T.C.

ISTANBUL ALTİNBAS UNİVERSİTY

(Graduate School of Science and Engineering) (Electrical and Computer Engineering)

WIRELESS SENSOR NETWORK (WSN) PREDICTABLE WEATHER STATION

Laith Mohammed Salim

Thesis

Dr. Sefer Kurnaz

Istanbul, (2019)

WIRELESS SENSOR NETWORK (WSN) PREDICTABLE WEATHER STATION

By

Laith Mohammed Salim

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

In partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2019

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sefer KURNAZ

Supervisor

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Prof. Dr. Osman N. UCAN	School of Engineering and Natural Scince, Altinbaş University	
Asst. Prof. Dr. Sefer KURNAZ	School of Engineering and Natural Science, Altinbaş University	
Asst. Prof. Dr. Zeynep ALTAN	Engineering and Architecture Faculty, Beykent University	

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Çağatay Aydın

Head of Department

Approval Date of Graduate School of Science and Engineering: 13/03/2019

Assoc. Prof. Dr. Oguz Bayat

Director

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Laith Mohammed Salim

DEDICATION

I would like to dedicate this work to my lovely family, for their invaluable efforts when I felt hopeless and weak in solving problems.

ACKNOWLEDGMENTS

I wish to express my acknowledgments to my supervisor, Asst. Prof. Dr. Sefer Kurnaz who was abundantly helpful and offered invaluable support with his sincerity and belief in me.

ABSTRACT

WIRELESS SENSOR NETWORK (WSN) PREDICTABLE WEATHER STATION

Salim, Laith Mohammed

M.S., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Dr. Sefer kurnaz

Co-Supervisor:

Date: 3/2019

Pages: 72

The object of this research is to build a wireless weather station that able to make a decision about playing in the right weather condıtıons. The weather station is designed to collect data from the environment. There are four sensors used to collect the required data. These sensors are temperature and humidity sensor, Light sensor, Rain detection sensor and dc motor as a wind sensor. The weather station contains two part. The first part placed outdoor for collecting data and sending them with nrf24l01 transceiver module as a sender. The second part placed indoor to receiving data from the outdoor station using the nrf24l01 transceiver module as a receiver. The electronic circuit of the indoor station designed using Arduino Uno. The electronic circuit of the indoor station designed using Arduino Nano. Each station provided with a Lipo battery with 350 mAh capacity and 7.4 volts. The nrf24l01 module has the ability to transmitting data with 100 meters.

The decision tree algorithm used to make the decision with famous dataset knowing as play golf dataset. The indoor station has an LCD screen to display the weather conditions and after that showing the decision by printing yes or no.

In conclusion; a predictable weather station was designed. The Tasks of the station examined. It can be concluded that the weather station is ready to accumulate data from every environment. Also, it is able to make a decision about playing the game or not playing.

Keywords: Weather station, Arduino, Sensors, Decision Tree.

ÖZET

KABLOSUZ ALGILAYICI AĞ (KAA) ÖNGÖRÜLEBILIR HAVA İSTASYONU

Salim, Laith Mohammed

M.S., Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi,

Tez Danışman: Dr. Sefer kurnaz

Danışman:

Tarih: 3/2019

Sayfalar: 72

Bu araştırmanın amacı, doğru hava koşullarında oynama hakkında karar verebilecek bir kablosuz hava istasyonu oluşturmaktır. Hava durumu istasyonu, ortamdan veri toplamak için tasarlanmıştır. Gerekli verileri toplamak için kullanılan dört sensör vardır. Bu sensörler sıcaklık ve nem sensörü, Işık sensörü, Yağmur algılama sensörü ve rüzgar sensörü olarak dc motordur. Meteoroloji istasyonu iki kısım içerir. İlk bölüm, veri toplamak ve onları gönderen olarak nrf24l01 alıcı-verici modülüyle göndermek için dışa yerleştirildi. İkinci kısım, nrf24l01 alıcı-verici modülünü bir alıcı olarak kullanarak dış istasyondan veri almak üzere iç mekana yerleştirildi. Arduino Uno kullanılarak tasarlanan iç ünitenin elektronik devresi. Arduino Nano kullanılarak tasarlanan iç ünitenin elektronik devresi. Her istasyon, 350 mAh kapasiteli ve 7.4 voltluk bir Lipo bataryası ile donatılmıştır. Nrf24l01 modülü 100 metre veri aktarma özelliğine sahiptir.

Karar ağacı algoritması, oyun veri kümesi olarak bilinen ünlü veri kümesiyle karar vermek için kullanılır. İç ünite, hava koşullarını görüntülemek için bir LCD ekrana sahiptir ve bundan sonra evet veya hayır yazdırarak kararı gösterir.

Sonuç olarak; tahmin edilebilir bir hava istasyonu tasarlandı. İstasyonun Görevleri incelendi. Hava durumu istasyonunun her ortamdan veri toplamaya hazır olduğu sonucuna varılabilir. Ayrıca, oyun oynamayı ya da oynamamayı da kararlaştırabilir.

Anahtar Kelimeler: Meteoroloji istasyonu, Arduino, Sensörler, Karar Ağacı.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

LCD	:	Liquid Crystal Display
ID3	:	Inducing Decision Trees
CART	:	Classification And Regression Trees
AHP	:	Analytical Hierarchy Process
MSP	:	Mixed Signal Processor
WSN	:	Wireless Sensor Network
GSM	:	Global System For Mobile
GPRS	:	General Packet Radio Service
LM35	:	Linear Monolithic 35
IOS	:	Institute Of Software
ANOVA	:	Analysis Of Variance
AWS	:	Automatic Weather Station
RAM	:	Random Access Memory
Mb	:	Megabit
MPH	:	Miles Per Hour
TTL	:	Transistor To Transistor Logic
LiPo	:	Lithium Polymer
IDE	:	Integrated Development Environment
DC	:	Direct Current
LDR	:	Light Dependent Resistor
Mbps	:	Megabits Per Second
Mm	:	Millimeter
SCK	:	Serial Clock
CSN	:	Chip Select Not
mAh	:	Milli Ampere Per Hour
MISO	:	Master In Slave Out
MOSI	:	Master Out Slave In

1. INTRODUCTION

1.1 GENERAL OVERVIEW

Weather situations are one of the most important issues that affect our daily activities in many domains. The general idea of weather stations is giving us information about the current weather situation. This information may be temperature degree, humidity degree or any other measurements such as rain, wind, and luminosity. Many different fileds can use this information such as entertainment, military, and agriculture. With time, many different systems suggested for monitoring weather situations. The classic one is working by collecting data and then sending this data through the cables to one station which is a central station. It has a financial disadvantage because it required many buildings and transmission network. Another solution can be considered as a modern solution. It is using sensors to collect data. The number of the sensors can be unlimited and is able to cover wide by transmission the data wirelessly to the central station [1].

Weather prediction is one of the most applications that has become important today. Weather prediction provides information about future weather and about the suitable climatic conditions. There are many types of weather prediction systems. Some of them work directly by monitoring the differences in the sky while the other work depending on complex math equations in order to predicate the weather situation. Weather prediction important for many fields such as flying, sports, agriculture, and travelling. Also, weather prediction plays an important role in military operations, there is a considerable historical record of instances when weather conditions have altered the course of battles [2].

There are many examples of weather station systems have been developed for various applications in recent years, based on different microcontroller architectures and sensor hardware technologies, such as Arduino (Mega, Uno, and Nano) and Raspberry Pi. These microcontrollers are suitable with a large number of sensors. These sensors have the ability to connect the data directly from the weather [3].

This study concentrates on the development and implementation of a low-cost weather station to measure parameters of the weather in real time. Also, it will be able to predict what if the weather suitable for playing or not depending on the decision tree method.

1.2 MOTIVATION

The Weather station which presented in this thesis was specifically developed for helping people to monitor the weather and to be able to take a decision for playing or not with the current weather conditions. The weather station must be able to collect accurate and varied data about the weather to be able to make decisions based on sensory inputs. Then showing the decision to the user by using the LCD screen.

1.3 PROBLEM STATEMENT

Which are the variables to be collected? This is the first and the main question to solve before designing a weather station. Wanted here is a weather station that able to make a decision about the weather status. That's mean the weather station must get data from the environment to be able to make decisions. To achieve this goal the components of the weather station need to be known. The most important specifications of our weather station are:

- Sensors: the type and number of sensors very important to get the required data. Otherwise, our decision about the weather will be incorrect.

- Predictive Method: we need an algorithm that able to build a decision depending on a number of conditions.

1.4 OBJECTIVES

The design of weather station that capable of collecting data to find out if the weather is good or not for plying. The object of this project is to design a weather station, so that includes the mechanics, dynamics, sensor fusion, localization, and navigation. There are two main objectives to implement our research:

1. Built a weather station that can be placed in the outdoor. The station will collect and send the data in real time to the indoor station.
2. The ability to take a decision about the weather by using one of the machine learning algorithms.

1.5 THESIS OUTLINE

This thesis contains six chapters. The second chapter is a general overview of the decision tree algorithm and related works with our research. The third chapter is about hardware components that have been used in our work and how assembled all the components. In the fourth chapter, we explained weather station software and how we built our machine learning algorithm for weather prediction. The the fifth chapter is the results and discussions. And the sixth chapter is the conclusion and future works of the thesis.

2. LITERATURE REVIEWS

2.1 INTRODUCTION

Machine learning methods are the main part of any predictive system to be built. There are many methods can be applied to get a weather prediction system. In this chapter, we are going to give general information about the decision tree which a machine learning method we choose to use. Also, we are going to review some previous works that are related to our topic.

2.2 DECISION TREE

Classification is the prediction approach in machine learning techniques. There are many algorithms based on a classification such as Instance-based, neural networks, Bayesian networks, support vector machine, and decision tree. Induced Decision tree (ID3) is the basic algorithm for constructing decision trees. Decision tree lists examples by ordering them in a tree form starting from the root which is the first node through a number of nodes until accessing the leaf node. This path from the root to the leaf contains many nodes each one represents a feature of the example. These paths separate each example to its class depending on the features. This technique is used in many fields and a lot of them has a big effect on our daily life. The radar device work depending on the decision tree by analysis the incoming signal to object depending on the features. The application of number and letter identification helps to build many applications such as electronic reader devices and car plate recognition. Also, the decision tree helped in a medical determination by a diagnosis of the tumor depending on the size and place of the tumor. In general, the decision tree has the ability to build decision or make difficult decisions easy by process them as a multi sub-decisions then combine them into one simple and easy decision [4].

The tree is manageable, and technically it appears simple to work. In fact, it is more exciting to make a tree that is adjusted to the possibilities of variables to be examined. The frequently balanced tree will be a useful outcome. If a sub-tree can just drive to a novel answer, then every sub-tree can be demoted to the easy result, this clarifies the means and does not modify the last decision. Ross Quinlan worked on this type of decision trees [5].

In the next paragraphs, we quickly explain the popular and base types of decision tree algorithms.

2.2.1 ID3

Algorithm Iterative Dichotomies 3 is an uncomplicated decision tree training algorithm presented in 1986 by Quinlan Ross. It is serially achieved and based on Hunt's algorithm. The central concept of the ID3 algorithm is to create the decision tree by using a top-down search for the provided sets to examine each attribute at each tree node. In the decision tree classification, information gain method is commonly used to define a suitable feature for every node of a created decision tree. Because we can choose the characteristic with the largest information gain (entropy decrease in the level of height) as the examination attribute of the current node. In this form, the information required to classify the training example subset collected from next on partitioning will be the smallest. So, the advantage of this feature for partitioning the individual set included in the current node will cause the mixture ratio of various types for all created individual subsets decreased to a minimum. Hence, the use of an information theory method will completely reduce the expected dividing amount of object classification [6], [7].

2.2.2 C4.5

C4.5 offered by the same researcher [Quinlan (1993)], applies gain ratio as dividing rules. The splitting terminates when the amount of examples to be split is under a specified threshold. Error-based pruning is done following the growing stage. C4.5 can work with numeric characteristics. It can also produce from a training set that includes missing values by applying corrected gain ratio tests. C4.5 algorithm presents many advances over the ID3. The common significant advances are [7]:

(1) Pruning: which mean deleting the nodes or subtree which do not lead to any class.

(2) Missing data: it has the ability to work with missing data.

(3) Continuous values: many machine learning technique is working with discrete values. The c4.5 has the ability to work with continues values by asserting a threshold and separate the data depending on this threshold into ranges which represent a subset for each attribute.

These three advances of c4.5 which make it different from the ID3 and giving it the ability to perform new application by using these differences.

After C4.5 the application started to be more economic and required higher efficiency techniques. The C5.0 is the next version of the C4.5 which contain much enhancement over the C4.5 such as the computation time. For example, if a simple job which takes one and a half hour to execute using the C4.5. The C5.0 can implement it with three and a half seconds. Also, the C5.0 has some advance in representation where it uses the memory more efficiency compared with the memory size taken by the C4.5 [7].

2.2.3 CART

The abbreviation CART refers to the tree of classification and regression together. This algorithm was introduced by Breiman in 1984. The basis of this algorithm is a binary theory, where each node contains only two outputs. In order to make divisions, the Towing Criteria method is used. After that, the tree is pruned using Cost-Complexity method. The most important feature in this algorithm is the tree of the gradient in the tree of the gradient there are several differences, but the most important difference is the representation of the categories using real numbers and not the category [7].

2.4 CLASSIFICATION WITH USING THE ID3 ALGORITHM.

In this section, we will take a closer look at how to building a tree using ID3. The root is the first node to be chosen in the tree. As the root is known as one of the attributes in the training set, so which one do we need to pick first?

To choose the root, we will determine the attribute that best classifies the training data. This attribute can be found by calculating the information gain. The information gain of this attribute is the highest. After that, this attribute will be used as a starting node for the tree called root. This procedure will repeat for each branch until scanning all attributes in the training data. This means performing a top-down search through the space of possible decision trees.

2.4.1 Calculate the Information Gain

Calculating the information gain depending on calculating the entropy which is coming from information theory. Entropy measures the amount of uncertainty in the data set S. The entropy can be found using the equation (2.1) below:

$$H(S) = \sum_{c \in C} - p(c) \log_2 p(c) \tag{2.1}$$

Where,

- S is the current dataset for which entropy is being calculated.
- C is the set of classes in S. C= {yes, no}
- P(c) is the proportion of the number of elements in class c to the number of elements in set S.

The equation (2.2) of information gain depending on the entropy will be in this form:

$$IG(A.S) = H(S) - \sum_{c \in C} - p(t)\,\mathrm{H}\,p(t) \qquad (2.2)$$

Where,

- H(S) is the entropy of set S.
- T is the subsets created from splitting set S by A such that $S = \bigcup_{t \in T}^{t}$.
- *p(t)* is the proportion of the number of elements in t to the number of elements in S.
- H(*t*) is the entropy of subset *t*.

The value of information gain will calculate to each feature in the training set. After that, subtract the general entropy from each result. The attribute with the largest result will be the root of the tree.

These are the common actions for achieving the ID3 decision tree:

1. Calculate the entropy for data.

2. for each characteristic/attribute:

   i. Find entropy for all definite amounts

   ii. Get medium information entropy for the current characteristic

   iii. Find the current characteristic gain.

3. Choose characteristic with a large gain.

4. Reform till built the whole tree.

From the Pseudo Code in [5], we can summarize the steps of ID3 by using the flowchart in the figure below.

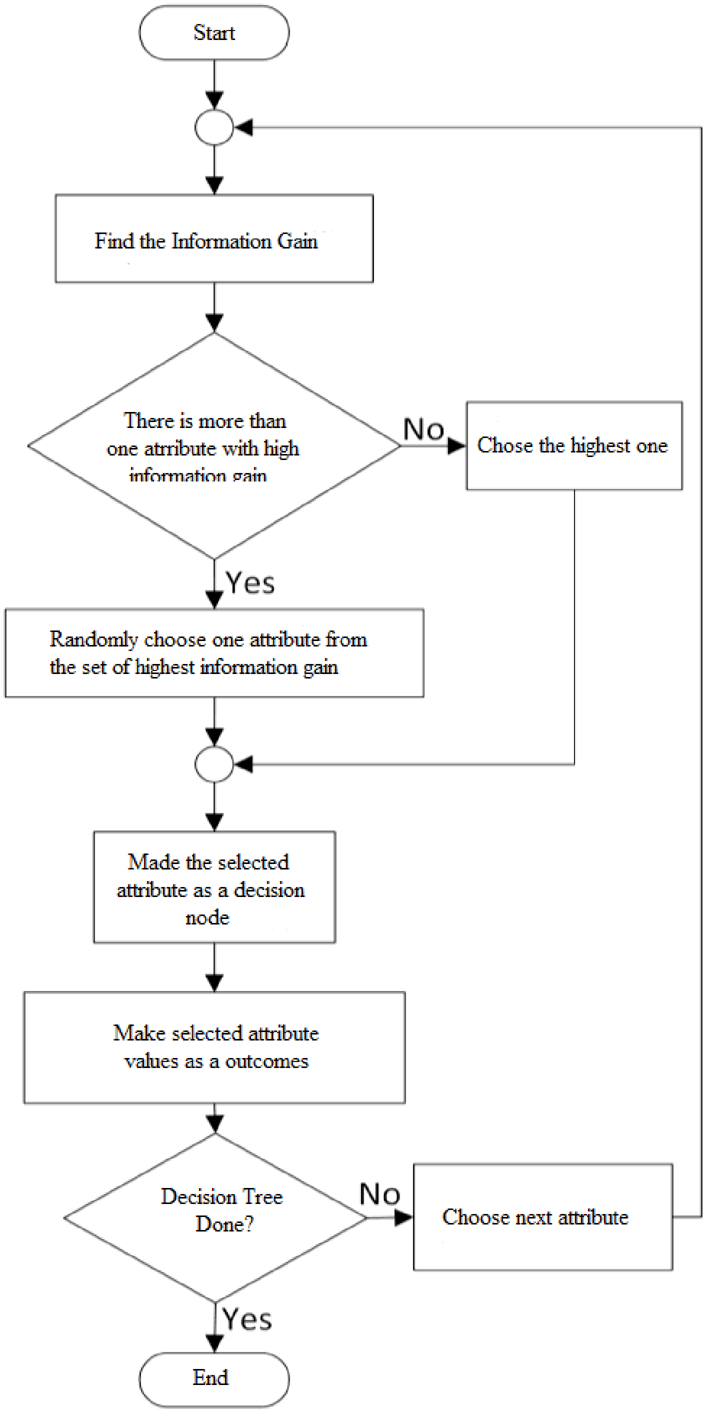

Figure 2.1: ID3 Algorithm Flowchart

2.5 LITERATURE REVIEWS

There are much research has been done to build a weather station and weather forecasting. In this section, we analyze the existing researches.

In [8], the researchers created a quality evaluation system for evaluating the work of automatic weather station. These system work based on the examination of the construction and performance status of the automatic weather station, depending on the theory of Analytical Hierarchy Process (AHP) in fuzzy systems. The result showed that the evaluation of the performance quality depending on enhanced fuzzy AHP was logical, understandable and possible. The results received in two ways: the way of a fuzzy vector, following the company degree of evaluation results on the different levels, it serves to recognize the performance state of the automatic weather station. Another way was a number, it gives to the horizontal correlation of various automatic weather stations and is simple to recognize basic problems.

The [9] showed a weather station which able to detection rain and powered by solar. This weather station could be controlled remotely and also is able to display the data on an LCD screen in a simple way that can read by users by using simple numbers. The weather station contains two part, first one a remote station which placed outdoor for collecting data from the weather and is powered by a solar panel. The second part is a base station which contains screen to displayed data. The first part of the station hold sensors to estimate humidity, temperature, rain, and light level. The object of the system was optimizing price and power. In the second station, the wireless receiver module (ZigBee based) receive the transmitted data and sends to the low power microcontroller (MSP 430) which drives the LCD display unit. The ZigBee module was mainly responsible for the wireless transmission of the signal to the receiver.

The model of prediction weather station by using machine learning techniques such as decision tree proposed by the [10] to prognosticate some event such as rain, fog, and noise by including average temperature, humidity, and pressure. The dataset which used for the decision tree was used for one year from the http:///www.wundergrounds.com. Only 64 instances used as a training set. Only three attributes had been delivered into the record (average humidity, average temp and sea level). For test the 72 instances made by choosing data randomly. Decision tree executed in Weka to facilitate the prediction of weather. The result showed that 46 tests were classified properly from the 72 test instances.

In [11], the number of weather stations and by using Wireless Sensor Network (WSN) technique, the built a system to monitor the weather in Perlis city. The collected data by the sensors nodes from the weather send to the computer system. They created a web page using for monitoring the state of the weather. The weather station able to collect data about rain, humidity, temperature, pressure, and light level. Robin Z530L, which is a high-performance computer able to do high computing with less power consumption used as a server for the system. This server able to store the data and the log information for the users.

In [12], they provided a survey by using information from the period of 2014 and 2015 especially between November and January to determine the situation of weather stations in Uganda. They found the density of the weather station was rare and many of the weather stations were not work correctly. For this reason and depending on the distribution of the running weather stations, they proposed new strategies to ensure climatological zones, security, and land policies. They visited 14 weather stations, 2 were manual, and 2 are working automatically while 10 works in two way, automatic and manual, weather stations in the same place. These 14 weather station collected the data from the weather and saved these data in the database which placed at or near to the weather stations. After the time between one and three hours, the database sent the data to monitoring station manually or automatically using any type of transmissions such as wifi, GSM/GPRS, or Ethernet connections for Internet access. At the monitoring station, the data combined from whole weather stations in the country to be supplied into prediction systems.

The [13], showed a weather station using Arduino board. They used Arduino microcontroller to collect data from the sensors. The sensors were LM35 Temperature, DHT22 Humidity, and BMP280 Barometric Pressure sensor. The used Bluetooth HC06 module to communicate the weather station with the computer. Arduino coding language used to program the microcontroller. Also, they used PuTTy which is a serial console to test wireless communication. For displaying the values graphically, they used Matlab. Their results showed the variation in heat, moisture, and pressure for two hours. The heat estimated by LM35, DHT22, and BMP280, moisture estimated by DHT22 and pressure by BMP280.

The [14] presented a real-time weather station work automatically for measure locally and continuously for some factors of weather such as humidity, barometric pressure, temperature, rain in addition to wind speed and direction. The weather station consisted of indoor weather station and outdoor weather station. These two stations connected wirelessly by using the 434 MHz wireless modules as a simple wireless connection between two devices. These two devices can send and receive data with a range of approximately equal to 150 meters. The outdoor weather station contained all the sensors. The indoor station displays the outdoor reading by using liquid LCD. Also, for monitoring the data, they built an iOS and Android application can be accessed from any device and the named the application: Blynk.

The [15] research aimed to produce and develop weather station with low cost for weather examination. This article discussed the review of weather stations with good efficiency initial model that satisfies all the features of a total station by using three kinds of sensors for the estimation of the humidity, temperature, and atmospheric pressure variables. These sensors are TMP36 sensor which measures temperature, RHT03 sensor which measures humidity, and BMP085 sensor which measure the pressure. An examination of change by using ANOVA to find the variance and an r&R test created to conclude which sensor affords greater achievement than the to give the excellent achievement. For this analysis, 100 copies were taken out at several times in order to average the units required to validate the test performed. For this round, the examples period started in order to pick the sensor, from the three sensors, with a greater performance versus the weather conditions.

The [16] researcher built weather station work in a real-time depending on mobile systems and by using Automatic Weather Station (AWS). The system contains many sensors. the data stored in a web server. This system with the sensors connected to the AWS. Data cycle was stating by collecting it from the weather by the sensors. Then it is going to the AWS using Weather Link software. After that, the data sent over the data logger using serial transmission mechanism to be stored in the web server. The Android application received the data and showed the information stored by the web server in real-time. The estimation parameter involved humidity, temperature, rainfall, air pressure, and solar radiation. The application used almost 58.7 Mb from the RAM. The use of 58.7 Mb RAM allowed the application to work and placed on Android devices that have smaller RAM for example 512 Mb. The application showed the current estimation parameters of weather situations. Also, there are graphs for each parameter in a separate menu.

In [17] the researcher worked to build weather station for monitoring the weather with low-cost material and allowing all people and companies including civil people and government companies to monitor the weather conditions. The main difficulty of this weather station was the wireless communication to allow all people in the country to access the weather data and view live readings by using any personal phone or computer. They built a number of weather stations with cost 50$ for each one. They applied number techniques in order to remove the outliers from the data. They made a test for their work which took one month. This month consisted of 15 days as a preliminary time and 15 days as a forecasting time. The result was good for predication weather for a short time period but it was less effective for the longtime predictions. The problem of longtime predictions was clearly because the accuracy of predictions decreased. The r-squared value showed that the accuracy decreasing by falling between 0.13 and 0.14.

The [3] work presented the development of a low-cost weather station capable of measuring meteorological data including air temperature, humidity, barometric pressure, wind speed, wind direction, and rainfall. A LiPo battery and four solar panels were used to power up the complete system. The weather station communicated through a Wi-Fi connection which allowed the user to access the data remotely. An external Raspberry-Pi was implemented to create a local MySQL database to extract the data collected and a (“phpMyAdmin”) administrator page was used to access the database. The weather station was mounted on an adjustable tripod base with a telescoping arm which raised the weather station sensors up to 3 meters and provided structural support capable of withstanding winds of up to 28.822 m/s (60 miles per hour (mph)). Data gathered from the weather station was compared to a professional-grade weather station located at the same site. Results showed that the low-cost weather station was a feasible option to use for site assessment purposes at a comfortable cost.

2.6 WEATHER STATION TYPES

There are many types of weather stations in the world. Each one different from the features, size, ability … etc. In this part, we are going to explain the famous available types in the world [18]:

2.6.1 House weather stations
2.6.2 Expert weather stations
2.6.3 Customized weather stations

2.6.1 House weather stations

This type of weather stations is simple and low price. It usually measures only the temperature and humidity. Some of them give analog readings and look like a clock with hands pointing to the temperature and humidity numbers. These are low speared but it is easy to read. Figure (2.1) showing the analog home weather station.

Figure 2.2: Analog home weather station.

The rest of the home weather stations is digital. In these types, the time or the date is given with the temperature and humidity. These two types give the readings depending on the indoor weather and didn't receive any data from the outdoor.

After the evaluation which occurs in the communication filed, some of the home weather stations has two parts. One placed outdoor which contains the sensors. The second part placed indoor which responsible for showing the received data. The outdoor part offers new features to be measured such as the sunlight and the wind speed. Figure 2.3 showing the indoor-outdoor home weather station.

Figure 2.3: Indoor outdoor home weather monitor

This type of home weather station can measure:

- The temperature degree for Indoor and outdoor.
- The ratio of humidity
- Atmospheric pressure
- The direction and speed of the wind.
- Sunlight
- Rain

2.6.2 Expert weather stations

The expert weather stations are building with some requested features. The station may require to be reliable for building reliable applications or it needs to be durability to face the hard conditions of the weather. There are many international standards and accuracy requirement initialized by some Entities such as National Institute of Standards and Technology that must be implemented and performed in the weather station in order to consider as an expert weather station. Expert weather stations can be different in size, type, and quality. Some of them similar to the house weather stations while some of them can include many multi substations and connected with a network and special building to hold them. This type can cover a wide range of area. Figure 2.4 showing the expert weather station.

Figure 2.4: Professional weather station

Most of the expert system collect data depending on the required. the end user can determine which conditions must be collected such as temperature, rainfall, sunlight humidity or any other conditions. For example, the farmers may request the temperature degree and humidity ratio in order to monitor the irrigation process. In the airports, the employees and pilots care about the wind speed and direction and the amount of rainfall in general.

The applications of the expert weather stations:

- Serious sports fans.
- Manufacturing and marketing companies
- Airports
- Farmers

- Government companies
- Fire departments
- Event administrators
- Insurance organizations

2.6.3 Customized weather stations:

There are many types of specialty weather station such as portable, Agricultural, and Remote.

The portable which knows as transportable weather stations vary from small size which can handle by the hand to give information about humidity and temperature only up to big size such as suitcase types which can give information about temperature, atmosphere pressure, rainfall etc. also, it has the ability to add any specific sensors such as nuclear pollution detectors. Figure 2.5 showing the handheld weather station.

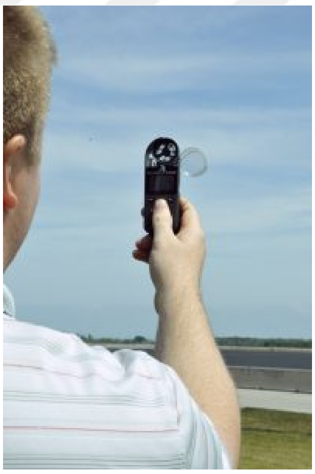

Figure 2.5: Portable weather station

The second type is the agricultural type of customized weather stations which contain sensors related to plants such as soil moisture and temperature. These stations also monitor the weather conditions in addition to soil conditions. One of the important features of this station is the networking in order to cover a wide range of area of agricultural then collecting data from this area in a central monitoring system which can be located indoor.

The third type is the remote weather stations. This type of stations built for the places which are difficult to reach. In general, these places are so far away and it is costly to reach it every day. The important features of this station are depending on solar power as a power source and it must contain a wide range of communication medium to send data away. Some examples of the places which suitable for this type of station are mountain, sea, and space.

Figure 2.6: Remote weather station

3. HARDWARE COMPONENTS

3.1 INTRODUCTION

This thesis proposed for designing a predictable weather station for helping human to monitor the weather and made a decision about playing a game or not. The weather station works automatically without any controller by an external work station. It supplied with a number of sensors to get information from the weather. This chapter will explain the hardware components used in the weather station and the assembling of the components.

3.2 HARDWARE COMPONENTS

The weather station contains two parts one will place indoor and the other will place outdoor. In the next paragraphs, all the details of the components will explain.

3.2.1 Arduino

Arduino is open-source hardware. It is able to produce instructions to many hardware components such as motors, sensors, screens etc.

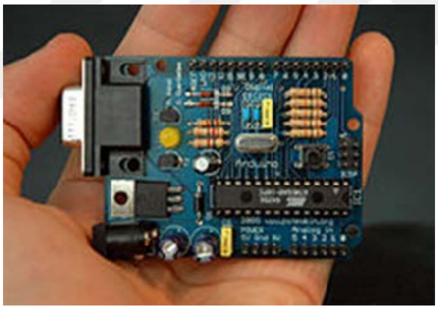

Figure 3.1: RS-232 serial interface.

The first microcontroller [19] started with serial interface known as RS-232 as shown in the figure (3.1) above. Using an Atmel ATmega8 microcontroller chip (the black chip in the lower left side in the figure above); at the top there are fourteen pins as a digital input and output pins, in the bottom of the board there are 6 analog pins, and in the left, the power source can be plugged in. Arduino microcontrollers can be used by a bootloader which doing the process of uploading the written code to the microcontroller. Board could be programmed by the code and uploading this code by using a serial cable with the computer. In small numbers of Arduino boards, there is a simple shifter circuit which works as a converter to convert the (TTL) which is a serial level to the corresponding signals. In current days the USB is using to make the connection between Arduino and computer.

There are many different types of Arduino boards developed until the present day. Table 3.1 showing a comparison between the numbers of a different type of Arduino boards [20].

Table 3.1: Types of Arduino boards

Board name	Processor	VCC (V)	USB	Clock (MHz)	PWM pins	Analog Input Pins	Digital Input Pins
ArduinoBT	ATmega328	5	Type	16	6	6	14
Duemilanove	ATmega168	5	None	16	6	6	14
Duemilanove	ATmega328	5	Regular	16	6	6	14
Diecimila	ATmega168	5	Regular	16	6	6	14
Esplora	ATmega32U4	5	Regular	16	-	-	-
Ethernet	ATmega328	5	Micro	16	4	6	14
Fio	ATmega328P	3.3	Regular	8	6	8	14
Leonardo	ATmega32U4	5	Mini	16	7	12	20
LilyPad	ATmega168V	2.7–5.5	Micro	8	6	6	14
LilyPad	ATmega328V	2.7–5.5	None	8	6	6	14
Mega	ATmega1280	5	None	16	15	16	54
Mega ADK	ATmega2560	5	Regular	16	15	16	54
Mega 2560	ATmega2560	5	Regular	16	15	16	54
Micro	ATmega32U4	5	Regular	16	7	12	20
Mini	ATmega328	5	Micro	16	6	8	14
Mini Pro	ATmega168	3.3	None	8	6	6	14
Mini Pro	ATmega168	5	None	16	6	6	14
Nano	ATmega168	5	None	16	6	8	14
Nano	ATmega328	5	Mini-B	16	6	8	14
Pro (168)	ATmega168	3.3	Mini-B	8	6	6	14
Pro (328)	ATmega328	5	None	16	6	6	14
Uno	ATmega328	5	None	16	6	6	14
Yún	ATmega32U4	5	Regular	16	7	12	20

The Arduino Uno board (which is the popular microcontroller nowadays), provides the same number of pins which introduced before. The tool which required for writing Arduino program can be any famous programming language to convert any instruction to a binary signal which is the target. Also, the company built an environment which can be used as an environment for all Atmel boards [19], [21], [22].

The Arduino NANO is also popular as the Arduino Uno. The Arduino Nano has a small size, breadboard-friendly, complete board, and based on the ATmega328P. It has the same functionality of the Arduino Uno but without the ability to using the DC power as a power source, and works with a Mini-B USB cable instead of a standard one. It is working with 5 Voltage. It contains 22 I/O pins (6 of which are analog). The board size equals 18 x 45 mm. figure (3.2) showing the Arduino NANO.

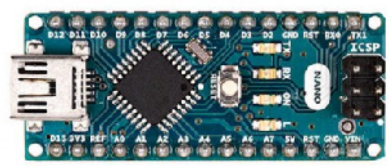

Figure 3.2: Arduino NANO

In most cases, the IDE is using for building Arduino code. The IDE which is abbreviated for the integrated development environment (IDE) written by Java language. It contains a window for editing the code. Also, it provides all the standard tools for text processing such as copy, cut, and paste. To upload the code to the board, the environment provides a simple method for compiling and uploading by pushing a one-click icon. It also includes a communication block and keys for basic operations.

The code now calling sketch [23],[24]. After that, they are kept on the computer as text files. They can by processing as a text file but they have the .ino extension. The environment maintains the languages C and C++ by applying specific commands of coding. The Arduino IDE provides a software library of the cables project, which provides several standard input and output methods. The general structure of the Arduino code contains two basic functions. The start which responsible for starting the program will all variables and initial values. The second function is the loop which is an infinite loop to execute all the instructions continuously. The environment is also responsible for convert the running code into a file. This file will be represented by the hexadecimal system to be executed by the Arduino. Most components have an open source library to make the board easy to use.

3.2.2 16x2 LCD Screen

The 16x2 Character LCD Display has the outline measurements of 80.0 by 36.0 mm and depth equal to 13.2 mm. This screen requires between 5 and 3 volts for working. The writing in the LCD can be implemented by using the pins in the screen which equal 16 pins or by using I2C interface module. Figure 3.3 showing the 16x2 LCD screen.

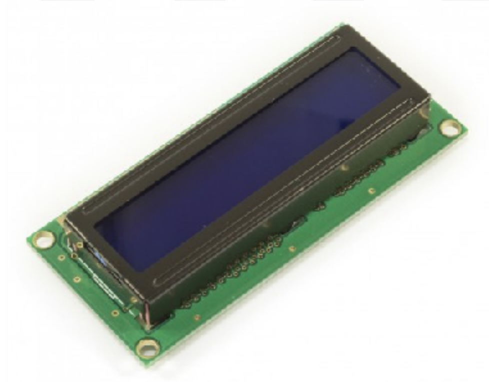

Figure 3.3: the 16x2 LCD screen

3.2.3 Battery

The basic kind of batteries which accepted now in the robot is the Lithium Polymer batteries which recognize as (LiPo). These batteries contain some lithium classes such as manganese lithium or any other kind of lithium. There are several different kinds of batteries such as lead acid, NiMH and NiCad but their weaknesses are many when comparing them with the lithium batteries. The main advantages of lithium are low weight and high capacity. Also, it quick to charge compared with the other.

The voltage is the important determinant which must be thought when picking the battery for the project. It is a critical determinant for the batteries. In common, lithium batteries can be made by merging a number of cells. The number of cells starts from 1 to infinity. Every cell produces 3.7 volts. In this form, the necessary voltage can be calculated first then depending on it the number of cells can be collected. For example, the couple of cells battery supplies 7.4 voltage while the battery with three cells supplies 11.1 voltage. Depending on the voltage and the capacity the charger of the battery can be chosen in an ideal way to guarantee fast recharging.

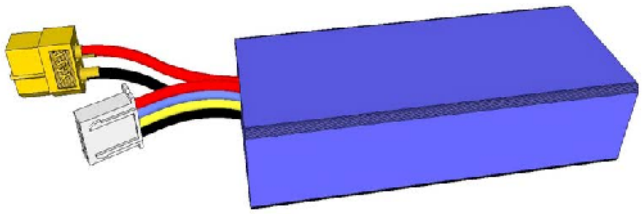

Figure 3.4: Lithium polymer (LiPo) Battery.

3.2.4 Sensors

After the evolutions which occur in the electronic world, the sensors become more popular and usable for many different tasks. There are many types of sensors with different sizes some big and some very small. The mean idea of the sensor is detecting and responding to the change which occurs in the environment. There are many different sensors for this reason there are different inputs of sensors such as temperature, water, gas or any other environmental phenomena. The output is an electronic voltage with value depending on the input changing. There are many characteristics that led to introduced different types of sensors. Some sensors are active which require an external source of power while some of them are passive in which the entirely provided by the measured signal without an excitation voltage. Also, some sensors are digital sensors where the signal produced or reflected by the sensor is binary. On the other hand, Analog sensors where the signal produced by the sensor is continuous and proportional to the measured.

We have a number of sensors in our weather station to collect data from the weather. We will explain the characters of each one in details.

3.2.4.1 Humidity and Temperature (DHT11) Sensor

The DHT11 sensor measures both relative humidity and temperature. The sensor is protected by a plastic case which makes it a bit bigger than the other sensors. The DHT11 sensor contains three pins one of them for data output with the voltage and ground. The DHT communicates with the microcontroller using a special library used for the DHT sensors which are described in its datasheet. It operates on 3.3 to 5.5V and can measure humidity between (15 and 95) % Relative humidity RH and temperatures between (-2 and 52) [13]. Figure 3.5 showing the DHT sensor.

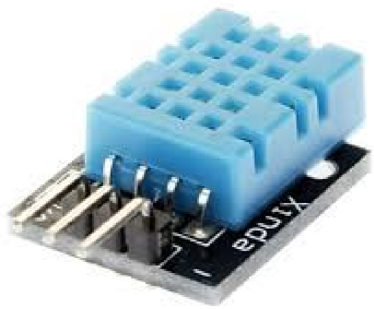

Figure 3.5: DHT Sensor.

3.2.4.2 Rain Sensor

The rain sensors are generally responsible for detecting rain only without measuring the amount or density of rain. This rain sensor works with Arduino and it contains two part. The first part which is a plate with the two electrodes positive and negative. Each electrode represented by many tracks without any connection with the Opposite electrodes. The raindrop work as a connector between them and in this way the rain is detecting. The second part is a small board responsible for detecting the communication between the two electrodes and sending signals depending on the plate. The sensor work with a voltage between 3 and 5 V. [25]. Figure 3.6 showing the Rain sensor with its two parts.

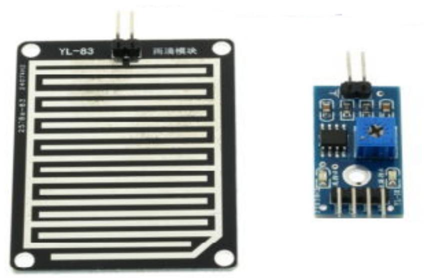

Figure 3.6: Rain Sensor.

3.2.4.3 Wind Sensor

One of the practical ways for detecting the wind is by using DC motor. The DC motor converts mechanical energy into electrical energy. Thus, if the wind generates mechanical energy should generate electricity. For this task, the DC motor with a plastic fan can be used. Some of DC motors come with a stand that makes them suitable for this task. Figure 3.7 showing the DC motor with a plastic fan and stand.

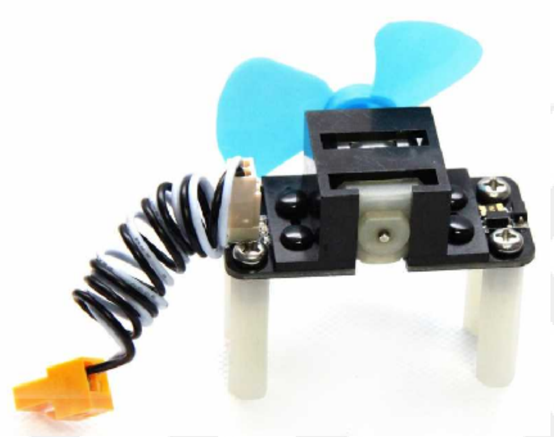

Figure 3.7: DC motor with a plastic fan and stand.

3.2.4.4 Light Sensor

The light sensor for Arduino contains a Light Dependent Resistor (LDR) with two legs. The LDR detect the light and then sending the information to the Arduino pin. Figure 3.8 showing the light sensor.

Figure 3.8: Light sensor

3.2.5 NRF24L01 Module

The nRF24L01 is a 2.4 GHz radio transceiver with a max data rate of 2 Mbps and the range is 100 meter in open area. The nRF24L01 is a low power consumption device which runs on 3.3V but the SPI pins on the device are 5V compatible. When it starts to transmit, the current peaks to 150 mAh. The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time [27]. figure 3.9 showing the nRF24l01 nodule.

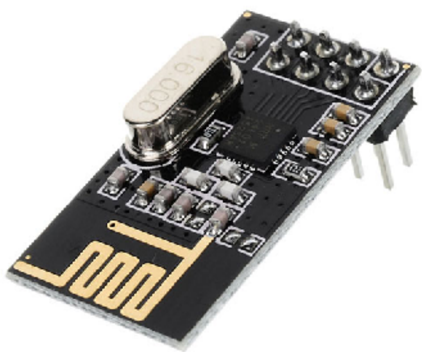

Figure 3.9: nRF24l01 Module

3.2.6 Plastic case

The plastic case is the last component of the weather station. For the outdoor sub-station the plastic case very important for saving the electronic components from the rain. For the indoor sub-station plastic case used to making the station shape more regular. This mean there are two plastic cases used one for the outdoor part and the second for the indoor part.

Given the number of the components used in the outdoor part, which contains a larger number, the size of the case has been larger from the one which used in the indoor part. The outdoor plastic case has 130x138x48mm dimensions. Figure 3.10 showing the outdoor plastic case.

Figure 3.10: Outdoor plastic case

The indoor plastic case has 115x83x32mm dimensions. Figure 3.11 showing the indoor plastic case.

Figure 3.11: Indoor plastic case

3.3 ASSEMBLING THE COMPONENTS

3.3.1 Outdoor Station

The outdoor station contains all the sensors. It is responsible for gathering the data. The first piece in the outdoor station is the microcontroller. We used Arduino NANO as a microcontroller for the outdoor station. The Arduino Nano has a small size which makes it suitable for placing it in the plastic case. After placing the Arduino Nano, we placed the lithium battery and supplied the Arduino with the power. The Arduino received the power with VCC pin and ground pin. Our battery voltage is 7.4 v which make it suitable for the Arduino which accept input voltage between 7 and 16 volt.

The second step is to connect the sensors to the Arduino. Most of the sensors give analog voltage as an output. Since the sensors give analog output, they are connecting to the analog pins on the Arduino. The Arduino uses built-in ADC (analog-to-digital converter) which converts the analog voltage (from 0-5V) into a digital value in the range of (0-1023). The first sensor is the DHT11 sensor which measures the humidity and temperature. This sensor has 3 pins two of them for power and one for data. The power pins connected to voltage and ground. The data pin connected to the analog pin A0 in the Arduino. We placed this sensor inside the plastic case for saving it from the rain. The plastic case open from the two sides which makes the sensor read real accurate values.

The second sensor is the rainfall sensor. As we mentioned before, the rainfall sensor has two parts. The first part the sensor plate which placed outside for detecting rain. This part connected to the second part with two wires. The second part which is a board placed inside the plastic case. It receives the signals from the output part and then sending it to the Arduino through the data pin to the analog pin A5. Also, this part receiving its power with a voltage and grand pins.

The third sensor is the light sensor. The light sensor placed outside. It already contains a protection layer from the rain. This sensor contains two pins. One of them for voltage. The second one connects to the analog pin A1 on the Arduino. A 100K resistor is also connected to the same leg and grounded.

The last sensor is the wind sensor. As we said before we used DC motor as a wind sensor. This sensor placed outside the plastic case. We cover the electric board with insulating adhesive for electricity. The Arduino receives the incoming voltage from converting the mechanical energy (by the wind) into electrical energy. The Arduino receiving this energy in analog pin A3.

After placing and connecting the sensors, the last piece in the outdoor station is the NRF24L01 transceiver. The NRF24L01 has 7 pins to connect. Two of them which is voltage and ground connecting to the 3.3 volts in the Arduino and to the ground. Three are connecting to the Serial Peripheral Interface (SPI) pins in the Arduino. The SPI pins of the Arduino are digital 11, 12 and 13 which are the Master in Slave out (MISO), Master out Slave in (MOSI) and Serial Clock (SCK) respectively. The rest two pins which are the CSN (Chip Select Not) and CE (Chip Enable) responsible for determining the state of the module into active or standby. They can receive signals from any pin with a digital signal. When the active mode is selected or the standby mode has selected the mode of the module changing from transmitting to command or the inverse. We connected them to digital pins 7 and 8 respectively.

Figure 3.12 showing our board after connected all the components. Figure 3.13 showing the outdoor station.

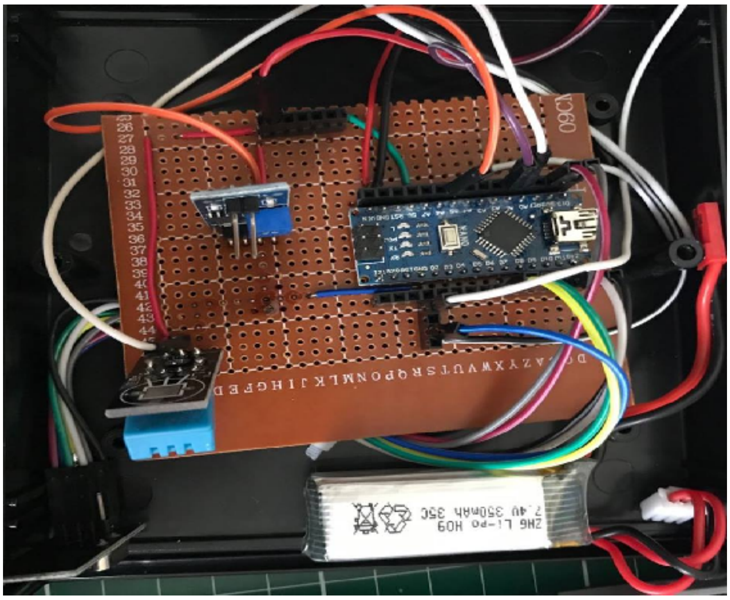

Figure 3.12: Sensors Board

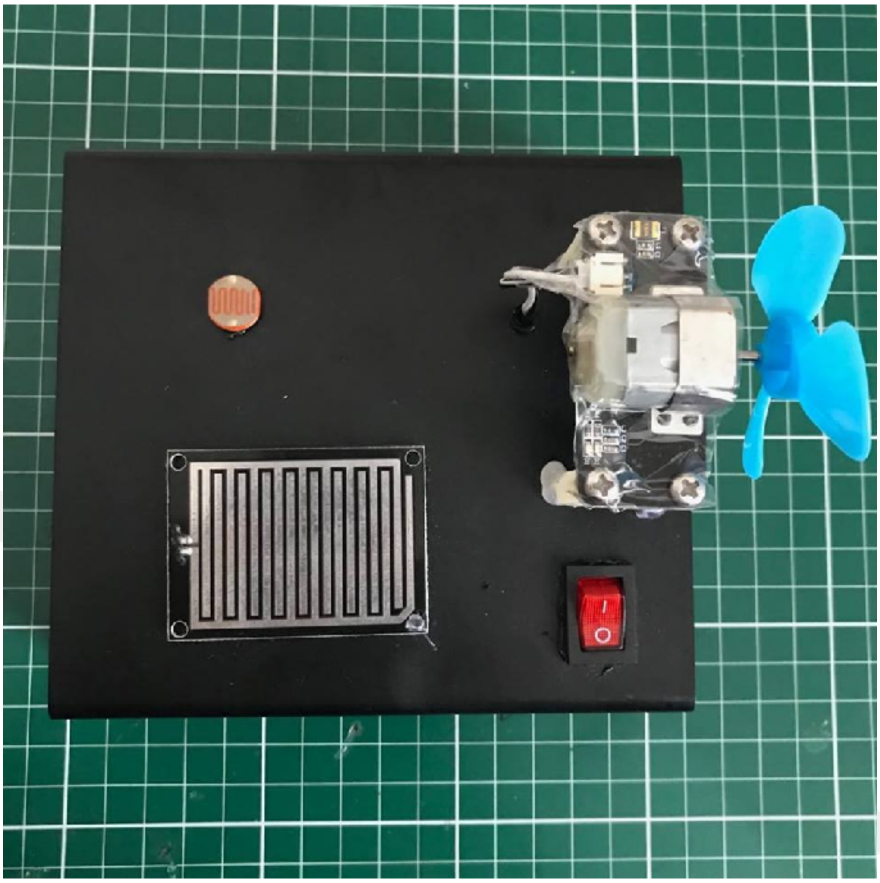

Figure 3.13: Outdoor Station

3.3.2 Indoor Station

The indoor station contains the display screen. It is responsible for receiving the data from the outdoor station and making a decision depending on the received data. The first piece in the indoor station is the microcontroller. We used Arduino Uno as a microcontroller for the indoor station. The Arduino Uno has all the required pins both for the LCD screen and the receiver which make it suitable for using it in the indoor station. After placing the Arduino Uno, we placed the lithium battery and supplied the Arduino with the power. The Arduino received the power with VCC pin and ground pin. Our battery voltage is 7.4 v which make it suitable for the Arduino which accept input voltage between 7 and 16 volt.

This station contains two components first the 16x2 LCD screen and second the NRF24L01 transceiver. For connecting the LCD screen we used I2C display module. Figure 3.14 showing this module. This module makes the connecting and the controlling easy with the LCD screen. Without it, it is necessary to connect 16 pins to the LCD screen while with it we connected only 4 pins.

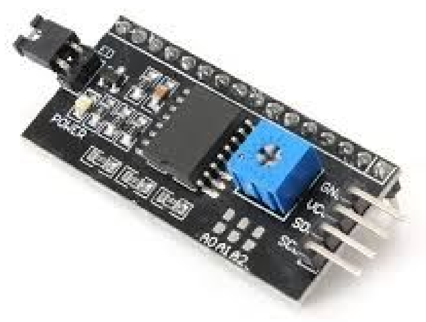

Figure 3.14: I2C Display Module.

This module contains 4 pins two for supply the screen with power and two pins for receiving the data from the Arduino to display it. The data pins are SCL (Serial Clock) and SDA (Serial Data Access) which connect directly to the Arduino SDA and SCL pins.

The last piece in the indoor station is the NRF24L01 transceiver. It is connecting as the case in the outdoor station without any difference. Figure 3.15 showing our board after connected all the components. Figure 3.16 showing the indoor station.

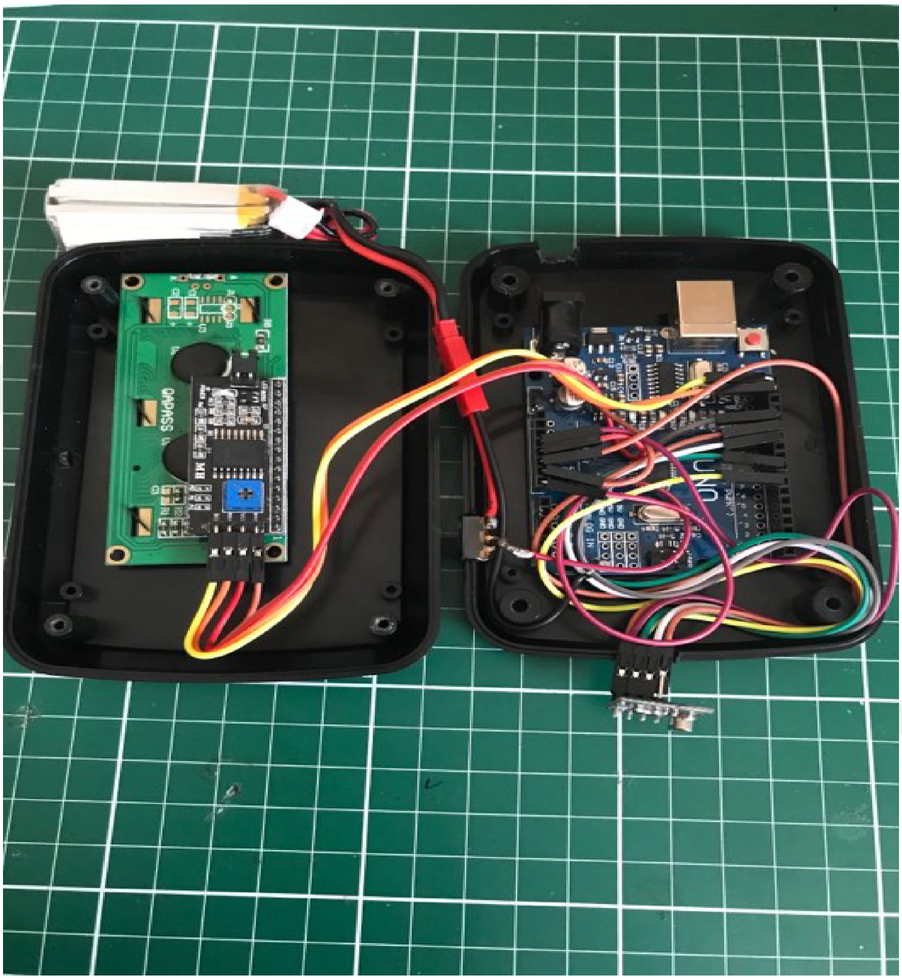

Figure 3.15: Indoor Weather Station Components.

Figure 3.16: Indoor Weather Station.

4. WEATHER STATION SOFTWARE

4.1 INTRODUCTION

This chapter explains the software part of the thesis. The weather station software built using Arduino integrated development environment or as known Arduino software (IDE). This environment allowing the developer to write their programs on text editor with using the toolbar which contains a lot of functions and menus. This software connects to Arduino boards using the USB port to upload the written program. The code was written by the developer known as sketches and it is saving as ino file. Figure 4.1 showing the Arduino software (IDE).

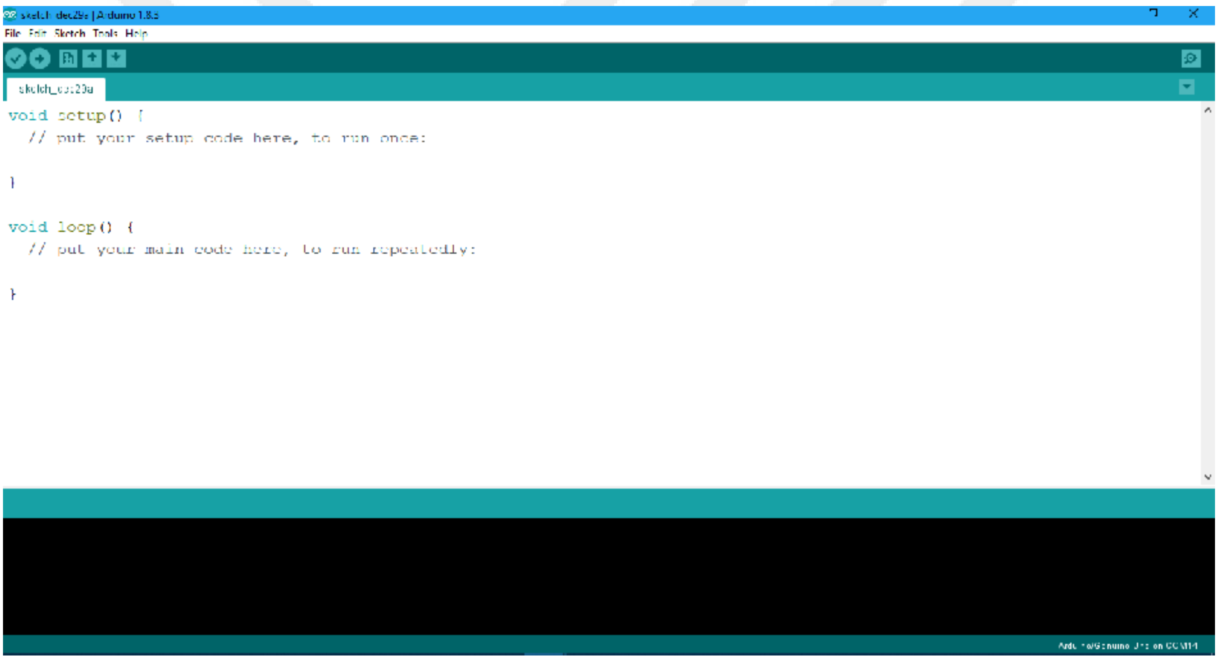

Figure 4.1: Arduino integrated development environment.

As we see in the figure above, there is two main part of the text editor. The setup area and loop area. In the setup area, the user can set up the I/O pins. Also, the communication type with the data rate in bits per second (baud) for serial data transmission.

In the loop area, the user can write the instructions which will implement in a loop cycle. This instruction can be output signal such as Light, Movement, sound …etc. or can be input signal such as sensors readings.

The software of this thesis contains two parts. The first one, the software for the outdoor part which deals with the sensors. The second part of software deals with the indoor station in which the decision tree will be implemented. So in this chapter, we will talk about two main parts:

4.2 Outdoor Station Software.

4.3 Indoor Station Software.

4.2 OUTDOOR STATION SOFTWARE

As we said before, the outdoor station dealing with the sensors. It is responsible for taking signals from the sensors and sent it directly to the indoor station. The software starts with including the required libraries. The first library is SPI (Serial Peripheral Interface) this library allow the Arduino to communicate with one or more SPI devices. The second two libraries are nRF24L01 and RF24. These two libraries required for using the nRF24L01 transceiver. The third library is DHT. This library used to access the dht11 temperature and humidity sensor [28].

After including the required libraries, the next step is to define an interface to the DHT library. After that, we will determine the pins of each sensor. For the temperature and humidity sensor, we used pin A0. For the light sensor, we used pin A1. For the rain sensor, we used pin A5. For the wind sensor, we used pin A3. Also, we set pin 7 and 8 for the CE (Chip Enable) and CSN (Chip Select Not) for the transceiver. After that, the address will place in an array of bytes. This array is known as a pipe. This address allows two devices to communicate with each other. The address is responsible for choosing which device will receive the data. The value of address contains 5 letters which give us the ability to generate many different addresses. We have only one receiver and for thus we will choose one address only.

After that, the initializing of the radio object is coming. The address will be set and of course here is the receiver address will write. Then set the level of power. There are two different levels of power amplifier Max and Min. we chose the maximum to use all the range of the model. Next, we have a function which sets module as a transmitter or receiver [29].

The readings are transmitting in a structure (Structure is a collection of variables of different data types under a single name) form. Where we but all the reading which are temperature, humidity, wind, light, and rain in one structure. This structure represents one-time unit readings. Then we send this package of data to the address we assigned before.

The system will access the sensors every 5 seconds to collect the data from the weather that mean the system will be idle for 5 seconds.

4.3 INDOOR STATION SOFTWARE

The indoor station is responsible for receiving the data from the outdoor station. Also, it is responsible to make a decision depending on the data. As we start before by including the required libraries, in this part of software we used all the mentioned before libraries in the first part of the software in adding to Wire and LiquidCrystal_I2C. These two libraries are using for control and access the LCD screen.

After including the required libraries, the next step is setting pin 7 and 8 for the CE (Chip Enable) and CSN (Chip Select Not) for the transceiver. The transceiver will use the same settings which used in the transmitter part but the function which sets module as transmitter not receiver will set as a receiver. The indoor station will receive the structure data sent by the outdoor station.

4.3.1 Creating the Decision Tree

The indoor station includes the implementing of the decision tree and provides a decision for playing or not. For building our decision tree, we used the playing game dataset. It is famous and very used in most machine learning solutions. It gives a yes or no decision depending on the conditions of the weather. The table below shows the playing game dataset [30]:

Table 4.1: Playing Dataset

Windy	*Temp Numeric*	*Temp Nominal*	*Humidity Numeric*	*Humidity Nominal*	*Outlook*	*Play*
FALSE	83	hot	86	high	overcast	yes
TRUE	64	cool	65	normal	overcast	yes
TRUE	72	mild	90	high	overcast	yes
FALSE	81	hot	75	normal	overcast	yes
FALSE	70	mild	96	high	rainy	yes
FALSE	68	cool	80	normal	rainy	yes
TRUE	65	cool	70	normal	rainy	no
FALSE	75	mild	80	normal	rainy	yes

TRUE	71	mild	91	high	rainy	no
FALSE	85	hot	85	high	sunny	no
TRUE	80	hot	90	high	sunny	no
FALSE	72	mild	95	high	sunny	no
FALSE	69	cool	70	normal	sunny	yes
TRUE	75	mild	70	normal	sunny	yes

To build the decision tree, first we calculated the entropy and information gain for each feature in the dataset. The dataset contains 4 features (outlook, temp, humidity, and windy) to figure out which attribute that best classify the training data and the equation (4.1) for computing entropy.

$$H(S) = \sum_{c \in C} - p(\mathrm{c}) \log_2 p(c) \quad (4.1)$$

Where S is the current dataset for which entropy is being calculated. C {yes, no} is the set of classes in S. P(c) is the proportion of the number of elements in class c to the number of elements in set S.

The equation (4.2) of information gain depending on the entropy will be in this form:

$$IG(A.S) = H(S) - \sum_{c \in C} - p(t)\, \mathrm{H}\, p(t) \quad (4.2)$$

Where the term H(S) is referred to the entropy of the dataset S. T refer to the subsets which splitting from the S. p(t) is the ratio of the elements in t to the total elements in the dataset. Finally, the entropy of the subset t denoted by H(t).

The steps to apply the above equations until finding the suitable attribute for building the decision tree can be summarized in these 4 points:

1. For the classification attribute, we calculate the entropy.
2. Each attribute will process in this way:
    1. Calculate entropy for each group in the attribute with corresponding categorical.
    2. Calculate the average entropy.
    3. Find the gain.

3. After finishing all attribute: choosing the highest gain.
4. Repeat until building the decision tree.

For calculate the entropy for data-set we will take the decision column and applying the equations. We have 14 instance 9 of them classified as yes and 5 classified as no:

$$\text{Play-yes} = -(9/14)*\log_2(9/14) = 0.41$$

$$\text{Play-no} = -(5/14)*\log_2(5/14) = 0.53$$

$$H(S) = \text{Play-yes} + \text{Play-no} = 0.94$$

For every feature we will calculate the entropy and information gain:

$$E(\text{Outlook} = \text{sunny}) = -2/5 \log 2/5 - 3/5 \log 3/5 = 0.971$$

$$E(\text{Outlook} = \text{overcast}) = -1 \log 1 - 0 \log 0 = 0$$

$$E(\text{Outlook} = \text{rainy}) = -3/5 \log 3/5 - 2/5 \log 2/5 = 0.971$$

Average entropy information for outlook:

$$I(\text{Outlook}) = 5/14 * 0.971 + 4/14 * 0 + 5/14 + 0.971 = 0.693$$

The gain of the outlook attribute will be:

$$\text{Gain}(\text{Outlook}) = E(S) - I(\text{Outlook}) = 0.94 - 0.693 = 0.247$$

The next attribute is the windy or not:

$$E(\text{windy} = \text{false}) = -6/8 \log 6/8 - 2/8 \log 2/8 = 0.811$$

$$E(\text{windy} = \text{true}) = -3/6 \log 3/6 - 3/6 \log 3/6 = 1$$

Average entropy information for outlook:

$$I(\text{windy}) = 8/14 + 0.811 + 6/14 * 1 = 0.892$$

The gain of the outlook attribute will be:

$$\text{Gain}(\text{windy}) = E(S) - I(\text{windy}) = 0.94 - 0.892 = 0.048$$

Similarity we can calculate for other two attributes (Humidity and Temp). Depending on the entropy of the dataset which equals 0.940, the results of each attribute are shown below:

Table 4.2: Entropy for each attribute

Outlook	Temperature
Info: 0.693 Gain: 0.940 – 0.693 = 0.247	Info: 0.911 Gain: 0.940 – 0.911 = 0.029
Humidity	**Windy**
Info: 0.788 Gain: 0.940 – 0.788 = 0.152	Info: 0.892 Gain: 0.940 – 0.892 = 0.048

So our root node is Outlook because we picking the highest result.

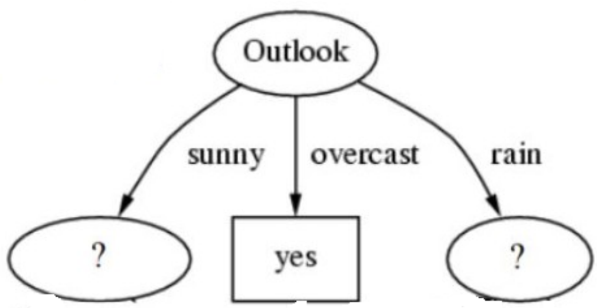

Figure 4.2: Root of the Weather Decision Tree

After repeating the same steps for the sub-trees, we get the following values of gain:

Gain (Temperature) = 0.571

Gain (Humidity) = 0.971

Gain (windy) = 0.02

So the humidity is selected as a next level of the decision tree. Figure 4.3 showing the tree in this stage. In this stage, the tree contains three paths. The first one when the weather is overcast and in this situation the decision will be yes always. The next path is when the weather is sunny. In this path, the decision is depending on the humidity. If the humidity is high then the decision will be no otherwise the decision will be yes.

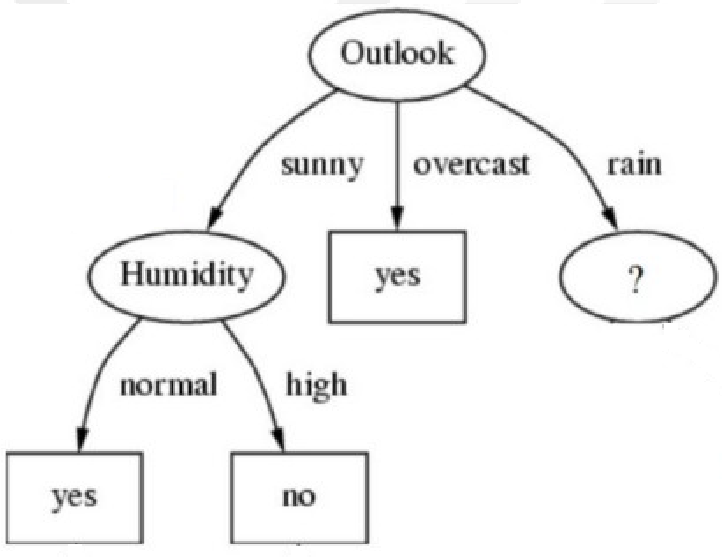

Figure 4.3: Next Level of the Weather Decision Tree

After repeating these processes, finally we will get the final decision tree as showing in figure 4.4 below:

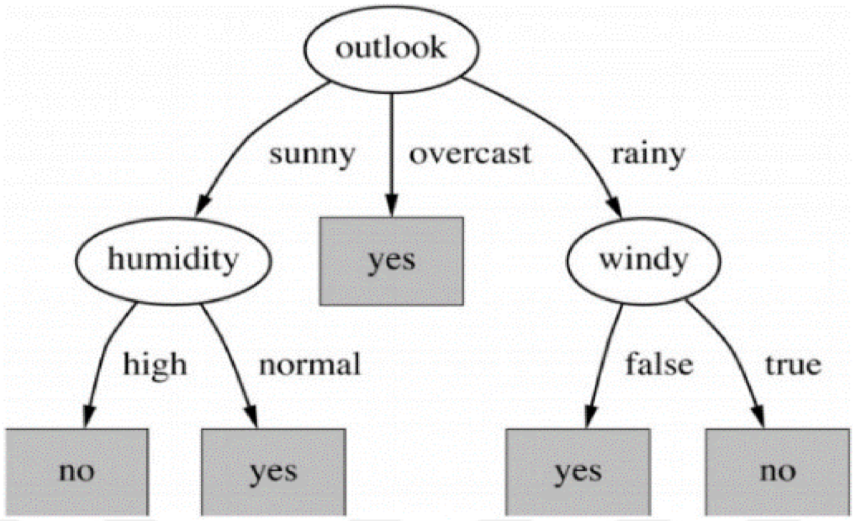

Figure 4.4: Final Weather Decision Tree

4.3.2 Implementing the Tree

After creating the decision tree, now we will implement it in the indoor software to be able to make a decision depending on real-time data coming from the sensors of the outdoor station.

The software will start by checking each path in the decision tree. First will start checking the outlook which represents by the light and rain sensors.

From the datasheet of the light sensor, it gives the value of the lux for sunlight between 300 and 600 with a 100K resistor. So if the light level is less or equal to 300 that means the outlook is overcast. Otherwise, the output is sunny. The rain sensor starting detecting rain when the value of voltage which receiving in the sensor pin reduce to 700 or less. So our rain decision will depend on this condition.

When the weather is sunny, our decision tree will move to the next weather condition which is the humidity. The humidity of the weather converted directly by the DHT library which provides us with a ratio of weather humidity. The decision tree will check the humidity ratio and depending on the dataset the ratio of humidity will consider as high if it is more or equal to 85%. Otherwise, it considers as low. If it is sunny and the humidity ratio high then the decision will be no. on the other hand, if it is sunny and the humidity low the decision will be yes.

The next path in our decision tree is the rain condition. As said before the rain sensor starting rain detection when the value of pin less than 700. In this path, the rain and the wind determine the decision. The wind sensor is a dc motor when it turns by the wind it will generate a signal start from 1 to 1024. The analog pin will receive this signal. The threshold we choose for the wind is 10 because the dc motor hard to turn and any small value will consider as a wind.

So the conditions of this path contain two statements, first the rain and second the wind. If both of these conditions are true then the decision will be no. otherwise, the decision will be yes.

5. RESULT AND DISCUSSION

In this part of the thesis, we are going to show the results that we get from the weather station and some of the image for the indoor part of the weather station during collecting the results.

Figure (5.1) showing multiple images for the LCD screen while receiving the different data from the outdoor station. Figure (5.2) showing the two possible decision in our decision tree. The data collected in random without focusing on the data at the same time or in one sequence. The aim of these figures is showing how the data are displaying on the LCD screen.

Figure 5.1: Data Showing in the LCD.

Figure 5.2: Decision Showing in the LCD.

The screen is showing the humidity ratio and the temperature measured in one screen. Then in the next screen, it is showing the light level and the detection of the rain. After that, the weather is windy or not are showing.

After that and depending on the showed data, the decision will appear on the screen. There are two decisions, the first one plays which meaning the decision is yes and the decision tree took one of the paths which contain the yes decision. The second one is no playing which means the no decision depending on the decision tree paths which led to the no decision.

Table 5.1 showing the readings that we got from the sensors when we ran it in real time with real data collected from the sensors.

Table 5.1: Sensors Readings with the corresponding decision

Rain	***Light Sensor***	***Temp Numeric***	***Humidity Numeric***	***Wind Sensor***	***Play***
0	195	17	71	0	Yes
0	199	17	55	1	Yes
1	143	19	69	0	Yes
1	159	17	74	0	Yes
1	160	18	77	0	Yes
1	128	19	65	1	No
0	528	18	54	1	Yes
0	627	17	82	0	Yes
0	508	18	93	0	No
0	580	18	93	0	No

As we see above, the sensors readings which came from the weather station contain numerical values. By using the threshold which we explained to them in the previous chapter, the decision tree will give the correct decision. The table above contains ten different situations for the weather conditions. The rain conditions contain 6 readings when there is no rain. Two of them when the outlook is overcast. The decision for this two situation is yes. The four readings come when the weather is sunny. In this case the decision depending on the ratio of humidity. Two of these reading has high humidity radio which led to no decision.

When the rain is detected, the decision tree is checking the wind condition. In our results, there is only one reading in which the weather rainy and windy, so the decision was no. otherwise, the decision is yes.

Figure (5.3), figure (5.4), figure (5.5) and figure (5.6) showing the changes and relations between light, temperature, and humidity when the decision is yes and when the decision is no.

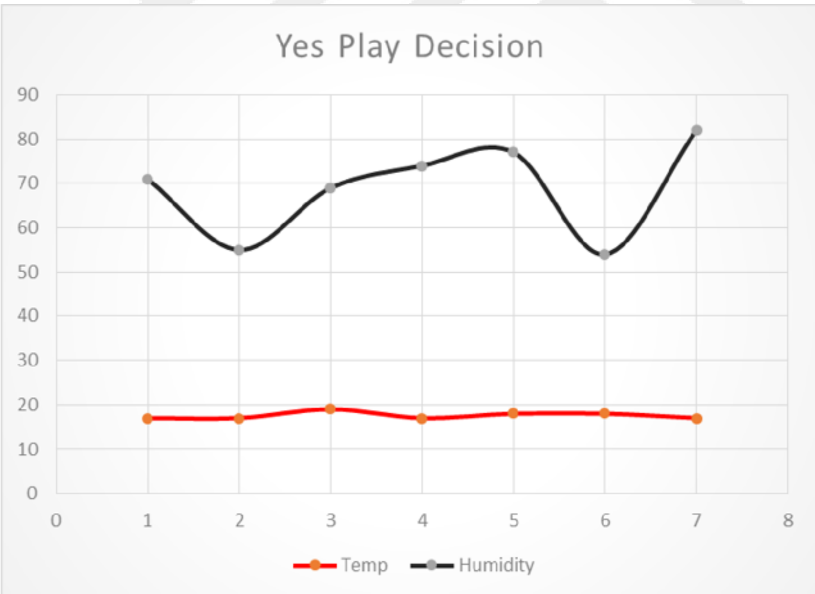

Figure 5.3: Relation between temperature and Humidity when the decision is yes.

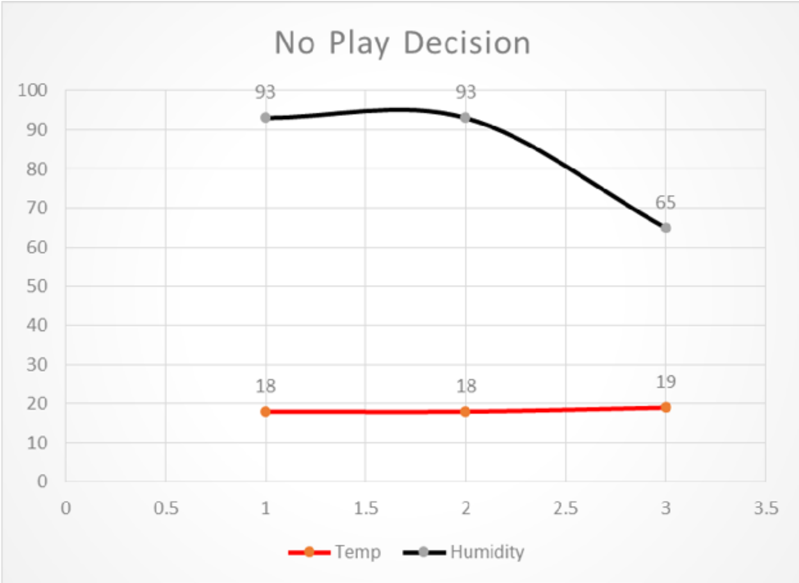

Figure 5.4: Relation between temperature and Humidity when the decision is no.

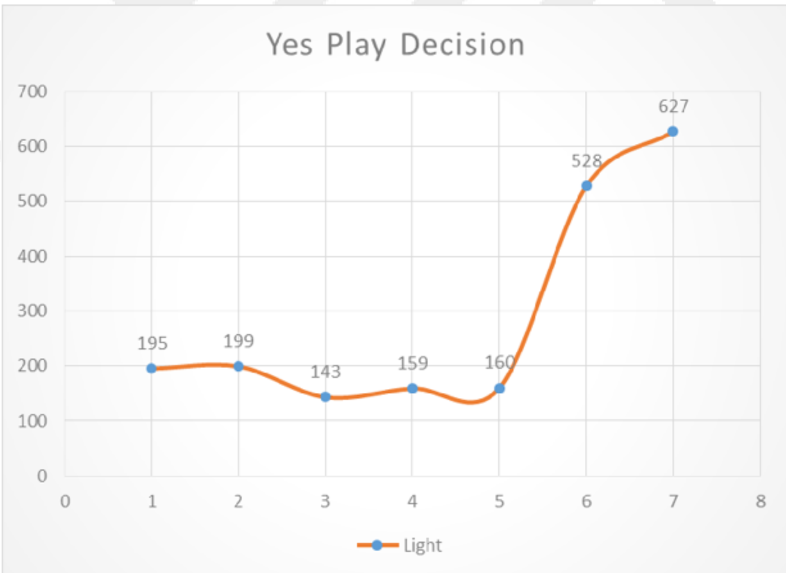

Figure 5.5: Light sensor readings when the decision is yes.

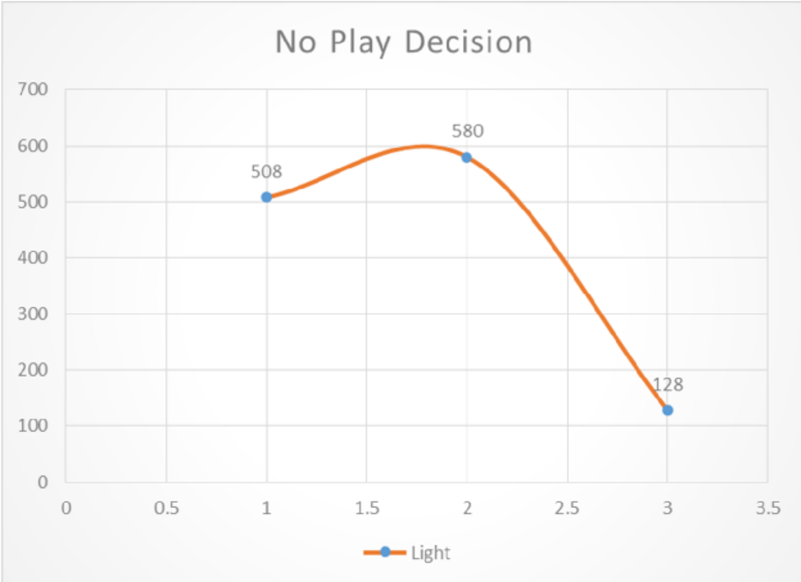

Figure 5.6: Light sensor readings when the decision is no.

6. CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

This thesis describes a weather station for making playing decision using Arduino and some sensors based on wireless NRF24L01 communication model. The used sensors are DHT11 temperature and humidity sensor, Light-Dependent Resistor as a light sensor, Rain sensor and dc motor as a wind sensor. The advantage that the incoming information is easy to understand because we are showing all the received information and the decision on the LCD screen. We used a decision tree algorithm to predicate the decision. The tree built using the famous game-playing dataset. After building the tree we used it with real-time data coming from the sensors. The weather station contains two part. First one placed outdoor and collect data depending on the sensors. The second part placed indoor and make a decision by displaying it on the LCD screen. As a result, our weather station can work directly and measuring the temperature, humidity, sunlight and able to detect rain and wind from any environment. Also, it is able to make a decision depending on the incoming data.

6.2 FUTURE WORKS

For future work, we advise using a long-range communication method to increase the range of distance for sending data. Also, we advise to make many substations from the outdoor station and placed them in a different area. The other advise is using the weather station to make predicates about the weather for incoming days not only the current time by using different techniques of machine learning.

REFERENCE

[1] Mircea Popa, and Catalin Iapa, “Embedded Weather Station with Remote Wireless Control”, 19th Telecommunications forum TELFOR, IEEE, 2011.

[2] Prashant Biradar, Sarfraz Ansari, Yashavant Paradkar, and Savita Lohiya, “Weather Prediction Using Data Mining”, International Journal of Engineering Development and Research, 2017.

[3] Diego A. Aponte-Roa, Luis Benitez Montalvan, Albert A. Espinoza, Luis Feliciano Velazquez, Ricardo Serrano, “Evaluation of a Low-Cost, Solar-Powered Weather Station for Small-Scale Wind Farm Site Selection”, IEEE, 2018.

[4] Shikha Chourasia, “Survey paper on improved methods of ID3 decision tree Classification”, (IJSR) International Journal of Scientific and Research Publications, Volume 3, December, 2013.

[5] Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali, “A comparative study of decision tree ID3 and C4.5”, (IJACSA) International Journal of Advanced Computer Science and Applications, 2014.

[6] Himani Sharma and Sunil Kumar, “A Survey on Decision Tree Algorithms of Classification in Data Mining”, International Journal of Science and Research (IJSR), 2015.

[7] Lior Rokach and Oded Maimon, “Data Mining With Decision Trees, Theory and Applications”, World Scientific Publishing Company, 2015.

[8] Jian-Ming Li, Lei Han, Shu-Yong Zhen, and Lun-Tao Yao, “Assessment Of Automatic Weather Station Operating Quality Based On Fuzzy AHP”, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, July, 2010.

[9] Arpita Ghosh, Abhay Srivastava, Atul Patidar, C.Sandeep, and Shanthi Prince, “Solar Powered Weather Station And Rain Detector”, Texas Instruments India Educators Conference, 2013.

[10] Rajesh Kumar, “Decision Tree for the Weather Forecasting”, International Journal of Computer Applications, 2013.

[11] E. Kanagaraj, L. Kamarudin, A. Zakaria, R. Gunasagaran, and A. Shakaff, “Cloud-based Remote Environmental Monitoring System with Distributed WSN Weather Stations”, IEEE, 2015.

[12] Mary Nsabagwa, Maximus Byamuka, Julianne Sansa, and Richard Okou, "Network Densification Strategies for Automatic Weather Stations: Challenges and Opportunities for Uganda", International Information Management Corporation, 2016.

[13] Frida Bafverfeldt, "Building an Arduino based weather station and connecting it as a slave to a control system", Uppsala University Publications, 2016.

[14] M. Farhat, M. Abdul-Niby, M. Abdullah, and Abdelbaset N., "A Low Cost Automated Weather Station for Real Time Local Measurements", Engineering, Technology & Applied Science Research 2017.

[15] Gabriel P., Alejandro C., Daniel M., Francisco E., and Dora C., "Design of a low cost weather station for detecting environmental changes", Espacios, SCImago Journal & Country Rank, 2017.

[16] Aris Munandar, Hanif Fakhrurroja, etc. "Design of Real-time Weather Monitoring System Based on Mobile Application using Automatic Weather Station", 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), 2017.

[17] Turner M. Bumbary, "Utilizing a Network of Wireless Weather Stations to Forecast Weather in Developing Countries", IEEE Integrated STEM Conference (ISEC), 2017.

[18] https://weatherstationdepot.com/different-types-weather-stations.

[19] S. W. Loke, "The Internet of Flying-Things: Opportunities and challenges with airborne fog computing and mobile cloud in the clouds", Dept. Compute. Sci. Inf. Technol., La Trobe Univ., Melbourne VIC, Australia, Jul. 2015, accessed on Jun. 4, 2016.

[20] https://www.oreilly.com/library/view/arduino-atechnical/9781491934319/c h04.html

[21] J. Rogers, "How drones are helping the Nepal earthquake relief effort", FoxNews.Com. Apr. 2015.

[22] R. Debra and C. McCullough, "Unmanned Aircraft Systems (UAS)", Guidebook in Development. Accessed on Mar. 7, 2016.

[23] B. J. O'Brien, D. G. Baran, and B. B. Luu, "Ad hoc networking for unmanned ground vehicles: Design and evaluation at command, control, communications, intelligence, surveillance and reconnaissance on-the-move," ARL, Adelphi, MD, USA, Army Res. Lab. Tech. Rep. ARL-TR-3991, Nov. 2006.

[24] Fahad Yassir Al Bazzaz, Hasan Huseyin Balık "Far Distance Uav Control and Object Detection Using IoT Network and Embeded Systems". Yildiz Techniques University, Turkey, 2018.

[25] http://blog.circuits4you.com/2016/04/interfacing-rain-sensor-to-arduino.html.

[26] Dipak V. and D. Sayyad, "Weather Monitoring Station: A Review", Int. Journal of Engineering Research and Application, 2016.

[27] Abid Rahim, Zeeshan Ali, Raushan Bharti, Sabeel N.S.P. and Ramya B.K., "Design and Implementation of a Low Cost Wireless Sensor Network using Arduino and nRF24L01", International Journal of Scientific Research Engineering & Technology (IJSRET), 2016.

[28] https://www.arduino.cc/en/reference/libraries

[29] https:// howtomechatronics .com /tutorials /Arduino /arduino-wireless-communication-nrf24l01-tutorial/

[30] https://medium.Com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithm -b93975f7a1f1

APPENDIX A

TRANSMITTER DEVICE CODE

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include "dht.h"
#define dht_apin A0 // Analog Pin is connected to dht sensor
dht DHT;
RF24 radio(7, 8); // CE, CSN
const byte address[6] = "00001";

typedef struct
{
  int H;
  int T;
  int L;
  int R;
  int Win;
}
W;
W weadat;
int lightLevel;
void setup()
  {
  pinMode(A3,INPUT);
  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MAX);
```

```
  radio.stopListening();
  }

void loop()
{
   DHT.read11(dht_apin);
   lightLevel = analogRead(A1);
   weadat.H=DHT.humidity;
   weadat.T=DHT.temperature;
   weadat.L=lightLevel;
   weadat.R=analogRead(A5);
   weadat.Win=analogRead(A3);
   if (weadat.Win > 10)
   {
     weadat.Win=1;
   }
   else
   {
     weadat.Win=0;
   }
   radio.write(&weadat, sizeof(weadat));
   weadat.Win=0;
   delay(5000);//Wait 5 seconds before accessing sensor again.
}
```

APPENDIX B

RECEIVER DEVICE CODE

```
#include <Wire.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
RF24 radio(7, 8); // CE, CSN
const byte address[6] = "00001";

typedef struct{
  int H;
  int T;
  int L;
  int R;
  int Win;
}
W;
W weadat;

void setup()
{
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Weather Station");
```

```
  delay(1000);
  radio.begin();
  radio.openReadingPipe(0, address);
  radio.setPALevel(RF24_PA_MAX);
  radio.startListening();
}
void loop()
{
   if (radio.available()) {
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Receving Data");
   delay(1000);
   radio.read(&weadat, sizeof(weadat));
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Humidity = ");
   lcd.setCursor(11,0);
   lcd.print(weadat.H);
   lcd.setCursor(0,1);
   lcd.print("Temperature = ");
   lcd.setCursor(14,1);
   lcd.print(weadat.T);
   delay(1000);
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Light = ");
   lcd.setCursor(10,0);
```

```
lcd.print(weadat.L);
lcd.setCursor(0,1);
if (weadat.R < 700){lcd.print("Rain Detected");}
else{lcd.print("no Rain");}
delay(1000);
lcd.clear();
lcd.setCursor(0,0);
if (weadat.Win == 1)
{lcd.print("Weather Windy");}
else {lcd.print("Not Windy");}
delay(1000);

//###### Decesion Tree ##########
if (weadat.L>300 && weadat.H>=85)
{ lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("No Playing");
}
else if (weadat.R < 700 && weadat.Win == 1)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("No Playing");
}
else if(weadat.L>300 && weadat.H<85)
{
  lcd.clear();
  lcd.setCursor(0,0);
```

```
      lcd.print("Play");
    }
    else if (weadat.L < 300)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Play");
    }
    else if (weadat.R < 700 && weadat.Win == 0)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Play");
    }
    //##############################################
    delay(1000);
  }
  lcd.clear();
}
```