

Universitatea Tehnica a Moldovei

Facultatea Calculatoare, Informatica si Microelectronica

Departamentul Informatica si Ingineria Sistemelor

Disciplina:

Bazele Transmiterii de Date

**Tema Nr. 6.4. Interfete si Protocoale de
Comunicare.**

Titular de curs:

Conf.univ.,dr. V. Ababii

Subiecte abordate:

- Controlorul I8255. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod paralel.
- Controlorul I8251. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod secvențial.
- Interfața UART/USART
- Interfața COM.
- Magistrala USB.
- Interfețe RS: RS-232, RS-422, RS-423, RS-485.
- Interfața Bluetooth.
- Interfața IrDA.
- Interfața I2C.
- Interfața SPI.
- Interfața CAN.
- Interfața Ethernet.
- Controloare specializate pentru implementarea interfeței Ethernet. Protocoale de comunicare.

Standarde USB.

USB 1.0 *Low Speed* - apărut în 1996, a fost mai mult un prototip pentru USB 1.1, capabil de transferuri de date la viteze de până la 1,5 Mbps.

USB 1.1 *Full Speed* - apărut în 1998, cu viteze de transfer de 12 Mbps.

USB 2.0 *Hi-Speed* - apărut în aprilie 2000 și suportă viteze de transfer teoretice de maxim 480 Mbps. Este compatibil cu USB 1.0 și USB 1.1.

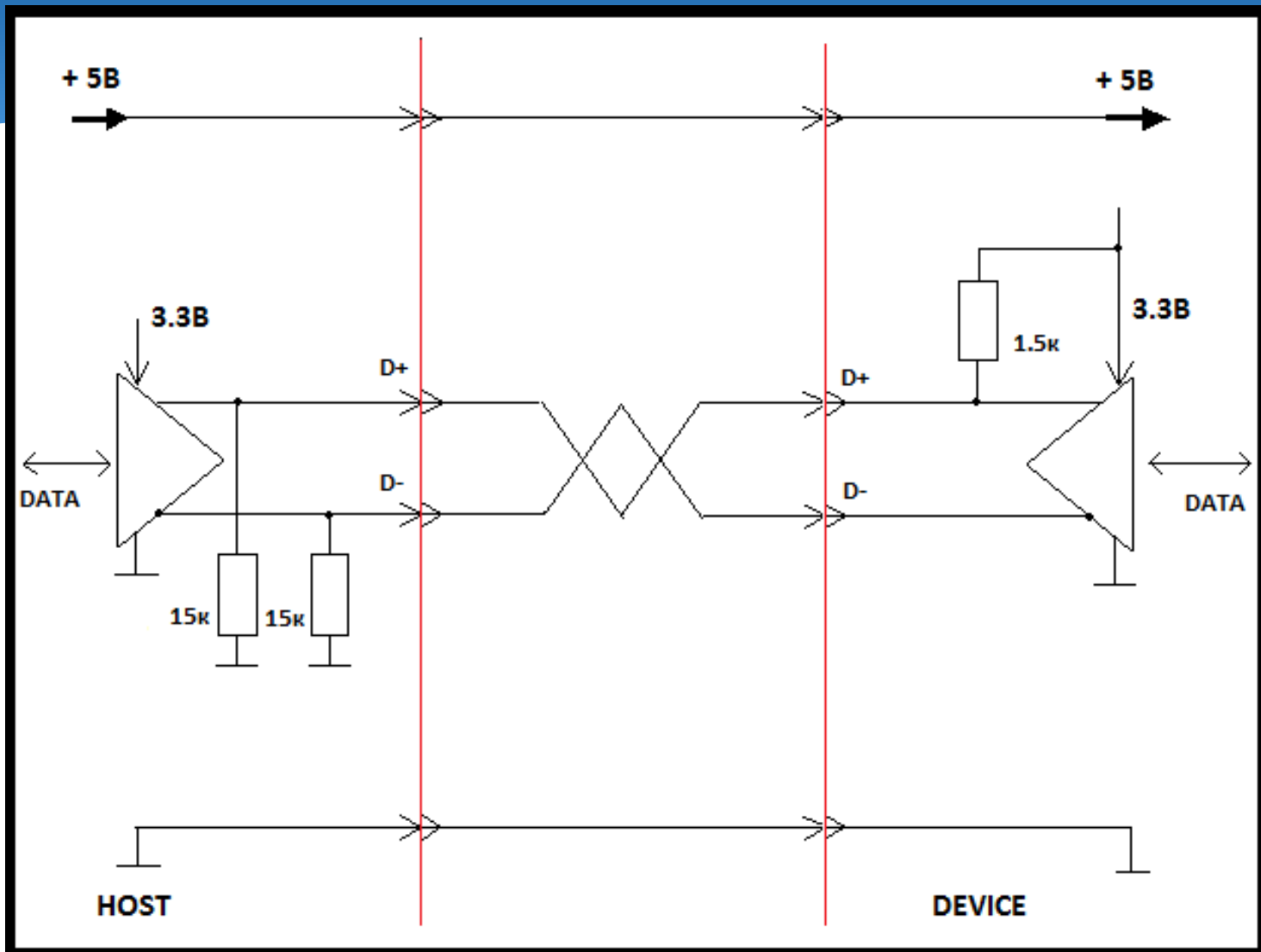
USB 3.0 *SuperSpeed* - a intrat pe piață în noiembrie 2008 și oferă viteze de transfer teoretice de până la 5 Gbps.

USB 3.1 *SuperSpeed+* - a apărut în iulie 2013. Este capabil de transferuri de date la viteze maxime teoretice de 10 Gbps, dublu față de USB.

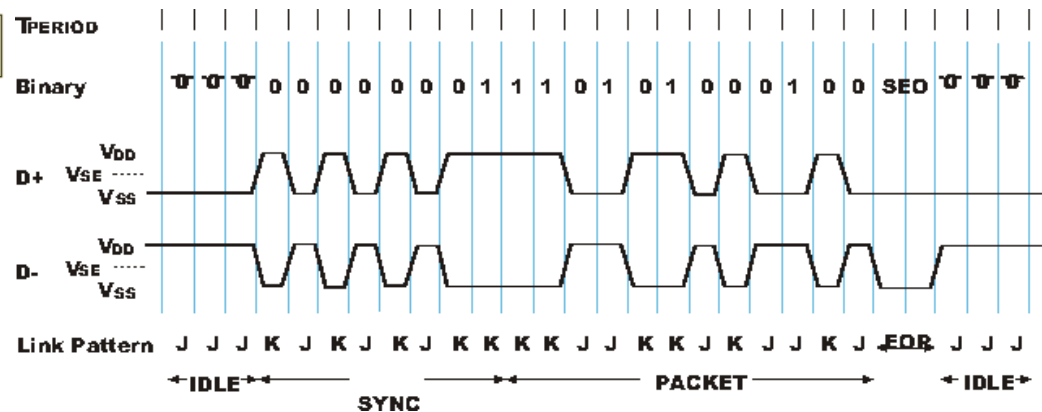
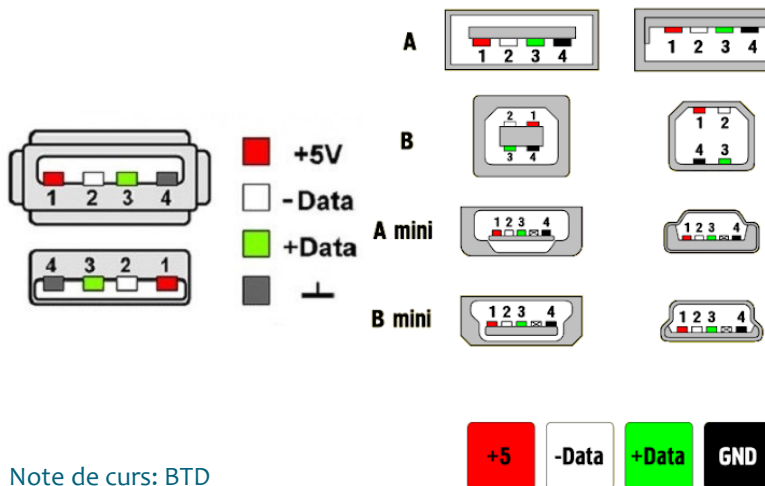
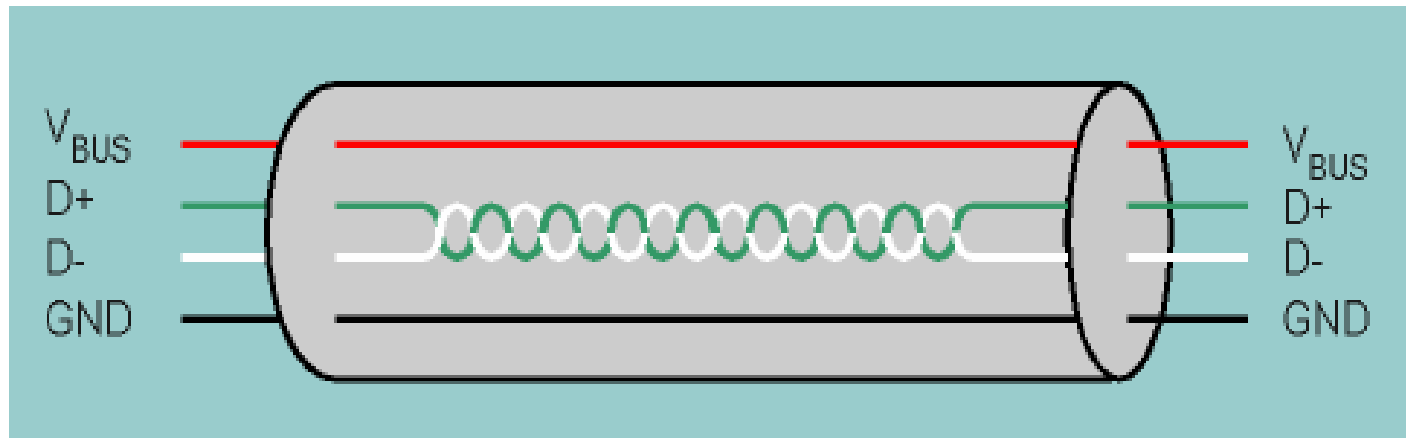
USB 3.2 - lansat în august 2017, introduce două noi moduri *SuperSpeed +* de transfer prin conectorul USB-C cu rate de 10 Gbit/s și 20 Gbit/s.

USB4 - este succesorul USB 3.2, se bazează pe specificația protocolului Thunderbolt 3 și a fost lansat pe 29 august 2019. Suportă un debit de 40 Gbit/s.

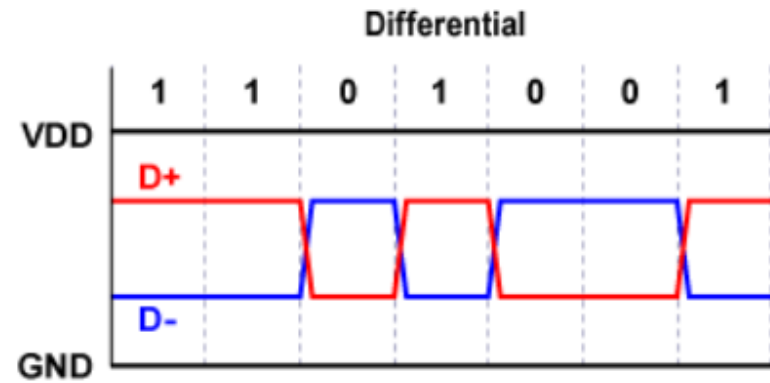
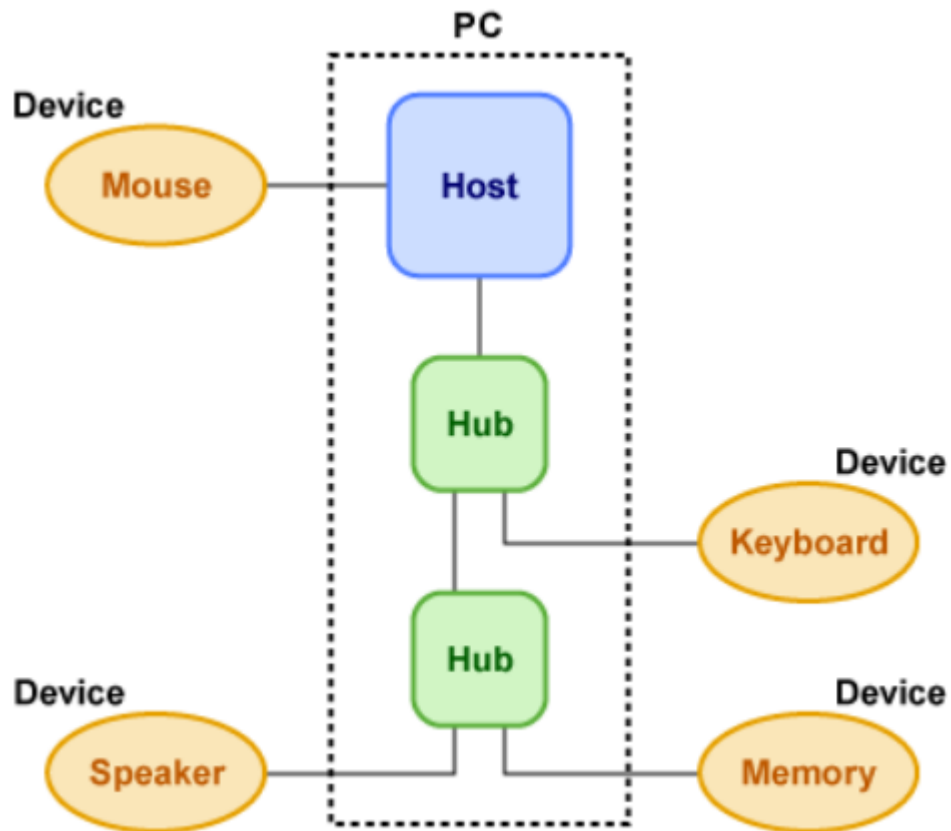
Interconectarea dispozitivelor USB.



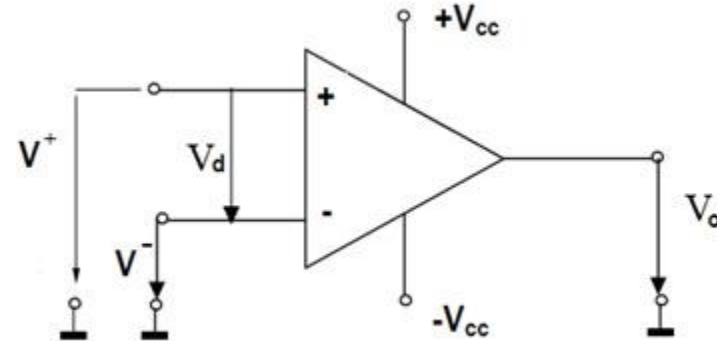
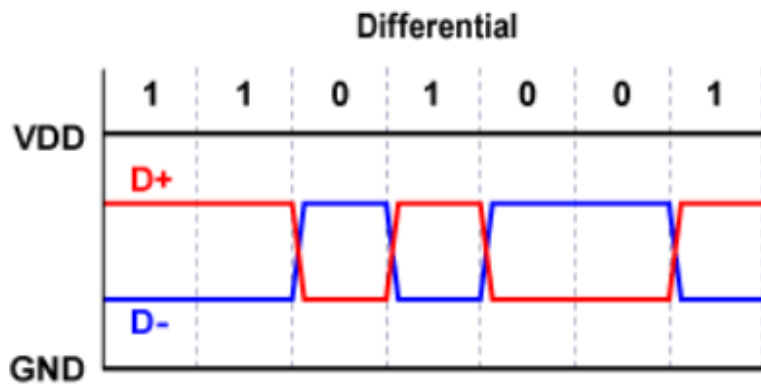
Cablul si conectoare USB.



Comunicarea in USB.



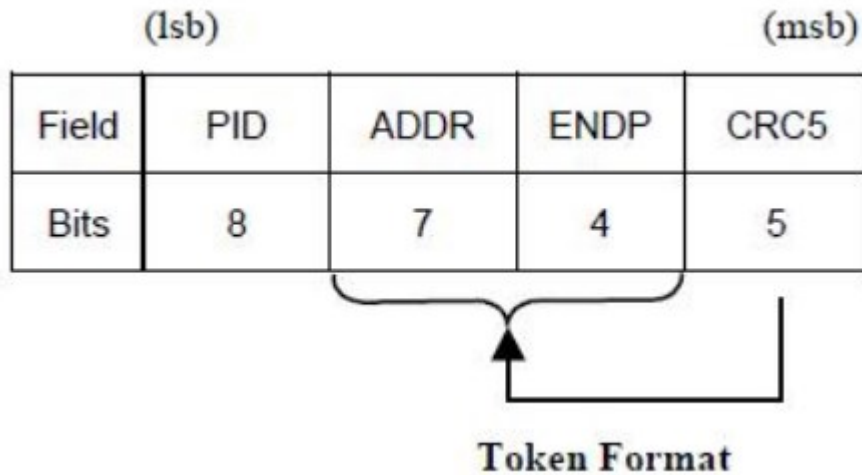
Codificarea a in USB.



$$V_o = (V^+) - (V^-)$$

Formatul dateor in USB.

Setup packet

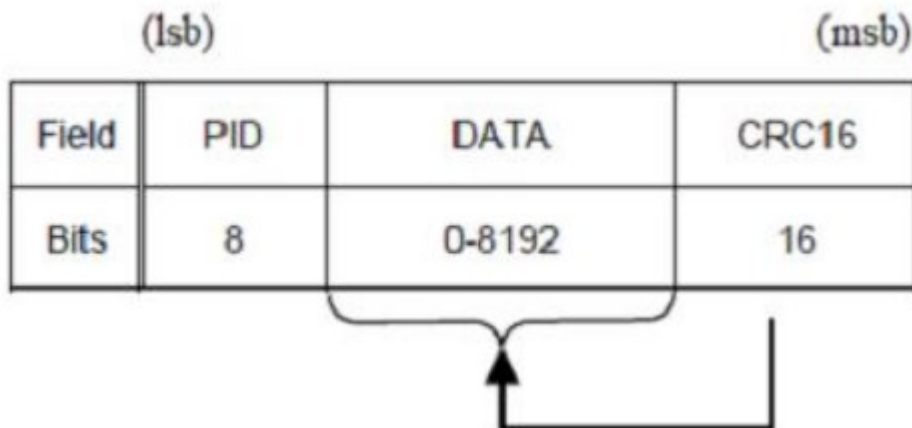


Packet ID

The address field specifies. Being 7 = 127 devices.

Cyclic Redundancy Checks

Data packet



End of packet

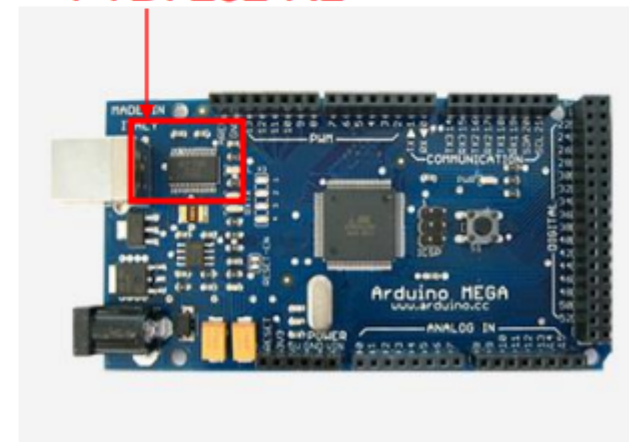
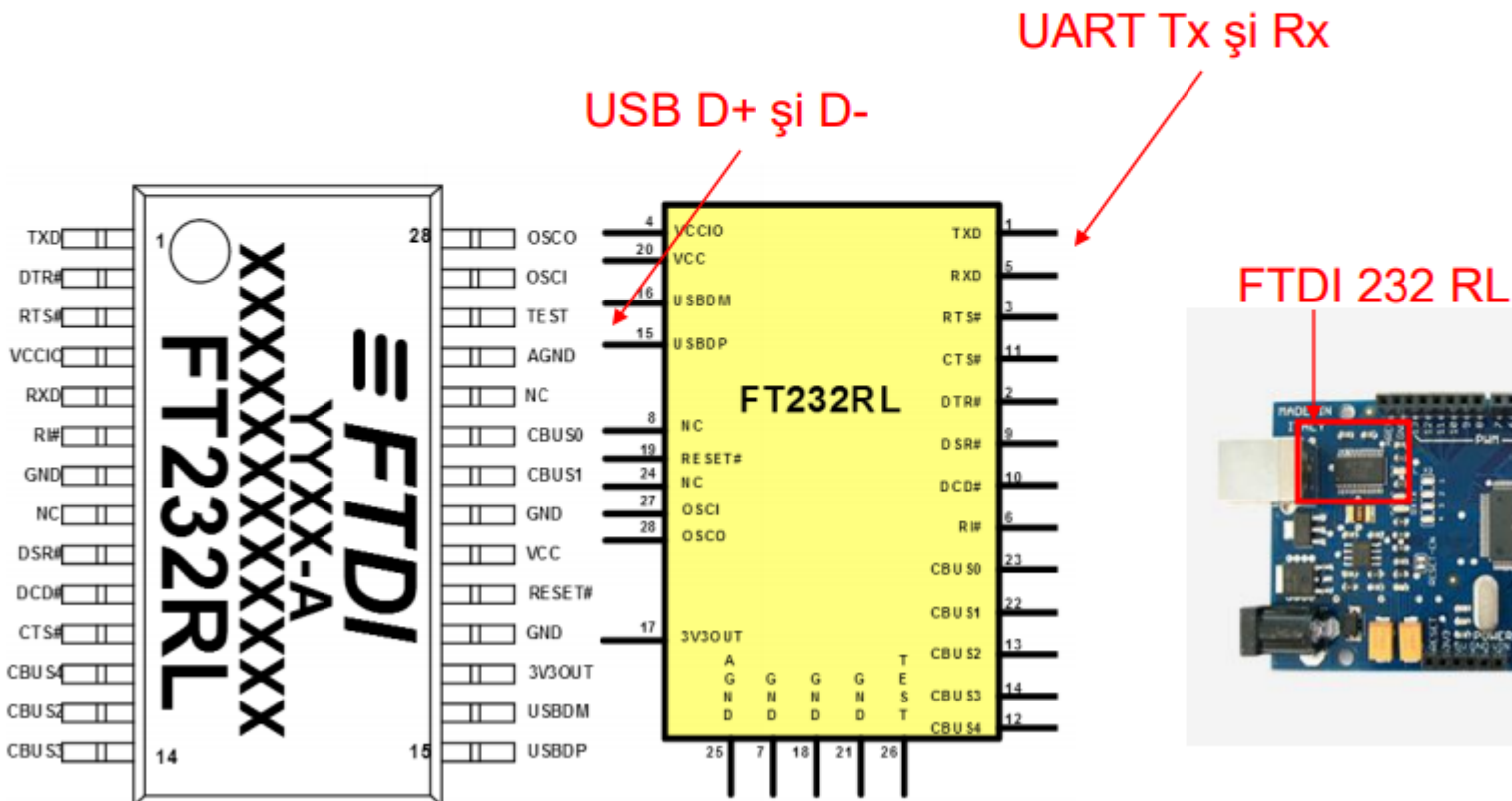
Convertorul UART in USB pentru Arduino.

UART și USB:

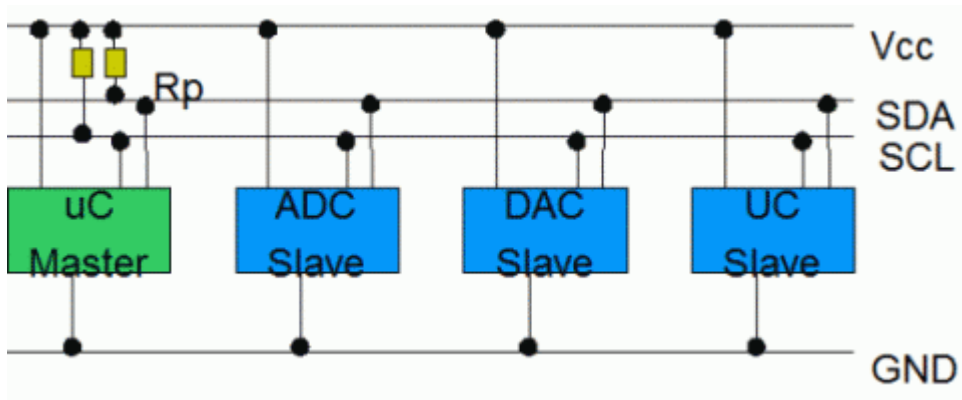
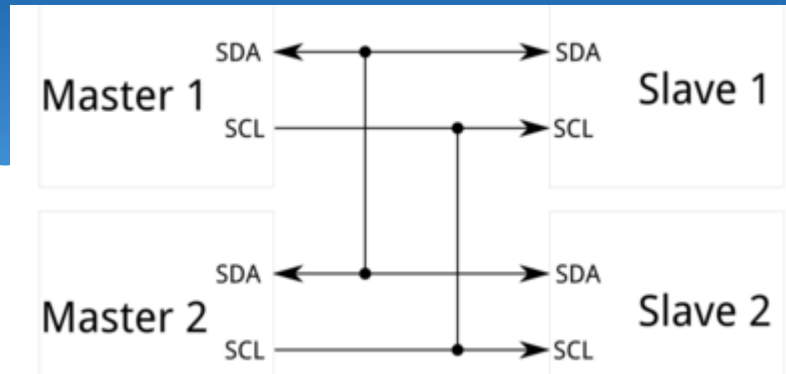
Folosirea unui adaptor FTDI (Future Technology Devices International Ltd)

Arduino Mega folosește the FT232RL chip

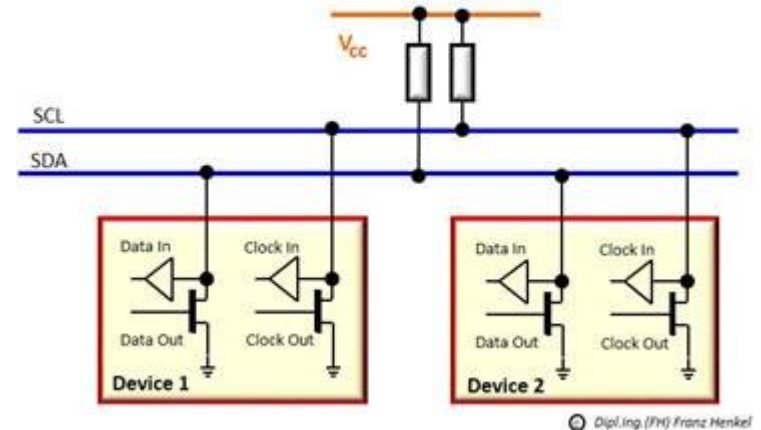
- Vizibil ca un port COM virtual la PC
- Conversie bi-direcțională între USB și UART



Interfata I2C.

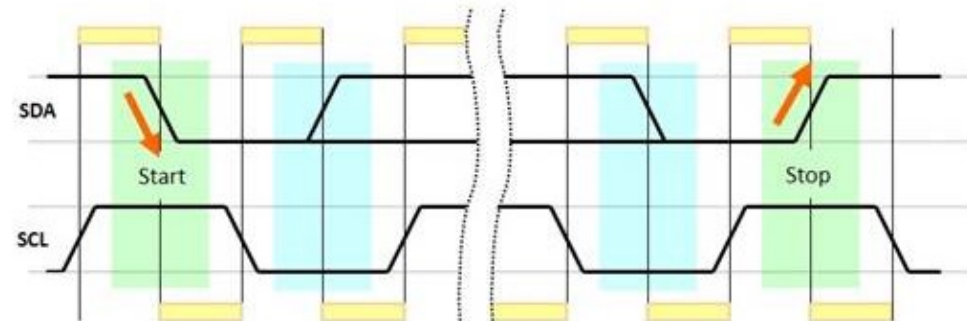
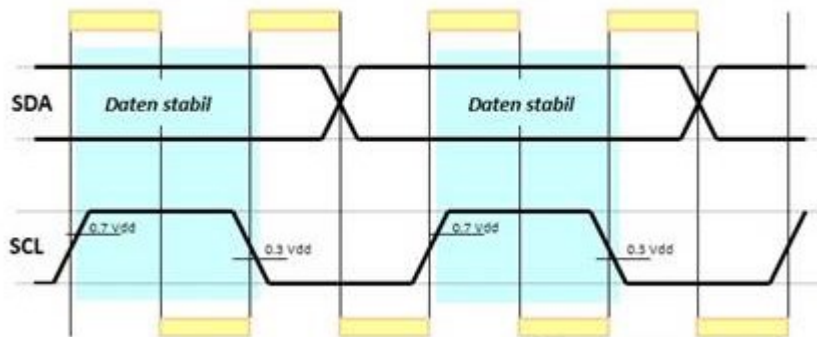
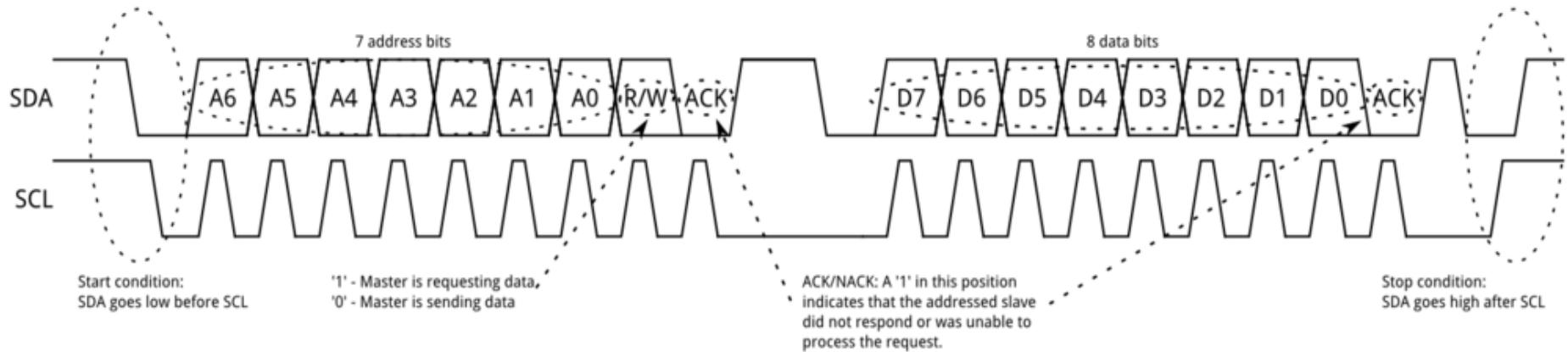


I²C Bus Physical Interface

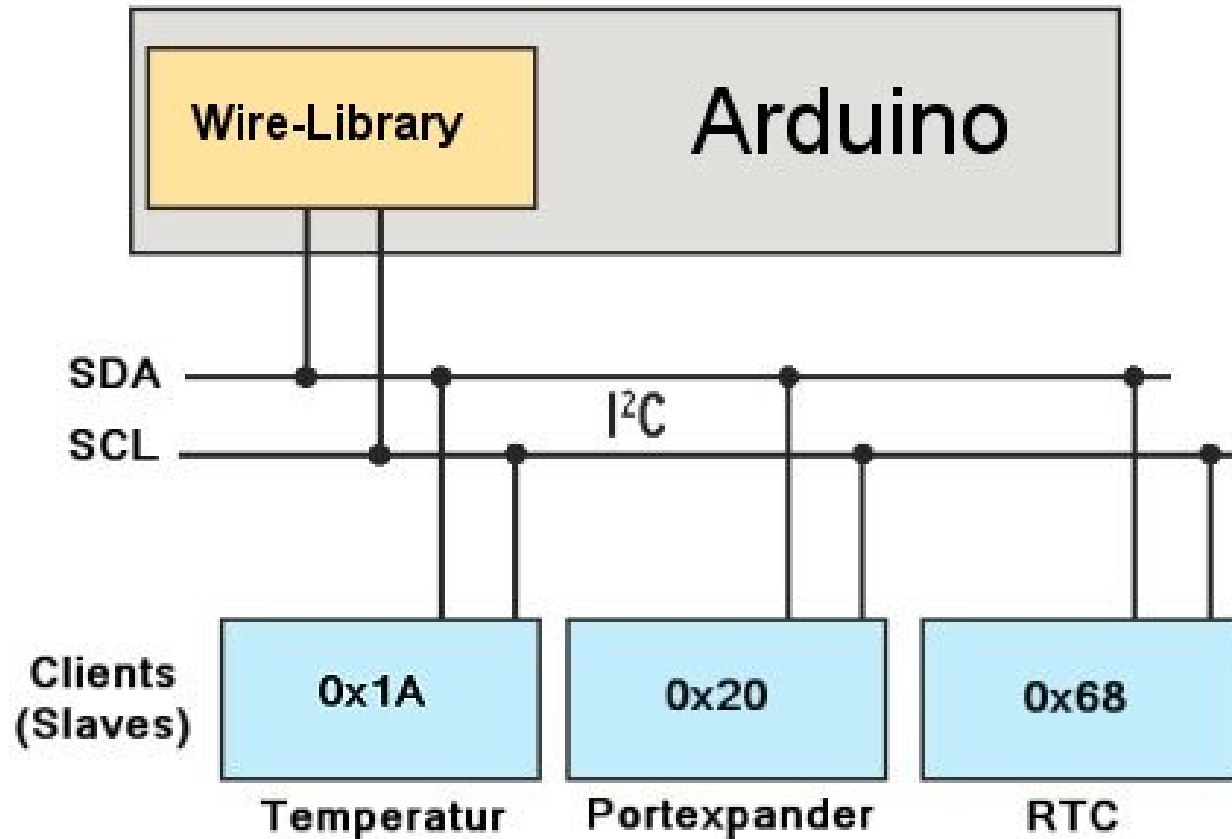


© Dipl.-Ing. (FH) Franz Henkel

Interfata I2C.



Interfata I2C.



Programarea comunicare dintre un masret si un slave I2C.

```
#include <Wire.h> /// Master
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(1);
  Wire.write('H');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(1);
  Wire.write('L');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(2);
  Wire.write('H');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(2);
  Wire.write('L');
  Wire.endTransmission();

  delay(500);
}
```

Note de c

```
#include <Wire.h> /// Slave
const byte slaved = 1;
void setup()
{
  Wire.begin(slaved);
  Wire.onReceive(receiveEvent);

  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);
}
void loop()
{
}
void receiveEvent(int howMany)
{
  char inChar;

  while(Wire.available() > 0)
  {
    inChar = Wire.read();

    if (inChar == 'H')
    {
      digitalWrite(13, HIGH);
    }
    else if (inChar == 'L')
    {
      digitalWrite(13, LOW);
    }
  }
}
```

Programarea comunicare dintre un masret si 2 slave I2C.

```
#include <Wire.h> /// Master
void setup()
{
  Wire.begin();
}
void loop()
{
  Wire.beginTransmission(1);
  Wire.write('H');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(1);
  Wire.write('L');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(2);
  Wire.write('H');
  Wire.endTransmission();

  delay(500);

  Wire.beginTransmission(2);
  Wire.write('L');
  Wire.endTransmission();

  delay(500);
}
```

```
#include <Wire.h> /// Slave 1
const byte slaved = 1;
void setup()
{
  Wire.begin(slaved);
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);

  pinMode(13,OUTPUT);
  digitalWrite(13,LOW);
}
void loop()
{
}
void receiveEvent(int howMany)
{
  char inChar;
  while(Wire.available() > 0)
  {
    inChar = Wire.read();
    if (inChar == 'H')
    {
      digitalWrite(13, HIGH);
    }
    else if (inChar == 'L')
    {
      digitalWrite(13, LOW);
    }
    else
    {
      Serial.println("Different from H and L");
    }
  }
}
```

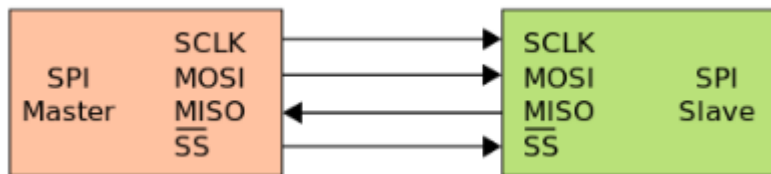
```
#include <Wire.h> /// Slave 2
const byte slaved = 2;
void setup()
{
  Wire.begin(slaved);
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);
}
void loop()
{
}
void receiveEvent(int howMany)
{
  char inChar;

  while(Wire.available() > 0)
  {
    inChar = Wire.read();

    Serial.println(inChar);
  }
}
```

Interfata SPI.

Interfața serială SPI (*Serial Peripheral Interface*) este o interfața sincronă standard de mare viteză, ce operează în mod full duplex. Numele ei a fost dat de Motorola. Ea e folosită ca sistem de magistrală serială sincronă pentru transmiterea de date, unde circuitele digitale pot să fie interconectate pe principiul master-slave. Aici, modul master/slave înseamnă că dispozitivul (circuitul) digital master inițiază cuvântul de date. Mai multe dispozitive (circuite) digitale slave sunt permise cu *slave select individual*, adică cu selectare individuală.



Serial Peripheral Interface (SPI)

- Comunicare seriala sincrona
- Mod de functionare full duplex
- Configurare Master sau Slave
- Frecventa variabila
- Se poate folosi pentru conexiune intre placi

SPI-ul are patru semnale logice specifice.

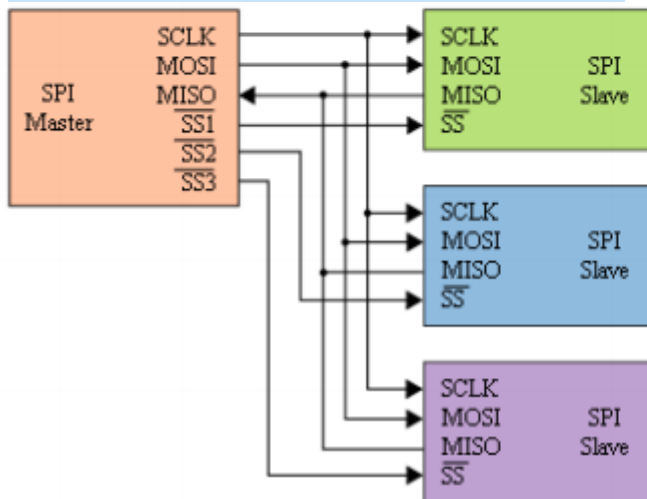
- **SCLK** - *Ceas serial* (ieșire din master).
- **MOSI/SIMO** - *Master Output, Slave Input* (ieșire master, intrare slave).
- **MISO/SOMI** - *Master Input, Slave Output* (intrare master, iesire slave).
- **SS** - *Slave Select* (active low, ieșire din master).

Interfata SPI.

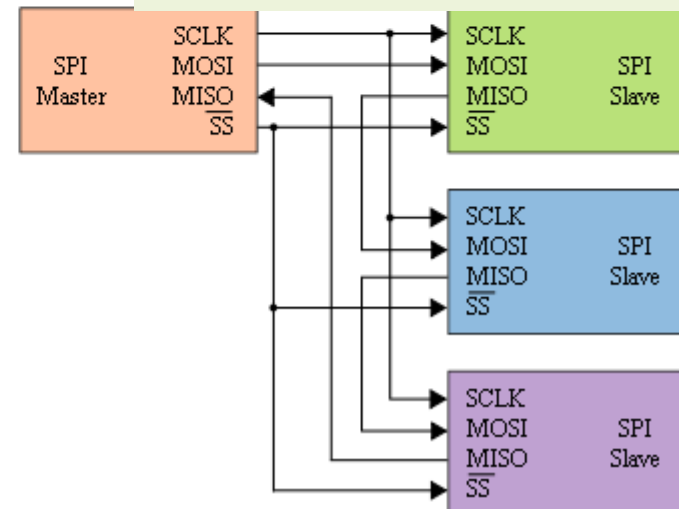
Utilizarea semnalului SS

- Pentru un dispozitiv “slave” **SS** este semnal de intrare
 - SS cu valoare 0 inseamna activarea dispozitivului slave. O tranzitie din 0 in 1 inseamna resetarea ciclului de transfer (marcheaza sfarsitul unui pachet)
 - SS cu valoare 1 – dispozitiv slave inactiv
- Pentru un dispozitiv “master” **SS** poate fi:
 - Iesire – prin el activeaza dispozitivul “slave” pentru comunicare
 - Intrare – daca se permit mai multe dispozitive master, o valoare ‘0’ la intrarea SS trece dispozitivul curent in modul “Slave”
- Configuratii cu mai multe dispozitive: - semnale **SS** independente sau “daisy chain”

Accesare selectiva - paralela

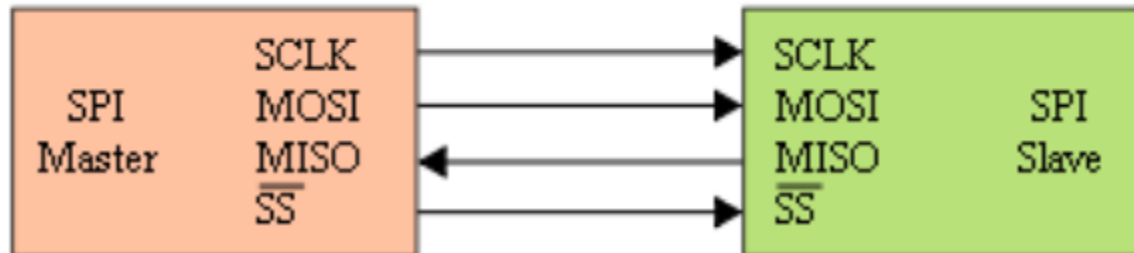


Accesare cu acces in serie



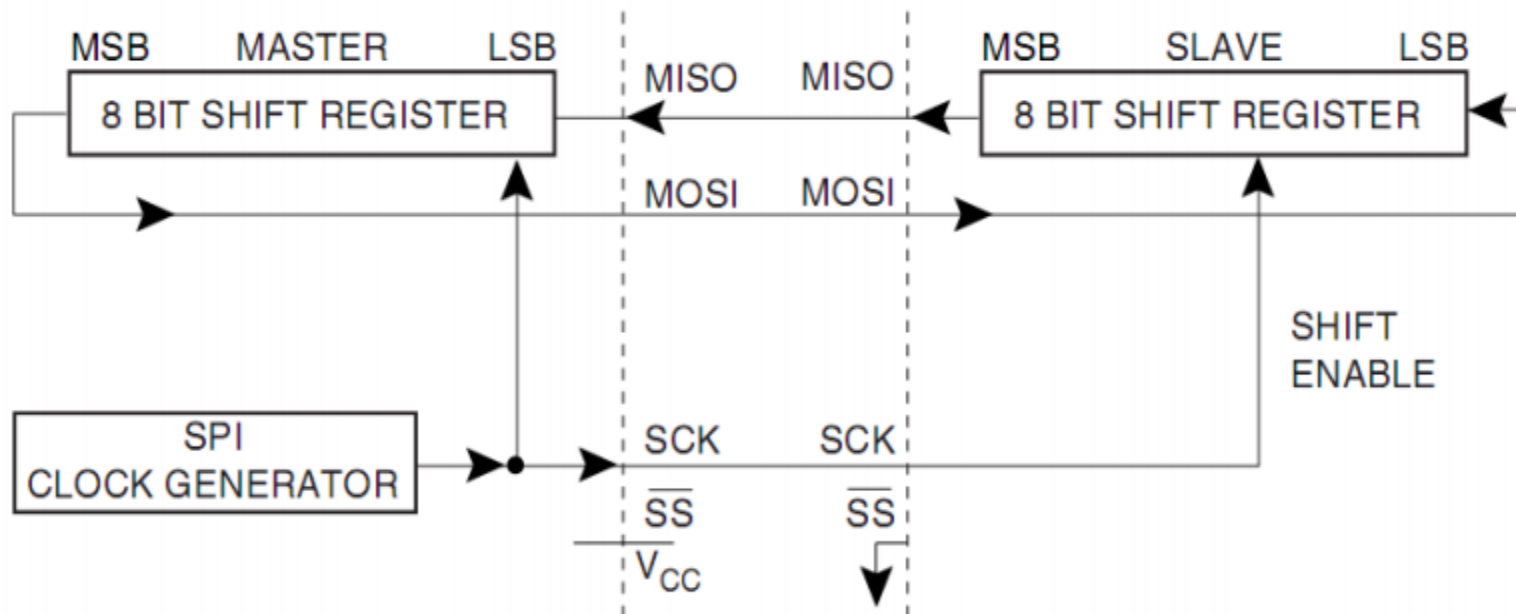
Modul de functionare. Interfata SPI.

- Master initiaza comunicatia prin activarea SS
- Master genereaza semnalul de ceas SCLK
- Pe fiecare perioada de ceas un bit se transmite de la master la slave, si un bit de la slave la master
- Dupa fiecare pachet de date (8, 16 biti,...) SS este dezactivat, pentru sincronizarea transmisiei

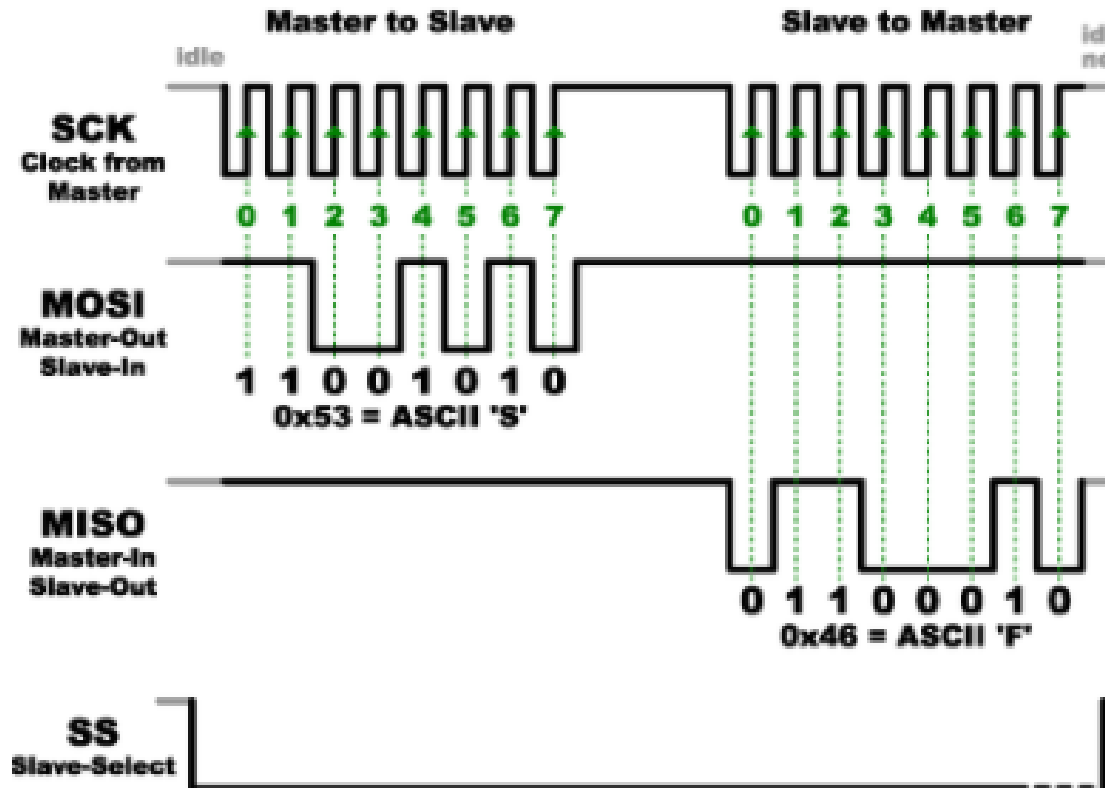


Modul de functionare. Interfata SPI.

- Ambii parteneri au cate un registru de deplasare intern, iesirile si intrarile fiind conectate prin MISO/MOSI
- Ambii registri au acelasi ceas, SCLK
- Cei doi registri formeaza impreuna un registru de rotatie
- Dupa un numar de perioade de ceas egal cu dimensiunea unui registru, Master si Slave fac schimb de date



Sincronizarea datelor. Interfata SPI.



Exemple de programare. Interfata SPI.

```
#include <SPI.h>
#include <SoftSPI.h>

// Create a new SPI port with:
// Pin 2 = MOSI,
// Pin 3 = MISO,
// Pin 4 = SCK
SoftSPI mySPI(2, 3, 4);

void setup() {
  mySPI.begin();
  Serial.begin(9600);
}

void loop() {
  static uint8_t v = 0;

  Serial.print("Sending value: ");
  Serial.print(v, HEX);
  uint8_t in = mySPI.transfer(v);
  Serial.print(" Got value: ");
  Serial.print(in, HEX);
  Serial.println(v == in ? " PASS" : " FAIL");
  delay(1000);
  v++;
}
```

```
// include the SPI library:
#include <SPI.h>
// set pin 10 as the slave select for the digital pot:
const int slaveSelectPin = 10;
void setup() {
  // set the slaveSelectPin as an output:
  pinMode(slaveSelectPin, OUTPUT);
  // initialize SPI:
  SPI.begin(); }
void loop() {
  // go through the six channels of the digital pot:
  for (int channel = 0; channel < 6; channel++) {
    // change the resistance on this channel from min to max:
    for (int level = 0; level < 255; level++) {
      digitalPotWrite(channel, level);
      delay(10); }
    // wait a second at the top:
    delay(100);
    // change the resistance on this channel from max to min:
    for (int level = 0; level < 255; level++) {
      digitalPotWrite(channel, 255 - level);
      delay(10);
    } } }
void digitalPotWrite(int address, int value) {
  // take the SS pin low to select the chip:
  digitalWrite(slaveSelectPin, LOW);
  delay(100);
  // send in the address and value via SPI:
  SPI.transfer(address);
  SPI.transfer(value);
  delay(100);
  // take the SS pin high to de-select the chip:
  digitalWrite(slaveSelectPin, HIGH);
}
```

Interfata CAN.

Magistrala CAN (Controller Area Network) este o magistrală serială utilizată în industria de automobile, cu scopul de a asigura comunicarea între mai multe microcontrolere fără utilizarea unui calculator-gazdă. Dezvoltată inițial de către firma Bosch, în anul 1983, specificația a fost lansată oficial în anul 1986 (CAN 1.2) și standardizată sub denumirea de ISO 11898. Ulterior, mai mulți producători de semiconductoare (Intel, Philips, Infineon, Texas Instruments, Motorola) au implementat periferice pe bază de CAN. În septembrie 1991, Bosch lansează versiunea a 2-a a specificației (CAN 2.0).

Pe lângă industria de automobile (sisteme de frânare, o gamă largă de senzori, lampi de semnalizare, controlul automat al ușilor) protocolul CAN a început să fie utilizat cu succes și în alte ramuri ale electronicii industriale (echipamente medicale, războaie de țesut).

Principial, diferențele dintre versiunea 1.2 și 2.0 a standardului, constau în domeniul de adresare a nodurilor, care a fost extins în noua versiune. Mai exact, CAN 1.2 definește doar un singur tip de mesaj (mesaj standard) având lungimea câmpul de identificare a nodului (Id) de 11 biți, pe când versiunea CAN 2.0 mai introduce, pe lângă tipul de mesaj definit anterior și un mesaj cu lungimea Id-ului de 29 de biți numit mesaj extins.

Interfata CAN.

Specificația de CAN definește mai multe nivele:

nivelul fizic - descrie modul de transmitere a semnalului pe magistrală (reprezentare unui bit, nivele de transmisie a semnalelor, aspecte legate de mediul de transmisie)

nivelul transfer - descrie tipurile de mesaje trimise/recepționate de un nod de la nivelul sau superior (obiect); tot în grija acestui nivel țin și aspectele legate de durata unui bit, sincronizare, formatul mesajelor, tehnici de arbitrare, confirmare, detecție de erori precum și mecanisme de restrângere a perturbațiilor

nivelul obiect - se ocupă cu aspecte ce țin de filtrarea și manipularea mesajelor nivelul aplicație

Tipurile de cadre

Transferul de mesaje se manifestă și totodată este controlat prin patru tipuri diferite de cadre:

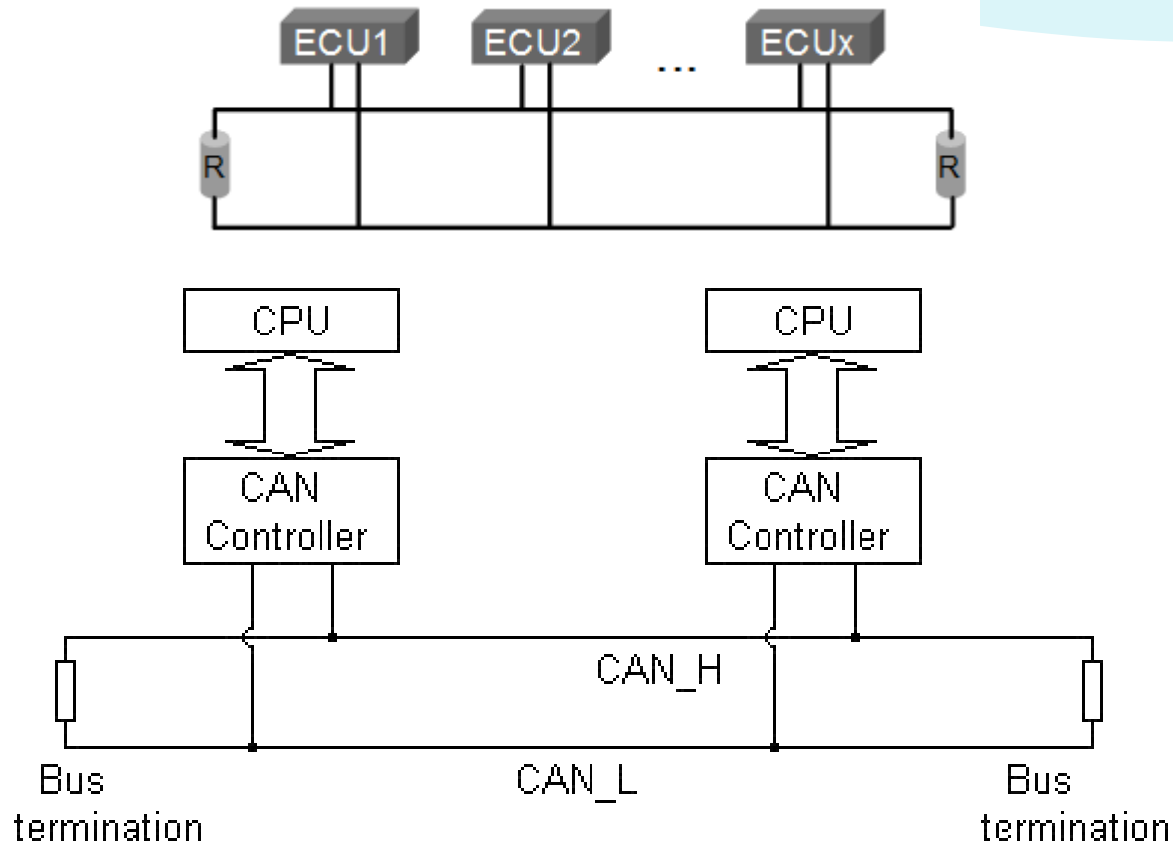
Cadrul de date (Data Frame) - transportă date de la transmițător la receptor

Cadru de solicitare (Remote Frame)- cadru de solicitare a unui cadru de date

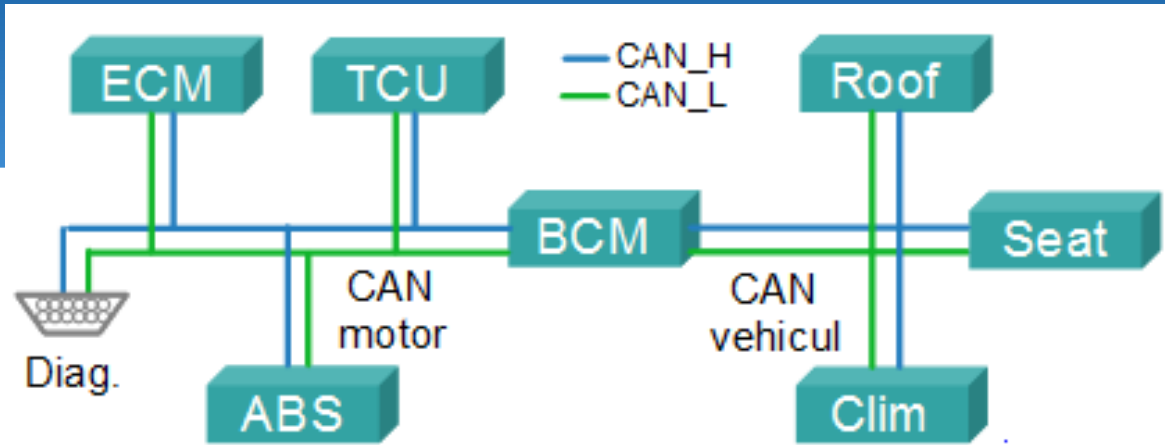
Cadrul de eroare (Error Frame) - transmis de fiecare nod la detecția unei erori pe magistrală

Cadru de supraîncărcare (Overload Frame) - solicită un timp suplimentar între cadrul anterior și cel următor

Nivelul fizic al protocolului CAN.



Exemplu de retea CAN.



ECM (Engine Control Module) – calculatorul de injecție (motor)

TCU (Transmission Control Unit) – calculatorul transmisiei automate

ABS (Anti-lock Braking System) – calculatorul sistemului de frânare

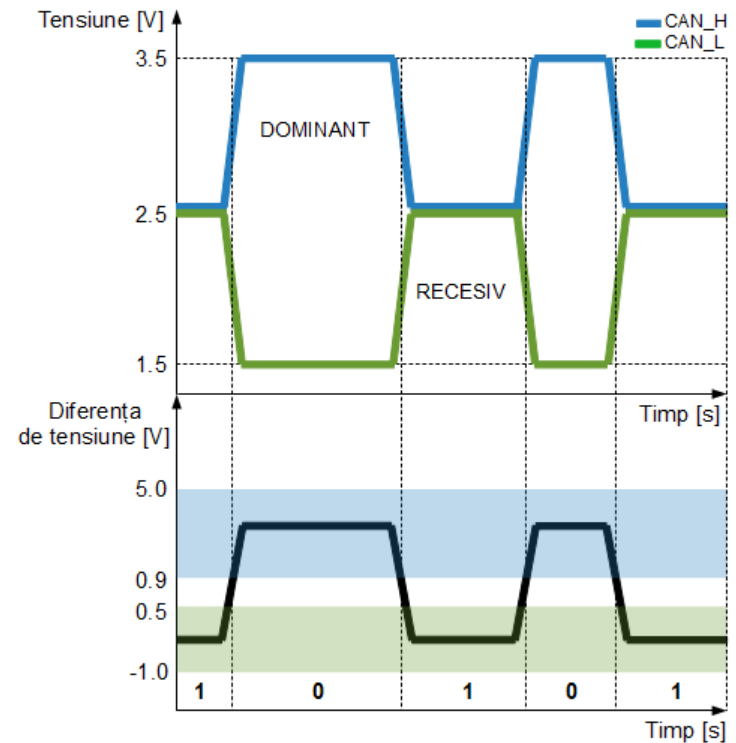
BCM (Body Control Module) – calculatorul de habitacul

Roof (Plafon) – calculatorul pentru controlul trapei

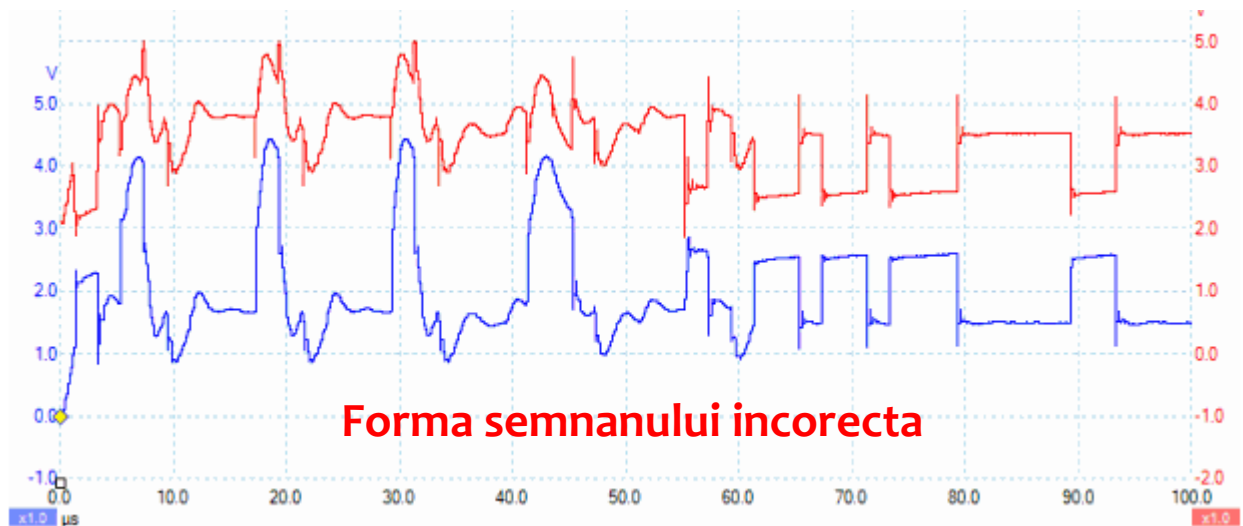
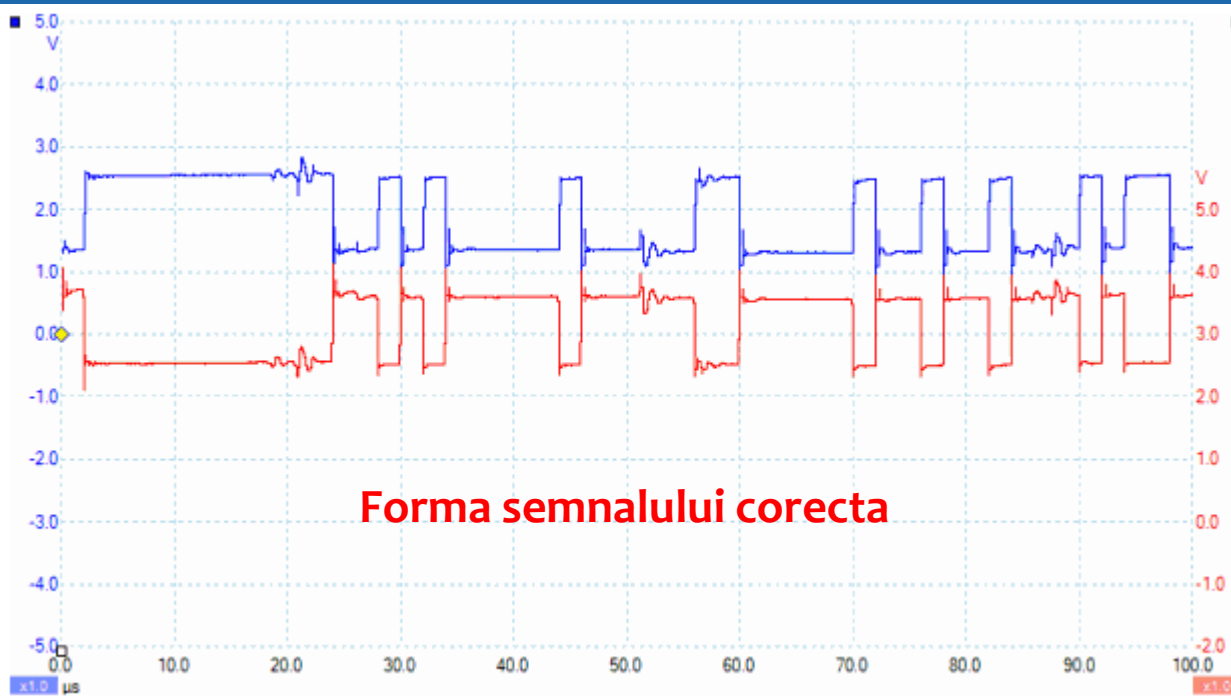
Seat (Scaun) – calculatorul pentru controlul scaunelor

Clim (climatizare) – calculatorul pentru controlul climatizării

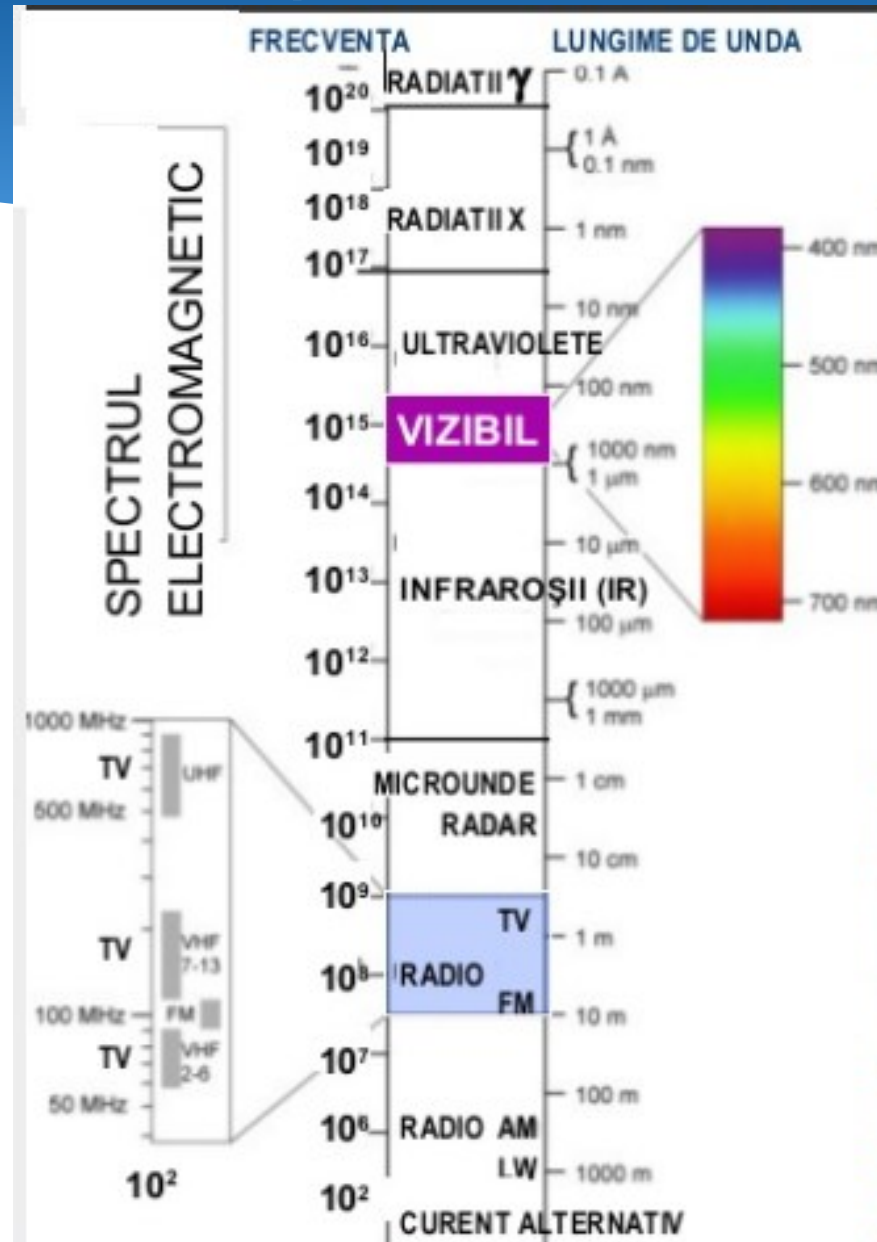
Diag. (diagnostic) – conectorul de diagnosticare



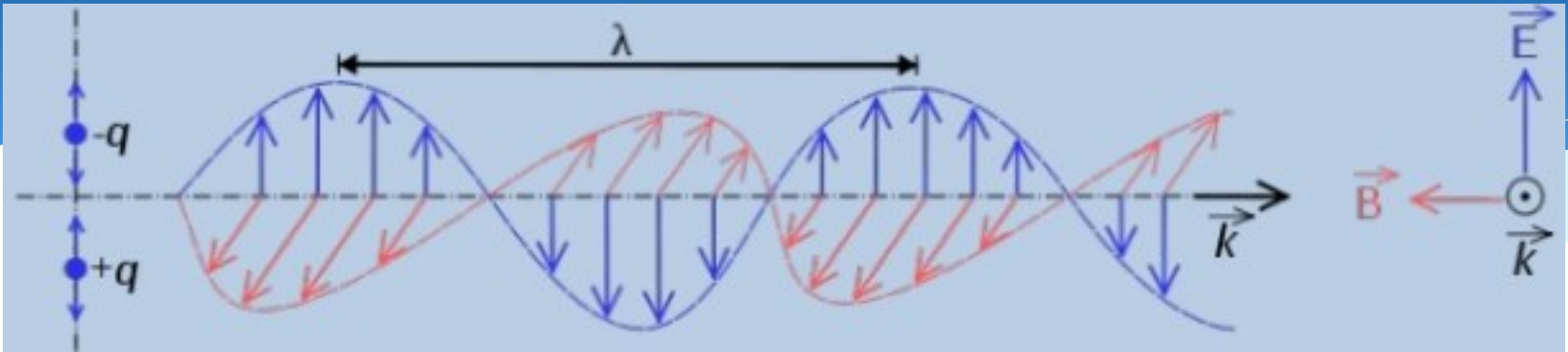
Fiabilitatea rețelei CAN.



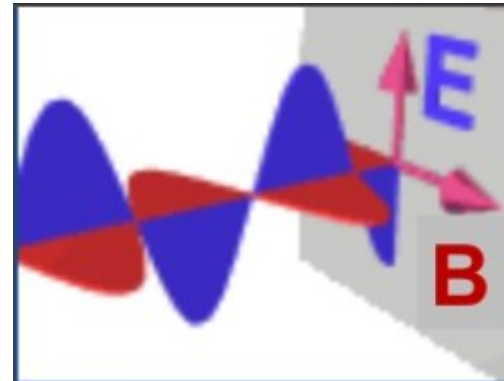
Comunicare Wireless. Spectrul undelor electromagnetice.



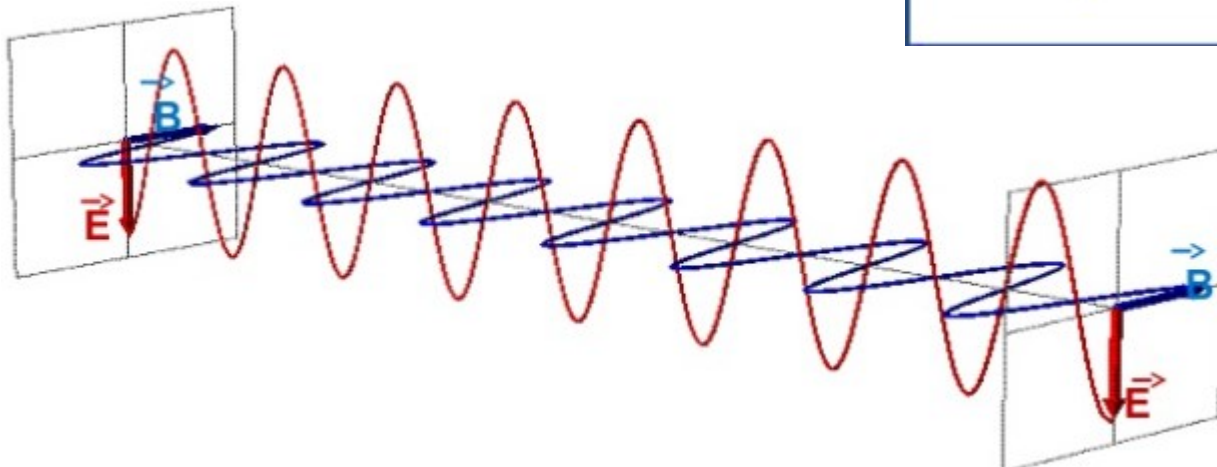
Comunicare Wireless. Propogarea undelor electromagnetice.



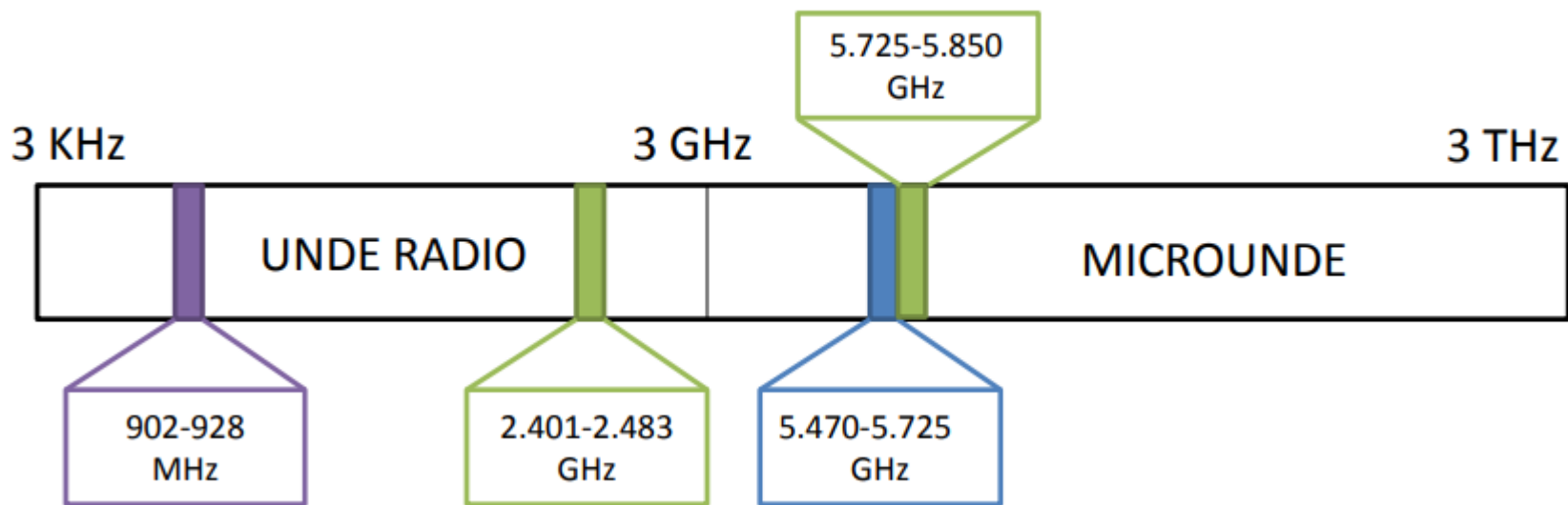
Propogarea spatiala a UEM



Vectorul intensitate \vec{E} a câmpului electric și vectorul inducție \vec{B} a câmpului magnetic



Benzile de unde utilizate pentru Transferul de Date.



Interfata IrDA.

Tehnologia IrDA (Infrared Data Association) definește un set de standarde al tehnologiei fără fir bazat pe comunicații prin infraroșu de transmitere și recepție de date pe distanță mică. IrDA a fost fondată la 28 iunie 1993 de un grup format din 50 de companii cu scopul de a standardiza comunicațiile în infraroșu.

Tehnologia IrDA este implementată în dispozitive portabile precum smartphone, laptop, PDA, camere video, periferice, telecomenzi, aparate medicale și industriale, auto etc. Primul standard, bazat pe portul serial RS-232 a fost aprobat în 1994. Acest standard folosește specificațiile portului serial, aceeași structură de date dar și limitele de viteză. În 1995 a fost aprobat un nou standard cu limita de viteză la 1Mbps.

În principal, tehnologia IrDA utilizează modulația ASK (Amplitude-shift keying) care este o formă de modulație a amplitudinii unui semnal analogic cu un semnal modulator reprezentat de un flux de biți.

Caracteristicile principale ale acestui tip de comunicații wireless, sunt transferul de date securizat, și rata de eroare foarte redusă, ceea ce îl face foarte eficient.

Dispozitivele IrDa folosesc LED-uri infraroșii pentru a emite radiație infraroșie care este focalizată într-o rază îngustă. Raza este modulată, pornită sau oprită, pentru a codifica datele. Receptorul folosește o fotodiodă pentru a converti radiația infraroșie în curent electric.

Pentru ca dispozitivele să comunice, trebuie să fie așezate în linie unul cu celălalt pe direcția de transmisie a luminii de infraroșu.

Suita de protocoale IrDA are rate de transfer de până la 16 Mbps pe o distanță în jur de aproximativ 1m, iar la distanța de 5m rata de transfer scade foarte mult, ajungând în jurul valorii de 75 kbps.

Interfata IrDA.

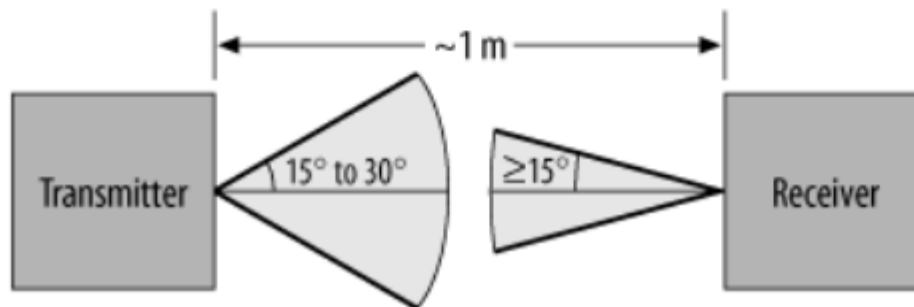
Infrared Data Association

Principalul obiectiv este de a furniza comunicații pe distanțe scurte între 2 dispozitive
Comunicatie 1 la 1

Standardul initial (v1.0) suporta viteze de transfer între 2400 și 15200 bps, pe distanțe de aproximativ 1m.

Comunicatia initiala se realizeaza cu viteza de 9600 bps și dispozitivele negociaza o rata de comunicare pentru transfer, mai mare sau mai mica (depinzand de capabilitatile fiecarui dispozitiv)

Standardul a fost extins să suporte viteze de comunicare mai mari, de 1.152 Mbps și 4 Mbps.



Codificare IrDA.

IrDA foloseste o schema de codare Return-to-Zero (RZ).

In codarea RZ, un cadru este format dintr-un interval de transmisie care este divizat in subintervale reprezentand biti individuali.

Un zero logic este reprezentat de un impuls cu durata de $\frac{3}{16}$ din latimea intervalului pentru un bit

Un unu logic este reprezentat prin absenta unui puls

Impulsul este mai scurt, pentru a economisi energie

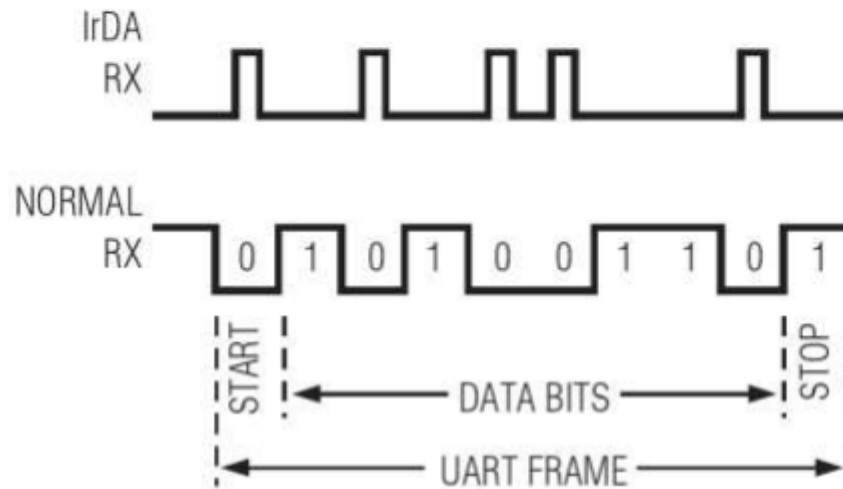
Transmiterea unui sir de zerouri => impulsuri de sincronizare



Codificare IrDA.

Pentru orice baud, pulsul de lumina poate fi ingust de 1.7 microsecunde (pt "0")

$1.7 \text{ us} \Rightarrow 3/16$ din perioada unui bit la 115200

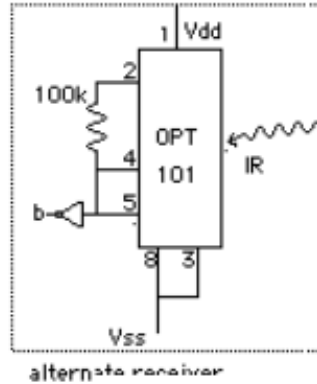
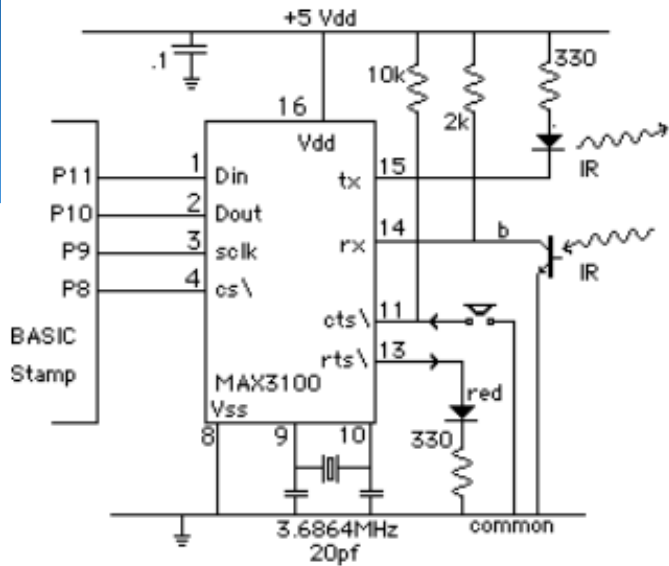


Codificare IrDA.



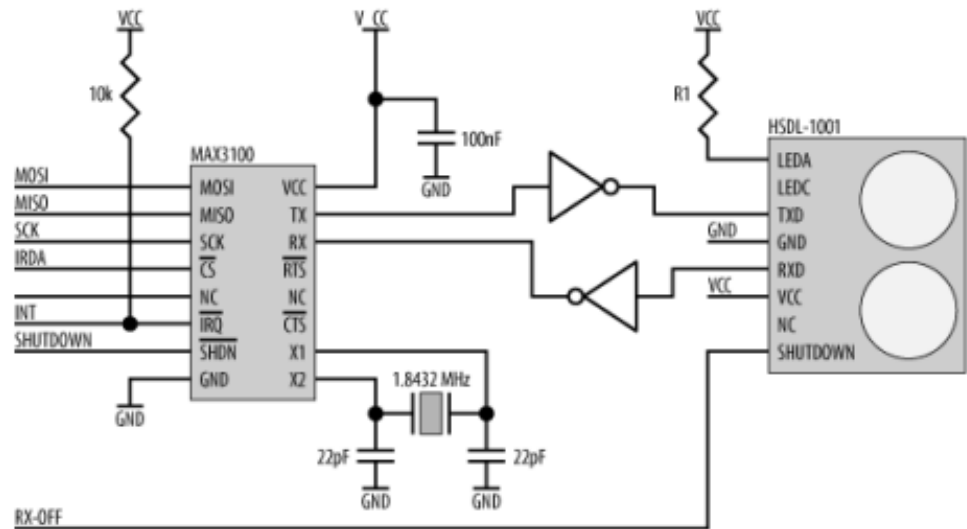
- La viteze de 4 Mbps se folosete PPM (Pulse Position Modulation):
- pozitia pulsului variaza (4 coduri)

Scheme de conectare IrDA.

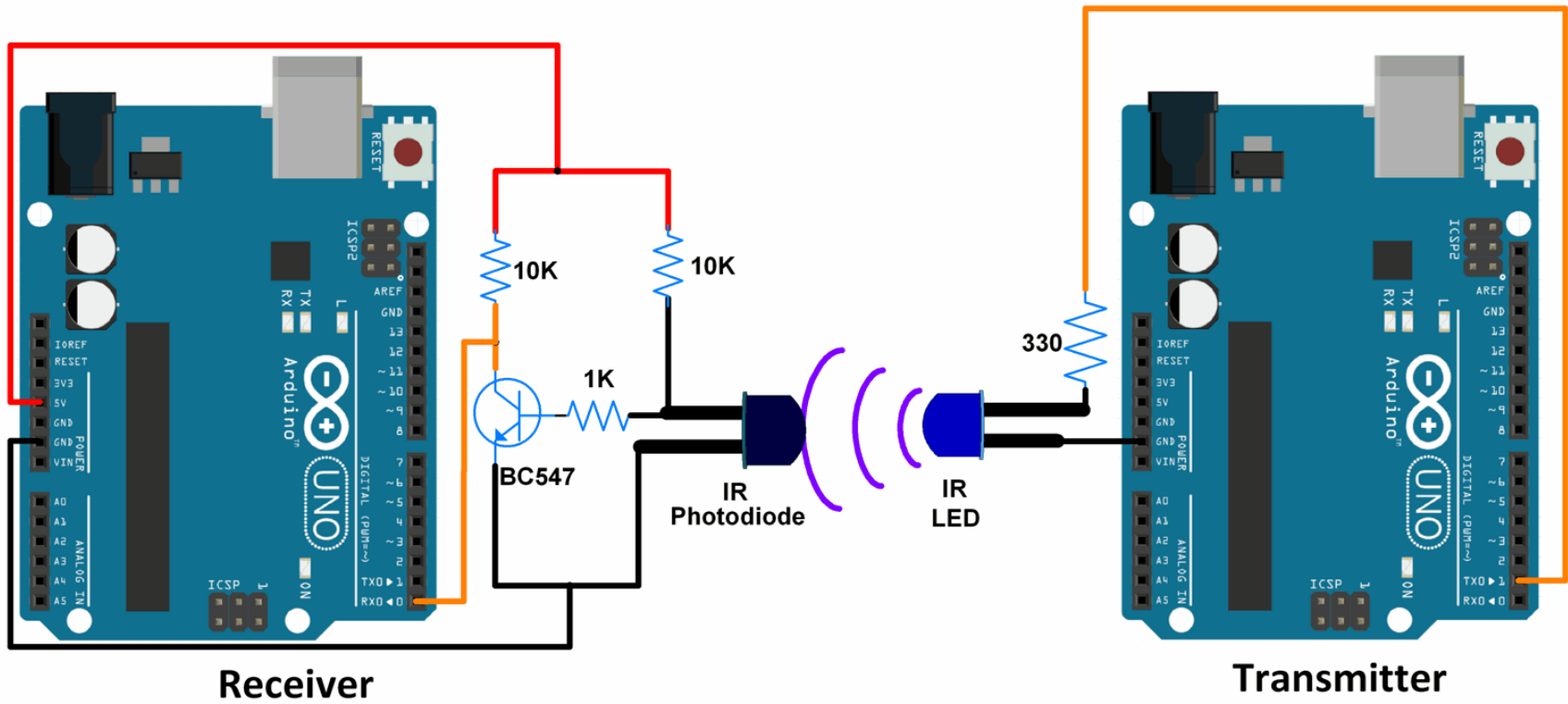


Q: 1.8432 MHz
 (multiplu de 115200
 =115200x16, pentru a
 realiza impulsul de
 3/16)
 R1 – reglaj curent LED IR
 V=2.1, I=250mA

Pentru MAX3100 interfata de
 comunicatie cu uC e SPI,



Scheme de conectare Arduino IrDA.



Programarea conectarea in Arduino IrDA.

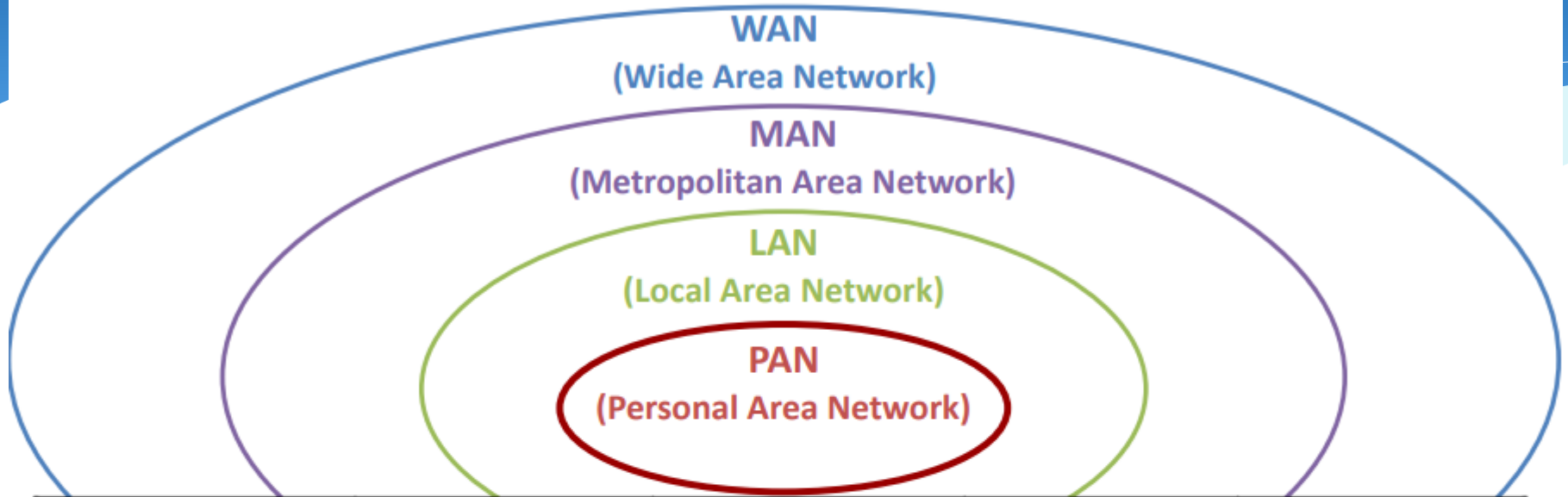
Transmite IrDA

```
void setup() {  
  Serial.begin(9600);      /* Define  
  baud rate for serial communication */  
}  
  
void loop() {  
  int count;  
  for(count = 0; count<100; count++)  
  {  
    Serial.println(count);  
    delay(1000);  
  }  
}
```

Receptie IrDA

```
void setup() {  
  Serial.begin(9600);      /* Define baud  
  rate for serial communication */  
}  
  
void loop() {  
  if(Serial.available())   /* If data is  
  available on serial port */  
  {  
    Serial.print(char(Serial.read())); /* Print  
    character received on to the serial monitor */  
  }  
}
```

Standarde Wireless.



	PAN	LAN	MAN	WAN
Standards	802.15, Bluetooth	802.11a, 802.11b, 802.11g, 802.11n	802.16 (WiMax) MMDS, LMDS	GSM, GPRS, CDMA, 2.5–3G
Speed	1 - 3 Mbps	1 – 300 Mbps	2 - 75 Mbps	0.384 - 10 Mbps
Range	Short	Medium	Medium–long	Long
Applications	Peer to peer, device to device	Enterprise networks	Fixed, last-mile access	PDAs, mobile phones, cellular access

Proprietati ale Undelor Electromagnetice.

Frecvența	Frecvențe mici	Frecvențe mari
Distanța de propagare	Mare	Mică
Interferențe	Mari	Mici
Costuri echipamente	Mici	Mari
Influența asupra organismelor	Mici	Mari

Interfata Bluetooth.

Bluetooth este un set de specificații (un standard) pentru o rețea personală PAN) fără fir (*wireless*), bazată pe unde radio. Bluetooth mai este cunoscut ca și standardul IEEE 802.15.1. Prin tehnologia Bluetooth se elimină firele și cablurile între dispozitive atât staționare cât și mobile, facilitează atât comunicațiile de date cât și pe cele vocale și oferă posibilitatea implementării unor rețele ad-hoc și a sincronizării între diverse dispozitive.

Specificația Bluetooth a fost formulată pentru prima dată în 1994 de Sven Mattisson și Jaap Haartsen, muncitori în orașul Lund, Suedia, la divizia de telefonie mobilă a companiei Ericsson. La 20 mai 1998 a fost fondată gruparea Bluetooth Special Interest Group (SIG), care are rolul de a gestiona tehnologia Bluetooth și de a urmări evoluția acestei tehnologii.

1994: crearea standardului de către de Sven Mattisson și Jaap Haartsen la compania Ericsson

1998: IBM, Intel, Nokia și Toshiba sunt partenere cu Ericsson pentru formarea grupului Bluetooth Special Interest Group (SIG)

1999: apare specificația 1.0, apoi 1.0B

1999: Ericsson a lansat primul telefon dotat cu Bluetooth, modelul Ericsson T39 [\[2\]](#)

2006: cea de-a doua generație Bluetooth v2.0 (apoi V2.1 în 2007). Noul standard a inclus și tehnologia ultrawideband UWB

2009: standardul Bluetooth 3.0 și varianta HS (High Speed),

2010: apariția Bluetooth 4.0 mai puternică și mai puțin consumatoare de energie, Bluetooth Low Energy, (BLE) sau Wibree.

2013: lansarea versiunii 4.1

2016: lansarea versiunii 5.

Interfata Bluetooth.

Lista de aplicații ale tehnologiei Bluetooth include:

Dispozitive mobile: Comunicare wireless cu smartphone-uri iOS și Android, tablete, dispozitive de fitness. Rețea wireless între PC-uri într-un spațiu închis și unde este necesară o lățime de bandă mică. Comunicații wireless cu dispozitive de intrare și ieșire PC, cele mai frecvente fiind mouse, tastatură, imprimantă, difuzoare wireless portabile, căști audio, USB, GPS.

Audio și divertisment: MP3 player, controlul wireless al consolelor de jocuri (Nintendo Wii și Sony PlayStation 3 folosesc tehnologia Bluetooth),

Industrie: au fost dezvoltate produse industriale bazate pe Bluetooth care sunt utilizate într-o mare varietate pentru automatizarea industrială și comunicarea fără fir între diferitele componente ale mașinilor.

Industria auto: comunicare wireless între telefoane mobile cu transmițătoare GSM încorporate, sistemul stereo auto compatibil Bluetooth.

Medicină: Monitoarele de glucoză din sânge, pulsometrele, inhalatoarele pentru astm și alte dispozitive medicale care pot fi purtate, utilizează tehnologia Bluetooth pentru a ajuta la administrarea medicamentelor, la diagnosticarea leziunilor și la transmiterea în siguranță a informațiilor critice de la pacienți la furnizori.

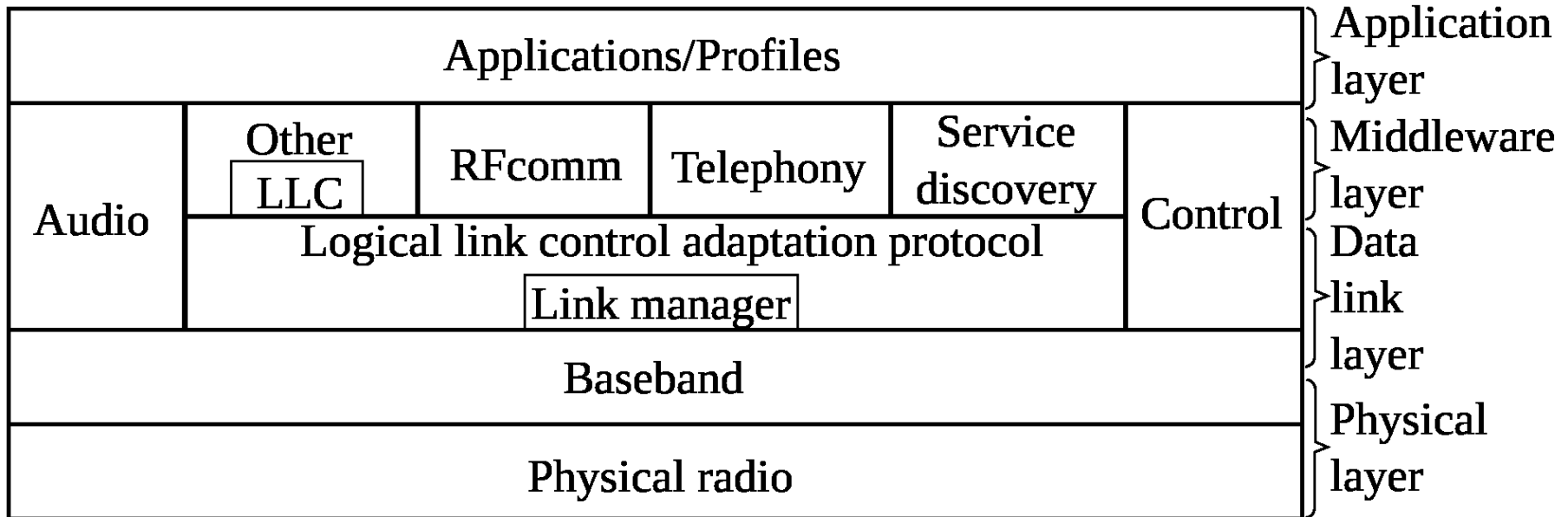
Domotică: Bluetooth permite controlul automat, centralizat al sistemelor esențiale ale unei clădiri, incluzând încălzirea, ventilația și aerul condiționat, iluminatul, sistemele de securitate.

Interfata Bluetooth.

Printr-o rețea Bluetooth se poate face schimb de informații între diverse aparate precum telefoane mobile, *laptop*-uri, calculatoare personale, imprimante, camere foto și video digitale sau console video prin unde radio criptate (sigure) și de rază mică, desigur numai dacă aparatele respective sunt înzestrate și cu Bluetooth. Aparatele care dispun de Bluetooth comunică între ele atunci când se află în aceeași rază de acțiune. Ele folosesc un sistem de comunicații radio, așa că nu este nevoie să fie poziționate față în față pentru a transmite; dacă transmisia este suficient de puternică, ele pot fi chiar și în camere diferite.

Clasa	Puterea maximă permisă (mW)	Puterea maximă permisă (dBm)	Raza (aproximativă)
Clasa 1	100 mW	20 dBm	~100 metri
Clasa 2	2,5 mW	4 dBm	~10 metri
Clasa 3	1 mW	0 dBm	~1 metru

Interfata Bluetooth.



Standarde IEEE 802.15.

Standardele IEEE 802.15, in ordinea apariției lor, sunt prezentate comparativ in tabelul de mai jos:

Standardul IEEE		Volumul de date transferat	Banda de frecvență	Aria de acoperire	Aplicații
802.15.1	Bluetooth	1Mbps	2.4-2.48GHz	10m	Telefoane celulare, calculatoare, PDA-uri, imprimante, microfoane și alte dispozitive digitale
802.15.3	High-rate WAPN	>20Mbps	3.1-10.6GHz	10m	Transfer multimedia, radar de înaltă rezoluție, senzor wireless de rețea, sisteme de radio-locății etc.
802.15.4	Low-rate WAPN	<0,25Mbps	868MHz 902-928MHz 2.4-2.48GHz	100m	Aplicații in domeniile industrial, agrar, medical etc.

BLE. Bluetooth Low Energy.

Rolul dispozitivelor in topologia unei retele BLE

*Master
Client*

*Can read/write data to
Slave/Server*



Central



Peripheral

Slave

Server

Has read/write data

Can receive broadcast data



Observer



Broadcaster

Has read-only broadcast data

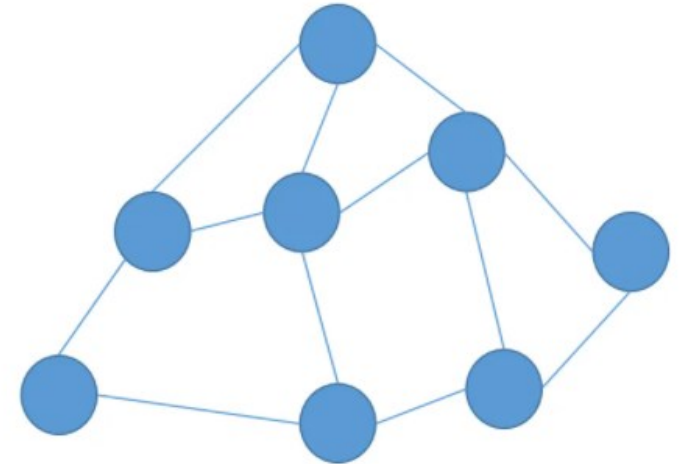
BLE. Topologii de retea.



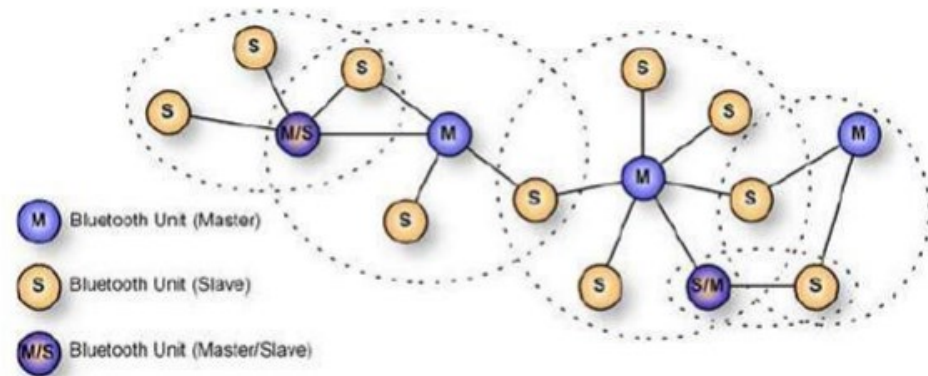
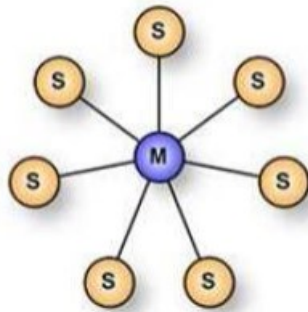
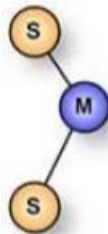
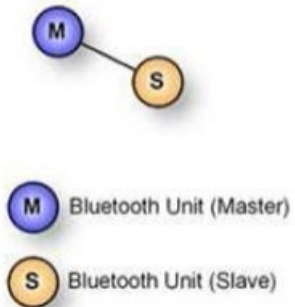
Punct la Punct



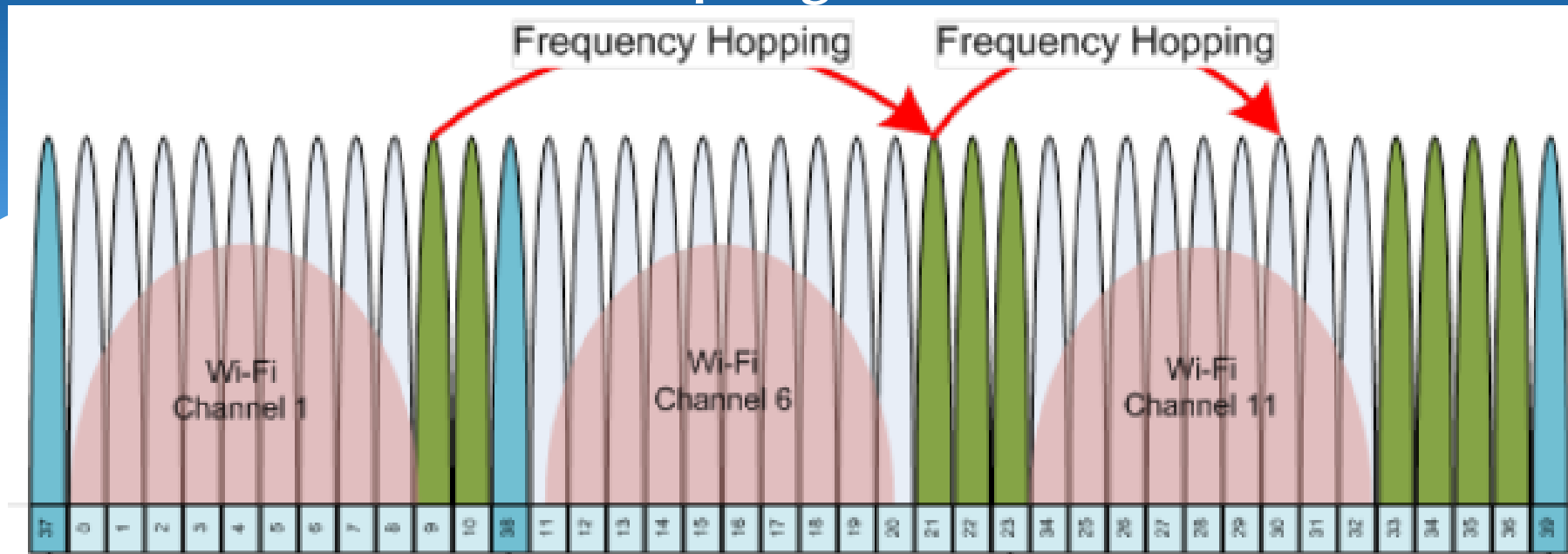
Broadcast



Mesh



BLE. Topologii de retea.



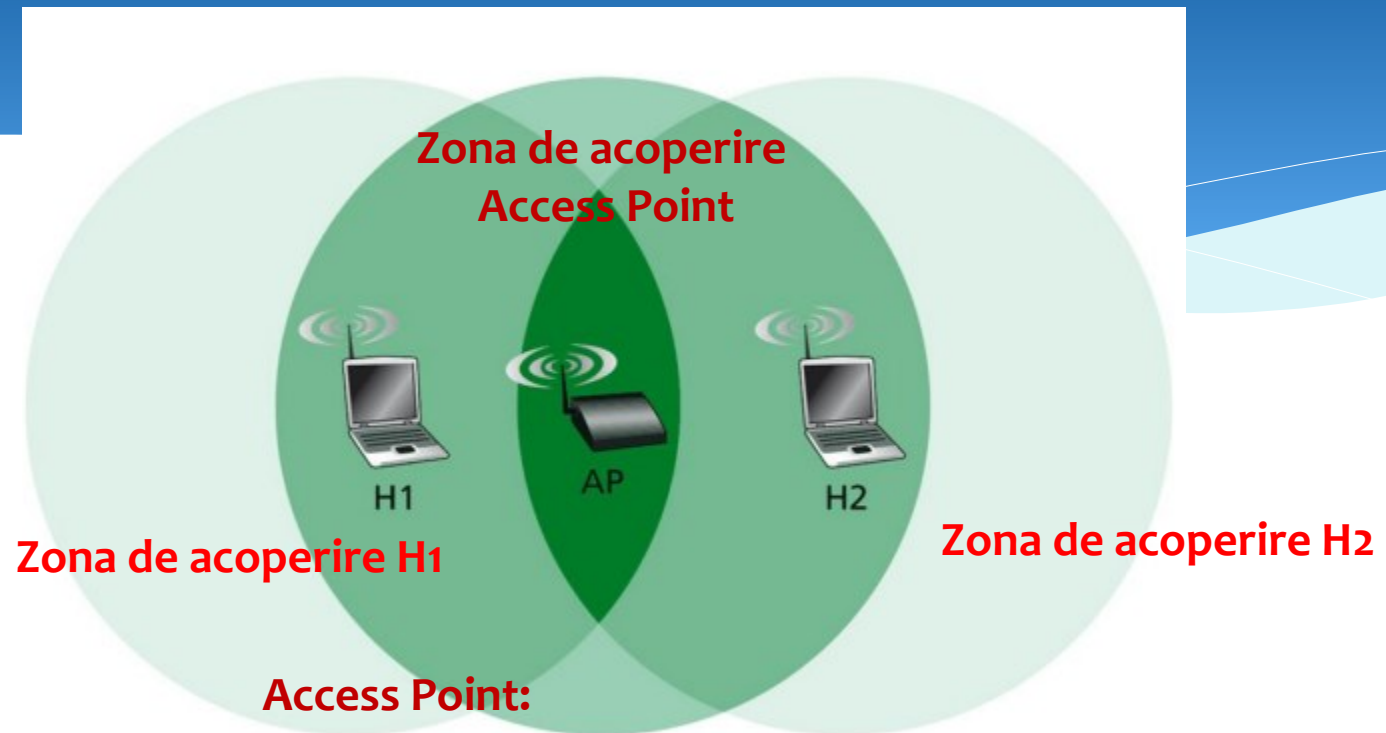
Tehnica de transmisie cu spectru împrăștiat prin salt de frecvență (Frequency-Hopped Spread-Spectrum – FH-SS)

- semnalul modulat FH-SS poate fi privit ca un semnal modulat de bandă îngustă transmis pe o purtătoare a cărei frecvență se schimbă la intervale constante de timp, luând valori dintr-un set finit de valori posibile numit „hopset”.
- lărgimea de bandă BW_c a semnalului transmis pe un canal al FH-SS rămâne constantă, în timp ce frecvența sa purtătoare se modifică în timp.
- salturile de frecvență sunt utilizate în sisteme de transmisie ale căror benzi de frecvențe sunt împărțite într-un număr finit de canale (sub-benzi), fiecare canal având o lărgime de bandă suficientă pentru a cuprinde cea mai mare parte a puterii semnalului modulat (FSK sau QAM).

Standarde WiFi.

Standard	802.11a	802.11b	802.11g	802.11n
Publicare	1999	1999	2003	2009
Frecvență	5GHz	2.4GHz	2.4GHz	2.4GHz / 5GHz
Viteză	54Mbps	11Mbps	54Mbps	160-600 Mbps
Modulare	OFDM	DSSS	OFDM, DSSS	OFDM
Acoperie Interior Exterior	35m 120m	38m 140m	38m 140m	70m 250m
Avantaje	Semnal puternic pe rază mică	Preț scăzut	Viteza mai mare ca b Compatibil cu b	Acoperire mare Viteză mare
Dezavantaje	Incompatibil cu b și g (mai răspândite)	Interferențe	Interferențe	Standard nou și încă scump

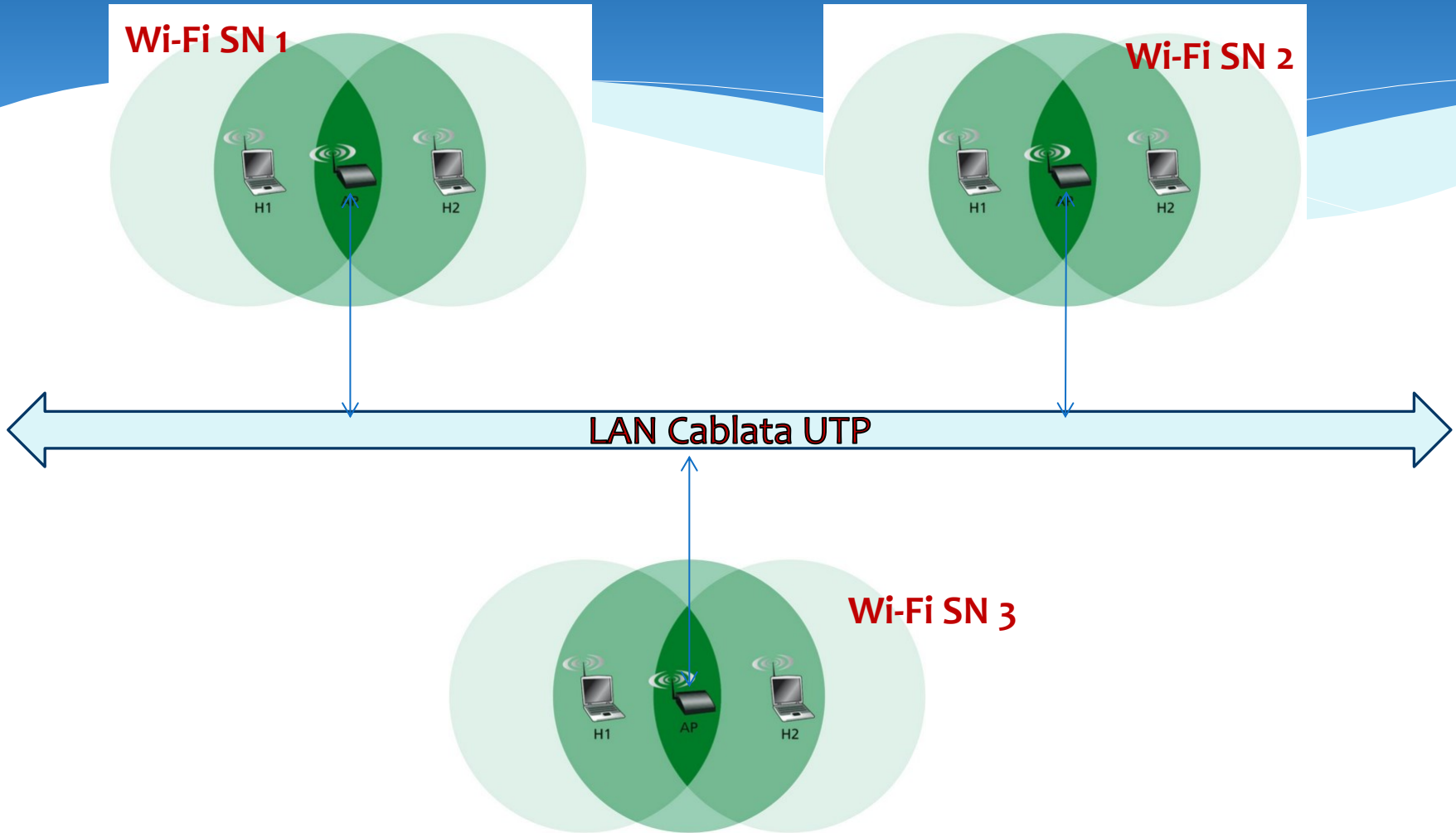
Tehnologia de comunicare WiFi.



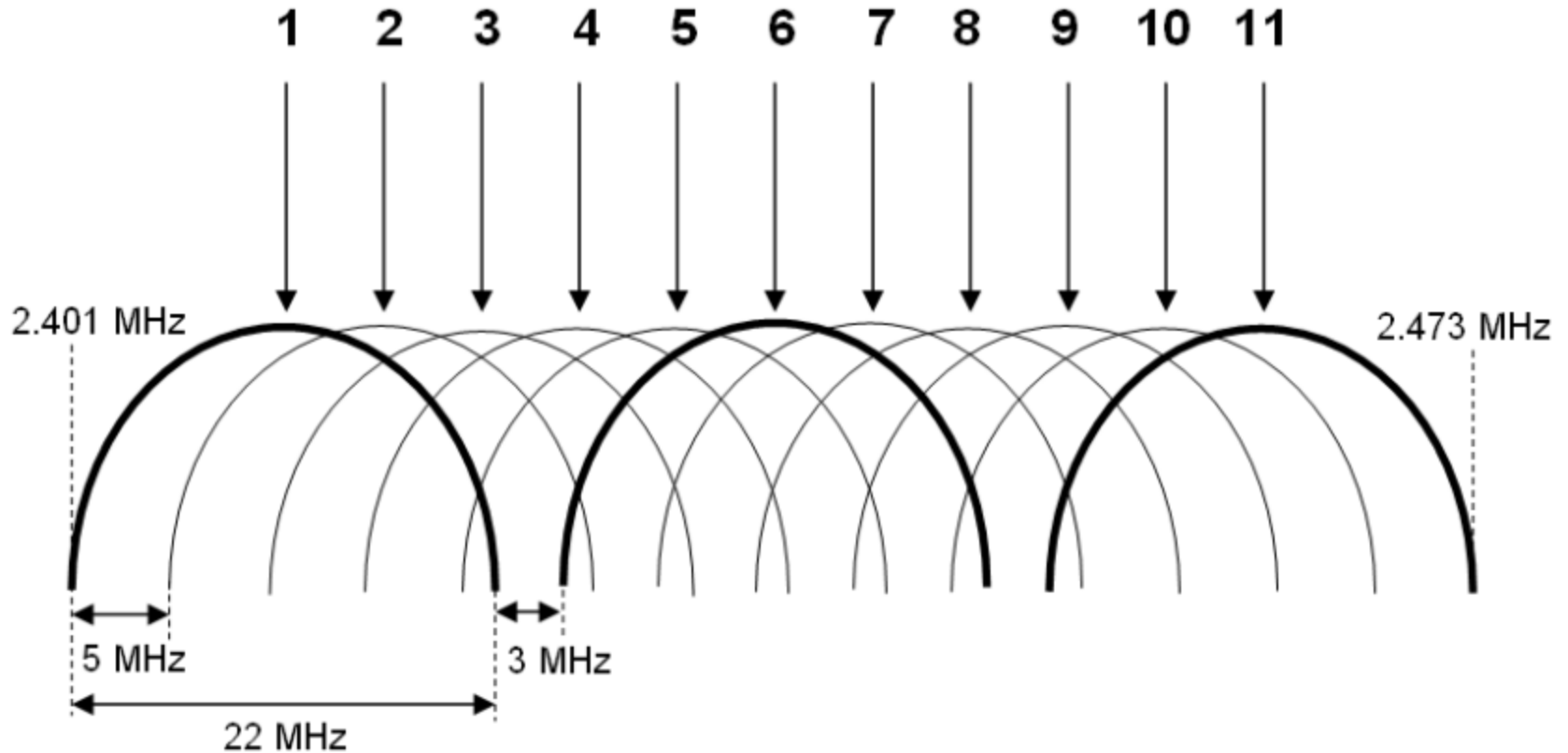
Access Point:

1. DHCP;
2. IP Virtuale;
3. Interconexiune H1 si H2;
4. Conectare la retea Internet;
5. Interconectarea mai multor LAN.

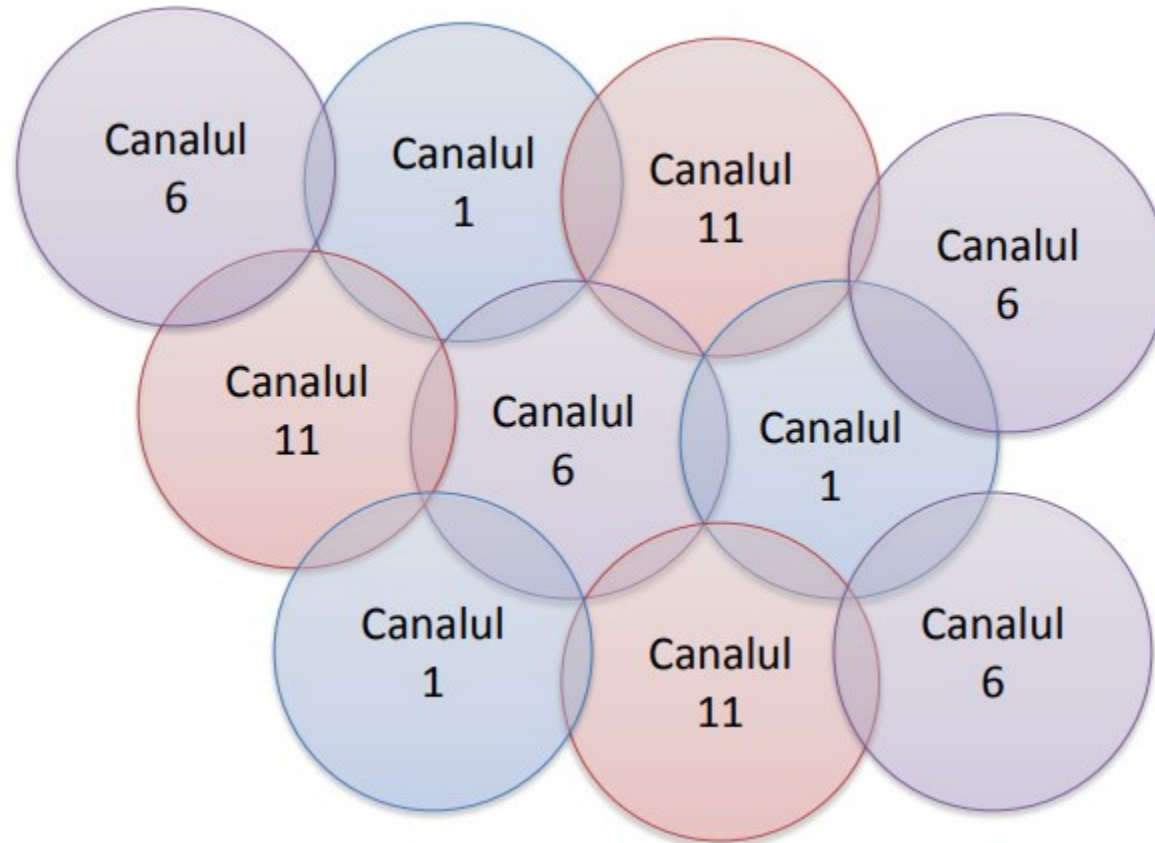
Tehnologia de comunicare WiFi.



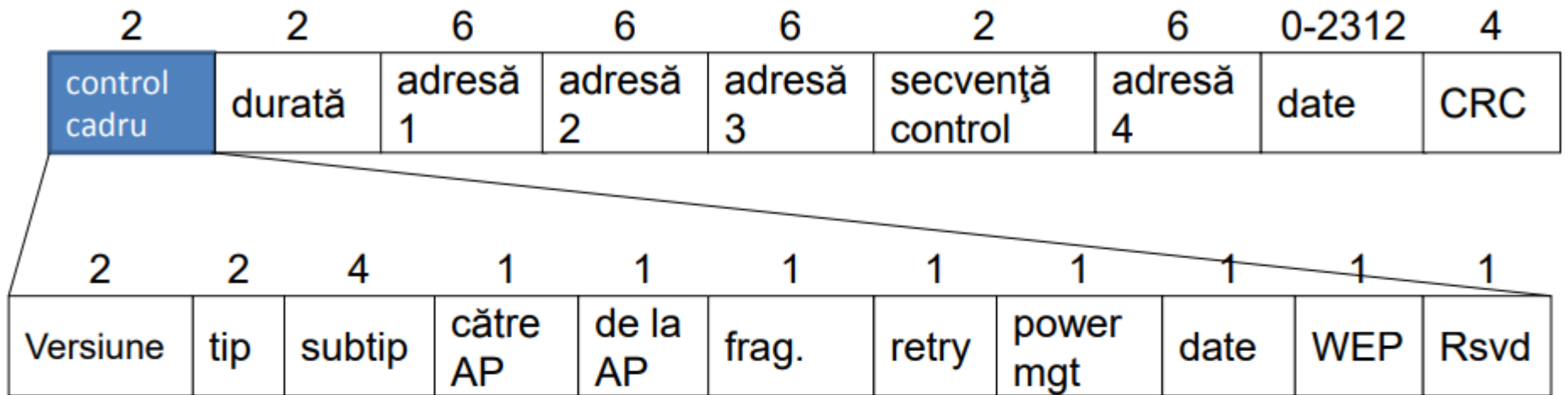
Repartizarea spatiului de frecvente WiFi.



Acoperirea unei suprafete mari folosind un numar redus de canale.

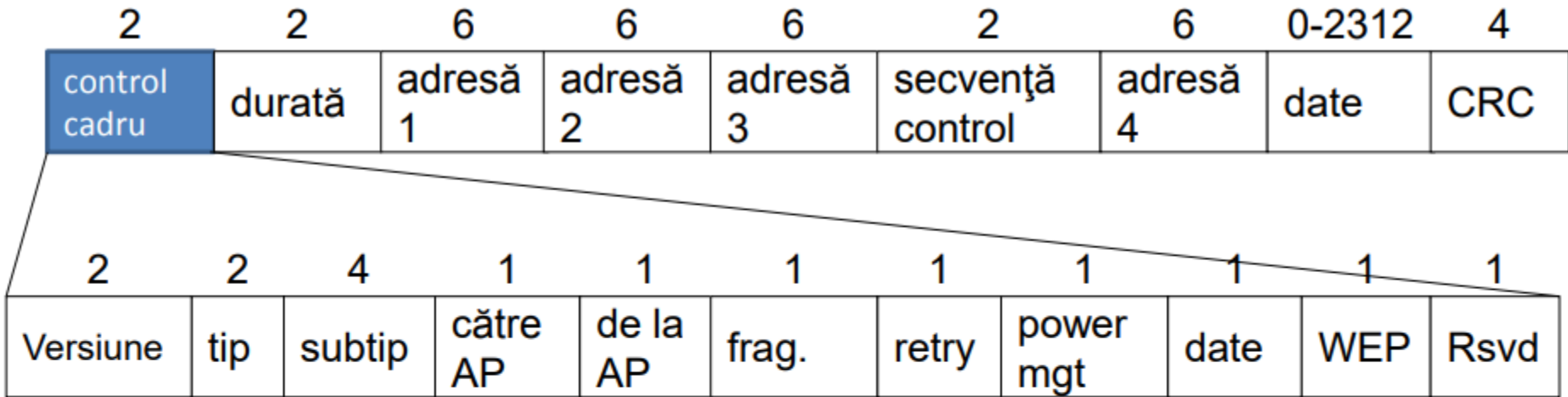


Formatul cadrului 802.11.



- Protocol Version – deocamdată 00
- Type – Management / Control / Data
- Subtype – Association Request, ACK, Data, etc.
- Către AP/de la AP – 1 dacă destinația/sursa este rețeaua cu fir
- Frag. – Mai sunt **fragmente** de transmis
- Retry – retransmisie a unui fragment transmis anterior
- Power mgt – stația va intra în mod consum redus, imediat după transmisie
- More Data – mai sunt frame-uri în bufferul de transmisie
- WEP – cadrul este criptat

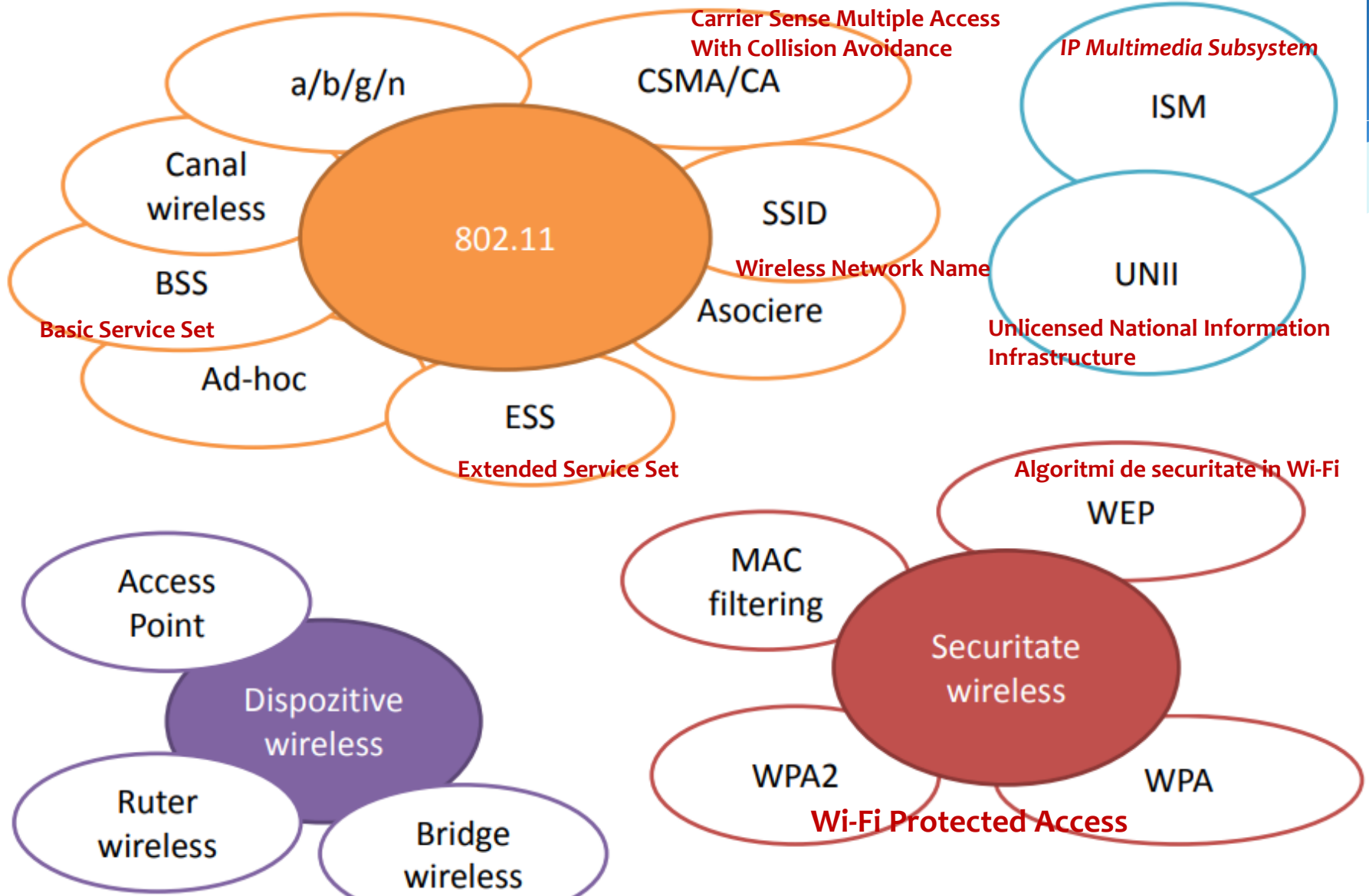
Formatul cadrului 802.11.



către AP	de la AP	adresa 1	adresa 2	adresa 3	adresa 4
0	0	destinație	sursă	BSSID	-
0	1	destinație	BSSID	sursă	-
1	0	BSSID	sursă	destinație	-
1	1	Adresă receptor	Adresă transmițător	destinație	sursă

- Către AP sau DS (Distribution System) = către rețeaua cu fir
- Durată – timpul de transmisie fragment + 1 CTS + 1 ACK + 3 x SIFS
- Control secvență – numărul de secvență al fragmentului
- CRC – suma ciclică de control pe 32 de biți

Definitii in Retele Wireless.



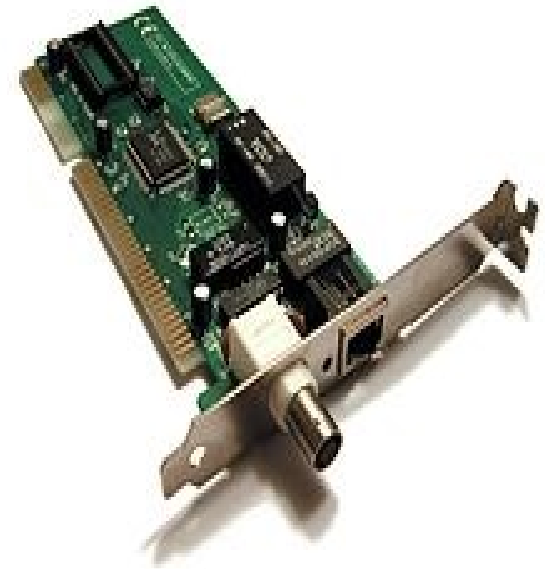
Interfata Ethernet (IEEE 802).

Placa de rețea numită și „adaptor de rețea” sau „placă cu interfață de rețea”, este un circuit imprimat ce permite comunicarea între calculator și rețea (Internet, LAN). Termenul corespunzător este *Network Interface Card*, abreviat *NIC*. I se spune placă, deoarece are aspectul unei plăci cu cablaj imprimat pe care sunt montate componentele electronice necesare conectării și comunicării cu rețeaua. Se montează, de obicei, într-un slot de pe placa de bază a calculatorului.

Prima placă de rețea a apărut odată cu standardul Ethernet în 1979, creat de DEC, Intel și Xerox.

O caracteristică importantă a unei plăci de rețea, este viteza de transfer a datelor (cantitatea maximă de date pe care o poate transfera în unitatea de timp) și se măsoară în Mbit/s (megabit pe secundă):

- 10 Mbit/s – viteză mică, plăci de rețea mai vechi;
- 100 Mbit/s – viteză obișnuită, plăcile de rețea folosite cel mai des;
- 1000 Mbit/s – viteză mai mare de transfer, sunt placile de rețea mai noi;
- > 1000 Mbit/s – plăcile de rețea profesionale.



Interfata Ethernet (IEEE 802).

Standard	Mediu de transmisie	Lungime maxima a segmentului(m)	Cod de linie	Observatie
100Base-T4	CAT 3;4;5; UTP(8 fire)	420	Manchester (BIF-L)	Semnalizari la 25MHz
100Base-TX	CAT 5 UTP(4 fire); CAT 1 STP(4fire)	420	4B5B(125MHz)	Transmisie full duplex la 100 Mbps;half-duplex la 200Mbps
100Base-FX	Cablu de fibra optica (2 fire)	420	RZ	Full duplex la 100Mbps la mare distanta

In prezent, se utilizeaza si standardul **Gigabit Ethernet** sau **GigaEthernet**, echivalent cu IEEE 803z, pentru LAN cu topologie fizica 'star', logica 'bus', 1000 Mbps, in variantele:

1000 Base CX-LAN in cablu coaxial dual de maxim 25 de metri.

1000 Base T-LAN cu CAT 5 UTP, segment de 100 m.

1000 Base SX-LAN, pe fibra optica multimod (830 nm), segmente de cablu de 550 m.

1000 Base LX-LAN, fie pe fibra optica multimod, pe segmente de cel mult 550 metri lungime, fie pe fibra optica unimod cu lungime a segmentelor de pana la 5 km. Transmisia se face pe lungimea de unda de 1270 nm.

Interfata Ethernet. Medii de comunicare.

Standardul EIA/TIA (Electronic Industries Association / Telecommunications Industries Association) 568 cuprinde specificațiile cablului UTP referitor la cablarea clădirilor comerciale.

- Categoria 2 (CAT2) este certificată pentru transmisii de date de până la 4 Mbps. Conține patru perechi torsadate;
- Categoria 3 (CAT3) este certificată pentru transmisii de date de până la 10 Mbps. Conține patru perechi torsadate;
- Categoria 4 (CAT4) este certificată pentru transmisii de date de până la 16 Mbps. Conține patru perechi torsadate;
- Categoria 5 (CAT5) este certificată pentru transmisii de date de până la 100 Mbps. Conține patru perechi torsadate;
- Categoria 5e (CAT5e) este certificată pentru transmisii de date de până la 100 Mbps. Conține patru perechi torsadate. Are mai multe torsadări pe metru decât cel de categoria 5. Este descris de standardul EIA/TIA 568-B. Este cel mai folosit tip de cablu;
- Categoria 6 (CAT6) este certificată pentru transmisii de date de până la 1 Gbps. Conține patru perechi răsucite. Impune specificații mai stricte pentru interferențe (crosstalk) și zgomotul de fundal (system noise);
- Categoria 6A (CAT6A) este certificată pentru transmisii de date de până la 10 Gbps. Conține patru perechi răsucite care pot avea un despărțitor central pentru a separa perechile din interiorul cablului.

Tipul de conector și priză folosit pentru cablul UTP și STP/FTP se numește 8 Position 8 Contact (8P8C).

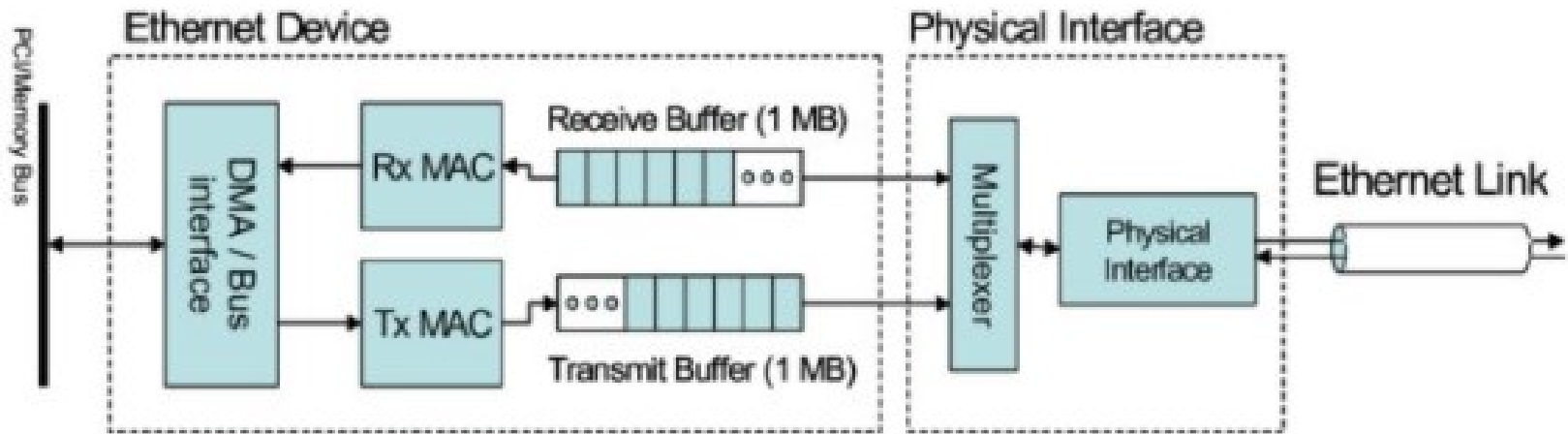
Denumirea mai răspândită este cea de de conector și priză RJ-45. Pentru cablul torsadat UTP sefolosește conectorul RJ-45 necranat, pentru STP și FTP conectorul RJ-45 ecranat

Interfata Ethernet.

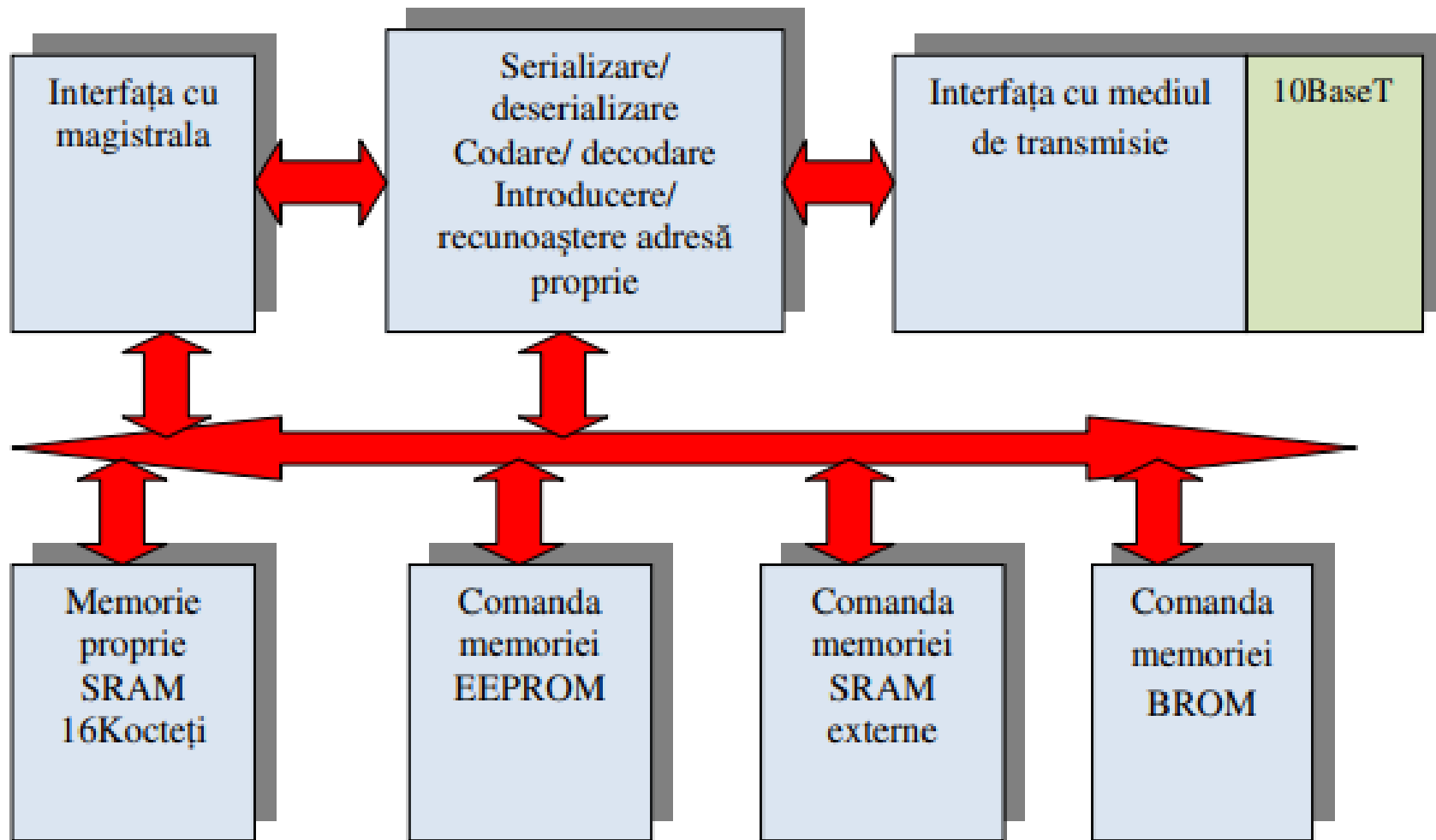
Funcțiile Interfetei Ethernet:

- Conectarea unui PC la rețeaua LAN prin RJ45;
- Controlul și logica de organizare a schimbului de date dintre două dispozitive;
- Interconectarea a două Sub-LAN;
- Adresa IP;
- Adresa MAC;
- Mască;
- Gateway;
- DNS;
- Protocoale de comunicare;
- Protocoale de securizare;
- Protocoale de prezentare a datelor;
- Memorie ROM (Ethernet BIOS);
- Memorie RAM (Bufer Cache);
- Formare codurilor pentru comunicare;
- Formarea semnalelor fizice.

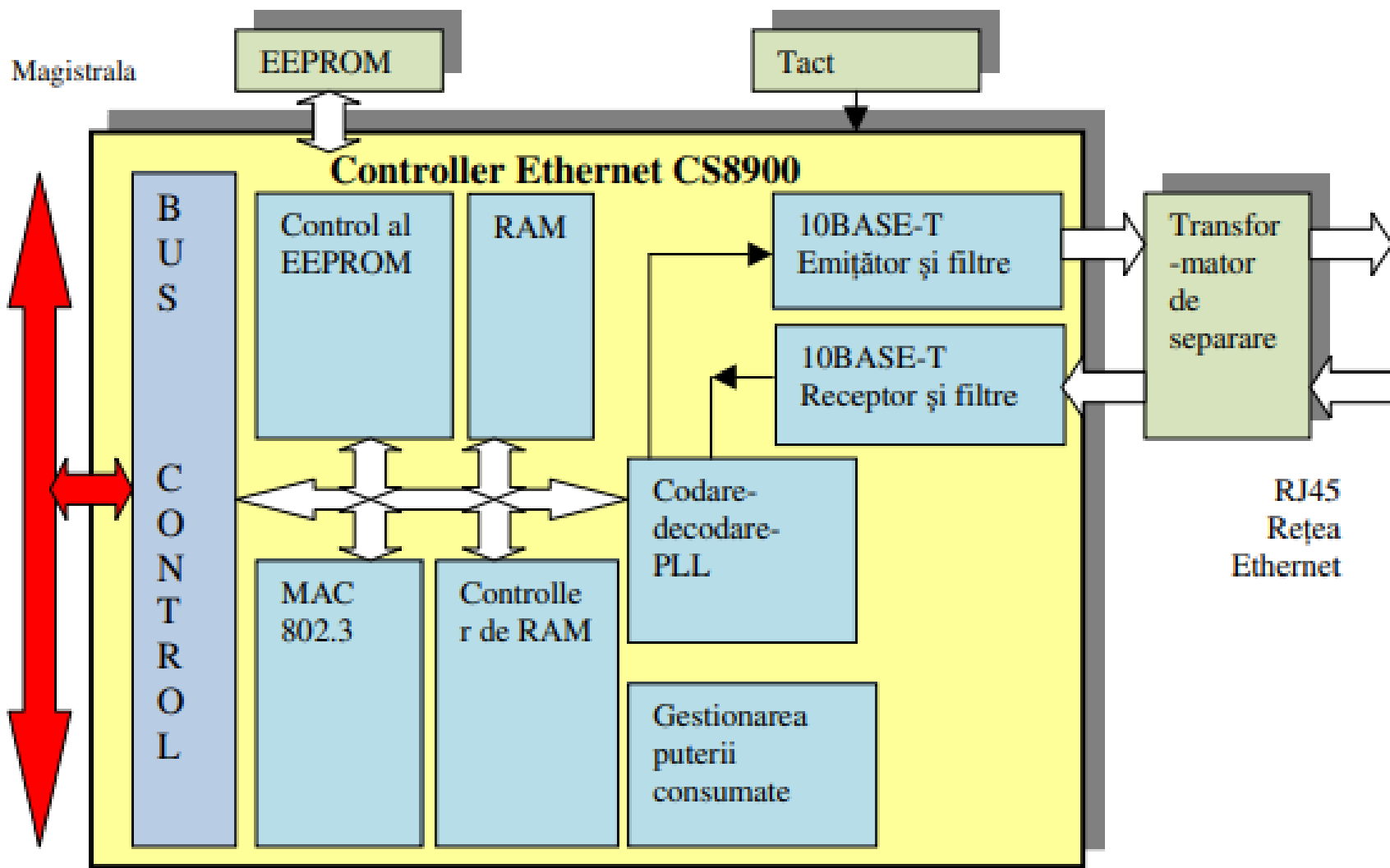
Interfata Ethernet. Structura functionala.



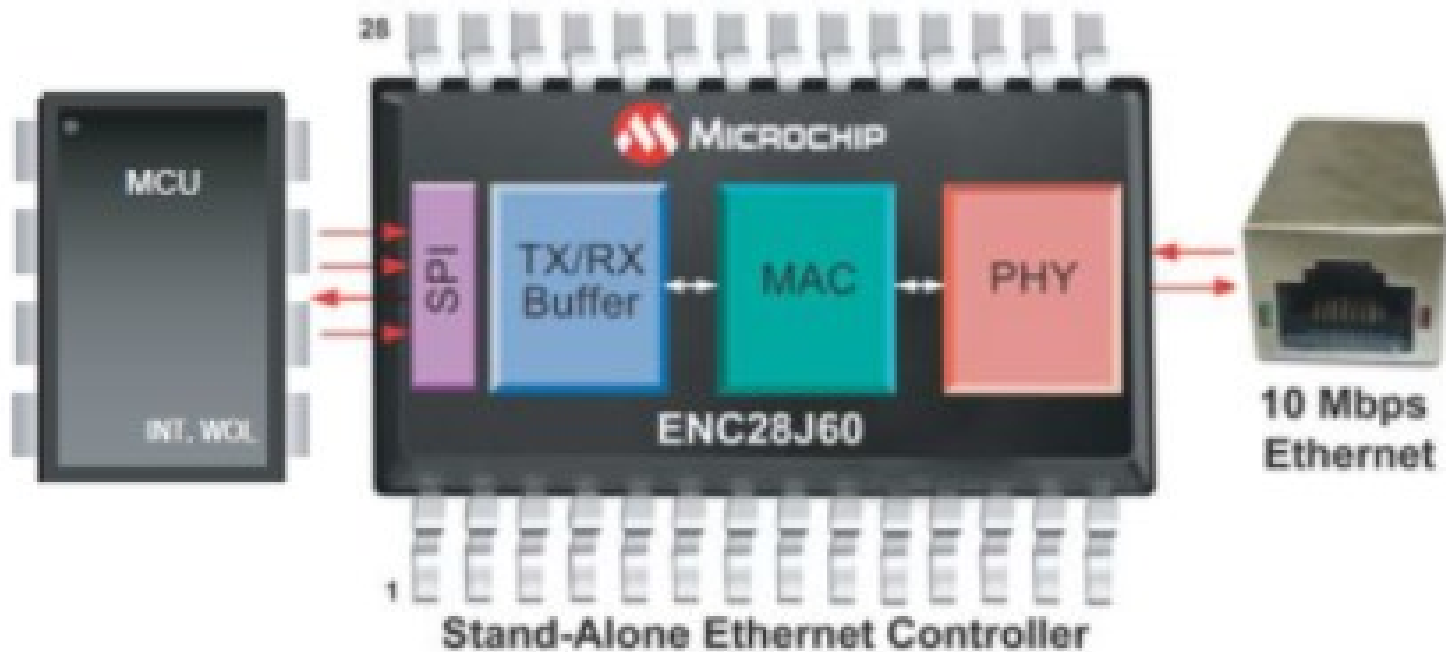
Interfata Ethernet. Structura RTL 8019.



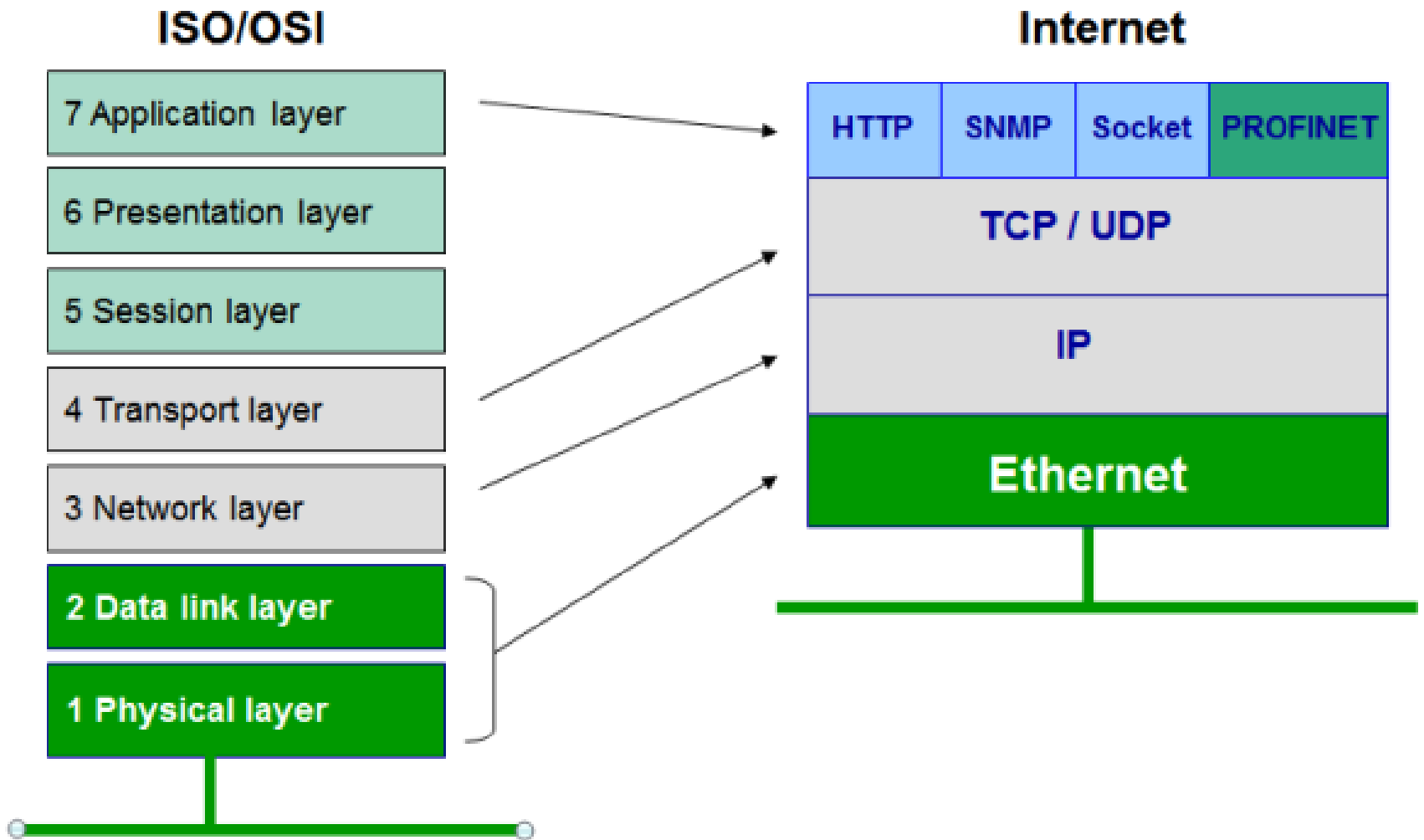
Interfata Ethernet. Schema bloc a circuitului CS8900.



Interfata Ethernet. Schema bloc a circuitului ENC28J60.



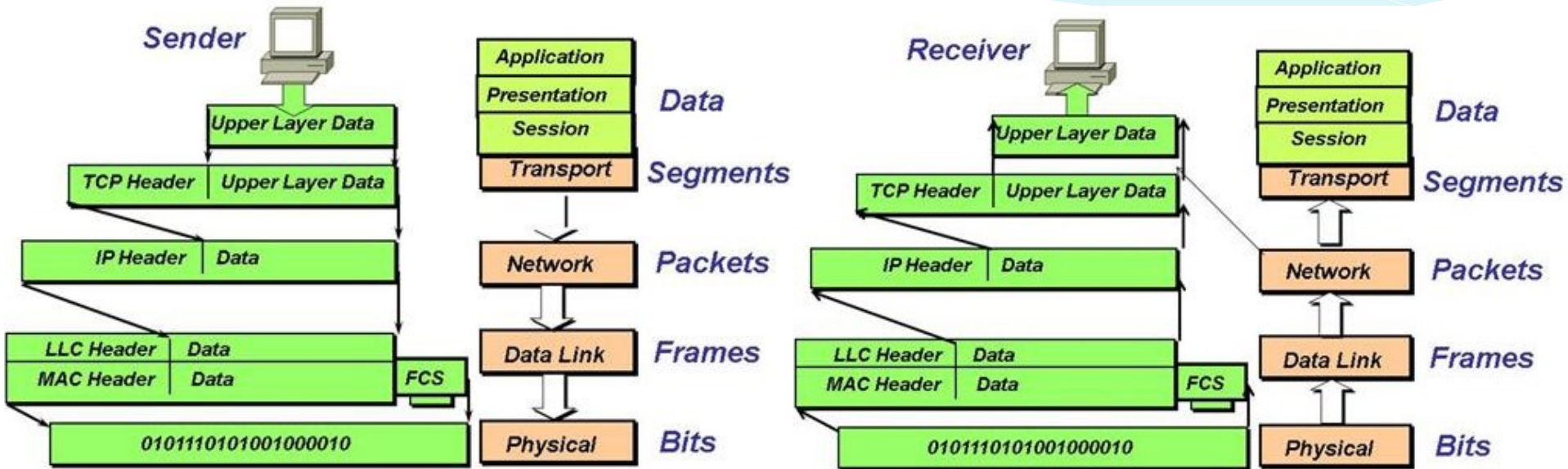
Protocolul Ethernet.



Codarea si Decodarea Datelor.

Procesul de codificare a datelor

Procesul de decodificare a datelor



Concluzii.

- Controlorul I8255. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod paralel.
- Controlorul I8251. Structura. Principiul de funcționare. Programarea și metode de utilizare la organizarea schimbului de date în cod secvențial.
- Interfața UART/USART
- Interfața COM.
- Magistrala USB.
- Interfețe RS: RS-232, RS-422, RS-423, RS-485.
- Interfața Bluetooth.
- Interfața IrDA.
- Interfața I2C.
- Interfața SPI.
- Interfața CAN.
- Interfața Ethernet.