

# BeMicro SDK

## System Console Demo

# Overview

- After building the hardware design that is released with this demo, you can start the System Console demo.
- The System Console demo provides dashboards for each of the components in the demo Qsys system.
- Each of the dashboards is explained briefly in the following slides.

# boardInit

- The boardInit dashboard coordinates the project files required to configure the FPGA device and initialize the System Console environment for the rest of the dashboards.
- To initialize the environment, this dashboard will configure the FPGA device, load the JDI file into the console and load the Qsys header module so the other dashboards can reference it.

The screenshot displays the 'boardInit' dashboard interface. It features three sections for file selection under the heading 'Project Files':

- SOF File:** A text input field containing './test\_project.sof' and a 'Choose SOF File' button.
- JDI File:** A text input field containing './test\_project.jdi' and a 'Choose JDI File' button.
- Qsys Header File:** A text input field containing 'headers/test\_sys\_top\_qsys.tcl' and a 'Choose Header File' button.

Below these sections is a 'Status' area with two indicators: a green circle next to 'Initialized' and a red circle next to 'Errors'. To the right of the status indicators is an 'Initialize' button.

# System ID

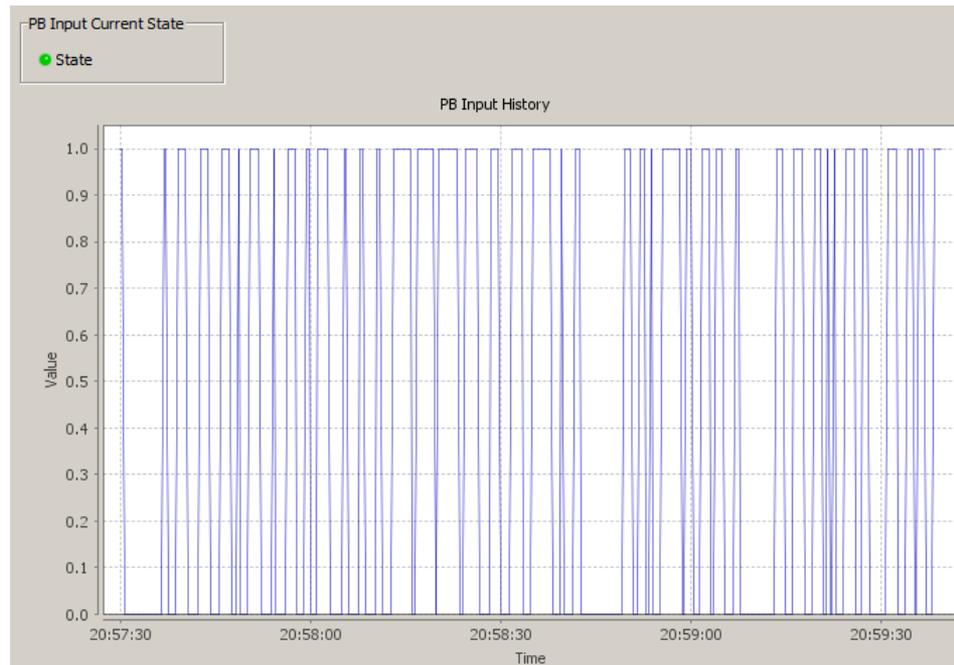
- The systemID dashboard is used to validate the system ID peripheral in the Qsys system.
- Once this dashboard successfully validates the system ID peripheral, all of the other dashboards can assume that the proper FPGA design is loaded and available in the environment.

The screenshot displays a dashboard for system validation. It is organized into several sections:

- System ID:** Contains two input fields. The 'Expected ID' field contains the hexadecimal value '0xfacecafe', and the 'Actual ID' field also contains '0xfacecafe'.
- System Timestamp:** Contains two input fields. The 'Expected TS' field contains '0x4f52c69a', and the 'Actual TS' field also contains '0x4f52c69a'.
- Base Address:** A single input field containing the hexadecimal value '0x00000440'.
- Status:** A section containing a green indicator light next to the text 'Valid', and an 'Initialize' button.

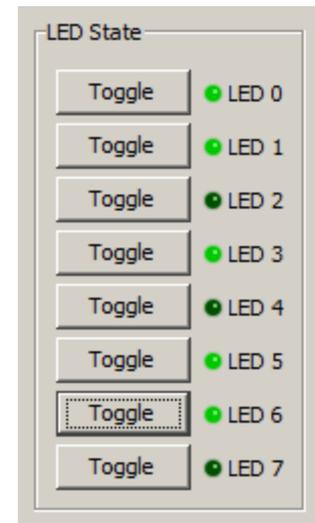
# Push Button PIO

- The pbPIO dashboard provides an LED widget to track the live state of the push button as well as a time chart widget to track the history of the push button.



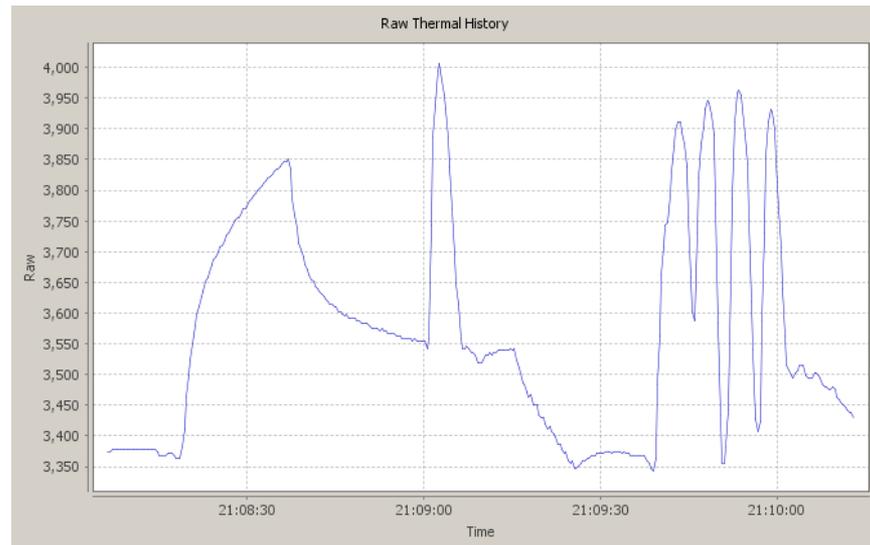
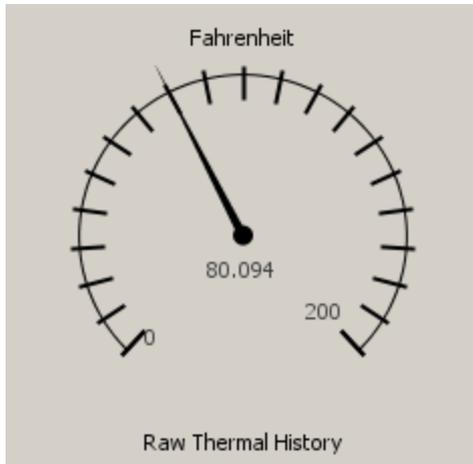
# LED PIO

- The ledPIO dashboard monitors the live state of the physical LED PIO bits in the Qsys design and displays them on LED widgets in the console.
- A “toggle” button is provided along with each LED to allow the user to toggle the state of the LED.



# Thermometer SPI

- The thermSPI dashboard provides a dial widget to display the current temperature reading in Fahrenheit degrees. A time chart widget is also provided to track the history of the raw thermometer samples.
- These widgets are constantly updated with the live value of the thermometer.



# On Chip RAM

- The ocRAM dashboard provides a collection of widgets to view a 64 byte window into the onchip RAM. The user can set the base address of the 64 byte view. This view is constantly updated with the live contents of the RAM.
- There are also widgets provided to allow the user to write 8, 16 and 32 bit data to arbitrary addresses in the memory.
- There are also widgets that can fill the entire RAM with zero, one or random patterns.

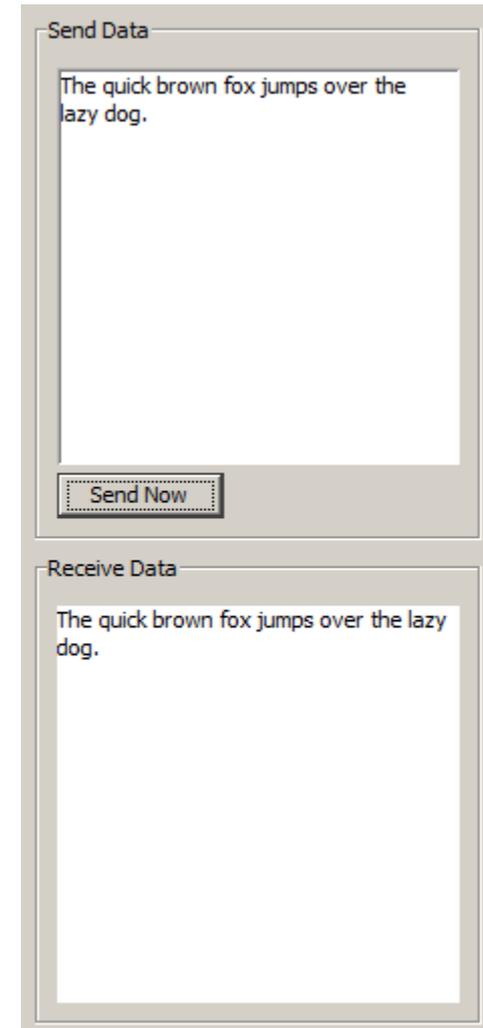
The screenshot displays the ocRAM dashboard interface, which is divided into several functional sections:

- View Memory:** This section allows users to view a 64-byte window of memory. It features an "Address" input field with the value "0x0100" and an "Apply New Address" button. Below this is a "Data" table showing the contents of four words at addresses 0x0100 through 0x0130.
- Change Memory:** This section provides tools for writing data to memory. It includes an "Address" input field (set to "0x0124") and a "Data" input field (set to "0xfacecafe"). Below these are three buttons: "Write 8-bit Value", "Write 16-bit Value", and "Write 32-bit Value".
- Fill Memory:** This section contains three buttons: "Fill Zero", "Fill One", and "Fill Random", used for initializing memory with specific patterns.

ADDR	WORD 0	WORD 1	WORD 2	WORD 3
0x0100	0x73198426	0x903315d2	0x09ddd7f4	0xc38cd93a
0x0110	0x5409d322	0x51048848	0x008f5886	0xc2fcc5b2
0x0120	0x61178d20	0xfacecafe	0xc2aa8cd8	0x4b051cc0
0x0130	0x3ca2a7ba	0xdab1c36e	0xcc95daec	0x825824e2

# Byte Stream

- The byteStream dashboard allows the user to send data down thru the bytestream component inside the Qsys system. The bytestream component has a FIFO connecting the source interface to its sink interface, so data that is sent down through the source is simply returned back through the sink.
- The receive data widget is continuously polling for live data coming back thru the sink interface.
- The user can type data into the send data widget and press the “Send Now” button to send the data down thru the source interface.



# JTAG UART

- The jtagUart dashboard allows the user to send data and receive data thru the JTAG UART component inside the Qsys system. The nios2-terminal utility can be used to capture data sent from the dashboard and send data into the dashboard..
- The receive data widget is continuously polling for live data coming back thru the JTAG UART interface.
- The user can type data into the send data widget and press the “Send Now” button to send the data down thru the JTAG UART interface.

```
Hello from System Console!!!
```

