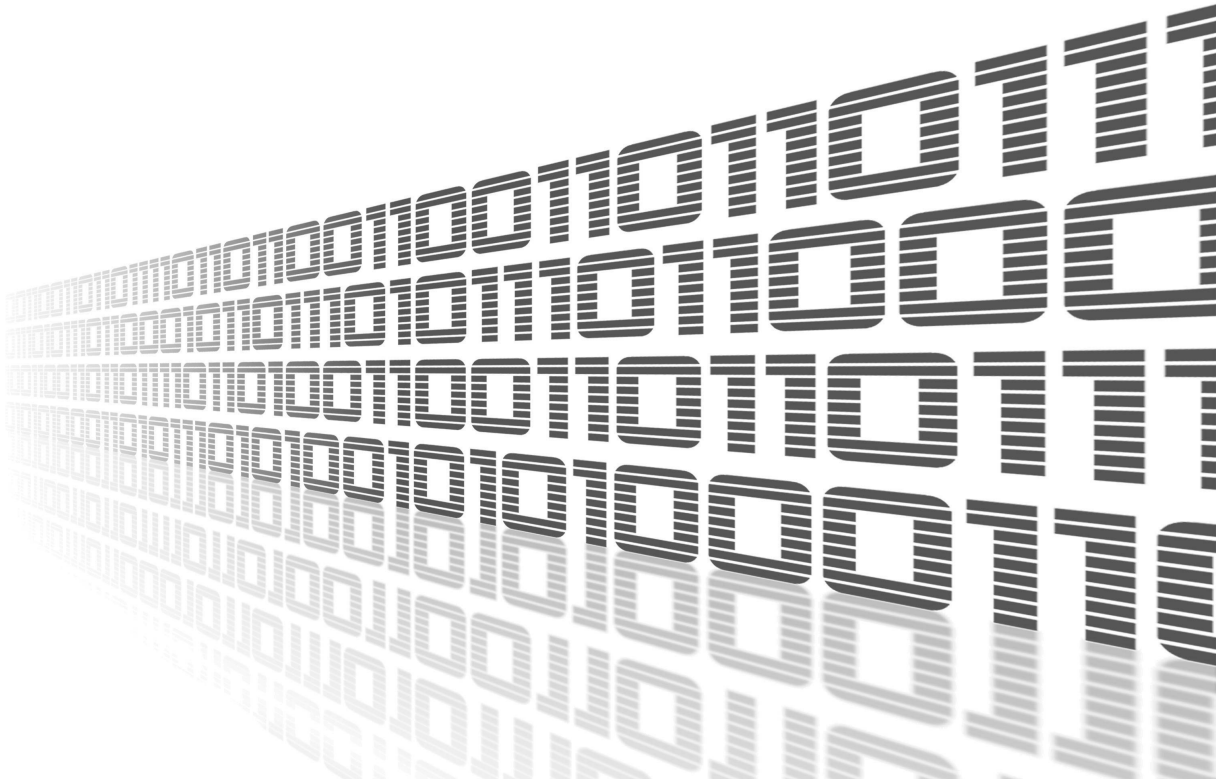
RouterApp

User Module

Azure IoT SDK Python

APPLICATION NOTE



ADVANTECH

Used Symbols

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

Attention – Problems that may arise in specific situations.

Information or notice – Useful tips or information of special interest.

Example – Example of function, command or script.

Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0008-EN, revised on March 7, 2022. Released in the Czech Republic.

Contents

List of Figures

1. User Module Description

This user module is compatible with *Advantech* routers of v3 platform only.

1.1 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.

1.2 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 user module for *Advantech* routers called *Azure IoT SDK Python*.

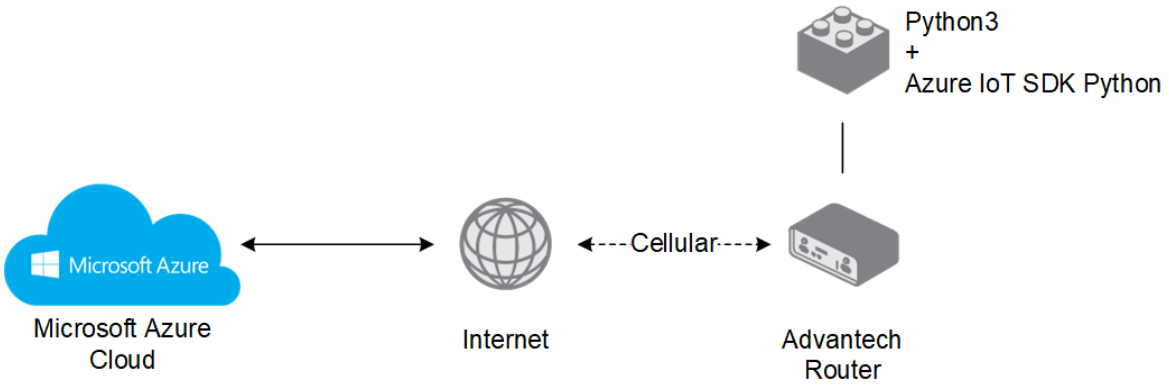


Figure 1: Router with *Python3* and *Azure IoT SDK Python* installed to connect Azure Cloud

Please note that there are two versions of this user module 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* user module 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* user module to be installed in the router – follow the instructions in Chapter 1.3. See the *Configuration Manual*, chapter *Customization –> User Modules*, for the description of how to upload a user module to the router.

1.3 Azure IoT SDK Python Dependency

It is necessary to install the *Python3* user module along with the *Azure IoT SDK Python* user module. *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.

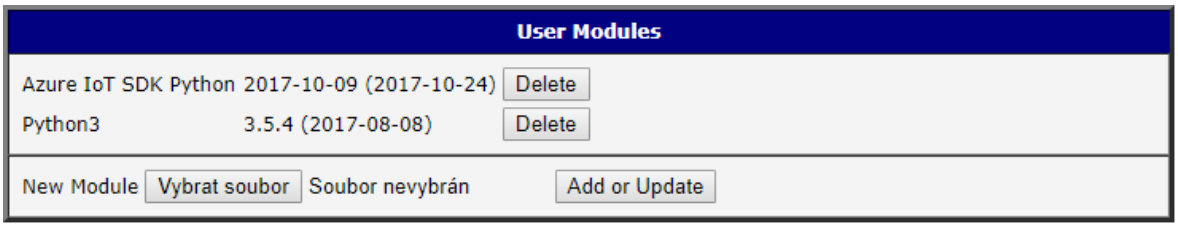


Figure 2: Python3 and Azure IoT SDK Python user modules installed

2. 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:

Figure 3: Example of listed available modules

2.1 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

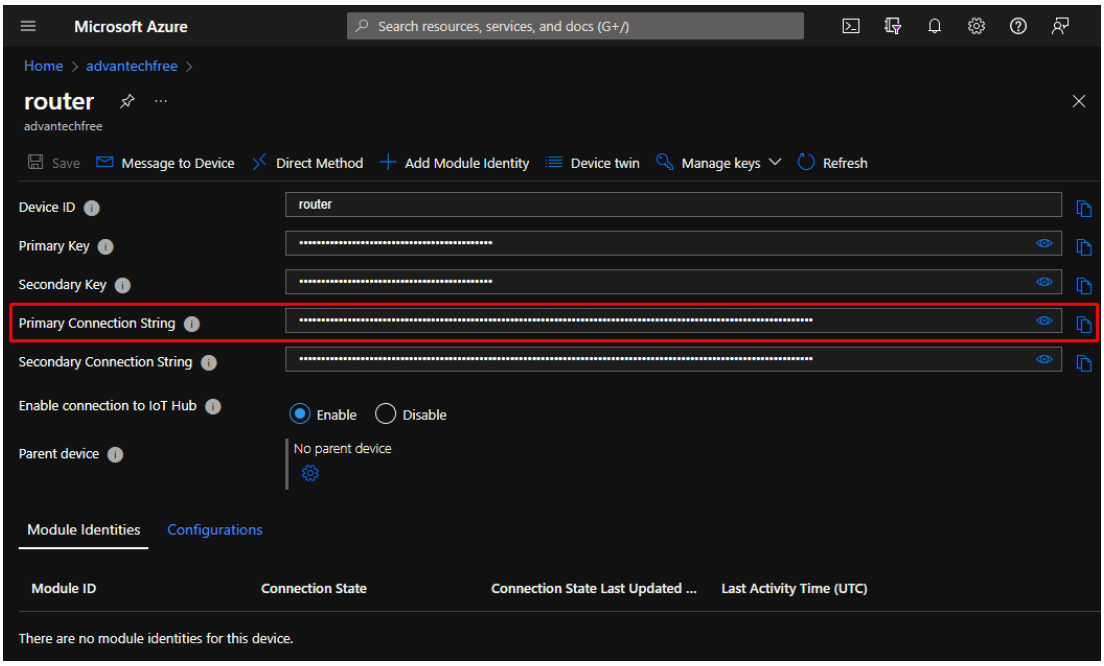


Figure 4: Primary connection string

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:

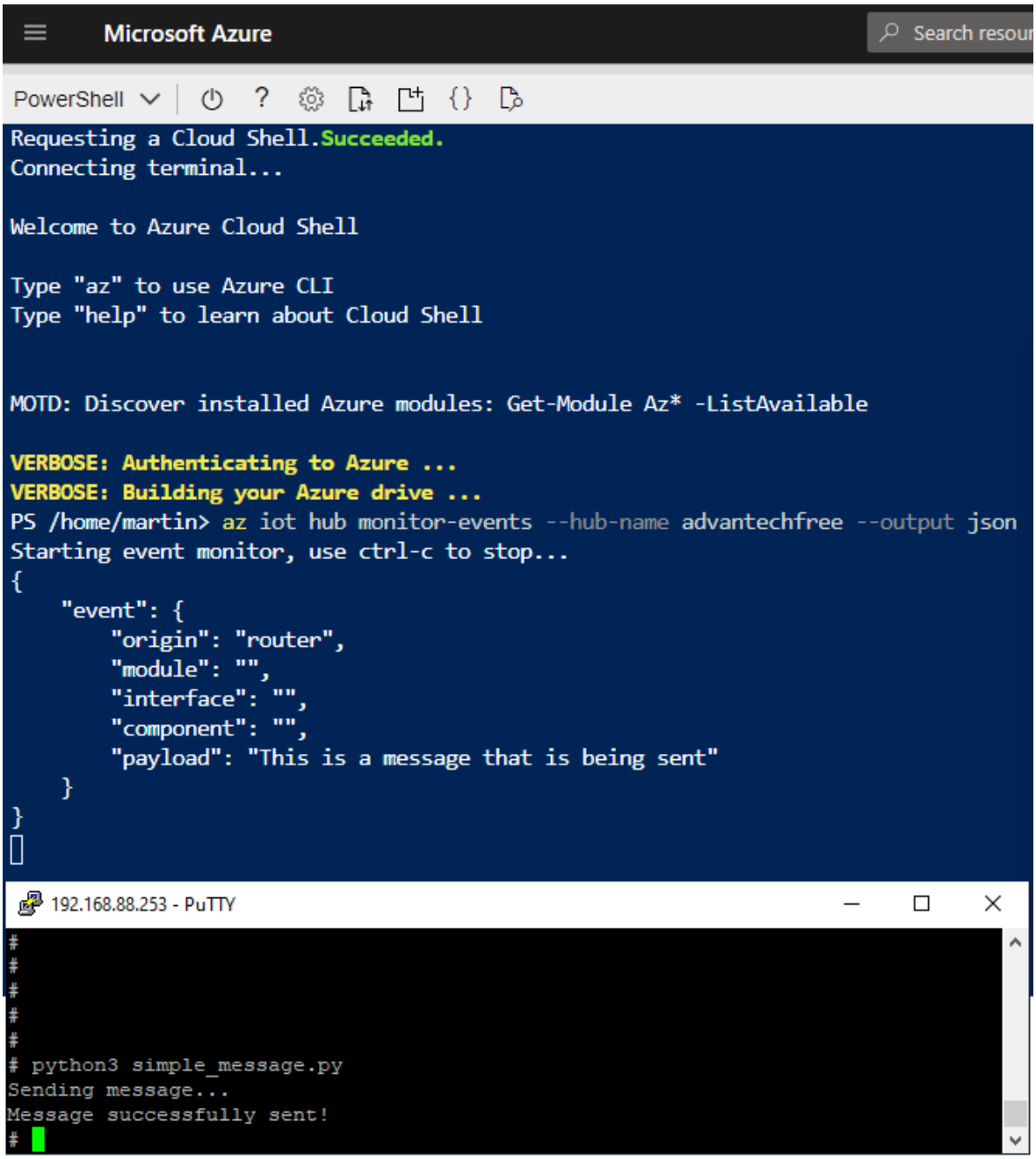


Figure 6: Communication in Azure shell

3. Related Documents

[1]	Advantech Czech:	**SmartFlex Configuration Manual** (MAN-0023-EN)
[2]	Advantech Czech:	**SmartMotion Configuration Manual** (MAN-0024-EN)
[3]	Advantech Czech:	**SmartStart Configuration Manual** (MAN-0022-EN)
[4]	Advantech Czech:	**ICR-3200 Configuration Manual** (MAN-0042-EN)
[5]	User Modules:	**icr.advantech.cz/user-modules**
[6]	Microsoft Azure:	Azure IoT Developer Center **https://azure.microsoft.com/en-us/develop/iot/**
[7]	GitHub:	**Microsoft Azure IoT SDKs for Python**

Product-related documents can be obtained on *Engineering Portal* at icr.advantech.cz address.