

3.1 Protocoale de comunicații industriale

3.1.1 . Protocoale de tip Fieldbus

Considerații generale

Un **Fieldbus** este rețea industrială pentru control distribuit în timp real.

Un sistem industrial complex automatizat – de exemplu o linie de asamblare – de obicei necesită o ierarhie organizată de sisteme de control pentru a funcționa. În această ierarhie este de obicei un computer central de unde un operator poate monitoriza sau programa întregul sistem. Acesta este în general legat la un strat central de controlere logic programabile (PLC) printr-un sistem de bare colectoare (de exemplu Ethernet). La capătul lanțului de control este fieldbus-ul care leagă PLC-ul de componente care de fapt lucrează cum ar fi senzori, actuatore, motoare, lumini de consolă, comutatoare și contactoare.

3.1.2 Protocoale de tip Modbus

Protocolul de tip **Modbus** este un protocol de comunicație bazat pe arhitectura master/slave sau client/server, proiectat de Modicon pentru utilizarea cu controlere logic programabile (PLC-uri). Cele mai importante motive pentru folosirea protocolului Modbus în detrimentul altor protocoale de comunicație sunt:

1. e publicat în amănunt
2. poate fi implementat în zile și nu luni
3. mută biții sau cuvintele neprelucrate fără a impune multe restricții producătorilor

Modbus permite administrarea unor dispozitive, de exemplu un sistem care măsoară temperatura și umiditatea și comunică rezultatele unui computer. Modbus este de obicei utilizat pentru a conecta un calculator cu o unitate terminală telecomandată (RTU) în control supervizat și sisteme de achiziție de date (SCADA). Versiunile protocolului Modbus există pentru port serial și Ethernet.

Modbus RTU este o reprezentare compactă binară a datei. Standardul Modbus ASCII este explicit și foarte detaliat. Ambele protocoale sunt bazate pe comunicația serială. Modbus/TCP este similar cu Modbus RTU, dar e transmis prin pachete de date TCP/IP.

Există de asemenea o versiune extinsă Modbus Plus (Modbus+ sau MB+), dar rămâne proprietatea Modicon. Este necesar un coprocesor special pentru a prelucra cadrele rapide tip HDLC . Folosește o pereche de fire torsadate cu o rată de transmisie de 1Mbit/s și are specificațiile de instalare asemănătoare cu sistemul EIA/RS-485. Totuși, nu este EIA/RS-485. MB+ include un transformator de izolare în fiecare nod. Câteva repetoare EIA/RS485 lucrează cu el prin efecte colaterale dar Modbus nu se poate cupla direct la calculator.

Fiecare aparat care utilizează Modbus are o adresă unică. Oricare aparat poate transmite o comandă modbus, dar de regulă doar aparatul principal face asta. O comandă modbus conține adresa Modbus a aparatului căruia îi este adresat. Doar aparatul căruia i se adresează modbusul va interpreta comanda, chiar dacă și celelalte aparate primesc acea comandă. Toate comenzile



Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

Modbus conține informații de verificare CRC asigurându-se că o comandă ajunge fără probleme. Comenzile Modbus de bază pot determina un RTU să își schimbe valoarea într-unul din regiștrii săi și de asemenea poate cere unui echipament să trimită înapoi una sau mai multe valori din regiștrii săi.

Există multe modem-uri care suportă modbus. Unele din ele sunt special proiectate pentru acest protocol. Diferitele tipuri de implementări (vezi Figura 3.1. 1) folosesc comunicații cu sau fără fire și chiar SMS sau GPRS. Problemele ce apar de obicei la proiectare includ latența mare și probleme cu sincronizarea.



Figura 3.1. 1. Privire de ansamblu asupra protocolului Modbus

Comunicații în rețele Modbus

Porturile Standard Modbus de pe controlerere Modicon folosesc o interfață compatibilă serială RS-232C care definește ieșirile din conector, cablajele, nivelurile de semnal, vitezele de transmisie și verificatoarele de paritate. Controlererele pot fi direcționate prin rețea sau prin modemuri.

Controlererele comunică folosind o tehnică de copiere (master-slave) în care doar un dispozitiv master poate iniția comunicările. Celelalte dispozitive (slave) răspund prin furnizarea datelor cerute de către dispozitivul master, sau prin executarea operației cerute. Dispozitivele tip „master” includ procesoare gazdă și panouri programabile. Dispozitivele tipice secundare „slaves” includ controlerere programabile.

Dispozitivul „master” poate adresa dispozitive „slaves” individual sau poate inițializa un mesaj tip „broadcast” către dispozitivele „slaves”. Dispozitivele „slaves” întorc un mesaj (un răspuns) la cererile care sunt adresate către ele în mod individual. Răspunsurile nu sunt returnate către cererile tip broadcast de la dispozitivele tip „master”.

Protocolul Modbus stabilește formatul cererilor tip „master” prin plasarea în adresa dispozitivului sau broadcast-ului a unei funcții cod care definește acțiunea cerută, fiecare dată care trebuie trimisă și un câmp de verificare a erorii. Mesajul de răspuns de la dispozitivul tip „slave” este de asemenea construit prin folosirea unui protocol Modbus. El conține câmpuri ce confirmă acțiunea desfășurată, fiecare dată ce trebuie întoarsă și un câmp de verificare a erorii. Dacă apare o eroare în recepția mesajului, sau dacă dispozitivul tip „slave” este incapabil de a realiza acțiunea cerută, dispozitivul tip „slave” va determina apariția unui mesaj de eroare și îl va trimite ca răspuns.

Comunicații pe alte tipuri de rețele

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

În plus față de capacitățile protocoalelor tip Modbus standard, unele modele de controlere Modicon pot comunica prin Modbus Plus cu ajutorul unor porturi încorporate sau adaptoare de rețea sau prin MAP folosind adaptoare de rețea.

În aceste rețele, controlerile comunică folosind o tehnică numită „peer-to-peer” prin care fiecare calculator poate iniția comunicații cu orice alt controler. Astfel, un controler poate funcționa fie ca dispozitiv tip “slave”, fie ca dispozitiv „master” în comunicații separate. Căi interne multiple sunt puse la dispoziție pentru a permite procesarea paralelă a comunicațiilor master/slave.

La nivelul mesajului, protocolul Modbus aplică principiul ”master/slave” chiar dacă metoda de comunicare în rețea este tip „peer-to peer”. Dacă un controler crează un mesaj, face acest lucru ca un dispozitiv “master” și așteaptă răspuns de la dispozitivul tip “slave”. În mod similar, când un controler primește un mesaj, apare un răspuns al dispozitivului tip “slave” care îl întoarce către controlerul original.

Ciclul cerere-răspuns poate fi urmarit in Figura 3.1. 2

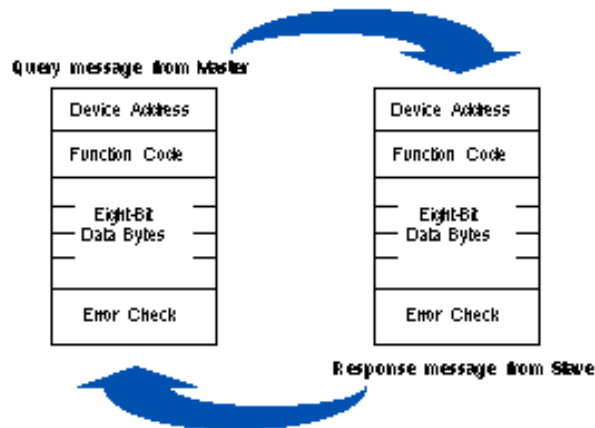


Figura 3.1. 2. Ciclul cerere-răspuns al dispozitivului tip „master”

Cererea

Codul de funcționare în cazul cererii îi transmite dispozitivului tip “slave” căruia i se adresează ce fel de acțiune să execute. Octeții de date conțin orice informație adițională de care dispozitivul tip „slave” va avea nevoie pentru a executa funcția. De exemplu, funcția cod 03 va cere dispozitivului tip “slave” să citească regiștrii și să răspundă la conținutul lor. Câmpul de date trebuie să conțină informația care să îi transmită dispozitivului tip “slave”, de la ce registru să plece și câți regiștrii să citească. Câmpul de verificare a erorii permite dispozitivului tip “slave” să valideze integritatea conținutului mesajului.

Răspunsul

Dacă dispozitivul tip “slave” generează un răspuns normal, codul funcției din răspuns este un rezultat al codului funcției de cerere. Octeții de date conțin datele colectate prin intermediul dispozitivului tip “slave” cum ar fi valorile de registru sau starea. Dacă apare o eroare, codul funcției este modificat pentru a indica faptul că răspunsul este o eroare și octeții de date conțin un cod care o descrie. Câmpul de verificare a erorii permite dispozitivului tip “master” să valideze integritatea conținutului mesajului.

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**Moduri de transmisie serială**

Controlerele pot fi setate pentru comunicarea pe rețele standard Modbus folosind oricare dintre cele două moduri de transmisie: ASCII sau RTU. Utilizatorii pot selecta modul dorit împreună cu parametrii seriali de comunicație (viteza baud, modul de verificare (de paritate, de control, etc...)) pe timpul configurării fiecărui controler. Parametrii modali și seriali trebuie să fie identici pentru toate dispozitivele rețelei Modbus.

Selecția modului ASCII sau RTU este caracteristic doar rețelelor standard Modbus și definește conținutul de biți ale câmpurilor de mesaj transmise în mod serial prin rețele. Determină, de asemenea, ce informații să fie împachetate în câmpurile de mesaj și decodate.

În cazul altor rețele, cum ar fi MAP și Modbus Plus, mesajele Modbus sunt plasate pe cadre care nu sunt în legătură cu transmisia serială. De exemplu, o cerere de citire a regiștrilor de înmagazinare a informației poate fi realizată între două controlere pe Modbus Plus fără a se lua în calcul setarea oricăruia din porturile Modbus.

Modul ASCII

Când controlerele sunt setate pentru comunicarea prin Modbus folosind ASCII (codul standard american de schimb de informații) fiecare octet dintr-un mesaj este trimis ca două caractere ASCII. Principalul avantaj al acestui tip este că permite intervale de timp de până la o secundă între caractere fără a cauza erori.

Sistemul de codare

- hexazecimal , caractere ASCII 0 ... 9, A ... F
- un caracter hexazecimal conținut în fiecare caracter ASCII al mesajului

Biții din octet

- 1 bit de start
- 7 biți de date, cel mai puțin important bit trimis primul
- 1 bit pentru paritate cu soț/fără soț – nici un bit în cazul neparității
- 1 bit de stop dacă se folosește paritatea – 2 biți în cazul neparității

Câmpul de verificare a erorii

- Verificarea redundanței longitudinale

Modul RTU

Când controlerele sunt configurate pentru comunicarea pe o rețea Modbus folosind RTU (remote terminal unit – unitate terminală telecomandată) fiecare octet dintr-un mesaj conține două caractere hexazecimale de 4 biți. Principalul avantaj al acestui mod este că densitatea de caractere este mai mare și permite o mai bună trecere a datelor decât ASCII pentru aceeași viteză de baud (telegrafiere). Fiecare mesaj trebuie să fie transmis într-un șir continuu.

Sistemul de codare

- 8 biți binar, hexadecimale 0 ... 9, A ... F
- două caractere hexazecimale conțin fiecare 8 biți plini de mesaje

Biți pe octeți

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

- 1 bit de start
- 8 biți de date, cel mai puțin important bit trimis primul
- 1 bit pe pentru paritate cu soț/fără soț – nici un bit în cazul neparității
- 1 bit de stop dacă se folosește paritatea – 2 biți în cazul neparității

Câmpul de verificare a erorii

- Verificarea ciclică a redundanței (CRC)

Conținutul mesajelor Modbus

În oricare din cele două moduri de transmisie serială (ASCII sau RTU), un mesaj Modbus este plasat de către dispozitivul emițător într-un cadru care are cunoscut punctul de început și de sfârșit. Acest lucru permite dispozitivelor receptoare să pornească de la începutul mesajului, să citească adresa și să determine cărui dispozitiv îi este adresat. Mesajele parțiale pot fi detectate și erorile pot fi setate ca rezultat.

În cazul rețelelor gen MAP sau Modbus Plus protocolul de rețea se ocupă de conținutul mesajelor cu delimitatorii de început și de sfârșit specifici rețelei. Aceste protocoale se ocupă de asemenea de livrarea către dispozitivul destinație, făcând câmpul de adrese Modbus să se integreze în mesajul care nu este necesar în cazul transmisiei actuale. Adresa Modbus este convertită într-o adresă de nod de rețea și cale de transmisie prin controler sau adaptorul rețelei.

Conținutul tip ASCII

În modul ASCII, începutul mesajului cu două puncte (:) caracter (ascii 3A hex) și se termină prin perechea de caractere sfârșit de linie CRLF (ASCII 0D și 0A hex).

Caracterele permise transmise pentru toate celelalte câmpuri sunt hexazecimale 0...9, A...F. Dispozitivele incluse în rețea monitorizează magistrala rețelei de transmisie în mod continuu pentru caracterul “:”. Când acesta este receptat, fiecare dispozitiv decodează câmpul următor (câmpul de adresă) pentru a afla dacă este dispozitivul adresat.

Intervale de până la o secundă pot apare între caractere în interiorul mesajului. Dacă apare un interval mai mare, dispozitivul de recepție își asumă eroarea apărută. Un mesaj cadru tipic este arătat mai jos:

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	# CHARS	2 CHARS	2 CHARS CRLF

Figura 3.1. 3. Mesaj cadru ASCII

Excepție

Cu controlerile 584 și 984A/B/X , un mesaj ASCII poate în mod normal să se termine după câmpul LRC fără să fie trimise caractere CRLF. Un interval de cel puțin o secundă trebuie să apară. Dacă acest lucru se întâmplă, controlerul va considera că mesajul se termină normal.

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**Conținutul tip RTU**

În modul RTU, mesajele încep cu un interval silențios de cel puțin 3.5 ori mai mare decât intervalul alocat unui caracter. Acesta este cel mai ușor de implementat ca un multiplu de timpi alocați caracterelor la viteza baud care se folosește în rețea (vezi T1-T2-T3-T4 în figura de mai jos). Următorul câmp transmis apoi este dispozitivul adresat.

După ultimul caracter transmis, un interval similar de cel puțin 3.5 ori timpul alocat unui caracter marchează sfârșitul mesajului. Un nou mesaj poate începe după acest interval.

Întregul cadru de semnal trebuie să fie transmis ca un șir continuu. Dacă un semnal silențios de mai mult de 1.5 ori timpul alocat unui caracter apare înainte de completarea cadrului, dispozitivul receptor elimină mesajul incomplet și face ca următorul octet să fie câmpul de adresă al unui nou mesaj.

Un mesaj cadru tipic este prezentat mai jos în Figura 3.1. 4:

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	n x 8 BITS	16 BITS	T1-T2-T3-T4

Figura 3.1. 4. Mesajul cadru RTU

Câmpurile mesajului Modbus**Interpretarea câmpului adresă**

Câmpul de adresă al cadrului mesajului conține două caractere (ASCII) sau 8 biți (RTU). Adresele valide ale dispozitivului “slave” se încadrează în gama 0...247 zecimale. Dispozitivele slave individuale au adrese cuprinse între valorile 1... 247. Un „master” se adresează unui “slave” prin plasarea adreselor slave-ului în câmpul de adrese al mesajului. Când dispozitivul “slave” își trimite răspunsul, își plasează propria adresă în câmpul de adresă al răspunsului pentru a face ca dispozitivul “master” să știe cărui dispozitiv “slave” îi răspunde.

Adresa „0” este folosit pentru adresa de difuzare, pe care toate dispozitivele “slave” o recunosc. Când un protocol Modbus este folosit pe rețele de nivel mai înalt, difuzarea s-ar putea să nu fie permisă sau poate fi înlocuită de alte metode. De exemplu, Modbus Plus folosește o bază de date împărțită global care poate fi reactualizată la fiecare rotație “token”.

Interpretarea câmpului funcție

Câmpul funcției cod din cadrul structurii mesajului conține două caractere (ASCII) sau 8 biți (RTU). Codurile valide se situează în intervalul 1...255 zecimale. Dintre acestea unele coduri sunt aplicabile tuturor controlerelor Modicon, în timp ce unele coduri se pot atribui doar anumitor modele și altele sunt rezervate unei utilizări viitoare.

Când un mesaj este trimis de la un dispozitivul “master” către un dispozitiv “slave” câmpul funcției cod îi transmite dispozitivul “slave” ce fel de acțiune să desfășoare. De exemplu, pentru citirea stărilor ON-OFF a unui grup de intrări distincte; citirea conținutului de date unui grup de regiștrii; pentru citirea stadiului diagnosticării dispozitivul “slave”, pentru a permite încărcarea, imprimarea sau verificarea programului din dispozitivul “slave”.

Când dispozitivul “slave” răspunde unui dispozitiv “master” folosește câmpul cod funcție să indice un răspuns normal (fără eroare) sau un alt tip de răspuns (numit răspuns excepție) unde eroarea apare. Pentru un răspuns normal, dispozitivul “slave” doar transmite

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

ecoul funcției cod originală. Pentru un răspuns excepție, dispozitivele “slave” întorc un cod care este echivalent cu funcția cod originală cu cel mai important bit setat la 1 logic.

De exemplu, pentru ca un mesaj de la dispozitivul “master” către dispozitivul “slave” să citească un grup de regiștrii de evidență trebuie să aibă următoarea funcție cod:

0000 0011 (Hexazecimal 03)

Dacă dispozitivul „slave” preia comanda fără eroare, același cod în răspuns. Dacă apare o excepție, întoarce:

1000 0011 (Hexazecimal 83)

În plus față de modificările funcției cod pentru o excepție de răspuns dispozitivul “slave” introduce un cod unic în câmpul de date al mesajului de răspuns. Acest lucru arată dispozitivului “master” ce fel de eroare apare sau motivul excepției.

Programul de aplicație al dispozitivul “master” are rolul de a interpreta răspunsurile excepție. Procesele tipice sunt ulterioare încercări ale mesajului, pentru a încerca mesaje de diagnostic dispozitivului “slave” și a anunța operatorii.

Conținutul câmpului de date

Câmpul de date este construit folosind seturi de câte două numere hexazecimale în domeniul 00 până la FF hexazecimal. Acestea pot fi făcute dintr-o pereche de caractere ASCII sau printr-un caracter RTU în funcție de modul de transmisie serială.

Câmpul de date al mesajelelor trimise de la un dispozitiv “master” la un dispozitiv “slave” conține informații adiționale pe care dispozitivul “slave” trebuie să le utilizeze ca să preia acțiunea definită de codul funcției. Acesta poate include puncte ca adrese discrete și de registru, cantitatea punctelor ce trebuie luate în calcul și numărarea octeților cu date efectivi din câmp.

De exemplu, dacă dispozitivul “master” cere unui dispozitiv “slave” să citească un grup de regiștrii (codul funcției 03), câmpul de date specifică registrul de început și câți regiștrii sunt de citit. Dacă dispozitivul “master” scrie unui grup de regiștrii din dispozitivul “slave” (codul funcției 10 hexazecimal), câmpul de date specifică registrul de început, câți regiștrii să scrie, numărătoarea octeților care urmează în câmpul de date, și data ce trebuie trecută în regiștrii.

Dacă nu apar erori, câmpul de date al unui răspuns de la un dispozitiv “slave” către un dispozitiv “master” conține data cerută. Dacă apare o eroare câmpul conține un cod de excepție pe care aplicația “master” o poate utiliza pentru determinarea următoarei acțiuni.

Câmpul de date poate fi inexistent (de lungime zero) în anumite tipuri de mesaje. De exemplu într-o cerere de la dispozitivul “master” pentru ca un “slave” să răspundă cu lista sa de evenimente de comunicație (funcția hexazecimală cod 0B), dispozitivul „slave” nu are nevoie de nici o informație suplimentară. Funcția cod specifică chiar ea acțiunea.

Conținutul câmpului de verificare a erorii

Două tipuri de metode de verificare a erorii se folosesc în rețelele Modbus. Conținutul câmpului de verificare a erorii depinde de metoda care se utilizează.

ASCII

Când modul ASCII se folosește pentru un tip de caracter, eroarea câmpului de verificare conține două caractere ASCII. Caracterele de verificare a erorii sunt rezultatul unui calcul de



Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

verificare a redundanței longitudinale (LRC), realizat asupra conținutului mesajului, exclusiv la început, unde sunt cele două puncte și terminând cu caracterele CRLF.

Caracterele LRC sunt adăugate mesajului la ultimul câmp care precedează caracterele CRLF.

RTU

Când modul RTU se folosește pentru tipul caracterului, eroarea câmpului la verificare conține o valoare de 16 biți implementată ca doi octeți. Eroarea la verificare este rezultatul unei verificări ciclice de redundanță ce are loc asupra conținutului mesajului.

Câmpul CRC este adăugat mesajului ca ultimul câmp al mesajului. Când acest lucru este finalizat, octetul cel mai puțin semnificativ al câmpului apare primul urmat de octetul cel mai semnificativ. Octetul cel mai semnificativ al CRC este ultimul octet trimis din mesaj.

Informații adiționale asupra verificării erorii vor fi prezentate în continuarea acestui capitol.

Pașii detaliați pentru generarea câmpurilor LRC și CRC pot fi găsite de asemenea în acest capitol.

Modul de transmisie serial

Când mesajele se transmit pe rețele standard Modbus, fiecare caracter sau octet este trimis în această ordine (de la stânga la dreapta):

Cel mai puțin semnificativ bit (LSB) ... cel mai semnificativ bit (MSB)

Cu câmpul de caractere ASCII ordinea de succedare a biților este

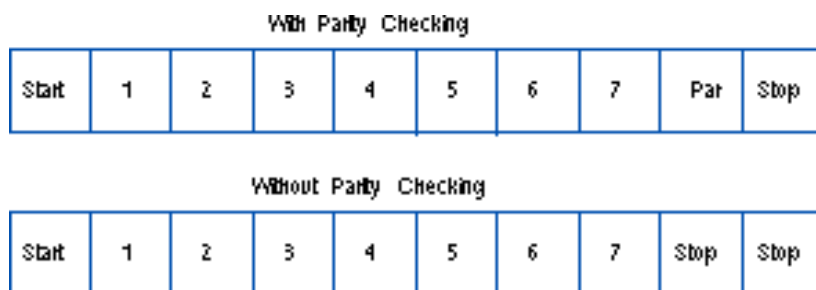


Figura 3.1. 5. Ordinea Biților (ASCII)

Cu cadrul de caractere RTU, ordinea de succedare a biților este prezentată în Figura 3.1. 6

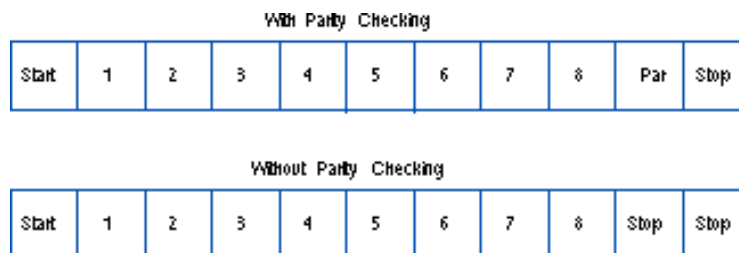


Figura 3.1. 6. Ordinea biților (RTU)

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**Metodele de verificare a erorii**

Rețelele standard Modbus folosesc două tipuri de verificare a erorii. Verificarea parității (par, impar) poate fi opțional aplicată fiecărui caracter. Verificarea cadrului (LRC sau CRC) se aplică întregului mesaj. Atât verificarea de caracter și verificarea cadrului mesajului sunt generate în dispozitivul "master" aplicat conținutului mesajului înaintea transmisiei. Dispozitivul "slave" verifică fiecare caracter și întregul cadru al întregului mesaj pe timpul recepției.

Dispozitivul "master" este configurat de utilizator pentru a aștepta un interval predeterminat de timp de a opri comunicarea. Acest interval este setat pentru a fi suficient de lung astfel încât orice dispozitiv tip "slave" să răspundă normal. Dacă dispozitivul tip "slave" detectează o eroare de transmisie, mesajul este oprit. Dispozitivul tip "slave" nu va construi un răspuns la dispozitivul tip "master". Deci, timpul de oprire va expira și va permite programului "master" să țină sub control eroarea.

Notă: Un mesaj adresat unui dispozitiv tip "slave" inexistent va cauza de asemenea o oprire.

Alte rețele cum ar fi MAP sau Modbus Plus folosesc cadre de verificare la un nivel deasupra conținutului Modbus al mesajului. În aceste rețele, verificarea câmpurilor de mesaj LRC sau CRC nu se aplică. În cazul unei erori de transmisie, protocoalele de comunicație specifice acelor rețele anunță dispozitivul origine că a apărut o eroare și îi permit să încerce din nou sau să renunțe, în funcție de cum a fost setat. Dacă mesajul este livrat dar dispozitivul tip "slave" nu poate să răspundă, o eroare timeout apare și poate fi detectată de programul dispozitivului tip "master".

Verificarea parității

Utilizatorii pot configura controlerele pentru verificarea parității pare sau impare sau pentru verificarea neparității. Acest lucru va determina modul în care bitul de paritate va fi setat în fiecare caracter.

Dacă oricare din paritățile pare sau impare se specifică, cantitatea unui bit va fi numărată în porțiuni de date ale fiecărui caracter (7 biți de date pentru modul ASCII sau 8 pentru RTU). Bitul de paritate va fi atunci setat la un rezultat de 0 sau 1 într-un total de biți 1 pari sau impari. De exemplu, acești 8 biți cu date sunt conținuți într-un câmp de caractere RTU.

1100 0101

Cantitatea totală de biți tip 1 din cadru este de patru. Dacă se folosește paritatea PAR, paritatea bitului aflat în câmp va fi 0, ducând cantitatea totală de biți tip 1 la un număr par și el (4). Dacă se folosește paritatea IMPAR paritatea va fi 1, ajungând la o cantitate impară (5).

Când este transmis mesajul, paritatea bitului este calculată și aplicat câmpului fiecărui caracter. Dispozitivul de recepție numără cantitatea biților 1 și setează eroarea dacă nu este aceeași cu cea configurată pentru acel dispozitiv (toate dispozitivele din rețeaua Modbus trebuie să fie configurate pentru a folosi aceeași metodă de verificare a parității).

De notat că verificarea parității poate să detecteze o eroare doar dacă un număr impar de biți este ridicat sau coborât într-un câmp de caractere pe durata transmisiei. De exemplu, dacă se apelează la verificarea parității tip IMPAR și doi biți tip 1 sunt eliminați dintr-un caracter ce conține trei, rezultatul va fi tot un număr impar (1).



Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

Dacă este specificată paritatea FĂRĂ PARITATE bitul de neparitate este transmis și verificarea FĂRĂ PARITATE poate fi făcută. Un bit de stop adițional este transmis pentru a umple cadrul caracterului.

Verificarea LRC

În modul ASCII, mesajele includ un câmp de verificare a erorii care se bazează pe metoda LRC. Câmpul LRC verifică conținutul mesajului în afară de cele două puncte de început și perechea de sfârșit CRLF. El este aplicat indiferent de orice metodă de verificare folosită pentru caracterele individuale ale mesajului.

Câmpul LRC este de un octet conținând o valoare binară de 8 biți. Valoarea LRC este calculată de dispozitivul de transmisie, care adaugă LRC-ul mesajului. Dispozitivul de recepție calculează un LRC pe durata recepției mesajului și compară valoarea calculată cu valoarea pe care o primește în câmpul LRC. Dacă cele două valori nu sunt egale, apare o eroare.

LRC este calculat prin adunarea octeților succesivi de câte 8 biți ai mesajului, îndepărtând caracterele de control și apoi complementând față de 2 rezultatul. Este realizat pe câmpul de mesaj ASCII excluzând caracterul două puncte de la începutul mesajului și excluzând perechea CRLF de la sfârșitul mesajului.

Într-o ordine gradată, funcția CKSM calculează un LRC din conținutul mesajului.

Verificarea CRC

În modul RTU, mesajele includ un câmp de verificare a erorii care se bazează pe metoda CRC. Câmpul CRC verifică conținutul întregului mesaj. Este aplicat fără a ține seama de orice metodă de verificare a parității pentru caracterele individuale ale mesajului.

Câmpul CRC este de doi octeți, conținând o valoare binară de 16 biți. Valoarea CRC este calculată printr-un dispozitiv de transmisie care adaugă CRC mesajului. Dispozitivul de recepție recalculează un CRC pe durata recepției mesajului și compară valoarea calculată cu valoarea pe care o primește în câmpul CRC.

CRC începe printr-o preîncărcare de 16 biți registru până la toate valorile de 1. Apoi, începe un proces de aplicare succesivă de octeți ai mesajului către conținutul actual al registrului. Doar 8 biți de date din fiecare caracter se folosesc pentru generarea CRC-ului. Biții de început sau de sfârșit și bitul de paritate nu se aplică CRC-ului.

Pe durata generării CRC, fiecare caracter de 8 biți este comparat SAU exclusiv cu conținutul registrului. Apoi, rezultatul este mutat în direcția celui mai puțin semnificativ bit (LSB), cu un zero în poziția cu cel mai semnificativ bit (MSB). LSB este extras și examinat. Dacă LSB era un 1, registrul este comparat SAU exclusiv cu un preset de valoare fixă. Dacă LSB-ul era un 0, nu are loc nici o comparație SAU.

Acest proces este repetat până când se realizează 8 mutări. După ultima (a 8-a) mutare următorii octeți de câte 8 biți sunt comparați SAU exclusiv cu valoarea curentă a registrului, și procesul se repetă pentru încă 8 mutări după cum s-a explicat mai sus. Conținutul final al registrului după ce toți octeții mesajului au fost aplicați, este valoarea CRC.

Când CRC este adăugat mesajului, bitul cel mai puțin semnificativ este adăugat primul, urmat de cel mai semnificativ bit.

În ordinea logică, funcția CKSM calculează un CRC din conținutul mesajului.



Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**3.1.3 Protocolul PROFIBUS****Generalități**

PROFIBUS este un protocol care conține trei variante compatibile între ele. PROFIBUS este un sistem deschis, standardizat în DIN 19-245. În 1996, acest standard a fost inclus în standardul european EN 50170 volumul 2 fără a i se aduce modificări.

Cele trei variante au fost specificate pentru diferite aplicații după cum se arată în Figura 8:

- PROFIBUS-FMS (Fieldbus Message Specification) – Specificațiile Mesajelor Fieldbus
- PROFIBUS-DP (Decentralized Periphery) – Periferice descentralizate
- PROFIBUS-PA (Process Automation) – Automatizare de procese

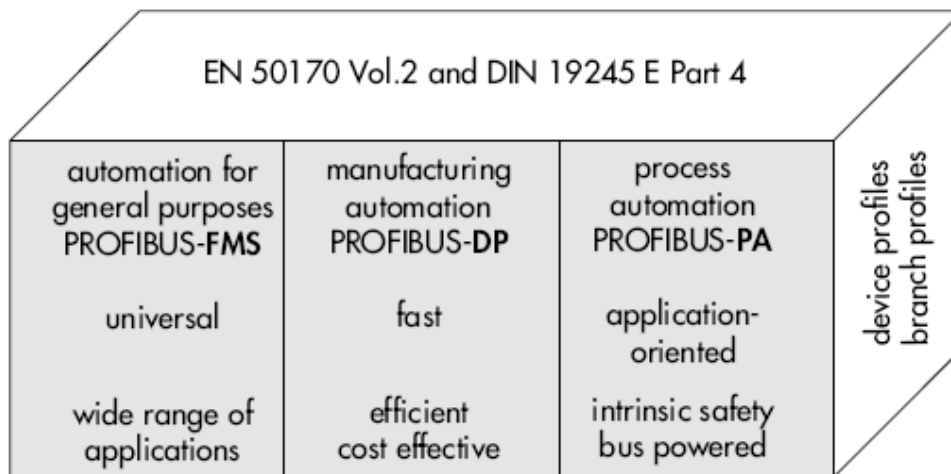


Figura 3.1. 7. Variantele PROFIBUS

- Varianta FMS oferă utilizatorului o mare varietate de funcții care, însă, îl fac mai complex de implementat comparativ cu celelalte variante. Puternicele servicii FMS pot fi utilizate pentru a rezolva chiar și sarcini complexe de comunicație. Această variantă PROFIBUS suportă comunicația între sisteme automatizate ca și schimbul de date între dispozitive plasate pe teren. Astfel, FMS poate fi utilizat pentru o largă paletă de aplicații, lucrând la o viteză medie de transmisie.
- Varianta DP este soluția de mare viteză a PROFIBUS. A fost proiectată și optimizată special pentru comunicația între sisteme automatizate și dispozitive descentralizate. Astfel, PROFIBUS-DP necesita mai puțin de 2ms pentru transmiterea a 1kbyte de date de intrare sau ieșire. Astfel chiar și cele mai critice comunicații din punctul de vedere al timpului pot fi rezolvate.
- PROFIBUS-DP comunică în mod exclusiv prin trafic de date ciclic. Fiecare dispozitiv schimbă datele sale de intrare-ieșire cu cu dispozitivul de automatizare , stăpânul de clasă 1, într-un anume ciclu de timp.
- În ingineria proceselor la fel ca și în automatizarea proceselor, monitorizarea operațiilor necesită un dispozitiv de vizualizare pe lângă cel de automatizare. Acest stăpân de clasă 2 este responsabil pentru funcțiile de pornire, parametrizare și monitorizare a

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

dispozitivelor din câmpul de lucru. Aceasta necesită ca datele dispozitivelor să fie citite sau scrise în timpul unor operații independente de ciclul de control.

- Deoarece specificațiile originale DP nu ofereau servicii speciale pentru aceste acțiuni, au fost definite extensii pentru acestea în 1997. Aceste extensii pot fi implementate în mod opțional și sunt compatibile cu cu protocolul DP existent și toate versiunile anterioare. Varianta DP extinsă este cunoscută ca PROFIBUS-DPV1 . pe lângă serviciile de comunicare ciclice, DP oferă și servicii aciclice cum ar fi mesaje de alarmă, diagnostic, parametrizare și control a dispozitivelor din câmp.
- A treia variantă PROFIBUS, PROFIBUS-PA, vine în întâmpinarea cerințelor speciale în automatizările proceselor. Comunicația PA se bazează pe serviciile oferite de DPV1, și este implementată ca un sistem parțial înglobat într-un sistem de comunicație DP de nivel înalt. Spre deosebire de aplicațiile automatizate din producție care cer timpi de ciclu scurți de câteva milisecunde, în automatizarea proceselor sunt importanți și alți factori precum următorii:
 - Tehnici de transmisie cu siguranță intrinsecă
 - Dispozitive de câmp alimentate prin bus
 - Transmisii robuste de date
 - Interoperabilitate (standardizarea funcțiilor dispozitivelor)

Toate cele trei variante PROFIBUS lucrează cu o singură metodă standardizată de acces la bus. Sunt de asemenea capabile să utilizeze aceeași tehnică de transmisie și să opereze simultan pe aceeași linie de comunicație. Mediul de transmisie utilizat include ori cabluri ecranate răsucite în pereche, sau unde radio.

Terminologie: master slave

În PROFIBUS-PA , ca și în DP, pot fi diferențiate trei tipuri de dispozitive după cum urmează:

- Stăpânul de clasă 1 (DPM1) este unitatea centrală de control a întregului sistem , de exemplu un PLC, care schimbă date cu dispozitivele din câmp într-un ciclu de mesaj specificat. Sunt transmise toate valorile măsurate, valorile comenzilor și datele de stare ale dispozitivelor. Atunci când este comparat cu tehnicile convenționale, sistemul de comunicație ciclică de clasă 1 își asumă sarcina semnalelor analogice standardizate de 4-20mA și în plus adaugă schimb bidirecțional de date.
- Stăpânii de clasă 2 (DPM2) sunt utilizați pentru scopuri de monitorizare și realizare de operații precum și la pornire. Schimbul de date asociat funcțiilor are loc dacă este necesar. De aceea, stăpânii de clasă 2 necesită în mod exclusiv comunicare aciclică.
- Un sclav este un periferic sau dispozitiv de câmp care comunică doar atunci când îi cere un stăpân acest lucru:
 - Actuatorii primesc informație de la stăpâni și influențează în mod activ procesul
 - Senzorii colectează date de stare și proces și oferă aceste informații stăpânilor.



Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**Implementarea modelului OSI**

Toate variantele PROFIBUS sunt bazate pe modelul de referință ISO/OSI pentru rețele de comunicație. Datorită necesităților în acest domeniu, au fost implementate din motive de eficiență, numai straturile 1 și 2, iar pentru versiunea FMS stratul 7. Reprezentarea modului în care s-a făcut această implementare poate fi văzută în Figura 9.

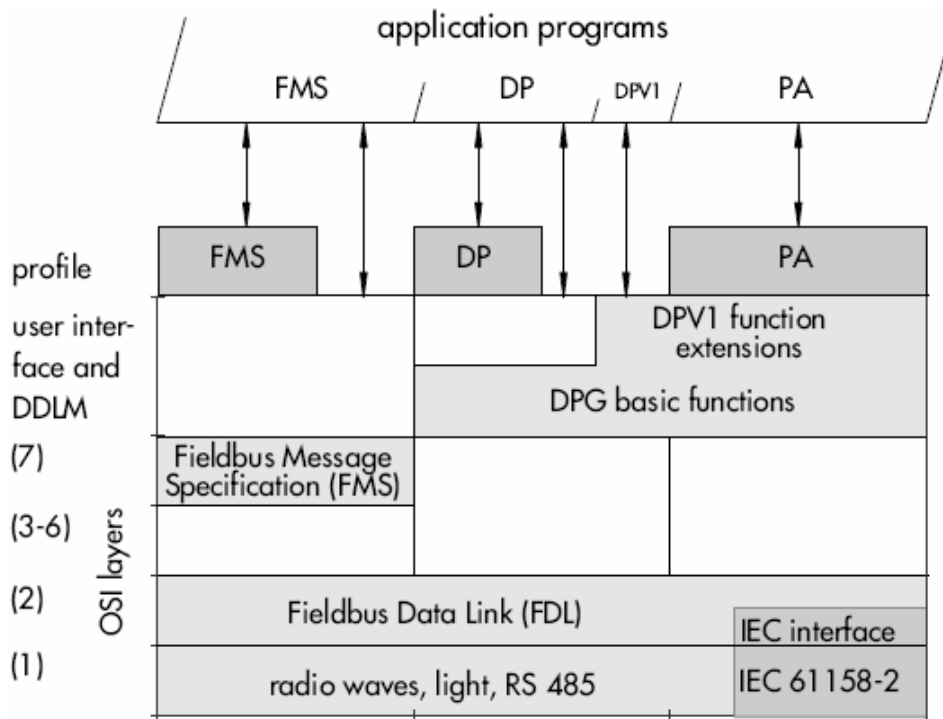


Figura 3.1. 8. Implementarea modelului ISO/OSI

Stratul fizic

Sunt diferite moduri de a realiza transmisia pentru PROFIBUS-PA:

- Prin standardul RS485, sau
- În concordanță cu IEC 61158-2 care a fost specificat în mod special pentru medii cu risc ridicat de explozie și unde alimentarea se face prin bus. Sunt patru variante IEC 61158-2, dar numai PROFIBUS-PA utilizează modul 31,25kbit/s în tensiune.

Accesul la bus și adresarea

În comunicația PROFIBUS sunt posibile sisteme multimaster (cu stăpâni multipli). Fiecare stăpân primește un identificator în cadrul unei ferestre de timp definite și are controlul absolut asupra comunicației din rețea pe durata acelei ferestre.

O adresă de 7 biți servește la identificarea participanților în rețea. Adresele din domeniul 0 la 127 și următoarele sunt rezervate:

- Adresa 126: adresa de bază pentru asignarea automată a adreselor prin stăpân
- Adresa 127: trimiterea telegramelor în rețea

Mecanisme de comunicație și protecție la erori

PROFIBUS –PA și –DP sunt echipate cu o serie de mecanisme pentru asigurarea unei comunicații fără probleme. De exemplu, încă de la inițializarea sistemului, sunt verificate câteva posibile surse de erori.

După inițializarea sistemului, sclavii sunt gata pentru schimbul de date numai dacă stăpânul a trimis mai întâi o telegramă de configurare și parametrizare. Numai când aceste telegrame verifică proprietățile funcționale, sclavul acceptă comenzile de la master. De exemplu, numărul de linii configurate de către stăpân trebuie să existe în realitate pe dispozitiv.

Cu ajutorul comenzilor get-cfg, poate fi încărcată configurarea tuturor sclavilor. Astfel, erorile de parametrizare pot fi evitate dacă stăpânul compară aranjamentul planificat al dispozitivelor cu configurația existentă în sistem. Informația necesară pentru aceasta – tipul dispozitivului, numărul de intrări și ieșiri, formatarea și lungimea datelor – sunt recepționate prin intermediul fișierelor și bazelor de date ale dispozitivului. În Figura 3.1. 9 este arătat ciclul tipic de inițializare la pornirea sistemului.

Secvența de inițializare			
Încărcarea sclavi	configurației de rețele	Stăpânul verifică configurația	Get_cfg
Scrierea parametrilor sclavilor	- servicii de comunicație - parametrii rețelei		Set_prm
Configurarea sclavilor	Intrări / Ieșiri		Set_cfg
Încărcarea sclavilor	configurației	Stăpânul verifică setările	Get_cfg

Figura 3.1. 9. Inițializarea la start a unui sistem PROFIBUS

Pe lângă verificarea erorilor în telegramele de date în timpul funcționării, procesul de comunicație este controlat și monitorizat de către mecanisme de siguranță controlate de către timp și protocol.

Fiecare stăpân controlează comunicația cu sclavii săi prin intermediul unor baze de timp speciale care verifică secvența de timp a datelor utile din traficul de rețea.

În ce privește sclavii, există o funcție specială de monitorizare care comută ieșirile într-o stare sigură, predefinită, dacă nu are loc nici o comunicație în intervalul de timp alocat.

Ieșirile sclavilor sunt protejate de un sistem securizat de protecție la accesare. Acesta asigură faptul că în sistemele multimaster accesul la scriere este permis numai stăpânului autorizat, în timp ce citirea intrărilor și ieșirilor poate fi făcută fără drepturi de acces.

Securitatea sistemului este sporită și de faptul că fiecare stăpân de clasă 1 transmite în mod regulat propria stare către toți sclavii asignați, în cursul unui interval de timp ce poate fi configurat utilizând o comandă specifică. Stăpânul poate fi parametrizat într-o așa manieră încât poate comuta toți sclavii într-o stare de siguranță și încetează transmisia de date în cazul unei erori în sistem (de exemplu se defectează un sclav).

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**3.1.4 Standarde de comunicații industriale**

Mediul industrial este caracterizat prin distante mari de transmisie și mai ales prin paraziti foarte puternici. In asemenea cazuri se recomanda standardul de transmitere a a datelor cunoscut sub denumirea de RS422 și RS 485.

RS422 este un standard diferential care permite transmiterea datelor la distante mult mai mari decit RS232. Impedanta de intrare ridicata a dispozitivelor RS422, combinata cu capabilitatile sporite ale iesirii permit o conexiune cu pina la 10 noduri (vezi Figura 3.1. 10).

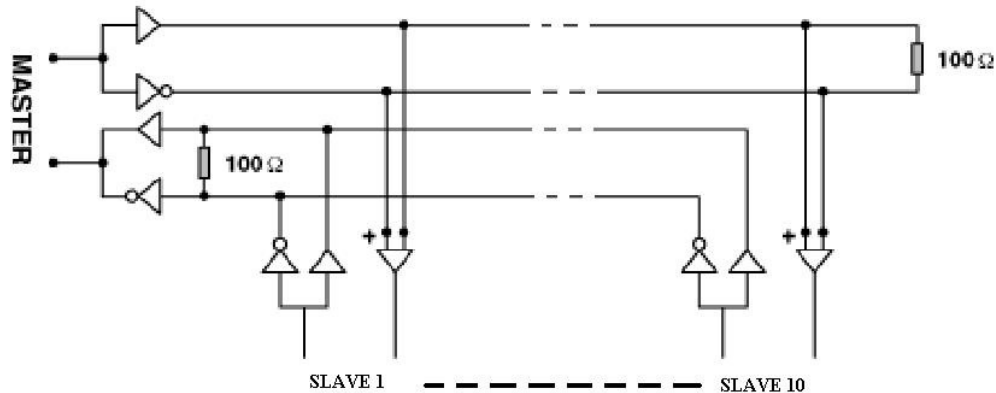


Figura 3.1. 10. Modul de interconectare al mai multor utilizatori de tip industrial bazat pe standardul RS422

Alt avantaj RS422 este utilizarea de cai de recepție și de transmisie separate pentru care nu este necesar controlul direcției. Orice procedura de handshaking necesara între dispozitive se realizează sau software (control XON/XOFF) sau hardware (pereche separata de semnal).

Parametru	Condiții	Min	Max	Unitate
Tensiune de ieșire in gol			± 10	V
Tensiune de ieșire in sarcina	Sarcina de 100Ω	+2 -2		V
Impedanța de ieșire	A la B		100	Ω
Curentul de scurtcircuit la ieșire	Intre ieșire și masa		150	mA
Timp de creștere la ieșire	Sarcina de 100Ω		10	% din durata unui bit
Tensiunea de mod comun	Sarcina de 100Ω		± 3	V
Senzitivitate receptor	$V_{cm} < \pm 7$ V		± 200	mV
Gama tensiunii de mod comun		-7	7	V

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

la receptor				
Impedanța de intrare receptor		4		kΩ
Tensiunea diferențială la receptor	Operațional		±10	V

Dispozitivele RS422 si RS485 se confunda adesea; unele se presupun a fi versiunea duplex a celorlalte. Diferențele de natura electrica sunt totuși datorate gamei tensiunilor de mod comun si impedanțelor de intrare ale receptorilor, astfel ca se utilizează in aplicații diferite. Dispozitivele RS485 îndeplinesc toate specificațiile RS422, după cum se observa din tabelul următor, astfel ca pot fi folosite in aplicații RS422, dar invers nu este posibil. Gama tensiunii de mod comun pentru RS485 este de la -7V la +12V, mult mai mare decât gama -3V la +3V permisa pentru RS422. De asemenea, impedanța de intrare minima a receptorilor este 4kΩ pentru RS422 si 12 kΩ pentru dispozitivele RS485.

Parametru	Condiții	Min	Max	Unitate
Tensiune de ieșire in gol		+1.5 -1.5	+6 -6	V
Tensiune de ieșire in sarcina	Sarcina de 100Ω	+1.5 -1.5	+5 -5	V
Curentul de scurtcircuit la ieșire	Intre ieșire si masa		250	mA
Timp de creștere la ieșire	Sarcina de 54Ω si 50pF		30	% din durata unui bit
Tensiunea de mod comun	Sarcina de 54Ω		±3	V
Senzitivitate receptor	-7V < V _{cm} < +12V		±200	mV
Gama tensiunii de mod comun la receptor		-7	12	V
Impedanța de intrare receptor		12		kΩ

Permitind utilizarea de cabluri lungi, dispozitivele RS485 au devenit un standard popular in aplicații tip point-of-sale, industriale si de telecomunicații. Gama sporita a tensiunii de mod comun permite linii mai lungi si o impedanța de intrare mai mare per nod, astfel ca se pot conecta mai multe noduri la bus.

3.1.5 Topologii de rețea

Configurația rețelei nu este definita in specificațiile RS422 sau RS485. In majoritatea cazurilor proiectantul poate astfel utiliza o configurație optimizata cerințelor fizice ale sistemului. Sisteme pe doua sau patru fire



Structuri hardware si algoritmi specifici microsistemelor EMBEDDED

Sistemele RS422 necesita o pereche de fire pentru fiecare semnal, deci o pereche pentru transmisie, una pentru receptie si perechi aditionale pentru fiecare semnal de control utilizat (daca este cazul). Capabilitatile tristate ale sistemelor RS485 permit utilizarea unei singure perechi de fire pentru a partaja semnalele de transmisie si de receptie la comunicatii semiduplex. Aceasta configuratie „pe doua fire“ (in realitate trebuie folosit si al treilea, semnalul de masa) reduce costul cablarii.

Dispozitivele RS485 pot fi configurate intern sau extern pentru configuratii pe doua sau patru fire. Dispozitivele RS485 configurate intern pe doua fire ofera conexiunile A si B, marcate uneori si „-“ si „+“.

Dispozitivele configurate pentru comunicatii pe patru fire dispun de conexiuni A si B atat pentru receptie cit si pentru transmisie. Utilizatorul poate realiza configuratia in doua fire prin conectarea impreuna a liniilor de transmisie cu cele de receptie, astfel ca dispozitivele avind configuratie in patru fire ofera proiectantului flexibilitatea cea mai mare. In orice caz, semnalul de masa trebuie sa fie conectat in sistem, pentru a tine tensiunea de mod comun la receptor intr-o gama sigura de functionare. Circuitul de interfata poate functiona si fara semnalul de masa, dar cu pretul sacrificarii fiabilitatii si a imunitatii la zgomot. Figurile urmatoare ilustreaza conexiunile in cazul sistemelor pe doua si respectiv patru fire.

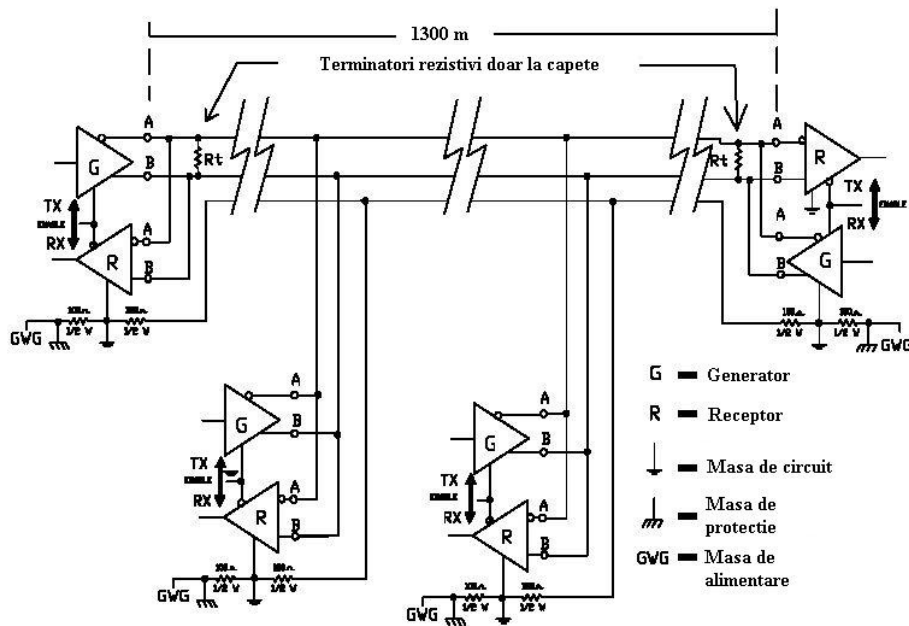


Figura 3.1. 11. Conexiunile sistemelor pe doua fire

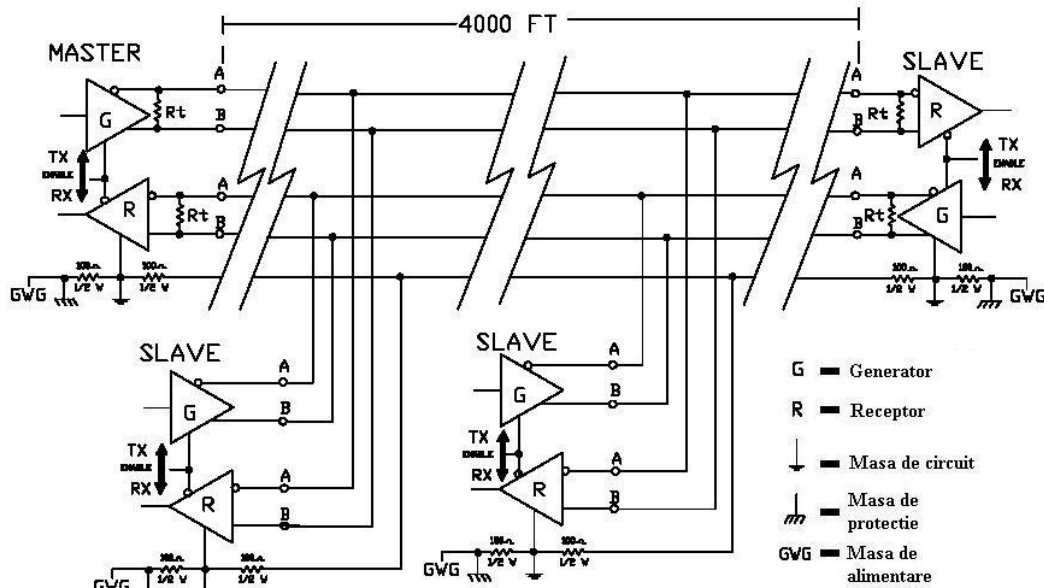


Figura 3.1. 12. Conexiunile sistemelor pe patru fire

3.1.6 Adaptarea liniilor

Rezistoarele de terminatie se utilizeaza pentru a adapta impedanta nodului cu impedanta liniei de transmisie utilizate. In cazul neadaptarii impedantelor semnalul transmis nu este complet absorbit de sarcina si o anumita fractiune este reflectata inapoi in linia de transmisie. Daca impedantele sursei, liniei de transmisie si a sarcinii sint egale atunci aceste reflexii nu apar. Rezistoarele de terminatie cresc sarcina etajelor driver, sporesc complexitatea instalarii, modifica cerintele de incarcare si fac modificarea sistemului mai complicata.

Decizia utilizarii adaptarii trebuie bazata pe studiul lungimii cablului si a ratei de transmisie a datelor din sistem. Ca regula generala, daca propagarea liniei de date este mult mai mica decit durata unui bit, nu este necesara adaptarea. Regula aceasta presupune ca reflexiile se vor atenua in citeva deplasari inainte si inapoi in linie. Intrucit unitatea UART receptoare va esantiona data la mijlocul duratei bitului, este important ca nivelul semnalului in acel moment sa fie stabil.

Exista mai multe metode de adaptare a liniilor de date. Metoda recomandata este adaptarea paralel. Aceasta metoda presupune conectarea in paralel cu liniile A si B ale receptorului a unei rezistente pentru a adapta impedanta caracteristica a liniei specificata de producatorul cablului (120 Ohmi este o valoare uzuala). Aceasta valoare descrie impedanta intrinseca a liniei de transmisie si nu depinde de lungimea liniei. O rezistenta de adaptare sub 90 Ohmi nu se recomanda a fi utilizata. Rezistoarele de adaptare trebuie plasate doar la extremitatile liniei si nu mai mult de doua intr-un sistem care nu utilizeaza repetitoare. Acest tip de adaptare adauga incarcare masiva in curent continuu si poate supraincarca tipurile de convertoare RS232 la RS485 alimentate din port. Alt tip de adaptare este cea in curent alternativ, prin adaugarea unei capacitati reduse in serie cu rezistenta de adaptare, astfel eliminind efectul incarcarii in curent continuu. Dezavantajul metodei este dependenta inalta a

Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

selectiei capacitatii de proprietatile sistemului. Figura 3.1. 13 prezinta cele doua modalitati de adaptare a liniilor

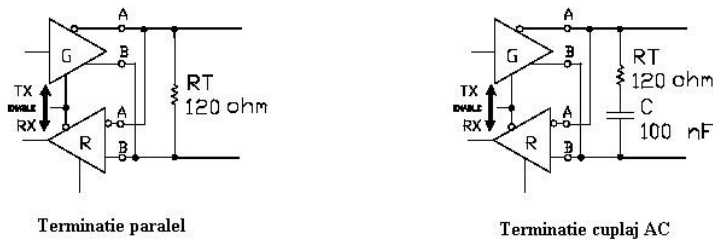


Figura 3.1. 13. Modalități de adaptare a liniilor cu elemente pasive

Bibliografie:

1. Istvan Sztojanov, Sever Pașca, Elisabeta Buzoianu, Aplicații hardware și software cu microcontrolerul PIC12F675, Editura Cavallioti, ISBN 978-973-7622-54-9, Bucuresti 2008
2. Istvan Sztojanov, Alexandru Vasile, Elisabeta Buzoianu, Sever Pașca, *Programarea microcontrolerelor din familia Intel, Aplicații practice hardware cu 80C552*, Editura Man-Dely, ISBN 973-85681-5-3, București 2004.
3. <http://vega.unitbv.ro/~romanca/EmbSys/>
4. <http://facultate.regielive.ro/cursuri/electronica/>