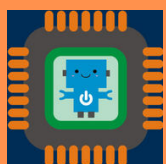


Să înțelegem microcontrolerele

Dezvoltat de MECB Ltd



A Trainers Toolkit To Foster STEM Skills Using Microcontroller Applications



Co-funded by the
Erasmus+ Programme
of the European Union

Project No. 2019-1-RO01-KA202-063965

This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Să înțelegem microcontrolerele

Cuprins



Istoric



Principiul de funcționare



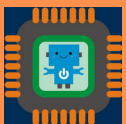
Tipuri de microcontrolere



Controlere Off-the-Shelf



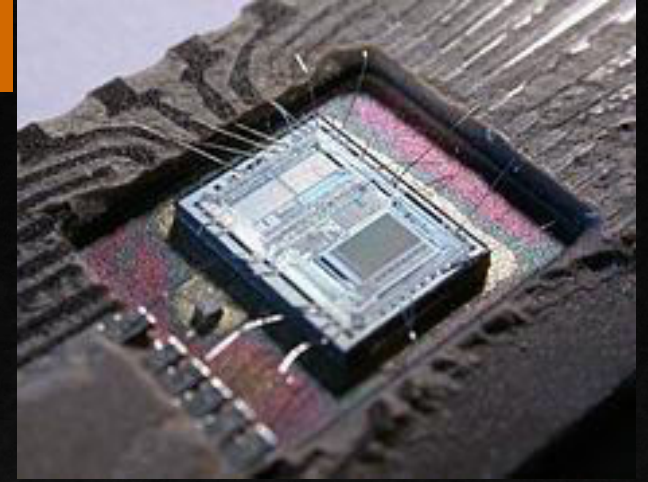
Sumar



Scurt istoric

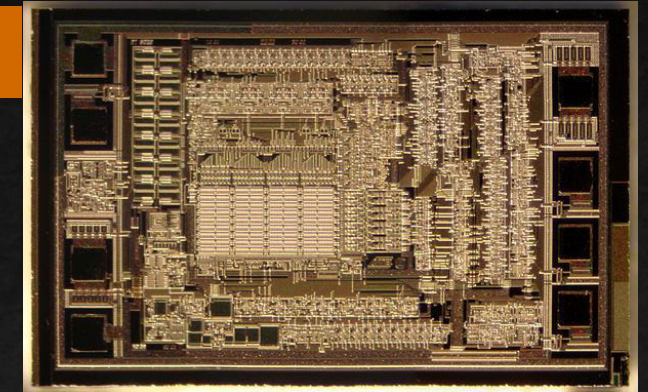
- Microcontrolerele au fost inventate în anii '70, deoarece pentru prima dată au fost incorporate memoria read-only sau read/write și procesorul pe un singur die/chip. De atunci, unul dintre cele mai populare microcontrolere care încă este utilizat astăzi e 8051 dezvoltat de Intel.
- A doua etapă în dezvoltarea microcontrolerelor a avut loc imediat după, tot în anii '70, a fost introducerea EEPROM (Electrically Erasable Programmable Read-Only Memory), ceea ce a permis microcontrolerului să-și șteargă memoria cu ajutorul unor dispozitive electronice mai compacte, permițând reducerea dimensiunilor și timpi de execuție mai rapizi și a permis o mai largă utilizare odată cu aplicabilitatea extinsă.
- Acestea s-au dezvoltat până în punctul în care Umanitatea depinde de această tehnologie să funcționeze, deoarece majoritatea tehnologiei moderne ar fi imposibilă fără ele.

1971

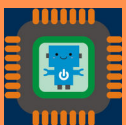


Die de la Intel, de Ioan Sameli, Wikipedia

1972



Circuit EEPROM , de Anonim, Wikipedia



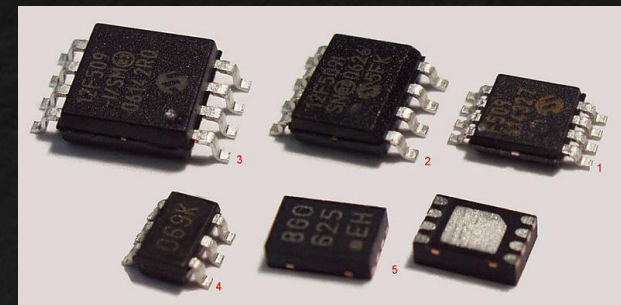


Ce este un microcontroler

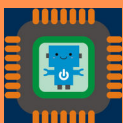
- Microcontrolerele sunt computere, care sunt făcute, proiectate și programate pentru aplicații specifice. Acestea sunt utilizate pentru controlul altor părți ale unui sistem electronic, prin intrări și ieșiri. Acestea funcționează exact ca un computer, cu diferența că, în loc de ceva uriaș, se poate încadra în interiorul palmei tale, sau chiar la vârful degetului.
- Datorită dimensiunilor mici pot fi folosite în majoritatea aplicațiilor, dar tot acest considerent le produce și dezavantajul de a avea un procesor, RAM, intrări și ieșiri mai mici, etc.



Desktop Computer vs Microcontroler, de Robot shop,
<https://www.robotshop.com/community/tutorials/show/how-to-make-a-robcomot-lesson-4-understanding-microcontrollers>



Microcontrolere mici , de Anonim ,
Wikimedia





Programarea unui microcontroler

- Toate controlerele rulează pe un limbaj numit Assembly Language, dar este obositor să programezi în acesta, astfel încât programele de nivel superior sunt utilizate pentru a accelera procesul, cum ar fi C #, Java, Python, etc.
- Deoarece controlerul trebuie să ruleze pe Assembly Language, un compilator (un program care convertește din limbajul de nivel înalt în cel de nivel scăzut) este utilizat. Acesta este un pas foarte important, deoarece sistemele electronice rulează pe Machine Code, nu pe programe de nivel înalt.
- Avantajul evident este că există mai puțin timp pierdut la scrierea codului, dar dezavantajul este că, de obicei, o mulțime de cod suplimentar va fi introdus în microcontroler, ceea ce de obicei îl încetinește.

```

MONITOR FOR 6802 1.4          9-14-80  TSC ASSEMBLER  PAGE   2

C000          ORG      ROM+$0000 BEGIN MONITOR
C000 8E 00 70  START  LDS      $STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A

0013          RESETA EQU  $00010011
0011          CTLREG EQU  $00010001

C003 86 13          INITA  LDA A  #RESETA  RESET ACIA
C005 B7 80 04          STA A  ACIA
C008 86 11          LDA A  #CTLREG  SET 8 BITS AND 2 STOP
C00A B7 80 04          STA A  ACIA

C00D 7E C0 F1          JMP      SIGNON  GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

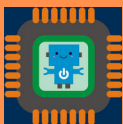
C010 B6 80 04  INCH  LDA A  ACIA      GET STATUS
C013 47          ASR A  ACIA      SHIFT RDRP FLAG INTO CARRY
C014 24 FA          BCC  INCH      RECIEVE NOT READY
C016 B6 80 05          LDA A  ACIA+1  GET CHAR
C019 84 7F          AND A  #$7F     MASK PARITY
C01B 7E C0 79          JMP      OUTCH  ECHO & RTS

*****
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: Digit in acc A
* CALLS: INCH
* DESTROYS: acc A
* Returns to monitor if not HEX input

C01E 80 F0          INHEX  BSR      INCH  GET A CHAR
C020 81 30          CMP A  #'0      ZERO
C022 2B 11          BMI  HEXERR  NOT HEX
C024 81 39          CMP A  #'9      NINE
C026 2F 0A          BLE  HEXRTS  GOOD HEX
C028 81 41          CMP A  #'A
C02A 2B 09          BMI  HEXERR  NOT HEX
C02C 81 46          CMP A  #'F
C02E 2E 05          BGT  HEXERR
C030 80 07          SUB A  #7      FIX A-F
C032 84 0F          HEXRTS  AND A  #$0F  CONVERT ASCII TO DIGIT
C034 39          RTS

C035 7E C0 AF  HEXERR  JMP      CTRL   RETURN TO CONTROL LOOP
    
```

Limbaj de asamblare, de Michael Holley, Wikimedia

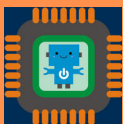
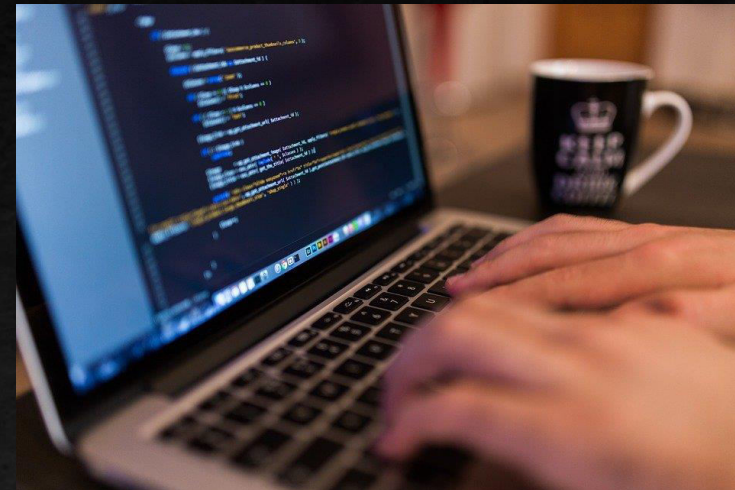




Programarea unui microcontroler

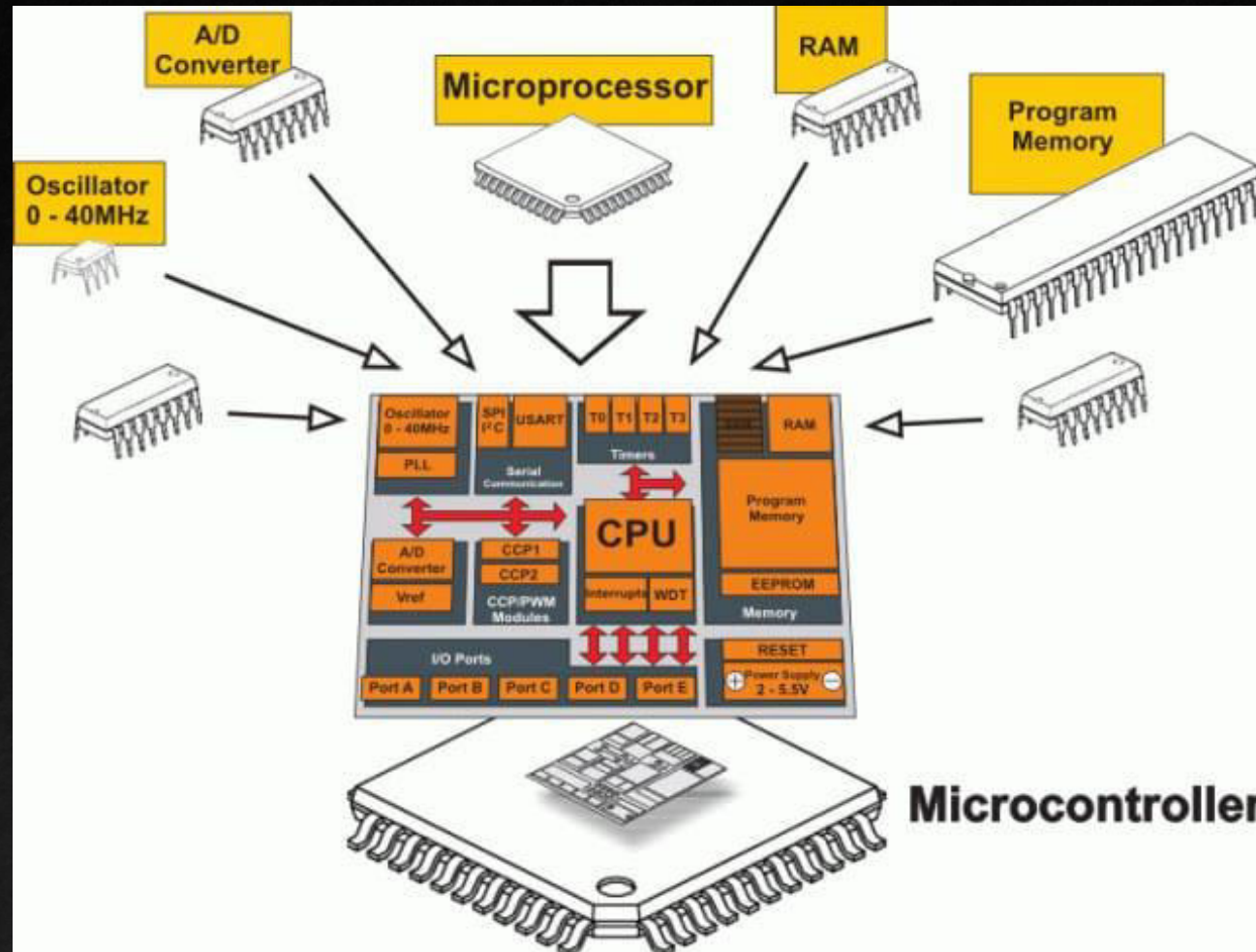
De obicei, procesul poate fi defalcat în 4 pași simpli;

1. Scrieți codul programului pe un computer
2. Compilați codul pentru microcontrolerul pe care îl utilizați
3. Conectați microcontrolerul la computer
4. Încărcați versiunea compilată a programului pe microcontrolerul dvs. (salvat în memoria programului)



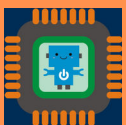


Componentele interne ale microcontrolerului



Componentele unui microcontroler, de Max Embedded,

<https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller>



A Trainers Toolkit To Foster STEM Skills Using
Microcontroller Applications

Project No. 2019-1-RO01-KA202-063965

This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

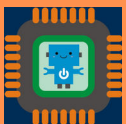


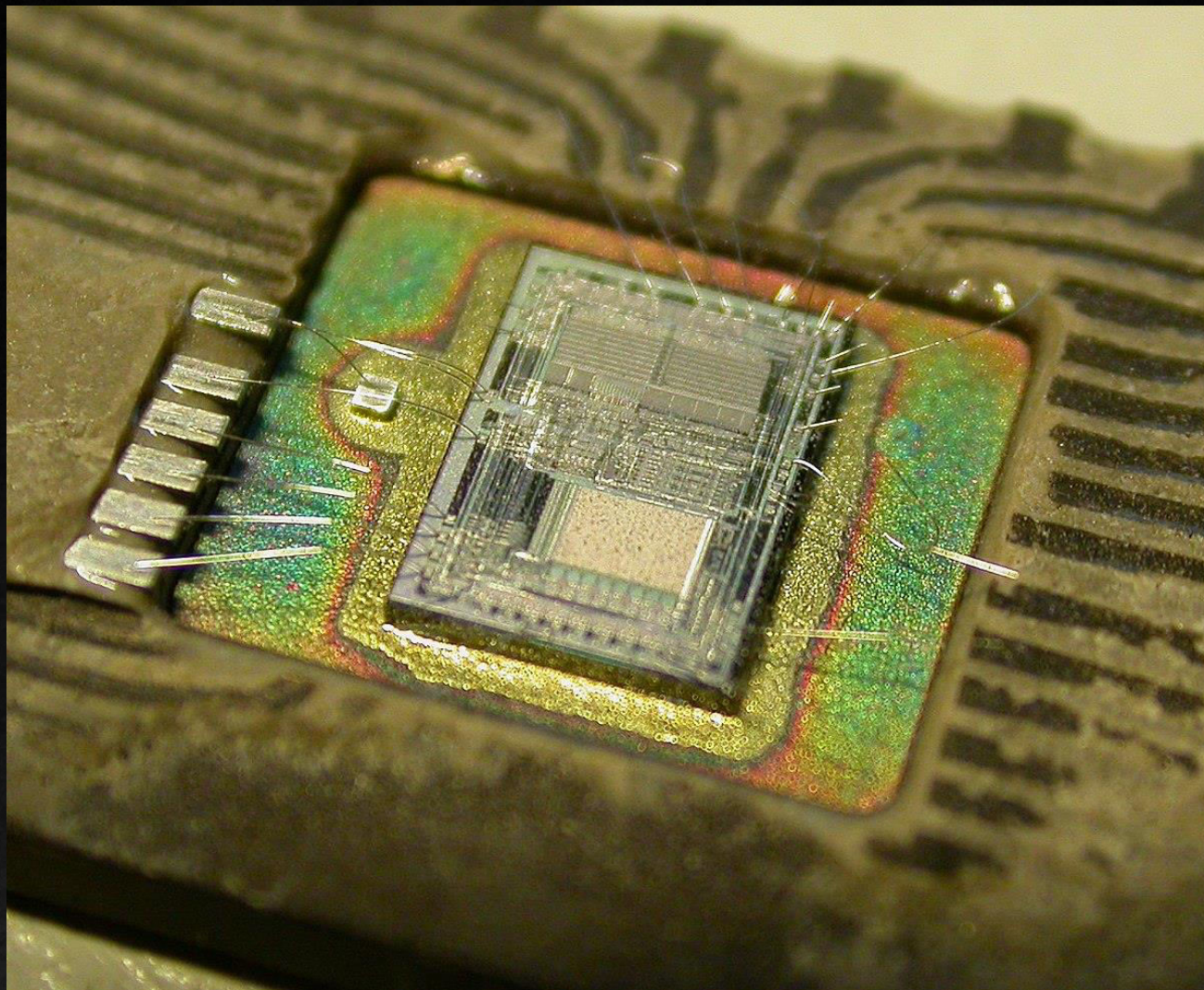
Co-funded by the
Erasmus+ Programme
of the European Union



Părți importante ale microcontrolerului

- RAM (Random Acces Memory): Aici microcontrolerul își stochează informațiile în timp ce funcționează. Aceasta este un depozit în care nicio informație nu este stocată dacă este dezactivată (Volatilă)
- CPU (Central Processing Unit): Acesta este locul în care computerul execută instrucțiunile date acestuia. Oscilatorul este folosit ca un clock intern.
- Memorie programabilă: Aici microcontrolerul stochează programul creat, care este introdus de obicei după fabricarea dispozitivului. Acest tip de memorie trebuie să fie non-volatilă, deoarece conține instrucțiunile despre modul în care ar trebui să funcționeze dispozitivul, care, fără ea, nu ar funcționa deloc.
- Porturile I/O: Aici microcontrolerul primește intrări și execută ieșiri. De obicei, acestea sunt digitale, adică fie ridicate, fie scăzute (0 / 1), deși acestea pot fi și pini analogici care pot da o serie de semnale.

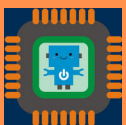




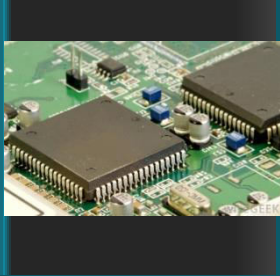
Microcontroler, de Unknown, Wikimedia

»» Tipuri de microcontrolere

- În prezent există zeci de mii de microcontrolere diferite pe piață, dar defalcarea acestora poate fi separată în trei subcategorii, care sunt:
 - 1. Microcontrolere încorporate
 - 2. Microcontrolere de la 8 la 32 de biți
 - 3. Procesoare de semnal digitale
- Acestea pot fi diferențiate, deoarece acest tip de microcontrolere, deși au funcții similare, au modele interne diferite, astfel încât acestea să poată funcționa mai bine în acel domeniu respectiv.



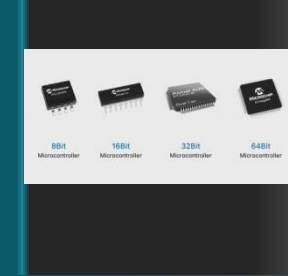
>>> Diferite tipuri de microcontrolere



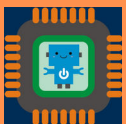
Microcontrolere
încorporate



Procesoare de semnal
digitale(DSP)

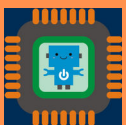
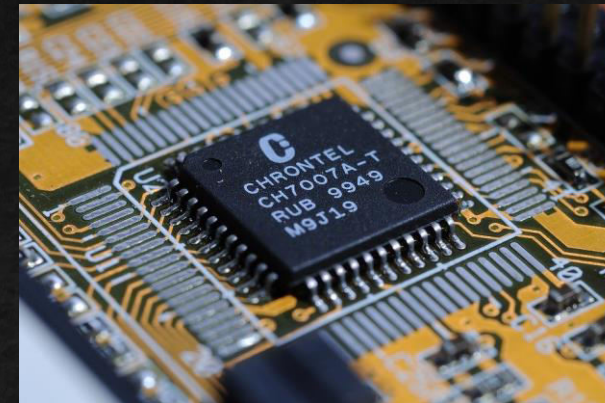


Microcontrolere
8 - 32 bit



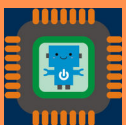
Microcontrolere încorporate

- Acestea sunt cele mai comune tipuri de microcontrolere, care sunt folosite în zilele noastre. Acestea sunt concepute pentru o funcție specifică și nu ar funcționa dacă ar fi implementate într-un scenariu diferit de cel pentru care au fost programate. Deoarece aplicațiile pentru care acestea sunt utilizate sunt foarte specifice, costurile pentru acest tip de microcontrolere sunt de obicei foarte mici, deoarece acestea nu necesită prea multe componente pentru a funcționa. Acestea sunt foarte utile pentru obiecte de zi cu zi, cum ar fi calculatoare, mașini de spălat, bancomate, telecomenzi etc.
- Datorită acestui fapt, au fost dezvoltate o serie de microcontrolere, cum ar fi Intel 8051, PIC, STM32 etc.



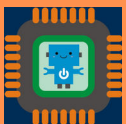
»»» Digital Signal Processors

- DSP-urile sunt microcontrolăre care au fost concepute pentru a optimiza semnale cu cuvinte reale în informații utile, adică aplicarea formulei matematice pentru procesarea semnalului. Datorită acestui fapt, designul microcipului este specific pentru îmbunătățirea vitezei de procesare a semnalului, făcându-l să reacționeze mai rapid la intrarea unui semnal în comparație cu orice alt microcontroller tipic.
- Cele mai tipice operațiuni efectuate de DSP sunt de obicei „Minus”, „Plus”, „Îmulțire” și „Împărțire”. Acest lucru este extrem de util deoarece monitoarele, microfoanele, modemurile etc. pot funcționa mult mai repede decât înainte.



Microcontrolere 8-32 bit

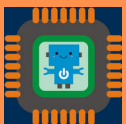
- Aceste microcontrolere sunt de obicei proiectate să funcționeze într-o multitudine de situații diferite, făcându-le utile pentru aplicații care trebuie să fie versatile. Acestea conțin în mod obișnuit toate componentele necesare necesare funcționării, adică RAM, EEPROM (Electrically Erasable Programmable Read-Only Memory) etc.
- Deoarece costurile au scăzut, aplicațiile pentru acest tip de microcontroler a crescut dramatic și, datorită versatilității și ușurinței de utilizare, au intrat în cea mai mare parte a vieții noastre cotidiene. De exemplu: ceasuri inteligente, roboți, PLC-uri (Programmable Logical Controller), etc ...



»»» Considerații pentru alegerea unui microcontroller

Există cinci componente principale care sunt analizate atunci când alegeți un microcontroller. Acestea sunt;

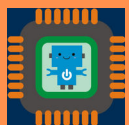
1. CPU, care va determina cât de rapid va executa microcontrollerul funcțiile
2. Porturile I / O, care va determina câte componente poate gestiona
3. Memoria, care va determina cât de complexă va fi funcția
4. Funcții speciale, unde orice alte componente necesare sunt necesare pentru ca acesta să funcționeze ca întreruptoare, temporizatoare, etc.
5. Dimensiuni fizice, care vor determina cât de mare va fi, dar și toate celelalte 4 componente menționate anterior



Microcontrolere Off the Shelf



Most Popular Microcontrollers, by The Engineering Projects,
<https://www.theengineeringprojects.com/2018/03/introduction-to-microcontrollers.html>



A Trainers Toolkit To Foster STEM Skills Using
Microcontroller Applications

Project No. 2019-1-RO01-KA202-063965

This project has been funded with support from the European Commission. The content reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by the
Erasmus+ Programme
of the European Union

Understanding Microcontrollers

Sumar

Acest slide încheie prezentarea introductivă a microcontrolerelor.

Până acum ar fi trebuit să aflați ceva despre următoarele subiecte:

1. Ce sunt Microcontrolerele
2. O idee generală despre modul în care funcționează
3. Diferitele aplicații pentru care sunt proiectate
4. Considerații atunci când alegeți un microcontroler

