Design and Development of a Home Automation System based on IoT

Project Team

Sl. No.	Reg. No.	Student Name
1	16ETEC004054	Raghavendra P H
2	16ETEC004047	Niveditha J
3	16ETEC004046	Nikhil C
4	16ETEC004032	M Rahul Naidu

Supervisor: Mr. Bharath Kumara
Co-Supervisor: Ms. Monica M

May 2019

FACULTY OF ENGINEERING AND TECHNOLOGY
M. S. RAMAIAH UNIVERSITY OF APPLIED SCIENCES
Bengaluru -560 054

FACULTY OF ENGINEERING AND TECHNOLOGY

FET

Certificate

This is to certify that the Project titled "Design and Development of a Home Automation based on IoT" is a bonafide work carried out in the Department of Electronic and Communication Engineering by Mr.Raghavendra P H (16ETEC004054), Ms. Niveditha J (16ETEC004047), Mr.Nikhil C (16ETEC004046) and Mr. M Rahul Naidu(16ETEC004032) in partial fulfilment of requirements for the award of B. Tech. Degree in Electronic and Communication of Ramaiah University of Applied Sciences.

May – 2019

Asst.Prof. Bharath Kumara
Supervisor

Dr. Raghavendra Venkatesh Kulkarni
Head – Dept. of ECE

Dr. M. Arulanantham
Dean-FET

Declaration

Design and development of a Home Automation based on IoT

The project work is submitted in partial fulfilment of academic requirements for the award of B. Tech. Degree in the Electronic and Communication Engineering of the Faculty of Engineering and Technology of M. S. Ramaiah University of Applied Sciences. The project report submitted herewith is a result of our own work and in conformance to the guidelines on plagiarism as laid out in the University Student Handbook. All sections of the text and results which have been obtained from other sources are fully referenced. We understand that cheating and plagiarism constitute a breach of University regulations, hence this project report has been passed through plagiarism check and the report has been submitted to the supervisor.

Sl. No.	Reg. No.	Student Name	Signature
1.	16ETEC004054	Raghavendra P H	
2.	16ETEC004047	Niveditha J	
3.	16ETEC004046	Nikhil C	
4.	16ETEC004032	M Rahul Naidu	

Date: 15 May 2019

Acknowledgements

We would like to express our since gratitude to our mini project supervisor, Mr. Bharath Kumara and co-supervisor, Ms. Monica M for guiding us through the entirety of this project. We would also like to thank our Electronic and Communication Engineering HOD Dr. Raghavendra Venkatesh Kulkarni and the Dean of the Faculty of Engineering and Technology Dr.Arulanantham for providing us an opportunity to create this project as a part of the curriculum. Finally, we extend our appreciation to our parents for their support and encouragement they provided us during the course of project.

Abstract

Home automation systems using Internet of Things (IoT) help reduce power consumption through optimal control and minimize the necessity of human intervention. IoT eases handling of household functions and automatically controls them.

The major issue with most IoT systems are that they lack data security measures. This is due to the fact that usually companies that develop such systems focus mostly on marketing and delivery of the product rather than actually improving the network and data security of these systems. Implementation of complex data encryption algorithms prove cumbersome as the data size transmitted by the sensors is small in size. The aim of this project is to provide a data security feature to the IoT based home automation systems.

This project addresses this issue by implementing an innovative encryption technique that specifically targets IoT systems. This project features three nodes connected to a common online database, called the Firebase. Two of these nodes are responsible for the data security feature. The first node collects data from sensors placed all around the house. It then encrypts and sends the data in real time to the database. The second node collects this data from the database, decrypts, processes and then expresses it at the actuators. The system also has a feature wherein if the network is hacked and data is rewritten, this manipulation of data is detected and an alert is sent to the resident/user.

This project also provides a water management feature to the home automation system via the third node. The third node measures the amount of water in over-head tank and switches on the pump when and if required. All of the above mentioned features can be customized by the user which can be accessed through an android application.[1]

List of Tables

List of Figures

Appendix - B

Abbreviation and Acronyms

AC	Air conditioner
CFL	Compact Fluorescent Lamp
DC motor	Digital Current motor
DHT	Digital Humidity and Temperature
IoT	Internet of Things
LDR	Light Dependant Resistor
LED	Light Emitting Diode
LPG	Liquefied Petroleum Gas
Node MCU	Node Micro Controller Unit
PIR	Passive Infrared
RFID	Radio Frequency Identification

Table of Contents

1. Introduction

Preamble to the Chapter:

This chapter deals with the introduction to the project which includes the current trend that is IoT and its disadvantages.

1.1 Introduction

Home automation using IOT denotes the concept of using mobile devices to control basic functions and applications of one's house through various control systems via the internet. The advantages of this type of organizing one's house is two-fold, one it works with minimal or no human intervention and hence it's termed 'automation', and the second advantage is that it transforms the house into a smart home, i.e., saves electric power as well as human energy. One of the implied features of Home automation is a Home Security system as statistics show that more than 90% of the people who opt for home automation do so for the purpose of security .[2]

1.2 Motivation

The major disadvantage of IoT systems as of now is the issue of data security. IoT manufacturers are more eager to produce and deliver their devices as fast as they can, without giving security too much of a thought. Most of these devices and IoT products don't get enough updates while, some don't get updates at all. There have been multiple cases where in hackers have been able to gain access to the network and have taken control. Some examples include brakes of cars being disabled which has led to serious injuries and even death, people being locked out of their own homes, etc. The other reason for not including encryption is that the encryption methods currently available, although complex, cannot be used for data transfer in case of IoT because of the fact that the data being sent from one node to the other through the internet is simply too small for any existing encryption scheme to protect it effectively.[3]

Thus this project is aimed at providing a new method of encryption scheme that is specifically targeted at IoT systems which can not only ensure the safety of the data being transmitted but also alert when the data is being manipulated by a hacker who manages to breach the network.

2. Background Theory

Preamble for the chapter:

This chapter deals with the sensors used in this particular project along with their working and specifications, and case studies as to how lack of data security in IoT systems have

2.1 Sensors:

2.1.1 PIR Sensor

PIR sensors are used to sense motion. It is used to detect whether there is some sort of movement within its range. It is one of the most commonly found in appliances and gadgets used at home or in commercial areas. Figure 2.1 shows a PIR sensor.

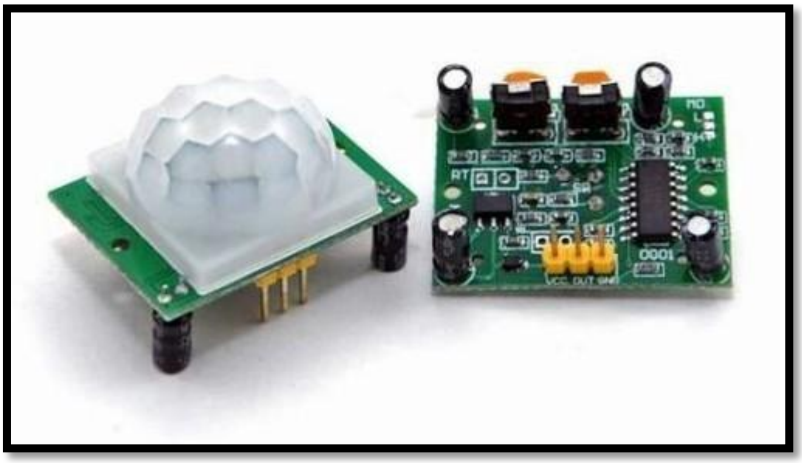


Figure 2.1 : PIR Sensor

The module actually consists of a Pyroelectric sensor which generates energy when exposed to heat. That means when a human or animal gets into the range of the sensor it detects movement due to the radiation emitted. The module also consists a specially designed cover named Fresnel lens, which focuses the infrared signals onto the pyroelectric sensor.

PIRs have adjustable settings and have a header installed in the 3-pin ground/out/power pads.

The PIR sensor has 3 pins, GND, OUT and Vcc, whose connections are as follows:

GND : Connected to GND on Arduino Board

OUT : Connected to an Arduino digital pin

Vcc : Connected to 5V on Arduino Board

2.1.2 Water Sensor

Water sensor brick is designed for water detection, which can be widely used in sensing rainfall, water level, and even liquid leakage. Figure 2.2 shows a water sensor.

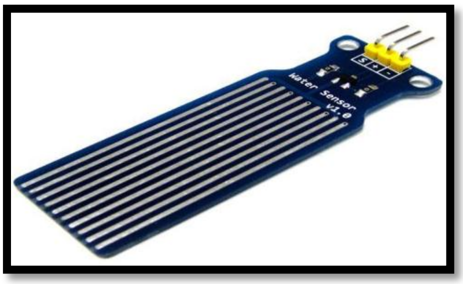


Figure 2.2 : Water Sensor

It can be used to detect the presence, the level, the volume and/or the absence of water. The sensor has an array of exposed traces, which read LOW when water is detected.

Water sensor has three terminals - S, V_{out}(+), and GND (-). The pins are connected as follows –

- V_{out}(+) : connected to +5V on the NodeMCU board.
- S: Connected to a digital pin number on NodeMCU board.
- GND: Connected to GND on NodeMCU board.

2.1.3 Ultrasonic Sensor

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module. Figure 2.3 shows an ultrasonic sensor.

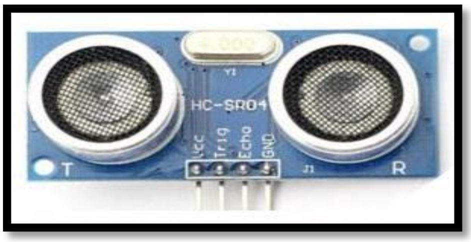


Figure 2.3 : Ultrasonic Sensor

Technical Specifications of Ultrasonic:

- Power Supply − +5V DC
- Quiescent Current − <2mA

- Working Current − 15mA
- Effectual Angle − <15°
- Ranging Distance − 2cm – 400 cm/1″ – 13ft
- Resolution − 0.3 cm
- Measuring Angle − 30 degree

The Ultrasonic sensor has four terminals - +5V, Trigger, Echo, and GND connected as follows −

- +5V pin: Connected to +5v on Arduino board.
- Trigger: Connected to a digital pin on the NodeMCU board.
- Echo: Connected to a digital pin on NodeMCU board.
- GND: Connected with GND on Arduino.

2.1.4 MQ2 Gas Sensor

MQ2 Gas Sensor reads the concentration of smoke, LPG, etc present in the surrounding area and converts it to an analog output voltage

The MQ-2 smoke sensor is sensitive to smoke and to the following flammable gases:

- LPG
- Butane
- Propane
- Methane
- Alcohol
- Hydrogen

The resistance of the sensor is different depending on the type of the gas. The smoke sensor has a built-in potentiometer that allows adjusting of the sensor sensitivity according to how accurately the gas concentration is to be detected. Figures 2.4 shows an MQ-2 gas sensor and Figure 2.5 gives the detail of the pins in an MQ-2 Gas sensor.

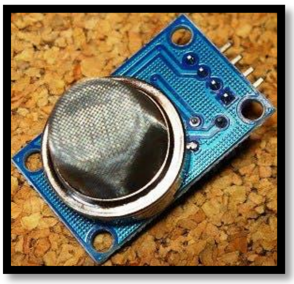


Figure 2.4 : MQ-2 Gas Sensor

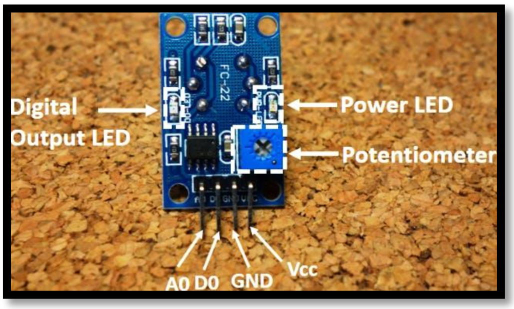


Figure 2.5 : Sensor Pin-out details

The voltage that the sensor gives as the output is proportional to the concentration of smoke/gas.

The output can be an analog signal (A0) that can be read with an analog input of the Arduino or a digital output (D0) that can be read with a digital input of the Arduino.

- The MQ-2 sensor has 4 pins, out of which only 3 can be used.

- D0: Connected to a Digital pin on NodeMCU Board
- GND: Connected to GND on NodeMCU Board
- VCC: Connected to 5V on NodeMCU Board

2.1.5 DHT11

A DHT11 sensor is used to detect temperature and humidity

There are two different versions of the DHT11. One type has four pins, and the other type has three pins and is mounted to a small PCB. The PCB mounted version is preferred because it includes a surface mounted 10K Ohm pull up resistor for the signal line.

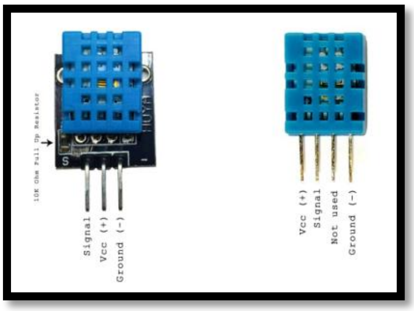


Figure 2.6 : DHT11 sensor with 3pins and 4pins

Working of DHT11 Sensor:

The DHT11 detects water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

The DHT11 measures temperature with a surface mounted NTC temperature sensor (thermistor) built into the unit

The ranges and accuracy of the DHT11 are as follows:

- Humidity Range: 20-90% RH
- Humidity Accuracy: ±5% RH
- Temperature Range: 0-50 °C
- Temperature Accuracy: ±2% °C
- Operating Voltage: 3V to 5.5V

2.1.6 LDR(Light Dependent Resistor)

LDRs are made from semiconductor materials to enable them to have their light-sensitive properties. An LDR is used in order to detect the intensity of light

The LDR is a special type of resistor that allows higher voltages to pass through it (low resistance) whenever there is a high intensity of light, and passes a low voltage (high resistance) whenever it is dark. Figure 2.7 shows an LDR

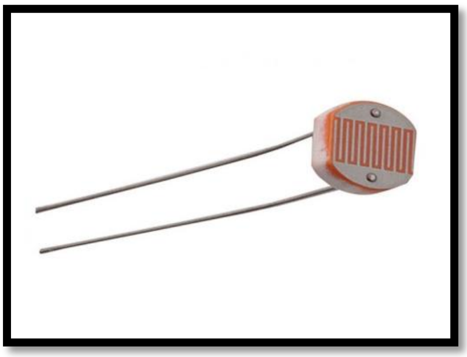


Figure 2.7 : LDR

Working of an LDR:

LDRs or PHOTO RESISTORS works on the principle of “Photo Conductivity”. According to this principle, whenever light falls on the surface of the LDR (in this case) the conductance of the element increases or in other words, the resistance of the LDR falls when the light falls on the surface of the LDR. This property of the decrease in resistance for the LDR is achieved because it is a property of semiconductor material used on the surface.

LDR gives out an analog voltage when connected to VCC (5V), which varies in magnitude in direct proportion to the input light intensity on it, and the voltage value across the LDR is read as a value from 0-1023 by the Arduino. When there is sufficient light in its environment or on its surface, the converted digital values read from the LDR through the Arduino will be in the range of 800-1023.

2.2 Data protection:

There's still a long way to go with IoT security and, unfortunately, precautions are rarely taken until disaster strikes. There has yet to come a method that harnesses all the benefits and conveniences that IoT brings while minimizing the potential for harm. That way, consumers will no longer have to compromise security for convenience.

A smart household thermostat, for example, doesn't just collect data about users' home temperature preferences. It also collects data about when users are and aren't home, as well as the number of people living in the household. Likewise, an IoT connected car, based on user activity throughout the day, can infer personal information about its users, such as where they work, where they live, and what their shopping preferences are.

While such opportunities for data collection might seem obvious to those in the technological industry, many consumers, content with the ease and convenience of IoT, don't realize the extent to which their personal data is analyzed and used.

2.3 Case study:

2.3.1 The Jeep Hack

The IBM security intelligence website reported the Jeep hack a few years ago. In July [2015], a team of researchers was able to take total control of a Jeep SUV using the vehicle's CAN bus. By exploiting a firmware update vulnerability, they hijacked the vehicle over the Sprint cellular network and discovered they could make it speed up, slow down and even veer off the road. It's proof of concept for emerging Internet of Things (IoT) hacks: While companies often ignore the security of peripheral devices or networks, the consequences can be disastrou.[4]

Better security protocols, strategies, and standards have to be developed if the IoT revolution is to continue to deliver value to people without compromising their security and privacy.

2.3.2 The Mirai Botnet (aka Dyn Attack)

Back in October of 2016, the largest DDoS attack ever was launched on service provider Dyn using an IoT botnet. This lead to huge portions of the internet going down, including Twitter, the Guardian, Netflix, Reddit, and CNN.

This IoT botnet was made possible by malware called Mirai. Once infected with Mirai, computers continually search the internet for vulnerable IoT devices and then use known default usernames and passwords to log in, infecting them with malware. These devices were things like digital cameras and DVR players.[5]

2.4 Conclusion

The above mentioned sensors , that is PIR sensor , water sensor, ultrasonic sensor ,MQ-2 gas sensor ,DHT11 sensor and LDR are used for this project and their requirements will be explained in the further chapters. Data protection is a major concern as given in the above case studies and it must be addressed if IoT as a field has to grow and so in this project we address the same issue by adding an encryption scheme that targets IoT devices.

3. Aim and Objectives

Preamble to Chapter

In this chapter Title, Aim and Objectives of Home automation and security system based on IoT with its methods and methodology have been discussed.

3.1 Title

Design and Development of a home automation system based on IoT.

3.2 Aim

To develop security measures to secure the data in IoT systems.

3.3 Objectives

- To perform literature review on various sensors, actuators and existing systems of IoT
- To establish the automation features of the home automation system
- To design, develop and test the working of home automation system with the sensors and actuators
- To develop and test an effective encryption algorithm for securing the data in IoT systems using sensor data as inputs
- To trans-receive the encrypted data between 2 nodes

3.4 Methods and Methodology/Approach to attain each objective

Table 3.1 Methods and Methodologies

Objective No.	Statement of the Objective	Method/ Methodology	Resources Utilised
1	To perform	Referring to journals, existing	Journal papers,

	literature review on various sensors,actuators and existing systems of IoT	models, patents on IoT systems and their implementations	books
2	To establish the automation features of the home automation system	Carry out literature survey on reviewing and deciding on the apt features to be used in the project	Journals, textbooks.
3	To design, develop and test the working of home automation system with the sensors and actuators	Perform test by uploading necessary instructions to the sensors and check their performance by running the code	Arduino IDE, LDR, sensors (DHT-11, MQ2&PIR), electronic keypad

4	To develop and test an effective encryption algorithm for securing the data in IoT systems using sensor data as inputs	Constructing an algorithm taking into consideration the ineffectiveness of existing encryption schemes	Arduino IDE, NodeMCU ESP 8266.
5	To trans-receive the encrypted data between 2 nodes	Synchronizing the node with the online database using database token.	Arduino IDE, NodeMCU ESP 8266, Firebase.

4. Problem Solving

Preamble to the Chapter:

This chapter deals with a practical approach to create a system which is capable of providing data security using a suitable encryption scheme along with an easy method to detect tampering data and the interface along with the components used for automating the house.

4.1 Design and Implementation:

4.1.1. Block Diagram and circuit diagrams:-

Figure 4.1 shown below is the block diagram of the system that has been implemented.

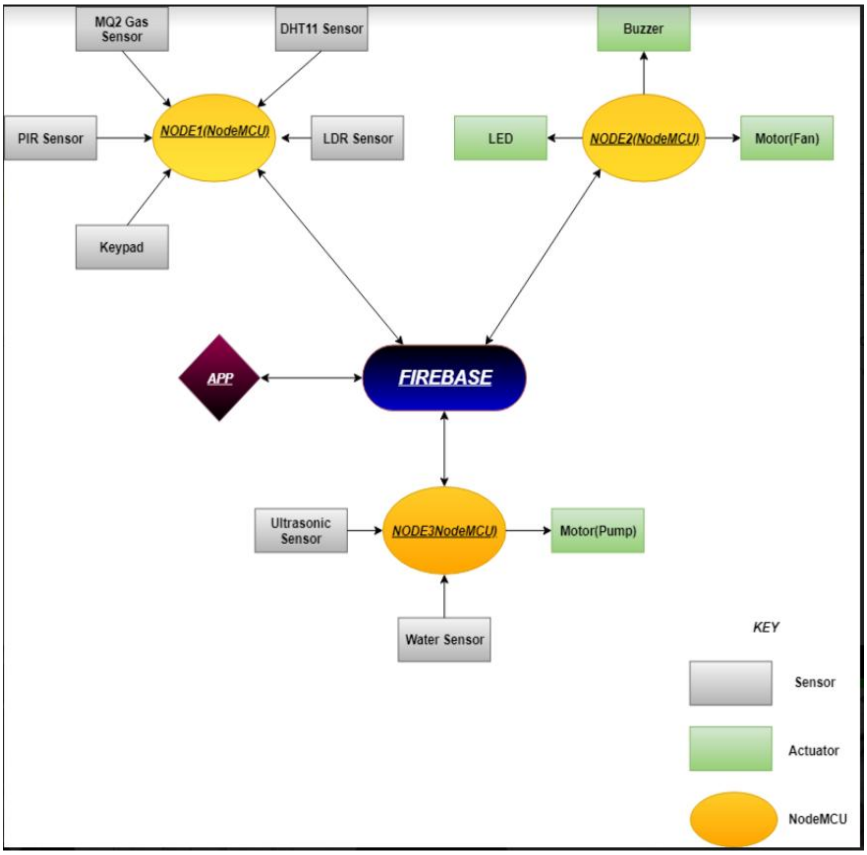


Figure 4.1 : Block diagram of the setup

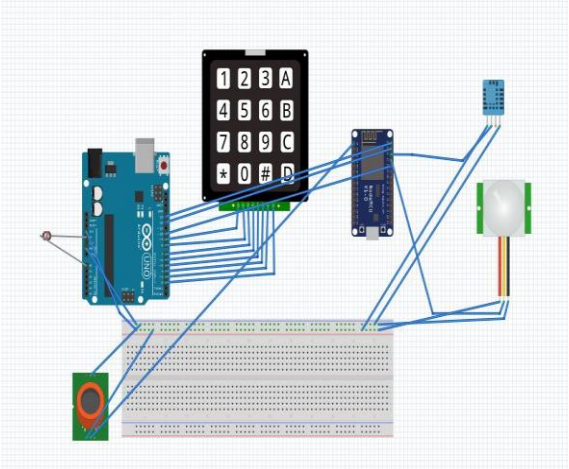


Figure 4.2 : Transmitter node

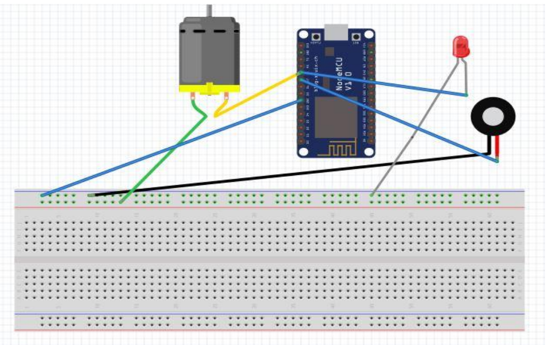


Figure 4.3 : Receiver node

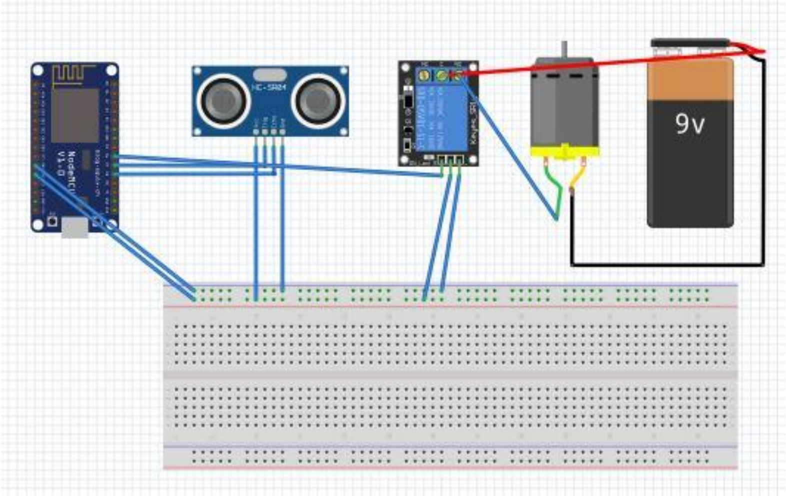


Figure 4.4 : Water management node

Figures 4.2 through 4.4 show the transmitter, receiver and the water management nodes respectively. The setup shown in Figures 4.1 through 4.4 contain a total of 3 nodes .The first node contains all the sensors from which data is collected and each bit of data is encrypted using a special type of encryption targeted at IoT devices. It collects data such as temperature and humidity from the DHT 11 sensor, motion from the PIR sensor, intensity of visible light from the LDR, a 4 x 4 keypad and an MQ-2 sensor .This node sends the sensor data to an open source real-time online database platform known as Firebase which is owned by Google. Each sensor data sent is stored under a certain tag name.

The second of the nodes collect the appropriate sensor data from the database using the tags present and decrypts this data so that it can be processed. This node contains all the actuators such as LEDs – the first of which switches on to indicate the locking/unlocking of the house in case the right pins are entered on the keypad or if someone uses the app to do the same and the second LED switches on or off depending on whether the amount of light detected by the LDR is below or above the set threshold respectively thus making it an automatic lighting feature for the house- ,buzzers are used to indicate the presence of a burglar when the house is locked and the PIR sensor detects motion or in the event of an outbreak of fire or leakage of LPG gas which is detected if the value of smoke or Butane in the air rises above the threshold ,and a motor is used to indicate the automatic switching ON/OFF of a fan or AC when the temperature measured by the DHT11 sensor goes above or below 30 degree Celsius respectively.

The last of the nodes is one that manages the water supply for the house by monitoring the level of water in the overhead tank using an ultrasonic sensor , encrypting this data and sending it through firebase. It then receives the data ,decrypts it and switches on the pump if the water in the tank goes below 40% and switches off the pump if the water in the tank goes above 90%. It also has water leakage sensors placed effectively to detect any leakage along the pipeline and a buzzer is set off in the event of a leakage

An android app is available to the user to control and customize the features provided. Each of the functionalities provided can be enabled and disabled, the water level can be monitored and the temperature and humidity of the house can be viewed from the app. Since security is one of the main features of this project the app can only be used by those who possess the username and password details and so not all who possess the app can use it to control the house.

4.2 Encryption:-

Encryption plays a huge part of data security from text messages to bank transactions to military communication. There are a horde of encryption schemes each with its own merits designed for different applications, but most of these are effective in encrypting data that has a significantly large size unlike data sent from one node to another in an IoT system and so no matter how complex the encryption scheme is it wouldn't be effective.

Ex : If the data to be sent is temperature which is equal to 25^{o}C applying an existing encryption algorithm to it would seem something like

$$\text{Encrypted data = Q\&} \qquad \text{...(1)}$$

Which isn't effective in the least. And thus arises a need to apply a new method to encrypt data that is applicable for IoT systems.

The encryption scheme used in this project has 3 stages:

4.2.1 Addition of junk data:

In order to increase the length of the data that goes out of a node junk data that is generated in a pseudo random fashion is added to the left and to the right of the actual sensor data. The junk as such has no relevance to the actual useful data to be transmitted .The useful data is separated from the junk by an underscore on both sides, this enables the node to identify and extract the actual data when it decrypts. If the temperature is 25^{o}C

Data with junk = 1jfH>=_25_P{^G& ...(2)

4.2.2 Rotation:

Rotation is one of the many kinds of encryption schemes used today .An array of all alphabets (Upper and lower case), digits and special characters is defined and each character appearing in the message has a fixed position in the array. The position number of that character is taken and rotated by 39 places and the character in the corresponding position of the rotated number replaces the original message character. The key to this encryption scheme is the order in which the characters are placed in the array, without which the cleartext cannot be obtained. For example , the actual data with the junk in Equation (2) after rotation is:

Rotated data : P$*&t.RU.9?5wJ+ ...(3)

4.2.3 Jumbling:

Although the rotated data seems protected ,a closer look would suggest to us (and to a hacker) that the character between the two dots is the actual useful data and thus the whole point of the encryption would fail. Thus arises the need for us to jumble the rotated data. This not only hides the useful data but also prevents the hacker from using the brute force method to decrypt the data because although decrypted , the data wouldn’t make any sense as its in a jumbled fashion. For the rotated data in Equation (3),

Jumbled data : 9w5+.&?U*J.tPR$ ...(4)

The decryption of the data on the receiver node happens in the exact same way but in a reverse manner.

4.3 Development of a Mobile application

Although the multitude of features provided seem to be useful, from a consumer perspective it is not always so. Thus a user friendly mobile application is built as part of the system which not only helps the user to monitor the status but also helps to control/customize the features of the system. There are different pages present in the app. They are

4.3.1 Login page

Since data security is the key aspect of this project one can expect the same when it comes to the app. The app has a login page where the user must enter the correct user name and password in order to gain access to the features provided by the app. This ensures safety of the house and its residents as not everyone who gets a hold of the app is able to access its features.

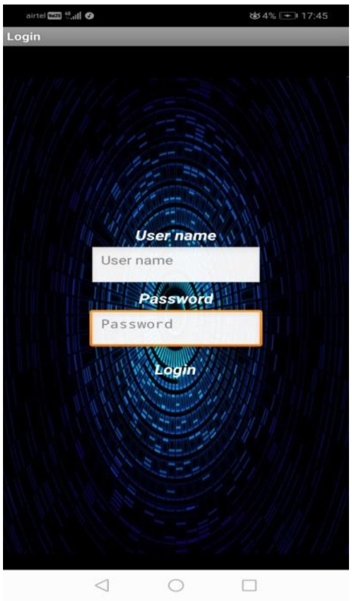


Figure 4.5 : Login Page

4.3.2 Options page

As soon as one logs into the app a multiple options page opens up from which one can select the desired action to be performed. The multiple options provided are as follows:

1. **Lock/Unlock** : This option can be selected when the user needs to lock or unlock the house using the app. Selecting this option takes one to another page where a slider is present, sliding which to the right locks the house and sliding it to the left unlocks the house.
2. **Water level monitoring** : This option is available for when the user wants the check the level of water present in the overhead tank.

3. Modes/Customization: This option is one of the key features of the app as it allows the user to exercise the flexibility of the system as he/she can choose to enable or disable any of the provided features. The customization option takes the user to another page where a column of sliders are present each with its own label .The sliders can be used to turn ON and OFF the following features

Gas detector (Smoke and LOG gas detection)
Automatic water pump
Automatic lighting
Motion sensor
Automatic fan/A.C.

4. Temperature: Choosing this option takes the user to another page where the temperature of the house is displayed in Celsius.

5. Humidity : Choosing this option takes the user to another page where the humidity level in the house is displayed in percentage.

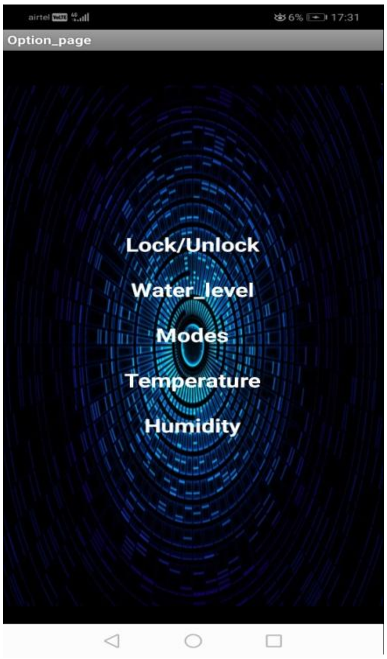


Figure 4.6 : Options Page

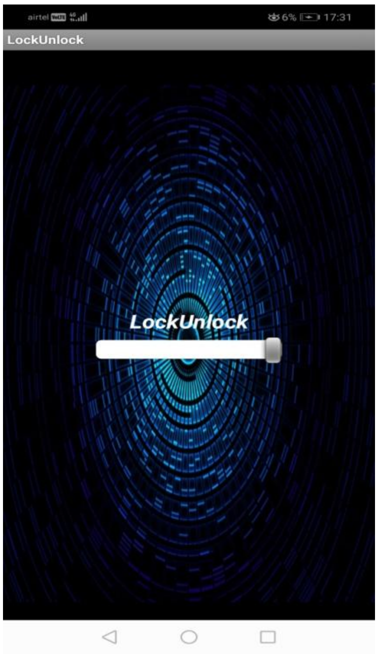


Figure 4.7 : Lock/Unlock Page

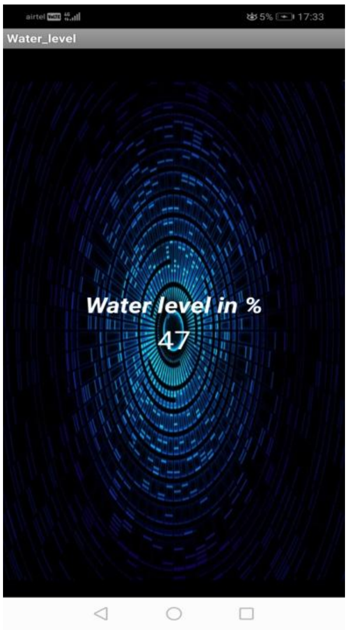


Figure 4.8 : Water level display

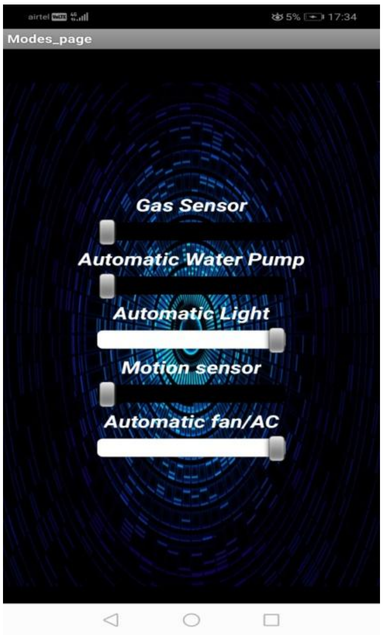


Figure 4.9 : Modes Page

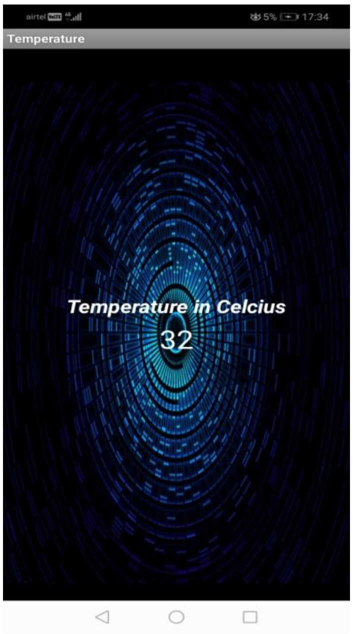


Figure 4.10 : Temperature display

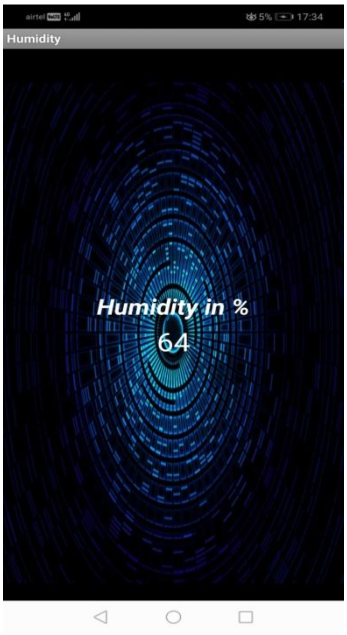


Figure 4.11 : Humidity display

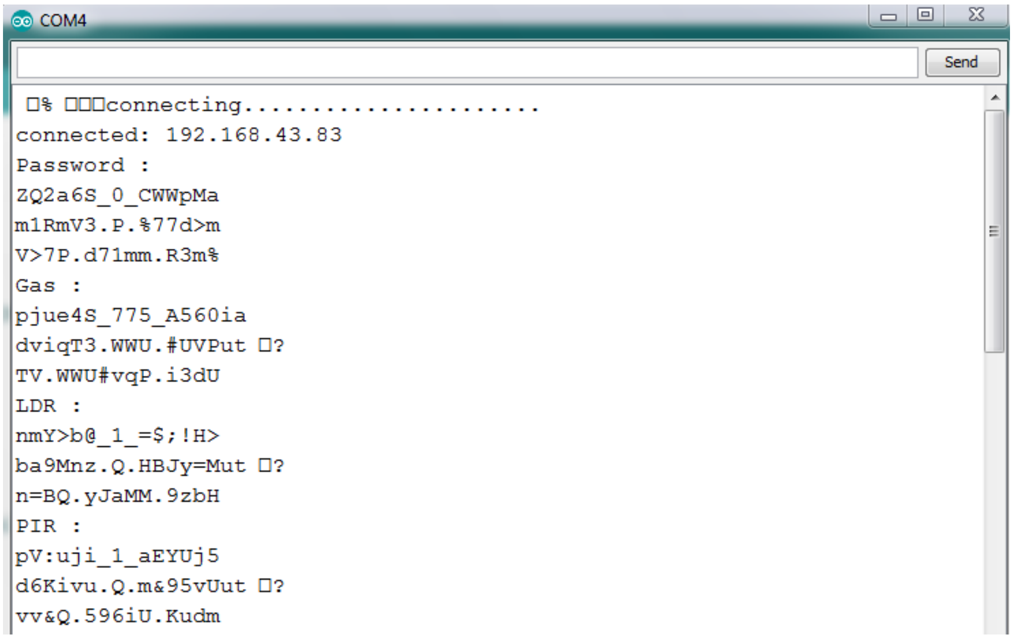
5. Results

Preamble to this chapter:

This chapter contains the results obtained by implementing the above mentioned design steps.

5.1 Results

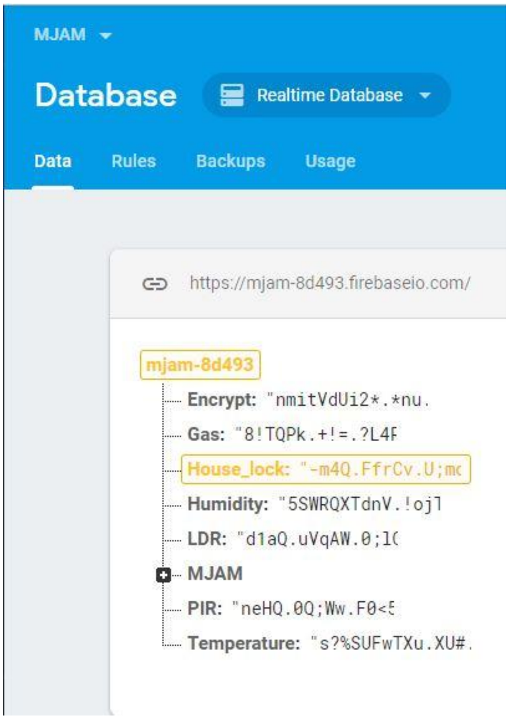
The result of the above mentioned design steps from the transmitter sending the encrypted to the receiver decrypting the data is shown. Figure 5.1 shows the serial monitor of the transmitter node.



```
 □% □□□connecting......................
connected: 192.168.43.83
Password :
ZQ2a6S_0_CWWpMa
m1RmV3.P.%77d>m
V>7P.d71mm.R3m%
Gas :
pjue4S_775_A560ia
dviqT3.WWU.#UVPut □?
TV.WWU#vqP.i3dU
LDR :
nmY>b@_1_=$;!H>
ba9Mnz.Q.HBJy=Mut □?
n=BQ.yJaMM.9zbH
PIR :
pV:uji_1_aEYUj5
d6Kivu.Q.m&95vUut □?
vv&Q.596iU.Kudm
```

Figure 5.1 : Serial Monitor of the transmitter node(node 1)

Sensors in node 1 collect and send data to the node which then encrypts it and sends the encrypt data to firebase under the names of their respective tags. The real-time database whose tags have a read-write enabled property collect and store the sent data from node 1 and retain that data until another update for the same tag is made. Figure 5.2 shows the real-time database on firebase.

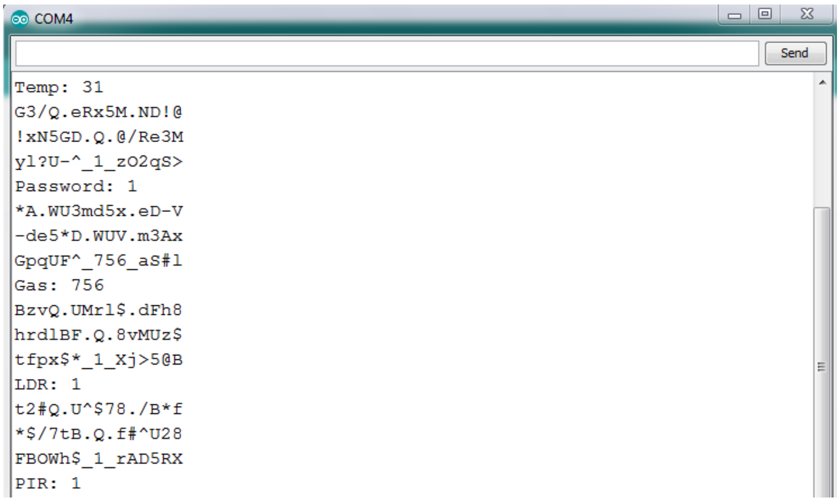


```
https://mjam-8d493.firebaseio.com/

mjam-8d493
  Encrypt: "nmitVdUi2*.*nu.
  Gas: "8!TQPk.+!=.?L4F
  House_lock: "-m4Q.FfrCv.U;mc
  Humidity: "5SWRQXTdnV.!oj1
  LDR: "d1aQ.uVqAW.0;1(
  MJAM
  PIR: "neHQ.0Q;Ww.F0<5
  Temperature: "s?%SUFwTXu.XU#.
```

Figure 5.2 : Real-time database

Node 2 collects the data from the database by accessing the corresponding tags , decrypts them, extracts the information and processes them to give the output to the serial monitor and to the actuators. Figure 5.3 shows the serial monitor of the receiver node



```
Temp: 31
G3/Q.eRx5M.ND!@
!xN5GD.Q.@/Re3M
yl?U-^_1_zO2qS>
Password: 1
*A.WU3md5x.eD-V
-de5*D.WUV.m3Ax
GpqUF^_756_aS#l
Gas: 756
BzvQ.UMrl$.dFh8
hrdlBF.Q.8vMUz$
tfpx$*_1_Xj>5@B
LDR: 1
t2#Q.U^$78./B*f
*$/7tB.Q.f#^U28
FBOWh$_1_rAD5RX
PIR: 1
```

Figure 5.3 : Serial Monitor of the Receiver node (node 2)

Figure 5.4 shows a picture of the model with all the nodes integrated and the sensors appropriately placed.

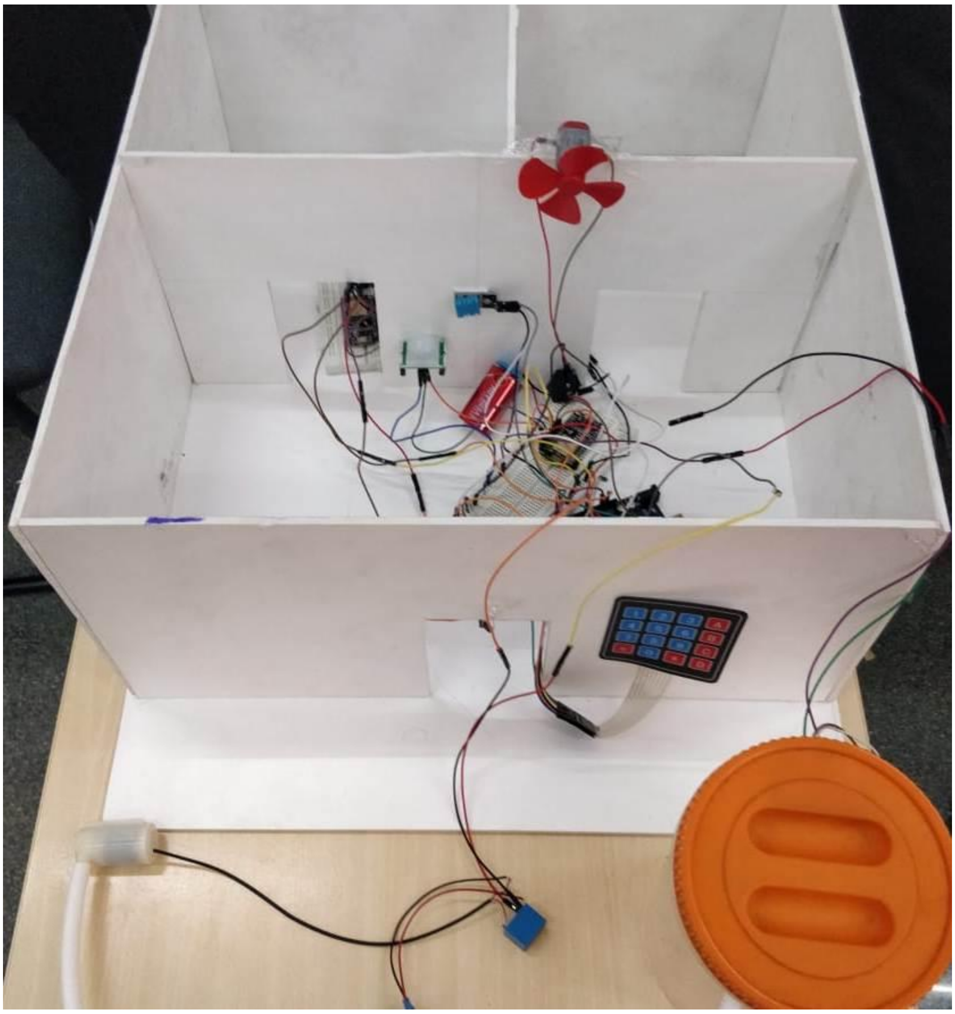


Figure 5.4 : Working model

6. Project Costing

Preamble to the Chapter:

This chapter gives a detailed report about the expenses made toward realising this project.

The software used to make this project which includes Firebase, Arduino IDE, Fritzing and MIT App inventor are open source .The expenses concentrate on only the hardware such as controller boards, sensors and actuators.

Table 6.1 Hardware cost

Sl. No.	Components/Devices	Cost(in Rupees)
1	Node MCU ESP8266 * 3	750
2	Arduino Uno	340
3	MQ2 gas sensor	115
4	DHT11 temperature and humidity sensor	75
5	PIR sensor	80
6	Keypad	90
7	LDR sensor	30
8	Ultrasonic Sensor	125
9	Water Sensor	50

10	Dc motor * 2	60
11	Buzzer * 2	30
12	Jumper Wires	100
13	Battery * 2	80
14	Relay * 2	80
15	LED * 2	20
TOTAL		2030

Although the model that is constructed is in a small scale and not in proportion to an actual house, the total cost is meagre when one considers the number of features that are provided. And in a scenario where this project is converted into a product the price of the nodes would come down drastically as there the nodes would be customized or specially designed to do one particular task whereas over here multipurpose nodes are used.

7. Conclusions and Suggestions for Future Work

Preamble to the chapter

This chapter contains the conclusion drawn from the implementation and observation of results of the project and the scope for improvement of the project for better and more effective data security.

7.1 Conclusion

This project encompasses the main features of a home automation system , that is automatic lightning ,automatic fan and automatic water level monitoring along with security features such as smoke detection, burglar alarm system and an electronic keypad lock all of which are customisable through a password protected android application. It is seen that the encryption scheme is successful as a potential hacker, without proper labels on the database would not be able to recognize the nature of the data being transmitted and even if this condition isn't satisfied it is impossible for one to differentiate between the actual data and the junk values. If a hacker decided to change the entire string of data being transmitted with another random string , an alert would be sent because when decrypted the actual data might very well not be an integer .Another key aspect of this encryption scheme is that the famed "brute force technique" where a hacker tries all possible combinations of the key to obtain the clear-text would not work as the data that is being transmitted along with being encrypted is also jumbled.

The encryption scheme employed in this project is not only applicable for home automation but for any IoT node to node or node to database communication thus it would help in expanding the IoT market as the major concern until now was data security.

7.2 Scope for improvement

As it can be seen in Figures 5.1, 5.2 and 5.3 the encrypted data which includes both the junk and actual data has a total length of 15 characters. This is because Firebase accepts only 15 characters under one label name. Including more characters would increase the effectiveness of the encryption as it increases ambiguity on the hacker's side.

When the house is locked the PIR sensor is activated and triggers the buzzer if it detects motion. This can lead to false alarms if there are pets in the house and so another possible improvement to the project is to include RFID sensors throughout the house and an RFID tag attached to the pet's collar so that when the pet comes within the range of the motion sensor the RFID reader would have classified the motion as that of the pet.

The other improvement that could be made to the project is to realise all the above mentioned features in a scale proportional to an actual house, an example would be to use a CFL bulb for lighting and an electromagnetic door lock instead of an LED.

References

For the Project work, Harvard referencing style must be followed:

[Referring a Journal Paper]

[1] *Dey, S., Roy, A., & Das, S. (2016). Home automation using Internet of Thing. New York: IEEE, pp.5-6.*

[2] *Kodali, Ravi & Jain, Vishal & Bose, Suvadeep & Boppana, Lakshmi. (2016). IoT based smart security and home automation system. 1286-1289. 10.1109/CCAA.2016.7813916.*

[3 (Kunal, 2018)(2019). *Smart Home Automation using IOT*. [online] Ijarcce.com. Available at: https://www.ijarcce.com/upload/2016/february-16/IJARCCE%20131.pdf

[Referring a Website]

[4] Brewster, T. (2019). *How Jeep Hackers Took Over Steering And Forced Emergency Stop At High Speed*. [online] Forbes. Available at: https://www.forbes.com/sites/thomasbrewster/2016/08/02/charlie-miller-chris-valasek-jeep-hackers-steering-brake/#4aa0fca463f4

[5] Hubijan, C. (2019). *Mirai Botnet DDoS Attack Type*. [online] corero.com. Available at: https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.html

Appendix

Appendix-A

Code for Transmitter node (Node 1)

```
symbols[] ="abcdefghijkl
void setup() {

  //Firebase
  Serial.begin(9600);
.......
}

void loop() {
  password = digitalRead(D0);
  gas = analogRead(A0);
  LDR = digitalRead(D2);
  PIR = digitalRead(D3);
  switch(m){
    case 2: v=String(password);
    Serial.println("Password :");
    break;
    case 3: v=String(gas);
    Serial.println("Gas :");
    break;
    case 4: v=String(LDR);
    Serial.println("LDR :");
    break;
    case 5: v=String(PIR);
    Serial.println("PIR :");
    break;
  }
```

```
n = v.length();
//Serial.println("Value of n is");
//Serial.println(n);

int n1 = floor((14 - (n))/2);
int n2 = ceil((14-(n))/2);
for(i = 0 ; i<n1 ; i++)
{
  before[i] = symbols[random(0,79)];
}

 for(i = 0 ;i<n2 ; i++)
{
  after[i] = symbols[random(0,78)];
}
v = before+"_" + v+"_"+ after ;
int j;
Serial.println(v);
for (i=0;i<16;i++)//For loop for the assignment of 40+ or - values
{
   for(j=0;j<79;j++)
   {

     if (v[i]==symbols[j])
     {
       if (j<39){
        encrypt[i]=symbols[j+39];
       }
       else if(j>=39){
        encrypt[i]=symbols[j-39];
       }

   }

     else if(v[i]==symbols[79])
     {
     encrypt[i]=symbols[80];
     }
}
}
```

```
 String ENCRYPT = String(encrypt);
 Serial.println(ENCRYPT);
 n=ENCRYPT.length();
 for (i=0;i<(n-13);i++) {
   Encrypt[i]=encrypt[i];
 }
 Encrypt[0] = encrypt[4];
 Encrypt[1] = encrypt[13];
 Encrypt[2] = encrypt[10];
 Encrypt[3] = encrypt[7];
 Encrypt[4] = encrypt[8];
 Encrypt[5] = encrypt[12];
 Encrypt[6] = encrypt[11];
 Encrypt[7] = encrypt[1];
 Encrypt[8] = encrypt[3];
 Encrypt[9] = encrypt[14];
 Encrypt[10] = encrypt[6];
 Encrypt[11] = encrypt[2];
 Encrypt[12] = encrypt[5];
 Encrypt[13] = encrypt[0];
 Encrypt[14] = encrypt[9];


 String ENcrypt = String(Encrypt);
 Serial.println(ENcrypt);
 //Firebase
 if(Firebase.failed()){
   Serial.println("Not connecting");
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
 }
....//Sending the sensor read values by encrypting them through firebase
 }
 m=m+1;
 if (m>5){
   m=2;
 }

 }
```

```
    digitalWrite(D6,LOW);
  }
```

Arduino Uno code(an extension of node 1)

```
#include <Keypad.h>
int i =0,j=0,Lockdown,door=0,LDR,PIR;
char password[4];
const byte ROWS = 4; //four rows
const byte COLS = 4; //three columns
char keys[ROWS][COLS] = {
  {'D','#','0','*'},
  {'C','9','8','7'},
  {'B','6','5','4'},
  {'A','3','2','1'}
};
byte rowPins[ROWS] = {5, 4, 3, 2}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {9, 8, 7, 6}; //connect to the column pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  //Serial.begin(9600);
  pinMode(A0,INPUT);
  pinMode(12,OUTPUT);
  pinMode(13,OUTPUT);
  pinMode(10,INPUT);
  pinMode(11,OUTPUT);
  pinMode(0,INPUT);
  pinMode(1,OUTPUT);
}

void loop(){
  char key = keypad.getKey();
  if (key != NO_KEY){
    //Serial.println(key);
    password[i]=key;
    i=i+1;
  }
  String Key = String(key);

  String Password = String(password);
   if (Password == "123A")
  {
    door = 0;//open
    j=0;
    //Serial.println(door);
  }
     if (Password == "A321")
  {
    door = 1;//close
    //Serial.println(door);
    j=0;
  }

  if ((i==4)&&(Password != "123A")) {
    j=j+1;
    if (j>4)
    {
      //Send an alert , do this at the end
      Lockdown = 1;//0 pin of arduino
      //Serial.println("Danger alert");
      delay(1000);
    }
}
if ((i==4)||Key == "#"){
    //Serial.println("reset");
    i=0;
  }
    LDR = analogRead(A0);
    if(LDR>500){
    digitalWrite(12,HIGH);
  }
   else if(LDR<=500){
    digitalWrite(12,LOW);
  }
  //Serial.print("LDR value is :");
  //Serial.println(LDR);
  digitalWrite(13,door);
```

```
 int Pumpin = digitalRead(10);
 digitalWrite(11,Pumpin);
 int motorin = digitalRead(0);
 digitalWrite(1,motorin);
 delay(200);
}
```

Code for receiver node (Node 2)

```
//Decryption part
char symbols[] ="abcdefghijkl!@#$%^&*-
=+;:<>?/0123456789mnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_.";


void setup() {
 Serial.begin(9600);
......

}

void loop() {

if(Firebase.failed()){
 Serial.println("Not connecting");
 Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
 switch(m){
   case 0: Firebase.getString("Humidity").toCharArray(received,sizeof(received));
   break;
   case 1: Firebase.getString("Temperature").toCharArray(received,sizeof(received));
   break;
   case 2: Firebase.getString("House_lock").toCharArray(received,sizeof(received));
   break;
   case 3: Firebase.getString("Gas").toCharArray(received,sizeof(received));
   break;
   case 4: Firebase.getString("LDR").toCharArray(received,sizeof(received));
   break;
   case 5: Firebase.getString("PIR").toCharArray(received,sizeof(received));
   break;
 }
 String Received = String(received);
```

```
n=Received.length();
Serial.println(Received);
for (i=0;i<(n-13);i++) {
 dejumble[i]=received[i];
}
dejumble[4]=received[0];
dejumble[13]=received[1];
dejumble[10]=received[2];
dejumble[7]=received[3];
dejumble[8]=received[4];
dejumble[12]=received[5];
dejumble[11]=received[6];
dejumble[1]=received[7];
dejumble[3]=received[8];
dejumble[14]=received[9];
dejumble[6]=received[10];
dejumble[2]=received[11];
dejumble[5]=received[12];
dejumble[0]=received[13];
dejumble[9]=received[14];
String Dejumble = String(dejumble);
Serial.println(Dejumble);
for (i=0;i<16;i++)//For loop for the assignment of 40+ or - values
{
  for(j=0;j<79;j++)
  {

   if (dejumble[i]==symbols[j])
   {
    if (j<39){
     decrypt[i]=symbols[j+39];
    }
    else if(j>=39){
     decrypt[i]=symbols[j-39];
    }

  }

   else if(dejumble[i]==symbols[80])
   {
      decrypt[i]=symbols[79];
      }
}
}
String Decrypt = String(decrypt);
Serial.println(Decrypt);

n= Decrypt.length();
j=0;
for (i=0;i<n;i++)
{
  if (Decrypt[i]==symbols[79])
  {
    while (decrypt[i+1]){
      answer[j]=decrypt[i+1];
      i=i+1;
      j=j+1;
    }
  }
  i=i+1;
}
int ans = atoi(answer);

}
// Processing the data after reception from firebase and decryption
Firebase.getString("LockUnlock_app").toCharArray(Pass,sizeof(Pass));
pass=atoi(Pass);

if (opass!=pass){
   if (pass ==1){
    digitalWrite(D8,HIGH);//Locked
  }
   else if(pass==0){
    digitalWrite(D8,LOW);//Unlocked
  }
  opass=pass;
}


  if (oPassword!=Password){
   if (Password ==1){
    digitalWrite(D8,HIGH);//Locked
  }
   else if (Password==0){
    digitalWrite(D8,LOW);//Unlocked
  }
  oPassword=Password;
 }

 Firebase.getString("Gas_app").toCharArray(appGas,sizeof(appGas));
 Gas_app = atoi(appGas);
 Firebase.getString("Temp_app").toCharArray(appTemp,sizeof(appTemp));
 Temp_app =atoi(appTemp);
 Firebase.getString("LDR_app").toCharArray(appLDR,sizeof(appLDR));
 LDR_app =atoi(appLDR);
 Firebase.getString("PIR_app").toCharArray(appPIR,sizeof(appPIR));
 PIR_app =atoi(appPIR);
 if(Gas_app==1){
 if (Gas>=200){
   digitalWrite(D1,HIGH);
   Gas_ubi=1;
 }
 else if(Gas<200){
   digitalWrite(D1,LOW);
   Gas_ubi = 0;
 }
 }
 if(Gas_app==0){
   digitalWrite(D1,LOW);
 Gas_ubi==0;
 }

if(Temp_app == 1){
 if(Temperature>=30){
   digitalWrite(D2,HIGH);

 }
 else if(Temperature<30){
   digitalWrite(D2,LOW);
 }
 }
 if(Temp_app ==0 ){
   digitalWrite(D2,LOW);

 }


lock_state = digitalRead(D0);

 if ((PIR_app==1)&&(lock_state==1)){
   if(PIR==1){
   digitalWrite(D7,HIGH);//Burglar
   Burglar_ubi = 1;
   }
   if(PIR==0){
    digitalWrite(D7,LOW);
   }
 }
 if((PIR_app==0)||lock_state==0){
   digitalWrite(D7,LOW);
 }

   m=m+1;
if(m>5){
   m=0;
 }
}
```

Water management node (Node 3)

```
//firebase part
  #include <ESP8266WiFi.h>
  #include <FirebaseArduino.h>
  #define FIREBASE_HOST "mjam-8d493.firebaseio.com"
  #define FIREBASE_AUTH "5t6jVnNcbwN7D0g6mT7ldraNIlBd2MLwU8DRf4nR"
  #define WIFI_SSID "Ragha"
  #define WIFI_PASSWORD "lightningblade"
char Enable[5];
const int pingPin = D1; // Trigger Pin of Ultrasonic Sensor
const int echoPin = D2; // Echo Pin of Ultrasonic Sensor
```

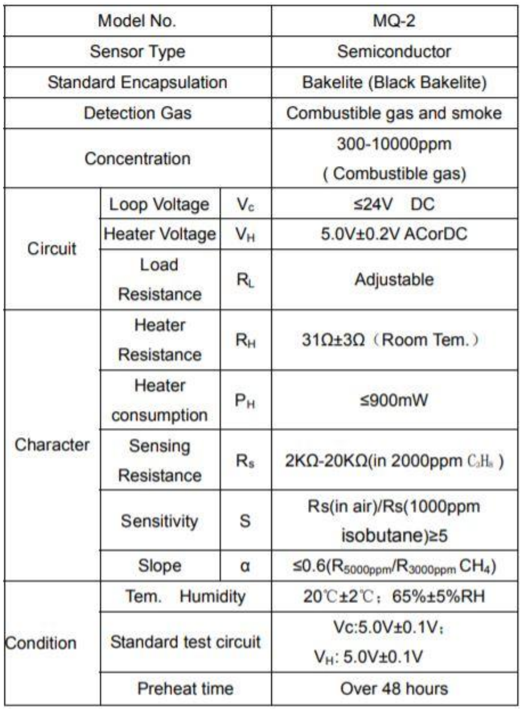
```
long percent;
int motor = 0,water_sensor,Leak,shortage;
void setup() {
   //Firebase
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);// Starting Serial Terminal
   digitalWrite(D3,INPUT);
}

void loop() {
  if(Firebase.failed()){
    Serial.println("Not connecting");
   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  }
  Firebase.getString("Automatic_WaterPump").toCharArray(Enable,sizeof(Enable));
   int enable = atoi(Enable);
   long duration, cm;
   percent = ((13-cm)*100)/12;
   Serial.print(percent);
   Serial.println("%");
   delay(100);
   if(enable ==1){
   if(percent<=40){
    motor = 1;
   }
   if(percent>=80){
    motor=0;
   }
   }
   if (enable==0){
    motor==0;
   }
   if(percent<=20){
    shortage = 1;
   }
   if(percent>20){
    shortage = 0;
   }
   water_sensor = digitalRead(D3);
   if (water_sensor==LOW){
    Leak = 0;
   }
   if (water_sensor == HIGH){
    Leak = 1;
   }
   String Percent = String(percent);
   Serial.println(Leak);
}
long microsecondsToCentimeters(long microseconds) {
   return microseconds / 29 / 2;   3
}
```

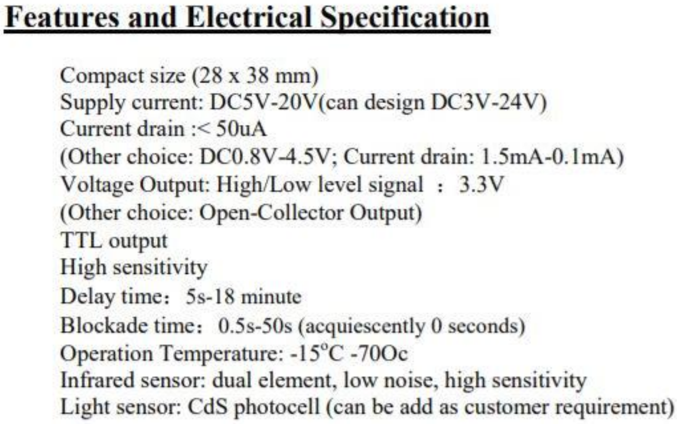
Appendix –B

Specification of the sensors used



Model No.			MQ-2
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite (Black Bakelite)
Detection Gas			Combustible gas and smoke
Concentration			300-10000ppm (Combustible gas)
Circuit	Loop Voltage	V_c	≤24V DC
	Heater Voltage	V_H	5.0V±0.2V ACorDC
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	31Ω±3Ω（Room Tem.）
	Heater consumption	P_H	≤900mW
	Sensing Resistance	R_s	2KΩ-20KΩ(in 2000ppm C_3H_8)
	Sensitivity	S	Rs(in air)/Rs(1000ppm isobutane)≥5
	Slope	α	≤0.6($R_{5000ppm}/R_{3000ppm}$ CH_4)
Condition	Tem. Humidity		20℃±2℃；65%±5%RH
	Standard test circuit		Vc:5.0V±0.1V； V_H: 5.0V±0.1V
	Preheat time		Over 48 hours

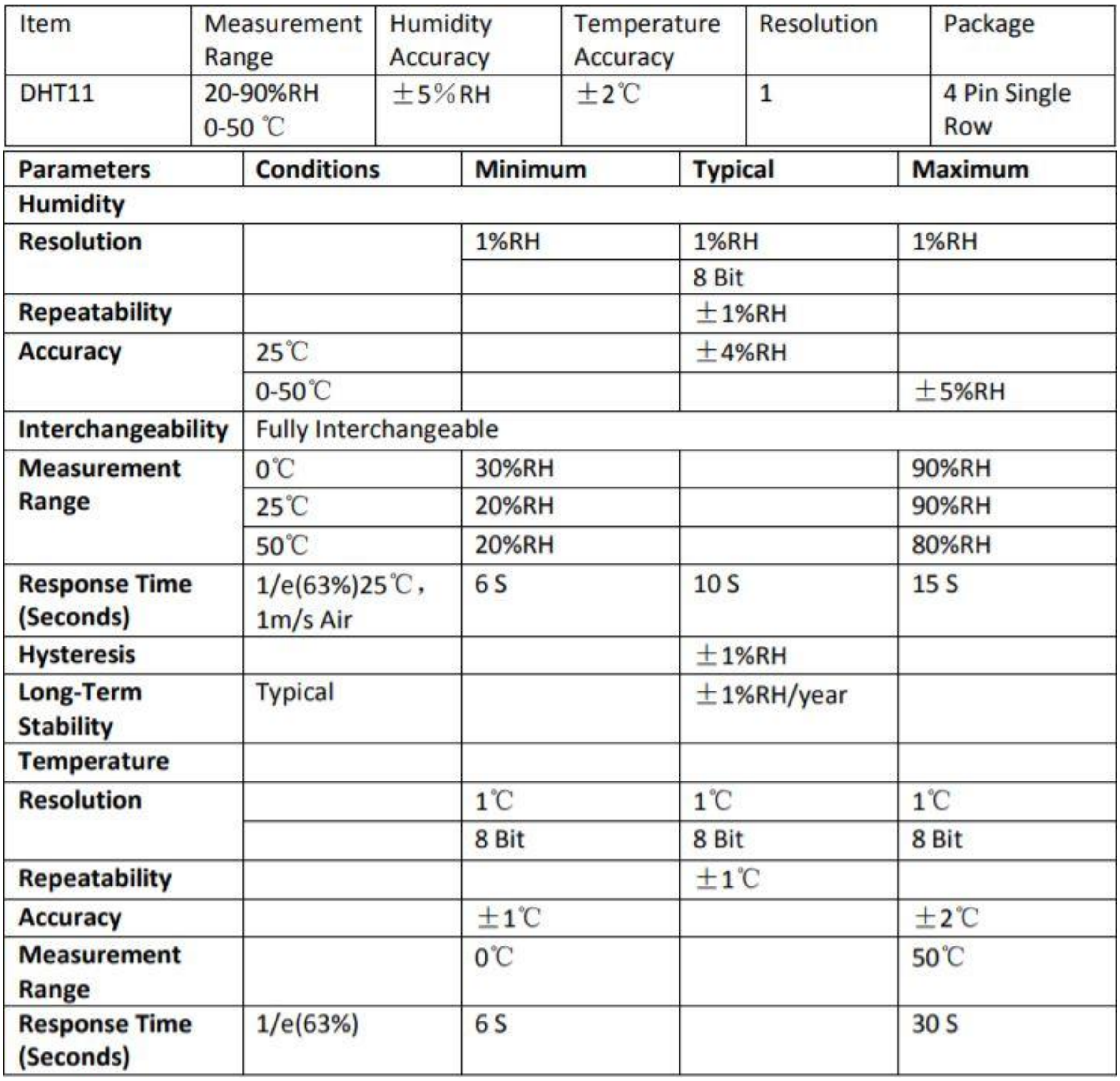
Figure 1 : MQ-2 Gas sensor Specifications



Features and Electrical Specification

Compact size (28 x 38 mm)
Supply current: DC5V-20V(can design DC3V-24V)
Current drain :< 50uA
(Other choice: DC0.8V-4.5V; Current drain: 1.5mA-0.1mA)
Voltage Output: High/Low level signal ： 3.3V
(Other choice: Open-Collector Output)
TTL output
High sensitivity
Delay time：5s-18 minute
Blockade time：0.5s-50s (acquiescently 0 seconds)
Operation Temperature: -15^{o}C -70Oc
Infrared sensor: dual element, low noise, high sensitivity
Light sensor: CdS photocell (can be add as customer requirement)

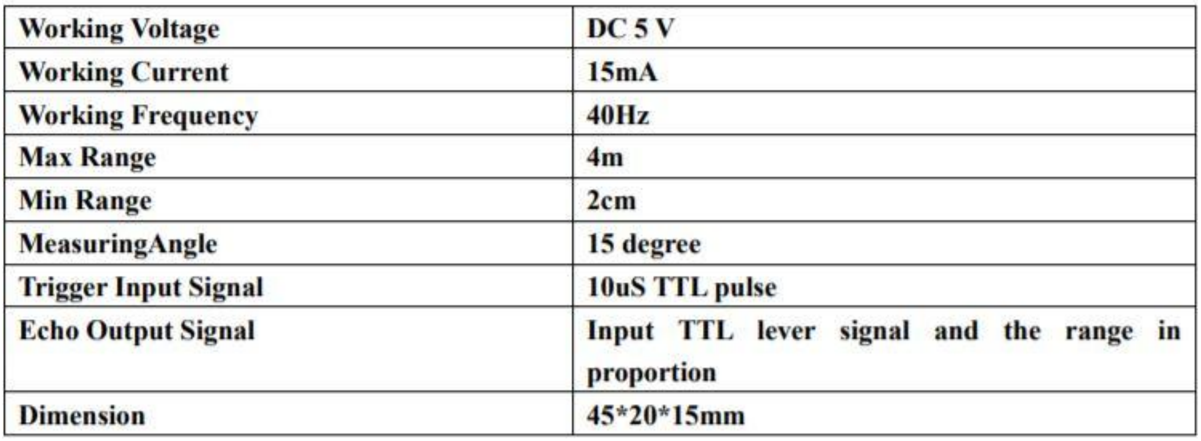
Figure 2 : PIR Sensor Specifications



Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

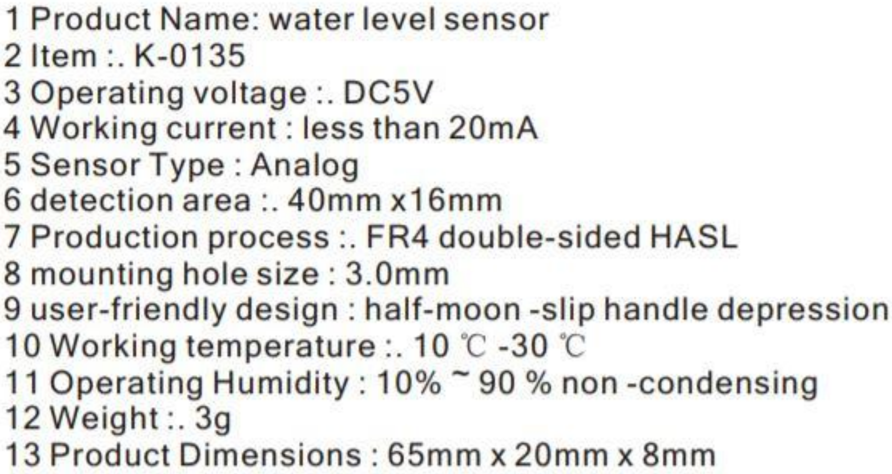
Parameters	**Conditions**	**Minimum**	**Typical**	**Maximum**
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃，1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Figure 3 : DHT11 Sensor Specifications



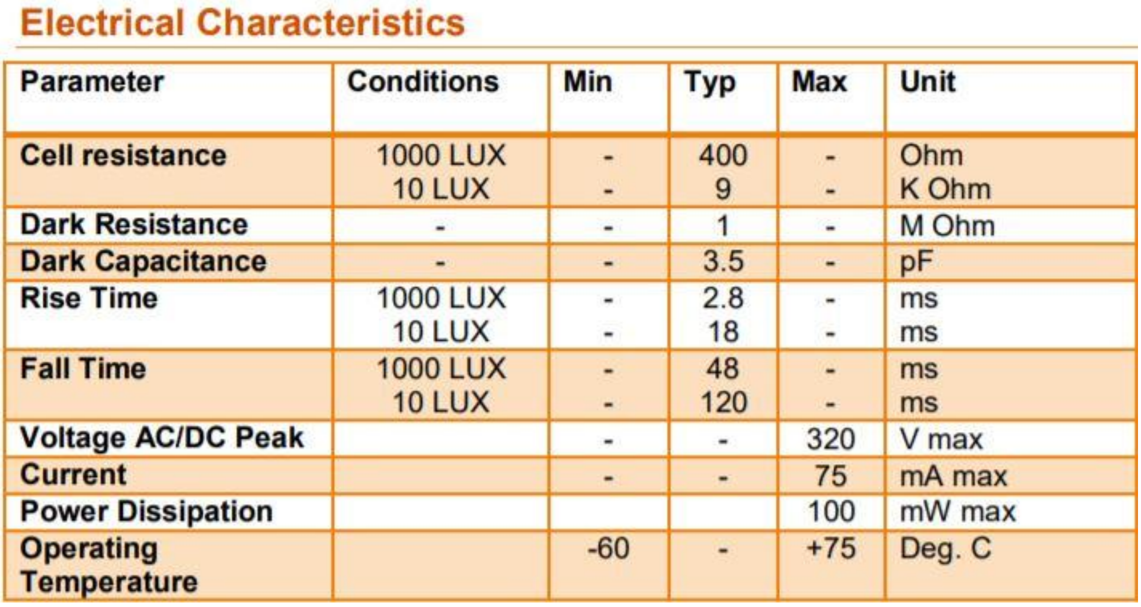
Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Figure 4 : Ultrasonic Sensor Specifications



1 Product Name: water level sensor
2 Item :. K-0135
3 Operating voltage :. DC5V
4 Working current : less than 20mA
5 Sensor Type : Analog
6 detection area :. 40mm x16mm
7 Production process :. FR4 double-sided HASL
8 mounting hole size : 3.0mm
9 user-friendly design : half-moon -slip handle depression
10 Working temperature :. 10 ℃ -30 ℃
11 Operating Humidity : 10% ~ 90 % non -condensing
12 Weight :. 3g
13 Product Dimensions : 65mm x 20mm x 8mm

Figure 5 : Water Sensor Specifications



Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX 10 LUX	- -	400 9	- -	Ohm K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX 10 LUX	- -	2.8 18	- -	ms ms
Fall Time	1000 LUX 10 LUX	- -	48 120	- -	ms ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

Figure 6 : LDR Specifications