Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

UM1744
User manual

STEVAL- MKI121V1 communication protocol

Introduction

The scope of this user manual is to present the communication protocol used between the STEVAL-MKI121V1 product evaluation board (Discovery-M1) and the iNEMO SDK (Software Development Kit). This communication protocol runs upon a physical communication channel based on USB virtual COM, which represents the physical channel used in the communication between the STEVAL-MKI121V1 and the PC.

The first chapter explains the general frame format and main rules used in the protocol.

The second chapter explains all the frames used in the actual release of the embedded firmware and Software Development Kit (SDK).

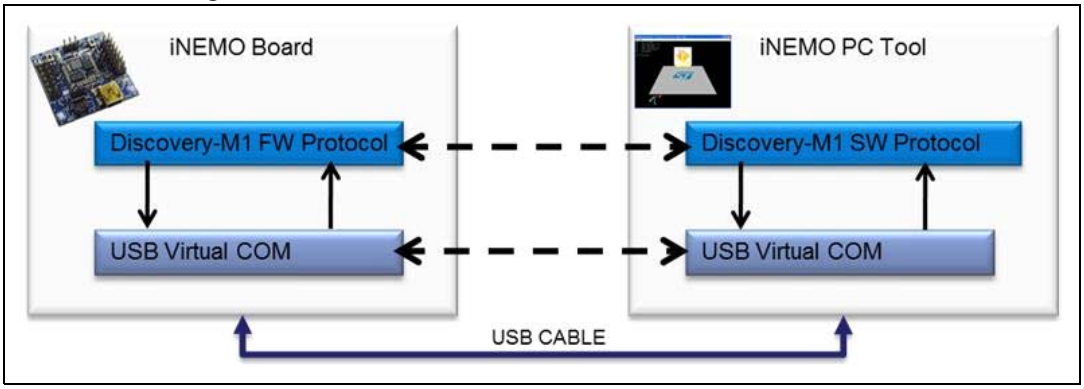
Contents

1 General frame format and protocol rules

1.1 Frame format

This paragraph explains the format of the frame used in the STEVAL-MKI121V1 communication protocol. Because, the STEVAL-MKI121V1 exchanges data and commands with the PC GUI through a physical communication channel based on a USB Virtual COM, each frame, described below, represents the payload of a USB frame.

Figure 1. STEVAL-MKI121V1 communication architecture



The frames are described as a sequence of fields in a specific order. All frame formats are depicted in the order in which they are passed to the USB driver, from left to right. Bits within each field are numbered from k-1 (leftmost and most significant) to 0 (rightmost and least significant), where the length of the field is k bits.

The frame format is composed of a header and an optional payload. The general frame shall be formatted as illustrated in *Figure 2*. The header is composed of three mandatory (M) fields, each of which is 1 byte in length, while the payload is an optional field whose maximum length is 61 bytes. See LF/MF field in the following section to overcome this limit.

Figure 2. General frame format

Bytes: 1	1	1	≤ 61
Frame Control (M)	Length (M)	Message ID (M)	Payload (O)

1.1.1 Frame control field

The frame control field is 1 byte in length and contains information defining the frame type and other control flags. The frame control field shall be formatted as illustrated in *Figure 3*.

Figure 3. Frame control field

Bit: 7	6	5	4	3	2	1	0
Frame Type		Ack	LF/MF	Frame Version		QoS	

The frame type subfield is 2 bits in length and shall be set to one of the values listed in *Table 1*.

Table 1. Frame type list

Value	Frame type
00	CONTROL
01	DATA
10	ACK
11	NACK

The Ack subfield is 1 bit in length and specifies whether an acknowledgment is required from the recipient on receipt of a DATA or CONTROL frame. If this subfield is set to one, the recipient shall send an acknowledgment frame only if, upon reception, the frame passes all the needed levels of filtering. If this subfield is set to zero, the recipient device shall not send an acknowledgment frame. It is possible to embed a payload in an acknowledgment frame (piggybacking) to send useful information to the transmitter and avoiding further transactions. When the Ack field is set to one and upon reception the frame doesn't pass the needed level of filtering, the recipient shall send a not-acknowledgment frame (NACK), whose payload is an error code (e.g. unsupported command, value out of range,…). In the ACK and/or NACK frames the Ack field shall be set to zero and ignored on reception.

The LF/MF (Last Fragment / More Fragment) subfield is 1 bit in length and it is used for fragmentation and reassembling. This field is set to zero to indicate a single frame or the last frame of a multiple-frame transaction. This field is set to 1 to indicate that other frames will follow all those belonging to the same transaction. In the ACK and NACK frames (with or without payload) fragmentation is not supported and this subfield shall be set to zero in transmission of ACK and NACK frames and ignored on reception.

The frame version subfield is 2 bits in length and shall be set to the non-reserved for future use (RFU) value listed in *Table 2*.

Table 2. Frame version list

Value	Frame version
00	Version 1
01	RFU
10	
11	

The QoS (Quality of Service) subfield is 2 bits in length and shall be set to one of the values listed in *Table 3*. This subfield allows the application to exchange and process data and control frames with different priorities.

Table 3. QoS list

Value	Frame version
00	Normal Priority
01	Medium Priority
10	High Priority
11	RFU

1.1.2 Length field

The length field is 1 byte in length and contains the number of bytes that follow the length field. Admitted values are in the range 1 ÷ 62.

1.1.3 Message ID field

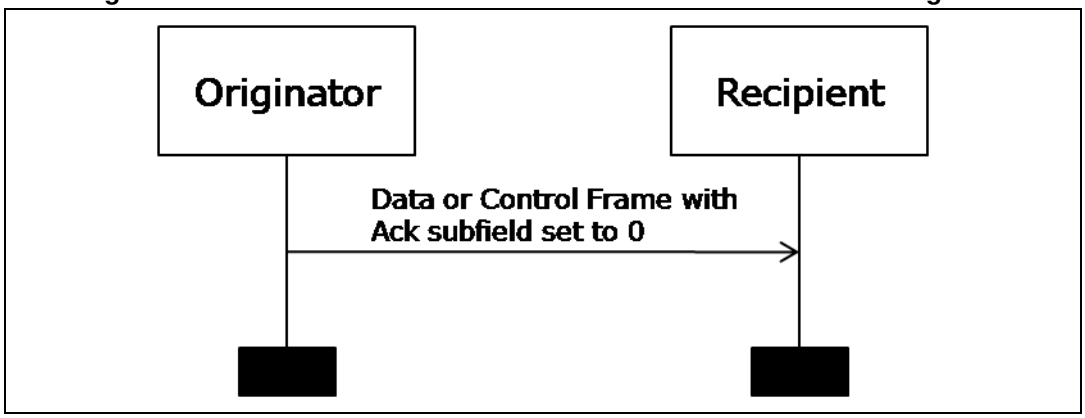
The message ID is 1 byte in length and contains an identifier of the user application messages. See *Section 2.2* and the following sections for further details.

1.2 Protocol rules

There are two types of transactions: acknowledgment or non-acknowledgment of the DATA or CONTROL frame.

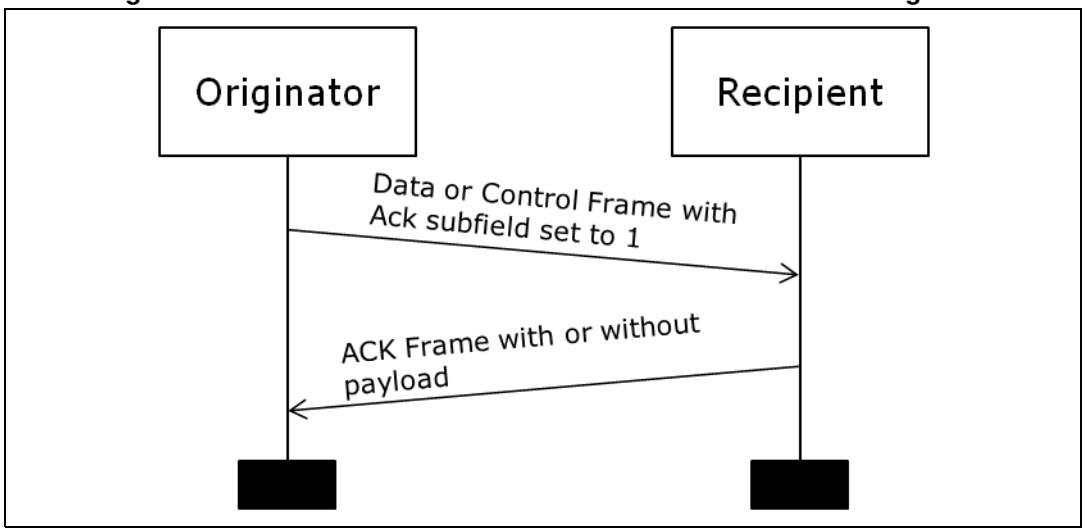
A DATA or CONTROL frame with the Ack subfield of its frame control field set to zero shall not be acknowledged by its intended recipient. The originating device (PC or Discovery-M1 board) shall assume that the transmission of the frame was successful. The message sequence chart in *Figure 4* shows the scenario for transmitting a single DATA or CONTROL frame from an originator to a recipient without requiring an acknowledgment.

Figure 4. Data or control frame transmission without an acknowledgment



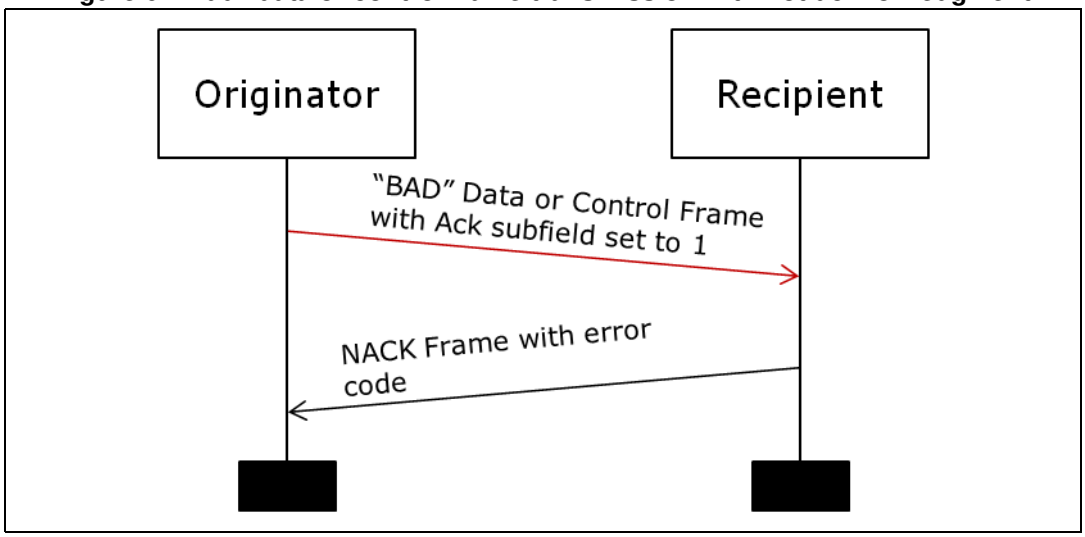
A DATA or CONTROL frame transmitted with the Ack subfield of its frame control field set to one shall be acknowledged by the recipient. If the intended recipient correctly receives the frame, it shall generate and send an ACK frame containing the same message ID from the DATA or CONTROL frame that is being acknowledged. It is possible also to include a payload in the ACK frame to transfer useful data from the recipient to the originator. The message sequence chart in *Figure 5* shows the scenario for transmitting a single DATA or CONTROL frame from an originator to a recipient with an acknowledgment.

Figure 5. Data or control frame transmission with an acknowledgment



If the frame received does not conform to all the required filtering rules, the recipient shall generate and send a NACK frame containing the same message ID from the DATA or CONTROL frame that is being acknowledged and containing the error code. The message sequence chart in *Figure 6* shows the scenario for transmitting a single "bad" DATA or CONTROL frame from an originator to a recipient with a not-acknowledgment.

Figure 6. "Bad" data or control frame transmission with not-acknowledgment



2 STEVAL-MKI121V1 frames

2.1 STEVAL-MKI121V1 frame types

The frames used in the STEVAL-MKI121V1 are classified in five types:

1. Communication control frames
2. Board information frames
3. Sensor setting frames
4. Acquisition sensor data frames

2.2 Communication control frames

Communication control frames are frames originated by the software PC (SDK or GUI) and used to send specific commands to the Discovery-M1 board. All the communication control frames are listed in *Table 4*.

Table 4. Communication control frames

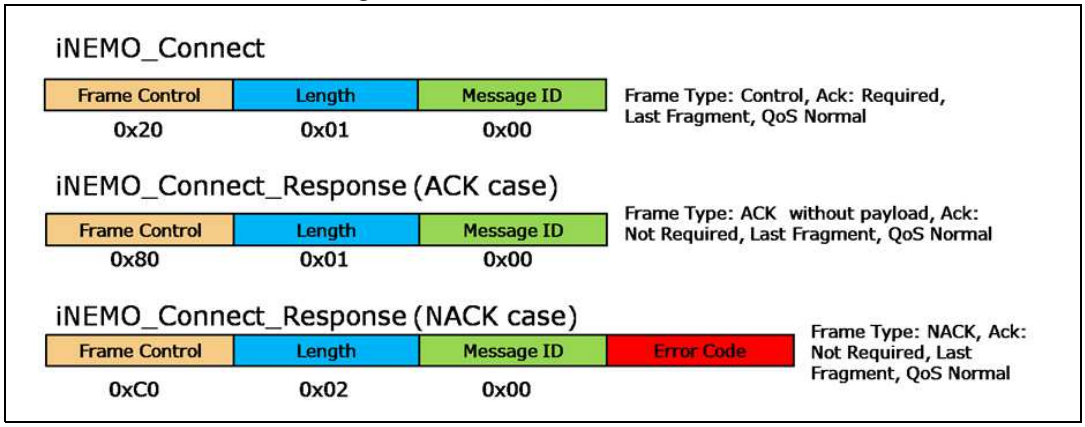
Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
iNEMO _Connect	CONTROL	Y	0x00	N	0		PC
iNEMO _Connect_Response	ACK	N	0x00	N	0		Discovery-M1
	NACK	N	0x00	N	1	Error Code	
iNEMO _Disconnect	CONTROL	Y	0x01	N	0		PC
iNEMO_Disconnect_Response	ACK	N	0x01	N	0		Discovery-M1
	NACK	N	0x01	N	1	Error Code	
iNEMO_Reset_Board	CONTROL	Y	0x02	N	0		PC
iNEMO_Reset_Board_Response	ACK	N	0x02	N	0		Discovery-M1
	NACK	N	0x02	N	1	Error Code	
iNEMO_Enter_DFU_Mode	CONTROL	Y	0x03	N	0		PC
iNEMO_Enter_DFU_Mode_ Response	ACK	N	0x03	N	0		Discovery-M1
	NACK	N	0x03	N	1	Error Code	
iNEMO_Trace	CONTROL	Y	0x07	N	0		PC
iNEMO_Trace_Response	ACK	N	0x07	N	0		Discovery-M1
	NACK	N	0x07	N	1	Error Code	
iNEMO_Trace_Data	DATA	N	0x07	M	Variable	String for debug purpose	
iNEMO_Led_Control	CONTROL	Y	0x08	N	1	0x00 OFF 0x01 ON	PC
iNEMO_Led_Control_Response	ACK	N	0x08	N	0		Discovery-M1
	NACK	N	0x08	N	1	Error Code	

2.2.1 iNEMO_Connect

The iNEMO_Connect command shall be the first command sent from the GUI or SDK to the Discovery-M1 board. Any other command sent before the iNEMO_Connect will not be processed by Discovery-M1. It works like a "ping" and opens the communication between the GUI or SDK and the Discovery-M1 board at the application level.

Figure 7 shows the frames involved in the iNEMO_Connect transaction.

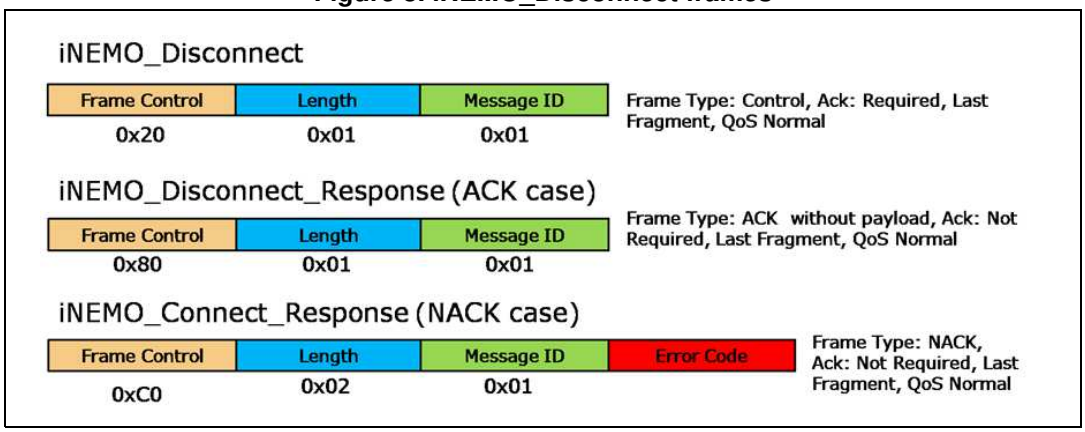
Figure 7. iNEMO_Connect frames



2.2.2 iNEMO_Disconnect

The iNEMO_Disconnect command closes the communication between the PC and the Discovery-M1 board. *Figure 8* shows the frames involved in the iNEMO_Disconnect transaction.

Figure 8. iNEMO_Disconnect frames



The GUI (or SDK), after receiving the ACK frame, shall close the USB Virtual Com. To re-open the communication only the iNEMO_Connect command shall be used.

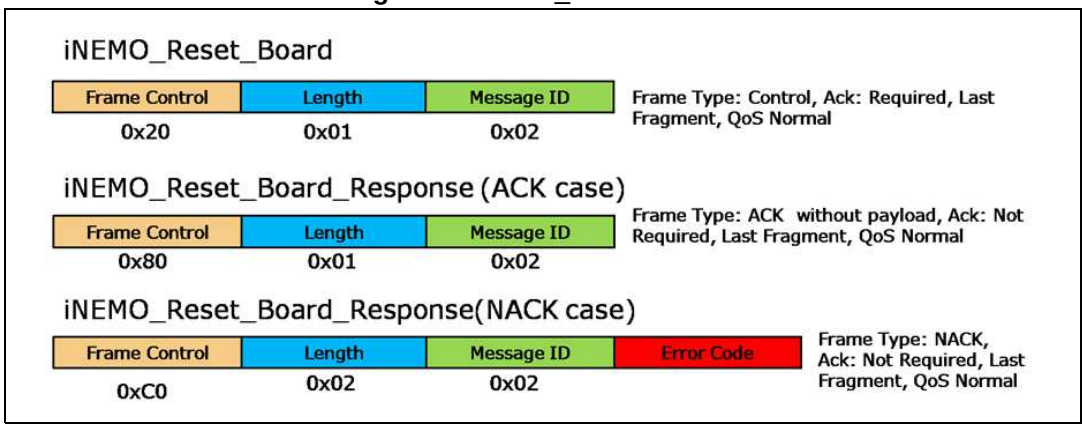
2.2.3 iNEMO_Reset

The iNEMO_Reset command implies a software reset of the Discovery-M1 board. After receiving the iNEMO_Reset command, the Discovery-M1 board replies with the ACK frame; then waits for 5 seconds before disconnecting the USB cable in software mode and invokes a software reset. The GUI (or SDK), after receiving the ACK frame, shall close the USB

Virtual Com. To re-open the communication only the iNEMO_Connect command shall be used.

Figure 9 shows the frames involved in the iNEMO_Reset transaction.

Figure 9. iNEMO_Reset frames

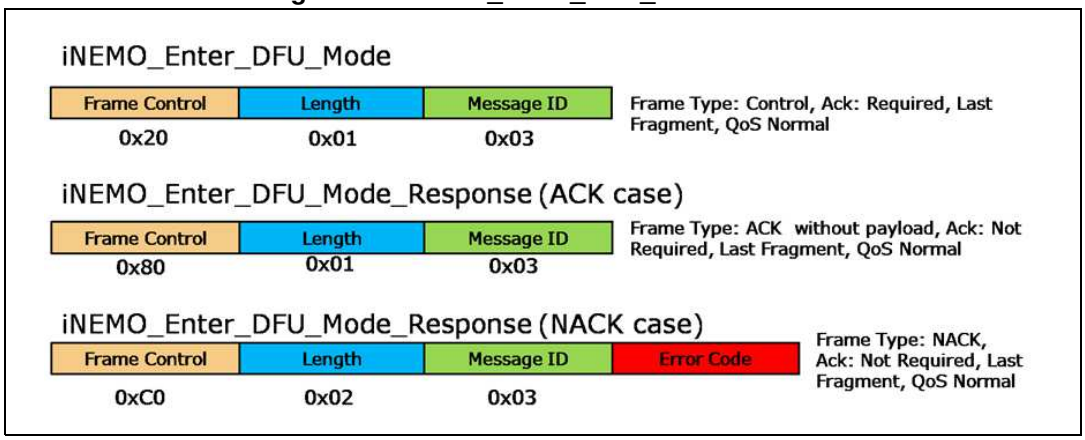


2.2.4 iNEMO_Enter_DFU_Mode

The iNEMO_Enter_DFU_Mode command allows the Discovery-M1 board to enter in DFU mode in software mode. After receiving the iNEMO_Enter_DFU_Mode command, the Discovery-M1 board replies with an ACK frame. Then it will set the Option Byte Data0 (at address 0x1FFFF804) to one, will disconnect the USB cable in software mode, and it will invoke a software reset. After reset, the Discovery-M1 will enter in DFU mode. After entering in DFU mode in software, the Discovery-M1 will change the Option Byte Data0 to zero. The user can leave the DFU mode in two ways: by unplugging and plugging in the USB cable (hardware mode), or by using the Leave_DFU_Mode command available in the DfuSe Demo PC application or in the GUI or SDK. The GUI (or SDK) shall close the USB Virtual Com after receiving the ACK frame.

Figure 10 shows the frames involved in the iNEMO_Enter_DFU_Mode transaction

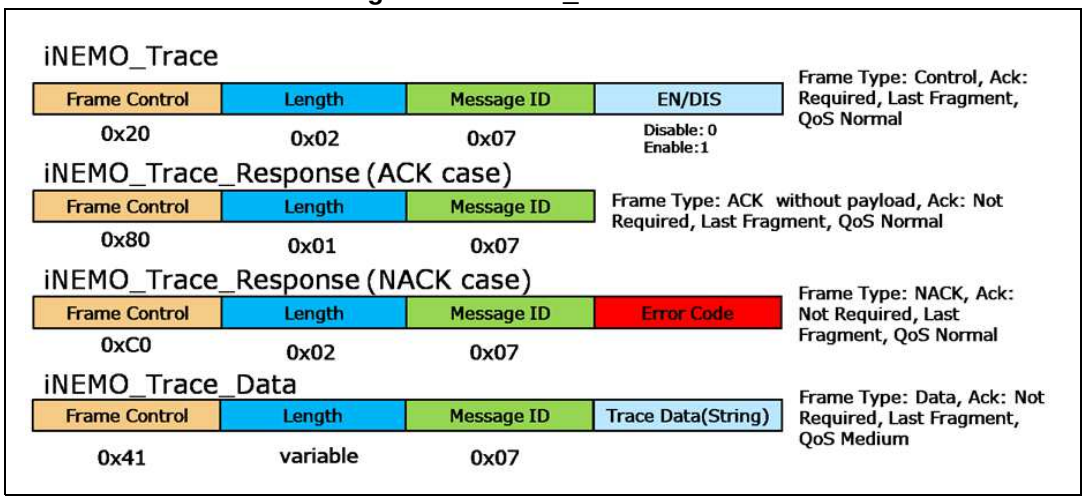
Figure 10. iNEMO_Enter_DFU_Mode frames



2.2.5 iNEMO_Trace

The iNEMO_Trace command allows the user to enable or disable "trace data". Trace data are used for debugging purposes and they will be string displayed in a debug window. The frames are asynchronous and shall have medium priority (QoS sub-field of frame control field). *Figure 11* shows the frames involved in the iNEMO_Trace transaction

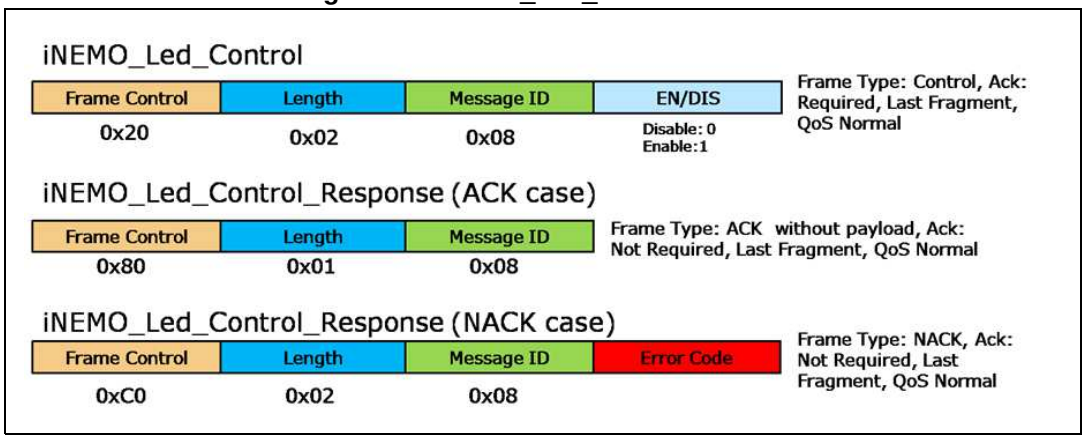
Figure 11. iNEMO_Trace frames



2.2.6 iNEMO_Led_Control

The iNEMO_Led_Control command allows turning on and off the LED available on the iNEMO board. *Figure 12* shows the frames involved in the iNEMO_Led_Control transaction.

Figure 12. iNEMO_Led_Control frames



2.3 Board information frames

Board information frames are frames originated by the software PC (SDK or GUI) and used to retrieve information about firmware and hardware features of the Discovery-M1 board. All the board information frames are listed in *Table 5*.

Table 5. Board information frames

Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
iNEMO_Get_Device_Mode	CONTROL	Y	0x10	N	0		PC
iNEMO_Get_Device_Mode_ Response	ACK	N	0x10	N	1	0x00 Sensor Mode 0x01 Master Mode	Discovery-M1
	NACK	N	0x10	N	1	Error Code	
iNEMO_Get_MCU_ID	CONTROL	Y	0x12	N	0		PC
iNEMO_Get_MCU_ID_Response	ACK	N	0x12	N	12	Unique Device ID	Discovery-M1
	NACK	N	0x12	N	1	Error Code	
iNEMO_Get_FW_Version	CONTROL	Y	0x13	N	0		PC
iNEMO_Get_FW_Version_ Response	ACK	N	0x13	N	Variable	String Firmware Version	Discovery-M1
	NACK	N	0x13	N	1	Error Code	
iNEMO_Get_HW_Version	CONTROL	Y	0x14	N	0	Date, Time	PC
iNEMO_Get_HW_Version_ Response	ACK	N	0x14	N	Variable	String Hardware Version	Discovery-M1
	NACK	N	0x14	N	1	Error Code	
iNEMO_Identify	CONTROL	Y	0x15	N	0		PC
iNEMO_Identify_Response	ACK	N	0x15	N	12	Unique Device ID	Discovery-M1
	NACK	N	0x15	N	1	Error Code	
iNEMO_Get_AHRS_Library	CONTROL	Y	0x17	N	0		PC
iNEMO_Get_AHRS_Library_ Response	ACK	N	0x17	N	Variable	AHRS enable/disable string	Discovery-M1
	NACK	N	0x17	N	1	Error Code	
iNEMO_Get_Libraries	CONTROL	Y	0x18	N	0		PC
iNEMO_Get_Libraries_Response	ACK	N	0x18	N	0	List of supported libraries	Discovery-M1
	NACK	N	0x18	N	1	Error Code	
iNEMO_Get_Available_Sensors	CONTROL	Y	0x19	N	0		PC

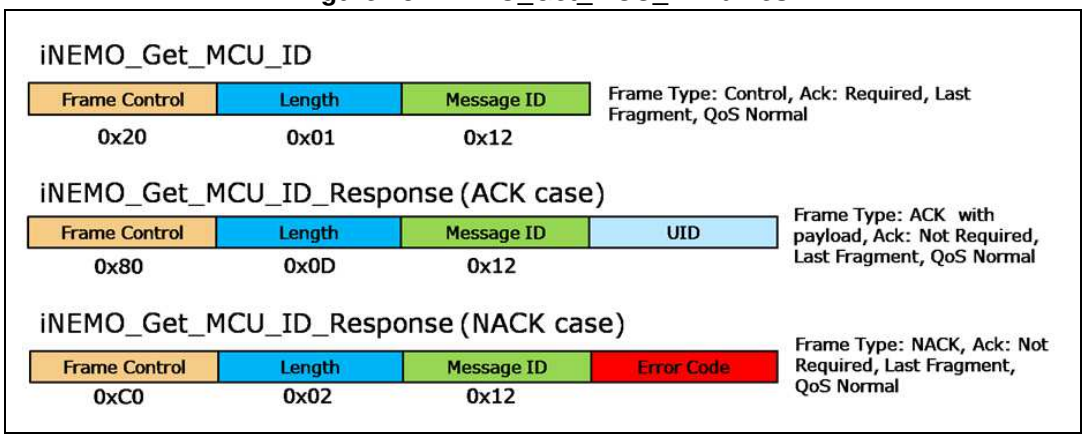
Table 5. Board information frames (continued)

Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
iNEMO_Get_Available_Sensors_ Response	ACK	N	0x19	N	1	List of available sensors	Discovery-M1
	NACK	N	0x19	N	1	Error Code	

2.3.1 iNEMO_Get_MCU_ID

The iNEMO_Get_MCU_ID command allows retrieving from the Discovery-M1 board the 96-bit unique device identifier of the STM32F103 microcontroller. *Figure 13* shows the frames involved in the iNEMO_Get_MCU_ID transaction.

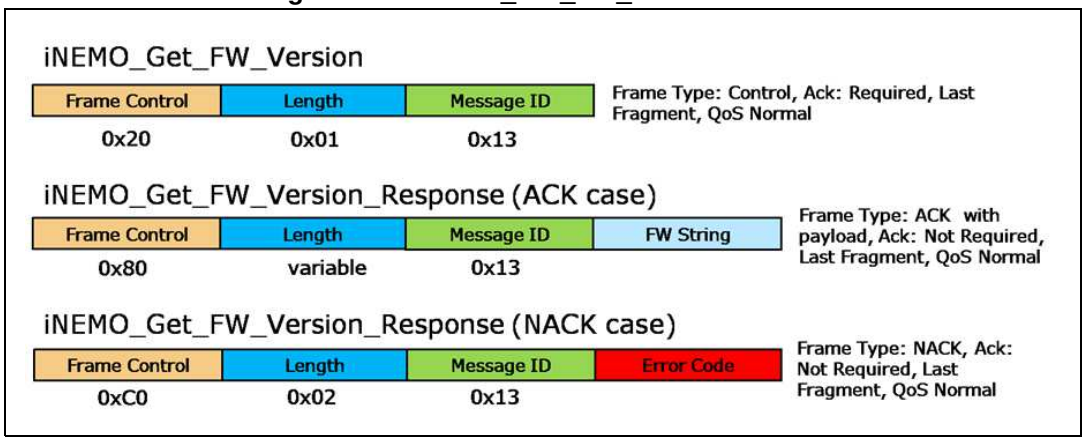
Figure 13. iNEMO_Get_MCU_ID frames



2.3.2 iNEMO_Get_FW_Version

The iNEMO_Get_FW_Version command allows retrieving the board firmware version. *Figure 14* shows the frames involved in the iNEMO_Get_FW_Version transaction

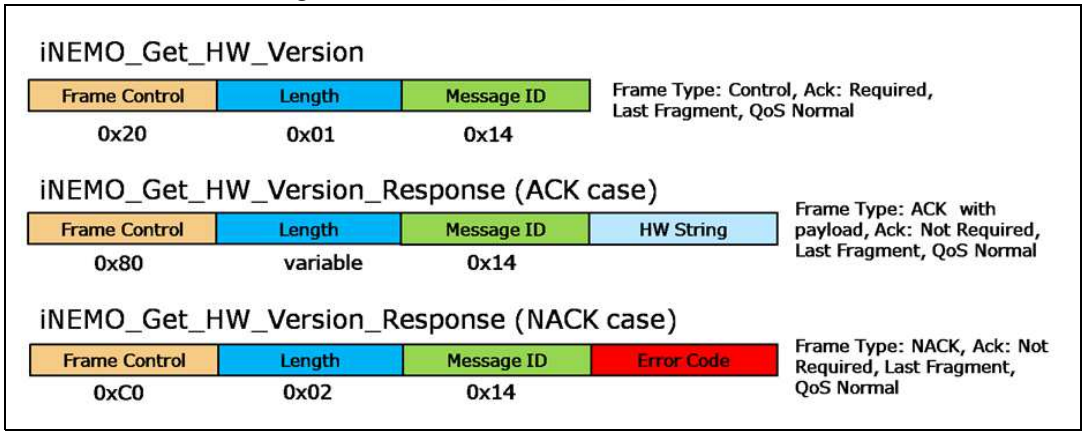
Figure 14. : iNEMO_Get_FW_Version frames



2.3.3 iNEMO_Get_HW_Version

The iNEMO_Get_HW_Version command allows retrieving the board hardware version. *Figure 15* shows the frames involved in the iNEMO_Get_HW_Version transaction.

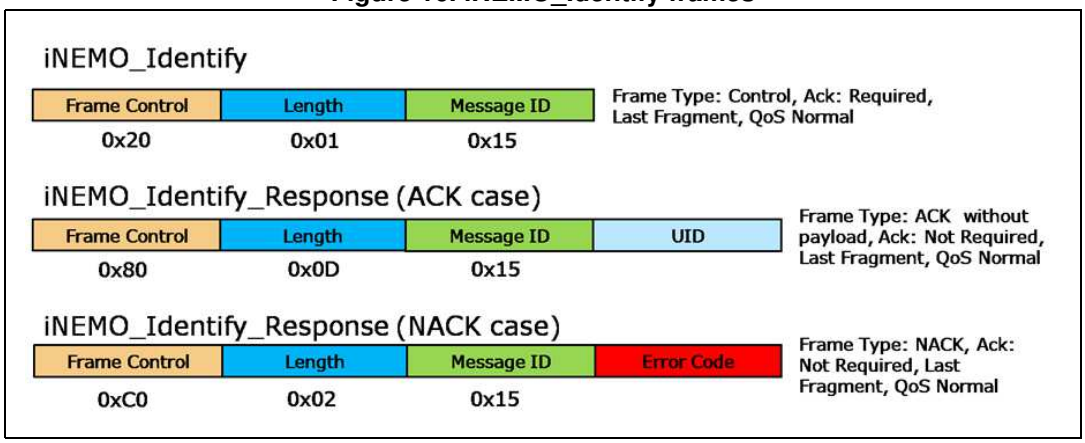
Figure 15. iNEMO_Get_HW_Version frames



2.3.4 iNEMO_Identify

The iNEMO_Identify command can be used to identify a Discovery-M1 board. Upon reception of the iNEMO_Identify command the Discovery-M1 board replies with an ACK containing the MCU Unique Device ID. Then the LED available on the board will blink 3 times. *Figure 16* shows the frames involved in the iNEMO_Identify transaction

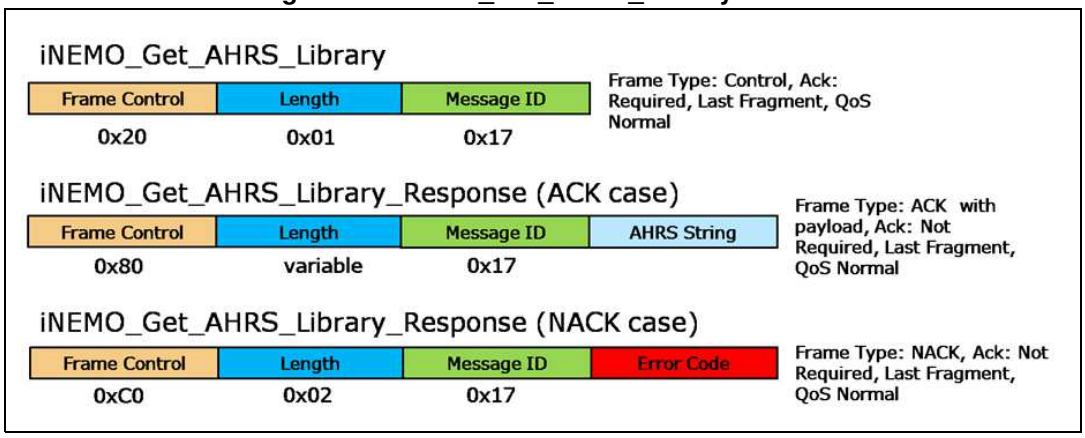
Figure 16. iNEMO_Identify frames



2.3.5 iNEMO_Get_AHRS_Library

The iNEMO_Get_AHRS_Library command allows knowing the version of the Discovery-M1 firmware Attitude Heading Reference System (AHRS) algorithm. The returned value is in string format. *Figure 17* shows the frames involved in the iNEMO_Get_AHRS_Library transaction.

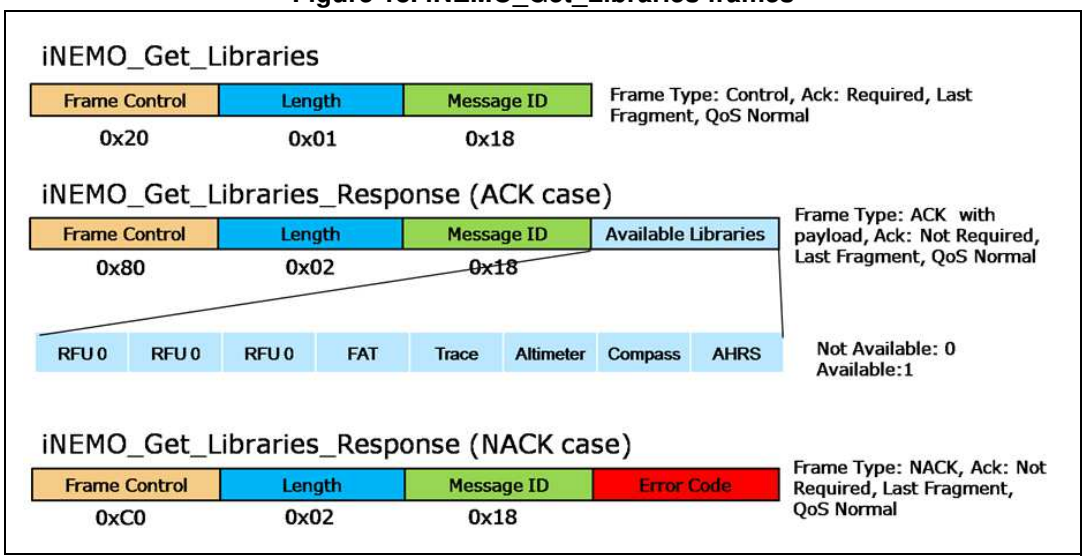
Figure 17. iNEMO_Get_AHRS_Library frames



2.3.6 iNEMO_Get_Libraries

The iNEMO_Get_Libraries command allows knowing which specific libraries are supported by the Discovery-M1 firmware. *Figure 18* shows the frames involved in the iNEMO_Get_Libraries transaction.

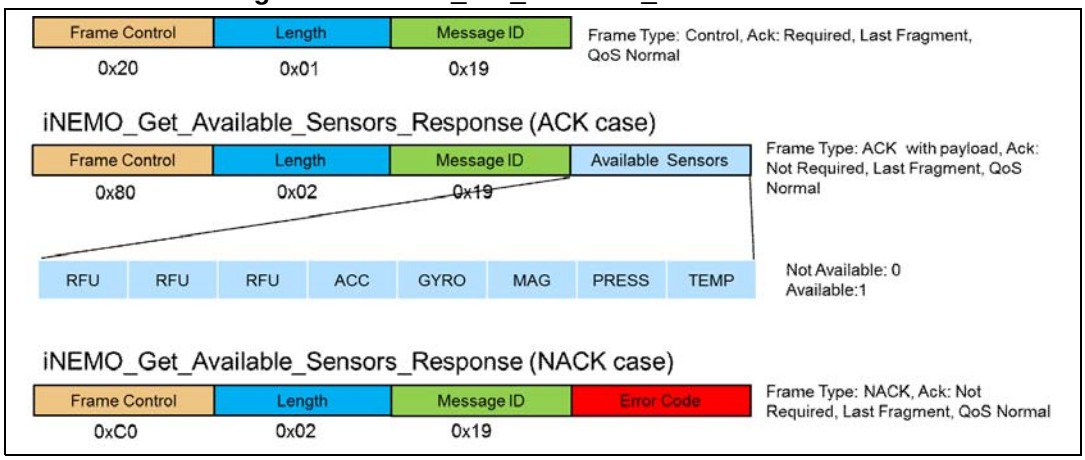
Figure 18. iNEMO_Get_Libraries frames



2.3.7 iNEMO_Get_Available_Sensors

The iNEMO_Get_Available_Sensors command allows knowing which specific sensors are supported by the Discovery-M1 firmware. *Figure 19* shows the frames involved in the iNEMO_Get_Available_Sensors transaction

Figure 19. iNEMO_Get_Available_Sensors frames



2.4 Sensor setting frames

Sensor setting frames are frames originated by the software PC (SDK or GUI) and used to set sensor parameters or to retrieve information about them. All the sensor setting frames are listed in *Table 6*.

Table 6. Sensor setting frames

Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
iNEMO_Set_Sensor_Parameter	CONTROL	Y	0x20	N	variable	Sensor_ Type, Sensor_ Parameter, Parameter_ Value	PC
iNEMO_Set_Sensor_Parameter_ Response	ACK	N	0x20	N	0		Discovery-M1
	NACK	N	0x20	N	1	Error Code	
iNEMO_Get_Sensor_Parameter	CONTROL	Y	0x21	N	2	Sensor_ Type, Sensor_ Parameter,	PC
iNEMO_Get_Sensor_Parameter_ Response	ACK	N	0x21	N	variable	Sensor_ Type, Sensor_ Parameter, Parameter_ Value	Discovery-M1
	NACK	N	0x21	N	1	Error Code	

Table 6. Sensor setting frames (continued)

Commands	Frame Type	Ack required	Message ID	QoS	Payload Length (in bytes)	Payload	Originator
iNEMO_Restore_Default_ Parameter	CONTROL	Y	0x22	N	2	Sensor_ Type, Sensor_ Parameter	PC
iNEMO_Restore_Default_ Parameter_Response	ACK	N	0x22	N	variable	Sensor_ Type, Sensor_ Parameter, Parameter_ Value	Discovery-M1
	NACK	N	0x22	N	1	Error Code	
iNEMO_Save_to_Flash	CONTROL	Y	0x23	N	0		PC
iNEMO_Save_to_Flash_ Response	ACK	N	0x23	N	0		Discovery-M1
	NACK	N	0x23	N	1	Error Code	
iNEMO_Load_from_Flash	CONTROL	Y	0x24	N			PC
iNEMO_Load_from_Flash_ Response	ACK	N	0x24	N			Discovery-M1
	NACK	N	0x24	N	1	Error Code	

2.4.1 iNEMO_Set_Sensor_Parameter

The iNEMO_Set_Sensor_Parameter command allows setting a specific sensor parameter. *Figure 20* shows the frames involved in the iNEMO_Set_Sensor_Parameter transaction.

Figure 20. iNEMO_Set_Sensor_Parameter frames

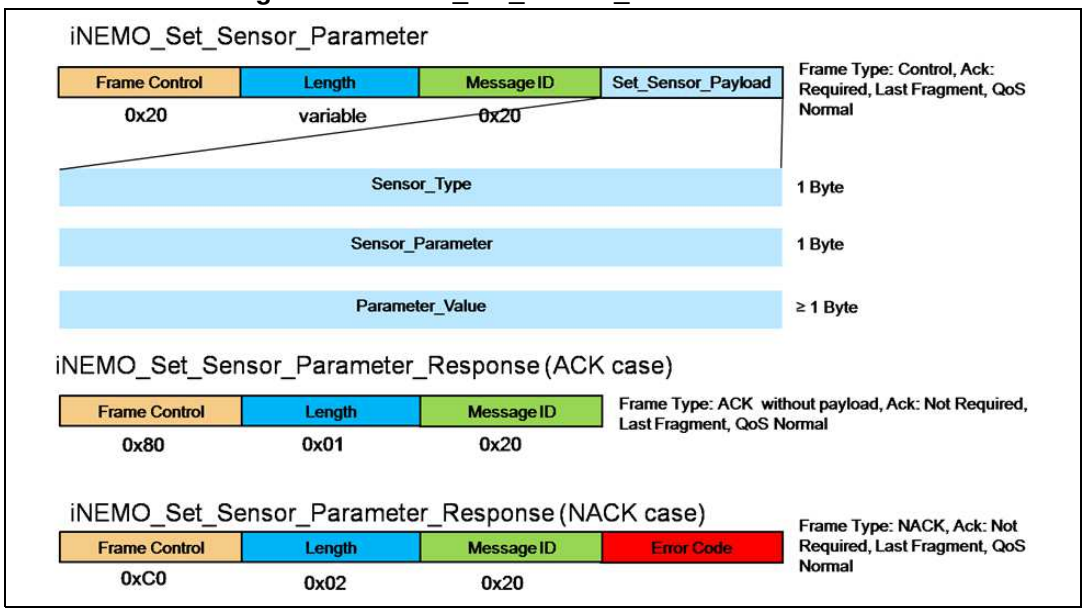


Table 7 describes the "Sensor_Type" field.

2.4.2 iNEMO_Get_Sensor_Parameter

The iNEMO_Get_Sensor_Parameter command allows retrieving from the Discovery-M1 a specific sensor parameter. *Figure 21* shows the frames involved in the iNEMO_Get_Sensor_Parameter transaction.

Figure 21. iNEMO_Get_Sensor_Parameter frames

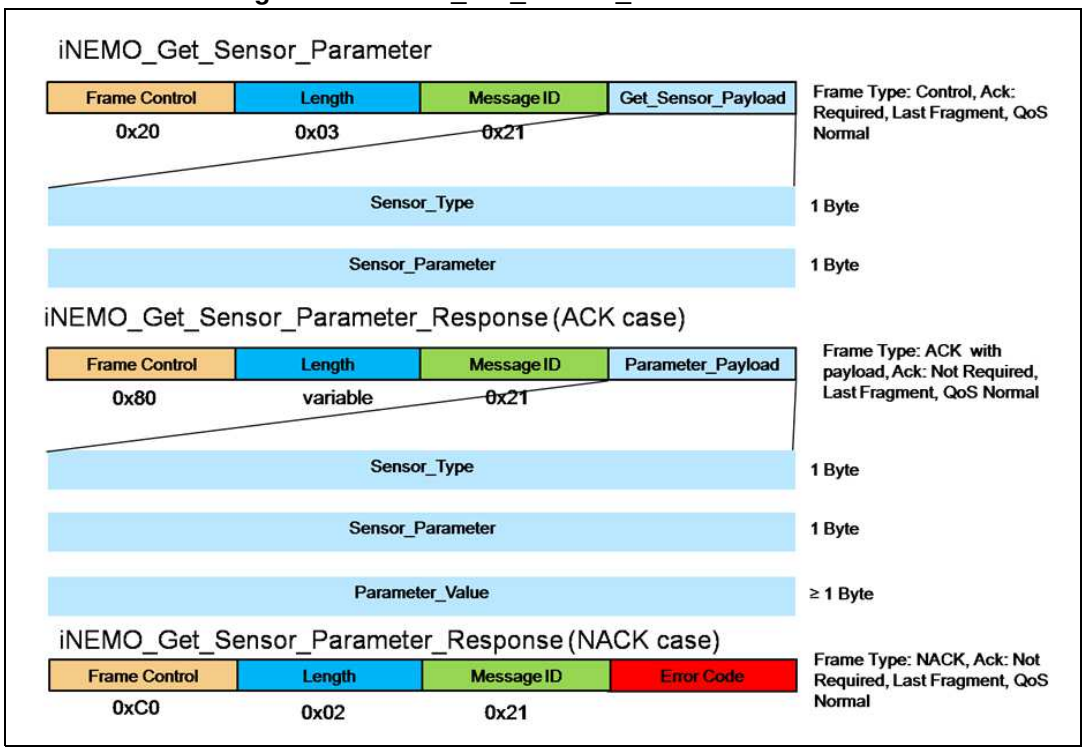


Table 7 describes the "Sensor_Type" field.

2.4.3 iNEMO_Restore_Default_Parameter

The iNEMO_Restore_Default_Parameter command allows restoring a default, specific sensor parameter. *Figure 22* shows the frames involved in the iNEMO_Restore_Default_Parameter transaction.

Figure 22. iNEMO_Restore_Default_Parameter frames

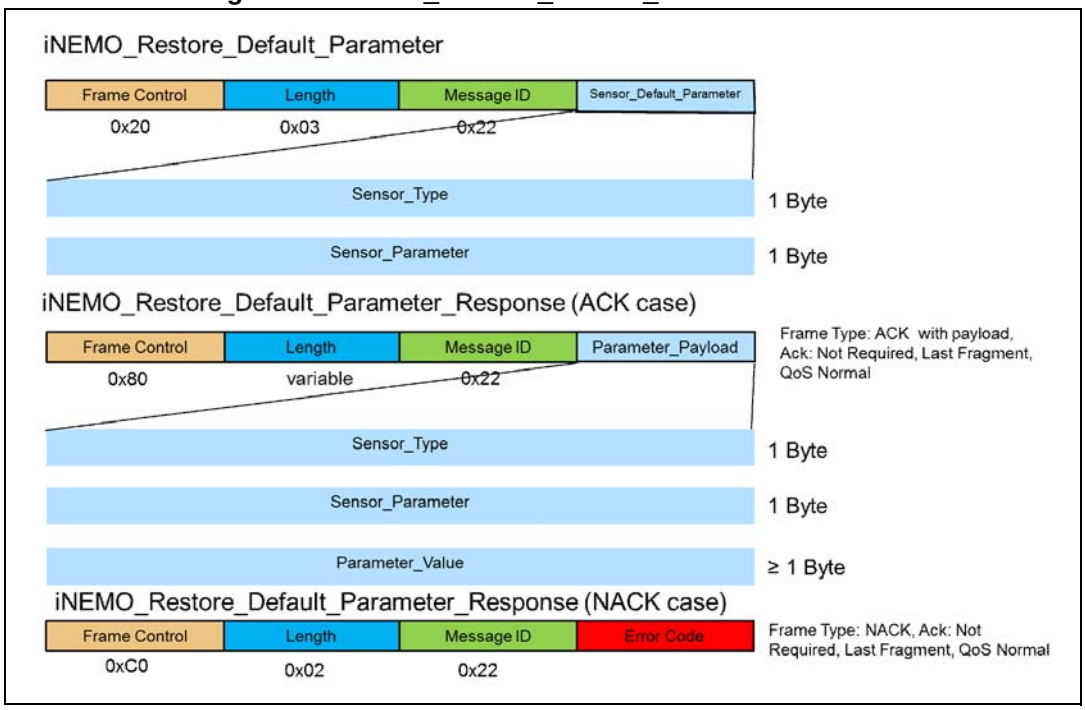


Table 7 describes the "Sensor_Type" list.

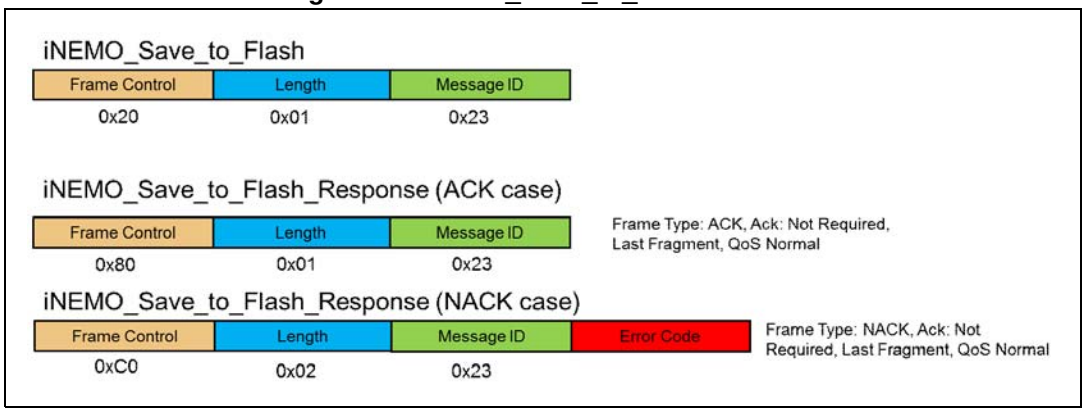
Table 7. Sensor_Type list

Sensor_Type Field	Sensor
0x00	3-axis accelerometer
0x01	3-axis magnetometer
0x02	3-axis gyroscope
0x03	-
0x04	Pressure
0x05	Temperature
0x06 – 0xFF	Reserved for Future Use

2.4.4 iNEMO_Save_to_Flash

The iNEMO_Save_to_Flash command allows storing the settings of the sensor parameters in Discovery-M1 flash. *Figure 23* shows the frames involved in the iNEMO_Save_to_Flash transaction.

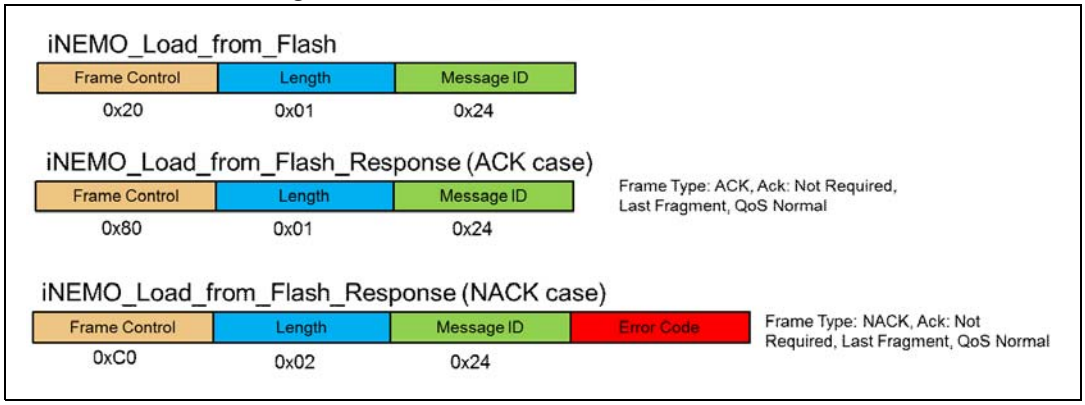
Figure 23. iNEMO_Save_to_Flash frames



2.4.5 iNEMO_Load_from_Flash

The iNEMO_Load_from_Flash command allows loading from Discovery-M1 flash the sensors parameters stored in it. *Figure 24* shows the frames involved in the iNEMO_Load_from_Flash transaction.

Figure 24. iNEMO_Load_from_Flash frames



2.4.6 Accelerometer "Sensor_Parameter" field

Table 8 describes the parameters of the accelerometer sensor and the values of the "Sensor_Parameter" field.

Table 8. Accelerometer Sensor_Parameter list

Sensor_Parameter field	Sensor
0x00	Output data rate
0x01	Full scale
0x02	Acc_HPF
0x03	Offset_X
0x04	Offset_Y
0x05	Offset_Z
0x06	Scale factor X

Table 8. Accelerometer Sensor_Parameter list (continued)

Sensor_Parameter field	Sensor
0x07	Scale factor Y
0x08	Scale factor Z
0xFF	Sensor name (read only)
0x09 – 0xFE	Reserved for Future Use

2.4.7 Accelerometer Output_Data_rate

The "Parameter_Value" field for the output data rate setting is 1 byte in length. *Table 9* describes the supported output data rate for the accelerometer.

Table 9. Accelerometer output data rate field

Parameter_Value field for accelerometer ODR	Output data rate (Hz)
0x00	1
0x01	10
0x02	25
0x03	50
0x04	100
0x05	200
0x06	400
0x07– 0xFF	RFU

2.4.8 Accelerometer full scale

The "Parameter_Value" field for the full-scale setting is 1 byte in length. *Table 10* describes the supported full scale for the accelerometer.

Table 10. Accelerometer full scale field

Parameter_Value field for accelerometer FS	Full scale (g)
0x00	±2g
0x01	±4g
0x02	±8g
0x03	±16g
0x04 – 0xFF	RFU

2.4.9 Accelerometer high-pass filter

The "Parameter_Value" field for the high-pass filter setting is 2 bytes in length as described in *Figure 25*. *Table 11* describes the possible cutoff frequencies.

Figure 25. "Parameter_Value" fields for accelerometer HPF setting

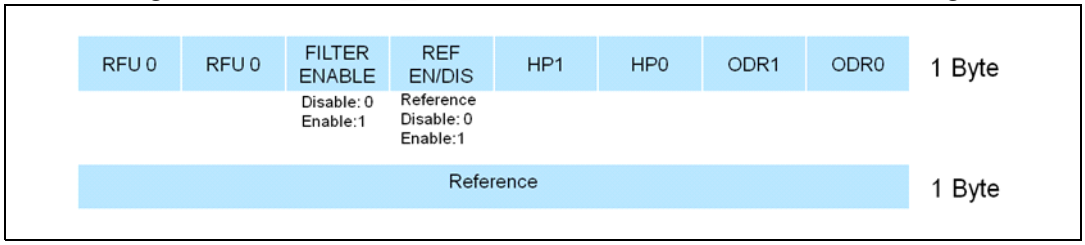


Table 11. Accelerometer high-pass filter setting

HP1	HP0	f_t[HZ] Data rate = 1 Hz	f_t[HZ] Data rate = 10 Hz	f_t[HZ] Data rate = 25 Hz	f_t[HZ] Data rate = 50 Hz	f_t[HZ] Data rate = 100 Hz	f_t[HZ] Data rate = 200 Hz	f_t[HZ] Data rate = 400 Hz
0	0	0.02	0.2	0.052	1.04	2.08	4.16	8.33
0	1	0.01	0.1	0.26	0.52	1.04	2.08	4.16
1	0	0.005	0.05	0.13	0.26	0.52	1.04	2.08
1	1	0.0026	0.026	0.065	0.13	0.26	0.52	1.04

For further details please refer to the LSM303DLHC datasheet .

2.4.10 Accelerometer offset

The "Parameter_Value" field for the offset (X-, Y- or Z-axis) setting is 2 bytes in length and expressed in milli-*g* (thousandth of gravitational force) as signed short (16-bit), with the most significant byte first.

2.4.11 Accelerometer scale factor

The "Parameter_Value" field for the scale factor (X-, Y- or Z-axis) setting is 2 bytes in length (abstract number not mg) as signed short (16-bit) multiplied x1000 and with the most significant byte first. For example, if in the setting view scale factor x is 1.230, it will be multiplied x1000 and sent as 1230 (signed short 16-bit) in this case.

2.4.12 Accelerometer sensor name

The "Parameter_Value" of the sensor name is a read-only field and it returns the name of accelerometer (LSM303DLHC) from Discovery-M1.

2.4.13 Magnetometer "Sensor_Parameter" field

Table 12 describes the parameters of the magnetometer sensor and the values of the "Sensor_Parameter" field.

Table 12. Accelerometer Sensor_Parameter list

Sensor_Parameter field	Parameter
0x00	Output data rate
0x01	Full scale
0x02	Operating mode
0x03	Offset_X
0x04	Offset_Y
0x05	Offset_Z
0x06	Scale factor X
0x07	Scale factor Y
0x08	Scale factor Z
0xFF	Sensor name (read only)
0x09 – 0xFE	RFU

2.4.14 Magnetometer Output_Data_rate

The "Parameter_Value" field for the output data rate setting is 1 byte in length. *Table 13* describes the output data rate supported for the magnetometer.

Table 13. Magnetometer output data rate field

Parameter_Value field for magnetometer ODR	Output data rate (Hz)
0x00	0.75
0x01	1.5
0x02	3
0x03	7.5
0x04	15
0x05	30
0x06	75
0x07	220
0x08 – 0xFF	RFU

2.4.15 Magnetometer full scale

The "Parameter_Value" field for the full-scale setting is 1 byte in length. *Table 14* describes the full scale supported for the magnetometer.

Table 14. Magnetometer full-scale field

Parameter_Value field for magnetometer FS	Full scale (gauss)
0x01	±1.3
0x02	±1.9
0x03	±2.5
0x04	±4.0
0x05	±4.7
0x06	±5.6
0x07	±8.1
0x00, 0x08 – 0xFF	Forbidden-RFU

2.4.16 Magnetometer operating mode

The "Parameter_Value" field for the operating mode setting is 1 byte in length as described in *Figure 26*. *Table 15* describes the possible magnetometer operating modes.

Figure 26. "Parameter_Value" fields for magnetometer operating mode setting

Table 15. Magnetometer operating mode setting

Mode1	Mode0	Magnetic sensor operating mode
0	0	Normal
0	1	Positive bias
1	0	Negative bias
1	1	Forbidden

For further details please refer to the LSM303DLHC datasheet.

2.4.17 Magnetometer offset

The "Parameter_Value" field for the offset (X-, Y- or Z-axis) setting is 2 bytes in length and expressed in milli-gauss (thousandth of gauss) as signed short (16-bit), with the most significant byte first.

2.4.18 Magnetometer scale factor

The "Parameter_Value" field for the scale factor (X-, Y- or Z-axis) setting is 2 bytes in length (abstract number not mgauss) as signed short (16-bit) multiplied x1000.

2.4.19 Magnetometer sensor name

The "Parameter_Value" of sensor name is a read-only field and it returns the name of magnetometer (LSM303DLHC) from Discovery-M1.

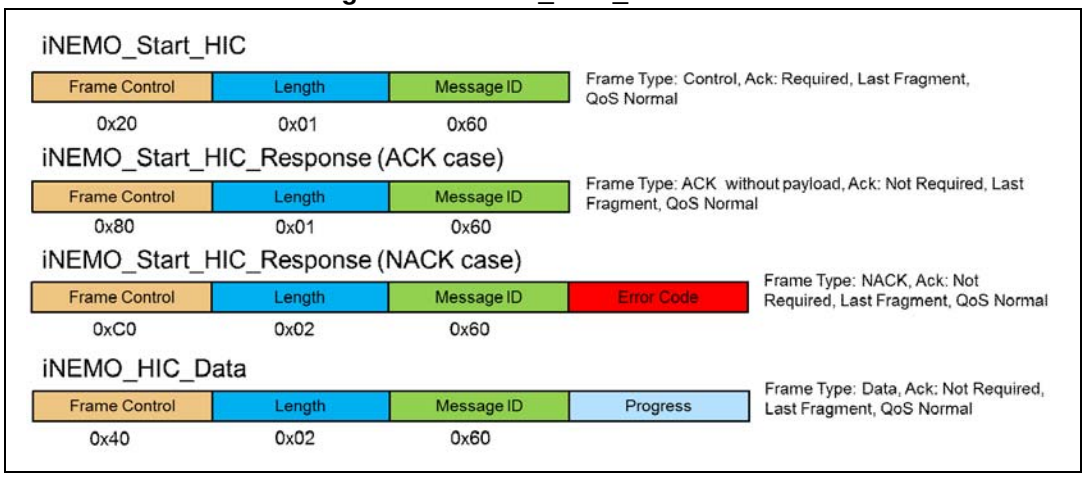
2.4.20 Calibration sensor frames

Calibration sensor frames are frames originated by the software PC (SDK or GUI) used to calibrate sensors. The calibration sensor frames implemented in this version of the protocol are related to the magnetometer hard-iron calibration (HIC) as described below.

2.4.21 iNEMO_Start_HIC

The iNEMO_Start_HIC command is used to start the magnetic sensor HIC procedure. *Table 27* shows the frames involved in the iNEMO_Start_HIC transaction.

Figure 27. iNEMO_Start_HIC frames



2.4.22 iNEMO_Abort_HIC

The iNEMO_Abort_HIC command is used to abort the magnetic sensor HIC procedure. *Table 28* shows the frames involved in the iNEMO_Abort_HIC transaction.

Figure 28. iNEMO_Abort_HIC frames