

Extension Unit Specification Sheet

(Release)

2016.01.06

Netvision Corp.

H.Yamada

1. Introduction

This document denotes the protocol of the vendor-specific Extension Unit on SVM-03 and SVM-03U board. Both SVM-03 and SVM-03U have a common extension unit interface that can be used to send setting data to the camera module through I2C, for example. The extension unit works as a part of USB Video Class (UVC) specification both can be recognized as a UVC device by a PC through USB 2.0/3.0. While in SVM-03U this UVC function is used to capture video by the host computer, the UVC function in SVM-03, which is a HDMI transmitter board, is usually used only for sending I2C and to write/read ROM data in SVM board.

In this document, all the controls of extension unit in SVM board are described in section 2. According to this list, you can send and receive I2C data with our host-side application such as NVCap (on Windows) or Linux-based application without installing any third party-made drivers. This would be also useful if you are a programmer of the host-side application. If you can control extension unit, you can easily utilize our extension unit function with this document and your SVM board.

2. Format of SVM Extension Unit Control

An extension unit can have 24 extension unit controls. In SVM architecture we use about 10 controls of them. Each control has individual (1) function, (2) data size (common for receiving and sending). A control is determined by its ID (here denoting as a term of Control Number) starting from 1. The data size is byte count in one sending / receiving transaction. The byte count used in extension unit of the SVM series is at most 32 bytes because USB2.0 driver may not support longer EP0 transaction.

2.1 I2C Send

Control Number: 1

Data Size: 32

Status: Implemented

(Host -> Device)

[D0] [D1] ... [D31]

D0[7]: Reserved (Set 0)

D0[6:0]: I2C Device Address

(The first byte of I2C packet would be generated by shifting D0 byte and setting additional W/R bit)

D1: Length of Data Body

D2 - D31: Data Body

(Even when the I2C packet is shorter than 30 bytes, the control length must be 32 bytes and you should pad zero at unused bytes.)

(Device -> Host)

D0: Send Completed

1: Completed

0: Not completed

D1 - D31: Unused

2.2 I2C Receive

Control Number: 2

Data Size: 32

Status: Implemented

(Host -> Device)

[D0] [D1] ... [D31]

D0[7]: Reserved (Set 0)

D0[6:0]: I2C Device Address

D1[7:3]: Reserved (Set 0)

D1[2:0]: Length of Preamble (set 0 for no preamble)

D2 - D[1+D1]: Preamble (usually the address of register)

D[2+D1]: Data Length To Be Received

(Device -> Host)

D0: Receive Completed Byte

1: Completed

0: Not completed

D1: Reserved

D2 - D31: Received Data

2.6 Firmware Info

Control Number: 6

Data Size: 4

Status: Implemented

(Host -> Device)

[D0] [D1] [D2] [D3]

(Device -> Host)

D0 - D1: FX3 Version Number (LSB First; unsigned)

10-89: SVM-03U

90-99: SVI-05U

100-199: SVM-03

D2 - D3: FPGA Version Number (LSB First; packed BCD)

2.7 FPGA Register Write

Control Number: 7

Data Size: 8

Status: Implemented

(Host -> Device)

[D0] ... [D7]

D0 - D3: Register Address to be Written (LSB First)

D4 - D7: Register Data to be Written (LSB First)

(Device -> Host)

(No function)

2.8 FPGA Register Read

Control Number: 8

Data Size: 4

Status: Implemented

(Host -> Device)

[D0] ... [D3]

D0 - D3: Register Address to be Read (LSB First)

(Device -> Host)

D0 - D3: Register Data (LSB First)