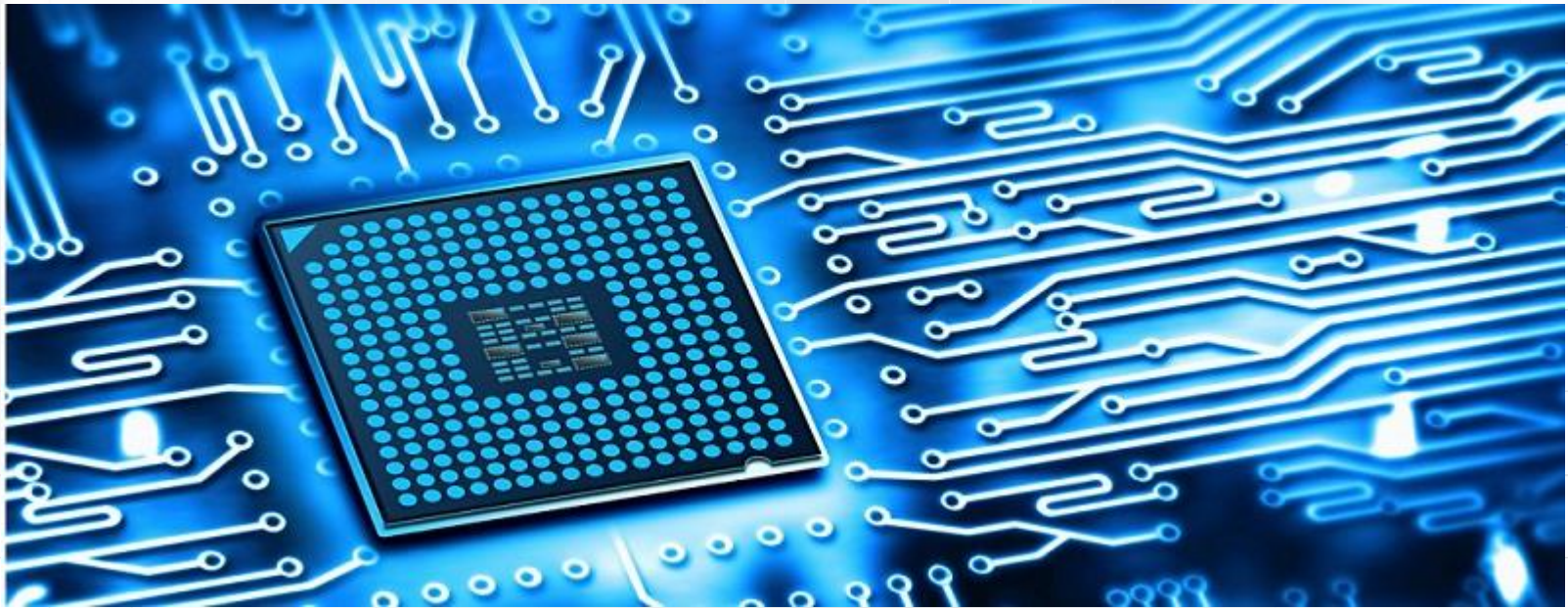




Politechnika  
Wroclawska

# Wbudowane systemy operacyjne



HR EXCELLENCE IN RESEARCH



Politechnika  
Wroclawska



# Wbudowane systemy operacyjne

HID cz. II

Dr inż. Damian Radzewicz



HR EXCELLENCE IN RESEARCH

Wrocław 2018



# Plan wykładu

## Urządzenia HID

- Obsługa systemowa urządzenia HID
- Przykład deskryptorów HID

# Obsługa systemowa urządzenia HID

Requesty standardowe:

- ❖ Get\_Descriptor Request
- ❖ Set\_Descriptor Request

Requesty specyficzne dla Klasy:

- ❖ Get\_Report Request
- ❖ Set\_Report Request
- ❖ Get\_Idle Request
- ❖ Set\_Idle Request
- ❖ Get\_Protocol Request
- ❖ Set\_Protocol Request

# Obsługa systemowa urządzenia HID

## Get\_Descriptor Request

Zwraca Deskryptor urządzenia

<b>Pole</b>	<b>Opis (klasa HID)</b>
<i>bmRequestType</i>	10000001
<i>bRequest</i>	GET_DESCRIPTOR (0x06)
<i>wValue</i>	Descriptor Type i Descriptor Index
<i>wIndex</i>	Interface Number
<i>wLength</i>	Descriptor Length
<i>Data</i>	Descriptor

# Obsługa systemowa urządzenia HID

## Get\_Descriptor Request

Zwraca Deskryptor urządzenia

Pole	Opis (klasa HID)
<i>bmRequestType</i>	10000001
<i>bRequest</i>	GET_DESCRIPTOR (0x06)
<i>wValue</i>	Descriptor Type i Descriptor Index
<i>wIndex</i>	Interface Number
<i>wLength</i>	Descriptor Length
<i>Data</i>	Descriptor

Starszy bajt

Młodszy bajt

Liczba interfejsów urządzenia

### Descriptor Type

7 Reserved (powinien być 0)

6..5 Typ

0=Standard

1=Class

2=Vendor

3=Reserved

4..0 Descriptor

Descriptor Index

Ustalona wartość – Physical Descriptor

0 – inne klasy deskryptora HID

# Obsługa systemowa urządzenia HID

## Set\_Descriptor Request

Oznacza żądanie zmiany deskryptorów urządzenia – request opcjonalny

<b>Pole</b>	<b>Opis (klasa HID)</b>
<i>bmRequestType</i>	00000001
<i>bRequest</i>	SET_DESCRIPTOR (0x07)
<i>wValue</i>	Descriptor Type i Descriptor Index
<i>wIndex</i>	Interface Number
<i>wLength</i>	Descriptor Length
<i>Data</i>	Descriptor

# Obsługa systemowa urządzenia HID

## Get\_Report Request

Umożliwia uzyskanie informacji o urządzeniu przez Control pipe.

Pole	Opis (klasa HID)
<i>bmRequestType</i>	10100001
<i>bRequest</i>	GET_REPORT
<i>wValue</i>	Report Type i Report ID
<i>wIndex</i>	Interface
<i>wLength</i>	Report Length
<i>Data</i>	Report

Starszy bajt

Młodszy bajt (0 jeśli jest nie używany)

### Report Type

01	Input
02	Output
03	Feature
04-FF	Reserved

Do raportów Input stosowana jest Interrupt In pipe

Do raportów Output można stosować Interrupt Out pipe  
ale jeśli nie jest zdefiniowana stosuje się Control pipe



# Obsługa systemowa urządzenia HID

## Set\_Report Request

Umożliwia przesłanie do urządzenia raportu w celu ustawienia stanu input, output lub feature.

Pole	Opis (klasa HID)
<i>bmRequestType</i>	00100001
<i>bRequest</i>	SET_REPORT
<i>wValue</i>	Report Type i Report ID
<i>wIndex</i>	Interface
<i>wLength</i>	Report Length
<i>Data</i>	Report

Znaczenie poszczególnych pól jest takie same jak w Get\_Report Request

Urządzenie może zignorować ten request

Alternatywnie ten request może służyć do resetowania ustawień

# Obsługa systemowa urządzenia HID

## Get\_Idle Request

Odczytuje parametry stanu bezczynności konkretnego raportu wejściowego.

<b>Pole</b>	<b>Opis (klasa HID)</b>
<i>bmRequestType</i>	10100001
<i>bRequest</i>	GET_IDLE
<i>wValue</i>	0 (zero) i Report ID
<i>wIndex</i>	Interface
<i>wLength</i>	1 (one)
<i>Data</i>	Idle rate

# Obsługa systemowa urządzenia HID

## Set\_Idle Request

Ustawia częstotliwość pracy konkretnego raportu wejściowego. Np. urządzenie działające z dwoma raportami wejściowymi może mieć ustawione dwa różne czasy bezczynności – 20 ms dla raportu ID1 oraz 500 ms dla raportu ID2.

Pole	Opis (klasa HID)
<i>bmRequestType</i>	00100001
<i>bRequest</i>	SET_IDLE
<i>wValue</i>	Duration i Report ID
<i>wIndex</i>	Interface
<i>wLength</i>	0 (zero)
<i>Data</i>	Not applicable

Rekomendowane wartości czasu bezczynności dla urządzeń HID to:  
500 ms (klawiatura), nieskończoność (myszka i joystick).

# Obsługa systemowa urządzenia HID

## Get\_Protocol Request

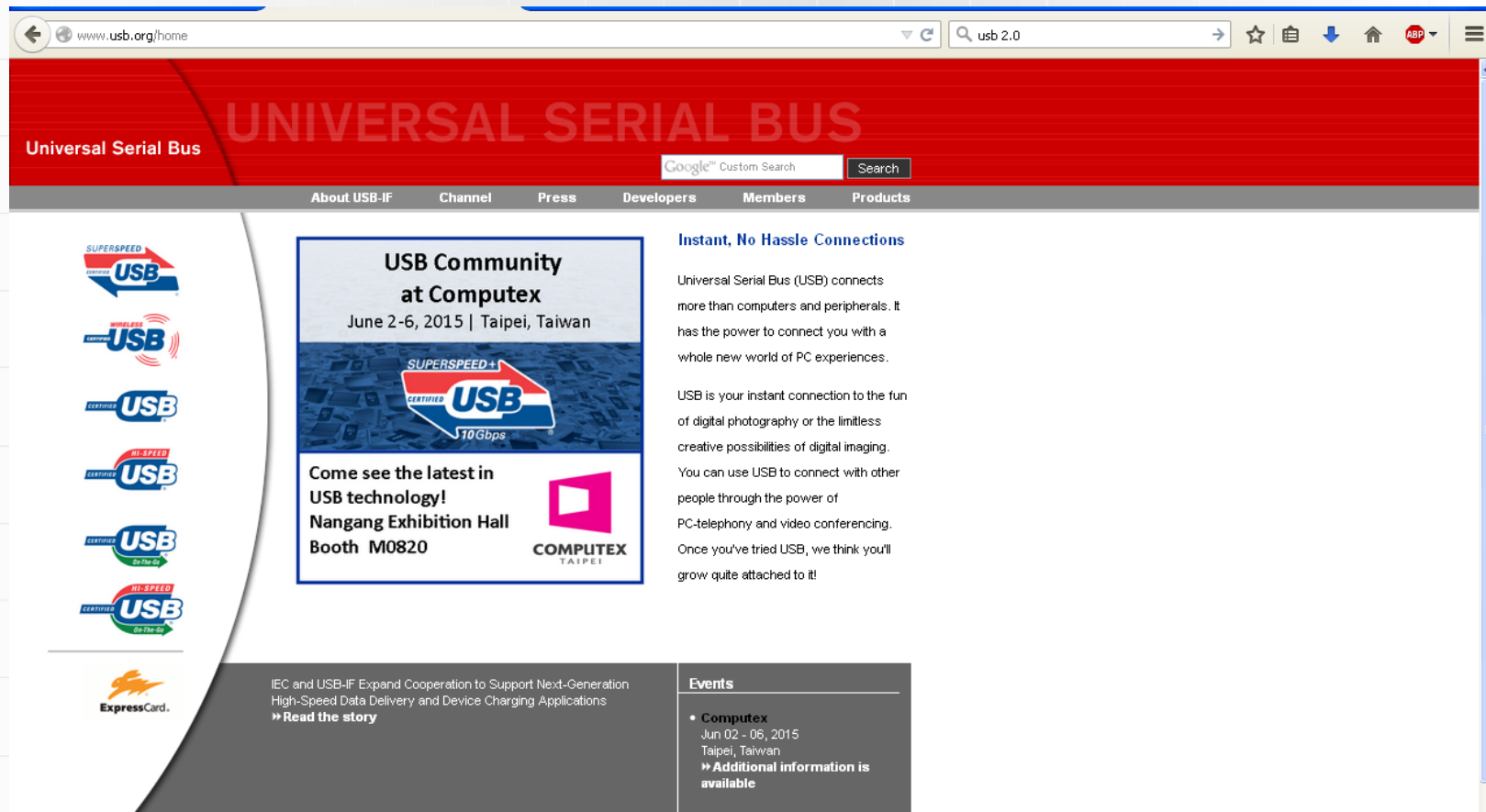
Odczytuje który z protokołów jest aktywny. Ten typ requestu jest również obsługiwany przez urządzenia z Boot subklasy.

Pole	Opis (klasa HID)
<i>bmRequestType</i>	10100001
<i>bRequest</i>	GET_PROTOCOL
<i>wValue</i>	0 (zero)
<i>wIndex</i>	Interface
<i>wLength</i>	1 (one)
<i>Data</i>	0 = Boot Protocol 1 = Report Protocol

# Obsługa systemowa urządzenia HID

<http://www.usb.org/home/>

<http://www.usb.org/developers/docs/>



The screenshot shows the homepage of the Universal Serial Bus (USB) website. The browser address bar displays "www.usb.org/home" and the search bar contains "usb 2.0". The main header features the text "UNIVERSAL SERIAL BUS" in large, semi-transparent letters, with "Universal Serial Bus" in smaller text to the left. Below the header is a navigation menu with links for "About USB-IF", "Channel", "Press", "Developers", "Members", and "Products".

On the left side, there is a vertical stack of USB logos: "SUPERSPEED USB", "WIRELESS USB", "CERTIFIED USB", "HI-SPEED CERTIFIED USB", "CERTIFIED USB On-The-Go", and "ExpressCard".

The central content area features a promotional box for "USB Community at Computex June 2-6, 2015 | Taipei, Taiwan". The box includes the "SUPERSPEED+ CERTIFIED USB 10Gbps" logo and the text: "Come see the latest in USB technology! Nangang Exhibition Hall Booth M0820" and the "COMPUTEX TAIPEI" logo.

To the right of this box is a section titled "Instant, No Hassle Connections" with the following text: "Universal Serial Bus (USB) connects more than computers and peripherals. It has the power to connect you with a whole new world of PC experiences. USB is your instant connection to the fun of digital photography or the limitless creative possibilities of digital imaging. You can use USB to connect with other people through the power of PC-telephony and video conferencing. Once you've tried USB, we think you'll grow quite attached to it!"

At the bottom, there is an "Events" section with a bullet point: "• Computex Jun 02 - 06, 2015 Taipei, Taiwan" followed by the text "Additional information is available".



# Przykład deskryptorów urządzenia HID

## Wykorzystaj USB

## Implementacja klasy HID

## interfejsu USB w STM32

Radosław Kwiecień

ELEKTRONIKA PRAKTYCZNA 3/2009

Zestaw ZL27ARM:

Kontroler STM32F103

- ❖ Sterowanie 8 diodami,
- ❖ Odczyt napięcia z potencjometru,
- ❖ Odczyt stanu dwóch przycisków

# Przykład deskryptorów urządzenia HID

## Deskryptor urządzenia:

List. 1. Kod źródłowy deskryptora urządzenia

```
/* USB Standard Device Descriptor */
const u8 CustomHID_DeviceDescriptor[CUSTOMHID_SIZ_DEVICE_DESC] =
{
    0x12,          //bLength
    USB_DEVICE_DESCRIPTOR_TYPE, //bDescriptorType
    0x00,          //bcdUSB
    0x02,
    0x00,          //bDeviceClass
    0x00,          //bDeviceSubClass
    0x00,          //bDeviceProtocol
    0x40,          //bMaxPacketSize = 40
    0x83,          //VendorID = 0x0483
    0x04,
    0x50,          //ProductID = 0x5750
    0x57,
    0x00,          //bcdDevice rel. 2.00
    0x02,
    1,             //Index of string descriptor describing manufacturer
    2,             //Index of string descriptor describing product
    3,             //Index of string descriptor describing the device serial number
    0x01          //bNumConfigurations
}
; /* CustomHID_DeviceDescriptor */
```

# Przykład deskryptorów urządzenia HID

## Deskryptory raportów:

List. 2. Kod deskryptora raportu LED

```
/* Led 1 */
0x85, 0x01,          //REPORT_ID (1)
0x09, 0x01,          //USAGE (LED 1)
0x15, 0x00,          //LOGICAL_MINIMUM (0)
0x25, 0x01,          //LOGICAL_MAXIMUM (1)
0x75, 0x08,          //REPORT_SIZE (8)
0x95, 0x01,          //REPORT_COUNT (1)
0xB1, 0x82,          //FEATURE (Data,Var,Abs,Vol)
0x85, 0x01,          //REPORT_ID (1)
0x09, 0x01,          //USAGE (LED 1)
0x91, 0x82,          //OUTPUT (Data,Var,Abs,Vol)
```

List. 3. Kod deskryptora raportu potencjometru

```
/* Potencjometr P1 */
0x85, 0x09,          //REPORT_ID (9)
0x09, 0x09,          //USAGE (P1)
0x15, 0x00,          //LOGICAL_MINIMUM (0)
0x26, 0xff, 0x00,    //LOGICAL_MAXIMUM (255)
0x75, 0x08,          //REPORT_SIZE (8)
0x81, 0x82,          //INPUT (Data,Var,Abs,Vol)
0x85, 0x09,          //REPORT_ID (9)
0x09, 0x09,          //USAGE (P1)
0xb1, 0x82,          //FEATURE (Data,Var,Abs,Vol)
```



# Przykład deskryptorów urządzenia HID

## Procedura obsługi diod LED:

List. 5. Kod procedury EP1\_OUT\_Callback

```
void EP1_OUT_Callback(void)
{
    BitAction Led_State;
    PMAToUserBufferCopy(Receive_Buffer, ENDP1_RXADDR, 2);
    if (Receive_Buffer[1] == 0)
    {
        Led_State = Bit_RESET;
    }
    else
    {
        Led_State = Bit_SET;
    }
    switch (Receive_Buffer[0]) // jakie ID raportu ?
    {
        case 1: /* Led 1 */
            GPIO_WriteBit(GPIOB, GPIO_Pin_8, Led_State);
            break;

        case 8: /* Led 8 */
            GPIO_WriteBit(GPIOB, GPIO_Pin_15, Led_State);
            break;
        default:
            GPIO_Write(GPIOB, ~(GPIO_Pin_8|GPIO_Pin_9|GPIO_Pin_10|GPIO_Pin_11|GPIO_
Pin_12|GPIO_Pin_13|GPIO_Pin_14|GPIO_Pin_15));
            break;
    }
    SetEPRxStatus(ENDP1, EP_RX_VALID);
}
```



**Koniec**