Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

Design and Implementation of Cloud-based Single-channel LoRa IIoT Gateway Using Raspberry Pi

Changqing Sun, Fuquan Zheng, Guangxu Zhou, Kun Guo
Institute of Automation, Qilu University of Technology (Shandong Academy of Sciences)
Shandong Provincial Key Laboratory of Automotive Electronics Technology

Abstract: With the emergence of industry 4.0 and Internet of things (IOT), more and more practical requirements need seamless connection from the smallest sensor to enterprise level systems and other systems with simple, convenient, reliable and economic approach. LoRa is a low-power wide area network (LPWAN) solution specifically designed for IoT devices that send and receive small amounts of data over a range of many kilometers with minimal power consumption. At the system architecture level, Lora is considered as a physical layer technology based on OSI model, mainly based on Chirp spread-spectrum(CSS) technology. LoRa is resulted in a robust and reliable technology compared to other standardized wireless solutions such as Zigbee , Bluetooth, WiFi, and 3G/4G cellular. In this paper, we have designed a single-channel LoRa gateway for Industrial IoT cloud applications using inexpensive single-board computer naming Raspberry Pi plus a LoRa add-on board with the necessary integrated circuits for wireless data communication. The system adopting the LoRa LPWAN technology consists of end devices, gateways, and cloud IoT services. The proposal gateway sends and receives LoRa messages to and from end devices and communicates with upstream cloud application servers through HTTP RESTful protocol and IoT-standard MQTT protocol, according to different data types and specific functional requirements. Unlike other ready-to-use LoRa gateways available on the market in which these gateways simply pass all the original LoRa packets back and forth between and network server over the air called “packet forwarding” mode, the proposal gateway contains smart-agent managing connectivity to devices, coordinating polling mechanism, decoding the packets, passing periodically the unified data to the public or private cloud service applications. Aside based on the embedded web server, the configuration of the gateway is carried out through the web page with desktop or mobile application and it can be updated anytime& anywhere when necessary. The proposal gateway is built cloud ready and can easily integrate with existing networks and enables low-cost, long range communication capability for a multitude of industry applications with minimum effort.
Key Words: LoRa, LoRaWAN, gateway, Industrial Internet of Things (IoT), Raspberry Pi

1 Introduction

More and more things in our world are getting connected to the internet，creating what's popularly called The Internet of Things（IoT）. For different communication technologies with low power consumption, many wireless IoT communication have been proposed and deployed. These standardized communication technologies include Bluetooth, ZigBee, Wi-Fi, or cellular(3G/4G). Comprehensive consideration of primary factors such as communication coverage, low power consumption, link budget and others, these technologies are not quite suitable for some IoT applications requiring long-range communication channels with low data rates and power consumption optimized. LoRa is a low-power wide area network solution specifically designed for IOT devices that send and receive small amounts of data over a range of many kilometers with minimal power consumption.

It operates on the unlicensed 433-, 868- or 915-MHz ISM (Industrial Scientific Medical) bands, depending on the region in which it is deployed. It uses the technology known as chirp spread spectrum (CSS) across a wide bandwidth to provide resilience to deliver interference or signal noise. The long-distance and low-power characteristics of LoRa make it an interesting candidate for industrial applications. LoRa networks are considered low-power wide area networks (LPWANs) [1]. LoRa is considered as a physical layer technology based on the OSI model. The network commonly consists of end devices, gateways (sometimes called concentrators) and application servers or cloud services. The key component is the gateway, which is responsible for collecting information and preprocessing from sensors and sending it to the data center [2]. Depending on the size and complexity of specific IoT project in the real world, the general architecture is to deploy IoT gateway devices or sensor hub to collect data from many sensor nodes and then forward that data on to an upstream data collection system such as private or public IoT cloud applications[3] [4].

To solve these problems，many works have been done in previous years on enabling direct integration of LoRa end device data into private or public IoT cloud applications[5]-[8] . Also, there are hundreds of ready-to-use LoRa gateways available on the market [9][10][11].But by whatever way, there are some shortages and deficiencies in structure and design of these gateways. Most of these gateways or convertors tend to be 'dumb' gateways. They don't do anything other than forward data on to the data center or an upstream collector. In comparison with the afore-mentioned methods and products, in this paper we propose a prototype gateway using cheap Raspberry Pi single-board computer and a simple LoRa add-on board for wireless data communication [14][12]. Powered by the MQTT messaging protocol and other cloud connectivity capabilities [17], the gateway polls and preprocess LoRa-based sensor data periodically. It is especially suitable for a variety of industrial real-time data acquisition and remote monitoring applications using cloud-based technology and services.

* This work was supported by Key research and development (R&D) plan of Shandong Province (No. 2019GGX101058), Major scientific and technological innovation projects of Shandong Province (No. 2019JZZY020810, No. 2019JZZY010448).

2 System Layout

Fig. 1 below provides a high-level overview of LoRa application architecture.

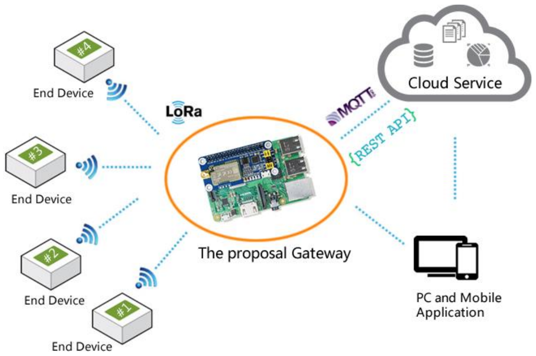


Fig. 1: The overview of LoRa application architecture

LoRa based application consists of end devices, gateway and network server or cloud-hosted web application. On the middle of the diagram is proposed gateway that comprise of Raspberry Pi mini-computer and LoRa add-on board. On the left half of the diagram depicts one or more LoRa end-devices. The right half of the diagram depicts the interaction with a cloud-hosted web application using the RESTful APIs cloud platform and MQTT protocol [16] [17][18]. Data and useful information are accessible to authorized users through desktop, web application and mobile APP. The network topology between end devices and gateway is star structure. End devices and gateways are connected wirelessly using LoRa communication. Gateways and network servers are connected using IP backhaul connections typically Ethernet or 4G with communication such as RESTful HTTP and MQTT protocol. In the basic architecture of above, the proposal gateway is key component that acting as only bidirectional relay, or protocol converter, with the cloud-hosted web application being responsible for decoding the packets sent by the end devices and generating the packets that should be sent back to end devices.

3 Implementation Details

Key components and flows are described at a high level below.

3.1 LoRa End Devices

Fig. 2 below is a schematic block diagram for LoRa end device. A typical LoRa end device will have some or all of the components and functions shown in the diagram.

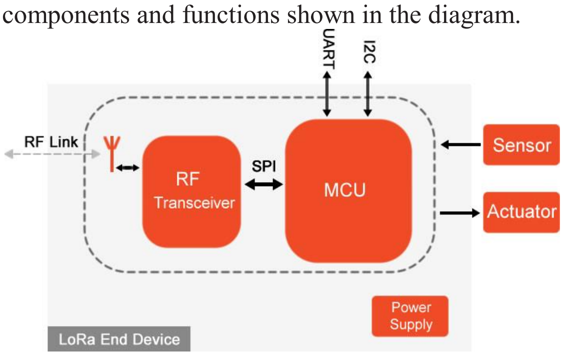


Fig. 2: Basic LoRa end device block diagram

MCU(microcontroller) interfaces with RF transceiver module through SPI protocol and runs many software functions such as sampling the data from external sensor, sending control signal to external actuator, formatting the sensor data into LoRa protocol's payload format and scheduling of LoRa messages to gateway. For accomplishing the above-mentioned functions, there are a lot suitable and cheaper microprocessors to choose from, for example ST Micro series Cortex-M0 MCU STM32F030F4 . RF transceiver converts the data from MCU with SPI interface and protocol to an analog radio signal by modulating it onto an RF carrier frequency. It also receives incoming RF radio messages, performs demodulation, converts the analog signals back to digital and forwards the incoming radio messages to the MCU. The module composed of Semtech SX1276 ultra long range spread spectrum wireless transceiver is intended for applications over a wide frequency range for example 433MHz ISM band in China. It is very suitable for our project [12]. End Devices is available peripheral I/O interfaces include UART and I2C through that communications with external equipment for completing tasks such as viewing or setting parameters, etc. It can be a standalone transmitter/receiver, or as part of a more complex IoT system.

3.2 Gateway Hardware Implementation

Fig. 3 below is a schematic block diagram for LoRa gateway.

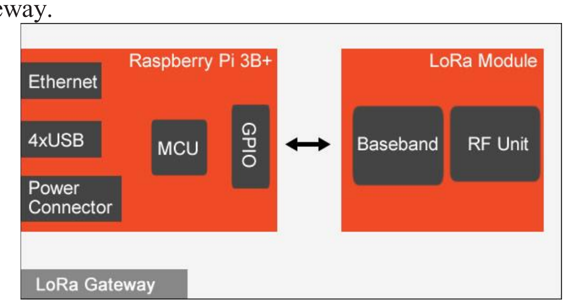


Fig. 3: Block diagram of the proposal gateway

On the left half of the diagram is a Raspberry Pi board and the right half of the diagram is a LoRa module. Raspberry Pi uses its GPIO to interface the LoRa module.

Fig. 4 below shows hardware Implementation of proposed system.

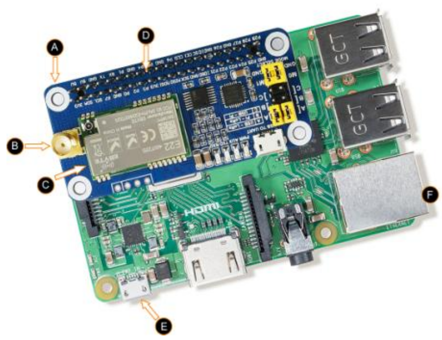


Fig. 4: Implementation hardware of proposed gateway. (A) LoRa add-on board(top); (B) external antenna connector; (C) LoRa module; (D) 40 pin GPIO connector; (E) Raspberry Pi 3B+ board(bottom); (F) power input; (G) RJ-45 Ethernet connector

It mainly comprises Raspberry Pi (RPi) board (top of Fig. 4) and LoRa add-on board (bottom of Fig. 4). Raspberry Pi board is mode 3B+ that supports gigabit RJ-45 Ethernet, dual-band 802.11ac wireless LAN, Bluetooth 4.2, external USB devices and much more. Based on a 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU and 1GB LPDDR2 SRAM, this board provides an effective platform for exploring more complex local-processing capabilities[13][14][15].

LoRa add-on board is a RPi expansion board based on Semtech SX1268 [12]. It covers 433MHz frequency band and allows data transmission up to 5km through UART serial port. It also has many excellent characteristics such as higher rate, longer communication distance, lower consumption, better safety and anti-interference. It is very suitable for various industrial control applications.

The Raspberry Pi 3B+ has two UART controllers named mini UART and pl011UART that can be used for UART serial interface. In this project, the mini UART used for communication with LoRa add-on board is mapped to TXD (GPIO 14) and RXD (GPIO 15) on 40PIN GPIO extension header.

The RPi 3 is external powered using a micro-USB cable which is connected to a 5V power supply that is rated to deliver a current of at least 2,500 mA for system stability and reliability.

The antenna is a key component for reaching the maximum distance in the wireless communication between end devices and the gateway. In this project, the external antenna is used to extend the communication range of the system.

4 Protocol Design and Software Implementation

The normal topology for this system is a star: all the end devices transmit towards a central gateway. The communication between gateway and end devices uses a protocol designed for the specific use. The proposal gateway implements a simple link protocol.

4.1 Communications Setup

Raspberry Pi uses the mini UART to communication with LoRa add-on board. The UART communication parameters are set as follows:

- Baudrate: 11520
- 8 Data bits
- No parity
- 1 Stop bit

4.2 Setup Configuration Parameters of LoRA Module

In order to configure the LoRA module, several parameters are available for the customization of the LoRa modulation include frequency, BW (Bandwidth), CR (coding rate) and SF (Spreading factor) and much more. Among these parameters, frequency is set to fixed value 433MHz. The 433MHz ISM bands is application free transmitting and receiving frequency in China, which can be directly deployed and not requiring a license. BW only can be chosen among 3 options: 125 kHz, 250 kHz or 500 kHz.CR is a number 4/N and N is between 5 and 8. It denotes that every 4 useful bits are going to be encoded by 5, 6, 7 or 8 transmission bits depending on its value. SF can choose from a number between 6 and 12 and this parameter is relevant in the spread spectrum technique [1][12].

In the reality application deploy, it should be determined according to the specific situation. In fact, there is a compromise between the distance range and the transmission speed. For example, the spreading factor has significant impact on the network coverage. It requires the gateway and end-devices to agree on using a single spreading factor for communication. Moreover, the longer range of the higher spreading factors (SF10, SF11, SF12) compared to the lower spreading factors (SF7, SF8, SF9) increases the possibility of collisions [12].

4.3 Sending Data to End Devices

We design the DATA command to send data from proposal gateway to end devices or any other LoRa equipped device. The packet structure of this command is shown in Table 1.

Table 1: The Structure of DATA Command

Field of packet	Value	Description
STX ASCII character	0×02	Start of Header
command	DATA	Meaning DATA command
Serial ID	Number between 1 and 255. 0 for broadcast message.	Each end device uniquely
Number of bytes [1 byte]	[1-200]	The number of bytes between this field and CR+LF
Data to send		
CR+LF ASCII characters	0x0D and 0x0A	Carry return + line feed
CRC16 code	[2 bytes]	2 hexadecimal bytes **CRC** between **STX** and CR+LF
ETX ASCII character	0×03	End of Transmission

As defined in Table 1, the frame of a command example will be: [STX] DATA 01020102[CR+LF]2A31[ETX]

The same command in hexadecimal format:

02 44 41 54 41 01 02 01 02 0D 0A 7A 05 03

The gateway supports two types of transmission named Unicast and Broadcast (Serial ID is 0 shown in Table 1). Unicast is used to send a packet to one specific end device in a network through its device address (Serial ID). In the Unicast process, we adopt a timeout mechanism to prevent a permanent wait for a packet to return. In addition, if the gateway does not receive the ACK confirmation message within the specified time interval, it will resend the packet up to three times or until the ACK is received.

Compared with Unicast, the disposal mechanism of Broadcast is simpler. Broadcast is used to send a packet to all end devices in a network and any end device within range will accept the packet. Broadcast just sends a packet and no ACK confirmation or retry is required.

4.4 Data Frame Structure from End Device to Gateway

The framework shown in Table 2 is designed in order to create sensor data frames of end device with a specific format with short payload size using low bit-rate protocols. The frame is composed of two different parts: header and payload.

Table 2: The Structure of Data Frame

Header			payload.			
STX STX STX	Number of bytes	Serial ID	Sensor_1	Sensor_2	···	Sensor_n

The structure fields are described below:

Start Delimiter [3 bytes]: It is composed of three STX ASCII character (Hexadecimal value is 0×02). This is a 3-byte field and it is necessary to identify each frame starting.

Number of bytes [1 byte]: This field specifies the number of bytes between this field and the end of the payload.

Serial ID [1 byte]: This is a one byte field which identifies each end device uniquely.

PAYLOAD: This part is composed of several sensor data. All data sent in these fields correspond to a predefined sensor data type.

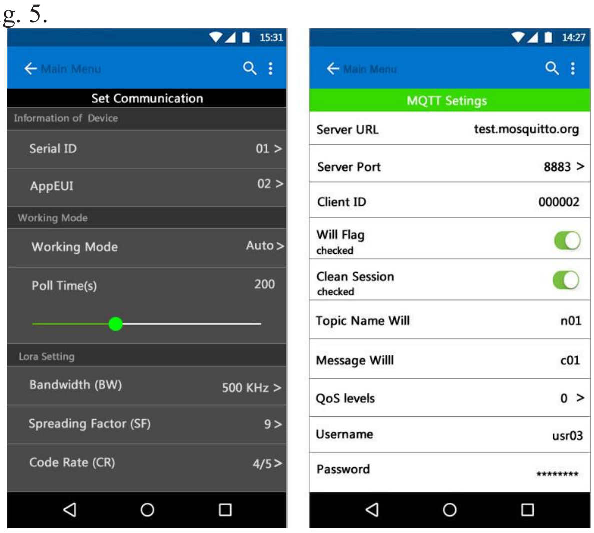
The sensor field consists of four bytes correspond to the single sensor value. For example the remote tank level sensor is a float number, the sensor field for 5.9m will be set as follows shown in Table 3.

Table 3: Simple Sensor Field

Sensor field 1			
Byte 1	**Byte 2**	**Byte3**	**Byte4**
0x9A	**0x99**	**0xA9**	**0x40**

4.5 Built-in Web Server Design

We use the Flask framework based Python template engine to set up a web server for parameter configuration[].Related tasks can be easily accomplished through PC , Smartphone APP or browser such as Microsoft Internet Explorer using HTTP protocol. Example of configuration parameter setting user interface is shown in Fig. 5.



(A) (B)

Fig. 5: Screenshots of Smartphone APP (A) Basic configuration for LoRa Module; (B) Main MQTT parameters and attributes settings

The section on the left half of the diagram is used to define the main parameters of LoRa Module such as various configurations include BW (Bandwidth), CR (coding rate) and SF (Spreading factor) .

The section on the right half of the diagram is used to define some major settings of MQTT protocol such as remote server information and specific communication parameters like QoS [17][18].

The screenshot in Fig. 6 below shows some settings between gateway and end devices.

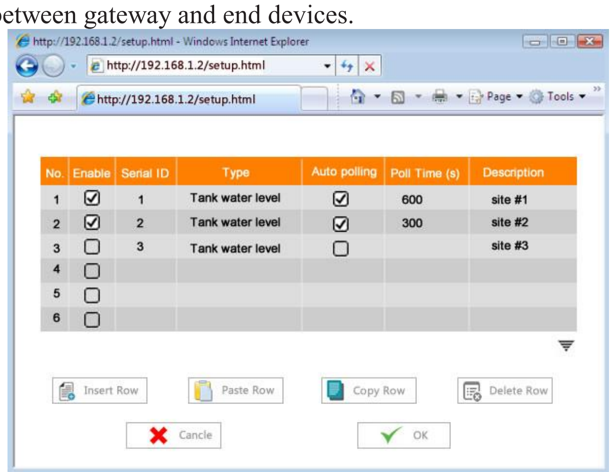


Fig.6: Shows some settings between gateway and end devices via IE browser on PC

The gateway software usually polls the sensor data periodically and allows user to easily configure the polling interval for each end device.

5 Conclusions

In this paper, we propose a prototype of LoRa-based IoT gateway using inexpensive credit-sized mini-computer Raspberry Pi and a plug-in printed circuit board with the necessary integrated circuits for data processing. The relevant hardware composition and software design are clearly described in detail. The system consists of devices or nodes, gateways, and cloud services. The communications protocol used between the devices and the gateways is LoRa with incredible performance compared to other competing technologies, and between the gateways and the cloud services is TCP/IP using wired or wireless network. The proposal gateway is suitable for building long-range communication channels with low data rates in wide array of applications such as machine condition monitoring and manufacturing automation application. It also can be used to replace some parts of new or existing solutions that require wireless connectivity and IoT cloud applications. With sample implementation based on the proposal gateway, the results mainly come from the good quality of the system design, the distance between end devices and gateway we achieved is up to 5 km because the lora module works well.

Nevertheless, with some minor problems were encountered in this project, and the proposal gateway was designed just only from the beginning as a LoRa-based IoT applications. Furthermore, the gateway must provide extensive and comprehensive protocol support. After all, the single-channel LoRa gateway is some limited compared to a real industry standard LoRaWAN-compatible multichannel gateway. Our future work will focus on expanding single channel to LoRaWAN-compatible multichannel by choosing more expensive RF chipsets such as SX1301/SX1278 instead of SX1272/SX1276 in hardware design and adopting full-featured LoRaWAN protocol in software[12][19], and then, this kind of improved gateway can handle data coming from many end devices simultaneously. In return, this will give end users greater flexibility in deploying applications that best suit their specific business.

References

[1] LoRa Alliance. White Paper: A Technical Overview of Lora and Lorawan; The LoRa Alliance: San Ramon, CA, USA, 2015.

[2] CHEN Xiaosong, Design and Implementation of Lo R a Low-cost Full-duplex Gateway, Radio Communications Technology,2019,45(02):206-209.

[3] Wei Yuantong, Dai yawen, Design of Industrial Monitoring ad hoc Network Protocol System Based on LoRa, Computer Measurement & Contro, 2019,27(02):225-228.

[4] QIAO Haiye, Design of equipment protection system based on lora communication technology, Automation & Instrumentation, 2019(08):73-75+79.

[5] ZHAO Er-gang，GUO Tian-yong，Design of Encrypted Wireless Communication System Based on LoRa Technology, Automation & Instrumentation,2019,34(11):1-4+8.

[6] Li Da,Yang Zhen, Design of LoRaWAN Integrated Gateway System Based on Embedded Linux, Microcontrollers & Embedded Systems，2019,19(07):10-14.

[7] ZHANG Min，ZHOU Zhiping，Design of remote monitoring system based on Lo R a, Transducer and Microsystem Technologies,2019,38(07):95-98.

[8] WUJin, ZHAOXin liang, ZHAOJun, Modulation and demodulation of LoRa internet of things technology, Computer Engineering and Design, 2019,40(03):617-622.

[9] LoRa gateway. www.dragino.com

[10] LoRa gateway. www.libelium.com

[11] LoRa gateway. https://www.cisco.com/c/en/us/products/routers/wirelessgateway-lorawan/index.html

[12] SX1276/77/78/79 WIRELESS, SENSING & TIMING DATASHEET Rev. 4 Available: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf [Accessed July. 9, 2019]

[13] http://www.raspberrypi.org, About raspberry Pi.

[14] Matt Richardson, Shawn Wallace, Getting Started with Raspberry Pi, 2nd Edition, Published by Maker Media,Inc., USA, 2014. Book ISBN: 978-1-457-18612-7

[15] Gay, W. W.: Raspberry Pi System Software Reference, Springer Science+Business Media, New York, 2014

[16] L. Richardson, M. Amundsen, RESTful Web APIs. O'Reilly Media, 2013.

[17] MQTT, Documentation: Protocol Specifications, MQTT. [Online]. Available: http://mqtt.org/documentation. [Accessed July. 17, 2019]

[18] MQTT protocol specification. [Online] http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf. [Accessed July. 12, 2019]

[19] Semtech SX1301,Digital Baseband Chip for outdoor LoRaWAN? macro gateways,https://www.semtech.com/products/wireless-rf/lora-gateways/SX1301