Project Olympus OpenBMC

Ali Larijani - Microsoft
Sanjoy Maity - AMI
• Microsoft Adoption of OpenBMC

• Microsoft OpenBMC contribution Roadmap

• Project Olympus OpenBMC

• Next Step
OpenBMC Adoption

- Adopted 1.5 Years ago as a result of Microsoft commitment to fully realize open H/W platform
- Targeted @ on low volume project Olympus H/W platform with moderate manageability
- Successfully customized to ship on 3 different Project Olympus H/W platform
- Created a community among Microsoft Partners
• Extended, Optimized and Shipped on Storage platform
• F/W update: Dual flash and Signing support
• Redfish: Storage profile
• SMBus for satellite controllers
• PMBus for PSU monitoring
• Fan Speed Control
• SOL/SSH
Project Olympus OpenBMC roadmap

- Project Olympus openBMC workgroup
- Redfish/Server profile
- IPMI/KCS/SSIF
- IPMB
- LED/Power Control
- Power Capping
- Remote media
- Remote Debug
- Secure boot
- Host tools
Technical Challenges

• To achieve a long-term supported stable codebase for product delivery
  • Marinating a Long Term Support codebase is necessary for successful product delivery

• To achieve a robust, production-worthy stack for the OCP platforms
  • Defining the QA metrics for the Long Term Supported codebase for real product delivery

• To address Security/Vulnerability advisories
  • Defining the Security Analysis (Vulnerability tests, PEN tests, Certifications etc.) Metrics
  • Support for all hardware security implementations for all platform architectures (HROT, Image Signing Process etc.)
AMI’s commitment to OpenBMC & OCP Platforms

LINUX FOUNDATION
OPENBMC PUBLIC TREE

PLATFORM ENABLEMENT
LTS BRANCH
MERGED PERIODICALLY

OPEN SOURCE BUILD SCRIPT
OPEN SOURCE FINAL IMAGE

MAINTAINING OCP PLATFORMS

OPENBMC LTS
OPEN SOURCE MAINTAINED BY AMI AND COMMUNITY

OCP COMMUNITY CONTRIBUTIONS OPENSOURCE
PROJECT RECIPES

CONTRIBUTIONS

OCP PLATFORM ENABLEMENTS
CORE CONTRIBUTIONS – SOC SUPPORT
KVM, VMEDIA
SECURITY
AMI’s contributions for Project Olympus

- **KCS Interface**
  - KCS Device Driver
  - KCS Interface Service

- **Serial Over LAN**
  - IPMI based SOL
  - Can be used with standard ipmitool – Ex: ipmitool -H <BMC IP> -I lanplus sol activate

- **I2C over LAN (Bridge)**
  - Read/Write I2C device over LAN interface
  - Can be used with standard ipmitool

- **IPMI LAN Interface**
### AMI’s contributions for Project Olympus

- **Chassis Power Usage**
  - Control Chassis power over LAN and KCS Interface
  - Standard ipmitool can be used for both KCS and LAN interface

- **Platform Sensor support:**

<table>
<thead>
<tr>
<th>Sensor Name</th>
<th>Sensor Value</th>
<th>Units</th>
<th>Status</th>
<th>LNR</th>
<th>LC</th>
<th>LNC</th>
<th>UNC</th>
<th>UC</th>
<th>UNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp_Outlet</td>
<td>23.00</td>
<td>degrees C</td>
<td>ok</td>
<td>na</td>
<td>5.000</td>
<td>10.000</td>
<td>60.000</td>
<td>80.000</td>
<td>na</td>
</tr>
<tr>
<td>Temp_2U_SensorB</td>
<td>21.00</td>
<td>degrees C</td>
<td>ok</td>
<td>na</td>
<td>5.000</td>
<td>10.000</td>
<td>60.000</td>
<td>80.000</td>
<td>na</td>
</tr>
<tr>
<td>P3V3</td>
<td>3.800</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>2.500</td>
<td>3.000</td>
<td>3.500</td>
<td>4.000</td>
<td>na</td>
</tr>
<tr>
<td>PSV</td>
<td>5.100</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>3.000</td>
<td>4.200</td>
<td>5.500</td>
<td>6.000</td>
<td>na</td>
</tr>
<tr>
<td>P12V_AUX</td>
<td>12.400</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>11.400</td>
<td>11.700</td>
<td>12.500</td>
<td>13.000</td>
<td>na</td>
</tr>
<tr>
<td>P1V05_STBY_PCH</td>
<td>1.060</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>0.700</td>
<td>0.900</td>
<td>1.200</td>
<td>1.500</td>
<td>na</td>
</tr>
<tr>
<td>PVNN_STBY_PCH</td>
<td>0.910</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>0.600</td>
<td>0.800</td>
<td>1.200</td>
<td>1.400</td>
<td>na</td>
</tr>
<tr>
<td>P3V3_STBY</td>
<td>3.300</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>2.500</td>
<td>3.000</td>
<td>3.500</td>
<td>4.000</td>
<td>na</td>
</tr>
<tr>
<td>PSV_STBY</td>
<td>5.100</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>4.000</td>
<td>4.500</td>
<td>5.500</td>
<td>6.000</td>
<td>na</td>
</tr>
<tr>
<td>P3V_VBAT</td>
<td>0.100</td>
<td>Volts</td>
<td>ok</td>
<td>na</td>
<td>2.500</td>
<td>2.800</td>
<td>3.300</td>
<td>3.700</td>
<td>na</td>
</tr>
</tbody>
</table>
### AMI’s contributions for Project Olympus

#### Supported IPMI Commands (both KCS & LAN Interface):

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Device ID</td>
<td>raw 0x6 0x1</td>
</tr>
<tr>
<td>Get Chassis Capabilities</td>
<td>raw 0x0 0x0</td>
</tr>
<tr>
<td>Get/Set Lan configuration</td>
<td>lan print 1</td>
</tr>
<tr>
<td>Get SEL Information</td>
<td>raw 0xa 0x40</td>
</tr>
<tr>
<td>Get SEL Time</td>
<td>0xa 0x48</td>
</tr>
<tr>
<td>Chassis Control Command</td>
<td>chassis power &lt;on/off/cycle/reset/status&gt;</td>
</tr>
<tr>
<td>Get SDR Information</td>
<td>sdr</td>
</tr>
<tr>
<td>Get SEL Information</td>
<td>sel</td>
</tr>
<tr>
<td>Reserve SEL</td>
<td>raw 0xa 0x42</td>
</tr>
<tr>
<td>Read FRU</td>
<td>fru</td>
</tr>
<tr>
<td>Get Sensor Information</td>
<td>sensor</td>
</tr>
<tr>
<td>Get Sensor Threshold</td>
<td>raw 0x4 0x27 &lt;sensor number&gt;</td>
</tr>
<tr>
<td>Get Sensor Reading</td>
<td>raw 0x4 0x2D</td>
</tr>
<tr>
<td>Get Sensor Type</td>
<td>raw 0x4 0x2F &lt;sensor number&gt;</td>
</tr>
<tr>
<td>I2C Master Write/Read</td>
<td>raw 0x6 0x52 0x6 0x4c 1 1</td>
</tr>
<tr>
<td>Get SOL Info</td>
<td>sol info</td>
</tr>
<tr>
<td>SOL Activate</td>
<td>sol activate</td>
</tr>
<tr>
<td>Get BT Capabilities</td>
<td>raw 0x6 0x36</td>
</tr>
<tr>
<td>Cold Reset</td>
<td>bmc reset cold</td>
</tr>
</tbody>
</table>
Next Step

- Continue development effort to complete Project Olympus Server manageability features
- Join Project Olympus workgroup and provide feedback
- Download and try the OpenBMC support for Project Olympus:
  - [https://github.com/ProjectOlympusOpenBMC](https://github.com/ProjectOlympusOpenBMC)