Request #: HUTRR109
Title: SoC Page
Spec Release: 1.3
Requester: Matthew Williams
Company: Microsoft

Current Status: **Approved**
Priority: Normal

Required Voter: ~~Apple~~
Required Voter: Intel
Required Voter: Logitech

Voting Begins: 23rd June 2022
Voting Ends: 30th June 2022
Voting Result: 5-0

Summary:
Establish a new Usage Page to describe a Hosts interaction and management of a SoC (System-on-Chip/microcontroller). In particular, Usages to allow the transfer of firmware from the Host to the SoC.

Scenario:
The hardware required for the persistent storage of firmware in SoCs/embedded-systems can significantly increase BOM (Bill of Materials) costs. A cost-saving alternative is to store firmware files on the Host (using its larger+cheaper persistent storage), being retrieved after Host initialization.

This mechanism also provides the Host a consistent way to update firmware present in the SoC.

Proposal:

20 – SoC Page (0x11)

SoC devices describe management/control interfaces to a SoC and its distinct embedded devices.

Usage ID	Usage Name	Usage Type
0x00	Reserved	
0x01	SocControl	CA
0x02	FirmwareTransfer	CL
0x03	FirmwareFileId	DV
0x04	FileOffsetInBytes	DV
0x05	FileTransferSizeMaxInBytes	DV
0x06	FilePayload	DV
0x07	FilePayloadSizeInBytes	DV
0x08	FilePayloadContainsLastBytes	DF
0x09	FileTransferStop	DF
0x0A	FileTransferTillEnd	DF

20.1 SoC Control Device

Usage Name	Usage Type	Usage Type
SocControl	CA	A device that provides basic control over a microcontroller/SoC (System-on-chip)

20.2 – Firmware Transfer

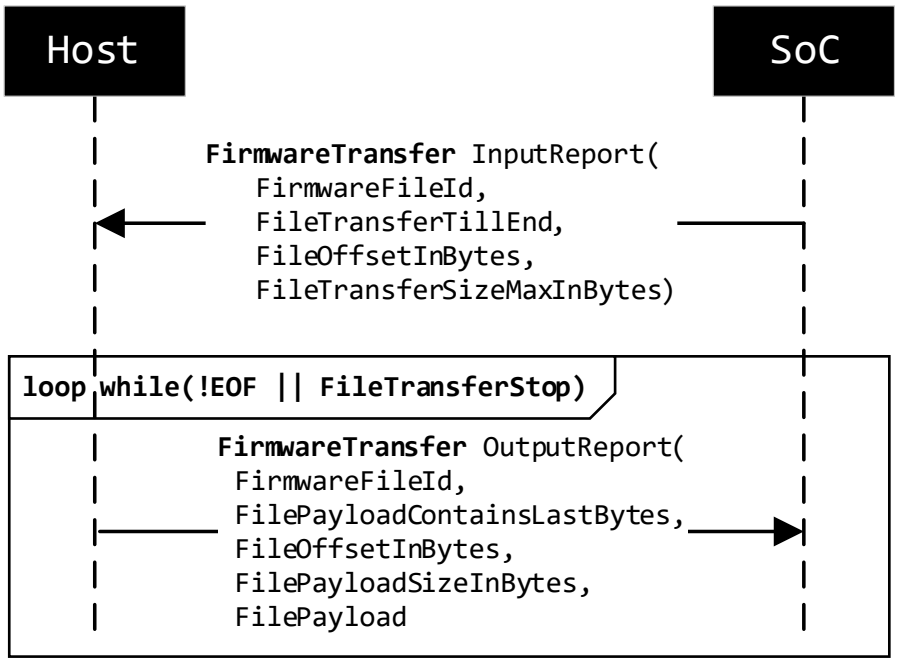
The below Usages describe how a SoC can request specific firmware files from a Host. A SoC may utilize multiple firmware files itself and/or retrieve on behalf of other attached-devices. As underlying transports have different maximum report sizes (e.g. USB == max(USHORT)) large files are expected to be split over multiple reports.

Note: It is not defined how the Host previously received the firmware files to transfer, or how the Host associates a particular file with a FirmwareFileId.

Having the SoC retrieve significant parts of its firmware (or for attached-devices) from the Host (e.g. at boot-time), can reduce the need for SoC-specific persistent storage. By instead relying on the Host (using it's larger+cheaper persistent storage) significant cost-savings may be realized.

Usage Name	**Usage Type**	**Description**
FirmwareTransfer	CL	Contains controls used to transfer firmware of a SoC (or a distinct embedded-device attached to the SoC)
FirmwareFileId	DV	Identifier for a specific Firmware file. Id == 0 is undefined. A SoC may have multiple firmware files (depending on its current application), or requests on-behalf of distinct attached embedded-devices. *Note: It is not defined how the SoC/Host establish the common file identifier.*
FileOffsetInBytes	DV	Offset (in bytes) from the start of the file. *In conjunction with FileTransferSizeMaxInBytes this allows a SoC to request only part of a file (e.g. because only a portion of SoC RAM is available and an internal firmware transfer is still pending).*
FileTransferSizeMaxInBytes	DV	Maximum bytes of the file to be transferred. This can vary across requests and may not be reached if the actual file size is less than specified. *In conjunction with FileOffsetInBytes this allows a SoC to request only part of a file (e.g. because only a portion of SoC RAM is available and an internal firmware transfer is still pending).*
FilePayload	DV	The actual (partial) File payload. May be split over multiple reports.
FilePayloadSizeInBytes	DV	Number of valid bytes of FilePayload. *Note: This is expected to be LogicalMax until the last transfer, which may only use part of the available FilePayload item.*

FilePayloadContainsLastBytes	DF	Indicates if the payload contains the last bytes of the file. When false, indicates more Output Reports should be expected. *Note: This flag is used as an alternative to specifying the actual size of the File, which the SoC could use to determine whether a report was the final expected one.*
FileTransferStop	DF	Indicates the current transfer should stop.
FileTransferTillEnd	DF	Indicates whether the SoC expects the Host to send only a single FilePayload report (containing the maximum bytes from the requested Offset), or multiple FilePayload reports without further requests from the SoC. The transfer can be stopped at any time by 'FileTransferStop'



Sample Descriptor .wara:

The below describes an SoC Control device that can request the Host to provide it one of 3 firmware files (ids 1-3).

After the Host receives the ‘FirmwareTransfer’ InputReport, it will send OutputReports (containing chunks of the Firmware file) until it has been completely received or the SoC sends the InputReport containing ‘FileTransferStop’.

If the Device detects an error (e.g. ‘FileOffsetInBytes’ in the OutputReport is not expected), it can stop the transfer (InputReport with ‘FileTransferStop’), and start the transfer again from the expected offset.

The final OutputReport sent by the Host will set FilePayloadContainsLastBytes to 1.

*Below document authored with **Waratah** (https://github.com/microsoft/hidtools)*

```
--------------------------------------------------------------------------------
[[usagePage]]
id = 0x0011
name = 'SoC'

    [[usagePage.usage]]
    id = 0x01
    name = 'SocControl'
    types = ['CA']

    [[usagePage.usage]]
    id = 0x02
    name = 'FirmwareTransfer'
    types = ['CL']

    [[usagePage.usage]]
    id = 0x03
    name = 'FirmwareFileId'
    types = ['DV']

    [[usagePage.usage]]
    id = 0x04
    name = 'FileOffsetInBytes'
    types = ['DV']

    [[usagePage.usage]]
    id = 0x05
    name = 'FileTransferSizeMaxInBytes'
    types = ['DV']

    [[usagePage.usage]]
    id = 0x06
    name = 'FilePayload'
    types = ['DV']

    [[usagePage.usage]]
    id = 0x07
```

```
    name = 'FilePayloadSizeInBytes'
    types = ['DV']

    [[usagePage.usage]]
    id = 0x08
    name = 'FilePayloadContainsLastBytes'
    types = ['DF']

    [[usagePage.usage]]
    id = 0x09
    name = 'FileTransferStop'
    types = ['DF']

    [[usagePage.usage]]
    id = 0x0A
    name = 'FileTransferTillEnd'
    types = ['DF']

[[applicationCollection]]
usage = ['SoC', 'SocControl']

    [[applicationCollection.inputReport]]

        [[applicationCollection.inputReport.logicalCollection]]
        usage = ['SoC', 'FirmwareTransfer']

            [[applicationCollection.inputReport.logicalCollection.variableItem]]
            usage = ['SoC', 'FirmwareFileId']
            logicalValueRange = [1, 3]

            [[applicationCollection.inputReport.logicalCollection.variableItem]]
            usage = ['SoC', 'FileTransferTillEnd']
            sizeInBits = 1
            logicalValueRange = 'maxUnsignedSizeRange'

            [[applicationCollection.inputReport.logicalCollection.variableItem]]
            usage = ['SoC', 'FileOffsetInBytes']
            sizeInBits = 16
            logicalValueRange = 'maxUnsignedSizeRange'

            [[applicationCollection.inputReport.logicalCollection.variableItem]]
            usage = ['SoC', 'FileTransferSizeMaxInBytes']
            sizeInBits = 16
            logicalValueRange = 'maxUnsignedSizeRange'

    [[applicationCollection.inputReport]]

        [[applicationCollection.inputReport.logicalCollection]]
        usage = ['SoC', 'FirmwareTransfer']

            [[applicationCollection.inputReport.logicalCollection.variableItem]]
```

```
        usage = ['SoC', 'FileTransferStop']
        sizeInBits = 1
        logicalValueRange = 'maxUnsignedSizeRange'

[[applicationCollection.outputReport]]

    [[applicationCollection.outputReport.logicalCollection]]
    usage = ['SoC', 'FirmwareTransfer']

        # LogicalRange same as the InputReport.
        [[applicationCollection.outputReport.logicalCollection.variableItem]]
        usage = ['SoC', 'FirmwareFileId']
        logicalValueRange = [1, 3]

        [[applicationCollection.outputReport.logicalCollection.variableItem]]
        usage = ['SoC', 'FilePayloadContainsLastBytes']
        sizeInBits = 1
        logicalValueRange = 'maxUnsignedSizeRange'

        [[applicationCollection.outputReport.logicalCollection.variableItem]]
        usage = ['SoC', 'FileOffsetInBytes']
        sizeInBits = 16
        logicalValueRange = 'maxUnsignedSizeRange'

        # Range is the same as FilePayload count.
        [[applicationCollection.outputReport.logicalCollection.variableItem]]
        usage = ['SoC', 'FilePayloadSizeInBytes']
        logicalValueRange = [0, 8000]

        [[applicationCollection.outputReport.logicalCollection.variableItem]]
        usage = ['SoC', 'FilePayload']
        sizeInBits = 8
        logicalValueRange = 'maxUnsignedSizeRange'
        count = 8000
```

Sample Descriptor:
Below document output from above .wara contents

```
0x05, 0x11,                    // UsagePage(SoC[17])
0x09, 0x01,                    // UsageId(SocControl[1])
0xA1, 0x01,                    // Collection(Application)
0x85, 0x01,                    //     ReportId(1)
0x09, 0x02,                    //     UsageId(FirmwareTransfer[2])
0xA1, 0x02,                    //     Collection(Logical)
0x09, 0x03,                    //         UsageId(FirmwareFileId[3])
0x15, 0x01,                    //         LogicalMinimum(1)
0x25, 0x03,                    //         LogicalMaximum(3)
0x95, 0x01,                    //         ReportCount(1)
0x75, 0x02,                    //         ReportSize(2)
0x81, 0x02,                    //         Input(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0x09, 0x0A,                    //         UsageId(FileTransferTillEnd[10])
0x15, 0x00,                    //         LogicalMinimum(0)
0x25, 0x01,                    //         LogicalMaximum(1)
0x75, 0x01,                    //         ReportSize(1)
0x81, 0x02,                    //         Input(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0x09, 0x04,                    //         UsageId(FileOffsetInBytes[4])
0x09, 0x05,                    //         UsageId(FileTransferSizeMaxInBytes[5])
0x27, 0xFF, 0xFF, 0x00, 0x00,  //         LogicalMaximum(65,535)
0x95, 0x02,                    //         ReportCount(2)
0x75, 0x10,                    //         ReportSize(16)
0x81, 0x02,                    //         Input(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0xC0,                          //     EndCollection()
0x95, 0x01,                    //     ReportCount(1)
0x75, 0x05,                    //     ReportSize(5)
0x81, 0x03,                    //     Input(Constant, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0x85, 0x02,                    //     ReportId(2)
0x09, 0x02,                    //     UsageId(FirmwareTransfer[2])
0xA1, 0x02,                    //     Collection(Logical)
0x09, 0x09,                    //         UsageId(FileTransferStop[9])
0x25, 0x01,                    //         LogicalMaximum(1)
0x75, 0x01,                    //         ReportSize(1)
0x81, 0x02,                    //         Input(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0xC0,                          //     EndCollection()
0x75, 0x07,                    //     ReportSize(7)
0x81, 0x03,                    //     Input(Constant, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, BitField)
0x85, 0x01,                    //     ReportId(1)
0x09, 0x02,                    //     UsageId(FirmwareTransfer[2])
0xA1, 0x02,                    //     Collection(Logical)
0x09, 0x03,                    //         UsageId(FirmwareFileId[3])
0x15, 0x01,                    //         LogicalMinimum(1)
0x25, 0x03,                    //         LogicalMaximum(3)
0x75, 0x02,                    //         ReportSize(2)
0x91, 0x02,                    //         Output(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, NonVolatile, BitField)
```

```
0x09, 0x08,                    //         UsageId(FilePayloadContainsLastBytes[8])
0x15, 0x00,                    //         LogicalMinimum(0)
0x25, 0x01,                    //         LogicalMaximum(1)
0x75, 0x01,                    //         ReportSize(1)
0x91, 0x02,                    //         Output(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, NonVolatile, BitField)
0x09, 0x04,                    //         UsageId(FileOffsetInBytes[4])
0x27, 0xFF, 0xFF, 0x00, 0x00,  //         LogicalMaximum(65,535)
0x75, 0x10,                    //         ReportSize(16)
0x91, 0x02,                    //         Output(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, NonVolatile, BitField)
0x09, 0x07,                    //         UsageId(FilePayloadSizeInBytes[7])
0x26, 0x40, 0x1F,              //         LogicalMaximum(8,000)
0x75, 0x0D,                    //         ReportSize(13)
0x91, 0x02,                    //         Output(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, NonVolatile, BitField)
0x09, 0x06,                    //         UsageId(FilePayload[6])
0x26, 0xFF, 0x00,              //         LogicalMaximum(255)
0x96, 0x40, 0x1F,              //         ReportCount(8000)
0x75, 0x08,                    //         ReportSize(8)
0x91, 0x02,                    //         Output(Data, Variable, Absolute, NoWrap, Linear,
PreferredState, NoNullPosition, NonVolatile, BitField)
0xC0,                          //     EndCollection()
0xC0,                          // EndCollection()
```