

Interfațarea circuitelor

Tipuri de interfețe. Protocoale seriale de comunicare.

Ce este o interfață?

Wiki:

*“In computing, an interface is a **shared boundary** across which two separate components of a computer system **exchange information**.”*

Tipuri de interfețe

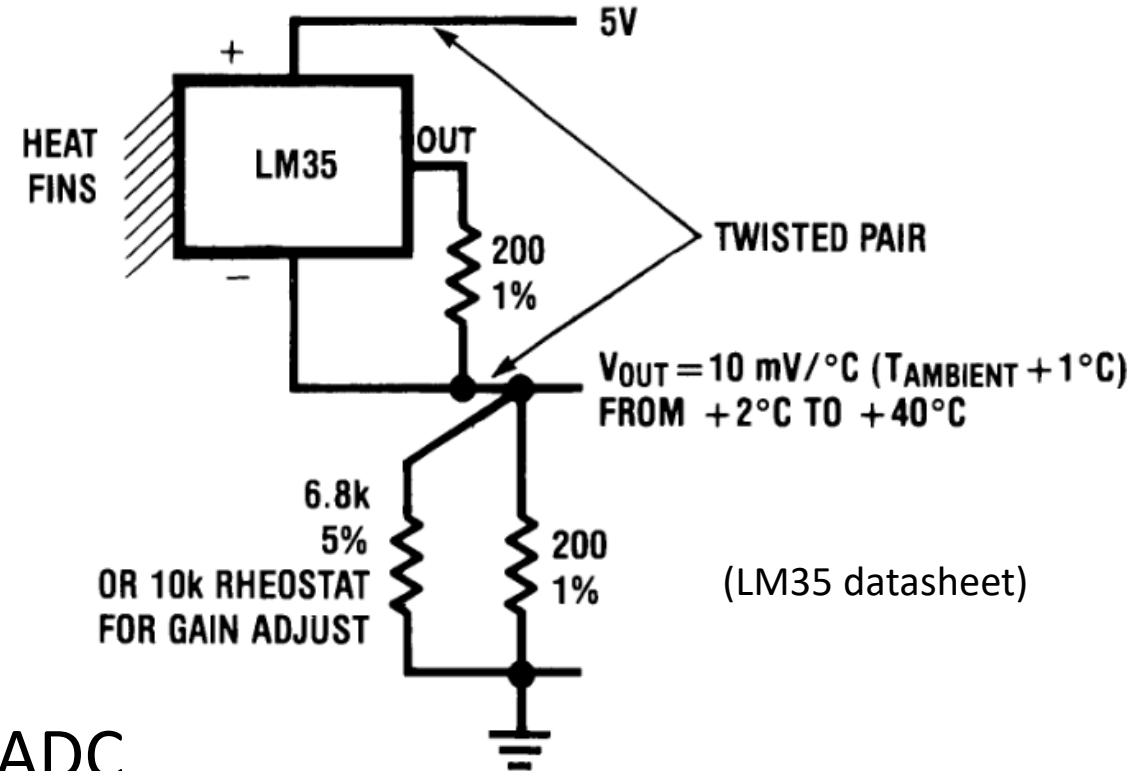
- **Analogice** – informația este transmisă sub forma unui semnal analogic
- **Digitale** – informația este transmisă sub forma unui semnal digital
 - **Paralele** – transmiterea informației are loc pe mai multe linii simultan permițând transmiterea concomitentă a biților
 - **Seriale** – transmiterea informației se face bit cu bit

Interfețe analogice

- Intrare – Porturi AD (analog to digital)
 - Prezent pe majoritatea platformelor embedded
- ieșire – Porturi DA (digital to analog)
 - Pe platforme specializate, mai rar întâlnit

Exemplu: Interfațarea unui senzor de temperatură

- Senzori de temperatură:
 - Elemente rezistive (RTD, termistor)
 - Necesită sursă constantă de curent
 - Ieșiri non-liniare
 - Circuite integrate
 - Soluție completă într-un chip
 - Ieșire liniară



Pin-ul de ieșire se conectează la un port ADC

Rezultatul conversiei ADC pe pinul desemnat = valoare proporțională cu temperatura mediului.

Interfețe paralele

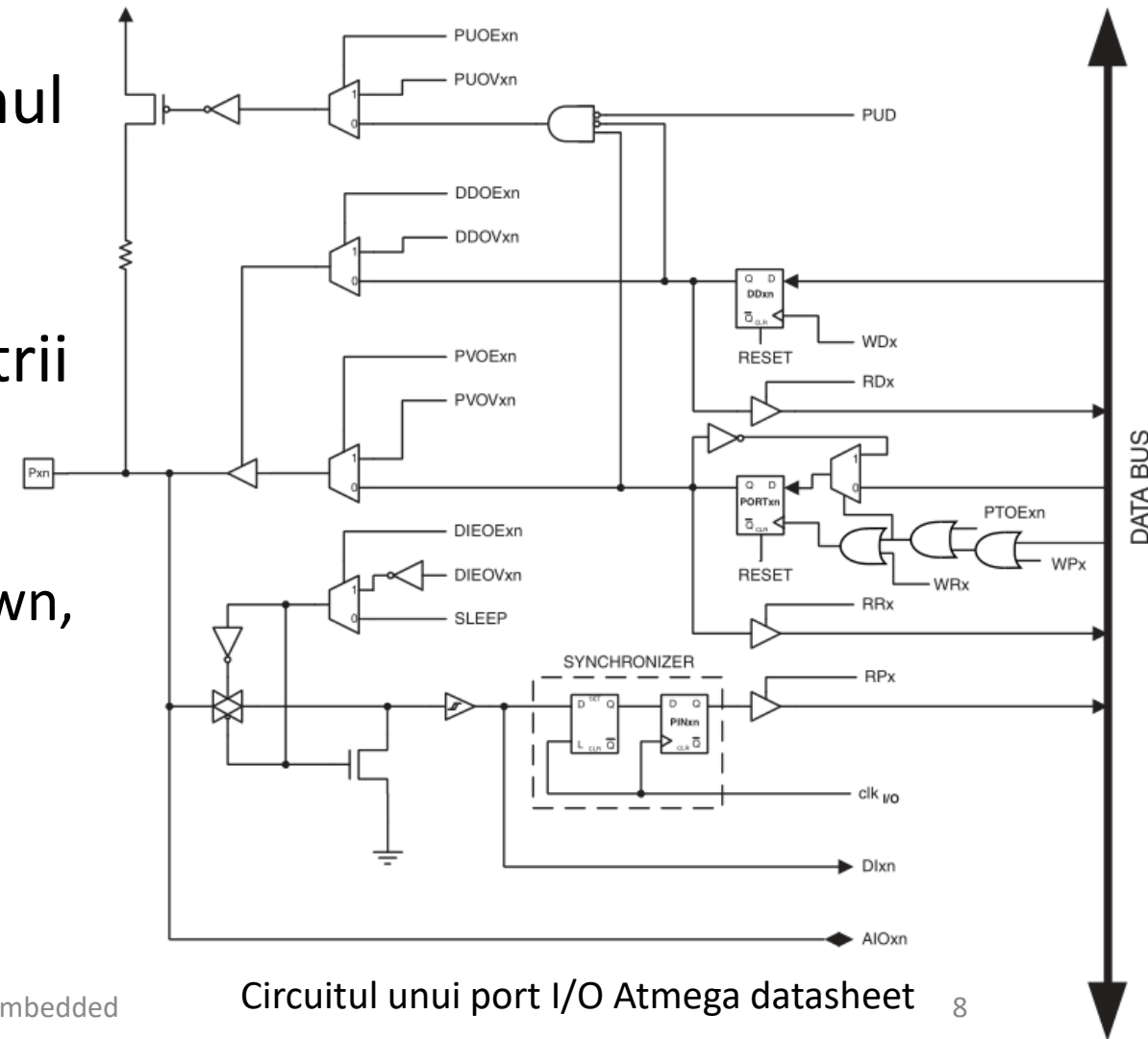
- Porturi paralele generice pentru intrare/ieșire
 - GPIO (General Purpose Input/Output)
 - Pot fi folosite pentru interfațare cu orice circuit pentru comunicare la nivel logic
 - Controlul comunicării trebuie implementat în totalitate la nivelul aplicației
- Interfețe pentru bus-uri externe
 - EBI (External Bus Interface)
 - Interconectează liniile interne de date și adrese interne cu dispozitive externe
 - Permite interfațarea cu circuite externe de memorie, display-uri sau alte periferice

Paralel versus Serial

- Paralel mai rapid decât serial? Da, pe distanțe mici.
- Pe distanțe mari semnalul este corupt din cauza interferențelor:
 - Interferența între linii
 - Zgomot
- Lățimea de bandă redusă față de cazul liniilor seriale:
 - Elemente capacitive
 - Inductanța mutuală

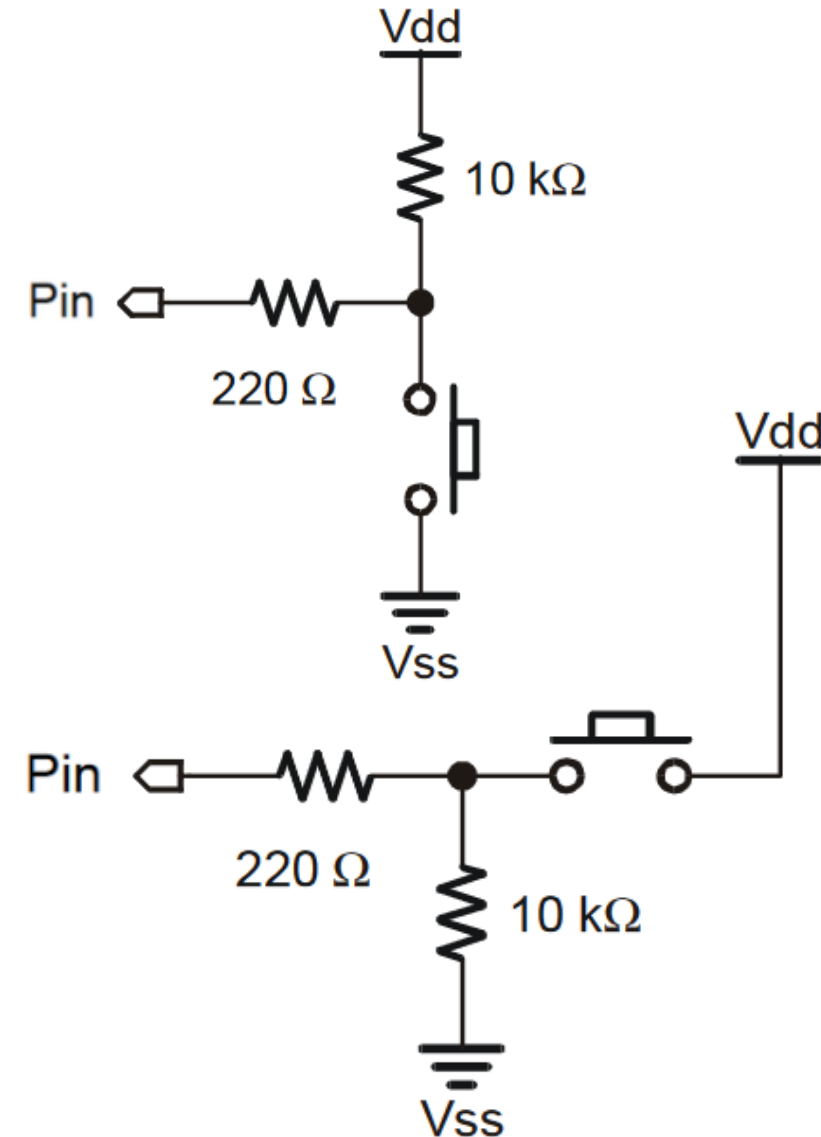
Porturi de intrare/ieșire

- Permit setarea unui nivel dorit pe pinul aferent sau citirea nivelului unei linii controlate de o altă componentă
- Caracteristici configurabile prin regiștrii dedicați:
 - Direcția portului
 - Caracteristici electrice: pull-up, pull-down, drive strength



Exemplu: Interfață pushbutton

- Circuit cu rezistență de pull-up:
 - Buton apăsat = '0'
 - Buton ne-apăsat = '1'
- Circuit cu rezistență de pull-down:
 - Buton apăsat = '1'
 - Buton ne-apăsat = '0'



Cum conectăm un pin de comandă?

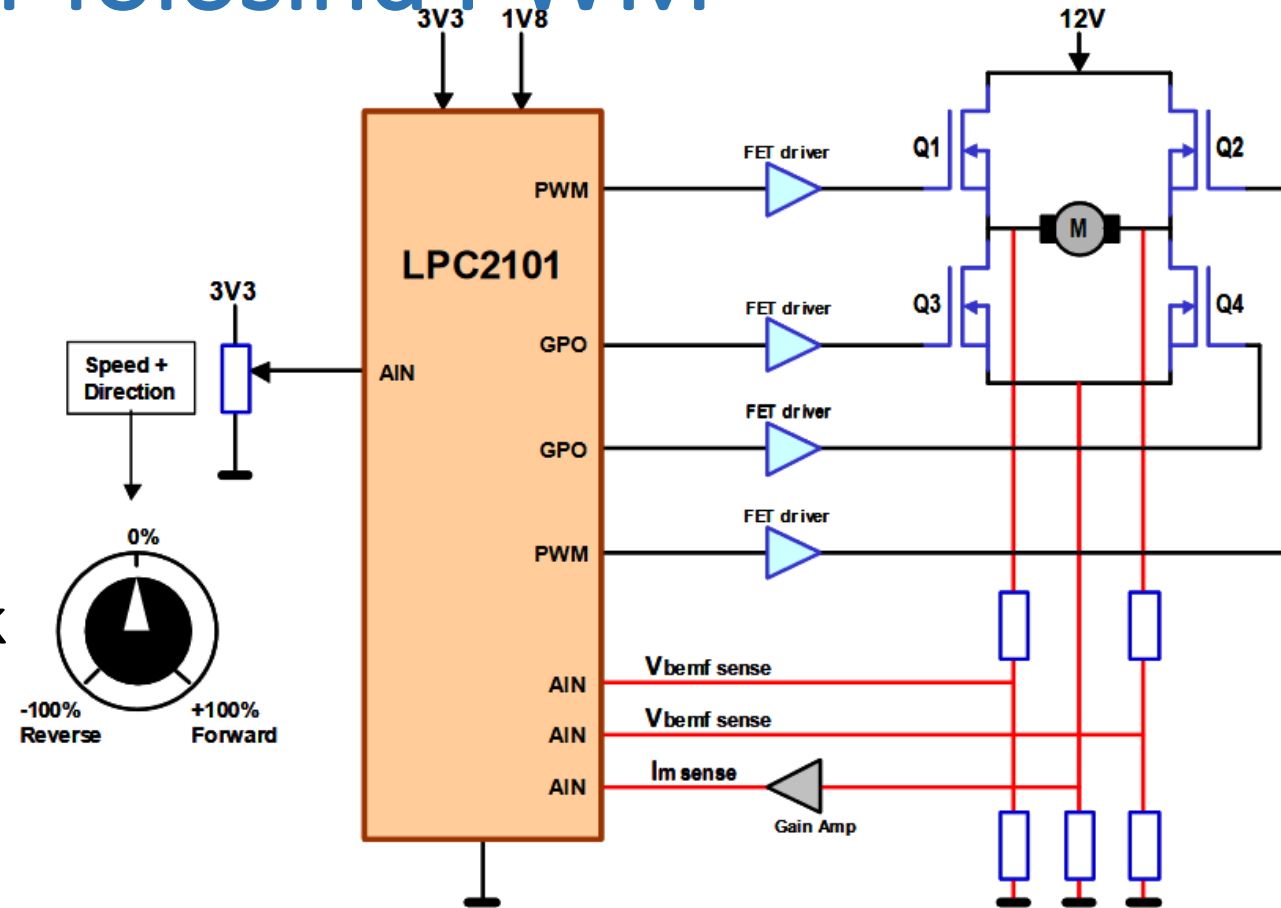
- Cum implementăm o comandă pornit/oprit pentru un LED, bec sau motor?
- Conexiune directă:
 - Source – LED-ul se va aprinde la setarea valorii '1'
 - Sink – LED-ul se va aprinde la setarea valorii '0'
 - Se folosesc rezistențe limitatoare de curent
 - Se va avea în vedere curentul maxim permis prin specificație
- Prin tranzistor/releu:
 - Pentru elemente de putere alimentarea este asigurată separat
 - Circuitul de comandă nu poate asigura curentul necesar

Funcții alternative ale porturilor

- Pulse Width Modulation
- Timer Input Capture
- Timer Output Compare

Exemplu: Control motor folosind PWM

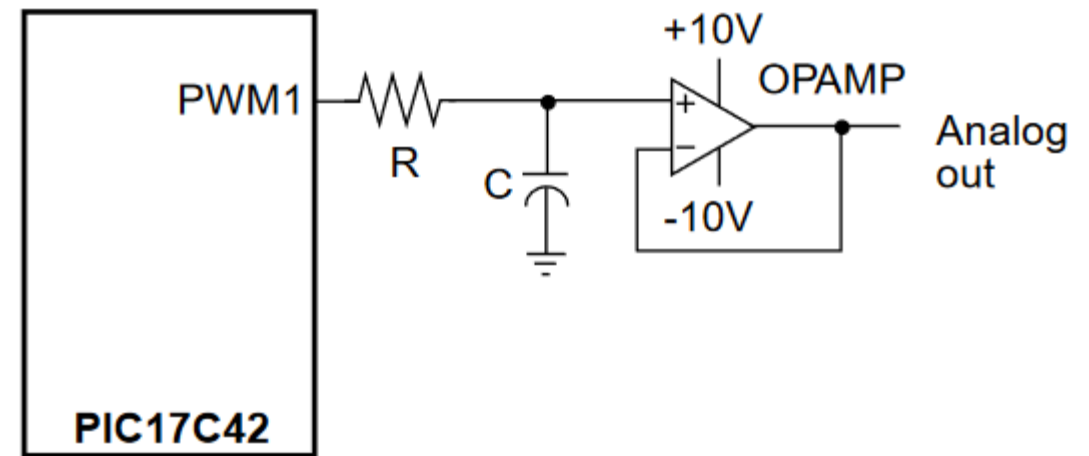
- Control motor curent continuu
- 2 porturi PWM controlează viteza
- 2 porturi I/O controlează direcția
- Porturi analogice pentru feedback și prescripție



(AN10513 Brushed DC motor control using the LPC2101)

Exemplu: Generare semnal analog folosind PWM

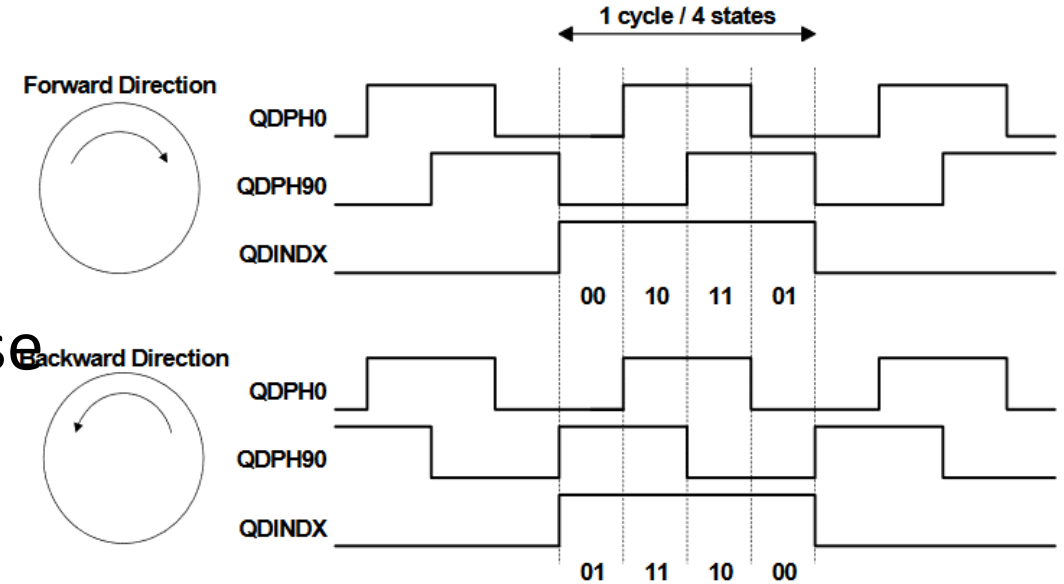
- Controlul formei de undă prin variația factorului de umplere și a frecvenței
- 1 port PWM conectat la un filtru extern trce-jos



(AN538 Using PWM to Generate Analog Output)

Exemplu: Interfațare encoder cvadratură

- Liniile digitale ale senzorului se conectează la linii digitale de tip input capture
- În funcție de succesiunea impulsurilor se poate afla direcția de deplasare
- În funcție de frecvența tranzițiilor (0->1 sau 1->0) se poate afla viteza de deplasare

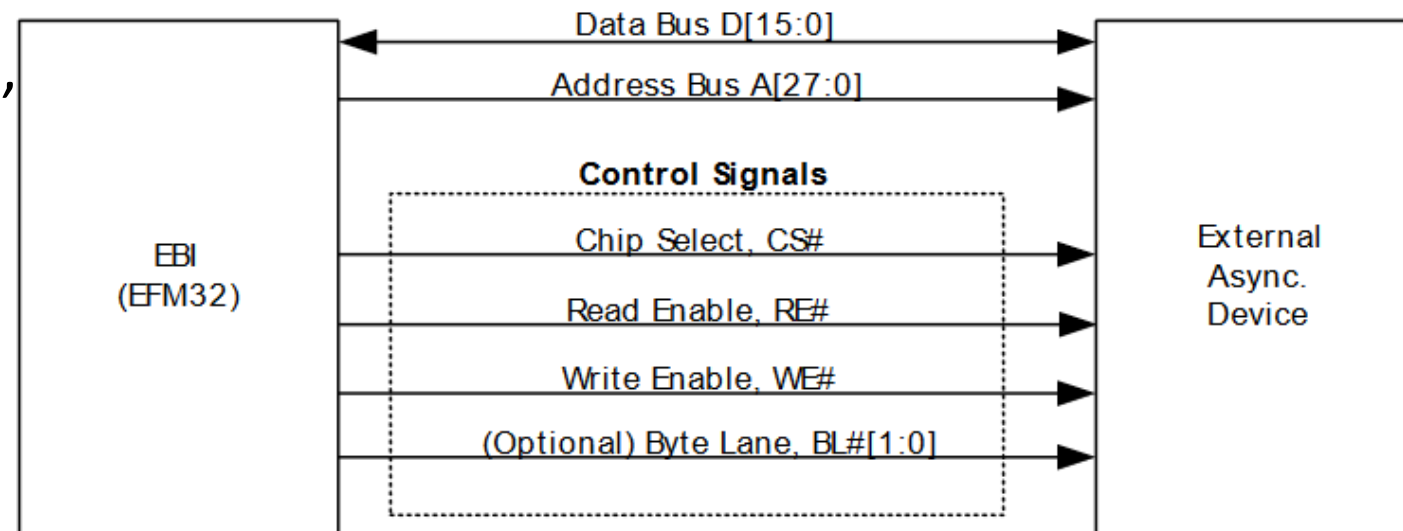


(AVR1600: Using the XMEGA Quadrature Decoder)

- Prin ce metodă realizăm citirea valorilor liniilor setate de senzor?

Interfețe pentru bus-uri externe

- Oferă toate liniile necesare conectării unor module externe precum memoriile:
 - Linii de adresare
 - Linii de date
 - Chip select
 - Linii de comandă (Read enable, Write enable)
- Circuitele externe pot să fie mapate în spațiul adresabil al procesorului

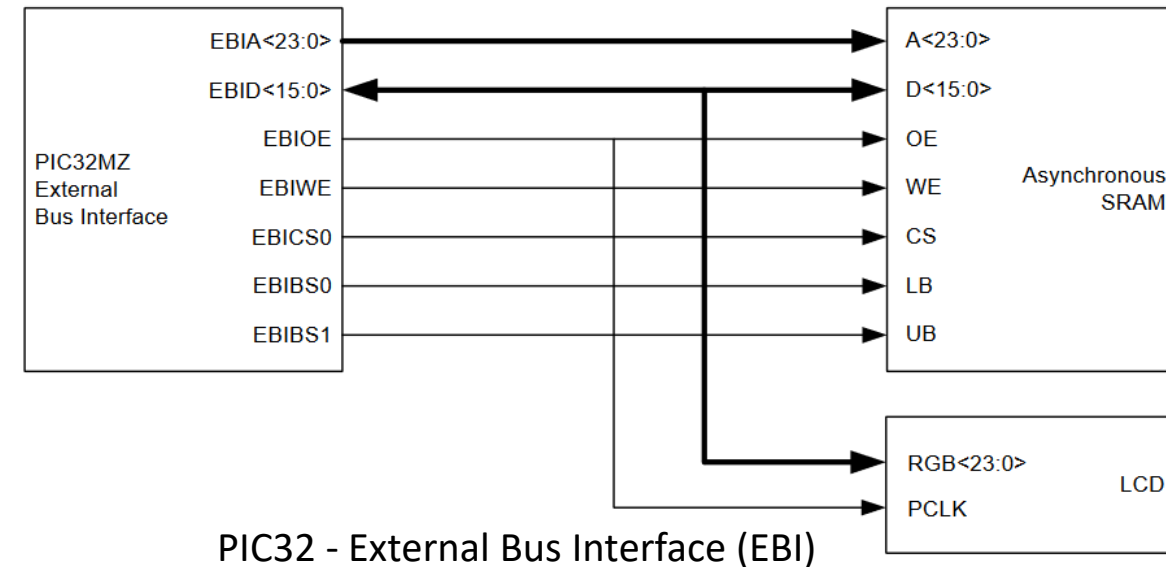


Schema EBI – microcontroller EFM32 Silicon Labs

Exemplu: Interfețe pentru bus-uri externe

- Conectarea unui display LCD împreună cu o memorie RAM ca buffer pentru afișarea datelor
- Microcontrollerul adresează datele din memoria RAM
- Odată puse pe liniile de date acestea sunt preluate și afișate de LCD

- Notă: un astfel de display LCD nu are nevoie de linii de adresare



PIC32 - External Bus Interface (EBI)

Interfețe seriale

- Distanța de comunicare:
 - Pe distanțe mici – pe același PCB
 - Pe distanțe mari – între PCB-uri sau componente ale unui sistem mai complex
- Sincronizare:
 - Sincrone – sincronizarea se face utilizând o linie suplimentară de clock
 - Asincrone – sincronizarea se face prin alte mecanisme specifice fiecărui protocol

Interfețe seriale

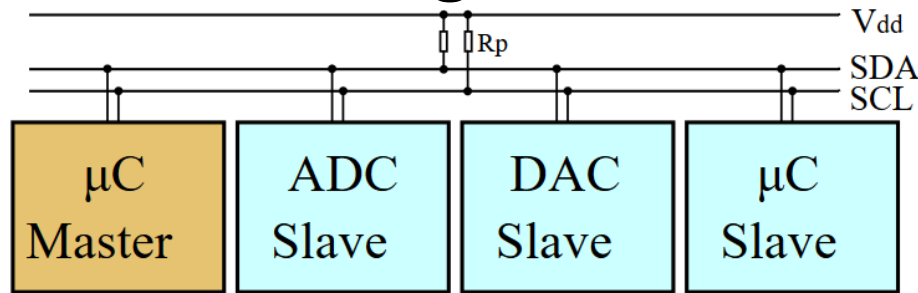
- Fluxul de date:
 - Simplex – comunicarea permisă într-o singură direcție
 - Half duplex – comunicare bidirecțională, transmiterea se face într-o singură direcție la un moment dat
 - Full duplex – comunicare bidirecțională, transmiterea datelor se poate face concomitent în ambele direcții

Bus-uri seriale pentru comunicare pe distanțe mici

- Folosesc în general nivele digitale logice

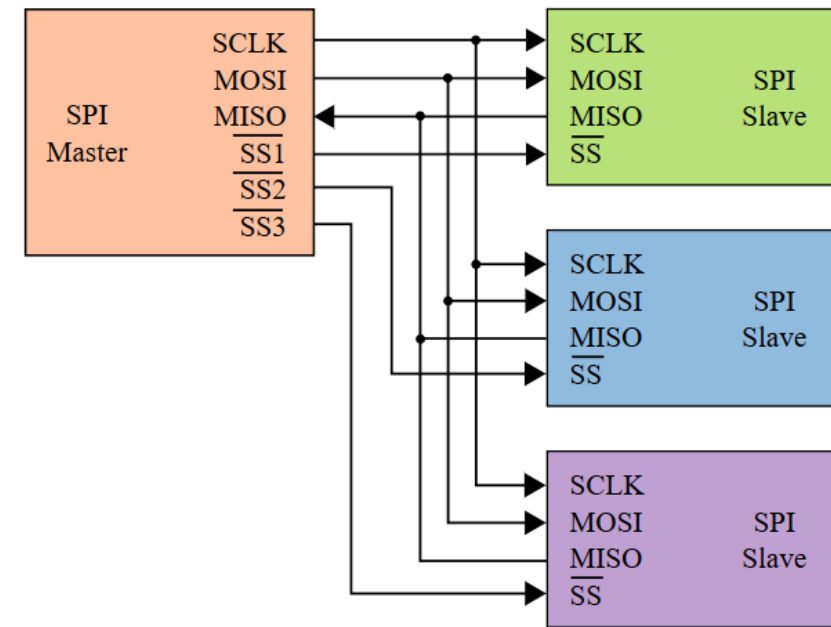
Exemple des utilizate:

- SPI - Serial Peripheral Interface
- I2C - Inter-Integrated Circuit



(<https://en.wikipedia.org/wiki/I%C2%B2C>)

- 1-Wire

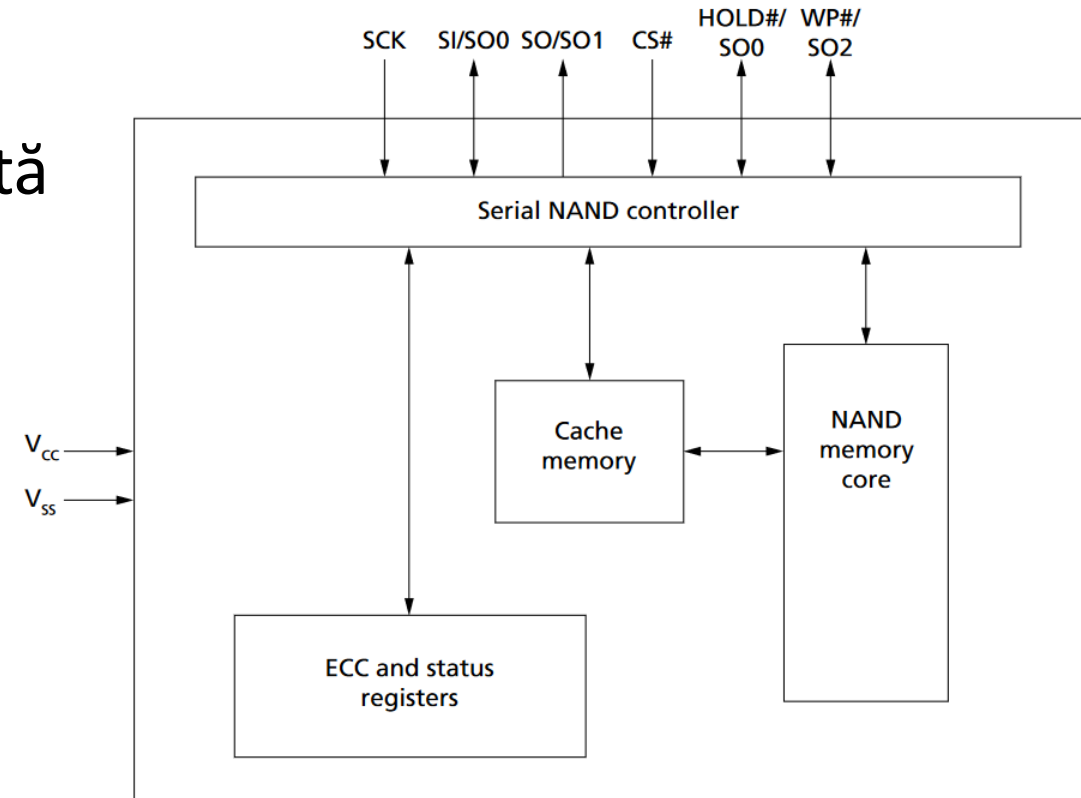


(https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus)

Exemplu: Memorie NAND - comunicare SPI

- Interfața standard paralelă este înlocuită cu o interfață SPI
- Utilizarea memoriei se face cu ajutorul unui set de comenzi definit prin specificația circuitului

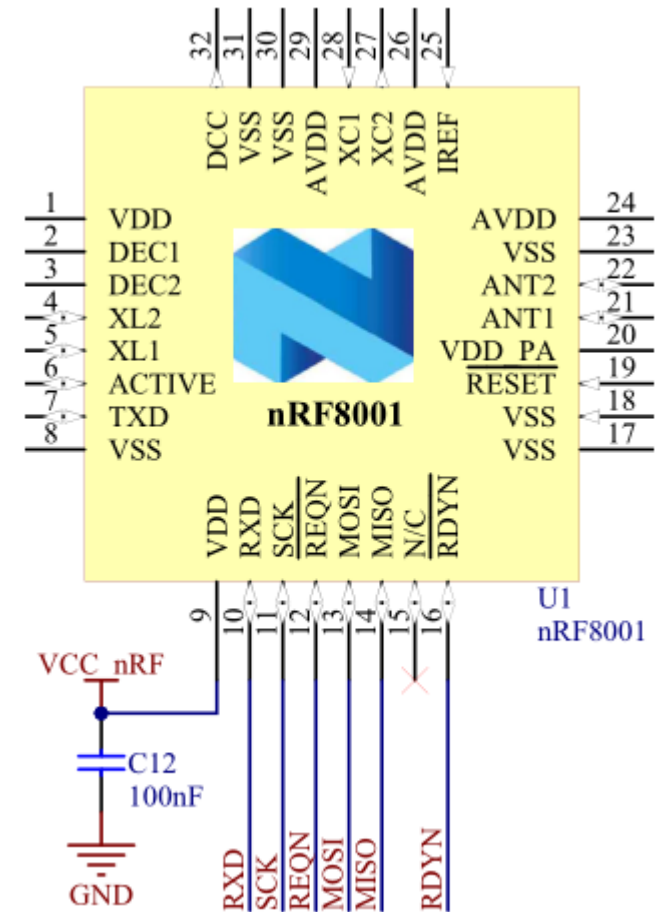
Command	Op Code	Address Bytes	Dummy Bytes	Data Bytes	Comments
BLOCK ERASE	D8h	3	0	0	Block erase
GET FEATURE	0Fh	1	0	1	Get features
PAGE READ	13h	3	0	0	Array read
PROGRAM EXECUTE	10h	3	0	0	Enter block/page address, no data, execute
PROGRAM LOAD RANDOM DATA	02h	2	0	1 to 2112	Load program data—2kB MAX
PROGRAM LOAD	84h	2	0	1 to 2112	Enter cache address/data
READ FROM CACHE	03h, 0Bh	2	1	1 to 2112	Output cache data at addr
READ FROM CACHE x2	3Bh	2	1	1 to 2112	Output cache data on SI and SO
READ FROM CACHE x4	6Bh	2	1	1 to 2112	Output cache data on SI, SO, WP#, HOLD#
READ ID	9Fh	0	1	2	Read device ID
RESET	FFh	0	0	0	Reset the device
SET FEATURE	1Fh	1	0	1	Set features
WRITE DISABLE	04h	0	0	0	
WRITE ENABLE	06h	0	0	0	



(NAND Flash Memory Serial Peripheral Interface (SPI)
MT29F1G01AAADD)

Exemplu: Modul Bluetooth

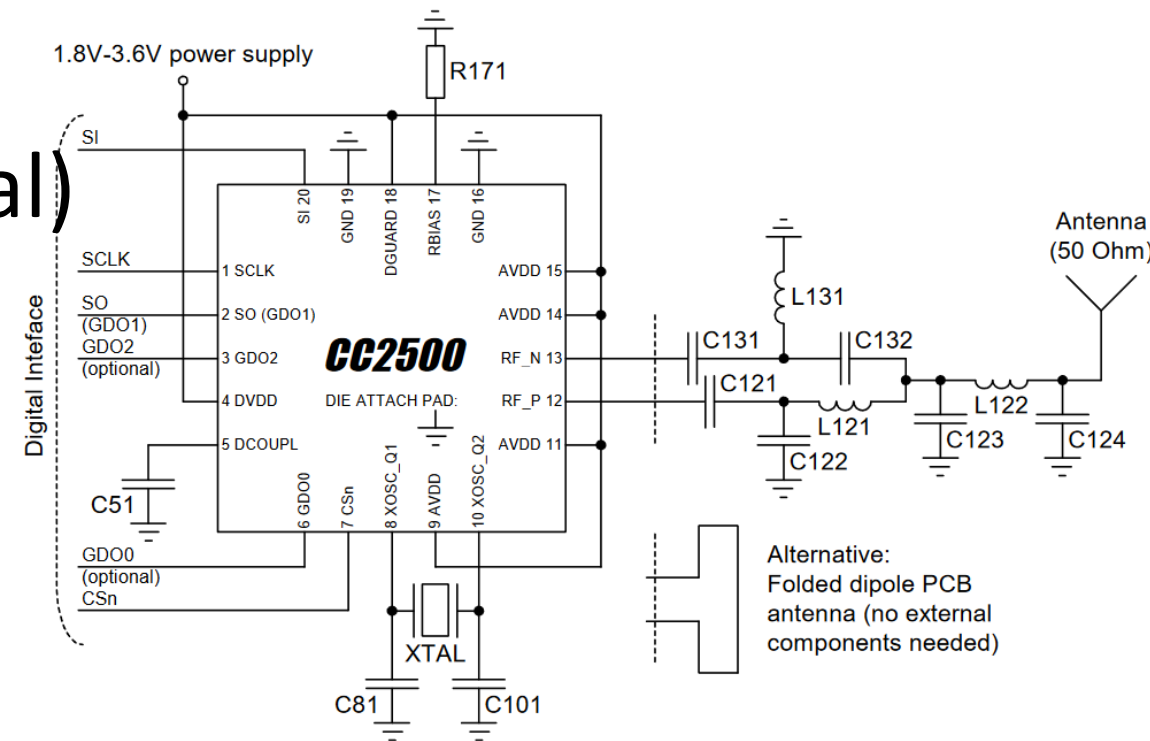
- Modul de comunicare Bluetooth Low Energy
- Implementează layerele specifice comunicației Bluetooth conform specificației
- Configurat și controlat prin conexiune SPI
- Interfață suplimentară UART pentru funcționalități de testare



(nRF8001 Single-chip Bluetooth low energy solution)

Exemplu: Modul comunicare RF 2,4GHz

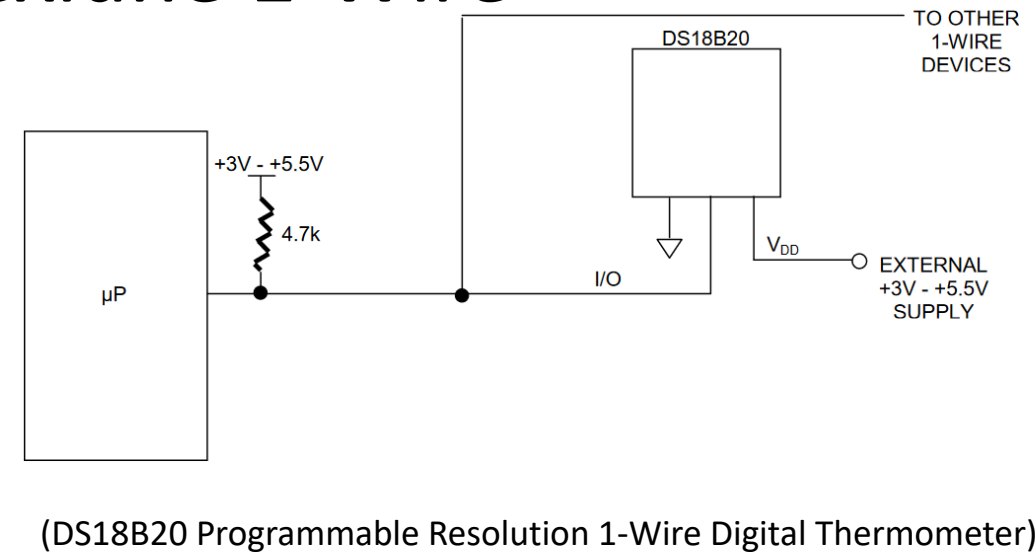
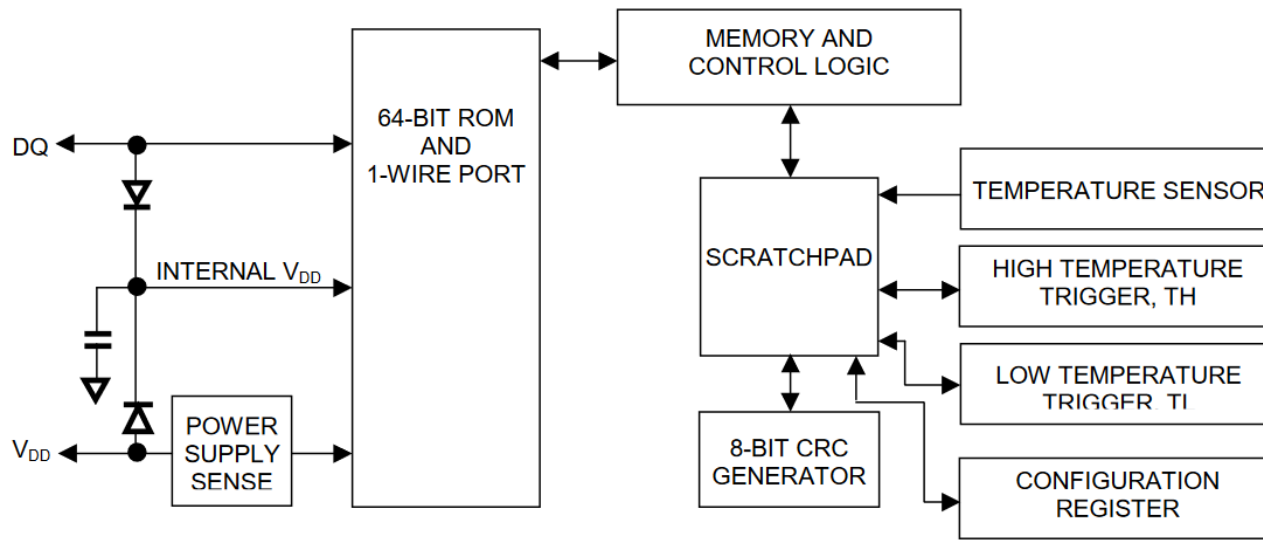
- Utilizat în comunicații RF în banda 2400-2483.5 MHz ISM (Industrial, Scientific și Medical) și SRD (Short Range Device)
- Configurat și controlat prin conexiune SPI



(CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver)

Exemplu: Senzor de temperatură inteligent

- Utilizat în control termostatic, sisteme industriale sau orice alt sistem care necesită măsurarea temperaturii
- Configurat și controlat prin conexiune 1-Wire



Comunicare serială pe distanțe mai mari

- UART
- LIN
- CAN
- Profibus
- Modbus
- Ethernet
- FlexRay
- KNX

(DS18B20 Programmable Resolution 1-Wire Digital Thermometer)

Comunicare serială pe distanțe mai mari

- Reprezentare specifică a informației pe bus la nivel fizic în funcție de protocol
- Necesită utilizarea a două componente ce implementează detaliile protocolului:
 - **Transceiver** – Layerul fizic – transformă informația din formă digitală în nivele fizice de tensiune pe bus
 - **Controller** – Layerul legătură de date (data link) – controlează transmisia și recepția datelor

Transceiver

- Circuit integrat de sine stătător care se conectează la controller prin linii digitale de date (Rx, Tx) și alte linii de control
- Reprezintă interfața dintre mediul digital și mediul fizic de pe bus

Controller

- Se regăsește sub formă de circuit integrat de sine stătător, dar mai des ca modul integrat al unui microcontroller
- În cazul în care se prezintă ca circuit separat se interfațează cu procesorul prin alte interfețe seriale (SPI, I2C)
- Ca modul al microcontrollerului controlul și configurarea lui se face direct prin regiștrii dedicați

Cum alegem metodele de interfațare?

- Se identifică componentele necesare:
 - Senzori
 - Actuatore
 - Memorii
 - Module de comunicare/transceivere
- Se evaluează posibilitățile și restricțiile de interfațare pentru componente:
 - Același tip de componente pot suporta interfețe diferite
 - Viteza de transfer a datelor trebuie luată în considerare

Influența interfețelor asupra procesorului ales

- Procesorul trebuie să suporte:
 - Numărul de conexiuni necesare
 - Tipul de conexiuni necesare
 - Managementul tuturor interfețelor fără a reduce performanța