

Balance Control of a Small Scale Sphere with an Innovative Android Application

G.Vordos

*Dept. of Industrial Design
and Production Eng.
University of West Attica
Athens, Greece*

A.Chatzopoulos

*Dept. of Industrial Design
and Production Eng.
University of West Attica
Athens, Greece*

M.Papoutsidakis

*Dept. of Industrial Design
and Production Eng.
University of West Attica
Athens, Greece*

D.Piromalis

*Dept. of Industrial Design
and Production Eng.
University of West Attica
Athens, Greece*

Abstract— The purpose of this study is to check the balance of the robotic sphere. The system is designed to operate with an internal mechanical structure to control the motion of the sphere in relation to its balance. The specificity of controlling the sphere, in a limited operating space, has been crucial in selecting the components used. The general structure of the application is based firstly on the use of an Arduino-family microcontroller, using the necessary peripherals and secondly, on the remote control application for Android devices. The combination of these factors help to achieve wireless communication with each other and the control of the motors. The following chapters will analyze the microcontroller code and the application that performs the construction control. Specifically, the analysis includes the construction of the ball and the adjustment of the microcontroller, the Bluetooth interface of the device, the motors and their controller and finally the remote-control application via the MIT platform App Inventor II for Android devices. Once the balance of the sphere is achieved, the purpose of this study is met, as soon as its application is feasible in the educational process. In terms of cost, the choice is satisfactory, since the structural features provide the ability to copy or upgrade the robotic sphere, in an economical way. In the future, the reliability of the ball balancing system can be upgraded, starting from the individual features, such as the spherical casing and the power supply and reaching the full automated balance of the robotic sphere.

Keywords—Android Application; Balanced Sphere; Remote Control, Mechatronics

I. INTRODUCTION – PURPOSE OF THE PROJECT

The purpose of this project is to construct a system of a balanced sphere, wirelessly controlled by an android application via MIT App Inventor II. The project will be referred as “Balanced Sphere” to the rest of the text.

II. DEFINITIONS

A. Robot

Robot or mechanical worker is called any mechanical device that has the ability to substitute the human factor on various tasks. Those tasks may include dangerous operations or the increase of productivity. Robots are divided into two categories: the fully autonomous and these who need direct human control [1]. The major feature of robots is their degree of freedom. The degree of freedom depends on the number of independent parameters that determine the location of an object in the space and generally define the agility of each robot [2].

B. Microcontroller

The Balanced Sphere is controlled by a microcontroller. Microcontrollers are a class of processors. More specifically, they are a microprocessor variant, as they also have the ability to operate with minimum external components, due to the large number of embedded subsystems available on the board of each microcontroller board. The main difference between microprocessors and microcontrollers is the increased computing power of microprocessors [3]. The application control board belongs to Arduino family, the exact version is Arduino Pro Mini, rated at 5V at 16MHz processing speed. The control board microcontroller, is the Atmel AVR Atmega328P [4].

C. Remote Control Methods

The Balanced Sphere control is achieved using a remote control technology. Remote control is the method of controlling one or more devices remotely. In particular, Remote Controlled vehicles (R/ C) are called the distance-controlled vehicles which use specialized transmitter-receiver technology [5]. The main transmitter-receiver technologies are Infrared (IR) [6] Radio Signal (RF) [7] and Bluetooth [8]. The Balanced Sphere uses the Bluetooth technology.

III. MAIN PARTS OF BALANCED SPHERE:

A. Arduino Pro Mini microcontroller

Arduino is an open source, user-friendly application on level programming and implementation platform [9]. The main advantages of this platform are the low cost, the transferability and the scalability. The construction of the Balanced Sphere uses the Arduino Pro Mini version. This version uses the AVR Atmega328P microprocessor,

provided by Atmel (USA) on 16MHz processing speed. It includes fourteen (14) digital and six (6) analog terminals that can be used as inputs or outputs, depending on the structure of the application. The operating voltage of the board is 5V [4]. The Pro Mini version, unlike the other versions, does not have an included FTDI board and requires an external board to program it. The choice of this version was based on space limitation within the sphere and also on the at the lower power consumption due to lack of FTDI communication.

B. Bluetooth HC-06 module

Bluetooth is an industry standard for Wireless Personal Area Networks (WPAN). It is a wireless short-range telecommunication technology, which can transmit microwave signals to digital devices [8]. The construction of the Balanced Sphere uses the version HC-06, which works exclusively as a slave module. The main features that led to the choice of this version are the installed subsystems and its operating features. The main subsystems are the antenna with a 2.4GHz wireless transceiver, the 8Mbit memory and the EDR. The operating voltage is within the range of 3.1 to 4.2 volts, while the power consumption belongs to the range of 30 to 40mA in the coupling mode and at 8mA in communication mode [10].

C. L293B H-Bridge

The H-Bridge is an electrical circuit that allows voltage to be applied to a load on forwards or backwards polarization, depending on the active pair of switches. According to Figure 1, the switches combinations that give the forwards or backwards polarization are the following S1-S4 for forwards polarization and S2-S3 for backwards. It is important not to implement one of the following combinations: S1-S3 or S2-S4. These combinations will cause engine disruption. All remaining combinations either do not cause any movement or cause a short circuit [11]. The applications bridge, is the SGS Thomson Microelectronics brand, double-type bridge L293B. This bridge has a power distribution capability up to 1A in each channel, also each channel is controlled by TTL (Transistor-Transistor Logic) logic to achieve high reliability in its operation

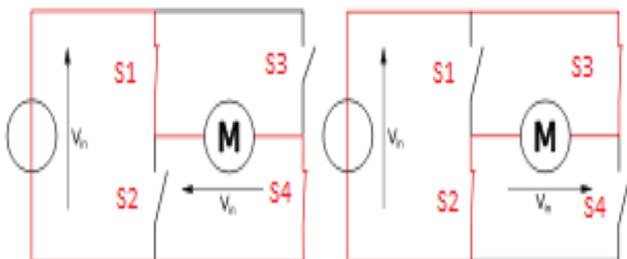


Fig. 1. Main operation states of an H-Bridge

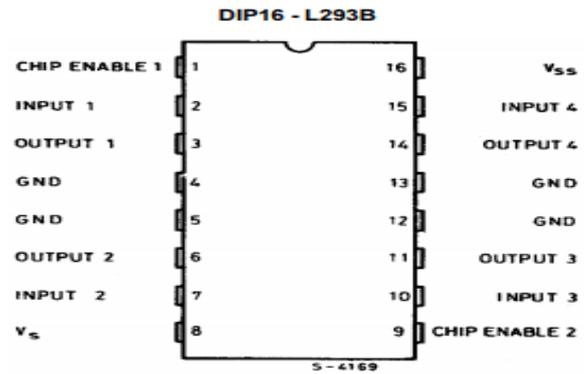


Fig. 2. L293B Schematic Diagram

As shown in Fig. 2, the bridge consists of 16 pins, with four of them acting as ground pins. Each of the two channels consists of 5 pins. The first one acts as an enable pin. The remaining 4 are divided into pairs of control inputs and outputs. The last two pins are the voltage provider pins of the logic circuit and the motors [12]. The selection of this bridge was based on the need of independence between the two DC motors and two-way movement control.

D. MG90s ServoMotor

A servomotor is a linear or circular actuator, which offers the ability of controlling position with great precision as well as speed and acceleration. A servomotor is based on a DC motor suitable for each application and a high sensitivity sensor to achieve position control. A controller also is required to process the feedback sensor elements by performing the required actions [13]. Two MG90s servomotors modified to operate as DC motors were used in the application. The main features that led to this choice were the small dimensions of the servomotors, along with their high torque attribute. The high torque is mainly present due to the aluminum transmission gear installed on each of them [14]. The value of torque is a determining factor as the construction is powered by batteries, which greatly increase the overall project weight.

E. 9 Volt PP3 battery

The Balanced Sphere power supply is available by using a pair of batteries or electric accumulators, in a parallel connection. These are the PP3 9Volt batteries with a nominal capacity of 600 mA. All PP3 batteries are prismatic orthogonal with curved edges and polarizing connections on top. Batteries are divided into two major categories: disposable and rechargeable batteries. The most popular types of disposable batteries are carbon-zinc, alkaline and less popular Li-FeS2 lithium batteries [15]. Disposable non-rechargeable battery options were rejected, as the goal was to implement the application in the long run and are also considered to be unprofitable.

The main types of rechargeable batteries are Nickel-Cadmium (NiCD), NiMH and Lithium Polymer (LiPO) [15]. The implementation of this application was based on the saving of space and the optimal balance of materials inside the sphere. The two main options were LiPo and NiMH batteries. LiPo batteries were the best option in terms of energy, but they had a major drawback on their high cost [16]. NiMH technology was selected, which was a much more economical choice and satisfactory performance [17].

IV. FUNCTIONAL DESIGN OF THE PROJECT

A. Operating Scenario

The idea of constructing this particular application comes from the fact that the ball is unstable because of its shape. Achieving stability requires a weight distribution that gives absolute balance. Also, high precision movements of the motors are required. This particularity led to the modification of the servo-motors to DC motors to achieve accuracy. Sphere Balance Control is achieved using an Arduino microcontroller, which version is the Pro Mini. Construction control is implemented by an application designed on the free Android application design platform, MIT App Inventor II using Bluetooth. Using the Android application, the drive commands are given to each engine, while the ability to remotely control engine speed is also provided. Beyond basic motion control parameters, it offers a few more possibilities. These capabilities, on the one hand, control the wireless interface by giving the choice of automated or manual connection and, on the other hand, the ability to select the display language of messages and functions within the application.

B. Coding Environment

The microcontroller is programmed via the help of the Arduino IDE (Integrated Development Environment), a JAVA-based application with C and C++ editor and compiler. In particular, coding is done with the Wiring Programming Language (C, C++). IDE uses GNU toolchain and AVR Libc tools to provide compilation of programs from C, C++ to appropriate AVR machine language commands and the avrdude program tool for sending the executable program to Arduino's Flash memory [9].

C. Wireless Controller Application

The sphere balance control is achieved by using a device that supports Android operating system. For the needs of the application, an android mobile phone device that supports Bluetooth technology was used. The control application has been implemented using the free MIT App Inventor II design platform which provide the ability of sharing the application on any device that has the above features. This platform uses Block-type programming, without requiring specialized programming languages to construct applications [18].

1) Controller's Application Analysis

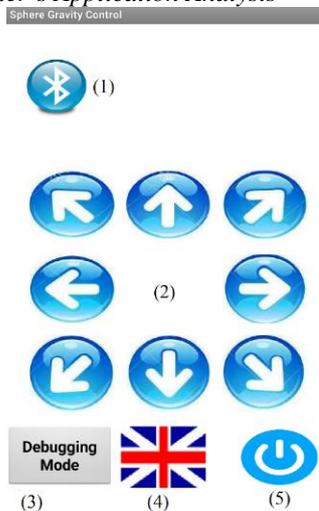


Fig. 3. Controller's Application (1)

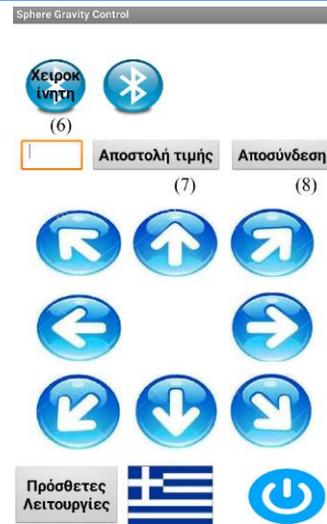


Fig. 4. Controller's Application (2)



Fig. 5. Controller's Application (3)

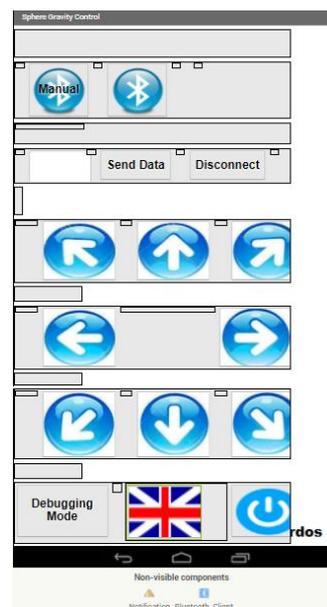


Fig. 6. Application's Structure

The above figures (fig. 3-6) shows the application desktop and structure form. Find below the references in brackets, numbers from 1 till 8 appearing on the screens.

The fig. 3, show the following touch screen buttons: Eight motion control buttons (reference No.2 on the figure above), a remote connection, without the need for device address selection (1), a management function display key (3), a language selector key (4) and finally the application exit button (5).

The motion control keys (2), when the device is offline, do not perform any function but operate only if there is an online connection. The language selector key (4) and also the application's exit button (5), keep functioning without the need of an active connection to the controlled construction. The language selector key performs between the two options of Greek and English operation languages. When the language changes, all popup messages and button names instantly adjust to the selected language.

Fig. 4, shows in Greek language version, both the application's main functions and the management features. Management features appear right after the use of the management button (3).

There are 3 main management features/ functions: manually selecting a controlled device address (6), wireless motor speed control (7) and disconnection of the controlled device (8). The existence of the device auto-connect button (1) has been implemented to make the Balanced Sphere operation smoother because the selection of the Bluetooth device address is avoided. The address of the device consists of symbols and letters, as shown in figure 3. The application does not block interfaces with other Bluetooth devices other than HC-05 or HC-06. However, if a connection to a different device is selected, the application will not work properly.

Fig. 5 shows the screenshot of the controller's application when it is successfully connected to the controlled device. Changes that occur are the display of "Bluetooth Connected" underneath the auto-login icon, as well as a Bluetooth Status pop-up window. It also shows the Bluetooth address that the controller application is connected to.

Finally, the activation of the motion control keys occurs. When the device is offline, if the management keys are pressed, they do not send any command. In such cases, these features indicate that there is no active Bluetooth connection via a popup window message.

Fig. 6 shows the application's screen arrangement, designed by the MIT App Inventor II platform. The application screen arrangement is based on the use of partition frames for each segment. The dimensions of the frames are adjusted to meet the screen dimensions of the monitor. The display diameter of the device (mobile phone used) is 5.5 inches. If the application is used from another device with a smaller diameter, the screen can be scrolled. Otherwise, if it is used from a device with a larger screen, large gaps will appear on the screen.

2) Controller Application Code

The application code is implemented by block-diagram programming. Each key, panel, Bluetooth Client and Notification use block control functions, making programming very easy.

Each command that causes changes to the device being controlled sends a number that is included in a byte variable. The variable has the size of one-byte number. One byte contains all numbers from 0 to 255. In the application, the numbers from 5 to 45, with a five-digit step, are used to control movements and the numbers from 150 to 250 are used to define the speed of the motors. The code follows:

Fig. 7. Operating screen block

Fig. 8. Language selector Block

Fig. 9. Close App Block

Once the application is started, the Bluetooth status of the monitor is checked. If the communication feature is not enabled on the controller's device, a popup warning message appears. Also, when booted, the application sets as invisible:

- the manual connection button and,
- the frame containing the additional functions.

Using the Debugging Mode button, the invisible elements appear again.

a) Application's display language selection

Fig. 8 shows the application's display language selection mode. This option operates with the help of a wide variable named "Language". The "Language" variable has two possible values, 0 and 1. When the value is zero, which is

the default value of the variable, the language is set to English. Also, the button icon is set to the English flag. Correspondingly, when the value is one, the language is Greek. Language selection operates depending on the time the user presses the button continuously. If this time is of short duration, then the variable gets 0, defining the English display. If the time is of long duration, the Greek display language is defined respectively. Finally, the Figure 9, shows the shutdown button function that closes the application.

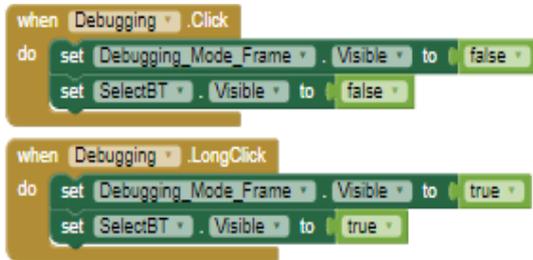


Fig. 10. Debugging Block

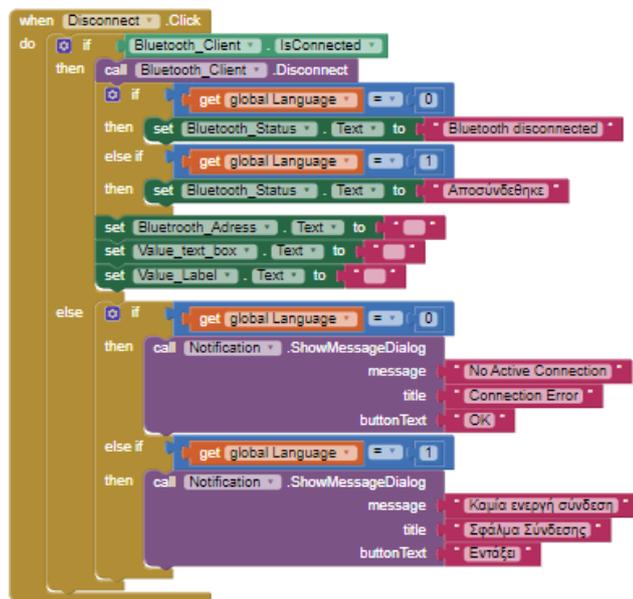


Fig. 11. Disconnect function Block

b) Additional Functions Mode

Fig. 10 shows the operation of the additional functions key. The additional functions operation concerns:

- the display status of the "Debugging_Frame", which contains the motor speed control and,
- disconnection functions of the tested device.

This button is similar to language selection and is based on key press time. If the time is of short duration, the frames are hidden, and if it is of long duration, they reappear respectively.

Fig. 11 shows the function of the disconnect button. This button, once pressed by the user, checks if there is an active connection. If there is a connection, it calls the device to disconnect and then checks for the active display language. Depending on the language status, it displays the disconnect message.

Then, it clears both the device address status and the motor speed control value. If there is no active connection, an alert window pops up informing the user that there is no active connection.

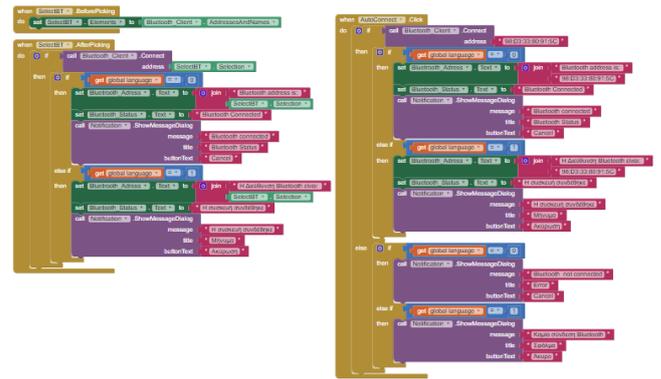


Fig. 12. Connection Functions block

c) Manual and Automatic connection functions

To the left side of the fig. 12 is the manual connection key and to the right side the auto-connect key. Manually selection works as a key, but actually is a list picking feature. When pressed, it displays the available for connection Bluetooth devices. Specifically, before choosing a connection, the list selection function defines as its variable the option specified in the list displayed. When the option is selected, it calls the wireless connection to the selected address.

Afterwards, it displays in a text field next to the selection key, the address of the selected device. A popup window with the connection display sequence appears as shown in fig. 12.

The language of display is controlled by the value of "Language" variable, affecting all pop-up messages. The auto-login button process is similar to the manual. The main difference and disadvantage of automatic connection is the default connection address.

This feature creates incompatibility problems in case of changing Bluetooth equipment or use with another project construction. Nevertheless, the creation of this button has as a key advantage: the faster connection of the two devices.

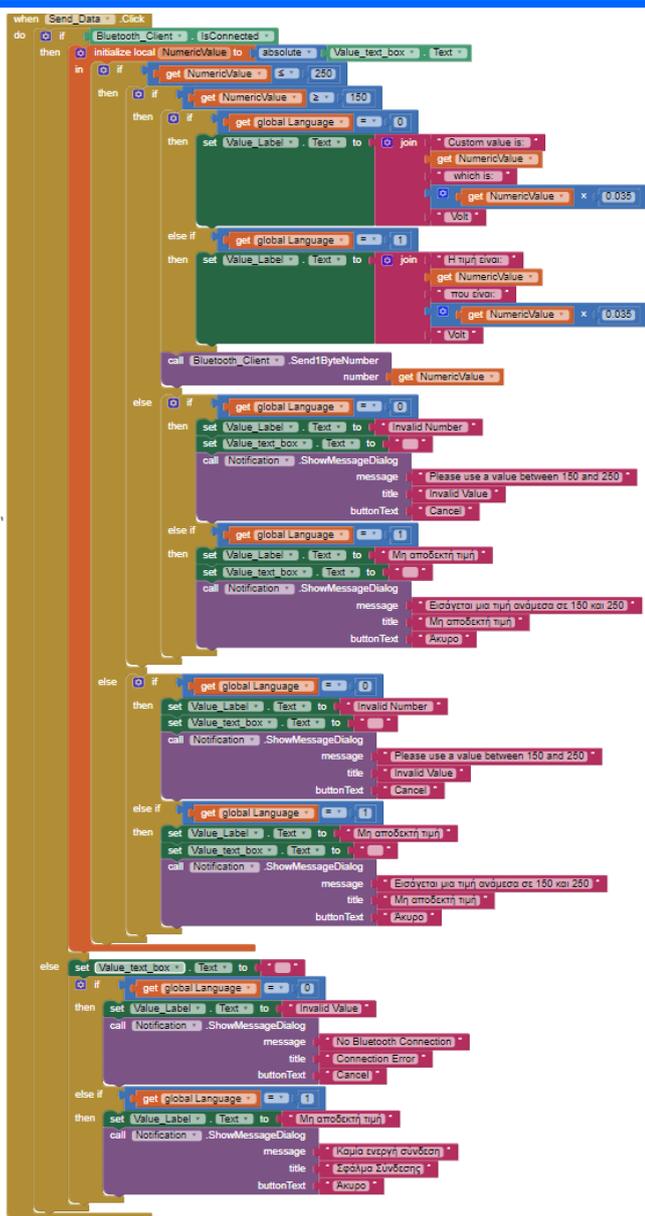


Fig. 13. Remote motor speed control

d) Remote motor speed control

Fig. 13 shows the remote motor speed control function. This function works by pressing a button called "Send_Data" and a text field named "Value_text_box". To have a proper outcome, this function needs:

- a value that is within the set limits 150-250 and,
- an active connection.

The desired value is set through the text field and is sent by pressing the Send Data key. Specifically, as soon as the "Send_Data" key is pressed, the connection is checked with the controlled device.

If the link is successful, it checks the content of the text field and stores it in the local variable "NumericValue". In order to achieve storage, it is necessary to control the contents in absolute value. If the text field is empty, the process stops.

This control is achieved only when there is text of at least one character within the field. Given the existence of a

single character within the field, control of the variable continues. If the variable is less than or equal to the maximum limit 250, it is rechecked.

This test examines whether the value is at the same time greater than or equal to the minimum limit 150. If both checks are met, the desired value is sent in the form of a number to the controlled device, via Bluetooth and the speed control of the motors is achieved.

Along with active speed control, there are also processes for displaying messages and notifications regarding the imported value. Images and messages have the ability to appear in both available languages.

The messages displayed when the input value is accepted relate to the size of the value in subdivisions of one byte and also in subdivision of the 9Volt scale.

For example, if the selected language is English and the imported value is 210, the displayed message will be "The value is: 210 that is 7.35V". In order to find the voltage, $V_{Motor} = 9/255 = 0.035V$ has been used, where 9V is the maximum input voltage for motors and 255 one-byte subdivisions.

If the input value does not belong within the range of limits set ($150 \leq \text{Numeric_Value} \leq 250$) then the notification window pops up and informs the user that the value entered is not acceptable, while indicating the acceptable limits for this function.

At the same time as the notification window appears, a message below the input field indicates that the value is not acceptable and the control text field is reset to its original state (blank field).

This initial state is achieved by defining its content as a blank text. Finally, if the Data Send button is pressed while there is no active connection, an alert window is displayed. The message informs the user that there is no active link, and the text box of the variable is also initialized.

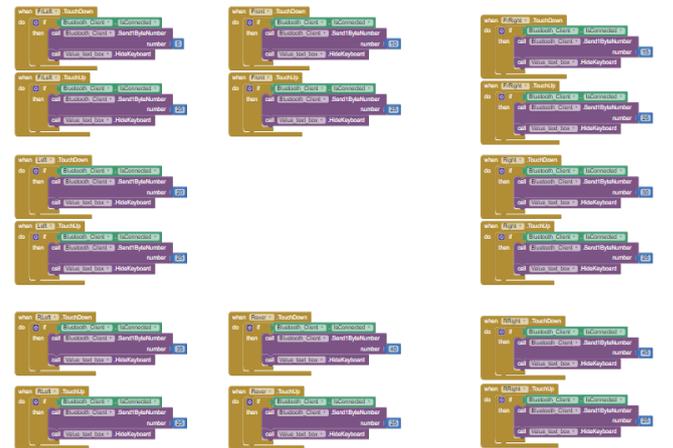


Fig. 14. Movement Control Keys Blocks

e) Movement Control Keys

Fig. 14 shows the operating code of the movement control keys. Movement control is achieved by sending a predetermined value for each possible move through the Bluetooth device. All moves carry the same code while changing only a shipping value. The movement control values start from 5 to 45. The operating mode of the control keys is based on the time of pressing the key. As long as a

button is pressed, if there is an active connection, it continuously sends a byte value that matches the movement. For example, in the front-direction function, as long as the key is pressed, the value 10 is continuously sent to the controlled device. When the key is not pressed anymore, the second part of the procedure is triggered. The value 25 is sent and both motors stop.

V. CONCLUSION

In this project, a wireless controlled robotic sphere has been completed. The purpose of the construction is to remotely control the balance of the sphere using two internal DC motors. Important features of the implementation of the work is the use of non-specialized parts and the ability to apply the construction in educational process. The construction consists of a small number of components. All the components used in the construction of the sphere are retail parts, without requiring special 3D component printing. An important feature is also the low cost of construction. The abovementioned aspects make the construction suitable for educational purposes.

The use of an Arduino family microcontroller has brought many benefits, as its coding is based on programming language C, it is open source and at the same time, these microcontrollers are a cost-effective option. An upgrade that would improve the feedback speed and reliability is the use of the Raspberry Pi III microprocessor. This microprocessor has higher computing power, increased memory and also a WiFi connection ability [19].

However, this upgrade requires specialized programming skills, while increasing the complexity and energy demands of construction. These requirements can be met by simultaneously upgrading the power supply from NiMH batteries to Lipo batteries. Nevertheless, this upgrade also increases the cost of manufacturing due to a specialized charger requirement and higher battery cost.

Bluetooth remote control technology is a reliable choice in wireless control. Its biggest disadvantage is that it is limited to apply at Android devices and also the limited range of operation. An alternative option is the control via Internet. The device that allows this upgrade is ESP8266 from Espressif Systems (China) [20] which gives the ability to control over any device that has a WiFi connection, uses the 802.11 b/g/n communication protocol and it's also connected to the Wi-Fi connection of the Balanced Sphere.

A downside to this upgrade is the need for a dedicated control application and the need to reprogram the Arduino microcontroller device in case of connection to a new WiFi address.

VI. ACKNOWLEDGEMENTS

Authors would like to acknowledge the University of West Attica postgraduate program of studies "MSc in Industrial Automation" for supporting this research project.

REFERENCES

- [1] <https://el.wikipedia.org/wiki/%CE%A1%CE%BF%CE%BC%CF%80%CF%8C%CF%84>, access 24/11/2017
- [2] <http://roboticsinformations.blogspot.gr/> access 24/11/2017
- [3] <https://el.wikipedia.org/wiki/%CE%9C%CE%B9%CE%BA%CF%81%CE%BF%CE%B5%CE%BB%CE%B5%CE%B3%CE%BA%CF%84%CE%AE%CF%82>, access 2/12/2017
- [4] <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ProMini16MHzv1.pdf>, access 16/7/2017
- [5] https://en.wikipedia.org/wiki/Radio-controlled_car, access 2/12/2017
- [6] <https://electronics.howstuffworks.com/remote-control1.htm>, access 2/12/2017
- [7] <https://electronics.howstuffworks.com/remote-control3.htm>, access 3/12/2017
- [8] <https://el.wikipedia.org/wiki/Bluetooth>, access 15/7/2017
- [9] <https://en.wikipedia.org/wiki/Arduino>, access 16/7/2017
- [10] <http://silabs.org.ua/bc4/hc06.pdf>, access 16/7/2017
- [11] https://en.wikipedia.org/wiki/H_bridge#Operation, access 18/7/2017
- [12] <http://www.st.com/content/ccc/resource/technical/document/datasheet/3f/2a/59/15/61/43/45/af/CD00000058.pdf/files/CD00000058.pdf/jcr:content/translations/en.CD00000058.pdf>, access 28/7/2017
- [13] <https://en.wikipedia.org/wiki/Servomotor>, access 28/7/2017
- [14] <http://www.towerpro.com.tw/product/mg90s-3/>, access 30/7/2017
- [15] https://en.wikipedia.org/wiki/Nine-volt_battery, access 2/8/2017
- [16] <http://www.radio-electronics.com/info/power-management/battery-technology/lithium-ion-battery-advantages-disadvantages.php>, access 6/10/2017
- [17] http://batteryuniversity.com/learn/article/nickel_based_batteries, access 5/10/2017
- [18] <http://appinventor.mit.edu/explore/about-us.html>, access 17/10/2017
- [19] https://en.wikipedia.org/wiki/Raspberry_Pi#Model_B, access 12/12/2017
- [20] https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf, access 9/12/2017