American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-8, Issue-5, pp-182-188
www.ajer.org
Research Paper Open Access

Analysis of Data Communication between Microcontrollers

Md. Emadul Hasan, PallabSahaVaskar and Md. Nur Amin Sajib
(Dept. of Electrical and Electronic Engineering, American International University-Bangladesh)
Corresponding Author: Md. Emadul Hasan

ABSTRACT: *This paper aims to analyze the data transferring between the two most widely used microcontrollers. Those are Arduino mega and Raspberry pi 3B. To transfer the sensor data from Arduino mega to Raspberry pi 3B over a USB cable, there is some limitation. Different kinds of limitation of data transferring between two selected microcontrollers and How to overcome that limitation is discussed in this paper. This data communication is very important for that kind of research and prototype making, which are related to the internet of things and uses two microcontrollers. This paper is also discussed data types and a unique way of continuously sending the data to raspberry pi 3B from Arduino and finally from raspberry pi 3B to the web server. To conduct this analysis four sensors were used. Those were a temperature sensor for analog data, soil-moisture, raindrop and PIR (passive infrared ray) sensor for digital data.*

KEYWORDS*-data communication,data types, sensor network, Arduino mega, Raspberry pi 3B, IoT.*

Date of Submission: 06-05-2019 Date of acceptance: 13-05-2019

I. INTRODUCTION

Next world is moving to its fourth-industrial revolution. One of the most trending technology is the internet of things. This technology will be updated day by day. One of the most important thing in this technology is data communication. Understanding data communication in depth is very important in IoT. As IoT is getting available for the user, Everyday new device will be connected through IoT. Challenges will appear because of the huge amount of data. Process these data and transfer from one device to another is very important. Improper data communication may cause of loss of data. It can be very dangerous in some cases. While the next generation will be dependable on the internet of things, data miscommunication can be harmful to humankind. Another problem appears when a single microcontroller like Arduino used with GSM module for IoT brings a limited opportunity to connect multiple fields with a single device for the same owner. To solve these problems a proper data communication is very important. Developing IoT based research, the researcher may develop a prototype. Most of the prototype contains one or more microcontrollers. Communication between these microcontrollers is very critical for storing proper data in the server, which is a very important term of IoT. This paper focusing on communication between microcontrollers for IoT purpose. To test these communication processes some sensor network were used. Two different microcontrollers with two different programming language used to check their communication process and problems. Finally, by analyzing all these data from different stages of IoT will conclude this paper. In Section II few peer-reviewed articles have been discussed which are related to this work. Section III, Block diagrams of the system and explains different parts of the project. Result and findings of this research described in section IV. Finally, the conclusion part is discussed in section V along with the recommendations for future development.

II. PREVIOUS RESEARCH WORK

There was not much analysis has been found, which was related to the selected microcontrollers. There was some research work represented below which are related to IoT and uses the same microcontrollers.

In [1], authors have discussed many communication protocols of the Internet of things. In addition, they compare commonly used different protocols of IoT.

In [2], authors have implemented a prototype in the lab for monitoring industrial pollution via IoT. In this research, authors have used Arduino mega. Then they send the data to the web server through localhost.

In [3], authors have developed an IoT based irrigation system, for which they have used soil-moisture, temperature and humidity sensor. For IoT, they have used GSM module with Arduino Uno.
The authors of [4] proposed an IoT based energy meter for the fair billing system. For which they have used an Arduino mega with a GSM module for sending data to the IoT server.
The authors of [5] proposed an IoT based irrigation system. For which they have used an ARM microcontroller with GPRS connectivity for sending data to the IoT server.
In [6], authors have designed a smart firming device. In which they have used two microcontrollers Arduino mega and Raspberry Pi 2. They have used raspberry pi 2 as the relay controller and connectivity of the internet from it. They have connected Arduino mega with Raspberry pi 2 by using a USB cable.
In [7], Authors have designed an IoT based intelligent field monitoring system. In which authors have used a soil-moisture sensor and temperature sensor. For controlling actuator, they have used Arduino. Arduino performed sending the data to Thingspeak server with Wi-Fi module.
The authors of [8] have developed smart irrigation and crop security system. Which contains humidity, temperature, Soil-moisture, PIR and float sensor. For controlling the irrigation and security tool and sending the data to the web server Raspberry pi was used.

III. SYSTEMDESIGN AND IMPLEMENTATION

The tested system has divided into three part. First part is sensor network and data collection via Arduino mega. The second part is the most important data sending to raspberry pi 3B from Arduino mega. The final part of this system is sending the data ThingSpeak server. All the hardware and software used, system block diagram and implementation method are discussed below.

A. Hardware requirements

1) Arduino mega

Arduino Mega is a microcontroller device that has 70 pins among which 16 are analog and 54 are digital and each of the pins can be configured as either input or output pin. Arduino mega can be considered a key processing element of this system as it collects data from the sensors and sends it to the Raspberry Pi and activates pump and repeller based on the code assigned.

2) Raspberry pi 3B

Raspberry Pi is a small credit-card sized computer used for basic computing and networking purpose. It is the key component of this system to access to the internet hence the server. First, the sensors provide the required field data to the microcontroller Arduino and thereafter it passes the data to the Raspberry Pi and then it transmits the received data to the server to be stored and viewed by the user. Raspberry Pi model 3B is used here which runs with a quad-core 1.2 GHz Broadcom BCM2837 64bit CPU and has 1GB RAM. It has an upgraded switched Micro USB power source up to 2.5A [9].

3) Soil-moisture sensor

Soil moisture sensor works by assessing the dielectric permittivity of the soil thus the moisture level. The sensor gives a logical high voltage when the moisture level is low and gives a logical low voltage when the moisture level is high. It operates between 3.3v- 5v.

4) Raindrop sensor

The rain-drop sensor is used for rain detection. Its power supply voltage ranges from 3.3v to 5v. It has RF-04 double-sided material with an area of 5cm*4cm. The digital output pin produces a logical High when there is no raindrop and logical Low when raindrop is detected on the surface of the sensor.

5) Temperature sensor (DS18B20)

Temperature sensor measures the temperature using an electrical signal. It uses a converter to transform the temperature value to an electrical value. The thermocouple sensor is used here to detect the irrigation water temperature.

6) Passive infrared ray (PIR) sensor

PIR sensor identifies the movement of any object passing near the sensor by detecting the level of infrared radiation as every object emits minimum radiation. Its detection range is usually up to 6 meters. As soon as the PIR sensor detects the presence of an object the repeller gets activated and starts functioning.

B. Software Requirement

1) Raspbian OS

Raspbian is a Debian based open source Linux operating system optimized to be used in Raspberry Pi. This operating system comes with some basic preinstalled programs and utilities to run the Raspberry Pi hardware. It provides the user with access to over 35000 precompiled software that is packed in a nice way for convenient installation on Raspberry Pi. [10]

2) Arduino IDE (Linux system)

Arduino integrated development environment or Arduino IDE is cross-platform software used for writing codes and uploading it to the Arduino board. The Arduino IDE comes with a software library provided by the Wiring project that has many common input and output methods.

3) Python IDLE 3.5

Python IDLE 3.5 is a built-in feature in the Raspbian operating system. This feature gives access to the user to run any kind of pre-prepared programme in python 3. It is a complete compiler and interpreter.

4) MobaXterm

MobaXterm is a windows software, which gives the advantage to the windows user to use laptop display, keyboard, and mouse as Raspberry Pi's monitor, keyboard and mouse. It's a completely virtual environment to control raspberry pi via laptop with ssh connection. It also provides access to use laptop internet connectivity to Raspberry Pi.

C. Web server

1) Thingspeak server

Thingspeak is a web server for practicing IoT project and. Where an API key produces by the system. With that data can be sent and receive from any device. Usually Thingspeak store one-year data for free.

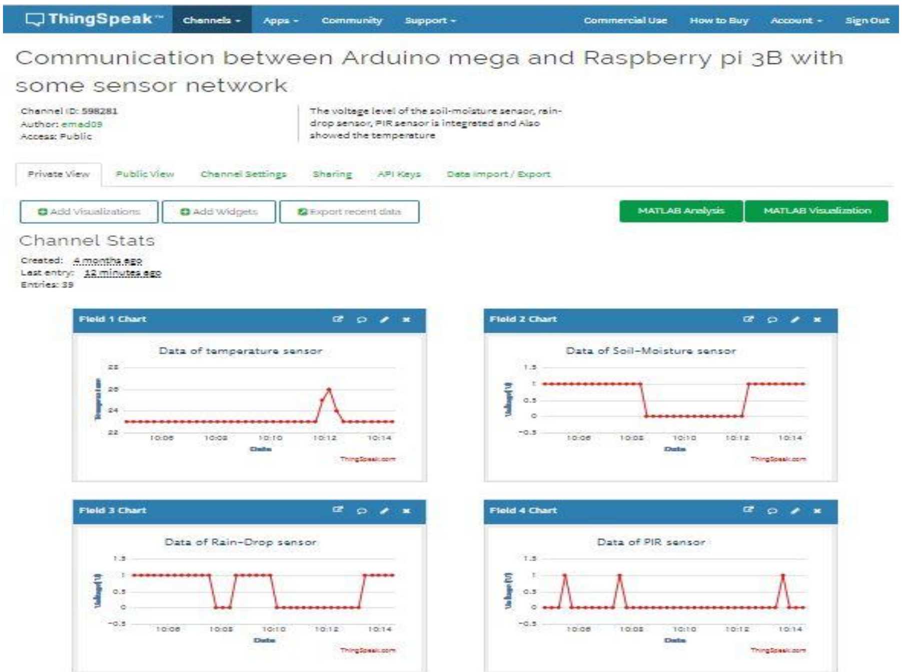

Fig. 1. User interface after logging into the ThingSpeak server

In "Fig. 1", the user interface of thingspeak server after login is represented. This is the private view of a channel.

D. Block diagram and implementation

There is some crucial part of this system. Those crucial terms of hardware implementation and programming are discusses below.

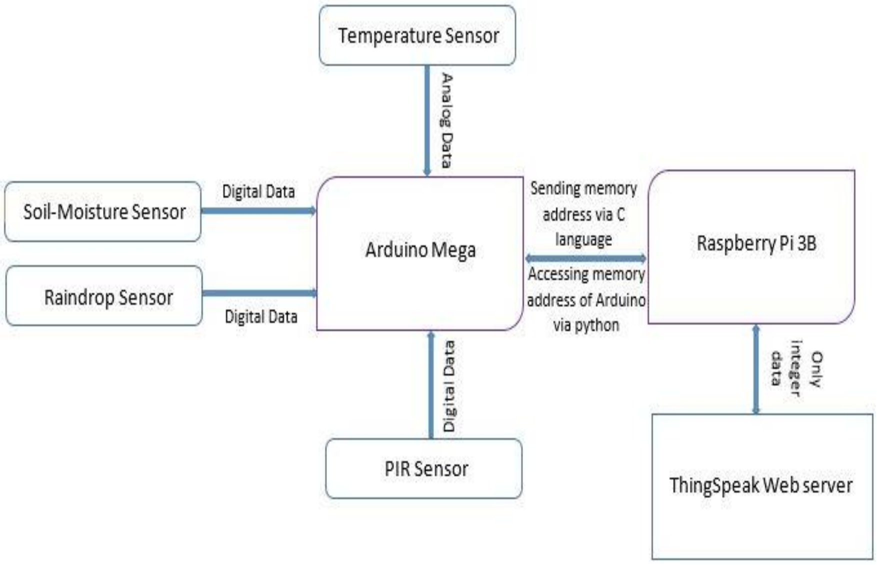

Fig. 2. Block diagram of the tested system

In the tested system, four sensors (soil-moisture, raindrop, PIR and temperature sensor) was connected to Arduino mega. A 1.2kΩ resistor must be connected between the data pin and Vcc of the temperature sensor to avoid the garbage value. Integer type variable was taken to receive the data from the sensor. Using other data type was causing some problem for this type of approach. Which will be discussed in the result and findings section All the sensor data memory was stored in a byte type data array. Which will contain the data memory address. This entire programme in Arduino was written in modified version of C-language. This memory address was sent to raspberry pi 3b. Which is pre-programmed with python 3. Which was connected to Arduino with a USB cable. Python serial was preinstalled in the raspberry pi. A python programme was run in the python shell 3.5.3. By the python programme memory address of the sensor stored data in the Arduino mega was received and by accessing the memory data sensor data was collected in the raspberry pi. The same python programme performed collecting the sensor data and sending those data to the web server simultaneously using API key produced by ThingSpeak web server. All the data of the sensor was taken in outdoor.

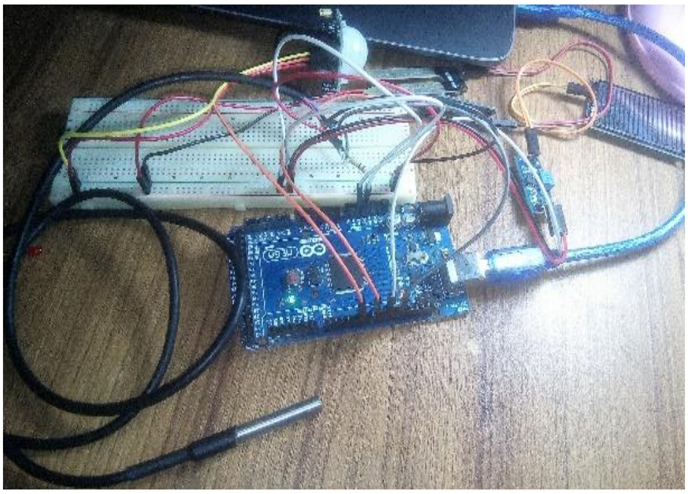

Fig. 3. Implemented prototype view in indoor

IV. EXPERIMENTAL RESULTS

After the successful implementation of the prototype, the whole system was run. Some problem arose and their possible findings, the solution is discussed below.

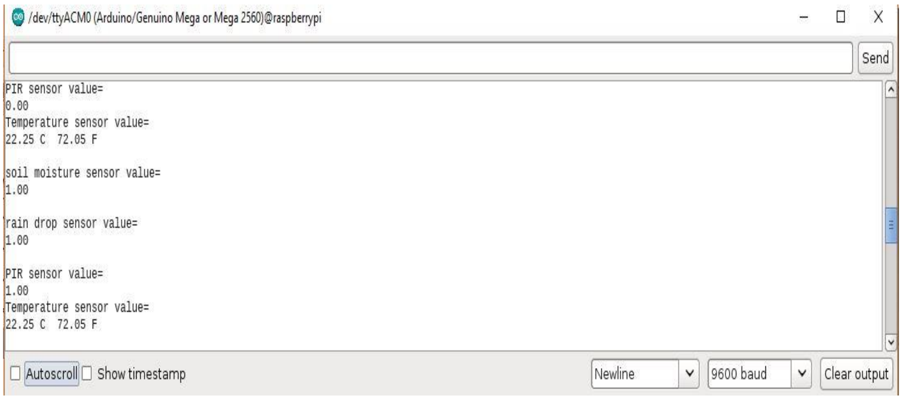

Fig. 4. Sensor Data read from Arduino mega serial monitor

"Fig. 4", data reading of sensor from the serial monitor of Arduino mega was represented. This serial monitor was used from the Arduino IDE software. Until this step, there was no problem with data type in the programming of Arduino.

Fig. 5. Received sensor data in Raspberry Pi (View on Python shell 3.5.3)

In "Fig. 5", data received in Raspberry pi from Arduino mega is represented. In this step, there was a limitation. Only integer number is received but no floating point number is received. If any float type variable used in the Arduino then raspberry pi will getting garbage value. To solve this problem only integer type variable was used. This problem only occurs when there is some floating point number. Except this, there was no problem with the integer data.

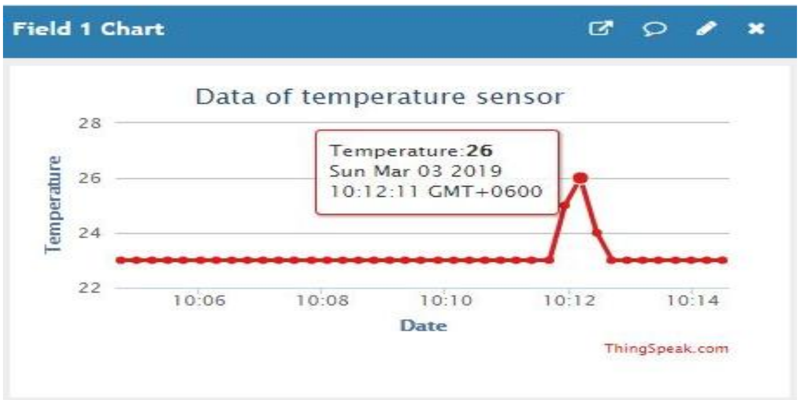

Fig. 6. Temperature sensor data in thingspeak server

In "Fig. 6", temperature sensor data along with time and specific date is represented. There are also some points, which indicates the previously stored data.

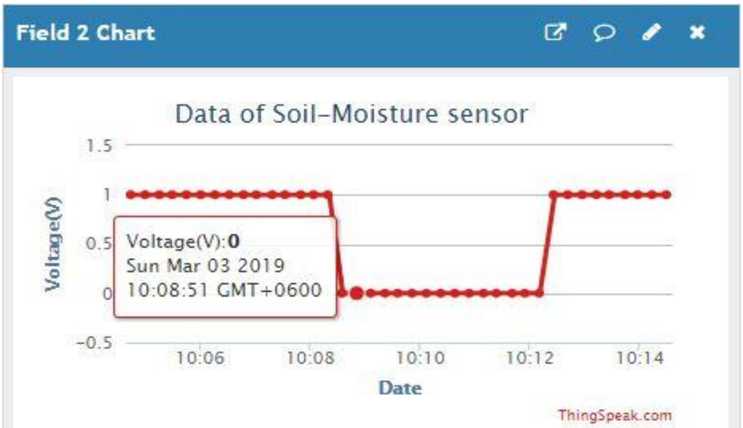

Fig. 7. Soil-moisture sensor data in Thingspeak server

In "Fig. 7", soil-moisture sensor data in the ThingSpeak server is represented with voltage vs date graph. The server automatically generates the graph. Where for low moisture level it produces high or "1" logical voltage and for high moisture level, it produces low logical voltage.

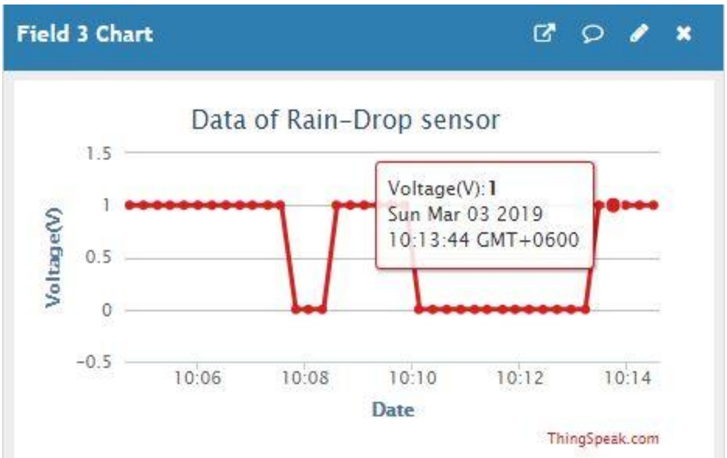

Fig. 8. Raindrop sensor data in Thingspeak server

Raindrop sensor data in Thingspeak server is represented in "Fig. 8", where for dry condition sensor produces high logical voltage and when there was rain detected it produced low logical voltage.

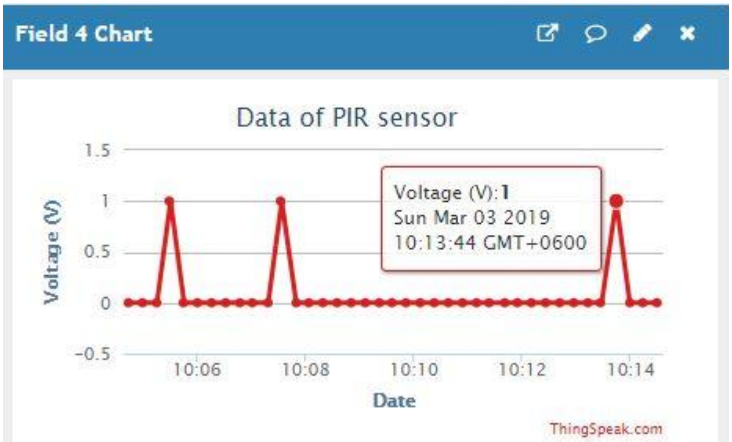

Fig. 9. PIR sensor data in Thingspeak server

PIR sensor data in Thingspeak server is represented in "Fig. 9", where for detection of movement of any object PIR sensor produces high logical voltage and when there was no movement detected it produced low logical voltage.

From this research, it is very clear there is a limitation while data is sent to the raspberry pi from Arduino mega. Only integer type variable may use in the Arduino programming. There is also some time delay in the process. The time delay is about five seconds. For sending the data in the web server there are about 20 seconds of delay. However, this delay may vary depending on the speed of internet connectivity.

V. CONCLUSION

This analysis emphasis on data communication between Arduino mega and raspberry pi. Accessing memory of Arduino mega via raspberry pi was also tested. One major limitation was found with the data type while raspberry pi accessing memory of Arduino. This limitation can be erased by multiplying 100 with the float data and making it an integer and send it to the Raspberry Pi 3B. After getting the data in raspberry pi then divide the data by 100 will make the data float again. In this way, floating point number data can be transfer. There is some issue of time delay. Which may vary with two parameters. One is the connection of internet and another is processor speed. But this approach will reduce the cost as multiple Arduino can be used for multiple purposes and connecting them together to a Raspberry Pi to sending or getting the data from the server will reduce the cost of IoT. Building wireless communication between these two microcontrollers may test in the future.

REFERENCES

[1]. S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of Things (IoT) communication protocols: Review,” 2017 8th International Conference on Information Technology (ICIT), 2017.

[2]. K. B. Swain, G. Santamanyu, and A. R. Senapati, “Smart industry pollution monitoring and controlling using LabVIEW based IoT,” 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), 2017.

[3]. M. N. Rajkumar, S. Abinaya, and V. V. Kumar, “Intelligent irrigation system — An IOT based approach,” 2017 International Conference on Innovations in Green Energy and Healthcare Technologies (IGEHT), 2017.

[4]. O. Jaiswal and D. Chaubisa, “Arduino mega and IOT based intelligent energy meter (IEM) to increase efficiency and accuracy in current billing methodology,” 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017.

[5]. V. Ramachandran, R. Ramalakshmi, and S. Srinivasan, “An Automated Irrigation System for Smart Agriculture Using the Internet of Things,” 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2018.

[6]. J. D. Stevens and T. Shaikh, “MicroCEA: Developing a Personal Urban Smart Farming Device,” 2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC), 2018.

[7]. M. Ashifuddinmondal and Z. Rehena, “IoT Based Intelligent Agriculture Field Monitoring System,” 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018.

[8]. R. R. Agale and D. P. Gaikwad, “Automated Irrigation and Crop Security System in Agriculture Using Internet of Things,” 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017.

[9]. “Raspberry Pi 3 Model B,” Rotate display 90º? - Raspberry Pi Forums. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed: 05-Mar-2019].

[10]. “Welcome to Raspbian” www.raspbian.org [Online] Available: https://www.raspbian.org/[Accessed: 05-Mar-2019].

Md. Emadul Hasan" Analysis of Data Communication between Microcontrollers"American Journal of Engineering Research (AJER), vol.8, no.05, 2019, pp.182-188