Capitolul 3: USB şi IEEE1394

3.1.Magistrala USB (Universal Serial Bus)

3.1.1.Descriere

Magistrala USB a fost introdusă cu dorința de a oferi utilizatorilor o interfață universală, cu viteză mare şi uşor de folosit, mai ieftină pentru că, fiind serială, cablurile şi conectorii costă mai puțin. Aceste considerente au impus magistrala USB pe piață, în prezent aceasta ocupând o cotă mare de piață în domeniul interfațărilor. Complexitatea USB este mai mare decât a interfețelor înlocuite, adică a interfeței RS232 şi 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 2013 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) şi 4,8Gbps la USB 3.0 (Super Speed);
- conectează până la 127 de dispozitive la un calculator gazdă, dar nu se pot conecta dispozitive USB fără gazdă ca la IEEE 1394;
- configurarea este automată, adică se poate conecta un dispozitiv USB fizic în mers (Hot Plug In). Se remarcă creşterea complexității software față de partea hardware;
- cablul conține linii de alimentare, aşa 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) şi 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) şi totodată furnizează informații necesare pentru construirea unui sistem în acest standard.

Magistrala USB defineşte trei categorii de dispozitive fizice:

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

Dispozitivele USB sunt conectate într-o topologie de tip stea multiplă. Topologia USB este reprezentată în figura 3.1. În nodul fiecărei stele se găseşte un hub.

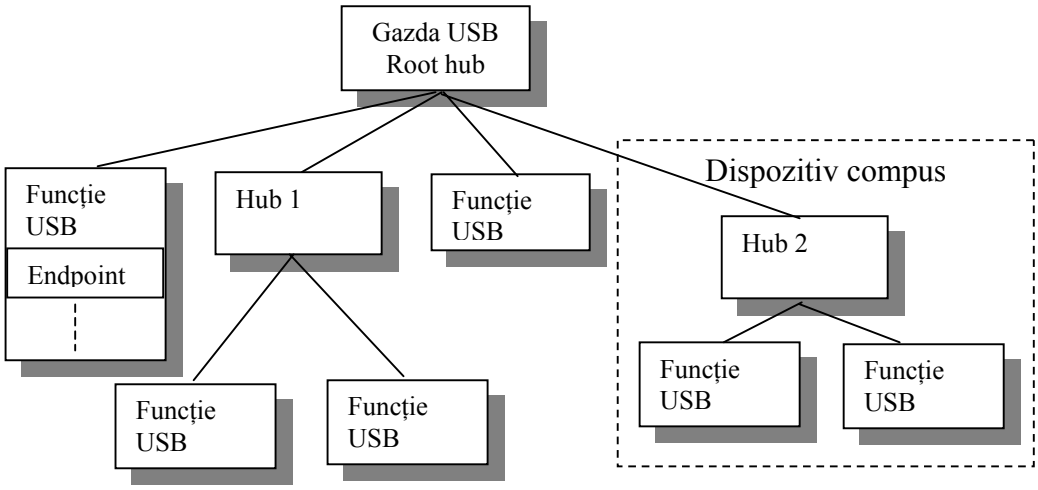

Figura 3.1: Topologia USB

Legătura este multipunct pe magistrală dar punct la punct între hub şi dispozitive. Este posibil ca un dispozitiv fizic să conțină mai multe funcții şi un hub, acest dispozitiv numindu-se compus. Un exemplu este o multifuncțională care conține imprimantă, scanner şi 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 şi sensul tranzacției, adresa dispozitivului USB şi 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.

Aspectele electrice şi mecanice ale interfeței sunt reglementate foarte precis în specificațiile de magistrală. Semnalele electrice sunt transmise diferențial (D+ şi D-). Codificarea utilizată este NRZI cu împănare de biți (*bit-stuffing*) şi 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}) şi potențialul de referință (GND), figura 3.2.

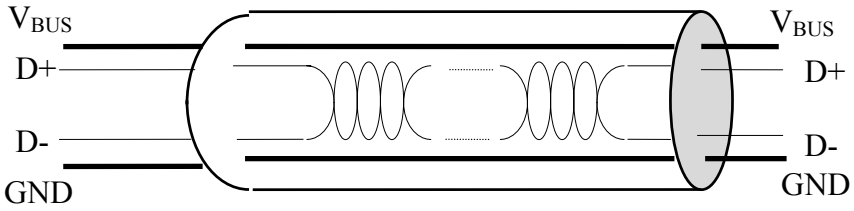

Figura 3.2: Cablu USB

Tensiunea transmisă pe linie nu este tensiunea de alimentare a calculatorului gazdă ci este gestionată de controllerul USB, aşa încât o suprasarcină este detectată şi un mesaj de eroare este afişat de sistemul de operare.

Uşurința cu care este utilizată USB rezultă din atributul special de tip plug-and-play al acestei magistrale. USB acceptă cuplarea şi 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 ataşarea 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 şi atribuie o adresă unică dispozitivului. La decuplare hub-ul dezactivează portul şi indică gazdei acest eveniment. Pentru a se adapta dinamic, sistemul software USB este permanent într-un proces de inventariere a magistralei (bus counting).

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

- transferuri de control (*Control Transfers*) - sunt folosite pentru configurare şi comandă şi 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 şi tolerează erorile de transmisie; datele izocrone sunt continue şi în timp real la toate nivelele: generare, emisie, recepție şi utilizare la receptor; acest tip de transfer este folosit pentru fluxuri de transfer în timp real cum ar fi sistemele audio.

3.1.2.Protocolul USB

USB foloseşte 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 dint-un cadru nu are o valoare fixă.

Majoritatea sistemelor folosesc cadre de 4 până la 8 biți de date. Biții sunt trimişi 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 defineşte 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)

Toate pachetele conțin la începutul lor un câmp de sincronizare, numit SYNC care permite buclei PLL pentru refacerea tactului din receptor să se sincronizeze, şi 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 şir 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 şi are lungimea de 8 biți. Primii patru biți indică tipul pachetului, iar următorii patru sunt în ordine primii patru complementați (complement față de 1) şi sunt folosiți ca biți de verificare pentru a confirma acuratețea primilor patru.

Orice transfer începe prin trimiterea de către gazdă a unui pachet de semnalizare. Un pachet are 32 de biți împărțiți în cinci câmpuri. Structura pachetului este reprezentată în figura 3.3A. Informația propriu-zisă este transferată în sistemele USB sub forma unor pachete de date. Structura acestui pachet este dată în figura 3.3B. După câmpurile SYNC şi 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 de 16b aflat la sfârşitul pachetului.

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 şi PID. Structura este reprezentată în figura 3.3C.

Câmpul PID defineşte 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.

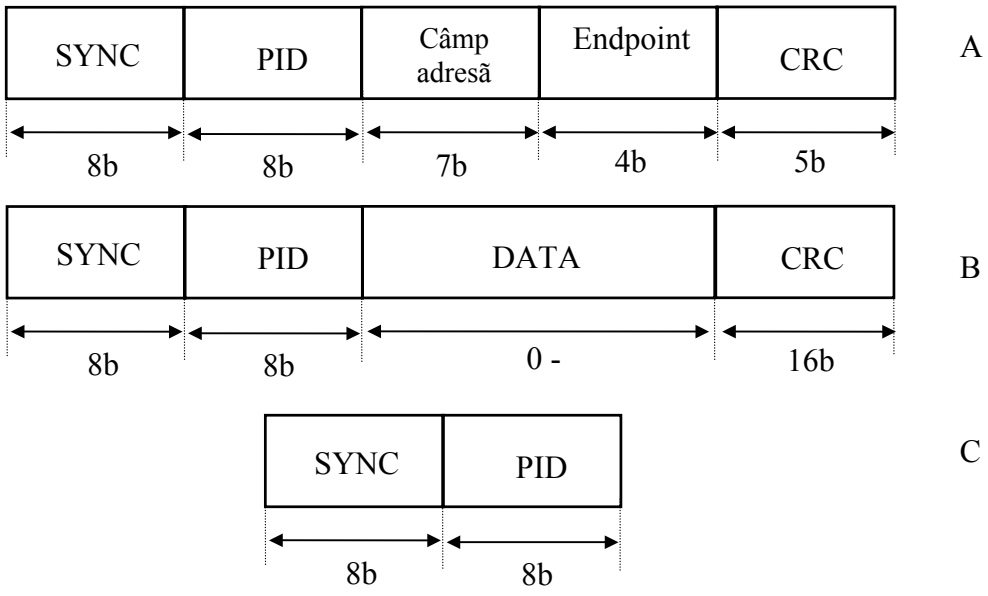

Figura 3.3: Structura pachetelor USB

3.1.3.Cuplarea unui MC la USB printr-o interfață specializată

Dacă aplicația necesită cuplarea unui MC la USB atunci există două variante, utilizarea unei interfețe specializate aşa 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) şi FT8U232BM (USB 2.0) şi cele de conversie USB-paralel FT8U245AM (USB 1.1) şi FT8U245BM (USB 2.0). Protocolul USB este încorporat total în circuit şi nu este nevoie de programarea formării sau gestionării cadrelor USB.

Emițătorul / receptorul USB transmit /recepționează datele USB. Motorul serial codifică / decodifică datele, asamblează cadrul USB, inserează sau verifică CRC. Datele sunt convertite în format paralel şi 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 şi constituie tactul intern al circuitului. Un generator de 3,3V alimentează blocurile interne dar tensiunea generată poate fi folosită şi 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 şi 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 numeşte Bit Bang.

Schema bloc a circuitului 245BM este dată în figura 3.4.

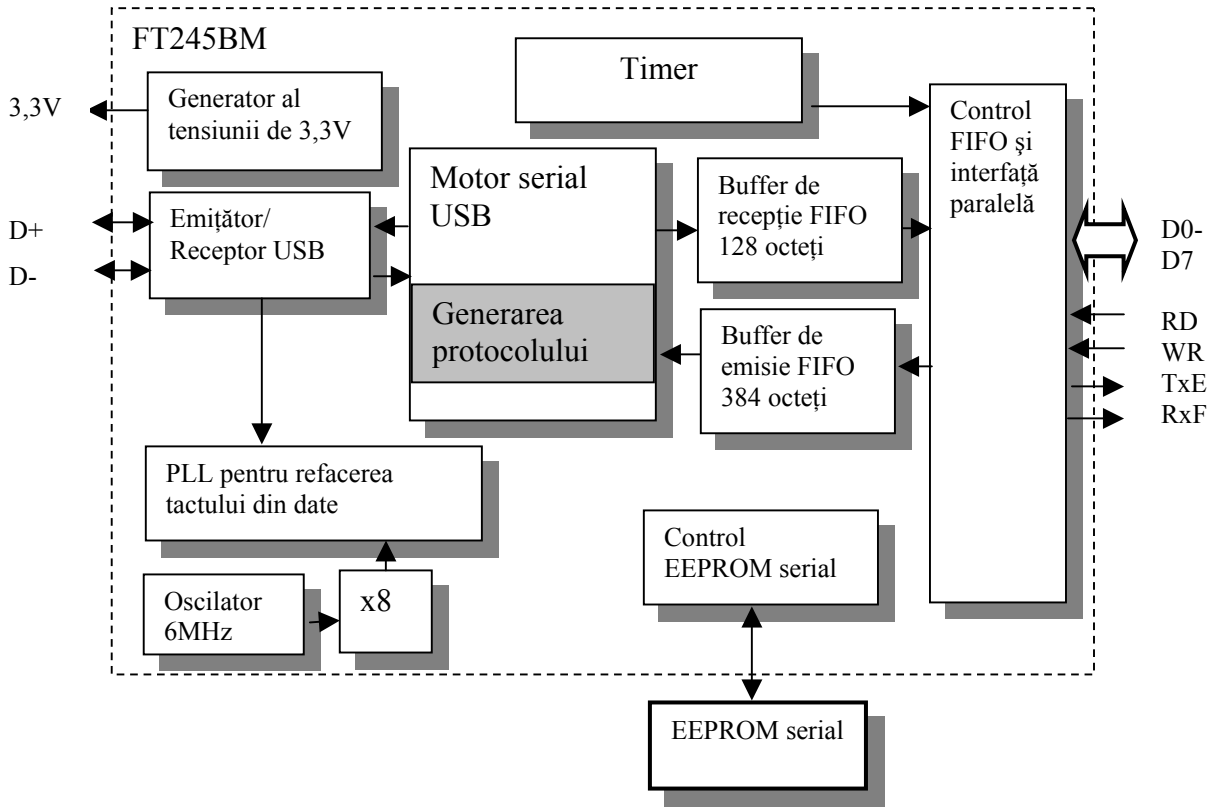

Figura 3.4: Schema bloc a unui circuit FT245BM

Scrierea datelor se face când TxE este în stare 0 logic. După memorarea octetului în bufferul de transmisie TxE devine din nou 0 logic, figura 3.5. La recepția datelor se folosește RxF care în stare 0 logic anunță că s-a recepționat un caracter, figura 3.5.

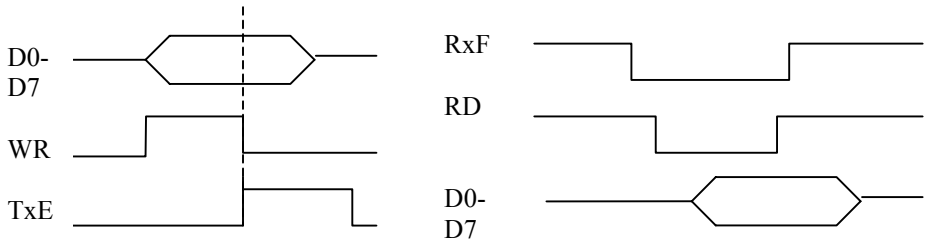

Figura 3.5: Scrierea / citirea datelor în mod paralel

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 şi în plus TxDEN un semnal de validare transmisie necesar la standardul RS485.

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

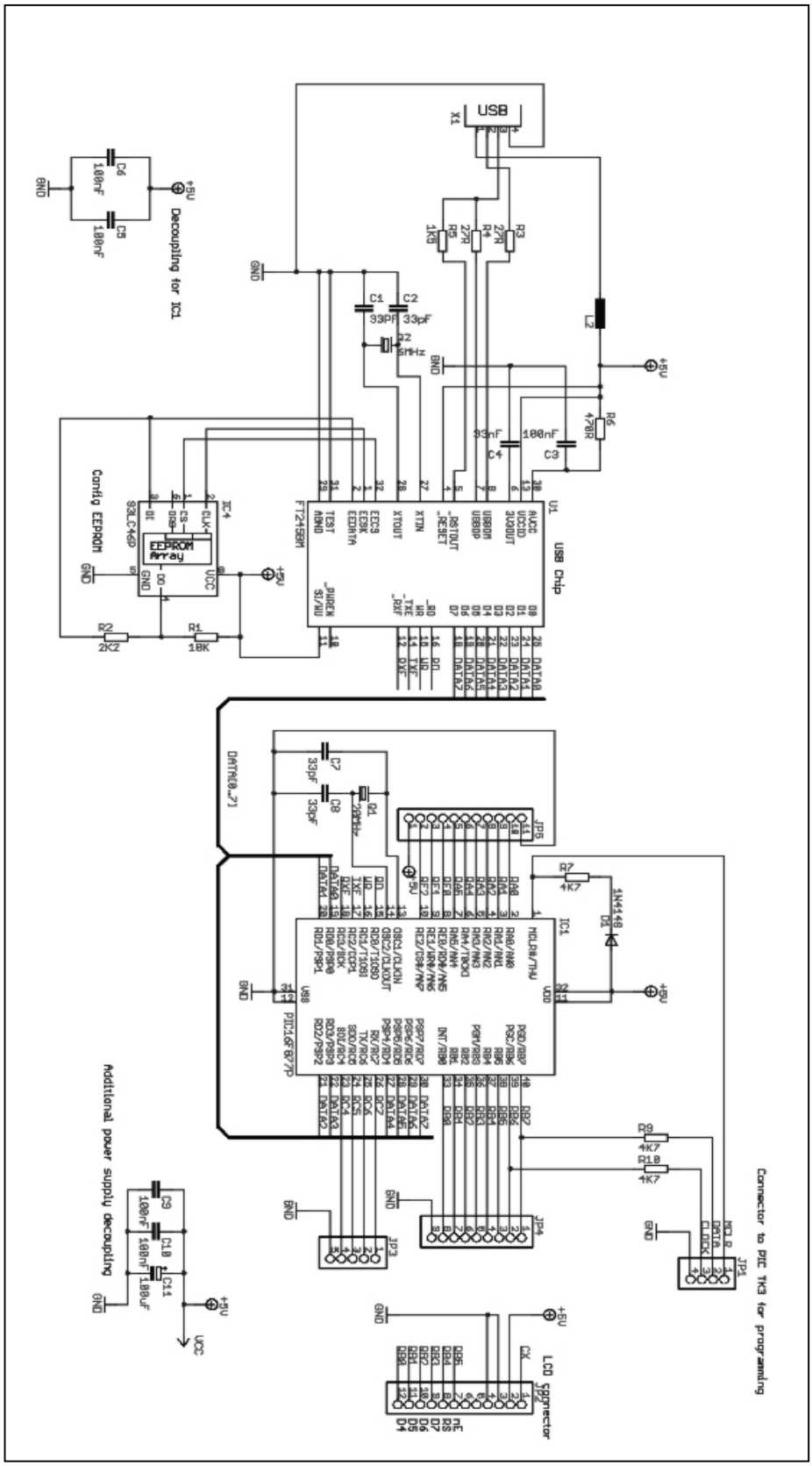

Figura 3.6: Conectarea FT245BM la un MC PIC16F

Interfața cu microcontrollerul este simplă şi uşor de implementat, constă ca şi hardware în conectarea câtorva semnale, TxD cu RxD la FT232BM şi cele 8 linii de date şi 4 de protocol la FT245BM. În figura 3.6 se vede simplitatea unei conectări a unui circuit FT245BM la un microcontroller PIC16F [3]. 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 3.7.

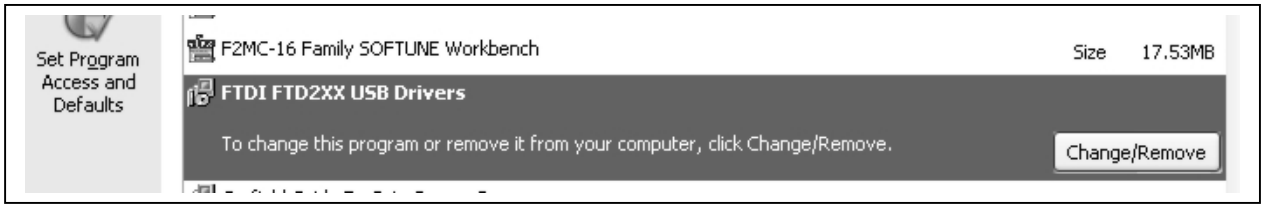

Figura 3.7: Driverul software instalat sub WINDOWS

Pentru transferul datelor FTDI pune la dispoziția utilizatorului o bibliotecă DLL şi documentație pentru programare (API). Astfel, o scriere / citire de date prin USB în FTDI se poate face cu o comandă FT_write / FT_read. Un exemplu de citire a tuturor datelor disponibile în circuit, preluat din D2XX Programmer's Guide [2]:

```
FT_HANDLE ftHandle;
FT_STATUS ftStatus;
DWORD EventDWord;
DWORD TxBytes;
DWORD RxBytes;
DWORD BytesReceived;
char RxBuffer[256];

ftStatus = FT_Open(0, &ftHandle);
if(ftStatus != FT_OK) {
	// FT_Open failed
	return;
}

FT_GetStatus(ftHandle,&RxBytes,&TxBytes,&EventDWord);
if (RxBytes > 0) {
	ftStatus = FT_Read(ftHandle,RxBuffer,RxBytes,&BytesReceived);
	if (ftStatus == FT_OK) {
		// FT_Read OK
	}
	else {
		// FT_Read Failed
	}
}

FT_Close(ftHandle);
```

Uşurința implementării unei interfețe USB a condus la realizarea unui proiect prin care a fost realizată o lucrare de laborator [4]. Proiectul a constat în realizarea unei plăci de interfață USB paralel cu circuitul FT245BM. La partea paralelă a circuitului a fost conectată o bară 8 de LED-uri ca ieşire şi 8 întrerupătoare ca intrare, figura 3.8. Sensul datelor a fost stabilit de un microîntrerupător. Circuitul FT245BM a fost lipit pe partea din spate a plăcii. Studenții au primit software-ul de lucru şi au avut ca sarcină aprinderea LED-urilor şi citirea întrerupătoarelor.

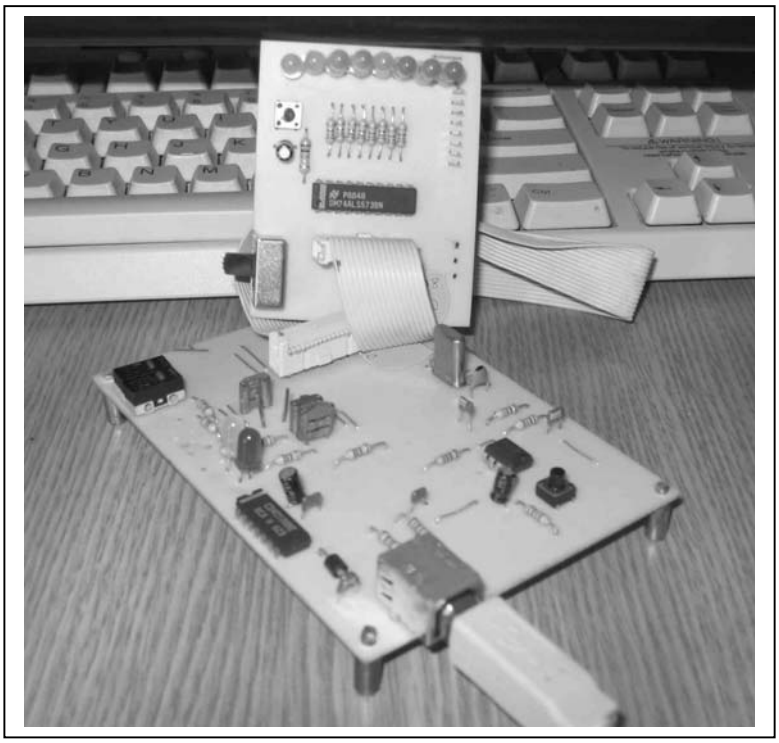

Figura 3.8: Lucrare de laborator cu FT245BM

3.1.4.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) şi 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 3.9.

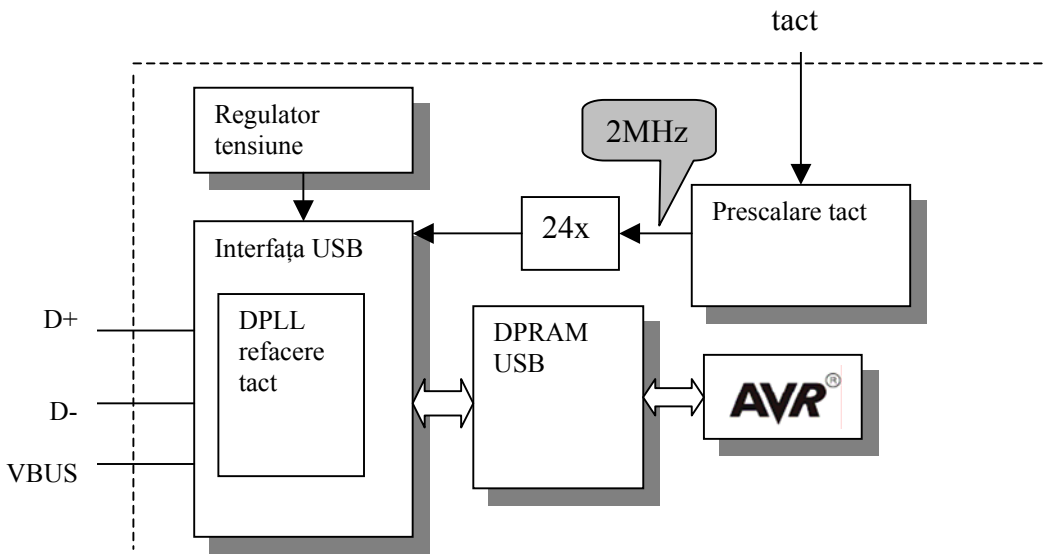

Figura 3.9 Schema bloc a interfeței USB

Tactul necesar pentru interfața USB este de 48MHz cu toleranța de 0,25%, deci trebuie acordată atenție tactului extern şi prescalării. Circuitul PLL este digital (DPLL) pentru a asigura performanțele necesare refacerii tactului. Regulatorul de tensiune asigură tensiunea necesară pentru D+ şi 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 denumeşte dispozitiv USB) cu alimentare de la USB (Figura 3.10 a);
- Funcție cu alimentare proprie (Figura 3.10 b);
- Gazdă USB (Figura 3.10 c), mod care nu este implementat în toate MC din familie.

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

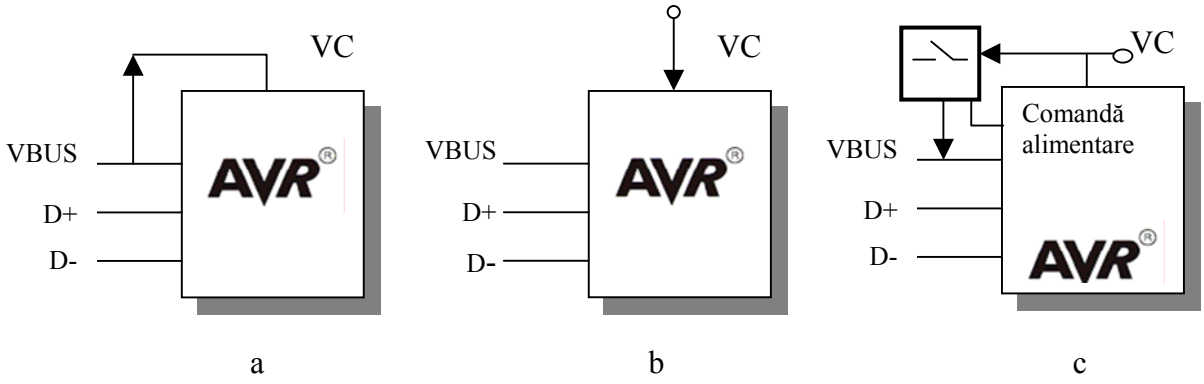

Figura 3.10: Modurile de lucru ale MC

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

Transferul de date bazat pe endpoint-uri şi pipe-uri este reprezentat în figura 3.11:

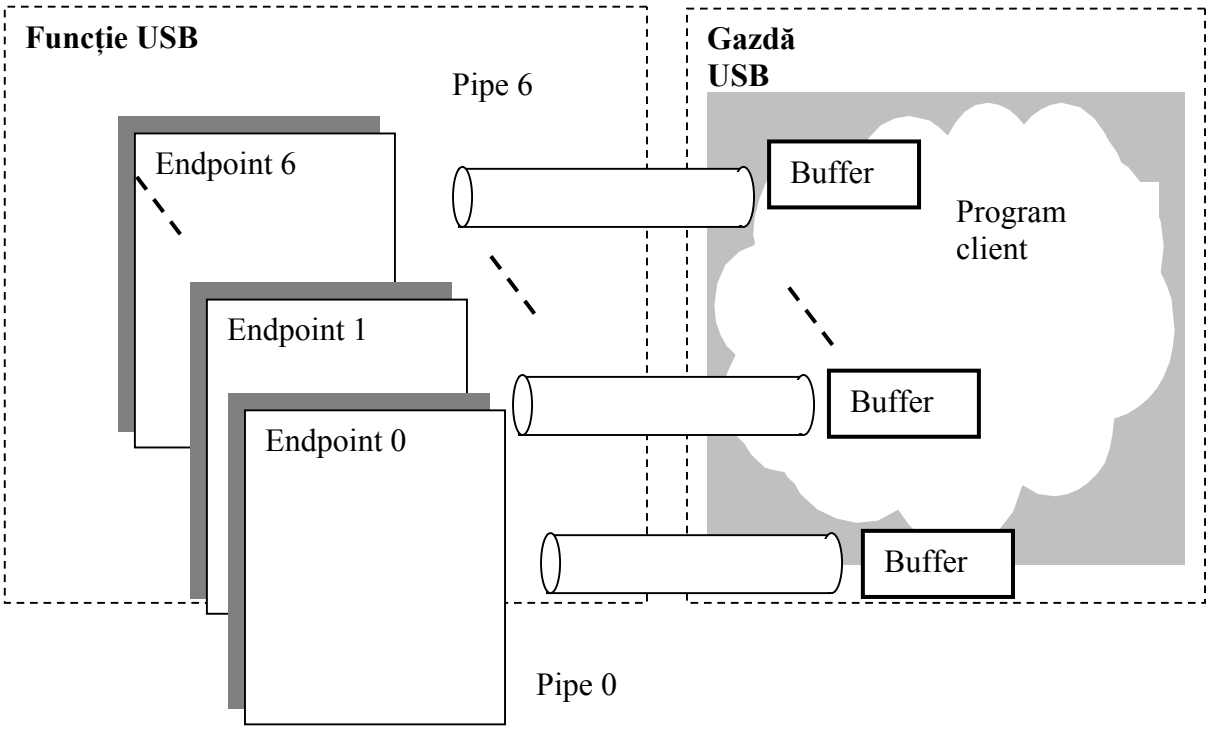

Figura 3.11: Transferul de date bazat pe endpoint-uri

Transmiterea datelor pe aceeaşi linie dar din surse diferite prevăzute cu buffer-e şi 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 şi 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 şi comanda unui întârzieri pentru ca PLL să se sincronizeze;
- validarea şi configurarea interfeței USB prin alegerea vitezei, configurarea endpoint-urilor etc.;
- ataşarea unui dispozitiv USB.

2.Oprirea interfeței USB:

- detaşarea dispozitivelor USB;
- invalidarea interfeței USB, a circuitului PLL şi a regulatorului de tensiune.

Intrarea în modul de lucru cu economie de energie se face astfel: se opreşte tactul, se opreşte bucla PLL, se validează întreruperile care scot interfața din acest mod de lucru, se opreşte CPU. Reintrarea în modul de lucru se face cu aceleaşi operații în ordine inversă. Ca exemplu de transfer de date prin USB, în figura 3.12 se arată diagrama de timp pentru un transfer de date de scriere:

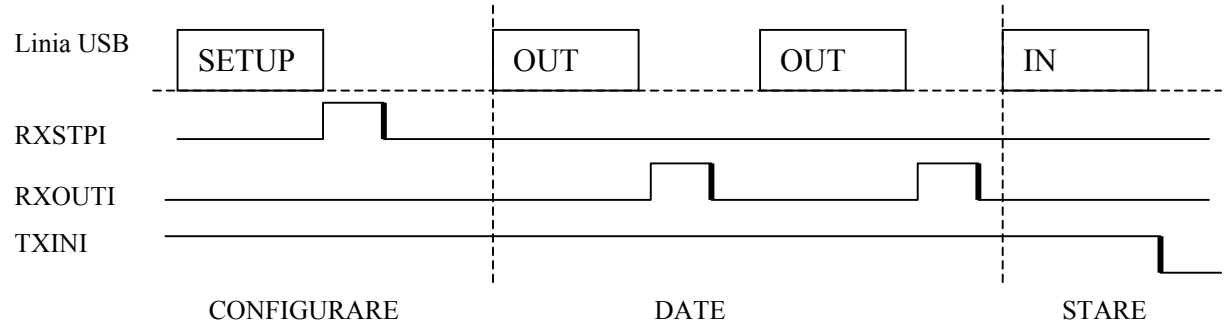

Figura 3.12: Scriere USB

Un pachet de semnalizare este generat pentru stabilirea destinației transferului. Se generează o întrerupere RXSTPI (generată la transmisia pachetului de semnalizare). Prin software se şterge informația din endpoint-ul transmis şi se transmit pachete de date de ieşire. După fiecare pachet de date transmis se generează o întrerupere RXOUTI pentru a informa MC de transmisia pachetului şi a putea şterge 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 îngroşate sunt cele generate prin comenzi software iar cele neîngroşate sunt generate hardware ca întreruperi.

Simplitatea constructivă a conectării unui MC pe USB este demonstrată de schema electrică a unei păci de dezvoltare, figura 3.13 [5].

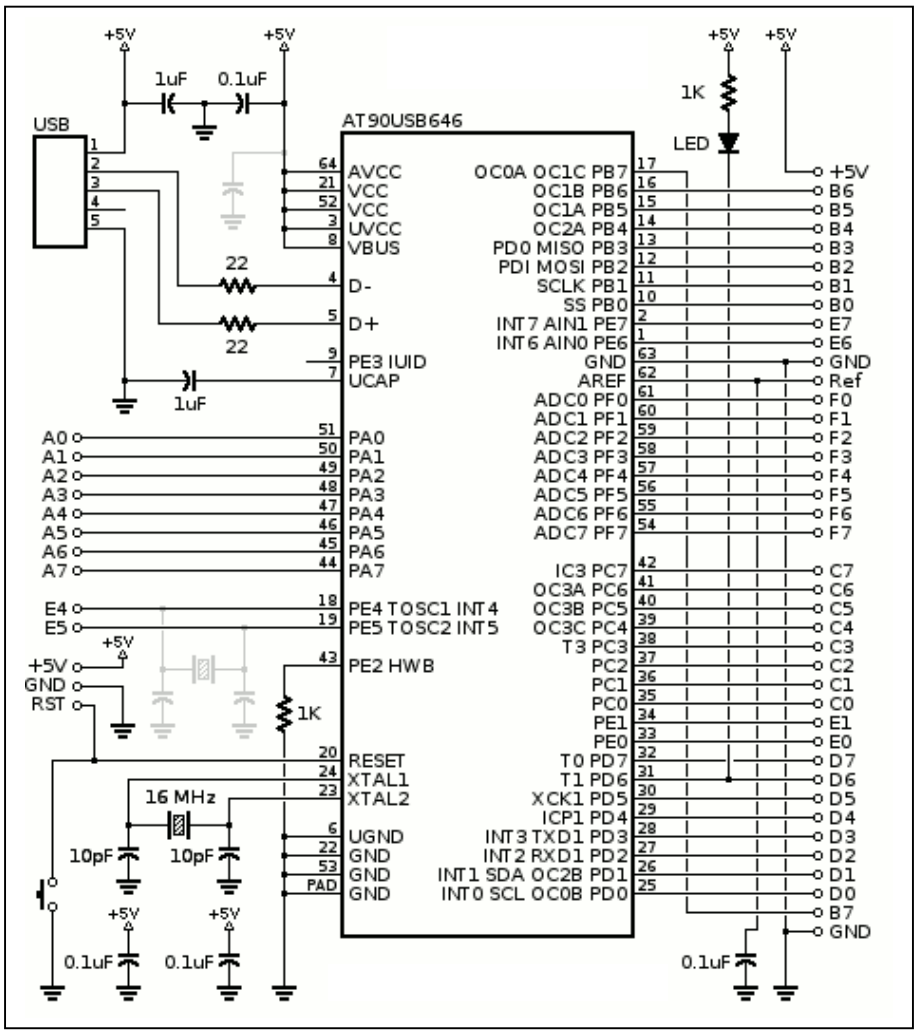

Figura 3.13: schema electrică a unei plăci de dezvoltare Teensy

3.2.IEEE 1394

IEEE 1394 este o interfață serială cunoscută sub numele de FireWire (Apple), i.LINK (Sony) şi LYNX (TI). Interfața a fost adoptată de HANA (High Definition Audio-Video Alliance) ca interfață standard disponibilă şi wireless, pe fibră optică sau pe cablu coaxial. Dezvoltarea interfeței a început în 1980 şi a fost încheiată în 1995. IEEE 1394 a fost aplicată şi în aviația militară ca magistrală pentru F-22 Raptor şi F-35. Navetele spațiale NASA au folosit IEEE 1394 pentru anumiți senzori. În industria auto a fost implementată o versiune numită IDB 1394. Cu toate că IEEE 1394 nu are răspândirea pe care o are USB, majoritatea camerelor digitale sunt echipate cu o astfel de interfață.

Ca şi în majoritatea comunicațiilor seriale transferul de date este bazat pe pachete. Canalul comun de date este conceput să poată fi folosit pe rând de fiecare dispozitiv care îl solicită. Există un interval de timp specificat (numit *fairness* interval) în cadrul căruia un dispozitiv are accesul la canalul de date comun. După ce dispozitivul a trimis un pachet de date se aşteaptă scurgerea unui timp de separare (numit *sub-action gap*) după care un alt dispozitiv poate

trimite un pachet. Dacă după scurgerea timpului de separare nici un dispozitiv nu are de transmis vreun pachet, urmează o secvență de reset.

Pentru a face posibilă funcționarea dispozitivelor care necesită flux de date în timp real, IEEE-1394 foloseşte un mod special de transfer, modul izocron, ca şi USB. Un dispozitiv ce necesită date izocrone emite la fiecare 125μs un pachet special de temporizare prin care asigură prioritatea transferului. Această schemă de arbitrare garantează un minim de buffer-e pentru date audio sau video (1 byte la dispozitive audio, până la 6 bytes la dispozitive video). Perioada de 125μs coincide cu perioada de eşantionare din sistemul de telefonie digitală, astfel interfața IEEE-1394 poate fi plasată în sistemul ISDN (Integrated Service Digital Network).

IEEE 1394 este asemănătoare cu USB, aşa încât este utilă o comparație:La IEEE 1394 nu este nevoie de un calculator gazdă;

- IEEE 1394 asigură o viteză efectivă de transfer mai mare decât USB (dovedit pe sistemul de operare MAC OS X dar cu rezultate contradictorii sub Windows);
- Implementarea IEEE 1394 are costuri mai mari: licența Apple (0.25$/sistem) şi hardware mai scump cu 1-2$;
- Ambele standarde pun la dispoziție prin cablul de transmisie de date o tensiune de alimentare, sunt plug and play şi admit hot swapping. IEEE 1394 admite tronsoane de cablu de maximum 4.5m şi poate alimenta o sarcină cu consum de până la 45W.
- Fiecare dispozitiv IEEE 1394 are un identificator propriu unic, (IEEE EUI-64) care este o adresă asemănătoare cu adresa MAC de 48 de biți.

În decursul timpului au fost realizate mai multe variante constructive:

a.FireWire 400 (IEEE 1394/1995). Versiunea originală poate transfera date cu viteze de 100, 200 sau 400 Mbps (S100, S200, S400) în mod half duplex. Modul de codificare al datelor este data strobe D/S;

b. FireWire 800 (IEEE 1394b/2002). Versiunea a doua asigură o viteză de 800Mbps în mod full duplex. Conectica este diferită față de varianta anterioară. Modul de codificare al datelor este 8B10B.

c.FireWire S800T (IEEE 1394c/2006). Versiunea a treia utilizează cablu Ethernet categoria 5e. Nu există încă implementări în sisteme disponibile pe piață datorită confuziei posibile la o placă de bază echipată cu 2 conectori RJ45, unul cu interfață Ethernet şi unul IEEE 1394.

d.**FireWire S1600 şi S3200**Se lucrează la versiunile de 1.6Gbps şi 3.2Gbps, care vor fi concurenți pentru USB 3.0. Conectorii sunt cei de la versiunea FireWire 800.

Se poate implementa o rețea de calculatoare prin legături IEEE 1394 în mod IPv4 sau IPv6. Sistemele de operare care include suport pentru acest tip de rețea sunt MAC OS X, Windows ME, 2000, XP şi Server 2003. Windows Vista şi Server 2008 nu mai conțin acest suport.

În figura 3.14 este dat un tabel cu conexiunile la conectorii IEEE 1394 cu 4, 6 şi 9 pini şi structura unui cablu IEEE 1394.

4 pini	6 pini	9 pini	Funcție	Descriere
	1	8	Vcc	30V nestabilizat
	2	6	Masă	Masă pentru tensiune
1	3	1	TPB-	semnal diferențial B
2	4	2	TPB+	semnal diferențial B
3	5	3	TPA-	semnal diferențial A
4	6	4	TPA+	semnal diferențial A
		5	ecran A	
		9	ecran B	

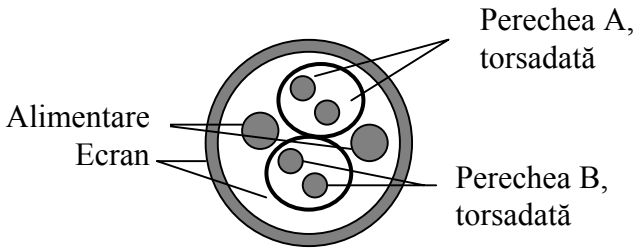

Figura 3.14: Tabel cu conexiunile la conectorii IEEE 1394 (sus) şi structura unui cablu IEEE 1394 (jos)

Codificarea datelor D/S este de fapt o codificare NRZ cu transmiterea tactului şi necesită 2 linii de semnal, una de date şi una de strob. Un SAU Exclusiv între cele 2 semnale reconstituie tactul, figura 3.15. Petru transmisia datelor este nevoie de ambele perechi FireWire, deci este posibil doar un transfer half duplex. Codificarea este aplicată la FireWire 400. Codificarea 8B10B a fost imaginată de Al. Widmer şi P. Franaszek de la IBM în 1983 şi IBM a obținut un patent. Răspândirea codificării a luat avânt după expirarea patentului. Aplicațiile dovedesc eficiența codificării: PCI Express, SATA, SAS, Fibre Channel, IEEE 1394b, Gigabit Ethernet (mai puțin la 1000BaseT), DVI, HDMI, USB 3.0 şi seamănă cu codificarea folosită la CD (Eight to Fourteen Modulation).

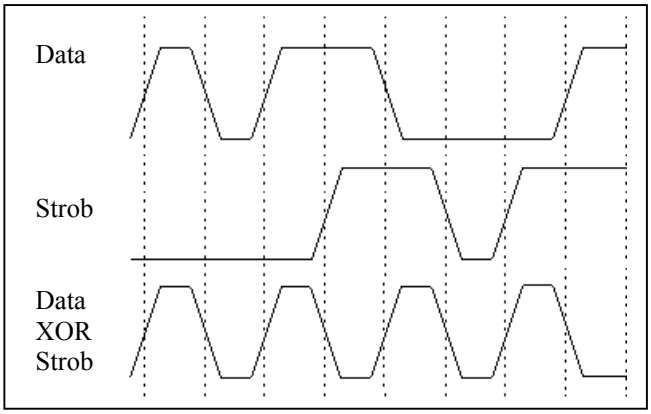

Figura 3.15: Codificarea Data /Strobe

În principiu codificarea asigură o componentă DC mică pentru ca şirul de date să poată trece prin transformatorul de separare Ethernet, adică numărul de 0-uri este aproape egal cu numărul de 1-uri. Într-un şir de 20 biți diferența între numărul de 0 şi de 1 poate fi maxim 2. Codul este autosincronizabil şi se admit maxim 5 valori de 0 sau de 1 succesive.

Codificarea atribuie la 8b o entitate de 10b numită simbol sau caracter. La 5b mai puțin semnificativi se atribuie 6b (porțiunea 5b/6b) iar la 3b mai semnificativi se atribuie 4b (porțiunea 3b/4b). Se definesc 12 simboluri speciale de control care marchează începutul cadrului, sfârşitul cadrului, skip, etc. Datorită codării cuvintelor de 8b cu simboluri de 10b anumite valori din cele 1024 pot fi excluse pentru a realiza condiția de a nu exista 5 valori de 0 sau de 1 consecutive.

Pe linie se transmite întâi porțiunea 5b/6b apoi 3b/4b. Datele pot fi notate ca D.x.y unde x este porțiunea 5b/6b şi poate fi 0-31 iar y este porțiunea 3b/4b şi poate fi 0-7 ca valori necodate.

Se defineşte RD (Running Disparity ca diferența între numărul de biți de 1 şi numărul de biți de 0. Se urmăreşte obținerea RD cât mai mic. În acest scop grupurile 5b/6b şi 3b/4b se stabilesc în funcție de RD anterior ca valori negate sau nenegate.

De exemplu:

D.00 se codifică ca 100111 (RD inițial este –1 şi rezultă RD=+1) sau 011000 (RD inițial este +1 şi rezultă RD=–1)

La fel, în funcție de RD inițial se codifică şi grupul 3b/4b

D.x.0 se codifică ca 1011 (RD inițial este –1 şi rezultă RD=+1) sau 0100 RD inițial este +1 şi rezultă RD=-1). Astfel în ipoteza RD inițial –1, D.00.0 se codifică ca 1001110100 şi rezultă RD=0

Topologia unei arhitecturi IEEE 1394 este de tip stea multiplă (arbore) cu posibilitatea de înlănțuire (daisy-chain).

În figura 3.16 sunt prezentate două spații de lucru unite cu un bridge. Cele 2 spații sunt izolate din punctul de vedere al traficului de date. Spațiul 1 de lucru ocupă mare parte a benzii din cauza traficului video, dar în spațiul de lucru 2 calculatorul are întregul control al traficului. Este posibil ca şi calculatorul 2 să solicite date video, chiar dacă calculatorul 1 este oprit.

Este figurat un repetor care măreşte distanța de conectare şi un splitter care adaugă 2 porturi unui port IEEE 1394.

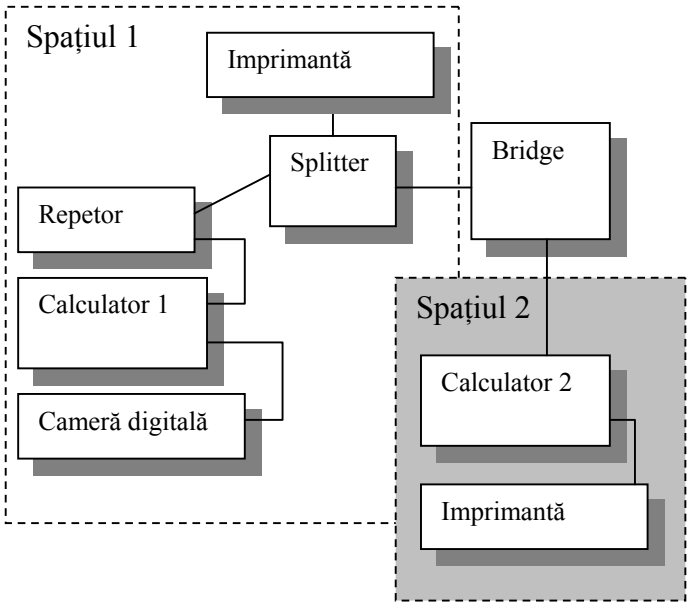

Figura 3.16: Topologia unei arhitecturi IEEE 1394

Pentru a transmite date în mod asincron dispozitivul IEEE 1394 compune un cadru care conține adresele sursei şi destinației, apoi date şi CRC. Când receptorul acceptă datele un cadru de confirmare este trimis la transmițător. Transmițătorul are posibilitatea să trimită încă 63 de cadre continuu pentru a mări viteza de transfer. Dacă cadrul de confirmare returnează o eroare se aplică o metodă de reacție la eroare.

În mod izocron emițătorul solicită un canal izocron iar dacă receptorul îl acceptă i se asigură un interval de timp de transfer pentru a asigura banda necesară transferului. Se pot defini până la 64 de canale izocrone. În exemplul din figura 3.17 în pachetul de date de 125μs sunt definite 2 intervale de timp pentru 2 transferuri izocrone. Timpul rămas liber se poate folosi la transferuri asincrone.

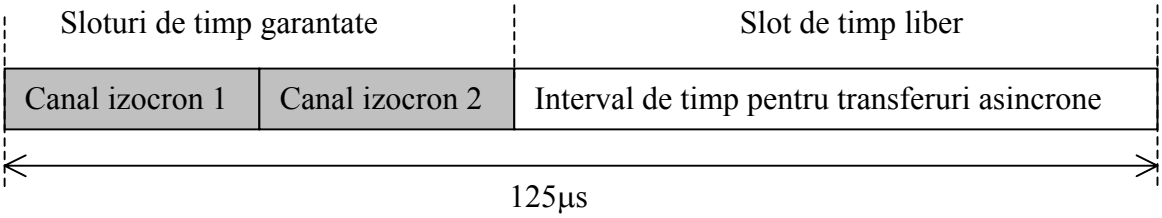

Figura 3.17: Transferuri izocrone pentru a asigura un flux de date în timp real

3.2.1.Module IEEE 1394

Nivelele ISO OSI (Open Systems Interconnection) în cazul IEEE 1394 sunt date simplificat în figura 3.18.

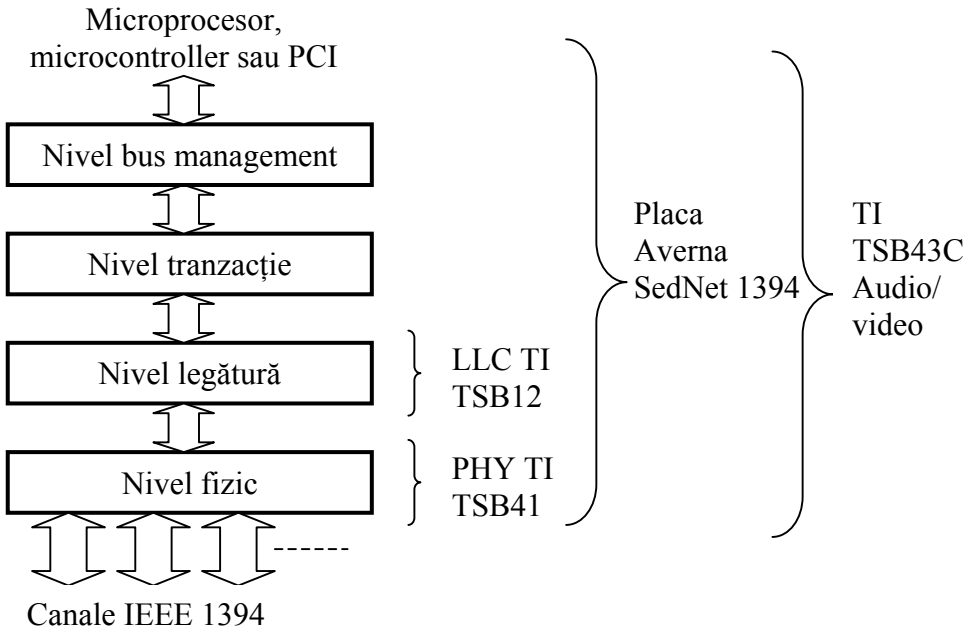

Figura 3.18: Structura stivei OSI la IEEE 1394

Nivelul bus-management

defineşte funcţiile de bază de control precum şi registrele de control şi de stare necesare dispozitivelor conectate pentru a-şi face porturile operaţionale. Acest nivel se ocupă şi de asigurarea canalelor, de arbitrare, mastering şi de erori.

Nivelul de tranzacţie

mediază operaţiile de scriere şi citire. Standardul permite la acest nivel operaţii cu cuvinte de lungime variabilă.

Nivelul de legătură

realizează controlul logic în legătura IEEE 1394. Acest nivel realizează formarea cadrele la transmisie şi extragerea informației din cadrele recepționate.

Nivelul fizic

presupune atât protocolul transferului cât şi mediul efectiv de transfer. Partea de protocol controlează accesul la legătură, iar partea de mediu este constituită din cabluri şi conectori. La acest nivel se realizează codarea şi decodarea datelor, se asigură nivelele de tensiune necesare şi se face arbitrarea magistralei.

Circuitele IEEE 1394 au un grad de complexitate mai mare decât cele USB, unul dintre motive fiind acela că pot lucra şi independent de calculatorul gazdă. Multe circuite sunt realizate cu magistrală PCI pentru a fi utilizate în calculatoare PC. În figura 3.18 sunt date câteva exemple de circuite şi plăci şi nivelele OSI pe care le acoperă. A fost figurat un circuit care acoperă nivelul fizic (PHY Pysical Layer Controller) şi unul care acoperă nivelul de legătură (LLC Link Layer Controller). Unele circuite, aşa cum este TSB43C de la Texas Instruments acoperă mai multe nivele şi încorporează blocuri de prelucrare audio video, un alt motiv pentru complexitatea mai mare a circuitelor.

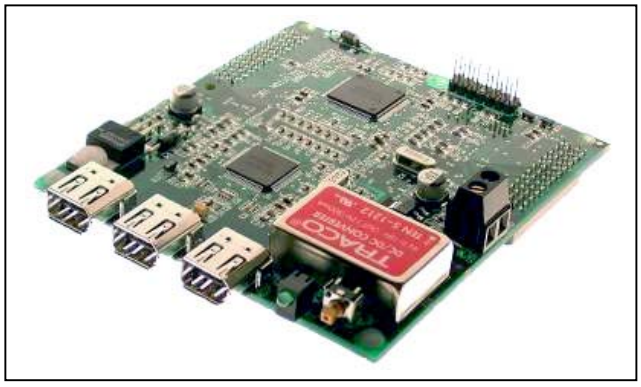

Figura 3.19: Placa IEEE 1394 SedNet

Placa SedNet de la Averna [6] este un sistem de dezvoltare IEEE 1394 cu microcontroller Motorola şi arată ca în figura 3.19.

Acest sistem de dezvoltare este o soluție hardware şi software completă pentru gestionarea unei comunicații IEEE 1394 între aplicația unui client care rulează pe un microcontroller care se conectează cu această placă prin intermediul unor linii de I/O sau o aplicație client care rulează pe microcontrollerul plăcii SedNet. Partea software se numeşte Micro-Stack, rulează pe microcontrollerul plăcii şi realizează nivelele bus-management şi tranzacție din comunicația IEEE 1394. Există şi varianta de a cumpăra sursa programului Micro-Stack şi de a o porta pe un alt microcontroller pentru a dezvolta o soluție hardware proprie. Alimentarea plăcii este realizată cu alimentator propriu sau prin cablul IEEE 1394, dar în al doilea caz acest caz nu este posibilă izolarea galvanică între controllerul de nivel fizic şi restul plăcii. Schema bloc a plăcii SedNet este dată în figura 3.20.

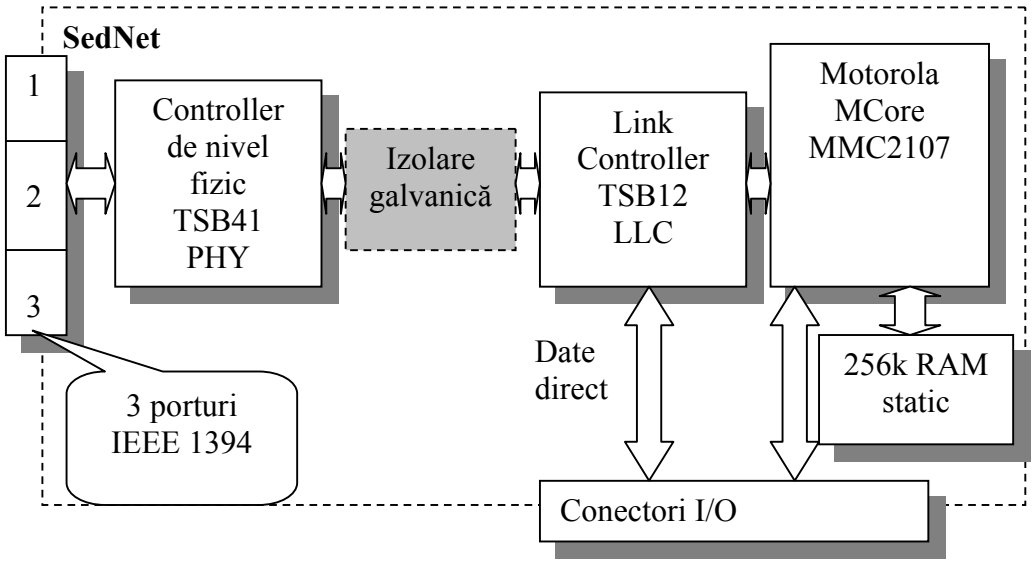

Figura 3.20: Schema bloc a plăcii SedNet

Microcontrollerul MMC2107 are încorporat 128k FLASH iar pentru testare şi debugging portul JTAG este scos la pinii de I/O. Accesul este posibil de la distanţă prin IEEE 1394 la toate resursele plăcii.

Se pot conecta maximum 62 dispozitive IEEE 1394, cu viteze de transfer posibile 100, 200 şi 400Mbps. Placa suportă atât transferuri izocrone cât şi asincrone, datele putând fi preluate direct, fără intermediul microcontrollerului MMC2107 de la controllerul de nivel legătură. Legătura directă este recomandată pentru transferul datelor cu volum mare, cum ar fi cele de la dispozitivele video. Semnalele de legătură cu microcontrollerul sunt cele obişnuite - un port RS232, SPI, JTAG, semnale de întrerupere, reset, tact, linii de I/O etc. Placa a fost special concepută pentru aplicaţii înglobate care nu conţin PC pentru că nu are interfaţă PCI.

3.2.2.Circuite IEEE 1394

Pentru exemplificare au fost alese circuite de la Texas Instruments, cele reprezentate în figura 3.17. Un circuit care acoperă nivelul fizic este **TSB41AB3**, cu 3 porturi IEEE1394. Interfaţa către circuitul care acoperă nivelul de legături este paralelă, pe 2, 4, sau 8 biţi la 49,152MHz, admite decuplare optică şi poate lucra cu circuite alimentate cu 3,3V sau 5V. Capsula este de 80 de pini. Schema bloc a circuitului este dată în figura 3.21.

Cele 3 porturi IEEE 1394 conţin transceivere diferenţiale pentru transmisie şi recepţie şi circuite care monitorizează starea conexiunii. Tactul intern este generat prin conectarea în exterior a unui cristal de 24,576MHz şi multiplicarea cu o buclă PLL pentru obţinerea tactului de 393,216MHz. Toate semnalele de tact necesare, cum ar fi cel de transfer de date paralel de

49,152MHz se obțin prin divizarea tactului de 393,216MHz. Circuitul admite moduri de lucru cu economie de energie. Astfel, în modul Power Down tactul de 393,216MHz este oprit.

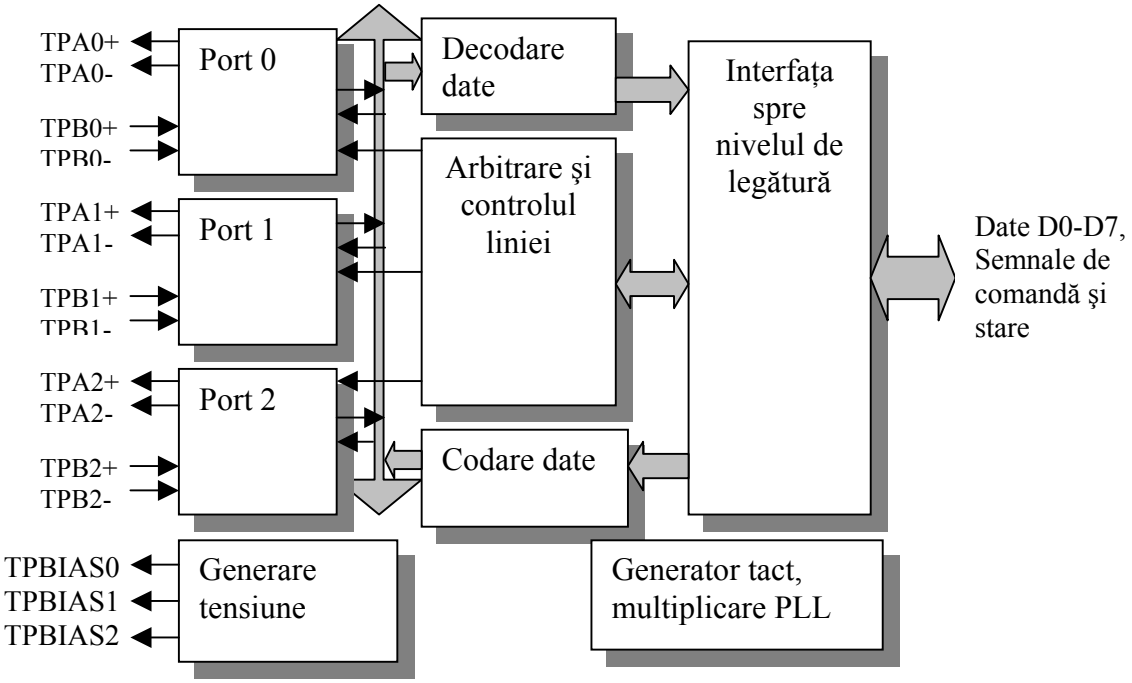

Figura 3.21: Schema bloc a circuitului controller PHY de la TI, TSB41AB3

Circuitul admite lucrul cu izolare galvanică spre controllerul LLC. Pentru aceasta, în cazul validării izolării galvanice furnizează date în mod diferențial spre LLC pentru ca să poată trece printr-un transformator.

Datele sunt transmise între PHY şi LLC în format paralel pe 2, 4 sau 8 biți, funcție de viteza de transfer cerută şi sunt stocate într-un buffer. După codarea datelor ele sunt transmise serial pe linie cu tactul de 392,216MHz, 196,608MHz sau 98,304MHz ceea ce realizează vitezele de transfer S400, S200 sau S100. Datele codate sunt transmise diferențial pe perechea TPA iar tactul pe TPB. La recepție transmițătoarele de linie sunt invalidate şi se face recepția datelor şi decodificarea lor cu aceleaşi variante de tact. Datele sunt refăcute cu ajutorul strob-ului, codificarea fiind D/S, apoi sunt grupate pe 2, 4 sau 8 biți în format paralel, resincronizate şi transmise către LLC cu tactul de 49,152MHz.

Fiecare port IEEE 1394 este prevăzut cu comparatoare a tensiunii comune preluate cu divizor rezistiv cuplat între liniile diferențiale. Informația oferită de comparatorul de pe liniile TPA este folosită în timpul arbitrării pentru stabilirea vitezei următorului pachet de date. Comparatorul de pe liniile TPB stabileşte dacă există o conexiune. Impedanța liniei de

transmisie este de 110Ω şi liniile sunt adaptate la capete cu câte 2 rezistoare serie de câte 56Ω. Tensiunea este de 1,86V şi este generată de PHY pentru fiecare port.

Circuitul LLC **TSB12LV32** este un circuit care acoperă nivelul legătură. Circuitul este încapsulat într-o capsulă de 100 de pini. Schema bloc a circuitului este dată în figura 3.22.

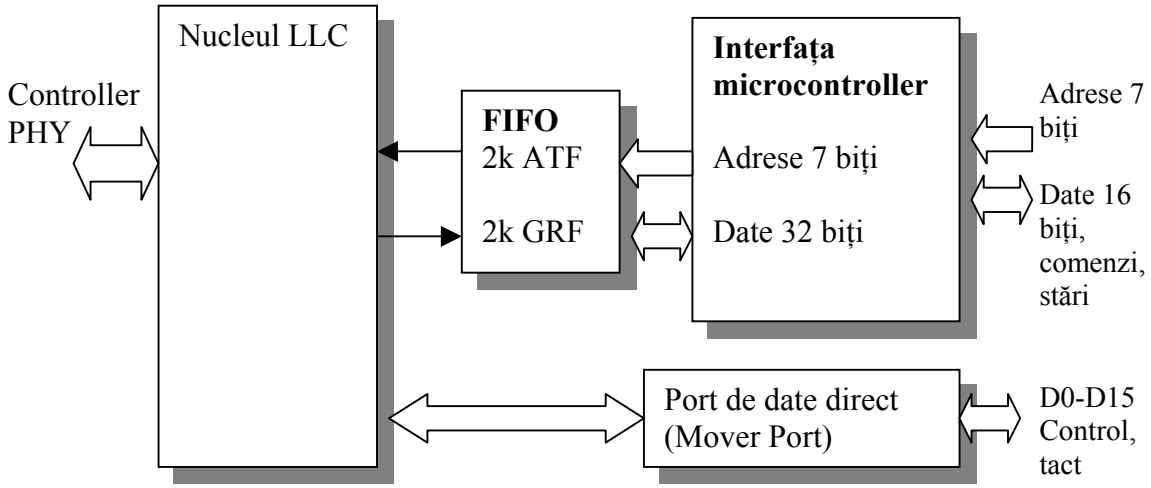

Figura 3.22: Schema bloc a circuitului controller LLC de la TI, TSB12LV32

Circuitul LLC asigură transmisia şi recepția cadrelor de date la viteza de maximum 400Mbps prin intermediul celor 2 FIFO de 2kocteți. Circuitul formează /descompune cadrele, calculează şi ataşează CRC-ul. Circuitul poate administra magistrala (bus manager) şi poate conduce transferuri izocrone (Isochronous Resource Manager IRM). Interfața cu microcontrollerul este paralelă pe 8/16 biți la frecvența de maximum 60MHz. Cele 2 FIFO de 2k la recepție (General Receive FIFO GRF) şi transmisie (Asynchronous Transmit FIFO ATF) sunt accesibili de către microcontroller. Portul direct de date (Data Mover Port DM) poate recepționa şi transmite date izocrone, asincrone şi streaming cu tact de maximum 25MHz şi la port se conectează de regulă o memorie de capacitate mare.

Ca şi microcontroller gazdă se pot folosi microcontrollere din familia Motorola 68000 sau Freescale ColdFire™ fără hardware suplimentar. Microcontrollerele şi microprocesoarele Freescale folosesc o arhitectură particulară a magistralei externe, numită FlexBus cu varianta Mini-FlexBus [7], pentru conectarea în exterior a memoriei sau altor circuite, asemănătoare cu magistrala prezentată în capitolul 2.

Ultimul circuit amintit aici este **TSB43CA43A**, care include atât controllerul PHY cât şi cel LLC, precum şi funcții audio video. Circuitul se bazează pe un nucleu ARM7, are 176 pini şi este destinat ca soluție *single chip* pentru interfațarea dispozitivelor audio video prin IEEE 1394. Cele 3 porturi IEEE 1394 care echipează circuitul pot asigura o rată de transfer de maximum 400Mbps. Circuitul are schema bloc din figura 3.23.

Blocul audio video se interfațează cu sursa de semnal prin două canale seriale HSDI (High Speed Data Interface) prin care pot circula date standard MPEG2-DVB (Digital Video Broadcasting, care este un mod de transmisie digital compresat al semnalului video într-un şir MPEG, datele fiind modulate OFDM, Orthogonal Frequency Division Multiplexing). Alte standarde admise sunt DV (Digital Video), DSS (Direct Satellite System, sistem proprietar al DirecTV, SUA, cu codare QPSK şi cadre de 127 octeți), audio DAC etc.

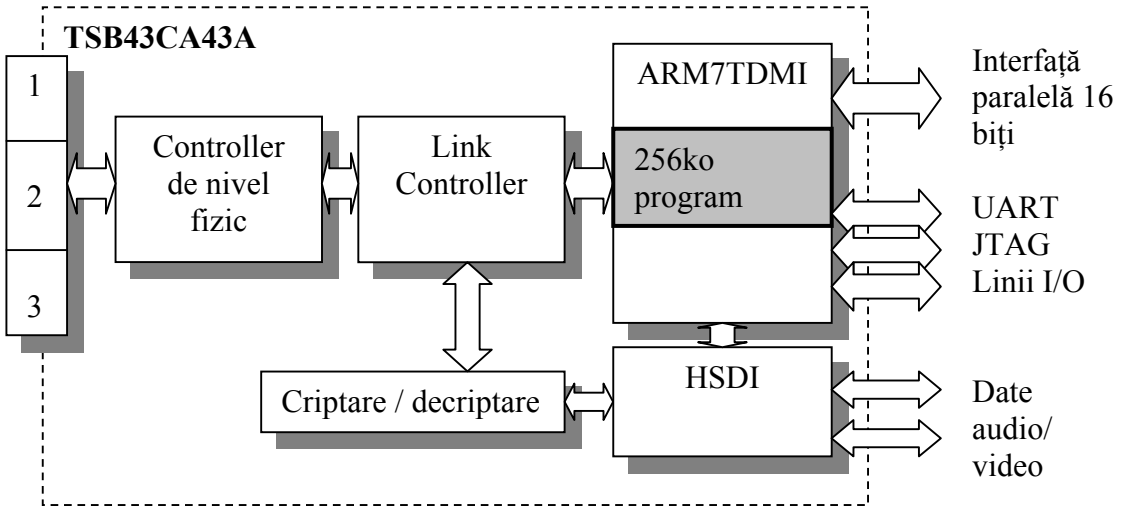

Figura 3.23: Schema bloc a circuitului TSB43CA43A

În 1998 5 firme (Hitachi, Intel, Matsushita, Sony şi Toshiba) au creat o asociere pentru a asigura protecția datelor audio video transmise prin cablu, protecție numită DTLA Digital Transmission Content Protection. Circuitul TSB43 conține un bloc DTLA (Digital Transmission Licensing Administrator) care asigură criptarea /decriptarea datelor cu ajutorul unui cifru realizat de Hitachi numit M6, prevăzut pentru fiecare canal HSDI. Calculele matematice sunt efectuate pe 160 biți.

Unitatea centrală este ARM7 la 50MHz cu mod de funcționare pe 32 de biți şi 16 biți, cu 256kocteți memorie de program şi suport JTAG pentru test şi punere la punct. Interfața cu exteriorul este printr-o interfață paralelă de 16 biți în mod sincron sau asincron la care se poate conecta o memorie externă sau un alt circuit. Un port UART asigură transferul de date la viteze mici. 11 linii de I/O de uz general completează posibilitățile de conectare a circuitului cu exteriorul.

Bibliografie

[1] Filatova A., *USB 3.0 Poised for Success*, 2009, [online], http://exectweets.com/2009/11/30/superspeed-usb-poised-for-success/

[2] www.ftdichip.com, [online]

[3] Stedman I., *PIC USB Interface*, [online], http://www.ianstedman.co.uk/Projects/PIC_USB_Interface/pic_usb_interface.html

[4] Gerigan C.**,** Ogrutan P., *USB Controller- Educational aspects*, The 10 International Conference on Optimization of Electrical and Electronic Equipment, Optim 2006, ISBN 973-635-702-3, Brasov

[5] *Teensy++ USB Development Board*, [online]

[6] http://www.averna.com/en/products/ieee1394/hardware.php

[7] Lobdell M., *Using the Mini-FlexBus External Bus Interface for ColdFire Microcontrollers* AN3854/2009, Freescale Semiconductor