

CyberAtom X-200

USER MANUAL

Copyrights © Softexor 2015
All Rights Reserved.

Contents

About.....	6
Block Diagram.....	6
Axes Conventions.....	6
System Startup.....	7
Selecting Power Source.....	7
Hardware Reset.....	7
Booting Stages.....	7
LED indicator.....	8
Getting Output Data.....	9
Orientation.....	9
Data-Ready Signal.....	9
Sensor Measurement Data.....	9
Settings.....	11
Calibration.....	12
Accelerometer Sensor.....	12
Magnetometer Sensor.....	12
Gyroscope Sensor.....	13
Tuning.....	14
Trusting Accelerometer Measurements.....	14
Trusting Magnetometer Measurements.....	14
I2C Interface.....	15
Request-Only Transaction.....	15
Request-Response Transaction.....	15
Request-Only Messages.....	16
REBOOT.....	16
RESET_GYR.....	16
WRITE_FLASH.....	16
Request-Response Messages.....	17
GET_SYS_INFO.....	17
GET_QUAT_DATA.....	18
GET_EULER_DATA.....	18
GET_ROT_RATE_DATA.....	18
GET_TEMP.....	18
SET_BAUD_RATE.....	19
SET_I2C_ADDR.....	19
GET_RAW_ACC.....	19
GET_RAW_MAG.....	20
GET_RAW_GYR.....	20
GET_NORM_ACC.....	21
GET_NORM_MAG.....	21
GET_NORM_GYR.....	21

GET_CALIB_ACC.....	21
GET_CALIB_MAG.....	22
GET_CALIB_GYR.....	22
GET_I2C_ADDR.....	22
GET_BAUD_RATE.....	22
UART Interface.....	24
Transmission Parameters.....	24
Message Format.....	24
Request Messages.....	25
GET_SYS_INFO.....	26
GET_QUAT_DATA.....	26
GET_EULER_DATA.....	26
GET_ROT_RATE_DATA.....	27
REBOOT.....	27
FACTORY_RESET.....	27
SET_ACC_CALIB_MAT.....	28
SET_MAG_CALIB_MAT.....	28
SET_GYR_CALIB_MAT.....	29
SET_FILTER_MAG.....	30
SET_FILTER_ACC.....	30
SET_FILTER_GYR.....	30
SET_FILTER_PROCN.....	31
GET_TEMP.....	31
SET_BAUD_RATE.....	31
SET_I2C_ADDR.....	32
RESET_GYR.....	32
WRITE_FLASH.....	33
GET_ACC_CALIB_MAT.....	33
GET_MAG_CALIB_MAT.....	33
GET_GYR_CALIB_MAT.....	34
GET_RAW_ACC.....	34
GET_RAW_MAG.....	34
GET_RAW_GYR.....	35
GET_NORM_ACC.....	35
GET_NORM_MAG.....	35
GET_NORM_GYR.....	36
GET_CALIB_ACC.....	36
GET_CALIB_MAG.....	36
GET_CALIB_GYR.....	36
REBOOT_BOOTLOADER.....	37
GET_FILTER_MAG.....	37
GET_FILTER_ACC.....	37
GET_FILTER_GYR.....	38
GET_FILTER_PROCN.....	38
GET_I2C_ADDR.....	38
GET_BAUD_RATE.....	39
Response Messages.....	39
SYS_INFO.....	40
QUAT_DATA.....	40
EULER_DATA.....	41
ROT_RATE_DATA.....	41
ACC_CALIB_MAT.....	41
MAG_CALIB_MAT.....	42
GYR_CALIB_MAT.....	42
FILTER_MAG.....	43
FILTER_ACC.....	43
FILTER_GYR.....	44
FILTER_PROCN.....	44

TEMP.....	45
BAUD_RATE.....	45
I2C_ADDR.....	45
CONFIRM.....	46
RAW_ACC.....	46
RAW_MAG.....	46
RAW_GYR.....	47
NORM_ACC.....	47
NORM_MAG.....	48
NORM_GYR.....	48
CALIB_ACC.....	48
CALIB_MAG.....	49
CALIB_GYR.....	49

USB Interface.....50

USB Message Format.....	50
Request Messages.....	50
GET_SYS_INFO.....	51
GET_QUAT_DATA.....	52
GET_EULER_DATA.....	52
GET_ROT_RATE_DATA.....	52
REBOOT.....	52
FACTORY_RESET.....	52
SET_ACC_CALIB_MAT.....	52
SET_MAG_CALIB_MAT.....	53
SET_GYR_CALIB_MAT.....	53
SET_FILTER_MAG.....	54
SET_FILTER_ACC.....	54
SET_FILTER_GYR.....	54
SET_FILTER_PROCN.....	55
GET_TEMP.....	55
SET_BAUD_RATE.....	55
SET_I2C_ADDR.....	56
RESET_GYR.....	56
WRITE_FLASH.....	56
GET_ACC_CALIB_MAT.....	56
GET_MAG_CALIB_MAT.....	56
GET_GYR_CALIB_MAT.....	56
GET_RAW_ACC.....	56
GET_RAW_MAG.....	57
GET_RAW_GYR.....	57
GET_NORM_ACC.....	57
GET_NORM_MAG.....	57
GET_NORM_GYR.....	57
GET_CALIB_ACC.....	57
GET_CALIB_MAG.....	57
GET_CALIB_GYR.....	58
REBOOT_BOOTLOADER.....	58
GET_FILTER_MAG.....	58
GET_FILTER_ACC.....	58
GET_FILTER_GYR.....	58
GET_FILTER_PROCN.....	58
GET_I2C_ADDR.....	58
GET_BAUD_RATE.....	58
GET_STATS.....	59
Response Messages.....	59
SYS_INFO.....	60
QUAT_DATA.....	60
EULER_DATA.....	60

ROT_RATE_DATA.....	61
ACC_CALIB_MAT.....	61
MAG_CALIB_MAT.....	61
GYR_CALIB_MAT.....	62
FILTER_MAG.....	62
FILTER_ACC.....	63
FILTER_GYR.....	63
FILTER_PROCN.....	63
TEMP.....	64
BAUD_RATE.....	64
I2C_ADDR.....	64
CONFIRM.....	64
RAW_ACC.....	64
RAW_MAG.....	65
RAW_GYR.....	65
NORM_ACC.....	65
NORM_MAG.....	65
NORM_GYR.....	66
CALIB_ACC.....	66
CALIB_MAG.....	66
CALIB_GYR.....	66
STATS.....	67
Troubleshooting.....	68
Factory Defaults.....	68
Configuring UDEV system on Linux Systems for CyberAtom Devices.....	68
Installing USB Driver for CyberAtom Devices on Windows.....	69
Identify CyberAtom Device Using USB Connection.....	70
Firmware Upgrade Using USB Interface.....	71
Firmware Upgrade Using UART Interface.....	71
Recovering Device Firmware.....	72
Legal Notice.....	73

About

CyberAtom X-200 AHRS (Attitude and Heading Reference System) is a small standalone embedded device that provides information about its orientation in 3D space.

The device is built with using MEMS 3-Axis inertial and magnetic sensors as well as on-board microcontroller. Special firmware installed is responsible for controlling sensor readings and perform advanced data fusion algorithm to estimate spatial orientation of the device in inertial reference frame.

The device provides several communication interfaces and simple protocol that makes it easily controllable by host microcontroller or PC computer.

High data update rate, very good accuracy and precision, as well as wide range of operational conditions make it applicable in many fields and types of engineering.

Block Diagram

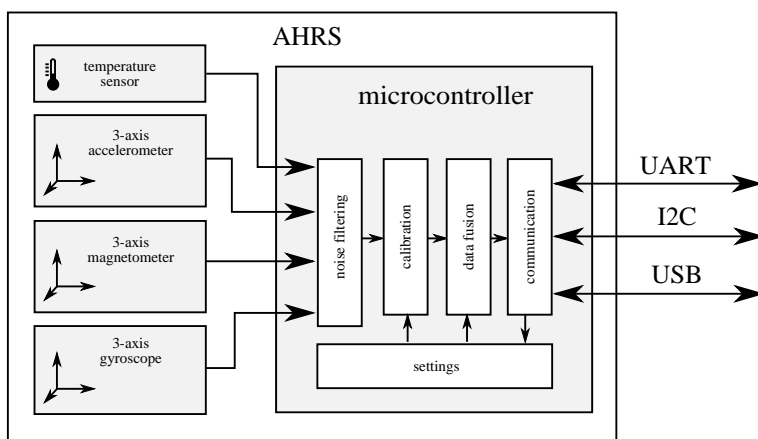


Figure 1: Block diagram of X-200 device.

Axes Conventions

CyberAtom device reports its orientation with regards to a reference frame following ENU axes conventions.

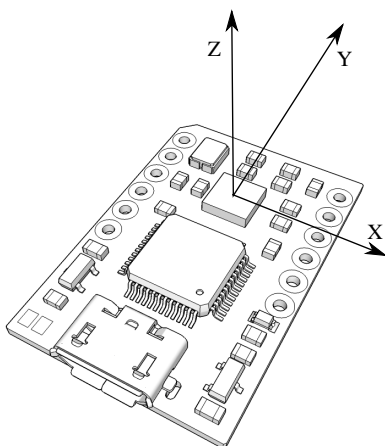


Figure 2: Axes Conventions for X-200 device.

System Startup

The device is designed to be powered from stabilized +3.3V source using one of VDD pins. It has built-in internal by-pass capacitors, so mounting external once aside the module is not necessary.

Selecting Power Source

In addition, X-200 AHRS device can be also powered from an USB port of a host computer. In this case the device is selecting power source as in the table below:

Table 1: Power Source Selection Logic

+3.3V at VDD pin	USB Connection	Powering Source
present	no	VDD pin
present	present	VDD pin
no	present	USB port
no	no	no

Hardware Reset

The device can be forced to reset using RESET pin. Low logic signal applied to this pin will cause the device to reset. Devices starts operating from when the RESET pin state is back at the high logic level (or remains unconnected)

Any changes in settings applied and not persisted in Flash memory are lost on device reset.

Booting Stages

When system is starting, it checks whether one of bootloader entry condition is fulfilled. These conditions are:

- JP1 jumper is short
- bootloader mode was requested during last normal operation

If neither of these conditions are met, the device switches to normal operation mode providing AHRS functionality. Otherwise, the devices enters bootloader mode offering firmware manipulation services.

The logic for booting state transitions is shown on the figure below.

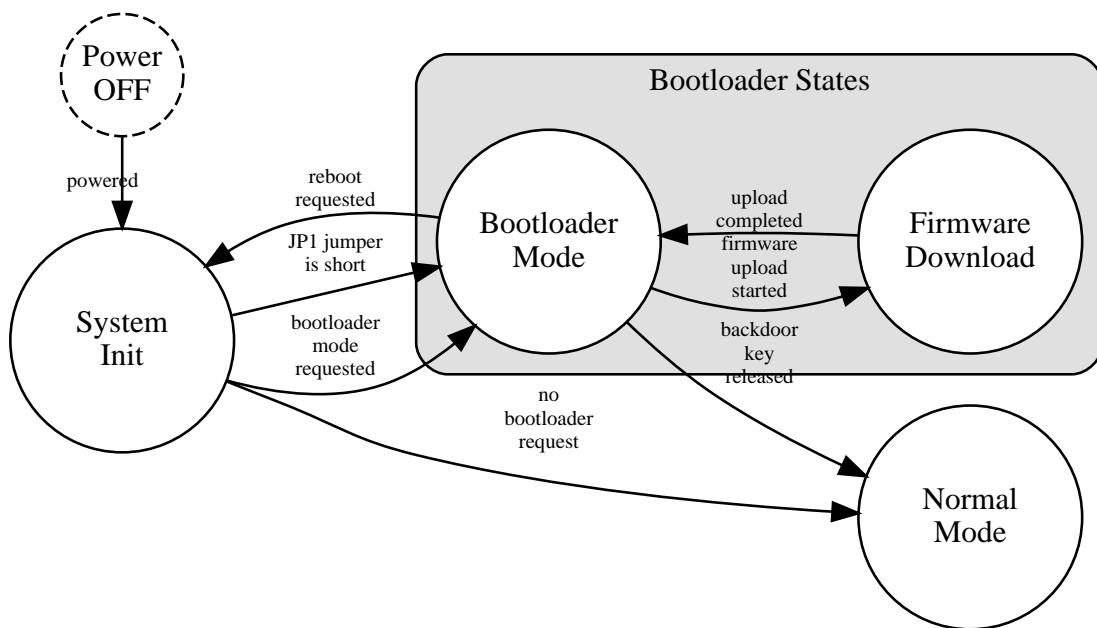


Figure 3: Booting States for CyberAtom AHRs Devices.

LED indicator

When powered on, the LED indicator blinks with frequency depending on the operational state as in table below.

Table 2: LED Behaviour

Operation State	LED Blink Behaviour
Normal Mode	blinking with approx. 1 Hz
Bootloader Mode	blinking with approx. 10 Hz
Firmware Download	blinking with approx. 10 Hz

Getting Output Data

Orientation

CyberAtom AHRS device reports its spatial orientation using one of two notations:

- quaternion,
- or Euler angles

They can be retrieved from the device over any communication interface using [GET_QUAT_DATA](#) and [GET_EULER_DATA](#) messages respectively.

Data-Ready Signal

CyberAtom AHRS device signals availability of new orientation data with pulling DRDY line into LOW logic state.

The DRDY line is kept LOW and goes back HIGH state only when orientation data is read by host with any of the following messages:

- [GET_QUAT_DATA](#)
- [GET_EULER_DATA](#)

Any communication interface that supports these messages can be used to reset the state of DRDY line.

Latching LOW state of the DRDY line does not cause latch on orientation data reported by the AHRS device. Messages reporting sensor's orientation will at any point return the latest available orientation data.

Sensor Measurement Data

The AHRS device can report data it retrieves from on-board sensors, such as 3-axis accelerometer, 3-axis magnetometer and 3-axis gyroscope.

Each sensor report data in a **raw** form which is then **normalized** (to appropriate physical unit) and then **calibrated**, using calibration matrixes to correct missalignment, bias and temperature drift.

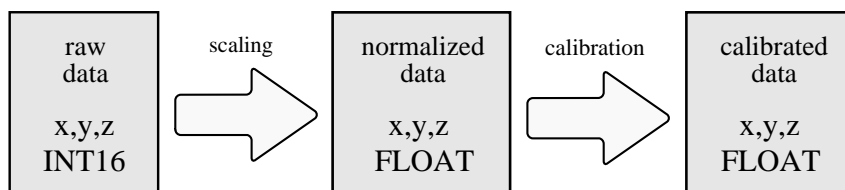


Figure 4: Sensor Data Processing Sequence.

Data from on-board sensors can be retrieved in any of these three forms over any communication interface using messages as in the table below.

Table 3: Sensor Data Request Messages

Sensor	Data Type	Communication Message to Use
accelerometer	raw	GET_RAW_ACC
accelerometer	normalized	GET_NORM_ACC
accelerometer	calibrated	GET_CALIB_ACC

Sensor	Data Type	Communication Message to Use
magnetometer	raw	GET_RAW_MAG
magnetometer	normalized	GET_NORM_MAG
magnetometer	calibrated	GET_CALIB_MAG
gyroscope	raw	GET_RAW_ACC
gyroscope	normalized	GET_NORM_GYR
gyroscope	calibrated	GET_CALIB_GYR

Settings

The performance of the device working in the field depends on several parameters used by the firmware at the runtime. They are referred as device settings.

During normal operation the settings data is available in various forms and stored in various types of memory. They are:

- RAM-stored settings
- Flash-stored settings
- Read-only factory defaults settings

RAM-stored Settings

AHRS processing algorithm relies on values from RAM-stored settings.

RAM-stored settings values are initialized from Flash-stored settings on each system power-on/reset event.

Any changes to those are immediately reflected in the way orientation data is being calculated.

All settings-related messages sent to the device manipulate on RAM-stored settings.

RAM settings are volatile. Any changes applied to those are lost on reset or power-off events.

Saving and Restoring Settings

It is possible at any time to load factory default settings to RAM using [FACTORY_RESET](#) message.

It is possible at any time to write current values of RAM-stored settings into Flash memory using [WRITE_FLASH](#) message.

Calibration

Quality operation of the device heavily relies on proper on-board sensors calibration.

X-200 AHR device is produced with pre-calibrated data that are valid for stand-alone operation. It has to be noted that assembling the device in larger electronic and/or electromechanical system can change operational conditions to the extent where re-calibration is necessary to achieve desired output data accuracy.

This chapter introduces background information on calibration technique for X-200 AHRS device.

All normalized on-board sensor measurements are re-calculated for compensating assembly misalignments, bias and temperature bias drift.

Accelerometer Sensor

Normalized on-board accelerometer measurements compound of x,y,z components taken for an orientation estimation are re-calculated using two calibration matrices:

$$\vec{a}_{calib} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \cdot \left(\vec{a}_{norm} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \Delta T \\ \Delta T^2 \end{bmatrix} \right)$$

where:

$c_{11}..c_{33}$ are misalignment compensation matrix elements

$b_{11}..b_{33}$ are bias compensation matrix elements

#T is a temperature between actual operation temperature and reference temperature of 25 °C.

Both matrices can be retrieved from the device using [GET_ACC_CALIB](#) message and can be changed by sending new values with [SET_ACC_CALIB](#) message.

Magnetometer Sensor

Normalized on-board magnetometer measurements compound of x,y,z components are taken for an orientation estimation are re-calculated using two calibration matrices:

$$\vec{m}_{calib} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \cdot \left(\vec{m}_{norm} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \Delta T \\ \Delta T^2 \end{bmatrix} \right)$$

where:

$c_{11}..c_{33}$ are misalignment compensation matrix elements

$b_{11}..b_{33}$ are bias compensation matrix elements

#T is a temperature between actual operation temperature and reference temperature of 25 °C.

Both matrices can be retrieved from the device using [GET_MAG_CALIB](#) message and can be changed by sending new values with [SET_MAG_CALIB](#) message.

Gyroscope Sensor

Normalized on-board gyroscope measurements compound of x,y,z components taken for an orientation estimation are re-calculated using two calibration matrices:

$$\vec{\omega}_{calib} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \cdot \left(\vec{\omega}_{norm} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ \Delta T \\ \Delta T^2 \end{bmatrix} \right)$$

where:

$c_{11}..c_{33}$ are misalignment compensation matrix elements

$b_{11}..b_{33}$ are bias compensation matrix elements

#T is a temerature between acutal operation temperature and reference temperature of 25 °C.

Both matrices can be retrieved from the device using [GET_GYR_CALIB](#) message and can be changed by sending new values with [SET_GYR_CALIB](#) message.

Tuning

X-200 AHRS device performs data fusion using modified implementation of Extended Kalman Filter. For best performance and accuracy of the output data, it is necessary to ensure filter parameters are best for given usage and operating conditions.

Filter tuning is performed by manipulating number of coefficients related to measurements from each of on-board MEMS sensor: accelerometer, gyroscope and magnetometer.

Trusting Accelerometer Measurements

The primary role of on-board accelerometer sensor is to continuously correct estimated orientation using Earth gravitational as reference of vertical direction. However, due to its nature, accelerometer sensor is a subject of distortion whenever vibration and/or other than gravity types of acceleration are applied to the AHRS frame of reference.

In case where accelerometer can no longer be assumed to measure only components of Earth gravity vector, trust to its measurements should be reduced to avoid distorting AHRS estimated orientation.

The trust to each of accelerometer measurement component (x,y,z) can be reduced by increasing values of accelerometer measurement matrix coefficients in relation to similar coefficients for gyroscope and magnetometer sensors.

Analogically, when AHRS is operating in environment where no distortions are applied to accelerometer sensor, AHRS performance can be improved by lowering values of accelerometer-related coefficients.

The accelerometer measurement coefficients can be changed with use of SET_FILTER_ACC message.

Trusting Magnetometer Measurements

The primary role of on-board magnetometer sensor is to continuously correct estimated heading using Earth magnetic field as reference of North-South direction. Magnetometer measurement however can be a subject of various distortions whenever foreign sources of magnetism are around.

In case where magnetometer measurements can no longer be assumed to be free from distortion from non-Earth magnetic fields, trust to its measurements should be reduced to reduce impact of these to AHRS orientation estimations.

The trust to each of magnetometer measurement component (x,y,z) can be reduced by increasing values of magnetometer measurement matrix coefficients in relation to similar coefficients for accelerometer and gyroscope sensors.

Analogically, when AHRS is operating in environment where no distortions are applied to magnetometer sensor, AHRS performance can be improved by lowering values of magnetometer-related coefficients.

The magnetometer measurement coefficients can be changed with use of SET_FILTER_MAG message.

I2C Interface

X-200 AHRS device can work as a slave device connected to an I2C bus.

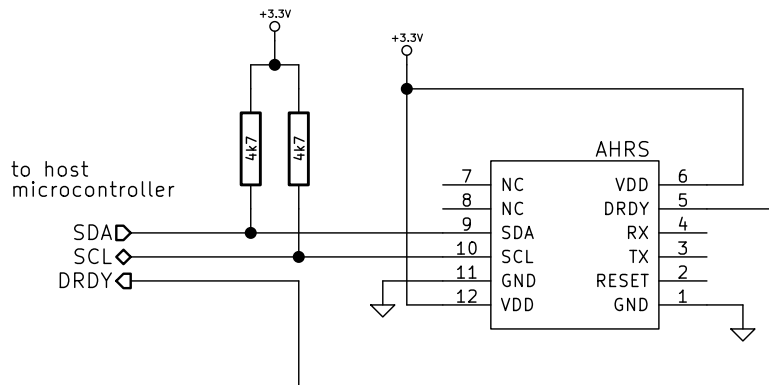


Figure 5: Example schematics of CyberAtom as an I2C slave device.

By default the device uses 0x30 as a slave address.

I2C interface can handle **request-only** as well as **request-response** transactions. There is a reduced set of messages supported in comparison to UART interface - no calibration nor tuning related messages are available on I2C interface.

Chapters below present these transactions in details.

Request-Only Transaction

The following diagram presents transaction of sending a request message the X-200 device.

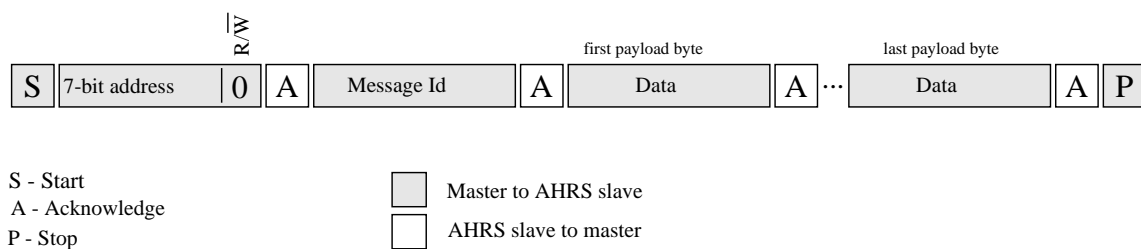


Figure 6: I2C request-only message transaction

Request-Response Transaction

The following diagram presents two-step operation for retrieving response message from the X-200 device.

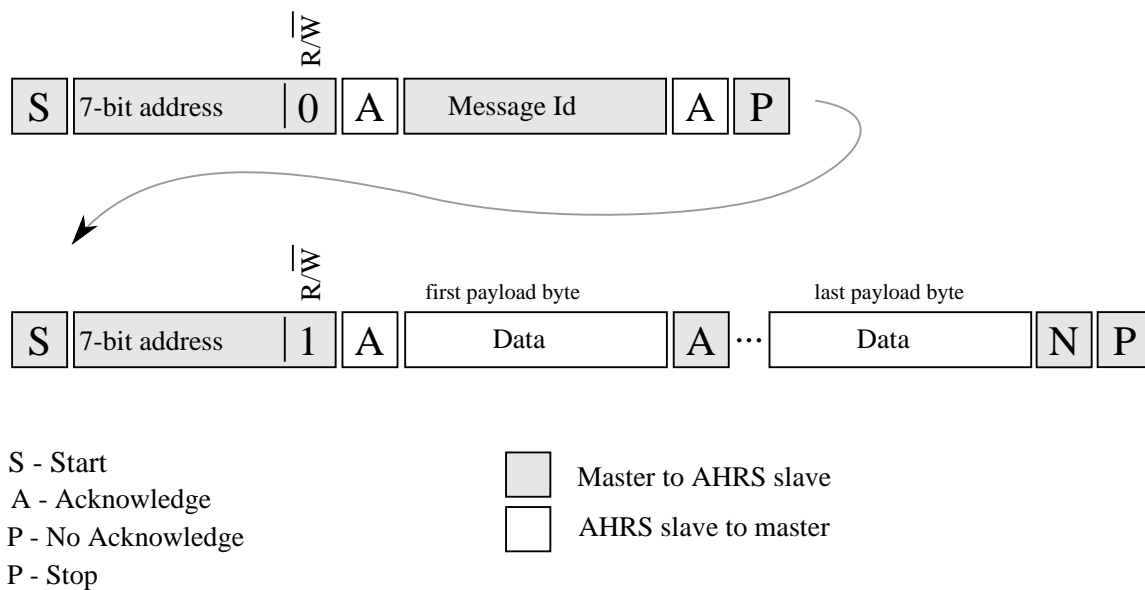


Figure 7: I2C request-response message transaction

Request-Only Messages

Table 4: List of I2C request messages.

Message	Message ID (hex)	Description
<i>REBOOT</i>	0x05	Reboots the device.
<i>RESET_GYR</i>	0x15	Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).
<i>WRITE_FLASH</i>	0x16	Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.

REBOOT

Message ID: **0x05**

Description: Reboots the device.

Request Data Bytes: 0

RESET_GYR

Message ID: **0x15**

Description: Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).

Request Data Bytes: 0

WRITE_FLASH

Message ID: **0x16**

Description: Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.

Request Data Bytes: 0

Request-Response Messages

Table 5: List of I2C request messages.

Message	Message ID (hex)	Description
GET_SYS_INFO	0x01	Requests sending back information on the device type and firmware version.
GET_QUAT_DATA	0x02	Requests sending back actual orientation in quaternion representation.
GET_EULER_DATA	0x03	Requests sending back actual orientation in Euler angles representation.
GET_ROT_RATE_DATA	0x04	Requests sending back actual rotation rates.
GET_TEMP	0x0F	Requests sending current temperature reading
SET_BAUD_RATE	0x10	Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
SET_I2C_ADDR	0x11	Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
GET_RAW_ACC	0x20	Requests sending raw accelerometer measurement data.
GET_RAW_MAG	0x21	Requests sending raw magnetometer measurement data.
GET_RAW_GYR	0x22	Requests sending raw gyroscope measurement data.
GET_NORM_ACC	0x23	Requests sending normalized accelerometer measurement data.
GET_NORM_MAG	0x24	Requests sending normalized magnetometer measurement data.
GET_NORM_GYR	0x25	Requests sending normalized gyroscope measurement data.
GET_CALIB_ACC	0x26	Requests sending calibrated accelerometer measurement data.
GET_CALIB_MAG	0x27	Requests sending calibrated magnetometer measurement data.
GET_CALIB_GYR	0x28	Requests sending calibrated gyroscope measurement data.
GET_I2C_ADDR	0x30	Requests currently set I2C address.
GET_BAUD_RATE	0x31	Requests currently set baud rate.

GET_SYS_INFO

Message ID: **0x01**

Description: Requests sending back information on the device type and firmware version.

Request Data Bytes: 0

Response data Bytes: 32

Table 6: Structure of response data bytes for GET_SYS_INFO message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	null-terminated string	(varies)	Device type identifier.

Byte Offset	Data Type	Value (hex)	Description
8	null-terminated string	(varies)	Firmware release version in MAJOR.MINOR.PATCH format, optionally extended with other symbols.

GET_QUAT_DATA

Message ID: **0x02**

Description: Requests sending back actual orientation in quaternion representation.

Request Data Bytes: 0

Response data Bytes: 16

Table 7: Structure of response data bytes for GET_QUAT_DATA message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	q0 component
4	IEEE754 float	(varies)	q1 component
8	IEEE754 float	(varies)	q2 component
12	IEEE754 float	(varies)	q3 component

GET_EULER_DATA

Message ID: **0x03**

Description: Requests sending back actual orientation in Euler angles representation.

Request Data Bytes: 0

Response data Bytes: 12

Table 8: Structure of response data bytes for GET_EULER_DATA message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	Pitch angle in degrees.
4	IEEE754 float	(varies)	Roll angle in degrees.
8	IEEE754 float	(varies)	Yaw angle in degrees.

GET_ROT_RATE_DATA

Message ID: **0x04**

Description: Requests sending back actual rotation rates.

Request Data Bytes: 0

Response data Bytes: 12

Table 9: Structure of response data bytes for GET_ROT_RATE_DATA message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	Rotation rate over X axis in degrees/second.
4	IEEE754 float	(varies)	Rotation rate over Y axis in degrees/second.
8	IEEE754 float	(varies)	Rotation rate over Z axis in degrees/second.

GET_TEMP

Message ID: **0x0F**

Description: Requests sending current temperature reading

Request Data Bytes: 0

Response data Bytes: 4

Table 10: Structure of response data bytes for GET_TEMP message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	Actual temperature value in Celsius degrees.

SET_BAUD_RATE

Message ID: **0x10**

Description: Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Request Data Bytes: 1

Table 11: Structure of request data bytes for SET_BAUD_RATE message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	New baud rate.

Response data Bytes: 1

Table 12: Structure of response data bytes for SET_BAUD_RATE message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	Current baud rate.

SET_I2C_ADDR

Message ID: **0x11**

Description: Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Request Data Bytes: 1

Table 13: Structure of request data bytes for SET_I2C_ADDR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	New 7-bit I2C slave address for the device.

Response data Bytes: 1

Table 14: Structure of response data bytes for SET_I2C_ADDR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	7-bit I2C slave device address.

GET_RAW_ACC

Message ID: **0x20**

Description: Requests sending raw accelerometer measurement data.

Request Data Bytes: 0

Response data Bytes: 6

Table 15: Structure of response data bytes for GET_RAW_ACC message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	16-bit signed, two's complement	(varies)	x component of acclerometer measurement
2	16-bit signed, two's complement	(varies)	y component of acclerometer measurement
4	16-bit signed, two's complement	(varies)	z component of acclerometer measurement

GET_RAW_MAG

Message ID: **0x21**

Description: Requests sending raw magnetometer measurement data.

Request Data Bytes: 0

Response data Bytes: 6

Table 16: Structure of response data bytes for GET_RAW_MAG message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	16-bit signed, two's complement	(varies)	x component of magnetometer measurement
2	16-bit signed, two's complement	(varies)	y component of magnetometer measurement
4	16-bit signed, two's complement	(varies)	z component of magnetometer measurement

GET_RAW_GYR

Message ID: **0x22**

Description: Requests sending raw gyroscope measurement data.

Request Data Bytes: 0

Response data Bytes: 6

Table 17: Structure of response data bytes for GET_RAW_GYR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	16-bit signed, two's complement	(varies)	x component of gyroscope measurement
2	16-bit signed, two's complement	(varies)	y component of gyroscope measurement
4	16-bit signed, two's complement	(varies)	z component of gyroscope measurement

GET_NORM_ACC

Message ID: **0x23**

Description: Requests sending normalized accelerometer measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 18: Structure of response data bytes for GET_NORM_ACC message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of accelerometer measurement
4	IEEE754 float	(varies)	y component of accelerometer measurement
8	IEEE754 float	(varies)	z component of accelerometer measurement

GET_NORM_MAG

Message ID: **0x24**

Description: Requests sending normalized magnetometer measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 19: Structure of response data bytes for GET_NORM_MAG message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of magnetometer measurement
4	IEEE754 float	(varies)	y component of magnetometer measurement
8	IEEE754 float	(varies)	z component of magnetometer measurement

GET_NORM_GYR

Message ID: **0x25**

Description: Requests sending normalized gyroscope measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 20: Structure of response data bytes for GET_NORM_GYR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of gyroscope measurement
4	IEEE754 float	(varies)	y component of gyroscope measurement
8	IEEE754 float	(varies)	z component of gyroscope measurement

GET_CALIB_ACC

Message ID: **0x26**

Description: Requests sending calibrated accelerometer measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 21: Structure of response data bytes for GET_CALIB_ACC message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of acclerometer measurement
4	IEEE754 float	(varies)	y component of acclerometer measurement
8	IEEE754 float	(varies)	z component of acclerometer measurement

GET_CALIB_MAG

Message ID: **0x27**

Description: Requests sending calibrated magnetometer measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 22: Structure of response data bytes for GET_CALIB_MAG message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of magnetometer measurement
4	IEEE754 float	(varies)	y component of magnetometer measurement
8	IEEE754 float	(varies)	z component of magnetometer measurement

GET_CALIB_GYR

Message ID: **0x28**

Description: Requests sending calibrated gyroscope measurement data.

Request Data Bytes: 0

Response data Bytes: 12

Table 23: Structure of response data bytes for GET_CALIB_GYR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	IEEE754 float	(varies)	x component of gyroscope measurement
4	IEEE754 float	(varies)	y component of gyroscope measurement
8	IEEE754 float	(varies)	z component of gyroscope measurement

GET_I2C_ADDR

Message ID: **0x30**

Description: Requests currently set I2C address.

Request Data Bytes: 0

Response data Bytes: 1

Table 24: Structure of response data bytes for GET_I2C_ADDR message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	7-bit I2C slave device address.

GET_BAUD_RATE

Message ID: **0x31**

Description: Requests currently set baud rate.

Request Data Bytes: 0

Response data Bytes: 1

Table 25: Structure of response data bytes for GET_BAUD_RATE message on I2C interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	(varies)	Current baud rate.

UART Interface

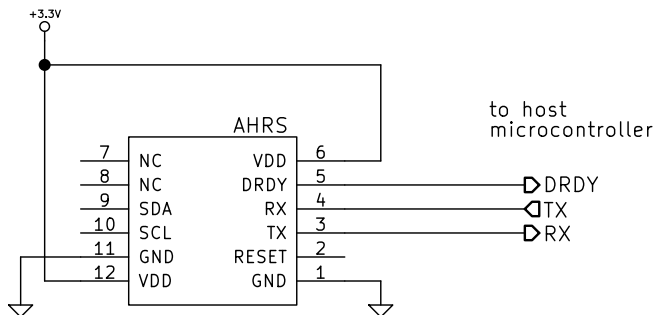


Figure 8: Example schematics of CyberAtom as an UART controlled device.

Transmission Parameters

The device operates with 8-N-1 serial transmission parameters.

The baud rate is configurable and can be changed remotely using any of communication interface.

The factory-default baud rate is 57600.

Message Format

Communication over serial port is handled in the form of data packets forming individual messages to and from the device.

Messages sent by a host device (microcontroller, PC etc.) to the AHRS device are referred as **request** messages, while messages that AHRS device sends to the host device are referred as **response** messages.

Both message types have the same structure and they are composed of:

- two header bytes identifying message data packet
- message ID byte
- two bytes value (little endian) containing payload length
- a number of payload data bytes
- single byte of a payload checksum

Any 16-bit or 32-bit integer values have little endian encoding.

The checksum value is a calculated byte-base sum of all message bytes (except the checksum one) modulo 256.

Table 26: Serial protocol message structure

Header Bytes	Message ID	Payload Length	Payload	Checksum
0x05 0xD3	byte	low_byte, high_byte	...	byte

Request Messages

Table 27: List of UART request messages.

Message	Message ID (hex)	Description
<i>GET_SYS_INFO</i>	0x01	Requests sending back information on the device type and firmware version.
<i>GET_QUAT_DATA</i>	0x02	Requests sending back actual orientation in quaternion representation.
<i>GET_EULER_DATA</i>	0x03	Requests sending back actual orientation in Euler angles representation.
<i>GET_ROT_RATE_DATA</i>	0x04	Requests sending back actual rotation rates.
<i>REBOOT</i>	0x05	Reboots the device.
<i>FACTORY_RESET</i>	0x07	Brings factory defaults settings to device RAM.
<i>SET_ACC_CALIB_MAT</i>	0x08	Sends calibration matrices for the accelerometer sensor.
<i>SET_MAG_CALIB_MAT</i>	0x09	Sends calibration matrices for the magnetometer sensor.
<i>SET_GYR_CALIB_MAT</i>	0x0A	Sends calibration matrices for the gyroscope sensor.
<i>SET_FILTER_MAG</i>	0x0B	Sends new values for measurement filter magnetometer matrix.
<i>SET_FILTER_ACC</i>	0x0C	Sends new values for measurement filter accelerometer matrix.
<i>SET_FILTER_GYR</i>	0x0D	Sends new values for measurement filter gyroscope matrix.
<i>SET_FILTER_PROCN</i>	0x0E	Sends new values for filter process noise matrix.
<i>GET_TEMP</i>	0x0F	Requests sending current temperature reading
<i>SET_BAUD_RATE</i>	0x10	Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
<i>SET_I2C_ADDR</i>	0x11	Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
<i>RESET_GYR</i>	0x15	Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).
<i>WRITE_FLASH</i>	0x16	Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.
<i>GET_ACC_CALIB_MAT</i>	0x17	Requests sending back actual accelerometer calibration settings.
<i>GET_MAG_CALIB_MAT</i>	0x18	Requests sending back actual magnetometer calibration settings.
<i>GET_GYR_CALIB_MAT</i>	0x19	Requests sending back actual gyroscope calibration settings.
<i>GET_RAW_ACC</i>	0x20	Requests sending raw accelerometer measurement data.
<i>GET_RAW_MAG</i>	0x21	Requests sending raw magnetometer measurement data.
<i>GET_RAW_GYR</i>	0x22	Requests sending raw gyroscope measurement data.
<i>GET_NORM_ACC</i>	0x23	Requests sending normalized accelerometer measurement data.
<i>GET_NORM_MAG</i>	0x24	Requests sending normalized magnetometer measurement data.
<i>GET_NORM_GYR</i>	0x25	Requests sending normalized gyroscope measurement data.
<i>GET_CALIB_ACC</i>	0x26	Requests sending calibrated accelerometer measurement data.
<i>GET_CALIB_MAG</i>	0x27	Requests sending calibrated magnetometer measurement data.

Message	Message ID (hex)	Description
GET_CALIB_GYR	0x28	Requests sending calibrated gyroscope measurement data.
REBOOT_BOOTLOADER	0x29	Requests rebooting to the bootloader mode
GET_FILTER_MAG	0x2B	Requests values for measurement filter magnetometer matrix.
GET_FILTER_ACC	0x2C	Requests values for measurement filter accelerometer matrix.
GET_FILTER_GYR	0x2D	Requests values for measurement filter gyroscope matrix.
GET_FILTER_PROCN	0x2E	Requests values for filter process noise matrix.
GET_I2C_ADDR	0x30	Requests currently set I2C address.
GET_BAUD_RATE	0x31	Requests currently set baud rate.

GET_SYS_INFO

Message ID: **0x01**

Description: Requests sending back information on the device type and firmware version.

Message size: 6 bytes

When device receives this message, it will reply with sending back [SYS_INFO](#) message.

Table 28: Structure of complete GET_SYS_INFO message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x01	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0D9	Checksum byte.

GET_QUAT_DATA

Message ID: **0x02**

Description: Requests sending back actual orientation in quaternion representation.

Message size: 6 bytes

When device receives this message, it will reply with sending back [QUAT_DATA](#) message.

Table 29: Structure of complete GET_QUAT_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x02	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0DA	Checksum byte.

GET_EULER_DATA

Message ID: **0x03**

Description: Requests sending back actual orientation in Euler angles representation.

Message size: 6 bytes

When device receives this message, it will reply with sending back [EULER_DATA](#) message.

Table 30: Structure of complete GET_EULER_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x03	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0DB	Checksum byte.

GET_ROT_RATE_DATA

Message ID: **0x04**

Description: Requests sending back actual rotation rates.

Message size: 6 bytes

When device receives this message, it will reply with sending back [ROT_RATE_DATA](#) message.

Table 31: Structure of complete GET_ROT_RATE_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x04	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0DC	Checksum byte.

REBOOT

Message ID: **0x05**

Description: Reboots the device.

Message size: 6 bytes

Table 32: Structure of complete REBOOT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x05	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0DD	Checksum byte.

FACTORY_RESET

Message ID: **0x07**

Description: Brings factory defaults settings to device RAM.

Message size: 6 bytes

Table 33: Structure of complete FACTORY_RESET message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x07	Message ID.

Byte Offset	Data Type	Value (hex)	Description
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0DF	Checksum byte.

SET_ACC_CALIB_MAT

Message ID: **0x08**

Description: Sends calibration matrices for the accelerometer sensor.

Message size: 78 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 34: Structure of complete SET_ACC_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x08	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

SET_MAG_CALIB_MAT

Message ID: **0x09**

Description: Sends calibration matrices for the magnetometer sensor.

Message size: 78 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 35: Structure of complete SET_MAG_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x09	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component

Byte Offset	Data Type	Value (hex)	Description
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

SET_GYR_CALIB_MAT

Message ID: **0x0A**

Description: Sends calibration matrices for the gyroscope sensor.

Message size: 78 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 36: Structure of complete SET_GYR_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0A	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

SET_FILTER_MAG

Message ID: **0x0B**

Description: Sends new values for measurement filter magnetometer matrix.

Message size: 18 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 37: Structure of complete SET_FILTER_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0B	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

SET_FILTER_ACC

Message ID: **0x0C**

Description: Sends new values for measurement filter accelerometer matrix.

Message size: 18 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 38: Structure of complete SET_FILTER_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0C	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

SET_FILTER_GYR

Message ID: **0x0D**

Description: Sends new values for measurement filter gyroscope matrix.

Message size: 18 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 39: Structure of complete SET_FILTER_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0D	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.

Byte Offset	Data Type	Value (hex)	Description
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

SET_FILTER_PROCN

Message ID: **0x0E**

Description: Sends new values for filter process noise matrix.

Message size: 34 bytes

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 40: Structure of complete SET_FILTER_PROCN message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0E	Message ID.
3	16-bit unsigned	0x01C	Number of bytes in payload.
5	IEEE754 float	(varies)	q11 matrix component
9	IEEE754 float	(varies)	q22 matrix component
13	IEEE754 float	(varies)	q33 matrix component
17	IEEE754 float	(varies)	q44 matrix component
21	IEEE754 float	(varies)	q55 matrix component
25	IEEE754 float	(varies)	q66 matrix component
29	IEEE754 float	(varies)	q77 matrix component
33	8-bit unsigned	(varies)	Checksum byte.

GET_TEMP

Message ID: **0x0F**

Description: Requests sending current temperature reading

Message size: 6 bytes

When device receives this message, it will reply with sending back *TEMP* message.

Table 41: Structure of complete GET_TEMP message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x0F	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0E7	Checksum byte.

SET_BAUD_RATE

Message ID: **0x10**

Description: Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Message size: 7 bytes

When device receives this message, it will reply with sending back [BAUD_RATE](#) message.

Table 42: Structure of complete SET_BAUD_RATE message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x10	Message ID.
3	16-bit unsigned	0x01	Number of bytes in payload.
5	8-bit unsigned	(varies)	New baud rate. One of the following values: 0x01 - 2400 bauds 0x02 - 4800 bauds 0x03 - 9600 bauds 0x04 - 19200 bauds 0x05 - 38400 bauds 0x06 - 57600 bauds 0x07 - 115200 bauds 0x08 - 230400 bauds 0x09 - 576000 bauds 0x0A - 921600 bauds
6	8-bit unsigned	(varies)	Checksum byte.

SET_I2C_ADDR

Message ID: **0x11**

Description: Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Message size: 7 bytes

When device receives this message, it will reply with sending back [I2C_ADDR](#) message.

Table 43: Structure of complete SET_I2C_ADDR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x11	Message ID.
3	16-bit unsigned	0x01	Number of bytes in payload.
5	8-bit unsigned	(varies)	New 7-bit I2C slave address for the device.
6	8-bit unsigned	(varies)	Checksum byte.

RESET_GYR

Message ID: **0x15**

Description: Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).

Message size: 6 bytes

Table 44: Structure of complete RESET_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x15	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0ED	Checksum byte.

WRITE_FLASH

Message ID: **0x16**

Description: Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.

Message size: 6 bytes

Table 45: Structure of complete WRITE_FLASH message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x16	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0EE	Checksum byte.

GET_ACC_CALIB_MAT

Message ID: **0x17**

Description: Requests sending back actual accelerometer calibration settings.

Message size: 6 bytes

When device receives this message, it will reply with sending back [ACC_CALIB_MAT](#) message.

Table 46: Structure of complete GET_ACC_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x17	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0EF	Checksum byte.

GET_MAG_CALIB_MAT

Message ID: **0x18**

Description: Requests sending back actual magnetometer calibration settings.

Message size: 6 bytes

When device receives this message, it will reply with sending back [MAG_CALIB_MAT](#) message.

Table 47: Structure of complete GET_MAG_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.

Byte Offset	Data Type	Value (hex)	Description
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x18	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0F0	Checksum byte.

GET_GYR_CALIB_MAT

Message ID: **0x19**

Description: Requests sending back actual gyroscope calibration settings.

Message size: 6 bytes

When device receives this message, it will reply with sending back [GYR_CALIB_MAT](#) message.

Table 48: Structure of complete GET_GYR_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x19	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0F1	Checksum byte.

GET_RAW_ACC

Message ID: **0x20**

Description: Requests sending raw accelerometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [RAW_ACC](#) message.

Table 49: Structure of complete GET_RAW_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x20	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0F8	Checksum byte.

GET_RAW_MAG

Message ID: **0x21**

Description: Requests sending raw magnetometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [RAW_MAG](#) message.

Table 50: Structure of complete GET_RAW_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x21	Message ID.

Byte Offset	Data Type	Value (hex)	Description
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0F9	Checksum byte.

GET_RAW_GYR

Message ID: **0x22**

Description: Requests sending raw gyroscope measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [RAW_GYR](#) message.

Table 51: Structure of complete GET_RAW_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x22	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FA	Checksum byte.

GET_NORM_ACC

Message ID: **0x23**

Description: Requests sending normalized accelerometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [NORM_ACC](#) message.

Table 52: Structure of complete GET_NORM_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x23	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FB	Checksum byte.

GET_NORM_MAG

Message ID: **0x24**

Description: Requests sending normalized magnetometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [NORM_MAG](#) message.

Table 53: Structure of complete GET_NORM_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x24	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FC	Checksum byte.

GET_NORM_GYR

Message ID: **0x25**

Description: Requests sending normalized gyroscope measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [NORM_GYR](#) message.

Table 54: Structure of complete GET_NORM_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x25	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FD	Checksum byte.

GET_CALIB_ACC

Message ID: **0x26**

Description: Requests sending calibrated accelerometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [CALIB_ACC](#) message.

Table 55: Structure of complete GET_CALIB_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x26	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FE	Checksum byte.

GET_CALIB_MAG

Message ID: **0x27**

Description: Requests sending calibrated magnetometer measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [CALIB_MAG](#) message.

Table 56: Structure of complete GET_CALIB_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x27	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x0FF	Checksum byte.

GET_CALIB_GYR

Message ID: **0x28**

Description: Requests sending calibrated gyroscope measurement data.

Message size: 6 bytes

When device receives this message, it will reply with sending back [CALIB_GYR](#) message.

Table 57: Structure of complete GET_CALIB_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x28	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x00	Checksum byte.

REBOOT_BOOTLOADER

Message ID: **0x29**

Description: Requests rebooting to the bootloader mode

Message size: 6 bytes

Table 58: Structure of complete REBOOT_BOOTLOADER message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x29	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x01	Checksum byte.

GET_FILTER_MAG

Message ID: **0x2B**

Description: Requests values for measurement filter magnetometer matrix.

Message size: 6 bytes

When device receives this message, it will reply with sending back [FILTER_MAG](#) message.

Table 59: Structure of complete GET_FILTER_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x2B	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x03	Checksum byte.

GET_FILTER_ACC

Message ID: **0x2C**

Description: Requests values for measurement filter accelerometer matrix.

Message size: 6 bytes

When device receives this message, it will reply with sending back [FILTER_ACC](#) message.

Table 60: Structure of complete GET_FILTER_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x2C	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x04	Checksum byte.

GET_FILTER_GYR

Message ID: **0x2D**

Description: Requests values for measurement filter gyroscope matrix.

Message size: 6 bytes

When device receives this message, it will reply with sending back [FILTER_GYR](#) message.

Table 61: Structure of complete GET_FILTER_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x2D	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x05	Checksum byte.

GET_FILTER_PROCN

Message ID: **0x2E**

Description: Requests values for filter process noise matrix.

Message size: 6 bytes

When device receives this message, it will reply with sending back [FILTER_PROCN](#) message.

Table 62: Structure of complete GET_FILTER_PROCN message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x2E	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x06	Checksum byte.

GET_I2C_ADDR

Message ID: **0x30**

Description: Requests currently set I2C address.

Message size: 6 bytes

When device receives this message, it will reply with sending back [I2C_ADDR](#) message.

Table 63: Structure of complete GET_I2C_ADDR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x30	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x08	Checksum byte.

GET_BAUD_RATE

Message ID: **0x31**

Description: Requests currently set baud rate.

Message size: 6 bytes

When device receives this message, it will reply with sending back [BAUD_RATE](#) message.

Table 64: Structure of complete GET_BAUD_RATE message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x31	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x09	Checksum byte.

Response Messages

Table 65: List of UART response messages.

Message	Message ID (hex)	Description
SYS_INFO	0x81	Provides information on the device type and installed firmware version.
QUAT_DATA	0x82	Provides actual orientation data in quaternion representation.
EULER_DATA	0x83	Provides actual orientation data in Euler angles representation.
ROT_RATE_DATA	0x84	Provides actual rotation rate data.
ACC_CALIB_MAT	0x88	Reports calibration matrix for the accelerometer sensor.
MAG_CALIB_MAT	0x89	Reports calibration matrix for the magnetometer sensor.
GYR_CALIB_MAT	0x8A	Reports calibration matrix for the gyroscope sensor.
FILTER_MAG	0x8B	Reports values of measurement filter magnetometer matrix.
FILTER_ACC	0x8C	Reports values of measurement filter accelerometer matrix.
FILTER_GYR	0x8D	Reports values of measurement filter gyroscope matrix.
FILTER_PROCN	0x8E	Reports values of filter process noise matrix.
TEMP	0x8F	Contains current temperature reading.
BAUD_RATE	0x90	Reports currently set baud rate.
I2C_ADDR	0x91	Reports actual I2C slave address.
CONFIRM	0x92	Confirms successful settings change operation.
RAW_ACC	0xA0	Provides raw latest accelerometer measurements as x,y,z components.

Message	Message ID (hex)	Description
<i>RAW_MAG</i>	0xA1	Provides raw latest magnetometer measurements as x,y,z components.
<i>RAW_GYR</i>	0xA2	Provides raw latest gyroscope measurements as x,y,z components.
<i>NORM_ACC</i>	0xA3	Provides latest normalized accelerometer measurements as x,y,z components.
<i>NORM_MAG</i>	0xA4	Provides latest normalized magnetometer measurements as x,y,z components.
<i>NORM_GYR</i>	0xA5	Provides latest normalized gyroscope measurements as x,y,z components.
<i>CALIB_ACC</i>	0xA6	Provides latest calibrated accelerometer measurements as x,y,z components.
<i>CALIB_MAG</i>	0xA7	Provides latest calibrated magnetometer measurements as x,y,z components.
<i>CALIB_GYR</i>	0xA8	Provides latest calibrated gyroscope measurements as x,y,z components.

SYS_INFO

Message ID: **0x81**

Description: Provides information on the device type and installed firmware version.

Message size: 38 bytes

Table 66: Structure of complete SYS_INFO message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x81	Message ID.
3	16-bit unsigned	0x020	Number of bytes in payload.
5	null-terminated string	(varies)	Device type identifier.
13	null-terminated string	(varies)	Firmware release version in MAJOR.MINOR.PATCH format, optionally extended with other symbols.
37	8-bit unsigned	(varies)	Checksum byte.

QUAT_DATA

Message ID: **0x82**

Description: Provides actual orientation data in quaternion representation.

Message size: 22 bytes

Table 67: Structure of complete QUAT_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x82	Message ID.
3	16-bit unsigned	0x010	Number of bytes in payload.
5	IEEE754 float	(varies)	q0 component
9	IEEE754 float	(varies)	q1 component
13	IEEE754 float	(varies)	q2 component
17	IEEE754 float	(varies)	q3 component
21	8-bit unsigned	(varies)	Checksum byte.

EULER_DATA

Message ID: **0x83**

Description: Provides actual orientation data in Euler angles representation.

Message size: 18 bytes

Table 68: Structure of complete EULER_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x83	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	Pitch angle in degrees.
9	IEEE754 float	(varies)	Roll angle in degrees.
13	IEEE754 float	(varies)	Yaw angle in degrees.
17	8-bit unsigned	(varies)	Checksum byte.

ROT_RATE_DATA

Message ID: **0x84**

Description: Provides actual rotation rate data.

Message size: 18 bytes

Table 69: Structure of complete ROT_RATE_DATA message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x84	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	Rotation rate over X axis in degrees/second.
9	IEEE754 float	(varies)	Rotation rate over Y axis in degrees/second.
13	IEEE754 float	(varies)	Rotation rate over Z axis in degrees/second.
17	8-bit unsigned	(varies)	Checksum byte.

ACC_CALIB_MAT

Message ID: **0x88**

Description: Reports calibration matrix for the accelerometer sensor.

Message size: 78 bytes

Table 70: Structure of complete ACC_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x88	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component

Byte Offset	Data Type	Value (hex)	Description
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

MAG_CALIB_MAT

Message ID: **0x89**

Description: Reports calibration matrix for the magnetometer sensor.

Message size: 78 bytes

Table 71: Structure of complete MAG_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x89	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

GYR_CALIB_MAT

Message ID: **0x8A**

Description: Reports calibration matrix for the gyroscope sensor.

Message size: 78 bytes

Table 72: Structure of complete GYR_CALIB_MAT message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8A	Message ID.
3	16-bit unsigned	0x048	Number of bytes in payload.
5	IEEE754 float	(varies)	c11 matrix component
9	IEEE754 float	(varies)	c12 matrix component
13	IEEE754 float	(varies)	c13 matrix component
17	IEEE754 float	(varies)	c21 matrix component
21	IEEE754 float	(varies)	c22 matrix component
25	IEEE754 float	(varies)	c23 matrix component
29	IEEE754 float	(varies)	c31 matrix component
33	IEEE754 float	(varies)	c32 matrix component
37	IEEE754 float	(varies)	c33 matrix component
41	IEEE754 float	(varies)	t11 matrix component
45	IEEE754 float	(varies)	t12 matrix component
49	IEEE754 float	(varies)	t13 matrix component
53	IEEE754 float	(varies)	t21 matrix component
57	IEEE754 float	(varies)	t22 matrix component
61	IEEE754 float	(varies)	t23 matrix component
65	IEEE754 float	(varies)	t31 matrix component
69	IEEE754 float	(varies)	t32 matrix component
73	IEEE754 float	(varies)	t33 matrix component
77	8-bit unsigned	(varies)	Checksum byte.

FILTER_MAG

Message ID: **0x8B**

Description: Reports values of measurement filter magnetometer matrix.

Message size: 18 bytes

Table 73: Structure of complete FILTER_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8B	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

FILTER_ACC

Message ID: **0x8C**

Description: Reports values of measurement filter accelerometer matrix.

Message size: 18 bytes

Table 74: Structure of complete FILTER_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8C	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

FILTER_GYR

Message ID: **0x8D**

Description: Reports values of measurement filter gyroscope matrix.

Message size: 18 bytes

Table 75: Structure of complete FILTER_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8D	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	h11 matrix component
9	IEEE754 float	(varies)	h22 matrix component
13	IEEE754 float	(varies)	h33 matrix component
17	8-bit unsigned	(varies)	Checksum byte.

FILTER_PROCN

Message ID: **0x8E**

Description: Reports values of filter process noise matrix.

Message size: 34 bytes

Table 76: Structure of complete FILTER_PROCN message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8E	Message ID.
3	16-bit unsigned	0x01C	Number of bytes in payload.
5	IEEE754 float	(varies)	q11 matrix component
9	IEEE754 float	(varies)	q22 matrix component
13	IEEE754 float	(varies)	q33 matrix component
17	IEEE754 float	(varies)	q44 matrix component
21	IEEE754 float	(varies)	q55 matrix component
25	IEEE754 float	(varies)	q66 matrix component
29	IEEE754 float	(varies)	q77 matrix component
33	8-bit unsigned	(varies)	Checksum byte.

TEMP

Message ID: **0x8F**

Description: Contains current temperature reading.

Message size: 10 bytes

Table 77: Structure of complete TEMP message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x8F	Message ID.
3	16-bit unsigned	0x04	Number of bytes in payload.
5	IEEE754 float	(varies)	Actual temperature value in Celsius degrees.
9	8-bit unsigned	(varies)	Checksum byte.

BAUD_RATE

Message ID: **0x90**

Description: Reports currently set baud rate.

Message size: 7 bytes

Table 78: Structure of complete BAUD_RATE message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x90	Message ID.
3	16-bit unsigned	0x01	Number of bytes in payload.
5	8-bit unsigned	(varies)	Current baud rate. One of the following values: 0x01 - 2400 bauds 0x02 - 4800 bauds 0x03 - 9600 bauds 0x04 - 19200 bauds 0x05 - 38400 bauds 0x06 - 57600 bauds 0x07 - 115200 bauds 0x08 - 230400 bauds 0x09 - 576000 bauds 0x0A - 921600 bauds
6	8-bit unsigned	(varies)	Checksum byte.

I2C_ADDR

Message ID: **0x91**

Description: Reports actual I2C slave address.

Message size: 7 bytes

Table 79: Structure of complete I2C_ADDR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x91	Message ID.
3	16-bit unsigned	0x01	Number of bytes in payload.
5	8-bit unsigned	(varies)	7-bit I2C slave device address.
6	8-bit unsigned	(varies)	Checksum byte.

CONFIRM

Message ID: **0x92**

Description: Confirms successful settings change operation.

Message size: 6 bytes

Table 80: Structure of complete CONFIRM message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0x92	Message ID.
3	16-bit unsigned	0x00	Number of bytes in payload.
5	8-bit unsigned	0x06A	Checksum byte.

RAW_ACC

Message ID: **0xA0**

Description: Provides raw latest accelerometer measurements as x,y,z components.

Message size: 12 bytes

Table 81: Structure of complete RAW_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA0	Message ID.
3	16-bit unsigned	0x06	Number of bytes in payload.
5	16-bit signed, two's complement	(varies)	x component of accelerometer measurement
7	16-bit signed, two's complement	(varies)	y component of accelerometer measurement
9	16-bit signed, two's complement	(varies)	z component of accelerometer measurement
11	8-bit unsigned	(varies)	Checksum byte.

RAW_MAG

Message ID: **0xA1**

Description: Provides raw latest magnetometer measurements as x,y,z components.

Message size: 12 bytes

Table 82: Structure of complete RAW_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA1	Message ID.
3	16-bit unsigned	0x06	Number of bytes in payload.
5	16-bit signed, two's complement	(varies)	x component of magnetometer measurement
7	16-bit signed, two's complement	(varies)	y component of magnetometer measurement
9	16-bit signed, two's complement	(varies)	z component of magnetometer measurement
11	8-bit unsigned	(varies)	Checksum byte.

RAW_GYR

Message ID: **0xA2**

Description: Provides raw latest gyroscope measurements as x,y,z components.

Message size: 12 bytes

Table 83: Structure of complete RAW_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA2	Message ID.
3	16-bit unsigned	0x06	Number of bytes in payload.
5	16-bit signed, two's complement	(varies)	x component of gyroscope measurement
7	16-bit signed, two's complement	(varies)	y component of gyroscope measurement
9	16-bit signed, two's complement	(varies)	z component of gyroscope measurement
11	8-bit unsigned	(varies)	Checksum byte.

NORM_ACC

Message ID: **0xA3**

Description: Provides latest normalized accelerometer measurements as x,y,z components.

Message size: 18 bytes

Table 84: Structure of complete NORM_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA3	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.

Byte Offset	Data Type	Value (hex)	Description
5	IEEE754 float	(varies)	x component of acclerometer measurement
9	IEEE754 float	(varies)	y component of acclerometer measurement
13	IEEE754 float	(varies)	z component of acclerometer measurement
17	8-bit unsigned	(varies)	Checksum byte.

NORM_MAG

Message ID: **0xA4**

Description: Provides latest normalized magnetometer measurements as x,y,z components.

Message size: 18 bytes

Table 85: Structure of complete NORM_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA4	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	x component of magnetometer measurement
9	IEEE754 float	(varies)	y component of magnetometer measurement
13	IEEE754 float	(varies)	z component of magnetometer measurement
17	8-bit unsigned	(varies)	Checksum byte.

NORM_GYR

Message ID: **0xA5**

Description: Provides latest normalized gyroscope measurements as x,y,z components.

Message size: 18 bytes

Table 86: Structure of complete NORM_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA5	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	x component of gyroscope measurement
9	IEEE754 float	(varies)	y component of gyroscope measurement
13	IEEE754 float	(varies)	z component of gyroscope measurement
17	8-bit unsigned	(varies)	Checksum byte.

CALIB_ACC

Message ID: **0xA6**

Description: Provides latest calibrated accelerometer measurements as x,y,z components.

Message size: 18 bytes

Table 87: Structure of complete CALIB_ACC message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.

Byte Offset	Data Type	Value (hex)	Description
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA6	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	x component of acclerometer measurement
9	IEEE754 float	(varies)	y component of acclerometer measurement
13	IEEE754 float	(varies)	z component of acclerometer measurement
17	8-bit unsigned	(varies)	Checksum byte.

CALIB_MAG

Message ID: **0xA7**

Description: Provides latest calibrated magnetometer measurements as x,y,z components.

Message size: 18 bytes

Table 88: Structure of complete CALIB_MAG message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA7	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	x component of magnetometer measurement
9	IEEE754 float	(varies)	y component of magnetometer measurement
13	IEEE754 float	(varies)	z component of magnetometer measurement
17	8-bit unsigned	(varies)	Checksum byte.

CALIB_GYR

Message ID: **0xA8**

Description: Provides latest calibrated gyroscope measurements as x,y,z components.

Message size: 18 bytes

Table 89: Structure of complete CALIB_GYR message on UART interface.

Byte Offset	Data Type	Value (hex)	Description
0	8-bit unsigned	0x05	First header byte.
1	8-bit unsigned	0xD3	Second header byte.
2	8-bit unsigned	0xA8	Message ID.
3	16-bit unsigned	0x0C	Number of bytes in payload.
5	IEEE754 float	(varies)	x component of gyroscope measurement
9	IEEE754 float	(varies)	y component of gyroscope measurement
13	IEEE754 float	(varies)	z component of gyroscope measurement
17	8-bit unsigned	(varies)	Checksum byte.

USB Interface

The X-200 AHRS device exposes the following USB interface:

- One default USB configuration.
- One USB interface.
- Two bulk endpoints:
 - one OUT endpoint under address 0x01 and 64 bytes packet size
 - one IN endpoint under address 0x81 and 64 bytes packet size

The device is identifying itself with:

```
idVendor      = 0xffff
idProduct    = 0x0002
```

and the following descriptor strings:

```
iManufacturer = Softexor
iProduct      = CyberAtom X-200
```

USB Message Format

Communication over USB port is handled in the form of bulk transfers of individual messages to- and from the device.

Messages sent by a host device (microcontroller, PC etc.) to the AHRS device are referred as **request** messages, while messages that host device retrieves from AHRS device are referred as **response** messages.

Requests and responses are realized as bulk transfer transactions, each having 1024 bytes of total length. A message occupies first N bytes of this transfer data, depending on a message type. The remaining part of the 1k data chunk is undefined and is ignored by the AHRS device and should be ignored by host.

Both message types have the same structure of:

- a single **message ID** byte
- optionally followed by a number of payload data bytes

Request Messages

Table 90: List of USB request messages.

Message	Message ID (hex)	Description
GET_SYS_INFO	0x01	Requests sending back information on the device type and firmware version.
GET_QUAT_DATA	0x02	Requests sending back actual orientation in quaternion representation.
GET_EULER_DATA	0x03	Requests sending back actual orientation in Euler angles representation.
GET_ROT_RATE_DATA	0x04	Requests sending back actual rotation rates.
REBOOT	0x05	Reboots the device.
FACTORY_RESET	0x07	Brings factory defaults settings to device RAM.
SET_ACC_CALIB_MAT	0x08	Sends calibration matrices for the accelerometer sensor.
SET_MAG_CALIB_MAT	0x09	Sends calibration matrices for the magnetometer sensor.

Message	Message ID (hex)	Description
<i>SET_GYR_CALIB_MAT</i>	0x0A	Sends calibration matrices for the gyroscope sensor.
<i>SET_FILTER_MAG</i>	0x0B	Sends new values for measurement filter magnetometer matrix.
<i>SET_FILTER_ACC</i>	0x0C	Sends new values for measurement filter accelerometer matrix.
<i>SET_FILTER_GYR</i>	0x0D	Sends new values for measurement filter gyroscope matrix.
<i>SET_FILTER_PROCN</i>	0x0E	Sends new values for filter process noise matrix.
<i>GET_TEMP</i>	0x0F	Requests sending current temperature reading
<i>SET_BAUD_RATE</i>	0x10	Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
<i>SET_I2C_ADDR</i>	0x11	Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.
<i>RESET_GYR</i>	0x15	Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).
<i>WRITE_FLASH</i>	0x16	Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.
<i>GET_ACC_CALIB_MAT</i>	0x17	Requests sending back actual accelerometer calibration settings.
<i>GET_MAG_CALIB_MAT</i>	0x18	Requests sending back actual magnetometer calibration settings.
<i>GET_GYR_CALIB_MAT</i>	0x19	Requests sending back actual gyroscope calibration settings.
<i>GET_RAW_ACC</i>	0x20	Requests sending raw accelerometer measurement data.
<i>GET_RAW_MAG</i>	0x21	Requests sending raw magnetometer measurement data.
<i>GET_RAW_GYR</i>	0x22	Requests sending raw gyroscope measurement data.
<i>GET_NORM_ACC</i>	0x23	Requests sending normalized accelerometer measurement data.
<i>GET_NORM_MAG</i>	0x24	Requests sending normalized magnetometer measurement data.
<i>GET_NORM_GYR</i>	0x25	Requests sending normalized gyroscope measurement data.
<i>GET_CALIB_ACC</i>	0x26	Requests sending calibrated accelerometer measurement data.
<i>GET_CALIB_MAG</i>	0x27	Requests sending calibrated magnetometer measurement data.
<i>GET_CALIB_GYR</i>	0x28	Requests sending calibrated gyroscope measurement data.
<i>REBOOT_BOOTLOADER</i>	0x29	Requests rebooting to the bootloader mode
<i>GET_FILTER_MAG</i>	0x2B	Requests values for measurement filter magnetometer matrix.
<i>GET_FILTER_ACC</i>	0x2C	Requests values for measurement filter accelerometer matrix.
<i>GET_FILTER_GYR</i>	0x2D	Requests values for measurement filter gyroscope matrix.
<i>GET_FILTER_PROCN</i>	0x2E	Requests values for filter process noise matrix.
<i>GET_I2C_ADDR</i>	0x30	Requests currently set I2C address.
<i>GET_BAUD_RATE</i>	0x31	Requests currently set baud rate.
<i>GET_STATS</i>	0x40	Requests operational statistics.

GET_SYS_INFO

Message ID: **0x01**

Description: Requests sending back information on the device type and firmware version.

Data Bytes: 0

When device receives this message, it will reply with sending back [SYS_INFO](#) message.

GET_QUAT_DATA

Message ID: **0x02**

Description: Requests sending back actual orientation in quaternion representation.

Data Bytes: 0

When device receives this message, it will reply with sending back [QUAT_DATA](#) message.

GET_EULER_DATA

Message ID: **0x03**

Description: Requests sending back actual orientation in Euler angles representation.

Data Bytes: 0

When device receives this message, it will reply with sending back [EULER_DATA](#) message.

GET_ROT_RATE_DATA

Message ID: **0x04**

Description: Requests sending back actual rotation rates.

Data Bytes: 0

When device receives this message, it will reply with sending back [ROT_RATE_DATA](#) message.

REBOOT

Message ID: **0x05**

Description: Reboots the device.

Data Bytes: 0

FACTORY_RESET

Message ID: **0x07**

Description: Brings factory defaults settings to device RAM.

Data Bytes: 0

SET_ACC_CALIB_MAT

Message ID: **0x08**

Description: Sends calibration matrices for the accelerometer sensor.

Data Bytes: 72

When device receives this message, it will reply with sending back [CONFIRM](#) message.

Table 91: Structure of data bytes for SET_ACC_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component

Byte Offset	Data Type	Description
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

SET_MAG_CALIB_MAT

Message ID: **0x09**

Description: Sends calibration matrices for the magnetometer sensor.

Data Bytes: 72

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 92: Structure of data bytes for SET_MAG_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

SET_GYR_CALIB_MAT

Message ID: **0x0A**

Description: Sends calibration matrices for the gyroscope sensor.

Data Bytes: 72

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 93: Structure of data bytes for SET_GYR_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

SET_FILTER_MAG

Message ID: **0x0B**

Description: Sends new values for measurement filter magnetometer matrix.

Data Bytes: 12

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 94: Structure of data bytes for SET_FILTER_MAG message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

SET_FILTER_ACC

Message ID: **0x0C**

Description: Sends new values for measurement filter accelerometer matrix.

Data Bytes: 12

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 95: Structure of data bytes for SET_FILTER_ACC message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

SET_FILTER_GYR

Message ID: **0x0D**

Description: Sends new values for measurement filter gyroscope matrix.

Data Bytes: 12

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 96: Structure of data bytes for SET_FILTER_GYR message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

SET_FILTER_PROCN

Message ID: **0x0E**

Description: Sends new values for filter process noise matrix.

Data Bytes: 28

When device receives this message, it will reply with sending back *CONFIRM* message.

Table 97: Structure of data bytes for SET_FILTER_PROCN message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	q11 matrix component
4	IEEE754 float	q22 matrix component
8	IEEE754 float	q33 matrix component
12	IEEE754 float	q44 matrix component
16	IEEE754 float	q55 matrix component
20	IEEE754 float	q66 matrix component
24	IEEE754 float	q77 matrix component

GET_TEMP

Message ID: **0x0F**

Description: Requests sending current temperature reading

Data Bytes: 0

When device receives this message, it will reply with sending back *TEMP* message.

SET_BAUD_RATE

Message ID: **0x10**

Description: Requests changing UART interface baud rate. Newly applied baud rate will be used after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Data Bytes: 1

When device receives this message, it will reply with sending back *BAUD_RATE* message.

Table 98: Structure of data bytes for SET_BAUD_RATE message on USB interface.

Byte Offset	Data Type	Description
0	8-bit unsigned	New baud rate.

SET_I2C_ADDR

Message ID: **0x11**

Description: Requests changing I2C slave address. Newly applied I2C slave address will be after next power-on or reset of the device. To make sure they persist, WRITE_FLASH command should be used afterwards.

Data Bytes: 1

When device receives this message, it will reply with sending back [I2C_ADDR](#) message.

Table 99: Structure of data bytes for SET_I2C_ADDR message on USB interface.

Byte Offset	Data Type	Description
0	8-bit unsigned	New 7-bit I2C slave address for the device.

RESET_GYR

Message ID: **0x15**

Description: Resets actual gyroscope settings (only in RAM) treating current rotation as bias only (not rotating device).

Data Bytes: 0

WRITE_FLASH

Message ID: **0x16**

Description: Saves current settings to device Flash for persistence. Once written, these settings will be available after next power-on.

Data Bytes: 0

GET_ACC_CALIB_MAT

Message ID: **0x17**

Description: Requests sending back actual accelerometer calibration settings.

Data Bytes: 0

When device receives this message, it will reply with sending back [ACC_CALIB_MAT](#) message.

GET_MAG_CALIB_MAT

Message ID: **0x18**

Description: Requests sending back actual magnetometer calibration settings.

Data Bytes: 0

When device receives this message, it will reply with sending back [MAG_CALIB_MAT](#) message.

GET_GYR_CALIB_MAT

Message ID: **0x19**

Description: Requests sending back actual gyroscope calibration settings.

Data Bytes: 0

When device receives this message, it will reply with sending back [GYR_CALIB_MAT](#) message.

GET_RAW_ACC

Message ID: **0x20**

Description: Requests sending raw accelerometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [RAW_ACC](#) message.

GET_RAW_MAG

Message ID: **0x21**

Description: Requests sending raw magnetometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [RAW_MAG](#) message.

GET_RAW_GYR

Message ID: **0x22**

Description: Requests sending raw gyroscope measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [RAW_GYR](#) message.

GET_NORM_ACC

Message ID: **0x23**

Description: Requests sending normalized accelerometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [NORM_ACC](#) message.

GET_NORM_MAG

Message ID: **0x24**

Description: Requests sending normalized magnetometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [NORM_MAG](#) message.

GET_NORM_GYR

Message ID: **0x25**

Description: Requests sending normalized gyroscope measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [NORM_GYR](#) message.

GET_CALIB_ACC

Message ID: **0x26**

Description: Requests sending calibrated accelerometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [CALIB_ACC](#) message.

GET_CALIB_MAG

Message ID: **0x27**

Description: Requests sending calibrated magnetometer measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [CALIB_MAG](#) message.

GET_CALIB_GYR

Message ID: **0x28**

Description: Requests sending calibrated gyroscope measurement data.

Data Bytes: 0

When device receives this message, it will reply with sending back [CALIB_GYR](#) message.

REBOOT_BOOTLOADER

Message ID: **0x29**

Description: Requests rebooting to the bootloader mode

Data Bytes: 0

GET_FILTER_MAG

Message ID: **0x2B**

Description: Requests values for measurement filter magnetometer matrix.

Data Bytes: 0

When device receives this message, it will reply with sending back [FILTER_MAG](#) message.

GET_FILTER_ACC

Message ID: **0x2C**

Description: Requests values for measurement filter accelerometer matrix.

Data Bytes: 0

When device receives this message, it will reply with sending back [FILTER_ACC](#) message.

GET_FILTER_GYR

Message ID: **0x2D**

Description: Requests values for measurement filter gyroscope matrix.

Data Bytes: 0

When device receives this message, it will reply with sending back [FILTER_GYR](#) message.

GET_FILTER_PROCN

Message ID: **0x2E**

Description: Requests values for filter process noise matrix.

Data Bytes: 0

When device receives this message, it will reply with sending back [FILTER_PROCN](#) message.

GET_I2C_ADDR

Message ID: **0x30**

Description: Requests currently set I2C address.

Data Bytes: 0

When device receives this message, it will reply with sending back [I2C_ADDR](#) message.

GET_BAUD_RATE

Message ID: **0x31**

Description: Requests currently set baud rate.

Data Bytes: 0

When device receives this message, it will reply with sending back *BAUD_RATE* message.

GET_STATS

Message ID: **0x40**

Description: Requests operational statistics.

Data Bytes: 32

When device receives this message, it will reply with sending back *STATS* message.

Table 100: Structure of data bytes for GET_STATS message on USB interface.

Byte Offset	Data Type	Description
0	32-bit signed, unsigned	Profiling table index 0
4	32-bit signed, unsigned	Profiling table index 1
8	32-bit signed, unsigned	Profiling table index 2
12	32-bit signed, unsigned	Profiling table index 3
16	32-bit signed, unsigned	Profiling table index 4
20	32-bit signed, unsigned	Profiling table index 5
24	32-bit signed, unsigned	Profiling table index 6
28	32-bit signed, unsigned	Profiling table index 8

Response Messages

Table 101: List of USB response messages.

Message	Message ID (hex)	Description
<i>SYS_INFO</i>	0x81	Provides information on the device type and installed firmware version.
<i>QUAT_DATA</i>	0x82	Provides actual orientation data in quaternion representation.
<i>EULER_DATA</i>	0x83	Provides actual orientation data in Euler angles representation.
<i>ROT_RATE_DATA</i>	0x84	Provides actual rotation rate data.
<i>ACC_CALIB_MAT</i>	0x88	Reports calibration matrix for the accelerometer sensor.
<i>MAG_CALIB_MAT</i>	0x89	Reports calibration matrix for the magnetometer sensor.
<i>GYR_CALIB_MAT</i>	0x8A	Reports calibration matrix for the gyroscope sensor.
<i>FILTER_MAG</i>	0x8B	Reports values of measurement filter magnetometer matrix.
<i>FILTER_ACC</i>	0x8C	Reports values of measurement filter accelerometer matrix.
<i>FILTER_GYR</i>	0x8D	Reports values of measurement filter gyroscope matrix.
<i>FILTER_PROCN</i>	0x8E	Reports values of filter process noise matrix.
<i>TEMP</i>	0x8F	Contains current temperature reading.
<i>BAUD_RATE</i>	0x90	Reports currently set baud rate.
<i>I2C_ADDR</i>	0x91	Reports actual I2C slave address.
<i>CONFIRM</i>	0x92	Confirms successful settings change operation.

Message	Message ID (hex)	Description
<i>RAW_ACC</i>	0xA0	Provides raw latest accelerometer measurements as x,y,z components.
<i>RAW_MAG</i>	0xA1	Provides raw latest magnetometer measurements as x,y,z components.
<i>RAW_GYR</i>	0xA2	Provides raw latest gyroscope measurements as x,y,z components.
<i>NORM_ACC</i>	0xA3	Provides latest normalized accelerometer measurements as x,y,z components.
<i>NORM_MAG</i>	0xA4	Provides latest normalized magnetometer measurements as x,y,z components.
<i>NORM_GYR</i>	0xA5	Provides latest normalized gyroscope measurements as x,y,z components.
<i>CALIB_ACC</i>	0xA6	Provides latest calibrated accelerometer measurements as x,y,z components.
<i>CALIB_MAG</i>	0xA7	Provides latest calibrated magnetometer measurements as x,y,z components.
<i>CALIB_GYR</i>	0xA8	Provides latest calibrated gyroscope measurements as x,y,z components.
<i>STATS</i>	0xC0	Provides operational statistics.

SYS_INFO

Message ID: **0x81**

Description: Provides information on the device type and installed firmware version.

Data Bytes: 32

Table 102: Structure of data bytes for SYS_INFO message on USB interface.

Byte Offset	Data Type	Description
0	null-terminated string	Device type identifier.
8	null-terminated string	Firmware release version in MAJOR.MINOR.PATCH format, optionally extended with other symbols.

QUAT_DATA

Message ID: **0x82**

Description: Provides actual orientation data in quaternion representation.

Data Bytes: 16

Table 103: Structure of data bytes for QUAT_DATA message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	q0 component
4	IEEE754 float	q1 component
8	IEEE754 float	q2 component
12	IEEE754 float	q3 component

EULER_DATA

Message ID: **0x83**

Description: Provides actual orientation data in Euler angles representation.

Data Bytes: 12

Table 104: Structure of data bytes for EULER_DATA message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	Pitch angle in degrees.
4	IEEE754 float	Roll angle in degrees.
8	IEEE754 float	Yaw angle in degrees.

ROT_RATE_DATA

Message ID: **0x84**

Description: Provides actual rotation rate data.

Data Bytes: 12

Table 105: Structure of data bytes for ROT_RATE_DATA message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	Rotation rate over X axis in degrees/second.
4	IEEE754 float	Rotation rate over Y axis in degrees/second.
8	IEEE754 float	Rotation rate over Z axis in degrees/second.

ACC_CALIB_MAT

Message ID: **0x88**

Description: Reports calibration matrix for the accelerometer sensor.

Data Bytes: 72

Table 106: Structure of data bytes for ACC_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

MAG_CALIB_MAT

Message ID: **0x89**

Description: Reports calibration matrix for the magnetometer sensor.

Data Bytes: 72

Table 107: Structure of data bytes for MAG_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

GYR_CALIB_MAT

Message ID: **0x8A**

Description: Reports calibration matrix for the gyroscope sensor.

Data Bytes: 72

Table 108: Structure of data bytes for GYR_CALIB_MAT message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	c11 matrix component
4	IEEE754 float	c12 matrix component
8	IEEE754 float	c13 matrix component
12	IEEE754 float	c21 matrix component
16	IEEE754 float	c22 matrix component
20	IEEE754 float	c23 matrix component
24	IEEE754 float	c31 matrix component
28	IEEE754 float	c32 matrix component
32	IEEE754 float	c33 matrix component
36	IEEE754 float	t11 matrix component
40	IEEE754 float	t12 matrix component
44	IEEE754 float	t13 matrix component
48	IEEE754 float	t21 matrix component
52	IEEE754 float	t22 matrix component
56	IEEE754 float	t23 matrix component
60	IEEE754 float	t31 matrix component
64	IEEE754 float	t32 matrix component
68	IEEE754 float	t33 matrix component

FILTER_MAG

Message ID: **0x8B**

Description: Reports values of measurement filter magnetometer matrix.

Data Bytes: 12

Table 109: Structure of data bytes for FILTER_MAG message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

FILTER_ACC

Message ID: **0x8C**

Description: Reports values of measurement filter accelerometer matrix.

Data Bytes: 12

Table 110: Structure of data bytes for FILTER_ACC message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

FILTER_GYR

Message ID: **0x8D**

Description: Reports values of measurement filter gyroscope matrix.

Data Bytes: 12

Table 111: Structure of data bytes for FILTER_GYR message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	h11 matrix component
4	IEEE754 float	h22 matrix component
8	IEEE754 float	h33 matrix component

FILTER_PROCN

Message ID: **0x8E**

Description: Reports values of filter process noise matrix.

Data Bytes: 28

Table 112: Structure of data bytes for FILTER_PROCN message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	q11 matrix component
4	IEEE754 float	q22 matrix component
8	IEEE754 float	q33 matrix component
12	IEEE754 float	q44 matrix component
16	IEEE754 float	q55 matrix component
20	IEEE754 float	q66 matrix component
24	IEEE754 float	q77 matrix component

TEMP

Message ID: **0x8F**

Description: Contains current temperature reading.

Data Bytes: 4

Table 113: Structure of data bytes for TEMP message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	Actual temperature value in Celsius degrees.

BAUD_RATE

Message ID: **0x90**

Description: Reports currently set baud rate.

Data Bytes: 1

Table 114: Structure of data bytes for BAUD_RATE message on USB interface.

Byte Offset	Data Type	Description
0	8-bit unsigned	Current baud rate.

I2C_ADDR

Message ID: **0x91**

Description: Reports actual I2C slave address.

Data Bytes: 1

Table 115: Structure of data bytes for I2C_ADDR message on USB interface.

Byte Offset	Data Type	Description
0	8-bit unsigned	7-bit I2C slave device address.

CONFIRM

Message ID: **0x92**

Description: Confirms successful settings change operation.

Data Bytes: 0

RAW_ACC

Message ID: **0xA0**

Description: Provides raw latest accelerometer measurements as x,y,z components.

Data Bytes: 6

Table 116: Structure of data bytes for RAW_ACC message on USB interface.

Byte Offset	Data Type	Description
0	16-bit signed, two's complement	x component of accelerometer measurement
2	16-bit signed, two's complement	y component of accelerometer measurement

Byte Offset	Data Type	Description
4	16-bit signed, two's complement	z component of acclerometer measurement

RAW_MAG

Message ID: **0xA1**

Description: Provides raw latest magnetometer measurements as x,y,z components.

Data Bytes: 6

Table 117: Structure of data bytes for RAW_MAG message on USB interface.

Byte Offset	Data Type	Description
0	16-bit signed, two's complement	x component of magnetometer measurement
2	16-bit signed, two's complement	y component of magnetometer measurement
4	16-bit signed, two's complement	z component of magnetometer measurement

RAW_GYR

Message ID: **0xA2**

Description: Provides raw latest gyroscope measurements as x,y,z components.

Data Bytes: 6

Table 118: Structure of data bytes for RAW_GYR message on USB interface.

Byte Offset	Data Type	Description
0	16-bit signed, two's complement	x component of gyroscope measurement
2	16-bit signed, two's complement	y component of gyroscope measurement
4	16-bit signed, two's complement	z component of gyroscope measurement

NORM_ACC

Message ID: **0xA3**

Description: Provides latest normalized accelerometer measurements as x,y,z components.

Data Bytes: 12

Table 119: Structure of data bytes for NORM_ACC message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of acclerometer measurement
4	IEEE754 float	y component of acclerometer measurement
8	IEEE754 float	z component of acclerometer measurement

NORM_MAG

Message ID: **0xA4**

Description: Provides latest normalized magnetometer measurements as x,y,z components.

Data Bytes: 12

Table 120: Structure of data bytes for NORM_MAG message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of magnetometer measurement
4	IEEE754 float	y component of magnetometer measurement
8	IEEE754 float	z component of magnetometer measurement

NORM_GYR

Message ID: **0xA5**

Description: Provides latest normalized gyroscope measurements as x,y,z components.

Data Bytes: 12

Table 121: Structure of data bytes for NORM_GYR message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of gyroscope measurement
4	IEEE754 float	y component of gyroscope measurement
8	IEEE754 float	z component of gyroscope measurement

CALIB_ACC

Message ID: **0xA6**

Description: Provides latest calibrated accelerometer measurements as x,y,z components.

Data Bytes: 12

Table 122: Structure of data bytes for CALIB_ACC message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of acclerometer measurement
4	IEEE754 float	y component of acclerometer measurement
8	IEEE754 float	z component of acclerometer measurement

CALIB_MAG

Message ID: **0xA7**

Description: Provides latest calibrated magnetometer measurements as x,y,z components.

Data Bytes: 12

Table 123: Structure of data bytes for CALIB_MAG message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of magnetometer measurement
4	IEEE754 float	y component of magnetometer measurement
8	IEEE754 float	z component of magnetometer measurement

CALIB_GYR

Message ID: **0xA8**

Description: Provides latest calibrated gyroscope measurements as x,y,z components.

Data Bytes: 12

Table 124: Structure of data bytes for CALIB_GYR message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	x component of gyroscope measurement
4	IEEE754 float	y component of gyroscope measurement
8	IEEE754 float	z component of gyroscope measurement

STATS

Message ID: **0xC0**

Description: Provides operational statistics.

Data Bytes: 8

Table 125: Structure of data bytes for STATS message on USB interface.

Byte Offset	Data Type	Description
0	IEEE754 float	sensors update rate
4	IEEE754 float	orientation update rate

Troubleshooting

Factory Defaults

Restoring factory default settings can be done by either:

- Sending FACTORY_RESET message over UART interface
- Using Cyber Studio application to perform the factory reset operation

This operation loads initial (factory default) settings to a device RAM memory. In order to persist the settings for later, use WRITE_FLASH message afterwards.

Configuring UDEV system on Linux Systems for CyberAtom Devices

Depending on your Linux setup it might necessary to give proper permissions to be able to access CyberAtom as USB devices using regular user privileges. The following procedure has been tested on Ubuntu 14.04 LTS and should be straightforward to apply on Debian or similar distributions.

1. Create user group `cyberatom`.

```
$ sudo groupadd cyberatom
```

2. Add the user account to this group.

```
$ usermod -a -G cyberatom <username>
```

3. You need to re-login to have user account reflected in the user account settings. This can be checked by:

```
$ groups
```

4. Within `/etc/udev/rules.d/` create rule file named `50-CyberAtom.rules` with content:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="ffff", ATTRS{idProduct}=="0002",
GROUP="cyberatom", MODE="0666"
SUBSYSTEM=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="df11",
GROUP="cyberatom", MODE="0666"
```

5. Restart udev system with:

```
sudo restart udev
```

or

```
sudo udevadm control --reload
```

6. Re-connect your CyberAtom device to the USB port of your PC machine.

You should be able to see the details of your CyberAtom device without any restrictions. E.g. the following command:

```
$ lsusb -d ffff:0002 -v
```

should give similar output:

```

Bus 003 Device 106: ID ffff:0002
Device Descriptor:
  bLength                18
  bDescriptorType        1
  bcdUSB                  2.00
  bDeviceClass            255 Vendor Specific Class
  bDeviceSubClass         255 Vendor Specific Subclass
  bDeviceProtocol         255 Vendor Specific Protocol
  bMaxPacketSize0         64
  idVendor                 0xffff
  idProduct               0x0002
  bcdDevice               2.00
  iManufacturer           1 Softexor
  iProduct                2 CyberAtom X-200
  iSerial                 3 738353020175136503006300
  bNumConfigurations      1
Configuration Descriptor:
  bLength                9
  bDescriptorType        2
  wTotalLength           32
  bNumInterfaces          1
  bConfigurationValue    1
  iConfiguration         0
  bmAttributes            0xc0
    Self Powered
  MaxPower                100mA
Interface Descriptor:
  bLength                9
  bDescriptorType        4
  bInterfaceNumber       0
  bAlternateSetting      0
  bNumEndpoints          2
  bInterfaceClass         0 (Defined at Interface level)
  bInterfaceSubClass     0
  bInterfaceProtocol     0
  iInterface              0
Endpoint Descriptor:
  bLength                7
  bDescriptorType        5
  bEndpointAddress       0x01  EP 1 OUT
  bmAttributes            2
    Transfer Type        Bulk
    Synch Type           None
    Usage Type           Data
  wMaxPacketSize         0x0040  1x 64 bytes
  bInterval              0
Endpoint Descriptor:
  bLength                7
  bDescriptorType        5
  bEndpointAddress       0x81  EP 1 IN
  bmAttributes            2
    Transfer Type        Bulk
    Synch Type           None
    Usage Type           Data
  wMaxPacketSize         0x0040  1x 64 bytes
  bInterval              0
Device Status:          0x0001
  Self Powered

```

Installing USB Driver for CyberAtom Devices on Windows

You will need administrator's rights to perform the installation.

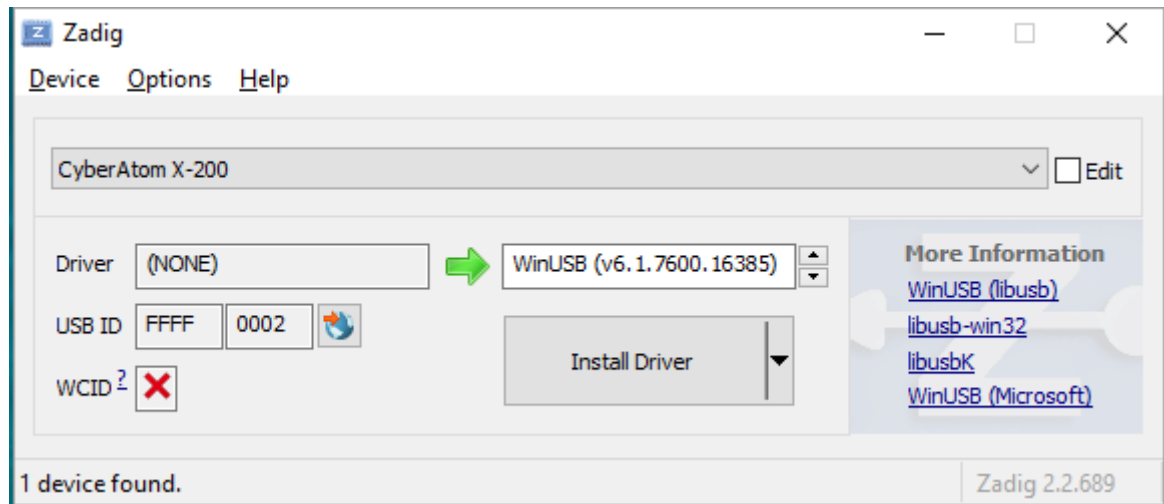
When a CyberAtom device is connected to a Windows PC for the very first time, Windows might report that it failed to find a USB driver automatically.

This document explains how-to install the driver manually.

The instruction is valid for Windows XP/Vista/7/8/10 versions.

1. Download **Zadig** application from zadig.akeo.ie appropriate for your Windows version.
2. Connect CyberAtom device to your PC using USB cable.
3. Run the Zadig application.

The application should detect CyberAtom device and the screen should look like that:



Make sure that the **Driver** field, has **WinUSB** option selected.

4. Click "Install Driver" button.

When installation process is finished, CyberAtom device can be found in Windows Device Manager under **Universal Serial Bus Devices** tree node.

Identify CyberAtom Device Using USB Connection

You need to have cyber-boot application (part of CyberAtom-utils package).

On Windows system you need to have USB driver for CyberAtom devices installed.

On Linux system you need to have proper configuration of udev system (if running this procedure with regular user rights).

Each CyberAtom device has unique serial number assigned to it. This number is permanent and is not changed during a firmware upgrade procedure.

This document describes how-to determine this unique device number.

1. Connect CyberAtom device to the PC with USB cable.
2. Run `cyber-boot` program with `--list` parameter.

On Linux systems that would be:

```
$ ./cyber-boot --list
```

On Windows systems that would be:

```
> cyber-boot.exe --list
```

The application will list all connected CyberAtom devices with the serial number value:

List of CyberAtom devices detected:

1. CyberAtom X-200 SN: 63235302107523D4D3009300

Firmware Upgrade Using USB Interface

On machine with Windows operating system you need to have installed USB Driver for CyberAtom devices.

On machine with Linux operating systems you may want configuring UDEV system for CyberAtom Devices to access it with regular user rights

The firmware upgrade operation is the following:

1. Connect CyberAtom to your PC using Micro-B USB cable.
2. Download latest firmware version and unzip it to get a *.dfu file.
3. Download utilities and unzip it to get cyber-boot executable available.
4. Use command line cyber-boot tool to upgrade the firmware.

On Windows systems that would be:

```
> cyber-boot.exe -b --upload /path/to/firmware.dfu
```

On Linux systems that would be:

```
$ ./cyber-boot -b --upload /path/to/firmware.dfu
```

The application will perform firmware upload giving the following progress indication:

```
Rebooting CyberAtom USB device.
Entering bootloader....
OK. Ready for upload.
Uploading... [=====] 64% 25456 bytes
```

Then - it will reboot the device.

You can use Cyber Studio to connect to the CyberAtom device to verify installed version of your firmware. It should match the one in the name of the firmware file.

Firmware Upgrade Using UART Interface

You need to have USB-to-UART adapter.

The firmware upgrade operation is the following:

1. Connect CyberAtom to your PC using USB-to-UART.
2. Download latest firmware version and unzip it to get a *.dfu file.
3. Download utilities and unzip it to get cyber-boot executable available.
4. Use command line cyber-boot tool to upgrade the firmware.

On Windows (assuming the adapter is reported as COM1) systems that would be:

```
> cyber-boot.exe -b --serial-port COM1 --upload /path/to/firmware.dfu
```

On Linux systems (assuming the adapter is reported as /dev/ttyUSB0 device) that would be:

```
$ ./cyber-boot -b --serial-port /dev/ttyUSB0 --upload /path/to/
firmware.dfu
```

The application will perform firmware upload giving the following progress indication:

```
Rebooting CyberAtom USB device.  
Entering bootloader...  
Waiting for memory layout info...  
Uploading...      [=====] 64%      25456 bytes
```

Then - it will reboot the device.

You can use Cyber Studio to connect to the CyberAtom device to verify installed version of your firmware. It should match the one in the name of the firmware file.

Recovering Device Firmware

If at any point the firmware is corrupted, it is possible to boot device to bootloader mode and repeat device firmware installation procedure.

1. Power OFF the device.
2. Join (e.g. solder) JP1 jumper on the device board.
3. Power ON the device.

The device will boot to the bootloader mode. At this point, cyber-boot tool can be used to flash the firmware. When done, JP1 jumper should be disconnected.

Legal Notice

Any information furnished by Softexor and its agents is believed to be accurate and reliable. All specifications are subject to change without notice. Responsibility for the use and application of Softexor materials rests with the end user, since Softexor and its agents cannot be aware of all potential uses. Softexor makes no warranties as to the fitness, merchantability or suitability of any Softexor materials or products for any specific or general uses. Softexor shall not be liable for incidental or consequential damages of any kind. All Softexor products are sold pursuant to the Softexors' Terms and Conditions of sale in effect from time to time, a copy of which will be furnished upon request.

Softexor, the Softexor Logo, and other marks are trade marks or registered trade marks of Softexor S.A or an affiliate company thereof. Other product or service names may be the property of third parties. Nothing herein provides a license under any Softexor or any third party intellectual property rights.

© Copyright 2015 Softexor. All Rights Reserved.