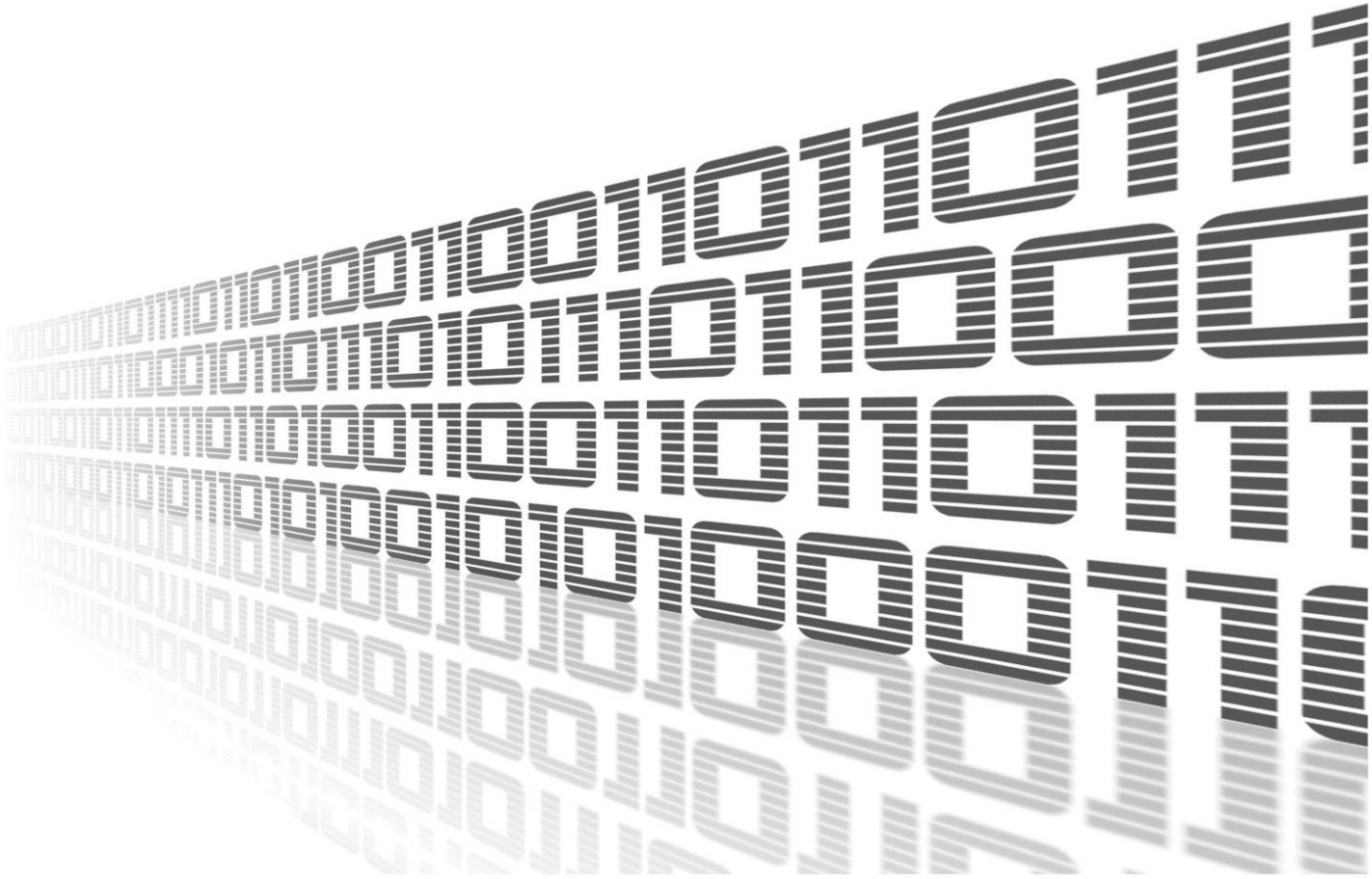
ADVANTECH

RouterApp

Azure IoT SDK Python



Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0008-EN, revision from 11th October, 2023.

Used symbols

Danger – Information regarding user safety or potential damage to the router.

Attention – Problems that can arise in specific situations.

Information – Useful tips or information of special interest.

Example – Example of function, command or script.

Contents

List of Figures

List of Tables

1. Changelog

1.1 Azure IoT SDK Python Changelog

v2017-10-09 (2017-10-24)

- First release.

v2018-02-20 (2018-02-20)

- Upgraded SDK python to version release_2018_02_20.

v2018-02-20 (2019-01-02)

- Added licenses information.

v2018-02-20 (2020-10-01)

- Updated CSS and HTML code to match firmware 6.2.0+.
- Linked statically with OpenSSL 1.0.2u.
- Linked statically with libcurl 7.72.0.
- Linked statically with zlib 1.2.11.

v2018-02-20 (2020-11-12)

- Upgraded python3 to 3.7.9.

v1.0.0 (2021-06-0)

- Fixed version string.

2. Router App Description

2.1 Azure IoT

Azure IoT is Microsoft's end-to-end IoT platform. Microsoft offers products like Azure IoT Hub to easily and securely connect your IoT devices to Microsoft Azure.

2.2 SDK for Python

It is possible to connect the devices to Azure IoT using open source device SDKs offered by Microsoft. These SDKs support multiple operating systems, and multiple programming languages, including Python. One of them – *Azure IoT Hub Device SDK for Python* – was implemented as a standalone Router App for *Advantech* routers called *Azure IoT SDK Python*.

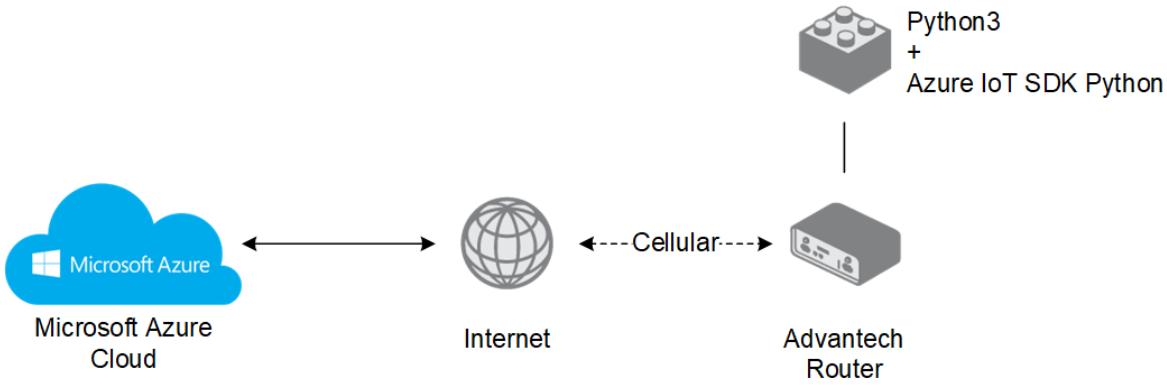


Figure 1: Router with *Python3* and *Azure IoT SDK Python* installed to connect Azure Cloud

Please note that there are two versions of this router app available, *Azure IoT SDK Python* and *Azure IoT SDK Python3 API Version 2*. The original version is still available due to the compatibility reasons and still can be used for existing implementation. *Azure IoT SDK Python version 2* was completely reworked to Python. The original version and version 2 are not compatible.

For more information, including features of the device SDK, see:
https://github.com/Azure/azure-iot-sdk-python/tree/master/device
Note that only "device SDK" part of the Python SDK was implemented.

More complex README file for Python SDK is available here:
https://github.com/Azure/azure-iot-sdk-python

SDK for deprecated version 1 is still available here:
https://github.com/Azure/azure-iot-sdk-python/tree/v1-deprecated

The *Azure IoT SDK Python* router app is not installed on *Advantech* routers by default. It can be downloaded from icr.advantech.cz/user-modules. There is dependency for *Azure IoT SDK Python* router app to be installed in the router – follow the instructions in Chapter 2.3. See the *Configuration Manual*, chapter *Customization –> Router Apps*, for the description of how to upload a router app to the router.

2.3 Azure IoT SDK Python Dependency

It is necessary to install the *Python3* router app along with the *Azure IoT SDK Python* router app. *Python3* is required for *Azure IoT SDK Python* to work – it is the separated module and it can be used as a standalone Python3 for other purposes.

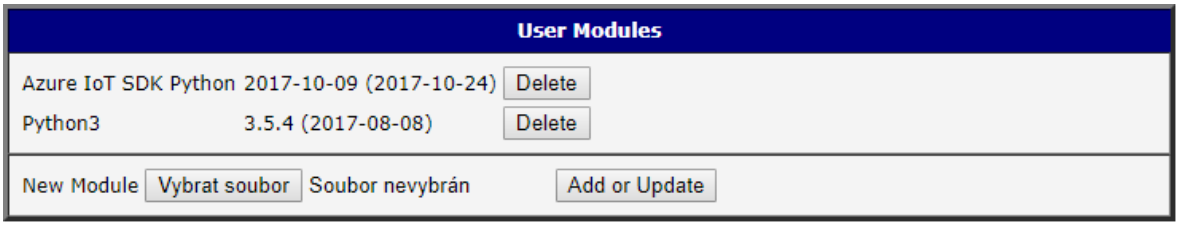


Figure 2: Python3 and Azure IoT SDK Python router apps installed

3. Available Python Modules

Installing *Python3* and *Azure IoT SDK Python* offers a set of standard and common Python modules, including these:

- os
- sys
- logging
- time
- datetime
- multiprocessing
- threading
- json
- uuid
- sqlite3
- textutils
- importlib
- shell
- compression
- subprocess
- tblib
- uuid

The full list of available Python modules can be obtained by typing the following command in the router's command line interface (available via SSH):

```
python3
```

The prompt will go to Python mode starting with ">>>". Go to Python help mode by typing:

```
help()
```

Now you are in the Python help mode starting with "help>" and you can type the following command for the full list of installed Python modules:

```
modules
```

See the example of output in the next Figure:

```
help> modules

Please wait a moment while I gather a list of all available modules...

CDROM               _weakrefset         heapq               shelve
DLFCN               abc                 hmac                shlex
IN                  aifc                html                shutil
TYPES               antigravity         http                signal
__future__          argparse            imaplib             site
_ast                array               imghdr              smtpd
_bisect             ast                 imp                 smtplib
_bootlocale         asynchat            importlib           sndhdr
_codecs             asyncio             inspect             socket
_codecs_cn          asyncore            io                  socketserver
_codecs_hk          atexit              ipaddress           spwd
_codecs_iso2022     audioop             itertools           sqlite3
_codecs_jp          base64              json                sre_compile
_codecs_kr          bdb                 keyword             sre_constants
_codecs_tw          binascii            linecache           sre_parse
_collections        binhex              locale              ssl
_collections_abc    bisect              logging             stat
_compat_pickle      builtins            lzma                statistics
_compression        bz2                 macpath             string
_crypt              cProfile            macurl2path         stringprep
_csv                calendar            mailbox             struct
_ctypes             cgi                 mailcap             subprocess
_ctypes_test        cgitb               marshal             sunau
_datetime           chunk               math                symbol
_decimal            cmath               mimetypes           symtable
_dummy_thread       cmd                 mmap                sys
_elementtree        code                modulefinder        sysconfig
_functools          codecs              multiprocessing     syslog
_hashlib            codeop              netrc               tabnanny
```

Figure 3: Example of listed available modules

3.1 Azure installation

Detailed information along with examples can be found here: https://github.com/Azure/azure-iot-sdk-python

1. Install Python3 with PIP into the router
2. Install python requirements – *Setuptools*, *azure-iot-device* (via routers CLI)

   ```
   pip3 install setuptools
   pip3 install azure-iot-device
   ```

3. Create a link for routers certificate:

   ```
   ln -s /etc/ssl/certs/ca-certificates.crt /usr/ssl/cert.pem
   ```

   (this will be permanently created) or include this line in your every Python script under async def main():

   ```
   os.environ["SSL_CERT_FILE"] = "/etc/ssl/certs/ca-certificates.crt"
   ```

   ```
   async def main():
       os.environ["SSL_CERT_FILE"] = "/etc/ssl/certs/ca-certificates.crt"
   ```

4. Create a Azure IoT Enviroment (Azure account, Azure IoT Hub, Device provisioning centre)
5. Create device in Azure IoT Hub and copy his Primary connection String into the clipboard

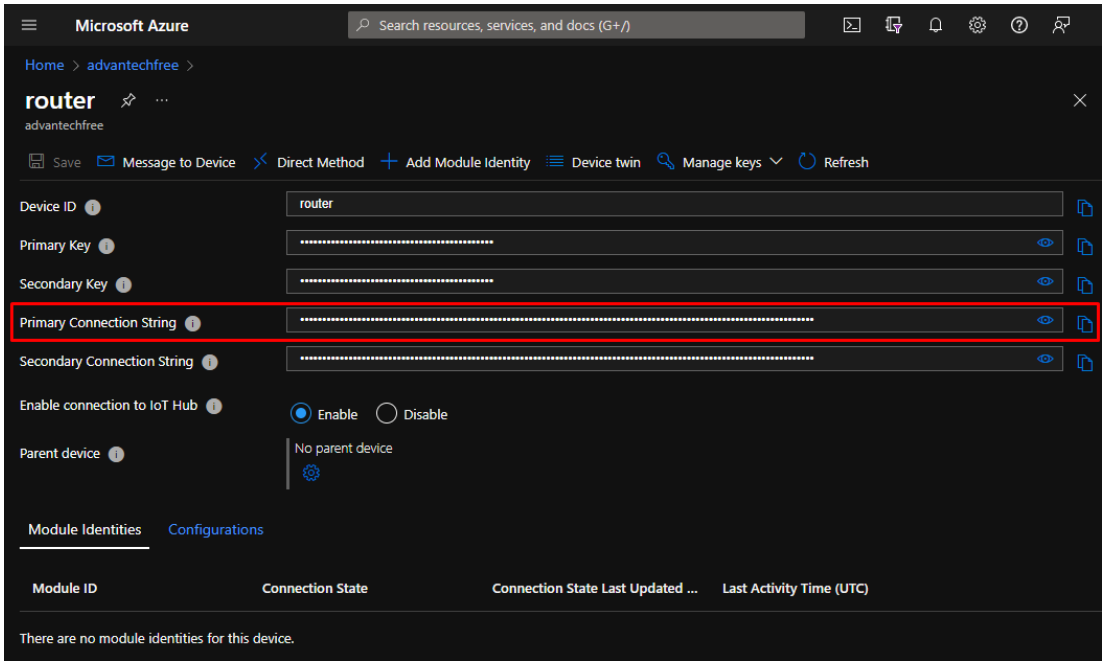


Figure 4: Primary connection string

6. Set a variable to Python enviromnent about the device in Azure to the router CLI:

```
export IOTHUB_DEVICE_CONNECTION_STRING="PASTE_THE_CONNECTION_STRING_HERE"
# export IOTHUB_DEVICE_CONNECTION_STRING="HostName=advantechfree.azure-devices.n
et;DeviceId=router;SharedAccessKey=r42+GvZr8LUnGuCvlgYCBPQ5nq8JJ4Ef4eR9RhtRnPM="
```

7. Start the Azure IoT Python script:

```
# python3 simple_message.py
Sending message...
Message successfully sent!
#
```

Figure 5: Starting the script

8. You can see the information about communication in Azure Shell:

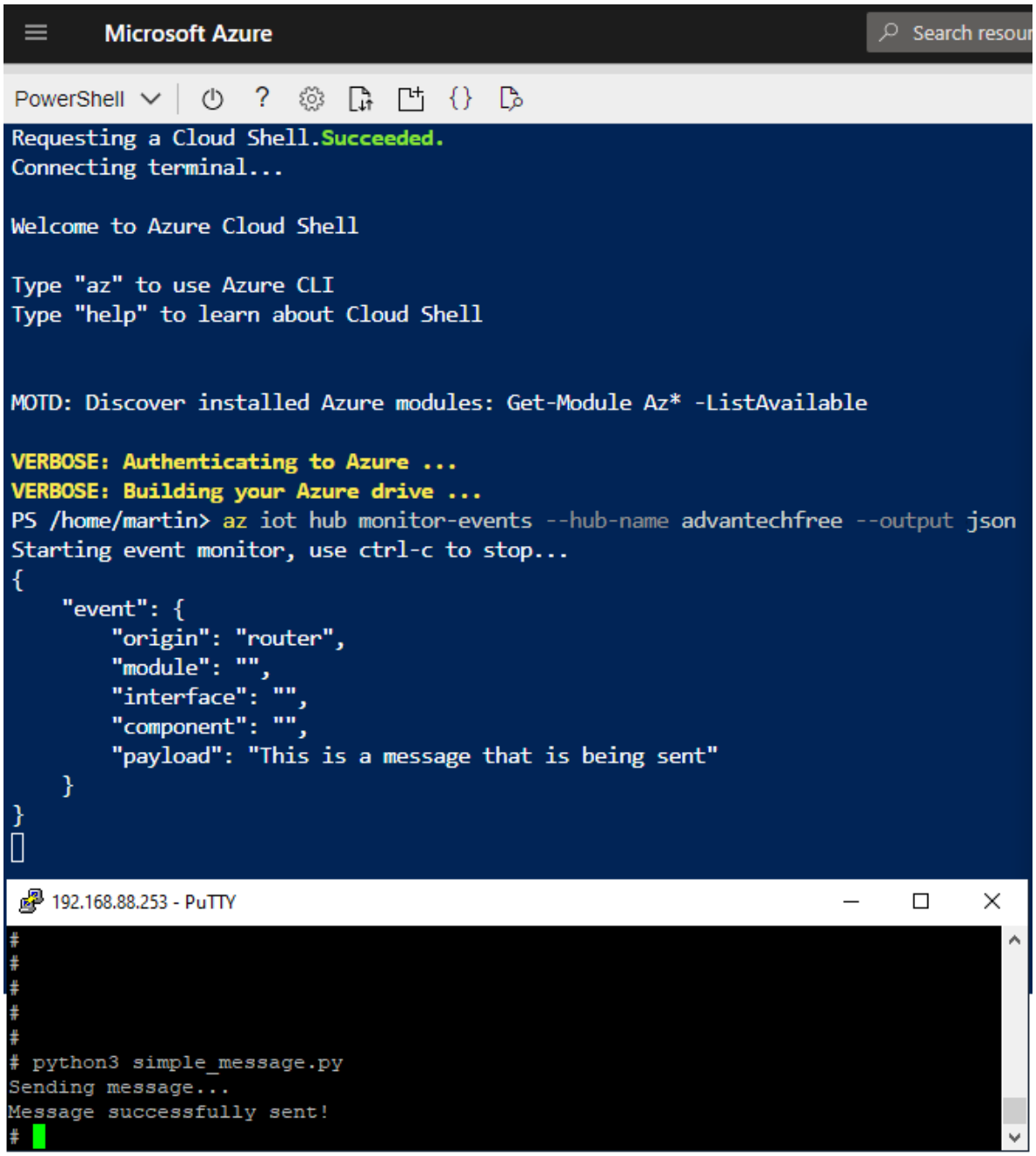


Figure 6: Communication in Azure shell

4. Related Documents

You can obtain product-related documents on *Engineering Portal* at *icr.advantech.cz* address.

To get your router's *Quick Start Guide*, *User Manual*, *Configuration Manual*, or *Firmware* go to the *Router Models* page, find the required model, and switch to the *Manuals* or *Firmware* tab, respectively.

The *Router Apps* installation packages and manuals are available on the *Router Apps* page.

For the *Development Documents*, go to the *DevZone* page.