

Fișe de Lucru Individuale - PLC Siemens S7-1200 și TIA Portal

FIȘA DE LUCRU Nr. 1

Configurarea Hardware și Primul Program în TIA Portal

Echipa: _____ Data: _____

OBIECTIVE:

- ☐ Crearea unui proiect nou în TIA Portal
- ☐ Configurarea hardware-ului S7-1200
- ☐ Realizarea primului program LAD
- ☐ Încărcarea și testarea programului

MATERIALE NECESARE:

- PC cu TIA Portal V17
 - CPU S7-1200 (1214C sau similar)
 - Cablu Ethernet
 - 2 butoane și 2 LED-uri de test
-

PARTEA A: CREAREA PROIECTULUI (15 min)

Pasul 1: Lansarea TIA Portal

1. Deschideți TIA Portal V17
2. Selectați "Create new project"
3. Nume proiect: "Lab_S7_[NumeEchipa]"
4. Clic pe "Create"

Pasul 2: Configurarea CPU

1. "First steps" → "Configure a device"
2. Selectați "SIMATIC S7-1200" → "CPU 1214C DC/DC/DC"
3. Versiunea firmware: V4.0 sau mai nouă

4. Clic pe "OK"

Verificare 1: Verificați că în Project Tree apare CPU 1214C

PARTEA B: CONFIGURAREA HARDWARE (20 min)

Pasul 3: Setarea adresei IP

1. Selectați CPU din rack
2. În "Properties" → "General" → "PROFINET interface"
3. IP address: 192.168.1.1X (X = numărul echipei)
4. Subnet mask: 255.255.255.0

Pasul 4: Adăugarea modulelor de expansiune

1. Din "Hardware catalog" → "Modules" → "Digital input"
2. Trageți "DI 16x24V DC" în Slot 1
3. Din "Digital output" → Trageți "DQ 16x24V DC" în Slot 2
4. Adresele se setează automat: %I0.0-I1.7, %Q0.0-Q1.7

Verificare 2: Rack-ul conține 3 module (CPU + DI + DQ)

PARTEA C: PRIMUL PROGRAM LAD (25 min)

Pasul 5: Deschiderea editorului LAD

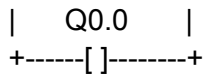
1. Project tree → "Program blocks" → "Main [OB1]"
2. Selectați limbajul "LAD" (Ladder Logic)
3. Apare grila pentru programare

Pasul 6: Programarea logicii START/STOP

Program cerut: Motor cu START (I0.0) și STOP (I0.1)

Network 1: Control motor principal

I0.0	I0.1	M0.0	Q0.0
----[]-----	[/]-----	[]-----	()----



Explicație simboluri:

[] = Contact normal deschis

[/] = Contact normal închis

() = Bobină de ieșire

Pasul 7: Introducerea instrucțiunilor

1. Din "Instructions" → "Basic instructions" → "Bit logic operations"
2. Trageți "Contact NO" pentru I0.0
3. Trageți "Contact NC" pentru I0.1
4. Trageți "Coil" pentru Q0.0
5. Pentru auto-menținere: Contact NO cu Q0.0 în paralel cu I0.0
6. Conectați toate elementele conform schemei

Verificare 3: Programul nu are erori de sintaxă (fără X roșii)

PARTEA D: ÎNCĂRCAREA ȘI TESTAREA (15 min)

Pasul 8: Compilarea programului

1. Clic pe "Compile" (iconița cu ciocanul)
2. Verificați în "Info" că sunt 0 erori
3. Dacă există erori, corectați-le înainte de a continua

Pasul 9: Conectarea la PLC

1. Conectați cablul Ethernet PC-PLC
2. "Online" → "Go online"
3. În fereastra de conexiune: "PG/PC interface" → "PN/IE"
4. "Start search" → selectați PLC-ul găsit
5. "Connect"

Pasul 10: Încărcarea programului

1. "Download to device" → "Software (all)"
2. La întrebări: "Yes" pentru toate
3. Așteptați mesajul "Download completed successfully"

4. "Start all" pentru a rula programul

Verificare 4: LED-ul RUN pe CPU este verde

PARTEA E: TESTAREA FUNCȚIONĂRII (10 min)

Pasul 11: Monitorizarea online

1. În programul LAD: "Monitor all"
 2. Elementele active apar cu fundal galben
 3. Testați funcționarea:
 - Apăsați I0.0 → Q0.0 trebuie să se activeze
 - Eliberați I0.0 → Q0.0 rămâne activ (auto-menținere)
 - Apăsați I0.1 → Q0.0 trebuie să se dezactiveze
-

ÎNTREBĂRI DE AUTOEVALUARE:

1. Care este diferența între contactul NO și NC?

2. De ce este necesară funcția de auto-menținere?

3. Ce semnifică adresa %I0.0 în sistemul S7?

OBSERVAȚII ȘI PROBLEME ÎNTÂLNITE:

Instructor: _____ Punctaj: ____/25

FIȘA DE LUCRU Nr. 2

Aplicația Transportor cu Detectare Piese

Echipa: _____ Data: _____

OBJECTIVE:

- ☐ Implementarea logicii pentru transportor industrial
- ☐ Utilizarea temporizatoarelor TON
- ☐ Programarea contoarelor CTU
- ☐ Testarea sistemului complet

MATERIALE NECESARE:

- Proiectul din Fișa 1 (configurat și funcțional)
- 4 senzori digitali (simulați prin butoane)
- 4 actuatori (simulați prin LED-uri)

PARTEA A: ANALIZA CERINTELOR (10 min)

Sistemul transportor are:

- Senzor detectare piesă: **I0.2**
- Buton START sistem: **I0.3**
- Buton STOP urgență: **I0.7**
- Motor transportor: **Q0.2**
- Cilindru sortare: **Q0.3**
- LED producție (verde): **Q0.4**
- LED alarmă (roșu): **Q0.5**

Logica de functionare:

1. START pornește motorul transportorului
2. Detectarea unei piese activează cilindrul pentru 3 secunde
3. Se contorizează piesele (max 999)
4. Alarmă dacă lipsesc piese mai mult de 20 secunde
5. STOP urgentă oprește totul instant

PARTEA B: PROGRAMAREA SISTEMULUI (40 min)

Network 1: Control motor transportor

Desenati schema LAD:

10.3 10.7 M1.0 Q0.2

--[]-----[/]-----[]----- ()--

Q0.2

+-----[]-----+

Explicație: START cu auto-menținere + STOP urgență

Pasul 1: Implementați Network 1

1. Creați Network 1 în Main [OB1]
2. Introduceți contactele și bobina conform schemei
3. Testați funcționarea START/STOP

Network 2: Detectare piesă și activare cilindru

```

I0.2           M1.1
--[P]------( )--

M1.1    IEC_Timer    Q0.3
--[ ]-----[TON]------( )--
      |
      PT:=T#3s
  
```

Pasul 2: Programarea temporizatorului

1. Creați Network 2
2. Din "Instructions" → "Extended instructions" → "IEC Timers"
3. Trageți blocul "TON" (Timer On Delay)
4. Setați PT := T#3s (3 secunde)
5. Conectați conform schemei

Network 3: Contorizarea pieselor

```

M1.1           IEC_Counter
--[ ]-----[CTU]-----
      |
      PV := 999
      |
      %MW10
      |
      (Valoare contor)
  
```

Pasul 3: Implementarea contorului

1. Din "Instructions" → "IEC Counters" → "CTU"
2. Setăți PV := 999 (valoare maximă)
3. Conectați ieșirea CV la %MW10 pentru vizualizare

Network 4: Semnalizare producție activă

```

I0.2      TOF_Production      Q0.4
--[ ]-----[TOF]------( )--
      |
      PT:=T#20s

```

Pasul 4: LED producție (verde)

1. Folosiți TOF (Timer Off Delay) cu PT := T#20s
2. LED-ul rămâne aprins atâta timp cât se detectează piese
3. Se stinge doar după 20s fără detecție

Network 5: Alarmă lipsă piese

```

I0.2      TON_Alarm           Q0.5
--[ /]-----[TON]------( )--
      |
      PT:=T#20s

```

Verificare: Toate network-urile sunt completate fără erori

PARTEA C: TESTAREA APLICAȚIEI (20 min)

Pasul 5: Compilare și încărcare

1. "Compile" → verificați 0 erori
2. "Download to device" → încărcați programul
3. "Start all" pentru rulare

Pasul 6: Testarea sistematică

Test 1 - START/STOP motor:

- [] I0.3 → Q0.2 = 1 (motor pornit)
- [] I0.7 → Q0.2 = 0 (stop urgență)

Test 2 - Detectare piese:

- [] I0.2 → Q0.3 = 1 pentru 3 secunde
- [] Contorul %MW10 se incrementează

Test 3 - LED producție:

- [] Cu piese detectate → Q0.4 = 1 (verde)
- [] Fără piese 20s → Q0.4 = 0

Test 4 - Alarmă:

- [] Fără piese 20s → Q0.5 = 1 (roșu)

PARTEA D: OPTIMIZĂRI (10 min)

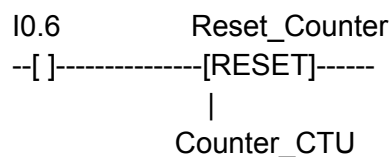
Pasul 7: Adăugați funcții suplimentare

Cerințe opționale:

1. Reset contor prin I0.6
2. Viteză variabilă motor (simulată)
3. Afișarea ratei de producție (piese/min)

Cod pentru reset contor:

Network 6: Reset contor



ÎNTREBĂRI DE AUTOEVALUARE:

1. Care este diferența între TON și TOF?

2. Ce se întâmplă dacă setați PV la contor = 0?

3. De ce folosim [P] pentru detectarea piesei?

REZULTATE TESTARE:

Funcții implementate cu succes:

- ☐ Control motor START/STOP
- ☐ Detectare și sortare piese
- ☐ Contorizare piese
- ☐ Semnalizare producție
- ☐ Alarmă lipsă piese

Probleme întâlnite și soluții:

Instructor: _____ Punctaj: ____/30

FIȘA DE LUCRU Nr. 3

Control Temperatură cu Regulator PID

Echipa: _____ Data: _____

OBIECTIVE:

- ☐ Configurarea intrărilor/ieșirilor analogice
- ☐ Implementarea regulatorului PID
- ☐ Scalarea semnalelor analogice
- ☐ Testarea controlului în buclă închisă

MATERIALE NECESARE:

- Modul AI 4x(0-10V) sau (4-20mA)
- Modul AQ 2x(0-10V)
- Potențiomtru pentru simulare senzor
- LED PWM pentru simulare actuator

PARTEA A: CONFIGURAREA MODULELOR ANALOGICE (15 min)

Pasul 1: Adăugarea modulului AI

1. În "Device configuration", selectați rack-ul
2. Hardware catalog → AI 4x13bit → 6ES7 231-4HF32-0XB0
3. Trageți în Slot 3

4. Adresele automate: %IW0, %IW2, %IW4, %IW6

Pasul 2: Configurarea canalului temperatură

1. Selectați modulul AI în rack
2. Channel 0 → "Properties"
3. Measurement type: "Voltage"
4. Range: "±10 V"
5. Enable "Smoothing" → factor: 4

Pasul 3: Adăugarea modulului AQ

1. Hardware catalog → AQ 2x14bit → 6ES7 232-4HB32-0XB0
2. Trageți în Slot 4
3. Adresele automate: %QW0, %QW2
4. Channel 0 → Range: "0 to 10 V"

Verificare: Configurația hardware este completată

PARTEA B: PROGRAMAREA SCALĂRII (20 min)

Network 1: Scalarea intrării de temperatură

Conversie: 0-10V → 0-400°C

Pasul 4: Implementarea scalării

1. Creați un nou Function Block: "FB_Scalare"
2. Din "Instructions" → "Move operations" → "SCALE_X"
3. Parametrii:
 - IN := %IW0 (intrarea analogică)
 - MIN := 0.0
 - MAX := 400.0
 - OUT := "Temperature_Current" (REAL)

Cod complet Network 1:

```
%IW0      SCALE_X      "Temperature_Current"
-----[SCALE_X]-----
```

|
MIN: 0.0
MAX: 400.0

Network 2: Setarea valorii dorite

"Setpoint_Manual" "Temperature_Setpoint"
-----[MOVE]-----
|
Value: 150.0°C

PARTEA C: IMPLEMENTAREA REGULATORULUI PID (25 min)

Pasul 5: Adăugarea blocului PID

1. Din "Instructions" → "Technology" → "PID control"
2. Trageți "PID_Compact"
3. Va crea automat un DB de instanță

Network 3: Blocul regulator PID

```
"PID_Temperature"  
[PID_Compact  
  Input := "Temperature_Current"  
  Setpoint := "Temperature_Setpoint"  
  Output := "Heater_Output"  
  Mode := INT#1  
  Config.Kp := 2.5  
  Config.Ti := T#45s  
  Config.Td := T#10s  
  Config.Sample_Time := T#1s  
]
```

Pasul 6: Configurarea parametrilor PID

Parametrii recomandați pentru control temperatură:

- Kp (Proportional): 2.5
- Ti (Integral): 45 secunde
- Td (Derivativ): 10 secunde
- Sample_Time: 1 secundă

Network 4: Scalarea ieșirii

"Heater_Output" SCALE_X %QW0
-----[SCALE_X]-----
|
MIN: 0
MAX: 27648

Verificare: Regulatorul PID este configurat complet

PARTEA D: TESTAREA SISTEMULUI (15 min)

Pasul 7: Încărcarea și pornirea

1. Compilați proiectul (0 erori)
2. Descărcați în PLC
3. Porniți programul

Pasul 8: Testarea manuală

Test setup:

1. Conectați potențiometrul la %IW0
2. Conectați LED-ul PWM la %QW0
3. Setati temperatura dorită la 150°C
4. Variați potențiometrul și observați răspunsul

Pasul 9: Monitorizarea parametrilor

Creați un Watch Table cu:

- "Temperature_Current" (REAL)
 - "Temperature_Setpoint" (REAL)
 - "Heater_Output" (REAL)
 - "PID_Temperature".Error (BOOL)
 - %IW0 (valoare brută intrare)
 - %QW0 (valoare brută ieșire)
-

PARTEA E: OPTIMIZAREA PERFORMANȚELOR (10 min)

Pasul 10: Ajustarea parametrilor

Testați cu diferite setări:

Test 1 - Răspuns rapid:

- Kp: 5.0, Ti: 30s, Td: 15s
- Observați overshoot-ul

Test 2 - Răspuns stabil:

- Kp: 1.5, Ti: 60s, Td: 5s
- Observați stabilitatea

Test 3 - Compromis:

- Kp: 2.5, Ti: 45s, Td: 10s
- Evaluați performanța generală

Criterii de evaluare răspuns:

- ☐ Timp de stabilizare < 30 secunde
 - ☐ Overshoot < 10% din setpoint
 - ☐ Eroare de regim permanent < 2°C
 - ☐ Fără oscilații în regim permanent
-

ÎNTREBĂRI DE AUTOEVALUARE:

1. Ce reprezintă parametrii Kp, Ti, Td în controlul PID?

2. De ce este necesară scalarea semnalelor analogice?

3. Cum influențează Sample_Time performanța regulatorului?

REZULTATE TESTARE:

Parametrii PID optimali găsiți:

- Kp: _____ Ti: _____ Td: _____

Performanțe obținute:

- Timp stabilizare: _____ secunde
- Overshoot maxim: _____ %

- Eroare finală: _____ °C

Instructor: _____ Punctaj: ____/35

FIȘA DE LUCRU Nr. 4

Crearea Interfeței HMI cu WinCC

Echipa: _____ Data: _____

OBIECTIVE:

- ☐ Crearea unui HMI în TIA Portal
- ☐ Designul ecranelor de comandă
- ☐ Conectarea variabilelor PLC-HMI
- ☐ Testarea interfeței complete

MATERIALE NECESARE:

- Proiectul S7 cu aplicația transportor funcțională
 - PC/panel HMI sau WinCC RT Advanced
-

PARTEA A: CONFIGURAREA HMI (20 min)

Pasul 1: Adăugarea dispozitivului HMI

1. Project tree → "Add new device"
2. Selectați "HMI" → "SIMATIC Panels"
3. Alegeți "KTP700 Basic" (7 inch, 800x480)
4. Nume: "HMI_Transportor"
5. "OK"

Pasul 2: Crearea conexiunii HMI-PLC

1. Selectați HMI din Project tree
2. "Properties" → "Connections"
3. "Add new connection"
4. Connection name: "Connection_1"
5. Partner: selectați CPU S7-1200
6. "OK"

Verificare: Conexiunea HMI-PLC este configurată

PARTEA B: DESIGNUL ECRANULUI PRINCIPAL (25 min)

Pasul 3: Deschiderea editorului HMI

1. HMI device → "Screens" → "Root screen"
2. Se deschide editorul grafic WinCC
3. Dimensiunea ecranului: 800x480 pixeli

Pasul 4: Layout-ul ecranului principal

Distribuirea elementelor pe ecran:

[LOGO]	SISTEM TRANSPORTOR v1.0	
STATUS SISTEM: [●] ONLINE		
Motor transportor:	[●] PORNIT	
Piese produse:	[123] bucăți	
Rata producție:	[45] piese/min	
COMENZI		
[START]	[STOP]	[RESET]
ALARME: []		

Pasul 5: Adăugarea elementelor grafice

5.1 - Titlul aplicației:

1. "Elements" → "Basic objects" → "Text"
2. Text: "SISTEM TRANSPORTOR v1.0"
3. Font: Arial, 18pt, Bold
4. Poziție: centrat sus

5.2 - Indicatori status LED:

1. "Elements" → "Basic objects" → "Circle"
2. Creați 2 cercuri (verde și roșu)
3. Proprietăți → "Animations" → "Appearance"
4. Conectați la variabilele PLC:
 - LED verde → "HMI_Connection_1".Q0.4
 - LED roșu → "HMI_Connection_1".Q0.5

5.3 - Afișaj numeric piese:

1. "Elements" → "Basic objects" → "I/O field"
 2. Data type: "UInt"
 3. "Properties" → "General" → "Process"
 4. Variable: "HMI_Connection_1".%MW10
-

PARTEA C: BUTOANELE DE COMANDĂ (20 min)

Pasul 6: Implementarea butoanelor

6.1 - Butonul START:

1. "Elements" → "Basic objects" → "Button"
2. Text: "START"
3. Culoare: Verde (#00FF00)
4. "Properties" → "Events" → "Click"
5. Function: "SetBit"
6. Variable: "HMI_Connection_1".I0.3

6.2 - Butonul STOP:

1. Buton roșu cu textul "STOP"
2. Event: "Click" → "ResetBit"
3. Variable: "HMI_Connection_1".I0.7

6.3 - Butonul RESET:

1. Buton galben cu textul "RESET"
 2. Event: "Click" → "SetBitWhileKeyPressed"
 3. Variable: "HMI_Connection_1".I0.6
-

PARTEA D: ECRANUL DE SETĂRI (15 min)

Pasul 7: Crearea celui de-al doilea ecran

1. "Screens" → clic dreapta → "Add screen"
2. Nume: "Settings_Screen"
3. Template: "Root screen" (pentru consistență)

Pasul 8: Elementele ecranului de setări

Layout ecran setări:

[←] SETĂRI SISTEM	
Timp activare cilindru:	
[3]	secunde
Timp alarmă lipsă piese:	
[20]	secunde
Număr maxim piese contor:	
[999]	bucăți
[SALVEAZĂ SETĂRI]	

Pasul 9: Navigația între ecrane

1. În ecranul principal: "Button" cu text "SETĂRI"
2. Event: "ActivateScreen"
3. Screen name: "Settings_Screen"
4. În ecranul setări: "Button" cu text "ÎNAPOI"
5. Event: "ActivateScreen"
6. Screen name: "Root screen"

PARTEA E: TESTAREA HMI (10 min)

Pasul 10: Simularea HMI

1. Compilați HMI-ul (0 erori)
2. "Online" → "Simulation" → "Start simulation"
3. Se deschide simulatorul HMI pe PC

Pasul 11: Testarea funcționalității

Checklist testare:

- ☐ Butoanele START/STOP funcționează
- ☐ LED-urile de status se actualizează
- ☐ Contorul de piese afișează valoarea corectă
- ☐ Navigația între ecrane funcționează
- ☐ Câmpurile de setări sunt editabile

Pasul 12: Transfer pe panelul real (opțional)

1. Conectați panelul HMI prin Ethernet
2. "Download to device" → selectați HMI panel
3. Transferați aplicația completă

ÎNTREBĂRI DE AUTOEVALUARE:

1. Care este diferența dintre SetBit și SetBitWhileKeyPressed?

2. De ce este importantă conectivitatea HMI-PLC în industrie?

3. Cum pot fi securizate comenzile critice în HMI?

EVALUAREA INTERFEȚEI:

Funcționalități implementate cu succes:

- ☐ Ecran principal complet
- ☐ Indicatori vizuali status
- ☐ Butoane comandă funcționale
- ☐ Ecran setări navigabil
- ☐ Afișaje numerice actualizate

Aspecte de îmbunătățit:

Instructor: _____ Punctaj: ____/25

FIȘA DE LUCRU Nr. 5

Debugging și Diagnosticare Avansată

Echipa: _____ Data: _____

OBIECTIVE:

- [] Utilizarea tool-urilor de debugging TIA Portal
- [] Diagnosticarea erorilor hardware/software
- [] Monitorizarea performanțelor sistemului
- [] Analiza semnalelor în timp real

MATERIALE NECESARE:

- Proiectul complet S7 cu HMI
 - Sistem cu erori intenționate pentru depanare
-

PARTEA A: TOOL-URILE DE DEBUGGING (20 min)

Pasul 1: Watch Tables - Monitorizarea variabilelor

1. Project tree → "Watch and force tables"
2. Clic dreapta → "Insert new watch table"
3. Nume: "Debug_Transportor"

Pasul 2: Configurarea Watch Table

Adăugați următoarele variabile:

Name	Address	Data Type	Monitor Value	Comment
START_Button	%I0.3	Bool		Buton pornire
STOP_Emergency	%I0.7	Bool		Stop urgență
Motor_Running	%Q0.2	Bool		Motor activ
Piece_Detected	%I0.2	Bool		Senzor piesă
Piece_Counter	%MW10	UInt		Contor piese
Cylinder_Active	%Q0.3	Bool		Cilindru sortare

| Production_LED | %Q0.4 | Bool | | LED producție |
| Alarm_LED | %Q0.5 | Bool | | LED alarmă |

Pasul 3: Activarea monitorizării

1. "Monitor all" pentru activarea monitorizării în timp real
2. Observați cum se modifică valorile la schimbarea intrărilor
3. Valorile TRUE apar evidențiate cu galben

Verificare: Watch Table afișează valorile în timp real

PARTEA B: FORȚAREA VARIABILELOR (15 min)

Pasul 4: Forțarea intrărilor pentru testare

ATENȚIE: Folosiți doar pentru testare, niciodată în producție!

1. În Watch Table, clic dreapta pe %I0.2
2. Selectați "Force" → "Force to '1'"
3. Observați activarea cilindrului și contorului
4. "Force" → "Force to '0'" pentru dezactivare

Pasul 5: Scenarii de testare sistematică

Test 1 - Simulare piesă detectată:

1. Forțați %I0.2 = TRUE pentru 1 secundă
2. Verificați: %Q0.3 activ 3 secunde, %MW10 incrementat

Test 2 - Testare alarmă lipsă piese:

1. Mențineți %I0.2 = FALSE timp de 25 secunde
2. Verificați: %Q0.5 = TRUE după 20 secunde

Test 3 - Stop urgență în timpul producției:

1. Forțați %I0.7 = TRUE în timpul funcționării
 2. Verificați: toate ieșirile se dezactivează instant
-

PARTEA C: TRACE & LOGIC ANALYZER (25 min)

Pasul 6: Configurarea Trace

1. "Online & Diagnostics" → "Trace"
2. "Configure trace" → selectați variabilele:
 - %I0.2 (Piece_Detected)
 - %Q0.3 (Cylinder_Active)
 - %MW10 (Piece_Counter)
 - "Temperature_Current" (dacă aveți control temperatură)

Pasul 7: Setarea trigger-ului

1. "Trigger settings":
 - Trigger variable: %I0.2 (Piece_Detected)
 - Trigger condition: "Positive edge"
 - Pre-trigger time: 2 seconds
 - Post-trigger time: 5 seconds

Pasul 8: Colectarea datelor

1. "Start trace"
2. Simulați detectarea unor piese (manual sau forțat)
3. După 10 detecții: "Stop trace"
4. Analizați graficele în "Trace view"

Analize de efectuat:

- Timpul de răspuns între detectare și activare cilindru
 - Consistența duratei de activare cilindru (3s)
 - Sincronizarea între evenimente
-

PARTEA D: DIAGNOSTICELE CPU (15 min)

Pasul 9: Informații sistem

1. "Online & Diagnostics" → "Diagnostics" → "General"
2. Verificați:
 - Operating state: RUN/STOP
 - Cycle time: Current/Min/Max
 - Memory utilization: Work/Load memory
 - I/O status: OK/ERROR pentru toate modulele

Pasul 10: Monitorizarea performanțelor

Parametrii de monitorizat:

- Cycle time curent: _____ ms (normal: 1-10ms)
- Cycle time maxim: _____ ms
- Load memory used: _____ % (max recomandat: 80%)
- Work memory used: _____ % (max recomandat: 90%)

Pasul 11: Diagnosticile modulelor I/O

1. Selectați fiecare modul din rack
 2. "Module information":
 - Status: OK/Maintenance required/Error
 - Supply voltage: în parametri normali
 - Channel status: individual pentru fiecare canal
-

PARTEA E: REZOLVAREA PROBLEMELOR TIPICE (15 min)

Problema 1: CPU în STOP mode

Cauze posibile și soluții:

- [] Program error → verificați diagnosticile
- [] Hardware fault → verificați LED-urile modulelor
- [] Memory overflow → reduceți complexitatea programului
- [] Communication error → verificați conexiunile

Pași rezolvare:

1. "Diagnostics" → citirea codului de eroare
2. Căutarea în "Help" TIA Portal
3. Corectarea problemei identificate
4. "Reset to factory settings" dacă necesar

Problema 2: I/O nu răspunde

Verificări sistematice:

- [] LED-uri status modul: RUN (verde), ERROR (roșu)
- [] Alimentarea 24V DC: tensiune corectă
- [] Cablajul: continuitate și polaritate
- [] Adresarea: coincidență program-hardware
- [] Configurația: parametri corecți în hardware config

Problema 3: Comunicația HMI întreruptă

Diagnostic pas cu pas:

1. Ping test: cmd → ping [IP_HMI]
 2. Verificarea conexiunii în TIA Portal
 3. Resetarea HMI-ului: menține buton power 10s
 4. Re-download aplicației HMI
-

PARTEA F: CREAREA UNUI RAPORT DE DIAGNOSTIC (10 min)

Pasul 12: Generarea raportului

1. "Tools" → "Generate report"
 2. Selectați elementele:
 - ☐ Hardware configuration
 - ☐ Program blocks documentation
 - ☐ Cross-reference list
 - ☐ Watch table content
 - ☐ Diagnostic information
 3. Format: PDF, Save location: Desktop
-

ÎNTREBĂRI DE AUTOEVALUARE:

1. Care sunt diferențele între "Monitor" și "Force"?

 2. De ce este periculos să folosiți "Force" în producție?

 3. Cum interpretați un cycle time de 50ms pentru aplicația noastră?

-

REZULTATELE DIAGNOSTICĂRII:

Performanțele sistemului:

- Cycle time mediu: _____ ms
- Utilizare memorie: _____ %
- Starea modulelor I/O: _____

Probleme identificate și rezolvate:

1. _____

2. _____
3. _____

Recomandări pentru îmbunătățire:

Instructor: _____ **Punctaj:** ____/30

EVALUAREA FINALĂ - CHECKLIST COMPETENȚE

Pentru obținerea punctajului maxim, studenții trebuie să demonstreze:

Competențe tehnice de bază (60%):

- ☐ Configurarea hardware S7-1200 în TIA Portal
- ☐ Programarea în LAD cu minimum 5 network-uri
- ☐ Implementarea temporizatoarelor și contoarelor
- ☐ Utilizarea semnalelor analogice cu scalare
- ☐ Crearea unei interfețe HMI funcționale

Competențe avansate (25%):

- ☐ Regulatorul PID implementat corect
- ☐ Debugging sistematic cu tool-uri TIA Portal
- ☐ Diagnosticarea și rezolvarea erorilor
- ☐ Optimizarea performanțelor sistemului

Competențe transversale (15%):

- ☐ Documentarea clară a codului
- ☐ Lucrul metodic și organizat
- ☐ Respectarea normelor de siguranță
- ☐ Colaborarea eficientă în echipă

PUNCTAJUL TOTAL MAXIM: 165 puncte **NOTA FINALĂ = (Punctaj obținut / 165) × 9 + 1**