International Journal of Electrical Electronics & Computer Science Engineering
Volume 2, Issue 4 (August, 2015) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

A Low Cost Internet of Things Network for Contamination Detection in Drinking Water Systems Using Raspberry Pi

Syeda Madeeha Anam[1], M Devender[2]
[1]M.Tech Scholar, Sudheer Reddy College of Engineering & Technology [Womens], Nizamabad, Telangana
[2]Associate Professor, Department of ECE, Sudheer Reddy College of Engineering & Technology [Womens], Nizamabad, Telangana

Abstract*: Internet of Things is the next big revolution of the world on digitalization of commercializing various modules/products. Everything is associated with the internet, some involves controlling and some involves monitoring the parameters from anywhere. In previous days, people talked about WSN (Wireless Sensor Network) and now, we propose ISN (Internet of Things Sensor Network) on IOT Framework using Raspberry Pi. In this paper, water contamination is checked with the help of various sensors which measures the physical parameters such as pH value, conductivity etc… A simple web server is designed and developed in Raspberry Pi and server-client connection is established for logging the data.

***Keywords*: Python API, ARM11 Processor, Raspberry Pi, Internet of Things**

I. INTRODUCTION

Now-a-days, controlling and monitoring plays a main role in our day to day life. Everything we can control using advanced technologies and we can also monitor the things we need. Now we can control and monitor anywhere using Internet of things. If you have Internet in your PC/Mobile you can directly upload the data you need and control it from internet itself. When we are talking about the Internet of things, it's a dream of things to be visualized.

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

ThingSpeak is an open source Internet of Things application and API to store and retrieve data from things using HTTP over the internet or via a local area network.

With think speak, you can create sensor logging applications, and social network of things with status updates.

In addition storing and retrieving numeric and alphanumeric data, the Thing Speak API allows for numeric data processing such as time scaling, averaging, median, summing, and rounding. Each Thing Speak channel supports data entries of up to 8 data fields, latitude, longitude, elevation, and status. The channel feeds support JSON, XML, and CSV formats for integration into applications.

The Thing Speak application also features time zone management, read/write API key management and JavaScript-based from High slide software.

II. SYSTEM ARCHITECTURE

The system architecture of this proposed system is divided into two different blocks.

ARM11 END: Hardware implementation for this proposed system is shown below with the simple blocks. pH Sensor, Conductivity Sensor, Moisture Sensor, Temperature Sensor, Water Level Sensor were connected through MCP320x SPI ADC as all the sensors will produce analog data. The MCP320x is connected at SPI port of Raspberry Pi and will read the data at regular periodic samples. A simple web server is designed and developed to upload the sensory data for logging. Raspberry Pi is having GPIO Hedar of 40 sizes, and a micro SD card where operating system boots out when required.

An Embedded Linux environment is created for working environment and a python program is capable of doing the hardware manipulations.

Power Supply block is designed and developed to generate power source for the ARM processor and its relevant components.

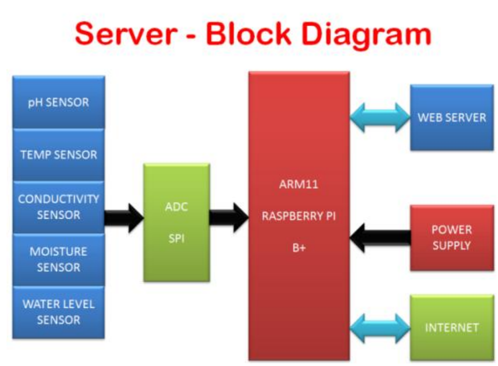


Figure 1: Server Block Diagram

International Journal of Electrical Electronics & Computer Science Engineering
Volume 2, Issue 4 (August, 2015) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

Client – Block Diagram

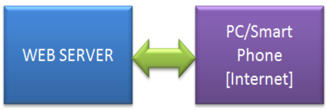


Figure 2: Client Block Diagram

SERVER END: A WEB SERVER is designed and developed for collecting the data from surroundings through Sensors and upload it in to web server. Manual UI is designed for understanding of process with the help of HTML and PYTHON. Using the concept of Internet of Things we are uploading the each individual sensor values to the web server, there I can monitor the sensor values.

III. IMPLEMENTATION

Hardware: In hardware implementation, ARM11 processor (Raspberry Pi) plays a key role in monitoring the system. Low-power consumption ARM11 processor (Raspberry Pi) operating at 3.3-5V, 50uA – 1A is designed and mounted on a PCB along with reset circuit and clock circuit. ARM11, a 32-bit processor with RISC architecture and having 40 GPIO Hedar with 8GB SD Card and 512 Bytes of RAM associated with this raspberry pi.

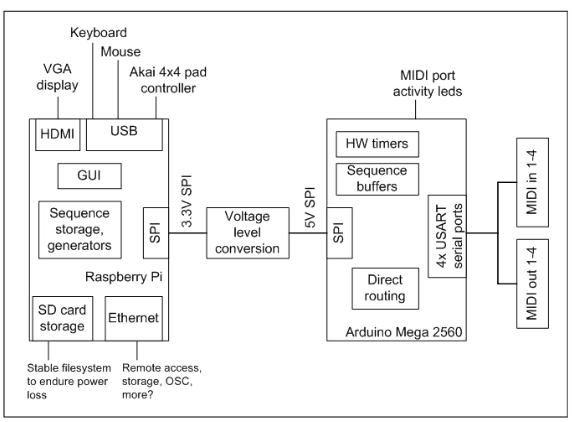


Figure 3: ARM11 Overview

Here, in the figure the 4 Quad USB ports and 10/100 base T Ethernet Socket, CSI Camera Connector, DSI Display Connector and 5V micro USB and HDMI port. We can see 40 GPIO Hedar and the processor on Raspberry Pi is BCM 2835 and with 512MB RAM. With the help of this GPIO Hedar, we can connect sensors/ inputs/ outputs and by using Quad USB we can connect the keyboard and mouse for software programming and also the to interface the other USB peripherals.

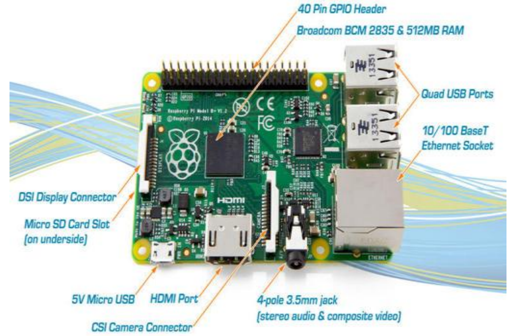
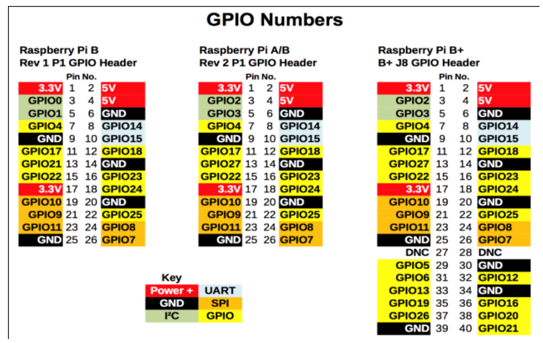


Figure 4: Raspberry Pi Overview

The pin layout of Raspberry Pi is showed below



GPIO Numbers

Raspberry Pi B
Rev 1 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO0	3	4	5V
GPIO1	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO21	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

Raspberry Pi A/B
Rev 2 P1 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7

Raspberry Pi B+
B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Key

Power +	UART
GND	SPI
I²C	GPIO

Figure 5: GPIO PIN LAYOUT

The following is the pin layout of MCP3204 which is used for converting the sensor data to digital.

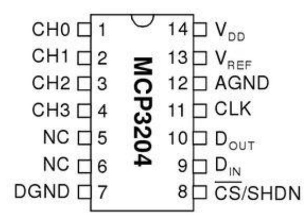


Figure 6: MCP3204 PIN LAYOUT

The following is the sensor which measures the acidity and also the moisture content using a SPDT switch as shown below of the water content dipped. That data will be analysed and given to MCP3204 for converting to digital.

Figure 7: pH and Moisture Level Sensor

The following sensor will give us the conductivity data when dipped into the water and the output of this sensor is connected to the MCP3204 for digitalization.

Figure 8: Water Conductivity Sensor

The water temperature is measured by the following sensor when dipped into the water and the output of this sensor is connected to MCP3204 for monitoring it.

Figure 9: Water Temperature Sensor

The following diagram will illustrate the connection of water level when dipped into it.

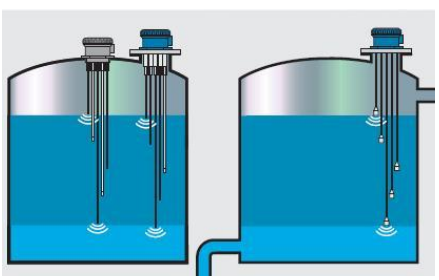


Figure 10: Water Level Concept

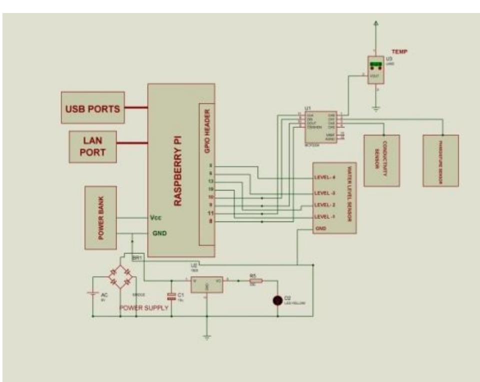


Figure 11: Schematic Diagram

Software: Here, to program ARM processor python -idle was used on Embedded Linux environment. ThingSpeak is an open source Internet of Things application and API to store and retrieve data from things using HTTP over the internet or via a local area network.

IV. ALGORITHM & FLOWCHART

Algorithm

Step 1: Boot Raspberry Pi with valid internet connectivity.

Step 2: Select channel 0 of MCP3204, read data from MCP and store it in a variable.

Step 3: Upload the data to Thing Speak Channel 0 indicated as Field 0.

Step 4: Display it on a log channel as a waveform.

Step 5: Repeat Step – 2 to Step – 4 for the remaining sensors until the power goes off.

Step 6: Now login to the server as a client with a valid username and password.

Step 7: Then sensors page will be opened upon a valid login attempt.

Flowchart:

The flowchart of this paper is shown below

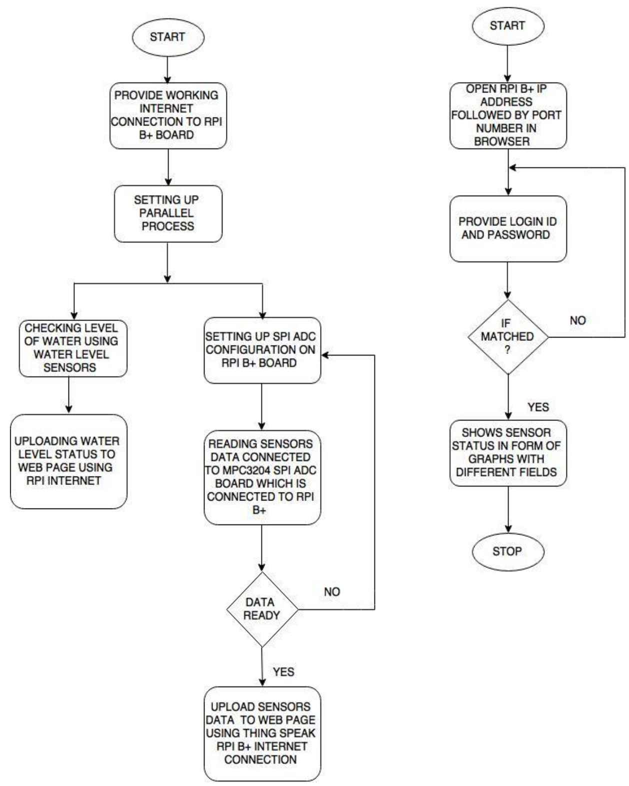


Figure 12: Flow Chart

International Journal of Electrical Electronics & Computer Science Engineering
Volume 2, Issue 4 (August, 2015) | E-ISSN : 2348-2273 | P-ISSN : 2454-1222
Available Online at www.ijeecse.com

V. RESULTS

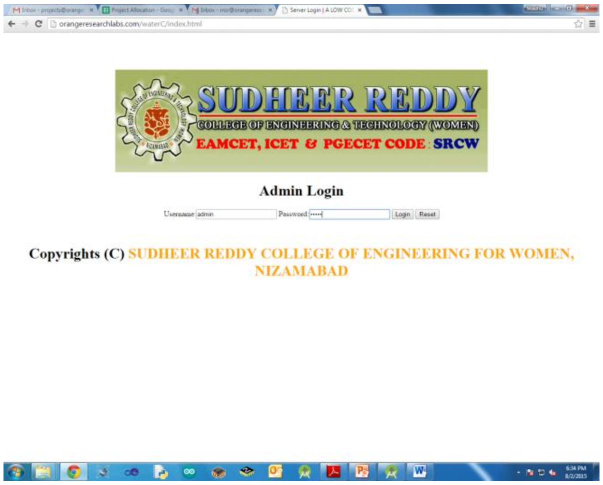


Figure 13: Server Login page

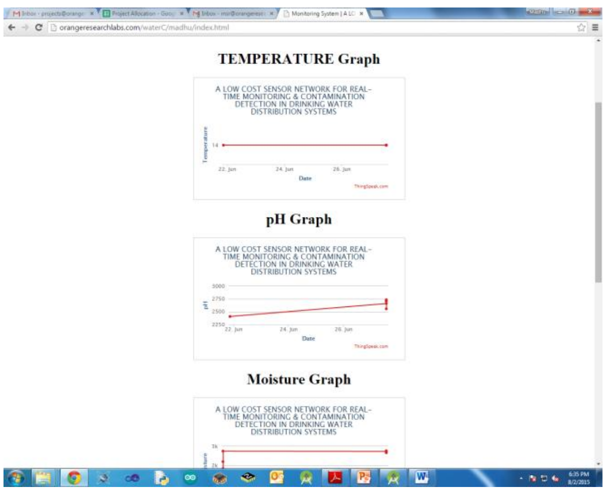


Figure 14: Sensor Upload 1

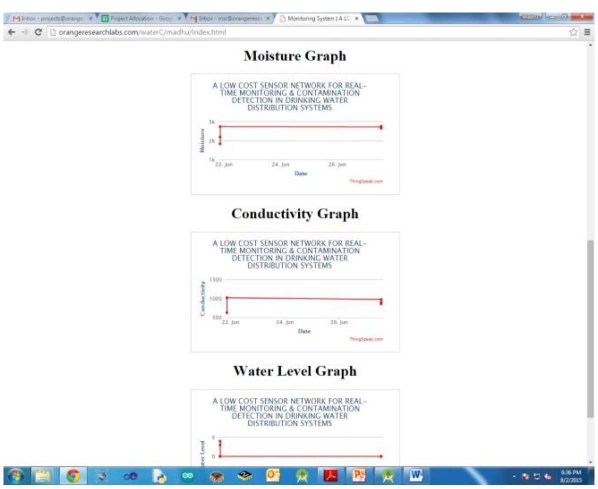


Figure 15: Sensor Upload 2

Figure 16: Final Prototype 1

Figure 17: Final Prototype 2

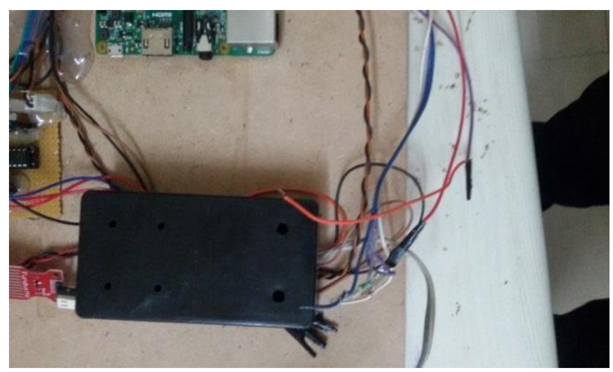


Figure 18: pH Sensor Interface

Figure 19: Power Supply Interface

VI. CONCLUSION

Here, in this paper a simple and secured Internet of Things Sensor Node is designed and developed and proposed. At this node, sensory data can be accessed with a secured login.

ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to Prof. P.Sayanna, HOD, ECE,, Sudheer Reddy College of Engineering& Technology College (Womens).as well as our Prof. Dr. K.S.Srikanth, Principal, Sudheer Reddy College of Engineering& Technology College (Womens). who gave me the golden opportunity to do this wonderful project on the topic (Internet of Things), which also helped me in doing a lot of Research and i came to know about so many new things we are really thankful to them.

And, secondly i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

VII. REFERENCES

[1] http://www.engineersgarage.com/arm-projects/introduction-to-arm-microcontroller-lpc2148

[2] https://www.pantechsolutions.net/microcontroller-boards/user-manual-arm7-lpc2148-development-kit

[3] http://www.nex-robotics.com/lpc2148-development-board/arm7-lpc2148-development-board.html

[4] http://www.ti.com/product/lm35

[5] https://en.wikipedia.org/wiki/Soil_moisture_sensor

[6] https://edis.ifas.ufl.edu/ae437

[7] https://en.wikipedia.org/wiki/Level_sensor

[8] https://en.wikipedia.org/wiki/ThingSpeak

[9] http://www.codeproject.com/Articles/845538/An-Introduction-to-ThingSpeak

[10]http://www.futurlec.com/Philips/LPC2148FBD64pr.shtml