# Senate Chamber Operator’s Hardware Panel

# Communication Protocol between SBC on Panel and QSys Processor

V. 1.0

February 2, 2017

## Description:

The Senate Chamber Operator’s Panel has a small computer that provides an interface to the QSys Audio DSP system for interaction with the panel buttons and LEDs. There are a series of actions that are required for control and feedback in both directions: from panel to DSP and from DSP to panel. This document describes a text based protocol for commands and responses between the two devices to be used for script coding in QSys.

## Communication Process:

The SBC will behave as a TCP server accepting connections from a TCP client on the QSys Core using port 8003. The communication between the two devices will be over Ethernet using raw TCP (no Telnet negotiation).

## Basic Command and Response Structure:

### Commands:

Commands are sent from the client (DSP) to the server (SBC) and follow this format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CATEGORY  UP TO 3 CHAR | COMPONENT TYPE  UP TO 3 CHAR | COMPONENT NUMBER/ID  UP TO 3 CHAR | ACTION  UP TO 3 CHAR  “SET” or “GET” | STATE/VALUE  UP TO 4 CHAR | END OF LINE  1 CHARACTER  ALWAYS 0x0D |

For example:

“BTN LED 1 SET 1”,0x0D // For setting the LED of button #1 to a value of 1

Presented in more JSON typical format

{

“category”: “BTN”,

“component”: “LED”,

“component\_id”: “1”,

“action”: “SET”,

“value”: “1”,

} 0x0D

When there is a set of LEDs that need to be changed, the request just holds an array in component\_id:

{

“category”: “BTN”,

“component”: “LED”,

“component\_id”: [“1”,”3”,”55”,”65”],

“action”: “SET”,

“value”: “1”,

} 0x0D

There are four categories that define the type of object that a command refers to and they include CFG, STS, BTN and RTR.

### Responses:

The SBC will respond to every command received. The format of the response is similar to a command as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CATEGORY  UP TO 3 CHAR | COMPONENT TYPE  UP TO 3 CHAR | COMPONENT NUMBER/ID  UP TO 3 CHAR | ACTION  UP TO 3 CHAR  “=” | STATE/VALUE  UP TO 4 CHAR | END OF LINE  1 CHARACTER  ALWAYS 0x0D |

For example:

“BTN LED 1 = 1”,0x0D // Returns the updated value of the LED for button #1 as 1

Response like this follows same rules:

{

“category”: “BTN”,

“component”: “LED”,

“component\_id”: “1”,

“action”: “=”,

“value”: “1”,

} 0x0D

In case a command sent to the SBC is not understood, the SBC should return an error response. An error response will follow the format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CATEGORY | COMPONENT TYPE | COMPONENT NUMBER/ID | ACTION | STATE/VALUE | END OF LINE |
| ERROR |  |  | = | 1 | 0x0D |

Error response in json format:

{

“category”: “ERROR”,

“component”: “”,

“component\_id”: “”,

“action”: “=”,

“value”: “1”,

} 0x0D

Optional description in error response:

{

“category”: “BTN”,

“component”: “”,

“component\_id”: “”,

“action”: “=”,

“value”: “1”,

“description”: “Invalid category or component”,

} 0x0D

Where the error state defines the type of error encountered and can be one of the following:

|  |  |
| --- | --- |
| Error State (Code) | Error Type |
| 1 | Invalid category or component. |
| 2 | State (parameter) out of range. |
| 3 | Command not understood / syntax invalid. |

### Unsolicited responses from SBC:

The SBC will send out unsolicited responses whenever the state of a hardware component on the panel changes. For example, during a button press or when the rotary encoder is changed. Unsolicited responses will follow the same format as standard responses shown above.

### Session Initiation:

Upon initial connection, the SBC shall reply with a panel status response to begin communications.

### Complete List of Commands:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Category | Type | ID | Available Actions | Range of States/Values |
| Configuration of Button LED Behavior:  **Slow Rate Setting (in Hertz)** | CFG | RTE | SLO | SET, GET, = | 1 – 10 |
| Configuration of Button LED Behavior:  **Fast Rate Setting (in Hertz)** | CFG | RTE | FST | SET, GET, = | 1 – 10 |
| Configuration of Button LED Behavior:  **Slow Duty Cycle Setting** | CFG | CYC | SLO | SET, GET, = | 0 – 100 |
| Configuration of Button LED Behavior:  **Fast Duty Cycle Setting** | CFG | CYC | FST | SET, GET, = | 0 – 100 |
| Configuration of Encoder Sensitivity:  **Encoder Sensitivity Setting** | CFG | ENC | SEN | SET, GET, = | 1 – 10 |
| Status Request for Board:  **Panel Status** | STS | SYS | STS | GET, = | 0 = Unknown  1 = OK  2+ = Error Code |
| Status Request for Board:  **Firmware Version** | STS | SYS | FW | GET, = | 3 character designator |
| Button Action:  **LED Behavior** | BTN | LED | # identifier for button | SET, GET, = | 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 = Flash Red / Green |
| Button Action:  **LED Behavior (Apply to All Semicircle Buttons)** | BTN | LED | ALL | SET, = | 0 – 7 |
| Button Action:  **LED Behavior (Apply to Array)** | BTN | LED | #,#,#.... | SET,= | 0 – 7 |
| Button Action:  **Switch Action (unsolicited response only)** | BTN | SW | # identifier for button | = | 1 = Button Pressed  0 = Button Released |
| Encoder Action:  **Encoder Display Value** | ENC | DIS | 0 | SET, GET, = | 0 – 100 |
| Encoder Action:  **Rotary Encoder Position** | ENC | POS | 0 | SET, GET, = | 0 – 100 |