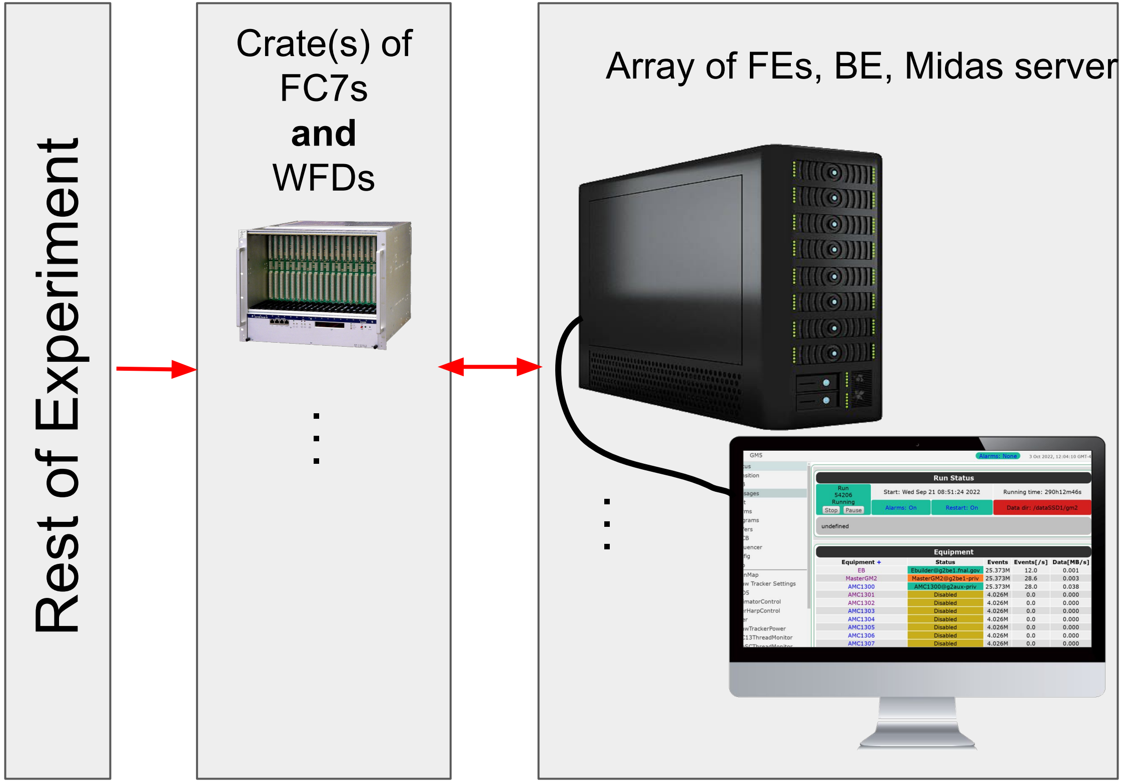
Data Acquisition (DAQ)

Jack Carlton
University of Kentucky

g-2 DAQ (Modified for One Crate Support)

- Retains g-2 hardware, but made more flexible
- Same general process:
  - Communicate with μTCA crate, initialize hardware
  - Read TCP packets from μTCA crate
  - Write to midas data banks



Run Status

Run 54206 Running

Start: Wed Sep 21 08:51:24 2022

Running time: 290h12m46s

Alarms: On

Restart: On

Data dir: /dataSSD1/gm2

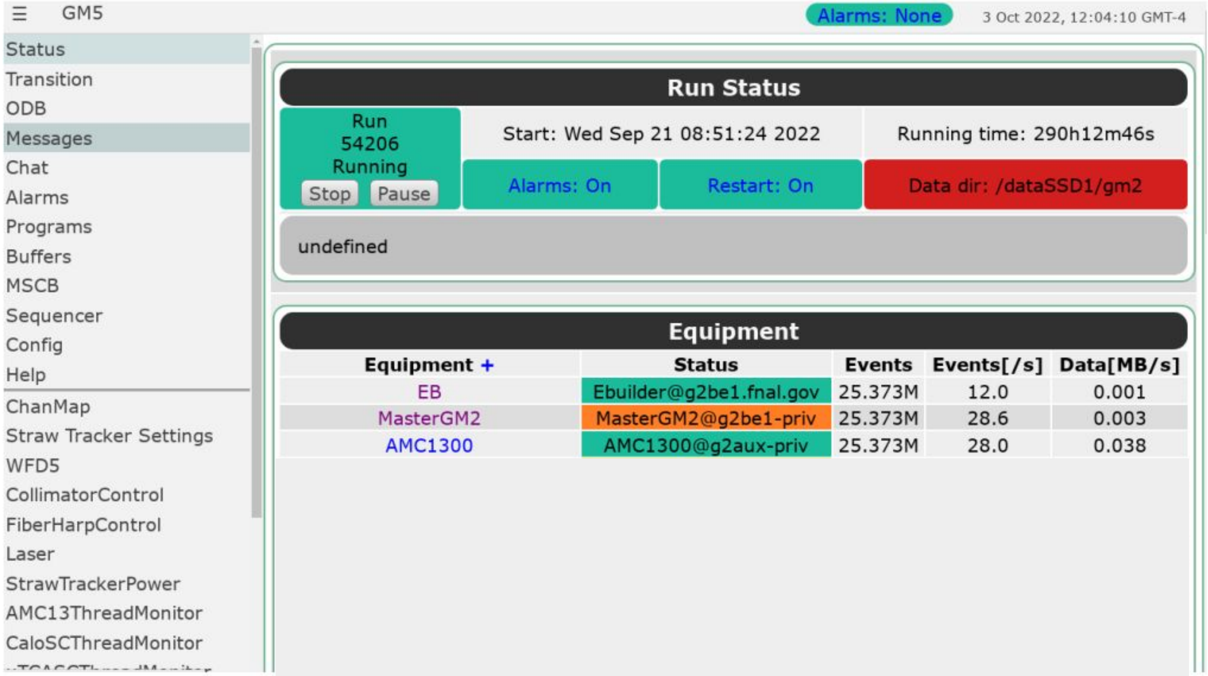
undefined

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
EB	Ebuilder@g2be1.fnal.gov	25.373M	12.0	0.001
MasterGM2	MasterGM2@g2be1-priv	25.373M	28.6	0.003
AMC1300	AMC1300@g2aux-priv	25.373M	28.0	0.038
AMC1301	Disabled	4.026M	0.0	0.000
AMC1302	Disabled	4.026M	0.0	0.000
AMC1303	Disabled	4.026M	0.0	0.000
AMC1304	Disabled	4.026M	0.0	0.000
AMC1305	Disabled	4.026M	0.0	0.000
AMC1306	Disabled	4.026M	0.0	0.000
AMC1307	Disabled	4.026M	0.0	0.000

Midas Framework

- C/C++ (mostly) package of modules for
  - run control,
  - expt. configuration
  - data readout
  - event building
  - data storage
  - slow control
  - alarm systems
  - Etc.
- Can link with custom software



Run Status

Run 54206 Running

Start: Wed Sep 21 08:51:24 2022

Running time: 290h12m46s

Alarms: On

Restart: On

Data dir: /dataSSD1/gm2

undefined

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
EB	Ebuilder@g2be1.fnal.gov	25.373M	12.0	0.001
MasterGM2	MasterGM2@g2be1-priv	25.373M	28.6	0.003
AMC1300	AMC1300@g2aux-priv	25.373M	28.0	0.038

Hardware Requirements

- Micro Telecom Computing (µTCA) crate with Modules:
  - Waveform Digitizers (WFD5(s)/Rider(s))
  - Controller (FC7)
  - MicroTCA Carrier Hub (MCH)
  - Advanced Mezzanine Card (AMC)
- “Frontend” computer with available PCIe slots for the following…
- Meinberg PCIe Clock Card
  - Custom connector
- 10 Gigabit Ethernet Network Interface Card (10GbE NIC)
  - SFP+ connectors
- Graphics Processing Unit (GPU) *Optional*

Software Requirements

- “Frontend” computer needs to be running Redhat-Enterprise Linux 7 (RHEL7)
  - Examples: Scientific Linux 7 (SL7), CentOS 7
- Midas
- Various other open source software libraries (root, boost, cactus, etc.)
- Some custom software libraries (DAQ frontend code, unpacking libraries, etc.)
- Software installation completely handled by installer on RHEL7 systems

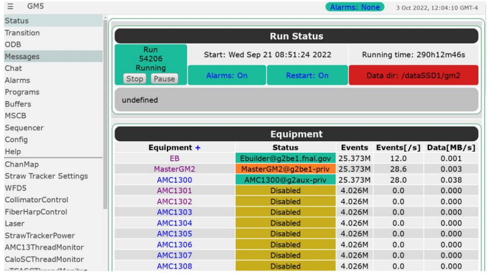
Installation, in a perfect world:

```
[1] git clone
git@github.com:PIONEER-Experi
ment/gm2daq-installer
[2] ./install.sh
```

patience…

```
[3] source
./setup_environment.sh
[4] ./start_midas_webpage.sh
```

Open browser, localhost:8080



Alarms: None

3 Oct 2022, 12:04:10 GMT-4

Run Status

Run 54206 Running

Start: Wed Sep 21 08:51:24 2022

Running time: 290h12m46s

Alarms: On

Restart: On

Data dir: /dataSSD1/gm2

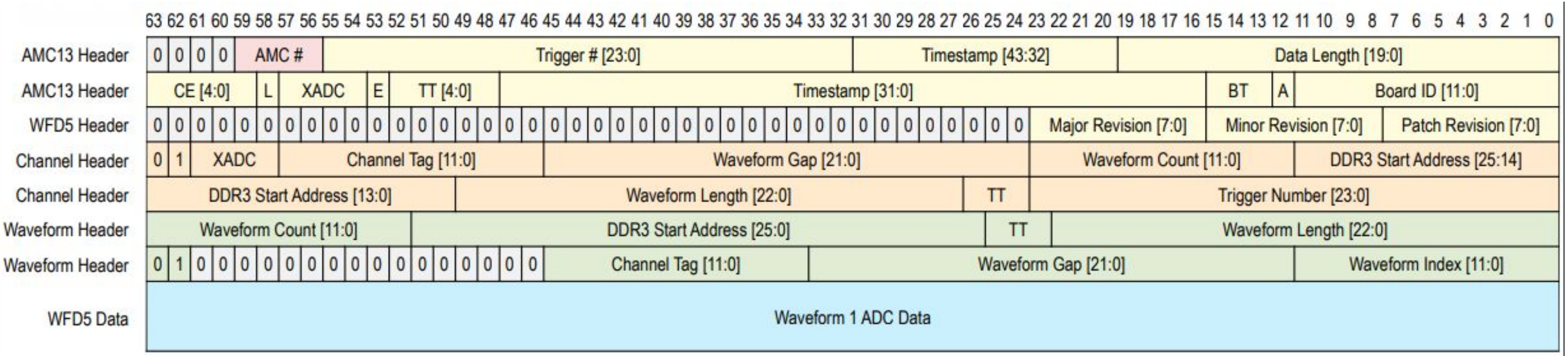
undefined

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
EB	Ebuilder@g2be1.fnal.gov	25.373M	12.0	0.001
MasterGM2	MasterGM2@g2be1-priv	25.373M	28.6	0.003
AMC1300	AMC1300@g2aux-priv	25.373M	28.0	0.038
AMC1301	Disabled	4.026M	0.0	0.000
AMC1302	Disabled	4.026M	0.0	0.000
AMC1303	Disabled	4.026M	0.0	0.000
AMC1304	Disabled	4.026M	0.0	0.000
AMC1305	Disabled	4.026M	0.0	0.000
AMC1306	Disabled	4.026M	0.0	0.000
AMC1307	Disabled	4.026M	0.0	0.000
AMC1308	Disabled	4.026M	0.0	0.000

Data Output

- Data is output “raw” in midas “CR” data banks
  - Written to run{#}.mid.lz4 files by mlogger
- Unpacked C++ data structure using unpacking library
  - Custom analyzers can import unpacking library
  - Unpacking library include in installer

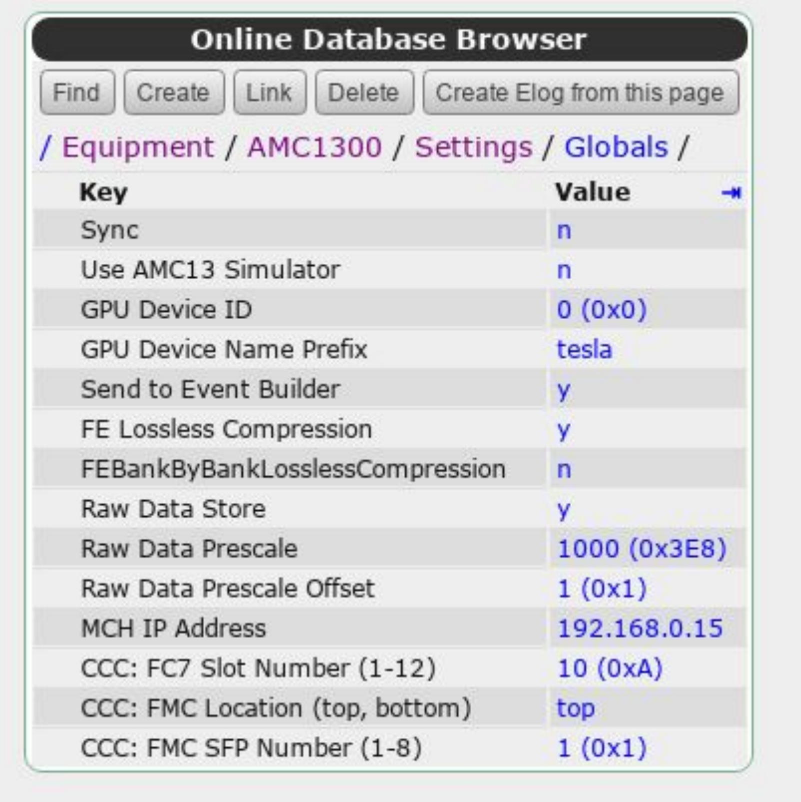


<table>
<tr><th></th><th>63</th><th>62</th><th>61</th><th>60</th><th>59</th><th>58</th><th>57</th><th>56</th><th>55</th><th>54</th><th>53</th><th>52</th><th>51</th><th>50</th><th>49</th><th>48</th><th>47</th><th>46</th><th>45</th><th>44</th><th>43</th><th>42</th><th>41</th><th>40</th><th>39</th><th>38</th><th>37</th><th>36</th><th>35</th><th>34</th><th>33</th><th>32</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>AMC13 Header</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="4">AMC #</td><td colspan="24">Trigger # [23:0]</td><td colspan="12">Timestamp [43:32]</td><td colspan="20">Data Length [19:0]</td></tr>
<tr><td>AMC13 Header</td><td colspan="5">CE [4:0]</td><td>L</td><td colspan="4">XADC</td><td>E</td><td colspan="5">TT [4:0]</td><td colspan="32">Timestamp [31:0]</td><td colspan="3">BT</td><td>A</td><td colspan="12">Board ID [11:0]</td></tr>
<tr><td>WFD5 Header</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Major Revision [7:0]</td><td colspan="8">Minor Revision [7:0]</td><td colspan="8">Patch Revision [7:0]</td></tr>
<tr><td>Channel Header</td><td>0</td><td>1</td><td colspan="4">XADC</td><td colspan="12">Channel Tag [11:0]</td><td colspan="22">Waveform Gap [21:0]</td><td colspan="12">Waveform Count [11:0]</td><td colspan="12">DDR3 Start Address [25:14]</td></tr>
<tr><td>Channel Header</td><td colspan="14">DDR3 Start Address [13:0]</td><td colspan="23">Waveform Length [22:0]</td><td colspan="3">TT</td><td colspan="24">Trigger Number [23:0]</td></tr>
<tr><td>Waveform Header</td><td colspan="12">Waveform Count [11:0]</td><td colspan="26">DDR3 Start Address [25:0]</td><td colspan="3">TT</td><td colspan="23">Waveform Length [22:0]</td></tr>
<tr><td>Waveform Header</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Channel Tag [11:0]</td><td colspan="22">Waveform Gap [21:0]</td><td colspan="12">Waveform Index [11:0]</td></tr>
<tr><td>WFD5 Data</td><td colspan="64">Waveform 1 ADC Data</td></tr>
</table>

Midas Demo

Online Database (ODB) [Demo backup]

- GUI on midas webpage
  - Also available command line
- Allows for “on the fly” adjustments between runs
- Built in configurations:
  - Midas webpage
  - Logger write location
  - Webpage update rate
  - etc.



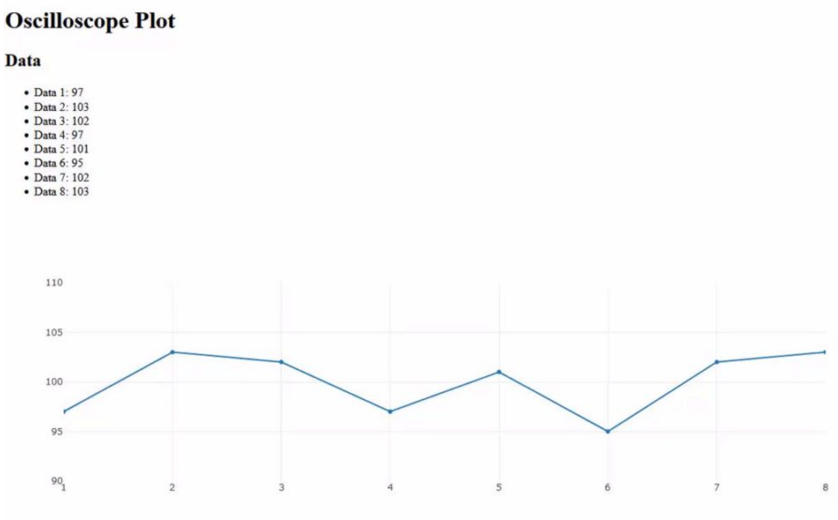
/ Equipment / AMC1300 / Settings / Globals /

Key	Value
Sync	n
Use AMC13 Simulator	n
GPU Device ID	0 (0x0)
GPU Device Name Prefix	tesla
Send to Event Builder	y
FE Lossless Compression	y
FEBankByBankLosslessCompression	n
Raw Data Store	y
Raw Data Prescale	1000 (0x3E8)
Raw Data Prescale Offset	1 (0x1)
MCH IP Address	192.168.0.15
CCC: FC7 Slot Number (1-12)	10 (0xA)
CCC: FMC Location (top, bottom)	top
CCC: FMC SFP Number (1-8)	1 (0x1)

Custom Software [Demo backup]

- Can write “clients” that connect to midas experiment
  - Python
  - C++

- Allows for user to write software to fit there needs, for example:
  - Data Quality Monitor
  - Offline Analysis
  - Automatic ODB management



Oscilloscope Plot

Data

- Data 1: 97
- Data 2: 103
- Data 3: 102
- Data 4: 97
- Data 5: 101
- Data 6: 95
- Data 7: 102
- Data 8: 103

Crude “proof of concept” DQM

Future Projects (Things We’re Working On)

- Ensuring UW machine has running DAQ before PSI beamtime

- Improve DQM framework to be more adaptable using midas, unpacking, and ZeroMQ libraries

- Direct communication between WFDs/FPGAs and CPU/GPU using PCIe communication
    - Avoids the need for μTCA crates
    - Speeds up data transfer rate (PCIe3x8 = 8GB/s = 64 Gb/s > 10 Gb/s)
    - Possibility for direct communication to GPU (faster data processing)