Content

Object Dictionary 1
File System 2
Application File System Interface Object (EAh) 2
Usage 3
Anybus File System Interface Object (0Ah) 3
Usage 3

Object Dictionary

The NP40 POWERLINK mode does not allow to access the object dictionary directly.

A XDD-File could read by scanning via Powerlink the INpact but it's not possible to read a XDD-File by the INpact and handle this values directly.

All necessary settings which must modified must set by the host application (IDL program on Windows or Linux) by the functions and NP40 objects. All other object dictionary settings must set by the POWERLINK master.

Here are some example objects I need to update within the object dictionary in order to properly integrate with the MN:

Object 1F98h: NMT_CycleTiming_REC

Subindex 3 = PResMaxLatency_U32 => set to 10000

➔ Is a const and cannot changed

Subindex 6 = ASndMaxLatency_U32 => set to 25000

➔ Is a const and cannot changed

Subindex 7 = MultiplCycleCnt_U8 => set to 4 (enable multiplexing)

Subindex 8 = AsyncMTU_U16 => set to 1500

➔ Must set by Master

Object 1F81h: NMT_NodeAssignment_AU32

Subindex 1 = NMT_NodeAssignment_AU32 => set to 0x80000207

Subindex 240 = NMT_NodeAssignment_AU32 => set to 0x80001001

➔ Must set by Master

Object 1F8Dh: NMT_PResPayloadLimitList_AU16

Subindex 1 = NMT_PResPayloadLimitList_AU16 => set to 1400

Subindex 240 = NMT_PResPayloadLimitList_AU16 => set to 200

➔ Must set by Master

The application has only access to the object entries described in chapter 4.1.3 where in the notes is a link to the POWERLINK object 0xE9.

This you'll find in chapter 11.2 Instance #1:

1 Vendor ID

2 Product Code

3 Revision high

4 Revision low

5 Serial Number Get UINT32 (set at production)

6 Manufacturer device name

7 Manufacturer hardware version

8 Manufacturer software version

10 Device type

14 Manufacturer name

And also indirect via Network Configuration Object (04h) things (Chapter 10.5 Instance #1 .. 15).

File System

Application File System Interface Object (EAh)

The folder "/applications" of the file system cannot used to do something for the NP40 or the used protocol stack itself.

It could be used to allow FTP clients to access files. These files must hold and managed by the Host application (PC software for Impact).

The NP40 send a request to the host application with command EAh, and this application must handle the files and access to it.

See chapter 13.6 Application File System Interface Object (EAh) and chapter 12.7 Anybus File System Interface Object (0Ah) in the software design guide: Software_DesignGuide_NP40_hms-hmsi-216-125_V4_0.pdf.

Usage

The command is sent to the host application to allow FTP clients and others access to files of the host.

Anybus File System Interface Object (0Ah)

Quotation from the Software Design Guide:

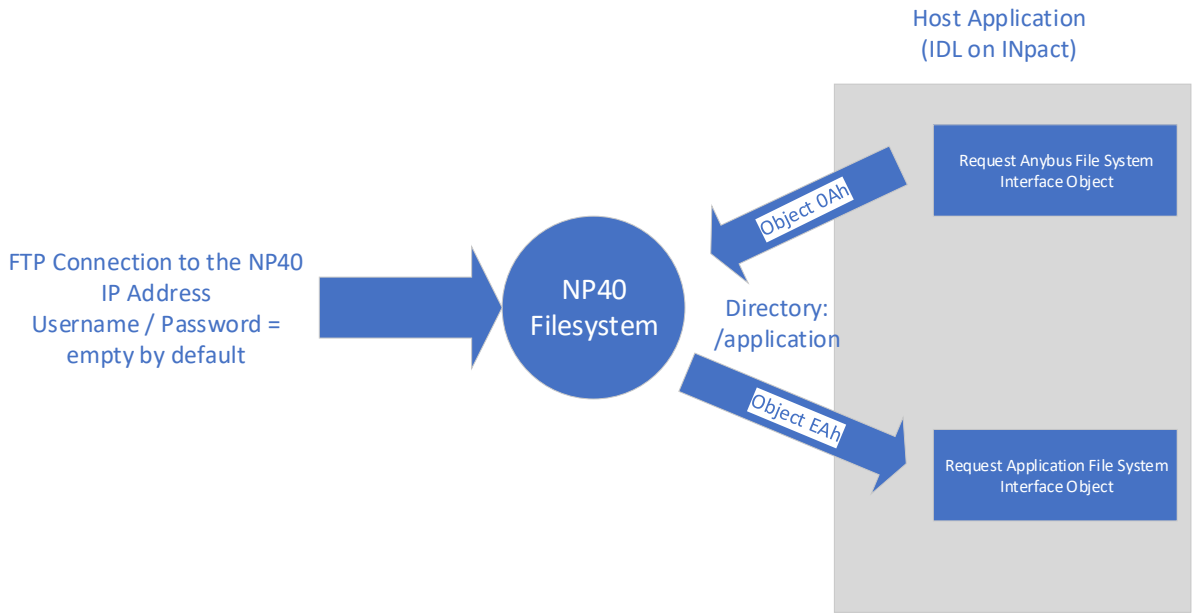
"For modules not supporting FTP, this makes it possible to store firmware files in the firmware directory using the File System Interface Object (0Ah). No other access to or use of the file system is possible for these modules."

Usage

So it is used to put a firmware to the folder "/firmware". This will update the NP40 at the next restart. Also it's possible to store some small files in "/vfs".

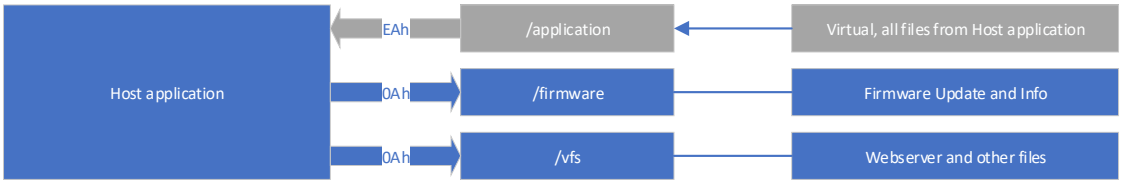
The filesystem on the NP 40 could be accessed by Object 0x0A.

When then the directory or a file should read (by Object 0x0A or a FTP client) then the NP40 requests this data via object 0xEA from the Host application.



File System

see Chapter 3.5 of the Software Design Guide



Scheme NP40 file system access