PAUL SCHERRER INSTITUT

PSI

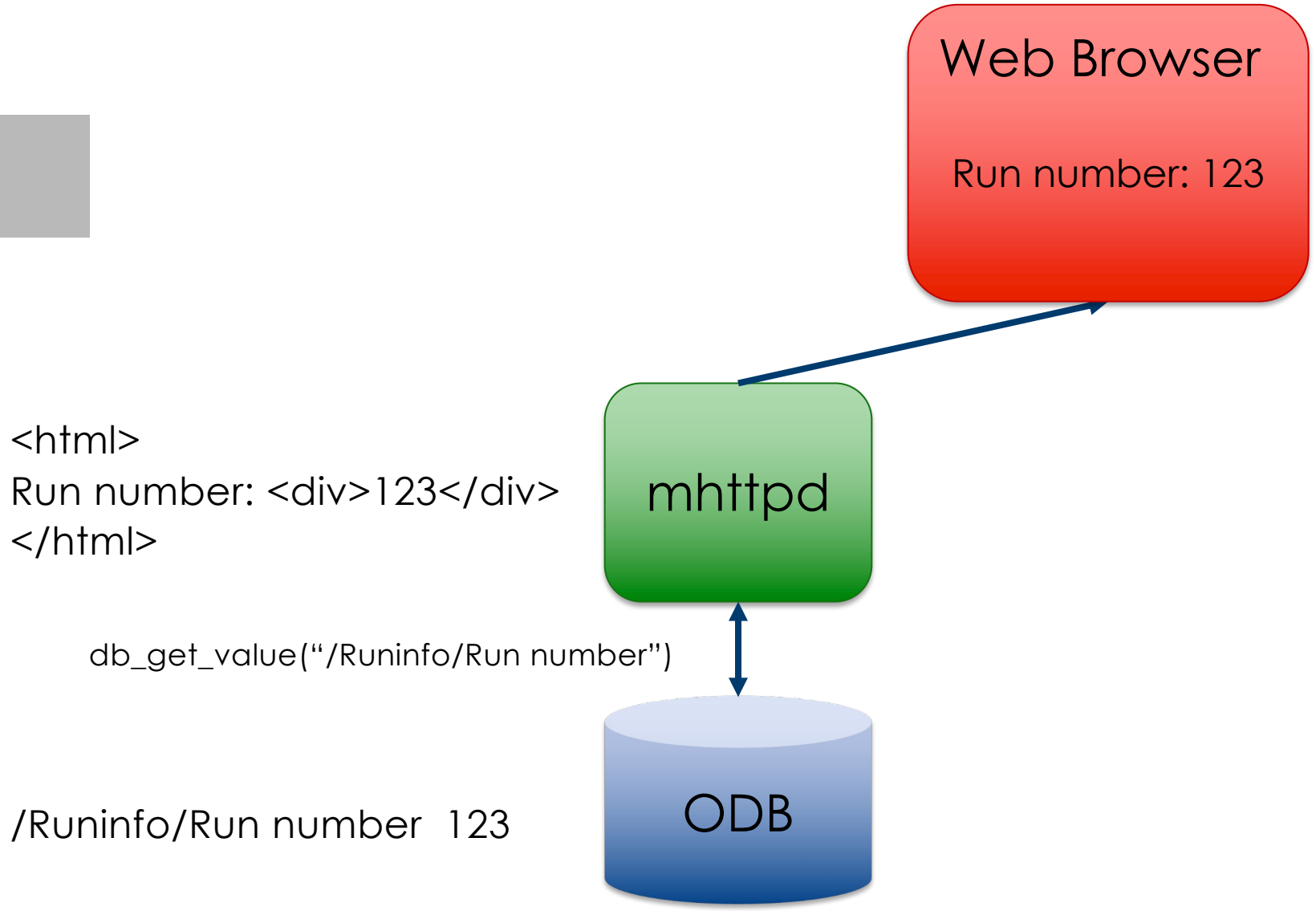
Stefan Ritt, Paul Scherrer Institute

# New MIDAS features

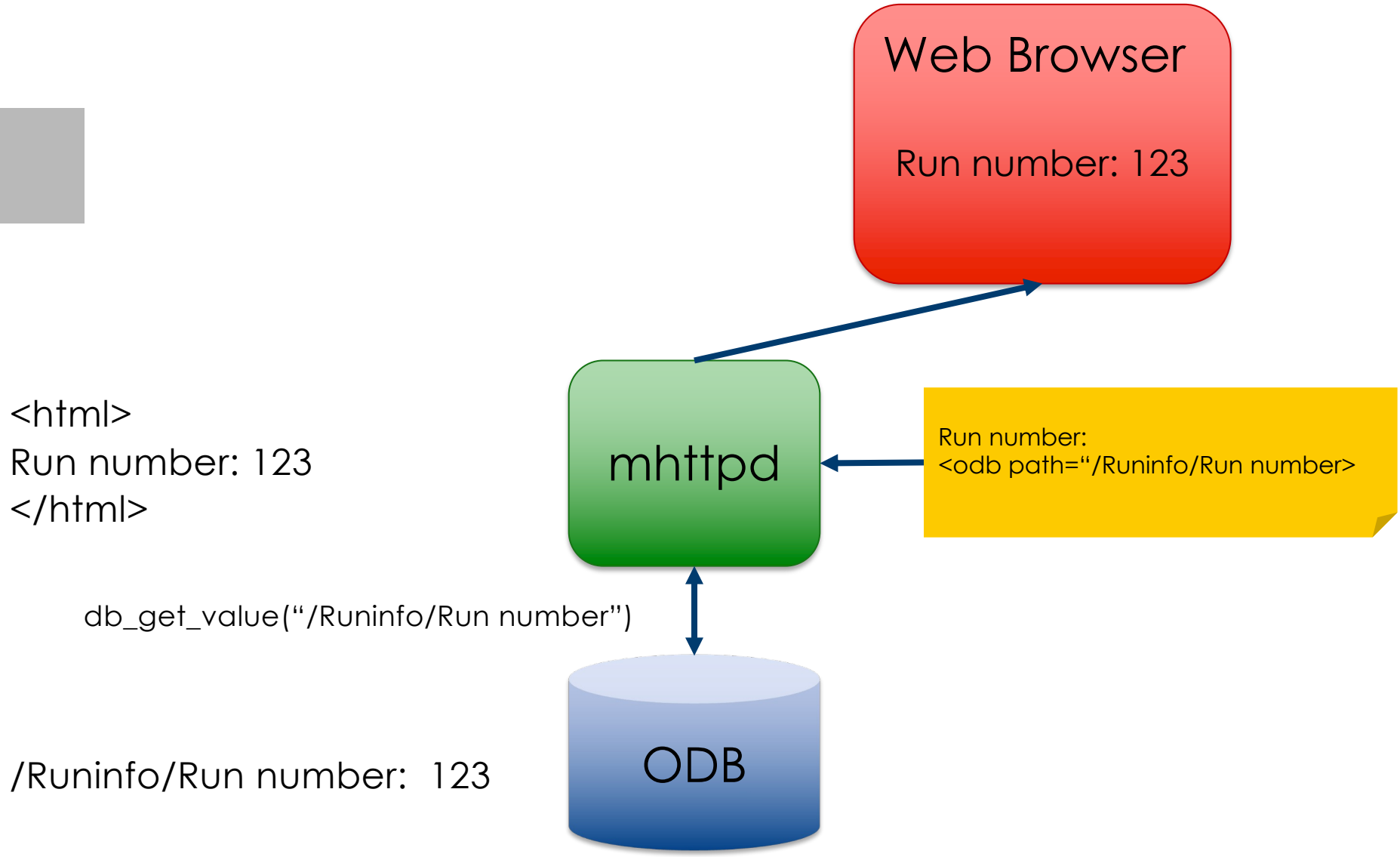
# New web interface to MIDAS

- Historically mhttpd generated HTML pages for experiments from its C code
  - Only possibility back in 1990's
  - Pages were static
  - "Custom pages" needed special syntax, limited use
- New way of doing things:
  - Extensive use of JavaScript in browsers to generate pages dynamically
  - JSON-RPC interface to ODB (KO) lets web pages interact with ODB and MIDAS
  - Each MIDAS function has a browser equivalent, e.g. mjsonrpc_db_get_value(), mjsonrpc_cm_exist(), …
  - All mhttpd pages get converted into plain HTML pages
  - Pages can be customized easily

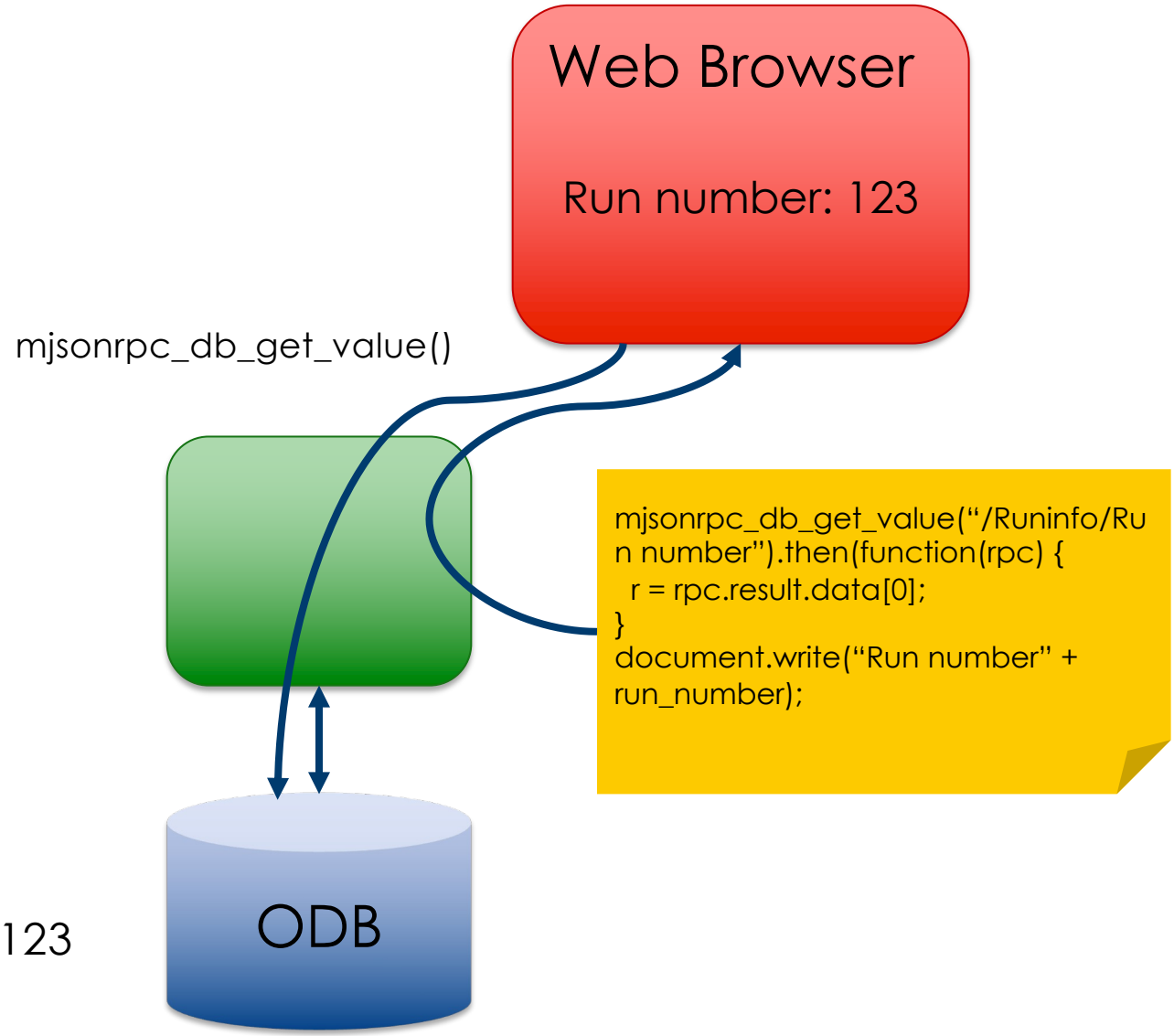
# The old way


Web Browser
Run number: 123
mhttpd
<html>
Run number: 123
</html>
db_get_value("/Runinfo/Run number")
ODB
/Runinfo/Run number 123

# Custom pages



# MJSON-RPC




/Runinfo/Run number: 123

# New HTML syntax

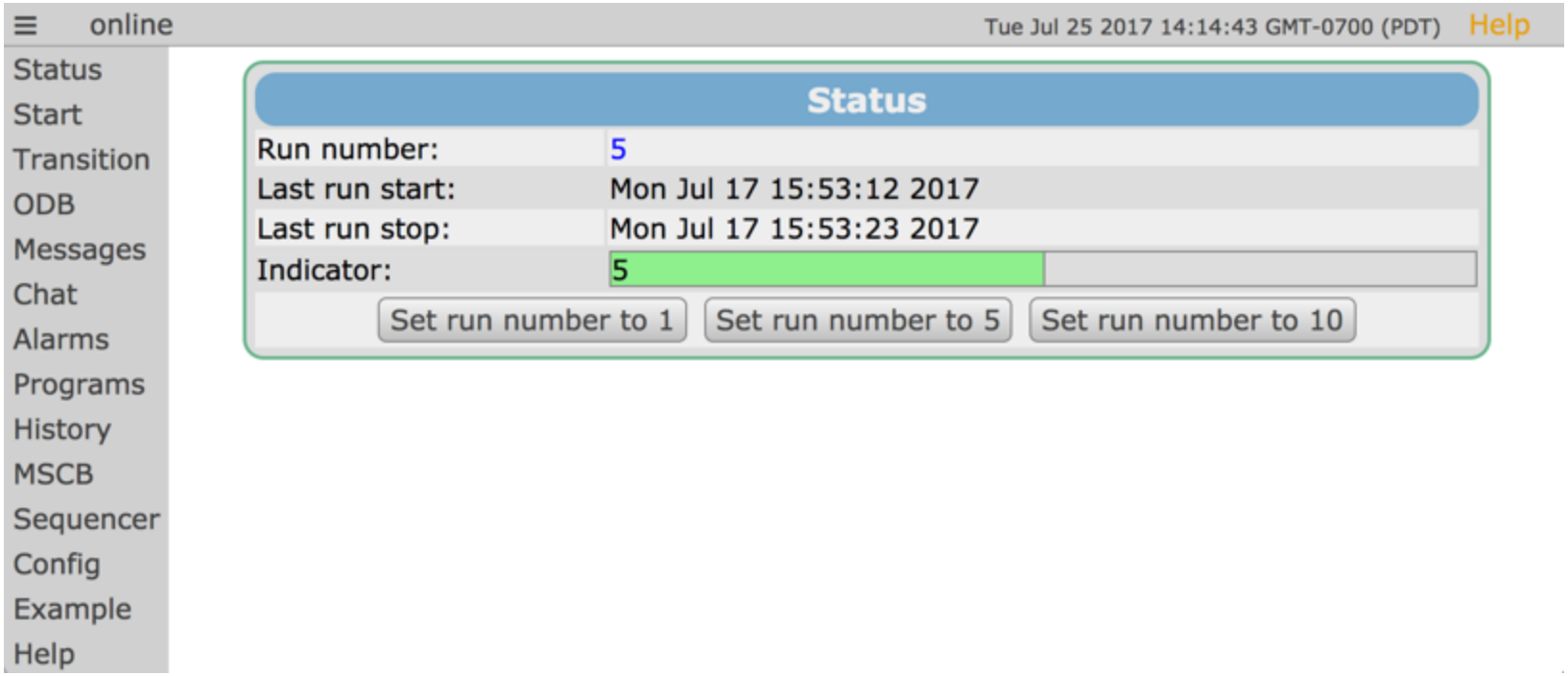
```
mjsonrpc_db_get_value("/Runinfo/Run number").then(function(rpc) {
  r = rpc.result.data[0];
  var r = document.getElementById("runnumber");
  r.innerHTML = "Run number: " + run_number;
}
```

```
<body class="mcss" onload="mhttpd_init('Test', 1000)">

Run number:
  <div name="modbvalue" data-odb-path="/Runinfo/Run number"
    data-odb-editable="1"></div>
```

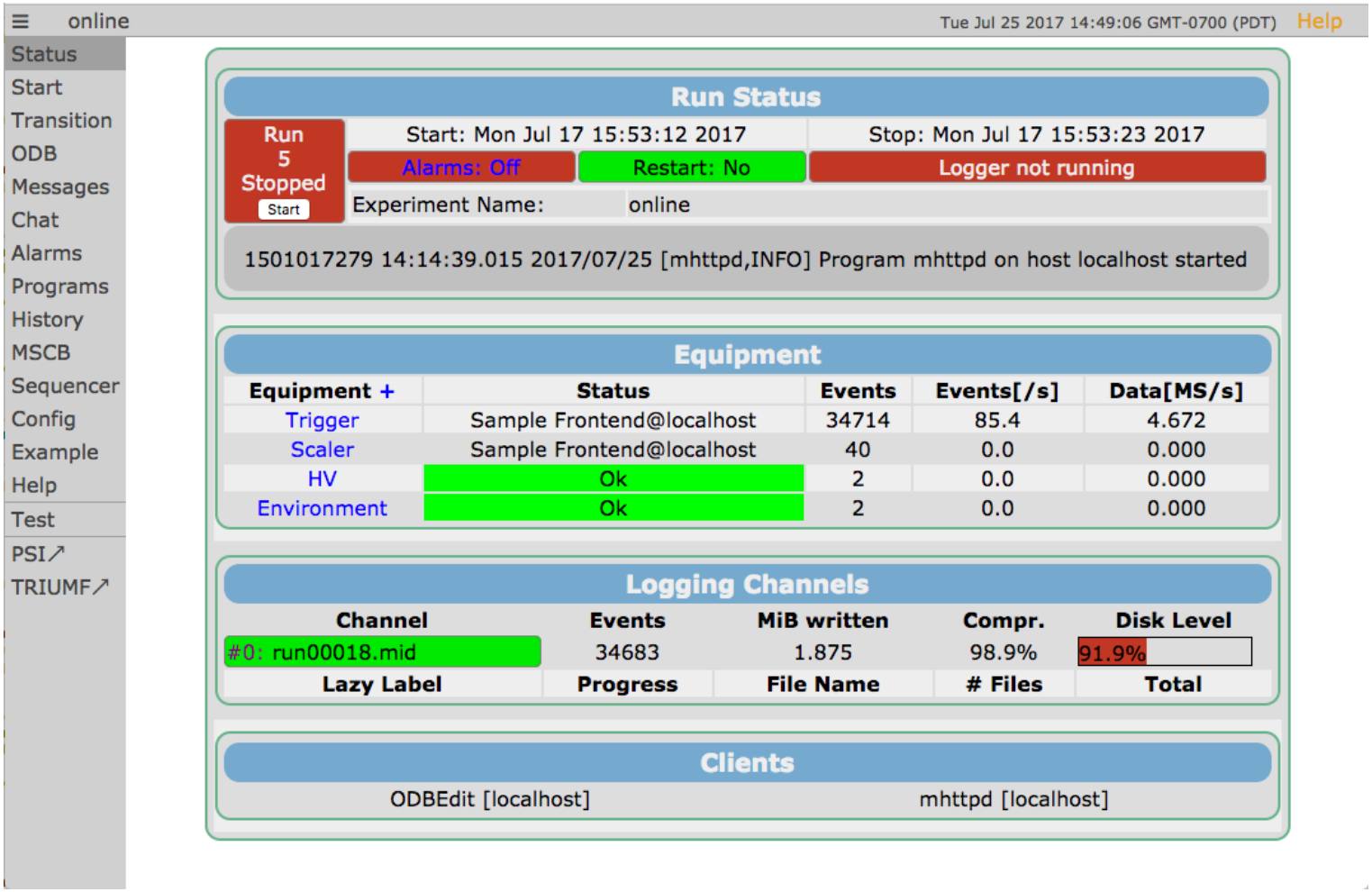
# Complete example

```
<body class="mcss"
onload="mhttpd_init('Test', 1000)">

<div id="mheader"></div>
<div id="msidenav"></div>

<div id="mmain">
  <table class="mtable">
    <tr>
      <th colspan="2"
class="mtableheader">Status</th>
    </tr>
    <tr>
      <td style="width: 200px">
        Run number:
      </td>
      <td>
        <div name="modbvalue" data-odb-
path="/Runinfo/Run number" data-odb-
editable="1"></div>
      </td>
      <td>
        <div name="modbbar" style="width: 500px"
data-odb-path="/Runinfo/Run number" data-
max-value="10"
            data-color="lightgreen"></div>
      </td>
    </tr>
    <tr>
      <td>
        <button name="modbbutton"
class="mbutton" data-odb-path="/Runinfo/Run
number" data-odb-value="1">Set run
          number to 1
        </button>
      </td>
    </tr>
  </table>
</div>
</body>
</html>
```

# Resulting HTML page



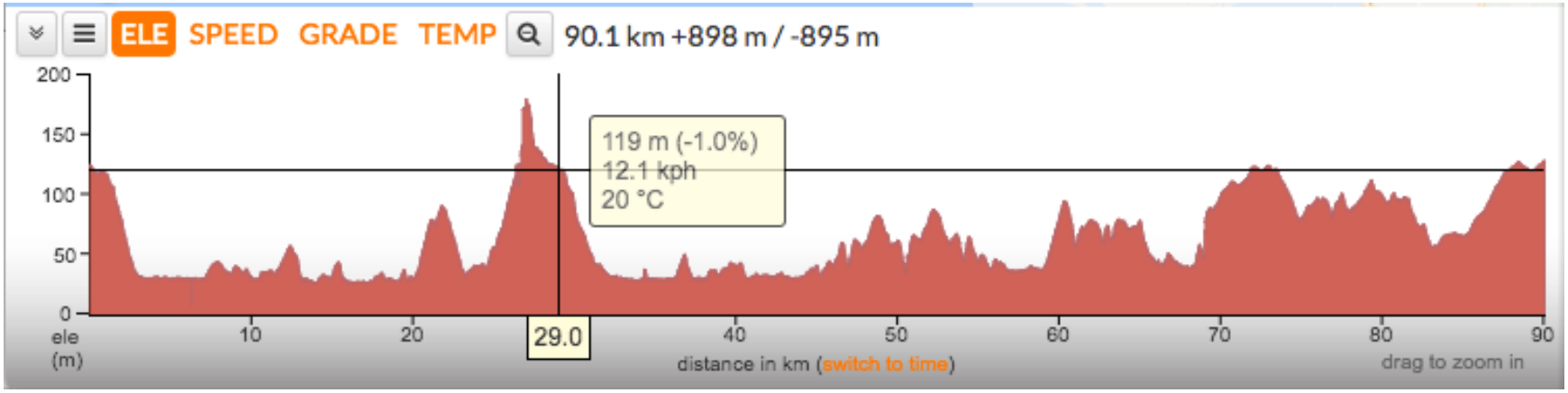
# New Status Page (Shouyi Ma)



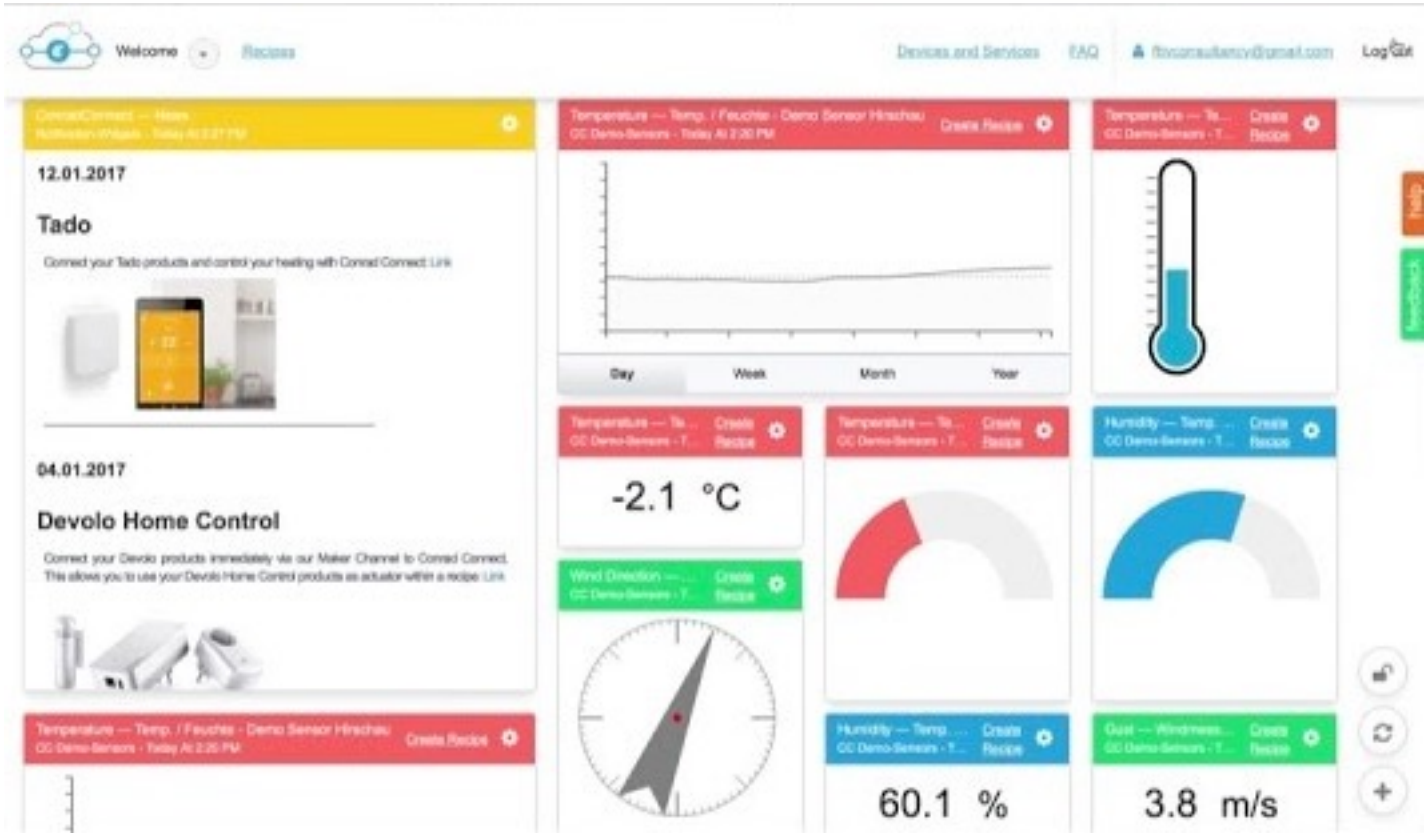
# General Concept

- Use special data-xxx tags for MIDAS HTML elements
  - ODB values (editable)
  - Horizontal color bar
  - Vertical color bar
  - General gauge with scales
  - Buttons to set some ODB value
  - History panel:

    <div name="**mhistory**" **data-panel**="Temperatures"
    **data-size**="1000px">

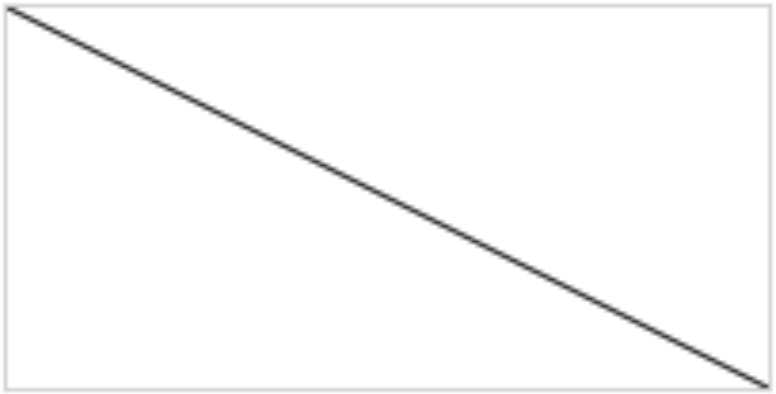


# Conrad Connect



# HTML5 Canvas

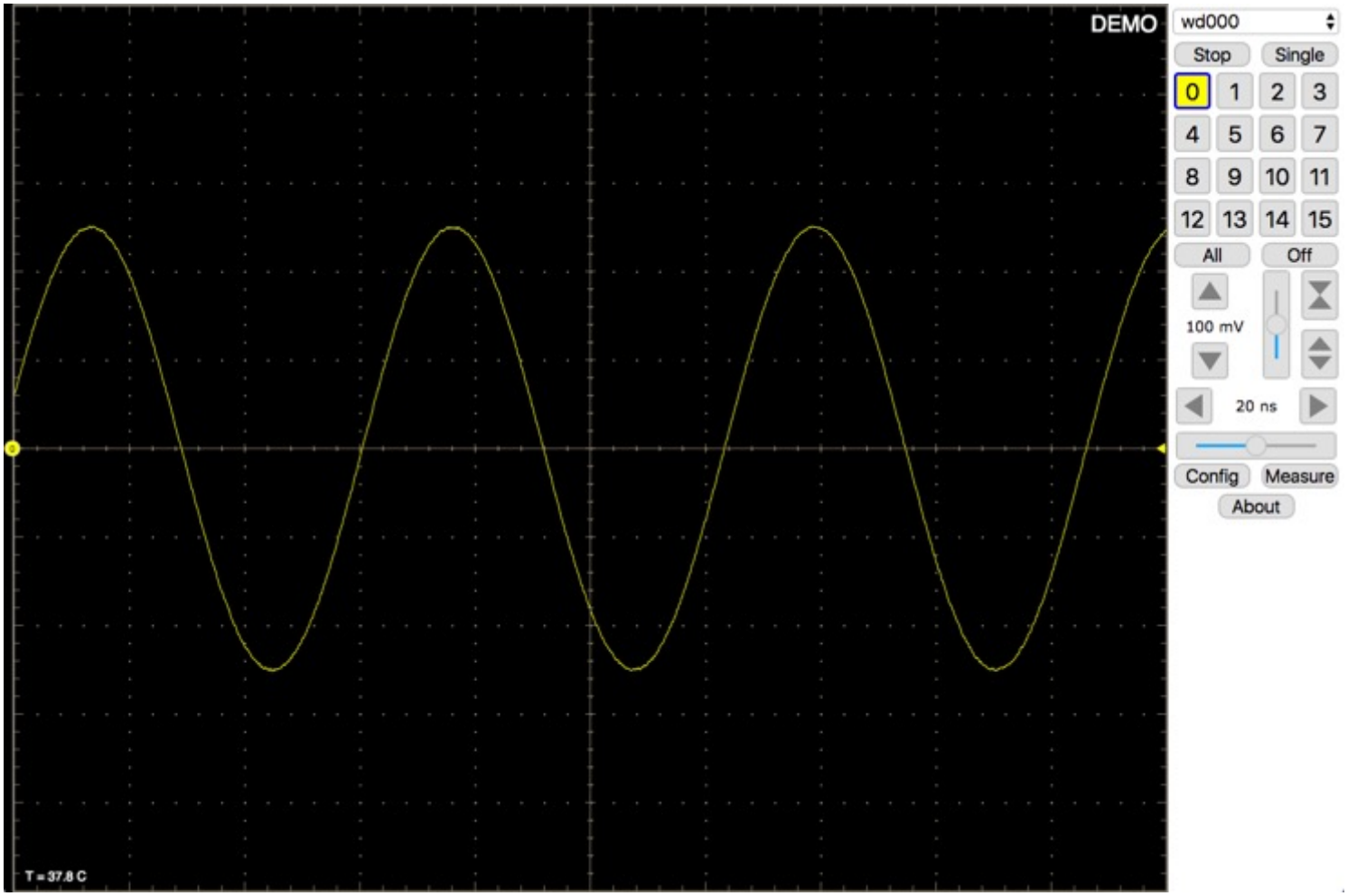
```
<canvas id="myCanvas" width="200" height="100">
</canvas>

var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```



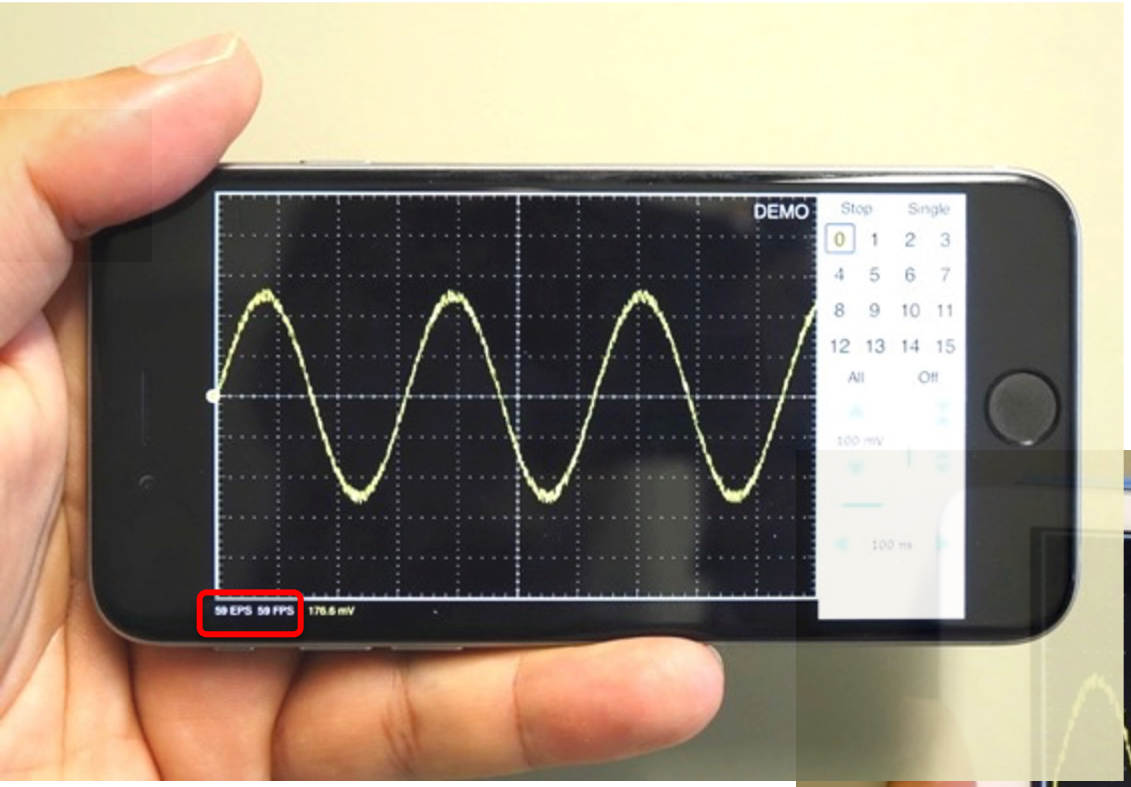
# JavaScript + canvas

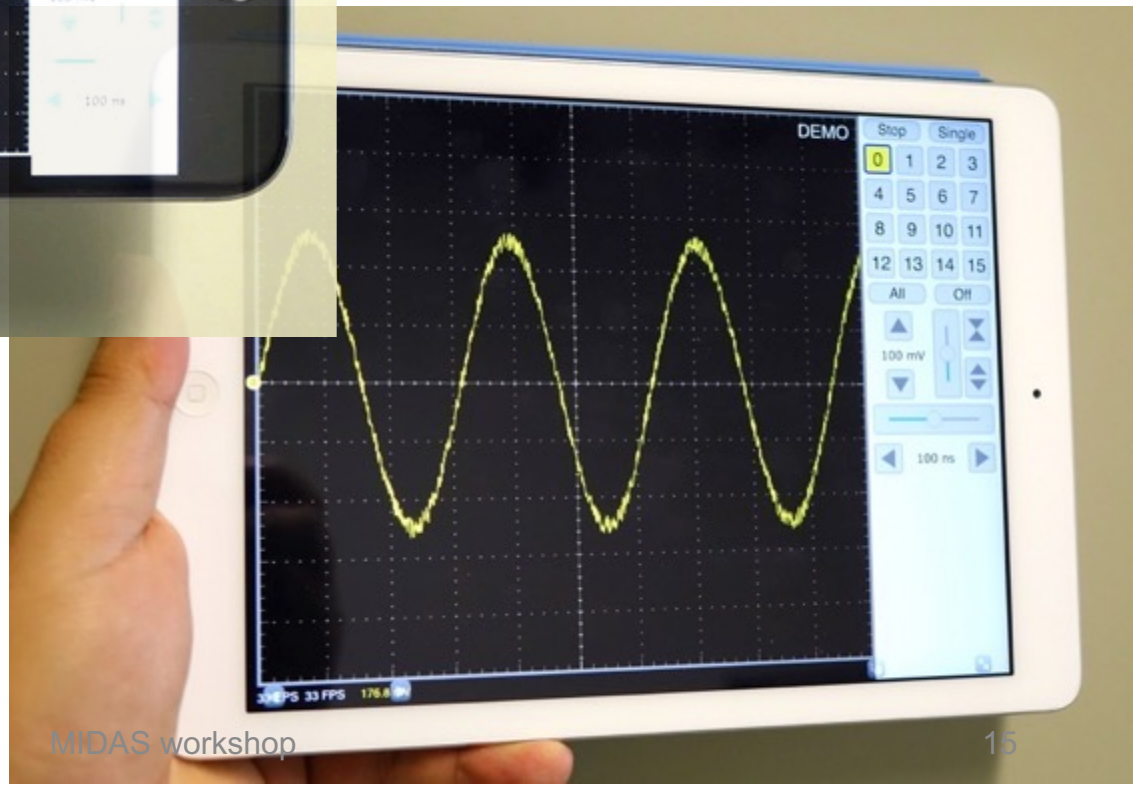
- Drawing in a canvas with JS is as fast as native (QT) applications a few years ago
- Drawing library quite complete (alpha, transformations, 3D, …)
- Development directly in browser, no compilation
- Runs on all OS including smart devices
- This is the future!

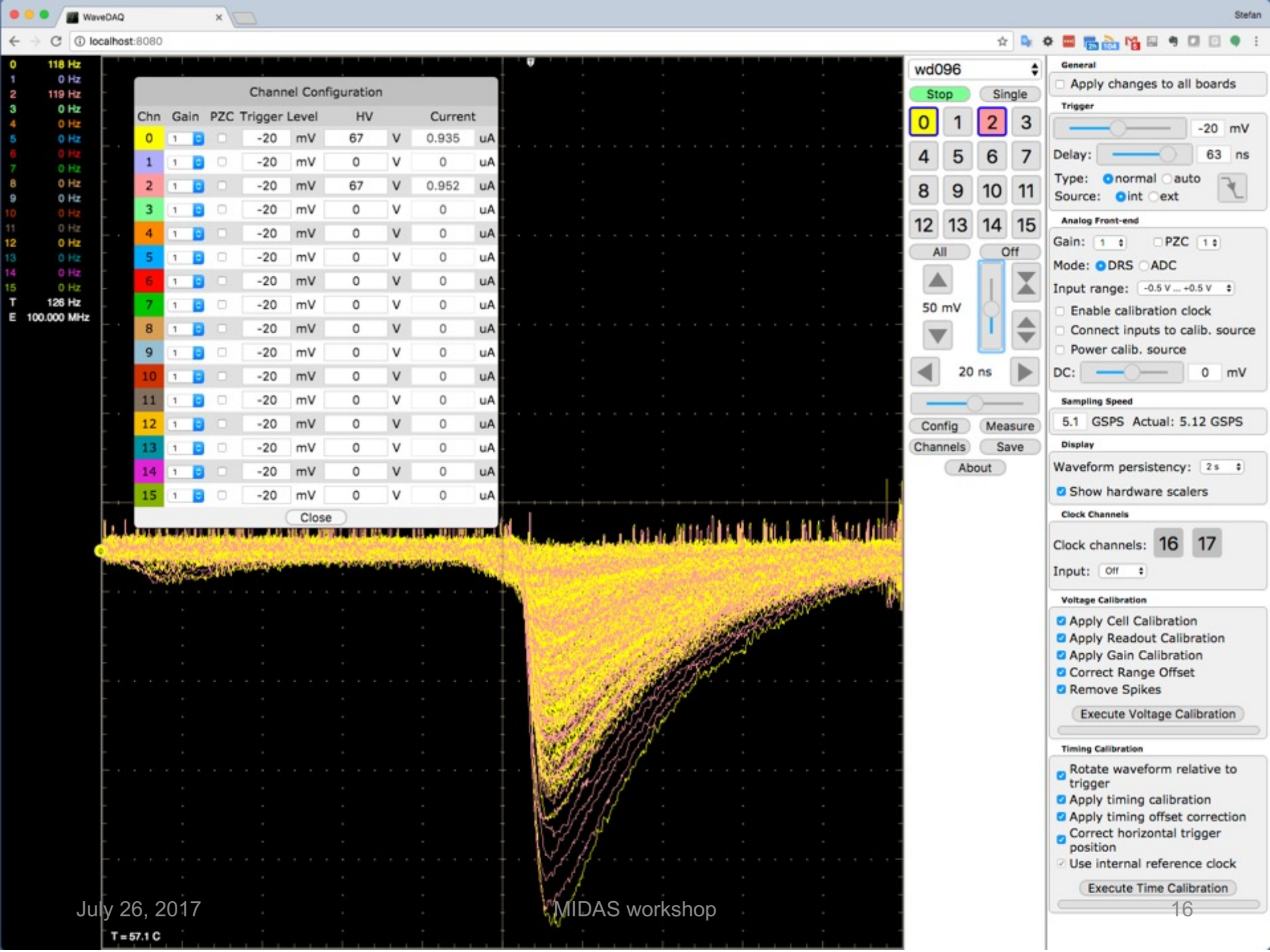
# http://midas.psi.ch/scope

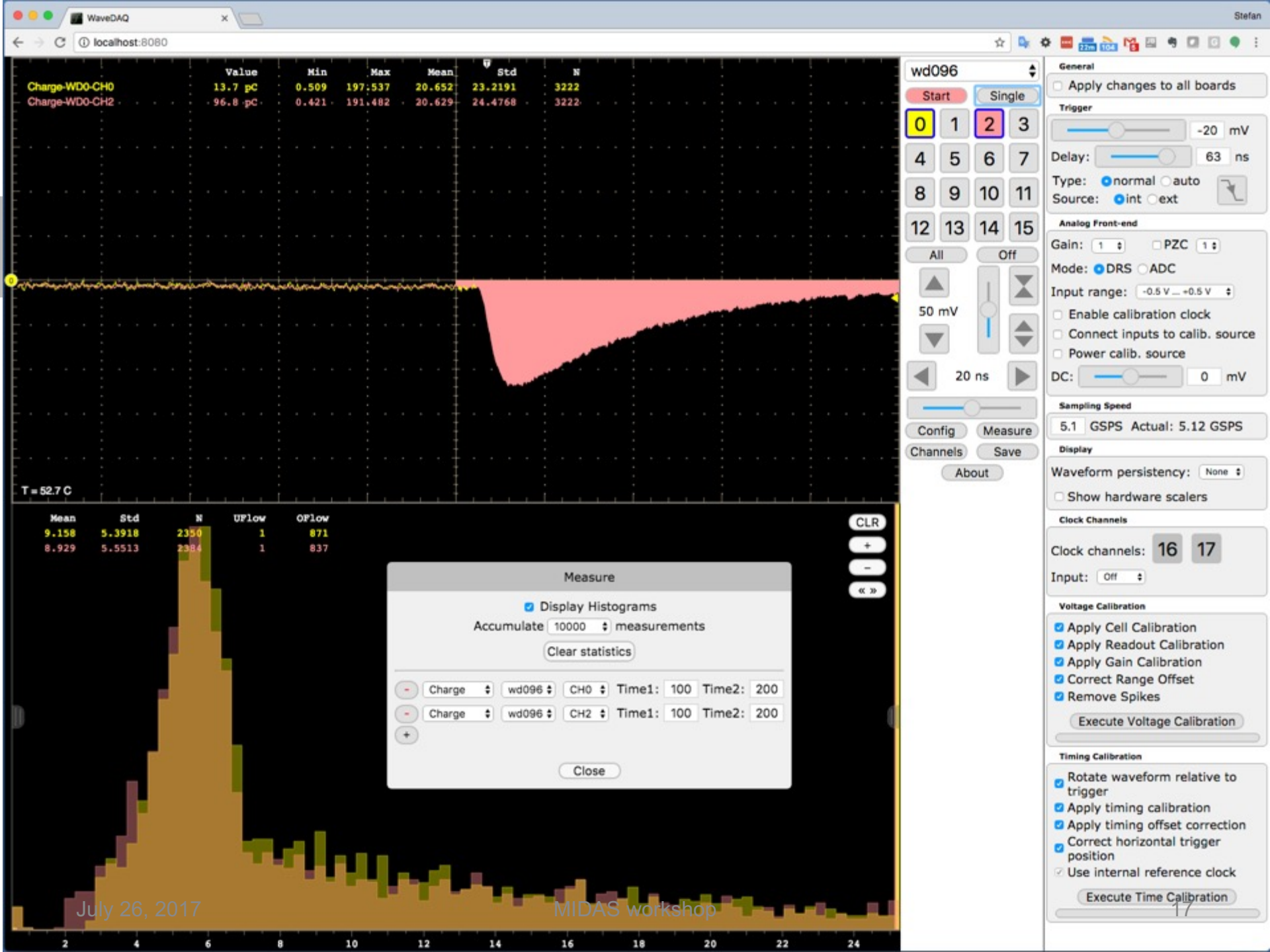


# Display on mobile devices

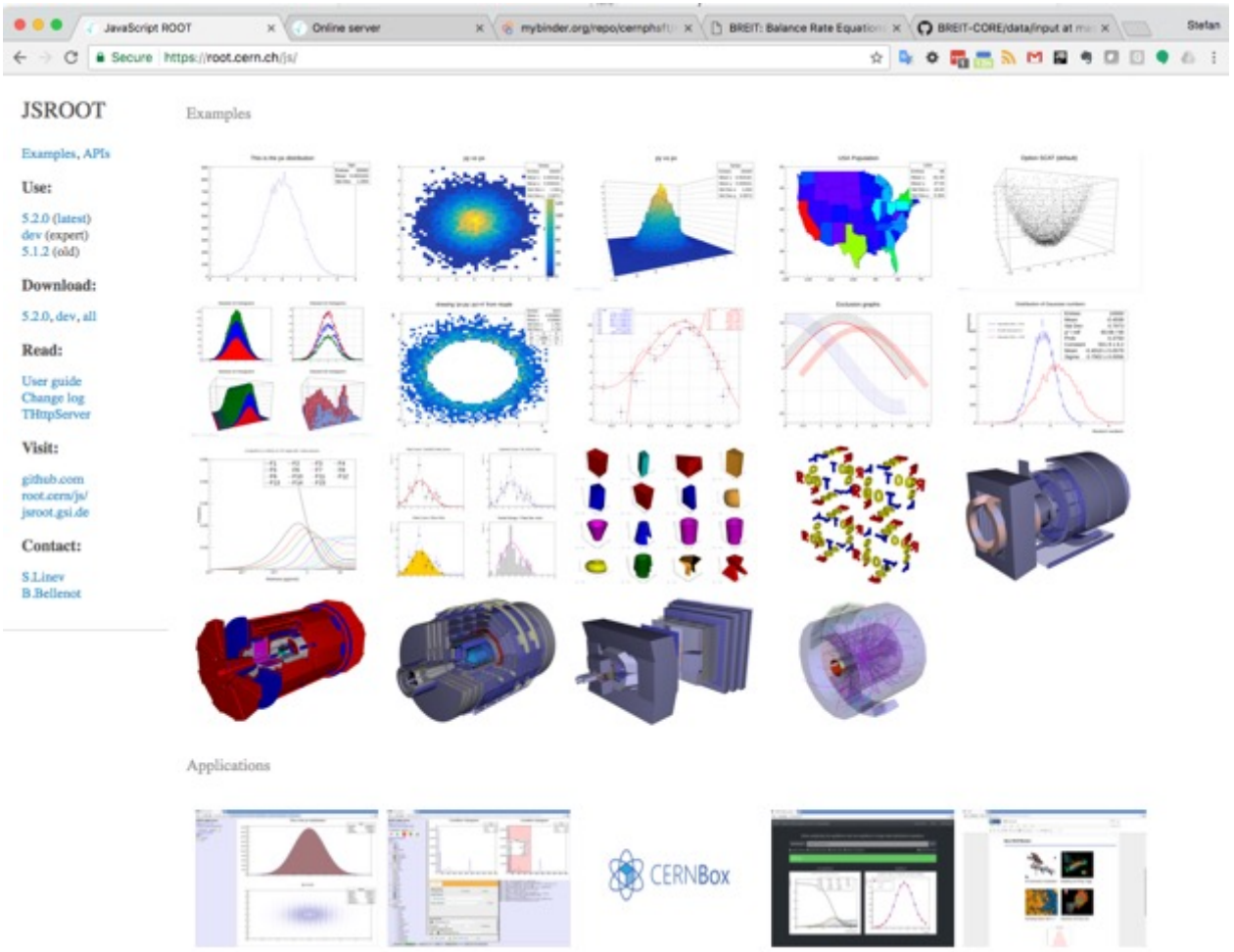



WaveDAQ
localhost:8080
Stefan
0 118 Hz
1 0 Hz
2 119 Hz
3 0 Hz
4 0 Hz
5 0 Hz
6 0 Hz
7 0 Hz
8 0 Hz
9 0 Hz
10 0 Hz
11 0 Hz
12 0 Hz
13 0 Hz
14 0 Hz
15 0 Hz
T 126 Hz
E 100.000 MHz
Channel Configuration
Chn Gain PZC Trigger Level HV Current
0 1 -20 mV 67 V 0.935 uA
1 1 -20 mV 0 V 0 uA
2 1 -20 mV 67 V 0.952 uA
3 1 -20 mV 0 V 0 uA
4 1 -20 mV 0 V 0 uA
5 1 -20 mV 0 V 0 uA
6 1 -20 mV 0 V 0 uA
7 1 -20 mV 0 V 0 uA
8 1 -20 mV 0 V 0 uA
9 1 -20 mV 0 V 0 uA
10 1 -20 mV 0 V 0 uA
11 1 -20 mV 0 V 0 uA
12 1 -20 mV 0 V 0 uA
13 1 -20 mV 0 V 0 uA
14 1 -20 mV 0 V 0 uA
15 1 -20 mV 0 V 0 uA
Close
wd096
Stop
Single
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
All
Off
50 mV
20 ns
Config
Measure
Channels
Save
About
General
Apply changes to all boards
Trigger
-20 mV
Delay: 63 ns
Type: normal auto
Source: int ext
Analog Front-end
Gain: 1 PZC 1
Mode: DRS ADC
Input range: -0.5 V ... +0.5 V
Enable calibration clock
Connect inputs to calib. source
Power calib. source
DC: 0 mV
Sampling Speed
5.1 GSPS Actual: 5.12 GSPS
Display
Waveform persistency: 2 s
Show hardware scalers
Clock Channels
Clock channels: 16 17
Input: Off
Voltage Calibration
Apply Cell Calibration
Apply Readout Calibration
Apply Gain Calibration
Correct Range Offset
Remove Spikes
Execute Voltage Calibration
Timing Calibration
Rotate waveform relative to trigger
Apply timing calibration
Apply timing offset correction
Correct horizontal trigger position
Use internal reference clock
Execute Time Calibration
T = 57.1 C


WaveDAQ
Stefan
localhost:8080
Value Min Max Mean Std N
Charge-WD0-CH0 13.7 pC 0.509 197.537 20.652 23.2191 3222
Charge-WD0-CH2 96.8 pC 0.421 191.482 20.629 24.4768 3222
T = 52.7 C
wd096
Start
Single
0 1 2 3
4 5 6 7
8 9 10 11
12 13 14 15
All
Off
50 mV
20 ns
Config
Measure
Channels
Save
About
General
Apply changes to all boards
Trigger
-20 mV
Delay: 63 ns
Type: normal auto
Source: int ext
Analog Front-end
Gain: 1
PZC 1
Mode: DRS ADC
Input range: -0.5 V ... +0.5 V
Enable calibration clock
Connect inputs to calib. source
Power calib. source
DC: 0 mV
Sampling Speed
5.1 GSPS Actual: 5.12 GSPS
Display
Waveform persistency: None
Show hardware scalers
Clock Channels
Clock channels: 16 17
Input: Off
Voltage Calibration
Apply Cell Calibration
Apply Readout Calibration
Apply Gain Calibration
Correct Range Offset
Remove Spikes
Execute Voltage Calibration
Timing Calibration
Rotate waveform relative to trigger
Apply timing calibration
Apply timing offset correction
Correct horizontal trigger position
Use internal reference clock
Execute Time Calibration
Mean Std N UFlow OFlow
9.158 5.3918 2350 1 871
8.929 5.5513 2384 1 837
CLR
Measure
Display Histograms
Accumulate 10000 measurements
Clear statistics
Charge wd096 CH0 Time1: 100 Time2: 200
Charge wd096 CH2 Time1: 100 Time2: 200
Close
2 4 6 8 10 12 14 16 18 20 22 24

# https://root.cern.ch/js/



# Conclusions and Ideas

- Moving MIDAS to JavaScript web pages offers huge opportunities in experiment specific customization
  - Page update without complete reload
  - Modify standard pages
- Combine ODB values, gauges, history plots, … in one page
- Add other web technologies in **same** page
  - CouchDB database integration for experiment configuration
  - Histograms and analyzer plots through JSROOT
  - (Direct link to devices with JSON interface)

# MIDAS in the future