**REPUBLIQUE ALGERIENNE
DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

**UNIVERSITE LARBI TEBESSI - TEBESSA
FACULTE DES SCIENCES ET TECHNOLOGIES
DEPARTEMENT DE GENIE ELECTRIQUE**

MEMOIRE

DE FIN D'ETUDES POUR L'OBTENTION DU DIPLOME DE MASTER EN

SPECIALITE

INSTUMENTATION

THEME

Conception et réalisation d'une application Robotique (Praying Robot)

Présenté par:

- HADFI Karima

Devant le jury :

Président	**:**	**LEMITA Abdallah**
Encadreur	**:**	**GOUDER Soraya**
Examinateur	**:**	**KHEMAISSIA Seddik**

Année Universitaire 2019 / 2020

بسم الله الرحمن الرحيم

Acknowledgement

Thanks **ALLAH** for giving me the inspiration to made the praying robot and giving me all ability for that

Thanks **mom** and **dad** for helping me to get the master sheepskin and for believing in my ability. Thanks **sisters** and **brothers** for helping me and believing in me. Thanks my friends whose helped me and trust my ability

I would like to express my gratitude to my professor **Mz Soraya GOUDER** for her patience, her availability and especially her judicious advice, which helped fuel my thinking she trusts in my capacities and I would like to thank her a lot for this

I would like to thank **Mr KHEMAISSIA** and **Mr LEMITA** for accepting to be my jury. And they were the best jury ever

I would like to thank my **professors**, who teach me in the

University of Tebessa

SUMMARY

Figure List

Table list

GENERAL

INTRODUCTION

General Introduction

Robotics is a vibrant field which grows in importance from year to year. It is also a subject that students enjoy at all levels from kinder garden to graduate School.

The aim of learning robotics varies with the age group:

- For young kids, robots are an educational and performal toy
- For students in middle and high schools, robotics can increase the motivation of students to study STEM (Science Technology Engineering Mathematics),
- At the introductory university level, students can learn how the physics, mathematics, and computer science that they study can be applied to practical engineering projects;
- Finally, upper level undergraduate and graduate students prepare for careers in robotics.

The life in those days is so easy thanks to the robots around the word, in every place there is a lot of robots with deferent forms, sizes and tasks, for example when we go to the toys market we will find a huge number of robots that our kids like to buy them, because the idea to see some toy moving it is funny and that makes us ask a question. Why we should not made a robot which does something different? Something more exciting? We answered these questions: by realizing a praying robot; it is a brand-new robot and the idea to make a robot who can pray and move like a real human is a great idea, this latter makes us asking a new question: is it possible?

In this memoire, you will find the answer by yourself.

Praying robot is a religious type of robots…it's doing the five pray of our religion Islam in the real time of pray or by remote control any time we want to.

The memoire contains three distinct parts:

In the first chapter, we give a brief history about robotics and based on the three laws that control the grow of robots. Also we cited different types of robots.

The second chapter contains all the tools we used to realize our praying robots.

The last chapter include the results got from our project. It is rich with scripts and figures presented praying robot.

Finally, we conclude our project and give some suggestions for eventual other functions.

CHAPTER 1

WHAT IS

ROBOTICS?

1.1. Introduction

The idea to design and build some sort of being, or devices that could carry out repetitive or heavy tasks, thus relieving men from this burden, dates back to ancient times. Since the Greek-Hellenistic age some of these devices, which were named "automata", have been designed and created by ingenious inventors, belonging to several different civilizations through the centuries. The term automata mainly refer to human-like devices, while the term robot has a more general meaning. [1]

Robotics is science of designing or building an application of robots. Simply, robotics may be defines as "the study of robots". The aim of robotics is to design an efficient robot.

It is needed because:

- Can do repetitive tasks and work with accuracy
- Can work in hazardous/dangerous temperature with fast speed. [2]

1.2. Robotic history

1.2.1. Three lows of robotics

The origin of the term robot is placed in more recent times namely, it comes from the Czech word "robota", meaning heavy work or forced labour .the introduction of this term is due to the Czech writer KarelČapek 1890-1938, who used it for the first time in 1920 in his novel R.U.R (Rossum's Universal Robots). [1]

The word "robotics", used to describe this field of study, was coined accidentally by the Russian –born , American scientist and science fiction writer, Isaac Asimov(1920-1992) in 1940s.[2]

- In 1495 Leonardo DaVinci designs a mechanical device that looks like an armored knight. The mechanisms inside '' Leonardo's robot '' are designed to make the knight move as if there was a real person inside. Inventors in medieval times often built machines like Leonardo's robot to amuse royalty.[3]

Figure 1.1: Leonardo Humanoid robot with internal mechanisms [10]

- In 1738 Jacques de vaucanson begins building automata in Grenoble, France. He builds three in all. His first was the flute player that could play twelve songs. This was closely followed by his second automaton that played a flute and a drum or tambourine, but by far his third was the most famous of them all. The duck was an example of vaucanson's attempt at what he called ''moving anatomy'', or modeling human or animal anatomy with mechanics. ''the duck moved, quacked, flapped its wings and even ate and digested food. [3]

- In 1770 Swiss clock makers and inventors of the modem wristwatch pierre jaquet-Droz and later joined by his son Henri-Louis jaquet-Droz start making automata for European royalty. They create three dolls, each with a unique function. One can write, another plays music, and the third draws pictures.[3]

Figure 1 2 The doll that can write [11]

-In 1940 Issac Asimov produces a series of short stories about robots starting with *A Strange Playfellow* (later renamed *Robbie*) for supper science stories magazine. The story is about a robot and its affection for a child that it is bound to

protect. Over the next 10 years he products more stories about robots that are eventually recompiled into the volume I, Robot in 1950.

Asimov is generally credited with the popularization of the term'' robotics'' which was first mentioned in his story *Runaround* in 1942. But probably Issac Asimov's most important contribution to the history of the robot is the creation of his three Laws of robotics

1. **First law**: Robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2. **Second law:** A robot must obey the orders given it by human beings except where such orders would conflict with the first law.

3. **Third law:** A robot must protect its own existence as long as such protection does not conflict with the first or second Law.

4. Asimov later adds a ''zeroth law'' to the list:

Zeroth law A robot may not injure humanity, or, through inaction, allow humanity to come to harm [3]

Since the introduction of these laws, robotics grows and has a new revolution on its history. This one is regrouped in following points:

- In 1967 The first industrial robot in Europe, a Unimate was installed at Metallverken, Uppsland Väsby, Sweden.[4]
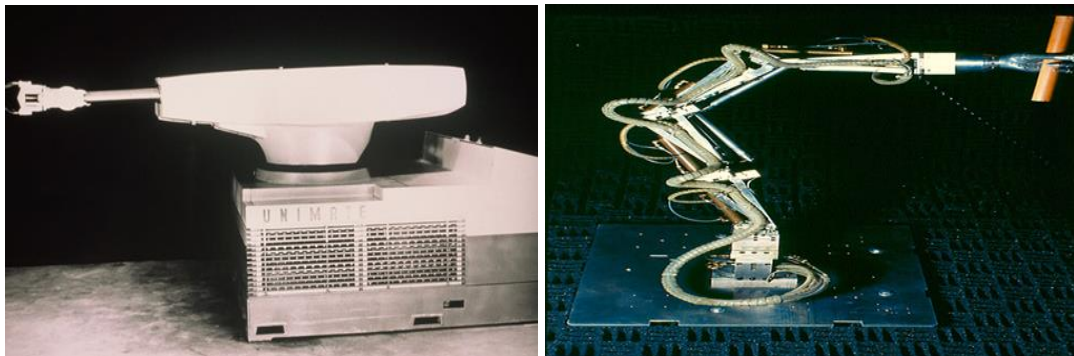
Figure 1.3 : Unimate II and Tentacle arm (In 1968 The octopus-like Tentacle Arm was developed by Marvin Minsky [4])

- From 1973, robots progressed quickly, KUKA moves from using Unimate robots to developing their own robots. Their robot, the famulus was the first robot to have six electromechanically driven axes.[4]

Figure 1.4: KUKA famulus [4]

- In 1999 Honda debuts the P3, the fruit of its decade long effort to build a humanoid   robot [3].

Figure 1.5:P3 robot of Honda [5]

- In 1997 the first node of the international Space Station is placed in orbit. Over the next several years more components will join it, including a robotic arm designed by Canadian company MD Robotics.[3]

The Pathfinder Mission lands on Mars. Its robotic rover Sojourner rolls down a ramp and onto Martian soil in early. It continues to broadcast data from the Martian surface until September.[3]
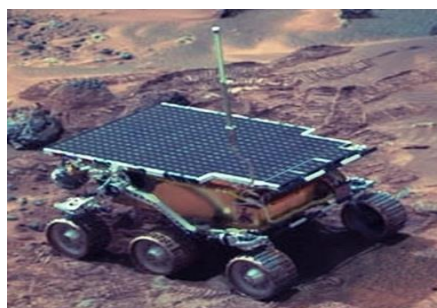
Figure 1.6: The Pathfinder Mission lands on Mars [12]

- In 1999 SONY releases the AIBO robotic pet [3].

Figure 1 7 : The AIBO robotic pet [5]

- In 2000 Honda debuts new humanoid robot ASIMO.[3]
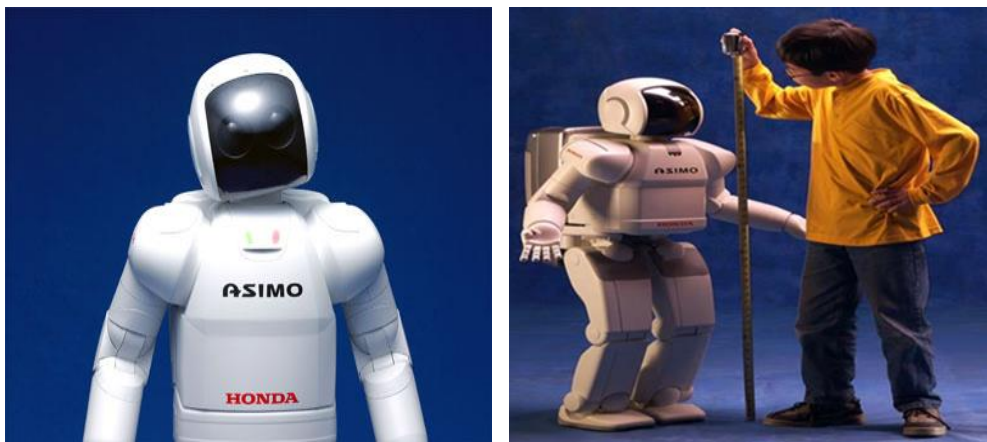
Figure 1.8: Honda's robot Asimo [5]

- In 2001 , the FDA clears the CyberKnife to treat tumors anywhere in the body.[3]

Figure 1.9: The CyberKnife [5]

- In 2003 NASA launches both the ''Spirit'' and ''Opportunity'' Mars exploration robot rovers.[3]
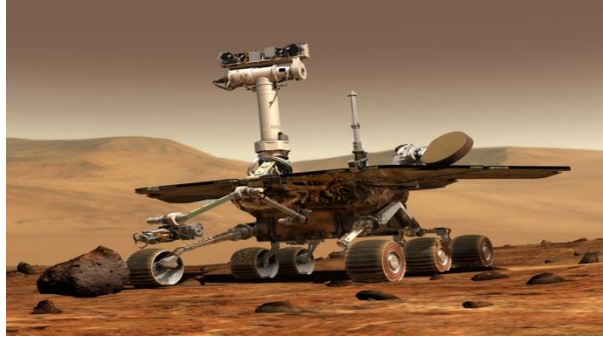
Figure 1.10: NASA Mars Rover [5]

SONY releases the AIBO ERS-7, it's 3rd generation robotic pet.[3]

Figure 1.11: The AIBO ERS-7 of SONY [13]

- In 2006 The second generation of LEGO MINDSTORMS is launched, spawning a new generation of robotics enthusiasts at home and in schools.[3]

Figure 1.12: LEGO MINDSTORMS [14]

- Motoman, Japan, launched super speed arc welding robots which reduces cycle times by 15%, the fastest welding robots in existence in 2007.[4]

Figure 1.13: Motoman EA1400N [4]

- In 2008 FANUC, Japan, launched a new heavy duty robot with a payload of almost 1,200 Kg.[4]

Figure 1.14: FANUC m2000 iA [4]

- In 2009 Yaskawa Motoman, Japan, introduces control system to sync up to 8 robots.[4]

Figure 1.15: Motoman multiple robot [4]

- In 2010 KUKA (Germany) launched a new series of shelf-mounted robots (Quantec) with a new controller KR C4.[4]

Figure 1.16: KUKA Quantec Konsol RGB 002 [4]

- In 2011 R2 is the first humanoid robot in space. Initially R2 was deployed on a fixed pedestal inside the ISS. Next steps include a leg for climbing through the corridors of the Space Station, upgrades for R2 to go outside into the vacuum of space.[4]
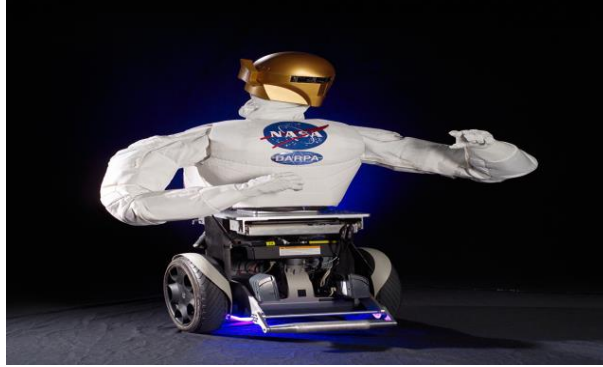
Figure 1.17: R2B robonaut on wheels [4]

- Kirobo is Japan's first robot astronaut, developed by University of Tokyo and Tomotaka Takahashi in 2013.

Its name mixes the words '' robot'' and ''kibo'', which is Japanese for '' hope''. (Kibo is also the name of the Japanese research module on the space station). [5]

Figure 1.18: Kirobo robot [6]

- hitchBOT, a robot created by a team of Canadian researchers in 2014, ''hitchhikes'' across Canada and Europe on a mission to explore cultural attitudes toward social robotics. hitchBOT is destroyed in philadelphia in 2015. [12]

Figure 1.19: HitchBOT [12]

- The robot Sophia is developed by HONG KONG based company Hanson Robotics in 2015.

Sophia is simultaneously a human-crafted science fiction character depicting the future of AI and robotics, and a platform for advanced robotics and AI research. [5]

Figure 1.20: Sophia robot [5]

- Atlas is a bipedal humanoid robot primarily developed by the American robotics company Boston Dynamics in 2016, with funding and oversight from the U.S. Defense Advanced Research Projects Agency DARPA. The 1.8 meter 6ft robot is designed for a variety of search and rescue tasks.[5]

Figure 1.21: Atlas robot [5]

- At IROS 2017, Honda revealed a prototype for a disaster relief robot in 2017, the 'E2-DR', who has the ability to grasp bars, adjust its hips and walk in a human-like manner.[7]

Figure 1.22: The 'E2-DR'[7]

- The Aeolus Bot is one of the robots moving around the floors of the Las Vegas showrooms at CES in 2018, and it's got an impressive list of capabilities: it can hand objects to you, vacuum the floor, clean away objects by recognizing them and putting them in the right place, and even act as a security guard.[8]

Figure 1.23: The Aeolus Bot [8]

- In 2019 Samsung released three robots at CES 2019: Bot Retail, Bot Care and Bot Air. Bot retail is the biggest of the three with large front display and basic shelving system at the back allowing it to deliver food or other items.[7]

Figure 1.24: Bot retail of Samsung [7]

- In this year Boston dynamics spot robot dog helps doctors treat COVID-19 (corona virus) patients.[7]

Figure 1.25: The dog robot daring corona virus time [7]

- The newme telepresence robots were designed by ANA group and works as remotely controlled, customizable avatars. During the virtual graduation the

robots, which were dresses in typical graduation caps and gowns, used tablets in place of the face allowing the students to see one another. [7]

Figure 1.26: Graduated robots [7]

1.3. Types of robots

Robots are primarily differentiated based on two categories: use and movement. Of course, there is a great deal of overlap in many of these categories; drones, for example, can be classified as either aerospace, consumer, or exploration. [12]

1.3.1. Industrial robots

Industrial robots often comprise the most basic form of machine stationary or semi-stationary device that executes a repetitive task. These robots are generally some of the least intelligent due to the fact that the work they do is incredibly simple and the environments in which they work are fairly free of external influences that could disrupt their routines.

TIP: many industrial robots have made the jump to managing software as well as the physical construction of products, most notably in marketing automation. G2 has thousands of real user reviews dedicated to helping you find the best **marketing automation software** for your needs.   [12]

Figure 1.27: Industrial robot [12]

1.3.2. Exploration robots

These cybernetic adventures can range in complexity from simple probes to fully autonomous spacecraft. They are used to explore the farthest reaches of space and the darkest trenches of the ocean floor, boldly going where no man has gone before.

Some of the more famous examples of these, such as the Mars Rover Opportunity, are a type of robot known as 'Remotely Operated Vehicle' (or ROV) that performs some autonomous functions while having the capability of being operated by a remote operator or pilot. These robots typically come equipped with advanced observation or manipulation features. These features allow them to gather data from their assigned environments in a more focused way than their less complex aerospace cousins. [12]

Figure 1.28: Exploration robot [12]

1.3.3. Consumer robots

Consumer robots are so commonplace that many people fail to see them as robots at all! These are the tittle household helpers that unobtrusively improve the lives of countless homeowners the world over. The classic example of a consumer robot is the Roomba, an autonomous cleaning machine complete with sensors to help it navigate any space you put it in.

More recently, simplistic al has started to fuse with consumer robots and the internet of things to give us devices such as the Amazon Alexa or Google home.

These stationary robots come equipped with what's referred to as a 'conversational Al' that can read things such as tone and context to make an educated guess as to the intent of the speaker. While these devices are still far from perfect, it's

exciting to imagine the quality of life they could provide us in the next half century. [12]

Figure 1.29: Consumer robot (Roomba) [14]

1.3.4. Medical robots

Aside from production, exploration, and menial tasks, robots can also be literal lifesavers.

Medical robots can range from autonomous arms that help surgeons perform delicate operations to the emerging field of mind-controlled robotic prosthetics and exoskeletons. While we probably won't see fully autonomous surgeons for many years to come, doctor-operated robots have pushed the boundaries of medicine in terms of what can be accomplished without having to resort to risky invasive procedures. Devices such as the daVinci system have made it possible to perform dangerous with a fraction of the normal risk. [12]

Figure 1.30: Medical robot [14]

1.3.5. Aerospace robots

Aerospace robots are, in some capacity, able to fly. They differ from exploration robots in that they don't include aquatic automatons or surface rovers.

Common forms of these robots are autonomous or remote controlled drones or spacecraft that can be used for a variety of purposes such as research, military intelligence, or deep space exploration. [5]

Figure 1.31: Aerospace robots (drone) [12]

1.3.6. Aquatic robots

Aquatic robots go far beyond deep-sea exploration.

They can work with the coast guard as unmanned boats and have often been used by marine biologists and conservationists to help supplement parts of marine ecosystems that have been ravaged by climate change and industrialization efforts like off-shore drilling.

Interestingly, Aquatic robots is also one of the subfields at the forefront of bio-robotics, where the robots developed take inspiration from organic animals in the wild for how they move.

Many researchers have found great success with modeling their swimming robots off of common underwater species such as fish or eels, whose innate abilities to navigate the underwater seascape is far superior to traditional methods of locomotion for human-made sea-craft. [12]
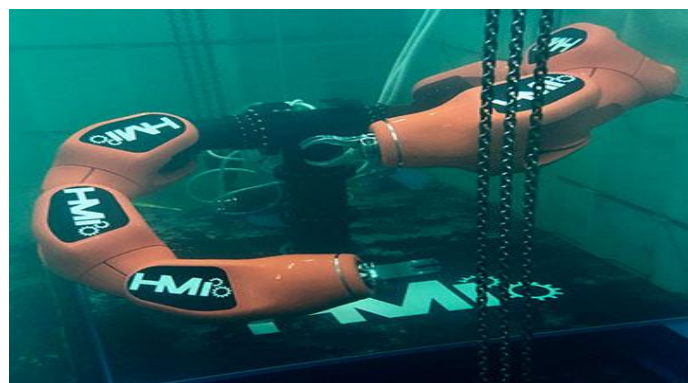
Figure 1.32: Aquatic robot [14]

1.4. Robot sensor

There are different sensors to choose from and we will identify the characteristics of few sensors, and also understand why and where they are used.

- Light sensors: A Light sensor is used to detect light and create a voltage difference. The two main light sensors generally used in robots are Photoresistor and Photovoltaic cells. Other kinds of light sensors like Phototubes, Phototransistors, CCD's etc. are rarely used.

- Sound sensor: this sensor (generally a microphone) detects sound and returns a voltage proportional to the sound level. A simple robot can be designed to navigate based on the sound it receives. Imagine a robot which turns right for one clap and turns left for two claps. Complex robots can use the same microphone for speech and voice recognition.

- Temperature sensor: What if your robot has to work in a desert and transmit ambient temperature? Simple solution is to use a temperature sensor. Tiny temperature sensor ICs provide voltage difference for a change in temperature. Few generally used temperature sensor IC's are LM34, LM35, TMP35, TMP36, and TMP37.

- Contact sensor: Contact sensors are those which require physical contact against other objects to trigger. A push button switch, limit switch or tactile bumper switch are all examples of contact sensors. These sensors are mostly used for obstacle avoidance robots.

- Other sensors for robots like: distance sensor, proximity sensor…eg. There are hundreds of sensors made today to sense virtually anything you can think of, and it is almost impossible to list all available sensors. [18]

1.5. Conclusion

In this chapter we talk about the history of robots but it is not the all history…it is just a general look for the robots, and we talk about the types of them, in the second chapter we will talk about our project and the tools that we used for it.

CHAPTER 2
TOOLS USED FOR CONTROLLING THE PRAYING ROBOT

2.1. Introduction

There is many different robots around the world with different forms, sizes, and tasks, especially in kids toys markets, there is who play music, who singing, who dancing which make us ask a question, why we should not made a robot which can learn our kids something's more important, we don't found something more important than praying; The kid will be affected by his favourite toy so why not became his toy praying and makes him praying too.

In this chapter, we will talk about tools using to realise our praying robot

2.2. Hardware parts

Praying robot is a religious type of robots…it's doing the five pray of our religion Islam in the real time of pray or by remote control any time we want to.

In our project we used many tools which are controlled by programs. Its composed by two parts:

- Hardware part: The hardware part is full of various electronics components.
- Software part

2.2.1. The Arduino board

2.2.1.1. Definition

Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex. This board contains a microcontroller which is able to be programmed to sense and control objects in the physical world. By responding to sensors and input, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays. Because of it's flexibility and low cost, Arduino has become a very popular choice for makers and makerspaces looking to create interactive hardware projects.

Arduino was introduced back in 2005 in Italy by Massimo Banzi as a way for non-engineers to have access to a low cost, simple tool for creating hardware projects. Since the board is open-source, it is released under a Creative Commons license which allows anyone to produce their own board. If you search the web, you will find there are hundreds of Arduino compatible clones and variations available but the only official boards have Arduino in it's name. [21]

2.2.1.2. Power

The Arduino can be powered via USB connexion or with an external power supply.

The power source is selected automatically.

External power can come either from an AC to DC adapter wall-wart or battery. The adapter can be connected by plugging a 2.1 mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the power connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7 V, however, the 5V pin may supply less than 5V and the board may be unstable.

If using more than 12 V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins is as follows:

- VIN: the input voltage to the Arduino board when it's using an external power source (as opposed to 5V from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V: the regulated power supply used to power the microcontroller and other components on the board. This come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- 3.3V: a 3.3 V supply generated by the on-board regulator. maximum current draw is 50mA.

- GND: ground pins. [12]

2.2.1.3. Types of the Arduino board

The list of Arduino boards includes the following such as

➢ Arduino Uno board
➢ Lilypad Arduino board
➢ Redboard Arduino board
➢ Arduino mega board
➢ Arduino Leonardo board

➢ The Arduino shields [5]

2.2.1.3.1. Advantages of Arduino board

◆ The Arduino boards are very easy to get started.

◆ The Arduino boards are used in the automatic room light control

◆ On the Arduino, both software, hardware, and IDE are open source

◆ the Arduino boards are used in the real-time application [12]

2.2.1.3.2. The Arduino Uno board

2.2.1.3.2.1. Definition

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmege328p microcontroller and developed by Arduino.cc. …the board has 14 digital I/O pins (six capable of PMW output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated development Environment), via a type B USB cable.[12]

2.2.1.3.2.2. Board Breakdown

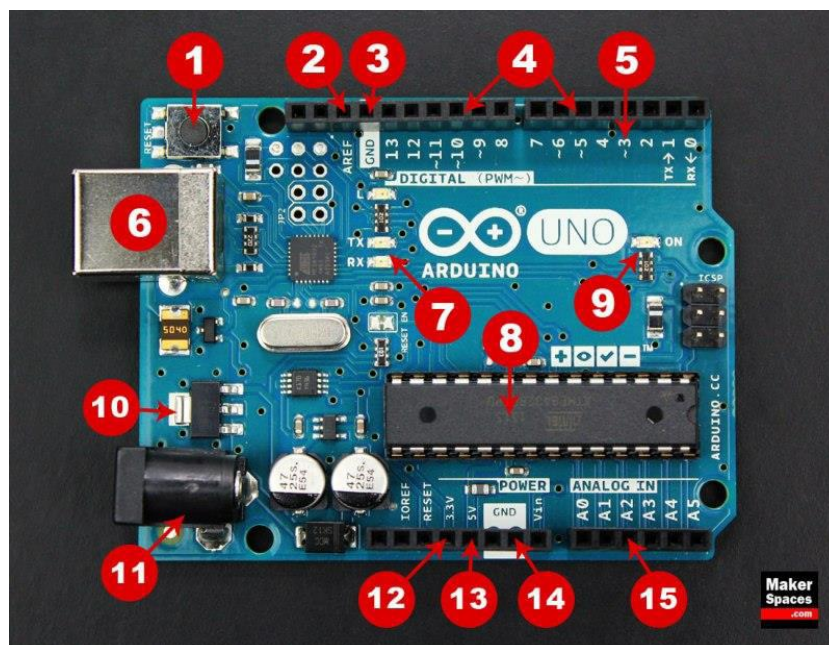Here are the components that make up an Arduino board and what each of their functions are.

Figure 2. 1: The Arduino Uno board [15]

1. Reset Button: this will restart any code that is loaded to the Arduino board.
2. AREF: stands for '' Analog Reference'' and is used to set an external reference voltage.
3. Ground pin: there are a few ground pins on the Arduino and they all work the same.
4. Digital input/output: pins 0-13 can be used for digital input or output.
5. PWM: the pins marked with the (~) symbol can simulate analog outout.
6. USB: Connection used for powering up your Arduino and uploading sketches.
7. TX/RX: transmit and receive data indication LEDS.
8. Atmega Microcontroller: this is the brains and is where the programs are stored.
9. Power LED: Indicator this LED lights up anytime the board is plugged in a power source.
10. Voltage Regulator: this controls the amount of voltage going into the Arduino board.
11. DC Power Barrel Jack: this is used for powering your Arduino with a power supply.
12. 3.3V Pin: this pin supplies 3.3 volts of power to your projects.
13. 5V Pin: this pin supplies 5 volts of power to your projects.
14. Ground Pins: there are a few ground pins on the Arduino and they all work the same.
15. Analog Pins: these pins can read the signal from an analog sensor and convert it to digital. [15]

2.2.1.3.2.3. Microcontroller

This ATMEGA-328 integrated chip consists of 28 pins. It consists of 6 analog inputs that are shown in the pin diagram. Analog inputs can be represented as PC0 to PC5. These analog input pins posses the continuous time signal which acts as an analog input for the system. Further it also consists of 12 digital inputs.

It can be represented as PD1 to PD11 which act as an digital input ports based on pulse width modulation PWM. These PWM, which transmit the signal in the form of discredited form. Both analog and digital input ports can be used for various applications for the input power supply, VCC and GND pins are used. Pins PB6 and PB7, which acts as a crystal to generate a clock signal. By using these crystal, we can

generate the clock signal and by these clock signals, we can use this clock signal for input sources.

PC6 pin are the one where it can be used for the reset option. Resetting the program can be done by using this PC6 pin.

The pin diagram of atmega-328 microcontroller can be shown below. [28]
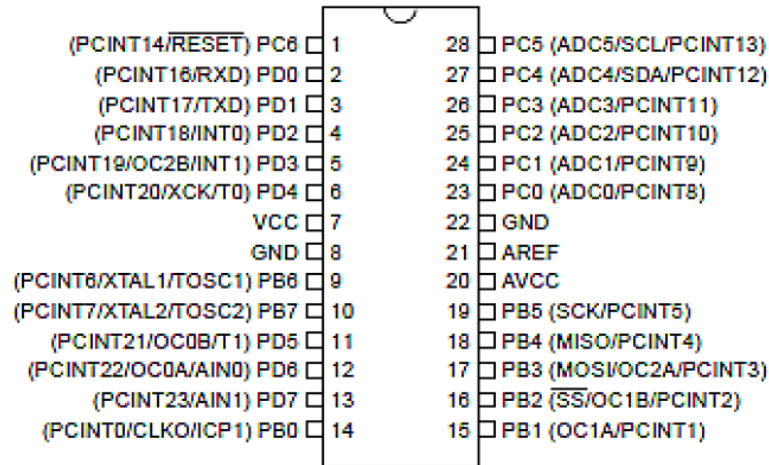
Figure 2. 2: ATMEGA-328 integrated chip [16]

2.2.1.3.2.4. Arduino Uno board specifications

Microcontroller	AT mega 328p
Operating voltage	5V
Input voltage (recommended)	7-12 V
Input voltage (limits)	6-20 V
Digital I/O pins	14 pins
Analog Input pins	06 pins
Flash memory	32 KB
Memory SRAM memory volatile	2 KB
Memory EEPROM memory non volatile	1 KB
Clock speed	15 MHz

Table 2. 1: The Arduino Uno board specifications [17]

2.2.2. Driver card PCA 9685

2.2.2.1. Definition

The PCA9685 is a 16-channel I2C-bus controlled LED controller optimized for Red/Green/Blue/Amber (RGBQ) color backlighting applications. Each LED output has individual 12-bit resolution (4096 steps) PWM controller with a fixed frequency. The controller operates at a programmable frequency from a typical 24 Hz to 1526 Hz with a duty cycle that is adjustable from 0% to 100% so the LED can be set to the same PWM frequency.

With the PCA9685 as the master chip, the 16-channel 12-bit PWM Servo Driver only needs 2 pins to control 16 servos, thus greatly reducing the occupant I/Os.

Moreover, it can be connected to 62 driver boards at most in a cascade way, which means it will be able to able to control 992 servos in total.[5]

Figure 2. 3: Driver card PCA 9685 [17]

2.2.2.2. Power pins

- **GND**- this is the power and signal ground pin, must be connected
- **VCC**-this is the logic power pin, connect this to the logic level you want to use for the PCA9685 output, should be 3-5V max, it's also used for the 10K pull-up on SCL/SDA so unless you have your own pull-up, have it match the microcontrollers logic level too
- **V+**-this is an optional pin that will supply distributed power to the servos. If you are not using for servos you can leave disconnected. It is not used at all by the chip. You can also inject power from the 2 pin terminal block at the top of the board. You should provide 5-6VDC if you are using servos. if you have to, you can go higher to 12VDC, but if you mess up and connect VCC to V+ you could damage your board [24]

2.2.2.3. Control pins

- SCL- I2C clock pin, connect to your microntrollers I2C clock line. Can use 3V or 5V logic, and has a weak pull-up to VCC

- SDA- I2C data pin , connected to your microcontrollers I2C data line. Can use 3V or 5V logic, and has a weak pull-up to VCC

- OE- Output enable. Can be used to quickly disable all outputs. When this pin is low all pins are enabled. Pull-up low by default so it's an optional pin [24]

2.2.2.4. Output Ports

There are 16 output ports. Each port has 3 pins: V+, GND and the PWM output. Each PWM runs completely independently but they must all have the same PWM frequency. That is, for LEDs you probably want1.0 KHz but servos need 60 Hz – so you cannot use half for LEDs 1.0 KHz and half 60 Hz [24]

2.2.2.5. Connecting a servo

Most servos come with a standard 3-pin female connector that will plug directly into the ground wire (usually black or brown) with the bottom row and the signal wire (usually yellow or white) on the top.[24]

2.2.2.6. Adding more servos

Up to 16 servos can be attached to one board. If you need to control more than 16 servos, additional boards can be chained as described on the next page.[24]

2.2.2.7. Chaining Drivers

Multiple Drivers up to 62 can be chained to control still more servos. With headers at both ends of the board, the wiring is as simple as connecting a 6-pin parallel cable from one board to the next. [24]

2.2.3. Servo motor

2.2.3.1. Definition

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate an object at some specific angles or distance, then you use servo motor. …if motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor.

2.2.3.2. Servo motor MG90

2.2.3.2.1. Definition

MG90S is a micro servo motor with metal gear. This small and lightweight servo comes with high output power, thus ideal for RC Airplane, quadcopter or robotic arms.[26]

Position ''0'' (1.5 ms pulse is middle), ''90'' (~2 ms pulse) is all way to the right, ''-90'' (~1 ms pulse) is all the way to the left. [25]

Figure 2. 4: Servo motor MG90 [17]

2.2.3.3. Servo motor MG90S specifications

Dimension	22.8* 12.2*28.5 mm
Weight	13.4 g
Operating voltage	4.8 to 6V
Operating speed	0.12s/60° sous 4.8 V
Stall torque	1.8 Kg/cm sous 4.8 V
Rotation angle	0° to 180°
Temperature range	0°c – 55°c

Table 2. 2: The MG90 specifications [17]

2.2.4. RTC (Real Time clock)

2.2.4.1. Definition

RTC means Real Time Clock. RTC modules are simply TIME and DATE remembering systems which have battery setup which in the absence of external power keeps the module running. This keeps the TIME and the DATE up to date. So we can have accurate TIME and DATE from RTC module whenever we want.[27]

2.2.4.2. RTC DS 3232

2.2.4.2.1. Definition

The DS3231 is a low-cost, extremely accurate I2C real-time clock RTC with an integrated temperature- compensated crystal oscillator TCXO and crystal. [27]
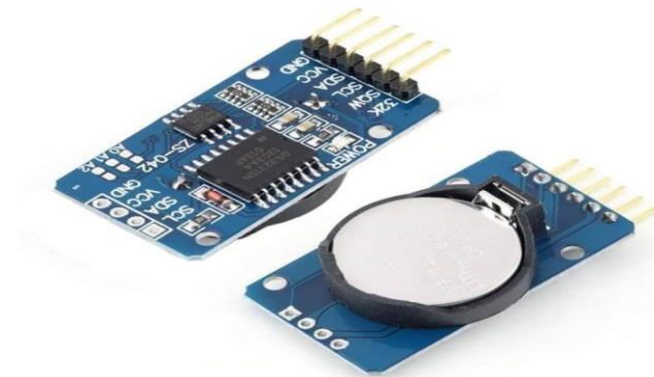
Figure 2. 5: The DS3231 RTC module [17]

2.2.4.3. DS 3231 RTC pin configuration

Pin name	Description
VCC	Connected to positive of power source
GND	Connected to ground
SDA	Serial Data pin I2C interface
SCL	Serial Clock pin I2C interface
SQW	Square Wave output pin
32K	32K oscillator output

Table 2. 3: The DS3231 RTC pin configuration [27]

2.2.4.4. DC3231 RTC module Applications

➔ Robotics

➔ Gaming

➔ Servers

➔ Computer peripherals

➔ GPS

➔ Utility power meters [27]

2.2.4.5. DS3231 RTC module specifications

Operating voltage	2.3 V- 5.5 V
Maximum voltage at SDA, SCL	VCC + 0.3 V
Operating temperature	-45°c to +80°c
Can operate on LOW voltages	
Consumes 500nA on battery backup	

Table 2. 4: DS3231 RTC module specifications [27]

2.2.5. TM1637 Based 4 Bits Red Digital Tube LED Display Module

2.2.5.1. Definition

TM1637 is a chip for driving 7-segment display. There are several modules using this chip to form a 4 digit numerical display module. TM1637 has also keyboard input capability, but is not used in this module.[22]
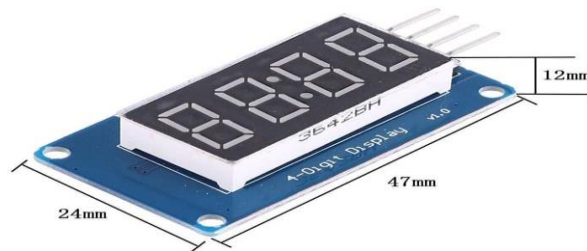
Figure 2. 6: The TM1637 module [17]

2.2.5.2. Applications of TM1637 based 4Bits Red Digital Tube LED Display Module

- ➔ DIY clock project.
- ➔ 4 digit display unit.
- ➔ Electrical project using 7 segment display.[22]

2.2.5.3. Hardware Connection

The display module has 4 pins

- ◆ VCC GND – Self explanatory
- ◆ CLK – Clock ;connect to any digital pin on the Arduino
- ◆ DIO – Data I/O; connect to any digital pin on the Arduino

When a TM1637 Display object is created, the clock and data pin numbers are passed as arguments and get bound. [22]

2.2.5.4. The TM1637 chip

2.2.5.4.1. Definition

TM1637 is a kind of LED (light-emitting diode display) drive control special circuit with keyboard scan interface and it's internally integrated with MCU digital interface, data latch, LED high pressure drive and keyboard scan. This product is in DIP20/SOP20 package type with excellent performance and high quality, which is mainly applicable to the display drive of induction cooker, micro-wave oven and small household electrical appliance.[22]

2.2.5.4.2. Function features

- Applied power CMOS technique
- The display mode (8segments*6 bit) supports output by common anode LED
- Keyboard scan (8*2 bit), with enhanced identification circuit with anti-interference keys
- Luminance adjustment circuit (adjustable 8 duty ration)
- Two-wire serial interface (CLK,DIO)
- Oscillating type: Built- in RC oscillator
- Built-in power-on reset circuit
- Built-in automatic blanking circuit
- Package type: DIP20/SOP20 [22]

2.2.6. SD Reader AMS1117 module

2.2.6.1. Definition

It is easily interfaced as a peripheral to your Arduino, and other embedded boards and microcontrollers. Through programming, you can read and write to the micro SD card over SPI interface, this SD Reader based on the AMS1117 chip. [5]
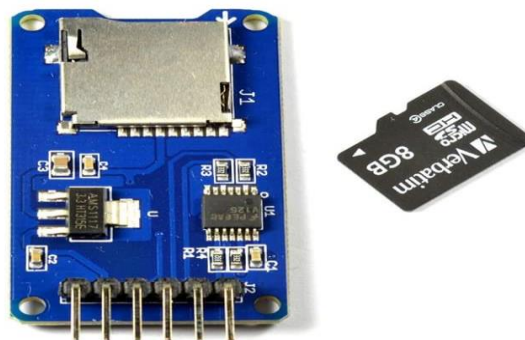

Figure 2. 7: SD Reader AMS1117 module [17]

2.2.6.2. Hardware Connection

The display module has 6 pins

◆ MISO - connect to pin12 on the Arduino

◆ MOSI - connect to pin11 on the Arduino

◆ SCK- connect to pin13 on the Arduino

◆ CS- connect to any digital pin on the Arduino

◆ VCC - connect to the 5V on the Arduino

◆ GND- connect to the GND on the Arduino

2.2.6.3. The AMS1117 chip

The AMS1117 series of adjustable and fixed voltage regulators are designed to provide up to 1A output current and to operate down to 1V input-to-output differential. The dropout voltage of the device is guaranteed maximum 1.3V, decreasing at lower load currents. [23]

2.2.6.4. Features

- Three terminal adjustable or fixed voltages (1.5V, 1.8V, 2.5V, 2.85V, 3.3V and 5V)

- Output current of 1A

- Operates down to 1V dropout

- Line regulation: 0.2 % Max

- Load regulation: 0.4 % Max [23]

2.2.6.5. Applications

- High efficiency linear regulators

- Post regulators for switching supplies

- 5V to 3.3V linear regulator

- Battery chargers

- Active SCSI terminators

- Power management for Notebook

- Battery powered instrumentation [23]

2.2.7. Memory card

A memory card or memory cartridge is an electronic data storage device used for storing digital information, typically using flash memory. These are commonly used in portable electronic devices, such as digital cameras, mobile phones,

computers, tablets, PDAs, portable media players, video game consoles, synthesizers, electronic keyboards and digital pianos. [5]

We used a memory card of 2GB.

2.2.8. 9Volt battery

2.2.8.1. Definition

The 9 volt battery is a common size of battery that was introduced for the early transistor radios. It has a rectangular prism shape with rounded edges and a polarized snap connector at the top. This type is commonly used in walkie-talkies, clocks and smoke detectors. [5]

2.2.8.2. Connectors

The battery has both terminals in a snap connector on one end. The smaller circular male terminal is positive, and the larger hexagonal or octagonal female terminal is the negative contact. The connectors on the battery are the same as on the load device; the smaller one connects to the larger one and vice versa.[5]

Figure 2. 8: 9 volt battery with connector [17]

2.2.9. Breadboard

Breadboard is a thin plastic boards used to hold electronic components (transistors, resistors, chips, etc…) that are wired together. Used to develop prototypes of electronic circuits, breadboards can be reused for future jobs.

They can be used to create one-of-a–kind systems but rarely become commercial products. See printed circuit board.

There are two major types of breadboards; these are solder and solderless boards. Solder boards are boards you have to solder components onto per the name. these are most of your standard circuit boards, and if you flip one over you'll notice that all of the connection. [19]
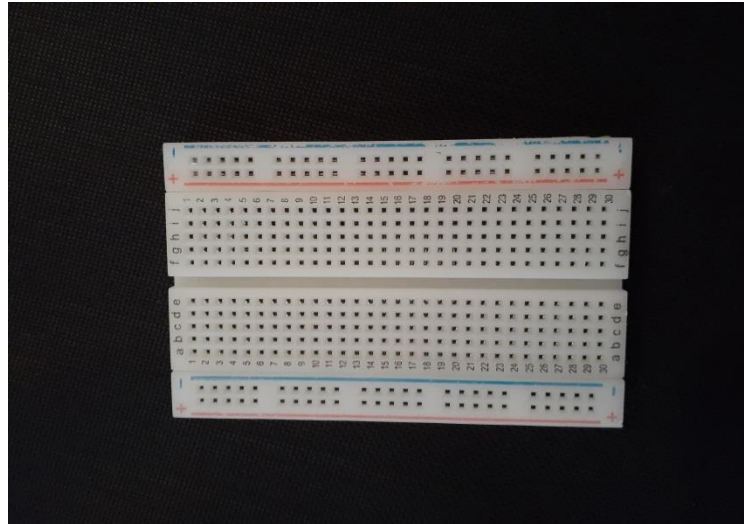
Figure 2. 9: Breadboard

2.2.10. Connection Wires

2.2.10.1. Definition

Jumper wires are simply wires that have connector pins at each end, aloowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. Fairly simple. In fact, it doesn't get much more basic than jumper wires. [20]

2.2.10.2. What do the colors mean?

Though jumper wires come in a variety of colors, the colors doesn't actually means anything. This means that a red jumper wire is technically the same as a black one. But the colors can be used to your advantage in order to differentiate between types of connections, such as ground or power. [20]

2.2.10.3. Types of jumper wires

Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. [20]
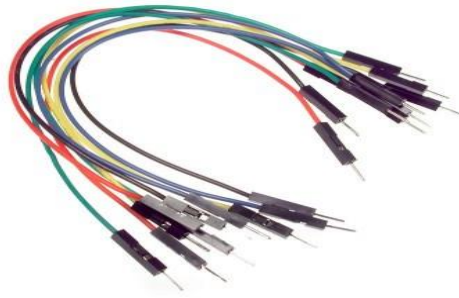
Figure 2. 10: Jumper wires [5]

2.3. Software

We used ARDUINO IDE to control the tools with a long program

2.3.1. Introduction to Arduino IDE

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.

The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the giver Arduino Module.

This environment supports both C and C++ languages.[9]

2.3.2. How to Download Arduino IDE

You can download the software from Arduino main website. The software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

The IDE environment is mainly distributed into three sections:

1. Menu Bar
2. Text Editor
3. Output Pane

As you download and open the IDE software, it will appear like an image
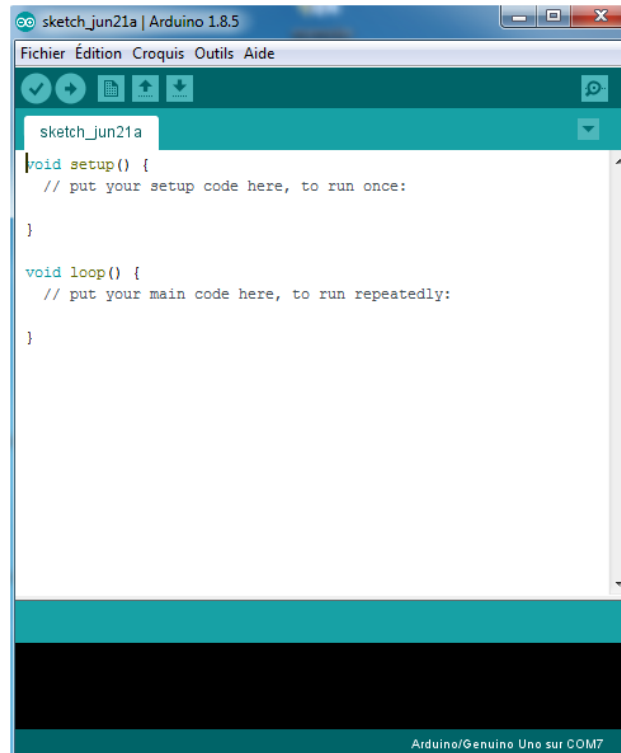
below. [9]

Figure 2. 11: The window after opening the IDE software

The bar appearing on the top is called Menu Bar that comes with five different options as follow. [9]

File- You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into. [9]

New	**This is used to open new text editor window to write your code**
Open	Used for opening the existing written code
Open Recent	The option reserved for opening recently closed program
Sketchbook	It stores the list of codes you have written for your project
Examples	Deferent examples already stored in the IDE software
Close	Used for closing the main screen window of recent tab. If two tabs are open, it will ask you again as you aim to close the second tab
Save	It is used for saving the recent program

Save as	It will allow you to save the recent program in your desired folder
Page setup	Page setup is used for modifying the page with portrait and landscape options. Some default page options are already given from which you can select the page you intend to work on
Print	It used for printing purpose and will send the command to the printer
Preferences	It is page with number of preferences you aim to setup for your text editor page
Quit	It will quit the whole software all at once

Table 2. 5: The number of further subdivisions the file option is categorized into. [9]

As you go to the preference section and check the compilation section, the Output pane will show the code compilation as you click the upload button. [9]
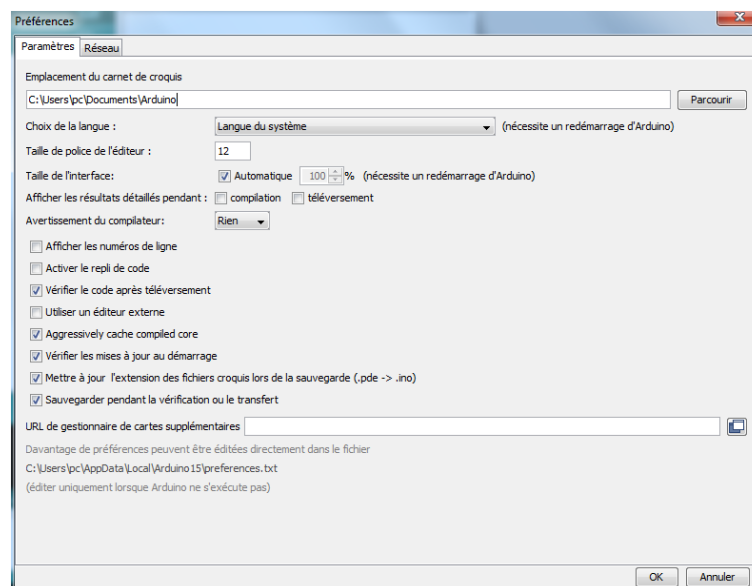

Figure 2. 12: The code compilation

And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve. [9]

Edit- Used for copying and pasting the code with further modification for font.

Sketch- For compiling and programming.

Tools- Mainly used for testing projects. The programmer section in this panel is used for burning a boot loader to the new microcontroller.

Help- In case you are feeling sceptical about software, complete help is available from getting started to troubleshooting.

The Six Buttons appearing under the Menu tab are connected with the running program as follow.

Figure 2. 13: The Six Buttons appearing under the Menu

2.3.3. Libraries

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to include Library.

As you click the Include Library and Add the respective library it will on the top of the sketch with a **#include** sign. Suppose, I include the SD library, it will appear on the text editor as   **#include< SD.h>**

Most of the library are preinstalled and come with the Arduino software. However, you can also download them from the external sources. [9]
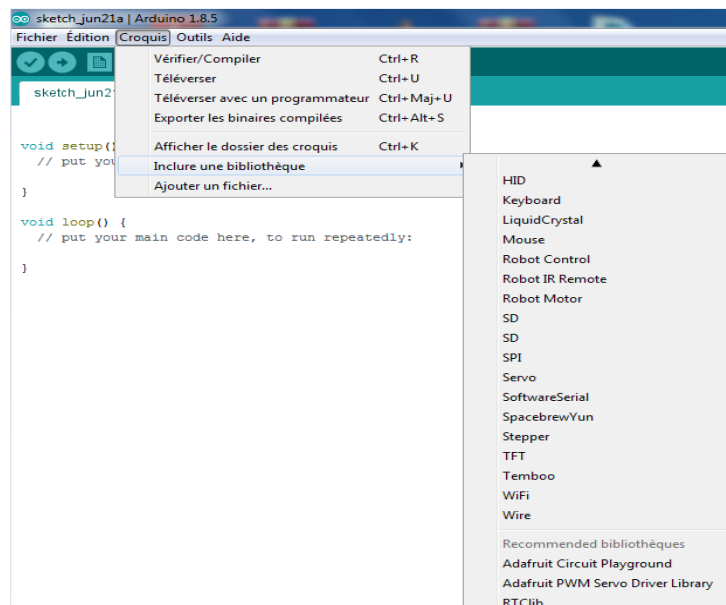

Figure 2. 14: How to add library

2.3.4. Making Pins Input or Output

The digitalRead and digitalwrite commands are used for addressing and making the Arduino pins as an input and output respectively.

These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalwrite starting with small ''d'' and write with capital ''W''. Writing it down with Digitalwrite or digitalwrite won't be calling or addressing any function. [9]

2.3.5. How to select the Board

In order to upload the sketch, you need to select the relevant board you are using and the ports for that operating system. As you click the tools on the Menu, it will open like the figure below. [9]
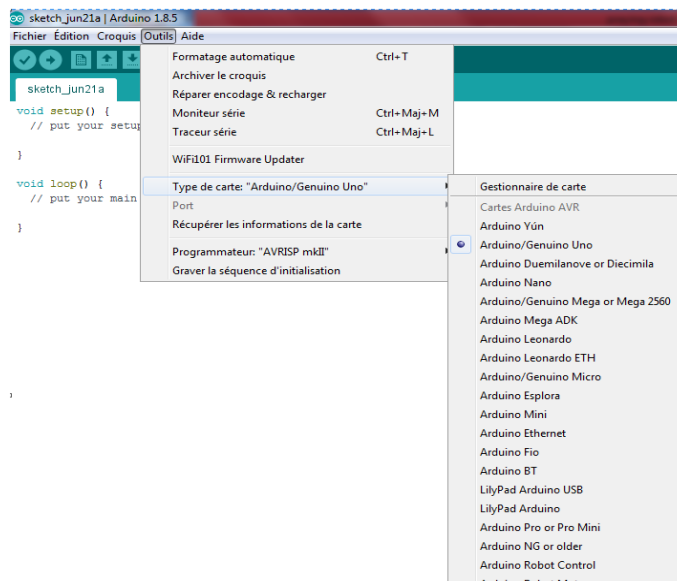

Figure 2. 15: How to select the Board

- Just go to the '' Board'' section and select the board you aim to work on. Similarly, COM1, COM2, COM3, COM4, COM5, COM7 or higher are reserved for the serial and USB board. You can look for the USB serial device in the ports section of the windows Device Manager.

- After correct selection of both Board and Serial port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.

- The sketch is written in the text editor and is then saved with the file extension. ino.

It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.

- Once you upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

- The amazing thing about this software is that no prior arrangement or bulk of mess is required to install this software, you will be writing your first program within 2 minutes after the installation of the IDE environment. [9]

2.3.6. Bootloader

As you go to the Tools section, you will find a bootloader at the end. It is very helpful to burn the code directly into the controller, setting you free from buying the external burner to burn the required code.

When you buy the new Arduino Module, the bootloader is already installed inside the controller. However, if you intend to buy a controller and put in the Arduino Module, you need to burn the bootloader again inside the controller by going to the Tools section and selecting the burn bootloader. [9]

2.4. Conclusion

The prying robot it is a good idea for the kids to makes them learn how to pray because the kid follows his favorite toy, so it is so good to be his favorite one teach him something important like praying.

The tools cited in this chapter are used to achieve our praying robot and in the next one we will talk about the programming and the simulating of each tool.

CHAPTER 3

CONTROL AND

SIMULATING OF

ROBOT

3.1. Introduction

The praying robot consists on several parts, each one need to be programming alone.

In this chapter, we program each part of our praying robot and we make sure that these parts operate together.

3.2. Connecting and circuit of connection

3.2.1. DS3231 RTC module

The first thing which is programming was the DS 3231 RTC module to make sure that we work with the real correct time.

3.2.1.1. Connection

It's easy to connect the DS3231 with the Arduino board

- ➢ **SCL** with **A5** of Arduino
- ➢ **SDA** with **A4** of Arduino
- ➢ **VCC** with **5V** of Arduino
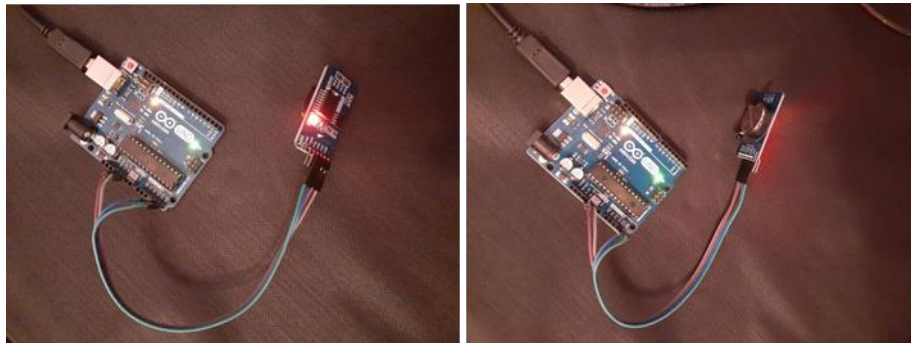- ➢ **GND** with **GND** of Arduino

Figure 3. 1: The DS3231 connected with the Arduino board

And to make sure that we made the real program there is just one way:

- • We must connect the RTC DS3231 module with the TM1637 module and watch the time on it, in the same time it's a testing for the TM1637 module.

3.2.1.2. Program of the first time of DS3231 module

The first program used to adjust the time is employing once. Then, the time will be correct for a long time because the DS3132 module has a battery inside to make sure that the module will be ever connecting with power.

The following script is used to progam the DS3132 module.

#include <Wire.h>

#include "RTClib.h"

RTC_DS3231 rtc;

```
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};
void setup () {
Serial.begin(9600);
 delay(3000); // wait for console opening
 if (! rtc.begin()) {
   Serial.println("Couldn't find RTC");
   while (1);
  }
if (rtc.lostPower()) {
   Serial.println("RTC lost power, lets set the time!");
   // following line sets the RTC to the date & time this sketch was compiled
   rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
   // This line sets the RTC with an explicit date & time, for example to set
   // January 21, 2014 at 3am you would call:
   rtc.adjust(DateTime(2020, 6, 5, 9, 55, 0));
  }
void loop () {
   DateTime now = rtc.now();
    Serial.print(now.year(), DEC);
   Serial.print('/');
   Serial.print(now.month(), DEC);
   Serial.print('/');
   Serial.print(now.day(), DEC);
   Serial.print(" (");
   Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
   Serial.print(") ");
   Serial.print(now.hour(), DEC);
   Serial.print(':');
   Serial.print(now.minute(), DEC);
   Serial.print(':');
   Serial.print(now.second(), DEC);
   Serial.println();
   Serial.print(" since midnight 1/1/1970 = ");
```

```
    Serial.print(now.unixtime());
    Serial.print("s = ");
    Serial.print(now.unixtime() / 86400L);
    Serial.println("d"); //calculate a date which is 7 days and 30 seconds into the
futureDateTime future (now + TimeSpan(7,12,30,6));
 Serial.print(" now + 7d + 30s: ");
    Serial.print(future.year(), DEC);
    Serial.print('/');
    Serial.print(future.month(), DEC);
    Serial.print('/');
    Serial.print(future.day(), DEC);
    Serial.print(' ');
    Serial.print(future.hour(), DEC);
    Serial.print(':');
    Serial.print(future.minute(), DEC);
    Serial.print(':');
    Serial.print(future.second(), DEC);
    Serial.println();
    Serial.println();
    delay(3000);
}
```

3.2.2. The TM1637 module

3.2.2.1. Connection

The TM1637 is easy to connected with the Arduino board too

> **CLK** with **PIN2** of Arduino

> **DIO** with **PIN3** of Arduino

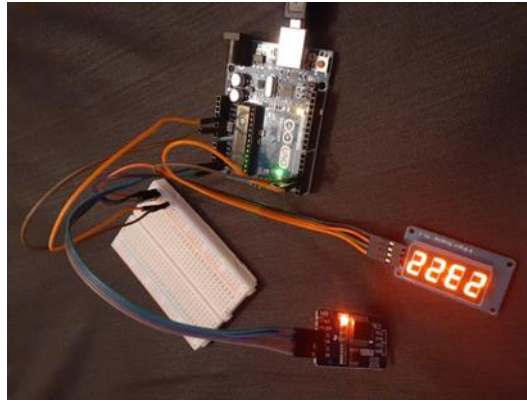> **VCC** with **5V** of Arduino

> **GND** with **GND** of Arduino

Figure 3. 2: The DS3231 and the TM1637 connected with the Arduino board

3.2.3. PCA 9685 module

To connect the **PCA 9685** module with the Arduino board we must connected with external power too, it is better to use the 9V battery.

3.2.3.1. Connection

- ➢ **SCL** with **A5** of Arduino
- ➢ **SDA** with **A4** of Arduino
- ➢ **VCC** with **5V** of Arduino
- ➢ **GND** with **GND** of Arduino
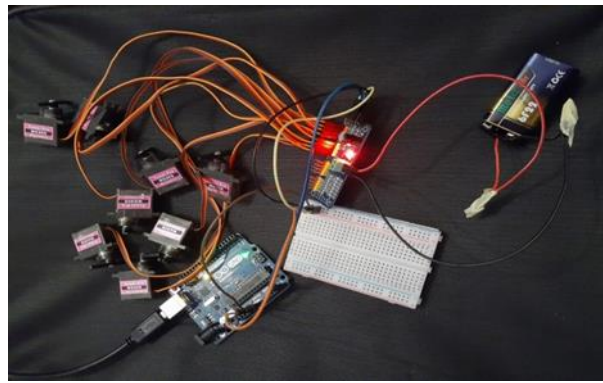- ➢ **V+** and **GND** with the **V+** and the **V-** of external power

Figure 3. 3: The PCA 9685 module connected with the Arduino board and 8 servo motors

3.2.3.2. Program of the PCA 9685

Firstly, We used a program to make sure that the module is ready to use.

the script employed is:

#include <Wire.h>

#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

```
#define SERVOMIN  150
#define SERVOMAX  600

// our servo # counter
uint8_t servonum = 0;

void setup() {
  Serial.begin(9600);
  Serial.println("8 channel Servo test!");

  pwm.begin();
  pwm.setPWMFreq(60);
  delay(10);
}
void setServoPulse(uint8_t n, double pulse) {
  double pulselength;

  pulselength = 1000000;   // 1,000,000 us per second
  pulselength /= 60;   // 60 Hz
  Serial.print(pulselength); Serial.println(" us per period");
  pulselength /= 4096;  // 12 bits of resolution
  Serial.print(pulselength); Serial.println(" us per bit");
  pulse *= 1000000;  // convert to us
  pulse /= pulselength;
  Serial.println(pulse);
  pwm.setPWM(n, 0, pulse);
}

void loop() {

  Serial.println(servonum);
  for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX; pulselen++)
{
```

```
    pwm.setPWM(servonum, 0, pulselen);

  }


  delay(500);
  for (uint16_t pulselen = SERVOMAX; pulselen > SERVOMIN; pulselen--) {
    pwm.setPWM(servonum, 0, pulselen);

  }
delay(500);


  servonum ++;
  if (servonum > 7) servonum = 0;

  }
```

This script is the same used for testing servomotors.

3.2.4. The AMS1117 module and the speaker

We used the speaker to play many wav files, and the best way for that it is using the AMS1117 SD Reader because it's so easy to connected it with the Arduino board and so easy to program it too.

3.2.4.1. Connection

- ➢ **MISO** with **PIN12** of Arduino
- ➢ **MOSI** with **PIN11** of Arduino
- ➢ **SCK**   with **PIN13** of Arduino
- ➢ **CS**   with **PIN10** of Arduino(the only pin that you can change it)
- ➢ **VCC**   with **5V** of Arduino
- ➢ **GND**   with **GND** of Arduino
- ➢ There are 2 wires in the speaker, one for the **GND** and one for any PIN you want, we used the **PIN9.**
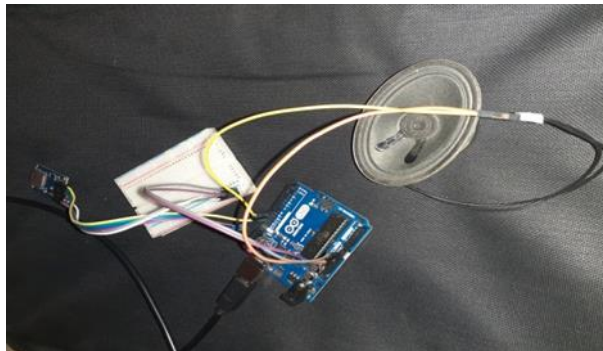

Figure 3. 4: The speaker and the SD reader connected with

It is so easy to play a wave file from SD Reader using the speaker.

```
#include <TMRpcm.h>
#include <SD.h>
#define SD_ChipSelectPin 10
TMRpcm tmrpcm;
void setup() {
  Serial.begin(9600);
  tmrpcm.speakerPin = 9;
  Serial.println("speakerPin");
  if (!SD.begin(SD_ChipSelectPin)) {
    Serial.println("initialized");
    return;
  }
 Serial.println("setVolume");
 // 0 to 7. Set volume level
 tmrpcm.setVolume (6);
 Serial.println("Play");
 tmrpcm.play("1.wav");
 Serial.println("Don");
 }
 void loop() {
  // put your main code here, to run repeatedly:
 }
```

3.3. Results and discussions

It is not easy to connect a lot of servo motors together to help a robot to move example praying, that is why we liked to work with the Arduino board to make it a little bit easy, and made a robot with a simple tools.

The prying robot has a lot of parts that can moves together or each one of them alone.
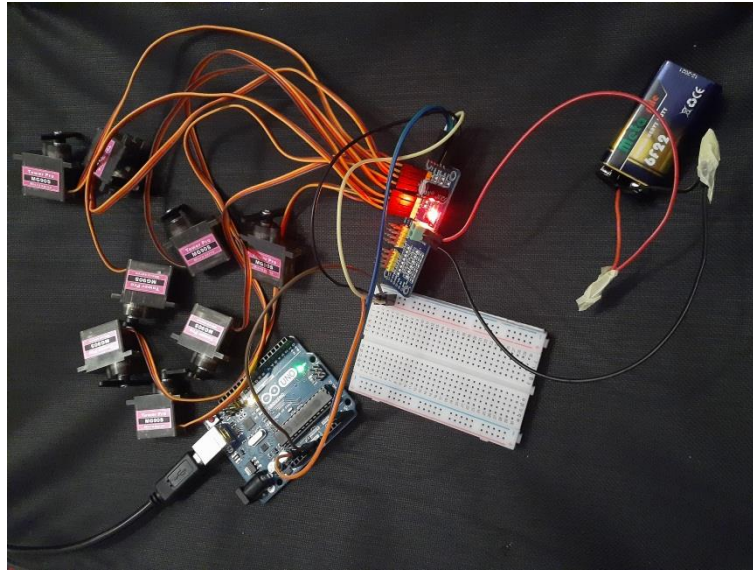
Figure 3. 5: The Arduino board with servo motors

- The arm

We used three servo motors in every single arm, one to move up and down, and the two others servo motors to move inside.
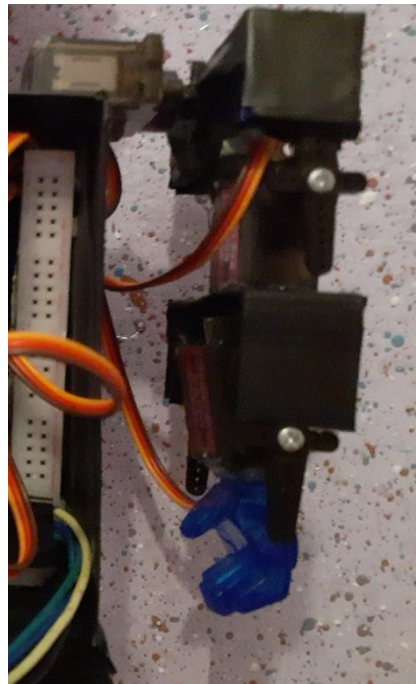

Figure 3. 6: The Arm of praying robot

We used two servo motors in every single leg to move inside and to make the body moves too.

Figure 3. 7: The legs of praying robot

-   The head

We just used one servo motor to turn the head right and left in the end of praying.
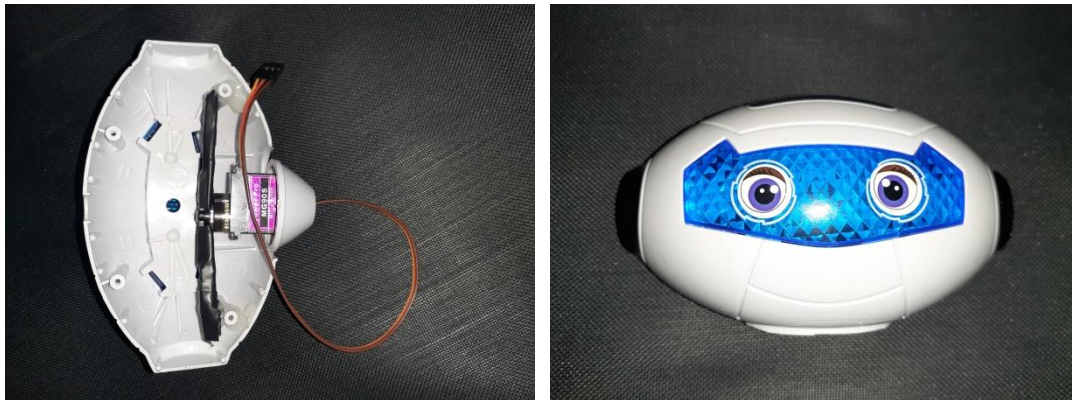

Figure 3. 8: The head of praying robot

-   The body

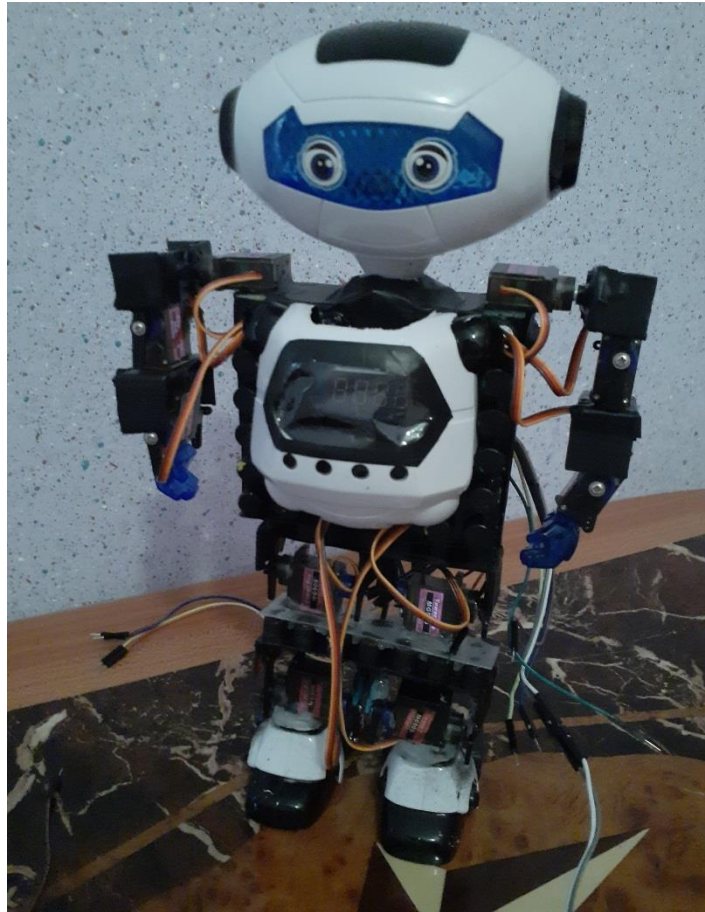We used a piece of plastic to put the other tools together: DS3231, TM1637, and PCA9685.
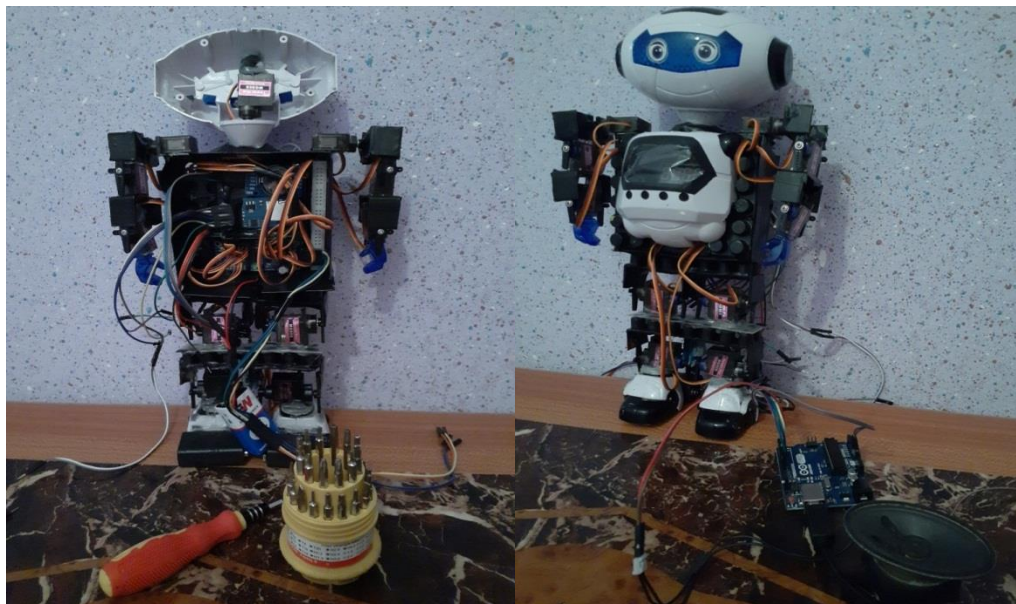
Figure 3. 9: The praying robot

Figure 3. 10: The praying robot

3.4. Conclusion

This chapter contains all scripts used to test our praying robot, this latter acts in regular movement to achieve a prayer.

Therefore, the DS3231 module was the first thing programmed to make sure that our robot prays five times in real time at the day.

Secondly, we program servomotors using the driver card PCA9685.

GENERAL CONCLUSION

General conclusion

The aim of our memoire is to realize a new, special and perfect project to get the graduation sheepskin.

Praying robot was a brand new robot we have to stand up and we make it... thanks **ALLAH**.

This memory contained three parts; in the first one we present the history of robotics and robots.

The second one, all the tools used for realizing the praying robot are mentioned. The last part present results of our robot, it contains scripts and figures of the praying robot.

It was so hard to made this robot and the big problem that we face it was the corona virus, but we did it and this is the Important thing.

Our outlook in the future is developing this robot such us:

- introducing sensors
- improve its appearance

And try to realize another robots using the small electronic tools, because the praying robot give us the inspiration to work more and more.

References

References

1. A_Brief_History_of_Industrial_Robotics_in_the_20th (pdf)
2. Roboticsprojectppt(pdf)
3. History of Robotics (pdf)
4. IFR International Federation of Robotics (pdf)
5. http://en.m.wikipedia.org
6. http://www-seeker-com
7. http://www.designboom.com
8. http://www.techradar.com
9. http://www.theengineeringprojects.com
10. IntroRob (pdf)
11. http://mymodernmet.com
12. Google research
13. NUMERIQUES (pdf)
14. https://robots.ieee.org
15. Arduino-For-Beginners (pdf)
16. ArduinoCottenceau2016 (pdf)
17. Facebook (Powertech)
18. robotics technology (word)
19. http://blog.digilentinc.com
20. http://blog.sparkfuneducation.com
21. Arduino-For-Beginners (pdf)
22. Tm1637-based-4-bit…-display-module.(pdf)
23. Ds1117.(pdf)
24. 16-channel-pwm-servo-driver(pdf)
25. sg90_datasheet (pdf)
26. http://www.components.com
27. http://datasheets.maximintegrated.com
28. IJIREEICE 6 (pdf)

Abstract

The idea to realize a robot can pray like a human is so different, this work is about controlling and realizing a prayer robot.

It's all about two parts: programming and realization. The programming part is based on the Arduino UNO board, and the realization is all about servomotors.

Keywords:

Robots, praying, Arduino, servomotors

Résumé

Ce mémoire s'inscrit dans le cadre de la modélisation et la réalisation d'un robot prieur. Cette réalisation était un défi qu'il fallait rapporter, il se porte sur deux partie : programmation et réalisation. La programmation est basée sur l'utilisation de la carte Arduino UNO. Su pondant, la réalisation est fondé sur les servomoteurs.

Mots clé :

Prier, robot, Arduino UNO, Servomoteurs

الملخص

الهدف من هذا العمل هو انجاز مشروع الكتروني مختلف عن باقي المشاريع, الروبوت المصلي هو مشروع تخرجنا لنهاية مسيرتنا الدراسية, ينقسم هذا العمل لقسمين:

البرمجة و التركيب. البرمجة اساسها لوحة الأردوينو و التركيب اساسه المحركات الذكية.

الكلمات المفتاحية:

روبوت، صلاة، أردوينو، محرك ذكي