

IoT Based Urban Climate Monitoring using Raspberry Pi

Rohini Shete and Sushma Agrawal

Abstract—Internet of Things is the web of physical objects that contain the embedded technology which is helping to develop man to machine or machine to machine communication. This paper mainly propounds a stand-alone system which is providing a dynamic datasheet about the parameters of the city environment. The system is using low cost low power ARM based mini-computer that is Raspberry Pi. It can communicate through Local Area Network (LAN) or the external Wi-Fi module. Commands from user are processed at the Raspberry Pi using Python language. The data can be monitored with other terminal devices like Laptop, Smart Phone and Tablet which is endowed with the internet facility. This framework is giving access to real-time information about an urban environment which includes the parameters: temperature, humidity, pressure, CO and harmful air pollutants.

Index Terms—Adafruit, Internet of Things (IoT), Monitoring, MQ Telemetry Transport (MQTT) Raspberry Pi

I. INTRODUCTION

This system has developed in the approach to promote a decent quality of life to people, a clean and sustainable environment. In [1] author has stated that focus is on sustainable and inclusive development with the help of the IT and electronics development. The emerging concept of IoT plays a vital role to help in attaining this mission. Policy decisions can be made based on real data. And more importantly, their impact can be monitored almost in real time. The IoT concept was put forth in 1999 by a member of the Radio Frequency Identification (RFID) development community. The increasing growth of the mobile devices and development in the area of communication, cloud computing, embedded systems has made the IoT concept more relevant. In [2] it is stated that IoT is a converging technology to create the smart environment and the integrated ecosystem.

Rohini Shete is with Electronics and Telecommunication Department, Government College of Engineering, Aurangabad, Maharashtra, India(email: rohinishete1107@gmail.com)

Sushma Agrawal is with Electronics and Tele-communication Department, Government College of Engineering, Aurangabad, Maharashtra, India(email: agrawal.sushma1@gmail.com)

Analysts predict that new IoT products and services will grow exponentially in next few years. IoT will boost the economic growth while improving our citizen's life. The general architecture of the IoT is complex to define because it involves the variety of devices and different link layer technologies. The system helps in providing the open access to the particular set of data. The citizens as well as the administrative authority are going to get the idea about the city environment at any remote place with the help of internet. The use of the smart and compact mini-computer kind of device that is Raspberry Pi is very useful in giving good internet connectivity and the use of it is also going to boost the speed of operation. This framework mainly envelopes the urban environmental information from the sensory level and also involves the networking, data management and integrates the real time data. In [3] authors state that the system can help in the great transformation in the city management and improve the life of citizens. The idea will also help in creating the public awareness among the people about the climate changes in the city.

The system has low cost and high latency. The system draws very less power compared to the existing systems and it is highly reliable. The Pi can act as an effective server and provides good internet connectivity. The weather monitoring systems based on other micro-controller needs extra Ethernet module to get internet connectivity but Pi has on board circuitry for this and the delay in operation is very less.

This paper is distributed in six sections. Section I gives the basic idea about the system. Section II explains the hardware and software architecture. The experimental setup in which sensors interfaced with the Raspberry Pi is shown in section III. Section IV signifies the practical outputs of the parameters monitored by the system. Section V and section VI conclude and foresee about this technology respectively.

II. SYSTEM DEVELOPMENT

A. Hardware Architecture

Fig. 1 shows the block diagram of the system. The Raspberry Pi as a cardinal controlling node of the proposed system. The environmental parameters that have been brought into focus are Temperature, Humidity, Pressure, Light Intensity, Carbon Monoxide and polluting gases. The data sensed by these sensors is continuously sent to the Raspberry Pi. The sensors, DHT22, BMP180 and BH1750 direct digital

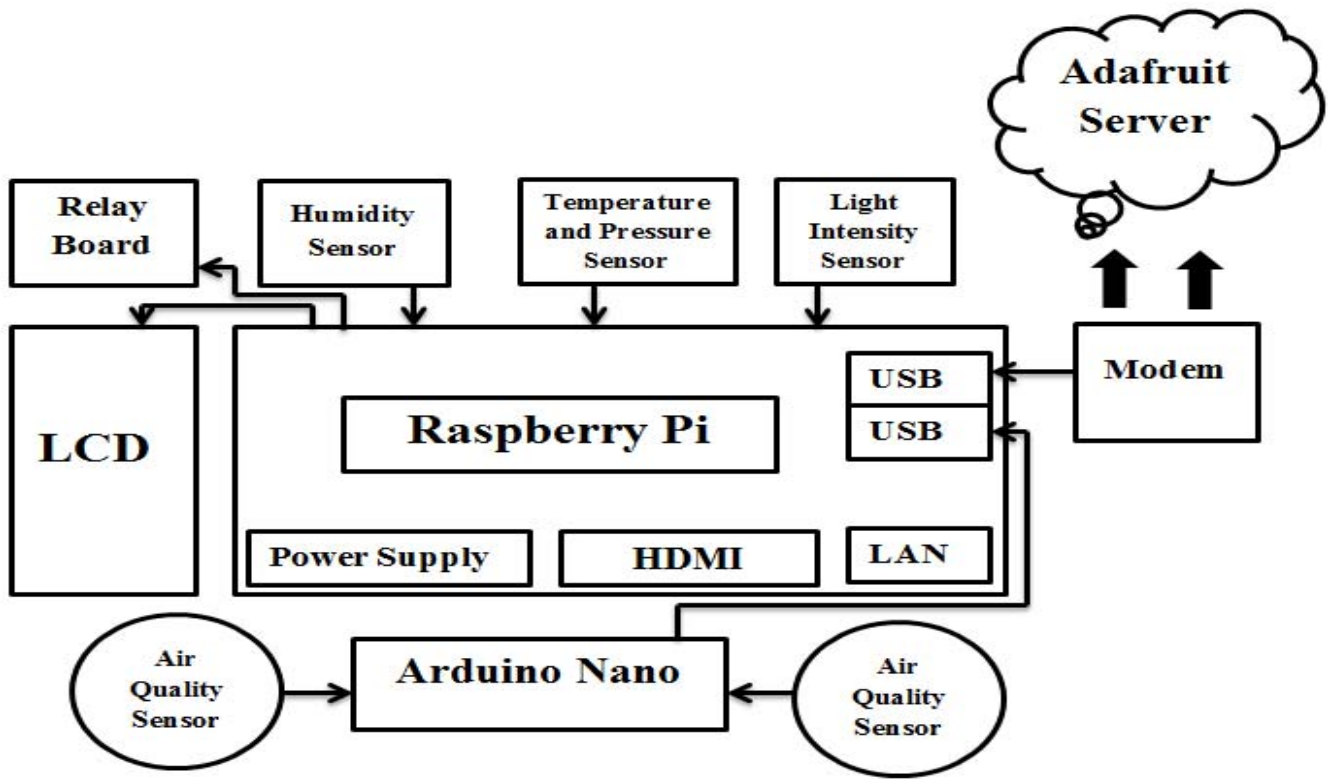


Fig. 1. Block Diagram

output that can be sensed by the GPIO pins of the Pi. The gas sensor MQ7 for Carbon Monoxide and air quality sensor MQ135 send their data to the analog pins of the Arduino Nano which is connected to the Raspberry Pi through the USB cable. The Arduino Nano is a compact and cheap module that can be easily connected to the Pi and acts as an effective ADC. A wireless modem is used for providing internet connectivity to the system. The client can use the link for the dashboard and gets access to it. The different feeds that are shown on the dashboard give dynamic data that is fed continuously to server. The MQTT protocol plays a key role in providing this communication between the sensor nodes and the clients. The client will be able to monitor the data but will not be able to make any changes in the parameters related to it.

A. Raspberry Pi Board

In [4], Raspberry Pi 2 model B is a 85*56mm device with BCM2836 quad core (4 processors in one chip) ARMv7 processor. It is superior to the Raspberry Pi 1 Model B+ with its 900MHz speed compared to the 700MHz speed of the former one. It has 1 GB RAM which makes it the best choice for using it with the better browser performance. It has 4 USB ports and a full HDMI port. It is also designed effectively with the 40 pin GPIO. The other features include a combined 3.5mm audio jack and composite video with Video Core 4 3D graphics core. It has a separate camera as well as display interface. Raspberry Pi can be connected to a local area network through Ethernet cable or USB Wi-Fi adapter, and then it can be accessed by any client in the network. The SSH remote login or putty software can be used by which we can get access to Raspberry Pi by just putting its IP address.

The raspberry pi is booted by micro SD card. The different operating systems which are supported by the Raspberry Pi are RASPBIAN, PIDORA, OPENELEC, RASPBMC, RISC OS and WINDOWS 10 as well. The system is using RASPBIAN operating system which is the massively popular OS also known as “distro”. PYTHON is main programming language which is a highly flexible wrapping kind of language. In [5] it is given that the large set of libraries like GPIO, TIME makes the coding very easy and efficient. Its RAM makes it commendable choice to get the real time data processed and accessed faster than other microcontroller based systems

B. Sensing Unit

The sensing unit consists of total 5 sensors. Table 1 shows the specification details of the two air quality sensor. The conductivity of the sensors varies as the concentration of the gases in the air changes. Table 2 gives the information about the three sensors which measures temperature, humidity, pressure and light intensity. These are small in size and low cost which makes the system reliable and compact.

TABLE I
AIR QUALITY PARAMETERS

Parameters Monitored	Operating Voltage	Measuring Range
Carbon monoxide	5 Volts	10-10000ppm
Ammonia, Sulphide, Benzene, Smoke	5 Volts	10-10000ppm

TABLE II
WEATHER PARAMETERS SENSING

Parameters Monitored	Operating Voltage	Measuring range	Sensor specification
Humidity	3.3-6 Volts DC	0-100 % RH	DHT22 with digital output and resistive type relative humidity measurement
Temperature	1.8-3.6 Volts DC	-40 to +85 degree Celsius	BMP180 with the piezo-resistive sensor and digital output via I2C bus
Pressure		300-1100hPa	
Light intensity level	1.8-4.5 Volts DC	1-65535 lx	BH1750 with the 16 bit digital output

C. Software Architecture

Adafruit IO is the server used by the proposed system. Adafruit IO provides the different set of libraries that are useful for implementing IoT or machine to machine communication. The libraries are available in different languages like C, JAVA, Java Script, .NET, Python etc. The system has used python library for developing client-server model. MQTT client structure is used based on Paho-mqtt. Paho-mqtt is providing us with an open source of libraries in different languages. In [6] it is stated that the communication that take place between a client and a server is effectively done in both transmit and receive manner using a bandwidth efficient and light weight protocol like MQTT. The general HTTP protocol involves the request/response kind of architecture. The MQTT protocol doesn't follow that but uses a publish/subscribe architecture. Publish/Subscribe is event-driven and messages are continuously pushed to clients.

Fig. 2 shows the central communication point that is MQTT broker. It takes care of all the messages that are transmitted and received. In [7] it is stated that each client that publishes a message to the broker includes a topic into the message. The topic is a field that gives routing information to the broker. Each client that wants to receive messages subscribes to a particular topic. The broker delivers all messages with the matching topic to the client. The client's communication completely relies on topics. The clients can communicate effectively without knowing each other. The transmitter of the data and receiver of the data are not at all dependent and hence the architecture gives efficient functionality. The client doesn't need to request all the time to get the required data. The broker keeps pushing or publishing the information continuously. Therefore each MQTT client has to maintain a TCP connection to the broker. If the connection gets interrupted, the buffering of messages is done and those messages are sent when the client is online again.

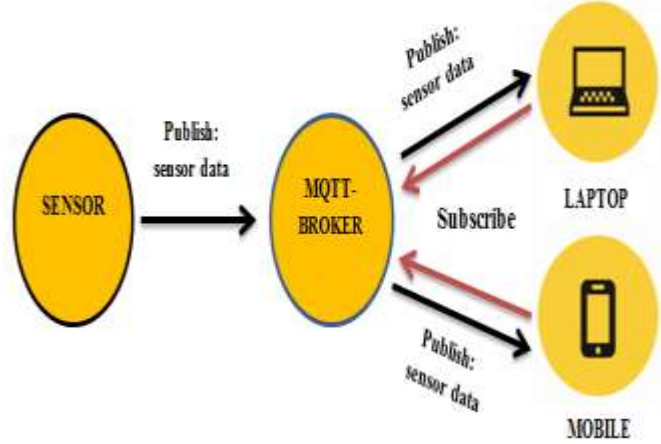


Fig. 2. Publish/Subscribe architecture of MQTT

The main concept of MQTT is based on the topics. This protocol can work as a tree structure and has different hierarchical levels. The topics that we mention can define the different hierarchical levels that are separated by slash. The client can subscribe for any kind of data or the entire sub-tree.

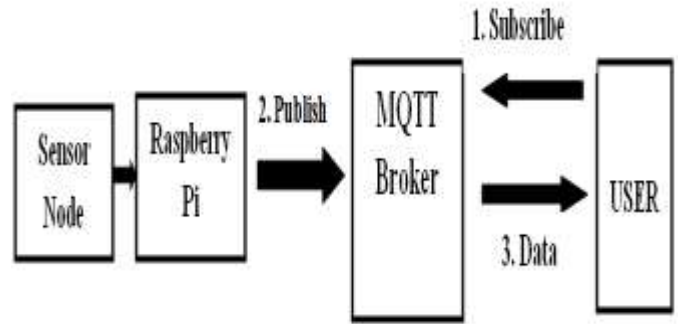


Fig. 3. MQTT based communication between user and server client

Fig. 3 depicts that how the communication takes place between the sensor clients i.e. Raspberry Pi with Sensor node and the user over MQTT. When the user subscribes for the particular data, it is published accordingly through the MQTT broker.

D. Workflow

In [8-11], the required libraries for the communication are found which is installed on raspberry pi using the commands `sudo pip install adafruit-io`. These will automatically install the Adafruit IO python client code for our python script. The simple commands like `aiio.send('feed', parameter value)` and `aiio.receive('button', status of button on dash board)` are used in our python code for server to client communication. The figure 4 depicts that user gets the data by entering the URL and visits the dashboard. The data will be continuously fed to the server and timely updating is shown on dashboard as well as a graph will also be generated.

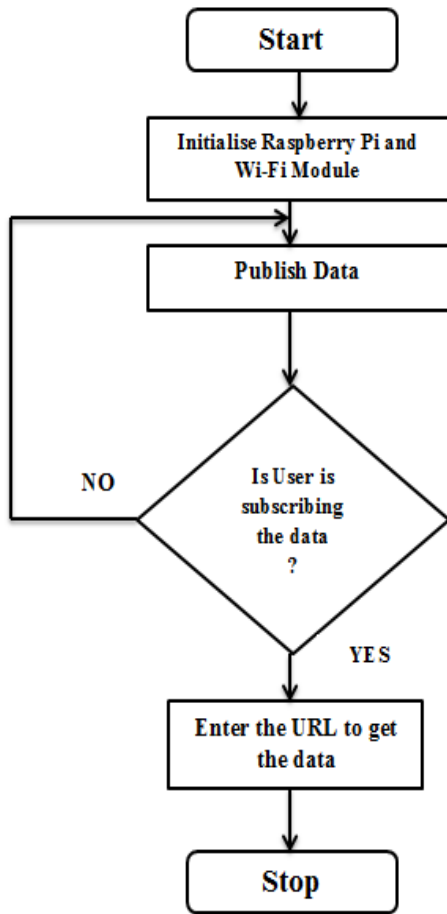


Fig. 4. Flow chart for workflow

III. EXPERIMENTAL SETUP

The Fig. 5 shows the experimental setup for the system. To avoid the complicated wiring from sensors to GPIO a small PCB is designed to connect sensors to GPIO and also to provide some signal conditioning.

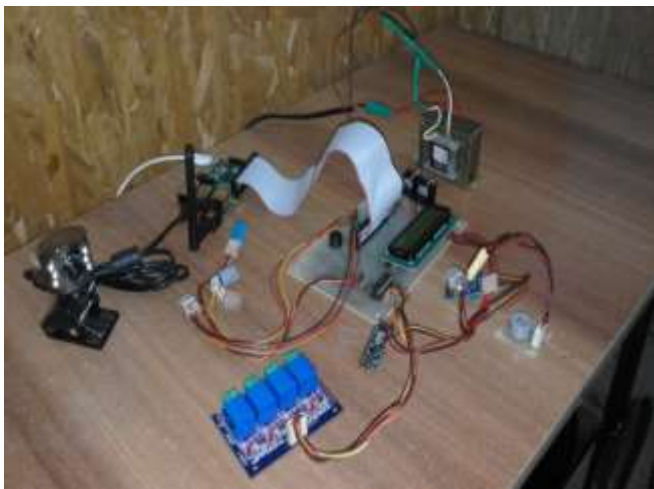


Fig.5 Experimental Setup



Fig. 6. Graph for temperature variation with time

The automatic graphs are generated for all the parameters on the dash board which will shows how the parameters are varying with respect to time. Fig. 6 depicts one of those graphs for temperature. The Different types of blocks on Adafruit IO have made the dash board presentation look more attractive and effective. One more key feature of the Adafruit IO is that it is very easy to edit the dash board designs. The authenticated user can design the dash board not only to monitor but also to control the parameters using that. The status of the particular sensor or actuator can also monitored by extracting that status from the raspberry pi and accordingly different controlling actions or control measures can be taken.



Fig. 7. Parameters shown on the dash board

TABLE III
COMPARATIVE RESULTS IN MORNING TIME

Sensor	Observed Value	Expected Value
Temperature (degree Celsius)	28.1	27
Pressure (Kilo Pascal)	95.67	97.28
Humidity (percentage)	45	42
Pollutant Concentration (ppm)	110	104
Carbon Monoxide (ppm)	260	300
Light Intensity (Lumens/sq.m)	45	50

REFERENCES

- [1] Andrea Zanella, Nicola Bui, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol.1, No.1, 2014.
- [2] "IoT-Converging Technologies for smart Environments and Integrated Ecosystems" by Ovidiu Vermesan and Peter Fiess.
- [3] N Pradeep Kumar, Ravi Kumar Jayoth, "Development of Cloud Based Light Intensity monitoring System using Raspberry pi," *International Conference on Industrial Instrumentation and Control(ICIC)*, 2015.
- [4] "Getting Started with Raspberry Pi" by Matt Richardson and Shawn Wallace.
- [5] <http://www.python.org>
- [6] Yuri F. Gomes, F.S.Santos, "Integrating MQTT and ISO/IEEE 11073 for Health Information Sharing in the Internet of Things," *IEEE International Conference on Consumer Electronics (ICCE)*, 2015.
- [7] Trude H. Bloebaum, Frank T. Johnsen, "Evaluating publish/subscribe approaches for use in tactical broadband networks," *Milcom 2015 Track 5- Selected Topics in Communication*, 2015.
- [8] <https://learn.adafruit.com/category/adafruit-io>
- [9] Sudhir Nikhade, "Wireless Sensor Network using Raspberry Pi and Zigbee for Environmental Monitoring system," *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials(ICSTM)*, pp 376-381, 2015
- [10] V.Sandeep, K.Lalith Gopal, "Globally Accessible Machine Automation Using Raspberry pi," *International Conference on Advances in Computing, Communication and Informatics(ICACCI)*, 2015.
- [11] Ke Hu, Sivaraman, "Design and Evaluation of a Metropolitan Air Pollution Sensing System," *IEEE Sensors Journal*, vol. 16, No. 5, 2016.

TABLE IV
COMPARATIVE RESULTS IN EVENING TIME

Sensor	Observed Value	Expected Value
Temperature (degree Celsius)	29.9	30.7
Pressure (Kilo Pascal)	94.376	96.583
Humidity (percentage)	43	39
Pollutant Concentration (ppm)	124	115
Carbon Monoxide (ppm)	506	400
Light Intensity (Lumens/sq.m)	36.67	40

Fig. 7 shows the values of the different parameters in the evening time. As the pollution level of the city increases in the evening due to heavy traffic, the concentration of the pollutants is high. The table 4 shows the comparative study of observed values of the parameters and the expected values in the evening time. The table 3 has stated the comparative study of the observed and expected values of parameters in the morning time. The observations are made for the Kranti Chauk area of the Aurangabad city in India. The expected values are taken from the weather statistics available online. The comparative study has shown that the system is working quite efficiently.

IV. CONCLUSION

The proposed system gives a compact, low power and low cost system for the monitoring data remotely. The use of the low cost mini-computer Raspberry Pi makes it efficient and reliable. The system aids in the sustainable growth of the city and improves the lives of the citizens. The ubiquitous availability of dynamic datasheets on the dashboard and the time to time graphical representation can help in planning the control measures against increasing pollution levels and create awareness among the people.

V. FUTURE SCOPE

The developments of algorithms for IoT that can sift through massive and diverse data to find the patterns that explain our world without bias is a complex problem. Interactive type of communication between citizens and system, enhanced level of speed and security and the protocols for communication which will be compatible to all kind of servers and clients are the few more challenges.