International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Wind Turbine Data Collection using IoT

Stanly Wilson, V.B. Kirubanand

Abstract: Renewable energy resources have gained an important place in human life. It is a fact that the amount of non-renewable energy resources is decreasing rapidly, at the same time, the need for renewable energy resources increases. The history of humanity teaches us to use the resources prudently and efficiently. This can happen only with well planning and devising ways to realize the same. Even in the case of efficient use of renewable energy, the data analysis has greater value, since better the analysis, better the efficient and effective use. This paper is an outcome of making the data analysis of a wind turbine by collecting and communicating the data to the cloud server. These details can be accessed locally with the use of an LCD module, and remotely with the web browser or through a mobile app.

Keywords**:** ***Arduino, Cloud Server, IoT, Transmission, Wind Turbine.***

I. INTRODUCTION

The energy in its various forms is inevitable to human life. It has taken its turn in various ways in human history and it will continue in the future too. The forms may change from time to time. In the earlier time burning wood was the source of energy. Later during the industrial revolution coal was used as the source of energy. Though the forms were different, basic was the fire in both cases. There are different sources of energy and they could be classified as renewable and non-renewable sources of energy. The limited availability of the non-renewable sources of energy has made the world to think and move towards renewable energy resources.

There are various renewable resources like solar, wind, hydro, tidal and others. Among these, the paper keeps its focus on the wind. The wind is a clean source of energy as it does not pollute the air and the environment. It has a significant role among other renewable energy sources. Wind energy is not a new form of energy for humans. The sailors have been using it for ages to travel in the water. In the land, it was used to pump water and to grind the grains. Wind energy has its source in solar energy, and so as long as Sun will be there that long wind energy will also remain [1].

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author
Stanly Wilson*, Computer Engineering, St Vincent Pallotti College of Engineering & Technology, Nagpur, India.
Dr. V.B. Kirubanand, Associate professor, Computer Science, Christ (Deemed to be University), Bangalore, India.

In modern times, the interest of wind as a major source of energy started when the world realized the limited supply of non-renewable energy resources and its highly fluctuating cost. The energy generation from wind can be both onshore and offshore. It is predicted that in the near future, the energy generation from offshore would exceed that of onshore power production [2]. The better utilization of wind energy comes with enhanced forecasting and for that, it is important to have data comes from the wind turbine [3].

The paper attempts to bring a solution to the data acquisition from the wind turbines. The model does not use any sensors. It just collects the data and sends to the cloud server for the purpose of remote monitoring and better analysis. The paper has the following sections. The proposed methodology brings out the working of the model with the help of the data flow diagram. The components that are employed would be discussed individually under the section component description. The deployment section brings out how the system works and the methods that are used. The results and discussion part shall analyze the outcome of the whole work. The conclusion proposes some applications and possibilities for future enhancements.

II. PROPOSED METHODOLOGY

There are five parts to this model. First is the collection of data. There are various sensors in the wind turbine. These sensors send the data in every microsecond and these values reach the data collector. The purpose of this model is to collect these values that reach the data collector and send them to the cloud server. The second part is data acquisition. It has two components. They are RS485 and Arduino Mega 2560. The data that reach the collector needs to be taken into the Arduino. If only it reaches the Arduino, they can be sent to the cloud server, and perform analysis on it. In order to do that RS485 interface is used. RS845 acts as an interface between the data controller and Arduino Mega. The third part is data storage. The components that come in this module are SD card and the cloud server *thinger.io* where the data will be stored. The data that arrive in the Arduino will be attached with a timestamp first. This will enable people to understand the time when the data recorded. The data which are acquired from the wind turbine are stored on the SD Card with a timestamp and at the same time, they are sent to the cloud server database *thinger.io*. Forth part is communication. The data need to be sent to a cloud server which is an online database that is used for this project. GSM module is used to send the data to the cloud server. This makes the remote access of the data possible. The fifth part is the LCD display.

The data stored in the cloud server could be seen with the web browser as well as the mobile app. There may be situations when someone wants to see the values that flow from the wind turbine and may not have access to the cloud server. There comes the use of the LCD screen. It is used to display some specific values that flow from the turbine. These values will be updated every 2 seconds. Figure 1 represents the data flow diagram.

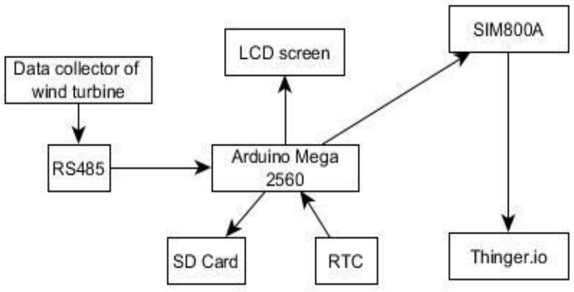


Fig. 1. Data flow diagram

III. COMPONENTS DESCRIPTION

The paper uses some components for achieving the desired outcome. The components gather the details that come to the controller and visualize them. The components used here are Arduino Mega 2560 as a microcontroller, SIM800A for the data transmission, LCD to display the details in the model itself, RS485 as a bridge between the controller and the Arduino Mega, DS3231 as a real-time clock and SD card to store the data in case of emergency.

A. Arduino Mega

Arduino Mega is a microcontroller, means a small (micro) device that controls devices and processes the data. It has 16 analog pins, 54 digital input/output pins, and 4 dedicated UART (universal asynchronous receiver/transmitter) ports. It has a memory of 256KB for the program while 4KB is allocated for the global variables. It is an AtMega2560 based microcontroller. The power to the board could be given directly or with the help of USB. The benefit of using Arduino Mega over Arduino Uno is the possibility of having the availability of more pins and memory, though Arduino Uno is cheaper. The paper makes use of the Mega board due to the requirement of more memory and pin options [4][5].

B. Sim 800A

This module supports GSM as well as GPRS. Here GSM gives the possibility for oral and text communication. Oral communication refers here is the voice call while text communication refers to SMS. GPRS gives the possibility of accessing the internet using this module. It supports communications on the frequencies of 850, 900,1800 and 1900. Its power consumption is quite less. This module has an RS232 interface which makes it compactable to use even with a computer [6][7].

C. LCD

It is a flat electronic display panel that uses the light modulation characteristics of the liquid crystals. The paper makes use of a 16x2 display that is very basic and widely used in many devices. It can represent words, numbers, and works similar to a seven-segment display. The benefits of LCD over the seven-segment display are the programmable capacity of LCD, less costly and it can even represent animations. The model 16x2 can represent 16 characters in a single line and it can have 2 lines, which means a maximum of 32 characters at a given moment of time. Each character is made up of a 5x7 pixel matrix. The LCD can be easily programmed in such a way that each character can be individually accessed and can display what is given by the program as output. It has 16 pins and most of which are required to connect to the board to get the desired display [8].

D. RS485

It is an interface unit based on the MAX485 chip. It has the receiver/transmission capabilities. It can convert data into a stream of serial data. RS485 is capable of half-duplex communication while this paper uses only the simplex transmission protocol. That means data flow to only one direction. Its medium and long-distance communication capabilities make it one of the widely used serial bus. Its predecessor RS232 could communicate a distance as small as 20 meters while RS485 can make it up to 1000 meters. RS485 comes as a good choice to have a reliable and safe real-time data acquisition systems [9][10].

E. RTC and SD Card

RTC stands for real-time clock. The module used here is DS3231. Its purpose is to provide the current date and time. It has a battery attached to it which makes RTC work even during the power failure or while the system is off. It needs to be programmed once by having the current date and time. Then it works like any other timing device providing the date and time. It is very useful in data analysis. SD card is used for the purpose of storing the data. Whenever the data comes to the microcontroller, with the timestamp from RTC, it will be written to the SD card too. This is useful when there is some trouble to communicate with the cloud server. The data will be in the SD card even if it does not reach the server [11].

IV. IMPLEMENTATION

The wind turbine that collects the values for this paper is located in Bangalore city, Karnataka, India. The sensors attached to the wind turbine collects the data and send them to the controller. RS485 collects these details and give them to the microcontroller. Figure 2 shows the circuit diagram of the proposed model.

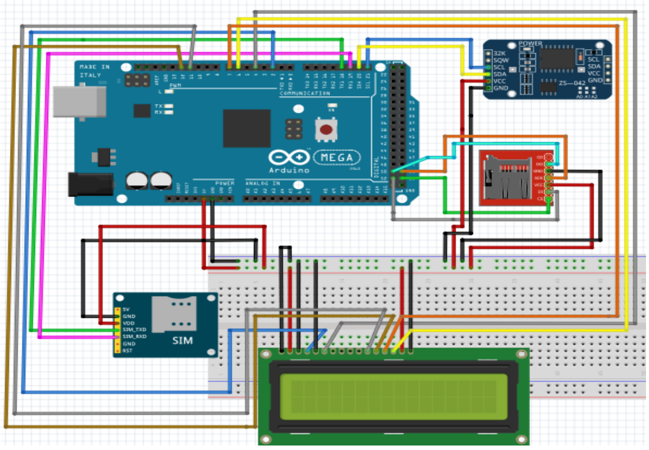


Fig. 2. Circuit diagram

Figure 3 shows the flow of the program and the process is as follows. Define the header files and the global variables. Set the cloud server details like the username, device id and credentials, and the APN for the GSM module. In the setup() initialize the SD card. If SD card is present the values are saved into the SD card else it indicates SD card error. Next, initialize RTC and set the values and start LCD. The code that sends the data to the cloud server database starts from here. So, the define database values and variables here. The column names are assigned to the variables and they are sent to *thinger.io* for each iteration. The loop() runs indefinitely. The displayTime() takes the values from the RTC and performs the conversions and returns the time, day and date.

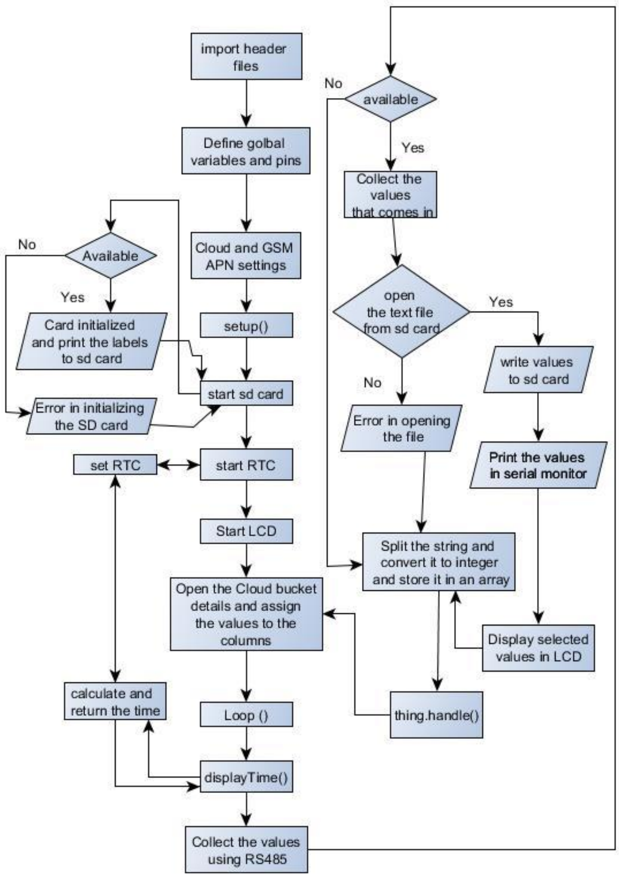


Fig. 3. Flow chart

RS485 acts as a bridge that collects the values from the controller. If there is any value that flows from the controller to RS485, then that value is taken in. It checks for the file to be written in the SD card. If the file is present, it writes the values to the SD card and then displays in the LCD screen those selected values. Whether a text file is present or not, if any values are received, they are to be processed. The data received will be a string of values. In order to store it in a tabularized manner, they need to be split. The split values are stored in an array which is a global variable. The function thing.handle() takes the control to the area in setup() where the values are sent to a cloud server. The loop function will be executed until an interrupt occurs.

V. RESULTS AND DISCUSSIONS

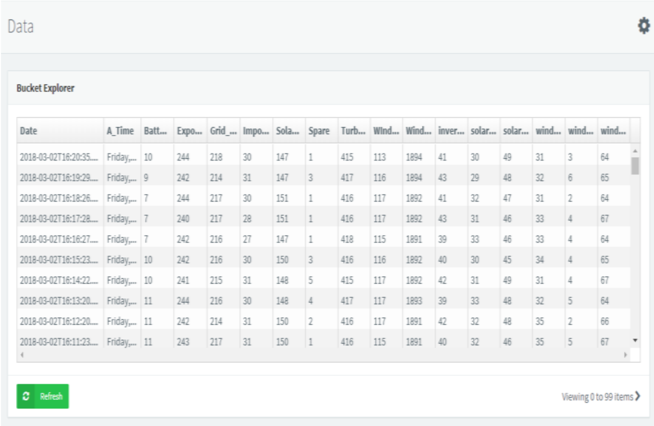
The database and related things are provided by *thinger.io* itself. The database in *thinger.io* is called a bucket where the values are stored. There is no restriction about the datatype, i.e., the data can be integer, character, Boolean, floating-point or a string. It takes any type of data and need not specify it anywhere in *thinger.io* during the creation of the fields. There are some prerequisites to send the values to *thinger.io*. One needs to register to the cloud server and those details need to be provided for the program. These details are a username, device id, and device credentials. These along with the APN details of the SIM communicate the details to the server. Figure 4 shows the required fields in the program.

```
#define USERNAME ""
#define DEVICE_ID ""
#define DEVICE_CREDENTIAL ""

// use your own APN config
#define APN_NAME ""
#define APN_USER ""
#define APN_PSWD ""
thing["Data"] >> [](pson & out)
{
    out["A_Time"] = timeStamp;
    out["wind voltage"] = a[0];
    out["wind speed"] = a[1];
    out["solar voltage"] =a[2] ;
    out["inverter current"] = a[3];
    out["wind current"] =a[4] ;
    out["Turbine_Rpm"] = a[5];
    out["Grid_voltage"] = a[6];
    out["Battery Current"] = a[7];
    out["solar current"] = a[8];
    out["Spare"] =a[9];
    out["Export_unit"] = a[10];
    out["Import_unit"] =a[11] ;
    out["WInd_KWHR"] =a[12] ;
    out["Solar_KWHR"] =a[13] ;
    out["Wind power"] =a[14] ;
};
```

Fig. 4. Data transmission function

The user name is the registered username, the device name is the name of the device that sends the data. The device name is necessary to identify uniquely the device that sends the data. Device credentials are generated during the creation of the device. These values can uniquely identify the user and the device from where the data come from. One device can have many buckets, i.e., the same device can send data to a different bucket at the same time. The bucket is specified in 'thing[ ]'. In the above code, the name of the bucket which takes the values is 'Data'.



Date	A_Time	Batt...	Expo...	Grid_...	Impo...	Sola...	Spare	Turb...	Wind...	Wind...	inver...	solar...	solar...	wind...	wind...	wind...
2018-03-02T16:20:35....	Friday,...	10	244	218	30	147	1	415	113	1894	41	30	49	31	3	64
2018-03-02T16:19:29....	Friday,...	9	242	214	31	147	3	417	116	1894	43	29	48	32	6	65
2018-03-02T16:18:26....	Friday,...	7	244	217	30	151	1	416	117	1892	41	32	47	31	2	64
2018-03-02T16:17:28....	Friday,...	7	240	217	28	151	1	416	117	1892	43	31	46	33	4	67
2018-03-02T16:16:27....	Friday,...	7	242	216	27	147	1	418	115	1891	39	33	46	33	4	64
2018-03-02T16:15:23....	Friday,...	10	242	216	30	150	3	416	116	1892	40	30	45	34	4	65
2018-03-02T16:14:22....	Friday,...	10	241	215	31	148	5	415	117	1892	42	31	49	31	4	67
2018-03-02T16:13:20....	Friday,...	11	244	216	30	148	4	417	117	1893	39	33	48	32	5	64
2018-03-02T16:12:20....	Friday,...	11	242	214	31	150	2	416	117	1891	42	32	48	35	2	66
2018-03-02T16:11:23....	Friday,...	11	243	217	31	150	1	416	115	1891	40	32	46	35	5	67

Fig. 5. Cloud storage

Here the values are sent to the Cloud using the 'out' method provided by the cloud server. The different fields are mentioned inside the program will be created inside the bucket when the data reaches the cloud server and continues to store it when the corresponding data reaches in *thinger.io.* Figure 5 shows the stored values in the database.

The 'date' shows the time and date of the arrival of the program and A_time gives the time generated from the RTC that is attached to the data when it is sent from the microcontroller. There are fifteen values that could be seen in the database other than the time. The purpose of this paper is to make this the remote monitoring of these details possible. The visualizations of the same could be done with the dashboard in the cloud server. Figure 6 shows the visualization of various paraments in the database. Depending on the use, one could change the style of these graphs. These are very much customizable depending on the need and requirements of the people.

Fig. 6. Data visualization

VI. CONCLUSION

The paper made an attempt to collect data from a wind turbine using IoT. This system uses a GSM module which gives the benefit of placing it in remote areas where there is less possible to have the WiFi. The study of the data received will help to understand the seasons when energy production is higher and lower. The paper could be enhanced by implementing machine learning techniques to understand different patterns of energy production. This would help to make predictive analysis that may enable better use of energy.

REFERENCES

1. Panwar, N. L., S. C. Kaushik, and Surendra Kothari. "Role of Renewable Energy Sources in Environmental Protection: A Review", *Renewable and Sustainable Energy Reviews*, vol. 15, no. 3, 2011, pp. 1513-1524.
2. Hameed, Z., J. Vatn, and J. Heggset. "Challenges in the Reliability and Maintainability Data Collection for Offshore Wind Turbines", *Renewable Energy*, vol. 36, no. 8, 2011, pp. 2154-2165.
3. Lei, Ma, Luan Shiyan, Jiang Chuanwen, Liu Hongling, and Zhang Yan. "A Review on the Forecasting of Wind Speed and Generated Power", *Renewable and Sustainable Energy Reviews*, vol. 13, no. 4, 2009, pp. 915-920.
4. Zeebaree, S. R., and Hajar M. Yasin. "Arduino Based Remote Controlling for Home: Power Saving, Security and Protection", *International Journal of Scientific & Engineering Research*, vol. 5, no. 8, 2014, pp. 266-272.
5. Tariq, AL-Kadi, AL-Tuwaijri Ziyad, and AL-Omran Abdullah. "Arduino Wi-Fi Network Analyzer", *Procedia Computer Science*, vol. 21, 2013, pp. 522-529.
6. Harikrishnegowda, G., K. Kavyashree, G. Jayakumar, and R. S. Supriya. "IoT Enabled Carbon Dioxide and Carbon Monoxide Monitoring and Control for Vehicles to Reduce Air Pollution", *Perspectives in Communication, Embedded-systems and Signal-processing-PiCES*, vol. 1, no. 6, 2017, pp. 104-107.
7. Wilson, Stanly, Tony Manuel, Peter Augustin D, "Smart Pollution Monitoring System", *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, 2019, pp. 1131 - 1136
8. Goud, E. Harshavardhan, A. Harshika, G. Akhil, D. Charishma, K. Bhupathi, and I. Kumara Swamy. "Real-Time Based Temperature Control Using Arduino", *International Journal of Innovations in Engineering and Technology*, vol. 8, no. 2, 2017, pp. 209-215.
9. Min, Dong, Zeng Xiao, Bi Sheng, and Guo Shiya. "Design and Implementation of the Multi-Channel RS485 IoT Gateway", In *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2012, pp. 366-370, IEEE.
10. Shi, Lei, and BaoLong Guo. "RS485/422 Solution in Embedded Access Control System", In *2009 2nd International Conference on Biomedical Engineering and Informatics*, 2009. pp. 1-4, IEEE.
11. Mahajan, Sagar A., and S. D. Markande. "Design of Intelligent System for Indoor Lighting", In *2016 International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2016, pp. 1-4. IEEE.