**Listing 1. Deskryptor raportu myszy**

char ReportDescriptor**[**52**]** **=** **{**

0x05**,** 0x01**,** //USAGE\_PAGE (Generic Desktop)

0x09**,** 0x02**,** //USAGE (Mouse)

0xa1**,** 0x01**,** //COLLECTION (Application)

0x09**,** 0x01**,** // USAGE (Pointer)

0xa1**,** 0x00**,** // COLLECTION (Physical)

0x05**,** 0x09**,** // USAGE\_PAGE (Button)

0x19**,** 0x01**,** // USAGE\_MINIMUM (Button 1)

0x29**,** 0x03**,** // USAGE\_MAXIMUM (Button 3)

0x15**,** 0x00**,** // LOGICAL\_MINIMUM (0)

0x25**,** 0x01**,** // LOGICAL\_MAXIMUM (1)

0x95**,** 0x03**,** // REPORT\_COUNT (3)

0x75**,** 0x01**,** // REPORT\_SIZE (1)

0x81**,** 0x02**,** // INPUT (Data,Var,Abs)

0x95**,** 0x01**,** // REPORT\_COUNT (1)

0x75**,** 0x05**,** // REPORT\_SIZE (5)

0x81**,** 0x01**,** // INPUT (Cnst,Ary,Abs)

0x05**,** 0x01**,** // USAGE\_PAGE (Generic Desktop)

0x09**,** 0x30**,** // USAGE (X)

0x09**,** 0x31**,** // USAGE (Y)

0x09**,** 0x38**,** // USAGE (Wheel)

0x15**,** 0x81**,** // LOGICAL\_MINIMUM (-127)

0x25**,** 0x7f**,** // LOGICAL\_MAXIMUM (127)

0x95**,** 0x03**,** // REPORT\_COUNT (3)

0x75**,** 0x08**,** // REPORT\_SIZE (8)

0x81**,** 0x06**,** // INPUT (Data,Var,Rel)

0xc0**,** // END\_COLLECTION

0xc0 //END\_COLLECTION

**};**

**Listing 2. Deskryptor raportu klawiatury**

char ReportDescriptor**[**61**]** **=** **{**

0x05**,** 0x01**,** //USAGE\_PAGE (Generic Desktop)

0x09**,** 0x06**,** //USAGE (Keyboard)

0xa1**,** 0x01**,** //COLLECTION (Application)

0x05**,** 0x07**,** // USAGE\_PAGE (Keyboard)

0x19**,** 0xe0**,** // USAGE\_MINIMUM (Keyboard LeftControl)

0x29**,** 0xe7**,** // USAGE\_MAXIMUM (Keyboard Right GUI)

0x15**,** 0x00**,** // LOGICAL\_MINIMUM (0)

0x25**,** 0x01**,** // LOGICAL\_MAXIMUM (1)

0x95**,** 0x08**,** // REPORT\_COUNT (8)

0x75**,** 0x01**,** // REPORT\_SIZE (1)

0x81**,** 0x02**,** // INPUT (Data,Var,Abs)

0x95**,** 0x01**,** // REPORT\_COUNT (1)

0x75**,** 0x08**,** // REPORT\_SIZE (8)

0x81**,** 0x01**,** // INPUT (Cnst,Ary,Abs)

0x19**,** 0x00**,** // USAGE\_MINIMUM (Reserved (no event indicated))

0x29**,** 0x65**,** // USAGE\_MAXIMUM (Keyboard Application)

0x15**,** 0x00**,** // LOGICAL\_MINIMUM (0)

0x25**,** 0x65**,** // LOGICAL\_MAXIMUM (101)

0x95**,** 0x06**,** // REPORT\_COUNT (6)

0x75**,** 0x08**,** // REPORT\_SIZE (8)

0x81**,** 0x00**,** // INPUT (Data,Ary,Abs)

0x05**,** 0x08**,** // USAGE\_PAGE (LEDs)

0x19**,** 0x01**,** // USAGE\_MINIMUM (Num Lock)

0x29**,** 0x05**,** // USAGE\_MAXIMUM (Kana)

0x95**,** 0x05**,** // REPORT\_COUNT (5)

0x75**,** 0x01**,** // REPORT\_SIZE (1)

0x91**,** 0x02**,** // OUTPUT (Data,Var,Abs)

0x95**,** 0x01**,** // REPORT\_COUNT (1)

0x75**,** 0x03**,** // REPORT\_SIZE (3)

0x91**,** 0x01**,** // OUTPUT (Cnst,Ary,Abs)

0xc0 //END\_COLLECTION

**};**

**Listing 3. Zmiany wprowadzone w pliku main.c w projekcie myszy**

/\* USER CODE BEGIN 0 \*/

#include "stm32l4yy.h"

#include "stm32nucleo64.h"

#include "usbd\_hid.h"

struct mouse\_report **{**

uint8\_t buttons**;**

int8\_t x**;**

int8\_t y**;**

int8\_t wheel**;**

**};**

void HAL\_SYSTICK\_Callback**(**void**)**

**{**

static uint8\_t tdiv**;**

**if** **(++**tdiv **==** 10**)**

**{**

tdiv **=** 0**;**

static const struct mouse\_report mrep**[**3**]** **=** **{**

**{**0**,** 100**,** 0**,** 0**},**

**{**0**,** 0**,** 100**,** 0**},**

**{**0**,** **-**100**,** **-**100**,** 0**}**

**};**

static uint8\_t phase **=** 0**;**

static uint8\_t khist**;**

**if** **((**khist **=** **(**khist **<<** 1 **|** BTN\_DOWN**)** **&** 3**)** **==** 1**)**

**{**

USBD\_HID\_SendReport**(&**hUsbDeviceFS**,** **(**uint8\_t**\*)&**mrep**[**phase**],** **sizeof(**struct mouse\_report**));**

**if** **(++**phase **==** **sizeof(**mrep**)** **/** **sizeof(**mrep**[**0**]))** phase **=** 0**;**

**}**

**}**

**}**

/\* USER CODE END 0 \*/

**Listing 4. Zmodyfikowany deskryptor urządzenia w pliku usbd\_custom\_hid\_if.c**

\_\_ALIGN\_BEGIN static uint8\_t CUSTOM\_HID\_ReportDesc\_FS**[**USBD\_CUSTOM\_HID\_REPORT\_DESC\_SIZE**]** \_\_ALIGN\_END **=**

**{**

/\* USER CODE BEGIN 0 \*/

0x05**,** 0x01**,** //USAGE\_PAGE (Generic Desktop)

0x09**,** 0x06**,** //USAGE (Keyboard)

0xa1**,** 0x01**,** //COLLECTION (Application)

0x05**,** 0x07**,** //USAGE\_PAGE (Keyboard)

0x19**,** 0xe0**,** //USAGE\_MINIMUM (Keyboard LeftControl)

0x29**,** 0xe7**,** //USAGE\_MAXIMUM (Keyboard Right GUI)

0x15**,** 0x00**,** //LOGICAL\_MINIMUM (0)

0x25**,** 0x01**,** //LOGICAL\_MAXIMUM (1)

0x95**,** 0x08**,** //REPORT\_COUNT (8)

0x75**,** 0x01**,** //REPORT\_SIZE (1)

0x81**,** 0x02**,** //INPUT (Data,Var,Abs)

0x95**,** 0x01**,** //REPORT\_COUNT (1)

0x75**,** 0x08**,** //REPORT\_SIZE (8)

0x81**,** 0x01**,** //INPUT (Cnst,Ary,Abs)

0x19**,** 0x00**,** //USAGE\_MINIMUM (Reserved (no event indicated))

0x29**,** 0x65**,** //USAGE\_MAXIMUM (Keyboard Application)

0x15**,** 0x00**,** //LOGICAL\_MINIMUM (0)

0x25**,** 0x65**,** //LOGICAL\_MAXIMUM (101)

0x95**,** 0x06**,** //REPORT\_COUNT (6)

0x75**,** 0x08**,** //REPORT\_SIZE (8)

0x81**,** 0x00**,** //INPUT (Data,Ary,Abs)

0x05**,** 0x08**,** //USAGE\_PAGE (LEDs)

0x19**,** 0x01**,** //USAGE\_MINIMUM (Num Lock)

0x29**,** 0x05**,** //USAGE\_MAXIMUM (Kana)

0x95**,** 0x05**,** //REPORT\_COUNT (5)

0x75**,** 0x01**,** //REPORT\_SIZE (1)

0x91**,** 0x02**,** //OUTPUT (Data,Var,Abs)

0x95**,** 0x01**,** //REPORT\_COUNT (1)

0x75**,** 0x03**,** //REPORT\_SIZE (3)

0x91**,** 0x01**,** //OUTPUT (Cnst,Ary,Abs)

//0xc0

// END\_COLLECTION

/\* USER CODE END 0 \*/

0xC0 /\* END\_COLLECTION \*/

**};**

**Listing 5. Funkcja CUSTOM\_HID\_OutEvent\_FS interpretująca raport wyjściowy**

static int8\_t CUSTOM\_HID\_OutEvent\_FS **(**uint8\_t event\_idx**,** uint8\_t state**)**

**{**

/\* USER CODE BEGIN 6 \*/

**if** **(**event\_idx **&** 2**)** LED\_PORT**->**BSRR **=** LED\_MSK**;**

**else** LED\_PORT**->**BRR **=** LED\_MSK**;**

**return** 0**;**

/\* USER CODE END 6 \*/

**}**

**Listing 6. Zmiany wprowadzone w pliku usbd\_custom\_hid\_if.c w projekcie klawiatury**

/\* USER CODE BEGIN 0 \*/

#include "stm32l4yy.h"

#include "stm32nucleo64.h"

#define KEY\_MOD\_LCTRL 0x01

#define KEY\_MOD\_LSHIFT 0x02

#define KEY\_MOD\_LALT 0x04

#define KEY\_MOD\_LMETA 0x08

#define KEY\_MOD\_RCTRL 0x10

#define KEY\_MOD\_RSHIFT 0x20

#define KEY\_MOD\_RALT 0x40

#define KEY\_MOD\_RMETA 0x80

#define KEY\_CAPSLOCK 0x39

#define KEY\_LET(v) (v - 'A' + 4) // A key - code 4

struct kbd\_report **{**

uint8\_t modifier**;**

uint8\_t reserved**;**

uint8\_t key**[**6**];**

**};**

void HAL\_SYSTICK\_Callback**(**void**)**

**{**

static const struct kbd\_report kreps**[**5**]** **=** **{**

**{**0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**},**

**{**0**,** 0**,** KEY\_LET**(**'A'**),** 0**,** 0**,** 0**,** 0**,** 0**},**

**{**KEY\_MOD\_LSHIFT**,** 0**,** KEY\_LET**(**'B'**),** 0**,** 0**,** 0**,** 0**,** 0**},**

**{**KEY\_MOD\_RALT**,** 0**,** KEY\_LET**(**'C'**),** 0**,** 0**,** 0**,** 0**,** 0**},**

**{**0**,** 0**,** KEY\_CAPSLOCK**,** 0**,** 0**,** 0**,** 0**,** 0**},**

**};**

static uint8\_t phase **=** 5**;**

static uint8\_t khist**;**

static uint8\_t tdiv**;**

**++** tdiv**;**

**if** **(**tdiv **%** 10 **==** 0

**&&** phase **==** 5

**&&** **(**khist **=** **(**khist **<<** 1 **|** BTN\_DOWN**)** **&** 3**)** **==** 1**))**

**{**

phase **=** 0**;**

**}**

**if** **(**tdiv **==** 100**)**

**{**

tdiv **=** 0**;**

USBD\_CUSTOM\_HID\_SendReport\_FS**((**uint8\_t**\*)&**kreps**[**phase**],** **sizeof(**struct kbd\_report**));**

**if** **(**phase **<** 5**)** **++**phase**;**

**}**

**}**

**}**

/\* USER CODE END 0 \*/