# IOT BASED MONITORING AND AUTOMATION SYSTEM FOR A SMART HOME

By

Md. Riffat Haider

19271002

A thesis submitted to the Department of Electrical & Electronic Engineering (EEE) in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical & Electronic Engineering (EEE)

Department of Electrical & Electronic Engineering (EEE)

Brac University

Summer 2019

© 2021. Md. Riffat Haider

All rights reserved.

## Declaration

It is hereby declared that

1. The project submitted is my own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. I have acknowledged all main sources of help.

Student's Full Name & Signature:

**Md. Riffat Haider**
19271002

# Approval

The project titled "IoT based monitoring and automation system for a smart home" submitted by

1. Md. Riffat Haider (19271002)

of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Engineering (M. Engg) in EEE on 29 April,2021.

**Examining Committee:**

Supervisor:
(Member)

Dr. A.S. Nazmul Huda
Assistant Professor
Department of Electrical and Electronic Engineering
Brac University, Bangladesh.

Internal Expert Examiner:
(Member)

Dr. Abu S.M. Mohsin
Assistant Professor
Department of Electrical and Electronic Engineering
Brac University, Bangladesh.

Chairperson:
(Chair)

Dr. Md. Mosaddequr Rahman
Professor and Chairperson
Department of Electrical and Electronic Engineering
Brac University, Bangladesh.

## Ethics Statement

I hereby declare that this project on “IoT based monitoring and automation system for a smart home” has met the research criteria for completing the degree which has been written and completed without any copy. All the information and data are the reflection of my work. Also, the systems and software coding used here are being done by us. I sometimes have collected some information from other papers where it is properly cited. There is nothing in this paper that can be related to any project or paper. I have completed total work with my own effort and this unique work is done by taking some assistance from supervisor and university.

## Abstract

The world is now observing the 4th industrial revolution where IoT plays a very important role in automation. The study focuses on the concept of designing and implementing different IoT devices in a home automation system and how we can monitor and optimize electrical equipment. The basic concept is to collect data from different sensors and based on that data we will perform certain tasks. The data will be sent remotely to a server. Real time notifications will be sent to users to notify certain events. Users can remotely turn off or on certain devices.



**Keywords:** IoT, wireless communication, cloud computing, home automation, sensor monitoring

## Dedication

This project is dedicated to my honorable parents and teachers.

## Acknowledgement

I would like to express sincere thanks to my thesis supervisor, Dr. A.S. Nazmul Huda, Assistant Professor, Dept. of Electrical & Electronic Engineering (EEE), Brac University, for his supervision to make a successful completion of the thesis. I am also grateful to Brac University for providing me the necessary help for the successful completion of this thesis.

# Table of Contents

# List of Tables

# List of Figures

## List of Acronyms

| | |
|---|---|
| IoT | Internet Of Things |
| HTTP | Hypertext Transfer Protocol |
| WAN | Wireless Area Network |
| PAN | Personal Area Network |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| LAN | Local Area Network |
| MQTT | Message Queuing Telemetry Transport |
| BAN | Body Area Network |
| SSL | Secure Sockets Layer |
| CURL | Client Uniform Resource Locator |
| SDN | Software Defined Network |
| LDR | Light Depending Resistor |

# Chapter 1

# Introduction

## 1.1 Introduction

A smart home is one that uses Internet-connected devices to track and control common household appliances such as lighting, water monitoring, and electrical equipment optimization. Smart homes allow not only for the control of everyday devices such as smart doors and lights, but also for the monitoring of home security. More home protection is provided by IP-based cameras, alarms, motion sensors, firefighting equipment, and connected door locks. IoT-connected wireless sensors are bringing the revolution to have such automations in the home.

The Internet of Things, in combination with sensors, enables a wide range of utilities and applications, such as smart metering to manage energy consumption, which can help communities and cities address the issue of depleting energy supplies. In the same way, all home tools collaborate by exchanging information, optimizing activities, and making decisions based on the information in the smart world. Sensors optimize home utilities based on human behavior, such as temperature and humidity sensors for automatic controlling.

Smart homes operate by automating the home and its appliances, reducing the amount of user input required to manage them. Arduino, which uses sensors and actuators to build smart home applications, is one of the most popular hardware platforms, and NodeMCU is commonly used for networking. For big data analytics and data prediction, the cloud structure is an important part.

The Internet of Things (IoT) is the network of physical devices and objects that are used in everyday life that are linked to the Internet. It is interconnected with a variety of objects that communicate with one another through sensors, actuators, and processors. The aim of the Internet

of Things is to achieve a high level of intelligence with minimal human interference [1]. It also encompasses all of the various applications, making various services and monitoring more knowledgeable, interactive, and effective [2]. Incorporating wireless sensor network is difficult due to the wide range of applications and technological differences among the devices [3]. Home automation [4–6] system having various security challenges [7] has different advantages like energy [8-13] management, less human activity, efficiency improvement and so on. Smart home makes a unique interconnected system [14] so that in future activity can be predicted [15]. A multi sensor approach [16] can help to achieve the goal. But due to different sensors, hardware and protocols, challenges [17] may come. Devices should be capable of self-monitoring in IoT smart homes to achieve self-maintenance and management, which is a major concern to improve their wellbeing and send a notification to the user [18].

## 1.2 Internet of Things (IoT) Basic Concept

The Internet plays a critical role in bringing people's lives together. Let's take a look at the past of the Internet of Things. The term "Internet of Things" was coined by Kevin Ashton. “Then I thought of the 'Internet of Things,' and I thought, 'that'll do – or maybe even better,'” he says. He thought about how smart objects (things) communicate [19]. IoT may be used in a variety of utilities and applications. IoT is made up of several components. There are a lot of them. Sensors, motorized devices, NFC, magnetic frequency devices, and other devices fall into this group. Smart objects can make life much simpler for ordinary people. “It allows people and objects to be linked anywhere, anywhere, anyplace, everything, and anywhere by using any route or any service,” according to the Internet of Things [20]. The Internet of Things (IoT) will assist us in understanding the data that surrounds us. Objects are linked together. As a result, a system change

to one object will lead to a system change to another object. The Internet of Things is divided into three layers:

- **Perception Layer:** The main layer of the Internet of Things [21]. The layer is used to house smart objects.
- **Network Layer:** This layer is in charge of transferring data from the physical layer to the cloud and vice versa.
- **Application Layer:** Analytics or reporting to the end user.

The machine-to-machine (M2M) interface is now widely used and plays an important role in the Internet of Things [22].

Figure 1.1: Internet of Things basic concept

To implement IoT successfully and gather value from devices, we have to make sure that the five necessary phases are adapted properly:

**1. Developing Phase:** Threats and emerging possibilities arising in the market as a result of IoT must be considered. This is the stage where information collection occurs. It is possible to make

use of the data provided by smart devices. The data can be used by large industries, businesses, and educational institutions.

**2. Networking Phase:** Smart devices send data to the cloud during this process.

**3. Aggregate Phase:** During this method, the devices themselves aggregate data.

**4. Examine Phase:** The data gathered from the analytics is needed to create the basic types, so the process must be calibrated.

**5. Operation Phase:** A specific procedure is carried out or automated based on certain values obtained during the analysis process. [23]

## 1.3 IoT Characteristics

Architecture of the network is changing rapidly and getting complex day by day. Firewalls, Encryption systems are used in today's networking system. It is very hard to maintain all the complex networks and servers. So, in the upcoming years, SDN would be perfect solution. Routers and switches are very common for the application of the current traditional network. The control plane used in the network is for packet forwarding. Data plane handles rules of the control plane. They are decoupled from each other. By decoupling, programming, controlling, automation become scalable and flexible enough.

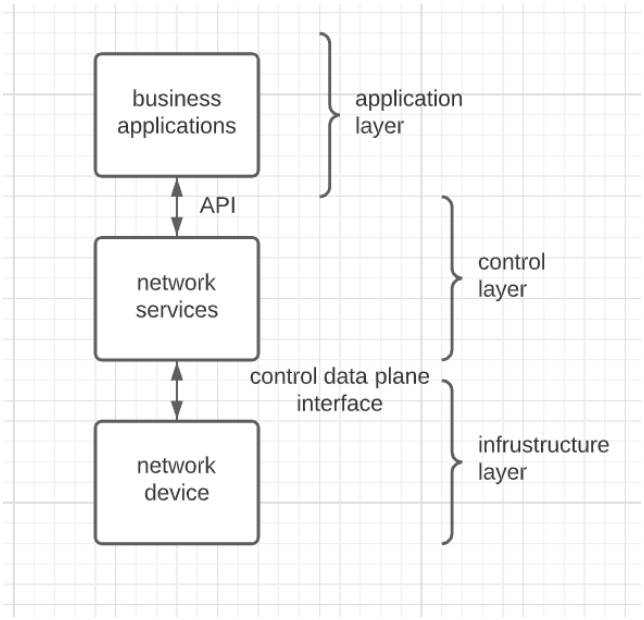


Figure 1.2: Software Defined Networking

SDN (Software-Defined Networking) is one of the most innovative networking concepts. It aids in the enhancement of a useful and efficient network control flow that simplifies asset costs while also benefiting a large number of operators. SDN has revolutionized how complex networks are managed, and it is divided into two sections: (1) The controllers are separated from the forwarding plane by SDN. (2) Decoupling allows end-users to reach the control level. It's difficult to tell the difference between the control and data levels in traditional networks. As a result, end-users are unaware of device network configurations.

In the traditional network, the data plane notifies the user about the data flow, and the control plane gives information access to user; in this, one device could take the decision of progressing the packet, and it also is in charge of maintaining the routing table information and required to reserve the data, and SDN was refined to design, build, and manage the networks that decouple the network layer and forwarding layer becoming directly programmable by the source automation tools.

## 1.4 IoT Architecture

SDN architecture concept is simple but powerful. SDN, a networking system, is a system with interconnected devices. Figure 1.3 demonstrates the communication between IoT controller and SDN controller.

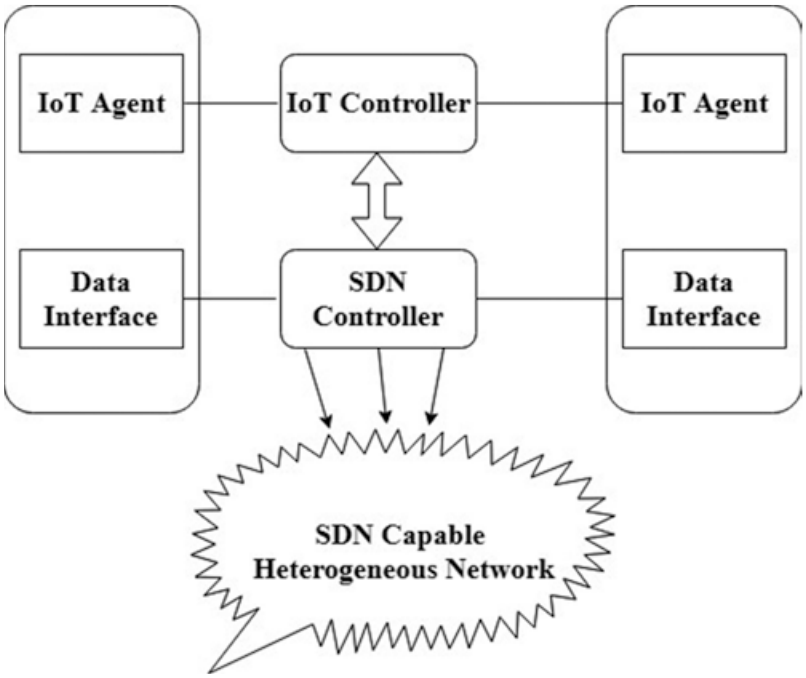


Figure 1.3: SDN Based IoT Architecture

**IoT Agent:** An IoT agent is a key component in the networking system. It lets a group of devices send data managed by their own protocol. Every IoT agent is supported by the controller of IoT. It gives information like mac address, physical location, etc. in order to send correct information. Every item has its unique identification number which helps controllers to detect the specific device.

**IoT Controller:** An effective IoT architecture consists of IoT controllers. Key decisions and information of IoT agents are handled by the controller. Controllers and gateways are middle-tier components that primarily collect information from front-end devices. However, the role of

controllers is overriding the information. The location of controllers will also vary based on the intelligence and constraints of the network connection to the front-end devices. IoT agents facilitate the registry address and the data that is accumulated is controlled by the controller.

**SDN Controller:** This controller manages the control flow. It enables centralized management of data. It creates a bridge between virtual and physical layers of communication.

## 1.5 Project Overview

With immense possibilities in automation, IoT can play a vital role by supporting a combined system. The main focus of the project is to implement different IoT devices in a smart home so that data can be collected through these devices and store them in the cloud to take various decisions and actions based on the data. We have collected readings from temperature, gas, LDR, fingerprint sensor, ultrasonic sensor and based on a certain condition, an email is sent. Based on the temperature and gas sensor reading, a user can be notified. Also, we are using force sensor for tracking our availability on the bed. We are using finger print sensor for activity logging, ultrasonic sensor for smart bin.

## 1.6 Summary

In the second chapter of this paper, a discussion is made about the ins and outs of an IoT architecture and how it works. Following that third chapter describes the automation process and its characteristics. Then chapter four is about the hardware and software implementation process and technique. Later, chapter five is the development of control algorithms. Suddenly, the uniqueness, substantiate goal and how an user will get benefited from this project sift canvas in chapter six. Finally, in chapter seven the paper was compacted enlightening the system customization with IoT devices and the forthcoming work on this engineering.

# Chapter 2

# Literature Review

## 2.1 Introduction

The current situation in which we find ourselves is undeniably urbanized. However, it does not have a very consistent and methodical way of life. The word "Internet of Things" refers to devices like electrical appliances, actuators, and sensors that communicate with one another over the Internet.

Because of the combination of IoT and internetworking, the concept of a smart city is now commercially viable. Different technology, such as information and networking, are used in smart cities to provide public services that are much more interactive and feasible. According to recent estimates, more than 6 billion people will be living in cities by 2050. To develop smart city architecture, it is essential to instill a smart IoT vision. Smart waste management, smart hospitals, smart air control, smart parking, smart buildings, and an air monitoring system are all part of this vision. Many plans have been made to transform a city into a smart city by integrating different criteria. We looked at some of the technologies that can be used to put these brilliant ideas into action in this paper.

Remote sensor systems have only recently been linked in a variety of applications associated with smart city-like conditions for monitoring the urban environment and instilling smart living. The authors assume that the network of things (IoT) is a complex component by interfacing and empowering gadgets to the web in this one-of-a-kind situation. In addition, tech-savvy accessories can provide and receive data for a better understanding of their surroundings. A notable test of relevant city services is gathering data from various provenances with the aim of learning

extraction. As a sobering reminder, the day is rapidly approaching when knowledge would be more costly than gadgets. IoT, WSN, mobile phones, and a general community of people are the primary sources of knowledge. Since all of the gadgets generate different types of data, their different configurations result in different rates of data flow, making it heterogeneous. The data's heterogeneity necessitates a framework that creates a virtual space in which this knowledge source may exist. A complex relationship exists between the various knowledge collections created by various applications associated with a savvy city hence paves the way of smart homes and automation.

## 2.2 IoT in Home Automation

The Internet of Things is all about contact between various devices in different locations in some way [24]. The Internet of Things (IoT) will play a significant role in every business and day-to-day activities. To illustrate the importance of the Internet of Things in the creation of Smart Cities, consider Smart Housing and Smart Vehicles. The Internet of Things can be used in an infinite number of ways in a Smart House. Since the key is linked to the internet, we can use it on doors so that if a friend or family member is waiting for anyone outside the building, anyone can quickly open the door. Also, one can integrate the IoT in the bedrooms, vehicles and so on [25, 26].

The Internet of Things (IoT) is described as the massive interconnection of smart devices. From small sensors to large vehicles, this term is used. Because of the importance of IoT networks, some people consider the Internet of Things to be the fourth generation of computers. Many people use them on a daily basis without even realizing it. Let's break down the IoT's name and meaning, "the evolving interconnection of smart devices," to see whether we need it or not. As a result, it refers to the link or contact between smart devices. Or, to put it another way, the Internet of Things (IoT) is the internet—communication or interconnection—of things like smart devices.

## 2.3 IoT System Working Principle

In the sections below, we'll explain what each one means and how they work together to form a full IoT system. We'll go through each of these components for greater understanding.

**1. Devices/Sensors:** To begin, sensors or devices gather data from their surroundings. This information may range from a simple temperature reading to a complete video stream.

Since multiple sensors can be bundled together or sensors can be part of a system that does more than just sense things, we use the term "sensors/devices." One's phone, for example, is a computer with several sensors (camera, accelerometer, GPS, and so on), but it is more than just a sensor since it can also perform a variety of actions.

Regardless of whether it's a standalone sensor or a complete system, data is collected from the environment in the first stage.

**2. Interconnectedness:** The data is then sent to the cloud, but it must first find a way to get there! Cellular, radio, WiFi, Bluetooth, low-power wide-area networks (LPWAN), connecting via a gateway/router, or connecting directly to the internet via ethernet are all options for connecting sensors/devices to the cloud.

Power consumption, range, and bandwidth are all tradeoffs in each alternative. The best communication solution depends on the IoT program, but they all accomplish the same goal: transferring data to the cloud.

**3. Information Processing:** After the data is transferred to the cloud, it is processed by machines. This may be anything as easy as double-checking that the temperature reading is within reasonable

limits. It may also be extremely complicated, such as identifying objects using computer vision on film (such as intruders on a property).

**4. User Interface:** The data is then made useful to the end user in some way. This could be done by sending a message to the customer (email, text, notification, etc). For example, if the temperature in the company's cold storage is too high, a text message will be sent.

A consumer might have access to an app that allows them to check in on the device in advance. A consumer can, for example, use a phone app or a web browser to check the video feeds on various assets. It isn't always a one-way lane, though. The user will be able to perform an action and influence the device depending on the IoT program.

Since a lot of hardware is involved in the technology or platform, there is a void in technical standardization. Companies that render IoT-based goods use random architectures that they are comfortable with and simple to implement due to a lack of standards [27].

Interoperability becomes a problem when various hardware and platforms are involved. This hardware makes use of various software to transfer and utilize data, and a larger software infrastructure on the network and on background servers would be required to deal with smart objects and provide support services. Since IoT devices are dynamic and mobile, fault tolerance is a major concern. They are constantly changing their state and behavior. It is necessary to structure the Internet of Things and have the ability to automatically adjust to changing circumstances [27].

## 2.4 IoT Communication Technology

The internet of things employs a variety of communication protocols and technologies. Bluetooth, Wifi, Radio Protocols, LTE-A, and WiFi-Direct are some of the main IoT technologies and

protocols (IoT Communication Protocols). These Internet of Things communication protocols are tailored to and fulfill the functional requirements of an IoT framework. Let's take a look at each of the seven IoT Communication Protocols/Techniques.

**1. Bluetooth Technology:** Protocols / Technology for short-range IoT communications. Bluetooth is a technology that has grown in importance in the computing and consumer product industries. It is expected to be critical for wearable devices in particular, which will bind to the IoT once again, but in many cases via a smartphone.

The new Bluetooth Low-Energy (BLE) – or Bluetooth Smart, as it is now known – protocol is an important IoT protocol. Importantly, although it has a range comparable to Bluetooth, it has been designed to use substantially less power.

**2. Zigbee Wireless Technology:** ZigBee is a wireless technology that is similar to Bluetooth and is mostly used in industrial settings. It offers low-power operation, high security, robustness, and high performance in complex systems, and it is well equipped to take advantage of wireless control and sensor networks in IoT applications.

The most recent version of ZigBee is version 3.0, which effectively unifies the different ZigBee wireless protocols into a single standard.

**3. The Z-Wave:** Z-Wave is a low-power RF communications IoT technology that is mainly used in home automation for products such as lamp controllers and sensors.

Although Z-Wave uses a simpler protocol than some others, allowing for quicker and easier production, Sigma Designs is the only manufacturer of chips, as opposed to multiple sources for other wireless technologies like ZigBee and others.

**4. Wi-Fi:** WiFi connectivity is one of the most common IoT communication protocols, and for many developers, it's an obvious option, particularly given the widespread availability of WiFi in the home and on LANs. There is a large current infrastructure that allows for quick data transfer and the handling of large amounts of data.

The most popular WiFi standard in homes and many companies right now is 802.11n, which has a frequency of hundreds of megabits per second and is good for file transfers but might be too power-hungry for many IoT applications.

**5. Cellular:** GSM/3G/4G cellular networking features can be used by any IoT program that needs service over longer distances. Although cellular can clearly send large amounts of data, especially in 4G, the cost and power consumption will be prohibitive for many applications.

However, sensor-based low-bandwidth-data projects that send very small amounts of data over the Internet can find it perfect.

**6. NFC (Near Field Communication):** Near Field Communication (NFC) is an Internet of Things (IoT) technology. It allows for easy and secure communication between electronic devices, especially smartphones, enabling customers to conduct transactions without having to be physically present.

It allows the user to attach electronic devices and access digital content. It basically expands the capabilities of contactless card technology by allowing devices to exchange data over a distance of less than 4cm.

**7. LoRaWAN:** LoRaWAN is a widely used Internet of Things (IoT) technology that targets wide-area network (WAN) applications. The LoRaWAN standard was created to give low-power WANs

the features they need to support low-cost mobile safe communication in IoT, smart city, and industrial applications.

Data rates range from 0.3 kbps to 50 kbps, and it is designed to meet low-power requirements and serve vast networks of millions of devices.

## 2.5 IoT Protocols

There are 4 key IoT protocols. They are discussed briefly in this section.

**1. Constrained Application Protocol (CoAP):** CoAP is an internet utility protocol for devices with limited resources. It was created to allow easy, constrained devices to connect to the Internet of Things through constrained networks with low bandwidth availability.

This protocol is designed for IoT systems that use HTTP protocols and is mainly used for machine-to-machine (M2M) communication. The UDP protocol is used by CoAP for its lightweight implementation. Restful architecture is also used, which is similar to the HTTP protocol. It employs dtls for the secure transfer of data within the slipping layer.

**2. Message Queue Telemetry Transport (MQTT):** MQTT (Message Queue Telemetry Transport) is a messaging protocol designed for M2M communication that was created in 1999 with the help of IBM's Andy Stanford-Clark and Arcom's Arlen Nipper. It's usually used in IoT for long-range tracking. The main task is to collect data from a large number of devices and to deliver its infrastructure. MQTT is a protocol that links devices and networks using packages and middleware.

All of the computers, including IBM's latest message sight appliance, link to facts concentrator servers. MQTT protocols work on top of TCP to provide simple and dependable data streams.

**3. Advanced Message Queuing Protocol (AMQP):** John O'Hara of JP Morgan Chase in London came up with this idea. AMQP is a message-oriented middleware environment software layer protocol. It facilitates secure verbal communication by using primitives like at-most-once, at-least-once, and exactly as soon as shipping.

AMQP – IoT protocols are made up of hard and fast components that route and save messages within a broker carrier, as well as a collection of policies for connecting them. Patron programs can communicate with the dealer and interact with the AMQP model using the AMQP protocol.

The following three additives are included in this version, and they may be linked into processing chains on the server to produce the desired capabilities.

**Exchange:** Receives messages from publisher-centric systems and sends them to message queues.

**Message Queue:** Holds messages until the eating client program can process them completely.

**Binding:** Describes how the message queue and the change are connected.

**4. Data Distribution Service (DDS):** The submit-subscribe method enables scalable, real-time, accurate, high-overall efficiency, and interoperable statistics shift. Multicasting is used by DDS to provide high-quality QoS to applications. DDS is used on a variety of platforms, from small devices to the cloud, and it facilitates green bandwidth use as well as the agile orchestration of system additives. The fundamental layers of the DDS – IoT protocols are the facts centric submit-subscribe (dcps) and statistics-local reconstruction layer (dlrl).

## 2.6 Summary

The concept of IoT in home automation is right now slowly but gradually becoming more appealing to all the people. But, there are a number of challenges that need to be overcome. On smart devices and houses, various types of attacks may be carried out, such as impersonation/identity spoofing, which seeks to absorb someone's energy on their behalf.

Eavesdropping is another assault on IoT-based systems, since it takes advantage of public communication networks to obtain information about a user's and household's energy usage. Data tampering allows attackers to alter the rate pricing of energy by manipulating the exchanged data. By modifying the readings of smart meters and sensors, attackers can obtain authorization and control access, as well as remotely track and customize energy usage information. Through analyzing usage data, it is possible to keep track of the users' personal details.

# Chapter 3

# Model Architecture

## 3.1 Communication Architecture

The communications among the different subsystems may be performed either wirelessly or by using conductive devices. The former is often more expensive in terms of hardware and requires more power. However, the advent of new conductive materials and soft/printed electronics will enable seamless and massive integration of sensors. In any case, note that such an interconnection subsystem is essential for the reliability, so its selection is the key.

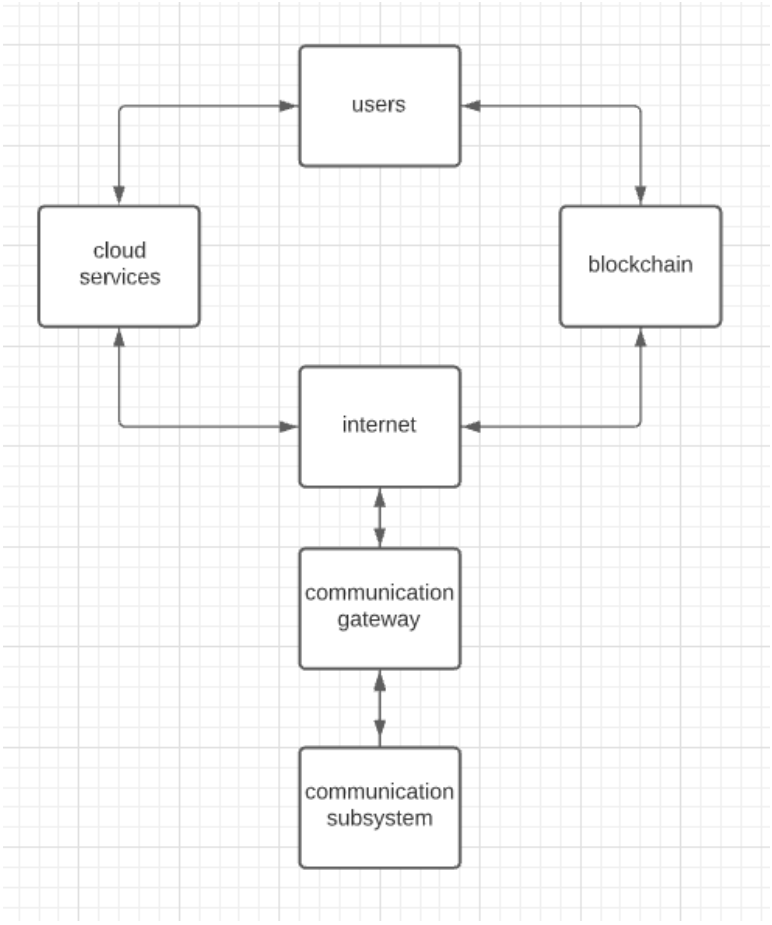


Figure 3.1: Generic architecture of IoT communication

Figure 3.1 also illustrates a typical IoT architecture that supports collecting data and that is composed by the following main components:

- A communications gateway that exchanges information with smart devices in order to send it through the Internet or an internal LAN to remote services provided, for instance, by cloud servers or a blockchain. The communications gateway can also process the received data and provide fast responses to the smart devices, thus acting as an edge or fog computing gateway [28].
- A cloud server that collects and stores data, and provides certain remote services to the smart devices and to remote users.
- A blockchain. Although it is not essential for the basic functioning of a smart home automation system, it enables different useful features like redundancy, data security and trustworthiness [29]. Moreover, a blockchain can run smart contracts (pieces of software that translate legal terms into code that can be run autonomously on a blockchain) [30], which allow for automating certain tasks according to the detected events.

  It is also important to note that most architectures divide the previously mentioned components into three layers [31]:

- Body Area Network (BAN). The components of each smart device are connected through a common network topology characterized by providing a really short range. There are various types of BANs such as Wearable BAN (WBAN), Implantable BAN (IBAN). WBANs need to be energy-efficient, since they mostly rely on batteries [32].

- Personal Area Network (PAN) or Local Area Network (LAN). This network collects data from smart devices and sends them to a cloud or remote server. PANs usually provide shorter ranges than LANs (usually up to 10 m). In the case of smart devices, communications are performed wirelessly, so the terms Wireless PANs (WPANs) and Wireless LANs (WLANs) are often used. An example of WPAN is Bluetooth, while WiFi is a type of WLAN. It is also worth pointing out that at this communications layer it is possible to provide mesh network communications so that smart devices can communicate with each other and with the objects and machines that surround them.
- Wide Area Network (WAN). This is a type of network like the Internet, which covers a really wide area thanks to the support of a distributed infrastructure. It is essential for many IoT applications, but in some cases (e.g., critical infrastructures [33, 34] or industrial environments [35]) their services may be provided through an internal LAN.

## 3.2 System Features

IoT paves the way to integrate monitoring devices into machines. Easily controlled remotely via server makes us more mobile and easy to access data. The focus of the project is to make the home appliances much more controllable and detectable via remote server. Main features would be:

1. Automatic device data sending process via email
2. Trigger special events based on specified data or a range of data or a data set
3. Control electrical parts with suitable device enabling or disabling
4. Trigger alarm to indicate the operators to stop or start certain devices at certain period of time
5. User based administrative privileges to change parameters of different important factors

6. Based on the device data, predict future occurrences of events specified by different devices
7. Auto connection reestablishment in case of broken network connection or switching between Wi-Fi or broadband network for reliable connection

## 3.3 Security

IoT gadgets are wireless and located near public areas. The IoT challenges are divided into two categories: technical and defense. The creative difficulties arise as a result of the heterogeneous and ubiquitous existence of IoT devices, while the security issues are linked to the norms and functionalities that must be adhered to in order to achieve a secure framework. There are many components that go into ensuring security, including:

• The item that runs on all IoT devices should be approved.

• When an IoT object is switched on, it should first verify that it is part of the system before transmitting data.

• Firewalling is essential in the IoT system to channel bundles facilitated to the devices since IoT devices are required to estimate memory capacities.

• Modifications and fixes on the computer should be presented in such a way that the device's data transfer cap is not exceeded.

## 3.4 Privacy

The importance of security in the Internet of Things has already been addressed. The difficulties of IoT arrangement on safeguarding security will be addressed in this section. The challenges can be divided into two categories: information gathering strategy and information anonymization.

• **Information Gathering Strategy:** It establishes the policy in the middle of the data collection process, dictating the type of data to be collected and the access control of a thing to the data. In the information accumulation process, the amount of data collected is limited by the information gathering policy. We can ensure privacy preservation because we know that collecting and storing personal information is safer.

• **Information Anonymization:** The most suitable strategies for ensuring information anonymization are cryptographic security and data connection concealment. Multiple specific cryptographic schemes can be considered due to the diversity of the things.

## 3.5 Authentication and Authorization

The process that decides whether an individual or user has access to resources is known as authorization. Reading and writing data, executing programs, and controlling actuators are just a few examples. It may involve denying or revoking access, particularly if the person or item is malicious.

The process that recognizes the entity and is needed for authorization is known as authentication. Authentication is required for authorization in most cases. Every IoT object must be able to identify and authenticate these objects without a doubt. Because of the existence of IoT and the large number of entities involved, the process can be difficult; additionally, artifacts will need to communicate with one another for the first time, and because of both of these factors, a system was created to mutually authenticate the entities in each and every interaction in the IoT.

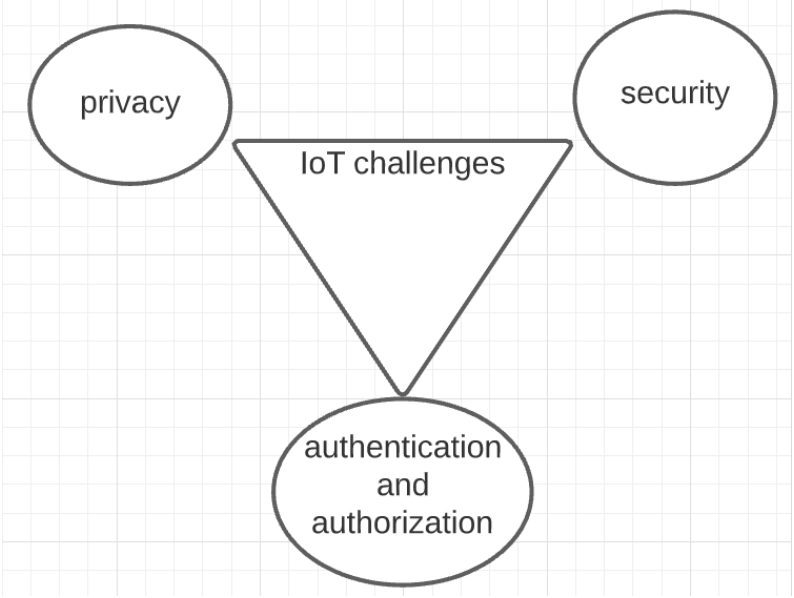


Figure 3.2: Challenges of IoT

# Chapter 4

# Hardware and Software

## 4.1 Hardware Implementation

### 4.1.1 ESP 8266 NodeMCU

NodeMCU is a low-cost open source IoT platform. It came with firmware that ran on Espressif Systems' ESP8266 Wi-Fi SoC and hardware that was built on the ESP-12 module at first. Later, the ESP32 32-bit MCU was introduced to the mix.

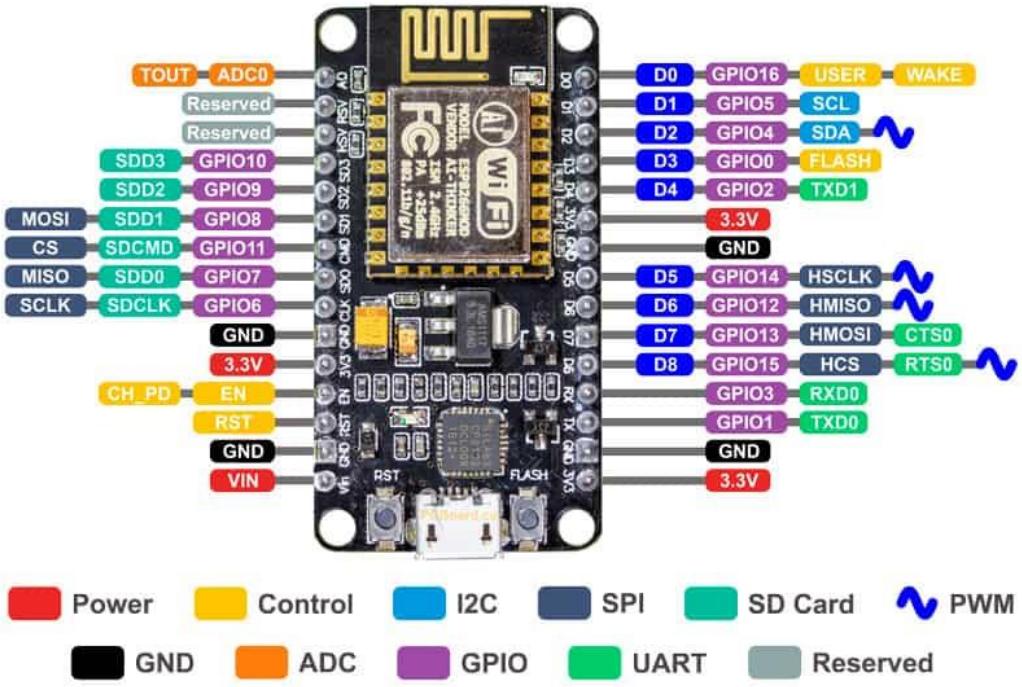


Figure 4.1: NodeMCU pinouts and functions

The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

- ADC channel – A 10-bit ADC channel.
- UART interface – UART interface is used to load code serially.

- PWM outputs – PWM pins for dimming LEDs or controlling motors.
- SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- I2S interface – I2S interface if anyone wants to add sound to the project.

Tremendous feature of ESP8266 is pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.

**NodeMCU ESP8266 Specifications & Features**

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside one's IoT projects

### 4.1.2 Arduino Mega

The Arduino Mega 2560 is an ATmega2560-based microcontroller module. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connecting it to a computer with a USB cable or powering it with a AC-to-DC adapter or battery is needed to get started.

**Features and Specifications**

- Microcontroller: ATmega2560
- Operating voltage: 5 V
- Input voltage (recommended): 7-12 V
- Digital I/O pins: 70 (of which 14 provide PWM output)
- Analog input pins: 16*
- DC current per I/O pin: 40 mA
- DC current for 3.3V pin: 50 mA
- Flash memory: 256 KB of which 8 KB used by bootloader
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock speed: 16 MHz
- Size: 4" x 2.1"
- Weight: 36 g
- Processor: ATmega2560 @ 16 MHz

- RAM size: 8 Kbytes
- Program memory size: 248 Kbytes
- Motor channels: 0
- User I/O lines: 70 (See Note 1)
- Max current on a single I/O: 40 mA
- Minimum operating voltage: 7 V
- Maximum output voltage: 12 V

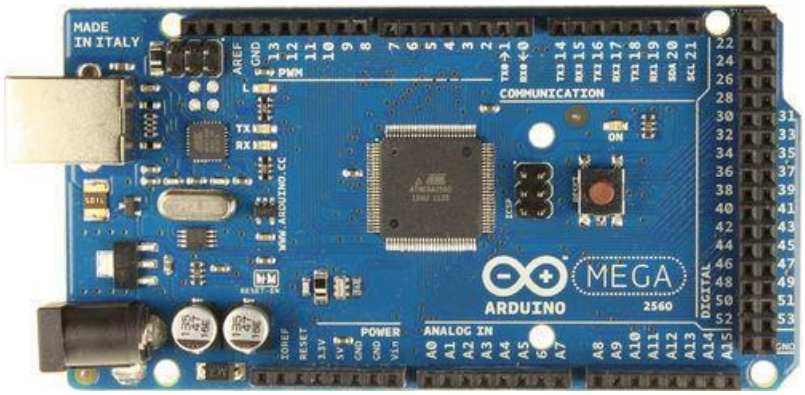


Figure 4.2: Arduino Mega

### 4.1.3 LM35 Temperature sensor

The LM35 series are precision integrated-circuit temperature devices with a linearly proportional output voltage to the temperature in degrees Celsius. In comparison to linear temperature sensors tuned in Kelvin, the LM35 unit has the advantage of not requiring the consumer to subtract a significant constant voltage from the output to obtain convenient Centigrade scaling. The LM35 system needs no external calibration or trimming to achieve standard accuracies of 14°C at room temperature and 34°C over a temperature range of 55°C to 150°C. Trimming and calibration at the wafer level ensure lower costs.

**Technical specifications**

- Local sensor accuracy (Max): (+/- C)0.5
- Operating temperature range (C): -40 to 110, -55 to 150, 0 to 100, 0 to 70
- Supply voltage (Min) (V): 4
- Supply voltage (Max) (V): 30
- Supply current (Max) (uA): 114
- Interface type: Analog output
- Sensor gain (mV/Deg C): 10

**Features**

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load


Figure 4.3: LM35 sensor

### 4.1.4 MQ2 Gas Sensor

The MQ2 gas sensor is one of the most widely used in the MQ sensor series. It's a Metal Oxide Semiconductor (MOS) style Gas Sensor, also known as Chemiresistors, since the detection is based on a change in the sensing material's resistance when the gas comes into contact with it. Gas concentrations can be detected using a simple voltage divider network. The MQ2 Gas Sensor runs on 5V DC and consumes about 800mW. It has a detection range of 200 to 10000ppm for LPG, Smoke, Alcohol, Propane, Hydrogen, Methane, and Carbon Monoxide.


Figure 4.4: MQ2 gas sensor

**Technical specifications**

- Operating voltage: 5V
- Load resistance: 20 KΩ
- Heater resistance: 33Ω ± 5%
- Heating consumption: <800mw
- Sensing Resistance: 10 KΩ – 60 KΩ
- Concentration Scope: 200 – 10000ppm
- Preheat Time: Over 24 hour

### 4.1.5 Fingerprint Sensor

To use the optical fingerprint sensor, we must meet two conditions. The first step is to enroll fingerprints, which entails assigning ID numbers to each print so that one can query them later. Once all of the prints have been enrolled, one can quickly scan the sensor by asking it to determine which ID (if any) is currently being photographed.

**Technical specifications**

- Supply voltage: 3.6 - 6.0VDC
- Operating current: 120mA max
- Peak current: 150mA max
- Fingerprint imaging time: <1.0 seconds
- Window area: 14mm x 18mm
- Signature file: 256 bytes
- Template file: 512 bytes

- Storage capacity: 162 templates
- Safety ratings (1-5 low to high safety)
- False Acceptance Rate: <0.001% (Security level 3)
- False Reject Rate: <1.0% (Security level 3)
- Interface: TTL Serial
- Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)
- Working temperature rating: -20C to +50C
- Working humidity: 40%-85% RH
- Full Dimensions: 56 x 20 x 21.5mm
- Exposed Dimensions (when placed in box): 21mm x 21mm x 21mm triangular
- Weight: 20 grams

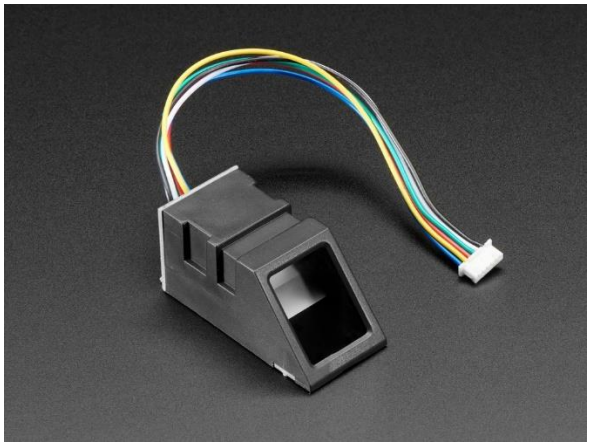

Figure 4.5: Fingerprint sensor

### 4.1.6 Ultrasonic Sensor

An ultrasonic sensor is an electronic system that uses ultrasonic sound waves to determine the distance between a target object and transforms the reflected sound into an electrical signal. Ultrasonic waves propagate faster than audible sound waves (i.e. the sound that humans can hear).

The transmitter (which emits sound using piezoelectric crystals) and the receiver are the two main components of ultrasonic sensors (which encounters the sound after it has travelled to and from the target).

**Technical specifications**

- Operating voltage: +5V
- Theoretical  Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz


Figure 4.6: Ultrasonic sensor

## 4.1.7 Light Depending Resistor

The Light Dependent Resistor (LDR) is made up of a piece of exposed semiconductor material, such as cadmium sulphide, that changes its electrical resistance from thousands of Ohms in the

dark to only a few hundred Ohms when light shines on it, causing hole-electron pairs to form in the material.

The net result is an increase in conductivity with a decrease in resistance in exchange for more illumination. Photoresistive cells often have a slow reaction time, taking several seconds to respond to a change in light intensity.

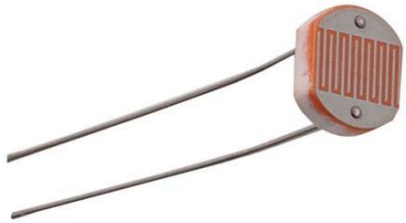

Figure 4.7: Light Depending Resistor

### 4.1.8 5-220V Relay

Relays are switches that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. As relay diagrams show, when a relay contact is normally open (NO), there is an open contact when the relay is not energized. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized. In either case, applying electrical current to the contacts will change their state.

Relays are generally used to switch smaller currents in a control circuit and do not usually control power consuming devices except for small motors and Solenoids that draw low amps. Nonetheless,

relays can “control” larger voltages and amperes by having an amplifying effect because a small voltage applied to a relays coil can result in a large voltage being switched by the contacts.

Protective relays can prevent equipment damage by detecting electrical abnormalities, including overcurrent, undercurrent, overloads and reverse currents. In addition, relays are also widely used to switch starting coils, heating elements, pilot lights and audible alarms.


Figure 4.8: Relay

## 4.2 Software

### 4.2.1 Arduino IDE

The Arduino Integrated Development Environment (IDE) is a cross-platform program written in C and C++ functions. It's used to write and upload programs to Arduino-compatible boards, as well as other vendor development boards with the support of third-party cores.

The GNU General Public License, version 2 is used to license the IDE's source code. The Arduino IDE uses special code structuring rules to support the languages C and C++. The Wiring project includes a software library that is included with the Arduino IDE and provides several standard input and output procedures. User-written code only needs two simple functions to start the sketch and the main program loop, which are compiled and connected into an executable cyclic executive program with the GNU toolchain, which is also included with the IDE distribution. The Arduino IDE uses the avrdude software to translate executable code into a text file in hexadecimal encoding, which is then loaded into the Arduino board's firmware by a loader program.


Figure 4.9: Arduino IDE

### 4.2.2 Ubidots

Ubidots is an Internet of Things (IoT) platform that enables innovators and businesses to test and scale IoT projects into production. The Ubidots platform allows users to send data to the cloud from any Internet-enabled computer, customize behavior and notifications based on real-time data, and activate the value of their data using visual tools. Ubidots provides a REST API that enables users to read and write data to the following resources: data sources, variables, values, events, and insights. The API accepts all HTTP and HTTPS requests and needs an API Key. Data will be protected with two more replication, encrypted storage and optional TLS/SSL data support. Permission groups can be customized to each module of the platform, making sure the right information is shown to the right user.

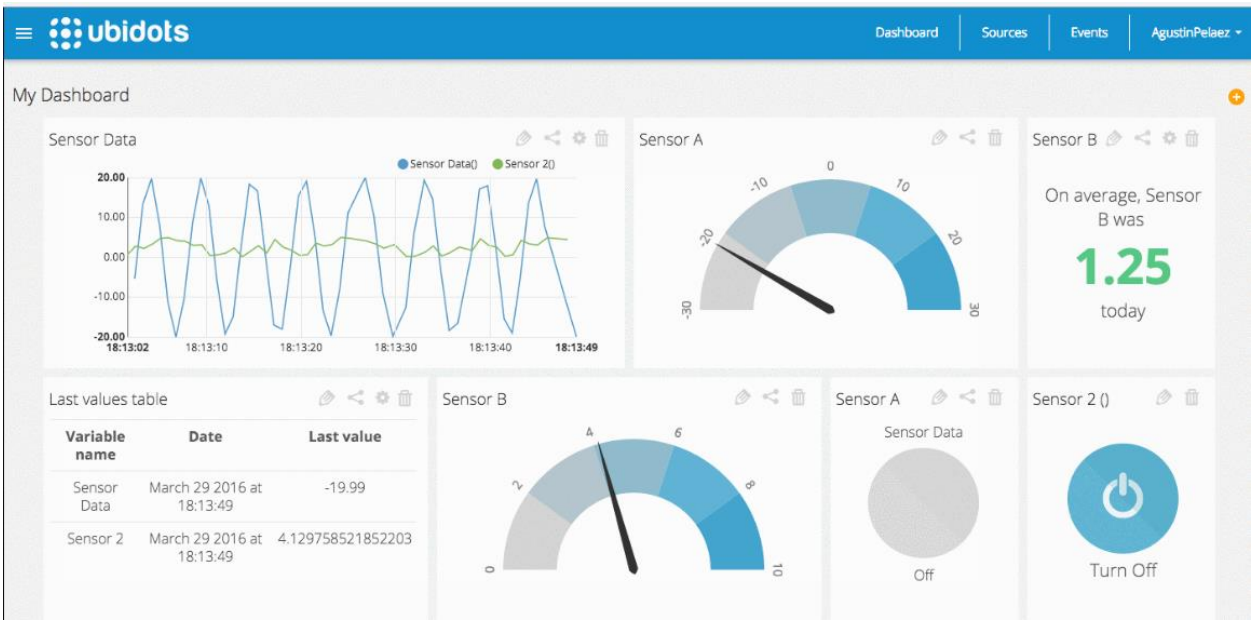


Figure 4.10: Ubidots interface

# Chapter 5

# Control Flow and Algorithms

## 5.1 Introduction

IoT devices work in a certain flow. A processor or microcontroller such as ATMEGA uses ESP8266 to send data collected through different sensors via Wireless Field or mobile data (GSM) over the internet via special API. The connection should be secured and data must be encrypted otherwise data leakage can be an issue. The Internet of Things (IoT) unites physical objects with the virtual world. Intelligent devices and machines are connected to each other and the Internet. They capture relevant information about their direct environment, then analyze and link it. The devices perform specific tasks on that basis. A sensor, for example, measures the temperature outside and the smart device it is installed in responds by turning up the heating. All of that is done automatically, without users taking any actions themselves. Users can still control the IoT devices remotely if they wish, for example, using an app on their smartphone.

## 5.2 Remote Data Processing

To implement different IoT devices, we need to first divide sections of the factory so that we could implement system one by one and get all those data together later. Let’s define different sections first.

### 5.2.1 Ubidots Basics

Every time a device updates a sensor value in a variable, a data-point or "dot" is created. Ubidots stores dots that come from our devices inside variables, and these stored dots have corresponding timestamps:

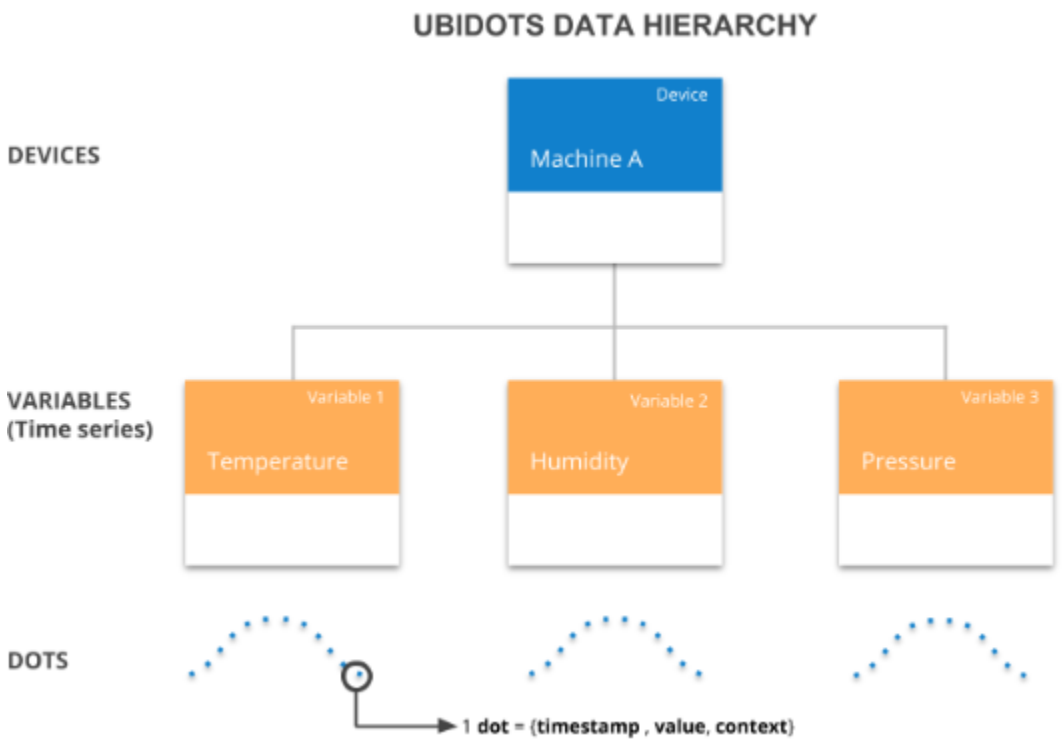


Figure 5.1: Ubidots data hierarchy

Each dot contains these items:

| **Item** | **Description** | **Mandatory** |
|---|---|---|
| value | A numerical value. Ubidots accepts up to 16 floating-point length numbers. | YES |
| timestamp | Unix Epoch time, in milliseconds. If not specified, then our servers will assign one upon reception. | NO |

| Item | Description | Mandatory |
|---|---|---|
| context | An arbitrary collection of key-value pairs. Mostly used to store the latitude and longitude coordinates of GPS devices. | NO |

Table 1: Number of items a dot contains in Ubidots

### 5.2.2 Timestamps

A timestamp, as best described here, is a way to track time as a running total of seconds. This count starts at the Unix Epoch on January 1st, 1970 at UTC. Therefore, the unix time stamp is merely the number of seconds between a particular date and the Unix Epoch. It is necessary to keep in mind that when one sends data to Ubidots, he must set the timestamp in milliseconds; also, if he retrieves a dot's timestamp, it will be in milliseconds.

"timestamp" : 1537453824000

The above timestamp corresponds to Thursday, September 20, 2018 2:30:24 PM.

### 5.2.3 Time Series

Based on the above, we can illustrate an Ubidots time series like this:

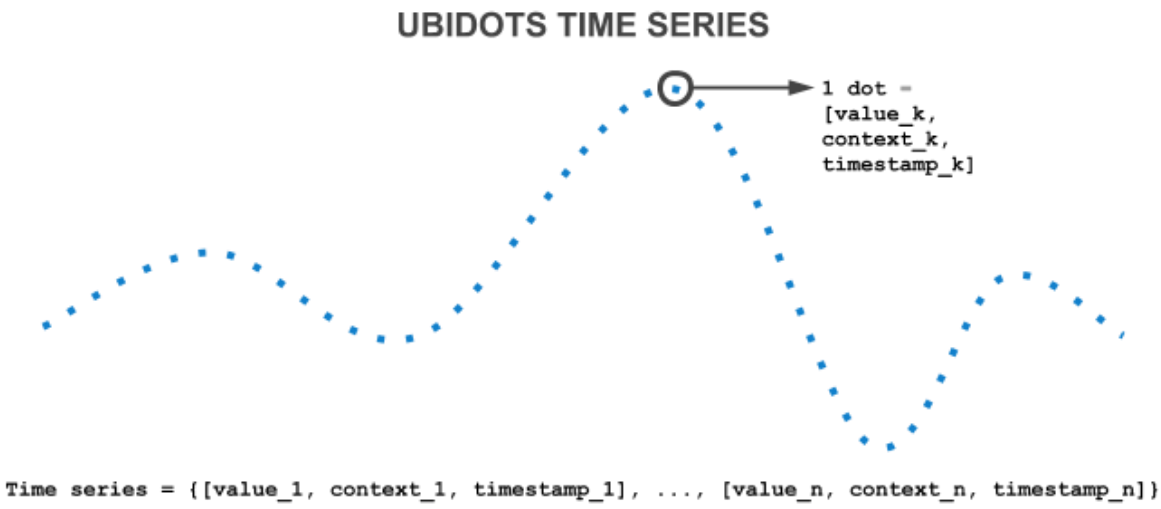


Figure 5.2: Ubidots time series

Ubidots is an agnostic platform, this means that it does not really care what hardware device one is using, as long as he is able to interact with us through at least one of these protocols:

- HTTP
- MQTT
- TCP/UDP

### 5.2.4 HTTP Requests

The following methods are specified within the HTTP standard:

| **HTTP Method** | **Description** |
|---|---|
| GET | Used to retrieve information |

| HTTP Method | Description |
|---|---|
| POST | Used to create a new element |
| PATCH | Used to update existing elements |
| DELETE | Used to delete existing elements |

Table 2: Methods specified within the HTTP standard

HTTP is a request/response protocol, this means that every request that someone makes is answered by the server. This response includes a number (response code) and a body. For example, when anyone makes a request to retrieve a file on a webpage, he/she builds a GET request. If the request is correct, the server will typically return a 200 response code, along with the file requested (body).

Under the same logic, when a device sends a GET request to the Ubidots' server (i.e. requesting the last value of a variable), then the server sends back a response with the status of the request (response code), and a body, which would be the a value with its context and timestamp.

An HTTP request also needs the parameters below:

- **Host**: Specifies the server one will be making HTTP requests to.
- **Path**: This is typically the remaining portion of the URL that specifies the resource one wants to consume, be it a variable or a device. For example, if an API endpoint is: industrial.api.Ubidots.com/api/v1.6/devices/my-device then the path would be /api/v1.6/devices/my-device

- **Headers**: Define the operating parameters of the HTTP request such as authentication, Content-Type, Content-Length, etc.
- **Body/payload**: In the case of POST and PATCH requests, this is the data sent by the device to the server. GET requests typically do not have a body because they are meant to request data, not to send data.

Ubidots accepts data as JavaScript Object Notation or JSON. JSON is a typical HTTP data type, it is a collection of name/value pairs. In various programming languages, this is treated as an object, record, struct, dictionary, hash table, keyed list, or associative array. It is also human readable and language independent. An example of a JSON data type that Ubidots accepts can be referenced below:

{"temperature": {"value":10, "timestamp": 1534881387000, "context": {"machine": "1st floor"}}}

A typical HTTP request to Ubidots should be set as below:

POST {PATH} HTTP/1.1<CR><LN>

Host: {HOST}<CR><LN>

User-Agent: {USER_AGENT}<CR><LN>

X-Auth-Token: {TOKEN}<CR><LN>

Content-Type: application/json<CR><LN>

Content-Length: {PAYLOAD_LENGTH}<CR><LN><CR><LN>

{PAYLOAD}

<CR><LN>

Where:

- {PATH}: Path to the resource to consume
  Example: /api/v1.6/variables/ to get user's variables information.
- {HOST}: Host URL.
  Example: industrial.api.Ubidots.com
- {USER_AGENT}: An optional string used to identify the type of client, be it by application type, operating system, software vendor or software version of the requesting user agent. Examples: ESP8266/1.0, Particle/1.2
- {TOKEN}: Unique key that authorizes one's device to ingest data inside the Ubidots account.
- {PAYLOAD_LENGTH}: The number of characters of one's payload. Example: The payload {"temperature": 20} will have a content-length of 19.
- {PAYLOAD}: Data to send.
  Example: {"temperature": 20}

Ubidots has wide variety dashboard. It has own rest API. We can use it to insert our data into database.

To interact with Ubidots REST API, an HTTPS client is needed. Here are a few options:

**Insomnia:** Simple yet powerful REST API Client with cookie management, environment variables, code generation, and authentication for Mac, Windows, and Linux.

**Postman:** Powerful, simple-to-use GUI that makes API development faster, easier, and better. Comes with API request history, collections, environments, tests, sharing and more.

**CURL:** Command line HTTP client with an intuitive UI, JSON support, syntax highlighting, plugins, and more.

**HTTPie:** Command line tool for transferring data using various protocols with URL syntax.

### 5.2.5 API URLs

API access can be made over HTTP or HTTPS, using the following endpoints based on the Ubidots Account type.

Using HTTPS is more secure so that we make sure the data travels encrypted, and avoid exposing API token and/or sensor data.

**HTTP**

| Ubidots Account | Endpoint | Port |
|---|---|---|
| Educational | http://things.Ubidots.com | 80 |
| Industrial | http://industrial.api.Ubidots.com | 80 |

Table 3: HTTP site for Ubidots API access

**HTTPS**

| Ubidots Account | Endpoint | Port |
|---|---|---|
| Educational | https://things.Ubidots.com | 443 |
| Industrial | https://industrial.api.Ubidots.com | 443 |

Table 4: HTTPS site for Ubidots API access

### 5.2.6 Sending Data

Ubidots REST API allows anyone to send data to the platform in two different ways:

- Send data to a device
- Send data to a variable

**Sending data to a device:**

**POST**

https://industrial.api.Ubidots.com/api/v1.6/devices/**{DEVICE_LABEL}**

**POST**

https://things.Ubidots.com/api/v1.6/devices/**{DEVICE_LABEL}**

**Request structure:**

POST /api/v1.6/devices/{DEVICE_LABEL} HTTP/1.1<CR><LN>

Host: {Host}<CR><LN>

User-Agent: {USER_AGENT}<CR><LN>

X-Auth-Token: {TOKEN}<CR><LN>

Content-Type: application/json<CR><LN>

Content-Length: {PAYLOAD_LENGTH}<CR><LN><CR><LN>

{PAYLOAD}

<CR><LN>

**Expected Response:**

HTTP/1.1 200 OK<CR><LN>

Server: nginx<CR><LN>

Date: Tue, 04 Sep 2018 22:35:06 GMT<CR><LN>

Content-Type: application/json<CR><LN>

Transfer-Encoding: chunked<CR><LN>

Vary: Cookie<CR><LN>

Allow: GET, POST, HEAD, OPTIONS<CR><LN><CR><LN>

{PAYLOAD_LENGTH_IN_HEXADECIMAL}<CR><LN>

{"{VARIABLE_LABEL}": [{"status_code": 201}]}<CR><LN>

0<CR><LN>

The easiest way to send values to Ubidots is by specifying the device label in the request path and making a POST request to it.

### 5.2.7 Ubidots Events

There are several types of Ubidots events. We will take a look at some of them. Ubidots already supports integrated events to allow anyone to send alerts and notifications to those who need to know when they need to know. Ubidots pre-built integrations include:

1. Email notifications
2. SMS notifications
3. Webhook events
4. Telegram notifications
5. Slack notifications
6. Voice Call notifications
7. Back to Normal notification
8. Geofence notifications

### 5.2.8 Timing of Ubidots Event Trigger

The figure below describes how the events engine triggers alerts inside an active event window. Note that data (blue line) passes through the threshold triggering an event; then the data must fall below the threshold against before Ubidots triggers then next event.

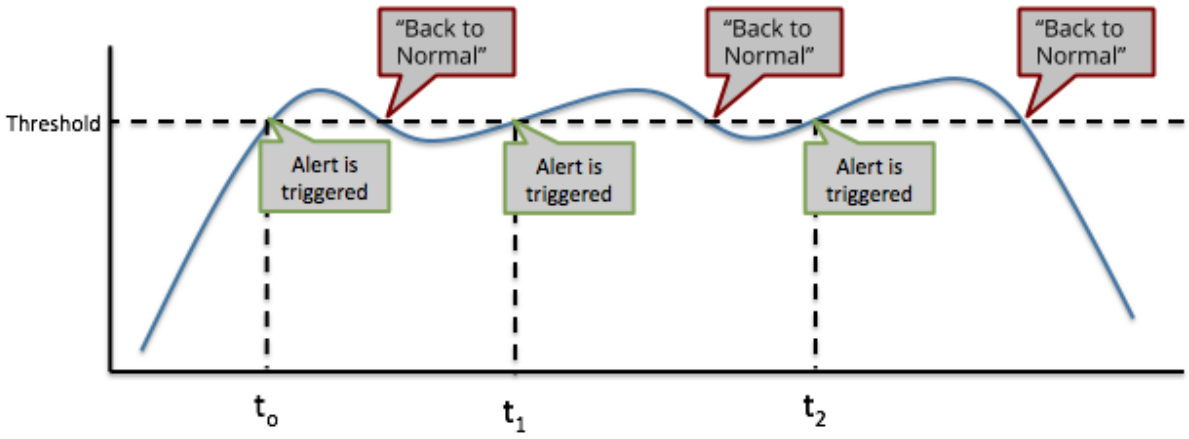


Figure 5.3: Ubidots event trigger system

After an alert is triggered, subsequent values will not be triggered again, even if they comply with the trigger conditions. A second trigger cannot take place unless the data values return below the threshold value and exceed the threshold again:

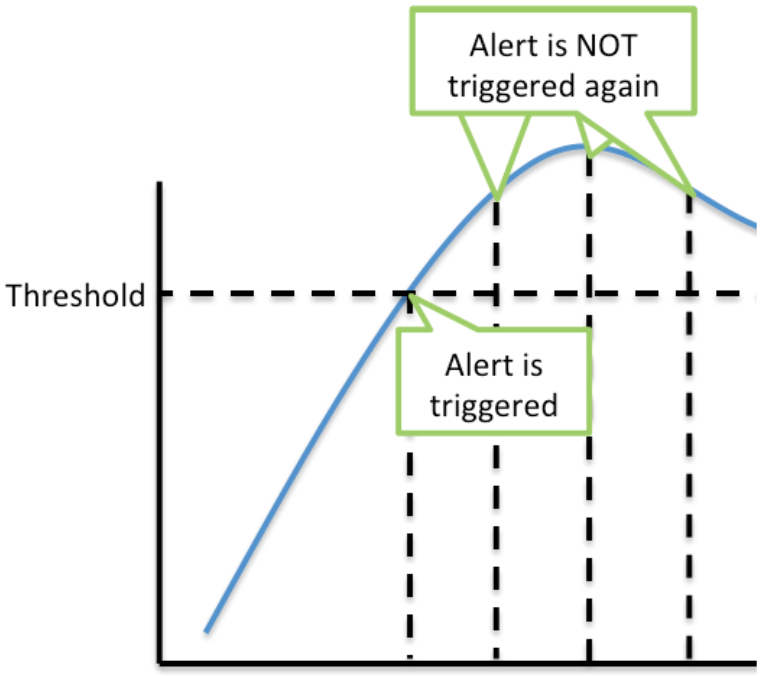


Figure 5.4: Ubidots alert trigger threshold

Events can only be triggered in an Active Event Window as depicted below.

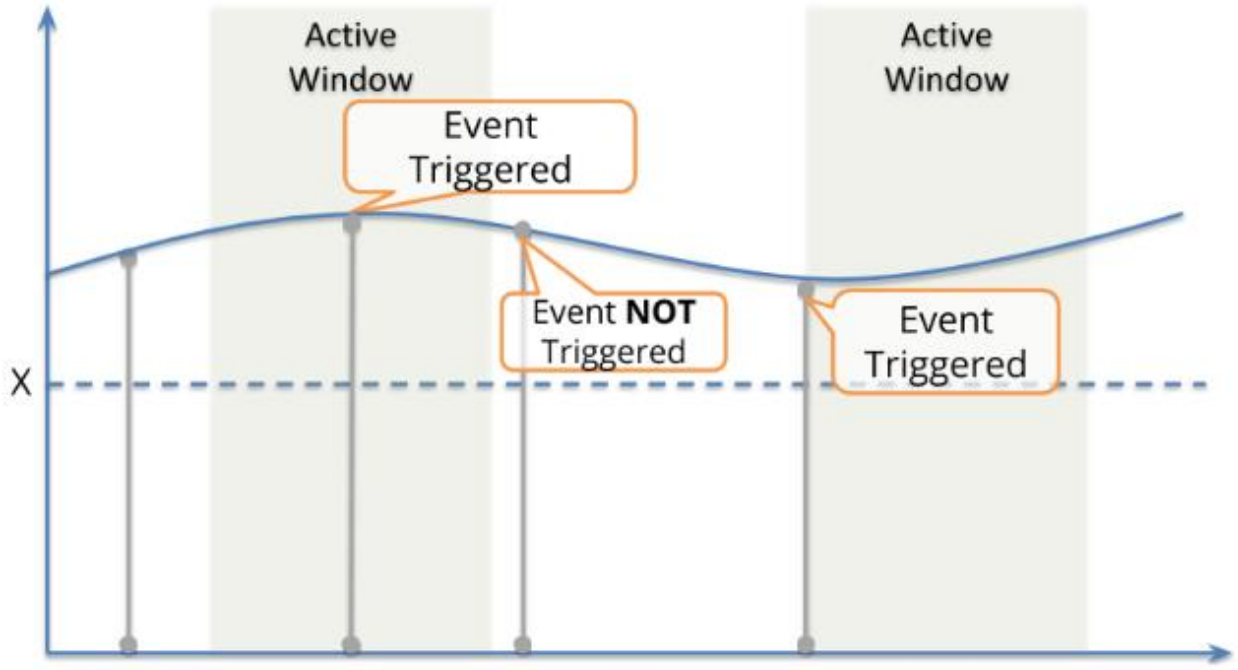


Figure 5.5: Ubidots active window for event trigger

'Has been inactive' condition events will only be triggered if the inactiveness is within the active window.

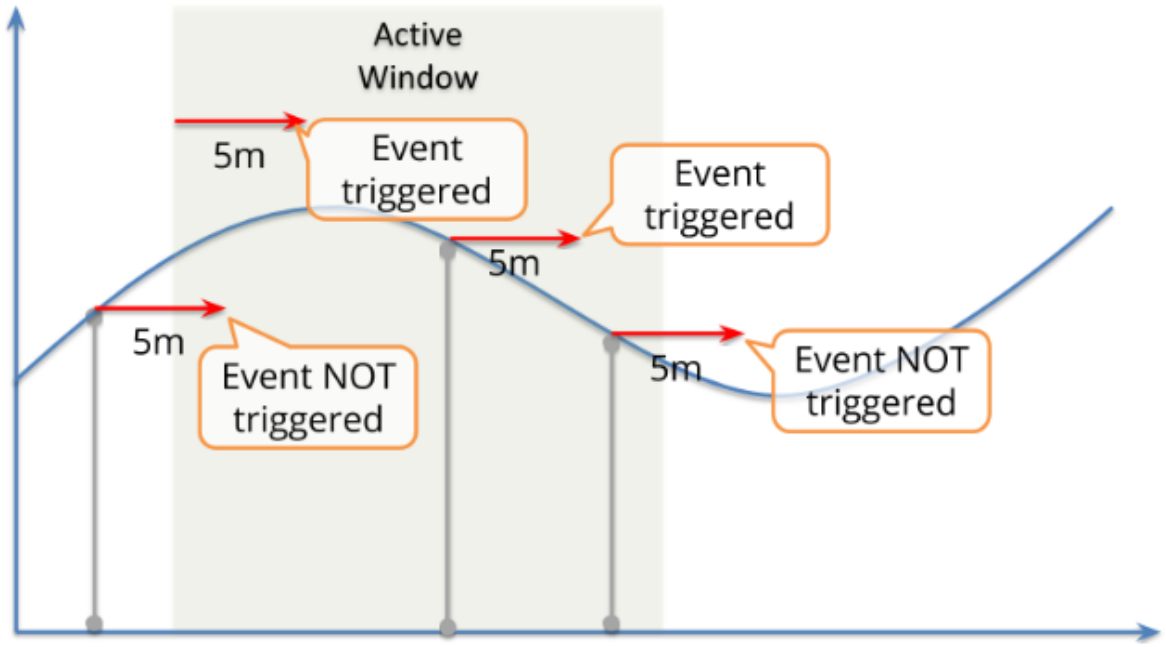


Figure 5.6: Ubidots active trigger within window

### 5.2.9 Conditional Event Creation

In Ubidots conditional events engine, multiple devices and multiple variables can be selected within a single event or built complex events with triggers exist for multi-logic events (If {dev 1, varA} or {device 2, varB} are 100, then send SMS message).

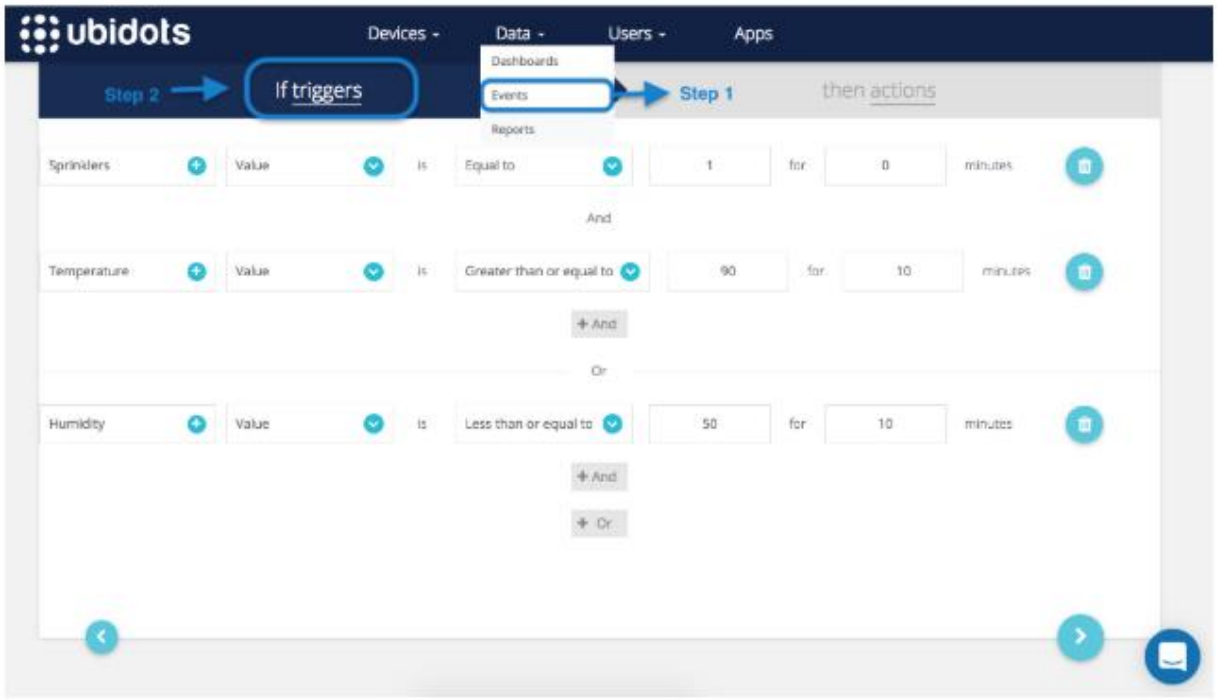


Figure 5.7: Ubidots trigger function in the web interface

# Chapter 6

# System Integration

## 6.1 Smart Fire or Heat Detection

We will use Ubidots to achieve our goal. As Ubidots has different features that allow us to easily integrate sensors. For sensors, we will use smoke detector, temperature sensor to sense fire or heat. Smoke sensor can be placed in different sections of a house such us kitchen, entrance, dining and so on.

For organizing those sensor data we will use Arduino YUN. Arduino YUN has built in wifi module. And also have mobile SIM slot which will use mobile data for connection.

### 6.1.1 Smoke Detector Working Procedure

A typical ionization chamber consists of two electrically charged plates and a radioactive source (typically Americium 241) for ionizing the air between the plates. The radioactive source emits particles that collide with the air molecules and dislodge their electrons. As the molecules lose electrons, they become positively charged ions. As other molecules gain electrons, they become negatively charged ions. Equal numbers of positive and negative ions are created. The positively charged ions are attracted to the negatively charged electrical plate, while the negatively charged ions are attracted to the positively charged plate. This creates a small ionization current that can be measured by electronic circuitry connected to the plates ("normal" condition in the detector). Particles of combustion are much larger than the ionized air molecules. As particles of combustion enter an ionization chamber, ionized air molecules collide and combine with them. Some particles become positively charged and some become negatively charged. As these relatively large

particles continue to combine with many other ions, they become recombination centers, and the total number of ionized particles in the chamber is reduced.

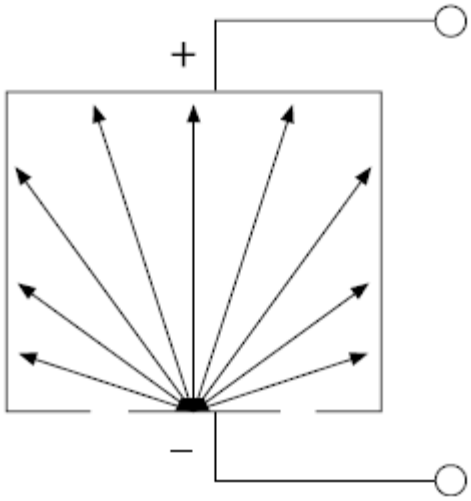


Figure 6.1: Particle radiation pattern

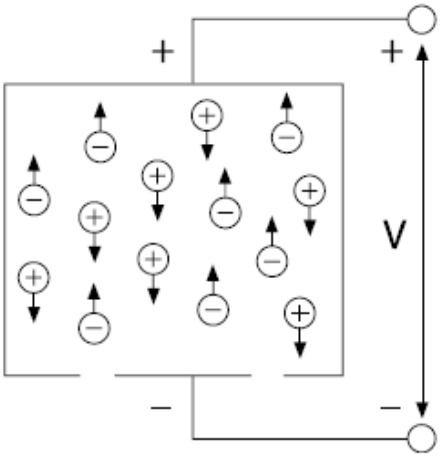


Figure 6.2: Ion distribution

This reduction in the ionized particles results in a decrease in the chamber current that is sensed by electronic circuitry monitoring the chamber. When the current is reduced by a predetermined

amount, a threshold is crossed and an “alarm” condition is established. Changes in humidity and atmospheric pressure affect the chamber current and create an effect similar to the effect of particles of combustion entering the sensing chamber. To compensate for the possible effects of humidity and pressure changes, the dual ionization chamber was developed and has become commonplace in the smoke detector market. A dual-chamber detector utilizes two ionization chambers; one is a sensing chamber, which is open to the outside air. The sensing chamber is affected by particulate matter, humidity, and atmospheric pressure. The other is a reference chamber, which is partially closed to outside air and is affected only by humidity and atmospheric pressure, because its tiny openings block the entry of larger particulate matter including particles of combustion. Electronic circuitry monitors both chambers and compares their outputs. If the humidity or the atmospheric pressure changes, the outputs of both chambers are affected equally and cancel each other. When combustion particles enter the sensing chamber, its current decreases while the current of the reference chamber remains unchanged. The resulting current imbalance is detected by the electronic circuitry. There are a number of conditions that can affect dual-chamber ionization sensors such as dust, excessive humidity (condensation), significant air currents, and tiny insects. All of these can be misread as particles of combustion by the electronic circuitry monitoring the sensors.

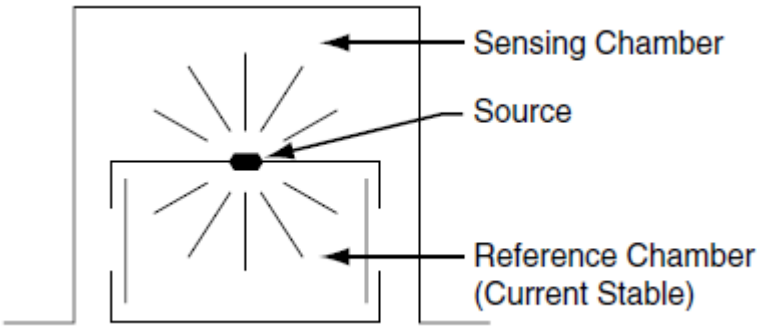


Figure 6.3: Dual chamber

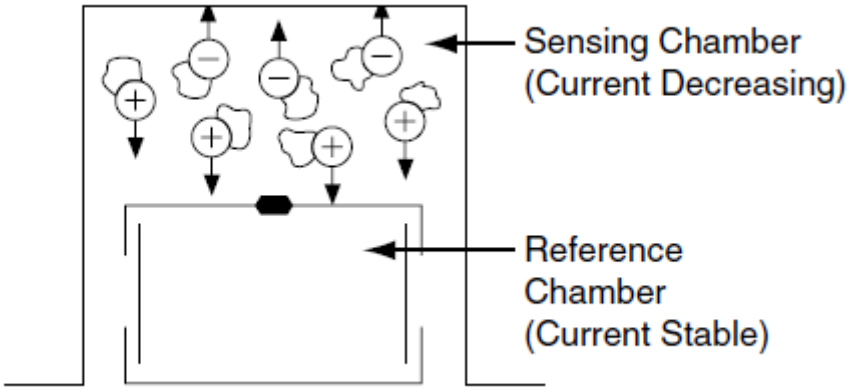


Figure 6.4: Dual chamber with particle combustion

### 6.1.2 Photoelectric Light Scattering Smoke Detector

The majority of photoelectric smoke detectors are of the spot kind and work on the theory of light scattering. A light-emitting diode (LED) is shone into a space that isn't normally visible to a photosensitive element, such as a photodiode. When smoke particles join the light path, light hits them and is reflected back onto the photosensitive system, causing the detector to respond.

A light source and a photosensitive receiving unit, such as a photodiode, are used in another type of photoelectric detector called a light obscuration detector. The reduction in light reaching the photosensitive system changes its performance as smoke particles partially block the light beam.

The detector's circuitry detects the change in output, and when the threshold is exceeded, an alarm is triggered. Obscuration detectors are typically projected beam detectors, with the light source spanning the covered region.

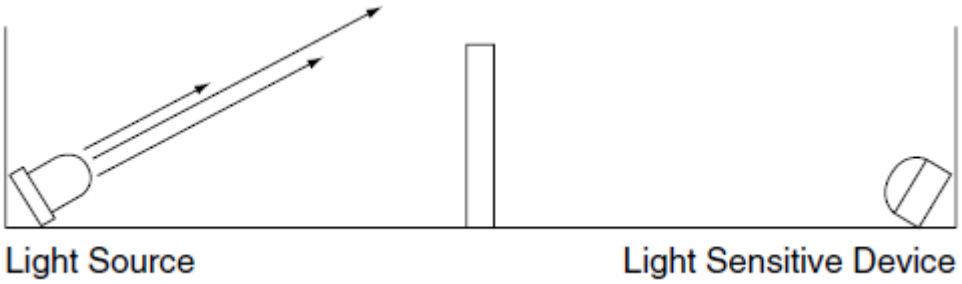


Figure 6.5: Light scattering detector

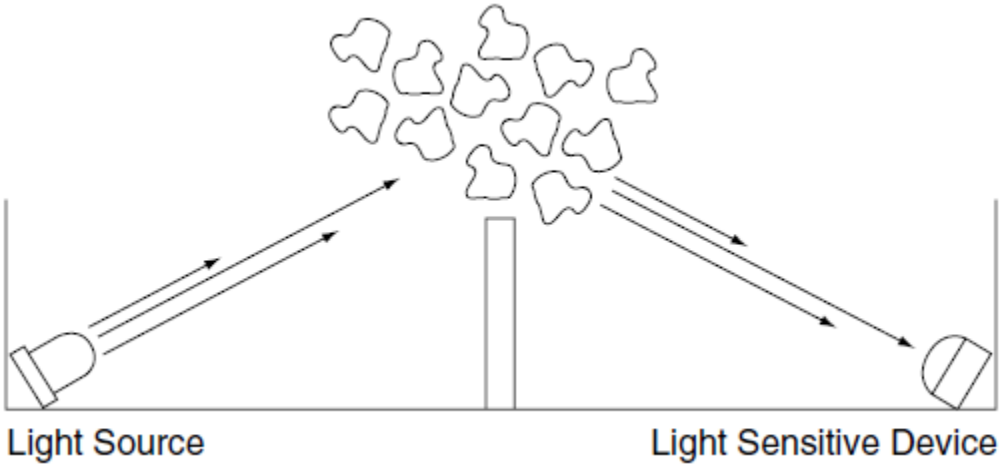


Figure 6.6: Light scattering detector with smoke

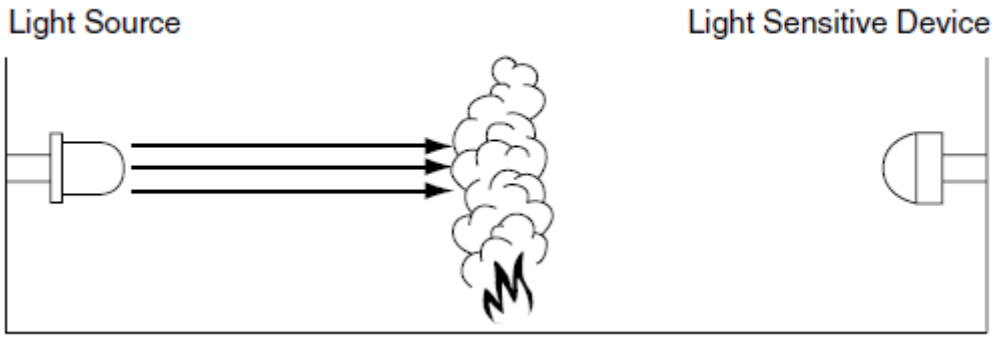


Figure 6.7: Light obscuration detector with smoke

## 6.2 Temperature Record

It is necessary to maintain or keep a record of the temperature as high temperate could cause damage to appliances. Keeping record using temperature sensor is easy. The temperature sensor can be placed in different places of a house. Data is passed via internet to the cloud specifically in this case Ubidots. SMS or email can be sent at over or under specific temperature.

## 6.3 Smart Efficient Lighting

A smart house is always careful about is efficiency. Power efficiency can make someone profitable as well as help the environment by saving energy. Depending on the environment lighting, efficient lighting can be done using light depending resistors. The record can be sent to cloud so that after a month, the cost of power usage can be seen with a graph. These automatic lights can be useful in gardens, parking, outside lighting of the house.

## 6.4 Fingerprint Based Activity Logger

Using fingerprint, we can easily track the entrance of our smart house. First we need to take images of our fingerprint. Fingerprint sensor uses optical image recognition for matching fingerprints. We can activate the door lock when sensor finds a match. The system can be turned on or off using

Ubidots API. So, when a match finds, it will send a post request to Ubidots API. In this way, one can check the timing of his entrance by checking the Ubidots historical data.

## 6.5 Smart Waste Bin

Wastage management can be a big hassle for an individual house owner. With smart wastage bin, when someone is near the bin at 30 cm, it will open up automatically. And, above 30cm, it will remain close. Another sensor is used for measuring the fullness of the bin. If it measures 3cm or less it will trigger an email to user that says the bin is full.

## 6.6 Experimental Results

To test sensors in different scenario, we will first need to connect the sensors properly with Arduino. First, we will use the smoke sensor to build a smart smoke detection system. The gas sensor connection with Arduino is given below.

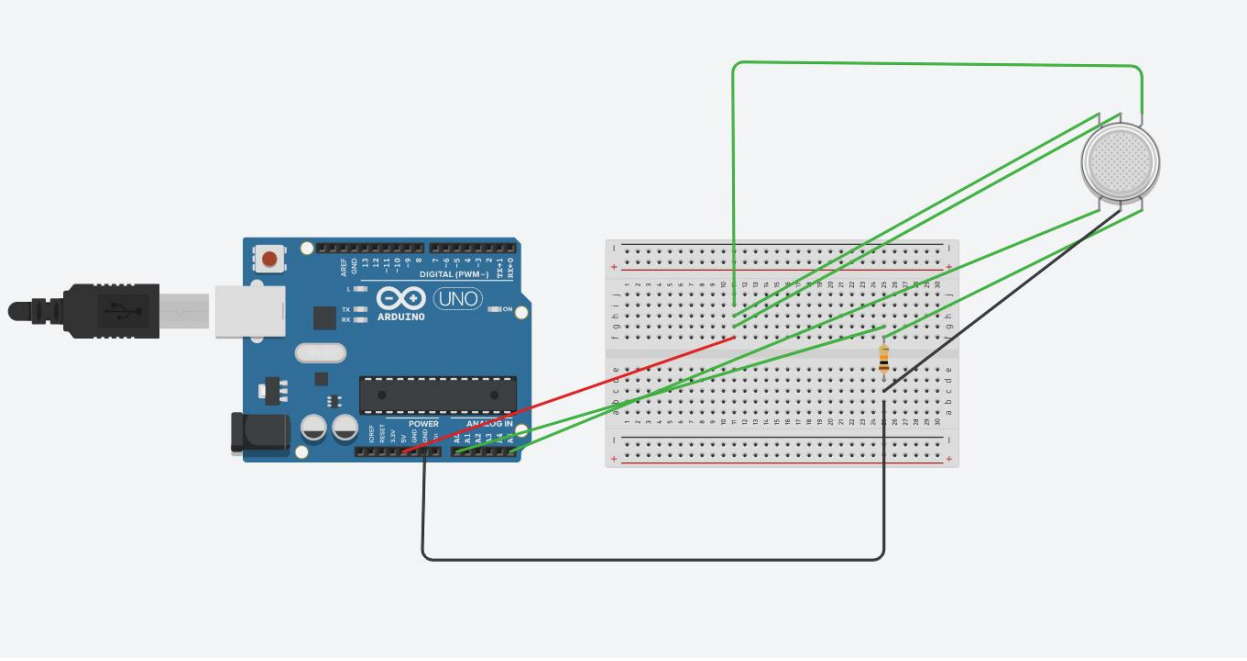


Figure 6.8: Arduino and gas sensor connection

The pin A0 is reading ADC value which can be used for detection of smoke. We will do a fire test. Production of unwanted gas can be a probable event of fire. We tested the sensor and got the below results.

| Sl | Analog Reading | Remarks |
|---|---|---|
| 1 | 420 | Without fire |
| 2 | 580 | With fire |
| 3 | 600 | With fire |
| 4 | 621 | With fire |
| 5 | 756 | With fire |
| 6 | 800 | With fire |
| 7 | 642 | Without fire |
| 8 | 625 | Without fire |

Table 5: Gas sensor experimental data

To measure the temperature, we will use lm35 module. The connection diagram is given below.

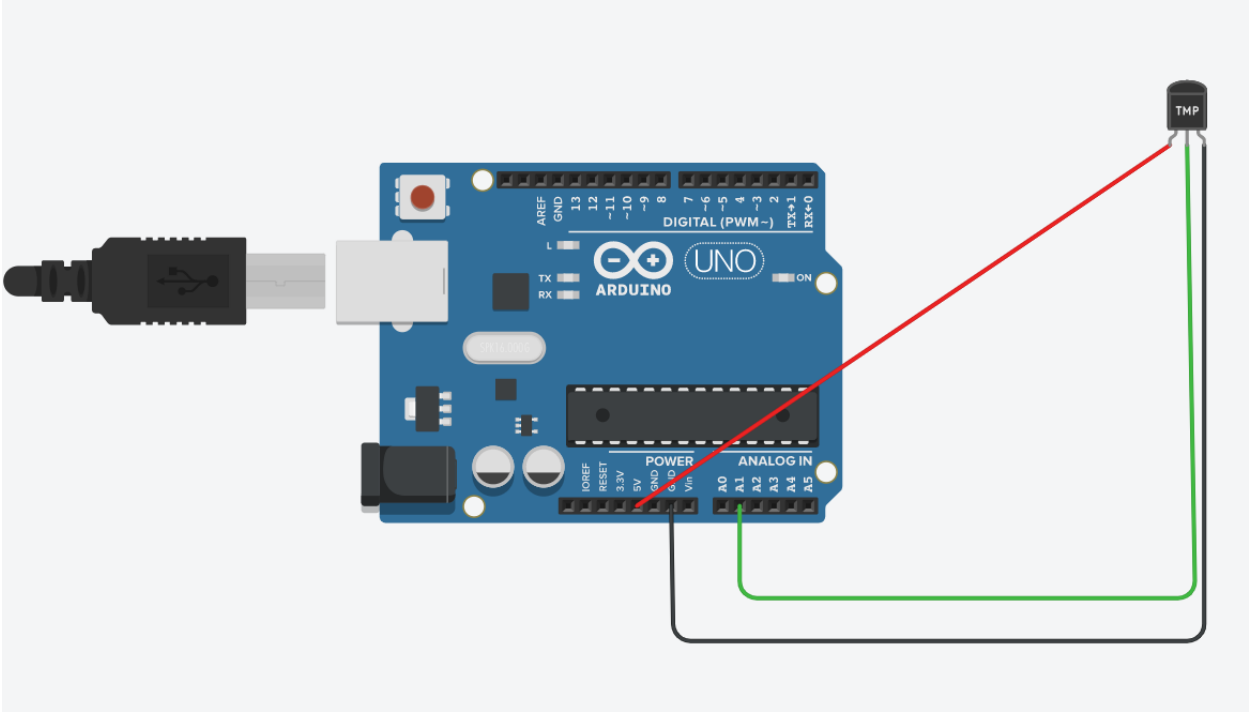


Figure 6.9: Arduino and temperature sensor connection diagram

Temperature sensor can be used for measuring critical temperature rise. The pin A1 is reading value for the temperature sensor.

| Sl | Temp (°C) | Remarks |
|---|---|---|
| 1 | 23.42 | Without fire |
| 2 | 23.42 | With fire |
| 3 | 27.69 | With fire |
| 4 | 27.96 | With fire |
| 5 | 34.03 | With fire |
| 6 | 36.54 | With fire |
| 7 | 35.99 | Without fire |
| 8 | 35.05 | Without fire |

Table 6: Temperature sensor experimental data

Ubidots has a graphical data interface that show the results at a specific date and time. Figure 6.11 shows the Ubidots graphical representation of table 6 sensor data.

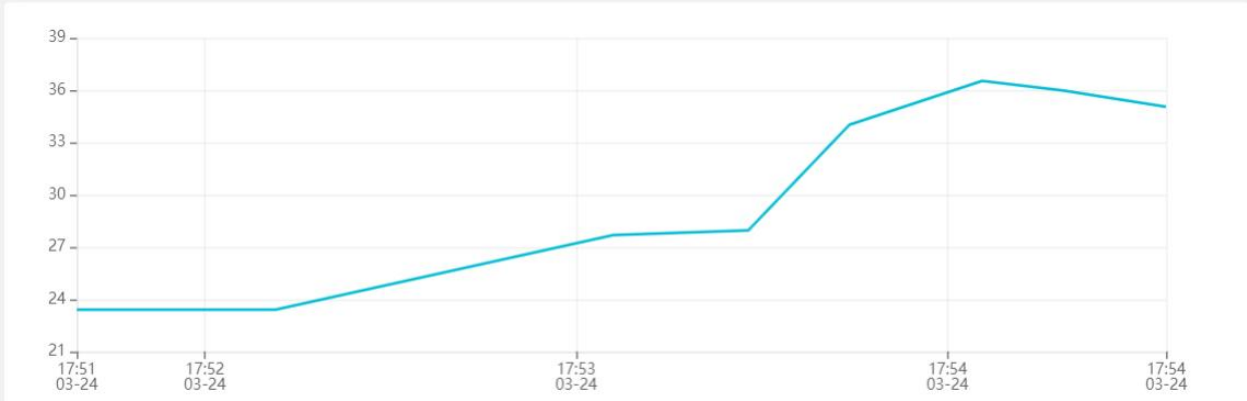


Figure 6.10: Ubidots temperature sensor data

With certain settings in the Ubidots, we can send SMS or email at specific sensor value or above or below a value and also with sending data from Ubidots to Arduino device, a machine or device can be turned on or off. In our case, we have set a condition of above or equal to 36 °C (read from temperature sensor) and above or equal to 750 (read from gas sensor) and set an email event in Ubidots if the condition meets. We set three events; one for temperature, one for gas and one for the 'and' condition. Figure 5.7 explains the procedure for events.

At 36.54 °C, we got an email from Ubidots, "Hey there, temperature was 36.54 at {date}{time} {GMT time}". Email can be configured in various ways using Ubidots given parameter. When the two condition met, we got a mail, "Hey there, it might be a danger." It was done by sending a third parameter to Ubidots. The "and" condition checking was done using Arduino. The email is customizable.

Device on or off time can be detected from the values. Ubidots saves records for certain date range so that user can easily check the historical data. They have pricing plans that allow users for a certain memory or cloud space for the data records.

For a smart bin, we are using ultrasonic sensor. 2 sets of sensor are used. One for tracking person near it, so that it can automatically open up, another one is for checking for the fullness of the bin. The connection diagram is given below.

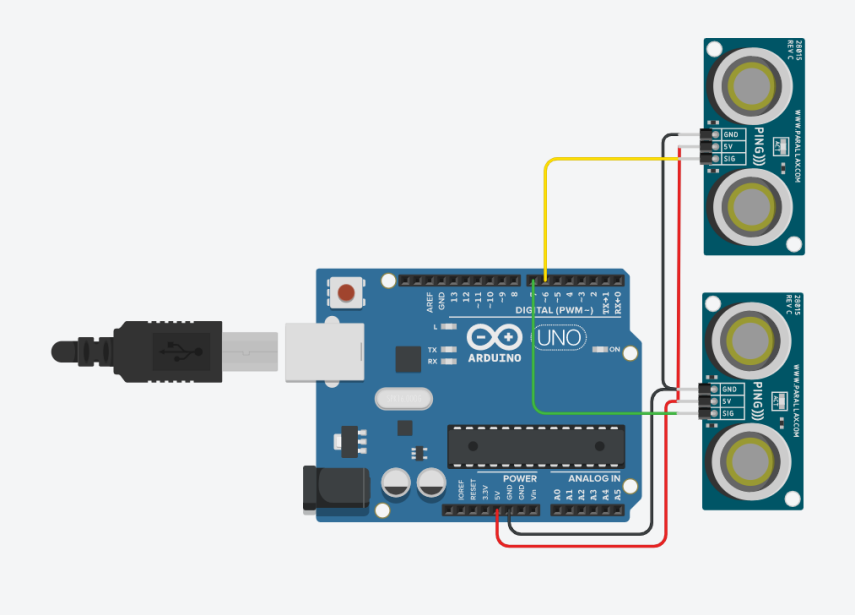


Figure 6.11: Ultrasonic sensor connection diagram

When sensor 1 senses equal or below 30cm, it triggers the 13 pin as high which can be used for opening the bin. The sensor 1 data taken is given below.

| **Sl** | **Distance (cm)** | **Remarks** |
|---|---|---|
| 1 | 42 | pin 13 low |
| 2 | 42 | pin 13 low |
| 3 | 36 | pin 13 low |
| 4 | 24 | pin 13 high |
| 5 | 24 | pin 13 high |
| 6 | 24 | pin 13 high |
| 7 | 86 | pin 13 low |

| | | |
|---|---|---|
| 8 | 86 | pin 13 low |

Table 7: Ultrasonic sensor experimental data

The same way sensor 2 is used. When it gets a value less than or equal 3 cm, Arduino will send a POST request to Ubidots. Then, based on the request, Ubidots send user a SMS saying that the bin is full.

Now, we will use the light depending resistor for measuring light and automatically turn on or off lights. We will use condition for analog reading equal or less than 400, then it is dark and lights will turn on. Anything beyond 400, light will stay turn off. Every minute the status will update to Ubidots so that we can track when the lights are turned on or off. A relay needs to be used when we are using LED light with 220V AC.

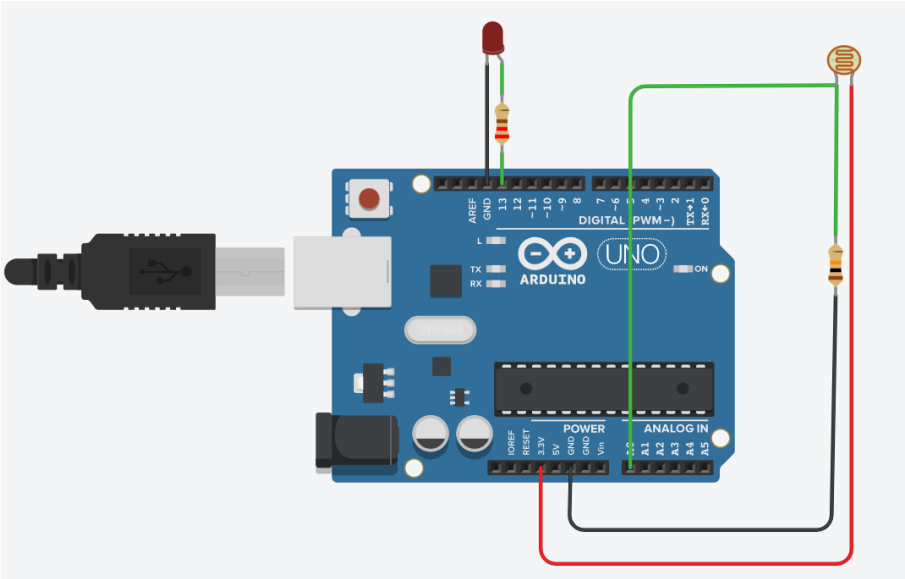


Figure 6.12: LDR connection with Arduino

As we are sending POST request every minute. So, we can see the activity of light on, off throughout the day. 1 is denoted as light turned on and 0 is denoted as turned off. The Ubidots data is given below.

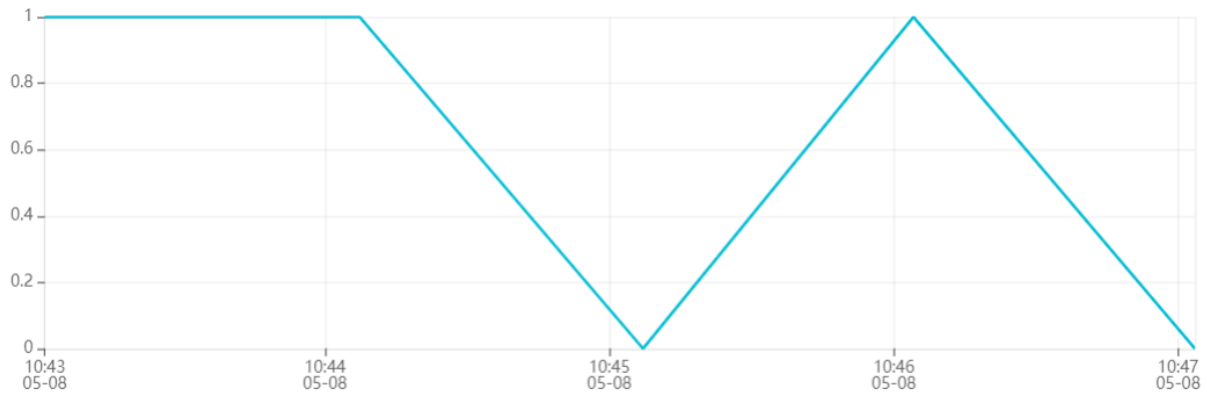


Figure 6.13: Ubidots light activity data based on LDR

Now we will take a look at the fingerprint based activity logger. The connection diagram of fingerprint sensor with Arduino is given below. First we need to store some finger prints. We will take 2 persons fingerprint, 1 will be finger print of person 1 and 2 will be for person 2. Now we will check in the detection mode. When it senses a match, it will send a POST request to Ubidots cloud. If it sends 1, that means it gets a match of person 1. Further coding can be done to use servo motor to control the door lock.

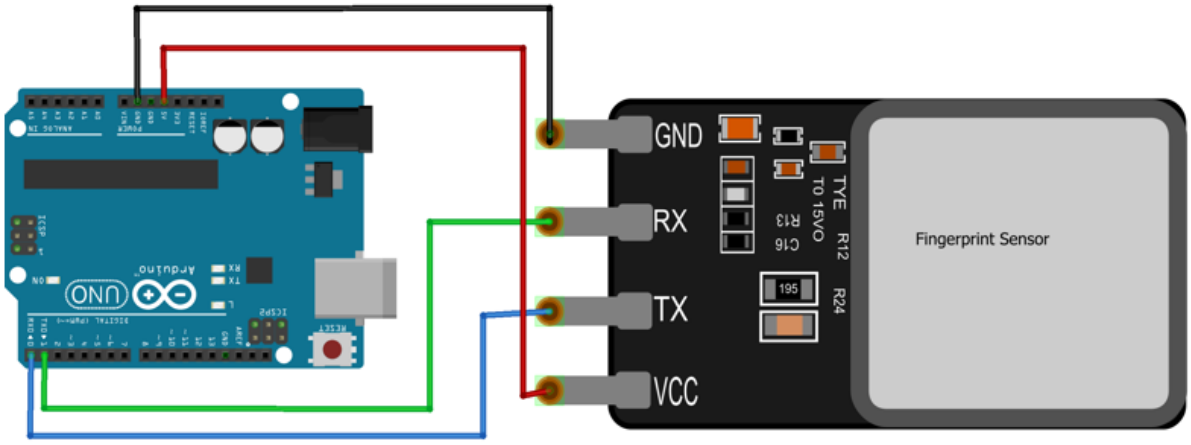


Figure 6.14: Arduino connection diagram with fingerprint sensor

So, every time a match is found, Arduino sends data to Ubidots. The Ubidots data is given below.

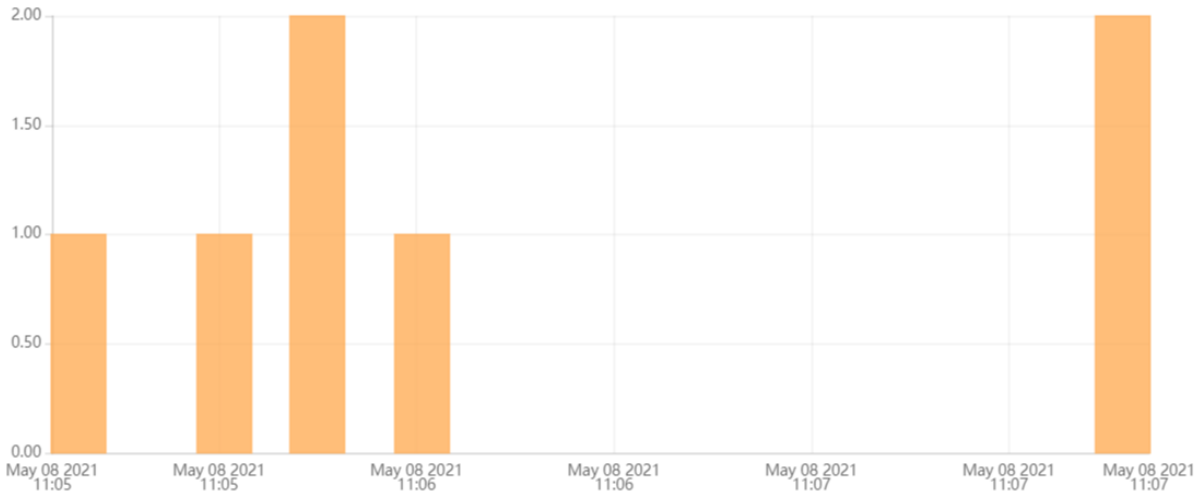


Figure 6.15: Ubidots data for fingerprint sensor

As we can see from the data, person 1 has 3 matches in different times while person 2 has 2 matches in two different times. From that result, we can see the record of activity of person 1 & 2 easily.

## 6.7 System Reliability

System reliability is an important part of a whole IoT ecosystem. Device data transfer must take place in a secure way so that data spoofing, hijacking or other issues don't occur. First, we will look at the cloud server side security measures.

- **HTTP with SSL Encryption:** HTTPS helps to protect data between server and devices by creating an encrypted environment.
- **Token Based Authentication:** Unlike traditional server-based authentication, where a username and password need to be sent in every request (constantly exposing them to potential attackers), token-based authentication assigns a signed token after the first request, which can then be used for sub-sequent requests. Ubidots refreshes token each time a request is sent. In this way, the security becomes strong.

Now, we will take a look at the user side security measures. We have used Arduino bridge library for Yun devices. The Bridge library simplifies communication between the ATmega32U4 and the AR9331. Bridge commands from the board microcontroller are interpreted by Python on the AR9331. Its role is to execute programs on the GNU/Linux side when asked by Arduino, provide a shared storage space for sharing data like sensor readings between the Arduino and the Internet, and receiving commands from the Internet and passing them directly to the Arduino. It creates a client that extends process for common CURL events. The request can be sent in an HTTPS server. So, it is secured. Again, as it creates a client on Linux, it is a more attack or malware free system as Linux is called a very secured and reliable system in the world. Therefore, using this well-developed Arduino library and a secured server side platform which is Ubidots, we have achieved a secured, reliable system.

# Chapter 7

# Challenges of IoT

## 7.1 Hardware Challenges

Since the controller is said to be a centralized gadget that serves as the network's brain, it is considered the primary source for attackers. There are a variety of traditional methods, and the action of various recent attacks is efficient and successful in disabling the controller. The controller, for example, can be the target of a classic DDoS attack. Aggressors will create a large number of fraud packets and send them all to the switches at the same time.

In switches, all fraud packets are treated as fresh, resulting in the least or equivalent amount of fraud transmission of requests to the controller. As a result, the controller's computational assets will be depleted in a short period of time.

In the eyes of attackers, the switch has also been deemed insecure. Since switches have the fewest hardware resources, attackers will first attack the channel of communication between controllers and switches; as a result, according to the OpenFlow protocol, the switches will most likely be changed to independent mode or fail-secure mode. The network's output will undoubtedly suffer as a result of this. So, to make the system reliable, we need to place the servers in a secured place.

## 7.2 Software Challenges

Through inserting malicious software into the IoT system, hackers or attackers have a significant effect on the system, resulting in a variety of outcomes. As a result, this type of attack has an impact on the system by refusing facilities, altering data, and gaining access to all personal information.

By using malicious code, the adversary will degrade the device. These codes are dissipated in this manner through email connections, which download all of the documents from the internet. As a result, the worm has the potential to replicate without the need for human interference. To discern the infection, we can now use the worm's indicator, IDS, and so on.

# Chapter 8

# Conclusion

## 8.1 Summary

This project depicts the possibilities of IoT devices implementation in a home automation system and the challenges to face. IoT has emerged as one of the important topics of research. It facilitates the interconnection of various objects and sensors to integrate with one another without the need for human intervention. The review of the IoT security threats and vulnerabilities has been shown. We have shown IoT architecture with multiple controllers. Then different sensors integration and working principle are described. The concept of using Ubidots is described as it can be a key software for industrial data cloud recording. Finally, different integration system and the challenges of IoT are described.

## 8.2 Future Work

The project is a work of scope of IoT application in a home automation system. As technology is evolving, there are more room for improvements. In future, machine learning and artificial intelligence can be implemented in different machine operation. New innovation in waste management, efficient water management [36-41], sewage management can be made. It needs further works and research to get these things done. The Internet of Things (IoT) is used all over the world. Wireless sensor networks allow sensing in any location around us, and these IoT devices must communicate with one another in a coordinated manner. On the one hand, these devices have many advantages in real life, but they are often vulnerable to multiple attacks. It's difficult to properly manage those instruments. When the entire procedure is dependent on a single central authority, an attack on that single core or any technical challenge to that central point can cause

that central point to fail. Blockchain was created with peer-to-peer networking in order to avoid such situations. There is no central server, and each node interacts with the others directly. Blockchain is ideal for IoT devices because of its anonymity, transparency, immutability, and decentralization. For better protection, further research and development is needed.

# References


[1] P. Seth, S.R. Sarangi, “Internet of things: architectures, protocols, and applications.” J. Electr. Comput. Eng. 2017, (2017)

[2] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, “An information framework for creating a smart city through internet of things.” *IEEE Internet Things J. 1(2),* 112–121 (2014)

[3] A. Giri, S. Dutta, S. Neogy, K. Dahal, Z. Pervez, “Internet of things (IoT): a survey on architecture, enabling technologies, applications and challenges.” In: Proceedings of the 1st International Conference on Internet of Things and Machine Learning, p. 7. ACM (2017)

[4] I. Froiz-Míguez, T.M. Fernández-Caramés, P. Fraga-Lamas, L. Castedo, “Design, Implementation and Practical Evaluation of an IoT Home Automation System for Fog Computing Applications Based on MQTT and ZigBee-WiFi Sensor Nodes.” *Sensors 2018,* 18, 2660

[5] M. Suárez-Albela, P. Fraga-Lamas, T.M. Fernández-Caramés, A. Dapena, M. González-López, “Home Automation System Based on Intelligent Transducer Enablers.” *Sensors 2016*, 16, 1595

[6] O. Blanco-Novoa, T.M. Fernández-Caramés, P. Fraga-Lamas, L. Castedo, “A Cost-Effective IoT System for Monitoring Indoor Radon Gas Concentration.” *Sensors 2018*, 18, 2198

[7] S. P Talari, M. Shafie-Khah, P. Siano, V. Loia, A. Tommasetti, J. Catalão, “A review of smart cities based on the internet of things concept.” *Energies. 10(4),* 421 (2017)

[8] O. Blanco-Novoa, T.M. Fernández-Caramés, P. Fraga-Lamas, L. Castedo, “An Electricity-Price Aware Open-Source Smart Socket for the Internet of Energy.” *Sensors 2017*, 17, 643

[9] T.M. Fernández-Caramés, “An Intelligent Power Outlet System for the Smart Home of the Internet-of-Things.” *Int. J. Distrib. Sens. Netw. 2015*, 11, 214805

[10] N. Z. Bawany, J.A. Shamsi, “Smart city architecture: vision and challenges.” *Int. J. Adv. Comput. Sci. Appl. 6(11),* 246–255 (2015)

[11] S.A. Al-Qaseemi, H.A. Almulhim, M.F. Almulhim, S.R. Chaudhry, “IoT architecture challenges and issues: lack of standardization.” In: Future Technologies Conference (FTC), pp. 731–738. IEEE (2016)

[12] Y. Li, F. Björck, H. Xue, “Iot architecture enabling dynamic security policies.” In: Proceedings of the 4th International Conference on Information and Network Security, pp. 50–54. ACM (2016)

[13] D.-M. Han, J.-H. Lim, “Design and implementation of smart home energy management systems based on zigbee.” *IEEE Tran. Consum. Electron. 56,* 1417–1425 (2010)

[14] C.-L. Wu, L.-C. Fu, “Design and realization of a framework for human–system interaction in smart homes.” *IEEE Trans. Syst. Man Cybern. 42,* 15–31 (2012)

[15] M.R. Alam, et al., “SPEED: an inhabitant activity prediction algorithm for smart homes.” *IEEE Trans. on Systems, Man and Cybernetics. 42*, 985–990 (2012)

[16] L. Chen, C.D. Nugent, H. Wang, “A knowledge-driven approach to activity recognition in smart homes.” *IEEE Trans. Knowl. Data Eng.* 961–974 (2012)

[17] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, “Internet of things for smart cities.” *IEEE Internet Things J. 1(1),* 22–32 (2014)

[18] S. Pradeep, T. Kousalya, K.A. Suresh, J. Edwin, “Iot and its connectivity challenges in smart home.” *Int. Res. J. Eng. Technol. 3,* 1040–1043 (2016)

[19] Cho Z.K., M.C.Y. Zhang, C.-W. Wang, C.-W. Hsu, C.-K. Chen, S. Shieh, “IoT security: ongoing challenges and research opportunities.” In: IEEE 7th International Conference on Service-Oriented Computing and Applications, pp. 230–234 (2014)

[20] K.S. Sahoo, B. Sahoo, A. Panda, “A secured SDN framework for IoT.” In: International Conference on Man and Machine Interfacing (MAMI), pp. 1–4 (2015)

[21] M. Conti, A. Dehghantanha, K. Franke, S. Watson, “Internet of Things Security and Forensics: Challenges and Opportunities.” Elsevier (2018)

[22] M. Nawir, A. Amir, N. Yaakob, O.B. Lynn, “Internet of Things (IoT): taxonomy of security attacks.” In: 3rd International Conference on Electronic Design (ICED), pp. 321–326 (2016)

[23] S.K. Tayyaba, M.A. Shah, O.A. Khan, A.W. Ahmed, “Software defined network (SDN) based Internet of Things (IoT): a road ahead.” In: Proceedings of the International Conference on Future Networks and Distributed Systems, p. 15 (2017)19. IEC TC 124 Strategic Business Plan. Available online: https://www.iec.ch/public/miscfiles/sbp/124.pdf (accessed on 31 October 2018)

[24] G.E.R.E. László, “An introduction and critical assessment of smart city developments.” *Deturope. 10(3),* 33–52 (2018)

[25] A.K. Kar, et al. (eds.), “Advances in Smart Cities: Smarter People, Governance, and Solutions.” *CRC Press, Boca Raton* (2017)

[26] S. Huh, S. Cho, S. Kim, “Managing IoT devices using blockchain platform.” In: 2017 19th International Conference on Advanced Communication Technology (ICACT). IEEE (2017)

[27] P. Raj and A. C. Raman, “The Internet of Things: Enabling Technologies, Platforms, and Use Cases.” Boca Raton, FL, USA: CRC Press (2017)

[28] European Committee for Standardization. SIS-CEN/TR 16298:2011: Textiles and Textile Products—Smart Textiles—Definitions, Categorisation, Applications and Standardization Needs; Technical Report; European Committee for Standardization: Brussels, Belgium, 2011

[29] SG 10 Wearable Smart Devices. Available online: http://www.iec.ch/dyn/www/f?p=103:85:0::::FSP_ORG_ ID,FSP_LANG_ID:12601,25 (accessed on 31 October 2018)

[30] T. Hughes-Riley, T. Dias, C. Cork, “A Historical Review of the Development of Electronic Textiles.” *Fibers 2018*, 6, 34

[31] A. Prahl, “Designing Wearable Sensors for Preventative Health: An Exploration of Material, Form and Function.” Ph.D. Thesis, University of the Arts London, London, UK, 2015

[32] T. Yang, D. Xie, Z. Li, H. Zhu, “Recent advances in wearable tactile sensors: Materials, sensing mechanisms, and device performance.” Mater. Sci. Eng. R Rep. 2017, 115, 1–37

[33] L. Allison, S. Hoxie, T.L. Andrew, “Towards seamlessly integrated textile electronics: Methods to coat fabrics and fibers with conducting polymers for electronic applications.” Chem. Commun. 2017, 53, 7182–7193

[34] M. Stoppa, A. Chiolerio, “Wearable Electronics and Smart Textiles: A Critical Review.” *Sensors 2014*, 14, 11957–11992

[35] T.M. Fernández-Caramés, P. Fraga-Lamas, “A Review on the Use of Blockchain for the Internet-of-Things.*” IEEE Access 2018*, 6, 32979–33001

[36] T. Robles, R. Alcarria, D.M. de Andrés, M.N. de la Cruz, R. Calero, S. Iglesias, M. López, "An IoT based reference architecture for smart water management processes." JoWUA. 6(1), 4–23 (2015)

[37] N. Ntuli, A. Abu-Mahfouz, "A simple security architecture for smart water management system." Procedia Comput. Sci. 83, 1164–1169 (2016)

[38] R. Nikhil, R. Rajender, G.R. Dushyantha, N. Jagadevi, "Smart water quality monitoring system using IoT environment. Int. J. Innov. Eng. Technol. 10(4), (2018).

[39] A. Purohit, U. Gokhale, "Real time water quality measurement system based on GSM." IOSR J. Electron. Commun. Eng. 9(3), 63–67 (2014)

[40] N.N. Beri, "Wireless sensor network based system design for chemical parameter monitoring in water." Int. J Electron. Commun. Soft Comput. Sci. Eng. 3(6)

[41] S. Wadekar, V. Vakare, R. Prajapati, S. Yadav, V. Yadav, "Smart water management using IOT." In: 2016 5th International Conference on Wireless Networks and Embedded Systems (WECON), pp. 1–4. IEEE (2016)