An IRMIS Extension: cRIO Wiring Information System

Kanglin Xu

May 15, 2017
Contents

• APS IRMIS brief
• LANSCE-RM device database system
  • Development environment and toolkits
  • LANSCE IRMIS and extended subsystems
  • Screenshots of implementation of IRMIS at LANSCE
• LANSCE cRIO wiring information system
  • cRIO: CompactRIO system
  • Development environment for NoSQL database
  • System statistics
  • IRMIS schemas associated with the wiring system
• Future work
IRMIS 2 was developed by Argonne National Lab. Today it covers many aspects of EPICS control systems including IOCs, PVs, cables, models, etc.

- IRMIS 2 basically has
  - RDB modules and schema
  - PV crawler for loading PVs from database
  - GUI Application by Java
- Stable and used at APS
- But NOT supported by IRMIS community any more
LANSCE-RM Device Database System is a site IRMIS implementation to install, manipulate, track, and search the data and information for LANSCE control systems.

- IRMIS 2 Schema and tables
- IRMIS 2 PV Crawler
- Application based upon REST – Representational State Transfer
- More schema other than those covered by IRMIS
- More applications and features
The LANSCE-RM Device Database System is developed by using many popular open source software, tools and frameworks.

- **LAMP** open source software packages
  - Refer to the first letters of **L**inux, **A**pache, **M**ySQL and **P**HP, **P**ERL or **P**ython
- **Python Django** framework for the server side
  - Pure python
  - MVC pattern
  - Free admin interface and user authentication ready
- **SQLAlchemy** SQL toolkit for database interface
- **jQuery** javascript library for the front-end client side and AJAX implementation
  - jQuery GUI framework for some special UI requirements
  - jQuery Mobile framework used for developing mobile-friendly pages
- **Elasticsearch** to provide powerful full-text search sever
The system has many more extensions and sub-systems for the LANSCE control system other than a local and site IRMIS implementation.

- The PV system including the schema and crawler based upon IRMIS 2 posted on the EPICS web site as mentioned.
  - PV search and report generator
  - PV history including those changed and deleted
- LANSCE wire scanner information system
  - Wire scanner and actuator general info like locations, chassis, IP, etc.
  - Configuration and initial parameters
- LANSCE daily log and archive management system
  - Auto archiving to save operators’ work and time
  - Full-text search for historic data
- LANSCE switch and network device system
- Legacy RICE channels system for timed/flavored data
Screenshots of implementation of IRMIS at LANSCE: PV Search Page - Similar to the APS IRMIS 2 Java Application

Tips: (1) wildcards "*" or "?" are allowed in the text fields to match anything. If the pattern itself has "*" or "?", use \"*\" or \"?\" instead. (2) For the channel text field, you are allowed to input more than one patterns separated by lines, that is, one line has one pattern.
Screenshots of implementation of IRMIS at LANSCE: PV search result based upon the previous slide – similar to the APS 2 Java application.

Ajax call to avoid to change the URL and refresh the whole page, and get each PV details easily and quickly.

Click the Record row to get the detailed FIELD/VALUE pairs.

39 PVs Match the previous search pattern
Screenshots of implementation of IRMIS at LANSCE: PV REPORT TEMPLATE – Dynamically Generate a PV Report with FIELDS at Users’ Choice

Users can select up to 8 field types to be included in the following report.

Other than FIELD types, IOC list, Rec Type list and text patterns are similar to PV search.
Screenshots of implementation of IRMIS at LANSCE: PV report generated dynamically

- Default Field Types
- Users-selected field types at the previous slide
- Can sort the results based on the FIELD clicked at both ASC and DESC directions
- DJANGO pagination used
- Ajax calls to get the complete FIELD/VALUE pairs at the next slide and to send you an XLS spreadsheet file created dynamically.
Screenshots of implementation of IRMIS at LANSCE: PV report generated dynamically - Continued

Click the radio button of 01JC001L05 to get the its field pairs in the grey box; And then click “Channel Info” button, we can get the white pop-up dialog with detailed info. Both are AJAX calls without leaving and refreshing the page.
The LANSCE cRIO wiring information system is designed as a subsystem and extension of LANSCE IRMIS device database system.

• To share wiring terminal codes between our hardware team, software team, operators and other users.
• To centralize the wiring information rather than each one has his own spreadsheet.
• To make the wiring information update easy.
• To save our engineers’ time and effort for troubleshooting and maintaining our control system.
• To associate and map with LANSCE IRMIS Device Database System and find information like
  • PV and record field values
  • IOC and its location
  • DB, DBD file locations, etc.
The cRIO, a product of National Instruments (NI), combines a real-time controller chassis, reconfigurable IO modules, and FPGA module.

- **cRIO Hardware and software**
  - Power PC, Intel Atom processor, and Intel Core i7 processor
  - DDR memory and nonvolatile storage.
  - Ethernet ports and RS232 serial ports
  - VxWorks for free with hardware or NI Linux Real-time with embedded UI and Mini Displayport
  - LabVIEW, a NI graphical programming language, for programming the hardware and particularly for coding FPGA

- **EPICS integrated via NI CA**
- **Embedded EPICS**
  - A Special Board Support Package (BSP) required
  - Additional configuration required
The cRIO wiring information system is developed and built on MongoDB, a NonSQL document-oriented system and other open source toolkits.

- **MongoDB 3.2.8** for data model rather than SQL schema because of
  - Fast development
  - Simple data model
  - Flexibility and no complex joins
  - Scalable architecture but the feature not really used in our site
  - A first step to upgrade our IRMIS system to a NonSQL system

- **Pymongo-3.3.0**
  - A Python distribution containing tools for working with MongoDB

- **mongo-c-driver, mongo-cxx-driver and swig**
  - To build C/C++ library to talk to MongoDB and an interface wrapper for our existed TCL applications

- **Other toolkits and frameworks mentioned above for IRMIS on our site**
Although this wiring information system is pretty simple, it does store important information for our daily work.

- IOC name
- cRIO Module name
- Module slot
- Terminal code
- PV name
- cRIO module channel
- Wire label
- Wire color
- IO item label

```
> use wireinfo
switched to db wireinfo
> db.stats()
{
    "db" : "wireinfo",
    "collections" : 1,
    "objects" : 4754,
    "avgObjSize" : 222.11022297013042,
    "dataSize" : 1055912,
    "storageSize" : 356352,
    "numExtents" : 0,
    "indexes" : 1,
    "indexSize" : 86016,
    "ok" : 1
}
```
We associate this system with our IRMIS device database system to query the PVs’ field values needed for control development and troubleshooting.

• To do so, we first join IRMIS tables including Record Names, Record Types, IOC names, Field Names, Field Values, etc.
• And then query and filter by PV name
Clicking the radio button of PV 21TM001L01 and then clicking “Channel Info” button, we can get its field values.

![Screenshot of LANSCE wiring information system and applications](image)

**Tips:**
1. You can click a table row to edit it and after editing the row, you click another row to make changes to be displayed.
2. Don’t forget to click Save Updated button to save changes before you leave this page.
3. You can click Save Updated button any time, after you change either a single row or multiple rows. Clicking this button would not leave this page.
4. By clicking a table header, you can sort the records with their respective columns either by ascending order or by descending order in toggle.
5. To prevent losing your input data, we suggest you log in before you make any changes.

---

### Table: LANSCE Device Database System

<table>
<thead>
<tr>
<th>ch loc</th>
<th>slot</th>
<th>node</th>
<th>terminal</th>
<th>record</th>
<th>CRIO CHN</th>
<th>Wire Label</th>
<th>Wire Color</th>
<th>IO Item Label</th>
<th>Other Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>m217a</td>
<td>5</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21XP001L03</td>
<td>D02</td>
<td>001:3:0014</td>
<td>Bn, Rd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21XP001L04</td>
<td>D03</td>
<td>004:9:0050</td>
<td>Bn, Rd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L01</td>
<td>D04</td>
<td>006:7:0068</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L02</td>
<td>D05</td>
<td>006:1:0062</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L03</td>
<td>D06</td>
<td>007:2:0073</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L04</td>
<td>D07</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L05</td>
<td>D08</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L06</td>
<td>D09</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L07</td>
<td>D10</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L08</td>
<td>D11</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L09</td>
<td>D12</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L10</td>
<td>D13</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L11</td>
<td>D14</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L12</td>
<td>D15</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m217a</td>
<td>3</td>
<td>m9477</td>
<td>T01: B5: J1: 1: 11</td>
<td>21TM001L13</td>
<td>D16</td>
<td>012:5:0051</td>
<td>Rd, Bn</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total records:** 4754 **Ordered by record # ( )**
Future Works

- New PV database implementations based on NoSQL at LANSCE
  - New crawler for NoSQL PV database
  - Applications based upon the backend
- IOC information system based on NoSQL
  - New crawler for IOC information system
  - Related applications
- Add more data like locations and rack numbers into the wiring information system