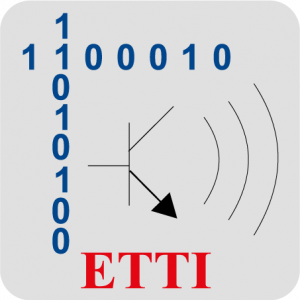
UNIVERSITATEA POLITEHNICA DIN BUCUREŞTI

**Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei**

** **

**PROIECT 2 – ELECTRONICĂ PROGRAMABILĂ**

**Nivelă electronică (boloboc) cu afișajul “bulei” pe LED-uri în format pozitiv sau negativ**

**Grupa:** 434C

**Student:** Petrache Cristina-Georgiana

**Bucureşti 2022**

***DESCRIERE MICROCONTROLER ATMEL ATMEGA164A***

Atmel® ATmega164A este un microcontroler CMOS de 8 biți cu putere redusă bazat pe arhitectura AVR cu RISC îmbunătățit. Prin executarea instrucțiunilor puternice într-un singur ciclu de ceas, ATmega164A atinge performanțe apropiate de 1MIPS pe MHz. Acest lucru permite proiectantului de sistem să optimizeze dispozitivul pentru consum de alimentare vs. viteza de procesare.

Nucleul Atmel AVR® combină un set bogat de instrucțiuni cu 32 de registre de lucru cu scop general. Toate cele 32 de registre sunt conectate direct la unitatea logică aritmetică (ALU), permițând două registre independente să fie accesate într-o singură instrucțiune executată într-un ciclu de ceas.

ATmega164A oferă următoarele caracteristici: 16Kbytes capacitate memorie Flash cu capacități Read-While-Write, 512bytes capacitate memorie EEPROM, 1Kbytes capacitate memorie SRAM, 32 linii de I/O de uz general, 32 registre de lucru cu scop general, contor în timp real (Real Time Counter - RTC), trei timere : 2 pe 8 biți și unul pe 16 biți, 6 canale PWM, două porturi seriale USART programabile, I2C, 8 adaptoare ADC pe 10 biți cu față de intrare diferențială opțională cu câștig programabil, un Watchdog Timer programabil cu oscilator intern, un port serial SPI, tensiune alimentare: 1.8-5.5 V, frecvență de funcționare 20 MHz, frecvență de sincronizare 20 MHz.

ISP On-chip Flash permite reprogramarea memoriei programului în sistem printr-o interfață serială SPI, de către un programator convențional de memorie nonvolatilă sau de către un program On-chip Boot care rulează pe nucleul AVR. Programul Boot poate folosi orice interfață pentru a descărca programul de aplicație în memoria Flash a aplicației. Software-ul din secțiunea Boot Flash va continua să funcționeze în timp ce secțiunea Flash a aplicației este actualizată, oferind operația Read-While-Write. Prin combinarea unui procesor RISC CPU pe 8 biți cu In-System Self-Programmable Flash pe un cip monolitic, Atmel ATmega164A este un microcontroler puternic care oferă o soluție extrem de flexibilă și rentabilă pentru multe aplicații de control încorporate.

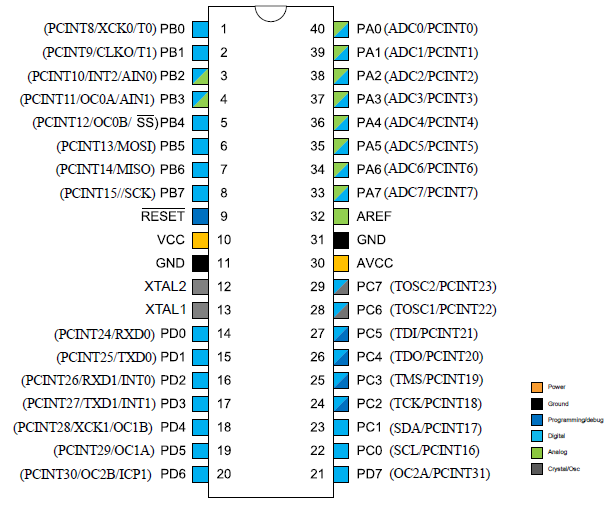
ATmega164A este susținut de o suită completă de instrumente de dezvoltare a programelor și sistemelor, incluzând: Compilatoare C, Asamblări Macro, Debugger / Simulatoare de program, Emulatoare și kituri de evaluare**.**

***SCHEMA BLOC ATMEGA164A***

Diagram

Description automatically generated

***CONFIGURAȚII PINI ATMEGA164A***



**1.** **VCC** Tensiune de alimentare digitală.

**2. GND** Împământare.

**3. Port A (PA [7: 0])** Acest port servește ca intrări analogice la convertorul analog-digital. Acesta este un port de I/O bidirecțional pe 8 biți, cu rezistențe interne de tragere, selectabile individual pentru fiecare bit.

**4. Port B (PB [7: 0])** Acesta este un port de I/O bidirecțional pe 8 biți, cu rezistențe interne de tragere, selectabile individual pentru fiecare bit.

**5. Port C (PC [7: 0])** Acesta este un port de I/O bidirecțional pe 8 biți, cu rezistențe interne de tragere, selectabile individual pentru fiecare bit.

**6. Port D (PD [7: 0])** Acesta este un port de I/O bidirecțional pe 8 biți, cu rezistențe interne de tragere, selectabile individual pentru fiecare bit.

**7. RESET** Resetați intrarea. Un nivel scăzut pe acest pin mai lung decât lungimea minimă a impulsului va genera o resetare, chiar dacă ceasul nu funcționează. Impulsurile mai scurte nu sunt garantate pentru a genera o resetare.

**8. XTAL1** Intrare în amplificatorul oscilator inversor și intrare în circuitul intern de funcționare a ceasului.

**9. XTAL2** Ieșire din amplificatorul oscilator inversor.

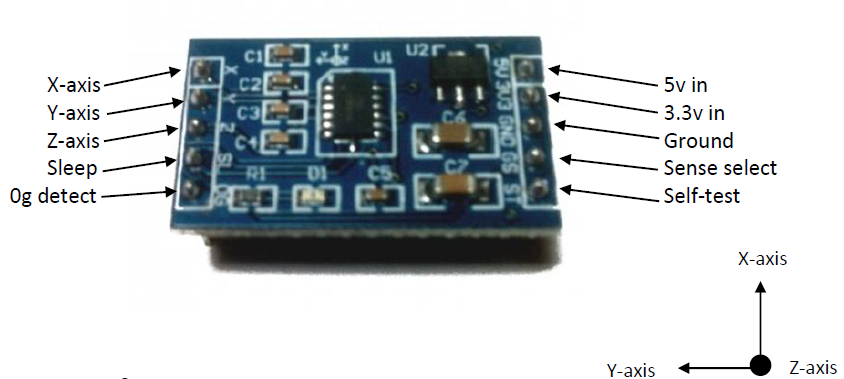
**10. AVCC** este pinul de tensiune de alimentare pentru portul A și convertorul analog-digital. Ar trebui să fie conectat extern la VCC, chiar dacă ADC nu este utilizat. Dacă se utilizează ADC, acesta trebuie conectat la VCC printr-un filtru trece-jos.

**11. AREF** Acesta este pinul de referință analogic pentru convertorul analog-digital.

***COMPONENTE HARDWARE UTILIZATE***

* Senzor MMA7361 (accelerometru pe 3 axe pentru măsurarea accelerației → determinarea unghiului de înclinație al plăcuței)
* 1 LED alb (se aprinde atunci când bolobocul este centrat; clipește atunci când bolobocul este centrat doar pe una dintre axe)
* 4 LED-uri de culoare verde (se aprind atunci când bolobocul este înclinat la un unghi mai mic de 30 de grade)
* 4 LED-uri de culoare galbenă (se aprind atunci când bolobocul este înclinat la un unghi de cel puțin 30 de grade)
* 4 LED-uri de culoare roșie (se aprind atunci când bolobocul este înclinat la un unghi de cel puțin 45 de grade)
* 6 rezistoare cu valoarea rezistenței de 100 Ω, 6 rezistoare cu valoarea rezistenței de 220 Ω, 2 rezistoare cu valoarea rezistenței de 330 Ω (pentru a limita curentul care trece prin LED-uri, respectiv prin pinul de SLEEP al senzorului)

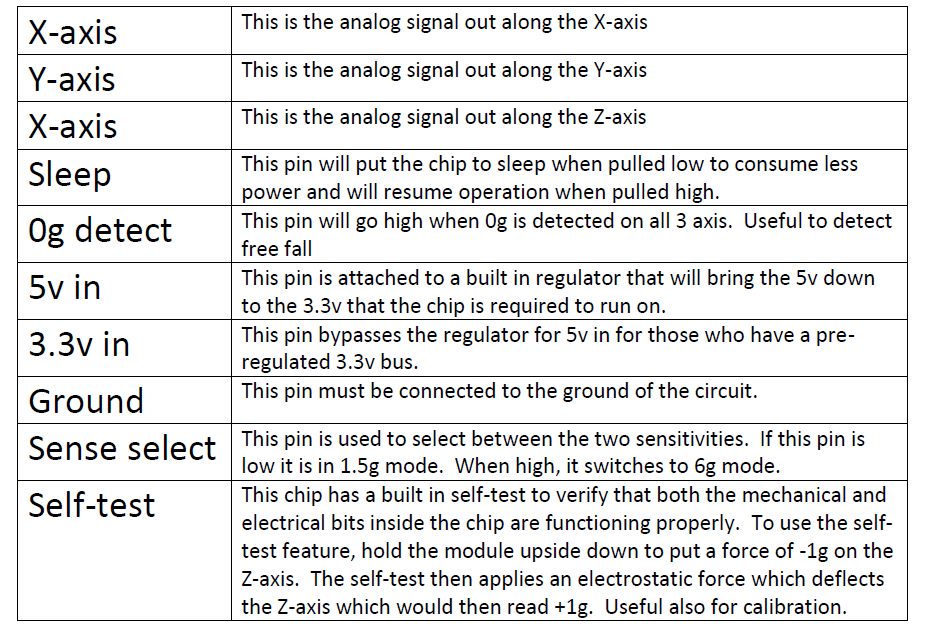
***SENZOR MMA7361***



**Diagrama circuitului**



**Descrierea pinilor**



* Senzorul are axele X și Y conectate la porturile PA0 (ADC0 – axa Y) și PA1 (ADC1 – axa X) ale microcontrolerului. Acesta ia datele din accelerometru și le realizează conversia din analog în digital. Porturile sunt folosite ca intrări.
* VCC de la senzor este conectat la VCC de la microcontroler.
* GND de la senzor este conectat la GND de la microcontroler.
* Pinul de SLEEP al senzorului este conectat la portul PB7 (folosit ca ieșire) al microcontrolerului. Acesta are legată o rezistență pentru a micșora curentul care trece prin pinul de SLEEP.

***LED-URI ȘI REZISTOARE***

* LED-urile de pe axa X sunt conectate la porturile PB0 – PB5 ale microcontrolerului care sunt setate ca ieșiri. Acestea au legate 6 rezistoare cu valoarea rezistenței de 100 Ω pentru a limita curentul care circulă prin LED-uri pentru a nu se arde.
* LED-urile de pe axa Y sunt conectate la porturile PC0 – PC5 ale microcontrolerului care sunt setate ca ieșiri. Acestea au legate 6 rezistoare cu valoarea rezistenței de 220 Ω pentru a limita curentul care circulă prin LED-uri pentru a nu se arde.
* LED-ul central este conectat la portul PC7 al microcontrolerului care este setat la ieșire. Acesta are legată un rezistor cu valoarea rezistenței de 330 Ω pentru a limita curentul care circulă prin LED pentru a nu se arde.

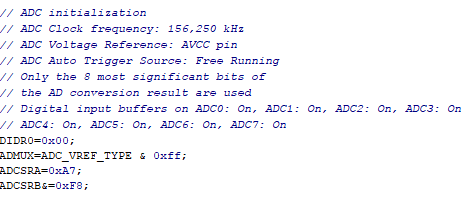
***DESCRIERE GENERALĂ A FUNCȚIONĂRII BOLOBOCULUI***

Senzorul MMA7361 este alimentat, măsoară accelerația pentru a determina unghiul de înclinație al plăcuței. Datele din accelerometru sunt transmise microcontrolerului care realizează conversia din analog în digital prin intermediul porturilor PA0 (ADC0) și PA1 (ADC1). Pe baza datelor respective am realizat un algoritm care controlează LED-urile în funcție de axă și unghiul de înclinație.

***EXPLICAȚII COD SURSĂ***

Pentru implementarea algoritmului necesar funcționării bolobocului am pornit de la codul pus la dispoziție pe platforma Microsoft Teams.

1. **Configurare ADC**



Pentru citirea datelor de la accelerometru am activat porturile ADC ale microcontrolerului ca input.

Text, letter

Description automatically generated

Am creat funcția de citire și conversie analog-digitală a datelor ce ajung la porturile ADC0 (PA0) și ADC1 (PA1).

Graphical user interface, text, application, email

Description automatically generated

În bucla principală, am apelat funcția read\_adc în funție de cazul în care mă aflu (nu pot lua datele de la ambele axe în același timp, am făcut câte un caz în care iau datele de la fiecare axă în parte prin intermediul câte unui case). Pentru axa X, citesc datele de la portul ADC1, iar pentru axa Y, citesc de la portul ADC0.





1. **Configurarea porturilor B și C**

Porturile B și C sunt inițializate ca output.

Text, letter

Description automatically generated

Portul B este folosit astfel:

* de la PORTB.0 până la PORTB.5 sunt dedicate ledurilor de pe axa X
  + - PORTB.0 → PORTB.2 pentru un sens de rotație
    - PORTB.3 → PORTB.5 pentru celălalt sens de rotație
* PORTB.7 este folosit pentru a dezactiva modul Sleep al accelerometrului
  + - Low = Sleep
    - High = Wake

Portul C este folosit astfel:

* de la PORTC.0 pana la PORTC.5 sunt dedicate ledurilor de pe axa Y
  + - PORTC.0 → PORTC.2 pentru un sens de rotație
    - PORTC.3 → PORTC.5 pentru celălalt sens de rotație
* PORTC.7 este folosit pentru ledul din centru

1. **Codul pentru leduri**

Pentru controlul ledurilor, am folosit în bucla principală un switch cu 2 case-uri (unul pentru axa X și unul pentru axa Y).

Inițial, în case se citește valoarea de la accelerometru pentru axa respectivă, urmată de un delay de 10 milisecunde pentru stabilizare. Apoi urmează 7 sintaxe de condiții logice care controlează jocul de lumini al ledurilor, în funcție de unghiul de înclinație al accelerometrului, astfel:

* ALB → centrat (cel puțin una dintre axe)

Graphical user interface, text

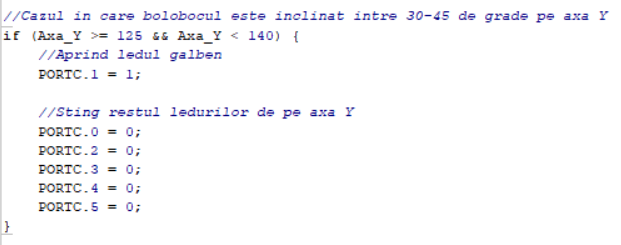
Description automatically generated

* VERDE → până la 30 de grade

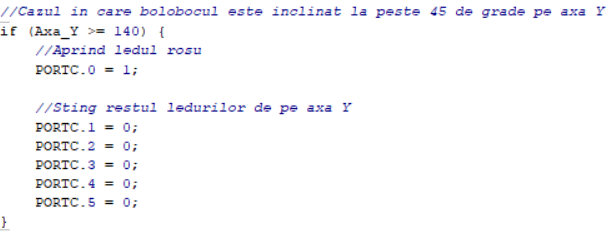
Graphical user interface, text, application

Description automatically generated

* GALBEN → între 30 si 45 de grade



* ROȘU → peste 45 de grade



La final am adăugat un delay de 200 de milisecunde pentru ca tranziția între leduri, în cazul schimbării unghiului, să fie suficient de lentă, dar și suficient de rapidă.