UDP Pervasive Protocol Implementation for Smart Home Environment on MyRIO using LabVIEW

Wijaya Kurniawan, Mochammad Hannats Hanafi Ichsan, Sabriansyah Rizqika Akbar
Computer Engineering, Faculty of Computer Science, Brawijaya University, Indonesia

Article Info

Article history:

Received Feb 1, 2017
Revised Jul 13, 2017
Accepted Jul 27, 2017

Keyword:

LabView
MyRIO
Pervasive
Smart home
UDP

ABSTRACT

Smart Home concept is to connect several devices to perform control and the device to be controlled to help any human needs. So it should have one device being Host and several devices being Client. The environment in this case is a home that there is too much technology that planted technologies that has much influence at human activities is called pervasive computing. The device communication should be has a protocol, at the research used UDP protocol because it doesn't need to negotiate before performing communication like a TCP. With UDP behavior did not use to validate at every send or deliver data because at Smart Home environment doesn't need a large or big data being processed. LabView used at this research based on previous research, but MyRIO has used this research to perform Client and Host performed by PC. So it can measure how usable the design that created. This research proves that LabView and MyRIO successfully implemented with the design that created to perform Pervasive Computing for Smart Home environment. Functional testing scenario conducted by every condition that on system environment, based on design all testing scenario successfully and working as well as expected.

Corresponding Author:

Wijaya Kurniawan,
Computer Engineering,
Faculty of Computer Science,
Brawijaya University.
Jl. Veteran no 5, Ketawanggede, Kec. Lowokwaru, Malang, Jawa Timur, Indonesia.
Email: wjaykurnia@ub.ac.id

1. INTRODUCTION

The internet utilization not only in devices such as computers or mobile phones, but in nearly every electronic device will require the use of the internet. This requirement aims to monitor and control for use of the device. One of six technologies will continue to evolve is Internet of Things (IoT), which will continue to evolve until 2025 [1]. On IoT Concept of electronics devices that locate in Smart Home concept is assumed by internet nodes, where each device expected to be connected with internet or different device inside of different object [2]. So that each device has a sensor, actuator, operating system, input/output control, internet connection and has a programming language [3]. A concept of smart home is to connect the device to perform control and the device to be controlled. But this technology must have the characteristics of cost, they do not have cable (all devices connected to wireless/radio frequency), it can evolve as we get the device, and can identify all devices that have varying services (ubiquitous) [4].

Pervasive computing is an environment where there are planted technologies that on certain environment object and that technology has too much influence of human activities [5]. Pervasive introduced much area for usability method, mobility usage and computation size of pervasive device [6]. So that the existence of these technologies is not felt and do not interference with human activities [7]. Pervasive technology would be very suitable to implement the smart home environment, where the home environment there are devices that are used daily, but their operation still has many limitations [8]. Some examples suppose the television and air conditioner (AC), both have a device like a remote control. However remote possessed both have different functions, in terms of functions and usefulness. And the existence of remote proficiency level is not always the human's grasp. So pervasive technology very suitable for a ubiquitous environment and the implementation on smart home where all devices have four models of which include device connectivity to the cloud, the device to the gateway, device to device and the back end of the data sharing [9]. In this study focused only on the model of device connectivity to the gateway only, so this study will show the relationship between the Host / Gateway with Device / Client owned by smart home devices.

On connectivity to the gateway device required a communication protocol, a communication protocol in the Application Layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). The use of TCP is the Client will perform data transmission to the host, then the client will ensure full data on the Host of the data sent or not [10]. On UDP, the client sends the data to the host, but the Client and Host will negotiate when before performing communication. TCP will be very efficient when the data is sent in large amounts, but in smart home environment is not required large amounts of data [11]. With UDP behavior that did not validate whether the data sent by the client has come to host and vice versa will make the process faster data transfers. UDP provides a feature broadcast packets to a local network or multicast to all subscribers [12].

LabVIEW is a graphical programming language that uses icons instead of text to create applications [13]. In contrast to text-based programming languages, where instructions determine program execution, Labview uses dataflow programming, where the flow of data determines execution [14]. LabView can easily connect other devices such as microcontroller, Arduino, IP camera and even computer networks [15]. Additionally, National Instruments has also developed highly compatible with the hardware such as NI LabVIEW MyRIO, NI CompactRIO and NI ELVIS is easier to develop a system [16]. NI myRIO is an embedded hardware that can be used for technology proven industry researchers and enable them to design systems with real, complex engineering systems faster and more affordable than ever before [18]. So in this research the implementation of Pervasive Protocol UDP will be used LabVIEW as a programming language by utilizing MyRIO as client devices with the PC as a Gateway.

Another study conducted using Salutation Discovery Protocol and Service Location Protocol [18], and implementing HTTP-CoAP at LabVIEW [19]. At our previous research, it is introduced how UDP work in two PC, one as Client and one else as Gateway [20]. UDP and Pervasive protocol were designed by the state machine and work well implemented with LabVIEW. In this research we will implement the Gateway in a PC and Client modeling with MyRIO. So the design of the system can be tested functionally to obtain correspondence between the designs of a system that was developed with the results obtained.

2. RESEARCH METHOD

In this section explained design of host and server network, Gateway – Client Design and Test design. The domain in this research is how Gateway / Host can communicate with Client at Local Area Network. So that it can be made the design for system testing and the results obtained on the availability of the system between a Host and Client. Tests will be performed to test the functionality of the system, not only to do with the fruit Client but the test will be performed up to three clients.

2.1. Host and Server Network

This research design conducted by ETSI standard [21] which systems have Machine to Machine (M2M) area network domain and implemented in Local Area Network. So design at pervasive using UDP shown in Figure 1 which has a Host and Client. Host / Gateway will be implemented on PC with LabVIEW interfaces, and Client will be implemented on MyRIO which has many features like analog input/output, wifi, led, push button, accelerometer etc.

Where the initial conditions, Host are in condition to wait Client listen to broadcast. Next after the Client to broadcast and broadcast packets are received by Host Client to introduce himself to the host to provide such review client name recognition, IP and services on the host. In Check Duplication Host Client is not directly accept the client, but to review the duplication check, whether the client already exists or not, even if the client already on other hosts. When there is no duplication Client, Host then sends ack, hostname, IP and host header will be stored by the Client.

Once the Client recognizes the host and the process are completed, Host will be back on the broadcast Client Listen to wait for another, but for the Client that have been identified previously will be done the next process is to communicate. As for the service that can be provided is a Client Services (Sensor, Input / Output etc) or the host can take control of all the features possessed by the Client. Design the way work is designed in this study Host and Client will be described in section Host and Client Design

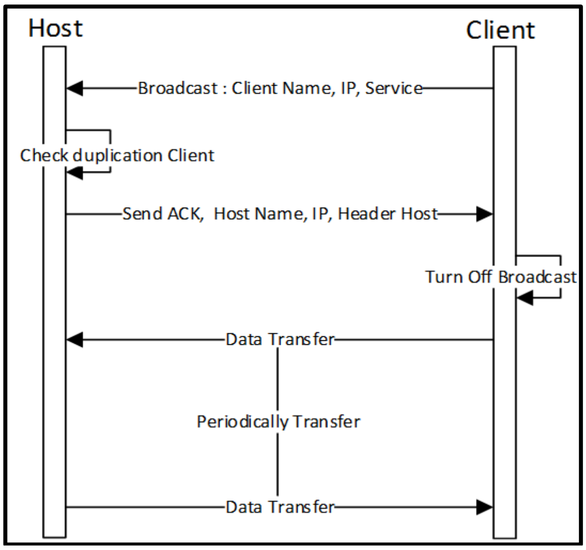


Figure 1. Pervasive Algorithm Machine to Machine Area Network [21]

2.2. Gateway / Host and Client Design

Based on previous research [20], the Gateway / Host and Client has designed but at implementation, they're implemented on PC and the Client will be implemented in MyRIO. The Host and Client process was designed in state machine diagram, as seen in Figure 2 and Figure 3 the design was conducted by Figure 1, where Host and Client don't have a connection until they well communicate.

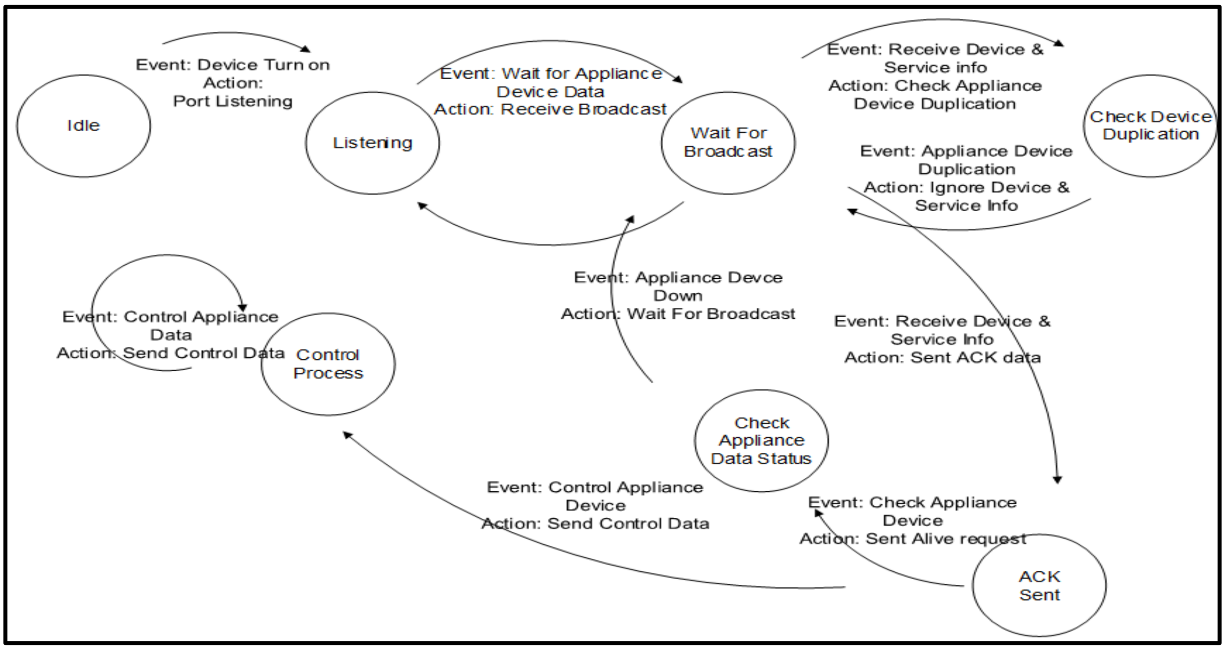


Figure 2. Host State Machine Diagram [20]

Each condition that designed on state machine diagram in Figure 2 for Host will explain as follows:

a. Initialize: in this condition shown by Idle. At this state, the host performs open UDP Port that given 6300 for Listener ID as a broadcast receiver. The second port is 6400 that given for Client ID, Client ID default is set by "0" it will increase by "1" increment if Host receives broadcast from Client. That id is used to communicate with Client to perform Send ACK and do Check Appliance Data Status to perform checking Client connection. Open UDP Port 6500, 6600 and 6700 for accelerometer each for x, y and z.

This state has an error handler to stop the process and show that Host doesn't have Client or otherwise. If Hostname created it will move to Listen to State.

b. Listen: in this condition perform Listen and Wait For Broadcast, this state has Stop and Go HW button, if that not press then UDP Port Listener ID always listen loop every 250ms then perform checking are there a client perform broadcast or not. If Host receives broadcast then move to Check HW State, if not Host still at Listen Condition. If Go HW pressed, then state move to Send Req State.
c. Check HW: is represented by Check Device Duplication, this state performs if Client recognized by Host. List of Client ID saved as an array which is their element known as Index identity. This state for checking is there any data duplication which has Client Service, IP HW and Service Number. If detected any duplication data then old data deleted then new data inserted into an array with a new index number. Else there is no duplication then directly insert with new Client ID.
d. Send ACK: perform ACK Sent and Check Appliance Data Status. This state which host would send ACK to Client that perform HW Check. ACK Send through UDP Port which has Client ID who has identity Host Name and IP Host. After this state moves to Listen State otherwise Host move to Send Req State to do Controlling device to Client.
e. Send Req: in this state represent by Control Process at state diagram. The host can receive and send data to Client. The host can Controlling Client by choosing which client will be controlled in an array which described at Check HW state. The host can control using opened UDP Port, 6500, 6600 and 6700 for accelerometer x, y, z and other. At this state, Host has performed checking Client connection. If connection lost then Host move to Listen State and show a message that Client is lost. If Client still has a connection, communication still continue.
f. Stop State: This state was not designed in the state machine, because an assumption Host never shutting down. But its uses to perform closing port which opens at Initialize State and clear array which has Client ID, because Client communicates with IP address after receiving ACK.

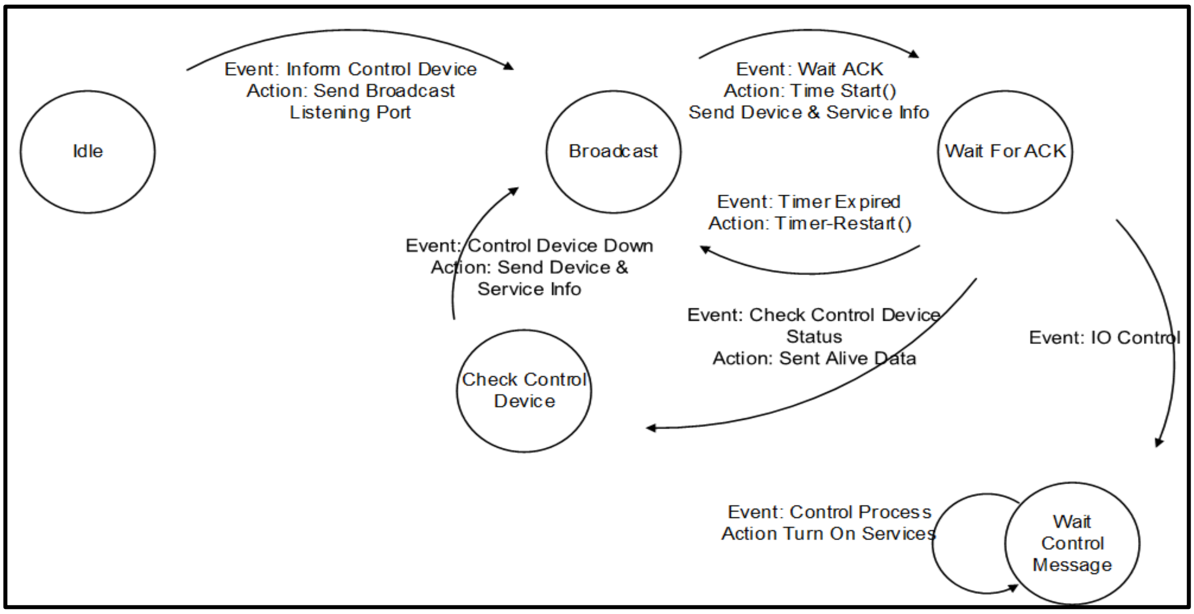


Figure 3. Client State Machine Diagram [20]

Each condition that designed on state machine diagram in Figure 3for Host will explain as follows:

a. Initialize: represent the Idle state, this state open four unused UDP port as a Broadcast ID. Broadcast ID used to communicate broadcast and check did Client communication with Host still exist or not. It has three Service ID at Broadcast ID, 1st Service ID used to send identity Client Service and receive a request from Host. 2nd Service ID used to send Boolean or any other data, and 3rd Service ID used to send sensor data. It has been made too error handler to handling error program and show which service name that has not been filled. If Service ID filled up then it moves to Broadcast State.
b. Broadcast: at this state, Client sends broadcast data to move through UDP Port Broadcast ID for 250ms that contain Identity Client (Service name, IP Client and Service Number). After broadcast message

receives by Host, then Host send ACK to Client and Client move to ACK state after receiving ACK from Host.

c. ACK: this state represented Wait For ACK.

d. Req: two state at Figure 3 at this state Wait for Control Message and Check Control Device. Wait Control Message is client waiting is that any request from host received by UDP Port through Service ID 1. If there is a request, then Client will send data through UDP Port Service ID 3. At the moment Service ID 2 will send Boolean data or other data to host. While Service ID 1 used to receive data from Host that contain data to turn on led Client. Check Control Device used to check any Client that disconnect or not. If there is no disconnect device, Client will continue to communicate with the host. But if Client disconnects with Host, then Client going back to broadcast state.

2.3. Functional Testing Scenario

To perform the test on whether the system is working properly or not, will be conducted a test of what has been done. In this system, the design of the system has to offer is made by a state machine, so the testing is done with state transition testing. State transition testing is used when the system designed and states defined by a finite number [22]. The state transition testing test is the valid testing to test system transition from one state to another and also test the valid transition based on system design [23]. Every state has an event and action that perform different condition. Every states and action based on a design from fig and fig will be conducted on functional testing result and analysis.

Table 1. Gateway Testing Scenario

T.ID	Gateway Initial State	Test Scenario and Description	Destination State
1	Idle	Turn on the device	Listening
2	Listening	Open Port	Wait For Broadcast
3.1	Wait For Broadcast	Host turn on first then turn on Client conducted by 1 Client	Check Device Duplication
3.2		Host turn on first then turn on Client conducted by n-Client	Check Device Duplication
3.3		Turn on Client then turn on Host Conducted by 1-Client	Check Device Duplication
3.4		Turn on Client then turn on Host Conducted by n-Client	Check Device Duplication
4.1	Check Device Duplication	Checking device duplication conducted by 1 Client with duplication occur	Wait for Broadcast
4.2		Checking device duplication conducted by more than 1 client (n-client) with duplication occur	Wait for Broadcast
4.3		Checking device duplication conducted by 1 Client without duplication	ACK Sent
4.4		Checking device duplication conducted by more than 1 client (n-client) without duplication occur	ACK Sent
5.1	ACK Sent	Sending ACK to 1-Client then connection lost	Check Appliance Data Status
5.2		Sending ACK to n-Client then connection lost	Check Appliance Data Status
5.3		Sending ACK to 1-Client without lost connection	Control Process
5.4		Sending ACK to n-Client without lost connection	Control Process
6.1	Check Appliance Data Status	Checking appliance data status to 1 Client	Wait for Broadcast
6.2		Checking appliance data status to n-Client	Wait for Broadcast
7.1	Control Process	Control all data by 1 client	Control Process
7.2		Control all data by n-client	Control Process
7.3		Turn off then turn on Host with 1 Client	Idle
7.4		Turn off then turn on Host with n-Client	Idle
7.5		Turn off then turn on 1 Client	Check Appliance Data Status
7.6		Turn off then turn on n-Client	Check Appliance Data Status

Every Host state machine diagram at Figure 2 test conducted by Table 1, every state representing by a header number at the table and every sub number representing any possibilities that could happen at that state. For example “1” it representing first state that namely Idle, then another example “3.1”,”3.2”,etc “x.1,”x.2”; “x” or “3” is representing the name of the state then “3.y”; “y” represent possibilities that happen in “x” state. By that definition, is used to give another state in Client from [20] Figure 3 to conduct Table 2. N-Client test scenario would carried out with 2 and 3 clients.

Client Initial State defining a starting testing state, then Test Scenario and Description given in order how to test that state, which gives an overview of how testing will be done. When we did the tests specified, we also determine the state of interest represented by Destination State. So in the functional testing scenario it will determine an expected result and actual result that generated by the system. So from this test can be measured as the level of conformity with the system design objectives.

Table 2. Client Testing Scenario

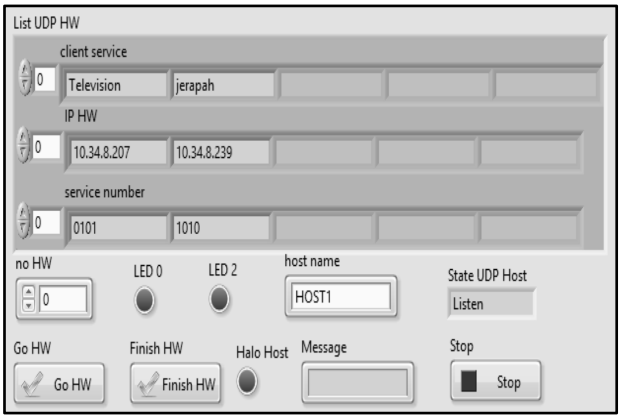
T.ID	Client Initial State	Test Scenario and Description	Destination State
8	Idle	Turn on client, opening Port for broadcast	Broadcast
9	Broadcast	Broadcast data until Host receive broadcast data then send ACK	Wait for ACK
10.1	Wait for ACK	Received ACK that delivered by Host	Wait for Control Message
10.2		ACK didn't receive until timeout	Broadcast
10.3		ACK Received but no control from Host	Check Control Device
11	Check Control Device	Check own service, that service still available or not, any possibilities go to broadcast state	Broadcast
12	Wait for Control Message	Device controlled by Host, downtime and controlling data conducted by the gateway.	Wait for Control Message

3. RESULTS AND ANALYSIS

In this section will be presented on how the system has been built. The initial Chapter 3.1. section will be used to indicate that the system has been made according to the design that was created earlier. Then in the second part Chapter 3.2. will be tested in accordance with the functional testing with the points that have been formulated previously at Table 2 and Table 3.

3.1. Gateway / Host and Client Implementation Result

After doing the design for the system along with its testing, the system implemented on LabVIEW. The system design for the host to be run on a PC while the Client can be run on a PC which can be uploaded and executed on MyRIO. Features provided on a customized system with a design created in the previous design, so it can be tested with scenarios designed to accommodate all the possibilities that occur in the system environment.



List UDP HW

client service: 0 | Television | jerapah

IP HW: 0 | 10.34.8.207 | 10.34.8.239

service number: 0 | 0101 | 1010

no HW: 0

host name: HOST1

State UDP Host: Listen

Figure 4. Gateway / Host Front Panel

Features created on the host represent at Figure 4, provided a place to accommodate the Client along with its services on the Array List is represented by HW UDP. No HW Client is provided to select a number which will be given control. Led 0, 1, 2 and 3 are used to display and provide control of service owned by the Client, in Fig mentioned only LED 0 and 2 only that describe a service owned by the client only 0 and 2. The Host Name is used to provide/initialize Hostname.

State UDP Host was used to identify which Host State at a time. Go and Finish HW used to initiate or carry out control on the client, in this case is MyRIO. While Halo Host and Message is a function assigned to if the client wants to do communications and to check the host, whether the host is connected or not. In this case applied to the State Check Control Device which is owned by the Client. The stop is used to stop the whole series of programs.

In Figure 5, provided one example of creating a state that represents the design that was created earlier. In the picture shown one example of a State that is Send Req. On State Send Req made a lot of possibilities to accommodate the problems that will occur in the system environment. It is represented by the Sub-State by conditioning the Error / No Error and False / True that as in the conventional coding using the If / Else or Switch Case.

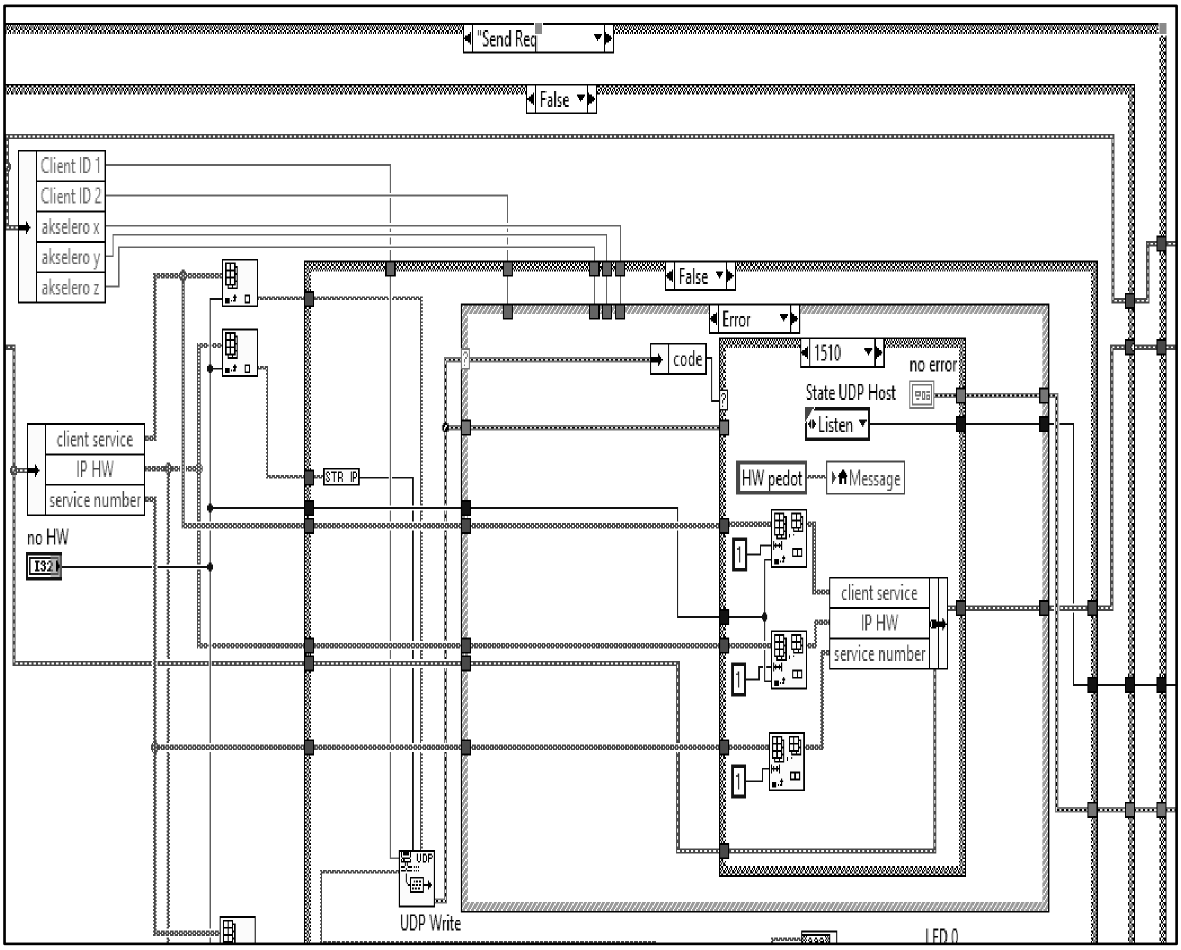


Figure 5. Gateway / Host Block Diagram

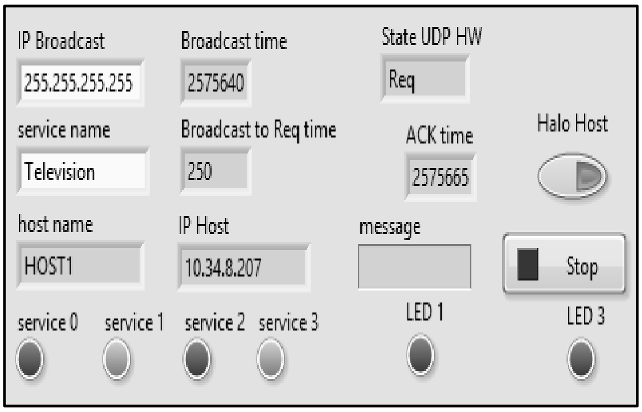

IP Broadcast 255.255.255.255
Broadcast time 2575640
State UDP HW Req
service name Television
Broadcast to Req time 250
ACK time 2575665
host name HOST1
IP Host 10.34.8.207
message

Figure 6. Client Front Panel (Will Be Uploaded into MyRIO)

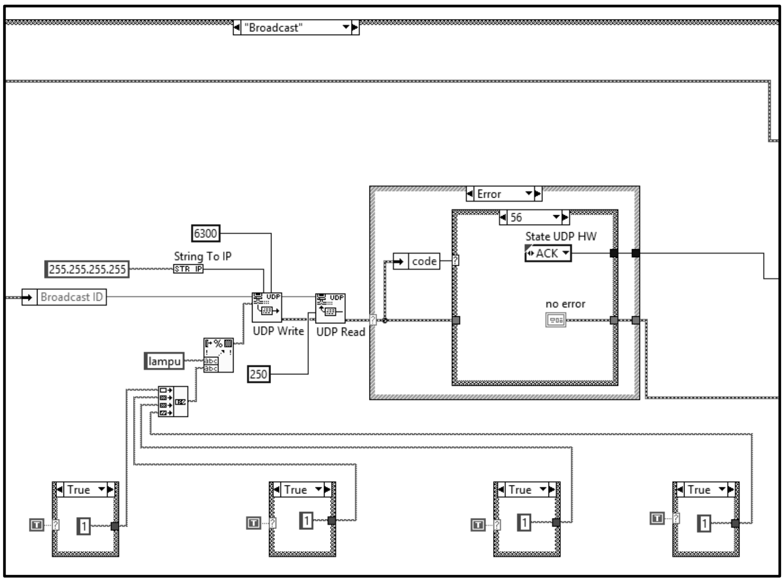


Figure 7. Client Block Diagram (Will Be Uploaded into MyRIO)

After creating code for Host / Gateway also made the front panel for the client. A front panel like on Figure 6 and block diagram in Figure 7. MyRIO does not have a front panel, but for coding the front panel must still be made to perform initial testing prior to deployment on MyRIO, Input and Output owned by MyRIO is physical not a form of software. At Figure 6 Front Panel of MyRIO, has been given several functions such as IP Broadcast to show the range of IP broadcast how he was carrying out. Then the Service Name is the name of the Client itself, for this name can be changed in accordance with the desired name or device is what is desired. Host Name function is to detect MyRIO which has been connected with Host and IP Host shows the IP owned by Host / Gateway. Broadcast Time is to indicate the length of time from Client / MyRIO to do a broadcast to find hosts and ACK Time indicates how long the exchange of ACK conducted between Client and Host.

Service 0-3 is a service that is owned by the client as indicated by LED owned, when service is turned on it will be indicated by the LED. Actually, there are four LED LED starting 0-3, but the Service is activated only 1 and 3 so that the LED that serves only 1 and 3. This can be done as desired or service that is owned by the Client. At Figure 7 shows one state from several that have been created on the block diagram, the figure shows one of the states owned by the client, namely Broadcast State.

3.2. Functional Testing Result and Analysis

After implementation and deployment system, system testing will be done according to the design in Table 1 and Table 2. In the examination table, Table 3 given multiple columns. The first column is T.ID, this column shows the same column in Table 1 and Table 2, which describes the testing of each side and state-owned. Expected Result show about the expected performance of the system, while the Actual Result is the actual results obtained from tests performed.

Having obtained the Actual Result will do the comparison given in the Status column. In the column will be given two conditions to declare the success of the system. The first condition is "Pass" indicates compatibility between Expected Result with Actual Result. The second condition is "Fail" which shows the discrepancy between the two. But in Actual result although not in accordance with Expected Result will be given the actual conditions occur although there is the possibility of a program error.

Table 3. System Testing Result

T.ID	Expected Result	Actual Result	Status
1	Gateway program compiles and runs without error.	Successfully compiling and running.	Pass
2	Port 6300, 6400, 6500, 6600, and 6700 Opened. Default Client ID is 0.	Port 6300, 6400, 6500, 6600, and 6700 Opened. And default Client ID is given by 0.	Pass
3.1	Host turn on first, listening and find 1 client.	Host successfully turned on and found 1 client.	Pass

T.ID	Expected Result	Actual Result	Status
3.2	Host turn on first, listening then find 2 and 3 clients.	Host successfully turned on and found 2 and 3 clients.	Pass
3.3	Host turn on last, listening and find 1 client.	1 client turned on first and discovered by Host that turned on last.	Pass
3.4	Host turn on last, listening then find 2 and 3 client.	2 and 3 clients turned on first and discovered by Host that turned on last.	Pass
4.1	Check and find duplication Client ID at the same client conducted by 1 client with duplication occur.	Host checking and did not find duplication, deleting old ID then given new ID for 1 client.	Pass
4.2	Check and find duplication Client ID at the same client conducted by 2 and 3 clients with duplication occur.	Host checking and did not find any duplication, deleting old ID then given new ID for 2 and 3 clients.	Pass
4.3	Check and find duplication Client ID at the same client conducted by 1 client without duplication occur.	Host found duplication, deleting old ID then given new ID for 1 client.	Pass
4.4	Check and find duplication Client ID at the same client conducted by 2 and 3 clients without duplication occur.	Host found duplication, deleting old ID then given new ID for 2 and 3 clients.	Pass
5.1	Host sends ACK to 1 client, after that disconnect network, then Host goes to Check Appliance Data Status.	Host sending ACK then delivered by 1 Client, whether it has a disconnect connection, Host went to Check Appliance Data Status State.	Pass
5.2	Host sends ACK to 2 clients, after that disconnect network, then Host goes to Check Appliance Data Status.	Host sending ACK then delivered by 2 and 3 Client, whether it has a disconnect connection, Host went to Check Appliance Data Status State.	Pass
5.3	Host sends ACK to 1 client without disconnect network, then Host goes to Control Process State.	Host sending ACK then delivered by 1 Client without disconnect connection then Host went to Control Process State.	Pass
5.4	Host sends ACK to 2 and 3 clients without disconnect network, then Host goes to Control Process State.	Host sending ACK then delivered by 1 Client without disconnect connection then Host went to Control Process State.	Pass
6.1	Checking appliance 1 Client disconnect status then Host goes to Wait For Broadcast State.	When 1 Client disconnected, Host goes back to Wait for Broadcast State.	Pass
6.2	Checking appliance 1 and 2 Client disconnect status then Host go to Wait For Broadcast State.	When 1 Client from 2 and 3 Client disconnected, Host goes back to Wait for Broadcast State.	Pass
7.1	Controlling all service that defined by 1 Client.	All services that own by 1 Client can be controlled by Host well, but after 10 until 15 seconds, error conducted by Host having error code 56.	Pass
7.2	Controlling all service that defined by 2 and 3 Client.	All services that own by 2 and 3 Client can be controlled by Host well, but after 10 until 15 seconds, error conducted by Host having error code 56.	Pass
7.3	Host turn off then turn on again and Host started again in Idle State conducted by 1 Client.	Host turned off then host went back to Idle State, 1 Client that connected before back to Broadcast State	Pass
7.4	Host turn off then turn on again and Host started again in Idle State conducted by 2 and 3 Client.	Host turned off then host went back to Idle State, 2 and 3 Client that connected before back to Broadcast State	Pass
7.5	Turn off 1 Client then turn on again, Host move to Check Appliance Data Status State.	1 The client turned off, Host back to Check Appliance Data Status State.	Pass
7.6	Turn off 2 and 3 Client then turn on again, Host move to Check Appliance Data Status State.	2 and 3 Client turned off, Host back to Check Appliance Data Status State. But the Client that did not turn off, still can be controlled by Host.	Pass
8	The client turns on, compiling and running without any error.	Successfully compiling and running.	Pass
9	Broadcasting data until finding a Host that sending ACK.	The client did a broadcast until found Host, then Host sends ACK and Client Finish broadcast.	Pass
10.1	Receive ACK then move to Wait Control Message State.	ACK Received, then move to Wait Control Message State.	Pass
10.2	The host never sending ACK until timeout.	Client stop waiting for ACK, then go back to broadcast state.	Pass
10.3	Receive ACK but Host never controls service that own by Client then move to Check Control Device State.	Client received ACK but no control message from Host, then Client move to Check Control Device state.	Pass
11	Checking all of the services that own by Client still available or not, then go to Broadcast state again to do broadcast again.	All services that own by Client checked by itself, after checked all Client back to broadcast state.	Pass
12	All service that owns by the client can be controlled by Host.	All of the services that own by Client can be controlled well by Host.	Pass

This test aims to obtain correspondence between the design was done, the implementation of the design of the system and to demonstrate the suitability of the design of the behavior and performance of LabVIEW and MyRIO. So it can be given any conclusions about the suitability of whether the system works or not. The design successfully implemented on LabView and successfully deploy on MyRIO, both can communicate well.

4. CONCLUSION

This research successfully compiling running, the design is working well as expected. The client can communicate well with host pervasively. Service discovery in this research performed by Client which can identify and start communicate with Host. After send and deliver the ACK, both can communicate well. The host can control any of service that owned and provided by Client. Host conducted by a PC and Client conducted by 3 MyRIO. System testing conducted by Design that created, the functional testing scenario was conducted by any possibilities that could happen in system environment seen from both sides between Host and Client. All functional testing scenario is Pass. But there are after several minutes communicate it performed error 56. That error conducted by random access memory buffer that owned by PC or MyRIO, but that error not conducted by design that created. For future work it can be performed again with various approaches to reduce memory running and implementing this system at any household environment.

ACKNOWLEDGEMENTS

The team thanks to the Laboratory of Computer System and Robotics, Computer Engineering Department, Faculty of Computer Science. This paper is an extension of work originally published and reported in “IAES 2016: International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2016)” at Semarang, East Java, Indonesia 22-25 November 2016, with the title “Lightweight UDP Pervasive Protocol in Smart Home Environment based on LabView”.

REFERENCES

[1] Atzori, L., Iera, A. and Morabito, G., "The Internet of Things: A survey"," *Elsevier Computer Networks,* pp. 2787-2805, 2010.

[2] Vermesan, O. and Friess, P, “Internet of Things - Global Technological and Societal Trends,” Aalborg, Denmark: River Publisher, 2011.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," *Elsevier Science,* vol. 38, no. 4, p. 393–422, 2002.

[4] Munir, S.A., Ren, B., Jiao, W., Wang, B., Xie, D. and Jian, M., *"Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing,"* in 21st International Conference on International Conference on, Advanced Information Networking and Applications Workshops, AINAW '07, 2007.

[5] Froehlich, J., Findlater, L. and Landay, J., *"The Design of Eco-Feedback Technology,"* in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Atlanta, Georgia, USA, 2010.

[6] Dourish, P. and Bell, G., “Divining A Digital Future - Mess and Mythology in Ubiquitous Computing,” London, England: The MIT Press, 2011.

[7] Ma, J., Yang, T.L., Apduhan, B.O., Huang, R., Barolli, L. and Takizawa, M., "Towards a Smart World and Ubiquitous Intelligence: A Walkthrough from Smart Things to Smart Hyperspaces and UbicKids," *International Journal of Pervasive Computing and Communications,* vol. 1, no. 1, pp. 53-68, 2005.

[8] Cook, D.J. and Das, S.K., "How smart are our environments? An updated look at the state of the art," *Pervasive and Mobile Computing,* vol. 3, no. 2, pp. 53-73, 2007.

[9] Rose, K., Eldridge, S. and Chapin, L., "The Internet of Things: An Overview Understanding the Issues and Challenges of a More Connected World," *The Internet Society (ISOC),* 2015.

[10] Ford, A., Raiciu, C., Handley, M. and Bonaventure, O., “TCP Extensions for Multipath Operation with Multiple Addresses”, RFC 6824, 2013.

[11] Andrea Zanella; Nicola Bui; Angelo Castellani; Lorenzo Vangelista; Michele Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal,* vol. 1, no. 1, pp. 22-32, 2014.

[12] Jack L. Burbank; Julia Andrusenko; Jared S. Everett; William T.M. Kasch, “Wireless Networking : Understanding Internetworking Challenges”, IEEE PRESS, 2013.

[13] M.Kalyan Chakravarthi; Nithya Venkatesan, "Experimental Validation of a Multi-Model PI Controller for a Non-Linear Hybrid System in LabVIEW," *TELKOMNIKA (Telecommunication Computing Electronics and Control),* vol. 13, no. 2, pp. 547-555, 2015.

[14] Jimenez, F.J. and Frutos, D.J., "Virtual Instrument for Measurement, Processing Data, and Visualization of Vibration Patterns of Piezoelectric Devices," *Elsevier,* vol. 27, no. 6, pp. 653-663, 2005.

[15] L. Vanfretti, V. H. Aarstrand, M. S. Almas, V. S. Perić and J. O. Gjerde, "A software development toolkit for real-time synchrophasor applications," in *IEEE Grenoble Conference*, Grenoble, 2013.

[16] Samual Ashcroft; Lindsay Watt; Jordan Laurie, "Proteus and Pixel Art - Accurate Object Placement Using an Independent Autonomous Vehicle," Queensland University of Technology, Queensland, 2015.

[17] Bhatti, N., Dhomeja, L.D. and Malkani, Y.A., *"Service Discovery Protocols in Pervasive Computing: A Review,"* in Multi-Topic Conference (INMIC), IEEE 17th International, 2014.

[18] Serna, M.A., Sreenan, J.C. and Fedor, S., "A Visual Programming Framework for Wireless Sensor Networks in Smart Home Application Singapore," in *Sensor Networks and Information Processing (ISSNIP),* Singapore, 2015.

[19] Kurniawan, W., Ichsan, M.H.H., Akbar, S.R., and Arwani, I., *"Lightweight UDP Pervasive Protocol in Smart Home Environment Based on Labview,"* in IAES International Conference on Electrical Engineering, Computer Science and Informatics (EECSI 2016), Semarang, Indonesia, 2016.

[20] J. e. a. Holler, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence," in *Elsevier*, Waltham, MA, 2014.

[21] D. Lee and M. Yannakakis, *"Principles and methods of testing finite state machines-a survey,"* in Proceedings of the IEEE, vol. 84, no. 8, pp. 1090-1123, 1996.

[22] C. d. S. Carvalho and T. Tsuchiya, *"Coverage Criteria for State Transition Testing and Model Checker-Based Test Case Generation,"* in 2014 Second International Symposium on Computing and Networking, pp. 596-598, Shizuoka, 2014.

BIOGRAPHIES OF AUTHORS

Wijaya Kurniawan received Bachelor Degree at Electrical Engineering - Brawijaya University, completed a master degree at Electrical Engineering - Brawijaya University. Presently lecturer at Computer Engineering, Faculty of Computer Science – Brawijaya University. Research interest Control System, Robotics and Embedded System.

Mochammad Hannats Hanafi Ichsan received Bachelor Degree at Informatics – Electronics Engineering Polytechnic Institute of Surabaya (EEPIS), completed a master degree at Electrical Engineering - Brawijaya University. Presently lecturer at Computer Engineering, Faculty of Computer Science – Brawijaya University. Research interest Artificial Intelligence, Human-Computer Interaction, Intelligence System And Optimization.

Sabriansyah Rizqika Akbar received Bachelor Degree at Electrical Engineering - Brawijaya University, completed master degree at Information Technology – Gajah Mada University. Presently lecturer at Computer Engineering, Faculty of Computer Science – Brawijaya University. Research interest Networking, Wireless Sensor Network, Internet of Things and Pervasive and Ubiquitous Computing.