

Metody připojování periferií

BI-MPP

Přednáška 7

Ing. Miroslav Skrbek, Ph.D.

Katedra číslicového návrhu

Fakulta informačních technologií

České vysoké učení technické v Praze

Miroslav Skrbek ©2010-2021

ZS2021/22



EVROPSKÁ
UNIE

Evropský sociální fond

Praha & EU: Investujeme do vaší budoucnosti

Agenda

- Realizace USB zařízení
- USB podsystém v PIC24FJ256GB106
- Příklady přenosů dat

Literatura

- Gook, M.: Hardwarová rozhraní – Průvodce programátora. Computer Press, Brno 2006. ISBN 80-251-1019-2
- Universal Serial Bus Specification 3.0, Revision 1.0, Listopad 2008
http://www.usb.org/developers/docs/usb_30_spec_092911.zip
- PIC24F Family Reference Manual, section 27, USB On-The-Go (OTG). Technická dokumentace Microchip Technology Incorporated. 2010
(<http://www.microchip.com>)
- PIC24FJ256GB110 Family Data Sheet. Technická dokumentace Microchip Technology Incorporated. 2009.
(<http://www.microchip.com>)

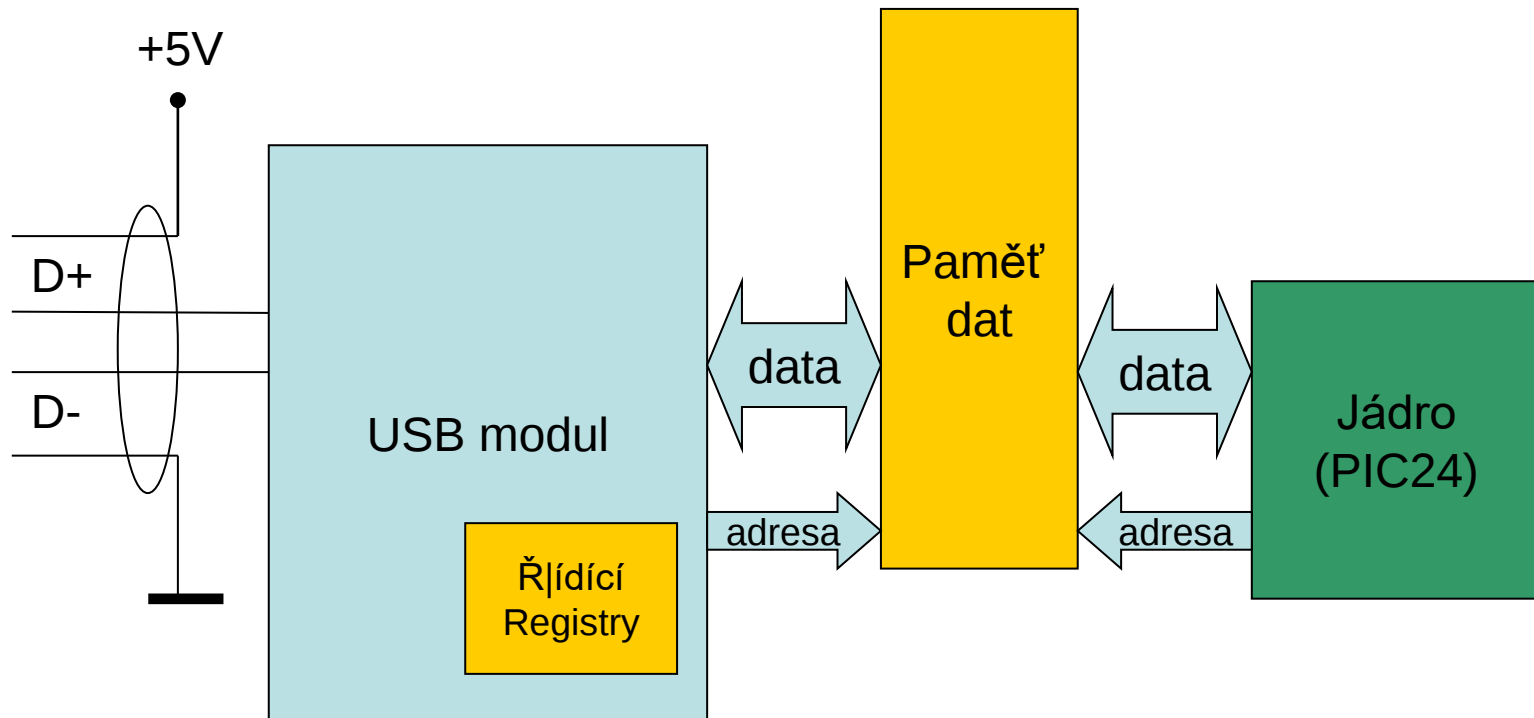
Realizace USB zařízení

- USB řadiče a mikrokontroléry podporují
 - USB zařízení
 - USB host (On-the-Go)
- USB rozhraní a jeho programová obsluha závisí na konkrétním výrobci řadiče nebo mikrokontroléru
- Výrobci poskytují vzorové programové vybavení pro USB, které návrháři upravují pro potřeby vyvíjeného zařízení

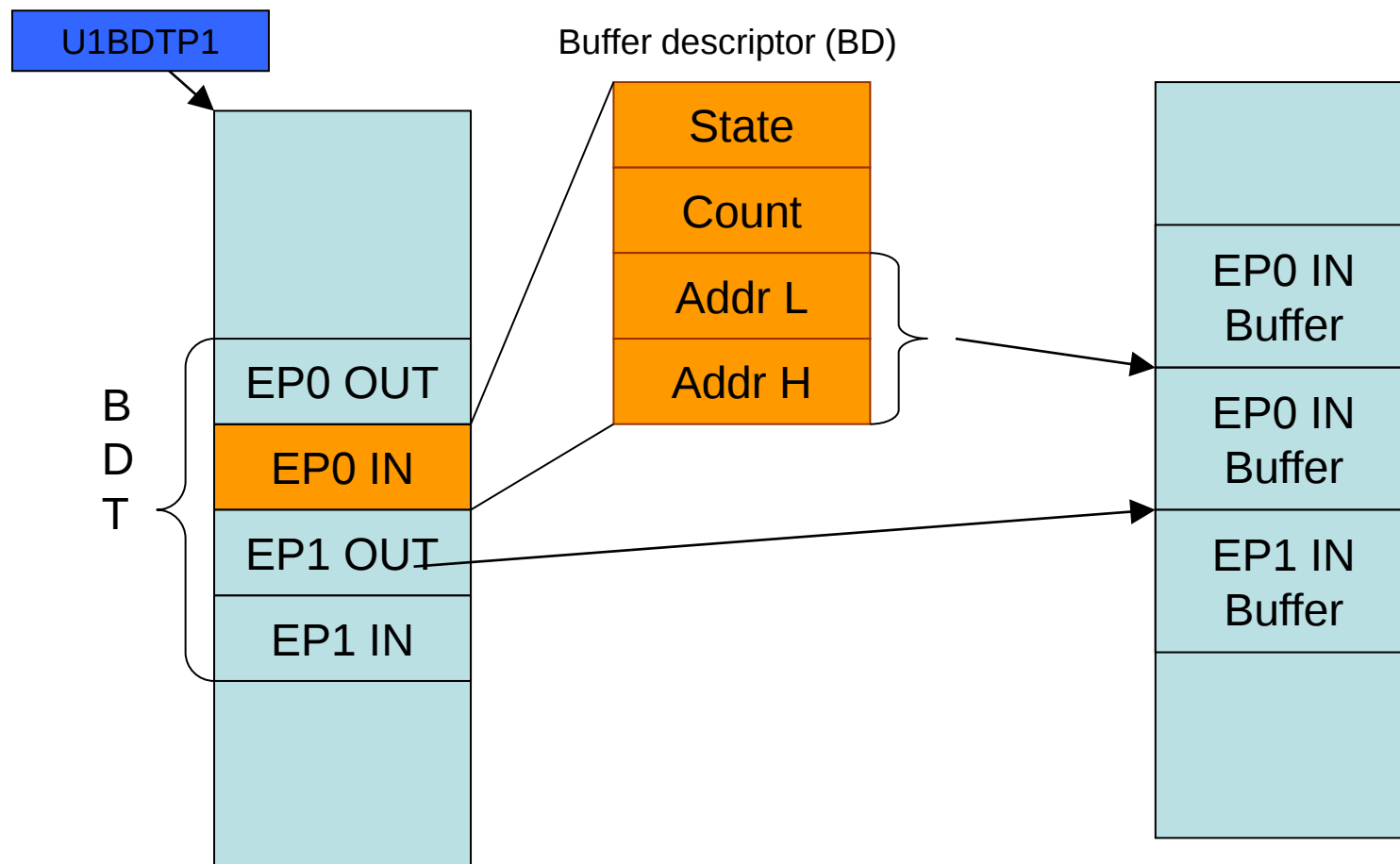
USB v PIC24FJ256GB106

- Hardware zajišťuje serializaci a deserializaci dat, detekci a řízení stavu linky (D+, D-)
- Data jsou ukládána/čtena do/z bufferů, které jsou definovány pro každý koncový bod zvlášť
- Dokončení přenosů dat, příjem speciálních paketů (např. SOF) a změny stavu linky vyvolávají přerušení
- Konfigurace USB řadiče se provádí přes registry řadiče

USB modul v PIC24FJ256GB106



Popisovače v USB modulu



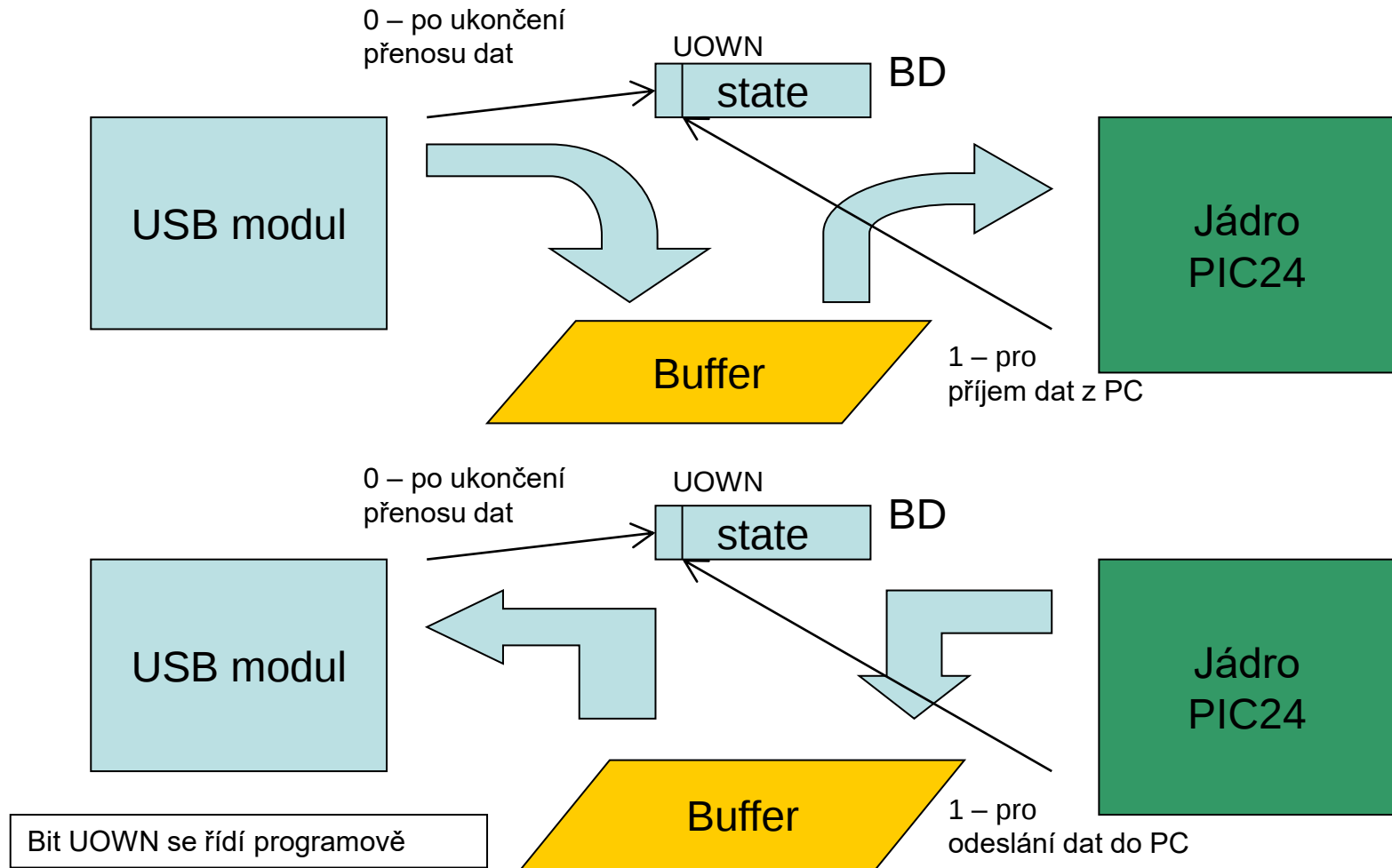
BDT – buffer descriptor table (tabulka popisovačů bufferů)

BD – buffer descriptor (popisovač bufferu)

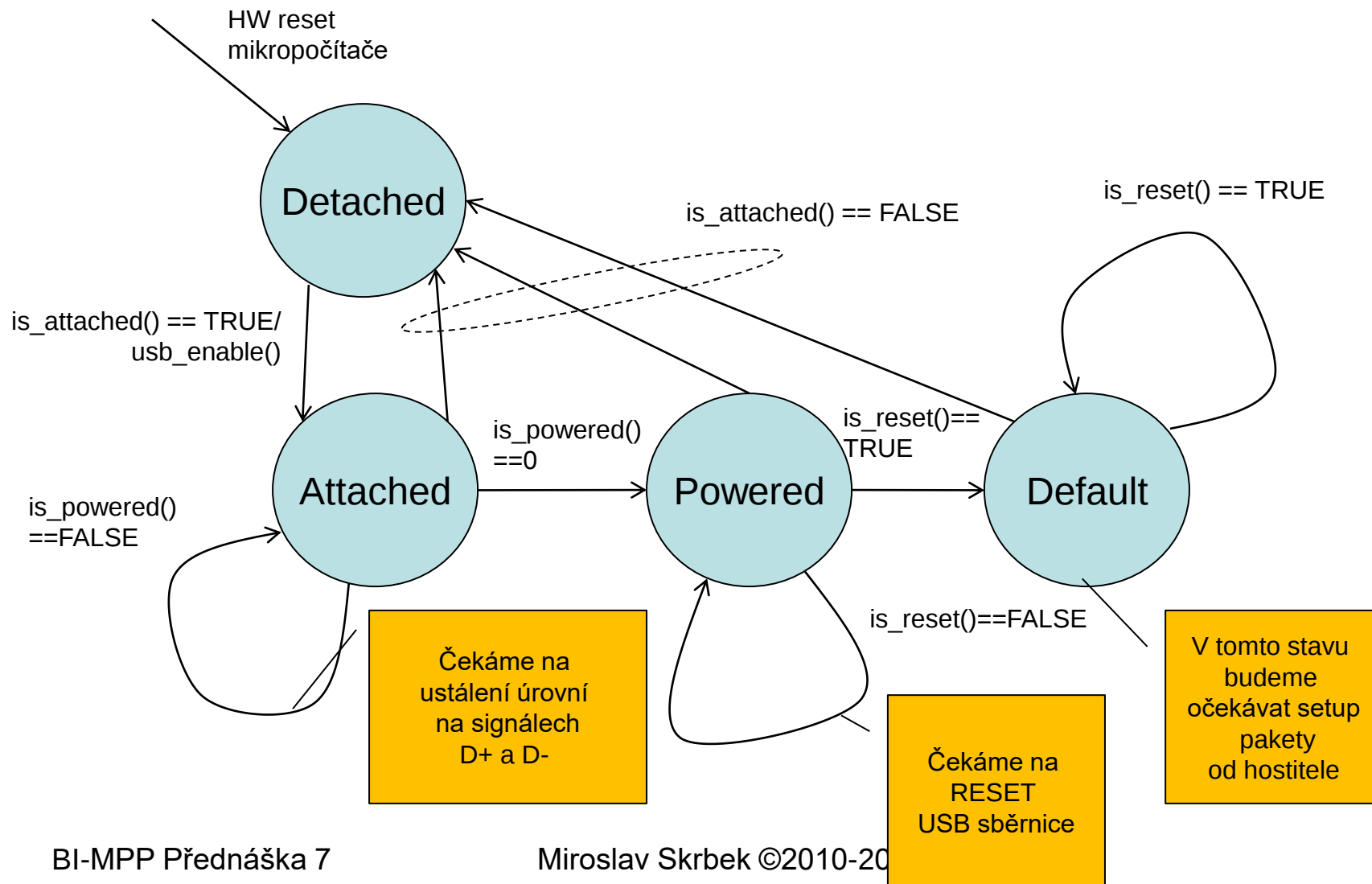
BDT (Buffer Descriptor Table)

- Tabulka obsahuje pro každý koncový bod a směr popisovač bufferu
- Popisovač bufferu obsahuje
 - Stav bufferu (state)
 - Počet přijatých/k odeslání bytů (count)
 - Adresa bufferu v paměti (Addr L, Addr H)
- Bit 7 stavu bufferu se synchronizuje přístup do bufferu ze strany USB řadiče a CPU

Řízení přístupu do bufferu



Stavy zařízení (zjednodušeno)



Další stavy zařízení

- Addressed
 - Do tohoto stavu se přechází ze stavu Default, když zařízení obdrží žádost o nastavení adresy zařízení a nastaví přijatou adresu v USB subsystému (`usb_set_address(...)`)
- Configured
 - Do tohoto stavu zařízení přechází ze stavu addressed po přijetí žádosti `SET_CONFIGURATION`

Knihovna USBLIB

- Knihovna základních funkcí pro ovládání USB modulu v mikropočítačích PIC24
- Poskytuje funkce pro
 - Vytvoření tabulky popisovačů bufferů (BDT)
 - Funkce pro detekci stavu USB modulu
 - Funkce pro zahájení přenosu dat z/do PC
 - Funkce pro kopírování dat z/do bufferů
- Knihovna byla vyvinuta pro potřeby výuky na FIT a FEL ČVUT v Praze, včetně portování pro mikropočítače řady PIC24F

Deklarace koncových bodů a vytvoření BDT

Deklarace BDT

```
DECLARE_EP_BEGIN
    DECLARE_EP(ep0);    // IN/OUT Control EP
    DECLARE_EP(ep1);    // IN Interrupt EP
    DECLARE_EP(ep2);    // OUT data EP
    DECLARE_EP(ep3);    // IN data EP
DECLARE_EP_END
```

Deklarace bufferů

```
DECLARE_BUFFER_BEGIN
    DECLARE_BUFFER(ep0_buf_out, EP0_OUT_BUF_SIZE);
    DECLARE_BUFFER(ep0_buf_in, EP0_IN_BUF_SIZE);
    DECLARE_BUFFER(ep1_buf_in, EP1_IN_BUF_SIZE);
    DECLARE_BUFFER(ep2_buf_out, EP2_OUT_BUF_SIZE);
    DECLARE_BUFFER(ep3_buf_in, EP3_IN_BUF_SIZE);
DECLARE_BUFFER_END
```

Funkce pro zjištění stavu USB modulu

- `int is_attached();`
vrací 1, pokud je zařízení připojeno k USB
- `int is_powered();`
vrací 1, pokud je USB datová linka ve stavu SE0
- `int is_reset();`
vrací 1, pokud byl detekován reset na USB sběrnici. Volání funkce příznak nuluje.
- `int is_idle();`
detekuje požadavek přechodu do stavu snížené spotřeby
- `int is_sof();`
detekuje příchod SOF paketu
- `int is_transfer_done();`
vrací 1, pokud byl dokončen přenos. Volání funkce příznak nenuluje.
- `int get_trn_status();`
vrací číslo koncového bodu, kde došlo k dokonšení přenosu. Tato funkce nuluje příznak dokončeného přenosu.

Funkce pro práci s endpointy a buffery

- `extern void usb_init_ep(int num, byte ep_type, ep_buf_desc_t* ep);`
inicializuje koncový bod
- `void usb_ep_transf_start(volatile ep_buf_desc_t* ep, byte transf_type, buf_ptr_t buffer, int size);`
zahájí přenos dat z/do PC
- `void copy_to_buffer(buf_ptr_t buf, byte* src, int size);`
kopíruje data z paměti do bufferu
- `void copy_from_buffer(buf_ptr_t buf, byte* dest, int size);`
kopíruje data z bufferu do paměti

Přenos do PC - příklad

Následující příklad ukazuje přenos odezvy na žádost GET_DESCRIPTOR – Device Descriptor do zařízení

```
// Inicializace koncového bodu
usb_init_ep(0, EP_SETUP_INOUT, &__bdt.ep0);

copy_to_buffer(ep0_buf_in, &device_descriptor,
    sizeof(device_descriptor));

// Zahájení přenosu dat do PC (očekává se SETUP paket)
usb_ep_transf_start(EP(ep0), USB_TRN_DATA1_IN, ep0_buf_in,
    sizeof(device_descriptor));

// čekej dokud není přenos dokončen
while (!is_transfer_done());

// zjištění koncového bodu, kde došlo k dokončení přenosu
int epnum = get_trn_status();
if (epnum == 0x00) {
    // naplánuj příjem potvrzení od PC
}
```

Přenos z PC - příklad

Následující příklad ukazuje přenos požadavku (USB DEVICE REQUEST) do zařízení

```
// Inicializace koncového bodu
usb_init_ep(0, EP_SETUP_INOUT, &__bdt.ep0);

// Zahájení přenosu dat z PC (očekává se SETUP packet)
usb_ep_transf_start(&__bdt.ep0, USB_TRN_SETUP, ep0_buf_out,
                    EP0_OUT_BUF_SIZE);

// čekej dokud není přenos dokončen
while (!is_transfer_done());

// zjištění koncového bodu, kde došlo k dokončení přenosu
int epnum = get_trn_status();
if (epnum == 0x00) {
    // data přijata na EP0, zkopírujeme obsah bufferu
    // do proměnné req (usb device request)
    copy_from_buffer(ep0_buf_out, (byte*)req,
                     sizeof(usb_device_req_t));
}
```