AIMB Protocol v2023

NMEA Based Microcontroller Communication Protocol Specification

17.04.2023

beCee Soft Art

© André Spitzner



Content

[1. Document history 3](#_Toc133293443)

[2. Definition 4](#_Toc133293444)

[3. Supported sentences 6](#_Toc133293445)

[3.1 ACCEL Accelerometer Sensor 6](#_Toc133293446)

[3.2 ACDC Voltage Sensor 7](#_Toc133293447)

[3.3 ACS – Current Sensor 8](#_Toc133293448)

[3.4 ALTITUDE - Altitude Over Barometric Pressure Sensor 9](#_Toc133293449)

[3.5 ANALOG - Analog Port Voltages 10](#_Toc133293450)

[3.6 ANALOGPLAIN - Analog Port Plain Values 11](#_Toc133293451)

[3.7 BUMPER – Bumper Sensor 12](#_Toc133293452)

[3.8 CO2 – Carbon Dioxide Sensor 13](#_Toc133293453)

[3.9 COMPASS – Magnetic Compass Sensor 14](#_Toc133293454)

[3.10 DISTANCE – Distance Sensor 15](#_Toc133293455)

[3.11 DUST – Dust Sensor 16](#_Toc133293456)

[3.12 ELTIME - Elapsed Time Measurement 17](#_Toc133293457)

[3.13 ENGINE – Engine Speed Command 18](#_Toc133293458)

[3.14 ENGINE2 – Dual Engine Speed Command 19](#_Toc133293459)

[3.15 ERROR – Device Error Message 20](#_Toc133293460)

[3.16 FRAME – Block Measurement 21](#_Toc133293461)

[3.17 FREERAM – Free Memory 22](#_Toc133293462)

[3.18 GAS – Universal Gas Sensor Measurement 23](#_Toc133293463)

[3.19 GPSFIX – GPS Positioning Fix 24](#_Toc133293464)

[3.20 GYRO – Gyroscope Sensor 25](#_Toc133293465)

[3.21 HALL – Hall Sensor 26](#_Toc133293466)

[3.22 HALT – Halt Actuator Command 27](#_Toc133293467)

[3.23 HUMIDITY – Relative Humidity Sensor 28](#_Toc133293468)

[3.24 LIFESIGN – Life Sign Message 29](#_Toc133293469)

[3.25 LIGHT – Light Sensor 30](#_Toc133293470)

[3.26 LRFC – Laser Range Finder Command 31](#_Toc133293471)

[3.27 LRFUDS – Laser Range Finder UDS Distances 32](#_Toc133293472)

[3.28 LRFUDSC – Laser Range Finder UDS Command 33](#_Toc133293473)

[3.29 MOVE – Move Engine Command 34](#_Toc133293474)

[3.30 MOVE2 – Move Engine Command With Direction 35](#_Toc133293475)

[3.31 O2 – Oxygen Sensor 36](#_Toc133293476)

[3.32 PIR – Infrared Motion Sensor 37](#_Toc133293477)

[3.33 POSITION – Geolocation Positioning 38](#_Toc133293478)

[3.34 PRESSURE – Barometric Pressure Sensor 39](#_Toc133293479)

[3.35 PROGSIZE – Program Size 40](#_Toc133293480)

[3.36 RVNFO – Robot Version Information 41](#_Toc133293481)

[3.37 SERVOC – Servo position Command 42](#_Toc133293482)

[3.38 SPEED – Speed Sensor 43](#_Toc133293483)

[3.39 STATE – HIGH/LOW State Sensor 44](#_Toc133293484)

[3.40 TAP – Tap Detection Sensor 45](#_Toc133293485)

[3.41 TEMP – Temperature Sensor 46](#_Toc133293486)

[3.42 TIMER – Timer Sensor 47](#_Toc133293487)

[3.43 TOP – PGTOP Vendor Id 48](#_Toc133293488)

[3.44 UDSC – UDS Command 49](#_Toc133293489)

[3.45 UNI – Universal Floating Point Value Sensor 50](#_Toc133293490)

[3.46 UNIINT – Universal Integer Value Sensor 51](#_Toc133293491)

[3.47 USEDRAM – Used RAM by Microcontroller 52](#_Toc133293492)

[3.48 VIBRATION – Vibration Sensor 53](#_Toc133293493)

[3.49 WARNING – Device Warning Information 54](#_Toc133293494)

# Document history

|  |  |  |
| --- | --- | --- |
| **Date of change** | **Author** | **Whats changed** |
| 2008-06-05 | André Spitzner | First version of AIMB protocol |
| 2016-12-27 | André Spitzner | Update of AIMB protocol, add new sentences |
| 2023-04-17 | André Spitzner | Document recreation and update to fit and document current implementation of AIMB objects |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Definition

The AIMB microcontroller protocol is based on National Marine Electronics Association (NMEA, US) protocol sentence definition.

This protocol is used for communication between bcplanet AIMB services and microcontrollers using AIMB.

As communication channels are direct connections to UART/Serial and network based WebSocket application channel. AIMB over HTTPS request are also supported by bcplanet.UNIVERSE.

It defers from NMEA definition in that point, that after device type and command name the first value is a device id, to specify the source address of sensor in network.

The protocol is plain text based.

Every sentence represents one sensor measurement or a command sentence.

**Supported directions of sentence:**

|  |  |  |
| --- | --- | --- |
| **Type** | **From** | **To** |
| Measurement | microcontroller | bcplanet AIMB |
| Command | bcplanet AIMB | microcontroller |

**Structure of a AIMB sentence:**

$[device id(2 char)][command],[sensor id],[data]\*[CRC][cr][lf]

* Every sentence starts with “$” character
* Device id is a 2 upper cased character string
* Command is an upper-cased character string
* Sensor id is a character string
* Data is a data block, where every value is separated by “,” comma char
* The CRC is an XOR HEX value from complete sentence without “\*” which separates the CRC from sentence
* Every sentence ends with carriage return and line feet characters(\r\n), this parameter is optional, bcplanet AIMB supports also “out of band” sensor sentences.

Example sentence:

$BCCO2,2024,416,ppm\*49

**Notice:**

**CRC is mandatory in bcplanet AIMB environment, sentences with wrong or missing CRC are dropped and logged as error.**

# Supported sentences

In the following chapters are the actual supported sensor measurements and command sentences for bcplanet AIMB are described in form and function.

## ACCEL Accelerometer Sensor

This measurement sentence was originally designed for ADXL345 accelerometer sensor. But should work also with sensors from other vendors when values in same range.

$--**ACCEL**,[1],[2],[3],[4],[5],[6],[7]\*[CRC]\r\n

1. Sensor Id
2. Raw X Value
3. Raw Y Value
4. Raw Z Value
5. Scaled X Value
6. Scaled Y Value
7. Scaled Z Value

## ACDC Voltage Sensor

$--**ACDC**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Voltage Value
3. Unit (V = Volt)

## ACS – Current Sensor

$--**ACS**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. ACS Value

## ALTITUDE - Altitude Over Barometric Pressure Sensor

$--**ALTITUDE**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Altitude Value
3. Unit (m = Meter)

## ANALOG - Analog Port Voltages

$--**ANALOG**,[1],[2],[3],[4],[5],[6],[7]\*[CRC]\r\n

1. Voltage on Analog A0
2. Voltage on Analog A1
3. Voltage on Analog A2
4. Voltage on Analog A3
5. Voltage on Analog A4
6. Voltage on Analog A5
7. Unit

## ANALOGPLAIN - Analog Port Plain Values

$--**ANALOGPLAIN**,[1],[2],[3],[4],[5],[6]\*[CRC]\r\n

1. Voltage on Analog A0
2. Voltage on Analog A1
3. Voltage on Analog A2
4. Voltage on Analog A3
5. Voltage on Analog A4
6. Voltage on Analog A5

## BUMPER – Bumper Sensor

$--**BUMPER**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Bumper Value

## CO2 – Carbon Dioxide Sensor

$--**CO2**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. CO2 ppm Value
3. Unit

## COMPASS – Magnetic Compass Sensor

$--**COMPASS**,[1],[2],[3],[4],[5],[6],[7],[8],[9]\*[CRC]\r\n

1. Sensor Id
2. Raw X Axis Value
3. Raw Y Axis Value
4. Raw Z Axis Value
5. Scaled X Axis Value
6. Scaled Y Axis Value
7. Scaled Y Axis Value
8. Heading Value
9. Radians Degrees Value

## DISTANCE – Distance Sensor

$--**DISTANCE**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Distance Value
3. Unit

## DUST – Dust Sensor

$--**DUST**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Dust Value
3. Unit

## ELTIME - Elapsed Time Measurement

$--**ELTIME**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Elapsed Time
3. Unit (micro / ms / s)

## ENGINE – Engine Speed Command

$--**ENGINE**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Engine Value

## ENGINE2 – Dual Engine Speed Command

$--**ENGINE2**,[1],[2],[3],[4],[5]\*[CRC]\r\n

1. Sensor Id
2. ENGINE 1 Value
3. ENGINE 1 Direction
4. ENGINE 2 Value
5. ENGINE 2 Direction

## ERROR – Device Error Message

$--**ERROR**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Error Id
3. Description

## FRAME – Block Measurement

This block measurement sentence is better to send over AIMB HTTPS or AIMB WebSocket network connection to reduce latency and optimize transmission speed. This kind of sentence could be very larger than other sentences, because the size is based on count of measurement values in the sentence.

$--**BLOCK**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Value 1 – n, values are comma separated
3. Values Count

## FREERAM – Free Memory

$--**FREERAM**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Free Ram in Bytes
3. Unit (Byte)

## GAS – Universal Gas Sensor Measurement

$--**GAS**,[1],[2],[3],[4],[5],[6],[7]\*[CRC]\r\n

1. Sensor Id
2. Analog Value
3. Resistance - calculated resistence depends on the sensor pulldown resistor
4. RO
5. Gas value in ppm
6. Unit
7. Gas name

## GPSFIX – GPS Positioning Fix

$--**GPSFIX**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Value
3. Unit

## GYRO – Gyroscope Sensor

$--**GYRO**,[1],[2],[3],[4],[5],[6],[7]\*[CRC]\r\n

1. Sensor Id
2. Raw X Value
3. Raw Y Value
4. Raw Z Value
5. Scaled X Value
6. Scaled Y Value
7. Scaled Z Value

## HALL – Hall Sensor

$--**HALL**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. HALL Value
3. Unit (A = Ampere)

## HALT – Halt Actuator Command

$**--HALT**,[1]\*[CRC]\r\n

1. Sensor Id

## HUMIDITY – Relative Humidity Sensor

$--**HUMIDITY**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Humidity Value, Relative in Percent
3. Unit

## LIFESIGN – Life Sign Message

$--**LIFESIGN**,[1]\*[CRC]\r\n

1. Current Date Time, Format "yyyy-MM-dd HH:mm:ss"

## LIGHT – Light Sensor

$--**LIGHT**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Light Value

## LRFC – Laser Range Finder Command

$--**LRFC**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Resolution in degrees
3. Command character

## LRFUDS – Laser Range Finder UDS Distances

$--**LRFUDS**,[1],[2],[3],[4],[5],[6],[7],[8]\*[CRC]\r\n

1. Sensor Id
2. Alpha
3. Beta
4. Lrf Distance
5. Unit
6. Uds Distance
7. Unit
8. Number

## LRFUDSC – Laser Range Finder UDS Command

$--**LRFUDSC**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Alpha angle
3. Beta angle

## MOVE – Move Engine Command

$--**MOVE**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Speed

## MOVE2 – Move Engine Command With Direction

$--**MOVE2**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Speed
3. Direction

## O2 – Oxygen Sensor

$--**O2**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. O2 ppm Value
3. Unit

## PIR – Infrared Motion Sensor

$--**PIR**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. State 0=Low / 1 = High

## POSITION – Geolocation Positioning

$--**GEOPOSITION**,[1],[2],[3],[4]\*[CRC]\r\n

1. Sensor Id
2. Latitude
3. Longitude
4. Altitude (m)

## PRESSURE – Barometric Pressure Sensor

$--**PRESSURE**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Pressure Value
3. Unit (B = Bar)

## PROGSIZE – Program Size

$--**PROGSIZE**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Used Ram in Bytes
3. Unit (Byte)

## RVNFO – Robot Version Information

$--**RVNFO**,[1],[2],[3]\*[CRC]\r\n

1. Identifier
2. Version
3. Description

## SERVOC – Servo position Command

$--**SERVOC**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Command character

## SPEED – Speed Sensor

$--**SPEED**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Speed Value
3. Unit

## STATE – HIGH/LOW State Sensor

$--**STATE**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. State 0=Disabled/1=Enabled

## TAP – Tap Detection Sensor

$--**TAP**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Tap State

## TEMP – Temperature Sensor

$--**TEMP**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Temperature Value
3. Unit

## TIMER – Timer Sensor

$--**TIMER**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Timer Value

## TOP – PGTOP Vendor Id

$--**TOP**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. ACS Value

## UDSC – UDS Command

$--**UDSC**,[1],[2]\*[CRC]\r\n

1. Sensor Id
2. Command character

## UNI – Universal Floating Point Value Sensor

$--**UNI**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Value float
3. Unit

## UNIINT – Universal Integer Value Sensor

$--**UNIINT**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Value INT
3. Unit

## USEDRAM – Used RAM by Microcontroller

$--**USEDRAM**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Used Ram in Bytes
3. Unit (Byte)

## VIBRATION – Vibration Sensor

$--**VIBRATION**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. Vibration Value
3. Unit (v = Volt)

## WARNING – Device Warning Information

$--**WARNING**,[1],[2],[3]\*[CRC]\r\n

1. Sensor Id
2. WARNING Id
3. Description