A SOFTWARE INTERFACE BETWEEN AIPS AND THE TV DEVICE AYDIN 5216

L. Feretti, M. Nanni

IRA 82/1985 Rapporto Vax

RAPPORTO VAX 11/780

CONSIGLIO NAZIONALE DELLE RICERCHE

ISTITUTO DI RADIOASTRONOMIA

c/o ISTITUTO DI FISICA « A. RIGHI »
Via Irnerio, 46 - 40126 BOLOGNA (Italy)

1. INTRODUCTION

The most useful implementation of the AIPS system (Fomalont and Bridle, 1983) includes the use of a peripheral TV device, capable of displaying images with multiple levels of color or intensity. Some programs use the TV in an interactive way, for instance to select boxes with the cursor or to change the color table; some simply display images at intermediate stages of reduction.

The TV device used at NRAO is the International Imaging System (IIS), Models 70/E and 70/F. Additionally, AIPS runs on the GOULD DeAnza, thanks to routines implemented by W.Jaffe. The basic architecture of IIS includes 4 image channels (memory planes), 4 graphic channels, a trackball. Each channel is 512x512 pixel in size and 8 bits wide; the graphic channels, each of 1 bit, can be overimposed to the image channels. Each channel can hold one 512x512 image or several smaller images, which can be displayed with up to 256 or 4096 colors simultaneously. The data stored on the channels can be handled directly from the TV device, for operations like zoom, scroll, sum and subtraction of images. The trackball with 4 buttons is used for controlling of the programs or for interactive operations. There are, moreover, several other subunits in the IIS, which include an input function memory, a histogram generator, a feedback arithmetic logic unit, shift and min/max registers. These subunits are used for esoteric operations, but

they should not be required for other kinds of TV device (for instance, they are not included in DeAnza).

We have implemented on the TV device available at the Istituto di Radioastronomia all the main operations performed by AIPS on the IIS. It is possible to display an image, to change color tables, to overlap graphic planes to the image, to use cursor for graphic management, to zoom and to scroll; it is instead not possible to overlap more images, due to the limitations of our device.

2.THE AYDIN 5216

The TV device available at IRA is the Aydin 5216, 'IMAGE' version, a display computer whose processing ability is built around the INTEL 8084 16-bit microprocessor. It has 1 channel 1024x1024 pixel in size and 8 bits wide, a programmable LUT, a joystick, a keyboard. The Aydin can support a system configuration with up to 8 planes, each of 1 bit, and each channel can be subdivided into halves and quarters, which are referred to as partitions, or minor channels.

An "Aydin Standard Software" must be loaded from the host computer to start up the pictorial device: this is the first operation to do after switching it on. Graphic commands may then be supplied through coded instructions by the host computer, or, alternatively, through keyboard inputs. The keyboard support

includes a set of programmed function keys, which can execute most of the available functions; it also has a program buffer in which keyboard function commands can be stored.

The basic commands of the device permit it to work in 2 ways:
1) pixel mode, in which data are addressed one pixel at a time in a number of selected planes; images can be displayed in this way with up to 256 colors;
2) graphic mode, in which graphics can be drawn on 8 independent planes of 1 bit which can be overlapped in the display.

The basic operations handled by the TV internal processor are performed only on display partitions; this implies that each operation (sum, multiplication, interpolation, zoom) requires both an input and an output display partition. This implies, in particular, that it is not possible to obtain a scrolling zoom.

Color values are based on an 8-bit word, where the least 3 bits drive the Red raster gun (with intensity levels from 0 to 7), the next 3 bits drive Green raster gun, and the high 2 bits drive the Blue gun (intensity 0 to 3). A switch on the TV screen enables black/white display of an image. In this case a scale of only 8 gray levels is available, since the contribution to different intensity levels is provided by each memory plane of 1 bit.

No histogram generator is available; therefore the histogram equalization of an image through specialized hardware is not possible.

A comparison between the original AIPS TV device and the

Aydin shows that the Aydin can display a bigger image (1024x1024 instead of 512x512 pixels), but it has only 1 image channel and no independent graphic planes. Since AIPS TV routines manage images of maximum 512x512 pixels, it follows that 3/4 of the total Aydin available memory (1 Mb) is in practice useless. Unfortunately, it is not possible to redefine this memory amount to produce 4 overlapping channels of 512x512, instead of 1 channel 1024x1024. We do not exclude the possibility of modifying, in the future, the AIPS routines to handle a 1024 image.

Other differences concern the interface between the host computer and the TV device and affect the speed of image loading and of operations controlled by the joystick: the Aydin is slower but the difference is not dramatic. The presence of only 1 memory channel of 8 bits in the Aydin is the real limit in our emulation of IIS, since at least an image memory plus some graphic planes are needed for the most important TV operations of AIPS. Therefore, we have reduced to 6 the number of bits of the image channel and we have devoted the 2 remaining planes of 1 bit to overlay graphic planes. In this way an image can be loaded and displayed with 64 colors. It is not possible to load many images and to display them simultaneously in overlapped channels; consequently it is not possible to do operations like blink, superposition of images, etc. It is instead possible to overlap two graphic planes (labels, contours, windows) to the image.

Interactive operations are handled on the IIS by means of a

trackball with 4 buttons. On the Aydin we have a complete alphanumeric keyboard; our software interface uses only the numerical keypad as buttons. Unfortunately the inquiry from the host computer of the button pushed is a complex process of enable/disable of the Aydin keyboard; this interrupts slow down the joystick based operations.

3.THE AIPS-AYDIN SOFTWARE INTERFACE

Special libraries of subroutines, subdivided by model of TV device, are contained in AIPS subdirectories for the management of the TV (Cotton et al., 1984). These subroutines are called Y routines, because all of them have name beginning with the letter Y. Moreover, special Z routines are used for the basic I/O operations of open, close, initialize, read/write.

The software interface between AIPS and the Aydin 5216 TV device consists of new Y and Z routines, which are in the subdirectory [AIPS.reldate.APL.YSUB.AYD]. The compile procedures write the object code into the link editor library in the [AIPS.reldate.APL] area.

We present in the following a short description of the implemented subroutines, grouped according to the general operation handled.

1 - Initialization routines:

YINIT initializes the TV device, clears the TV memories,

resets the TV I/O parameters,

YTVOPN performs a system "OPEN" on the TV device, by calling the appropriate Z routine,

YTVCIN initializes the common which describes the characteristics of the interactive display devices and the common which has the current status parameters of the TV; in the present case we have put, in particular,

NGRAY=1 (number of image planes)

NGRAPH=2 (number of graphics planes)

MAXINT=63 (max number of intensity levels)

MXZOOM=2 (max zoom power)

ZAYDPW loads the Aydin Standard Software and sets the screen to a resolution of 512x512 pixel,

ZAYDOP opens the logical files.

2 - Close routines:

YZERO clears a gray or graphics memory by the fastest possible method,

YTVCLS closes the TV device by calling the appropriate Z routine,

ZAYDCL closes the TV device.

3 - Channel selection routines:

YSLECT enables and disables gray and graphics planes,

YGRAPH enables and disables graphics overlay planes by altering the graphics color LUT.

4 - LUT management routines:

YLNCLR computes a piecewise linear LUT and writes it to the

Aydin,

YOFM reads/writes the individual channel LUTs for Black and White enhancement,

YLUT reads/writes full channel LUTs to Aydin for pseudo-color enhancement,

YGGRAM reads/writes the Aydin graphics color assignment LUT.

5 - Write routines:

YCHRW writes characters into the image or graphics plane,

YCNECT writes a line segment on the image or graphics plane,

YGYHDR enables the device to accept image data,

YIMGIO reads/writes a line of image data from/to a gray scale or a graphics plane.

6 - Cursor read/write routines:

YCUCOR reads the cursor position and returns its coordinates in unit of image pixels,

YCURSE enables/disables the cursor,

YCRCTL reads/writes the button values and the cursor position in units of screen pixels.

7 - Scroll and zoom routines:

YSCROL performs non destructive scroll of the image and/or graphics plane,

YZOOM performs a zoom of the image displayed on the screen, up to an image 4 times as large.

The operations described above are realized by calling appropriate routines, which either are Aydin provided routines or have been written by us for slightly more complicated commands.

All these subroutines are contained in the directory [DGRNEW] and its subdirectories, and the object modules are in the library [DGRNEW]AYDLIB.OLB, which is linked with the other AIPS libraries.

As pointed out in the previous section, the joystick based operations are rather slow, due the interrupts sent by the host computer to the Aydin, when the pushed button must be queried. We note also that no answer is returned to the host if no button is pushed, i.e. the answer BUTTON=0 is not interpreted as "no button pushed". The slowness of the host-Aydin interaction implies that the TV device is unable to perform all the operations requested by the host; some commands are not executed and the synchronism between the host and the TV is lost, giving rise to an uncontrolled situation of errors and wrong answers, resulting in the TV device crash. To solve this problem, we have modified a few high level routines, which are quite general and TV independent, in order to allow an asynchronous motion of the cursor in all the cases, except when the cursor modifies the LUTs. The modification consists of a call to the new subroutine DLINEW (instead of DLINTR), which inhibits the host to send command to Aydin while cursor is moving. Control is returned to the host as soon as a button is pressed. The routines affected are AU6, AU6B, GRBOXS, GRLUTS, TVFIDL. The routine DLINEW is in the subdirectory with the Y+Z routines.

4.INSTALLATION OF AIPS WITH THE USE OF AYDIN

The implementation of the Aydin under AIPS is best done when AIPS is installed. The AIPS installation consists of two steps: the first one (@ILOAD) loads all the routines from tape to disk, the second one (@IBUILD) compiles and links the tasks.

The Aydin routines are inserted between these two steps. To easily perform all the needed operations, we have written in a tape the three following backup save sets:

AYDBOO.BCK : installation procedures

AYDSUB.BCK : AIPS Y+Z routines

AYDLIB.BCK : Aydin subroutines

After the step @ILOAD, the tape with the previous files must be mounted with the command:

```
$MOUNT/FOR MT:  TAPE TAPE
```

and then the first of the save sets must be loaded and a command file must be copied in the directory [AIPS], as follows

```
$BACK MT:AYDBOO.BCK  [AIPS.INST]
$COPY [AIPS.INST]AYDINSTL.COM   [AIPS]AYDINSTL.COM
```

At this point, it is sufficient to run

```
$@AYDINSTL
```

This command performs the following operations:

1 - Assign logical TVDIR to the subdirectory

[AIPS.reldate.APL.YSUB.AYD]

2 - Load the other save sets from tape to disk, if necessary

3 - Edit [AIPS] command files to substitute IIS with AYD

4 - Edit some high level routines to substitute the call to DLINTR with DLINEW

5 - Edit command files to include the Aydin library in the link (files to be edited are listed in [AIPS.INST]LIST.LNK)

6 - Edit [AIPS.reldate] command files to substitute IIS with AYD

7 - Edit utility files to include Aydin library in the link (files to be edited are listed in [AIPS.INST]LIST.UTL)

If no error message is given, the step @IBUILD can be done and the standard installation procedure can be continued.

The procedure described above has been debugged with the 15JUL85 version of AIPS. For subsequent versions, the file AYDINSTL.COM may need to be changed if modifications of the high level routines edited by step 4 or of the command files have occurred. We hope that the middle level routines that use the IIS are not subject to change.

5.USER MANUAL

The use of AIPS verbs and tasks with Aydin is not substantially different from that with other TV devices, except that tasks which need more than one image channel are not implemented, due to the limitations of the Aydin. We list in table 1 the available verbs and tasks (V,T), in alphabetical order; their use is extensively explained by the AIPS help files.

There are some minor incoveniences in the use of some of them, which are indicated in the "Note" column. We indicate with "b/w" the commands that use a black/white table: they need the use of the manual switch on the TV monitor (see Sec. 2). Some commands do not give any error message, but do not perform any visible operation: they are indicated with "no". Finally, some operations which involve the motion of the cursor work in a different way from the original (IIS) implementation, since an asyncronous communication of the cursor position has been introduced (see Sec. 3): they are indicated with "a.c.". In these routines, the user moves the cursor to the desired position and, after, hits the convenient button to perform the desired operation (see table 2).

It is recommended that the TV user waits a few seconds after moving the cursor before any other operation. In the case of a mess, pressing "BUFFER" on the Aydin keyboard (the second from left of the second row) may correct the situation.

The use of commands which return the cursor position may be problematic, after zooming and/or scrolling an image. While IMXY seems to give a correct result, others may be not O.K.

Some tasks use the TV to display data at intermediate stages of reduction; the most used are APCLN and MX. In these tasks a message says:

"HIT BUTTON D WITHIN 15 SECONDS TO STOP CLEANING NOW
HIT BUTTON A,B C TO CONTINUE SOONER".

As mentioned above, a button must always be pressed to return the

control to the host computer, therefore in the previous tasks buttons A, B or C must be pressed to continue the reduction: if no button is pressed the reduction does not continue automatically, since the host waits for a button answer. For the same reason, there is not the time limit of 15 seconds to stop the program.

TABLE 1

Verbs and Tasks

Name	V/T	Note	Name	V/T	Note
BLANK	T		TVINIT	V	
CURBLINK	V	no	TVLABEL	V	
CURVALUE	V	a.c.	TVLOD	V	
GRCLEAR	V		TVLUT	V	b/w
GROFF	V	no	TVMLUT	V	b/w
GRON	V	no	TVNAME	V	
IMERASE	V		TVOFF	V	no
IMPOS	V		TVON	V	no
IMWEDGE	V		TVPL	T	
IMXY	V		TVPOS	V	
OFFPSEUDO	V		TVPSEUDO	V	
OFFSCROL	V		TVRESET	V	
OFFTRANS	V	b/w	TVSCROL	V	
OFFZOOM	V		TVSLICE	V	a.c.
REBOX	V		TVSTAT	V	
SETXWIN	V		TVWEDGE	V	
TVALL	V		TVWINDOW	V	
TVBOX	V	a.c.	TVXY	V	
TVCLEAR	V		TVZOOM	V	a.c.
TVFIDDLE	V		TVWLABEL	V	

TABLE 2 Button Conversion

IIS trackball	Aydin keypad
A	1
B	2
C	4
D	8
'Reset'	'BUFFER'

ACKNOWLEDGEMENTS

We wish to thank the Radiosterrenwacht Dwingeloo and the Kapteyn Laboratorium of Groningen for giving us the opportunity of studying the performances of TV devices.

REFERENCES

Cotton, W.D. et al: 1984, Going AIPS, Charlottesville

Fomalont, E.B., Bridle, A.H.: 1983, The AIPS Cookbook, National Radio Astronomy Observatory, Charlottesville