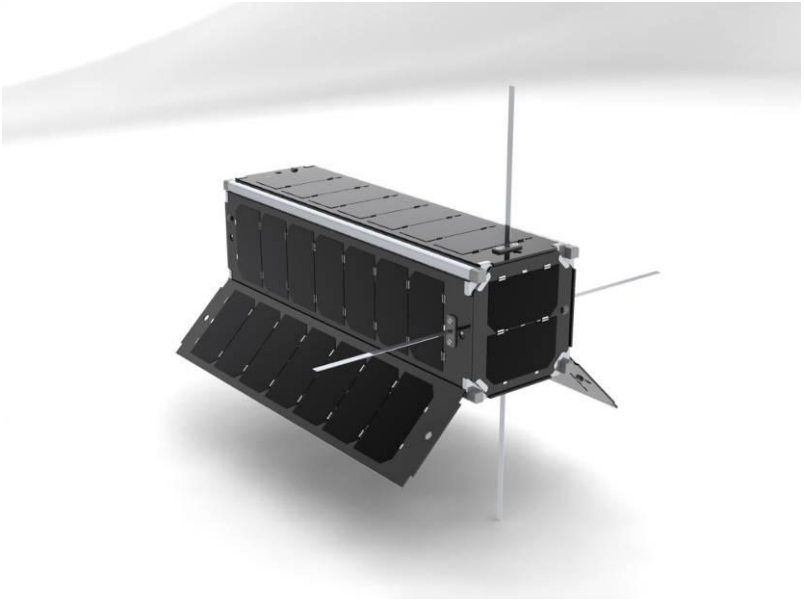
UKube-1 Program

Payload Protocol & Packet Definition

Technical Note 009 / Issue F

Public



Prepared by : *Steve Greenland (KTP)*
Jamie Bowman (Steepest Ascent)
Alan Kane (Clyde Space)

Date : *3rd June 2011*

Checked by :

Date :

1 OVERVIEW

All Commands and Responses between the Payload and Platform shall be routed through a single Payload Controller, implemented within the payload by the Payload Provider. Two data buses are available to the Payload Controller: Payload I2C and Comms SPI. The former is a required interface, and the latter an optional interface.

Commands are divided into two types. Those commands intended to issue instructions or data to the payload are classified as type 1 commands. Those commands intended to read data from the payload are classified as type 2 commands.

To issue a command the platform sends a type 1 command packet to the desired payload as illustrated in Figure 1-1. On receipt of the command the payload should check the CRC of the command packet. If the CRC is correct then the payload should prepare an acknowledge packet (see section 1.1), if the CRC fails or the payload is unable to respond for any other reason the payload should prepare an error packet (an acknowledge packet with the error bit set and an error code in the identifier field). After sending the last byte of the command to the payload the platform will pause for 2ms after which it will read the response.

To read data the platform issues a type 2 command to the desired payload as illustrated in Figure 1-2. On receipt of the command the payload should check the CRC of the command packet. If the CRC is correct then the payload should prepare the requested data packet (see section 1.1), if the CRC fails then the payload should prepare an error (see section 1.1). After sending the last byte of the command to the payload the platform will pause for 2ms after which it will check for a response. Once the response has been read the platform will check the CRC of the response packet. The platform will respond with an acknowledgement packet to the payload. If the CRC check is passed the identifier byte in the response will read 0x7E. If the response CRC check is failed the platform will set the error bit and insert an error code into the identifier field. The payload should wait up to 2ms for this response packet and if it not correctly received it should assume the data has not reached the platform.

When waiting for any response the listener should enforce a 2ms timeout and if no response is received in this time the listener should stop waiting and proceed with an error handling routine.

The Payload Controller shall be compatible with nine Commands, discussed in section 1.2, received via the Payload I2C data bus and, if implemented, respond to data requests via the Comms SPI. Additional Commands may be defined for specific payloads, subject to confirmation by the Platform Provider.

The Payload shall always be a slave on the Payload I2C and Comms SPI data buses and therefore only issue responses when polled by the platform.

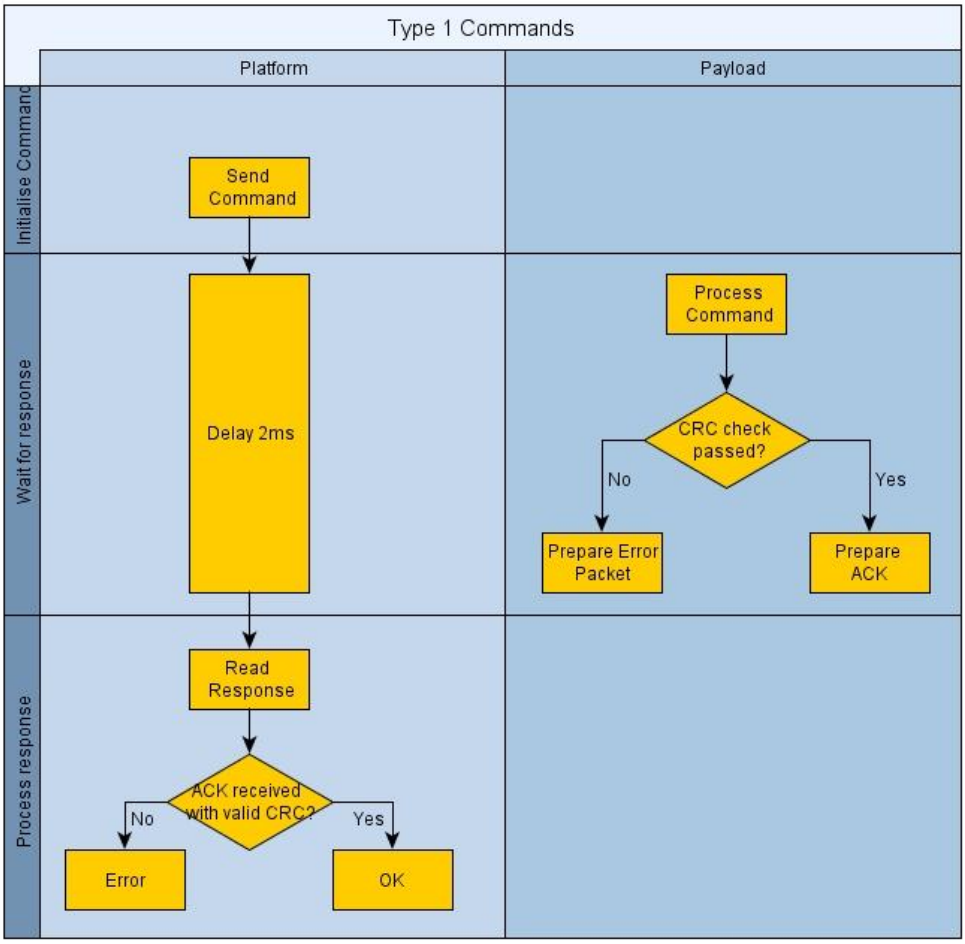


Figure 1-1UKube-1 Type 1 Command Packet Handling

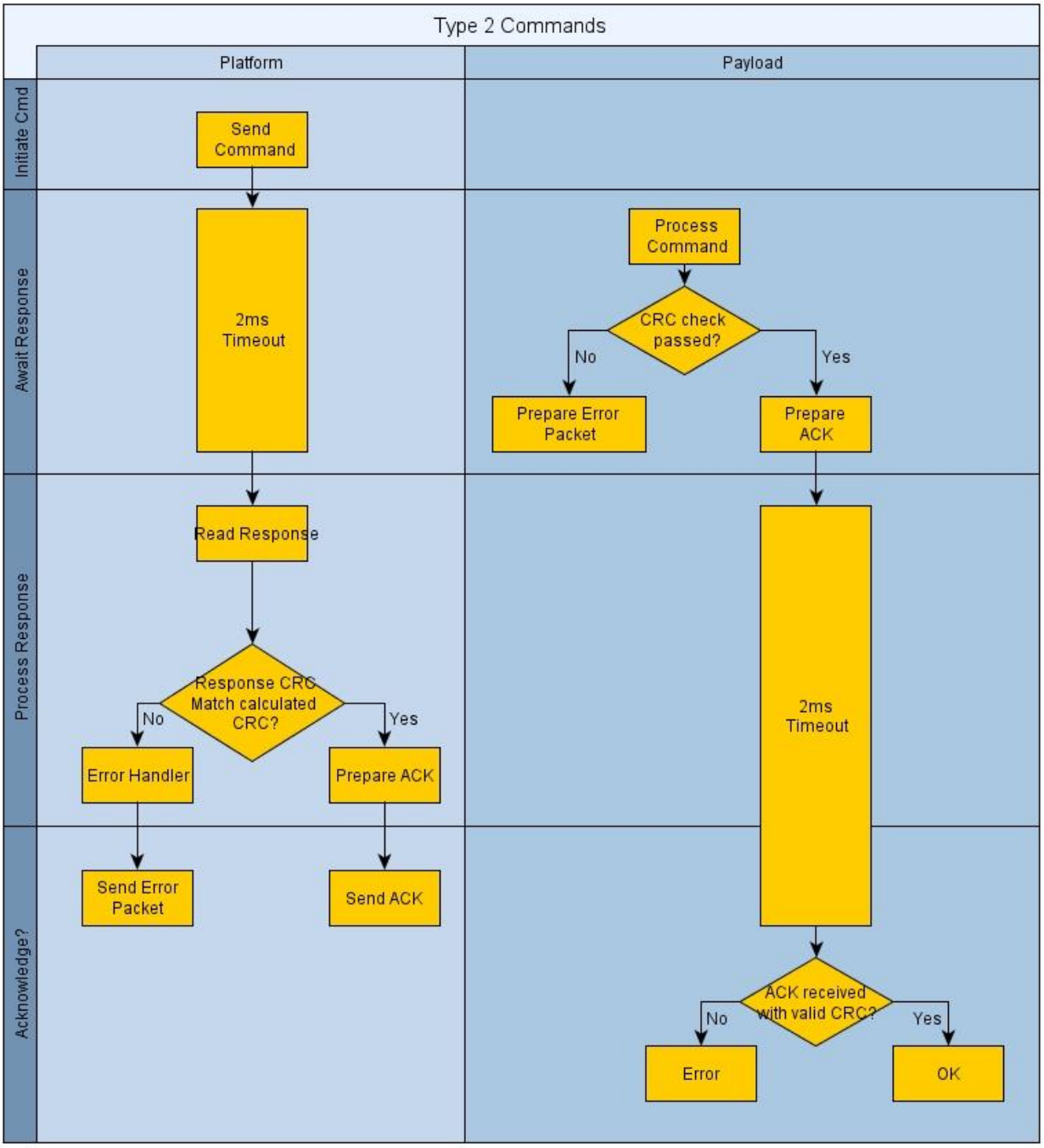


Figure 1-2 UKube Type 2 Packet Handling

1.1 Payload Packet Concept

The UKube-1 standard packet is intended for basic communications with low overhead, but with future compatibility for more complex packet framing. For the Payload, only basic functionality is required. The data packet types are shown in Figure 1-3.

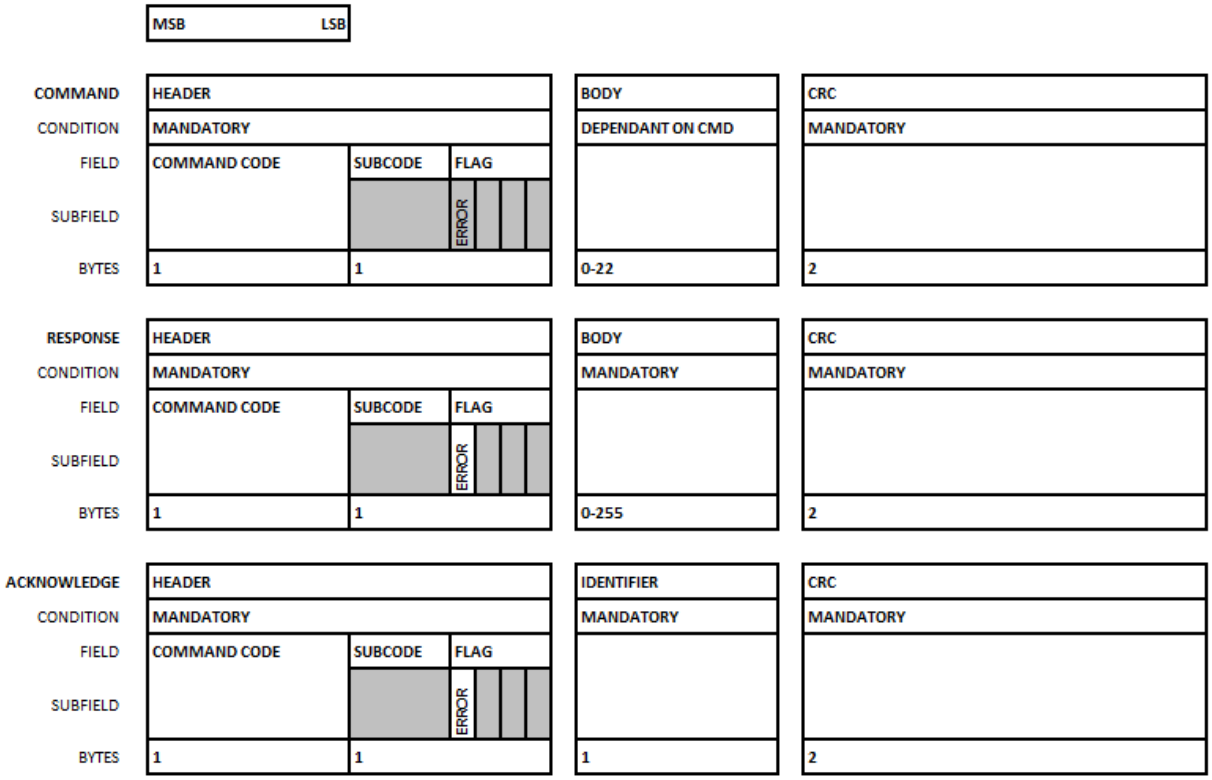


Figure 1-3 UKube Payload Protocol Packet Types

With regard the packet format, the first byte defines the Command Code, and the second byte is reserved for further definition of the packet. The body defines command or response parameters if required. The acknowledge packet contains a fixed code of 0x7E as the identifier byte when indicating a success. An error packet is an acknowledge packet with the error bit set and an error code in the identifier field. Any error codes shall be defined in accordance with the guidance in section 1.5. A CRC check shall be included in all the commands and responses.

All packet lengths are determinate with the lengths being indicated in section 1.7. The maximum amount of payload data within a packet is 256 bytes. Including the additional information within the packet the maximum packet length is 260 bytes. Any packet received by the payload from the Platform will have a maximum length of 24 bytes.

1.2 Payload Command Concept

The Payload Controller shall respond to the following commands on the Payload I2C data bus, further information on the commands can be found in section 1.7.

1. Payload Operation Initialise; to initialise the Payload into a specific operating mode and set operational parameters following switch-on of the payload power lines.
2. Payload Operation Status; to poll the Payload for current status at a frequency of 1-60 s whilst the payload is active.
3. Payload Operation Update; to reinitialise or update the Payload into a specific operation mode, or to update operational parameters.
4. Payload Parameter Write; to set a specific parameter value within the Payload.
5. Payload Parameter Read; to get a specific parameter value from the Payload.
6. Payload Priority Data Receive; to receive priority data from the Payload and transfer to mass memory for downlink.
7. Payload Data Receive; to receive any data from the Payload and transfer to mass memory for downlink.
8. Payload SPI Data Transfer; to prepare the Payload for transferring data via the Comms SPI.
9. Payload Shutdown; to warn the Payload of imminent (+30 s) shutdown of the payload power lines.

These commands will be issued in accordance with the command concept shown in Figure 1-4. Once powered on and initialised by the platform the payload will be polled for a status update at a regular interval (1 to 60 seconds).

The status update message can be used by the payload to request data transfers, mode changes or configuration parameter updates from the platform. A non-zero value in a data waiting or priority data waiting field will trigger a data transfer or priority data transfer operation from the platform. If the Stream ready bit in the payload request flags is set then an SPI transfer will be scheduled instead of an I2C transfer for non-priority data.

A non-zero parameter ID indicates a parameter write is requested to update the value. If any operations are requested by the payload these will be issued in the following I2C transactions. There is no guarantee that all requested operations will be conducted prior to the next status update poll.

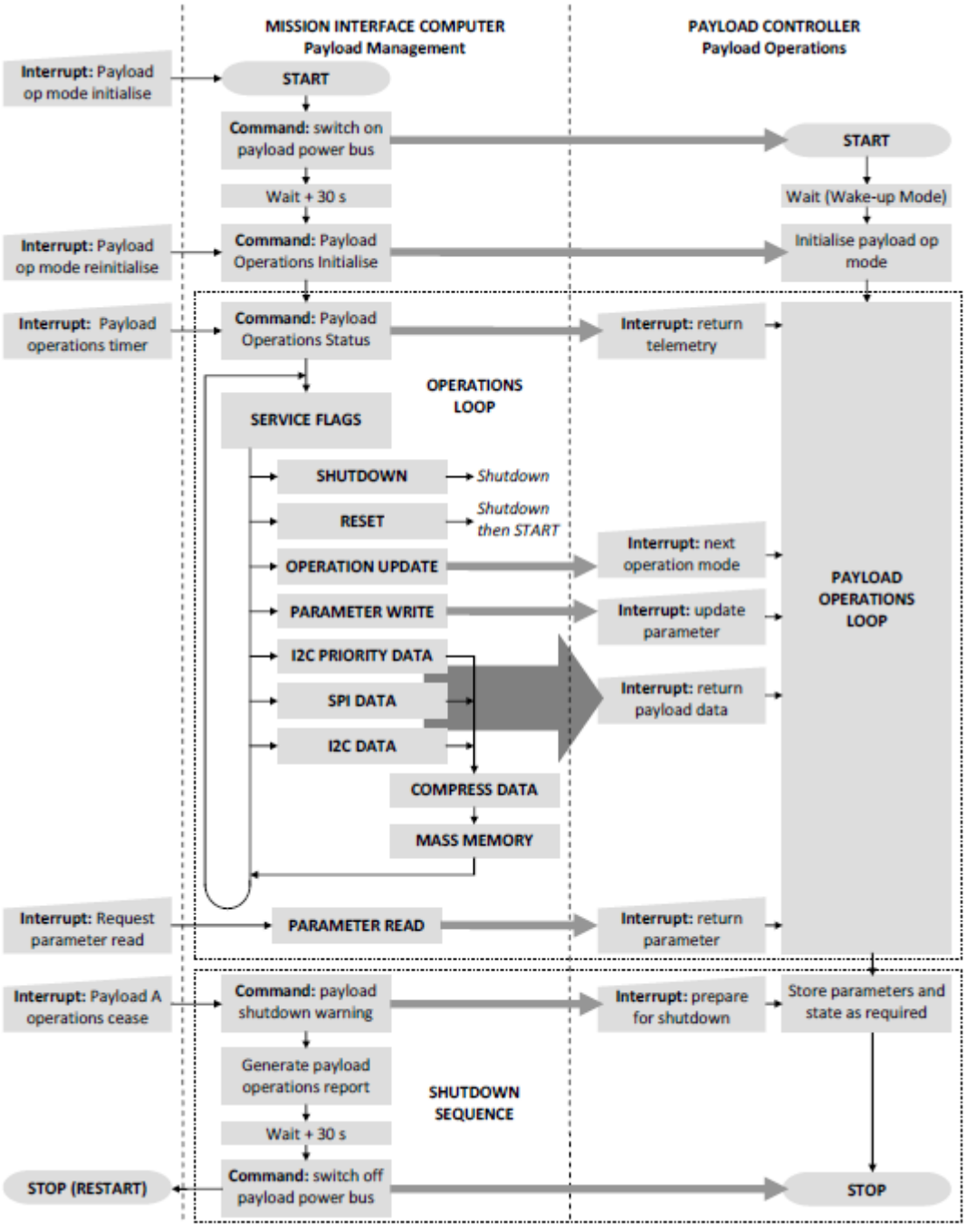


Figure 1-4 Payload Operations Concept

1.3 SPI Implementation

I2C will only ever be used in response to an I2C command initiating the SPI transfer. The packet format discussed will be sent over the SPI bus, with the Slave Select line held in an active position for the full duration of the packet. If data is received with an incorrect CRC check, then the data shall be discarded and an error packet returned.

1.4 I2C Implementation

The packet format discussed will be sent as data over I2C links. As per the I2C standard, the data shall be preceded by the slave device address (7-bits) and a single read/write bit (1-bit). Data received by a slave which is shorter than expected will fail its CRC check and shall be discarded and an error packet will then be returned. Similarly if data is received with an incorrect CRC check, the data shall be discarded and an error packet returned.

If the platform does not receive an acknowledgement or error packet in response to a command it will attempt to resend the command up to three times with a 10 ms delay between attempts. If no response is received after the third attempt the platform will assume the payload has suffered a failure and will power it down.

1.5 Error Handling

When a payload returns an error code action will be taken by the payload based on its error handling module. The actions that can be taken are listed in Table 1-1. Payload designers may request any of these actions be associated with an error code although this is subject to agreement with the platform provider. The default action in the event of an undefined error will be to log the error and shutdown the payload until instructed by telecommand to re-activate it.

The error codes available are in the range 0x01 to 0xFF. 0x7E is reserved for use in the acknowledge packet. The error codes available will be allocated by the platform provider to individual payloads as illustrated in Table 1-1.

Action	Description	Example of use
Log	The error code will be noted in the platform error log for later download to the ground station. No further action will be taken.	Payload asked to capture data when internal memory is full.
Retry	Error code will be noted in the platform error log. The platform will attempt to resend the command up to 3 times with a 10ms delay between attempts. If after 3 attempts no successful response has been received platform will abort and shutdown payload.	Payload indicating a CRC failure to the platform. Attempt to resend the command in case corruption was caused by noise or other temporary event.
Reset	Error code will be noted in the platform error log. The platform will issue a payload shutdown command, delay 30 seconds then cycle power, and re-initialise the payload. When resetting if no acknowledge is received to the shutdown command the payload will continue with the power cycle regardless.	Something has caused the payload to enter an unexpected state.
Shutdown	Error code will be noted in the platform error log. The platform will issue a payload shutdown command, delay 30 seconds switch off power to the payload. When shutting down if no acknowledge is received to the shutdown command the payload will continue with the power cycle regardless.	An unrecoverable error has been detected at the payload.

Table 1-1 Payload Error Handling Actions

Error Code	Description
0x01	CRC check on last received packet failed
0x02	Last command unrecognised
0x03-0x37	Reserved for global error codes
0x37-0x69	Reserved for payload 1 Error codes
0x69-0x7D, 0x7F-0x9B	Reserved for payload 2 error codes
0x9B-0xCD	Reserved for payload 3 error codes
0xCD-0xFF	Reserved for payload 4 error codes

Table 1-2 Error Code Allocations

1.6 Definition

0x00	Indicates hexadecimal values
0b00000000	Indicates binary values
[003 – 004]	Corresponds to bytes 3 to 4 within the packet.

1.7 Packet Definitions

The composition of the packets within the protocol format is illustrated in the following pages.

0x9001 Payload Operations Initialise

Type 1 Command

COMMAND : Payload Operation Initialise
HEADER : 0x9001

DESCRIPTION : Initialisation command for payload to enter

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	Payload Operation Flag	2		16-bit flag	16-bit flag		Payload Provider defined flags, for use with the operation mode for configuring the payload
[004 - 007]	Onboard Time	4		time	0-4.2E6 s | 0-999 ms	1 ms	Onboard time with second synchronised to the synchronisation pulse
[008 - 009]	Payload Priority Data Limit	2		unsigned int	0-16 MB	256 B	Data limit for current operations that will be accepted as Priority Data in units of 256 B
[010 - 011]	Payload Priority Data Remaining	2		unsigned int	0-16 MB	256 B	Remaining data for current operations that will be accepted as Priority Data in units of 256 B
[012 - 015]	Payload Mass Memory Limit	4		unsigned long	0-4096 MB	256 B	Mass memory limit in platform available for use by the Payload
[016 - 019]	Payload Mass Memory Remaining	4		unsigned long	0-4096 MB	256 B	Remaining mass memory in platform available for use by the Payload
[020 - 021]	Payload Poll Frequency	2		unsigned short	1-60000 ms	1 ms	Frequency at which Platform will issue Payload Status command
[022 - 023]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9101 Payload Operation Status

Type 2 Command

COMMAND : Payload Operation Status
HEADER : 0x9101

DESCRIPTION : Poll Payload for current status

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Response Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Payload Operation Mode	1		unsigned char	0x00-0xFF		Payload Provider defined value, describing the operation mode for the payload. 0x00 is always the wake-up mode.
[003 - 004]	Payload Operation Flag	2		16-bit flag	16-bit flag		Payload Provider defined flags, for use with the operation mode for configuring the payload
[005 - 006]	Payload Priority Data Waiting	2		unsigned int	0-16 MB	256 B	Priority Data in Payload waiting for transfer in units of 256 B
[007 - 010]	Payload Data Waiting	4		unsigned long	0-16 MB	256 B	Total data in Payload waiting for transfer
[011 - 011]	Payload Request	1		8-bit flag	8-bit flag		Payload request flags (from MSB: 0 = Payload Update; 1 = Next Mode in mission sequence; 2 = Disable Compression; 3 = Stream Ready; 6 = Reset Payload; 7 = Shutdown)
[012 - 012]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[013 - 013]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[014 - 014]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[015 - 015]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[016 - 016]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[017 - 017]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[018 - 018]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[019 - 019]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[020 - 021]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9201 Payload Operation Update

Type 1 Command

COMMAND : Payload Operation Update
HEADER : 0x9201

DESCRIPTION : Update Payload into operational mode

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Payload Operation Mode	1		unsigned char	0x00-0xFF		Payload Provider defined value, describing the operation mode for the payload. 0x00 is always the wake-up mode.
[003 - 004]	Payload Operation Flag	2		16-bit flag	16-bit flag		Payload Provider defined flags, for use with the operation mode for configuring the payload
[005 - 008]	Onboard Time	4		time	0-4.2E6 s | 0-999 ms	1 ms	Onboard time with second synchronised to the synchronisation pulse
[009 - 010]	Payload Priority Data Limit	2		unsigned int	0-16 MB	256 B	Data limit for current operations that will be accepted as Priority Data in units of 256 B
[011 - 012]	Payload Priority Data Remaining	2		unsigned int	0-16 MB	256 B	Remaining data for current operations that will be accepted as Priority Data in units of 256 B
[013 - 016]	Payload Mass Memory Limit	4		unsigned long	0-4096 MB	256 B	Mass memory limit in platform available for use by the Payload
[017 - 020]	Payload Mass Memory Remaining	4		unsigned long	0-4096 MB	256 B	Remaining mass memory in platform available for use by the Payload
[021 - 022]	Payload Poll Frequency	2		unsigned short	1-60000 ms	1 ms	Frequency at which Platform will issue Payload Status command
[023 - 024]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9301 Payload Parameter Write

Type 1 Command

COMMAND : Payload Parameter Write
HEADER : 0x9301

DESCRIPTION : Provide parameter to the Payload

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[003 - 004]	Payload Parameter Value	2		unsigned short	0x0000-0xFFFF		16 bit parameter value passed to the Payload
[005 - 006]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9401 Payload Parameter Read

Type 2 Command

COMMAND : Payload Parameter Read
HEADER : 0x9401

DESCRIPTION : Provide parameter to the Payload

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Payload Parameter ID	1		unsigned char	0x00-0xFF		Parameter ID for associated parameter value
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Response Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	Payload Parameter Value	2		unsigned short	0x0000-0xFFFF		16 bit parameter value passed to the Payload
[004 - 005]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9501 Payload Priority Data Transfer

Type 2 Command

COMMAND : Payload Priority Data Transfer
HEADER : 0x9501

DESCRIPTION : Transfer of priority data to mass memory

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Response Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 257]	Payload Data	256		user defined	256 * (0x00-0xFF)		User defined Payload Data packet
[258 - 259]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9601 Payload Data Transfer

Type 2 Command

COMMAND : Payload Data Transfer
HEADER : 0x9601

DESCRIPTION : Transfer of any data to mass memory

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Response Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 257]	Payload Data	256		user defined	256 * (0x00-0xFF)		User defined Payload Data packet
[258 - 259]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9701 Payload SPI Data Transfer

Type 2 Command

COMMAND : Payload SPI Data Transfer
HEADER : 0x9701

DESCRIPTION : Prepare Payload for Data Transfer via SPI

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	SPI Transfer Settings	1		unsigned char	0x00-0xFF		Specifies the settings and expected response by the slave next time the slave select is enabled. 0x00 indicates data only, in units of 256 B at 1 Mbps. 0x01 indicates data only, in units of 256 B at 512kbps. 0x02 indicates data only, in units of 256 B at 256kbps.
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Response Parameters							
RESPONSE VIA SPI							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 257]	Payload Data	256		user defined	256 * (0x00-0xFF)		User defined Payload Data packet
[258 - 259]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9F01 Payload Shutdown

Type 1 Command

COMMAND : Payload Shutdown
HEADER : 0x9F01

DESCRIPTION : Prepare for shutdown

Command Parameters		Length	Var	Format	Range	Resolution	Description
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 003]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC
Acknowledge Parameters							
[000 - 000]	CMD Byte	1		unsigned char	0x00-0xFF		Unique code used to identify command
[001 - 001]	Flag Byte	1		unsigned char	0x00-0xFF		Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1		unsigned char	0x00-0xFF		Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2		unsigned short	0x0000-0xFFFF		CCITT 16 bit CRC

0x9XXX Error Packet

May be substituted for the response packet in any Type 2 command transaction.

COMMAND : Acknowledge or Error response packet
HEADER : As per command being responded to

DESCRIPTION : Provide error packet to platform

Response Parameters

[000 - 000]	CMD Byte	1	unsigned char	0x00-0xFF	Unique code used to identify command
[001 - 001]	Flag Byte	1	unsigned char	0x00-0xFF	Bit 0 = CRC used Bit 3 = Error . Remaining Flags used for future expansion.
[002 - 002]	Identifier	1	unsigned char	0x00-0xFF	Error Code or Acknowledge. 0x7E = Acknowledge
[003 - 004]	CRC	2	unsigned short	0x0000-0xFFFF	CCITT 16 bit CRC

1.8 Communication Examples

A series of worked examples are provided to illustrate the communications between the platform and payload for a range of commands under both nominal and error conditions.

1.8.1 Type 1 Command – Payload Initialise

Scenario applicable to Payload Initialise, Payload Operation Update, Payload Parameter Write, Payload Shutdown.

The following example illustrates a nominal transaction.

Initialise - nominal

Platform I2C write	Platform I2C read	Notes
0x90		Command
0x01		Flag Byte
0x01		Desired Operation mode
0x00		Payload Operation Flag MSB
0x00		Payload Operation Flag LSB
0x00		Onboard Time MSB
0x00		Onboard Time
0x00		Onboard Time
0x01		Onboard Time LSB
0xFF		Payload Priority Data Limit MSB
0xFF		Payload Priority Data Limit LSB
0xFF		Payload Priority Data Remaining MSB
0xFF		Payload Priority Data Remaining LSB
0xFF		Payload Mass Memory Data Limit MSB
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit LSB
0xFF		Payload Mass Memory Data Remaining MSB
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining LSB
0x03		Payload Poll Frequency MSB
0xE8		Payload Poll Frequency LSB
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x90	Command
	0x01	Flag Byte
	0x7E	ACK Identifier
	0x	CRC MSB
	0x	CRC LSB

The following example illustrates a transaction where there has been an error at the payload which prevents it responding. In this case the 2ms timeout will expire at the platform and this will trigger an error handler.

Initialise - error at payload

Platform I2C write	Platform I2C read	Notes
0x90		Command
0x01		Flag Byte
0x01		Desired Operation mode
0x00		Payload Operation Flag MSB
0x00		Payload Operation Flag LSB
0x00		Onboard Time MSB
0x00		Onboard Time
0x00		Onboard Time
0x01		Onboard Time LSB
0xFF		Payload Priority Data Limit MSB
0xFF		Payload Priority Data Limit LSB
0xFF		Payload Priority Data Remaining MSB
0xFF		Payload Priority Data Remaining LSB
0xFF		Payload Mass Memory Data Limit MSB
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit LSB
0xFF		Payload Mass Memory Data Remaining MSB
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining LSB
0x03		Payload Poll Frequency MSB
0xE8		Payload Poll Frequency LSB
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
		Platform triggers exception as payload didn't respond

The following example illustrates a transaction where the message received by the payload has been corrupted and therefore the CRC check is failed. The payload will not take any action to the command and will respond with an error packet. Based on the error code received the platform will trigger an error handler as described in section 1.5.

Initialise - received command corrupted

Platform I2C write	Platform I2C read	Notes
0x90		Command
0x01		Flag Byte
0x01		Desired Operation mode
0x00		Payload Operation Flag MSB
0x00		Payload Operation Flag LSB
0x00		Onboard Time MSB
0x00		Onboard Time
0x00		Onboard Time
0x01		Onboard Time LSB
0xFF		Payload Priority Data Limit MSB
0xFF		Payload Priority Data Limit LSB
0xFF		Payload Priority Data Remaining MSB
0xFF		Payload Priority Data Remaining LSB
0xFF		Payload Mass Memory Data Limit MSB
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit LSB
0xFF		Payload Mass Memory Data Remaining MSB
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining LSB
0x03		Payload Poll Frequency MSB
0xE8		Payload Poll Frequency LSB
0x		CRC MSB -doesn't match CRC calculated from message packet
0x		CRC LSB -doesn't match CRC calculated from message packet
2ms delay		to allow payload to prepare response
	0x90	Command
	0x09	Flag Byte
	0x01	Error Code Identifier
	0x	CRC MSB
	0x	CRC LSB

The following example illustrates a transaction where the payload has received the communication successfully and has acted on it however the acknowledgement packet has been corrupted during transmission to the platform. In this case the platform will detect the failed CRC and trigger an error handler. The platform will reissue the command which will result in the payload receiving the same command twice.

Initialise - command received but ACK got corrupted

Platform I2C write	Platform I2C read	Notes
0x90		Command
0x01		Flag Byte
0x01		Desired Operation mode
0x00		Payload Operation Flag MSB
0x00		Payload Operation Flag LSB
0x00		Onboard Time MSB
0x00		Onboard Time
0x00		Onboard Time
0x01		Onboard Time LSB
0xFF		Payload Priority Data Limit MSB
0xFF		Payload Priority Data Limit LSB
0xFF		Payload Priority Data Remaining MSB
0xFF		Payload Priority Data Remaining LSB
0xFF		Payload Mass Memory Data Limit MSB
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit
0xFF		Payload Mass Memory Data Limit LSB
0xFF		Payload Mass Memory Data Remaining MSB
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining
0xFF		Payload Mass Memory Data Remaining LSB
0x03		Payload Poll Frequency MSB
0xE8		Payload Poll Frequency LSB
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x90	Command
	0x09	Flag Byte
	0x7E	ACK Identifier
	0x	CRC MSB
	0x	CRC LSB
		CRC check failed so platform error handler triggered. Platform will resend command up to 3 times. If still no successful acknowledgement will shut down payload.

1.8.2 Type 2 Command – Payload Operations Update

Scenario Applicable to Payload Operation Status, Payload Parameter Read, Payload Priority Data Transfer, Payload Data transfer.

The following example illustrates a nominal transaction.

Status - nominal

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x91	Command
	0x01	Flag Byte
	0x01	Current Operation mode
	0x00	Payload Operation Flag MSB
	0x00	Payload Operation Flag LSB
	0xFF	Payload Priority Data Waiting MSB
	0xFF	Payload Priority Data Waiting LSB
	0xFF	Payload Mass Memory Data Waiting MSB
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting LSB
	0x00	Payload Request
	0x01	Payload Parameter ID (request ID 1 update)
	0x02	Payload Parameter ID (request ID 2 update)
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x	CRC MSB
	0x	CRC LSB
2ms delay		to allow payload to prepare response
0x90		Command
0x01		Flag Byte
0x7E		ACK Identifier
0x		CRC MSB
0x		CRC LSB

The following example illustrates a transaction where there has been an error at the payload which is indicated in an error packet. The code described in the error byte will be used to initiate an exception handler at the platform.

Status - error at payload

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x91	Command
	0x09	Flag Byte
	0x02	Error Msg
	0x	
	0x	CRC LSB
0x90		Command
0x01		Flag Byte
0x7E		ACK Identifier
0x		CRC MSB
0x		CRC LSB

The following example illustrates a transaction where there has been an error at the payload which prevents it responding. In this case the 2ms timeout will expire at the platform and this will trigger an error handler.

Status - error at payload prevents it responding

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
		Platform triggers exception as payload didn't respond

The following example illustrates a transaction where the message received by the payload has been corrupted and therefore the CRC check is failed. The payload will not take any action to the command and respond with an error packet. This will trigger an exception handler at the platform.

Status - received command and it was corrupted

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB -doesn't match CRC calculated from message packet
0x		CRC LSB -doesn't match CRC calculated from message packet
2ms delay		to allow payload to prepare response
	0x91	Command
	0x09	Flag Byte
	0x01	Error Msg
	0x	CRC MSB
	0x	CRC LSB
0x90		Command
0x01		Flag Byte
0x7E		ACK Identifier
0x		CRC MSB
0x		CRC LSB

The following example illustrates a transaction where the platform has received the data packet successfully and has acted on it however the acknowledgement packet has been corrupted during transmission to the payload. In this case the payload will receive a CRC which does not match the data transmission. It is up to the payload designer to decide whether they will retain the data for future transmission or discard it in this scenario.

Status - corruption in ack

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x91	Command
	0x01	Flag Byte
	0x01	Current Operation mode
	0x00	Payload Operation Flag MSB
	0x00	Payload Operation Flag LSB
	0xFF	Payload Priority Data Waiting MSB
	0xFF	Payload Priority Data Waiting LSB
	0xFF	Payload Mass Memory Data Waiting MSB
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting LSB
	0x00	Payload Request
	0x01	Payload Parameter ID (request ID 1 update)
	0x02	Payload Parameter ID (request ID 2 update)
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x	CRC MSB
	0x	CRC LSB
2ms delay		to allow payload to prepare response
0x91		Command
0x01		Flag Byte
0x7E		Identifier Acknowledge
0x		CRC MSB -doesn't match CRC calculated from message packet
0x		CRC LSB -doesn't match CRC calculated from message packet
		Payload decides whether to keep data for re-transmission. Risks duplicating data

The following example illustrates a transaction where the platform has received the data packet successfully and has responded with a data packet. The data packet has then been corrupted during transmission to the platform. In this case the platform will receive a CRC which does not match the data packet and will therefore respond with an error packet. It is up to the payload designer to decide whether they will retain the data for future retransmission or discard it in this scenario.

Status - corruption in data response

Platform I2C write	Platform I2C read	Notes
0x91		Command
0x01		Flag Byte
0x		CRC MSB
0x		CRC LSB
2ms delay		to allow payload to prepare response
	0x91	Command
	0x01	Flag Byte
	0x01	Current Operation mode
	0x00	Payload Operation Flag MSB
	0x00	Payload Operation Flag LSB
	0xFF	Payload Priority Data Waiting MSB
	0xFF	Payload Priority Data Waiting LSB
	0xFF	Payload Mass Memory Data Waiting MSB
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting
	0xFF	Payload Mass Memory Data Waiting LSB
	0x00	Payload Request
	0x01	Payload Parameter ID (request ID 1 update)
	0x02	Payload Parameter ID (request ID 2 update)
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x00	Payload Parameter ID
	0x	CRC MSB -doesn't match CRC calculated from message packet
	0x	CRC LSB -doesn't match CRC calculated from message packet
2ms delay		to allow payload to prepare response
0x91		Command
0x09		Flag Byte
0x01		Error Identifier
0x		CRC MSB
0x		CRC LSB
		Payload decides whether to keep data for re-transmission

1.8.3 SPI Data Tranfer

This example is only applicable to the SPI data transfer. The following illustrates a nominal transaction.

Status - nominal

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB
0x			CRC LSB
	2ms delay		to allow payload to prepare response
		0x97	Command
		0x01	Flag Byte
		0x00	Data MSB
		0x01	Data
		0x02	Data
		0x03	Data
		0x04	Data
		0x05	Data
		0x06	Data
		0x07	Data
		0x08	Data
		0x09	Data
		0x10	Data
		0x11	Data
		0x12	Data
		0x13	Data
		0x14	Data
		0x15	Data
		.	.
		.	.
		.	.
		0x245	Data
		0x246	Data
		0x247	Data
		0x248	Data
		0x249	Data
		0x250	Data
		0x251	Data
		0x252	Data
		0x253	Data
		0x254	Data
		0x255	Data LSB
		0x	CRC MSB
		0x	CRC LSB
	2ms delay		to allow platform to prepare response
0x90			Command
0x01			Flag Byte
0x7E			ACK Identifier
0x			CRC MSB
0x			CRC LSB

The following example illustrates a transaction where there has been an error at the payload which is indicated in an error packet. The code described in the error byte will be used to initiate an exception handler at the platform.

Status - error at payload

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB
0x			CRC LSB
	2ms delay		to allow payload to prepare response
		0x97	Command
		0x09	Flag Byte
		0x00	Error Msg
		0x	CRC MSB
		0x	CRC LSB
0x90			Command
0x01			Flag Byte
0x7E			ACK Identifier
0x			CRC MSB
0x			CRC LSB

The following example illustrates a transaction where there has been an error at the payload which prevents it responding. In this case the 2ms timeout will expire at the platform and this will trigger an error handler.

Status - error at payload prevents it responding

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB
0x			CRC LSB
	2ms delay		to allow payload to prepare response
			Platform triggers exception as payload didn't respond

The following example illustrates a transaction where the message received by the payload has been corrupted and therefore the CRC check is failed. The payload will not take any action to the command and will respond with an error packet. This will trigger an exception handler at the platform.

Status - received command and it was corrupted

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB -doesn't match CRC calculated from message packet
0x			CRC LSB -doesn't match CRC calculated from message packet
	2ms delay		to allow payload to prepare response
		0x97	Command
		0x09	Flag Byte
		0x01	Error Msg
		0x	CRC MSB
		0x	CRC LSB
0x90			Command
0x01			Flag Byte
0x7E			ACK Identifier
0x			CRC MSB
0x			CRC LSB

The following example illustrates a transaction where the payload has received the command packet successfully and has acted on it, returning a successful data packet, however the acknowledgement packet has been corrupted during transmission to the payload. In this case the payload will receive a CRC which does not match the acknowledge transmission. It is up to the payload designer to decide whether they will retain the data for future transmission or discard it in this scenario.

Status - corruption in ack

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB
0x			CRC LSB
	2ms delay		to allow payload to prepare response
		0x97	Command
		0x01	Flag Byte
		0x00	Data MSB
		0x01	Data
		0x02	Data
		0x03	Data
		0x04	Data
		0x05	Data
		0x06	Data
		0x07	Data
		0x08	Data
		0x09	Data
		0x10	Data
		0x11	Data
		0x12	Data
		0x13	Data
		0x14	Data
		0x15	Data
		.	.
		.	.
		.	.
		0x245	Data
		0x246	Data
		0x247	Data
		0x248	Data
		0x249	Data
		0x250	Data
		0x251	Data
		0x252	Data
		0x253	Data
		0x254	Data
		0x255	Data LSB
		0x	CRC MSB
		0x	CRC LSB
	2ms delay		to allow platform to prepare response
0x90			Command
0x01			Flag Byte
0x7E			ACK Identifier
0x			CRC MSB

0x	CRC LSB Payload must decide whether to keep data for re-transmission

The following example illustrates a transaction where the payload has received the command packet successfully and has responded with a data packet. The data packet has then been corrupted during transmission to the platform. In this case the platform will receive a CRC which does not match the data packet and will therefore respond with an error packet. It is up to the payload designer to decide whether they will retain the data for future retransmission or discard it in this scenario.

Status - corruption in data response

Platform I2C write	Platform I2C read	Platform SPI read	Notes
0x97			Command
0x01			Flag Byte
0x00			SPI Transfer Settings
0x			CRC MSB
0x			CRC LSB
	2ms delay		to allow payload to prepare response
		0x97	Command
		0x01	Flag Byte
		0x00	Data MSB
		0x01	Data
		0x02	Data
		0x03	Data
		0x04	Data
		0x05	Data
		0x06	Data
		0x07	Data
		0x08	Data
		0x09	Data
		0x10	Data
		0x11	Data
		0x12	Data
		0x13	Data
		0x14	Data
		0x15	Data
		.	.
		.	.
		.	.
		0x245	Data
		0x246	Data
		0x247	Data
		0x248	Data
		0x249	Data
		0x250	Data
		0x251	Data
		0x252	Data
		0x253	Data
		0x254	Data
		0x255	Data LSB
		0x	CRC MSB -doesn't match CRC calculated from message packet
		0x	CRC LSB -doesn't match CRC calculated from message packet
	2ms delay		to allow platform to prepare response
0x90			Command
0x09			Flag Byte
0x01			Error Identifier to signify CRC fail on data
0x			CRC MSB

0x	CRC LSB Payload must decide whether to keep data for re-transmission