CiA® 814

Implementation and user guideline for CiA® 417

Part 1: Bootloader

Version: 1.0.0
07 December 2015

HISTORY

Date	Changes
2015-12-07	*Publication of Version 1.0.0* as application note NOTE: This document has been converted into “docx format”. The conversion caused minor layout differences to the predecessor document in “doc format”. The technical content word-by-word is the very same.

General information on licensing and patents

CAN in AUTOMATION (CiA) calls attention to the possibility that some of the elements of this CiA specification may be subject of patent rights. CiA shall not be responsible for identifying any or all such patent rights.

Because this specification is licensed free of charge, there is no warranty for this specification, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holder and/or other parties provide this specification “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the correctness and completeness of the specification is with you. Should this specification prove failures, you assume the cost of all necessary servicing, repair or correction.

Trademarks

CANopen and CiA are registered community trademarks of CAN in Automation. The use is restricted for CiA members or owners of CANopen vendor ID. More detailed terms for the use are available from CiA.

CANwizard® is a registered trademark of BÖHNKE + PARTNER GmbH. More detailed terms for the use are available from Schmersal GmbH (formerly: BÖHNKE + PARTNER GmbH).

CAN in Automation e. V.
Kontumazgarten 3
DE - 90429 Nuremberg, Germany
Tel.: +49-911-928819-0
Fax: +49-911-928819-79
Url: www.can-cia.org
Email: headquarters@can-cia.org

CONTENTS

1 Scope

This application note provides implementation hints for CiA 417's bootloader (CiA 417 version 2.2.0 and higher).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

/CiA301/ CiA 301 version 4.2.0, CANopen application layer and communication profile

/CiA302-2/ CiA 302 version 4.1.0, Additional application layer functions Part 2 Network management

/CiA417-1/ CiA 417-1 version 2.2.0, CANopen application profile for lift control systems – Part 1: General definitions

/CiA417-2/ CiA 417-2 version 2.2.0, CANopen application profile for lift control systems – Part 2: Virtual device definitions

/CiA417-3-1/ CiA 417-3-1 version 2.1.0, CANopen application profile for lift control systems – Part 3-1: Pre-defined PDOs for lift application 1

/CiA417-4/ CiA 417-4 version 2.2.0, CANopen application profile for lift control systems – Part 4: Detailed application object specification

/CiA801/ CiA 801 version 1.1.0, CANopen automatic bit-rate detection

/ISO8859-1/ ISO/IEC 8859-1:1998 Information technology -- 8-bit single-byte coded graphic character sets -- Part 1: Latin alphabet No. 1

3 Terms and definitions

For the purpose of this document, the following terms and definitions and those given in /CiA301/, /CiA302-2/, /CiA417-1/, /CiA417-2/, /CiA417-3-1/ and /CiA417-4/ apply.

4 Symbols and abbreviated terms

For the purpose of this document, the following symbols and abbreviated terms and those given in /CiA301/, /CiA302-2/, /CiA417-1/, /CiA417-2/, /CiA417-3-1/ and /CiA417-4/ apply.

5 Node-ID assignment

The bootloader uses a fixed node-ID $7E_h$ (126_d).

6 Operating principles

6.1 General

This clause provides a description of the firmware download principles of CANopen lift bootloader (see /CiA417-1/ and /CiA417-2/).

6.2 Functional description

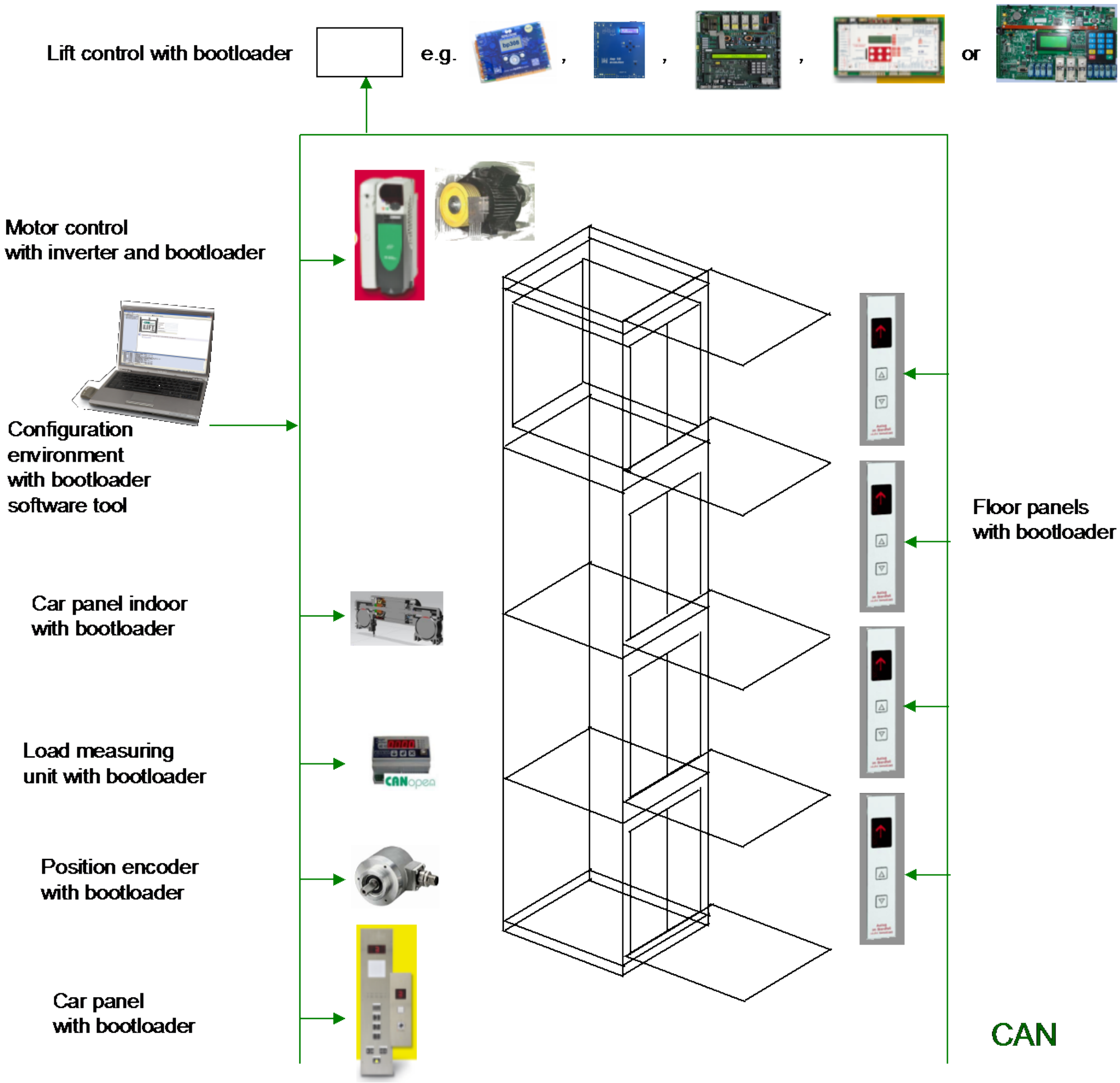
The program development of CiA 417 compliant devices is an on-going process. Sometimes it is necessary to update the firmware of already installed devices. For this purpose the CANopen lift bootloader functionality is included in the CiA 417 specification. With this technique it is possible to update devices in the CANopen lift network using any CiA 417 compatible flash tool (see Figure 1).

Inside the CANopen lift device the software is split into two parts: the bootloader software and the application software (see Figure 2). The bootloader software is fixed in the device and is not overwritten. Only the application software is updated. The bootloader supports minimum number of CANopen communication services as for example NMT, SDO and error control (i.e. Heartbeat).

The bootloader has mainly two functions:

— On start-up the bootloader is always first executed. It checks whether the bootloader code itself is to be activated. Additionally, it checks for a valid application program. If the application code is valid, this code is executed and the device resumes operation.

— If the bootloader is called up, it waits for commands sent using CAN and provides the functions to update the application (e. g. erase flash, program flash, verify flash).



Legend

Arrows show direction of the firmware or application settings download.

Figure 1 – Example of CANopen lift network with bootloader capabilities

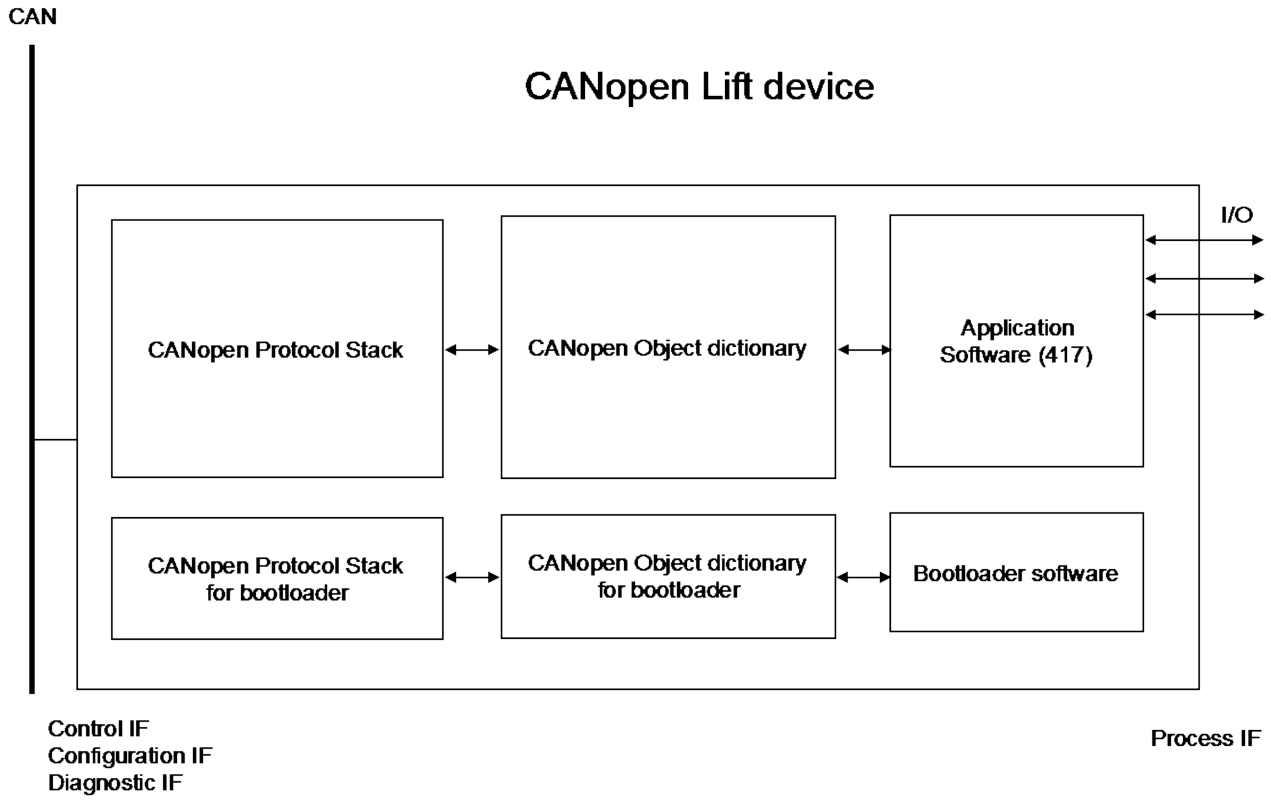
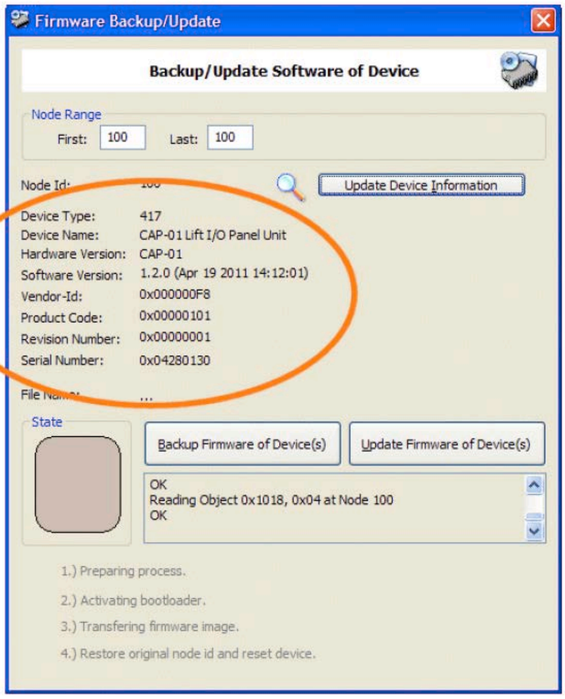

CAN
CANopen Lift device
CANopen Protocol Stack
CANopen Object dictionary
Application Software (417)
I/O
CANopen Protocol Stack for bootloader
CANopen Object dictionary for bootloader
Bootloader software
Control IF
Configuration IF
Diagnostic IF
Process IF

Figure 2 – Example of CANopen lift devices' internal structure

6.3 Update process

The basic process of update the application software is shown with on example of a CANwizard tool. The update tool reads object 1000_h, 1008_h, 1009_h, $100A_h$, 1018 01_h, 1018 02_h, 1018 03_h, 1018 04_h from the object dictionary of the node to be updated.


Firmware Backup/Update
Backup/Update Software of Device
Update Device Information
Backup Firmware of Device(s)
Update Firmware of Device(s)

Node Range
First: 100 Last: 100

Node Id: 100

Device Type: 417
Device Name: CAP-01 Lift I/O Panel Unit
Hardware Version: CAP-01
Software Version: 1.2.0 (Apr 19 2011 14:12:01)
Vendor-Id: 0x000000F8
Product Code: 0x00000101
Revision Number: 0x00000001
Serial Number: 0x04280130

File Name: ...

State

```
OK
Reading Object 0x1018, 0x04 at Node 100
OK
```

1.) Preparing process.

2.) Activating bootloader.

3.) Transfering firmware image.

4.) Restore original node id and reset device.

Figure 3 – Example of firmware update tool dialog part 1

Even if these data is not required for updating the firmware, it helps a user to prevent accidentally accessing the wrong node.

The update process consists of several steps:

- Preparing process

  The tool verifies, if the bootloader mode is active in the device. Also it verifies, if any other node is already in bootloader mode. The update tool warns a user, if any heartbeat from node-ID 126_d is already detected, before the update process starts.

  If there are other NMT masters in the network, the tool requests the NMT master privilege for controlling the network behaviour of nodes. For this purpose it uses the NMT flying master. For details on flying master see /CiA302-2/.

- Activating bootloader

  The next step includes the start of the bootloader by writing a password phrase to object 6005_h. If the application of the device does not signal any abort code, the update tool restarts the device by writing the “Reset program” command to object $1F51_h$. Alternatively, the tool sends an NMT application reset command as a trigger to activate the bootloader. The application in the device activates its bootloader by that. This is done by any kind of global variable or signal to start up the underlying bootloader code.

  After the bootloader activation, the node boots up using the node-ID 126_d. The update tool waits up to 10s for the first heartbeat of the bootloader. While waiting for the first heartbeat of node 126_d, the update tool tries reading object 1000_h “Device Type” from the bootloader every 125 ms to provide assistance, if the node is the only one on the CAN network and uses the automatic bit-rate detection mechanism (see /CiA801/). The content of the object 1000_h is a FE00 $01A1_h$ for a CiA 417 bootloader or the older code 424F $4F54_h$.

- Data transfer

  After receiving the heartbeat from bootoader (NMT Pre-Operational), the update tool writes the password phrase again to object 6005_h “Lock/unlock parameters: Firmware update”. If the bootloader does not generate any abort code, the update tool starts reading (backup firmware) or writing (update firmware) to object $1F50_h$.

- Restart device

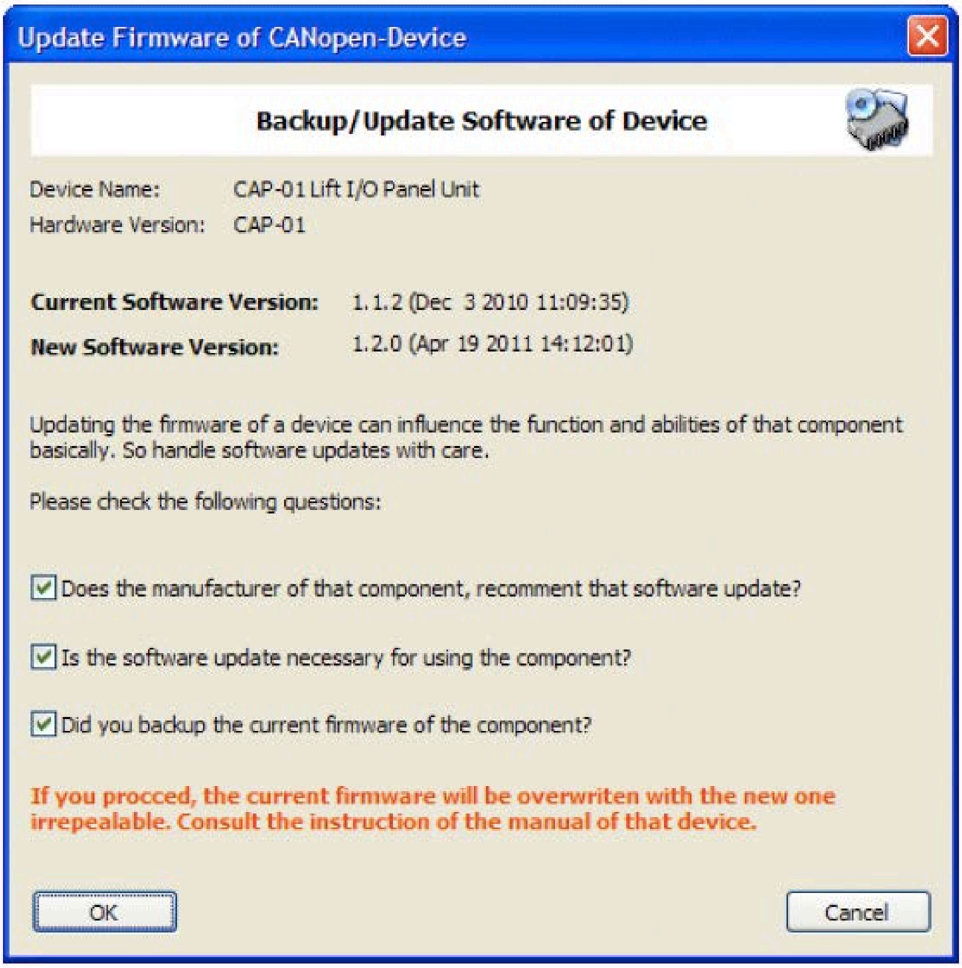
  After successful download of the data, the tool restarts the device by writing the command “Reset program” to object $1F51_h$. If this object does not exist, the tool sends the NMT application reset command.

  By this the bootloader activates the new application program, if its vendor and product specific internal unique identifier and its CRC identify it as a valid application. If a valid application is detected it starts up immediately using the original node-ID. If no valid application program is detected, the bootloader keeps running or restarts automatically, even if the node is powered up, keeping node-ID 126_d.

6.4 Backup / Update other application data

For backing up or updating other application data, like parameter sets or EEPROM images the same procedure is applied, by reading or writing from or to another sub-index of the object $1F50_h$.

The object 1F50 01_h is used for firmware download and the object 1F50 02_h is used for parameter set download.



Backup/Update Software of Device

Device Name: CAP-01 Lift I/O Panel Unit
Hardware Version: CAP-01

Current Software Version: 1.1.2 (Dec 3 2010 11:09:35)
New Software Version: 1.2.0 (Apr 19 2011 14:12:01)

Updating the firmware of a device can influence the function and abilities of that component basically. So handle software updates with care.

Please check the following questions:

☑ Does the manufacturer of that component, recomment that software update?

☑ Is the software update necessary for using the component?

☑ Did you backup the current firmware of the component?

If you procced, the current firmware will be overwriten with the new one irrepealable. Consult the instruction of the manual of that device.

Figure 4 – Example of firmware update tool dialog part 2

6.5 Version handling

To provide a simple way for the update tool to read out the version information of a binary image file, the content of object 100Ah is represented as a visible string in the image file. In /CiA417-1/ the application program file format is defined. With the prefix "$CANopenSoftwareVersion" the software version string is identified.

EXAMPLE Code in C-programming language:

```
const char sw_version[] = { "$CANopenSoftwareVersion\0" /*+24*/ "1.2.0 ("__DATE__", "__TIME__")" };
/* 0x100A Manufacturer software version */
{ OBJ_MAKE_ID( 0x100A, 0 ), ATTR_CONST, VISIBLE_STRING, sw_version + 24 },
```

This provides a simple and platform independent method of parsing the binary image file for the software version. The prefix "$CANopenSoftwareVersion" is a visible and NULL terminated string that is encoded as /ISO8859-1/.

Checking for proper Vendor-ID, product code and format of the firmware image file is done by the bootloader rather than the update tool. If it is a wrong firmware file, the bootloader generates the abort code 0800 0020h "Data cannot be transferred or stored to the application", when the update tool writes to object 1F50h "Program download".