



Palestine Polytechnic University

College of Information Technology And Computer Engineering

Department of Computer Systems Engineering

Project Name: Patients server robot using Brain-Computer interface.

Project Team:

Dumooa Al-Asafrah

Ruba Titi .

Supervisor:

Dr. Alaa Halawani

**This work has been submitted for the Bachelor's degree in
Computer Systems Engineering**

Acknowledgement

The beginning and the best beginning, and before anything else, we thank God Almighty for granting us success and facilitating us to reach this stage and complete this project.

We thank him for his facilitation, conciliation, and care that accompanied us and surmounted all difficulties and obstacles.

We extend our thanks and gratitude to the supervisor, Dr. Alaa Al-Halawani, who did not hesitate to provide advice, guidance and instructions to complete this work despite all the pressures, he put us in his first interest.

And do not forget to extend our thanks to M. Wael Al-Takruri, who accompanied us throughout the work trip on the project, and was our reference in difficult times.

We would like to thank the friends who accompanied us during the study trip. We thank them for their support, encouragement and presence.

At the end , we had the right to thank our parents for their material and moral support and for all the effort they made for us to get here.

Abstract

The ability of a person to meet his needs is one of the most important things that he should have in order to practice his normal life. This project addresses the idea of a server robot for people who suffer from disability (paralyzed), so that the robot provides and meets their basic needs (hunger, thirst, pain) without anyone's help. The methodology that we followed in this project is using an EPOC EMOTIV headset, the Python language for programming, in addition to sensors that complement and help the robot. The robot is able to recognize a patient's need by reading his thoughts when he needs a certain thing and to bring the recognized patient's need to him. The long term of our work is to have a nurse robot specific for a patient that is capable of meeting most of the patient's needs even if he does not have it in the same place - applying the Locality principle.

List of contents

1. Chapter1: Introduction	7
1.1. Overview	7
1.2. Motivation and importance	7
1.3. Problem statement.....	7
1.4. Objectives	7
1.5. Short system description	8
1.6. System requirement	9
2. Chapter2: Background.	9
2.1. Overview.....	9
2.2. Brain-Computer interface.	9
2.3. Literature review	10
2.3.1. BCI achievements outside the PPU	10
2.3.2. BCI achievements inside the PPU	11
2.4. Hardware Components Options	11
2.5. Software options	25
3. Chapter3: Design	29
3.1. Overview.....	29
3.2. Brief description of the System.....	29
3.3. Mechanical and Hardware Design of the system.....	29
3.4. Hardware setup	32
3.5. Block diagram for the robot system.....	32
3.6. System Schematics.....	34
3.7. System programming Flowchart.....	35
4. Chapter 4: Implementation.....	36
4.1. Overview.....	36
4.2. Brain signals acquisition implementation	36
4.3. Hardware implementation.....	37
4.4. Software Implementation.....	40
4.4.1. Raspberry pi implementation	40
4.4.2. Arduino Implementation	42
4.4.3. Implementation of automatically signing in the user account.....	42

4.4.4.	Signal Analysis Implementation	43
4.4.5.	Implementation of automatically activation the SDK.....	43
4.4.6.	Implementation of automatically running the python script	43
4.4.7.	Implementation of control the robot when a specific signal is initiated by the user	43
5.	Chapter5: Validation and Results	46
5.1.	Hardware Testing.....	46
5.1.1.	Arduino Testing	46
5.1.2.	EPOC EMOTIV Headset testing	46
5.1.3.	DC motors and drivers testing	47
5.1.4.	Lead Screw mechanism testing.....	47
5.1.5.	PIR sensors testing.....	47
5.1.6.	System overall Test.....	47
5.2.	Software Testing	48
5.2.1.	Pyfirmata library testing.....	48
5.2.2.	AutoHotkey testing	48
5.3.	Results.....	48
	Conclusion and Future work.....	50
	References.....	51

List of Figures

Figure 1.1 :	BCI system components	8
Figure 2.1 :	BCI system [2].....	10
Figure 2.2 :	ECoG procedure[8]	12
Figure 2.3 :	EEG procedure[9].....	12
Figure 2.4 :	EMOTIV EPOC Headset[10].....	12
Figure 2.5 :	The hanging Aluminum boxes on a holder	14
Figure 2.6 :	The hanging Iron boxes on a holder	14
Figure 2.7 :	IR sensor.....	15
Figure 2.8:	PIR sensor.....	15
Figure 2.9:	Lead-Screw mechanism.....	18
Figure 2.10 :	Horseshoe magnet	18

Figure 2.11 : Stepper motor	21
Figure 2.12: DC motor	21
Figure 2.13: L298N driver	22
Figure 2.14 : L29Sd driver.....	22
Figure 2.15 : 12 volt battery.....	23
Figure 2.16 : shaft coupling	24
Figure 2.17 : Arduino mega	24
Figure 2.18 : ASUS ZenPower power bank.....	25
Figure 2.19 : EPOC Control Panel.....	26
Figure 2.20 : OnScreen keyboard in windows 10 system.....	28
Figure 3.1 : The mechanical design of the robot.....	30
Figure 3.2 : The mechanical design of the robot.....	30
Figure 3.3 : The mechanical system design	31
Figure 3.4 : System Block diagram.....	32
Figure 3.5 : System Schematic diagram.....	34
Figure 3.6 : System Flowchart	35
Figure 4.1 : The EPOC EMOTIV Headset channels that worked	36
Figure 4.2 : The final arrangement of the robot	37
Figure 4.3 : The final arrangement of the robot	37
Figure 4.4 : the robot structure with motors, wheels, Lead-Screw and boxes container.	38
Figure 4.5 : The boxes and boxes holder	39
Figure 4.6 : PIR sensors fixing.....	39
Figure 4.7 : The connected robot components	40
Figure 4.8 : Windows on raspberry imager.....	40
Figure 4.9 : Windows on Raspberry images	41
Figure 4.10 : StandardFirmata library.....	42
Figure 4.11 : Autologon application	42
Figure 4.12 : Autohotkey script	45

List of Tables

Table 1 : Comparison between ECoG, EEG and EMOTIV EPOC headset	12
Table 2 : Comparison between Raspberry Pi3 and the ordinary computer.....	14
Table 3: Comparison between Aluminum and Iron boxes.....	14
Table 4 : Comparison between IR and PIR sensors.....	15
Table 5 : Comparison between Lead Screw and Horseshoe	18
Table 6 : Comparison between Stepper and DC motor.....	20
Table 7 : Comparison between L298N and L2983d drivers	22
Table 8 : The results of the project	49

1. Chapter1: Introduction.

1.1. Overview

We all witness the difficulties that are faced by people with a physical disability such as paralyzed, hemiplegic, mobility deficiency, limb amputations or those who have an injured organ in practicing their daily life. Therefore scientists and researchers try to find alternatives to the affected organs of the disabled people that carry out the functions of the affected organ. And finally they managed to invent Brain-Computer Interface technology, which is a computer-based system that allows their users to communicate or control external devices using brain signals rather than the brain's normal output pathways of peripheral nerves and muscles [1] .

1.2. Motivation and importance

Because our specialization is computer system engineering and our passion is toward the medical humanitarian field, we decided to employ computer theories to help us serve our passion. As our project aims to provide paralyzed patients with a computer-guided assistant tool that performs some of the functions of their affected organs .

1.3. Problem statement

The problem that we are addressing is the inability of the paralyzed people to meet their hunger, thirst and pain main needs independently due to their defected organs. And up to now, there is no technology- based tools that carry out meeting their main needs on their behalf.

1.4. Objectives

1. Design a robot that has a cube base parallel to the ground with wheels to put the food, medicine or drink on it.

2. Using a particular sensor modality to acquire brain signals.
3. Analysis and classify the acquired signals using specific SDK to detect if the hunger, thirst and pain signals.
4. Translate the detected signal into commands that operate the robot to respond to the user's needs as follows: if the detected signal is a signal that indicates that the user feels hunger, the robot will bring the food box and serve it to the user. As for if the detected signal is a signal that indicates that the user feels pain , the robot will bring the medicine box and serve it to the user. And if the detected signal is a signal that indicates that the user feels thirsty , the robot will bring the drink box and serve it to the user.

1.5. Short system description

The system falls under Brain-Computer Interface(BCI) systems. So it comprises the components of the BCI system, which are basically sensor devices with signal processing circuit, computer, and output device which is the robot in our project. And they are connected sequentially together.

The sensors sense the initiated user's brain signals then pass them to the processing circuit that processes them and sends them to the computer. The computer reads the acquired signals, analyzes them by extracting the features of the read signals and translates them into commands that are sent to the robot.

The commands direct the robot to bring the food, water, or the medicine and serve it to the user if the initiated user's brain signal is hunger, thirst, or pain signal respectively.

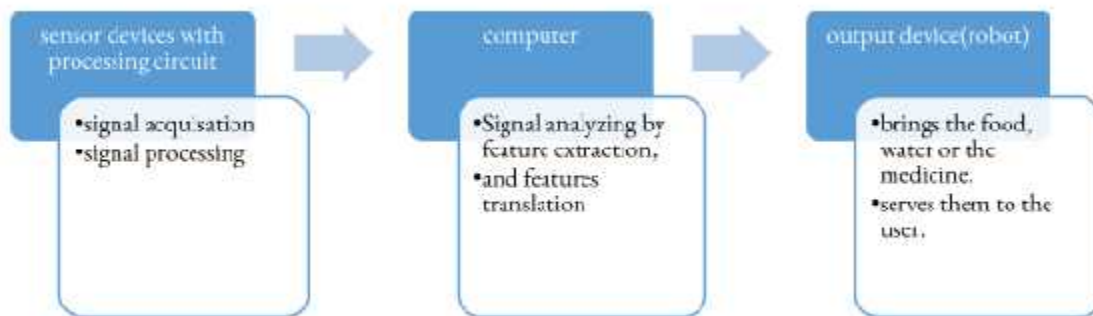


Figure 1.1 : BCI system components

1.6. System requirement

- Assistant robot that responds automatically to the user's hunger, thirst and pain needs,
- If the user felt hunger, it brings the food to the user and serves it to him,
- If the user felt pain, it brings the desired medicine to the user and serves it to him,
- And if the user felt thirst, bring water to the user and serve it to him.

2. Chapter2: Background.

2.1. Overview

This chapter shows a detailed theoretical background of the project, a brief description of hardware components and software techniques/methods that we are going to use in this project.

2.2. Brain-Computer interface.

It's a recently developed technology which has emerged from the intensified research and experiences about the human brain. It aims mainly to enable it's user to interact with the surrounding environment using his thoughts (electrical signals that are initiated when something is thought of) rather than his peripheral nerves and muscles.

It comprises hardware and software components which work in integrated sequential form to read the user thoughts then do an action that does the user's intention. And that's done by doing the BCI components to acquire the user's brain signals(thoughts) and processing them, analyze them by feature extraction, and translate them by translation algorithm into commands that relay to an output device that carries out the task of doing the desired action [1].

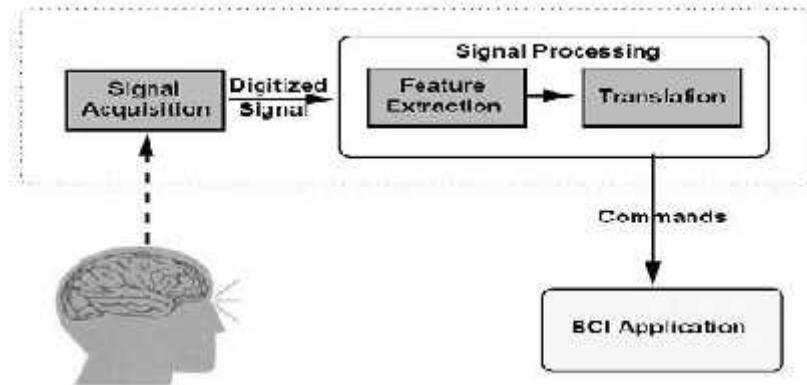


Figure 2.1 : : BCI system [2].

2.3. Literature review

The first attempt to evaluate the feasibility and practicality of utilizing the brain signals in a man-computer dialogue was introduced by pioneering work of Dr. J. Vidal in the early 1970's [3]. After that, practical attempts are started to implement Dr. J. Vidal idea on the real.

In this section we will show some BCI achievements outside the PPU and inside the PPU.

2.3.1. BCI achievements outside the PPU

- In the late 1970s, Wolpaw trained volunteers to control sensorimotor rhythm(one of the brain signals types) amplitudes and use them to move a cursor on a computer screen to aim to provide a new communication and control modality for individuals with severe motor deficits [4].
- By 2006, a microelectrode array (a devices that contain multiple (tens to thousands) microelectrodes through which neural signals are obtained) was implanted in the primary motor cortex(the primary brain region of the motor system) of a young man with complete Tetraplegia. Using the signals obtained from this electrode array, a BCI system enabled the patient to open simulated e-mail, operate a television, open and close a prosthetic hand, and perform rudimentary actions with a robotic arm [5].
- In 2011, Krusienski and Shih demonstrated that the recorded signals directly from the cortical surface can be translated by a BCI to allow a person to accurately write words by a visual keyboard without voice or limb movements [6].

- Sunny et al suggested in 2014 a purely non-invasive in nature and cost effective system that actuates a robotic arm with the help of device commands derived from EEG signals [7].

2.3.2. **BCI achievements inside the PPU**

- In 2016, Razan et al controlled turning off and on the light and opening and closing the door by EEG signals.
- In 2018, Nibal et al Developed EEG and EMG based system for controlling robotic arm movements.
- In 2021, Iman et al controlled a wheel chair by EEG for eye movements.

Some of those achievements have a big value in helping patients with amputated limbs such as controlling artificial arms but also they are not basic where the amputated limb patient can live without an artificial arm, in addition such achievements are not helpful to the paralyzed people. Others such as writing words by a visual keyboard without voice or limb movements, opening simulated e-mail or operating a television, opening and closing the door and turning off and on the light are helpful to people with amputated limbs and paralyzed patients but they can live without doing them. Thus there is still a gap that there is no tool that takes care of the basic needs of the paralyzed patients that the patient can not live without them which are the food, water and medicine. So that our project will address this gap and provide a tool to the paralyzed people that carries out the task of meeting their basic needs on their behalf.




2.4. **Hardware Components Options**

We will show the different options of each component by comparing them in tables.

2.4.1. **To acquire brain signals we can use:**

- Electrocorticography (ECoG)
- Electroencephalography (EEG)
- EMOTIV EPOC headset

Table 1 : Comparison between ECoG, EEG and EMOTIV EPOC headset

	Electrocorticography (ECoG)	Electroencephalography (EEG)	EMOTIV EPOC headset
Illustrative image	 <p>Figure 2.2 : : ECoG procedure[8]</p>	 <p>Figure 2.3 : : EEG procedure[9]</p>	 <p>Figure 2.4: EMOTIV EPOC Headset[10]</p>
What is it?	It is one of the used electrophysiological sensors to sense brain signals	It is one of the used electrophysiological sensors to sense brain signals.	It's a tool that provides brain data. It comprises 16EEG electrodes(16 chanel), wireless USB and SDK.
Methodology in acquiring signals or brain data	By using electrodes placed directly on the exposed surface of the brain to get electrical activity from the cerebral cortex.	By using electrodes placed along the scalp to get brain signals.	By using it's EEG electrodes to sense the signals that are sent to the USB. Then the USB that is connected to the computer passes them to the SDK that exists in the computer. The SDK analyzes and classifies the signals and provides the

			signals class as brain data.
The used procedure in acquiring signals	It requires invasive procedure(which is a medical procedure that invades (enters) the body, usually by cutting or puncturing the skin or by inserting instruments into the body) to sense brain signals	It requires a noninvasive procedure (which is a procedure that does not require any skin incision or the insertion of any instruments or medical devices into the body) to sense brain signals.	It requires a noninvasive procedure to sense brain signals.
Availability	Unavailable in our college.	Available in our college.	Available in our college.
Output	It provides unprocessed unusable brain signals.	It provides unprocessed unusable brain signals.	It provides processed usable brain signals.

Chosen Design Option: EMOTIV EPOC headset.

We chose using EMOTIV EPOC headset since it is easy to use, noninvasive, and available in our college in addition to its ability to Provide processed usable brain signals.

2.4.2. To contain and use the SDK that analysis and translate the acquired signals we can use:

- Ordinary computer
- Raspberry pi

Table 2 : Comparison between Raspberry Pi3 and the ordinary computer

	Ordinary computer	Raspberry pi3
Cost	More expensive	It's a single board computer but it's Less expensive
Size	Larger size	Smaller size

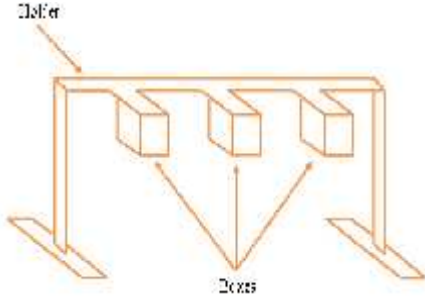
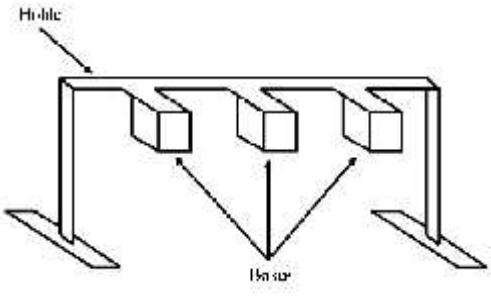
Chosen Design Option: Raspberry Pi 3

We chose Raspberry pi because it has a smaller size and it's less expensive than the computer.

2.4.3. To contain the food, water, and medicine we can use:

- Aluminum boxes
- Iron boxes

Table 3: Comparison between Aluminum and Iron boxes

	Aluminum boxes	Iron boxes
Illustrative image	 <p>Figure 2.5 : The hanging Aluminum boxes on a holder</p>	 <p>Figure 2.6 : The hanging Iron boxes on a holder</p>

Structure and place	Which are three Aluminum boxes that are hanging by a holder placed in the user's room. Each of those boxes will contain either food, water or medicine.	Which are three Iron boxes that are placed hanging by a holder placed in the user's room. Each of those boxes will contain either food, water or medicine.
Cost	Less expensive	More expensive
Weight	Less weight	Higher weight



Chosen Design Option: Aluminum boxes

We chose using the Aluminum boxes because they have less weight and less cost.

2.4.4. To identify the route that leads the robot to the boxes place we can use:

- PIR sensors
- IR sensors

Table 4 : Comparison between IR and PIR sensors

	Infrared(IR) sensors	Passive Infrared(PIR) sensors
Illustrative image	 <p>Figure 2.7 : IR sensor</p>	 <p>Figure 2.8: PIR sensor</p>

Description	<p>It is an electronic device that consists of an IR LED that emits an infrared radiation and is called emitter and an IR photodiode that senses the infrared radiation and is called detector[13].</p>	<p>It is an electronic device that consists of two slots, each slot has an IR photodiode(detector) that senses the infrared radiation with wavelength range 4 to 20 microns, and they are connected to a differential amplifier[13]. It is covered with plastic Fresnel lenses that focus the radiation on it.</p>
Work principle	<p>The IR emitter emits an infrared radiation with a certain wavelength. When that radiation falls on a white surface it will reflect, then the reflected radiation will fall on the IR detector. Therefore, the detector resistance will decrease so its output voltage will increase.</p>	<p>When no heat emitter body or radiation reflector object in the sensing area of the sensor, The two PIR detectors in the two slots of the sensor senses the same amount of radiation. Therefore, there is no difference between the sensed radiation of the detectors and no output voltage on the amplifier. If a heat emitter body or radiation reflector object enters the sensing area then the emitted radiation from those surfaces will intercept the first slot of the sensor, thus a positive difference will be generated in the output of the amplifier that causes generating a positive output voltage from the sensor. But if the body or the object left the sensing area, the radiation on the first slot will be low and the second slot of the sensor senses the radiation, thus the difference of the radiation on the two slots will be negative so the output signal will be negative</p>

		voltage.
How can we benefit from it?	<p>We will use two IR sensors, where:</p> <ul style="list-style-type: none"> ➤ When the right sensor falls on the drawn path that the robot will walk on, and the left sensor does not fall on that path, then the robot will go toward the right. ➤ When the left sensor falls on the path and the right sensor does not fall on the path, then the robot will go toward the left. ➤ if the two sensors fall on the path, the robot will stop. ➤ if the two sensors do not fall on the path, the robot will go forward. 	<p>We will use two IR sensors, where:</p> <ul style="list-style-type: none"> ➤ When the right sensor falls on the drawn path that the robot will walk on, and the left sensor does not fall on that path, then the robot will go toward the left. ➤ When the left sensor falls on the path and the right sensor does not fall on the path, then the robot will go toward the right. ➤ if the two sensors fall on the path, the robot will go forward. ➤ if the two sensors do not fall on the path, the robot will stop.
Sensing range	Up to 20 cm	25 cm to 6 meters
Sensitivity	Lower sensitivity. where it senses only the reflected radiation by a white surface.	Higher sensitivity. where it senses the radiation from any object that emits or reflects the radiation.

Chosen Design Option: Passive Infrared sensor(PIR)

We chose the PIR sensor because it is more sensitive to the radiation than the IR sensor. We will use two PIR sensors, one to sense the right path and another to sense the left path.

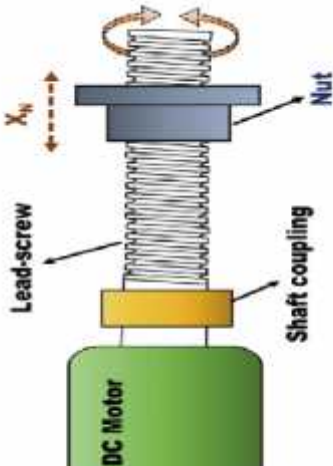

2.4.5. To draw the path that the robot will walk on to go to the boxes we will use a reflective papers

Where we chose to use a PIR sensor to follow the path, thus we need to draw the path to use a material that reflects Infrared radiation so that the sensor can sense it.

2.4.6. To make the robot take the required box from its place and puts the box on it and return the box to its place we can use:

- Lead Screw mechanism
- Horseshoe magnet

Table 5 : Comparison between Lead Screw and Horseshoe

	Lead Screw mechanism	Horseshoe magnet
Illustrative image	 <p>Figure 2.9: Lead-Screw mechanism</p>	 <p>Figure 2.10 : Horseshoe magnet</p>
Description	It is a mechanical mechanism that links between different machine parts and creates linear motion from rotational motion.	A Horseshoe magnet placed in the above base of the robot. it has an attractive force higher

	<p>As it is shown in Figure 2.9 It consists of :</p> <ol style="list-style-type: none"> 1. shaft coupling which is a mechanical part that connects the motor to another machine part (which is here the screw)to let that part turn with the motor. 2. The bearing, which is a mechanical part, contains a circular bore to contain a turning shaft(which is here the screw). it “reduces the friction and makes rotation more smooth” and “Protects the part that supports the rotation, and maintains the correct position for the rotating shaft”[14]. 3. Screw which is a mechanical turning shaft that connects to the part that turns in the machine. 4. and a nut which is a sliding mechanical part that carries the load that should go forward and backward. <p>The DC motor is fixed on the middle of the below base of the</p>	<p>than the attractive force of the magnet that is placed in the box cover.</p>
--	---	---

	robot and the nut attaches to a flat square aluminum piece that has little high edges and exists above the robot that we will call it the box carrier.	
How can we benefit from it?	When the motor turns in a certain direction , then the Lead-screw will turn with it, thus the nut will go up, therefore the box carries will go up also and carries the required box.	When the robot arrives under the required box, the magnet will attract the box and put it on the robot.
Availability	It's easy to obtain it, where it is available in the market.	It's hard to obtain a magnet with the required specification.
Reliability	High	Low

Chosen Design Option: Lead Screw mechanism


We chose the Lead Screw because it is easier to be available, easier to use, and has higher reliability than the magnet.

2.4.7. To move the robot we can use:

- Stepper motors
- Dc motors

Table 6 : Comparison between Stepper and DC motor

	Stepper motor	Dc motor

Illustrative image	 <p data-bbox="633 567 925 598">Figure 2.11 : Stepper motor</p>	 <p data-bbox="1063 577 1307 609">Figure 2.12: DC motor</p>
Controlling	Easily and directly controlled by microcontroller	Easily controlled by connected driver with microcontroller
Response time	Slow	Faster
Effect of Overloading	The stepper motor can be slipped if overloaded and the error cannot be detected.	If an overload occurs, it can be detected.

Chosen Design Option: DC motor



We chose the DC motor because it has faster response and the most important point is the effect of overload. where the robot might face overloading when carrying the box to the user. The DC motor can deal with that without problems occurring, but the Stepper motor will slip and that causes big problems. We will use four DC motors to make the robot move and one to turn the Lead-Screw.

2.4.8. To control DC motors we can use :

- L298N driver

➤ L2983d driver

Table 7 : Comparison between L298N and L2983d drivers

	L298N driver	L293d driver
Illustrative image	 <p>Figure 2.13: L298N driver</p>	 <p>Figure 2.14 : L29Sd driver</p>
What is it?	It is an electronic device that controls motor work. It receives control commands from a microcontroller then does the control process.	It is an electronic device that controls motor work. It receives control commands from a microcontroller then does the control process.
Efficiency	30–70%	50–90%
Drive voltage	5V-46V	4.5V-36V
Availability	It is previously available with us.	It is not available with us.
Weight	26 gm	35 gm

Chosen Design Option: L2983d driver

We chose the L298N driver because it is available previously with us and drives higher voltage than the L293d driver. We will use two L293d drivers, one to drive the motors that connect to the wheels and another to drive the motor that turns the Lead-Screw.

2.4.9. To supply the drivers with the required voltage to operate the motors we will use a 12 volt battery.

We will use two 12 volt batteries, one with 5 ampere current to supply the driver of the motors that are responsible for moving the robot, another with 7 ampere current to supply the driver that drives the motor that turns the Lead-Screw.



Figure 2.15 : 12 volt battery

2.4.10. To able the Robot to walk on the ground we will use wheels

2.4.11. To assemble the motors with the wheels we will use shaft couplings.



Figure 2.16 : shaft coupling

2.4.12.To control work of motor drivers and the sensors and organize them we will use Arduino mega:

Which is a programmable microcontroller that has several input and output pins that we can connect to the electronic devices that we would use to control their work. It has 54 digital I/O pins, 16 analog input pins, 5volt operating voltage and 40 mA DC current per I/O pin[21].

We chose to use the arduino rather than the raspberry pi itself to control the process because we use windows 10 operating system on the raspberry pi as we explain later, and the libraries that deal with the raspberry pi pins cannot install on windows. Moreover, using the arduino provides more safety to the raspberry pi, where the heavy loads that we will control may lead to heat the raspberry pi so it may be disabled.



Figure 2.17 : Arduino mega

2.4.13.To supply the raspberry pi with the required power we will use ASUS ZenPower power bank.

Which is a power container that we can charge it and charge from it using a USB cable. It has two USB outputs, the first output out 5volt with 1 ampere and the second output out 5volt with 3 ampere.

We chose to use it because we need a power source that can be carried, where the robot will move to bring the box.



Figure 2.18 : ASUS ZenPower power bank

2.5. Software options

2.5.1. To operate the raspberry pi we will use Windows10 operating system

Where we chose to use the raspberry pi computer and that computer must contain the EPOC EMOTIV SDK, thus we should choose the operating system that can operate on the raspberry pi architecture and at the same time can operate the EPOC EMOTIV SDK.

But the EPOC EMOTIV SDK operates only on windows systems. Therefore we should use a windows operating system to operate the raspberry pi, Which is now possible thanks to the WoR project that produced a windows10 system that operates on raspberry pi architecture. Therefore we chose to use the

Windows10 system, where it is the only existing operating system that can do both requirements.

2.5.2. To automatically sign in the raspberry pi user account we will use Autologon application

which is an application that when we install it and activate it on the computer it will be caused to auto login the user account that we determined to it on the computer.

2.5.3. To analyze the acquired brain signals we will use EPOC EMOTIV SDK.

Where we chose to use EPOC EMOTIV headset to acquire the brain signals, therefore we must use the EPOC EMOTIV SDK to analyze the acquired signals and obtain usable brain data. And we cannot analysis the signals that are acquired using EPOC EMOTIV headset by our own software, and that because there is no any hardware information about the EPOC EMOTIV headset, so we don't know how this headset work to obtain and process the signals, thus we cannot write a software deals with it.



Figure 2.19 : EPOC Control Panel

As it is shown in the above figure, the SDK software has an interface that we can deal with, which is the EPOC control panel. This control Panel has several tabs, each tab detecting certain signal classes.

The tab that will benefit from it in our system is the cognitive suit tab, which detects five signal classes, the first class is the class that is detected when the user thinks to move the box right, the second will be detected when the user thinks to move the box left, the third will be detected when the user thinks to push the box, the fourth will be detected when the user thinks to pull the box, and the last one will be detected when the user thinks to disappear the box,

In the cognitive tab there are five scroll down bars that we will use to choose the type of the signal that would be detected. and there is beside each scroll bar an empty box to enter the key that would be to trigger on the keyboard when the SDK detects the determined signal class in the box that is beside the box that contains the letter.

2.5.4. To receive the letters that are sent from the SDK Control panel to be triggered we will use OnScreen keyboard

Which is a software keyboard available in most operating systems that shows on the computer screen and we can use it to trigger on keys and hotkeys by the mouse clicks or some applications instead of the hardware keyboard.



Figure 2.20 : OnScreen keyboard in windows 10 system

2.5.5. To automatically activate the SDK control panel and On Screen keyboard we will use AutoHotkey programming language

Which is a “is a free, open-source scripting language for Windows that allows users to easily create small to complex scripts for all kinds of tasks such as: form fillers, auto-clicking, macros, etc”[23].

2.5.6. To write the program that controls the system we will use the python programming language.

We are going to use python programming language because it is easy to code where it is Object-Oriented Language, Free and Open Source and has large standard libraries that can help us to optimize our code. We will use the anaconda environment to run the python script because it is an integrated environment that interprets and compiles the python scripts, and sends the compiled script to the raspberry pi processor to execute it. But the other

windows python packages only interpret the script and produce an interpreted script that the raspberry pi processor cannot execute.

3. Chapter3: Design

3.1. Overview

This chapter shows the overall design of the robot and the way its components are connected to each other, it discusses the flowchart , the block diagram and schematic diagram of the system .

3.2. Brief description of the System

The main goal that we are working to achieve through our project is to build an assistant robot for paralyzed people that helps them to meet their basic hunger, thirst and pain needs by reading their thoughts(brain signals). So, we will use a BCI system to build that robot that acquires brain signals, analyzes them and translates them into commands that orient the robot to what should be done.

Our robot will be placed next to the patient and it will read the brain signals of the patient using the headset that the patient wears on his head. It will also determine certain signals among the signals that it reads where it can know from them if the patient is hungry, thirsty or feels pain. If it detects that there is a signal that signifies that he is hungry, thirsty or feels pain it will determine the path that leads to the desired place based on the detected signal, will go through the determined path to bring the box that contains the required need, after carrying the box it will go to serve the patient by going back from the same path that it came from and waits until the patient finishes, Finally after the patient finishes from his need, he will initiate a brain signal again, the robot will know that the service is done, then it will return the box to its place and it will return to its original place.

3.3. Mechanical and Hardware Design of the system

As shown in figure 3.1, the robot is in the shape of a cube and has four wheels. The lower and upper surfaces of the robot are made from wood, the raspberry pi, the arduino, batteries, power bank, and the drivers are placed on the robot's lower

surface. The motor that turns the Lead-Screw is fixed on the middle of the robot's lower surface.

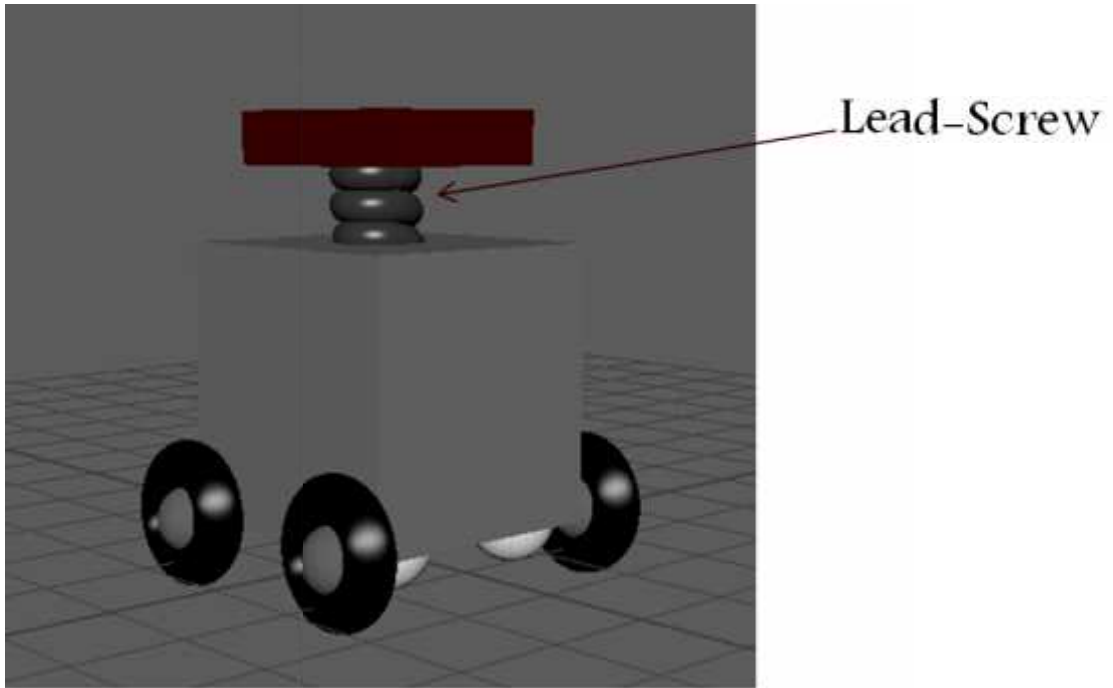


Figure 3.1 : The mechanical design of the robot

The Lead-Screw is connected to the motor by shaft coupling, and the Lead-Screw nut is attached to the box carrier that exists above the robot's upper surface as it is shown in figure 3.2 .

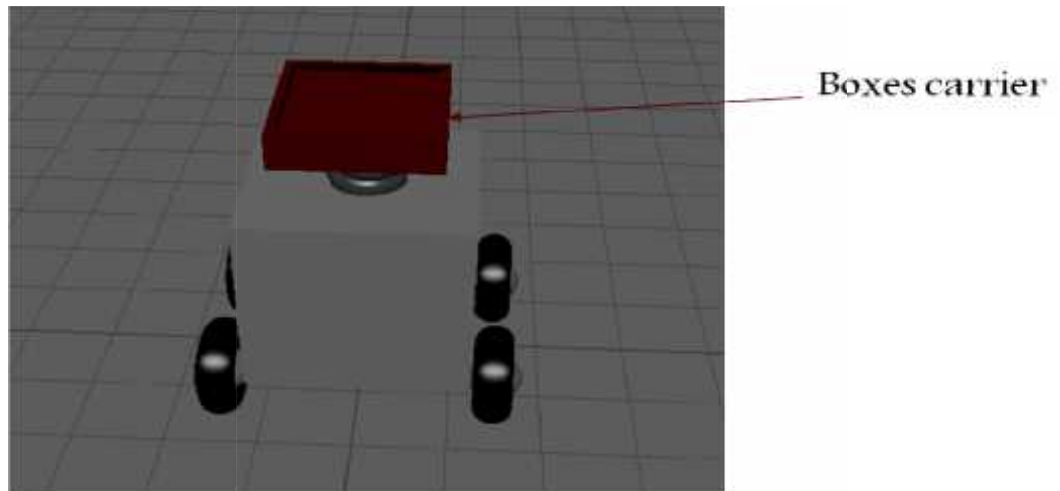


Figure 3.2 : The mechanical design of the robot

On the corresponding side as it is shown in figure 3.3 below, there are three hanging boxes by the holder. One of them for water, another for medicine and the third for food. Each box has a specific path that leads to it and a flat rectangular cover fixed to the holder horizontally and is 50cm from the ground. Each box is designed where it can cover by pushing it toward its cover and entering its cover in the hole that at the top of the box, and removing the cover by pulling it away from the cover.

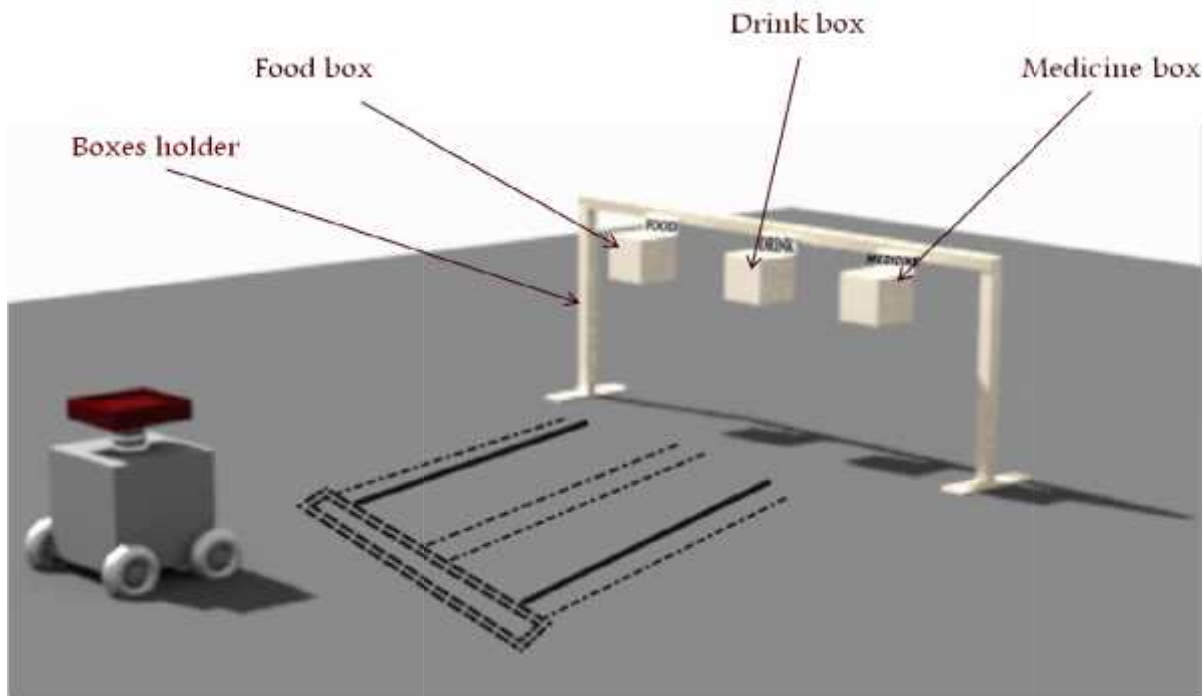


Figure 3.3 : The mechanical system design

The robot will reach the determined box and stand under it to carry it. Thereafter the box carrier will rise until its base collides with the box base then it stops rising . Then the box will settle on the robot. The robot will go back, pull the box away from its cover and go to the patient carrying the box and serve him.

After the robot finished servicing it should return the box to its place, so it will go and stand below the hanging cover of the box that it carries, then the box cover

will enter the box that the robot carries and the box will become hanging. After that the robot goes the box carrier down and returns to its original place.

3.4. Hardware setup

The robot parts are controlled by the Raspberry pi that is installed inside it as it. This raspberry pi connects to the USB that receives the brain signals from the Headset and passes them to the raspberry pi that connects to the arduino that takes the commands from the raspberry pi and arrives them to the goal device that is connected to it. And those devices that are connected to the arduino are the motor drivers that direct the motors to go to the determined path and the sensors that determine the appropriate path.

3.5. Block diagram for the robot system

Figure3.2 shows how the main components of the robot system are connected and the workflow of the system.

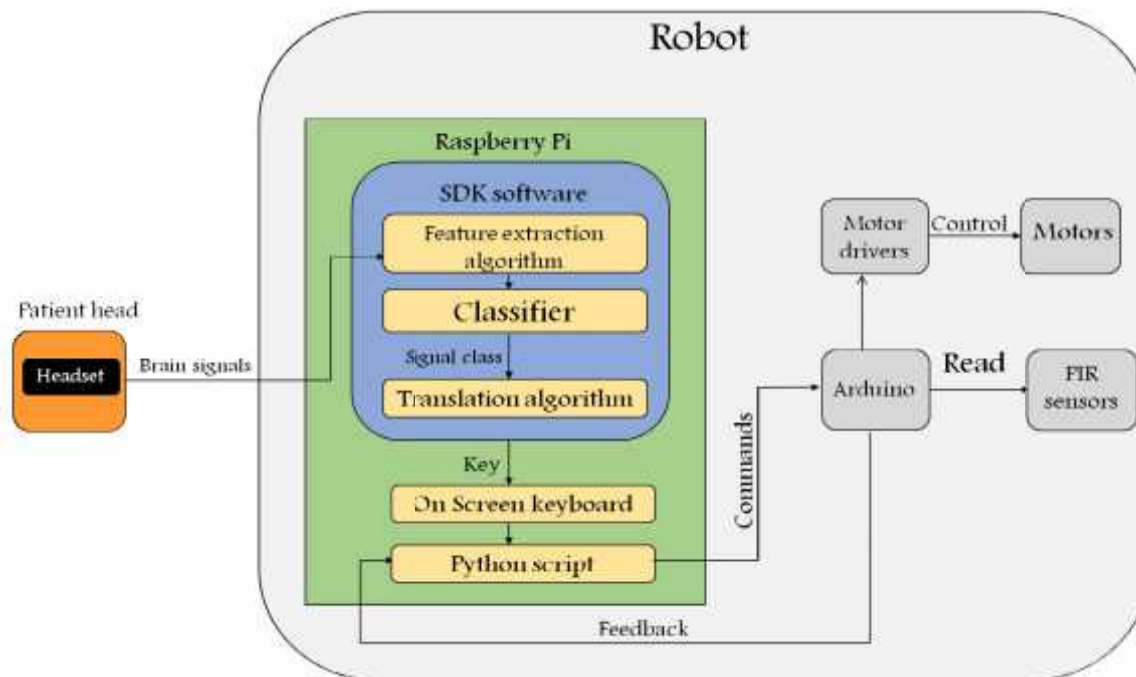


Figure 3.4 : System Block diagram

As figure 3.2 shows, the robot system consists of the EMOTIV EPOC Headset and the robot that contains Raspberry pi computer, Arduino microcontroller, and electrical parts to be controlled.

When the user wears the Headset on his head, the Headset senses the user's brain signals, processes them and sends them to the raspberry pi. The raspberry pi receives the signals from the Headset and passes them to the SDK software that exists in it. The SDK software analyzes the signals by feature extraction, classifies them by a classifier then provides the signal class as data. The signal class is translated by the SDK into a letter that was previously determined to the SDK to be triggered on the On Screen keyboard when it detects that class of signal . Another software that we will write in the raspberry pi waits until one of the determined letters is triggered, then the Raspberry pi sends commands through the Arduino to the motor drivers to control the motors and to the sensors to read data from them to determine the required path.

3.6. System Schematics

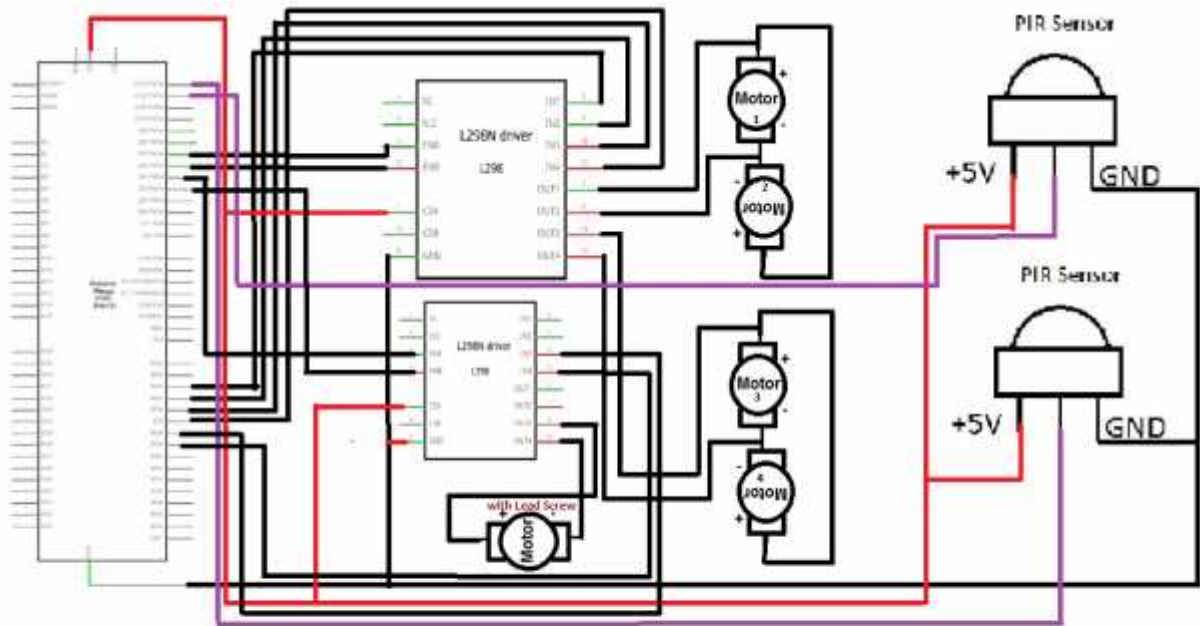


Figure 3.5 : System Schematic diagram

3.7. System programming Flowchart

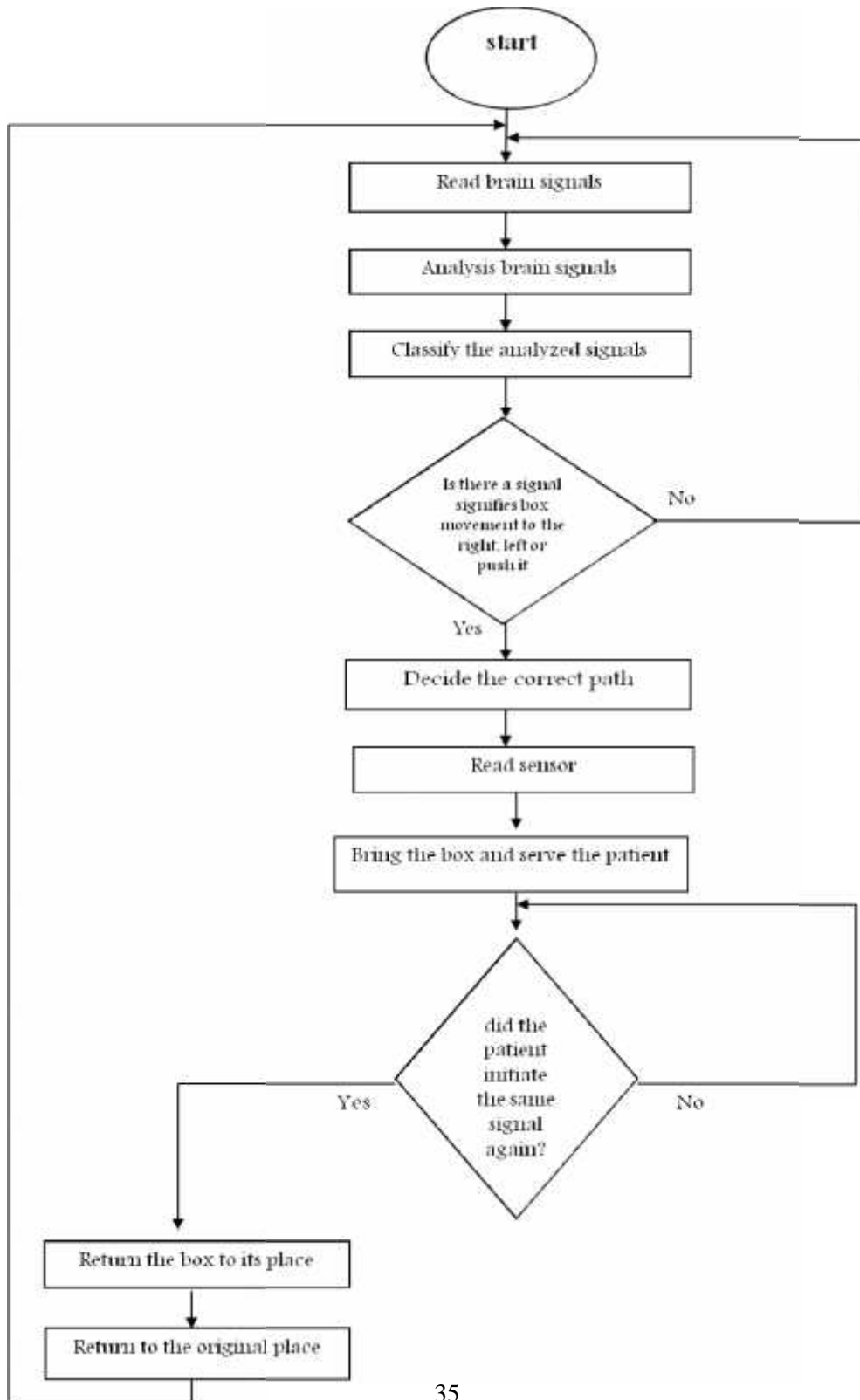


Figure 3.6 : System Flowchart

4. Chapter 4: Implementation

4.1. Overview

In this chapter we will clarify how we implemented our system so that at the end it is able to achieve its desired objectives. And to illustrate that process-the process of the implementation-it should be explained in detail how the hardware components connect with each other to operate the entire system. and the role of the software components to operate the system. So, this is what we will present here.

4.2. Brain signals acquisition implementation

That was done using the EPOC EMOTIV tool. The following steps clarify the implementation process:

- The EPOC EMOTIV Headset was worn on a head and turned on,
- The EPOC EMOTIV USB Flash was entered in the raspberry pi computer,
- The EPOC EMOTIV SDK on the computer was activated and prepared to receive the signals. The greenHeadset channels that are shown in figure 4.1 below are the channels that worked successfully with us.
- The EPOC EMOTIVE SDK was trained to classify the received signals from the Headset.

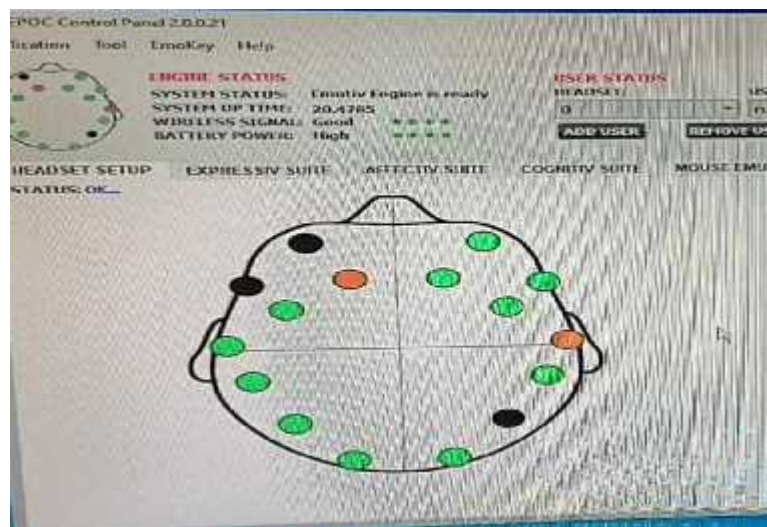


Figure 4.1 : The EPOC EMOTIV Headset channels that worked

4.3. Hardware implementation

This process included assembling system components to be as it is shown in the previous chapter in figure 3.3 and in the schematic diagram in figure 3.5.

Figure 4.2 and 4.3 show The final arrangement of the robot components after assembling them.



Figure 4.2 : The final arrangement of the robot



Figure 4.3 : The final arrangement of the robot

And below we will clarify the steps that we did to get that structure that shown in the previous figure.

- Firstly, the wooden robot structure was designed in a 30*30 cube form from the wood.
- After that, the motors that were responsible for the robot movement were connected to the bottom of the below surface of the robot structure. Then the wheels were assembled to them.
- Next the motor that was responsible for the Lead screw movement were fixed on the middle of the below surface of the robot structure. And the lead screw was assembled with the motor.
- The box's carrier is assembled with the Lead-Screw. All above assemblies are shown in figure 4.3.

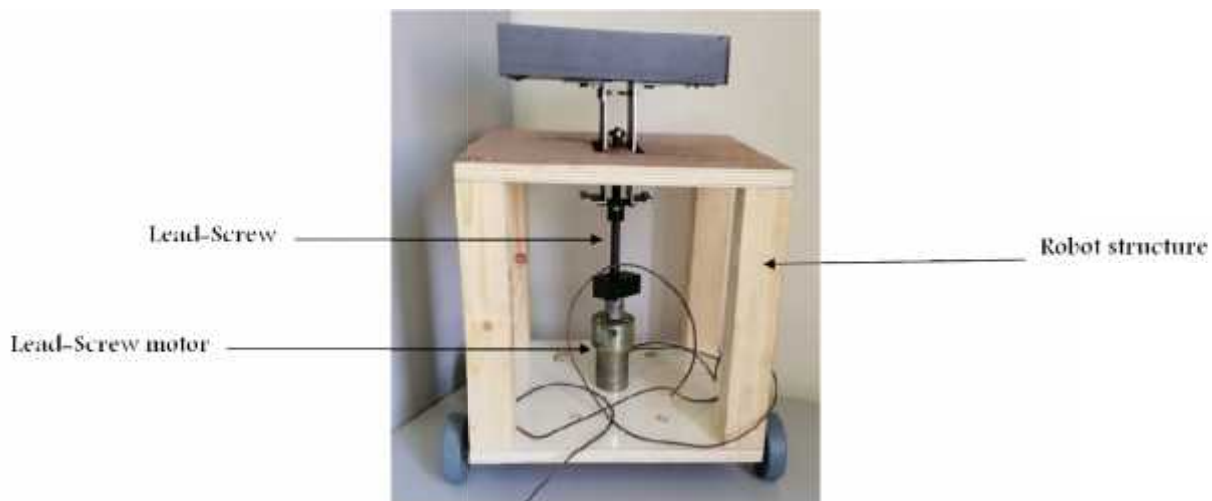


Figure 4.4 : the robot structure with motors, wheels, Lead-Screw and boxes container.

- Afterwards, the box holder and the boxes were designed as it is shown in figure 4.4.



Figure 4.5 : The boxes and boxes holder

- Then, the PIR sensors were placed on the left and the right sides of the front edge of the below surface of the robot structure as it is shown in figure 4.5.

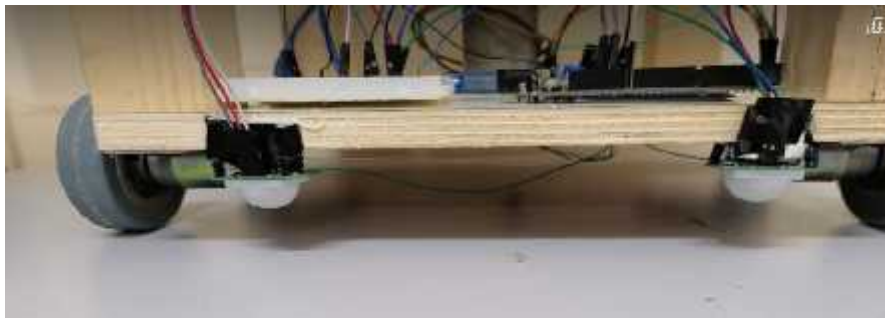


Figure 4.6 : PIR sensors fixing

- Thereafter, the Raspberry pi, the Arduino, the power bank, the drivers, and the batteries are placed on the below surface of the robot structure.
- Finally the raspberry pi, the Arduino, the power bank, the sensors, the motors, the drivers, and the batteries are connected to each other as it is shown in figure 4.6.

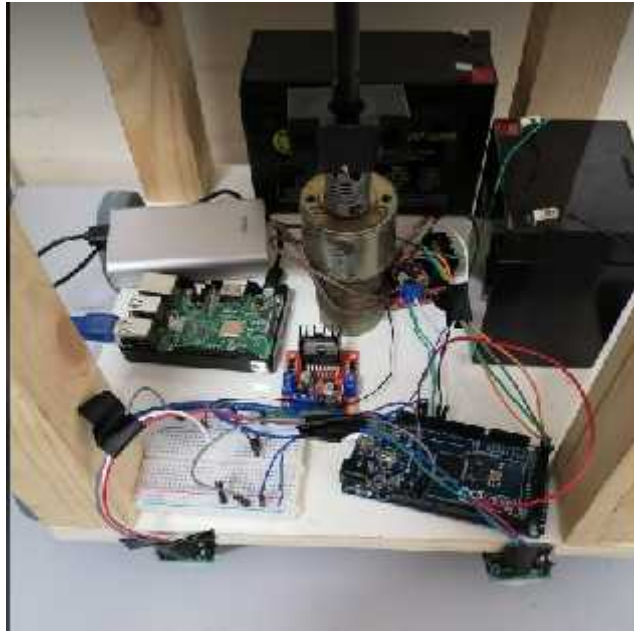


Figure 4.7 : The connected robot components

4.4. Software Implementation

In this section we will address the details of the software components. We will present each software module and clarify the role of each one in operating our system.

4.4.1. Raspberry pi implementation

Here is the process of preparing the raspberry pi for working, which includes:

- downloading a windows on raspberry imager from worproject.ml website.

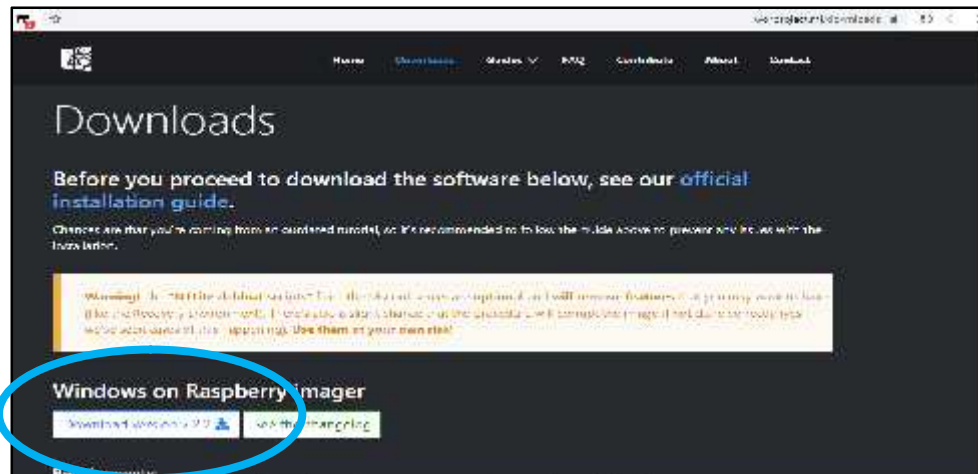


Figure 4.8 : Windows on raspberry imager

- downloading a copy of windows 10 operating system images that its build is Feature update to Windows 10, version 2004 (19041.330) arm64 from <https://uupdump.net/> website,



Figure 4.9: Windows on Raspberry images

- Extracting the files from the downloaded folders (*19041.330_arm64_en-us_core_447da2b8_convert.zip* and *WoR_Release_2.1.2.zip* folders)
- Opening the extracted folder (*19041.330_arm64_en-us_core_447da2b8_convert*)
- Creating the ISO file of the system by clicking on (*uup_download_windows.cmd*), which takes many hours to complete.
- After the ISO file is created the system images (ISO file) was copied on the SD card of raspberry pi using *WoR.exe* application which is existing in (*WoR_Release_2.1.2*) extracted folder,
- After that, the SD card was inserted into the raspberry pi and supplied the raspberry pi with the required power. And finally the entire windows10 system was ready to use on raspberry pi.

4.4.2. Arduino Implementation

In this stage we prepared the arduino to be ready to communicate with the raspberry pi to receive commands from it and send data to it. And we done that by:

- downloading arduino ide,
- opening the StandardFirmata sketch from File<Examples<Firmata<StandardFirmata

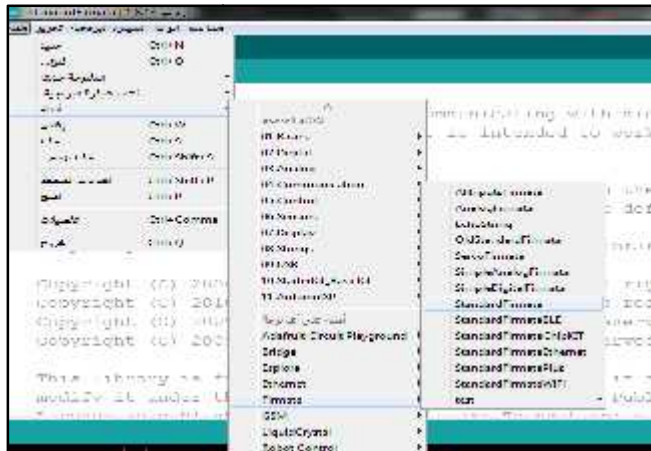


Figure 4.10 : StandardFirmata library

- and uploading StandardFirmata sketch on the arduino board.

4.4.3. Implementation of automatically signing in the user account

That was done by following steps:

- Downloading Autologon application
- extracting the files from the downloaded folder
- opening Aoutologon.exe application
- Agreeing the license agreement
- fulfilling the popped window and clicking on enable button

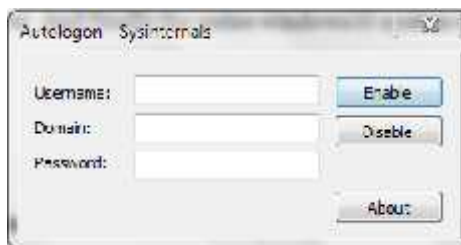


Figure 4.11 : Autologon application

4.4.4. Signal Analysis Implementation

That was done by installing on the raspberry pi a pre-installed software development toolkit (SDK) on the CD which comes with EMOTIV EPOC headset when buying it from the source

4.4.5. Implementation of automatically activation the SDK

This is done by the following:

- installing AutoHotKey programming language.
- writing an AutoHotkey script that :
 - automatically runs and activates the SDK control panel,
 - automatically enters the user name in the box that pops when the control panel is activated,
 - automatically moves between control panel tabs to go to the cognitive suite tab
 - and automatically determines for each signal a letter that would be triggered on the keyboard when that signal is detected.
- and making the script run as administrator

4.4.6. Implementation of automatically running the python script

This is done by copy and paste the script in the startup folder that is accessed to it by “*C:\Users\lenovo\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup*” path. So after the raspberry pi operates and logs in in the user account the python script that is placed in the startup folder will run automatically and the execution of the script will begin.

4.4.7. Implementation of control the robot when a specific signal is initiated by the user

Let us remember together some previously mentioned points before going deep in this section.

The process of control of the robot is started from the user's brain, when he thinks to move the box right, left or push it, then a specific signal will be initiated in its brain. That signal will be sensed and processed through the headset that the user wears that sends the signal after processing it to the robot. Now , robot work has started, which is bringing the required box to the user and returning it after the user finishes from it.

But how does the robot work? The answer will be the body of that section.

After supplying the raspberry pi that is placed on the robot with the power bank, it will turn on, automatically sign in the user account and automatically run the python script. then the script will do the following:

- refreshes the raspberry pi several times to speed up the execution of the script,
- runs the AutoHotkey script that activates and ready the SDK control panel. Then the SDK will be ready to receive the signals from the headset. So that when the SDK receives a signal, it analyzes it, determines its class and triggers the key on the OnScreen keyboard that is determined before to be triggered when it detects that class of signal.
- executes the first module which is `main()` module that listens to the keyboard triggering and determines which of the others modules should be executed based on the triggered key by the SDK.
- Once the SDK trigger on a key for the first time, the script executes the second module which is `Bring_The_Box()` module that orients the robot to the proper path to bring the required box and serves it to the user based on the signal class,
- Once the SDK trigger on the same key again, the script executes the third function which is `Return_The_Box()` module that orients the robot to return the box to its place and then return beside the user.

Now we will address in more detail each module and function that we used in executing above tasks.

4.4.7.1. `Loop()`function

Which executes `keyboard.press('f5')` function that presses on the F5 hotkey that refreshes the computer desktop when it is pressed.

4.4.7.2. `Keyboard.press("win")`and `keyboard.press("m")`functions

Which are minimizing the activated cmd window that executes the python script to the taskbar, so that to move the activation to the desktop.

4.4.7.3. `Keyboard.press('A')` and `keyboard.press("enter")`functions

Then press on the AutoHotKey file that nemed "A.ahk" and place it on the desktop and run it.

4.4.7.4. AutoHotKey file

It executes:

- `Run("path")` function: that runs the EPOC EMOTIV Control panel and activates it.
- `Send("key")` functions: that sends to the activated window(which is the EPOC Control panel) the key that is determined in its parameter. Therefore the determined key will be triggered, thus it will do a certain clicking or moving on the activated window. Figure 4.8 shows the keys that is sent to the EPOC Control panel so that leads to :
 - enter the user name by clicking on the enter hotkey
 - move to the cognitive suit tab.
 - and determine the letters that would be triggered when detect certain signals.

```

Run "C:\Program Files (x86)\Emotiv EPOC Control Panel v2.0.0.21\Application
Sleep 90000
Send {Enter} ; To enter the headset user name
sleep 5000

Loop, 7 ; To move to the first tab
[
Send [Tab]
Sleep 3000
]

Loop, 3 ; To move to the cognitive suit tab
[
Send [Right]
Sleep 3000
]

Loop, 17 ; To move to the first key box
[
Send {Tab}
Sleep 3000
]
Send [Space] ; To show the text box that the key is written in it
Sleep 3000

```

Figure 4.12 : Autohotkey script

4.4.7.5. `main()` module

Which contains:

- `keyboard.read_key("key")` function:

Which waits until the determined key in its parameter is triggered then it will store the triggered key. That key will be triggered by the SDK when it detects a certain signal.

4.4.7.6. **if() statements:**

Which decides whether to execute the `Bring_The_Box()` module or the `Return_The_Box` module based on the triggered key by the SDK .

4.4.7.7. **Bring_The_Box and Return_The_Box modules**

Which contain:

- **DC motors** modules: Which operates the motors, directs them toward the right, the left or forward based on the triggered letter.
- **PIR sensors** module: Which controls motors movement so that the robot walks on the path that leads to the required box.
- **Carry Box** module: Which operates the Lead-Screw motor to go up and down.

5. Chapter5: Validation and Results

In this chapter we will illustrate the testing process of each component in the system and the overall operation of the system and its software. We will discuss the results that obtained from the system.

5.1. Hardware Testing

In this section we are going to discuss the testing process of each hardware component.

5.1.1. Arduino Testing

At first we checked and ensured that the Arduino was working correctly with the raspberry pi, so we did many tests for that. First we run on the raspberry pi a simple python code that blinks a built-in LED on Arduino, by this way we made sure we can use the Arduino as a linkage between the raspberry pi and the other system components with no problems. Then we tested the input pins by connecting the PIR sensor to the Arduino and running the python code on the raspberry pi that reads data from it.

5.1.2. EPOC EMOTIV Headsettesting

We checked the work of the headset to make sure that it was working as it should, first it was placed on the head at the appropriate way to get the signals, then the USB connection that receives the signals was connected to the Raspberry Pi. After that we activated the EPOC Control panel and its settings were set. The headset training was done to be able to extract the signal class when the user thinks to move the box right, left and push it. Then we checked the results by typing the letter R to be triggered when the user thinks to move the box to the right,

also we typed the letter L to be triggered when the user thinks to move the box to the left, and we write a python script that turns on a led when the letter R is triggered and turns off it when the L letter is triggered, and this was done successfully.

5.1.3. DC motors and drivers testing

We used DC motors with wheels to move the robot and we checked the motors by connecting them to the Arduino through the drivers as shown in figure 5.1.3.1. And then we write a python script on the raspberry pi to move them clockwise (Forward) and counterclockwise (Backward). Then we noted its movement through the movement of the wheels. And so on the DC motors with wheels and drivers were checked at the same time.

5.1.4. Lead Screw mechanism testing

We checked the work of the Lead-Screw mechanism by connecting the Lead-Screw nut to a piece of metal to fix the nut and then connecting the screw with a DC motor to move it. We connected the motor with its own driver and then provided it with the necessary voltage by the betray. The result was that the screw converted its rotational movement that resulted from the rotation of the motor into a linear movement of the nut, which raised the metal piece that connected to the nut. And when the voltage source was reversed on the motor, then the linear movement of the nut was reversed towards lower.

5.1.5. PIR sensors testing

We checked the PIR motion sensors to make sure that they were working properly. We connected them to the Arduino, and then wrote and ran a python script that printed the sensor readings. The readings were 0 (False) if the movement was not recognized and 1 (True) if the movement was recognized (hand motion). And the results shown in the figure showed that the sensor was working correctly.

5.1.6. System overall Test

To ensure that the system works in an integrated manner with its three parts: the headset, the robot and the platform of the boxes, we written a python script so that when we trigger the R letter, the robot moved in the right path. and When it reached at the end of the path, the screw raised the box container, then the box was attached to the container, then the robot returned from the same path. When the service ended, we triggered the letter R again, so the

robot went again from the same path that followed it to bring the box to return the box to its place. When the path ended, the screw moved in reverse, after that the box container was lowered and the box settled in its place. Finally, the robot returned to its starting point, and waited for further signals.

5.2. Software Testing

In the previous section of testing hardware components we discussed the related software testing. In this section we will discuss what is left, which is the testing of pyFirmata library.

5.2.1. Pyfirmata library testing

To test the effectiveness of the pyFirmata library, we uploaded the library to the Arduino and then made two tests:

- Firstly, we ran a Python script that controlled the LED lighting embedded in the Arduino and it worked successfully.
- Secondly, we ran a Python script that runs a motor that is connected to the Arduino and we control its movement through the Python script.
- Third, to test the Input data in the pyFirmata, we run a Python code that reads the PIR sensors connected to the Arduino and prints the results of the readings and it was completed successfully.

5.2.2. AutoHotkey testing

To test the effectiveness of the AutoHotKey script in activating the SDK, we written an AutoHotKey script that runs the EPOC Control panel, clicking on its buttons, and navigating between its tabs, and that was done successfully

5.3. Results

Enabling people who are unable to meet their needs without others help to meet their basic needs independently and giving them a kind of self-confidence is our motivation in this project, which aims to create a robot that stands next to the patient and responds to his thoughts when the patient feels hungry, thirsty and pain by reading his brain signals using the Headset that the patient wears. Where if the patient thinks in moving the box right then the robot recognizes that the patient is hungry and brings the food box to his, if the patient thinks in moving the box left

then the robot recognizes that the patient is thirsty and brings the drink box to his, and if the patient thinks in moving the box left then the robot recognizes that the patient feels pain and brings the medicine box to his.

The implementation of such a robot with such capabilities is not easy to some extent, because its work is based on the brain signals that are sensed by the EPOC EMOTIV Headset. To benefit from the sensed brain signals by the EPOC EMOTIV Headset we must use EPOC EMOTIV SDK, and that what formed a challenge to us, because the EPOC EMOTIV SDK only installs on windows operating systems, and we use raspberry pi computer that operates by Linux operating system instead of the ordinary computer, therefore we failed at the beginning to install the EPOC SDK on raspberry pi. But after a lot of failed attempts in installing EPOC SDK on Linux and thanks to the WoR project we were finally able to install windows 10 system on raspberry pi. Thus, we were able to install EPOC SDK to raspberry pi and benefit from the sensed brain signals.

Most results of achieving our objectives were successful. The only part that didn't work appropriately is the ability of the robot in determining and walking on the appropriate path, where we used at the beginning IR sensors to make the robot determine its route, and after a lot of failed attempts we recognized that we used the wrong sensors. And the sensors that work right need a certain material that reflects the infrared radiation, but the time has elapsed after we got the right sensors. The below table shows the final project results.

Table 8 : The results of the project

Objective	Result
Acquire brain signals	Successful
Analysis the acquired signals and get brain data	Successful
Make the robot walks	Successful
Make the robot knows where it	Failed

should go	
Make the robot takes the box from its place and carries it	Successful
Make the robot return the box to its place	Successful

Conclusion and Future work

Through this project, we have sought to help the disabled people by providing them some services. And we have been able to do that thanks to the modern technology that entered every life field. Where We have benefited from BCI technology to achieve our goals.

The principle on which the project is based is the presence of a robot capable of movement and equipped with a Bluetooth feature to receive the brain signals of the patient through the headset that the patient wears, and equipped with a computer to recognize the received signals. Then it makes an appropriate decision to bring the service that suits the patient's needs.

At the end of the project, and despite the challenges we faced from the software and hardware devices, the project goals were set at the beginning were achieved partially.

As for the expected future work from the project, first: Building and designing an intelligent robot with localization capability that meets a lot of patient's needs, not being limited only to three needs. And supporting it with additional features that help the patient, such as issuing a voice to the patient's needs and sending a notification to the patient's relatives' phone.

References

- [1] Shih, J., Krusienski, D., & Wolpaw, "Brain-computer interfaces in medicine," <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3497935/>, March, 2020.
- [2] Somayaji, Julie Thorpe Paul C. van Oorschot Anil Somayaji Anil, "Pass-thoughts: Authenticating With Our Minds.," DBLP, 2005.
- [3] J. J. Vidal, "Toward direct brain-computer communication," PubMed, 1973.
- [4] Wolpaw, Jonathan R, and Dennis J McFarland, "Control of a Two-Dimensional Movement Signal by a Noninvasive Brain-Computer Interface in Humans," www.ncbi.nlm.nih.gov/pmc/articles/PMC535103/, 21 Dec. 2004.
- [5] Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH; Branner A, Chen D, Penn RD and Donoghue JP, "Neuronal Ensemble Control of Prosthetic Devices by a Human with Tetraplegia.," pubmed.ncbi.nlm.nih.gov/16838014/, 2006.
- [6] Krusienski, Dean J, and Jerry J Shih, "Control of a Visual Keyboard Using an Electrographic Brain-Computer Interface," Neurorehabilitation and Neural Repair, U.S. National Library of Medicine, May 2011.
- [7] Sunny, T.D., et al., "Robotic Arm with Brain – Computer Interfacing.," 9 July 2016.

- [8] "Electrocorticography," <https://en.wikipedia.org/wiki/Electrocorticography>, 2020, December 06.
- [9] Siuly, S., Li, Y., & Zhang, "Electroencephalogram (EEG) and Its Background," https://link.springer.com/chapter/10.1007/978-3-319-47653-7_1, 1970, January 01.
- [10] "Why We'd Love to Try the Emotiv EPOC," <https://lucid.me/blog/wed-love-try-emotiv-epoc/>, 2016, September 25.
- [11] "Metal Stand with 3 Wooden Boxes," <https://sapphiresecrets.co.uk/products/metal-stand-with-3-wooden-boxes>.
- [12] "How to Cover Boxes with Contact Paper," <https://www.homeyohmy.com/cover-boxes-contact-paper/>.