

1. MAGISTRALA SERIALĂ UNIVERSALĂ - USB (UNIVERSAL SERIAL BUS)

obligatoriu

1.1. DESCRIERE GENERALĂ

Magistrala USB reprezintă soluția oferită comunicațiilor seriale de noua generație de calculatoare PC. Este o interfață serială rapidă, bidirecțională, ieftină și ușor de folosit. USB a fost creată ca un standard industrial, o extensie a arhitecturii PC orientată spre armonizarea cu standardele de comunicație din telefonie, ceea ce este numit CTI (Computer Telephony Integration). Acest aspect este considerat fundamental din punct de vedere al aplicațiilor generației următoare.

Avantajele acestei soluții față de bătrâna interfață serială RS-232 transformată prin îmbunătățiri în EIA/TIA-232-E sunt:

- rata de transfer - poate atinge 12 Mbps față de 115 000 bps;
- conectează până la 127 de dispozitive la PC, (ceea ce înseamnă că operează ca o magistrală) față de numai 2 dispozitive;
- ușor de utilizat de către utilizatorul final (end user) - adăugarea/eliminarea de dispozitive în/din sistem este foarte comodă;
- are un protocol flexibil;
- este o soluție ieftină de interconectare.

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.

USB este o magistrală pe cablu care permite schimb de date între un calculator gazdă și o gamă largă de periferice accesibile simultan. Magistrala permite ca perifericul să fie atașat, configurat, folosit și deconectat în timp ce gazda și celelalte periferice operează.

USB a fost proiectată în primul rând pentru utilizatorii care nu doresc să intre în detalii de instalare hardware, astfel sistemul complicat de cablare a fost înlocuit cu un control software. Toate problemele presupuse de interconectarea mai multor dispozitive cu performanțe și rate de transfer diferite sunt tratate prin software.

Magistrala USB definește trei categorii de dispozitive fizice:

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

Acestea sunt interconectate într-o topologie specifică de tip stea multiplă. Topologia USB este reprezentată sugestiv în Fig. 1.1. Din figură se poate observa că USB face legătura între dispozitivele USB și gazdă. În nodul fiecărei stele se găsește un hub. Între elemente legătura este făcută cu segmente de cablu: fiecare segment de cablu face legătura punct la

punct (point-to-point) între gazdă și o funcție sau un hub sau între hub și o funcție sau un alt hub.

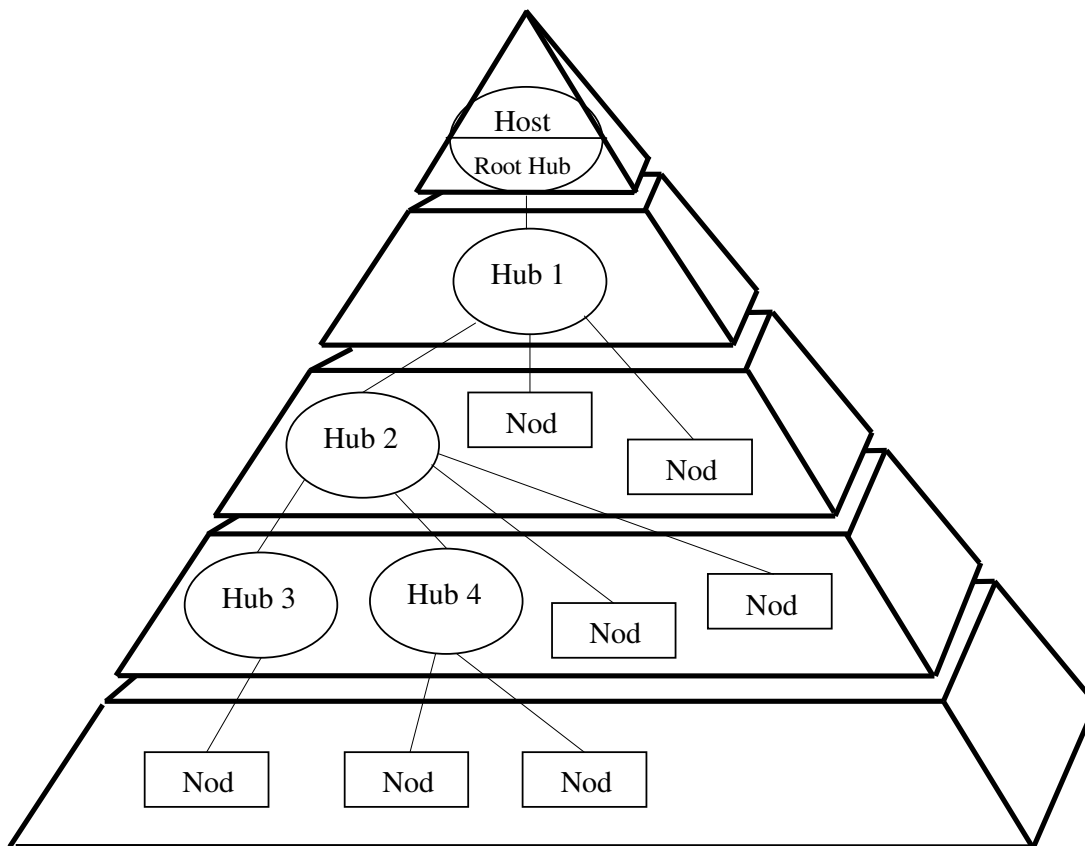


Fig. 1.1. Topologia magistralei USB

În sistem există o singură gazdă. Interfața dintre gazdă și sistemul USB se numește Host Controller; poate fi o combinație hardware, software sau firmware. Gazda are implicit un hub care se numește Root Hub și pune la dispoziție unul sau mai multe puncte de legătură. Punctele de legătură se numesc porturi.

Hub-urile și funcțiile USB sunt referite global ca dispozitive USB. Un hub furnizează puncte de legătură suplimentare, iar o funcție furnizează facilități suplimentare sistemului.

Dispozitivele USB recunosc protocolul USB, răspund la operații standard și recunosc informația descriptivă din informația vehiculată în magistrală. Dispozitivele și gazda USB vor fi descrise mai amănunțit în unul din subcapitolele următoare.

În ceea ce privește inițiatorul transferurilor de date pe magistrală, acestea este doar gazda. Protocolul folosit este protocol prin interogare (de tip polled). Datele vehiculate pe magistrală sunt grupate în pachete; 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 autoselectează 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. Descrierea în amănunt a aspectelor concrete legate de protocol este reluată în unul din capitolele următoare.

Aspectele electrice și mecanice ale interfeței sunt reglementate foarte precis în specificațiile de magistrală. Semnalele electrice sunt vehiculate sub formă de semnal diferențial (D+ și D-) și sunt posibile două rate de transfer: 12 Mbps (full-speed) și 1.5 Mbps (low-speed). Ambele rate de transfer sunt suportate în aceeași magistrală prin comutare dinamică automată între moduri. Semnalul de ceas (clock) este transmis codificat simultan cu datele diferențiale. Codificarea utilizată este NRZI cu împănare de biți (*bit-stuffing*).

Transportarea semnalelor se face printr-un cablu cu patru fire conductoare. Semnalul util este transportat pe două conductoare torsiadate. Pe celelalte două conductoare cablul mai transportă tensiunea de alimentare nominală de +5V (V_{BUS}) și potențialul de referință (GND). În acest fel pot fi alimentate prin cablul de legătură dispozitive conectate la magistrală. În Fig. 1.2. este reprezentat un cablu USB.

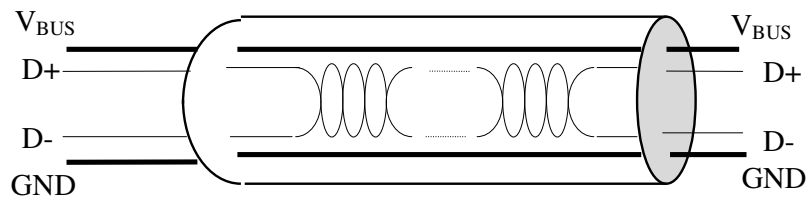


Fig. 1.2. Cablu USB

Toate dispozitivele au o conexiune în sus, amonte (*upstream*) și una sau mai multe conexiuni în jos, aval (*downstream*). Conectorii pentru aceste legături nu sunt interschimbabili, eliminând posibilitatea de buclare la hub.

Alimentarea dispozitivelor este o problemă de magistrală la USB. Gazda poate furniza alimentare dispozitivelor USB conectate direct. Din punctul de vedere al alimentării, dispozitivele pot fi dispozitive care dispun de sursă proprie de alimentare (*self powered device*) sau dispozitive care se alimentează prin cablu (*bus powered device*). Un hub poate, ca și gazda, să furnizeze alimentare dispozitivelor atașate.

Un atu al USB este siguranța crescută a transferurilor de date, se poate spune că este o magistrală robustă. La asigurarea acestei calități își dau concursul o serie de factori: principii funcționale, accesorii implementate în acest scop, restricții constructive bine definite. Acesta este principalul motiv pentru care semnalele purtătoare de informație sunt semnale electrice diferențiale. Pentru a garanta securitatea datelor este implementat un aparat CRC (soft, hard sau combinat). În caz de eroare sistemul hotărăște retransmiterea de trei ori a pachetului alterat, după care este informat utilizatorul care decide asupra modului de acțiune.

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. După aceea constată dacă dispozitivul este o funcție sau un hub. La decuplare hub-ul dezactivează portul și indică gazdei acest eveniment care este tratat în continuare de soft-ul gazdei. Pentru a se adapta dinamic, sistemul software USB este permanent într-un proces de inventariere a magistralei (*bus counting*).

obligatoriu

1.2. ELEMENTE ARHITECTURALE

1.2.1. Tipuri de transferuri USB

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.

1.2.2. Dispozitive USB

Dispozitivul USB este numit în specificații device. Toate dispozitivele sunt accesate printr-o adresă USB care se atribuie la atașarea acestuia în sistem. Fiecare dispozitiv USB poate dispune de una sau mai multe pipe-uri prin care gazda comunică cu dispozitivul. Toate dispozitivele posedă un pipe special, *endpoint zero*, care este privit ca pipe de control. Mecanismul de accesare a informației prin acest port este comun tuturor dispozitivelor. Pipe-ului endpoint zero îi este asociată informația ce descrie complet dispozitivul USB:

- informație standard - clasa de dispozitiv, informații de power management, producător;
- informație de clasă - semnificația depinde de clasa de dispozitive;
- informații de producător - la alegerea producătorului.

Există două clase de dispozitive: hub (distribuitor) și funcție.

1.2.2.1. Dispozitive HUB

Dispozitivele hub sunt elemente cheie în arhitectura plug-and-play a USB. Fig. 1.3. arată un hub tipic .

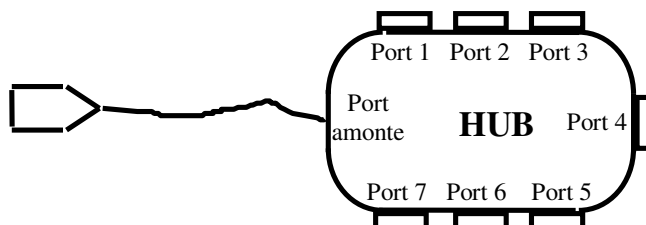


Fig 1.3 Hub tipic

Hub-urile sunt folosite pentru a simplifica conectica din punctul de vedere al utilizatorului și totodată determină un sistem robust și ieftin.

Punctele de atașare se numesc porturi. Există un port spre gazdă (amonte - upstream port) și mai multe porturi spre alte dispozitive (aval - downstream port).

Un hub are două componente: un repetor (*Hub Repeater*) și un controler (*Hub Controller*). Repetorul este un comutator comandat prin protocol, care face legătura între portul amonte cu unul din porturile aval. Controlerul conține registre de interfață care fac posibilă comunicarea cu gazda, pentru configurare și comandă.

1.2.2.2. Dispozitive funcții

Funcția este un dispozitiv USB capabil să transmită informații de date sau control prin magistrală. În mod obișnuit, o funcție este un periferic independent cu un cablu care se atașează la un port al unui hub. Este posibil ca o unitate fizică să includă mai multe funcții și un hub, dispunând de un singur cablu de conexiune. Acesta se numește dispozitiv compus și este văzut de gazdă ca un hub cu mai multe dispozitive nedetașabile. Fiecare funcție posedă informație de configurare prin care sunt descrise pentru gazdă facilitățile și resursele necesare.

Exemple de funcții de dispozitiv:

- dispozitiv de localizare - mouse, tabletă digitizoare, light pen;
- dispozitiv de intrare - tastatura;
- dispozitiv de ieșire - imprimanta;
- adaptor de telefonie - ISDN.

Fig. 1.4. prezintă un sistem USB la un calculator PC model desktop.

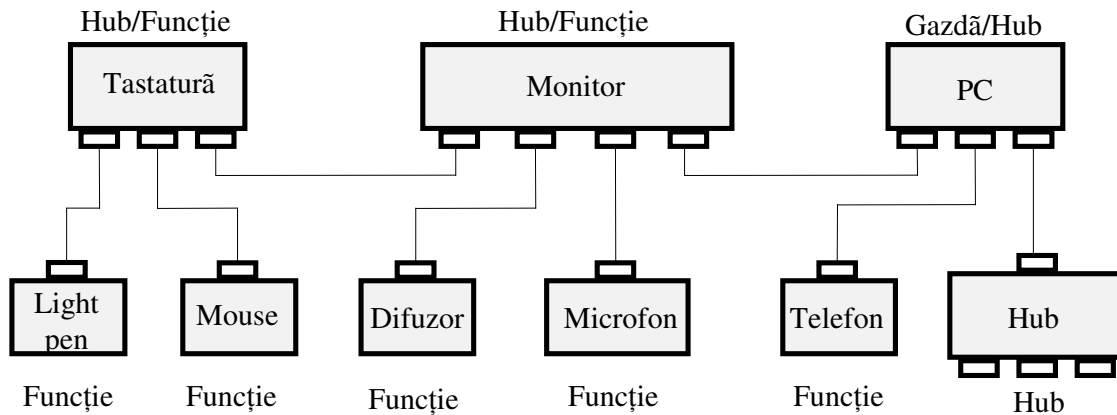


Fig. 1.4. Sistem USB la un calculator PC model desktop

1.2.3. Gazda USB

Gazda ocupă o poziție specială în coordonarea sistemului USB. Pe lângă poziția fizică specială, gazda are responsabilități speciale în ceea ce privește magistrala și dispozitivele atașate. Controlează accesul la magistrală; un dispozitiv are acces la magistrală doar dacă îi este acordat de gazdă. Gazda este responsabilă pentru următoarele acțiuni:

- detectarea conectării/deconectării de dispozitive USB;
- administrează transferurile de control între gazdă și dispozitive USB;
- administrează transferurile de date între gazdă și dispozitive USB;
- culege informații de stare și statistici de activitate;
- furnizează alimentare dispozitivelor USB atașate.

Gazda interacționează cu dispozitivele USB prin controlerul gazdei. Există de asemenea o interacțiune între software-ul de sistem și software-ul de dispozitiv pe următoarele cinci domenii:

- numărare și configurare de dispozitive;
- transferuri izocrone;
- transferuri asincrone de date;
- administrarea alimentării (power management);
- informații de administrare de magistrală și de dispozitiv.

1.2.4. Extensii arhitecturale

Arhitectura USB este extensibilă la nivelul interfeței dintre driver-ul Host Controller-ului și driver-ul USB. În viitor este posibil să se dezvolte arhitecturi cu mai multe Host Controller-e și cu drivere asociate.

optional

1.3. FLUXUL DE DATE ÎN MAGISTRALA USB

1.3.1. Reprezentarea elementelor funcționale

Un mod simplu de a vedea legătura între gazdă și dispozitive este reprezentat în Fig. 1.5. O abordare mai profundă este reprezentată în figura următoare, Fig. 1.6.

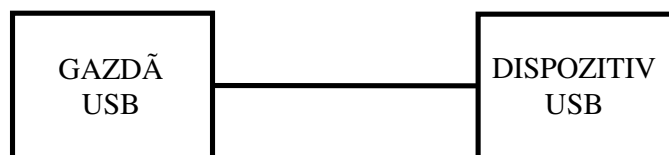


Fig. 1.5. Reprezentare simplă a legăturii gazdă-dispozitiv

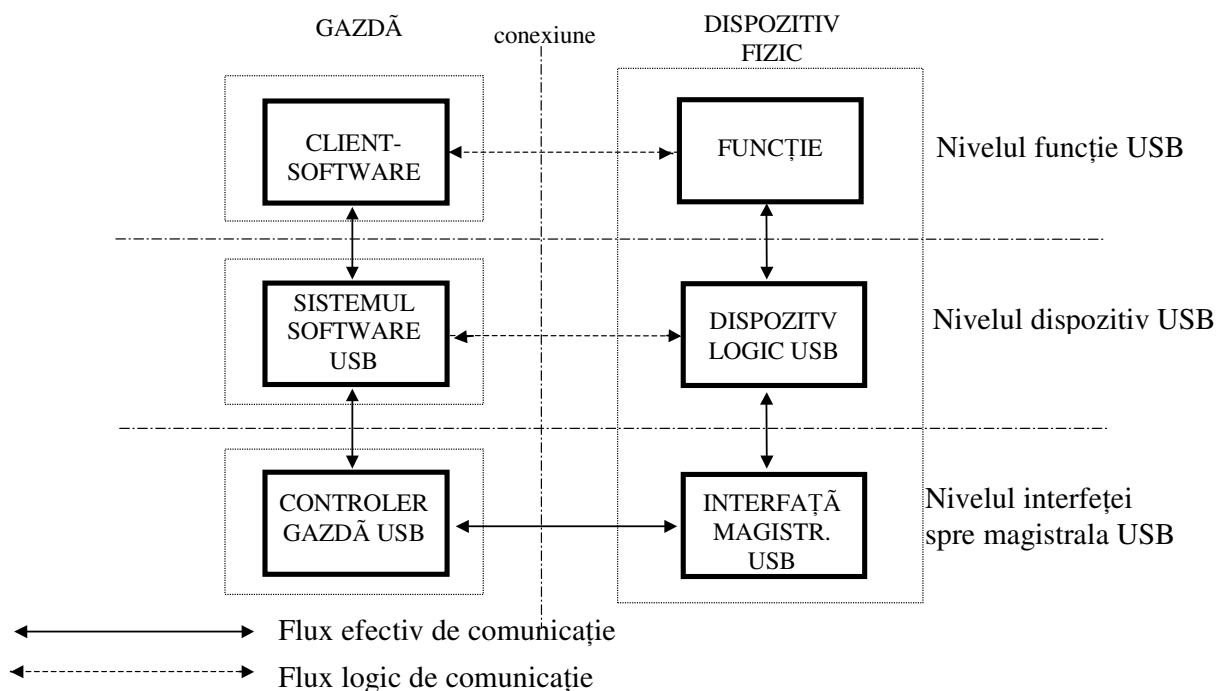


Fig.1.6. Legătură gazdă-dispozitiv

Dispozitivul fizic USB este o componentă hardware aflată la capătul unui cablu, capabilă să execute funcții utile.

Componenta client-software este reprezentată de software-ul care rulează pe gazdă; fiecărui dispozitiv îi corespunde o componentă client-software care se regăsește la gazdă. Aceasta se livrează fie cu sistemul de operare, fie odată cu dispozitivul USB.

Componenta sistem-software-USB este software-ul care suportă magistrala USB într-un sistem de operare particular. Această componentă se livrează cu sistemul de operare și este independentă de dispozitivul USB particular sau de componenta client-software.

Componenta controler-gază-USB (USB-Host Controller) reprezintă interfața spre magistrală a controlerului și este constituită din hardware+software care permit dispozitivelor să se atașeze la gazdă.

Responsabilitățile și drepturile sunt împărțite între cele patru componente descrise. O simplă conexiune gazdă-dispozitiv presupune o interacțiune pe mai multe nivele între mai multe entități.

- Nivelul interfeței spre magistrala USB furnizează conectarea fizică a semnalelor între gazdă și dispozitiv;
- Nivelul dispozitiv USB reprezintă modelul care execută operații generice între software-sistem și dispozitiv;
- Nivelul funcție USB reprezintă modelul care conferă facilități suplimentare gazdei prin entitatea client-software adecvată dispozitivului;
- Nivelele dispozitiv USB și funcție USB modelează o comunicație logică, comunicație ce se realizează efectiv fizic prin nivelul interfeței spre magistrala USB.

1.3.2. Topologia fizică a magistralei

Dispozitivele sunt conectate la gazdă printr-o rețea de tip stea (multiplă) ca în Fig. 1.1.

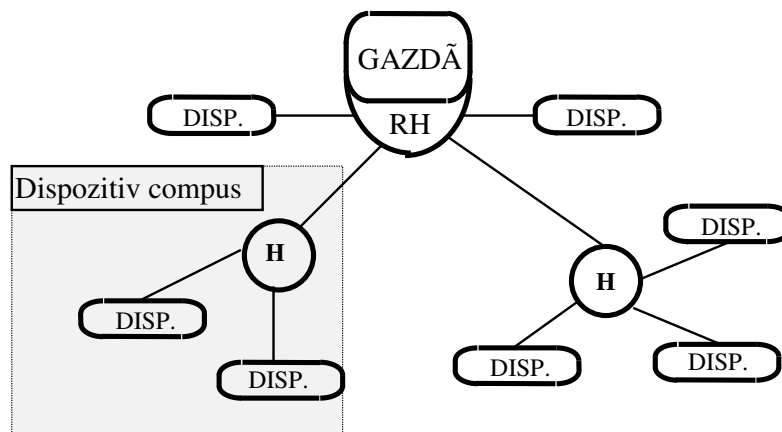


Fig. 1.7. Topologia magistralei USB

Punctele de atașare sunt puse la dispoziție prin clasa de dispozitive numite hub (distribuitor). Punctele de atașare sunt porturile. Gazda are un hub inclus, numit root-hub (RH în figură) prin care furnizează unul sau mai multe puncte de legătură. Dispozitivele care conferă gazdei facilități suplimentare se numesc funcții. Într-un dispozitiv fizic pot fi înglobate mai multe funcții. Unitatea fizică conține în acest caz un hub la care sunt atașate permanent în configurație fixă funcțiile individuale; acesta este un dispozitiv compus.

1.3.3. Topologia logică a magistralei

În timp ce conexiunea fizică este o stea multiplă, gazda comunică cu fiecare dispozitiv logic ca și cum ar fi direct conectat la hub-ul rădăcină (root hub). Modelul logic este ilustrat în Fig. 1.8.

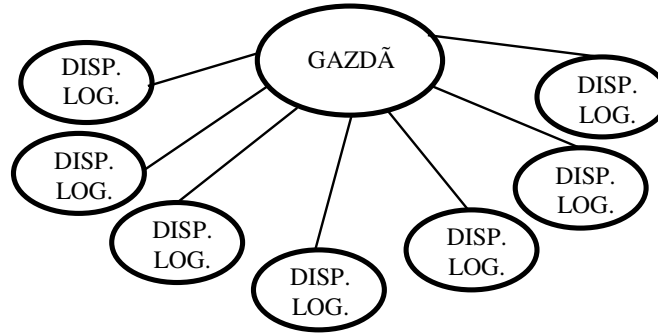


Fig. 1.8. Topologia logică a magistralei USB

1.3.4. Relația între client-software și funcție

Deși atât topologia fizică cât și topologia logică reflectă caracterul partajat al magistralei USB, la activarea unei funcții, imaginea interacțiunii dintre componenta client-software și funcție este ca și cum nu ar exista alte elemente în sistem. Client-software interacționează doar cu interfața de interes. Modelul de interacțiune este redat în Fig. 1.9.

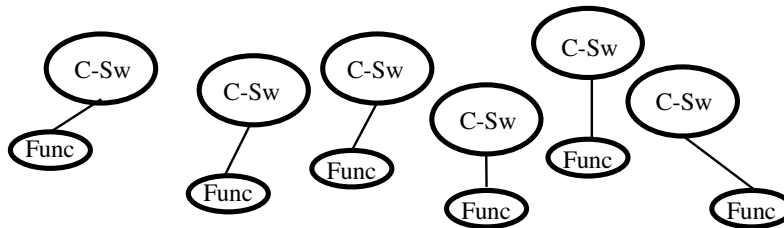


Fig. 1.9. Relația între client-software și funcție

obligatoriu

1.4. INTERFAȚA ELECTRICĂ

1.4.1. Caracteristicile circuitelor emițătoare

USB folosește circuite diferențiale pentru elaborarea semnalului electric emis pe cablul USB. Nivele de tensiune garantate de aceste circuite sunt cele din Tabelul 1.1.

Tabel 1.1. Nivele de tensiune garantate de circuitele emițătoare pe cablu USB

Parametru	minim	maxim	unitate de măsură
V _{OL}	0.0	0.3	V
V _{OH}	2.8	3.6	V

Ieșirile circuitelor emițătoare de magistrală trebuie să fie de tip three-state pentru a permite operații bidirecționale half-duplex.

Structura unui circuit emițător de magistrală este reprezentată în Fig. 1.10

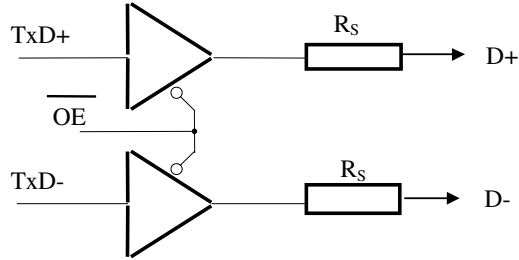


Fig.1.10. Circuit emițător de magistrală

Rezistența R_S este inclusă în structura emițătorului. Rezistența echivalentă de ieșire a emițătorului trebuie să fie între 28Ω și 44Ω .

Specificațiile de magistrală prevăd restricții și pentru timpii de creștere și descreștere ai semnalului de date. Acesta se măsoară între 10% și 90% din excursia semnalului care poate să ia valori între $-1.0V$ și $+4.6V$. Figura 1.11. definește timpii de creștere (rise time) și descreștere (fall time).

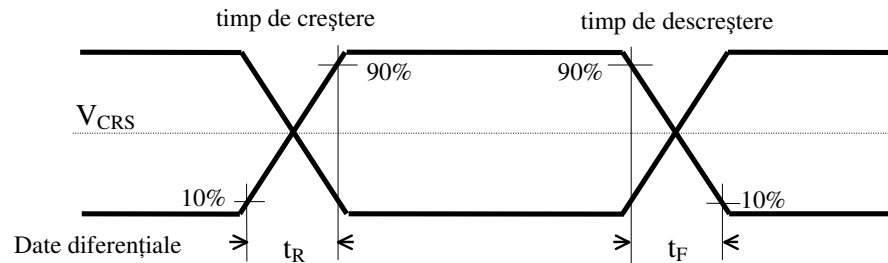


Fig. 1.11. Timpii de creștere și descreștere ai semnalului de date

V_{CRS} , potențialul de intersecție al semnalului diferențial (Crossover voltage), poate lua valori între $1.3V$ și $2.0V$.

1.4.2. Caracteristicile circuitelor receptoare

Receptorul de linie este un circuit cu intrare diferențială. Nivelele de tensiune acceptate la intrare sunt cele din Tabelul 1.2.

Tabel 1.2. Nivele de tensiune acceptate de circuitele receptoare USB

Parametru	minim	maxim	unitate de măsură
V_{IH}	2.0	-	V
V_{IL}	-	0.8	V
V_{DI}	0.2	-	V
V_{CM}	0.8	2.5	V

V_{DI} este sensibilitatea diferențială la intrare, iar V_{CM} este tensiunea de mod comun a intrărilor diferențiale. Atât intrarea D+ cât și intrarea D- pot fi temporar mai mici decât V_{IH} minim în timpul tranzițiilor semnalelor diferențiale; intervalul nu are voie să depășească 14ns la transferul de mare viteză (12 Mbps) și 210 ns la transferul de mică viteză (1.5 Mbps).

obligatoriu

1.5. INTERFAȚA MECANICĂ

Topologia USB se realizează conectând aval de hub-ul gazdei un port amonte al unui hub sau o funcție, folosind o conecție definită și cabluri de asemenea cu caracteristici stabilite prin specificațiile mecanice ale USB. Deoarece USB poate opera la două viteze diferite, pentru fiecare din ele se fac precizări pentru caracteristicile cablului.

Cablul conține patru fire conductoare, două purtătoare ale semnalului diferențial, unul transportă tensiunea de alimentare și un conductor pentru potențialul de masă. Această structură este valabilă pentru ambele viteze posibile prin USB.

Pentru viteze mari, 12 Mbps, conductoarele care transportă semnalul diferențial sunt torsadate și toate patru conductoarele sunt ecranate.

O secțiune printr-un astfel de cablu este redată în Fig. 1.12.

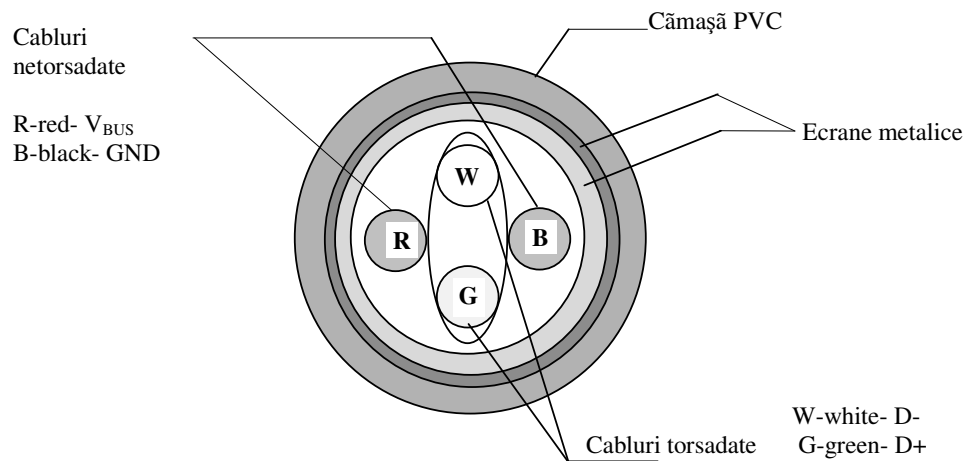


Fig. 1.12. Structura constructivă a cablului USB

La viteze scăzute, 1.5 Mbps, cerințele pentru cablu nu sunt atât de dure; nu este necesar ecranul și nici torsadarea conductoarelor.

Cablul USB trebuie marcat de către producător. Pentru cablul de mică viteză nu este obligatorie folosirea unui cablu marcat. Cablurile de mare viteză pot fi folosite și la conexiunile cu transferuri de mică viteză.

Deși restricțiile sunt dure pentru cabluri, lungimea acestora nu poate depăși 5m. Datorită topologiei sistemului USB, dispozitivele se pot afla practic la distanțe mai mari de 5m de gazdă sau între ele.

În ceea ce privește conectorii, aceștia sunt concepuți în două serii, *A* și *B*. Seria *A* pentru conexiuni amonte și seria *B* pentru conexiuni aval. Specificațiile USB dau o întreagă listă a standardelor prin care se reglementează caracteristicile materialelor folosite. Dintre aceste documente amintim:

- American National Standard/Electronic Industries Association ANSI/EIA-364-C (12/94)
- American Standard Test Materials ASTM-D-4565 și ASTM-D-4566
- Underwriters Laboratory, Inc UL STD-94 și UL Subject-444

obligatoriu

1.6. PROTOCOLUL USB

1.6.1. Descriere generală

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. Majoritatea sistemelor folosesc cadre de 4 până la 8 biți de date.

La transmisii seriale 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. În diagramele ce urmează, biții sunt reprezentați de la stânga la dreapta în ordinea în care sunt trimiși pe magistrala serială. 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)

Toate pachetele conțin la începutul lor un câmp de sincronizare, la care se va face referire în continuare cu numele SYNC, și un câmp identificator de pachet, la care se va face referire în continuare cu PID (Packet Identifier).

1.6.1.1. Câmpul SYNC

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. Aceste tranziții permit tuturor dispozitivelor conectate la USB să-și reseteze ceasul și să se sincronizeze cu gazda. 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. Succesiunea datelor și forma semnalului din câmpul SYNC sunt reprezentate în Fig. 1. 13.

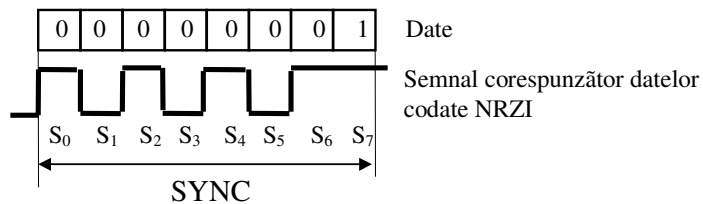


Fig. 1.13. Câmpul SYNC

Ultimii doi biți din câmpul SYNC sunt folosiți pentru a indica sfârșitul câmpului și totodată începutul următorului câmp, PID.

1.6.1.2. Câmpul PID

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. Structura câmpului PID este reprezentată în Fig. 1.14.

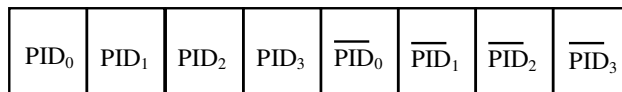


Fig. 1.14. Câmpul PID

Primii patru biți responsabili cu definirea tipului pachetului sunt folosiți în două etape. Cei mai semnificativi doi biți specifică tipul pachetului, iar ceilalți doi biți împart pachetele în categorii. Tabelul 1.3. redă regula de interpretare a informației din PID.

Tabel 1.3. Semnificația biților din câmpul PID pentru stabilirea tipului de pachet

Cod	Tip pachet
xx00xx11	Pachet special
xx01xx10	Pachet de semnalizare
xx10xx01	Pachet de dialog
xx11xx00	Pachet de date

1.6.2. 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. Structura pachetului este reprezentată în Fig. 1.15.

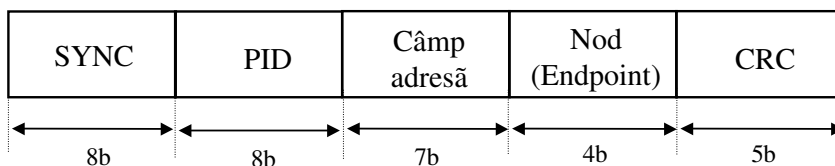


Fig. 1.15. Structura pachetului de semnalizare

Primele două câmpuri, fiecare cu lungimea de un octet, iau forma standard a tuturor pachetelor USB.

În cadrul câmpului PID se definesc pentru acest tip de pachet patru categorii de pachete de semnalizare conform Tabelului 1.4.

Tabel 1.4. Semnificația PID pentru stabilirea categoriei pachetului de semnalizare

PID	Categorie pachet semnalizare
00011110	Pachet de ieșire (OUT)
01011010	Pachet de început cadru (SOF)
10010110	Pachet de intrare (IN)
11010010	Pachet de comandă (SETUP)

Pachetul de ieșire (OUT) poartă datele de la gazdă la dispozitiv.

Pachetul de intrare (IN) poartă datele de la dispozitiv la gazdă.

Pachetul de comandă (SETUP) vizează un anume nod (Endpoint).

Pachetul de început de cadru (SOF-Start Of Frame) este difuzat tuturor dispozitivelor; structura acestui pachet diferă de a celorlalte trei pachete de semnalizare.

Pentru pachetele IN, OUT și SETUP, următorii 7 biți după PID sunt interpretați ca și câmp de adresă pentru a identifica dispozitivul pe care gazda vrea să-l apeleze pentru comandă sau transfer de date. Următorii 4 biți furnizează un număr de nod (Endpoint).

Un nod reprezintă o secțiune a unei funcții USB adresabilă individual. Conceptul de nod dă proiectanților de hardware posibilitatea divizării unui dispozitiv fizic în unități logice separate.

Ultimul câmp, de 5 biți, este folosit pentru verificări CRC, asigurând astfel integritatea transferului pachetului de date. În suma de control sunt incluse toate câmpurile în afară de PID, care este protejat prin structura sa.

1.6.2.1. Pachetul de început de cadru

Gazda emite un pachet de început de cadru la 1.00 ms în cadrul unei tranzații de semnalizare. Toate dispozitivele reacționează și decodifică acest pachet, dar nu trimit înapoi semnal de recepție a pachetului. Structura pachetului de început de cadru este cea din Fig. 1.16.

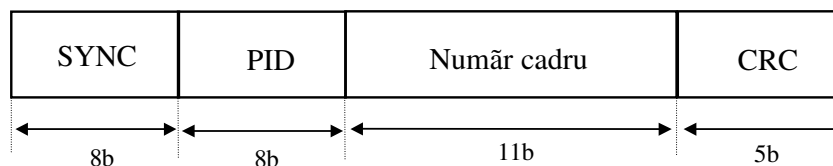


Fig.1.16. Structura pachetului de început de cadru

Câmpul de 11 biți conține numărul cadrului care este atribuit de gazdă în mod crescător de la 0 la 7FFH (2047), după care începe din nou de la 0. Câmpul este folosit ca informație de sincronizare pe magistrala USB.

1.6.3. Pachetul de date

Informația propriu-zisă este transferată în sistemele USB sub forma unor pachete de date. Structura acestui pachet este redată în Fig. 1.17.

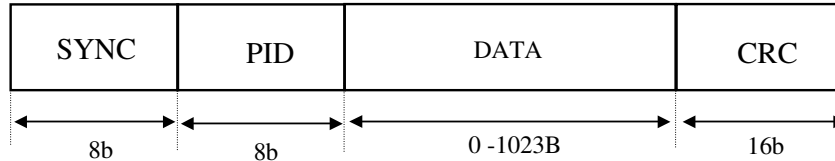


Fig.1.17. Structura pachetului de DATE

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 de 16b aflat la sfârșitul pachetului.

În câmpul PID se definesc două categorii de câmpuri de date; DATA0 și DATA1 conform Tabelului 1.5.

Tabel 1.5. Semnificația PID pentru stabilirea categoriei pachetului de date

PID	Categoria pachet date
00111100	DATA0
10110100	DATA1

Din punct de vedere funcțional, cele două categorii de pachete de date formează între emițător și receptor un sistem adițional de verificare a erorilor. Emițătorul oscilează între DATA0 și DATA1 pentru a indica faptul că a recepționat o confirmare validă a recepției pachetului precedent.

Exemplu:

Emițătorul trimite un pachet DATA0; dacă receptorul a preluat cu succes datele, emite un pachet handshake (de dialog) prin care confirmă emițătorului că a preluat corect datele. În urma interpretării pachetului handshake, emițătorul trimite următorul pachet de tip DATA1, ceea ce indică receptorului că mesajul său de confirmare (ACK) a fost interpretat corect de emițător.

1.6.4. Pachetul handshake

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 Fig. 1.18.

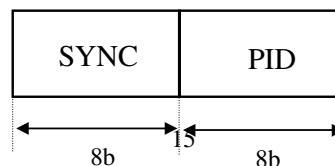


Fig. 1.18. Structura pachetului handshake

Câmpul PID definește trei categorii de pachete handshake conform Tabelului 1.6.

Tabel 1.6 Semnificația PID pentru stabilirea categoriei pachetului handshake

PID	Categoria pachet handshake
00101101	ACK
10100101	NAK
11100001	STALL

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.

obligatoriu

2. Controllere USB din familia FTDI [4]

Din introducerea anterioara reiese complexitatea USB si necesitatea folosirii unor circuite specializate. Se pot folosi interfete de conversie USB paralele sau serial [4] sau microcontrollere cu USB [5]. In acest material se opteaza pentru prima varianta.

2.1. Circuitul interfata USB - format paralel, FT245BM

Circuitele cu sufixul AM sunt capabile USB 1.1 iar cele cu sufixul BM sunt capabile USB 2.0. Protocolul USB este incorporat total in circuit si nu este nevoie de programarea formarii si gestionarii cadrelor USB. Schema bloc a circuitului este data in figura 1:

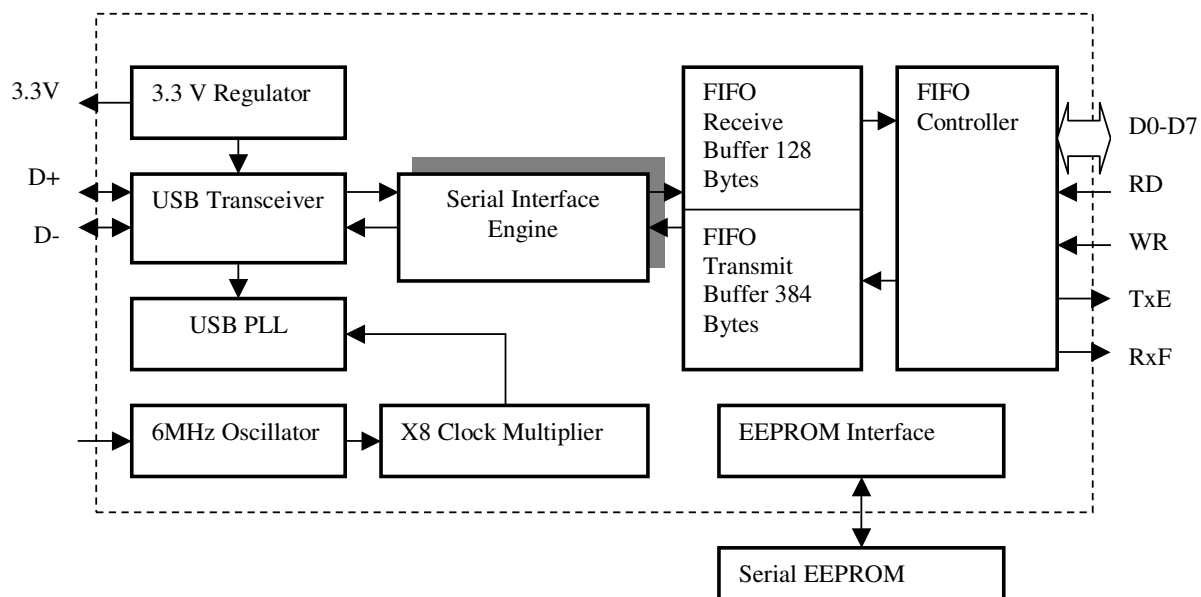


Figura 1: Schema bloc a interfeței USB - format paralel FT 245BM

Pentru a putea realiza o lucrare practica cu acest tip de interfata, anume obtinerea unor rezultate spectaculoase, la circuitul FTDI se conecteaza un grup de 8 LED-uri si un grup de 8 intreruptoare, ca in figura 2:

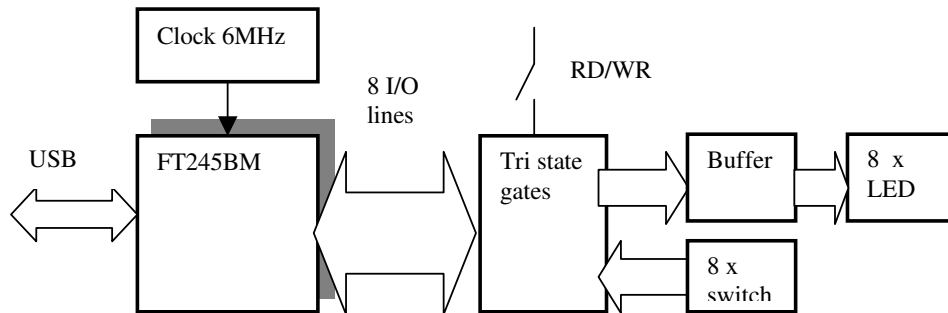


Figura 2: schema bloc a modului de laborator cu FT245BM

Switch-ul RD/WR stabileste sensul transferului de date de la calculatorul gazda prin USB spre LED-uri. Pentru a obtine o lumina statica pe LED-uri, valorile logice sunt buffer-ate.

2.2. Circuitul interfata USB - format serial, FT232BM

Acest circuit are aceeasi schema bloc interna ca si interfata paralela. Singurul bloc diferit este controllerul FIFO care contine semnalele standard interfetei seriale RS232: TxD, RxD, RTS, CTS, DTR, DSR, DCD, RI. Circuitul mai are un semnal specific RS485, precum si 2 semnale la care se pot conecta LED-uri indicind transmisia/ receptia RS232. Blocul controller FIFO contine si un generator de rata de Baud. Modulul de laborator echipat cu acest circuit are schema bloc data in figura 3:

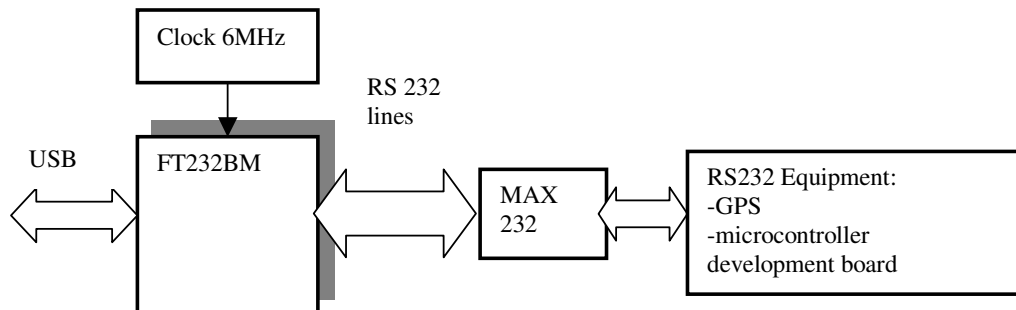


Figura 2: schema bloc a modului de laborator cu FT232BM

2.3.Procedura de lucru

Interfata USB- paralel

In primul rand se instaleaza driverele USB pentru circuitele FTDI care sunt furnizate pe pagina FTDI [4]. Se atrage atentia studentilor asupra procesului de configurare/reconfigurare. Configurarea corecta este verificata in *Device Manager*, figura 3:



Figura 3: captura din Device manager in care se vede driverul pentru circuitul FTDI

Modulul de laborator cu circuitul FTDI este aratat in figura 4:

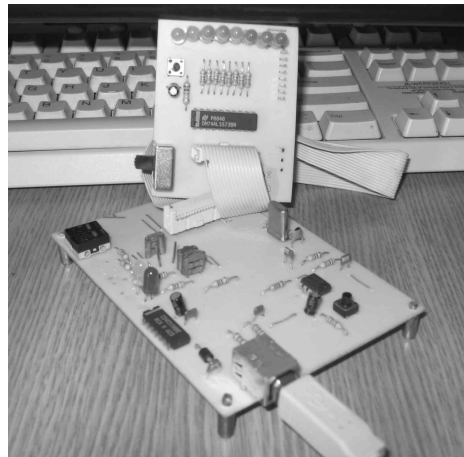


Figura 4: modulul de laborator cu FT245BM (circuitul FTDI este lipit pe spatele placii)

FTDI pune la dispozitia utilizatorilor o biblioteca de functii. Aceasta biblioteca se explica studentilor si li se cere sa foloseasca doar funtiile de trimitere caracter de la calculatorul gazda prin USB si receptie caracter. Dupa familiarizarea cu aceste functii se poate trimite temporizat unele cuvinte care sa creeze un efect luminos de lumina dinamica pe LED-uri. Pentru cei care obtin acest rezultat se cere ca secventa de aprindere sa se schimbe la primirea unui cod care se formeaza cu switch-urile de pe placa.

Interfata USB- serial

Aceasta lucrare este mult mai simpla decat cea precedenta pentru ca FTDI pune la dispozitia utilizatorilor un program care defineste un port serial (COM) virtual. Lucrarea consta in trimiterea printr-un program de tip Hyperterminal a unor cuvinte spre un echipament serial conectat. Spectaculozitatea temei creste daca echipamentul conectat este un sistem GPS serial. O alta varianta folosita a fost conectarea prin USB la calculatorul gazda a unui sistem de dezvoltare cu microcontroller si schimbul de date prin transfer de date USB. La aceasta a doua varianta se pot gasi aplicatii practice deoarece putine

microcontrollere au integrata o interfata USB, asa incit conectarea unui circuit FTDI la un microcontroller prin interfata seriala duce la marirea posibilitatilor de conectivitate a unor mici echipamente de automatizare. La conectarea directa cu un microcontroller se poate renunta la circuitele de modificare de nivel si schema devine simplu de implementat.

[1] C. Gerigan, P. Ogrutan, Tehnici de interfatare, Editura Transilvania Brasov

[4] www.ftdichip.com

[5] P. Ogruțan, C. Gerigan, N. Banciu “ Memories Interfaces and Peripherals. Specialized Interfaces” (“Memorii, interfețe și periferice. Interfețe specializate, Ed. Transilvania Brașov, 2003, 190 pag., ISBN 973-635-118-1