Libera

Libera Base and How It Communicates to EPICS Server

Peter Leban on behalf of SW team, June 15, 2010, Hsinchu Taiwan

Instrumentation Technologies

Contents of the talk

- **Libera BASE: background, roles, advantages**
- **Interfaces**
- **Libera EPICS CA server**
- **Mapping between Libera BASE and EPICS CA server with example**
- **References**

Basic Application
Support Environment

Libera

Libera BASE narrows the gap between your HW and the machine Control System

- **Easy start-up and control**
- **Rapid prototyping**
- **Rapid application development**
- **Connection of the instrument with the Control System**
- **Overall system relieability assurance**

Libera

Libera BASE – background

- **Many instruments, many people → control systems matrix → complex support**
- **Leverage development between different solutions**
- **Common building blocks (HW monitoring, data streams, FPGA artifacts)**

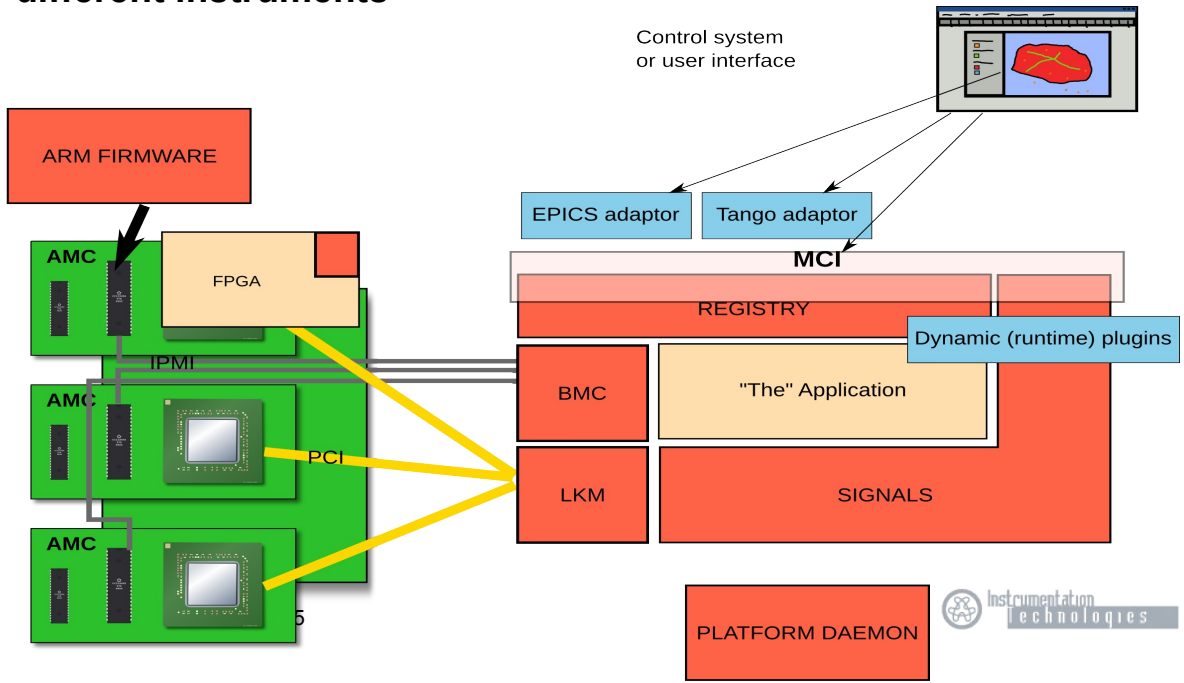
Instrument || Controls	EPICS	Tango	DOOCS / TINE	Matlab	WEB	Twitter client
Libera Brilliance+	1	2	3	4	5	6
Libera Single Pass H	7	8	9	10	11	12
Libera LLRF	13	14	15	16	17	18

Libera

Libera BASE – roles

- **Hardware abstraction (Platform B, µTCA, µTCA4physics, PCI express...)**
- **Common denominator for different instruments**
- **Middle layer**
- **Development framework**
- **External interface**

Focus in the application!



Instrumentation Technologies

Libera

Libera BASE – advantages

- **Various HW technologies (CPU capacity, processor modules, FGOFB module)**
- **Reconfigurable instrument (add-on modules)**
- **Easy to learn (use cases, examples)**
- **Application and Algorithm Sharing**

  Example:
  - Model the function in the Matlab
  - Use the signals from Libera BASE
  - Integrate the function into application software using SDK

Libera

Interfaces

- **i-reg and i-sig**
- **Platform management based on IPMI**
- **MCI – external API**
  - **List / dump all registry nodes**
  - **Info – attributes**
  - **Get / Set – value access**
  - **Listen – notification**
  - **Acquire – stream and data on demand**

```
// Get reference to the SA signal from the registry

ireg::Node root = GetRoot();
ireg::Node sa_signal = root.GetNode({"boards", "raf", "signals", "sa"});

// Acquire the signal (1024 samples)

isig::SignalSharedPtr signal =
ireg::SignalNode::CreateRemoteSignal(sa_signal);
typedef isig::RemoteStream<isig::SignalTraitsVarInt16> SaStream; SaStream *strm = dynamic_cast<SaStream*>
(signal.get());

SaStream::Client cl(strm, "my-client");
SaStream::Buffer a(cl.NewBuffer(1024));

isig::SuccessCode ret = cl.Read(a);

// and dump the acquired signal

for(auto i = 0; i < a.GetLength(); i++) {
    for (size_t j = 0; j < a[i].GetComponents(); ++j)
        std::cout << setw(11) <<  a[i][j] <<  "  ";
    std::cout << std::endl;
}
```


Instrumentation Technologies

Libera

Implementation in the instrument (Libera Brilliance+)

APPLICATION SPECIFIC

- Processor modules
- Timing module
- Fast orbit feedback module

PLATFORM & GENERIC

- Health (fans, voltages, T°)
- Hardware identification
- Common (prefixes)

- Control parameters
- Signal access

Libera

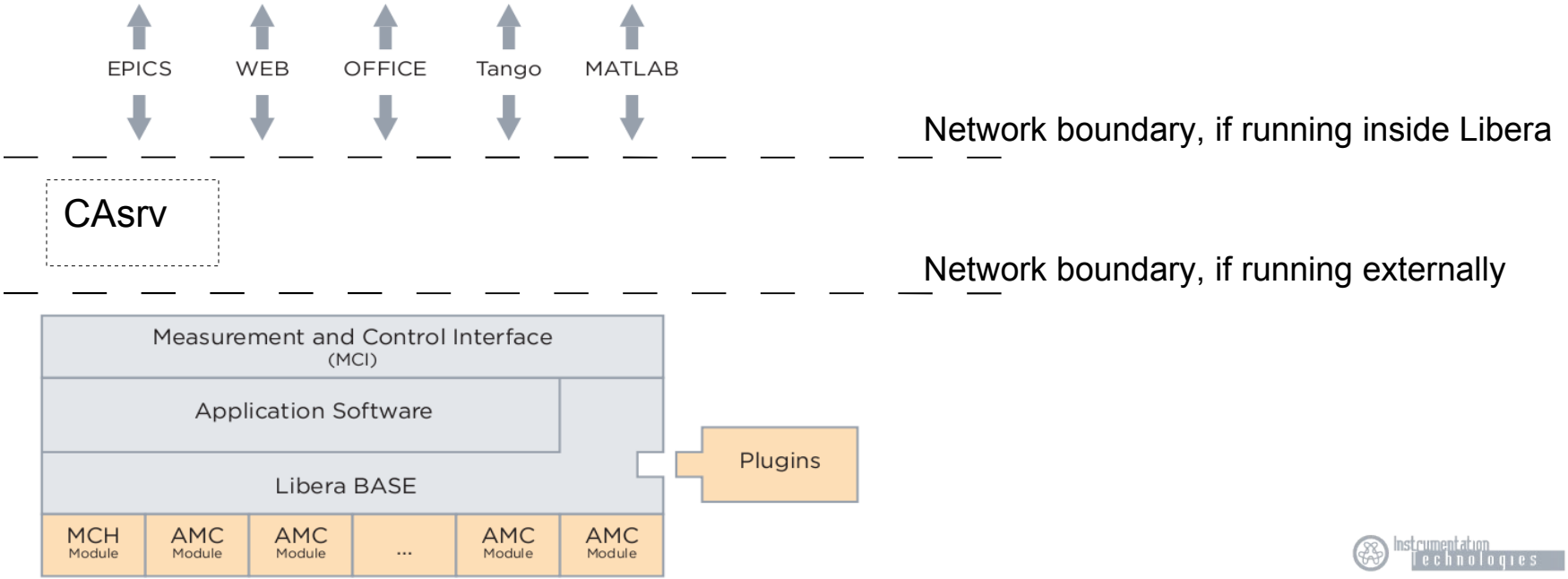
Channel Access server

- **Libera uses EPICS base 3.14.12**
- **CA server library included**
- **Developers' opinion: very useful, easy-to-use C++ code**
- **CA server runs inside Libera instrument (e.g. Brilliance+)**
- **Benefits:**
  - **Runs on standard Linux distribution (Ubuntu), currently 10.04 LTS, kernel 2.6.38**
  - **Modular COMe, not limited to embedded processors: from Atom to i7 Intel processors**
  - **Supports various instrument's configuration (modules)**
  - **Configurable buffer lengths**

Libera

Where does Channel Access server run

- **Runs inside the Libera instrument … but could run externally:**



Instrumentation Technologies

Libera

Libera EPICS CA srv configuration

- **General: boards / modules**
- **Board specific**
- **Health specific**

```
root@libera:/var/opt/libera/cfg/epics# ll
total 76
drwxr-xr-x 2 root root  4096 2011-06-03 15:42 ./
drwxr-xr-x 3 root root  4096 2011-06-03 15:38 ../
-rwxr-xr-x 1 root root   383 2011-06-02 11:41 epics_appCommon.xml*
-rwxr-xr-x 1 root root   944 2011-06-02 11:41 epics_appCommon.xsd*
-rwxr-xr-x 1 root root  2051 2011-06-02 11:41 epics_common_types.xsd*
-rwxr-xr-x 1 root root  3089 2011-06-02 11:41 epics_health.xml*
-rwxr-xr-x 1 root root  2447 2011-06-02 11:41 epics_health.xsd*
-rwxr-xr-x 1 root root   885 2011-06-02 11:41 epics_local.xml*
-rwxr-xr-x 1 root root  1096 2011-06-02 11:41 epics_local.xsd*
-rwxr-xr-x 1 root root 12385 2011-06-02 11:41 epics_raf.xml*
-rwxr-xr-x 1 root root  6020 2011-06-02 11:41 epics_raf.xsd*
-rwxr-xr-x 1 root root  2099 2011-06-02 11:41 epics_tim.xml*
-rwxr-xr-x 1 root root  2669 2011-06-02 11:41 epics_tim.xsd*
-rwxr-xr-x 1 root root  2589 2011-06-06 07:54 epics.xml*
-rwxr-xr-x 1 root root  3605 2011-06-02 11:41 epics.xsd*
```

Libera

Mapping Libera BASE <--> EPICS CAsrv

- **Registry mapping to EPICS CA server configuration**
- **All nodes can be assigned to PVs**
- **All properties are transferred to EPICS CA server automatically:**
  - **Units**
  - **Parameter limit values**
  - **Alarm low/high**
  - **Type integer/double**
- **Dynamic PV creation**
- **Configurable PV names, linked to registry path**

Libera

Libera Base – example of SA data strem

- **Info about SA data stream**

```
root@ubuntu-libera:/# libera-ireg info boards.raf3.signals.sa
-------------------------------------------------
   Registry hostname    : IP_127-0-0-1
   Node name            : sa
   Value                : No value for this node
   Value type           : Undefined
   Validator expression :
   Num of values        : 0
   Full path            : boards,raf3,signals,sa
   Num of children      : 6
   Flags                : readable signal
   Root                 : false
   Parent node name     : signals
   Children             : access_type, atom_size, group_size, components_names, components_number, components_type
-------------------------------------------------
```

```
    sa
      access_type=Stream
      atom_size=64
      group_size=1
      components_names=Va,Vb,Vc,Vd,Sum,Q,X,Y,LMT_l,LMT_h,r1,r2,r3,r4,r5,status
      components_number=16
      components_type=int32
```

Libera

CA server configuration

- **example of SA data stream access**

```
<!-- SA streams -->
<streams prefix="SA" length="1">
    <stream prefix="VA_MONITOR" path="sa" sigType="Va"></stream>
    <stream prefix="VB_MONITOR" path="sa" sigType="Vb"></stream>
    <stream prefix="VC_MONITOR" path="sa" sigType="Vc"></stream>
    <stream prefix="VD_MONITOR" path="sa" sigType="Vd"></stream>
    <stream prefix="Q_MONITOR" path="sa" sigType="Q"></stream>
    <stream prefix="SUM_MONITOR" path="sa" sigType="Sum"></stream>
    <stream prefix="X_MONITOR" path="sa" sigType="X"></stream>
    <stream prefix="Y_MONITOR" path="sa" sigType="Y"></stream>
</streams>
```

```
caget LIBERA:BPM1_SA_VA_MONITOR
caget LIBERA:BPM1_SA_VB_MONITOR
caget LIBERA:BPM1_SA_X_MONITOR
caget LIBERA:BPM1_SA_Y_MONITOR
```

Libera

References

- **EPICS CA server runs currently in Libera Brilliance+:**
  - **Taiwan Light Source for TPS (NSRRC, Taiwan)**
  - **Advanced Photon Source (ANL, USA)**
  - **Petra-III (DESY, Germany)**
- **EPICS CA server runs currently in Libera LLRF at EMMA (UK)**
- **To be added soon: Libera Single Pass H**

Libera

Thank you for your attention.

support@i-tech.si

peter.leban@i-tech.si