bss

Security
through Innovation

Serial interface description

Art. No. 301000.0 / 301001.0

HISAC -
RFID reader

Table of contents

1. Disclaimer

Disclaimer

Passing on and copying of these documents, utilisation and communication of their contents is not permitted unless expressly approved. Any infringement shall result in liability for damages. All rights reserved.

We have checked the contents of this publication for conformity with the components described. Nevertheless, deviations cannot be ruled out, so that we cannot guarantee complete conformity. The information in this publication is checked regularly and any necessary corrections are included in subsequent editions.

We are grateful for any suggestions for improvement.

We reserve the right to make technical changes.

Serial Interface Description - HISAC

Item no.:	301000.0 / 301001.0
Version:	1.1
Date:	05.10.22
Document:	30100xSerB_ENG

2. Document history

Changes to version Fw105 (2021-03-16):
2022-08-04 Fw116 Formatting (no change in content)
2022-06-21 Fw116 setting rfid signal
2022-04-21 Fw113 cmdSetRfidTest, cmdSetVolume, cmdGetVolume
2021-10-20 Fw111 rearrange some CAN commands for compatibility with HD-Lock, cmdSetBrightness, cmdGetBrightness, cmdSetWiegandBits, cmdGetWiegandBits,cmdSetPhaseOffset
2021-05-04 Fw107 touch state is reported periodically
2021-03-16 cmdSetColor, cmdSetBeepFrequency, cmdSetSignalTime

3. Activation

The CAN bus is activated via a tristate input (optionally CAN, Wiegand, no RFID)
If the bus select (Wiegand/CAN_/Busoff-) input is at GND potential during the switch-on phase, the CAN bus is selected.

4. Baud rate

The default baud rate is 250 Kbit/s and is configurable (via assignment table). The expected data volumes are very low, so stability and maximization of line lengths should be the priority.

5. Protocol

Data is transmitted in LSB format. I.e. if a UINT16 is transmitted, the first CAN byte is the less significant byte, followed by the significant byte.

Example:

Type	Value	Can Data Length	0	1	2	3
uint8	0x01	1	0x01			
uint16	0x1234	2	0x34	0x12		
uint32	0x12345678	4	0x78	0x56	0x34	0x12

6. Message structure of RFID button (to central control) (ClientEvent ce)

Each unit uses exactly one CAN identifier (standard or extended) as sender. This CAN identifier corresponds to the serial number as Extended-Id when delivered. This ID can be reprogrammed with SetId.

With SetId=0, the CAN identifier corresponds to the serial number as Extended-Id. All information is transmitted via this CAN identifier.

The 0th byte is used to distinguish (**MessageType**) how the content of the message is coded. The 1st byte is used for some messages to further distinguish the message.

MessageType	Byte 0	Byte 1	Byte 2-7 (6 byte payload)	Length	Broadcasting time
Unused	0x00				only from the HD-Lock
Status	0x01	0	uint16 flashing 1/0, uint16 Application/BL 1/0, uint16 projectID	8	all 2000ms
FwUpdate CRC	0x01	1	uint16 RequestNumber, uint32 crc32	8	response to cmdVerfiyCrc
Version	0x02	Touchpress	uint16 blVers, uint16 fwVers, uint16 hwVers	8	all 2000ms
Touch Down	0x03	1		2	when button is pressed
Touch Release	0x03	0		2	when button is released
CommunicationLost	0x04			1	when communication is lost and the default pattern is applied
Options	0x05	0	uint16 Options	4	response to cmdGetOptions
Brightness	0x06	brightness		2	response to cmdGetBrightness (0..200%)
WiegandBits	0x07	No. of Bits		2	response to cmdGetWiegandBits
Volume	0x08	volume		2	response to cmdGetVolume (0..100%)
RFID_Mifare	0x20	UID.LSB	UID.MSB	8	when new RFID was found
RFID_AES	0x21	UID.LSB	UID.MSB	8	when new RFID was found
RFID_EM4100	0x30	UID.LSB	UID.MSB	8	when new RFID was found
RFID_Hitag2	0x31	UID.LSB	UID.MSB	8	when new RFID was found
StdOut	0x40	Text[0]	Text[1]..Text[6]	2-8	Output of text messages (UTF8 encoding)
AnalogLed	0x50	0	uint16 R, uint16 G, uint16 B	8	response to cmdRequestAnalogLed

AnalogAux	0x51	0	uint16 supply, uint16 mode	6	response to cmdRequestAnalogAux (Voltage and BusSelect)
Acknowledge	0xFF	Command		2	response to any cidCommand

7. Message structure from central control to # RFID button

Information from the controller to the RFID button is done via a block of CAN standard IDs. The block must be contiguous and the lower byte (0x70n) determines the message ID described below. If the destination address is an Extended-Id, the MSB must be set. Using the target address 0x000 (std. Id), all # RFID buttons are addressed simultaneously (broadcast).

Message	CAN ID	Byte 0-3	Parameter	Length
cidCommand	0x700	Destination address	Command and Parameter	8
cidSetId	0x701	Destination address	uint32 NewId	8
cidFwuEraseSector	0x706	Destination address	uint32 Address	8
cidFwuWriteFlash	0x707	Destination address	uint16 Offset, uint16 Data	8

7.1. SetId (Set CAN ID)

For identifiers (newId) greater than 0x7FF, an extended ID (29 bit) is automatically set, until then a standard ID (11 bit).
If an extended ID up to 0x000007ff is to be used, the MSB must be set (newId = newId | 0x80000000).
With newId == 0, the serial number (the lower 29 bits) of the RFID button is set as the extended ID. The new CANId is only taken over at restart, i.e. a call **cmdSaveSettingsToFlash** with 5th byte == 1 must be made (or **cmdSaveSettingsToFlash** with 5th byte == 0 and then **cmdReboot**).

8. Commands (Command cmd)

Command (byte 4)	Length	Description	Byte 5	Byte 6	Byte 7	Description	Settings
0		Unused				only from the HD-Lock	
1	8	cmdSetCANbaudrate	x	Baud rate		Set baud rate	[x]
2	5	cmdFirmwareUpdateEnter					
3	5	cmdBootloaderUpdateEnter					
4	5	cmdUpdateLeave					
5	8	cmdSaveSettingsToFlash	x	1=Rebbot 0=NoReboot	x	Settings are saved and then a reboot can take place.	
6	6	cmdSetDesfireApplication	Bit0-7	Bit8-15	Bit16-23		[x]
7	8	cmdSetOptions	Option 0-7	Option 8-15	Option 16-23		[x]
8	8	cmdVerifyCrc	0 (unused)	Request # LSB	Request # MSB		
9	5	cmdGetOptions					
10 (0x0A)	8	cmdSetDesfireAESKey	Offset 0-7	Bit0-7	Bit8-15		[x]
11 (0x0B)	5	cmdReboot					
12 (0x0C)	6	cmdSetLogLevel	newLogLevel				
13 (0x0D)	6	cmdSetPhaseOffset	value				
14 (0x0E)	5	cmdRequestAnalogLed					
15 (0x0F)	5	cmdRequestAnalogAux					
16 (0x10)	8	cmdSetDefaultPattern	Bit0-7	Bit8-15	Bit16-23	This default pattern is used when the CAN communication timeout is exceeded. Structure like cmdSendSignal	[x]
17 (0x11)	8	cmdSetCommunicationTimeout	0 (unused)	Bit0-7	Bit8-15	Set the communication timeout for CAN in seconds. If no communication takes place, the default pattern is executed. 0=No timeout	[x]
18 (0x12)	6	cmdSetBrightness	brightness			Brightness of the LEDs 0..200%	[x]
19 (0x13)	5	cmdGetBrightness					

20 (0x14)	6	cmdSetWiegandBits	No. of Bits			max. 64 - number of bits in Wiegand mode used	[x]
21 (0x15)	5	cmdGetWiegandBits					
22 (0x16)	8	cmdSendSignal	Bit0-7	Bit8-15	Bit16-23	Start signaling	
23 (0x17)	8	cmdSetRfidTest	1/0			at 1 continuous signal when RFID detected	
24 (0x18)	6	cmdSetVolume	volume			Volume of the signal generator 0..100%	[x]
25 (0x19)	5	cmdGetVolume					
26 (0x1A)	5	cmdSetRfidPattern	bit0-7	bit8-15	bit16-23	rfid signal pattern plays when a new RFID was found	[x]
33.. 47 (0x21..0x2F)	8	cmdSetColor1..15	R	G	B	Set colour value (RGB)	
49..63 (0x31..0x3F)	7	cmdSetBeepFrequency1..15	Bit0-7	Bit8-15		Set frequency (Hz)	
64..78 (0x40..0x4E)	7	mdSetSignalTime0..14	Bit0-7	Bit8-15		Set signal duration (ms)	

8.1. cmdSetCANbaudrate

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	1	0	Value	0

Value	baud rate	max. length
00	5 Kbit/s	10000m

01	10 Kbit/s	5000m
02	20 Kbit/s	2500m
03	33,333 Kbit/s	
04	50 Kbit/s	1000m
05	83.333 Kbit/s	
06	100 Kbit/s	
07	125 Kbit/s	500m
08	250 Kbit/s	250m default
09	500 Kbit/s	100m
10	800 Kbit/s	
11	1 Mbit/s	25m

The new CANbaudrate is only adopted at restart, i.e. a call **cmdSaveSettingsToFlash** with 5th byte == 1 must be made or **cmdSaveSettingsToFlash** with 5th byte == 0 and then **cmdReboot**.

8.2. cmdSetDesfireAESKey

- multiplexed by byte 5
- example Key 0x8180 7170 6160 5150 4140 3130 2120 1110

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	0x0A	0	0x10	0x11

0x700	Destination address	0x0A	1	0x20	0x21
0x700	Destination address	0x0A	2	0x30	0x31
0x700	Destination address	0x0A	3	0x40	0x41
0x700	Destination address	0x0A	4	0x50	0x51
0x700	Destination address	0x0A	5	0x60	0x61
0x700	Destination address	0x0A	6	0x70	0x71
0x700	Destination address	0x0A	7	0x80	0x81

The new AES key is immediately active but not yet permanently saved.
To ensure that the key is also active after a restart, a **cmdSaveSettingsToFlash** call must be made.

8.3. cmdSetDesfireApplication

- e.g. desfire evl application key 0x123456

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	6	0x12	0x34	0x56

The new application is immediately active but not yet permanently saved.
To ensure that the application is also active after a restart, a **cmdSaveSettingsToFlash** call must be made.

8.4. cmdSetOptions

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	7	Bit0-7	Bit8-15	Bit16-23

Byte	Bit	Name	Description
5	0	enable em4100	1=125kHz, EM4100 protocol active, 0=disable
5	1	enable hitag	1=125kHz, Hitag protocol active, 0=disable
5	2	enable mifare	1=13.56Mhz, Mifare protocol active, 0=disable
5	3	enable aes	1=13.56Mhz, Desfire protocol active, 0=disable
5	5	rfid signal	1=Acoustic signal when ID read, 0=disable
5	6	tbd.	
5	7	tbd.	
6	0..7	tbd.	

The changed options are immediately active but not yet permanently saved.
To ensure that the options are also active after a restart, a call **cmdSaveSettingsToFlash** must be made.

8.5. cmdSetWiegandBits

CAN ID	Byte 0-3	Byte 4	Byte 5
0x700	Destination address	0x14	Bit width 10 to 64 bits

Example bit width 37 (35 bit user data)
Rule 1 = XAAAAAAAAAAAAAAAAAAAAAAAAAX
Rule 2 = EDDDDDDDDDDDDDXXXXXXXXXXXXXXXXX
Rule 3 = XXXXXXXXXXXXXXXDDDDDDDDDDDDDDDDO
X - Do not observe
A - Transponder number
D - Used to check the parity check of the rule.
E - Equal parity of 'D' bits
O - Unequal parity of 'D' bits

For odd bit widths, the middle bit is included in both parities.

8.6. cmdSendSignal

The SendSignal function activates a combination of light and sound with the defined repetition and colour.

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	0x16	Bit0-7	Bit8-15	Bit16-24

Bit	Byte	Id	Meaning
0..3	5 low nibble	Repeat	1..14 repetitions, 0=abort/stop, 15=endless
4..7	5 high nibble	Ontime	0..14 Duration, 15=endless
8..11	6 low nibble	Offtime	0..14Duration, 15=endless

12.. 15	6 high nibble	OnColour	0=Black(=Off),...Colors
16..20	7 low nibble	Tone	0=Off, otherwise Tone1..Tone15
20..24	7 high nibble	OffColor	0=Black(=Off),...Colors

e.g. 0x03 16 65 -> Tone 3=C6, Colour 1=White,100ms off time, 100ms on time, 5 repetitions

CAN ID	**Byte 0-3**	**Byte 4**	**Byte 5**		**Byte 6**		**Byte 7**	
			Ontime Bit 4..7	Repeat Bit 0..3	Colour Bit 12..15	Offtime Bit 8..11	Unused Bit 20..24	Tone Bit 16..20
0x700	Destination address	0x16	6	5	1	6	0	3
			Ontime: 100ms	5 x	Colour: white	Oftime: 100ms		Tone=C6

e.g. 0x00 29 3F -> Continuous red flashing (50ms on /250ms off), no sound

CAN ID	**Byte 0-3**	**Byte 4**	**Byte 5**		**Byte 6**		**Byte 7**	
			Ontime Bit 4..7	Repeat Bit 0..3	Colour Bit 12..15	Offtime Bit 8..11	Unused Bit 20..24	Tone Bit 16..20
0x700	Destination address	0x16	3	0xF	2	9	0	0
			Ontime: 50ms	permanent	Colour: red	Oftime: 250ms		Tone=C6

e.g. 0x00 10 F1 -> White continuous light without sound

CAN ID	Byte 0-3	Byte 4	Byte 5		Byte 6		Byte 7	
			Ontime	Repeat	Colour	Offtime	Unused	Tone
			Bit 4..7	Bit 0..3	Bit 12..15	Bit 8..11	Bit 20..24	Bit 16..20
0x700	Destination address	0x16	0xF	01	1	0	0	0
			endless	1 x	white	off		off

Colors

#	Colour						
0	Black/Off	**4**	Blue	**8**	Chartreuse	**12(0xC)**	Rose
1	White	**5**	Yellow	**9**	Dark orange	**13(0xD)**	Violet
2	Red	**6**	Cyan	**10(0xA)**	Springgreen	**14(0xE)**	Turquoise
3	Green	**7**	Magenta	**11(0xB)**	Azure	**15(0xF)**	Orange

Tone

#	Tone						
0	Off	**4**	D6	**8**	A6	**12 (0xC)**	E7
1	A5	**5**	E6	**9**	B6	**13 (0xD)**	F7
2	B5	**6**	F6	**10 (0xA)**	C7	**14 (0xE)**	G7
3	C6	**7**	G6	**11 (0xB)**	D7	**15 (0xF)**	A7

Time

#	Time						
0	20ms	4	60ms	8	200ms	12 **(0xC)**	600ms
1	30ms	5	80ms	9	250ms	13 **(0xD)**	800ms
2	40ms	6	100ms	10 **(0xA)**	300ms	14 **(0xE)**	1000ms
3	50ms	7	150ms	11 **(0xB)**	400ms	15 **(0xF)**	permanent

8.7. cmd SetColor

The SendSignal function activates a combination of light and sound with the defined repetition and colour.

With this function, the colours Color1 to Color15 can be freely configured with **cmdSetColor1** to **cmdSetColor15. The** changed colours are only retained until a restart.

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6	Byte 7
0x700	Destination address	0x21..0x2F	R	G	B

Colors

Byte4 SetColor..	Colour	Default
	Black/off	Cannot be changed
0x21	Color1	White
0x22	Color2	Red
0x23	Color3	Green
0x24	Color4	Blue

0x25	Color5	Yellow
0x26	Color6	Cyan
0x27	Color7	Magenta
0x28	Color8	Chartreuse
0x29	Color9	Dark orange
0x2A	Color10	Springgreen
0x2B	Color11	Azure
0x2C	Color12	Rose
0x2D	Color13	Violet
0x2E	Color14	Turquoise
0x2F	Color15	Orange

8.8. cmdSetBeepFrequency

With the function cmdSetBeepFrequency, BeepFrequency1 to BeepFrequency15 can be freely configured. The changed frequencies are only retained until the restart. Sets the frequency in Hz.

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6
0x700	Destination address	0x31..0x3F	Frequency LB	Frequency LB

Colour

Byte4 SetBeepFrequency...	Colour	Default
	Off	Cannot be changed
0x31	BeepFrequency1	A5
0x32	BeepFrequency2	B5
0x33	BeepFrequency3	C6
0x34	BeepFrequency4	D6
0x35	BeepFrequency5	E6
0x36	BeepFrequency6	F6
0x37	BeepFrequency7	G6
0x38	BeepFrequency8	A6
0x39	BeepFrequency9	B6
0x3A	BeepFrequency10	C7
0x3B	BeepFrequency11	D7
0x3C	BeepFrequency12	E7
0x3D	BeepFrequency13	F7
0x3E	BeepFrequency14	G7
0x3F	BeepFrequency15	A7

8.9. cmdSetSignalTime

Duration0 to Duration14 can be freely configured with cmdSetSignalTime0 to cmdSetSignalTime14. The changed times are only retained until the restart. Sets the signal duration in ms.

CAN ID	Byte 0-3	Byte 4	Byte 5	Byte 6
0x700	Destination address	0x40..0x4E	Signal duration LB	Signal duration HB

Colour

Byte4 SetSignalTime	Colour	Default
0x40	Duration0	20ms
0x41	Duration1	30ms
0x42	Duration2	40ms
0x43	Duration3	50ms
0x44	Duration4	60ms
0x45	Duration5	80ms
0x46	Duration6	100ms
0x47	Duration7	150ms
0x48	Duration8	200ms
0x49	Duration9	250ms
0x4A	Duration10	300ms
0x4B	Duration11	400ms

0x4C	Duration12	600ms
0x4D	Duration13	800ms
0x4E	Duration14	1000ms
	Duration15	Permanent cannot be changed

Example

With the Linux canutils, the following message can be created to generate an acoustic signal for all.

```
cansend can0 700#000000.16.65.13.03
#            ^^^                       CAN id for commands
#                ^^^^^^^^              Target Device (0=Broadcast)
#                        ^^            Command (22=cmdSendSignal)
#                           ^          OnTime, 6=100ms
#                            ^         Repeat, 5x
#                              ^       Color, 1=red
#                               ^      Offtime, 3=50ms
#                                 ^    OffColor, 0=black
#                                  ^   Pitch (Tone 3=C)

# the same command only to the RFID button with the standard Id 1234 (0x04D2)
cansend can0 700#D204.0000.16.65.13.03
```

```
# some answers from the button of Id Id 1234
candump can0
0.0123 can0 04D2#03.01                  # button down event
1.0023 can0 04D2#03.00                  # button release event
2.0023 can0 04D2#21.0123456789ABCD      # mifare rfid detected (UID=0123456789ABCD)
```