

Sistem de pozitionare pe 3 axe folosind protocolul ETHERNET

Student: Mihaela RĂDĂUCEANU, TSTC anul IV
Indrumator: asist. Silviu EPURE

Abstract — Lucrarea de fata prezinta un sistem de comanda pentru o masina de pozitionare pe 3 axe. Aplicatia realizata este o interfata hard si soft intre un calculator PC si circuitul de comanda al motoarelor folosite la pozitionare. Partea hardware este reprezentata de un microcontroller cu interfata „ethernet”, iar partea software contine aplicatia ce ruleaza pe PC, si algoritmul implementat in microcontroller

Termeni folositi— ethernet, TCP/IP, microcontroller, CAD, motor pas-cu pas, TCP, g-code

I. INTRODUCERE

In industrie se folosesc tot mai multe masini automate (roboti) ce au ca scop realizarea repetata, precisa si fiabila a unei actiuni. Programarea acestor roboti se face in functie de configuratia lor hardware, si de activitatea pe care trebuie s-o intreprinda.

Un sistem de pozitionare pe 3 axe se apropie de notiunea de „robot” daca consideram actiunea ce trebuie realizata ca fiind „pozitionarea unui obiect (freza, senzor, sursa de lumina etc.) intr-un sistem de coordonate cu 3 axe”. Parametrii sistemului si modul in care acesta raspunde comenzilor reprezinta „programul” robotului.

Traectoria pe care o urmareste obiectul pozitionat este unica pentru fiecare aplicatie realizata, si este receptionata de robot ori: linie cu linie, la momentul executiei, ori ca fisier cu coordonatele si actiunile ce trebuie executate.

Realizarea unui obiect finit incepe cu proiectarea lui folosind o aplicatie CAD (computer aided design). Se obtine astfel un obiect virtual cu proprietatile dorite. Daca acest obiect se poate realiza fizic prin prelucrare mecanica, folosind un robot asemenea celui descris mai sus, pornind de la modelul virtual se genereaza traectoria pe care trebuie s-o urmeze freza aschietoare pentru a obtine obiectul fizic.

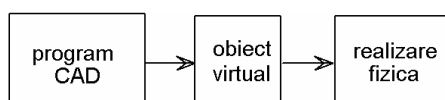


Figura 1. Pasii de la idee la obiect fizic

Pentru scrierea fisierului cu traectoria dorita, se foloseste limbajul tehnic „G-CODE”.

Pana in acest punct, s-a folosit calculatorul. Pentru trimiterea comenzilor spre robot este nevoie de un circuit de interfata care sa lege fizic calculatorul de robot, si o aplicatie software care sa trimita comenzile.

Se poate folosi interfata paralela, seriala, USB, sau ethernet. Se poate scrie relativ usor o aplicatie care sa foloseasca portul paralel sau serial al PC-ului pentru comunicare. Practic inasa, calculatoarele de tip „laptop” sunt tot mai utilizate – iar acestea au renuntat la portul paralel si serial. In cel mai bun caz, se poate folosi un convertor USB/serial.

Raman ca interfete viabile: USB si ethernet. Protocolul USB ofera viteza mare de comunicare, inasa:

- este complex si dificil de folosit;
- limiteaza distanta intre PC si robot la cativa metri;
- nu asigura izolarea galvanica intre robot si PC (robotul este actionat de obicei cu motoare de curent continuu sau alternativ de tensiune mare, iar patrunderea acestei tensiuni in PC prin portul USB duce in cel mai bun caz doar la distrugerea acelu port USB.)

Interfata ethernet este si ea una complexa, inasa asigura o izolare galvanica implicita intre PC si robot. De asemenea, distanta intre cele doua componente nu este limitata sub nici o forma.

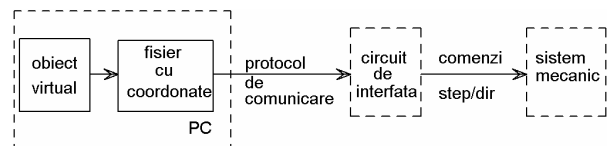


Figura 2. Structura bloc a sistemului

Aplicatia ce ruleaza pe PC a fost scrisa in limbajul Java iar circuitul de interfata a fost realizat cu un microcontroller de tip 18f67J60 cu interfata ethernet.

II. PREZENTAREA SISTEMULUI DE POZITIONARE FOLOSIT

In acest proiect am urmarit comandarea unui sistem mecanic de pozitionare, realizat cu 3 motoare pas-cu pas.

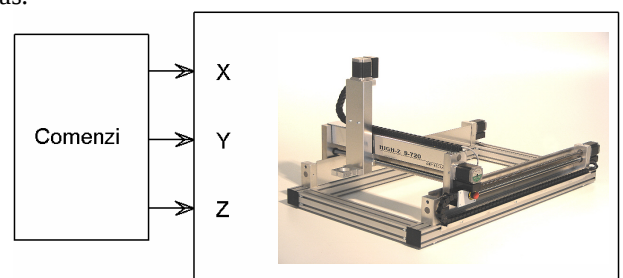


Figura 3. Sistemul mecanic de pozitionare

Fiecare motor este actionat de cate un circuit electronic (driver) cu doua intrari: „STEP” si „DIR”, TTL compatibile. Astfel, la primirea unui impuls pe intrarea STEP, driverul va roti motorul corespunzator cu un pas, la stanga sau la dreapta, in functie de semnalul aplicat la intrarea DIR.

O caracteristica importanta a partii mecanice o reprezinta legatura intre 1 pas al motorului, si distanta parcursa de obiect $\approx 3\mu\text{m}$.

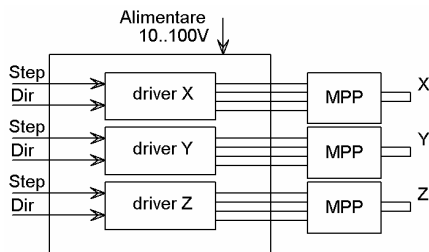


Figura 4. Circuitul de comanda a motoarelor

III. PREZENTAREA PARTII SOFTWARE

Aceasta parte este formata din:

- ansamblul programelor CAD folosite la proiectarea obiectului virtual si la generarea fisierului g-code,
- aplicatia software ce trimite comenzile din fisierul g-code catre
- algoritmul implementat in circuitul de interfata.

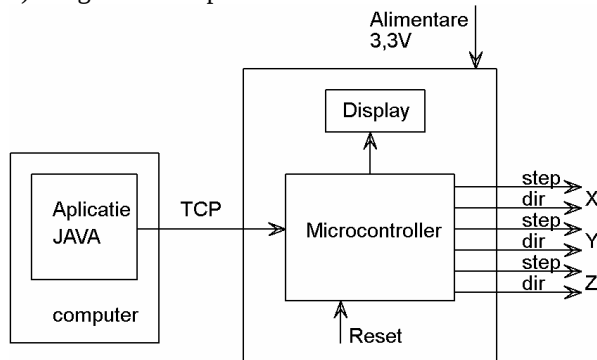


Figura 5. Structura sistemului propus spre realizare

Aplicatia JAVA are scopul de a prelua fisierul g-code si de a-l trimite linie cu linie microcontroller-ului prin TCP. Este realizata ca applet si ruleaza in interiorul unei pagini html. Functii de pozitionare manuala a sistemului vor fi implementate ulterior, in interfata grafica a appletului.

Aplicatiile Java au un avantaj important: sunt „platform independent”, adica pot rula pe majoritatea sistemelor de operare (windows, linux).

TCP (Transmission Control Protocol) este un protocol orientat pe conexiune (spre deosebire de UDP -User Datagram Protocol- care este fara conexiune) ce permite transferul sigur si fiabil al datelor intre procesele aplicatiilor ce ruleaza pe calculatoare conectate in retea. Asigura considerabil mai multe facilitati pentru aplicatii decat UDP, mai ales recuperarea erorilor, controlul transmisiei si siguranta.

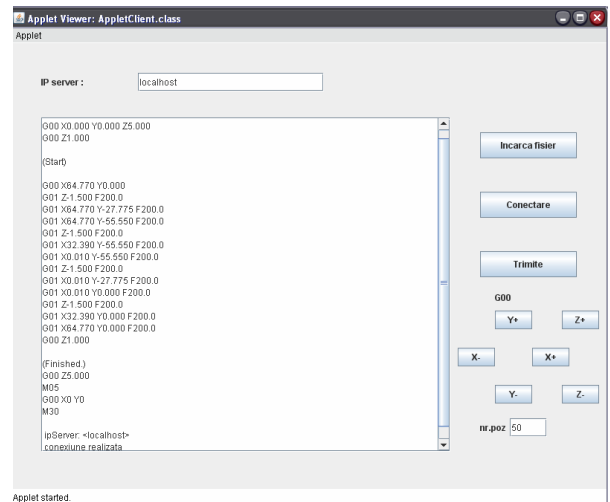


Figura 6. Aspectul interfeței Java

Circuitul de interfata, realizat cu microcontroller este un server http ce proceseaza doar comenzile TCP primite pe un anumit port. Aplicatia ce ruleaza pe calculator este „clientul TCP”.

Pentru scrierea algoritmului ce ruleaza in microcontroller s-a folosit un limbaj de nivel inalt – mikropascal (pascal pentru microcontroller).

Compilatorul ofera facilitati importante in ceea ce priveste implementarea protocolului TCP (sau UDP) in microcontrollerele de tip 18FxxJ60. Exista functii predefinite pentru initializarea microcontrollerului, pentru setarea adresei MAC si a IP-ului etc.

Microcontrollerul receptioneaza comanda (linia din fisierul g-code) trimisa de PC, recunoaste functia ceruta (G00, G01, increment si decrement) si o executa. De exemplu, din sirul de caractere receptionat, prezentat in figura urmatoare, se pastreaza doar „G00 X200Y150Z120”, si reprezinta: „G00=pozitionare rapida, fara traiectorie impusa” pana la coordonatele „X=200, Y=150, Z=120”



Figura 7. Circuitul de interfata – afisand sirul de caractere receptionat prin TCP

S-a trimis linia de comanda G00 X200 Y150 Z12.

Display-ul este unul de tip grafic, folosit curent in telefoanele mobile. A fost folosit pentru afisarea diferitelor variabile din algoritm, care altfel ar fi fost imposibil de urmarit. Legatura intre microcontroller si display este unidirectionala, implementata prin protocolul SPI.

IV. CONCLUZII

S-a realizat si testat un circuit electronic, ce conecteaza prin ethernet un PC de sistemul de pozitionare pe 3 axe.

S-a dezvoltat o aplicatie in limbajul Java, care citeste un fisier de comenzi g-code si il trimite linie cu linie serverului.

S-a scris un program pentru microcontroller, ce recunoaste si implementeaza functiile g-code G00 (pozitionare rapida) si G01 (pozitionare prin interpolare liniara a axelor).

Algoritmii pot fi extinsi, pentru implementarea intregului set de functii g-code, implementarea unor comenzi auxiliare, salvarea parametrilor in memoria EPROM a microcontroller-ului (precum pozitia curenta, mm/pas, viteza maxima de pozitionare etc.).

V. BIBLIOGRAFIE

<http://ro.wikipedia.org/wiki/TCP/IP>

<http://en.wikipedia.org/wiki/Ethernet>

<http://ww1.microchip.com/downloads/en/DeviceDoc/39762d.pdf>