SD Host Controller: Communications Protocol

This SD Host Controller (SDHC) will be using a non-textual binary data packet based protocol to communicate with Host systems. And all the packets transmitted on the interface channel share uniform packet formats.

A communication session is always initialized by a Host system, i.e. ARM processor. The SDHC won't send any data out automatically after powering up.

If a data packet is sent from Host to SDHC, such a packet is called a **Request**. Once the SDHC receives a request, it will reply to the Host system with a data packet called a **Response**.

SDHC will only start performing related operations required by a Host system once after it receives a request. If SDHC should reply to the Host system, it will send one or more required response packets.

Request Packets Format:

All request packets sent by a Host system share the following common format. Little endian byte order is used.

Start Flag	Command	Payload Size	Payload Data	Checksum
1 byte	1 byte	1 byte	0-255 bytes	1 byte

The expected transmission order is from left to right and shouldn't take more than 5 seconds. Not all requests require payloads, such as a STOP command.

A fixed 0xA5 byte is used for each request packet, SDHC uses this byte as the identification of a new request packet. An 8-bit (1-byte) command field must follow the start flag byte.

If the current request carries extra payload data, an 8-bit (1-byte) payload size field is required to be transmitted after sending the command field and then follows the payload data. After the payload data has been transmitted, an 8-bit (1-byte) checksum field calculated from the previous sent data should be transmitted.

The checksum value can be calculated using the following equation:

$$checksum = 0 \oplus 0xA5 \oplus Cmd \oplus PayloadSize \oplus Payload[0] \oplus ... \oplus Payload[n]$$

Note: Timing consideration

All bytes within a request packet must be transmitted to SDHC within 5 seconds.Otherwise, the communication stack of SDHC will discard the current request packet.

Response Packet Format:

All the response packets are divided into two classes: response descriptors and data responses. If the current request received by SDHC requires a response, SDHC will always send a response descriptor packet first and then send one or more data response packets based on the type of requests. Only one response descriptor packet will be sent out during a request/response session. The response descriptors carry the information of the incoming data responses. All the response descriptors share the same format.

The format response descriptors is depicted in the following figure with the transmission order going from left to right:

Start Flag1	Start Flag2	Data Response Length	Send Mode	Data Type
1 byte (0xA5)	1 byte (0x5A)	30 bits	2 bits	1 byte

A response descriptor uses fixed two bytes' pattern 0xA5 0x5A for the host system to identify the start of a response descriptor. The 30 bit Data Response Length field records the size of a single incoming data response packet in bytes. (All the incoming data response packets within a request/response session should have the same format and length). The 2 bits Send Mode field describes the request/response mode of the current session. Its values are listed below:

Send Mode	Description
0x0	Single Request – Single Response mode, SDHC will send only one data response packet in the current session.
0x1	Single Request – Multiple Response mode, SDHC will continuously send out data response packets with the same format in the current session.
0x2	Reserved for future use
0x3	Reserved for future use

The 1-byte Data Type describes the type of the incoming data response packets. It is related to the type of the request SDHC just received. Host systems can choose different data receiving and handling policy based on this field. Different from response descriptors, there is no common format used among response data packets. Each type of response data has its own data format and packet length based on its type.

Requests:

All available requests are listed in the below table with their detailed descriptions given in the following sections:

Request Name	Value	Payload	Response Mode	SDHC Operation
RESET	0x20	NO	No response	Reset SD Host Controller registers and sends to the IDLE state
STOP	0x25	NO		Reserved for future implementation
WRITE	0x4A	YES	Multiple responses	Reserved for future implementation
READ	0x7A	YES		Reserved for future implementation
ERASE	0xF0	YES		Reserved for future implementation
GET_SD_INFO	0x15	NO	Single response	Send out SD card information (e.g. size, name, etc)
GET_CTRL_INFO	0x1B	NO		Reserved for future implementation

RESET (0x20):
Sets software reset until SDHC FSM has acknowledged it by sending back a reset_clear flag. This resets all SDHC registers and sends SD FSM back to the IDLE state.

STOP (0x25):
Reserved for future implementation.

WRITE (0x4A):
Reserved for future implementation.

READ (0x7A):
Reserved for future implementation.

ERASE (0xF0):
Reserved for future implementation.

GET_SD_INFO:
SDHC contains SD cards CID register from initialization. This command simply polls from the CID register and sends all bytes to the Host system. Refer to the datasheet for CID register fields.

GET_CTRL_INFO:
Reserved for future implementation.