

Интеграция модулей Telit в Azure IoT Hub.

Практическая реализация

Облачный сервис Microsoft Azure предлагает «Центр «Интернета вещей» (Azure IoT Hub) для управления устройствами, собирающими данные о физическом мире, включая окружающую среду, различного рода процессы (в том числе технологические) и самого человека. В статье на практическом примере рассматриваются возможности совместной работы модулей сотовой связи Telit со службой Azure IoT Hub.

Алексей Рудневский
rudnevsky.a@atoma.spb.ru

Компания Telit производит модули передачи данных, предназначенные, с одной стороны, для интеграции с устройствами «Интернета вещей» (Internet of Things, IoT), а с другой — для взаимодействия с облачными сервисами IoT. Рассмотрим возможности совместной работы модулей сотовой связи Telit (в частности UE866-EU) с облачным сервисом Microsoft Azure — «Центром «Интернета вещей» (Azure IoT Hub) на практическом примере.

Сервис Microsoft Azure IoT Hub позволяет получать данные практически с любого количества различных устройств, реализующих на практике принципы IoT, в реальном времени

и отправлять команды на устройства, а также управлять устройствами.

Для начала работы с Azure IoT Hub необходимо иметь учетную запись Microsoft. Получение такой учетной записи тривиально и выходит за рамки данной статьи. После этого необходимо зайти на портал Microsoft Azure [1] и создать собственный «Центр «Интернета вещей» (рис. 1). В нашем примере он будет называться RuCenter, соответственно, имя узла будет *RuCenter.azure-devices.net* (оно потребуется чуть позже, при привязке устройства). Далее необходимо создать само устройство в «Центре», для этого проще всего воспользоваться «Обозревателем устройств» (рис. 2). При желании, Microsoft предлагает использовать

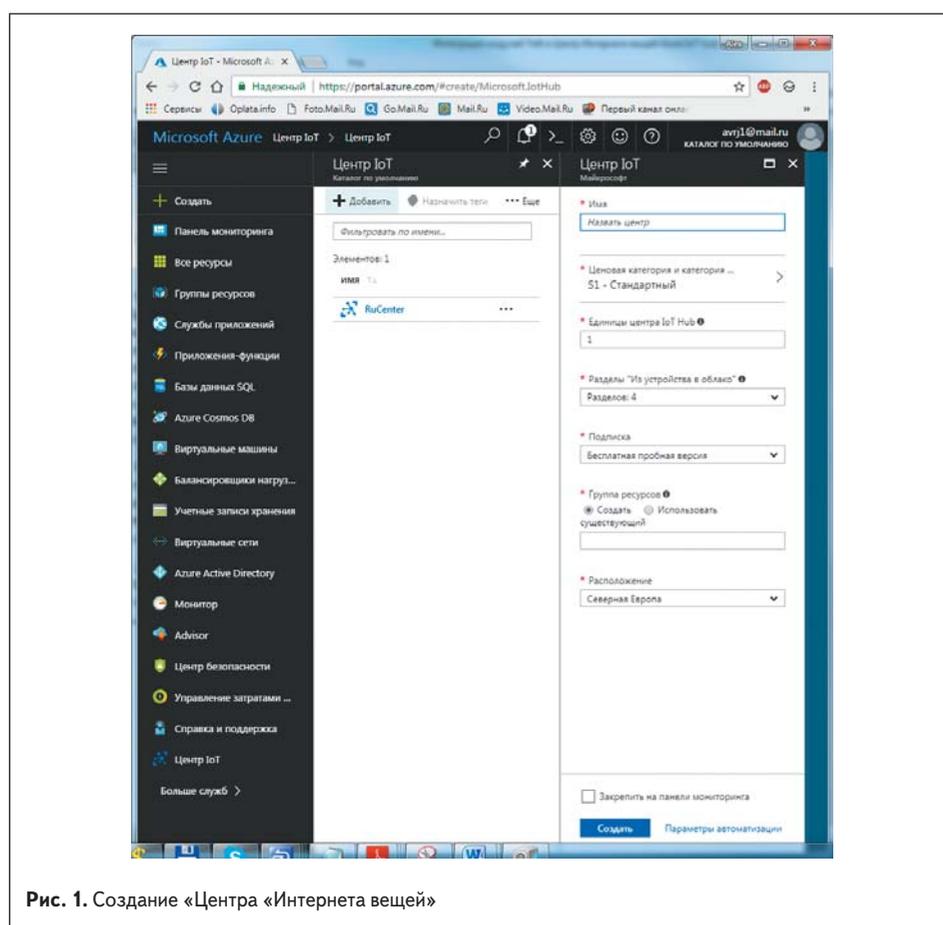


Рис. 1. Создание «Центра «Интернета вещей»

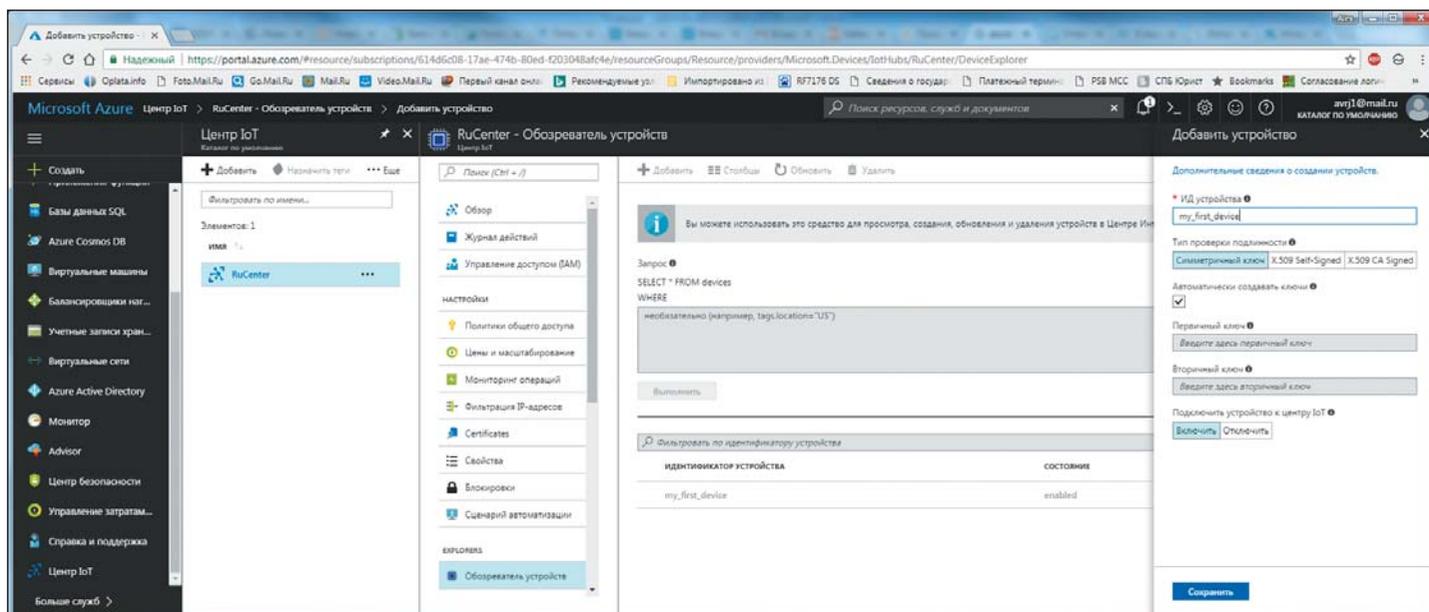


Рис. 2. Создание устройства в «Центре «Интернета вещей»

вместо веб-интерфейса приложение Device Explorer, выполняющее аналогичные функции. Назовем наше устройство *my_first_device*. После создания устройства система автоматически создаст два симметричных ключа общего доступа, которые также потребуются для привязки устройства. После этого «Центр» уже готов к передаче сообщений в устройство и приему сообщений из устройства в IoT Hub. Для контроля переданных и принятых сообщений может использоваться несколько различных методов. Самым простым является использование уже упомянутого приложения Device Explorer, где можно непосредственно увидеть сами сообщения. Для того чтобы Device Explorer привязался к созданному ранее «Центру «Интернета вещей», необходимо его сконфигурировать. Для этого на вкладке **Configuration** необходимо ввести строку подключения к Azure IoT Hub, полученную из веб-интерфейса (рис. 3). На этом первичное конфигурирование IoT Hub заканчивается, и можно переходить к собственно устройству.

В качестве устройства будем использовать модуль UMTS/HSPA Telit UE866-EU [2]. Этот самый малогабаритный в мире модуль UMTS/HSPA с возможностью работы как в 3G-, так и в 2G-сетях уже рассматривался нами ранее [3]. Российский дистрибьютор Telit выпускает оригинальную отладочную плату Telit UE866-EU + SE868V3 bundle [4] (рис. 4), позволяющую испытать большинство функций UE866-EU без использования дополнительного оборудования (базовой платы, блока питания и пр.).

Существенным преимуществом модулей передачи данных Telit является то, что все они содержат в составе прошивки протокол MQTT [5] — один из протоколов IoT, который поддерживается также и Azure IoT Hub. Кроме того, во все модули сотовой связи Telit, включая UE866-EU, имплементирован протокол SSL/TLS, требующийся для обеспечения безопасности передаваемых данных. Основываясь на этих наработках, специалисты Telit разработали приложение AppZone [6], работающее непосредственно с Azure IoT Hub и управляемое со стороны модуля UE866-EU AT командами. AT-команды UE-866 для работы с Azure IoT Hub приведены в таблице 1.

В качестве примера отправим тестовое сообщение из UE866-EU в Azure IoT Hub. Ниже приведен лог AT-команд модуля, необходимый для инициализации соединения с «Центром» и передачи сообщения от устройства:

```
AT+CGDCONT=1,"IP","INTERNET" // параметры PDP контента, необходимые для активации пакетного соединения
OK
AT#SGACT=1,1 // активация пакетного соединения
#SGACT: 100.82.125.252 // IP-адрес, выданный оператором
AT#AIHCFG="RuCenter","azure-devices.net","my_first_device","sZKk/1pio2CU2qL4fegT2WoOnpnmVwERi7y13hYZWo=",2 // конфигурирование устройства
OK
```

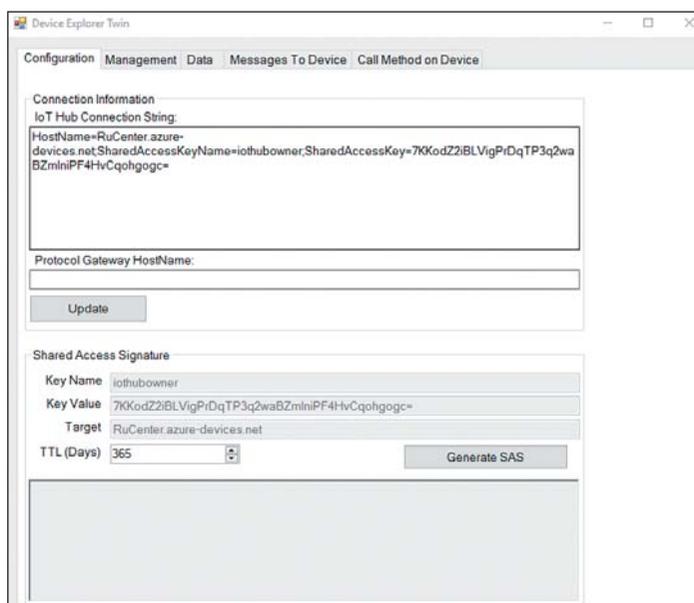


Рис. 3. Конфигурирование Device Explorer



Рис. 4. Отладочная плата Telit UE866-EU + SE868V3 bundle

```
at#aihconn=1 // установка соединения с «Центром «Интернета вещей»
OK
#AIH: CONNECTED // соединение установлено
at#aihsnd=6 // отправка сообщения длиной 6 байт
> test77 // тело сообщения
OK
```

Полученное в IoT Hub сообщение отображается во вкладке **Data** приложения Device Explorer (рис. 5). Как видим, сообщение доставлено в том же виде, что было отправлено из устройства.

Сообщения из Azure IoT Hub, отправленные на устройство, отображаются в виде URC (Unsolicited Result Code):

```
#AIH:<msgid>,<correlationId>,<userDefinedContentType>,<userDefinedContentEncoding> [>,<key1>,<value1> [>,<key2>,<value2>] [>,<DATA>,<user_message></pre>
</div>
<div data-bbox="43 269 480 307" data-label="Text>
<p>Все поля URC идентичны свойствам сообщения, которые можно увидеть в приложении Device Explorer (рис. 6). В нашем примере на устройство приходит строка:</p>
</div>
<div data-bbox="58 319 469 332" data-label="Code-Block>
<pre>#AIH: 67b73dec-22f4-422b-800d-36a4ddf5d3d3,<null>,<null>,<null>,<DATA>,<Hello, World!</pre>
</div>
<div data-bbox="43 344 479 369" data-label="Text>
<p>Как и в случае передачи из устройства в IoT Hub, сообщение передано без искажений.</p>
</div>
<div data-bbox="43 369 480 457" data-label="Text>
<p>Таким образом, Telit реализовал в своих модулях простой и удобный интерфейс взаимодействия с Azure IoT Hub. Реализация рассчитана на модули сотовой связи с поддержкой AppZone, но легко может быть портирована и на другие модули передачи данных — Bluetooth, Wi-Fi, LoRa и другие, поскольку поддержка MQTT и SSL имеется во всех модулях Telit. Это еще раз подтверждает лидерство Telit как производителя модулей для «Интернета вещей». ■</p>
</div>
<div data-bbox="43 478 146 494" data-label="Section-Header>
<h2>Литература</h2>
</div>
<div data-bbox="43 493 480 618" data-label="List-Group>
<ol>
<li>1. <a href="https://portal.azure.com">https://portal.azure.com</a>
<li>2. <a href="http://www.telit.com/wp-content/uploads/2017/09/Telit_UE866_Datasheet_AG.pdf">www.telit.com/wp-content/uploads/2017/09/Telit_UE866_Datasheet_AG.pdf</a>
<li>3. Рудневский А. Новая концепция унификации Telit: «гнездовая» система в семействе xE866 // Беспроводные технологии. 2016. № 2.</li>
<li>4. <a href="http://atoma.spb.ru/catalog/4118/otladochnaja-plata-telit-ue866-eu-se868v3-bundle">http://atoma.spb.ru/catalog/4118/otladochnaja-plata-telit-ue866-eu-se868v3-bundle</a>
<li>5. <a href="http://i-o-t.ru/protokol-mqtt/>http://i-o-t.ru/protokol-mqtt/</a>
<li>6. Рудневский А. AppZone от Telit: практическое создание приложений // Беспроводные технологии. 2016. № 1.</li>
</ol>
</div>
<div data-bbox="43 638 475 936" data-label="Image>
<img alt="Screenshot of Device Explorer Twin showing received messages in the Data tab. The Event Hub is RuCenter and the Device ID is my_first_device. The message content is test77. The output pane shows system properties like correlation-id, content-type, and content-encoding.">
</div>
<div data-bbox="53 915 297 929" data-label="Caption>
<p>Рис. 5. Сообщения, принятые из устройства</p>
</div>
<div data-bbox="489 71 886 85" data-label="Caption>
<p>Таблица 1. AT-команды UE866-EU для работы с Azure IoT Hub</p>
</div>
<div data-bbox="489 84 933 613" data-label="Table>
<table border="1">
<thead>
<tr>
<th colspan="2">#AIHCFG – конфигурирование клиента «Центра «Интернета вещей».</th>
</tr>
</thead>
<tbody>
<tr>
 AT#AIHCFG=<i>{<iothub_name>,<iothub_suffix>,<device_id>,<device_shared_key></i> | Параметры:                      <i>{<iothub_name>}</i> – имя узла IoT Hub (первая часть). В нашем случае полное имя узла RuCenter.azure-devices.net, поэтому первая часть будет RuCenter.                      <i>{<iothub_suffix>}</i> – вторая часть имени узла. В примере выше будет azure-devices.net.                      <i>{<device_id>}</i> – имя устройства. Ранее было создано устройство my_first_device.                      <i>{<device_shared_key>}</i> – первичный (или вторичный) симметричный ключ общего доступа. || AT#AIHCFG? | По команде возвращается текущее состояние конфигурации в формате: #AIHCFG: <i>{<iothub_name>,<iothub_suffix>,<device_id>,<device_shared_key></i> |
| AT#AIHCFG=? | Возвращается диапазон допустимых значений параметров. |
| |  |  | | --- | --- | | AT#AIHCONN=<i>{<op></i> | <i>{<op>}</i> – операция: «0» – разорвать соединение, «1» – установить соединение. Примечание 1: перед выполнением команды должно быть произведено конфигурирование командой AT#AIHCFG.                      Примечание 2: PDP-контент должен быть активным перед выполнением команды. Он может быть активирован командой AT#SGACT=1,1 (предварительно нужно сконфигурировать контент командой AT+CGDCONT). После установления соединения модуль формирует сообщение #AIH: CONNECTED.                      После разрыва соединения модуль формирует сообщение #AIH: DISCONNECTED. | | AT#AIHCONN? | По команде возвращается текущее состояние соединения в формате: #AIHCONN: <i>{<status>,<status>}</i>                      «0» – неправильный PDP-контент;                      «1» – соединение неактивно;                      «2» – попытка соединения;                      «3» – соединение установлено;                      «4» – готовность к соединению. | | AT#AIHCONN=? | Возвращается диапазон допустимых значений параметров. | | |  |  | | --- | --- | | AT#AIHSND=<i>{<data_len></i> | Команда отправляет данные в «Центр «Интернета вещей», соединение с которым было установлено ранее. <i>{<data_len>}</i> – длина сообщения в байтах, допустимый диапазон 1–1500. Модуль отвечает приглашением > и ожидает приема данных. По достижении заданной длины сообщение автоматически отправляется. | | AT#AIHSND=? | Возвращается диапазон допустимых значений параметров. | | |  |  | | --- | --- | | AT#AIHVER | По команде возвращается версия приложения, записанная в модуль. | | AT#AIHVER? | Идентично AT#AIHVER | | | | | | |

```

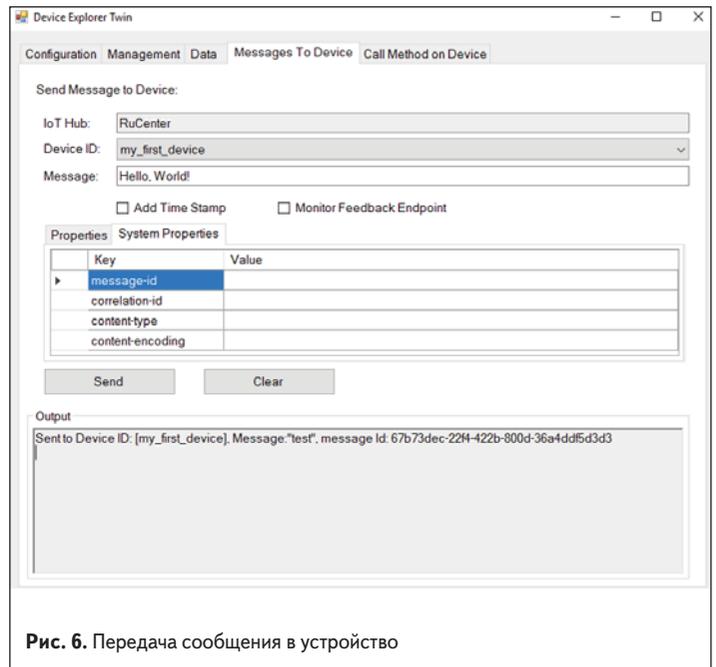


Рис. 6. Передача сообщения в устройство