Rev. 0 - September 15th, 2023

R7780

Command Line Interface Quick Start Guide

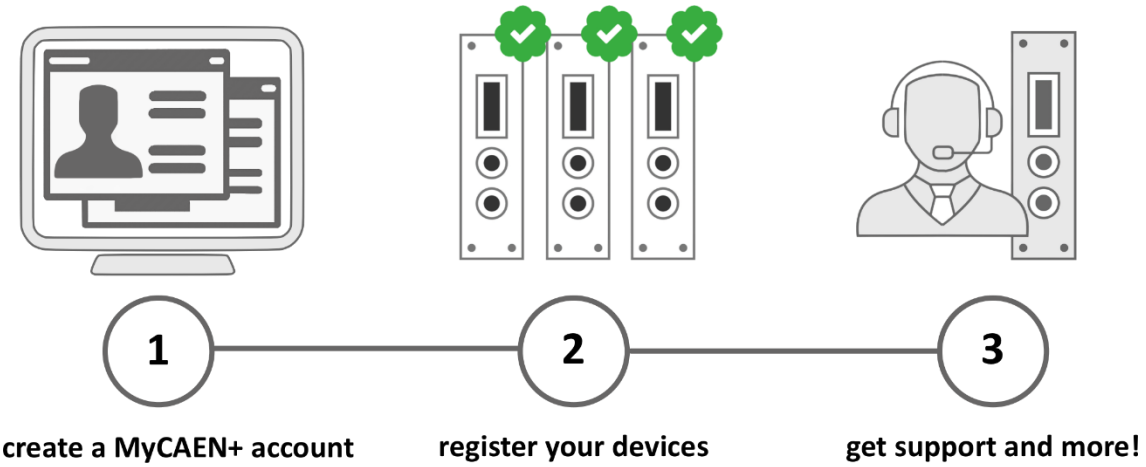
CAEN
Tools for Discovery

Quick Start Guide
UM0000

Register your device

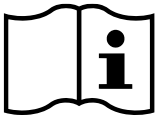
Register your device to your **MyCAEN+** account and get access to our customer services, such as notification for new firmware or software upgrade, tracking service procedures or open a ticket for assistance. **MyCAEN+** accounts have a dedicated support service for their registered products. A set of basic information can be shared with the operator, speeding up the troubleshooting process and improving the efficiency of the support interactions.

MyCAEN+ dashboard is designed to offer you a direct access to all our after sales services. Registration is totally free, to create an account go to https://www.caen.it/become-mycaenplus-user and fill the registration form with your data.



https://www.caen.it/become-mycaenplus-user/

Purpose of this Quick Start Guide



This Quick Start Guide provides an overview of the available methods and shows how to connect to and use the APIs

Change Document Record

Date	Revision	Changes
September15th, 2023	00	Initial release

Symbols, Abbreviated Terms and Notations

API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
GPS	Global Positioning System
URL	Uniform Resource Locator
IP	Internet Protocol
OS	Operating system

Manufacturer Contacts

CAEN S.p.A.
Via Vetraia, 11 55049 Viareggio (LU) - ITALY
Tel. +39.0584.388.398 Fax +39.0584.388.959
www.caen.it | info@caen.it

Limitation of Responsibility

If the warnings contained in this manual are not followed, CAEN will not be responsible for damage caused by improper use of the device. The manufacturer declines all responsibility for damage resulting from failure to comply with the instructions for use of the product. The equipment must be used as described in the user manual, with particular regard to the intended use, using only accessories as specified by the manufacturer. No modification or repair can be performed.

Disclaimer

Table of Contents

List of Figures

1 Introduction

The R7780 is a Neutron Coincidence and Multiplicity Analyzer module combining the functions of a Shift Register and a Pulse Train Recorder. It can be configured in such a way as to provide time-stamped lists of the incoming events and the overall neutron counting information (coincidence timing, multiplicity distributions of coincident events, etc.) required for the analysis in Nuclear Safeguards and nuclear material process monitoring.

After the start-up sequence based on a programmable configuration file, the device can collect data without external control on two removable SD cards.
The designed APIs allow to access the data collected by the R7780 device remotely via HTTP commands, making it easy to integrate R7780 into your existing systems and workflows. Furthermore, the APIs provide advanced functions to monitor the device status, to read the current configuration and to send some key commands.

This guide provides an overview of the available methods and shows how to connect to and use the APIs. Whether building a custom radiation monitoring solution or incorporating R7780 data into existing tools, the APIs offer a flexible and powerful way to access the data needed.

2 Methods

The R7780 device API methods are based on the HTTP protocol and include standard HTTP verbs such as GET, PUT, POST, DELETE, and OPTIONS. These methods are used to retrieve, update, create, and delete resources on the device, making it easy to interact with the data collected by the device.

The API returns data in JSON format, and results file can be downloaded locally on a computer once a connection has been established with the device, allowing for easy integration into your applications and systems. In this chapter, we will explore each of the available methods.

2.1 Description

The methods implemented have a specific function in the REST API paradigm:

- **GET**: can be used to retrieve real-time data and information from the device in JSON format
- **PUT**: can be used to update some device settings
- **POST**: creates a new resource on the R7780 server, i.e. uploading a new firmware.
- **DELETE**: removes an entity on the R7780 server, i.e. an old log file
- **OPTIONS**: returns specific information on the endpoint, usually used by browsers for checking CORS

3 Quick Start Guide

This chapter provides examples of how to use the R7780 device API methods to interact with the device's resources and retrieve data.

These examples illustrate the various capabilities of the API and can be helpful for getting started using the API in applications and systems. Each example includes a Bash or PowerShell script and demonstrates how to call the API from the command line via a Linux or Windows web client.

Whether building a custom monitoring solution or incorporating R7780 data into existing tools, these examples provide a useful starting point for working with the API.

3.1 Windows

Here's is shown how to connect to a R7780 with an IP address of 10.105.254.33.

Windows provides an integrated Web Client called Invoke-WebRequest (the documentation is available at https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/invoke-webrequest?view=powershell-7.3).

Here are some steps to pass credentials to the Windows HTTP client:

- Set the URL name which is composed by the host-name and the API path:
  $address="http://10.105.254.33/"
  $api='api/files/filesystem/sdcard1/data/2023-07'
  $url_folder=$address + $api
- Set Username and password variables:
  $user='admin'
  $password='admin'
  $pair="${user}:${password}"
- Encode variable pair in base 64:
  $encoded = [System.Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($pair))
- Create an Http Basic Authentication Header:
  $basicAuthHeader = "Basic $encoded"
  $ Headers = @{
  Authorization = $basicAuthHeader
  }
- Finally invoke the web client with the authentication header:
  For instance, if you want to get the list of the available files inside a folder on SD card 1 you must issue the following command
  Invoke-WebRequest -Uri "http://r7780-ip-address/api/files/filesystem/sdcard1/folder-name" -Headers $Headers
- In the case of PUT/POST data the Invoke-WebRequest requires the *-Method METHOD-NAME* (default method is GET) and the *-BODY $params* variable, which contains the data:
  Invoke-WebRequest -Uri "http://r7780-ip-address/api/configuration/time" -Headers $Headers -Method PUT -Body $config

Here is an example of the implementation of a PowerShell script that reads the list of the available files inside a folder on the SD card and saves it in the local directory as JSON file (*/folder_list_R7780.json)*. The script file is called ***example.ps1***.

```
1   #paths & urls
2   $address="http://10.105.254.33/"
3   $api="api/files/filesystem/sdcard1/data/2023-07"
4   $url_folder=$address+$api
5   $output=$PSScriptRoot
6   $path=$output + "\folder_list_R7780.json"
7
8   #credentials
9   $user='admin'
10  $password='admin'
11  $pair=$user+":"+$password
12
13  $encoded=[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($pair))
14  $basicAuthHeader = "Basic $encoded"
15  $headers = @{
16      Authorization = $basicAuthHeader
17  }
18
19  #call
20  $result = Invoke-WebRequest -Uri $url_folder -Headers $headers
21
22  #save data to file
23  $json = ($result | ConvertFrom-Json | ConvertTo-Json)
24  $json *> $path
```

The first six lines are needed to set the R7780 URL, the output folder and output file name. The *$api* variable is the API path. *$url_folder* is the composition of both needed as argument of the Invoke-WebRequest.
The steps to pass credentials to the Windows HTTP client, previously described, are all implemented between line 6 and 18, while lines 23-24 print the JSON results coming from the Invoke-WebRequest in the output file.

The script can be launched by command line as shown in **Figure 3.1**

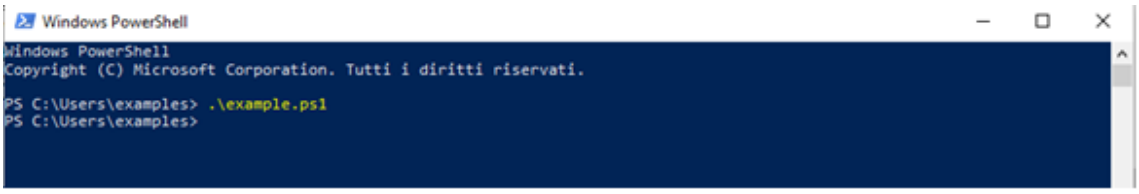


Figure 3.1: How to launch the *example.ps1* PowerShell script

The output file (folder_list_R7780.*json*) can be found in the output folder as a JSON file. The content is shown in **Figure 3.2.**

```
{
    "uri":  "/sdcard1/data/2023-07",
    "is_dir":  true,
    "contents":  [
                    {
                        "uri":  "R7780_03066_2023-07-01_000005.dataz",
                        "is_dir":  false,
                        "lmt":  "2023-07-01T00:14:34.000Z",
                        "size":  200065
                    },
                    {
                        "uri":  "R7780_03067_2023-07-01_001738.dataz",
                        "is_dir":  false,
                        "lmt":  "2023-07-01T00:34:18.000Z",
                        "size":  225468
                    },
                    {
                        "uri":  "R7780_03068_2023-07-01_003721.dataz",
                        "is_dir":  false,
                        "lmt":  "2023-07-01T00:54:02.000Z",
                        "size":  225473
                    },
                    {
                        "uri":  "R7780_03069_2023-07-01_005705.dataz",
                        "is_dir":  false,
                        "lmt":  "2023-07-01T01:13:44.000Z",
                        "size":  225471
                    },
                    {
                        "uri":  "R7780_03070_2023-07-01_011648.dataz",
                        "is_dir":  false,
                        "lmt":  "2023-07-01T01:33:28.000Z",
                        "size":  225471
                    },
                ]
}
```


Figure 3.2: Content of the output file of the *report/{ID}* API.

Once the list of the available files is known, the following example (**example2.ps1**) shows how to download one of the .dataz files. The file is saved as test.dataz inside the current folder.

```
#paths & urls
$address="http://10.105.254.33/"
$api="api/files/filesystem/sdcard1/data/2023-07/R7780 03066 2023-07-01 000005.dataz"
$url_file=$address+$api
$output=$PSScriptRoot
$path=$output + "\test.dataz"

#credentials
$user='admin'
$password='admin'
$pair=$user+":"+$password

$encoded=[Convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes($pair))
$basicAuthHeader = "Basic $encoded"
$headers = @{
    Authorization = $basicAuthHeader
}
#call and save the file
Invoke-WebRequest -Uri $url_file -Headers $headers -OutFile $path
```

3.2 Linux

There are severals Web Clients that can be used on Linux OS: *wget* and *curl* are the most common choices.

This example shows how to connect to a R7780 using the *curl* Web-Client (documentation available at https://everything.curl.dev/).

```
#!/bin/sh
curl -u admin:admin http://10.105.254.33/api/status  -o status.json
```

The script reads the current status of the R7780 and saves it to a .json file:

Login credentials are passed as argument of the *-u* option.

- The hostname is passed right after
- The output file name and path are passed as argument of the *-o* option (default path is the current folder)
- Default method of *curl* is GET

The script can be launched by command line by digiting *sh example.sh*. The terminal output is shown in **Figure 3.3**.

```
lara@lara-Ub:~/Desktop$ sh example.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   194  100   194    0     0  12933      0 --:--:-- --:--:-- --:--:-- 12933
```

Figure 3.3: Terminal output of the *example.sh* script

The output files are saved in the current operating folder, the file content is shown in **Figure 3.4**.


```
{
    "timestamp":  1694791943803,
    "acqStatus":  1,
    "genericError":  0,
    "fifoFullError":  0,
    "vmon":  198065,
    "imon":  15003,
    "hvStatus":  12289,
    "v5mon":  5134,
    "v12mon":  12163,
    "lvStatus":  5,
    "temperatures":  [
                        31,
                        32,
                        29,
                        48.597
                    ]
}
```


Figure 3.4: Json file containing the status info of the R7780

In the following, a list of the available APIs can be found. The methods previously listed in section 2 can be used according to the API functionality.

/api/files returns content of root folder

/api/files/filesystem/path/to/some/file or folder downloads selected file or shows selected folder content

/api/files/download?file=xxx&folder=yyyy downloads selected file

/api/files/filesystem/sdcardX/?afterDate=2020-09-01&beforeDate=2020-10-08 downloads files in date range

/api/files/sdcard/:id/format formats selected sd card

/api/files/sdcard/restart restarts sd cards

/api/files/sdcard/unmount unmounts sd cards

/api/health to check if api are up and running

/api/status returns the status

/api/storage returns storage information

/auth/user return active user

/auth/token POST generates a token

/api/user returns user information

/api/user/:name returns user called name's information

/api/user/:name/role PUT updates selected user roles

/api/user/:name/password PUT updates selected user password

/api/system/remark to create a remark

/api/system/reboot to reboot

/api/system/daq/start starts acquisition

/api/system/daq/stop stops acquisition

/api/configuration/firmware POST receives a new firmware

/api/configuration/power/5v turns 5v on/off

/api/configuration/power/12v turns 12v on/off

/api/configuration/reset POST resets configuration

/api/configuration/reset PATCH updates live configuration

/api/configuration/live returns live configuration

/api/configuration/time returns current time utc

/api/configuration/time/local returns current time in local timezone

/api/configuration/time/rtc returns current time stored in rtc

/api/configuration/time PUT sets current time

4 Technical Support

To contact CAEN specialists for requests on the software, hardware, and board return and repair, it is necessary a MyCAEN+ account on www.caen.it:

https://www.caen.it/support-services/getting-started-with-mycaen-portal/

All the instructions for use the Support platform are in the document:

A paper copy of the document is delivered with CAEN boards.

The document is downloadable for free in PDF digital format at:

www.caen.it/safety-information-product-support

R7780
Command Line Interface Quick Start Guide

CAEN GmbH
Eckehardweg 10
42653 - Solingen
Germany
Phone +49 212 254 40 77
Fax +49 212 254 40 79
info@caen-de.com
www.caen-de.com

CAEN Technologies, Inc.
1 Edgewater Street - Suite 101
Staten Island, NY 10305
USA
Phone: +1 (718) 981-0401
Fax: +1 (718) 556-9185
info@caentechnologies.com
www.caentechnologies.com

CAEN S.p.A.
Via Vetraia 11
55049 - Viareggio
Italy
Phone +39 0584 388 398
Fax +39 0584 388 959
info@caen.it
www.caen.it

CAENspa INDIA Private Limited
B205, BLDG42, B Wing,
Azad Nagar Sangam CHS,
Mhada Layout, Azad Nagar, Andheri (W)
Mumbai, Mumbai City,
Maharashtra, India, 400053
info@caen-india.in
www.caen-india.in

UM9781 - R7780 rev. 0 - September 15th, 2023 00000-00-R7780-MUTX

CAEN Tools for Discovery

Electronic Instrumentation