Mohamed Khider University of Biskra

Faculty of Sciences and Technology

Department of Electrical engineering

MASTER'S THESIS

Electrical Engineering

Telecommunications

Networks and Telecommunications

Submitted and Defended

by:

Yahiaoui Chaima

On: September 23, 2020

Web Based Controlled Robot With Live Video Streaming

Jury:

Mr.	Benakcha abdelhamid	**Pr**	University of Biskra	**President**
Mr.	Boukerdin salah eddin	**MAA**	University of Biskra	**Examiner**
Mr.	Bekhouche khaled	**MCA**	University of Biskra	**Supervisor**

Academic Year: 2019 –2020

الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي و البحث العلمي

Ministry Of Higher Education and Scientific Research

Mohamed Khider Biskra University

Faculty of Sciences and Technology

Electrical Engineering Department

Field: Telecommunication

Option: Networks and Telecommunication

A Dissertation for the Fulfillment of the Requirement of a

Master's Degree

Web Based Controlled Robot With Live Video Streaming

Presented by:

Yahiaoui Chaima

Favorable opinion of the supervisor:

Bekhouche khaled

Favorable opinion of the Jury President:

Benakcha abdelhamid

Stamp and signature

.

الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي و البحث العلمي

Ministry Of Higher Education and Scientific Research

Mohamed Khider Biskra University

Faculty of Sciences and Technology

Electrical Engineering Department

Field: Telecommunication

Option: Networks and Telecommunication

A Dissertation for the Fulfillment of the Requirement of a

Theme:

Web Based Controlled Robot With Live Video Streaming

Proposed and directed by: Bekhouche khaled

ABSTRACT

Robotics are omnipresent in our daily life, especially when they comes to performing dangerous tasks, which can be automated, or require great precision. In this project we build a mobile robot with four wheels that we control it through a web page from anywhere in the world, this page contains a live video it allows us to see where is the robot and direct it through this live video, as well a four keys that control the direction of the robot : right, left, forward and backward.

الملخص

الروبوتات منتشرة في كل مكان في حياتنا اليومية ، خاصة عندما يتعلق الأمر بأداء المهام الخطرة ، والتي يمكن أن تكون آلية أو تتطلب دقة كبيرة .في هذا المشروع ، نقوم ببناء روبوت متحرك بأربع عجلات نتحكم فيه من خلال صفحة ويب من أي مكان تحتوي هذه الصفحة على فيديو مباشر يسمح لنا برؤية مكان الروبوت وتوجيهه عبر هذا الفيديو المباشر ، بالإضافة إلى أربعة مفاتيح تتحكم في اتجاه الروبوت :اليمين واليسار والأمام والخلف.

Keywords: IoT, Mobile Robot, Raspberry pi, Control, Live Video, Web Page

Dedication

I would like dedicated this thesis to:

My dear parents, for everything you have given provided me with

support, encouragement and prayer for completing my work

My brother and my sisters

My supervisor Khaled Bekhouche

And to everyone who helped and encouraged me

Acknowledgments

First, I thank almighty Allah, who gave us the strength, the will and above all the courage to accomplish this thesis.

I would like to thank my supervisor Khaled Bakhouche for his guidance, support and encouragement.also he was a source of confidence and positive energy for me, throughout of preparation this thesis.

I am thankful to the jury members: Mr. BENAKCHA abdelhamid and Mr. BOUKERDINE salah eddin for reviewing and evaluating my work.

Finally, I thank all my family for everything ; their support and their prayers for reach my goal.

Contents

Chapter I: Generalities on the Internet of Things

Chapter II: Mobile Robot

Chapter III: The Raspberry Pi

Chapter IV: Programming Languages

Chapter V: Building A Robot And Control It

List of Figures

List of Tables

List of Abbreviations:

IoT. Internet of Things.

ITU-GSI . International Telecommunication Union-Global Standards Initiative.

LAN . Local Area Network.

RFID . Radio-frequency identification.

NFC . Near Field Communication.

WAN . wide area network.

HTTP. HyperText Transfer Protocol.

LoRaWan . Long Range Wide Area Network.

WPAN . Wireless Personal Area Network.

MQTT. Message Queue Telemetry Transport.

CoAP . Constrained Application Protocol.

UDP . User Datagram Protocol.

AMQP . Advanced Message Queuing Protocol.

M2M . Machine-to-Machine Communication Protocol.

XMPP . Extensible Messaging and Presence Protocol.

AI . artificial intelligence.

RAM . Random Access Memory.

CPU. central processing unit

GPU . graphics processing unit

GPIO . The general purpose input and output.

HTML . Hyper Text Markup Language.

CSS . Cascading Style Sheets.

PHP . Hypertext Pre-processor.

VNC . Virtual Network Computing.

VPN . Virtual Private Network.

General Introduction

General introduction

During the last few years, Internet of Things (IoT) devices have started becoming a more and more important component in our daily lives. these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

In this context, robotics is also omnipresent in our daily lives, especially when it comes to performing dangerous tasks, which can be automated, or require great precision.

In these autonomous or semi-autonomous robotic systems the sensors occupy a predominant place, this is what allows the systems to understand their environment, the analysis is made possible thanks to the microprocessors which will process and react according to the different signals reported to it using various communication protocols. But we always asked ourselves how these robots were made and how we control it?.

In our project, we will try to explain and simplify all the ideas related to mobile robots and how to control it.

In the first chapter, we will begin with the definition of the internet of things and get to know their architecture. also we will mention the components of IoT, its protocols, in the end, knowing some application and The Challenges Faced by Internet of Things. Second chapter, we will try to provide a simple explanation about mobile robots, their classifications and types , also we will give the definition of embedded systems and your relationship with robotics. The third chapter, we will talk about the history of Raspberry Pi, current models of Raspberry Pi and we explain the difference between them, also we will list components of Raspberry pi 3 and it different operating systems. The fourth chapter, contains the programming languages that I will use in my project (HTML, CSS, PHP and JavaScript). In the last chapter, we will learn how to use and program our Raspberry Pi in a robotic environment. we will see a simple synoptic diagram For clarify our work, this diagram contains two important parts: the first one is hardware:we will present the components of robot and how to make a robot , and the second is the software: preparation of the raspberry pi and create a control webpage.

Chapter I

Generalities on the Internet of Things

I.1 Introduction:

The IoT is considered as a pillar of future Internet and expected to enable intelligent operations and advanced communications of devices, smart objects, systems, and services. Indeed, it is a new revolution in communication technology which means that everything, from tires to hairbrush, will be assigned a unique identifier so can be addressed, connected to other things and exchange information [1].

In this chapter, we will give some definitions about the Internet of Things and get to know their architecture,after that we will mention the components of iot , also we will present the principle of IoT and types of its protocols ,finally we will give some applications and the challenges faced by Internet of Things.

I.2 Definition of IoT:

The internet of things has many definitions, so we chose two simple definition that clarify their meaning:

- ITU-GSI has defined IoT as “the network of physical objects - devices, vehicles, buildings and other items - embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data”. The notion of IoT is to interconnect various things making use of electronic sensors. Its major goal is to get embedded and involved in every aspect of the real world. Attainment of this goal finally brings us where we ever want to reach, i.e., increased efficiency, availability and optimized use of resources [2].

- Internet of things (IoT) represents the connection between any devices with Internet including cell phone, home automation system and wearable devices . This new technology can be considered the phase changer of the health care applications concerning the patient’s health using low cost [3].

I.3 IOT architecture:

There is no single consensus on architecture for IoT, which is agreed universally. Different architectures have been proposed by different researchers.

The Three and Five-Layer Architectures. The most basic architecture is three-layer architecture as shown in figure I.1. It was introduced in the early stages of research in this area. It has three layers, namely, the perception, network, and application layers [4].

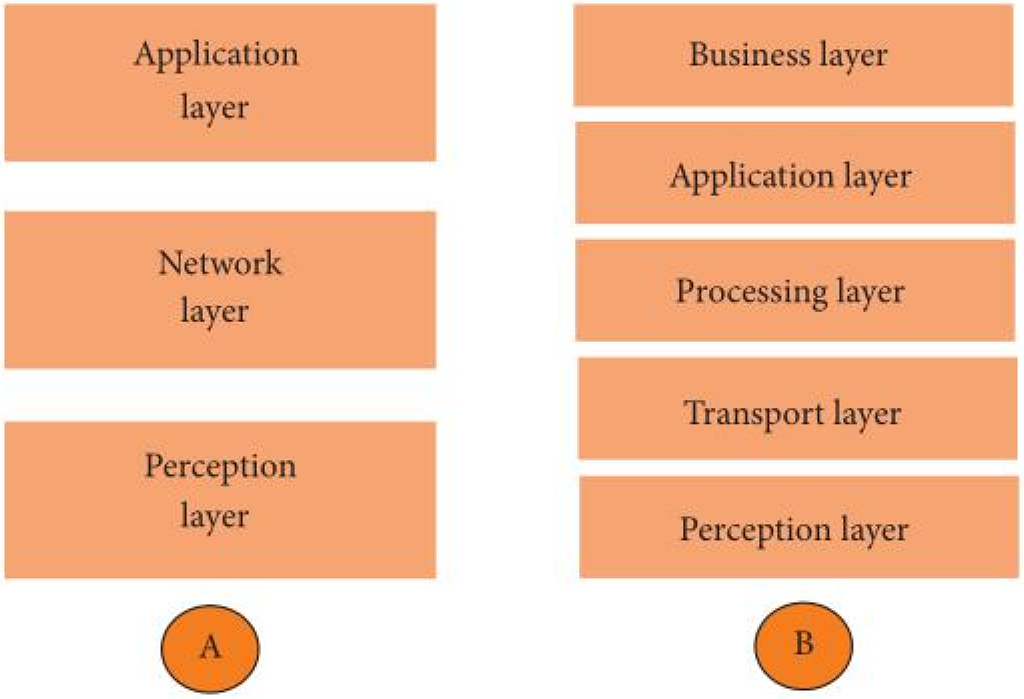


Figure I.1: Architecture of IoT: (A): three layers and (B): five layers.

- **The perception layer:** is the physical layer, which has sensors for sensing and gathering information about the environment. It senses some physical parameters or identifies other smart objects in the environment [4]. These smart sensors are continuously collecting data from the environment and transmit the information to the next layer [5].
- **The network layer:**The network layer is like the neural network and brain of loT, its main function is transmitting and processing information. The network layer includes a convergence network of communication and Internet network, network management center, information center and intelligent processing center, etc. The network layer will transmit and process the information obtained from perception layer [6].
- **The application layer:** is responsible for delivering application specific services to the user. It defines various applications in which the Internet of Things can be deployed [4]. The IoT application covers “smart” environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy [7].

The three-layer architecture defines the main idea of the Internet of Things, but it is not sufficient for research on IoT because research often focuses on finer aspects of the Internet of Things. That is why, we have many more layered architectures proposed in the literature. One is the five layer architecture, which additionally includes the processing and business layers. The five layers are perception, transport, processing, application, and business layers. The role of the perception and application layers is the same as the architecture with three layers. We outline the function of the remaining three layers [4].

- **The transport layer:** transfers the sensor data from the perception layer to the processing layer and vice versa through networks such as wireless, 3G, LAN,Bluetooth, RFID, and NFC [4].

- **The processing layer:** The processing layer is also known as the middleware layer. It stores, analyzes, and processes huge amounts of data that comes from the transport layer. It can manage and provide a diverse set of services to the lower layers. It employs many technologies such as databases, cloud computing, and big data processing modules [4].

- **The business layer:** The Business Layer is like a manager of the Internet of Things, including managing the applications, the relevant business model and other business. The Business Layer not only manages the release and charging of various applications, but also the research on business model and profit model. As we all known, success of a technology not only depends on the priority on technology, but also the innovation and reasonable of business model. Based on this piont, the Internet of Things can not have effective and long-term development without the research on business model. Meanwhile, this layer should manage the users' privacy which is equally important in the Internet of Things [6].

I.4 IoT components:

A complete IoT system integrates four distinct components: sensors/devices, connectivity, data processing, and a user interface. Below I will briefly explain each component and what it does [8].

1) Sensors/Devices: First, sensors or devices help in collecting very minute data from the surrounding environment. All of this collected data can have various degrees of complexities ranging from a simple temperature monitoring sensor or a complex full video feed.

A device can have multiple sensors that can bundle together to do more than just sense things. For example, our phone is a device that has multiple sensors such as GPS, accelerometer, camera but our phone does not simply sense things.

The most rudimentary step will always remain to pick and collect data from the surrounding environment be it a standalone sensor or multiple devices [9].

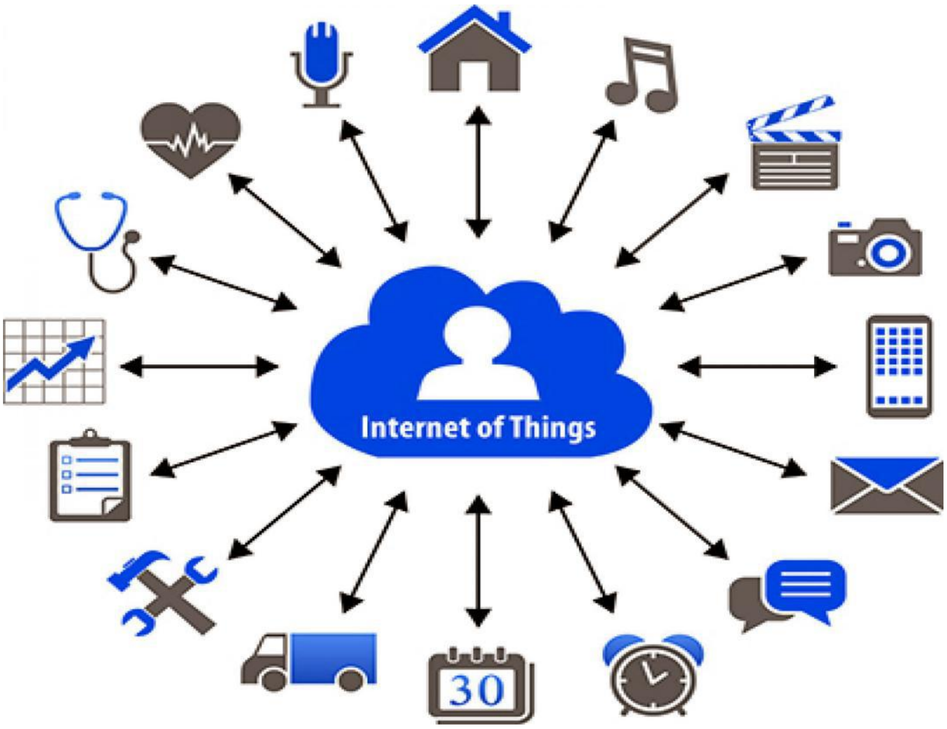


Figure I.2:IoT Components – Sensors/Devices

2) Connectivity: In connectivity, the data collected by sensors requires to send the cloud infrastructure, but it needs a medium to transport.

Once the data is collected, it is transferred to the cloud infrastructure which is also called as IOT platforms. These transactions need a medium like satellite networks, Wi-Fi, Bluetooth, wide-area networks (WAN).

The effectiveness of IoT security highly depends on the speed and availability of these mediums.

This medium is chosen by its specifications between power consumption, bandwidth, and range. So it is important to deploy the best connectivity option [10].

Figure I.3:IoT Components – Connectivity

3) Data Processing: Once the data gets to the cloud, software performs some kind of processing on it [8].

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly as not to negatively affect the end product, or data output.

Data processing starts with data in its raw form and converts it into a more readable format (graphs, documents, etc.), giving it the form and context necessary to be interpreted by computers and utilized by employees throughout an organization [11].

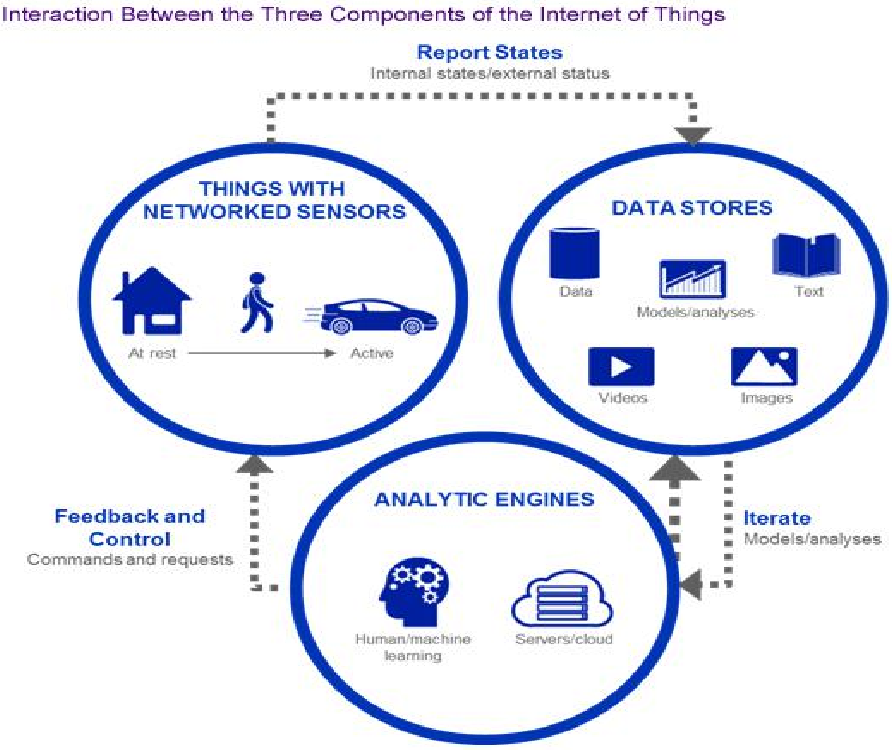


Figure I.4: IoT Components – Data Processing

4) User Interface: Next, the information is made useful to the end-user in some way. This could be via an alert to the user (email, text, notification, etc). For example, a text alert when the temperature is too high in the company's cold storage [8].

User interfaces are the visible, tangible part of the IoT system which can be accessible by users. Designers will have to make sure a well designed user interface for minimum effort for users and encourage more interactions [5].

I.5 Principle of Internet of Things:

An IoT system consists of sensors/devices which "talk" to the cloud through some kind of connectivity. Once the data gets to the cloud, software processes it and then might decide to perform an action, such as sending an alert or automatically adjusting the sensors/devices without the need for the user.

But if the user input is needed or if the user simply wants to check in on the system, a user interface allows them to do so. Any adjustments or actions that the user makes are then sent in the opposite direction through the system: from the user interface, to the cloud, and back to the sensors/devices to make some kind of change [8].

Humans can interact with the gadgets to set them up, give them instructions or access the data, but the devices do most of the work on their own without human intervention.
This is because the IOT enables things to connect and collect significant data from various sources to share the data over the internet to communicate with other devices [10].

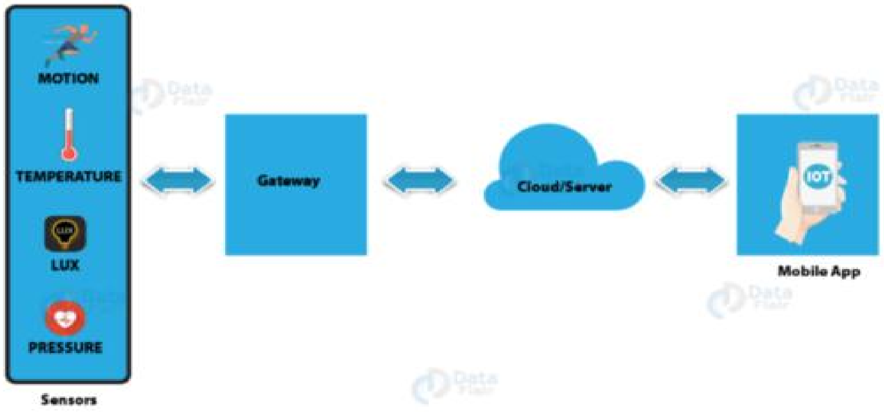


Figure I.5: Internet of Things Principle

I.6 Types of IoT Protocols:

IoT protocols and standards can be broadly classified into two separate categories as shown in Figure I.5 [12]:

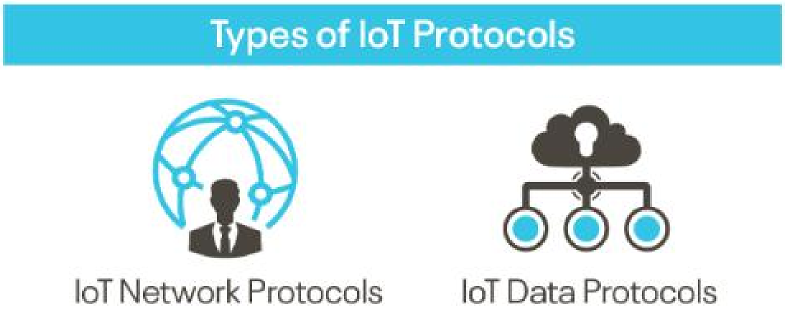


Figure I.6:Types of IoT protocols

I.6.1 IoT Network Protocols:

IoT network protocols are used to connect devices over the network. These are the set of communication protocols typically used over the Internet. Using IoT network protocols, end-to-end data communication within the scope of the network is allowed. Following are the various IoT Network protocols [12]:

1. HTTP (HyperText Transfer Protocol): HTTP is the main protocol used by World Wide Web for communication. HTTP defines how the messages are formatted and transmitted across the internet. HTTP protocol is based on client server model. A browser is like client because it is used to send request to server. Server then sends the response back to the client. The default port for the server to listen for the request is 80. HTTP protocol is a request/response stateless protocol . Main function of HTTP is to transmit resources across the internet. A resource can be a file, A CGI script, or a document written in any available languages. The format of the request and response message is very much similar.An HTTP request has mainly three parts: a) request line, b) HTTP header, and c) an optional HTTP body [13].

2. LoRaWan (Long Range Wide Area Network): The LoRaWAN is a Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated ‘things’ to the internet in regional, national or global networks, and targets key Internet of

Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services [14].

Figure I.7:LoRaWan logo

3. Bluetooth: This is one of the most widely used protocol in short-range communication WPAN (Wireless Personal Area Network). It plays an important role among IoT network protocols. It is a standard for wireless data transmission. It plays an important role in connecting wearable devices. The BLE (Bluetooth Low Energy) is a low-energy version of the Bluetooth protocol that reduces the power consumption and plays an important role in connecting devices [15].

Figure I.8:Bluetooth logo

4. ZigBee: is an IoT protocol that allows smart objects to work together [12].The ZigBee Alliance was arguably the first to finalize a complete protocol stack for low-power mesh networks. The ZigBee protocol stack is rooted entirely in the legacy IEEE802.15.4-2003 and IEEE802.15.4-2006 MAC protocol. It differentiates between full-function devices (FFD) and reduced-function devices (RFD), where the former are typically used as routing nodes, the latter as non-routing leaf nodes. A ZigBee network consists of a number of FFDs collecting data from RFDs located around it [16].

Figure I.9:Zegbee logo

I.6.2 IoT Data Protocols:

IoT data protocols are used to connect low power IoT devices. These protocols provide point-to-point communication with the hardware at the user side without any Internet connection. Connectivity in IoT data protocols is through a wired or a cellular network. Some of the IoT data protocols are [12]:

1. Message Queue Telemetry Transport (MQTT): MQTT is one of the most commonly used protocols concerning IoT. MQTT enables resource-constrained IoT devices to send, or publish, information about a given topic to a server that functions as an MQTT message broker. The broker then pushes the information out to those clients that have previously subscribed to the topic. To a human, a topic looks like a hierarchical file path. Clients can subscribe to a specific level of a topic's hierarchy or use a wild-card character to subscribe to multiple levels [17].

2. Constrained Application Protocol (CoAP): CoAP is a software protocol that enables simple constrained “things” such as low-power sensors and actuators to communicate interactively via the internet. It runs on devices that support the User Datagram Protocol (UDP), and implements a lightweight” application layer that features small message sizes, message management and lightweight message overhead ideally suited for low-power, low-memory devices [18] .

3. Advanced Message Queuing Protocol (AMQP): is an application layer protocol that focuses on process-to-process communication across IP networks. An encoding schema and a set of procedures allow for two different servers to communicate regardless of the technology used. Overall, the goal of AMQP is to enable message passing through broker services over TCP/IP connections. AMQP is considered a compact protocol, since it’s a binary protocol, meaning that everything sent over AMQP is binary data. A binary protocol avoids sending useless data over the wire [19].

3. Machine-to-Machine (M2M) Communication Protocol: It is an open industry protocol built to provide remote application management of IoT devices. M2M communication protocols are cost-effective and use public networks. It creates an environment where two machines communicate and exchange data. This protocol supports the self-monitoring of machines and allows the systems to adapt according to the changing environment.

M2M communication protocols are used for smart homes, automated vehicle authentication, vending machines, and ATM machines [12].

4. Extensible Messaging and Presence Protocol (XMPP): XMPP is a short form for Extensible Messaging Presence Protocol. It’s protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time. This protocol is mostly used by instant messaging applications like WhatsApp.

Let’s dive into each character of word XMPP:

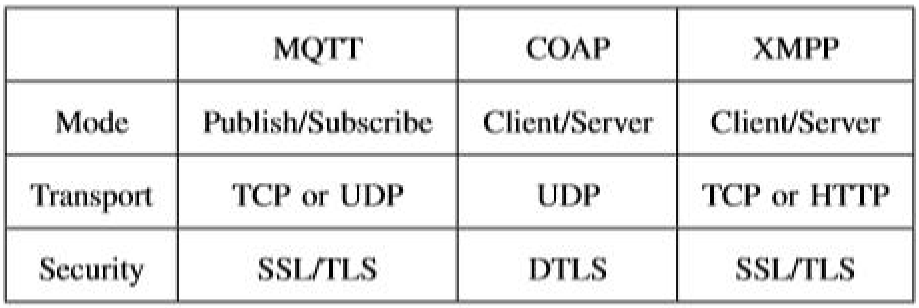
X : It means eXtensible. XMPP is a open source project which can be changed or extended according to the need.

M : XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.

P : It determines whether you are online/offline/busy. It indicates the state.

P : XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other [20].

-Comparison of Messaging Protocols:



	MQTT	COAP	XMPP
Mode	Publish/Subscribe	Client/Server	Client/Server
Transport	TCP or UDP	UDP	TCP or HTTP
Security	SSL/TLS	DTLS	SSL/TLS

Table I.1: Comparison of Messaging Protocols

The above table summarizes the mode, transport and security of the three messaging protocols, MQTT, COAP and XMPP. Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), both frequently referred to as “SSL”, are cryptographic protocols that provide communications security over a computer network. Both MQTT and XMPP adapt this secure connection system to ensure reliable security. MQTT relies on TCP as transport protocol, which means by default the connection does not use an encrypted communication. To encrypt the whole MQTT communication, most of the MQTT brokers allow the use of TLS instead of plain TCP. The same is the case with XMPP protocol.

On the other hand, In information technology, the Datagram Transport Layer Security (DTLS) communications protocol provides communications security for datagram protocols. DTLS allows datagram-based applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. CoAP uses datagram transport layer security (DTLS) as the security protocol for communication and authentication of communicating devices [21].

I.7 IoT Applications:

Potential applications of the IoT are numerous and diverse, permeating into practically all areas of every-day life of individuals, enterprises, and society as a whole. The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy. Below are some of the IOT applications [7].

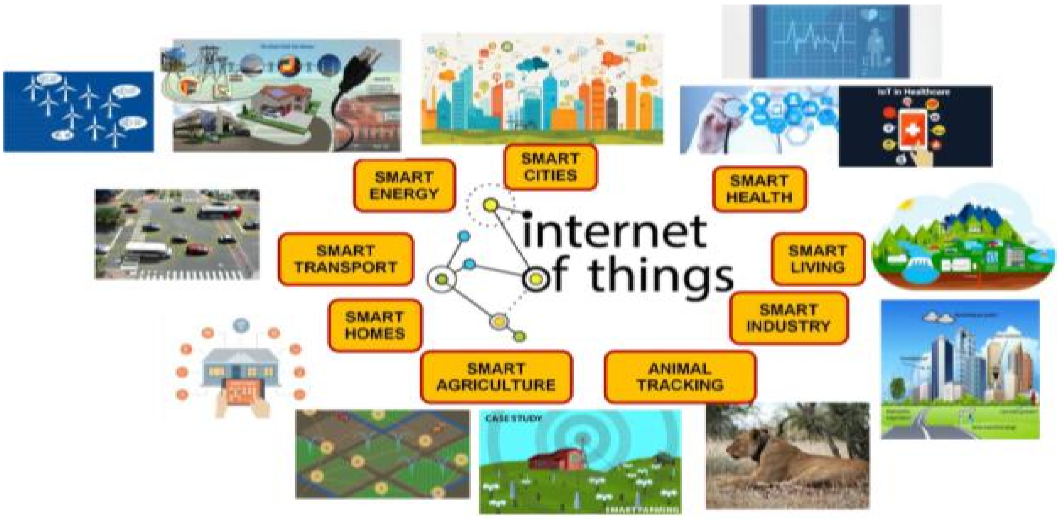


Figure I.10: IoT applications

1. **Smart home:** Smart Home clearly stands out, ranking as highest Internet of Things application on all measured channels. More than 60,000 people currently search for the term "Smart Home" each month. This is not a surprise. The IoT Analytics company database for Smart Home includes 256 companies and startups. More companies are active in smart home than any other application in the field of IoT. The total amount of funding for Smart Home startups currently exceeds $2.5bn. This list includes

prominent startup names such as Nest or AlertMe as well as a number of multinational corporations like Philips, Haier, or Belkin [22].

2. **Smart City:** The smart city like the name suggests is a very big innovation and spans a wide variety of use cases, from water distribution to traffic management to waste management, environmental monitoring, and urban security. The reason why it is so popular is that it tries to remove the discomfort and problems of people who live in cities. IoT solutions offered in the Smart City area solve various city-related problems comprising of traffic, reduce air and noise pollution and help make cities safer [23].

3. **Smart grids:** The Smart Grid represents an unprecedented opportunity to move the energy industry into a new era of reliability, availability, and efficiency that will contribute to our economic and environmental health. During the transition period, it will be critical to carry out testing, technology improvements, consumer education, development of standards and regulations, and information sharing between projects to ensure that the benefits we envision from the Smart Grid become a reality [24].

4. **Industrial Automation:** Automation in industry is very critical, as there are harsh and unpleasant conditions for human beings. Instead of following manual processes, when taking a step further with automated control – by connecting the systems and controlling over the Internet, it gives great control for the employer/manager and monitoring since you can monitor not only from the company's premise but from anywhere. This facilitates productivity and ensures tasks/people can be monitored simultaneously [2].

5. **Connected Health (Digital health/Telehealth/Telemedicine):** IoT has various applications in healthcare, which are from remote monitoring equipment to advance & smart sensors to equipment integration. It has the potential to improve how physicians deliver care and also keep patients safe and healthy. Healthcare IoT can allow patients to spend more time interacting with their doctors by which it can boost patient engagement and satisfaction. From personal fitness sensors to surgical robots, IoT in healthcare brings new tools updated with the latest technology in the ecosystem that helps in developing better healthcare. IoT helps in revolutionizing healthcare and provides pocket-friendly solutions for the patient and healthcare professional [23].

I.8 The Challenges Faced by Internet of Things:

1. Managing title and definition of things: Due to the large number of things, a good management system is required.

2. Standard specification: Provision of standards for the purpose of providing interoperability between the different devices and objects in the Internet of Thing applications.

3. Security, privacy and data reliability: Sensitive and aggregated data must be protected from attacks, violations and data leakage. Data encryption is essential to ensure data integrity and reliability when processed, transmitted and stored.

4. Big data size: Internet of Things is a source of large data sets due to the large number of connected devices and these big data require attention in transfer, storage and analysis.

5. Protection of objects: As a result of the large number of entities connected to the Internet of Things that are distributed over a wide geographical area, it is necessary to prevent subversion or unwanted changes by intruders and hackers.

6. Local and community considerations: Internet of Things can be global, so service providers should be familiar with the various local and international laws and must protect the rights of users [25].

I.9 Conclusion:

Throughout this chapter, we provided an overview about internet of things so that we simplified it's concept also we presented IoT architectures, and noted that there are two types of architecture most used. We talked about the different elements and the principle of IoT, also we classified the protocols into two types. Finally, we got to know the internet of things applications and challenges that you will face.

Chapter II

Mobile Robot

II.1 Introduction:

As technology advances in the last tow decades, it has been envisioned that in the very near future, robotic systems will become part and parcel of our everyday lives. robots are machines intended to replace Man in different applications such as surgery .

In this chapter, we will try to provide a simple explanation about mobile robots, their classifications and types , in the end, we will give the definition of embedded systems and its relationship with robotics.

II.2 History of mobile robot:

Robotics currently represents one of mankind's greatest accomplishments and is the single greatest attempt of mankind to produce an artificial, sentient being. It is only in recent years that manufacturers are making robotics increasingly available and attainable to the general public.

The focus of this timeline is to provide the reader with a general overview of robotics (with a focus more on mobile robots):[1]

1939-1945: During World War II the first mobile robots emerged as a result of technical advances on a number of relatively new research fields like computer science and cybernetics. They were mostly flying bombs. Examples are smart bombs that only detonate within a certain range of the target, the use of guiding systems and radar control [2].

1948: W. Grey Walter created his first robots; Elmer and Elsie, also known as the turtle robots. The robots were capable of finding their charging station when their battery power ran low [1].

1954: George Devol designed the first truly programmable robot and called it UNIMATE for "Universal Automation." (US patent 2 998 237) [1].

1960's: One of the first operational, industrial robots in North America appeared in
the early 1960's in a candy factory in Kitchener, Ontario [1].

1966–1972: the Stanford Research Institute is building and doing research on Shakey the Robot, a robot named after its jerky motion. Shakey had a camera, a rangefinder, bump sensors and a radio link. Shakey was the first robot that could reason about its actions. This means that Shakey could be given very general commands, and that the robot would figure out the necessary steps to accomplish the given task [2].

Early 1980s: The team of Ernst Dickmanns at Bundeswehr University Munich builds the first robot cars, driving up to 55 mph on empty streets [2].

1990s: Joseph Engelberger, father of the industrial robotic arm, works with colleagues to design the first commercially available autonomous mobile hospital robots, sold by Helpmate [2].

1999: Sony introduces Aibo, a robotic dog capable of seeing, walking and interacting with its environment [2].

2002: Nevena Bozinovska, Gjorgi Jovancevski, and Stevo Bozinovski carried out Internet-based robot control.A mobile robot in United States was controlled by students in Europe [2].

2016: The Multi-Function Agile Remote-Controlled Robot (MARCbot) is for the first time used by US police [2].

II.3 definition of mobile robot:

A mobile robot is a machine controlled by software that use sensors and other technology to identify its surroundings and move around its environment. Mobile robots function using a combination of artificial intelligence (AI) and physical robotic elements, such as wheels, tracks and legs. Mobile robots are becoming increasingly popular across different business sectors. They are used to assist with work processes and even accomplish tasks that are impossible or dangerous for human workers [3].

II.4 classification of robot:

II.4.1 Classification of robots by environment and mechanism of interaction:

Robots can be classified according to the environment in which they operate.The most common distinction is between fixed and mobile robots. These two types of robots have very different working environments and therefore require very different capabilities [4].

- **Fixed robots:** are mostly industrial robotic manipulators that work in well defined environments adapted for robots. Industrial robots perform specific repetitive tasks such soldering or painting parts in car manufacturing plants [4].
- **Mobile robots:** Mobile robots have the capability to move around in their environment and are not fixed to one physical location [5]. Such environments can include unpredictable entities like humans and animals. Examples of mobile robots are robotic vacuum cleaners and self-driving cars [4].

There are three main environments for mobile robots that require significantly different design principles because they differ in the mechanism of motion: aquatic(underwater exploration), terrestrial (cars) and aerial (drones). Again, the classification is not strict, for example, there are amphibious robots that move in both water and on the ground. Robots for these three environments can be further divided into subclasses: terrestrial robots can have legs or wheels or tracks, and aerial robots can belighter-than air balloons or heavier-than-air aircraft,which are in turn divided into fixed-wing and rotary-wing (helicopters) [4].

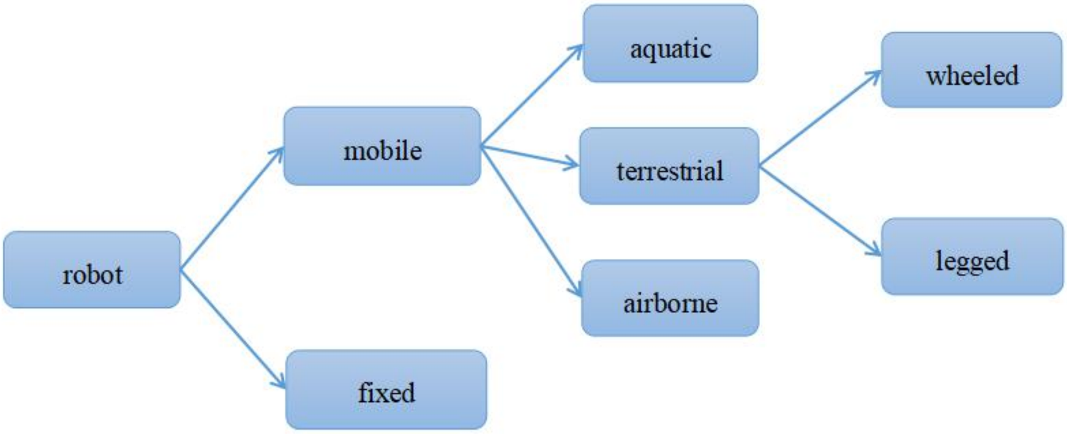


Figure II.1:Classification of robots by environment and mechanism of interaction

II.4.2 Classification of robots by application field:

Robots can be classified by intended application field and the tasks they perform . We mentioned industrial robots which work in well-defined environments on production tasks. The first robots were industrial robots because the well-defined environment simplified their design.Service robots,on the other hand,assist humans in their tasks. These include chores at home like vacuum clears, transportation like self-driving cars,and defense applications such as reconnaissance drones.Medicine,too, has seen increasing use of robots in surgery, rehabilitation and training. These are recent applications that require improved sensors and a closer interaction with the user [4].

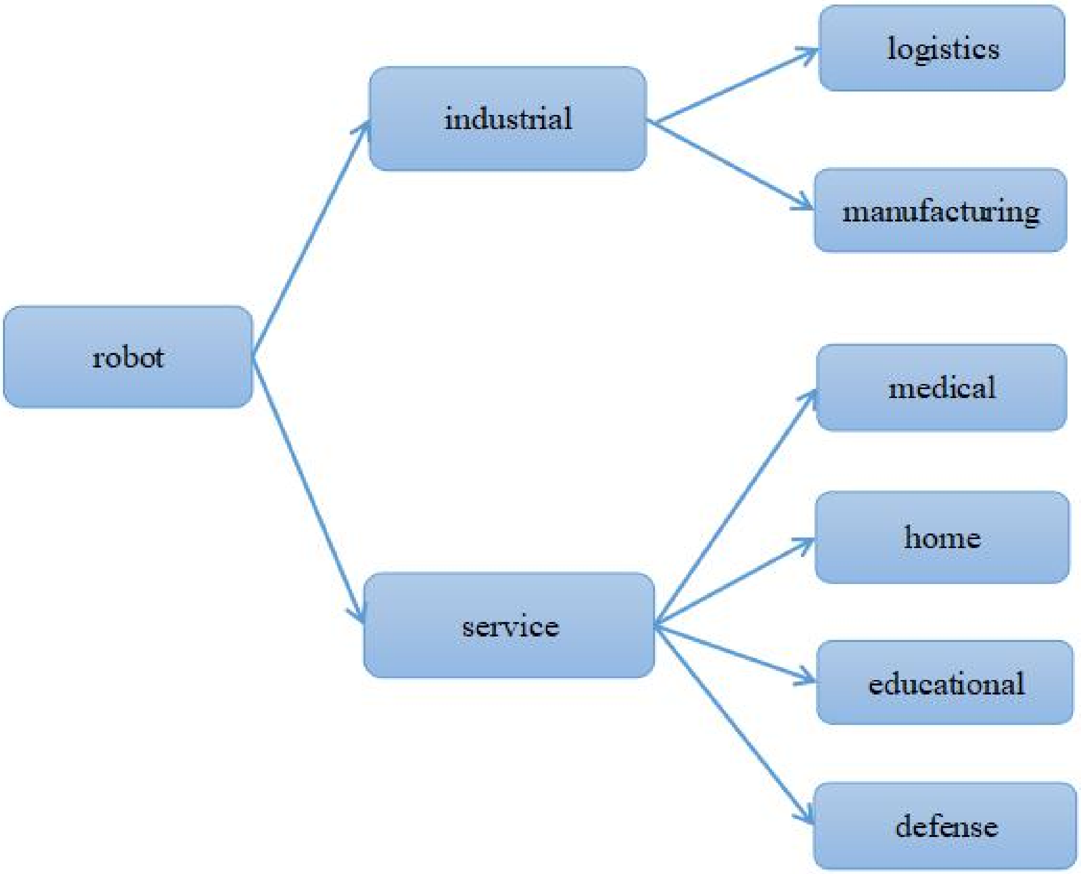


Figure II.2:Classification of robots by application field

II.5 Types of robots:

There are two possible ways for divide robots. First,types of robot by their application and second by the way they move (or don't):

II.5.1 Types of robots by application:

1. Industrial robots: These types of robots come with an arm used to carry out recurring tasks[6]. Usually these are articulated arms specifically developed for such applications as welding, material handling, painting and others [7].

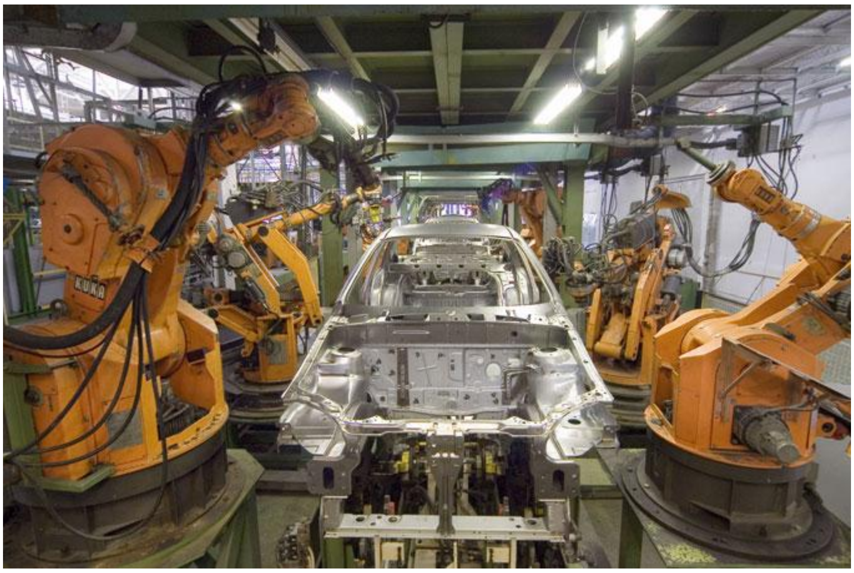


Figure II.3: Robots on an assembly line in a car factory

2. Domestic or household robots: are robots programmed to do household works, educate or entertain [8] for example: robotic vacuum cleaners, robotic pool cleaners, sweepers, gutter cleaners and other robots that can do different chores.Also, some surveillance and telepresence robots could be regarded as household robots if used in that environment [7].

Figure II.4:Robot Vacuum Cleaner

3. Medical robots: there has been an increasing use of robots in the scientific discipline for surgery, rehabilitation and training. In robotic surgery,the doctor controls the robotic arm using a very small tool that is attached to the robotic arm via a control computer. In such case the general practitioner makes small cuts to insert the units into the human body. A skinny tube attached to the camera in the front is used to capture the enlarged real-time picture of the phase of the body the place the surgical procedure will take place. The robot matches the movement of the hand of the surgeon, and the software of the laptop tunes the precision tow of the movement of the hand such that the surgical procedure will become especially accurate [9].

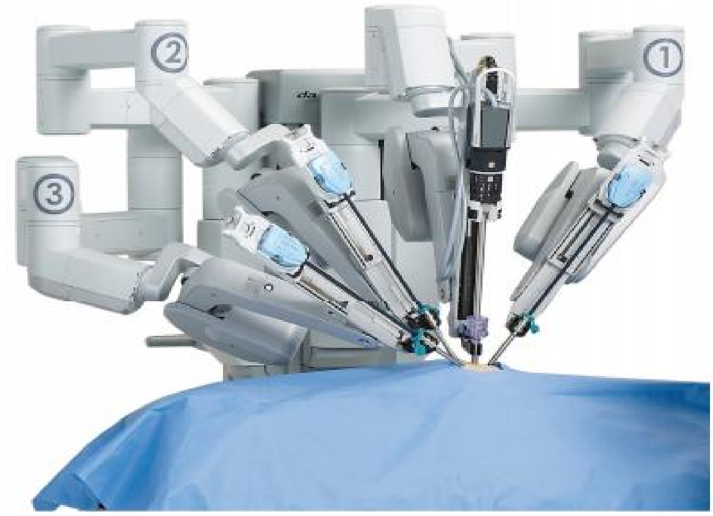


Figure II.5: A Robot Performing Surgery

4. Military robots: Military robots are related to very different applications and differ to those robots that are used for industrial production, in a way that they do not produce things, but have to interact in warfare. The concept, however, is the same; a device has to fulfill some tasks automatically to qualify as a robot. In military terms, technology focuses on the control of missiles and vehicles in order to have unmanned devices that are either tele-operated or that find their way, automatically guided by laser beams or GPS satellites. Today, military robots can be divided into several categories: guided missiles; military spacecraft; unmanned aerial vehicles (UAV); unmanned ground vehicles (UGV); remotely operated vehicles (ROV); and autonomous underwater vehicles (AUV) [10].

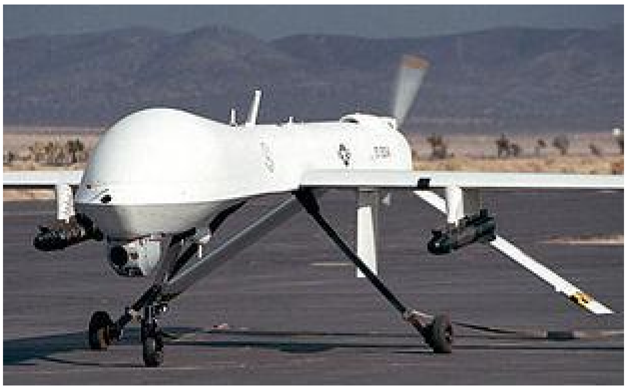


Figure II.6:Armed Predator drone

II.5.2 Types of robots by locomotion and kinematics:

1. Stationary robots: Stationary robots are those that perform their task without changing positions. The term “stationary” is more associated with the base of the robot and not the whole robot. The robot moves above the base to perform the desired operation. These robots manipulate their environment by controlling the position and orientation of an end-effector. End-effectors could be a drilling, welding, or gripper device [11].
2. Flying robots: are useful in search and rescue missions as they can be used to survey large areas of land looking for victims. By using sensors on the robots, flying robots can be sent into areas where it is too dangerous or unfeasible to send human or canine rescue teams, and relay information back to a central hub, who can then direct rescue teams to where they are most needed.

In the field of flying robotics, big research questions include the design of: novel hardware than can resist collisions and is easy to transport, new control methods for autonomous flight and innovative sensors for safe navigation [12].

Figure II.7:Drone flying in the forest

3. Swimming robots: Aquatic robots can sail, submerge or crawl underwater, Robotic fish can be used during human-induced ecological disasters that are affecting life in aquatic environments, such as oil spills, and man-made structures, such as dams, Driving force on aquatic robots can be tails, fins, thrusters, wings, thrusters, paddles, paddle wheels, air pumps etc, based on your robot design [13].

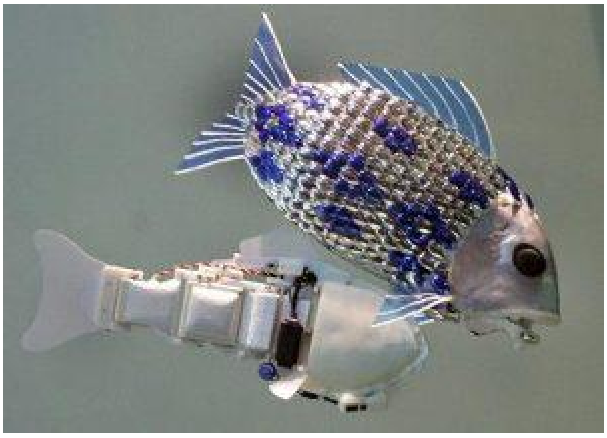


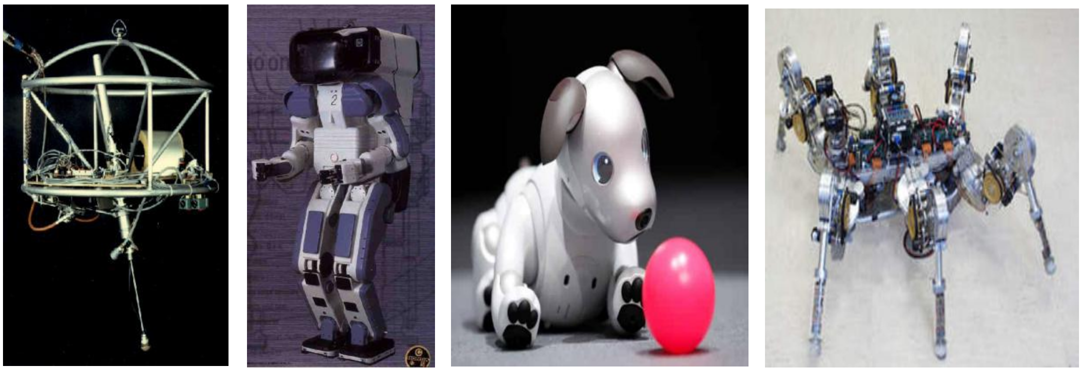
Figure II.8:Fish robots

4. Legged robots: Legged Robots use legs to move from one place to another. Legs provide an active suspension , so the motion of the main body of the robot can be largely decoupled from the terrain profile. With each step, a leg is temporarily lifted off the ground, so that discontinuous terrain can be overcome as well, allowing locomotion in places out of reach otherwise.

Legs are usually articulated rigid bodies, assumed to contact the environment only with their end effector. In most cases, this contact is unilateral, meaning that the robot can push but not pull on contact surfaces. In some cases, grasping, suction cups, magnets, adhesive materials or miniature spine arrays can provide additional grip.

Adapting wheeled vehicles to rough terrain has led in some cases to implant wheels on legs, with any combination of passive or active wheels, passive or active legs, combining the flexibility of articulated legs on rough terrain with the efficiency of wheels on flat terrain . On steep slopes, legged robots can also use rappelling to avoid tumbling down .

Legged robots are composed of a central body (also called trunk or torso) with legs attached to it. Most common are monopods, bipeds (e.g. humanoids), quadrupeds and hexapods, with one, two, four and six legs, respectively. Less common are robots with three, five, seven or more legs [14].



(a) (b) (c) (d)

Figure II.9: (a): One Leg Hopper, (b): The Humanoid Robot (two legs), (c): Aibo (dog robot four legs), (d): Lauron (six legs).

5. Wheeled robots: A typical wheeled mobile robot can be regarded as a multibody system formed by three elemental modules, i.e. wheel module, suspension module and chassis (base) module. Each elemental module can be constructed by different mechanisms or rigid/articulated structures [15].

- **Types of Robot Wheels:** There are several different types of wheels available for robots, each with their own advantages and disadvantages.

There are four (4) Types of Wheels for Robots as shown in this table [16]:

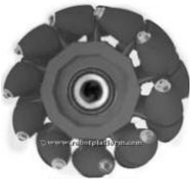
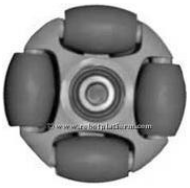
Standard Wheels	Caster and Ball Wheels	Mecanum Wheels	Omni Wheels
-Most common type -Two degrees of freedom (forward or in reverse) -Wide range of sizes and mounts -Can be used as drive or idler wheel -Selection of tire treads 	-Orientable wheels (they can travel in just about any direction) -Used as idler wheel or for balance -Mostly used for indoor designs 	-Multi-directional wheels -Used as drive or idler wheels -Rollers at angle, usually 45° -Different left and right wheels -Used indoors or outdoors 	-Wheels with rollers at 90° -Four degrees of freedom -Used as drive or idler wheels -Best for indoor use 

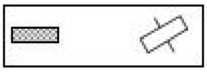
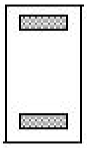
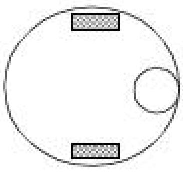
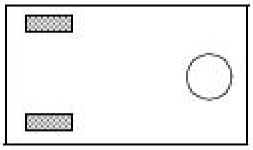
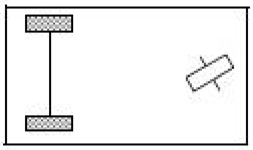
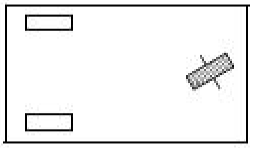
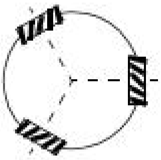
Table II.1: The Four Basic Wheel Types

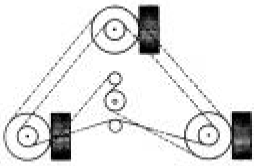
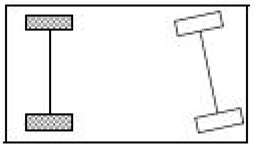
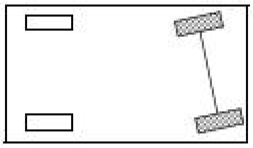
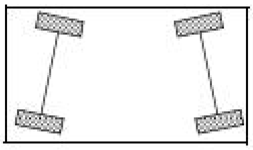
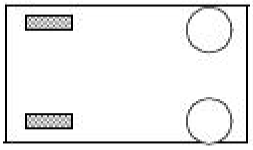
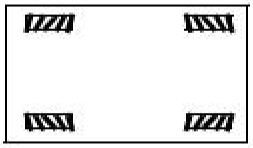
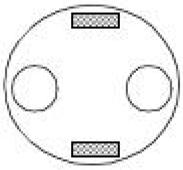
- **wheel configurations ordered by the number of wheels:**

Table II.1 gives an overview of wheel configurations ordered by the number of wheels. This table shows both the selection of particular wheel types and their geometric configuration on the robot chassis [17].

Icons for the each wheel type are as follows:

○	unpowered omnidirectional wheel (spheric, castor, swedish)
▨	motorized swedish wheel (Stanford wheel)
▭	unpowered standard wheel
▤	motorized standard wheel
▤○	motorized and steered castor wheel
╪	steered standard wheel
⌶	connected wheels

number of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	bicycle, motorcycle
		Two-wheel differential drive with the COM below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRobII EPFL
		Two independently driven wheels in the rear/front, one unpowered omnidirectional wheel in the front/rear	many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, one steered free wheel in front	Piaggio mini-trucks
		Two free wheels in rear, one steered traction wheel in front	Neptune (Carnegie-Mellon University, Hero-1
		3 motorized swedish or spheric wheels arranged in a triangle. Omnidirectional movement is possible.	Stanford-wheel Tribolo EPFL, Palm Pilot Robot Kit

		3 synchronously motorized and steered wheels. The orientation is not controllable.	‘synchro drive’ Denning MRV-2, Georgia Institute of Technology, IRobot B24, Nomad 200
4		2 motorized wheels in the rear, 2 steered wheels in the front; Steering has to be different for the two wheels to avoid slipping/ skidding.	car with rear wheel drive
		2 motorized and steered wheels in the front, 2 free wheels in the rear; Steering has to be different for the two wheels to avoid slipping/skidding.	car with front wheel drive
		4 steered and motorized wheels	four wheel drive, four wheel steering
		Two traction wheels (differential) in rear/front, two omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheel	CMU Uranus
		Two wheel differential drive with two additional points of contact	EPFL Khepera, Hyperbot Chip

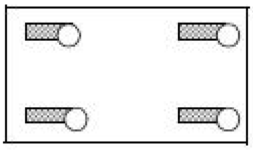
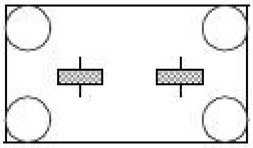
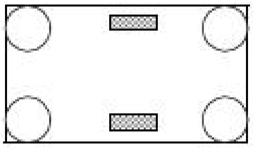
		Four motorized and steered castor wheels	Nomadic XR4000
6		Two motorized and steered wheels aligned in center, one omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, one omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)

Table II.2 : Wheel configurations for rolling vehicles[17]

- **wheel control theory:** there are different types of wheels and their arrangements. Once you have decided on how many and what type of wheels will robot will have, we will need to put in a plan on how to control them. Below described are few control mechanisms to drive and steer your robot [18].

1. **Differential Drive/Differential wheel:** This is the most common control mechanism for robot builders [18], The differential-drive mobile robot (DDMR) has a two wheeled , where each wheel is driven independently. Forward motion is achieved when both wheels are driven at the same rate, turning right is achieved by driving the left wheel at a higher rate than the right wheel and vice-versa for turning left. This type of mobile robot can turn on the spot by driving one wheel forward and second wheel in opposite direction at the same rate. Third wheel is a castor wheel needed for the stability of mobile robot [19].

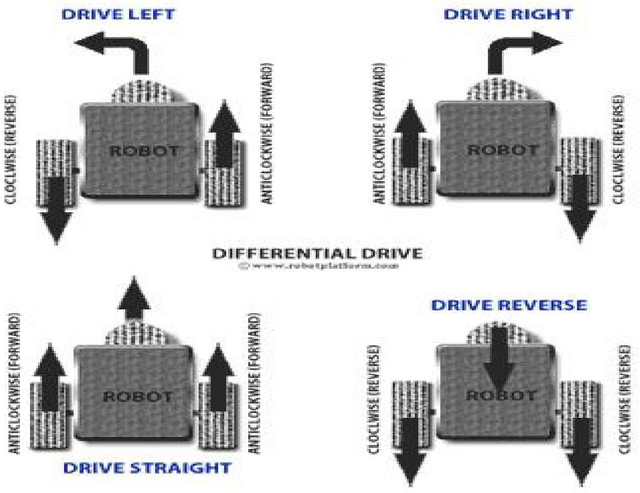


Figure II.10: Differential Drive

2. **Tricycle Drive:** Tricycle robot is designed with a front steering wheel controlled by a motor. The two rear wheels are attached to a common axle driven by a single motor with two degrees of freedom (either forwards or reverse). this robot can't spin like a differential drive robot and does not have a 90° turn due to their limited radius of curvature. Few robots have both steer and drive controlled by the front wheel and the rear wheels act as supporting wheels to maintain equilibrium [18].

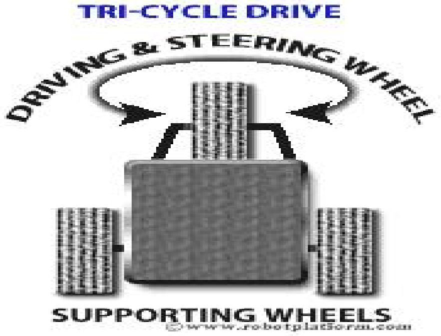


Figure II.11 : Tricycle drive

3. **Ackermann Steering:** The main purpose of a perfect Ackermann system is to reduce tyre scrub when cornering by making the inside wheel turn on a tighter radius than the outer wheel. Without this the inside tyre would have too much slip angle to turn and would be forced to give way by scrubbing the tyre. The tyre scrub effectively slows the car down and reduces grip in the scrubbing tyre[20]. The advantage in this design is increased control, better stability and maneuverability on road, less slippage and less power consumption [18].

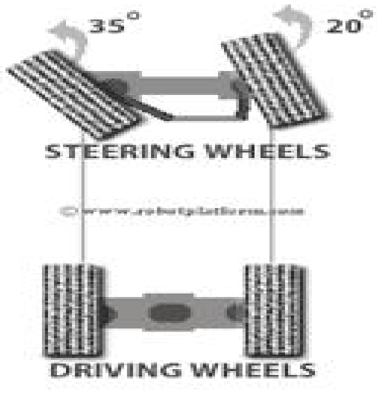


Figure II.12 :Ackermann Steering

II.6 Embedded systems of Robotics:

II.6.1 Definition of embedded system:

An embedded system is a combination of computer hardware and software designed for a specific function or functions within a larger system. The systems can be programmable or with fixed functionality. Industrial machines, consumer electronics, agricultural and process industry devices, automobiles, medical equipment, cameras, as well as mobile devices, are possible locations for an embedded system [21].

II.6.2 Embedded systems and Robotics:

Building upon the foundation of software engineering, a degree in Robotics and Embedded Systems can span mechanical design, digital logic design, embedded programming, machine vision and adaptive algorithm development and design of autonomous robotic systems.we will implement microprocessors and optimize robotics technology using digital hardware design, software programming, and digital and analog circuit design analysis.

we will also apply knowledge of transducers, actuators, and advanced software development to mechanically design and engineer as well as analyze advanced robotics technology and embedded systems [22].

II.7 Conclusion:

In this chapter we talked about mobile robots and their evolution over time, after that we tried to simplify the concept of mobile robot , also we gave the different classifications and types of robotics by locomotion and kinematics and their application. Finally, we presented the definition of embedded systems and showed their relationship with robotics.

Chapter III

The Raspberry Pi

III.1 Introduction:

Computers embedded under the Linux operating system are massively present in modern technologies (transport, multimedia, mobile phone, cameras ...), among this computers we find the Raspberry Pi.

In this chapter we will provide a brief history of Raspberry Pi, after that we will give current models of Raspberry Pi and we explain the difference between them, also we will list components of Raspberry pi 3 and its different operating systems, in the end we will see some examples about applications of raspberry pi.

III.2 History of raspberry pi:

Raspberry Pi- A name which is now not a new technical buzzword for researchers in the area of Computer Science, Electronics & Embedded Systems engineering worldwide.

The foundation seed for development for Raspberry Pi journey started in 2006, when researchers named: Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft at University of Cambridge's Computer Laboratory became stunned to see the decline in the skill level of A Level students and students applying for computer science. The main idea behind their stepping stone development was to give kids tiny and affordable computer in the period where computers were expensive and programming practice among kids was not supported by parents of children in U.K.

The team lead by Eben Upton developed several versions of working prototypes from year 2006 to 2008 and the final released version was named as "RASPBERRY Pi".

In 2008, the processors used in mobile devices were becoming cheaper and powerful and have full potential to support and run multimedia and all sorts of programming.

The Raspberry Pi was officially created in year 2012 (February 2012) and the developing stone was laid by Raspberry Pi Foundation and within 3 years, the model B entered mass level production with Element 14 and RS Electronics and within 2 years of official launch of Pi, 2 million pieces are sold till date [1].

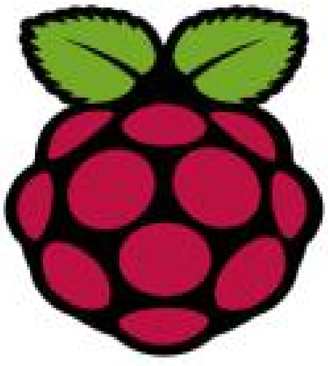


Figure III.1 :Raspberry Pi Logo

III.3 Current models of Raspberry Pi:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse [2].

The Raspberry Pi Foundation works to put the power of computing and digital making into the hands of people all over the world. It does this by providing low-cost, high-performance computers that people use to learn, solve problems, and have fun [3].

There are different models of Raspberry Pi, among them:

- Raspberry Pi 4 Model B: The Raspberry Pi 4 Model B (Pi4B) is the first of a new generation of Raspberry Pi computers supporting more RAM and with significantly enhanced CPU, GPU and I/O performance; all within a similar form factor, power envelope and cost as the previous generation Raspberry Pi 3B+.The Pi4B is available with either 1, 2 and 4 Gigabytes of LPDDR4 SDRAM [4].
- Raspberry Pi 3 Model B+: The Raspberry Pi 3 Model B+ was launched in March 2018. It uses a 1.4GHz 64-bit quad-core Arm Cortex-A53 CPU, has 1GB RAM, gigabit Ethernet, integrated 802.11ac/n wireless LAN, and Bluetooth 4.2 [5].
- Raspberry Pi 3 Model B: The Raspberry Pi 3 Model B is the third generation Raspberry Pi [6], and it was launched in February 2016. It uses a 1.2GHz 64-bit quad-core Arm Cortex-A53 CPU, has 1GB RAM, integrated 802.11n wireless LAN, and Bluetooth 4.1 [5].
- Raspberry Pi 2 Model B: Announced on 2nd Feb 2015, the Raspberry Pi 2 Model B v1.2 board uses BCM2837, it has which contains a 1 GB SDRAM - shared by the GPU and CPU and quad-core ARM Cortex-A53 processor [7].

- in the table 3.1 we will give simple comparision between the raspberry pi models

models of raspberry pi	CPU	RAM	Soc type	SD/MMC	Ethernet	WI-FI	Bluetooth
Raspberry Pi 4 Model B	1.5GHz Quad-Core	1GB 2GB 4GB	Broadcom BCM2711	microSD	Gigabit	2.4GHz and 5GHz 802.11 b/g/n/ac	Bluetooth 5.0

Raspberry Pi 3 Model B+	1.4GHZ Quad-Core	1GB	Broadcom BCM2837B0	microSD	Gigabit - Over USB 2.0	2.4GHz and 5GHz 802.11 b/g/n/ac	Bluetooth 4.2
Raspberry Pi 3 Model B	1.4GHZ Quad-Core	1GB	Broadcom BCM2837	microSD	RJ45 10/100 MBit/s	802.11n	Bluetooth 4.1
Raspberry Pi 2 Model B	900MHZ Quad-Core	1GB	Broadcom BCM2836	microSD	RJ45 10/100 MBit/s	No existed	No existed

Table III.1: Raspberry PI models comparison

III.4 The Raspberry Pi’s components:

Like any computer, the Pi is made up of various different components, each of which has a role to play in making it work [8].

In my project i will use the Raspberry Pi 3 model B, and its component are shown in the figure III.2:

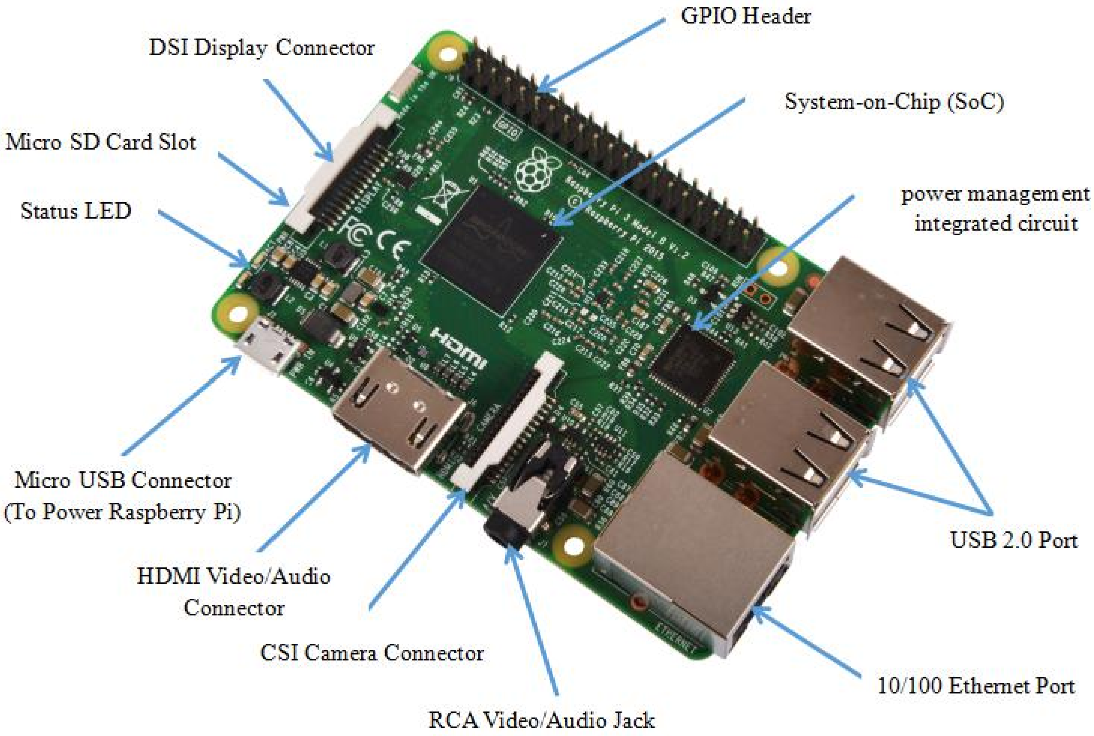


Figure III.2: Raspberry pi 3 Components

III.4.1 The system-on-chip (SoC):

The name system-on-chip is a great indicator of what you would find if you prised the metal cover off: a silicon chip, known as an integrated circuit, which contains the bulk of the Raspberry Pi's system. This includes the central processing unit (CPU), commonly thought of as the 'brain' of a computer, and the graphics processing unit (GPU), which handles the visual side of things [9].

Figure III.3: system-on-chip (SoC)

III.4.2 Random access memory (RAM):

On the underside of the Raspberry Pi you'll find exactly that: another chip, which looks like a small, black, plastic square . This is the Pi's random access memory (RAM). When you're working on the Pi, it's the RAM that holds what you're doing; only when you save your work will it be written to the microSD card. Together, these components form the Pi's volatile and non-volatile memories: the volatile RAM loses its contents whenever the Pi is powered off, while the non-volatile micro SD card keeps its contents [9].

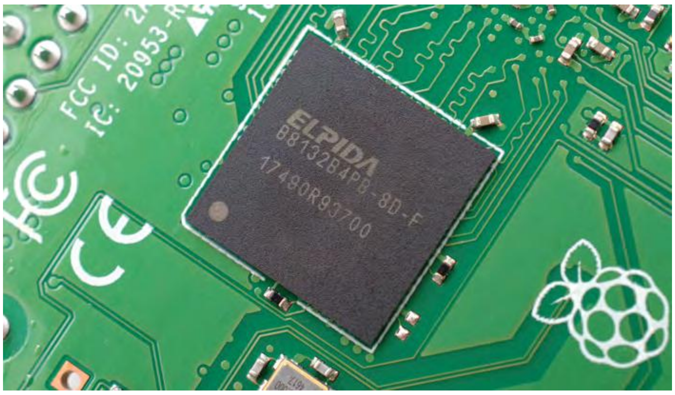


Figure III.4: random access memory (RAM)

III.4.3 Radio module:

Turning the board over again you'll find another metal lid to the upper-right, this one featuring an etched Raspberry Pi logo . This covers the radio, the component which gives the Raspberry Pi the ability to communicate with devices wirelessly.

The radio itself acts as two main components, in fact: a WiFi radio, for connecting to computer networks; and a Bluetooth radio, for connecting to peripherals like mice and for sending data to or receiving data from nearby smart devices like sensors or smart phones [9].

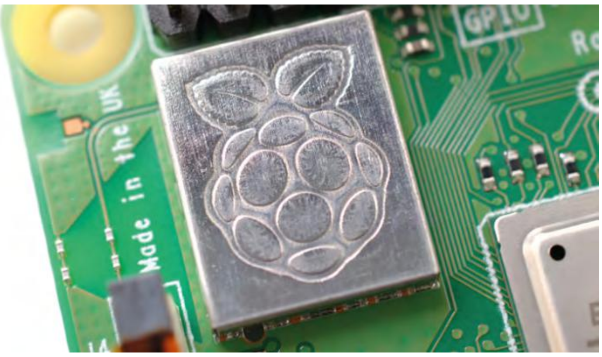


Figure III.5: radio module

III.4.4 Power management integrated circuit (PMIC):

Another black, plastic-covered chip can be seen to the bottom edge of the board, just behind the middle set of USB ports. This is the network and USB controller, and is responsible for running the Ethernet port and the four USB ports. A final black chip, much smaller than the rest, can be found a little bit above the micro USB power connector to the upper-left of the board this is known as a power management integrated circuit (PMIC), and handles turning the power that comes in from the micro USB port into the power the Pi needs to run [9].

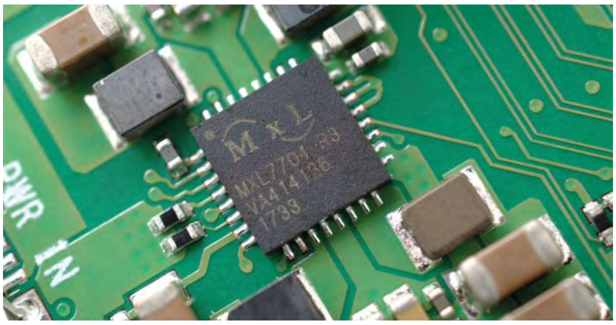


Figure III.6: power management integrated circuit (PMIC)

III.4.5 USB Ports:

The Raspberry Pi 3 has four USB ports, allowing you connect it to keyboards, mice, digital cameras, and USB sticks containing all your files [10]. These are known as USB 2.0 ports, which means they are based on version two of the Universal Serial Bus standard [9].

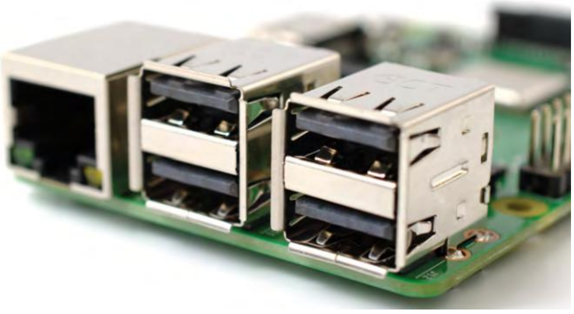


Figure III.7: USB ports

III.4.6 Ethernet Port:

The Ethernet port of the raspberry pi is the main gateway for communicating with additional devices. The raspberry pi Ethernet port is used to plug your home router to access the internet [11].

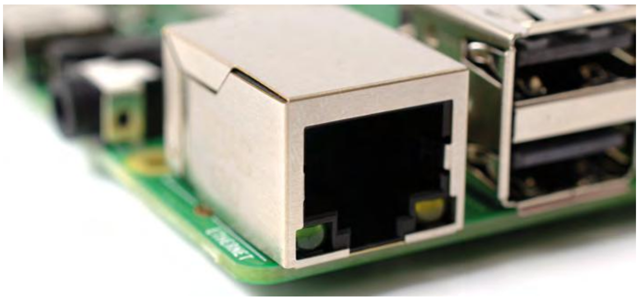


Figure III.8: Ethernet port

III.4.7 3.5 mm Audio-Visual (AV) jack:

Just above the Ethernet port, on the left-hand edge of the Raspberry Pi, is a 3.5 mm audio-visual (AV) jack. This is also known as the headphone jack, and it can be used for that exact purpose – though you’ll get better sound connecting it to amplified speakers rather than headphones. It has a hidden, extra feature, though: as well as audio, the 3.5 mm AV jack carries a video signal which can be connected to TVs, projectors, and other displays that support a composite video signal using a special cable known as a tip-ring-ring-sleeve (TRRS) adapter [9].

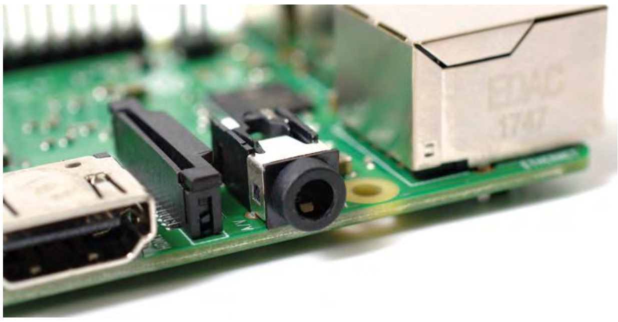


Figure III.9: 3.5 mm audio-visual (AV) jack

III.4.8 Camera connector:

Is a strange-looking connector with a plastic flap which can be pulled up; this is the camera connector, also known as the Camera Serial Interface (CSI) . This allows you to use the specially designed Raspberry Pi Camera Module [9].

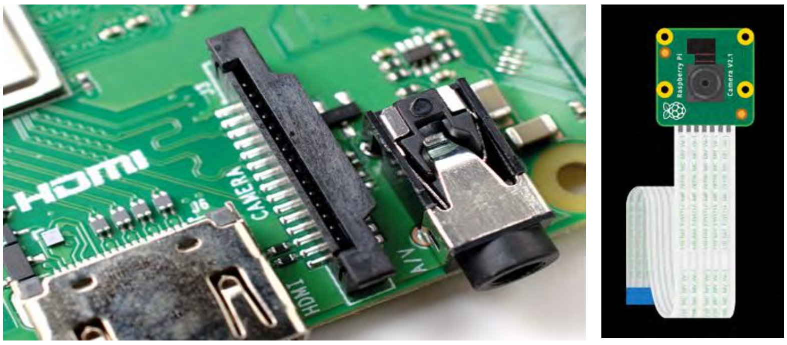


Figure III.10:Camera connector

III.4.9 HDMI Port:

This connector allows you to hook up a high-definition television or other compatible device using an HDMI cable [12].

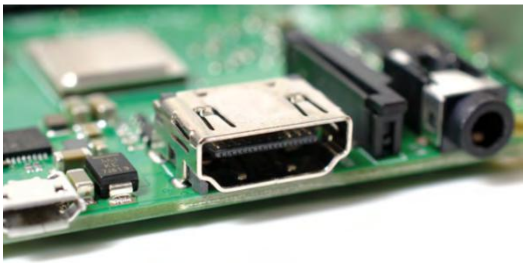


Figure III.11: HDMI Port

III.4.10 Micro USB Power Port:

The power source cable is a small switch, which is placed on side of the shield. The main purpose of the power source connector is to enable an external power source [11].

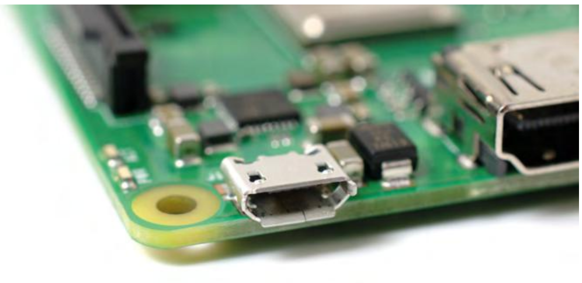


Figure III.12:Micro USB Power Port

III.4.11 Display Connector (DSI):

Or Display Serial Interface (DSI): The connection options of the raspberry pi board are two types such as HDMI and Composite.Many LCD and HD TV monitors can be attached using an HDMI male cable and with a low-cost adaptor [11].

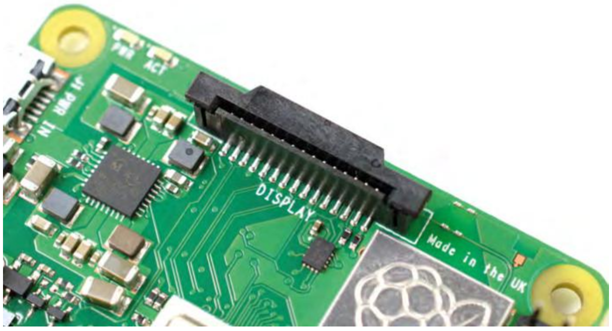


Figure III.13: Display Connector (DSI)

III.4.12 GPIO header:

The general purpose input and output pins are used in the raspberry pi to associate with the other electronic boards. These pins can accept input & output commands based on programming raspberry pi. The raspberry pi 3 model B affords 40 digital GPIO pins. These pins are used to connect other electronic components. For example, you can connect it to the temperature sensor to transmit digital data [11].

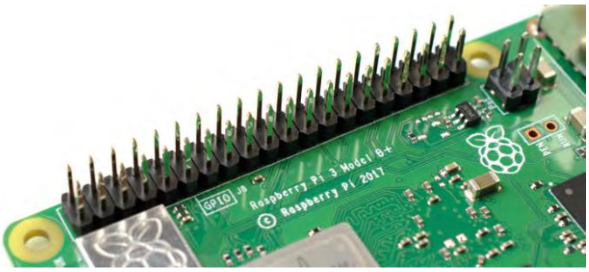


Figure III.14: GPIO header

III.4.13 Micro SD card connector:

A little SD card is used as the Raspberry Pi’s hard drive. This is where the operating system will live once you’ve put it on there. Most computers won’t be able to directly connect to a microSD card, but you can get an adaptor that plugs into normal SD card slots [10].

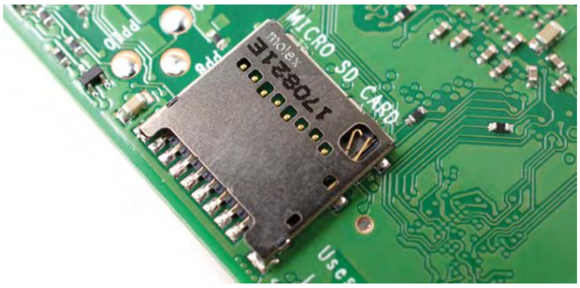


Figure III.15: Micro SD card connector

III.4.14 LEDs:

Light-emitting diodes, for all of your indicator light needs [12].

Figure III.16: Status LED

III.5 The different operating systems for the Raspberry Pi:

There are different operating systems for raspberry pi, among them:

III.5.1 Raspbian:

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware [13], it was released in 2012 by the Raspbian foundation [14].

An operating system is the set of basic programs and utilities that make your Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on your Raspberry Pi [13].

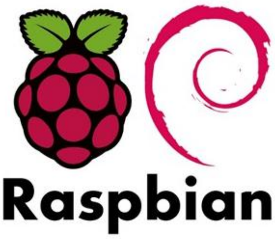


Figure III.17: Raspbian logo

III.5.2 Ubuntu Core:

Ubuntu Core is the version of Ubuntu designed for Internet of Things applications. Ubuntu is the most popular Linux-based Operating System in the world with over 20+ derivatives and given that it has an active and welcoming forum, it will be easy to get up and running with Ubuntu Snappy Core on your Raspberry Pi [15].

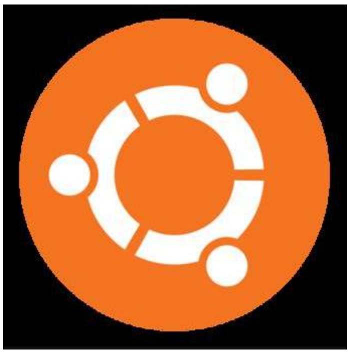


Figure III.18: Ubuntu Core logo

III.5.3 Kali Linux:

An open source, Debian-based Linux distribution used in security auditing and penetration testing.

Kali contains hundreds of tools to aid in various information security tasks [16].

Figure III.19: Kali Linux Logo

III.5.4 Pidora:

Is a Fedora Remix optimized for the Raspberry Pi, which in turn was based on the Linux kernel, developed by the community-supported Fedora Project and owned by Red Hat.

Pidora although somewhat similar to Raspbian is slightly different as it's distributed with a handful of different software. Some of the included software includes text editors, programming languages, and more [17].

Figure III.20: Pidora Logo

III.5.5 OpenELEC:

OpenELEC (short for "Open Embedded Linux Entertainment Center") is a free and open source, small and very fast booting, embedded operating system.

OpenELEC works on most computers you can install Kodi on, even the Raspberry Pi [18].

Figure III.21: Openelec Logo

III.5.6 Arch Linux:

First released in 2002, Arch Linux is a Linux distribution for i686 and x86-64 computer architectures. Its design focuses on simplicity, security, and efficiency.

among his distributions: Arch Linux ARM, which runs on ARM devices including the Raspberry Pi [19].

Figure III.22: Arch Linux Logo

III.6 Applications:

Raspberry Pi's applications are wonderfully diverse. In addition to fulfilling many standard tasks, the micro-computer has evolved to perform more unusual jobs. among these applications we find [20]:

- Web server (such as Apache, lighttpd and nginx) [20]
- Smart coffee machine pump [20]
- Smart Home Headquarters (control and management home heating, lamps, blinds, refrigerators, washing machines, etc.) [20]
- Raspberry Pi security system [20]
- Environmental sensing/monitoring (e.g. Weather Station) [21]
- Robotics (like web based controlled robot) [21]

III.7 Conclusion:

Raspberry Pi is considered a small form factor device has unlimited possibilities of project development in area of embedded systems.

This chapter contains generalities about the raspberry pi, where we provided the different models of raspberry pi and we got to know the difference between them such as: RAM, CPU, soc type,...,etc; also we presented and explained each component separately, finally we mentioned some of the operating systems of raspberry pi and its applications.

Chapter IV

Programming Languages

IV.1 Introduction:

A programming language is a set of commands, instructions, and other syntax use to create a software program.

Nowadays, there are different programming languages like JavaScript, PHP, C, C++, Python,..., etc; this programming languages used to control the performance of the computer or machine.

In this chapter; we will see the programming languages that i will use in my project. The HTML language use for prepare and describing the structure of web page; the PHP and JavaScript are a programming languages for building dynamic and interactive web site.In the end for control the style of web document we need the CSS.

IV.2 HTML (Hyper Text Markup Language):

HTML is the Standard Markup Language. It is used for developing Web Pages. HTML is Hyper Text Markup Language and is used for describing the structure of web pages. Various Tags are used in HTML like "heading", "paragraph", "table", and so on [1].

IV.2.1 HTML Document Structure:

A basic HTML page is a document that typically has the file extension **.html**. All HTML documents follow the same basic structure so that the browser that renders the file knows what to do. The basic structure on which all web pages are built looks like this [2]:

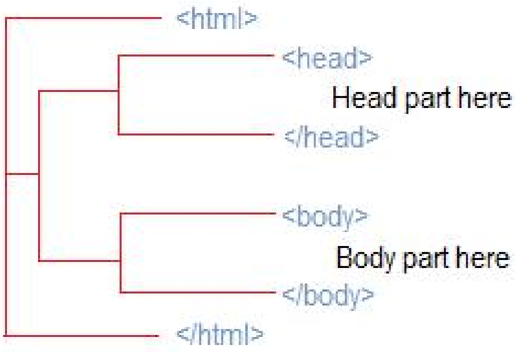


Figure IV.1: HTML Structure

IV.2.2 HTML Root Element:

The HTML root element contains: one <head> element and one <body> element.

- Head Element: The HEAD element contains information about the current document, such as its title, keywords that may be useful to search engines, and other data that is not considered document content [3].

All of the HTML elements that can be used inside the <head> element are: **base - link - noscript - script - style - title** [2].

- Body Element: There can only be one <body> element in an HTML document because this element is the container that holds the content of the document. All of the content that you see rendered in the browser is contained within this element [2].

IV.2.3 HTML Tags:

HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their corresponding closing tags. For example, <html> has its closing tag</html> and <body> tag has its closing tag </body> tag etc [4].

Here's a table of some HTML tags :

Start Tag	Content	End Tag
<html>	Creates an HTML document	</html>
<head>	Sets off the title & other info that isn't displayed	</head>
<body>	Sets off the visible portion of the document	</body>
<title>	Puts name of the document in the title bar; when bookmarking pages, this is what is bookmarked	</title>
<p>	Creates a new paragraph	</p>
<br>	AInserts a line break (carriage return)	/
<div>	Used to format block content with CSS	</div>
<h1> to <h6>	Creates headlines -- H1=largest, H6=smallest	</h1> to </h6>
<b>	Creates bold text (should use instead)	</b>
<i>	Creates italicized text (should use instead)	</i>

<ul>	Creates an unordered list	</ul>
<li>	Encompasses each list item	</li>
<hr>	Inserts a horizontal rule	/
<a href ="URL"> clickable text	Creates a hyperlink to a Uniform Resource Locator	</a>
<table>	Creates a table	</table>
<tr>	Sets off each row in a table	</tr>
<td>	Sets off each cell in a row	</td>
<th>	Sets off the table header (a normal cell with bold, centered text)	</th>

Table IV.1: HTML tags list [5]

IV.3 CSS (Cascading Style Sheets):

CSS is used to control the style of a web document in a simple and easy way.They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML [6][7].

IV.3.1 External Stylesheet:

An external CSS stylesheet can be applied to any number of HTML documents by placing a <link> element in each HTML document.

The attribute rel of the <link> tag has to be set to "stylesheet" , and the href attribute to the relative or absolute path to the stylesheet.

It is recommended that the <link> tag be placed in the HTML file's <head> tag so that the styles are loaded before the elements they style.

Link: **<link rel="stylesheet" type="text/css" href="foldername/style.css">** [8].

IV.3.2 Internal Styles:

CSS enclosed in <style></style> tags within an HTML document functions like an external stylesheet, except that it lives in the HTML document it styles instead of in a separate file, and therefore can only be applied to the document in which it lives. Note that this element must be inside the <head> element for HTML validation (though it
Will work in all current browsers if placed in body) [8].

Example:

```
<html>
    <head>
        <style>
              h1 {
                  color: green; text-decoration: underline;
                  }
              p {
                  font-size: 25px;
                 }
        </style>
    </head>
    <body>
        <h1>Hello world!</h1>
        <p> CSS: Cascading Style Sheets</p>
    </body>
 </html>
```

IV.3.3 CSS Basic Properties:

Property	Description
background	A shorthand property for all the background-* properties
background-color	Specifies the background color of an element
background-image	Specifies one or more background images for an element

border	A shorthand property for border-width, border-style and border-color
border-bottom	A shorthand property for border-bottom-width, border-bottom-style and border-bottom-color
border-color	Sets the color of the four borders
font	A shorthand property for the font-style, font-variant, font-weight, font-size/line-height, and the font-family properties
font-size	Specifies the font size of text
font-family	Specifies the font family for text
margin	Sets all the margin properties in one declaration
padding	A shorthand property for all the padding-* properties
text-align	Aligns the text in an element (left, right, center, justify)

Table IV.2 :CSS Basic Properties[9]

IV.4 PHP (Hypertext Pre-processor):

PHP is a programming language for building dynamic, interactive Web sites. As a general rule, PHP programs run on a Web server, and serve Web pages to visitors on request. One of the key features of PHP is that you can embed PHP code within HTML Web pages, making it very easy for you to create dynamic content quickly.[10]

FigureIV.2 : PHP logo

IV.4.1 PHP environment setup:

In order to develop and run PHP Web pages, three vital components need to be installed on your computer system:

Web Server: PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server.

Database : PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.

PHP Parser: In order to process PHP script instructions, a parser must be installed to generate HTML output that can be sent to the Web Browser [11].

IV.4.2 PHP Tags:

The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP.' There are four ways to do this [11]:

tags	Description
Standard: **<? php...?>**	-The most universally effective PHP tag style is: <? php...?> . -These tags are the standard method to embed PHP code in a file.
Short-open (SGML-style): **<?...?>**	we can disable or enable these tags with the option short_open_tag . Short tags: -are disallowed in all major PHP coding standards. -are discouraged in the official documentation.

	-are disabled by default in most distributions. -interfere with inline XML's processing instructions. -are not accepted in code submissions by most open source projects.
ASP-style: **<%...%>**	-ASP-style tags mimic the tags used by Active Server Pages to delineate code blocks. -To use ASP-style tags, we will need to set the configuration option in php.ini file.
HTML script: **<script language="PHP">...</script>**	HTML script tags look like this: <script language="PHP">...</script>

Table IV.3:PHP Tags.[11] [12]

IV.5 JavaScript:

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles [13].

IV.5.1 JavaScript in an HTML Page:

To include JavaScript inside a page, you need to wrap it in <script> tags [14]:

```
<script type="text/javascript">
//JS code goes here
</script>
```

We can place the <script> tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the <head> tags [15].

IV.5.2 Call an External JavaScript File:

We can also place JavaScript in its own file and name it inside your HTML. That way, we can keep different types of code separate from one another, making for better-organized files such as [14]:

```
<script src="file.js"></script><code></code>
```

IV.5.3 Data types in JavaScript:

The six data types in JavaScript are as follows [16]:

Data type	Description
Numbers	represents numeric values e.g. 100
Strings	represents sequence of characters e.g. "hello"
Booleans	represents boolean value either false or true
Null	represents null i.e. no value at all
Undefined	represents undefined value
Objects	represents instance through which we can access members

Table IV.4:data types in JavaScript [16][17]

IV.5.4 JavaScript operators:

JavaScript supports the following types of operators:

operator	Description	example

Arithmetic Operators	Arithmetic operators are the basic operators that we use to do sums in JavaScript like : addition (+), multiplication (*), subtraction (-),...	A+B=20 (Addition)
		A=13
		B=7

Comparison Operators	Comparison operators are used in logical statements	A == B (equal) A < B (less than)

	to determine equality or difference between variables or values like: Equal, Not Equal, Less than,...	A != B (not equal)
Logical (or Relational) Operators	Logical operators perform logical operations: AND, OR and NOT.	-&& (Logical AND) If both the operands are non-zero, then the condition becomes true. Ex: (A && B) is true. -|| (Logical OR) If any of the two operands are non-zero, then the condition becomes true. Ex: (A || B) is true. -! (Logical NOT) Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false. Ex: ! (A && B) is false.
Assignment Operators	An assignment operator assigns a value to its left operand based on the value of its right operand.	= (Simple Assignment) Assigns values from the right side operand to the left side operand. += (Add and Assignment) C += A is equivalent to C = C + A.

Conditional (or ternary) Operators	The conditional operator is used as a shortcut for standard if statement. It takes three operands. **Syntax:** Condition ? expr1: expr2	If Condition is true? Then value X : Otherwise value Y

Table IV.5:JavaScript operator [18]

IV.5.5 JavaScript Function:

JavaScript functions are defined with the function keyword. To create a function we can use a function declaration like this [19]:

```
function functionName(parameters) {
  // code to be executed
}
```

IV.6 Conclusion:

This chapter contains programming languages and their basics that we need to build a web page with easy and simple way.

We saw the HTML language which is a simple language to build the structure of web page and to control its style we need CSS, and for building a dynamic and interactive web site we need PHP and JavaScript.

Chapter V

Building A Robot And Control It

V.1 Introduction:

Future is all about robots, robot can perform tasks where humans cannot. Generally robot is a mix of electronic, electrical and mechanical engineering and can do the tasks automatically on its own or under the supervision of humans.

In this chapter, we will program our Raspberry Pi for the use in a robotic environment. We will present the components of the robot that we need, after that we will see a simple synoptic diagram to clarify our work, this diagram contains two important parts: the first one is hardware: we will present the components of robot and how to make a robot, and the second is the software: preparation of the raspberry pi and create a web page.

V.2 objectives:

- Learn how to make and build a mobile robot.
- Learn some programming languages.
- Know the raspberry pi and how to prepare it.
- How can access to the raspberry pi from anywhere.
- Controlling a robot from anywhere via a web page.

V.3 Components:

- Raspberry Pi 3 Model B- Motor DC (2)- Motor driver (L298H-bridge Module) -Jumper Wires- USB Webcam- Battery 9V- Micro SD card (16 GB).

V.4 Synoptic diagram:

In order to clarify our work, we have made a synoptic diagram sowing the different components that we needed to control the robot, it contains two important parts: the first one is hardware (components of robot), and the second is the software (Raspberry pi and web page programming).

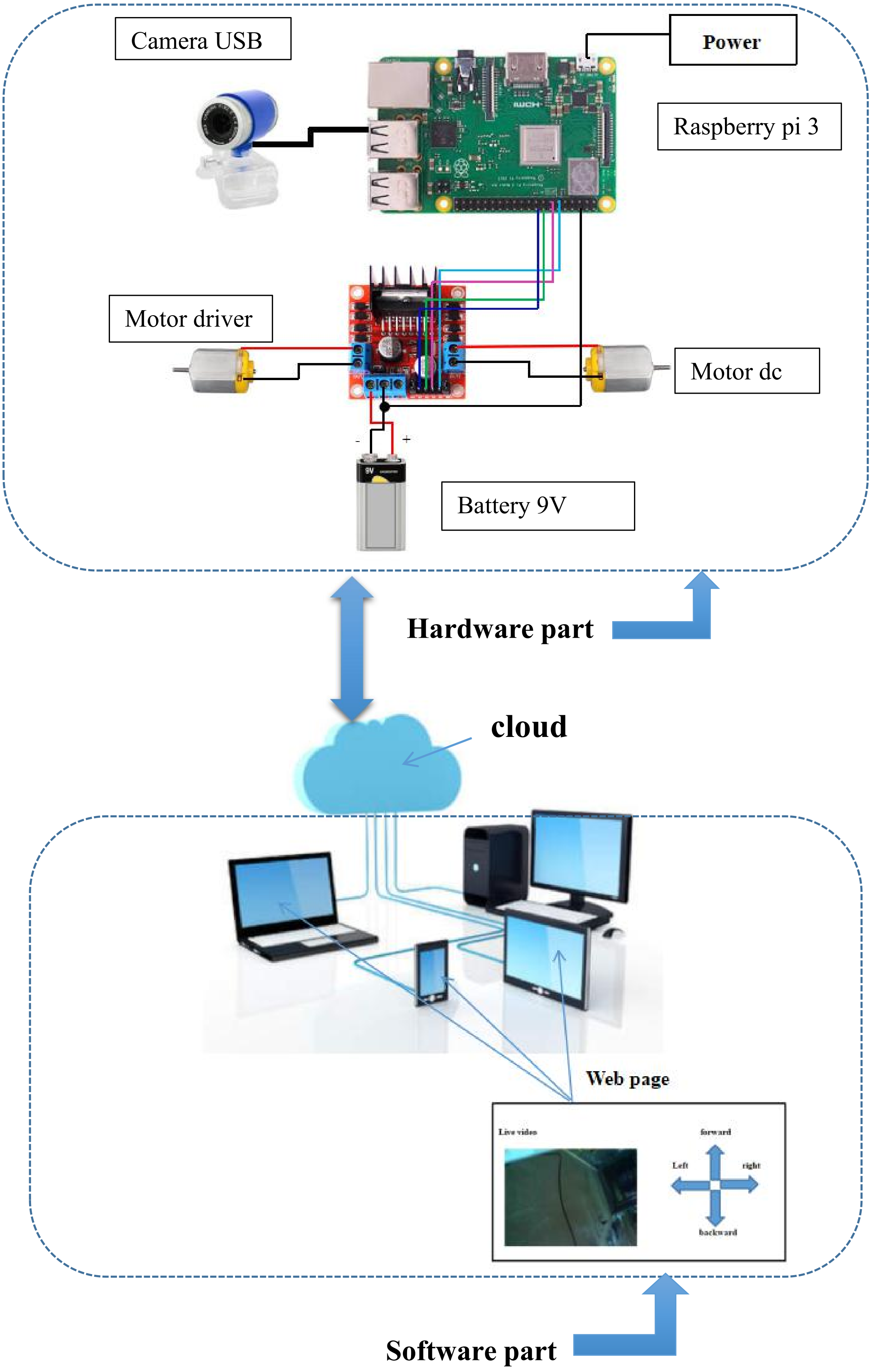


Figure V.1: Synoptic diagram.

V.4.1 Hardware part:

V.4.1.1 Robot Components:

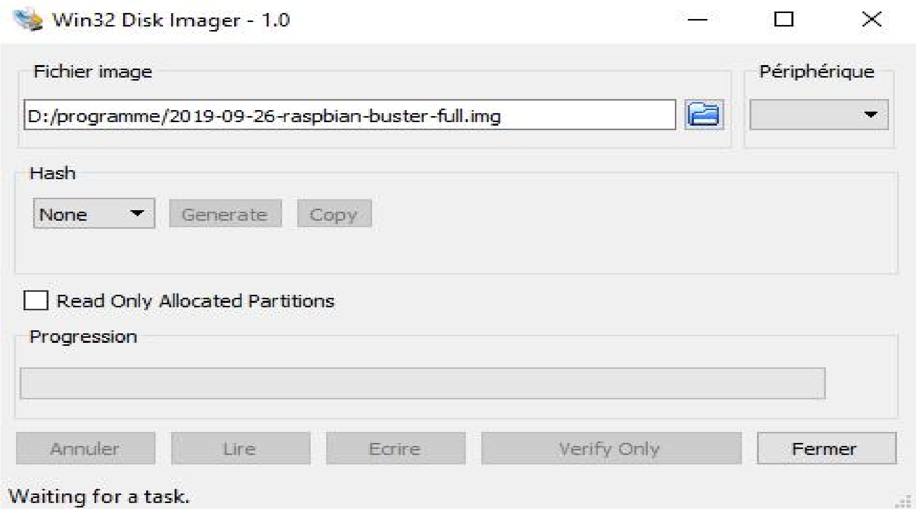
- **The Raspberry Pi:**

The Raspberry Pi is a small computer, inexpensive, and often sufficient for many uses, it also helps to encourage learning to program. That's why I have chosen it to perform and accomplish my project.

❖ **Preparing the Raspberry Pi:**

Step 01: Installation of the "Raspbian" OS

To install the operating system in raspberry pi, we must download Raspbian from the official site: **raspberry pi.org/downloads.** Then we install it on the memory card (16 GB) using image writing software: "Win32 Disk Imager".



Win32 Disk Imager - 1.0

Fichier image: D:/programme/2019-09-26-raspbian-buster-full.img

Hash: None

□ Read Only Allocated Partitions

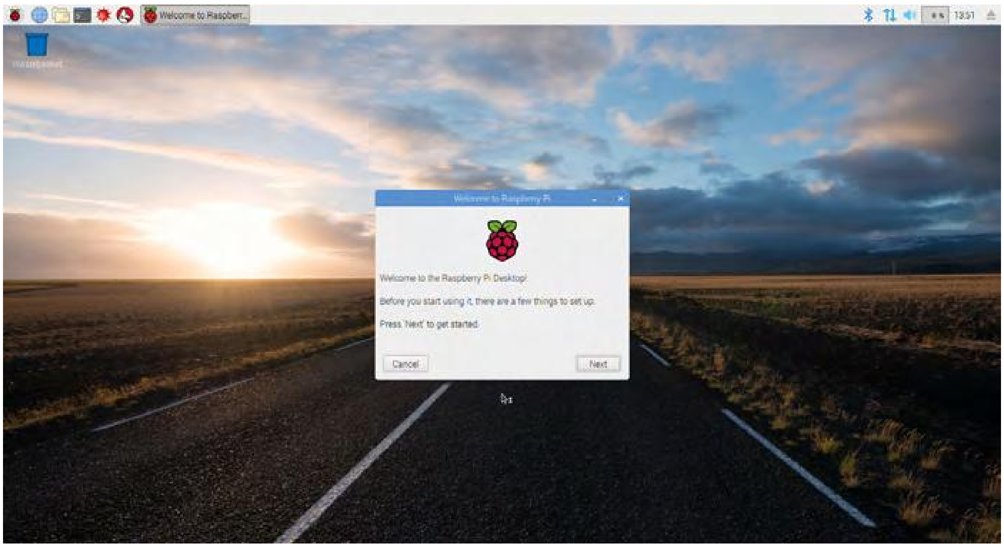
Waiting for a task.

Figure V.2: win32 Disk Imager

Step 02: System start

One should follow these steps to start the Raspberry Pi for the first time:

- Insert the micro SD card into Raspberry Pi’s storage.
- Connect a keyboard and mouse.
- Connect the HDMI output to TV or monitor.
- Connect a power supply (is the very last step in the hardware setup process).
- Finally, we connect the power supply to a mains socket and switch the socket on; the Raspberry Pi will immediately start running.



Welcome to the Raspberry Pi Desktop!

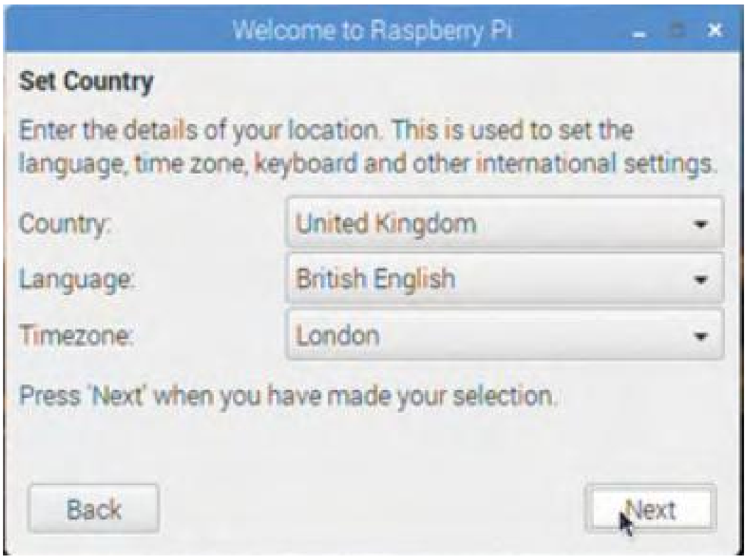
Before you start using it, there are a few things to set up.

Press 'Next' to get started.

Figure V.3: The Raspbian desktop

Step 03: Configuration of Raspberry Pi

- ➢ Select a language: choose the country, language, and time zone, and click Next.



Set Country

Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.

Country: United Kingdom

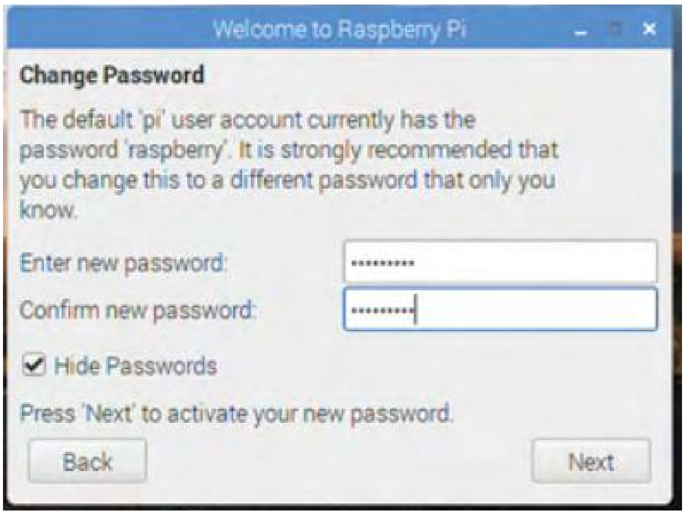
Language: British English

Timezone: London

Press 'Next' when you have made your selection.

Figure V.4: Selecting a language, country, and time zone.

- ➢ Setting a new password: password for the 'pi' user which is the standard user name for all Raspberry Pi computers running Raspbian is 'raspberry', we need to change this password for more security.



Welcome to Raspberry Pi

Change Password

The default 'pi' user account currently has the password 'raspberry'. It is strongly recommended that you change this to a different password that only you know.

Enter new password: ••••••••••

Confirm new password: ••••••••••

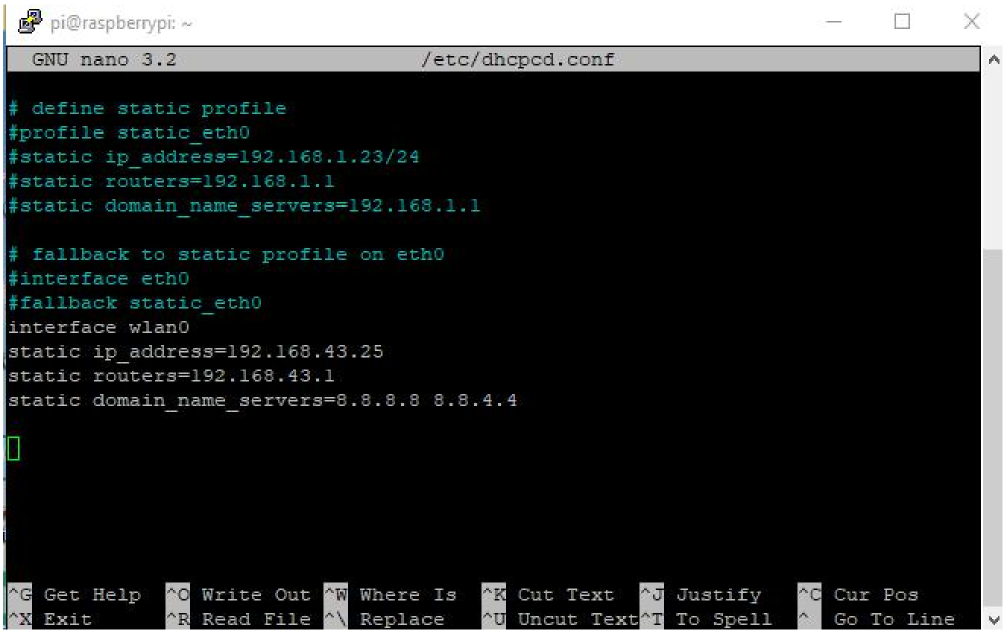
☑ Hide Passwords

Press 'Next' to activate your new password.

Figure V.5: Setting a new password

Step 04: Setting static IP for Raspberry Pi

sudo nano /etc/dhcpcd.conf



pi@raspberrypi: ~

```
  GNU nano 3.2                    /etc/dhcpcd.conf

# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0
interface wlan0
static ip_address=192.168.43.25
static routers=192.168.43.1
static domain_name_servers=8.8.8.8 8.8.4.4




^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text^T To Spell    ^_ Go To Line
```

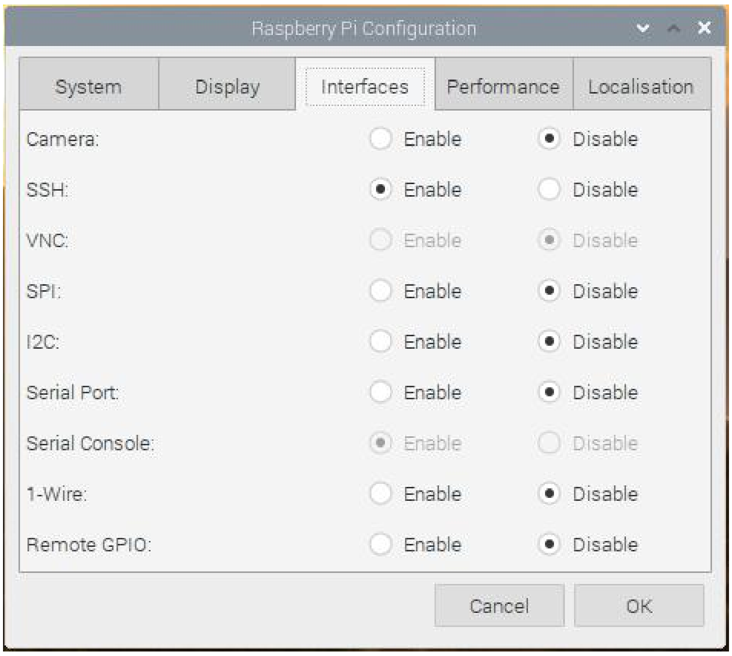
Figure V.6: Setting static IP

Step 05: SSH and remote desktop VNC

-Enabling SSH and Using PuTTY:

Secure Shell, sometimes referred to as Secure Socket Shell, is a protocol which allows you to connect securely to a remote computer or a server by using a text-based interface [1].

To be able to connect via our computer, we must activate "ssh" on the Raspberry Pi. To do this, we click preferences then chose raspberry pi configuration after that enable ssh.

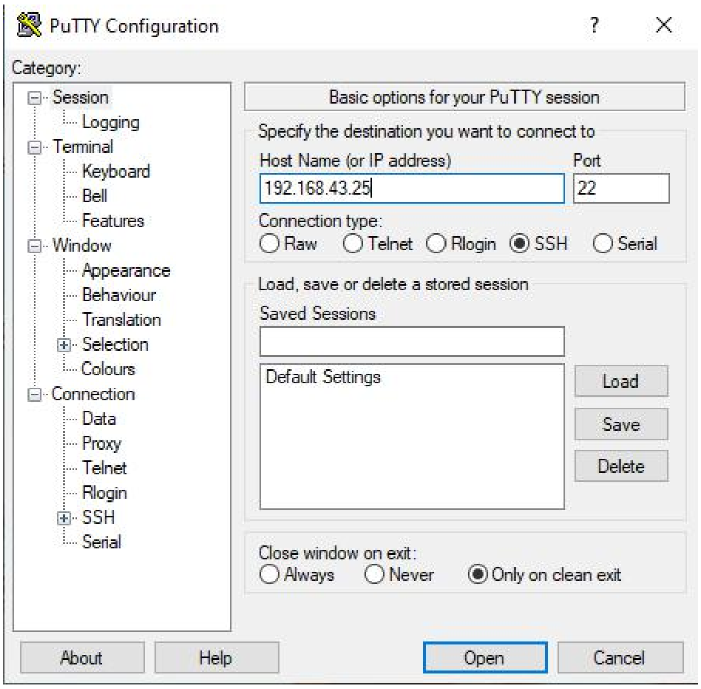


Raspberry Pi Configuration

Camera:	○ Enable	● Disable
SSH:	● Enable	○ Disable
VNC:	○ Enable	● Disable
SPI:	○ Enable	● Disable
I2C:	○ Enable	● Disable
Serial Port:	○ Enable	● Disable
Serial Console:	● Enable	○ Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

Figure V.7: Enable SSH

-After validation, ssh is activated, we install PuTTY to use it on the computer. The settings are made as following (SSH connection, port 22):



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.43.25

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure V.8: PuTTY configuration

-**VNC** (Virtual Network Computing) is a graphical desktop sharing system that allows you to remotely control the desktop interface of one computer from another computer or mobile device. VNC Viewer transmits the keyboard and either mouse or touch events to VNC Server, and receives updates to the screen in return [2].

-Enabling VNC Server:

On your Raspberry Pi, boot into the graphical desktop.

- Select **Menu** > **Preferences** > **Raspberry Pi Configuration** > **Interfaces**.
- Ensure **VNC** is **Enabled**.

Installing VNC:

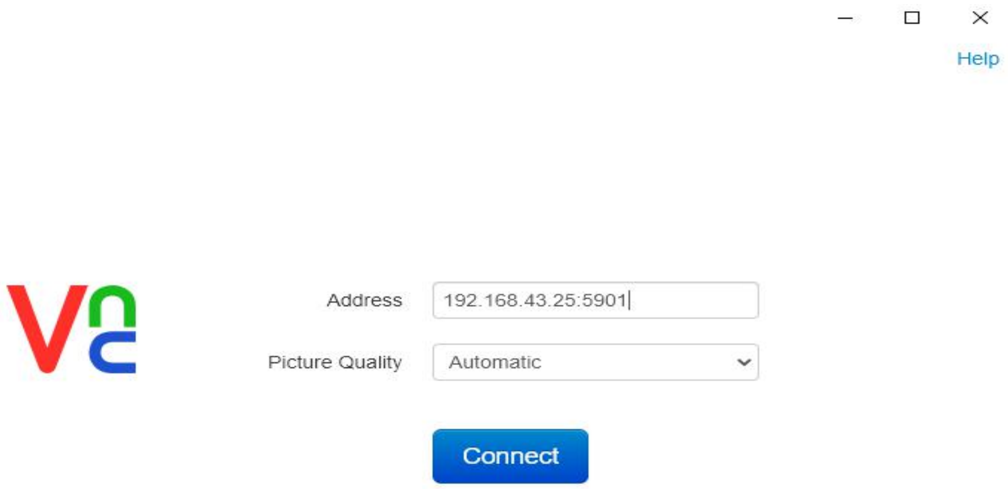
On the Raspberry Pi (terminal):

- sudo apt-get install tightvncserver
- vncserver

On the Windows:

we use Google chrome to install VNC viewer:

- Enter the IP address into VNC Viewer and port (5901):

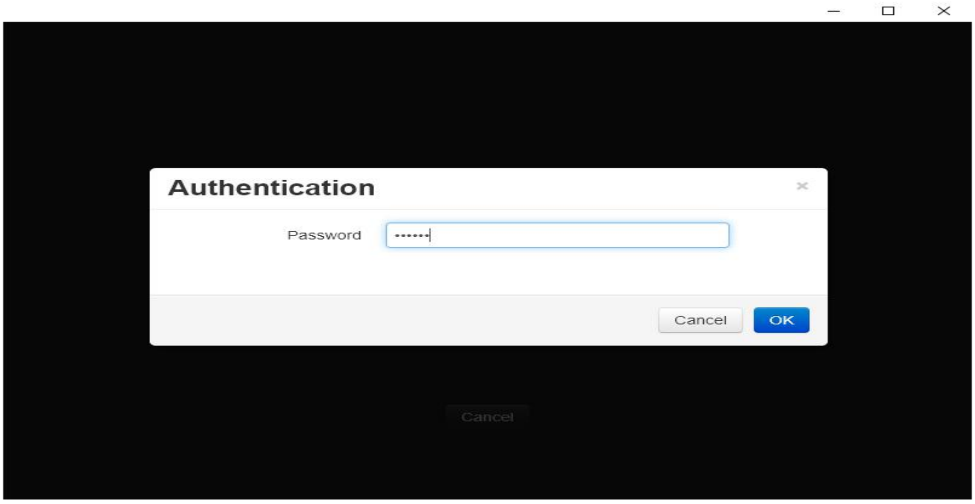


Address: 192.168.43.25:5901

Picture Quality: Automatic

Figure V.9: VNC Viewer

- Enter the password:



Authentication

Password ••••••

figure V.10: VNC authentication

-Now we can see the desktop of the Raspberry Pi inside a window on our computer.

- **Motor driver:**

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes [3].

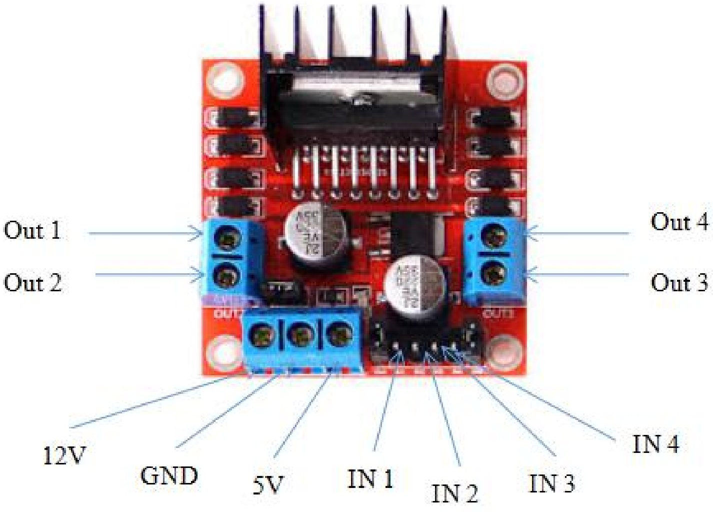


Figure V.11: motor driver L289 N

- **Motor DC:**

A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current, and convert this energy into mechanical rotation [4].

In my project I have used two DC motors to provide the motion for the robot.

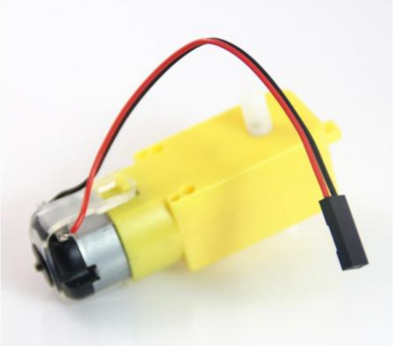


Figure V.12: DC Motor.

- **USB Webcam :**

A USB webcam is a camera that connects to a computer, usually through plugging it in to a USB port on the machine. The video is fed to the computer where a software application lets you view the pictures and also transfer them to the Internet [5].

Figure V.13: USB Webcam.

- **battery:**

To power the motor driver, we used a 9V battery.

Figure V.14: 9V Battery.

- **Jumper Wires:**

To connect the different electronic parts we used jumper wires male/female and female/female.

Figure V.15: Jumper Wires female/female and male/female.

V.4.1.2 Preparing the robot:

A robot is an electromechanical device that is able to interact in one way or another with its environment, making independent decisions or actions in order to achieve a specific task.

The robot consists of the following elements as show in the figure V.11:

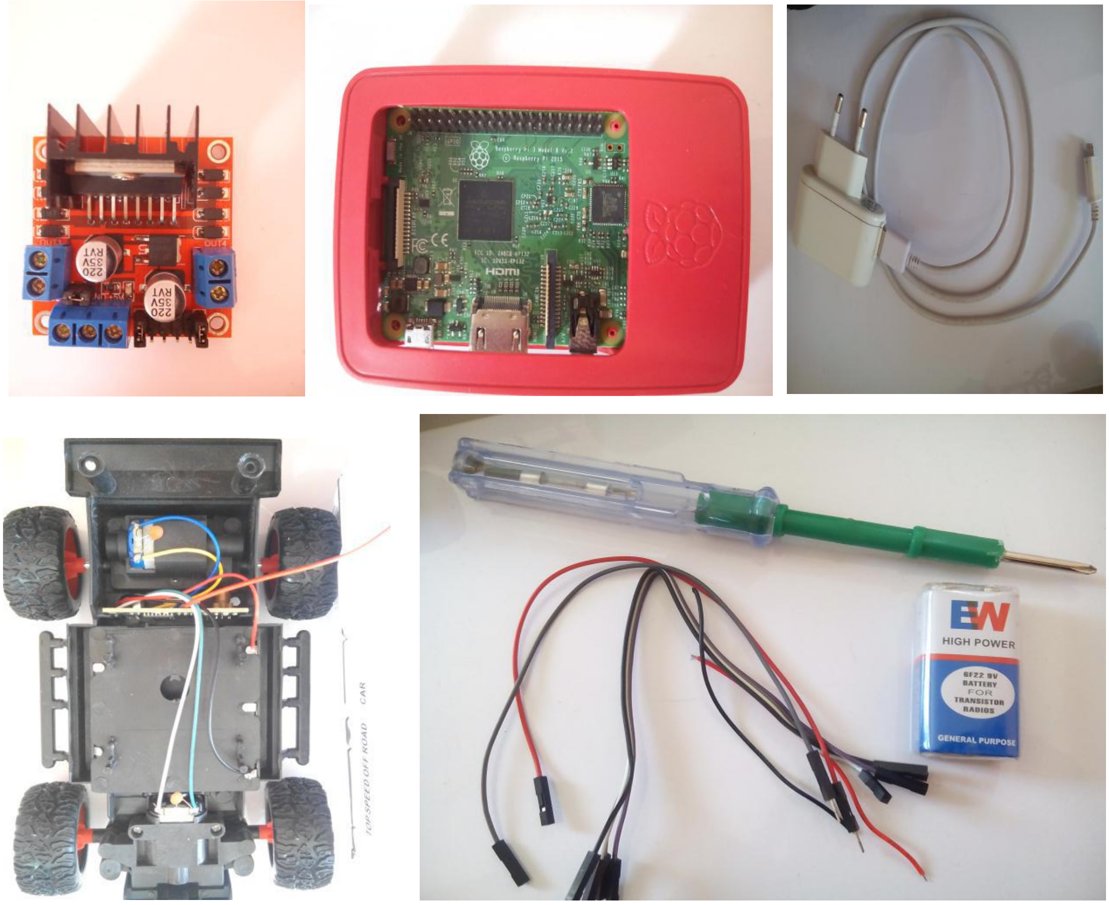


Figure V.16: Robot components.

-Connection of electronic parts:

Step 01: Connecting DC motors with L298 H-bridge:

In order to connect the motors, we follow this table:

L298 H-bridge	**Motor wires**
Out 1	The left black wires are connected to (-)
Out 2	The left red wires are connected to (+)
Out 3	The right Black wires are connected to (-)
Out 4	The right red wires are connected to (+)

Table V.1: Connecting motors.

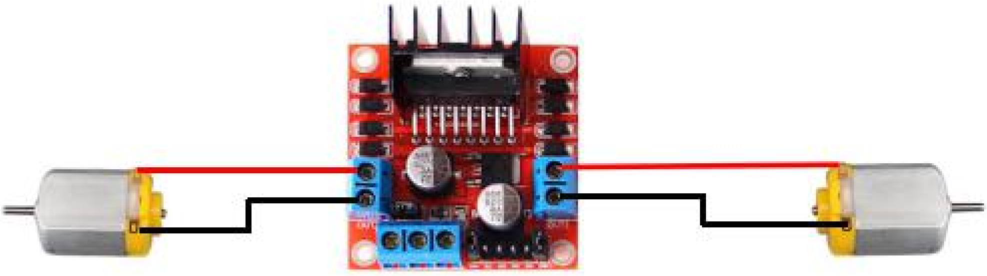


Figure V.17: Circuit Diagram (2 DC Motors with L298H-bridge).

Step 02: connect the power to the L298H-bridge

We follow this table:

L298H-bridge	**Battery**
12 V	Red Wire for battery (+) with 12 V
GND	Black wire for battery (-) with GND

Table V.2: Connect the power with L298H-bridge.

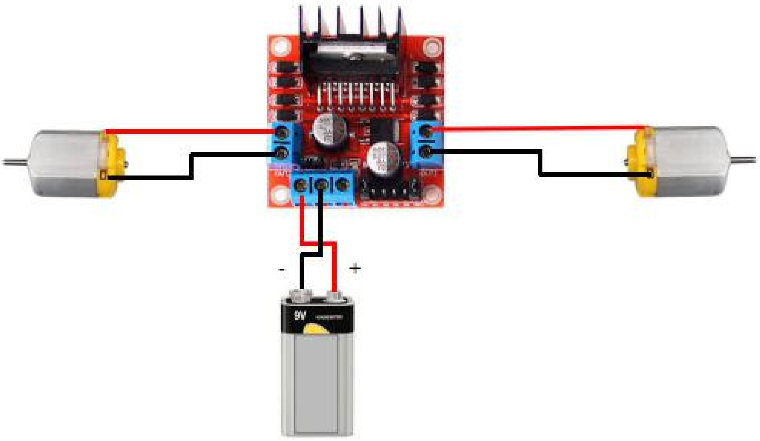


Figure V.18: Circuit diagram (connection between L298H-bridge and battery).

Step 03: connect the Raspberry Pi with L298H-bridge

L298H-bridge	**Raspberry Pi**
IN 1	GPIO 0
IN 2	GPIO 7
IN 3	GPIO 3
IN 4	GPIO 2
GND	GND

Table V.3: Connect the Raspberry Pi with L298H-bridge.

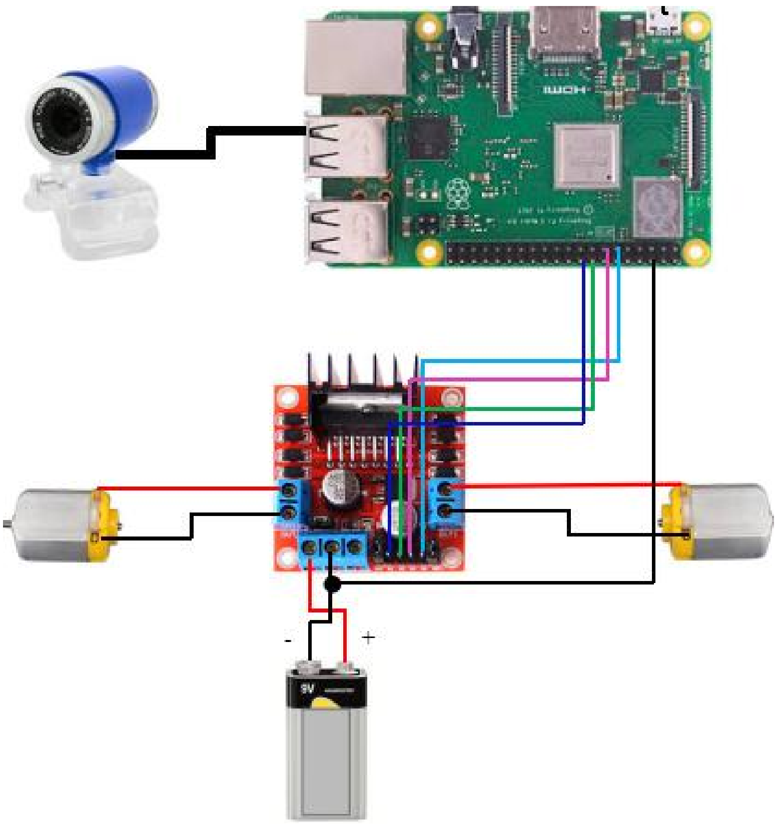


Figure V.19: Circuit diagram (all elements).

- **The final form of the robot:**

Finally, after assembling all the pieces I obtain the robot shown in figure V.20.

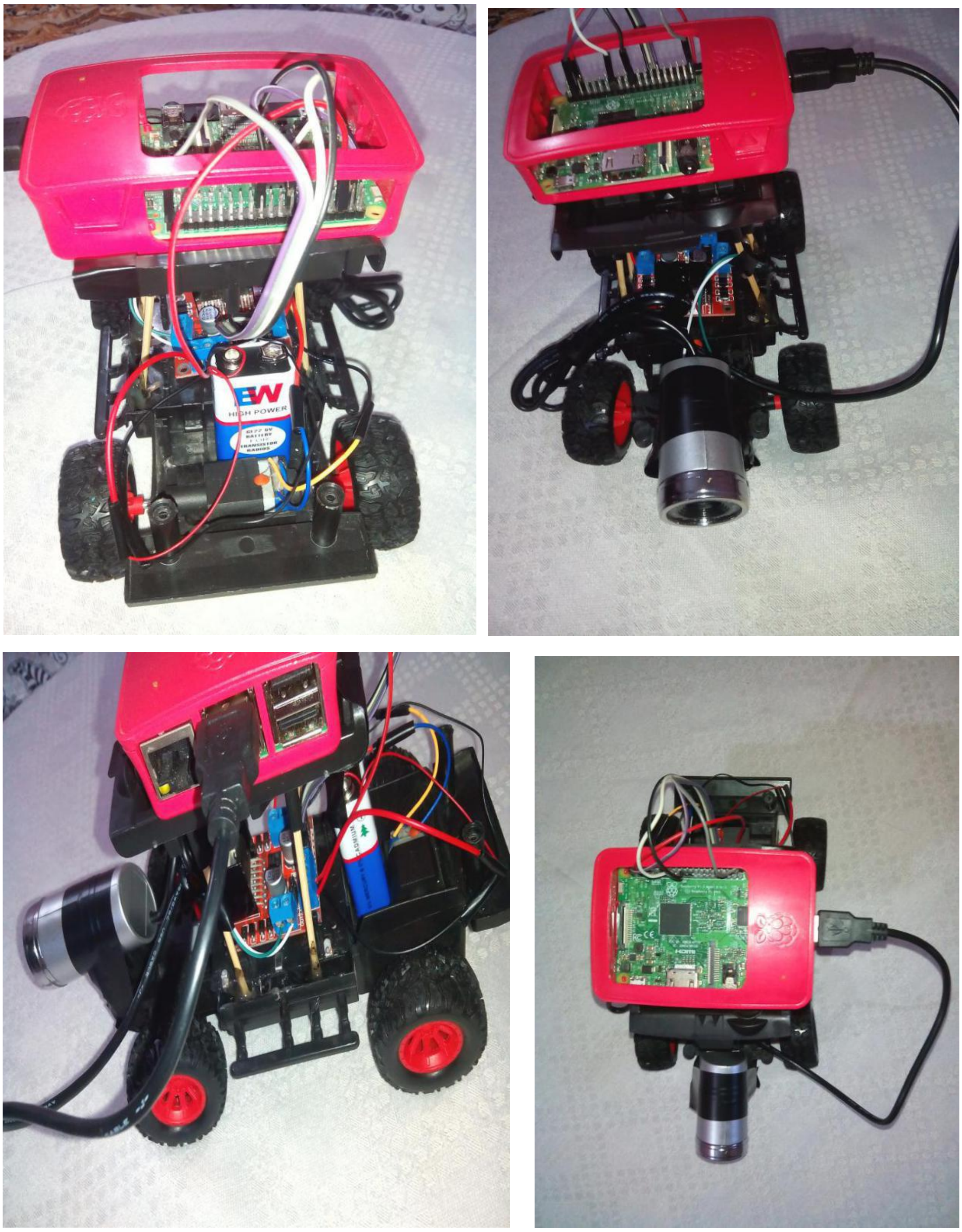


Figure V.20: The Mobile Robot.

V.4.2 Software part:

In this part we will create a web page on the net, as it will be compatible with all devices. We need to install the web server (Apache) on Raspberry Pi also we need to install MySQL database, HTTP and PHP server.

Step 01: Installing the Wiring Pi Library

WiringPi is a library written in C used to access GPIO pins on Raspberry Pi for BCM2835 (Broadcom Processor) SoC (System on Chip) [6].

On the terminal:

- sudo apt-get install git-core
- cd Desktop
- git clone git://git.drogon.net/wiringpi
- cd wiringpi
- ./build

Now wiringpi library is installed.

Step 02: We Install web server (Apache)

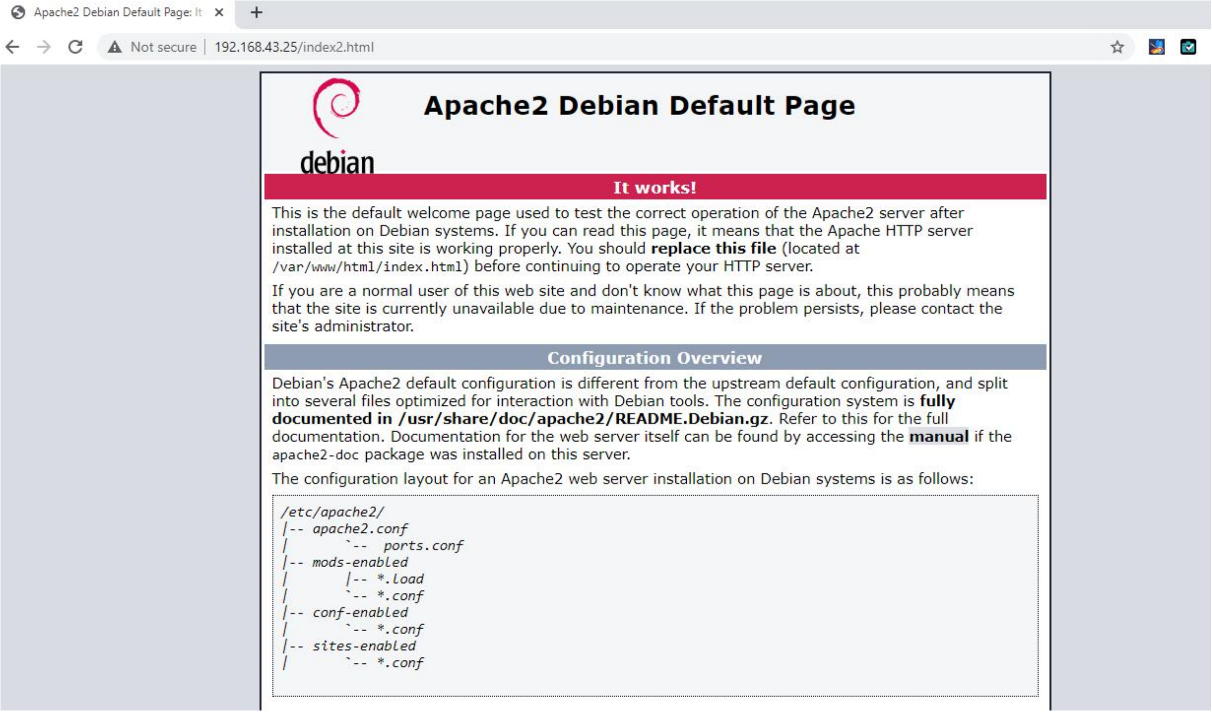
Apache is an open-source and free web server software that powers around 40% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation [7].

To install Apache on the Raspberry Pi, we must apply these commands:

- sudo apt-get update
- sudo apt-get install apache2 -y

To allow Apache to process PHP files, you need to install the PHP5 for Apache.

- sudo apt-get install libapache2-mod-php -y


Apache2 Debian Default Page: It
Not secure | 192.168.43.25/index2.html
debian

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Figure V.21: Apache

Step 03: Install MySQL database

MySQL is an open source software, it is a database management system that allows you to manage relational databases.

MySQL can run on various platforms UNIX, Linux, Windows, etc. we also can install it on a server or even in a desktop [8].

- sudo apt-get install mysql-server
- sudo mysql_secure_installation (new password)
- sudo apt-get install phpmyadmin

Installation of PHPMyAdmin:

PHPMyAdmin allows us to quickly and legibly see the content of our database, or to manipulate it without having to do our MySQL queries ourselves.

- sudo apt-get install phpmyadmin

will ask us several questions concerning its settings:

- ➢ we choose an Apache server
- ➢ Password then confirmation

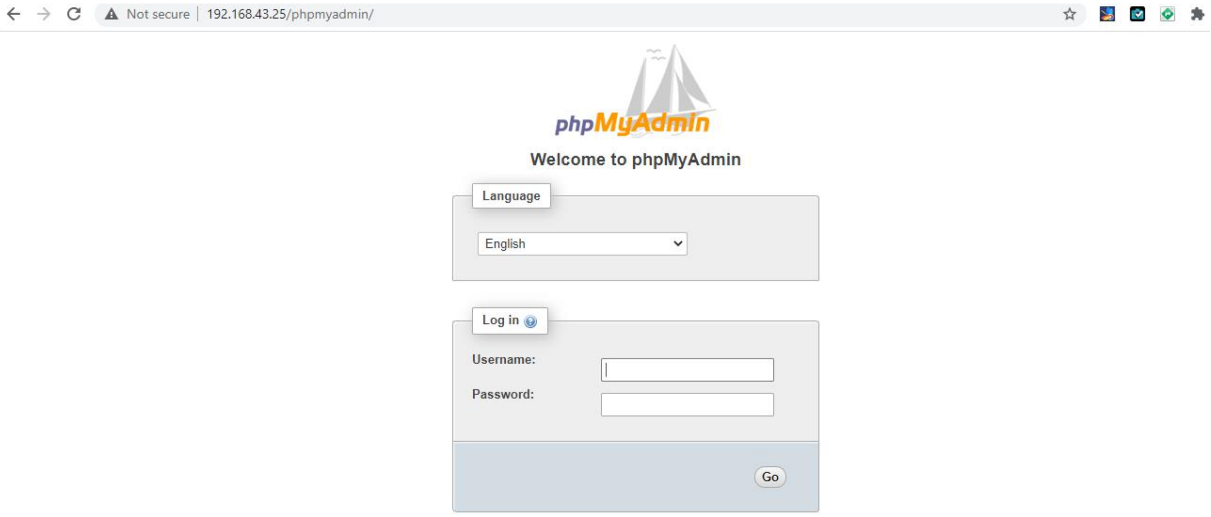
We will also need to modify the Apache configuration file to include our PHPMyAdmin installation.

- ➢ sudo nano /etc/apache2/apache2.conf

- Now at the bottom of this file we enter the following line:

Include /etc/phpmyadmin/apache.conf

- we save and quit by pressing CTRL + X then Y
- sudo /etc/init.d/apache2 restart.



Not secure | 192.168.43.25/phpmyadmin/

Welcome to phpMyAdmin

Language: English

Figure V.22: PHP MyAdmin

Step 04: Live video streaming online

- sudo apt-get install motion

Motion setting:

- sudo nano /etc/motion/motion.conf (change some settings)

Search for the following lines, then change them to the following:

- sudo nano /etc/default/motion (Start_motion_daemon=yes (we change no to yes))
- sudo service motion start

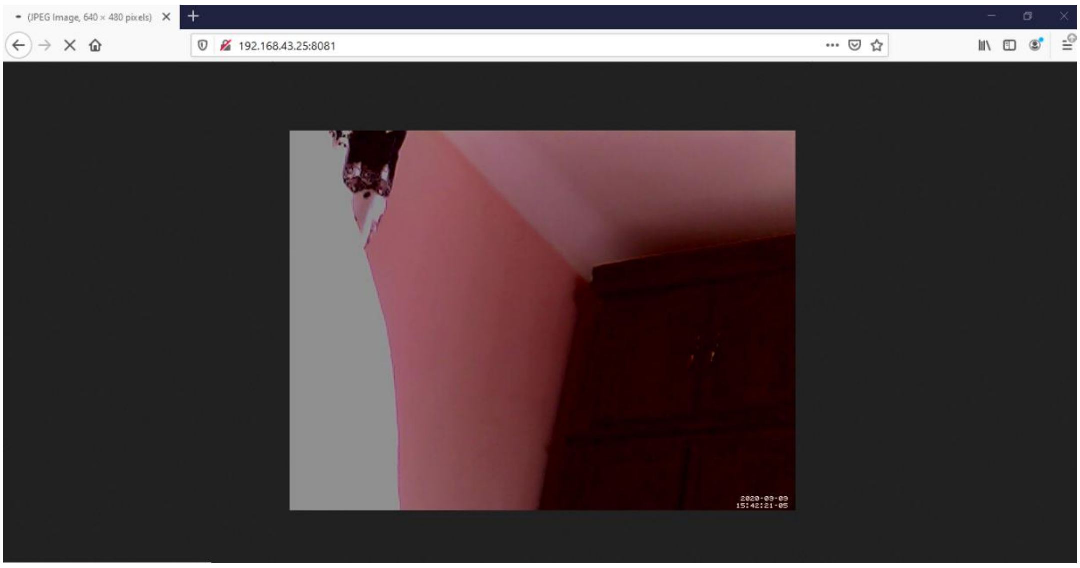


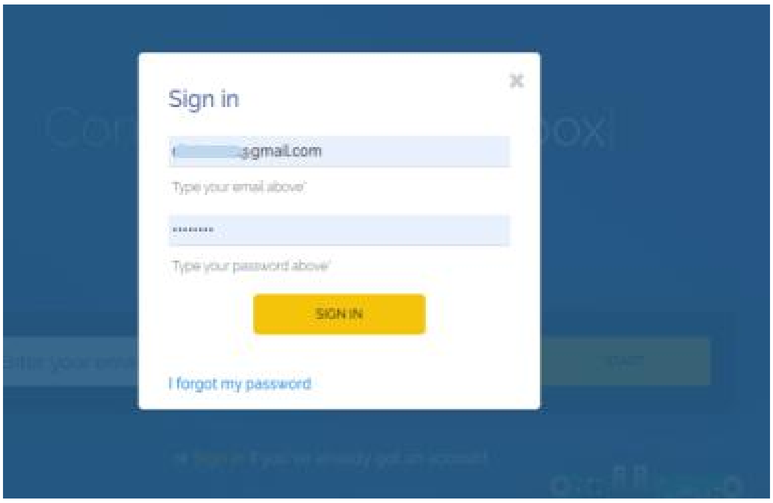
Figure V.23: live video

Step 05: Controlling a robot from anywhere

To control the robot via the internet from anywhere I choose DataLicity Software for controlling the raspberry pi.

Dataplicity is essentially a VPN for your Raspberry Pi. It not only lets you connect to your Pi remotely but you also can "wormhole" a web server through the system, allowing you to run a mini website from the comfort of your all-in-one computer.

Setup is as simple as entering your email address and pasting a command into your Pi's terminal. The command essentially creates a new user and enables the VPN tunnel [9].



Sign in

[redacted]gmail.com

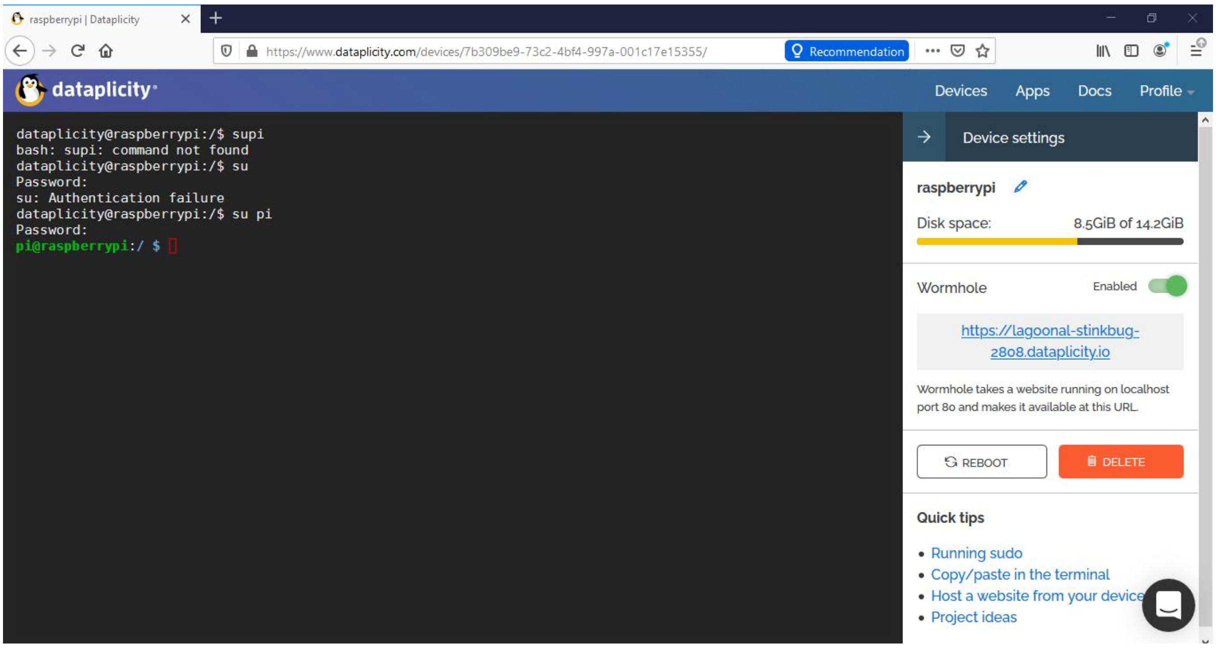
Type your email above

••••••••

Type your password above

Figure V.24: Sign in Dataplicity

We can access to the terminal of Raspberry Pi; just enter the user: **su pi** and the password:



raspberrypi | Dataplicity
https://www.dataplicity.com/devices/7b309be9-73c2-4bf4-997a-001c17e15355/
Recommendation
dataplicity
Devices
Apps
Docs
Profile
Device settings
REBOOT
DELETE

```
dataplicity@raspberrypi:/$ supi
bash: supi: command not found
dataplicity@raspberrypi:/$ su
Password:
su: Authentication failure
dataplicity@raspberrypi:/$ su pi
Password:
pi@raspberrypi:/ $ 
```

raspberrypi

Disk space: 8.5GiB of 14.2GiB

Wormhole Enabled

https://lagoonal-stinkbug-2808.dataplicity.io

Wormhole takes a website running on localhost port 80 and makes it available at this URL.

Quick tips

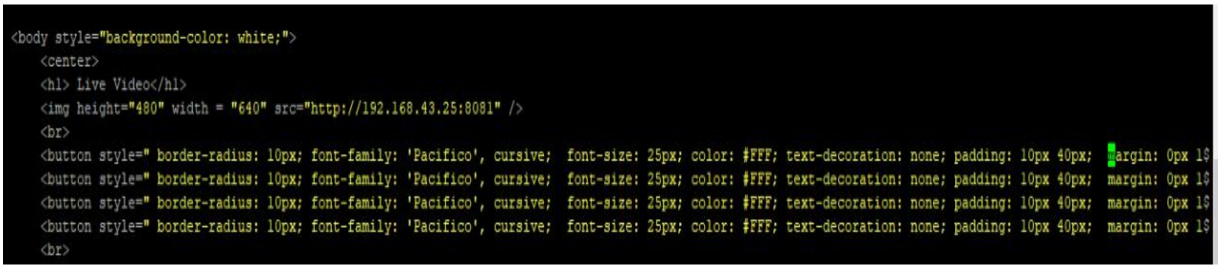
- Running sudo
- Copy/paste in the terminal
- Host a website from your device
- Project ideas

Figure V.25: Terminal of Raspberry Pi on Dataplicity

V.5 Create a web page and control the robot:

we will create a web page that contains a live video and four keys that control the direction of the robot, so we need three basic files:

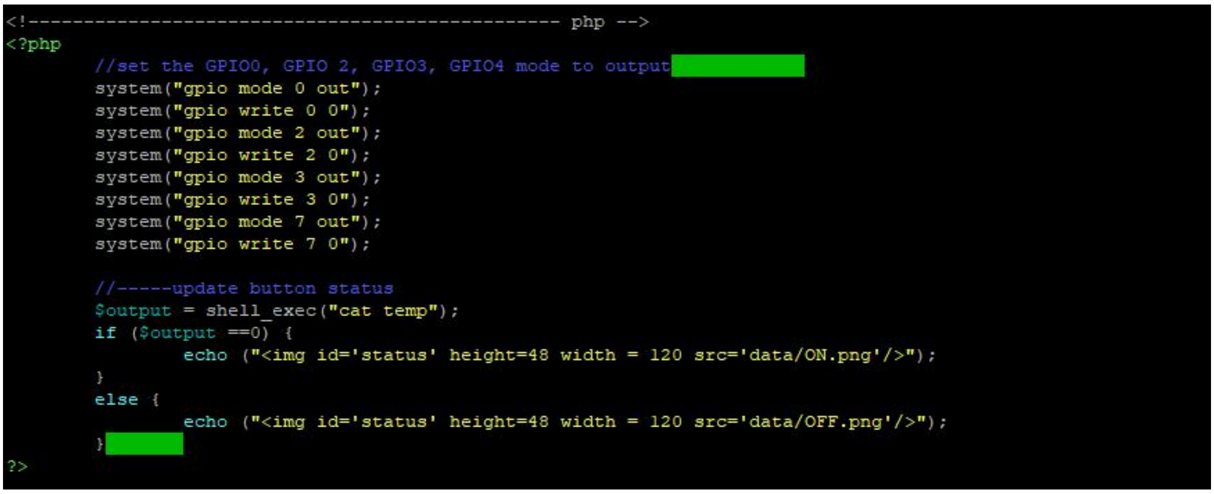
- ➢ First we need “file1.php” to create the web page and the server interaction. This page contains the   keys for control the motors and live video broadcast.
- ● Live video insertion and four control keys:



```
<body style="background-color: white;">
    <center>
    <h1> Live Video</h1>
    <img height="480" width = "640" src="http://192.168.43.25:8081" />
    <br>
    <button style=" border-radius: 10px; font-family: 'Pacifico', cursive;  font-size: 25px; color: #FFF; text-decoration: none; padding: 10px 40px;  margin: 0px 1$
    <button style=" border-radius: 10px; font-family: 'Pacifico', cursive;  font-size: 25px; color: #FFF; text-decoration: none; padding: 10px 40px;  margin: 0px 1$
    <button style=" border-radius: 10px; font-family: 'Pacifico', cursive;  font-size: 25px; color: #FFF; text-decoration: none; padding: 10px 40px;  margin: 0px 1$
    <button style=" border-radius: 10px; font-family: 'Pacifico', cursive;  font-size: 25px; color: #FFF; text-decoration: none; padding: 10px 40px;  margin: 0px 1$
    <br>
```

Figure V.26: Live video insertion and four control keys

- Adjust GPIO settings used on the Raspberry Pi:



```
<!----------------------------------------------- php -->
<?php
        //set the GPIO0, GPIO 2, GPIO3, GPIO4 mode to output
        system("gpio mode 0 out");
        system("gpio write 0 0");
        system("gpio mode 2 out");
        system("gpio write 2 0");
        system("gpio mode 3 out");
        system("gpio write 3 0");
        system("gpio mode 7 out");
        system("gpio write 7 0");

        //-----update button status
        $output = shell_exec("cat temp");
        if ($output ==0) {
                echo ("<img id='status' height=48 width = 120 src='data/ON.png'/>");
        }
        else {
                echo ("<img id='status' height=48 width = 120 src='data/OFF.png'/>");
        }
?>
```

Figure V.27: Adjust GPIO settings used on the Raspberry Pi

- ➢ the second file php (file 2.php) file receives the content from JavaScript file (file.js), and then the signals are applied to the GPIO used in the project by sending signals to the motors to control them, either 0v or 5v.
- ➢ JavaScript file (File.js) contains the functions for controlling Raspberry Pi ports.

❖ The results and discussion:

I was able to create a web page that contains live video broadcast as well a four keys that allow controlling the direction of the robot (right, left, forward, backward), but unfortunately, I could only control the robot in the left and right direction. The tow keys responsible for controlling the forward and backward direction did not work .

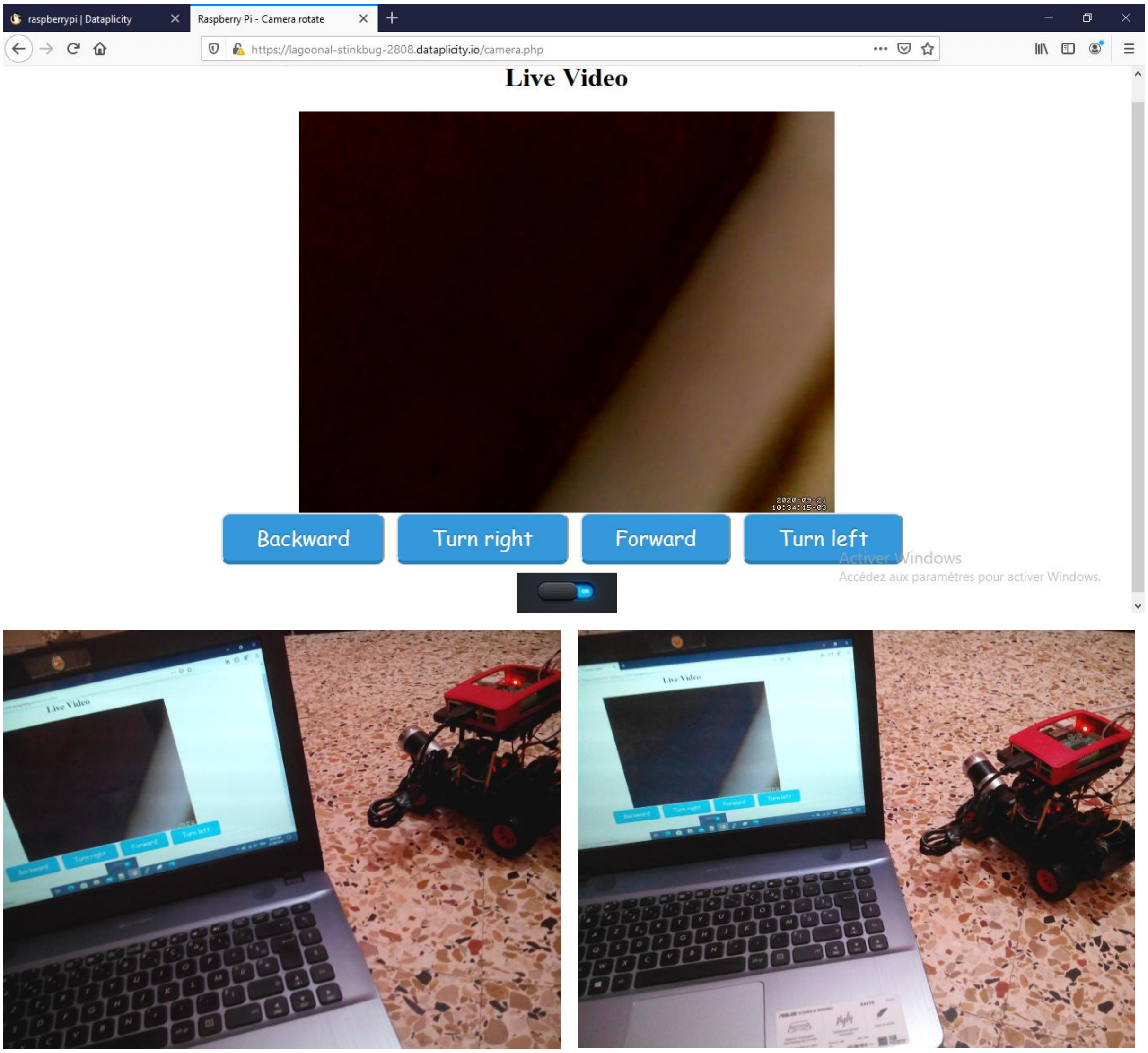

raspberrypi | Dataplicity
Raspberry Pi - Camera rotate
https://lagoonal-stinkbug-2808.dataplicity.io/camera.php
Live Video
2020-09-21
10:34:15-03
Backward
Turn right
Forward
Turn left
Activer Windows
Accédez aux paramètres pour activer Windows.

Figure V.28: Web page and control the robot

V.6 conclusion:

In this chapter, we have shown step by step how to build a robot. after that we saw the steps to start up the raspberry pi and how to prepare and program it, also we found a solution for access our Raspberry Pi remotely from anywhere to control the GPIOs and therefore driving the mobile robot and to view the video from the webcam through a web page.

General Conclusion

General Conclusion

At the end of this project, we were able to discover and consolidate our knowledge in different fields like: electronics and programming (programming languages: PHP, JavaScript, HTML, ...,etc), and world of robotics.

We remarked that the use of the robot in our daily live is increasing, so we find it compensates the human beings in many dangerous areas that need high precision such as surgery.

In this project, we were able to make and build a mobile robot depending on two (2) motors with a driver that controls its movement. The camera is the eye of the robot, it allow to recognize its way and discover the obstacles around it; this camera and the sufficient motors (two motors DC and motor driver) connected with the mind of the robot, and this last one is the Raspberry Pi.

So that we can control it through a web page by giving it orders (turn right, turn left, forward and backward) and it executes them.

Finally, this research is very interesting because it is in line with technological development, and at the same time we found many problems, including the lack of equipment, and Our knowledge of programming and electronics are insufficient when we started this project.

Bibliographies

***Chapter I:** Generalities on the Internet of Things*

[1]:Antar Shaddad Abdul-Qawy, Pramod P. J, E. Magesh, T. Srinivasulu , The Internet of Things (IoT): An Overview, International Journal of Engineering Research and Applications, Vol. 5, Issue 12, (Part - 2) December 2015, pp.71-82

[2]:P. Matta and B. Pant, INTERNET-OF-THINGS: GENESIS, CHALLENGES AND APPLICATIONS, Journal of Engineering Science and Technology, Vol. 14, No. 3 (2019) 1717 - 1750

[3]:BOUCETTA ABDEL HAKIM, July 7, 2019, Development of a smart mobile garden robot for supervising greenhouses, Master Memory university Mohamed Khider of biskra..

[4]:Sethi P, R. Sarangi S, Internet of Things: Architectures, Protocols, and Applications, Journal of Electrical and Computer Engineering, Volume 2017, Article ID 9324035, 25 pages

[5]: https://www.rfpage.com/what-are-the-major-components-of-internet-of-things/

[6]:Miao Wu, Ting-lie Lu, Fei-Yang Ling, ling Sun, Hui-Ying Research on the architecture of Internet of things 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), **DOI:** 10.1109/ICACTE.2010.5579493, IEEE

[7]:Keyur K Patel1, Sunil M Patel2 PG Scholar1 Assistant Professor2, May 2016 Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges, International Journal of Engineering Science and Computing, Volume 6 Issue No. 5

[8]:https://www.leverege.com/blogpost/iot-explained-how-does-an-iot-system-actually-work

[9]:https://data-flair.training/blogs/how-iot-works/

[10]:https://brainywit.com/how-iot-works/

[11]:https://www.talend.com/resources/what-is-data-processing/

[12]:https://www.kelltontech.com/kellton-tech-blog/internet-of-things-protocols-standards

[13]:Sh. Rajinder Singh,Dr. Satish Kumar , An Overview of World Wide Web Protocol (Hypertext Transfer Protocol and Hypertext Transfer Protocol Secure, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 5, May 2016

[14]:https://lora-alliance.org/about-lorawan

[15]:https://www.survivingwithandroid.com/iot-protocols-list/

[16]:https://www.sciencedirect.com/topics/engineering/zigbee-protocol

[17]:https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

[18]:https://www.omaspecworks.org/constrained-application-protocol-coap-is-iots-modern-protocol/

[19]:https://www.cloudamqp.com/blog/2019-11-21-what-is-amqp-and-why-is-it-used-in-rabbitmq.html

[20]:https://www.geeksforgeeks.org/xmpp-protocol/

[21]:Tanya Mohan Tukade, R M Banakar, Data transfer protocols in IoT-an overview, International Journal of Pure and Applied Mathematics, Volume 118 No. 16 2018, 121-138

[22]:https://iot-analytics.com/10-internet-of-things-applications/

[23]:https://data-flair.training/blogs/iot-applications/

[24]:https://www.smartgrid.gov/the_smart_grid/smart_grid.html

[25]:Safa A. Ahmed 1, Nahla Fadhil Alwan 2, Ammar Mohamed Ali 3, Overview for Internet of Things: Basics, Components and Applications, Journal Of University Of Anbar For Pure Science, Vol.12 : No.3: 2018

Chapter II: *Mobile Robot*

[1] **:** https://www.robotshop.com/media/files/PDF/timeline.pdf

[2] : https://en.wikipedia.org/wiki/Mobile_robot

[3] :https://internetofthingsagenda.techtarget.com/definition/mobile-robot-mobile-robotics

[4]: Mordechai Ben-Ari, Francesco Mondada, 2018, Elements of Robotics, Springer Nature,P:[2-4]

[5] : https://www.wikiwand.com/en/Mobile_robot

[6] : https://robots.net/robotics/types-of-robots/

[7]: http://www.allonrobots.com/types-of-robots.html

[8]:http://www.vacuumcleanerhistory.com/vacuum-cleaner-development/domestic-robots-history-and-future/

[9]:Wail Ikhenache , July 07,2019, Autonomous mobile robot, Master Memory university Mohamed Khider of biskra.

[10]: GERALD MIES, May 31, 2010, Military robots of the present and the future, Vol. 9, No. 1 (2010) 125–137, Germany

[11]:https://www.machinedesign.com/markets/robotics/article/21835000/whats-the-difference-between-industrial-robots

[12]:https://nccr-robotics.ch/research-areas/mobile-rescue-robots/ flying-robots/

[13]:https://www.online-sciences.com/robotics/aquatic-robots-swimming-robots-or-robot-fish-types-uses-cons-pros/

[14]:Claudio Semini, Pierre-Brice, Wieber Legged Robots, 30 Jan 2020, Springer Encyclopedia of Robotics, pp.1-11, In press. hal-02461604

[15]:Zirong Luo , Jianzhong Shang , Guowu Wei , and Lei Ren Module-based structure design of wheeled mobile robot Received: 26 October 2017 – Revised: 15 January 2018 – Accepted: 16 January 2018 – Published: 27 February 2018

[16]:https://dronebotworkshop.com/real-robot-004

[17]: Nourbakhsh, Illah R, and Roland Siegwart,2004 Introduction to autonomous mobile robots,The MIT Press Cambridge, Massachusetts London, England P:142.150

[18]:http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html

[19]: Sandeep Kumar Malu & Jharna Majumdar Kinematics, Localization and Control of Differential Drive Mobile Robot, Global Journal of Researches in Engineering, Volume 14 Issue 1 Version 1.0 Year 2014, Global Journals Inc. (USA)

[20]: https://suspensionsecrets.co.uk/ackermann

[21]:https://internetofthingsagenda.techtarget.com/definition/ embedded-system

[22]:https://www.uat.edu/robotics-and-embedded-systems-degree

Chapter III: *The Raspberry Pi*

[1]: Anand Nayyar, Vikram Puri, Raspberry Pi- A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 12, December 2015

[2] :https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/

[3] : https://opensource.com/resources/raspberry-pi

[4]: Raspberry Pi 4 Model B Datasheet Copyright Raspberry Pi (Trading) Ltd. 2019, Available at: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf

[5] : https://www.raspberrypi.org/documentation/faqs/

[6] : https://thepihut.com/products/raspberry-pi-3-model-b

[7] : https://www.element14.com/community/community/raspberry-pi/raspberrypi2

[8]: Gareth Halfacree ,«THE OFFICIAL RASPBERRY PI BEGINNER'S GUIDE V2», First published in 2019 by Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS

[9]: Gareth Halfacree «THE OFFICIAL RASPBERRY PI BEGINNER'S GUIDE V1» , First published in 2018 by Raspberry Pi Trading Ltd, Station Road, Cambridge, CB1 2JH, P:[10-17]

[10]: The Official Raspberry Pi Projects Book , Available at: raspberrypi.org/magpi

[11]:https://www.watelectronics.com/know-all-about-raspberry-pi- board-technology/

[12]:https://computer.howstuffworks.com/raspberry-pi2.htm

[13]:https://www.raspbian.org/

[14]:http://www.linuxandubuntu.com/home/raspbian-os-everyone-attached

[15]:https://www.fossmint.com/operating-systems-for-raspberry-pi/

[16]:Dr. Chafic BouSaba, Guilford College,Ms. Tiera Kazar, Guilford College, Dr. Will C. Pizio, Guilford College, 2016, Wireless Network Security using Raspberry Pi,American Society for Engineering Education

[17]:https://blog.idrsolutions.com/2015/01/the-8-cool-raspberry-pi-operating-systemsprojects-for-beginners/

[18]:https://kodi.wiki/view/Archive:OpenELEC

[19]:https://www.computerhope.com/jargon/a/arch-linux.htm

[20]:https://www.ionos.com/digitalguide/server/know-how/raspberry-pi-projects/

[21]:https://components101.com/microcontrollers/raspberry-pi-3-pinout-features-datasheet

Chapter IV: *Programming Languages*

[1]:Ankush Sharma, Aakanksha May 2018 Introduction to HTML (Hyper Text Markup Language) - A Review Paper International Journal of Science and Research (IJSR)

[2]:https://bethsoderberg.com/blog/html-basics-elements-tags-and-document-structure/

[3]:https://www.w3.org/TR/html401/struct/global.html#h-7.4

[4]:https://www.tutorialspoint.com/html/html_tutorial.pdf

[5]:https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf

[6]:https://www.tutorialspoint.com/css/css_tutorial.pdf

[7]:https://techterms.com/definition/css

[8]:Stack Overflow Documentation CSS Notes for Professionals book June 2019, Available at: https://goalkicker.com/CSSBook

[9] :https://www.w3schools.com/cssref/

[10]:Matt Doyle , 2010, ”Beginning PHP 5.3”, by Wiley Publishing, Inc., Indianapolis, Indiana ISBN: 978-0-470-41396-8, the United States of America, P:3.

[11]: PHP Hypertext Preprocesso. Available at: https://www.tutorialspoint.com/php/php_tutorial.pdf

[12]:Stack Overflow Documentation PHP Notes for Professionals book, Available at: https://goalkicker.com/PHPBook (Free Programming Books)

[13]:https://developer.mozilla.org/en-US/docs/Web/JavaScript

[14]:https://websitesetup.org/javascript-cheat-sheet/

[15]: JavaScript language. Available at: https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf

[16]:Steve Suehring, JavaScript Step by Step, Third Edition,2013

[17]:https://www.javatpoint.com/javascript-data-types

[18]:https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_Operators

[19]:https://www.w3schools.com/js/js_function_definition.asp

Chapter V: *Building A Robot And Control It*

[1] :https://phoenixnap.com/kb/ssh-to-connect-to-remote-server-linux-or-windows

[2] :https://www.raspberrypi.org/documentation/remote-access/vnc/

[3] :http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf

[4] :https://uk.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/dc-motors-guide

[5] :http://www.camsecure.co.uk/USBwebcamIPwebcam.htm

[6] :https://www.electronicwings.com/raspberry-pi/how-to-use-wiringpi-library-on-raspberry-pi

[7] :https://www.hostinger.com/tutorials/what-is-apache

[8] :https://www.mysqltutorial.org/what-is-mysql/

[9] :https://techcrunch.com/2016/12/13/dataplicity-lets-you-access-your-raspberry-pi-from-anywhere/