

Web-based Hardware Control

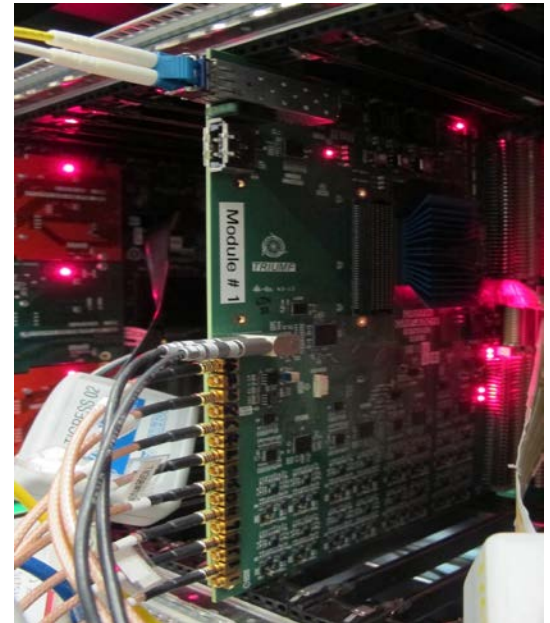
Leveraging MSCB + JSON

Background

Initially tasked with Firmware for GRIF-16
Module

Other devices need control

- GRIF-C
- GRIF-CDM



GRIFFIN Requirements

Stand-alone

- No MIDAS Required

Web-based

- No Software to Install

User-Friendly

- Expert not required

Introductions

What is being controlled?

- ADCs, DACs, Sensors, GPIO, HDL

What is MSCB?

- Small embedded control protocol

What is JSON?

- { "keys": "values", "arrays": ["a": 0, "b": 1] }

Why MSCB?

Ultralightweight

- Originally for 8051s, RS485

Extensible

- Just don't break the existing tools

Pre-existing tools

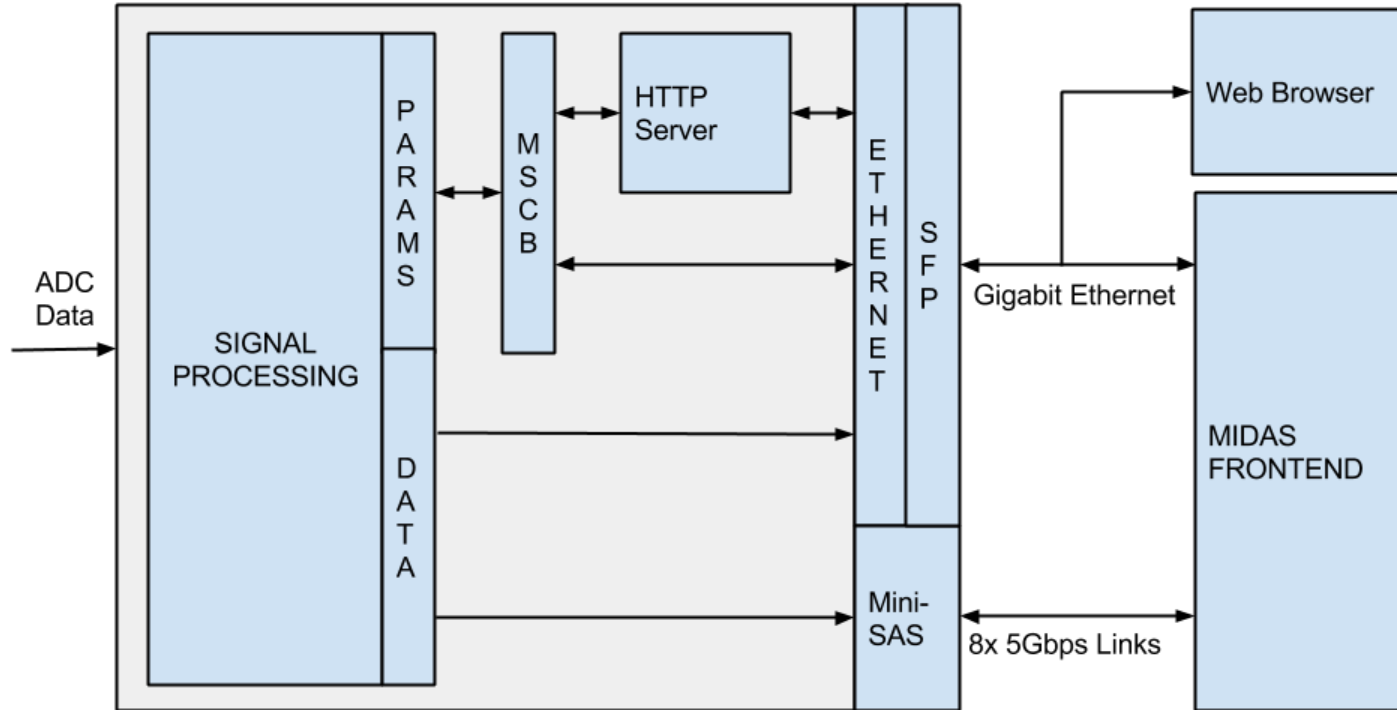
- Useful backdoor

Environment

Arria V FPGA

- Soft-core 32-bit NIOS II processor
- uC/OS-II Real-time Kernel
- Iniche Network Stack
- 64MB DDR3 RAM (using under 1MB)
- Gigabit Ethernet via SFP Module

Data Flow



Components

MSCB Core Library

- Handles uC/OS-II locking and MSCB protocol

Nodes

- Handle Interface to actual Hardware

Submaster

- MSCB Router, UDP

Webserver

- JSON, HTTP

Javascript

- MSCB Client Library, AngularJS (optional)

MSCB Core Library

Handles the Protocol

- Non-OS specific code

Called by Everything

- Node, Submaster, Webserver

Not Stand-alone

- Requires small stubbed out OS-related functions

Nodes

- Each Node is Separate uc/OS Task
- Provide Interface to Hardware
- Configurable Update Rate
- Self-contained
- Simple
- List of Parameters
- Data Verification

Submaster

Backdoor

- Provides low-level alternate control scheme

IP Control

- Controlled via standard MSCB Node

Extensible

- UDP, USB, Serial all possible

Webserver

Mongoose 4.x Based

- Hacked up to compile under NIOS/Iniche

Asymmetric Data Paths

- Acts as MSCB Bridge for Writes
- Pulls directly from Node data for Reads

Javascript

Small MSCB Library

Uses Typed Arrays Out

JSON In

Embedded in Larger Framework

Client-side Timestamping

Polling

JSON Example

```
{ "nodeId": 1, "vars": {
  "serial": { "id": 0, "w": 18, "u": 56, "p": 0, "f": 0, "s": 0, "d": "0x00E9428A08620709"},
  "cpu_temp": { "id": 1, "w": 4, "u": 8, "p": 0, "f": 2, "s": 0, "d": 31},
  "cc_lock": { "id": 2, "w": 4, "u": 50, "p": 0, "f": 0, "s": 0, "d": true},
  "cc_freq0": { "id": 3, "w": 4, "u": 20, "p": 0, "f": 0, "s": 0, "d": 100000480},
  "cc_freq1": { "id": 4, "w": 4, "u": 20, "p": 0, "f": 0, "s": 0, "d": 200000960},
  "cc_refck": { "id": 5, "w": 4, "u": 20, "p": 0, "f": 0, "s": 0, "d": 50000000},
  "dc_freq": { "id": 6, "w": 4, "u": 20, "p": 0, "f": 0, "s": 0, "d": 100000480},
  "hw_sw_m": { "id": 7, "w": 4, "u": 54, "p": 0, "f": 2, "s": 0, "d": 0},
  "hw_id": { "id": 8, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1195782454},
  "hw_time": { "id": 9, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1432950354},
  "sw_id": { "id": 10, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1195782454},
  "sw_time": { "id": 11, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1432945244},
  "hwbuild": { "id": 12, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1432950354},
  "swbuild": { "id": 13, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 1432985600},
  "up_time": { "id": 14, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 67649},
  "dac_ch0": { "id": 15, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 0},
  "dac_ch1": { "id": 16, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 0},
  "ref_clk": { "id": 17, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 3},
  "allowWr": { "id": 18, "w": 4, "u": 50, "p": 0, "f": 0, "s": 0, "d": false},
  "sp_run": { "id": 19, "w": 4, "u": 50, "p": 0, "f": 0, "s": 0, "d": false},
  "sp_rst": { "id": 20, "w": 4, "u": 50, "p": 0, "f": 0, "s": 0, "d": false},
  "sp_mod": { "id": 21, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 0},
  "sp_invr": { "id": 22, "w": 4, "u": 50, "p": 0, "f": 0, "s": 0, "d": true},
  "a_cid0": { "id": 23, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 143},
  "a_cid1": { "id": 24, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 143},
  "a_cid2": { "id": 25, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 143},
  "a_cid3": { "id": 26, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 143},
  "a_cid4": { "id": 27, "w": 4, "u": 54, "p": 0, "f": 0, "s": 0, "d": 143}
```

Results

Tiny Footprint

- Takes a small fraction of available resources

Rapid Parameter Adding

- One line addition in a few files


Responsive

- Sub 50ms responses typical

Customizable UI

- Sliders, Drop-downs, Lists, etc

Results (continued)

 **GRIF16 Module 4** CHANNELS **SETTINGS** RATES

DAQ **CLOCK** ADC STATUS MINI-SAS ETHERNET SFP

Using Clock Cleaner:

Reference Clock: Internal Oscillator
 eSATA

Clock Cleaner Locked:

Clock Cleaner Loss Count:

Clock Cleaner Reference Frequency:

DATA Clock Frequency:

ADC Clock Frequency:

DAC Clock Frequency:

Expert Settings

DAQ

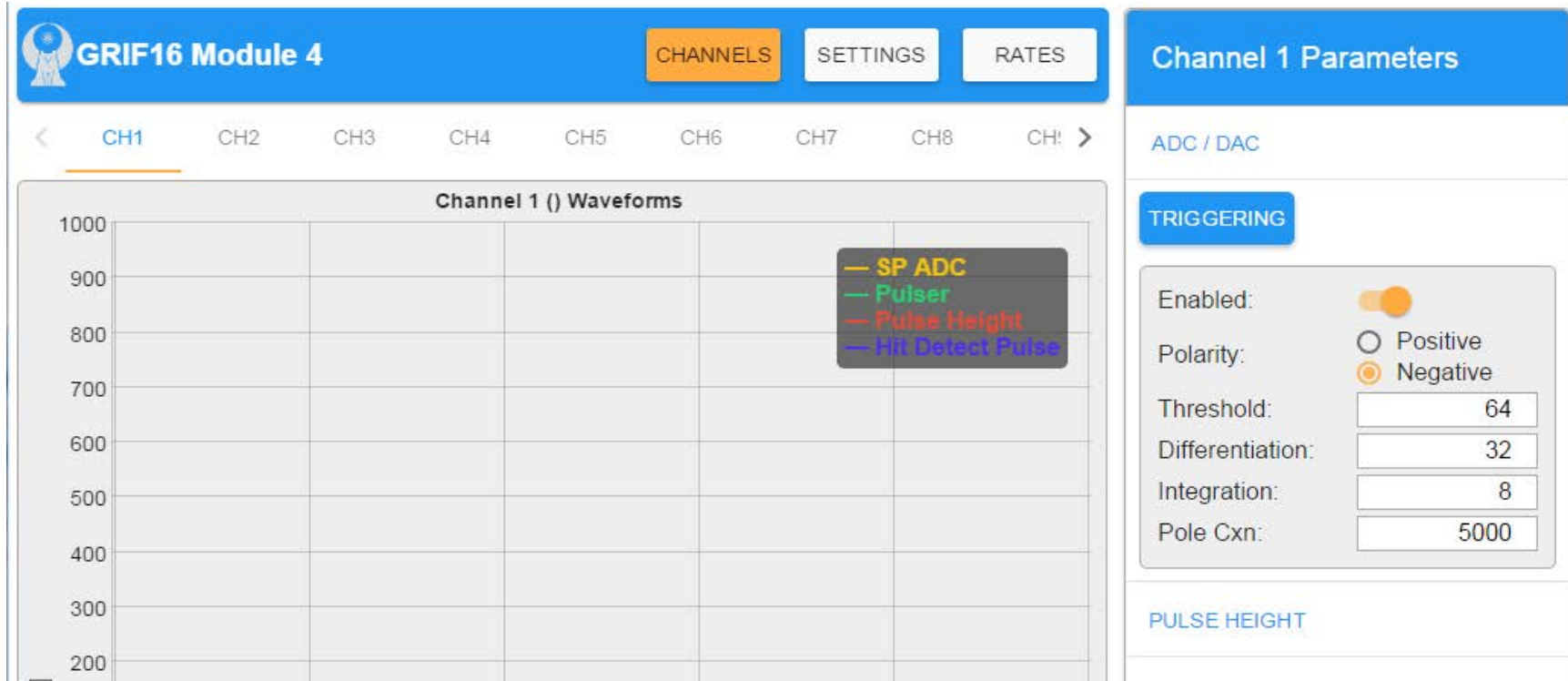
DEBUG

Reset Signal Processing:

Module Level:

Module Status

Results (continued)



Issues

Not widely used

Lacks built-in Meta-data

Polling only

Security

Future

Websockets

Easy MIDAS Integration

Code Cleanup

Questions