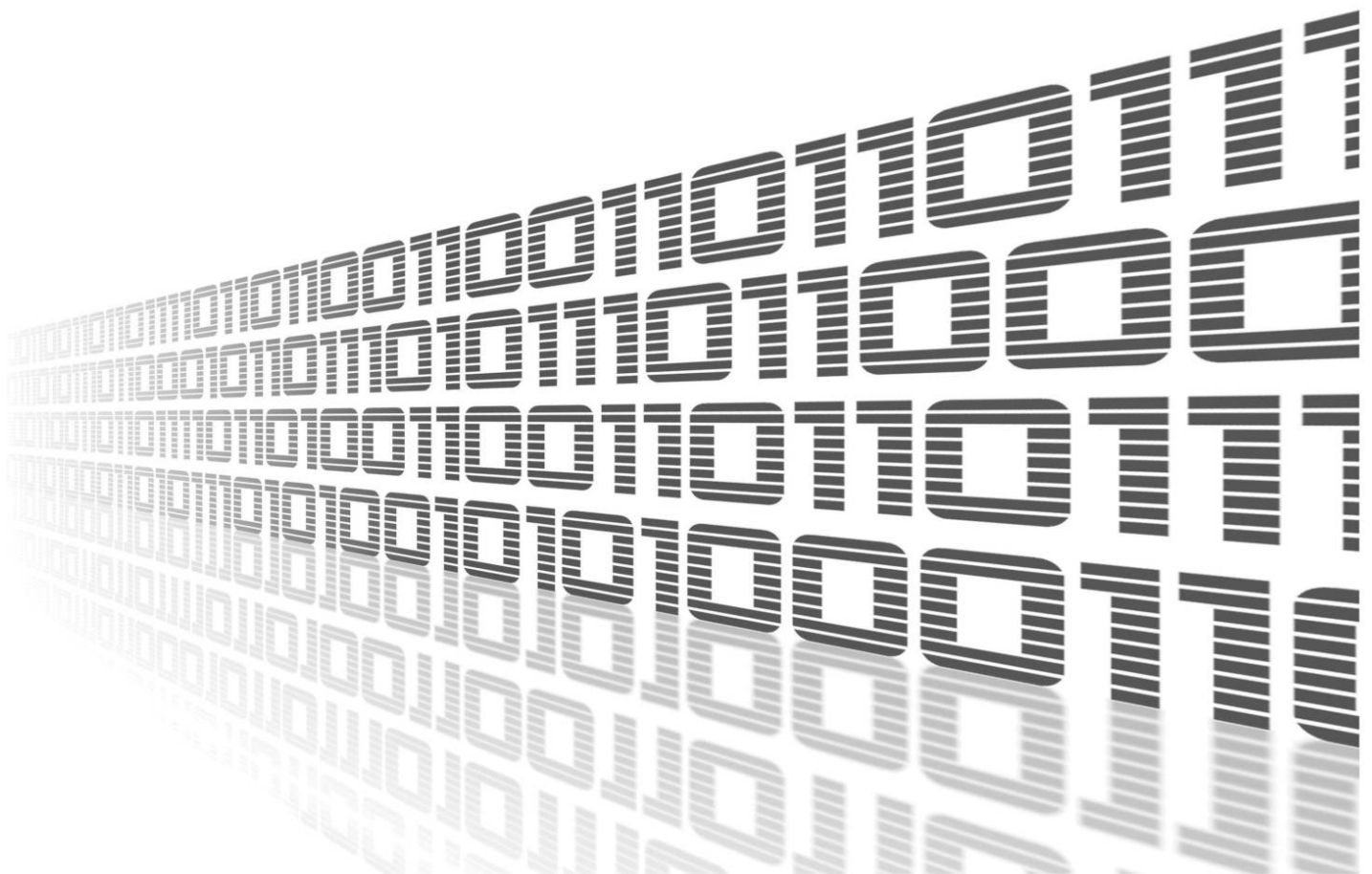


ADVANTECH



Azure IoT SDK Python



© 2023 Advantech Czech s.r.o. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photography, recording, or any information storage and retrieval system without written consent. Information in this manual is subject to change without notice, and it does not represent a commitment on the part of Advantech.

Advantech Czech s.r.o. shall not be liable for incidental or consequential damages resulting from the furnishing, performance, or use of this manual.

All brand names used in this manual are the registered trademarks of their respective owners. The use of trademarks or other designations in this publication is for reference purposes only and does not constitute an endorsement by the trademark holder.

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

1. Changelog	1
1.1 Azure IoT SDK Python Changelog	1
2. Router App Description	2
2.1 Azure IoT	2
2.2 SDK for Python	2
2.3 Azure IoT SDK Python Dependency	3
3. Available Python Modules	4
3.1 Azure installation	5
4. Related Documents	8

List of Figures

1 Router with <i>Python3</i> and <i>Azure IoT SDK Python</i> installed to connect Azure Cloud	2
2 Python3 and Azure IoT SDK Python router apps installed	3
3 Example of listed available modules	5
4 Primary connection string	6
5 Starting the script	6
6 Communication in Azure shell	7

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*.

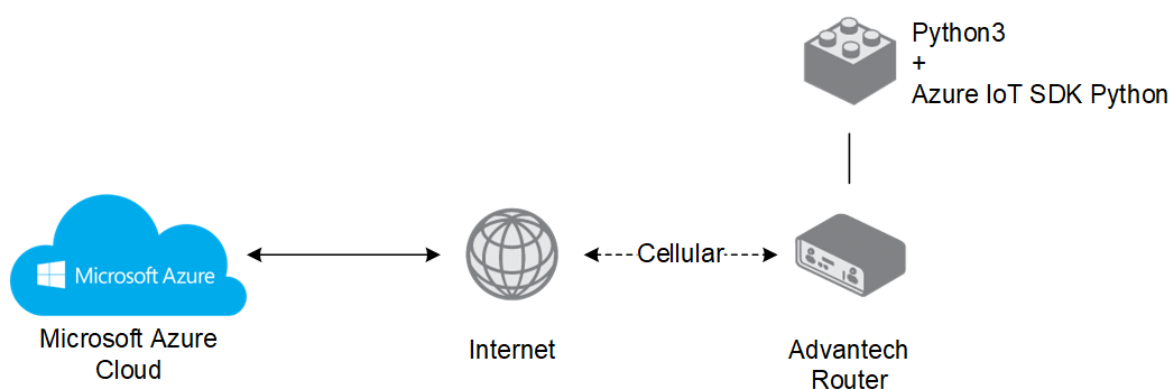


Figure 1: Router with *Python3* and *Azure IoT SDK Python* installed to connect Azure Cloud

i 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.

i 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.

User Modules			
Azure IoT SDK Python	2017-10-09 (2017-10-24)		Delete
Python3	3.5.4 (2017-08-08)		Delete
New Module	<input type="text" value="Vybrat soubor"/>	Soubor nevybrán	Add or Update

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         imgchr       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  cglib        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

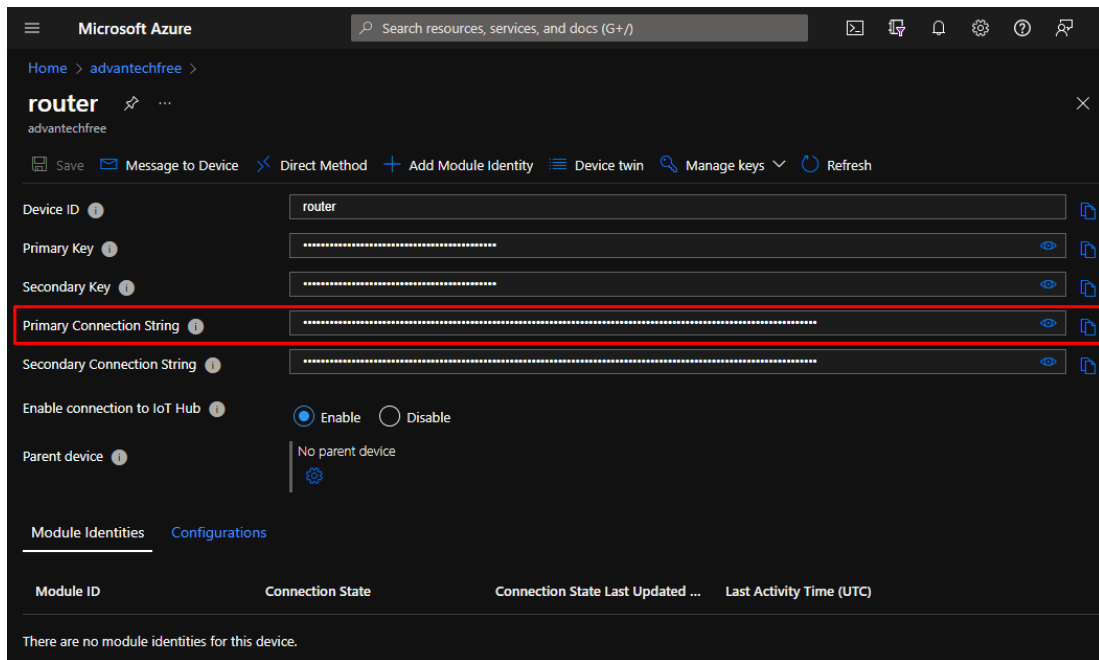


Figure 4: Primary connection string

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

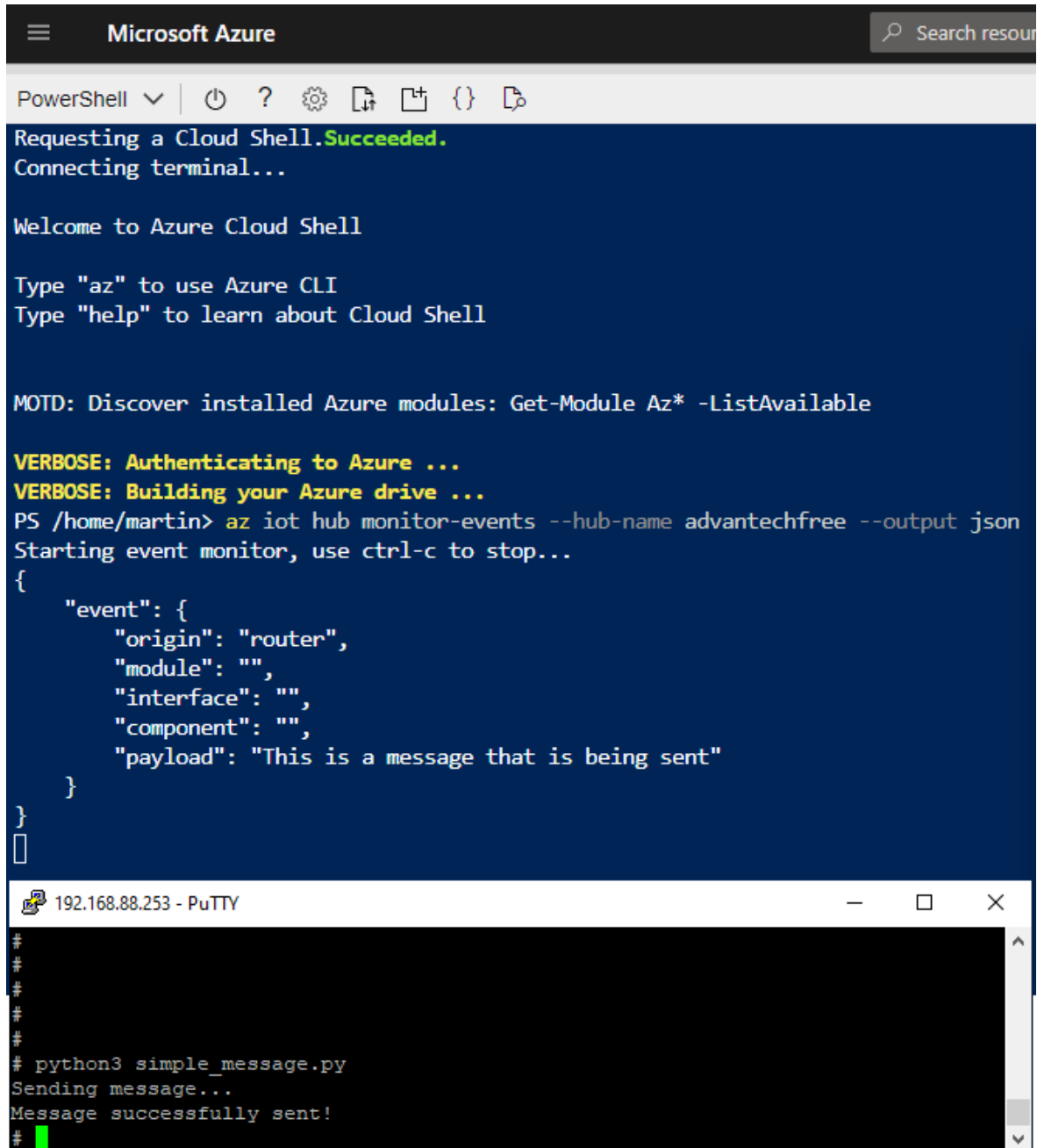
```
export IOTHUB_DEVICE_CONNECTION_STRING="PASTE_THE_CONNECTION_STRING_HERE"
# export IOTHUB_DEVICE_CONNECTION_STRING="HostName=advantechfree.azure-devices.net;DeviceId=router;SharedAccessKey=r42+GvZr8LUnGuCvlgYCBPQ5nq8JJ4Ef4eR9RhtRnPM="
```

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:



The screenshot displays the Microsoft Azure Cloud Shell environment. The top bar shows the 'Microsoft Azure' logo and a search box. Below the top bar, the shell type is set to 'PowerShell'. The main terminal area has a dark blue background and shows the following text:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

MOTD: Discover installed Azure modules: Get-Module Az* -ListAvailable

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/martin> az iot hub monitor-events --hub-name advantechfree --output json
Starting event monitor, use ctrl-c to stop...
{
  "event": {
    "origin": "router",
    "module": "",
    "interface": "",
    "component": "",
    "payload": "This is a message that is being sent"
  }
}
[]
```

At the bottom of the screenshot, a separate window titled '192.168.88.253 - PuTTY' is visible. It shows a terminal session with the following text:

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

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.