

Построение корпоративной сети LoRaWAN

с использованием открытого программного обеспечения

В статье описывается настройка корпоративной сети LoRaWAN и приводятся примеры использования открытого программного обеспечения, необходимого для функционирования клиентских устройств, шлюзов, сервера и служб, использующих сеть для решения прикладных задач. Рекомендуются бюджетные аппаратные средства, не требующие значительных капитальных вложений на этапе пуска и тестовой эксплуатации. Данная информация может быть полезной для инженеров — разработчиков систем связи, внедряющих новые перспективные технологии в области телеметрии, телеуправления и сигнализации.

Сергей Гаевский
info@nuvoton.com.ua

Беспроводные технологии связи сегодня являются основой для построения современных систем автоматизации технологических процессов и бытовой инфраструктуры. Необходимость постоянного обмена данными между управляющими и исполнительными звеньями определяет высокие требования к радиоканалу, а загруженность эфира накладывает определенные ограничения на параметры радиоустройств. Реализация надежных систем связи повышенного радиуса действия и низкой мощности, соответствующих требованиям региональных законодательств в области регулирования использования частотного спектра для работы без лицензии, является приоритетом для разработчиков.

Технология построения радиосетей LoRaWAN в безлицензионных ISM-диапазонах с использованием устройств мощностью до 14 дБ обеспечивает простое решение прикладных задач по сбору информации и телеуправлению, являясь достойной альтернативой использованию GSM-устройств и значительно превосходя последние в плане экономии энергопотребления. В [1, 2] рассказано о принципах модуляции LoRa и о возможности построения радиоканала «точка–точка» с использованием радиомодулей на базе чипсетов SX12xx, представленных на рынке компанией Semtech. Подобное решение является наиболее бюджетным по используемым компонентам, но, как правило, требует значительных усилий программиста по реализации оптимального протокола радиообмена, соответствующего заданным техническим условиям. Радиочипсеты являются лишь симплексными радиопередатчиками, поэтому бремя управления сетью полностью ложится на прикладную задачу, значительно повышая требования

к программному обеспечению. Пользователь должен позаботиться о планировании сеансов передачи клиентских устройств, предотвращать возникающие коллизии, обрабатывать подтверждения доставки данных и обеспечивать повторные передачи при потерях пакетов. Кроме того, необходимо грамотно реализовать шифрование и аутентификацию, блокируя возможные действия злоумышленников по вмешательству в работу инфраструктуры, обслуживаемой радиосетью.

Производитель LoRa-радиомодулей компания Semtech совместно с корпорацией IBM разработала сетевой стек LoRaWAN, предоставляющий «из коробки» функции, необходимые для эффективного управления пулом клиентских радиоустройств, и обеспечивающий надежную защиту радиокommunikаций.

Архитектура сети LoRaWAN представляет «звезду звезд» с центральным сервером, несколькими шлюзами и множеством клиентских устройств, обменивающихся данными с сервером через ближайшие доступные шлюзы (рис. 1).

Идентификация клиентских устройств в сети осуществляется по уникальному восьмибайтному идентификатору устройства (*devuid*) и назначаемому пользователем четырехбайтному имени (*devaddr*), которое отображается на сервере. Также указывается назначение устройства: восьмибайтный идентификатор службы, с которой данное устройство будет работать (*appid*), и 16-байтный ключ аутентификации в данной службе (*appkey*). Регистрация в сети (JOIN) может выполняться двумя способами: с помощью процедуры OTAA (over-the-air activation) или процедуры ABP (activation by Personalization). OTAA требует наличия соот-

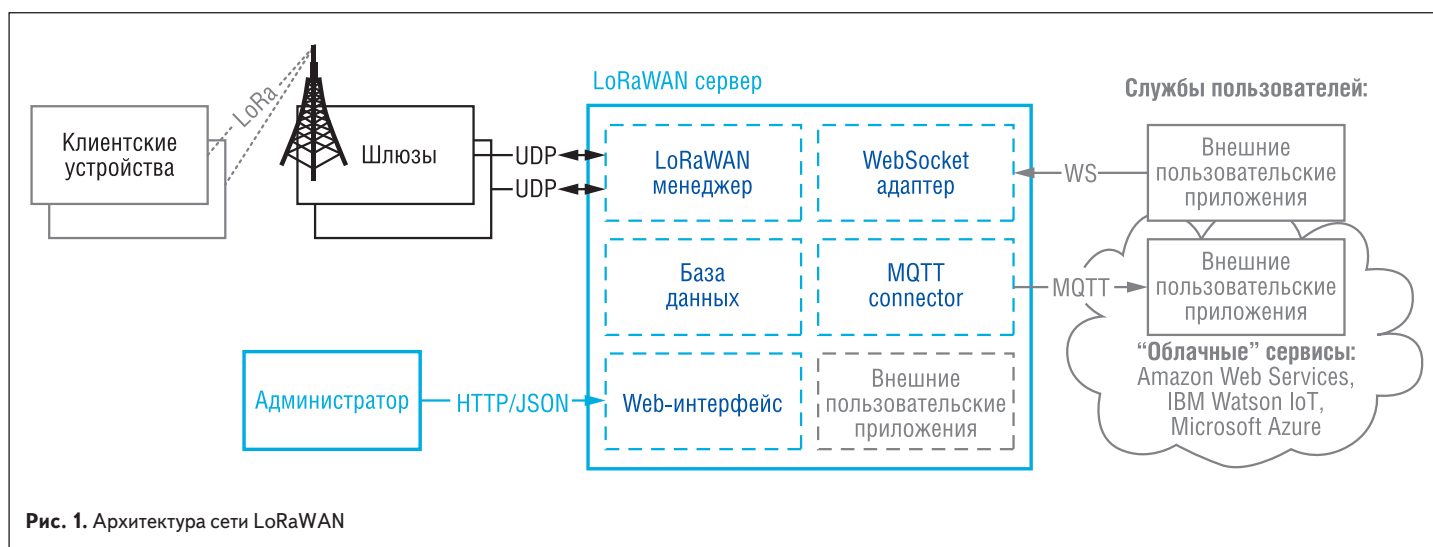


Рис. 1. Архитектура сети LoRaWAN

ветствующих параметров `devaddr`, `appkey` и `appkey` в самом устройстве и на сервере. Сессионные ключи шифрования данных (для сетевого уровня — `nwkskey`, для уровня приложения — `appskey`) будут сгенерированы автоматически при первичной регистрации в сети. Процедура ABP требует указать `devaddr` и непосредственно `nwkskey` и `appskey` в клиентском устройстве и в настройках сервера, тем самым исключается процедура получения сеансовых ключей через эфир: клиентское устройство будет готово к работе в своей сети сразу после первичного включения.

После регистрации в сети клиентское устройство может функционировать в одном из трех классов:

- Устройства класса А большую часть времени находятся в режиме сна, периодически просыпаются, отправляют данные, незначительное время ожидают подтверждения от сервера, затем снова «засыпают» до следующего сеанса в соответствии со своим расписанием. Такой режим работы, обеспечивая максимальное энергосбережение, имеет и недостаток: в период между коротким обменом данными устройства недоступны для сервера, и последнему приходится ждать очередного сеанса связи.
- Устройства класса С подавляющее большинство времени находятся в режиме приема, за исключением коротких сеансов передачи данных. Такие устройства потребляют значительно больше энергии, но они в любой момент доступны для сервера.
- Устройства класса В предоставляют разумную альтернативу: большую часть времени они находятся в режиме сна, но, тем не менее, между отправками данных регулярно «просыпаются» и некоторое время слушают эфир. Эти временные интервалы строго синхронизированы с сервером. Таким образом, передача данных на устройство класса В осуществляется намного быстрее, чем на устройство класса А, потому что не надо дожидаться очередного сеанса передачи данных. К сожалению, на сегодня практически нет свободного серверного программного обеспечения с поддержкой устройств класса В.

Более подробное описание работы сети LoRaWAN можно найти в спецификации и региональных требованиях, предоставляемых международной сертифицирующей организацией LoRa Alliance по запросу. Эти документы также можно найти и в свободном доступе [3, 4]. LoRa Alliance является единственной сертифицирующей организацией LoRa-устройств и подтверждает их соответствие действующей LoRaWAN-спецификации. Более подробно процедура сертификации описана в регламентирующем документе [5]. LoRa Alliance также определяет присвоение клиентскому устройству уникального идентификатора `devaddr` из пула, выделенного для каждого производителя — члена организации. Следует заметить, что наличие у LoRaWAN-модуля сертификата LoRa Alliance не освобождает от дополнительной сертификации как самого модуля, так и изделия, содержащего этот модуль, в стране использования в соответствии с региональными требованиями к радиопередающим устройствам.

Преимущество сертифицированных устройств — это возможность работы в публичной глобальной сети LoRaWAN, развертываемой при участии LoRa Alliance во многих странах мира. Альтернативой является построение корпоративных локальных LoRaWAN-сетей с использованием клиентских устройств, выполненных на базе радиомодулей физического уровня LoRa, управляемых микроконтроллером с пользовательским программным обеспечением, включающим LoRaWAN-стек. Как для клиентских устройств, так и для серверов доступно свободное программное обеспечение, позволяющее создавать собственные решения, использующие технологию LoRaWAN.

Рассмотрим основные элементы, составляющие инфраструктуру сети LoRaWAN.

Наиболее многочисленными являются клиентские устройства, в идеале — максимально дешевые и с малым энергопотреблением. Самый простой путь — использование готовых модулей, имеющих UART-командный интерфейс для настройки, управления и обмена данными. Примерами являются RN2483 от Microchip и UM801LP от Manthink. Такие модули сертифицированы LoRa Alliance

и могут работать в публичных сетях, в том числе предоставляющих бесплатный сервис в тестовом режиме.

Если планируется развертывание собственной корпоративной сети в регионах, где публичные сервисы пока недоступны, то наиболее оптимальным решением будет использование открытого LoRaWAN-стека и реализация транспортного уровня в единственном микроконтроллере, обеспечивающем функционирование клиентского устройства в целом. Известны две свободные реализации LoRaWAN-стека: версия от производителя чипсетов, компании Semtech [6], и альтернативная — от компании IBM [7]. На наш взгляд, более оптимальным является стек от IBM. Он поддерживает клиентские устройства класса В, его код более компактен и менее требователен к ресурсам контроллера. К сожалению, в мае 2017 г. компания IBM объявила о прекращении поддержки данного продукта (на сегодня остается доступной версия 1.6), но авторы заявили о готовности неофициально поддерживать разработку в открытом сообществе.

Скомпилированный код LMic v1.6 занимает около 20 кбайт и требует около 2 кбайт оперативной памяти. Исходная версия стека содержит демонстрационные проекты, выполненные в средах разработки Keil и IAR для микроконтроллера STM32L156C6 (Cortex F3 на 32 МГц, 32 кбайт ПЗУ, 10 кбайт ОЗУ). Код выполнен на С, использует только целочисленную арифметику, требует два цифровых выхода, три входа (с возможностью генерации прерываний) и таймер. Нам удалось успешно портировать стек в бюджетный микроконтроллер M058SEAN от компании Nuvoton (Cortex M0 на 50 МГц, 32 кбайт ПЗУ, 4 кбайт ОЗУ). Процесс портирования довольно простой и заключается в реализации в файле `hal.c` стандартных функций доступа к указанной выше периферии конкретного микроконтроллера.

Наиболее дорогостоящими элементами LoRaWAN-сети являются шлюзы. Основная задача шлюза — принять данные от клиентских устройств и передать их на сервер, используя интернет-подключение, а также ретранслировать данные, отправляемые сервером, в эфир. Функционально шлюз представляет

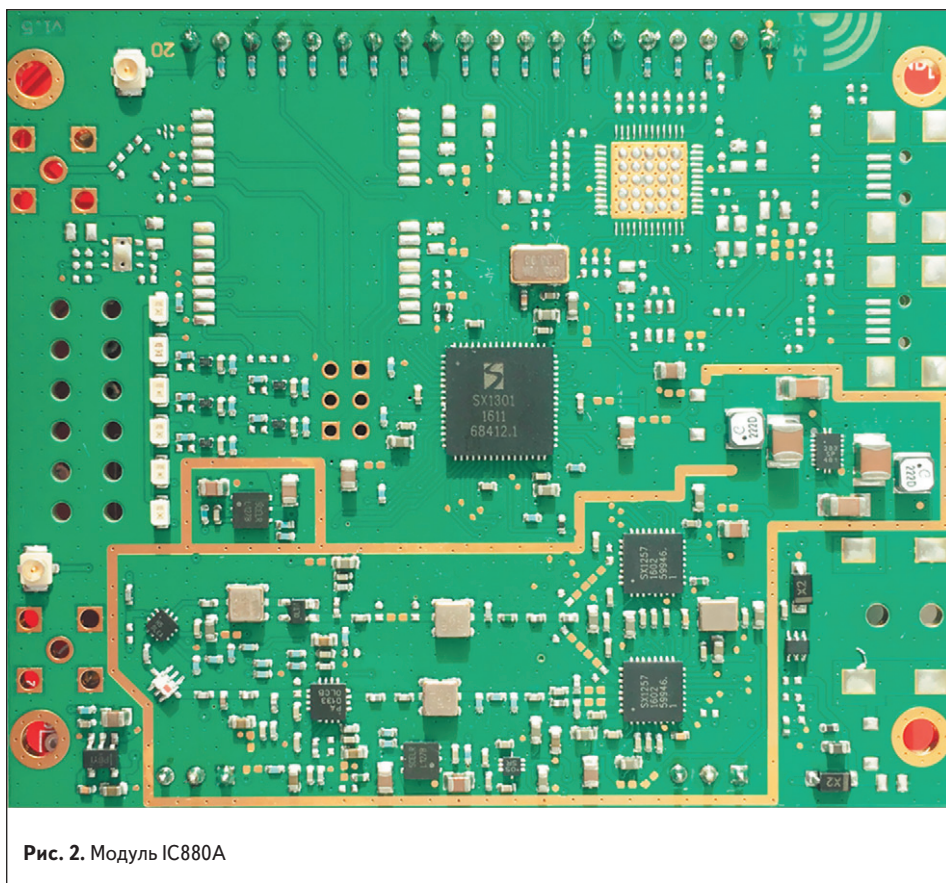


Рис. 2. Модуль IC880A

собой многоканальный приемопередатчик. Радиоканал обычно выполняется на двух чипсетах SX1257, а обработка производится сигнальным процессором SX1301. Подобный шлюз способен одновременно обрабатывать до восьми LoRa-пакетов в своем рабочем диапазоне частот. Более мощные модели содержат больше приемопередатчиков и процессоров, что позволяет расширить рабочий диапазон и увеличить производительность шлюза. Как показала практика, надежное функционирование корпоративной сети вполне могут обеспечить даже простые шлюзы с одним процессором на борту. Масштабировать сеть лучше путем увеличения количества шлюзов, равномерно распределяя их по охватываемой площади, чем за счет увеличения производительности единственного шлюза.

Представленные сегодня на рынке шлюзы являются, в основном, дорогими профессиональными устройствами, как правило, в IP67-исполнении, предназначенными для непосредственного монтажа на радиомачтах или крышах зданий с целью максимального покрытия региона. Можно выделить модели Wirnet Station от Kerlink и GDOx02 от Manthink — высокопроизводительные внешние шлюзы, предварительно настроенные для быстрого развертывания корпоративной LoRaWAN-сети.

Сравнительно высокая стоимость таких шлюзов сориентировала рынок на поиск альтернативных решений, в качестве которых были представлены модули, реализующие лишь радиоканал шлюза и требующие внешнее устройство управления (обычно компактный одноплатный Linux-компьютер), обеспечиваю-

щее функционирование радиоканала и коммуникацию с сервером через Интернет. Линейка подобных шлюзов сегодня представлена моделями IC880A от IMST и RHF0M31 от RisingHF. Интегрированный радиоканал с цифровым сигнальным процессором собран на небольшой плате и управляется через стандартный SPI. Тестировался модуль IC880A (рис. 2) совместно с бюджетным одноплатным компьютером Orange Pi PC от компании Xunlong. Физическое подключение осуществлялось по четырехпроводному SPI и линии сброса (табл. 1). Обе платы питались от общего источника 5 В/2 А, ток потребления IC880A достигал 250 мА, а Orange Pi PC — 1000 мА на пиках нагрузки.

Для удобства тестирования в одноплатном компьютере использовалась операционная система (ОС) Debian_xfce_desktop, предоставляющая графический интерфейс через HDMI-видео. Образ ОС [8] был записан на бюджетную microSD-карту Kingston 4GB Class 4 (SDC4/4GBSP) с реальной скоростью записи около 10 Мбайт/с. После логина (*root*

Таблица 1. Подключение радиомодуля LoRa-шлюза IC880A к одноплатному компьютеру Orange Pi PC

Сигнал	Номер вывода IC880A	Номер вывода Orange Pi PC
Reset (I07)	13	29
SCLK	14	23
MISO	15	21
MOSI	16	19
NSS	17	24
+5B	21	4
GND	22	6

с паролем *orangepc*) выполнялись стандартные процедуры установки размера раздела файловой системы (в соответствии с используемой SD-картой) и обновления информации о пакетах, содержащихся в репозиториях Debian:

```
/usr/local/bin/fs_resize
apt-get update
```

Для управления физической линией сброса IC880A удобно использовать стандартную библиотеку доступа к интерфейсу ввода/вывода компьютера — WiringOP, устанавливаемую с github:

```
git clone https://github.com/
zhaolei/WiringOP.git -b h3
cd WiringOP
chmod +x ./build
./build
cd..
```

Для управления радиоканалом IC880A использовался драйвер от производителя платы, также доступный на github:

```
git clone https://github.com/Loranet/
lora_gateway
cd lora_gateway
make
cd..
```

И, наконец, для обеспечения обмена с LoRaWAN-сервером использовалось приложение *packet_forwarder*:

```
git clone https://github.com/Loranet/
packet_forwarder
cd packet_forwarder
make
cd..
```

Для запуска данного приложения был создан скрипт *start_pfd.sh*, содержащий команды:

```
sleep 10
gpio mode 7 out
gpio write 7 1
sleep 0.5
gpio write 7 0
sleep 0.5
cd /root/packet_forwarder/lora_
pkt_fwd/
./lora_pkt_fwd
exit 0
```

Пауза в начале скрипта необходима для корректной работы в режиме автозапуска: так как Orange Pi PC не имеет аппаратных часов реального времени, то после старта компьютера перед запуском шлюза необходимо подождать обновления времени через Интернет средствами NTP. В дальнейшем автозапуск программного обеспечения шлюза обеспечивался путем создания по образцу стандартного скрипта LoRa в папке *init.d* и при необходимости управлялся командами:

```
update-rc.d lora defaults
update-rc.d -f lora remove
```

После установки программного обеспечения необходимо в файле `/root/packet_forwarder/lorapkt_fwd/global_conf.json` в параметре `server_address` указать IP-адрес или доменное имя LoRaWAN-сервера, с которым будет работать шлюз. Также, в случае возможного использования нескольких шлюзов, необходимо назначить каждому из них уникальный MAC-адрес в файле `/root/packet_forwarder/lorapkt_fwd/local_conf.json`. После выполнения указанных операций шлюз будет полностью готов к работе в составе корпоративной сети.

«Сердцем» LoRaWAN-сети является сервер, поддерживающий соединение с одним или несколькими шлюзами и управляющий пулом клиентских устройств, обеспечивая прием данных и помещая их в пользовательский интерфейс. Для тестирования мы использовали сервер с открытым кодом, поддерживающий неограниченное количество шлюзов и клиентских устройств, предоставляя данные клиенту или его приложению через веб-интерфейс, WebSocket или MQTT-протоколы [9]. Код сервера кроссплатформный и выполнен на языке Erlang (требует версию OTP 19 или выше). При тестировании сервер был запущен на компьютере Intel Pentium G3220 на 3 ГГц, 4 Гбайт ОЗУ под управлением 32-битной Windows 7. Для установки Erlang использовался инсталлятор Erlang OTP 20.0 Windows 32-bit Binary File [10]. Сам сервер был загружен в виде архива `lorawan-server-0.4.8.tar.gz` и с помощью WinRAR распакован в рабочую папку. Запуск сервера выполнялся непосредственно из папки `bin` скриптом `lorawan-server.bat`.

Для получения пакетов от шлюзов сервер использует UDP-порт 1680. Веб-интерфейс, предназначенный для настройки и контроля, доступен через TCP-порт 8080 (по умолчанию открыт доступ для пользователя `admin` с паролем `admin`). Настройка сервера заключается в добавлении шлюзов и клиентских устройств.

Для добавления шлюзов в веб-интерфейсе необходимо использовать меню **Infrastructure** → **Gateways** → **Create** и заполнить поля, отмеченные звездочкой: MAC (должен соответствовать MAC-адресу, заданному при настройке данного

шлюза), NetID (произвольный код сети, общий для всех шлюзов), TX Chain (обязательно «0»), Location (указать расположение данного шлюза на карте).

Для добавления клиентских устройств используется меню **Devices** → **Create** (для устройств, регистрирующихся в сети по процедуре OTAA) или **Nodes** → **Create** (для устройств, регистрирующихся по процедуре ABP, см. выше). Устройства, однажды успешно зарегистрировавшиеся по процедуре OTAA, будут автоматически добавлены в раздел **Nodes**.

Для обеспечения процедуры регистрации OTAA в меню **Devices** → **Create** для каждого добавляемого клиентского устройства необходимо указать его `Region=EU 863-870MHz` и `Application=WebSocket` (интерфейс, по которому данные с этого устройства будут доступны пользовательскому приложению). Параметры `devuid`, `appuid` и `appkey` должны соответствовать таковым в добавляемом клиентском устройстве.

Для обеспечения процедуры ABP в меню **Nodes** → **Create** необходимо также указать `Region=EU 863-870MHz` и `Application=WebSocket`. Параметры `devaddr`, `nwkskey` и `appskey` должны соответствовать таковым в добавляемом клиентском устройстве.

После настройки последние данные, полученные от активных клиентских устройств, можно увидеть в меню **Received Frames**. Для получения клиентским приложением принятых данных через интерфейс WebSocket при добавлении устройств можно указать необязательный параметр `Group=ABC` и подключаться к серверу по ссылке `ws://server:8080/ws/groups/ABC/json`. Данные, принятые от участников группы ABC, будут передаваться подключившемуся клиентскому приложению в виде json-пакетов, содержащих, кроме самих данных, дополнительную информацию о клиентском устройстве (уровень ошибок, текущие параметры модуляции, используемый шлюз и т. п.).

В итоге нами была развернута тестовая бюджетная корпоративная сеть LoRaWAN, состоящая из малопотребляющих клиентских устройств, периодически отсылающих данные

телеметрии, шлюза и сервера, управляющего сетью и предоставляющего полученные данные клиентскому приложению. Нами использовалось свободное программное обеспечение, адаптированное под конкретную пользовательскую задачу. Оборудование, необходимое для развертывания подобной сети, всегда доступно для заказа в нашей компании, а квалифицированные инженеры готовы поделиться своим опытом и оказать техническую поддержку в выборе, установке и адаптации программного обеспечения для решения вашей прикладной задачи. ■

Литература

1. Е. Рахно, С. Гаевский. Сети LoRaWAN — особенности украинского рынка // CHIP NEWS Украина. Инженерная микроэлектроника. 2016. № 10.
2. С. Гаевский. Чипсеты SX127x с радиоканалом LoRa: архитектура, управление и применение в системах связи // CHIP NEWS Украина. Инженерная микроэлектроника. 2017. № 3.
3. LoRaWAN Specification V1.0.2. LoRa Alliance, 2016. http://wiki.lahoud.fr/lib/exe/fetch.php?media=lorawan102-20161012_1398_1.pdf
4. LoRaWAN 1.0.2 Regional Parameters. LoRa Alliance, 2017. www.thethingsnetwork.org/forum/uploads/default/original/2X/d/dca7977549092f2c594a601caa739e1bacbee59.pdf
5. Certification Policies and Procedures. LoRa Alliance, 2015. www.ipaustralia.gov.au/sites/g/files/net856/f/certification_rules/1764592_170502_initial_rules.pdf
6. LoRaWAN endpoint stack implementation and example projects. Semtech, 2017. <https://github.com/Lora-net/LoRaMacnode>
7. IBM-LMIC: LoRaWAN MAC on embedded systems. IBM, Zurich, 2017. <https://github.com/mcci-catena/ibm-lmic>
8. Orange Pi One: Debian XFCE 2016-12-12. <http://piboard.io/topic/6/операционные-системы-для-orange-pi>
9. Compact server for private LoRa networks. <https://github.com/gotthardp/lorawan-server>
10. Erlang OTP 20.0 Windows 32-bit Binary File. www.erlang.org/downloads