Mercuryの img ファイルの画像データを ASCI 変換し、excel や GraphR にて表示する

実行例（アンダーライン部が入力）

```
C:¥Documents and Settings¥My Documents >img-bmp_csv.exe
 <<<< Convert .img to .csv or .bmp     Ver=091219 >>>>
 Input file name *.img =? Axial1.img
 === Header of *.img is ===
Header top of 256bytes ia ...
{
HEADER_BYTES= 3584;
BYTE_ORDER=little_endian;
CCD_DETECTOR_ADC_OFFSET=0;
CCD_DETECTOR_DESCRIPTION=Mercury (2x2 bin mode);
CCD_DETECTOR_DIMENSIONS=512 512;
CCD_DETECTOR_IDENTIFICATION=MSC_MED_MERCURY_SN999;
CCD_DETECTOR_OPTIONS=imagekind:2 dezingermode:on

header is end
 1: monitor data and convert to csv file
 2: convert img to bmp (512*512 pixcel)
    Select 1 or 2 =? 2
===== BMP file out =====
 Intensity I= 1 ~ 1,048,543 is scaled as
       53*log10(I/16)
Input bmp file name ?= Axial1.bmp
 BG is 0: white, 1:black, =? 1

C:¥Documents and Settings¥My Documents >
```

Axial1.img という Mercury CCD の 512*512 ピクセルの画像を BMP 形式に変換する。

BG=1 の時、背景が黒で、ブラッグ斑点が白で表示され、BG=0 のときは白黒逆転表示。

```
header is end
 1: monitor data and convert to csv file
 2: convert img to bmp (512*512 pixcel)
    Select 1 or 2 =? 1
 If x<1 or 512<x or y<1 or 512<y then QUIT
Value at x,y =? 128,256
 151
Value at x,y =? -1,-1
csv file out   0:No / 1:Yes =? 1
 inout output region (x1,y1)-(x2,y2)
x1,y1=100,200
   value= 96
x2,y2=140,240
   value= 96
OK ?   0:No / 1:Yes =? 1
 output *.csv file name =? Axial1csv
```

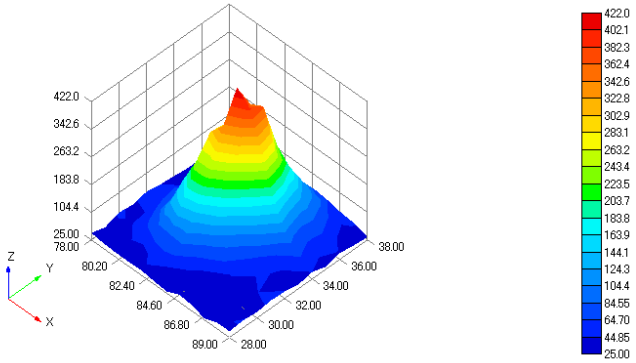
テキスト形式(csv)で出力するときは 1 を選ぶ。

指定した x,y 座標の強度をモニターする。

画像は左上が(1,1)、右下が(512,512)
水平方向に x 軸。垂直下方に y 軸
(x,y)に座標を入れると、そのピクセルの強度が表示。
0 以下または 513 以上を指定すると、強度モニターを終了する。

テキストデータに出力する範囲を指定する。
ＯＫ？で 0 と答えると、範囲指定をやり直せる。

このcsvファイルを GraphR で表示する。



あるいはエクセルで読み込んで処理できる。

BMP 形式のファイルは Windows のソフトで表示できる。

Mercury CCD では１ピクセルが２バイトで記述されているが、BMP でコンパクトに表すため、１ピクセルを１バイトに縮減している。そのため、

R=53*log10(PV/16)

という、対数変換をしている。但し、ピーク値 PV が１６以下に対しては R はゼロとする。

ソースプログラム（img-bmp_csv.for　2009/12/19　23:54　5KB）

```
* Mercury IMG file read and scv or bmp out 2009/12/19
* IMG file format
*   Header part is 3584byte (87 lines)
*   Standard data is consisted of 512x512 bytes
*      Intensity date is 2 bytes
*      Intensity=ichar(byte1)+256*ichar(byte2)

*      input(1) *.img and output(2) *.bmp or *.csv file names
       character*64 fnm1,fnm2
       character*256 da
       integer pv(512,512),pvdat(512)
       print *, '<<<< Convert .img to .csv or .bmp   Ver=091219 >>>>'
       write(6,'(a$)') ' Input file name *.img =? '
       read(5,*) fnm1

*      check of header part
       open(1,file=fnm1,access='DIRECT',recl=1)
       write(*,*) '=== Header of *.img is ==='
       read(1,rec=1) da
       write(*,'(A/A/)') 'Header top of 256bytes ia ...',da
       write(*,'(A)') 'header is end'
```

```
*----- intensity data reading -----
      j=3585
      do 20 iy=1,512
        call rd1ln(j,pvdat)
      do 20 ix=1,512
        pv(ix,iy)=pvdat(ix)
   20 continue
      close(1)
*---- select job ----
      write(*,'(A)') ' 1: monitor data and convert to csv file'
      write(*,'(A)') ' 2: convert img to bmp (512*512 pixcel)'
      write(*,'(A$)') '   Select 1 or 2 =? '
      read(*,*) lans
      if(lans.le.1) then
*----- monitor intensity data
*      print *, 'Input pixecl position (x,y)'
      print *, 'If x<1 or 512<x or y<1 or 512<y then QUIT'
   40 write(6,'(A$)')  'Value at x,y =? '
        read(5,*) ix,iy
          if(ix.le.0.or.iy.le.0) goto 60
          if(ix.gt.512.or.iy.gt.512) goto 60
        print *,pv(ix,iy)
      goto 40
*----- output into csv format
   60 write(*,'(A$)') 'csv file out  0:No / 1:Yes =? '
      read(*,*) jans
      if(jans.gt.0) then
   65   write(*,'(A)') ' inout output region (x1,y1)-(x2,y2)'
          write(*,'(A$)') 'x1,y1='
          read(*,*) x1,y1
          if(x1.le.0.or.y1.le.0) goto 70
            print *,' value= ',pv(x1,y1)
          write(*,'(A$)') 'x2,y2='
          read(*,*) x2,y2
          if(x2.gt.512.or.y2.gt.512) goto 70
            print *,' value= ',pv(x1,y1)
   70   write(*,'(A$)') 'OK ?  0:No / 1:Yes =? '
          read(*,*) JYN
          if(JYN.le.0) goto 65
        write(*,'(A$)') ' output *.csv file name =? '
        read(*,*) fnm2
        open(2,file=fnm2,status='UNKNOWN')
          write(2,'(A)') 'ﾃﾞｰﾀ形式, 1'
          write(2,'(A,A)')  'Mercury img file',FNM1
          write(2,'(A)') ' cut from 512*512 data'
          write(2,'(512(A1,I5))') (',',ix,ix=x1,x2)
        do 50 iy=y1,y2
          write(2,'(I5$)') iy
          write(2,'(511(A1,I5))') (',',pv(ix,iy),ix=x1,x2)
   50   continue
```

```
      close(2)
      endif
      else
*===== write in bmp format
      write(*,'(A)') '===== BMP file out ====='
      write(*,'(A)') ' Intensity I= 1 ~ 1,048,543 is scaled as'
      write(*,'(A)') '     53*log10(I/16) '
      write(*,'(A$)') 'Input bmp file name ?= '
      read(*,*) fnm2
        open(2,file=fnm2,access='DIRECT',recl=1)
      write(*,'(A$)') ' BG is 0: white, 1:black, =? '
        read(*,*) jwb
*----- bmp header
      call header(jwb)
*----- 1 byte data write
      n=1078
      do 90 iy=512,1,-1
      do 90 ix=1,512
        pk=pv(ix,iy)
      if(pk.le.16) then
        pk=0.0
      else
        pk=53.0*log10(pk/16)
      endif
        n=n+1
        write(2,rec=n) char(int(pk))
   90 continue
      close(2)
      endif
*----- QUIT -----
      stop
      end

      subroutine rd1ln(j,pvdat)
      integer pvdat(512)
      character*1 da1,da2
      integer j
      do 10 ix=1,512
        read(1,rec=j) da1
        j=j+1
        read(1,rec=j) da2
        pv=ichar(da1)+256*ichar(da2)
        j=j+1
      if(pv.ge.32768) then
        pv=(pv-32768)*32
      endif
        pvdat(ix)=pv
   10 continue
      return
      end
```

データは２バイト
　０～２５５までは先頭１バイト目
　これ以上は後の２バイト目に
二進法でファイルに記載されている。
　ただし、32,768 以上になると５ビットシフトし、最大強度は 1,048,543
　これを復元するために
　if(pv.ge.32768) pv=(pv-32768)*32
と変換している。

```
*--------------  bmp file header  -----------
      subroutine header(jwb)
      integer id(54)
      data id/66,77,54,4,4,0,0,0,0,0,54,4,0,0,40,0,0,0,0,2,0,0,0,2,
     +  0,0,1,0,8,7*0,4,0,175,27,0,0,175,27,10*0/
      do 10 n=1,54
        write(2,rec=n) char(id(n))
   10 continue
      n=55
      if(jwb.gt.0) then
      do 20 nn=0,255
        write(2,rec=n) char(nn)
        write(2,rec=n+1) char(nn)
        write(2,rec=n+2) char(nn)
        write(2,rec=n+3) char(0)
   20 n=n+4
      else
      do 30 nn=255,0,-1
        write(2,rec=n) char(nn)
        write(2,rec=n+1) char(nn)
        write(2,rec=n+2) char(nn)
        write(2,rec=n+3) char(0)
   30 n=n+4
      endif
      return
      end
```

データに関するメモ
○　Mercury の img ファイル
通常はヘッダーに記載通り、512×512 ピクセルの画像である。
ヘッダー部は 3584 バイトが設定されており、ASCI コード〈　〉で囲まれたデータで表示されており、余分な領域には 00(ヌル)が書かれている。
○画素１つあたり、２バイトのデータで構成されている。先頭の１バイト目は０～２５５の強度の値で、２バイト目が２５６～６５２７９の上位を２の補数表現で記載している。
○但し、32,768 以上になると５ビットシフトし、最大強度 1,048,543 までデータに記載できるようになっている。
これを復元するため、プログラムでは　if(pv.ge.32768) pv=(pv-32768)*32
と変換している。
○Mercury 上では画面の左上が(0.5,0.5)で右下が(511.5,511.5)と表示されるが、本プログラムでは整数化して(1,1)~(512,512)としている。
○画面の周辺部は３～６ピクセルの幅で０（ヌル）のデータで縁取られている。