Faculty of Electrical and Electronic Engineering Technology

THE DEVELOPMENT OF AN IOT BASED AUTOMATED GARMENT RACK WITH ESP8266

MUHAMMAD ALIF BIN ABDUL RAHMAN

Bachelor of Electronics Engineering Technology (Industrial Electronic) with Honours

2023

THE DEVELOPMENT OF AN IOT BASED AUTOMATED GARMENT RACK WITH ESP8266

MUHAMMAD ALIF BIN ABDUL RAHMAN

A project report submitted.

in partial fulfilment of the requirements for the degree of Bachelor of Electronics Engineering Technology with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI KEJUTERAAN ELEKTRIK DAN ELEKTRONIK

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek: The Development of An IOT Based Automated Garment Rack with ESP8266

Sesi Pengajian : 1 2023/2024

Saya Muhammad Alif Bin Abdul Rahman mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

4. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
1. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
2. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
3. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	**TIDAK TERHAD**	

(TANDATANGAN PENULIS)
Alamat Tetap: No 83, Peringkat 2 felda

Ijok, Selama, 34120, perak

Darul Ridzuan.

Tarikh: **12/1/2024**

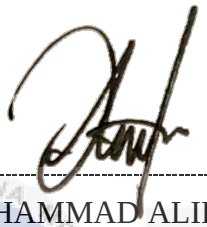
(COP DAN TANDATANGAN PENYELIA)
TS. KHAIRUL ANUAR BIN A. RAHMAN
Jurutera Pengajar
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Tarikh 31/1/2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled The Development Of An IOT Based Automated Garment Rack with ESP8266 is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

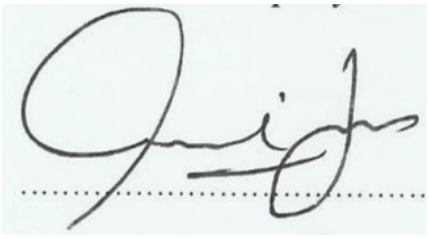
Signature : 

Student Name : MUHAMMAD ALIF BIN ABDUL RAHMAN

Date : **12/1/2024**

APPROVAL

I approve that this Bachelor Degree Project 1 (PSM1) report entitled “Project Title” is sufficient for submission.

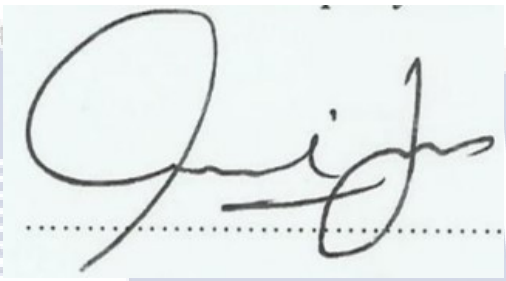
Signature : 

Supervisor Name : Ts. KHAIRUL ANUAR BIN A. RAHMAN

Date : **31/1/2024**

APPROVAL

I hereby declare that I have checked this project report, and in my opinion, this project reportis adequate in terms of scope and quality for the award of the degree of Bachelor of ElectricalEngineering Technology with Honours.

Signature : 

Supervisor Name : Ts. KHAIRUL ANUAR BIN A. RAHMAN

Date : **31/1/2024**

DEDICATION

To my beloved mother, father,

siblings

and friends, thanks

ABSTRACT

People have needed different kinds of clothes from the beginning of time, and this requirement has only gotten more complicated over time. Maintaining one's wardrobe's cleanliness is a crucial duty that involves a few processes, such as washing, drying, and hanging the clothing. This notion came to be after observing the issues that society frequently faces. For those who have employment and limited free time at home, this process results in a large rise in the amount of time required. One of the frequent issues is that clothing doesn't dry correctly because of erratic weather conditions. The previous system's drying procedure calls for human work to pull in and pull out the clothing in accordance with the weather. It will take time and effort to successfully complete this process. The objective of this project is to develop an Internet of Things-based automatic garment rack that will make it simpler for consumers to dry their clothes. It is predicted that this system will work in line with the environment, specifically the status of the clothing and the weather, to save time and energy.

ABSTRAK

Sejak awal manusia, orang ramai mempunyai keperluan untuk pelbagai bentuk pakaian, yang semakin kompleks dari masa ke masa. Menjaga kebersihan almari pakaian seseorang adalah tugas penting yang memerlukan beberapa langkah, termasuk membasuh, mengeringkan dan menyidai pakaian. Idea ini diperoleh selepas melihat masalah yang sering berlaku kepada masyarakat. Proses ini menyebabkan peningkatan ketara dalam jumlah masa yang diperlukan, khususnya bagi individu yang mempunyai kerjaya dan mempunyai sedikit masa untuk diluangkan di rumah mereka. Salah satu masalah biasa ialah pakaian tidak dapat kering dengan baik kerana faktor cuaca yang tidak menentu. Proses pengeringan untuk sistem lama memerlukan tenaga buruh manusia untuk mengambil keluar masuk pakaian mengikut syarat. Untuk menyelesaikan proses ini dengan jayanya akan memerlukan masa dan usaha. Matlamat projek ini adalah untuk mencipta rak pakaian automatik yang berasaskan Internet Perkara dan akan memudahkan pengguna mengeringkan pakaian mereka. Bagi menjimatkan masa dan tenaga, sistem ini dijangka akan berfungsi mengikut persekitaran, khususnya keadaan pakaian dan cuaca.

ACKNOWLEGMENT

First and foremost, I would like to express my gratitude to my supervisor, Ts. Khairul Anuar Bin A. Rahman for their precious guidance, words of wisdom and patient throughout this project.

I am indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the financial support or information sharing through library research enables me to accomplish this project. Not forgetting my fellow colleague, friends for willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents, and family members for their love and prayer during the period of my study. An honourable mention also goes to Muhammad Adib Bin Abdul Rahman and Fauziah Binti Zakaria for all the motivation and understanding.

Finally, I would like to thank all the staffs at the Seksyen Projek and Lab helper (UTeM), fellow colleagues and classmates, the faculty members, as well as other individual who are not listed here for being co-operative and very helpful.

TABLES OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

SYMBOL	TITLE	PAGE
τ	Torque	**42**
P	Power	**42**
π	mathematical constant approximately equal to 3.14159	**42**
ω	The angular velocity	**42**

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

Since the dawn of time, clothing has been a need for humans, and it has changed with time. Maintaining the cleanliness of your clothing requires a time-consuming process that includes washing, drying, and hanging. Particularly for those with careers and little free time at home, this process takes a lot of time. The most common method of drying clothes is on a traditional clothesline, which is a fixture in most homes. A portable garment rack has taken the place of the clothesline in modern times. Since the garment rack is simple to set up and relocate, those who rent a room or live in apartments are the most likely to use it. The garment rack makes it simple for customers to transport still-wet items to the inside during rainy days and to the outside during sunny days to continue the drying process. Since it will take a lot of time, the user does not have to choose every item of clothing and hang it back. This idea proposes features that would allow customers to manage their outdoor-dried clothing using a clothes rack that had temperature and humidity sensors. It will automatically move dependent on the situation after detecting whether the clothing is wet or dry. While sending the input to Arduino, the raindrop sensor detects the raindrop.

1.2 Problem Statement

A busy lifestyle is one of the issues with how the project's idea was developed. Couples with busy schedules and dual careers are the target audience for this issue. Even though washing machines and dryers can speed up and simplify the process, most people can afford to own bot especially the dryer. Their clothes still need to be hung up to dry.

1.3 Project Objective

The major goal of this project is to offer a solution to the issue with garments drying in the sun. The following objectives are more precise:

a) Implement algorithms or strategies to optimize human energy usage, ensuring efficient operation of the garment rack.

b) To develop an automated garment rack that can work automatically and efficiently in different weather to dry the clothes.

c) To evaluate how the Arduino UNO work in controlling the garment rack in different temperature and humidity.

1.4 Scope of Project

This project is primarily concerned with applications in private homes. The drying process will be simplified for users, which will cut down on the amount of time it takes. This will be an advantage. This project will be usable by anyone and everyone in general, but its primary target audience will be people with constrained resourcesin terms of both time and money, in particular working people. The use of the sensor will be applied to solve the problem and ensure that the project is carried outin an orderly fashion. The sensor will gather information about its surroundings and respond appropriately.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

People are preoccupied with work or other activities outside the home in today's modern civilization. Every day used clothing must be washed. Malaysia is located on the equator, or khatulistiwa, which causes the weather to be unpredictable. It's sweltering and muggy outside. Sometimes it rains, other times it's hot. When clothes are hung outside and no one is home, no one can remove the garment rack from the rain, which results in the clothes being damp. By relocating the rack beneath the roof during downpours, this idea will help keep the clothing dry.

2.2 History of Clothing

Clothing (also known as clothes, apparel, and attire) are items worn on the body. Clothing is often made of fabrics or textiles, but it has evolved over time to include garments fashioned from animal skin and other thin sheets of materials and natural things present in the environment. Clothing is mostly exclusive to humans and is a component of all human societies. Gender, body type, Clothing (also referred to as garments, apparel, and attire) area unit things worn on the body. Covering is usually fabricated from materials or textiles; however it's evolved over time to incorporate clothes designed from animal product and alternative skinny sheets of materials and natural things gift within the setting.

Covering is generally exclusive to humans and may be a part of all human societies. Gender, body type, social group influences, and geographic issues all influence the number and magnificence of garments worn. Covering covers the trunk, footwear covers the feet, gloves shield the hands, and caps and headgear shield the pinnacle. Eyewear and jewellery don't seem to be unremarkably thought of covering things, though they play a big role in fashion and covering.

as costume societal influences, and geographic considerations all influence the amount andstyle of clothes worn. Clothing covers the torso, footwear covers the feet, gloves protect thehands, and caps and headgear protect the head. Eyewear and jewellery are not commonly considered clothing items, although they play a significant role in fashion and clothing as costume.[1]

Clothing will offer protection from the weather, rough surfaces, sharp stones, rash-causing plants, and bug bites by acting as a barrier between the skin and therefore the atmosphere.[2] Covering will offer insulation against cold or hot temperatures, additionally as a hygiene barrier, keeping infectious and unsafe components removed from the body. It will shield the foot from injury and discomfort, additionally as aid in navigation during a style of conditions. Covering conjointly protects against ultraviolet light rays. It may be wont to scale back glare or improve vision in harsh settings, like brimmed hats. Covering is utilized for injury interference in numerous vocations and occupations, sports, and warfare. Covering with pockets, belts, or loops may be wont to carry things whereas releasing up one's hands.

Clothing conjointly has vital social group implications. Covering could be a versatile social norm. it's going to imply modesty. It may be embarrassing to be naked ahead of individuals. In several components of the planet, not carrying covering public ally that exposes private parts, breasts, or buttocks is also thought-about infraction. The foremost usually encountered minimum found cross-culturally and despite atmosphere is bone space or sex organ coverage, reflective social convention because the basis of customs. Covering may also convey social position, affluence, cluster affiliation, and individualism.

Some sorts of personal protecting instrumentality are covering, like coveralls, chaps, or a doctor's white coat, with a similar maintenance and clean up necessities as different textiles (boxing gloves perform each as protecting instrumentality and as a sparring weapon, therefore the instrumentality facet rises higher than the glove aspect). Protecting accessories are additional specific varieties of protecting instrumentality, like face shields. At the opposite finish of the spectrum, self-contained diving suits or house suits or form-fitting body covers that perform as a type of dress while not being garments intrinsically, however contain enough advanced technology to be additional of a tool than a garment. As wearable technology embeds helpful gadgets directly into the material, this line can still blur; the sanctioning technologies embody radical low power consumption and versatile electronic substrates.

Clothing may also be a private installation (ice skates, roller skates, lading pants, different out of doors survival gear, one-woman band) or a concealment system (stage magicians, hidden linings or pockets in craftsmanship, integrated holsters for hid carry, merchandise- laden trench coats on the black market wherever the aim of the covering oftentimes carries over into disguise). Associate outfit or ensemble could be a form of apparel that serves a definite purpose, whether rhetorical or practical.

Figure 2.1: Evolution of human clothing.

2.3 Garment Rack/Clothesline

A garment rack is a rail used in retailers to showcase apparel such as shirts and coats (Collin English dictionary, md). The garment rack is utilized in this

project to replace the clothesline at home, especially for people who live on a terrace or in an apartment with little spaceto dry their clothes. Outside, to make it easier to transport, the present simple garment rack or clothing rack on the market was built with a single rode and two pairs of rollers. [3]

Figure 2.2: Garment rack with roller.[3]

Because of Malaysia's uncertain weather, a garment rack was proposed in this proposal. Malaysia, located near the equator, has a tropical environment with high temperatures and rain all year, according to researchers. While monsoonal rainfall is growing more variable and unpredictable from year to year one source stated that Malaysia has tropical weather year-round [4], yet the climate is often rather humid due to its proximity to water. According to the same source, Malaysia's weather is never overly hot, with temperatures ranging from a warm 20 °C to 30 °C on average throughout the year [5][6]; however, the highlands receive colder temperatures. With uncertain weather, managing the weather isimpossible, and as a result, hanging cleaned clothing on a clothesline on an open field or terrace anguishes the user of the pouring likelihood.

2.4 IoT

The Internet of Things may be a network that connects something to the net mistreatment predefined protocols info and knowledge and data sensing instrumentation to share information and communicate to realize sensible recognitions, locating, tracing, monitoring, and administration. During this paper, we tend to be coated in brief what IOT is, however IOT supports various technologies, its design, characteristics, and applications, the IOT purposeful read, and what the long run difficulties for IoT square measure.[7]

Figure 2.3: Fundamental components of IoT.[7]

The Internet of Things (IOT) conception was coined in 1999 by a member of the frequency Identification (RFID) development community, and it's recently become additional relevant to the sensible world, attributable to the proliferation of mobile devices, embedded and omnipresent communication, cloud computing, and knowledge analytics. Contemplate a world during which billions of things will sense, communicate, and share data via public or non-public web Protocol (IP) networks [8]. Knowledge is collected, evaluated, and accustomed begin action on these interconnected things on an everyday basis, providing a wealth of intelligence for coming up with, management, and

deciding. This can be the web of Things world (IOT). The web of Things (IOT) could be a network of physical things, in keeping with the foremost current definition. The web is not any longer simply a network of computers; it's evolved into a network of devices of all shapes and sizes, together with vehicles, sensible phones, home appliances, toys, cameras, medical instruments and industrial systems, animals, people, and buildings, all connected, all act and sharing data supported predefined protocols so as to realize sensible reorganizations, positioning, tracing, safe and management, and even personal time period on-line observation and change[10]. We tend to classify IOT into 3 teams, as follows:

The internet of things is formed from 3 components:

(i) People to people

(ii) People to machine

(iii) Machine to machine, Interacting via the internet.

The Internet of Things (IoT) may be a thought and a paradigm that considers the pervasive presence within the surroundings of a range of things objects that may act with different one another and join forces with other things objects via wireless and wired connections and distinctive addressing schemes to make new applications services and win common goals. The analysis and development hurdles to make a wise world area unit huge.

2.5 Smart Home

A smart home system has highly smart features that make living easier. You

can control home appliances with your smart phone in smart home automation. The main goal of a smart home system is to reduce human effort, electricity consumption, and to assist the elderly and children. As wireless communication technology advances, we will be able to access and control home appliances remotely. In terms of security, if an intruder attempts to enter forcefully, an antitheft system will sound an alarm and inform the owner, allowing the owner to take immediate action. Another dangerous serious concern is gas leaking in the home. LPG gas is a combustible gas. As a result, the system will raise an alarm and warn the user in this scenario as well. Many technologies, such as GSM, Bluetooth, and IOT, can be used for home automation and security [9]. Home appliances can be controlled using GSM technology by sending messages to the system. Bluetooth technology makes use of smart phones and an Android software. Whereas IOT, or the internet of things, is a somewhat large field. IoT refers to the interconnection of physical devices, buildings, and other items that are integrated with electronics, software, sensors, and internet connectivity. It is feasible to automate and safeguard a home with IOT extremely well by using a server generated over Wi-Fi for communication [11], sensors such as a gas leakage sensor, a motion sensor for an intruder, and a smartphone, tablet, or laptop as a GUI. The PIC microcontroller-based smart home system is a low-cost and efficient controller. The system employs ZigBee, a short- range wireless communication module. A smoke sensor is used for security. If smoke is detected, an SMS alarm is sent to the user.

Figure 2.4: Smart Home

2.6 Arduino

The Arduino boards were built at the Ivrea Interaction Design Institute. These Arduino boards evolve in response to new requirements and issues, evolving from simple 8-bit boards to many products such as IoT (Internet of Things) applications, 3D printing, wearable computing, and embedded computing. All the boards are open source, allowing users to design their own and modify them to fulfil their requirements. [12]

Over the years, Arduino boards have been used to construct hundreds of projects ranging from basic domestic items to complicated scientific apparatus.[13] A multinational community of designers, artists, students, programmers, amateurs, and professionals have gathered around this open-source stage, and their contributions have added up to an astounding amount of available information that may be of immense use to both novices and specialists alike.[14]

All boards are totally open source, allowing users to develop them on their own and then modify according to their own requirements.[15] Arduino boards have been used to make utilized of projects ranging from basic domestic items to complicated scientific instruments throughout the years.

An international community of designers, artists, students, programmers, hobbyists, and professionals have gathered around this open-source stage, and their contributions have added up to an incredible quantity of available knowledge that may be of tremendous value to both novices and experts alike. There are different Arduino boards:

I. NodeMCU ESP8266

II. LilyPad Arduino

III. Red Board

IV. Arduino Mega (R3)

V. Arduino Leonardo

2.6.1 Node MCU

NodeMCU may be a low-cost open source IoT platform.It at first included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Frameworks, and equipment which was based on the ESP-12 module.NodeMCU is an open source firmware for which open source prototyping board plans are accessible. The title "NodeMCU" combines "hub" and "MCU" (micro-controller unit). As Arduino.cc started creating new MCU boards based on non-AVR processors just like the ARM/SAM

MCU utilized within the Arduino Due, they required to alter the Arduino IDE so it would be generally simple to alter the IDE to back interchange toolchains to permit Arduino C/C++ to be compiled for these modern processors.[16] They did this with the presentation of the Board Chief and the SAM Center. A "center" is the collection of program components required by the Board Director and the Arduino IDE to compile an Arduino C/C++ source record for the target MCU's machine dialect. A few ESP8266 devotees created an Arduino center for the ESP8266 WiFi SoC, prevalently called the "ESP8266 Center for the Arduino IDE".

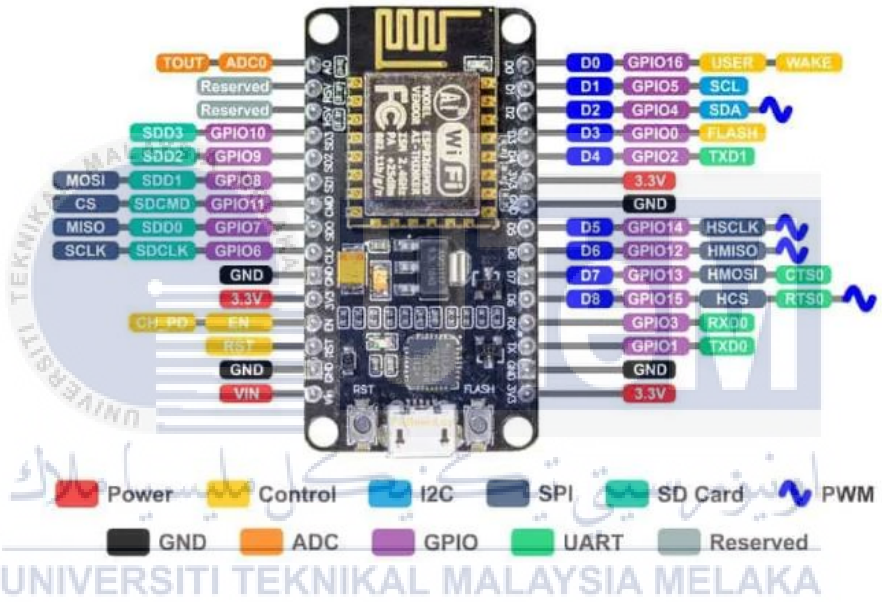


Figure 2.5 NodeMCU ESP8266.

2.6.2 Lilypad Arduino

This board is created to be readily integrated into wearable applications. The other Arduino boards share the same features, such as a lightweight, spherical container designed to minimize snagging and profile, as well as wide tabs that can be sewed down and attached with conductive thread. This Arduino board is made up of an Atmega 328 with the Arduino bootloader and a few extra components to keep it as

minimal as possible. This board's power supply ranges from 2V to 5V and features wide pin- out holes for easy sewing and connection. Each pin is wired to positive and negative terminals and is used to operate input and output devices such as lights, motors, and switches.[8]

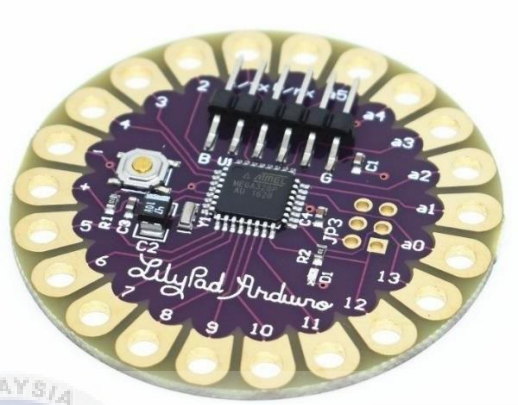


Figure 2.6: Lilypad Arduino.

Leah Buechley created this Arduino technology, and each LilyPad was cleverly constructed with huge connecting pads allowing them to be stitched into clothing. There are numerous input, output, and sensor boards available, all of which are washable.

2.6.3 Arduino MEGA

The Arduino Mega is a Microcontroller that runs on the Atmega2560. It has 54 digital input and output pins, 14 of which are utilized for PWM output, 16 foranalogue inputs and 4 for the hardware serial port of the UART. There are pins such as a 16 MHz crystal oscillator, USB connection, RESET pin, anICSP header, and a power jack.[17]

Figure 2.7: Arduino MEGA.[8]

In addition to the AREF, this Arduino Mega has SDA and SCL pins. There are two new IOREF pins near the RESET pin that allow the shields to adapt to voltage supplied by the board. The other is unconnected and will be used in the future. Features of the Arduino Mega R3:

a. Atmega2560 Microcontrollerb.Input voltage – 7-12V

c.54 Digital I/O Pins (14 PWM outputs)

d.16 Analog Inputs

e.256k Flash Memory

f. 16Mhz Clock Speed

2.6.4 Arduino Leonardo

Leonardo Arduino board is a Microcontroller board that is based on the

data sheet for the Atmega32u4. It contains 20 digital input and out pins, with 7 pins used for pulse width modulation output and 12 pins used for analogue input, as well as a 16MHz crystal oscillator, a micro-USB connector, a RESET pin, and a power socket.[8]

Figure 2.8: Arduino Leonardo.[8]

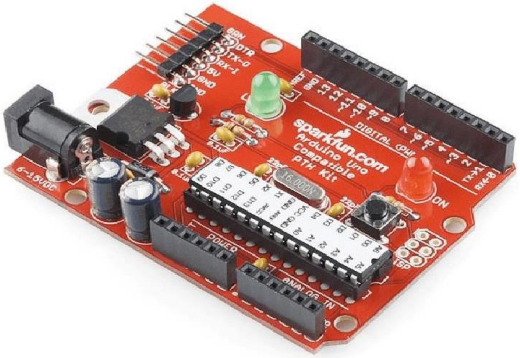
It comes with everything you need to support the microcontroller; simply connect itto a computer through USB or power it is using an AC-to-DC adapter or battery to getstarted. The Leonardo differs from all previous boards in that the Atmega32u4 includes built-in USB connection, removing the requirement for an additional processor.

This enables the Leonardo to appear as a mouse and keyboard, as well as a virtual (CDC) serial COM port, to a connected computer. It also has other ramifications forthe board's conduct.

2.6.5 Arduino Red board

The Arduino red boards are programmed using a mini-B USB cable and the Arduino IDE software. It will function in Windows8 without any changes to the security system. The Arduino red board is a lot more consistent since it uses USB and FTDI chipsthat are flat on the back.[8]

It is quite simple to create and incorporate into the project design. Simply plug in the board, choose an Arduino UNO from the menu, and you're ready to upload the application. Using the barrel jack, you may



operate the RedBoard via USB connection.

Figure 2.9: Arduino Redboard.[8]

2.6.6 Arduino UNO

The Arduino UNO R3 is a brand-new board with various additional functionality compared to earlier Arduino boards. The Arduino UNO employs the Atmega16U2 rather than the 8U2 and hence has a quicker transfer rate and greater memory. There is no need for additional devices for Linux and Mac, and the UNO may be used as a keyboard, mouse, joystick.[18]

Figure 2.10: Arduino UNO.

2.7 Previous Related Research

People nowadays spend less time at home due to corporate culture, and some seek the assistance of cutting-edge technologies for safety and other fundamental needs. These issues are addressed by smart home automation devices. This article, titled 'Design and development of Automatic retractable roof for clothesline with notification,' provides a solution to this problem. The main goal of this project is to keep washed garments from becoming damp while hanging on a clothesline. This issue happens when the user is not in a reachable location while it rains. A movable roof that detects rain and protects the clothesline from it. As a result, the suggested hardware includes three modules: sensing, driving, and notification systems. The NodeMCU microcontroller controls the sensing and driving components. The notification system includes the user's login information as well as the status of the retractable roof. The driving mechanism, as with many retractable roofs, is a tubular motor. Because tubular motors operate via servo mechanisms, servo motors are used as driving modules in this prototype. And, unlike past iterations, this proposed system is simpler to develop and less expensive to build. In this article, the researcher creates an automated roof for clothesline when it detects rain.

The rain action is detected by a rain sensor. A dual-role device, such as a switch and intensity measurement device, is suitable for this purpose. This module is made up of two boards: the detecting board and the control board. First, the control unit includes an LED indicator as well as a potentiometer for various uses. The control board's analogue signals measure the intensity of rainfall. Rainfall is detected via a digital signal, on the other hand. This rain sensor signal operates in reversed logic. A high output means that there will be no rain

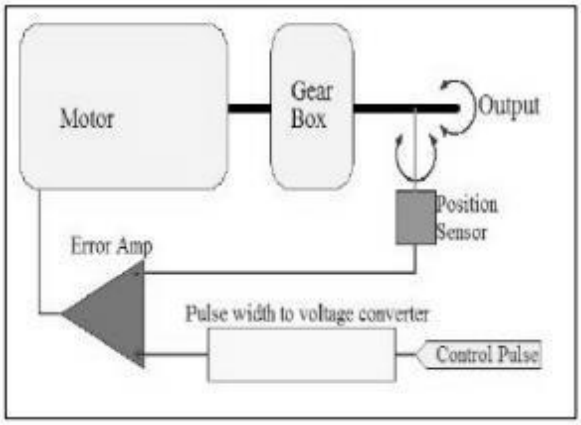


Figure 2.11: Rain sensor controller mechanism with DC motor

In contrast to DC motors, the DC motors may position themselves precisely at an angle using control signals. The location of the motor shaft is determined by the control signal. As the result, this basic DC motor is employed in a variety of applications. The size and torque od DC motors varies. The Node MCU power supply is mostly used to power low voltage, mini sized servo motor.[19] The GFM925W DC motor is recommended for the ARRC system toque and rpm for pull or push heavy duty load.[20]

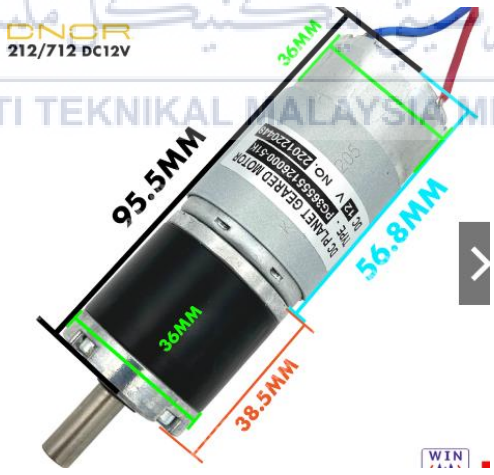


Figure 2.12: DC motor design

In this project, a NodeMCU serves as the controller. As an input, a rain sensor continuously detects rain. Interaction with the web server is handled by the ESP8266 Wi-Fi module.[21] The roof mechanism is driven

by an MG 995.

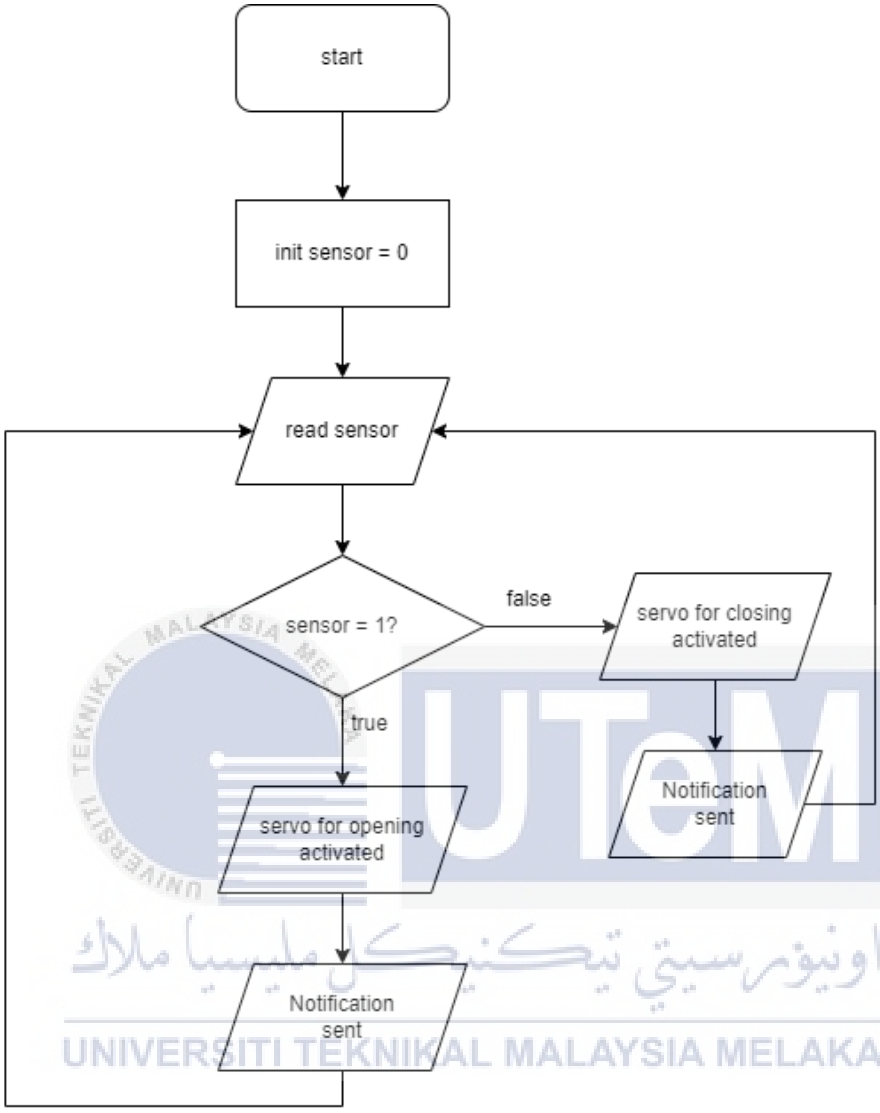


Figure 2.13: Rain sensor flow chart

2.8 Comparison

Based on prior research projects that have been handled, several differences in methodology may be investigated and contrasted. As a result, the table below analyses the research publications in terms of technique, benefits, and downsides.

No	Method	Advantages	Disadvantages
1	Humidity sensor for Arduino based Smart Irrigation System using GSM Module, Humidity Moisture Sensor, Weather and Inverter.	Complete sensor	Expensive
2	Design of Raindrop detector based on Arduino	Easy to control. Store data in app	
3	Soil moisture monitoring using IoT enabled. 22 utilize sensors with neural networks	Store data Reduce water usage	
4	Rainfall Monitoring Using Acoustic Sensors with a mounted Wiznet W5100 Ethernet Shield, which connects to a network for remote data access.	Has alarm system to alert	Large area used
5	Remote control via Android smart phone of 22 utilize-based automated irrigation systemusing moisture sensor via the HC-05 module to an Android smart phone.	Wireless	

Table 2.1 Comparison of previous Research

Summary

The domain and the difficulties associated with it were discovered and explained in this chapter. This chapter also includes other relevant works that have been reviewed to find a solution for this project. The methods and devices utilized in related work are contained in the ideas. The approach provided in this chapter is also based on research from related areas. While in this project are arduino uno is a versatile microcontroller board that allows users to build and program various electronic projects. It provides a wide range of inputs and outputs, making it suitable for integrating sensors and others. NodeMCU, on the other hand, is an open-source IoT platform based on the ESP8266 Wi-Fi module. It combines the capabilities of a microcontroller with built-in Wi-Fi connectivity, making it ideal for connecting devices to the internet and enabling remote control and monitoring in a smart home context.

When Arduino Uno and NodeMCU are employed in a smart home setup, they can work together to enable automation and interconnectivity. For instance, Arduino Uno can be used to interface with various sensors such as temperature, humidity, motion, and light sensors, gathering data about the home environment. Through NodeMCU, users can develop a web server or use cloud-based platforms to monitor and control the connected devices in their home. This includes adjusting thermostat settings, controlling lights, managing security systems, and more.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, process methodology was disseminated as advice throughout the project's process. There are two basic techniques to development: Agile and Waterfall. Agile was chosen as the development process methodology for this project. The Agile technique was developed in direct response to Waterfall detractors who felt that there was too much room for problemsto go ignored until the project was nearing conclusion. Teams divide each feature down into the smallest discernible bit of labour and provide incremental value over time with Agile development.

3.2 Methodology

This project will be developed using agile methodology because it is appropriate for software and hardware development. This methodology saves a lot of time by optimizing development tasks and avoiding errors that can occur during the planning stages. In general, agile has six lifecycle phases that must be used to finish the entire project, 'The six phases are requirement, design, build, test, deploy, and review,' as shown in Figure 3.1 below.

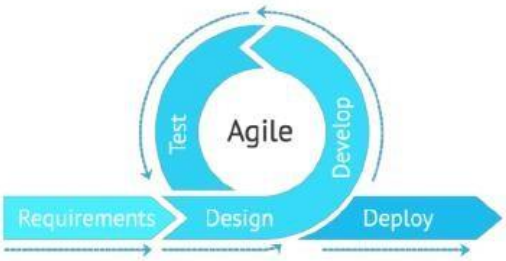


Figure 3.1: Agile model.

3.2.1 Requirement Phase

The specific needs for developing a project were defined during the requirement phase. This phase is divided into two sub-phases: requirements determination and analysis study. To collect data for the project, all requirements including objective, problem statements, and scope were determined. Problems arose among the residents in the nearby area. Instead, the project was recorded in the form of a proposal document. After the planned project has been approved for continuation, the following stage is to decide on the project's title. After the problem and proposed project were discovered and confirmed, an aim was presented where the current garment rack had to be improved to a better system. Analysis studies are required to assess the previous project or any related work for the developer to use as a reference. It is also suggested that the current flaws be identified in order to improve the features. According to the project aim, this project necessitated an examination of earlier relevant work as well as an analysis based on an IoT that can be fitted to the specified objectives.

3.2.2 Design Phase

A justification for project design must be completed during this phase to ensure at least a general understanding of how the finished result should look. The offered design can serve as a reference for achieving the project's objectives. In other words, this phase project must havean outcome on how the project's architecture is seen at the end of this phase.

3.2.3 Develop Phase

The development phases demonstrate a hands-on approach to completing the entire project. It helps with hardware and software implementation, such as putting up hardware components and coding for the system.

3.2.4 Test Phase

During this phase, the entire project will be tested. Every aspect of the projects was examined, including the software and hardware. This phase was required to determine whether the final look met the requirements or needed to be improved.

3.2.5 Deploy Phase

At this point, the projects were ready for utilization in environments by product end customers. The project will be tested in the field to assess how well it works. Because this project is focused on development, this phase will be the project's restriction, as described in the previous chapter.

3.3 Flowchart

The most critical part in ensuring the success of this project is project planning. Proper planning should be 27tilized2727 to ensure that the project is built and that the obstacles encountered while doing so are 27tilized27. To execute the project on schedule, a methodical flow chart is required. This project's overall flow chart is illustrated below.

3.3.1 Flowchart of FYP1

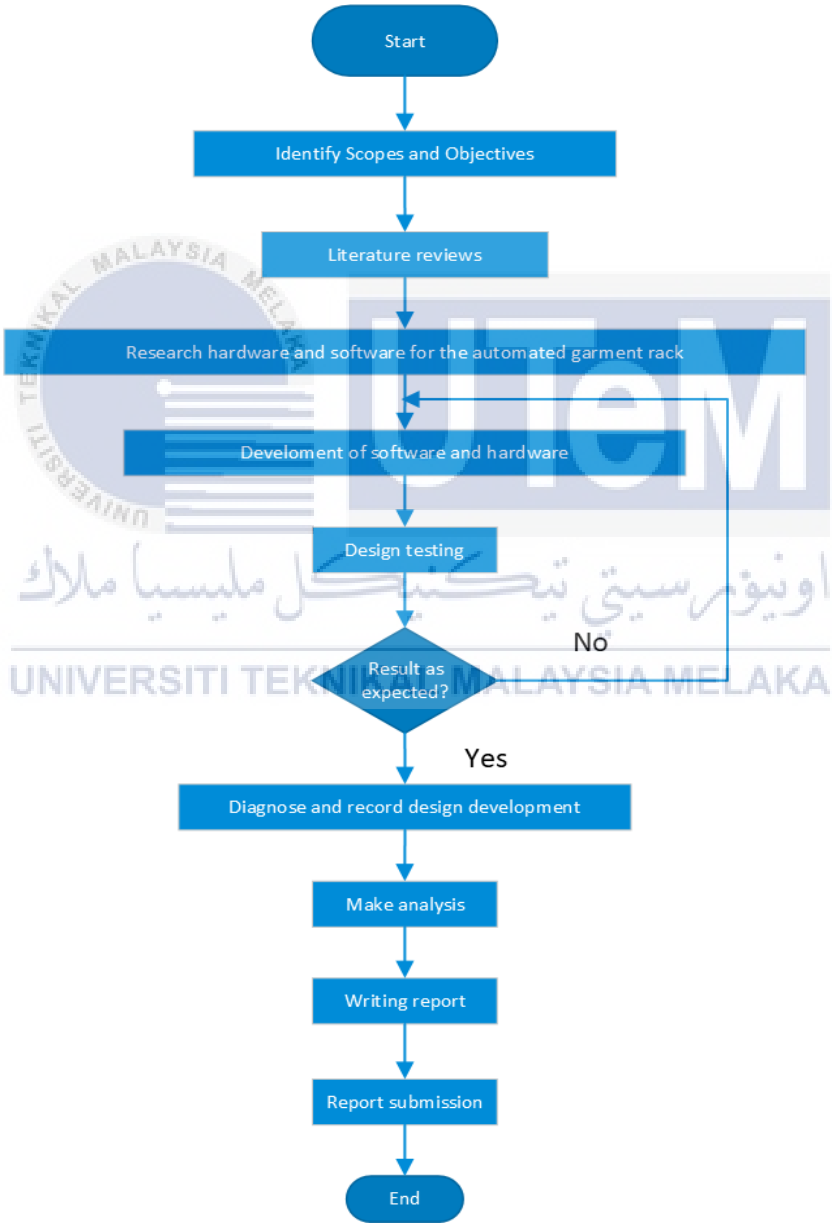


Figure 3.2: Flowchart of FYP1

3.3.2 Ghattchart of FYP1

Week / Activity	W1	W2	W3	W4	W5		W6	W7	W8	W9	W10	W11	W12	W13	W14
Proposal															
Planning															
Report writing															
Software implementation															
Circuit conduct															
Prototype design															

Table 3.1: Project Gantt chart for PSM 1

LIST OF TASKS FOR BDP 2	NUMBER OF WEEKS													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Purchasing Project of Components and BDP 2 Briefing								MID SEMESTER BREAK						
Equipment Testing and Circuit Testing														
Coding Development for IoT Implementation														
Circuit Designing and Hardware Planning														
Hardware Assembling and Project Testing														
Chapter 4 and Chapter 5														
Discussion With Supervisor														
Poster Development														
Submission of Report														

3.3.3 Software Implementation Flowchart

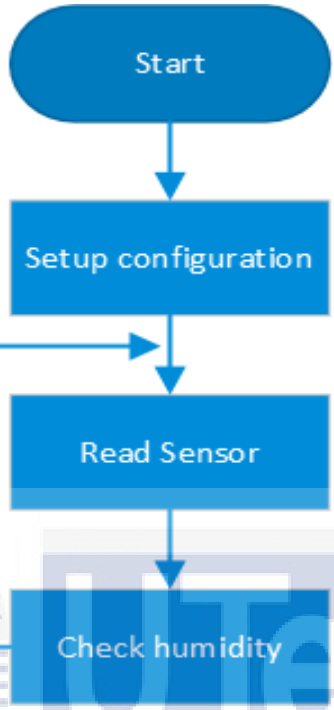


Figure 3.3: Software implementation flowchart

3.3.4 Hardware Implementation Flowchart

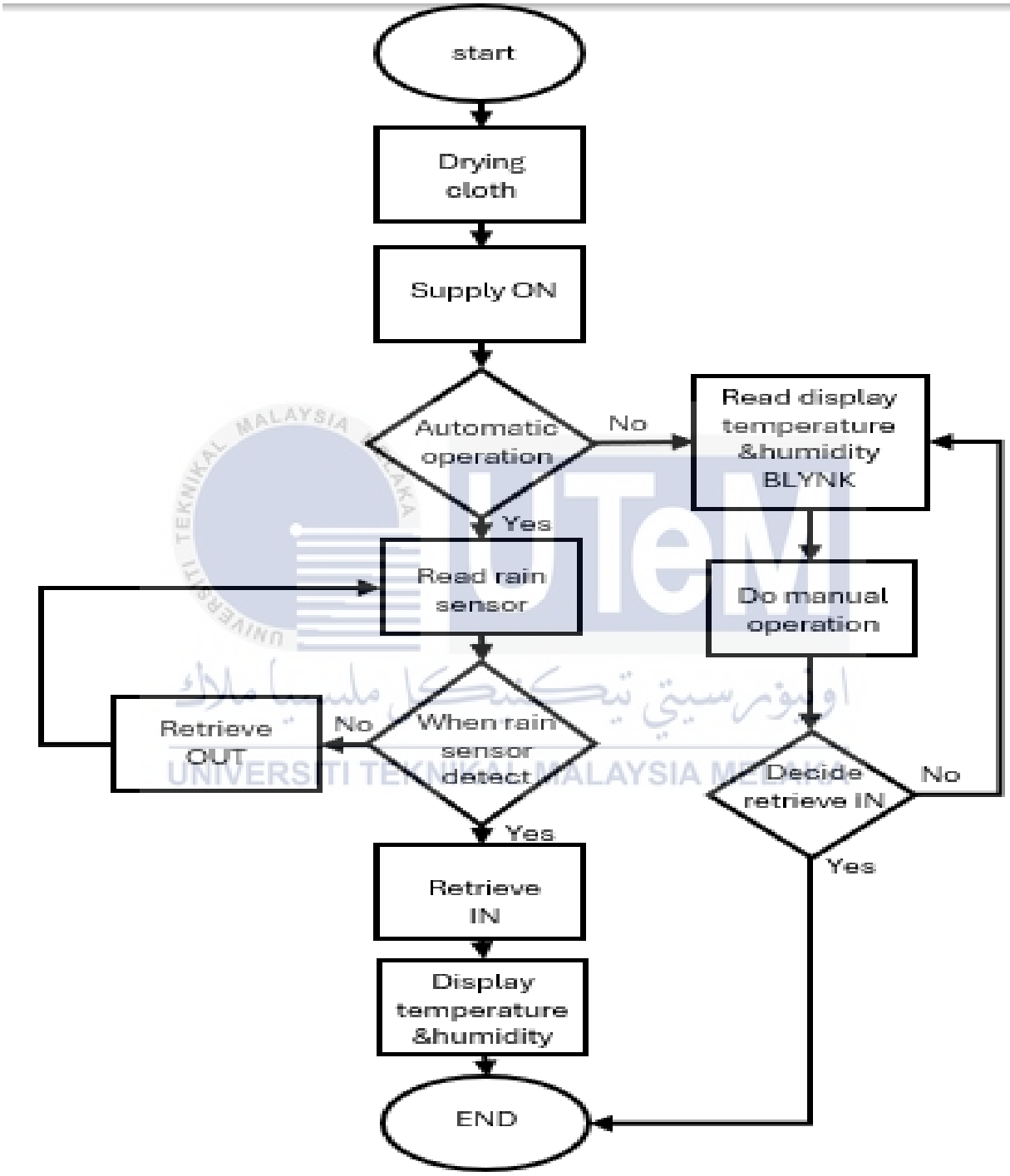


Figure 3.4: Hardware implementation flowchart

3.4 System Hardware Design

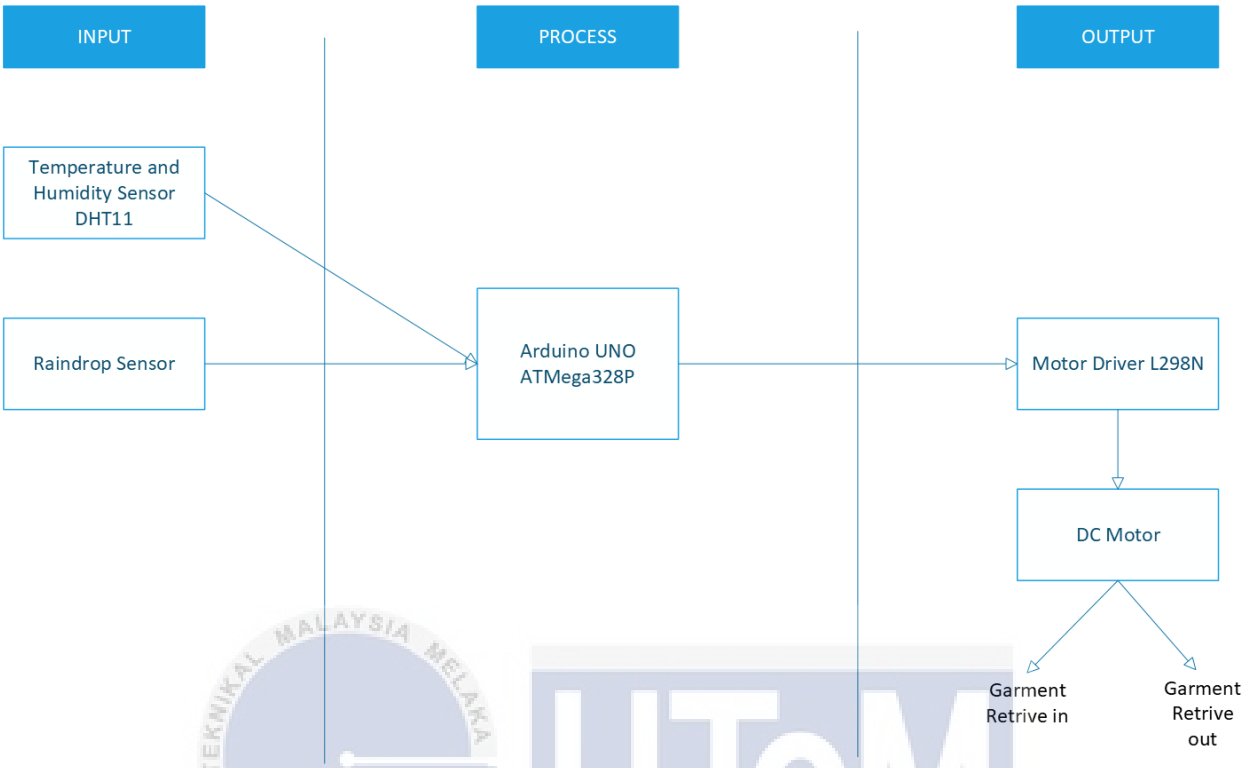


Figure 3.5: System hardware design

3.5 Hardware Requirement

a. Node microcontroller Unit

The NodeMCU (Hub MicroController Unit) is an open-source computer program and equipment improvement environment built around an cheap System-on-a-Chip (SoC) called the ESP8266. The ESP8266, outlined and fabricated by Espressif Frameworks, contains the pivotal components of a computer: CPU, Smash, organizing (WiFi), and indeed a advanced working framework and SDK. That creates it an amazing choice for Web of Things (IoT) ventures of all sorts.

The NodeMCU is accessible in different bundle styles. Common to all the plans is

the base ESP8266 center. Plans based on the engineering have kept up the standard 30-pin format. A few plans utilize the more common contract (0.9″) impression, whereas others utilize a wide (1.1″) impression – an critical thought to be mindful of.

Figure 3.6: NodeMCU ESP8266

Power pins - There are 4 power Pins. Vin pin and there 3.3V pins. The NodeMCU/ESP8266 board has a VIN pin that is used to power both the board and any peripherals connected to it. The onboard voltage regulator on the NodeMCU/ESP8266 module makes sure that the supplied voltage is regulated and stabilised for the proper operation of the board, and this power supply can be connected directly to the VIN pin. The NodeMCU/ESP8266 board's voltage needs can be met by connecting a regulated 5V power source to the VIN pin, which will also be regulated by the inbuilt regulator. The NodeMCU/ESP8266 board has 3.3V pins that can be used to power external components and provide an output from the voltage regulator included into the device. When connected to the NodeMCU/ESP8266 board, these pins provide a regulated 3.3V voltage that can be used to power a variety of external devices or components. As the internal voltage regulator maintains a constant voltage output for their functioning, using these 3.3V pins will guarantee a stable power source for your peripherals.

GND (Ground) – The ground of pins of NodeMCU.

I2C Pins - You may connect I2C sensors and peripherals to the NodeMCU/ESP8266 board via the I2C interface. Both I2C Master and I2C Slave modes are supported. Programming can be used to control I2C capabilities, and the maximum clock frequency is 100 kHz. For proper communication between the connected slave device and the connected master device, it is crucial to make sure that the I2C clock frequency is set higher than the slowest clock frequency supported by the slave device.

GPI0 Pins - The 17 GPIO pins on the NodeMCU/ESP8266 can be dynamically allocated to various I2C, I2S, UART, PWM, IR remote control, LED light, and button operations. Each GPIO with digital capability can be adjusted to high impedance, internal pull-up, or internal pull-down. It can also be set to edge-trigger or level-trigger when setup as an input to produce CPU interrupts.

ADC Channel - A 10-bit precision SAR ADC (Analog-to-Digital Converter) is built into the NodeMCU board. The VDD3P3 pin's power supply voltage as well as the TOUT pin's input voltage can both be measured using the ADC. It's crucial to remember that these two tasks cannot be carried out at the same time.

UART Pins - Two UART interfaces, UART0 and UART1, are available on the NodeMCU/ESP8266 board. These interfaces allow asynchronous communication protocols like RS232 and RS485. These interfaces are capable of 4.5 Mbps of communication speed. General communication can be done using the TXD0, RXD0, RST0, and CTS0 pins on UART0. The TXD1 pin of UART1, on the other hand, is mostly used for printing log information because it can only accept data transmitted through it.

SPI Pins - Two SPI (Serial Peripheral Interface) interfaces—SPI and HSPI--that can function as both slave and master devices are included in the NodeMCU/ESP8266 board. These SPI interfaces provide a number of practical features, such as:

- A versatile SPI format transfer using 4 timing mode.
- Support for divided clocks at 80 MHz and clock frequencies up to 80 MHz
- a 64-byte FIFO (First-In-First-Out) buffer that may hold data.

SDI0 Pins - An SDIO (Secure Digital Input/Output) interface on the NodeMCU/ESP8266 board enables direct connection with SD cards. With its support for 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0, SD cards and the board may transport data at fast speeds.

PWM Pins - Pulse Width Modulation (PWM) on the board's four channels allows for programmable control of digital motors and LEDs. Programming these PWM outputs will enable you to specify the required frequency in the range of 1000 s to 10000 s (equivalent to frequencies of 100 Hz and 1 kHz).

Control Pins - Specific pins on the NodeMCU/ESP8266 board are used to regulate it. These pins include the WAKE pin, the Chip Enable (EN) pin, and the Reset (RST) pin.

EN: To enable and make the ESP8266 chip fully functional, put the EN pin to the HIGH state. On the other hand, the chip enters a low-power mode when the EN pin is pulled LOW.

RST: The ESP8266 chip can be reset using the RST pin, enabling a new beginning or reinitialization.

WAKE: The chip is awakened from a deep sleep state by the WAKE pin, returning it to an active state for regular operation.

~ (PWM) - Control pins are used by the NodeMCU/ESP8266 board to govern its functionality. The Chip Enable (EN) pin, Reset (RST) pin, and Wake (WAKE) pin are made up of these pins.

EN: The ESP8266 chip is turned on by pulling the EN pin HIGH, while

pulling it LOW switches the chip to a low-power mode.

RST: The ESP8266 chip is reset using the RST pin.

WAKE: The WAKE pin brings the chip out of deep slumber and back into active mode.

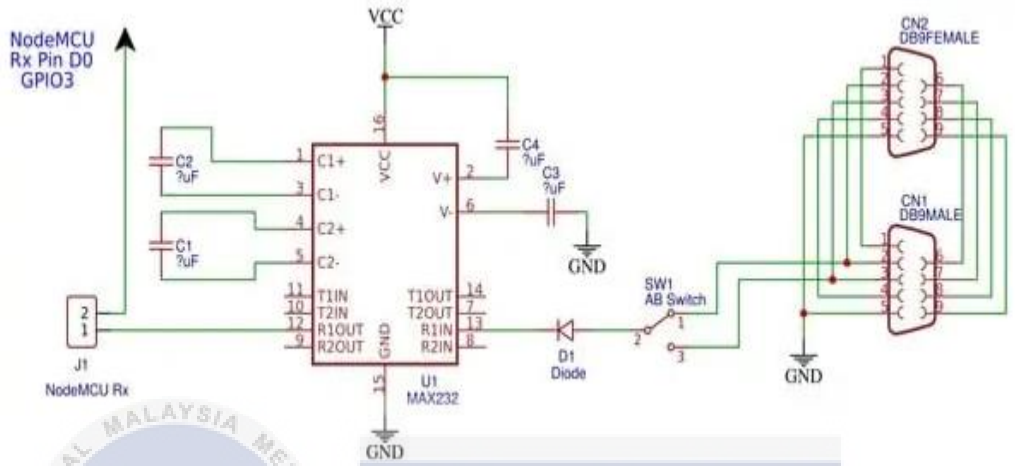
How USB to serial Converter change - There is a USB to Serial Converter built into every NodeMCU board. The CP2102 chipset is used in the suggested design, which is present in authentic and officially authorised Amica NodeMCU modules, providing maximum compatibility. The CH340G converter, however, is frequently used by less expensive modules like the LoLin units. Rare designs might make use of different converters, like the FTDI chipset. It's crucial to install the correct driver for the NodeMCU depending on the operating system. The CP2102 chipset is typically detected by Windows 10 immediately, however the CH340G can need a separate installation.

- The Silicon Labs support website has the CP2102 drivers available for download. To reduce potential problems, it's crucial to periodically update and install the most recent drivers in your development environment. Android, Linux, Mac, and Windows can all use the drivers. In addition, you can obtain a local copy of the CP2102 drivers (v10.1.8) from our website. To make sure you have the most recent versions of the driver, it is advised that you go to the website of the original manufacturer.

- WCH continuously updates and maintains the CH340G converter's drivers. The drivers may be downloaded from their Driver Download website and work with Windows, Mac, Linux, and Android. In addition, you can obtain a local copy of the

CH340G drivers (version 3.5). To make sure you have the most recent versions of the driver, it is advised that you go to the website of the original manufacturer.

NodeMCU Carrier Board Schematic



Two DB-09 connectors and a switch with the designation "SW1" can be seen on the NodeMCU Carrier Board schematic. The switch can be toggled between Pin 2 and Pin 3 to operate the level converter and is in charge of choosing the data source from either DB-09 connector.

b. Arduino UNO

A microcontroller board called Arduino UNO is based on the ATmega328P. It contains 6 analogue inputs, a 16 MHz ceramic resonator, 14 digital input/output pins (six of which can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. It comes with everything needed to support the microcontroller; to get started, just plug in a USB cable, an AC-to-DC adapter, or a battery. You can experiment with your UNO without being overly concerned that you'll make a mistake; in the worst case, you can replace the chip for a few dollars and start over.

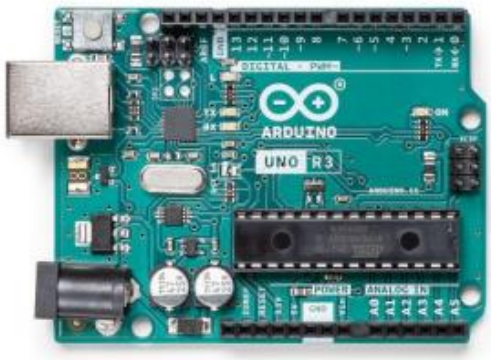


Figure 3.7: The top View of Arduino UNO R3

c. Temperature and Humidity Sensor DHT11

The DHT11 is a temperature and humidity sensor that is widely used. The senso includes a dedicated NTC to detect temperature and an 8-bit microprocessor to output temperature and humidity measurements as serial data. The sensor is also factory calibrated, making it simple to connect to other microcontrollers. The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1%.

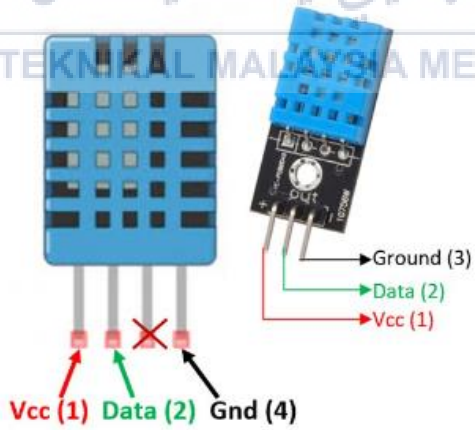


Figure 3.8: DHT11

The DHT11 Sensor produces serial data and is fully calibrated, making setup

incredibly simple. Below is a diagram of how this sensor is connected.

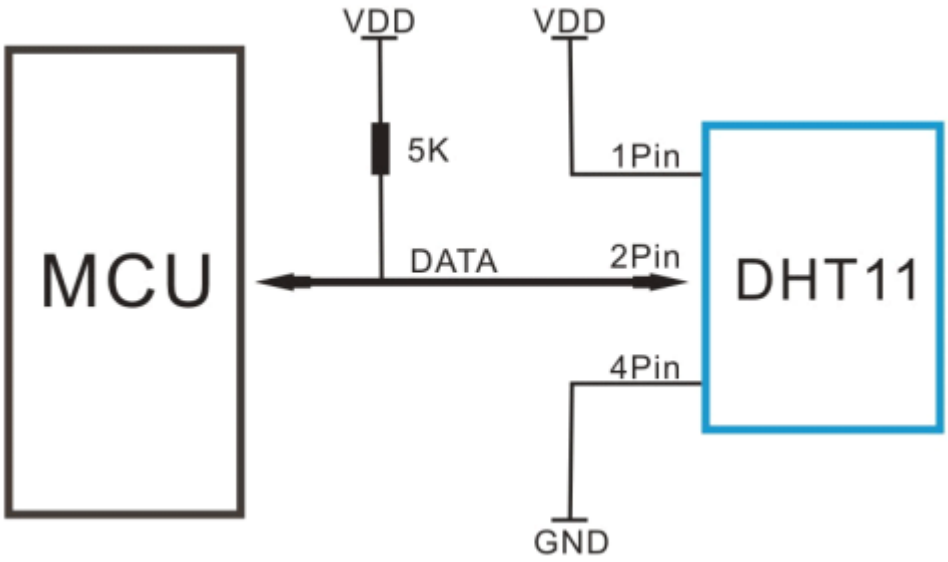


Figure 3.9 Block diagram for DHT11 between with NodeMCU.

As you can see, a 5K pull-up resistor is being utilised to connect the data pin to an MCU I/O pin. Temperature and humidity values are output as serial data on this data pin. There are pre-made libraries for it that will give you a head start if you're trying to interface DHT11 with Arduino.

The datasheet provided below will be useful if you're trying to interface it with another MCU. The data pin's output will be in the following order: 8 bits of humidity integer data, 8 bits of humidity decimal data, 8 bits of temperature integer data, 8 bits of temperature fractional data, and 8 bits of parity bit. The I/O pin must be briefly brought low and then held high as illustrated in the timing diagram below to instruct the DHT11 module to send these data.

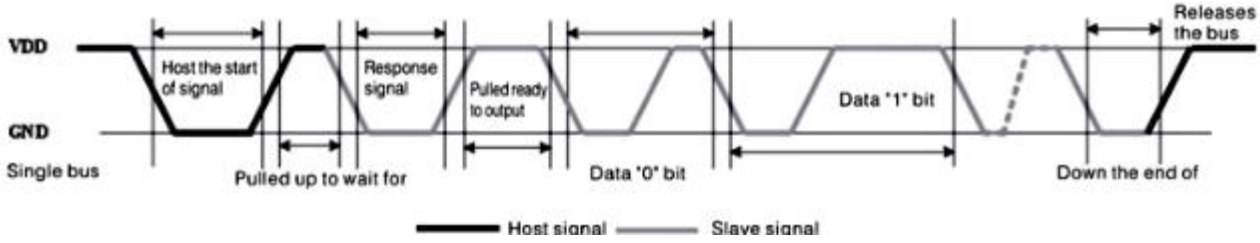


Figure 3.10 data pin input and output timing diagram.

d. Raindrop Sensor

As illustrated in the diagram below, the rain board module is linked to the raindrop sensor's control module. The raindrop sensor's control module has four outputs. VCC is linked to a 5V power supply. The module's GND pin is linked to the ground. For digital output, the DO pin is connected to the microcontroller's digital pin, alternatively the analogue pin can be used. To use the analogue output, connect the A0 pin to the ADC pin of a microcontroller. In the case of Arduino, it contains 6 ADC pins, which can be used directly without the use of an AD converter. A potentiometer, LN393 comparator, LFDs, capacitors, and resistors make up the sensor module. The components of the control module are depicted in the pinout image above. The rain board module is made up of copper tracks that function as variable resistors. Its resistance changes depending on how wet the rain board is.

Figure 3.11: Raindrop sensor

e. DC Motor GFM925W

We can control a DC motor by attaching an L298 bridge IC to an Arduino.

The most common form of motor is a direct current (DC) motor. Normally, DC motors have only two leads, one positive and one negative. The motor will rotate if these two lines are connected directly to a battery. If the leads are switched, the motor will rotate in the opposite direction. The DC motor will be utilized to retrieve the garment rack in this project.

Figure 3.12: DC motor

The auto gate motor has a DCMOTO GFM925W built-in hybrid power system that can save up to 95% on electricity costs, reducing energy use and CO2 emissions. Please feel free to browse our website's selection of DCMOTO GFM925W compatible auto gate accessories.

The built-in electromagnetic lock is a device that uses electricity to secure a door or gate. It has a hybrid power version that helps reduce CO^2 emissions by 95%. It also includes features like an auto close timer, an intruder alarm, and a

motion lock to enhance security. Additionally, it comes with an anti-spy wireless multifunction keypad and remote control for convenient operation. An optional safety beam sensor can be added for further safety measures. The main power is used to run the motor or charge the battery, while the backup battery consumes only 0.2 Watt when in standby mode. If the backup battery fails, the system will automatically switch to normal power mode and continue to function normally.

Specification:

- Electric motor - DC 12V
- Main power suppler - AC 230V (Hybrid Model)
- Backup battery - DV 12V 7AH
- Reduction gear with 4 stages planetary type
- Gear ration at 1:2310
- Output torque (Dynamic) - 2,000kg-cm
- Output torque (Static) - 3,240kg-cm
- Output speed (at full speed) - 1.6 r.p.m.

CALCULATION

RPM OF Rotation Motor

Revolutions Per Minute (RPM) is referred to as RPM in short. It measures the rate of revolution of the rotor component, or the number of complete rotations the rotor completes in a minute. On our website, the motor's spindle/motor shaft GFM925W rotational speed is measured using the RPM unit.

To calculate RPM for an AC induction motor, you multiply the frequency in Hertz (Hz) by 60.

- (Hz x 60 x 2) / number of poles = no-load RPM

To calculate the slip by subtracting the rated full load speed from the synchronous speed, dividing that answer by the synchronous speed and multiplying that answer by 100:

- ((synchronous speed-rated full-load speed) / (synchronous speed)) x 100 = slip rating

To find the full-load RPM, you convert the slip rating to RPM and then subtract it from the no-load RPM:

- To convert the slip rating to RPM: RPM x slip rating = RPM slip
- To calculate the full-load RPM: RPM – RPM slip = full-load RPM

Torque

Torque(τ)= 2 π.RPMPower(P).60

- τ is the torque in Newton-meters (Nm).
- *P* is the power in watts.
- Π is a mathematical constant approximately equal to 3.14159.
- RPM is the rotational speed in revolutions per minute.

The angular velocity (ω) is related to the rotational speed (RPM) by the equation: $\omega = 2\pi \times RPM/60$

f. Motor Driver

This L298N Dual H-Bridge Motor Driver Integrated Circuit Module is a high-power motor driver module that can power both DC and stepper motors. The L298N Module is capable of controlling up to four DC motors or two DC motors with directional and speed control up to 2A each in both directions. The L298N is a dual H-bridge motor driver that can regulate the speed and direction of two DC motors at the same time. The module can power DC motors with voltages from 5V to 35V and peak currents of up to 2 Ampere. It works great for robotic applications and connects easily to microcontrollers with only a few control lines needed for each motor. Relays, TTL logic gates, basic manual switches, and other devices can all be interfaced with it. This board has built-in protective diodes, a +5V regulator, and power LED indications.

Figure 3.13: L298N Motor Driver Module

Brief data:

• Input Voltage: 3.2V~40Vdc. Brief Data:

• Driver: L298N Dual H Bridge DC Motor Driver

• Power Supply: DC 5 V - 35 V

• Peak current: 2 Amp

• Operating current range: 0 ~ 36mA

• Control signal input voltage range :

• Low: -0.3V ≤ Vin ≤ 1.5V.

• High: 2.3V ≤ Vin ≤ Vss.

• Enable signal input voltage range :

o Low: -0.3 ≤ Vin ≤ 1.5V (control signal is invalid).

o High: 2.3V ≤ Vin ≤ Vss (control signal active).

• Maximum power consumption: 20W (when the temperature T = 75 ℃).

• Storage temperature: -25 ℃ ~ +130 ℃.

• On-board +5V regulated Output supply (supply to controller board i.e. Arduino).

• Size: 3.4cm x 4.3cm x 2.7cm

Schematic diagram

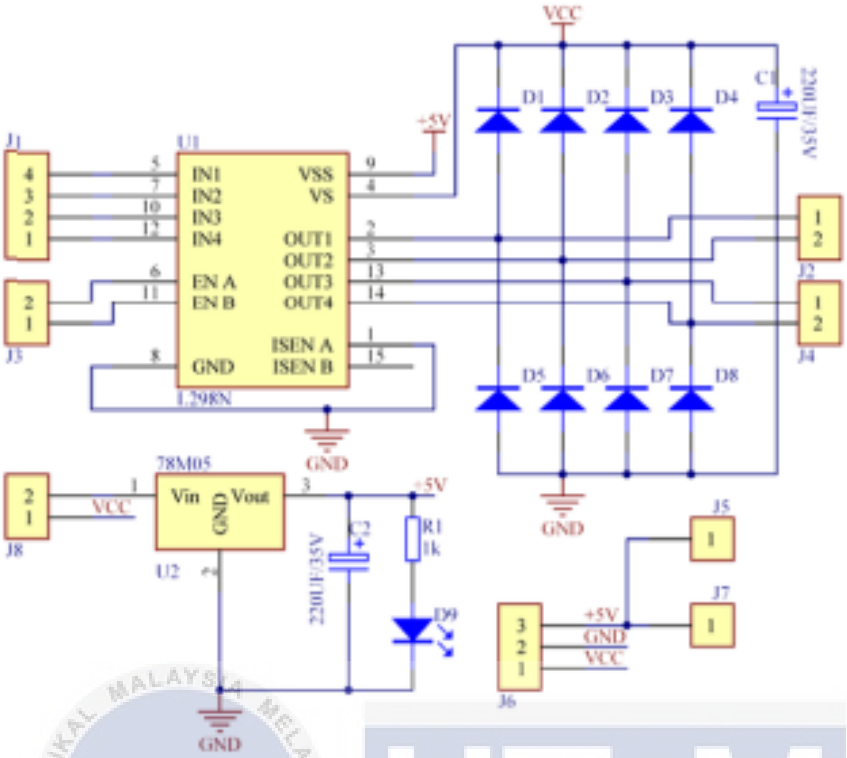


Figure 3.14 L298N Motor Driver Module Schematic Diagram

Board Dimension & Pins Label

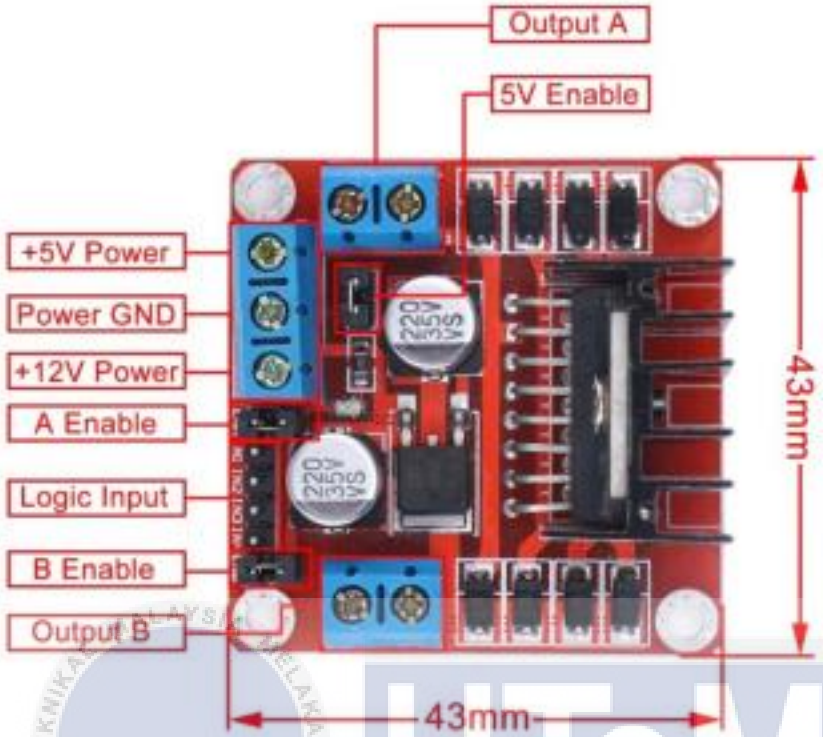


Figure 3.15: L298N Motor Driver Module Board Dimension & Pins label

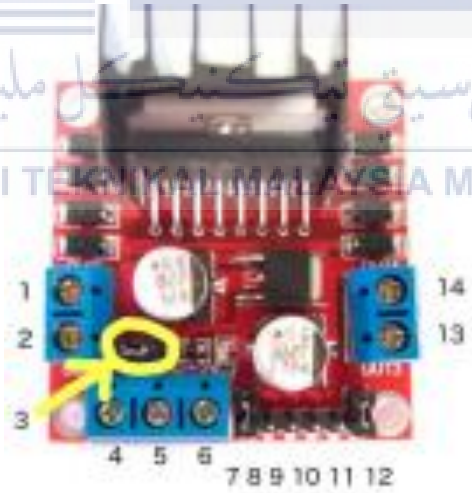


Figure 3.16: L29N Motor Driver for connection Pins

Connect the modules between 1,2,13 and 14 with the relevant A+, A-, B+ and B- cables from the stepper motor. Overlays the pairs at module points 7 and 12 with the jumpers provided with the L298N module. Points 4 (positive) and 5

(negative/ground) should then be connected to the power supply in the appropriate manner. Once more, if the power source for your stepper motor is than 12V, attach the jumper to the module at point 3 to create a clean 5V power supply for NodeMCU. Afterward, attach the digital pins D0, D1, D2, D3, D4 and D5 of the NOdeMCU to the corresponding pins IN1 of the LN98 module.

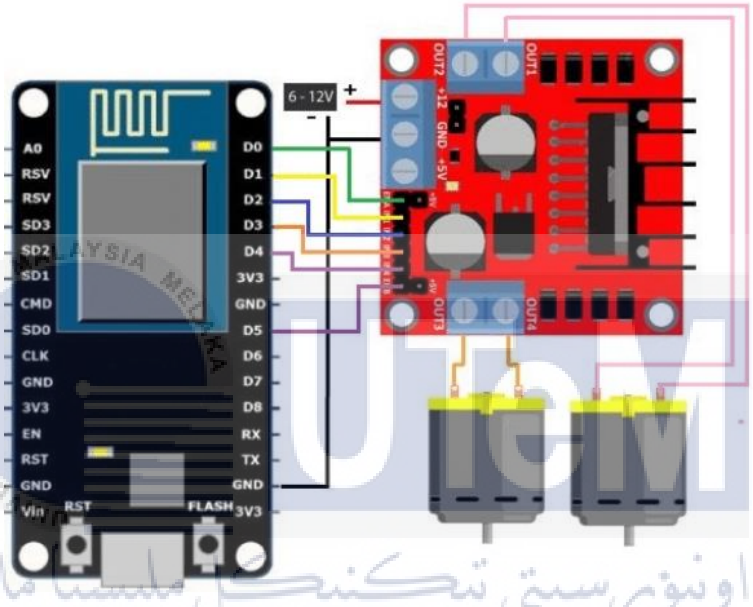


Figure 3.17 Interfacing of NodeMCU with L293n motor driver

g. DC Motor controller high current

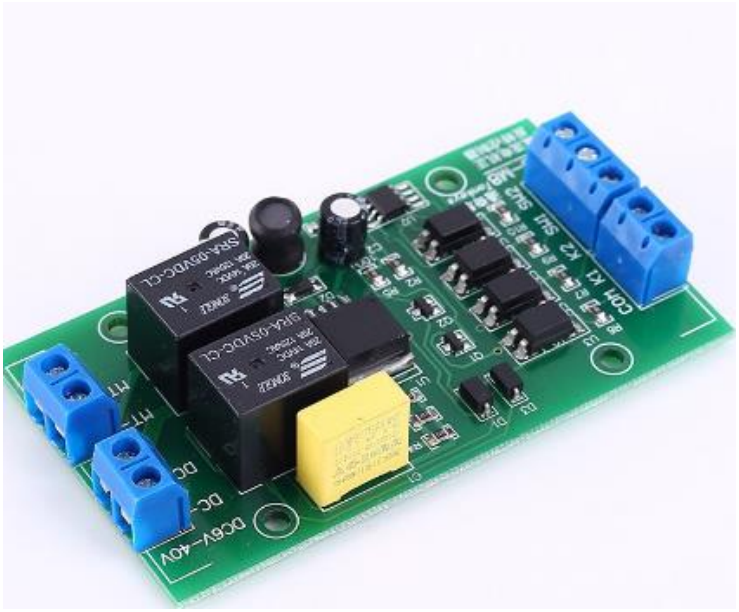


Figure 3.18 Design of the hardware DC Motor controller high current board.

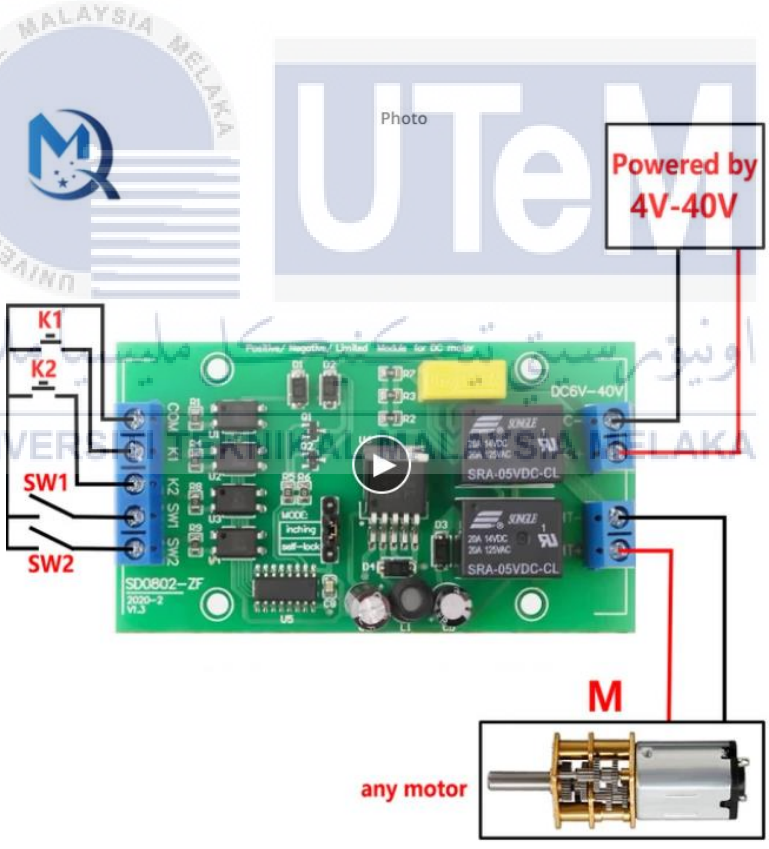


Figure 3.19 Interfacing of DC Motor controller high current control with Motor.

Motor forward and reverse controller is used to control motor work status to make

sure motor can be precisely controlled. Applicable to occasions where the motor control requirements are increased.

Features:

1.Low power consumption

2.Long work time

3.Automatic control

4.Support DC motor forward and reverse

5.Support Limit operation

6.Work output LED indicator

7.Support Self-locking Mode

Parameters:

1.Item Name:TKS-M8 DC Motor Driver Module

2.Model:TKS-M8

3.Work voltage:DC 4V-40V

4.Work mode:Trigger mode

5.Work Temperature:-40℃~85℃

6.Work Humidity:0%~95%RH

7.Size:83*46*17mm

Funciton:

1.Press button 'K1' motor forward rotation.Then stop if press button 'SW1'.Motor reverse rotation if press button 'K2'.

2.Press button 'K2' motor reverse rotation.Then stop if press button 'SW2'.Motor forward rotation if press button 'K1'.

Application:

1.Toy

2.Intelligent robot control

3.Smart car

4.Curtain

5.Mechanical control

h. Limit switch

Figure 3.20 Structure of Limit Switch

The electrical circuit that manages the machine's moving parts is then controlled by the limit switch. A DC motor can be started, stopped, slowed down, or speeded up using these switches as pilot devices for magnetic starter control circuits.

i. LCD Display

Figure 3.21 Equipment of LCD

These kinds of LCD displays are widely used in many electronics projects since they are excellent for displaying basic information, such as sensor data, and are also extremely affordable. In essence, the 204 LCD panel is a larger (more rows and columns) version. The display is ideal for displaying a huge amount of text without scrolling because it offers space for 20 columns of characters on 4 rows. The 58 pixel resolution of each column ensures that it may be seen well from a fair distance.

j. Jumper Wire

Male-to-male, male-to-female, and female-to-female jumper wires are the most common. The distinction between the two lies in the wire's terminating point. Male ends have a protruding pin and can plug into anything, but female ends do not and must be plugged into.Male-to-male jumper wires are the most popular and will be used the most. When connecting two ports on a

breadboard, a male-to-male wire will be required.

Figure 3.22: Jumper wires

3.6 Software Requirement

a. Arduino IDE

The Arduino IDE is an open-source program that makes it simple to update the source code on the board. It is compatible with Windows, Linux, and Mac OS. To make it operate properly while uploading the source code, the model and port number must be selected.

Figure 3.23: Arduino IDE

b. Blynk IoT application

Blynk are the device provisioning, sensor data visualisation, remote control via mobile and web applications, Over-The-Air firmware upgrades, secure cloud, data analytics, user and access management, alarms, and automations are all things you need to develop and manage connected gear. Low-volume producers of agricultural machinery, sophisticated HVAC systems, smart home devices, and everything in between are powered by the Blynk platform. With link the NodeMCU 8266 with the code are in IoT system.

Figure 3.24: Blynk application for IoT

3.7 Summary

Based on this chapter, the method used to run this project are explained step by step. The components used are also explained. This includes software and hardware used in this project. Next, the flowchart of this project are explained how the flow of the project are done, the software and hardware flowchart are drawn to see how each components work to complete the input and output of the project. By combining the power of Arduino Uno and NodeMCU, users can create a DIY smart home system that integrates IoT technology. The Arduino Uno handles sensor data acquisition and actuator control, while the NodeMCU enables remote connectivity and access. Together, they provide a flexible and cost-effective solution for building a smart home environment with automation, monitoring, and control capabilities.

By integrating these components, a smart garment rack can be created. The Arduino Uno can be programmed to receive commands from the NodeMCU board via serial communication or Wi-Fi. The NodeMCU board, connected to the home network or the internet, serves as the control interface for the smart garment rack.

The DC motor can be attached to the garment rack to enable automated movement. The Arduino Uno can control the motor's speed and direction based on commands received from the NodeMCU. For example, users can use a mobile application or a web interface to send commands to the NodeMCU, instructing the motor to rotate the garment rack clockwise or counterclockwise, or to stop its movement.

CHAPTER 4

RESULTS AND DISCUSSION

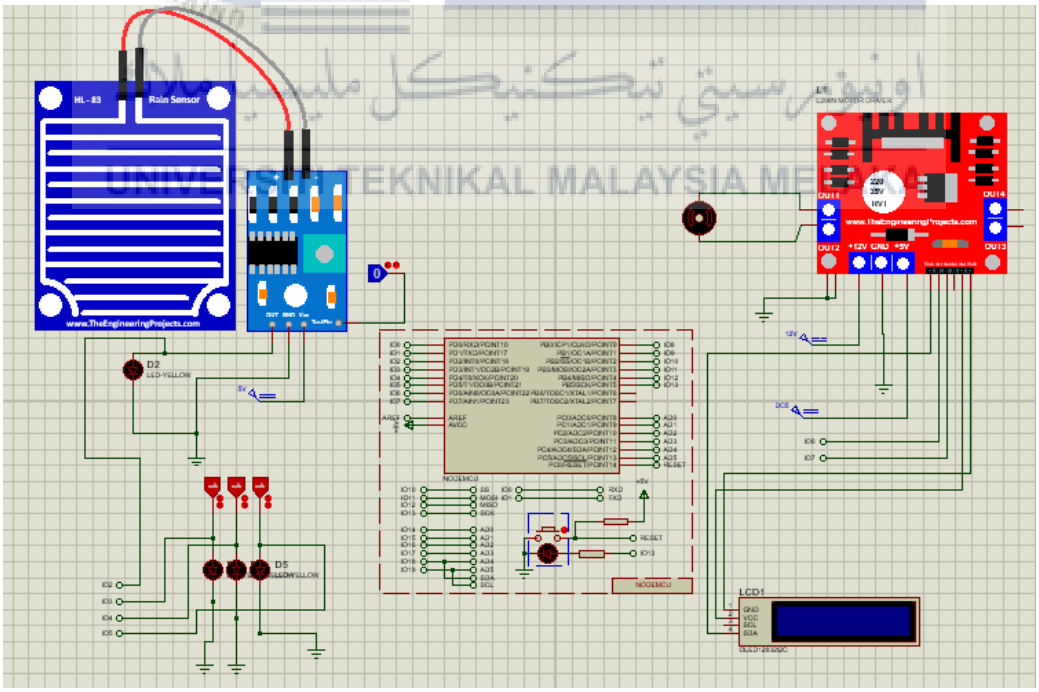
4.1 Introduction

This chapter presents the results and analysis on the development of automated garment rack. This system results are tested based on the expected output data according to the used method. The final outcomes of these reviews and evaluations are used as a guide to see if the project is a accomplishing the project objective.

4.2 Result and Analysis

4.2.1 Schematic Design

Schematic diagram is used to design the hardware and components connection the circuit by using simulation software such as Proteus. Below is the schematic layout for this project.



The figure 4.1 Schematic design of circuit

4.2.2 Project configuration environment setup

All the components relate to jumper wires and 9 volt of external power source to move the DC motor. Below are the table of details of the pins number that connected to NodeMCU.

Hardware	**Wire**	**Pins**
Raindrop sensor	Digital output	D8
	VCC	5V
	Ground	Ground
Temperature and humidity sensor	VCC	5V
	Signal	D7
	Ground	Ground
Motor driver	ENB	D3
	IN1	D4
	IN2	D5
	12V	9V Battery
	Ground	Ground
	5V	LOAD
DC motor	Positive	OUT1
	Negative	OUT2
LCD 16x2 I2C	Power	5V
	Ground	Ground
	SDA	A4
	SCL	A5

Table 4.1: Details of pin number

4.2.3 Data Analysis form result

Analyzing data from a garment rack can provide valuable insights into various aspects of motor rotation and temperature result:

DAY 1 afternoon with mid heat

Load (kg)	RPM motor	Humidity	Temperature
2.2	68	73	33
2.0	69	74	33
1.8	69	74	32
1.6	69	74	32
1.4	69	73	33
1.2	69	72	33

Table 4.2: Table of result rpm and temperature day 1

DAY 2 afternoon with heavy rain

Load (kg)	RPM motor	Humidity	Temperature
2.3	68	75	29
2.2	68	74	30
2.1	69	74	30
1.9	69	76	30
1.8	69	76	31
1.7	69	76	30

Table 4.3: Table of result rpm and temperature day 2

DAY 3 morning with heavy rain

Load (kg)	RPM motor	Humidity	Temperature
2.3	68	75	29
2.2	68	75	29
2.1	69	75	29
1.9	69	77	28
1.8	69	78	29
1.7	69	77	29

Table 4.4: Table of result rpm and temperature day 3

DAY 4 afternoon with very hot

Load (kg)	RPM motor	Humidity	Temperature
2.2	68	73	32
2.0	69	71	32
1.8	69	73	32
1.6	69	70	33
1.4	69	70	33
1.2	70	71	34

Table 4.5: Table of result rpm and temperature day 4

Comparison of between heat day dry and rainy day

Heat day					Rainy day				
Load (kg)	**RPM motor**	**Torque(τ) (Nm)**	**Angular velocity in radians 1/s (ω)**	**Power (W)**	**Load(kg)**	**RPM motor**	**Torque(τ) (Nm)**	**Angular velocity in radians 1/s (ω)**	**Power(W)**
2.2	68	1.854	7.121	13.2	2.3	67	1.881	7.016	13.2
2.0	69	1.827	7.226	13.2	2.2	67	1.881	7.016	13.2
1.8	69	1.827	7.226	13.2	2.1	67	1.881	7.016	13.2
1.6	69	1.827	7.226	13.2	1.9	68	1.854	7.121	13.2
1.4	69	1.827	7.226	13.2	1.8	68	1.854	7.121	13.2
1.2	70	1.800	7.330	13.2	1.7	68	1.854	7.121	13.2

Table 4.6: Table of comparison heat day and rainy day

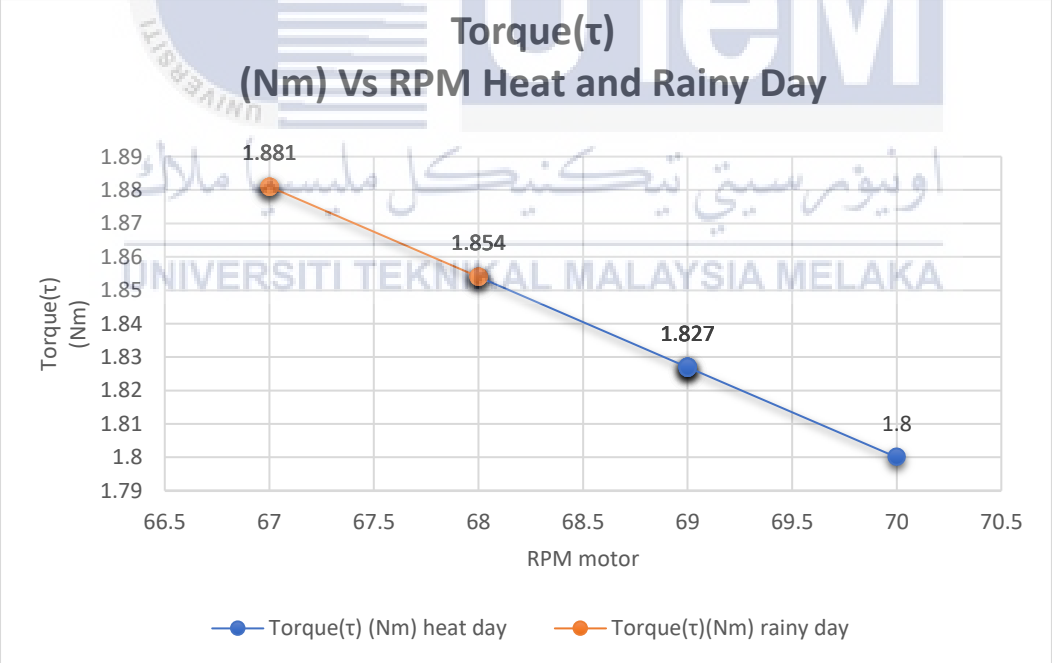


Figure 4.2 graph between toque and RPM motor in heat day and rainy day.

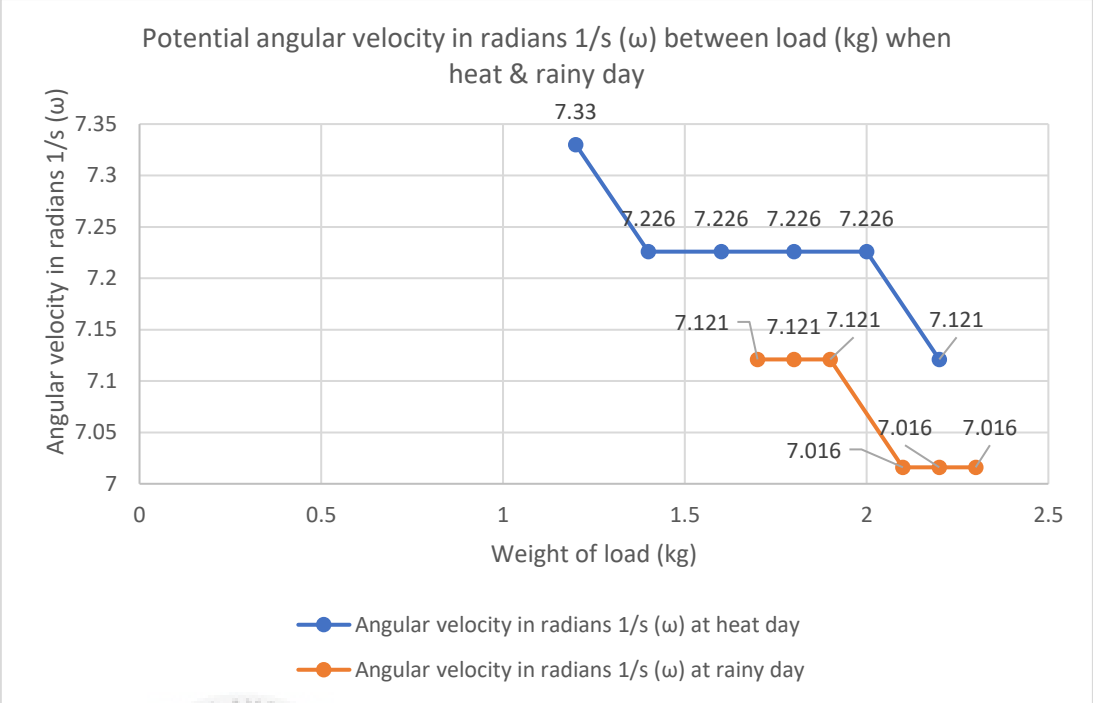


Figure 4.3 Graph show Potential Angular velocity in radians 1/s (ω) between load(kg) when heat day and rainy day.

4.2.4 Project Configuration Environment Setup

The system is configured using the Arduino IDE software. To configure the system, connect the Arduino UNO to the USB port. Arduino code is written in C++. Before compiling the code. the DHTll library was imported to the software to easily access the function of temperature and humidity sensor. DHT11 library is imported to the software to allow the functionality of temperature and humidity sensor.

```
#define BLYNK_TEMPLATE_ID "TMPL6GN4LvLja"
#define BLYNK_TEMPLATE_NAME "Automatic Cloth Dryer"
#define BLYNK_AUTH_TOKEN "sD8oliMdNpX4eYEEeOGvuhlBHUiNqwcd"
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

/* Fill-in your Template ID (only if using Blynk.Cloud) */
//#define BLYNK_TEMPLATE_ID   "YourTemplateID"


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "DHT.h"
#define DHTPIN D7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

Figure 4.4: Include DHT11 library in NodeMCU.

Next, the pin connected to the Arduino are defined in the code. All the pins number are defined referring to table 4.1.

```
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "wifitest";
char pass[] = "wifitest";

int rain_sensor1;
int Auto_Manual;
int Motor_on_off;
boolean state = 0;
float h, t,f;
```


Figure 4.5: Defining the pins number

dht.readHumidity() function is to read data from the temperature and humidity sensor. The values were displayed on the serial monitor according to the sensor. The system will determine the DC motor direction.

```
void dht_11(){

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
```

Figure 4.6: dht.readHumidity()

```
COM4

Humidity: 71.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 71.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 632  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 633  PWM = 157
Digital value : DRY
Humidity: 70.00%  Temperature: 31.30C Pot value = 632  PWM = 157
```


Figure 4.7: Serial monitor

4.3 Project Testing

This section detailed how to assess and evaluate systems in order to attain the system's final goal by involving the testing process. The testing phase is separated

into several sections: test plan for preparation, test design for the testing process, and test result and analvsis for process satisfaction.

Test functionality	Test connectivity of raindrop sensor with NodeMCU.
Precondition	• Raindrop sensor setup • Arduino IDE setup
Execution steps	1. Connect raindrop sensor to NodeMCU 2. Upload scrip in Arduino IDE 3. Open serial monitor on Arduino IDE
Expected result	DHT11 sensor sends input to Arduino and display on serial monitor
Error message	N/A
Result	Pass

Table 4.7: Connectivity test of raindrop sensor

Test functionality	Test connectivity of DC motor with motor driver and NodeMCU
Precondition	• Motor driver setup • 9V external power source setup • DC motor setup
Execution steps	1. Connect motor driver to Arduino 2. Connect DC motor to motor driver 3. Open serial monitor on Arduino IDE
Expected result	DC motor moves
Error message	N/A
Result	Pass

Table 4.8: Connectivity test of DC motor

4.4 Data Testing

In this section, the validation of connectivity status data from the input will be implemented.

4.4.1 Arduino connectivity test

The Arduino microcontroller are connected to the laptop using USB cable. The USB port are set in the Arduino IDE.

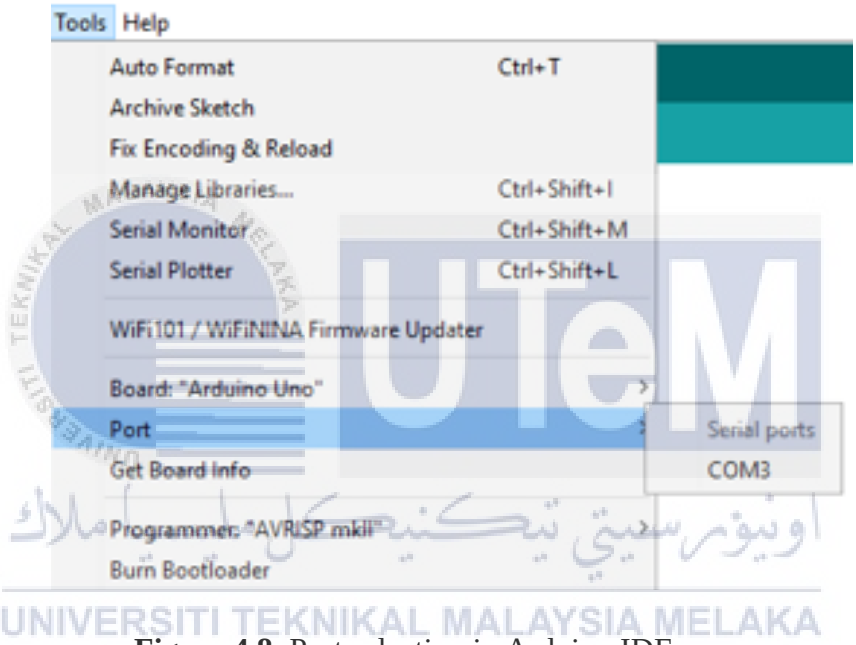


Figure 4.8: Port selection in Arduino IDE

4.4.2 DHT11 connectivity test

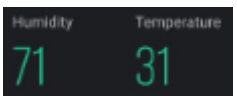


Figure 4.9: Data from DHT11 sensor

4.5 Test result and analysis

4.5.1 Wet clothes on sunny heat day control by automatic rain sensor

The components are set up and connected to the power supply. The data show on serial monitor are observed and the garment retrieved out.

Figure 4.10: No rain detected

4.5.2 Wet clothes on rainy day control by automatic rain sensor

A few drops of water were sprinkled on the raindrop sensor. The data are shown in the serial monitor show that the raindrop sensor is wet and the garment retrieved in.

Figure 4.11: Rain detected.

4.5.3 Wet clothes on sunny heat day control by IOT system.

Controlling wet clothes drying using an IoT (Internet of Things) system on a sunny day involves temperature, humidity, integrating sensors, communication devices, and actuators to automate the process retrieve out.

Figure 4.12: Manual retrieve out using IOT system (Blynk).

4.5.4 Wet clothes on rainy day control by IOT system.

Controlling wet clothes drying using an IoT system on a sunny day involves integrating sensors, communication devices, and actuators to automate the process retrieve in.

Figure 4.13: Manual retrieve in using IOT system (Blynk).

CHAPTER 5

CONCLUSION & FUTURE WORK

5.1 Conclusion

This project will aim to enhance the functionality of a garment rack by integrating it with IoT technology, the ESP8266 microcontroller, and a rain sensor. The project's objective is to create a smart garment rack that can not only automate garment management and it will also protect garments from rain.

The project will use builds upon the foundation of the IoT-based automated garment rack with ESP8266, as described in the previous conclusion. In addition to the existing features, this project incorporates a rain sensor into the system.

The rain sensor will design to detect the presence of rain or moisture in the environment. It is integrated with the garment rack, allowing it to monitor the weather conditions in real-time. When the rain sensor detects rain, it sends a signal to the ESP8266 microcontroller, triggering a response from the garment rack.

Upon receiving the signal, the garment rack will initiate a protective action. For example, it can automatically cover the garments with a waterproof or water-resistant material, ensuring that they remain dry and unaffected by the rain. This feature is especially useful in scenarios where the garment rack is placed outdoors or in areas prone to rain or high humidity.

The IoT integration enables users will be able to remotely monitor the status of the garment rack, including the rain sensor readings. Users also will access this information through a companion mobile application. In addition, they can receive notifications or alerts when the rain sensor detects rain, allowing them to take appropriate actions or make decisions regarding the garments.

By combining the IoT technology, the ESP8266 microcontroller, and the rain sensor, this project creates an innovative solution for garment management and protection.

The automated garment rack not only streamlines the organization and accessibility of garments but also safeguards them from unfavourable weather conditions. This project has the potential to find applications in various settings, such as homes, retail stores, or outdoor events, where garment protection from rain or moisture is crucial.

5.2 Future work

For future work, the project can be enhanced as follow:

I. Send notification can control manually if the weather is rain to mobile phone.
II. Can be energy efficiency of residents.
III. From weather and humidity surrounding can control of movement of garment rack.

REFERENCES

[1] J. Cox and H. Dittmar, "The functions of clothes and clothing (dis)satisfaction: A gender analysis among British students," *Journal of Consumer Policy*, vol. 18, no. 2–3, pp. 237–265, 2018. doi:10.1007/bf01016513

[2] O. Uzelac, "Basic characteristics of insurance of designer clothes and accessories," *Tokovi osiguranja*, vol. 37, no. 4, pp. 71–91, 2021. doi:10.5937/tokosig2103071u

[3] Z. M. Yusoff *et al.*, "Smart clothline system based on internet of thing (IOT)," *MATEC Web of Conferences*, vol. 248, p. 02002, 2018. doi:10.1051/matecconf/201824802002

[4] M. R. Asyraf, M. R. Ishak, M. R. Razman, and M. Chandrasekar, "Fundamentals of creep, testing methods and development of test rig for the full-scale crossarm: A Review," *Jurnal Teknologi*, vol. 81, no. 4, 2019. doi:10.11113/jt.v81.13402

[5] M. L. Tan, N. Samat, N. W. Chan, A. J. Lee, and C. Li, "Analysis of precipitation and temperature extremes over the Muda River basin, Malaysia," *Water*, vol. 11, no. 2, p. 283, 2019. doi:10.3390/w11020283

[6] C. A. James, A. G. B. Willmott, A. Dhawan, C. Stewart, and O. R. Gibson, "Increased air temperature decreases high-speed, but not total distance, in international field hockey," *Temperature*, vol. 9, no. 4, pp. 357–372, 2021. doi:10.1080/23328940.2021.1997535

[7] S. Ameer, J. Benson, and R. Sandhu, "An attribute-based approach toward a secured smart-home IOT access control and a comparison with a role-based approach," *Information*, vol. 13, no. 2, p. 60, 2022. doi:10.3390/info13020060

[8] H. Luo *et al.*, "G2F: A secure user authentication for rapid smart home iot management," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10884–10895, 2021. doi:10.1109/jiot.2021.3050710

[9] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Benefits and risks of Smart Home Technologies," *Energy Policy*, vol. 103, pp. 72–83, 2017. doi:10.1016/j.enpol.2016.12.047

[10] D. Marikyan, S. Papagiannidis, and E. Alamanos, "A systematic review of the Smart Home Literature: A user perspective," *Technological Forecasting and Social Change*, vol. 138, pp. 139–154, 2019. doi:10.1016/j.techfore.2018.08.015

[11] A. Johari and R. Alsaqour, "Blockchain-based model for Smart Home Network Security," *International Journal of Computer Networks and Applications*, vol. 9, no. 4, p. 497, 2022. doi:10.22247/ijcna/2022/214509

[12] M. Naing and N. N. Hlaing, “Arduino based Smart Home Automation System,” *International Journal of Trend in Scientific Research and Development*, vol. Volume-3, no. Issue-4, pp. 276–280, 2019. doi:10.31142/ijtsrd23719

[13] Daniel Trento, Ticiana Patel Weiss Trento, and Eduardo Krüger, “Application of Arduino-based systems as monitoring tools in Indoor Comfort Studies: A Bibliometric analysis,” *International Journal of Architectural Engineering Technology*, vol. 7, pp. 1–12, 2020. doi:10.15377/2409-9821.2020.07.1

[14] S.-M. Kim, Y. Choi, and J. Suh, “Applications of the open-source hardware Arduino platform in the mining industry: A Review,” *Applied Sciences*, vol. 10, no. 14, p. 5018, 2020. doi:10.3390/app10145018

[15] A. Singh, “Development of an Arduino-based embedded system,” *SSRN Electronic Journal*, 2019. doi:10.2139/ssrn.3413434

[16] D. B. hertanto, R. Asnawi, F. Surwi, and N. Setiawan, “Prototype development of distance detection system based on the internet of things using ESP 8266 WIFI Nodemcu module,” *Journal of Physics: Conference Series*, vol. 2111, no. 1, p. 012049, 2021. doi:10.1088/1742-6596/2111/1/012049

[17] “Application of Arduino devices in various IOT application,” *Renewable and Nonrenewable Energy*, pp. 39–45, 2022. doi:10.46632/rne/1/1/7

[18] P. Hatta, D. F. Dadungawigra, and A. Budianto, “Design and implementation of microcontroller-based building automation for smart computer laboratory,” *IJIE (Indonesian Journal of Informatics Education)*, vol. 3, no. 2, 2020. doi:10.20961/ijie.v3i1.27304

[19] T. Banu, “Two way control ESP8266 node MCU based robotic vehicle,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 7, pp. 730–732, 2020. doi:10.22214/ijraset.2020.30338

[20] G. Li, Y. Wu, and P. Xu, “An observer-based fixed-time position tracking strategy for DC Torque Motor Systems,” *Mechanical Systems and Signal Processing*, vol. 142, p. 106774, 2020. doi:10.1016/j.ymssp.2020.106774

[21] M. F. Abdul Ghani, M. A. Mohd Yusoff, H. L. Wong, S. C. Yip, and C. K. Tan, “A prepaid energy billing system with credit shortage forecasting,” *Lecture Notes in Electrical Engineering*, pp. 299–311, 2022. doi:10.1007/978-981-16-8515-6_24

APPENDICS

Code for Arduino Uno

```
//NODEMCU
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(8, 7);
#define TX 7 //D2
#define RX 8 //D1

//LCD
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

//SWITCH
#define SW1 12
#define SW2 13
int SW1_state;
int SW2_state;
//MOTOR DRIVER
#define ENA 11
#define IN1 10
#define IN2 9

//TEMPERATURE SENSOR
#include "DHT.h"
#define DHTPIN 6
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
float h, t,f;

//RAIN SENSOR
#define RS1 2
#define RS2 3
#define RS3 4
#define RS4 5

int RS1_state;
int RS2_state;
int RS3_state;
int RS4_state;

void setup() {
Serial.begin(9600);
dht.begin();
lcd.begin();
lcd.backlight();

mySerial.begin(4800);

pinMode(RS1 , INPUT);
pinMode(RS2 , INPUT);
pinMode(RS3 , INPUT);
pinMode(RS4 , INPUT);
pinMode(ENA, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
```

```
pinMode(SW1, INPUT);
pinMode(SW2,INPUT);

}

void loop() {

//SWITCH
SW1_state = digitalRead(SW1);
SW2_state = digitalRead(SW2);
Serial.print("SW1:");
Serial.print(SW1_state);
Serial.print(" SW2:");
Serial.println(SW2_state);

lcd.setCursor(6,1);
lcd.print(SW1_state);
lcd.setCursor(7,1);
lcd.print(SW2_state);

//TEMPERATURE SENSOR
h = dht.readHumidity();

t = dht.readTemperature();

f = dht.readTemperature(true);

if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println(F("Failed to read from DHT sensor!"));
```

```
  return;
}

float hif = dht.computeHeatIndex(f, h);
float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));
Serial.print(h);
Serial.print(F("%  Temperature: "));
Serial.print(t);
Serial.print(F("°C "));
Serial.print(f);
Serial.print(F("°F  Heat index: "));
Serial.print(hic);
Serial.print(F("°C "));
Serial.print(hif);
Serial.println(F("°F"));

lcd.setCursor(0,0);
lcd.print("H:");
lcd.setCursor(2,0);
lcd.print(h);
lcd.setCursor(8,0);
lcd.print("T:");
lcd.setCursor(10,0);
lcd.print(t);

//RAIN SENSOR
RS1_state = digitalRead(RS1);
```

```
RS2_state = digitalRead(RS2);
RS3_state = digitalRead(RS3);
RS4_state = digitalRead(RS4);

lcd.setCursor(0,1);
lcd.print(RS1_state);
lcd.setCursor(1,1);
lcd.print(RS2_state);
lcd.setCursor(2,1);
lcd.print(RS3_state);
lcd.setCursor(3,1);
lcd.print(RS4_state);

Serial.print("RS1:");
Serial.print(RS1_state);
Serial.print(" RS2:");
Serial.print(RS2_state);
Serial.print(" RS3:");
Serial.print(RS3_state);
Serial.print(" RS4:");
Serial.println(RS4_state);

//MOTOR DRIVER
analogWrite(ENA, 80);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);

tx();
```

```
delay(1000);
}

void tx(){
    StaticJsonDocument<200> doc;

    doc["Humidity"] = h;
    doc["Temperature"] = t;
    doc["RS1"] = RS1_state;
    doc["RS2"] = RS2_state;
    doc["RS3"] = RS3_state;
    doc["RS4"] = RS4_state;
    doc["SW1"] = SW1_state;
    doc["SW2"] = SW2_state;


   serializeJson(doc, mySerial);
}
```

Code for ESP82866 wifi Module

```
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID   "TMPL6GN4LvLja"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_TEMPLATE_ID "TMPL6GN4LvLja"
#define BLYNK_TEMPLATE_NAME "Automatic Cloth Dryer"
#define BLYNK_AUTH_TOKEN "sD8oliMdNpX4eYEEeOGvuh1BHUiNqwcd"

// Your WiFi credentials.
char auth[] = "sD8oliMdNpX4eYEEeOGvuh1BHUiNqwcd";
// Set password to "" for open networks.
char ssid[] = "Faris_2.5GHz";
char pass[] = "jalandi5";
```



```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(D2, D1); // RX, TX

unsigned long previousMillis = 0;
int State = 0;

int h , t , RS1_state, RS2_state ,RS3_state, RS4_state , SW1_state , SW2_state;
int state=0;
```

```
int state_mode = 0;
int Mot_state = 0;

int value1 = 2222;
int value2 = 1123;

void setup() {

  Serial.begin(57600);
   mySerial.begin(4800);
   Blynk.begin(auth, ssid, pass);
}

void loop() {

   Blynk.run();

  unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= 1000) {

    previousMillis = currentMillis;
    tx();
  }
  rx();

  state =  RS1_state*RS2_state*RS3_state*RS4_state;
  Blynk.virtualWrite(V4, state);
```

```
}

BLYNK_WRITE(V5){
  state_mode = param.asInt();
  Serial.println(state_mode);
}
BLYNK_WRITE(V6){
  Mot_state = param.asInt();
  Serial.println(Mot_state);
}

void tx(){
    StaticJsonDocument<200> doc1;

    doc1["state_mode"] = state_mode;
    doc1["Mot_state"] = Mot_state;

  serializeJson(doc1, mySerial);
  Serial.print("State :");Serial.print(state_mode);Serial.print("
Mot:");Serial.println(Mot_state);
}

void rx(){
      if (mySerial.available())
  {
    StaticJsonDocument<200> doc;

    DeserializationError err = deserializeJson(doc, mySerial);
```

```
if (err == DeserializationError::Ok)
{

  h = doc["Humidity"].as<int>();
  t = doc["Temperature"].as<int>();
  RS1_state = doc["RS1"].as<int>();
  RS2_state = doc["RS2"].as<int>();
  RS3_state = doc["RS3"].as<int>();
  RS4_state = doc["RS4"].as<int>();
  SW1_state = doc["SW1"].as<int>();
  SW2_state = doc["SW2"].as<int>();

  Serial.print("H:");Serial.println(h);
  Serial.print("T:");Serial.println(t);
  Serial.print("RS1:");Serial.println(RS1_state);
  Serial.print("RS2:");Serial.println(RS2_state);
  Serial.print("RS3:");Serial.println(RS3_state);
  Serial.print("RS4:");Serial.println(RS4_state);
  Serial.print("SW1:");Serial.println(SW1_state);
  Serial.print("SW2:");Serial.println(SW2_state);

  Blynk.virtualWrite(V2,t );
  Blynk.virtualWrite(V3,h );

}
else
{
  Serial.print("deserializeJson() returned ");
```

```
      Serial.println(err.c_str());

      while (mySerial.available() > 0)
        mySerial.read();
    }
  }
}
```