

Remote tending of modern broadcast transmitters

Gerino Mappatao, Isaiah Mari Zebedia Bautista, Marc Kevin Orsos, Mark Aldrine Ribo,
Joseph Castillo

Department of Electronics and Communications Engineering, De La Salle University, Philippines

Article Info

Article history:

Received Jan 20, 2019

Revised Mar 30, 2019

Accepted Apr 30, 2019

Keywords:

Remote metering

Remote monitoring

Transmitter control

Transmitter monitoring

Transmitter monitoring

ABSTRACT

Once broadcast transmitters are put into operation, regulatory bodies in different countries require the equipment to be monitored, data logged and controlled. Normally, transmitter personnel turn ON the transmitter at the start of the broadcasting activities, and turn it OFF at the end of the activities. They also regularly record the transmitter meter readings on output power, voltage and current as long as the transmitter is on-air. These data must be logged and saved for future references. These procedures are done by transmitter personnel manually. With the advent of internet technology, this paper presents a system to remotely monitor, gather and record meter readings, and a provision to control the turning ON/OFF as well as to adjust the output power of a number of digital transmitters. All these procedures are done in one central location. The system was implemented and then tested using a digital television transmitter. Results of the tests show that the developed system capable of performing the functions of a transmitter technician in monitoring, logging and control of broadcast transmitters remotely. With the use of the system, broadcast networks can reduce operating costs in tending their transmitters but still able to comply with the requirements.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Gerino Mappatao,

Department of Electronics and Communications Engineering,

De La Salle University,

2401 Taft Avenue, Malate, Manila, Philippines.

Email: gerino.mappatao@dlsu.edu.ph

1. INTRODUCTION

The telecommunications regulatory body of the Philippines, in particular, requires all broadcast transmitters in the country to continuously monitor and log transmitter meter readings for as long as the transmitter is on-air [1, 2]. Transmitter logs are stored for at least two years for future references. In a normal setting, transmitters are manned by technicians doing the monitoring and logging. They also turn ON the transmitter before the start of the first program of the station; they turn OFF the same at the end of the last program or at any instance as the need arises, especially on emergency situations. Data to be logged, as required by the government, are the times the transmitter was turned ON and turned OFF, and the voltage and current of the final stage of the transmitter. The voltage and current readings are recorded every 30 minutes. On top of these, technicians are also required to log any problem encountered during the operation of the transmitter, if there is any. If the operation of the transmitter is interrupted due to malfunctioning and it needs to be shutdown, the incident should also be placed in the log. Likewise, the time the transmitter was put back in operation are also logged. These technician functions in multiple transmitter sites are aimed of being accomplished by the proposed system remotely at a particular central monitoring location.

There are basically two types of transmitters in the Philippine broadcasting industry: the modern and the analog transmitters. These terms are used in this paper to differentiate one from the other. The former have RJ45 ports in their exciter and power amplifier that allow computers to have direct access to it. Aside

from that, the exciters also have web interfaces that show transmitter status and parameter values that could be used for monitoring and control of the transmitter. The exciter of the transmitter uses an LCD (Liquid Crystal Display) in displaying the meter readings. On the other hand, analog broadcast transmitters do not have neither RJ45 port nor web interface. LED (Light Emitting Diode) or LCD displays are used in these transmitters to display meter readings. The proposed system presented in this paper uses only the modern type of transmitters, particularly the Anywave® television transmitter. Aside from the exciter, Anywave transmitters have a Power Amplifier (PA) composed of several amplifier modules.

The system set-up of the proposed system is illustrated in Figure 1. The system is composed of a single CMC (Central Monitoring Computer), several transmitter servers and a GSM module. The CMC is a computer that receives data from the transmitters, sends command signals to the transmitters, stores transmitter data as well as prints the same if so desired. Data from each transmitter are transmitted to the CMC via the internet on a regular basis. The communication between the CMC and the transmitters are made possible by the installation of a server at each transmitter location. The server is a cheap computer laptop that communicates with the CMC for the transfer of data from both directions: transfer of data from transmitter to CMC and reception of control signal from CMC to transmitter. The GSM module allows the sending of SMS (Short Message Service) to alert the tender at the CMC of any eventuality that needs prompt attention.

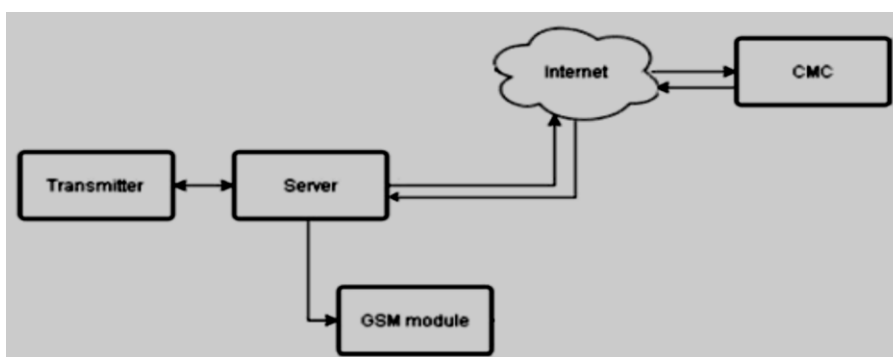


Figure 1. Set-up of the proposed system for the remote monitoring, logging and control of modern transmitters

Each transmitter sends its data every 30 minutes to the CMC. In the proposed system, the voltage, current, SWR (Standing Wave Ratio) and output power readings are transmitted. If a very high SWR reading is detected, the system sends an SMS to the CMC to alarm the technician.

A similar system is proposed in [3] but using an embedded system and the functions are limited to the control and monitoring of transmitter status. It lacks the logging feature which make it ineligible to meet all regulatory requirements. Methods of implementating a remote monitoring system that are applicable to the proposed system are discussed in [4] and similar systems are discussed in [5-8], especially with the use of smart control algorithms.

2. SYSTEM DESIGN

The features and operation of the proposed system in the transmitter side are illustrated in Figure 2. The system in the transmitter side initially gathers transmitter data and then stored locally. Normally, these data are also transmitted automatically and regularly every 30 minutes to the CMC. However, when data are not received successfully by the CMC, e.g. when there is temporary interruption in internet connection, the CMC sends requests for retransmission by sending the last data entered in its log. The transmitter side then sends all other missing (unreceived) data up to the current data stored in the remote side. The value of the SWR is also being checked periodically, whether it is within the acceptable of 1.0 to 1.8. When the value exceeds 1.8, the system in the transmitter side sends an alarm in the CMC as well as an SMS to a particular cellphone number with the use of a GSM module.

To send data from end to another, and vice-versa, the computers in the transmitter and CMC must be connected through a network. The network connection of the proposed system is shown in Figure 3. To connect the server in the transmitter site to the CMC using the internet, a Virtual Private Network between the CMC is created using Hamachi instead of using a router with port forwarding [9].

The transmitter locations are equipped with pocket Wi-Fi, no routers are installed so the only way to exchange data between the server and the CMC is to use the IP Address assigned to the computer by the Hamachi network. The network connection acts as if the CMC and the server are connected to a Local Area Network, but using the internet. Each computer at the remote location is installed with XAMPP so that Apache and MySQL can be used for the database. The transmitter used in the prototype has an exciter and a power amplifier that have built-in web interfaces. Using a web browser, URLs are used to access the exciter and the power amplifier. Power and control (ON/OFF and transmitter power increase/decrease) are done in the exciter, while the transmitter data come from the power amplifier parameters.

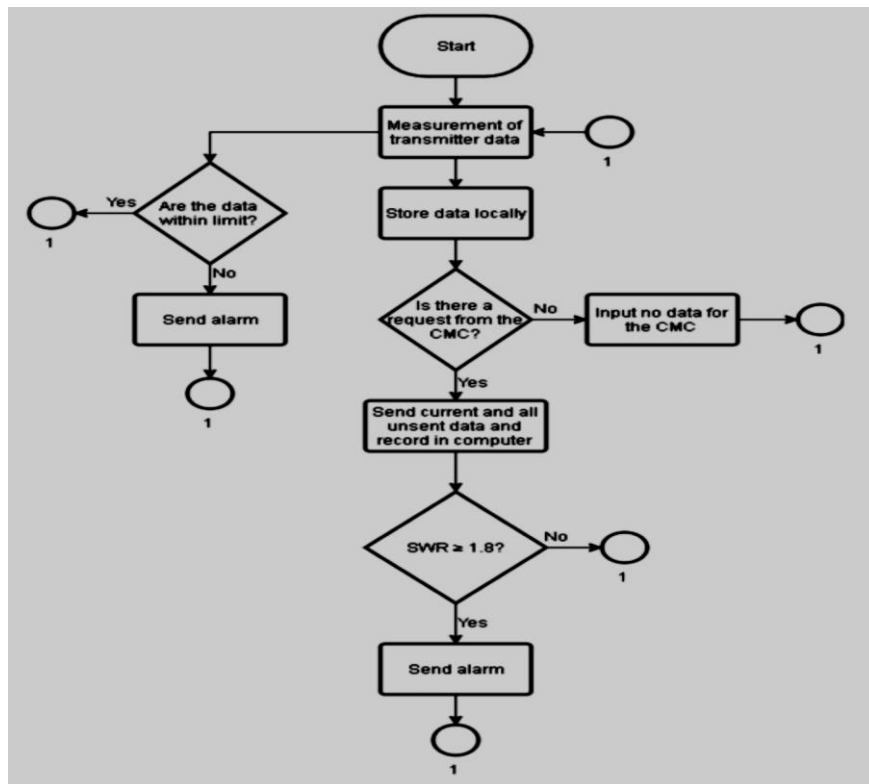


Figure 2. Flow diagram of the proposed system in the transmitter side

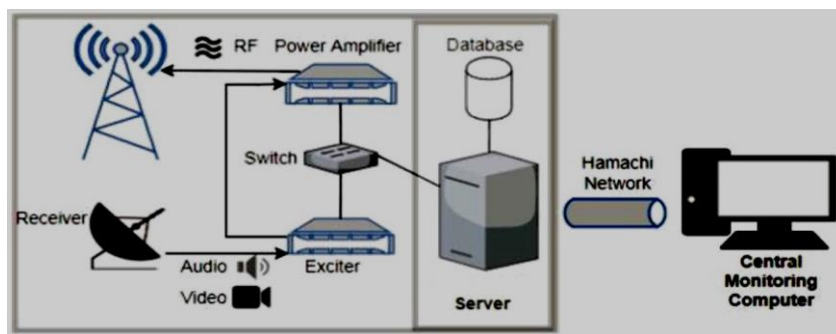


Figure 3. Network diagram of the proposed system

2.1. The Transmitter Server

The transmitter server is used to carry out the required functions described above. Specifically, it is a Java application that accepts data from the transmitter and is responsible for sending the data to the CMC. The application runs in the Netbeans IDE. The libraries imported in this code are (1) MySQL JDBC, a driver used for establishing the connection of Java and MySQL database; (2) ArduLink, a compilation of Arduino

coordination protocols used for communication and control of Arduino boards; and (3) RXTXcom, Java native library providing serial and parallel communication for the Java Development Toolkit (JDK); this library is required by the Ardulink library to establish serial communication with the Arduino board. The corresponding packages and descriptions are listed Table 1.

Table 1. Transmitter server packages

No.	Library	Used for
1	java.io.*	System input and output through data streams, serialization and the file system
2	java.net.*	Classes for implementing networking applications
3	java.sql.*	API for accessing and processing data stored in a database
4	java.text.SimpleDateFormat	Class for formatting and parsing dates in a locale-sensitive manner
5	java.util.List	To access elements by their integer index and search for elements in the list
6	java.time.Instant	An instantaneous point on the time-line
7	java.time.ZoneId	For changing the time-zone ID
8	java.time.ZonedDateTime	For reading the time-zone ID
9	java.time.format.DateTimeFormatter	For changing the Date and Time format
10	org.jsoup.Jsoup	Parse HTML into a Document
11	org.jsoup.nodes.Document	Create a valid, empty shell of a document, suitable for adding more elements
12	org.jsoup.nodes.Element	For getting HTML tag name, attributes, and child nodes
13	org.zu.ardulink.Link	To send commands to the Arduino
14	org.openqa.selenium.By	For finding HTML elements
15	org.openqa.selenium.WebDriver	For starting the webdriver
16	org.openqa.selenium.chrome.ChromeDriver	For opening Chrome web browser
17	org.openqa.selenium.support.ui.Select	For executing commands in the web browser

The implementation of the functions of the server uses multi-threading to run multiple classes at once. This allows the application to execute different processes independently as soon as the Thread.start() is executed in either in the Main code or in any class. The classes that are created in the application are: Real-time, Sending Real-time Logging, Sending Logs Retransmission Status, Power Control. The flow diagram of these applications are shown in Figure 4. The real-time class gets the source code of the PA interface and extracts the meter readings. In the Anywave transmitter used in the prototype, the following transmitter data are extracted: Voltage, Current1, Current2, Current3, Current4, Current5, Current6, Current7, Incident Power, Reflected Power, and SWR. The current values (Current1, Current2, etc.) correspond to the currents drawn by each power amplifier module in the transmitter. Transmitter data are saved on a set() Method and are to be called by the Sending Real-time and Logging classes. The logging process computes for the number of minutes and seconds before the time (in minutes) is set to 00 or 30. This allows the logging on the database to be with an interval of 30 minutes, the logging interval required by telecommunications regulatory bodies. After locally saving the data, the values are sent to the CMC. At the same time, the status class checks if the Standing Wave Ratio (SWR) exceeds the maximum set value, which is 1.8. If it exceeds 1.8, an SMS is sent. The power class is used for the remote turning ON and OFF of the transmitter, while the control class is used to remotely adjust the power (reduce or increase) of the transmitter. Both the power and control classes wait for control signal from the CMC before they execute. Selenium is used to control the interface of the exciter for transmitter power and control.

Lastly, the retransmission class is used for the transmission of transmitter data that never reached the CMC. This ensures that all stored data available in the transmitter side will be copied in the CMC. This class waits for the CMC to send the last row of the excel file, i.e., on the client side. The received CMC data file is then compared with the entries stored in the transmitter side. This is to determine the data to be retransmitted. The start trigger of these threads are written in the Main class of the application. Upon starting the application, the threads describe above should be running at the same time independently.

2.2. The Transmitter Database

The XAMPP of the server has 3 databases: Transmitter, status, and power. These facilitate the storage of the transmitter data, alarm, and transmitter power ON/OFF. Particularly, the transmitter database saves the transmitter ID, Voltage, Current1, Current2, Current3, Current4, Current5, Current6, Current7, Incident Power, Reflected Power, SWR, and the Timestamp of the data. The alerts database saves the transmitter ID, Alert, and the Timestamp of the alarm. The power database save the transmitter ID, Status, and the Timestamp of the time the transmitter was turned on/off. These data are extracted from the web interface of the transmitter through Selenium web driver. The driver is an open source library for Java that allows programs and scripts to control and manipulate web browsers. It works at any operating system and supports other programming languages.

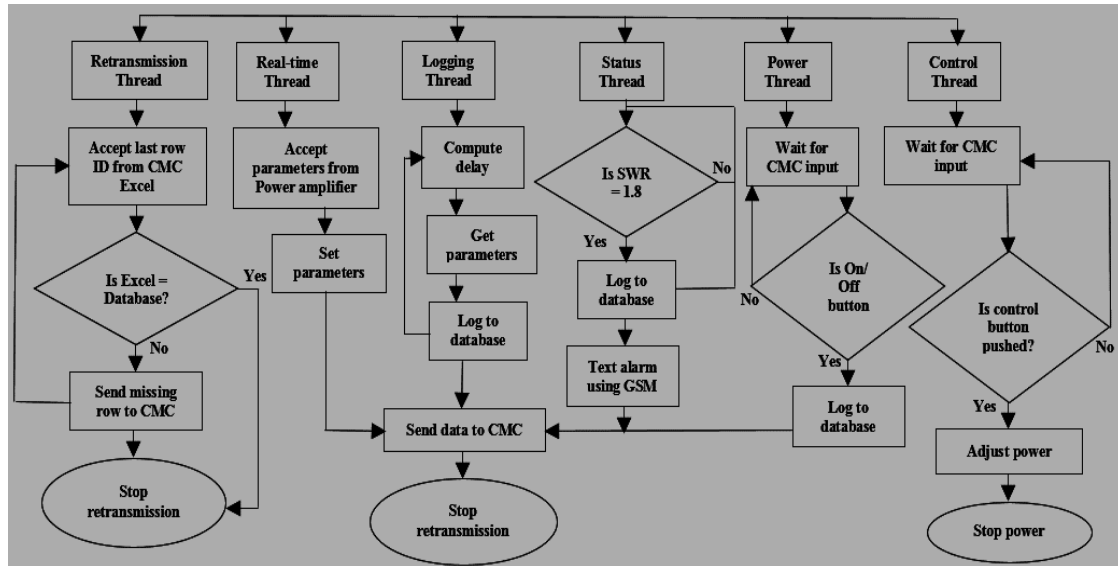


Figure 4. Flowchart of the transmitter server.

2.3. The GSM Module

GSM-based systems using Arduino microcontroller find vast applications in the sending of alert messages such as [10-17]. In the proposed system, a GSM module is implemented using an Arduino SIM800L in a Gizduino microcontroller board. SMS messages that the modules send are predetermined messages which were integrated in the Arduino code. The messages are (a) ‘Warning: SWR has exceeded limit’, (b) ‘Warning: Transmitter is turned off’, (c) ‘SWR is back to normal’, and (d). ‘Transmitter is turned on’. The Arduino code is triggered from the Java code of the server for these SMS alerts to be sent.

2.4. The Central Monitoring Computer (CMC) Client

It is in the CMC where all data from the transmitters will be gathered, monitor the present status, etc. Basically, it contains the GUI (Graphical User Interface) that shows the current transmitter data as well as a few buttons for remote transmitter control, and the excel file containing the logged transmitter data. The GUI is designed to monitor the status of several transmitters. It uses the Jxl Library that allows users to read, write, create and modify sheets in an Excel.xls) workbook during runtime. For the Anywave transmitter, 11 parameters are displayed in the interface at any given time. These are the voltage, current 1-7, incident power, reflected power and the SWR values. The current transmitter status (ON/OFF) is also displayed together with the 30-minute interval logs, with the latest log inserted on the top row of the table. The power control signal is typed on a text field and pressing the ‘Set Power’ button sends the data to the transmitter server. Figure 5 shows the GUI of the system for the monitoring of three transmitters; however, the system can monitor more than three. Each transmitter has its own power switching buttons located at the left side of the interface. These buttons enable the sending of command signals to the local computer to turn ON or turn OFF the transmitter. The first button at the right side of the interface is the ‘Transmit Specific Date button’ which allows the user to retransmit data at particular dates. This allows the sending of data at specific dates/time in case the CMC did not receive said data. The ‘Transmit All’ button request the remote server to retransmit the data based on the last entry in the CMC excel file. The GUI displays the latest meter readings every time new data are received. Pressing the ‘disconnect button’ closes the socket of the threads. This is necessary, especially when internet connection is interrupted; it is necessary to close the sockets of the threads first before pressing the ‘transmit all’ button again to prevent errors from being generated. The ‘View Entire Data Log’ button opens the excel file containing the logged data. The ‘print’ button allows the printing of the excel file.

3. RESULTS AND DISCUSSION

In After the construction of the prototype, it was tested in a television transmitter and an FM transmitter. Due to space constraints, only the transmitter data located in Tuguegarao City in northern Philippines, a relay station of UNTV, are presented. The results of the tests conducted on the prototype are presented below.

3.1. Logging at the CMC

Previous transmitter data are logged and saved in Excel files at the CMC. These files are automatically created in the CMC to save the logs coming from the server side. In the code that was developed to create the files, if there is no existing Excel file, a file will be created. If a file exists, the code will just append a new file to the existing file. The file contains all the logs made and saved in the database; a file is created per transmitter distinct from the other transmitters. The file format allows the creation of a file containing transmitter data for a particular date or a range dates if so desired. Figure 6 shows a sample of this file containing data collected early this year.

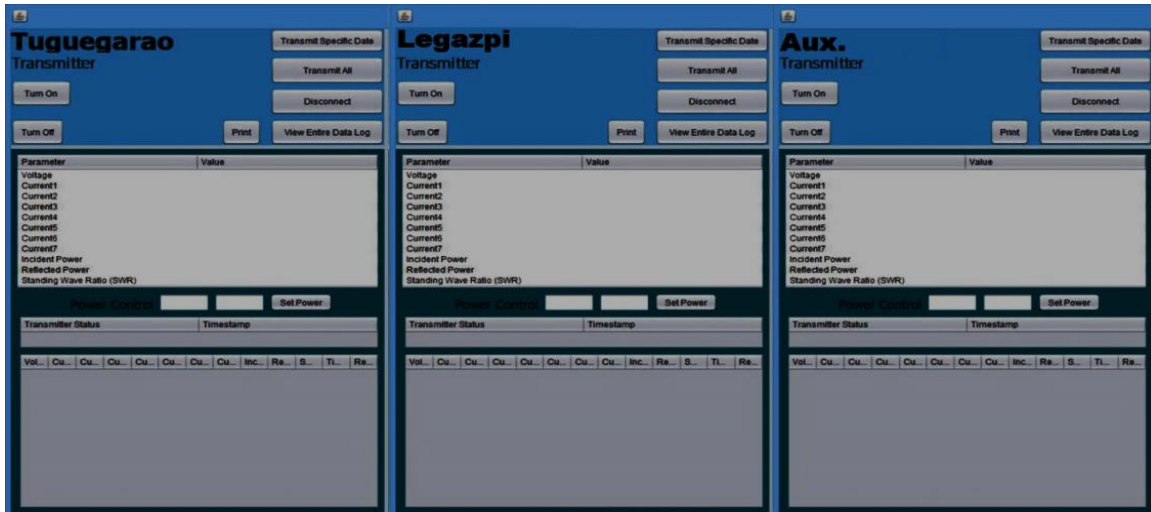


Figure 5. The GUI of the proposed system

ID	Voltage	Current1	Current2	Current3	Current4	Current5	Current6	Current7	Incident Power	Reflected Power	SWR	Timestamp	Remarks
1	49.5	0.01	1.59	7.71	7.65	7.27	7.82	0	1027.08	23.42	1.35	2018-01-26	
2	49.5	0.02	1.58	7.64	7.62	7.05	7.51	0	1027.08	23.55	1.35	2018-01-26	
3	49.5	0.02	1.58	7.58	7.49	7.25	7.48	0	1027.08	23.42	1.35	2018-01-26	
4	49.5	0.01	1.59	7.73	7.78	7.45	7.65	0	1027.08	23.42	1.35	2018-01-26	
5	49.5	0.01	1.59	7.84	7.67	7.31	7.77	0	1021.6	23.42	1.35	2018-01-26	
6	49.5	0.01	1.59	7.82	7.8	7.43	7.91	0	1021.6	23.3	1.35	2018-01-26	
7	49.5	0.01	1.6	7.92	7.92	7.46	7.91	0	1027.08	23.42	1.35	2018-01-26	
8	49.5	0.01	1.59	7.72	7.85	7.31	7.78	0	1027.08	23.42	1.35	2018-01-26	
9	49.5	0.01	1.59	7.84	7.67	7.32	7.79	0	1021.6	23.42	1.35	2018-01-26	
10	49.5	0.01	1.59	7.85	7.91	7.56	7.79	0	1021.6	23.42	1.35	2018-01-26	
11	49.5	0.02	1.58	7.87	7.8	7.43	7.77	0	1021.6	23.42	1.35	2018-01-26	
12	49.5	0.02	1.58	7.49	7.4	7.21	7.52	0	1027.08	23.55	1.35	2018-01-26	
13	49.5	0.01	1.6	7.86	7.82	7.54	7.76	0	1021.6	23.42	1.35	2018-01-26	
14	49.5	0.02	1.59	7.73	7.75	7.45	7.8	0	1021.6	23.42	1.35	2018-01-26	
15	49.5	0.01	1.58	7.47	7.45	7.1	7.5	0	1021.6	23.42	1.35	2018-01-26	
16	49.5	0.01	1.58	7.74	7.75	7.45	7.63	0	1021.6	23.3	1.35	2018-01-26	
17	49.5	0.01	1.59	7.87	7.68	7.34	7.72	0	1016.15	23.3	1.35	2018-01-26	
18	49.5	0.02	1.59	7.79	7.82	7.3	7.67	0	1021.6	23.3	1.35	2018-01-26	
19	49.5	0.01	1.59	7.73	7.68	7.32	7.82	0	1016.15	23.3	1.35	2018-01-26	
20	49.5	0.01	1.59	7.86	7.85	7.32	7.65	0	1016.15	23.3	1.35	2018-01-26	
21	49.5	0.02	1.58	7.56	7.62	7.03	7.6	0	1021.6	23.42	1.35	2018-01-26	
22	49.5	0.01	1.59	7.87	7.66	7.42	7.64	0	1021.6	23.3	1.35	2018-01-26	
23	49.5	0.01	1.59	7.82	7.82	7.33	7.82	0	1021.6	23.42	1.35	2018-01-26	
24	49.5	0.01	1.59	7.87	7.77	7.34	7.8	0	1021.6	23.42	1.35	2018-01-26	
25	49.5	0.01	1.6	7.86	7.87	7.48	7.91	0	1021.6	23.3	1.35	2018-01-26	
26	49.5	0.02	1.59	7.86	7.83	7.58	7.92	0	1027.08	23.42	1.35	2018-01-26	
27	49.5	0.02	1.59	7.83	7.68	7.33	7.83	0	1032.58	23.67	1.35	2018-01-26	

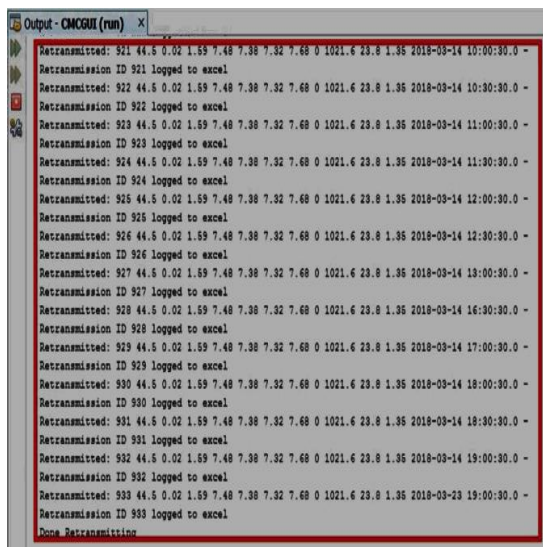
Figure 6. Screenshot of transmitter data in Excel file

3.2. GUI Display

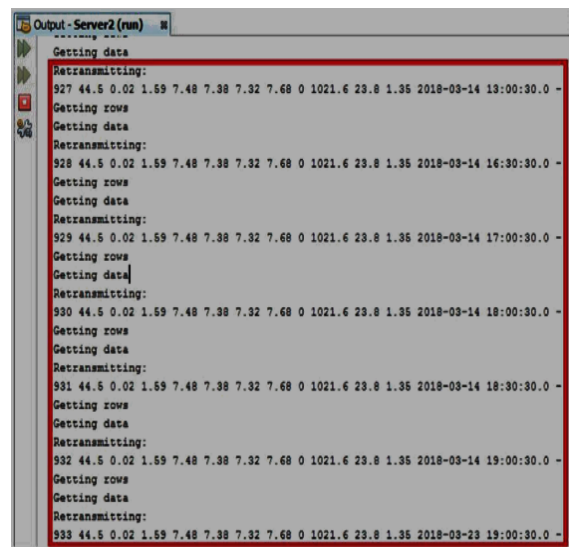
Tests were also conducted on the GUI of the system. A sample GUI filled with data is shown in Figure 8(a). Values displayed at the transmitter GUI were taken as samples to be compared to the latest outputs of the Netbeans IDE in the local and central computers. Both the data displayed in the GUI of Figure 7(a) and the central computer IDE output shown in Figure 7(b) are the same with the output of the local computer IDE (shown in Figure 7(c)).



(a)



(b)



(c)

Figure 7. Transmitter data (a) displayed in the GUI, (b) appearing at the central computer IDE output, and (c) originating at the output of the Netbeans ID of the local computer

3.3. Transmitter Powering and Power Control

The remote powering and power control of the transmitter has been successfully implemented and performs excellently. When the turn ON or turn OFF buttons in the GUI are pressed, the Selenium web driver launches the web browser at the local computer and logs in to the web interface [18]. The necessary setting from the drop down menu is chosen and submits it to the exciter database, then closes the browser. The same process was done in the process of varying the power of the transmitter by pressing the ‘set power’ button.

The request from the GUI with the power value is sent to the local computer. All current power settings are saved in the database of the exciter. These settings are maintained even in the advent of power failures or multiple turning ON/OFF the transmitter. Figure 8 (a) shows the screen shots of the web interface after the transmitter is turned- on (top) and turned-off (bottom). Figure 8(b) shows the web interface indicating the original power gain (top), after entering the new power gain (middle), and the new or the effected power gain (bottom). Please note that the transmitter power is varied by varying the gain.

3.4. GSM Module

The tests conducted on the GSM module use a mix of actual and simulated values. To test the sending of 'Warning: SWR has exceeded limit' and 'SWR is back to normal' messages are not possible in actual transmitter operation. These codes are simulated in the Java code of the servers. Simulation of the GSM in the Java code was done by using the java library ArduLink and the command link.writeSerial. The latter command found in the Java code, gives a serial input to the Arduino code for the GSM module [19-25]. When there is an abnormal transmitter condition found in the java code such as turning off the transmitter or SWR value being equal or lower than 1.8 the link.writeSerial command is activated. In all trials done, the system successfully sent the correct warning message. The text messages received during these trial are shown in Figure 9.

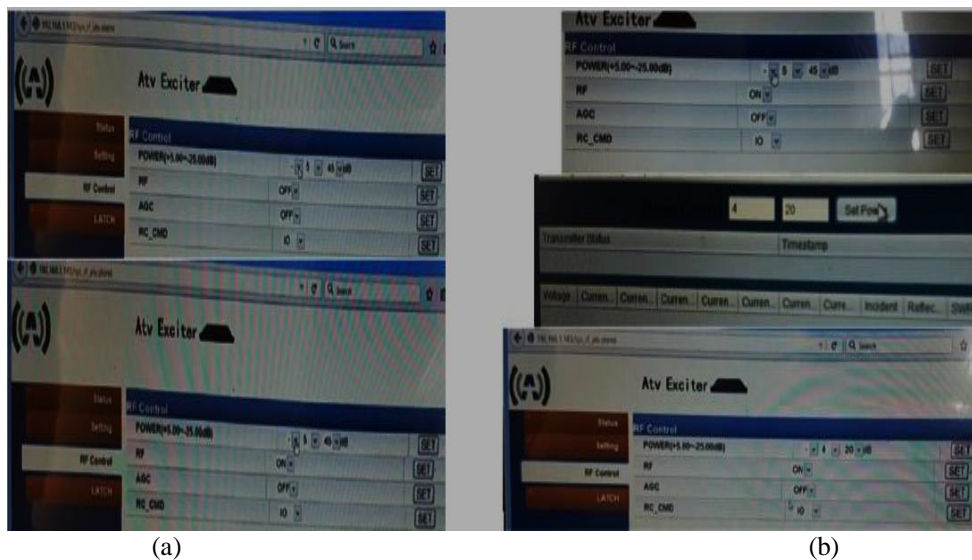


Figure 8. Illustrations of the web interface showing (a) the turning-on and -off of the transmitter, and (b) transmitter power variation

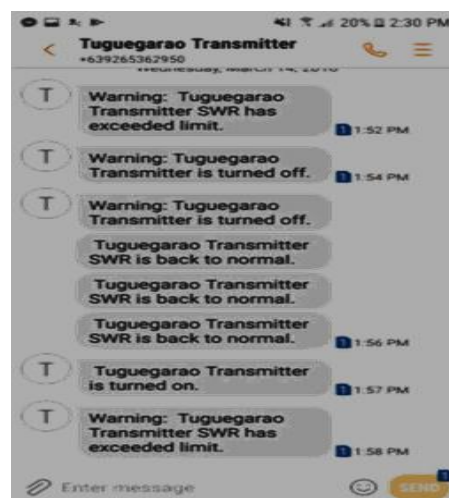


Figure 9. GSM Text Messages received during the tests

4. CONCLUSION

The test results show that the proposed system serves the purpose of remotely tending the operation of several modern transmitters at the same time. Further, the remote tending capability of the proposed system also satisfies the requirements of regulatory bodies in real-time monitoring, logging transmitter data regularly, and printing of logged transmitter data. On top of the required features, the system also has the capacity to give alarms remotely to the CMC in the GUI and through SMS in times of emergencies. The system also has the capacity to resend transmitter data for the CMC to recover back lost data during transmission or during internet disconnections. With the centralized tending of broadcast transmitters, broadcast networks can save much on human resources and operating costs. The use of the system in a broadcast network also reduces the health hazards that RF energy can induce to transmitter technicians. The proposed system is ideal for transmitters located in mountainous or rugged terrains and with relatively poor accessibility. Since each local transmitter server has an IP address, so theoretically, the number of transmitters that can be monitored using the proposed system is infinity; practically, it is limited by the number of GUIs that can be displayed on the computer monitor.

ACKNOWLEDGEMENTS

The authors would like to thank Engr. Arman Ursal, Quest Broadcasting, Inc., DLSU-URCO and UNTV for the realization of this paper.

REFERENCES

- [1] KBP, "Technical Standards and Operating Requirements for Broadcast Stations in the Philippines," 1991.
- [2] J.C. Whitaker, "Standard Handbook of Broadcast Engineering", McGraw-Hill Standard Handbooks, 2005.
- [3] Shu-ping Xu, et al., "The Design and Implement of Transmitter Remote Control System Based on the Embed Web Server," International Conference on Multimedia Technology, 2010, pp. 1 - 4.
- [4] E. Manir and S. M. Çınar, "On the Investigation of a Designed Smart Telemetry Control Algorithm," *National Conference on Electrical, Electronics and Biomedical Engineering (ELECO)*, 2016, pp. 650 – 654.
- [5] N. Kulkarni, S. V. N. L. Lalitha, S. A. Deokar, "Real time control and monitoring of grid power systems using cloud computing," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 941-949, 2019.
- [6] G. Quoc-Anh, N. Dinh-Chinh, T. Duc-Nghia, T. Duc-Tan, K. Nguyen Thi, K. Sandrasegaran, "Wireless Technology for Monitoring Site-specific Landslide in Vietnam," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 4448-4455, 2018.
- [7] I. Mukhlash, D. Yuanda, c M. Iqbal, "Mining Fuzzy Time Interval Periodic Patterns in Smart Home Data," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 3374-3385, 2018.
- [8] B. S. Badreddine, E. A. Chaker, S. Denys, "Development of a Java-based Application for Environmental Remote Sensing Data Processing," *International Journal of Electrical and Computer Engineering (IJECE)*, 2019
- [9] C. Fancy, L. M. M. Thanveer, "An evaluation of alternative protocols-based Virtual Private LAN Service (VPLS)," *International Conference on IoT and Application (ICIOT)*, 2017, pp. 1 – 6
- [10] W. H. W. Hassan, A. Z. Jidin, S. A. C. Aziz, N. Rahim, "Flood Disaster Indicator of Water Level Monitoring System," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 1694-1699, 2019.
- [11] P. W. Digarse and S. L. Patil, "Arduino UNO and GSM based wireless health monitoring system for patients," *International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2017, pp. 583 – 588.
- [12] N. S. Ali, Z. A. A. Alyasseri, A. Abdulmohson, "Real-Time Heart Pulse Monitoring Technique Using Wireless Sensor Network and Mobile Application," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 5118-5126, 2018.
- [13] H. Dahou, R. El Gouri, Mohammed Alareqi, K. Mateur, A. Mezouari, A. Zemmouri, L. Hlou, "Design and Implementation Intelligent Adaptive Front-lighting System of Automobile using Digital Technology on Arduino Board," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 521-529, 2018.
- [14] T. S. Gunawan, Y. M. S. Munir, M. Kartiwi, H. Mansor, "Design and Implementation of Portable Outdoor Air Quality Measurement System using Arduino," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 280-290, 2018.
- [15] M. I. Younis, T. F. Hussein, "Design and Implementation of a Contactless Smart House Network System," *International Journal of Electrical and Computer Engineering (IJECE)*, 2018, pp 4663-4672, 2018
- [16] D. Mlakić, S. Nikolovski, E. Alibašić, "Designing Automatic Meter Reading System Using Open Source Hardware and Software," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 3282-3291, 2017.
- [17] S. A. C. Aziz, A. F. Kadmin, N. Rahim, W. H. W. Hassan, I. F. A. Aziz, M. S. Hamid, R. A. Hamzah, "Development of automatic healthcare instruction system via movement gesture sensor for paralysis patient," *International Journal of Electrical and Computer Engineering (IJECE)*, 2019
- [18] J. Howse, "Training detectors and recognizers in Python and OpenCV," *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2014, pp. 1 – 2.
- [19] S. Fakhra A. G, A. Fauzan K, M. Saad H, R. Affendi H, K. H. Fen, "Development of a portable community video surveillance system," *International Journal of Electrical and Computer Engineering (IJECE)*, pp 1819-1826, 2019.

- [20] S. S. Arnob, A. S. Khan, R. Shelim, M. Chowdhury, "Safe sailing: GSM and GPS controlled autonomous boat with overweight detection and obstacle avoidance," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, pp 714-723, 2019.
- [21] N. A. J. Salih, I. J. Hasan, N. I. Abdulkhaleq, "Design and implementation of a smart monitoring system for water quality of fish farms." *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, pp 44-50, 2019.
- [22] K. J. Yong, M. H. Salih, "Design and implementation of embedded auto car parking system using FPGA for emergency conditions," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, pp 876-883, 2019.
- [23] M. Ahsan, R. Drolia, A. Kumar S, "Smart Walking Stick for the Visually Challenged," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, pp 1282-1288, 2018.
- [24] N. A. A. Rahman, A. B. Jambek, "Biomedical health monitoring system design and analysis," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, pp 1056-1064, 2019.
- [25] G. N. Solidakis, et al., "An Arduino-based subsystem for controlling UAVs through GSM," *6th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, 2017, pp. 1 – 4.

BIOGRAPHIES OF AUTHORS



Gerino P. Mappatao was born in Tuao, Cagayan, Philippines. He received the B.S. degree in electronics and communications engineering (ECE) from Saint Louis University, Baguio City, Philippines in 1989, the M.S. and PhD degrees in ECE from De La Salle University-Manila, Philippines in 1998 and 2012, respectively. He is currently an Associate Professor of Electronics and Communications Engineering at De La Salle University-Manila. He authored and co-authored papers in conference proceedings and journals on antennas, broadcast engineering, wireless communications and image processing.



Isaiah Mari Zebedia S. Bautista was born in Manila, Philippines. He received the B.S. degree in electronics and communications engineering (ECE) from De La Salle University, Manila in 2018. He is a former Assistant Vice President of the Electronics and Communications Engineering Society (ECES) and his research interests include Arduino microcontroller programming.



Joseph C. Castillo was born in Manila, Philippines. He received the B.S. degree in electronics and communications engineering (ECE) from De La Salle University, Manila in 2018. He is a former Vice President of the Electronics and Communications Engineering Society (ECES) and his research interests include Data Communications.



Marc Kevin J. Orsos was born in Manila, Philippines. He received the B.S. degree in electronics and communications engineering (ECE) from De La Salle University, Manila in 2018. He is a former Assistant Vice President of the Electronics and Communications Engineering Society (ECES) and his research interests include web development and computer vision.



Mark Aldrine C. Ribo was born in Manila, Philippines. He received the B.S. degree in electronics and communications engineering (ECE) from De La Salle University Manila in 2018. His research interests include Java Programming.