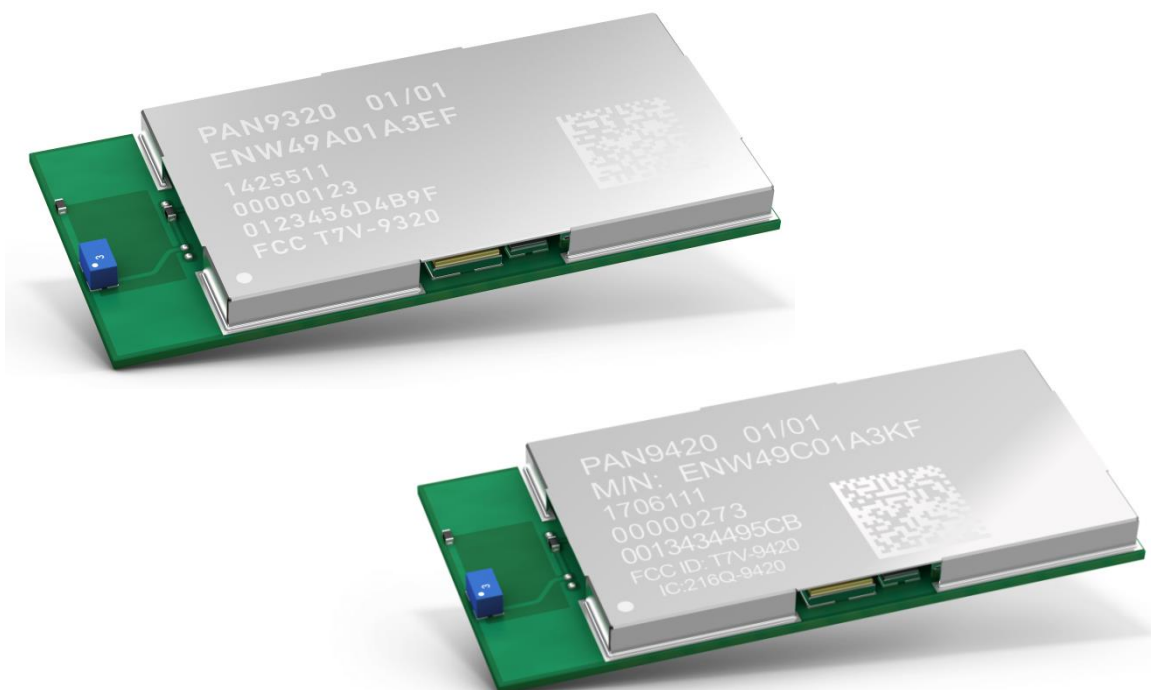


PAN9320/PAN9420

Software Update Over The Air

Application Note

Rev. 1.0



By purchase of any of the products described in this document the customer accepts the document's validity and declares their agreement and understanding of its contents and recommendations. Panasonic Industrial Devices Europe GmbH (Panasonic) reserves the right to make changes as required at any time without notification.

© Panasonic Industrial Devices Europe GmbH 2017.

This document is copyrighted. Reproduction of this document is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Do not disclose it to a third party.

All rights reserved.

This Application Note does not lodge the claim to be complete and free of mistakes.

The information contained herein is presented only as guidance for Product use. No responsibility is assumed by Panasonic for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

Description of hardware, software, and other information in this document is only intended to illustrate the functionality of the referred Panasonic product. It should not be construed as guaranteeing specific functionality of the product as described or suitable for a particular application.

Any provided (source) code shall not be used or incorporated into any products or systems whose manufacture, use or sale is prohibited under any applicable laws or regulations.

Any outlined or referenced (source) code within this document is provided on an "as is" basis without any right to technical support or updates and without warranty of any kind on a free of charge basis according to § 516 German Civil Law (BGB) including without limitation, any warranties or conditions of title, non-infringement, merchantability, or fitness for a particular purpose. Customer acknowledges that (source) code may bear defects and errors.

The third-party tools mentioned in this document are offered by independent third-party providers who are solely responsible for these products. Panasonic has no responsibility whatsoever for the performance, product descriptions, specifications, referenced content, or any and all claims or representations of these third-party providers. Panasonic makes no warranty whatsoever, neither express nor implied, with respect to the goods, the referenced contents, or any and all claims or representations of the third-party providers.

To the maximum extent allowable by Law Panasonic assumes no liability whatsoever including without limitation, indirect, consequential, special, or incidental damages or loss, including without limitation loss of profits, loss of opportunities, business interruption, and loss of data.

Table of Contents

1	About This Document.....	4
1.1	Purpose and Audience	4
1.2	Revision History.....	4
1.3	Use of Symbols	4
1.4	Related Documents	5
2	Introduction	6
3	Prerequisites	7
3.1	Web Server.....	7
3.2	Certificates.....	7
3.3	Terminal Software	7
4	Module Basics.....	8
4.1	Memory Layout.....	8
4.2	File System.....	9
4.3	Configuration Layout.....	10
5	HTTP(S) Communication Setup.....	12
6	Creating Certificates.....	13
6.1	Certification Authority.....	13
6.2	Server Certificate.....	13
6.3	Client Certificate	14
7	Enabling HTTP(S) Updates	15
8	Creating a Firmware Update File	17
9	XAMPP Apache Web Server	18
10	Performing a Firmware Update.....	19
11	Appendix	21
11.1	Contact Details	21

1 About This Document

1.1 Purpose and Audience

The purpose of this document is to explain how to set up and perform software updates over the air using the PAN9320/PAN9420 Wi-Fi module and a web server.

In detail, the document describes how to

- set up a local web server for development and testing,
- create test certificates for the server and the client using a test certification authority,
- set up the PAN9320/PAN9420 to use the feature and
- perform the actual software update.




The PAN9320/PAN9420 Wi-Fi module is referred to as “the PAN9x20” or “the module” within this document.

The document addresses software engineers who have the intention to use the implemented software update process of the module in their application.

1.2 Revision History

Revision	Date	Modifications/Remarks
0.1	13.06.2017	Initial draft
0.2	10.07.2017	Updated document title, disclaimer, header, footer and the document version of related documents.
1.0	31.07.2017	Added chapter “Module Basics” and configuration file description in chapter “Enabling HTTP(S) Updates”, editorial review

1.3 Use of Symbols

Symbol	Description
	Note Indicates important information for the proper use of the product. Non-observance can lead to errors.
	Attention Indicates important notes that, if not observed, can put the product's functionality at risk.
	Tip Indicates useful information designed to facilitate working with the software.
⇒ [chapter number] [chapter title]	Cross reference Indicates cross references within the document. Example: Description of the symbols used in this document ⇒ 1.3 Use of Symbols .

Symbol	Description
✓	Requirement Indicates a requirement that must be met before the corresponding tasks can be completed.
→	Result Indicates the result of a task or the result of a series of tasks.
This font	GUI text Indicates fixed terms and text of the graphical user interface. Example: Click Save .
Menu > Menu item	Path Indicates a path, e.g. to access a dialog. Example: In the menu, select File > Setup page .
This font	File names, messages, user input Indicates file names or messages and information displayed on the screen or to be selected or entered by the user. Examples: <code>main.c</code> contains the actual module initialization. The message <code>Failed to save your data</code> is displayed. Enter the value <code>Product 123</code> .
[Key]	Key Indicates a key on the keyboard, e.g. [F10].

1.4 Related Documents

[1] Panasonic. PAN9320 Communication Specification Firmware V1.9.6.4

[2] Panasonic. PAN9420 Communication Specification Firmware V1.1.0.6

Please refer to the Panasonic website for more related documents

⇒ [11.1.2 Product Information](#).

2 Introduction

The PAN9x20 enables to perform software updates over the air using a web server.

In an application this feature enables a manufacturer to use a dedicated update server on the internet to distribute bugfixes, feature enhancements or firmware updates for its products.

The updates can address various targets such as the PAN9x20

- firmware,
- flash images (certificates, websites, etc.) and
- configuration as well as
- host controller firmware.

As there are different types of web servers with different setup procedures and configurations, it is beyond the scope of this document to explain them all in detail. Thus, this document simply describes an exemplary secure firmware update using the local Apache web server from within XAMPP. The setup and configuration can then be transferred to other web servers accordingly.

3 Prerequisites

This chapter describes which prerequisites have to be met to successfully work through the remaining document.

3.1 Web Server

To perform HTTP(S) client requests from the PAN9x20 to a web server it is inevitable to have some sort of web server. This server can either be online or just an offline local test server. In this document an Apache web server from within XAMPP is used.

3.2 Certificates

If a secure HTTP(S) communication is desired, certificates are necessary. Hence, to create keys, certificate signing requests and certificates a TLS implementation is required. In this document OpenSSL is used to take over this task.

3.3 Terminal Software

When using a PAN9x20 ETU in combination with a motherboard, the module can be controlled from a computer using a terminal, such as

- HTerm
- Tera Term or
- HyperTerminal.

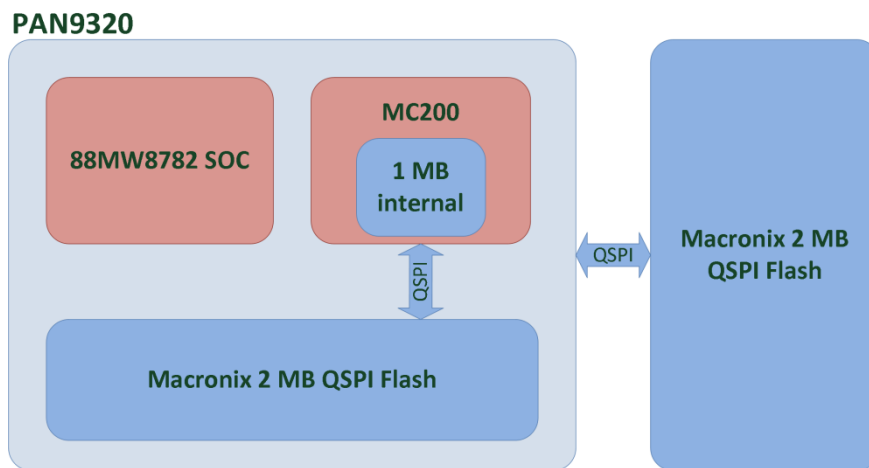
4 Module Basics

This chapter describes important PAN9x20 basics, which are essential to understand in order to successfully create update files.

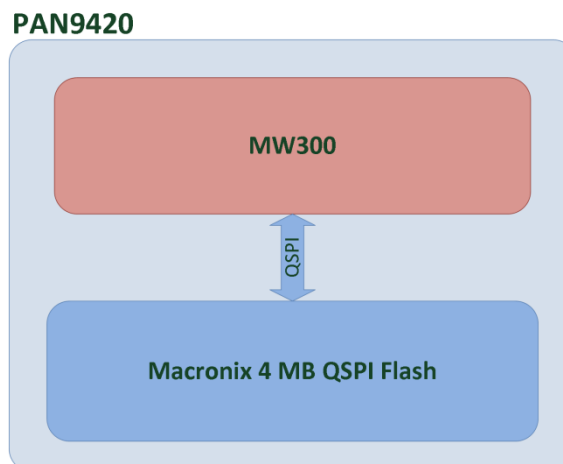
4.1 Memory Layout

The PAN9320 has 3 MB of internal flash memory, of which 1 MB is available for user content such as configuration files, webpages, certificates, etc. and 2 MB are reserved for module firmware and caching firmware uploads.

If more than 1 MB of flash memory is required for user content, additional 2 MB of external flash can be added to the application using the PAN9320 QSPI interface.



The PAN9420 has 4 MB of internal flash memory, of which 2 MB are available for user content and the remaining 2 MB are reserved for internal usage.



4.2 File System

The PAN9x20 utilizes a file system in order to systematically store and find resources such as webpage files, certificates and others. The file system is used by the HTTP webserver as well as the Transport Layer Security (TLS) of various software modules. The webserver uses it to search for webpage files, whereas the TLS uses it to search for valid certificates.

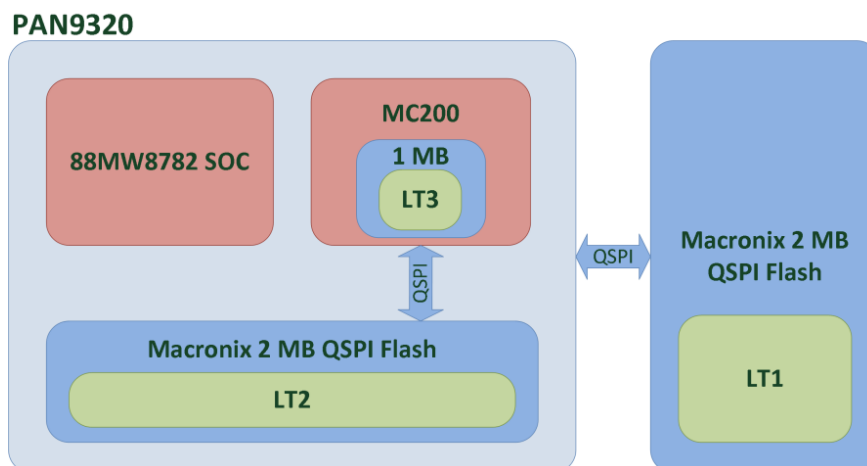
The user-editable folders in the file system are listed and described below:

Path	Description
/web	Websites and resources which are accessible via webserver. Note: The filename including the path must not exceed 80 characters.
/cert	Location for trusted webserver certificates. The certificates are used from the mail and HTTPC module to establish TLS connections. Note: The file name of the certificate must be the subject key identifier of the certificate and it must have the .der format (e.g. 949c41491e583f48f20206dc743693c6f322ec9a.der)
/cert/own	Location for the PAN9x20 webserver certificates. Note: The certificate name must be <code>pan9320.der</code> respectively <code>pan9420.der</code> and the private key name must be <code>pan9320.key</code> respectively <code>pan9420.key</code> .
/cert/fwu	Location for trusted firmware update servers. Note: The file name of the certificate must be the subject key identifier of the certificate and it must have the .der format (e.g. 949c41491e583f48f20206dc743693c6f322ec9a.der)

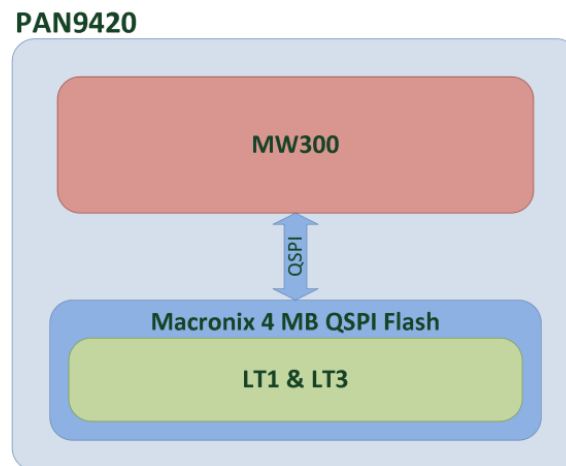
Assuming that external flash is connected, the file system will search for resources in all three file partitions, with the following prioritization:

1. The first search starts in LT 1 (look-up table).
2. If the data is not in LT1, search in LT2.
3. If the data is not in LT1 and LT2, load LT3.

To show the different locations of look-up tables in the PAN9320 memory, the initial memory layout can be extended as follows:



The memory layout of the PAN9420 can be extended as follows:



On the PAN9420 there is no LT1 due to missing external memory. In this case a resource search starts in LT2.

Beside the mentioned searching mechanism, it is possible to address PAN9x20 resources directly using a URL with a prefix.

Prefix	Flash Memory PAN9320	Flash Memory PAN9420
LT1/	External	Internal (PAN9420 module)
LT2/	Internal (PAN9320 module)	
LT3/	Internal (PAN9320 MC200 MCU)	Internal (PAN9420 module)

Example:

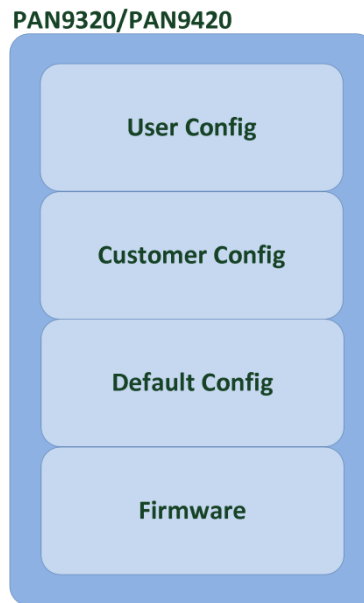
To directly access the PAN9x20 fallback webpage, enter the following URL into the browser:

<http://192.168.1.1/LT3/index.html>

4.3 Configuration Layout

The configuration of the PAN9x20 consists of three configuration options.

- Default Config: Contains basic configuration parameters.
- Customer Config: Configuration parameters set by the customer.
- User Config: Configuration parameters set by the user at runtime.



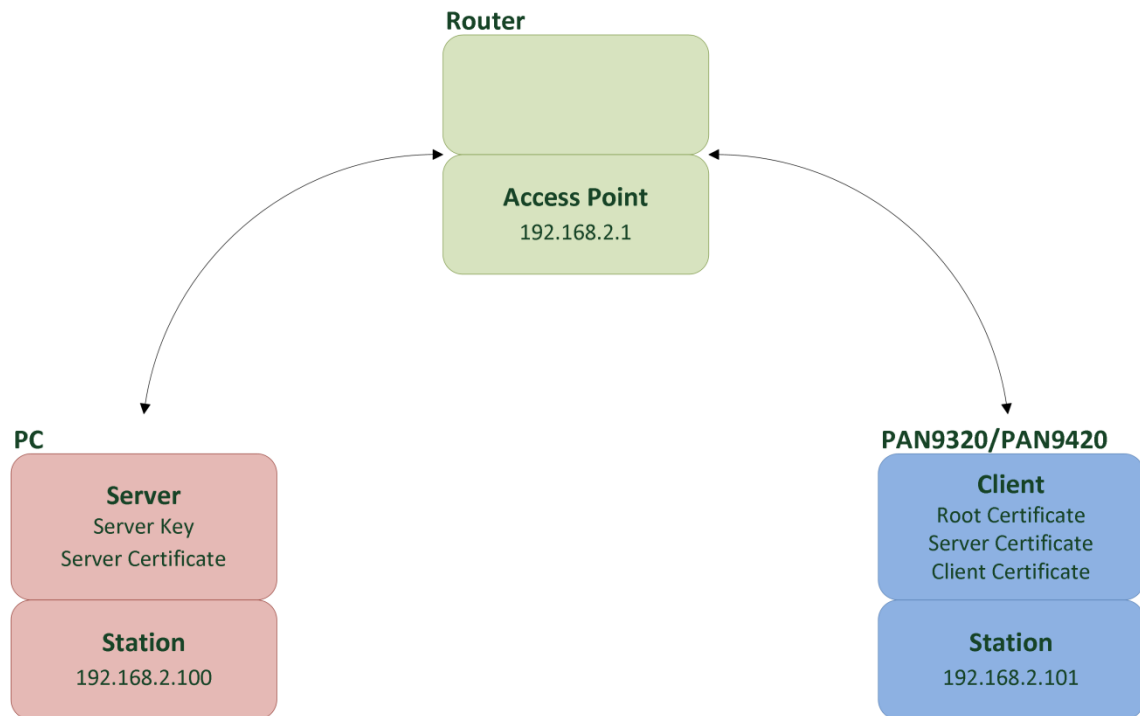
The basis of the PAN9x20 configuration layout is the module firmware with the bootloader and radio driver. The behavior of this firmware can be configured over three configuration layers of which only two can be used by the customer. The default configuration is done by Panasonic in order not to endanger the certification of the module and the end application.

The customer configuration enables to customize most of the PAN9x20 default settings of access point, station, UART interfaces, GPIOs, HTTP client, firmware update server and others.

The user configuration is the configuration which is done by host and user at runtime and can address most of the previously mentioned settings depending on the user rights.

5 HTTP(S) Communication Setup

This chapter describes and depicts the local test setup used for the update process in this document. The IP addresses shown in the figure below should help to understand the configuration of the setup process described in the following chapters. In a proper application this setup would obviously differ, while the server would be online and there would be multiple clients sending and requesting information.



As depicted above, a computer is running a server application, which again is hosting the resources for the firmware update. This computer is connected to a network in order to serve its services to other computers and clients. From the security point of view, the server uses its server key as well as its server certificate to provide communication security using TLS.

On the client side, the PAN9x20 is running a client application requesting update information and files from the server. This node is connected to the network the server is on.

The full setup can be operated in a secure as well as in an insecure mode. This document will only show the secure mode as the insecure mode is much easier and self-evident.

6 Creating Certificates

In the following sections, OpenSSL is used to create a test Certification Authority (CA) as well as generate test certificates for the server and the client.

The process of creating a certificate is similar for all the three different instances. First, a private key is generated. Then, a certificate signing request (CSR) is generated, which contains the public key, identifying information and integrity protection. Finally, the certificate itself will be generated.

The configuration and extension files used during the process are attached to this document.

6.1 Certification Authority

Create a private key:

```
openssl genrsa -out ca.key 1024
```

Create the CSR using a config file:

```
openssl req -config ca.config -new -key ca.key -out ca.csr
```

Create the x509 certificate using an extension file (x509v3) and output the certificate as well as the extension information such as the subject key identifier.

```
openssl x509 -days 1095 -extfile ca.ext -signkey ca.key -in ca.csr -req -out ca.crt -text
```

Copy the subject key identifier from the output, convert the upper case characters to lower case characters, remove the colons and save the identifier to an additional file. It will be used in a bit.

Convert the certificate from the CRT format to the DER format.

```
openssl x509 -outform der -in ca.crt -out ca.der
```

Now, rename the certificate with the DER format with the stored subject key identifier.

6.2 Server Certificate

Create a private key:

```
openssl genrsa -out server.key 1024
```

Create the CSR using a config file:

```
openssl req -config server.config -new -key server.key -out server.csr
```

Create a file which stores the serials of the certificates issued by the CA:

```
echo -ne '01' > ca.serial
```

Create the x509 certificate using an extension file (x509v3) and output the certificate as well as the extension information such as the subject key identifier.

```
openssl x509 -days 730 -extfile server.ext -CA ca.crt -CAkey ca.key -CAserial ca.serial -in server.csr -req -out server.crt -text
```

Copy the subject key identifier from the output, convert the upper case characters to lower case characters, remove the colons and save the identifier to an additional file. It will be used in a bit. Convert the certificate from the CRT format to the DER format.

```
openssl x509 -outform der -in server.crt -out server.der
```

Now, rename the certificate with the DER format with the stored subject key identifier.

6.3 Client Certificate

Create a private key:

```
openssl genrsa -out client.key 1024
```

Create the CSR using a config file:

```
openssl req -config client.config -new -key client.key -out client.csr
```

Create the x509 certificate using an extension file (x509v3) and output the certificate as well as the extension information such as the subject key identifier.

```
openssl x509 -days 730 -extfile client.ext -CA ca.crt -CAkey ca.key -CAserial ca.serial -in client.csr -req -out client.crt -text
```

Copy the subject key identifier from the output, convert the upper case characters to lower case characters, remove the colons and save the identifier to an additional file. It will be used in a bit. Convert the certificate from the CRT format to the DER format.

```
openssl x509 -outform der -in client.crt -out client.der
```

Now, rename the certificate with the DER format with the stored subject key identifier.

7 Enabling HTTP(S) Updates

During the development of an application, the PAN9x20 might not contain the final configuration needed for the product. In this case it is necessary to tell the module which configuration to use by updating its configuration manually with a firmware update (fwu) file.

This firmware file required to enable updates using HTTP(S) consists of two binary files, which will be generated separately first and combined later on. The binary files contain the customer configuration file and the folder structure **cert/fwu** containing the CA, server and client certificates in DER format, named with the subject key identifier from within the certificates.

These are the detailed steps to generate the customer configuration binary file:

1. Amend the section [FIRMWARE-UPDATE] according to your needs.

Here is an example of the settings used for a local server:

```
[FIRMWARE-UPDATE]
server_adr = 192.168.2.100
server_port = 443
server_res = Resource.php
token = ""
security = 2
```

The attribute server_adr can either be an IP address as shown above or a DNS name.

2. Generate a binary file using the `CustomerConfig.exe` and the following command:

```
CustomerConfig.exe -i customer.cfg
```

These are the detailed steps to generate the binary file containing the certificates (if desired):

1. Create a folder named **cert**.
2. Inside this folder create a folder called **fwu**.
3. Inside the folder **cert/fwu** place the CA, server and client certificate named with the subject key identifier of the certificate in lower case letters and in the DER format.
4. Create a binary file using the tool `Dir2Flash.exe` and the following command:

```
Dir2Flash.exe --name=serialflash.bin --input-path=. \ --output-path=.
```

These two binary files need to be combined to the final firmware update file. In order to do that, create an additional file called `fwu.txt`. This file states where the files need to be stored in flash.

```
CustomerCfg.bin 0x0102 1 0
serialflash.bin 0x0101 2 0
```

The structure of this configuration file is: **file name**, **file type**, **file ID** and **version number**, which can be used multiple times to combine multiple files. A detailed description of the parameters can be found below.

Parameter	Description
File name	The file name including the path.
File type	The type of the file (firmware, configuration file, etc.).
File ID	The ID of the file. This specifies the desired location of the file in the memory.
Version	The version number of the file (currently not used by the tool <code>fwgen.exe</code>).

The possible file types and IDs can be found in the table below.

File Type	Description
0x0003	Host controller firmware. No file ID is used.
0x0101	Flash image. Images must be generated with <code>Dir2Flash.exe</code> . File IDs PAN9320: 0 = Internal Flash (Size: 987 KB) 1 = External Flash (Size: 2 MB) File IDs PAN9420: 0 = Internal Flash (Size: 2 MB)
0x0102	Configuration file. Config files must be generated with <code>CustomerConfig.exe</code> . File ID: 1 = Customer configuration file

Finally, create the firmware update file using the tool `fwgen.exe` with the following command for the PAN9320.

```
fwgen.exe -v 0.0.0.1 -c "Panasonic GmbH" -p PAN9320 -i fwu.txt
```

For the PAN9420 please use the following command.

```
fwgen.exe -v 0.0.0.1 -c "Panasonic GmbH" -p PAN9420 -i fwu.txt
```



Please note that the specified version (e.g. `-v 0.0.0.1`) has to be equal or higher than the one on the module. Otherwise, the module will not accept the file as a valid update file.

The current version can be requested using the following command:

```
get system version<\r><\n>
```

Now, the only thing left to do is to upload this file onto the PAN9x20 using the webserver address <http://192.168.1.1/LT3/index.html>.

8 Creating a Firmware Update File

As mentioned initially, software updates can address various targets such as the PAN9x20

- firmware,
- flash images (certificates, websites, etc.) and
- configuration as well as
- host controller firmware.

Firmware files for the module itself can only be created by Panasonic and will be provided to fix bugs in the module firmware where required.

In chapter [⇒ 7 Enabling HTTP\(S\) Updates](#) it was described how to create firmware update files containing certificates and a module configuration, thus the following description will describe how to create a host controller firmware update file.

1. Create a file that updates the host controller firmware. In the example the accompanying test file `HostUpdate.txt` with the following dummy content is used.

```
This is a firmware update for the Host Controller!
```

The host controller application then has to take care of the usage of the given firmware update information.

2. Create a new text file that tells the module which host controller firmware update file should be used. In the example the accompanying test file `hostupdate_cfg.txt` with the following content is used.

```
HostUpdate.txt 0x0003 0 0
```

3. Finally, create the firmware update file using the tool `fwgen.exe` with the following command for the PAN9320.

```
fwgen.exe -v 1.0.0.0 -c "Panasonic GmbH" -p PAN9320 -i hostupdate_cfg.txt
```

For the PAN9420 please use the following command.

```
fwgen.exe -v 1.0.0.0 -c "Panasonic GmbH" -p PAN9420 -i hostupdate_cfg.txt
```

The result of this process is a firmware update file called `FwOutput.fwu`. In the example this file was renamed to `firmware.fwu` and placed on the server as described in chapter [⇒ 9 XAMPP Apache Web Server](#).

9 XAMPP Apache Web Server

1. Install XAMPP with Apache – All other services are not used for this set up.
2. Go to **C:\xampp\apache\conf** and place the server certificate, CSR and private key in the appropriate folders.
3. Go to **C:\xampp\htdocs** and create a folder called **firmwares**. In there, create a folder for each firmware in the format "X.X.X.X" (e.g. 1.0.0.0). Inside these folders, place the appropriate fwu-file named `firmware.fwu`.
4. Go to **C:\xampp\htdocs** and place the accompanying `Resource.php` file. It checks the firmware requests and returns the firmware if available.
5. Go to **C:\xampp\htdocs** and place the accompanying `fwstatus.json` file. It provides information about the latest firmware available on the server.
6. Go to **C:\xampp\apache\conf**, open the file `httpd.conf` and bind Apache to the specific IP address and port in order to prevent Apache from listening to all bound IP addresses:

```
#Listen 80
Listen 192.168.2.100:443
```

7. Go to **C:\xampp\apache\conf\extra**, open the file `httpd-vhosts.conf` and add the following lines with the appropriate settings for your host to the end of the file:

```
<VirtualHost *:443>
    ServerAdmin admin@testserver.com
    DocumentRoot "C:/xampp/htdocs/"
    ServerName 192.168.2.100
    ErrorLog "logs/testserver-error.log"
    CustomLog "logs/testserver-access.log" common
    SSLEngine on
    SSLCertificateFile "C:/xampp/apache/ssl.crt/server.crt"
    SSLCertificateKeyFile "C:/xampp/apache/ssl.key/server.key"
</VirtualHost>
```

8. Connect the computer running the server to the local network.
9. Start the Apache server.



Please note that the specified server name (192.168.2.100) has to match the actual IP address of the server in use.

10 Performing a Firmware Update

There are two essential things to successfully perform a firmware update:

- ✓ The module has to be connected to the network the server is on. If the server is online, the module will need an internet connection.
- ✓ The time of the module has to be properly set up in order for the module to validate the server certificate.

After these things are taken care of, we can perform a firmware version check using the following command:

```
set fwu mode check<\r><\n>
```

This should return the response:

```
set fwu mode 0 check<\r><\n>
```

After that, use the following command to retrieve the information about the latest firmware version available on the server:

```
get fwu info<\r><\n>
```

This should return the response:

```
Template:      get fwu info 0 <timestamp> <version><\r><\n>
Example:      get fwu info 0 5 V1.0.0.0<\r><\n>
```

Use the following commands to update the module firmware:

```
set fwu mode update<\r><\n>
```

This should return the response:

```
set fwu mode 0 update<\r><\n>
```

Now, use the following status command to check whether the update process is active or an error occurred:

```
get fwu status<\r><\n>
```

This should return the response:

```
get fwu status 0 1<\r><\n>
```

The success of the update is indicated with the status number 1. If the update failed, which is indicated with the status number 4, please try to get a Wireshark instance in between the server and the client in order to check whether a data exchange could be established at first or which

side canceled the exchange due to various reasons. Exemplary Wireshark trace files are included in the folder accompanying this application note.

In this scenario of a host controller firmware update, the module does also provide an information message on the command UART:

```
info fwu file 0x3 0 0 50<\r><\n>
```

After that, the module internally forwards the received update information to the binary UART, which should read exactly the same as the update file:

```
This is a firmware update for the Host Controller!
```

Finally, the host controller has to handle the update information according to the application requirements.

11 Appendix

11.1 Contact Details

11.1.1 Contact Us

Please contact your local Panasonic Sales office for details on additional product options and services:

For Panasonic Sales assistance in the **EU**, visit

<https://eu.industrial.panasonic.com/about-us/contact-us>

Email: wireless@eu.panasonic.com

For Panasonic Sales assistance in **North America**, visit the Panasonic Sales & Support Tool to find assistance near you at

<https://na.industrial.panasonic.com/distributors>

Please visit the **Panasonic Wireless Technical Forum** to submit a question at

<https://forum.na.industrial.panasonic.com>

11.1.2 Product Information

Please refer to the Panasonic Wireless Connectivity website for further information on our products and related documents:

For complete Panasonic product details in the **EU**, visit

<http://pideu.panasonic.de/products/wireless-modules.html>

For complete Panasonic product details in **North America**, visit

<http://www.panasonic.com/rfmodules>