Packet.

Abbreviations:

Abbreviation:	Description:
START_TAG	Indicates start of packet transmission, value is always 0x55.
INFO_BYTE	Provides information abot payload.
RESP	Response (flag).
ENCR	Encryption (flag).
CMD	Command (byte).
PARAM	Parameter (byte).
RESP_CODE	Response code.
DATA	Additional data transmitted in packet.
RESP_DATA	Response data. Same as *DATA*, but used only when packet is received form InputStick and *RESP* flag is set.

Packet structure:

Name:	Size [B]
Header	2
Payload part 1	16
Payload part ...	16
Payload part N (max N = 17)	16

Header:

Header allows to recognize start of data transmission (START_TAG) and provides basic information about following data (*INFO_BYTE*).

Byte:	Description:	Values:
0	*START_TAG*	0x55
1	*INFO_BYTE*	See table

INFO_BYTE:

Bit:	Description	Values	
7	*RESP* - response flag	0	NOT SET
		1	SET
6	*ENCR* - encryption flag	0	NOT SET
		1	SET
5	Reserved		
4 .. 0	Number of payload parts	0x00 - 0x11	

RESP flag:

a) When sending a packet to InputStick:

If *RESP* flag is SET, InputStick will send response to the packet. Response will consist of *CMD*, *RESP_CODE* and optionally *RESP_DATA*.

If *RESP* flag is NOT SET, InputStick will execute the command without providing any kind of response.

b) When receiving a packet from InputStick:

If *RESP* flag is SET it means that the packet is a response to previously sent packet.

If *RESP* flag is NOT SET it means that the packet was sent by InputStick on its own (most likely it is a status update).

Remarks:
DO NOT send any other packet with *RESP* flag SET, until response to previous one is received. You can send any number of packets with *RESP* flag NOT SET.

It is possible that between sending a packet with *RESP* flag SET and receiving response to the packet, you will receive packets with *RESP* flag NOT SET.

Waiting for response decreases data throughput and increases latency. Avoid settings *RESP* flag if possible. Example: when sending HID reports, settings *RESP* flag allows to learn if reports were successfully put in buffer or dropped because of buffer overflow. However, if you keep track of how many reports were sent to InputStick and how many were already received by USB host (and removed from buffer), buffer overflow case will never occur, so there is no point in requesting a response.

ENCRYPTION flag:

When *ENCR* flag is set, payload is encrypted using AES128. For more details see *Security* chapter.

Number of payload parts:

Value of this field must be between N = 1 and 17. Header must be followed by N * 16 bytes.

Payload:

Field:	Description:
CRC32	Cyclic redundancy check value. Calculation starts with CMD byte and ends with last *DATA*, *RESP_DATA* or padding byte.
CMD	Command to execute.
PARAM	Parameter for command. If no parameter is specifdied, use 0x00.
RESP_CODE	Informs wheather *CMD* was successfully executed (0x01) or there was an error encountered
DATA	Additional data transmitted in packet.
RESP_DATA	Response data. Same as *DATA*, but used only when packet is received form InputStick and *RESP* flag is set.
Padding	If total number of bytes in packet is not a multiple of 16, add padding bytes (0x00).

When sending packet to InputStick:

Byte:	Description:
0..3	CRC32
4	*CMD*
5	*PARAM*
6...	*DATA*
	Padding*

When packet is received form InputStick and *RESP* is set:

Byte:	Description:
0..3	CRC32
4	*CMD*
5	*RESP_CODE*
6...	*RESP_DATA*
	Padding*

Remarks:
In this case, *CMD* will be identical to *CMD* from packet for which the response was requested.

When packet is received form InputStick and *RESP* is NOT SET (status update):

Byte:	Description:
0..3	CRC32
4	*CMD*
5...	*DATA*
	Padding*

Remarks:
Note that there is not *RESP_CODE*.