Для того, чтобы подключить Galileo к компьютеру, есть несколько опций - Ethernet, USB и отладочный 6-пиновый кабель. Самым простым и доступным является подключение по Ethernet.

Проблемой является то, что по умолчанию сеть на Galileo не настраивается, и увидеть ее даже после подключения кабелем не получится.

Есть два решения этой проблемы:

1. Подключить Galileo к роутеру, который раздаст ей IP-адрес (по Ethernet или Wi-Fi). Это самый удобный и функциональный способ - плата сможет выходить в интернет.
2. Подключить плату к компьютеру и раздать ей IP-адрес с помощью собственного DHCP-сервера. Можно использовать OpenDHCPServer. Для его настройки необходимо сконфигурировать интерфейс, к которому будет подключаться Galileo, на статический адрес (например, 192.168.0.1) и добавить этот адрес для прослушки в конфигурацию OpenDHCPServer, после чего запустить его как сервис либо standalone. В этом случае для того, чтобы плата увидела интернет, нужно проводить дополнительные настройки.

После получения IP-адреса можно подключиться к плате по SSH.

На Galileo установлен образ Yocto Linux, который имеет свою специфику, но поставить на него необходимое ПО можно.

Для этого нужно подключиться по SSH.

Мы будем использовать [Putty](http://putty.org).

1. Первым шагом надо определить IP-адрес машины. Если подключили к роутеру, можно посмотреть на администраторской панели роутера, какой IP-адрес был роздан DHCP. Если OpenDHCPServer, то в консоли увидите, что IP роздан.
2. Вводим нужные данные в Putty.

Machine generated alternative text:
PuTTY Configuration 
Cat egory 
Logging 
Terminal 
Keybo ard 
Features 
Window 
Behaviour 
Translation 
Colours 
x 
Basic options for your Pu TTY session 
Specify the destination you want to connect to 
Host Name (or IP address) 
Connection type 
C) Raw C) Telnet C) Rlogin @SSH C) 
Load save or delete a stored session 
Load 
Saved Sessions 
Galileo 
Defautt Settings 
Galileo

1. Логин root, пароля по умолчанию нет.

Machine generated alternative text:
- PUTTY 
as: root

1. Начинаем настраивать экосистему. В данной инструкции будем использовать Python и устанавливать Azure SDK for Python, но вы можете установить любой другой доступный SDK либо самостоятельно написать необходимый код.

Загружаем Azure SDK for Python:

git clone <https://github.com/Azure/azure-sdk-for-python.git>

cd azure-sdk-for-python/

python setup.py install

Устанавливаем актуальную дату:

rdate -s time.nist.gov

(надо выполнять, чтобы не получить ошибки <Error><Code>401</Code><Detail>ExpiredToken: при отправке сообщения в очередь Azure)

Устанавливаем и обновляет необходимый софт:

opkg update

opkg install openssl

opkg install curl

Устанавливаем PIP - он понадобится для удобного управления модулями Python.

Процесс установки может быть достаточно долгим (около 5-10 минут), при этом может быть не виден. Не обрывайте его.

curl <https://bootstrap.pypa.io/get-pip.py> -k | python

1. Создадим файл для нашего примера. Данная очередь предназначена для тестирования.

from azure.servicebus import ServiceBusService,Message, Queue

import base64

service\_namespace = 'demostand-ns'

key\_name = 'AuthRead'

key\_value = 'ax2UKT7ppf0ti8uslO8p+HSF4qfprqLKTdY893Io+to='

queue\_name = 'galileo'

message = Message(base64.b64encode('Hello World'))

sbs = ServiceBusService(service\_namespace,

shared\_access\_key\_name=key\_name,

shared\_access\_key\_value=key\_value)

sbs.send\_queue\_message(queue\_name,message)

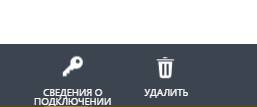
1. Если вы хотите создать новую очередь, вам нужно создать новый аккаунт Microsoft Azure и очередь. В пространстве имен можно создавать другие очереди и др.

Machine generated alternative text:
C03AATb 
nail 
CllYXbA MY11bTMMELU4A 
CllYXEEHAß UMHA 
MOSW1bHbW1 OXBAT 
VISUAL STUDIO ONLINE 
CllYXbA BIZTALK 
CETb LOCTABKM 
conoxworo (CDN) 
ABTOMATW3AUMR 
O"EPEAb 
PA3AE11 
PETPAHCJ19U"9 
KOHUEHTPATOP YBEAOMnEHM',1 
KOHUEHTPATOP CObb1TW1 
x 
"MH OUEPEA',I 
SHCTPOE C03AAWIE 
HACTPAMBAEMOE 
C03AAWIE 
PEWIOH 
UeHTpanEHaR qacTS CLEA 
"MH nPOCTPAHCTBA NEH 
.servicebus.windows.net 
C03AATb OHEPEAb

1. Перейдите на страницу созданной очереди.

Machine generated alternative text:
BCE 
MMR 
testqueue 
OLIEPELVI 
COCTORHAE 
AKTMBHO 
KOHUEHTPATOPSI COSSIT',WI 
MACIITASI,IPOBAHVIE 
HACTPOI,IKVI 
BPEMR nOCfiELHEro 
Heaoc-ryn HO 
TEKYLUMI,I PA3MEP 
MAKCMMAIISHSIVI PA3M.. 
16 
OUEPELI,I

1. Нажмите "Сведения для подключения".



1. Для того, чтобы взаимодействовать с очередью, нам нужно создать правило доступа. Нажмите на "Щелкните здесь для настройки".

Machine generated alternative text:
CBeael-lMR no nocyny K noxnoqe'-l?lh0 
Vlcnonsay".Te CBeaeH'-1F O noAKn-oveH'-w noAKDFoveHMR K "testqueue" 
HeT HaCTpoeHHbIX SAS CYLI_ÅHOCTVI. 
U.ÅEJIKHVITE 3AECb HACTPOÉIWI -Y

1. Введите в нижний блок название правила fullaccess, после чего выберите уровень доступа "Управление" - полный доступ к очереди. Нажмите "Сохранить".

Machine generated alternative text:
nom.1TMWl 061_qero aocvna 
MMR 
fullaccess 
MMA HOBOÜ nonMTMKM 
PABPEU_IEHMF 
YnpaBneHMe, OTnpaEMTB, npocnyunqeawe

1. Снова нажмите на "Сведения для подключения" и скопируйте название и ключ доступа.

Machine generated alternative text:
CBeael-lV4R no nocyny K noxnoqeHPlh0 
Vlcnonsay".Te CBeaeH'-1F O noAKn-oveH'-w noAKDFoveHMR K "testqueue" 
p 
ns.servicebus.windows.net/;SharedAccessKeyNam• 
MMR 
fullaccess 
CTPOKA nonKJIH)UEHVIR

Теперь вы можете заменить тестовые значения в коде на новые значения из вашей собственной очереди.

Для того, чтобы запустить пример, понадобится установить модуль:

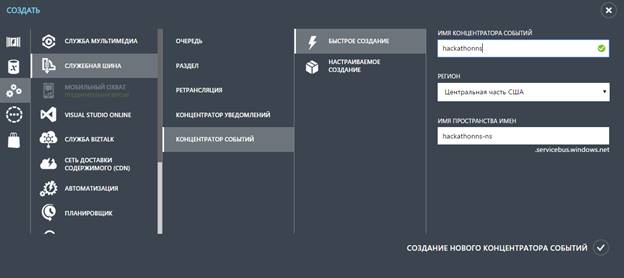
pip install python-dateutil

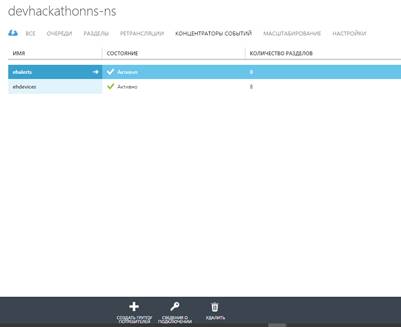
Теперь можно выполнить пример:

python 1.py

Пример будет посылать простые сообщения в очередь.

Аналогично происходит создание Event Hub (гораздо более подходящий для маленьких сообщений механизм для сбора сообщения).





Для управления и доступа к Event Hubs используется собственный механизм безопасности.

Кроме этого, вам нужно создать группу потребителей, используя которую, вы сможете потом забирать сообщения из очереди.

Пример для отсылки сообщений в Event Hub на Python.

Данный пример:

* 1. Создает клиента для отправки в Event Hub
  2. Формирует JSON-пакет из нескольких полей и значений
  3. Отправляет JSON-пакет в Event Hub в бесконечном цикле, выводя в консоль HTTP-код 201. Это нормальное поведение. В случае ошибки будет другой код.

Вы можете изменить код так, как вам необходимо, включив сбор данных с датчиков либо генерируя поток случайных данных, если у вас нет возможности использовать датчики.

#!/user/bin/python

import sys

import azure

import socket

from azure.servicebus import (

\_service\_bus\_error\_handler

)

from azure.servicebus.servicebusservice import (

ServiceBusService,

ServiceBusSASAuthentication

)

from azure.http import (

HTTPRequest,

HTTPError

)

from azure.http.httpclient import \_HTTPClient

class EventHubClient(object):

def sendMessage(self,body,partition):

eventHubHost = "devhackathonns-ns.servicebus.windows.net"

httpclient = \_HTTPClient(service\_instance=self)

sasKeyName = "D1"

sasKeyValue = "urummVtx0vv7RkBhv8aloLOpeSaOK79+ghYgwEoAWt0="

authentication = ServiceBusSASAuthentication(sasKeyName,sasKeyValue)

request = HTTPRequest()

request.method = "POST"

request.host = eventHubHost

request.protocol\_override = "https"

request.path = "/ehdevices/publishers/" + partition + "/messages?api-version=2014-05"

request.body = body

request.headers.append(('Content-Type', 'application/atom+xml;type=entry;charset=utf-8'))

authentication.sign\_request(request, httpclient)

request.headers.append(('Content-Length', str(len(request.body))))

status = 0

try:

resp = httpclient.perform\_request(request)

status = resp.status

except HTTPError as ex:

status = ex.status

return status

class EventDataParser(object):

hubClient = EventHubClient()

parser = EventDataParser()

hostname = socket.gethostname()

body = "{ \"temp1\" : 0.0, \"temp2\" : 0}"

while True:

hubStatus = hubClient.sendMessage(body,hostname)

# return the HTTP status to the caller

print hubStatus

Вы можете получить сообщения из приложения с помощью API, из Service Bus Explorer либо Stream Analytics - удобного сервиса для потоковой обработки данных.

Создайте Stream Analytics с помощью инструкции - <http://azure.microsoft.com/ru-ru/documentation/articles/stream-analytics-get-started/>

Подставив вместо имеющихся данных по Event Hub ваши данные, вы сможете перенаправить поступающий поток данных от устройства туда, куда вам необходимо - в другую очередь Event Hub, в хранилище (например, в виде JSON) и др.

Оттуда вы можете достать эти данные и обработать их с помощью любых средств - например, HDInsight или внутри приложения.