

```

#!/usr/bin/python
#### Meritev in prikaz smernega diagrama - S53MV 26.09.2016 ####

#izvajanje potrebuje najmanj Anaconda2-4.1.1-Windows-x86 in pyserial-2.7.win32_py3k
import time,serial,math #uporabljene funkcije
rxime='/dev/ttyUSB1'      #ime zaporednega vmesnika sprejemnika
vrtime='/dev/ttyUSB0'    #ime zaporednega vmesnika vrtiljaka

print '*** Meritev in prikaz smernega diagrama - S53MV 26.09.2016 ***'
ff=raw_input('Ime zapisa (prazno=nova meritev): ')

if len(ff)==0:          #nova meritev
    nn=raw_input('Naslov: ')

    x=[0.0]              #pripravi obe polji za rezultat
    y=[99.99]
    i=1
    while i<=3600:
        x=x+[i*0.1]
        y=y+[99.99]
        i=i+1

    #odpri comporta rx in vrt, 115200bps, 8bit, brez paritete
    print 'Pocakaj na lenobni USB...'
    rx=serial.Serial(rxime,115200,timeout=3.0)
    rx.setRTS(False)    #sprosti /BOOTLOADER
    rx.setDTR(False)   #sprosti /RESET
    vrt=serial.Serial(vrtime,115200,timeout=3.0)
    vrt.setRTS(False)  #sprosti /BOOTLOADER
    vrt.setDTR(False)  #sprosti /RESET
    time.sleep(3)
    vrt.flushInput()   #pocisti sprejemni vmesnik vrtiljaka
    vrt.write('6D')    #pozeni vrtiljak v desno

    i=0                  #zanka merjenja antene
    m=-1.0
    while i<3600:
        s=vrt.readline()
        k=float(s[0:5])
        rx.flushInput()
        rx.readline()
        s=rx.readline()
        r=float(s[0:5])
        print 'Kot:',k, ' Jakost:',r, 'dB ',chr(13),
        i=int(k*10.0+0.5)
        y[i]=r
        if r>m:          #iscsi maksimum
            m=r
            km1=k
        if r==m:
            km2=k

    vrt.write('9L')     #vrtiljak v levo z navisjo hitrostjo
    t=time.ctime()      #zabelezi cas meritve
    f=t[20:24]+t[4:7]+t[8:10]+t[11:13]+t[14:16]+t[17:19]+' .csv'   #ime
    izhodnega zapisa
    print '\r\nMeritev v zapis:',f
    iz=open(f,'w')      #shrani izmerjene podatke v izhodni zapis
    iz.write(t+'\r\n')  #prva vrstica: cas meritve
    iz.write(nn+'\r\n') #druga vrstica: naslov meritve
    i=0                 #3601 vrstic podatkov: kot+jakost
    while i<=3600:

```

```

        iz.write('%05.1f'%(x[i])+ ' '+'%05.2f'%(y[i])+'\r\n')
        i=i+1
    iz.close()           #zapri izhodni zapis
    rx.close()          #zapri oba comporta
    vrt.close()

else:                   #preberi podatke iz vhodnega zapisa
    vz=open(ff, 'r')
    s=vz.readline()    #prva vrstica: cas meritve
    t=s[0:(len(s)-2)]
    s=vz.readline()    #druga vrstica: naslov meritve
    nn=s[0:(len(s)-2)]
    i=0                 #3601 vrstic podatkov: kot+jakost
    x=[]
    y=[]
    m=-1.0
    while i<=3600:
        s=vz.readline()
        k=float(s[0:5])
        r=float(s[6:11])
        x=x+[k]
        y=y+[r]
        if r>m:         #iscsi maksimum
            m=r
            km1=k
        if r==m:
            km2=k
        i=i+1
    vz.close()         #zapri vhodni zapis

print nn,t             #izpisi naslov in cas meritve

i=0                   #normaliziraj smerni diagram na max
yl=[]                 #pretvori v linearno skalo yl
while i<=3600:
    y[i]=y[i]-m
    yl=yl+[math.pow(10.0,y[i]*0.05)]
    i=i+1

km=(km1+km2)*0.5     #povprecni polozaj maksimuma
om='Odklon: '+str(km-180.0)
print om

i=1                   #integriraj smerni diagram
s=0.0
im=int(km*10+0.5)
while i<1800:
    i1=im-i
    if i1<0:
        i1=i1+3600
    i2=im+i
    if i2>3600:
        i2=i2-3600
    ff=math.pow(10.0,(y[i1]*0.1))+math.pow(10.0,(y[i2]*0.1))
    s=s+ff*math.sin(math.radians(i*0.1))
    i=i+1
D=(7200/math.pi)/s   #izracunaj smernost
dd='Smernost: %5.1f'%(D)+' = %5.2f'%(10*math.log(D,10))+' dBi'
print dd

i=int(km*10+0.5)     #poisci -3dB meji glavnega lista
while i>=0:

```

```

        km1=i*0.1
        if y[i]<-3.0:
            i=0
        i=i-1
i=int(km*10+0.5)
while i<=3600:
    km2=i*0.1
    if y[i]<-3.0:
        i=3600
    i=i+1
ss='Sirina -3dB: '+str(km2-km1)
print ss

j=1
while j>0:
    j=int(input('Izberi diagram (1=k.log, 2=k.lin, 3=p.log, 4=p.lin, 0=izhod): '))
    if j>0 and j<5:
        print 'Pocakaj na nalaganje grafike...'
        import matplotlib.pyplot as plt
        if j==1 or j==2: #risanje v kartezičnih koordinatah?
            plt.plot([km,km],[-40.0,2.0],'y-') #narisi položaj maksimuma in
sirino lista
            plt.plot([km1,km1],[-40.0,2.0],'r-')
            plt.plot([km2,km2],[-40.0,2.0],'r-')
            if j==1:
                plt.plot(x,y,'b-') #narisi log smerni diagram
                plt.axis([0.0,360.0,-40.0,2.0])
                plt.yticks
                ([-40.0,-35.0,-30.0,-25.0,-20.0,-15.0,-10.0,-5.0,0.0])
                plt.ylabel('Relativna jakost sprejema v dB')
            else:
                plt.plot(x,yl,'b-') #narisi lin smerni diagram
                plt.axis([0.0,360.0,0.0,1.05])
                plt.yticks([0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0])
                plt.ylabel('Jakost sprejema (amplituda) v linearnem merilu')
            plt.xticks([0.0,45.0,90.0,135.0,180.0,225.0,270.0,315.0,360.0])
            plt.grid(True)
            plt.title(t+'\n'+nn)
            plt.xlabel(ss+' '+om+' '+dd)
            plt.show()
        if j==3 or j==4: #risanje v polarnih koordinatah
            #priprava krogov in pretvorba podatkov
            x1=[]
            y1=[]
            x2=[]
            y2=[]
            x3=[]
            y3=[]
            x4=[]
            y4=[]
            xx=[]
            yy=[]
            i=0
            while i<=3600:
                s=math.sin(math.radians(i*0.1))
                c=math.cos(math.radians(i*0.1))
                x1=x1+[s/4]
                y1=y1+[c/4]
                x2=x2+[s/2]
                y2=y2+[c/2]
                x3=x3+[3*s/4]
                y3=y3+[3*c/4]
                x4=x4+[s]

```

```

y4=y4+[c]
if j==3:
    a=(y[i]+40.0)/40.0      #logaritemsko merilo
    if a<0.0:
        a=0.0
    else:
        a=y1[i]            #linearno merilo
    xx=xx+[a*s]
    yy=yy+[-a*c]
    i=i+1
l=0.2                      #narisi kroge koordinatne mreze
plt.plot(x1,y1,'g-',lw=l)  #krog polmera 0.25
plt.plot(x2,y2,'g-',lw=l)  #krog polmera 0.5
plt.plot(x3,y3,'g-',lw=l)  #krog polmera 0.75
plt.plot(x4,y4,'g-',lw=l)  #krog polmera 1
k=0                        #razdelki po 30 stopinj
while k<360:
    xk=math.sin(math.radians(k))*0.25
    yk=math.cos(math.radians(k))*0.25
    plt.plot([0.0,xk],[0.0,yk],'g-',lw=l)
    k=k+30
k=0                        #razdelki po 15 stopinj
while k<360:
    xz=math.sin(math.radians(k))*0.25
    yz=math.cos(math.radians(k))*0.25
    xk=math.sin(math.radians(k))*0.5
    yk=math.cos(math.radians(k))*0.5
    plt.plot([xz,xk],[yz,yk],'g-',lw=l)
    k=k+15
k=0                        #razdelki po 5 stopinj
while k<360:
    xz=math.sin(math.radians(k))*0.5
    yz=math.cos(math.radians(k))*0.5
    xk=math.sin(math.radians(k))
    yk=math.cos(math.radians(k))
    plt.plot([xz,xk],[yz,yk],'g-',lw=l)
    k=k+5
r=1.03                    #narisi polozaj maksimuma in sirino lista
plt.plot([0.0,math.sin(math.radians(km))*r],[0.0,-math.cos
(math.radians(km))*r],'y-')
plt.plot([0.0,math.sin(math.radians(km1))*r],[0.0,-math.cos
(math.radians(km1))*r],'r-')
plt.plot([0.0,math.sin(math.radians(km2))*r],[0.0,-math.cos
(math.radians(km2))*r],'r-')

plt.plot(xx,yy,'b-')      #narisi smerni diagram
plt.axis([-1.05,1.05,-1.05,1.05])
plt.xticks([])
plt.yticks([])
plt.title(t+'\n'+nn)
plt.xlabel(ss+'      '+om+'      '+dd)
if j==3:
    plt.ylabel('Logaritemsko radialno merilo (razpon 40dB)')
else:
    plt.ylabel('Linearno radialno merilo za jakost (amplituda)')
plt.axes().set_aspect('equal')
plt.show()

```

#konec programa