UNIVERSITATEA POLITEHNICA DIN BUCUREŞTI
FACULTATEA DE ELECTRONICĂ, TELECOMUNICAŢII ŞI TEHNOLOGIA INFORMAŢIEI

Sistem Automat de Sortare a Monedelor

Student: Rusu Ioana - Sorina

Grupa: 422E

2024-2025

CUPRINS

LISTA FIGURILOR

LISTA TABELELOR

INTRODUCERE

Mașinile de sortat și numărat monede sunt instrumente foarte utile pentru manipularea monedelor de diferite valute. Ele sunt folosite frecvent în bănci, puncte de colectare sau în alte situații în care este nevoie de o procesare rapidă și eficientă a unor cantități mari de monede. Principalul lor avantaj este reducerea timpului necesar sortării manuale, precum și eliminarea erorilor umane, ceea ce contribuie la o acuratețe mai mare.

Aceste sisteme pot fi echipate cu funcții suplimentare, cum ar fi numărarea monedelor, calcularea sumei totale sau detectarea monedelor false sau neconforme. Identificarea fiecărei monede se face pe baza unor caracteristici bine definite, cum ar fi diametrul, greutatea sau grosimea, iar apoi aceasta este direcționată către un compartiment corespunzător.

În acest proiect ne propunem să realizăm un sistem automat de sortare a monedelor, care folosește atât circuite logice combinaționale, cât și secvențiale, pentru ca monedele să fie detectate, identificate și sortate în mod automat. Sistemul include senzori optici sau de lumină pentru măsurarea diametrului, senzori de greutate pentru determinarea masei, precum și indicatori vizuali și sonori, cum ar fi afișaje cu 7 segmente, LED-uri și buzzere.

În continuare, ne vom concentra predominant pe partea software a sistemului, folosind programul CodeVision AVR (versiunea 4.03), iar simulările vor fi realizate cu ajutorul AVR Studio (versiunea 4.13). Sistemul va fi testat în mediu virtual folosind o placă cu microcontroler ATmega164P.

CAPITOLUL 1: DESCRIEREA GENERALĂ A SISTEMULUI

1.1 PREZENTAREA SISTEMULUI

Sistemul descris are rolul de a identifica monede de diferite valori (5 bani, 10 bani și 50 bani) pe baza caracteristicilor fizice ale acestora, mai exact diametrul și greutatea. Pentru această identificare sunt utilizați doi senzori principali: un **senzor optic**, care determină dimensiunea monedei (diametrul), și un **senzor de greutate**, care măsoară masa acesteia. Informațiile oferite de cei doi senzori sunt prelucrate de către un **microcontroler**, iar pe baza valorilor obținute se determină tipul monedei.

După identificare, moneda este direcționată către unul dintre cele trei **compartimente** specifice (pentru 5 bani, 10 bani sau 50 bani) printr-un sistem de **clapete controlate electronic**. În cazul în care moneda introdusă nu corespunde valorilor recunoscute de sistem (dimensiune sau greutate incorectă), aceasta este deviată către un **compartiment de eroare**, iar utilizatorul este notificat corespunzător.

Sistemul este alcătuit din următoarele componente:

- **Microcontroler ATMega164P** – elementul central al sistemului, care gestionează toate procesele: citirea senzorilor, analiza datelor, comanda clapetelor, afișarea rezultatului și notificările.
- **Senzor optic** – detectează diametrul monedei.
- **Senzor de greutate** – măsoară masa monedei pentru validare suplimentară.
- **Afișaj cu 7 segmente** – indică valoarea ultimei monede detectate sau un mesaj de eroare.
- **Patru clapete acționate electronic** – direcționează moneda către compartimentul corespunzător valorii sale.
- **Buzzer** – semnal sonor care se activează în cazul detectării unei monede invalide.
- **Două LED-uri (roșu și verde)** – oferă feedback vizual: LED-ul verde se aprinde când sistemul este pregătit pentru monedă, iar cel roșu când moneda este respinsă.

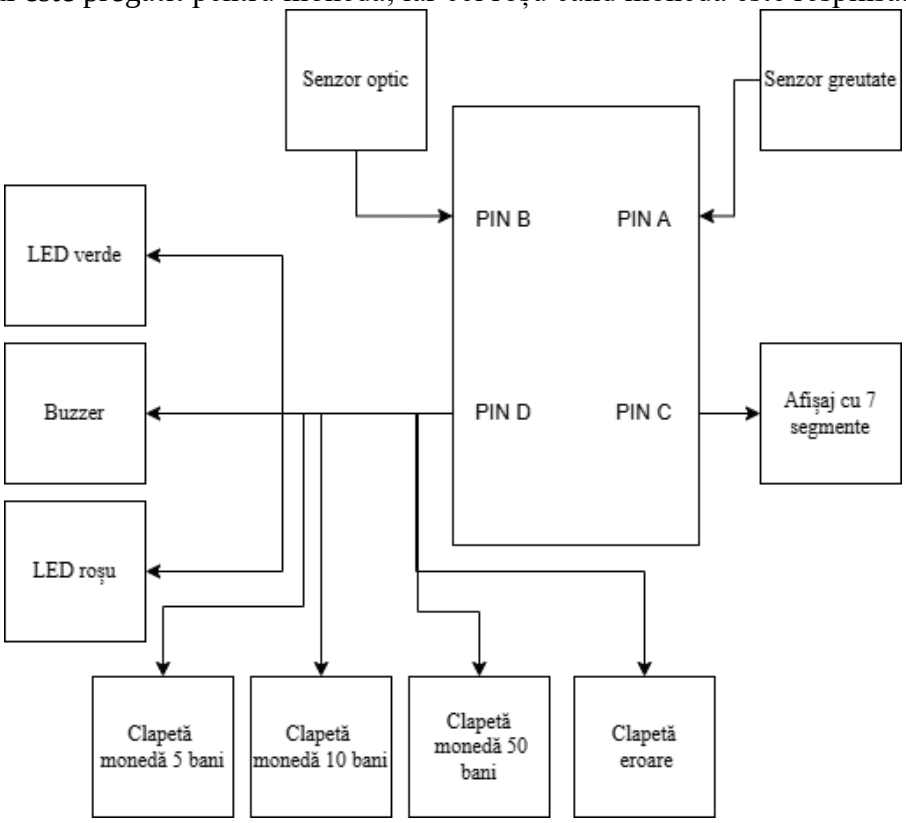


Fig. 2.1 Schema bloc a sistemului

1.2 MOD DE FUNCŢIONARE

Automatul de stare finit (FSM) controlează succesiunea stărilor prin care trece sistemul în procesul de sortare a monedelor. Acesta asigură o tranziție logică și coerentă între fiecare etapă a procesului, începând de la detectarea monedei și până la finalizarea sortării acesteia.

- **Starea 1 – Așteptare:** Sistemul se află în stare de repaus, pregătit să primească o nouă monedă. Pentru a semnaliza acest lucru utilizatorului, se aprinde LED-ul verde.
- **Starea 2 – Măsurare:** În această etapă, senzorii activi măsoară caracteristicile fizice ale monedei: diametrul (cu ajutorul senzorului optic) și greutatea (cu ajutorul senzorului de masă).
- **Starea 3 – Comparare:** Datele obținute de la senzori sunt analizate, iar sistemul stabilește valoarea monedei în funcție de caracteristicile predefinite (ex. 5 bani, 10 bani, 50 bani).
- **Starea 4 – Direcționare:** Pe baza rezultatului obținut, mecanismul de sortare acționează una dintre cele trei clapete pentru a direcționa moneda către compartimentul corespunzător. Afișajul cu 7 segmente indică valoarea monedei detectate. În cazul în care moneda nu este recunoscută (valori incorecte de greutate sau diametru), se activează LED-ul roșu și buzzer-ul pentru a semnaliza o eroare, iar moneda este trimisă în compartimentul de eroare.
- **Starea 5 – Reset:** Sistemul revine la starea inițială de așteptare, pregătit pentru a procesa următoarea monedă introdusă.

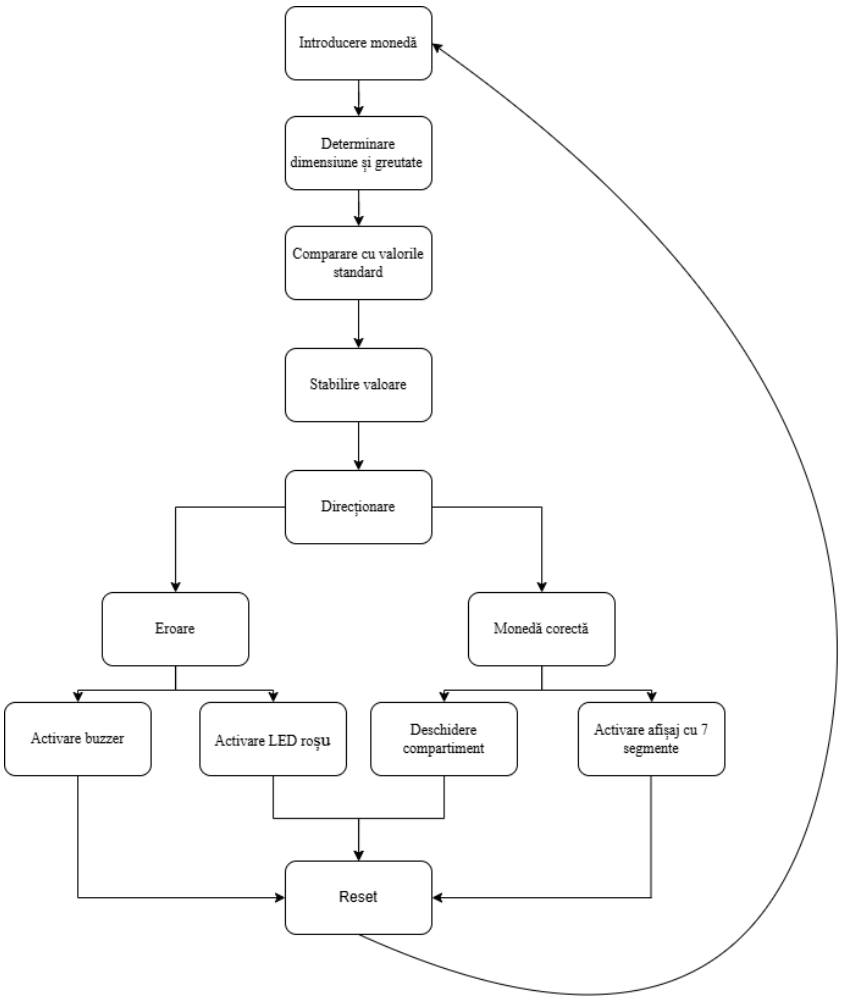


Fig. 1.2 Diagrama cazurilor de utilizare a sistemului

CAPITOLUL 2: PROIECTARE ȘI IMPLEMENTARE

2.1 ETAPA DE PROIECTARE

Greutățile și diametrele utilizate în acest proiect corespund specificațiilor actuale ale monedelor de 5, 10 și 50 de bani aflate în circulație în România.

Moneda	Greutate (g)	Diametru (mm)
5 bani	2.81	18.2
10 bani	4	22.5
50 bani	6.1	23.74

Tabelul 2.1 Greutățile și diametrele specifice fiecărei monede

Pentru scrierea codului, am utilizat aplicația **CodeVision AVR**, codul fiind scris în limbajul de programare **C**. La crearea proiectului, au fost realizate următoarele setări:

- **Chip → Chip Settings → Chip**: am selectat opțiunea *ATmega164P*. Din aceeași secțiune, am modificat frecvența de ceas la **10 MHz**, aceasta influențând direct viteza de execuție a instrucțiunilor.
- **Port → Port Settings**: fiecare port a fost configurat în funcție de rolul său în procesul de programare și control al componentelor hardware.

Port A								
Data Direction	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	In	In	In	In	In	In	In	In
Pullup/Output Value	P	P	P	P	P	P	P	P

Tabelul 2.2 Configurarea portului A

Port B								
Data Direction	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	In	In	In	In	In	In	In	In
Pullup/Output Value	P	P	P	P	P	P	P	P

Tabelul 2.3 Configurarea portului B

Port C								
Data Direction	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	Out	Out	Out	Out	Out	Out	Out	Out
Pullup/Output Value	1	1	1	1	1	1	1	1

Tabelul 2.4 Configurarea portului C

Port D								
Data Direction	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
	Out	Out	Out	Out	Out	Out	Out	Out
Pullup/Output Value	1	1	1	1	1	1	1	1

Tabelul 2.5 Configurarea portului D

- **Timers/Counters → Clock Value → Timer0 → 9.766 kHz**: este utilizat un semnal de ceas de 9.766 kHz pentru *Timer0*. Bifarea opțiunii **Overflow Interrupt** permite generarea unei întreruperi de fiecare dată când *Timer0* ajunge la valoarea maximă și revine la zero. Aceasta întrerupere este tratată printr-o funcție specială numită *Interrupt Service Routine* (**ISR**). În plus, valoarea **Timer Value → 3Ch** (în hexazecimal) definește punctul de la care începe numărătoarea, astfel încât întreruperea să fie declanșată aproximativ la fiecare **20 ms**.

Configurația completă este prezentată în **Anexa 1.**

2.2 SEMNIFICAŢIA PORTURILOR

Fiecare port are un rol bine definit în buna funcționare a sistemului.

Portul A contribuie la determinarea **greutății** monedei. Biții 0, 1 și 2 rețin greutățile specifice monedelor de 5, 10 și respectiv 50 de bani. Astfel:

- Dacă senzorul de greutate detectează **2,81 grame**, se activează **pinul 0**.
- Dacă detectează **4 grame**, se activează **pinul 1**.
- Dacă detectează **6,1 grame**, se activează **pinul 2**.
- Dacă detectează **altă greutate**, se activează **pinul 3**.

Port A							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Greutate moneda 5 bani	Greutate moneda 10 bani	Greutate moneda 50 bani	Alt caz				

Tabelul 2.5 Semnificatia bitiilor din portul A

Portul B este responsabil pentru determinarea **diametrului** monedei. Biții 0, 1 și 2 rețin diametrele caracteristice monedelor de 5, 10 și respectiv 50 de bani. Astfel:

- Dacă senzorul optic detectează **18,2 mm**, se activează **pinul 0**.
- Pentru **22,5 mm**, se activează **pinul 1**.
- Pentru **23,75 mm**, se activează **pinul 2**.
- Dacă este detectat **un alt diametru**, se activează **pinul 3**.

Port B							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Diametru moneda 5 bani	Diametru moneda 10 bani	Diametru moneda 50 bani	Alt caz				

Tabelul 2.6 Semnificatia bitiilor din portul B

Portul C controlează afișajul cu 7 segmente, care permite vizualizarea valorii monedei.

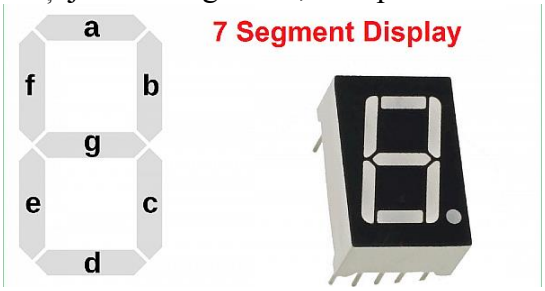


Fig. 2.1 Notații afișaj 7 segmente

Port C							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
a	b	c	d	e	f	g	

Tabelul 2.7 Semnificația bitilor din portul C

În funcție de valoarea monedei, vor fi activați următorii pini pentru a afișa cifra corespunzătoare:

- Pentru valoarea de **5 bani** → apare cifra 1 (segmentele b și c) → 1001 1111
- Pentru valoarea de **10 bani** → apare cifra 2 (segmentele a, b, d, e, g) → 0010 0011
- Pentru valoarea de **50 bani** → apare cifra 3 (segmentele a, b, c, d, g) → 00001101

Portul D controlează afișajele vizuale și semnalele audio, în funcție de rezultatul procesului de sortare:

- Dacă sistemul este **pregătit** să primească o nouă monedă, se aprinde **LED-ul verde** (bitul 0 al portului D).
  - Dacă este detectată o **eroare** (monedă invalidă sau necunoscută), se activează **LED-ul roșu** și **buzzer-ul** (biții 1 și 2) si **deschisa clapeta corespunzatoare (Bitul 3** – clapetă pentru **eroare**)
- În cazul unei detectări corecte, **este deschisa clapeta corespunzatoare**, în funcție de activarea unuia dintre următorii biți:
  - **Bitul 4** – clapetă pentru moneda de **5 bani**,
  - **Bitul 5** – clapetă pentru moneda de **10 bani**,
  - **Bitul 6** – clapetă pentru moneda de **50 bani**.

Port D							
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
LED Verde	Buzzer	LED Rosu	Clapetă eroare	Clapetă 5 bani	Clapetă 10 bani	Clapetă 50 bani	

Tabelul 2.7 Semnificatia bitiilor din portul D

2.3 IMPLEMENTAREA CODULUI SURSA

2.3.1 DECLARAREA VARIABILELOR UTILIZATE ÎN PROGRAM

Pentru implementarea funcționalităților sistemului de sortare automată a monedelor, au fost definite următoarele variabile:

- `unsigned char Q = 0;` - reprezintă starea curentă a **automatului finit (FSM)** care controlează sistemul. Fiecare valoare a variabilei corespunde unei etape din procesul de sortare (ex: așteptare, măsurare, comparare etc.).
- `int ag, ad, bg, bd, cg, cd;` - aceste variabile sunt utilizate pentru stocarea valorilor de **greutate (g)** și **diametru (d)** pentru fiecare tip de monedă:
  - `ag`, `ad` → greutate și diametru pentru moneda de **5 bani**
  - `bg`, `bd` → greutate și diametru pentru moneda de **10 bani**
  - `cg`, `cd` → greutate și diametru pentru moneda de **50 bani**

Pentru fiecare monedă se verifică dacă greutatea și diametrul detectate corespund valorilor așteptate:

- `char greutate_5bani = 0;`
- `char diametru_5bani = 0;`

- `char greutate_10bani = 0;`
- `char diametru_10bani = 0;`
- `char greutate_50bani = 0;`
- `char diametru_50bani = 0;`

Aceste variabile sunt folosite pentru a valida fiecare caracteristică a monedelor. Dacă valoarea corespunde, variabila asociată va fi setată la **1** (adevărat), altfel rămân **0**.

După ce ambele condiții (greutate și diametru) sunt validate pentru o monedă, variabila aferentă este setată la **1**, semnalând că moneda respectivă a fost **recunoscută corect**.

- `char moneda_5bani = 0;`
- `char moneda_10bani = 0;`
- `char moneda_50bani = 0;`

Pentru a semnala o **monedă invalidă sau necunoscută,** am folosit:

- `char eroare = 0;`

În cazul în care niciuna dintre condițiile de validare nu este îndeplinită, această variabilă va fi activată.

Într-un sistem real, greutatea si diametrul ar fi furnizate de către senzorii fizici. În cadrul simulării în AVR Studio, aceste valori vor fi **introduse manual**, pentru a reproduce comportamentul senzorilor în condiții de test.

- `float greutate_introdusa, diametru_introdus;`

2.3.2 IMPLEMENTAREA CAZURILOR

Sistemul este structurat pe baza unui **automat finit cu 5 stări** (numerotate de la 0 la 4), fiecare implementată cu ajutorul unei instrucțiuni switch-case. Trecerea de la o stare la alta este controlată prin variabila Q.

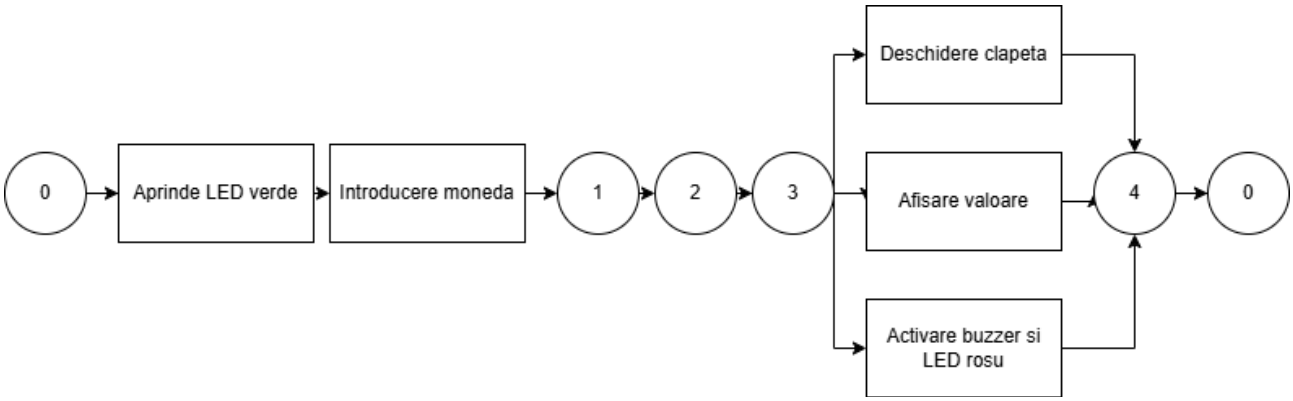


Fig. 2.2 Diagrama secvențiale corespunzătoare codului sursă

Starea 0 – Așteptare

În această stare, sistemul este pregătit să primească o monedă nouă. LED-ul verde este aprins pentru a semnaliza utilizatorului disponibilitatea sistemului. Pentru simulare, valorile greutății și diametrului sunt introduse manual.

```
case 0: //asteptare
      Q = Q+1;

      PORTD.0 = 0; //semnaleaza ca poate primi o noua moneda
      greutate_introdusa = 2.81;
      diametru_introdus = 18.2;

break;
```

Starea 1 – Măsurare

Sistemul compară valorile introduse cu cele standard și setează bitii corespunzători în porturile A și B, în funcție de valorile detectate. Ulterior, valorile sunt salvate în variabilele auxiliare pentru validare.

```
case 1: //masurare
      Q = Q+1;

      PORTD.0 = 1;
      if(greutate_introdusa == 2.81)
            PINA.0 = 1;
      else if(greutate_introdusa == 4)
            PINA.1 = 1;
      else if (greutate_introdusa == 6.1)
            PINA.2 = 1;
      else
            PINA.3 = 1;

      if(diametru_introdus == 18.2)
            PINB.0 = 1;
      else if(diametru_introdus== 22.5)
            PINB.1 = 1;
      else if (diametru_introdus == 23.75)
            PINB.2 = 1;
      else
            PINB.3 = 1;

      ag = PINA.0;
      ad = PINB.0;

      bg = PINA.1;
      bd = PINB.1;

      cg = PINA.2;
      cd = PINB.2;

break;
```

Starea 2 – Comparare

În această stare are loc validarea fiecărei monede prin verificarea combinațiilor de greutate și diametru. Fiecare pereche validă va activa o variabilă asociată monedei respective, altfel se va constata o eroare.

```
case 2: //comparare
      Q = Q+1;
      if(ag == 1) //a detectat greutatea monedei de 5 bani
            greutate_5bani ++;
      if(ad == 1) //a detectat diametrul monedei de 10 bani
            diametru_5bani ++;
      if(bg == 1) //a detectat greutatea monedei de 10 bani
            greutate_10bani ++;
      if(bd == 1) //a detectat diametrul monedei de 10 bani
            diametru_10bani ++;
      if(cg == 1) //a detectat greutatea monedei de 50 bani
            greutate_50bani ++;
      if(cd == 1) //a detectat diametrul monedei de 50 bani
            diametru_50bani ++;

break;
```

Starea 3 – Direcționare

Sistemul decide care monedă a fost introdusă și activează clapeta corespunzătoare. Tot aici se face afișarea pe 7 segmente, iar în caz de eroare sunt activate semnalele acustice și vizuale (buzzer și LED roșu).

```
case 3:      //directionare

      Q = Q+1;

      //detectare

      if(((greutate_5bani == 1) && (diametru_5bani == 1)) == 1)
            moneda_5bani ++;    //a fost detectata moneda de 5 bani
      else if(((greutate_10bani == 1) && (diametru_10bani == 1)) == 1)
            moneda_10bani ++;    //a fost detectata moneda de 10 bani
      else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
            moneda_50bani ++;    //a fost detectata moneda de 50 bani
      else
            eroare++;

//directionare

      if(moneda_5bani == 1){
            PORTD.4 = 0;          //DESCHIDEM CLAPETA 5 BANI
            PORTC = 0x9F;}
      else if (moneda_10bani == 1){
            PORTD.5 = 0;                //DESCHIDEM CLAPETA 10 BANI
            PORTC = 0x23;}
      else if (moneda_50bani == 1){
            PORTD.6 = 0;                      //DESCHIDEM CLAPETA 50 BANI
            PORTC = 0x0D;}
      else{
            PORTD.1 = 0;   //APRINDEM BUZZER
            PORTD.2 = 0;   //APRINDEM LED
            PORTD.3 = 0;   //APRINDEM CLAPETA EROARE
      }
break;
```

Starea 4 – Resetare

Se resetează toate valorile și sistemul revine la starea inițială, pregătit pentru o nouă monedă.

```
case 4: //RESET
      Q = 0;
      PORTA = 0xFF;
      PORTB = 0XFF;
      PORTC = 0XFF;
      PORTD = 0XFF;
      greutate_5bani = 0;
      diametru_5bani = 0;
      greutate_10bani = 0;
      diametru_10bani = 0;
      greutate_50bani = 0;
      diametru_50bani = 0;
      moneda_5bani = 0;
      moneda_10bani = 0;
      moneda_50bani = 0;
      eroare = 0;
break;
```

Codul sursă complet al programului este inclus în **Anexa 2**.

CAPITOLUL 3. TESTAREA FUNCȚIONALITĂȚII SISTEMULUI

Pentru a verifica funcționalitatea completă a sistemului de sortare a monedelor, au fost realizate teste în cadrul simulării din AVR Studio, urmărind fiecare etapă a automatului de stare finit (FSM) și reacțiile asociate ale sistemului. După fiecare testare, în momentul introducerii altor valori, am folosit opţiunea "*Build all*" din CodeVision AVR. Testarea se poate face și prin comentarea valorilor introduse manual și setarea biţilor în mod direct.

3.1 TESTARE MONEDA 5 BANI

Starea 0 – Așteptare:

La inițializarea sistemului, LED-ul verde se aprinde, semnalând disponibilitatea pentru introducerea unei monede. În fereastra *Watch*, se observă modificarea valorilor variabilelor `greutate_introdusa` și `diametru_introdus`, care trec de la 0 la valorile 2.81, respectiv 18.2, conform specificațiilor pentru moneda de 5 bani.

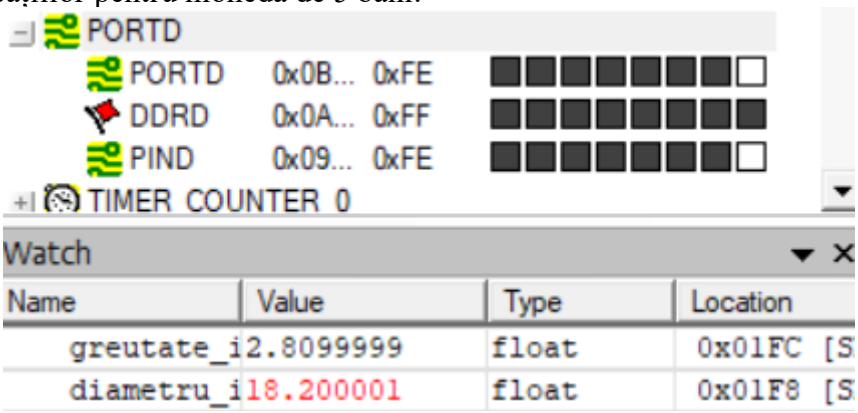


Fig. 3.1 LED verde activ + valori introduse

Starea 1 – Măsurare:

LED-ul verde se stinge. Sistemul detectează greutatea și diametrul specifice, activând pinii `PINA.0` și `PINB.0`. Tot în această etapă, variabilele `ag` și `ad` își modifică valoarea, preluând starea acestor pini.

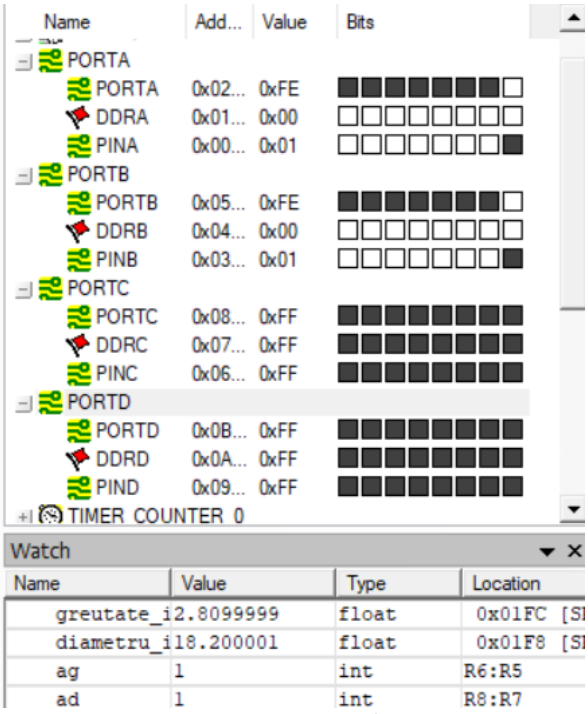


Fig 3.2 Pini activi + modificare ag,ad

Starea 2 – Comparare:

Se verifică combinația greutate si diametru. Fiind o combinație validă pentru moneda de 5 bani, variabilele `greutate_5bani` și `diametru_5bani` sunt incrementate, confirmând recunoașterea acesteia.

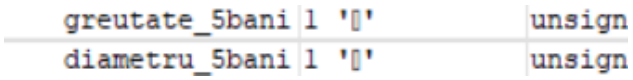


Fig 3.3 Incrementarea variabilei validare greutate/diametru

Starea 3 – Direcționare:

Moneda este validată complet prin incrementarea variabilei `moneda_5bani`.

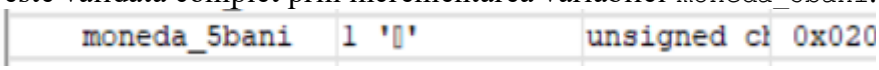


Fig 3.4 Validare moneda

Se deschide clapeta asociată acestei monede (activare `PORTD.4`) și se setează valorile corespunzătoare pe portul `PORTC`, astfel încât să fie afișată valoarea monedei pe display-ul cu 7 segmente.

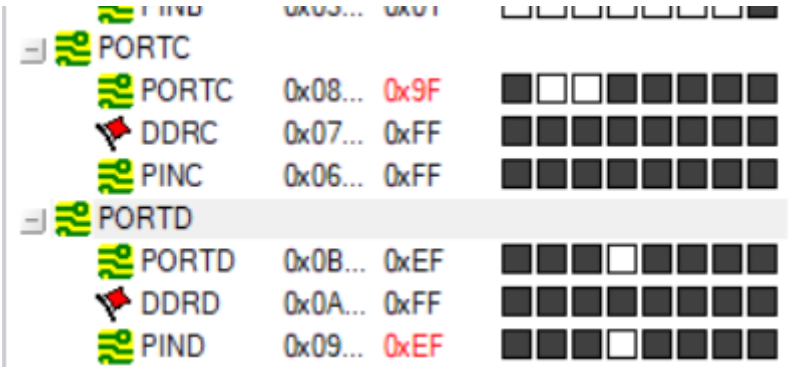


Fig 3.5 clapetă activă + afișaj 5 bani

Starea 4 – Reset:

Sistemul revine la valorile inițiale: toți pinii sunt dezactivați, variabilele de validare și eroare sunt resetate, iar sistemul intră din nou în stare de așteptare.

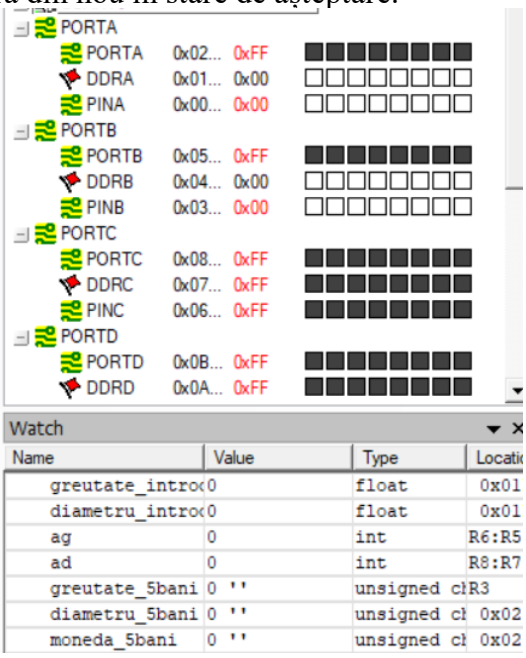


Fig 3.6 Reset complet variabile

3.2 TESTARE MONEDA INVALIDĂ

Starea 0 – Așteptare:

Sistemul pornește în starea de așteptare, semnalizând disponibilitatea prin aprinderea LED-ului verde (`PORTD.0 = 0`). În simulare, introducem manual valorile `greutate_introdusa = 2` și `diametru_introdus = 3`.

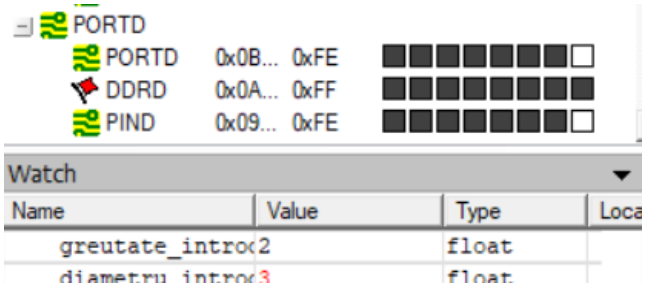


Fig. 3.7 LED verde activ + valori introduse

Starea 1 – Măsurare:

LED-ul verde se stinge. Sistemul compară valorile introduse cu cele prestabilite. Nicio valoare nu corespunde cu cele pentru monedele valide (2.81g, 4g, 6.1g / 18.2mm, 22.5mm, 23.75mm), astfel că se activează pinii de **eroare** `PINA.3` și `PINB.3`. Variabilele `ag`, `bg`, `cg`, `ad`, `bd`, `cd` rămân 0.

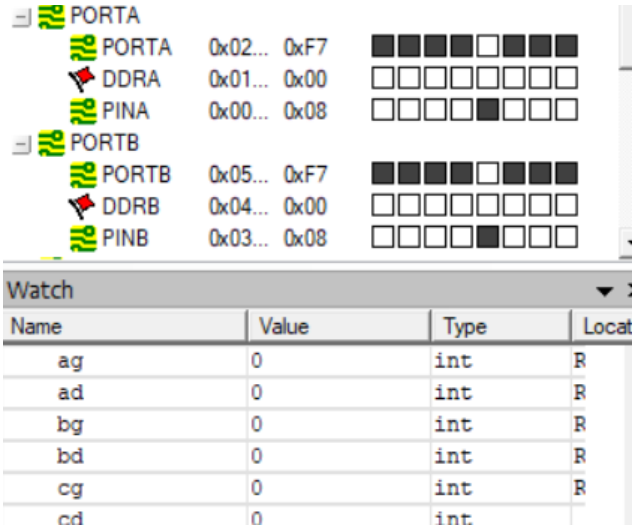


Fig 3.8 Pini activi + modificare ag,ad,bg,bd,cg,cd

Starea 2 – Comparare:

Nici o combinație validă de greutate + diametru nu este detectată, deci niciuna dintre variabilele `greutate_Xbani` sau `diametru_Xbani` nu este incrementată.

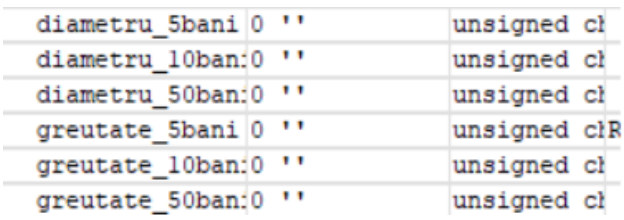


Fig 3.9 Variabilele validare greutate/diametru

Starea 3 – Direcționare:

Pentru că nu a fost validată nicio monedă, sistemul consideră că este vorba despre o eroare.

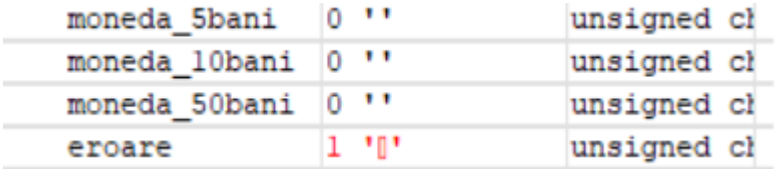


Fig 3.10 Eroare

Variabila `eroare` este incrementată, iar sistemul activează:

- LED-ul roșu (`PORTD.2 = 0`)
- Buzzer-ul (`PORTD.1 = 0`)
- Clapeta de respingere (`PORTD.3 = 0`)
- Display-ul nu afișează nicio valoare.

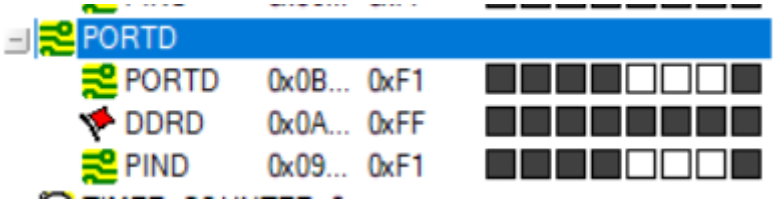


Fig 3.11 Activare clapetă eroare, buzzer, LED rosu

Starea 4 – Reset:

Sistemul revine la valorile inițiale: toți pinii sunt dezactivați, variabilele de validare și eroare sunt resetate, iar sistemul intră din nou în stare de așteptare.

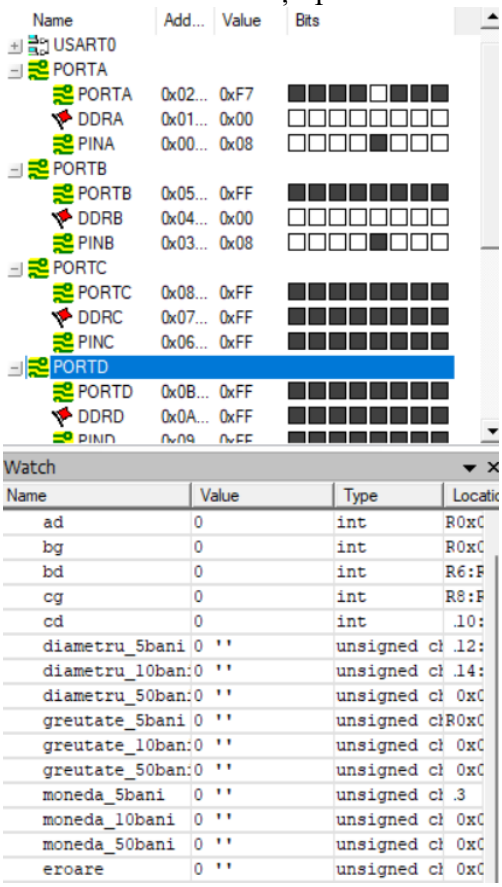


Fig 3.12 Reset complet variabile

Pentru testarea completă a sistemului, au fost verificate și celelalte două cazuri posibile:

- **Moneda de 10 bani** – prin introducerea valorilor:
  - **Greutate**: 4g
  - **Diametru**: 22.5mm
- **Moneda de 50 bani** – prin introducerea valorilor:
  - **Greutate**: 6.1g
  - **Diametru**: 23.75mm

În ambele situații, sistemul a detectat corect caracteristicile specifice fiecărei monede, a activat pinii corespunzători, a incrementat variabilele de validare, a aprins clapeta potrivită și a afișat valoarea corectă pe display-ul cu 7 segmente.

Detalii complete despre stările sistemului, variabilele modificate și semnalele activate pentru aceste cazuri pot fi consultate în **Anexa 3.**

CONCLUZII

Proiectul realizat demonstrează funcționarea corectă a unui sistem automat de sortare a monedelor, bazat pe un automat finit de stare (FSM) și circuite combinaționale. Prin utilizarea senzorilor de greutate și diametru, sistemul este capabil să recunoască și să sorteze corect monedele românești de 5, 10 și 50 de bani.

În urma testelor efectuate:

- Sistemul a reacționat corespunzător în toate stările definite, de la așteptare până la resetare.
- Monedele valide au fost corect identificate și direcționate în compartimentele potrivite, cu afișarea valorii pe display-ul cu 7 segmente.
- În cazul introducerii unei monede invalide, sistemul a semnalat eroarea prin LED roșu și buzzer, fără a activa o clapetă de sortare.

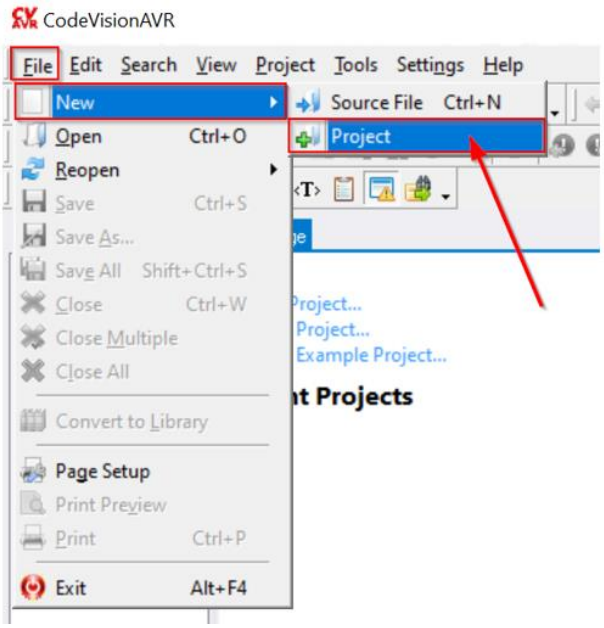
Astfel, implementarea propusă îndeplinește cerințele unui sistem automatizat de sortare, oferind atât feedback vizual, cât și audio, și poate fi extinsă cu ușurință pentru alte tipuri de monede sau validări suplimentare.

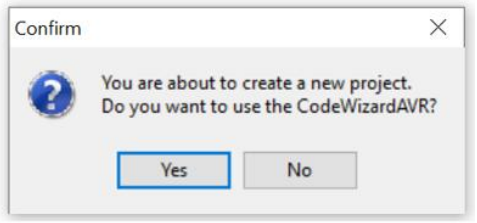
BIBLIOGRAFIE

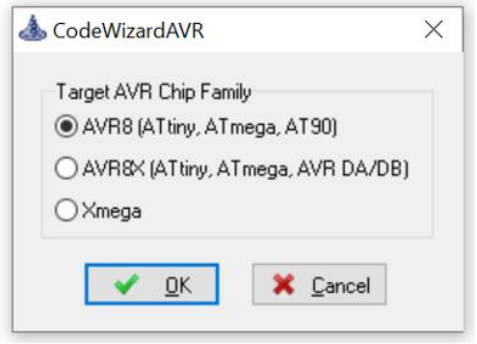
Cinci bani (monedă românească)
Zece bani (monedă românească)
Cincizeci de bani (monedă românească)
7 Segment Arduino Interface
Pagina disciplinei AMP2.MC

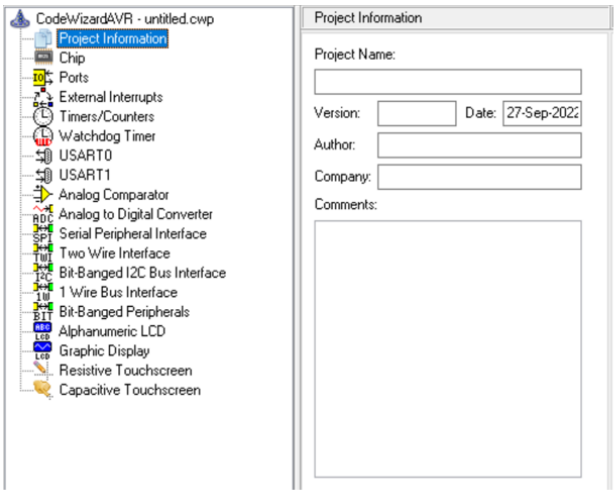
ANEXE

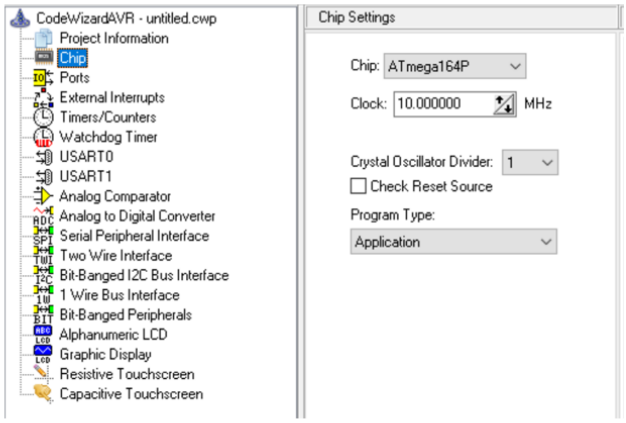
ANEXA 1

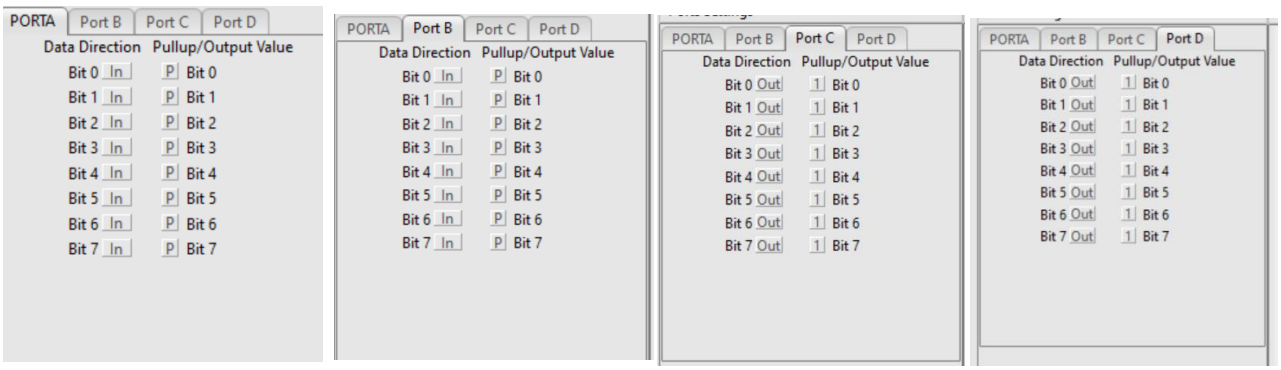


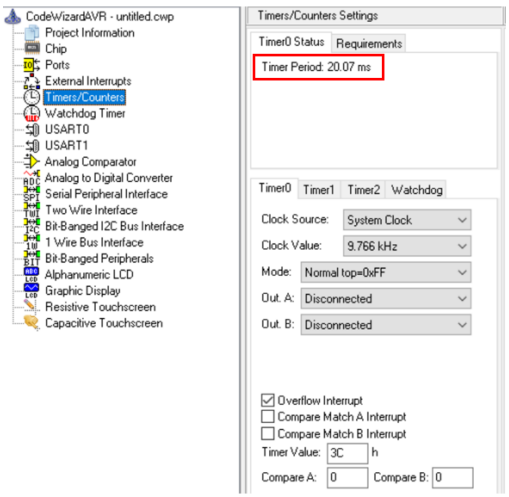


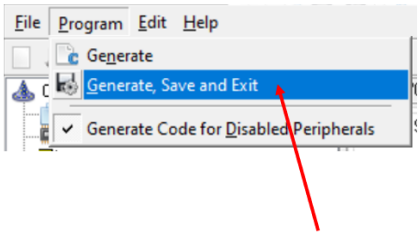









CodeWizardAVR - untitled.cwp
Project Information
Chip
Ports
External Interrupts
Timers/Counters
Watchdog Timer
USART0
USART1
Analog Comparator
Analog to Digital Converter
Serial Peripheral Interface
Two Wire Interface
Bit-Banged I2C Bus Interface
1 Wire Bus Interface
Bit-Banged Peripherals
Alphanumeric LCD
Graphic Display
Resistive Touchscreen
Capacitive Touchscreen
Timers/Counters Settings
Timer0 Status
Requirements
Timer Period: 20.07 ms
Timer0
Timer1
Timer2
Watchdog
Clock Source:
System Clock
Clock Value:
9.766 kHz
Mode:
Normal top=0xFF
Out. A:
Disconnected
Out. B:
Disconnected
Overflow Interrupt
Compare Match A Interrupt
Compare Match B Interrupt
Timer Value:
3C
h
Compare A:
0
Compare B:
0


File
Program
Edit
Help
Generate
Generate, Save and Exit
Generate Code for Disabled Peripherals

ANEXA 2

```
/*****************************************************
This program was created by the CodeWizardAVR V4.03
Automatic Program Generator
© Copyright 1998-2024 Pavel Haiduc, HP InfoTech S.R.L.
http://www.hpinfotech.ro

Project :
Version :
Date    : 20.05.2025
Author  :
Company :
Comments:


Chip type               : ATmega164
Program type            : Application
AVR Core Clock frequency: 10,000000 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 256
*****************************************************/

// I/O Registers definitions
#include <mega164.h>

// Declare your global variables here

unsigned char Q = 0;
int ag,ad,bg,bd,cg,cd;

//presupunem urmatoarele greutati si diametre ale monedelor:
// moneda 5 bani: diametru - 18,2 mm , greutate - 2,81 g
// moneda 10 bani: diametru - 22.5 mm, greutate - 4 g
// moneda 50 bani: diametru - 23.75 mm, greutate - 6,1 g
char greutate_5bani = 0;
char diametru_5bani = 0;
char greutate_10bani = 0;
char diametru_10bani = 0;
char greutate_50bani = 0;
char diametru_50bani = 0;
char moneda_5bani = 0;
char moneda_10bani = 0;
char moneda_50bani = 0;
char eroare = 0;

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Reinitialize Timer 0 value
TCNT0=0x3C;
// Place your code here
```

```
}

void main(void)
{
// Declare your local variables here

float greutate_introdusa, diametru_introdus;

// Clock Oscillator division factor: 1
#pragma optsize-
CLKPR=(1<<CLKPCE);
CLKPR=(0<<CLKPCE) | (0<<CLKPS3) | (0<<CLKPS2) | (0<<CLKPS1) | (0<<CLKPS0);
#ifdef _OPTIMIZE_SIZE_
#pragma optsize+
#endif

// Input/Output Ports initialization
// Port A initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRA=(0<<DDA7) | (0<<DDA6) | (0<<DDA5) | (0<<DDA4) | (0<<DDA3) | (0<<DDA2) |
(0<<DDA1) | (0<<DDA0);
// State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTA=(1<<PORTA7) | (1<<PORTA6) | (1<<PORTA5) | (1<<PORTA4) | (1<<PORTA3) |
(1<<PORTA2) | (1<<PORTA1) | (1<<PORTA0);

// Port B initialization
// Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (0<<DDB6) | (0<<DDB5) | (0<<DDB4) | (0<<DDB3) | (0<<DDB2) |
(0<<DDB1) | (0<<DDB0);
// State: Bit7=P Bit6=P Bit5=P Bit4=P Bit3=P Bit2=P Bit1=P Bit0=P
PORTB=(1<<PORTB7) | (1<<PORTB6) | (1<<PORTB5) | (1<<PORTB4) | (1<<PORTB3) |
(1<<PORTB2) | (1<<PORTB1) | (1<<PORTB0);

// Port C initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRC=(1<<DDC7) | (1<<DDC6) | (1<<DDC5) | (1<<DDC4) | (1<<DDC3) | (1<<DDC2) |
(1<<DDC1) | (1<<DDC0);
// State: Bit7=1 Bit6=1 Bit5=1 Bit4=1 Bit3=1 Bit2=1 Bit1=1 Bit0=1
PORTC=(1<<PORTC7) | (1<<PORTC6) | (1<<PORTC5) | (1<<PORTC4) | (1<<PORTC3) |
(1<<PORTC2) | (1<<PORTC1) | (1<<PORTC0);

// Port D initialization
// Function: Bit7=Out Bit6=Out Bit5=Out Bit4=Out Bit3=Out Bit2=Out Bit1=Out Bit0=Out
DDRD=(1<<DDD7) | (1<<DDD6) | (1<<DDD5) | (1<<DDD4) | (1<<DDD3) | (1<<DDD2) |
(1<<DDD1) | (1<<DDD0);
// State: Bit7=1 Bit6=1 Bit5=1 Bit4=1 Bit3=1 Bit2=1 Bit1=1 Bit0=1
PORTD=(1<<PORTD7) | (1<<PORTD6) | (1<<PORTD5) | (1<<PORTD4) | (1<<PORTD3) |
(1<<PORTD2) | (1<<PORTD1) | (1<<PORTD0);

// Timer/Counter 0 initialization
// Clock source: System Clock
```

```
// Clock value: 9,766 kHz
// Mode: Normal top=0xFF
// OC0A output: Disconnected
// OC0B output: Disconnected
// Timer Period: 20,07 ms
TCCR0A=(0<<COM0A1) | (0<<COM0A0) | (0<<COM0B1) | (0<<COM0B0) | (0<<WGM01) |
(0<<WGM00);
TCCR0B=(0<<WGM02) | (1<<CS02) | (0<<CS01) | (1<<CS00);
TCNT0=0x3C;
OCR0A=0x00;
OCR0B=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Disconnected
// OC1B output: Disconnected
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=(0<<COM1A1) | (0<<COM1A0) | (0<<COM1B1) | (0<<COM1B0) | (0<<WGM11) |
(0<<WGM10);
TCCR1B=(0<<ICNC1) | (0<<ICES1) | (0<<WGM13) | (0<<WGM12) | (0<<CS12) | (0<<CS11) |
(0<<CS10);
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2A output: Disconnected
// OC2B output: Disconnected
ASSR=(0<<EXCLK) | (0<<AS2);
TCCR2A=(0<<COM2A1) | (0<<COM2A0) | (0<<COM2B1) | (0<<COM2B0) | (0<<WGM21) |
(0<<WGM20);
TCCR2B=(0<<WGM22) | (0<<CS22) | (0<<CS21) | (0<<CS20);
TCNT2=0x00;
OCR2A=0x00;
OCR2B=0x00;

// Timer/Counter 0 Interrupt(s) initialization
```

```
TIMSK0=(0<<OCIE0B) | (0<<OCIE0A) | (1<<TOIE0);

// Timer/Counter 1 Interrupt(s) initialization
TIMSK1=(0<<ICIE1) | (0<<OCIE1B) | (0<<OCIE1A) | (0<<TOIE1);

// Timer/Counter 2 Interrupt(s) initialization
TIMSK2=(0<<OCIE2B) | (0<<OCIE2A) | (0<<TOIE2);

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// Interrupt on any change on pins PCINT0-7: Off
// Interrupt on any change on pins PCINT8-15: Off
// Interrupt on any change on pins PCINT16-23: Off
// Interrupt on any change on pins PCINT24-31: Off
EICRA=(0<<ISC21) | (0<<ISC20) | (0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
EIMSK=(0<<INT2) | (0<<INT1) | (0<<INT0);
PCICR=(0<<PCIE3) | (0<<PCIE2) | (0<<PCIE1) | (0<<PCIE0);

// USART0 initialization
// USART0 disabled
UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (0<<RXEN0) | (0<<TXEN0) |
(0<<UCSZ02) | (0<<RXB80) | (0<<TXB80);

// USART1 initialization
// USART1 disabled
UCSR1B=(0<<RXCIE1) | (0<<TXCIE1) | (0<<UDRIE1) | (0<<RXEN1) | (0<<TXEN1) |
(0<<UCSZ12) | (0<<RXB81) | (0<<TXB81);

// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) | (0<<ACIC) | (0<<ACIS1) |
(0<<ACIS0);
ADCSRB=(0<<ACME);
// Digital input buffer on AIN0: On
// Digital input buffer on AIN1: On
DIDR1=(0<<AIN0D) | (0<<AIN1D);

// ADC initialization
// ADC disabled
ADCSRA=(0<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) | (0<<ADPS2) |
(0<<ADPS1) | (0<<ADPS0);

// SPI initialization
// SPI disabled
SPCR=(0<<SPIE) | (0<<SPE) | (0<<DORD) | (0<<MSTR) | (0<<CPOL) | (0<<CPHA) | (0<<SPR1)
| (0<<SPR0);
```

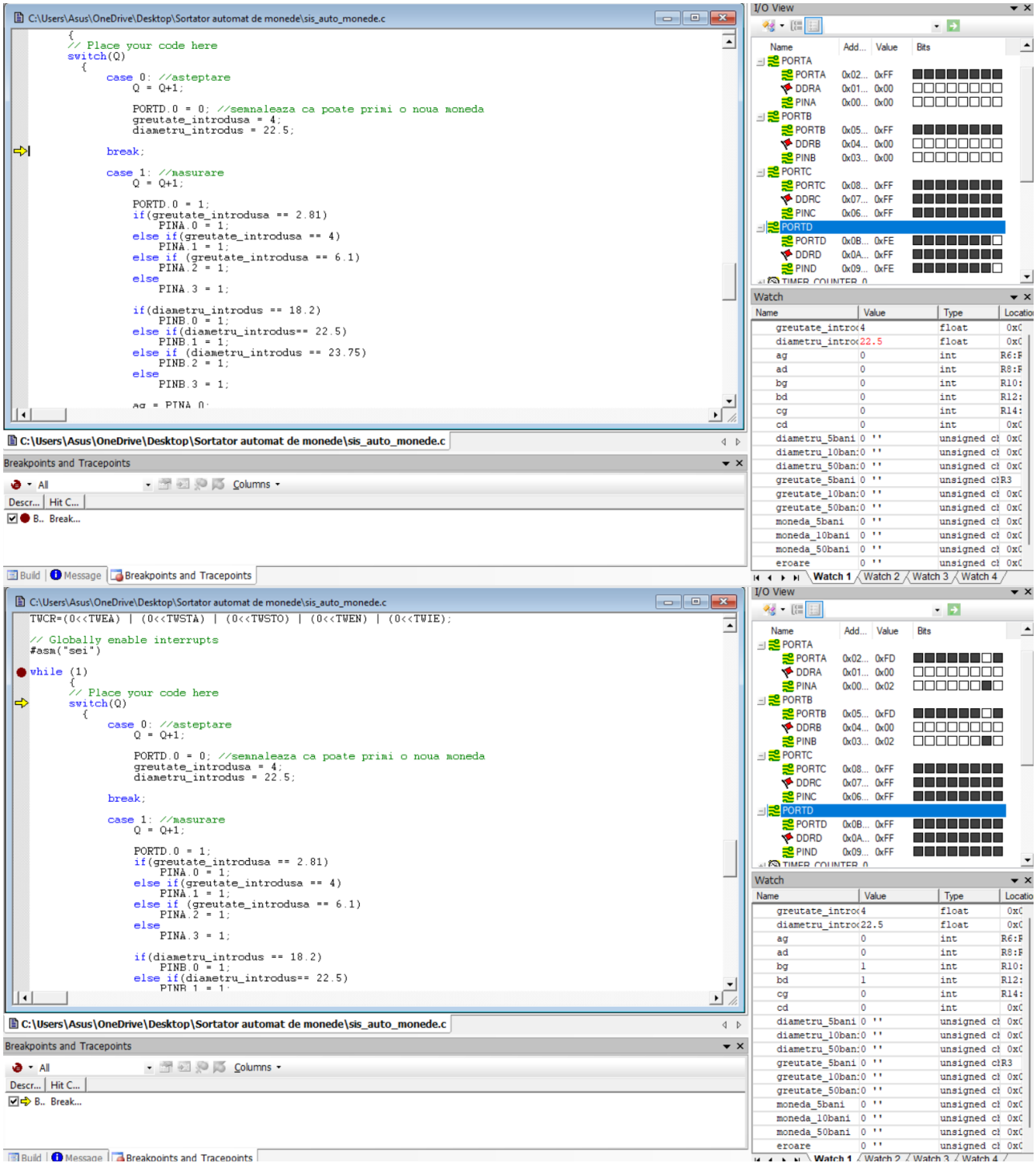
```
// TWI initialization
// TWI disabled
TWCR=(0<<TWEA) | (0<<TWSTA) | (0<<TWSTO) | (0<<TWEN) | (0<<TWIE);

// Globally enable interrupts
#asm("sei")

while (1)
      {
      // Place your code here
      switch(Q)
        {
           case 0: //asteptare
              Q = Q+1;

              PORTD.0 = 0; //semnaleaza ca poate primi o noua moneda
              greutate_introdusa = 2;
              diametru_introdus = 3;

           break;

           case 1: //masurare
              Q = Q+1;

              PORTD.0 = 1;
              if(greutate_introdusa == 2.81)
                 PINA.0 = 1;
              else if(greutate_introdusa == 4)
                 PINA.1 = 1;
              else if (greutate_introdusa == 6.1)
                 PINA.2 = 1;
              else
                 PINA.3 = 1;

              if(diametru_introdus == 18.2)
                 PINB.0 = 1;
              else if(diametru_introdus== 22.5)
                 PINB.1 = 1;
              else if (diametru_introdus == 23.75)
                 PINB.2 = 1;
              else
                 PINB.3 = 1;

              ag = PINA.0;
              ad = PINB.0;

              bg = PINA.1;
              bd = PINB.1;

              cg = PINA.2;
              cd = PINB.2;
```

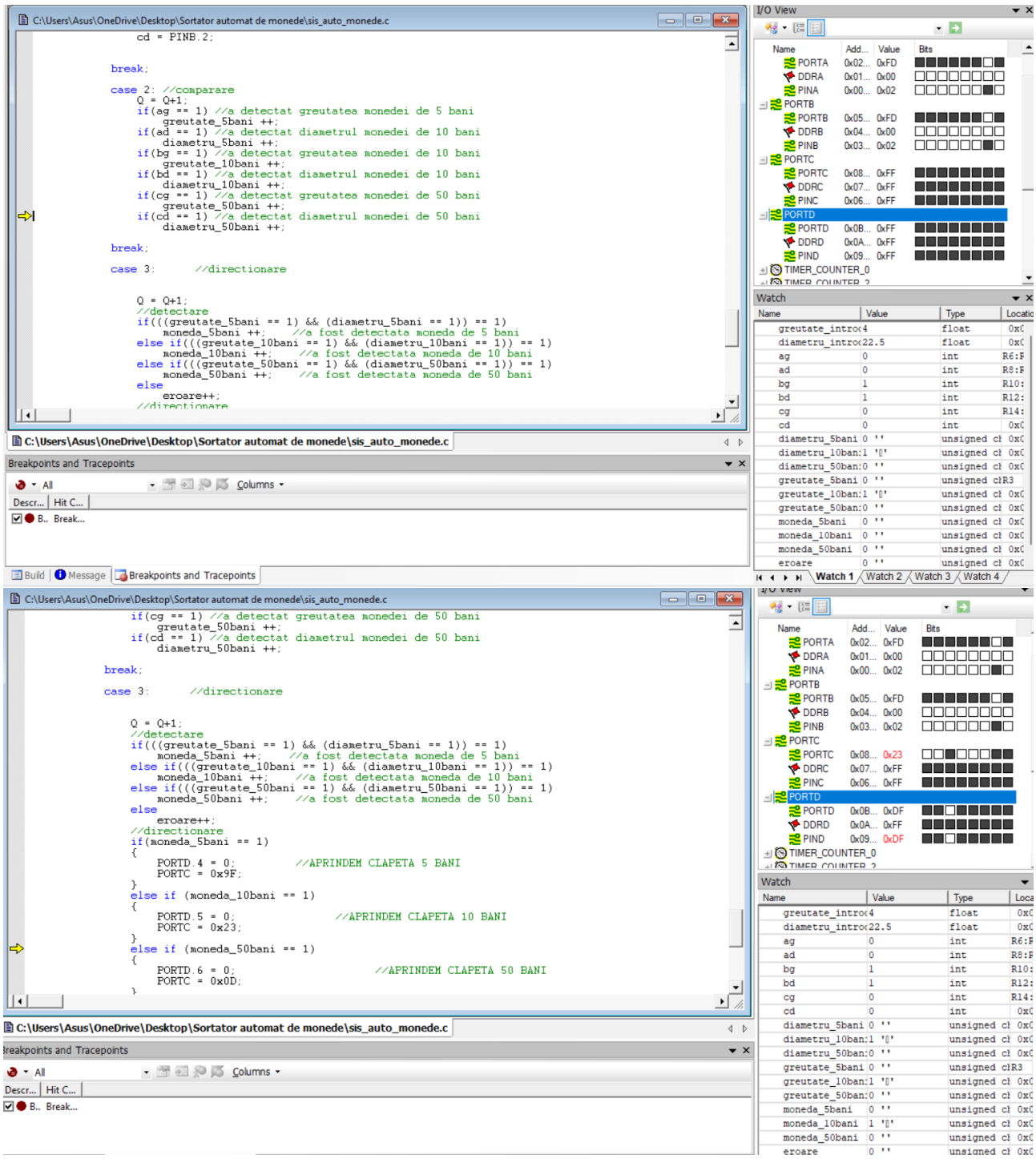
```
break;

case 2: //comparare
   Q = Q+1;
   if(ag == 1) //a detectat greutatea monedei de 5 bani
      greutate_5bani ++;
   if(ad == 1) //a detectat diametrul monedei de 10 bani
      diametru_5bani ++;
   if(bg == 1) //a detectat greutatea monedei de 10 bani
      greutate_10bani ++;
   if(bd == 1) //a detectat diametrul monedei de 10 bani
      diametru_10bani ++;
   if(cg == 1) //a detectat greutatea monedei de 50 bani
      greutate_50bani ++;
   if(cd == 1) //a detectat diametrul monedei de 50 bani
      diametru_50bani ++;

break;

case 3:     //directionare


   Q = Q+1;
   //detectare
   if(((greutate_5bani == 1) && (diametru_5bani == 1)) == 1)
      moneda_5bani ++;    //a fost detectata moneda de 5 bani
   else if(((greutate_10bani == 1) && (diametru_10bani == 1)) == 1)
      moneda_10bani ++;    //a fost detectata moneda de 10 bani
   else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
      moneda_50bani ++;    //a fost detectata moneda de 50 bani
   else
      eroare++;
   //directionare
   if(moneda_5bani == 1)
   {
      PORTD.4 = 0;        //APRINDEM CLAPETA 5 BANI
      PORTC = 0x9F;
   }
   else if (moneda_10bani == 1)
   {
      PORTD.5 = 0;              //APRINDEM CLAPETA 10 BANI
      PORTC = 0x23;
   }
   else if (moneda_50bani == 1)
   {
      PORTD.6 = 0;                  //APRINDEM CLAPETA 50 BANI
      PORTC = 0x0D;
   }
   else
      {
```

```
                PORTD.1 = 0;   //APRINDEM BUZZER
                PORTD.2 = 0;   //APRINDEM LED
                PORTD.3 = 0;   //APRINDEM CLAPETA EROARE
              }
        break;

        case 4: //RESET
            Q = 0;
            PORTA = 0XFF;
            PORTB = 0XFF;
            PORTC = 0XFF;
            PORTD = 0XFF;
            greutate_5bani = 0;
            diametru_5bani = 0;
            greutate_10bani = 0;
            diametru_10bani = 0;
            greutate_50bani = 0;
            diametru_50bani = 0;
            moneda_5bani = 0;
            moneda_10bani = 0;
            moneda_50bani = 0;
            eroare = 0;
        break;
      }

    }
}
```

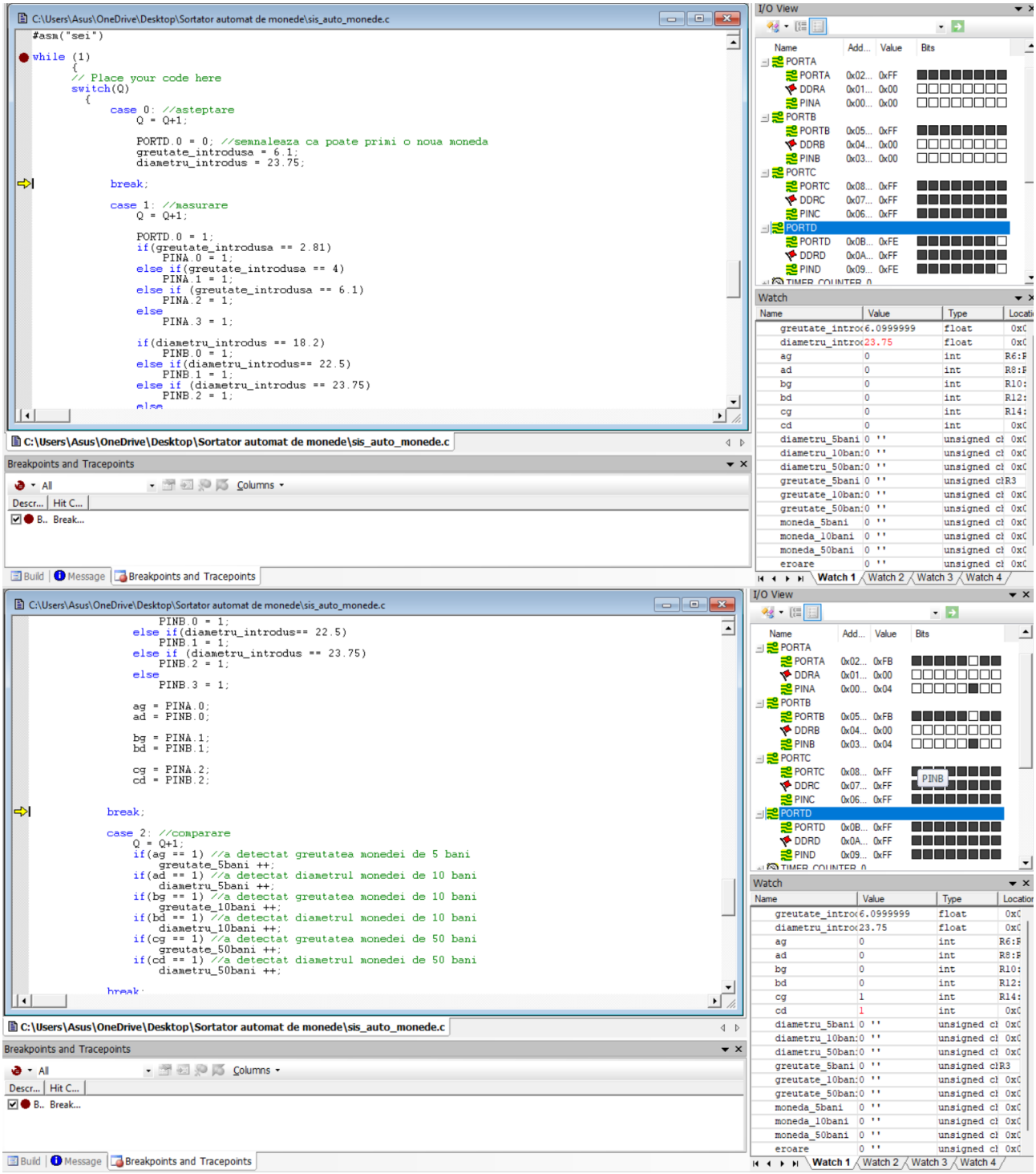
ANEXA 3

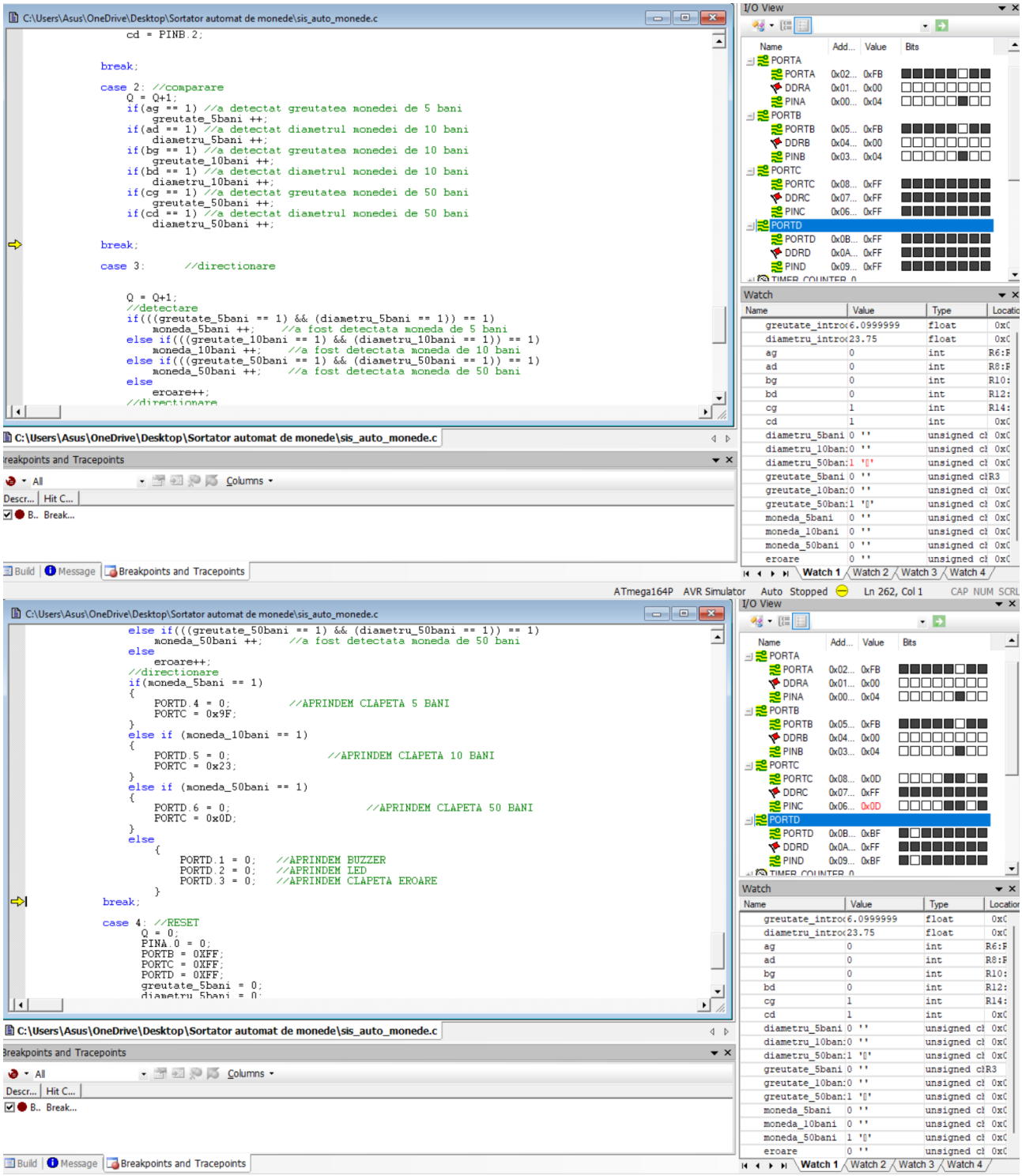
SIMULARE MONEDA 10 BANI




C:\Users\Asus\OneDrive\Desktop\Sortator automat de monede\sis_auto_monede.c
cd = PINB.2;
break;
case 2: //comparare
Q = Q+1;
if(ag == 1) //a detectat greutatea monedei de 5 bani
greutate_5bani ++;
if(ad == 1) //a detectat diametrul monedei de 10 bani
diametru_5bani ++;
if(bg == 1) //a detectat greutatea monedei de 10 bani
greutate_10bani ++;
if(bd == 1) //a detectat diametrul monedei de 10 bani
diametru_10bani ++;
if(cg == 1) //a detectat greutatea monedei de 50 bani
greutate_50bani ++;
if(cd == 1) //a detectat diametrul monedei de 50 bani
diametru_50bani ++;
break;
case 3: //directionare
Q = Q+1;
//detectare
if(((greutate_5bani == 1) && (diametru_5bani == 1)) == 1)
moneda_5bani ++; //a fost detectata moneda de 5 bani
else if(((greutate_10bani == 1) && (diametru_10bani == 1)) == 1)
moneda_10bani ++; //a fost detectata moneda de 10 bani
else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
moneda_50bani ++; //a fost detectata moneda de 50 bani
else
eroare++;
//directionare
C:\Users\Asus\OneDrive\Desktop\Sortator automat de monede\sis_auto_monede.c
Breakpoints and Tracepoints
All
Columns
Descr...
Hit C...
B.. Break...
Build
Message
Breakpoints and Tracepoints
I/O View
Name
Add...
Value
Bits
PORTA 0x02... 0xFD
DDRA 0x01... 0x00
PINA 0x00... 0x02
PORTB
PORTB 0x05... 0xFD
DDRB 0x04... 0x00
PINB 0x03... 0x02
PORTC
PORTC 0x08... 0xFF
DDRC 0x07... 0xFF
PINC 0x06... 0xFF
PORTD
PORTD 0x0B... 0xFF
DDRD 0x0A... 0xFF
PIND 0x09... 0xFF
TIMER_COUNTER_0
Watch
Name
Value
Type
greutate_intro 4 float
diametru_intro 22.5 float
ag 0 int
ad 0 int
bg 1 int
bd 1 int
cg 0 int
cd 0 int
diametru_5bani 0 '' unsigned ch
diametru_10ban 1 '' unsigned ch
diametru_50ban 0 '' unsigned ch
greutate_5bani 0 '' unsigned ch
greutate_10ban 1 '' unsigned ch
greutate_50ban 0 '' unsigned ch
moneda_5bani 0 '' unsigned ch
moneda_10bani 0 '' unsigned ch
moneda_50bani 0 '' unsigned ch
eroare 0 '' unsigned ch
Watch 1
Watch 2
Watch 3
Watch 4
if(cg == 1) //a detectat greutatea monedei de 50 bani
greutate_50bani ++;
if(cd == 1) //a detectat diametrul monedei de 50 bani
diametru_50bani ++;
break;
case 3: //directionare
Q = Q+1;
//detectare
if(((greutate_5bani == 1) && (diametru_5bani == 1)) == 1)
moneda_5bani ++; //a fost detectata moneda de 5 bani
else if(((greutate_10bani == 1) && (diametru_10bani == 1)) == 1)
moneda_10bani ++; //a fost detectata moneda de 10 bani
else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
moneda_50bani ++; //a fost detectata moneda de 50 bani
else
eroare++;
//directionare
if(moneda_5bani == 1)
{
PORTD.4 = 0; //APRINDEM CLAPETA 5 BANI
PORTC = 0x9F;
}
else if (moneda_10bani == 1)
{
PORTD.5 = 0; //APRINDEM CLAPETA 10 BANI
PORTC = 0x23;
}
else if (moneda_50bani == 1)
{
PORTD.6 = 0; //APRINDEM CLAPETA 50 BANI
PORTC = 0x0D;
}
C:\Users\Asus\OneDrive\Desktop\Sortator automat de monede\sis_auto_monede.c
Breakpoints and Tracepoints
All
Columns
Descr...
Hit C...
B.. Break...
I/O View
Name
Add...
Value
Bits
PORTA 0x02... 0xFD
DDRA 0x01... 0x00
PINA 0x00... 0x02
PORTB
PORTB 0x05... 0xFD
DDRB 0x04... 0x00
PINB 0x03... 0x02
PORTC
PORTC 0x08... 0x23
DDRC 0x07... 0xFF
PINC 0x06... 0xFF
PORTD
PORTD 0x0B... 0xDF
DDRD 0x0A... 0xFF
PIND 0x09... 0xDF
TIMER_COUNTER_0
Watch
Name
Value
Type
greutate_intro 4 float
diametru_intro 22.5 float
ag 0 int
ad 0 int
bg 1 int
bd 1 int
cg 0 int
cd 0 int
diametru_5bani 0 '' unsigned ch
diametru_10ban 1 '' unsigned ch
diametru_50ban 0 '' unsigned ch
greutate_5bani 0 '' unsigned ch
greutate_10ban 1 '' unsigned ch
greutate_50ban 0 '' unsigned ch
moneda_5bani 0 '' unsigned ch
moneda_10bani 1 '' unsigned ch
moneda_50bani 0 '' unsigned ch
eroare 0 '' unsigned ch

SIMULARE MONEDA 50 BANI




C:\Users\Asus\OneDrive\Desktop\Sortator automat de monede\sis_auto_monede.c
cd = PINB.2;
break;
case 2: //comparare
Q = Q+1;
if(ag == 1) //a detectat greutatea monedei de 5 bani
greutate_5bani ++;
if(ad == 1) //a detectat diametrul monedei de 10 bani
diametru_5bani ++;
if(bg == 1) //a detectat greutatea monedei de 10 bani
greutate_10bani ++;
if(bd == 1) //a detectat diametrul monedei de 10 bani
diametru_10bani ++;
if(cg == 1) //a detectat greutatea monedei de 50 bani
greutate_50bani ++;
if(cd == 1) //a detectat diametrul monedei de 50 bani
diametru_50bani ++;
break;
case 3: //directionare
Q = Q+1;
//detectare
if(((greutate_5bani == 1) && (diametru_5bani == 1)) == 1)
moneda_5bani ++; //a fost detectata moneda de 5 bani
else if(((greutate_10bani == 1) && (diametru_10bani == 1)) == 1)
moneda_10bani ++; //a fost detectata moneda de 10 bani
else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
moneda_50bani ++; //a fost detectata moneda de 50 bani
else
eroare++;
//directionare
I/O View
Name
Add...
Value
Bits
PORTA
PORTA 0x02... 0xFB
DDRA 0x01... 0x00
PINA 0x00... 0x04
PORTB
PORTB 0x05... 0xFB
DDRB 0x04... 0x00
PINB 0x03... 0x04
PORTC
PORTC 0x08... 0xFF
DDRC 0x07... 0xFF
PINC 0x06... 0xFF
PORTD
PORTD 0x0B... 0xFF
DDRD 0x0A... 0xFF
PIND 0x09... 0xFF
Watch
Name
Value
Type
greutate_intro 6.0999999 float
diametru_intro 23.75 float
ag 0 int
ad 0 int
bg 0 int
bd 0 int
cg 1 int
cd 1 int
diametru_5bani 0 '' unsigned ch
diametru_10ban 0 '' unsigned ch
diametru_50ban 1 '' unsigned ch
greutate_5bani 0 '' unsigned ch
greutate_10ban 0 '' unsigned ch
greutate_50ban 1 '' unsigned ch
moneda_5bani 0 '' unsigned ch
moneda_10bani 0 '' unsigned ch
moneda_50bani 0 '' unsigned ch
eroare 0 '' unsigned ch
Watch 1
Watch 2
Watch 3
Watch 4
Breakpoints and Tracepoints
All
Columns
Descr...
Hit C...
B.. Break...
Build
Message
Breakpoints and Tracepoints
ATmega164P AVR Simulator Auto Stopped Ln 262, Col 1
CAP NUM SCRL
C:\Users\Asus\OneDrive\Desktop\Sortator automat de monede\sis_auto_monede.c
else if(((greutate_50bani == 1) && (diametru_50bani == 1)) == 1)
moneda_50bani ++; //a fost detectata moneda de 50 bani
else
eroare++;
//directionare
if(moneda_5bani == 1)
{
PORTD.4 = 0; //APRINDEM CLAPETA 5 BANI
PORTC = 0x9F;
}
else if (moneda_10bani == 1)
{
PORTD.5 = 0; //APRINDEM CLAPETA 10 BANI
PORTC = 0x23;
}
else if (moneda_50bani == 1)
{
PORTD.6 = 0; //APRINDEM CLAPETA 50 BANI
PORTC = 0x0D;
}
else
{
PORTD.1 = 0; //APRINDEM BUZZER
PORTD.2 = 0; //APRINDEM LED
PORTD.3 = 0; //APRINDEM CLAPETA EROARE
}
break;
case 4: //RESET
Q = 0;
PINA.0 = 0;
PORTB = 0XFF;
PORTC = 0XFF;
PORTD = 0XFF;
greutate_5bani = 0;
diametru_5bani = 0;
I/O View
Watch
Breakpoints and Tracepoints