PRL TN-2005-84

PRL Technical Note

Interactive Image Acquisition and Display Program FLAREVISION for ImagePoint CCD Camera

SANJAY GOSAIN

MAY 2005

Physical Research Laboratory
Navrangpura, Ahmedabad 380009

Contents

Introduction:

The solar H-alpha images are routinely observed from the island observing station of Udaipur Solar Observatory amidst Fatehsagar Lake. In the past these observations were recorded on 35 mm format Kodak Technical Pan 2415 photographic films. With the introduction of electronic CCD cameras in the observatory in early 90s the observations were recorded digitally using frame-grabbers and also using time-lapsed video cassette recorder (VCR). With the development in CCD technology and associated hardware and software the modern cameras are interfaced to PC with possibility of direct digital readout and no need for frame-grabbers. Currently the CCD camera in use at USO for flare observations is ImagePoint CCD camera from M/s Photometrics Inc., USA. This camera has both video output as well as digital readout port. Initially, this camera was used in a video mode in which the video output of this camera was fed to the frame grabber installed in the PC and the observer used to capture still frames and save them to hard disk manually. This observing mode had many limitations such as (i) non uniform cadence, (ii) too much manual intervention, (iii) re-sampling of original image data in frame grabber and, (iv) electronic noise interference in the images.

The possibility of digital readout from the CCD camera was not fully exploited as user-friendly software for this purpose was not in place. It was realized that the limitations of the earlier observing mode, mentioned above, could be completely overcome if the digital readout mode of this camera could be exploited. So, initiative was taken at USO for development of an interactive image acquisition and display program which we call FLARE-VISION, because this camera is dedicated for the purpose of observing solar flares in H-alpha wavelength. The utility programs written in C language were supplied by the camera manufacturer for digital readout from the camera. Taking that as the starting point we developed the fully interactive image acquisition and display tool FLARE-VISION which is described in this technical report.

1. CCD Camera Description:

The detailed description of the camera hardware is available in the camera manuals so here we give in brief the key features of the CCD camera. The camera is an inter-line transfer CCD camera with CCIR format. The camera has square pixel array of 640x480 pixels and the pixel size is 11 microns. The 8 bit digitizer digitizes the pixel signal and stores the values in an onboard buffer. The video signal is generated by scanning this frame buffer continuously. The digital readout of this onboard buffer is possible using the parallel port (also called Centronics port) of the PC which has 8 bit I/O. While the digital readout is in progress the video mode will show black out as the frame buffer is busy with parallel port. The camera has a utility program to demonstrate the parallel-port readout capability. While with video mode one can put a BNC cable and get the video images onto a CRT monitor or feed to a frame grabber card and grab the frames. In the present setup we use video only for monitoring purpose while the actual observations are made using parallel port readout.

1.1 Image Acquisition:

The image acquisition is performed via a C program, which reads image data in the camera buffer via the Centronics port of the PC. The main function used is *readpic()* which is available from the vendor in utility program disk. The program as such is not directly usable and needs to be modified to suit our needs. The modification done is to write a separate C program which calls *readpic()* function along with commandline argument so a standalone EXE is made which now can be called by *Shell()* functionin Visual Basic or *SPAWN()* function in IDL. Once we have such a EXE this can easily be called for image acquisition with appropriate arguments from higher level programming languages like VB or IDL. This is the philosophy used in FLARE VISION. The details of *readpic()* function is given in Appendix III. So, in our implementation we have used VB for providing a GUI and user can pass arguments via input BOXES. Once we have user

parameters from the VB form an IDL function is then executed with these arguments. The IDL then creates appropriate log files andd list of image files and executes the camera EXE via spawn() function.

2. Desired features in the software:

The desired features in the interactive image acquisition and display program FLAREVISON can be listed as follows:

(i) digital readout from the camera,
(ii) two modes of observing namely **FLARE** mode (*for observing flaring active regions or filament/prominence eruptions*) and **NORMAL** mode (*for observing quiet regions, filaments, prominences, or non-flaring active regions*)
(iii) variable cadence of observations,
(iv) number of images to acquire,
(v) keeping all images in fast observing mode and only the best contrast frames in normal slow mode of observations,
(vi) generate unique folder and image file name which are self explanatory and keep log of all the observing settings
(vii) a display tool for online image display and batch display of existing data with variable speed,
(viii) an online interactive data analysis tool for visualizing histogram etc. and
(ix) making MPEG movies out of a sequence of flare images.

3. Modes of observation:

There are two modes of observations followed at USO. Namely, these are **FLARE** and **NORMAL** modes. The main difference in theses two modes is that the **FLARE** one aims at capturing the images of dynamic events like solar flares, filament

eruptions, prominence eruptions etc. that can last from few to several minutes. So, to capture the evolution of the event every single frame is captured and kept, irrespective of the image quality. While in **NORMAL** mode the aim is to capture morphology of solar active regions, which are not in a dynamic state. Here idea is to keep few, typically one good picture every five minutes, of the active region. So, while in **FLARE** mode sequence of images are captured in burst mode and kept, in **NORMAL** mode a burst of 10 to 20 images (depending on the atmospheric seeing on that day, good seeing requires less frames) is taken and passed on to a program which computes the best image from this burst and keeps that image for record while the user has a choice to discard rest of the images. The best contrast is computed by measuring a parameter, which is the integral of RMS intensity fluctuations in an image.

In this way **NORMAL** mode is repeated every five minutes and hence a best image is kept and to capture any dynamic event a **FLARE** mode is selected.

4. GUI Layout and implementation:

The layout of the graphical user interface was designed in Visual Basic. A snapshot of the interface is shown in figure 1. As can be seen this form has four main sections as described below

(i) **GENERAL PARAMETERS SETTINGS**: In this module main parameters of observations can be set, like the current date, hard-drive, observer's name, NOAA active region number, comments about seeing conditions etc.

(ii) **FLARE MODE SETTINGS:** In this module one can set the parameters of fast mode observations, like number of frames to acquire and the cadence,

(iii) **NORMAL MODE SETTINGS:** In this module one can set the parameters for normal slow mode observations, like number of frames to acquire and cadence, while each frame is the best frame from a burst of frames. The

number of frames in burst can also be set according to seeing conditions and desired data rate,

(iv) **IMAGE DISPLAY:** In this module one selects the list file which will be opened and all the images listed in that file will be displayed in image display panel with a speed which can also be defined.

A. Visual Basic :

The implementation of these features was done using Visual Basic as the main platform for the overall program flow, while directory creation and log file, list file creation are done using intrinsic file I/O features of VB programming language. The parameters can be passed through the text box on the form and event can be initiated with buttons. The event driven programming makes it very easy to control the overall operations. The camera control is done mainly by modifying the demo C program supplied by the vendor so that internal parameters in the C program can be accessed and modified by passing command-line arguments. Thus by executing the executable program on console or DOS shell with appropriate arguments one can acquire images as desired. This can also be executed inside Visual Basic by using ***Shell()*** function which can execute the programs by invoking a DOS shell. Methodology has been to take user input from the VB form and parse these inputs to form a single command and execute it using ***Shell()*** function.

Once a sequence of frames is acquired and a list file containing the names of these images is generated these can be accessed later and displayed as a movie offline. The choice of Visual Basic was made because of its rapid application development environment and easy interface with other packages like IDL.

B. Interactive Data Language (IDL) ActiveX interface:

Although the basic programming features of VB are good but specialized features like image manipulation and processing are not intrinsic to VB and need to be added by calling external image processing and graphics libraries. Here we have chosen the Interactive Data Language (IDL) a popular image processing language in solar astronomy, so that libraries available in IDL can itself be used. IDL provides an ActiveX component called IDLDrawX3 which is available for use in any Microsoft language capable of calling COM (Component Object Model) components. This component is available on the VB toolbar and can be embedded in the VB form interface. The component is a graphics widget on which an image can be displayed or can be used for plotting. One can also access to IDL which is an interpretative language. So, one can execute IDL image functions and graphics display commands. We use this for displaying the acquired images from the camera and also for displaying histogram of the images. The images can be loaded and continuously displayed like a movie on the window. Also, once the image is open it is available as a two-dimensional array in IDL on which one can do variety of manipulation with functions available in IDL. Also, one can change image format from one form to another. We use IDL functions to compute the image contrast and pick the best frame from a burst of frames. IDL provides MPEG creation routines which are used for creating a MPEG movie from a sequence of image files. The MPEG feature is very useful to put the movie of a flare on web which small in size compared to size of all the individual frames. This is very useful for sharing the flare events over email or displaying on webpage.

5. Future Improvements:

In future once can provide following features in the FLAREVISION

(i) **The data-cube generation:** The number of frames acquired per second is about 0.3 in present mode as the readout is via parallel port and this includes

writing time on hard-disk also. In future one can define the data-cube size and allocate memory in system RAM and once the frames are acquired from camera these are stacked in data-cube. This will avoid writing onto hard-disk every time and only at the end writing is done. Thus we can get slightly better frame rates.

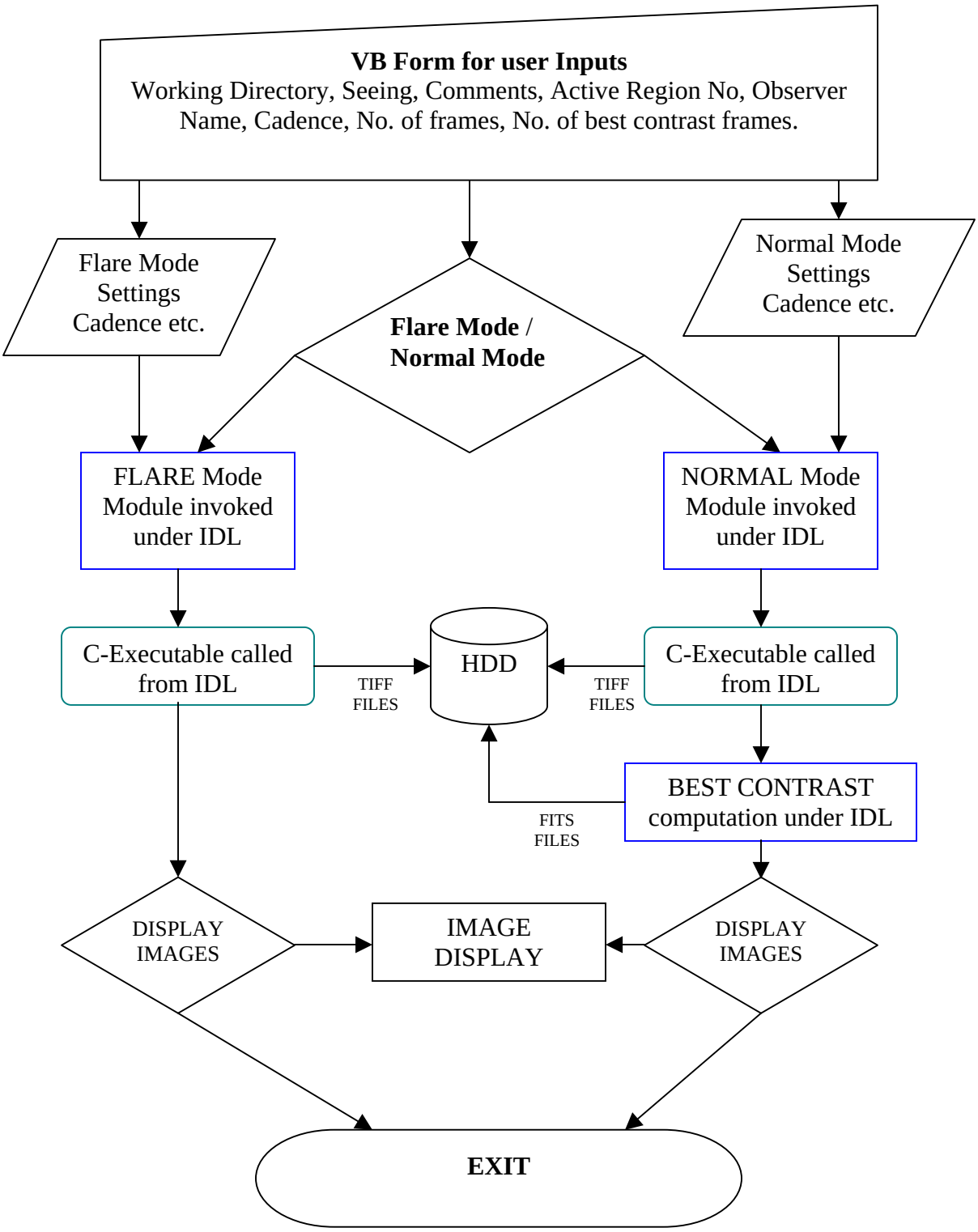
(ii) **Remote Camera settings:** At present the camera settings like the exposure time, gain, LUT etc can be controlled via a keypad. In future one can provide these features via the serial port. This can be done by sending appropriate commands as mentioned in CCD manual over serial port to camera. This will bypass the keypad and can be done remotely.

6. Discussion and Conclusions:

In this software a hierarchical approach to accomplish complex task is implemented. At the lowest level a C –program (see Appendix-III) is used for acquiring the image data using Parallel port and saved onto hard disk as a TIFF file. Thus, basic data acquisition is done by a simple executable "***Islobs.exe***", which can be called with appropriate command-line parameters. At the highest-level Visual Basic is used for user inputs and image display purposes only. While a sequence of nested IDL functions is used for implementing two different modes of observations that is NORMAL and FLARE mode described above. IDL is employed here mainly due to two reasons first, for online image processing like in NORMAL mode where best contrast image is computed from burst of images and saved as FITS file, and second for displaying images on IDL DrawX component on VB form. Thus, the flow of the program is like shown in flow chart below.

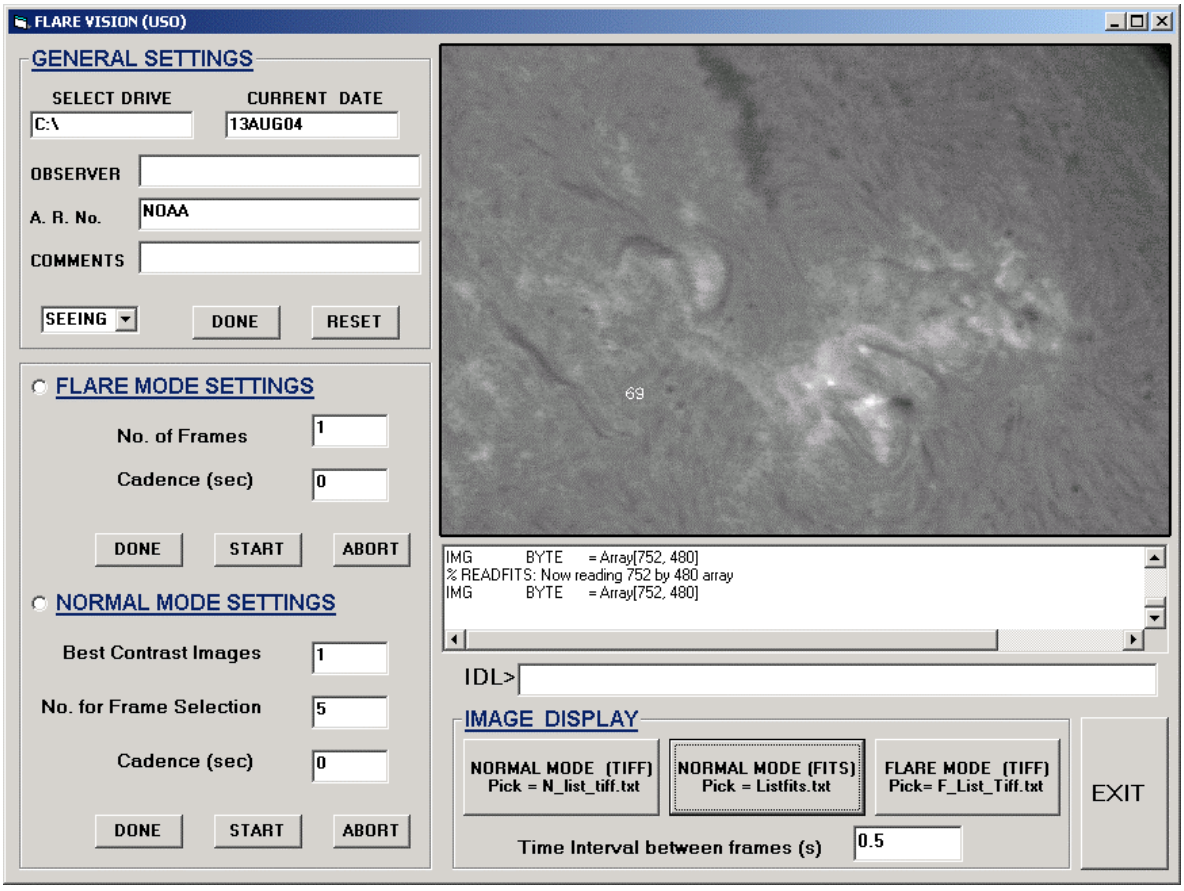
Appendix I

FLOWCHART



Appendix II

SNAPSHOT OF FLAREVISION GUI FOR IMAGEPOINT CAMERA



Appendix III

(VB Code for FLAREVISION)

```
**********    GLOBAL DECLARATIONS FOR VARIABLES   **************
Dim flfr, flcad, nfr, nbfr, ncad, n As Variant
Dim obser, commen, arno, seing, workd, dates, mode, cmd As String

** DISPLAYING NORMAL MODE BEST CONTRAST IMAGES IN GRAPHICS WINDOW **
Private Sub Command10_Click()
Dim cmdn As String
Dim ti As Integer
ti = Val(timgap.Text)           ***** TIME GAP BETWEEN 2 IMAGES ****
cmdn = "dispfits_vb," + Str(ti) ** IDL function dispfits_vb(timegap) **
IDLDrawWidget1.ExecuteStr (cmdn)
End Sub

** DISPLAYING NORMAL MODE ALL IMAGES IN GRAPHICS WINDOW **
Private Sub Command12_Click()
Dim cmdn As String
Dim ti As Integer
ti = Val(timgap.Text)           ***** TIME GAP BETWEEN 2 IMAGES ****
cmdn = "n_disptiff_vb," + Str(ti) **IDL function n_disptiff_vb(timegap)
IDLDrawWidget1.ExecuteStr (cmdn)
End Sub

** DISPLAYING FLARE MODE ALL IMAGES IN GRAPHICS WINDOW **
Private Sub Command14_Click()
Dim cmdn As String
Dim ti As Integer
ti = Val(timgap.Text)           ***** TIME GAP BETWEEN 2 IMAGES ****

cmdn = "f_disptiff_vb," + Str(ti) **IDL function f_disptiff_vb(timegap)
IDLDrawWidget1.ExecuteStr (cmdn)
End Sub

******* INITIALIZATION OF VB FORM WITH IDL GRAPHICS COMPONENT ******
Private Sub Form_Load()
n = IDLDrawWidget1.InitIDL(Form1.hWnd)
If n <= 0 Then
MsgBox ("IDL failed to initialize")

End
```

```
End If
IDLDrawWidget1.CreateDrawWidget
IDLDrawWidget1.SetOutputWnd (IDL_Output_Box.hWnd)
End Sub

******* INITIALIZATION OF INPUT BOXES ON VB FORM ******
Private Sub GeneralParams_Click()
workd = workdir.Text     *****  WORKING DIRECTORY ******
dates = datebox.Text     *****  CURRENT DATE  *****
obser = observer.Text    ***** NAME OF OBSERVER ****
arno = arnum.Text        ***** ACTIVE REGION No. ***
commen = comments.Text   **** GENERAL COMMENTS  ***
seing = seeing.Text      **** SEEING QUALITY ****
End Sub

******* INPUT COMMAND FOR IDL VIA VB FORM ON KEYPRESS******
Private Sub CmdLine_KeyPress(KeyAscii As Integer)
Dim bRet As String

    If KeyAscii = vbKeyReturn Then
        If Len(CMDLine.Text) > 0 Then
            'Send the command to IDL and clear
            bRet = IDLDrawWidget1.ExecuteStr(CMDLine.Text)
            CMDLine.Text = ""
        End If
    End If
End Sub

******* RESETTING VALUES OF INPUT BOXES ON VB FORM ******
Private Sub reset_Click()
workdir.Text = "C:\"
datebox.Text = "13AUG04"
observer.Text = " "
arnum.Text = "NOAA"
comments.Text = " "
seeing.Text = "Seeing"
End Sub


******* SELECTION OF FLARE MODE FOR OBSERVATION ******
Private Sub Option1_Click()
mode = "f"     'Flare Mode
End Sub

******* PARAMETERS FOR FLARE MODE OBSERVATION ******
```

```
Private Sub flaredone_Click()
flfr = Val(Flareframes.Text)  **** No. OF FRAMES TO CAPTURE
flcad = Val(Flarecadence.Text) **** TIME INTERVAL
End Sub

*** INVOKE FLARE MODE OBSERVATION VIA IDL FUNCTION NEWOBSERV_VB() ***
Private Sub flarestart_Click()
Dim i As Variant
Dim flfn, flcom As String
flcom = "newobserv_vb,'" + dates + "','" + arno + "','" + mode + "'," +
Str(flfr) + "," + Str(flcad)
IDLDrawWidget1.ExecuteStr (flcom)
End Sub

*** ABORT THE PROGRAM ***
Private Sub ABRTFL_Click()
IDLDrawWidget1.DoExit
End
End Sub

******* SELECTION OF NORMAL MODE FOR OBSERVATION ******
Private Sub Option2_Click()
mode = "n"    'Normal Mode
End Sub

******* PARAMETERS FOR NORMAL MODE OBSERVATION ******
Private Sub normaldone_Click()
nbfr = Val(normalbestframes.Text) **** No. OF GOOD FRAMES
nfr = Val(normalframes.Text) **** No. OF RAW IMAGES
ncad = Val(normalcadence.Text) **** TIME INTERVAL
End Sub

*** INVOKE NORMAL MODE OBSERVATION VIA IDL FUNCTION NEWOBSERV_VB() ***
Private Sub normalstart_Click()
Dim j, k As Variant
Dim nfn, ncom As String
ncom = "newobserv_vb,'" + dates + "','" + arno + "','" + mode + "'," +
Str(nbfr) + "," + Str(ncad) + "," + Str(nfr)
IDLDrawWidget1.ExecuteStr (ncom)
End Sub
*** ABORT THE PROGRAM ***
Private Sub ABRTNM_Click()
IDLDrawWidget1.DoExit
```

```
End
End Sub

*** EXIT THE PROGRAM ***
Private Sub EXIT_Click()
IDLDrawWidget1.DoExit
End
End Sub
```

Appendix IV

(IDL Code for Observations)

```
*** MAIN IDL FUNCTION NEWOBSERV_VB() WHICH IS CALLED FROM VB ***

pro newobserv_vb,dat,noaa,mode,nof,tim,nf
***FUNCTION NEWOBSERV_VB*********************************************
SYNTAX : NEWOBSERV_VB,arg1,arg2,arg3,arg4,arg5,arg6
Arg1  =     Date of observation,
Arg2  =     NOAA Active Region No.,
Arg3  =     Observation Mode,
Arg4  =     No. of images to capture,
Arg5  =     Time Interval,
Arg6  =     No. of raw images from which best contrast image is computed
            for Normal Mode Observations,
*********************************************************************

*****Generate time and date information from system clock ************
*****    Using this date & time create a new directory    ************
tnd=strcompress(strmid(systime(),7,4)+strmid(systime(),3,4)+strmid(systi
me(),22,2)+'_'+strmid(systime(),11,2)+strmid(systime(),14,2)+strmid(syst
ime(),17,2),/remove_all)
dname=strcompress('Uso'+tnd+'_'+noaa,/remove_all)
creat_dir,dname,'C:\'+dat

*****Generate a LOG file & store observing parameters ************
get_lun,main_log_lun

openw,main_log_lun,strcompress('log'+noaa+'.txt')
close,main_log_lun
openu,main_log_lun,strcompress('log'+noaa+'.txt'),/append

printf,main_log_lun,"******* LOG FILE *******"
printf,main_log_lun,"TIME/DATE      : ", systime()
printf,main_log_lun,"Main Directory : ", dat
printf,main_log_lun,"Sub-directory  : ", dname

***Create TXT file & store names of NORMAL Mode ALL image files ****
get_lun,tlun
openw,tlun,'N_list_tiff.txt'


close,tlun
```

```
free_lun,tlun

***Create TXT file & store names of FLARE Mode image files *********
get_lun,flun
openw,flun,'F_list_tiff.txt'
close,flun
free_lun,flun


***Create TXT file & store names of NORMAL Mode BEST image files****
get_lun,fit_lun
openw,fit_lun,'list_fits.txt'
close,fit_lun
free_lun,fit_lun

******************** Check for observ mode flag*********************
if (mode eq 'n') then goto, NORMAL
if (mode eq 'f') then goto, FLARE
if (mode eq 'N') then goto, NORMAL
if (mode eq 'F') then goto, FLARE
printf,main_log_lun,'Mode of observation is :',mode

;***********************************************************
;********** START OF FLARE MODE OBSERVATION BLOCK ***********
;********** Calls Another IDL function Observ_flare_vb()****
FLARE:
    print,'FLARE mode observation Module'
      observ_flare_vb,main_log_lun,nof,tim
      goto, endl

;***********************************************************
;********* START OF NORMAL MODE OBSERVATION BLOCK ***********
;********** Calls Another IDL function Normal_observ_vb()****
NORMAL:
    print,'NORMAL mode observation Module'
    normal_observ_vb,main_log_lun,nof,tim,nf
    goto, endl

ENDL:

end
```

IDL SUB_FUNCTIONS CALLED FROM NEW_OBSERV_VB()

NORMAL_OBSERV_VB()

```
pro normal_observ_vb,loglun,nof,tim,nf
;+
;     PROGRAM         normal mode observing program
;
; PURPOSE    To observe in normal mode
;            Ask for observing parameters and write them to main
;            log file associated with LUN (loglun).
;            Get sequence of images using get_ccd_image() function
;            Find maximum contrast image using best_contrast() function
;            Write it to FITS format and the name of this fits file
;            is written to main log file.
;            In between it generates norm.txt files as list buffer
;-

*******Print function arguments in log file*******
printf,loglun,'Number of images : ',nof
printf,loglun,'Time interval (in sec) :',tim
printf,loglun,'No. of frames of selection:',nf
printf,loglun,'Sequence Start Time in UT:',systime()

get_lun,tiflst_lun
openu,tiflst_lun,'N_list_tiff.txt',/append

get_lun,fit_lun
openu,fit_lun,'list_fits.txt',/append

for lo1=1,nof do begin

      get_lun,tif_lun
      openw,tif_lun,'norm.txt'
```

```
********Call IDL Function get_ccd_image_normal() for image ******
******** capture                                          **********
            get_ccdimage_normal,nf,0,tif_lun,tiflst_lun
      close,tif_lun
      free_lun,tif_lun
********Call IDL Function best_contrast() for computing best ****
********    contrast image                                ******
            best_contrast,'norm.txt',nf,loglun,fit_lun
      wait,tim
print,'Best Frame No.:',lo1
endfor
print,'NORMAL MODE SEQUENCE DONE'
close,tiflst_lun
free_lun,tiflst_lun
close,fit_lun
free_lun,fit_lun
;_____________________________________________________
printf,loglun,'Sequence End Time in UT:',systime()
;_____________________________________________________
end
```

OBSERV_FLARE_VB()

```
pro observ_flare_vb,logunit,nof,tim
;+
;PROGRAM          OBSERV_FLARE
;SYNTAX           observ_flare, logunit
;PURPOSE          To observe in flare mode and (i) set all  parameters
;                 observation in this block and write log of this in
;                  logfile associated with LUN (logunit). Also put file
;                 names in logfile associated with LUN (logunit).
;REVISION         Written by Sanjay Gosain for automated island
;                  observation.

*******Print function arguments in log file*******
printf,logunit,'Number of frames : ',nof
printf,logunit,'Time interval (in sec) :',tim
```

```
printf,logunit,'Sequence Start Time in UT:',systime()
get_lun,fl_lun
openu,fl_lun,'F_list_tiff.txt',/append
********Call IDL Function get_ccdimage_flare() for image ******
******** capture                                          **********
get_ccdimage_flare,nof,tim,logunit,fl_lun
close,fl_lun
free_lun,fl_lun
printf,logunit,'Sequence End Time in UT:',systime()

end
```

***** *GET_CCDIMAGE_NORMAL()* WHICH IS CALLED FROM ********
******* IDL FUNCTION *NORMAL_OBSERV_VB()* *****

```
pro get_ccdimage_normal,nof,tim,log_lun,tiflst_lun
s1="
s2="
s4='islobs'
f1n="

for k=1,nof do begin
s1=systime()
;********Computer Time must be set in UT ************
s2=strcompress('U'+strmid(s1,11,2)+strmid(s1,14,2)+strmid(s1,17,2),/rem
ove_all)
printf,log_lun,s2
printf,tiflst_lun,s2
f1n=strcompress(s4+' '+s2)
spawn,f1n,/hide
wait,tim
print,'Normal Mode Frame No.:',k
img=read_tiff(s2)
tvscl,img
endfor
print,'NORMAL MODE SEQUENCE DONE'
end
```

***** *GET_CCDIMAGE_FLARE()* WHICH IS CALLED FROM *******
******* MAIN IDL FUNCTION *OBSERV_FLARE_VB()* *****

```
pro get_ccdimage_flare,nof,tim,log_lun,fl_lun

s1="
s2="
s4='islobs'
f1n="

for k=1,nof do begin
s1=systime()
;********Computer Time must be set in UT ************
s2=strcompress('U'+strmid(s1,11,2)+strmid(s1,14,2)+strmid(s1,17,2),/rem
ove_all)
printf,log_lun,s2
printf,fl_lun,s2
f1n=strcompress(s4+' '+s2)
spawn,f1n,/hide
wait,tim
print,'Flare Mode Frame No.:',k
img=read_tiff(s2)
tvscl,img,order=1
endfor
print,'FLARE MODE SEQUENCE DONE'
end
```

***** IDL FUNCTION *BEST_CONTRAST()* WHICH IS CALLED FROM ********

******* MAIN IDL FUNCTION *NORMAL_OBSERV_VB()* *****

```
pro best_contrast,listfile,nf,lunlog,lunfits
;+
;PROGRAM     best_contrast
;Purpose     To find the best frame from a sequence of TIFF files read
from CCD and write that frame as fits file
;Inputs
;     LISTFILE    Name of file containing the sequence of TIFF files
;     NF          Number of TIFF files in LISTFILE
;     LUNFITS     LUN for the file keeping the FITS file names
;
; Written by Sanjay Gosain on 16th May 2002 for SPAR AUTOMATIC
; OBSERVING PROGRAM
;-

nx=752
ny=480
fn="

openr,u,listfile,/get_lun

names=strarr(nf)
con=fltarr(nf)
imarr=fltarr(nx,ny,nf)
smarr=fltarr(nx,ny)

      k=0
repeat begin
      readf,u,fn
      names(k)=fn
      img=read_tiff(fn)
      imarr(*,*,k)=float(img)
      k=k+1
endrep until eof(u)
      close,u
      free_lun,u
```

```
for it=0,nf-1 do begin
smarr(*,*)=imarr(*,*,it)
sum=0.0
      for k1=0,nx-4 do begin
      for k2=0,ny-4 do begin
      sum=sum+(smarr(k1+3,k2)-smarr(k1,k2))^2 + (smarr(k1,k2+3)-
smarr(k1,k2))^2
      endfor
      endfor
dm=total(smarr*smarr)/float(n_elements(smarr))
con(it)=sum/dm
endfor

maxc=max(con,mc)

writefits,strcompress(names(mc)+'.fits',/remove_all),byte(imarr(*,*,mc)
)
printf,lunlog,names(mc)+'.fits'
printf,lunfits,names(mc)+'.fits'

end
```

Appendix-V

C-Language Code For Reading Digitized Image from Camera Buffer

ISLOBS.C

(This code makes use of *read_pic( )* function, which does data readout by parallel port. The program below is compiled to create Object file and then linked with the vendor supplied Object files)

```
#include <bios.h>
#include <conio.h>
#include <ctype.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dos.h>
#include <time.h>
#include <sys\timeb.h>
#include "cbios.h"

/* Declarations of functions*/

int read_pic(char *filename, int fmt, int width, int height, int hoffs,
int voffs, int field, int prn);

main(int argc, char *argv[])
{
      int width,height,hoff,voff,i,it;
      struct timeb t,tb,tb1;
      char da[100],s1[30];
      FILE *f1,*f2;

/* Check for argument */
      if (argc != 2) {
                printf("Needs filename");
                exit(1);
                }

/*call read_pic() to capture image*/

      read_pic(argv[1],1,752,480,0,0,0,1);

return 0;

}
```

References:

1. Interactive Data Language (IDL) 5.6 Manuals from Research Systems Inc., USA.

2. Online material from www.vbtutor.com

3. Online material from www.rsinc.com

4. Search engine www.google.com