

Implementation of IoT based Smart Classroom: Room Occupancy Estimation, Automatic Seat Allocation and Environment Monitoring

by

Rakin Hassan

14321012

Sadia Salauddin

15121037

A thesis submitted to the Department of Electrical and Electronic Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering
Brac University
April 2020

© 2020. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis 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 thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rakin Hassan
14321012

Sadia Salauddin
15121037

Approval

The thesis/project titled “Implementation of IoT based Smart Classroom: Room Occupancy Estimation, Automatic Seat Allocation and Environment Monitoring” submitted by

1. Rakin Hassan (14321012)
2. Sadia Salauddin (15121037)

Of Spring, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc.in Electrical and Electronic Engineering on June 25, 2020.

Examining Committee:

Supervisor:
Farzana Shabnam

Farzana Shabnam
Senior Lecturer
Department of Electrical and Electronic Engineering
Brac University

Program Coordinator:
Abu S.M. Mohsin, PhD

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

Head of Department:
Professor Shahidul Islam Khan, PhD

Professor Shahidul Islam Khan, PhD
Chairperson
Department of Electrical and Electronic Engineering
Brac University

Abstract

IoT or Internet of Things was a term coined by Kevin Ashton in 1999. It is used to describe a system where different physical and digital devices are connected together and can communicate via the internet. The main goal of our project is to use IR sensors to count the total number of students that are entering and exiting the classroom and thus calculating the total number of students that are present. Furthermore, we are using Radio-frequency identification(RFID) to create a card scanning system that will allow us to allocate seats for the students during exam time. We also installed temperature, humidity and flame sensors inside the classroom which will give data in real time and can be used for both emergency or any regular data collecting purposes. The microcontroller we used to power all these systems was Arduino Uno. All the information collected by the Arduino will be uploaded to a server via NodeMCU or ESP8266 modules where a university administrator can access all the information in real time. Nowadays, most of the universities take attendance manually. This is a problem as manual attendance is not always fully effective. Calculating the total number of students using IR sensors will make it very easy for the lecturers to crosscheck with the attendance if need be. For one, student count can be a very important factor in emergency situations and the police and/or the fire department can need this information in such situations. Secondly, manually taking the attendance and crosschecking can be a tedious and time consuming task for the lecturers and this can save a lot of time. On top of that, students can give attendance for their friends who are not actually present in the class and getting this information can help reduce proxies. Also, the data collected can be used by the university or other people or organizations for R&D. In exam time, automatic seat allocation will help reduce hassle and save time. We plan to do all these in a very cost effective way.

Keywords: IoT; Internet of Things; Room Occupancy; Automatic Seat Allocation; Environment Monitoring; Arduino UNO; IR Sensor; RFID

Acknowledgement

Firstly, all praise to Allah for whom our thesis have been completed without any major interruption.

Secondly, to our Supervisor Mrs. Farzana Shabnam miss for her kind support and advice in our work. Her immense guidance helped us to get our thesis research completed.

And finally to our family for their support and prayers throughout our research period.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iii
Dedication	iv
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Nomenclature	ix
1 Introduction	1
1.1 What is IoT	1
1.2 History of IoT	1
1.3 Prototype	2
1.4 Aims and Objectives	3
2 Related Work	4
2.1 Smart Classroom	4
2.1.1 Different Methods of Counting	4
2.1.2 Preferred Method	5
2.1.3 Algorithms and Software Used for Prediction	5
2.1.4 Issues Faced	6
2.2 Smart Parking System	6
2.2.1 Proposed Architecture	7
2.2.2 Network and System Architecture	8
2.2.3 Implementation and Working	10
2.3 Water Management and Environmental Monitoring	11
2.3.1 Devices Used	11
2.3.2 Implementation	12

3	Hardware Mechanisms	13
3.1	Prototype	13
3.1.1	Model Description	13
3.1.2	Model Architecture	14
3.1.3	Images of the Hardware	15
3.2	Room Occupancy	17
3.2.1	Equipment Used	17
3.2.2	Model Description	17
3.2.3	Model Architecture	19
3.2.4	Images of the Hardware	20
3.3	Automatic Seat Allocation	21
3.3.1	Equipment Used	21
3.3.2	Model Description:	21
3.3.3	Model Architecture	22
3.3.4	Images of the Hardware	23
3.4	Environment Monitoring	25
3.4.1	Equipment Used	25
3.4.2	Model Description	25
3.4.3	Model Architecture	27
3.4.4	Images of the hardware	28
4	Algorithm	29
4.1	Occupancy Estimation	29
4.2	Seat Allocation	31
4.3	Environment Monitoring	33
5	Readings and Results	34
5.1	Hardware Setup	34
5.2	Project Implementation in Real Life	36
5.3	Results	36
6	Challenges,Solution, Application and Future Work	38
6.1	Challenges	38
6.2	Solutions	38
6.3	Application	39
6.4	Future Work	40
7	Conclusion	41
	Bibliography	43

List of Figures

2.1	Architecture of the Proposed System	8
2.2	Smart Parking System	9
2.3	Block diagram of the System	11
3.1	Model Architecture of the Prototype Design	14
3.2	(Top to Bottom) Prototype Before and after IR Interruption	15
3.3	Prototype During RFID Scan	16
3.4	Model Architecture of Room Occupancy and Automatic Seat Allocation . .	19
3.5	(Left to Right) IR Sensor Receiver and Transmitter	20
3.6	(Left to Right) Before and After Students Enter The Classroom	21
3.7	(Left to Right) RFID Checking In and Checking Out	23
3.8	Server showing Student Information	24
3.9	Server Showing Allocated Seat for Examinees	25
3.10	Model Architecture of Flame, Temperature and Humidity Detector	27
3.11	Hardware Image of Flame, Temperature and Humidity Detector	28
4.1	Flowchart of Occupancy Estimation	30
4.2	Pseudo Code of Occupancy Estimation	31
4.3	Flowchart of Seat Allocation	32
4.4	Pseudo Code of Seat Allocation	33
4.5	Pseudo Code of Environment Monitoring	33
5.1	Energy Conversion Laboratory Doorway	34
5.2	Hardware Image of Room Occupancy and Automatic Seat Allocation Sys- tem	35
5.3	Ongoing Energy Conversion Lab Class	36

List of Tables

2.1	Different Devices and Sensors Used	6
2.2	Nomenclature table.	9

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

IoT Internet of Things

RFID Radio-frequency identification

XAMPP Cross-Platform, Apache, MySQL, PHP and Perl

MEMS Micro-Electro-Mechanical Systems

GPS Global Positioning System

Chapter 1

Introduction

1.1 What is IoT

The Internet of Things, or IoT is a system where digital or mechanical devices, everyday objects like cars or houses, and even animals can be connected to the internet [28]. This is done by using sensors, microcontrollers and communication devices to connect to the internet and transfer data. These data can be uploaded to the cloud or stored locally where it can be used for multiple purposes. It can also be connected to other devices that are connected to the network and share data with each other if required. These happen without human beings controlling or interacting with these devices. The initial setup and the maintenance work is still obviously done by humans, but the core work that is already set up, keep on working as programmed.

Nowadays, the usage of IoT can be discerned in every aspect of life but it wasn't always like this. Now, it is widely used in many sectors including manufacturing, agriculture, law enforcement, real estate, retail, financial sectors, and so on [25]. Any object you can think of might already be part of the Internet of Things, without us even realizing it. The main objective of IoT is the collection and transfer of data and if it is working efficiently, work pressure can be shifted from humans to machines. As IoT is becoming more common, people are using their smartphones to control their homes among other things very easily. Before the introduction of IoT, people had homes with minimal security surveillance [12] and after IoT came into the scene, cameras could be connected to a Wi-Fi network and controlled via an app. Even the door could be controlled remotely via an app or voice.

1.2 History of IoT

While the importance of the presence of IoT is often considered of immense importance, it has not been present for a long period of time. On the contrary, the concept itself is fairly new, but there were evidence that the imagination of machines communicating with one other machines existed in the early 1800s [18].

One of the first official instance of IoT was seen before the name was even official. Around the early 80's, local programmer, David Nichols, from the Carnegie Melon University in Pittsburgh, Pennsylvania applied the concept of IoT on a Coca Cola vending machine situated at the same university premise [20]. He would link the refrigerated machine to the internet to check for the availability of the drink and also the status whether the drink is cold or not before going near the vending machine.

Finally in 1999, Kevin Ashton, Co-Founder of the Massachusetts Institute of Technology's Auto-ID Center, came up with the term "IoT". Ashton, while working as brand manager, conducted a presentation named "The Internet of Things" for the brand Proctor Gamble. He was supposed to help advertise a line of lipsticks [18]. But whenever he went to the local stores he found out that one particular colour of lipstick was always sold out. So he proposed the idea of tracking these lipsticks with RFID tags that would help them track and monitor the products. So in a way RFID was the predecessor to what we know today as IoT.

In post 2010, IoT has advanced to new heights and now it not only utilizes the internet but also wireless communication, MEMS and embedded systems. Ranging from Automation to wireless sensor networks, GPS and control systems, etc. support the IoT. The scope of IoT is always expanding and evolving over time and its possibilities are endless.

1.3 Prototype

IoT can be used in numerous ways. For example, a car uploading its data can both be beneficiary for the customer and the manufacturer. Like if a car's tyre has been damaged, this will show up immediately on the car's dashboard for the owner to see and take action as required. The information that will be uploaded to the manufacturer's server can be used to improve their products and/or update the customer's warranty information. IoT can also be used to make a smart home, where all your home appliances can be controlled by your smartphone from anywhere in the world as long as you are connected to the internet. It can also be used in the medical field. For example, if you are wearing a smart band that is monitoring your blood pressure, diabetes and other things and your body suddenly shows signs of abnormality. The device can be programmed to directly contact your doctor. The list of its functions is numerous. Statistics show IoT is an ever-growing field and is growing exponentially in the billions [13].

Sutjarittham et al. [16] wrote their paper on IoT based sensors to measure the attendance in a classroom and AI to predict its future behavior. Since they found out by statistics that in a university's class, enrolled students and students attending a class were not always equal, they believed that their algorithm would be more economical for the university in both terms of money and space. Pham et.al. [6] researched on the Smart parking system using IoT. A smart parking strategy was developed to decrease the cost of employing individuals and optimize the use of funds for car-park owners in the creation of traffic management schemes. The common method of parking requires a lot of factors, such as time, luck, experience etc. and as a result the complications increase in a city with high vehicle density. However, this is not an ideal solution because the car park could generally be far from the location of the customer. In latest years, study has used vehicle-to-vehicle communication. It can be seen that IoT can also be used to do Environmental monitoring [10] of a smart campus by using Wireless Sensor Network and Arduino. IoT based water management and distribution system [4] using a low-cost ultrasonic water level sensor and a wireless network using a sub- GHz campus scale is also seen.

1.4 Aims and Objectives

The main goal of our project is to use IR sensors to count the total number of students that are entering and exiting the classroom and thus calculating the total number of students that are present. Furthermore, we are using Radio-frequency identification(RFID) to create a card scanning system that will allow us to allocate seats for the students during exam time. We also installed temperature, humidity and flame sensors inside the classroom which will give data in real time and can be used for both emergency or any regular data collecting purposes. The microcontroller we used to power all these systems was Arduino Uno. All the information collected by the Arduino will be uploaded to a server via NodeMCU or ESP8266 modules where a university administrator can access all the information in real time. All these information will help in emergency situations, save time, reduce proxies and help in data collection which can be required for any reason. We tend to do all these in a very cost effective way.

Chapter 2

Related Work

2.1 Smart Classroom

In the paper Sutjarittham et al. [16] took attendance of their university classes and made prediction based on their algorithms, which they believed will be cost effective. The implementation was done in four steps. The first step was by using sensors to measure class occupancy and compare and contrast the cost, accuracy, privacy and understand its pros and cons of the different methods and sensors. After experimenting with a few methods, they decided on using beam sensors to do the counting. The second step was by monitoring and taking data using their preferred counting method from 9 lecture halls and 250 courses over a period of 18 weeks and they also released this data to the public. The third step was by using AI to successfully predict future classroom attendance using three algorithms which were multiple regression, random forest and support vector regression with RMSE error as low as 0.16. The fourth thing they implemented was that they allocated rooms to courses based on this predictive attendance rather than by enrollment. After predicting the attendances and allocating the classrooms accordingly, they found out they could save the university 10% of room costs with minimal risk of student overflow.

2.1.1 Different Methods of Counting

We have seen other authors using different methods of counting in other papers. Gaonkar et al. used a combination of door sensor and PIR [15]. Sutjarittham et al. also decided to study a few works by other authors and then tried a few sensors and devices for counting. They avoided the sensors that send the data they collected back to the manufacturers as this would be a data breach as well as extra expenditure. The commercial sensors they used were- EvolvePlus Wireless Beam Counter, EvolvePlus Overhead Camera, Steinel HPD Camera (pre-market release), and Steinel Presence Detector. They also got two Wi-Fi access points from the university to compare the results for which they had to get ethics clearance. The Beam Counter is an Infra-Red sensor that is mounted on the door frame that counts the number of people entering and leaving through the doorway every 30 seconds. The Overhead Camera is a thermal sensor and works similar to the Beam Counter. Both these data are transferred to the main database through the internet. The HPD Camera is set in the corner of the room where there is full view. It is a people counting sensor that uses built in image processing to count every 30 seconds. The Presence Detector is something that gives a binary indi-

cation if the room is empty or not but not an actual people counter. The Wi-Fi AP data was not live, but it was available to them the next day at 7 am. All these data were taken and stored in the master database. Applying sensors throughout the school was not possible due to cost restraints so they used each type of sensor on each of the classrooms.

2.1.2 Preferred Method

After trying the above sensors, they decided on using the beam counters. The beam counter was easily installed by themselves but the positioning was tricky, as it is required to be mounted at a waist level. It is battery powered and the battery can last around a year. It is the most accurate of the four especially when the doorway is narrower. It uses an external gateway which is priced around a thousand dollars but it can be the most cost effective among the four. So firstly, the beam counters communicate their count data to sensor gateways that are installed in each room via wireless protocol. The gateway then communicates with the university private VLAN using Ethernet where the data is stored. Then the data is converted into a particular format and stored in the master database.

2.1.3 Algorithms and Software Used for Prediction

The authors explained in more detail the machine learning algorithms they used to predict the attendance after they collected the data. They used Multiple Regression (MR), Random Forest (RF) and Support Vector Regression (SVR). Commonly used loss functions such as MSE and MAE treats under and over prediction equally. This should not be the case as under prediction is actually more harmful as many students might not be able to attend the class for it. To make up for this difference Quantile Loss Function was used. Ordinary least square and Quantile Regression were also used for loss functions for each. After tinkering a bit, they found the perfect value of the Quantile Parameter to be 0.75 which gives a RMSE value of 0.2. For predicting upcoming semester's attendance, they saw that many things change (student lifestyle, availability of course, change in faculty etc.). They found out for this the best algorithm is Quantile RF. They had four sets of constraints (Course, room, capacity and schedule) with their individual formulas where the variables included courses, classroom availability, costs etc. To solve the algorithm, they used Constraint Programming (CP) algorithm using Google optimization tools for Python. They used CP and tried various methods and obtained few outcomes based on the method. In each case they noted down the results and also tried to find the overflow classes from it. The authors revealed their outcomes. They found that if they used the enrollment-based method the cost remained the same. But when using the attendance-based allocation the cost fell gradually with each week. Counting actual attendance obviously gave the best result but it was not feasible. Their prediction-based allocation lowered the cost by 12% and adding a 20% margin lowered the cost 5% more. Coming to class overflow, the enrollment and attendance based had no class overflow obviously but the prediction based caused an overflow of 1.25-2.70% and goes down to 1% by adding a 20% margin.

2.1.4 Issues Faced

They worked with the campus staff to do the aforementioned counting and calculating. One method of counting was door based which they did with the entry and exit doorways. But more than one person entering a room together would mess up the count. They also tried a course based method where they counted for each individual course. For the first few days they had actual people counting the students in a classroom and then they calculated the difference in the two counts which was not feasible. They also used a method called MAPE (Mean Absolute Percentage Error). The occupancy data is always counted whereas the attendance is only counted during class hours.

Devices Used	Factors Taken into Account	Used in Final Implementation or not
Beam Counter	<u>Installation</u> : Can be done by oneself <u>Accuracy</u> : High <u>Cost</u> : Moderate <u>Privacy</u> : Not Privacy Invading	Used
Thermal Sensor	<u>Installation</u> : Needs a professional technician <u>Accuracy</u> : Low <u>Cost</u> : Expensive <u>Privacy</u> : Not Privacy Invading	Not Used
HPD Camera	<u>Installation</u> : Moderate issues <u>Accuracy</u> : Moderate <u>Cost</u> : Moderate <u>Privacy</u> : Moderate Privacy Invading	Not used
PIR Sensor	<u>Installation</u> : Needs a professional technician <u>Accuracy</u> : Just gives a binary indication if people are present or not <u>Cost</u> : Moderate <u>Privacy</u> : Not Privacy Invading	Not Used

Table 2.1: Different Devices and Sensors Used

2.2 Smart Parking System

The idea of intelligent campus has achieved a good deal of popularity in latest times. Thanks to the evolution of the Internet, the concept of an intelligent campus now seems to be feasible. Consistent attempts are being produced in the field of IoT to maximize the productivity and efficiency of urban infrastructure. Problems such as traffic congestion, limited car parks and road safety are addressed by IoT. Thanh Nam Pham et. al (2015) [6] showed a study where vehicle-to-vehicle and vehicle-to-infrastructure was used to interact with the assistance of multiple wireless network techniques such as radio frequency detection (RFID), Zigbee, wireless lunch network and the Internet. This research aimed at providing driver data about neighboring parking spaces and making a reservation minutes sooner using supported equipment such as smartphones or tablet PCs. In addition, the services used each vehicle's ID when booking

a parking space. This was also replicated by Khanna and Anand (2016), who also proposed a smart parking system which was implemented using a mobile application connected to the cloud [9]. The scheme enables the user to understand the accessibility of parking spaces in real time.

2.2.1 Proposed Architecture

Cloud computing and IoT have experienced a major evolution. Both techniques have their benefits, however many shared benefits may be expected from their inclusion. On one side, IoT can tackle its technological limitations, such as storage, processing and power, by leveraging Cloud's infinite capacities and resources [5]. On the other hand, cloud can also extend its reach to cope with real world entities in a more distributed and dynamic way through the use of IoT. Both of the projects used these things as their main tools. Thanh Nam Pham et al. used the concept of IoT [28]. The system utilized the RFID technology WSN [1] to monitor vehicle parks. An RFID reader counted the proportion of free parking spaces in each parking lot. RFID promoted the execution of a large-scale scheme at low price. The scheme offered a mechanism for avoiding conflicts in the car park and helped minimize wasted time in searching for a parking space. The user can select an appropriate parking space after logging into the scheme. Information on the chosen parking place will be verified by notification to the customer and also updating the status as 'pending'. If the system determines that no vehicle is parked in that room after a certain period of pending time, it shifts the status to 'available'. The system will update the status from the WSN node (the parking space status) when a fresh vehicle joins the scheme. Therefore, the general parking system status is always updated in real time. The scheme will assist the parking time for each parking space in real time and can support the company with hourly parking fees. The smart IoT parking system which had a cloud-based server, allowed a driver to search and find information on parking spaces from each car park without the need to directly access the local server node by directly accessing it. The local unit consisted of control unit, screen and RFID tag or ID Card. Another thing which was associated with the architecture is the software client, which allowed users to install it on their smartphones and operate with 3G/4G mobile connections. Khanna and Anand used a technology which consists of a storage capacity which is required for collecting, accessing, processing, visualizing and sharing large amounts of data. Cloud offers unlimited, low-cost and on-demand storage ability, making it the best and most cost-effective solution to handle IoT-generated information. Data gathered from multiple sensors is generally transferred to more strong nodes where it can be aggregated and processed. Then, by using built-in apps, IoT devices could monitor and control stuff in real-time through distant places. IoT includes the use of instruments of a heterogeneous nature. These systems may have distinct hardware or software settings as a consequence of compatibility problems. It is very hard to guarantee interoperability between these devices in an IoT setting. Cloud helps to address this issue as it offers a common platform where different devices can communicate.

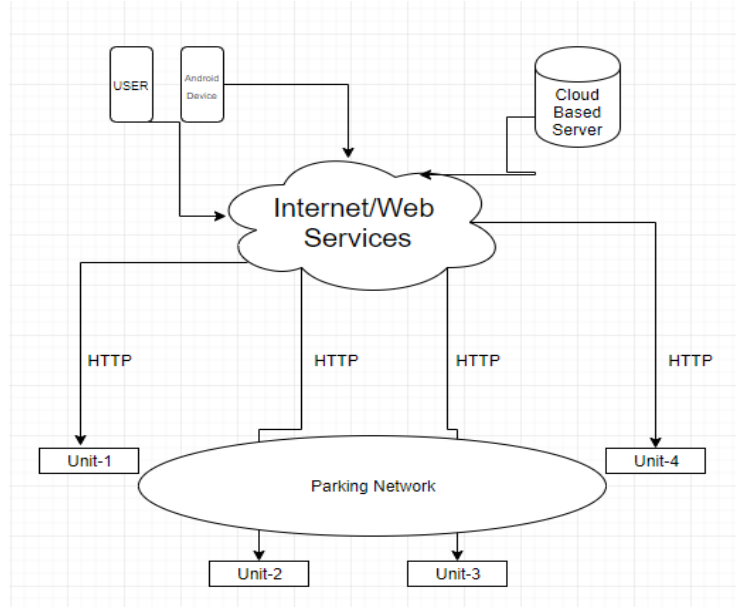


Figure 2.1: Architecture of the Proposed System

Figure 2.1 depicts that a user is connecting to see the availability of a parking slot with the help of internet/web services. The cloud based server is gathering all the information of the availability. The HTTP request is sent to know and acknowledge the availability for a spot per unit.

2.2.2 Network and System Architecture

The network architecture was built upon a parking network and the neighboring table of nodes construction. The authors used the infrastructure/backbone parking network (CPN) architecture. A wired and a wireless connection were used. This sort of parking network involves routers that act as the linked customer infrastructure. The CPN infrastructure/backbone can be constructed to enable sensor networks to communicate using wireless radio systems. The routers form a network of self-configuring and self-healing links. Gateway features can be used to connect routers to the Internet. This strategy also referred to as infrastructure meshing, offered the backbone for standard customers and allowed routers to integrate CPNs with current WSNs through gateway/bridge functionality. Conventional customers with the same radio techniques as routers can interact directly with routers. It had been assumed by the authors that each car park is a node in a CPN. The system architecture used a high level architecture along with a mathematical model. The parking scheme that we suggested is made up of different actors working in sync with each other. Below is the mathematical model that describes the smart car park scheme.

Symbol	Meaning
PT	Parking Time
CN	Driver's Car Number
AP	Amount Paid
ID	User ID
PS	Parling Slot
Di	Driver
OR	Occupancy Rate
$X_i()$	Input
$Y_i()$	Output
$CF()$	Computation Function
$I()$	Identity Function

Table 2.2: Nomenclature table.

$D_i = X_i(PT, CN, AP, ID, PS)$ // Driver provides input to the input function

$X_i() = CF(PS, PT)$ // Input function notifies the computation function

$X_i() = I(AP, CN, ID)$ // Input function notifies the identity function

$OR_i = CF(PS, PT) = Y_i()$ // Computation function notifies the output function and the resultant is stored in form of the occupancy rate.

$OR_i = 0$ or 1 // Occupancy rate can either be 0 or 1. Where 0 specifies occupied (red) and 1 means vacant (green).

The following Fig. 2.2 gives an outlined view of the complete system.

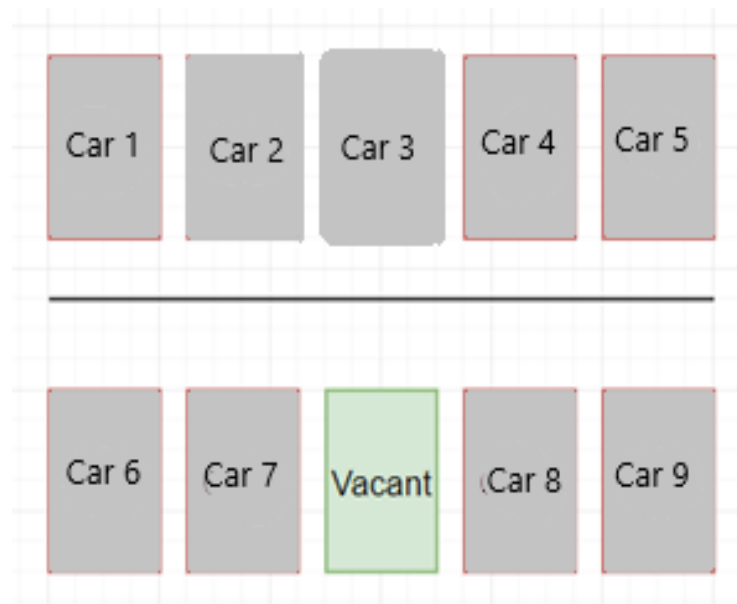


Figure 2.2: Smart Parking System

In the figure, the green box indicates vacant parking space and the grey box indicates occupied parking space.

One of the main actors is a parking sensor which is made of infrared, passive infrared and ultrasonic sensors. The ultrasonic sensors are wirelessly linked to the raspberry pi by using the ESP8266 chip. The ESP8266 WiFi chip consists of a self-contained SOC

with an embedded TCP/IP protocol stack that enables any microcontroller to access the Wi-Fi network. Another actor is the Raspberry Pi which is a processor on chip. A python script running in the chip checks the status of various GPIO pins and updates the information to the IBM MQTT server through MQTT protocol over a channel which is used on top of the TCP/IP protocol. One of the primary actors is the mobile application. The mobile application functions as an interface for end-users to communicate with the system. The application is created in the Apache Cordova and Angular Js frameworks using JavaScript as a programming language. The aim of using Apache Cordova is to produce apps that can run on android and iOS platforms with the same source code. The request is linked to the IBM MQTT server through a secure channel and a 2-factor permission. The aim of this mobile application is to provide data on the accessibility of parking spaces and to allow the end user to book a slot accordingly. Transfer of data usually happens on JSON format.

2.2.3 Implementation and Working

When a user wishes to locate a parking slot, they need to log in their scheme. A request message is sent to search for a free parking slot after effective login. The system will then send back a reply signal containing the data, including the address of the car park and the instructions to reach it. Parking selection is based on the feature $F(\alpha, \beta)$, which is calculated based on the vehicles present place and parking place. If the present car park is full, the system will forward the vehicle to a car park with a minimum $F(\alpha, \beta)$ value. When the customer arrives at the car park, he/she must be allowed to enter. This permission is obtained through the RFID technology or by scanning the user card. This mechanism is easy but economic. If the data is right, parking is permitted to the user. If the present car park is complete, the system will send a suggestion signal that contains data on a brand new car park, including email and fresh instructions, at a minimum price. The fresh car park will be chosen based on the present car park's neighboring table.

Another process contains a fact of booking a parking slot, parking a car in that slot and leaving the parking area is explained with the help of the following flow chart.

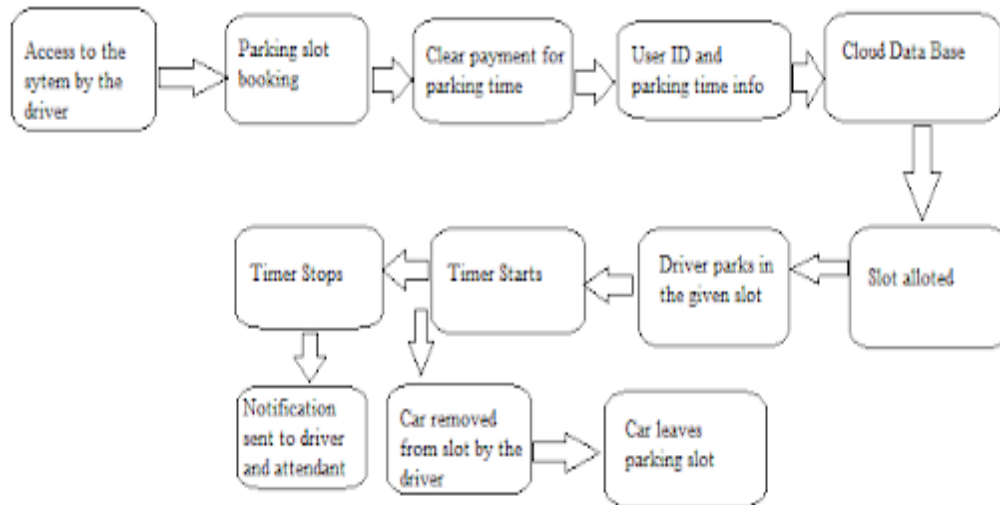


Figure 2.3: Block diagram of the System

2.3 Water Management and Environmental Monitoring

The authors Sastra et al. [10] has worked to make the Universitas Udayana into a smart campus by implementing environmental monitoring of the campus. They came up with two ways on how they could measure the data and upload it to the internet. They also built a web server to present the data to the internet. The first scenario is the one they had started to work on and can only be used when there is Wi-Fi service available. The second scenario is the one they were looking to work on next and would be useful when the sensors are placed in remote locations and there is no Wi-Fi service available. Water scarcity is severe in the modern work. To save water, a water development system was developed by Verma et al. [7]. Their design includes an IoT based water management system using Sensors and a wireless network.

2.3.1 Devices Used

Enviromental monitoring: WSN or Wireless Sensor Network is a network that has sensor nodes and a processor. These sensors are placed in various places where one would want to measure a particular physical quantity. The sensors all measure data separately but pass the data together in one central hub. If someone wants to add video surveillance to it they would need a camera and the network would be known as Visual Sensor Network. The first device they talk about is Arduino. Arduino is an open source microcontroller. Both the software and hardware are designed for ease of use. The board has both analog and digital I/O. It can be powered on by connecting to the PC via USB, using AC to DC adapter or even batteries. The particular board they used here was Arduino Mega 2560. A protocol they used is XBee 802.15.4. This module provides wireless endpoint connectivity to devices with IEEE 802.15.4 networking protocol for point to multipoint or peer to peer networking. The last device they talked about was ESP8266 Wi-Fi module. This is a microcontroller which has built in Wi-Fi module. It

can either be connected to an Arduino to give it Wi-Fi access without using a shield or it can work on its own, acting both as a programmable microcontroller and a Wi-Fi router. Its power consumption is also very low.

Water management: The design proposed in the paper by the author Verma et al. is to use a low cost ultrasonic water level sensor that can detect the level of water up to 10m in depth, as well as the volume of the water in the overhead tanks and ground level reservoirs. The authors tested their designs in the IISc campus. They tested two different designs. First design was by installing buoy or pressure sensors at the end of the tank to take readings. The readings received from these pressure sensors were however tedious. A huge disadvantage of this design was that the sensors, used at the bottom of the tanks, contaminated the tank, making the water unsuitable to use. The second design proposed was by using ultrasonic sensors with sub-GHz communication network which covers maximum distance. However, another problem that rose with this design was that the ultrasonic sensors available in the market were expensive. The challenge was to make the ultrasonic sensor design cheap and feasible. To achieve this low cost design, ultrasonic transducers were placed under the lid of the tanks, which will transmit signals and then pick up the reflected ones by the water surface and in total give the dimension of the tank filled with water.

2.3.2 Implementation

Weather monitoring: The authors Sastra et al. [10] had two scenarios for two purposes set in place. Both of them used sensors such as temperature, humidity, flame, sound, gas, rain sensors and even a camera. In scenario one, they would only connect ESP8266 modules to the sensors and the data collected would directly be uploaded to the internet. The second scenario used Arduinos connected to these sensors. The Arduino in turn was connected to 802.15.4 XBee modules which act as the Wi-Fi routers. The information was transmitted via the XBee to another receiving XBee which had a Arduino Wi-Fi shield connected to it and through there, the data was uploaded to the internet or web server. They also created a web server using Ruby on Rails on Ubuntu with MySQL database. The code was provided by thingspeak. They further clarified that communication between client and servers used HTTP POST and GET methods and security used WRITE and READ keys.

Water management: The network design by Verma et al. consisted of three nodes- 1) gateway node (receives data from the sensors) 2) end node (connected to the sensors) 3) relay node (modified end node, both transmits and receives data from their own node). A tree network topology is used connecting the ground level reservoirs (GLR) and the overhead tanks (OHT). Each GLR communicates with the closest OHT and transmit data to the gateway node, which is vaulted in the administration building of the campus. The sensor nodes supposedly need to be maintenance free, so they are powered by the solar panel of rating 12W/12V and charges the batteries used in the system. The system could run upto 10 days without any replacement of the battery.

Chapter 3

Hardware Mechanisms

3.1 Prototype

3.1.1 Model Description

At first we created a prototype to get an idea of the whole project. This was like a testing phase and we used the ideas from here to make our final project. Our main aim for this prototype was to estimate the classroom occupancy at a given moment as accurately as possible and to allocate a seat for the students entering the classroom. After reading and studying different methods, we decided on using our current model/method taking into account the accuracy and the cost. We decided on using Arduino Uno to make the project. We connected the ATmega328p chip found in Arduino Uno to a breadboard and connected all other peripherals with it. Two IR sensors were connected to the chip and with the LCD display. One RFID module was also connected to the chip and the LCD. An ESP8266 WiFi module was used to connect to the internet and pass data to the server and keep the server's information updated. A potentiometer, 16 MHz Crystal and a reset button were also used. To calculate the number of students entering and exiting the classroom, two IR sensors were placed side by side. Let's take the sensor closest to the door to be sensor 'A' and the one further away to be sensor 'B' to understand the logic. So, if a person was entering the doorway from outside, 'A' would cut first and then 'B'. So, this would be taken as someone entering the classroom. If a person was exiting from the classroom, 'B' would cut first and then 'A'. This reading would be taken as output. It could also happen a student either entering or exiting a classroom could change his/her decision halfway and does not enter/exit. In that situation only one of the sensors would be cut and it would not be taken as either entry or exit. Both the IR sensors had to be cut for an event to occur. Also, when a student entered or exited, the LCD display on the breadboard would display the information and update in real time. So, if there were three students who entered the classroom, the IR sensor pair could be cut three times and the LCD display would show "Students: 3". To make the second part of the project, which was allocating a seat for the students, we used RFID board, key tag and cards. This could be useful during exam time. Each student would have either a card or a tag allocated to them. When a student entered the class, they would have to scan the allocated card to the RFID board and the student's name, id and a seat number allocated to them would be displayed on the LCD screen. If they scanned it again it would be read as check out. The ESP would also send this data to the server and this info along with the time of check in and check out would

be shown on an administrator's screen. We used an external power adaptor instead of connecting to a PC using USB cable to ensure it got enough power. A power supply was used to give power to the whole system, with the microchip and LCD connected to the 5V and the RFID connected to the 3.3V port of the power supply. Equipment used for our model are:

- 1) Power Supply
- 2) 16*2 Character Display
- 3) ATmega328p Chip
- 4) MFRC-522 RFID Module
- 5) IR Sensors
- 6) ESP8266 WiFi module

3.1.2 Model Architecture

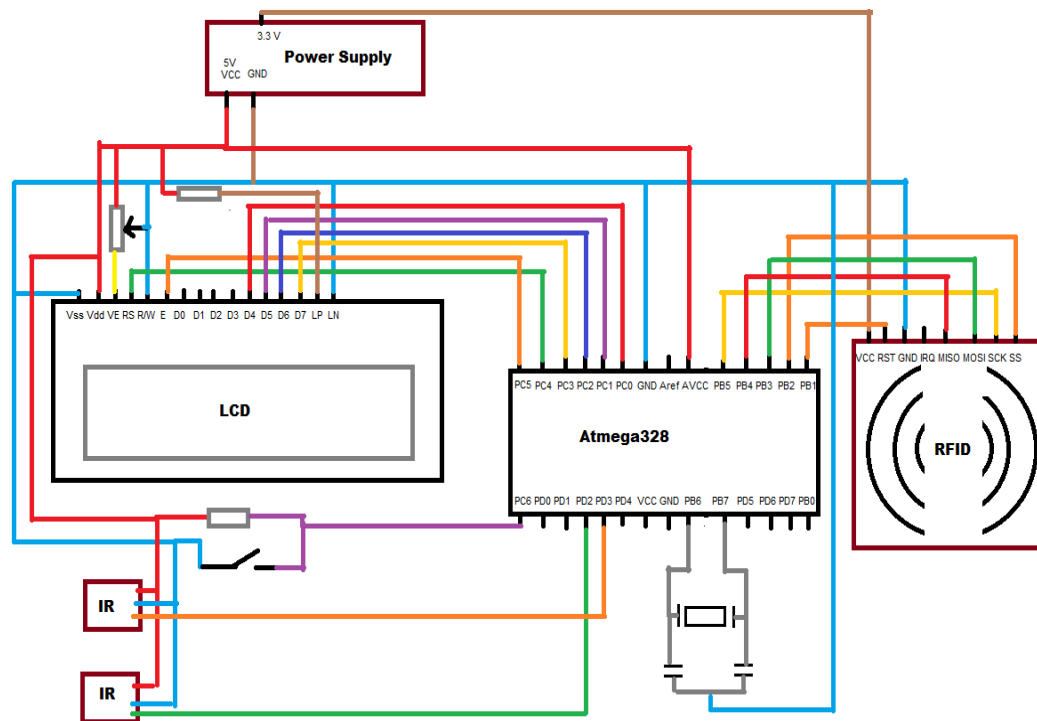


Figure 3.1: Model Architecture of the Prototype Design

This shows our whole connection diagram. A power supply was used to power the whole system, where the 5V was used to power the ATmega328p chip, the LCD display, the IR sensors and the 3.3V was used to power the RFID module. The diagram also shows the 16 MHz crystal and the reset button.

3.1.3 Images of the Hardware

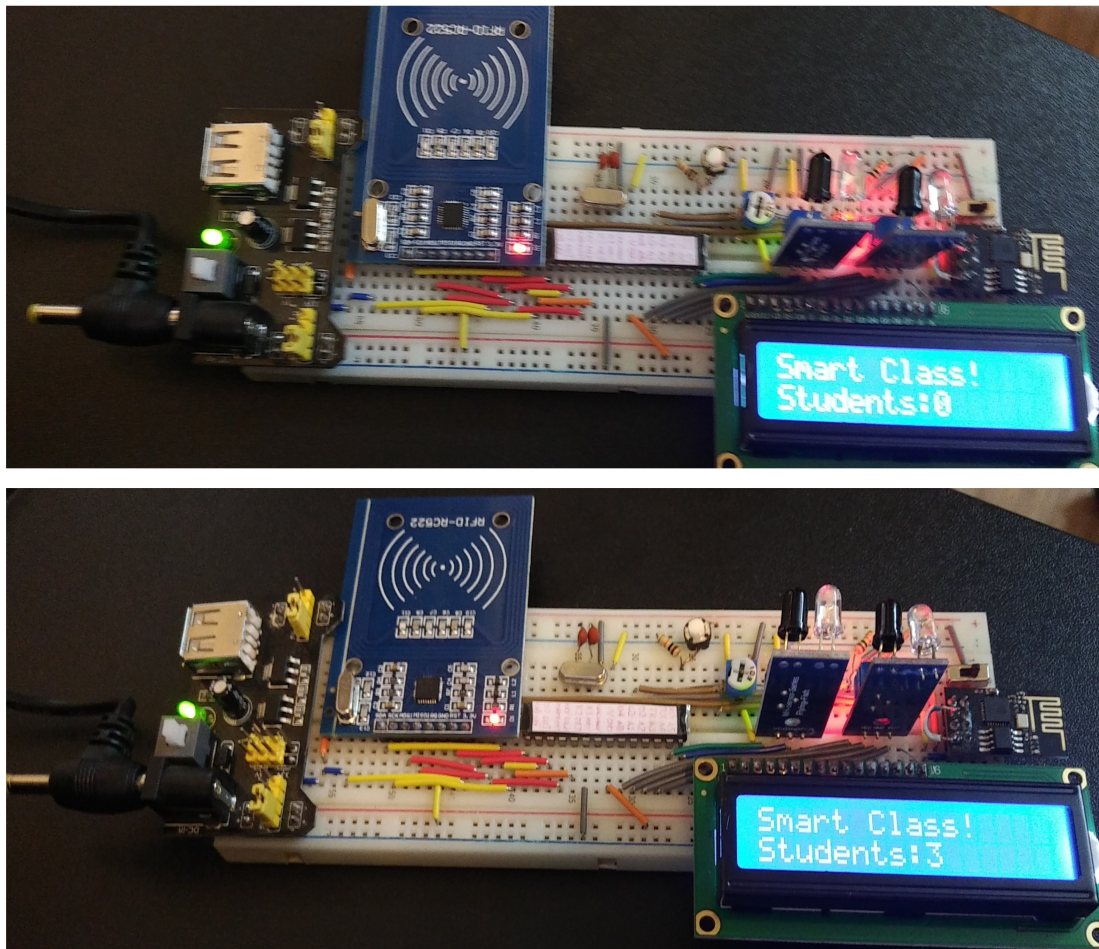


Figure 3.2: (Top to Bottom) Prototype Before and after IR Interruption

The top image is what the display will show after the system fully initializes. Since no students has entered yet and the IR sensors has not been cut the display shows "Students: 0". The image at the bottom is after the IR sensors has been cut as students enter. The display will change and show the number of students who entered. In this case the sensors were cut 3 times as 3 students has entered and thus the display shows "Students: 3".

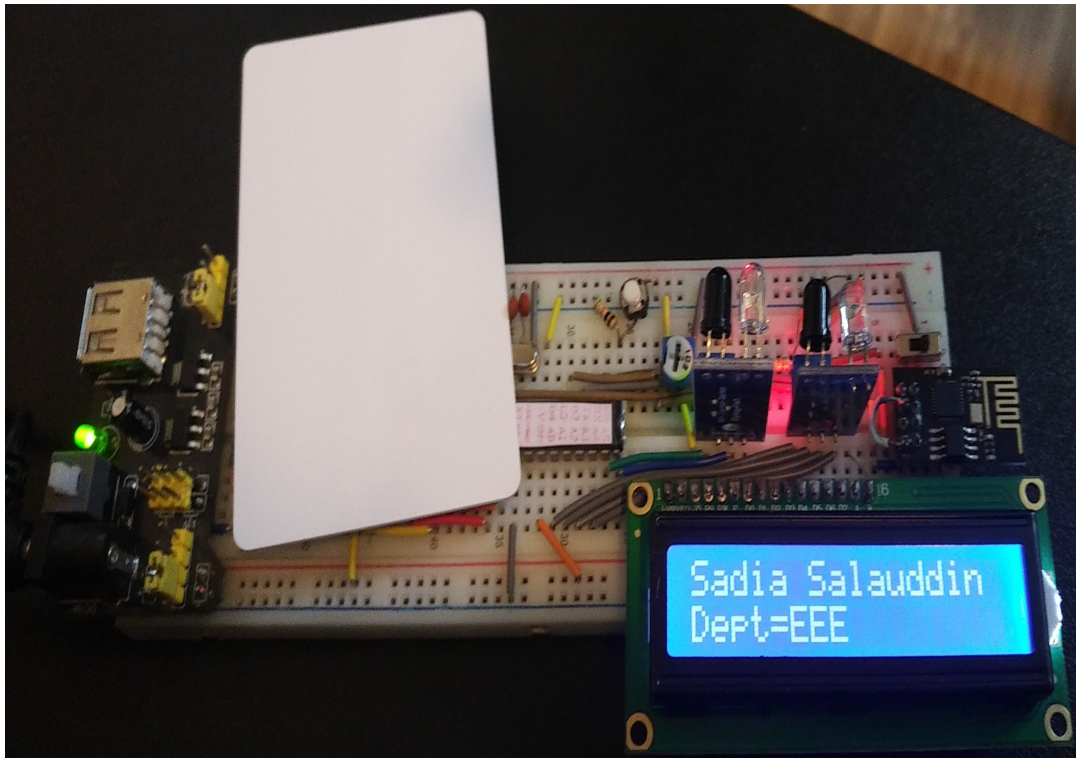


Figure 3.3: Prototype During RFID Scan

When the RFID card is scanned the display will show the name and department of the student who scanned the card.

3.2 Room Occupancy

3.2.1 Equipment Used

- 1) Power Supply
- 2) 16*2 Character Display
- 3) Arduino UNO
- 4) NodeMCU - V1 (ESP 12E Module)
- 5) 2 IR Transmitter and Receiver sets

3.2.2 Model Description

Taking attendance manually can have a lot of disadvantages. To diminish the problems we have developed an IoT based smart, cost effective and easy occupancy sensing method. This is not only diminishing the problems of manual attendance but also creating extra advantages.

Firstly, one of the most important aspects to occupancy counting in a classroom is for emergency situations. Fire/Police departments always wants the data of students present in an university in times of emergencies.

Secondly, at least 5-10 minutes is spent to take the attendance of the whole class. An average university class is around 80 minutes and if 10 minutes is wasted every class taking attendance, for every 8 classes, one class is wasted. This can be a problem for both the faculties and the students.

Thirdly, it can get tiresome for the faculties to take attendance and crosscheck with the actual students being present in the class. This can increase the chance of proxies and fake attendances, since every university has a mandatory rule that students need a certain percentage of attendance to sit for the final exam [11].

Finally, all this data that is collected can be useful for students, teachers, departments and organizations wanting to use the data for research purposes.

In our project, the main microcontroller we used was an Arduino UNO, which was connected to a 5V power supply. The 240V from the outlet is transformed into 9V while entering the power supply, which then provides the whole system with 5V. There are two sets of 5V and GND pins with the power supply as seen in fig. 3.4. One set is connected to the Arduino UNO and the other one is connected to both the transmitting and receiving IR sensors. Infra Red(IR) sensors are electronic devices that can transmit and receive electromagnetic waves within the wavelength of 700nm to 1400nm [24]. There are two types of IR sensors, active and passive [22]. We needed and used the active type, which had its own two variations. First we tried with Reflectance IR sensors, which had the transmitter and receiver at the same side. The transmitter would emit IR rays and when it hit an object, it would reflect back the ray to the receiver and a signal would be received. The one that we decided to use due to reasons mentioned later, is called break beam sensor. In this particular IR sensor, the IR transmitter is at one end and at the other end is the IR receiver. The IR transmitter continuously emits IR wave into the receiver until an object comes in between them and the beam is cut off. This loss of signal can be used in any way we desire when connected to the Arduino. In our project, the two IR sensors were placed at the bottom where the door opens. At the left side of the door the receiving IR is placed and at the opposite side is the transmitter. The receiving IR has been fitted with aluminum foil so that there is

less interference. The working principles of the IR sensors are same as the prototype we created. Let's take the sensor closest to the door to be sensor 'A' and the one further away to be sensor 'B' to understand the logic. So, if a person was entering the doorway from outside, 'A' would cut first and then 'B'. So, this would be taken as someone entering the classroom. If a person was exiting from the classroom, 'B' would cut first and then 'A'. These readings would be taken as the output data. It could also happen a student either entering or exiting a classroom could change his/her decision halfway and did not enter/exit. In that situation only one of the sensors would be cut and it would not be taken as either entry or exit. Both the IR sensors had to be cut for an event to occur. These data would be transmitted from the the IR sensors through the Arduino and then be displayed on the screen. When a student entered or exited, the LCD display on the breadboard would display the information and update in real time. So, if there were two students who entered the classroom, since the IR sensor pair could be cut two times the LCD display would show "Students: 2".

Arduino UNO's 5V is used to power the LCD display while the 3.3V is used to power the NodeMCU. A NodeMCU is also connected to the Arduino, which is an open source Lua-based firmware [23]. The particular development board we used has a ESP8266 12E module, which is a microcontroller unit with a built in Wi-Fi module. There was a more recent and powerful board on the market called ESP32, but we decided to go with ESP8266 as it was cheaper. Lua language is generally used to write code for NodeMCU but we can use Arduino IDE to program it as well, which is what we did. The Tx pin of the Arduino is connected to the Rx pin of the NodeMCU and the Rx is connected to the Tx. The Arduino transfers data to the NodeMCU through UART communication to be uploaded to the server. The NodeMCU collects the data and connects to the access point, which is a cell phone in our case, and sends the data to the server. The server displays this data of how many students are present in the classroom to be accessed by the administrators.

3.2.3 Model Architecture

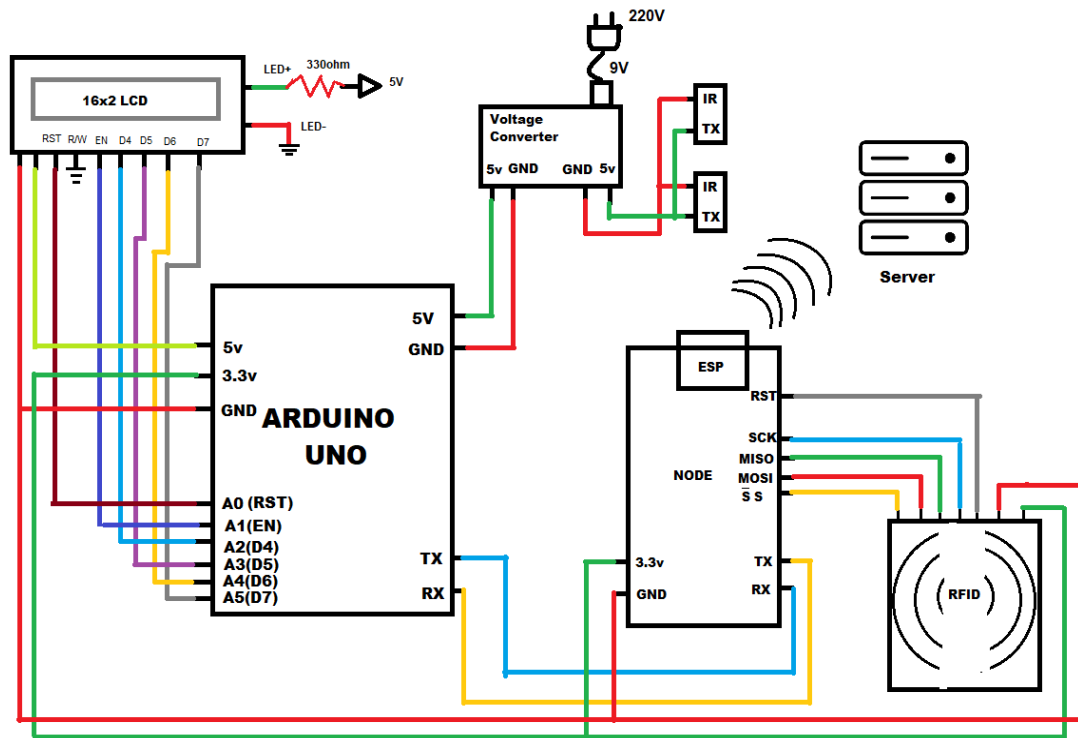


Figure 3.4: Model Architecture of Room Occupancy and Automatic Seat Allocation

This shows the model architecture for both the room occupancy and automatic seat allocation systems. The Arduino UNO is connected to all the devices including LCD display, IR sensors, NodeMCU and RFID modules. The NodeMCU powers the RFID module but transfers the data to the Arduino so that it can be shown on the display. The Arduino transfers its data to the NodeMCU so the data can be uploaded to the server. The hardware image of this architecture was shown in fig. 5.2

3.2.4 Images of the Hardware

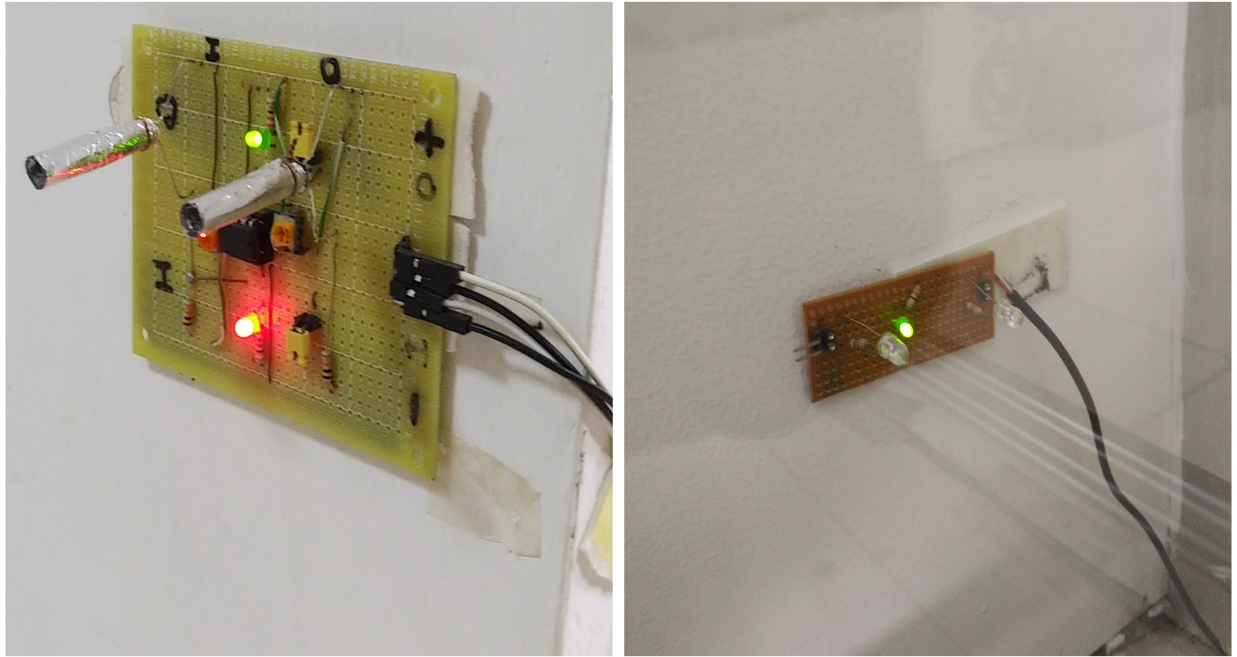


Figure 3.5: (Left to Right) IR Sensor Receiver and Transmitter

The IR sensor receiver and transmitter is shown here. The sensors are placed around 1 feet above the ground. The receiver and transmitter has to be properly aligned for it to work. The LEDs glow to show that the connections are working properly and the alignment is all right. When students get in or out and the sensors are cut, the LEDs will stop glowing during that duration. The receiver IR sensors were covered in aluminum foil as there was too much interference from outside and was not working as desired. After using the foil the result was greatly improved.

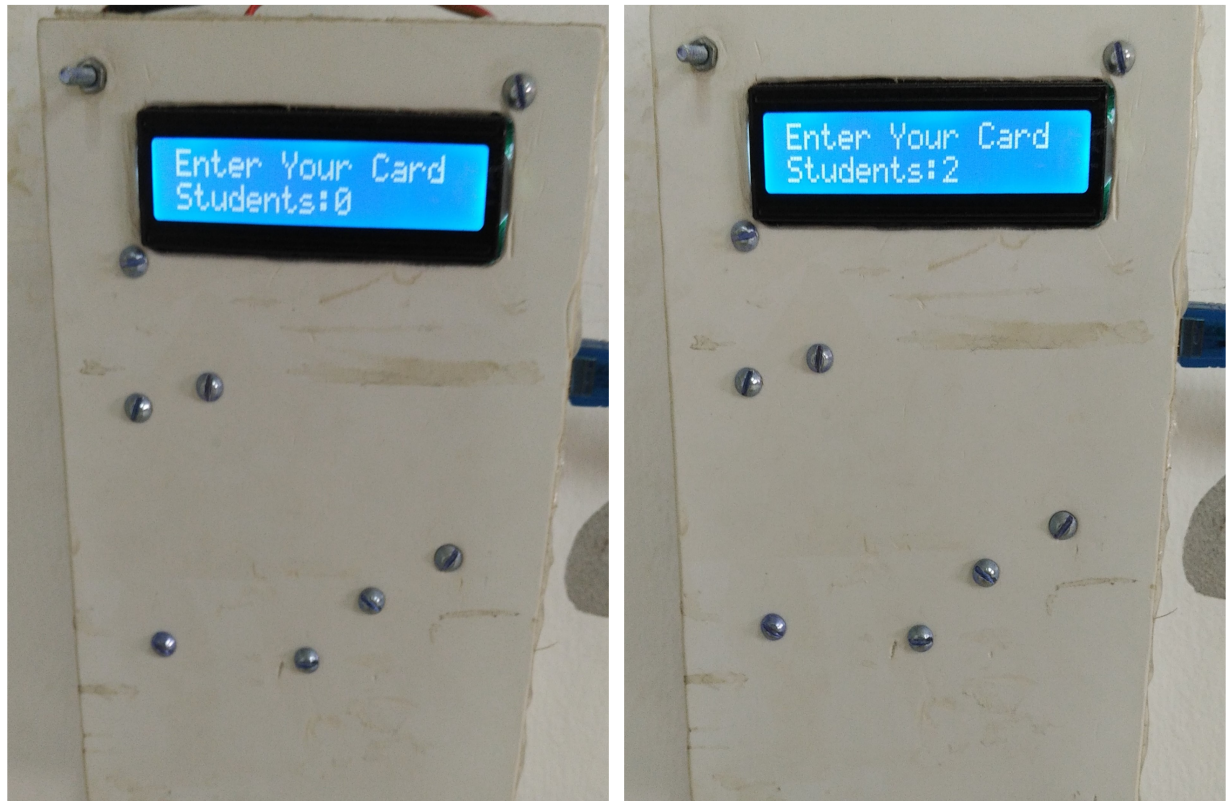


Figure 3.6: (Left to Right) Before and After Students Enter The Classroom

These images show the LCD display set at the entrance of the classroom. The image on the left indicates that the sensor has not been cut and thus no students have entered the classroom yet. The image on the right shows the display after the sensor has been cut twice, indicating that two students have entered the classroom.

3.3 Automatic Seat Allocation

3.3.1 Equipment Used

- 1) Power Supply
- 2) 16*2 Character Display
- 3) Arduino UNO
- 4) NodeMCU - V1 (ESP 12E Module)
- 5) MFRC-522 RFID Module
- 6) RFID Card and Tag

3.3.2 Model Description:

Automatic seat allocation is another part of our project. This can be seen as an extension to the occupancy estimation that we have mentioned in section 3.1. Allocating a seat automatically will reduce hassle and save time even further. It will especially be very useful during exam time for both the students and faculties. The students do not have to look for their seats but directly go and sit to their assigned ones. Now, looking from the teacher's perspective, they would have easier time identifying the students

and crosschecking with the server's data. In order to make this happen, we have used RFID technology.

RFID or Radio Frequency Identification is a technology which can be used to track objects, humans etc and uses radio frequency waves to pass on information and data [27]. RFID can be of two types, active and passive [26]. In our project we have used the passive type which does not have an internal battery and needs an external power source to run. RFID belongs to a group of technologies called Automatic Identification and Data Capture (AIDC) [3], which is a type of technology that identifies objects automatically and collects these data to transfer it to a server without human interaction. RFID is made up of three elements, RFID tag, reader and an antennae. RFID tag is made up of a radio receiver and a transmitter. As shown in fig. 3.4, the RFID module is connected to the NodeMCU and communicates through SPI communication. 3.3V is required to power the RFID module. The connection between RFID module and NodeMCU is as follows- The SCK of the RFID module is connected to the SS of the NodeMCU, MISO with MOSI, MOSI with MISO, SS with SCK and RST with RST. This part also works similar to the prototype we created. When the RFID card is scanned at the reader, the data is sent to the NodeMCU. Since the RFID we used was an Integrated RFID reader, it contained an internal antennae already. RFID card or tag has a unique product code much like bar codes and when the card is scanned that antennae at the reader communicates with the tag. The NodeMCU then uploads the data to the server after connecting to the Access Point, which then gives connection to the server. Here the university administrators will have full viewing control. When the NodeMCU scans, it gets a numeric value 1-9. Values 1 and 2 are for Rakin Hassan, 3 and 4 are for Sadia Salauddin and 9 for unregistered. These numeric values are sent to the Arduino Uno by the NodeMCU through UART communication, which then displays the students name, ID and allocated seat in the LCD display as shown in fig. 3.8. This could be useful during exam time. Each student would have either a card or a tag allocated to them. When a student entered the class, they would have to scan the allocated card to the RFID board and the student's name, id and a seat number allocated to them would be displayed on the LCD screen. If they scanned it again it would be read as check out. There is also a visual representation of the filled and empty seats displayed on the server. The green colour indicates that the seat has not been allocated and the red indicates that the seat has already been allocated to some student.

3.3.3 Model Architecture

The model architecture for the Automatic Seat Allocation section is at 3.4

3.3.4 Images of the Hardware



Figure 3.7: (Left to Right) RFID Checking In and Checking Out

The images show the display before and after the scanning of the RFID card. The display will show the full name of the students, the ID and if they have checked in or out. The left images indicates a student checking in before entering the classroom and the one on the right shows another student checking out while exiting the classroom.



Inspiring Excellence

Temperature: 28.20 C
Humidity: 45.00 %
Time: 22-Feb-2020 04:15:54:pm
Person Count: 6

Student Details UB50201

Serial	Name	ID	Dept	Status	Seat	Time
1	Rakin_Hassan	14321012	EEE	Checked_In	4	22-Feb-2020 11:58:30:am
2	Rakin_Hassan	14321012	EEE	Checked_Out	4	22-Feb-2020 11:58:40:am
3	Rakin_Hassan	14321012	EEE	Checked_In	4	22-Feb-2020 12:03:28:pm
4	Rakin_Hassan	14321012	EEE	Checked_Out	4	22-Feb-2020 12:03:37:pm
5	Sadia_Salauddin	15121037	EEE	Checked_In	1	22-Feb-2020 12:03:52:pm
6	Sadia_Salauddin	15121037	EEE	Checked_Out	1	22-Feb-2020 12:03:53:pm
7	Wasee_Ahmed	17321029	EEE	Checked_In	8	22-Feb-2020 12:07:58:pm
8	Wasee_Ahmed	17321029	EEE	Checked_Out	8	22-Feb-2020 12:08:09:pm
9	Rakin_Hassan	14321012	EEE	Checked_In	4	22-Feb-2020 12:08:19:pm
10	Rakin_Hassan	14321012	EEE	Checked_Out	4	22-Feb-2020 12:08:29:pm
11	Sadia_Salauddin	15121037	EEE	Checked_In	1	22-Feb-2020 12:08:38:pm

Figure 3.8: Server showing Student Information

This is the image of the server. This page is only accessible by the university administrators. This will display the name, ID, department and the allocated seat number of the students. The time and date of the students checking in and out will also be displayed. The total number of students inside the classroom, which is a part of the room occupancy system, will obviously be present. The temperature and humidity readings are also shown and a fire warning will appear if a fire breaks out. This part will be discussed in the next section.

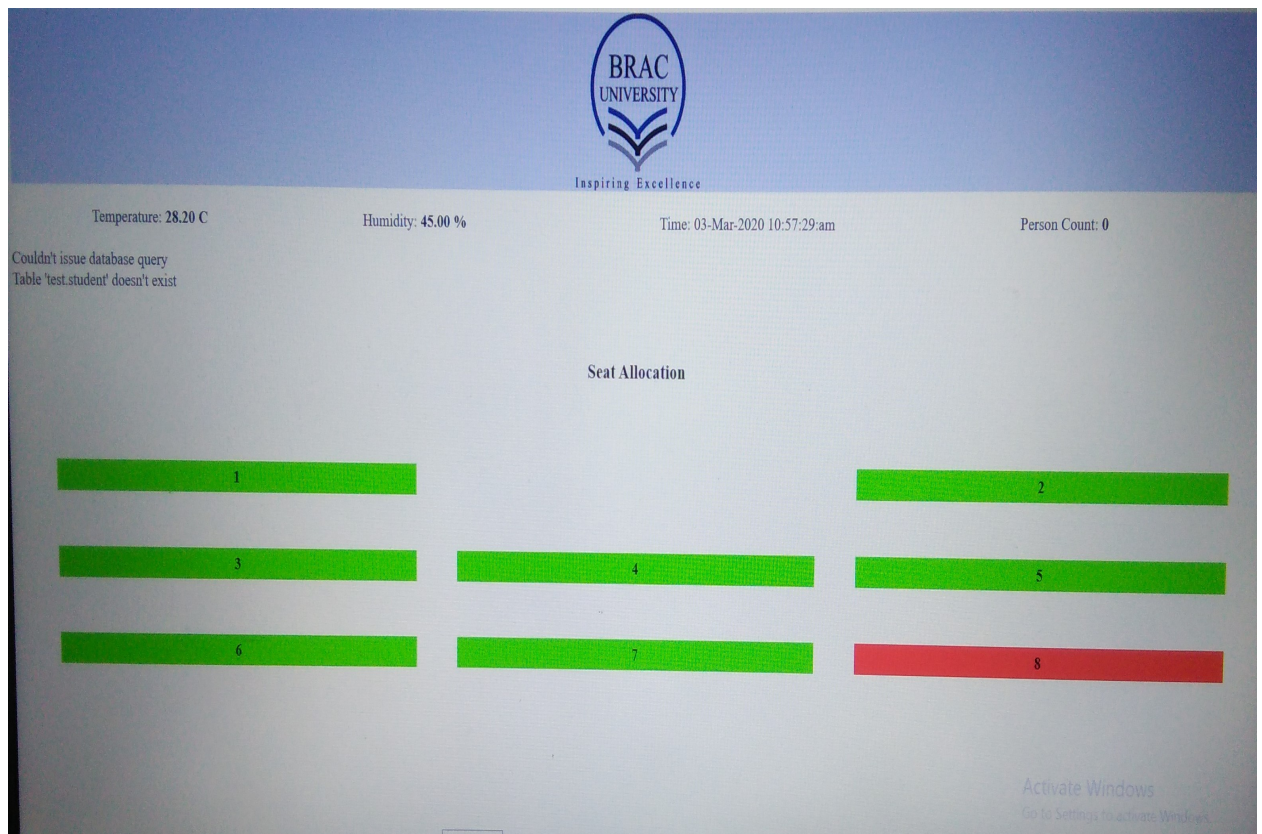


Figure 3.9: Server Showing Allocated Seat for Examinees

This is a part of the server where the allocated seat of a student after they have scanned their RFID card will be displayed in a diagrammatic way. The green boxes are the seats that are still empty and the red ones are filled by students.

3.4 Environment Monitoring

3.4.1 Equipment Used

- 1) Arduino UNO
- 2) DHT22 Temperature and Humidity Sensor
- 3) Flame Sensor
- 4) ESP8266 WiFi module
- 5) Buzzer

3.4.2 Model Description

Flame Sensor

Infrared radiation belongs to the electromagnetic spectrum. This has longer wavelength than ultra violet rays or xrays. Infrared rays are transferred from hotter regions to cooler regions without using any medium like air or water. Flame detector is a sensor, made using photodiodes that detects the presence of IR rays from flame or fire and responds accordingly. The response to the fire may vary according to the desired

output- alarm, buzzer, light etc. It can also be made to activate a fire suspension system. There are different existing methods by which flames can be detected such as- Ultraviolet detector, Infrared thermal camera, IR detector, near IR array detector, UV/IR detector etc. As we are keeping our cost in check, the method we have used for our model is IR based flame sensor. The concept to this method is that when any object burns or is on fire, the flame emits a small amount of IR light [4]. Flame sensor module comprises of photodiode, capacitor, potentiometer, resistor and an LM393 comparator. This IR released by the flame is detected by the photodiode in the IR receiver on the sensor module. The comparator constantly looks for differences in voltage across the IR receiver. The fire is detected as the flame sensor detects the flame signal that is generated by the IR source [19], the output will show low voltage (0V) and if it shows high voltage (5V) then it means no fire. The buzzer we have used for our model buzzes when the voltage is low (0V).

Temperature and Humidity Sensor

Humidity is the presence of water in our surrounding air. Humidity or the gaseous state of water is merely visible to our naked eyes. The level of humidity in air changes according to the weather or climatic changes. This level affects different biological or chemical processes. For the safety, health and possible damage concern of goods, devices or chemicals in a laboratory, it is wise for any university to observe the temperature and humidity and keep them at optimum level. In the United States, the guidelines for labs has been set by the Food and Drug Administration. The regulations describe the optimal temperature should be between 20 and 25 degrees Celsius and humidity should be between 30 to 50 percent [21]. There are several ways to ensure the safety of any laboratory and classrooms and one of them is continuously monitoring the temperature and humidity change. We have used relative humidity (RH) to monitor humidity in the classroom. RH is the humidity calculated as a percentage to the absolute humidity. The sensor that we have used in our temperature and humidity detection model is DHT22 [17]. The advantage of using this is that it is a low cost and small sized sensor and readily available in the market. This sensor has a transmission distance of about 20m and is very suitable for our temperature/humidity module. DHT22 measures the surrounding air using its capacitive humidity sensor and a thermistor and then sends out the digital signal on the data pin. We have connected 5V power supply with the first pin and the second pin to the data input and the fourth pin to the ground [8]. Third pin remains null. Similar model can be created using a DHT11 sensor, however DHT22 is more precise, more accurate and covers greater range of temperature and humidity. But DHT22 is more expensive than DHT11. 3.11 shows the image of the temperature and humidity detector. The white meshed part is the DHT22 that we have used in our model. We have also used Arduino and ESP8266 in our model. Firstly, the sensor sends out highest data bit. The data is sent out in decimal and in integrated form and single-bus is used for communication. The data collected from the surrounding is then uploaded in the university server, made by using XAMPP [2].

3.4.3 Model Architecture

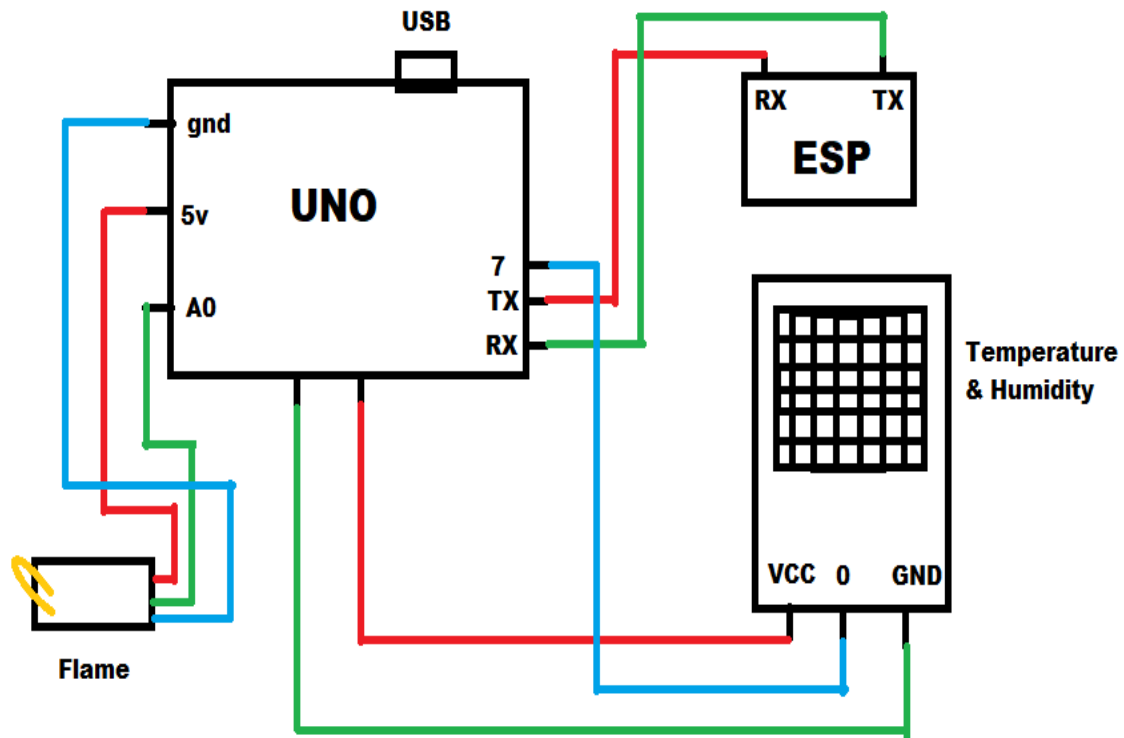


Figure 3.10: Model Architecture of Flame, Temperature and Humidity Detector

The above diagram shows the connection between flame sensor and temperature/humidity and Arduino UNO. Ground pin of flame sensor is connected to the ground pin in Arduino UNO. The digital output pin is connected to the A0 pin and the power pin is connected to the 5V power supply of the Arduino. Again, the Vcc pin in the temperature/humidity sensor is connected to 5V power supply in Arduino. The second pin is connected to the data input and the GND pin is connected with the ground. The Tx and Rx pin in the Arduino UNO is connected with the Rx and Tx of the ESP8266 respectively. The data is passed through the ESP8266 to the server and is uploaded there.

3.4.4 Images of the hardware

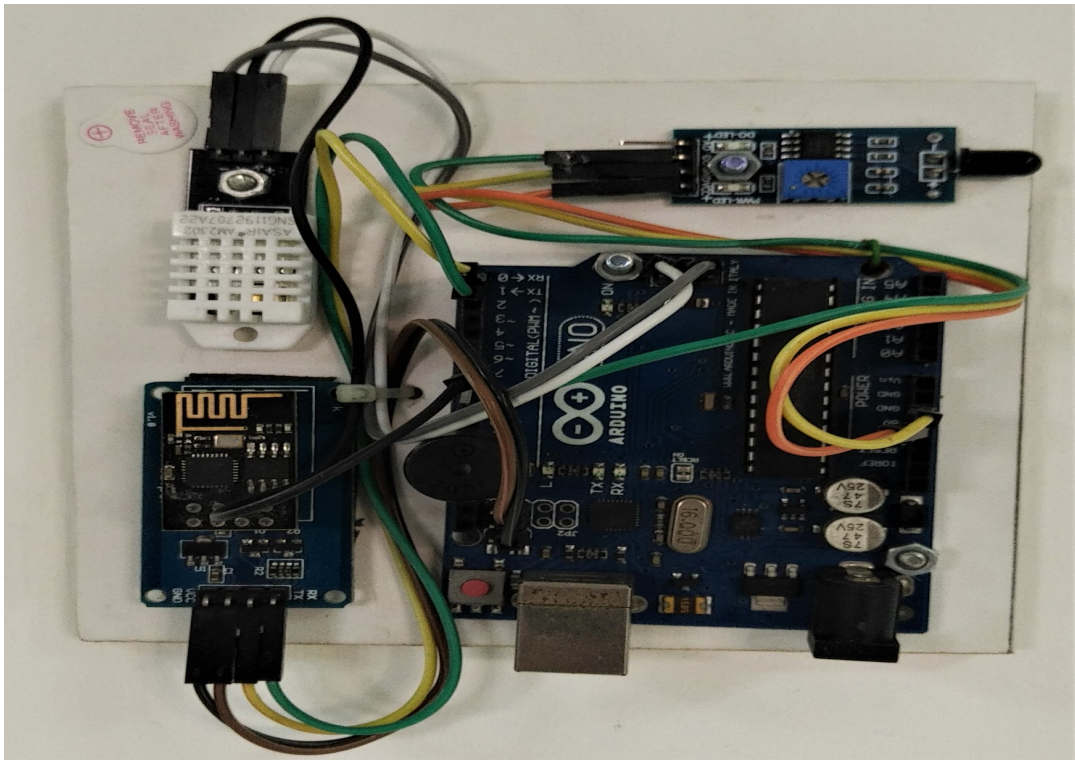


Figure 3.11: Hardware Image of Flame, Temperature and Humidity Detector

This image sensor shows the part of our work where we build a flame, temperature and humidity sensor system. The white meshed part is the DHT22 that we have used in our model. We have used Arduino and ESP8266 in our model. All these data will be directly uploaded to the server via the Wi-Fi module. There is also a buzzer which will start buzzing if a fire breaks out and a warning will also be shown on the server's page so the admins can take necessary steps.

Chapter 4

Algorithm

In order to describe the operation of the system, we proposed an algorithm for our project.

We have used NodeMCU to upload the information to the server. NodeMCU is a low cost, open source firmware. It is based on the ESP8266 Wi-Fi module. To develop our local server we have used XAMPP. XAMPP stands for cross-platform, Apache, MySQL, PHP and Perl. Using this we can build any WordPress site offline, by making a local server using our computer.

4.1 Occupancy Estimation

When any student enters the classroom to attend any class in a given time slot, he/she cuts the IR rays coming from the transmitter travelling towards the receiver. Two transmitters and two receiver both aligned, are mounted on the opposite ends of the door frame. The IR sensors are connected to the Arduino. While entering the classroom, the first beam is cut and then the second beam, which shows that students are entering. A signal is sent to the system and the count increases. While coming out of the classroom, the second beam is cut and then the first beam. This ensures that the student is coming out of the classroom and the count decreases. We have mounted the transmitter and the receiver at the bottom of the door frame, so when a person enters or leaves, their legs are cut and a reading is taken.

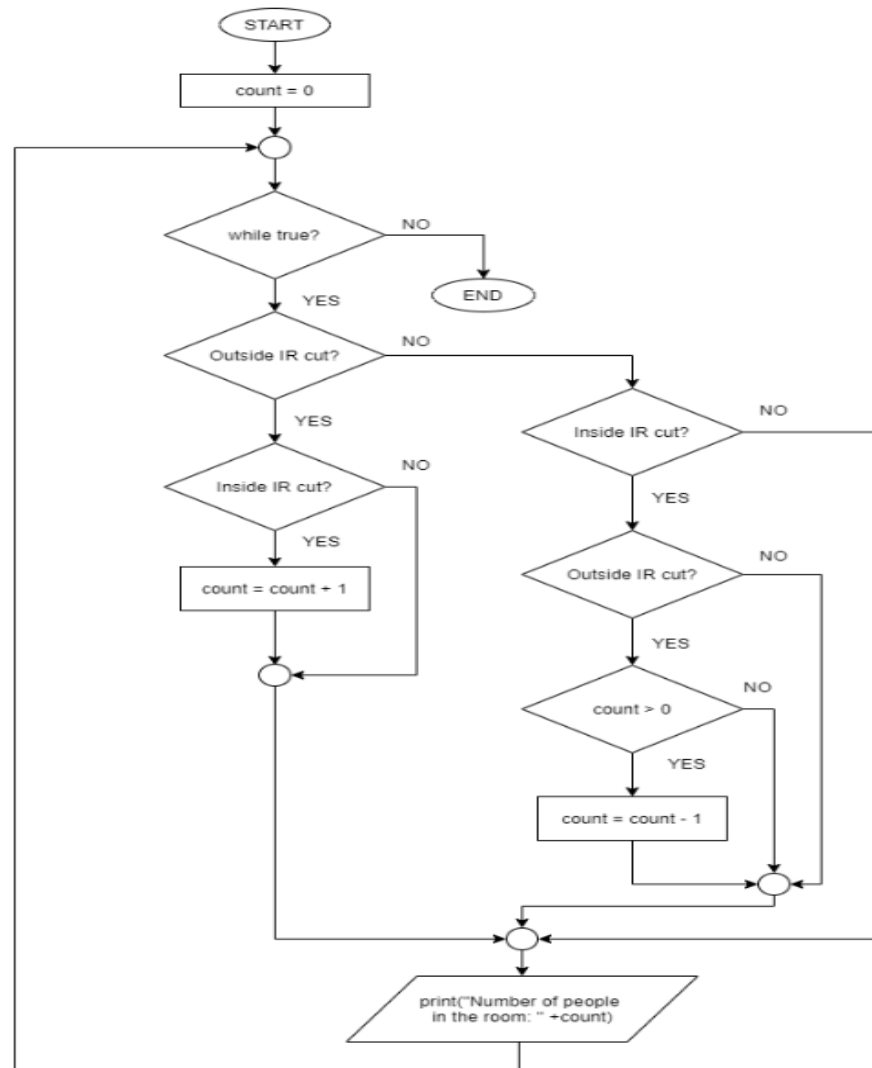


Figure 4.1: Flowchart of Occupancy Estimation

```

people_count = 0
previous_people_count = 0

While True:
    if Serial is available:
        previous_people_count = people_count
        people_count = SerialRead()

    if people_count is not equal to previous_people_count:
        previous_people_count = people_count
        sendToServer(Person=people_count)

    rfid_number = readRFIDcard()
    student_details = getStudentDataFromServer(RFID=rfid_number)

    if student_details is NULL:
        SerialWrite(-1)
    else:
        if student_details->current_status is 'OUT' and people_count > 0:
            people_count = people_count - 1
        else:
            people_count = people_count + 1

    SerialWrite(student_details->id)

```

Figure 4.2: Pseudo Code of Occupancy Estimation

4.2 Seat Allocation

When the student wants to enter the classroom to sit for any particular exam, he/she must insert his/her ID card in the RFID module attached with NodeMCU mounted outside the classroom (in our case it was placed inside the classroom). When the RFID detects the information of the student, it sends a request to the system to book a seat. The system then sends back a response and books a seat for the student in real time. The seat booked for the student is displayed in the LCD and this data is uploaded and stored in the server. The authorization to enter classroom can only be achieved via RFID technology or by scanning the individuals ID card. If the ID card is declined, the student will not be able to seat for the exam. This situation can be controlled by the faculties and the higher authorities.

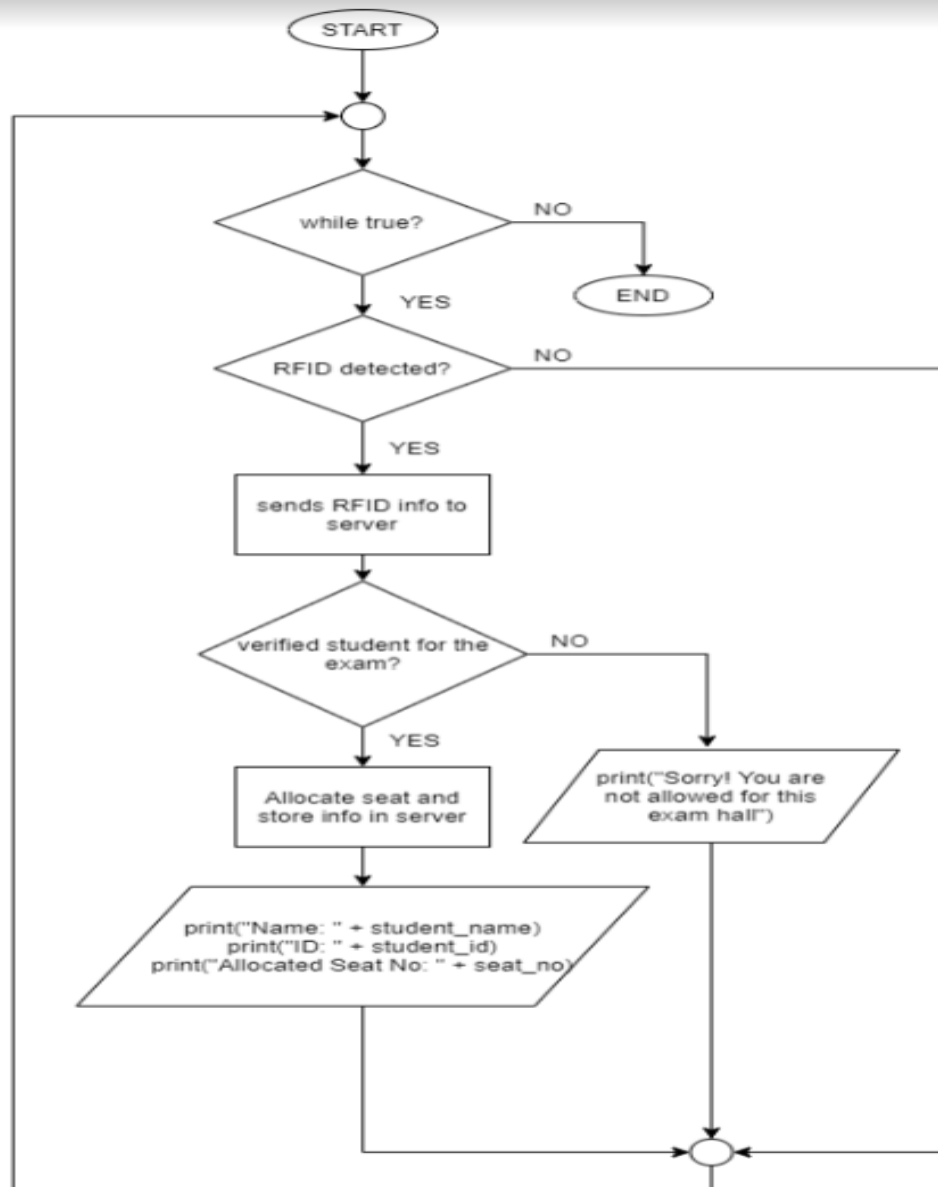


Figure 4.3: Flowchart of Seat Allocation

```

id = 0

While True:
    if Serial is available:
        id = SerialRead()

        student_details = getStudentDataFromServer(ID=id)

        previous_clock = current_clock()

    if current_clock – previous_clock < 4 seconds:

        if student_details is NULL:
            showLCD("You are not authorized")

        else:
            showLCD(student_details->name)
            showLCD(student_details->student_id + " " + student_details-> current_status)

    else:

        people_count = getDataFomServer(PERSON)
        showLCD("Students: " + people_count)

```

Figure 4.4: Pseudo Code of Seat Allocation

4.3 Environment Monitoring

```

humidity = 0
temperature = 0
fire = false

While True:
    humidity = readHumidity()
    temperature = readTemperature()

    fire = readFlameSensor()

    sendToServer(humidity, temperature, fire)

```

Figure 4.5: Pseudo Code of Environment Monitoring

Chapter 5

Readings and Results

5.1 Hardware Setup

The whole setup of our project was done in our own BRAC University. The room we chose was Energy Conversion Laboratory in building no. 5 (UB50603). The door of the lab was a Norman door where we had to push the door to get in and pull it to get out. Since our model was made for a sliding door, we had to keep the Norman door open to simulate that.



Figure 5.1: Energy Conversion Laboratory Doorway

The IR sensors were installed on the walls on either side of the door at one feet above the ground. From the perspective of anyone entering the room, the IR receivers were placed at the left side of the door and the IR transmitters were placed at the right side, which when the door opened fully would fall behind the door. We have assembled a box which was made of PVC sheet which can be seen in fig. 3.6. The setup is such

that the LCD display is at the top, the Arduino UNO below that, at the bottom is the NodeMCU and beside that is the RFID module. This is shown in the figure below.

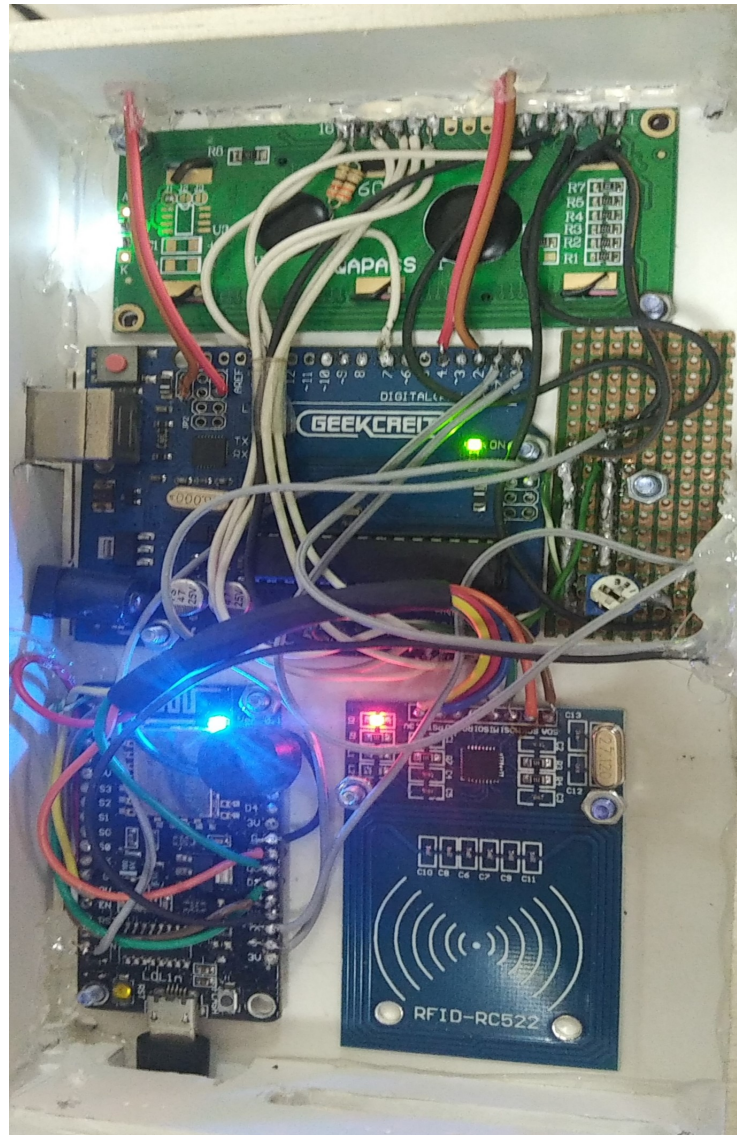


Figure 5.2: Hardware Image of Room Occupancy and Automatic Seat Allocation System

This setup was placed inside a hollow PVC box and we created another back covering with the same material. On the box we cut up some slits to make it easily accessible. The LCD display's part also had a slit so that the screen could be seen from the outside, the RFID had an opening at the bottom to slide the card in and also the Arduino UNO and NodeMCU had small slits so that USB cables could be connected. The box was held in place on the wall by the help of double sided tapes. The flame, temperature and humidity sensor unit was placed on another PVC box and taped to another wall shown in fig. 3.11. The wires were placed on the ground and on the walls with the help of double sided tapes so that someone does not trip or the connection does not become loose or displaced. The 5V power supply and the fire sensor unit was directly plugged to the wall socket. The RFID scanner and NodeMCU unit still was not getting enough power, so we had to power the Arduino UNO using a USB cable connected to our laptop. To connect to the server we used a mobile phone as a hotspot.

5.2 Project Implementation in Real Life

After we setup the hardware as mentioned in the above section, we took three readings on three different occasions. On two occasions we used our fellow department mates and friends and lined them up. Then they were told to enter the classroom one by one and the readings were noted.

On another occasion we took readings in EEE224 lab class that is held at that particular room. Permissions were taken beforehand of our respective faculties by us. The class started at 11 am on Tuesday, 03.03.20. We went to the room early in the morning to set everything up and have a few test runs by ourselves. We also had to fix a few minor issues that showed up before the class started. Around 11 am the students started coming in and before they could enter, we requested them to get inside the classroom one by one. We manually counted the number of students present on that day and crosschecked it with the reading the LCD display and the server were showing. We also checked if the RFID was working by scanning our cards few times. The flame, temperature and humidity Sensors were powered by another Arduino and were taking readings continuously and updating to the server.



Figure 5.3: Ongoing Energy Conversion Lab Class

5.3 Results

Room Occupancy

After the students got inside the classroom we were seeing that the student count showing on the LCD display and the server were exactly the same and was updating almost instantly.

However, when we crosschecked with our own manual headcount we saw that there was a huge difference between it and the digital count on all three occasions. For e.g when we took data of the EEE224 lab class, there were total 15 students present in the classroom but our sensor data showed 24 students. We then conducted our own tests and found out that the problem was that the IR sensors were counting two legs of the

same person as two different students. We tried a lot of things to solve this problem but could not come up with a solution using the same equipment. To solve the problem we realized we had to divide our total value by two, since two legs were counted instead of one. Let,

A (The actual students/headcount of the classroom) = 15

N (The leg count shown in the LCD and Server) = 24

S (Digital student count after dividing by 2) = $N/2 = 24/2 = 12$

But we can still see that this does not give 100% accuracy.

Therefore, the percentage accuracy = $(S/A)*100 = (12/15)*100 = 80\%$

Automatic Seat Allocation

We programmed four RFID cards to be used with the reader. The cards were assigned to Rakin Hassan, Sadia Salauddin, Wasee Ahmed and Mushfiquir Rahim. Since we only had four cards, the students could not be assigned their individual cards. So we were testing the RFID cards by scanning it ourselves. The RFID showed no problem at all when scanned, and displayed the students name, ID and status on the display almost instantaneously. After 4 seconds of showing the information on the display, another card could be scanned again.

The information about the automatic seat allocation on the server was also almost instantaneously updated as well. It also showed more information than the LCD display could show. In addition to the name, ID and status, the server also showed the department and the allocated seat number of the student scanning the card. The server also displays a 2D seating arrangement diagram with all the booked seats showing red and the unbooked seats showing green 3.9. For e.g when we scanned Mushfiquir Rahim's card the first time, the seat colour turned red and when we re-scanned it while checking out, the colour turned green again indicating the seat was then empty.

Environment Monitoring

The information provided by this unit was not shown on the LCD display.

The data was transferred to the server by the ESP module. The temperature and humidity was shown on the server's homepage in Celsius scale and was in sync with the official readings. We used a lighter in front of the flame sensor and the buzzer immediately started buzzing and a warning sign started displaying on the server's page. The temperature at that day was 28.2 °C and the humidity was 45%.

Chapter 6

Challenges, Solution, Application and Future Work

6.1 Challenges

Before we implemented our project and were brainstorming, we came up with multiple ideas on how to build our project up. But we were unsure of what method to use for the occupancy estimation section. First we tried with a laser module and LDR. The output was not accurate, nor was the module working in a stable manner. We then shifted to IR sensors and used reflectance IR sensors. The output was better but we expected more accurate results. We faced problems on what type of IR sensor to use later on.

Furthermore, while setting up the hardware we faced some problems as well. All our separate units were installed in PVC boards and we had to find a way to put it up on the walls. As drilling them to the wall was not allowed by the university authorities, it caused a problem. The wires were also a problem as they were sticking out in many places. This could be a safety issue as someone passing by could trip on the wires. Also, while cleaning or if someone tripped on it the wires could come loose or any other component could get damaged. Another issue we faced was aligning the IR transmitters and receivers perfectly. The two sets had to be perfectly aligned for it to work. For instance, while we went to take the readings during the lab class, we went and found out that it had been misaligned again, even though the past day we had aligned it perfectly. So, we understood that if a passerby would bump into it this problem would recur in the future. Another problem we detected was that the IR receiver would face interference as all the IR rays within its range would hit the sensors from all the sides. After the set up had been done we still found some shortcomings we had. The RFID and NodeMCU unit was not getting enough power from the power supply and sometimes the LCD was blinking or turning off. Also, a major issue we faced was the IR sensors counting both legs of a student and counting them as two students instead of one.

6.2 Solutions

After carrying out all the test runs and the data collection from the lab class we have found a few problems as mentioned above. We thought of some solutions that could

improve the overall project and make it even better in the future. To fix the first issue which was accuracy problem regarding occupancy estimation, during the final hardware implementation stage we decided on using break beam IR sensors. This made the data way more accurate and worked in a stable manner.

In order to put the units with the PVC boards on the wall, we used double sided tapes to overcome the mentioned issue. To fix the wiring issue we had to make sure the wires were kept in place with tapes and the units kept in such a way so that someone would not accidentally bump into it. To overcome the difficulty we faced aligning the IR sensors, we used green and red LEDs to indicate if the two sets are aligned. The LEDs only lit up if the alignment was perfect and when the IR ray was not being cut. To solve the interference issue we installed aluminum foil (shown in fig. 3.5) on the IR receivers so that only the IR transmitter rays being transmitted from the opposite side could fall on them.

To fix the power issue the RFID and NodeMCU unit had, we had to give an external power supply using a USB cable connected to our laptop. This fixed the issue totally. Then the major issue we had regarding the IR sensors counting both the legs, we had to divide the result we got by two, to get an optimal result, as mentioned in the previous chapter.

6.3 Application

Our proposed system, prominently, will hold information about real-time room occupancy status from a remote location. Moreover, the exam seat allocation can also be monitored in a similar manner. Exam seat allocation system will come in handy during any admission or semester final exams in universities. Furthermore, this system can be used by various organization for holding written assessments during recruitment procedures.

On the other hand, the room occupancy tells us the number of people present in a particular room, can be used in classrooms, laboratories or office spaces to monitor the vacancy status of the said rooms.

Apart from using room occupancy and seat allocation in universities and schools, this system has a number of usages in different fields. The data collected can also be used by the university or other people or organizations for R&D. One of the most important aspect of the project is that it can be used to estimate the number of people in a building (offices, institutes or residence), in case of any emergencies such as fire hazards or earthquake. Fire/Police departments always wants the data of people present at the building in times of emergencies. This information can be accessed from the local server and will help the rescue team during their rescue mission.

We can also integrate our said model with automatic lighting and temperature control system for classrooms, offices or homes to be more energy efficient. Example, the light and temperature can be automatically adjusted based on the number of people present in the room [14].

6.4 Future Work

We could make some changes and improvements in the future to make the whole project better. We plan on making an algorithm that will divide the student count by 2 and upload it to the server automatically. Also, our plans for the RFID cards were only to assign them to students during exam time. But, since the IR sensors were not working perfectly on its own, if all the students were given the cards at all the time, this would improve the accuracy of the attendance to more than the current 80%. We know when the RFID card is scanned, the name of the student, ID and their allocated seat number will be updated to the server. This will be very easy to crosscheck with the IR sensor data. The accuracy could be increased even more if more expensive equipment were used. This depends on the university's budget and allowance. Equipment such as beam sensors or HPD camera could make the data a lot more accurate.

Chapter 7

Conclusion

The world we live in now is heavily technology based and IoT plays a huge role to make life easier and advanced for all of us. We used the core essence of IoT in our university environment to make a smart classroom, which will make life simple and helpful for the faculties, staff and students. We also believe that IoT is not a luxury any longer but a necessity to develop a smart, safe and secured university campus. Automatic attendance system, allocating seats for the examinees and environment monitoring (flame, temperature and humidity detector) were the parts we focused on in our project. These data were also uploaded to the server in real time for university administrators to use. All our sections were working properly and giving good results but only the room occupancy section was 80% accurate. We gave ideas on the solutions to this and other small problems we faced and wrote down some future plans in the above chapter. We know IoT is not a brand new or an exclusive concept in the world any longer, but our focus was trying to make this concept more accessible to our part of the country at a very reasonable price so that it can be implemented nationwide in a very short span of time.

Bibliography

- [1] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks*, vol. 4, no. 2, pp. 351–367, 2004.
- [2] D. Dvorski, *Installing, configuring, and developing with Xampp*. Canada: Skills, 2007.
- [3] B. Song and C. Mitchell, "Rfid authentication protocol for low-cost tags", *Proceedings of the first ACM conference on Wireless network security - WiSec '08*, 2008. DOI: 10.1145/1352533.1352556.
- [4] M. Al-Abbas, "Heating by catalytic gas infrared rays," *Energy Engineering*, vol. 108, no. 6, pp. 26–45, 2011.
- [5] J. Höller, V. Tsiatsis, C. Mulligan, S. Karnouskos, S. Avesand, and D. Boyle, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Amsterdam, The Netherlands: Elsevier, 2014.
- [6] T. Pham, M. Tsai, D. Nguyen, C. Dow, and D. Deng, "A cloud-based smart-parking system based on internet-of-things technologies", *IEEE Access*, vol. 3, pp. 1581–1591, 2015.
- [7] P. Verma, A. Kumar, N. Rathod, P. Jain, S. Mallikarjun, R. Subramanian, and R. Sundaresan, "Towards an iot based water management system for a campus", 2015. DOI: 10.1109/JIOT.2019.2902410.
- [8] M. Bogdan, "How to use the dht22 sensor for measuring temperature and humidity with the arduino board," *ACTA Universitatis Cibiniensis*, vol. 68, no. 1, pp. 22–25, 2016.
- [9] A. Khanna and R. Anand, "Iot based smart parking system.", in *2016 International Conference on Internet of Things and Applications*, Jan. 2016, pp. 266–270.
- [10] N. Sastra and D. Wiharta, "Environmental monitoring as an iot application in building smart campus of universitas udayana", in *International Conference on Smart Green Technology in Electrical and Information Systems*, 2016. DOI: 10.1109/isc2.2015.7366152Y.
- [11] M. Islam, K. Hasan, M. Billah, and M. Uddin, "Development of smartphone-based student attendance system", in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2017.
- [12] S. Jagadheesh and S. Sowmya, "Iot based smart security surveillance system," *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 7, no. 5, pp. 481–486, 2018.
- [13] E. Yadav, E. Mittal, and H. Yadav, "Iot: Challenges and issues in indian perspective", in *3rd International Conference On Internet of Things: Smart Innovation and Usages*, 2018. DOI: 10.1109/iot-siu.2018.8519869.

- [14] L. Barik, "Iot based temperature and humidity controlling using arduino and raspberry pi", *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, 2019.
- [15] P. Gaonkar, J. Bapat, D. Das, and S. Rao, "Occupancy estimation in semi-public spaces using sensor fusion and context awareness.", in *Technology for Impactful Sustainable Development*, 2019.
- [16] T. Sutjarittham, H. Gharakheili, S. Kanhere, and V. Sivaraman, "Experiences with iot and ai in a smart campus for optimizing classroom usage", *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7595–7607, 2019. DOI: 10.1109/JIOT.2019.2902410.
- [17] 5. dht22 - digital temperature and humidity sensor. [Online]. Available: <https://cityos-air.readme.io/docs/4-dht22-digital-temperature-humidity-sensor>.
- [18] A brief history of the internet of things - dataversity. [Online]. Available: https://www.dataversity.net/brief-history-internet-things/?fbclid=IwAR0oMbAayOyvfJmIMy5Poq9wDA90Z9CrTHT_b3C1AdDy2_XNe55h8#.
- [19] Arduino flame sensor interfacing to build a fire alarm system. [Online]. Available: <https://circuitdigest.com/microcontroller-projects/arduino-flame-sensor-interfacing?fbclid=IwAR10TzxObTP5Jw1RAGybXp6-uRPqP1mc3cUMaXW3l3iomBCokbstgg3LMf>.
- [20] W. Goddard, *History of iot: What it is, how it works, where it's come from, and where it's going - itchronicles*. [Online]. Available: https://itchronicles.com/iot/history-of-iot-what-it-is-how-it-works-where-its-come-from-and-where-its-going/?fbclid=IwAR13nbPzPUtM7Ni0ULCZBg36J9IvL1AUicvRjbgp_IWsvDO-RL6SkqNITVY.
- [21] How to maintain optimal laboratory temperature and humidity. [Online]. Available: <https://www.sensoscientific.com/blog-maintain-laboratory-temperature-humidity/>.
- [22] Infrared sensor - how it works, types, applications, advantage disadvantage. [Online]. Available: <https://electricalfundablog.com/infrared-sensor/>.
- [23] Introduction to nodemcu. [Online]. Available: https://www.electronicwings.com/nodemcu/introduction-to-nodemcu?fbclid=IwAR2_Ux2b44ehp5QAeO_3pNCNu27dN5NJ5nzVZKwyLNbvomVQy2siURUNsUw.
- [24] Ir (infrared) obstacle detection sensor circuit. [Online]. Available: <https://www.electronicshub.org/ir-sensor/>.
- [25] Smarter future in retail with iot | hansaworld blog. [Online]. Available: <https://www.hansaworld.com/en/blog/1342>.
- [26] What are the different types of rfid technology? [Online]. Available: <https://lowrolutions.com/blog/what-are-the-different-types-of-rfid-technology/>.
- [27] What is rfid (radio frequency identification)? [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification?fbclid=IwAR2PvmKJmoJCdEiIXh4ajA>.
- [28] I. Wigmore, A. Gillis, L. Rosencrance, and S. Shea, *Internet of things (iot)*. [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>.