INTERNATIONAL STANDARD

ISO 11783-14

First edition
2013-09-15

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 14: Sequence control

Tracteurs et matériels agricoles et forestiers — Réseaux de commande et de communication de données en série —

Partie 14: Contrôle de séquence

Reference number
ISO 11783-14:2013(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2. www.iso.org/directives

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received. www.iso.org/patents

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

ISO 11783 consists of the following parts, under the general title *Tractors and machinery for agriculture and forestry — Serial control and communications data network*:

— *Part 1: General standard for mobile data communication*

— *Part 2: Physical layer*

— *Part 3: Data link layer*

— *Part 4: Network layer*

— *Part 5: Network management*

— *Part 6: Virtual terminal*

— *Part 7: Implement messages application layer*

— *Part 8: Power train messages*

— *Part 9: Tractor ECU*

— *Part 10: Task controller and management information system data interchange*

— *Part 11: Mobile data element dictionary*

— *Part 12: Diagnostics services*

— *Part 13: File server*

— *Part 14: Sequence control*

Introduction

ISO 11783 specifies a communications system for agricultural equipment based on the ISO 11898-1 protocol. SAE J1939 [1] documents, on which parts of ISO 11783 are based, were developed jointly for use in truck and bus applications and for construction and agriculture applications. Joint documents were completed to allow electronic units that meet the truck and bus SAE J1939 specifications to be used by agricultural and forestry equipment with minimal changes.

General information on ISO 11783 is to be found in ISO 11783-1. The purpose of ISO 11783 is to provide an open, interconnected system for on-board electronic systems. It is intended to enable electronic control units (ECUs) to communicate with each other, providing a standardized system.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this part of ISO 11783 may involve the use of a patent concerning the controller area network (CAN) protocol referred to throughout the document.

ISO takes no position concerning the evidence, validity and scope of this patent.

The holder of this patent has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Robert Bosch GmbH

Wernerstrasse 51

Postfach 30 02 20

D-70442 Stuttgart-Feuerbach

Germany

Attention is drawn to the possibility that some of the elements of this part of ISO 11783 may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

Tractors and machinery for agriculture and forestry — Serial control and communications data network —

Part 14: Sequence control

1 Scope

ISO 11783 as a whole specifies a serial data network for control and communications on forestry or agricultural tractors, mounted, semi-mounted, towed or self propelled implements. Its purpose is to standardize the method and format of transfer of data between sensors, actuators, control elements, information storage and display units whether mounted, or part of the tractor or any implement.

This part of ISO 11783 specifies a sequence control system, such as a headland management system, which includes tractor and implement functions in one system. The system allows recording of multiple sequences of operator-activated functions from a tractor or any other control function on the ISO 11783 network, and replaying them on operator command.

2 Normative references

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

ISO 11783-1, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 1: General standard for mobile data communication*

ISO 11783-3, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 3: Data link layer*

ISO 11783-5, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 5: Network management*

ISO 11783-6, *Tractors and machinery for agriculture and forestry — Serial control and communications data network — Part 6: Virtual terminal*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11783-1 and ISO 11783-3 and the following apply.

3.1
abort
halt the sequence play back and immediately stop all motion initiated during the sequence

EXAMPLE The hitch stops at the current position.

3.2
active sequence control master
sequence control master selected by the operator to control the sequence control system

Note 1 to entry: Only one sequence control master (SCM) is permitted to be active at any time.

3.3
cancel
prematurely stop sequence recording

3.4
client command
action that is communicated by the sequence control client during recording and executed by the sequence control client if received during the play back of a sequence

EXAMPLE Start power take-off (PTO), stop PTO, open baler rear gate, close baler rear gate, sprayer master valve on, sprayer master valve off.

3.5
client function
mechanic, hydraulic or electric function of the tractor or implement that may be engaged/disengaged, started/stopped or changed by an operator action and may accept commands from the sequence control master during sequence play back

EXAMPLE PTO, hitch, auxiliary valves, baler rear gate, sprayer master valve.

3.6
disabled sequence control client
sequence control client that does not respond to state changes of the active sequence control master and therefore does not participate in sequence play back and recording

3.7
enabled sequence control client
sequence control client that has been selected by the active sequence control master to participate in sequence play back and recording

3.8
graphical representation object
object of the SCCWS's (sequence control client working set) object pool that is suitable to be referenced by SCD (sequence control data definition) objects in cases where the SCM (sequence control master) is required to display graphical information about the SCC (sequence control client), a client function or a function state

EXAMPLE A Picture Graphic object, a Rectangle object or a Polygon object would be suitable to be referenced by the Graphical representation attribute of an SCD state object.

3.9
inactive sequence control master
sequence control master connected to the system but not controlling the sequence control system

3.10
safe state
operating mode of a system with an acceptable level of risk for operator or bystander even when the control system fails or partly fails

3.11
sequence control client
SCC
control functions connected to the ISO 11783 network that provide client functions which may be used for sequence control

Note 1 to entry: The operator is able to activate these client functions manually by inputs such as the user interface soft keys, physical buttons or auxiliary control.

Note 2 to entry: The tractor may identify itself as SCC with its client functions, SCM or as both.

3.12
sequence control client working set
SCCWS
working set as defined in ISO 11783-6, where either the working set master and/or one or more working set members act as SCC

3.13
sequence control data definition
SCD
set of objects describing the recordable functions transferred by sequence control clients to the sequence control master including the function IDs, state IDs, preferred trigger method, icons for the visualization and function name for each of the functions supported

Note 1 to entry: Details are defined in Annex A.

3.14
sequence control master
SCM
controller of the sequence control system that initiates the recording and play back phase of sequences and stores the client commands received from the sequence control clients during recording with one of its provided trigger points

3.15
sequence control master object pool
SCMOP
VT (virtual terminal) object pool sent by the SCM providing the user interface for the interaction of the operator with the SCM

Note 1 to entry: VT and object pools are discussed in ISO 11783-6

3.16
sequence control client object pool
SCCOP
VT object pool sent by the SCCWS containing at least all the objects required by the SCM to display information from the SCC's SCD properly on the VT

Note 1 to entry: These objects are usually referenced in the SCMOP.

Note 2 to entry: VT and object pools are discussed in ISO 11783-6.

3.17
sequence control sequence
SCS
set of functions and/or actions performed by one or more sequence control clients in the system during the recording phase, including the related trigger information for the activation of the functions

3.18
sequence control system
SC
system with an SCM and one or more SCC performing the sequence control functionality as described in this part of ISO 11783

3.19
sequence control trigger
method to activate SCC functions in a sequence control sequence

EXAMPLE 1 For a time trigger, the SCM uses the time between two function activations of the sequence, independent of the distance driven as the method of a function's activation.

EXAMPLE 2 For a distance trigger, the SCM uses the distance driven between the activation of two functions as the method of a function's activation.

3.20
sequence number
number which uniquely identifies the sequence(s) in the SC communication, allowing the use of multiple sequences within one system

EXAMPLE The SCM supports one sequence for entering the headland (sequence number = 1) and one for leaving the headland (sequence number = 2).

3.21
textual representation object
object of the SCCWS's object pool that is suitable to be referenced by SCD objects in cases where the SCM needs to display textual information about the SCC, a client function or a function state

EXAMPLE An Output String object would be suitable to be referenced by the Designator attribute of an SCD.

3.22
transaction number
TAN
method to synchronize the command and response messages

Note 1 to entry: See 4.6 for further information.

4 Technical requirements

4.1 General

This part of ISO 11783 specifies a control system that enables machines to perform automatic functions such as headland turns or water way traverse control in the field. It allows the operator to automate a series of tractor and implement functions being performed each time while reaching or leaving for instance the headland; functions normally activated manually by the operator.

The sequence control system consists of a sequence control master and a number of sequence control clients communicating on the ISO 11783 network. This part of ISO 11783 defines the data formats, requirements and services used for the communication between the CFs participating in the sequence control system.

After an operator starts the recording phase of the SC, the operator activates all the client functions or actions to be automated in one sequence by using each SCC operator interface (recording of normal manual operation of the functions). The SCM receives on the ISO 11783 network the information about the activated client function or action from the CFs containing the client function (SCC) via client commands. The SCM then stores this activated client function information together with the sequence control trigger information assigned to be used for this function by the SCM. These stored sequences may then be replayed multiple times (e.g. on operator command). The SCM transmits the client command to the SCCs when the referring trigger point is reached after start of sequence replay. On receipt of an associated client command, the SCC performs the commanded client function or actions in the same way as if it were manually activated by the operator via the client's proprietary input. The SCCs are independent from each other; therefore, no direct communication between them is required.

Each sequence has a unique number within the SC; however, it is up to the SCM implementation to determine how operators can identify the individual sequences in case of support for multiple ones (e.g. associate proprietary names to each individual sequence). The SCM may group and save multiple sequences (e.g. one for approaching and one for leaving the headland) under a unique descriptor such as "Seeding". The SCM may indicate the recorded sequence(s) on the user interface with icons and/or text designators provided by the SCC(s) representing each function or action. Depending on the SCM implementation, it is possible for the operator to have the ability to manually define sequences or edit the sequences by changing the timing (trigger point) between the client functions or other function parameters. It is possible for the SCM to provide the ability for the operator to store and reload sequences for later usage using the same machine configuration (e.g. particular tractor-implement combination).

The SCM shall provide a means for setup of sequences (RECORD or EDIT) and to activate the play back of a selected sequence (PLAY BACK); the SCM may provide a means to the operator for SCM configuration (CONFIG).

The SCM may also provide a means to display the identified SCCs and their automated/recordable functions, actions and preferred trigger options for operator review. The SCM may provide a means to the operator to enable or disable certain SCCs (see Figure 9) available on the network to reduce the complexity of the system configuration during recording and edit, and also for reducing the busload. The SCM shall only include the enabled SCCs in the sequence management.

The support of a sequence control system may be implemented by any CF connected to the ISO 11783 network.

4.2 Sequence control user interface

The SCM shall provide a user interface for the purpose of warnings and needed operator interaction by connecting to the VT and uploading its main screen layout (object pool) accordingly (further details on VT and object pools are defined in ISO 11783-6).

To allow optional SCM features, like viewing sequence details, the SCCs have to provide the graphical and textual representation of their sequence control functions, etc. to the SCM. The availability of both a textual designator and a graphic for each SCD object gives the SCM implementation the freedom to be represented with text only, graphic only or a combination of both for displaying SCC objects.

This communication concept uses the External Object Pointer introduced with the VT version 5 (further details are defined in ISO 11783-6) to avoid the requirement that the SCM has to handle large graphic objects or language updates for the individual SCCs. This requires that the individual SCCs or their working set master load their graphic and text objects as part of their SCCOP into the same VT as used by the SCM and provide reference information only in their SCD to the SCM. The SCM adds External Object Pointer references in its data masks at the locations where such objects may be shown. These reference objects point to the objects in the referenced SCCOP to allow the VT to display the desired information in the screen of the SCM.

The SCM and the working set master of the SCCWSs shall connect and upload their object pool to the VT with the function instance 0 to avoid additional synchronization overhead on the VT of choice between the SCM and its SCCs. However, this can mean that an SCCWS has to maintain two VT connections in parallel in case the main operator interaction for the SCCWS is handled by an additional VT (function instance > 0) (see Figure 1). This concept of using the VT with the function instance 0 follows the Auxiliary Control definitions in ISO 11783-6 and allows the SCCWS to merge and share objects between both functionalities in one SCCOP. It also enables the SCC to use different languages and character sets for its representation by using the capabilities provided by ISO 11783-6 (e.g. use of Unicode in the textual representation objects) directly between SCC and VT without SCM interaction.

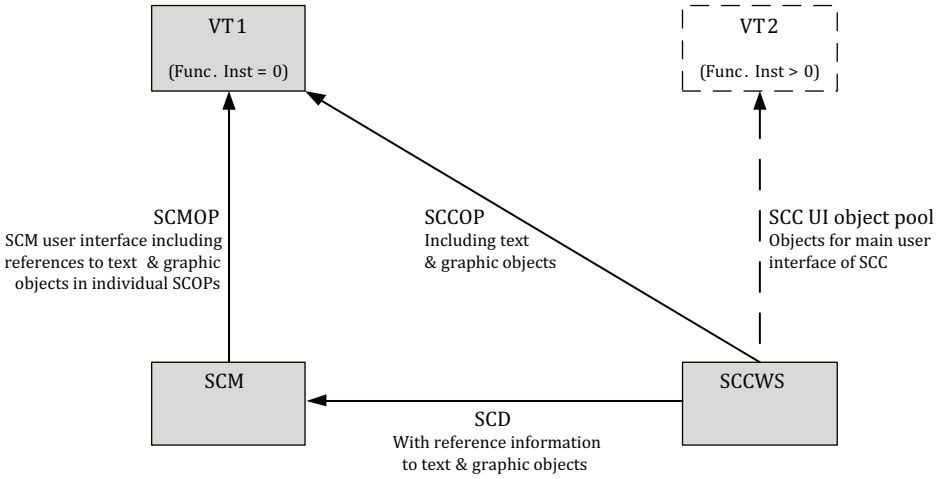


Figure 1 — User interface communication concept

4.2.1 User interface initialization

During initialization, the SCM uploads a new or reactivates its previous loaded object pool to the VT.

The SCM may use external object pointer references to the text and graphic objects in the SCCOP of its clients to display for instance sequence details (details on the external reference objects are defined in ISO 11783-6).

The external object pointer concept is NAME-based to ensure appropriate identification of the SCCOP and SCCWS in the object references. The SCMOP includes External Reference NAME Objects (details are defined in ISO 11783-6) to avoid the frequent communication of the complete NAME with each change of a reference to an object in an SCCOP. The communication of the SCM to the VT includes the two byte object ID of the External Reference NAME Object instead (see keys 22 and 25 in Figure 3).

The SCM has no means to check for the object type in the individual object references provided in the SCD. It knows the object IDs only. Therefore, it is the responsibility of the SCC to ensure the use of appropriate objects for the graphical and textual representation. SCMs shall provide provisions to deal with object references rejected by the VT.

NOTE 1 Despite the fact that multiple SCCs may be included in one SC and each one provides a number of references to different graphics and textual objects in a structure as shown in Figure A.2, the following example focuses only on an SC with one SCC and two referenced objects to explain the concept applying to all of those objects involved in a true system.

During the initial SCM object pool upload to the VT, the External Reference NAME Objects shall be disabled and the External Object Pointer object attributes shall be set to the NULL Object Id (see ISO 11783-6 for details).

Figure 2 shows an example of the object hierarchy in the Object Pools loaded into the VT for a simple SCC and an SCM connected to the VT. Key 31 represents the SCCOP including one text object (key 33) and one graphic object (key 34) in the VT's memory. The SCC also loaded its SCD (key 36) into the SCM memory; while the SCM transferred its default object pool into the VT memory (key 21). In this example, the SCM includes two external reference pointers in its screen layout (one graphic with key 23 and one text with key 24), both of which are disabled after start-up.

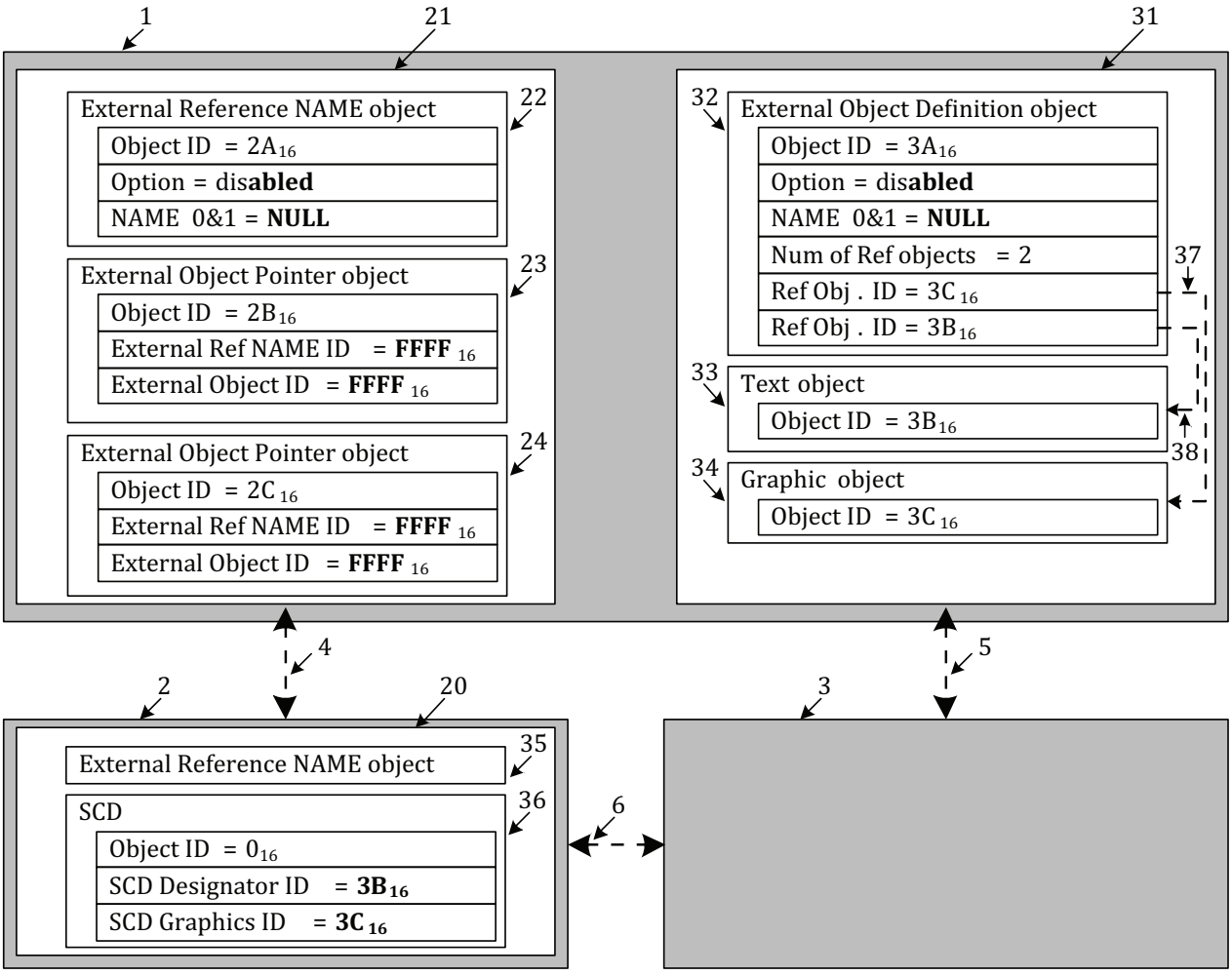


Figure 2 — Initial status of external object pointer — Example

NOTE 2 The keys for Figures 2 and 3 are listed under Figure 3.

The SCCWS shall validate the NAME of the active SCM followed by an update of the NAME field in its External Object Definition object (key 32) via Change Attribute command. The activation of the External Object Definition object (Option = enabled; key 32) allows the VT to establish references from the SCMOP to the defined referenced objects (key 37 and 38) in the SCCOP (key 31). Key 39 in Figure 2 represents the established reference to the SCMOP.

The SCM shall also validate the NAME of the referenced SCCs followed by an update of the related NAME field in its External Reference NAME object. The activation of the External Reference NAME object via Change Attribute command established the general connection to the SCCOP. Key 26 in Figure 2 represents the established reference to the SCCOP.

The SCM configures its External Object Pointer objects (key 23 and 24) with the reference data received in the SCD (key 36). Usually, two Change Attribute commands are needed per External Object Pointer object; one to set the ID of the External Reference NAME object (key 25) to point to the correct WSOP and the second to set the External Object ID (key 27 and 28) of the desired object.

Figure 2 shows the established references to a graphic object (e.g. a working set designator icon) and a text object (e.g. a working set designator) in the SCCOP of the connected SCC. The VT finds the referenced SCCOP for each External Object Pointer object in the screen layout of the SCM via keys 25 and 26 while key 39 allows a check if the SCC permitted the SCM to reference its objects. With an established reference to the SCCOP, the External Object IDs are unambiguously connected to the referenced objects. Keys 27 and 37 and 28 and 38 represent this connection, while going through the External Object Definition

object (key 32) reflects the additional checking for access permission; only objects listed in the External Object Definition object may be referenced.

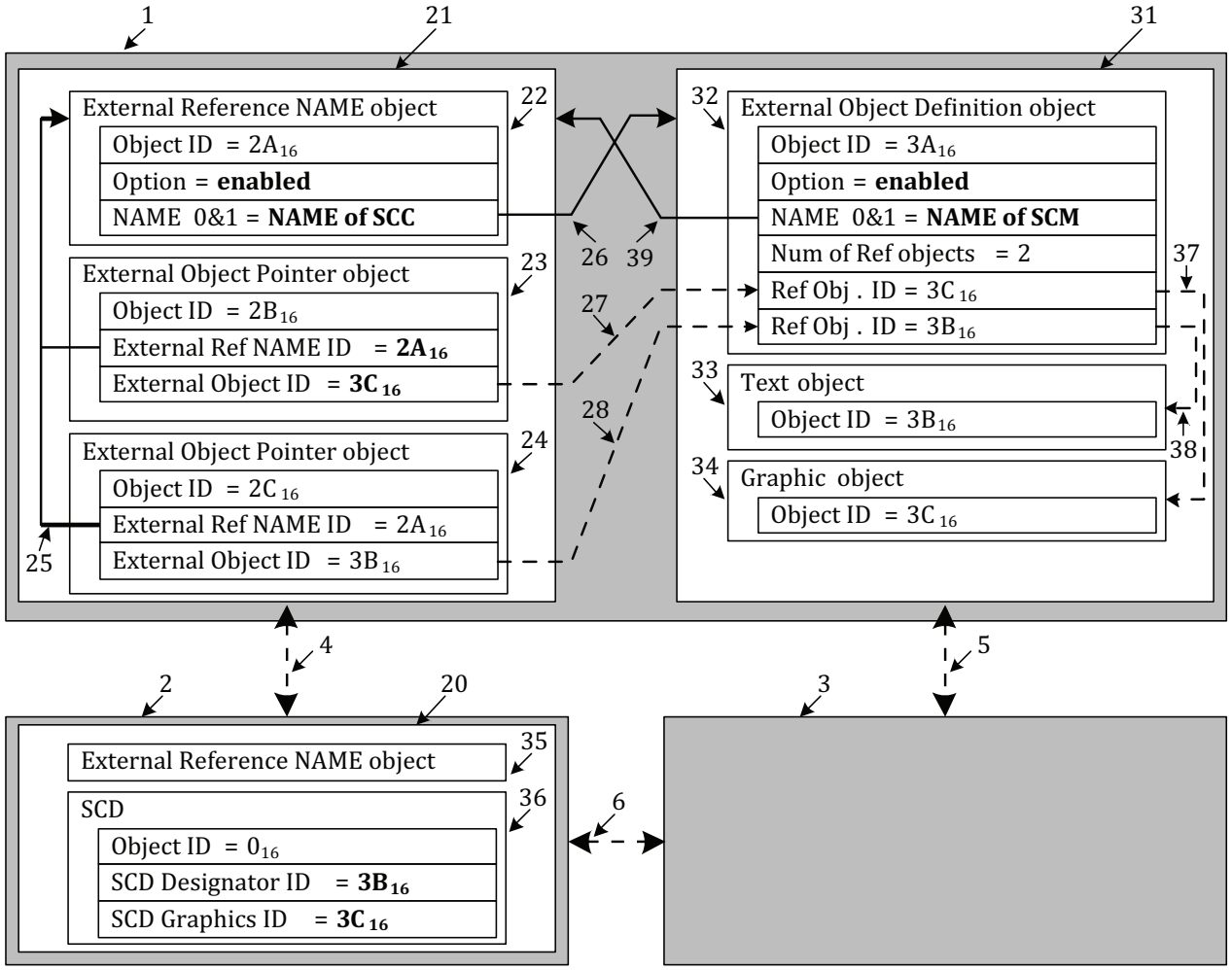


Figure 3 — Status of External Object Pointers after initialization — Example

Keys for Figure 2 and Figure 3:

1 VT Object Pool volatile memory

2 SCM

20 SCM volatile memory

21 SCM Object Pool

22 External Reference NAME object including NAME of referenced SCCWS;
one object per referenced SCCWS needed

23 External Object Pointer object to reference a graphic object (placeholder in the data mask)

24 External Object Pointer object to reference a text object (placeholder in the data mask)

3 SCC

31 SCCWS Object Pool

32 External Object Definition object defining the objects which may be referenced by the SCM specified in the NAME 0&1 attributes

33 Working Set Designator of SCCWS

34 Working Set Icon of SCCWS

35 NAME of SCC received through Address Arbitration

36 SCD of connected SCCWS

Keys for Figure 2 and Figure 3:

25 Virtual reference to External Reference NAME object to identify referenced Object Pool to VT

26 Virtual reference to the SCCOP established by the enabled External Object Pointer object in the SCMOP

27 Virtual reference to e.g. the Working Set Icon of SCCWS in SCCOP established by the enabled External Reference NAME and External Object Pointer

28 Virtual reference to e.g. the working set Designator of SCCWS in SCCOP established by the enabled External Reference NAME and External Object Pointer

37 Virtual reference to Working Set Icon that may be referenced by the SCM

38 Virtual reference to Working Set Designator that may be referenced by the SCM

39 Virtual reference to the SCMOP established by the enabled External Object Pointer object in the SCCOP

4 SCM-VT communication including SCMOP upload and attribute updates

5 SCCWS-VT communication including SCCOP upload and attribute update

6 SCC-SCM communication including addresses arbitration for NAME identification and SCD upload to SCM

4.3 Working sets with master/member configuration

In system configurations with a working set master and one or more working set members (as defined in ISO 11783-6), the responsibilities for the connections from the SCC to the VT and from the SCC to the SCM have to be considered separately.

4.3.1 SCC as working set master

In cases where an SCC is the working set master, the SCC is responsible for managing the VT connection as well as the connection to the SCM.

The SCCWS Master NAME field in the SCD basic object (defined in A.3) shall be set to the NAME of the SCC itself.

4.3.2 SCC as working set member

In cases where the SCC is a working set member, the SCC's working set master is responsible for managing the VT connection (its object pool is the SCCOP including graphical and textual objects referenced by the SCC in its SCD). The SCC is responsible for the connection to the SCM (including the SCD upload).

The SCC needs to know the object IDs of graphical representation objects and textual representation objects from the working set master's object pool to reference them in its SCD. The communication of this information is not subject of this part of ISO 11783.

The SCCWS Master NAME field in the SCD basic object (defined in A.3) shall be set to the NAME of the working set master.

4.4 Sequence management functionality

Sequence management functionality is described as logic entity of software residing in any CF connected to the ISO 11783 network.

The SCM may limit the access and sequence control data definition uploads of SCCs due to memory limitations or other restrictions. The SCM shall hold only one SCD per SCC in non-volatile memory.

Each SCM stores and manages its recorded sequences autonomously. This part of ISO 11783 does not define a standardized means for transferring sequences from one SCM to other SCMs or to a desktop computer.

4.4.1 Sequence control system operating states

The following clauses define the states reported by the active SCM and SCCs and the transitions between them. Figure 3 and Figure 4 illustrate the transitions between the different states of the SCM and SCC.

4.4.1.1 SCM operation

SCM and SCS states:

Inactive: This SCM state indicates that the SCM is not participating in the overall SC communication. This is the default state of an SCM after start-up.

Ready: ready for “Recording” or “Play Back”

Recording: actively recording a sequence.

Recording Completion: state entered if at least one client function could not finish its execution before the operator stopped the “Recording”; reported starts of additional functions are not allowed in this state and shall be ignored by the SCM.

Play Back: actively executing a sequence

Abort: state entered and transmitted to all CFs to halt a “Play Back” phase (see definitions in 4.4.3).

The states “Ready”, “Recording”, “Recording Completion”, “Play Back” and “Abort” are considered the active states, where the active SCM controls the SC communication.

The relationships between the different states are illustrated in Figure 4.

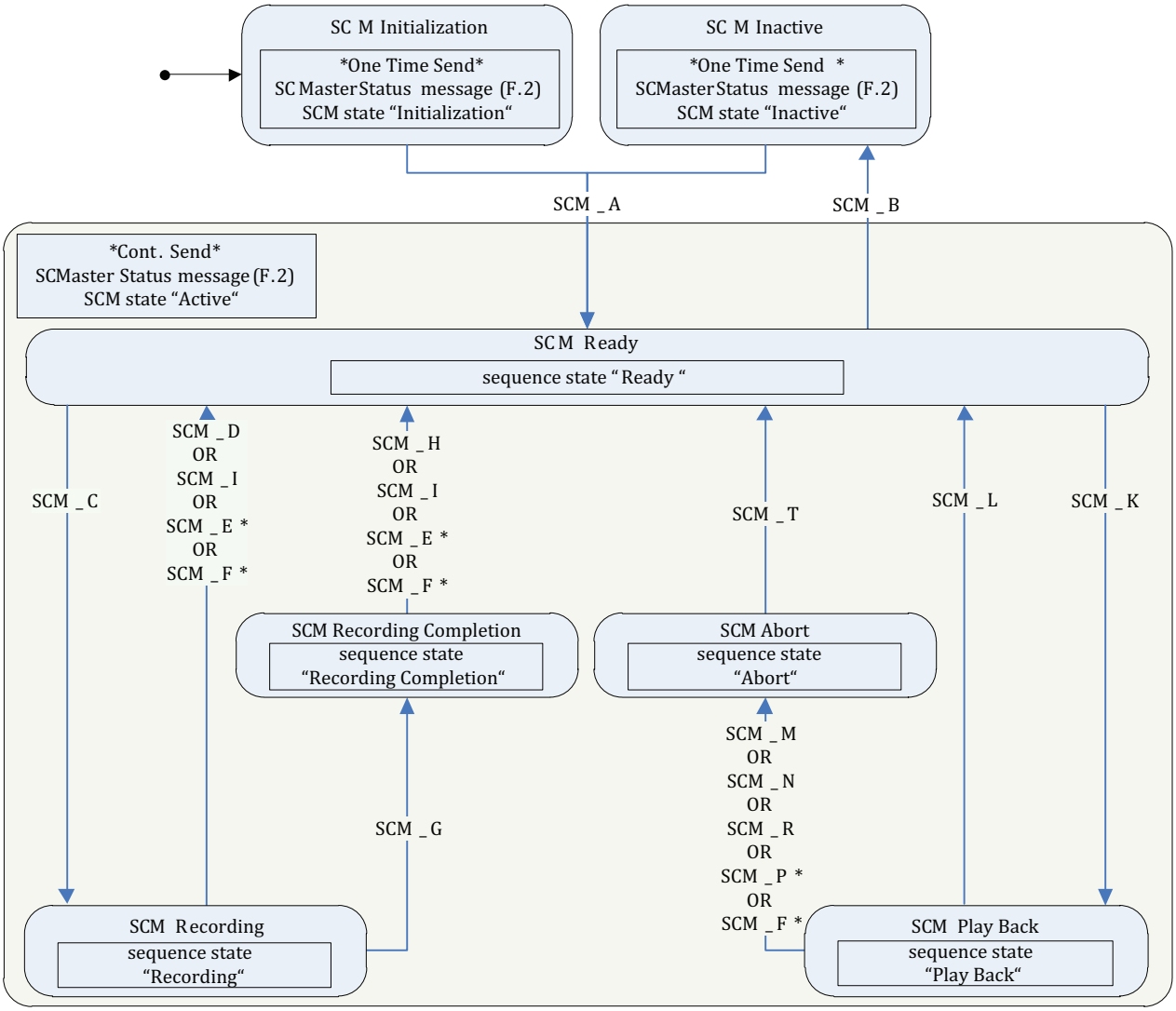


Figure 4 — SCM state diagram

Conditions for state transitions:			
SCM_A:	operator Input: “Master ON	SCM_K:	“Start Play Back” (e.g. through operator input)
SCM_B:	operator Input: “Master OFF	SCM_L:	Normal sequence end
SCM_C:	operator input: “Start Recording”	SCM_M:	any SCC sent SCClientAbort message
SCM_D:	operator input: “Stop Recording” AND all recorded client functions sent function execution state as [10 (function execution completed) OR 11 (error during execution)]	SCM_N	operator input: “Abort Play Back”
SCM_E *:	operator input: “Cancel Record	SCM_P *:	E.7 SCClientExecutionStatus: function execution state == 11 (error during execution)
SCM_F *:	E.4 SCExecutionIndication: function execution state == 11 (error during execution)	SCM_R *:	internal SCM abort condition (e.g. Timeout of any SCC message or SCC unexpectedly falls back to disabled)
SCM_G:	operator input: “Stop Recording” AND at least one recorded client function did not send function execution state as [10 (function execution completed) OR 11 (error during execution)]	SCM_T:	All enabled SCCs reflect “Abort” state to SCM
SCM_H:	all included client functions sent function execution state as [10 (function execution completed) OR 11 (error during execution)]		
SCM_I:	internal SCM cancel condition (e.g. Timeout of any SCC message or SCC unexpectedly falls back to disabled)		
* = optional requirement			

4.4.1.2 SCC operation

SCC states:

Disabled: The SCC is either not ready to execute received client commands based on internal conditions or settings, or it was commanded to this state by the active SCM to reduce the complexity of the system configuration and bus load. The active SCM may command the SCC to be “disabled” if an SCC is not part of the selected SCS or if the operator chose the SCC as not being required for a recording phase.

This is the default state of an SCC after start-up

Ready: The SCC is ready to follow the active SCM state or to execute received client commands from the active SCM. On receiving a client command, the SCC shall change its state to “Play Back”.

Recording: The SCC is ready to send client commands during the recording phase.

Play Back: The SCC is ready to execute client commands received from the active SCM.

Abort: The actual play back phase shall be halted (see definitions in 4.4.3).

The states “Ready”, “Recording”, “Play Back” and “Abort” are considered the enabled SCC states, where the SCC participates in the SC communication. The states “Ready”, “Recording” and “Play Back” are

only allowed to be entered on command from the active SCM (see definitions in F.3 or E.6). The "Abort" state is allowed to be entered on command from the active SCM (see definitions in F.2), via the global SCClientAbort (see definitions in E.3) from another SCC or by the SCC itself.

While the SCM communicates the "Recording Completion" state, those SCCs having completed their functions shall fall back to "Ready"; only those still completing a function shall stay in "Recording" until the function is done. SCCs shall not indicate new function activations while in "Recording Completion". The relationships between the different states are illustrated in Figure 5.

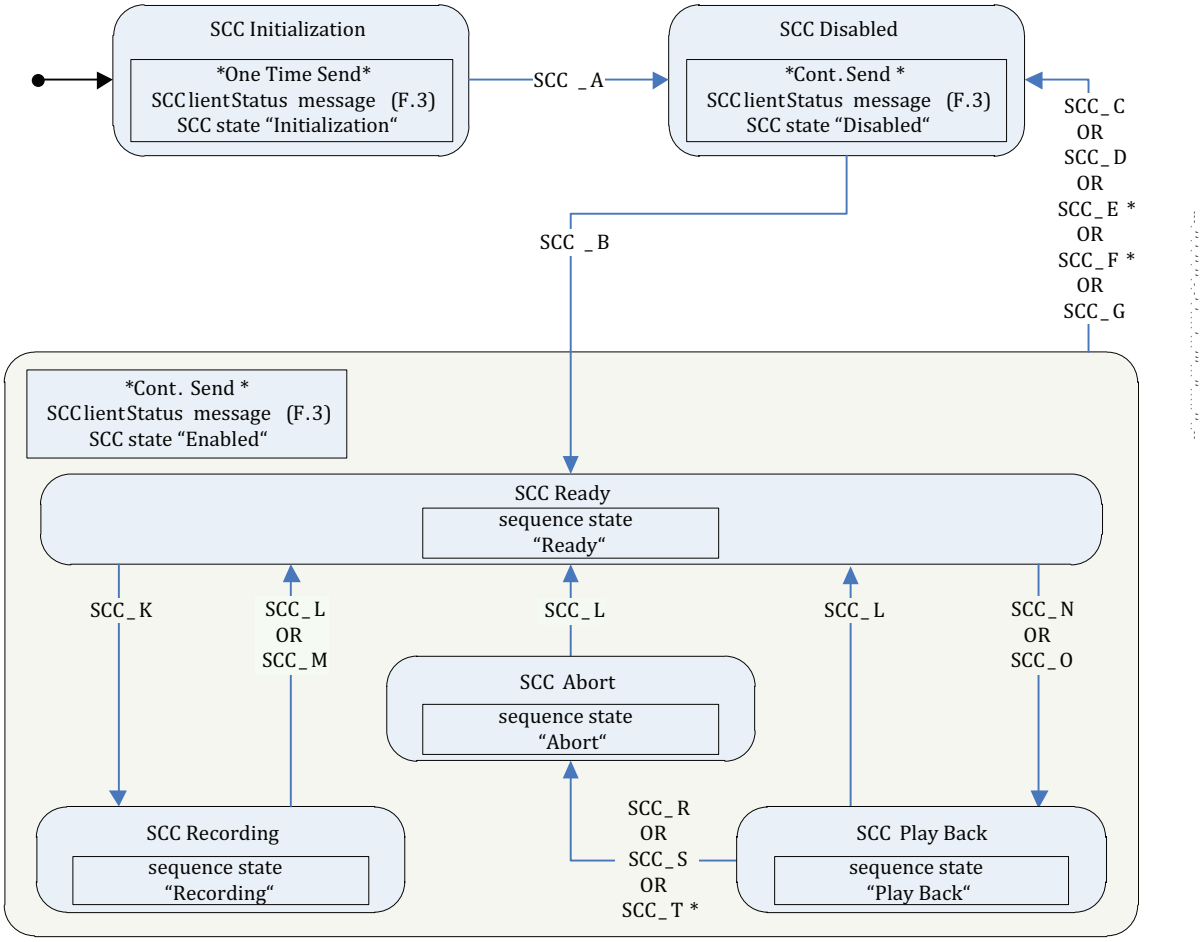


Figure 5 — SCC state diagram

Conditions for state transitions:			
SCC_A:	received SCM state “active”	SCC_L:	SCM state “Ready”
SCC_B:	SCM sends SCStateCommand “enabled”	SCC_M:	SCM state “Recording Completion AND all function execution of this SCC completed
SCC_C:	SCM sends SCStateCommand “disabled”	SCC_N:	SCM state “Play Back”
SCC_D:	Timeout for SCMasterStatus message	SCC_O:	receiving SCMasterExecutionCommand
SCC_E	Operator set SCC to not take part in SC system communication	SCC_R:	SCM state “Abort”
SCC_F *:	Critical internal SCC conditions	SCC_S:	any SCC send SCClientAbort
SCC_G:	SCM sends NACK for any SCC message	SCC_T *:	internal condition to abort SCS
SCC_K:	SCM state “Recording”		
* = optional requirement			

4.4.2 Initialization

At start-up of the system, the SCMs and SCCs identify themselves as independent units and participants on the ISO 11783 network (see definitions of ISO 11783-5 for address claiming process).

By default, all SCMs shall start inactive and all SCCs shall start disabled. Both, SCMs and SCCs, shall indicate their start-up by sending a status message (see definitions in F.2 and F.3) with the sending state set to initialization. This allows, for instance, the other participants in the communication to detect unexpected resets. For more details on SC initialization, refer to Figure 6, Figure 7 and Figure 8.

4.4.2.1 SCM initialization

a) The SCM shall identify itself to the VT with the function instance 0 and upload its object pool. By default, all external object pointers are inactive (parameters set to $FFFF_{16}$ – see definitions in ISO 11783-6 for details).

b) All SCMs shall stay inactive until the operator activates one using its user interface. The operator shall be able to change the active SCM at any time (see definitions in 4.4.4.1).

c) The selected SCM shall announce its activation and start-up by sending continuously the SCMasterStatus message (see definitions in F.2) indicating its active SCM state and the appropriate sequence state.

d) After its start-up, the active SCM shall delete all loaded SCD information from volatile memory and shall NACK all incoming SCClientStatus messages (see definitions of F.3) other than the start-up indication (see definitions of F.3 SCClientStatus messages byte 2) until an SCC indicates its start-up.

   The SCM may only receive SCClientStatus messages right after its own start-up when it has performed a quick reset within the interval of the SCMasterStatus message.

e) Upon receiving the SCDActivationCommand (see definitions of D.8) the SCM shall start parsing the SCD. The SCM indicates its parsing activity in its SCMasterStatus message (see definitions in F.2).

4.4.2.2 SCC initialization

a) After receiving an SCMasterStatus message (see definitions in F.2) from an active SCM, the SCCs shall begin sending continuously their SCClientStatus messages (see definitions in F.3) with the state set to disabled. An SCC is only allowed to change its state to enabled if the active SCM commands the SCC to be enabled (see definitions in 4.4.3). The first SCClientStatus message (see definitions in F.3) shall indicate the client initialization (see definitions in F.3, byte 2).

If the SCC starts up faster than the SCM, it shall wait for the SCMasterStatus message before it starts to send its own SCClientStatus message to indicate its initialization.

b) The SCC shall start its initialization by requesting the capabilities of the active SCM (see definitions in C.2 and C.3) to ensure that the version of the supported standard of the SCM and the SCC match; additionally, it shall be ensured that the object pool of the SCC fulfils the SCM requirements in respect of size of graphic objects (see definitions in C.3, byte 3 and 4) or font definitions (see definitions in C.3, byte 5 to 10) to avoid that important information gets clipped by the VT in case the objects are larger than defined.

c) The SCC's working set master shall identify itself to the VT with the function instance 0 and query this VT for its capabilities such as colour versus monochrome or language settings, adapt its object pool accordingly (see definitions in ISO 11783-6) and upload it to the VT. If the SCC's working set master does not provide its own user interface via this VT, it shall as a minimum upload the SCCOP to provide the textual and graphical objects for the SCMOP.

d) The SCCs next step is to check if its SCD is available in the SCM (see definitions in D.2 and D.3). The SCM shall use the NAME Mask information provided in the SCD basic objects (see definitions in A.3) to identify if the available SCD matches the requesting SCC (see definitions in A.3).

e) If there is an SCD matching the NAME information available in the SCM, the SCC shall check the version response (e.g. SW updates on the SCC may make the stored SCD invalid) and machine configuration reflected by the SCD to ensure that the SCD represents the client functions available at the SCC (see definitions in A.3, D.4 and D.5).

The content of the version label and configuration label is proprietary to the manufacturer of the SCC.

f) If the SCC finds a matching version of its SCD in the non-volatile memory or if the SCC has to upload its SCD, the SCC shall send an SCMasterMemoryRequest (see definitions in C.4) indicating the size of its SCD to be loaded or uploaded. The SCM shall respond with the SCMasterMemoryResponse (see definitions in C.5) including a status indication if the SCC can start the upload of the SCD by sending an SCDTransfer (see definitions in D.6) or send the SCDLoadCommand (see definitions in D.14) to load the SCD in volatile memory.

1) If no SCD is available or the version and/or configuration stored in the SCM do not match the data available from the SCC, the SCC shall upload a valid SCD (see definitions in A.3). If the SCC receives an SCMasterMemoryResponse (see definitions in C.5) indicating that the SCM is ready to receive or load the SCD (status code 00_{16}), the SCC shall start uploading its SCD using the SCDTransfer (see definitions in D.6). If the status code is 01_{16}, the SCC is not permitted to upload its SCD. The handling of this situation is proprietary to the SCC design.

The SCM shall respond with the SCDTransferResponse (see definitions in D.7) indicating the success or failure of the SCD transfer.

2) If a valid SCD is identified in the non-volatile memory of the SCM, the SCC may decide to use that SCD. Once the SCC receives an SCMasterMemoryResponse (see definitions in C.5) indicating that the SCM is ready to receive or load the SCD (status code 00_{16}), the SCC shall send an SCDLoadCommand (see definitions in D.14) to initiate the loading of the SCD into volatile memory to make it available for the use by the SCM.

The SCM shall respond with the SCDLoadResponse (see definitions in D.15) indicating the success or failure of loading the SCD.

g) After successfully transmitting or loading an SCD, the SCC shall activate the SCD (see definitions in D.8). This initiates the parsing for SCD integrity in the SCM. The SCC shall wait for positive feedback (see definitions in D.9) before it is allowed to continue the SC communication other than the SCClientStatus message (see definitions in F.3) communication.

h) To avoid uploading the SCD at each start-up, the SCC may initiate the storing of the successfully parsed SCD in the non-volatile memory of the SCM by sending the SCDStoreCommand (see definitions

in D.12). The handling of a possible negative response in the SCDStoreCommandResponse (see definitions in D.13) is up to the design of the SCC.

i) After successful completion of the initialization process, the SCC shall continue SC communications specified in the following clauses; however, it shall remain in the disabled state until the active SCM commands it to be enabled (see definitions in E.2).

j) Whenever an SCC receives a NACK of its status message, it shall return to the disabled state and may reinitialize the connection with the SCM.

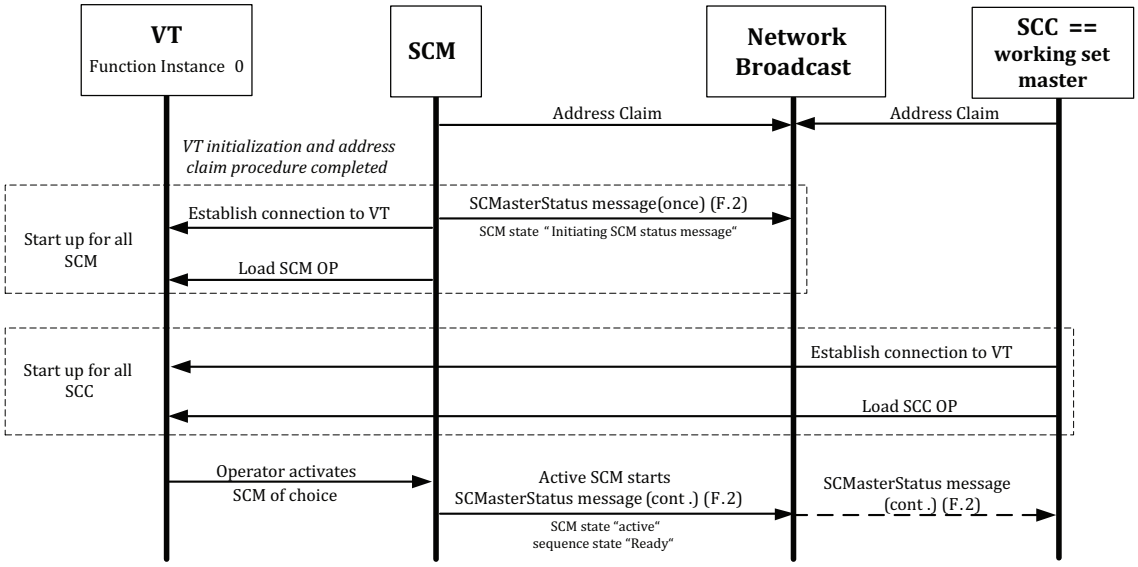


Figure 6 — Initialization message flow diagram for SCC as working set master

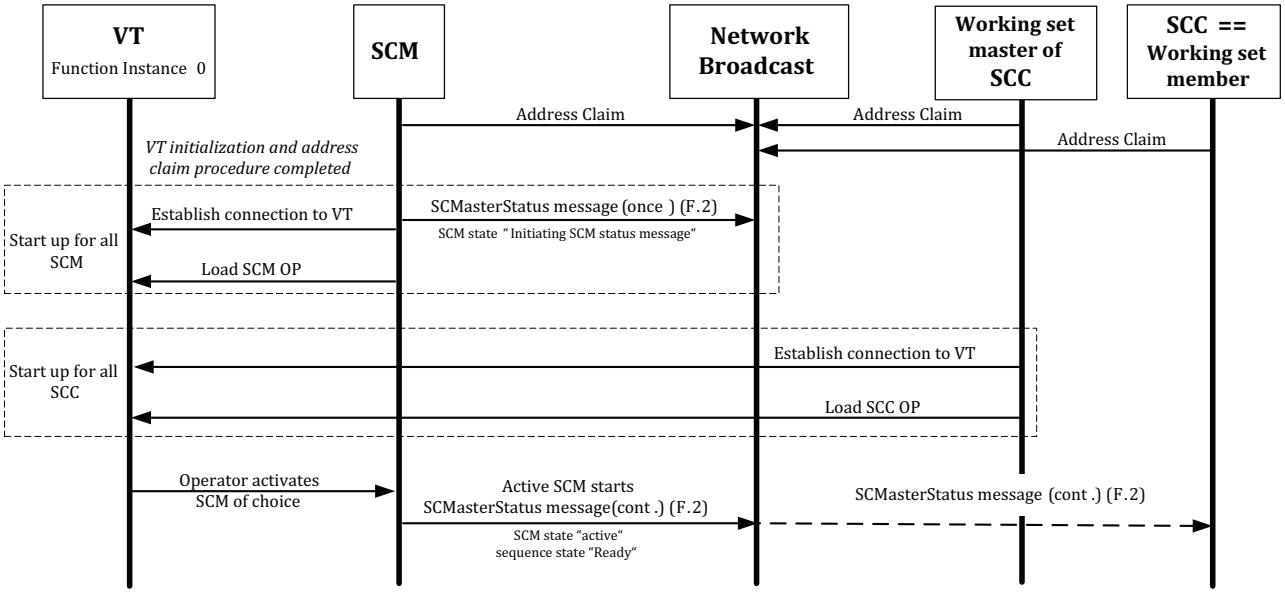


Figure 7 — Initialization message flow diagram for SCC as working set member

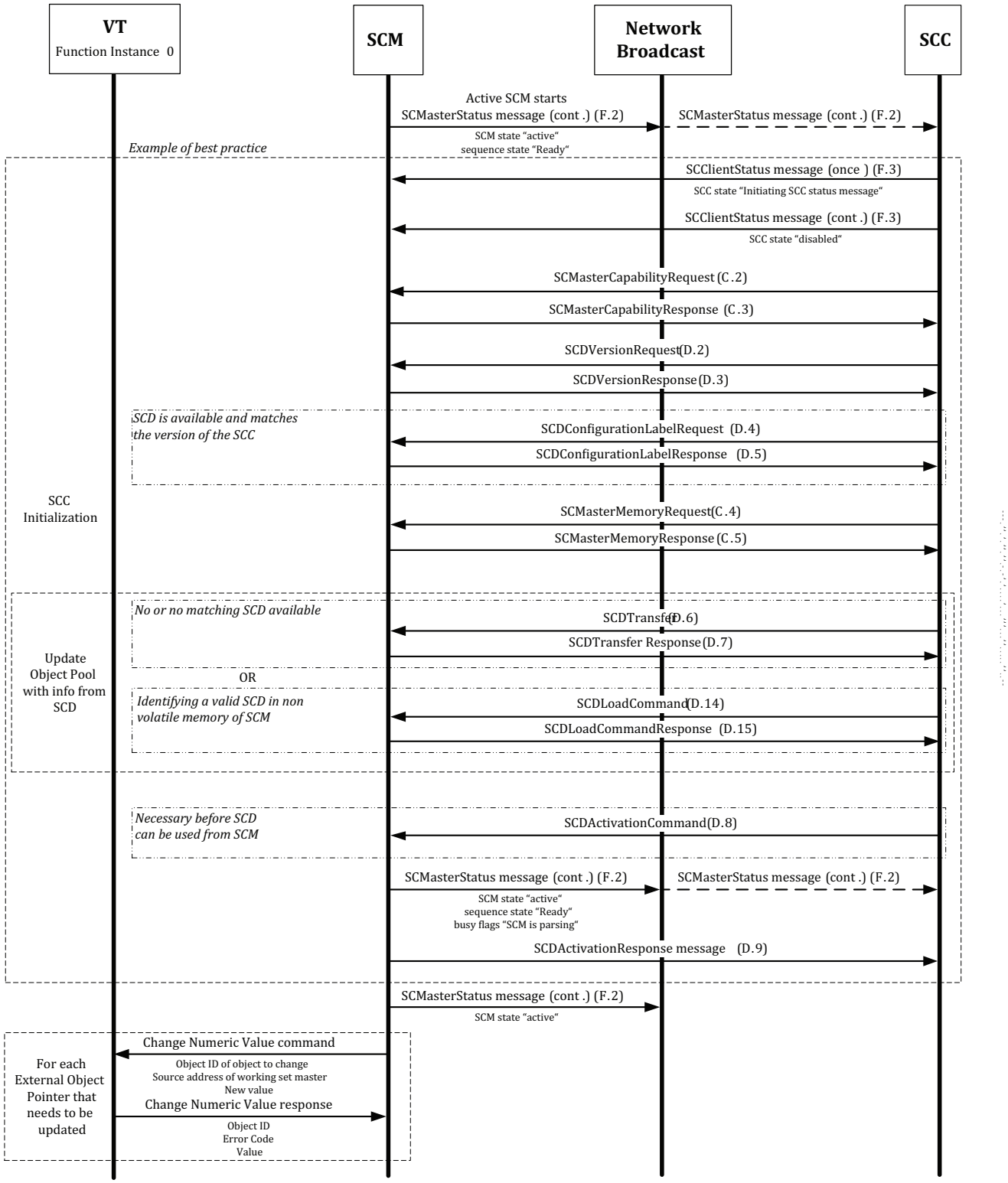


Figure 8 — Initialization message flow diagram

4.4.3 SCM/SCC state behaviour

The active SCM controls the state of the system. The SCM shall transmit its actual sequence state ("Ready", "Recording", "Recording Completion", "Play Back" or "Abort") to the ISO 11783 network in the

SCMasterStatus message (see definitions in F.2). The enabled SCCs shall transmit in their SCClientStatus message (see definitions in F.3) one of the following:

— “Abort” if SCC needs to halt the play back phase;

— “Recording” if the SCC has not completed its function execution and the SCM state is ”Recording Completion”;

— ”Ready” if the SCC completed its function execution and the SCM state is “Recording completion”;

— the same sequence state that is transmitted in the SCMasterStatus message (see definitions in F.2) in all other cases, even when one or multiple client functions are faulty.

During the states ”Recording”, ”Recording Completion”, ”Play Back” and ”Abort”, the SCM shall send the number of the selected sequence in its SCMasterStatus message (see definitions in F.2). The sequence numbers sent by the SCM and SCC shall be used for diagnostic purposes (i.e. to detect that SCM and SCC are out of synch) and the reaction is proprietary to the SCM or SCC. In the majority of cases, the sequence number transmitted by the SCM and SCC will be identical.

The active SCM shall only store client commands of enabled SCCs reflecting the recording state in their SCClientStatus message (see definitions in F.3). The state behaviour for the ”Recording” and ”Recording Completion” phase is described in 4.4.5.2.

The active SCM shall not send client commands to SCCs unless “Play Back” or “Ready” state are indicated in their SCClientStatus messages (see definitions in F.3). The state behaviour for the play back phase is described in 4.4.7.

The active SCM shall only accept SCClientAbort (see definitions in E.3) during ”Play Back” state. The state behaviour for a halted play back is specified in 4.4.7.3.

At selection of a sequence, the active SCM may command the state of all participating SCCs to become enabled and the state of all other SCCs present to become disabled (see definitions in Figure 9) prior to starting the play back phase. The enabling may also be done once after loading a set of sequences, where the SCM may enable all SCCs involved in this set.

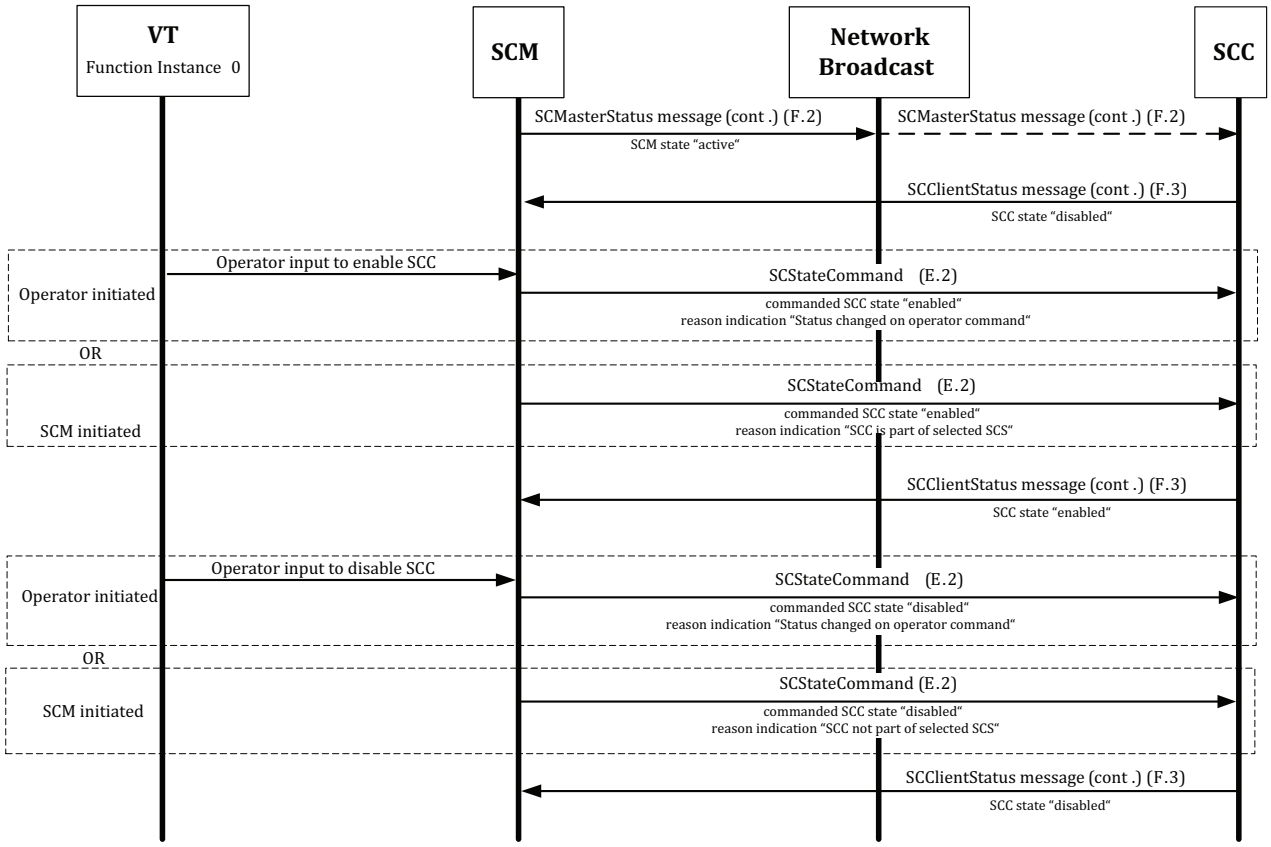


Figure 9 — State command message flow diagram

4.4.4 Shutdown

An SCM shall provide a means to be disabled by the operator. The SCCs may provide a means for the operator to select if the SCC shall no longer take part in the SC communication. This subclause describes the requirements for a graceful shutdown of an SCM or SCC.

4.4.4.1 SCM shutdown

For deactivation, the SCM shall be in its "Ready" state.

To activate another SCM, the operator shall deactivate the active SCM before activating the new one. If the operator tries to activate an SCM while another SCM is still active, the currently inactive SCM shall inform the operator about first deselecting the still active SCM.

After receiving the operator input for deactivation, the active SCM shall continue to transmit the "Ready" state in its SCMasterStatus message (see definitions in F.2) while it sends commands to disable the enabled SCCs (see definitions in E.2).

After all SCCs reflect the disabled SCC state, the SCM shall indicate its inactive state in its SCMasterStatus message (see definitions in F.2) before it stops sending its status message. For SCCs not accepting the disable command, refer to 4.5.3.5. This allows the active SCM to become inactive and one other available SCM to be activated by the operator. However, if the SCMs miss this one announcement, the timeout of the SCMasterStatus message (see definitions in F.2) shall occur before assuming that the previous active SCM has been deactivated.

Deactivated SCMs shall not send further SCMasterStatus messages (see definitions in F.2).

Disabled SCCs shall not initiate warnings to the operator on timeouts of the SCMasterStatus message (see definitions in F.2), but shall interpret the disappearing SCMasterStatus message (see definitions

in F.2) as a graceful shutdown. Accordingly, the SCCs shall stop their own SCClientStatus message (see definitions in F.3) until an SCM initiates a new SC communication.

4.4.4.2 SCC shutdown

An SCC may provide a means (e.g. VT screen, proprietary screen, dedicated button, etc.) to disable the SCC, for example, to take the SCC out of the ongoing SC communication. To avoid unnecessary warnings when the operator disables the SCC, the SCC shall first set its state in its status message to disabled (see definitions in F.3 SCClientStatus message byte 2) to inform the active SCM accordingly. The SCClientStatus message (see definitions in F.3) shall be continued until receiving the first NACK of this message to avoid SCClientStatus message (see definitions in F.3) timeouts (see definitions in 4.5.3.6).

4.4.5 Sequence setup

The SCM shall provide at minimum one method to define a sequence based on the SCD information available from the connected SCCs.

EXAMPLE 1 A manual sequence definition by the operator (EDIT)

EXAMPLE 2 A recording phase where the operator commands are recorded by the SCM (RECORDING) while the operator activates each desired function manually.

The SCM may provide a means to the operator to enable/disable SCCs to take part in the sequence setup (in particular to reduce the complexity of the system during manual sequence definition).

The SCM shall ensure the use of a unique sequence number for all sequences which may be activated during an operation, to ensure the proper system status identification.

EXAMPLE 3 The SCM provides 4 inputs to activate 4 individual sequences for specific operations in a field; this SCM shall assign a unique sequence number to the individual sequences assigned to these inputs.

4.4.5.1 Sequence control sequence

An SCS shall consist of

— the NAMEs, SCD Version Labels, and SCD Configuration Labels of all SCCs involved in the SCS

— the SCC reference, SCC function ID, state ID, trigger point, and function value (see definitions in 4.4.8) for each step of the SCS.

4.4.5.2 Sequence recording

If the SCM provides the possibility for the operator to record a sequence, the SCM shall provide an appropriate way to start and stop recording.

During normal operation, the operator may choose to record a new sequence or to rerecord a previous one.

The SCM is only allowed to enter the “Recording” state when all enabled SCCs are present and indicate in their SCClientStatus message (see definitions in F.3) that they are “Ready”.

The start of the recording phase is indicated by the SCM by changing its state to ”Recording” in the SCMasterStatus message (see definitions in F.2). The SCCs shall follow the SCM state immediately, even when the ClientFunctionErrorState of the SCClientStatus message (see definitions in F.3) indicates errors in one or more client function. There shall only be a single SCS and recording phase active at a given time.

The SCM shall only record client functions of enabled SCCs reflecting the ”Recording” state in their status message. In case an SCC does not follow the SCM state within the timeout defined in Annex F.3 the SCM shall take the action specified in 4.5.3.4.

The recorded client functions are typically a response to an operator action during recording; the operator manually activates the client function that triggers an SCExecutionIndication (see definitions in E.4) from

the SCC to the SCM. SCCs shall not send SCExecutionIndication (see definitions in E.4) for faulty functions. The SCC may choose to not report a client command in response to every operator action, e.g. due to a condition that temporarily or permanently prevents the function from responding to the operator input. In some cases, the SCC may also choose to report client commands of steps that are a result of indirect operator action. For example, a single operator action during the recording phase may cause a sprayer to both raise the boom and shut off flow. In this case, the implement may report this as two separate client commands to the SCM or as a single composite client command. The SCC would not typically report commands that are a result of an action/command from another system such as a Task Controller.

Once recording starts, the active SCM shall record the step information defined in 4.4.5.1 for each received SCExecutionIndication (see definitions in E.4) of enabled SCCs reflecting the ”Recording” state.

The SCM shall record the related trigger whenever an SCC reports the start of the function execution for a client function (see details in E.4).

The function value shall be recorded when the SCC indicates the completion of the function execution (see details in E.4).

A client function shall only be saved in the SCS if the SCM receives both indications for ‘function execution started’ and ‘function execution completed’.

For each function activation, the SCC shall send an SCExecutionIndication (see definitions in E.4) as a client command to the SCM, including the function ID, state ID and function value of the client function to be recorded. The TAN of this initial client command of a client function shall be the previous one of the SCC increased by one.

The SCC shall repeat every SCExecutionIndication (see definitions in E.4) with the same TAN on a 100 ms interval until the SCM changes back to ”Ready” state, or until a timeout (see definitions in F.2), or until the SCM acknowledges the client command by sending an SCExecutionIndicationResponse (see definitions in E.5).

Figure 10 illustrates the message process during a sequence recording.

The SCC client command may be one of the following:

a) Combined Start and Complete client command

   The SCC client command shall have the function execution state set to 00 (= function execution started and completed simultaneously) when the client function is started and simultaneously completed again (e.g. short trigger to switch something on that runs continuously afterwards).

b) Individual Start and Complete response

   The SCExecutionIndication (see definitions in E.4) shall have the function execution state set to 01 (= function execution started) when the started function continues to run for a certain time (e.g. a hydraulic cylinder moves for a certain time).

   When the execution of the client function is completed (e.g. the hydraulic cylinder reached its target position), the SCC shall indicate this by sending another SCExecutionIndication (see definitions in E.4) with the same content and TAN as above, but the function execution state set to 10 (= function execution completed). This second indication may contain a process variable in the function value to be stored with the command in the sequence.

c) Individual Start and Complete but with error in-between

   This scenario starts as (b) with the SCExecutionIndication (see definitions in E.4) indicating the start of the execution but with an error in the client function before it may be completed. In this case the SCC shall send the second SCExecutionIndication (see definitions in E.4) with the function execution state set to 11 (= error during execution) to indicate the error condition and the same TAN as in the first client command. The SCC shall not send an additional SCExecutionIndication (see definitions in E.4) to indicate that the function execution completed.

If there is a faulty function, the SCC shall indicate this error also in its SCClientStatus message (see definitions in F.3). If the ClientFunctionErrorState (see definitions in F.3 SCClientStatus message byte 5) is set to 03_{16} (required operator confirmation), the SCM shall request more detailed information with the SCClientFunctionErrorRequest (see definitions in E.9) and shall inform the operator about the faulty client functions indicated in the SCClientFunctionErrorResponse (see definitions in E.10 function error structure byte b bit 8). If operator confirmation is not required, the SCM may request more detailed information while it is up to the SCM to either ignore the faulty client function and to continue the recording phase or to cancel the whole activity to allow the operator to fix the issue.

d) Individual Start and Complete but with operator activation of same client function in-between

This scenario starts as (b) with the SCExecutionIndication (see definitions in E.4) indicating the start of the execution but with an operator activation of the same client function before it may be completed. In this case the SCC shall send the second SCExecutionIndication (see definitions in E.4) with the same TAN as before but indicating the completed function execution.

After sending the second SCExecutionIndication (see definitions in E.4) for the first operator activation, the SCC shall immediately send another SCExecutionIndication (see definitions in E.4) using the previous TAN incremented by one and function execution state indication dependant on the case description (a) – (c).

e) Start of an function while the previous one is not completed

It is feasible that the next function has to be activated while the previous one of the same SCC is not yet completed (e.g. hitch is still in the process to go down while the PTO is activated when reaching a certain hitch height). The individual TANs for the two functions allow a clear identification of the commands and their related responses.

On receiving an SCExecutionIndication (see definitions in E.4), the SCM shall send an SCExecutionIndicationResponse (see definitions in E.5) with the following information:

— the same function ID, state ID, function execution state and TAN as received in the SCExecutionIndication (see definitions in E.4)

— indication in byte 4 if the client command was recorded or the reason why the client command was discarded. The SCC of the discarded function may inform the operator that the client command sent was discarded by the SCM.

During recording, the reception of SCClientStatus messages (see definitions in F.3) of all enabled SCCs shall be validated by the SCM and as soon as the state of an SCC falls back to disabled (see definitions in 4.5.3.6), or any SCC message is timed out (see definitions in Annex B), the complete sequence recording shall be cancelled by the SCM (see definitions in 4.5.2). The SCM communicates the cancelation by changing its state to “Ready”.

The SCM shall provide a means for the operator to stop the recording phase. When the SCM stops the recording phase, it shall take one of the following actions:

— if the SCM is still waiting for some function execution complete indications it shall enter the ”Recording Completion” state. Once they are all received the SCM shall enter the ”Ready” state;

— otherwise, the SCM shall immediately enter the ”Ready” state which shall be reflected by the SCCs immediately.

SCCs receiving the SCM state ”Recording Completion” shall take one of the following actions:

— if all client functions of this SCC completed their execution and sent the function execution completed indication the SCC shall enter the ”Ready” state;

— otherwise, the SCC shall stay in the ”Recording” state, finish the ongoing client function execution, send the function execution completed indication and as soon as the SCC completes all its client functions it shall enter the ”Ready” state. While finishing the “Recording” state the SCC is not permitted to indicate new client commands.

If and how recorded or manually defined sequences are stored in non-volatile memory is up to the design of the SCM. The SCM may cancel the recording phase by directly entering the ”Ready” state which accordingly shall be reflected by the enabled SCCs. However, an SCC may continue to perform its client functions, since they are not activated by the sequence control system but by the operator.

If not all SCCs follow to the ”Ready” state, the SCM shall behave as defined in 4.5 (see 4.5.3.4).

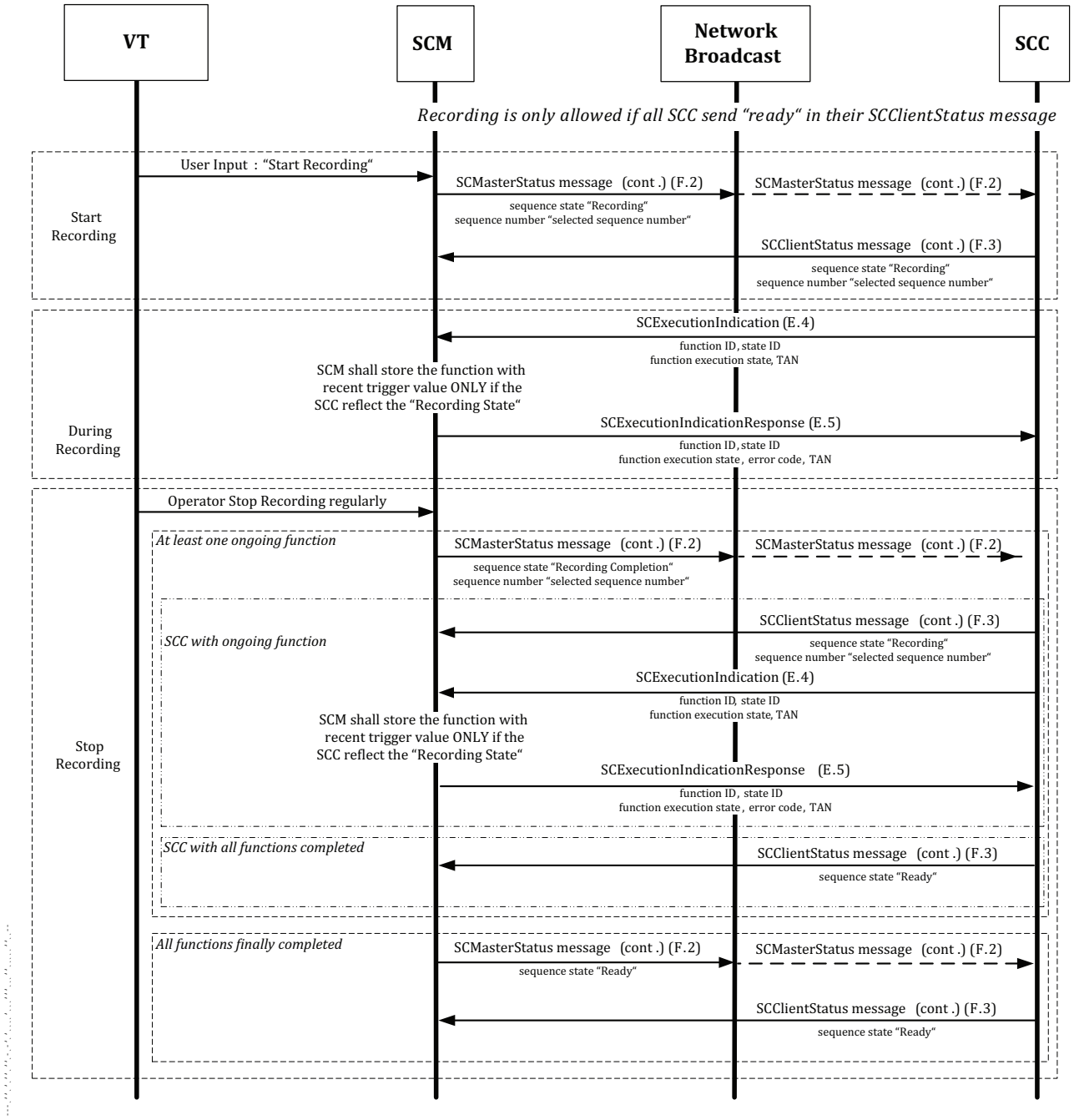


Figure 10 — Recording message flow diagram

4.4.5.3 Manual sequence edit

“Manual Edit” allows the operator to manually change an already available sequence or to define a new sequence without performing the client functions in the recording mode.

If the SCM provides the possibility for the operator to setup a sequence via ”Manual Edit”, this shall be an internal state of the SCM and shall not be indicated to the SCCs. The global system state provided by the SCM shall be ”Ready” with the Sequence number set to FF_{16}.

In “Manual Edit”, the SCM shall provide the client functions, associated function states from the enabled SCCs, a function value according to the presentation settings defined in the SCD function object (see definitions in A.4) and the available trigger method for each function in its edit screens (see definitions in Annex A).

“Manual Edit” shall provide a means to the operator to set the information defined in 4.4.5.1 for each step of the SCS.

If a function value is defined by the SCC to not be changeable by the operator (see definitions in A.4), the function value shall be

— set to $FFFFFFFF_{16}$ when a new step is created, and

— preserved when an existing step is edited.

4.4.6 Sequence validation

The active SCM shall validate the selected sequence, or set of sequences in the SC system before allowing the play back of this sequence(s) for the following cases:

— after load of previously stored sequence(s);

— on timeout of any SCClientStatus message (see definitions in F.3);

— after completion of manual sequence definition or editing;

— after unexpected state change of any SCC (see definitions in F.3 byte 2);

— after any SCD upload.

The SCM shall check the availability of the involved SCCs and their SCDs. The SCM may inform the operator and shall not allow the play back of a sequence that does not meet the following validation requirements.

— The 64bit NAMEs of the connected SCCs and their working set masters listed within the SCD match those stored with the selected sequence. The SCC NAME Mask shall be applied to this comparison as described in A.3 SCD basic object.

— The SCD version label and SCD Configuration Label match those stored with the selected sequence.

— All functions of the selected sequence are contained in the loaded SCDs.

The SCM shall allow the operator to make the appropriate changes to the SCS to meet the validation requirements.

After a successful validation, the SCM may ensure that all participating SCCs in the selected sequence(s) are enabled and the SCM may disable all other SCCs by sending SCStateCommand (see definitions in E.2) to the individual SCCs.

4.4.7 Sequence play back

4.4.7.1 General

The SCM may provide a means to select and load a previously stored sequence(s) (see definitions in 4.4.9). The SCM shall validate the selected sequence(s) (see definitions in 4.4.6); otherwise, the operator needs to record or manually define a sequence to be used.

The activation of the sequence play back is proprietary to the active SCM. It may provide multiple inputs like softkeys on a VT, Auxiliary Inputs (see detailed definitions in ISO 11783-6) and/or dedicated buttons in the operator's vicinity.

An SCS needs to be successfully validated (see definitions in 4.4.6) before the SCM is allowed to play the sequence back. If the sequence validation failed due to for example an SCC being included in the sequence but not present on the ISO 11783 network, the SCM may provide a proprietary means to the operator to take an appropriate action for getting the sequence valid for example by removing client functions from the SCS.

The transition from "Ready" to "Play Back" shall only be allowed if all SCCs included in the selected sequence indicate in their SCClientStatus message (see definitions in F.3) that they are "Ready".

However, for a quick operator and system reaction, the SCM may provide a means to start the next play back phase even when the previous one has not completed yet (one or multiple SCCs are still in "Play Back" for the previous sequence since they are still performing the last commands). In this case, the SCM shall stop sending commands for the first sequence and change the sequence number in its SCMasterStatus message (see definitions in F.2) to indicate the change to the SCCs before starting to send commands for the next sequence because only one sequence is allowed to be active at any time. The SCM has to be aware that those SCC still performing commands from the previous sequence may send feedback on those activities, what shall not be seen as a fault.

The SCM indicates the start of the play back phase by setting its state to "Play Back" in the SCMasterStatus message (see definitions in F.2). The enabled SCCs shall follow the SCM state to "Play Back" in their SCClientStatus message (see definitions in F.3). However, the SCM may send client commands even to SCCs that are still reporting the "Ready" state to ensure an immediate start of the "Play Back". The SCC missing the SCM state change and receiving the first client command shall interpret this command as the trigger to change its status to "Play Back" state before executing the client command. An SCC shall always follow the latest received command of the SCM. If an SCC does not follow the SCM state within the timeout defined in Annex F.3, the SCM shall act as specified in 4.5.3.4.

To ensure that the SCM and SCC do not get out of synch during the different stages of a client function execution, the following rules shall apply each time a trigger event stored in the executed SCS is reached for execution of the next client function listed in the replayed sequence:

— The SCM shall send an SCMasterExecutionCommand (see definitions in E.6) to the client function owning SCC, including the function ID, state ID and function value of the client function to be executed. The TAN of this initial execution command for a client function shall be the previous one of the SCM increased by one.

— The SCM shall repeat the SCMasterExecutionCommand (see definitions in E.6) with the same TAN on a 100ms interval until the SCC replies with the SCClientExecutionStatus (see definitions in E.7) or until a timeout.

— The SCC shall respond on every command of an SCM with an SCClientExecutionStatus (see definitions in E.7), including the function ID, state ID and sequence number received by the SCM, and repeat this message on a 100 ms interval until the SCM changes its state to "Abort", or until a timeout (see definitions in E.1), or until the SCC reaches the point to report the function execution completed, or until the SCC reaches a condition requiring an abort of the execution (see definitions in 4.4.7.3), or until the SCM acknowledges the response by sending an SCClientExecutionAcknowledgement (see definitions in E.8).

— The SCM shall respond to every SCClientExecutionStatus (see definitions in E.7) with an SCClientExecutionAcknowledgement (see definitions in E.8) with the same function ID, state ID and function execution state as in the related SCClientExecutionStatus (see definitions in E.7), and with the TAN as in the related SCMasterExecutionCommand (see definitions in E.6).

Figure 11 illustrates the message process during a sequence play back.

4.4.7.2 SCClientExecutionStatus

The SCClientExecutionStatus (see definitions in E.7) may be one out of the following:

a) Combined Start and Complete

The SCC reply shall have the function execution state set to 00 (= function execution started and completed simultaneously) when the client function is started and simultaneously completed again (e.g. short trigger to switch a function on that operates continuously after the command; means the operator interaction is a momentary button press). The TAN of this reply is the same as used by the SCM in the SCMasterExecutionCommand (see definitions in E.6).

b) Individual Start and Complete

The SCC reply shall have the function execution state set to 01 (= function execution started) when the function started continues to run for a certain amount of time (e.g. a hydraulic cylinder moves for a certain amount of time). The TAN of this reply is the same as used in the related SCMasterExecutionCommand (see definitions in E.6).

When the execution of the client function is complete, the SCC shall indicate this by sending another SCClientExecutionStatus (see definitions in E.7) with the same content as above, but with the function execution state set to 10 (= function execution completed).

c) Individual Start and Complete but with error in-between

This scenario starts as (b) with the SCC reply indicating the start of the execution but with an error in the client function before it can be completed. In this case, the SCC shall send the second SCClientExecutionStatus (see definitions in E.7) with the function execution state set to 11 (= error during execution) to indicate the error condition. The TAN of this reply is the same as used in the related SCMasterExecutionCommand (see definitions in E.6). The SCC shall not send an additional SCClientExecutionStatus (see definitions in E.7) indicating that the function execution is completed.

d) Execution Command for a faulty client function

If the SCC cannot execute a commanded client function due to an error in the function or on operator input, the SCC shall report the reason as an Error Code in the SCClientExecutionStatus (see definitions in E.7) with the function execution state set to 11 (= error during execution) to the SCM. The TAN of this reply is the same as that used in the related SCMasterExecutionCommand (see definitions in E.6). The SCC shall not send an additional SCClientExecutionStatus (see definitions in E.7) indicating that the function execution completed.

e) Individual Start and Complete but with overlapped Command for the same client function

If an SCC receives a execution command for the same client function it is still executing, the SCC shall first send one SCClientExecutionStatus (see definitions in E.7) using the TAN of the first execution command and indicating the completion of this function (function execution state set to 10 = function execution completed).

After sending the complete indication for the first command, the SCC shall immediately send out a second SCClientExecutionStatus (see definitions in E.7) using the TAN of the second SCMasterExecutionCommand (see definitions in E.6) and the function execution state indication dependant on the case description (a) – (d).

Whenever a client function is faulty, the SCC shall indicate this in the SCClientStatus message (see definitions in F.3) in addition to the SCClientExecutionStatus (see definitions in E.7). If the ClientFunctionErrorState (see definitions in F.3 byte 5) is set to 03_{16} (required operator confirmation), the SCM shall request more detailed information with the SCClientFunctionErrorRequest (see definitions in E.9) and shall inform the operator about the faulty client functions indicated in the SCClientFunctionErrorResponse (see definitions in E.10, byte b bit 8). If operator confirmation is not required, the SCM may request more detailed information; it is up to the SCM to either ignore the faulty client function and to continue the play back phase or to halt the whole activity by changing its state to "Abort" (see definitions in 4.4.7.3).

If the SCM does not receive a response message with the right TAN, client function ID and state ID within the timeout, the SCM (see definitions in E.1) shall halt the ”Play Back”.

During play back, the reception of SCClientStatus messages (see definitions in F.3) of all enabled SCCs shall be validated by the SCM and as soon as the state of an SCC falls back to disabled, or any SCC message is timed out (see definitions in F.3), the complete sequence play back shall be halted by the SCM (see definitions in 4.5.2 and 4.4.7.3).

Any operator activation of client functions during play back shall be indicated to the SCM via the SCExecutionIndication (see definitions in E.4) by setting the function execution state to 11 (= error during execution). The SCM may halt the whole sequence play back if the activated client function is part of the SCS being executed or may decide to not send any new commands to that client function during this play back phase. The SCM is also allowed to just ignore the SCExecutionIndication (see definitions in E.4) during play back with regards to the need to halt the play back phase or to inform the operator but not with regards to the message communication.

The SCC shall repeat its SCExecutionIndication (see definitions in E.4) on a 100ms interval until the SCM changes back to “Ready” state, or until a timeout (see definitions in E.1), or until the SCM acknowledges the client command by sending an SCExecutionIndicationResponse (see definitions in E.5) with the same function ID, state ID and TAN.

A play back normally ends with the indication of the completion of the last function execution by the executing SCC. After receiving all outstanding function execution states as completed or based on internal conditions, the SCM shall enter the “Ready” state and this shall be reflected by the enabled SCCs in the SCClientStatus message (see definitions in F.3) accordingly. The SCM may enter the “Ready” state even if not all outstanding function execution states were received as completed (e.g. SCM internal timeout for play back phase).

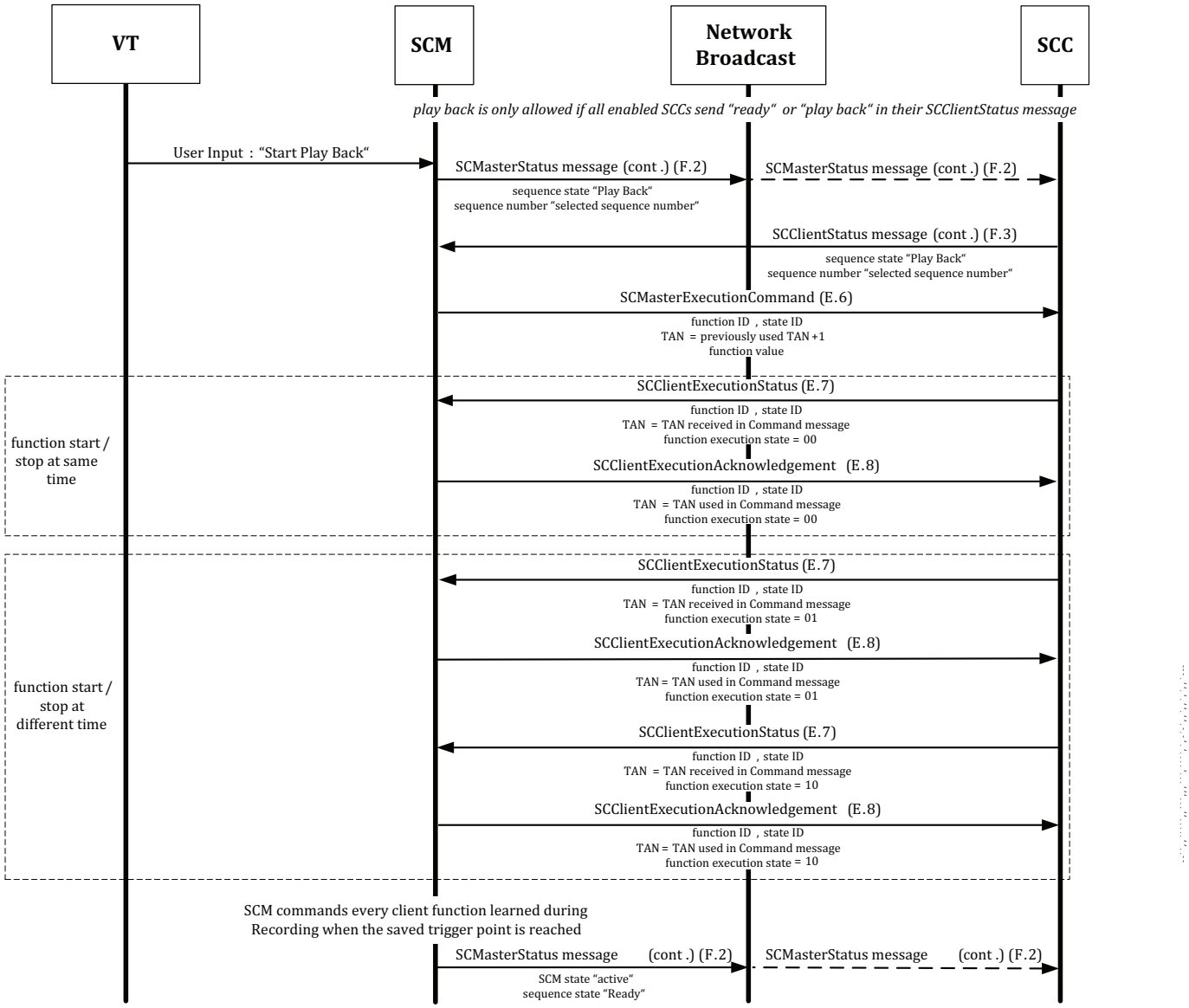


Figure 11 — Play back message flow diagram

4.4.7.3 Abort play back

The active SCM shall provide a means for an operator to halt the "Play Back". If internal conditions occur during "Play Back" requiring an "Abort", the SCM may initiate an "Abort" without operator input. If an SCC has to halt the ongoing play back phase, it shall send the global SCClientAbort (see definitions in E.3) until it receives the "Abort" state from the active SCM and the SCC shall report the "Abort" to the active SCM with the "Abort" state in its SCClientStatus message (see definitions in F.3). Only during "Play Back" shall the active SCM immediately change the system state to "Abort" upon receiving an SCClientAbort (see definitions in E.3) from any enabled SCC or recognizing that an SCC transmits the "Abort" state. If the active SCM needs to halt the actual play back phase itself, it shall enter the "Abort" state that shall be reflected by the SCCs immediately. During the "Abort" state, the SCCs are responsible for commanding any client function(s) which were commanded by the SCM in the current sequence into a safe state defined by the SCC. SCC functions not being commanded by the SCM shall not be affected by the SCM halt. In the case of an "Abort", the SCM may notify the operator accordingly.

After the SCM receives the "Abort" state from all enabled SCCs, it shall go back to the "Ready" state and this shall be reflected in the SCClientStatus message (see definitions in F.3) by the SCCs accordingly.

Figure 12 illustrates the message process for halting play back.

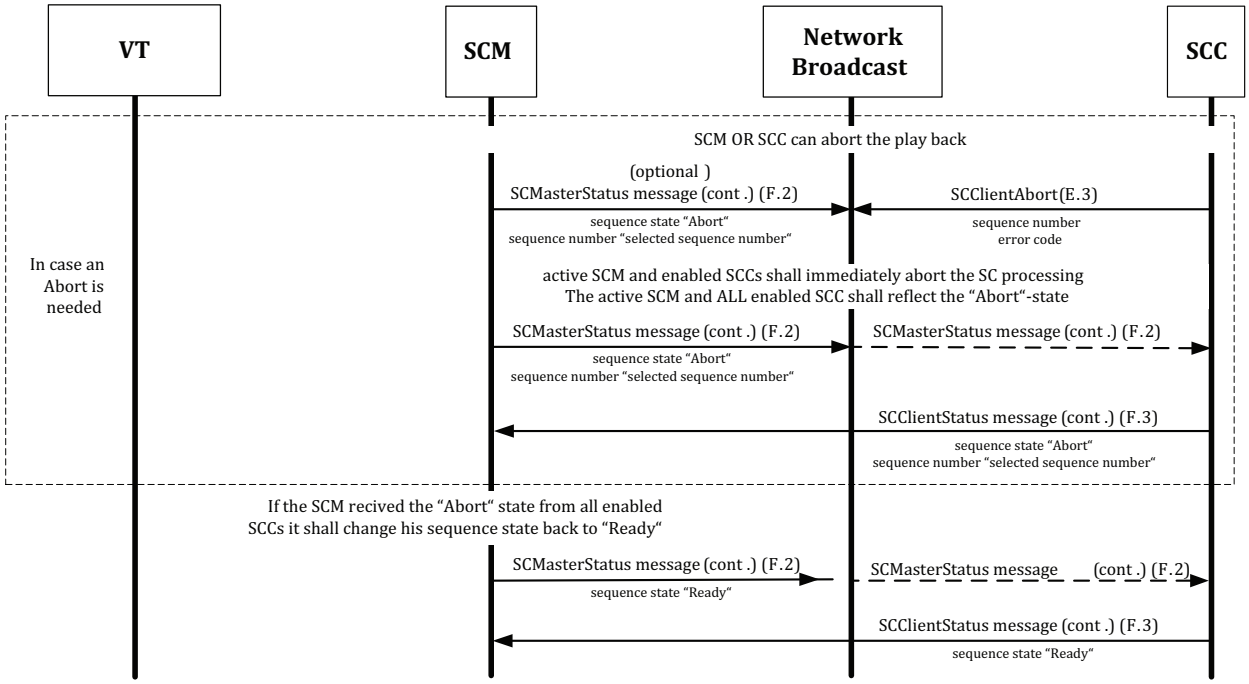


Figure 12 — Abort message flow diagram

4.4.8 Function value

Each step in an SCS can have a function value associated. The intention of this process variable is to allow the SCC to store specific settings or process parameters with each function execution stored in an SCS. The function value will be sent back to the SCC with each command to perform the function again during play back. The meaning of this 4 byte process variable is proprietary to the function owning SCC.

The function value is usually sent by the SCC with the SCExecutionIndication (see definitions in E.4) indicating the state "function execution completed" during recording or is manually defined by the operator in edit mode. If an SCC only indicates the start of client function activation, it shall set the function value to $FFFFFFFF_{16}$. The SCC defines in its SCD function object (see definitions in A.4) if and how the function value is displayed on the screens of the SCM.

The function value shall be stored together with the trigger timing and ID of each individual step in the SCS. During play back, the function value shall be sent back to the SCC as part of the function execution command (see definitions in E.6). The SCM shall allow the activation of a particular client function multiple times independent of each other within one or multiple sequences (e.g. the full hitch height may be reached in multiple steps within one sequence and the SCM shall treat each step individually in its sequence).

EXAMPLE During "Recording", the operator lifts the 3-point-hitch to a certain position. The affected SCC indicates the start of the operator input with the SCExecutionIndication (see definitions in E.4) to the SCM. When the hitch reaches the intended position, the operator stops the hitch movement and the SCC indicates the "function execution completion" including the reached position as function value to the SCM who stores the value with this function in the SCS (see details in SCExecutionIndication in E.4).

During "Play Back", the SCM sends this function value as part of the SCMasterExecutionCommand (see definitions in E.6) back to the SCC. This allows the SCC to perform the function in the same manner as the operator has done during recording and stop at the very same hitch position.

4.4.9 Sequence management

This subclause specifies the management process to handle recorded sequences to be supported by the SCM. The menu of the SCM may provide a means by utilizing the user interface capabilities to review, load or replace stored sequences. It is proprietary to the SCM to allow the operator to review, save, load or replace stored sequences and link the sequences to an input for activation of a play back phase.

4.4.9.1 Sequence storage

The SCM may store a single sequence or a set of sequences (e.g. one for entering the headland and one for leaving into main field again) together with a unique text descriptor. An SCM providing a means to store recorded sequences shall take appropriate measures to allow for a sequence data integrity check when the sequence is loaded at a later time (i.e. store a checksum together with the sequence data).

4.4.9.2 Sequence loading

An SCM providing a means to store recorded sequences shall provide a means to load those sequences again. According to 4.4.9.1, the sequence data may represent a single sequence or a set of sequences. When loading a sequence or set of sequences, the SCM shall check the integrity of the sequence data and it shall not allow loading corrupted sequences.

After loading a sequence or set of sequences, the SCM shall validate the sequences, as described in 4.4.6.

4.4.9.3 Sequence editor

The SCM may provide the means to edit the recorded sequence(s) to allow the operator to change details such as the timing or distance between two functions or to change the trigger event for a certain client function if the default one does not serve the operation appropriately

EXAMPLE 1 Change the trigger from distance-based to time-based.

The sequence editor may provides a means to the operator to manually chain up functions in a way that one function has to wait for the completion of the previous one.

EXAMPLE 2 The PTO can't be switched on before the hitch reached a certain position.

If the SCM provides a "Edit-Mode", the operator may use this mode for manual definition of a sequence (see definitions in 4.4.5.3).

4.5 Error handling

Multiple error conditions are described in this document, while the general behaviour is summarized in this subclause. Dependent on the failure, the SCM, one or multiple SCCs, or both together, shall take appropriate action.

The SCCs report the condition of their client functions in their SCClientStatus message (see definitions in F.3). If the SCM recognizes that the error state of one or more client functions has changed since the last report (see definitions in F.3 byte 5 value 02_{16}), it may request detailed information with the SCClientFunctionErrorRequest (see definitions in E.9). If an SCC indicates that one of its client functions changed the error state and needs operator confirmation (F.3 byte 5 value 03_{16}), the SCM shall request the detailed information. The SCClientFunctionErrorResponse (see definitions in E.10) is at least used by the SCM to find out the faults where operator confirmation is required (see definitions in E.10, byte 3-N error structure byte b bit 8) and possibly determines if a client function in one of the sequences is faulty or if other, not used client functions are affected. After sending the report, the SCC shall change its ClientFunctionErrorState back to 01_{16}. The SCM shall immediately report faulty client functions to the operator which are transmitted with the operator confirmation required bit set. The SCM may include the operator in the decision process on the further proceeding.

4.5.1 Function failure

A detected failure of a client function that is reported by the SCC shall be acted upon by both the SCC and the SCM. The SCC shall be responsible for taking the appropriate action. The action taken is dependent on the particular SCC and can, for example, include stopping the current executing action or simply ignoring further commands. The SCM, in this case, may react by skipping any steps for this client function in the current sequence until the SCC reports the failure has cleared. This means if the failure exists at beginning of a subsequent sequence, the SCM may continue to skip the function during that sequence. Clearing of the detected error is the responsibility of the SCC and may include operator interaction through the SCC user interface.

The system reaction on function errors may depend if the faulty client function is included in one of the loaded sequences or not.

4.5.2 Communication failure

The action taken on a communication error/timeout is dependent on whether the SCC or the SCM detected the failure. If the SCC detects a communication error, it shall take appropriate action under consideration of general safety requirements.

If the SCM detects a communication failure with a client function or set of client functions being part of the selected sequence, the SCM shall either not allow recording or play back or the SCM shall immediately cancel the recording phase or halt the play back phase if it is already active. Since communication problems can be intermittent, the SCM may require operator intervention before allowing a recording or play back phase again.

NOTE More specific system behaviour on certain communication issues is covered under 4.5.3.

4.5.3 Non-conform behaviour of a participant

The following subclauses specify the action taken by the SCM and SCC on non-conform system behaviour.

4.5.3.1 Unexpected SCC messages

The SCM shall NACK all SC messages coming from SCCs being not part of a valid connection according to the definitions of 4.4.2. Since a loaded and validated SCD is part of a successfully completed initialization process, connections to SCCs whose SCD was deleted by the SCM, are considered invalid (see examples in following subclauses of 4.5.3).

If after the initialization process, an SCC sends an SC message that it is not supposed to send, the SCM shall NACK this message.

EXAMPLE A disabled SCC sends out an SCExecutionIndication (see definitions in E.4).

The SCC may re-establish its connection to the SCM when receiving unexpected NACKs.

4.5.3.2 Missing SCC responses

The SCM shall delete the SCD from the volatile memory of SCCs not responding to commands and requests within the timeout defined in E.1 and Annex F or not sending their frequent status message. This forces these SCCs to re-establish the connection to the active SCM since the SCM will NACK all incoming SC messages of SCCs without SCD available (see details in 4.5.3.1).

4.5.3.3 Multiple SCMs active

If an SCC recognizes that more than one SCM is active on the ISO 11783 network by receiving different SCMasterStatus messages (see definitions in F.2), the SCC shall immediately change its status to disabled.

4.5.3.4 SCC does not follow the SCM state

If an SCC does not follow the SCM sequence state within the timeout definitions of F.3, the SCM shall delete the SCD from volatile memory and NACK the SC messages of this SCC to ensure an appropriate reestablishment of the connection. The SCM may inform the operator about this step. The only exception to the general rule of following the SCM state all the time is the state "Recording Completion" where a deviation of the SCC state is required (see definitions in 4.4.5.2).

4.5.3.5 Incorrect SCC behaviour

If an SCC does not change its SCC state to disabled or enabled within the timeout definitions of E.1, the SCM shall delete the SCD from volatile memory and NACK all SC messages from this SCC including the SCClientStatus message (see definitions in F.3) and not send any messages to this SCC until the SCC reports the correct SCC state to which it was commanded.

4.5.3.6 Unexpected SCC state change to disabled

If the active SCM detects an unexpected SCC status change to disabled, then the SCM shall delete the SCD of this SCC from volatile memory to ensure an appropriate reestablishment of the connection and NACK the SC messages of this SCC. The active SCM responsible for the sequence integrity shall take appropriate action when this SCC is part of the selected sequence(s) to ensure the related client functions get taken out of the sequence(s). This may require operator interaction.

4.6 Communication strategy

In the communication between SCM and SCC, it is possible that a response to a command is not received because of transient communication problems. The failure may happen during the command message (i.e. the SCM does not receive the client command), or the failure may happen during the response message (i.e. the SCC does not receive the response of the SCM). The initiator of the command cannot distinguish between these two cases and will repeat the command to get the needed behaviour.

Therefore, the use of TANs is required for some message pairs defined in Annex E to ensure that repeated commands can be detected and responses can be related to the original command (see definitions in 4.4.5.2 and 4.4.7). The sender shall use the same TAN as before when repeating a command and the receiver shall use the same TAN as received with the command for its response.

Additionally, this TAN is needed to ensure that a certain response message can always be associated to the original command. Therefore, if an SCM or SCC receives a response message, it shall first make sure the TAN matches the original command from itself. If there is a positive TAN match, the SCM/SCC shall ensure that the client function ID and state ID does also match. Only if the TAN, client function ID and state ID match, shall the SCM/SCC process this message; otherwise, it shall ignore the message.

The usage of TANs allows that multiple client commands may be active simultaneously. For example, one SCC may have more than one SCExecutionIndication (see definitions in E.4) active at a given time, when the completion of one command is still pending while the operator activates already the next command.

Each participant in the SC communication has its own TAN counter, which will be increased by one for each new client command transmission initiated (e.g. during indication of a client function execution while recording or for commands to execute client functions during play back).

Annex A
(normative)

Sequence control data definition

A.1 General

A.1.1 Overview

This annex defines the details of the SCD to be loaded into the SCM by each SCC during system initialization. The SCD includes the definition of all recordable functions of a particular SCC, including references to describing graphics and text for each function for the visualization of the sequences stored in the SCM.

The object definitions are similar to the object usage in ISO 11783-6, including the SCD transfer from the SCC to the SCM by means of the ISO 11783 transport protocol and/or extended transport protocol.

All used object IDs shall be unique inside the whole SCD within the number range of 0 to 65534, while 65535 ($FFFF_{16}$) is reserved for use as the "NULL" object ID.

Each SCC shall introduce its individual SCD to the active SCM. Dependencies between multiple devices or CFs on the network need to be handled within the working set; perhaps only one of them may represent the working set as the SCC and communicate with the SCM accordingly.

A.1.2 Nomenclature

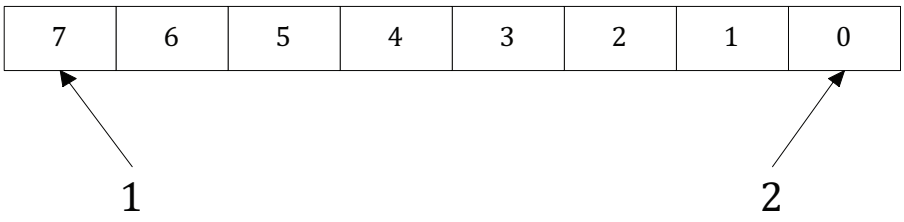
The following data types and nomenclature are used in the object definitions in Annex A.

Bitmask A set of logical bit values. Size is 1 byte. Bitmasks always have Bit 0 defined as the least significant bit. (See Figure A.1.)

Byte Array A sequence of 1 byte unsigned integer values of a defined length.

Float IEEE 754-1985 standard 32-bit floating point numeric value. Size is 4 bytes.

Integer Signed or unsigned integer numeric value. Possible sizes are 1, 2 or 4 bytes.



Key
1 most significant
2 least significant

Figure A.1 — Bit positions in a bitmask

A.2 SCD object relationship diagram

Figure A.2 illustrates the relationship between the SCD objects and referenced VT objects.

References to VT objects are used for designators and graphical representation of the individual SCD objects. The SCD objects contain the object ID used in the referenced SCCOP. The SCM uses these references to fill the external object pointers in its own VT object pool. The VT utilizes the referenced objects in the screen layouts of the SCM accordingly.

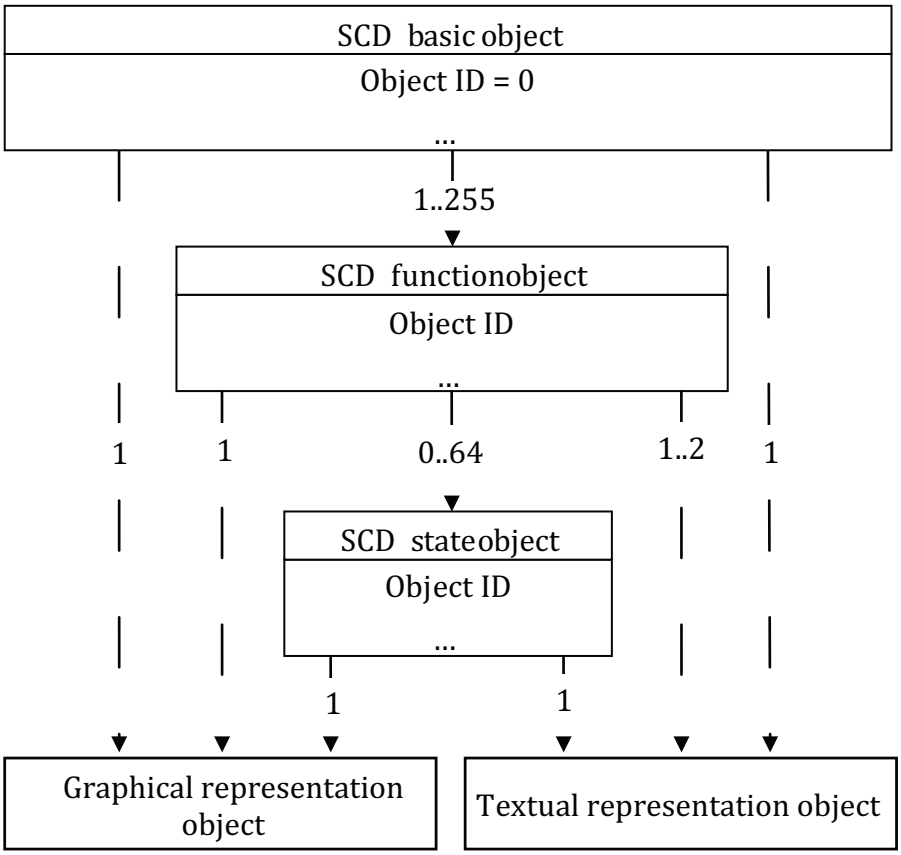


Figure A.2 — SCD object relationship diagram

A.3 SCD basic object

The SCD basic object (see Table A.1) is the root object of the SCD object pool. Only a single basic object is allowed for each SCD.

Table A.1 — SCD basic object

Attribute name	Type	Size (byte)	Value / Range	Record byte	Description
Object ID	Integer	2	=0	1-2	Unique identifier of this object. Object ID = 0 for the basic object
Type	Integer	1	=0	3	Object type = 0 for basic object
SC Version	Integer	1	0-2	4	Version of SC standard supported by SCC 0 = DIS 1 = FDIS 2 = IS
SCCWS Master NAME	Integer	8		5-12	The NAME of the Working Set Master owning the Object Pool in the VT holding the referenced graphics and texts.
SCC NAME Mask	Integer	8		13-20	Mask used by the SCM to filter the connected SCCs by their NAME information to identify the matching SCD when stored in the memory of the SCM. Bit = 1 means this bit of the NAME shall match. Bit = 0 means this bit of the NAME may be ignored.
SCD Designator	Integer	2	0-65534	21-22	Object ID of textual representation object in the referenced SCCOP containing an SCC description
SCD Graphic	Integer	2	0-65534	23-24	Object ID of graphical representation object in the referenced SCCOP containing the graphical representation of the SCC
SCD Version Label Length (N)	Integer	1	N = 1-64	25	Length definition of the following SCD Version Label in number of bytes.
SCD Configuration Label Length (M)	Integer	1	M = 1-64	26	Length definition of the following SCD Configuration Label in number of bytes.
Number of function objects to follow (O)	Integer	1	O = 1-255	27	Number of SCD function objects (see definitions in A.4) to follow
SCD Version Label	Byte Array	1-64		28 - 28+N-1	SCD version label Pad with spaces as necessary to satisfy size attribute.
SCD Configuration Label	Byte Array	1-64	0-255 for each byte	28+N - 28+N+M-1	SCD configuration label Pad with spaces as necessary to satisfy size attribute.
Repeat: Object ID	Integer	2	0-65534 for each object	28+N+M - 28+N+M+(O*2)-1	List of SC object IDs of associated SCD function objects (see definitions in A.4)

The SCM shall store the NAME of the SCC owning the SCD together with the SCD. This is not part of the SCD itself. The SCC NAME Mask included in the SCD basic object, will be used by the SCM to identify the SCC matching a stored SCD during next start-up(s). The SCC NAME Mask allows the SCC manufacturer to define how strict the SCM may filter for the same system configuration during next start-up. The SCM can be forced to expect the exact same implement, for instance, by setting all mask bits to 1 (all bytes of SCC NAME Mask = FF_{16}). But it can also allow the usage of one machine out of a fleet of identical machines (e.g. only the identity field in the NAME would be different and these bits of the NAME Mask would be set to 0). However, the SCM may be stricter in its NAME filtering than the SCC defines in its NAME Mask.

The SCCWS Master NAME allows the use of SC in a master member configuration where the SCC is not the working set master. This field holds the NAME of the working set master owning the object pool in the VT providing the graphics and texts referenced in the SCD. Usually, this field holds the NAME of the SCC itself, only when the SCC is a member without its own OP, then it holds the NAME of its working set master.

The SCD Version Label is intended to be used by the SCC to identify the version of the SCD stored in the SCM from previous usage (e.g. software version of the matching SCC).

The SCD Version Label may be displayed to an operator and may be used as part of a file name. As such, the SCC shall apply the following rules. SCD Version Labels shall be constructed of visible characters from font type ISO/IEC 8859-1 (Latin 1) character set. SCD Version Labels shall be padded with trailing blanks to produce a character string of the length defined in the SCD Version Label Length field. In addition, the following characters shall not be used in an SCD Version Label string:

\	[$5C_{16}$]	Reverse Solidus (Back slash)
“	[22_{16}]	Quotation mark (Double quote)
‘	[27_{16}]	Apostrophe (Single quote)
`	[60_{16}]	Grave Accent (Back tic)
/	[$2F_{16}$]	Solidus (Forward slash)
:	[$3A_{16}$]	Colon
*	[$2A_{16}$]	Asterisk
<	[$3C_{16}$]	Less-than sign
>	[$3E_{16}$]	Greater-than sign
|	[$7C_{16}$]	Vertical line
?	[$3F_{16}$]	Question mark

The SCD Configuration Label is intended to be used by the SCC to identify the machine configuration represented by the SCD stored in the SCM (e.g. installed options on the machine, etc.). The content of the SCD Version Label and SCD Configuration Label is proprietary to the manufacturer of the SCC. It is the task of the SCC to request and check the SCD Version Label and SCD Configuration Label of an SCD stored in an SCM before it is activated by the SCC. However, to ensure the integrity of SCSs, the SCM shall keep track of the SCD Version Label and SCD Configuration Label being used during the definition of an SCS. The length of these two label strings is limited to a maximum of 64 characters each to avoid a disproportional increase of the size of an SCS in memory.

The SCD Designator and SCD Graphic shall give some representative information about the SCC, since the SCM may use them to display the content of a certain sequence to the operator.

The Designator string is limited in its length by the constraints of the VT string object (see detailed definitions in ISO 11783-6), but the SCM may clip it to match the constraints of the SCM user interface (concerning, for example, display size or screen layout). The minimum length visible shall be 32 characters to ensure minimum information visible to the operator.

The SCCOP includes all layout information including font type, font size, font colour, etc. for the referenced textual representation objects. To ensure appropriate layouts, the SCM shall provide the desired font size, etc. through the SCMasterCapabilityResponse (see definition in C.3) to the SCCs to allow SCCOP adjustments.

A.4 SCD function object

The SCD function object provides all information regarding the client functions supported by the SCC for the sequence control system. The activation of these client functions may be recorded as operator actions during the recording phase (the SCM stores the ID of the client function activated) and may be triggered by the SCM during the replay phases. It also includes the preferred trigger mode, which may be one specific mode only; this may be used by the SCM as the default trigger mode during recording phase. The graphical and text representation will be used by the SCM to represent the actions in its screen-layout (e.g. while displaying the content of the stored sequence to the operator). The different states supported by the SCD function object (see Table A.2) are defined in the referenced SCD state objects (see definitions in A.5).

Table A.2 — SCD function object

Attribute name	Type	Size (byte)	Value / Range	Record byte	Description		
Object ID	Integer	2	1-65534	1-2	Unique identifier of this object		
						0 =	reserved for SCD basic object (see A.3)
						65535 =	reserved
Type	Integer	1	=1	3	Object type = 1 for function object		
Function ID	Integer	1	0-255	4	Unique function ID used as reference in the CAN messages		
Preferred trigger	Integer	1	1, 2, 255	5	Preferred trigger mode setting		
						0 =	reserved
						1 =	time-based function to be executed when the time counter reaches the documented value
						2 =	distance-based function to be executed when the distance counter reaches the documented value
						3-254=	reserved
						255 =	don't care – no special trigger mode defined for this function
					Only one trigger mode may be the preferred one.		
Attributes	Bitmask	1	0-2	6	Additional attributes		
					Bit 0-1 =		Numeric function value presentation:
						00 =	the SCM shall not display the numeric function value associated to this function
						01 =	the SCM shall display the numeric function value associated to this function in its sequence edit screen (when such an SCreen is supported by the SCM)

Table A.2 *(continued)*

Attribute name	Type	Size (byte)	Value / Range	Record byte	Description		
						10 =	the SCM shall display and allow the operator to change the associated numeric function value with the given value limitations of the SCD function object in its sequence edit screen (when such an SCreen is supported by the SCM)
						11 =	reserved
					Bit 2-7 =	reserved (set to zero)	
Numeric Function Value Min	Integer	4	0-[2^32-1]	7-10	Raw minimum value for the input of the numeric function value before scaling. Offset and scaling shall be applied to determine the actual minimum value. Set to $FFFFFFFF_{16}$ when the numeric function value presentation in the attributes is set to 00 or 01.		
Numeric Function Value Max	Integer	4	0-[2^32-1]	11-14	Raw maximum value for the input of the numeric function value before scaling. Offset and scaling shall be applied to determine the actual maximum value. Set to $FFFFFFFF_{16}$ when the numeric function value presentation in the attributes is set to 00 or 01.		
Numeric Function Value Offset	Signed Integer	4	–2^31 to [2^31-1]	15-18	Offset to be applied to the numeric function value for display (32-bit signed integer). Set to $FFFFFFFF_{16}$ when the numeric function value presentation in the attributes is set to 00.		
Numeric Function Value Scale	Float	4		19-22	Scale to be applied to the value for display. Set to $FFFFFFFF_{16}$ when the numeric function value representation in the attributes is set to 00.		
Numeric Function Value number of decimals	Integer	1	0-7	23	Specifies the number of decimals to display after the decimal point. Set to FF_{16} when the numeric function value representation in the attributes is set to 00.		
Numeric Function Value Unit String	Integer	2	0-65534	24-25	Object ID of referenced textual representation object containing the unit string to be displayed with the numeric function value. Set to $FFFF_{16}$ when the numeric function value representation in the attributes is set to 00.		
Designator	Integer	2	0-65534	26-27	Object ID of textual representation object in the referenced SCCOP containing the name label of the function		
Graphic representation	Integer	2	0-65534	28-29	Object ID of graphical representation object in the referenced SCCOP containing the graphical representation describing the function		

Table A.2 *(continued)*

Attribute name	Type	Size (byte)	Value / Range	Record byte	Description
Number of state objects to follow	Integer	1	0-64	30	Number of SCD state objects (see definitions in A.5) to follow
Repeat: Object ID	Integer	2	0-65534	31...	List of SC object IDs of SCD state objects (see definitions in A.5) defining the supported states of this function

The Function ID is used for the communication of the function over the ISO 11783 network to keep the message data field length within the maximum of 8 bytes in the related CAN messages. The field size of one byte limits the maximum number of functions that can be defined in one SCD to 255.

The Preferred Trigger Modes allow the SCC manufacturer to define which trigger mode makes the most sense for the operation of this client function. However, it is not mandatory for the SCM to support all trigger modes and it is not mandatory for the SCM to respect this Preferred Trigger Mode.

The SCC may send a numeric function value with each completion of a function during recording, which the SCM shall store with this function activation in its sequence. The SCM shall send this value with the activation message, during the replay phase, back to the SCC. This allows the SCC to store a specific process value with each activation of a client function (e.g. the height of the tractor hitch). The SCC defines in its SCD function object if and how this function value may be displayed or in which range it may be manipulated by the operator in the sequence overview screen; when such an SCreen is supported by the SCM.

The stored and transmitted numeric function value is always a 32-bit unsigned integer but the displayed value is scaled and formatted according to the following equations:

Displayed Value = (Value Attribute + Offset) * Scaling Factor

Transmitted Numeric Function Value = (Displayed Value / Scaling Factor) – Offset = Value Attribute

Scaled Upper Limit = (Upper Limit + Offset) * Scaling Factor

Scaled Lower Limit = (Lower Limit + Offset) * Scaling Factor

Lower Limit < = value attribute < = Upper Limit

Scaled Lower Limit < = Displayed value < = Scaled Upper Limit

Each SCD function object may reference multiple SCD state objects (see definitions in A.5) representing the different states of the function; e.g. for a sprayer master valve, the states could be on and off. The function sprayer master valve would be represented by its designator and graphic representation and the two states would be represented by the associated SCD state objects (see definitions in A.5).

The SCM may provide a means in its sequence edit screen to change the state manually; this could be an input list object based on the designators or graphic representation of the referenced SCD state object (see definitions in A.5).

The designator string is only limited in its length by the constraints of the VT string object (see definitions in ISO 11783-6), but the SCM may clip it to match the constraints of the SCM user interface (concerning for example display size or screen layout). The minimum length visible shall be 32 characters to ensure minimum information visible to the operator.

The SCCOP includes all layout information including font type, font size, font colour, etc. for the referenced textual representation objects. To ensure appropriate layouts, the SCM shall provide the desired font size, etc. through the SCMasterCapabilityResponse (see definition in C.3) to the SCCs to allow for font adjustments accordingly.

A.5 SCD state object

Each SCD state object (see Table A.3) defines one state of the referencing SCD function object (see definitions in A.4).

The graphical and text representation will be used by the SCM to represent the state of the associated client function in its screen layout (e.g. while displaying the content of the stored sequence to the operator), if such an SCreen is supported by the SCM.

Table A.3 — SCD state object

Attribute name	Type	Size (byte)	Value / Range	Record byte	Description
Object ID	Integer	2	1-65534	1-2	Unique identifier of this object 0 = reserved for SCD basic object (see A.3) 65535 = reserved
Type	Integer	1	=2	3	Object type = 2 for state object
State ID	Integer	1	0-63	4	Used to reference this state in CAN messages
Designator	Integer	2	0-65534	5-6	Object ID of textual representation object in the referenced SCCOP containing the name label of the state
Graphic representation	Integer	2	0-65534	7-8	Object ID of graphical representation object in the referenced SCCOP containing the graphical representation of the state

The State ID is used for the communication of the state over the ISO 11783 network to keep the message data field length within the maximum of 8 bytes of a single message. The field size of six bits limits the maximum number of states per function to 64. State ID 0 to 63 is always associated to the parent function ID.

One SCD state object definition may be referenced by multiple SCD function objects within an SCD while the SCM creates separate entities to track the individual states.

The designator string is only limited in its length by the constraints of the VT string object (see definitions in ISO 11783-6), but the SCM may clip it to match the constraints of the SCM user interface (concerning for example display size or screen layout). The minimum length visible shall be 32 characters to ensure the minimum information visible to the operator.

The SCCOP includes all layout information including font type, font size, font colour, etc. for the referenced textual representation objects. To ensure appropriate layouts, the SCM shall provide the desired font size, etc. through the SCMasterCapabilityResponse (see definition in C.3) to the SCCs to allow for font adjustments accordingly.

Annex B
(normative)

Message definition

B.1 Sequence control CAN messages

Two PGNs are reserved for the SC communication between the SCM and SCCs: SCM to SCC; SCC to SCM.

The default priority is set to 4 to ensure that other, higher priority messages are not disturbed by the SC communication (e.g. the virtual terminal auxiliary control system might be used to overwrite SC automatic functions).

B.2 SCM to SCC

Transmission repetition rate:	As required
Data length:	variable (minimum 8 bytes)
Data page field:	0
PDU format field:	142
PDU specific field:	Destination address
Default priority:	4
Parameter group number:	36352 ($0x008E00_{16}$)

B.3 SCC to SCM

Transmission repetition rate:	As required
Data length:	variable (minimum 8 bytes)
Data page field:	0
PDU format field:	141
PDU specific field:	Destination address
Default priority:	4
Parameter group number:	36096 ($0x008D00_{16}$)

Annex C
(normative)

Technical data messages

C.1 General

The technical data messages are used to request the characteristics of the SCM. They consist of the request for data by the SCC and the response by the SCM. Messages are sent using the PGNs given in Annex B.

C.2 SCMasterCapabilityRequest

The SCMasterCapabilityRequest is sent by the SCC to request the capabilities of the active SCM before the SCD gets uploaded. The SCC communicates its supported SC Standard version to allow SCM cross checking.

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	90_{16}	Message code
Byte 2:		Version of SC Standard supported by SCC
00_{16}		DIS
01_{16}		FDIS
02_{16}		IS
03_{16} - FF_{16}		Reserved
Bytes 3 - 8:		Reserved, transmit as FF_{16}

C.3 SCMasterCapabilityResponse

The SCMasterCapabilityResponse shall be sent by the SCM as a reply to the SCMasterCapabilityRequest (see definitions in C.2). These data are used by the SCCs to adjust their SCD before it gets uploaded to the

SCM and to configure the SCCOP loaded in the VT to match the SCM's object sizes in respect of graphic object size or font attributes. This message requires the use of a transport protocol defined in ISO 11783-3.

Transmission repetition rate:		In response to SCMasterCapabilityRequest (see definitions in C.2)
Data length:		10 Byte
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	90_{16}	Message code
Bytes 2:		Version of SC Standard supported by SCM
00_{16}		DIS
01_{16}		FDIS
02_{16}		IS
03_{16} - FF_{16}		Reserved
Byte 3:		Number of pixels on the X axis for graphics objects (minimum 32 pixels)
Byte 4:		Number of pixels on the Y axis for graphics objects (minimum 32 pixels)
Bytes 5, 6		Maximum number of displayed characters for designator strings (minimum of 32 is required)
Byte 7:		Font colour (see ISO 11783-6, Font Attributes Object)
Byte 8:		Font size (see ISO 11783-6, Font Attributes Object)
Byte 9:		Font type (see ISO 11783-6, Font Attributes Object)
Byte 10:		Font style (see ISO 11783-6, Font Attributes Object)

To foster reuse of objects from the SCCOP, the SCM shall report the soft key size of the used VT for the size of graphic objects (bytes 3 and 4).

C.4 SCMasterMemoryRequest

The SCMasterMemoryRequest is sent by the SCC to determine if the SCM has enough volatile memory available before the SCD gets uploaded.

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	91_{16}	Message code
Bytes 2 - 5:		Volatile memory required, number of bytes of the SCD to be uploaded
Bytes 6 - 8:		Reserved, transmit as FF_{16}

C.5 SCMasterMemoryResponse

The SCMasterMemoryResponse shall be sent by the SCM as a reply to the SCMasterMemoryRequest (see definitions in C.4).

Transmission repetition rate:		In response to SCMasterMemoryRequest (see definitions in C.4)
Data length:		8 bytes
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	91_{16}	Message code
Byte 2:		Status
	00_{16}	There can be enough volatile memory. However, because there is overhead associated with object storage it is impossible to predict whether there is enough memory available.
	01_{16}	There is not enough volatile memory available. Do not transmit SCD or send load command to load SCD in volatile memory.
	02_{16} - FF_{16}	Reserved
Bytes 3 - 8:		Reserved, transmit as FF_{16}

Annex D
(normative)

SCD operation messages

D.1 General

SCD operation messages are used during system configuration. They consist of requests of the SCCs with responses of the active SCM as defined in the following clauses. Messages are sent using the PGNs given in Annex B.

There are no specific timeouts defined for the individual responses, since the needed memory handling within the SCM is very much dependent on the size of for instance the SCD involved. It is important that the SCCs respect the busy flags in the SCMasterStatus message (see definitions in F.2) during determination of their timeouts.

The SCM may provide functions to store and to restore a complete SCD. The availability and organization of the non-volatile storage area is SCM-specific. Storing and restoring an SCD includes all object definitions (see definitions in Annex A). If the SCM provides the possibility to store SCDs, there shall be a method inside the SCM to assign a stored SCD uniquely to an SCC. There is only one SCD per SCC allowed to be stored in non-volatile memory.

D.2 SCDVersionRequest

The SCDVersionRequest is used by the SCC during start-up to check if the active SCM has a stored SCD of the requesting SCC available and if the version of the SCD matches the SCC version. The SCM will respond with the SCDVersionResponse (see definitions in D.3).

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	01_{16}	Message code
Bytes2 - 8:		Reserved, transmit as FF_{16}

D.3 SCDVersionResponse

The SCDVersionResponse shall be sent by the active SCM as response to an SCDVersionRequest (see definitions in D.2) of an SCC. If the error code (byte 2) indicates no error, this message will include the complete SCD version label string of the stored SCD pool of the requesting SCC. Depending on the string

length, the use of transport protocol may be necessary. In case of any error code other than zero, there is no version data included and the frame length is set to 8 data bytes with data bytes 4 to 8 sent as FF_{16}.

Transmission repetition rate:		In response to SCDVersionRequest (see definitions in D.2)
Data length:		Variable
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	01_{16}	Message code
Byte2:		Error code
	00_{16}	No error – version string attached
	01_{16}	No matching SCD stored – no version string attached
	02_{16} - FE_{16}	Reserved
	FF_{16}	Any other error – no version string attached
Byte 3		Length of version label string
	Bits 1-6	Number of characters (bytes) to follow (0 if byte 2 > 00_{16})
	Bits 7-8	Reserved, transmit as 0
Bytes 4 - 8:	$FFFFFFFFFF_{16}$	When error code > 00_{16}
Bytes 4 - N:		Actual version label, when error code = 00_{16}

D.4 SCDConfigurationLabelRequest

The SCDConfigurationLabelRequest is used by the SCC to check if the active SCM has a stored SCD of the requesting SCC available and if the configuration stored in the SCD matches the current configuration of the SCC. The SCM will respond with the SCDConfigurationLabelResponse (see definitions in D.5).

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	02_{16}	Message code
Bytes2 - 8:		Reserved, transmit as FF_{16}

D.5 SCDConfigurationLabelResponse

The SCDConfigurationLabelResponse shall be sent by the active SCM as response to an SCDConfigurationLabelRequest (see definitions in D.4) of an SCC. If the error code indicates no error, this message will include the complete SCD configuration label string of the stored SCD of the requesting SCC. Depending on the string length, the use of transport protocol may be necessary. In case of any error

code other than zero, there is no configuration data included and the frame length is set to 8 data bytes with data bytes 4 to 8 sent as FF_{16}.

Transmission repetition rate:		In response to SCDConfigurationLabelRequest (see definitions in D.4)
Data length:		Variable
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	02_{16}	Message code
Byte2:		Error code
	00_{16}	No error – configuration label string attached
	01_{16}	No matching SCD stored – no configuration label string attached
	02_{16} - FE_{16}	Reserved
	FF_{16}	Any other error – no configuration label string attached
Byte 3:		Length of configuration label string
	Bits 1-6	Number of characters (bytes) to follow (0 if byte 2 > 00_{16})
	Bits 7-8	Reserved, transmit as 0
Bytes 4 - 8:	$FFFFFFFFFF_{16}$	When error code > 00_{16}
Bytes 4 - N:		Actual configuration label, when error code = 00_{16}

D.6 SCDTransfer

The SCDTransfer is used by the SCC to transfer its SCD to the active SCM.

This message requires the use of a transport protocol defined in ISO 11783-3. If the data length exceeds 1785 bytes, the extended transport protocol used in ISO 11783-6 shall be used.

Transmission repetition rate:		On request
Data length:		Variable
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	03_{16}	Message code
Bytes 2 - N:		SCD data bytes

D.7 SCDTransferResponse

Although the transfer of an SCD pool is handled by a transport protocol service with its own confirmation and error handling, the SCM shall provide this confirmation message to indicate success or failure of

SCD pool reception. If no error is indicated, the SCD pool was correctly received by the SCM and may be further used.

Transmission repetition rate: In response to SCDTransfer (see definitions in D.6)

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1:	03_{16}	Message code
Byte 2:		Error code
	00_{16}	No error – SCD was correctly received
	01_{16}	Error occurred during transfer
	02_{16} - FE_{16}	Reserved
	FF_{16}	Any other error
Bytes 3 - 8:		Reserved, transmit as FF_{16}

D.8 SCDActivationCommand

The SCDActivationCommand shall be sent by the SCC to activate its SCD after it was either successfully transmitted (no errors indicated in D.7 SCDTransferResponse) or loaded (see definitions in D.14 SCDLoadCommand). Receiving this command will initiate the parsing of the SCD by the SCM. The SCM shall indicate the ongoing parsing in the busy flags of its SCMasterStatus message (see definitions in F.2). The SCM shall respond with the SCDActivationResponse (see definitions in D.9) indicating the SCD integrity.

Transmission repetition rate: On request

Data length: 8 bytes

Parameter group number: SCC to SCM, destination-specific

Byte 1:	04_{16}	Message code
Bytes2 - 8:		Reserved, transmit as FF16

D.9 SCDActivationResponse

The SCDActivationResponse is sent by the SCM after completion of the SCD parsing to indicate the SCD integrity in the error code field of this message. Any error code other than 0 indicates that the SCD may not be used by the system.

Transmission repetition rate: In response to SCDActivationCommand (see definitions in D.8)

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1:	04_{16}	Message code
Byte 2:		Error code
	00_{16}	No error – SCD was correctly processed
	01_{16}	Unknown object type
	02_{16}	Basic object does not exist
	03_{16}	Referenced function object does not exist
	04_{16}	Referenced state object does not exist
	05_{16}	Object ID is not unique
	06_{16}	Function ID is not unique
	07_{16}	State ID is not unique
	08_{16}	Designator string length is zero
	09_{16}	SCD version label length incorrect (either zero or >63 byte)
	$0A_{16}$	SCD version label contains not allowed characters (see definitions in A.3)
	$0B_{16}$	SCD configuration label length incorrect (either zero or >63 byte)
	$0C_{16}$	Version of SC standard supported by SCC is not supported by SCM (see definitions in A.3)
	$0D_{16}$ - FE_{16}	Reserved
	FF_{16}	Any other error
Bytes 3 - 4:		Object ID of first faulty object (= $FFFF_{16}$ for error code 00_{16})
Bytes 5 - 8:		Reserved, transmit as FF_{16}

D.10 SCDDeleteCommand

The SCDDeleteCommand will be sent by the SCC to delete an existing SCD out of the SCM memory. It may also be used to delete a partially sent SCD when the connection to the SCM gets cancelled due to memory constraints at the SCM side or similar.

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	05_{16}	Message code
Byte 2:		Command type
	00_{16}	Delete SCD in volatile memory
	01_{16}	Delete SCD in non-volatile memory
	02 - FF_{16}	Reserved
Bytes 3 - 8:		Reserved, transmit as FF_{16}

D.11 SCDDeleteResponse

The SCDDeleteResponse is sent by the SCM after completion of deleting an SCD.

Transmission repetition rate:		In response to SCDDeleteCommand (see definitions in D.10)
Data length:		8 bytes
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	05_{16}	Message code
Byte 2:		Error code
	00_{16}	No error – SCD was correctly deleted
	01_{16}	SCD does not exist
	02 - FF_{16}	Reserved
	FF_{16}	Any other error
Bytes 3 - 8:		Reserved, transmit as FF_{16}

D.12 SCDStoreCommand

The SCDStoreCommand will be sent by an SCC to the active SCM to initiate the storing of the SCD in non-volatile memory of the SCM once the activation and parsing of the SCD was successful. The SCM shall ensure

that only one SCD of a particular client is stored in non-volatile memory by using the NAME mask information in the SCD basic object (see definitions in A.3) accordingly. Older versions of the SCD will be overwritten.

Transmission repetition rate: On request

Data length: 8 bytes

Parameter group number: SCC to SCM, destination-specific

Byte 1: 06_{16} Message code

Bytes 2 - 8: Reserved, transmit as FF_{16}

D.13 SCDStoreCommandResponse

The SCDStoreCommandResponse is sent by the SCM after completion of the SCD storage in non-volatile memory. Any error code other than 0 indicates that the SCD could not be stored, nevertheless the SCD may be used in volatile memory, but has to be uploaded with next system startup again.

Transmission repetition rate: In response to SCDStoreCommand (see definitions in D.12)

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1: 06_{16} Message code

Byte 2: Error code

- 00_{16} No error – SCD was successfully stored
- 01_{16} SCD was not completely parsed yet
- 02_{16} Not enough non-volatile memory available
- 03_{16} - FE_{16} Reserved
- FF_{16} Any other error

Bytes 3 - 8: Reserved, transmit as FF_{16}

D.14 SCDLoadCommand

The SCDLoadCommand will be sent by an SCC to the active SCM to initiate the loading of the SCD available in non-volatile memory of the SCM into volatile memory (see definitions in 4.4.2).

Transmission repetition rate: On request

Data length: 8 bytes

Parameter group number: SCC to SCM, destination-specific

Byte 1: 0716 Message code

Bytes 2 - 8: Reserved, transmit as FF16

D.15 SCDLoadResponse

The SCDLoadResponse is sent by the SCM after completion of loading the SCD from non-volatile into volatile memory. Any error code other than 0 indicates that the SCD could not be loaded.

Transmission repetition rate:		In response to SCDLoadCommand (see definitions in D.14)
Data length:		8 Byte
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	071_6	Message code
Byte 2:		Error code
	00_{16}	No error – SCD was successfully loaded
	01_{16}	SCD not found
	02_{16}	File system error or SCD data corruption
	03_{16}	Not enough volatile memory available
	$0416 - FE_{16}$	Reserved
	FF_{16}	Any other error
Bytes 3 - 8:		Reserved, transmit as FF_{16}

Annex E
(normative)

Sequence control messages

E.1 General

The sequence control messages defined in this annex are used during sequence recording and playback. Messages are sent using the PGNs given in Annex B.

All commands or requests defined in this annex are followed by a response from the other communication partner, either in a dedicated response message or in a status message defined in Annex F. Details are defined for each individual command or response in the following clauses.

The commanding or requesting CF shall repeat its command or request with a repetition rate up to 5 times per second until the response is received or no response could be received for a timeout of at least 1.6 s. A minimum of 100 ms shall be ensured between the repeated commands or requests.

E.2 SCStateCommand

The SCStateCommand is used by the active SCM to command an SCC to a certain state (enabled or disabled). This, for instance, is used during system configuration, prior to a recording phase or after activating a new sequence to keep only the SCCs enabled which participate in the selected sequence to reduce the complexity of the system configuration and bus load. The state of an SCC may only be commanded by an active SCM or by the device itself (e.g. by operator input at the available user interface of the device itself).

The state change will be reflected in the state bits of the SCClientStatus message (see definitions in F.3) of the function; therefore, no dedicated response message is defined.

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	30_{16}	Message code
Byte 2:		Commanded SCC state
	00_{16}	Disabled
	01_{16}	Enabled
	02_{16} - FF_{16}	Reserved
Byte 3:	FF_{16}	Reserved (to keep error code in all messages in the same byte)
Byte 4:		Reason indicator
	00_{16}	Reserved
	01_{16}	Status change on operator command

	02_{16}	SCC not part of selected SCS
	03_{16}	SCC is part of selected SCS
	04_{16} - FE_{16}	Reserved
	FF_{16}	No specific reason
Bytes 5 - 8:		Reserved, transmit as FF_{16}

E.3 SCClientAbort

The SCClientAbort may be initiated by any enabled SCC during play back phase only, to indicate any error condition requiring the immediate termination of the current sequence processing.

The active SCM shall reflect the sequence state "Abort" in its SCMasterStatus message (see definitions in F.2) as response to this SCClientAbort.

The active SCM and all enabled SCCs shall halt the SC processing immediately after receiving this SCClientAbort command. SCCs may enter a safe state and wait for further operator instructions. However, the SCMasterStatus message (see definitions in F.2) and SCClientStatus message (see definitions in F.3) are continued.

A halted play back may not be restarted without dedicated operator confirmation to the active SCM. The SCM shall provide provisions for this task.

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCC, global
Byte 1:	31_{16}	Message code
Bytes 2 - 3:		Reserved (to keep sequence number and error code in the same byte), transmit as FF16
Byte 4:	FF_{16}	Error code
	00_{16}	Reserved
	01_{16}	Timeout on status message
	02_{16}	Operator interaction
	03_{16}	Error during function execution
	04_{16} - FE_{16}	Reserved
	FF_{16}	Any other error
Bytes 5 - 8:		Reserved, transmit as FF16

E.4 SCExecutionIndication

The SCExecutionIndication is sent on client function activation during "Recording" or "Play Back" only. Only enabled SCCs are allowed to send this message.

Transmission repetition rate: On request

Data length: 8 bytes

Parameter group number: SCC to SCM, destination-specific

Byte 1: 32_{16} Message code

Byte 2: Function ID as defined in SCD function object (see definitions in A.4)

Byte 3: State ID and function execution state

Bits 1-6: State ID as defined in the SCD state object (see definitions in A.5)

Bits 7,8: Function execution state

- 00 = function execution started and completed simultaneously
- 01 = function execution started
- 10 = function execution completed
- 11 = error during execution or operator interaction during playback

Byte 4: TAN

Bytes 5 - 8: $FFFFFFFF_{16}$ When the function execution state is function execution started only

Bytes 5 - 8: Function value when the function execution completed bit is set (see definitions in 4.4.8)

E.5 SCExecutionIndicationResponse

The SCExecutionIndicationResponse is sent by the active SCM as response to any SCExecutionIndication (see definitions in E.4) to the related SCC. The function ID, state ID and function execution state shall reflect the information sent with the SCExecutionIndication (see definitions in E.4). The error codes, other than 00_{16}, indicate the SCM could not store the client command in the active SCS. The TAN shall be the same as received in the SCExecutionIndication (see definitions in E.4) being responded to.

Transmission repetition rate: In response to SCExecutionIndication (see definitions in E.4)

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1: 32_{16} Message code

Byte 2: Function ID as defined in SCD function object (see definitions in A.4)

Byte 3: State ID and function execution state

Bits 1-6: State ID as defined in SCD state object (see definitions in A.5)

Bits 7,8 Function execution state

- 00 = function execution started and completed simultaneously
- 01 = function execution started
- 10 = function execution completed
- 11 = error during execution or operator interaction during playback

Byte 4: Error code

00_{16}	No error, client command stored in the SCS
01_{16}	Function ID not defined
02_{16}	State ID not defined
03_{16}	Function execution state mismatch
04_{16}	SCM out of memory
05_{16}	Play back continues, no further commands will be sent to this client function (only used during play back and if both function-execution-bits are set to 1)
06_{16}	Play back halted due to operator interaction (only used during play back and if both function execution bits are set to 1)
07_{16} - FE_{16}	Reserved
FF_{16}	Any other error

Byte 5: TAN

Bytes 6 - 8 Reserved, transmit as FF_{16}

E.6 SCMasterExecutionCommand

The SCMasterExecutionCommand is used by the SCM to trigger SCC client functions during the play back phase when the associated trigger condition (e.g. time or distance) is reached. The SCM sends the command directly to the function owning SCC, which has to perform the client function accordingly. The triggered SCC shall acknowledge the start and completion of this command with an SCClientExecutionStatus (see definitions in E.7 and 4.4.7).

Transmission repetition rate: On request

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1: 3316 Message code

Byte 2: Function ID as defined during recording

Byte 3: State ID as defined during recording

Bits 1 - 6 State ID

Bits 7 - 8 Reserved, transmit as 0

Byte 4: TAN

Bytes 5 - 8: Function value as defined during recording (see definitions in 4.4.8 Function value)

E.7 SCClientExecutionStatus

The SCClientExecutionStatus will be sent by the SCC as response to any SCMasterExecutionCommand (see definitions in E.6) received from the SCM to confirm the reception and execution of the command (see4.4.7 Sequence play back).

The TAN shall be the same as received in the SCMasterExecutionCommand (see definitions in E.6) being responded to.

Transmission repetition rate: In response to SCMasterExecutionCommand (see definitions in E.6)

Data length: 8 bytes

Parameter group number: SCC to SCM, destination-specific

Byte 1: 34_{16} Message code

Byte 2: Function ID as defined during recording

Byte 3: State ID and function execution state

Bits 1-6 State ID as defined in the SCD state object (see definitions in A.5)

Bits 7,8 Function execution state

00 = function execution started and completed simultaneously

01 = function execution started

10 = function execution completed

11 = error during execution

Byte 4: Error code

00_{16} No error

01_{16} Error in the function

02_{16} Operator interaction

03_{16} Conditions of SCC do not allow the execution

04_{16} - FE_{16} Reserved

FF_{16} Any other error

Byte 5:	TAN
Bytes 6 - 8	Reserved, transmit as FF_{16}

E.8 SCClientExecutionAcknowledgement

The SCClientExecutionAcknowledgement is sent by the SCM to acknowledge the reception of the SCClientExecutionStatus (see definitions in E.7) of the SCC executing a client function during Play Back (see definitions in 4.4.7 Sequence play back).

The TAN shall be the same as received in the SCClientExecutionStatus (see definitions in E.7) being responded to.

Transmission repetition rate: In response to SCClientExecutionStatus (see definitions in E.7)

Data length: 8 bytes

Parameter group number: SCM to SCC, destination-specific

Byte 1:	34_{16}	Message code
Byte 2:		Function ID as defined during recording
Byte 3:		State ID and function execution state
	Bits 1 - 6	State ID as defined in the SCD state object (see definitions in A.5)
	Bits 7 - 8	Function execution state acknowledgement

00 = acknowledgement of simultaneous function execution start and completion

01 = acknowledgement of function execution started

10 = acknowledgement of function execution completed

11 = acknowledgement of error during execution

Byte 4:	TAN
Bytes 5 - 8:	Reserved, transmit as FF_{16}

E.9 SCClientFunctionErrorRequest

The SCClientFunctionErrorRequest is used by the SCM to request the actual status of faulty client functions. The SCCs report in their status message if one or multiple client functions are not available

for use, while the SCM uses this message to request the details from the individual SCCs. The SCC shall respond with the SCClientFunctionErrorResponse (see definitions in E.10).

Transmission repetition rate:		On request
Data length:		8 bytes
Parameter group number:		SCM to SCC, destination-specific
Byte 1:	35_{16}	Message code
Bytes 2 - 8:		Reserved, transmit as FF_{16}

E.10 SCClientFunctionErrorResponse

The SCClientFunctionErrorResponse is sent by the SCC in response to the SCClientFunctionErrorRequest (see definitions in E.9) from the active SCM. The SCC shall report the client function IDs of the faulty client functions, together with an error code indicating the error state of the faulty client function (2 bytes per function). Depending on the number of active errors, this message may require the use of TP. In case of less than three faulty client functions being sent, the message shall be padded with FF_{16} up to 8 data bytes.

The assumption is that the SCC will usually inform the operator themselves when a client function is not available or faulty. However, if the SCC does not have its own user interface or for other reasons, the SCC may request from that the SCM inform the operator about the not available client function (function error structure byte b bit 8). The SCM shall prompt the operator immediately after receiving a function error structure with the operator confirmation required bit set. The SCM may use information like designators or graphical representation out of the related SCD for the operator information.

On transmission of this message, the SCC shall reset the ClientFunctionErrorState to 01_{16} in its SCClientStatus message (see definitions in F.3) until the status of one or multiple client functions changes again.

Transmission repetition rate:		In response to SCClientFunctionErrorRequest (see definitions in E.9)
Data length:		Variable
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	35_{16}	Message code
Byte 2:	0-255	Number of function error structures to follow
Bytes 3 - 8:	$FFFFFFFFFFFF_{16}$	When the number of function error structures to follow = 0
Bytes 3 - N:		List of function error structures as defined below
	Function error structure	
	Byte a	function ID of faulty function object as defined in SCD function object (see definitions in A.4)
	Byte b	error status of client function:
	Bits 1 - 4	Error status indicator
		00_{16} Reserved (do not report functions without error)

01_{16}	One time error
02_{16}	Permanent error
03_{16}	Client function is locked
04_{16}	Client function deactivated at the SCC by the operator
05_{16} - $0E_{16}$	Reserved
$0F_{16}$	Any other error

Bits 5 - 7 Reserved, transmit as 0

Bit 8 1 = operator confirmation required

Annex F
(normative)

Status messages

F.1 General

The status messages allow the SCC to determine the state of the SCM and vice versa.

Messages are sent using the PGNs given in Annex B.

F.2 SCMasterStatus message

This message is sent by the active SCM as a broadcast message to all CFs. It is sent immediately on state change of any of bytes 2 to 5, once per second during "Ready" state and 5 messages per second during active "Recording", "Recording Completion", "Play Back" or "Abort" state. A minimum of 100 ms shall be ensured between the individual SCMasterStatus message.

Inactive SCMs shall not allow the change of their state while another SCM is indicating its active state in its status message; the active SCM shall not allow the change of its status while a recording or play back of a sequence is ongoing.

Inactive SCMs shall send their SCMasterStatus message only once directly after start-up indicating the initialization and once directly after a state change (operator input) with the status inactive.

Transmission repetition rate:		On change of any of byte 2 to 5; 5 times per second during sequence states 02_{16}, 03_{16}, 04_{16}, and 05_{16}; once per second during all other sequence states in byte 4.
Data length:		8 bytes
Parameter group number:		SCM to SCCs, broadcast
Byte 1:	95_{16}	Message code
Byte 2:		State of sending SCM
	00_{16}	Inactive
	01_{16}	Active
	02_{16}	Initialization (once at start-up)
	03_{16} - FF_{16}	Reserved
Byte 3:		Sequence number
	00_{16} - 31_{16}	Number of selected sequence
	32_{16} - FE_{16}	Reserved
	FF_{16}	When byte 2 is set to inactive or when sequence state = Ready

Byte 4:	Sequence state	
	00_{16}	Reserved
	01_{16}	Ready
	02_{16}	Recording
	03_{16}	Recording completion
	04_{16}	Play back
	05_{16}	Abort
	06_{16}-FE_{16}	Reserved
	FF_{16}	When byte 2 is set to inactive
Byte 5	Busy flags	
	Bit 1	1 = SCM is busy with non-volatile memory access
	Bit 2	1 = SCM is parsing an SCD
	Bits 3 - 8	Reserved, transmit as 0
Bytes 6 - 8:	Reserved, transmit as FF_{16}	

A timeout of 600 ms for the SCMasterStatus message shall be applied in the “Recording”, “Recording Completion”, “Play Back” or “Abort” state. A timeout of 3 s shall be applied in the “Ready” state.

The busy flags allow the SCM to indicate to the SCCs that the response to requests may be delayed. SCCs shall pause their timeout measurement for pending responses from the SCM while one of these flags is set.

F.3 SCClientStatus message

The SCClientStatus message is sent by all SCCs supporting the sequence control concept while they detect an active SCM on the ISO 11783 network. It is sent immediately on state change of any of the bytes 2 to 5, once per second during the “Ready” state or if SCC is disabled, and 5 messages per second during the active “Recording”, “Recording Completion”, “Play Back” or “Abort” state. A minimum of 100 ms shall be ensured between the individual SCClientStatus messages.

The SCC shall indicate once its initialization right after start-up to ensure that the SCM detects such quick start-up immediately.

Receiving a NACK of the SCClientStatus message from the active SCM is an indication that the SCM is not synchronized with the SCC. The SCC has to re-establish the connection to the active SCM.

The sequence number (byte 3) shall be used to indicate the actual sequence number for which the SCC is sending an SCExecutionIndication (see definitions in E.4) or for which the SCC is actually executing an SCMasterExecutionCommand (see definitions in E.6). The sequence number in the SCClientStatus message is intended to be used for diagnostic purposes only.

The SCCs shall indicate with the ClientFunctionErrorState (byte 5) at any given time if one or multiple of its client functions are not available for use in a sequence or if the error state of any of its client functions

changed since the last detailed report to the SCM to allow the SCM to request the status of the individual client functions with the SCClientFunctionErrorRequest (see definitions in E.9 and E.10).

Transmission repetition rate:		On change of any of byte 2 to 5, 5 times per second during sequence states 02_{16}, 04_{16}, and 05_{16}; once per second during all other sequence states in byte 4
Data length:		8 bytes
Parameter group number:		SCC to SCM, destination-specific
Byte 1:	96_{16}	Message code
Byte 2:		State of sending SCC
	00_{16}	Disabled
	01_{16}	Enabled
	02_{16}	Initialization (once at start-up)
	03_{16} - FF_{16}	Reserved
Byte 3:		Sequence number
	00_{16} - 31_{16}	Number of active sequence
	32_{16} - FE_{16}	Reserved
	FF_{16}	When byte 2 is set to disabled or when sequence state is “Ready”
Byte 4:		Sequence state
	00_{16}	Reserved
	01_{16}	Ready
	02_{16}	Recording
	03_{16}	Reserved
	04_{16}	Play back
	05_{16}	Abort
	06_{16}-FE_{16}	Reserved
	FF_{16}	When byte 2 is set to disabled or initialization
Byte 5		ClientFunctionErrorState
	00_{16}	No errors
	01_{16}	Error state of the available client functions did not change since last report to SCM

02_{16}	State of one or more client functions changed since last report to SCM. This gets set when the faulty function does not require the SCM to take action (e.g. SCC does its own warning as in manual mode or the function is not that critical).
03_{16}	State of one or more client functions requiring operator confirmation changed since last report to SCM (see definitions in 4.5). This state has priority over state 2 meaning that this remains set even when a function requiring a state 2 becomes active after the previous state 3 was requested.
04_{16} - FE_{16}	Reserved
FF_{16}	When byte 2 is set to disabled

Bytes 6 - 8: Reserved, transmit as FF_{16}

A timeout of 600 ms for the SCClientStatus message shall be applied in the “Recording”, “Play Back” or “Abort” state. A timeout of 3 s shall be applied in the “Ready” state.

Bibliography

[1] SAE J1939, *Recommended Practice for a Serial Control and Communications Vehicle Network*

[2] AEF. (Ag Industry Electronics Foundation, http://www.aef-online.org), guidelines on implementation of ISO11783 under consideration of relevant safety aspects

[3] ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No.*

ICS 65.060.01;35.240.99

Price based on 65 pages