How to design a scalable embedded product

supporting different NXP i.MX applications processor families

Speakers

- Robert Thompson, Director Global i.MX Ecosystem, NXP Semiconductors
- Ofer Austerlitz, VP Business Development & Sales, Variscite
- Aviad Hadad, R&D Hardware expert, Variscite
- Pierluigi Passaro, R&D Software expert, Variscite
Webinar Agenda

- The System-on-Module concept for a scalable design
- NXP's i.MX 8 product portfolio
- Variscite pin-to-pin System-on-Module families
- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
  - Software aspects
  - Q&As
Webinar Agenda

- The System-on-Module concept for a scalable design
  Ofer Austerlitz, VP Business Development & Sales, Variscite

- NXP's i.MX 8 product portfolio

- Variscite pin-to-pin System-on-Module families

- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
  - Software aspects
  - Q&As
The System-on-Module concept

The basis for a scalable design
A typical System-on-Module

- The VAR-SOM-MX8M-MINI based on i.MX 8M Mini
- DART-MX8M-MINI Size: 55.0 x 30.0 mm (~2.2 x 1.2 in)
System-on-Module:

Off-The-Shelf stability,
Custom design flexibility

- Fast time to market
- Reduce R&D cost
- Scalable AND Upgradeable
- 100% yield – All the time
- Production-ready software support
- Tested and used by hundreds of customers
Webinar Agenda

- The System-on-Module concept for a scalable design
- NXP's i.MX 8 product portfolio
  Robert Thompson, Director Global i.MX Ecosystem, NXP Semiconductors
- Variscite pin-to-pin System-on-Module families
- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
  - Software aspects
  - Q&As
i.MX 8M Family of Applications Processors

- **i.MX 6**
  - QuadPlus
  - Quad
  - DualPlus
  - Dual
  - DualLite
  - Solo
  - SoloX
  - SoloLite
  - SLL
  - ULZ

- **i.MX 8**
  - A72
  - A53
  - M4

- **i.MX 8M**
  - Family
  - Advanced Graphics, Vision & Performance

- **i.MX 8X**
  - Family
  - Advanced Computing, Audio/Video & Voice

- **i.MX 7**
  - Family
  - Safety Certifiable & Efficient Performance
  - Arm® v8-A (32-bit/ 64-bit)

- **i.MX 7ULP**
  - Family
  - Ultra Low Power with Graphics
  - Arm® v7-A (32-bit)
i.MX 8 Series: Target Applications

Advanced graphics, video, image processing, vision, audio and voice

i.MX 8M Family
Advanced Computing, Audio/Video & Voice

i.MX 8X Family
Safety Certifiable & Efficient Performance

i.MX 8 Family
Advanced Graphics, Vision & Performance
i.MX 8 Series: Scalable Solutions

Scalable series of three Arm V8 64-bit (/32-bit) based SoC Families

i.MX 8M Family
Advanced Computing, Audio/Video & Voice

i.MX 8M QuadLite
i.MX 8M Quad, Dual
i.MX 8M Mini QuadLite, DualLite, SoloLite
Pin Compatible

Pin Compatible

i.MX 8 DualX

i.MX 8QuadXPlus, 8DualXPlus

i.MX 8M Quad, Dual
i.MX 8M Mini Quad, Dual, Solo
Pin Compatible

70% Hardware and Software Reuse

Software Compatible (including GPU Tools)

i.MX 8X Family
Safety Certifiable & Efficient Performance

i.MX 8QuadMax, 8QuadPlus

i.MX 8QuadXPlus

i.MX 8QuadMax

i.MX 8QuadPlus

i.MX 8QuadMax

i.MX 8QuadPlus

i.MX 8X Family
Advanced Graphics, Vision & Performance

A53

M4

A35

M4

A72

A53

M4

M4

A53

M4

Advanced Computing,
Audio/Video & Voice

Advanced Graphics,
Vision & Performance

Safety Certifiable &
Efficient Performance

70% Hardware and Software Reuse

Software Compatible (including GPU Tools)
## i.MX 8M: Scalable broad market Solutions

<table>
<thead>
<tr>
<th>i.MX 8M <strong>Quad</strong> Family</th>
<th>i.MX 8M <strong>Mini</strong> Family</th>
<th>i.MX 8M <strong>Nano</strong> Family</th>
<th>i.MX 8M <strong>Plus</strong> Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>A53 M4 GPU</td>
<td>A53 M4 GPU</td>
<td>A53 M7 GPU</td>
<td>A53 M7 NPU DSP GPU</td>
</tr>
<tr>
<td>Available NOW</td>
<td>Available NOW</td>
<td>Available NOW</td>
<td>4Q 2020</td>
</tr>
<tr>
<td>i.MX 8M Quad</td>
<td>i.MX 8M Mini Quad/Dual</td>
<td>i.MX 8M Nano</td>
<td>i.MX 8M Plus</td>
</tr>
<tr>
<td>i.MX 8M Dual</td>
<td>QuadLite</td>
<td>Quad/DualLite/</td>
<td>Quad/Dual</td>
</tr>
<tr>
<td>i.MX 8M QuadLite</td>
<td>DualLite/Solo</td>
<td>SoloLite</td>
<td></td>
</tr>
</tbody>
</table>

- A53: Arm Cortex-A53
- M4: Arm Cortex-M4
- M7: Arm Cortex-M7
- NPU: Nervous Processing Unit
- DSP: Digital Signal Processor
- GPU: Graphics Processing Unit

**17x17 Package**

**Pin Compatible 14x14 Package**

**11x11 Package**

**15x15 Package**

**Available NOW**

**4Q 2020**

Scalable series of **FOUR** Arm V8 64-bit (/32-bit) based SoC Families

Software Compatible (including GPU Tools)
i.MX 8M Target Applications

Consumer & Pro Audio Systems

Smart Home & Building Automation

Industrial HMI, Vision and Automation

Enterprise, Commercial & Healthcare
Webinar Agenda

- The System-on-Module concept for a scalable design
- NXP’s i.MX 8 product portfolio
- Variscite pin-to-pin System-on-Module families
  Ofer Austerlitz, VP Business Development & Sales, Variscite
- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
  - Software aspects
  - Q&As
Variscite Pin2Pin product families
Two highly scalable product families based on NXP processors

VAR-SOM Pin2Pin Family

<table>
<thead>
<tr>
<th>Model</th>
<th>Processor Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR-SOM-MX8M-PLUS</td>
<td>NXP i.MX 8 M Plus 4x 2GHz Cortex-A53</td>
</tr>
<tr>
<td>VAR-SOM-MX8</td>
<td>NXP i.MX 8X 4x 1.5GHz Cortex-A53</td>
</tr>
<tr>
<td>VAR-SOM-MX8X</td>
<td>NXP i.MX 8M Mini 4x 1.2GHz Cortex-A35</td>
</tr>
<tr>
<td>VAR-SOM-MX8M-MINI</td>
<td>NXP i.MX 8M Nano 4x 1.5GHz Cortex-A53</td>
</tr>
<tr>
<td>VAR-SOM-MX8M-NANO</td>
<td>NXP i.MX 6 4x 1.2GHz Cortex-A9</td>
</tr>
<tr>
<td>VAR-SOM-MX6</td>
<td>NXP i.MX 6 2x 1GHz Cortex-A9</td>
</tr>
<tr>
<td>VAR-SOM-6UL</td>
<td>NXP i.MX 6 UL/ULL/ULZ 900MHz Cortex-A7</td>
</tr>
</tbody>
</table>

DART Pin2Pin Family

<table>
<thead>
<tr>
<th>Model</th>
<th>Processor Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>DART-MX8M-PLUS</td>
<td>NXP i.MX 8 M Plus 4x 2GHz Cortex-A53</td>
</tr>
<tr>
<td>DART-MX8M</td>
<td>NXP i.MX 8M 4x 1.5GHz Cortex-A53</td>
</tr>
<tr>
<td>DART-MX8M-MINI</td>
<td>NXP i.MX 8M Mini 4x 1.8GHz Cortex-A53</td>
</tr>
</tbody>
</table>

Note: pin2pin compatibility depends on pinmux options
Webinar Agenda

- The System-on-Module concept for a scalable design
- NXP’s i.MX 8 product portfolio
- Variscite pin-to-pin System-on-Module families

- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
    - Aviad Hadad, R&D Hardware expert, Variscite
  - Software aspects
  - Q&As
VAR-SOM Pin2Pin family

**Main compatible features of the VAR-SOM family:**

**Display:**
- LVDS
- Touch controller

**Networking:**
- Ethernet
- Certified Dual band 802.11 ac/a/b/g/n Wi-Fi + BT v4.2 module

**Audio:**
- Analog Headphone, Line In
- Digital microphone + Digital audio I/F

**Multimedia:**
- MIPI CSI2 Camera

**Connectivity:**
- USB 2.0/3.0
- PCIe Gen 2.0
- SD/MMC

**OS support:**
- Linux – Yocto, Debian

**Unique features in specific modules**
- 4K H.265/H.264 Decode
- 1080p H.265/H.264 encode/decode
- HDMI/eDP/DP
- USB 3.0
- Dual Gbit Ethernet
- Parallel CSI
- ADC
- SATA
- Parallel RGB
- E-ink display support

**Symphony-Board EVK main Features:**

- Display connection - MIPI-DSI, LVDS, HDMI/DP (via Extension Card)
- Capacitive, Resistive touch connector
- Dual Gigabit Ethernet via RJ45 connector
- Mini PCIe Socket
- USB 3.0 OTG Type C
- USB 2.0 Host Type A
- Headphones, Line In jack, DMIC
- Micro SD card slot
- MIPI CSI camera, Parallel camera (via Extension Card)
- Micro SATA connector
VAR-SOM Pin2Pin family – video intro
Many interface options exist especially with the pin mux options. We selected the most commonly used interfaces in the embedded market and decided to make them pin2pin (blue) the others can be “partial” pin2pin or even unique for a specific processor/SoM.

Conclusion: one needs to pay attention when designing the custom carrier board and make sure that his own interfaces are falling into the pin2pin options.
A glance at the VAR-SOM-MX8M-MINI datasheet

- SO-DIMM pinout table
- Pinmux table
- Interface Pinout table

### 7.3.1. VAR-SOM-MX8M-MINI SO-DIMM Pin-out

#### Table 3: SO-DIMM PIN-OUT

<table>
<thead>
<tr>
<th>PIN</th>
<th>ASSY</th>
<th>BALL NAME</th>
<th>GPIO3</th>
<th>NOTES</th>
<th>BALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No EC</td>
<td>ENET_TX_CTL</td>
<td>GPIO3_1022</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE24</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>Digital Ground</td>
<td>GPIO3_1024</td>
<td>With “EC” configuration this pin in Not Connected.</td>
<td>NC, EC</td>
</tr>
<tr>
<td>3</td>
<td>No EC</td>
<td>ENET_TDO</td>
<td>GPIO3_1016</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE25</td>
</tr>
<tr>
<td>5</td>
<td>EC</td>
<td>ETH_TX [0-1]</td>
<td>GPIO3_1026</td>
<td>Signal source to Ethernet PHY.</td>
<td>AR8033.11</td>
</tr>
<tr>
<td>4</td>
<td>No EC</td>
<td>ENET_RXD0</td>
<td>GPIO3_1019</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE27</td>
</tr>
<tr>
<td>6</td>
<td>EC</td>
<td>ETH_RX [0-1]</td>
<td>GPIO3_1019</td>
<td>Signal source to Ethernet PHY.</td>
<td>AR8033.17</td>
</tr>
<tr>
<td>7</td>
<td>No EC</td>
<td>ENET_RD1</td>
<td>GPIO3_1027</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE28</td>
</tr>
<tr>
<td>8</td>
<td>GND</td>
<td>Digital Ground</td>
<td>GPIO3_1029</td>
<td>Signal source to Ethernet PHY.</td>
<td>AR8033.18</td>
</tr>
<tr>
<td>9</td>
<td>No EC</td>
<td>ENET_TD1</td>
<td>GPIO3_1020</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE29</td>
</tr>
<tr>
<td>10</td>
<td>EC</td>
<td>ETH_TX0</td>
<td>GPIO3_1028</td>
<td>Signal source to Ethernet PHY.</td>
<td>AR8033.14</td>
</tr>
<tr>
<td>11</td>
<td>No EC</td>
<td>ENET_TDI</td>
<td>GPIO3_1021</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE26</td>
</tr>
<tr>
<td>12</td>
<td>EC</td>
<td>ETH_TX1</td>
<td>GPIO3_1029</td>
<td>Signal source to Ethernet PHY.</td>
<td>AR8033.20</td>
</tr>
<tr>
<td>13</td>
<td>No EC</td>
<td>ENET_RXD</td>
<td>GPIO3_1022</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE27</td>
</tr>
<tr>
<td>14</td>
<td>GND</td>
<td>Digital Ground</td>
<td>GPIO3_1024</td>
<td>Power supply for VDD_ENET pin</td>
<td>SOC_AFE28</td>
</tr>
<tr>
<td>15</td>
<td>No EC</td>
<td>ENET_RX_CTL</td>
<td>GPIO3_1024</td>
<td>Powered by VDD_ENET pin</td>
<td>SOC_AFE28</td>
</tr>
</tbody>
</table>

### 7.4. VAR-SOM-MX8M-MINI Pin-Mux

This section lists the SOM connectors with the available functions on each pin.

#### Table 4: VAR-SOM-MX8M-MINI PINMUX

<table>
<thead>
<tr>
<th>PIN</th>
<th>ASSY</th>
<th>BALL</th>
<th>ALT</th>
<th>ALT</th>
<th>ALT</th>
<th>ALT</th>
<th>ALT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No EC</td>
<td>ENET_TX_CTL</td>
<td>ENET_TX_CLK</td>
<td>ENET_TX_DB0</td>
<td>ENET_TX_DB1</td>
<td>ENET_TX_DB2</td>
<td>ENET_TX_DB3</td>
</tr>
<tr>
<td>0</td>
<td>No EC</td>
<td>ENET_TDI</td>
<td>ENET_TDO</td>
<td>ENET_TX_DB0</td>
<td>ENET_TX_DB1</td>
<td>ENET_TX_DB2</td>
<td>ENET_TX_DB3</td>
</tr>
<tr>
<td>0</td>
<td>No EC</td>
<td>ENET_RXD0</td>
<td>ENET_RXD1</td>
<td>ENET_TX_DB0</td>
<td>ENET_TX_DB1</td>
<td>ENET_TX_DB2</td>
<td>ENET_TX_DB3</td>
</tr>
<tr>
<td>0</td>
<td>No EC</td>
<td>ENET_RXD2</td>
<td>ENET_RXD3</td>
<td>ENET_TX_DB0</td>
<td>ENET_TX_DB1</td>
<td>ENET_TX_DB2</td>
<td>ENET_TX_DB3</td>
</tr>
</tbody>
</table>

### 8.11.1.1. UART1 Signals

#### Table 32: UART1 Signals

<table>
<thead>
<tr>
<th>PIN</th>
<th>ASSY</th>
<th>ALT_NAME</th>
<th>ALT</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>UART1_CTS_R</td>
<td>UART1_CTS_R</td>
<td>1</td>
<td>UART1 pin 18</td>
</tr>
<tr>
<td>25</td>
<td>UART1_CTS_R</td>
<td>UART1_CTS_R</td>
<td>2</td>
<td>UART1 pin 19</td>
</tr>
<tr>
<td>57</td>
<td>UART1_CTS_R</td>
<td>UART1_CTS_R</td>
<td>3</td>
<td>UART1 pin 20</td>
</tr>
<tr>
<td>124</td>
<td>UART1_CTS_R</td>
<td>UART1_CTS_R</td>
<td>4</td>
<td>UART1 pin 21</td>
</tr>
</tbody>
</table>

19
Scalable Embedded design – Pinmuxing

Using the VAR-SOMs Compatibility spreadsheet:

- VAR-SOM-xx Pinmux tab available for each SoM in the VAR-SOM Pin2Pin family

<table>
<thead>
<tr>
<th>PIN</th>
<th>Name</th>
<th>SOM Usage</th>
<th>Applicable Configuration</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>NC/\CAN_TX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>&quot;ECSPI1_MOSI/UART3_TXD&quot;/GPIO5_ID07</td>
<td>ECSPI1</td>
<td>CN or TP</td>
<td>Shared Odd</td>
</tr>
<tr>
<td>46</td>
<td>NC/\CAN_RX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>GPIO1_ID02/&quot;WDOG_B&quot;/&quot;WDOG_ANY&quot;</td>
<td>WDOG_B</td>
<td>Always connected to PMIC_WDOG_B Input</td>
<td>Odd</td>
</tr>
<tr>
<td>48</td>
<td>SOM 3V3_PER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>SA13_RCC/GPT1_CLK/SA15_RCC/&quot;UART2_CTS_B&quot;/GPIO4_ID29</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>51</td>
<td>SA13_RCC/GPT1COMPARE1/SA15_RXD/&quot;UART2_RTS_B&quot;/GPIO4_ID30</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>52</td>
<td>SA15_RCC/GPT3_CAPTURE1/SA15_RXD/&quot;UART2_TXD&quot;/GPIO5_ID90</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>53</td>
<td>SA15_RCC/GPT3_CAPTURE2/SA15_RXD/&quot;UART2_RXD&quot;/GPIO4_ID31</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>54</td>
<td>UART3_RXD/GCSPI3_SCL/&quot;GPIO5_ID22</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>55</td>
<td>UART3_RXD/GCSPI3_SCS/&quot;GPIO5_ID26</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>56</td>
<td>UART3_TXD/GCSPI3_MOSI/&quot;GPIO5_ID23</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>57</td>
<td>UART3_TXD/GCSPI3_MISO/&quot;GPIO5_ID27</td>
<td>UART2</td>
<td>WBO or WB</td>
<td>Function can be released with BT disabled</td>
</tr>
<tr>
<td>58</td>
<td>GPIO1_N/C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>GPIO3_N/C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>SD2_CLK/&quot;GPIO3_ID13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>SD2_DATA0/&quot;GPIO3_ID17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>SD2_DATA0/&quot;GPIO3_ID15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>SD2_DATA1/&quot;GPIO3_ID16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>SD2_CMD/&quot;GPIO3_ID14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>SD2_DATA1/&quot;GPIO3_ID18</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scalable Embedded design – Pinmuxing

Using the VAR-SOMs Compatibility spreadsheet:

- “CompatibilityPinMap” tab - ALL VAR-SOM MOST COMMON FUNCTION table
Scalable Embedded design – Pinmuxing

Using the VAR-SOMs Compatibility spreadsheet:

- “CompatibilityPinMap” tab - ALL VAR-SOM MOST COMMON FUNCTION table (Cont.)

<table>
<thead>
<tr>
<th>PIN</th>
<th>VAR-SOM-MX8</th>
<th>VAR-SOM-MX8X</th>
<th>VAR-SOM-MX8XMM</th>
<th>VAR-SOM-MX8XMIN</th>
<th>VAR-SOM-MX8XMAX</th>
<th>VAR-SOM-MX8XMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>1</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>39</td>
<td>ECP11-50</td>
<td>SOM_PIN1000</td>
<td>SOM_PIN1000</td>
<td>SOM_PIN1000</td>
<td>SOM_PIN1000</td>
<td>SOM_PIN1000</td>
</tr>
<tr>
<td>40</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>41</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>45</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>46</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>47</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>48</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>49</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>50</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>51</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>52</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>53</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>54</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>55</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>56</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>57</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>58</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>59</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>60</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>61</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>62</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>63</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>64</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>65</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>66</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>67</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
<tr>
<td>68</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
<td>SPI32-001</td>
</tr>
</tbody>
</table>
An Overview of the Symphony-Board schematics

Carrier board design—Symphony-Board introduction

COMPATIBILITY LIST
Describes the ALT per SOM for compatibility.
Order of names: (MX6/MX8/MX8X/MX8M/MX8MM)

CUSTOM BOARD ALT
Describes ALT which can be demonstrated on CustomBoard.
"XX" denotes special SOM assembly option.
Order of names: [MX6/MX8/MX8X/MX8M/MX8MM]

FOR CROSS PROBING BETWEEN SOM SYMBOL AND THE SPECIFIC SOM CONNECTOR USED, SET THE "IMPLEMENTATION" PROPERTY VALUE IN SOM PORT SYMBOL TO ONE OF THE FOLLOWING:
1. VAR-SOM-MX6
2. VAR-SOM-MX8
3. VAR-SOM-MX8X
4. VAR-SOM-MX8M-MINI
5. VAR-SOM-MX8M-NANO
6. VAR-SOM-MX8M-PLUS

IMPLEMENTATION = VAR-SOM-MX8M-MINI
Carrier board design– Symphony-Board introduction

The Symphony-Board Schematics Hierarchy includes the dedicated Schematic page for all SOMs. Off-page connectors naming follow index denoted on SOM page.
Start the design by adopting the EVK pinmux selection as a baseline for the common required interfaces and go along adding additional interfaces from the available pins as required.

When using SOM pins for alternate functions not demonstrated in EVK, verify pins are not used internally by SOM for other functions. Refer to the SOM datasheet Pinout tables.

Some incompatibility may exist between SOM pinouts. The EVK schematics has dealt with most common ones, follow it.
Carrier board design guidelines/recommendations – Schematics

- Follow as much as possible EVK portions related to Power, reset and boot.
  - Boot pins - Verify pins are not driven externally to an undesirable state
  - Reset - Verify handling of POR signal in your carrier board
  - Power -
    o Ensure sufficient supply to module
    o Depending on Power supply proximity, add bulk capacitance for SOM power supply pins
    o Pay attention to design of Carrier board peripherals 3.3V supply and the carrier board power rails discharge

- USB – verify connections of VBUS, ID signals

- Add ESD and EMI protection devices on sensitive Interfaces
Carrier board design - Layout

Placement
- Verify SOM and holes’ placement using reference design files
- When placing components under SOM check height constraints
- Try to minimize distance of High speed interfaces connectors from SOM pins
- Place bulk capacitance of SOM power supply near SOM pins

Stackup
- Suggest using a minimum of 6 layer stackup
- Work with your PCB manufacture to achieve a stackup which provides all impedances with practical trace width/spacing dimensions and reference planes for high speed signals
Carrier board design - Layout

Routing
- Give priority to High speed signals such as PCI, USB 3.0, LVDS and HDMI
- Make effort to route high speed buses as a group in same layer
- length match signals of same bus, match intra pair (Positive/Negative) signals of differential pairs
- Ensure a continuous reference plane for impedance controlled signals
- Minimize crosstalk between high speed interfaces
- Keep high speed/sensitive signals away from noisy lines

Power
- Follow placement and routing guidelines of power supplies to reduce noise avoid EMC issues
- Use short thick connections
- Use multiple vias to connect between planes
Carrier board design – useful links

Design source files are available on our FTP:
ftp://customerv:Variscite1@ftp.variscite.com/VAR-SOM-MX8/Hardware/Symphony-Board/V1.3A

Mechanical 2D/3D files of SOM/carrier boards can be found under documentation tab of SOM/carrier boards page in variscite website.
SOMs: 
https://www.variscite.com/products/system-on-module-som/
Carrier boards: 
https://www.variscite.com/products/single-board-computers/
Webinar Agenda

- The System-on-Module concept for a scalable design
- NXP's i.MX 8 product portfolio
- Variscite pin-to-pin System-on-Module families

- How to design a scalable embedded product supporting various i.MX processors:
  - Hardware aspects
  - Software aspects
    - Pierluigi Passaro, R&D Software expert, Variscite
  - Q&As
SW scalability

- Scalability definitions
- Android approach
- Linux kernel HAL
- Wrapper libraries
- Application design
System vs Software scalability

- System scalability
  - capability of a system to adapt and fit to specific problems

- Software scalability
  - capability of a software to correctly manage an increased workload (users)

- Variscite SoM scalability
  - capability to run the same application on the same board while using different SoMs
Android approach: the HAL

- 1 SDK
- 1 apk
- several devices
Android SDK limitations

- No direct access to HW busses
  - GPIO
  - I2C
  - SPI
  - UART
  - USB

- Designed to limit user access to HW: Android SDK focus on devices functionalities
  - LEDs/keys (GPIO)
  - Audio (I2C)
  - Sensors (SPI)
  - Bluetooth (UART)
  - Cameras (USB)
HAL requirements for Linux kernel

- Availability of drivers exposing standard interfaces
  - Audio
  - Power supply
  - Touchscreens

- Device trees exposing standard interfaces

```plaintext
default-brightness-level = <8>;
status = "ok";
};
```
Variscite Device tree design: resource availability

- Expose the same interfaces using the same SoM pins

- NXP pinmuxing
  - SymphonyBoard J18.3/5 provide UARTs
    - VAR-SOM-MX8M-MINI
    - VAR-SOM-MX8M-NANO
    - VAR-SOM-MX8X
    - VAR-SOM-MX8
Customer Device tree design: avoid using low level GPIOs

- **gpio-keys**: allow exposing key events instead of GPIO indexes

- **gpio-leds**: allow exposing symbolic names to set GPIO status

```c
gpio-keys {
    compatible = "gpio-keys";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_gpio_keys>;

    up {
        label = "Up";
        gpios = <&gpio4 18 GPIO_ACTIVE_LOW>;
        linux,code = <KEY_UP>;
    }
};

gpio-leds {
    compatible = "gpio-leds";
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_leds>;

    emmc {
        label = "eMMC";
        gpios = <&gpio4 17 GPIO_ACTIVE_HIGH>;
        linux,default-trigger = "mmc0";
    }
};
```
Customer Device tree design: avoid I2C / SPI raw access

- Use dedicated kernel drivers for devices connected to the busses -
  This will provide standard interfaces regardless of the used bus

```c
%i2c2 {
    clock-frequency = <1000000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c2>;
    status = "okay";

    /* DS1337 RTC module */
    rtc@0x68 {
        status = "okay";
        compatible = "dallas,ds1337";
        reg = <0x68>;
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_rtc>;
        interrupt-parent = <&gpio1>;
        interrupts = <15 IRQ_TYPE_EDGE_FALLING>;
    }
};
```
Device tree limitations

- Missing resources (Second Ethernet port, GPU, VPU, ...)

- Different device naming (UARTs)
  - SymphonyBoard J18.3/5 provide UARTs
    - VAR-SOM-MX8M-MINI => ttymxc2
    - VAR-SOM-MX8M-NANO => ttymxc2
    - VAR-SOM-MX8X => ttylp1
    - VAR-SOM-MX8 => ttylp2
middleware: user-space HAL

- Design dedicated libraries
  - probing HW capabilities
  - providing SoC/SoM specific capabilities

- Design dedicated configuration files
  - providing resource specific data (like UART names)
Application design

- No HW dependencies

- Develop and test the application on a specific SoM (or even on a Linux PC)

- Not just HW scalability (one board / several SoMs)

- Not just SW scalability (one applications / several SoMs)

- Simultaneous HW/SW scalability (one board, one application / several SoMs)
Beyond HW/SW scalability

- Docker containers
  - Devices virtualization
  - Resource (processes) virtualizations
Contact Information

Email | sales@variscite.com

Website | www.variscite.com

Variscite wiki | www.variwiki.com

Customer Portal | Variscite Portal
Questions?
THANK YOU!