Semiconductor Optoelectronics
ISSN: 1001-5868

AN APPROACH TO SMART HOME SYSTEM USING ARDUINO AND ESP01 WI-FI MODULE

Ganga Shirisha M S#1, Dr. Chandrakant Naikodi #2
#1Assistant Professor, Department of Computer Science,
SSA GFG College(A), Bellary-583101
#2Associate Professor and Chairman, Department of Studies in Computer Science,
Davangere University, Davangere-577007
1shirisha1677@gmail.com

Abstract— Internet of Things (IoT) has recently made significant strides, making it one of the most promising study areas for developing cutting-edge and practical technologies in the future. The ideal smart house will be able to automate daily duties utilising a variety of electronic equipment and gadgets. Smart home automation has a significant positive impact on human life. In this paper an autonomous system is progressed where it can perform monitoring the condition of smart environment via a variety of sensors and communicate the required data to the cloud in order to often change the comfort level in the home. As part of this paper, an attempt made to implement prototype, where the setup is using an ESP8266-01 connectivity module, an Arduino UNO microcontroller, a DHT11 sensor for measuring temperature and humidity, and the ThingSpeak cloud for data storage. We have evaluated the system by conducting experiment on different environmental conditions.
Keywords: Arduino UNO, DTH11, Sensors, ESP8266-01, IoT, Wi-Fi, ThingSpeak

1. INTRODUCTION

The expectation of leading a simple yet technologically advanced existence continues to rise as rapid technological advancement always seeks to benefit humans. Humans now rely heavily on the internet for their social and educational activities; without it, they would be completely useless.

Home automation is the use of computers and information technology to conveniently operate household equipment. The home, housework, or other household function has been automated. Centralized management of Light, Appliances, Temperature, and other systems may be a part of home automation, which aims to increase security, comfort, and energy efficiency.

House automation has been a long staple of science fiction literature, but it has only just started to become a reality because to the broad adoption of electricity in homes and the explosive growth of information technology.

The first automated home appliances, such the kettle in 1889 and the washing machine in 1904, were introduced roughly 100 years ago when electric power was first introduced to domestic homes.

Because it is now much more accessible and simple thanks to connectivity via Smartphone and

tablets, home automation is becoming more and more popular. The "Internet of Things" idea and the rise in popularity of home automation go hand in hand. Systems and appliances are able to communicate in an integrated way through the integration of information technologies with the home environment, which has advantages for comfort, energy efficiency, and safety.
The Internet of things (IoT) devices not only control but also keep an eye on the many mechanical, electrical, and electronic systems utilized in different kinds of infrastructure. A single user (also known as the admin) controls these devices that are connected to the cloud server, and their commands are then sent to or received by all authorized users connected to that network. Using various network infrastructures, various electrical and electronic equipment are connected and remotely controlled.
For storing the data on the Thingspeak cloud, we have utilized an ESP8266-01 Wi-Fi module and an Arduino UNO board in this project. The remaining sections of this essay are structured as follows: An overview of the system is given in Section 2 explains the hardware and software design, Section 4 describes the implementation, and Section 5 discusses the results and analysis of the experiment. The report comes to a close by examining the recommendations and additional research needed to improve the system. Future study will focus on creating a full-scale smart home automation system and leveraging blockchain technology to secure the gadget, making it more secure overall.

1. HARDWARE AND SOFTWARE USED:

1.1 ARDUINO UNO

One of Arduino standard boards is the UNO. Here, "one" is denoted by the Italian term UNO. The name UNO was given to the Arduino Software's original release in order to distinguish it. Additionally, it was Arduino's first-ever USB board to be made available. It is regarded as a strong board that is employed in many projects. The Arduino UNO board is shown in Figure 1 and was created by Arduino.cc.
The ATmega328P microprocessor is the foundation of the Arduino UNO. Comparatively speaking to other boards, such as the Arduino Mega board, etc., it is simple to use. Shields, various circuits, and digital and analog Input/output (I/O) pins make up the board.

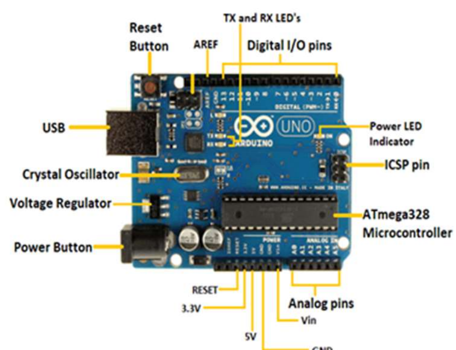


Figure 1: Arduino UNO

The Arduino UNO has 14 digital pins, a USB port, a power jack, and an ICSP (In-Circuit Serial Programming) header in addition to 6 analogue pin inputs. It is based on IDE for programming. which stands for Integrated Development Environment

1.2 **ESP8266-01** Serial Wireless Transceiver Module (Serial WIFI).

The ESP8266 shown in Figure 2 is capable of offloading all Wi-Fi networking tasks from another application processor or hosting an application. Each ESP8266-01 module has an AT command set firmware that has been pre-programmed, so all you have to do is connect it to your Arduino device to get nearly the same Wi-Fi features i.e. similar to Wi-Fi Shield. The ESP8266 module is a very affordable board.

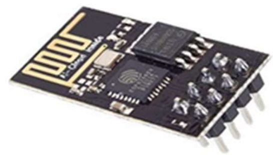


Figure 2: ESP8266-01

1.3 DTH11 Sensor

A cheap digital sensor for detecting humidity and temperature is the DHT11 (shown in Figure 3). To instantly detect humidity and temperature, this sensor may be simply interfaced with any micro-controller, including Arduino, Raspberry Pi, etc. The DHT11 sensor comprises of a thermostat for measuring temperature and a capacitive humidity-sensing device. The DHT11 has a temperature range of 0 to 50 degrees Celsius with a 2-degree precision.

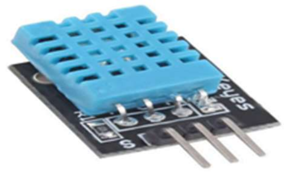


Figure 3: DTH11 Sensor

1.4 THE ARDUINO SKETCH

An open-source programme called Arduino IDE (Integrated Development Environment) is used to create, compile, test, and upload programmes to Arduino boards. There are various particular methods and functions introduced, and the code is written in C/C++.

If no compilation errors occur and the number of dynamic variables does not exceed the available memory, the Arduino programme, known as "sketch,". A sketch can be safely uploaded onto the board as shown in Figure 4,

Since the board's built-in flash memory is non-volatile, the sketch is not lost when the power is turned off. In this manner, the board will automatically reload the sketch and run when powered up again.

Figure 4: ARDUINO SKETCH

1.5 THINGSPEAK

An IoT analytics platform service called ThingSpeak enables you to gather, visualise, and analyse real-time data streams in the cloud. Data sent by your devices to ThingSpeak is instantly visualised by ThingSpeak. You can analyse data online as it comes in thanks to ThingSpeak's ability to run MATLAB code. For IoT systems that need analytics, ThingSpeak is frequently used for prototype and proof of concept solutions.

The Figure 5 below can be used to summarise numerous IoT systems at a high level:
The IoT "things"—smart devices—that reside at the network's edge are shown on the left. Wearable technology, wireless temperature sensors, heart rate monitors, hydraulic pressure sensors, and factory floor equipment are just a few examples of the devices that collects the data.
We have the cloud in the middle, where data from many sources is combined and instantly evaluated, frequently by an IoT analytics platform created for this function.
To enable the engineer or scientist to prototype algorithms that may ultimately run in the cloud or on the smart device itself, the data is retrieved from the IoT platform and placed into a desktop software environment.
All of these components are makeup IoT system. ThingSpeak provides a platform for swiftly gathering and analysing data from internet-connected sensors and fits it in the cloud area as shown in the Figure 5.

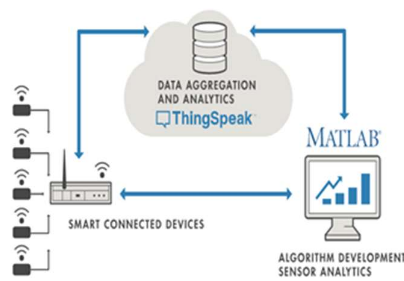


Figure 5: THINGSPEAK

2. Literature Survey

As part of literature study, we tried to provide all the latest development and their corresponding future trends. Following table highlights contribution, limitations and future enhancements.

Sl. No	Reference Paper	Contribution	Limitations	Future Enhancement
1	Sharkawy .et al.[1]	The prototype for smart home was designed and controlled via an android phone and the arduino platform. Which was accomplished using the Bluetooth module HC-06.	The data can be accessed within the range of the Bluetooth and not beyond that.	Can be modified to store and access the data beyond the range.

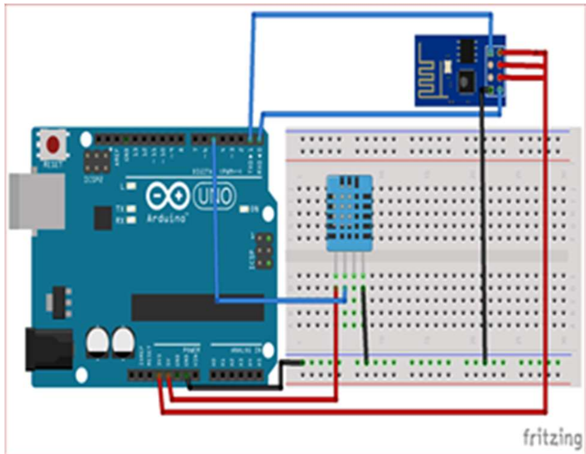
2	Chioran . et al.[4]	Developed an smart home system in which the data was successfully exchanged between the arduino board and the windows application.	The stored data can be used for analysing but cannot be remotely accessed and the board was connected to the laptop.	Can modify the system so that the data can be stored onto the cloud and sent to the mobile phone so that it can be accessed remotely.
3	Naing, Ma., et al.[6]	The prototype was designed and implemented with 2 arduino Nano microcontrollers which sent SMS alerts to the owner and also read different sensors data and displayed on LCD.	Date can be viewed only in real time in the display and cannot be used to analyse as it is not stored on to the cloud.	Can modify the system to store the sensed data on to the cloud for further analysing.
4	Satapathy .,et al[8]	The experimental setup was designed and implemented for a smart home where the data was successfully sensed by the sensors and sent to the mobile phone and LCD display.	Even though Wi-Fi device was used in the experiment the data can only be viewed in the real time and cannot be used for analysing.	Modify the system to store the data onto the cloud. Which can be further used for analysing the data.
5	Mitra .,et al. [9]	The experimental setup was done for the smart home where the data was successfully sensed b the sensor and sent to the web/ android application for controlling the device.	As the data is not stored anywhere we cannot use it for analysing.	Can be modified to store the sensed data and be used for analysing it.
6	Abel A.Zandamela [11]	The prototype was designed and implemented for a smart home system. Which was able to send real-time video and GSM based information of break-in fire, motion detection and temp and humidity monitoring.	Date can be viewed only in real time and cannot be used to analyse as it is not stored on to the cloud.	Can modify the system to store the sensed data on to the cloud for further analysing.
7				

	Asadullah., et al.[12]	The proposed system has the ability to interface up to 18 home appliances and sensors. It was analyzed and tested within the range of 20 meters and it achieved 100% accuracy.	System is only able to control the appliances within short range and the sensed data was not stored.	Can modify the system to increase the range and store the sensed data.
8	Mahalakshmi,G., et al.[13]	The experimental setup was done for the smart home system where the data was successfully sensed by the sensors and transferred to the web application or the android phone for the real time viewing.	Data can be only viewed in real time and cannot be further used for analysing as it is not stored.	Can be modified to store the data on to the cloud which can be further used for analysing purpose.
9	Mehta.,et al.[14]	The system was developed and implemented for the smart home where the data from the sensor are passed to web/ android application to take control over the device remotely.	As the data can be viewed in real time and not stored anywhere it cannot be used for analysing.	Can be modified to store the data on to the cloud and use it to analyse remotely.
10	J.Chandramohan., et al.[15]	The experimental setup was designed and implemented for a smart home where the data was successfully sensed by the sensors and can be controlled by smart phone and web page.	Even though Wi-Fi device was used in the experiment the data can only be viewed in the real time and cannot be used for analysing.	Modify the system to store the data onto the cloud. Which can be further used for analysing the data.

Smart home using IoT has gained a lot of interest, and recent publications in this field. In the above table include few surveys and review papers. Some of the papers [3,4,5,6,7,8,9,10] even though Wi-Fi module was used the data can be viewed in the real-time but cannot be used for analysing as it was not stored. In the paper [2] the data was stored and can be used for analysing but cannot be accessed remotely as the device was physically connected to the system without using Wi-Fi or Bluetooth device. According to paper [1] thought the experiment used Bluetooth device the data was not stored for analysing and can be accessed only within the range of the Bluetooth device.

3. Implementation

The experimental setup was done as shown in the Figure 7 using breadboard and jumper wiresthe DTH11, arduino UNO and the ESP01 Wi-Fi module were connected. The entire circuit diagram is shown in the figure 6.



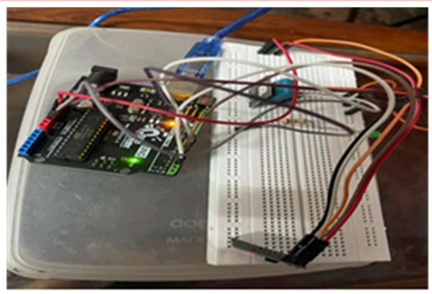


Figure 7: Experimental Setup

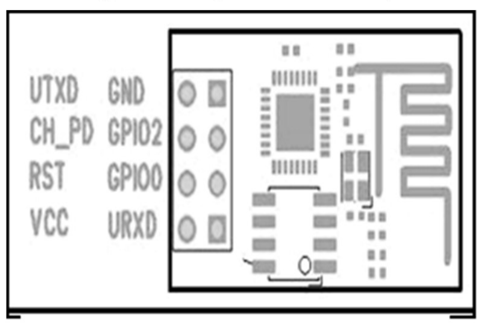


Figure 8: ESP8266-01

Here we will use 5 pins of ESP01 to connect with Arduino UNO Board. Which include VCC, CH_PD (Chip Power Down Pin), GND, URXD, and UTXD pins as shown in the Figure 8. Arduino Uno has a single UART interface found on pin 0 (RX0) and pin 1 (TX0). Therefore, URXD is connected to TX0 and UTXD will be connected to RX0 pin of Arduino UNO. The VCC and CH_PD of the ESP01 are connected to the 3.3V output pin of Arduino UNO. CH_PD is active low therefore passing 3.3V to it will enable the chip. Lastly, connect the ground of the ESP01 with the ground of the Arduino UNO.

The code was written in Arduino Integrated Development Environment (IDE) [also be called as the arduino sketch] and uploaded on to arduino UNO microcontroller. A power supply of 5 volts was provided to the Arduino UNO to process the code.

The experiment was successfully executed the DTH11 sensor sensed the temperature and humidity and the readings were successfully sent to the thingspeak cloud and stored.

4. Results and Analysis

The data stored in the thingspeak cloud can be used to analyse and use for taking decisions. Figure 9 and Figure 10 shows the graph of the data uploaded to the cloud. To show temperature and humidity respectively.

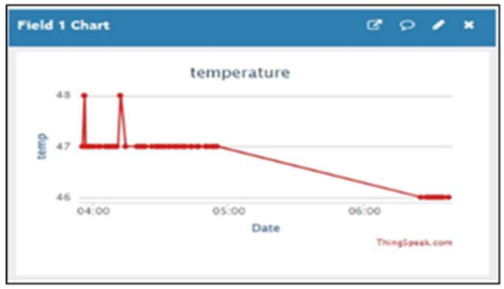


Figure 9: Temperature readings in Thingspeak

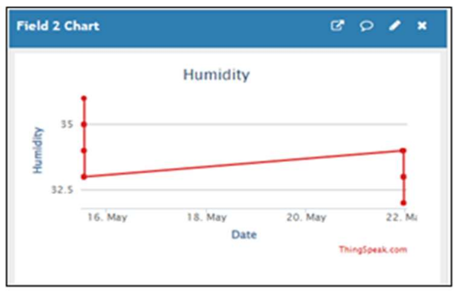


Figure 10: Humidity reading in Thingspeak

The uploaded data can be used to perform analysis such as finding the maximum and minimum temperature as shown in the Figure 11, calculating the average humidity as shown in the Figure 12 and converting temperature from Fahrenheit to Celsius as shown in Figure 13.

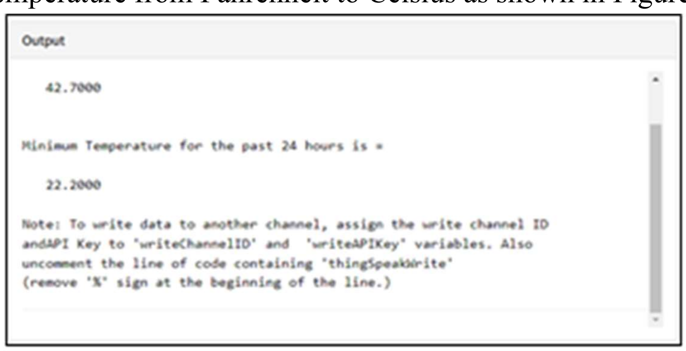


Figure 11: Analyzing the minimum temperature for the past 24 hours

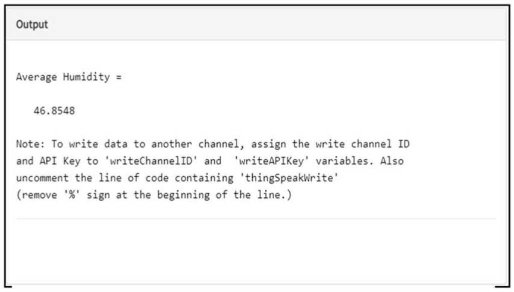


Figure 12: Analyzing Average Humidity

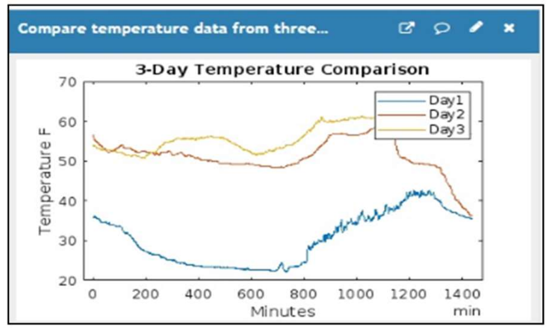


Figure 14: Analyzing of 3-day compression in Thingspeak

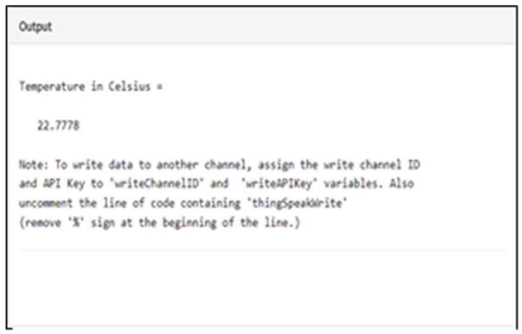


Figure 13: Converting temperature from Fahrenheit to Celsius

Furthermore, the data can also be used for making comparisons like the temperatures of 3 days and the correlation between temperature and humidity as shown in the Figure 14 and Figure15 respectively.

The following piece of code is used to compare the temperature of 3 days and draw the graph.

The following piece of code is used to find the Correlation analysis between temperature and humidity.

```
data = thingSpeakRead(readChannelID,'Fields',
         [TemperatureFieldID HumidityFieldID],
          ...'NumPoints',300, ...'ReadKey',readAPIKey);
temperatureData = data(:,1);
humidityData = data(:,2);
scatter(temperatureData,humidityData);
```

```
oneDay = [datetime('yesterday') datetime('today')];
temperatureDay1 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
                  'dateRange', oneDay, 'ReadKey',readAPIKey);
temperatureDay2 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
                  'dateRange',oneDay-days(1),'ReadKey',readAPIKey);
temperatureDay3 = thingSpeakRead(readChannelID,'Fields',myFieldID, ...
                  'dateRange', oneDay-days(2),'ReadKey',readAPIKey);
myTimes1 = minutes(1:length(temperatureDay1));
myTimes2 = minutes(1:length(temperatureDay2));
myTimes3 = minutes(1:length(temperatureDay3));
plot(myTimes1,temperatureDay1, myTimes2,temperatureDay2, myTimes3,
temperatureDay3);
```

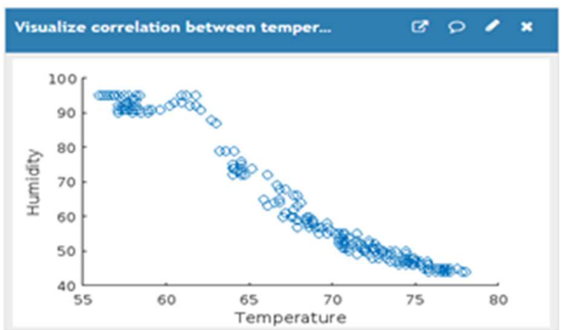


Figure 15: Correlation analysis between temperature and humidity

The DHT11 is a calibrated temperature and humidity sensor module that connects to an Arduino's digital IO pin with precision. The temperature is provided by the DHT11 in Celsius format. The temperature is converted by the Arduino program into Fahrenheit, Kelvin, and Rankine before being sent over serial port.

DHT11 digital temperature / humidity sensor delivers temperatures between 0°C and +50°C and humidity between 0% to 100%.

The temperature accuracy is ±2°C (maximum).

The digital IO pin on the Arduino is wired to the DHT11 data pin. Every two seconds, Arduino scans the temperature and humidity and delivers the data to the serial port.

The following list contains the Celsius to other scale conversion formula.

Fahrenheit: - T(°F) = T(°C) × 9/5 + 32

Kelvin: - T(K) = T(°C) + 273.15

Rankine: - T(°R) = (T(°C) + 273.15) × 9/5

The DHT11 can measure relative humidity only. So first, you should know what the relative humidity is and how to calculate relative humidity.

The total mass of water vapour in a particular volume or mass of air is the absolute humidity. The temperature is not taken into account. The difference between the amount of water vapour that is actually in the air and the amount that the air may possibly contain at a certain

temperature is known as relative humidity.
The formula to calculate relative humidity is:

$$RH = (\frac{\rho_w}{\rho_s}) \times 100\%$$

Where,

RH means – Relative Humidity
ρw means – density of water vapor at a certain temperature,
ρs means – the density of water vapor at saturation at that temperature.
The relative humidity is expressed as a percentage. 0% RH means the air is completely dry and at 100% RH condensation occurs.
Absolute humidity is the measure of the total mass of water vapor in a given volume of air. Mathematically, absolute humidity can be written as the ratio of the volume of the air and water vapor mixture.

$$AH = \frac{mH_2O}{V_{net}}$$

where,
AH means - Absolute humidity
mH_2O means – Mass of Water vapour
V_{net} means – Volume of the air and water vapour mixture
While variations in air temperature or pressure have an impact on the former's value, the computation of absolute humidity does not account for the system's temperature.
Specific humidity SH is the ratio of water vapour mass mw to the total (i.e., including dry) air mass m (namely, m=mw+md). . It's usually referred to as the 'moisture content'. The definition is described as

$$SH = \frac{m_w}{m} = \frac{m_w}{m_w + m_d}$$

Where,

SH means - Specific humidity

m_w means- mass of water vapour

m_d means – mass of dry air

5. CONCLUSION:

Electrical components in a home are connected via a smart home system. This paper presents a design and implementation concept for a Wi-Fi module with Arduino microcontroller board-based smart home automation system. The microcontroller board is attached to the sensors and signals from relevant sensors are received, the home environment can be automatically monitored and accessible. To access the devices remotely, you can use any Android-based smart phone with integrated Wi-Fi. The embedded Temperature and Humidity sensors were evaluated and during system performance testing, and the cloud environment is used to store the data. Using Internet communications, entire system is managed. It is important access that the data transfer between the microcontroller board and the cloud is successful. With the help of these systems, we can create smart houses that are affordable, adaptable, and capable of fixing problems while consuming less energy. Moreover, the user can increase the number of sensors without changing the entire system hence scalability.

REFERENCES

1. Sharkawy, Abdel-Nasser, et al. "Development of Smart Home Applications Based on Arduino and Android Platforms: An Experimental Work." Automation 3.4 (2022): 579-595.
2. Hamdan, Yasir Babiker. "Smart home environment future challenges and issues-a survey." Journal of Electronics 3.01 (2021): 239-246.
3. Sunwar, Shanty, Asmita Tamang, and Sachin Shrestha. "SMART HOME USING IOT." KEC Conference 2021
4. Chioran, Daniel, and Honoriu Valean. "Arduino based smart home automation system." International Journal of Advanced Computer Science and Applications 11.4 (2020).
5. Sarkar, Moupriya, and Sushmita Pramanik. "Smart Home Using Artificial Intelligence." BKGC SCHOLARS (2020): 69-74
6. Naing, Ma, and Ni San Hlaing. "Arduino based smart home automation system." International Journal of Trend in Scientific Research and Development (IJTSRD) 3.4 (2019): 276-280.
7. Nag, Anindya, et al. "IoT for smart homes." Sensors in the Age of the Internet of Things: Technologies and Applications 171 (2019).
8. Satapathy, Lalit Mohan, Samir Kumar Bastia, and Nihar Mohanty. "Arduino based home automation using Internet of things (IoT)." International Journal of Pure and Applied Mathematics 118.17 (2018): 769-778.
9. Mitra, Pritha, Kaustav Madhab Chatterjee, and Asit B. Bhattacharya. "Design of a low cost home automation system using arduino wifi module." Techno International Journal of Health, Engineering, Management and Science (TIJHEMS) (2018).
10. Wadhwani, Siddharth, et al. "Smart home automation and security system using Arduino and IOT." International Research Journal of Engineering and Technology (IRJET) 5.2 (2018): 1357-1359.
11. Zandamela, Abel A. "An approach to smart home security system using arduino." Electrical Engineering: An International Journal (EEIJ) 4.2/3 (2017): 1-18.
12. Asadullah, Muhammad, and Khalil Ullah. "Smart home automation system using Bluetooth technology." 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT). IEEE, 2017.
13. Mahalakshmi, G., and M. Vigneshwaran. "IOT based home automation using Arduino." Int. J. Eng. Adv. Res. Technol 3.8 (2017): 1-6.
14. Mehta, Harsh, et al. "IOT based home automation system using arduino board." International Research Journal of Engineering and Technology (IRJET) 4.01 (2017): 1541-1544.
15. Chandramohan, J., et al. "Intelligent smart home automation and security system using Arduino and Wi-fi." International Journal of Engineering And Computer Science (IJECS) 6.3 (2017): 20694-20698.
16. Alaa, Mussab, et al. "A review of smart home applications based on Internet of Things." Journal of network and computer applications 97 (2017): 48-65.
17. Rajani, U. S., and A. A. Kadar. "GSM based Home Security System using PIR sensor."

Int. Journal of Engineering Research and Applications 8.2 (2017): 87-89.

18. Wang, Yanping, and Zongtao Chi. "System of wireless temperature and humidity monitoring based on Arduino Uno platform." 2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC). IEEE, 2016.
19. Anusha, S., M. Madhavi, and R. Hemalatha. "Home automation using atmega328 microcontroller and android application." International Research Journal of Engineering and Technology (IRJET) 2.06 (2015): 865-868.
20. Bingol, Okan, et al. "Web-based smart home automation: PLC-controlled implementation." Acta Polytechnica Hungarica 11.3 (2014): 51-63.
21. Piyare, Rajeev. "Internet of things: ubiquitous home control and monitoring system using android based smart phone." International journal of Internet of Things 2.1 (2013): 5-11.
22. Miorandi, Daniele, et al. "Internet of things: Vision, applications and research challenges." Ad hoc networks 10.7 (2012): 1497-1516
23. Piyare, Rajeev, and M. Tazil. "Bluetooth based home automation system using cell phone." 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE). IEEE, 2011.