**A&QT-R 2004 (THETA 14)**
**2004 IEEE-TTTC - International Conference on Automation, Quality and Testing, Robotics**
**May 13 - 15, 2004, Cluj-Napoca, Romania**

# Robotic Arm Drive

**Radu Mihai Câmpeanu Ioan Stoian[*] Sebastian Covaci Gheorghe Ratcu**

*"Transilvania" University of Brasov, Electrical Engineering and Computer Science Faculty,*
*Automation Department, Str. Mihai Viteazul nr. 5, RO-500174 Brasov,*
*Tel/Fax ++4268-418836, e-mail: rcampeanu@marconi.unitbv.ro*

**IPA-Center for Technology Transfer, str. Republicii Nr. 109, 3400 Cluj-Napoca*

**Abstract:** The paper presents a complete, low speed, open loop, drive system for a robotic arm with stepping motors as actuators, under software control on a Personal Computer (PC), used for educational purposes. Firstly the hardware is presented. It is composed of an Optical Interface Unit and a Stepping Motors (SM) Power Unit. Secondly the software, realised in Visual Basic, is presented.

**Key words:** Robotic Arm; Stepping Motor; Visual Basic.

## 1. SYSTEM PRESENTATION

The system (Fig 1) is composed of an experimental robotic arm, using as actuators SM, a Power Unit for SM drive, a Personal Computer to control the arm, an Optical Interface Unit which separates the PC from the SM Power Unit and three DC voltage sources.

The robotic arm is a three dimension with aprehensor type using four SM.

The SM belong to one of the most common type, with four phases.

As a less expensive solution [2], it is used the parallel port of the PC. It provides 5 input bits and 12 output bits (8 for data, 4 for control).

For DC supply three different DC sources are used. First of them it has the same voltage reference as the PC and supply DC voltage for the part of the optical interface which is connected to the PC. The two others have the reference voltage the same with the motors, the motors' drive unit and the part of the optical interface which is connected to this unit. In this manner the isolation between the PC and the rest of the drive system is complete. The principal role is PC protection.

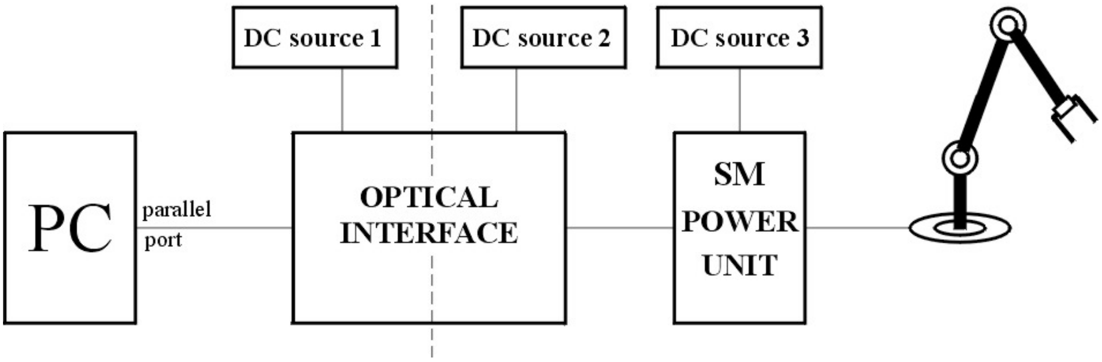


Fig. 1. Robotic arm drive system

## 2. OPTICAL INTERFACE

The electronic scheme contains 17 separator modules where 12 modules are for input and 5 modules for output. This design permit a later development of a close loop control of the robotic arm.

A module has two parts:

- LED command, consisting of a NPN transistor (T3)

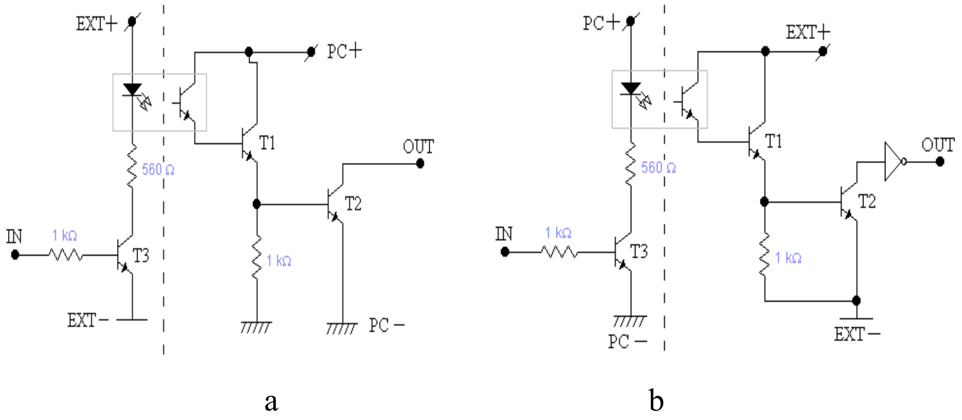
- signal amplification delivered by a phototransistor, made of two transistors and a logical gate (NOT).

NPN transistor (T1) receives the signal from the phototransistor and sends it amplified to the second NPN transistor's (T2) base which is initially maintained turned off by the resistor from base.

T2 puts to ground the input of the NOT gate and this puts the output in "1". The NOT gate's scope is to provide a logical signal to the optical interface output identical with the one from the output of the parallel port.

The Input module doesn't use NOT gates because the parallel port inputs are active in "0".

Schematic for the the two types of modules are presented in Fig 2.



a b

Fig. 2. The input module (a) and the output module (b)

## 3. SM POWER UNIT

SM command program is generated by computer in this case, and delivered via optical interface to the SM Power Unit which commands the motor phases correctly.

SM Power Unit (Fig. 3) has the following parts:

- four command modules for every phase
- four groups of modules for every motor
- two power supplies: 12V for SM and 5V for NAND gates.

For A, B, C, D phases selecting logical "1" at 0, 1 ,2, 3 inputs is used. For motors selecting logical "1" at 4, 5, 6, 7 inputs is used. Each command pin is connected to the correspondent port pin (D0-:-D7).

The amplifier on every phase (Fig. 4) has a logical gate NAND with 2 inputs, two transistors and a protection diode.

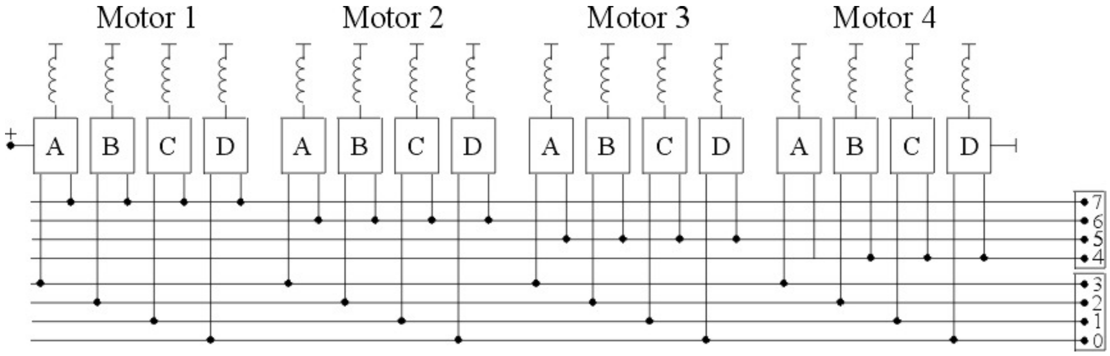


Fig. 3. SM power unit

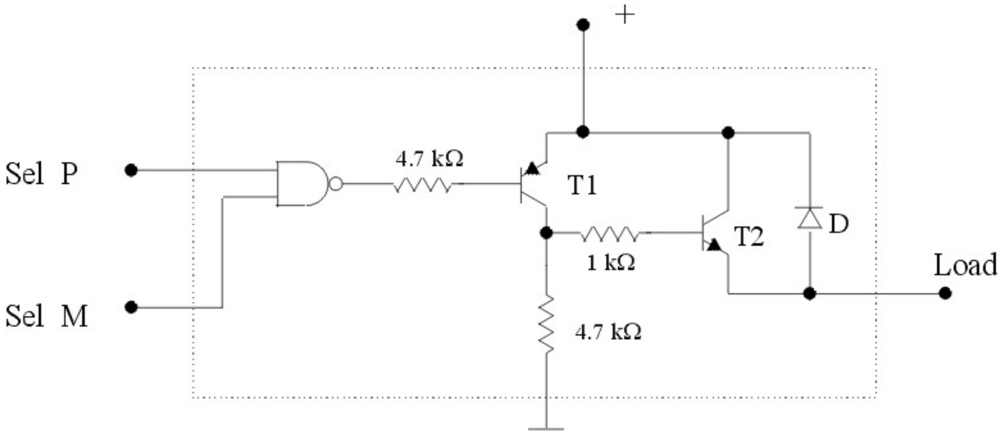


Fig. 4. Motor phase amplifier

SM are commanded in a mixed sequence, the stepping angle being divided in two. Half-step angle is made by combining the two types of commands, the simple and the double. The command table and the time diagrams for SM phase voltages are presented in Table 1 and Fig. 4.

Table 1

| Forward | | | | Reverse | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

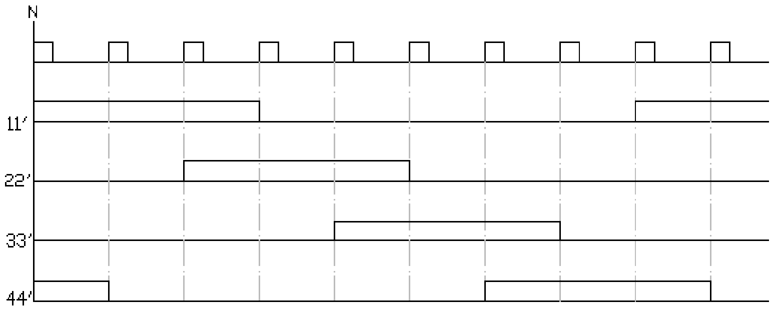


Fig. 5. SM phase voltages

To change the spinning sense of the rotor a reverse sequence is used.

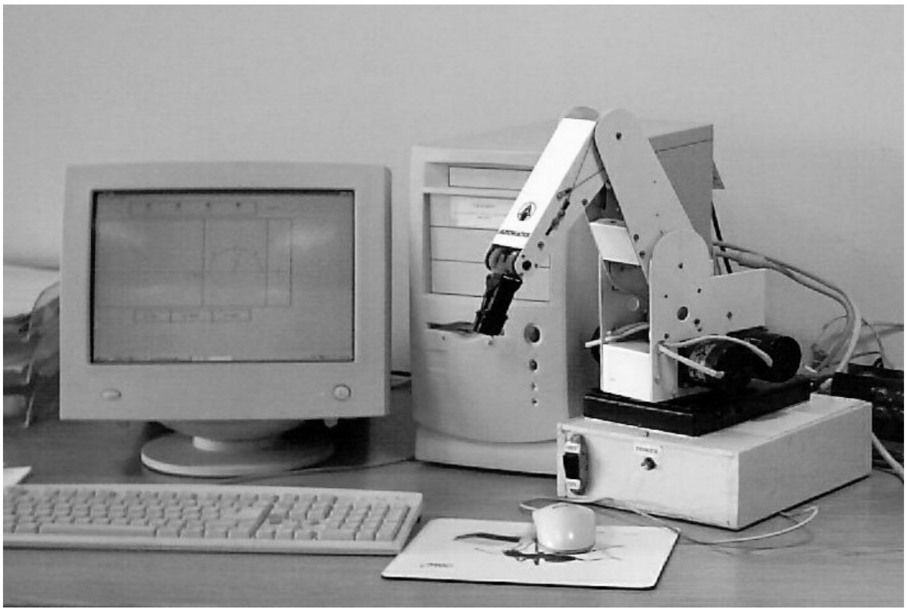


Fig. 6. The robotic arm drive system

The entire robotic arm drive is presented in Fig. 6.

## 4. COMMAND PROGRAM

The command program was made in VisualBasic 6.0

It uses the ActiveX control “parallel.ocx” made in VisualC 6.0 which allow the command of parallel port in VisualBasic and SAPI 5.0 for voice command.

```
               Command routine - left sense for stepping motor:
Private sub StepLeft()
 Call pc.out(addr, 0)
  For i = 0 To 3              'Hardware stepping pulse (mixed)
    tmp = 2 ^ (a(i) - 1)
      Call pc.out(addr, Motor + tmp)
       Sleep (DelTime)
        If i = 3 Then
          tmp = 2 ^ (a(i) - 1) + 2 ^ (a(0) - 1)
         Else
          tmp = 2 ^ (a(i) - 1) + 2 ^ (a(i + 1) - 1)
        End If
      Call pc.out(addr, Motor + tmp)
   Sleep (DelTime)
 Next i
Call pc.out(addr, 0)
End Sub
```

Command routines generate a command sequence of SM phases. The logical schema of mixed command of SM was presented earlier.

In routines there where used the commands:

*Call pc.out(addr,n) ,* this sets the port bits in *n* (0 -:- 255) , where: *pc* is the ActiveX control, *addr* is the address of parallel port.

*Sleep (DelTime)* , generates a delay with *DelTime* milliseconds necessary in angular speed adjustment of SM.

The a() vector contains the real order of phases: a(0) = 3 a(1) = 1 a(2) = 4 a(3) = 2

The initial and the final state of port is set in logical “0”.

The graphic interface of the command program is presented in Fig. 7.

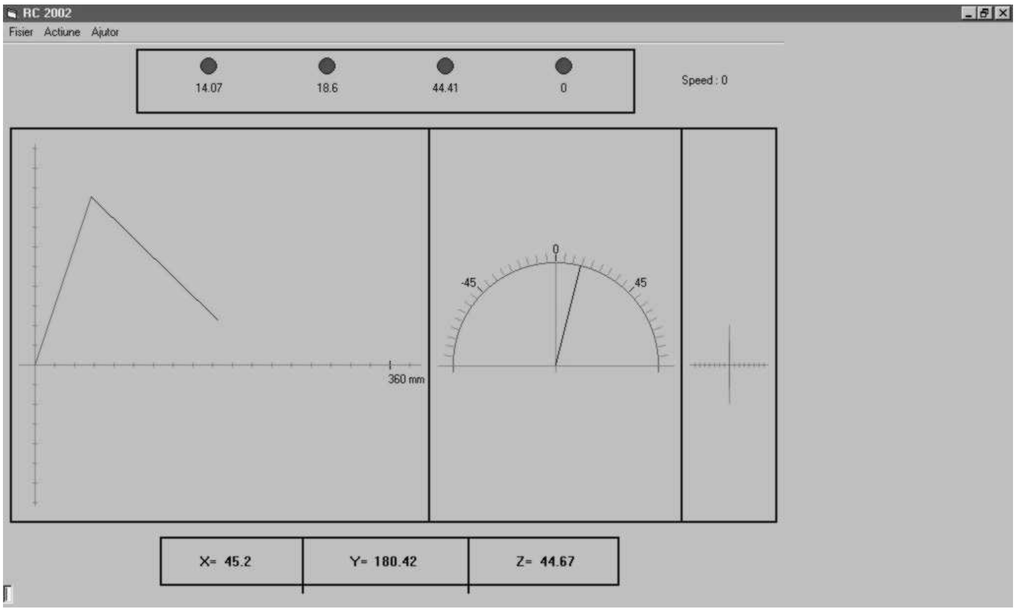


Fig. 7. The graphic interface

## 5. CONCLUSIONS

The hardware and the software of an inexpensive PD drive system for a robot arm using stepping motors as actuators have been presented.

The drive system has proved reliable and easy to use in practice. The low cost is an important advantage.

## REFERENCES

[1] Acarnley P. P., (1982): Stepping Motors: a guide to modern theory and practice. *IEE Control Engineering*, series 19.
[2] Dunlop G. R., (1992): Simple Step Motor Control for Multiaxe Positioning. In: *Proceedings of the 6th Conference on Power Electronic and Motion Control*, vol 2, pp 424-428, Budapest.
[3] Halvorson M., (2000): Visual Basic 6.0 Professional, Microsoft.