

Komunikácia prostredníctvom USB na báze modulu DLP-USB245M

Cvičenie č. 9

Úvod k problematike

Komunikáciu mikroradiča MSP430F169 na vývojovej doske MSP430F169EVB prostredníctvom sériovej zbernice USB zabezpečuje modul DLP-USB245M firmy DLP Design. Ide o zásobník FIFO s paralelným 8-bitovým prístupom.

Podrobný opis vlastností modulu je možné nájsť v priloženej dokumentácii `dlp-usb245mv15.pdf`.

Pomocou signálu WR (aktívna úroveň HI) je možné zapisovať jeden byte do zásobníka. To je možné robiť, pokiaľ je príznakový signál TXE# = LO. V prípade, že TXE# = HI je 384 úrovňový zásobník plný a treba počkať, pokiaľ modul neodošle aspoň časť dát a TXE neklesne na LO. Analogicky funguje čítanie, kde RXF# signalizuje úrovňou LO prítomnosť aspoň jedného bytu v 128 bytovom zásobníku prijímača. Na čítanie dát používame signál RD#. Po \downarrow RD# uloží modul na zbernicu (dátové vývody portu P1) hodnotu zo zásobníka. Po \uparrow RD# sa pripraví ďalšie dáta ak sú v zásobníku na prečítanie. K modulu sú dodávané aj ovládače pre obsluhu zariadenia pomocou PC. Pre jeho použitie nie je teda nutné podrobne poznať protokol rozhrania USB, keďže modul zabezpečuje všetku komunikáciu a prenos dát. V ideálnom prípade je možné dosiahnuť prenosovú rýchlosť až 1MB/s. Keďže signály RXF# a TXE# sú pripojené na port P2 je možné ich využiť ako zdroje prerušenia a teda môže prebiehať plne automatická komunikácia, bez nutnosti programovo kontrolovať stav signálu RXF#. Na PC sa modul bude javiť ako virtuálne COM zariadenie. Ovládače podporujú väčšinu používaných operačných systémov.

1. V prostredí CCS vytvorte nový projekt s názvom `mr_cv9`. K projektu pripojte dodané súbory:

- `oscillators_config.h`
- `board_config.h`
- `delay.h`
- `usb.h`
- `main.c`
- `oscillators_config.c`
- `board_config.c`
- `delay.c`
- `usb.c`

2. V súbore `main.c` vykonajte nasledujúce úpravy:

- pomocou funkcie `Set_system_clock()` nastavte oscilátor XT2CLK ako zdroj pre MCLK aj SMCLK, pričom pre oba zdroje bude signál delený ôsmymi. Signál ACLK nebude aktívny.
- funkciou `Board_shutdown()` uveďte vývojovú dosku do definovaného stavu
- funkciou `usb_init()` nastavte definovaný stav pre porty a signály, ktoré bude používať USB modul
- funkciou `__bis_SR_register(GIE)` globálne povoľte prerušenia
- v nekonečnom cykle `while(1)` posielať prostredníctvom USB čísla od 0 do 255 v 100ms intervaloch a zároveň kontrolujte, či nebol prijatý byte z PC. Ak bol prijatý byte, zobrazte ho na LED pripojených na porte P4.
- v súbore `main.c` sa musí nachádzať aj obsluha prerušenia od portu P2 (signál RXF#). V rámci tejto obsluhy program prijme byte dát a uloží ho do globálnej premennej `received_usb_data`, zároveň nastaví príznak `received_data_flag`, aby funkcia v nekonečnom cykle mohla vyhodnotiť prijatie bytu a rozsvietiť ho na LED

3. V súbore `usb.c` napíšte telo dvoch funkcií:
 - funkcia `usb_init()` zabezpečí, aby všetky vývody portu P1 boli nastavené ako vstupy, nastaví počiatočný stav signálov RD a WR modulu DLP (viď komunikačný protokol pre čítanie a zápis bytu v dokumentácii `dlp-usb245mv15.pdf`), a nastaví port P2: vývody P2.0 (RD#) a P2.1 (WR) budú výstupné, ostatné vývody portu P2 budú vstupné. Zároveň je potrebné povoliť prerušenie pri prechode signálu RXF# (vývod P2.3) z vysokej na nízku úroveň.
 - funkcia `usb_write()` zabezpečí vyslanie jedného bytu v zmysle protokolu uvedeného v dokumentácii modulu DLP
4. Program skompilujte a odladte na vývojovej doske. Pre testovanie USB komunikácie s vývojovou doskou použite dodaný program `dlptest.exe`.
5. Bonusová úloha v rámci miniprojektu: odosielajte dáta zadávané z klávesnice a prijaté dáta zobrazujte na LCD displeji. Za úspešné vyriešenie bonusovej úlohy získava študent 3 body, ktorými môže nahradiť prípadnú stratu pri zápočtových testoch. Vyriešenú úlohu je potrebné odovzdať do ďalšieho cvičenia.