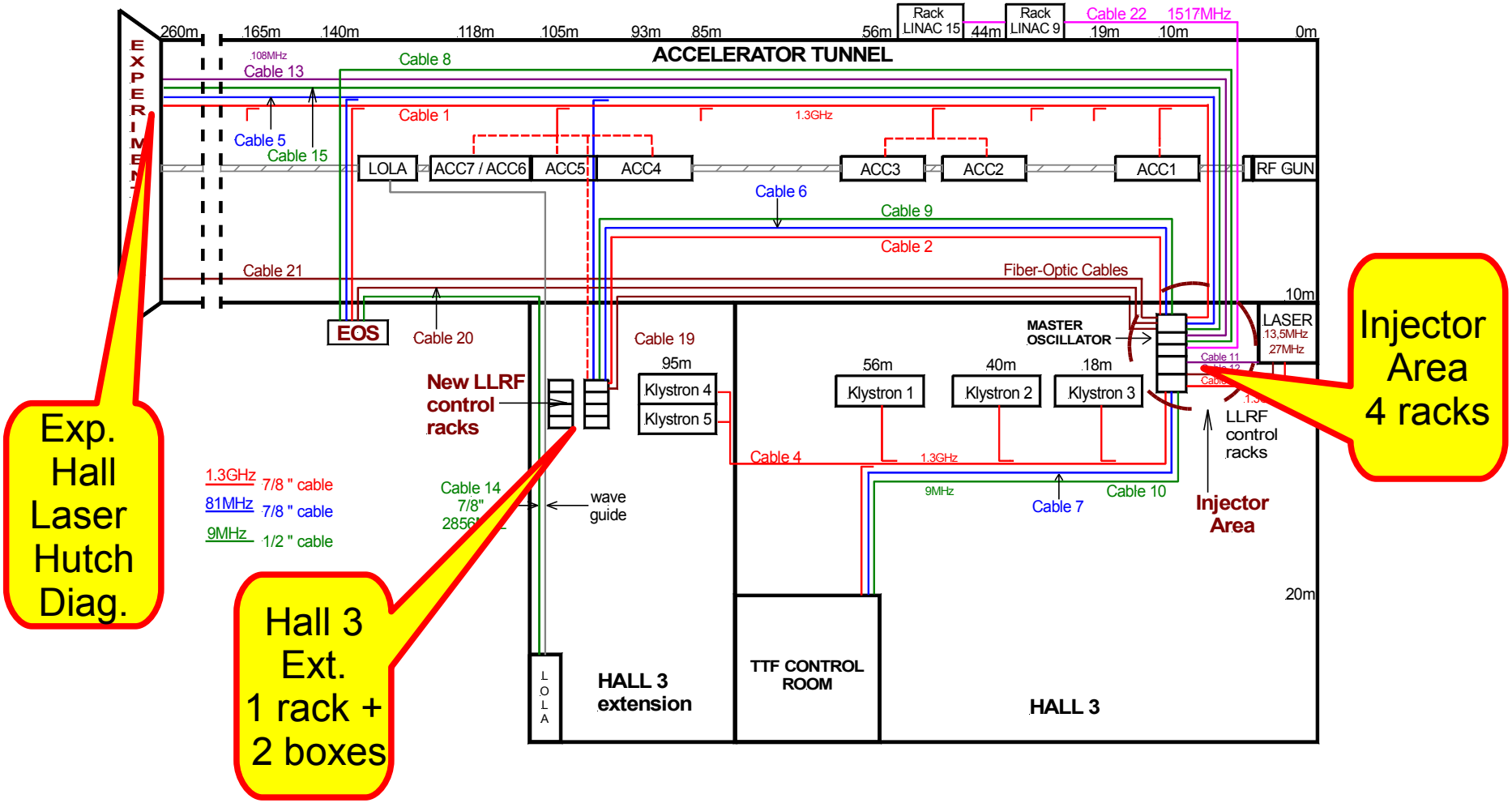
FLASH MO Diagnostics

Krzysztof Czuba
Piotr Więckowski

Master Oscillator – Overview and History

- From Simple box to a Master Oscillator System
- Generated frequencies:50 Hz, 1 MHz, 9 MHz, 13.5 MHz, 27 MHz, 81 MHz, 108 MHz, 1.3 GHz, 1.517 GHz, 2.856 GHz
- Complex distribution system with many tens of outputs
- Battery backed-up power supply
- Reliability required → supported by diagnostics with on-line readouts

MO System Layout

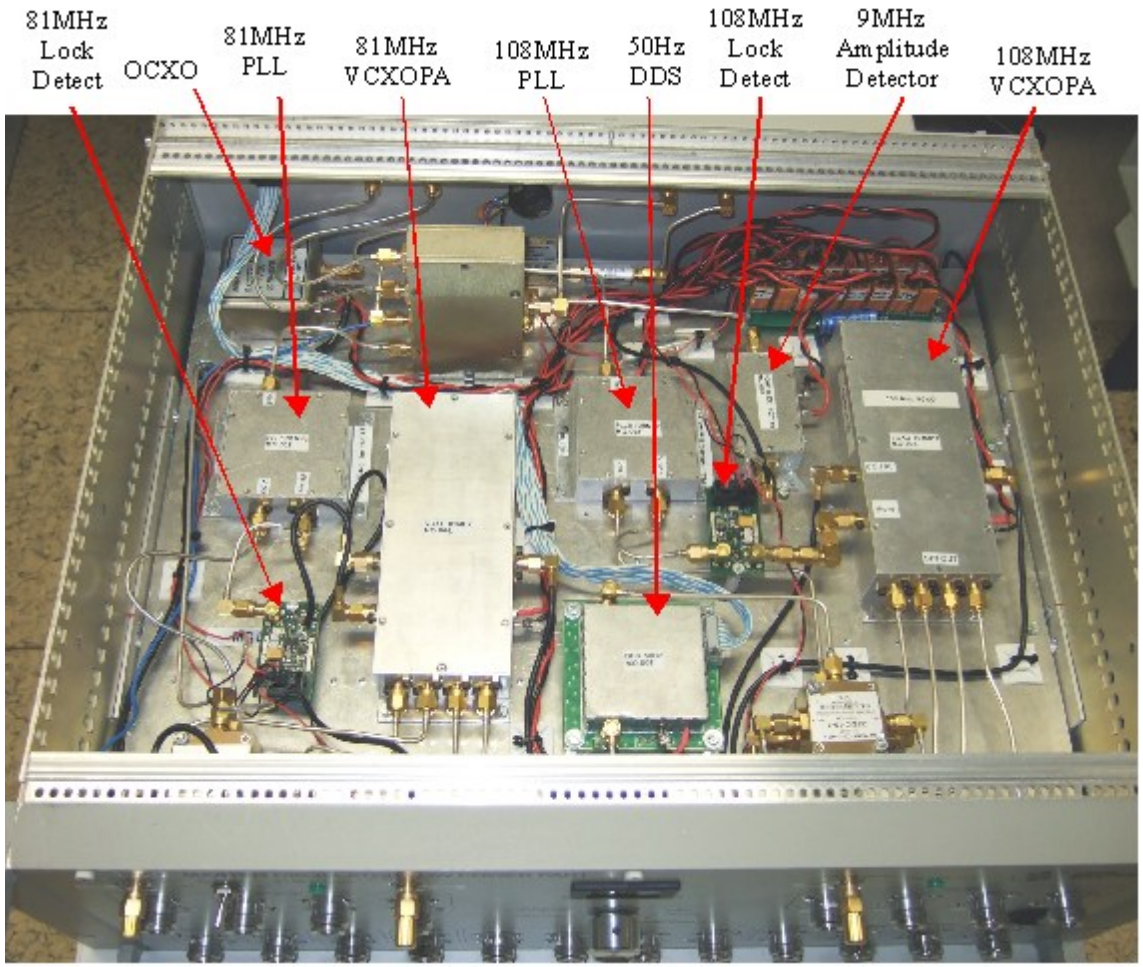


MO System Racks

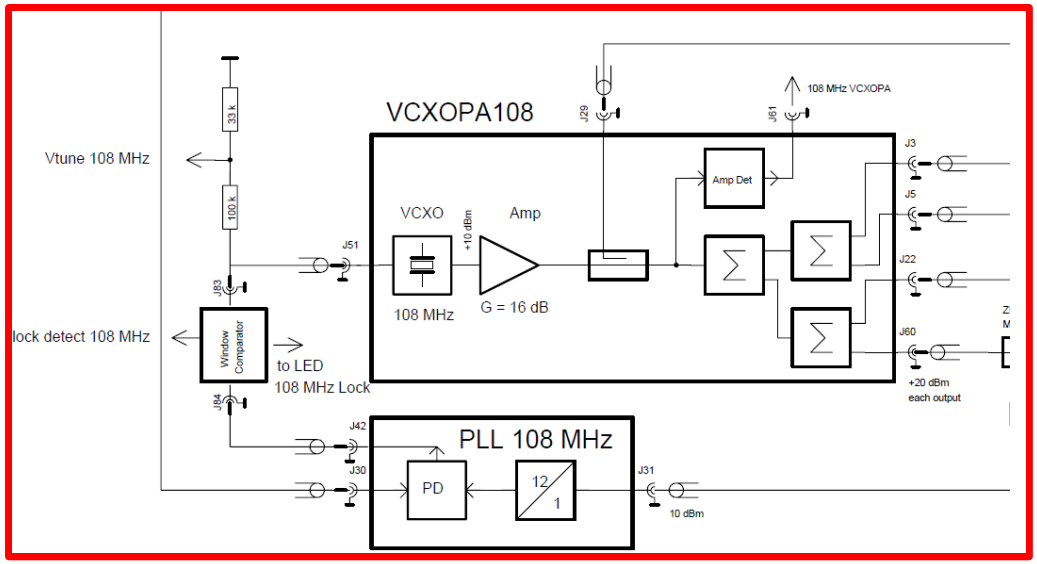
Injector Area

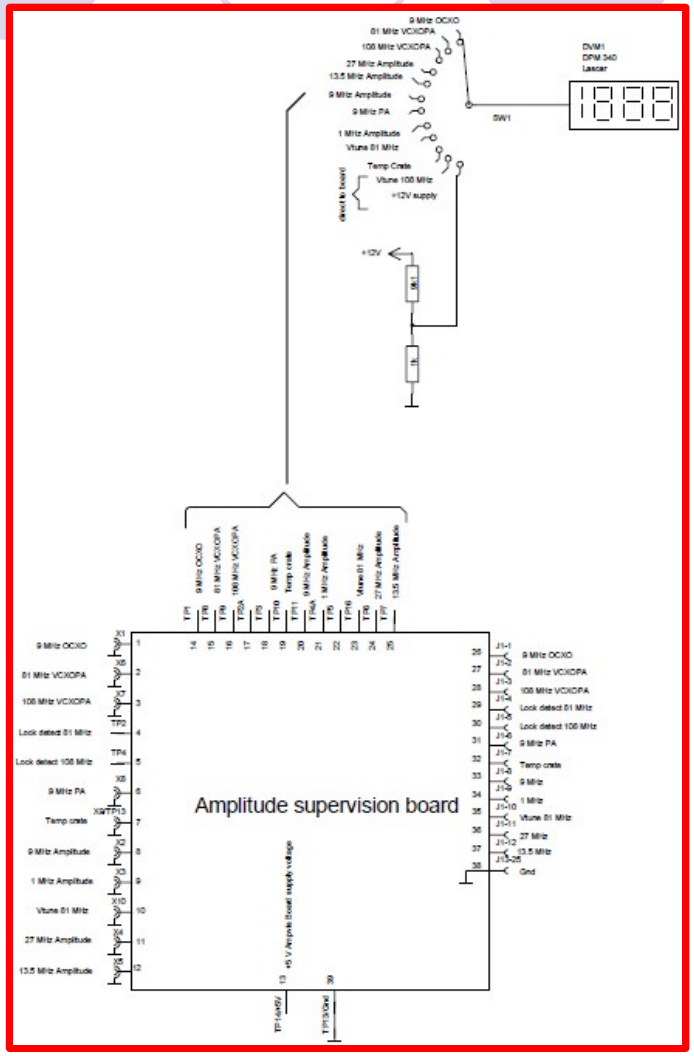
H3 Extension Subdistribution

MO Boxes – Integrated Diagnostics (1)



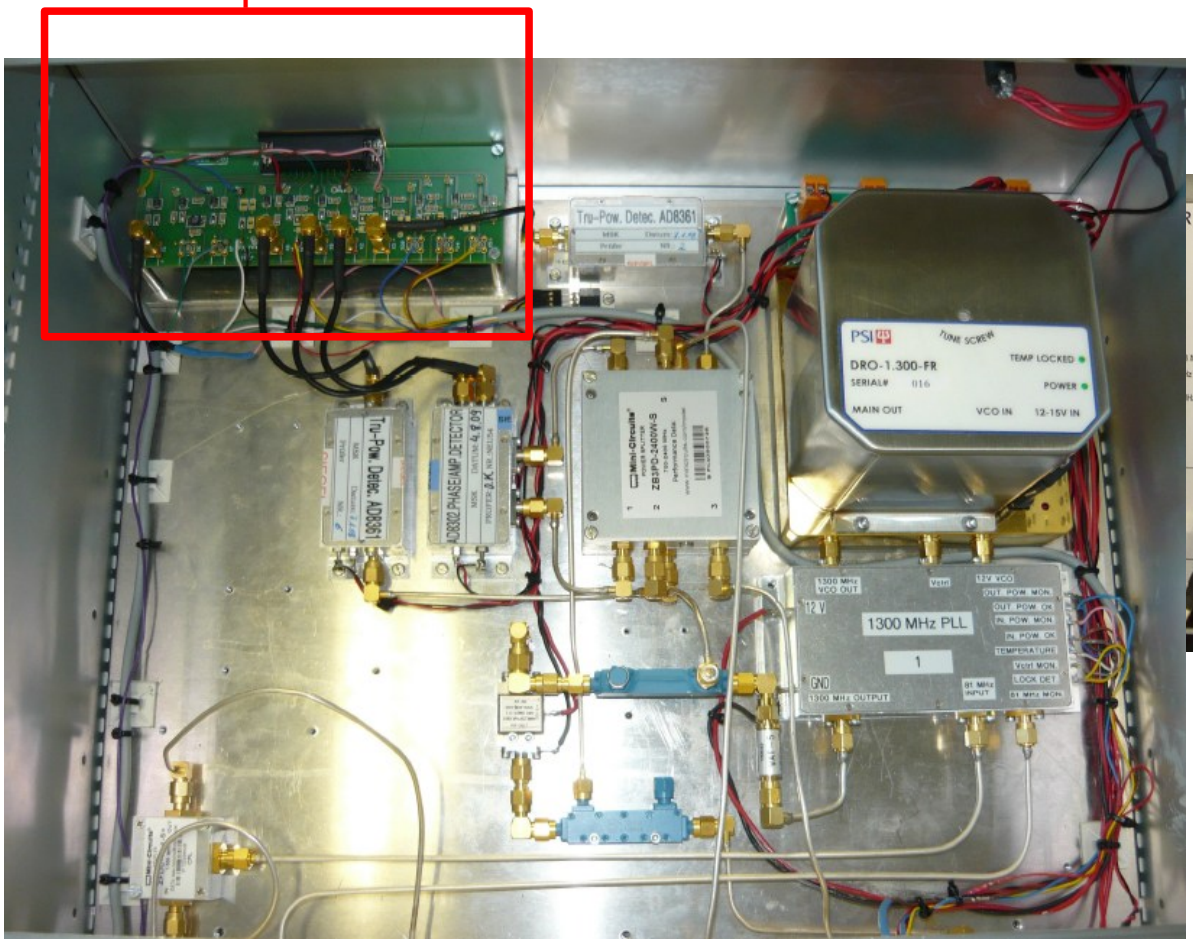
MO Boxes – Integrated Diagnostics (2)

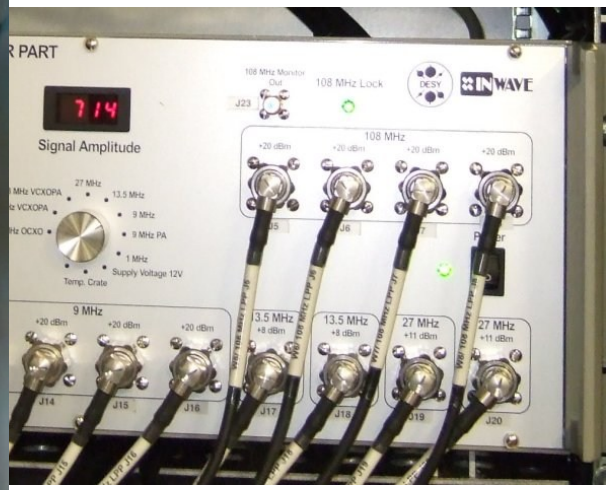




AmpVis Board and Front Panel Display

AmpVis Board



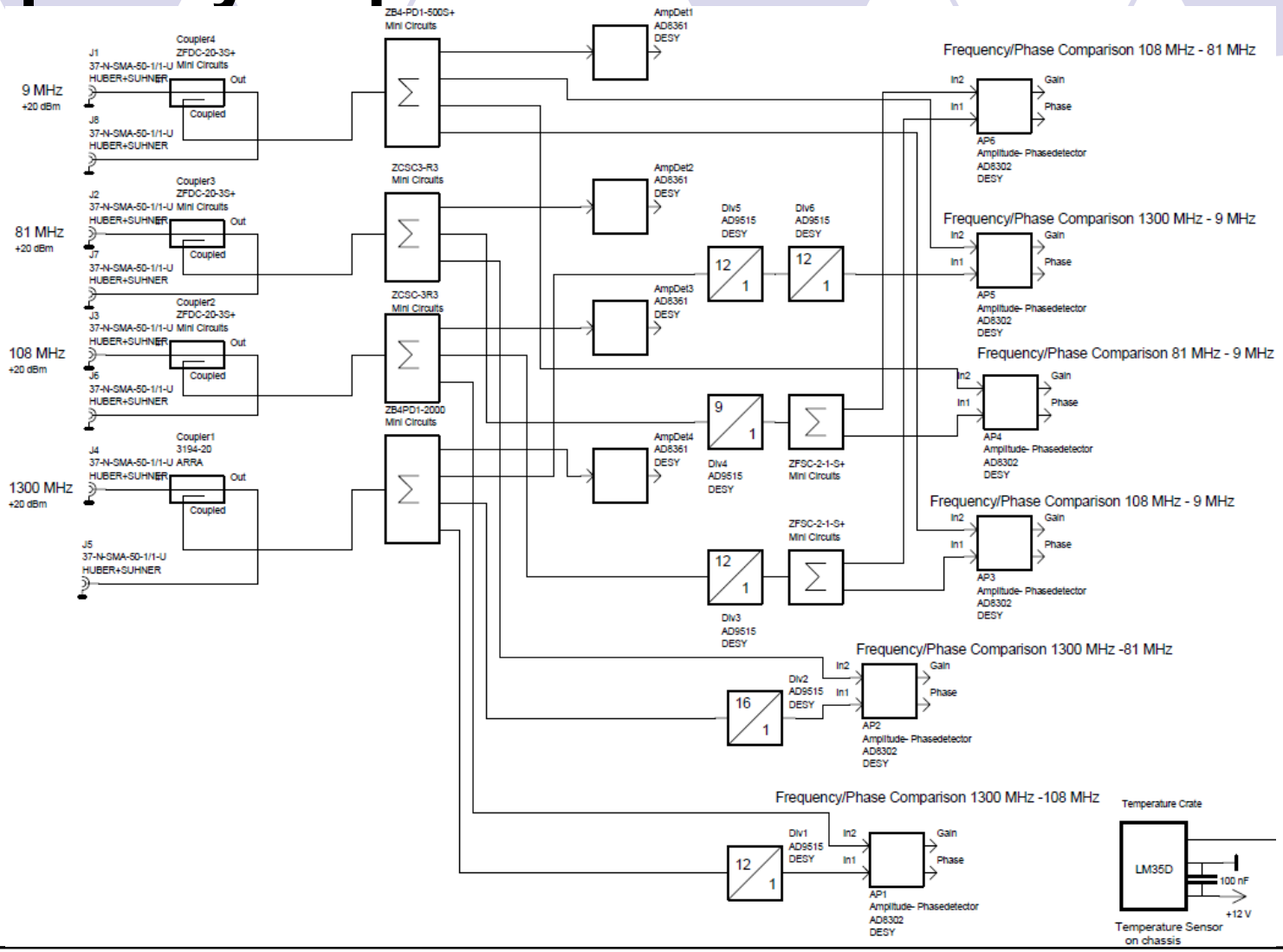


Diagnosed Parameters - Summary

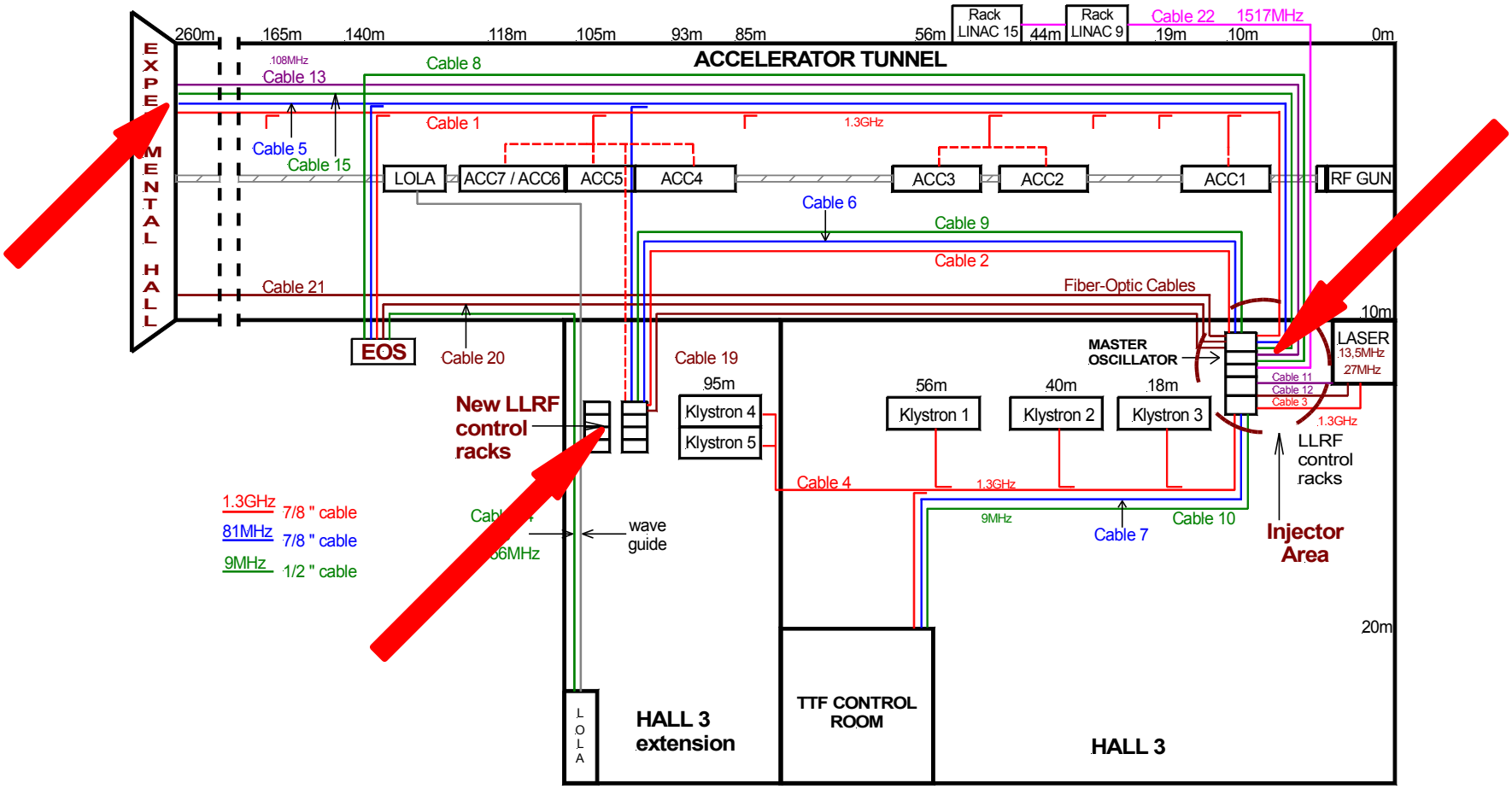
Measured are:

- Power levels
- Phase changes over amplifiers
- VSWR of most important cables
- Crate and component temperatures
- PLL lock status
- VCO control voltages
- Power supply voltages and currents
- Phase changes between various frequencies (Frequency Supervision Boxes)

Frequency Supervision Box

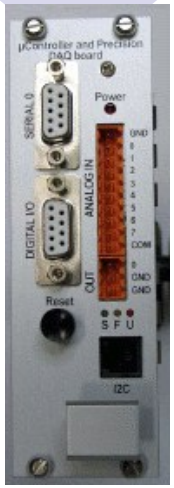


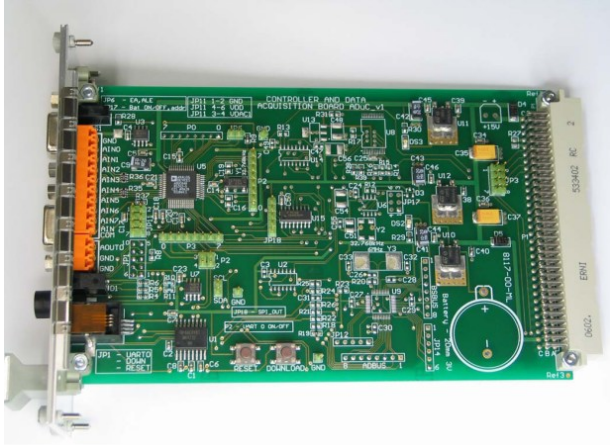
FSB Locations



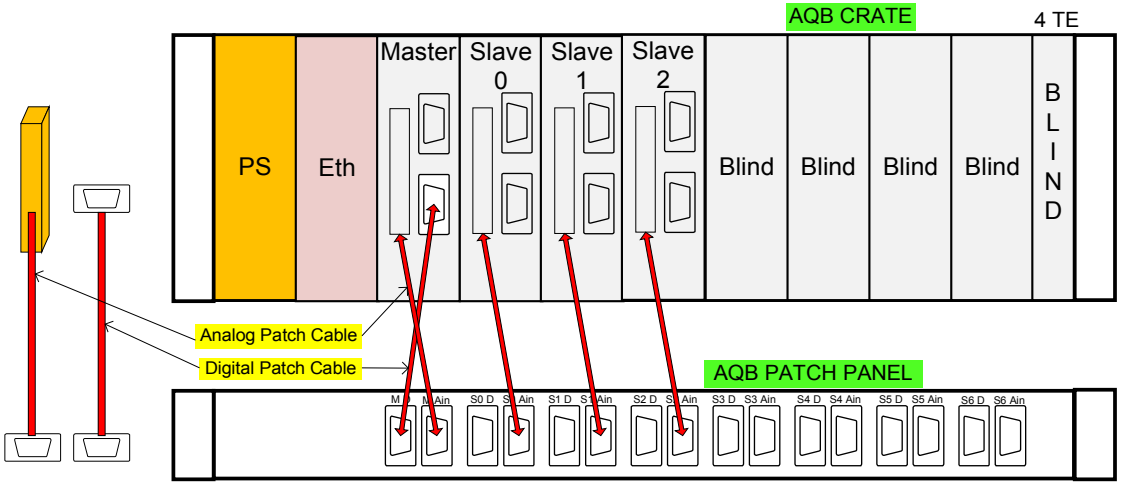
Diagnostic Data Acquisition Hardware

Data Acquisition PCB

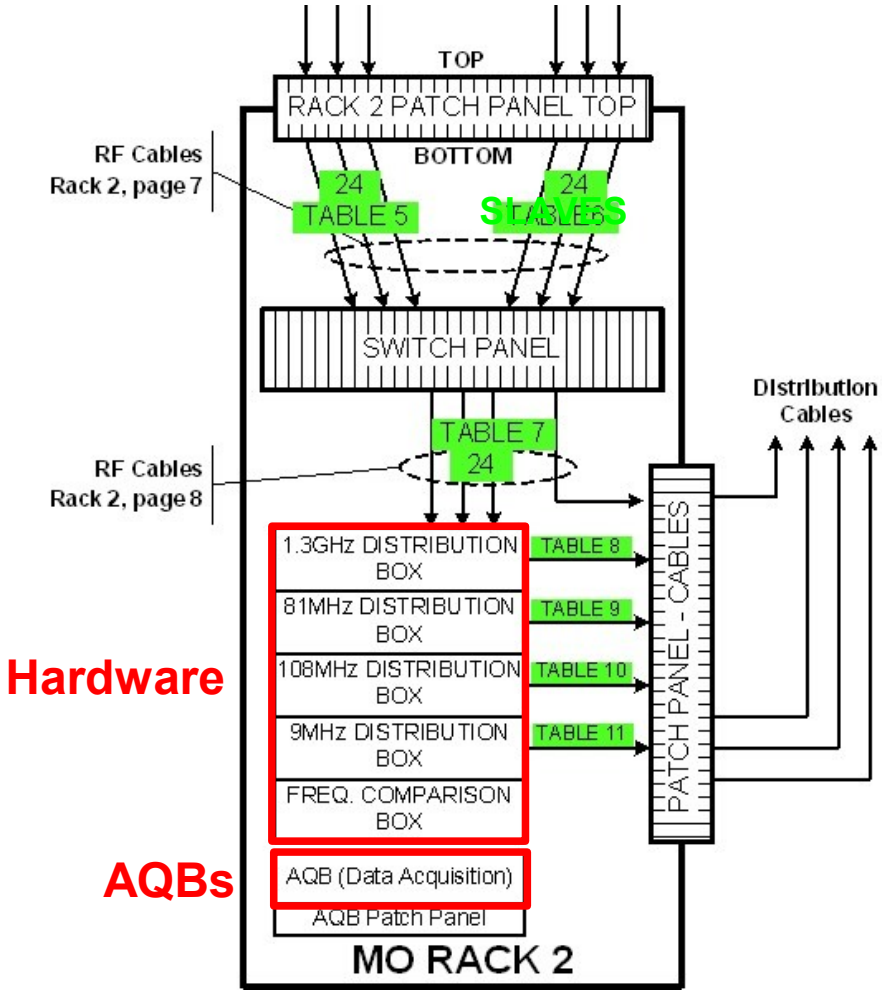




Monitorig Crate and Patch Panel



Acquisition boards (AQBs) - location



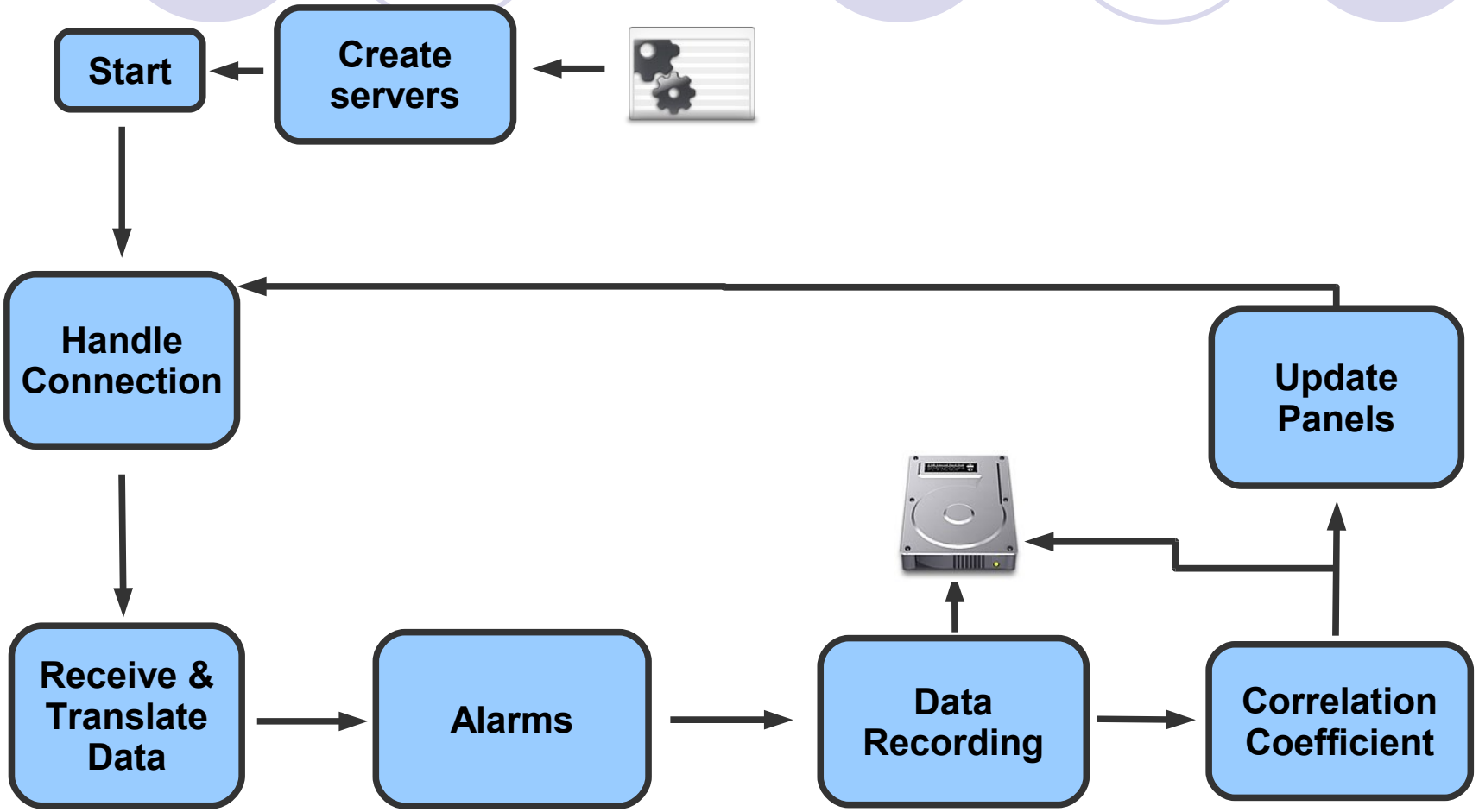
Acquisition boards (AQBs) - overview

Slaves: 0 to 7 modules; 8 x 16-bit channels analog data each

Master: 1 module; 8 x 16-bit channels analog data + 8-bit digital data

Ethernet module: communication

Server – main loop



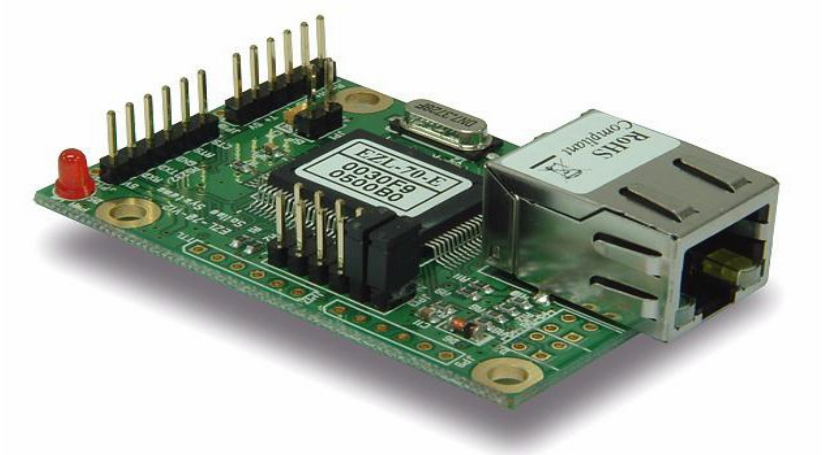
Server – creation process

- Configuration File:
  - One file – many servers
  - Connection parameters
  - All channels' attributes
    - Units
    - Signals' types
    - Signals' names
    - Alarm conditions

```
SLAVE0_AIN7.DESC:        "LPP Crate Temperature"
SLAVE0_AIN7.TYPE:        2
SLAVE0_AIN7.PARAM:       1  0  70  20
SLAVE1_AIN0:             0.483226
SLAVE1_AIN0.FILT:        1  0.001  0  0
SLAVE1_AIN0.EGU:         0 0 3 0 "V"
SLAVE1_AIN0.DESC:        "HPA1300 Transducer Gain"
SLAVE1_AIN0.TYPE:        1
SLAVE1_AIN0.PARAM:       1  0  2.5  0.5
SLAVE1_AIN1:             186.991
SLAVE1_AIN1.FILT:        1  0.01  0  0
SLAVE1_AIN1.EGU:         0 0 200 0 "deg"
SLAVE1_AIN1.DESC:        "HPA1300 Transducer Phase"
SLAVE1_AIN1.TYPE:        3
SLAVE1_AIN1.PARAM:       1  0  180  10
SLAVE1_AIN2:             1.19505
SLAVE1_AIN2.FILT:        1  0.001  0  0
SLAVE1_AIN2.EGU:         0 0 3 0 "V"
SLAVE1_AIN2.DESC:        "HPA1300 VSWR Amplitude"
SLAVE1_AIN2.TYPE:        1
SLAVE1_AIN2.PARAM:       1  0  2.5  0.5
SLAVE1_AIN3:             202.506
SLAVE1_AIN3.FILT:        1  0.01  0  0
SLAVE1_AIN3.EGU:         0 0 200 0 "deg"
SLAVE1_AIN3.DESC:        "HPA1300 VSWR Phase"
SLAVE1_AIN3.TYPE:        3
SLAVE1_AIN3.PARAM:       1  0  180  10
SLAVE1_AIN4:             2.26053
SLAVE1_AIN4.FILT:        1  0.001  0  0
SLAVE1_AIN4.EGU:         0 0 3 0 "V"
SLAVE1_AIN4.DESC:        "HPA1300 Output Amplitude"
SLAVE1_AIN4.TYPE:        1
SLAVE1_AIN4.PARAM:       1  0  2.5  0.5
SLAVE1_AIN5:             118.81
SLAVE1_AIN5.FILT:        1  0.01  0  0
SLAVE1_AIN5.EGU:         0 0 200 0 "deg"
SLAVE1_AIN5.DESC:        "PLL1300 Post Amplitude Phase"
SLAVE1_AIN5.TYPE:        3
```

Server – connection

- Connection:
  - TCP/IP
  - Two types of devices:
    - Ethernet to RS232
    - LAN/GPIB Gateway
  - Automatic reconnect
  - Reporting bad links



Server – receiving & translating data

- Up to 15 signal types per server:
  - Voltage
  - Temperature
  - Phase
  - Digital Data
  - Power
  - Voltage Standing Wave Ratio
  - ...

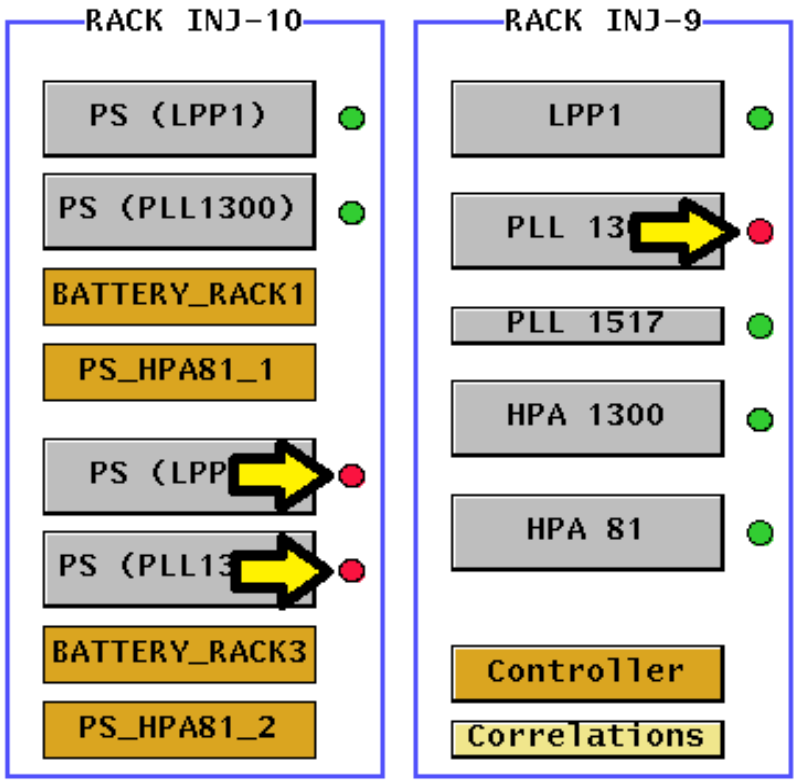
MO Rack 1/MO Rack 3

Cable 1 (LPP)

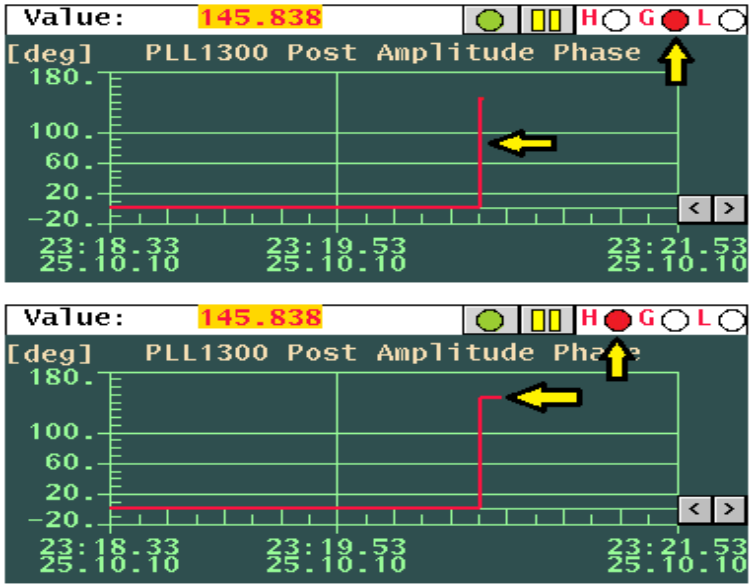
D-sub 25 Pin no.	D-SUB 9 Label	D-SUB 9 Pin no.	Controller Board	Controller Board Input	Signal Name
1	M Ain	1	Master	AIN0	9MHz OCXO Amplitude
2	M Ain	2	Master	AIN1	81MHz VCXOPA Amplitude
3	M Ain	3	Master	AIN2	108MHz VCXOPA Amplitude
4	M D	1	Master	D1 (AIN7)	81MHz PLL Lock
5	M D	2	Master	D2 (AIN6)	108MHz PLL Lock
6	M Ain	6	Master	AIN5	9MhzPA Amplitude
7	S0 Ain	8	Slave0	AIN7	Crate Temperature
8	M Ain	8	Master	AIN7	9Mhz Amplitude

Server – alarms

- Global alarm
  - 1 alarm per device

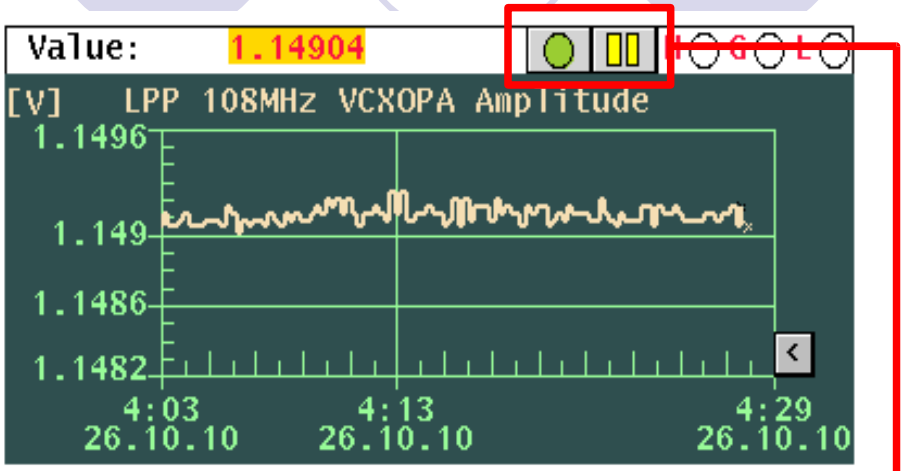


- 3 types of signal alarm
  - Value too low
  - Gradient too big
  - Value too high



Server – signals – data recording

- Recording system:
  - Handling multiple files simultaneously (no limit)
  - On demand
  - Raw data for external processing
  - Long recording period possible (weeks/months)
  - Pause in recording possible

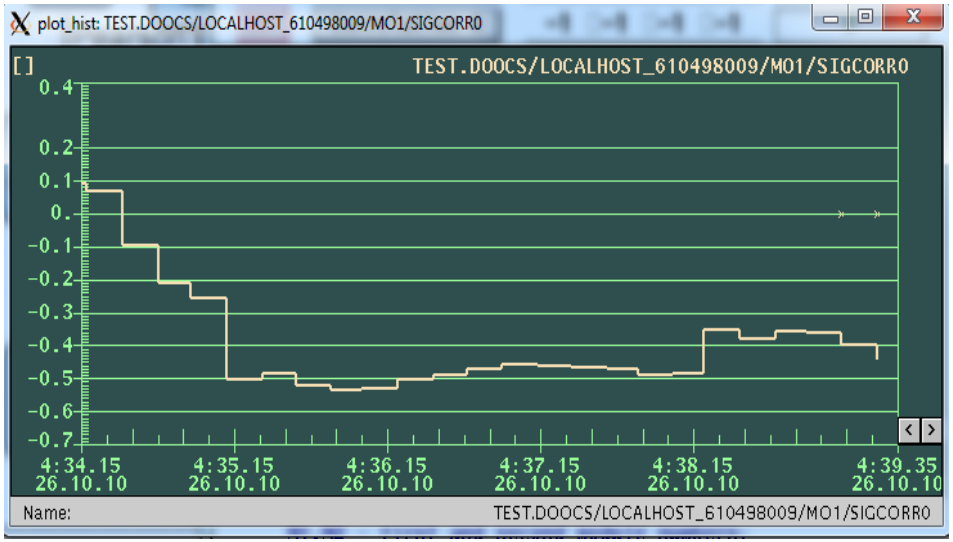


1000	0.366981089115	Tue Sep 21 13:16:52 2010
1001	0.367019265890	Tue Sep 21 13:16:57 2010
1002	0.366981089115	Tue Sep 21 13:17:05 2010
1003	0.366981089115	Tue Sep 21 13:17:10 2010
1004	0.366942912340	Tue Sep 21 13:17:15 2010
1005	0.366981089115	Tue Sep 21 13:17:20 2010
1006	0.366981089115	Tue Sep 21 13:17:29 2010
1007	0.367019265890	Tue Sep 21 13:17:33 2010
1008	0.366981089115	Tue Sep 21 13:17:38 2010
1009	0.366981089115	Tue Sep 21 13:17:43 2010

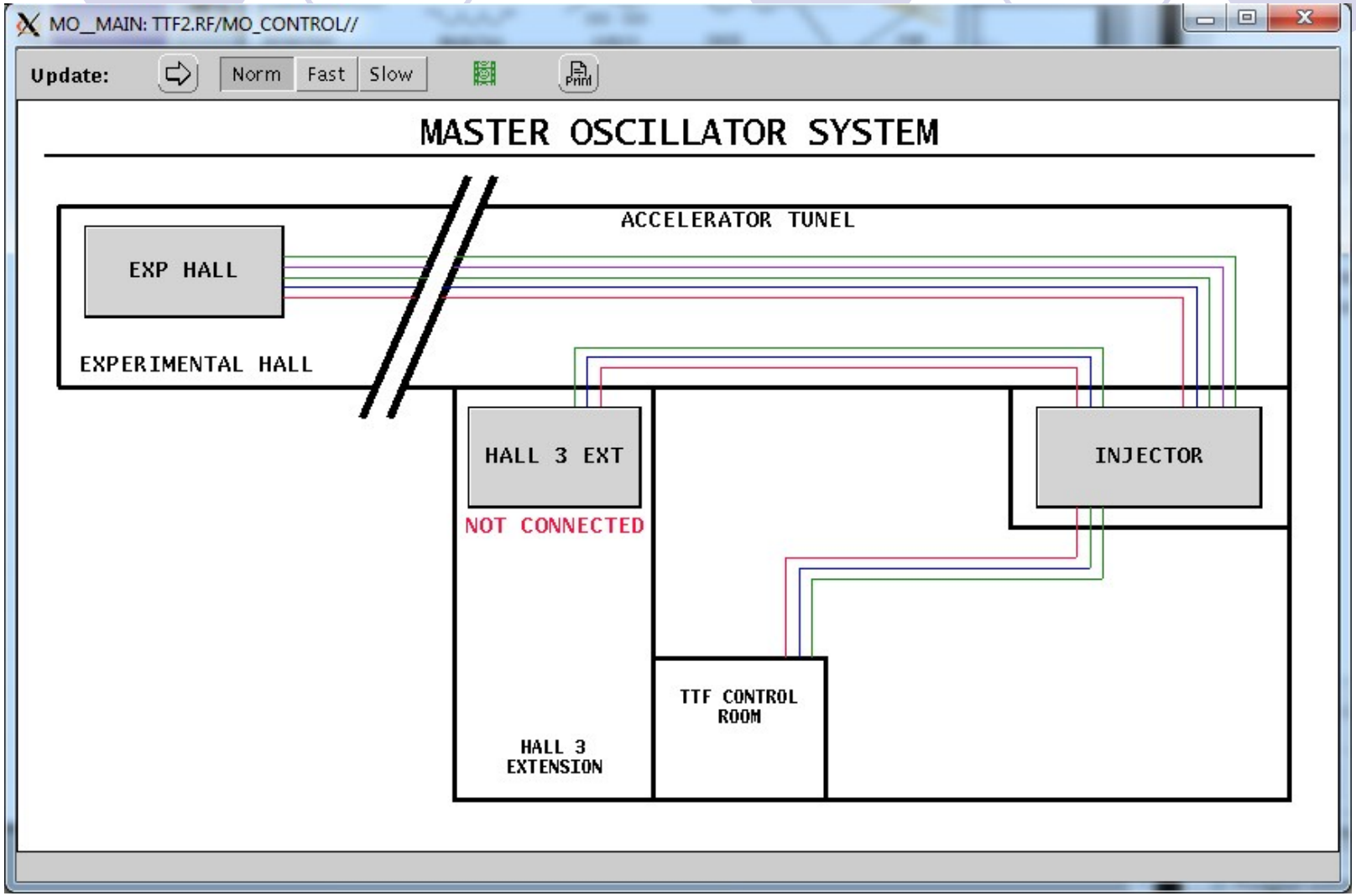
Server – signals – correlations

- Correlations system:
  - Pearson's correlation coefficient for up to 15 signal pairs per server
  - On demand recording
  - Long recording period possible (weeks/months)

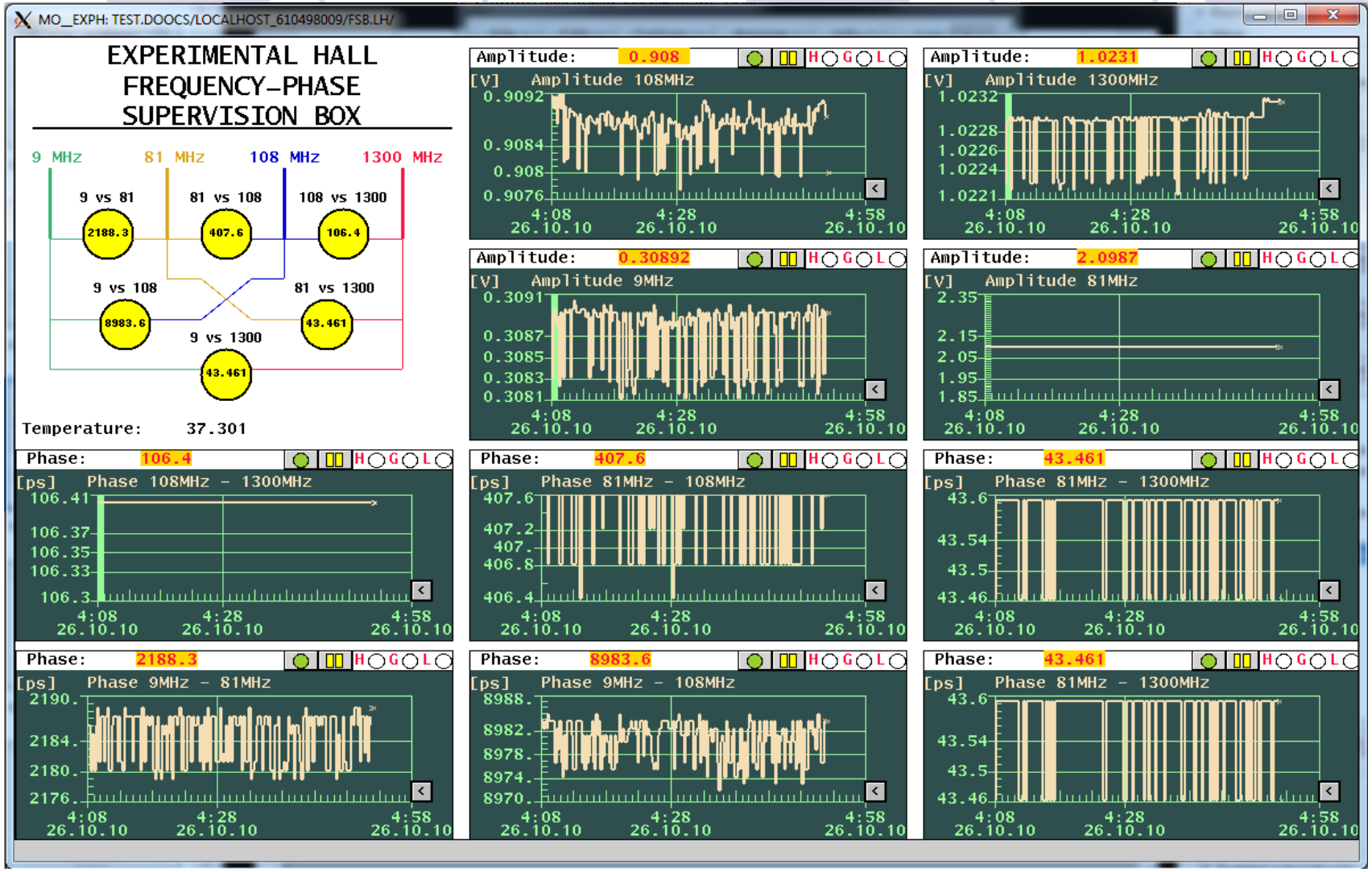
$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - (E(X))^2}\,\sqrt{E(Y^2) - (E(Y))^2}}.$$



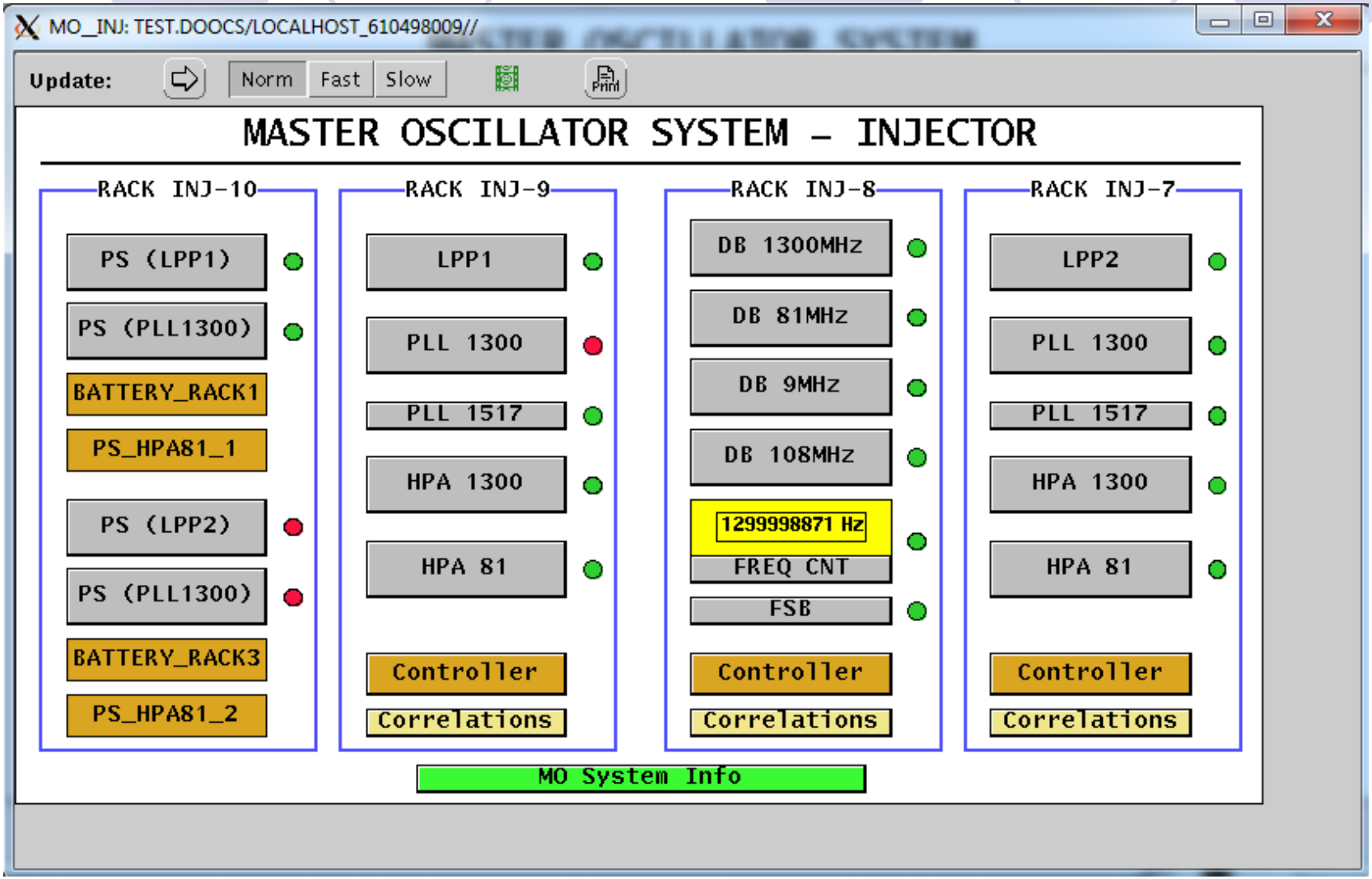
Server – user interface – main panel



Server – user interface - FSB



Server – user interface - injector



Work status

- In progress:
  - Hall 3 Ext & Master Oscillator 2 connections to AQB
- To do:
  - Adjust filters and alarms
  - Hall 3 Ext. user panels
  - Remote reset signals

```
{
    char _sSlave[7]="SLAVE";
    const char _sDigits[11]="0123456789";
    /*** create needed modules (MO_module class) ***/
    p_Modules = (MO_module**)malloc(_code*sizeof(MO_module*));
    p_Modules[0] = new MO_module("CTRL_DIG",this);
    p_Modules[1] = new MO_module("MASTER",this);
    printf("FCT_CODE: %d\n",_code);
    for(int i=2;i<_code;++i)
    {
        memcpy(_sSlave+5,&_sDigits[i-2],1);
        printf("%s\n",_sSlave);
        p_Modules[i] = new MO_module(_sSlave,this);
    }
    _trig = _code-2;
    if (!conf_done)
    {
        list_append();
        conf_done = 1;
    }
}

void    EqFctMO::init ( )
{
    char    HistFile[80];
    // Init history
    sprintf( HistFile, "%s/DATA_%s.hist", "hist", name_str() );
    arch  = new ArchiveFile( HistFile,  "mo_channels", NOINDEX );
    contr_dio_hist_.book_hist( arch, name_str(), ring_buffer, memory_buffer );

    for(int i=0;i<_trig+2;++i)
    {
        p_Modules[i]->setArchive(arch);
    }
    MOenet_.enet_init( arch, name_.value() );
    online();
    reconnect_cnt_ = 0;
}
```

Thank you for your attention