

Michaela Logue
Dr. Michael Castelaz
PARI
July 25, 2003

Polaris Monitoring Project

Polaris, the brightest star of the constellation Ursa Minoris is located at the tip of the handle of the Little Dipper. As the brightest of all Cepheid Variables, many astronomers have closely monitored Polaris in the past. Beginning after 1945, such astronomers noticed a rate of increase in Polaris' period as well as a rate of decrease in its amplitude. It was then suggested in 1993 that Polaris would cease all pulsations within the next few years. Much to everyone's surprise, in 1998 astronomers noticed that Polaris did not cease pulsating; furthermore, its amplitude decrease also stopped. With such unpredictable behavior, much is still left to be discovered about Polaris. The amplitude of Polaris' variations is about 0.03 mag, and its period is 3.97 days. Some astronomers suspect that we have been witnessing the evolution of Polaris as a Cepheid Variable: that this may be the process by which a Cepheid Variable evolves out of pulsating in its first overtone to pulsating with its fundamental period - resulting in a more stable and stereotypical Cepheid Variable. The goals of this Project are to monitor Polaris 24 hours a day, 7 days a week, providing live images of Polaris and a data base for continuous light curves of Polaris.

Polaris will be monitored with a 12-inch LX200 Meade telescope and a SBIG STV CCD. A script written in Microsoft Visual Basic 6.0 is used to control and coordinate telescope movements with CCD imaging. The Polaris Monitoring Program allows the user to input the number of images to stack, the exposure time, the interval between stacks, the coordinates for a check star, and the number of Polaris stacks

between check star stacks. Currently, star number 286 of the Bright Star Catalogue, with a magnitude of 6.46, is set as the default check star. The Polaris telescope is located in the North Star building of the optical ridge at PARI. The building is equipped with a 24 inch X 26 inch window and is oriented due north. With the use another Visual Basic script the images are analyzed, resulting in a continuous light curve of Polaris.

Development of the Visual Basic script occurred in three stages: STV control, telescope control, and automated photometry.

Understanding the Code:

Originally, this project was to use STV Control, an SBIG software packaged designed to remotely control the STV CCD. However, complications arose when the limitations of STV Control affected the project plans. It was mandated that the camera would take a set of images, a user input number of images to stack, and then wait an user input interval before taking another set of images. The solution to this complication was to adapt the Solar/Lunar CCD's visual basic program and restructure it to suit the purposes of the Polaris CCD. Since the Solar/Lunar project also utilized a SBIG STV CCD, the STV control module was left unabridged. However, the command buttons and menu features had to be redone. The end result allowed the user to input a few variables and simply press the start button.

The form on the left is the original windows program for the Solar/Lunar project. The form on the right is the windows program adapted from the Solar/Lunar project for the Polaris project. Initially, the program was only to be used for CCD control.



The code for the start button is as follows:

```

Private Sub cmdStart_Click()

Dim PauseTime, Start, Finish, TotalTime

Do While cmdStop.Enabled = True

For ncount = 1 To ImageStack

    lblStack.Text = Str(ncount)
    JDy = Cdbl(Format(Date, "yyyy"))
    JDD = Cdbl(Format(Date, "y"))
    JulianDate = 2451544 + (JDy - 2000) * 365.25 + JDD + (Timer / 3600 + 5 -
        DLS) / 24
    PolName = "c:/Polaris/P" + Str(JulianDate)
    Open PolName For Output As #31
    Call LetsGo
    Close #31

Next ncount

PauseTime = IntervalTime ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
    DoEvents ' Yield to other processes.

```

```
Loop
Finish = Timer ' Set end time.
TotalTime = Finish - Start ' Calculate total time.
MsgBox "Paused for " & TotalTime & " seconds"
```

```
Loop

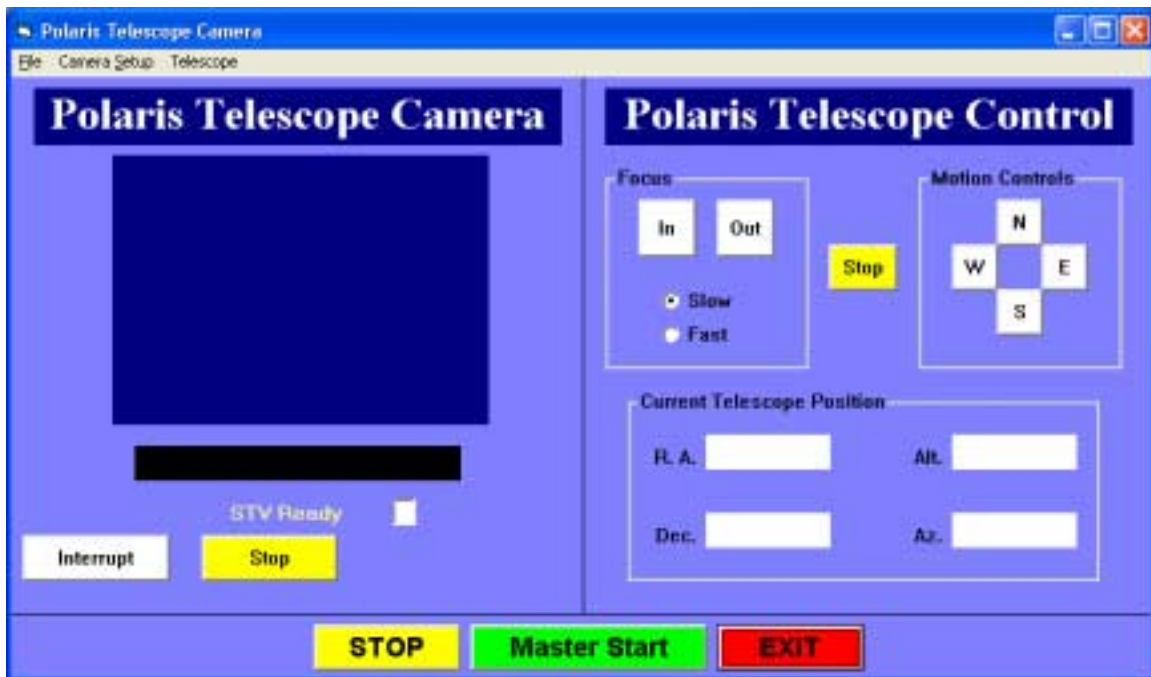
cmdStop.Enabled = True

End Sub
```

The Stop button controls the main loop. While the Stop button is enabled the CCD will continually take a set of images and then wait for a specific interval time. The inner “for” loop causes the camera to take images back-to-back for stacking; these images are then saved to a file called “Polaris” on the computer’s hard drive, with the name “P”+ the Julian Date. In this code, the interval time is the variable IntervalTime and the number of images to stack is the variable ImageStack. Both these variables are defined with user input values. Furthermore, the user input data is located under the Setup menu title; there the user may choose whatever interval time, exposure length, and number of images to stack they wish to have.

With the completion of the STV controls, the next item to be programmed was the telescope movement. Since this project is monitoring the intensity of Polaris, a method had to be developed capable of measuring the magnitude with an accuracy of 0.005 mag. It was decided that the telescope would need to point at Polaris most of the night and day, but slew to a star with a known magnitude every hour or so. Initially it was thought that TheSky software would be used to handle the telescope movements, the predicament which evolved dealt with choreographing the CCD image taking and the telescope movements. Although it is factual to assume that there is a set time between image sets, the rate at which the pictures downloaded into the computer add extra time to the total

loop time. Thus, TheSky software was abandoned and it was decided that code could be added to the STV program that would control our Meade telescope simultaneously with the CCD. Since the Meade users guide included serial port commands for controlling the telescope, it was not too difficult to write the code for the basic commands. After all the necessary buttons were added, the new Polaris program appeared as follows:



The drop down menu items under telescope include: connecting to the telescope, slewing, syncing, setting the slew rate, and setting the check star intervals. Under the slewing command, the user could slew to Polaris or to a check star. The master start button was created and developed a loop in which the camera would take a predetermined number of image sets of Polaris and then slew to the check star where it would take one image set before slewing back to Polaris and repeating the loop. The code for the Master Star is as follows:

```

Private Sub cmdMasterStart_Click()

Dim cat As Integer

Do While cmdMasterStop.Enabled = True

For cat = 1 To CheckStarTime

Call CamStart

Next cat

ComPort.Output = "Sr 01:09.4#"
ComPort.Output = "Sd +86*16#"
ComPort.Output = ":MS#"

PauseTime = 5 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
    DoEvents ' Yield to other processes.
Loop
Finish = Timer ' Set end time.
TotalTime = Finish - Start ' Calculate total time.
'MsgBox "Paused for " & TotalTime & " seconds"

Call CamStart

ComPort.Output = ":LS 19#"
ComPort.Output = ":MS#"

PauseTime = 5 ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
    DoEvents ' Yield to other processes.
Loop
Finish = Timer ' Set end time.
TotalTime = Finish - Start ' Calculate total time.
'MsgBox "Paused for " & TotalTime & " seconds"

Loop

cmdMasterStop.Enabled = True

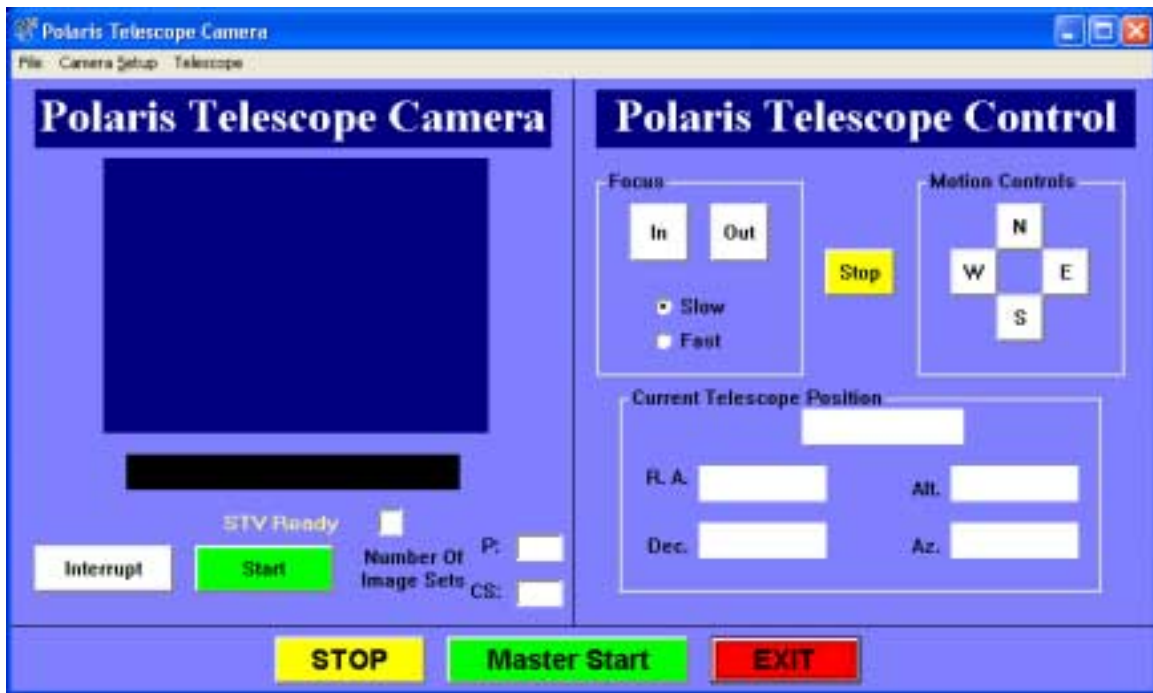
End Sub

```

Similarly as before, the larger Stop button controls the outermost loop, thus it is possible for the loop to continue indefinitely. The inner loop controls the number of image sets of

Polaris which are to be taken. Once outside of that loop, the telescope slews to a predetermined check star, calls on the CamStart function (the previous Start Button), takes an image set of the check star, and then slews back to Polaris before starting the loop again.

While this code worked well, it was modified to allow the user to input the coordinates for the check star and it placed the slew wait in the code of the CamStart function. Furthermore, code has been added which controls the filter color of the CCD. The code still needs to have the filter wheel color changes automated in the Mater Start function. Currently, the Polaris program appears as follows:



This current form also displays the number of image sets of Polaris and the Check Star which have been acquired. The Master Start subroutine was also cleaned up a bit, resulting in the following code:

```
Private Sub cmdMasterStart_Click()
```

```

Dim cat As Integer

Do While cmdMasterStop.Enabled = True

' First we let the camera take a given number of sets of
'   stacked images

    For cat = 1 To CheckStarTime

        Call cmdCamStart_Click

    Next cat

' Now we slew the telescope over to check stars
' Also, the current coordinates gets updated
' And there's a 10 second wait for the slew

    Call mnuCSSlew_Click

' Now we take an image set

    Call cmdCamStart_Click

' Now we slew back to Polaris, update the coordinates

    Call mnuPolSlew_Click

Loop

cmdMasterStop.Enabled = True

End Sub

```

The CSSlew and the PolSlew subroutines now contain code allowing the user to input coordinates for the check star, update the current location caption and coordinates.

With the program successfully orchestrating the CCD and the telescope, the last remaining part was the photometry. Originally MIRA 6.0 was the intended image analysis package. With its batch processing tools, it was thought that MIRA would be able to easily analyze the images and yield the magnitudes. However, another method was eventually decided upon. The Bright Star program already had Visual Basic code written which took an image, found the largest star, and then found the average pixel

value of the largest star. Based on the previous images of Polaris, we knew that either the image will only contain Polaris, at a low enough exposure, or it would depict Polaris and its companion star, an eighth magnitude star. In the latter case, the assuming that the images are focused properly, it is expected that the program will differentiate between the two stars and only record Polaris's average pixel value. The PicVal() function was copied from the Bright Star Program and pasted into the Polaris program. It was then decided to utilize this function in the CamStart() function. Then code was also written to save the average pixel values, stack the images, and record magnitudes based on the check star images. The code appears as follows:

```
Private Sub cmdCamStart_Click()

Dim ncount As Integer
Dim PauseTime, Start, Finish, TotalTime, JDy, JDd, JulianDate, PolName, DLS
Dim Brightness As Long

'The loop is driven in the master start sub routine, so it's been taken out
' of here.

Do While cmdStop.Enabled = True

First the camera takes the specified number of images to stack
Open "c:/Polaris/Deltamag.dat" For Output As #22

For ncount = 1 To ImageStack

    lblStack.Text = Str(ncount)
    JDy = CDBl(Format(Date, "yyyy"))
    JDd = CDBl(Format(Date, "y"))
    JulianDate = 2451544 + (JDy - 2000) * 365.25 + JDd + (Timer / 3600 + 5 - DLS) / 24
    FileCount = FileCount + 1
    PolName = "c:/Polaris/P" + Str(FileCount)
    Open PolName For Output As #31
    Call LetsGo
    Close #31

    Call PicVal

    StackArray(ncount) = StarTrans

Next ncount

Now we update the counts
```

```

If lblCurrent.Caption = "Polaris" Then

    Polcount = Polcount + 1

    txtPol.Text = Polcount

End If

If lblCurrent.Caption = "Micki's Star" Then

    CScout = CScout + 1

    txtCS.Text = CScout

End If

Polnumber = 0
CSnumber = 0

For i = 1 To ImageStack Step 1
    If lblCurrent.Caption = "Polaris" Then
        Polnumber = Polnumber + StackArray(i)
    End If

    If lblCurrent.Caption = "Micki's Star" Then
        CSnumber = CSnumber + StackArray(i)
    End If

Next

If lblCurrent.Caption = "Micki's Star" Then

    For i = Polcount - (CheckStarTime - 1) To Polcount Step 1
        PolMag = CSMag + 2.5 * (Log(CSnumber / PolRawArray(i))) / Log(10)
        PolMagArray(i) = Str(PolMag) + " " + Str(JulianDate)
        Print #22, PolMagArray(i)
    Next i

End If

If lblCurrent.Caption = "Polaris" Then
    PolRawArray(Polcount) = Polnumber
End If

'Wait the specified interval time

PauseTime = IntervalTime ' Set duration.
Start = Timer ' Set start time.
Do While Timer < Start + PauseTime
    DoEvents ' Yield to other processes.
Loop
Finish = Timer ' Set end time.
TotalTime = Finish - Start ' Calculate total time.

```

```
        MsgBox "Paused for " & TotalTime & " seconds"  
    Loop  
    'cmdStop.Enabled = True  
End Sub
```

When the CCD takes an image, the PicVal function is called and the absolute average of the star is saved in an array (StackArray). Now, if the image set was of Polaris, then the values in the stack array get summed and stored in the PolArray, with an index based on the Polcount. However, if the image set was of the check star, then the values in the StackArray get summed and stored as the value CSnumber. This number is then used to calculate the magnitudes of the previous Polaris image sets. Each time a magnitude is calculated, the data is sent to the hard drive and saved along with the current Julian Date.

Still to Do:

Code:

Although the code does determine the magnitude of the images, it performs this operation on raw images. There still needs to be a Dark and a Flat image taken and saved. The raw images must first have the Dark image subtracted and be divided by the Flat image before the magnitudes are determined. I would suggest a separate function to be created which could perform the dark subtraction and flat division. This function could then be called upon before the PicVal function. Hence, the magnitudes will be determined based upon clean images.

Although the code for the color filters on the filter wheel has been written, it has not yet been tested. It was observed that the sequence of button presses on the Polaris STV and the Solar/Lunar STV are different, and thus the program must be tested on the actual Polaris STV.

Telescope:

The Meade LX200 twelve-inch telescope has had technical malfunctions beginning July 14, 2003. Upon powering up the telescope, the hand paddle reads “Meade” and never progresses to the “LX200” screen. After switching out all the parts of the twelve-inch Meade with the Furman ten-inch Meade, aside from the RA motor, and still not having any success in starting up the Polaris telescope, it is still undetermined as to why the telescope has failed or how to fix the problem.

Website:

Finally, the Polaris website must also be revised as there are grammatical errors in the Monitoring Project site. Also, in time it was intended to add Native American stories to the History site.