

上海东软载波微电子有限公司

VTHID 通讯协议

ES32F0283

修订历史

版本	修改日期	更改概要
V1.0	2022-09-02	初版

地 址：上海市徐汇区古美路 1515 号凤凰园 12 号楼 3 楼

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：http: //www.essemi.com/

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系

目录

第 1 章	VTHID 通讯协议	4
1. 1	HID Keyboard 协议.....	4
1. 1. 1	基本格式	4
1. 1. 2	Report ID	4
1. 1. 3	指令集.....	4
1. 1. 4	指令格式	5
1. 2	HID DFU 通讯协议	21
1. 2. 1	基本格式	21
1. 2. 2	Report ID	21
1. 2. 3	指令集.....	21
1. 2. 4	回应代码	21
1. 2. 5	指令格式	22

第1章 VTHID 通讯协议

此协议定义 USB Host 与 HID 装置通讯的协议格式，内容包含 Gaming Keyboard 控制及 DFU 通讯协议。

1.1 HID Keyboard 协议

此协议系以 HID Gaming 装置为基础所定义的通讯协议，协议定义灯效控制及键盘相关附属功能。

1.1.1 基本格式

所有指令都是以 Control Request 的 SET_REPORT 命令发送给 HID 装置，并透过 Interrupt In 回传结果，数据以 HEX, Little Endian 的格式发送。

SET_REPORT / Interrupt IN 的数据长度固定为 64Bytes (包含 1 Byte Report ID)，如果指令有额外的数据要传送，则在 SET_REPORT 指令之后，再以 SET_REPORT 传送数据，如果指令有额外的数据要接收，则在 SET_REPORT 指令之后，另外从对应的 Endpoint 以 Interrupt IN 接收数据。

1.1.2 Report ID

Report ID	Code	Description
OUT_REPORT_ID	0xE8	Report ID for Output Data
IN_REPORT_ID	0xE9	Report ID for Input Data

1.1.3 指令集

Command	Code	Description
GET_MODEL	0x80	Get Device Model Name
SET_SLEEP	0x01	Set Sleep Time and Lighting Mode
GET_SLEEP	0x81	Get Sleep Time and Lighting Mode
SET_KEY_MACRO	0x02	Set Single Key Macro
GET_KEY_MACRO	0x82	Get Single key Macro
SET_KEY_RGB	0x03	Set Single Key RGBW Control
GET_KEY_RGB	0x83	Get Single key RGBW Control
SET_ANIMATION	0x04	Set Active Animation
GET_ANIMATION	0x84	Get Active Animation
SET_PROFILE	0x05	Set Active Profile
GET_PROFILE	0x85	Get Active Profile
SET_RGB_MAP	0x06	Set Whole Keyboard RGB Control
GET_RGB_MAP	0x86	Get Whole Keyboard RGB Control
SET_LANGUAGE	0x07	Set Language
GET_LANGUAGE	0x87	Get Language
RESET_DEFAULT	0x0F	Reset to default setting

Command	Code	Description
SET_MACRO_DATA	0x10	Set Whole Keyboard Macro
GET_MACRO_DATA	0x90	Get Whole Keyboard Macro
SET_PROFILE_DATA	0x11	Set Whole Keyboard Profile
GET_PROFILE_DATA	0x91	Get Whole Keyboard Profile
SET_BOOTUP	0x12	Set Boot Up Animation
GET_BOOTUP	0x92	Get Boot Up Animation
SET_FLASH_DATA	0x13	Set External Flash Data
GET_FLASH_DATA	0x93	Get External Flash Data

1.1.4 指令格式

指令格式中的 Interface 系指对应的 HID Gaming 协议的 HID Interface Number

指令格式中的 Endpoint 系指对应的 HID Gaming 协议的 HID Endpoint Number

1.1.4.1 GET_MODEL (0x80)

Get device model name

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x80	0x00					
	Byte 0	Byte 1	Byte 2 ~~ Byte 15			Byte 16 ~~ Byte 63		
EP IN	0xE9	0x80	Model Name			0x00		

Model Name = ASCII string (15 characters or zero terminated if less than 15 characters)

1.1.4.2 SET_SLEEP (0x01)

Set sleep time and lighting mode

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 ~~ Byte 63			
OUT	0xE8	0x01	Sleep Time	Lighting Mode	0x00			

Sleep Time:

0x00 = Disable (Default)

0x01 = 10 minutes

0x02 = 15 minutes

0x03 = 30 minutes

0x04 = 45 minutes

0x05 = 60 minutes

0x06 = 90 minutes

0x07 = 120 minutes

Lighting Mode:

0x00 = Black (Default)

0x01 = Breathing

1. 1. 4. 3 GET_SLEEP (0x81)

Get sleep time and lighting mode

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x81	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 ~~ Byte 63			
EP IN	0xE9	0x81	Sleep Time	Lighting Mode	0x00			

Sleep Time:

0x00 = Disable (Default)

0x01 = 10 minutes

0x02 = 15 minutes

0x03 = 30 minutes

0x04 = 45 minutes

0x05 = 60 minutes

0x06 = 90 minutes

0x07 = 120 minutes

Lighting Mode:

0x00 = Black (Default)

0x01 = Breathing

1. 1. 4. 4 SET_KEY_MACRO (0x02)

Set single key macro

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5 ~~ Byte 63		
OUT	0xE8	0x02	Key Index	Mode	Code	0x00		

Key Index = 0x00 ~ 0x7F (128 keys)

Mode:

0x01 = Normal Key

0x02 = User Key

0x03 = Macro Key

Code:

Mode = 0x01 (Normal): Key Code = 0x00

Mode = 0x02 (User): Key Code = User Code

Mode = 0x03 (Macro): Key Code = Macro #

1. 1. 4. 5 GET_KEY_MACRO (0x82)

Get single key macro

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
OUT	0xE8	0x82	Key Index	0x00				
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5 ~~ Byte 63		
EP IN	0xE9	0x82	Key Index	Mode	Code	0x00		

Key Index = 0x00 ~ 0x7F (128 keys)

Mode:

0x01 = Normal Key

0x02 = User Key

0x03 = Macro Key

Code:

Mode = 0x01 (Normal): Key Code = 0x00

Mode = 0x02 (User): Key Code = User Code

Mode = 0x03 (Macro): Key Code = Macro #

1. 1. 4. 6 SET_KEY_RGB (0x03)

Set single key RGB control

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x03	Key Index	R Color	G Color	B Color	W Data	0x00
	Byte 8 ~~ Byte 63							
OUT	0x00							

Key Index = 0x00 ~ 0x7F (128 keys)

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

1. 1. 4. 7 GET_KEY_RGB (0x83)

Get single key RGB control

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
OUT	0xE8	0x83	Key Index	0x00				
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x83	Key Index	R Color	G Color	B Color	W Data	0x00
	Byte 8 ~~ Byte 63							
EP IN	0x00							

Key Index = 0x00 ~ 0x7F (128 keys)

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

1. 1. 4. 8 SET_ANIMATION (0x04)

Set active animation

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x04	Animation ID	R Color	G Color	B Color	W Data	Speed
	Byte 8	Byte 9	Byte 10 ~~ Byte 63					
OUT	Display	Direction	0x00					

Animation ID:

0x00 = Off (Default)

0x01 = Solid

0x02 = Breathing

0x03 = Reactive

0x04 = Rain

0x05 = Gradient

0x06 = Fade

0x07 = Ripple

0x08 = Wave

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFf

W Data = Brightness: 0x0 ~ 0x0A (0% ~ 100%)

Speed = 0x01 ~ 0xFF (Units in 10ms)

Display:

0x01 = Show RGB color (Default)

0x02 = Rainbow (based on RGB color)

0x03 = Random (based on RGB color)

0x1x = Divide into x segments to display different color (based on RGB color)

Direction:

0x01 = Left to Right (Default)

0x02 = Right to Left

0x03 = Top to Bottom

0x04 = Bottom to Top

1. 1. 4. 9 GET_ANIMATION (0x84)

Get active animation

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x84	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x84	Animation ID	R Color	G Color	B Color	W Data	Speed
	Byte 8	Byte 9	Byte 10 ~~ Byte 63					
EP IN	Display	Direction	0x00					

Animation ID:

0x00 = Off (Default)

0x01 = Solid

0x02 = Breathing

0x03 = Reactive

0x04 = Rain

0x05 = Gradient

0x06 = Fade

0x07 = Ripple

0x08 = Wave

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x0 ~ 0x0A (0% ~ 100%)

Speed = 0x01 ~ 0xFF (Units in 10ms)

Display:

0x01 = Show RGB color (Default)

0x02 = Rainbow (based on RGB color)

0x03 = Random (based on RGB color)

0x1x = Divide into x segments to display different color (based on RGB color)

Direction:

0x01 = Left to Right (Default)

0x02 = Right to Left

0x03 = Top to Bottom

0x04 = Bottom to Top

1. 1. 4. 10 SET_PROFILE (0x05)

Set active profile

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
OUT	0xE8	0x05	Profile #	0x00				

Profile # = 0x00 ~ 0x04 (0x00 = Disable)

1. 1. 4. 11 GET_PROFILE (0x85)

Set active profile

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x85	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
EP IN	0xE9	0x85	Profile #	0x00				

Profile # = 0x00 ~ 0x04 (0x00 = Disable)

1. 1. 4. 12 SET_RGB_MAP (0x06)

Set whole keyboard RGB control, based on keyboard model to define key numbers.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
OUT	0xE8	0x06	Index	0x00	Key N Map	~~	Key N+10 Map	0x00

Index = 0x01 ~ 0x0C (128 Keys), Bit7 = 1 means the last OUT packet

Map Data (R, G, B, W):

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

1. 1. 4. 13 GET_RGB_MAP (0x86)

Get whole keyboard RGB control, based on keyboard model to define key numbers.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1~~ Byte 63						
OUT	0xE8	0x86	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
EP IN	0xE9	0x86	Index	0x00	Key N Map	~~	Key N+10 Map	0x00

Index = 0x01 ~ 0x0C (128 Keys), Bit7 = 1 means the last IN packet

Map Data (R, G, B, W) :

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

1. 1. 4. 14 SET_LANGUAGE (0x07)

Set language mode

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
OUT	0xE8	0x07	Language	0x00				

Language:

0x01 = US

0x02 = UK

0x03 = JP

0x04 = KR

1. 1. 4. 15 GET_LANGUAGE (0x87)

Get language mode

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x87	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
EP IN	0xE9	0x87	Language	0x00				

Language :

0x01 = US

0x02 = UK

0x03 = JP

0x04 = KR

1. 1. 4. 16 RESET_DEFAULT (0x0F)

Reset to default setting

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x0F	0x00					

1. 1. 4. 17 SET_MACRO_DATA (0x10)

Set macro keys combination, support for 120 keys combination.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
OUT	0xE8	0x10	Macro #	Index	Key N Macro	~~	Key N+10 Macro	0x00

Macro # = 0x01 ~ 0x40

Index = 0x01 ~ 0x0B (120 Keys), Bit7 = 1 means the last OUT packet

Macro Data (Code, Mode, Times, Reserved):

Code = Key code

Mode: Mode = 0x01: Key make then release

Mode = 0x02: Key make

Mode = 0x03: Key release

Times = Delay time between two key code, delay time = times * 20ms

Reserved = 0x00

1. 1. 4. 18 GET_MACRO_DATA (0x90)

Get macro keys combination, support for 120 keys combination.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~~ Byte 63				
OUT	0xE8	0x90	Macro #	0x00				
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
EP IN	0xE9	0x90	Macro #	Index	Key N Macro	~~	Key N+10 Macro	0x00

Macro # = 0x01 ~ 0x40

Index = 0x01 ~ 0x0B (120 Keys), Bit7 = 1 means the last IN packet

Macro Data (Code, Mode, Times, Reserved):

Code = Key code

Mode:

Mode = 0x01: Key make then release

Mode = 0x02: Key make

Mode = 0x03: Key release

Times = Delay time between two key code, delay time = times * 20ms

Reserved = 0x00

1. 1. 4. 19 SET_PROFILE_DATA (0x11)

Set whole key profile configuration, based on keyboard model to define key numbers.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x11	Profile #	0x1	Animation ID	R Color	G Color	B Color
	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12 ~~ Byte 63			
OUT	W Data	Speed	Display	Direction	0x00			
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
OUT	0xE8	0x11	Profile #	Index	Key N Profile	~~	Key N+10 Profile	0x00

Profile # = 0x01 ~ 0x04

Index = 0x02 ~ 0x0D (128 Keys), Bit7 = 1 means the last OUT packet

Animation ID:

0x00 = Off (Default)

0x01 = Solid
0x02 = Breathing
0x03 = Reactive
0x04 = Rain
0x05 = Gradient
0x06 = Fade
0x07 = Ripple
0x08 = Wave

R Color = 0x00 ~ 0xFF
G Color = 0x00 ~ 0xFF
B Color = 0x00 ~ 0xFF
W Data = Brightness: 0x0 ~ 0x0A (0% ~ 100%)
Speed = 0x01 ~ 0xFF (Units in 10ms)

Display:

0x01 = Show RGB color (Default)
0x02 = Rainbow (based on RGB color)
0x03 = Random (based on RGB color)
0x1x = Divide into x segments to display different color (based on RGB color)

Direction:

0x01 = Left to Right (Default)
0x02 = Right to Left
0x03 = Top to Bottom
0x04 = Bottom to Top

Profile Data (Index, Modex, Code, Reserved):

Key Index = 0x00 ~ 0x7F (128 keys), Index = 0xFF means the last key

Mode:

Mode = 0x01: Normal Key
Mode = 0x02 = User Key
Mode = 0x03 = Macro Key

Code:

Mode = 0x01 (Normal): Code = 0x00
Mode = 0x02 (User): Code = User Key Code
Mode = 0x03 (Macro): Code = Macro #
Reserved = 0x00

1. 1. 4. 20 GET_PROFILE_DATA (0x91)

Get whole key profile configuration, based on keyboard model to define key numbers.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3 ~ Byte 63				
OUT	0xE8	0x91	Profile #	0x00				
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x91	Profile #	0x1	Animation ID	R Color	G Color	B Color
	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12 ~ Byte 63			
EP IN	W Data	Speed	Display	Direction	0x00			
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~7	~~	Byte 44~47	Byte 48~63
EP IN	0xE9	0x91	Profile #	Index	Key N Profile	~~	Key N+10 Profile	0x00

Profile # = 0x01 ~ 0x04

Index = 0x02 ~ 0x0D (128 Keys), Bit7 = 1 means the last OUT packet

Animation ID:

0x00 = Off (Default)

0x01 = Solid

0x02 = Breathing

0x03 = Reactive

0x04 = Rain

0x05 = Gradient

0x06 = Fade

0x07 = Ripple

0x08 = Wave

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x0 ~ 0x0A (0% ~ 100%)

Speed = 0x01 ~ 0xFF (Units in 10ms)

Display:

0x01 = Show RGB color (Default)

0x02 = Rainbow (based on RGB color)

0x03 = Random (based on RGB color)

0x1x = Divide into x segments to display different color (based on RGB color)

Direction:

0x01 = Left to Right (Default)

0x02 = Right to Left

0x03 = Top to Bottom

0x04 = Bottom to Top

Profile Data (Index, Modex, Code, Reserved):

Key Index = 0x00 ~ 0x7F (128 keys), Index = 0xFF means the last key

Mode:

Mode = 0x01: Normal Key

Mode = 0x02 = User Key

Mode = 0x03 = Macro Key

Code:

Mode = 0x01 (Normal): Code = 0x00

Mode = 0x02 (User): Code = User Key Code

Mode = 0x03 (Macro): Code = Macro #

Reserved = 0x00

1. 1. 4. 21 SET_BOOTUP (0x12)

Set boot up key animation, support for maximum 56 keys animation.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~9	~~	Byte 40~45	Byte 46~64
OUT	0xE8	0x12	Index	0x00	BootUp N	~~	BootUp N+6	0x00

Index = 0x01 ~ 0x08 (56 Keys), Bit7 = 1 means the last OUT packet

Bootup Data:

Key Index = 0x00 ~ 0x7F (128 keys), Index = 0xFF means the end key

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

Times = Delay time between two key code, delay time = times * 20ms

1. 1. 4. 22 GET_BOOTUP (0x92)

Get boot up key animation, support for maximum 56 keys animation.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2 ~~ Byte 63					
OUT	0xE8	0x92	0x00					
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4~9	~~	Byte 40~45	Byte 46~64
EP IN	0xE9	0x92	Index	0x00	BootUp N	~~	BootUp N+6	0x00

Index = 0x01 ~ 0x08 (56 Keys), Bit7 = 1 means the last IN packet

Bootup Data:

Key Index = 0x00 ~ 0x7F (128 keys), Index = 0xFF means the end key

R Color = 0x00 ~ 0xFF

G Color = 0x00 ~ 0xFF

B Color = 0x00 ~ 0xFF

W Data = Brightness: 0x00 ~ 0x0A (0% ~ 100%)

Times = Delay time between two key code, delay time = times * 20ms

1. 1. 4. 23 SET_FLASH_DATA (0x13)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x13	Address 0	Address 1	Address 2	Address 3	Count 0	Count 1
	Byte 8 ~ Byte 39				Byte 40 ~ Byte 63			
OUT	Packet Data 0~31				0x00			

Address = 0x00000000 ~ 0xFF000000

Address = 0xFF000000 to erase all flash

Count = 0x0000 ~ 0x0020

1. 1. 4. 24 GET_FLASH_DATA (0x93)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x93	Address 0	Address 1	Address 2	Address 3	Count 0	Count 1
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x93	Address 0	Address 1	Address 2	Address 3	Count 0	Count 1
	Byte 8 ~ Byte 39				Byte 40 ~ Byte 63			
OUT	Packet Data 0~31				0x00			

Address = 0x00000000 ~ 0xFF000000

Count = 0x0000 ~ 0x0020

1. 1. 4. 25 Key Index

0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
ESC	NULL	F1	F2	F3	F4	F5	F6
0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
F7	F8	F9	F10	F11	F12	PrnScr	Scroll
0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
Pause	Null	Null	Null	Null	`	1	2
0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
3	4	5	6	7	8	9	0
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
-	=	BKSpace	Insert	Home	PgUp	NumLock	/
0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
*	-	Tab	Q	W	E	R	T
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
Y	U	I	O	P	[]	\
0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
Del	End	PgDn	7	8	9	+	CapsLock
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
A	S	D	F	G	H	J	K
0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
L	;	'	Enter	Null	Null	Null	Null
0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
4	5	6	+	SHift	Z	X	C
0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
V	B	N	M	,	.	/	Null
0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
Null	Shift	Null	Up	Null	1	2	3
0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
Enter	Ctrl	GUI	Alt	Null	Null	Space	Null
0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
Null	Null	Alt	Fn	App	Null	Ctrl	Left
0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
Down	Right	0	Null	.	Null	Null	Null

1. 1. 4. 26 Animation ID

ID	Animation	Description
0x00	Off	不显示灯效
0x01	Solid	以单一颜色显示
0x02	Breathe	以单一颜色由亮至暗再由暗至亮重复显示
0x03	Reactive	只在按键显示再逐渐变暗
0x04	Rain	以单一颜色由上至下延伸
0x05	Gradient	以渐层的方式分区显示
0x06	Fade	以渐暗再渐亮的方次切换至另一个颜色
0x07	Ripple	以按键为中心向外扩散
0x08	Wave	以水波纹的效果向同一方向移动

1. 1. 4. 27 Example Descriptor Data

定义一个对应 Gaming 协议的 HID Interface, Report Descriptor 及 Endpoint Descriptor
 定义一个对应 Gaming 协议的 Report Descriptor, Report Size = 8, Report Count = 64
 其中, Report Descriptor 需定义 Output Report ID 为 0xE8, Input Report ID 为 0xE9
 Interface Descriptor 及 Report Descriptor 参考如下:

Interface Descriptor:

```

0x09,          //Length=9
0x04,          //Type=4
Interface Number, //Interface Number
0x00,          //Alternate Setting
0x02,          //Number of Endpoints
0x03,          //HID class
0x00,          //Sub class
0x00,          //Protocol
0x00,          //String index
//HID descriptor
0x09,          //Length=9
0x21,          //Type=0x21
0x0111,        //bcdHID
0x00,          //bCountryCode
0x01,          //Number of HID class descriptors
0x22,          //Report descriptor type
0x1d,          //Total length of Report descriptor
//Endpoint In Descriptor
0x07,          //length=7
0x05,          //type=5
    
```

```
(0x80 | EP_IN),          //EPx, In
0x03,                    //Interrupt
0x40,                    //Max packet size=64
0x01,                    //Interval=1ms
//Endpoint Out Descriptor
0x07,                    //length=7
0x05,                    //type=5
EP_OUT,                  //EPx, Out
0x03,                    //Interrupt
0x40,                    //Max packet size=64
0x01,                    //Interval=1ms
```

Report Descriptor:

```
0x06, 0xf0, 0xff,        //Usage Page (Vendor)
0x09, 0xe0,              //  Usage (Vendor_Control)
0xa1, 0x01,              //Collection (Application)
0x15, 0x00,              //  Logical Minimum (0)
0x26, 0xff, 0x00,        //  Logical Maximum (255)
0x75, 0x08,              //  Report Size (8)
0x95, 0x3f,              //  Report Count (63)
//Vendor Write INT
0x09, 0xe8,              //  Usage (Write_INT)
0x85, 0xe8,              //  Report ID (0xe8)
0x91, 0x02,              //  Output (Absolute, Variable, Data)
//Vendor Read INT
0x09, 0xe9,              //  Usage (Read_INT)
0x85, 0xe9,              //  Report ID (0xe9)
0x81, 0x02,              //  Input (Absolute, Variable, Data)
0xc0,                    //End Collection (Application)
```

1.2 HID DFU 通讯协议

此在线更新协议系以 HID 装置为基础所定义的通讯协议，装置依据实际应用可区分为 APP 装置及 IAP 装置。

APP 装置系指实际应用的 USB 装置，由于 DFU 本身为自有协议，因此 APP 装置需定义一组对应 DFU 协议的 Interface Descriptor。

IAP 装置本身即为实现 DFU 更新协议的 HID 装置。

1.2.1 基本格式

HOST AP 透过 Control Request 的 SET_REPORT 将命令发送给 HID 装置，执行结果则透过 Interrupt In 回传，数据均以 HEX, Little Endian 的格式发送，数据长度固定为 64Bytes (包含 1 Byte Report ID)

如果指令有额外的数据要传送(WRITE_FLASH / READ_FLASH)，则在命令发送之后，另以 SET_REPORT 的格式发送数据，并以 Interrupt In 回传执行结果

1.2.2 Report ID

Report ID	Code	Description
OUT_REPORT_ID	0xE8	Report ID for Output Data
IN_REPORT_ID	0xE9	Report ID for Input Data

1.2.3 指令集

Command	Code	Description
GET_CONFIG	0xE0	Get BLDR Address and Firmware Version (APP)
SET_RANGE	0x61	Set IAP Write Address and IAP Size (IAP)
GET_RANGE	0xE1	Get Available IAP Address and IAP Size (IAP)
WRITE_FLASH	0x62	Write Data to Flash (IAP)
READ_FLASH	0xE2	Read Data from Flash (IAP)
READ_CHECKSUM	0xE3	Read Flash Checksum (IAP)
RUN_APP	0x68	Run Application (IAP / APP)

1.2.4 回应代码

Response Code	Description
0x00	Success
0x41	Invalid Command
0x42	Parameter Error
0x43	Address Error
0x44	Size Error

1.2.5 指令格式

指令格式中的 Interface 系指对应的 HID DFU 协议的 HID Interface Descriptor，对于 IAP 装置的指令而言，Interface = 0，对于 APP 装置的指令而言，Interface 需视实际应用调整。

1.2.5.1 GET CONFIG (0xE0, APP)

GET_CONFIG 指令只适用于 APP 装置，主要是取得 IAP 的 Flash 地址及应用装置的版本信息。IAP 的 Flash 地址可作为对 APP 装置发 RUN_APP 的地址参数。

Parameter: None

Response: Command + Response Code + BLDR Address (4 Bytes) + FW Version (4 Bytes)

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte	~~		Byte 61	Byte 62	Byte 63
OUT	0xE8	0xE0	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0xE0	Response	Address 0	Address 1	Address 2	Address 3	Version 0
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
EP IN	Version 1	Version 2	Version 3	~~		0x00	0x00	0x00

1.2.5.2 SET RANGE (0x61, IAP)

SET_RANGE 指令主要是告知 IAP 应用程序要刻录的地址及占用 Flash 的大小，如果要刻录的地址及大小超出 IAP 的范围，IAP 将回应 Address Error 或 Size Error。

透过 GET_RANGE 指令可以取得 IAP 的 Flash 起始地址及大小。

Parameter: APP Start Address (4 Bytes) + APP Size (4 Bytes)

Response: Command + Response Code

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x61	Address 0	Address 1	Address 2	Address 3	Range 0	Range 1
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
OUT	Range 2	Range 3	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x61	Response	0x00	0x00	0x00	0x00	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
EP IN	0x00	0x00	0x00	~~		0x00	0x00	0x00

1. 2. 5. 3 GET RANGE (0xE1, IAP)

GET_RANGE 指令主要是取得 IAP 可刻录的 Flash 起始地址及大小，应用程序可依据此讯息判断要更新的程序是否超出此范围。

Parameter: None

Response: Command + Response Code + IAP Address (4 Bytes) + IAP Size (4 Bytes)

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0xE1	0x00	0x00	0x00	0x00	0x00	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
OUT	Range 3	0x00	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0xE1	Response	Address 0	Address 1	Address 2	Address 3	Range 0
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
EP IN	Range 1	Range 2	Range 3	~~		0x00	0x00	0x00

1. 2. 5. 4 WRITE FLASH (0x62, IAP)

WRITE_FLASH 指令指示 IAP 将对应的数据写入 Flash，参数中的 Address 及 Byte Count 对应写入的 Flash 地址及写入的位数，当 HostAP 下完 WRITE FLASH (0x62)的指令后，即透过 SET_REPORT 发送数据封包，每笔数据封包最大为 32Bytes，在发送完所有数据封包后，Host AP 即透过 Interrupt In 读取 IAP 的执行结果。

Parameter: Write Address (4 Bytes) + Write Byte Count (2 Bytes) + Write Data

Response: Command + Response Code

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x62	Address 0	Address 1	Address 2	Address 3	Count 0	Count 1
	Byte 8	Byte 9	Byte 10	~~		Byte Count+5	Byte Count+6	Byte Count+7
OUT	Data 0	Data 1	Data 2	~~		Data Count-3	Data Count-2	Data Count-1
	Byte Count+8	Byte Count+9	Byte Count+10	~~		Byte 61	Byte 62	Byte 63
OUT	0x00	0x00	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0x62	Response	0x00	0x00	0x00	0x00	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte 13	Byte 14	Byte 15
EP IN	0x00	0x00	0x00	~~		0x00	0x00	0x00

1. 2. 5. 5 READ FLASH (0xE2, IAP)

READ_FLASH 指令指示 IAP 将 Flash 对应的数据回传，参数中的 Address 及 Byte Count 对应读取的 Flash 地址及读取的位数，当 Host AP 下完 READ FLASH (0xE2)的指令后，即透过 Interrupt In 读取指令的执行结果读取，如果结果正确(Response Code = 0x00)，Host AP 即透过 Interrupt In 读取数据封包，每笔数据封包最大为 32Bytes。

Parameter: Read Address (4 Bytes) + Read Byte Count (2 Bytes)

Response: Command + Response Code + Read Data

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0xE2	Address 0	Address 1	Address 2	Address 3	Count 0	Count 1
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
OUT	0x00	0x00	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0xE2	Response	0x00	0x00	0x00	0x00	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte Count+5	Byte Count+6	Byte Count+7
EP IN	Data 0	Data 1	Data 2	~~		Data Count-3	Data Count-2	Data Count-1
	Byte Count+8	Byte Count+9	Byte Count+10	~~		Byte 61	Byte 62	Byte 63
EP IN	0x00	0x00	0x00	~~		0x00	0x00	0x00

1.2.5.6 READ CHECKSUM (0xE3, IAP)

READ_CHECKSUM 指令用于取得 Flash 区块的数据加总和, 除了以 READ_FLASH 来确认数据刻录的正确性外, 亦可经由读取 Flash 区块的 CheckSum 来判断数据刻录的正确性, CheckSum 以 32 位的数据发送。

Parameter: Address (4 Bytes) + Size (4 Bytes)

Response: Command + Response Code + CheckSum (4 Bytes)

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0xE3	Address 0	Address 1	Address 2	Address 3	Range 0	Range 1
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
OUT	Range 2	Range 3	0x00	~~		0x00	0x00	0x00

Response	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EP IN	0xE9	0xE3	Response	checksum 0	checksum 1	checksum 2	checksum 3	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
EP IN	0x00	0x00	0x00	~~		0x00	0x00	0x00

1.2.5.7 RUN APP (0x68, APP / IAP)

RUN_APP 指令可在不重新插拔 USB 连接线的状况下直接跳转至指定的地址执行应用程序, 参数中的 Address 即为要跳转的地址, 此指令可直接由 Application 跳转至 IAP, 亦可由 IAP 跳转至 Application, 由于 IAP 收到 RUN_APP 指令后会立即执行跳转, 因此 RUN_APP 无须响应执行结果。

从 Application 跳转至 IAP 的 Address 可经由 GET_CONFIG(0xE0)取得。

Parameter: Run Address (4 Bytes)

Response: No Response for RUN_APP

Command	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
SETUP	0x21	0x09	0x00	0x02	Interface	0x00	0x40	0x00
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
OUT	0xE8	0x68	Address 0	Address 1	Address 2	Address 3	0x00	0x00
	Byte 8	Byte 9	Byte 10	~~		Byte 61	Byte 62	Byte 63
OUT	0x00	0x00	0x00	~~		0x00	0x00	0x00

1. 2. 5. 8 Example DFU Descriptor Data

定义一个对应 DFU 协议的 HID Interface, Report Descriptor 及 Endpoint Descriptor。
定义一个对应 DFU 协议的 Report Descriptor, Report Size = 8, Report Count = 63。
其中, Report Descriptor 需定义 Output Report ID 为 0xE8, Input Report ID 为 0xE9。
Interface Descriptor 及 Report Descriptor 参考如下:

Interface Descriptor:

```

0x09,           //Length=9
0x04,           //Type=4
Interface Number, //Interface Number
0x00,           //Alternate Setting
0x02,           //Number of Endpoints
0x03,           //HID class
0x00,           //Sub class
0x00,           //Protocol
0x00,           //String index
//HID descriptor
0x09,           //Length=9
0x21,           //Type=0x21
0x0111,         //bcdHID
0x00,           //bCountryCode
0x01,           //Number of HID class descriptors
0x22,           //Report descriptor type
0x1d,           //Total length of Report descriptor
//Endpoint In Descriptor
0x07,           //length=7
0x05,           //type=5
(0x80 | EP_IN), //EPx, In
0x03,           //Interrupt
0x40,           //Max packet size=64
0x01,           //Interval=1ms
//Endpoint Out Descriptor
0x07,           //length=7
0x05,           //type=5
EP_OUT,         //EPx, Out
0x03,           //Interrupt
0x40,           //Max packet size=64
0x01,           //Interval=1ms

```

Report Descriptor:

```
0x06, 0xf0, 0xff,      //Usage Page (Vendor)
0x09, 0xe0,            //   Usage (Vendor_Control)
0xa1, 0x01,            //Collection (Application)
0x15, 0x00,            //   Logical Minimum (0)
0x26, 0xff, 0x00,      //   Logical Maximum (255)
0x75, 0x08,            //   Report Size (8)
0x95, 0x3f,            //   Report Count (63)
//Vendor Write INT
0x09, 0xe8,            //   Usage (Write_INT)
0x85, 0xe8,            //   Report ID (0xe8)
0x91, 0x02,            //   Output (Absolute, Variable, Data)
//Vendor Read INT
0x09, 0xe9,            //   Usage (Read_INT)
0x85, 0xe9,            //   Report ID (0xe9)
0x81, 0x02,            //   Input (Absolute, Variable, Data)
0xc0,                  //End Collection (Application)
```

1. 2. 5. 9 Example APP Interface Descriptor

对于 APP 而言，需定义一个对应 DFU 协议的 HID Interface，Endpoint 及 Report Descriptor。其中，Report Descriptor 需定义 Output Report ID 为 0xE8, Input Report ID 为 0xE9。

Descriptor Data 参考如下：

Interface Descriptor:

```
0x09,                  //Length=9
0x04,                  //Type=4
Interface Number,      //Interface Number
0x00,                  //Alternate Setting
0x01,                  //Number of Endpoints
0x03,                  //HID class
0x00,                  //Sub class
0x00,                  //Protocol
0x00,                  //String index
//HID descriptor
0x09,                  //Length=9
0x21,                  //Type=0x21
0x0111,                //bcdHID
0x00,                  //bCountryCode
0x01,                  //Number of HID class descriptors
```

```
0x22,           //Report descriptor type
0x1d,           //Total length of Report descriptor
//Endpoint In Descriptor
0x07,           //Length=7
0x05,           //Type=5
(0x80 | EP_IN), //EPx, In
0x03,           //Interrupt
0x40,           //Max Packet Size=64
0x01,           //Interval=16ms
```

Report Descriptor:

```
0x06, 0xf0, 0xff, //Usage Page (Vendor)
0x09, 0xe0,       //  Usage (Vendor_Control)
0xa1, 0x01,       //Collection (Application)
0x15, 0x00,       //  Logical Minimum (0)
0x26, 0xff, 0x00, //  Logical Maximum (255)
0x75, 0x08,       //  Report Size (8)
0x95, 0x3f,       //  Report Count (63)
//Vendor Write Int
0x09, 0xe8,       //  Usage (Write_INT)
0x85, 0xe8,       //  Report ID (0xe8)
0x91, 0x02,       //  Output (Absolute, Variable, Data)
//Vendor Read Int
0x09, 0xe9,       //  Usage (Read_INT)
0x85, 0xe9,       //  Report ID (0xe9)
0x81, 0x02,       //  Input (Absolute, Variable, Data)
0xc0,             //End Collection (Application)
```