**K2 Senate Audio Control Panel**

**BeagleBone / Control Board Communications**

Version 0.5, 5/16/2017

**Message Format**

The following message format is used in both directions.

|  |  |  |
| --- | --- | --- |
| Byte # | Function | Notes |
| 0 | start char | 0xe8 |
| 1 | length | # of bytes that follow excluding checksum & stop char |
| 2 | command/status |  |
| 3 | parameter(s) | set to 0 if not used |
| 3 + length | checksum |  |
| 4 + length | stop char | 0xee |

All command messages are responded to with a status message. The status message may be an ACK, return data for a “get” command or an error message.

**Command Messages (to the control board)**

Command messages fall within 4 ranges:

* 0x20 – 0x2f = configuration commands
* 0x30 – 0x3f = status request commands
* 0x40 – 0x4f = button LED related commands
* 0x50 – 0x5f = encoder related commands

For consistency, the LSB of the command byte is ‘0’ for “set” commands and ‘1’ for “get” commands.

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Hex # | Parameter(s) | Notes |
| Set button LED slow rate | 0x20 | 1 – 10 | Rate in hertz |
| Get button LED slow rate | 0x21 | N/A |  |
| Set button LED fast rate | 0x22 | 1 – 10 | Rate in hertz |
| Get button LED fast rate | 0x23 | N/A |  |
| Set button LED slow duty cycle | 0x24 | 0 - 100 | On-time percentage of period |
| Get button LED slow duty cycle | 0x25 | N/A |  |
| Set button LED fast duty cycle | 0x26 | 0 - 100 | On-time percentage of period |
| Get button LED fast duty cycle | 0x27 | N/A |  |
| Set encoder sensitivity | 0x28 | 1 - 10 | (what does this really mean??) |
| Get encoder sensitivity | 0x29 | N/A |  |
| Set LED PWM duty cycle | 0x2a | 1 - 100 | On-time percentage of period |
| Set LED PWM period | 0x2b | Range TBD | Period in increments of 100Hz |
| Get panel status | 0x31 | N/A |  |
| Get firmware version | 0x33 | N/A |  |
| Set button LED behavior | 0x40 | 1 = action  2 = button # | See list of button actions below  See list of button numbers below |
| Get button LED behavior | 0x41 | button # |  |
| Send button LED list | 0x42 | 1-16 = button # | Array for 16 button numbers, will terminate at first 0 |
| Execute button LED list behavior | 0x44 | action | Perform action on LED list(s) |
| Set encoder display value | 0x50 | 0 – 100 | How to convert to 16 LEDs on/off |
| Get encoder display value | 0x51 | N/A |  |
| Set encoder position | 0x52 | 0 – 100 |  |
| Get encoder position | 0x53 | N/A |  |
|  |  |  |  |

Button numbers:

* 0x01 – 0xcb = individual buttons
* 0xf0 = semicircle group
* 0xf1 = center group
* 0xf2 = upper-right group
* 0xf3 = upper-left group
* 0xf8 = all buttons

Button actions:

* 0 = Off
* 1 = Solid Green
* 2 = Slow Flash Green
* 3 = Fast Flash Green
* 4 = Solid Red
* 5 = Slow Flash Red
* 6 = Fast Flash Red
* 7 = Slow alternate Red / Green
* 8 = Fast alternate Red / Green

**Status & Exception Messages (from the control board)**

These message codes fall within 3 ranges

* 0 – 0x1f = report normal user activity
* 0x20 – 0x5f = response to command requesting information
* 0x80 – 0xff = message ACK and exception/error reporting

For responses to commands requesting information, the status code is the same as the command code being responded to.

|  |  |  |  |
| --- | --- | --- | --- |
| Status/exception | Hex # | Parameter(s) | Notes |
| Switch action | 0x10 | 1 = button #  2 = on/off | Sent on change of button state, may happen at any time. |
| Encoder position changed | 0x11 | 0 – 100 | New encoder position |
| Get button LED slow rate | 0x21 | 1 – 10 | Rate in hertz |
| Get button LED fast rate | 0x23 | 1 – 10 | Rate in hertz |
| Get button LED slow duty cycle | 0x25 | 0 - 100 | On-time percentage of period |
| Get button LED fast duty cycle | 0x27 | 0 - 100 | On-time percentage of period |
| Get encoder sensitivity | 0x29 | 1 - 10 | (what does this really mean??) |
| Get panel status | 0x31 | Panel status | See definitions below |
| Get firmware version | 0x33 | 3 bytes |  |
| Get button LED behavior | 0x41 | 1 = action  2 = button # | See list of button actions below  See list of button numbers below |
| Get encoder display value | 0x51 | 0 – 100 |  |
| Get encoder position | 0x53 | 0 – 100 |  |
| ACK | 0x80 | N/A | Command executed successfully |
| Exception | 0x90 | Exception code | Used to report HW issues (low voltage, button stuck on, etc.), may happen at any time. |
| Error | 0xf0 | Error code | Used to report error response to command |

The BeagleBone will respond to unsolicited messages from the control board micro with an ACK or error message so the control board micro may re-send the message. If the BeagleBone has a timeout or error while receiving a solicited message it will re-send the command message.

**Panel status**

The Panel Status byte is bit oriented as defined below.

|  |  |
| --- | --- |
| Bit # | Meaning |
| 7 | 1 = Reset occurred since last command |
| 6 | Reserved, set to ‘0’ |
| 5 | Reserved, set to ‘0’ |
| 4 | Reserved, set to ‘0’ |
| 3 | Reserved, set to ‘0’ |
| 2 | 1 = Error reported |
| 1 | 1 = Exception reported |
| 0 | 1 = Ready to receive commands |

All bits except bit 0 will be cleared after the response is sent for a “Get Panel Status” command.

**Exception codes**

Below are the possible exception codes reported with the Exception message.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Exception code | Hex # | Parameter(s) | Severity | Notes |
| Power-up reset occurred | 0x10 | N/A | See notes | Low severity for normal power-up, high severity if during normal operation |
| Other reset occurred | 0x11 | Reset source | High | Define based on micro reset source bits |
| High/low 3.3V | 0x21 | A/D value | High | Outside of valid range (nominal +/- 5%) |
| High/low 5V | 0x22 | A/D value | Med | Outside of valid range (nominal +/- 5%) |
| High/low 12V | 0x24 | A/D value | High | Outside of valid range (nominal +/- 10%) |
| Button stuck on | 0x30 | Button # | High | Button on longer than TBD seconds |

Note: any combination of the High/low voltage exceptions code may be set at the same time so 0x23, 0x25, 0x26 and 0x27 are valid exception codes calling out multiple voltage problems.

**Error codes**

Below are the possible error codes reported with the Error message. Error codes below 0x80 are indicative of a properly formatted message with incorrect contents. Error codes above 0x80 are indicative of badly formatted messages, dropped characters, UART errors, etc.

|  |  |  |
| --- | --- | --- |
| Error code | Hex # | Meaning |
| Invalid command | 0x11 | Invalid command code |
| Incorrect length | 0x12 | Length byte in message isn’t valid for the command type |
| Invalid parameter | 0x13 | One or more parameters were invalid for the command |
| RX misc error | 0x81 | UART line status register reported error |
| Bad START character | 0x82 | Incorrect message START character |
| Invalid length | 0x83 | Message length exceeds maximum message length |
| Bad checksum | 0x84 | Message checksum doesn’t match calculated checksum |
| Bad STOP character | 0x85 | Incorrect message STOP character |
| RX timeout | 0x86 | UART reported receive timeout |
| RX buffer overflow | 0x87 | Message exceeded maximum message length |
| RX message timeout | 0x88 | started to receive a message but didn't receive complete message |

**RTS/CTS Handshake**

The RTS/CTS handshake is intended to prevent receiving unsolicited messages while waiting for the response from a previously transmitted message. The BeagleBone will normally drive the CTS signal to the control board micros in-active (low) and the control board micros will normally drive their RTS signals to the BeagleBone in-active (low).

The control board micros will assume they can always respond to a command since the response is solicited. Before sending an unsolicited message, the control board micro will set its RTS signal high and wait for its CTS signal to go active before sending the message. If the BeagleBone has started a command/response exchange with that particular micro, it will hold CTS inactive until it receives the response and then set CTS active. If a command/response exchange with that micro is not in progress, the BeagleBone will set CTS active and not start a command/response exchange until it has received the unsolicited message and set CTS inactive.

Revisions:

Version 0.1:

* First version for review/comment within DGCD

Version 0.2:

* Changed “Execute button LED list behavior” command from 0x43 to 0x44 to be consistent with the other commands (LSB indicating “set” or “get”).
* Changed the start and stop characters to values outside the range of valid button numbers.

Version 0.3:

* Added “Set LED brightness” command
* Started exception code list
* Started error code list

Version 0.4:

* Changed note for length field in message format description to clarify that it does not include the stop character
* Added “Set LED PWM period” command and changed “Set LED brightness” command to be on-time percentage of the period (duty cycle). This allows complete control of PWM dimming from the BeagleBone.

Version 0.5:

* Redefined exception codes to allow for reporting multiple voltage problems with a single exception message
* Added definitions for the “get panel status” response status byte.
* Cleanup of the Parameters column in the table in the Status & Exception Messages section.
* Changed LED action #7 to “slow alternate red/green and added “fast alternate red/green” as action #8.