

**INAF-Osservatorio astronomico di Torino**

*Technical Report nr.150*

Valutazione degli effetti della  
nuvolosità sui dati acquisiti  
durante l'eclisse totale di sole  
dell'11 luglio 2010

G. Capobianco

*Pino Torinese, 27 maggio 2011*

**INAF-Osservatorio astronomico di Torino**

*Technical Report nr. 150*

Valutazione degli effetti della nuvolosità sui  
dati acquisiti durante l' eclisse totale di sole  
dell' 11 luglio 2010

G. Capobianco

*Pino Torinese, 27 maggio 2011*



# ECLIPSE CLOUDS EVALUATION

---

**Gerardo Capobianco**

**Vers. 0**

**Rev. 0**

**Date: 13 gennaio 2011**



## **Sommario**

Introduzione .....	3
Selezione delle immagini .....	4
Codice sorgente del software di analisi.....	7

## **Introduzione**

Scopo di questo documento è l'identificazione dei frames acquisiti con i due strumenti dell'OATo durante la totalità dell'eclisse di Sole dell'11 luglio 2010, che risentono in modo limitato o assente della presenza delle nubi che hanno coperto il cielo per parte dell'evento e spiegare i criteri con cui questa è stata fatta. Uno studio quantitativo risulta essere molto difficoltoso, perché gli effetti delle nubi sono diversificati, in particolare, le nubi bianche diffondono la luce coronale, mentre quelle scure ne abbassano l'intensità. Inoltre, i tempi di esposizione del CorMag sono abbastanza lunghi (8 secondi) da presentare una combinazione dei due effetti. Si è fatto, quindi, solo uno studio qualitativo, identificando i frames che risultano essere poco influenzati dagli effetti delle nuvole. Nel caso del polarimetro E-KPol la situazione è

resa ancora più difficoltosa dal fatto che la luce diffusa è polarizzata, quindi ciò ha evidenti effetti sulle misure. Circa il 50% delle misure effettuate risulta “da scartare” da questa analisi.

### Selezione delle immagini

Per eliminare gli effetti di polarizzazione nella selezione delle immagini con nubi (entrambi gli strumenti OATo fanno misure polarimetriche), si sono utilizzati i dati acquisiti con il VLC#1 dell’Università di Firenze. Questo strumento è costituito da un telescopio Borg 76ED (500mm focal length aperture f/6.5 w/ G filter (Baader)) ed una camera CCD FLI MicroLine 8300 (3326x2504 pixel). Per ogni immagine acquisita, si sottrae il dark, vengono eliminati i pixel saturi e calcolata la media su tutto il frame (eccetto i pixel saturi), normalizzata poi per pixel e per secondo di esposizione. Accanto a questo valore, viene calcolata anche la media su un quadratino (40x40 pixel) del disco “nero” della luna. Questi valori vengono poi espressi in funzione del tempo a cui è stata acquisita l’immagine. Collocando in questo plot dove temporalmente si trovano i frames acquisiti con i due strumenti OATo, è possibile scegliere quelli “buoni”, ovvero quelli che presentano una scarsa presenza di nubi. Le immagini scartate sono quelle che presentano valori di intensità normalizzata sensibilmente maggiore o minore di quella media. In particolare, il criterio di selezione delle immagini prevede le seguenti soglie per i conteggi normalizzati su tutto il frame:

Max: 28000 ADU;

Min: 9550 ADU.

Inoltre la variazione dell’intensità media con quella sulla Luna deve essere inferiore al 37%.

I frames che soddisfano questo criterio sono evidenziati in ciano nei plot di Figura 2 e 3. I frames selezionati del CorMag sono: 2, 3, 4, 5, 6, 7, 8, 9, (12), (13), 17, 18, 19, 20, 21, 26, (27). I frames riportati tra parentesi presentano delle nubi, ma possono essere parzialmente utilizzati.

Mentre per E-KPol: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 22, 23, 24, 32, 33, 34, 35, 36, 37, 38, 47, 48, 49.

In Figura 1 si riporta un esempio delle aree utilizzate di un frame per il calcolo delle medie. L’area azzurra è quella esclusa dal calcolo della media su tutto il frame perché i pixel sono saturi, mentre il quadrato verde, indica l’area utilizzata per calcolare la media sulla Luna.

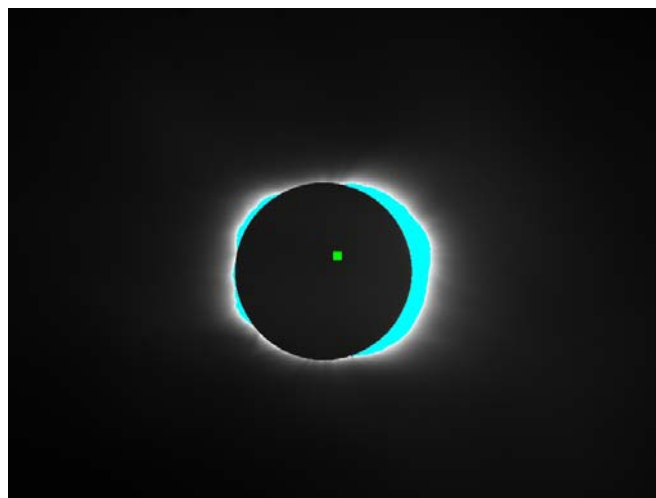


Figura 1 – Esempio di aree selezionate per il calcolo delle medie (in azzurro la parte esclusa dal calcolo della media su tutto il frame ed in verde l’area utilizzata per il calcolo della media sulla Luna).

Nel caso di CorMag, i parametri di acquisizione sono riportati sul grafico di figura 2. Per quanto riguarda lo scan in lunghezza d'onda, si hanno le seguenti lunghezze d'onda (tra parentesi il numero sequenziale del frame ed in grassetto le immagini che possono essere utilizzate per certi scopi, ma che presentano un certo grado di effetti delle nubi):

- (2) 5301 Å
- (3) 5301.4 Å
- (4) 5301.8 Å
- (5) 5302.2 Å
- (6) 5302.6 Å
- (7,21) 5303 Å
- (8) 5303.4 Å
- (9) 5303.8 Å
- (27) 5304.2 Å**
- (12) 5305 Å**
- (13) 5305.4 Å**

Per quanto riguarda le immagini da utilizzare per la polarimetria, si possono utilizzare i seguenti per le misure in banda:

- (21,7) 5303 Å -> 0 deg
- (20) 5303 Å -> 60 deg
- (19) 5303 Å -> 120 deg

Per quelle fuori-banda si possono utilizzare misure a diverse lunghezze d'onda:

- (27) 5304.2 Å -> 0 deg**
- (9) 5303.8 Å -> 0 deg
- (17) 5308 Å -> 60 deg
- (26) 5304.2 Å -> 60 deg
- (18) 5308 Å -> 120 deg

La lista dei nomi dei files che soddisfano il criterio sopra citato è la seguente:

- CorMagImage\_2010.07.11.18.45.50.fits (2);
- CorMagImage\_2010.07.11.18.46.00.fits (3);
- CorMagImage\_2010.07.11.18.46.09.fits (4);
- CorMagImage\_2010.07.11.18.46.19.fits (5);
- CorMagImage\_2010.07.11.18.46.28.fits (6);
- CorMagImage\_2010.07.11.18.46.38.fits (7);
- CorMagImage\_2010.07.11.18.46.47.fits (8);
- CorMagImage\_2010.07.11.18.46.56.fits (9);
- CorMagImage\_2010.07.11.18.47.25.fits (12);
- CorMagImage\_2010.07.11.18.47.34.fits (13);

CorMagImage\_2010.07.11.18.48.12.fits (17);  
 CorMagImage\_2010.07.11.18.48.21.fits (18);  
 CorMagImage\_2010.07.11.18.48.31.fits (19);  
 CorMagImage\_2010.07.11.18.48.40.fits (20);  
 CorMagImage\_2010.07.11.18.48.50.fits (21);  
 CorMagImage\_2010.07.11.18.49.37.fits (26);  
 CorMagImage\_2010.07.11.18.49.46.fits (27).

## VLC#1 Intensity plot

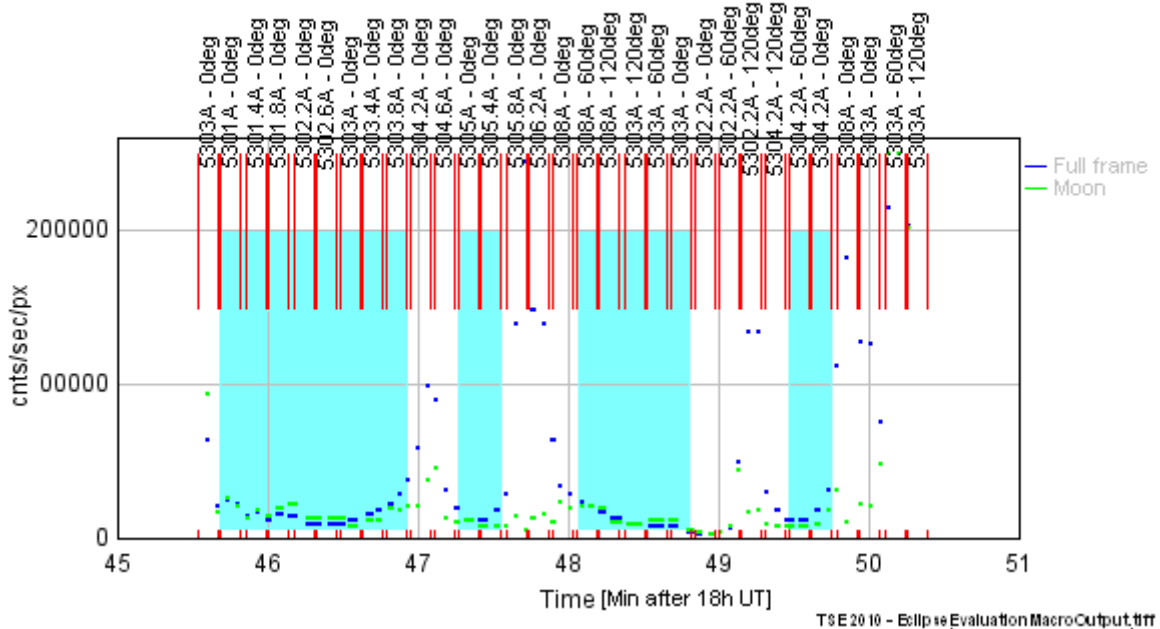


Figura 2 – Selezione dei dati del CorMag

Nel caso di E-KPol, invece, si ha:

4.5V -> 3 frames con exp time di 250 ms (9Image20100711184624.fits; 33Image20100711184821.fits; 49Image20100711184940.fits) e 3 con exp time di 1 sec (5Image20100711184603.fits; 13Image20100711184643.fits; 37Image20100711184840.fits);

5.4V -> 2 frames con exp time di 250 ms (10Image20100711184628.fits; 34Image20100711184825.fits) e 4 con exp time di 1 sec (6Image20100711184608.fits; 14Image20100711184647.fits; 22Image20100711184727.fits; 38Image20100711184845.fits);

7V -> 2 frames con exp time di 250 ms (11Image20100711184632.fits; 35Image20100711184830.fits) e 4 con exp time di 1 sec (7Image20100711184613.fits; 15Image20100711184652.fits; 23Image20100711184731.fits; 47Image20100711184929.fits);

10V -> 2 frames con exp time di 250 ms (12Image20100711184636.fits; 36Image20100711184834.fits) e 4 con exp time di 1 sec (8Image20100711184618.fits; 24Image20100711184736.fits; 32Image20100711184816.fits; 48Image20100711184934.fits).

Si hanno quindi a disposizione, almeno 2 sequenze utili.

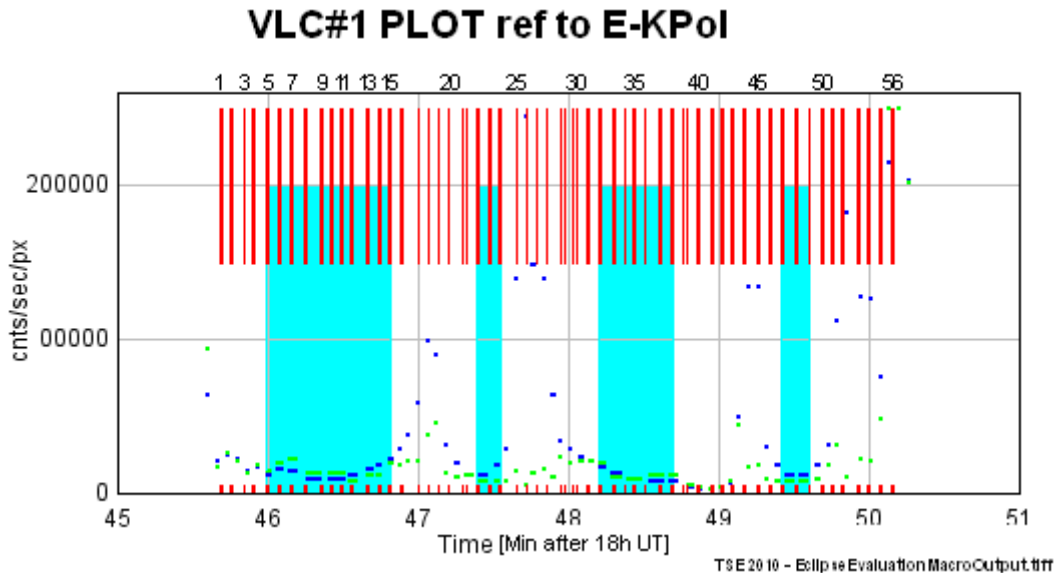


Figura 3 – Selezione dei dati dell'E-KPol

## Codice sorgente del software di analisi

L'analisi è stata fatta utilizzando una macro per ImageJ riportata sotto. Scelta la directory dei dati di VLC#1 e quella dei dark, il programma cerca il dark associato ad ogni frame, lo sottrae e calcola i valori medi di intensità su questo frame (vedi paragrafo precedente). Completata questa operazione chiede se si vogliono overplottare i dati di CorMag o di E-KPol e la directory in cui questi sono contenuti. Il programma visualizza i plot come quelli di Figura2 e 3 che è poi possibile salvare nel formato desiderato.

### Macro

```
//*****
// EclipseCloudsEvaluationMacro[ECEM]
// Author: G. Capobianco INAF-OATo
// capobianco@oato.inaf.it
// Date: 18 dec 2010
// Software version: 0.1
// Software release: 2
// License: GNU GPL 3
// Comments:
// Copyright: © 2010 G. Capobianco INAF-OATo
//
//*****
//
// Questa macro legge i files contenuti nella dir scelta, controlla che siano tutti
fits,
// associa ad ogni frame il rispettivo dark, scelto tra quelli contenuti nella
directory scelta,
// Quindi apre il file ed il corrispondente dark e lo sottrae. Sull'immagine calibrata
```



```

// riporta la media dei pixels non saturi normalizzati al secondo. Plotta i conteggi in
// funziona del tempo a cui sono stati acquisiti. A questo plot aggiunge la posizione
// dei
// frame del CorMag (o dell'E-KPol) e il plot della media di 1600 pixels sul centro
// della luna.
//
sep=File.separator;
    //Multiplatform
datapath = getDirectory("Choose Data Directory");
    //Select Data Directory
darkpath = getDirectory("Choose Dark Data Directory");
    //Select Dark Data Directory
datalist = getFileList(datapath);
    //List of data files
darklist = getFileList(darkpath);
    //List of Dark files
DataFiles=lengthOf(datalist);
    //Number of files and folders in data directory
DarkFiles=lengthOf(darklist);
    //Number of files and folders in dark directory
//
//print ("EclipseCloudsEvaluationMacro[ECEM] Report.txt");
    //PRINT
//print ("The data folder "+ datapath + " contains "+ DataFiles + " files and
//folders"); //PRINT
//print ("The dark data folder "+ darkpath + " contains "+ DarkFiles + " files and
//folders"); //PRINT
//
//$$$$ Scan for fits files in data folder $$$$
//
a=0; //Number
of data files fits standard compatible
for (i=0; i<DataFiles;i++){
    fitstype=CheckFits(datalist[i]);
    //Call the CheckFits function
    if (fitstype==1){a++;}
}
fitsfilesnum=newArray(a);
j=0;
for (k=0; k<DataFiles;k++){
    fitstype=CheckFits(datalist[k]);
    //Call the CheckFits function
    if (fitstype==1){
        fitsfilesnum[j]=datalist[k];
        j++;
    }
}
//$$$$ Scan for fits files in dark data folder $$$$
//
b=0; //Number
of dark files fits standard compatible
for (i=0; i<DarkFiles;i++){
    fitstype=CheckFits(darklist[i]);
    //Call the CheckFits function
    if (fitstype==1){b++;}
}
darkfitsfilesnum=newArray(b);
j=0;
for (k=0; k<DarkFiles;k++){

```



```

sum=0; //Sum of
all not saturated pixels
thresholdmax=60000;
thresholdmin=300;

for(k=0;k<getWidth;k++){
  for(j=0;j<getHeight;j++){
//      Overlay.drawLine(0,j,k,j);
//      Overlay.show;
    if (getPixel(k,j)<thresholdmax && getPixel(k,j)>thresholdmin){
      sum=sum+ getPixel(k,j);
      pixnotsat++;
    }
  }
}
Xc=1598; //X coord
of the center of the moon
Yc=1328; //Y coord
of the center of the moon
sumcent=0;
for (h=Xc-20;h<Xc+20;h++){
  for (u=Yc-20;u<Yc+20;u++){
    sumcent=sumcent+getPixel(h,u);
  }
}
if(imghead[6]==0){
  imghead[6]= darkimg[3];
}
if(pixnotsat==0){
  avg[i]=65000/imghead[6];
}
else{
  avg[i]=(sum/pixnotsat)/imghead[6];
}
countscnt[i]=((sumcent/1600)/imghead[6]);
if(countscnt[i]>250000){countscnt[i]=250000;}
selectImage(id);
close();
selectImage(iddark);
close();
selectImage(idcal);
close();
q0=parseInt(substring(imghead[5],0,2));
q1=parseInt(substring(imghead[5],3,5));
q2=parseInt(substring(imghead[5],6,8));
timestart[i]=q1+q2/60;
timeclose[i]=timestart[i]+(imghead[6]/60);
print(i+1 + "      "+ fitsfilesnum[i]+"      "+ avg[i]+" with exposure time
of"+imghead[6]+" seconds, acquired at: "+imghead[5]+"      "+timestart[i]+" Number of not
saturated pixels: "+pixnotsat+" Average counts over the moon:"+countscnt[i]);
}
Array.getStatistics(timestart,Tmin,Tmax,Tm,Tsd);
Array.getStatistics(avg,Cmin,Cmax,Cm,Csd);
//print(Tmin+"      "+Tmax+"      "+ Cmin+"      "+Cmax);
Plot.create("Intensity vs Time plot", "Time","cnts/sec/px");
Plot.setLimits(45,51,0,260000);
Plot.setLineWidth(2);
//Plot.setColor("green");
//Plot.add("circles",timestart,avg);
//Plot.add("circles",timeclose,avg);

```

```

for(i=0;i<a;i++){
    Plot.setColor("blue");
    Plot.drawLine(timestart[i],avg[i],timeclose[i],avg[i]);
}
//Plot.setColor("orange");
//Plot.add("boxes",timestart,countscnt);
//Plot.add("boxes",timeclose,countscnt);
for(i=0;i<a;i++){
    Plot.setColor("green");
    Plot.drawLine(timestart[i],countscnt[i],timeclose[i],countscnt[i]);
}
//Plot.show();
//
//
Dialog.create("Choose Instrument");
Dialog.addCheckbox("Cormag[T]/EKPol[F]",1);
Dialog.show();
inst=Dialog.getCheckbox();
//
//CORMAG or EKPol DATA
cmdatapath = getDirectory("Choose CorMag Data Directory");
    //Select CorMag Data Directory
cmdatalist = getFileList(cmdatapath);
    //List of data files
cmDataFiles=lengthOf(cmdatalist);
    //Number of files and folders in data directory
//
//$$$$ Scan for fits files in data folder $$$$
//
c=0; //Number
of data files fits standard compatible
for (i=0; i<cmDataFiles;i++){
    fitstype=CheckFits(cmdatalist[i]);
    //Call the CheckFits function
    if (fitstype==1){c++;}
}
cmfitsfilesnum=newArray(c);
j=0;
for (k=0; k<cmDataFiles;k++){
    fitstype=CheckFits(cmdatalist[k]);
    //Call the CheckFits function
    if (fitstype==1){
        cmfitsfilesnum[j]=cmdatalist[k];
        j++;
    }
}
//$$$ Return the start and shutter closed timed
TOpenShut=newArray(c);
TCloseShut=newArray(c)
for (i=0;i<c;i++){
    CMfilenamecompl=cmdatapath+cmfitsfilesnum[i];
    datetimes=ReadCMStartAndEndTime(CMfilenamecompl,inst);
    TOpenShut[i]=datetimes[0];
    TCloseShut[i]=datetimes[1];
    if (inst==1){str="CorMagFrame";}
    else{str="E-KPolFrame";}
    print(str + i + " TOpenShutter= "+TOpenShut[i]+" TCloseShutter= "+TCloseShut[i]);
}
for(i=0;i<c;i++){
    Plot.setLineWidth(1);

```

```

Plot.setColor("red");
Plot.drawLine(TOpenShut[i],150000,TOpenShut[i],250000);
Plot.drawLine(TCloseShut[i],150000,TCloseShut[i],250000);
Plot.drawLine(TOpenShut[i], 0,TOpenShut[i],5000);
Plot.drawLine(TCloseShut[i], 0,TCloseShut[i],5000);
}
Plot.show();
//saveAs ("jpg");
print("The macro is closed");
//saveAs ("txt");
//
//
//*****FUNCTIONS USED IN THE MAIN PROGRAM*****
//
//=====
// Function name: CheckFits
// Input: filename;
// Output: 1 the filename if fits file 0 if different extension
// Created: 2010 oct 28
// Comments: Check if the file extension is fits compatible
//=====
function CheckFits (filename){

fits=endsWith(filename, ".fits")||endsWith(filename, ".fit")||endsWith(filename, ".fts");
return fits;
}
//
//=====
// Function name: ReadHeader
// Input: filename;
// Output: array of data file info
// Created: 2010 oct 28
// Comments:
//=====
function ReadHeader (filename){
par=newArray(8);
imgString=File.openAsString(filename);
indexN1=indexOf(imgString,"NAXIS1");
indexN2=indexOf(imgString,"NAXIS2");
indexBSC=indexOf(imgString,"BSCALE");
indexBZE=indexOf(imgString,"BZERO  =");
indexTime=indexOf(imgString,"DATE-OBS");
indexExpT=indexOf(imgString,"EXPTIME");
indexImgT=indexOf(imgString,"IMAGETYP");
// print(filename);
N1=substring(imgString,indexN1+26,indexN1+31);
// print(N1);
N2=substring(imgString,indexN2+26,indexN2+31);
// print(N2);
BSC=substring(imgString,indexBSC+12,indexBSC+31);
// print(BSC);
B0=substring(imgString,indexBZE+12,indexBZE+31);
// print(B0);
Time=substring(imgString,indexTime+22,indexTime+30);
// print(Time);
ET=parseFloat(substring(imgString,indexExpT+10,indexExpT+31));
// print(ET);
IT=substring(imgString,indexImgT+11,indexImgT+25);
// print(IT);
par[0]=filename;

```

```

par[1]=N1;
par[2]=N2;
par[3]=BSC;
par[4]=B0;
par[5]=Time;
par[6]=ET;
par[7]=IT;
return par;
}
//
//=====
// Function name: SearchDark
// Input: filename, headerdatafilearray,darkpath, darkfileslist;
// Output: array with filename, BZERO, BSCALE and exposure time
// Created: 2010 oct 28,29
// Comments: Search the dark frame related to the data file
//=====
function SearchDark (datafilename,headerdatafilearray, darkpath, darkfileslist){
    arrayout=newArray(4);
    exptimedatafile=headerdatafilearray[6];
    darkexptime=-1;
    i=0;
    b=lengthOf(darkfileslist);
    while (i<=b && darkexptime != exptimedatafile){
        if (i==b){
            //The case i=b is the
            case that all the dark frames have different exp time
            Dialog.create("Select the dark file");
            Dialog.addMessage("No dark file has been found for the file: " + datafilename+ "
Please entry your choice");
            Dialog.addChoice("dark file", darkfileslist);
            Dialog.show();
            darkfilename=darkpath+ Dialog.getChoice();
        }
        else{
            darkfilename = darkpath+darkfileslist[i];
        }
        //print (darkfilename);
        darkhead = ReadHeader (darkfilename);
        darkexptime= darkhead[6];
        darkframetype=darkhead[7];
        indexdarkcom=indexOf(darkframetype,"Dark Frame");
        //print(darkexptime + " resp to " + exptimedatafile);
        if (indexdarkcom<0){darkexptime=-1;}
        i++;
    }
    arrayout[0]=darkfilename;
    arrayout[1]=darkhead[3];
    arrayout[2]=darkhead[4];
    arrayout[3]=darkexptime;
    return arrayout;
}
//
//=====
// Function name: ReadCMStartAndEndTime
// Input: filename
// Output: start time and end time
// Created: 2010 nov 10
// Last Mod: 2010 dec 18 (compatible for CorMag and E-KPol)
// Comments: Read the date time in the header of the CorMag
// data file and return start time and end time

```

```

//=====
function ReadCMStartAndEndTime(filename,CM){
  par=newArray(2);
  imgString=File.openAsString(filename);
  indexofdatetime=indexOf(imgString,"DATETIME");
  indexofexptime=indexOf(imgString,"EXPTIME");
  hh=substring(imgString,indexofdatetime+22,indexofdatetime+24);
  mm=substring(imgString,indexofdatetime+25,indexofdatetime+27);
  ss=substring(imgString,indexofdatetime+28,indexofdatetime+30);
  exptime=substring(imgString,indexofexptime+24,indexofexptime+30);
  expcutind=indexOf(exptime,",");
  expt=substring(exptime,0,expcutind);
//  print(expt);
  if(CM==1){
    endtime=parseInt(mm)+(parseInt(ss)/60)-(1/60);
    starttime=endtime-(8/60);
  }
  else{
    indexoftime=indexOf(imgString,"TIMEOBS");
    min=substring(imgString,indexoftime+25,indexoftime+27);
    sec=substring(imgString,indexoftime+28,indexoftime+30);
    starttime=parseInt(min)+(parseInt(sec)/60);
    endtime=starttime+(parseInt(expt)/60000);
  }
  par[0]=starttime;
  par[1]=endtime;
  return par;
}

```