

9. Magistrala USB (Universal Serial Bus)



Cuprins

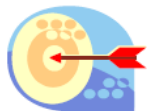
Cuprins modul

- 9.1. Descriere și caracteristici
- 9.2. Arhitectura magistralei
- 9.3. Nivelul fizic
- 9.4. Transferul de date prin cadre
- 9.5. Cuplarea unui microcontroller (MC) la USB printr-o interfață specializată
- 9.6. Microcontrollere cu USB integrat



Introducere

Cursul “**Magistrala USB (Universal Serial Bus)**“ prezintă la început structura generală, avantajele și performanțele acestei magistrale. Nivelul fizic este prezentat pe scurt, după care se prezintă transferul de date prin cadre mai detaliat, cu un exemplu de transfer. Ca și informații exemplificative cu caracter practic sunt prezentate modurile în care se poate construi un dispozitiv electronic cu interfață USB utilizând un circuit specializat sau un microcontroller cu interfață USB integrată.



Obiective

După parcurgerea acestui curs studenții vor înțelege conceptele generale și avantajele USB și vor putea să:

- Vadă diferența între modul asincron de transfer de date și cel sincron;
- Aprecieze diferențele între transferul de date prin cadre și cel gestionat cu semnale auxiliare;
- Aleagă varianta optimă de conectare prin USB în condițiile unei aplicații cu microcontroller.

Obiective specifice:

1. Învățarea unor tipuri de codare digitală. Înțelegerea caracteristicilor cerute codării;
2. Cunoașterea unor tipuri de transmisii seriale și interfețe seriale ca structură, protocol și interfețe tipice;
3. Învățarea unor interfețe tipice interne din microcontrollere.



Durata medie de studiu individual

Durata medie de studiu individual este de 2 ore.

9.1.Descriere și caracteristici

În prezent conectarea pe USB este cea mai răspândită variantă de conectare a dispozitivelor periferice la calculatoare PC. Printre perifericele cu USB se pot menționa imprimantele, stick-urile de memorie flash, aparatele foto, telefoanele mobile dar și tensiometre, jucării sau fierbătoare de cafea, figura 9.1.



Figura 9.1. Exemple de dispozitive USB

Magistrala USB este bazată pe o transmisie serială, sincronă, entitatea este cadrul (blocul) de date, codificare de grup cu adăugare de biți, cu refacerea tactului din datele citite, verificarea corectitudinii transferului cu CRC.

Magistrala USB a fost introdusă cu dorința de a oferi utilizatorilor o interfață universală, cu viteză mare și ușor de folosit, mai ieftină pentru că, fiind serială, cablurile și conectorii costă mai puțin. Aceste considerente au impus magistrala USB pe piață, în prezent aceasta ocupând o cotă mare de piață în domeniul interfațării. Complexitatea USB este mai mare decât a interfețelor înlocuite, adică a interfeței RS232 și a interfeței paralele, prin urmare implementarea ei în microcontrollere a fost mai dificilă. În 2008 au fost vândute peste 3 miliarde de dispozitive USB, iar intrarea pe piață a USB 3.0 (SuperSpeed USB) în anul 2009, cu o estimare de vânzări până în 2015 de 25% din totalul dispozitivelor USB asigură condițiile supraviețuirii îndelungate. Caracteristicile principale ale magistralei USB:

- rata de transfer este de 1,5 Mbps la USB 1.0 (Low Speed), 12Mbps la USB 1.1 (Full Speed), 480Mbps la USB 2.0 (Hi Speed) și 4,8Gbps la USB 3.0 (Super Speed);

- conectează până la 127 de dispozitive la un calculator gazdă, într-o topologie de tip stea multiplă. Nu se pot conecta dispozitive USB fără gazdă (ceea ce este posibil la interfața IEEE 1394);
- configurarea este automată, adică se poate conecta un dispozitiv USB fizic în mers (Hot Plug In). Se remarcă creșterea complexității software față de partea hardware;
- cablul conține linii de alimentare, așa că dispozitivele USB pot fi alimentate de la gazdă (bus powered device) sau pot avea alimentare proprie (self powered device). Din acest motiv cablurile au conectori diferiți pentru conectarea spre gazdă (upstream) și spre dispozitiv (downstream);
- distanța de conectare este de maximum 5m, distanța se poate mări prin inserarea de hub-uri.

Specificațiile acestei magistrale descriu atributele de magistrală, definesc protocolul, tipurile de tranzacții, administrarea magistralei (bus management) și totodată furnizează informații necesare pentru construirea unui sistem în acest standard.

Magistrala USB definește trei categorii de dispozitive fizice:

- gazda USB (USB Host)
- funcții USB (USB function)
- distribuitoare USB (USB Hub)

9.2.Arhitectura magistralei

Dispozitivele USB sunt conectate într-o topologie de tip stea multiplă. Topologia USB este reprezentată în figura 9.2. În nodul fiecărei stele se găsește un hub.

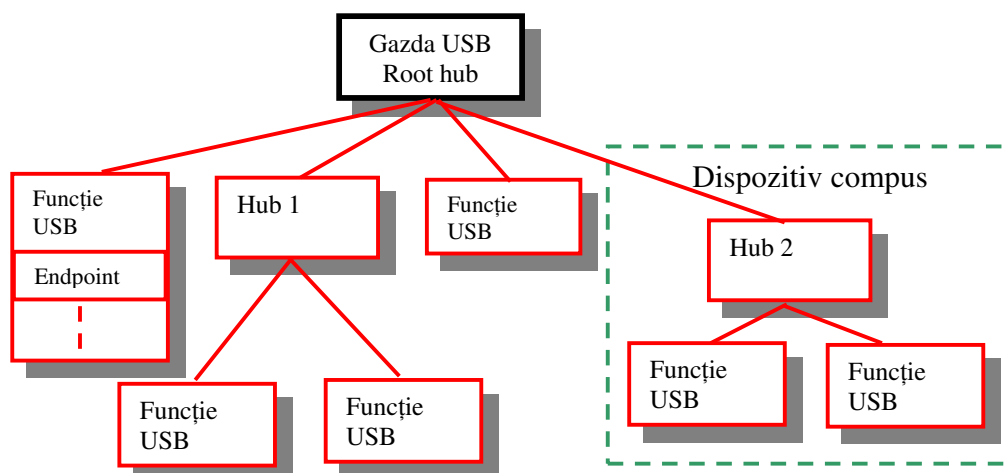


Figura 9.2. Arhitectura USB

Legătura este multipunct pe magistrală dar punct la punct între hub și dispozitive. Este posibil ca un dispozitiv fizic să conțină mai multe funcții și un hub, acest dispozitiv numindu-se compus. Un exemplu este o multifuncțională care conține imprimantă, scanner și fax, toate acestea fiind funcții USB.

Fiecare dispozitiv USB poate dispune de una sau mai multe endpoint-uri prin care gazda comunică cu dispozitivul. Un endpoint este un registru intern, adresabil de gazdă în care se pot trimite sau din care se pot citi informații specifice. Toate dispozitivele posedă un endpoint special, *endpoint zero*, care este privit ca pipe de control. Pipe-ului endpoint zero îi este asociată informația ce descrie complet dispozitivul USB: clasa de dispozitiv, informații de power management, producător etc.

Inițiatorul transferurilor de date pe magistrală este gazda USB. Protocolul folosit este protocol prin interogare (de tip polled). Datele vehiculate pe magistrală sunt grupate în pachete iar o tranzacție de magistrală implică transmiterea a cel mult trei pachete. Fiecare tranzacție începe prin trimiterea de către gazdă a unui pachet de semnalizare - *token packet*- care descrie tipul și sensul tranzacției, adresa dispozitivului USB și numărul nodului destinație (endpoint). Dispozitivul adresat se selectează prin decodificarea adresei ce-i corespunde. Urmează transferul de date de la gazdă spre dispozitivul adresat sau invers, după cum este specificat în pachetul de semnalizare. Receptorul răspunde în această tranzacție printr-un pachet de dialog -*handshake packet*- prin care se confirmă (sau nu) încheierea cu succes a transferului de date.

9.3.Nivelul fizic

Aspectele electrice și mecanice ale interfeței sunt reglementate foarte precis în specificațiile de magistrală. Semnalele electrice sunt transmise diferențial (D+ și D-). Codificarea utilizată este NRZI cu împănare de biți (*bit-stuffing*) și tactul de recepție este generat din datele transmise, codul fiind autosincronizabil.

Cablul USB are patru fire, semnalul util este transportat pe două conductoare torsadate iar pe celelalte două conductoare cablul transportă tensiunea de alimentare nominală de +5V (V_{BUS}) și potențialul de referință (GND), figura 9.3.

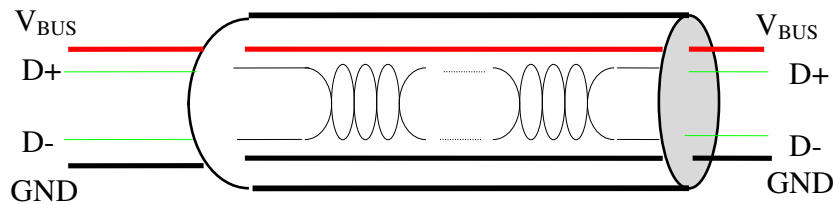


Figura 9.3. cablu USB

Tensiunea transmisă pe linie nu este tensiunea de alimentare a calculatorului gazdă ci este gestionată de controllerul USB, așa încât o suprasarcină este detectată și un mesaj de eroare este afișat de sistemul de operare. Câteva variante de cabluri USB sunt date în figura 9.4.



Figura 9.4. Cabluri USB, de la stânga spre dreapta: prelungitor, downstream-upstream, mini USB, USB 3.0

Ușurința cu care este utilizată USB rezultă din atributul special de tip plug-and-play al acestei magistrale. USB acceptă cuplarea și decuplarea de dispozitive în orice moment; sistemul software se adaptează dinamic la modificările fizice de topologie. Un dispozitiv USB este plasat fizic în structură prin atașarea la portul unui hub. Hub-ul dispune de indicatori de stare la fiecare port pentru a semnaliza cuplarea sau decuplarea unui dispozitiv. Gazda sesizează semnalizarea de la hub și atribuie o adresă unică dispozitivului. La decuplare hub-ul dezactivează portul și indică gazdei acest eveniment. Pentru a se adapta dinamic, sistemul software USB este permanent într-un proces de inventariere a magistralei (bus counting).

9.4. Transferul de date prin cadre

Arhitectura USB distinge patru tipuri de bază de transferuri de date:

- transferuri de control (*Control Transfers*) - sunt folosite pentru configurare și comandă și obligatoriu trebuie să fie suportate de toate perifericele;
- transferuri cu volum mare de date (*Bulk Data Transfers*) permit dispozitivelor să schimbe cantități mari de informație cu gazda pe măsură ce magistrala devine disponibilă, (ex.: camere digitale, scannere sau imprimante);
- transferuri prin întreruperi (*Interrupt Data Transfers*) a fost proiectat ca suport pentru periferice de intrare controlate de om, (tastatură, mouse, joystick), care au nevoie să comunice rar, cantități mici de date. Datele transferate în acest mod sunt caractere, coordonate sau semnalizări de evenimente organizate în unul sau mai mulți octeți;
- transferuri izocrone (*Isochronous Transfers*) - asigură un acces garantat la magistrală, flux de date constant și tolerează erorile de transmisie. Datele izocrone sunt continue și în timp real la toate nivelele: generare, emisie, recepție și utilizare la receptor. Acest tip de transfer este folosit pentru fluxuri de transfer în timp real cum ar fi sistemele audio.

USB folosește un protocol bazat pe pachete de date (Data Packet). Un pachet de date este o colecție de cadre de date (Data Frame). Numărul de biți dintr-un cadru nu are o valoare fixă. Biții sunt trimiși spre magistrală astfel: primul bit este cel mai puțin semnificativ bit (LSB) din cadru, urmat de bitul mai semnificativ până la trimiterea celui mai semnificativ (MSB) bit din cadrul respectiv.

Protocolul USB definește patru tipuri de pachete de date:

- pachet de semnalizare (Token Packet)
- pachet de date (Data Packet)
- pachet de dialog (Handshake Packet)
- pachet special (Special Packet)

a. Pachetul de semnalizare (Token Packet)

Orice transfer începe prin trimiterea de către gazdă a unui pachet de semnalizare. Un pachet are 32 de biți împărțiți în cinci câmpuri, figura 9.5.

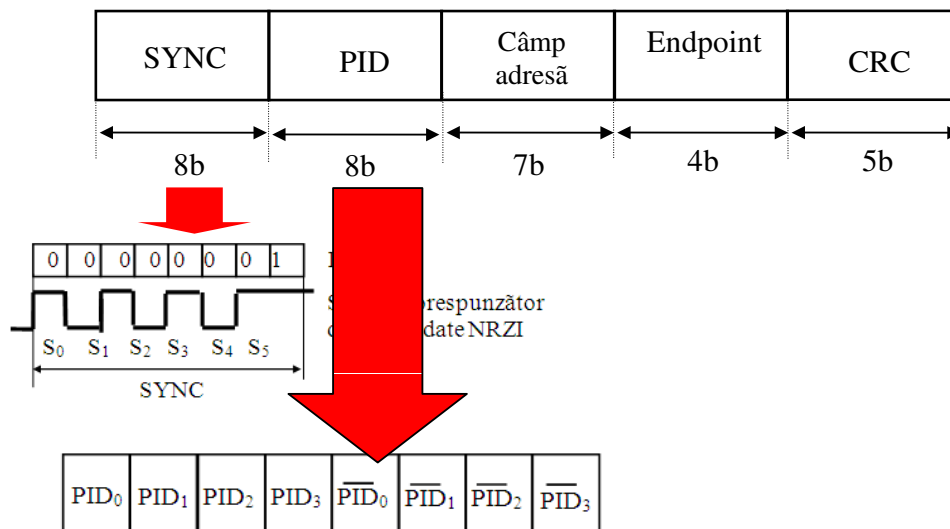


Figura 9.5. Pachetul de semnalizare (sus), câmpul de sincronizare (mijloc) și câmpul PID (jos)

Toate pachetele conțin la începutul lor un câmp de sincronizare, numit SYNC care permite buclei PLL pentru refacerea tactului din receptor să se sincronizeze, și un câmp identificator de pachet, numit PID (Packet Identifier).

SYNC este primul câmp din orice pachet USB. Câmpul de sincronizare este constituit dintr-o serie de biți care produc un șir dens de tranziții utilizând schema de codificare NRZI cerută de standardul USB. Câmpul apare ca o serie de trei tranziții 1/0 urmată de o marcă cu lățimea a două impulsuri. Datele din câmp au succesiunea de valori 0000 0001. Câmpul PID urmează câmpului SYNC într-un pachet USB și are lungimea de 8 biți. Primii patru biți indică tipul pachetului, iar următorii patru sunt în ordine primii patru complementați (complement față de 1) și sunt folosiți ca biți de verificare pentru a confirma acuratețea primilor patru. Câmpul PID definește trei categorii de pachete handshake:

- Pachetul handshake ACK indică emițătorului că pachetul de date a fost recepționat fără erori;
- Pachetul handshake NAK indică faptul că o funcție nu a fost capabilă să recepționeze date de la gazdă (într-o tranzacție OUT) sau că o funcție nu are date de transmis gazdei (într-o tranzacție IN). O gazdă nu poate trimite niciodată NAK;
- Pachetul STALL este emis de o funcție ca răspuns la un pachet de semnalizare IN sau după o tranzacție de date OUT, indicând că funcția nu este capabilă să emită sau să recepționeze date. Gazda nu poate răspunde cu pachet STALL.

b. Pachetul de date și pachetul handshake

Informația propriu-zisă este transferată în sistemele USB sub forma unor pachete de date, figura 9.6. După câmpurile SYNC și PID urmează câmpul de date care este compus dintr-un număr întreg de octeți, de la 0B la 1023B. Corectitudinea câmpului de date este asigurată prin câmpul de verificare ciclică CRC de 16b aflat la sfârșitul pachetului.

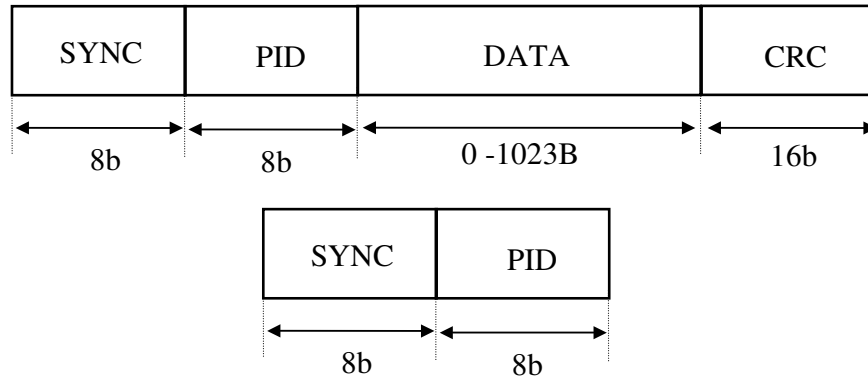


Figura 9.6. Structura pachetelor de date (sus) și handshake (jos)

Pachetele handshake sau de dialog sunt folosite pentru a raporta starea unui transfer de date, pentru a indica recepția cu succes a datelor sau pentru a întoarce valori care indică acceptarea/ respingerea unei comenzi sau o stare de HALT la dispozitiv. Acest tip de pachet este compus doar din două câmpuri; SYNC și PID. Structura este reprezentată în figura 9.6 jos.

Un dialog simplificat este dat în figura 9.7. Cu un pachet de semnalizare se indică adresa dispozitivului și tipul tranzacției care presupunem că este un transfer de date spre gazda USB. Dispozitivul USB răspunde cu pachetul de date. Gazda confirmă primirea cu un pachet handshake.

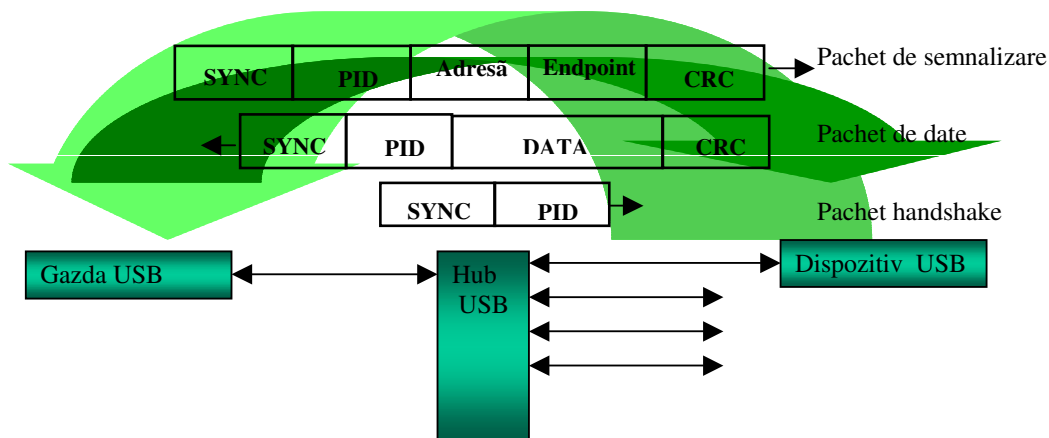


Figura 9.7. Dialog USB simplificat

9.5. Cuplarea unui microcontroller (MC) la USB printr-o interfață specializată

Dacă aplicația necesită cuplarea unui MC la USB atunci există două variante, utilizarea unei interfețe specializate așa cum sunt cele din familia FTDI sau alegerea unui MC care are interfață USB integrată.

Circuitele FTDI [2] cel mai cunoscute sunt cele de conversie USB-RS232 FT8U232AM (USB 1.1) și FT8U232BM (USB 2.0) și cele de conversie USB-paralel FT8U245AM (USB 1.1) și FT8U245BM (USB 2.0). Protocolul USB este încorporat total în circuit și nu este nevoie de programarea formării sau gestionării cadrelor USB.

Emitătorul / receptorul USB transmit /recepționează datele USB. Motorul serial codifică / decodifică datele, assemblează cadrul USB, inserează sau verifică CRC. Datele sunt convertite în format paralel și sunt transferate printr-un protocol paralel simplu.

Un generator de tact de 6MHz cu un cristal în exterior generează semnalul de tact, care este multiplicat de 8 ori și constituie tactul intern al circuitului. Un generator de 3,3V alimentează blocurile interne dar tensiunea generată poate fi folosită și în exterior. EEPROM-ul serial memorează date privitoare la configurația circuitului.

Datele în format paralel pot fi citite sau scrise printr-un protocol controlat de semnalele RD, WR, TxE și RxF dar pot fi transferate automat cu o periodicitate dată de un timer intern, ceea ce face posibile aplicații în care FTDI nu este cuplat în partea paralelă la un microcontroller ci la un simplu element de execuție sau traductor. Acest mod de lucru se numește Bit Bang.

Schema bloc a circuitului 245BM este dată în figura 9.8.

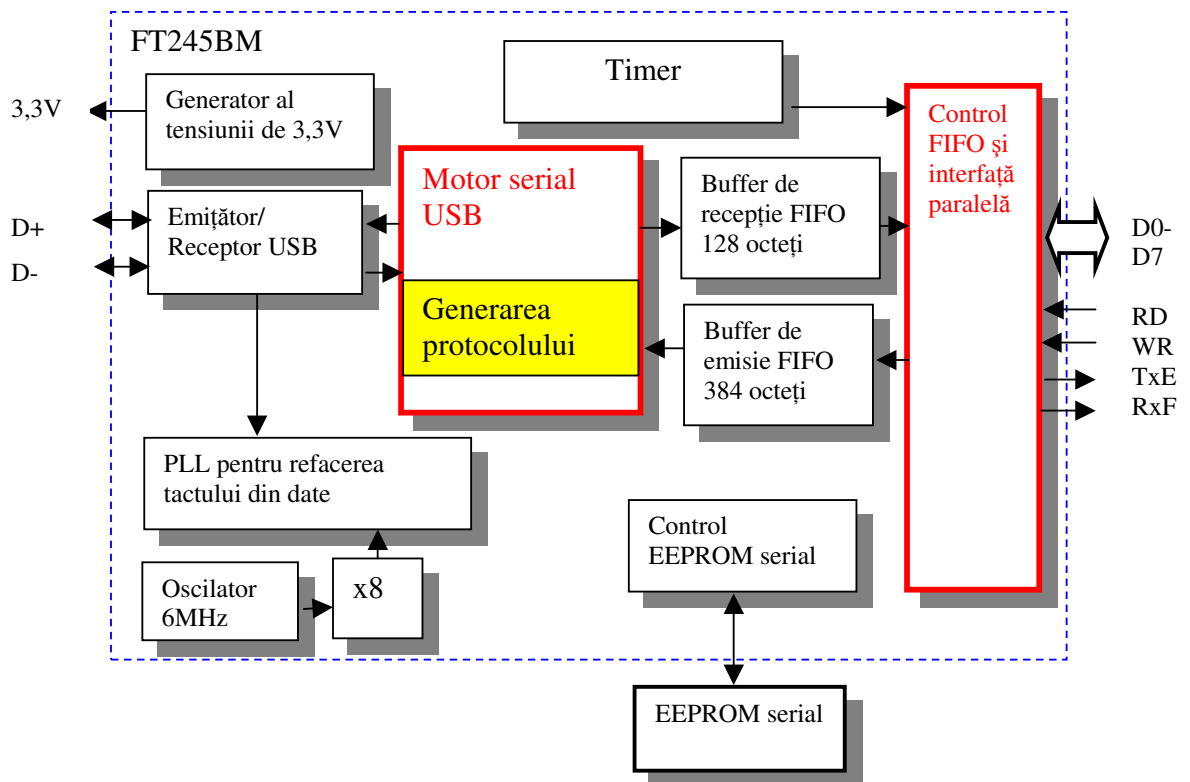


Figura 9.8. Schema bloc a unui circuit FT245BM

Scrierea datelor se face când TxE este în stare 0 logic. După memorarea octetului în bufferul de transmisie TxE devine din nou 0 logic. La recepția datelor se folosește RxF care în stare 0 logic anunță că s-a recepționat un caracter, figura 9.9.

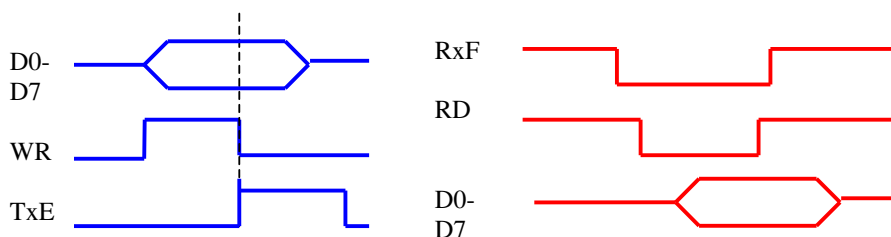


Figura 9.9. Scrierea / citirea datelor în mod paralel

Circuitul FT232BM are o schemă bloc asemănătoare, diferența fiind blocul de interfață care este în acest caz serială. Semnalele sunt cele de la RS232: TxD, RxD, RTS, CTS, DTR, DSR, DCD, RI și în plus TxDEN un semnal de validare transmisie necesar la standardul RS485.

Două semnale care arată că se transmit sau se recepționează date TxLED și RxLED pot să fie folosite la comanda unor indicatoare luminoase de activitate. Protocoalele permise sunt cele hard DTR sau CTS și soft Xon-Xoff. Un circuit generator de rată de Baud asigură tactul standard necesar transmisiei.

Interfața cu microcontrollerul este simplă și ușor de implementat, constă ca și hardware în conectarea câtorva semnale, TxD cu RxD la FT232BM și cele 8 linii de date și 4 de protocol la FT245BM. În figura 6 se vede simplitatea unei conectări a unui circuit

Interfața cu calculatorul gazdă, de regulă un PC înseamnă în primul rând descărcarea driverelor de pe site-ul FTDI pentru sistemul de operare instalat. Cuplarea circuitului FTDI la USB (în cazul sistemului de operare WINDOWS) va avea ca efect dialogul “Found new hardware...). După instalarea driverelor circuitul FTDI va apărea ca în figura 9.10:

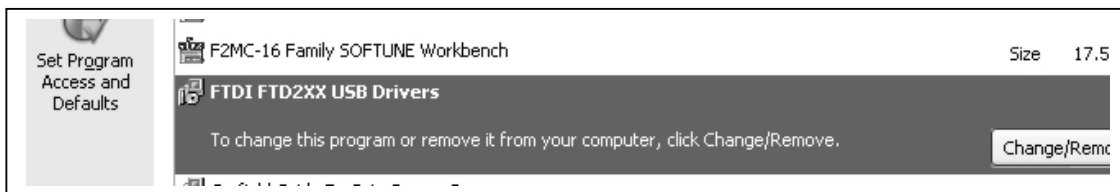


Figura 9.10: driverul software instalat sub WINDOWS

Pentru transferul datelor FTDI pune la dispoziția utilizatorului o bibliotecă DLL și documentație pentru programare (API). Astfel, o scriere / citire de date prin USB în FTDI se poate face cu o comandă FT_write / FT_read.

9.6. Microcontrollere cu USB integrat

Un model de microcontroller cu USB integrat este ATMEL AT90USB care este disponibil în diverse combinații de memorie. Interfața USB are următoarele caracteristici:

- Viteza este de 1,5 Mbps la USB 1.0 (Low Speed), 12Mbps la USB 1.1 (Full Speed);

- Conține 7 endpoint-uri cu dimensiunile de 64 octeți (endpoint 0, de control), 256 octeți (endpoint 1) și câte 64 octeți celelalte;
 - Conține o memorie dual port DPRAM de 832 de octeți pentru endpoint-uri.
- Schema bloc a interfeței USB integrate este dată în figura 9.11:

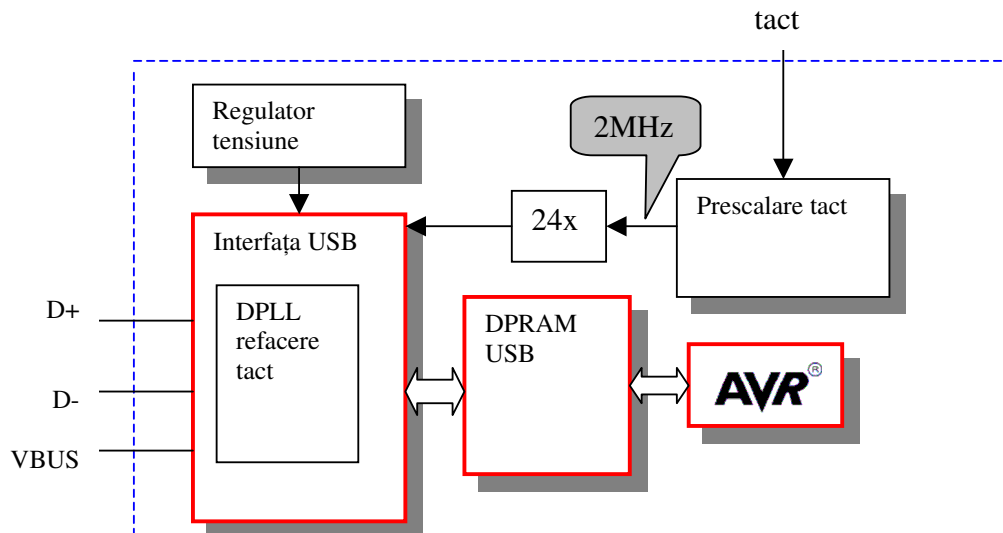


Figura 9.11: schema bloc a interfeței USB

Tactul necesar pentru interfața USB este de 48MHz cu toleranța de 0,25%, deci trebuie acordată atenție tactului extern și prescalării. Circuitul PLL este digital (DPLL) pentru a asigura performanțele necesare refacerii tactului. Regulatorul de tensiune asigură tensiunea necesară pentru D+ și D- (3V...3,6V) din tensiunea de alimentare a circuitului (5V). MC admite lucrul OTG (On-The-Go) conform cu suplimentul specificațiilor USB, care permite transmisia de semnalizări pe lina VBUS.

Modurile în care lucrează MC pot fi:

- Funcție (ATMEL denumește dispozitiv USB) cu alimentare de la USB (Figura 9.12 a);
- Funcție cu alimentare proprie (Figura 9.12 b);
- Gazdă USB (Figura 9.12 c), mod care nu este implementat în toate MC din familie.

Modul de lucru gazdă sau funcție și vitezele de transfer se selectează cu valori logice la pini de comandă.

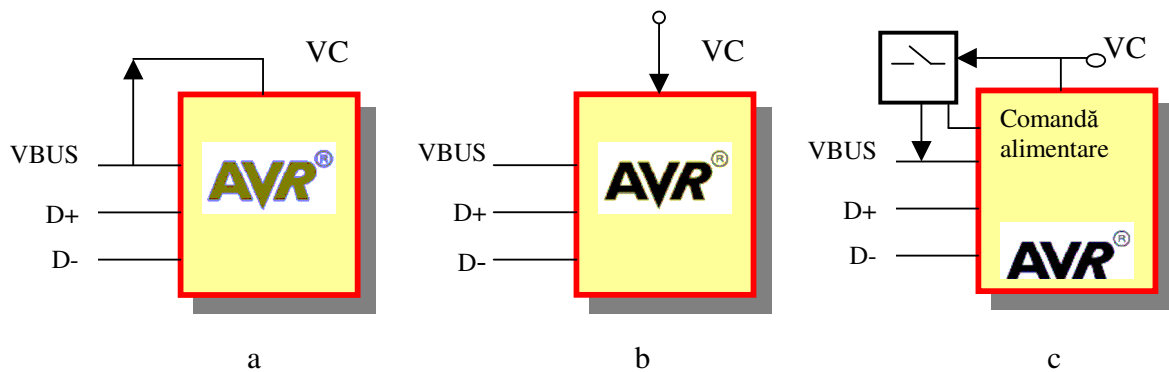


Figura 9.12: modurile de lucru ale MC

Pe liniile de date D+ și D- se inserează rezistoare serie de 22Ω .

Transferul de date bazat pe endpoint-uri și pipe-uri este reprezentat în figura 9.13:

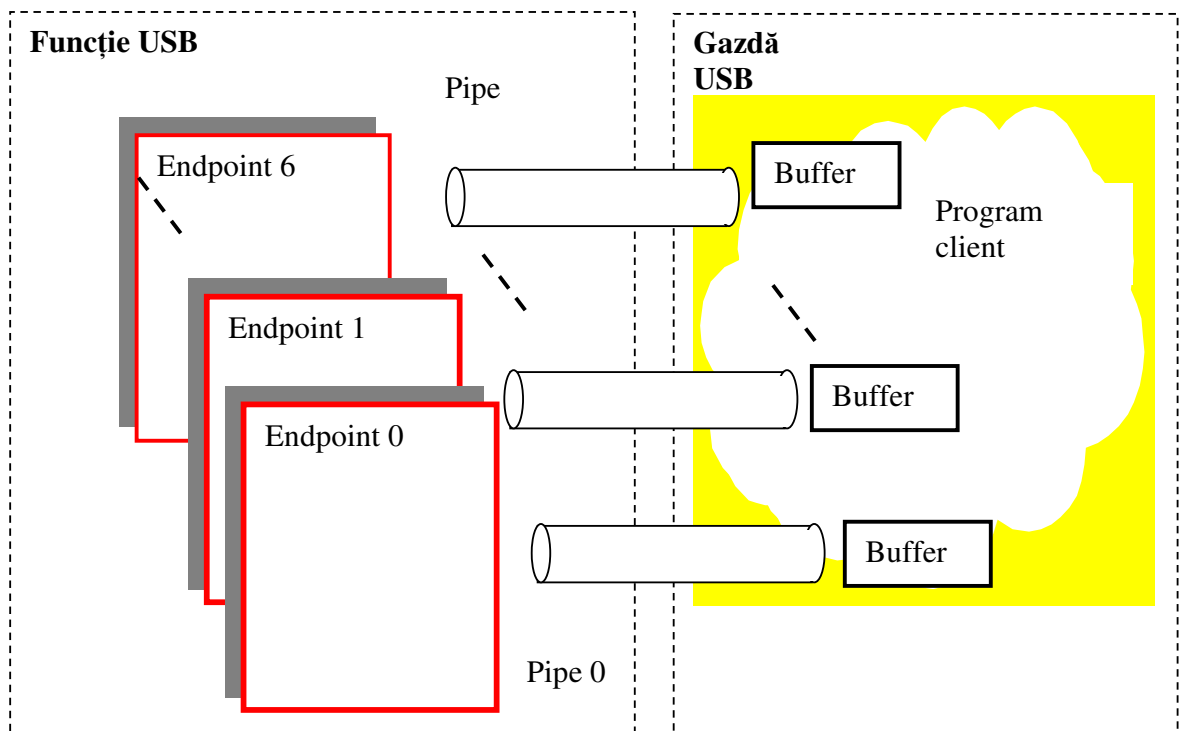


Figura 9.13. Transferul de date bazat pe endpoint-uri

Transmiterea datelor pe aceeași linie dar din surse diferite prevăzute cu buffer-e și spre destinații diferite fluidizează traficul. Mărimea endpoint-urilor poate fi programată în MC, cu anumite condiționări. Fiecare endpoint poate cere o întrerupere atunci când este plin (la recepție) sau gol (la emisie). Registrele de programare a USB au semnificații diferite pentru modul de lucru gazdă sau funcție.

MC are posibilitatea de lucru cu economie de energie în următoarele moduri:

- Mod *Idle*, în care CPU este oprit, repornirea fiind posibilă la orice întrerupere USB;
- Mod *Power Down*, în care CPU și periferia sunt oprite, repornirea fiind posibilă doar la anumite întreruperi USB;
- Mod *Freeze Clock*, în care tactul pentru USB este oprit. Intrarea în acest mod se poate comanda printr-un bit într-un registru de comandă USB. Repornirea este posibilă doar la anumite întreruperi USB.

Modul de lucru ca gazdă USB sau dispozitiv USB este determinat de valoarea logică a unui pin (UID) sau software prin poziționarea unui pin într-un registru USB. Modul de lucru Low Speed sau Full Speed se poate selecta prin valoarea logică a doi pini externi. Valorile logice de comandă pot fi stabilite local cu rezistoare sau de la distanță.

În funcție de modul de lucru ales pentru interfața USB, software-ul trebuie să comande următoarele acțiuni:

1. Pornirea interfeței USB

- pornirea regulatorului de tensiune;
- configurarea PLL, validarea PLL și comanda unui întârzieri pentru ca PLL să se sincronizeze;
- validarea și configurarea interfeței USB prin alegerea vitezei, configurarea endpoint-urilor etc.;
- atașarea unui dispozitiv USB.

2. Oprirea interfeței USB:

- detașarea dispozitivelor USB;
- invalidarea interfeței USB, a circuitului PLL și a regulatorului de tensiune.

Intrarea în modul de lucru cu economie de energie se face astfel: se oprește tactul, se oprește bucla PLL, se validează întreruperile care scot interfața din acest mod de lucru, se oprește CPU. Reintrarea în modul de lucru se face cu aceleași operații în ordine inversă. Ca exemplu de transfer de date prin USB, în figura 9.14 se arată diagrama de timp pentru un transfer de date de scriere:

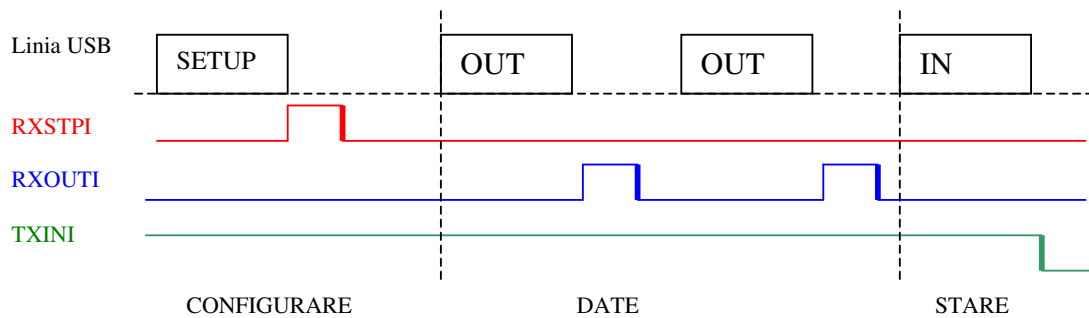


Figura 9.14. Scriere USB

Un pachet de semnalizare este generat pentru stabilirea destinației transferului. Se generează o întrerupere RXSTPI (generată la transmisia pachetului de semnalizare). Prin software se șterge informația din endpoint-ul transmis și se transmit pachete de date de ieșire. După fiecare pachet de date transmis se generează o întrerupere RXOUTI pentru a informa MC de transmisia pachetului și a putea șterge conținutul endpoint-ului folosit. După recepționarea pachetelor de date dispozitivul destinație răspunde cu un pachet de dialog pentru a confirma primirea datelor. Se generează un semnal TXINI în zero pentru a permite recepționarea unui alt pachet de dialog. Fronturile îngroșate sunt cele generate prin comenzi software iar cele neîngroșate sunt generate hardware ca întreruperi.



Rezumat

Cursul care tratează magistrala USB începe cu o descriere a contextului în care a apărut această magistrală, caracteristicile și avantajele ei. Arhitectura magistralei este descrisă alături de elementele care fac parte din structura acesteia. O scurtă prezentare a nivelului fizic și a cablurilor de conexiune folosește studenților ca utilizatori ai echipamentelor USB. Mai detaliat este prezentat protocolul de transfer de date prin cadre de date, arătând structura și tipurile cadrelor de date. Un exemplu de dialog cu cadre de date arată desfășurarea unei operații de transfer.

Pentru ca studenții să poată fi capabili să proiecteze un sistem propriu cuplat la USB sunt prezentate două variante, realizarea unei interfețe USB cu un circuit specializat și cu un microcontroller cu USB integrat.



Bibliografie

1. M. Romanca, P. Ogrutan, *Sisteme cu calculator incorporat. Aplicatii cu microcontrollere*, Editura Universitatii Transilvania Brasov, 2011, pag. 19-31
online la:
<http://vega.unitbv.ro/~ogrutan/Microcontrollere2011/3-usb-ieee1394.pdf>
2. C. Gerigan, P. Ogruțan, *Tehnici de interfațare*, Ed. Transilvania Brașov, 2000, 315p., ISBN 973-9474-94-2, pag. 114-130, online la:
<http://vega.unitbv.ro/~ogrutan/ti/index.html>
3. <http://www.usb.org/home>
4. <http://www.ftdichips.com/>

Test de autoevaluare



1. Caracteristicile USB sunt:

R

- (a) distanța de conectare este de minimum 5m
- (b) un dispozitiv fizic se poate conecta și deconecta în timpul funcționării
- (c) se pot conecta maximum 255 de dispozitive în topologie de stea multiplă
- (d) dispozitivele USB pot fi alimentate prin cablu USB de la gazdă

I. Vezi pagina 2 și pagina 3

2. Un endpoint este un dispozitiv USB care poate fi adresat de gazda USB și de care pot fi conectate alte dispozitive formând un dispozitiv complex

R

- adevărat
- fals

I. Vezi pagina 4

3. Sesizarea cuplării unui dispozitiv USB de către gazdă se face astfel: dispozitivul trimite un cadru de date în care este stocată adresa lui și gazda o memorează

R

- adevărat
- fals

I. Vezi pagina 5

4. Transmisia USB este serială, cu refacerea tactului din datele transmise. Codul trebuie să fie autosincronizabil pentru că:

R

- (a) viteza de transfer este mare
- (b) precizia tactului trebuie să fie mare, cu o eroare de maximum 0,25%
- (c) tactul este refăcut de receptor din datele recepționate
- (d) la USB nu se transmite tactul pe linie și nici nu este transmisie cu tact standard

I. Vezi pagina 4

5. Transferurile izocrone asigură un acces cu debit garantat, și cu verificarea CRC. La apariția unei erori transferul se reia sau se oprește.

R

- adevărat
- fals

I. Vezi pagina 6

6. Câmpul SYNC cu care începe orice cadru USB conține un șir dens de tranziții pentru a permite buclei PLL din receptor să se sincronizeze.

R

- adevărat
- fals

I. Vezi pagina 7

7. Câmpul PID de 8 biți conține informație privind tipul pachetului în primii 4 biți și codul CRC în următorii 4 biți pentru a confirma acuratețea primilor 4.



- adevărat
 fals

I. Vezi pagina 7

8. Circuitele convertitoare USB serial/paralel nu pot funcționa decât conectate la un sistem gazdă/ microcontroller care transferă datele convertite.



- adevărat
 fals

I. Vezi pagina 9

9. Protocolul de citire pe interfața paralelă a convertorului FTDI245 constă în: dacă RxF (RxFull) este 0 înseamnă că s-a recepționat un caracter și gazda poate activa RD. Datele sunt eșantionate pe frontul negativ al semnalului RD. RxF revine la 1 până la recepționarea unui nou caracter.



- adevărat
 fals

I. Vezi pagina 10, figura 9.9.

R

Răspunsuri corecte:

1. b, d pagina 2 și 3
2. fals, pagina 4
3. fals, pagina 5
4. c și d, pagina 4
5. fals, pagina 6
6. adevărat, pagina 7
7. fals, pagina 7
8. fals, pagina 9
9. adevărat, figura 9.9.