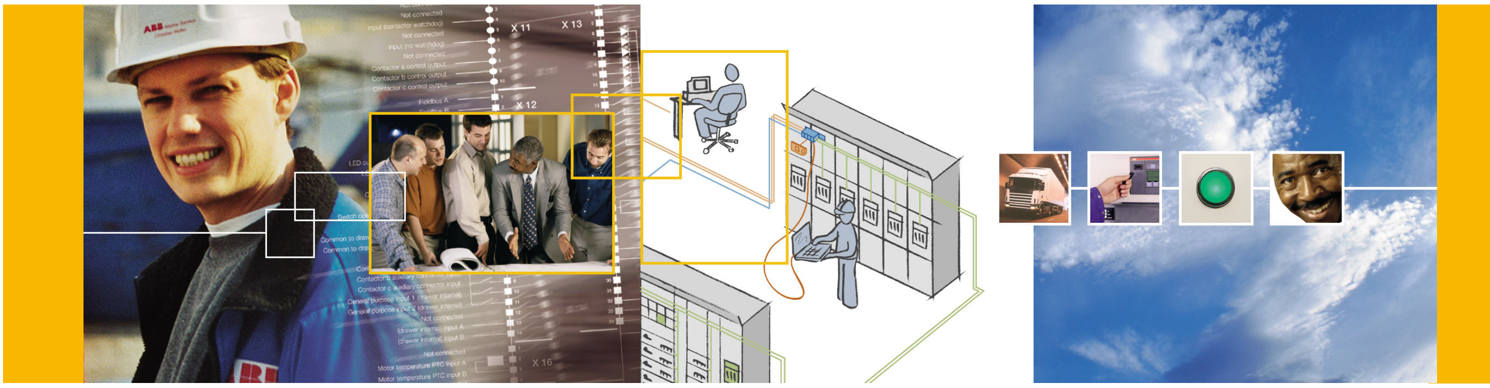
Protect^{IT} – MNS Motor Management INSUM®

Control Access Guide
Version 2.3



ABB

INSUM®
CONTROL ACCESS (CA) Guide

Version 2.3

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Schaltanlagentechnik GmbH. ABB Schaltanlagentechnik GmbH assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Schaltanlagentechnik GmbH be liable for direct, indirect, special, incidental, or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Schaltanlagentechnik GmbH be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Schaltanlagentechnik GmbH's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose. Permission to translate the document shall be obtained from ABB Schaltanlagentechnik GmbH. The translated document shall be sent to ABB Schaltanlagentechnik GmbH together with the confirmation that the content of the document is the same.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

© 2002 ABB Schaltanlagentechnik GmbH, Germany

TRADEMARKS

MNS and INSUM are registered trademarks of ABB Schaltanlagentechnik GmbH

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

Echelon, LON, LONWORKS, LonTalk, Neuron are trademarks of Echelon Corporation registered in U.S. and other countries.

Reference document 1TGB330002 R3.1

INSUM®
CONTROL ACCESS (CA) Guide

Notes:

1 Introduction

1.1 Objective

This document describes the function of the Control Access within the INSUM system. As a further guide-line an application example is given. It shall help the user to understand the functionality.

1.2 Related Software Version

This guide describes the CA functionality for INSUM 2.3.

1.3 Related Documentation

1TGC 901007 B0201 INSUM Technical Information
1TGC 901021 M0201 INSUM MCU Users Guide
1TGC 901026 M0201 INSUM MCU Parameter Description
1TGC 901034 M0201 INSUM MMI Operating Instruction
1TGC 901030 M0201 INSUM MMI Quick Guide
1TGC 901042 M0201 INSUM Modbus Gateway Manual
1TGC 901052 M0201 INSUM Profibus Gateway Manual
1TGC 901060 M0201 INSUM Ethernet Gateway Manual
1TGC 901080 M0201 INSUM System Clock Manual
1TGC 901091 M0201 INSUM Failsafe Guide
1TGC 901092 M0201 INSUM Dual Redundancy Guide
1TGC 901093 M0201 INSUM Network Management Guide
SACE RH 0080 Rev.I PR112/ PD-L LON Works Interface V2.0
1SEP 407948 P0001 Users Manual Intelligent Tier Switch (ITS)

Notes:

2 Functional Description of Control Access

2.1 Purpose of Control Access

Control Access defines the operation authority of ICU devices connected to the INSUM bus. Without CA all ICU devices are equal with their rights to control a field unit (MCU)

Field Units such as MCUs can be controlled by commands (START, STOP, RESET) coming via bus or commands coming via hardware inputs. There are two scenarios to define the interlocking between bus and hardware commands.

- Without Control Access all ICU devices are equal and a simple way of interlocking between commands coming via bus or hardware is the use of the hardware switch LOCAL/REMOTE. In case the switch is in position LOCAL commands coming via bus are ignored. In case the switch is in position REMOTE only commands via bus are executed. The MCU does not evaluate from which "bus device" a command was received. Bus devices in this sense are SU devices like GWs (DCS), MMI or OS.

- With Control Access function the INSUM system can be set up in such a way that the interlocking mechanism takes also into account which of the SU devices sent a command. The MCU accepts commands via bus only from one of the SU devices **at a time**. Control Access also covers a mechanism to pass the "right to send commands" from one SU device to the other. For this purpose priorities are assigned to the SU devices.

2.2 Rules of Control Access

The Control Access functionality in INSUM is based on the following rules and requirements:

- Operation authority can be defined separately for each MCU
- At a time t0 one MCU can be controlled from one ICU device only
- The device with the highest priority can transfer the CA to one with lower priority
- The device with higher priority can request CA from one with lower priority
- To get the operation authority back, a device with a higher priority passes the CA to itself
- Passing of CA to devices with higher priority is not possible
- MCU accepts only to pass the CA to devices available in the SULifeList
- MCU knows the current CA owner and accepts commands only from the CA owner
- If a CA owner „dies“ then the operation authority will be released by the MCU

Note:

CA currently supported by:	MODBUS GW, Ethernet GW, MMI, OS
CA currently not supported by:	PROFIBUS GW
CA request not supported by:	MMI, INSUM OS

2.3 Structure of Control Access

2.3.1 MCU Control Access Table

The Motor Control Unit (MCU) provides a "Control Access Table". In the "Control Access Table" the hierarchy between all *stations* is stored. Each *station* is assigned a priority in this table. All bus-devices are referenced by their LON-address, the others are fixed.

The contence of this table has to be the same in all MCU's connected to the same backplane (SU-unit). The priorities correspond to bit representation. The maximum number of priorities is 14 whereas only 12 bus-stations are possible. A SU device can only be assigned to one Address priority.

Notes:

Table 1. Control Access Table in MCU

Bit ref.		Priorities	Domain/Subnet/Node address of SU-device	example filling	remark
0	highest priority	Address priority 1:	LOCAL-HW	0/0/0	fixed
1	↓	Address priority 2:	Address of SU-device	0/5/10	parametrizable
2	↓	Address priority 3:	Address of SU-device	0/5/11	parametrizable
3	↓	Address priority 4:	Address of SU-device	0/5/30	parametrizable
4	↓	Address priority 5:	Address of SU-device	0/5/20	parametrizable
5	↓	Address priority 6:	Address of SU-device	0/5/21	parametrizable
6	↓	Address priority 7:	Address of SU-device	0/0/0	parametrizable
7	↓	Address priority 8:	Address of SU-device	0/0/0	parametrizable
8	↓	Address priority 9:	Address of SU-device	0/0/0	parametrizable
9	↓	Address priority 10:	Address of SU-device	0/0/0	parametrizable
10	↓	Address priority 11:	Address of SU-device	0/0/0	parametrizable
11	↓	Address priority 12:	Address of SU-device	0/0/0	parametrizable
12	↓	Address priority 13:	Address of SU-device	0/0/0	parametrizable
13	lowest priority	Address priority 14:	LOCAL-SW	0/0/0	fixed
14	--	--	--	0/0/0	not used
15	--	--	--	0/0/0	CA bit

Implementation Note for parameter value file of MCU:
- Each entry out of the 16 Address Table Entries which is not assigned to a LON-address has to be filled with 0/0/0.
- If existing in the parameter set the Address Table entries for Priority 0, 13, 14, 15 must be 0/0/0

2.3.2 SULifeList

General

The SULifeList indicates which of the SU devices are "alive". It is received by all MCU's belonging to one system.

(If the actual CA-owner dies, the CA is set to "RELEASE" by the MCU. "RELEASE" means that MCU does not execute any control command. CA can only be assigned to SU-devices being "alive". In case of RELEASE the "CAPass" is granted from every *station*. -> see following chapters).

The SULifeList is a 16-bit structure (SNVT_state) where only Bit1...Bit12 are used. Each ICU device is assigned one of these bits analog to the Control Access Table. The bit is "1" when the device is "alive" and "0" when the device is not "alive" on the LON bus. Bits which are not used have to be set to "0".

The generation and distribution of the SULifeList:

All SU devices send a "nvoLifesign" to each other (the Watcher). This NV is also a 16-bit structure. Each SU device sets the bit corresponding to its own address priority (CA-Priority) in the Lifesign structure. All SU devices receive the Lifesign from all other SU devices. The combination of all these Lifesign-Bits results in the SULifeList. The SU device with the highest priority in the SULifeList is in charge to send it (as broadcast) to all MCU's. If the SU device with the highest priority "dies", the SU device then having the highest priority in the List takes over sending the SULifeList.

The Lifesign is sent with heartbeat rate defined by parameter "SU lifesign heartbeat". Supervision of this Lifesign heartbeat follows parameter "SU lifesign timeout". SGCLifelist is cyclically sent according to parameter "SU lifelist heartbeat".

The MCU supervises the SGCLifeList according to parameter "SU LifeList timeout". If the "SU LifeList timeout" expires and no SGCLifeList was received the CA is RELEASED by the MCU.

Notes:

2.4 ActualCA1

Each MCU sends the ActualCA1 to all SU-devices. It reports which of the stations is "CA-owner" or whether CA is RELEASED for that particular MCU. In addition it reports whether CA is "enabled" or "disabled" in the MCU. The structure is as follows:

Table 2. ActualCA List sent by MCU

Bit	Description	Values
15	CA Enabled	Enabled = 1, Disabled = 0 (reflects MCU parameter)
14	Reserved	Bit = 0
13	priority 14: LOCAL by CAPass – Lowest priority	Bit = 1 -> MCU is switched by CAPass to LOCAL
12	CA-owner address priority 13	Bit = 1 -> SU device with that priority is CA-owner
11	CA-owner address priority 12	"
10	CA-owner address priority 11	"
9	CA-owner address priority 10	"
8	CA-owner address priority 9	"
7	CA-owner address priority 8	"
6	CA-owner address priority 7	"
5	CA-owner address priority 6	"
4	CA-owner address priority 5	"
3	CA-owner address priority 4	"
2	CA-owner address priority 3	"
1	CA-owner address priority 2	"
0	priority 1: Status of MCU binary input LOCAL/REMOTE – Highest priority	Bit = 1-> status of MCU binary input is LOCAL Bit = 0 -> status of MCU binary input is REMOTE

Table 3. Explanation of bits

Meaning of Bits	ActualCA1	
Bit	**CA Enabled (Bit 15 = 1)**	**CA Disabled (Bit 15 = 0)**
Bits 0 .. 13 = 0, Bit14 = 0	CA = RELEASED MCU does not execute any control command. With "CAPass" the CA can be assigned again to one of the *stations*. (for more details refer to chapters below)	CA = BUS all devices connected to the "BUS" (GW, MMI, OS...) are allowed to control the motor.
Bit0 = 1, Bit1...Bit13 = 0, Bit14 = 0	CA = LOCAL-HW	CA = LOCAL-HW
Bit13 = 1, Bit0...Bit12 = 0, Bit14 = 0	CA = LOCAL-SW	CA = LOCAL-SW
One out of Bit1...Bit12 = 1, Bit 0 = 0, Bit13 = 0, Bit14=0	BUS-device with respective CA priority is now "CA-owner".	-- (not a valid combination!)

Note:
Other combinations of bits in nvoActualCA1 are not allowed !

Notes:

2.5 CAPass

Each SU-device has a command output "nvoCAPass" for each of the MCU's. It is used when a SU-device wants to pass CA to itself or to another *station* for a particular MCU. It is a 16 Bit structure where Bit1...Bit13 correspond to the station priorities as shown for ActualCA1 (refer to chapter 2.4). CAPass can also be used to set CA to RELEASED state. The CAPass command is only granted by MCU if the rules defined in chapter 2.5.1 apply.

The structure is as follows:

Table 4. CAPass List

Bit	CA priority	Description	Code hex
15	--	Bit = 0 (reserved)	
14	--	Bit = 0 (reserved)	
13	14	Bit = 1 -> CAPass to LOCAL (MCU binary inputs)	0x2000
12	13	Bit = 1 -> CAPass for SU-device priority 13	0x1000
11	12	Bit = 1 -> CAPass for SU-device priority 12	0x0800
10	11	Bit = 1 -> CAPass for SU-device priority 11	0x0400
9	10	Bit = 1 -> CAPass for SU-device priority 10	0x0200
8	9	Bit = 1 -> CAPass for SU-device priority 9	0x0100
7	8	Bit = 1 -> CAPass for SU-device priority 8	0x0080
6	7	Bit = 1 -> CAPass for SU-device priority 7	0x0040
5	6	Bit = 1 -> CAPass for SU-device priority 6	0x0020
4	5	Bit = 1 -> CAPass for SU-device priority 5	0x0010
3	4	Bit = 1 -> CAPass for SU-device priority 4	0x0008
2	3	Bit = 1 -> CAPass for SU-device priority 3	0x0004
1	2	Bit = 1 -> CAPass for SU-device with priority 2	0x0002
0	1	Bit = 0 (reserved)	
		All Bits = 0 -> CA released	0x0000

Table 5. Explanation of bits

Meaning of Bits	CAPass	
Bit	**CA Enabled**	**CA Disabled**
Bits 0 .. 15 = 0,	command to set CA = RELEASED	command to set CA = BUS
Bit13 = 1, Bit0...Bit12 = 0, Bit14 .. Bit15 = 0	command to set CA = LOCAL-SW	command to set CA = LOCAL-SW
One out of Bit1...Bit12 = 1, Bit 0 = 0, Bit13 = 0, Bit14 .. Bit15 = 0	command to set CA for BUS-device with respective priority	--

Notes:

2.5.1 Rules for CA Passing (CA = ENABLED)

All Rules defined in this chapter are valid if the Control Access is enabled.

2.5.1.1 Role of MCU

The MCU is the instance which evaluates control commands. Therefore the MCU takes into consideration the source address of a control command and the address of the actual "CA-owner". In case the source of the command and the CA belong to the same *station* the command is passed to the motor.

Control commands in this sense are START (all different starts of all drive types), STOP and RESET.

The MCU is the instance which evaluates CAPass. After having received a CAPass the actual CA-owner is changed by the MCU according to the rules defined below. They depend on:

- the current CA-owner,
- the hierarchy set up in the Control Access Table
- the contence of SULifeList.

The MCU receives the SULifeList in defined time intervals telling which of the SU-devices are "alive" on the bus. In case the actual "CA-owner" "dies", MCU sets the ActualCA1 to RELEASE. This means that the MCU does not execute any control command. With "CAPass" the CA can be assigned again to one of the remaining *stations*.

2.5.1.2 Description of Passing

1. The *station* owning the CA or a *station* above of it in the hierarchy can cause the changeover of CA to all *stations* which are below of it in the hierarchy.

For this a special message has to be sent to the MCU with a structure as follows:

- The source address is available in each LON telegram.
- The data is the value/contence of the network variable.

Source address	Data
station owning CA or with higher priority	station to which to pass CA

2. *Stations* which are not in the SULifeList anymore cannot receive CA. If a *station* owns CA and is removed out of the SULifeList the ActualCA will be set to RELEASE state by the MCU.

3. In case CA is RELEASED, any station can do a *CAPass* to itself or another station below of it in the hierarchy.

4. A *station* owning CA can set the ActualCA to RELEASE state by sending a CAPass with contence 0x0000. The *station* then claiming for CA first will get CA.

The following examples show some scenarios of passing CA between *stations*.

Notes:

2.5.2 Examples for "CAPass" messages

In Table 6 below the sequential order of the *stations* in the Control Access Table gives the highest priority inside SU-devices to GW1. In this case the "Master-DCS" communicates via GW1 and therefore is the "Master" for passing CA. All other *stations* have a lower priority except LOCAL/REMOTE switch.

Table 6. Example setting for GW1 has address priority 2

priority	station	Control Access Table (MCU parameter) domain/subnet/node	SGCLifeList (provided as NV)	ActualCA (provided as NV)
1, highest	binary input LOCAL/REMOTE	--	--	NO_CA
2, ↓	GW 1	0/5/10	PRESENT	NO_CA
3, ↓	GW 2	0/5/11	PRESENT	NO_CA
4, ↓	OS	0/5/30	PRESENT	NO_CA
5, ↓	MMI 1	0/5/20	NOT_PRESENT	NO_CA
6, ↓	MMI 2	0/5/21	PRESENT	CA
...	...	...	...	...
14, lowest	--	--	--	NO_CA

Rule: The actual CA-owner can pass CA to another station below of it in the hierarchy.

Example: GW1 changes the CA from itself to MMI 2 with the following CAPass message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 2		5/10 (priority 2)	priority 6		priority 6

Rule: A station with higher priority than the actual CA-owner can pass CA to another station below of it in the hierarchy.

Example: GW 1 changes the CA from MMI 2 to OS with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 6		5/10 (priority 2)	priority 4		priority 4

Rule: A station with higher priority than the actual CA-owner can pass CA to itself

Example: GW1 changes the CA from GW2 to itself with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 3		5/10 (priority 2)	priority 2		priority 2

Rule: *Stations* which are not in the SGCLifeList anymore cannot receive CA

GW1 tries to change the CA from itself to MMI 1 with the following *CAPass* message. This *CAPass* is denied.

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 2		5/10 (priority 2)	priority 5		priority 2

Notes:

Rule: In case CA is RELEASED, any station can do a *CAPass* to itself.

Example: OS changes the CA to itself with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
RELEASE		5/30 (priority 4)	priority 4		priority 4

Rule: In case CA is RELEASED, any station can do a *CAPass* to another station below of it in the hierarchy.

Example: GW1 changes the CA from RELEASE to OS with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
RELEASE		5/10 (priority 2)	priority 4		priority 4

Rule: In case CA is RELEASED, a station cannot pass CA to a station above of it in the hierarchy.

Example: MMI 2 tries to change the CA from RELEASE to GW1 with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
RELEASE		5/21 (priority 6)	priority 2		RELEASE

Rule: A station owning CA can set CA to RELEASE

Example: GW2 changes the CA to RELEASE with the following CAPass message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
RELEASE		5/11 (priority 3)	RELEASE		RELEASE

Rule: A station not owning CA cannot set CA to RELEASE

Example: GW2 changes the CA to RELEASE with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 2		5/11 (priority 3)	RELEASE		priority 2

Rule: A station owning CA cannot pass CA to a station above of it in the hierarchy

Example: OS tries to pass the CA to GW1 with the following *CAPass* message:

Previous		"CAPass" message			Result
ActualCA	→	Source address	Data	→	ActualCA
priority 4		5/30 (priority 4)	priority 2		priority 4

Notes:

2.6 MCU binary inputs

Besides the commands via bus, the MCU can be controlled via binary inputs (START1, START2, STOP and RESET). The status of the binary input LOCAL/REMOTE defines whether the commands via binary inputs are accepted or not. LOCAL means that CA is assigned to the binary inputs. (commands via bus are not accepted).

2.6.1 LOCAL/REMOTE switch (highest priority)

2.6.1.1 CA = ENABLED

In the Control Access Table the "LOCAL/REMOTE switch" has the highest priority which is fixed.

If the binary input LOCAL/REMOTE is switched to LOCAL MCU automatically assigns CA to the binary inputs. CA is then given to "LOCAL-HW". In this case no other *station* which is a member in the Control Access Table can get CA because they all have a lower priority.

binary input LOCAL/REMOTE		ActualCA
position LOCAL	→	priority 1 = LOCAL-HW

If the switch changes to REMOTE CA is set to RELEASE.

binary input LOCAL/REMOTE		ActualCA
change to REMOTE	→	RELEASE

Now, any other *station* which is a member in the Control Access Table can claim for CA.

2.6.1.2 CA = DISABLED

If the binary input LOCAL/REMOTE is switched to LOCAL MCU automatically assigns CA to the binary inputs. CA is then given to "LOCAL-HW". In case the switch is put back to REMOTE, CA is given to BUS (all control commands via "bus" are executed).

2.6.2 CAPass to LOCAL (lowest priority)

2.6.2.1 CA = ENABLED:

In addition to CA-assignment when the switch is put to LOCAL, the CA can be passed with a "CAPass-message" to the binary inputs. In this case the binary inputs have the lowest priority (address priority 14) in the Control Access Table. This priority is fixed. The following message shows how CA can be passed from a SU-device to the binary inputs:

Previous		**"CAPass" message**			**Result**
ActualCA	→	Source address	Data	→	ActualCA
priority 6		priority 6	priority 14		priority 14 = LOCAL-SW

If CA is assigned to priority 14 (LOCAL-SW), all other *stations* can do a CAPass according to the rules defined above because they have a higher priority.

Note: It is recommended not to use "set CA" to LOCAL via hardware switch and via CAPass-message in parallel.

2.6.2.2 CA = DISABLED:

With CAPass message CA can be set to LOCAL-SW as follows:

Previous		**"CAPass" message**			**Result**
ActualCA	→	Source address	Data	→	ActualCA
BUS		Any BUS-device	priority 14		priority 14 = LOCAL-SW

Notes:

2.7 Failsafe

If the MCU goes to Failsafe, the CA is RELEASED, no matter which of the SU-devices was CA-owner before. In case the CA was put to LOCAL-HW or LOCAL-SW, the MCU ignores the Failsafe functionality (and therefore will not RELEASE the CA). If CA is then passed back to one of the SU-devices, the Failsafe functionality is active again.

Failsafe functionality "active" means that in case of communication loss the MCU swiches motor to parameterized failsafe mode (NOP, START, STOP...).

The following matrix applies for Failsafe:

Table 7. CA and Failsafe

ActualCA	**CA=ENABLED**
CA-owner = one of SU-devices (priority 2...13)	Failsafe functionality active. In case of Failsafe, CA is RELEASED.
CA-owner = LOCAL-HW or LOCAL-SW (priority 1 or priority 14)	Failsafe functionality not active.
CA=Released	Failsafe functionality active, CA stays RELEASED
ActualCA	**CA=DISABLED**
CA-owner = BUS	Failsafe functionality active. In case of Failsafe, CA is RELEASED.
CA-owner = LOCAL-HW or LOCAL-SW	Failsafe functionality not active.

2.8 CA and control commands (especially RESET)

Table 8. CA and control commands

CA=ENABLED	**START (according to drive type) STOP TOL Bypass group start group stop**	**RESET**
CA-owner = one of SU-devices (priority 2...13)	executed for CA-owner	executed for CA-owner
CA-owner = LOCAL-HW or LOCAL-SW (priority 1 or priority 14)	executed via binary inputs, not via BUS.	executed via binary inputs
CA = RELEASED	not executed	not executed

CA=DISABLED	**START (according to drive type) STOP**	**RESET**
CA-owner = BUS	executed for command via BUS	executed for command via BUS
CA-owner = LOCAL-HW or LOCAL-SW	executed via binary inputs, not via BUS.	executed via binary inputs and via BUS

Note: If the RESET command is executed the MCU takes also into account the parameter “Reset mode”.

Example: CA=LOCAL-HW, reset mode = BUS/Remote for respective protection function, the Reset command via binary inputs will not be performed.

Notes:

2.9 CA-Request

Each MCU provides a request input (nviCARequest) where all *stations* controlling this MCU can initiate a CA-Request to the SU-devices for that particular MCU.

After having received this request, the MCU passes it on to the SU-devices via its output nvoCARequestFb. The telegram of CARequestFb then contains the source address of the MCU for which CA is requested and the value of this nv contains the address priority of the *station* for which CA is requested

nviCARequest and nvoCARequestFb have a 16 Bit structure whereas Bit1...Bit13 are defined as for ActualCA. Handling of a Request is a task of the SU-devices, or in case of GW's the request handling has to be done by the connected DCS.

If a request is answered positive, a CAPass message for the requesting device has to be generated.

2.10 Transitions for MCU parameter CA Enable/Disable

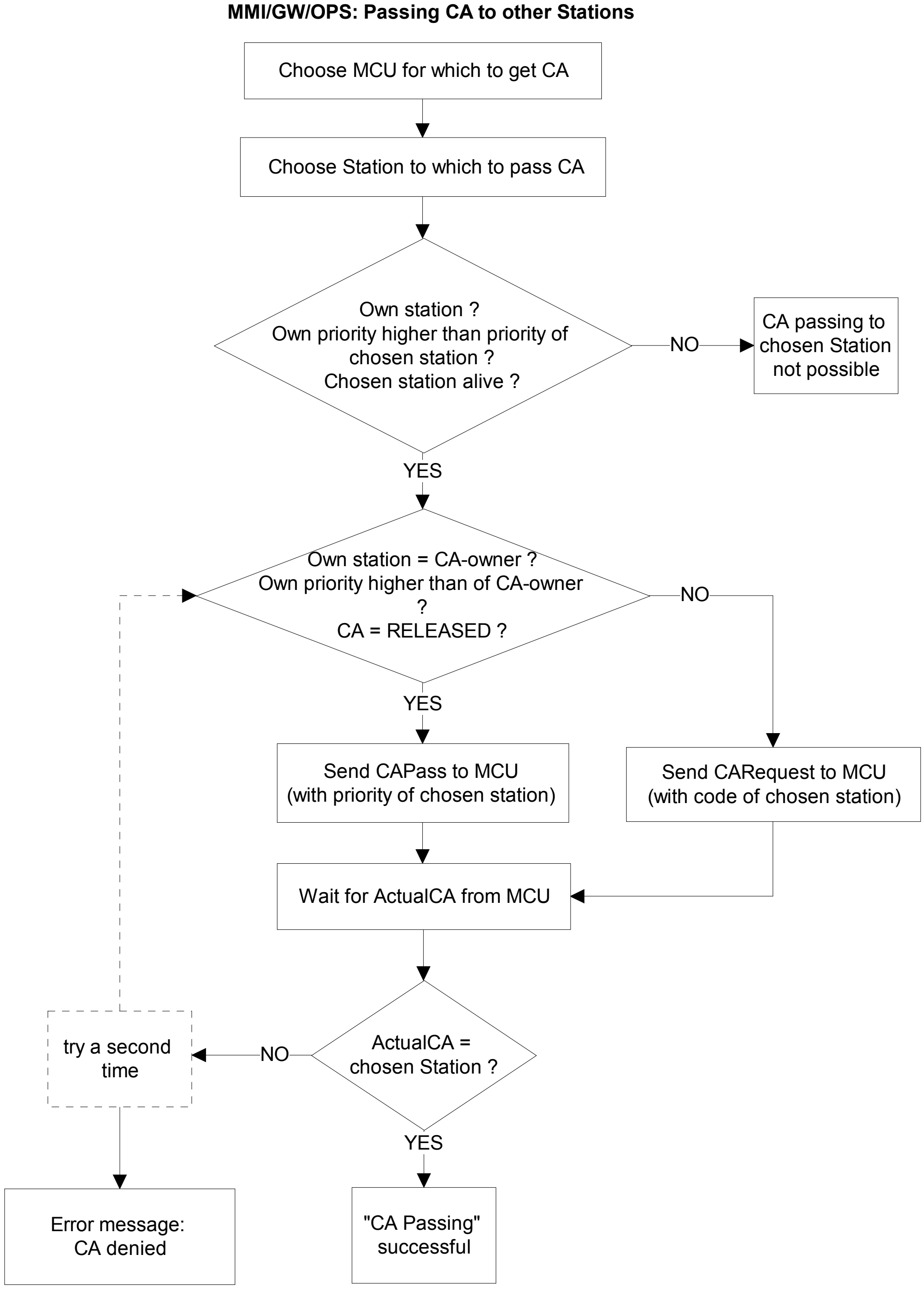
The following table shows how the MCU reacts if the CA function is enabled or disabled

Table 9. Transitions

Parameter CA Function	ActualCA	
enable -> disable	LOCAL-HW	-> LOCAL-HW
	LOCAL-SW	-> LOCAL-SW
	SU-device	-> BUS
disable -> enable	LOCAL-HW	-> LOCAL-HW
	LOCAL-SW	-> LOCAL-SW
	BUS	-> RELEASE

Notes:

2.11 Flow Chart CA Passing



Notes:

3 CA Parameters

3.1 SU devices (MMI, GW, OS):

All SU-devices have the same set of parameters used to configure Control Access. The setting has to be identical for all SU-devices connected to the same system. These parameters also have to match the MCU parameters for Control Access.

Table 10. Parameter List

Group	Name english	Range english	Default setting
SYSTEM	SU lifesign heartb:	1 (1) 60 sec	2 sec
SYSTEM	SU lifesign timeout:	1 (1) 100 sec	6 sec
SYSTEM	SU lifelist heartb.:	0: DISABLED 1 (1) 60 sec	DISABLED
SYSTEM	CA priority:	priority 2...priority 13 code Bit1...Bit12 =1	priority 2
SYSTEM	CA name:	10 characters	MOD GW

SU lifesign heartbeat:
Each SU-device sends cyclically a lifesign on the backbone-LON bus. This parameter defines the time interval at which the lifesign is to be sent repeatedly by the SU-device.

SU lifesign timeout:
Each SU-device supervises the lifesigns of the other SU-devices connected to the backbone- LON bus. If no lifesign was received during the time interval defined by this parameter the respective SU-device is removed out of the lifelist.

The following rule should be followed: SU lifesign timeout >= 3 * SU lifesign heartbeat

SU lifelist heartbeat:
The SU lifelist shows which of the SU-devices are "alive".

This parameter defines the time interval at which the SU Lifelist is to be sent repeatedly by the SU-device to the MCU's. It has to correspond with MCU parameter "CONTROL ACCESS/SU LifeList timeout".

The following rule should be followed: SU lifesign timeout >= 3 * SU lifesign heartbeat

CA priority:
This parameter defines the priority assigned to the SU-device inside the MCU-Control Access Table. The rules for passing CA between stations depend on this priority.

CA name:
This text is used to reference the SU-device in all MMI-menues dealing with Control Access.

Notes:

3.2 MCU

The setting of Control Access parameters has to be identical for all MCU's connected to the same system. These parameters also have to match the SU-device parameters for Control Access.

Table 11. Parameter List

MMI-Group	MMI-Name	MMI-Range write	Default setting
Control access	Function:	enabled disabled	disabled
Control access	Address priority 2 Address priority 3 Address priority 4 Address priority 5…13	Domain / Subnet / Node Domain: 0;1 Subnet: 0 (1) 255 Node: 0 (1) 127	2: 0 / 005 / 010 (GW) 3: 0 / 005 / 020 (MMI) 4: 0 / 005 / 030 (OS) 5-13: 0 / 000 / 000
Control access	SU Lifelist timeout:	1 (1) 255 s	20s

Function:

This parameter is used to chose between the 2 setups of Control Access:

"DISABLED" means CA differentiates between

- LOCAL-HW
- LOCAL-SW
- BUS (all SU devices have the same rights)

"ENABLED" means CA differentiates between

- LOCAL-HW
- LOCAL-SW
- SU-DEVICES (only one of the connected SU device is allowed to control the motor at a time)

Address priorities:

These parameters make up the "Control Access Table". If CA is ENABLED it defines the priorities between the SU-devices. The SU-devices are referenced by their LON address consisting of Domain/Subnet/Node whereas "Domain" is set to "0". The hierarchy is used to define the rules for "CA-Passing".

This set of parameters has to match the parameter "CA priority" in SU-devices.

SU LifeList timeout:

Each SU-device is assigned a bit in the SULifeList. The presence of a SU-device in this list is supervised according to the time interval defined by this parameter.

This parameter has to match the SU-parameter „SU lifesign timeout“ and "SU lifelist heartbeat".

The following rule should be followed: The following rule should be followed:

2 * SU lifesign timeout + 1 * SU lifelist heartbeat << SU lifelist timeout

Note: If LifeList sender "dies" the maximum switch over delay for next LifeList-sender is
2 * SU lifesign timeout + 1 * SU lifelist heartbeat

Notes:

4 Application Example to use CA

The below application example illustrates more on how to utilise the CA mechanism and implement it in the process control system (DCS).

4.1 Application

The application is derived based on the normal requirement of process control application wherein the INSUM system is interfaced with the plant process control system and there is a need to exercise the CA mechanism. Additionally to have the possibility to set the Local Control Mode on the MCU from DCS via the soft command, besides the hardwired Local Control Mode on the MCU.

The application example is aimed to have an optimised implementation in DCS and utilises only minimum signal exchange between DCS and INSUM. It is based on the ABB Control System Advant OCS. Some information herein requires minimum knowledge about the Advant system to understand abbreviations and names used.

The DCS interface uses the INSUM MODBUS Gateway, no details are given here about the MODBUS configuration.

4.1.1 Control Stations participating

- Local Control Station in the field (hardwired)
- MODBUS Gateway (represents DCS)
- INSUM Man Machine Interface Unit
- INSUM Operator Station
- Local Control activated from DCS (LOCAL soft)

4.1.2 Hierarchy of control

The hierarchy of participating control stations should normally be assigned based on project specific re-quirements.

As this example is about the implementation of CA mechanism in the DCS, the DCS is considered as the higest station in the CA hierarchy.

The below table shows the CA priorities assigned for the CA stations. But as per the philosophy of CA Mechanism the hardwired local control mode will have the default higest prority. The assignement of the CA priorities for the participating CA stations are as below.

CA Priorities	CA Stations
01	Local Control Station (Default)
02	MODBUS Gateway (DCS)
03	INSUM Man Machine Interface Unit
04	INSUM Operator Station
14	Local Mode (Soft) (fixed)

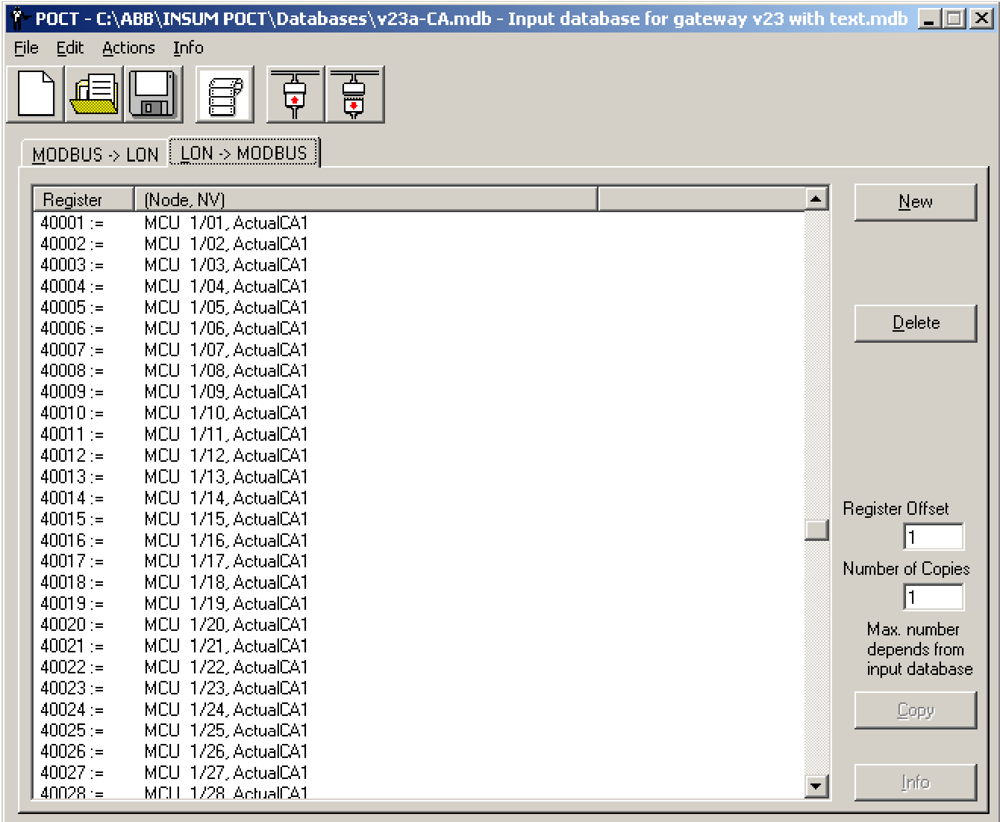
The DCS will have the following status information on Actual CA owner for each of the MCU via MODBUS Gateway. The bit set indicates the active mode = CA owner.

MODBUS Address	LON network variable	Status Description
4xxxx	nwActualCA1	CA owner: 0x0001 (hex) -> Local (hardwired) CA owner: 0x0002 (hex) -> for DCS CA owner: 0x0004 (hex) -> for MMI CA owner: 0x0008 (hex) -> for INSUM OS CA owner: 0x2000 (hex) -> for Local (soft)

Notes:

Example:

The table shows an example for MODBUS register setting of nvwActualCA1; the register can be set individually for any application with the help of the POCT tool.



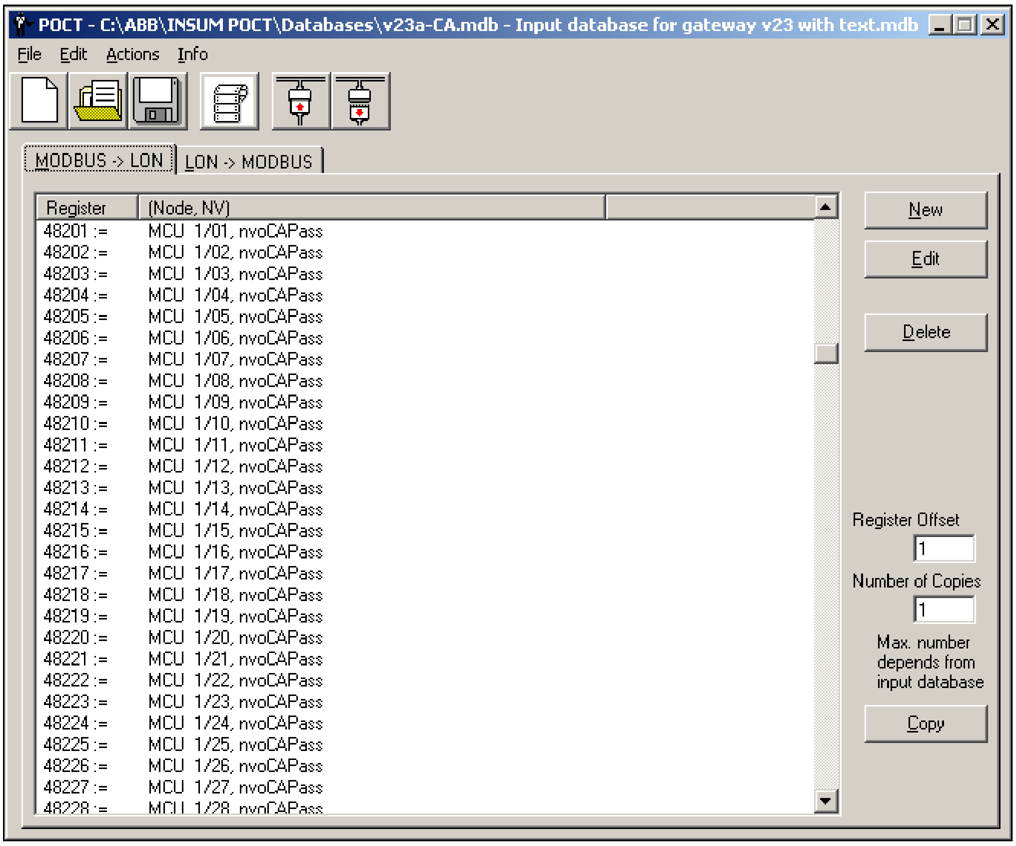
The DCS will authorise the CA to other monitoring stations i.e. MMI, INSUM OS, and LOCAL (soft) by sending the respective hex code (word register) to each of the MCU in the nvoCAPass network variable.

MODBUS Address	LON network variable	Description of codes
4xxxxx	nvoCAPass	CA priority 2: 0x0002 (hex) for DCS CA priority 3: 0x0004 (hex) for MMI CA priority 4: 0x0008 (hex) for OS CA priority 14:0x2000 (hex) for Local (soft)

Notes:

Example:

The table shows an example for MODBUS register setting of nvoCAPass; the register can be set individually for any application with the help of the POCT tool.



The DCS will utilise the status information of CA to indicate the CA owner of the MCU whereas the output word will be controlled to pass the control access of the MCU to the other monitoring station requesting the CA of the MCU.

4.2 CA handling and Indication control

The AC 450 utilises the MOTCON functional PC element as the motor control object. The CA handling and indication control is worked out around this element.

The main motor object display of ‘MOTCON’ will basically differentiate the mode as below
- Sequence (Remote): CA owner of MCU is DCS
- Local: CA owner of MCU can be any of the one from Local (hardwired), MMI, INSUM OS, or Local (soft)

An additional overview display, defined to show the detailed information supported by INSUM will show the exact indication for Actual CA owner of MCU when in Local.

Note: The indication in DCS should be **acknowledged based**.

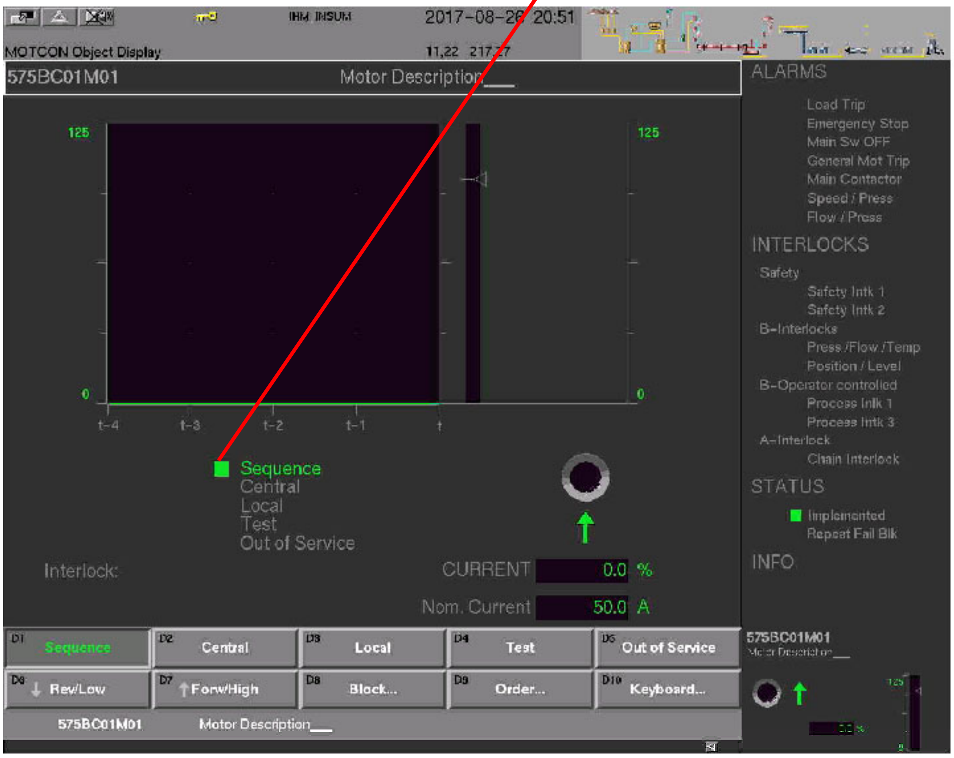
4.2.1 Operation

The MOTCON will utilise the order handling commands on the main dialogue menu of the motor object as below.
- Sequence: To order and indicate the CA owner as DCS. i.e. the AC450 controller.
- Local: To pass the CA to Local (soft) station and indication as Local

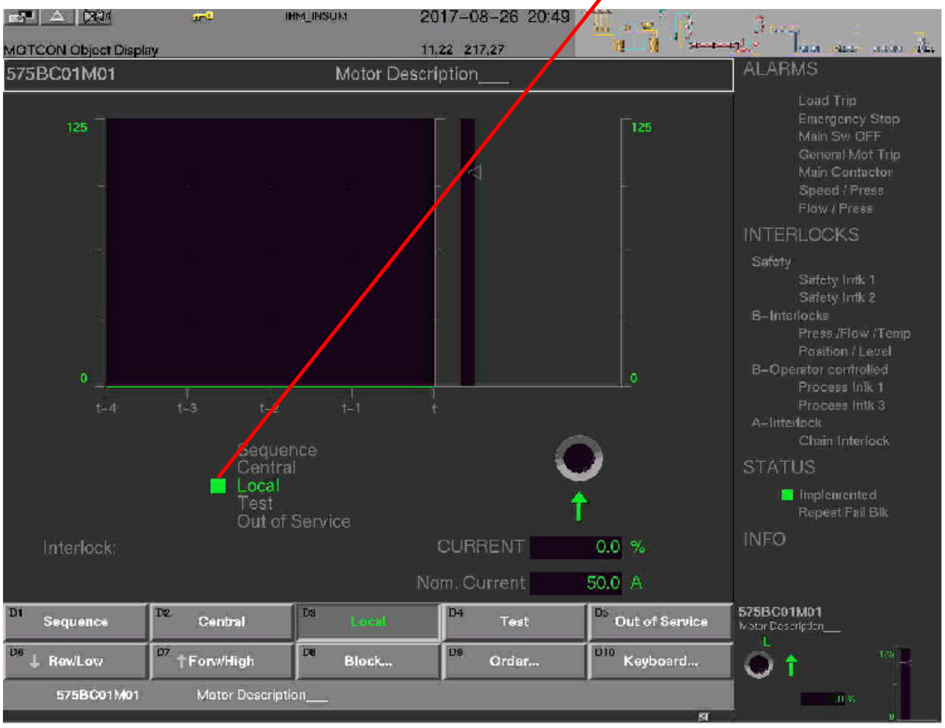
Following screen copies shall demonstrate HMI on AC450:

Notes:

CA owner of the MCU is DCS

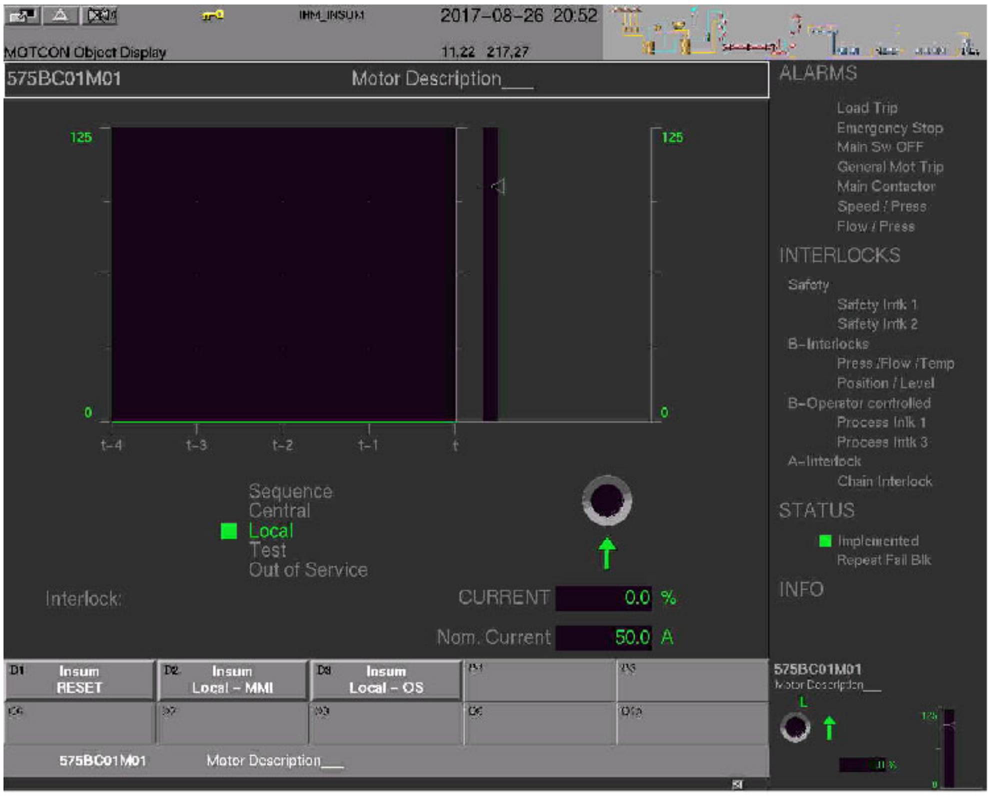


The CA owner is Local



Notes:

The MOTCON Object dialogue key ‘D6-Order’ can support the three output command order from Motcon. This can be effectively utilised to pass the CA to MMI and INSUM OS. one of the outputs will be utilised to reset the faults on the MCU. The order nomenclature can be changed as per the following hard copy outputs will be utilised to reset the faults on the MCU.



The MMI and INSUM OS will indicate the CA Owner in accordance with the parametrized names of the SU devices. Those CA owner names can be set via MMI:

1. Choose menu < SYSTEMCONFIGURATION >

2. Select Device

3. Choose submenu < SYSTEM>

4. Select CA owner name

5. Use function key EDIT to define or change this name (10 characters possible)

Notes:

4.2.2 Plant philosophy for operating the CA mechanism

In order to utilise the CA mechanism effectively the plant personnel must exercise the following guidelines.

The CA request for the desired MCU from the desired location of control must always be requested to the process operator in the control room. The mode of communication can be radio controlled or by the telephone.

The confirmation of the authorisation of CA must always be confirmed from the process operator before operating the MCU after the CA request.

It is recommended that the operation of MCU from any of the location must be co-ordinated among the process operator, MMI and INSUM OS user, and operational personnel operating via Local (soft) mode.

Scenario 1
Actual CA Owner of the MCU: DCS
Requested CA Owner of the MCU: Local (soft)

After the request for Local (soft) mode by the operational personnel, the operator at the control station evaluates the situation and can take the decision on granting the CA for the requested motor. When permitted operator selects the requested motor control objects and on the dialog menu orders the mode local. When the indication for the object changes to local (acknowledged based), confirms the operational personnel in the field about the permission to operate from the Local push button station.

Although it will be possible for the process operator to take back the control access at any time, it is advisable to inform the operating personnel in the field that the local permission is disabled.

Also the operating personnel must inform the process operator after the completion of process tuning to take back the control to DCS. It is also advisable that the operating personnel keep informed the MMI and INSUM OS operator regarding the local operation.

Scenario 2
Actual CA Owner of the MCU: DCS
Requested CA Owner of the MCU: MMI or INSUM OS

Same as described in Scenario 1.

Scenario 3
Actual CA Owner of the MCU: DCS
Invoked CA Owner of the MCU: Local (hardwired)

The maintenance personnel inform the process operator about the need to carry out the maintenance operation on the desired Motor. Although this mode does not require the passing of the CA from the process operator but informing the process operator about invoking the mode by key switch is advisable.

The maintenance personnel carry out the required maintenance during this time the motor object on the DCS will continue to show the local mode. After the completion, the maintenance personnel put back the mode to remote.

Notes:

Annex A: INSUM Terms and Abbreviations

Abbreviation	Term	Explanation / Comments
	Alarm	Alarm is defined as status transition from any state to abnormal state. Status transition to abnormal state can be data crossing over the predefined alarm limit.
	Backplane	INSUM backbone, holds following INSUM devices: Router, Gateways, Clock, Power Supply. Part of the INSUM Communication Unit, see ICU
CA	Control Access	A function of INSUM system that allows definition of operating privileges for each device level (e.g. PCS, Gateway, field device)
CAT	Control Access Table	Table containing control access privileges
CB	Circuit Breaker	Circuit breaker unit (here: ABB SACE Emax with electronic release PR112-PD/LON)
CT	Current Transformer	Current Transformer
DCS	Distributed Control System	see also PCS
Eth	Ethernet	Ethernet is a local area network (LAN) technology. The Ethernet standard specifies the physical medium, access control rules and the message frames.
	Event	An event is a status transition from one state to another. It can be defined as alarm, if the state is defined as abnormal or as warning as a pre-alarm state.
FD	Field Device	Term for devices connected to the LON fieldbus (e.g. motor control units or circuit breaker protection)
FU	Field Unit	see Field Device
GPI	General Purpose Input	Digital input on MCU for general use
GPO	General Purpose Output	Digital output on MCU for general use
GPS	Global Positioning System	System to detect local position, universal time and time zone, GPS technology provides accurate time to a system
GW	Gateway	A Gateway is used as an interface between LON protocol in INSUM and other communication protocols (e.g. TCP/IP, PROFIBUS, MODBUS)
HMI	Human Machine Interface	Generic expression for switchgear level communication interfaces to field devices, either switchboard mounted or hand held
ICU	INSUM Communications Unit	INSUM Communications Unit consists of devices such as backplane, Gateways, Routers, System Clock and Power Supply. It provides the communication interface within INSUM and between INSUM and control systems. Formerly used expressions: SGC, SU
INSUM	INSUM	Integrated System for User optimized Motor Management. The concept of INSUM is to provide a platform for integration of smart components, apparatus and software tools for engineering and operation of the motor control switchgea
INSUM OS	INSUM Operator Station	Tool to parameterise, monitor and control devices in the INSUM system
ITS	Integrated Tier Switch	The Intelligent Tier Switch is an ABB SlimLine switch fuse with integrated sensors and microprocessor based electronics for measurement and surveillance
LON	Local Operating Network	LON is used as an abbreviation for LonWorks network. A variation of LON is used as a switchgear bus in the INSUM system

Notes:

Abbreviation	Term	Explanation / Comments
LonTalk	LonTalk protocol	Fieldbus communication protocol used in LonWorks networks
LonWorks	LonWorks network	A communication network built using LonWorks network technology, including e.g. Neuron chip and LonTalk protocol
MCU	Motor Control Unit	Motor Control Unit is a common name for a product range of electronic motor controller devices (field device) in INSUM. A MCU is located in a MNS motor starter, where its main tasks are protection, control and monitoring of motor and the related motor starter equipment.
MMI	Man Machine Interface	The switchgear level INSUM HMI device to parameterize and control communication and field devices.
MNS	MNS	ABB Modular Low Voltage Switchgear
	MODBUS, MODBUS RTU	Fieldbus communication protocol
NV,nv	LON Network Variable	Network variable is a data item in LonTalk protocol application containing max. 31 bytes of data.
Nvi, nvi	LON Network Variable input	LON bus input variable
Nvo, nvo	LON Network Variable output	LON bus output variable
OS	Operator Station	see INSUM OS
PCS	Process Control System	High level process control system
PLC	Programmable Local Controller	Low level control unit
PR	Programmable Release	Circuit breaker protection/release unit (here: ABB SACE Emax PR112-PD/LON)
	PROFIBUS DP	Fieldbus communication protocol with cyclic data transfer
	PROFIBUS DP-V1	Fieldbus communication protocol, extension of PROFIBUS DP allowing acyclic data transfer and multi master.
PTB	Physikalisch-Technische Bundesanstalt	Authorized body in Germany to approve Ex-e applications.
PTC	Positive Temperature Coefficient	A temperature sensitive resistor used to detect high motor temperature and to trip the motor if an alarm level is reached.
RCU	Remote Control Unit	Locally installed control device for motor starter, interacting directly with starter passing MCU for local operations.
	Router	Connection device in the LON network to interconnect different LON subnets. Part of the INSUM Communications Unit.
RTC	Real Time Clock	Part of the INSUM System Clock and and optionally time master of the INSUM system
SCADA	Supervisory Control and Data Acquisition	
SGC	Switchgear Controller	Former term used for INSUM Communications Unit
SU	Switchgear Unit	Former term used for INSUM Communications Unit
	System Clock	INSUM device providing time synchronisation between a time master and all MCUs. Part of the INSUM Communication Unit, see ICU
TCP/IP	Transmission Control Protocol /Internet Protocol	TCP/IP is a high-level, connection oriented, reliable, full duplex communication protocol developed for integration of the heterogenous systems.
TFLC	Thermal Full Load Current	See MCU Parameter Description for explanation

Notes:

Abbreviation	Term	Explanation / Comments
TOL	Thermal Overload	See MCU Parameter Description for explanation
	Trip	A consequence of an alarm activated or an external trip command from another device to stop the motor or trip the circuit breaker.
UTC	Coordinated Universal Time	Coordinated Universal Time is the international time standard, formerly referred to as Greenwich Meridian Time (GMT). Zero (0) hours UTC is midnight in Greenwich England, which lies on the zero longitudinal meridian. Universal time is based on a 24 hours clock.
VU	Voltage Unit	Voltage measurement and power supply unit for MCU 2
	Wink	The Wink function enables identifcation of a device on the LON network. When a device receives a Wink-message via the fieldbus, it responds with a visual indication (flashing LED)

ABB Schaltanlagentechnik GmbH
Wallstadter Str. 59
D - 68526 Ladenburg / Germany

Related Products, News, Local Contacts:
www.abb.com/mns

Editor: DEAST/BT
Publication No: 1TGC901090M0201