CiA Draft Standard Proposal 305

Layer setting services (LSS) and protocols

This DSP is for CiA members only and may be changed without notification.

Version: 2.2

26 August 2008

HISTORY

Date	Changes
2000-05-31	*Publication of version 1.0 as draft standard proposal*
2002-02-01	*Publication of version 1.1 as draft standard proposal*
2002-11-05	*Publication of version 1.1.1 as draft standard proposal*
2005-12-31	*Publication of version 2.0 as draft standard proposal*
2008-08-26	*Publication of version 2.2 as draft standard proposal* • Added LSS Fastscan service

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.

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 5
2 Normative references 5
3 Abbreviations and definitions 5
3.1 Abbreviations 5
3.2 Definitions 5
4 Operating principles 6
4.1 Introduction 6
4.2 LSS master device and LSS slave devices 6
4.3 LSS services, protocols and messages 6
4.4 Performance requirements 7
4.5 Finite state automaton 7
4.5.1 Introduction 7
4.5.2 LSS finite state automaton 7
4.6 Switch state services 9
4.6.1 Introduction 9
4.6.2 Switch state global service 9
4.6.3 Switch state selective service 9
4.7 Configuration services 9
4.7.1 Introduction 9
4.7.2 Configure node-ID service 9
4.7.3 Configure bit timing parameters service 10
4.7.4 Activate bit timing parameters service....10
4.7.5 Store configuration service 11
4.8 Inquiry services 12
4.8.1 Introduction 12
4.8.2 Inquire LSS address service 12
4.8.3 Inquire node-ID service....12
4.9 Identification services 13
4.9.1 Introduction 13
4.9.2 LSS identify remote slave service 13
4.9.3 LSS identify slave service 13
4.9.4 LSS identify non-configured remote slave service 13
4.9.5 LSS identify non-configured slave service....13
4.10 The Fastscan service 13
4.10.1 Introduction 13
4.10.2 LSS Fastscan service 13
4.10.3 LSS Fastscan procedure 15
5 LSS protocols 20
5.1 Introduction 20
5.2 LSS slave synchronization 20
5.3 LSS protocol definitions 20
5.4 Switch mode protocols 20
5.4.1 Switch state global protocol 20
5.4.2 Switch state selective protocol....21
5.5 Configuration protocols 22
5.5.1 Configure node-ID protocol 22

5.5.2 Configure bit timing parameters protocol....23
5.5.3 Activate bit timing parameters protocol....24
5.5.4 Store configuration protocol....25
5.6 Inquiry protocols....26
5.6.1 Inquire LSS address protocols....26
5.6.2 Inquire node-ID protocol....28
5.7 Identification protocols....28
5.7.1 LSS identify remote slave protocol....28
5.7.2 LSS identify slave protocol....30
5.7.3 LSS identify non-configured remote slave protocol....30
5.7.4 LSS identify non-configured slave protocol....30
5.7.5 LSS Fastscan protocol....31
A.1 Implementation and application hints....31
A.1.1 Invalid LSS messages....31
A.1.2 Timeouts....31
A.1.3 Usage of LSS services....32
A.1.4 Special hints for LSS Fastscan....33
Annex B Corrigendum....36
B.1 Clause 4.10.2....36
B.2 Clause 4.10.3....36

1 Scope

This document specifies the layer setting services (LSS) and protocols for CANopen. These services and protocols are used to inquire or to change the settings of several parameters of the physical layer, data link layer, and application layer on a CANopen device with LSS slave capability by a CANopen device with LSS master capability via the CAN network.

The following parameters may be inquired or changed:

- Node-ID of the CANopen device
- Bit timing parameters of the physical layer (bit rate)
- LSS address compliant to the identity object (1018_h)

2 Normative references

/ISO11898-1/ ISO 11898-1, Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signaling

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

/CiA205-1/ CiA 205-1:1996, CAN application layer for industrial applications – Part 1: LMT service specification

/CiA205-2/ CiA 205-2:1996, CAN application layer for industrial applications – Part 2: LMT protocol specification

3 Abbreviations and definitions

3.1 Abbreviations

FSA Finite state automata

LMT Layer management

LSS Layer setting services

3.2 Definitions

The definitions given in /ISO11898-1/ and /CiA301/ apply to this specification, too.

automatic bit rate detection
feature of a CAN interface to adjust automatically to the bit rate of other CAN interfaces

LSS address
128-bit number to identify each node uniquely, consisting of the vendor-ID, product code, revision number and serial number with 32 bit each, defined in /CiA301/ object 1018_h

LSS number
each of the four 32-bit values of the LSS address

IDNumber
variable used by LSS Fastscan protocol to evaluate an LSS number

4 Operating principles

4.1 Introduction

By using layer setting services and protocols an LSS slave device may be configured via the CAN network without using any hardware components like DIP-switches for setting the node-ID or bit timing parameters.

The LSS slave device shall support the following services:

- Switch state selective
- Switch state global
- Store configured parameters

For configuring the bit timing, the following LSS services are mandatory in addition:

- Configure bit timing parameters
- Activate bit timing parameters

For configuring the node-ID the following LSS services are mandatory in addition:

- Configure node-ID

All other services defined in this specification may be implemented additionally.

4.2 LSS master device and LSS slave devices

The CANopen device that configures other devices via the CANopen network is called the LSS master device. There shall be only one LSS master device in a network. The LSS master device has no attributes. It shall reside in the CANopen device with NMT master capability.

The CANopen device that is configured by the LSS master device via a CANopen network is called the LSS slave device. The number of LSS slave devices in a network is not limited. The LSS slave device has the following attributes:

The LSS address consists of vendor-id, product-code, revision-number and serial number. The value shall be identical to the corresponding values given in the CANopen identity object (1018h). There shall exist no other LSS slave device with the very same LSS address within the same network.

The node-ID is valid if it is in the range of 01h to $7F_h$; a node-ID of FF_h identifies a not configured CANopen device. Node-IDs in the range of 80_h to FE_h shall not be used.

NOTE If only one LSS slave device is connected to the LSS master device, it is not necessary to provide a unique LSS address.

4.3 LSS services, protocols and messages

By means of LSS services, the LSS master device requests services to be performed by the LSS slave devices. These services shall be implemented by the protocols as defined in this specification.

Communication between LSS master device and LSS slave devices is accomplished by the LSS protocols. The LSS protocols shall use only two COBs that shall have a DLC of 8:

- LSS master message
- LSS slave message

The LSS master message shall exist exactly once in a CANopen network. The LSS master message sends commands and configuration parameters to the LSS slave device.

The LSS slave device may respond to the LSS master messages by LSS slave messages. If the LSS slave message does not contain variable parameters, it may be used simultaneously by several LSS slave device. If the LSS slave message contains variable parameters, it shall be used only by one LSS slave device.

4.4 Performance requirements

The manufacturer of an LSS slave device is responsible to indicate the maximum bit rate, at which the LSS slave device is able to handle in minimum a burst of six consecutive LSS messages using the very same CAN-ID. The manufacturer is also responsible to indicate the maximum number of messages send in a burst, which the LSS slave device is able to handle.

NOTE For higher bit rates, it is possible to use an LSS master device capable not to burst LSS messages but send them with delays, or to use only switch mode global services in peer-to-peer communication between the LSS master and the LSS slave device.

4.5 Finite state automaton

4.5.1 Introduction

The LSS FSA defines the behavior of the LSS slave device. It is an abstraction to describe the LSS slave device behavior as the LSS master device experiences it. It is controlled by LSS COBs produced by the LSS master device, or NMT COBs produced by the NMT master device, or local NMT state transitions. Figure 1 shows the remote and local control of the LSS FSA. NMT local state transition may be initiated due to internal events, e.g. the error behavior (1029_h) object (see /CiA301/). The logical operation does not fall into the scope of this specification.

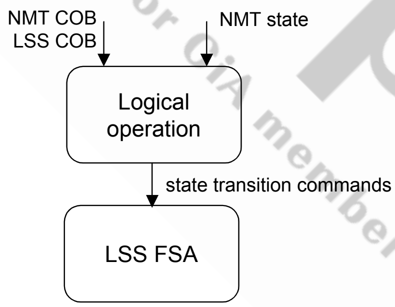


Figure 1 – Remote and local control

NOTE The LSS FSA does not provide state information.

4.5.2 LSS finite state automaton

The LSS FSA is defined in Figure 2 and it shall provide the following states:

[0] Initial: Pseudo state, indicating the activation of the FSA.

[1] LSS waiting: In this state, only the services as defined in Table 1 shall be supported.

[2] LSS configuration: In this state, only the services as defined in Table 1 shall be supported.

[3] Final: Pseudo state, indicating the deactivation of the FSA.

The LSS services shall be supported according to the current state of the LSS FSA as stated in Table 1.

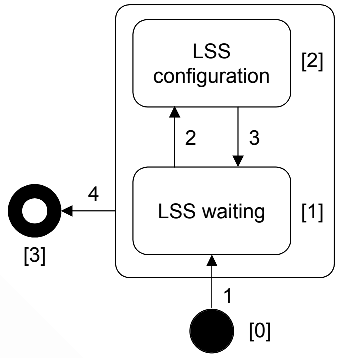


Figure 2 – LSS finite state automaton

NOTE The LSS slave device has not the capability to verify if other LSS slave devices are also in configuration state. This means the LSS master device is responsible for correctness and sequence of LSS service requests.

Table 1 – State behavior and supported services

Services	LSS waiting	LSS configuration
Switch state global	yes	yes
Switch state selective	yes	no
Activate bit timing parameters	no	yes
Configure bit timing parameters	no	yes
Configure node-ID	no	yes
Store configuration	no	yes
Inquire LSS address	no	yes
Inquire node-ID	no	yes
LSS identify remote slave	yes	yes
LSS identify slave	yes	yes
LSS identify non-configured remote slave	yes	yes
LSS identify non-configured slave	yes	yes
LSS Fastscan	yes	no

The LSS FSA shall support the following transitions as defined in Table 2.

Table 2 – Events and actions

Transition	Event(s)	Actions(s)
1	Automatic transition after initial entry into either NMT pre-operational state, or NMT stopped state, or NMT reset communication sub-state with node-ID equals FF_h (NOTE)	none
2	LSS switch state global command with parameter 'configuration_switch' or switch state selective command	none
3	LSS switch state global command with parameter 'waiting_switch'	none
4	Automatic transition if invalid node-ID has been changed and the new node-ID has been successfully stored in non-volatile memory AND state switch to LSS waiting was commanded	none
NOTE Once the LSS FSA is entered further state transitions in the NMT FSA from NMT pre-operational to NMT stopped state and vice versa does not lead to re-entering the LSS FSA		

4.6 Switch state services

4.6.1 Introduction

The switch state services control the LSS finite state automaton of the LSS slave device.

4.6.2 Switch state global service

By means of the service defined in Figure 3, the LSS master device shall switch all LSS slave devices in the network into LSS waiting state or LSS configuration state.

Parameter	Request/indication
Argument	**Mandatory**
switch	Mandatory
waiting_switch	Selection
configuration_switch	Selection

Figure 3 – Switch state global service

4.6.3 Switch state selective service

By means of the service defined in Figure 4, the LSS master device shall switch the LSS slave device into LSS configuration state, whose LSS address attributes equals the LSS_address.

Parameter	Request/indication	Response/confirmation
Argument	**Mandatory**	
LSS_address	Mandatory	
Remote result		**Mandatory**

Figure 4 – Switch state selective service

4.7 Configuration services

4.7.1 Introduction

The configuration services are used to configure the node-ID or the bit timing parameters and to activate the configured bit timing parameters. There is also a service to store the configuration.

4.7.2 Configure node-ID service

By means of the service defined in Figure 5, the LSS master device shall configure the node-ID of a single LSS slave device. The remote result parameter shall confirm the success or failure of the service. In case of failure, the reason may be provided.

Parameter	Request/indication	Response/confirmation
Argument	**Mandatory**	
Node-ID	mandatory	
Remote result		**Mandatory**
success		selection
failure		selection
reason		optional

Figure 5 – Configure node-ID service

NOTE 1 This service does not store automatically the configured node-ID. This is performed with the store configuration service.

NOTE 2 The LSS master device is responsible to switch one and only one LSS slave device into LSS configuration state before requesting this service.

4.7.3 Configure bit timing parameters service

By means of the service defined in Figure 6, the LSS master device shall configure the new bit timing on a single LSS slave device. The remote result parameter shall confirm the success or failure of the service. In case of failure the reason may be provided.

By means of the table_selector the bit timing parameter table to be used shall be specified. In the bit timing parameter table the bit timing parameters for different bit rates are specified. With table_selector value ´0´ the standard CiA bit timing parameter table shall be referenced (see Table 3). The table_index shall select the entry (bit rate) in the selected table (value '0' refers to the highest bit rate).

The service shall not store autonomously the bit timing parameters; this may be done with the store configuration service.

Parameter	Request/indication	Response/confirmation
Argument	**Mandatory**	
table_selector	mandatory	
table_index	mandatory	
Remote result		**Mandatory**
success		selection
failure		selection
reason		optional

Figure 6 – Configure bit timing parameters service

NOTE The LSS master device is responsible to switch one and only one LSS slave device into LSS configuration state before requesting this service.

Table 3 shows indices of the bit rates defined in /CiA301/.

Table 3 – Bit rate indices

Index	Bit rate
0	1 Mbit/s
1	800 kbit/s
2	500 kbit/s
3	250 kbit/s
4	125 kbit/s
5	reserved
6	50 kbit/s
7	20 kbit/s
8	10 kbit/s
9	Automatic bit rate detection

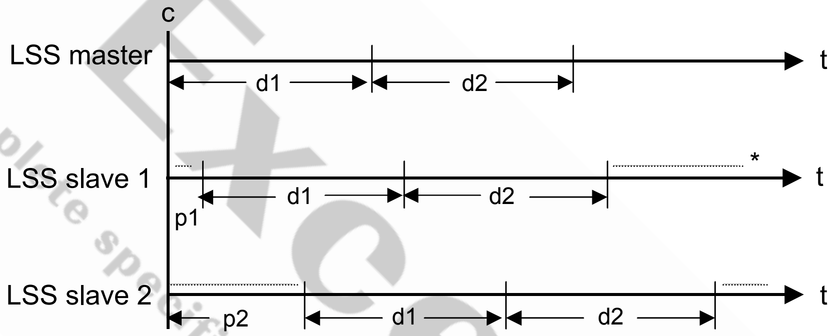
4.7.4 Activate bit timing parameters service

By means of the service defined in Figure 7, the LSS master shall activate the bit timing as defined by the configure bit timing parameters service.

Parameter	Request/indication
Argument	**Mandatory**
switch_delay	mandatory

Figure 7 – Activate bit timing parameters service

The switch_delay parameter shall specify the length of two delay periods of equal length, which are necessary to avoid operating the bus with differing bit timing parameters. Each LSS slave device shall perform the actual switch of the bit timing parameters ´switch_delay´ (given in milliseconds) after service indication. After performing the switch, a device shall not transmit any messages before the second ´switch_delay´ time has passed. Figure 8 shows an example of this timing behavior.



c: initiation of command
p1, p2: individual processing delay
d1: duration of first switch_delay period
d2: duration of second switch_delay period

* Dotted lines indicate that a CANopen device may transmit CAN messages

Figure 8 – LSS master and LSS slave timing

NOTE LSS slave devices have different processing times for performing the activate bit timing parameters command and messages that are transmitted before this command may still be in the receive queue of a device. This means that a device may still transmit CAN messages with the previous bit timing during the duration of the processing delay. Therefore switch_delay should be longer than the longest processing time of any LSS slave device in the network in order to avoid that a device already switches while another device still transmits using the previous bit timing parameters. After the time specified by switch_delay has passed the first time, every device shall perform the switch during the second duration of switch_delay. Therefore after switch_delay has passed the second time, all devices are guaranteed to be listening with the new bit timing parameters.

4.7.5 Store configuration service

By means of the service defined in Figure 9, the LSS master device shall request the LSS slave device to store the configured parameters in non-volatile memory. The remote result parameter shall confirm the success or failure of the service. In case of failure the reason may be provided.

Parameter	Request/indication	Response/confirmation
Argument	**Mandatory**	
Remote result		**Mandatory**
success		selection
failure		selection
reason		optional

Figure 9 – Store configuration service

NOTE The LSS master device is responsible to switch one and only one LSS slave device into LSS configuration state before requesting this service.

4.8 Inquiry services

4.8.1 Introduction

These services are used to inquire the LSS address or the node-ID. By means of the Inquire LSS address service, the LSS master device shall request the LSS address (vendor-id, product-code, revision-number, serial-number) from the LSS slave device. By means of the Inquire node-ID service, the LSS master device shall request the actual node-ID of the LSS slave device. The inquiry services shall only be executed in LSS configuration state.

4.8.2 Inquire LSS address service

By means of the service defined in Figure 10, the LSS master device shall determine a single LSS-address parameter of one LSS slave device. The remote result parameter shall confirm the LSS address parameters.

Parameter	Request/indication	Response/confirmation
Argument	**Mandatory**	
code	selection	
vendor-id code	mandatory	
product-code code	mandatory	
revision-number code	mandatory	
serial-number code	mandatory	
Remote result		**Mandatory**
LSS_address		selection
vendor-id		mandatory
product-code		mandatory
revision-number		mandatory
serial-number		mandatory

Figure 10 – Inquire LSS address service

NOTE The LSS master device is responsible to switch one and only one LSS slave device into LSS configuration state before requesting this service.

4.8.3 Inquire node-ID service

By means of the service defined in Figure 11, the LSS master device shall determine the node-ID of the LSS slave device that is in LSS configuration state. The remote result parameter shall be the node-ID.

Parameter	Response/confirmation
Argument	
Remote result	**Mandatory**
node-ID	mandatory

Figure 11 – Inquire node-ID service

NOTE The LSS master device is responsible to switch one and only one LSS slave device into LSS configuration state before requesting this service.

4.9 Identification services

4.9.1 Introduction

These services identify LSS slave devices.

4.9.2 LSS identify remote slave service

By means of the service defined in Figure 12, the LSS master device shall request all LSS slave devices to identify themselves by means of the 'LSS identify slave' service, whose LSS address meets the LSS_Address_sel. The LSS_Address_sel shall consist of a single vendor-ID and a single product code and a span of revision and serial numbers determined by a low and high number. This service shall be unconfirmed.

Parameter	Request/indication
Argument	**Mandatory**
LSS_address_sel	mandatory

Figure 12 – LSS identify remote slave service

4.9.3 LSS identify slave service

By means of this service, an LSS slave device shall indicate, that it is a slave with an LSS address within the LSS_Address_sel given by an LSS identify remote slave service executed prior to this service. The service shall be unconfirmed.

4.9.4 LSS identify non-configured remote slave service

By means of this service, the LSS master device shall request all LSS slave devices to identify themselves by means of the 'LSS identify non-configured slave' service, whose node-ID is not valid (FF_h). This service shall be unconfirmed.

4.9.5 LSS identify non-configured slave service

By means of this service, an LSS slave device shall indicate, that it is an LSS slave device, whose node-ID is not valid (FF_h), if an LSS identify non-configured remote slave service is executed by the LSS master device prior to this service. The service shall be unconfirmed.

4.10 The Fastscan service

4.10.1 Introduction

LSS Fastscan may reduce the time of a network scan for the following reasons:

- Master is sending only one instead of six remote identification messages.
- It is possible to identify devices although skipping LSS numbers, e.g. the serial number may be ignored.
- It is possible to identify vendor-ID and product code.

4.10.2 LSS Fastscan service

By means of the LSS Fastscan service defined in Figure 13, the LSS master shall request all un-configured LSS slaves to identify themselves. The parameters of Figure 13 are described in Table 4.

NOTE The order of the nodes to be configured depends on the nodes' LSS address. The '0' has a higher priority than the '1'. The scan is starting at the most significant bit 31.

Parameter	Request/Indication
Argument	**Mandatory**
IDNumber	Mandatory
BitChecked	Mandatory
LSSSub	Mandatory
LSSNext	Mandatory

Figure 13 – LSS Fastscan service

Table 4 – Parameter description

Name	Data type	Description
IDNumber	Unsigned32	Depending on the value of LSSSub this variable evaluates one of the four LSS numbers (vendor ID, product code, revision or serial number), which the LSS master currently tries to identify. At the beginning of the identification process its value is 0000 0000_h, at the end it is equal to the appropriate LSS number.
BitChecked	Unsigned8	This value shall determine which bits of the current IDNumber are being checked. It is defined in Table 5. For BitChecked:= *n*, the bits *n* to 31 of the slave's current LSS number are compared with the IDNumber (see above).
LSSSub	Unsigned8	This value shall select, which part of the LSS-ID is currently transmitted in IDNumber. The values are defined in Table 6.
LSSNext	Unsigned8	The LSS master shall use this parameter to inform the LSS slaves which LSSSub value will be used in the next request. Only if an LSS slave has responded (acknowledged) to the current request, it shall continue in the next cycle.

Table 5 – Value definition of BitChecked

Value	Definition
00_h to $1F_h$ (00_d to 31_d)	Lowest bit of IDNumber's bit area to be checked
20_h to $7F_h$ (32_d to 127_d)	Reserved
80_h (128_d)	All LSS slaves respond and all slaves are reset
81_h to FF_h (129_d to 255_d)	Reserved

Table 6 – Value definition of LSSSub

Value	LSS number
0	Vendor ID
1	Product code
2	Revision number
3	Serial number
4 to 255	Reserved

Any LSS slave which is still participating to the LSS Fastscan service, shall send the message LSS identify slave to acknowledge matching parts of its current LSS number.

4.10.3 LSS Fastscan procedure

The LSS Fastscan procedure is defined by Figure 14, Figure 15 and Figure 16. LSSPos is a variable in the LSS slave that keeps track of which part of the LSS number is currently processed. Table 7 shows the value definition.

With LSSPos equal to FF_h, the LSS slave shall not participate the service anymore until BitChecked equal to 128 (80_h) is received.

Table 7 – Value definition of LSSPos

LSSPos	Description
00_h	Vendor-ID
01_h	Product code
02_h	Revision number
03_h	Serial number
04_h to FE_h	reserved
FF_h	Not participating

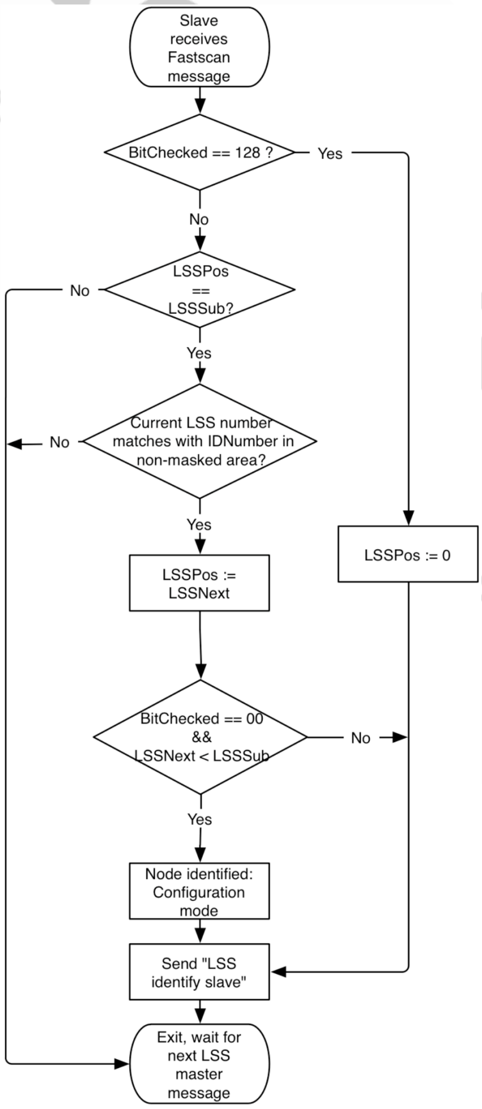


Figure 14 – LSS slave's view of LSS Fastscan procedure

To scan for a complete 32-bit LSS number, the LSS master shall process 33 times the service LSS Fastscan. In 32 rounds, bit 0 to 31 are determined. If the least significant bit of the LSS number on the LSS slave is ‘1’ there will be no acknowledge with the service LSS identify slave. To ensure, that the LSS number has been determined correctly, round 33 transmits the complete LSS number in order to receive an acknowledge.

NOTE An LSS master shall always process one additional round, whether the last bit is ‘1’ or not.

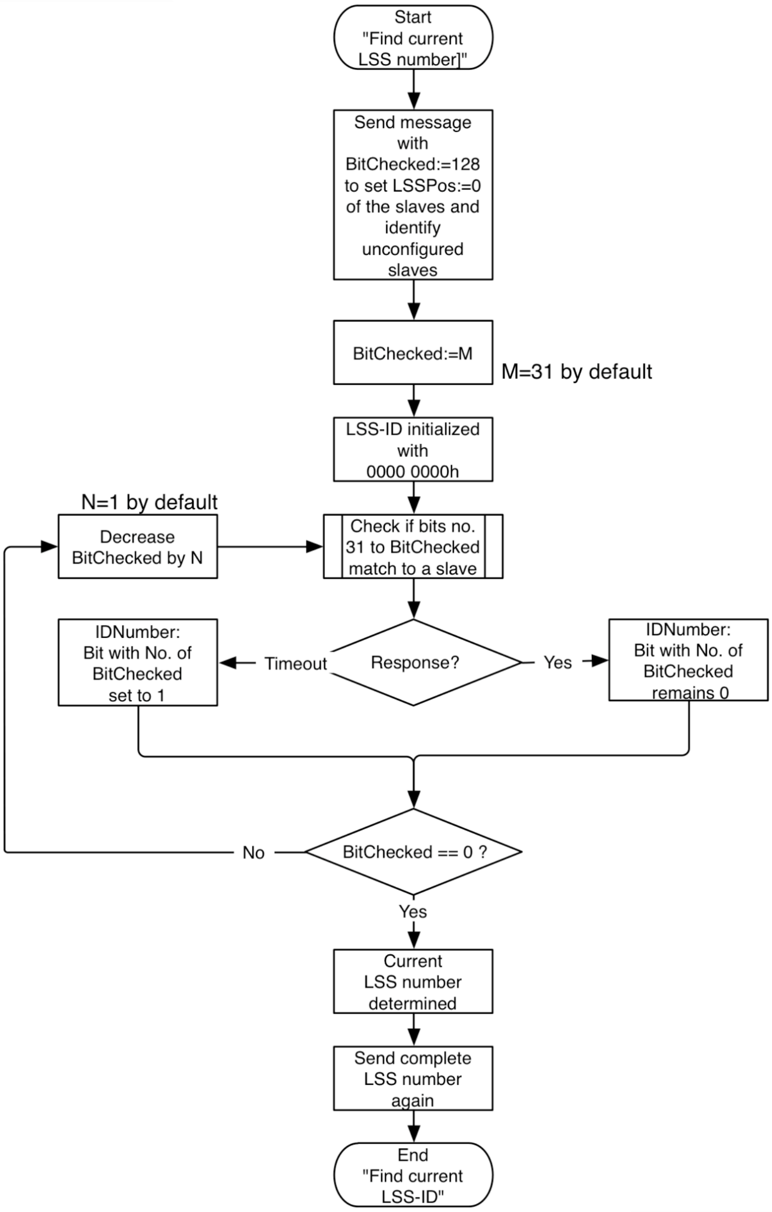


Figure 15 – LSS master's view of detecting exactly one LSS number

After all bits of an LSS number are identified, LSSSub is set to LSSNext.

The LSS Fastscan capable master shall repeat these scan cycles for all four 32 bit values of the 128 bit LSS address as shown in Figure 16. The last message sent by the LSS master shall contain the complete serial number, a BitChecked value of 0, an LSSSub value of 3 and an LSSNext value of 0. This shall automatically switch the identified LSS slave selectively into configuration mode. Subsequently, it may be configured by the LSS master.

NOTE The service LSS switch state selective needs not be used.

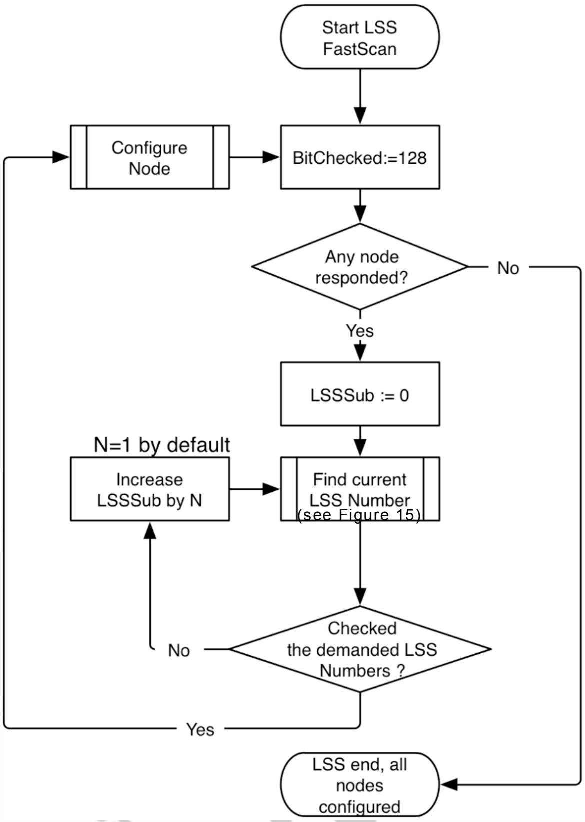


Figure 16 – LSS Fastscan overview

The whole scan process shall be repeated until no more unconfigured LSS Fastscan devices are detected.

NOTE Only un-configured LSS slaves with node ID equal to FF_h shall participate to LSS Fastscan

The following message sequence charts (Figure 17, Figure 18 and Figure 19) depict the LSS Fastscan service.

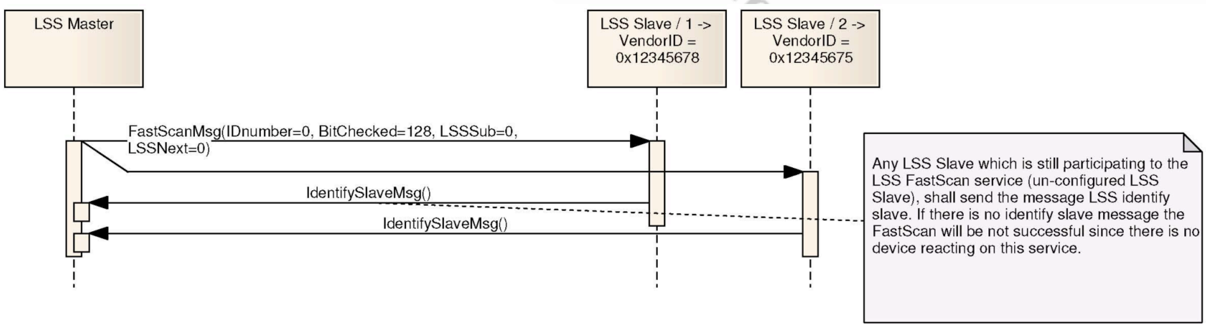


Figure 17 – Example of starting sequence of LSS Fastscan

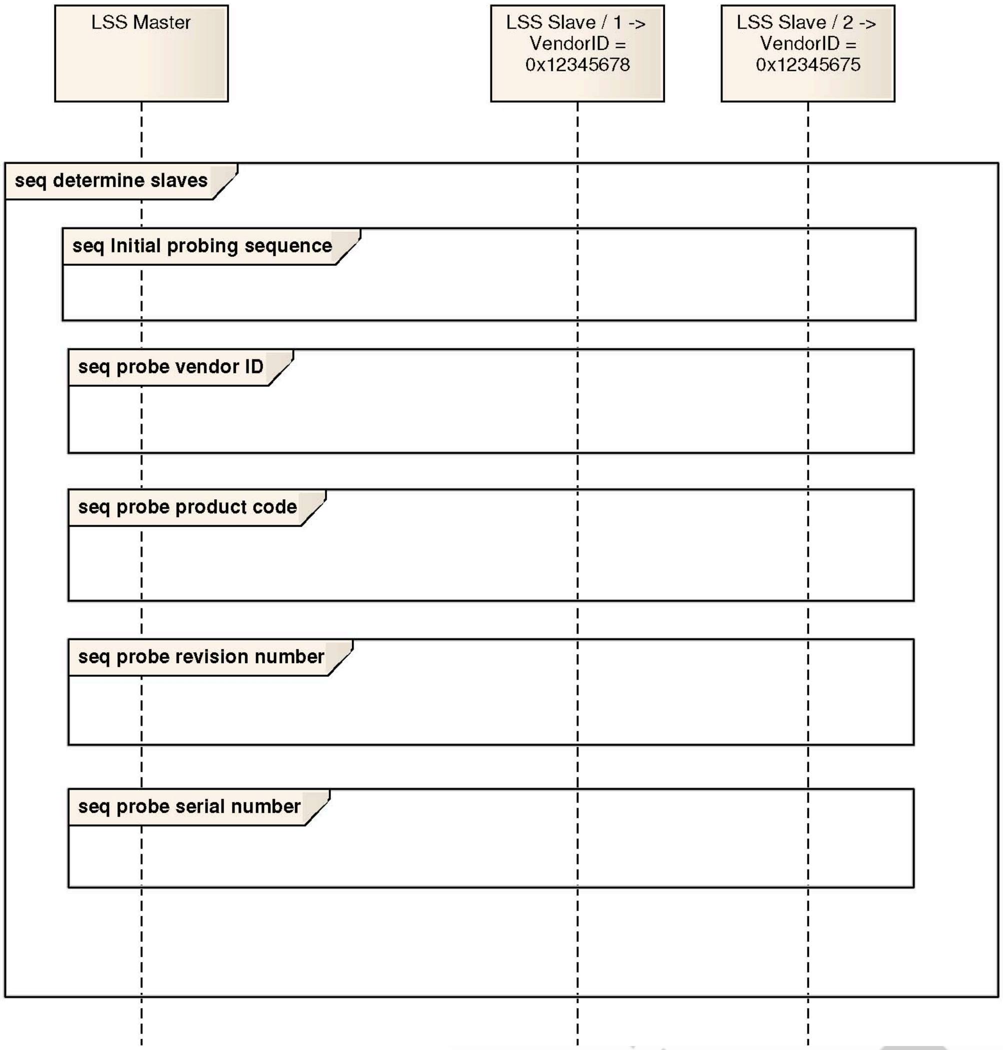


Figure 18 – Overview on the activities to identify exactly one unconfigured LSS slave in the network

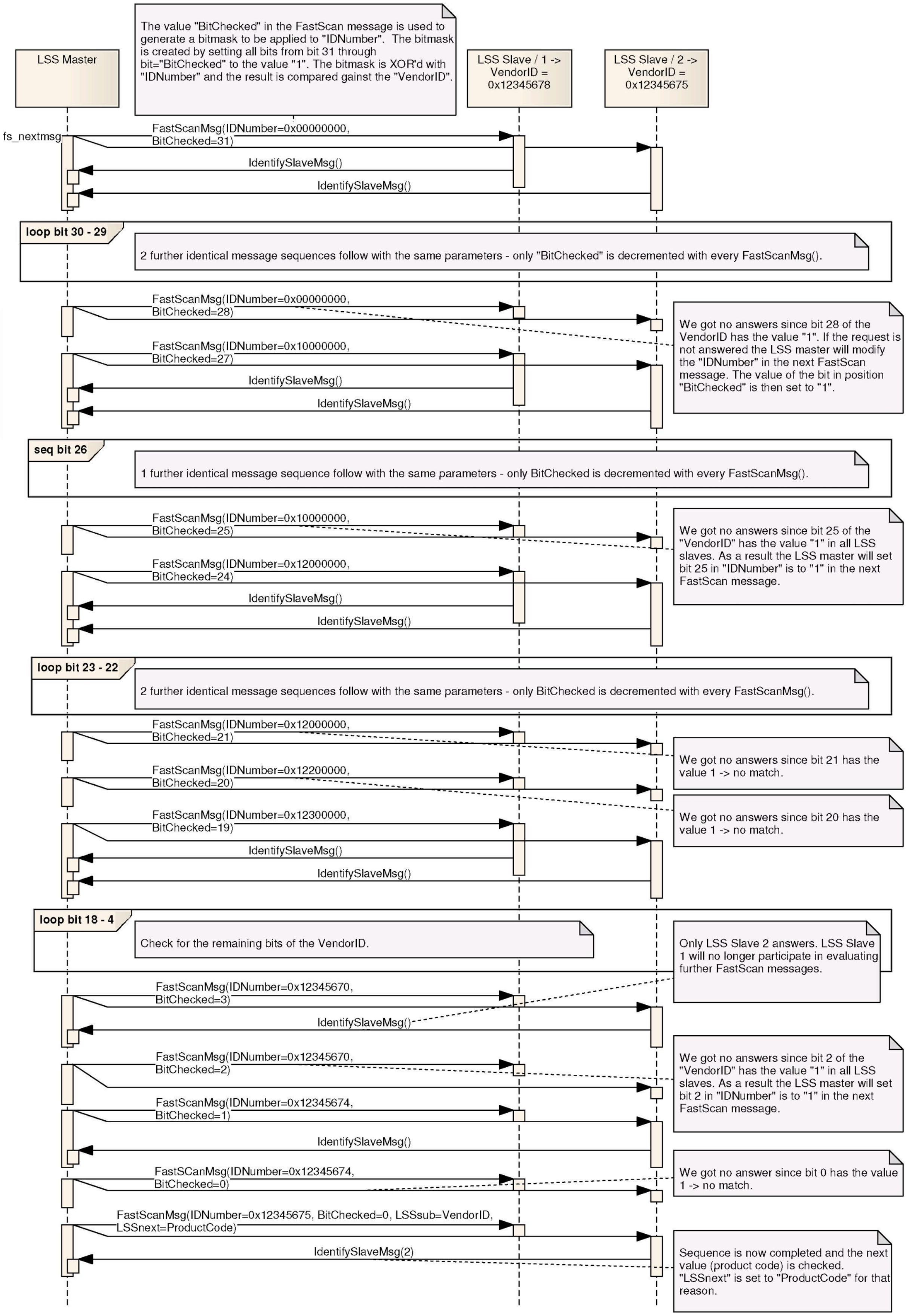
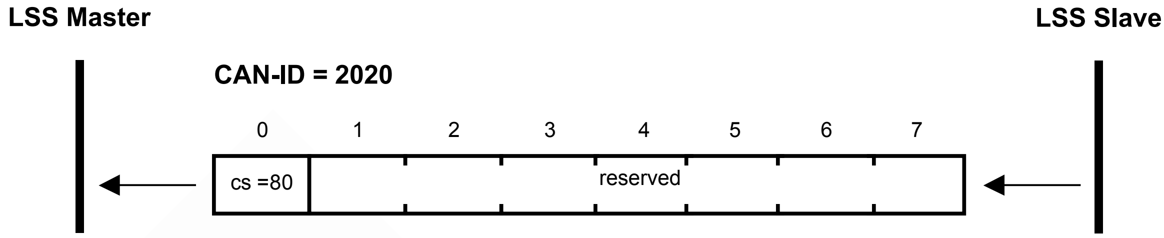


Figure 19 – Example on identifying the vendor ID

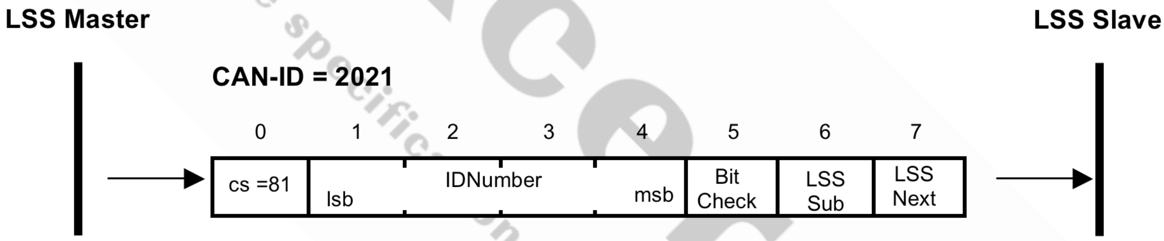


- cs:
  LSS command specifier
  80 for LSS identify non-configured slave
- reserved:
  All bytes shall be set to '0'.

Figure 34 – LSS identify non-configured slave protocol

5.7.5 LSS Fastscan protocol

The protocol defined in Figure 35 shall be used to implement the LSS Network Identification service.



- cs:
  LSS command specifier
  81 for LSS Fastscan
- IDNumber, BitChecked, LSSSub, LSSNext:
  see 4.10.2

Figure 35 – LSS Fastscan protocol

A.1 Implementation and application hints

A.1.1 Invalid LSS messages

An LSS message is invalid if it has a CAN-ID that is used by the LSS protocols, but contains invalid parameter values according to the LSS protocols. This may be caused by errors in the data link layer (see /ISO11898-1/) or implementation errors. An invalid LSS message shall be handled locally in the LSS implementation. As far as the LSS protocol is concerned, an invalid LSS message shall be ignored.

A.1.2 Timeouts

Since LSS messages may be ignored or not served, the response of a confirmed LSS service may never arrive. To resolve this situation, an LSS master device shall, after a certain amount of time, a timeout is indicated locally. A timeout is not a confirmation of the LSS service. A timeout indicates that the service has not completed yet. Timeout values are considered to be implementation-specific. However, it is recommended that an LSS master implementation shall provide facilities to adjust these timeout values to the requirements of the application.

A.1.3 Usage of LSS services

A.1.3.1 Transition to LSS configuration state

A.1.3.1.1 LSS address known

If the LSS master device likes to switch a specific LSS slave device into LSS configuration state, the LSS master device requests a switch mode selective service with the known LSS address. The LSS address (vendor-ID, product-code, revision number, and serial number) is defined in clause 4.2.

If there is only one LSS slave device and one LSS master device in the network, the LSS master device may alternatively request the switch mode global service with the parameter 'switch configuration state'.

A.1.3.1.2 Vendor-ID and product-code known

The LSS master device may request the LSS identify remote slave service. All LSS slave devices that match the requested LSS address range respond by means of the LSS identify slave service. The LSS master device narrows the LSS address range to a single LSS address. If there is still an LSS slave device responding with an LSS identify slave service, the LSS slave device is identified. Now the LSS address is known (see clause A.1.3.1.1).

A.1.3.1.3 LSS address unknown in a direct connection

There shall be only one LSS slave device in the network. The LSS master device may request the LSS switch state global command with the parameter 'configuration_switch'.

A.1.3.2 Setting the node-ID

A.1.3.2.1 Device with node-ID of FF_h

If an LSS slave device has no valid node-ID, it transits automatically on entry of the NMT sub-state reset communication into the LSS waiting state. The LSS master device switches the LSS slave device into LSS configuration state (see clause A.1.3.1) and requests the configure node-ID service. Subsequently, the LSS master device may switch the LSS slave device back into LSS waiting state. If the node-ID is valid, the LSS slave device automatically returns into NMT communication reset state followed by an automatic transition to NMT pre-operational state.

If there is no valid node-ID stored in with the LSS store configuration service, a subsequent command NMT reset node will cause a re-entering of the LSS FSA with the invalid node-ID FF_h. Issuing a NMT reset communication service causes just a re-use of the last valid distributed node-ID.

A.1.3.2.2 Configuration of node-ID

The LSS master application or the system designer is responsible for the unique node-ID assignment and distribution to the networked LSS slave devices. If the LSS master device likes to configure an LSS slave device's node-ID, the LSS slave device shall be the one and only device in LSS configuration state. The LSS master device may then request the LSS configure node-ID service. The LSS master device should request the LSS store configuration service. Then the LSS master device may request the LSS switch global service with the parameter 'switch waiting state'. The new node-ID only becomes active on the next entry into the NMT reset communication sub-state. This may be achieved by means of NMT reset commands using the initial ("old") node-ID.

A.1.3.3 Inquiring LSS address and node-ID

The inquiry services may only be requested by the LSS master device, if a single LSS slave device is in LSS configuration state.

A.1.3.4 Changing the bit timing parameter

Changing the bit timing parameter is only possible if all devices support the corresponding LSS services or they provide automatic bit rate detection capability. All LSS slave devices shall be in NMT stopped state of in NMT pre-operational state, but no LSS slave device shall send any CAN message. This means the heartbeat shall be disabled when the activate bit timing service is performed. Then the LSS master device shall request the following sequence of services individually for all LSS slave devices:

- Switch state selective service (1)
- Configure bit timing service (2)
- Store configuration service (3)
- Switch state global service with parameter 'switch waiting state' (4)

If one of the services 2 or 3 is not completed successfully, the LSS master device should change all settings to the bit timing currently used.

If all LSS slave devices are configured correctly, the LSS master device may request the LSS switch state global service with parameter 'configuration_switch', and the activate bit timing parameters service. After successful activation of the bit-timing parameters, the LSS master device transits the LSS slave device into LSS waiting state.

A.1.3.5 Storing configured parameters

If the LSS master device requests an LSS slave device to store the configured parameters, it transits this and only this device into LSS configuration state, and subsequently requests the store configuration service. After successful storage, the LSS master device transits the LSS slave device into LSS waiting state.

A.1.4 Special hints for LSS Fastscan

A.1.4.1 The bitchecked variable

Example: The current LSS number, which is being checked is the Vendor ID. BitChecked has a value of 16. Then all LSS Fastscan Slaves respond whose own Vendor ID bits 16 to 31 match with bits 16 to 31 of IDNumber.

A.1.4.2 Timeouts

This document does not define Timeouts.

It may happen, that the response from a slave arrives at the master much too late. If the LSS master has already requested the next round, he gets an acknowledge from a slave, whose LSSNumber perhaps does not match any more.

To avoid this behavior one should consider the following implementation hints:

The duration an LSS master is waiting to receive a response from an LSS slave is shall be called $T_{timeout}$. An LSS slave shall grant a response time within $T_{response}$. Considering this condition for the slaves, after $T_{response}$ all slaves are supposed to have answered. Additionally, it shall apply $T_{response} < T_{timeout}$.

NOTE As the LSS-messages have low priority PDOs and EMCYs may delay the transmission of LSS-messages.

A.1.4.3 Optimizations

An application profile may define to use a sub-set of the LSS identification only. In that case the scanned range of bits may be reduced.

This is demonstrated in the following example.

In this example the existing 128-Bit LSS address is reduced to 80 bits by the following definition (Unused bits of the 128-Bit full LSS address are set to zero in order to be backwards compatible with other LSS implementations):

- CANopen vendor ID, 32-bit object dictionary entry at index 1018_h sub-index 01_h
- Product code, 16-bit at index 1018_h sub-index 02_h, lower 16 bit only. Upper 16 bit always 0000_h.
- Revision number, at index 1018_h sub-index 03_h always 0000 0000_h
- Serial number, 32-bit at index 1018_h sub-index 04_h

At the end of the vendor ID scanning the LSS Master starts the product code scan with the request of IDNumber 0000 0000_h, BitChecked 16, LSSSub 1 and LSSNext 1. This request is only responded to by LSS slaves that have all upper 16 bit of the product code set to zero.

At the end of the product code scan a similar shortcut request is made to skip the scan of the revision number. The LSS Master request for that uses IDNumber 0000 0000_h, BitChecked 0, LSSSub 2 and LSSNext 3. LSS slaves, which are still actively participating in the scan will only continue if their entire revision number is zero.

In comparison to the full 128 Bit scan this optimized scan requires 48+1 less LSS master requests during a scan cycle.

A.1.4.4 Pseudo code example

This pseudo code as shown in Table 8 demonstrates a scan, which is processed by an LSS-master.

`SendLSSIdentify()` sends the LSS Fastscan message. Return value is true, if at least one slave responds before timeout.

`LSS_ID` is an array containing the four unsigned32 values of the LSS address.

Table 8 – Pseudo code for an LSS master

```
BEGIN
  Init all variables with zero
  BitChecked = 80h
  IF SendLSSIdentify(LSS_ID[0],BitChecked,LSSSub,LSSNext)
    WHILE LSSSub < 4
      LSS_ID[LSSSub] = 0
      BitChecked = 32
      WHILE (BitChecked > 0)
        Decrement BitChecked
        IF BitChecked == 0 AND LSSNext < 4
          LSSNext = LSSSub + 1;
        ENDIF
        IF NOT SendLSSIdentify(LSS_ID[LSSSub],BitChecked,LSSSub,LSSNext)
          // No response, set this bit
          Set bit BitChecked in LSS_ID[LSSSub]
        ENDIF
      ENDWHILE
      // repeat request with complete LSS_ID[LSSSub]
        IF NOT SendLSSIdentify(LSS_ID[LSSSub],BitChecked,LSSSub,LSSNext)
          RETURN FALSE
        ENDIF
      ENDIF
      // Now the next 32 bits will be scanned
      Increment LSSSub
    ENDWHILE
    // Now LSS_ID contains the entire 128 bits scanned
    RETURN TRUE and LSS_ID
  ENDIF
  RETURN FALSE
END
```