IPMB Driver for a Satellite MC

The Intelligent Platform Management Bus or IPMB, is an I2C bus that provides a standardized interconnection between different boards within a chassis. This interconnection is between the baseboard management (BMC) and chassis electronics. IPMB is also associated with the messaging protocol through the IPMB bus.

The devices using the IPMB are usually management controllers that perform management functions such as servicing the front panel interface, monitoring the baseboard, hot-swapping disk drivers in the system chassis, etc...

When an IPMB is implemented in the system, the BMC serves as a controller to give system software access to the IPMB. The BMC sends IPMI requests to a device (usually a Satellite Management Controller or Satellite MC) via IPMB and the device sends a response back to the BMC.

For more information on IPMB and the format of an IPMB message, refer to the IPMB and IPMI specifications.

IPMB driver for Satellite MC

ipmb-dev-int - This is the driver needed on a Satellite MC to receive IPMB messages from a BMC and send a response back. This driver works with the I2C driver and a userspace program such as OpenIPMI:

1. It is an I2C slave backend driver. So, it defines a callback function to set the Satellite MC as an I2C slave. This callback function handles the received IPMI requests.
2. It defines the read and write functions to enable a user space program (such as OpenIPMI) to communicate with the kernel.

Load the IPMB driver

The driver needs to be loaded at boot time or manually first. First, make sure you have the following in your config file: CONFIG_IPMB_DEVICE_INTERFACE=y

1. If you want the driver to be loaded at boot time:

a. Add this entry to your ACPI table, under the appropriate SMBus:

```
Device (SMB0) // Example SMBus host controller
{
Name (_HID, "<Vendor-Specific HID>") // Vendor-Specific HID
Name (_UID, 0) // Unique ID of particular host controller
:
:
  Device (IPMB)
  {
    Name (_HID, "IPMB0001") // IPMB device interface
    Name (_UID, 0) // Unique device identifier
  }
}
```

b. Example for device tree:

```
&i2c2 {
       status = "okay";

       ipmb@10 {
               compatible = "ipmb-dev";
               reg = <0x10>;
               i2c-protocol;
       };
};
```

If xmit of data to be done using raw i2c block vs smbus then "i2c-protocol" needs to be defined as above.

2. Manually from Linux:

```
modprobe ipmb-dev-int
```

Instantiate the device

After loading the driver, you can instantiate the device as described in 'Documentation/i2c/instantiating-devices.rst'. If you have multiple BMCs, each connected to your Satellite MC via a different I2C bus, you can instantiate a device for each of those BMCs.

The name of the instantiated device contains the I2C bus number associated with it as follows:

```
BMC1 ------ IPMB/I2C bus 1 ---------|   /dev/ipmb-1
                              Satellite MC
BMC1 ------ IPMB/I2C bus 2 ---------|   /dev/ipmb-2
```

For instance, you can instantiate the ipmb-dev-int device from user space at the 7 bit address 0x10 on bus 2:

```
# echo ipmb-dev 0x1010 > /sys/bus/i2c/devices/i2c-2/new_device
```

This will create the device file /dev/ipmb-2, which can be accessed by the user space program. The device needs to be instantiated before running the user space program.