

SCADA Controlled Smart Home Using Raspberry Pi3

Bilal Naji Alhasnawi¹ and Basil H. Jasim²

^{1,2}*Department of Electrical Engineering, University of Basrah, Iraq*
bilalnaji11@yahoo.com, hanbas632@gmail.com

Abstract

The main objective of this article is to develop a smart home system using SCADA, Raspberry Pi3 Module and Wemos-D1 board. In this paper, we are using SCADA, TCP/IP protocol and MQTT to develop a smart home. Modern homes are expected to have centralized control system, instead of conventional switches which are fitted on walls. If the user wants to control the appliances, they have to move near the wall and control the switches. This method is inconvenient for the elderly and disabled people. So their difficulty to operate the appliances can be supported by the proposed smart home system which has a remote module to control the appliances. Remote controlled smart home system fulfills the needs of the people. Remote control operation is achieved by a PC (Personal Computer) is supported by IGSS SCADA. In the PC with TCP/IP protocol acts as a transmitter, which sends commands to the Raspberry Pi3 module. Raspberry Pi3 receives the commands from SCADA. Raspberry Pi3 send the commands with addresses to a terminal unit (Wemos-D1 boards) by MQTT. Each terminal unit receives the address from the Raspberry Pi3, if the terminal unit address matches the Raspberry Pi3, it can control devices or measured parameter, then sends this parameter with its address to the Raspberry Pi3. The Raspberry Pi3 sends these data to owner's PC. In this article Wemos-D1 controller board which contains an ESP8266 IC is used. The loads are interfaced to the Wemos-D1 controller using electromagnetic Relays.

Keyword: *Raspberry Pi3, Wemos-D1 boards, IGSS SCADA, Appliances*

1. Introduction

The smart home system can be defined as the system which enables home's owner to control home appliances and other activates manually and wirelessly locally and remotely (from anywhere in the world) and also to manage home activities by the same ways, also it has smart (automated) security and safety features and also it has capability of smartly manage the energy consumed by the home [1]. In this system, the management of the home process will be automatically according to the conditions that the designer will be determined it or according to frequent duties. This system has several advantages as compared with the manual system. It saves the house owner time, good management of the home, rest of a house owner and saving energy resources.

These smart home systems based on Wireless Sensor Networks (WSNs) are ones that can interrelate intelligently to provide safe living and comfort. The smart home is worldwide a technology for achieving the precision home management. The renewable energy can be used in a smart system for powering the system to make its operation in an independent manner on classical power successes.

2. Literature Review

In 2004 [2], home automation interfaces and device definitions are introduced to let interoperability among ZigBee appliances produced by numerous manufacturers of

Received (November 16, 2017), Review Result (February 6, 2018), Accepted (February 11, 2018)

electrical equipment, meters, and smart energy enabling products. In 2005 [3], Liew, implemented a complete control system to control heating, ventilating, and air conditioning simulated with LabView, the whole system uses the home power line network for connectivity based on the X-10 standard. In 2008 [4], Karataş and Aksoy, used the X-10 standard as the basis of their microcontroller based home automation system and the PIC microcontrollers as a Master – slave approach. They used microcontrollers in control units for data processing and TDA5051A integrated modem for data transmission on the line. They reduced noise by choosing the X10 protocol that sends control message twice; modem uses coupler circuit on sides connected to line .In 2008 [5], BESTEPE used the Public Switch Telephone Network (PSTN) to read a digital electricity meter remotely through microcontroller connected with the serial port of the PC. The software used for remote reading was designed with Delphi 7.0 by using APRO (Asynchronous Professional) libraries. In 2009 [6], Wilson worked on the detection of the undesired interactions between home devices and services while allowing positive interactions between services and devices. He provided an online manager, which is responsible for avoiding interactions among services. He also produced one of the plug and play home network approaches. During his project he tried to minimize user interventions as possible as he can, through introducing the approach of environmental variable. In 2010 [7], LeMay designed a heterogeneous robust sub networks that manage to survive from a disaster. He provided a methodology to enhance the network to form an Emergency-Response Network (ERN) using techniques from mobile ad-hoc networks. This ERN was to provide ability for home network to contact rescuers and /or relatives (specific contacts) in case of disasters. In 2012 [8], Szabo designed a home automation system based on connecting the i²home system with the PLC (programmable Logic Controller), which enables usage of wanted devices (PIR, magnetic, sensors, relays, switches). The system was designed to fulfill the requirements of persons with mild cognitive disabilities and older persons. In 2004 [9], Hasaj designed an interesting smart home system. In his work, the system brain contains a neural network that able to learn automatically the inhabitants' behavior and help them in their everyday routine. The software part designed using JAVA, developed on Linux OS. The system designed to support multiple interfaces. In 2016 [11] smart home designed uses, MQTT Publish/Subscribe Protocol and Django Web Framework to give users the capability to integrate many open source devices with open-source tools and mobile site optimization.

3. Smart Home Activities Using SCADA

SCADA (Supervisory Control and Data Acquisition) is the technology or in another word is a software that enables system designers to collect data from one or more distant facilities and send control instructions to those facilities [13]. This paper presents a solution for smart home using SCADA software. The selected SCADA software is a free version of IGSS (Interactive Graphical SCADA System), and the reasons for this selection are that this software is free and contains almost all of the SCADA features. SCADA software is installed on Windows PC (laptop), and then it should be configured to communicate with the system hardware using one of the available wires or wireless communication ports such as serial RS232 and TCP/IP protocol. In the designed system, IGSS SCADA software is connected to the base station unit (Raspberry Pi3) via TCP/IP protocol, and by this connection, SCADA software can now be linked to all of the system parameters.

4. Methodology

The computer send commands to the base station unit module (Raspberry Pi3) by Wi-Fi network (TCP/IP network). Then, base station unit sends commands to terminal units (Wemos-D1 board) by MQTT network. This flow also works the other way around, the

terminal units send information to the base station unit. Base station unit sent information to the computer through a SCADA by Wi-Fi. Figure 1 shows the diagram of the proposed SCADA system.

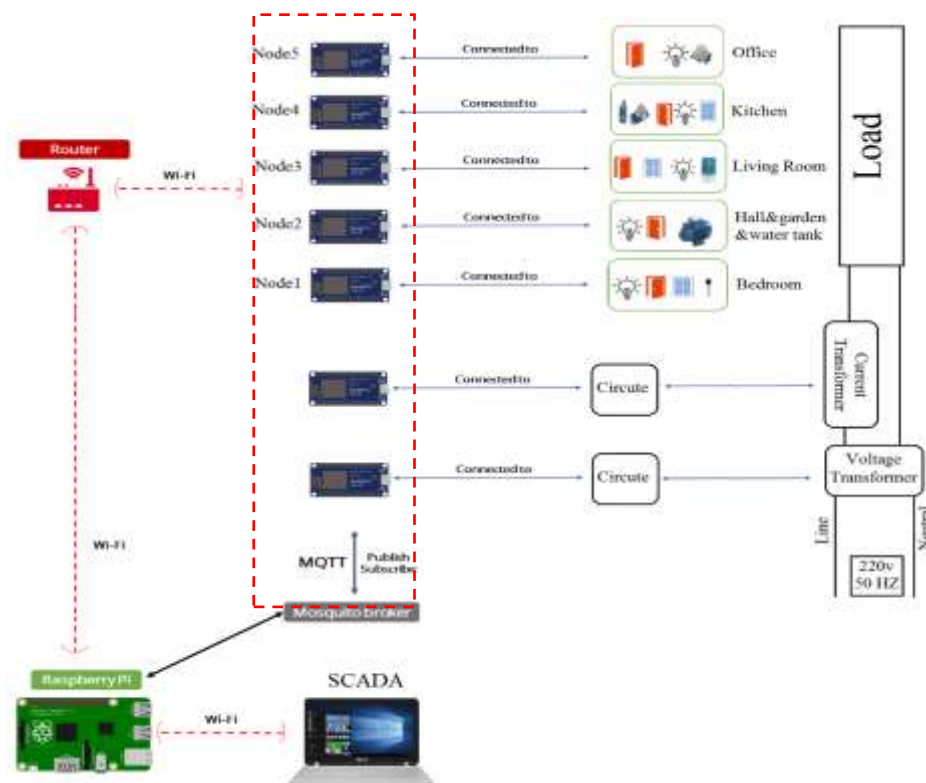


Figure 1. Proposed Overall SCADA System

In this work, a point-to-multipoint topology is used. The Python software has been used for base station unit configuration for building the wireless Wi-Fi network. In this network, all nodes are managed by base station unit. Figure 2 shows part of the base station unit configuration by using the Python software.

```

#!/usr/bin/python
import sys
import os
import time
import socket
import threading
import json
import paho.mqtt.client as mqtt

# MQTT Broker Configuration
MQTT_BROKER = "192.168.1.100"
MQTT_TOPIC = "scada/status"

# Device Configuration
DEVICE_IP = "192.168.1.101"
DEVICE_PORT = 8080

# MQTT Client
client = mqtt.Client("SCADA_Device")
client.connect(MQTT_BROKER)

# Socket Server
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind((DEVICE_IP, DEVICE_PORT))
server_socket.listen(5)

def handle_client(client_socket):
    data = client_socket.recv(1024)
    if data:
        # Parse JSON data
        try:
            json_data = json.loads(data)
            # Publish to MQTT
            client.publish(MQTT_TOPIC, json.dumps(json_data))
        except:
            print("Invalid JSON data")
    client_socket.close()

while True:
    client_socket, address = server_socket.accept()
    thread = threading.Thread(target=handle_client, args=(client_socket,))
    thread.start()

```

Figure 2. The base Station Unit Configuration by using the Python Software

To provide greater detail, it is assumed that each microcontroller board of terminal unit is identified via a single name, and the following scheme is employed for the topics:

1. Messages, from a definite board, are published by the board into MQTT broker using the topic, esp/node name/out. Where a node name identifies the microcontroller (terminal unit).
2. The client messages are published to the particular board by the MQTT broker with the topic, esp/node name/in.

MQTT bridges of a node most subscribe into the topic, esp/node name/in. Forward messages received on this topic through Wi-Fi technology. The same bridge most publish messages from the node into the topic, esp/node name/out. Conversely, specific node client would subscribe into the topic, esp/node name/out, to accept messages from the node, and publish messages to esp/node name/in to send messages into the node. Client's applications and MQTT bridges are available in the Python. Figure 3 shows overview of the SCADA system.

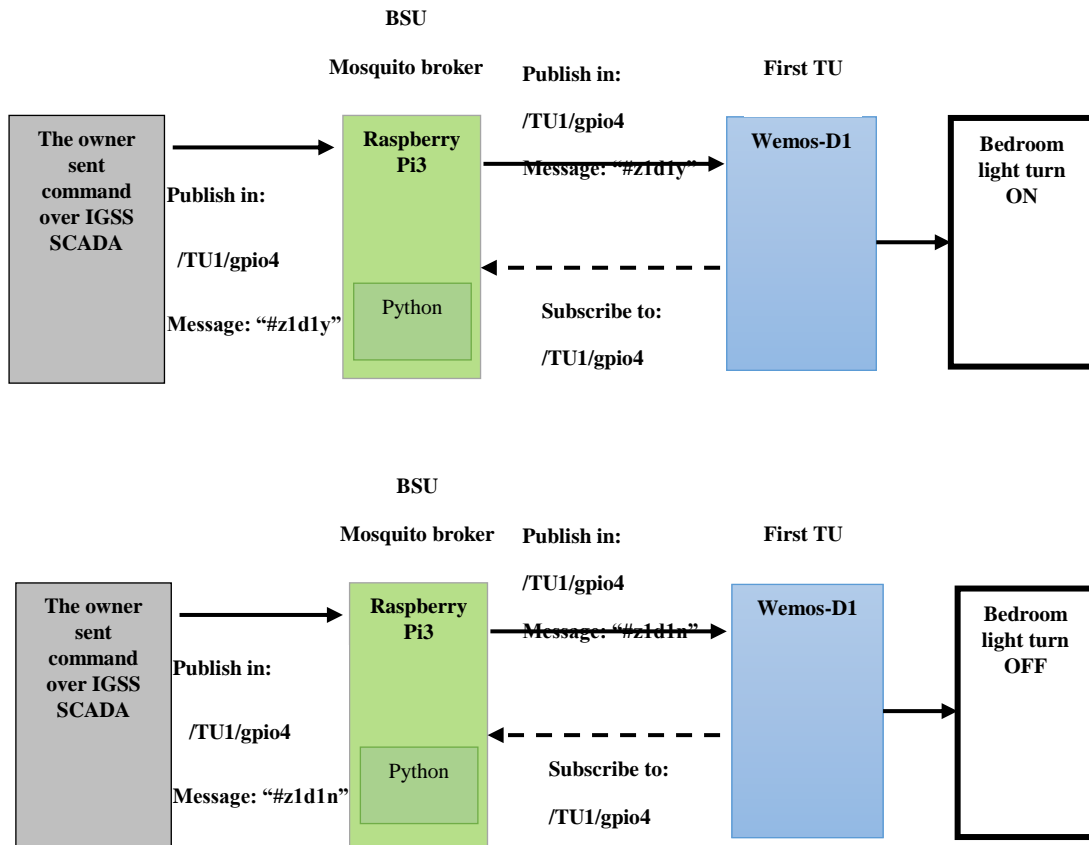


Figure 3. Overview of the SCADA System

5. System Hardware Design

The system hardware consists of a single base station unit, numerous Terminal node lights, fans, water level sensor, gas sensor, flam sensor, and motion sensor. The details are as follows:

1. The Base Station Unit

The base station unit (BSU) plays an important role in the proposed system. The BSU represents the network or the system coordinator. The hardware of the base station consists of a Raspberry pi3 board. Figure 4a shows a block diagram of the system base station, Figure 4b shows the internal structure of a prototype BSU that used for implementing the system.

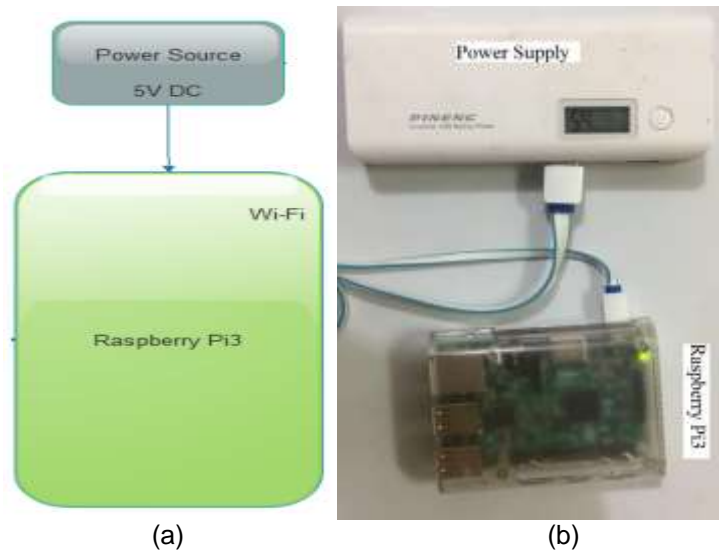


Figure 4. (a) Schematic Diagram of the BSU, (b) System BSU Internal Structure

1.1 The Raspberry Pi3 Board

The RaspberryPi is a sequence of credit card sized single board computer. All RaspberryPi model advantages a Broadcom system on a chip (SoC), which includes an ARM-compatible CPU and an on-chip GPU. Since Raspberry Pi is actually not a microcontroller, but a small-size and fully functional computer, it offers very good computational powers. With CPU speed ranges from (700MHz_1.2GHz), and on board RAM of 1GB, the Raspberry Pi is powerful enough for every complex control operation that would be required at nodes. Though it does not have on- board Flash Memory, an external SD card should be used with the capability from 8- 16GB allows to store multiple programs and even images from security camera module if needed. The Raspberry Pi could also load and run Linux Operating System from micro SD card. The Linux OS allows the Raspberry Pi to do multitasking, and provide a lot of flexibility and easiness when dealing with the real-time requirement of the system. Raspberry Pi also offers 40 general purpose GPIO pins. Moreover, the new version of Raspberry Pi 3 Model B released in February 2016 provides built-in Wi-Fi, which reduces a lot of works and costs buying and installing a separate Wi-Fi module to microcontrollers [14, 15].

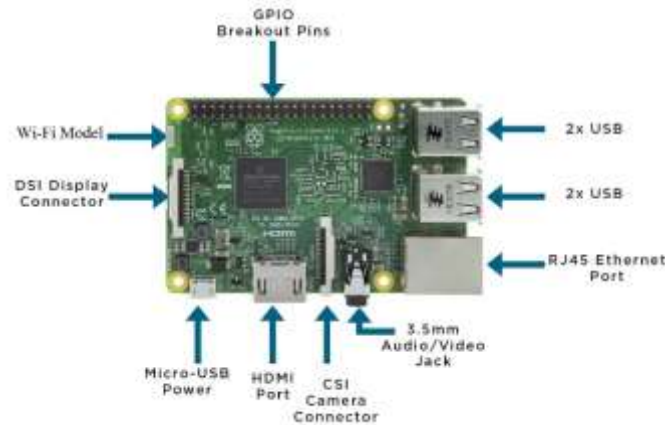


Figure 5. Raspberry Pi3 Board

1.2 The Terminal Units

The Terminal Units (TUs) are the microcontroller system that responsible for the measurements of temperature humidity, gas, flame, water level and light intensity depending on the sensors content in the node. The LM35 sensor is used for sensing the surrounding relative temperature in the bedroom field; The DHT-22 sensor is used for sensing the relative humidity and the temperature in the living room field. The LDR sensor is used for sensing the surrounding relative light intensity in the garden field; The motion sensor is used for sensing motion in the office field and the ultrasonic sensor used for detecting the water level in the tank.

The controller of the nodes is a Wemos-d1 board which is responsible for gathering and processing sensors data and sending the obtained information to BSU. Figure 6 shows the internal structure of a prototype TUs that used for implementing the system.



Figure 6. The Internal Structure of the System TUs

The TU (Wemos-d1 board) is configured by the Arduino (IDE) to be the coordinator of end devices. By supporting the Arduino (IDE) which supported by the C or C++, the ESP8266 microcontroller that content in the Wemos-d1 board is programmed for supporting the proposed protocol of the system.

A) *Wemos-d1 Board*

Wemos-d1 board is high-performance, open-source microcontroller boards based on flexible, easy to employ software and hardware. The language is C. Wemos-D1 board is built with an ESP8266 microcontroller. It consists of 14 input/output digital pins, 1 analog

input, 1K Bytes EEPROM, 2K bytes SRAM, 32K bytes ISP flash memory, 80 MHz crystal oscillator, serial communication UART represented in pin 0 (RX) and pin 1 (TX), SPI serial port defined in GPIO10 which represents select slave (SS) GPIO, GPIO11 which is the master out slave in (MOSI) GPIO, GPIO12 specified to the master in slave out (MISO) GPIO and GPIO13 is the serial clock (SCK) GPIO. The Wemos-d1 board can be powered through the USB connector, AC to DC adapter or using the battery [16][17].

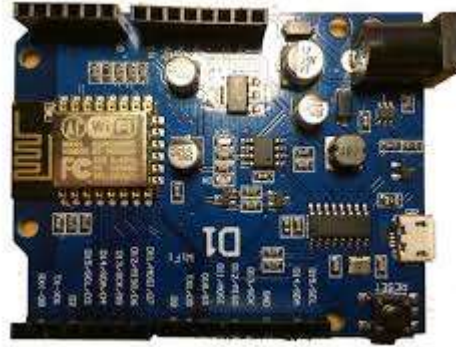


Figure 7. Wemose-D1 Board

Also, the Wemose-d1 is a great platform for any smart home system. Can create an MQTT (Message Queue Telemetry Transportation) communication, control outputs, read inputs and interrupts.

1) The First TU (the Bedroom)

The first TU is responsible for the measurements of temperature and control (turn on/off) of light, Air condition and control (open/close) door and window. The LM35 sensor is used for sensing the temperature. Figure 8 shows the schematic diagram of the first TU.

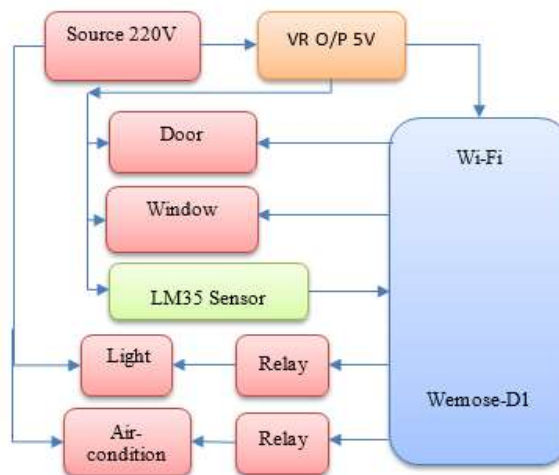


Figure 8. The System First Node Schematic Diagram

2) The Second TU (the hall, garden and water tank)

The second TU is responsible for the measurements of the Water level of the tank, the intensity of the garden light. Control (turn on/off) of garden light, water pump, hall light.

Control (open/close) hall door. The ultrasonic sensor is used for sensing the Water level of the tank. The LDR sensor is used for sensing the intensity of the garden light.

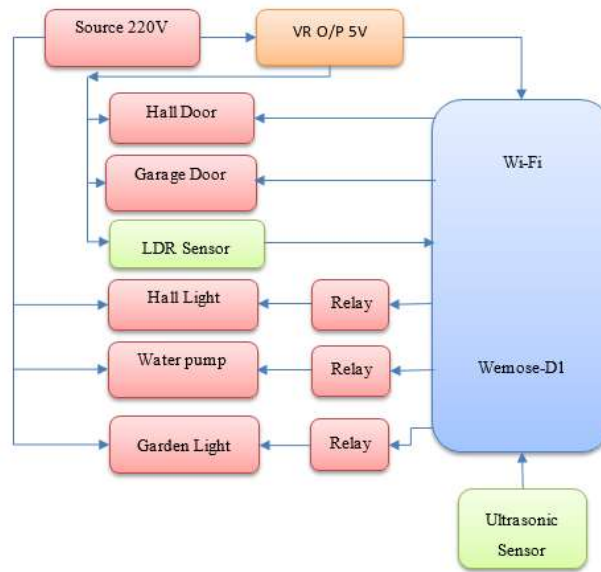


Figure 9. The System Second Node Schematic Diagram

3) The Third TU (the living room).

The third TU is responsible for the measurements of temperature, humidity, control (turn on/off) of light, fan, control (open/close) door and window. The DHT11 sensor is used for sensing the temperature and humidity. Figure 10 shows the schematic diagram of a system node.

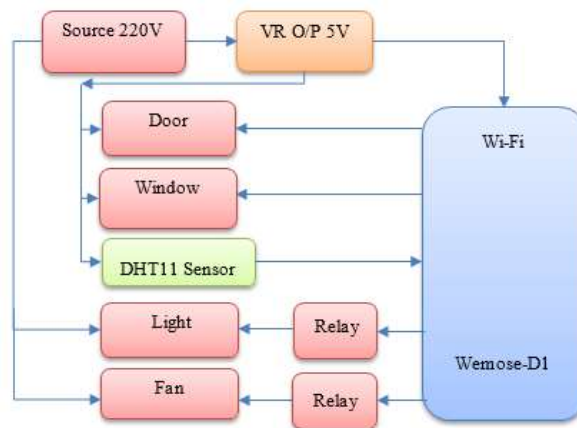


Figure 10. The System Third Node Schematic Diagram

4) The forth TU (the kitchen)

The fourth TU responsible for the measurements of gas and flame, Control (turn on/off) of light, refrigerator, Control (open/close) kitchen doors and windows. Figure 11, shows the schematic diagram of a system node.

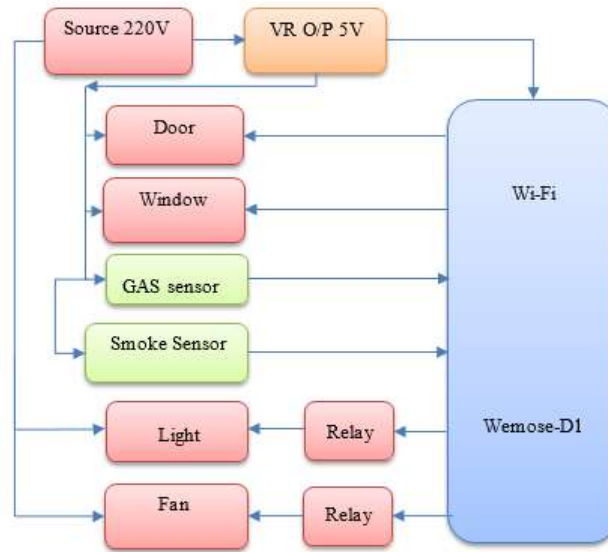


Figure 11. The System Forth Node Schematic Diagram

5) The fifth TU (the office)

The fifth TU responsible for the measurements of office motion, Control (turn on/off) of light, computer, Control (open/close) office doors. Figure 12 shows the schematic diagram of a system node.

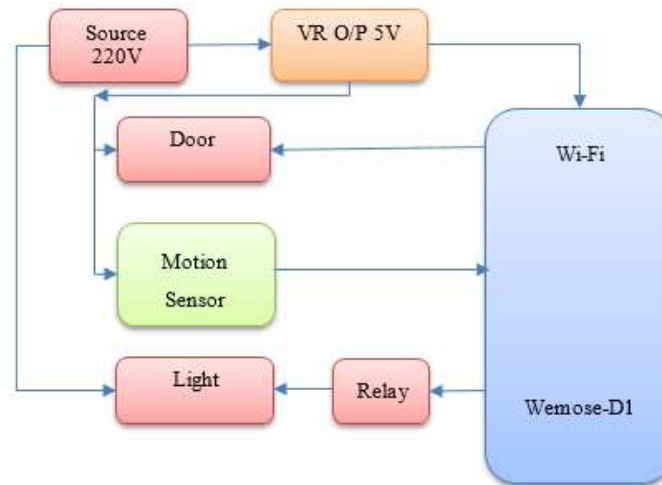


Figure 12. The System Fifth Node Schematic Diagram

6. Hardware Implementation

The prototype design aims to assemble all of the components of the smart house, including servo motors, Raspberry Pi3, Wemose-d1 board, lights, and sensors. The prototype design consists of the bedroom, living room, kitchen, office, hall, garden, water tank, and car garage. In this prototype, nine servo motors were used, which are responsible for closing and opening the doors and windows. Servo motor 1 was installed on the door of the bedroom to open and close the door, whereas servo motor 2 was installed on the window of the bedroom to open and close the window, as shown in Figure 13 Below.

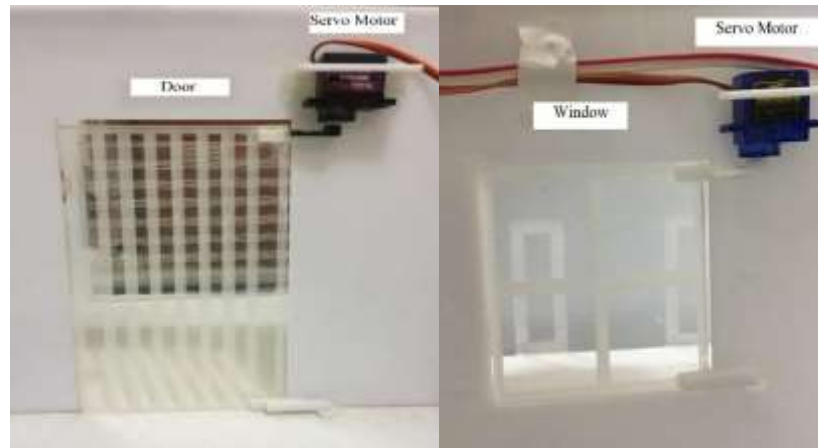


Figure 13. The Process of Installing the Servo Motor on Door and Window

The whole system was assembled in the smart house prototype, which was designed to accommodate all of the hardware components. The final design of the smart house system is illustrated in Figure 14.

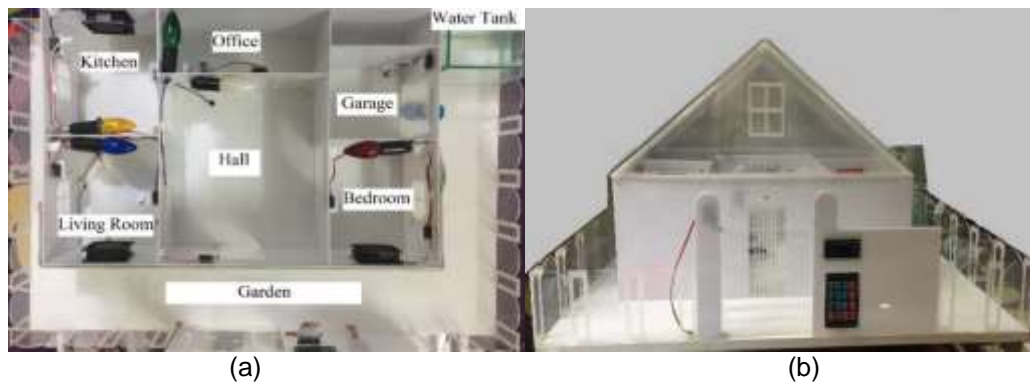


Figure 14. (a) and (b) Show the Final Design of the Smart House Prototype

7. System Software

The software of the SCADA designed systems in this part consists of three parts, IGSS SCADA software, Python program in BSU (Raspberry Pi3) which acts as a server for SCADA software and Arduino IDE software for Wemose-d1 board.

1) Interactive SCADA System

The Interactive IGSS software licensing is based on objects. The objects are of different types (digital, analogic, counter, table) and each type of object has a preconfigured number of atoms. As it is known, the atoms are of different types (*e.g.*, alarm, in/out/local, data type) and some of them may be used in defining an object only with specific templates.

Usually (classical implementation), each IGSS object has an associated physical object in the location (*e.g.*, mixer, ventilator, blower, pump, sensor, *etc.*) or a part of a physical object. Each IGSS object contains several atoms that can be associated to an I/O variable or can be locally defined. Mainly, according to the classical perspective, the main atom of the IGSS object (*e.g.*, State Value – digital object, Actual Value – analog object) incorporates the most significant value of the physical object (*e.g.*, pressure – actual values, level, flow, pump/mixer/blower state of functioning, *etc.*). The alarming and reporting strategies are defined around these main atoms.

Some atoms are used for alarming in the classical IGSS perspective (e.g., Low Alarm, High Alarm, Low Limit, High Limit, etc.). In order to use the entire capability of the alarming module in the higher level SCADA context, optimizations are necessary because the software codes are already implemented in the local SCADA or PLCs systems (e.g., the different types of alarms are usually coming directly, as various variable types; in order to utilize the Alarming type atoms it is mandatory to define template structures and to particularize them in order to access more bits in a tag made available by the automation of the local systems). The free value type atoms are foreseen to map analogic values. Considering classical implementations these atoms are somehow in the shadow of the main atom, providing a value of smaller interest that is harder to manipulate afterwards. It is stated in the classical perspective of IGSS that a free value cannot trigger alarms. Considering the above- mentioned information, if a physical object has 3 possible alarm states (e.g., overheating, overcurrent, leakage) that are represented by 3 variables in the OPC server, the classical IGSS perspective will consider 3 IGSS objects in order to trigger the 3 alarms. The values from the free value type atoms are not designed to be represented in the standard reports of IGSS. The standard reports with all the corresponding filtering options are designed for the main atoms.

Considering a higher-level SCADA perspective and the classical implementation manner, the resources can be excessively consumed leading to larger license packages (implicitly higher costs). The following chapters are presenting solutions to optimize implementations and to considerably reduce the object necessities. Figure 15 shows the IGSS SCADA program.

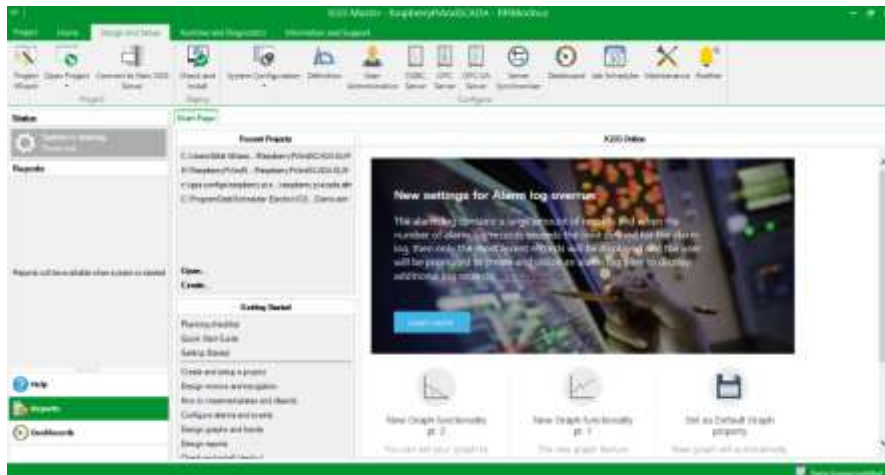


Figure 15. The IGSS SCADA Program

8. The SCADA System Protocol

The system protocol which connects the SCADA software with BSU can be described as follows: SCADA software sends query to BSU to get current sensor readings or sending command to control an appliance, BSU responds to SCADA query or command via the Python server program, where if it is a query then the server sends all of the sensor readings to SCADA which in turn update sensor values in its window. While if it is a command, then the server responds by sending this command to the concerned TU. Figure 16 shows a flowchart of the proposed SCADA system.

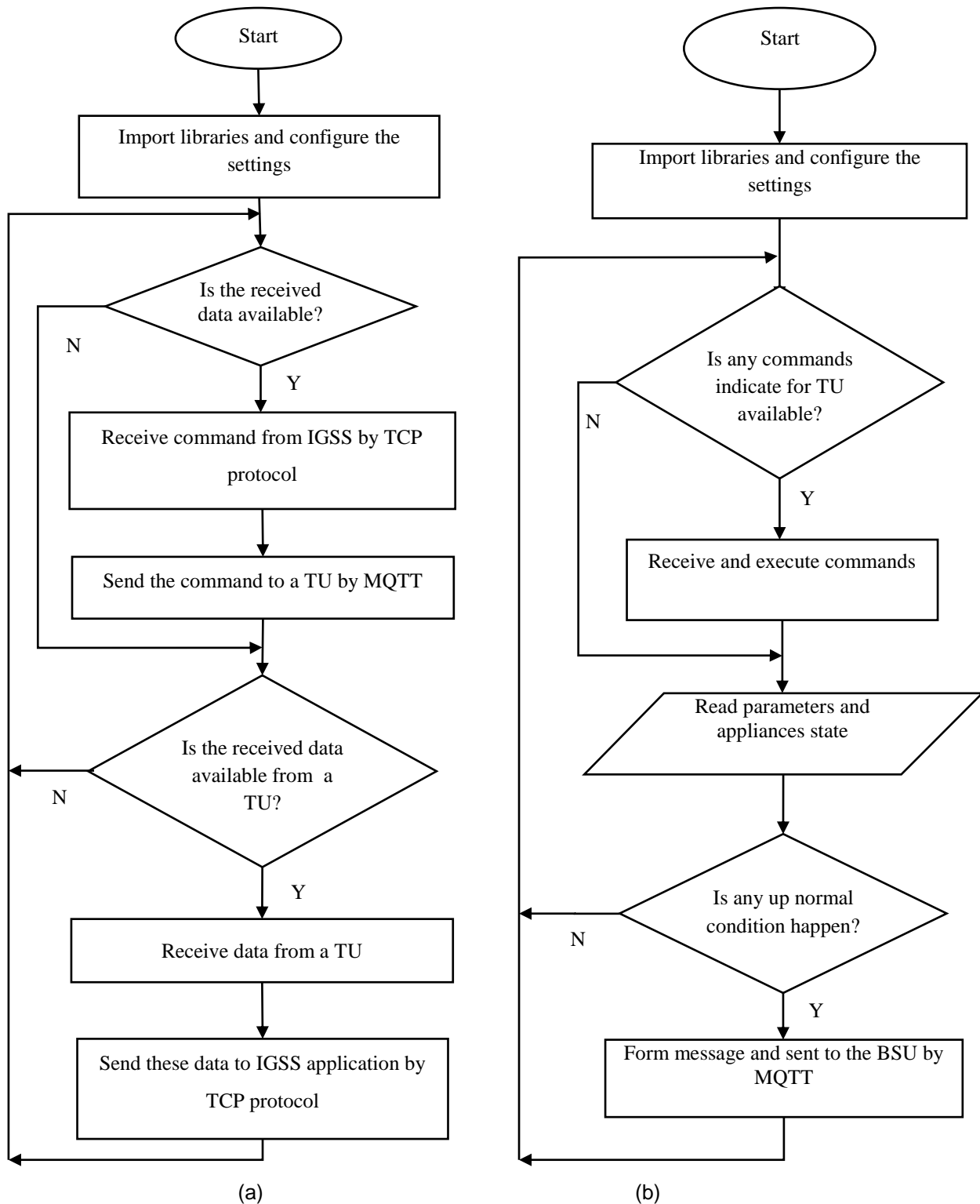
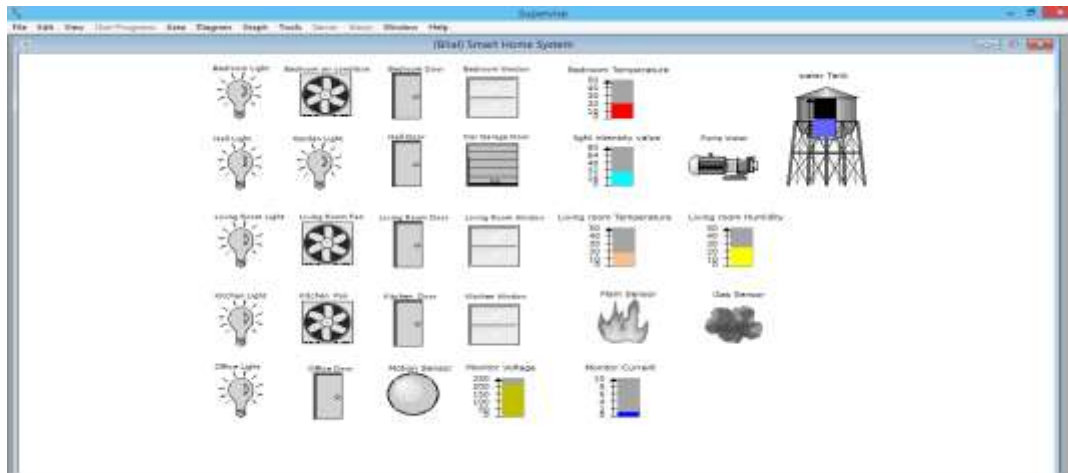


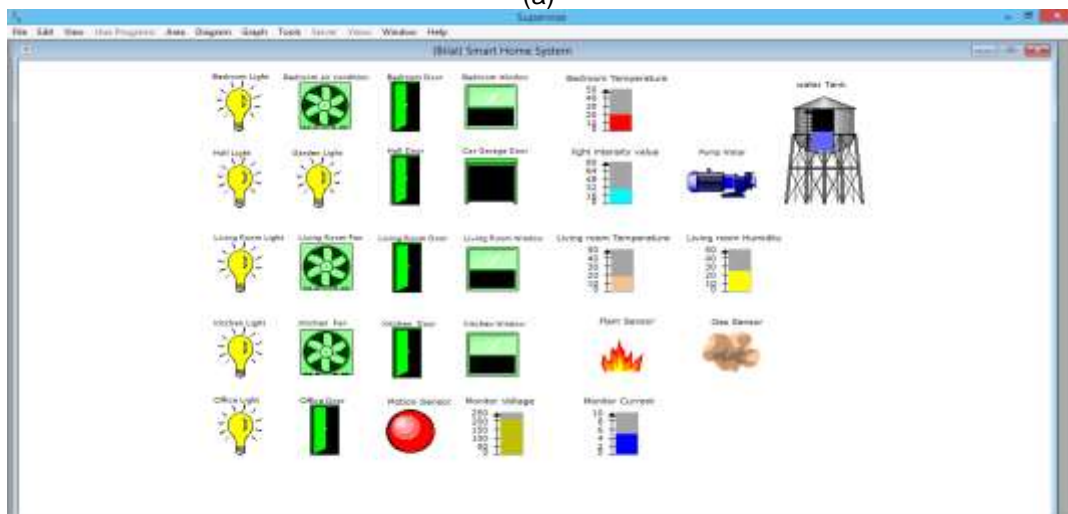
Figure 16. Flowchart of SCADA Technology, (a) BSU, (b) TU

9. The SCADA System Testing

The proposed SCADA system has been applied on the prototype home. Figure 17 (a) shows the window with all appliances are turned off while Figure 17 (b) shows the same window all the appliances have been turned on.



(a)



(b)

Figure 17. Monitoring and Control of Home Appliances over SCADA Application, (a) The Window with all Appliances are Turned Off and (b) The Window with all Appliances are Turned On

10. Conclusion

The proposed work has developed a smart home system using SCADA, Raspberry Pi3 and Wemose-D1 board. This is a cost-effective, low maintenance and user friendly automation system to help the elderly and differently abled people. Main motive of this paper is to have a centralized control system to operate the appliances through a IGSS SCADA. It also saves the time for operating appliances to the consumers. In conventional switches needs more wires for connecting load. But in proposed system, the need for wire is drastically reduced. The appliances in the home can be controlled and its status (ON or OFF condition) can be monitored from a remote location. The idea proposed in this article can also be extended for the smart of industries, malls and hospitals.

References

- [1] K. Gill, "Enhancing the Security of Wireless Sensor Network based Home Automation Systems", Doctoral Thesis, Loughborough University, London, (2009) October.
- [2] F. Beştepe, "Microcontroller-Based Multiport Communication System for digital electricity meters", M.Sc. Thesis in Electrical and Electronic Engineering in the Middle East Technical University, (2004) December.

- [3] M. Wilson, "An Online Environmental Approach to Service Interaction Management in Home Automation", PhD Thesis in Computer Science, University of Stirling, (2005) September.
- [4] Z. Lin and L. Fu, "Multi-user Preference Model and Service Provision in a Smart Home Environment", Automation Science and Engineering, IEEE International Conference, iss. 22-25, (2007) September, pp. 759-764.
- [5] M. LeMay, "Dependable Emergency-Response Networking Based on Retaskable Network Infrastructures", M.Sc. Thesis in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, (2008).
- [6] A. Szabó, "Smart House", Diploma Thesis in Computer Science and Engineering, Czech Technical University in Prague Faculty of Electrical Engineering, (2008) May.
- [7] M. Hasaj, "Smart Home – Opportunity to Make Life Easier", Msc. Thesis in Computer Science, University of Catalonia, (2008) September 3.
- [8] M. Sikandar, H. Khiyal, A. Khan and E. Shehzadi, "Sms Based Wireless Home Appliance Control System (Hacs) For Automating Appliances and Security", IEEE Technology, vol. 6, (2009) January.
- [9] D. Han and J.-H. Lim, "Smart home energy management system using IEEE 802.15. 4 and zigbee", IEEE Transactions on Consumer Electronics, vol. 56, no. 3, (2010), pp. 1403-1410.
- [10] J. Byun, "An intelligent self-adjusting sensor for smart home services based on ZigBee communications", IEEE Transactions on Consumer Electronics, vol. 58, no. 3, (2012), pp. 794-802.
- [11] A. Ahmim, T. Le, E. Ososanya and S. Haghani, "Design and Implementation of a Home Automation System for Smart Grid Applications", IEEE, (2016).
- [12] G. Barbon, M. Margolis, F. Palumbo, F. Raimondi and N. Weldin, "Taking Arduino to the Internet of Things: The ASIP programming Model", Elsevier, (2016).
- [13] A. A. Bakar and H. Hashim, "Implementation of SCADA System for DC Motor Control", International Conference on Computer and Communication Engineering (ICCCE 2010), (2010), pp. 11-13.
- [14] Raspberry Pi (Trading) Ltd, "Datasheet of Raspberry Pi Compute Module CM1, CM2 and CM3L", Version 1.0, (2016) October.
- [15] A. Wiley Brand, "Raspberry Pi for Dummies", John Wiley and Sons, New Jersey, (2013).
- [16] Espressif, "ESP8266EX Datasheet", Version 5.2, (2016).
- [17] R. K. Kodali and K. S. Mahesh, "A Low Cost Implementation of MQTT Using ESP8266", IEEE, 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), (2016) December, pp. 404-408.

Authors



Bilal Naji Alhasnawi received his B.S. in Electrical Engineering from Kufa University, Najaf, Iraq, in 2015. He is currently an M.S. candidate at Basrah University. His current research interests include Smart Home systems.



Dr Basil H. Jasim received his B.S. in electrical engineering from Basrah University, Basrah, Iraq, in 1996, and his M.Sc. and Ph.D. degrees in electrical and computer engineering from Basrah University, in 2000 and 2009. He is currently an Assistant Professor in the School of Electrical and Electronics Engineering, Basrah University, Basrah, Iraq. From 2000 to 2009, he was a researcher with Iraq Telecom (IT), Basrah, Iraq, where he worked on designing and implementing home network systems. His research interests include power system state estimation, electricity markets and security of smart grids.