iSYS-6030 API

Devices: iSYS-6030

Revision: 6

Date: 2021-11-11

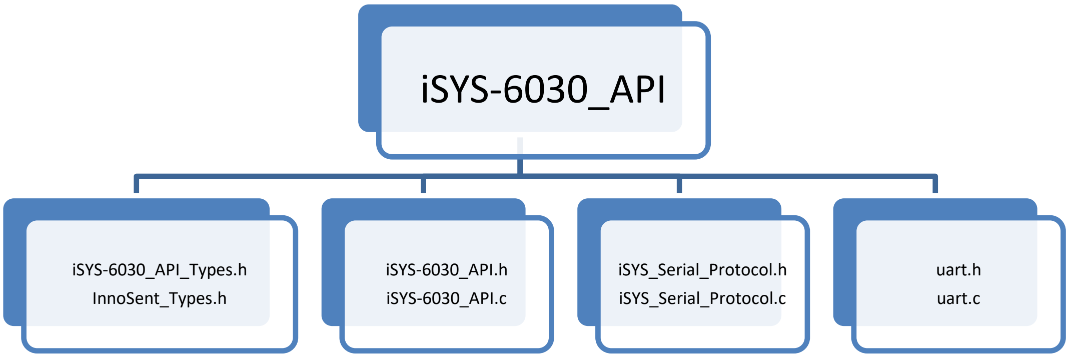
Table of content

History

Revision	Date	Change log	Author
1	2020-07-23	- Initial Release	CIB
2	2020-08-14	- Replaced example frame in protocol description with command which works with iSYS-6030	JW
3	2020-10-26	- Added application protocol with detailed description of each command	JW
4	2020-11-09	- Removed user transmit power back off	JW
5	2020-12-23	- Added new 25 Hz measurement mode (section 6.4.2) - Added protocol description for target list function and range list function with fixed 15 target length (section 6.8)	JW
6	2021-11-11	- Added description of reset sensor command - Added startup message from boot loader	JW

1. Get Radar API running in your own application

The iSYS-6030_API consists of the following components:



- **iSYS-6030_API_Types.h**
- **InnoSent_Types.h**

  Contains all basic type definitions used by the API.

  (Copy this into your project folder)
- **iSYS-6030_APi.(h/c)**

  Contains all commands for the iSYS-6030.

  (Copy this into your project folder)
- **iSYS_Serial_Protocol.(h/c)**

  Contains all necessary definitions, type definitions and structures featured by the API.

  For additional information on how the commands are constructed, please see the source code.

  (Copy this into your project folder)
- **uart.(c/h)**

  Contains routines for low level serial communication.

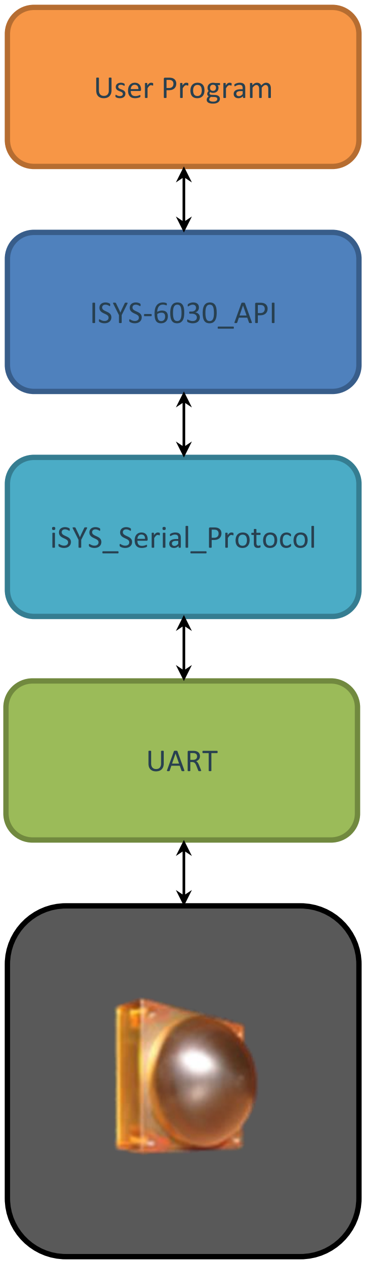
  (Copy this into your project folder or replace with serial communication for your hardware)

Also see delivered example projects on how to use iSYS-6030_API.

For a detailed description of the supported command refer to chapter 5.

2. API Overview

The iSYS-6030_API is designed to be included into a Qt program or for direct use with a microcontroller. Two example programs are provided.



3. Communication workflow

3.1. Transmission

3.1.1. Connection

The iSYS-6030 has a serial interface connection according to UART. Additional drivers for RS232 or RS485 2-wire and 4-wire standard can be connected using the 14 Pin header or Molex connector. For RS485 2-wire an additional signal for direction switching between Receive and Transmit mode is provide on the 14 Pin header. For the assignment of this pin refer to the datasheet of the iSYS-6030.

3.1.2. Communication

The communication uses the master/slave principle to support multiple slave devices (sensors) on one RS485 bus.

3.1.3. Addressing

The sensor device address is coded in one byte of each transmitted frame. This features 254 individual iSYS device addresses, the master device address and the broadcast address.

Each communication has to be initiated by the master device on address #1.

Frames send with the broadcast address (#0) as destination will be accepted from all connected devices on the bus.

Table 1: address range

Address	Description
0	Broadcast address
1	Master device address (bus master of RS485 connection)
2 to 255	Individual sensor slave addresses: default address: **iSYS-60XX = 100**;

3.1.4. Transmit format

8N1 transmission format: 1 start bit, 8 data bit, parity – none, 1 stop bit

3.1.5. Devices default baud rates

Table 2: default baudrates

Baud rate	devices
115.2 $\frac{kBit}{s}$	- iSYS-6030

3.1.6. Transmit frames

The communication works by transmitting two different frame types, differentiated through different start delimiters. Each frame contains one of three start delimiters, the destination and source address, function code, checksum and end delimiter. The whole frame must be send without breaks between the single bytes.

Table 3: frame with variable data length

SD2	LE	LEr	SD2	DA	SA	FC	PDU	FCS	ED
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte n+7	Byte n+8	Byte n+9

Table 4: frame with fixed data length

SD3	DA	SA	FC	PDU	FCS	ED
Byte 0	Byte 1	Byte 2	Byte 3	Byte n+4	Byte n+5	Byte n+6

SD: start delimiter to differ type of frame
1 = frame with variable data length (SD2 = 0x68), normally used for sensor communication
2 = frame with fixed data length (SD3 = 0xA2) only used for some answer from iSYS-6030
LE: length of the net data (data incl. DA, SA & FC appropriate byte 4 to byte n+7)
LEr: repetition of the net data length
DA: destination address
SA: source address
FC: function code
PDU: protocol data unit
FCS: frame checksum (addition of Bytes from Byte 4 to Byte n+7)
ED: end delimiter (ED = 0x16)

Incorrect or destroyed frames are cast away by the iSYS-6030 without replying.

All accepted frames are acknowledged with a frame including the function code and possible requested data by the iSYS-6030.

Received frames which cannot be execute are replied with the function code 0xFD (Failure).

3.1.7. Receive frames

The answer iSYS-6030 answer frames use the same protocol described in section 0.

3.1.8. Error Management

The transmission reliability is based on a frame checksum which is calculated from each transmitting device. After receiving a frame the device recalculates the checksum and compares it with the checksum transmitted inside the received frame. If the checksums don’t match, the frame is incorrect and must be cast away without replying.

Each frame has to be send as a continuous stream of data. When the time between two bytes exceeds the maximum delay time, the frame is discarded and the device waits for a new frame. The iSYS-6030 use a maximum delay time of about 10ms.

The master device (customer device) should use a pre-defined timeout to cancel transmissions.

3.1.9. Calculation of the frame checksum

The frame checksum is calculated by adding all bytes of source address, destination address, the function code and the protocol data unit. Overflows during the addition of the bytes are ignored.

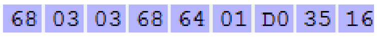
Table 5 show an example frame with the bytes used for checksum calculation highlighted green. Byte 1 to 3 (highlighted red) are SD2 exclusive and not used for checksum calculation.

$$Checksum = 0x64 + 0x01 + 0xD6 + 0x01 + 0x04 = 0x140 \rightarrow FCS = 0x40$$

Table 5: Example frame (data used for checksum calculation are highlighted in green)

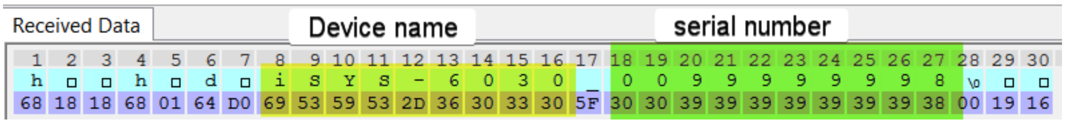
SD:	LE:	LEr:	SD:	DA:	SA:	FC:	PDU:	FCS:	ED:
0x68	0x05	0x05	0x68	0x64	0x01	0xD6	0x01 0x04	0x40	0x16

3.1.10. Example Frame (Read Device name)



```
68 03 03 68 64 01 D0 35 16
```

Figure 1: master requests device name from device address 100 – command string in HEX format



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
h	□	□	h	□	d	□	i	S	Y	S	-	6	0	3	0	_	0	0	9	9	9	9	9	9	9	8	\0	□	□
68	18	18	68	01	64	D0	69	53	59	53	2D	36	30	33	30	5F	30	30	39	39	39	39	39	39	39	38	00	19	16

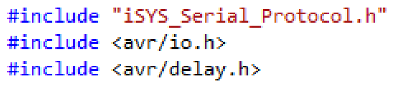
Figure 2: answer from device with device address 100

4. iSYS-6030_API on embedded systems

This example was tested on an Atmega328P microcontroller. If used on a different type, minor changes have to be made. Only the files *uart.h* and *uart.c* need to be modified.

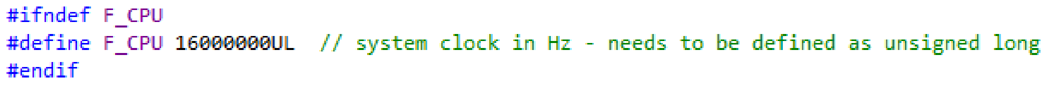
- *uart.h*

  Include the microcontroller's respective IO and delay functions to ensure correct data transmit/receive.



```
#include "iSYS_Serial_Protocol.h"
#include <avr/io.h>
#include <avr/delay.h>
```

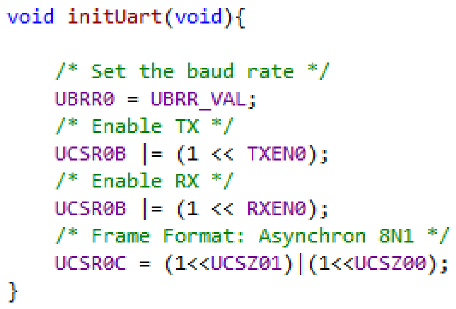
  The value for F_CPU must be adjusted to the microcontroller's clock frequency to make sure the baud rate and delays work correctly.



```
#ifndef F_CPU
#define F_CPU 16000000UL  // system clock in Hz - needs to be defined as unsigned long
#endif
```

- *uart.c*

  Change the UART registers in functions *initUart*, *uartTx* and *uartRx.*



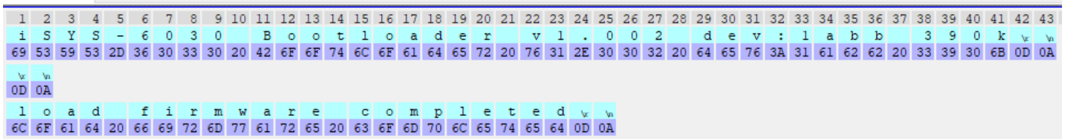
```
void initUart(void){

    /* Set the baud rate */
    UBRR0 = UBRR_VAL;
    /* Enable TX */
    UCSR0B |= (1 << TXEN0);
    /* Enable RX */
    UCSR0B |= (1 << RXEN0);
    /* Frame Format: Asynchron 8N1 */
    UCSR0C = (1<<UCSZ01)|(1<<UCSZ00);
}
```

5. Sensor start up message

When the iSYS-6030 powers up it will enter the secondary bootloader which then loads the application firmware. This bootloader allows an update of the sensor application firmware. This firmware update is normally done using the firmware update functionality provided by the iSYS-6030-GUI.

The purpose of this chapter is to show the messages the bootloader sends before the application firmware is ready to communicate.

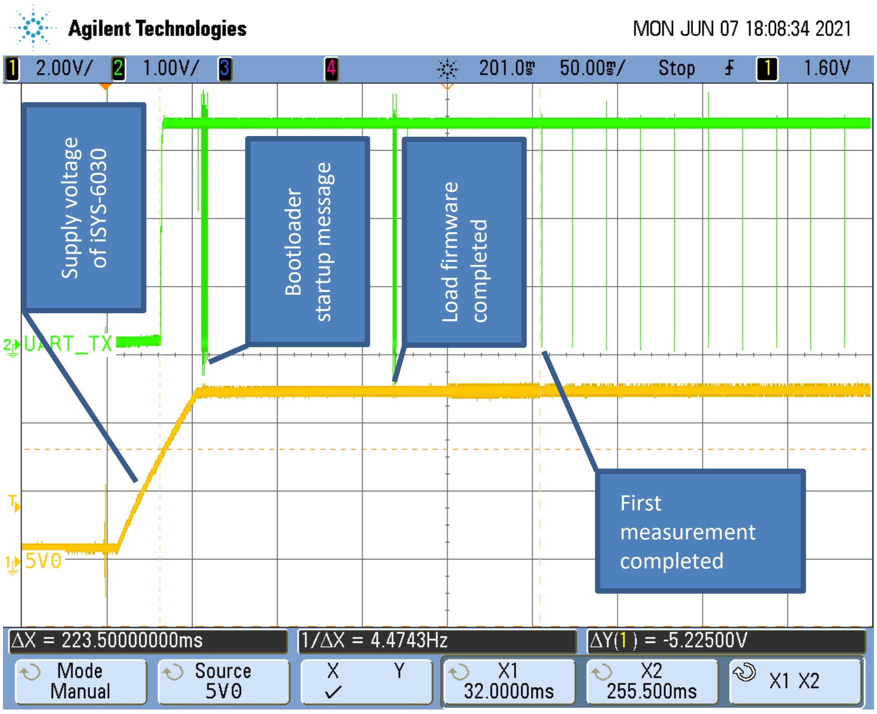
Figure 3 shows the messages send at startup. The first messages signals the startup of the bootloader, which then loads the firmware and sends **“load firmware completed\r\n”.**



```
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
  i  S  Y  S  -  6  0  3  0     B  o  o  t  l  o  a  d  e  r     v  1  .  0  0  2     d  e  v  :  1  a  b  b     3  9  0  k \r \n
 69 53 59 53 2D 36 30 33 30 20 42 6F 6F 74 6C 6F 61 64 65 72 20 76 31 2E 30 30 32 20 64 65 76 3A 31 61 62 62 20 33 39 30 6B 0D 0A
 \r \n
 0D 0A
  l  o  a  d     f  i  r  m  w  a  r  e     c  o  m  p  l  e  t  e  d \r \n
 6C 6F 61 64 20 66 69 72 6D 77 61 72 65 20 63 6F 6D 70 6C 65 74 65 64 0D 0A
```

Figure 3: Startup message send by the bootloader at sensor power up

This startup until the iSYS-6030 application firmware starts measuring takes about 225ms.



ΔX = 223.50000000ms

1/ΔX = 4.4743Hz

ΔY(1) = -5.22500V

Supply voltage of iSYS-6030

Bootloader startup message

Load firmware completed

First measurement completed

6. Application protocol

This chapter lists the different commands supported by the iSYS-6030. It is intended for users which want to implement the communication with iSYS-6030 on their own hardware without relying on the provided iSYS-6030 source code.

6.1. Function code overview

Table 6: available function codes

Function code	Function	Example
0xBC	Reset sensor	Triggers a software reset of the sensor
0xD0	Read Device name	Read device name as ASCII string
0xD1	Commands	Commands, e.g. start/stop measurement
0xD2	Read sensor settings	Read sensor parameters, e.g. sensor address
0xD3	Write sensor settings	Write sensor parameters, e.g. sensor address
0xD4	Read application settings	Read application parameters, e.g. min/max range
0xD5	Write application settings	Write application parameters, e.g. min/max range
0xD6	Read calibration settings	Read calibration settings, e.g. firmware version
0xD9	Read target list	Read target-list
0xDA	Read legacy target list	Read target-list (old function codes from iSYS-600x)
0xDF	NVM	NVM (none volatile memory) functions, e.g. set factory default settings or save settings
0xFD	Failure	Failure

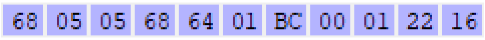
Some functions use additional sub-function codes. This splits the function code down to access single parameters. The sub-function codes are the first two data bytes and explained in the following sections if available.

6.1. Reset sensor (0xBC and sub function code 0x00 01)

This command request a software reset of the sensor. The iSYS-6030 will acknowledge this command and trigger a software reset. The iSYS-6030 then restarts from the bootloader which then loads the application firmware.

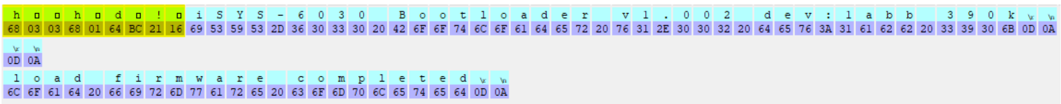
Figure 5 shows the answer from the sensor. The yellow marked frame is the command acknowledge from the application firmware. The rest of the communication is the bootloader start up message which the sensor transmits at power up.

Example:



```
68 05 05 68 64 01 BC 00 01 22 16
```

Figure 4: master device requests restart from the sensor



```
 h  □  □  h  □  d  □  !  □  i  S  Y  S  -  6  0  3  0     B  o  o  t  l  o  a  d  e  r     v  1  .  0  0  2     d  e  v  :  1  a  b  b     3  9  0  k \r \n
68 03 03 68 01 64 BC 21 16 69 53 59 53 2D 36 30 33 30 20 42 6F 6F 74 6C 6F 61 64 65 72 20 76 31 2E 30 30 32 20 64 65 76 3A 31 61 62 62 20 33 39 30 6B 0D 0A
\r \n
0D 0A
 l  o  a  d     f  i  r  m  w  a  r  e     c  o  m  p  l  e  t  e  d \r \n
6C 6F 61 64 20 66 69 72 6D 77 61 72 65 20 63 6F 6D 70 6C 65 74 65 64 0D 0A
```

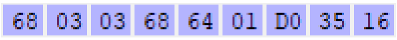
Figure 5: answer from sensor

API function:
packetStatus_t **iSYS6030_ResetSensor**(iSYS6030Handle_t sensor);

6.2. Read device name (0xD0)

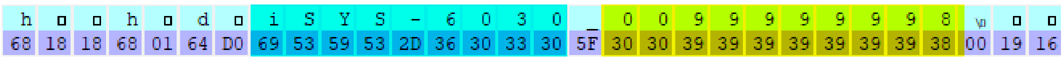
This command requests the device name and the serial number to identify the iSYS-6030 with an individual device number. The request is send by the master device within a frame of variable data length as shown in Figure 6. The slave device (sensor) sends its device name and serial number as a null-terminated ASCII string within a frame of variable data length. An example answer is shown in Figure 7.

Example:



```
68 03 03 68 64 01 D0 35 16
```

Figure 6: master device requests device name from sensor



```
 h  □  □  h  □  d  □  i  S  Y  S  -  6  0  3  0  _  0  0  9  9  9  9  9  9  9  8 \0  □  □
68 18 18 68 01 64 D0 69 53 59 53 2D 36 30 33 30 5F 30 30 39 39 39 39 39 39 39 38 00 19 16
```

Figure 7: answer from sensor

API function:
packetStatus_t **iSYS6030_getDeviceName**(iSYS6030Handle_t sensor, uint8_t* deviceNameString, uint8_t maxSize)

6.3. Commands (0xD1)

The master device sends the function code and one available sub-function code from Table 7 in a frame of variable data length. If the transmission is successful, the sensor acknowledges with the function code and possible data within a frame of variable data length.

Table 7: available sub function codes

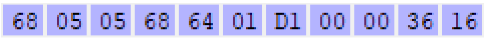
Sub-function codes	**Description**	**Example**
0x00 00	Start acquisition	Starts the sensor measurement
0x00 01	Stop acquisition	Stops the sensor measurement (This also stops modulation)
0x01 09	Read sensor temperature	Returns the temperature of the iSYS-6030

6.3.1.Start/Stop acquisition (0x00 00 / 0x00 01)

This sub-function codes are used to start or stop the measurement of the iSYS-6030. Some functions are only available when the acquisition is started or stopped.

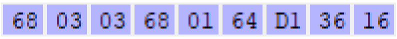
Examples:

- *Start acquisition*



68 05 05 68 64 01 D1 00 00 36 16

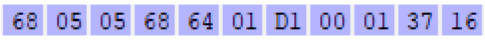
Figure 8: master device requests start of acquisition



68 03 03 68 01 64 D1 36 16

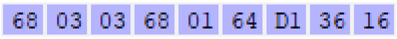
Figure 9: acknowledge from sensor

- *Stop acquisition*



68 05 05 68 64 01 D1 00 01 37 16

Figure 10: master device requests stop of acquisition



68 03 03 68 01 64 D1 36 16

Figure 11: acknowledge from sensor

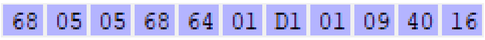
API function:

```
packetStatus_t iSYS6030_startAcquisition(iSYS6030Handle_t sensor);
packetStatus_t iSYS6030_stopAcquisition(iSYS6030Handle_t sensor);
```

6.3.2. Read sensor temperature (0x01 09)

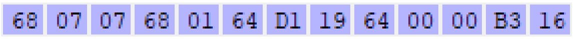
This sub-function code is used to read the temperature from the iSYS-6030. The sensor returns the current temperature in 0.01°C in the first two PDU bytes of a frame as sint16_t.

Examples:



```
68 05 05 68 64 01 D1 01 09 40 16
```

Figure 12: master device requests sensor temperature



```
68 07 07 68 01 64 D1 19 64 00 00 B3 16
```

Figure 13: sensor temperature (0x1964 = 6500 => 65.00°C)

API function:

```
packetStatus_t iSYS6030_ReadSensorTemperature(iSYS6030Handle_t sensor,
float *pSensorTemp);
```

6.4. Sensor settings (0xD2/0xD3)

The master device can read and write sensor settings from the connected iSYS-6030.

For reading the different settings, the master sends the function code 0xD2 and the sub-function code within a frame of variable data length. The iSYS-6030 answers with the function code and the requested data within a frame of variable data length.

For writing data to the iSYS-6030, the master sends a frame with function code 0xD3, the sub-function code and the data within a frame of variable data length.

***Note**: All writes are saved within volatile RAM on the sensor device. To save the sensor setting in non-volatile memory use the commands specified in section **6.8**.*

Table 8 shows a list of supported sub-function codes.

Table 8: available sub-function codes

Sub-function code	Sub-function	Description	Data-bytes
0x00 01	Address	Read/write the RS485 bus address (allowed values 2-255)	2 (uint16_t)
0x00 10	Measurement Mode	Read/write the measurement mode of the iSYS-6030	2 (uint16_t)
0x00 16	Threshold Sensitivity	Read/write the threshold sensitivity	2 (sint16_t)

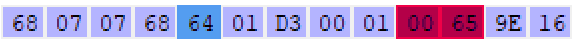
6.4.1.Read/Write sensor address (0x00 01)

This sub-function code is used to read the sensor address or change the default sensor address. The request is send within a frame of variable data length. The following example changes the sensor address from 0x64 (100) to 0x65 (101). The sensor acknowledges the request after changing the device address with the function code as shown in Figure 15.

The address can be read from the sensor using the function code 0xD2 and sub-function code 0x00 01 within a frame of variable data length. This is usually used to determine the address from the connected sensor, e.g. device address was changed previously and is no longer known. In the example all connected sensors will answer the request, because the broadcast address (0x00) is used. Therefore only one sensor should be connected when using the broadcast address.

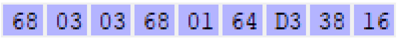
Examples:

- *Set new sensor address*



```
68 07 07 68 64 01 D3 00 01 00 65 9E 16
```

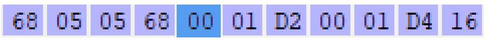
Figure 14: request from master device (only device with address 0x64) to change sensor address to 0x65



```
68 03 03 68 01 64 D3 38 16
```

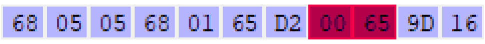
Figure 15: Acknowledge from sensor device

- *Read sensor address*



```
68 05 05 68 00 01 D2 00 01 D4 16
```

Figure 16: master requests sensor address using the broadcast address



```
68 05 05 68 01 65 D2 00 65 9D 16
```

Figure 17: acknowledge from sensor device with address (0x65) in protocol data unit

API functions:

packetStatus_t **iSYS6030_readRs485Address**(iSYS6030Handle_t sensor, uint8_t *address);

packetStatus_t **iSYS6030_writeRs485Address**(iSYS6030Handle_t sensor, uint8_t address);

6.4.2. Measurement mode (0x00 10)

The iSYS-6030 supports different measurement modes descripted in Table 9. This sub-function code is used to change the measurement mode or read the mode from the sensor.

Notes:

- Changing the measurement mode to one of the multi target modes does not disable the single target filter. This has to be separately done using the command from section 6.5.4

Table 9: overview of the measurement modes with code

PDU	Mode	Description	Update rate
0x00 00	Single target mode	Single target mode (fast measurement mode single target only)	50 Hz (20 ms)
0x00 01	Multi target mode (10 Hz)	Multi target mode (up to 10 targets)	10 Hz (100 ms)
0x00 02	Long integration mode	Long Integration mode (long time multi target mode with improved SNR)	4 Hz (250 ms)
0x00 03	Multi target mode (25 Hz)	Multi target mode (up to 15 targets)	25 Hz (40 ms)

Examples:

- *set measurement mode single target*

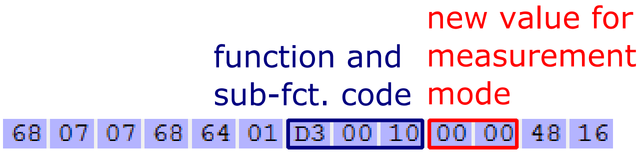

function and sub-fct. code
new value for measurement mode
68 07 07 68 64 01 D3 00 10 00 00 48 16

Figure 18: master requests fast measurement mode (0x00 00)

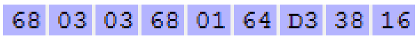


Figure 19: acknowledge from sensor

- *Read measurement mode*

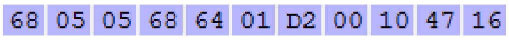


Figure 20: master requests measurement mode from sensor

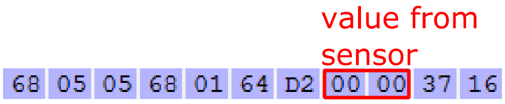


Figure 21: iSYS acknowledges with function code and measurement mode value (0x00 00: fast mode)

API functions:

packetStatus_t **iSYS6030_readMeasurementMode**(iSYS6030Handle_t sensor, iSYS6030_measurement_mode_t *pMode);
packetStatus_t **iSYS6030_writeMeasurementMode**(iSYS6030Handle_t sensor, iSYS6030_measurement_mode_t mode);

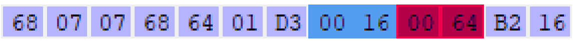
6.4.3. Threshold sensitivity (0x00 16)

This sub-function codes allow to change the offset of the threshold applied during the threshold calculation. These are the same values used in the iSYS-GUI-6030.

The values are transmitted as sint16_t data type in tenth of dB. Only writing of values between +10.0 dB and +30.0 dB is recommended. Values up to 100dB are possible. Requesting invalid values is replied with failure.

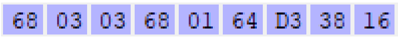
Examples:

- *Write threshold sensitivity*



```
68 07 07 68 64 01 D3 00 16 00 64 B2 16
```

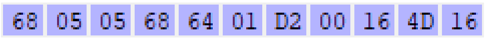
Figure 22: master requests new threshold sensitivity value (10dB)



```
68 03 03 68 01 64 D3 38 16
```

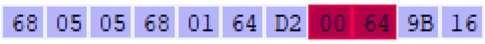
Figure 23: acknowledge from sensor

- *Read threshold sensitivity left*



```
68 05 05 68 64 01 D2 00 16 4D 16
```

Figure 24: master requests threshold sensitivity value from sensor



```
68 05 05 68 01 64 D2 00 64 9B 16
```

Figure 25: iSYS-6030 acknowledges with function code and threshold sensitivity value in PDU (0x00 64 = 100 => 10dB)

API functions:

```
packetStatus_t iSYS6030_readThreshold(iSYS6030Handle_t sensor, sint16_t* threshold);
packetStatus_t iSYS6030_writeThreshold(iSYS6030Handle_t sensor, sint16_t threshold);
```

6.5. Application settings (0xD4/0xD5)

The master device can read and write application settings to adapt the iSYS functionality to the desired application. A list of the available settings on the different iSYS is shown in Table 11.

For reading the different settings, the master sends a frame with function code (0xD4) and sub-function code. The iSYS-6030 answers with the function code 0xD4 and the requested data using as frame of variable data length.

For writing data to the iSYS-60, the master sends a frame with function code (0xD5), sub-function code and the data.

In both cases the requests are send within a frame of variable data length and sub-function code within the first two bytes of the PDU.

The following table lists the available sub-function codes used for the configuration of the three different digital outputs. The "X" in the sub-function codes refers to the number of the target list filter set (only filter set 1 is supported).

A more detailed description for each sub-function with working examples for output 1 follows in the next subsections.

***Note**: This settings are only stored in volatile RAM on the iSYS-6030. To save the setting in non-volatile memory use the commands specified in section* **6.8**.

Table 10: available sub-function codes

Sub-function code	Sub-function	Description	Data-bytes
0x0X 08	Range min	Min possible range (value in tenth of m)	2 (sint16_t)
0x0X 09	Range max	Max possible range (value in tenth of m)	2 (sint16_t)
0x0X 0A	Signal min	Min possible signal strength (value in tenth of dB)	2 (sint16_t)
0x0X 0B	Signal max	Max possible signal strength (value in tenth of dB)	2 (sint16_t)
0x0X 15	Single target filter type	Type of single target filter • 0=highest amplitude • 1=mean • 2=median • 3=min (closest) • 4=max (furthest)	2 (uint16_t)
0x0X 16	Single target filter signal	Signal for single target filter • 0=off • 2=range radial	2 (uint16_t)
0x07 0C	Digital Output Configuration	Configuration for the digital output functionality of the iSYS-6030	4 x uint8_t 1 x float32_t

Table 11: supported settings in the different iSYS-6030 modes

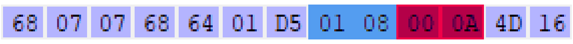
Device / mode	Range (min/max)	Signal (min/max)	Output signal target filter
iSYS-6030	Yes (0-40.0m)	Yes (0dB-255.0dB)	Yes
iSYS-6030 (long integration mode)	Yes (0-20.0m)	Yes (0dB-255.0dB)	Yes

6.5.1.Range min/max (0x0X 08 / 0x0X 09)

This sub-function codes are used to set the range boundaries of the detection area for the selected target filter set (currently only filter set 1 is supported). The target filter set is coded in the first byte of the sub-function code. The range value is transmitted as sint16_t in tenth of meter within the third and fourth PDU byte of a frame of variable data length. Write attempts of unsupported values are replied with failure by the iSYS-6030.

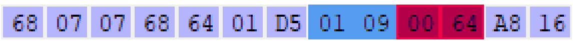
Examples:

- *To set the detection area for output 1 between 1m and 10m following frames are send from the master to the iSYS-6030. Both are replied with the function code as shown in Figure 28.*



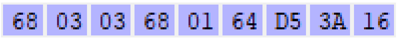
```
68 07 07 68 64 01 D5 01 08 00 0A 4D 16
```

Figure 26: master sends write request with new min range value (1m)



```
68 07 07 68 64 01 D5 01 09 00 64 A8 16
```

Figure 27: master sends write request with new max range value (10m)



```
68 03 03 68 01 64 D5 3A 16
```

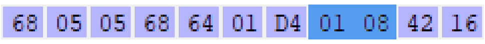
Figure 28: Acknowledge after sucessful write requests from iSYS-6030

API functions:

```
packetStatus_t iSYS6030_writeTargetListFilterMinRange(iSYS6030Handle_t sensor,
iSYS6030TargetListOutput_t targetListOutput, float minRange_m);
packetStatus_t iSYS6030_writeTargetListFilterMaxRange(iSYS6030Handle_t sensor,
iSYS6030TargetListOutput_t targetListOutput, float maxRange_m);
```

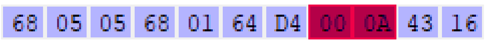
- *The frames send to read the range boundary values from the iSYS-6030 for target list filter set 1 are illustrated in the following figures.*

  - *Min range value*



```
68 05 05 68 64 01 D4 01 08 42 16
```

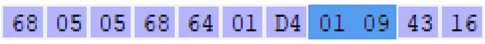
Figure 29: master sends read request for min range value to iSYS-6030



```
68 05 05 68 01 64 D4 00 0A 43 16
```

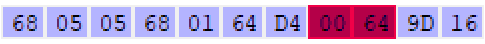
Figure 30: iSYS sends min range value (0x00 0A = 10 => 1m) in a SD2 frame with function code 0xD4

  - *Max range value*



```
68 05 05 68 64 01 D4 01 09 43 16
```

Figure 31: master sends read request for max range value to iSYS



```
68 05 05 68 01 64 D4 00 64 9D 16
```

Figure 32: iSYS sends max range value (0x00 64 = 100 => 10m) in a SD2 frame with function code 0xD4

API functions:

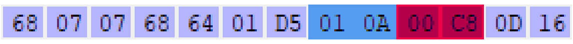
packetStatus_t **iSYS6030_readTargetListFilterMinRange**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float *pMinRange_m);
packetStatus_t **iSYS6030_readTargetListFilterMaxRange**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float *pMaxRange_m);

6.5.2. Signal min/max (0x0X 0A / 0x0X 0B)

This sub-function codes are used to set the signal strength boundaries of the detection area for the selected target filter set (currently only filter set 1 is supported). The target list filter set number is coded in the first byte of the sub-function code. The signal value is transmitted as sint16_t in tenth of dB within the third and fourth PDU byte of a frame of variable data length. Write attempts of unsupported values are replied with failure by the iSYS-6030.

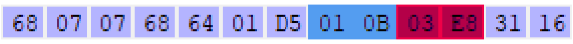
Examples:

- *To set the detection area for target list filter set 1 between 20dB and 100dB the following frames are send from the master to the iSYS-6030. Both are replied with the function code as shown in Figure 35.*



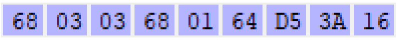
```
68 07 07 68 64 01 D5 01 0A 00 C8 0D 16
```

Figure 33: master sends write request with new min signal value (20dB)



```
68 07 07 68 64 01 D5 01 0B 03 E8 31 16
```

Figure 34: master sends write request with new max signal value (100dB)



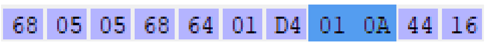
```
68 03 03 68 01 64 D5 3A 16
```

Figure 35: Acknowledge after sucessful write requests from iSYS-6030

API functions:

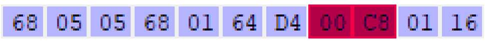
packetStatus_t **iSYS6030_writeTargetListFilterMinSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float minSignal_dB);
packetStatus_t **iSYS6030_writeTargetListFilterMaxSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float maxSignal_dB);

- *The signal boundary values for target filter set are read with following frames*
  - *Min signal value*



```
68 05 05 68 64 01 D4 01 0A 44 16
```

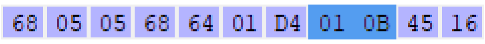
Figure 36: master sends read request for min signal value to iSYS-6030



```
68 05 05 68 01 64 D4 00 C8 01 16
```

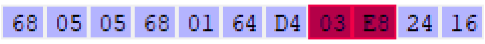
Figure 37: iSYS-6030 sends min signal value (0x00 C8 = 200 => 20dB)
in a SD2 frame with function code 0xD4

  - *Max signal value*



```
68 05 05 68 64 01 D4 01 0B 45 16
```

Figure 38: master sends read request for max signal value to iSYS-6030



```
68 05 05 68 01 64 D4 03 E8 24 16
```

Figure 39: iSYS sends max signal value (0x03 E8 = 1000 => 100dB)
in a SD2 frame with function code 0xD4

API functions:

packetStatus_t **iSYS6030_readTargetListFilterMinSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float *pMinSignal_dB);
packetStatus_t **iSYS6030_readTargetListFilterMaxSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, float *pMaxSignal_dB);

6.5.3.Single target filter type (0x0X 15)

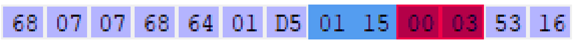
This sub-function code sets the single target filter type for the selected target filter set (currently only filter set 1 is supported). The target filter set is coded in the first byte of the sub-function code. The filter type is coded in the third and fourth PDU byte of a frame of variable data length. The supported values are shown in Table 12. Write attempts of unsupported values are replied with failure by the iSYS-6030.

Table 12: values for setting single target filter type (sub-function code 0x0X 15)

Value	Description
0x00 00	Highest amplitude
0x00 01	mean
0x00 02	median
0x00 03	Min (closest)
0x00 04	Max (furthest)

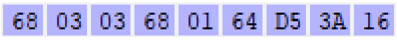
Examples:

- *The following example sets the single target filter type to minimum. The frame is replied on success with the function frame shown in Figure 41.*



```
68 07 07 68 64 01 D5 01 15 00 03 53 16
```

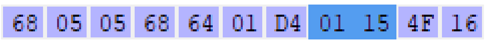
Figure 40: master sends write request with new target list filter (0x00 03 = min filter)



```
68 03 03 68 01 64 D5 3A 16
```

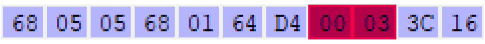
Figure 41: Acknowledge after sucessful write requests from iSYS-6030

- *The following frame is send to read the single target filter type from the iSYS-6030 for target list filter set 1.*



```
68 05 05 68 64 01 D4 01 15 4F 16
```

Figure 42: master sends read request for single filter type to iSYS-6030



```
68 05 05 68 01 64 D4 00 03 3C 16
```

Figure 43: iSYS sends output target filter setting for the filter (0x00 03 = min filter) in a SD2 frame with function code 0xD4

API functions:

packetStatus_t **iSYS6030_writeTargetListFilterSingleTargetType**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, iSYS6030SingleTargetFilterType_t type);
packetStatus_t **iSYS6030_readTargetListFilterSingleTargetType**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, iSYS6030SingleTargetFilterType_t *pType);

6.5.4.Single target filter signal (0x0X 16)

This sub-function code sets the single target filter signal for the selected target filter set (currently only filter set 1 is supported). The target filter set number is coded in the first byte of the sub-function code. The filter signal is coded in the third and fourth PDU byte of a frame of variable data length. This function code together with the output filter type from chapter 6.5.3 determines the used single target filter. For example setting the filter type to min and the filter signal to range radial configures the iSYS-6030 to use a min range single target filter which returns the closest detected target within the detection area.

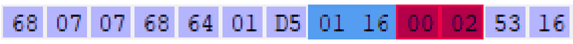
The supported values are shown in Table 13. Write attempts of unsupported values are replied with failure by the iSYS-6030. Setting the filter signal to 0x00 00 disables the single target filter for the selected output and all targets within the detection window are outputted. Use this to output an unfiltered target list.

Table 13: values for setting single target filter signal (sub-function code 0x0X 16)

Value	Description
0x00 00	Off (not supported in measurement mode single target)
0x00 02	Range radial

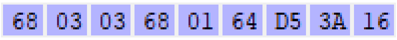
Examples:

- *The following example sets the single target filter signal to range radial. The frame is replied on success with the frame shown in Figure 45.*



```
68 07 07 68 64 01 D5 01 16 00 02 53 16
```

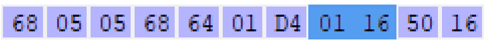
Figure 44: master sends write request with new signal for filter (0x00 02 = range radial)



```
68 03 03 68 01 64 D5 3A 16
```

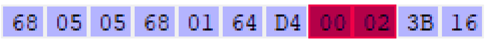
Figure 45: Acknowledge after sucessful write requests from iSYS-6030

The following frame is send to read the single target filter signal from the iSYS-6030 for target list filter set 1.



```
68 05 05 68 64 01 D4 01 16 50 16
```

Figure 46: master sends read request single target filter signal to iSYS-6030



```
68 05 05 68 01 64 D4 00 02 3B 16
```

Figure 47: iSYS sends output signal setting for the filter (0x00 02 = range radial) in a SD2 frame with function code 0xD4

API functions:

packetStatus_t **iSYS6030_writeTargetListFilterSingleTargetSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, iSYS6030SingleTargetFilterSignal_t signal);
packetStatus_t **iSYS6030_readTargetListFilterSingleTargetSignal**(iSYS6030Handle_t sensor, iSYS6030TargetListOutput_t targetListOutput, iSYS6030SingleTargetFilterSignal_t *pSignal);

6.5.5. Digital output Configuration (0x07 0C)

This sub-function code reads or writes the configuration of the digital outputs of the iSYS-6030. The iSYS-6030 provides up to four individually configurable digital outputs. Table 14 provides a list of the available output functionalities.

Write attempts of unsupported configurations are replied with failure by the iSYS-6030.

Table 14: available digital output functions

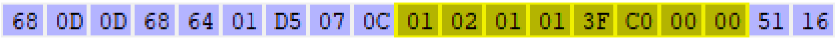
Value	Function	Description
0x00	None	Output is disabled and outputs inactive state
0x01	Status	outputs active state after iSYS-6030 finished start up and first measurement was started
0x02	Under Range	Outputs active state if detected target range is below the configured threshold value
0x03	Over Range	Outputs active state if detected target range is above the configured threshold value
0x04	Under temperature	Outputs active state if internally measured temperature of the iSYS-6030 is below the configured threshold value
0x05	Over temperature	Outputs active state if internally measured temperature of the iSYS-6030 above the configured threshold value
0x06	Detection	Outputs active state if at least one target was detected
0x07	UART_TX_ENABLE	Outputs provides a TX_ENABLE signal which can be used for direction switching when using a RS-485 Half-Duplex Transceiver (high: sensor is transmitting data, low: sensor can receive data)

Table 15: PDU of a configuration frame received from the iSYS-6030

PDU Byte [0]	PDU Byte [1]	PDU Byte [2]	PDU Byte [3]	PDU Byte [4…7]
Digital output number [0..3]	Digital output function (Table 14)	Active State: **0:** Low Active **1:** High Active	Target filter set number. Only target filter set 1 supported. If not used set to 0	Threshold value (m or °C) depending on selected function. Send as float32_t with Big Endian Byte order. If not used set 0

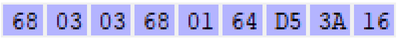
Examples:

- *The following example sets the digital output 1 for under range with threshold 1.500m and high active. The frame is replied on success with the frame shown in Figure 49.*



```
68 0D 0D 68 64 01 D5 07 0C 01 02 01 01 3F C0 00 00 51 16
```

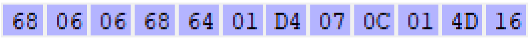
*Figure 48: master sends write request with new digital output configuration (**0x01**: output1, **0x02:** under range function, **0x01**: active high **0x01**: target list filter set 1, **0x3FC00000**: 1.500m)*



```
68 03 03 68 01 64 D5 3A 16
```

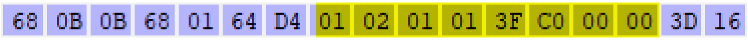
Figure 49: Acknowledge after sucessful write requests from iSYS-6030

- *The following frame is send to read the digital output configuration from the iSYS-6030 for digital output 1.*



```
68 06 06 68 64 01 D4 07 0C 01 4D 16
```

Figure 50: master requests digital output configuration for digital output 1 from iSYS-6030



```
68 0B 0B 68 01 64 D4 01 02 01 01 3F C0 00 00 3D 16
```

*Figure 51: iSYS sends digital output configuration for output 1 (**0x01**: output1, **0x02:** under range function, **0x01**: active high **0x01**: target list filter set 1, **0x3FC00000**: 1.500m)*

API functions:

packetStatus_t **iSYS6030_WriteDigitalOutputConfig**(iSYS6030Handle_t sensor, iSYS6030DigOuts_t digOutput, iSYS6030DigOutFunctions_t digOutFct, iSYS6030DigOutStates_t digOutActiveState, iSYS6030TargetListOutput_t tlFilterNr, float digThldValue);

packetStatus_t **iSYS6030_readDigitalOutputConfig**(iSYS6030Handle_t sensor, iSYS6030DigOuts_t digOutput, iSYS6030DigOutFunctions_t *pDigOutFct, iSYS6030DigOutStates_t *pDigOutActiveState, iSYS6030TargetListOutput_t *pTlFilterNr, float *pDigThldValue);

6.6. Read calibration settings (0xD6)

This function code is used to read the calibration setting from a connected iSYS-6030. A list of the available sub-function codes is shown in Table 16.

The master sends a frame of variable data length with function code (0xD6) and the sub-function code. The iSYS-6030 answers with the function code and the requested data or failure (see 6.10) within a frame of variable data length.

Table 16: available sub-function codes for function code 0xD6

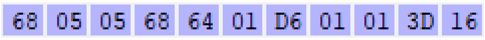
Sub-function code	Sub-function	Description	Data-bytes
0x01 01	Firmware version	Returns the firmware version running on the connected iSYS-6030	6 (3 x uint16_t)
0x01 02	DSP-Hardware version	Returns the hardware version of the connected iSYS-6030	6 (3 x uint16_t)
0x01 04	Product info	Returns the product info (product code) of the connected iSYS-6030	2 (1 x uint16_t)
0x02 20	Bootloader version	Returns the bootloader version of the connected iSYS-6030	6 (3 x uint16_t)

6.6.1. Firmware version (0x01 01)

This sub-function code is used to read the version of the firmware running on the iSYS-6030. The version is send as three uint16_t values within the data section of the variable data length frame. The first uint16_t determines the major version in front of the decimal separator. The second uint16_t value determines the decimal places behind the decimal separator. The third value contains the minor version of the firmware.

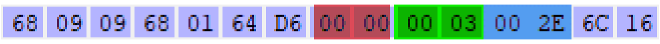
Example:

The following example show the frame send for requesting the firmware version and the decoding of the answer frame from the iSYS-6030 device.



```
68 05 05 68 64 01 D6 01 01 3D 16
```

Figure 52: master device requests firmware version from the iSYS with device address 128



```
68 09 09 68 01 64 D6 00 00 00 03 00 2E 6C 16
```

Figure 53: frame send with firmware version from the iSYS-6030 device

Decoding of the example answer frame:

Major version:	0x0000 = 0
Fixed point place:	0x0003 = 3
Minor version:	0x002E = 46
Firmware version:	**0.046**

API function:

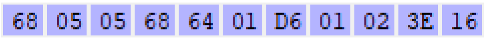
```
packetStatus_t iSYS6030_readFirmwareVersion(iSYS6030Handle_t sensor, uint16_t *pMajor,
uint16_t *pFix, uint16_t *pMin);
```

6.6.2. Get hardware version (0x01 02)

This sub-function code is used to read out the hardware version of the iSYS-6030. The return value is transmitted as three uint16 values inside the data section of a variable data length frame. Byte one and two are the major version. Byte three and four are the fixed point place (this is used to determinate the length of the version). Byte five and six are the minor version.

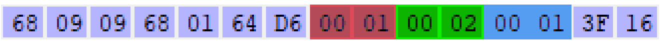
Examples:

- *The following example shows how the iSYS-6030 returns the requested hardware version*



```
68 05 05 68 64 01 D6 01 02 3E 16
```

Figure 54: requesting the hardware version from iSYS-6030



```
68 09 09 68 01 64 D6 00 01 00 02 00 01 3F 16
```

Figure 55: frame send with hardware version from the iSYS-6030

Decoding of the example answer frame:

Major version:	0x0001 = 1
Fixed point place:	0x0002 = 2
Minor version:	0x0001 = 1
Hardware version:	**1.01**

API function:

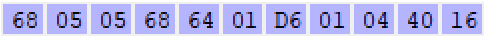
packetStatus_t **iSYS6030_readHardwareVersion**(iSYS6030Handle_t sensor, uint16_t *pMajor, uint16_t *pFix, uint16_t *pMin);

6.6.3. Get product information (0x01 04)

This function code is used to read the product information from the sensor. The return value is transmitted as uint16 value in the data section of the frame.

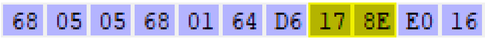
Examples:

- *The following example shows how the iSYS-6030 returns the requested product information (product code)*



```
68 05 05 68 64 01 D6 01 04 40 16
```

Figure 56: requesting the product information from the iSYS-6030



```
68 05 05 68 01 64 D6 17 8E E0 16
```

Figure 57: answer with the product information which was requested (0x17 0x8E = 6030)

API function:

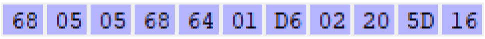
packetStatus_t **iSYS6030_readProductInfo**(iSYS6030Handle_t sensor,uint16_t *pProductInfo);

6.6.4. Get Bootloader version (0x02 20)

This sub-function code requests the version of the bootloader running on the iSYS-6030. The three uint16_t values send in the data section of a frame of variable data length represent the bootloader version. The first uint16_t determines the major version in front of the decimal separator. The second uint16_t value determines the number of decimal places behind the decimal separator. The third value contains the minor bootloader version. The iSYS-6030 answers with failure, if the command is not supported. Bootloader version 65535.65535 is send if the firmware supports the command, but no valid bootloader version could be read.

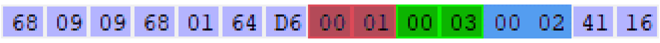
Example:

The following example shows the frame send for requesting the bootloader version and the decoding of the answer frame from the iSYS-6030.



68 05 05 68 64 01 D6 02 20 5D 16

Figure 58: master device requests bootloader version of the iSYS with device address 100



68 09 09 68 01 64 D6 00 01 00 03 00 02 41 16

Figure 59: frame send with bootloader version from the iSYS-6030

Decoding of the example answer frame:

Major version:	0x0001 = 1
Fixed point place:	0x0003 = 3
Minor version:	0x0002 = 2
Bootloader version:	**1.002**

API function:

packetStatus_t **iSYS6030_readBootloaderVersion**(iSYS6030Handle_t sensor, uint16_t *pMajor, uint16_t *pFix, uint16_t *pMin);

6.7. Read Target list (0xD9)

With this command the master device requests the target list of one measurement cycle. This is the target list processed from one measurement cycle filtered with the selected target list filter. This command only works if the iSYS-6030 is in active measurement mode. If the sensor is not in measurement mode, all target list request will be answered with failure. If the request is send after stopping a previous active measurement the target list from the last cycle can still be requested once. For changing the sensor mode refer to section 6.3.1.

The targets are sent one after the other in the data section of a frame of variable data length (SD2 frame).

After complete transmission of the target list a new target list can be requested. The iSYS-6030 sends the next target list as soon as a new one is available. You can request the target lists with the iSYS-6030 cycle time given in Table 9 using this function code.

Different target lists can be requested. The type of the target list is specified in the first two bytes of the PDU. The first byte in the PDU determines the parameters used for filtering the target list. The iSYS-6030 only supports target list filter parameter sets, which can be set as described in section 6.5. For an overview of the supported filter settings refer to Table 11. The second byte determines the requested target list type. Table 17 shows the available target list types.

Table 17: available target list types

Value	Description
0x00	Target list type single
0x01	Target list type fix 10 targets
0x20	Target list with variable length (Target list only contains the number of detected targets. Frame length depended on number of detected targets)

All target list types are supported in all measurement modes. Following examples show some valid requests and the output from the iSYS-6030:

iSYS-6030 measurement mode	Target list request	Target list output from iSYS-6030
Single target mode	Target list type single	Target list with one target filtered according to configured target list filter
	Target list type fix 10 targets	Target list with one target filtered according to configured target list filter and 9 dummy targets filled with zeros
multi target mode / long integration mode	Target list type single	Target list with one target filtered according to configured target list filter (even if disabled)
	Target list type fix 10 targets	Target list with up to 10 targets (sorted ascending by range). - Target list is filled with dummy targets if fewer than 10 targets are detected. - If more than 10 targets could be detected only the 10 closest targets in the configured min/max range are outputted.

Table 18: PDU for target list

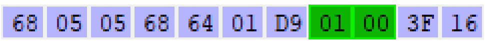
PDU[0]	PDU[1]	PDU[2...3]	PDU[4...7]		PDU[56...57]	PDU[58...61]
List number (number of target list filter set)	Number of targets	Signal of target 1	Range of target 1		Signal of target 10	Range of target 10

Table 19: resolution range of target list data

Parameter	Type	Range of values
Signal	signed integer 16 bit	-327.68 … 327.67 [dB]
Range	unsigned integer 32 bit	0 … 4,294,967,295 [µm]

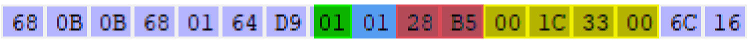
The following sections show example frames for requesting the different target list types filtered with target list filter set 1.

6.7.1. Request target list type single target



```
68 05 05 68 64 01 D9 01 00 3F 16
```

Figure 60: master requests target list type single target



```
68 0B 0B 68 01 64 D9 01 01 28 B5 00 1C 33 00 6C 16
```

Figure 61: iSYS-6030 send target list with one target

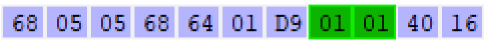
Table 20: decoding of target list output from Figure 61.

List / Output number	**Number of targets**	**Signal target 1 (104.21 dB)**	**Range target 1 (1.848064 m)**
0x01	0x01	0x28 B5 = 10,421	0x00 1C 33 00 = 1,848,064 µm

API function:

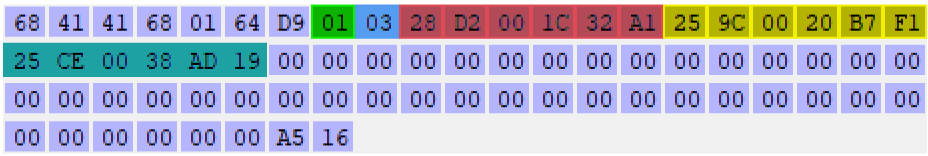
packetStatus_t **iSYS6030_readTargetListSingleTarget**(iSYS6030Handle_t sensor, iSYS6030TargetList_t *pTargetList, iSYS6030TargetListOutput_t targetListOutput);

6.7.2.Request target list type with fix 10 targets



```
68 05 05 68 64 01 D9 01 01 40 16
```

Figure 62: master requests target list type fix 10 targets



```
68 41 41 68 01 64 D9 01 03 28 D2 00 1C 32 A1 25 9C 00 20 B7 F1
25 CE 00 38 AD 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 A5 16
```

Figure 63: iSYS-6030 send target list with up to 10 targets

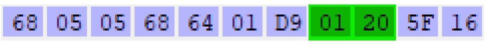
Table 21: decoding of target list output from Figure 63

Parameter		Hex value	Decimal value
List / Output number		0x01	1
Number of targets		0x03	3
Target 1	Signal	0x28 D2	104.5 dB
	Range	0x00 1C 32 A1	1.847969 m
Target 2	Signal	0x25 9C	96.28 dB
	Range	0x00 20 B7 F1	2.144241 m
Target 3	Signal	0x25 CE	96.78 dB
	Range	0x00 38 AD 19	3.714329 m
Target 4 … 10 (dummy targets)	Signal	0x00 00	0
	Range	0x00 00 00 00	0

API function:

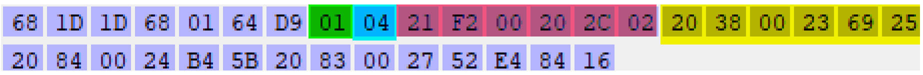
packetStatus_t **iSYS6030_readTargetListMultiTarget10**(iSYS6030Handle_t sensor, iSYS6030TargetList_t *pTargetList, iSYS6030TargetListOutput_t targetListOutput);

6.7.3.Request target list type variable length



```
68 05 05 68 64 01 D9 01 20 5F 16
```

Figure 64: master requests target list type variable length



```
68 1D 1D 68 01 64 D9 01 04 21 F2 00 20 2C 02 20 38 00 23 69 25
20 84 00 24 B4 5B 20 83 00 27 52 E4 84 16
```

Figure 65: iSYS-6030 send target list with variable length

Table 22: decoding of target list output from Figure 63

Parameter		Hex value	Decimal value
List / Output number		0x01	1
Number of targets		0x04	4
Target 1	Signal	0x21 F2	86.90 dB
	Range	0x00 20 2C 02	2.108418 m
Target 2	Signal	0x20 38	82.48 dB
	Range	0x00 23 69 25	2.320677 m
Target 3	Signal	0x20 84	83.24 dB
	Range	0x00 24 B4 5B	2.405467 m
Target 4	Signal	0x20 83	83.23 dB
	Range	0x00 27 52 E4	2.577124 m

API function:

packetStatus_t **iSYS6030_readTargetListVariableLength**(iSYS6030Handle_t sensor, iSYS6030TargetList_t *pTargetList, iSYS6030TargetListOutput_t targetListOutput);

6.8. Read legacy Target list (0xDA)

With this legacy command the master device requests the target list of one measurement cycle with the old format inherited from the iSYS-6003, iYS-6004 and iSYS-6005. This is the target list processed from one measurement cycle filtered with the selected target list filter. This command only works if the iSYS-6030 is in active measurement mode. If the sensor is not in measurement mode, all target list request will be answered with failure. If the request is send after stopping a previous active measurement the target list from the last cycle can still be requested once. For changing the sensor mode refer to section 6.3.1.

The targets are sent one after another in the data section of a SD3 frame, the maximum number of targets is the number of targets from the measurement mode selected with the command from section 6.3.1.

After complete transmission of the target list a new target list can be requested. The iSYS-6030 sends the next target list as soon as a new one is available. You can request the target lists with the iSYS-6030 times given in Table 9 using this function code.

Different target lists can be requested. The type of the target list is specified in the first two bytes of the PDU. The first byte in the PDU determines the parameters used for filtering the target list. The iSYS-6030 only supports target list filter parameter sets, which can be set as described in section 6.5. For an overview of the supported filter settings refer to Table 11. The second byte determines the requested target list type. Table 17 shows the available target list types.

Table 23: available target list types

Value	**Description**
0x20	Output of sint32_t variable types (32-Bit output)
0xA0	Requesting sint32_t variable types target list with fixed length for 15 targets

All target list types are supported in all measurement modes. Following examples show some valid requests and the output from the iSYS-6030:

iSYS-6030 measurement mode	Target list request	Target list output from iSYS-6030
Single target mode	32-Bit output	Target list with one target filtered according to configured target list filter
	32-Bit output with fix 15 targets	Target list with one target filtered according to configured target list filter and 14 dummy targets filled with zeros
multi target mode / long integration mode	32-Bit output	Target up to maximum number of targets filtered according to the selected target list filter
	32-Bit output with fix 15 targets	Target list with up to 15 targets (sorted ascending by range). - Target list is filled with dummy targets if fewer than 15 targets are detected. - If more than 15 targets could be detected only the 15 closest targets in the configured min/max range are outputted.

6.8.1.Format 32-Bit target list

In the first byte of the answer frame data section (PDU) the requested target list is coded. The second byte specifies the number of targets.

Table 24: PDU for 32 Bit target list

PDU[0]	PDU[1]	PDU[2…3]	PDU[4…7]	PDU[8…11]	PDU[12…15]
List number (number of target list filter set)	Number of targets	Signal of target 1	velocity of target 1	Range of target 1	Angle of target 1

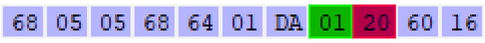
Table 25: resolution range of target list data

Parameter	Type	Range of values
Signal	unsigned integer 16 bit	0 … 655.35 [dB]
Velocity (v)	signed integer 32 bit	-2,147,483,648 … 2,147,483,647 [mm/s]
Range	signed integer 32 bit	-2,147,483,648 … 2,147,483,647 [µm]
Angle (a)	signed integer 32 bit	-2,147,483,648 … 2,147,483,647 [mill°]

Example:

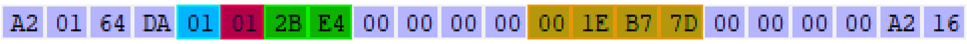
The following example frames show the requesting of the 30-Bit resolution target list filtered with target list filter set 1.

- *With 32-Bit resolution (0x20 in PDU Byte[1]):*



```
68 05 05 68 64 01 DA 01 20 60 16
```

Figure 66: master requests target list type single target



```
A2 01 64 DA 01 01 2B E4 00 00 00 00 00 1E B7 7D 00 00 00 00 A2 16
```

Figure 67: iSYS-6030 send target list with one target

Table 26: decoding of target list output from Figure 67.

List / Output number	Number of targets	Signal target 1 (112.36 dB)	Velocity target 1 (0m/s)	Range target 1 (2.013053 m)	Angle target 1 (0°)
0x01	0x01	0x2B E4 = 11,236	0x00 00 00 00 = mm/s	0x00 1E B7 7D = 2,013,053 µm	0x00 00 00 00 = mill°

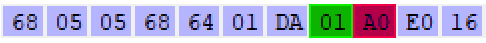
6.8.2. Fixed length target list

The answer frame for the fixed length target list uses the same structure as the 32-Bit target list described in section 6.8.1. Different to the normal 32-Bit target list the fixed length target list always transmits a fixed number of 15 targets. If fewer target as the number transmitted in the list the remaining targets are filled with zeros.

Example:

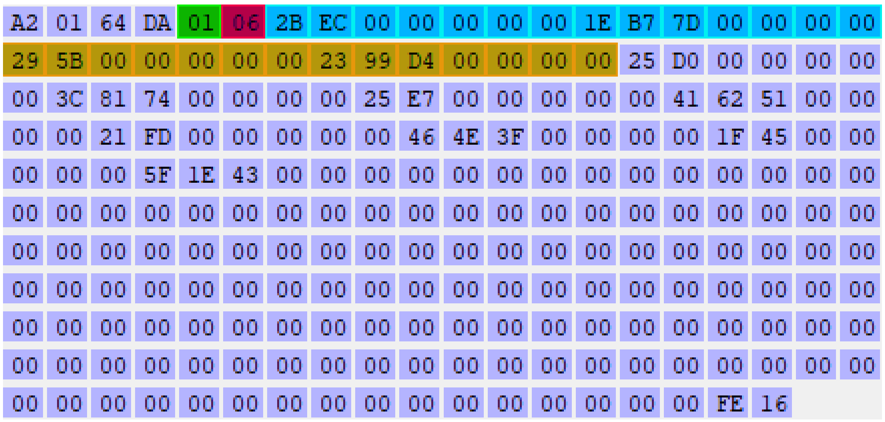
The following example frames show the requesting of the 32-Bit resolution target list filtered with target list filter set 1.

- *32-Bit resolution and 15 target fixed length (0xA0 in PDU Byte[1]):*



```
68 05 05 68 64 01 DA 01 A0 E0 16
```

Figure 68: master requests target list type fix 15 targets



```
A2 01 64 DA 01 06 2B EC 00 00 00 00 00 1E B7 7D 00 00 00 00
29 5B 00 00 00 00 00 23 99 D4 00 00 00 00 25 D0 00 00 00 00
00 3C 81 74 00 00 00 00 25 E7 00 00 00 00 00 41 62 51 00 00
00 00 21 FD 00 00 00 00 00 46 4E 3F 00 00 00 00 1F 45 00 00
00 00 00 5F 1E 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FE 16
```

Figure 69: iSYS-6030 send target list with up to 15 targets

Table 27: decoding of target list output from Figure 69

Parameter		Hex value	Decimal value
List / Output number		0x01	1
Number of targets		0x06	6
Target 1	Signal	0x2B EC	112.44 dB
	Velocity	0x00 00 00 00	0 m/s
	Range	0x00 1E B7 7D	2.013053 m
	angle	0x00 00 00 00	0°
Target 2	Signal	0x29 5B	105.87 dB
	Velocity	0x00 00 00 00	0 m/s
	Range	0x00 23 99 D4	2.333140 m
	angle	0x00 00 00 00	0°
Target 3 … 6	…	…	…
Target 7 … 15 (dummy targets)	…	…	…

API function:

```
packetStatus_t iSYS6030_readLegacyTargetListMultiTarget15(iSYS6030Handle_t sensor,
iSYS6030LegacyTargetList_t *pTargetList, iSYS6030TargetListOutput_t targetListOutput);
```

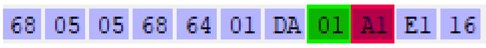
6.8.3. Fixed length range list

The answer frame for the range list length target list uses a structure similar to the 32-Bit fixed length target list described in section 6.8.2 without the unused velocity and angle information. It transmits a fixed length frame which can contain up to 15 ranges. If fewer ranges were detected the remaining frame is filled with zeros.

Example:

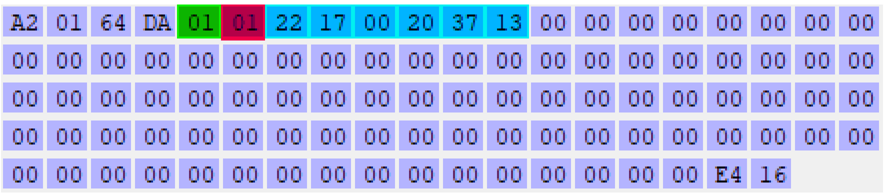
The following example frames show the requesting of the 32-Bit resolution range list filtered with target list filter set 1.

- *32-Bit resolution and 15 ranges fixed length (0xA1 in PDU Byte[1]):*



```
68 05 05 68 64 01 DA 01 A1 E1 16
```

Figure 70: master requests range list type fix 15 targets



```
A2 01 64 DA 01 01 22 17 00 20 37 13 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E4 16
```

Figure 71: iSYS-6030 send range list with up to 15 targets

Table 28: decoding of range list output from Figure 69

Parameter		Hex value	Decimal value
List / Output number		0x01	1
Number of targets		0x01	1
Target 1	Signal	0x22 17	87.27 dB
	Range	0x00 20 37 13	2.111251 m
Target 2 … 15 (dummy targets)	…	…	…

API function:

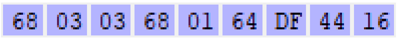
packetStatus_t **iSYS6030_readLegacyRangeList15RangesFix**(iSYS6030Handle_t sensor, iSYS6030TargetList_t *pTargetList, iSYS6030TargetListOutput_t targetListOutput)

6.9. NVM (0xDF)

These function code together with the sub-function codes in Table 29 are used to set the factory default settings of the sensor device or save changes to the sensor and application settings in non-volatile EEPROM on the Sensor device. All valid requests are answered by the iSYS device on success with the frame from Figure 72. Invalid request frames are answered with failure.

Table 29: none volatile memory (NVM) sub-function codes

Sub-function code	Sub-function	Description
0x01	Set factory settings	Restores the factory default settings of the iSYS device. This includes the complete sensor and application settings.
0x04	Save sensor and application settings	Saves settings from volatile RAM to none volatile memory.

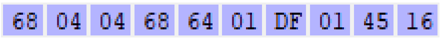


```
68 03 03 68 01 64 DF 44 16
```

Figure 72: iSYS-6030 acknowledge send after successful NVM command execution

6.9.1.Set factory settings

The **sub-function code 0x01** restores the factory default settings in NVM of the connected iSYS-6030. All setting changes made by the user are overwritten. The request frame is shown in Figure 73.



```
68 04 04 68 64 01 DF 01 45 16
```

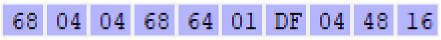
Figure 73: request frame for restoring the factory default settings of the iSYS device

API function:

packetStatus_t **iSYS6030_setFactorySettings**(iSYS6030Handle_t sensor);

6.9.2.Save sensor and application settings

This sub-function code is used to save the setting changed with the commands described in section 6.4 (sensor settings) and section 6.5 (application) in nonvolatile memory on the sensor.



```
68 04 04 68 64 01 DF 04 48 16
```

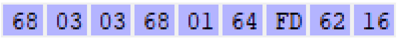
Figure 74: master devices requests for the saving of the sensor and application settings to NVM from the iSYS-6030

API function:

packetStatus_t **iSYS6030_saveSensorAndApplSetting**(iSYS6030Handle_t sensor);

6.10. Failure (0xFD)

This function code is send by the iSYS-6030, if a correct transmitted frame cannot execute. This can happen for example, if the values of the transmitted data were out of range, the requested function code is not supported or the iSYS-6030 is not in the right measurement mode.



```
68 03 03 68 01 64 FD 62 16
```

Figure 75: Frame send by iSYS-6030 if correct transmitted frame cannot execute

6.11. *Configuration example for iSYS-6030*

This chapter shows an example configuration for the iSYS-6030

Following setting are configured

- Multi target Mode
- Single target Filter off
- Range filter 1m … 10m

Table 30: configuration example for iSYS-6030

Step	Description	Request frame from master	Answer frame from iSYS-6030
1	Stop Acquisition	68 05 05 68 64 01 D1 00 01 37 16	68 03 03 68 01 64 D1 36 16
2	Set multi target mode	68 07 07 68 64 01 D3 00 10 00 01 49 16	68 03 03 68 01 64 D3 38 16
3	Disable single target filter	68 07 07 68 64 01 D5 01 16 00 00 51 16	68 03 03 68 01 64 D5 3A 16
4	Set min range filter	68 07 07 68 64 01 D5 01 08 00 0A 4D 16	68 03 03 68 01 64 D5 3A 16
5	Set max range filter	68 07 07 68 64 01 D5 01 09 00 64 A8 16	68 03 03 68 01 64 D5 3A 16
6	Start Acquisition	68 05 05 68 64 01 D1 00 00 36 16	68 03 03 68 01 64 D1 36 16
7	Request target list	68 05 05 68 64 01 D9 01 01 40 16	68 41 41 68 01 64 D9 01 03 26 F5 00 20 D8 22 21 7E 00 38 63 2A 1F 02 00 3D 99 8D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5F 16