AD-A199 925

**Title:** Microcomputer Software and Interface for Control of a Microscope Scanning Stage

**Authors:** Benjamin R. Lee, Helmut Ludl, C. Patrick Reynolds

**Abstract:**

**COSTAR Codes:**
Computer; Image Processing, Computer-Assisted; Software; Microcomputers; Microscopy; User-Computer Interface

**Source of Funding Numbers:**

<table>
<thead>
<tr>
<th>PROGRAM ELEMENT NO.</th>
<th>PROJECT NO.</th>
<th>TASK NO.</th>
<th>WORK UNIT ACCESSION NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>62233</td>
<td>NN33C30</td>
<td>006.1003</td>
<td>DN377031</td>
</tr>
</tbody>
</table>

**DATE OF REPORT:** 1988

**PAGE COUNT:** 7

**Supplementary Notation:**
Analytical and Quantitative Cytology and Histology, Vol. 10, No. 1, Feb. 1988, pp. 47-53

**Distribution/Availability:**
Approved for public release; distribution is unlimited

**Classification:** Unclassified

**Security Classification of This Page:** Unclassified

**DTIC Table:**
Unannounced

**Notes:**

- Distribution/Availability may be used without restriction.
- All other editions are obsolete.
Microcomputer Software and Interface for Control of a Microscope Scanning Stage

Benjamin R. Lee
Helmut Ludl
C. Patrick Reynolds, M.D., Ph.D.

Motorized scanning stages are valuable in microscopy systems that employ digital image analysis and for the development of semiautomatic computer-assisted microscope systems; the development of standard software "tools" to control such stages will facilitate their integration into a variety of computer-based systems. A set of Microsoft BASIC and Turbo PASCAL programs that interfaces a microprocessor-controlled stepper motor microscope stage (MDACE 1000) to an IBM PC or PC-AT or compatible microcomputer via a serial interface (RS-232) is described. These programs can be integrated into other software written in either BASIC or PASCAL, or used via a menu program that directs the routines to control scanning patterns and to locate the microscope stage to a selected area of the slide. Coordinates of significant events on a slide can be stored on a disk file to allow future examination. The software and interface also provide control of a filter wheel in the microscope for use in multicolor fluorescence assays.

Motorized microscope stage movement can significantly increase the power of cell analysis systems using digital image processing. Such stages are also valuable in microscopy without image analysis. Through automation, precise movement of the stage is achieved, thereby allowing uniform, complete and accurate scans of a slide. Systems have been developed that use stepper motor stages to track the movement of living cells, to trace and reconstruct nerve pathways, to scan cervical smears to produce an automated differential blood count and to perform other image analysis procedures. Such systems were developed either as complete analyzers with dedicated computers or using various minicomputers. However, with constantly decreasing prices and increasing power, microcomputers are becoming attractive for developing image analysis systems. Moreover, the low cost of microcomputers should allow their widespread use in manual microscopy for data collection. Cell analysis with such systems, whether performed by image analysis or by manual methods, would benefit from interfacing the microcomputer to a motorized scanning stage.

This paper describes a package of modular programs that interface a microcomputer to a commercially available microprocessor-controlled stepper motor stage. These programs have been developed in both PASCAL and BASIC, allowing their incorporation as software "tools" into microcomputer-based microscopy systems developed in either of these languages.

From the Transplantation Research Program Center, Naval Medical Research Institute, Bethesda, Maryland, and Ludl Electronic Products, Scarsdale, New York, U.S.A.

Mr. Lee, an Undergraduate Student at Cornell University, was Biological Aid, Naval Medical Research Institute.

Dr. Reynolds was Research Medical Officer, Naval Medical Research Institute, and is now Assistant Professor of Pediatrics, Jonsson Comprehensive Cancer Center, University of California at Los Angeles Medical Center, Los Angeles, California.

Address reprint requests to: C. Patrick Reynolds, M.D., Ph.D., Department of Pediatrics, UCLA Medical Center, Los Angeles, California 90024.

This study was supported by Naval Medical Research and Development Command Work Unit MFS-527.007-0003. The opinions and assertions contained herein are the private views of the authors and are not to be construed as official or reflecting the views of the Navy Department or the Naval Service at large.

Received for publication January 21, 1987.

Accepted for publication June 4, 1987.
Materials and Methods

The programs to control the stepper motor stage were developed on an IBM PC-AT with a standard serial interface. The stage controller consists of a 6-MHz 8085 microprocessor with 24K RAM and 24K ROM in addition to a proportional X,Y joystick (MDACE 1000, Ludl Electronics, Scarsdale, New York). The MDACE 1000 provides a serial RS-232 interface that connects an IBM PC or PC-AT microcomputer to the MDACE controller, allowing the microcomputer to control variable scan patterns and to interrogate stage coordinates. The MDACE controller also controls a filter wheel (positioned between the microscope and a video camera), which allows switching of six band-pass filters. The stage interfaced to the controller contains stepper motors and 1.0-mm lead screws in the steppers with drivers for the X and Y axes. The various components of the MDACE 1000 consist of the stage controller, filter wheel controller, main power supply, programming module and central processing unit (Figure 1).

Routines to control the stage were first written in Microsoft BASIC (Microsoft Corporation, Bellevue, Washington) and then translated to Turbo PASCAL (Borland International, Scotts Valley, California). Serial input/output (I/O) procedures from the Turbo Asynch Tools Package (Blaise Computing, Berkeley, California) were used to control the serial interface. A DCE RS-232 cable was used to interface the controller to the computer (Figure 2).

The MDACE 1000 allows programmable increments from one step to 8,000,000 steps, a scanning distance of ±8,000,000 steps, a scanning speed from 0.001 mm to a maximum of 16.00 mm per second (50,000 steps per second) and a communication rate from 110 to 9600 baud. The use of 1.0-mm lead screws allows stepping increments as small as 0.1 μm.

Results

As originally designed, the MDACE 1000 provides programmable control of the stage stepper motors via a keypad on the controller. Scan patterns can be programmed using the keypad and stored in the non-volatile random access memory of the MDACE 1000. To improve the user interface and versatility of the system and to allow interaction with computer image analysis systems, we developed software to control the stage from a microcomputer.

The communications syntax follows standard RS-232 asynchronous serial communication protocol. In sending data to the stage controller, such as target positions or stepper motor speed, the process described in Figure 3 is followed. The stage controller acknowledges each instruction by returning an "echo" of the command. However, reading information from the stage controller, such as the current position of the motors, requires that an inquiry code is sent (Figure 4). An "echo" of the instruction is not
Table I  ASCII and Hexadecimal Equivalent Instruction Codes Used to Control the Stage Via the Serial Communications Port

<table>
<thead>
<tr>
<th>Code description</th>
<th>HEX code</th>
<th>ASCII code</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ DISPLAY</td>
<td>$H72</td>
<td>114</td>
</tr>
<tr>
<td>ENABLE CPU DISPLAY</td>
<td>$H44</td>
<td>68</td>
</tr>
<tr>
<td>DISABLE CPU DISPLAY</td>
<td>$H45</td>
<td>69</td>
</tr>
<tr>
<td>FILTER WHEEL ADVANCE</td>
<td>$H46</td>
<td>70</td>
</tr>
<tr>
<td>FILTER WHEEL REVERSE</td>
<td>$H4E</td>
<td>78</td>
</tr>
<tr>
<td>TEST BUSY (MOTOR)</td>
<td>$H0</td>
<td>00</td>
</tr>
<tr>
<td>LOAD BASE</td>
<td>$H41</td>
<td>65</td>
</tr>
<tr>
<td>READ BASE</td>
<td>$H61</td>
<td>97</td>
</tr>
<tr>
<td>LOAD SPEED</td>
<td>$H53</td>
<td>83</td>
</tr>
<tr>
<td>RUN MOTOR</td>
<td>$H47</td>
<td>71</td>
</tr>
</tbody>
</table>

An example of the program in Turbo PASCAL that adjusts the stage’s stepper motor speed is shown in Figure 5. The program opens the serial port and then sends the appropriate ASCII code to the stage, which adjusts the stepper motor rate using the protocol outlined in Figure 3. Similar programs control other features of the stage by using the appropriate ASCII codes from Table I. Although identical routines for control of the stage were written in both Microsoft BASIC and Turbo PASCAL, the PASCAL syntax allows defining tasks as “procedures.” The individual procedures can then be easily utilized according to the programmer’s needs. Table II lists the individual PASCAL procedures developed to control the stage; Table III lists the functions provided by the Turbo Asynch Tools used in serial I/O communications. By combining the appropriate procedures and functions, programs for scanning various types of specimens can be easily and rapidly developed.

Another feature this software provides is the ability to store coordinates of any event identified during scanning. Marking a significant event is achieved by storing to a disk file the number of motor steps moved along the X and Y axis from a predetermined “base” or zero point. Returning to the event is accomplished simply by incrementing to the same number of motor steps away from a predetermined base. This method allows the software to store an unlimited number of events. The 1-µm resolution of the stage allows accurate relocation of a significant object.

Two methods of defining scanning patterns can be used with this software. One method consists of setting the size of the scan increment, establishing the number of increments, and looping through the movement procedure the appropriate number of times. The second method reads X-Y coordinates from a file, and then sends the stage to the desired location. A necessary capability for both methods is to set a reference point, or coordinate origin of the slide, so that the stage is aligned and can then return to marked locations of significant events. Manual
control of the stage motors is provided through a joystick controller, which is useful for setting this reference point.

To provide a “user-friendly” interface with the stage operator, the various routines for control of the stage can be integrated under control of a menu-driven program. The menu from an example control program is shown in Figure 6. The program is designed to allow variable scan patterns for particular slides with options to (1) move defined distances along the X and Y axes, (2) adjust the stepper motor speed, (3) set the coordinate origin (reference point = 0,0) as the current stage location, (4) display coordinates of the current location, (5) change the scan increment size, (6) rotate the filter wheel and (7) perform user-defined scan patterns of a slide.

Discussion
The microcomputer software developed for a commercially available motorized microscope stage controller provides microcomputer control of various scan patterns and motor speeds and storage of coordinates allowing relocation of areas of interest. Via a

---

Table II

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convert</td>
<td>Converts data into integers</td>
</tr>
<tr>
<td>stage_drive</td>
<td>I/O using Turbo Asynch Tools</td>
</tr>
<tr>
<td>Open_ComI</td>
<td>Opens the serial port calling Asynch functions</td>
</tr>
<tr>
<td>stage_write</td>
<td>I/O using Turbo PASCAL standard I/O functions</td>
</tr>
<tr>
<td>Speed_Rate</td>
<td>Sets the stepper speed of the motor</td>
</tr>
<tr>
<td>Position_Absolute</td>
<td>Sets the coordinate origin (0,0)</td>
</tr>
<tr>
<td>Move_Stage</td>
<td>Sends codes to the respective motors</td>
</tr>
<tr>
<td>InquireCoordinates</td>
<td>Reads the position of the stepper motor</td>
</tr>
<tr>
<td>ReadDisplay</td>
<td>Reads the keypad display</td>
</tr>
<tr>
<td>Exist</td>
<td>Checks the disk directory for an existing file</td>
</tr>
<tr>
<td>Stepper</td>
<td>Sets a target position and moves the stage</td>
</tr>
<tr>
<td>StageX</td>
<td>Sends a target position to the x-axis motor</td>
</tr>
<tr>
<td>Test_Busy</td>
<td>Tests to see if a motor is still moving</td>
</tr>
<tr>
<td>StageY</td>
<td>Sends a target position to the y-axis motor</td>
</tr>
<tr>
<td>SetUp</td>
<td>Sets up the scan patterns</td>
</tr>
<tr>
<td>Scan_Slide</td>
<td>Implements the scan pattern and saves coordinates to a disk file</td>
</tr>
<tr>
<td>Filter</td>
<td>Sends codes to the filter wheel</td>
</tr>
<tr>
<td>Speed</td>
<td>Changes the stepper rate of the motors</td>
</tr>
<tr>
<td>Readback</td>
<td>Recalls saved coordinates of significant locations</td>
</tr>
<tr>
<td>Menu</td>
<td>Displays a menu of the possible choices</td>
</tr>
</tbody>
</table>

---

Table III

<table>
<thead>
<tr>
<th>Turbo Asynch functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToHex</td>
<td>Converts a decimal value to hexadecimal</td>
</tr>
<tr>
<td>Open A1</td>
<td>Initializes the serial port</td>
</tr>
<tr>
<td>Close A1</td>
<td>Closes the serial port</td>
</tr>
<tr>
<td>WrtChA1</td>
<td>Writes a character to the output buffer of the serial port</td>
</tr>
<tr>
<td>RdChA1</td>
<td>Reads a character from the input buffer of the serial port</td>
</tr>
</tbody>
</table>
serial (RS-232) interface, the microcomputer is able to record the number of motor steps that the stage moves, thus defining an X-Y coordinate axis. The ability to interrogate and store coordinates of observed "events" to a disk file allows recall of the observed object for observation at a later time. In addition to control of the stage, the MDACE controller provides control of a filter wheel positioned between the microscope and an intensified (ISIT) video camera. The software developed also allows control of the filter wheel by the microcomputer. We have found the filter wheel useful for changing band-pass filters in multicolor fluorescence assays.

Computer-controlled microscope stages provide rapid, accurate slide movement, thus reducing operator fatigue in scanning large numbers of specimens. In addition, such a system allows semiautomatic collection of various data during microscopy, which can be coordinated with locations on the slide. Examples of techniques in which such "computer-aided microscopy" can be useful include grain counting in autoradiography, reading various immunologic assays and neuroanatomic mapping. The programs described here will also be useful for more automated systems that use digital image analysis for detection and quantitation of various types of cells.

An advantage of developing programs for microcomputers is that a large number of programmers are working in an identical (or at least compatible) environment. This has led several vendors to develop powerful sets of programming "tools" that provide callable subroutines for various tasks, thus speeding up program development. We have made use of such tools in developing this software (Blaise Computing Turbo Asynch). The PASCAL software we have developed can be modified and compiled with Microsoft PASCAL (Microsoft Corporation, Bellevue, Washington) or PASCAL-2 (Oregon Software, Portland, Oregon), both of which allow linking to compiled FORTRAN or C programs, thus extending the value of this software as "tools" for programmers in other languages.

One of the obstacles to developing effective automated microscopy systems has been the necessity of obtaining hardware (often custom designed) that will accommodate software developed by other investiga-

**Figure 5**

I/O procedures. Instructions are sent via serial RS-232 asynchronous format. Shown are sample procedures for reading and writing to the serial port and setting the stepping rate of the motor. Parameters follow normal PASCAL syntax.

```pascal
procedure stage_drive(order:integer);
begin ( stage_write )
  Errorcode:= WrtchA1(COM1,chr(order));
  Errorcode:= RdchA1(COM1,ch,InQSize,Portstatus);
end ( stage_write );

procedure Open_Com1;
begin ( Open_Com1 )
  Errorcode:= OpenA1(COM1,100,100,0,addr(iobutton));
  if (Errorcode = Port__Open__Already) then
    write('Com1 is already opened. '
          Errorcode)
  else if (ErrorCode = OK) then
    write('Com1 successfully opened')
  else
    write('Cannot open Com1. Error Code = ',ErrCode:3);
write(InQSize,' characters remain in the input buffer.');
end; ( Open_com1 )

procedure stage_write(order:integer);
begin ( stage_write )
  write(aux, ch(order));
  read(aux, ch);
  writeln('code echo ', ch);
end ( stage_write );

procedure Speed_Rate(device:integer);
  write speed to the stage
begin ( Speed_Rate )
  Stage_drive(58);   { first : }
  Stage_drive(device);   { dev = 0 }  
  Stage_drive(63);   { inst = $1153 }  
  Stage_drive(2);   { data = 2 }  
  Convert(50000.0);
  Stage__drive(idat__1);
  Stage__drive(idat__2);
  Stage_drive(58);   { last : }
end ( Speed_Rate )
```

**Figure 6**

An example of a menu for the microcomputer-controlled microscope scanning stage programs.

**MICROCOMPUTER INTERFACE FOR CONTROL OF A MICROSCOPE SCANNING STAGE**

[X] Stage Movement - x-axis variable stage movement  
[Y] Stage Movement - y-axis variable stage movement  
[S]peed Control - Change the scanning rate  
[Position] [Absolute] - reset current stage position as (0,0)  
[Display] the keyboard CPU coordinates  
[Inquire] [Coordinates] - read the coordinates of the motor stepper  
[F]ilter Wheel - step forward one filter  
[F]ilter [R]everse - step reverse one filter  
[Display] [Current Filter]  
[M]ove - combined X and Y stage movement  
Redisplay this menu

[Q]uit, and return to data processing  
Table IV  Comparison of Several Commercially Available Microscope Scanning Stages

<table>
<thead>
<tr>
<th>Stage (and vendor)</th>
<th>Resolution (µm)</th>
<th>Scan area (mm²)</th>
<th>Maximum speed (mm/s)</th>
<th>Approximate cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marzhauser Mechanical Stage EK32</td>
<td>0.1</td>
<td>75 x 50</td>
<td>20</td>
<td>$5,000 (stage)</td>
</tr>
<tr>
<td>Boeckeler Motorized X-Y System</td>
<td>5.0</td>
<td>150 x 150</td>
<td>50</td>
<td>$10,875*</td>
</tr>
<tr>
<td>Stepper Motor Positioning Stages</td>
<td>1.0</td>
<td>50 x 50</td>
<td>25</td>
<td>$9,345*</td>
</tr>
<tr>
<td>MDACE 1000 XY Stage Controller</td>
<td>1.0</td>
<td>88 x 88</td>
<td>6.7</td>
<td>$8,895</td>
</tr>
<tr>
<td>99500e Stage and Controller</td>
<td>0.1</td>
<td>75 x 75</td>
<td>16</td>
<td>$5,948 (stage)</td>
</tr>
</tbody>
</table>

*User must provide the stage from an existing microscope; price includes modification of existing stage and object code for IBM PC control software.

As the number of such software tools increases, development of microcomputer digital image microscopy systems will become increasingly easy. The software presented here provides a useful set of tools for microscope stage control, not only for integration with image processing systems, but also for use in semimannual (computer-assisted) microscopy.

References
18. Glaser EM, McMullen NT: Semaautomate microscopy as a


