Basecam GPS_IMU 1.x serial protocol specification

Applicable for GPS_IMU boards with firmware 1.x

Revision history:

- rev. 0.1 - 01.03.2019: preliminary version

Overview

Communications is initiated from the remote side (host) by sending *outgoing* commands. The controller may do some action and send response (for the host it is an *incoming* command).

Remote side is responsible for preventing output and input buffers from overflow. For example, if requested too big amount of data that does not fit into the output buffer, the excessive data in response will be skipped. Input and output buffers are 512 bytes size.

Board can work on different serial baud rates, adjustable by the parameters, with the 115200 as default value.

The main set of coordinate systems:

Ground reference frames:
NED (North-East-Down)

- Right-handed, Cartesian, non-inertial
- Geodetic frame with origin located at the surface of Earth (WGS84 ellipsoid)
- Positive N-axis points towards North (tangent to WGS84 ellipsoid)
- Positive E-axis points towards East (tangent to WGS84 ellipsoid)
- Positive D-axis points down into the ground, completing the right-handed system

ECEF (Earth-Centered Earth-Fixed) (In developing)

- Right-handed, Cartesian, non-inertial
- Frame with origin located at the center of Earth
- Fixed to and rotates with Earth
- Positive X-axis aligns with the WGS84 X-axis, which aligns with the International Earth Rotation and Reference Systems Service (IERS) Prime Meridian.
- Positive Z-axis aligns with the WGS84 Z-axis, which aligns with the IERS Reference Pole (IRP) that points towards the North Pole.
- Positive Y-axis aligns with the WGS84 Y-axis, completing the right-handed system.

LLA (Latitude, Longitude, Altitude)

- Non-inertial
- Geodetic frame with origin located at the surface of Earth (WGS84 ellipsoid)
- Latitude is defined as the angle from the equatorial plane to a line normal to the surface of the WGS84 ellipsoid at the location of the vehicle
- Longitude is defined as the east-west angular displacement measured positive to the east from the IERS Reference Meridian to the location of the vehicle

Body reference frames:
XYZ (X, Y, and Z axes labeled on the hardware)

- Right-handed
- Positive right-hand rotation
- Roll angle rotation around the X-axis

- Pitch angle rotation around the Y-axis
- Yaw (heading) angle rotation around the Z-axis

AHRS (Attitude and heading reference system) format:

QUAT (quaternions (w, x, y, z))
- Body frame to NED frame
- The first term is the scalar value

DCM6 (rotation matrix, direction cosine matrix)
- Body frame to NED frame
- Contains only the first and third rows of the rotation matrix.
- The second row can be calculated as cross-product of the first and third rows of the rotation matrix.

DCM9 (rotation matrix, direction cosine matrix)
- Body frame to NED frame
- Regular form of rotation matrix

Euler angles (1-2-3) (roll, pitch, yaw (heading))

Euler angles (3-2-1) (yaw (heading), pitch, roll)

Message format

Each command consists of the *header* and the *body*, both with checksum. Commands with the wrong header or body checksum, or with the body size that differs from expected, should be ignored. Parser should scan incoming datastream for the next start character and try to restore synchronization from it.

Input and output commands have the same format, described below:

header				body			crc16	
start character $ (0x24)	command ID, 0..255	payload size N=0..255	header checksum	payload, variable size			message checksum	
0	1	2	3	4	...	4+N-1	4+N	4+N+1

Header checksum is calculated as (command ID + payload_size) modulo 256 (operation "modulo" means least significant byte of the sum).

Message checksum is calculated as a CRC16 over the header bytes and payload bytes, starting from index 1 to index 4+N-1. A reference implementation of CRC16 using polynomial 0x8005 is given in the appendix A.

Example messages

CMD_GET_USER_CONF_LOG:

header				crc16	
0	1	2	3	4	5
0x24	0x0C	0x00	0x0C	0x60	0x03

CMD_USER_CONF_LOG:
Active channels for STREAM1 (ACTIV_CH_MASK = 0x00000109): 0, 3, 8.
Interval between the data samples for STREAM1 (INTERVAL_MS = 0x0064): 100 ms.
Active channels for STREAM2 (ACTIV_CH_MASK = 0x00000000): all disabled.
Interval between the data samples for STREAM2 (INTERVAL_MS = 0x0064): 100 ms.

header				payload												crc16	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0x24	0x0D	0x0C	0x19	0x09	0x01	0x00	0x00	0x64	0x00	0x00	0x00	0x00	0x00	0x64	0x00	0xD5	0xE8

Data type notation

- 1u – 1 byte unsigned
- 1s – 1 byte signed
- 2u – 2 byte unsigned (little-endian order)
- 2s – 2 byte signed (little-endian order)
- 4f – float (IEEE-754 standard)
- 4s – 4 bytes signed (little-endian order)
- 4u – 4 bytes unsigned (little-endian order)
- 8d – double (IEEE-754 standard)
- string – ASCII character array, first byte is array size
- Nb – byte array size N

Commands brief definition

Incoming (from sensor to controller):

Name	ID	
CMD_CONFIRM	1	Confirmation of previous command or finished calibration
CMD_RESET_NOTIFY	3	Notification on device reset
CMD_DEVICE_INFO	5	Board and firmware information
CMD_DATA	8	Configurable realtime data
CMD_USER_CONF_LOG	13	Configuration of user data log

Outgoing (from controller to sensor):

Name	ID	
CMD_RESET	2	Reset device
CMD_GET_DEVICE_INFO	4	Request board and firmware information

CMD_GET_DAT	6	Request configurable realtime data
CMD_GET_DATA_STREAM	7	Register or update *data stream*
CMD_CALIB	9	Calibration of the built-in sensors
CMD_BOOT_MODE	10	Enter firmware update mode (STM32 hardware loader)
CMD_USER_DATA_LOG	11	Contains data for logging to SD
CMD_GET_USER_CONF_LOG	12	Request configuration of user data log

Incoming commands

CMD_CONFIRM – confirmation of previous command or finished calibration

Name	Type	Possible values, remarks
CMD_ID	1u	Command ID to confirm
DATA	2u	DATA depends on command to be confirm

CMD_DEVICE_INFO – board and firmware information

Name	Type	Possible values, remarks
HARDWARE_VER	4u	Hardware version. This field includes the two values defined below. HARDWARE_VER_MAJOR = (int)(HARDWARE_VER >> 8) HARDWARE_VER_MINOR = (int)(HARDWARE_VER & 0x000000FF) The major version number is defined as the position number of a non-zero bit in the HARDWARE_VER_MAJOR.
HARDWARE_CMP	4u	Used as a bitmask for the major part of the HARDWARE_VER field to determine software and hardware compatibility. Test Example: HARDWARE_VER & HARDWARE_CMP & 0xFFFFFF00
SOFTWARE_VER	2u	Format: X.Y, where X = (int)(SOFTWARE_VER/100), Y = (int)(SOFTWARE_VER%100)
BUILD_NUMBER	4u	
MCU_SN	12b	
RESERVED	16b	

CMD_RESET_NOTIFY – notification on device reset

Name	Type	Possible values, remarks
CMD_ID	1u	ID of the command that caused the reset

CMD_DATA – configurable realtime data

Name	Size	Type	Bit	Name structure	Units	Possible values, remarks
FLAGS	Each bit in this field encodes the data set included in this command. Value is copied from the "FLAGS" field in the CMD_GET_DATA. See its specification for details.					
	4	4u				
FLAGS_EXT	This field present only if the `FLAGS.bit31` is set. It extends the possible set of flags. Reserved for future use.					

	4	4u				
TIMESTAMP_MS	Timestamp					
	4	4u	Timestamp		ms	
AHRS_STATUS						
	2	2u	bit0	ATTITUDE_INIT_OK		set if attitude is initialized from accelerometer
			bit1	HEADING_INIT_OK		set if heading is initialized from the reference sensor (compass or GPS)
			bit2	HEADING_REF_ENABLED		set if heading is referenced by sensor (compass or GPS)
			bit3	GNSS_REF_ENABLED		set is GNSS is used in a sensor fusion algorithm
			bit4 - bit5	QUALITY_CONDITION		BAD = 0, COARSE = 1, GOOD = 2, FINE = 3
HW_STATUS						
	2	2u	bit0	TERMOSTAT_TARGET		set if thermostat has reached the target temperature
			bit1	RTC_BAT_VALID*		set if the battery (backup RTC) is installed and not discharged
			bit2	SD_INSTALLED		set if SD is installed and functions correctly
			bit3	GNSS_ERROR		set if HW error GNSS subsystem
			bit4	MAG_ERROR		
			bit5	IMU_ERROR		
			bit6	CALIB		
FUSION_QLT	Assessment of the quality of sensor fusion on a scale from 0 to 255. Value of 255 means the best quality.					
	5	1u	FUSION_QLT_IMU			IMU
		1u	FUSION_QLT_MAG			Magnetometer
		1u	FUSION_QLT_GNSS			GNSS
		1u	FUSION_QLT_BARO			Barometer
		1u	Reserved			
DCM6	Attitude in DCM6 format.					
	24	4f	DCM11			
		4f	DCM12			
		4f	DCM13			
		4f	DCM31			
		4f	DCM32			
		4f	DCM33			
QUAT	Attitude in quaternion format.					
	16	4f	Q_W			

		4f	Q_X		
		4f	Q_Y		
		4f	Q_Z		
EULER321	Attitude in Euler angle (3-2-1) format.				
	12	4f	YAW	*degree*	
		4f	PITCH	*degree*	
		4f	ROLL	*degree*	
ACC_XYZ_LINER	Linear acceleration (without gravity) in XYZ reference frame. Gravity component has been removed using the current gravity reference vector estimate. In the stationary case, measurements on all axes are near zero.				
	12	4f	ACCEL_X	*m/s^2*	
		4f	ACCEL_Y	*m/s^2*	
		4f	ACCEL_Z	*m/s^2*	
ACC_NED_LINER	Linear acceleration (without gravity) in NED reference frame. Gravity component has been removed using the current gravity reference vector estimate. In the stationary case, measurements on all axes are near zero.				
	12	4f	ACCEL_N	*m/s^2*	
		4f	ACCEL_E	*m/s^2*	
		4f	ACCEL_D	*m/s^2*	
VELO_XYZ	Velocity in XYZ reference frame.				
	12	4f	VELO_X	*m/s*	
		4f	VELO_Y	*m/s*	
		4f	VELO_Z	*m/s*	
VELO_NED	Velocity in NED reference frame.				
	12	4f	VELO_N	*m/s*	
		4f	VELO_E	*m/s*	
		4f	VELO_D	*m/s*	
VELO_U*	Velocity uncertainty.				
	4	4f	VELO_U	*m/s*	
POS_NED	Position in NED reference frame. Starting point is taken as the origin.				
	12	4f	POS_N	*m*	
		4f	POS_E	*m*	
		4f	POS_D	*m*	
POS_LLA	Position in LLA reference frame.				
	24	8d	POS_LAT	*degree*	Latitude
		8d	POS_LON	*degree*	Longitude
		8d	POS_ALT	*m*	Altitude
POS_U*	Position uncertainty.				

	4	4f	POS_U	*m*	
MAG_XYZ	Magnetic field in XYZ reference frame. This measurement is compensated by the static (Hard/Soft Iron calibration stored in flash) and by the dynamic calibration (Hard Iron).				
	12	4f	MAG_X	*Gauss*	
		4f	MAG_Y	*Gauss*	
		4f	MAG_Z	*Gauss*	
MAG_NED	Magnetic field in NED reference frame. This measurement is compensated by the static (Hard/Soft Iron calibration stored in flash) and by the dynamic calibration (Hard Iron). The current attitude solution is used to map the measurement from the measured XYZ frame to NED frame.				
	12	4f	MAG_N	*Gauss*	
		4f	MAG_E	*Gauss*	
		4f	MAG_D	*Gauss*	
GYR_XYZ	Angular rate in XYZ reference frame. This measurement is compensated by the static (calibration stored in flash) and dynamic calibration.				
	12	4f	GYR_X	rad/s	
		4f	GYR_Y	rad/s	
		4f	GYR_Z	rad/s	
GYR_NED	Angular rate in NED reference frame. This measurement is compensated by the static (calibration stored in flash) and dynamic calibration. The current attitude solution is used to map the measurement from the measured XYZ frame to NED frame.				
	12	4f	GYR_N	rad/s	
		4f	GYR_E	rad/s	
		4f	GYR_D	rad/s	
ACC_XYZ	Acceleration (with gravity) in XYZ reference frame. This measurement is compensated by the static calibration (calibration stored in flash).				
	12	4f	ACC_X	*m/s^2*	
		4f	ACC_Y	*m/s^2*	
		4f	ACC_Z	*m/s^2*	
ACC_NED	Acceleration (with gravity) in NED reference frame. This measurement is compensated by the static calibration (calibration stored in flash). The current attitude solution is used to map the measurement from the measured XYZ frame to NED frame.				
	12	4f	ACC_N	*m/s^2*	
		4f	ACC_E	*m/s^2*	
		4f	ACC_D	*m/s^2*	
GNSS_STATE	GNSS state solution. Update data rate 10 Hz.				
	2	1u	GNSS_FIX		0 – no fix, 1 – dead reckoning only, 2 – 2D-fix, 3 – 3D-fix
		1u	GNSS_SAT		The number of tracked GNSS satellites
GNSS_POS_LLA	GNSS position in LLA reference frame. Update data rate 10 Hz.				

	24	8d	GNSS_LAT	*degree*	Latitude
		8d	GNSS_LON	*degree*	Longitude
		8d	GNSS_ALT	*m*	Altitude
GNSS_DOP	GNSS dilution of precision (DOP). Update data rate 10 Hz.				
	28	4f	gDOP		Geometric DOP
		4f	pDOP		Position DOP
		4f	tDOP		Time DOP
		4f	vDOP		Vertical DOP
		4f	hDOP		Horizontal DOP
		4f	nDOP		Northing DOP
		4f	eDOP		Easting DOP
GNSS_VEL_NED	GNSS velocity in NED reference frame. Update data rate 10 Hz.				
	12	4f	GNSS_VEL_N	*m/s*	
		4f	GNSS_VEL_E	*m/s*	
		4f	GNSS_VEL_D	*m/s*	
GNSS_VEL_U	GNSS velocity uncertainty. Update data rate 10 Hz.				
	4	4f	GNSS_VEL_U	*m/s*	
BARO_PRSR	Absolute air pressure.				
	4	4f	BARO_PRSR	kPa	Typical pressure at sea level would be around 101.325 kPa.
BARO_ALT	Barometric altitude.				
	4	4f	BARO_ALT	m	
TEMP_BOARD	Sensor temperature on board				
	12	4f	TEMP_IMU	C	
		4f	TEMP_BARO	C	
		4f	TEMP_CPU*	C	
AVERAGE_TIME	Precise time interval where data was averaged for this sample (if averaging is enabled by the AVG_MASK).				
	4	4f	AVERAGE_TIME	*s*	

CMD_USER_CONF_LOG – configuration of user-defined data enabled for logging

	Name	Type	Bit	Possible values, remarks
STREAM1	ACTIV_PIPE_MASK	4u	bit0 - bit31	Bitmask of the pipes enabled for logging in this stream, where the bit number corresponds to the index of a pipe. To save bandwidth, send only data of pipes that are enabled.
	INTERVAL_MS	2u		Interval between log events in this stream. Use it for reference only; you can send data with different interval (see CMD_USER_DATA_LOG for details)

STREAM2	ACTIV_PIPE_MASK	4u	bit0 - bit31	the same as above
	INTERVAL_MS	2u		the same as above

Outgoing commands

CMD_GET_DEVICE_INFO – request board and firmware information

No parameters

CMD_RESET – reset device

Name	Type	Possible values, remarks
CONFIRM	1u	0 – no confirmation 1 - command CMD_RESET_NOTIFY will be sent back for confirmation before device reset
DELAY_MS	2u	Waits for a given time (in ms) before reset.

CMD_GET_DATA_STREAM – register or update *data stream* – a commands sent by the controller with the fixed rate

For each serial interface, only one unique combination of CMD_ID + CONFIG bytes may be registered. If the data stream is already registered, it will be updated. To unregister it, specify INTERVAL_MS=0. The total number of data streams over all serial interfaces is limited to 10.

Take care of the serial bandwidth: if data flow exceeds bandwidth, particular samples may be skipped.

The interval is maintained with the +-1ms tolerance for the individual sample, but the averaged sample rate exactly matches to specified. If the data stream is successfully registered or updated, the CMD_CONFIRM is sent in answer.

All vector-like variables (for example, gyroscope and accelerometer) may be pre-integrated to process them at lower data rate without loosing of information. The averaging can be enabled using the AVG_MASK parameter. Averaged values have the same units as the instant values. They can be converted to integrals (*theta_angle, theta_velocity*) by multiplying by the "AVERAGE_TIME" variable.

$$avg(v(t), t, T) = \frac{\int_{t}^{t+T} v(t) \cdot dt}{T},$$

Name	Type	Possible values, remarks
CMD_ID	1u	Command ID to be sent by this data stream. All supported commands are listed for the "CONFIG" parameter below. If the command is set to 0, all data streams will be disabled.
INTERVAL_MS	2u	Interval between messages, in milliseconds. Send value 0 to unregister data stream.
CONFIG	8b	Bit mask specified in the CMD_GET_DATA • **FLAGS1** – 4u • **FLAGS2** – 4u
AVG_MASK	8b	For the bits in mask set to 1, the corresponding data will be averaged on the given time interval INTERVAL_MS. The exact average time for each sample in this data stream can be received in the variable AVERAGE_TIME • **FLAGS1_AVG** – 4u • **FLAGS2 _AVG** – 4u
RESERVED	16b	

CMD_GET_DATA – request configurable realtime data

Name	Type	Bit		Possible values, remarks
FLAGS	*A detailed description of the data structure is provided in the CMD_DATA specification.* Each bit specifies which data set to include in response:			
	4u	bit0	TIMESTAMP_MS	Timestamp
		bit1	AHRS_STATUS	AHRS status
		bit2	HW_STATUS	Hardware status
		bit3	FUSION_QLT	Assessment of the quality of sensor fusion
		bit4	DCM6	Attitude in DCM6 format
		bit5	QUAT	Attitude in quaternion format
		bit6	EULER321	Attitude in Euler angle (3-2-1) format
		bit7	ACCEL_XYZ	Linear acceleration (without gravity) in XYZ reference frame
		bit8	ACCEL_NED	Linear acceleration (without gravity) in NED reference frame
		bit9	VELO_XYZ	Velocity in XYZ reference frame
		bit10	VELO_NED	Velocity in NED reference frame
		bit11	VELO_U	Velocity uncertainty
		bit12	POS_NED	Position in NED reference frame
		bit13	POS_LLA	Position in LLA reference frame
		bit14	POS_U	Position uncertainty
		bit15	MAG_XYZ	Magnetic field in XYZ reference frame
		bit16	MAG_NED	Magnetic field in NED reference frame
		bit17	GYR_XYZ	Angular rate in XYZ reference frame

		bit18	GYR_NED	Angular rate in NED reference frame
		bit19	ACC_XYZ	Acceleration (with gravity) in XYZ reference frame
		bit20	ACC_NED	Acceleration (with gravity) in NED reference frame
		bit21	GNSS_STATE	GNSS state solution
		bit22	GNSS_POS_LLA	GNSS position in LLA reference frame.
		bit23	GNSS_DOP	GNSS dilution of precision (DOP)
		bit24	GNSS_VEL_NED	GNSS velocity in NED reference frame
		bit25	GNSS_VEL_U	GNSS velocity uncertainty
		bit26	BARO_PRSR	Absolute air pressure
		bit27	BARO_ALT	Barometric altitude
		bit28	TEMP_BOARD	Sensor temperature on board
		bit29	AVERAGE_TIME	Time interval for averaging
		bit30	RESERVED	
		bit31	use FLAGS_EXT parameter	
FLAGS_EXT	This value is used only if `FLAGS.bit31` is set. It extends the range of "FLAGS" field and reserved for future use.			
	4u	bit0	RESERVED	
		bit1	RESERVED	
		bit2	RESERVED	
		bit3	RESERVED	
		bit4	RESERVED	
		bit5	RESERVED	
		bit6	RESERVED	
		bit7	RESERVED	
		bit8	RESERVED	
		bit9	RESERVED	
		bit10	RESERVED	
		bit11	RESERVED	
		bit12	RESERVED	
		bit13	RESERVED	
		bit14	RESERVED	
		bit15	RESERVED	
		bit16	RESERVED	
		bit17	RESERVED	
		bit18	RESERVED	
		bit19	RESERVED	
		bit20	RESERVED	
		bit21	RESERVED	
		bit22	RESERVED	
		bit23	RESERVED	

		bit24	RESERVED	
		bit25	RESERVED	
		bit26	RESERVED	
		bit27	RESERVED	
		bit28	RESERVED	
		bit29	RESERVED	
		bit30	RESERVED	
		bit31	RESERVED	
RESERVED	4b			

CMD_CALIB – calibration of the built-in sensor

If the calibration process is successfully started, the CMD_CONFIRM (with DATA [1u: SENSOR_TYPE, 1u: 0]) is sent in response. After successful completion of the calibration process, the CMD_CONFIRM (with DATA [1u: SENSOR_TYPE, 1u: 1]) is sent.

Name	Type	Possible values, remarks
SENSOR_TYPE	1u	1 – Accelerometer, 2 – Gyroscope, 3 – Magnetometer
RESERVED	10b	

CMD_BOOT_MODE – Enter firmware update mode (STM32 hardware loader)

Name	Type	Possible values, remarks
CONFIRM	1u	0 – no confirmation 1 - command CMD_RESET_NOTIFY will be sent back for confirmation before device reset
DELAY_MS	2u	Waits for a given time (in ms) before reset and enter firmware update mode

CMD_USER_DATA_LOG – Contains data for logging to SD card

Send user-defined data to be logged to SD card, if it is configured and enabled in the "CONF_LOG.INI". Data goes in a pipes, each pipe have its type and number of values, specified in the "PIPE_CONF" field. This configuration should exactly match the pipe configuration in the "CONF_LOG.INI", otherwise data will be skipped.

The PIPES[] array should be ordered by the index of a pipe.

You can send several sets of pipes with different rates in multiple messges, if there are high-rate and low-rate varying data.

Note the the logging event is not synchronized with this message - it always use the latestly arrived data, regardless of the rate it comes. You can pass a custom timestamp as a part of user-defined data to have precise time information.

<table>
<tr><th colspan="2">Name</th><th>Type</th><th>Bit</th><th>Name structure</th><th>Possible values, remarks</th></tr>
<tr><td colspan="2">ACTIVE_PIPE_MASK</td><td>4u</td><td>bit0 - bit31</td><td colspan="2">Bitmask specify active pipes transferred in this message. The index of each bit (0..31) corresponds to the index of each pipe; the number of enabled bits should match the number N of elements in PIPES[] array further in this message.</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td rowspan="4">PIPES[N]</td><td rowspan="3">PIPE_CONF</td><td rowspan="3">1u</td><td>bit0 - bit3</td><td>PIPE_SIZE</td><td>Number of values in this channel, 1..15</td></tr>
<tr><td>bit4 - bit5</td><td>PIPE_TYPE</td><td>Type of values:
0 – reserved,
1 – 4f,
2 – 4s,
3 – 2s,</td></tr>
<tr><td>bit6 - bit7</td><td>RESERVED</td><td></td></tr>
<tr><td>PIPE_DATA</td><td>Variable</td><td colspan="3">Data set defined as an array of values with type (PIPE_TYPE) and size (PIPE_SIZE).</td></tr>
</table>

CMD_GET_USER_CONF_LOG – request a configuration of user-defined data for logging

No parameters.

The CMD_USER_CONF_LOG is sent in response.

Appendix A: Code examples

CRC16 reference implementation in C

```
void crc16_update(uint16_t length, uint8_t *data, uint8_t crc[2]) {
    uint16_t counter;
    uint16_t polynom = 0x8005;
    uint16_t crc_register = (uint16_t)crc[0] | ((uint16_t)crc[1] << 8);
    uint8_t shift_register;
    uint8_t data_bit, crc_bit;

    for (counter = 0; counter < length; counter++) {
        for (shift_register = 0x01; shift_register > 0x00; shift_register <<= 1) {
            data_bit = (data[counter] & shift_register) ? 1 : 0;
            crc_bit = crc_register >> 15;
            crc_register <<= 1;

            if (data_bit != crc_bit) crc_register ^= polynom;
        }
    }

    crc[0] = crc_register;
    crc[1] = (crc_register >> 8);
}

void crc16_calculate(uint16_t length, uint8_t *data, uint8_t crc[2]) {
    crc[0] = 0; crc[1] = 0;
    crc16_update(length, data, crc);
}
```

Command ID definitions

```
#define CMD_CONFIRM  1
#define CMD_RESET 2
#define CMD_RESET_NOTIFY 3
#define CMD_GET_DEVICE_INFO 4
#define CMD_DEVICE_INFO 5
#define CMD_GET_DATA 6
#define CMD_GET_DATA_STREAM 7
#define CMD_DATA 8
#define CMD_CALIB 9
#define CMD_BOOT_MODE 10
```