International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Smart City Monitoring System using Wi-Fi

Vatsal Parikh[1], Shashank Thakare[2], Yogesh Thakare[3]

[1, 2, 3] Electronics and Telecommunication Engineering Department, Pune Vidhyarthi Griha's College of Engineering and Technology, 44, Vidya Nagari, Shivdarshan, Parvati, Pune – 411009, Maharashtra, India

Abstract: ***This paper presents an application of internet of things to enable a smart city by monitoring crucial environmental parameters such as temperature, noise, pollution. The system is also designed to monitor and control street light and traffic signal for effective use of electrical energy and smooth vehicular transportation. All the necessary parameters to be monitored are sensed using various sensors readily available. The measured data from sensor is processed using intelligent microcontroller based system. This processed data is passed on to android smartphone through Bluetooth and made available wirelessly on the server for storage and access continuously through internet. This kind of monitoring system can be used to find out disturbances in different parts of the city and take appropriate actions accordingly.***

Keywords: Android, Database, Wi-Fi, Smart City, IoT**.**

1. Introduction

As reported in [1], 70% of the world population is expected to live in cities and surrounding regions by 2050. As per census 2011, in India, nearly 31% of current population lives in urban areas and contributes 63% of India's GDP [2].
Around 2030, it is expected that urban population will increase to 40 % and contribute 75% of India's GDP [3]. Thus, with increasing urbanization, comprehensive development of physical, institutional, social and economic infrastructure will become mandatory to improve quality of life and attracting people and investments to the city. This will help the growth and development in the cities. Development of Smart Cities will remain a step ahead in that direction [4]. Smartness of a city is driven and enabled technologically by the emergent Internet of Things (IoT) [5].
We are moving towards a radical evolution of the current internet into a ubiquitous network of interconnected objects that not only harvests information from the environments (sensing) and interacts with the physical world (actuation/command/control), but also uses existing Internet standards to provide services for information transfer, analytics, control and action [6].

World leading municipalities, in terms of services and quality of life, have provided efficient services to their citizens by the forward thinking and use of technology in monitoring various environmental parameters. Most of these systems consist of sensor, data storage device, and computer at a base station where experts analyze the data [7]. Our motivation behind this implementation was to use the concept of IoT to solve basic city problems more smartly and efficiently.

2. Proposed System Design

The proposed smart city monitoring system using Wi-Fi is shown in Figure. 1. It consists of end user node and remote database sections. The end user node includes the microcontroller, various sensors and android smartphone whereas remote database includes Java client as user interface, Java server, and database.

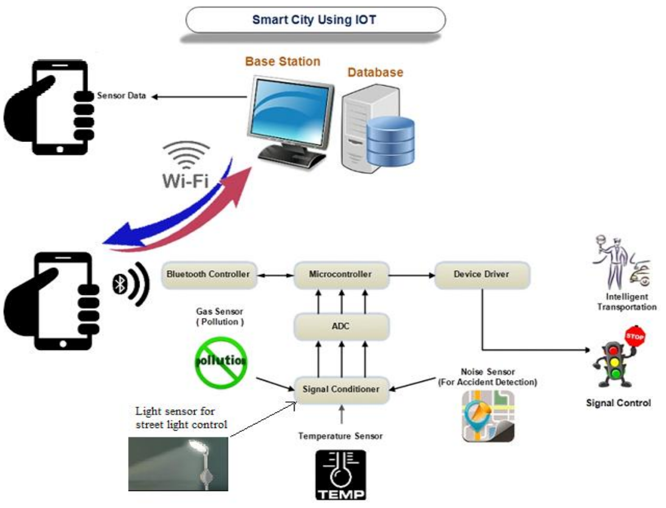


Figure 1: Block diagram of the proposed system

As shown in Figure 1, various city parameters to be measured are traffic flow, pollution level, noise level, temperature, and light intensity. IR sensor is used to measure the density of traffic at a traffic junction. The feedback from the IR sensors is used to turn the traffic lights green, yellow, or red to make a smart signaling for the traffic. Gas sensor measures the amount of carbon monoxide in air to know the level of pollution in a particular area to be displayed. Microphone sensor detects any unusual spikes in noise level. Thermistor is used to sense the temperature. Light dependent resistor detects the intensity of sunlight falling on the road. During the evening time when light intensity is low, streetlights are to be turned on, else streetlights are to be turned off. Output of temperature sensor, light sensor and gas sensor are given directly to the microcontroller without any signal conditioning, but the output voltage level obtained from sound sensor is very low, hence it requires a three-stage amplification signal conditioning circuit. The processed output of the sound sensor is given to the microcontroller. The analogue to digital converter (ADC) embedded in the microcontroller converts all the sensor output in digital form. The digital output has been serially communicated using Bluetooth module to android application.

The sensor data available with android application are to be transferred to MySQL database on Java server using Wi-Fi. MySQL database can be viewed using Java client which displays the current sensor values obtained and maintains a log of all the sensor data for the desired number of days. The specifications of the proposed system are shown in Table 1.

Table 1: Proposed system specifications

1.	Power supply	12V x 1A, 5V x 1A
2.	Operating temperature	-10°C to 70°C
3.	Operating frequency	2.4GHz ISM band
4.	Communication range	Wi-Fi : 30 meters Bluetooth : 4-10 meters
5.	Memory requirements	32Kb microcontroller memory 5MB android ROM storage 100KB remote server log
6.	Light sensor Temperature Sensor Gas sensor Sound sensor Infrared sensor	Power Dissipation 400mW/C Max. Allowable Power 550mW 5.0V±0.2V AC or DC 4.5V, Omnidirectional. 5V x 1A, Wavelength: 940nm
7.	Android device	Android based smartphone with inbuilt Wi-Fi, Bluetooth
8.	Android version	Android 2.3 or above
9.	Remote database	MySQL 5.1 or above
10.	Server	Glassfish server 3 (NetBeans 7.1)
11.	Java client	JDK 1.6 or above, NetBeans 7.1)

3. Software Design

In the proposed system, microcontroller acts as slave and android device acts as master. The data from the microcontroller is sent to android application only when android application requests for it. Figure 2 shows various functionalities of android application. The android application asks for IP address of Wi-Fi for connecting to the remote database server. Once the application is connected to remote server login appears screen asking for authentication. After logging, one can select one of the three options i. check senor, to check whether all sensors are working properly, ii. check device, to check whether the device is working properly and iii. click to control city, to take user to the next screen. By switching the topmost button ON the android application starts requesting microcontroller for the data available on the microcontroller port. This serial communication happens through Bluetooth module. The data obtained is displayed on the android application and transmitted in real-time to a remote server. Android application also has a functionality of noting down threshold values of each sensor. If the value obtained from sensor goes above (or below) threshold then an alert message is sent to respective departments instantaneously so that suitable action can be taken.

Figure 3 shows Java client process flowchart based on client-server model. Glassfish server module in Netbeans 7.1 is used to build a server that has the required web services and Java DataBase Connectivity (JDBC). Java client is the main application program for collecting, storing and displaying all sensor data obtained from various parts of the city.

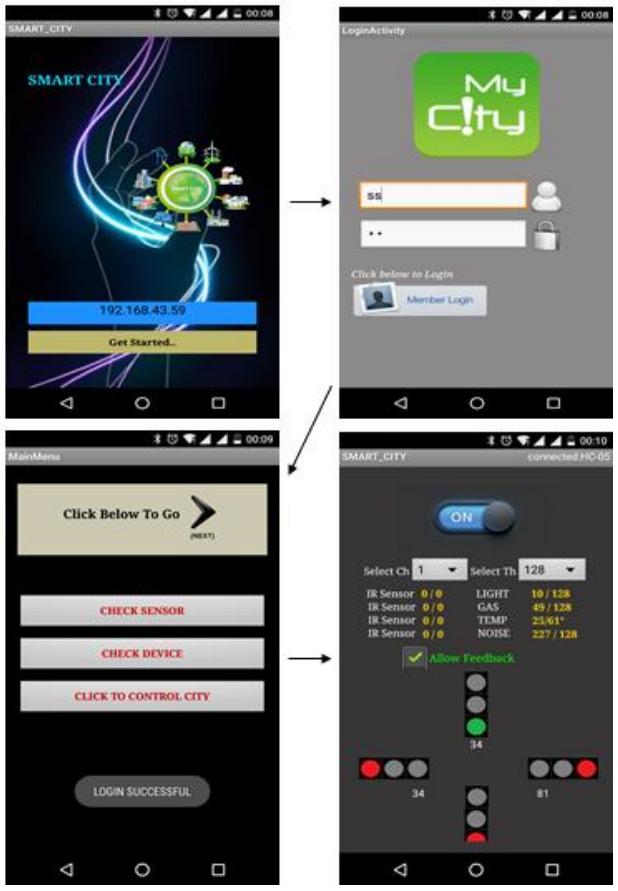


Figure 2: Android application process flow

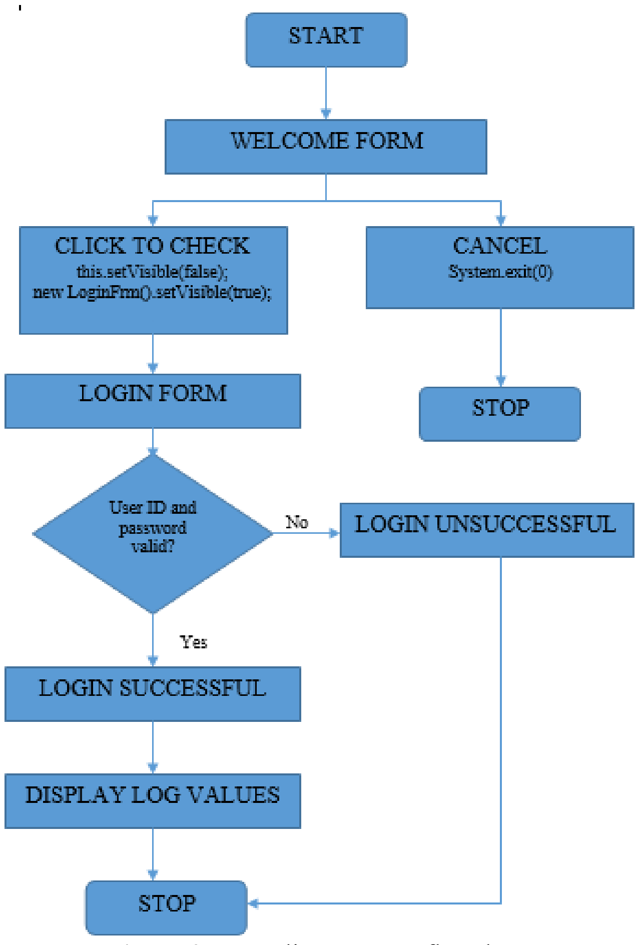
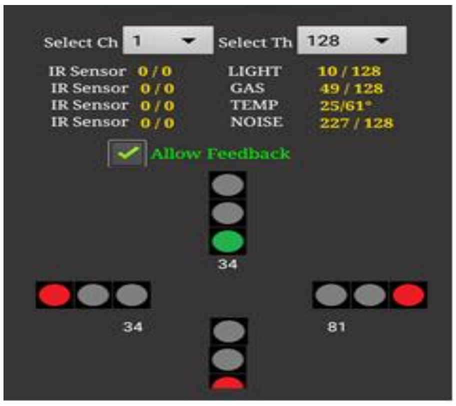


Figure 3: Java client process flowchart

4. Results and Discussion

The measured sensor values are displayed on android smartphone as shown in Figure 4.



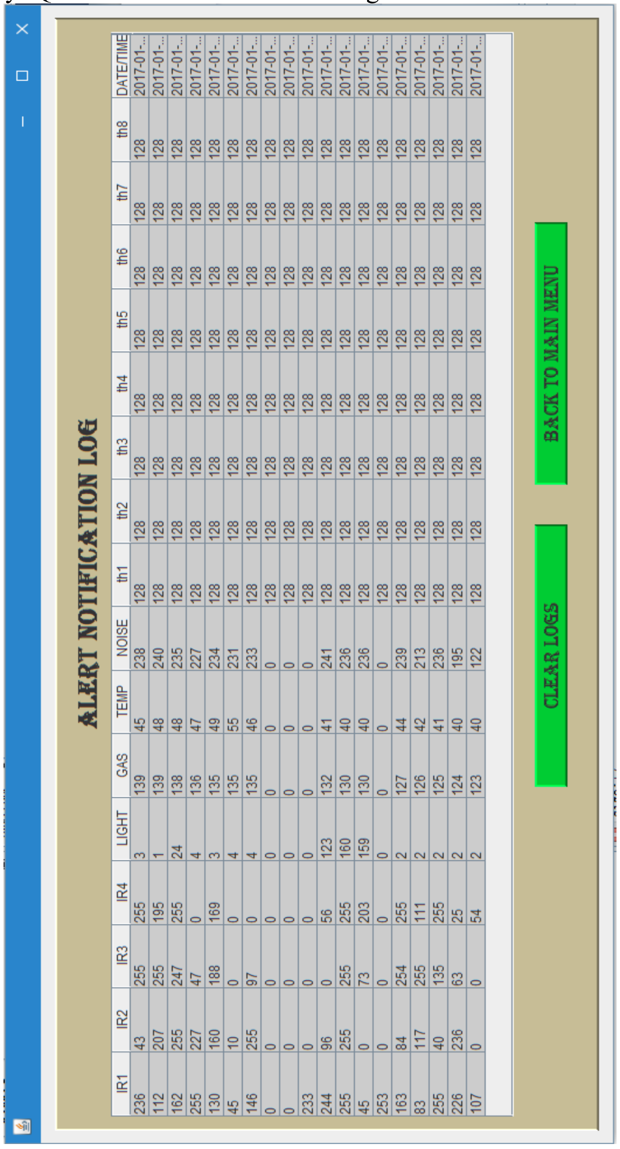
Select Ch 1 | Select Th 128

IR Sensor	0 / 0	LIGHT	10 / 128
IR Sensor	0 / 0	GAS	49 / 128
IR Sensor	0 / 0	TEMP	25/61°
IR Sensor	0 / 0	NOISE	227 / 128

✓ Allow Feedback

Figure 4: Snapshot of sensor values on application

This data is then sent through Wi-Fi to a remote database server equipped with Java client and monitored at the rate of one dataset per minute. The database values obtained in MySQL database are as shown in Figure 5.



ALERT NOTIFICATION LOG

IR1	IR2	IR3	IR4	LIGHT	GAS	TEMP	NOISE	th1	th2	th3	th4	th5	th6	th7	th8	DATE/TIME
236	43	255	255	3	139	45	238	128	128	128	128	128	128	128	128	2017-01-...
112	207	255	195	1	139	48	240	128	128	128	128	128	128	128	128	2017-01-...
162	255	247	255	24	138	48	235	128	128	128	128	128	128	128	128	2017-01-...
255	227	47	0	4	136	47	227	128	128	128	128	128	128	128	128	2017-01-...
130	160	188	169	3	135	49	234	128	128	128	128	128	128	128	128	2017-01-...
45	10	0	0	4	135	55	231	128	128	128	128	128	128	128	128	2017-01-...
146	255	97	0	4	135	46	233	128	128	128	128	128	128	128	128	2017-01-...
0	0	0	0	0	0	0	0	128	128	128	128	128	128	128	128	2017-01-...
0	0	0	0	0	0	0	0	128	128	128	128	128	128	128	128	2017-01-...
233	0	0	0	0	0	0	0	128	128	128	128	128	128	128	128	2017-01-...
244	96	0	56	123	132	41	241	128	128	128	128	128	128	128	128	2017-01-...
255	255	255	255	160	130	40	236	128	128	128	128	128	128	128	128	2017-01-...
45	0	73	203	159	130	40	236	128	128	128	128	128	128	128	128	2017-01-...
253	0	0	0	0	0	0	0	128	128	128	128	128	128	128	128	2017-01-...
163	84	254	255	2	127	44	239	128	128	128	128	128	128	128	128	2017-01-...
83	117	255	111	2	126	42	213	128	128	128	128	128	128	128	128	2017-01-...
255	40	135	255	2	125	41	236	128	128	128	128	128	128	128	128	2017-01-...
226	236	63	25	2	124	40	195	128	128	128	128	128	128	128	128	2017-01-...
107	0	0	54	2	123	40	122	128	128	128	128	128	128	128	128	2017-01-...

Figure 5: Snapshot of measured values on remote database

The measured parameter values are tested for accuracy using weather.com for temperature sensor, lux meter for light sensor, sound level meter for sound sensor, carbon monoxide detector for gas sensor, and were found in close agreement. Some of the constraints of the proposed system are, need for compatibility with legacy software versions, large memory requirements. This system can be improved to detect and generate automated alert of inconsistent values. Further, memory requirements can be reduced by archiving the old data onto another server.

5. Conclusion

Smart city monitoring system is a solution catering to the needs of current and future generations. This is possible through effective use of IoT enabled technologies. This paper aims to provide such a solution by using intelligent microcontroller based system with Android and Java.

6. Future Scope

This system can be modified by using different sensors to behave as water metering system or smart parking system. Android smartphone camera can be leveraged to form a scalable city surveillance system.

References

[1] J. Belissent, ‘Getting Clever About Smart Cities: New Opportunities Require New Business Models,’ Forrester Research Inc., New York, NY, USA, 2010.
[2] Census 2011, [Online] Available: http://censusindia.gov.in/2011-common/censusdataonline.html
[3] T.M. Vinod Kumar, ‘Making Delhi a Smart City: Economic Buoyancy with Spatial Justice’ in Smart Economy in Smart Cities, 1st ed: Springer, 2017, ch. 20, sec. 20.1, pp. 497.
[4] Smartcities.gov.in, ‘Smart Cities Mission Statement & Guidelines’, June 2015. [Online] Available: http://smartcities.gov.in/writereaddata/smartcityguidelines.pdf
[5] L. Atzori, A. Iera, and G. Morabito, ‘The Internet of Things: A survey,’ Comput. Netw., vol. 54, no. 15, pp. 2787–2805, 2010.
[6] Smart Networked Objects and Internet of Things, white paper, Association Instituts Carnot, Greece, 2011.
[7] Jiong Jin, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami, ‘An Information Framework for Creating a Smart City Through Internet of Things’ IEEE Internet of Things Journal, vol. 1, no. 2, April 2014.