle Trail and Experiment III - Full Path - Customized Course

To accomplish this route, the sequence of commands is needed: F60 → E90 → F60 → E90 → F60 → E90 → F60→P.

As Experiment II, we chose to simulate the path of an equilateral triangle with sides measuring 60 centimeters. Thus, this path would treat cognitive issues logic and mathematics somewhat deeper in relation to Experiment I, since one must know the rules that the sum of interior angles of a triangle is 180 degrees and complementary angle an equilateral triangle is worth 180 by subtracting its value.

The route is shown in Fig. 2.6 Experiment II. The vehicle starts moving from the starting point and should follow the direction of the red arrows. The sequence of commands to accomplish this path is: F60 → E120 → F60 → E120 → F60 →P.

To turn the vehicle right triangle in each line join, should apply a value of 120 °, where 60 ° is within the triangle, more than 120 ° outside the triangle, resulting in 180 °.

Finally, as Experiment III, was chosen to simulate a complete path with variations of routes in length and changes in direction, again treating cognitive issues of logic and mathematics. Figure 2.6 illustrates this third experiment.

In this way, we perceive a greater amount of movement of the vehicle. The command to run the route is: F10 → E90 → F5 → D90 → F10 → D90 → F13 → E90 → F10 → E90 → F25 → D90 → F10 → D90 → F10 → D90 → F3 → E90 → F10 → E90 → F25 →P.

2.5 Analysis of Costs CARduino

To meet one of the objectives proposed in this research, this section an overview of the construction costs of CARduino prototype and feasibility of studying this type of technology will be presented. They found 3 sites specializing in trading Arduino components in Brazil: Filipeflop (www.filipeflop.com), Laboratório de Garagem (www.labdegaragem.com.br) and Usina Info (www.usinainfo.com.br).

As can be seen in Table 2.3, the values of the components shown, refer to a query made in January 2016. For this analysis disregarded the value of the resistors.

Given the values compared in Table 2.3, the cost of USD 105.11 to produce a CARduino prototype can be considered low, especially for its variety of applications in use, the gains highlighting that can be obtained for cognitive development children, youth and adults.

Note that this section had only a cost analysis for the development of the physical part (hardware) of CARduino, not including the time effort for the development of both the embedded software on the Arduino as Android in controller software.

Table 2.3 Price comparison component

Component	FilipeFlop	Laboratório de Garagem	Usina Info
Arduino MEGA	USD 31.25	USD 28.12	USD 42.34
Motor Shield	USD 10.92	USD 13.79	USD 14.75
Chassis/ Engines	USD 34,37	USD 18.37	USD 45.00
Sensor HC-SR04	USD 5.91	USD 4.77	USD 5.42
HC-05 Module	USD 15.61	USD 13.26	USD 16.39
Jumpers	USD 6.22	USD 3.73	USD 4.89
Resistors	USD 0.04	USD 0.04	USD 0.11
Total site	**USD 103.38**	**USD 82.09**	**USD 128.86**
Average price	**USD 105.11**		

2.6 Conclusion

This project aimed to design an automated vehicle prototype built with Arduino and controlled with software developed on Android that can perform manual or automatic paths.

Until now research and analyzing the simulation of experiments shown, it is believed that it is feasible to use the prototype designed to cognitive development, for future users can learn to insert custom paths that can process logic issues and more complex mathematics allowing the prototype perform the desired movements.

Analyzing the financial costs of design, it is believed that it is feasible to construct this type of prototype because it presents a low cost of the components used, particularly if they choose in a large scale production. It is worth noting that both the Java programming language as the language for Arduino in development are free, not burdening additional costs for the development of the project, pointing out that this applies also the tools used for development.

An additional contribution of this research is the practical presentation of how to integrate applications developed on Android with circuits produced using Arduino. In addition, not found in the literature any work reportasse systematically, how to enable and use the administrative mode of the Arduino Bluetooth module, this work presented in detail those settings at both the software and the hardware level.

To complement this research, it is suggested as a future work quality analysis of developed code-sources from the point of view of Software Testing to address security issues in embedded software. It is also recommended to apply the automated vehicle prototype developed in conjunction with the controller software for Android in groups of children, adolescents and adults to prove the viability of the same on cognitive issues, since taking the opportunity to identify shortcomings and make the necessary refinements. Finally, one can study the feasibility of adding new features and sensors to increase the range of use of the prototype, for example: rover, access to inhospitable areas, among others.

References

1. Chiocheti, M. S, & Romano, B. L. (2015). Uma abordagem para analisar o desenvolvimento do raciocínio lógico com apoio computacional. Monografia do Curso de Tecnologia em Sistemas para Internet – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus São João da Boa Vista.
2. Ascencio, A. F. G. (2010). *Fundamentos da programação de computadores: algoritmos, pascal e C/C++* (2nd ed.). São Paulo: Pearson Prentice Hall.
3. Wirsing, B. (2014). *Sending and receiving data via bluetooth with an android device*. White Paper Android Developer.
4. Cardoso, L. F. C. (2014). Sistemas de automação residencial via rede celular usando microcontroladores e sensores. *Revista de Engenharia da Universidade Católica de Petrópolis, 8*(2), 68–83.
5. Moreira, A. S., Portela, A. M., & Silva, R. (2012). Uso da plataforma Arduino no desenvolvimento de soluções tecnológicas para pesquisas de dados atmosféricos na Amazônia. 2012, Revista Científica AMAZÔNICA ISSN 2179 6513, p. 119.
6. Scriptore, D. B., & Junior, José de M. (2014). Banco de dados distribuído para consulta de temperatura e umidade utilizando Arduino e Android. 2014. 6p. Paranavaí, 2014.
7. Passos, B. P. (2011). Sistema Bluetooth para controle de acessórios veiculares utilizando Smartphone com Android. 2011, 83p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) Centro Universitário de Brasília UniCEUB, Brasília, 2011.
8. Lim, W., Torres, H. K., Oppus, C. M. (2014). *An agricultural telemetry system implemented using an Arduino-Android interface*. Philippines: IEEE Conference Publications.
9. Watve, O. J. (2015). Implementation of real time factory information system using arduino and android. IEEE Conference Publications.
10. Glauber, N. (2015). *Dominando o Android* (1st ed.). São Paulo: Novatec.

Springer

http://www.springer.com/978-3-319-54977-4

Information Technology - New Generations
14th International Conference on Information Technology
Latifi, S. (Ed.)
2018, XIX, 985 p. 599 illus., 455 illus. in color., Hardcover
ISBN: 978-3-319-54977-4