

## Home Appliance Control with Publish Subscribe in Social Media

Sabriansyah Rizqika Akbar<sup>\*1</sup>, Eko Setiawan<sup>2</sup>, Achmad Basuki<sup>3</sup>

University Of Brawijaya

Jl. Veteran Malang, East Java, Phone: +62 0341 551611/ Fax:+62 0341 565420

\*Corresponding author, e-mail: sabrian@ub.ac.id<sup>3</sup>, ekosetiawan@ub.ac.id<sup>2</sup>, abazh@ub.ac.id<sup>3</sup>

### Abstract

Nowadays, Internet social media has enriched the way people to communicate and interact each other. Will it be possible for people to interact with their home appliances around? This paper proposes a new approach in smart home system that made possible for people to remotely interact with their appliances using social media networks. In this paper, we present a smart home prototype system that leverages Twitter's Application Program Interface (API) to remotely control home appliances over the Internet. Experiment results showed that the system immediately responds to remote commands sent over a social media account to control home appliances. The system responds the command in 3672.96 ms. Publish-subscribe method work better in mass announcement communication system. Home system could notice all householders in less than 6 s independently from number of householder. Our proposed method gives alternative solution to build reliable, fast and simple control method.

**Keywords:** Publish Subscribe, Smart Home, Social Media

### 1. Introduction

Smart home have several important utilities and services parameter to be considered such as automation home appliances control and remote access [1]. Recent developments in home automation have significantly improved living of the householders to easily interact with their home appliances to adjust temperature, lighting and other devices [2],[3]. The interaction with the home appliances can even be performed remotely over the Internet or some GSM networks. The previous approaches in home automation for controlling appliances usually use a client-server system; where a client sends a message and the server do action upon receiving the message. The server must be always on to response to the message. Short Message Service (SMS) gateway and Hypertext Transfer Protocol (HTTP) are the two common ways in sending message of client-server system [2]. Even though SMS provides a simple solution of communication ways without the Internet connection, its message type is limited to text only (no picture/video). While the HTTP offers the multimedia message delivery service, it depends on an always-on Web server with a stable Internet connectivity. Moreover, HTTP that used TCP/IP network provides monitoring and controlling function easily and worldwide access [4]. Availability, security, and costs are some of issues in their deployments because we should provide and maintain the infrastructure by ourself.

Nowadays, Smartphones and Internet social media have eased and enriched the way people communicate [5]. Social network are popular tool used to connect with friends, colleagues and/or family. Internet social media enables people to easily exchange multimedia data [6]. Based on that fact, we propose architecture of interactive home automation that leverages Internet social media platform, such as Twitter. In this recent time, Twitter are not only used as a communication media between human to human, but also human to machine and machine to machine. Microblogging tools such Twitter are now able to use as an interaction media between human and robots [7] and house hold heating system [8]. Twitter social media network is a publish-subscribe system that offers better way of communication than the traditional client-server system. Our paper made smart home system to act as publisher while the householder is subscriber with the Twitter communication mechanism. Both of the publisher and subscriber are having Twitter accounts and "follow" each other. To make the publisher (smart home system) account secure, we create an account with closed features and if we want to control the smart home system we should follow the publisher account and the publisher should accept it. Afterwards, the publisher will inform their followers if have an information about

the home condition, and the subscriber can send a home appliance control message via Twitter Direct messages applications. Since the subscriber can have information sent from the publisher without request to it, the subscriber doesn't have to wait for notifications [9]. With our smart home communication infrastructure, we are not need an always-on Web server and the communication infrastructure are provided by third party social network services.

Twitter, a social media platform, provides Application Program Interface (API) that gives privilege to access user data via HTTP format. We use two kind of account in our home automation prototype: a householder account and a home system account. The home system account is connected to a home automation prototype, mainly a microcomputer with low-power processor, Rasberrry-Pi. We propose to use microcomputer that combines computer and microcontroller function to communicate over TCP/IP network [4]. The prototype is equipped with a Passive Infrared (PIR) sensor as a motion detector and a High Definition (HD) USB camera to capture the home environment. An electrical load is also connected to the prototype to emulate the control system of home automation.

Overall, our home automation is able to receive control message from householder and forward the response to the control devices. Our home automation prototype will send a notification to the householder account if some motions were detected by PIR sensor. The householder is also able to control their home appliances by sending control message and responded by the electrical load. Recent home condition could be monitored by sending request captured-image to system account. Finally our research compared the complexity of packet transmission and the feature of our smart home communication mechanism with the client-server models.

## 2. Research Method

We designed smart home system prototype to implement Twitter publish/subscribe interactions. The prototype consist of microcomputer Rasberrry pi [10] equipped with several input output device and the computer network interface. The input output device selection is based on the smart home monitoring device categorized as sensor, physiological device, and multimedia device [1]. **Error! Reference source not found.** described our input output device categorized as sensor and multimedia device. We put Passive Infrared Sensor (PIR) and enable Generic Purpose Input Output (GPIO) as a digital switch and electric load driver switch [11]. Our prototype is also equipped with HD camera for home monitor and captures an image event. GPIO is used for simulation purpose. Later on we put an application and Twitter Account inside the prototype to make householder able to remote the smart home systems via Twitter. PIR sensor simulate if there is intruder.

In Application design, a program code must be prefixed with initialization of the Twitter API using user token and secret code. Communication between the automation system is designed by using social media data streams that always listen to messages sent on social media. If there is message sent via direct message to the house holder account, the system will respond with the specified command. The flow diagram for the software is shown in Figure 2.

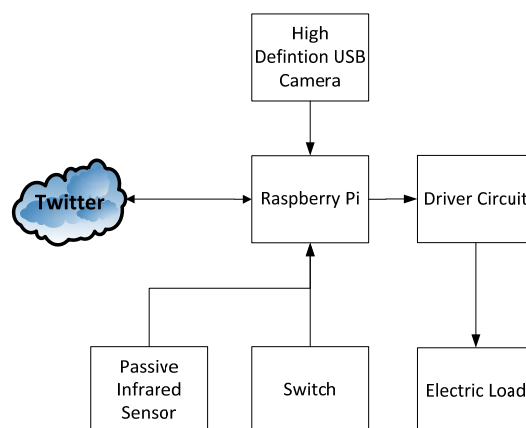


Figure 1. Smart Home Prototype

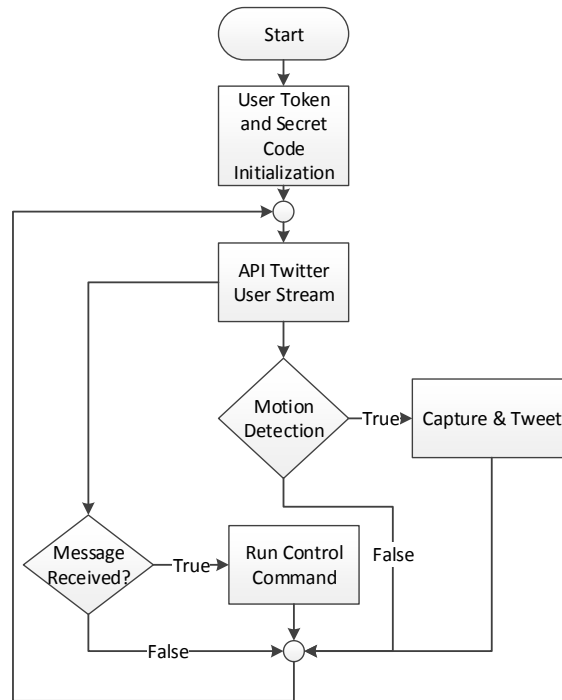


Figure 2. Smart Home System Twitter API

To enable Twitter publish and subscribe system, we create publisher and subscriber Twitter account inside the system environment. Subscriber is the householder Twitter account and publisher is smart home systems account. As shown in Figure 3, we named @smahom3 as our smart home systems Twitter account. For the householder we give @hahan @ekostwn as their account. Publisher will inform all subscribers whether there is condition change in smart home system. We simulated the condition change in smart home systems in GPIO, if there is trigger from switch or passive infrared sensor, the smart home systems will publish the information to the subscribers. The subscribers will also able to change the conditions GPIO switch and an electric load by sending direct message to the smart home systems. Format message is shown in Table 1.

Table 1. Publisher Message Format

Input	Event
#home kamera on	Capture an image from attached camera and publish the picture to all used
#home kontak1 on #home kontak1 off #home kontak2 on #home kontak2 off	Remotely changed status in electronic load driver switch GPIO
#home info	Ask the system about recognized message format

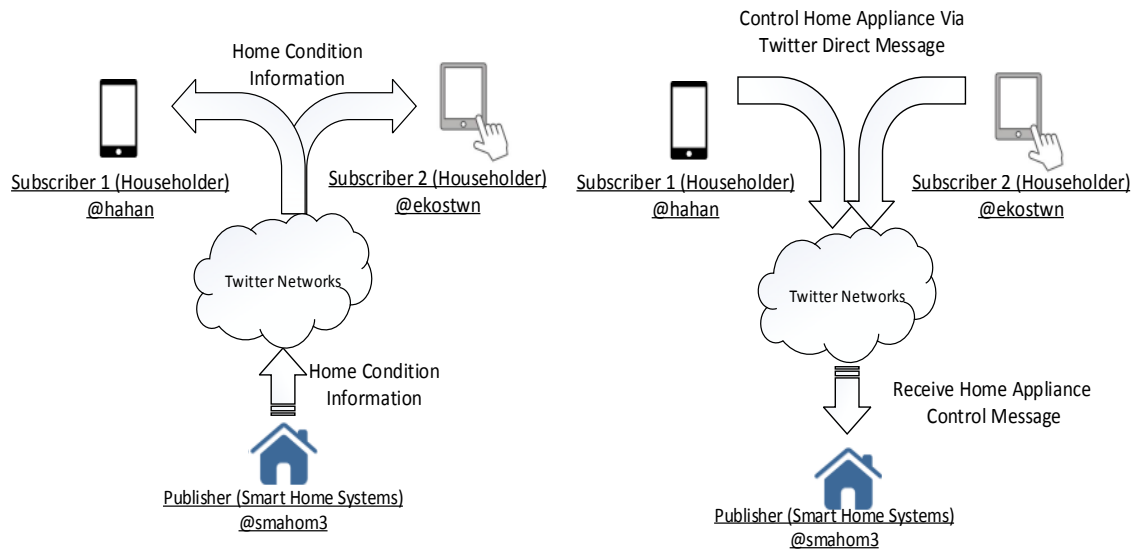


Figure 3. Smart Home systems Publish and Subscribe Scenario

### 3. Result and Analysis

To validate our system, we build a prototype in our laboratory as shown in **Error! Reference source not found.** The system consist of Raspberry-Pi as microcomputer, GPIO interface circuit, home appliance, USB webcam, PIR sensor and push-buttons as input. GPIO interface circuit relay the low voltage of microcomputer logic to high voltage of home appliance. Using the interface circuit, microcomputer can control home appliance in different voltage. Microcomputer was connected to the internet network via Ethernet cable.

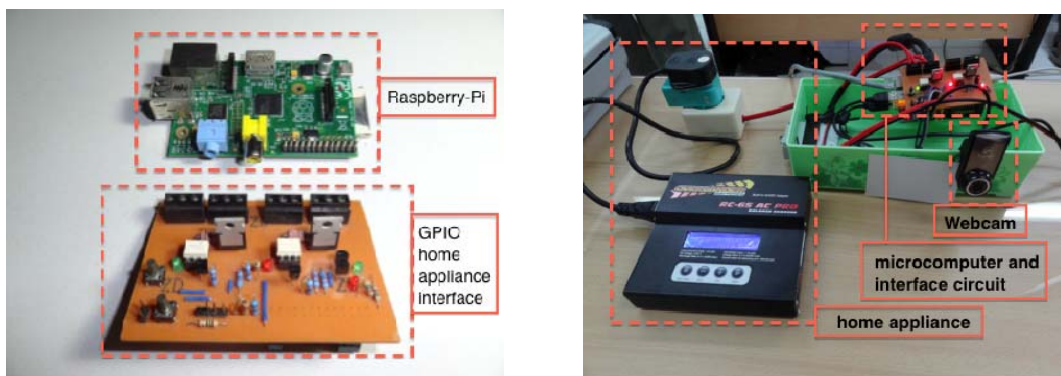


Figure 4. Smart Home systems prototype

The home system worked on Linux operating system environment. Flow diagram of system was implemented in Python language which programmed on Raspberry Pi. Python library generate HTTP message format which according to Twitter API. Webcam was handled by ffmpeg application. Overall program is linked each other by bash command. Bash command execute automatically when the system booting-up. Hence, by turning on the hardware, all program will run automatically.

#### 3.1. Execution Time

To get the information how fast the system works, we conduct a home remote control scenario and measure execution time. In the scenario, householder send an instruction to the system via Direct Message of Twitter. Home system will respond by stream data to the Twitter.

Twitter as a “broker”, will send a notification to the home system. At the next step, home system will parse the message, decapsulation and execute based on command. Home system publish an notification via Twitter and all subscribers, who are all householders, will receive recent information about house condition without request in into system.

We examined execution time by conducting 10 trials. Each control proses consist of three main sequences. Home system check the new message, execute message and publish the report to all subscribers. The home system only need average time 3672.96 ms to read, execute and report to all householder. The experiment show that the publish and subscribe mechanism only need 1434.33 ms average time for get new data as shown in Table 2. Table 2 shows the detail of execution time per process. The execution time for check new message gained from code shown in

Figure 5. We measure the time between the message come to our system, parsing the string as variables and turn on ACT led as an indicator whether a message started with “#home” format had arrived.

```
def on_message(self, message):
    #get time before message arrived
    start_time = time.time()
    print "DM: %s: %s" % (message.sender_screen_name, message.text)
    #control by Dirrect Message
    msg_date = str(message.created_at)
    msg_str = str(message.text)
    msg_user = str(message.sender_screen_name)
    part = msg_str.split(' ')
    if part[0] == '#home':          #hashtag checking
        GPIO.output(11, GPIO.HIGH) #ACT LED on
    #get time after the message arrived
    print("--- %s seconds ---" % (time.time() - start_time))
```

Figure 5. Execution Time Code for Check New Message

We also measure the average time needed to execute a relay switch. The execute command execution time code shown in

Figure 6 . For example, user send a message with format “#home kontak1 on”then an ACT led will be activated and then relay will on. We connect the relay with a neon bulb to check the relay status. We measure the process after the message format are parsed until we get the ‘on’ string and activate the relay.

```

elif part[1] == 'kontak1':
    start_time = time.time()
    c = time.ctime()
    b = c.split(' ')
    if part[2] == 'on':
        GPIO.output(15, GPIO.HIGH) #activate relay
        print("--- %s seconds ---" % (time.time() - start_time))
        sta = '@' + msg_user + ' kontak1 on (' + b[3] + ' )'
        api.update_status(status=sta)
        print sta
    elif part[2] == 'off':
        GPIO.output(15, GPIO.LOW) #deactivate relay
        print("--- %s seconds ---" % (time.time() - start_time))
        sta = '@' + msg_user + ' kontak1 off (' + b[3] + ' )'
        api.update_status(status=sta)
        print sta
    
```

Figure 6. Execution Time Code for Execute Command  
 Table 2. Execution time per process

No. Of trial	Process execution time (ms)			Total execution time (ms)
	Check new message	Execute the command	Publish the status	
1	1471.80	1180.68	1067.10	3719.58
2	1419.97	1187.61	988.59	3596.17
3	1430.10	1176.12	1022.29	3628.51
4	1404.67	1188.89	1041.04	3634.60
5	1449.48	1284.12	1007.52	3741.12
6	1399.18	1190.31	1046.64	3636.13
7	1471.45	1185.02	992.96	3649.43
8	1400.98	1292.42	1041.04	3734.44
9	1444.71	1194.59	1089.35	3728.65
10	1450.97	1156.49	1053.50	3660.96
<b>Average</b>	<b>1434.33</b>	<b>1203.63</b>	<b>1035.00</b>	<b>3672.96</b>

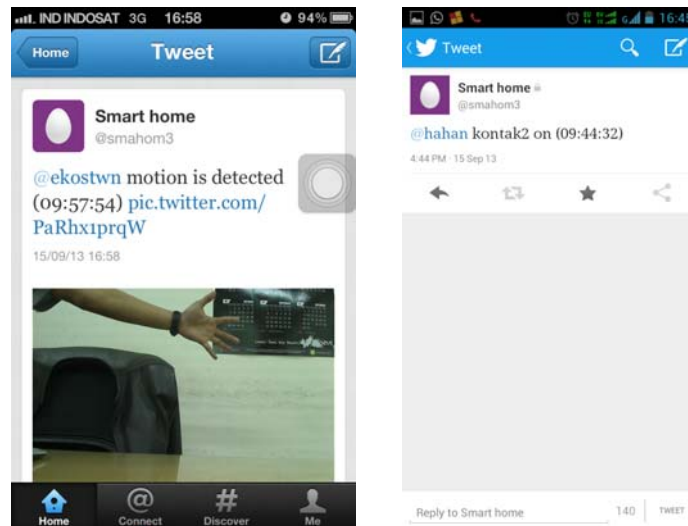


Figure 7. User interface in IOS and Android

We also examine the publish and subscribe respond time in multiple user. The experiment hold in several smartphones. One user will send a command to the home system account via default Twitter application on smartphone. Home system will execute the command

and publish the report by mentioned all user. Some notification will received on user smartphone. We test on IOS and Android operating system smartphone, shown in

Figure 7. Timer start to be counted when first user send command. Timer stop when all user already received the notification. **Error! Reference source not found.** show correlation between execution time on user-side and number of user.

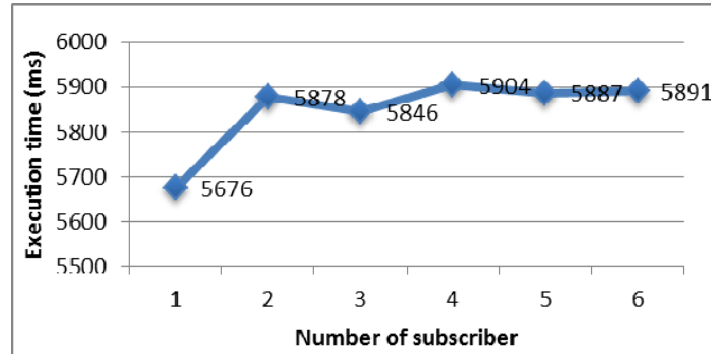


Figure 8. Time response in multiple subscriber

Table 3. Client-server and publish-subscribe comparison

No	Feature	Client-server	Publish-subscribe
1	Always-on host	Required. Always-on host must exist as server. When client send request, server must be on or it will lost the information	Not compulsory. Owner can sent command even home system is off. The command will be stored in broker and it will automatically downloaded when home system is on.
2	IP public	Required. Server or home system must have unique IP public to be able receive request	Not required. IP address can be changed dynamically. Householder can send command even the IP address is changed
3	Network maintenance	Require periodic maintenance.	Already handled by third party broker.
4	Time of transfer	Depend on number of client. Many householders will require much time to transfer	Short and not depend on householder. System can send to householder at the same time
5	Security	System design can determine the level of its security	Network security depend on broker management

Based on **Error! Reference source not found.**, it show that the number of subscriber did not take effect on execution time. Publish subscribe system deliver the short execution time for multiple user. Incremental of execution time was not slightly different between each number of user. This happen due to parallel process at each user. One user in publish subscribe system created connection to the “broker” separately. Every process was handle by Twitter as a “broker”. Home system also create its own connection to Twitter. When some update occurred by home system, every user was notified the change by Twitter parallelly.

Finally, Our research compare client-server system to publish-subscribe system. Each point of view is described on Table 3. Based on Table 3, the publish-subscribe mechanism give more advantages than client-server. Publish-subscribe provide a better communication way. Publish-subscribe has high-dependently to the “broker” system. Selection of better and stable “broker” will ensure lifetime system longger. Client-server system is more independent system than publish-subscribe. Our proposed method show the better system in handling mass communication.

#### 4. Conclusion

In this paper, a concept of publish-subscribe mechanism in smart home interaction was presented. It was shown that publish-subscribe has real-time response. The system could respond user in 3.7 s averaged. The publish-subscribe system show the best way in handling mass data communication. The data could be distributed spreadly in a short time. From the comparison to client-server, publish-subscribe propose lack of maintenance, simple

infrastructure with high capability in handling mass communication. Selection of good broker service is very important in publish-subscribe lifetime.

As part of future research, data encryption in publish-subscribe should be conducted. Building encryption system will keep secret data flowing in public system. Learning the behavioral of householder via its publish will provide a smart system. The home system will understand desire without saying anything.

### References

- [1] MR Alam, MBI Reaz, MAM Ali. A Review of Smart Homes—Past, Present, and Future. *IEEE Transactions On Systems, Man, and Cybernetics*. 2012; 42(6): 1190-1203.
- [2] V Ahmed, S Ladhake. *Novel Low Cost Remote Monitoring System for Home Automation using Cell Phone*. International Conference on Computing Intelligence and Communication System, Gwalior. 2011.
- [3] M Chan, D Estève, C Escriba, E Campo. A review of smart homes—Present state and future challenges. *Computer Methods and Programs in Biomedicine*. 2008; 91(1): 55-81.
- [4] Ashari. Distributed Monitoring and Controlling Using Microcontroller and Virtual Internet Protocol. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2010; 8(2): 285-292.
- [5] Ross EO, M Sisic, J Arseneault, M Simmering, R Orr. Personality and motivations associated with Facebook use. *Computers in Human Behavior*. 2009; 25: 578-586.
- [6] K Curran, S Morisson, S McCauley. Google+ vs Facebook: The Comparison. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2012, 10(2): 379-388.
- [7] Bell TK, S Lauria, R Macredie, J Sutton. Microblogging as a mechanism for human–robot interaction. *Knowledge-Based Systems*. 2014; 69: 64-77.
- [8] NA Gamagedara A, L Stanislao, S Love. Twitter Controls the Household Heating System. *International Journal of Sustainable Energy Development (IJSED)*. 2013; 2(3/4): 101-104.
- [9] A Virgillito. *Publish/Subscribe Communication Systems: from Models to Applications*. Universit'a degli Studi di Roma "La Sapienza" Dipartimento di Informatica e Sistemistica, Roma. 2003.
- [10] Raspberry Pi Foundation. *Raspberry pi*. [Online]. Available: <http://www.raspberrypi.org/>. [Acesso em 12 December 2014].
- [11] Texas Instruments. *MOC3020 THRU MOC3023 Optocouplers/Optoisolators*. Dallas, Texas. 1998.