

Implementation of an 2-wheel Educational Platform for STEM Applications

V.Mavrovounioti

*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 aim of this project lies in the construction of an educational two-wheel robot for preschool children. This construction will allow children to give commands to said robot via a screen and keyboards, as well as to interact with it through sound and light sensors. The objective of this construction is to introduce children to the basic ideas of mechatronics without reading skills as a prerequisite, but instead through images, light and auditory indicators. In the following chapters, the theoretical framework will be presented first, by analyzing the nature of mechatronics and its dimensions, as well as its role in education. The theoretical framework will be concluded with the development of mechatronics and the positive consequences of its use in education, which is the overarching goal of this study.

Keywords—*Robotics, Preschool Devices, Sensors, Controllers, programming*

I. INTRODUCTION

A. Mechatronics

Technology is defined as the practical application of the ideas emerging from the positive sciences as well as all the knowledge related to its application. It was mainly due to the need of man to adapt to the environment and control it, as it evolved and it began to apply to other disciplines. Nowadays, technology is not only used to meet people's living needs but also to communicate, entertain and educate. Technology is now changing rapidly. One day's innovation can be discarded the following day. At the same pace, the technological knowledge grows, the technology industries are also growing. An industry that has distinguished itself in recent years is Mechatronics. The word mechatronics originally appeared in the 1970s in Japan. Mechatronics is the composition of computer science with engineering and electrical engineering. The result of mechatronics is mixed systems with electrical, electronic and mechanical parts operated by a control system, automatic control. The building blocks of this technology are five [10]. The first element is the modeling of physical systems, in this process you are studying the system to be tested and then created "objects" that can copy its behavior. With the modeling process, we can draw

conclusions about the studied system more easily and quickly. The second element of mechatronics is the application of signal and system theory. Signals carry information and commands while systems produce or process information and execute commands. The third and equally important element of mechatronics is the application of computer technology as well as logic systems. This piece is owned by Microprocessors. Microprocessors, or microcontrollers, are the brain of the system, collect signals, process them and decide what the commands will be. The fourth element is software. The software is the set of programs and processes running on a computer, a microprocessor. The software is the "logic" the brain uses to draw conclusions. The fifth element is Sensors and Activators. Sensors [6] are components that can understand environmental conditions. A sensor can "read" one physical size and then convert it into a measurable size, such as electrical voltage, to something that the microprocessor can understand. Instead, the activators get a command from the microprocessor and activate the engineer to execute the command.

B. Main Purpose

The main purpose of this dissertation is to build a floor robot. This robot has a dual role, educational and entertaining since it is a game for preschool children to introduce them to concepts such as programming, automation, and sensors. As the child interacts with the robot he will see that the commands he gives to the robot and the conditions in his environment are combined and conducts a result. This ability gives the child a sense of security in terms of technology, in addition, it helps to develop his logic and his combinational thinking. As the age at which the construction is intended is preschool and there is no reading ability, a set of audio and light indicators as well as symbols for child and robot communication will be used. The robot was not designed to replace the teacher but to offer him a new tool.

II. SYSTEM

A. Hardware

For the construction the following materials were used: 1 Arduino Mega [2], 1 LCD (16X2) [3], robotic platform [1], 4 buttons with integrated led, 2 round buttons, 1 push button [5], 1 sound sensor, 1 LDR, 1

speaker, 2 DC motors [4], 1 driver (L293D), 1 potentiometer, 12 resistors, 2 electronic boards, 2 small breadboard, 1 drive ball, 1 12 volt battery and a 2-way switch.

In addition to the basic components for the construction were used a Dremel tool, electronic soldering iron, heat glue gun, crystalline silicone, multimeter, screws, spacers, heat shrinkable insulating cables, insulating tape, cable ties, and various hand tools.

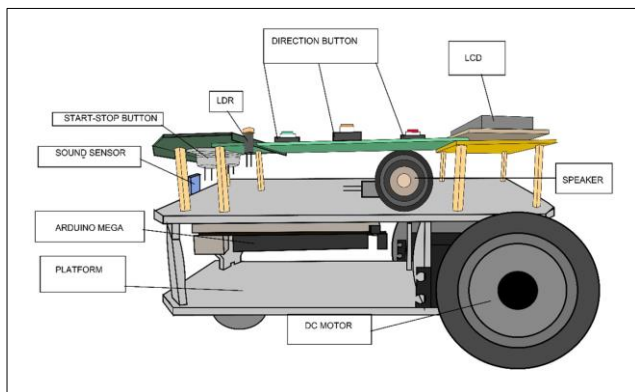


Fig. 1. Profile plan of the construction

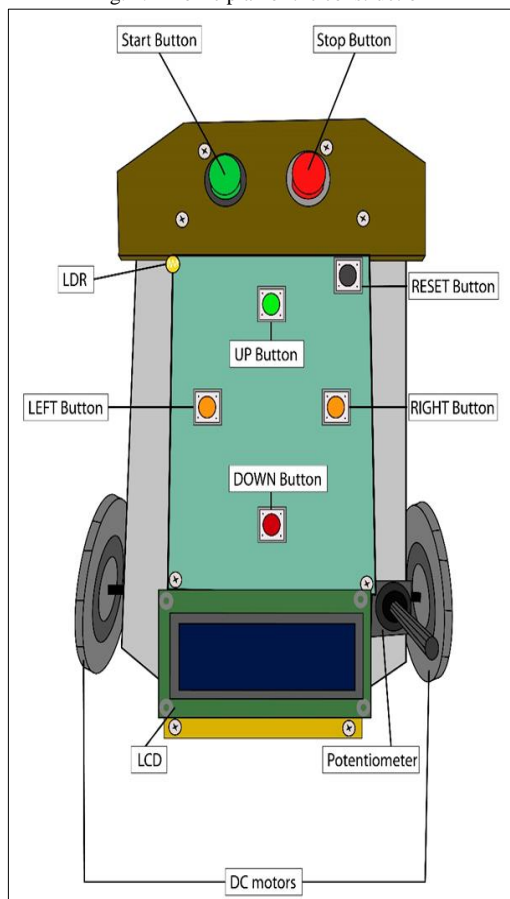


Fig. 2. Fig. 1. Reflected plan of the construction

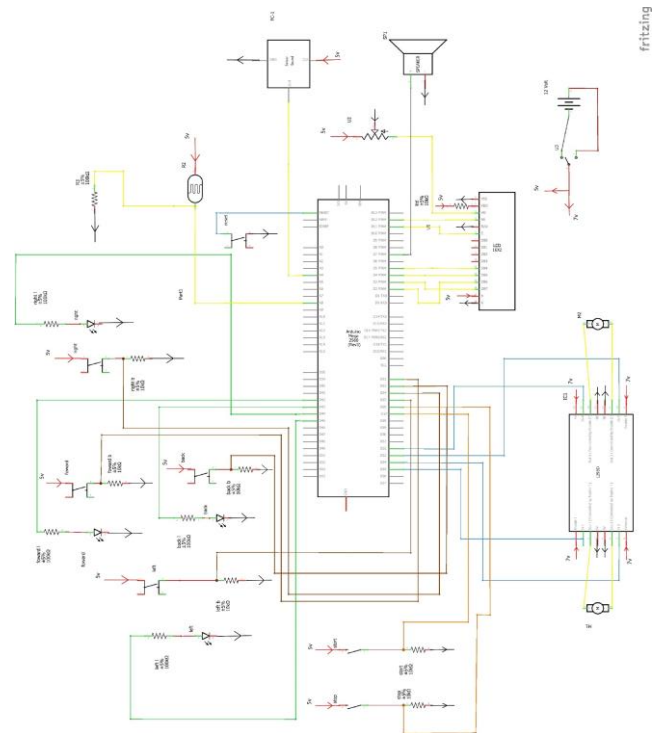


Fig. 3. Electronic plan

B. Scenario

The dissertation has 4 programs installed. The user can navigate between the programs through a start menu, with the use of the buttons, when the user has picked the program he wants, he presses the button Play to select it. The return to the start menu, it is made by pressing the Reset button. In all the programs, the user enters the movement commands (up, down, right, left) that are shown on the LCD screen. Once the commands are entered, then by pressing the start button, the car will begin to execute the commands and navigate to the specific directions given by the user. With the stop button, the user can stop the movement of the car or, if it is immobile, he can delete one command per click.

In the first program, the user enters the movement commands (up, down, right, left) and then by pressing Play they are executed for one second, during the execution of the movement, one note is produced.

In the second program, the user follows the same procedure with a slight difference. Once the start button is pressed, the car will start to initialize the user's input and in order to move to the next command, a strong sound must be created. If the sound frequency is stronger than the sounds sensors given frequency, then it will proceed to next move. This sound can also be created by clapping.

The third program is about moving with light. After inserting the commands and pressing the start button, the commands will follow normally as in the first program but only when the light sensor receives enough brightness. In the event that the value the sensor reads drops below the desired value then the car stops immediately and if it returns to the previous value then it continues from the command that was

left. We can reach the desired brightest value by using a flashlight.

The last program follows the same logic as the third but with the difference that the car moves in the dark and when the light sensor reads a value greater than the desired value it stops immediately.

III. SOFTWARE

At the beginning of the code, the various variables used in the code are initialized and all the libraries that are needed are imported.

In the following pictures, the flowchart of the loop function is demonstrated and after the other function of the program are explained.

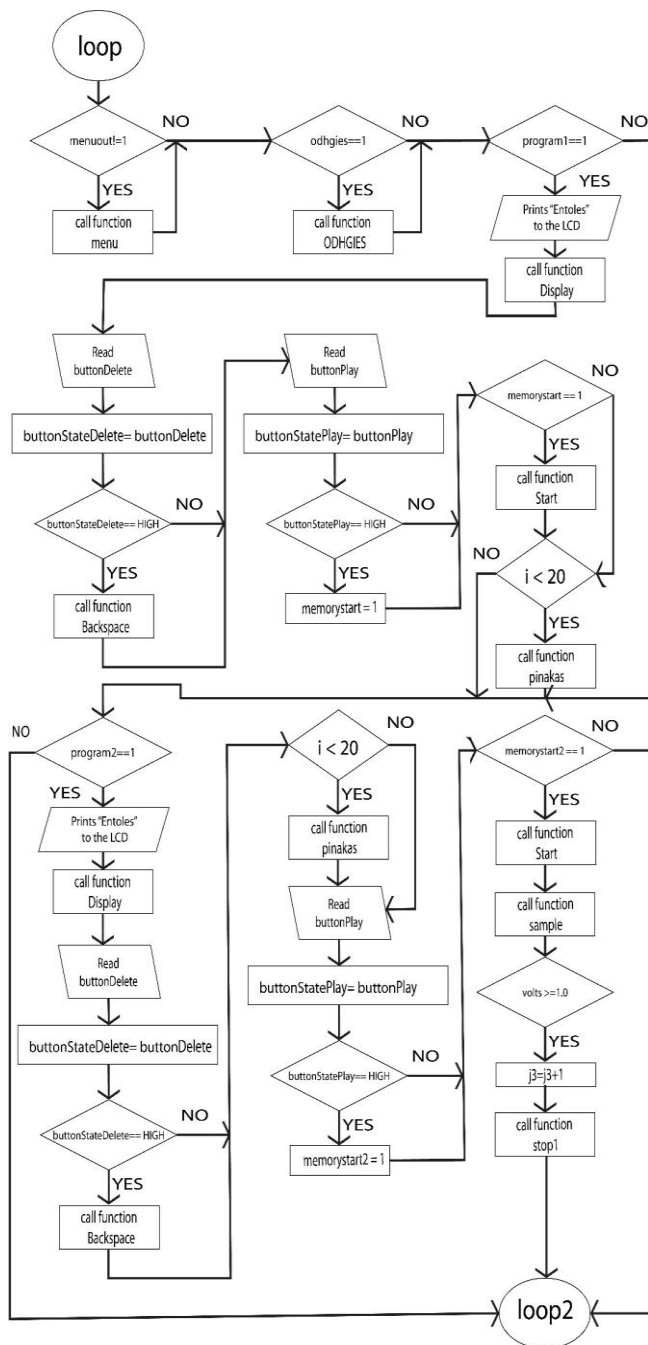


Fig. 2. Flowchart: Loop ,part 1/3

Before the void setup function some custom characters [7] are created, so we could to demonstrate them on an LCD display later.

The setup function runs once, at the start of the code, and is used to initialize and define the variables. The loop function runs repeatedly and the master code is written there.

Through the Loop function, the Start menu is initially called, and then, according to what is selected, the basic programs are called.

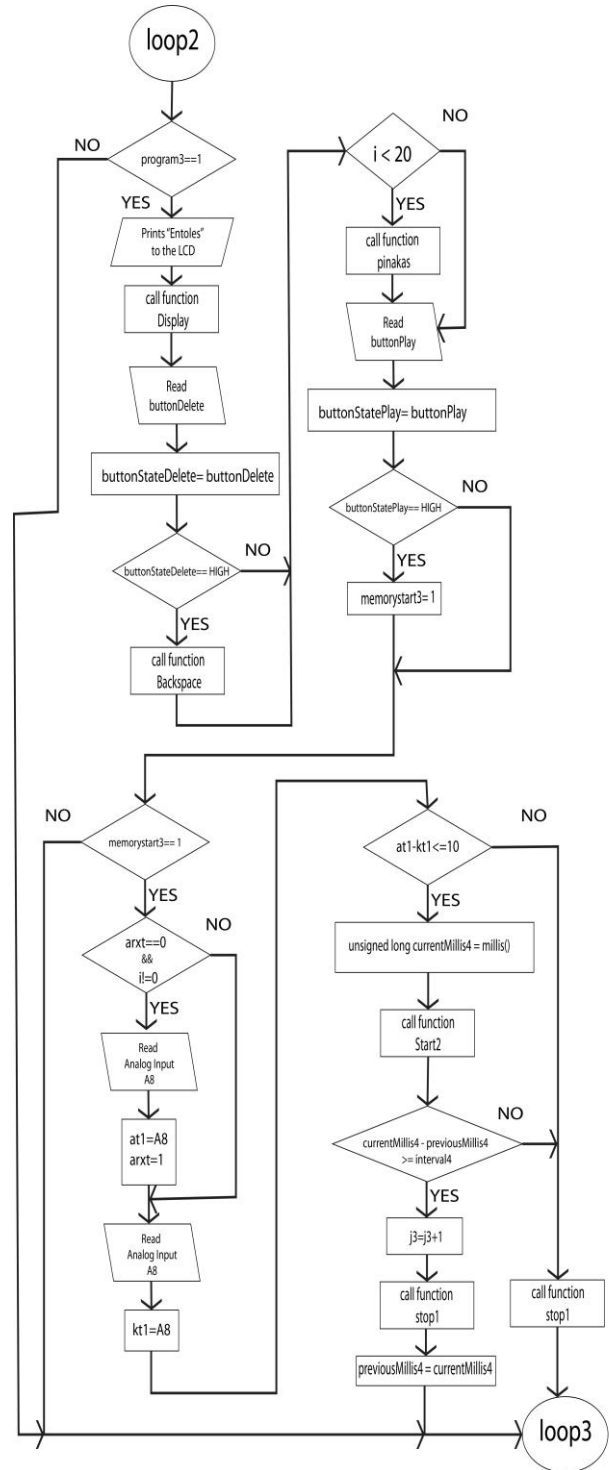


Fig. 3. Flowchart: Loop ,part 2/3

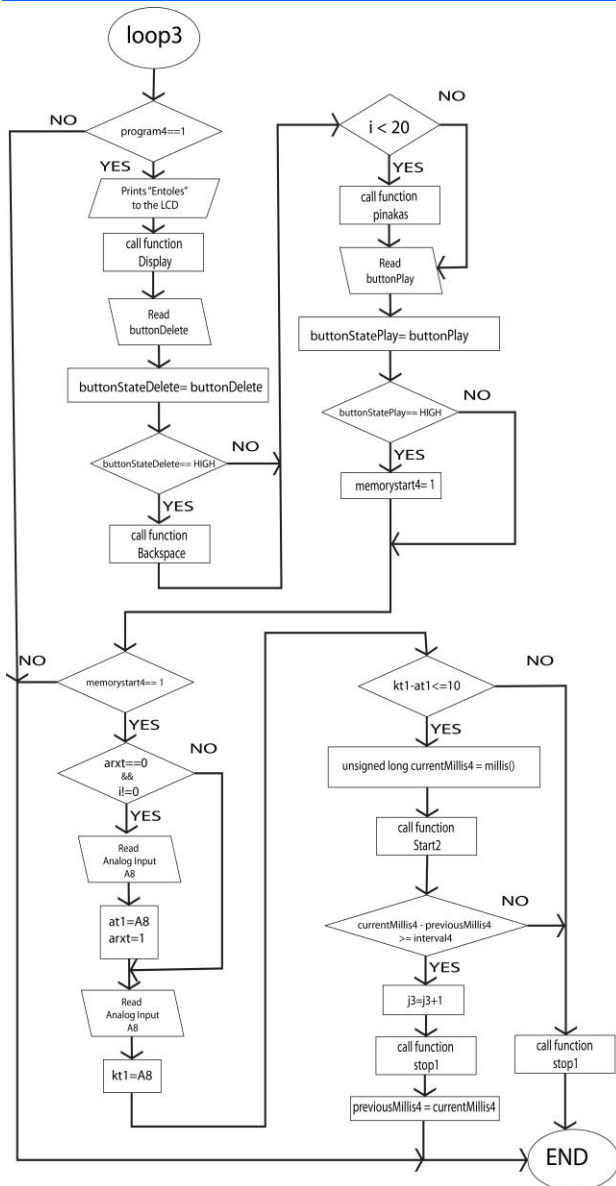


Fig. 4. Flowchart: Loop ,part 2/3

- ODHGIES

During the ODHGIES function, the user navigates between the tables Odhgies1, Odhgies2 Odhgies3, and Odhgies4 by using the move keys. Inside those tables there are written Instructions for each program.

- pinakas

The pinakas function, reads the motion keys and stores the motion commands in the kinhseis[i] table. Each time a key is pressed, a note is produced.

- Display

In the Display function, the movements stored in the kinhseis[i] table are displayed on the LCD screen in character-like format.

- Start

In the Start function, the movements stored in the kinhseis[i] table are read, and the mprosta(forward), pisw(backwards), deksia(right), aristera(left), and stop1 functions are called.

- mprosta

During the mprosta function, the motors are commanded to move forward and turn on the corresponding led for 1sec as well as produce a sound for 0.5sec.

- pisw

During the pisw function, the motors are commanded to move backwards and turn on the corresponding led for 1sec as well as produce a sound for 0.5sec.

- deksia

During the deksia function, the engines are commanded to turn right and turn on the corresponding led for 1sec as well as generate a sound for 0.5sec.

- aristera

During the aristera function, the engines are commanded to turn left and turn on the corresponding led for 1sec as well as generate a sound for 0.5sec.

- stop1

During stop1 function, all the engines stop and all the leds are turned off.

- Start2

In the Start2 function, the movements stored in the kinhseis [i] table are read, and the mprosta2, pisw2, deksia2, aristera2 and stop1 functions are called.

- mprosta2

During the mprosta2 function, the motors are commanded to move forward and turn on the corresponding led.

- pisw2

During the pisw2 function, the motors are commanded to move back and turn on the corresponding led.

- deksia2

During the deksia2 function, the engines are commanded to turn right and turn on the corresponding led.

- aristera2

At the aristera2 function, the engines are commanded to turn left and turn on the corresponding led.

- sample

The sample function calculates the amplitude pulse of a sound wave and converts it into volt.[12]

- menu

The menu function reads the motion buttons and, depending on the selection made, shows the title of each program along with symbols describing it. When the Play button is pressed, the program that is displayed at that time on the screen is selected.

IV. CONCLUSION

The software of Arduino has two basic functions the first one eyes wide shut up and the second one is void Loop. The word void in front of the name of the function means that they don't return any value at the end of them. Set-up function runs only once upon the controller stands that's why in that function we said the values of the program, on the contrary the function Loop it is run continuously, that's where we write the main program.

Arduino is a simple processor and can run one program at a time, unlike other more powerful processors. That is why we need to have a more cautious approach when we want to achieve multiplexing [8].

Each program was implemented separately and then integrated together, but it was observed that the programs affected one another. Various techniques have been used to avoid such problems. The basic specification was the use of multiple function, so the loop function has fewer lines of code, the commands are grouped, and the function's time is more specific.

Something else that caused a problem in joining programs was the delay command. The delay when called pauses all the processes, "freezes" Arduino in the state that is at that moment. To deal with this, a set of commands, especially the command millis(), was used.

As the navigation between programs is done with a single menu, the Reset pin was used to return to the home screen. Using this pin and not using code, we

confirm that all variables have returned to their original value.

Finally, because Arduino supplies up to 5 volts of current, while a multiplexing system needs much more voltage, a solution had to be found for powering the construction. The main problem in the power supply was the engines, which to operate at a satisfactory speed need a higher voltage than the one Arduino can provide. Problems to the other components as well, such as buttons that had a long response time. For this reason, a process has been carried out in the battery union with the switch so that from the 12 volt of the battery the 5 go to Arduino, while the remaining 7 volts are directed to the motors.

V. ACKNOWLEDGEMENTS

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

VI. REFERENCES

- [1] <https://grobotronics.com/shadow-chassis.html>
- [2] <https://el.wikipedia.org/wiki/Arduino>
- [3] https://en.wikipedia.org/wiki/Liquid-crystal_display
- [4] Electricity 4: AC/DC Motors, Controls, and Maintenance from Jeffrey J. Keljik
- [5] <https://el.wikipedia.org/wiki/sensor>
- [6] <https://www.arduino.cc/en/Reference/LiquidCrystalCreateChar>
- [7] <http://www.instructables.com/id/Multi-task-Your-Arduino/>
- [8] <https://grobotronics.com/rocker-switch-on-off-red.html>
- [9] <http://www.ti.com/lit/ds/symlink/1293.pdf>
- [10] <https://programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>