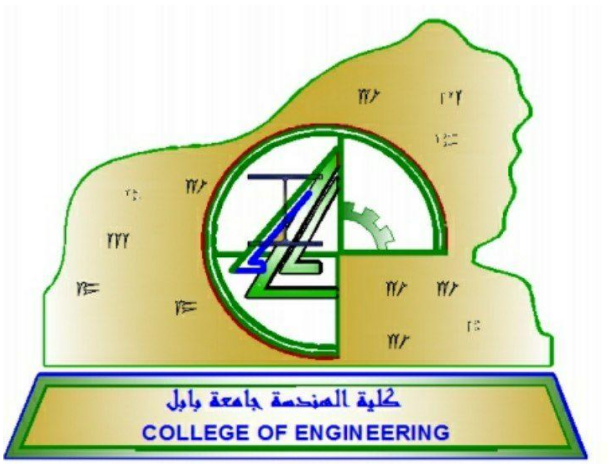
University of Babylon
Faculty of Engineering
Electrical Engineering Department



Design and Implementation of Home Automation To Reduce Power Consumption

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science (B.Sc.) in Electrical Engineering

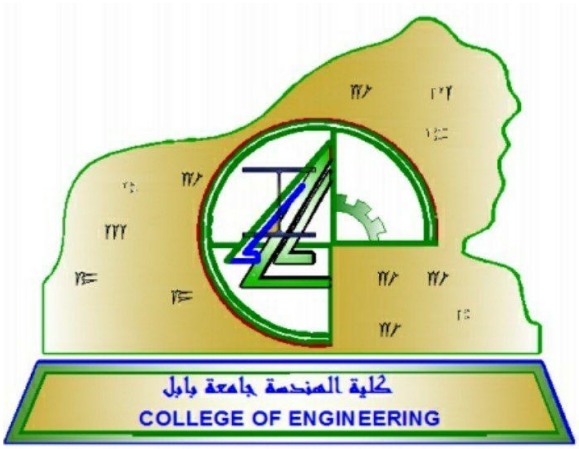
By

Huda Saad Kamel
Ali Hussien Abdel Red
Ahmed Abdel Hassan
Mohamed Ali Hamoudi

Dr. Laith Ali Abdul-Rahaim

Academic Year 2021-2022

جامعة بابل / كلية الهندسة
قسم هندسة الكهرباء



تصميم وتنفيذ التشغيل الآلي للمنزل لتقليل استهلاك الطاقة

قدم هذا المشروع لإستكمال جزء من متطلبات الحصول على درجة البكالوريوس في الهندسة الكهربائية

قدم من قبل

هدى سعد كامل
علي حسين عبد الرضا
أحمد عبد الحسن يوسف
محمد علي حمودي

أشراف

الدكتور / ليث علي عبد الرحيم

العام الدراسي ٢٠٢١-٢٠٢٢

{بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ}

(قَالَ الَّذِي عِنْدَهُ عِلْمٌ مِنَ الْكِتَابِ أَنَا آتِيكَ بِهِ قَبْلَ أَنْ يَرْتَدَّ إِلَيْكَ طَرْفُكَ فَلَمَّا رَآهُ مُسْتَقِرًّا عِنْدَهُ قَالَ هَذَا مِنْ فَضْلِ رَبِّي لِيَبْلُوَنِي أَأَشْكُرُ أَمْ أَكْفُرُ وَمَنْ شَكَرَ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ وَمَنْ كَفَرَ فَإِنَّ رَبِّي غَنِيٌّ كَرِيمٌ) [النمل: ٤٠].

"صدق الله العليّ العظيم"

إهداء

إلى من أنزل الله عليه القرآن ليهدينا به وقال له أقرأ سيدنا ونبينا محمد ((صلى الله عليه وآله وسلم)) .

إلى من علمني الإجتهاد وتحمل المسؤوليه الذي كافح من أجلنا وأوصلني لما أنا عليه حتى سرت على خطاه ((أبي المحترم)).

إلى السيدة الفاضلة التي ساندتني في كل الظروف والأوقات ((أمي الغالية)).

إلى أخوتي وأخواتي وإلى جميع أصدقائي الذين مشوا معي في هذا المشوار.

وأخيرا كل الفضل إلى اساتذتي الموقرين الذين مهدوا لنا الطريق لكم منا ماحصدنا من زرعكم .

((كل الشكر والتقدير))

Certificate

The project entitled:

Design and Implementation of Home Automation To Reduce Power Consumption

Which is being submitted by

Huda Saad Kamel
Ahmed Abdel Hassan
Ali Hussien Abdel Reda
Mohamed Ali Hamoudi

In the fulfillment of requirement for the award of the B.Sc. degree in Electrical Engineering. This has been carried out under my supervision and accepted for presentation & examination

Signature :
Supervisor's name :Dr Laith Ali Abdul-Rahaim
Date: / /.2022

CERTIFICATE

This project entitled

Design and Implementation of Home Automation To Reduce Power Consumption

Which is being submitted by

Huda Saad Kamel
Ahmed Abdel Hassan
Ali Hussien Abdel Reda
Mohamed Ali Hamoudi

he partial fulfillment of requirement for the award of the B.Sc. degree in Electrical Engineering has been discussed by us and all the suggested recommendations during the discussion are carried out

1st Examiner (The supervisor):
Signature:

Name :
Date: / / 2022

2nd Examiner
Signature :

Name :
Date: / / 2022

3rd Examiner
Signature:

Name :
Date: / / 2022

4th Examiner
signature :

Name :
Date : / / 2022

5th Examiner
Signature:

Name :
Date: / / 2022

6th Examiner
signature:

Name :
Date : / / 2022

Design and Implementation of Home Automation To Reduce Power Consumption

Abstract

As technology advances, our standards to enjoy a decent standard of life are increasingly strong. One of the ways in which affects comfort is in relation to temperature and humidity, whether inside or outside buildings. This research focuses on the study of monitoring and controlling the power consumption at our homes by controlling temperature and humidity through designing and implementing an automation system. This system can be completely operated and secure using Wi-Fi networks as communication protocol, and Node MCU as a wireless hub. The proposed system consists of temperature sensor, relay, and ESP8226 Wi-Fi that programmed the sensors to maintain the sensor metrics and transfer them to the programmed with simple interface. The system has been implemented with a high latency so that all readings are automatically updated and displayed. The designed proposed system has efficient, secure, rapid responsive real time, and control it automatically and manually at the same time.

1.Introduction

One of the most popular technologies that have become an integral part of the computing world nowadays is the principle of cloud computing. Cloud computing is a utility technology while using the Internet to access applications. Cloud computing offers various services and resources such as storage, devices, networks, databases and software applications to customers as a demand service [1]. With the spread of the concept of computing, the approach to managing and controlling the building environment has become a basic need especially in workplaces in institutions and company buildings for two main reasons; namely reducing energy consumption to the largest possible extent, in addition to creating suitable working conditions in terms of thermal comfort and indoor air quality [2]. In general, a monitoring system refers to an electronic machine that can constantly track and report one or more physical parameters including but not limited to the temperature and relative humidity [3]. Maintaining surveillance in any critical area is vital to fulfilling protection and regulatory enforcement needs. All over, there are a broad array of systems in which the control of temperature and humidity levels are crucial components [4]. Deterioration monitoring would provide early warning of initial problems enabling maintenance programs to be planned and scheduled, thereby reducing related costs. Furthermore, data from tracking systems (which would be introduced in part by enhanced efficiency prediction models) brings potential savings in life cycle costs. Thermal comfort is the condition of what a person thinks and experiences while they are in a certain setting such as temperature, and humidity [5]. It can be judged subjectively. If the atmosphere is not healthy, it can impact the work performance of the human being and trigger health hazards. Temperature and humidity should be selected primarily depending on the purpose of the room. Comfortable values for people are assumed to be 20-23˚C and humidity 40-60%. These values are also safe for the building and objects inside it, such as furniture, books and fabrics [6]. A control device will ensure the server temperature range is retained at (20–23) degrees Celsius. However, it could be more efficient to have human capital to track the temperature and humidity of the data center at night, rather than during the day [7].

Periodic surveillance by IT staff which involves physical access through a computer network, which may raise the likelihood of getting static build up, burglary, vandalism of infrastructure, and unauthorized access.
The goal of this project is to build and create a real-time temperature and humidity monitoring device that can be implemented quickly using the usage IoT Cloud platform. In addition, this study is interested in mainly Interest the thermal comfort of people in company building environment. The proposed system includes how to monitor and control climatic inside the building by a special web page for the company's building in which the principle of security is observed and only authorized persons are allowed to control it. Moreover, data is stored and transferred to the cloud server database programmed to the web server via Appatshy and Mysql formats which are updated every 5 seconds to give accurate and immediate decisions for appropriate action.

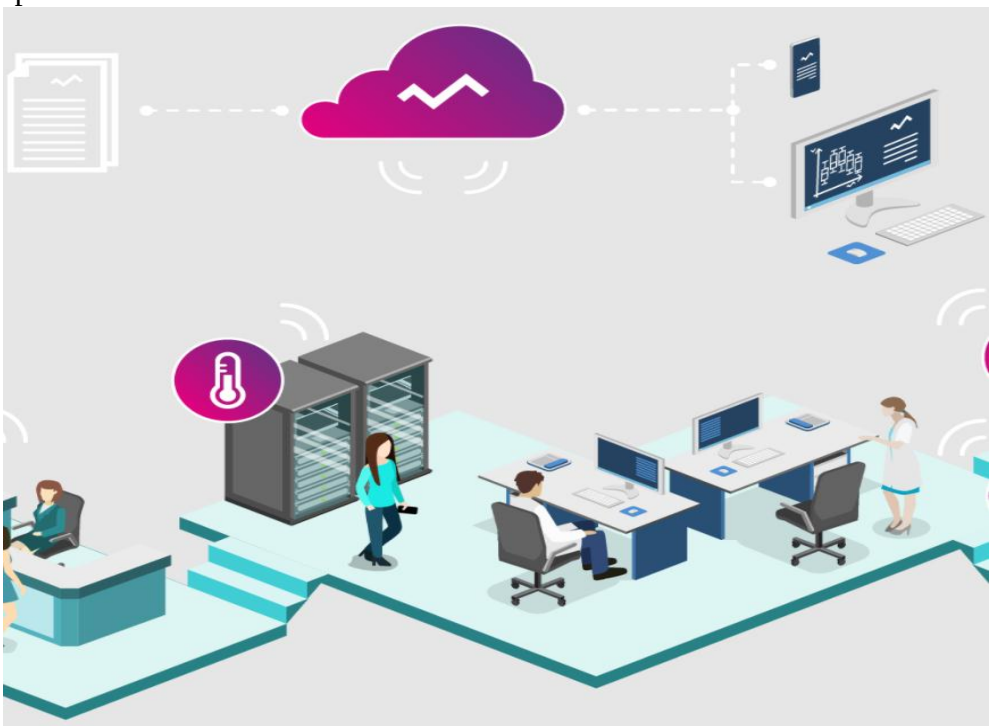


Figure 1. Concept of temperature and humidity automation for company building.

2. Literature Review

The researchers established other methods in temperature and humidity automation topics. A brief overview of some significant contribution's literature is provided in this section.
In [2], the authors study the construction of a model weather station, including temperature and humidity, to measure different weather data. This type of weather station is designed to conduct uninhabited weather data measurements. The calculated data is sent wirelessly to the remote station for the information to be registered and displayed on different gadgets. Wi-Fi links that create an interlocking network for secure data connections are used to plan this wireless link. In addition, the results encourage residents to take the appropriate precautions.
In this study [8], an automated device for regulating room temperature was suggested using a pic- sized microcontroller. This is a device that is suitable for older persons. The gui in this device calculated ambient temperature using an LM35 temperature sensor and showed the temperature on an LCD terminal. The temperature sensor

captures the daily temperature. Dependent on the temperature adjustments, the fan can automatically turn on or off. The fan's temperature was higher than the reference temperature and lower than the reference temperature at a specified interval. The reference value is entered by keying in numbers. Relay works to complete the switching role. Software proteus 8 is used to produce the simulation result.

The proposed and validated IoT-WSMP device for controlling temperature, relative humidity and light has been proposed in [9]. This system has used wireless sensing and tracking capability to track temperature, humidity, and illumination in buildings. In the built system, data is transmitted from the transmitter node to the receiver node by using a custom hopping mechanism. An Android application built which runs on a smartphone and transfers data from the LabVIEW program to the application.

The authors in [10] introduced a novel platform that enables functionality to interoperate between sensors and actuators. They aimed to minimize energy use and find a new approach to aggregate data that would appeal to the desires of the people utilizing it. It is hoped to leverage all the data accessible utilizing the proposed architecture. Deep models are used by customers who aim to change their regular habits in an energy efficient manner.

In [11], the smart air conditioner controller which implements the technology of the Internet of Things (loT) is proposed in this paper. This research develop application that can control temperature and humidity-based power consumption in air conditioners. In this analysis, an additional system to track the use of electricity called Termo has been created. Termo consists of an ESP8266 microcontroller, a temperature sensor, an infrared LED transmitter, and a humidity sensor.

In this paper [12], automation is accomplished by the use of a microcontroller that facilitates the regulation of auto room temperature and toggle switching. Depending on the changes in the environment's temperature, the electric fan dynamically adjusts the pace. A combination of sensor, controller, driver and motor with the integration of embedded driven programming includes this electrical hardware system. The machine takes the data from the temperature sensor, transfers it to the microcontroller and controls the performance of the AC heater and shows the output status on the LCD monitor.

An IoT-based smart home management system has been suggested in [13] that uses sensors, actuators, smartphones, web services and micro-controllers. In systems with devices that are turned on and off, the architecture comes into play when the devices are turned off. Being an open-source platform, an individual/group has more control over the security features. This design proposes an architecture that supports the Internet and open-source systems in the IoT. The system's implementation shows that it can be adapted to meet evolving consumer needs. The findings show that a decentralized version is cost-effective, open source, efficient, reliable and scalable.

In this study [14], authors proposed an IoT based server room monitoring system to provide information while controlling temperature and humidity within the server room. In order to track data from everywhere, the proposed device continuously transfers data to the cloud. The device is fitted with features for direct monitoring and control to dynamically provide users with alerts via the telegram application. The conclusion of this paper is that the generated framework will retrieve the server room's temperature and humidity data and the website will display the temperature and humidity data that has been taken. If the temperature and humidity in the server room exceed the cap, the device will send telegram alerts and set the temperature of the air conditioner.

This paper [15] describes how to design and implement a device using wireless sensor

networks to obtain environmental data. The framework is designed for web-based and mobile-based applications where data is fed into a web-based application and then processed at the cloud. Using the Internet and mobile apps, consumers can view their own data, as well as data from other users. In these applications, temperature and humidity sensors are used. To reduce reliance on human intervention, the web application sends a warning if a measured property exceeds a specified range. The results of the QKMF and Telcom applications indicate that this device is accurate and can be used to monitor air temperature and humidity. The data can be used to send warning emails and to analyze how the environment is improving.

In [16], an interoperable Internet of Things (IoT) platform was created using Web of object (WoO) and cloud for smart home system. The proposed architecture offers inter-operability between existing devices and communication protocols, which allows users to access their devices from anywhere. The device assigns each sensor data and actuates a specific URI that is managed by the device. Based on temperature and humidity sensors, the gateway offers interoperability with other devices such as lights, fan controls, and tank monitoring. The gateway also incorporates information from sensors and actuators in the cloud, and stores it for later use.

In this article [17], the cloud management network, which consists of the data center air conditioning system was suggested. This article explains the roles and real circumstances of each part of the system in detail. Environmental monitoring, air conditioning, connectivity, ventilation, and temperature control are part of the data center air conditioning system, while the cloud infrastructure offers data storage, big data analysis and prediction, and up-layer application. The experimental outcome shows that the proposed plan will substantially reduce energy consumption in the data center without reducing the refrigeration effect of the data center.

3. Component Description of the Proposed System

3.1 Hardware components

1-Arduino UNO

For each room you want to automate, take separate Arduino UNO.

2-Raspberry Pi 2 Model B.

3-Jumper wires (generic).

4-Breadboard (generic).

5-LDR (Light Dependent Resistor)

LDR provides different voltage reading for different light intensity.

6-PIR Motion Sensor (generic).

Passive IR detects human presence.

7-DHT22

output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity. sensing technology, assuring its reliability and stability.Its sensing elements is connected with 8-bit single-chip

Computer

8-Relay (generic).

You can use any kind of relay module. No of relay will depend upon how many devices you want to control. [No Of Devices = No Of Relay].

9-Siemens Contactor (3TF32 00-0AP0).

Contactor is going to be used to control water-pump of maximum rating of AC3: 16A 415VAC. Before purchasing contactor, please check ratings of your water-pump and your contry dependencies if there. "3TF32 00-AP0" is capable to drive up-to 16A and 415VAC water-pump. This contactor operates usinf in-built coil. Coil ratings are (specific to 3TF32 00-0AP0 only) 250VAC.

10-Miniature Circuit Breaker (1P C16A).

MCB is attached before contactor to protect water-pump from electrical faults and overload. Here MCB also have specific ratings. Our contactor is of 16A and thus I have used 16A MCB. You can modify according to your convenience.

11-Temp and Humidity Sensor (BMS Field Device).

Setra's new Humidity product are designed to continuously monitor the relative humidity and temperature in critical environments.

12-LCD screen.

A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers.

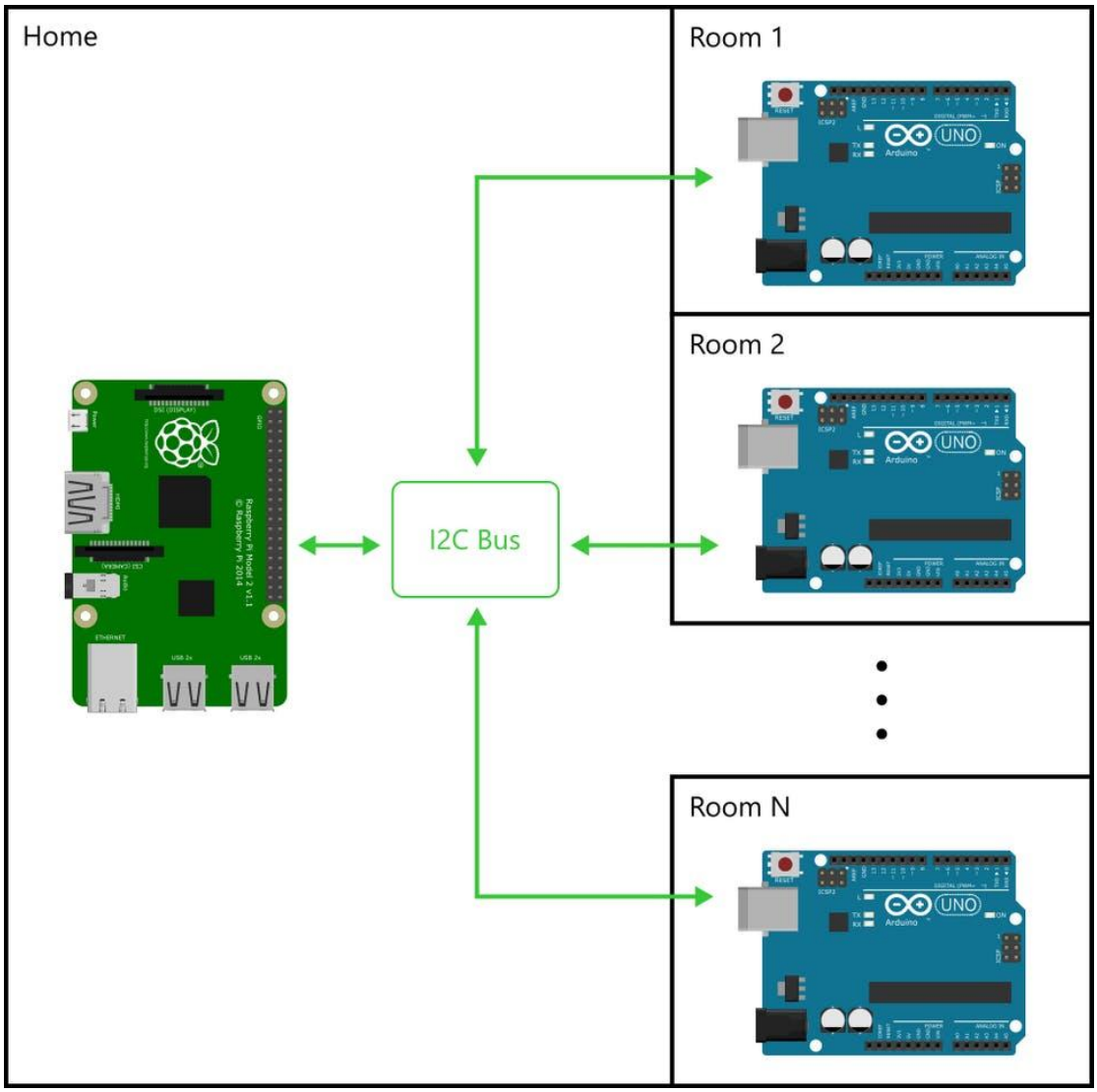
13-Heater.

14- Photo Sensor.

is a device that detects a difference in the light level received from the light source. The sensor is made up of a light source, an amplifier, signal converter, and an output.

In today's era, technology can enhance human life. Technology is evolving decade by decade. Automation was a science fiction earlier but not today. By combining latest technology with home, we can build an awesome home. With the Raspberry Pi and Windows 10, we can build a home automation system that is capable of operating home devices automatically.

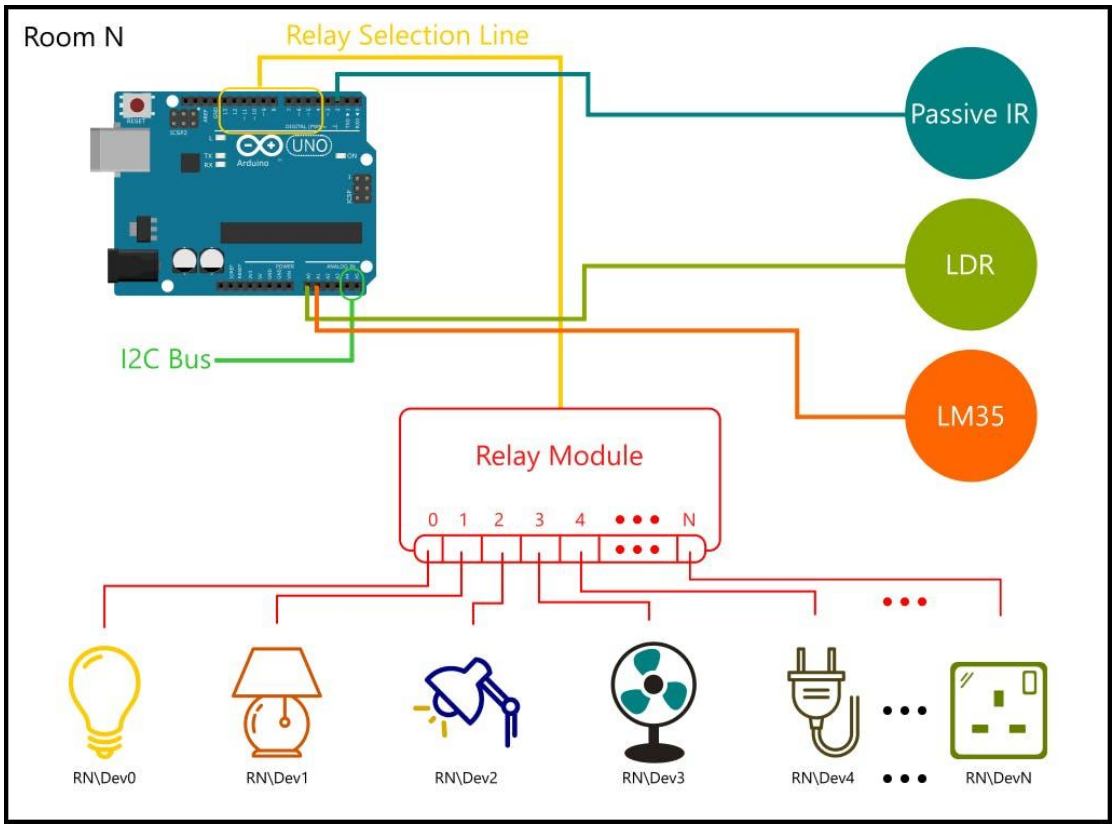
Before starting the project, lets understand basics first. Consider the following image (Overall Configuration):



3.2 Configuration for Room: -

Now considering room scenario, an Arduino UNO will control devices and reads sensor data. Periodically, Raspberry Pi requests for the sensor data collected by Arduino UNO. The figure "Room Architecture" depicts how the Arduino UNO will connects with the devices and sensors. Each room have multiple controllable devices (i.e. Light(s), Fan, Wall Socket(s), etc.), one Passive IR (to detect human presence in

the room), one temperature sensor (LM35 to collect room temperature) and LDR (to detect light intensity near room window).

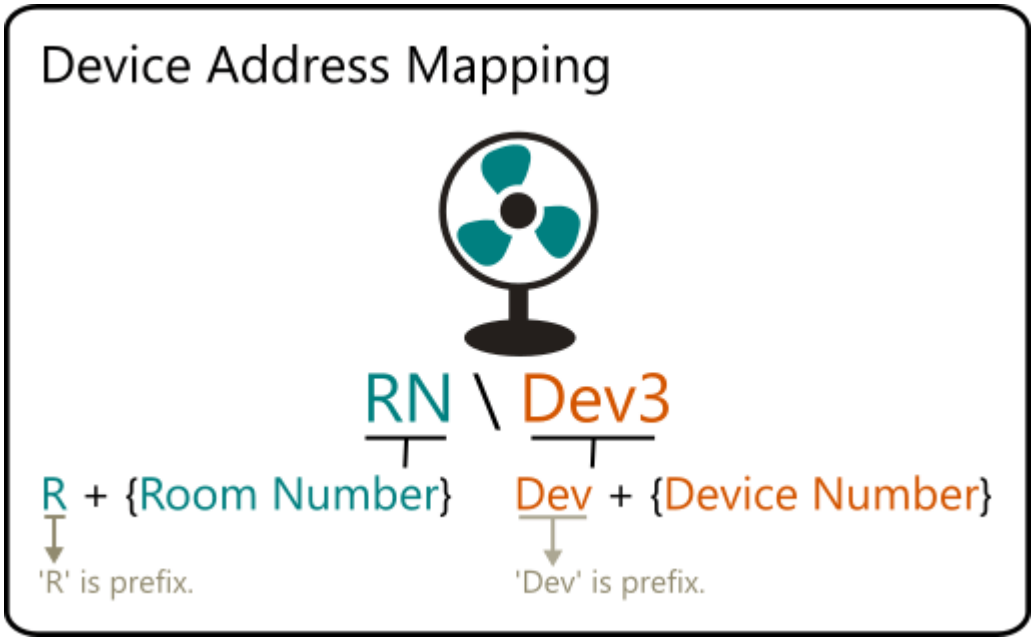


Device Address Mapping :-

The most important part is how we will identify devices? Its simple. We will make device address by combining room number with device number.

For example:

Room Number 1 having three devices. Device address will be:R1\Dev0 R1\Dev1 R1\Dev2



So up till now we know the overall configuration. In short, a Raspberry Pi will act as primary controller. Each room have its own one Arduino UNO which act as slave of Raspberry Pi. Communication between Raspberry Pi and Arduino UNO will be done using I2C.

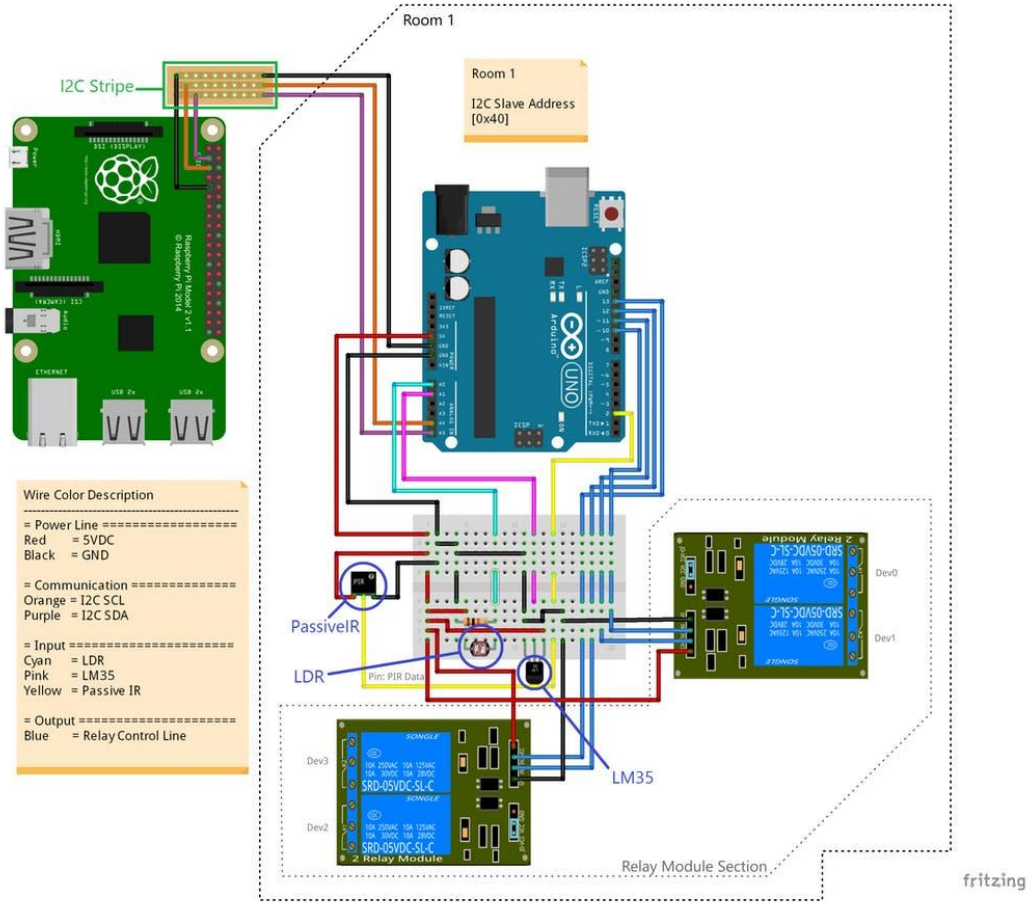
Schematics: -

To make it easier, let s start implementing for first room. Refer schematic to connect components in proper way.

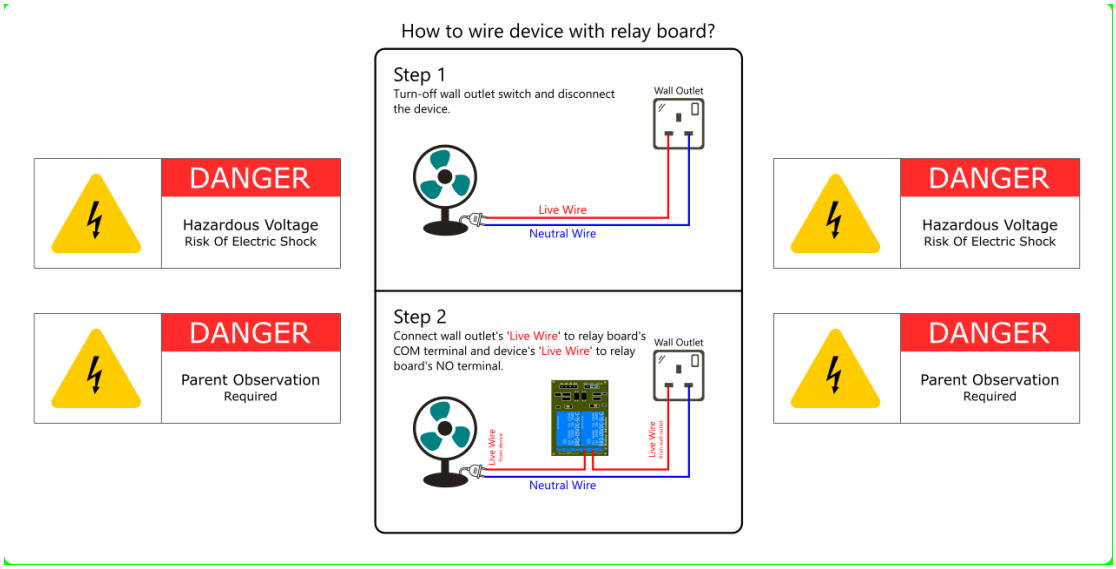
Note: I2C address for Room 1's Arduino is 0x40. You can change address but make sure to change it back in Aruino's sketch and room's slave address in the app provided below, too. Note that each Arduino resides on the bus must have unique slave address.

I2C Bus Stripe: To attach more than one Arduino you can use I2C Bus Stripe as show below.

The relay shown in the schematic are operated by 5VDC. You may have different rating of coil to drive relay. In such situation you need logic level converter circuit.



<<How to wire device with relay board?



All done at hardware side. Now Its time to build software.

3.2 Software

Our project consists of a Raspberry Pi 2 and Arduino UNO. Raspberry Pi 2 software developed in Visual Studio 2015. I am assuming that reader is aware with project creation for Raspberry Pi 2 in Visual Studio 2015. I am also assuming that user have intermediate knowledge of Arduino framework, Visual C# and Windows Universal XAML.

Let's understand software in following formation:

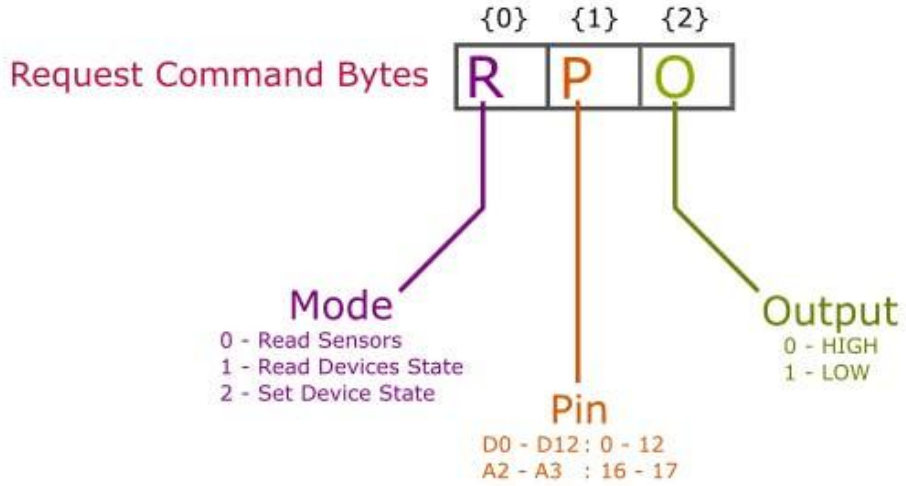
- Protocol (How Raspberry Pi 2 and Arduino talks).
- Class Structure (How Raspberry Pi 2 maintains such complicate devices and rooms).
- User Interface (Wire-frame).

Protocol (How Raspberry Pi 2 and Arduino Talks over I2C).
Before going further, lets first decide how Raspberry Pi and Arduino talks. To make a reliable protocol, we must first have clear objectives or goals for the communication. In consideration of this project, goals are:

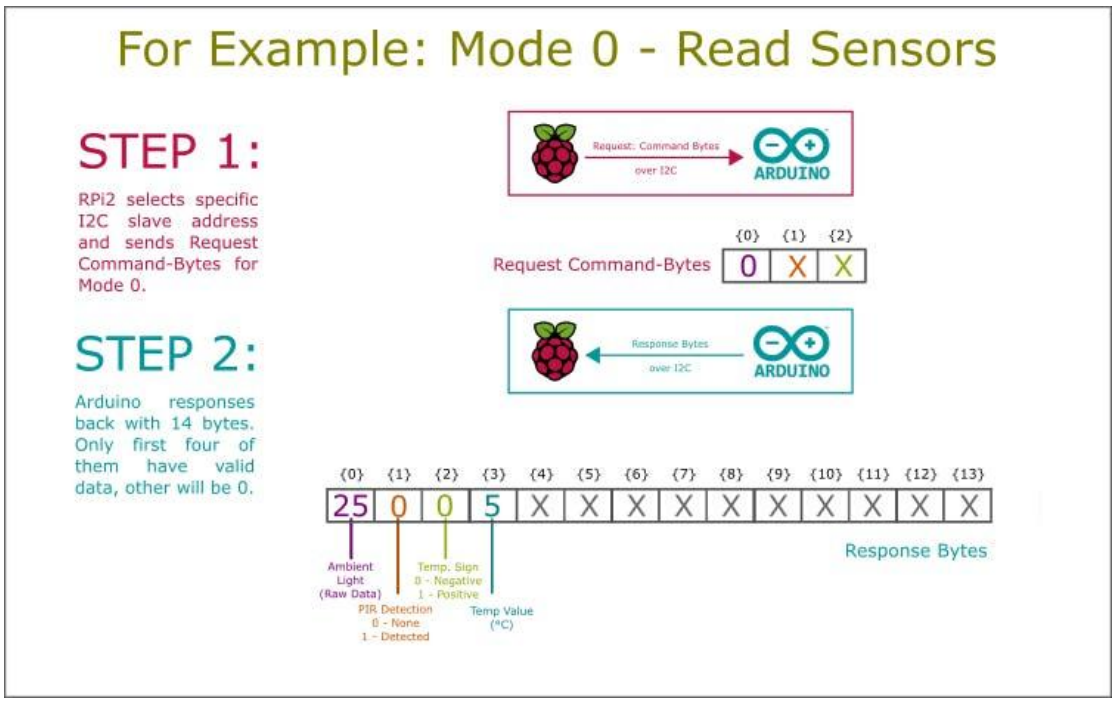
- Read sensors.
- Read device's state.
- Set device state.

Let s start defining protocol first. Protocol define rules to communicate over the bus. Protocol is nothing more than byte sequence.

I have defined protocol for sending and receiving bytes. Sending bytes are fixed of three while receiving byte array is of fourteen bytes.
Refer following schematics to understand protocol defined for this project ('X' denotes random value or '0', it will be ignored while communicating):



1/2. Protocol: Control Bytes (From RPi2 to Arduino)



Class named 'I2C_Helper' contains implementation of protocol in PRi2's Universal app project. This class is available in Universal Windows Project. To open it: Goto Solution Explorer > Library > Communication folder. On Arduino side, it is easy to understand I2C library.

Now, we have defined a proper communication scheme that is able to give status and values of sensors, devices and also provides accessibility to set device state.

Class Structure (How Raspberry Pi 2 Maintain Objects):

As discussed first, this project considered whole home. Home consists of multiple rooms and room consists of multiple devices. Thus, the OOP structure of such configuration is shown below:

Primary class "Home" consists of multiple room objects as generic list of room (List<Room>). Home class provides static method to load and save home object on the Pi to use them later.

Room consists of devices as generic list of devices (List<Device>) and sensor structure to encapsulate all sensors into one.

Device consists of details of them and functions to turn on and off them using protocol Mode - 2. Ambient Light, Passive IR and Temperature class provides access to the room's environmental data. Room object maintain sensor data and periodically refresh them using protocol Mode - 0.
Note: Class diagram is provided in project. To view it, goto 'Solution Explorer' and double click on 'ClassDiagram.cd'.

3.3 User Interface

Any application must be user-friendly. To make user-friendly application, start with wire-frame. Assume that you are going to use your own application and find out the main objectives and how to integrate them in such way that they will be most easy to end-user.

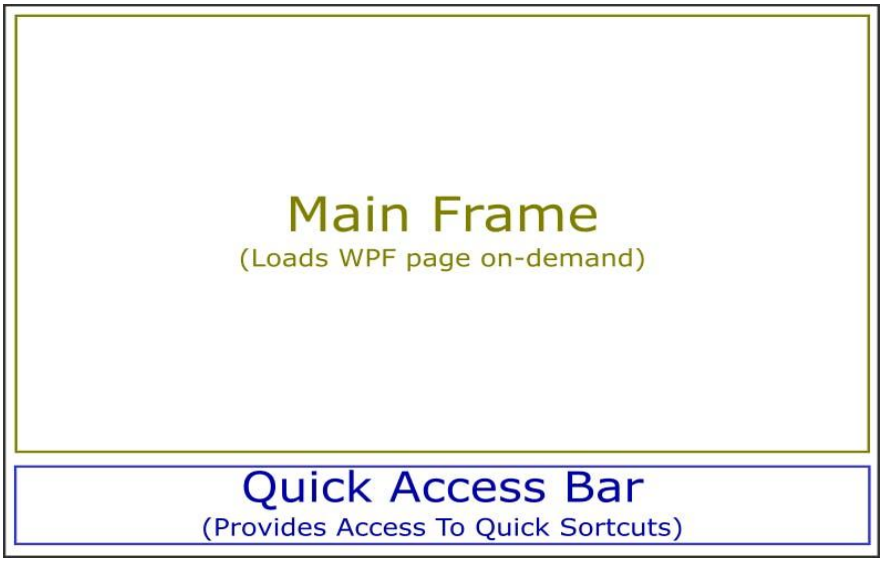
In this application's scenario, our main objectives are:

- Room Management.
- Device Management.
- External Services like GSM Communication, Internet Communication, etc.

To do so, We have splatted problem into:

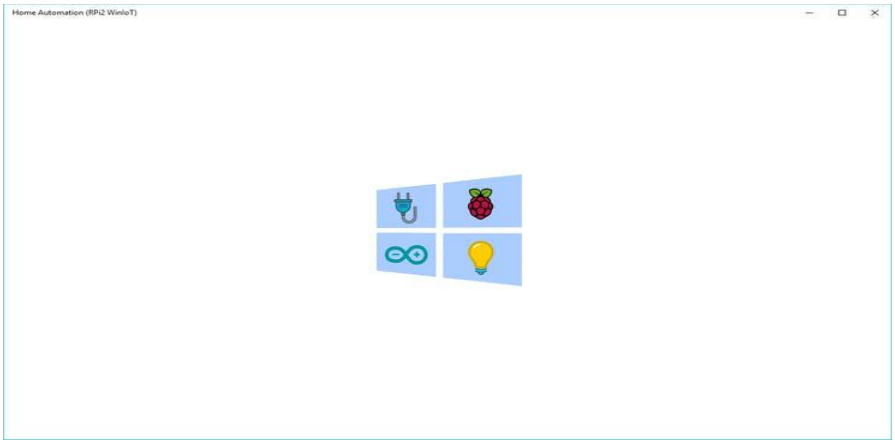
- Home Page: Provides basic device status information, date-time and lock.
- Favorite Devices: Direct access to favorite devices.
- Room Page: Provides access to configured rooms and their devices.
- Common Water Devices: Water-pump and Geyser.
- Settings/Configuration: Provides management of rooms and their devices.

Wire-frame:



Based on the wire-frame, I have developed UI which is shown next. You can

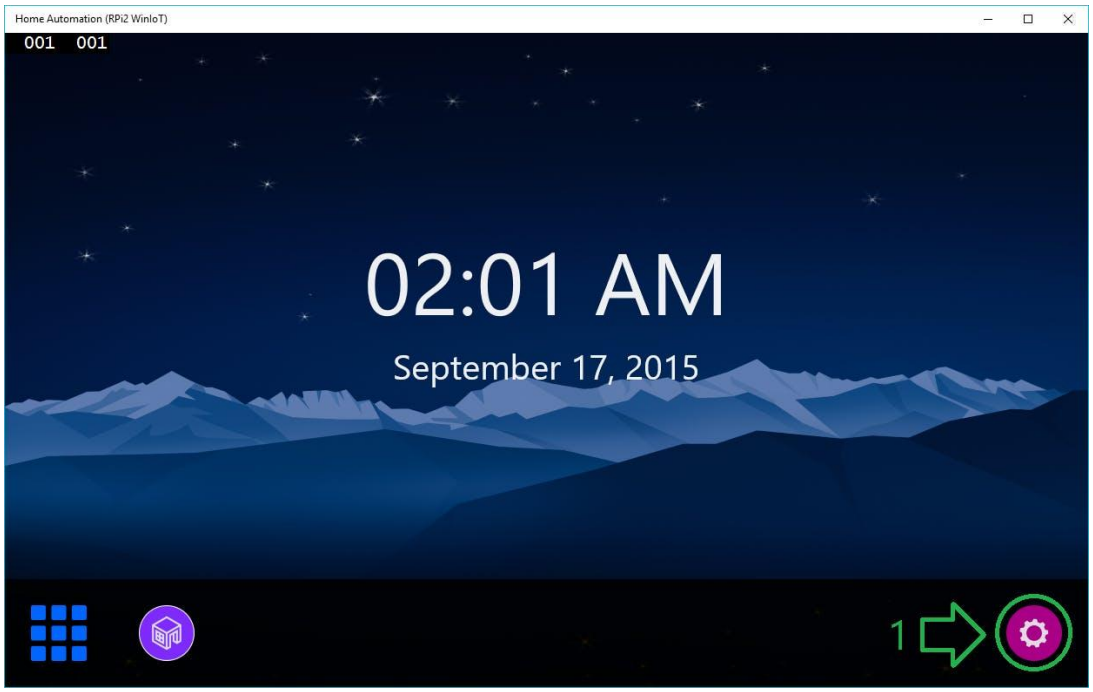
download complete source code. Link to the download is provided at the end of the article.
I am assuming that you are familiar with XAML (people having WPF background will have advantage). XAML Source code of the project is well commented so I am not going to explain it here as its not feasible. If you are newer to WPF, refer this link for getting started with WPF and MSDN link as a reference.



So, have already developed UI. Hope you can modify it up to your needs. Feel free to comment for the help.

Configure proposed system

We have tried to make this software as easy as possible. With minor configuration, you can operate devices directly using this Raspberry Pi 2. Step-by-step configuration is shown below:



control water-pump and geyser using proposd app

Anyone can attach water-pump and geyser same way you add device. But you can't attach water-pump or geyser directly to the relay board to power them up. It will blow-up relay board as well as hazardous.

To operate water-pump or geyser, connect water-pump or geyser with contactor and connect contactor's coil with relay board. So now, when you operate device, relay activates contactor's coil and thus pump or geyser starts/stop. Before purchasing contactor, please make sure its coil rating and contactor's rating. Contactor are available in so many varieties. So, verify it with your water-pump's rating. It is advisable to purchase slightly higher rated contactor than your water-pump's rating.

Known Issues

((Arduino Boot up))

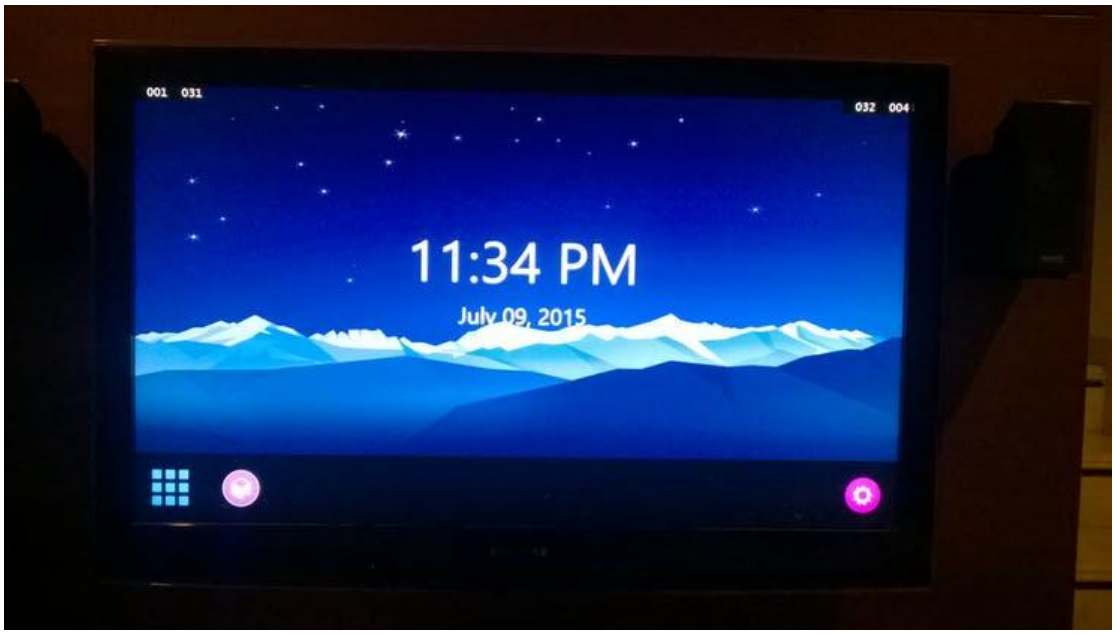
When Arduio boots, it flickers D13 pin. It is good to do not connect any relay pin on D13 because when Arduino boots or reboots, D13 will be flickerd once and if any device connected with Relay (controlled via D13) will flicker, too.

Solution: There are two options: first one, simply do not connect realy with D13. Second one is bit complicated. You can rewrite Aruino 's Opti Boot firmware that do not use D13 while booting and burn that boot loader to Arduino.

Clock Issue

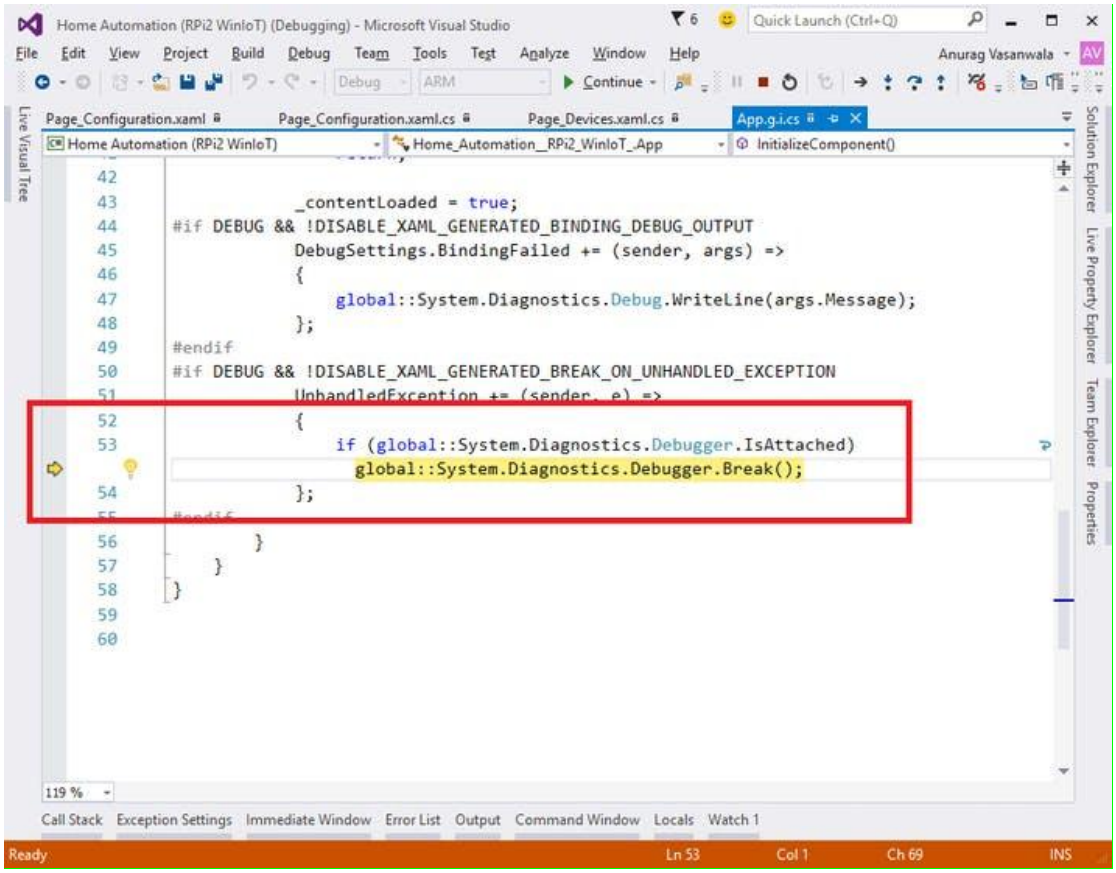
Raspberry Pi 2 does not have on-board Real Time Clock chip. Thus, it is not possible to maintain time after power-failure or fresh boot. Thus, external RTC must be attached and programmed to maintain date time.

Solution: NTP can be used but needs Internet connection or even Arduino can deal with RTC and Raspberry Pi requests for the date time at boot. It is also good to attach external RTC directly with RPi 2.



3.4 Debugging Issue: -

We had faced many issues while developing Universal App in C#. Universal app is superset of WinRT and thus so much asynchronous operation needed to be performed. If error arises in asynchronous operation, it becomes headache to solve. Whenever any exception or error occurred, most of time I faced following screen:



No point to the error prone line, no programmer friendly message, nothing. I have to place so many breakpoints to find out the cause of the error. In debugging, sometimes universal app just shuts-down without any prior notice and suddenly my watch window becomes useless. The cause of the behavior is described at.
Solution: Place breakpoint before the expected error prone line.

In short, at this moment (while this article was published) universal app is great but lacks of debugging features as conventional Windows application have.

There is no limitation when it comes to features, new ideas or even to overcome existing limitation. But it is not feasible to add each and every feature at once. At this release, this article lets you understand real power of Raspberry Pi 2 and Windows Io T. Windows Universal XAML is great GUI framework for Windows 10 Io T & Raspberry Pi 2 have good graphics processor on board. By combining these two, an extraordinary looking GUI solution can be created. In addition, Raspberry Pi 2 have 900MHz quad-core CPU, which is quite awesome for multi-threaded (in UWP, Task) solutions. This is the basic to intermediate project for those who wants to learn

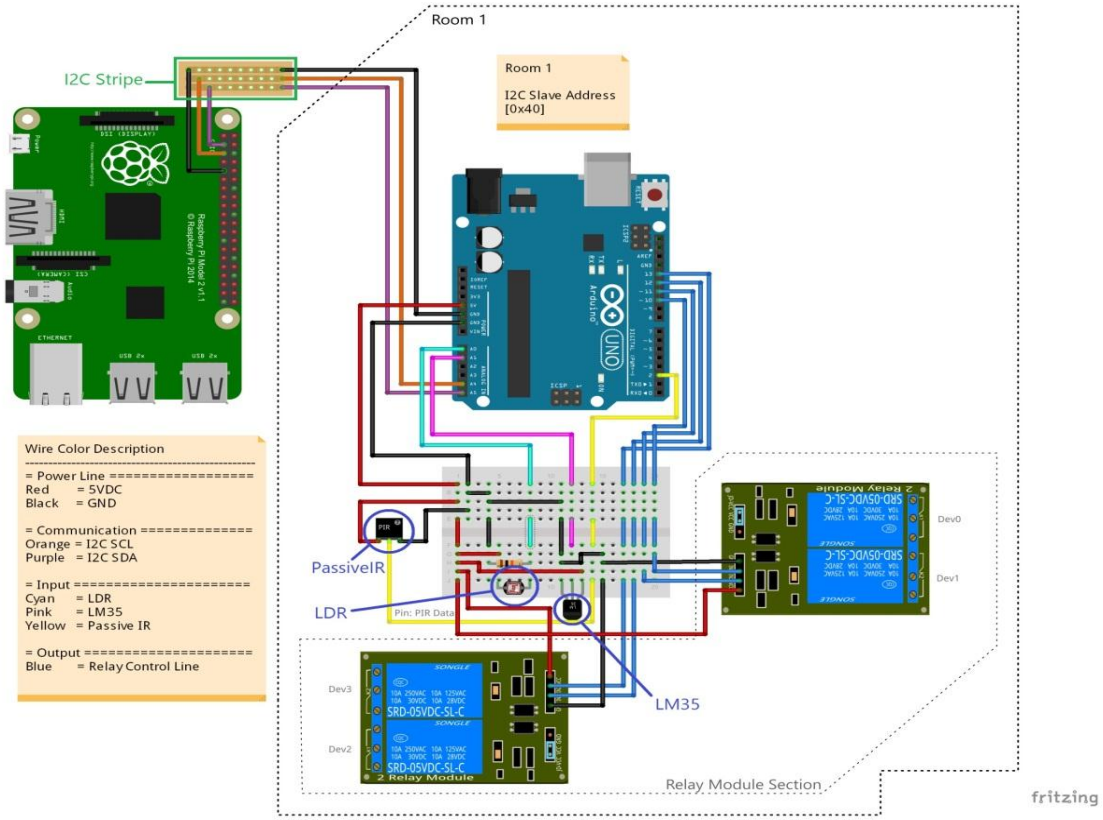
fundamentals of embedded and software system (i.e. I2C bus communication, how to operate appliances, custom protocol design, OOP design for real-world application and wire-frame).

In future, we can add capability to communicate over remote devices using Radio Frequency or Infra-Red instead of I2C bus. Web-management portal using Azure can be integrated for mobile devices. Further the real automation will be integrated like event-based operation, timed operation with RTC chip. For example, turn on backyard lights at 7:00 P.M. and turn back them off at 10:00 P.M.; a good example for event based is to turn on garden's lights when ambient light intensity decreased below specific intensity and so on. So, there is no limitation for this new Windows 10 Io T Core platform for Raspberry Pi 2.

4. Proposed System Design:

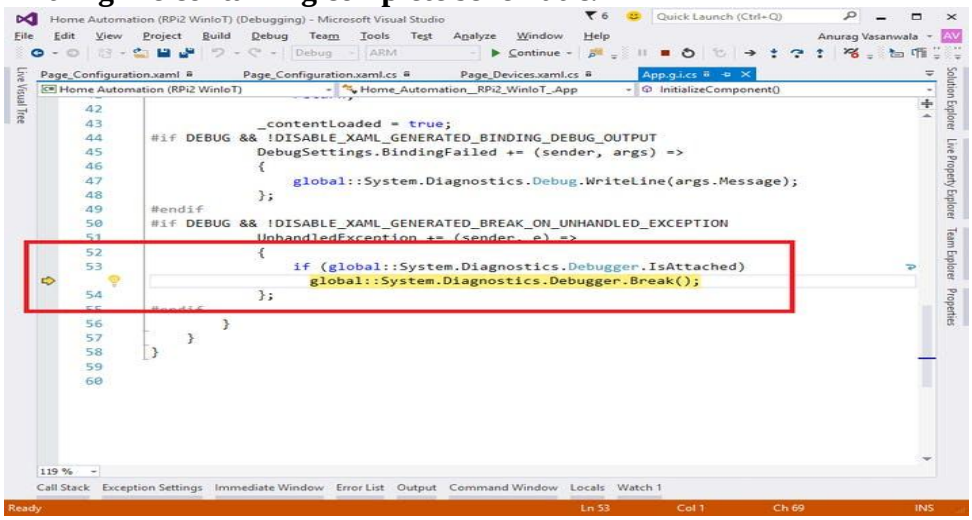
4.1. Proposed Schematic for one room

We can add multiple room by attaching individual Arduino with unique I2C slave address on the bus.



4.2 Fritzing Schematic File:-

Fritzing file containing complete schematic.



Code:-

***/**

Arduino Sketch v0.4

This sketch is written for "Home Automation using Raspberry Pi 2 and Window 10 Io T "Refer this link:

https://www.hackster.io/AnuragVasanwala/home-automation

This sketch is tested on Atmega328p only.

This sketch prepares an Arduino device as slave device on I2C bus operated by
Raspberry Pi 2 running Windows 10 Io T Core.

Objectives:
+Periodically collect sensor data (Function: loop(
+On Recevive, collect 3-bytes mode instruction
and performa operation based upon it. (Function: Receive Data(
+On Request, send 14-bytes response array based
on selected mode by On Receive. (Function: Send Data(

This sketch is provided as it is without any WARRANTY. You can use it for personal as well as
commercial use.

I am not liable for any loss of data or injuries caused by this sketch.

```
/*

#include <Wire. h  <
#define _DEBUG_

 */Arduino's I2C Slave Address/*
#define SLAVE_ADDRESS 0x40

 */PIN DECLARATION/*
int Pin _Ambient Light _LDR = A0؛
int Pin _Passive IR = 2؛
int Pin _Temperature = A1؛

 */Global Variable/*
volatile short Value _Ambient Light _LDR, Value _Temperature؛
volatile bool  Value _Passive IR؛

 */Protocol Variable/*
byte Mode, Pin, Value؛
byte Response[14؛[


void setup()
}
          //Initialize pins
   Pin Mode(Pin _Ambient Light _LDR, INPUT؛(
   Pin Mode(Pin _Passive IR, INPUT؛(
   Pin Mode(Pin _Temperature, INPUT؛(
   Pin Mode(0, OUTPUT؛(
   Pin Mode(1, OUTPUT؛(
   Pin Mode(3, OUTPUT؛(
   Pin Mode(4, OUTPUT؛(
   Pin Mode(5, OUTPUT؛(
   Pin Mode(6, OUTPUT؛(
   Pin Mode(7, OUTPUT؛(
   Pin Mode(8, OUTPUT؛(
   Pin Mode(9, OUTPUT؛(
   Pin Mode(10, OUTPUT؛(
   Pin Mode(11, OUTPUT؛(
   Pin Mode(12, OUTPUT؛(
   Pin Mode(13, OUTPUT؛(
   Pin Mode(A2, OUTPUT؛(
   Pin Mode(A3, OUTPUT؛(

#ifdef _DEBUG_
   Serial .begin(9600؛(
#end if

          //Initialize I2C Slave on address 'SLAVE_ADDRESS'
   Wire .begin(SLAVE_ADDRESS؛(
```

```
   Wire .on Request(Send Data‹(
   Wire .on Receive(Receive Data‹(
{

void loop()
}
 //  Read LDR
       //  Arduino supports 10-bit Analog Read.
          //  Thus we need to convert it into 8-bit.
  Value _Ambient Light_ LDR = analog Read(Pin _Ambient Light _LDR‹(
  Value _Ambient Light _LDR = map(Value _Ambient Light _LDR, 0, 1023, 0,
255‹(

 //  Read Passive IR value
  Value _Passive IR = (digital Read(Pin _Passive IR) == HIGH) ? true : false‹

 //  Read Temperature Sensor and Convert Voltage into Celsius
  Value _Temperature = (short)((float)(analog Read(Pin _Temperature) *
0.48828125‹((

 //  Wait for 100 ms
  delay(100‹(
{

 //Callback for I2C Received Data
void Receive Data(int byte Count(
}
          //Read first byte which is Protocol Mode
  Mode = Wire .read‹()

          //Read second byte which is Pin. Only Valid for Mode 2
  Pin = Wire .read‹()

          //Read third byte which is Pin-Value. Only Valid for Mode 2
  Value = Wire .read‹()

          //Signal specified pin if Mode 2 is received
  if (Mode == 2(
}
     Digital Write(Pin, Value‹(
{

#if def _DEBUG_
   Serial .print(Mode‹(
   Serial .print‹(" ")
   Serial .print(Pin‹(
   Serial .print‹(" ")
   Serial .print ln(Value‹(
#end if
{
```

```
void Send Data()
}
   switch (Mode(
}
      case 0: // Mode: Read Sensor
         Response[0] = (byte)Value _Ambient Light _LDR؛

 //          Value _Passive IR is boolean so that we need to convert it into byte
         Response[1] = (byte)((Value _Passive IR == true) ? 1 : 0؛(

 //          Response[2] byte is Sign byte for Temperature
- - ٠        //           ve Temperature
+ - ١        //           ve Temperature
         Response[2] = (byte)((Value _Temperature < 0) ? 0 : 1؛(

         Serial. Print ln(Value _Temperature؛(

- //          ve Temperature can't be sent in byte. Convert it into +ve equivalent
         Response[3] = (byte)((Value _Temperature < 0) ? (Value
_Temperature*(-1)) : Value _Temperature؛(
         break؛
      case 1: // Mode: Read Devices State
         Response[0] = (digital Read(0) == HIGH) ? 1 : 0؛
         Response[1] = (digital Read(1) == HIGH) ? 1 : 0؛
         Response[2] = (digital Read(3) == HIGH) ? 1 : 0؛
         Response[3] = (digital Read(4) == HIGH) ? 1 : 0؛
         Response[4] = (digital Read(5) == HIGH) ? 1 : 0؛
         Response[5] = (digital Read(6) == HIGH) ? 1 : 0؛
         Response[6] = (digital Read(7) == HIGH) ? 1 : 0؛
         Response[7] = (digital Read(8) == HIGH) ? 1 : 0؛
         Response[8] = (digital Read(9) == HIGH) ? 1 : 0؛
         Response[9] = (digital Read(10) == HIGH) ? 1 : 0؛
         Response[10] = (digital Read(11) == HIGH) ? 1 : 0؛
         Response[11] = (digital Read(12) == HIGH) ? 1 : 0؛
         Response[12] = (digital Read(A2) == HIGH) ? 1 : 0؛
         Response[13] = (digital Read(A3) == HIGH) ? 1 : 0؛
         break؛
      case 2: // Mode: Set Device State
         Response[0] = (digital Read(Pin) == HIGH) ? 1 : 0؛
         break؛
      default:
         break؛
{

          //Wire back response
  Wire .write(Response, 14؛(
}
```

4.3 Discussion:

There are two modes of results for the proposed system, which are automatic and manual control for the inside and outside of the home

 Automatic Control System

o When the temperature value for the inside home system is below 18C°, the microcontroller will turn on the heater LED and at the same time the user will see on the website that the heater icon is turned on. When the temperature value is between 18-25C°, the microcontroller will turn on the fan and the user will see at the same time that the fan icon is activated on the website. When the temperature value is higher than 25C°, the microcontroller will turn on the air-conditioning LED and the user will see that the air-conditioning icon is on .

o If the value of light intensity exceeds 700, the microcontroller will turn on the internal light LED and the user will see on the web site that the internal light icon is turned on

o When the output value of the PIR sensor is high, the microcontroller will turn on the buzzer and the user will see on the website that the icon for motion-alarm is on .

o If the gas sensor reading exceeds the threshold value, the microcontroller will turn on smoke buzzer and the user can see on the website that the value of the sensor exceeds the threshold value, and the icon of a buzzer in automatic control is turned on.

o When the value of the water level sensor for the outside home is higher than threshold value (10cm), the water pump is turned off. When the value of sensor is below the threshold value (10cm), the water pump is turned on.

 Manual Control for the Inside and Outside Home System happen When the homeowner wishes to use manual control of the outside home, such as of fan, internal light or external light, motor pump water, heaters, and air-conditioning, the system will receive data from the website using WiFi connection.

5. Conclusion and Future Work

The project describes a low-cost and convenient method for tracking and regulating temperature and humidity in company building. The availability of such system is extremely preferred particularly, with the establishments, companies. This system give comfort to employees within these institutions and raise their production capacity in addition to reducing electrical energy consumption by maintaining temperatures within their normal limits. In this study used DHT11 sensor examine information about the humidity and temperature of the company building. Arduino Uno is MCU, that get information of temperature from DHT11 sensor and process it and offer to an ESP8266 Module. The ESP8266 module is a Wi-Fi chip that can exchange information to Internet of Things cloud. The monitoring device first shows the monitoring information on the user interface on a web page that was developed, and the system used to view and process. When the device senses an indoor value within the building that is distinct from the fixed value, the device uses the wireless control method to control the temperature/humidity regulating systems to obtain the optimal effect of the thermal comfort. The administrative operations centre is responsible for all of the cloud-related activities. In this proposed system, using HTML/CSS websites to monitor and track data in the database to perform various algorithm analyses and then transfer sensor measurements to cloud server data base which is programmed to web server via Appatshy and Mysql formats. The system had been implemented with high time response so that all readings updated and appeared spontaneously every 5 second. The designed system had been presented very good time response with an effective and secure and rapid response real time system should be achieved.

It is proposed in the future work looks to improve this system by applying more sensors as gas and motion sensor in order to offer the system a robust functionality. In addition, it will transmit and demonstrate the sensed data wirelessly by attaching the smart phone via Bluetooth module or Wi-Fi chip. Finally, data from the sensors can be sent to unique phone numbers through cellular strategy.

References

[1] S. K. Mishra, D. Puthal, B. Sahoo, S. K. Jena, and M. S. J. T. J. o. S. Obaidat, "An adaptive task allocation technique for green cloud computing," vol. 74, no. 1, pp. 370-385, 2018.

[2] I. Sarkar, B. Pal, A. Datta, and S. Roy, "Wi-Fi-based portable weather station for monitoring temperature, relative humidity, pressure, precipitation, wind speed, and direction," in *Information and Communication Technology for Sustainable Development*: Springer, 2020, pp. 399-404.

[3] I. Froiz-Míguez, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. J. S. Castedo, "Design, implementation and practical evaluation of an IoT home automation system for fog computing applications based on MQTT and ZigBee-WiFi sensor nodes," vol. 18, no. 8, p. 2660, 2018.

[4] J. Yun and K.-H. J. S. Won, "Building environment analysis based on temperature and humidity for smart energy systems," vol. 12, no. 10, pp. 13458-13470, 2012.

[5] W.-T. Sung, S.-J. Hsiao, and J.-A. J. J. o. S. Shih, "Construction of indoor thermal comfort environmental monitoring system based on the IoT architecture," vol. 2019, 2019.

[6] G. Satyanarayana and S. Mazaruddin, "Wireless sensor based remote monitoring system for agriculture using ZigBee and GPS," in *Proceedings of the Conference on Advances in Communication and Control Systems-2013*, 2013, pp. 110-114: Atlantis Press.

[7] R. Ab Rahman, U. R. a. Hashim, S. J. B. o. E. E. Ahmad, and Informatics, "IoT based temperature and humidity monitoring framework," vol. 9, no. 1, pp. 229-237, 2020.

[8] M. Saad, H. Abdoalgader, and M. Mohamed, "Automatic Fan Speed Control System Using Microcontroller," in *6th Int'l Conference on Electrical, Electronics & Civil Engineering (ICEECE'2014) Nov*, 2014, pp. 27-28.

[9] J. Shah and B. J. P. T. Mishra, "Customized IoT enabled wireless sensing and monitoring platform for smart buildings," vol. 23, pp. 256-263, 2016.

[10] D. Popa, F. Pop, C. Serbanescu, A. J. N. C. Castiglione, and Applications, "Deep learning model for home automation and energy reduction in a smart home environment platform," vol. 31, no. 5, pp. 1317-1337, 2019.

[11] R. R. Hariadi, A. Yuniarti, I. Kuswardayan, D. Herumurti, S. Arifiani, and A. A. Yunanto, "Termo: Smart Air Conditioner Controller Integrated with Temperature and Humidity Sensor," in *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, 2019, pp. 312-315: IEEE.

[12] F. Sharmin, N. N. Moon, M. Saifuzzaman, A. Hasan, M. A. Hossain, and F. N. Nur, "Humidity Based Automated Room Temperature Controller Using IoT," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 2019, pp. 226-231: IEEE.

[13] K. Naik and S. J. W. N. Patel, "An open source smart home management system based on IOT," pp. 1-7, 2018.

[14] M. A. P. Utomo, A. Aziz, and B. Harjito, "Server room temperature & humidity monitoring based on Internet of Thing (IoT)," in *Journal of Physics: Conference Series*, 2019, vol. 1306, no. 1, p. 012030: IOP Publishing.

[15] X. Bajrami, I. J. e. Murturi, and i. E. u. Informationstechnik, "An efficient approach to monitoring environmental conditions using a wireless sensor network and NodeMCU," vol. 135, no. 3, pp. 294-301, 2018.

[16] A. Iqbal *et al.*, "Interoperable Internet-of-Things platform for smart home system using Web-of-Objects and cloud," vol. 38, pp. 636-646, 2018.

[17] Q. Liu, Y. Ma, M. Alhussein, Y. Zhang, and L. J. C. N. Peng, "Green data center with IoT sensing and cloud-assisted smart temperature control system," vol. 101, pp. 104-112, 2016.

[18] M. Amarlingam, P. K. Mishra, K. D. Prasad, and P. Rajalakshmi, "Compressed sensing for different sensors: A real scenario for WSN and IoT," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016, pp. 289-294: IEEE.

[19] T. D. Roy and J. Singh, "A Note on Wired and Wireless Sensor Communication Using Arduino Board and NodeMCU," in *Smart Systems and IoT: Innovations in Computing*: Springer, 2020, pp. 281-292.

[20] S. Shakthidhar, P. Srikrishnan, S. Santhosh, and M. Sandhya, "Arduino and NodeMcu based Ingenious Household Objects Monitoring and Control Environment," in *2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 2019, vol. 1, pp. 119-124: IEEE.

[21] J. N. Robbins, *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics*. " O'Reilly Media, Inc.", 2012.

[22] A.-V. Nedelcu, F. Sandu, M. Machedon-Pisu, M. Alexandru, and P. Ogrutan, "Wireless-based remote monitoring and control of intelligent buildings," in *2009 IEEE International Workshop on Robotic and Sensors Environments*, 2009, pp. 47-52: IEEE.

[23] M. A. Zamora-Izquierdo, J. Santa, and A. F. J. I. P. C. Gómez-Skarmeta, "An integral and networked home automation solution for indoor ambient intelligence," vol. 9, no. 4, pp. 66-77, 2010.

[24] G. Vivek and M. Sunil, "Enabling IOT services using WIFI-ZigBee gateway for a home automation system," in *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2015, pp. 77-80: IEEE.

[25] R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266," in *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2016, pp. 1-5: IEEE.

[26] J. Xiao and J. T. J. P. C. S. Li, "Design and implementation of intelligent temperature and humidity monitoring system based on ZigBee and WiFi," vol. 166, pp. 419-422, 2020.