Interpic communication

The interpic communication protocol is wrapped by the i2c communication protocol

General format (ignoring i2c start, stop, ack bits) :

<CMD> <SLOTN><BLOCKN>

The commands, sorted by sender, are:

|  |  |  |
| --- | --- | --- |
| Command | Byte Identifier | Description |
| INVALID\_COMMAND | 0xFF | The command is not recognized. |
| RECEIVE\_ERROR | 0xFE | Indicates a problem with the data received. |
| END\_OF\_TRANSMISSION | 0xFD | Indicates the end of the transmission sequence. Used when data transmission may take several instances of data transfer. |
| Front End Commands | | |
| REQUEST\_CARD\_UPDATE | 0x02 | Sent when the front end would like to refresh the ids of the cards present in its memory. Usually sent as a result of receiving notice that the cards have changed on the board |
| REQUEST\_CARD\_DATA | 0x04 | Command specifying the card slot and block |
| WRITE\_CARD\_BLOCK | 0xFC | Instruction to write 32 bits to a specified block and card slot |
| WRITE\_AFI | 0xFB | Not currently implemented |
| WRITE\_DSFID | 0xFA | Not currently implemented |
| Back End Commands | | |
| CARD\_CHANGE | 0x01 | Sent to indicate that one or more cards have changed on the board |
| CARD\_UID | 0x02 | Response to REQUEST\_CARD\_UPDATE. Sends the card UID and corresponding board location |
| CARD\_DATA\_BLOCK | 0x04 | Response to REQUEST\_CARD\_DATA. Sends the data from the specified block in the specified slot. |
|  |  |  |
|  |  |  |
|  |  |  |

Note that several of the byte codes are the same. The protocol allows for this because the front end will only receive the commands sent from the backend. The codes are still completely unique.