Micro/Nano-satellites
On-board Software Framework Design
and
Its Implementation in Hodoyoshi Satellites

Jun’ichi Takisawa, Shinichi Nakasuka
(ISSL, The University of Tokyo)
Sotaro Kobayashi, Tran Ngoc Lan Huong, Shinichi Kimura
(Tokyo University of Science)
Outline

- Background
- Overview of the Framework
- Driver Design and Cross-Platform Capability
- Cmd/TIm Handling Application
- Verification Platform
- Conclusions
Background
Evolutions of Micro/Nano-Satellites

Feasibility Study
Ex. XI-IV(Launch 2003)
XI-V(Launch 2005)

Technology Demonstration
Ex. PRISM(Launch 2009)

Practical Mission
Ex. Nano-JASMINE
(Launch 2013, Planned)

Advanced Missions and Longer Lifetime
Requires Higher Reliability
Problems and Strategy

- How to realize higher reliability with
  - Low development cost
  - Short development period

- Key Strategy = Reuse of satellite components
  - Proven components increase reliability directly
  - Reuse reduces many tests
    and realizes low cost and short development period
  - Reuse enables feedback from past operation
    and realizes continuous improvement of reliability

Components = Hardware and also Software
Hardware Side Status

- Many proven components and knowledge are accumulated from past missions
- Many Micro/Nano-satellite components companies form a sustainable supply chain
Software Side Status

- Insufficient knowledge accumulation
  - Each satellite has each independent software

Software should also absorb hardware and mission differences in each satellite.

In order to realize reuse-oriented software, systematic framework is really needed

Completely Different
On-Board Software Framework
Overview of the Framework
Overview of the Framework

Core System
- Mode Management
- Transition Management
- Task Management

User Side
- Transition Sequence
- Middleware
- Driver
- Application
- Task List
S/W design flow on this Framework

- Requirement Definition
- Mode Definition
- Task List Definition
- Application Definition
Example of H3 Mode Definition

Initial Sequence → Safe Mode → Arming → Standby Mode

- S/C Sep.
- Cmd
- Abort to Safe
- Finish Checkout
- Low Bat. Voltage
- Enter to Checkout
- Check Out Mode
**Example of H3 TaskList**

**Definition**

### Execution Timing

<table>
<thead>
<tr>
<th>Step</th>
<th>Top Level</th>
<th>GS Packet Handler</th>
<th>Interval 500</th>
<th>AOBC Packet Handler</th>
<th>TLM Handler</th>
<th>SHU TLM</th>
<th>Debug Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CCSDS Rx Analyzer</td>
<td>B2 Rx</td>
<td>EPS TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>Screen</td>
</tr>
<tr>
<td>1</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>WDB CLEAR</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>⚪ Cycle Tag</td>
</tr>
<tr>
<td>2</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>HRM TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>⚪ WDB</td>
</tr>
<tr>
<td>3</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>AD590 Phase 1</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>HRM</td>
</tr>
<tr>
<td>4</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>STRx TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>CMD</td>
</tr>
<tr>
<td>5</td>
<td>CCSDS Rx Analyzer</td>
<td>B0 Tx</td>
<td>TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>TLM</td>
</tr>
<tr>
<td>6</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>UVC</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>PCU</td>
</tr>
<tr>
<td>7</td>
<td>CCSDS Rx Analyzer</td>
<td>B3 Rx</td>
<td>SHU Packet Handler</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>BPOU</td>
</tr>
<tr>
<td>8</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td></td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>MPDU</td>
</tr>
<tr>
<td>9</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td></td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>STRx</td>
</tr>
<tr>
<td>10</td>
<td>CCSDS Rx Analyzer</td>
<td>B2 Rx</td>
<td>EPS TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>AD590 ch6-3</td>
</tr>
<tr>
<td>11</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>CSAS TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>AD590 ch3-7</td>
</tr>
<tr>
<td>12</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>DC Rx</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>AD590 ch6-6</td>
</tr>
<tr>
<td>13</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>AD590 Phase 2</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>AD590 chc-1</td>
</tr>
<tr>
<td>14</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>STRx TLM</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>CSAS P-Side</td>
</tr>
<tr>
<td>15</td>
<td>CCSDS Rx Analyzer</td>
<td>B2 Tx</td>
<td></td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>CSAS M-Side</td>
</tr>
<tr>
<td>16</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td>Heater Control</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>UVC</td>
</tr>
<tr>
<td>17</td>
<td>CCSDS Rx Analyzer</td>
<td>B3 Rx</td>
<td>SHU Packet Handler</td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td>Heater Ctrl</td>
</tr>
<tr>
<td>18</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td></td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>CCSDS Rx Analyzer</td>
<td>⚪</td>
<td></td>
<td>B0 Rx Packet Analyzer</td>
<td>Gen CCSDS Tx</td>
<td>B3 Tx</td>
<td></td>
</tr>
</tbody>
</table>

Each applications
Driver Design and Cross-Platform Capability
Layered Design of Driver Software

Layered Structure → Flexibility against H/W Differences

Application Layer
- Packet Handling
- CMD Handling
- TLM Gen.
- S/W UVC
- Heater Control

Driver Layer
- PCU
- Comm.
- STT
- RW

Middleware Layer

Hardware Layer

Driver/Application I/F
Absorb Component Differences

Middleware-Driver I/F
Absorb Platform Differences
Drivers Cross-Platform Capability

- Hodoyoshi-3 OBC

- BoCCHAN-1
Cmd/Tim Handling Application
Problems on Cmd/Tlm Software

- Command and telemetry code are likely to be added or modified during the satellite development process.
- On-board software must correspond not only to mission design but also to the operation database of the ground system.
Automatic Code Generation System

Satellite Design Documents

Telemetry & Command Code Generator
- Auto-generated Source Code
  - On-board Software

Ground System Database Generator
- Auto-generated Database
  - Operation Software

Direct Connection
Actual Generation System in Hodoyoshi-3

Ground System Database Generator

Direct Connection

On-board Software
Verification Platform
## Software Verification Steps

<table>
<thead>
<tr>
<th>Test configuration</th>
<th>① MILS (Model In-the-Loop Simulation)</th>
<th>② SILS (Software In-the-Loop Simulation)</th>
<th>③ HILS (Hardware In-the-Loop Simulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Things to test</td>
<td>Control Logic, mode transition</td>
<td>C code</td>
<td>C code on OBC, Sensor/actuator drivers</td>
</tr>
<tr>
<td>Tools</td>
<td>MATLAB/Simulink</td>
<td>MATLAB/Simulink, I/F soft</td>
<td>S/W: MATLAB/Simulink, LabVIEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H/W: AD converter, RS422-USB</td>
</tr>
</tbody>
</table>
Configuration of HILS
Conclusions
Conclusions

- Practical micro/nano satellite requires higher reliability than usual micro/nano satellites and “Reuse” is a key strategy to solve this problem with low cost and short development period.
- To establish reuse oriented satellite software development environment, we design “On-bard Software Framework”.
- The development of satellite software based on that framework is in progress under Hodoyoshi project, and its first results have been used for Hodoyoshi-3 & 4.
- Accumulation of software developed by many satellite developers based on the same framework is now being organized in UNISEC (university)