MQTT Protocol Discovery Service for an Iot-based Gensets Monitoring System

Bruno de Borba
AQTech Engenharia e Instrumentação S.A.
4150 Rod. SC 401 #23
Florianopolis, SC, Brazil, 88032-005

Roberto Alexandre Dias
Instituto Federal de Santa Catarina – IFSC
950 Av. Mauro Ramos
Florianopolis, SC, Brazil, 88020-300

Fabrizio Leal Freitas
AQTech Engenharia e Instrumentação S.A.
4150 Rod. SC 401 #23
Florianopolis, SC, Brazil, 88032-005

ABSTRACT

Condition monitoring systems designed for large power plants are too expensive to apply to small energy assets like generator sets (gensets). To enable the online monitoring of these smaller assets low cost hardware is needed, which implies limitations in processing and communication capacity. The concept of Internet of Things (IoT) and the use of communication protocols like Message Queuing Telemetry Transport (MQTT) are ideal to deal with those limitations, being a coherent alternative and economically viable to monitor low cost generation assets.

Discovery Services enables devices to broadcast their available features and commands, minimizing management effort. Until now, the MQTT protocol specification does not have a standardized discovery service.

This work proposes a MQTT protocol Discovery Service and its use for the provision of data collected in an IoT-based condition monitoring system for generator sets using Modbus RTU protocol over RS-485 serial bus. A prototype of the proposed system was built for testing and demonstration of the system.

General Terms

Algorithms, Electronic Instrumentation, Communication Protocols, Monitoring Systems.

Keywords

IoT, Internet of Things, Discovery Service, MQTT, Message Queuing Telemetry Transport, generator set.

1. INTRODUCTION

The concept of the Internet of Things (IoT) is to enable a variety of objects to connect to each other [1]. The Internet of Things is a concept of making the internet and communication between objects or “things” pervasive, where they can interact and cooperate with each other to achieve a goal. To make IoT feasible and allow hundreds of thousands of things to be connected, those things need to be inexpensive, which implies low processing, storage and communication capacity [2].

To allow the communication between several devices it is also necessary to establish one or more communication protocols between them. Because these devices are of limited capacity, these protocols must be adequate to handle low bandwidth, high latency networks and communication instability.

Low-cost devices and limited communication networks are features that fit the needs of the genset condition monitoring market since cost is a critical factor that can make monitoring not feasible and communication usually occurs on mobile or Wi-Fi networks.

The IoT concept is growing fast. The number of connected "things" is expected to reach 212 billion entities worldwide by the end of 2020 [3]. By 2022, it is expected that M2M (Machine to Machine) communication traffic shall constitute up to 45% of all Internet traffic [4]. The McKinsey Institute reported that the number of connected machines grew by 300% in the last 5 years [5]. The total annual impact caused by IoT is estimated between 2.7 trillion and 6.2 trillion dollars in 2025 [5]. Traffic monitoring in a mobile network in the United States showed a 250% growth of M2M data traffic in 2011 [6].

All these numbers and projections point to a potentially significant and fast-paced growth of Internet of Things in a near future. This growth provides a unique opportunity for traditional equipment manufacturers to turn their products into "smart things" [7].

Many of the traditional systems use Modbus protocol to allow process monitoring and control. In the generator sets market this protocol is practically consolidated (the facto Standard) in Programmable Logic Controllers (PLCs). There are several models of PLCs in the market, each one with its own particularities and different Modbus maps.

The objective of this work is to propose a MQTT protocol Discovery Service with no modifications in the protocol specification and to use this service to provide data in an IoT-based genset condition monitoring system using RTU Modbus protocol on RS-485 serial bus.

2. RELEVANCE

Electricity is an essential resource for society and country development and power generation is an activity that demands large investments. To ensure the return on those investments, the availability and useful file of equipment must be maximized, which makes asset management a critical success factor. One of the main tools of maintenance management is condition monitoring of generators, a subject that has been the target of numerous academic and corporate studies in recent decades.

The practical use of the project lies in the application of the proposed monitoring solution, through a prototype system, for the online monitoring of generator sets based on variables available in its Programmable Logic Controllers. The monitoring hardware should have a very low cost compared to more complex monitoring solutions commonly applied to larger power plants.

The results of this project are of great value for the power sector, for owners/users of generator sets and for society. As an economic result include increased availability and lifespan of generators. In social context it contributes to a better use of resources, to reduce the unavailability of the power grid and to distributed generation during peak consumption hours.

Events such as the 2001 blackout and the energy crisis initiated in 2012 (both in Brazil) have boosted the use of generator sets as an independent utility source in the electricity sector [8]. Estimates indicate approximately 200,000 generator sets in Brazil in 2017, making the market attractive in terms of volume when compared to the traditional power sector with a few large power plants.

3. PROBLEM DEFINITION

To guarantee the return of investments in electricity generation projects, it is essential to apply predictive maintenance philosophies instead of corrective and preventive policies. In this scenario, the online monitoring of generators plays a fundamental role to drive the transition of these philosophies, as it provides the necessary means to estimate the state of the asset and predict the failures before they occur [9].

In centralized power plants the nominal power is usually of the order of MW, in some cases up to GW. In the case of generator sets the power is usually in the order of kW. In this market, the cost of the asset is much lower in relation to a hydroelectric power plant, even when compared to the smaller ones. For this reason, the monitoring system cost must be very low so that it does not represent a very large part of the cost of the generation asset, making monitoring impossible.

Moreover, the installation sites of the generator sets are the most diverse. In some cases, Wi-Fi hotspots are available but, in most cases, it is necessary to use mobile networks to transmit monitoring data. In both cases the communication can be considered unstable, high latency and possibly low bandwidth.

Due to these characteristics, the concepts of IoT and the MQTT protocol are perfectly adapted to this type of application, since it must deal with low processing capacity (low hardware cost) and high latency networks, low communication bandwidth and high instability.

To develop this system viable economically (whose main characteristic is the low cost of the hardware) it is necessary to consider it in high volume. The concept of cloud computing is perfectly aligned with these needs, since it brings benefits such as flexibility, security and scalability.

In a system designed for high-scale application the concept of "zeroconf" is extremely desirable. This concept aims to reduce the effort required to manage the applications on the server side, since the application itself provides the data required for the operation of the system.

Whenever a data publication is performed via MQTT protocol, it is also necessary to publish the "location" where this information must be stored to be retransmitted to the devices that are registered to receive this data. In the MQTT protocol this "place" is called a "topic". Commonly, brokers available on the Internet implement access control lists (ACLs) to manage user permissions in a hierarchy of topics.

In the MQTT protocol all data is transmitted as text. When transmitting, for example, an integer, this number must be converted to text and encapsulated in the protocol. If an application wants to use this number as an integer after receiving the transmission, it must first convert the text to integer. This implies that the application must know the data type that is being received on each topic if you want to create rules about the received value.

If we consider a case where an application uses the MQTT protocol to receive data from various equipment models, for example gensets, it is common for at least some of the data to be different between the equipment models. Even when data refers to the same variable, for example battery voltage, different models can provide data with different digit numbers or different engineering units.

Simpler applications that do not give special treatment to the value or type of information being received can simply present the information to the user as text in a simple interface.

In order to be able to show the information considering the data type that is being received the user must manually inform the application of the information regarding the monitored signal. For example, if the application is to receive the fuel level in a variable ranging from 0% to 100% and the user wants to use a graphic to present this information in an analog pointer component, him/her must manually configure the application to display this variable. If we imagine that each application has a few tens of monitored variables and that the monitoring system has hundreds or thousands of monitored generator sets, this is an activity that requires a lot of manual effort.

The proposal is that the application not only publish the monitoring data but also publish information about the signal that is being monitored, like data type (text, integer, float point, etc.), magnitude type (level, voltage, current, pressure, etc.), engineering unit, largest and the smallest value expected for the variable (useful for graphic components in supervisors), etc.

With this information the application can create rules and present data in a more user-friendly way. When defining a variable such as fuel level, for example, the application may be able to automatically create a graphic marker component of fuel. By extrapolating this rule to distinct types of variables, the application may be able to dynamically and automatically mount a dashboard and present the information to users in a much more pleasant and intelligible way.

As related works, a request for an implementation of a MQTT server discovery service through UDP multicast packets was identified [10]. The proposal of this work is slightly different, since it does not aim to identify the accessible MQTT servers in a computer network, but to identify the available data in an MQTT Broker and its associated configurations.

Another reference that closely related to the proposed work supports only some types of sensors and does not standardize the topics that should be used for the publication of common signal configurations, like datatype, unit, etc. [11], which weakens the interoperability of the solution with different systems.

4. PROPOSED SOLUTION

4.1 Architecture

Generator sets are composed of a combustion engine coupled to an alternator [9]. The large majority of these generator sets are controlled by a Programmable Logic Controller (PCL) and supports Modbus RTU protocol.

Modbus is an application layer messaging protocol created in 1979 by Modicon, positioned at level 7 of the OSI model, and provides client/server communication between devices connected to different types of buses or networks [12].

Modbus protocol is a request/response protocol in which each service is specified by a function code. Because it is free of licensing fees and suits various physical media (such as RS-232, RS-485 and Ethernet), Modbus is used in thousands of devices, is a very cheap communication solution to use and is considered a standard (the facto standard) [13].

Monitoring data can be read from the genset PLC via Modbus RTU protocol in a RS-485 serial bus in most cases. Usually this is the interface that is available for the various PLC types. Some PLC models also use the RS-232 interface for Modbus RTU.

The IoT-based monitoring system shall transfer data over the internet using MQTT protocol. The Broker is a server that manages the exchange of messages between the publishers (who publishes the data) and the subscribers (who receives the data). This publication may take place via Wi-Fi network, GPRS network or some other TCP/IP network infrastructure supported by the monitoring system.

In our architecture, heavy processing and warehousing work to manage large numbers of publishers and subscribers is on the Broker's behalf, which can run in the cloud, leaving embedded systems with fewer responsibilities, enabling low-cost monitoring devices.

Subscribers can be mobile applications, WEB applications or any other program that is able to connect to the Broker via MQTT protocol and receive data, including a SCADA (Supervisory Control and Data Acquisition) platform.

The proposed system has a Discovery Service over the MQTT protocol with no modifications on the protocol specification. To this end, the "facilities" of the discovery service are managed as "topics" in the MQTT protocol, like the data itself. Through this system it is not necessary to modify the specification of the MQTT protocol, so that the implementation of the discovery service can be performed on applications already developed and made available on the Internet.

This system allows a company that provides generator set management services to manage various equipment from different customers, from distinct brands and models, without worrying so much about the implementation of the system in new equipment models.

The discovery service allows managed devices to automatically advertise to the management application, describing their functionality and commands, minimizing the service provider's management effort. Once the monitoring system publishes its facilities, the management system must be able to interpret these "features" and display the monitoring information correctly.

There are many MQTT brokers available on the internet, many with access control lists (ACL) that can hierarchically control access to device data, so that only authorized users can obtain monitored system information based on their credentials access.

Because gensets are relatively small and can be mobile, the environment to be adopted adheres to the IoT paradigm and the management application operates with cloud computing capabilities. Figure 1 shows the proposed monitoring system architecture.

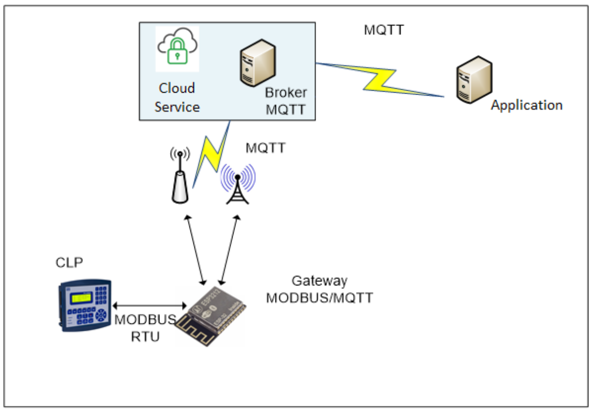


Fig 1: Monitoring system architecture

In this model, the Programmable Logic Controller controls the generator set (usually diesel) and communicates via the Modbus RTU protocol with a management module which in turn communicates through of the Message Queuing Telemetry Transport (MQTT) protocol with a middleware (Broker) operating in cloud computing. This broker can make the data available to a remote management WEB application accessible from personal computers and/or mobile or other application that communicates through MQTT.

The MQTT protocol operates in publisher/subscriber mode. When the generator set is initialized it publishes the list of facilities encapsulated in the topics of the MQTT protocol to the Broker. These facilities will be made available in the management application only for authenticated users with credentials that have access to the monitored system.

4.2 MQTT Discovery Service Definition

The proposed approach provides for the application to publish to the broker information regarding the monitored signal as soon as the connection is stabilized after authentication. This information is published in topics, so it is not necessary to change the specification of the MQTT protocol.

The Table 1 defines the values and description of each one of the topics. The topic "Value" is updated periodically, while other topics must be published at least once at system startup, because their values does not change during monitoring. Configuration topics must be published with the “retain flag” of the MQTT protocol. In this way applications that subscribes for these topics will receive the latest publication data even if the application starts after the monitoring system. This doesn’t happen without the “retain flag”.

Table 1. Description of the topics

Topic	Value	Description
Value	-	Value of the monitored signal, obtained through some transducer, communication protocol, etc.
Revision	0	Discovery Service Revision. Defines the remaining fields available for the specified discovery

		service and should be incremented in future releases.
Description	-	Description of the monitored signal.
DataType	0 – Boolean (0-1) 1 – Text (ASCII) 2 – Uint8 3 – Int8 4 – Uint16 5 – Int16 6 – Uint32 7 – Int32 8 – Uint64 9 – Int64 10 – 32 bits (single precision) float point 11 – 64 bits (double precision) float point	Data type of the monitored signal.
Magnitude	0 – Voltage 1 – Current 2 – Power 3 – Frequency 4 – Angle 5 – Time 6 – Pressure 7 – Temperature 8 – Level 9 – Key (open / closed)	Physical quantity of the signal.
Unit	-	Engineering unit of the monitored signal.
DecimalDigits	-	Number of decimal digits (unsigned integer).
Maximum	-	Highest expected value for the signal in the same data type of the signal.
Minimum	-	Lowest expected value for the signal in the same data type of the signal.

The proposed discovery service defines that only the first two fields are mandatory: value and revision. "Boolean" and "text" data types, for example, have no engineering unit, decimal digits, and no expected maximum and minimum values, as this information does not make sense for these data types.

However, the system should publish as much information as possible. In this way subscribe applications can make use of the received information about monitoring data.

5. PROTOTYPE USE CASE

The developed prototype system uses the "Wemos D1 Mini" Development Kit [14], which features the "ESP8266EX" Chip [15]. This chip is widely used by enthusiasts in IoT projects, as it is easily integrated to Arduino's development interface [16], allowing the use of several libraries available in the interface. Other features that contribute to the popularity of the chip is its low cost and integrated Wi-Fi system (standard Wireless 802.11 b/g/n), usually with built-in antenna on the boards that integrate the chip.

The development Kit has 16 pins, built-in Wi-Fi antenna, micro USB connector and reset button. Use this platform is to just connect it to the computer via a USB cable and install the "USB-TTL" converter driver built into the development kit itself. The Figure 2 shows the development kit.

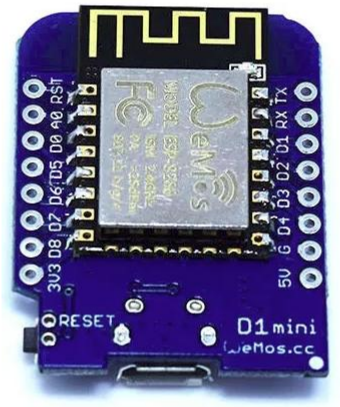


Fig 2: Wemos D1 Mini Development Kit

Some pins of the development kit are general-purpose, such as serial communication. The "5v" pin is directly connected to the USB power supply and serves to feed the kit voltage regulator. The pin "3V3" is connected to the output of the voltage regulator of the development kit and feeds the ESP8266EX chip, which works at 3,3v (Power and logic).

The "Rx" and "Tx" pins are connected to the "USB TTL" converter built-in in the kit, so that you can record and run programs in the development kit directly from the USB cable, without any auxiliary circuits. You can also use these pins when you do not have a USB cable connected to the serial communication kit with other equipment. The development kit also features 4MB Flash memory, which can be split to use as a program memory or as a filesystem memory.

Data acquisition from PLC using Modbus RTU protocol on RS-485 half-duplex serial bus is done by the prototype through a RS485 transceiver MAX3485 [17] connected to the pins "Rx" and "Tx". The direction control of the RS-485 half-duplex is carried out through a GPIO (general purpose input/output).

When the PLC provides Modbus protocol via RS-232 it is needed to replace the transceiver MAX3485 by MAX3232 [18]. In this case you do not need to use the GPIO direction control pins and no modification is required in the equipment firmware.

The prototype is powered by a 7-40VDC power source compatible with the voltage provided by automotive batteries used in the gensets. It also has interface for Debug via serial and interface RS-485 for connection with the PLC.

The configuration of the prototype is done via Wi-Fi network by a computer or smartphone. However, this feature is only available when the hardware is restarted in configuration mode (with the CONFIG button pressed).

The hardware monitor also the battery voltage by an AD converter integrated to the microcontroller, providing this information via MQTT protocol for remote monitoring.

The firmware record is done directly by the microcontroller's serial interface when it is restarted in firmware write mode (FLASH button pressed).

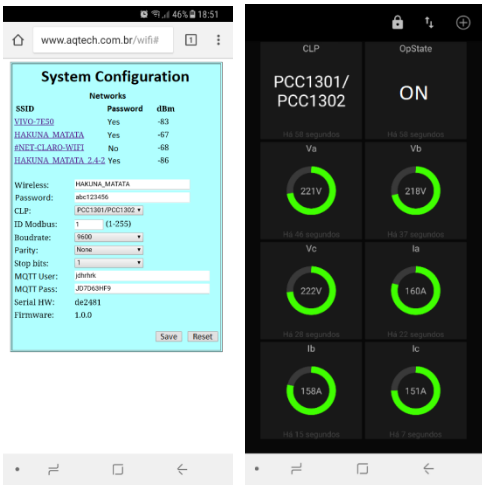
The system also has 3 LEDs to indicate the state of the equipment. It is easy to identify, for instance, a network failure (Wi-Fi or GPRS) or a serial communication failure (Modbus).

When the equipment is turned on or rebooted an initial configuration is performed, which involves the direction setting of the microcontroller ports among other settings.

If the Setup mode button is pressed, the software starts the setup mode via Wi-Fi. In this mode the equipment creates a Wi-Fi network where clients like a smartphone or a notebook can connect and be used to configure the hardware using a WEB browser.

After configuring and restarting the equipment in normal mode, the firmware enters the main loop that involves the data acquisition (the ADC of the microcontroller and Modbus RTU), the transmission of the data through MQTT and the wait for the next loop.

Figure 3 shows the prototype configuration page accessed using a smartphone browser (left) and a dashboard showing monitoring data using the mobile App "MQTT Dash" (right).



System Configuration

Networks

SSID	Password	dBm
VIVO-7E50	Yes	-83
HAKUNA_MATATA	Yes	-67
#NET-CLARO-WIFI	No	-68
HAKUNA_MATATA_2.4-2	Yes	-86

Wireless: HAKUNA_MATATA
Password: abc123456
CLP: PCC1301/PCC1302
ID Modbus: 1 (1-255)
Boudrate: 9600
Parity: None
Stop bits: 1
MQTT User: jdhrhrk
MQTT Pass: JD7D63HF9
Serial HW: de2481
Firmware: 1.0.0

CLP: PCC1301/ PCC1302 — Há 58 segundos
OpState: ON — Há 58 segundos
Va: 221V — Há 46 segundos
Vb: 218V — Há 37 segundos
Vc: 222V — Há 28 segundos
Ia: 160A — Há 22 segundos
Ib: 158A — Há 15 segundos
Ic: 151A — Há 7 segundos

Fig 3: Configuration page (left) and monitoring Dashboard (right)

In this scenario CloudMQTT Broker (www.cloudmqtt.com) was used as middleware acting in cloud computing. As a Modbus server (to simulate monitoring data) a proprietary software and hardware of the company AQTech was used for functional validation, acting like a genset PLC.

The application "MQTT Dash" (available at Play Store for Android smartphones) was also used. However, there are several alternatives for free applications that implement the MQTT protocol.

The “MQTT Dash” App doesn’t implement the proposed discovery service but allows the user to configure graphical components to show the monitored signals in different ways like Text, partially filled circles and others. So MQTT Dash was used to show that these configurations could be done automatically using the proposed discovery service system.

6. CONCLUSIONS

The proposed solution defines that features of the monitoring system are published in the MQTT broker at system startup. This information is published in MQTT topics (payload) so that it does not change the protocol specification. With this strategy it is possible to use existing applications on the market with the proposed discovery service approach without any changes.

Applications that receive the monitoring data can refer to the topics that have the discovery service facilities, for example, to improve the user experience on custom SCADA interfaces according to data types of monitored signals. If these applications want to consult the facilities of the discovery service, they just have to subscribe to the desired topics.

The objective of the work was achieved since the developed system implements the discovery service without any changes in the specification of the MQTT protocol. A prototype was developed for validation of the proposed approach and the results were satisfactory.

As future work it is possible to implement a SCADA software for different platforms to make use of the facilities available in systems that use the proposed discovery service. This software can dynamically build and present a dashboard based on monitored signals. The graphical representation of the measured quantities can be displayed based on the signal characteristics, available in the discovery service. For example, the software can show help buttons near the monitored signals that inform the description of each signal.

Based on the number of decimal digits field, the software can also show monitored signals with the desired number of decimal digits automatically. Based on the maximum and minimum fields, future software can configure graphical representations, like progress bars, according to each individual signal.

SCADA software could also store monitored values in a database with the correct data type, like Integers or Booleans, making use of database resources for specific data types. This can be done automatically using the field “data type” of the discovery service. Without this feature, SCADA software would have to store all monitored data in text format because it wouldn’t know the datatypes of each monitored signal. Storing signals in the correct data type allows the application, for example, to execute commands like calculate the maximum, minimum and mean values of a signal in specific time intervals using SQL queries.

The implementation of TLS tunneling and MQTT payload encryption is also valid as future work for applications where security is a critical requirement. The protocol has no built-in security technology or encryption besides simple user authentication with username and password. Due to this, authentication and encryption over SSL are commonly used in applications where data security is an important requirement. However, SSL can be very costly for very limited devices. One approach that could optimize security without major processing and memory consequences would be to encrypt the payload and authenticate through a Hash system.

7. REFERENCES

[1] Martins, I. R., Zem, J. L., “Estudo dos protocolos de comunicação MQTT e COAP para aplicações Machine-to-Machine e Internet das Coisas”, 2015.

[2] Torres, A. B. B., Rocha, A. R., Souza, J. N., “Análise de desempenho de Brokers MQTT em sistemas de baixo custo”. Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat). Universidade Federal do Ceara (UFC), 2016.

[3] J. Gantz and D. Reinsel, “The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in

the far east," IDC iView: IDC Anal. Future, vol. 2007, pp. 1–16, Dec. 2012.

[4] D. Evans, "The Internet of things: How the next evolution of the Internet is changing everything", CISCO, San Jose, CA, USA, White Paper, 2011.

[5] J. Manyika et al., "Disruptive Technologies: Advances that Will Transform Life, Business, and the Global Economy". San Francisco, CA, USA: McKinsey Global Institute, 2013.

[6] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, e J. Wang, "A first look at cellular machine-to-machine traffic: Large scale measurement and characterization", in Proc. ACM SIGMETRICS Perform. Eval. Rev., pp. 65–76, 2012.

[7] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. e Ayyash, M., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", IEEE Communication Surveys & Tutorials, Vol. 17, No. 4, 2015.

[8] M.S. Crestani. "Dificuldades e oportunidades da crise". Revista Eletricidade Moderna, São Paulo, ano 43, n.490, p.6. 2015.

[9] Ferreira, M. P., Freitas, F. L., Matsuo, T. K., Borba, B., Fonseca, J. E. R., "Monitoramento Online como Ferramenta para Otimização da Manutenção de Geradores: Uma Tecnologia 100% Nacional em Arquitetura Distribuída", XI Simpósio de Automação de Sistemas Elétricos, Campinas – SP, 2015.

[10] https://issues.oasis-open.org/browse/MQTT-267. Access 02/20/2018.

[11] https://home-assistant.io/docs/mqtt/discovery/. Access 02/20/2018.

[12] Modbus Application Protocol Specification V1.1b3. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. Access 10/26/2016.

[13] Modbus FAQ. http://www.modbus.org/faq.php. Access 10/26/2016.

[14] D1 mini. https://wiki.wemos.cc/products:d1:d1_mini. Access 11/23/2017.

[15] ESP8266EX Datasheet. http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. Access 11/23/2017.

[16] ESP8266 core for Arduino. https://github.com/esp8266/Arduino. Access 11/23/2017.

[17] 3.3V-Powered, 10Mbps and Slew-Rate-Limited True RS-485/RS-422 Transceivers. https://datasheets.maximintegrated.com/en/ds/MAX3483-MAX3491.pdf. Access 11/23/2017.

[18] MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver with ±15-kV ESD Protection. http://www.ti.com/lit/ds/symlink/max3232.pdf. Access 11/23/2017.