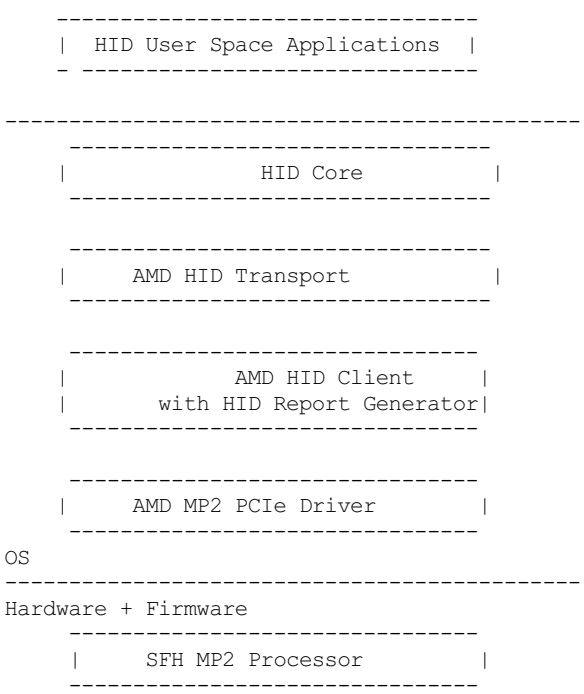
AMD Sensor Fusion Hub

AMD Sensor Fusion Hub (SFH) is part of an SOC starting from Ryzen-based platforms. The solution is working well on several OEM products. AMD SFH uses HID over PCIe bus. In terms of architecture it resembles ISH, however the major difference is all the HID reports are generated as part of the kernel driver.

Block Diagram



AMD HID Transport Layer

AMD SFH transport is also implemented as a bus. Each client application executing in the AMD MP2 is registered as a device on this bus. Here, MP2 is an ARM core connected to x86 for processing sensor data. The layer, which binds each device (AMD SFH HID driver) identifies the device type and registers with the HID core. Transport layer attaches a constant "struct hid_ll_driver" object with each device. Once a device is registered with HID core, the callbacks provided via this struct are used by HID core to communicate with the device. AMD HID Transport layer implements the synchronous calls.

AMD HID Client Layer

This layer is responsible to implement HID requests and descriptors. As firmware is OS agnostic, HID client layer fills the HID request structure and descriptors. HID client layer is complex as it is interface between MP2 PCIe layer and HID. HID client layer initializes the MP2 PCIe layer and holds the instance of MP2 layer. It identifies the number of sensors connected using MP2-PCIe layer. Based on that allocates the DRAM address for each and every sensor and passes it to MP2-PCIe driver. On enumeration of each sensor, client layer fills the HID Descriptor structure and HID input report structure. HID Feature report structure is optional. The report descriptor structure varies from sensor to sensor.

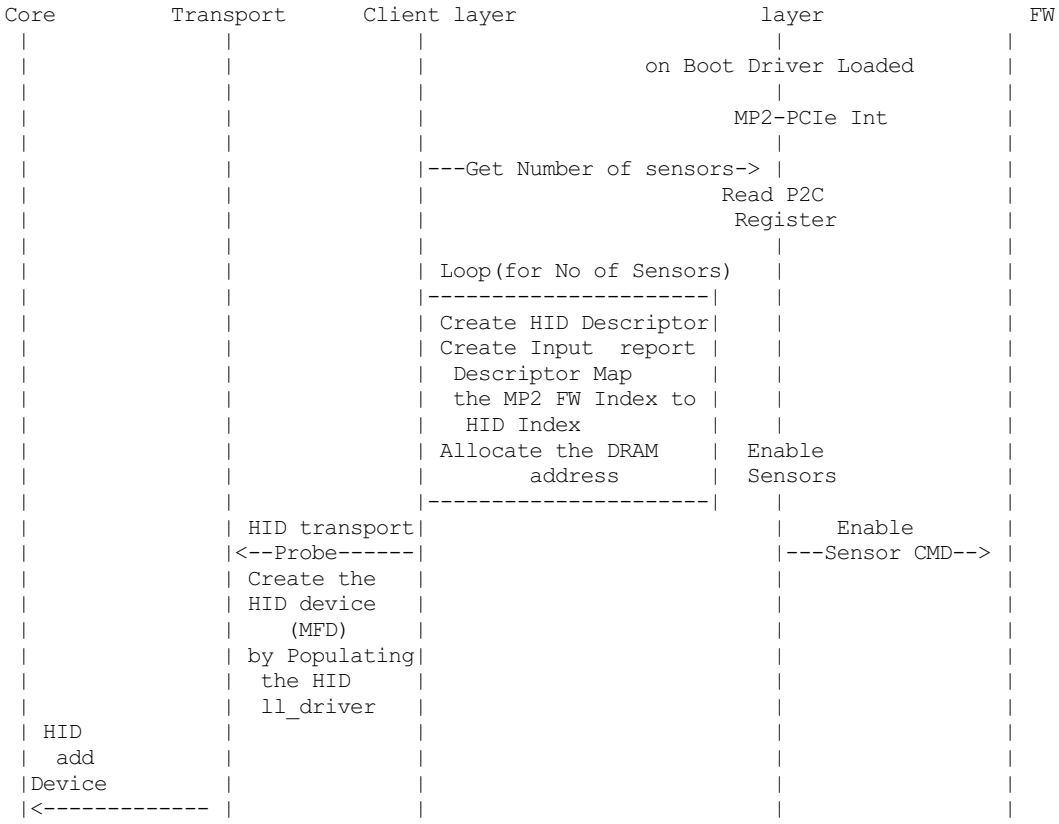
AMD MP2 PCIe layer

MP2 PCIe Layer is responsible for making all transactions with the firmware over PCIe. The connection establishment between firmware and PCIe happens here.

The communication between X86 and MP2 is split into three parts. 1. Command transfer via the C2P mailbox registers. 2. Data transfer via DRAM. 3. Supported sensor info via P2C registers.

Commands are sent to MP2 using C2P Mailbox registers. Writing into C2P Message registers generates interrupt to MP2. The client layer allocates the physical memory and the same is sent to MP2 via the PCI layer. MP2 firmware writes the command output to the access DRAM memory which the client layer has allocated. Firmware always writes minimum of 32 bytes into DRAM. So as a protocol driver shall allocate minimum of 32 bytes DRAM space.

Enumeration and Probing flow



Data Flow from Application to the AMD SFH Driver