Media Engineering and Technology Faculty
German University in Cairo

Mini-Ki Building a Robot to Teach Programming

Bachelor Thesis

Author: Khaled Mohamed Ahmed Hassan
Supervisor: Dr. Nada Sharaf
Submission Date: 26 May, 2019

Media Engineering and Technology Faculty
German University in Cairo

Mini-Ki Building a Robot to Teach Programming

Bachelor Thesis

Author: Khaled Mohamed Ahmed Hassan
Supervisor: Dr. Nada Sharaf
Submission Date: 26 May, 2019

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowledgement has been made in the text to all other material used

Khaled Mohamed Ahmed Hassan
26 May, 2019

Acknowledgements

First of all, I would sincerely love to express my deepest gratitude and appreciation to my supervisor Dr. Nada Sharaf, who has been following up my work, helping and supporting me through the whole process. I would also like to thank her for her expert guidance, great effort, and endless motivation. Moreover, I would like to thank my brothers for their continuous support. Last but not least, I would like to express my love and gratitude to my mother who has been encouraging me all the time. Without her, I would not have reached where I am today. Finally, special thanks goes to my dearest Yara Dorgham.

Abstract

In recent years, Programming has become one of the most needed jobs in the market. Also, computational thinking has become a trend nowadays. This prompted and increased the demand of teaching children both programming and computational thinking at early age. This study aims to investigate how effectively young children can master foundational programming based on tangible robotic user interface. This was accomplished by designing and implementing a new educational robotic kit for young aged children (between 8 and 11), that uses sensors and motors, to teach them the basics of programming, computational thinking, and some hardware in a simple and fun way. Children can learn three main programming concepts (Sequential, Conditions, and Loops) while playing with this robotic kit. A sample of N=16 children participated in this research. The kit was tested using between-group experimental design with two sub-groups: experimental group and control group. The experimental group used the kit while control group was taught using traditional educational method (explaining on the board). The programming skills of the children were assessed before and after teaching them whether using the robotic kit or the traditional method. Pre- and post-test scores were compared to observe the learning gain of each group. Also, the engagement level and the system usability of each group were recorded. The results showed a significant difference between the two groups with a P-value$<$0.05 for the learning gain, the engagement level and the system usability. Accordingly, using educational robot is considered to be an effective method to teach young children basics of programming.

Contents

Chapter 1

Introduction

Nowadays as we reach the twenty first century, programming, a field of computer science, in which algorithms and codes are written for the development of software programs, is becoming a fundamental field. Programming has become a very important area that inspires and encourages many people to grow among the next generations. It has become an important skill in today's world. Computer scientists and programmers develop all the applications we use every day. Everything now depends basically on technology. Thus, it is essential for everyone to know how to write their own programs. It is also a good practice for children to learn these fundamental concepts in their early childhood [1].

Everyday children encounter digital technologies from video games to mobile phones to iPads and tablets, but most of the children do not know how these things really work [2]. New technologies are increasingly affecting the ways young children are playing and learning [2]. Although not every child needs to be a programmer, coding gives children the basic tools of problem solving and computational thinking. Computational thinking can also teach children how problems can be solved in a structured and efficient way by modeling the problem, breaking it down into smaller sub-problems, figuring out solutions for these problems and testing them out at the end. So, now the trend is to let the children study computer science not only for the purpose of implementing applications, games, websites or programs, but also to enhance their way of thinking and problem-solving skills at young age.

The work presented in this thesis mainly focuses on teaching children programming by building a mini robot. The robot uses some sensors, motors and actuators where the child can program this robot using TUI (Tangible User Interface) to do some functions. The aim of this robot is to teach children how to think and program. Robots have been chosen because they have proven to be an effective methodology for teaching children logical thinking and programming [3, 4]. This robotic kit uses a variety of advanced types of sensors that the children encounter in their daily life to do more complicated functions.

The aim of this study is to investigate the effect of using such robotic kits in teaching children programming concepts. The goal is to examine the result of using IOT (Internet of Things) systems that are embedded with sensors, software, electronics, and connectivity in helping the children to understand the fundamentals of programming at young age.

Teaching programming concepts and robotics allows children to learn important ideas that help them understand some of the daily objects that they deal with. Children can understand ideas from computer science and engineering when learning programming concepts and robotics [5]. Moreover, coding using robots shows the children that they can create with technology because the robot moves and behaves based on the commands and the instructions that the child gives it. In addition, robots offer tangible and playful way for children to engage with both the T of the technology and the E of the engineering in the early childhood STEM curricula. The child can directly view the impact of his or her programming commands on the actions of the robot [1, 4]. Using robotics, children engage joyfully with the process of learning how and why motors and sensors work [2].

This thesis is comprised of 7 chapters. Chapter 2 introduces the background which first discusses the concept of programming, then it discusses the Arduino board. Chapter 3 includes the prior work related to the field of teaching children programming. Chapter 4 describes the methodology used in this study. Chapter 5 tackles the approach used for the experimental design to test the robotic kit. Chapter 6 shows the tests conducted and the results. Chapter 7 concludes the thesis with an overall summary about the study, and contains recommended future work needed to improve the robot.

Chapter 2

Background

This chapter introduces the concept of programming and the advantages of learning coding at young age. It also discusses the Arduino board.

2.1 Coding and Programming Language

Coding is a basic language of the digital age. It is a new fancy word for computer programming where it is the process of creating a step by step instructions that the computer understands to design a program that works in gaming systems, cars, tablets, cell phones and even washing machines [6]. Coding is basically the computer language used to develop applications, websites and software. It is a way to instruct the computer to perform various tasks. Without this, we would have had nothing of the major technology on which we rely on, such as Facebook, smart phones, or browsers. Everything works with code. Simply, the code is what tells your computer what to do. Programs are written with programming languages.

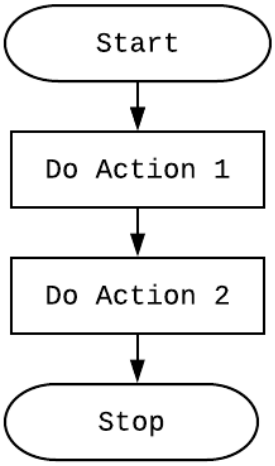
Programming language is a set of commands, instructions, and other syntax used to perform specific tasks [6]. It is just like any natural language like English, French, German or Spanish. It is a formal language that allows to do anything you want or to create any project you want creatively. Computer programs and computer software are all written with programming languages. Usually, for some of the commands, the programming language uses real words to make it easier for a human to read the language. Many programming languages use punctuation as a normal language. There are many programming languages exists, but there are some languages that are more popular and more used by programmers like Python, Java, C, C++, JavaScript, Go and some other languages.

2.2 Programming Concepts

There are three main concepts in programming that are fundamental and essential for anyone who is learning programming [7].

2.2.1 Sequential Programming

The concept of sequential algorithm in programming can be defined as the order of writing the set of instructions to perform a specific, or a desired function. The sequential programming means that the program operates sequentially, one instruction at a time, in a consecutive ordered execution of commands. Figure 2.1 shows a flowchart that describes the flow of a sequential program in addition to a simple sequential program using Java programming language.



(a) Sequential Programming Flowchart

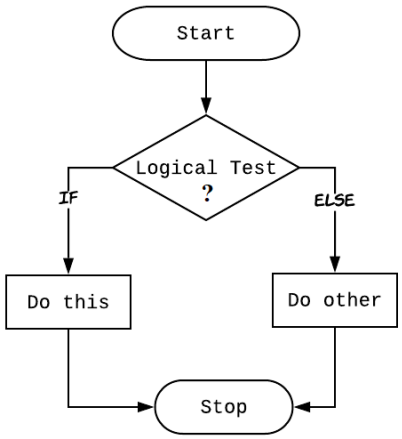
```java
public class learnCoding{

    public static void main(String []args){
        int x = 5;
        int y = 2;
        int result = x+y;
        System.out.println(result);
    }
}
```

(b) Simple Sequential Program Using Java That Print The Sum Of Two Numbers.

Figure 2.1: Sequential Programming

2.2.2 Conditional Programming

The second type is the conditional programming which evaluates the condition to true or false based on the program flow. It is as if the computer makes choices on which way to proceed. If the condition is satisfied, specific instructions will only get executed. If the condition is not satisfied, other instructions will get executed then. Figure 2.2 shows a flowchart that describes the flow of a conditional program in addition to a simple conditional program using Java programming language.



(a) Conditional Programming Flowchart

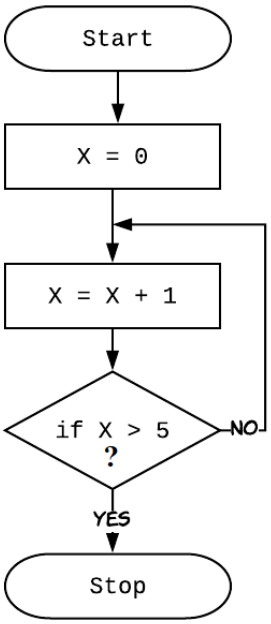
```java
public class learnCoding{

    public static void main(String []args){
        int x = 5;
        int y = 2;
        if(x>y)
            {
                System.out.println(x);
            }
        else
            {
                System.out.println(y);
            }
    }
}
```

(b) Simple Conditional Program Using Java That Prints The Larger Number Between Two Numbers.

Figure 2.2: Conditional Programming

2.2.3 Iteration Programming

The third type is the iteration programming, which is a very useful concept in programming. It is used to let the computer repeats the execution of a set of instructions for a specific number of times or until a condition is met. It is used instead of repeating the instructions several times. Figure 2.3 shows a flowchart that describes the flow of an iteration program in addition to a simple iteration program using Java programming language.



(a) Iteration Programming Flowchart

```java
public class learnCoding{

    public static void main(String []args){
        int x = 5;
        int y = 2;
        while(y>0)
            {
                System.out.println(x);
                y = y - 1;
            }
    }
}
```

(b) Simple Iteration Program Using Java That Prints a Number Twice

Figure 2.3: Iteration Programming

2.3 Advantages of Learning Coding at Young Age

Learning coding at the early age offers children experiences that integrate problem solving, thinking, and communication [8, 9]. These skills are considered to be the 21st century skills that are important and valuable for the future success of the children in the digital world. Therefore, according to research, 90% of the parents in the U.S. want their children to learn computer programming as it will be crucial for many jobs in the future. Coding helps kids to better understand and control their world. Actually, anybody can learn how to code [10]. Additionally, there are some other reasons why coding at early age is encouraged [8, 10, 11, 12]:

- Coding Is Fun and satisfying - Considering the fact that the children create a small application or a game on their mobile phones or their computers, they will enjoy their time and have a lot of fun.
- Coding enhances problem-solving skills - Learning to program helps the children to develop and enhance their critical thinking skills, and strengthen their problem-solving skills. Because when children code to let the computer do a specific task, they breakdown the problems into smaller sub-problems and do some connections between these smaller sub-problems. This skill is needed in our everyday's life not in coding only.
- Coding encourages and fosters creativity - Given a problem, there are more than one solution/way to solve this problem. Every child can solve the same problem in a different way, but some answers may be not efficient as the others. So, creativity takes place.

- Youngsters learn new skills faster - kids learn new things without much effort, and with less time than older people.
- Coding is extensively used and is in high demand - Nowadays, we rely on technology and computer science in our daily lives. Cars, mobiles, tablets, and TVs depends mainly on technology and coding, so when children learn coding, they will understand the world around them easily.

2.4 Arduino

The Arduino[1]is a very powerful, single board micro-controller, with a wide range of applications, and interactive controls [13, 14]. Arduino consists of a physical board and an IDE (Integrated Development Environment), which is written in Java programming language, used to type in and upload computer programs to the physical board, see figure 2.4. It is intended for electronics developers, and engineers [15]. It is an open source tool used for building electronics projects. It has an easy design for interchangeable modules to be added flexibly on the board itself. The main functionality of the Arduino is to be able to control or sense changes in the physical environments. This can be done using a basic concept of receiving inputs from various sources such as a variety of sensors, Bluetooth, wireless devices, and internet. Then It produces certain outputs based on the given inputs. These outputs could be in the form of moving a motor, lighting LEDs, producing sounds, or even sending an HTTP request to a server. There are two main functions that are necessary for an Arduino program to compile and run:

- Setup () - this function runs only once at the beginning of the program. It can be used to initialize variables, declaring the mode of the pin whether it is input or output, and some other initialization settings.
- Loop () this function is called repeatedly and is halted once the board is turned off.

There are multiple advantages of using Arduino over any other micro-controller because it simplifies the hardware and software development required to operate a system [15]. Additionally, it offers some advantages over other system including:

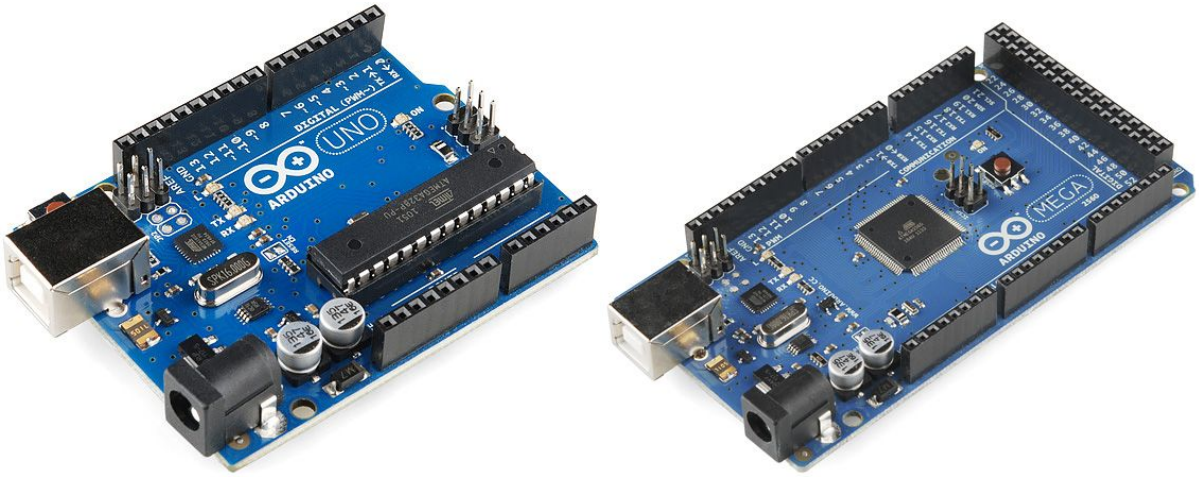
- Inexpensive - compared to other micro-controller platforms, Arduino boards are relatively low-priced.
- Simple clear programming environment - The programming environment of Arduino is user friendly. It is easy to use for beginners, yet flexible enough for advanced users to take advantage of as well. Since it does not limit the user to a specific programming language, it instead runs compiled assembly code using C/C++ compilers, meaning that the user can program the Arduino in Either C, C++ or micro C.

[1]https://www.arduino.cc/

- Cross platform - meaning that it could run on Windows, Mac OSX and Linux operating systems unlike other micro-controller systems which are limited to Windows only.
- Open source and extensible hardware - Experienced circuit designers can therefore make, expand and improve their own version of the module.

There are many models of Arduino boards. The most basic one is called Arduino Uno and a more advanced model called Arduino Mega. Figure 2.5 shows the two Arduino boards. Both of them have a standard A/B USB cable to be able to connect them to a pc for powering up or uploading programs. In addition, they both have an operating voltage of 5 volts. However, what differentiate them is that Arduino Uno has 14 pins and Arduino Mega has 54 pins. The Arduino Uno has an internal flash memory of 32KB, in contrast to the Mega board which has 256KB [16]. Arduino Uno is based on ATMEL micro-controller ATmega328 while Arduino Mega is based on ATMEL micro-controller ATmega2560 [17].

Figure 2.4: The Arduino IDE In Its Default State.



(a) Arduino Uno (b) Arduino Mega

Figure 2.5: Arduino Boards

Chapter 3

Related Work

A lot of related work has been carried out in the area of programming for young children. They used various teaching methods. Some work was done in the form of serious games that are implemented to teach children computational thinking and programming skills like Scratch, Kodable, Alice, Hour of Code [18, 19]. They teach the children with fun and creativity. They teach the children programming concepts like sequential, conditions, loops, variables, data structures etc.

3.1 Scratch

Scratch[1] is a block-based visual programming game that teaches many of the programming concepts mentioned above. It was developed by MIT university. Students collect the blocks appropriately by using drag and drop to create their desired program or game, see figure 3.1. It allows the children to learn programming by allowing them to build games through building blocks of code which is written in a simplified natural language [20]. The players choose to implement whatever the function they want. It is used as an introductory language as creation of interesting programs is easy, and many skills can be learned. Students found Scratch easy to use, and wanted to know more as they liked programming [21].

3.2 Alice

Alice[2] is an educational programming language that uses drag and drop environment to create programs. It also offers creating and developing 3D animations using 3D models. Users of Alice can place objects from a gallery into the virtual world that they have imagined, and then tiles can be programmed by dragging and dropping which represent logical structures. In addition, camera and lightning can be manipulated for further enhancement [22]. Both Scratch and Alice teach also more advanced programming concepts such as threads, events, and triggers. See figure 3.2.

[1] https://scratch.mit.edu/
[2] http://www.alice.org/

3.3 Hour of Code

Hour of Code[3], which is another platform for teaching children programming, was launched by code.org. The Hour of Code involved getting people to write short snippets of code to achieve pre-specified goals using Blockly, a visual block programming language [23]. It differs from Scratch and Alice because the player has to implement specific functions using the available blocks. The player builds the program that he/she was asked to do, see figure 3.3. Once the player finishes the required function, he/she is then asked to build the following function [24].

3.4 Kodable

Kodable[4] is a screen-based game designed for iPads. It is a maze game [25]. The players are required to provide the set of steps that will get them out of the maze using the up, down, right, and left symbols provided, see figure 3.4. The players are also required to collect coins while finding their way out. However, due to the lack of any description for the signs and symbols used in the game, the child can not understand alone the concepts especially in the more advanced levels that use loops. So, an instructor is needed while the child is playing to simplify the concept of each level.

On the other hand, there exist some tangible tools and kits that are used in the field of teaching children programming like mBot Ranger robot kit, and KIBO.

3.5 Makeblock mBot Ranger

The mBot Ranger[5] is an educational robot designed to help children explore and learn more about robotics. The mBot Ranger makes use of precision metal components and a pre-assembled Arduino controller board to build three different robots [26]. It comes with 3 control modes. First mode is obstacle avoidance mode where the robot can automatically sense obstacle ahead, and change the path to avoid it. The second mode is line-follow mode where the robot can travel freely along various black and white line. The third mode is the manual control mode where the children use a remote control to program mBot directly. It can be programmed using Scratch graphical programming language, and Arduino IDE to build up complex programs. Figure 3.5 shows mBot robotic kit.

3.6 KIBO

In addition, another tool to teach children programming is KIBO robotic kit[6]. It was created by Kinder Lab. It is designed for young children aged between 4-7 years old. It is a screen free

[3] https://hourofcode.com/
[4] https://www.kodable.com/
[5] https://www.makeblock.com/steam-kits/mbot
[6] http://kinderlabrobotics.com/kibo/

robot kit that allows the children to create and design their own robot. Children program the robot using tangible code made of wooden blocks without working with any form of screens. These blocks fit together with dowels and holes, with code printed on each side of the block. This kit includes 2 wheels, motors, distance sensor, sound sensor, light sensor, and a lantern for light output to glow white, blue, or red, see figure 3.6. Children get to decide what they want the robot to do, and then start arranging the blocks [5].

All of these previously mentioned tools and games encourage many of the children to learn more about computer science and engineering in the future. They can qualify the students to easily learn advanced concepts by self-learning through online courses or as a subject in school later on.

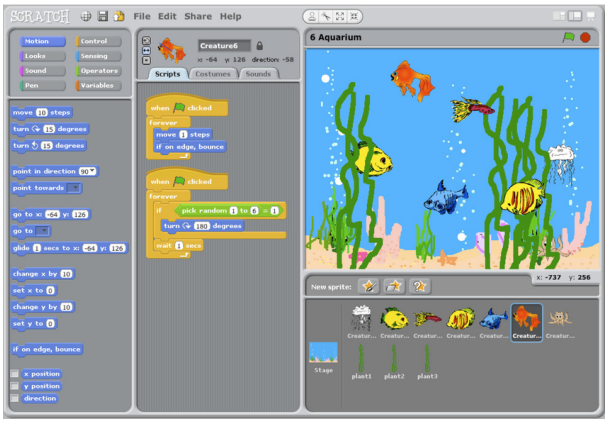


Figure 3.1: The Scratch user interface.

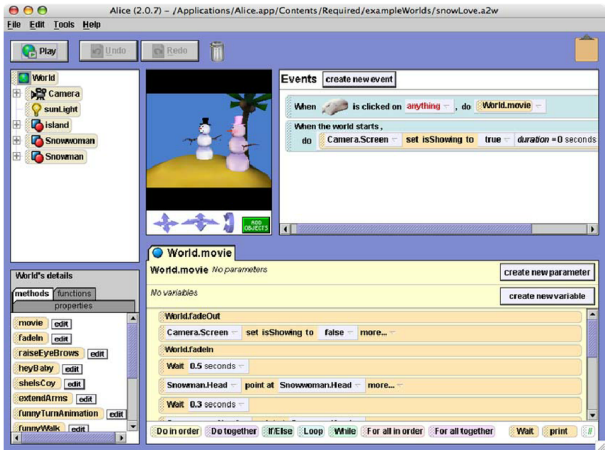
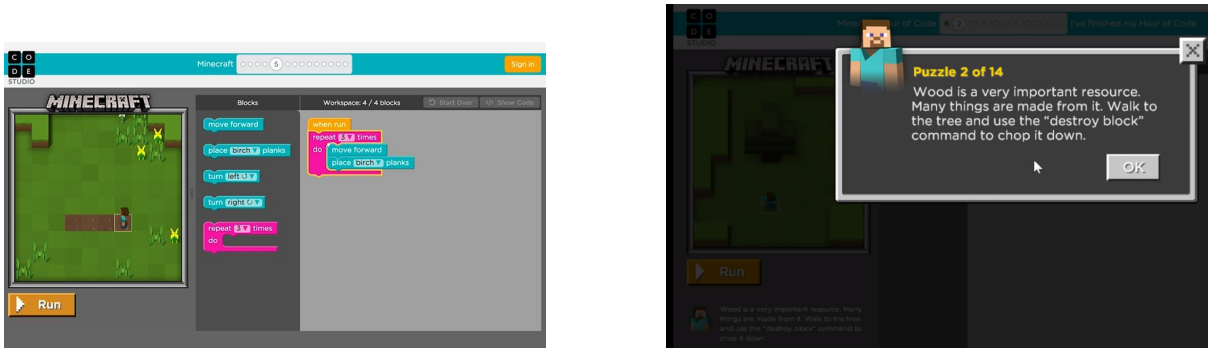


Figure 3.2: The Alice programming environment.



(a) Hour of Code Minecraft

(b) A task in Hour of Code

Figure 3.3: Hour of Code

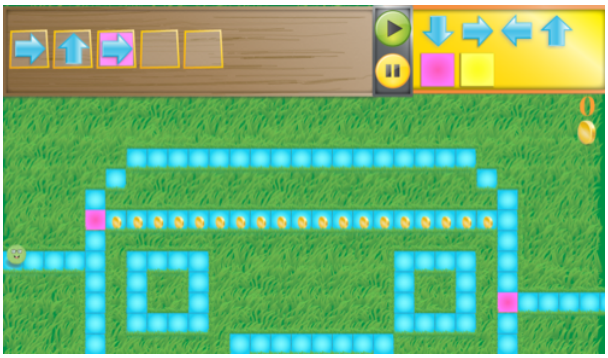


Figure 3.4: Kodable

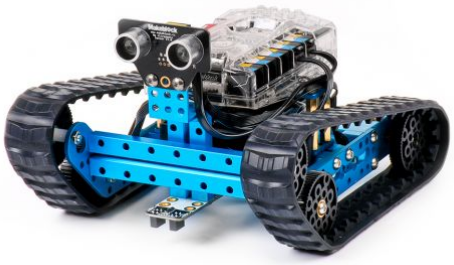


Figure 3.5: mBot Ranger

Figure 3.6: KIBO

Chapter 4

Methodology

This chapter introduces the architecture of the robot with all the features that it provides. It also describes all the modules and the sensors used along with the programming language of the robot. It discusses how to program the robot too.

4.1 Hardware System Architecture

There are several sensors that are embedded in the robot to help the children learn the basics of programming concepts and hardware. This section discusses the internal hardware and sensors used.

4.1.1 Micro-controller

Any electronic system must have MCU (Micro-Controller Unit) to operate. A micro-controller is a small computer on a single integrated circuit(a set of electronic circuits on one small flat piece or a chip of semiconductor material that is normally silicon). A micro-controller contains at least one CPU (processor cores), memory, and programmable input/output peripherals to be used in embedded applications. There are several micro-controllers to be used like BeagleBone, Raspberry Pi, and Arduino. However, the robot is implemented based on the Arduino board. The Arduino board is equipped with a micro-controller and sets of pins that are used to interface the sensors and the shields used by the robot. The robot was implemented using two different types of Arduino boards to share the workload (Arduino Mega and Arduino Uno) as shown in figure 4.1. Both boards communicate together through a protocol named I2C. The I2C protocol is a nice serial protocol because it enables the two micro-controllers to communicate together in any embedded electronic system by using two wires.

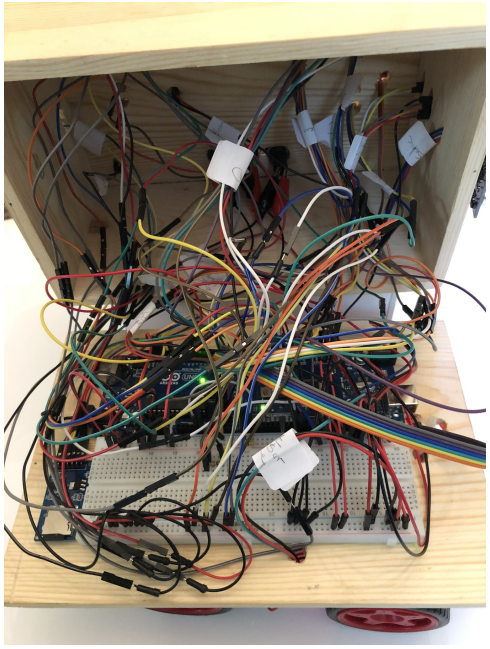


Figure 4.1: The Internal Connections of The Robot and the Two Arduino Boards That Are Used.

4.1.2 Chassis and Motors

The robot consists of a wooden chassis that is connected to four wheels to enable the movement of the robot in different directions. Each wheel is connected to a DC motor so that it can be activated or deactivated separately. All the DC motors are connected to H-bridge, an electronic circuit that switches the polarity of a voltage applied to a motor, to allow the DC motors to run forward or backwards. Figure 4.2 shows the chassis along with the motors.

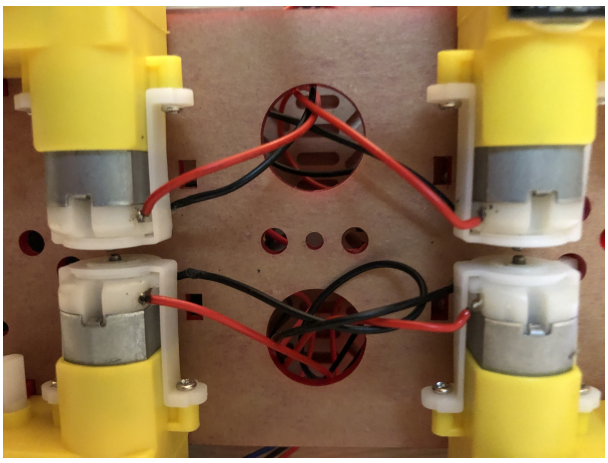


Figure 4.2: The Wooden Chassis and Four Motors.

4.1.3 Scanner of The Blocks

To be able to scan the programmable blocks that the child sorts them, a scanner is embedded in the robot that can reads RFID (radio-frequency identification) cards as shown in figure 4.3. These cards are invisibly attached to the wooden blocks. Each card is programmed to do a specific function. Once the robot finishes scanning the program, it will translate internally the scanned blocks into some program or commands that the robot can execute and perform.

Figure 4.3: The Scanner Of The Blocks.

4.1.4 Object Detection

In order to let the robot detects whether there is an object in front of it or not, an ultrasonic sensor is used as figure 4.4 shows. This sensor measures the distance by using ultrasonic waves. The ultrasonic waves are emitted from the head of the sensor, and the reflected waves are received back from the target. The ultrasonic sensor measures the distance to the target by measuring the time between the emission and reception.

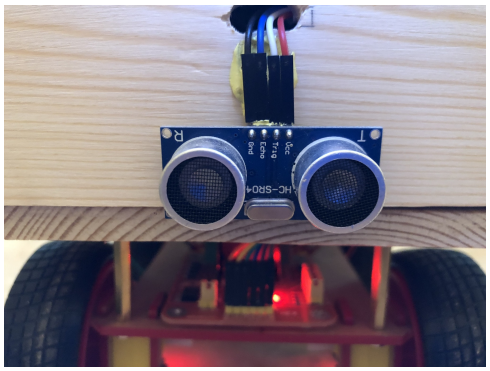


Figure 4.4: The Ultrasonic Sensor.

4.1.5 Color Detection

To let the robot identify different colors, a color sensor is used where it detects the color of the surface in RGB (red, green, blue) scale, see figure 4.5. The used color sensor is programmed to differentiate between 5 different colors which are: red, green, yellow, orange and blue.

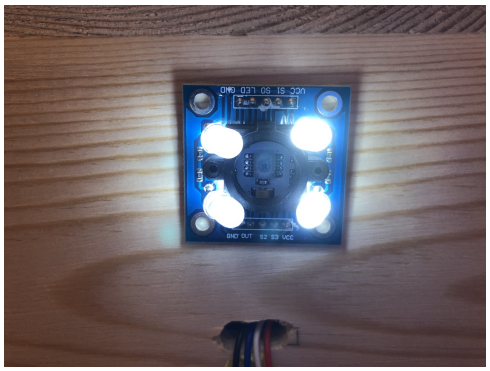


Figure 4.5: Color Sensor.

4.1.6 Clap Detection

The robot can recognize claps using the sound sensor that is embedded. The set point level of the sensor is adjusted using a potentiometer that is located at the back of the sensor to spot claps only. When the sensor detects that the sound has reached the set point, it flashes a led indicating that it has recognized a clap. Figure 4.6 shows the sound sensor used.

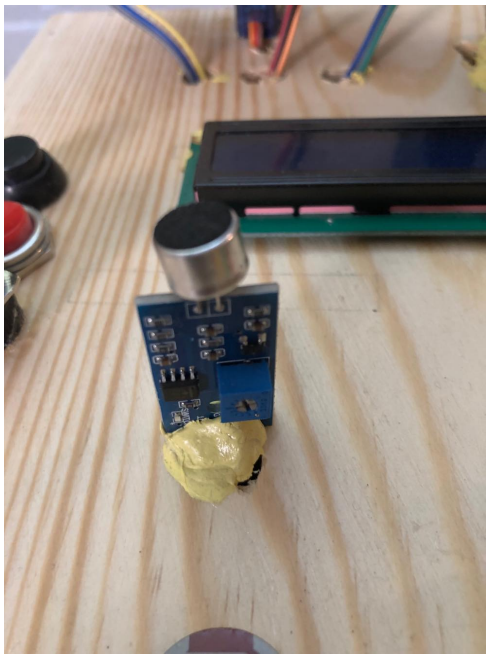


Figure 4.6: The Sound Sensor Which Is Used To Detect Claps.

4.1.7 Light Detection

Light and dark rooms can be recognized using a photo-resistor or LDR (light-dependent resistor). Using this resistor, it can read the light intensity of the surrounding room, and based on the reading, it can detect whether the room is dark or bright. Figure 4.7 shows the shape of the resistor.

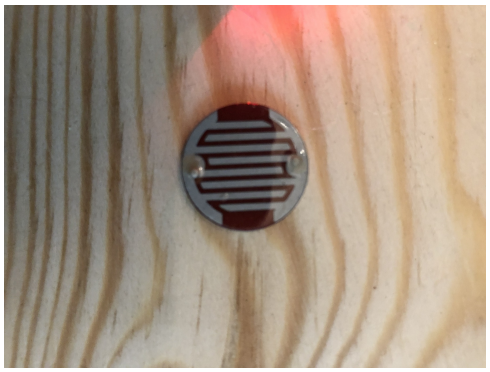


Figure 4.7: The Photo-resistor (LDR).

4.1.8 Head Sensing and Orientation

The robot has a head that is attached to a servo motor, a motor which allows the control of the angle of rotation and speed of rotation. This motor is used in order to let the head rotate to the

right, rotate to the left or look forward. In addition, three touch sensors are placed to the head of the robot. One on the right, one on the left, and one in the middle of the head to detect any touch happens to any part of the head. See figure 4.8

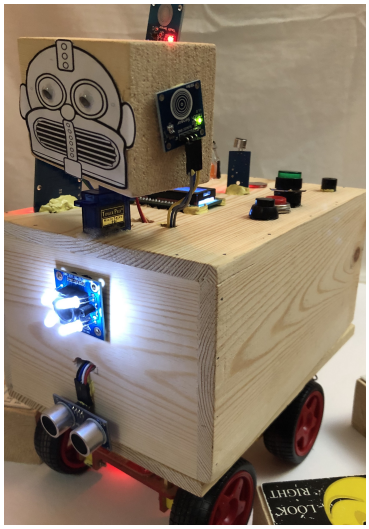
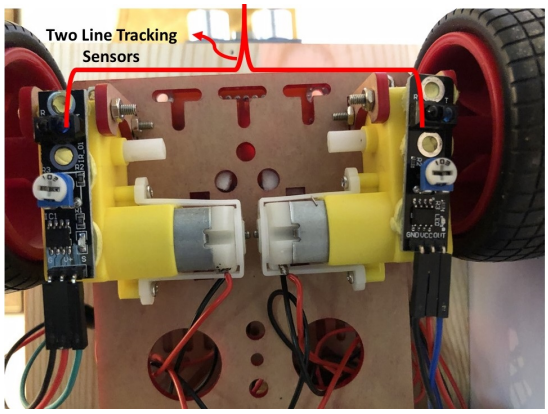


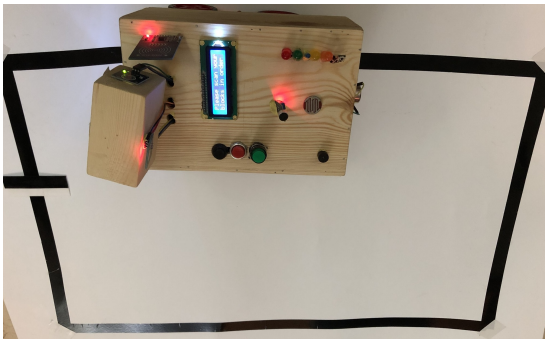
Figure 4.8: The Head Of The Robot Along With 3-Touch Sensors.

4.1.9 Line Tracking

In order to be able to track black lines, line follower sensor is used. The robot uses two-line follower sensors to be able to follow black lines on the floor as shown in figure 4.9. The idea of having two sensors not only one is that each one is placed near the front wheels (one at the right and one at the left). If the right sensor detects that it is over a black line now, so the robot has to turn slightly right. Same thing happens when the left sensor detects a black line, the robot has to turn slightly left to adjust its orientation.



(a) Two Line Tracking Sensors.



(b) The Robot Following Black-Lines.

Figure 4.9: Line Tracking

4.1.10 Light Output

Five different LEDs (Light-Emitting Diodes) are added. Each one has a specific color. The colors of the embedded LEDs are red, green, yellow, orange, and blue. They can be programmed to turn on, flash, or turn off. Figure 4.10 shows the LEDs are turned on.

Figure 4.10: LEDs Are Turned On.

4.1.11 Sounds

In order to produce sounds, a buzzer is added so that it can be programmed to beep. In addition, a speaker is added to help the child if he/she has something wrong with program structure as shown in figure 4.11.

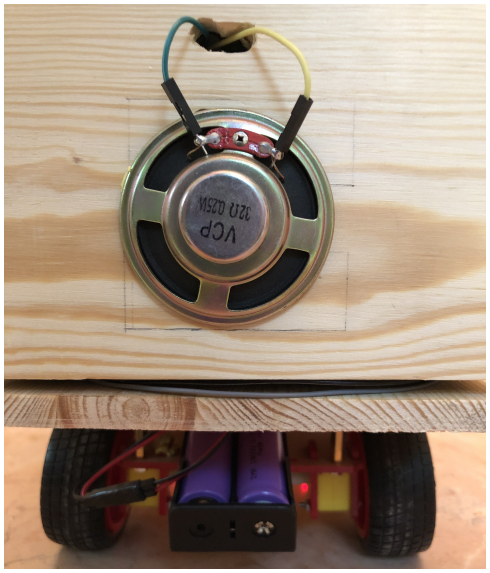
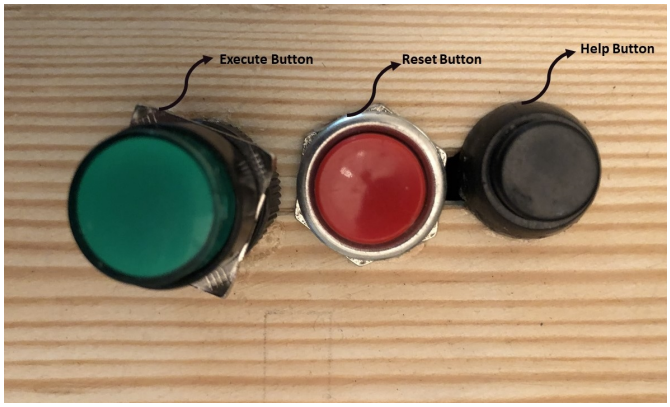


Figure 4.11: A Speaker To Play Help Tips.

4.1.12 Robot Control

In order to make things easier, 3 buttons are added to the robot as figure 4.12 shows. The green button is used to let the robot start executing what it scanned. When the child finishes scanning, he/she will press on the start execution button to see the robot functioning. The robot will not function unless the code structure is correct. The black button is used for help. If the program that the child has scanned contains some errors, the robot will not function, and the child will have the chance to use this help button. On pressing the help button, a specific audio file will be played through the speaker that will guide the child to the part in the program that contains the error. Accordingly, the child can know what is wrong, and what is preventing the robot from executing the program. The last button is used to reset all the robot to its default settings. Usually this button is pressed to start scanning a new program. Moreover, to enter a counter for repetitions and loops, an infrared receiver, and infrared remote control are used. The infrared receiver receives the signals which are transmitted by the remote, decodes these signals to a number, and then uses this number as a counter to do repetitions.



(a) 3 Buttons To Control The Robot

(b) IR-Receiver and Remote

Figure 4.12: Robot Control

4.1.13 Feedback

An LCD screen is added to the robot to display some important tips and messages like displaying what the robot is executing now, what condition is the robot checking, was the condition satisfied or not, the loop count if there was a repetition in the designed program, an external event that the robot is waiting for, and whether the structure of the scanned program is correct or not. Figure 4.13 shows an example of a feedback that the robot displays on the screen.

Figure 4.13: A Feedback Displayed The Screen.

4.1.14 The Robot

All of these sensors, motors, actuators, and electronic components are combined and embedded together to build the robot as figure 4.14 shows.

Figure 4.14: The Robot.

4.2 Programming Language of The Robot

The robot is designed to be programmed using TUI (Tangible User Interface) which is represented in the wooden programmable blocks as shown in figure 4.15. The child can scan the wooden blocks (command or program) that he/she wants to execute in order. Tangible wooden

blocks are used instead of GUI (Graphical User Interface), which relies on pictures and words on a computer screen like drag-and-drop, because of many reasons [27, 28, 29]:

- More fun
- More learn-able and enjoyable.
- Improves problem solving behavior.
- Easier for the children than holding a mouse for dragging and dropping.
- Children seem to be more involved.
- Blocks are familiar and playful objects.
- Ability to see, touch, organize and assemble the commands.

Furthermore, many educational theorists have stated that children can learn and think best when playing, moving, building and engaging with concrete objects. So, the robot is programmed using wooden programming blocks without the use of a computer, tablet or any other form of a screen. These wooden blocks are embedded with electronic identification card, to uniquely identify each block. The installed scanner in the robot allows the children to scan the invisible ID cards of the programming blocks and send their program to the robot to perform their desired function. When the child wants to design a series of actions to let the robot perform, the child will assemble the program as a line of the wooden programming blocks, scan each block by order using the embedded scanner in the robot, and watch the robot performing the desired functions.

Figure 4.15: The Wooden Programmable Blocks.

4.3 Functionalities of The Robot

The child has the opportunity to select what he/she wants the robot to do. He/she can make a program of whatever number of wooden blocks he/she wants from the available blocks. Fourty wooden blocks were built and associated with this robotic kit. Each of the available wooden blocks is programmed to do a definite function. Table 4.1 explains the functionality of each block. These programmable blocks cover the three programming concepts which are sequential programming, conditional programming, iteration programming.

Table 4.1: Wooden Programmable Blocks

Block	Function
Begin Program	The first block of any program.
End Program	The last block of any program.
Move Forward	Let the robot move one step forward.
Move Backward	Let the robot move one step backward.
Turn Right	Let the robot turn right.
Turn Left	Let the robot turn left.
Spin Right	Let the robot rotate 360 degrees to the right.
Spin Left	Let the robot rotate 360 degrees to the left.
Track Black Lines	Let the robot follow black lines on the ground.
Beep	Let the robot Beep.
Look Forward	Adjust the face of the robot to look forward.
Look Right	Adjust the face of the robot to look right.
Look Left	Adjust the face of the robot to look left.
Wait Until	Let the robot wait until a specific external event happens. Example: wait for a clap, wait for specific colors to be detected (red, green, blue, yellow, orange), wait for the room to be dark/bright.
If	To make conditions in the program. Must be followed by a something that can be checked on.
Then	The part of the program to be executed if the condition is satisfied.
Else	The part of the program to be executed if the condition is not satisfied.
End If	To terminate the if statement.
Repeat	To do a loop or a repetition in the program.
End Repeat	To terminate the repeat statement.
Turn on Led	To turn on a specific led. Must be followed by the led in which the user wants to turn it on.
Turn off Led	To turn off a specific led. Must be followed by the led in which the user wants to turn it off.
Flash Led	To flash/blink a specific led. Must be followed by the led in which the user wants to flash it.
Red Led	To control the red led.
Green Led	To control the green led.
Blue Led	To control the blue led.
Yellow Led	To control the yellow led.

continued on next page.

continued from previous page.

Wooden Programmable Blocks	
Block	Function
Orange Led	To control the orange led.
Object Near	To let the robot check whether there is object near or not.
Bright Room	To check whether the room is bright or not. Can also be used with wait statement.
Dark Room	To check whether the room is dark or not. Can also be used with wait statement.
Red Color Detected	To check whether the robot sees a red color or not. Can also be used with wait statement.
Green Color Detected	To check whether the robot sees a green color or not. Can also be used with wait statement.
Blue Color Detected	To check whether the robot sees a blue color or not. Can also be used with wait statement.
Yellow Color Detected	To check whether the robot sees a yellow color or not. Can also be used with wait statement.
Orange Color Detected	To check whether the robot sees an orange color or not. Can also be used with wait statement.
Clap	Let the robot detect/wait for a clap.
Right Face Touch	Let the robot check whether something touches the right part of its head or not. Can be used with wait statement.
Middle Face Touch	Let the robot check whether something touches the middle part of its head or not. Can be used with wait statement.
Left Face Touch	Let the robot check whether something touches the left part of its head or not. Can be used with wait statement.

4.4 Programming the Robot

To program the robot, the child must first assemble the wooden blocks that will be used in the program in order. The child must place each of the programmable blocks near the scanner in order to scan the whole program as shown in figure 4.16. After each block is scanned, the green led flashes, and a beep will be produced by the buzzer to notify the child that the block is scanned successfully. When the child scans a specific block, the robot transforms the action of the scanned block into code or program that it can understand and perform later. Every block should be scanned at most once in the program, so that the concept of the loop/repeat could be introduced. The only blocks that can be scanned more than once are: Turn on Led, Turn off Led, Flash Led, Red Led, Green Led, Blue Led, Yellow Led, and Orange Led so that the

process of turning on/off/flashing many LEDs in a single program can be more easier. Once the child finishes scanning all the needed blocks for the required program, the start execute button shown in figure 4.12 is pressed to let the robot start functioning. However, the robot will not function unless the program structure is 100 % correct. So, if the child missed something, or did something wrong in the code, the robot will not execute the scanned program. By then, the child can use the help button shown in figure 4.12 to listen for some tips. Every time the child presses the help button, a different help tip is played through the available speakers to guide him to solve the error. Here are the rules that must be available in each scanned program:

- The Begin Program block must be the first block to be scanned.
- The End Program block must be the last block to be scanned.
- If there was a loop in the program, the Repeat block must be scanned, the counter of the loop must be entered using the remote, and the End Repeat block must be scanned to terminate the part which will be executed more than once.
- If there was a conditional check in the program, the If block must be scanned, followed by the condition, followed by the Then block, followed by the instruction(s) to be executed in case the condition is satisfied, followed by the Else block, followed by the instruction(s) to be executed in case the condition is not satisfied, followed by the End If block to terminate the conditional part.
- If there was a wait statement, the Wait Until block must be scanned followed by the event in which the robot will wait for it.

The above rules are basically the main programming concepts. So, failing to abide by these mentioned rules will lead the robot to not execute the program. Figure 4.17 shows 3 simple programs using wooden blocks that covers (sequential, conditions, looping) concepts.

Figure 4.16: A Child Scanning The Blocks.

(a) Sequential Program

(b) Conditional Program

(c) Iteration Program

Figure 4.17: Three Simple Programs Using Wooden Blocks That Covers The Three Programming Concepts.

Chapter 5

Experimental Design

The work done in this thesis studies the effect of using an educational robot to teach young children the main concepts of programming versus the traditional learning methods. This is achieved by building up the robot, and testing it with young children. The extent to which the children have enjoyed, interacted with the robot, and learned from it is examined. The aim of the experiment is to prove or reject the null hypothesis. The null hypothesis states that the effect of using the robot does not differ from the fact that traditional educational methods are used in learning programming. The first hypothesis (H1) states that the learning gain of the two methods does not differ. The second hypothesis (H2) claims that the commitment and the engagement level of the participants of the two groups is not different. The third hypothesis (H3) states that there is no difference between the system usability of the two methods.

5.1 Focus Group

A focus group is a group of people assembled to participate in a discussion about a product to provide feed-back before testing or launching [30]. A focus group was made to get feedback with the help of a school before the experiment was conducted. The school provided five children to try the robot, and gave their opinion to improve it, see figure 5.1. The children said that the robot is taking much time to execute every block. Meaning that there was some time delay between executing the function of the block and the execution of the following one. This was solved by decreasing the delay time between the execution of each block. Another comment is that the connections of the robot (including the wiring and boards) must be totally covered and invisible. This was supposed to be done before conducting the focus group, but the lack of time did not allow to make it then. However, the robot was totally covered, and connections were all hidden before doing the experiment. Moreover, children said that there must be another feedback or notification to indicate that the block has been scanned successfully other than the flashing of the green led as some times they did not notice the flashing led. So, a modification was done by programming the buzzer to produce a sound when scanning each block.

Figure 5.1: Focus Group

5.2 Between-Group Design

The experimental design process is simply a way of gathering tests which provide the most information possible to find the effects of varying different factors on the outcome of a process. An experiment can be designed in many ways. The one which is used to test the project was the between-group design. A between-group design is an experiment that has two groups or more. Each group is tested by a different testing factor simultaneously [31].

5.2.1 Participants

Twenty participants took part in this experiment. However, the results of four participants were excluded due to their basic knowledge in programming. So only the results of sixteen participants were collected and counted. All of the sixteen participants had no previous experience in coding. They were divided into two independent groups with an equal number of participants, a control group (n = 8) and an experimental group (n = 8). The participants were all children aging between (8-11) years. The experimental group used the robot, and the control group was exposed to traditional face-to-face learning method. There were no any criteria in the selection of the children. However, for ensuring the experimental homogeneity, everyone must not have any previous programming background.

5.2.2 Measures

Learning Gain Test
A test was prepared to assess the learning gain of the participated children. It is a short story with a problem to solve, see Appendix A. The problem targets the three programming concepts

(sequential, conditions, and loops) in addition to some hardware concepts. The children have the same 40 blocks printed with the test sheet to number them in order to solve the problem. Some children needed help in reading and understanding the flow of the question. The participant is tested before learning with the traditional method or engaging with the robot to know if he/she has previous knowledge in programming. After finishing the experiment, the participant is tested again to know how much information did he/she gain. The children took this test before and after engaging with the robot or learning on board. The pre and the post tests are compared to assess the knowledge gained by the children. Both the pre and post tests were the identically the same to guarantee that both of them have the same difficulty level.

Engagement Level Test
Engagement is the concept of how the person is being committed to something. Several factors define the engagement while learning like fun, control, excitement, enjoyment, interest, attention, and enthusiasm [32]. This questionnaire contains eleven questions, see Appendix B. These questions are 5-Likert scale used to indicate the overall flow and experience that the participants have gone through by using the robot or the normal teaching method. The questionnaire is given to the child at the end of the experiment. The child must choose an answer for each question. The options of the answer were Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree.

System Usability Scale
The System Usability Scale (SUS) provides a good and reliable tool for measuring the usability of the system (ease of use) [33]. It consists of a 10-point questionnaire that is given at the end of the experiment, see Appendix B. The respondents have 5 response options to choose for each question from Strongly Disagree to Strongly Agree. The test offers the benefits of using it with reliable results on small sample size, being very easy for the participants, and allowing to distinguish effectively between usable and useless systems.

5.2.3 Procedure

Each participant of the two groups was experimented individually. Every participant was given 45 minutes session to finish the whole experiment. The process of the experiment consisted of 4 phases as shown in figure 5.2. The first phase was a hard-copied pre-test that the child tries to answer it in 10 minutes. Once the 10 minutes have finished, and whether the child finished solving the pre-test or not, the second phase of the experiment starts which lasts for 25-30 minutes. This phase differs depending on the group. For the experimental group, participants are introduced to the sensors available in the robot and learn the three programming concepts by engaging with the robot, knowing how it works, how it is programmed, and physically realizing the function of each wooden block. On the other hand, the participants of the control group sit and listen to explanations on the board for the same programming concepts. Last but not least, during phase three which lasts for 10 minutes, the participants of both groups are asked to solve the post-test, which is the same test that was given for them at the beginning of the session to know how much they learned. Finally, they are requested to fill in the engagement level and system usability scale questionnaires in the last 5 minutes. Figure 5.3 and 5.4 show children doing the experiment.

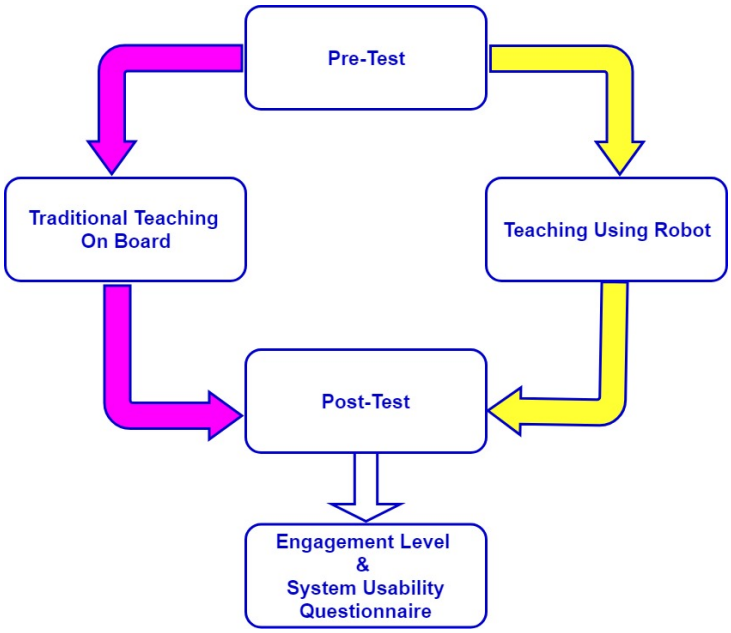
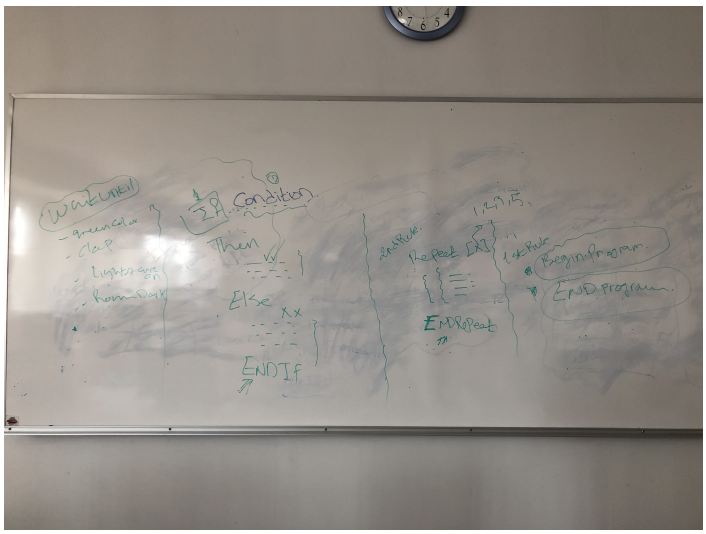


Figure 5.2: Experiment Process

Figure 5.3: A Child From The Experimental Group Solving Pre/Post Test

(a) A Child From The Control Group Solving Pre/Post Test



(b) Explanation On Board For The Control Group

Figure 5.4: Control Group

Chapter 6

Testing and Results

The learning gain, engagement level, and system usability scale were measured after using the corresponding learning approach to compare between both learning methods. In this chapter, the results of all the previous tests that have been conducted during the experiment are reported. For analyzing the data comparison between the two groups, an independent t-test is applied using SPSS (Statistical Package for the Social Sciences). An independent sample t-test is used to clarify whether the learning gain, the system usability scale, and the level of involvement between the control group and the experimental group are significantly different or not.

6.1 Learning Gain Test Results

This test aims to measure the variation between the learning result of the group participants exposed to the traditional learning methods and the other group exposed to the educational robot. It is calculated by subtracting the grade of the pre-test from the post-test. The difference between the results of the tests showed that the learning gain of the children that had the chance to utilize the robot was higher than the learning gain of the children that were taught using the traditional methods. The independent t-test also proved that the learning gain of the experimental group (the group that used the robot) (M = 6.9375, SD = 2.11183) was significantly higher than the gain of the control group (the traditional learning method group) (M = 4.3750, SD = 1.92261) ($t(14) = 2.538$, $p < 0.05$) as shown in table 6.1 and 6.2. So, this rejects the hypothesis which stated that there is no difference in the learning gain level between teaching using the traditional learning method and teaching using the robot (H1).

Table 6.1: Independent t-test results of the learning gain

Group	N	Mean	Standard Deviation
Experimental	8	6.9375	2.11183
Control	8	4.3750	1.92261

Table 6.2: Independent t-test results of the learning gain

t	p	df
2.538	0.024	14

6.2 Engagement Level Test Results

The aim of this test is to measure the overall level of engagement and involvement of the children of both groups. The group that achieved the highest engagement level was determined by comparing the results of both groups, the control group and the experimental group. The results of the independent t-test of the two groups illustrated that the engagement level of the participants of the group that was exposed to the robotic kit (M = 4.1525, SD = 0.43657) was significantly higher than the engagement level of the traditional classroom group. (M = 2.9275, SD = 0.76999) ($t(14) = 3.914$, $p < 0.05$) as shown in table 6.3 and 6.4. This rejects the hypothesis which stated that there is no difference in the engagement level between the usage of the robot and the traditional learning method (H2).

Table 6.3: Independent t-test results of the engagement level

Group	N	Mean	Standard Deviation
Experimental	8	4.1525	0.43657
Control	8	2.9275	0.76999

Table 6.4: Independent t-test results of the engagement level

t	p	df
3.914	0.002	14

6.3 System Usability Scale Results

The aim of this test is to measure the usability of the system and the satisfaction of the users. A system is considered usable if it is efficient, satisfying and intuitive. The analysis of the results demonstrated that the usability of the robot (M = 4.0, SD = 0.48697) was significantly higher than the traditional learning classroom (M = 2.8750, SD = 0.64973) ($t(14) = 3.919$, $p < 0.05$)

as tables 6.5 and 6.6 show. This rejects the hypothesis which stated that there is no difference in the system usability of the robot and that of the traditional learning method (H3).

Table 6.5: Independent t-test results of the system usability

Group	N	Mean	Standard Deviation
Experimental	8	4.0	0.48697
Control	8	2.8750	0.64973

Table 6.6: Independent t-test results of the system usability

t	p	df
3.919	0.002	14

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Studies show that technology is a very strong educational tool. The integration of technology in education can improve the learning process and the results of the students. It also helps in making learning more meaningful, easier, and enjoyable. Robotics and programming offer a new and exciting way to address the T of the technology and the E of the engineering that are most neglected in early childhood STEM education.

The work presented in this thesis was to implement an educational robot for young aged children of the age between 8-11 that teaches the three main programming concepts (sequential, Conditions, loops) along with some hardware without the use of any computers or screen-time. The children can program this robot using tangible programmable wooden blocks. The primary objective of this study is to test the effective use of that educational robot in teaching children basics of programming, computational thinking, and hardware. This study demonstrated that young children can master the foundational concepts regarding programming a robot in addition to complex programming concepts such as loops and conditional statements.

Two different groups participated in this study. One group tested the robot, and the other group was taught using traditional educational method by explaining on board. By comparing the knowledge gain of the participants of the two groups, it was concluded that using the robot in learning has more effect on the learning gain and engagement level of the children than using the traditional learning method.

7.2 Limitations and Future Work

The only limitation that was faced during testing is that every child was limited with only 45 minutes to finish the experiment. This led to explaining the concepts quickly, and not getting into deep details. So, the children have to follow-up and receive much information in a short time. Consequently, better results could happen if the session was longer.

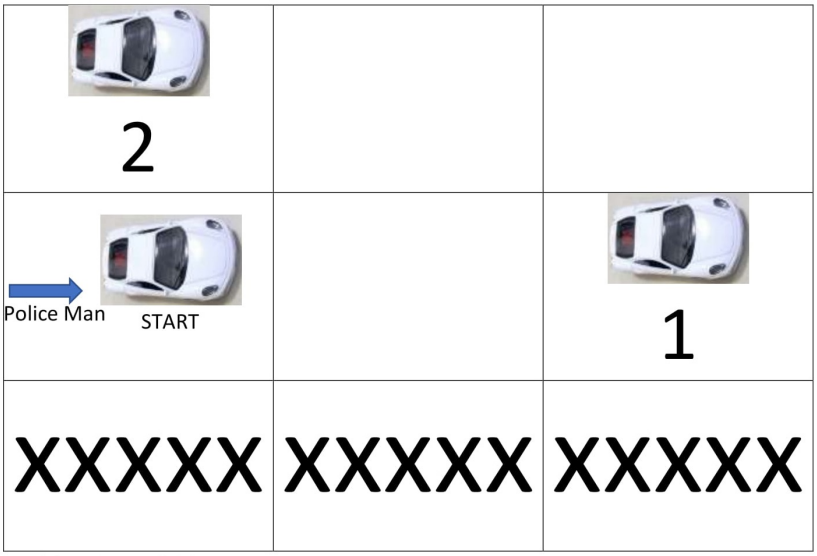
Wide variety of ideas can be added to the robot. New sensors can be added as well as new programming blocks to expand the functionalities of the robot. Moreover, the robot needs to target more advanced concepts in programming and computational thinking like data structures, and methods. In addition, the robot supported only one If statement and one repeat loop in the program. So further work has to consider adding multiple If statements, multiple repeat loops, nested If statement, and nested loops in the code. Furthermore, higher quality and higher volume speakers could be used as the children were having difficulty in listening to the helping tips due to the noise produced by the speaker.

Additionally, based on the feedback collected from the experiment done, some extra features can be added to the robot to satisfy all the needs of the children. Many children see that it will be very useful and helpful to display the program that they have scanned so far on a screen since sometimes the children forget what they have scanned. Another feature that was suggested which is the ability to edit the program while scanning because if the children have done something wrong while scanning, they have to repeat the whole scanning process from the beginning. One more feature is that to separate the scanning module from the robot itself so that the children do not get annoyed from holding each block or holding the whole robot to scan the program.

Appendix

Appendix A

Pre/Post Test



Problem Description:

The robot wants to run from a policeman that is coming from the back. The robot must rotate his face right and left to see where to go. He found a wall when he looked right preventing its movement to the right, so it has to go forward or go left depending on whether it will find something avoiding its movement in front of it or not. If it didn't find anything in front of it avoiding its motion, so it has to go ahead for two steps (at location 1 in the grid), and beeps once with every step it takes. If there was any obstacle avoiding its movement ahead, then it must go to the left (at location 2 in the grid). Whether it goes to location 1 or 2, it must wait until it hears a clap then flash a red led. Note: the final orientation of the robot must be as in the figures indicated above. Please mention also the hardware components that will be used by the robot (motors, sensors,…).

Possible Solution:

- **Begin Program, Look Right, Look Left, Look Forward, If, Object Near, Then, Turn Right, Move Backward, Turn Left, Else, Repeat(2), Move Forward, Beep, End Repeat, End If, Wait Until, Clap, Flash Led, Red Led, End Program.**

Solution Sheet

Name: Age: Comment:

If	End Program	Move Backward	Wait Until	Green Color detected
Begin Program	Turn on Led	End If	Room Dark	Turn Left
Then	Red Led	Flash Led	Move Forward	Green Led
Else	Blue Led	Repeat	End Repeat	Look Right
Turn Right	Spin Right	Clap	Object Near	Bright Detected
Middle Touch Detected	Red Color Detected	Beep	Yellow Led	Turn Off
Yellow Color Detected	Spin Left	Orange Color Detected	Dark Detected	Look Right
Look Forward	Orange Led	Blue Color Detected	Look Left	Track Black Lines

Pre-Test Solution

Post-Test Solution

Appendix B

Questionnaire

Engagement Level Test

Questions used to measure control, interest and enjoyment

- I felt in control of what I was doing.
- I was absorbed intensely by the activity
- I found the activities enjoyable
- I thought about other things
- I found the activities interesting
- I was frustrated by what I was doing
- The activities bored me
- I was aware of distractions
- the activities excited my curiosity
- I knew the right thing to do
- It required a lot of effort for me to concentrate on the activities

System Usability Scale

Questions used to measure usability of the system

- I think that i would like to use this tool frequently
- I found this tool unnecessarily complex
- I thought this tool was easy to use
- I think that I would need assistance to be able to use this tool
- I found the various functions in this tool were well integrated
- I thought there was too much inconsistency in this tool
- I would imagine the most people would learn to use this tool very quickly
- I found this tool very cumbersome/awkward to use
- I felt very confident using this tool
- I needed to learn a lot of things before I could get going with this tool

Appendix C

Lists

List of Figures

List of Tables

Bibliography

[1] Marina Bers, Safoura Seddighin, and Amanda Sullivan. Ready for robotics: Bringing together the t and e of stem in early childhood teacher education. *Journal of Technology and Teacher Education*, 21(3):355–377, 2013.

[2] Amanda Sullivan and Marina Umaschi Bers. Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, 26(1):3–20, 2016.

[3] Francisco J Estrada. Practical robotics in computer science using the lego nxt: An experience report. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education*, pages 329–334. ACM, 2017.

[4] Elizabeth R Kazakoff and Marina Umaschi Bers. Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4):553–573, 2014.

[5] Mollie Elkin, Amanda Sullivan, and Marina Umaschi Bers. Programming with the kibo robotics kit in preschool classrooms. *Computers in the Schools*, 33(3):169–186, 2016.

[6] Alan F Blackwell. What is programming? In *PPIG*, page 20, 2002.

[7] Control structures. https://docs.oracle.com/cd/A84870_01/doc/appdev.816/a77069/03_struc.htm.

[8] Douglas H Clements and Dominic F Gullo. Effects of computer programming on young children's cognition. *Journal of educational psychology*, 76(6):1051, 1984.

[9] Why teach kids programming? https://www.robocamp.eu/en/why-teach-kids-programming/.

[10] Sezer Kanbul and Huseyin Uzunboylu. Importance of coding education and robotic applications for achieving 21st-century skills in north cyprus. *International Journal of Emerging Technologies in Learning*, 12(1), 2017.

[11] Coding for kids: Reasons kids should get started, and how they can find success. https://www.idtech.com/blog/5-reasons-your-child-should-learn-to-code.

[12] Why children have to learn the importance of robotics and coding. https://www.gigglebot.io/blogs/news/why-children-have-to-learn-the-importance-of-robotics-and-coding.

[13] Massimo Banzi and Michael Shiloh. *Getting started with Arduino: the open source electronics prototyping platform*. Maker Media, Inc., 2014.

[14] What is arduino? http://www.arduino.cc..

[15] Gerald W Recktenwald and David E Hall. Using arduino as a platform for programming, design and measurement in a freshman engineering course. In *Proceedings of the American Society for Engineering Education Annual Conference & Exposition*, 2011.

[16] Arduino uno vs arduino mega. https://www.arrow.com/en/research-and-events/articles/arduino-uno-vs-mega-vs-micro.

[17] RH Kumar, AU Roopa, and DEVI4 P SATHIYA. Arduino atmega-328 microcontroller. *International journal of innovative research in electrical, electronics, instrumentation and control engineering*, 3(4):27–29, 2015.

[18] Cagin Kazimoglu, Mary Kiernan, Liz Bacon, and Lachlan Mackinnon. A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioral Sciences*, 47:1991–1999, 2012.

[19] 10 tools to teach kids the basics of programming. https://www.hongkiat.com/blog/programming-tools-kids/.

[20] Michal Armoni, Orni Meerbaum-Salant, and Mordechai Ben-Ari. From scratch to real programming. *ACM Transactions on Computing Education (TOCE)*, 14(4):25, 2015.

[21] Filiz Kalelioğlu and Yasemin Gülbahar. The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, 13(1), 2014.

[22] Edward R Sykes. Determining the effectiveness of the 3d alice programming environment at the computer science i level. *Journal of Educational Computing Research*, 36(2):223–244, 2007.

[23] Jie Du, Hayden Wimmer, and Roy Rada. hour of code: A case study. In *Proceedings of the EDSIG Conference ISSN*, volume 2473, page 3857, 2017.

[24] Jie Liu, Hayden Wimmer, and Roy Rada. " hour of code: Can it change students attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice*, 15:53, 2016.

[25] Kodable teaches kids to code before they learn to read. https://readwrite.com/2013/04/23/kodable-teaches-kids-to-code-before-they-learn-to-read/.

[26] mbot is a powerful robot for beginners. https://newlearningtimes.com/cms/article/3777/mbot-is-a-powerful-robot-for-beginners.

[27] Michael S Horn, Erin Treacy Solovey, R Jordan Crouser, and Robert JK Jacob. Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 975–984. ACM, 2009.

[28] Amanda Strawhacker, Amanda Sullivan, and Marina Umaschi Bers. Tui, gui, hui: is a bimodal interface truly worth the sum of its parts? In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 309–312. ACM, 2013.

[29] Peta Wyeth and Helen C Purchase. Tangible programming elements for young children. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 774–775. ACM, 2002.

[30] Focus groups. https://bha.health.maryland.gov/OMPP/Documents/CollectingData.pdf.

[31] Between group design. https://explorable.com/between-subjects-design..

[32] Rosa Mikeal Martey, Kate Kenski, James Folkestad, Laurie Feldman, Elana Gordis, Adrienne Shaw, Jennifer Stromer-Galley, Ben Clegg, Hui Zhang, Nissim Kaufman, et al. Measuring game engagement: multiple methods and construct complexity. *Simulation & Gaming*, 45(4-5):528–547, 2014.

[33] System usability scale. https://blog.hubspot.com/service/system-usability-scale-sus.