

23. Seminar Radijske Komunikacije

Samogradnja antenske merilnice

Matjaž Vidmar

LSO, FE, Ljubljana, 31.1.–2.2.2018

Seznam prosojnic predavanja: Samogradnja antenske merilnice

- 1 – Določanje smernosti antene
- 2 – Ravnina E smernega diagrama
- 3 – Ravnina H smernega diagrama
- 4 – Kazalčna meritev z vektorskim voltmetrom
- 5 – Meritev amplitude z moduliranim oddajnikom
- 6 – Meritev amplitude z lock-in sprejemnikom
- 7 – Predstavljena izbira merilnih pripomočkov
- 8 – Zasnova vrtiljaka s koračnim motorjem
- 9 – Mikrokrmilnik z vmesnikom VCP
- 10 – Elektronika vrtiljaka
- 11 – Izvedbe vrtiljakov s koračnim motorjem
- 12 – Meritev ω z žiroskopom
- 13 – Ulomkovna zanka MAX2871
- 14 – Načrt merilnega izvora z MAX2871
- 15 – Ulomkovni merilni izvor
- 16 – Načrt izvora 8..12GHz
- 17 – Izvor 8..12GHz
- 18 – 12GHz merilni detektor za lock-in sprejemnik
- 19 – Lock-in sprejemnik
- 20 – Lock-in DSP
- 21 – Programski jezik Python
- 22 – Krmiljenje meritve 1. del – zajem in obdelava podatkov
- 23 – Krmiljenje meritve 2. del – risanje smernih diagramov
- 24 – Izmerjeni smerni diagram v ravnini E
- 25 – Izmerjeni smerni diagram v ravnini H

$$G = \eta D$$

Teorija:
$$D = \frac{4 \pi |F(\Theta_{MAX}, \Phi_{MAX})|^2}{\oint_{4\pi} |F(\Theta, \Phi)|^2 d\Omega} = \frac{4 \pi |F(\Theta_{MAX}, \Phi_{MAX})|^2}{\int_0^\pi \int_0^{2\pi} |F(\Theta, \Phi)|^2 \sin \Theta d\Theta d\Phi}$$

Meritev N prerezov $\Phi_1, \Phi_2, \Phi_3 \dots \Phi_N$:
$$\oint_{4\pi} |F(\Theta, \Phi)|^2 d\Omega \approx \approx \frac{2\pi}{N} \int_0^\pi \left[|F(\Theta, \Phi_1)|^2 + |F(\Theta, \Phi_2)|^2 + |F(\Theta, \Phi_3)|^2 + \dots + |F(\Theta, \Phi_N)|^2 \right] \sin \Theta d\Theta$$

Meritev N približkov smernosti D_j @ $j=1, 2, 3, 4 \dots N$ $-\pi \leq \Theta \leq \pi$:

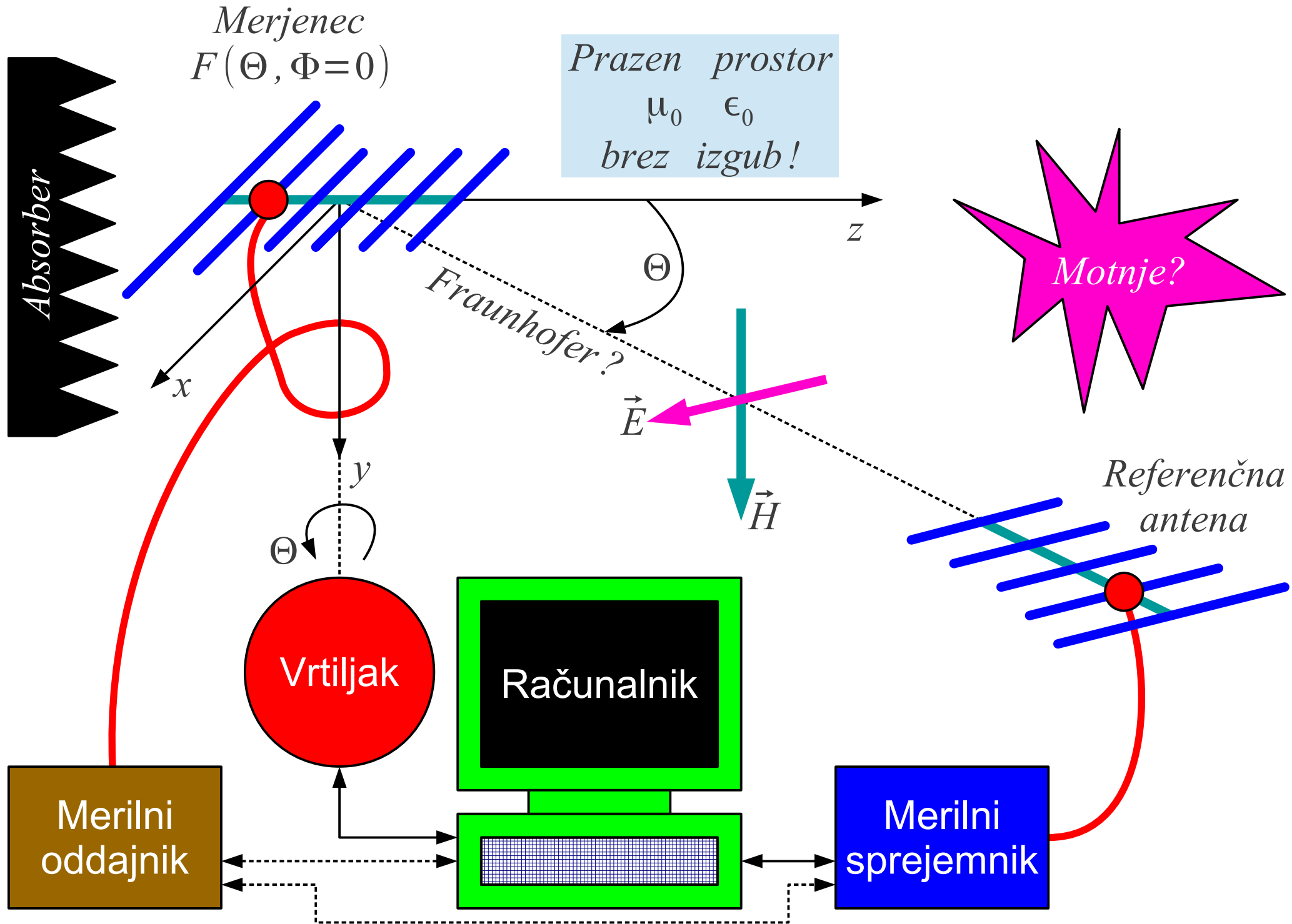
$$D_j = \frac{4 |F(\Theta_{MAX}=0, \Phi_{MAX}=\Phi_j)|^2}{\int_{-\pi}^{\pi} |F(\Theta, \Phi_j)|^2 \sin \Theta d\Theta} \quad D = \frac{N}{\frac{1}{D_1} + \frac{1}{D_2} + \frac{1}{D_3} + \dots + \frac{1}{D_N}}$$

Dva prereza $\Phi_1 = \Phi_E = 0$ (ravnina E) in $\Phi_2 = \Phi_H = \pi/2$ (ravnina H):

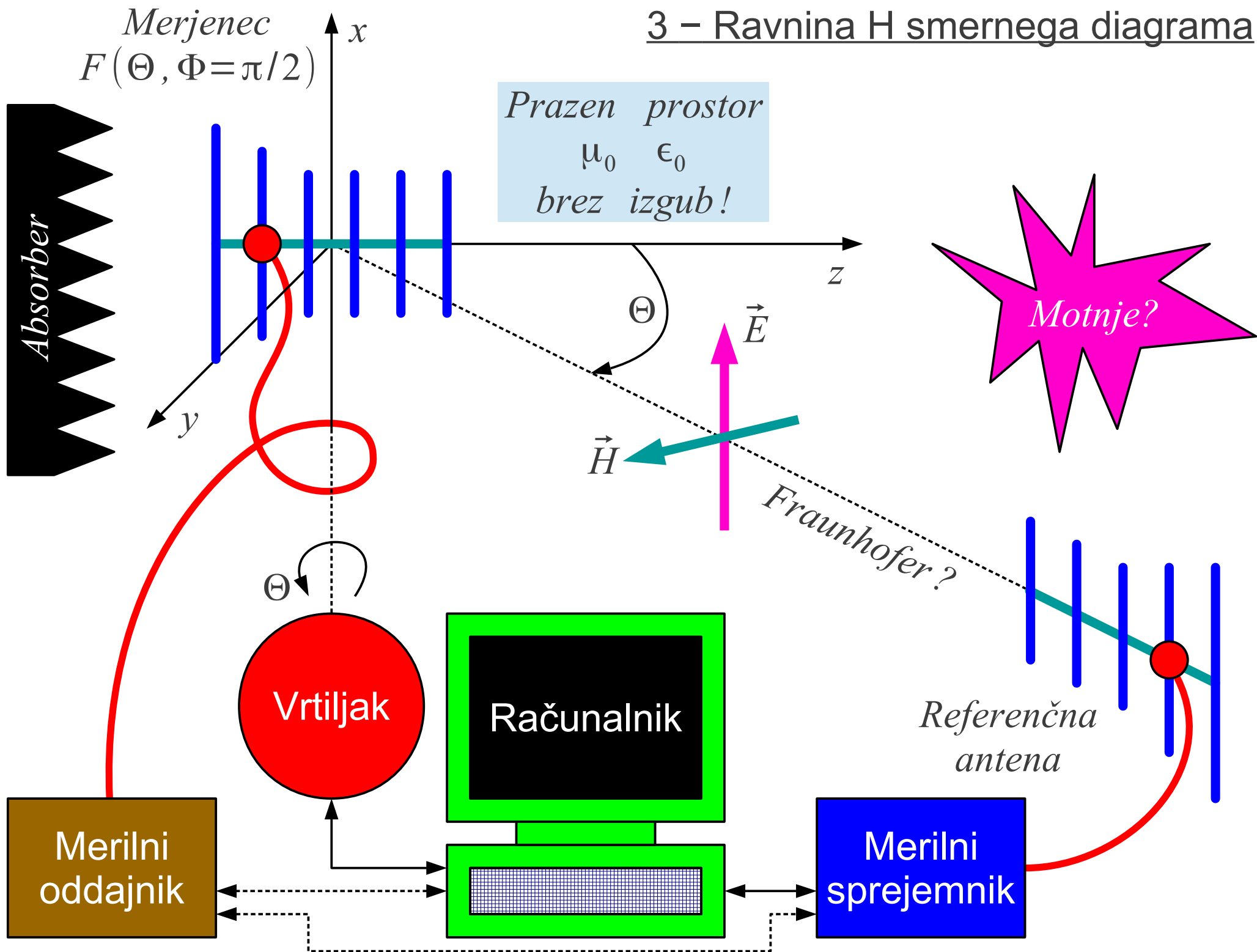
$$D_E = \frac{4 |F(\Theta_{MAX}=0, 0)|^2}{\int_{-\pi}^{\pi} |F(\Theta, 0)|^2 \sin \Theta d\Theta} \quad D_H = \frac{4 |F(\Theta_{MAX}=0, \pi/2)|^2}{\int_{-\pi}^{\pi} |F(\Theta, \pi/2)|^2 \sin \Theta d\Theta} \quad D = \frac{2}{\frac{1}{D_E} + \frac{1}{D_H}}$$

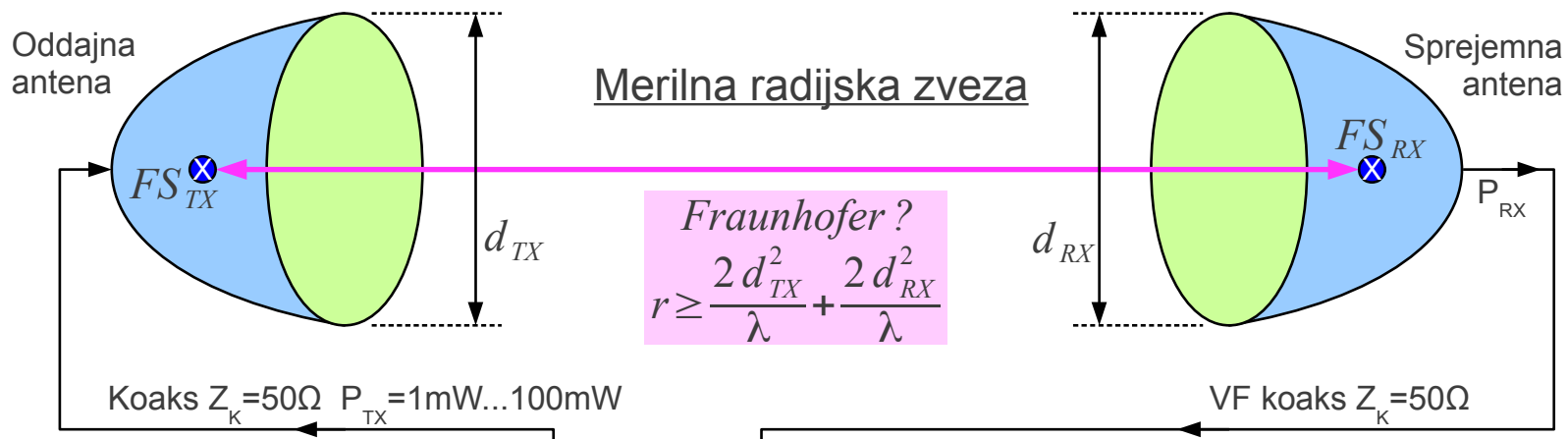
1 – Določanje smernosti antene

2 – Ravnina E smernega diagrama

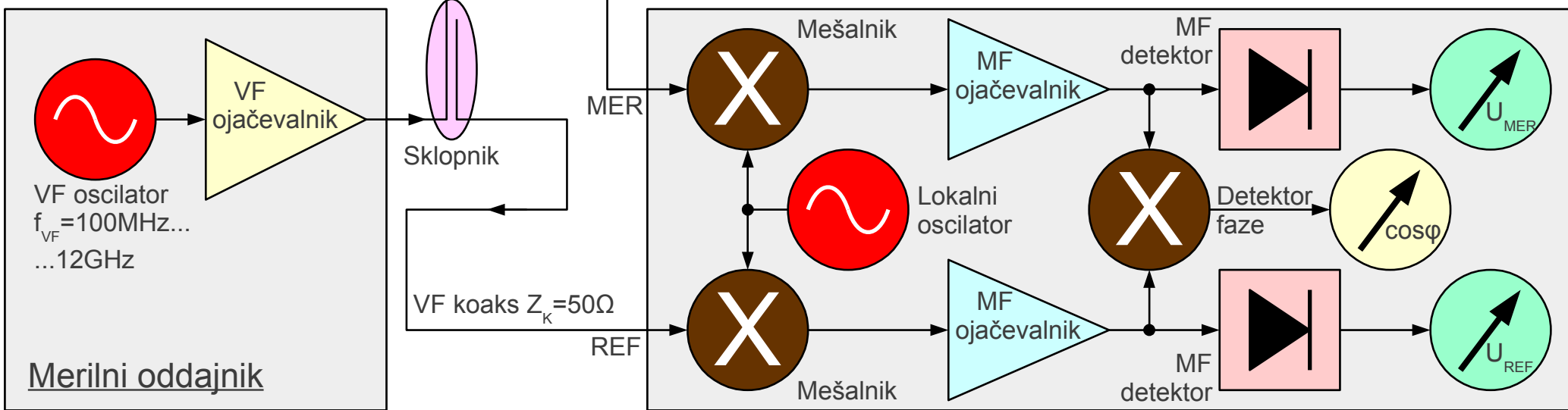


3 – Ravnina H smernega diagrama

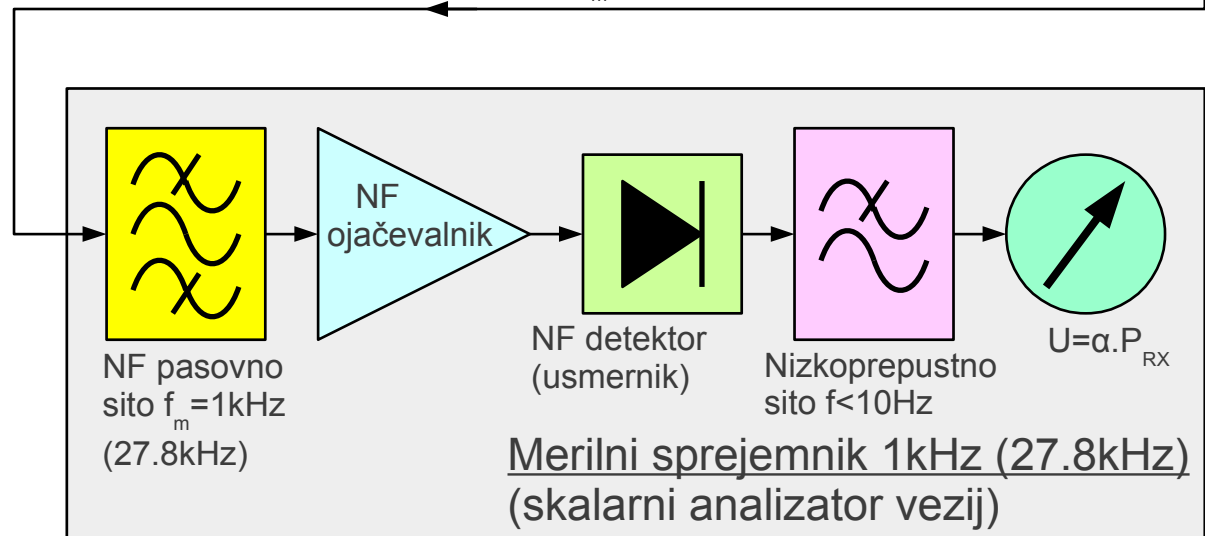
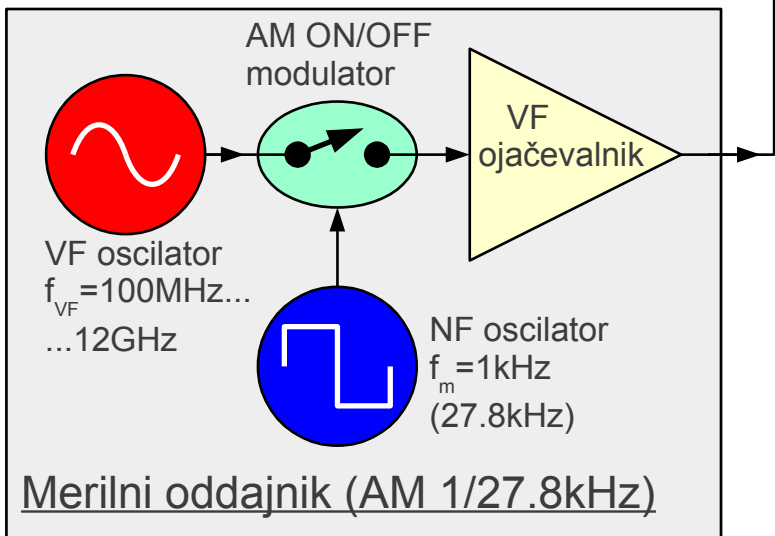
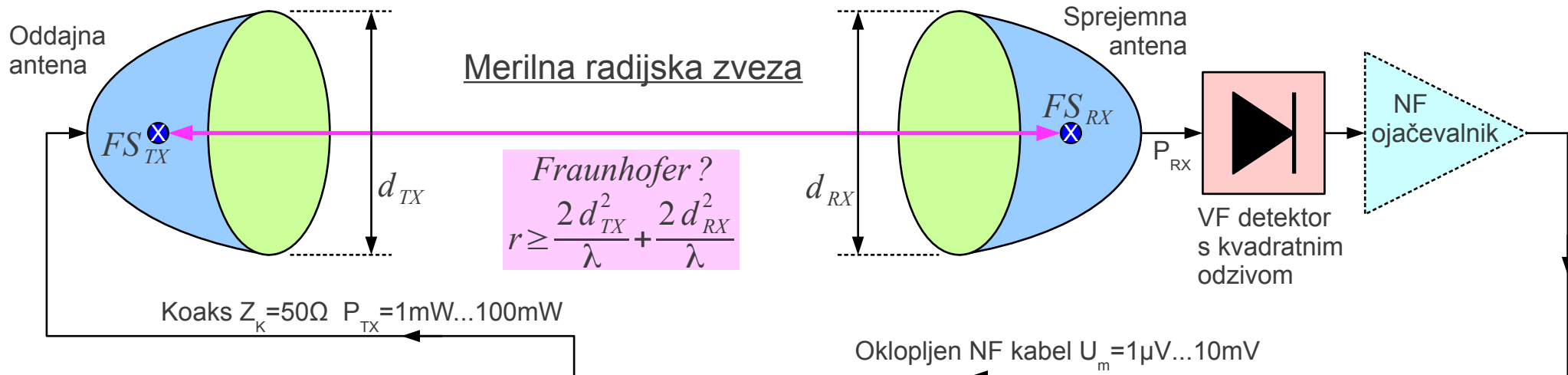




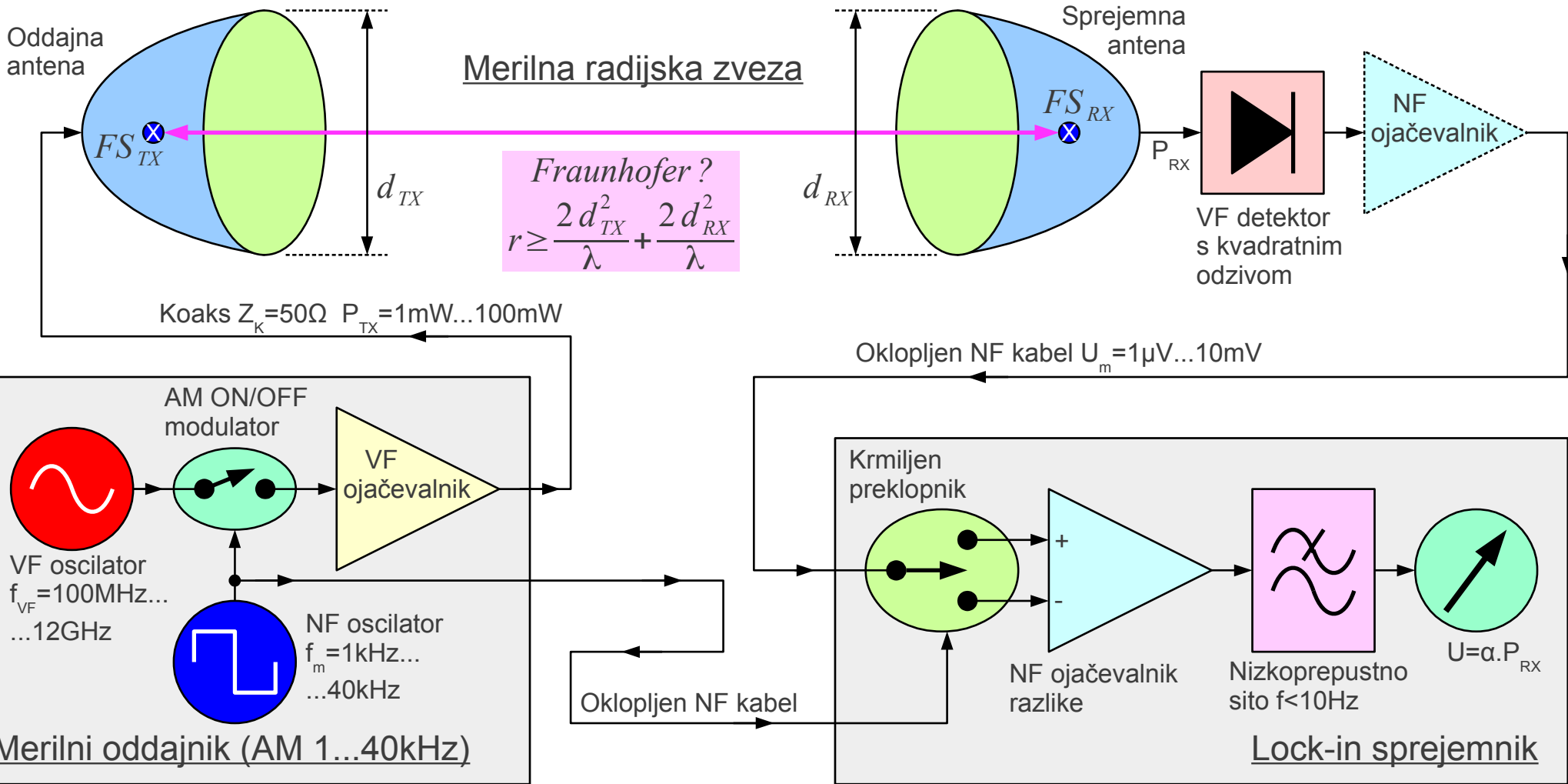
Vektorski voltmeter
(vektorski analizator vezij)



4 – Kazalčna meritev z vektorskim voltmetrom



5 – Meritev amplitude z moduliranim oddajnikom



6 – Meritev amplitude z lock-in sprejemnikom

Nova merilna oprema:

- krčenje tržišča zaradi BITE (Built-In Test Equipment)?
- visoka cena in slaba kakovost zaradi maloserijske izdelave?
- celovitost ponudbe za antensko merilnico?
- neučinkovito oziroma neobstoječe servisiranje?
- licenčna programska oprema?
- časovno trajanje podpore proizvajalcev?

Rabljena merilna oprema (10...20 let):

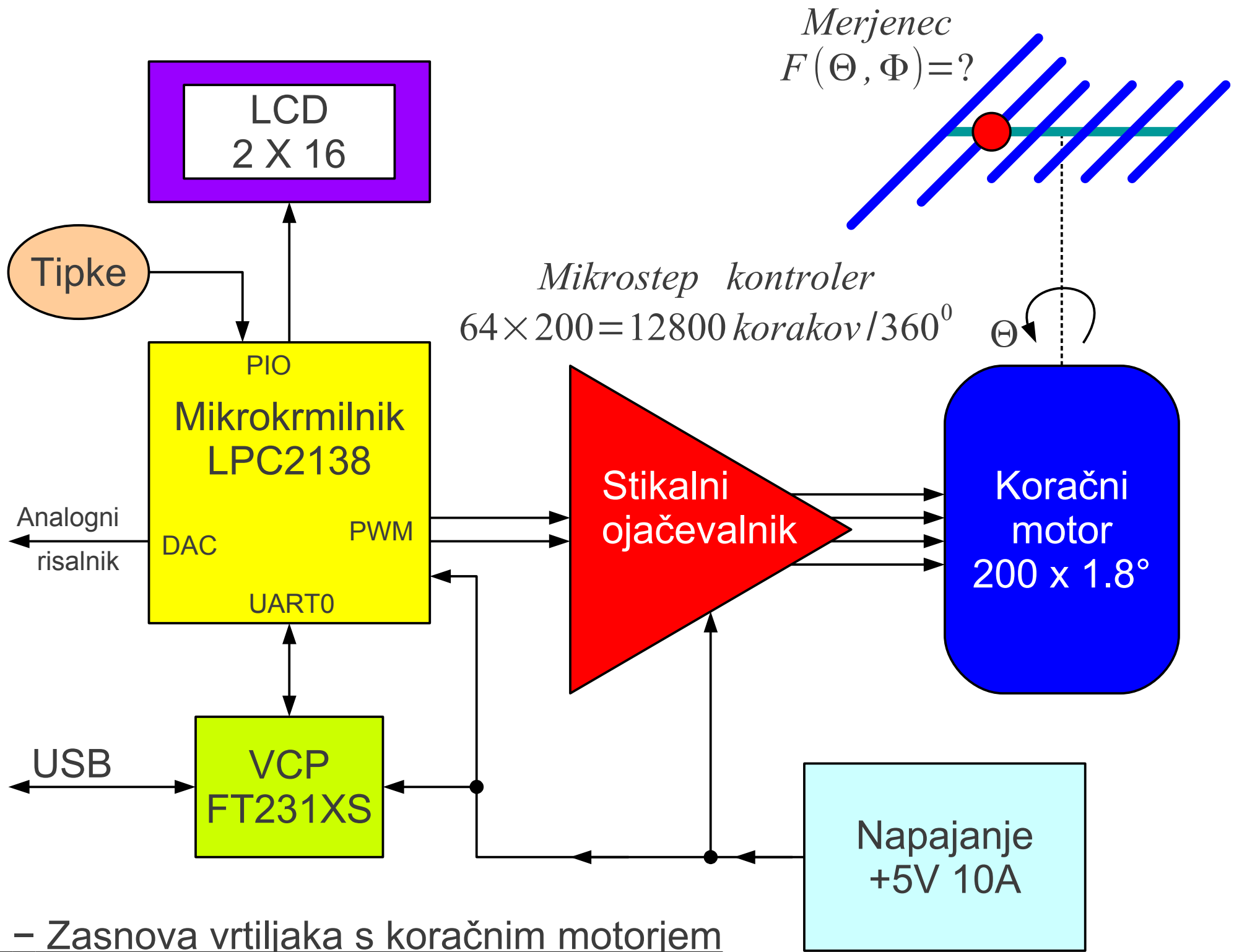
- velika masa, izmere, poraba in hrup starih merilnih inštrumentov?
- izredno dragi vmesniki GPIB z licenčno programsko podporo?
- razpoložljivost rezervnih delov (kanibalizacija)?
- zahteva po samogradnji manjkajočih merilnih pripomočkov?

Samogradnja merilne opreme:

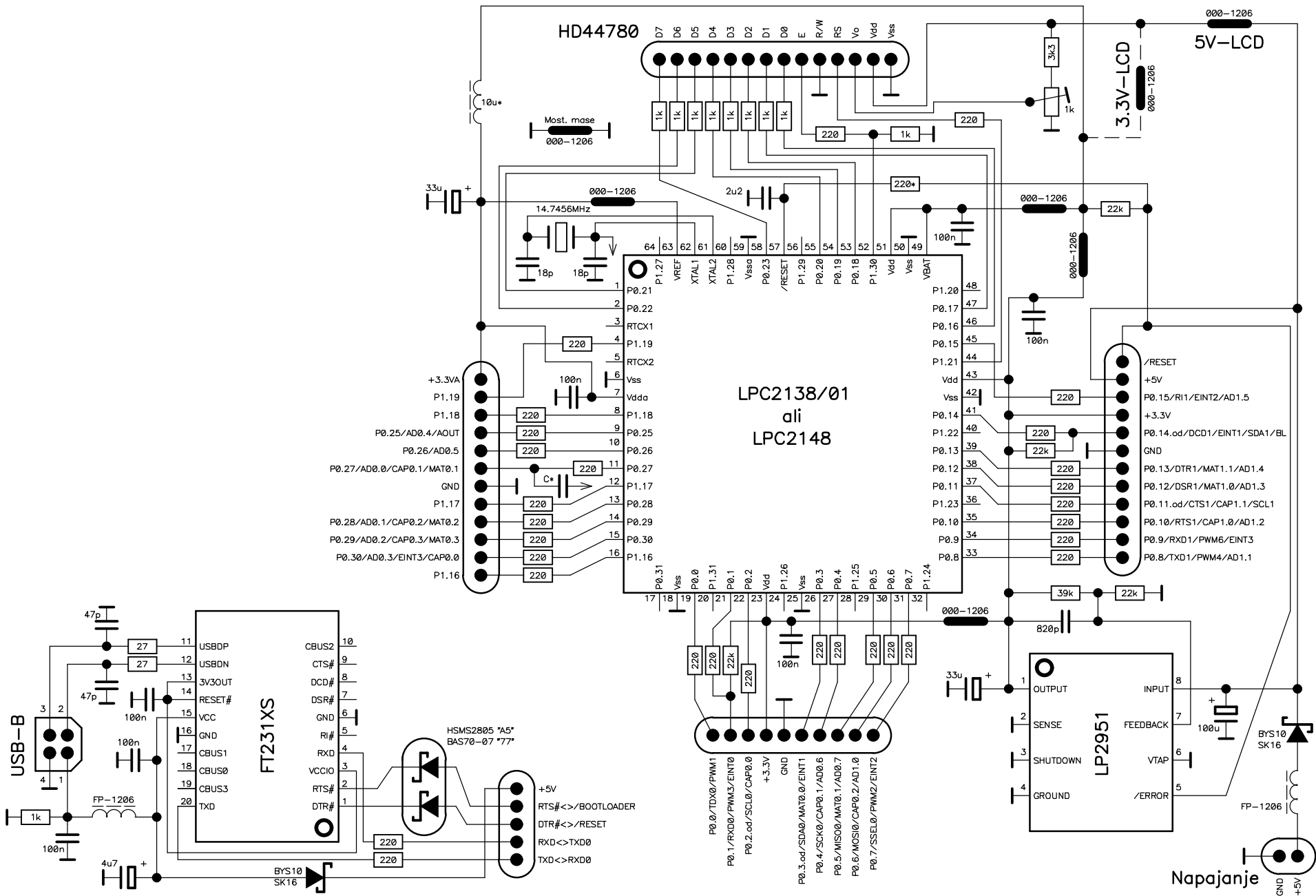
- možnosti samogradnje srednje zahtevnih merilnih inštrumentov?
- natančnost in zanesljivost samogradnje s sodobnimi gradniki?
- potrebna količina dela (čas je glavna cena) za samogradnjo?

7 – Predstavljena izbira merilnih pripomočkov

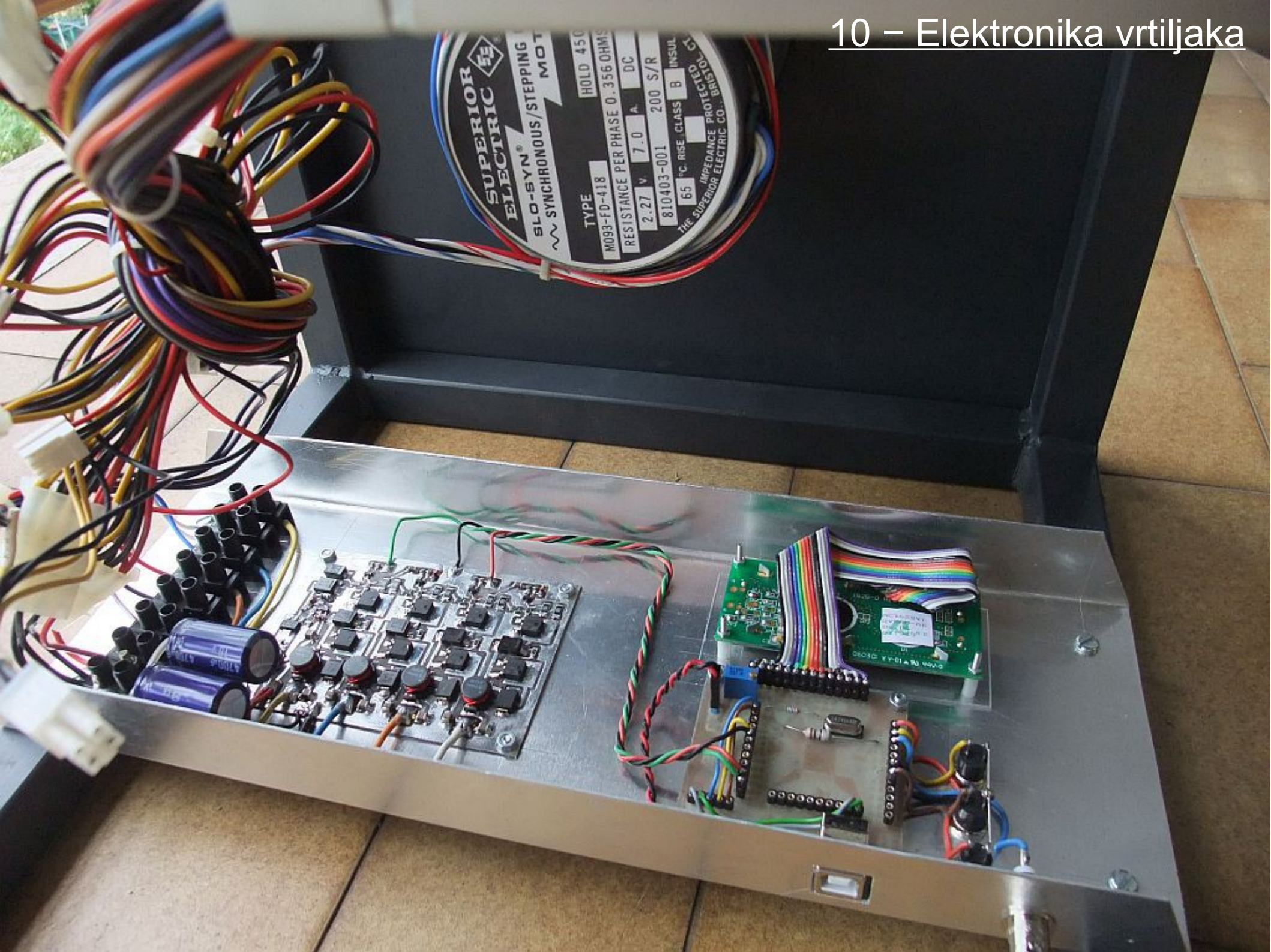
- samogradnja antenskega vrtiljaka s koračnim motorjem
- samogradnja visokofrekvenčnega vira z ulomkovno zanko do 12GHz
- samogradnja lock-in sprejemnika z detektorjem do 12GHz

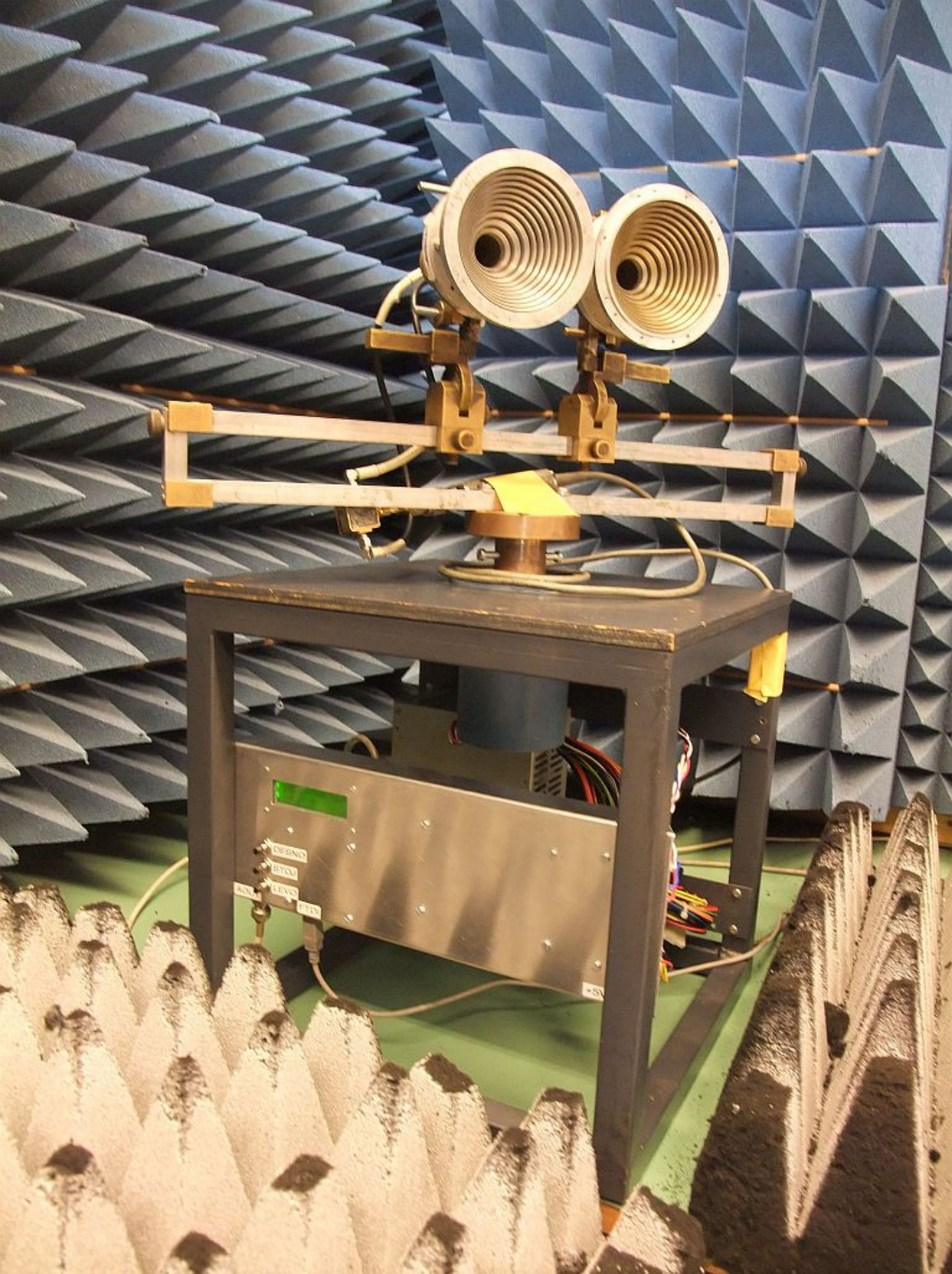


8 – Zasnova vrtiljaka s koračnim motorjem

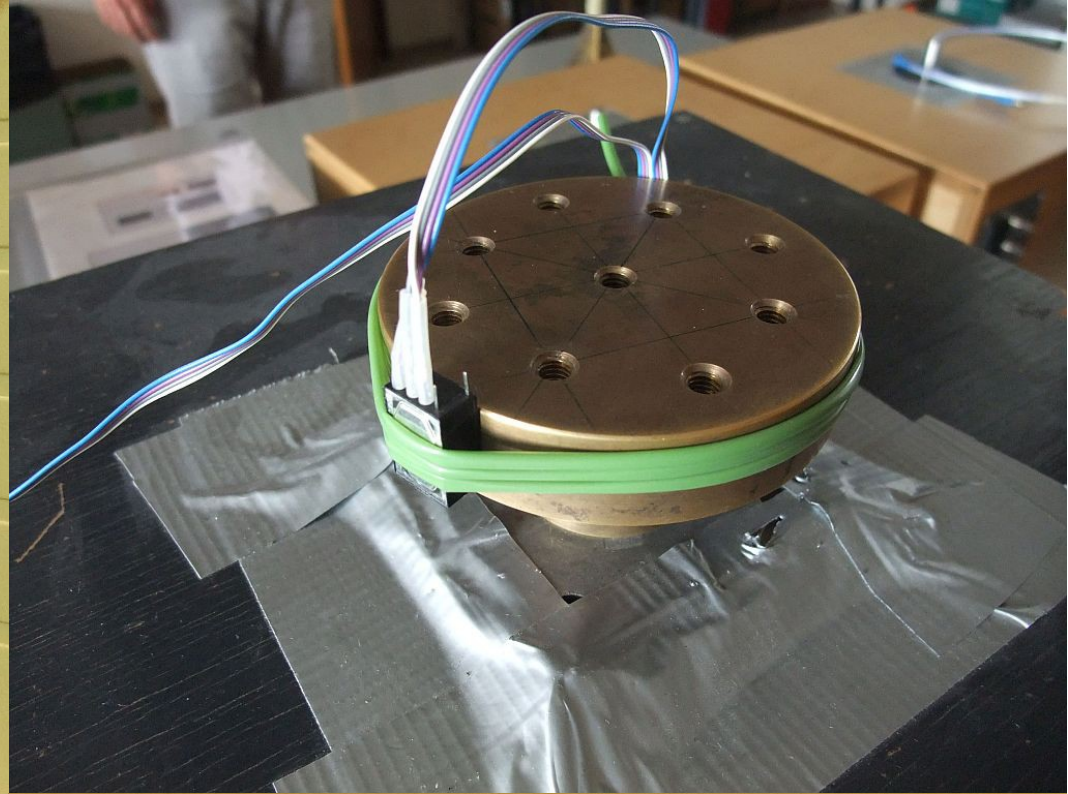
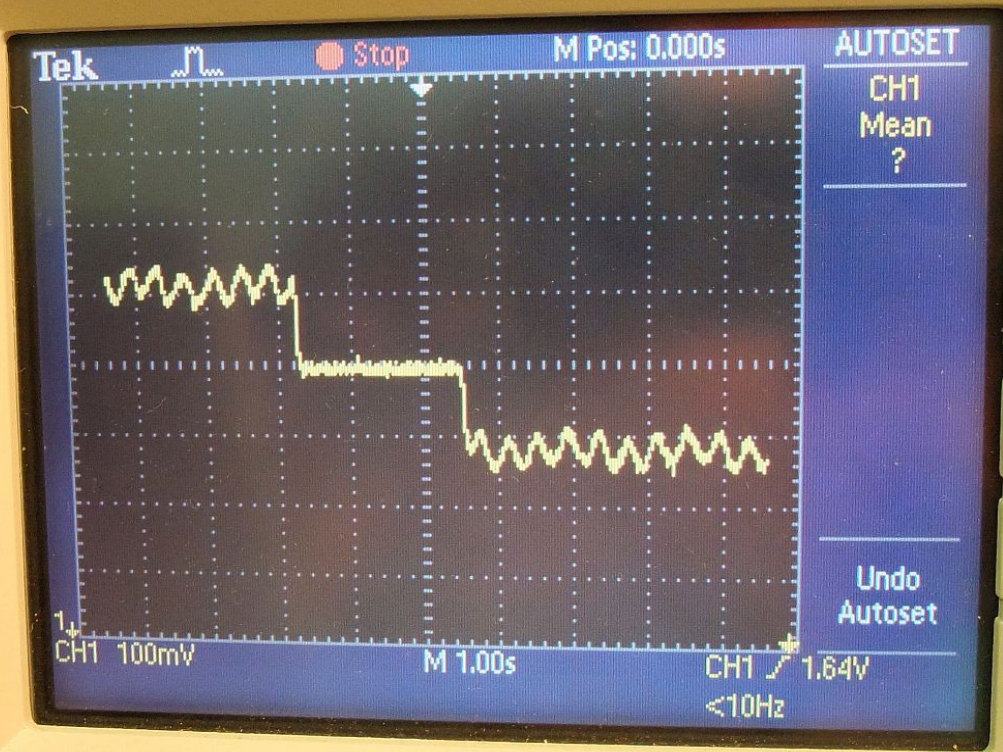


9 – Mikrokrmilnik z vmesnikom VCP





11- Izvedbe vrtiljakov s koračnim motorjem



Krmiljenje koračnega motorja:

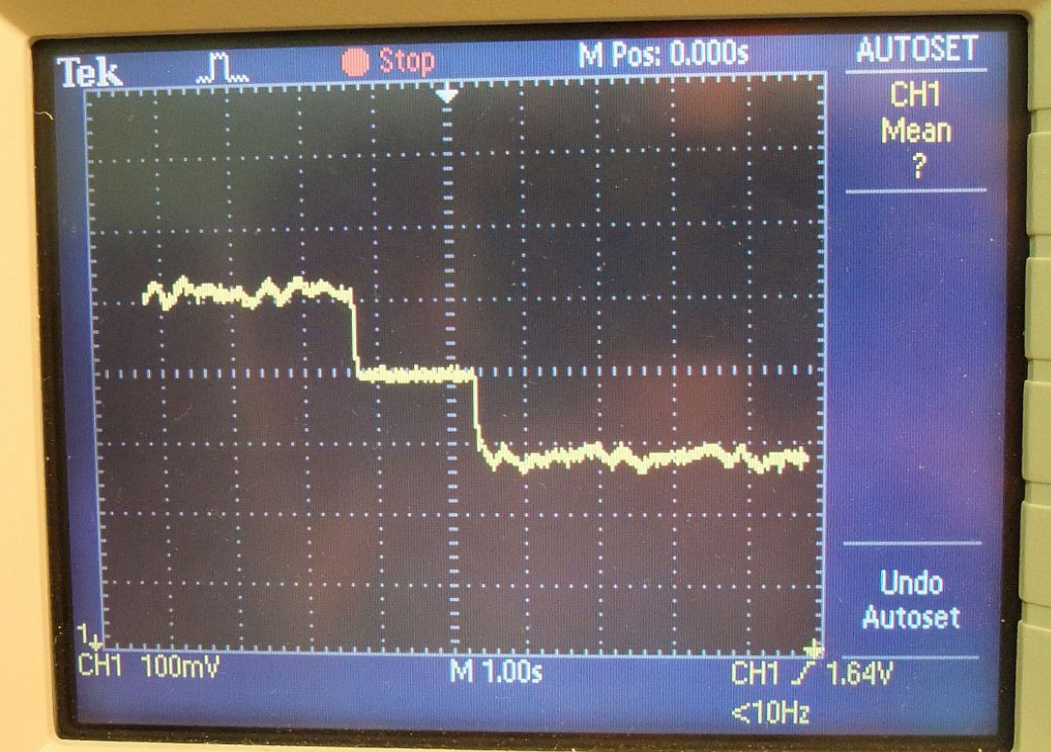
$$I_1 = A \cos \omega t$$

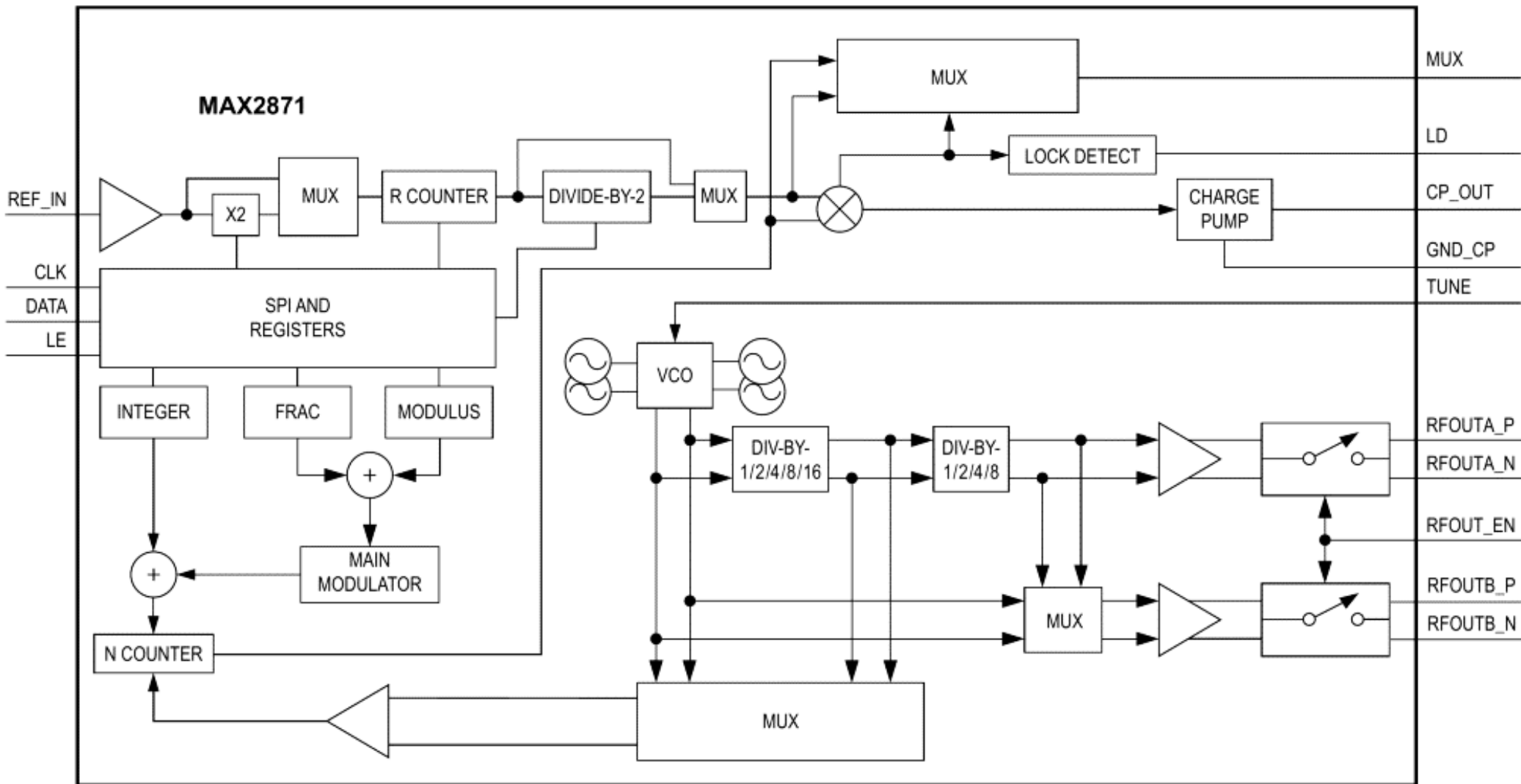
$$I_2 = A \sin \omega t$$

Krmiljenje koračnega motorja:

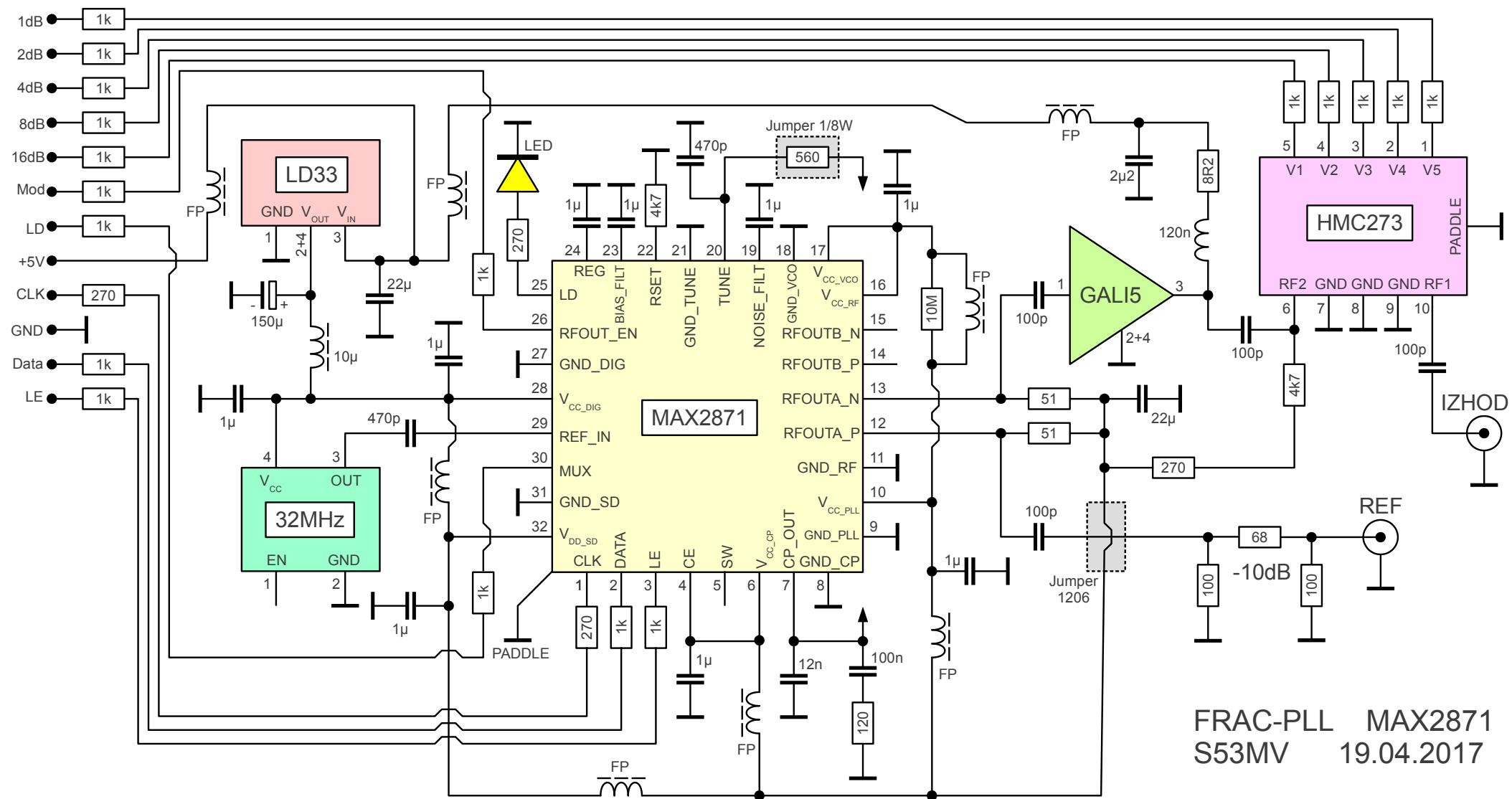
$$I_1 = A \cos \omega t + B \cos 3 \omega t$$

$$I_2 = A \sin \omega t + B \sin 3 \omega t$$



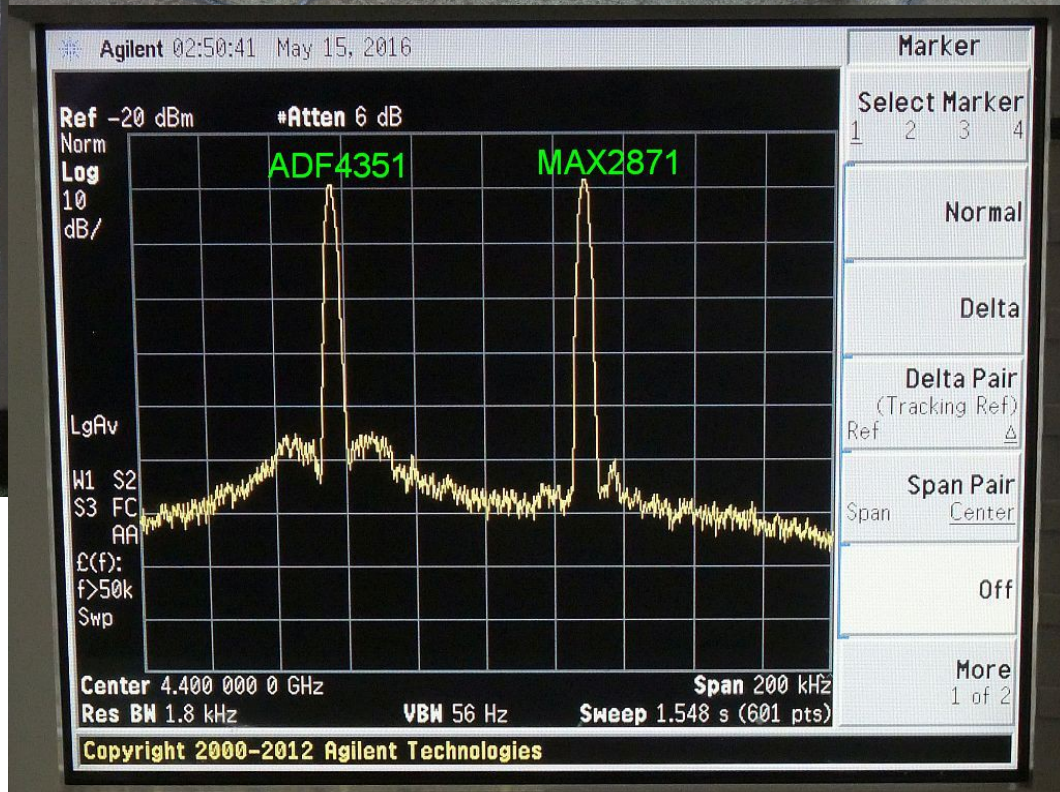
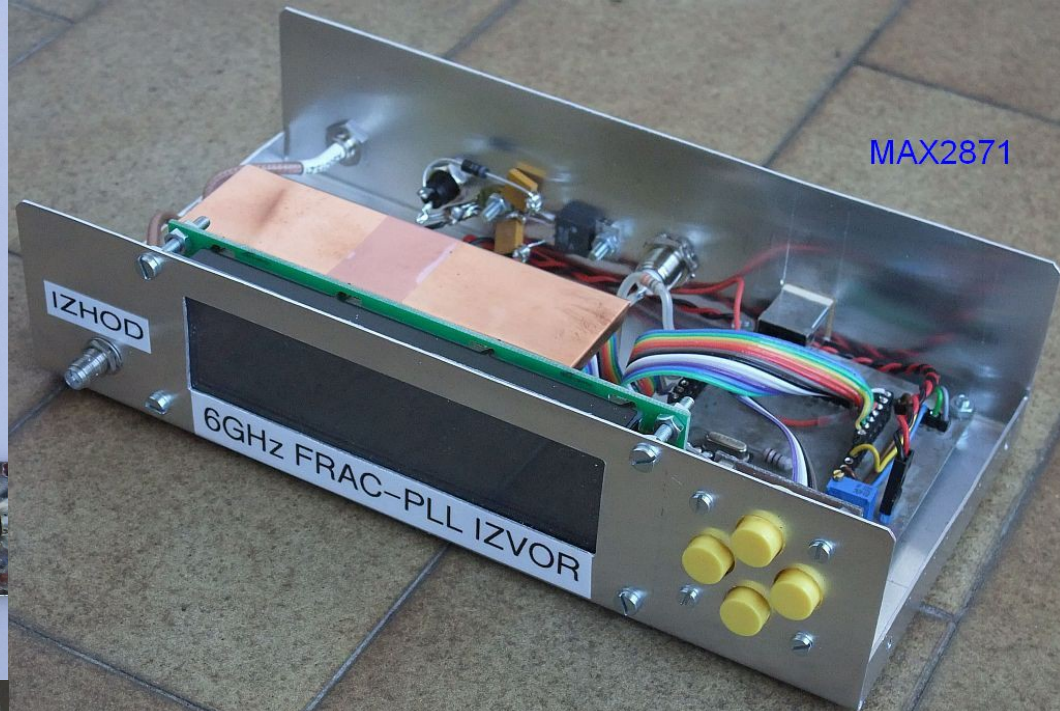
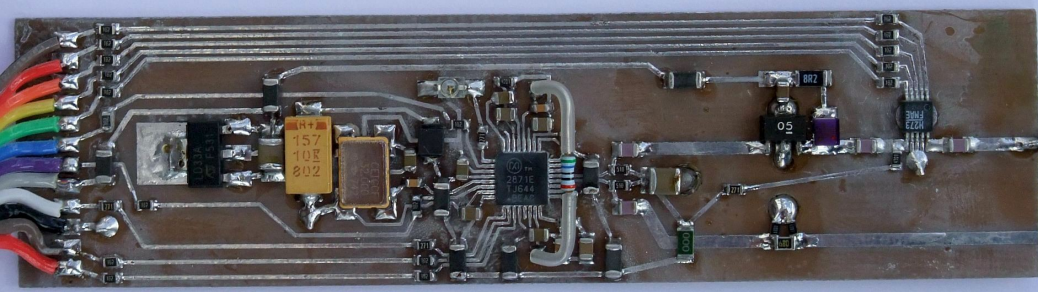
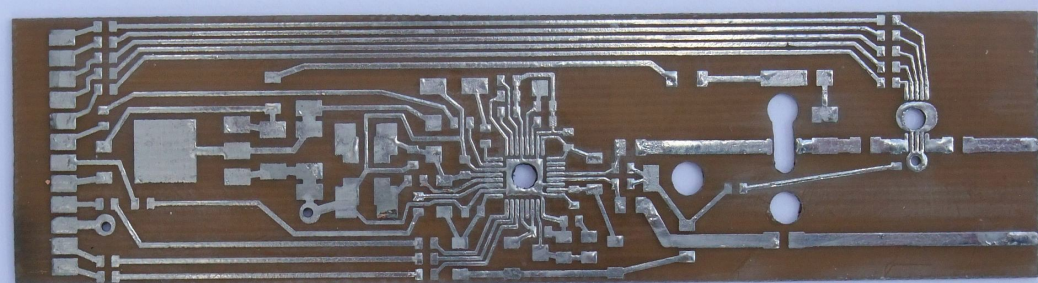


Frekvenčno območje 22MHz ... 6.4GHz

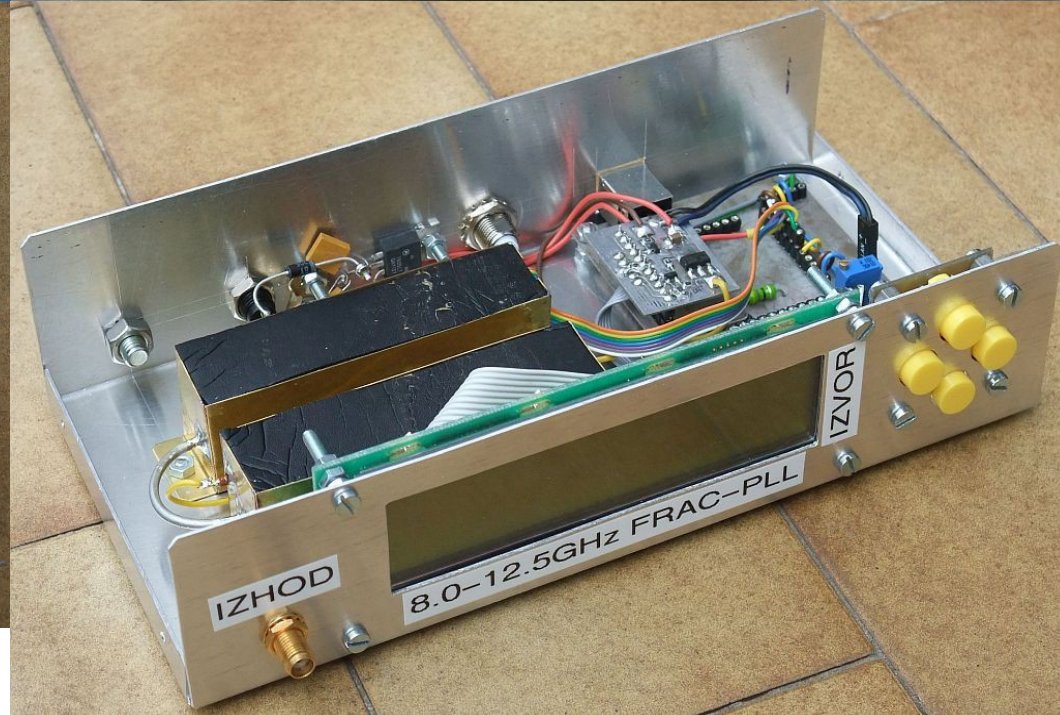
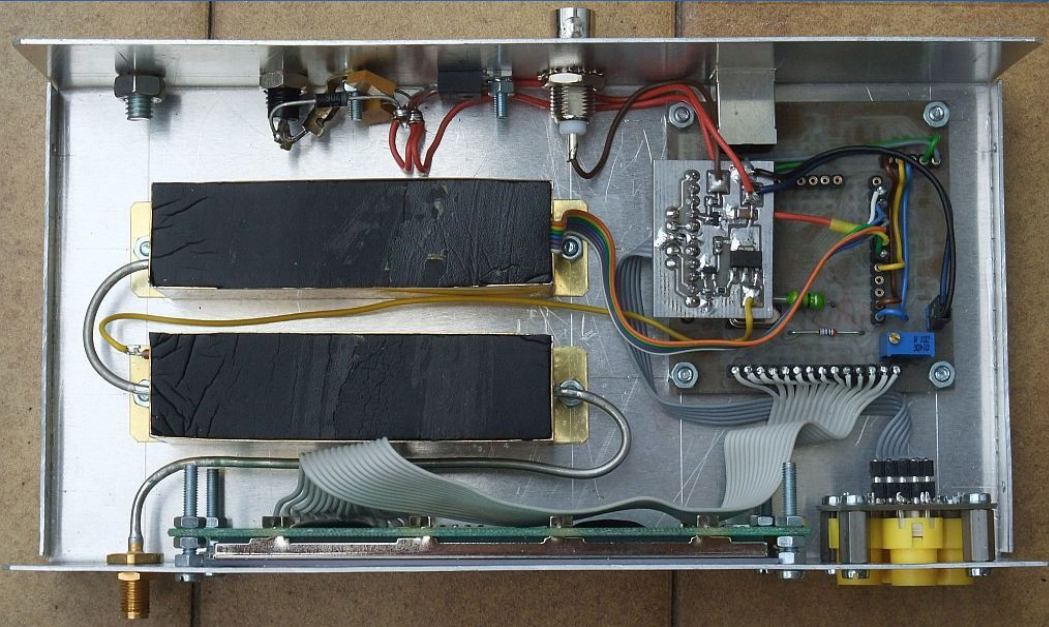
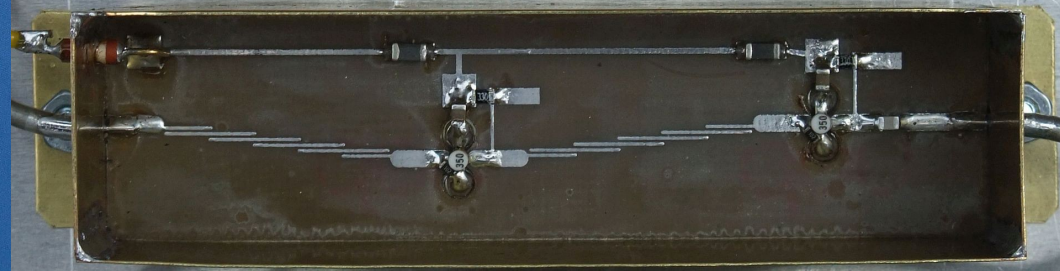
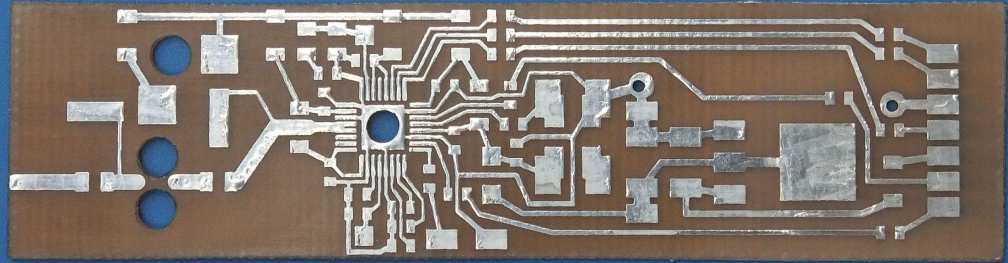
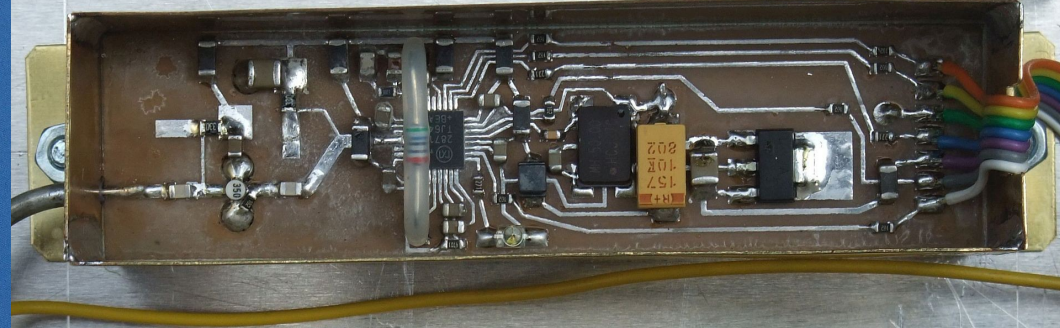
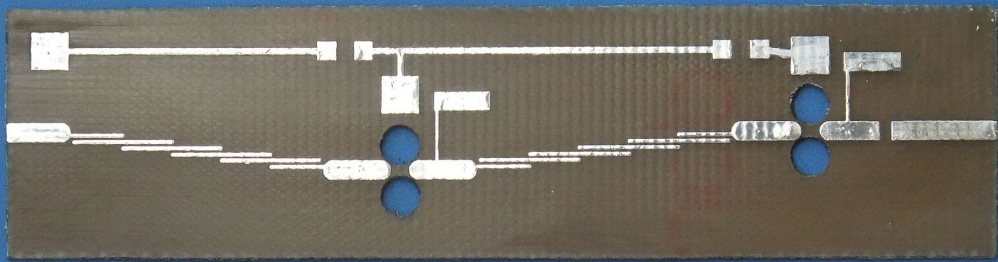


FRAC-PLL MAX2871
S53MV 19.04.2017

14 – Načrt merilnega izvora z MAX2871

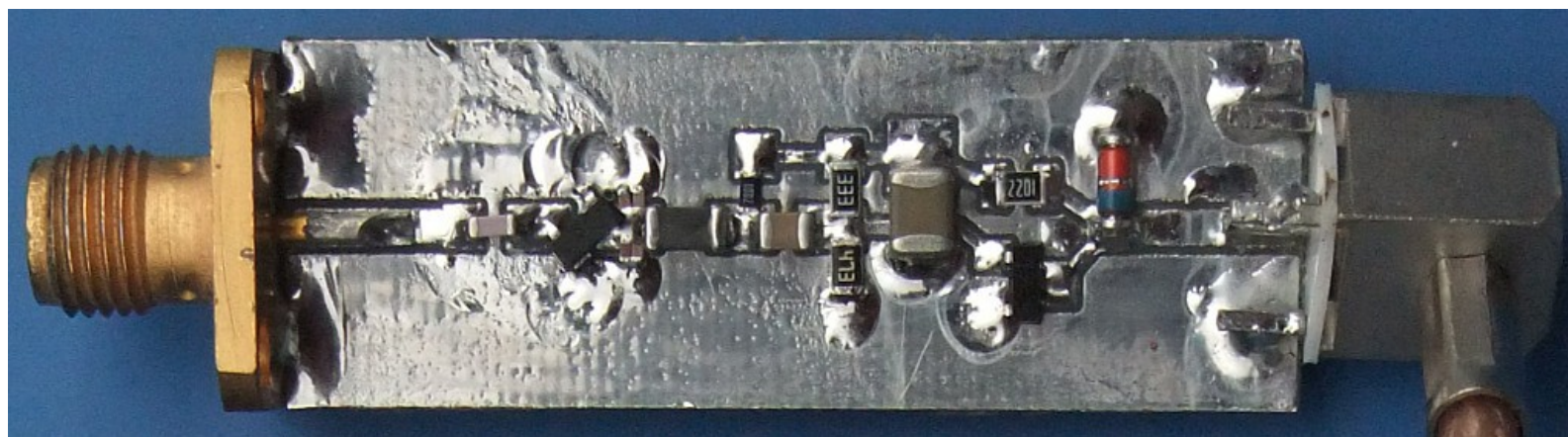
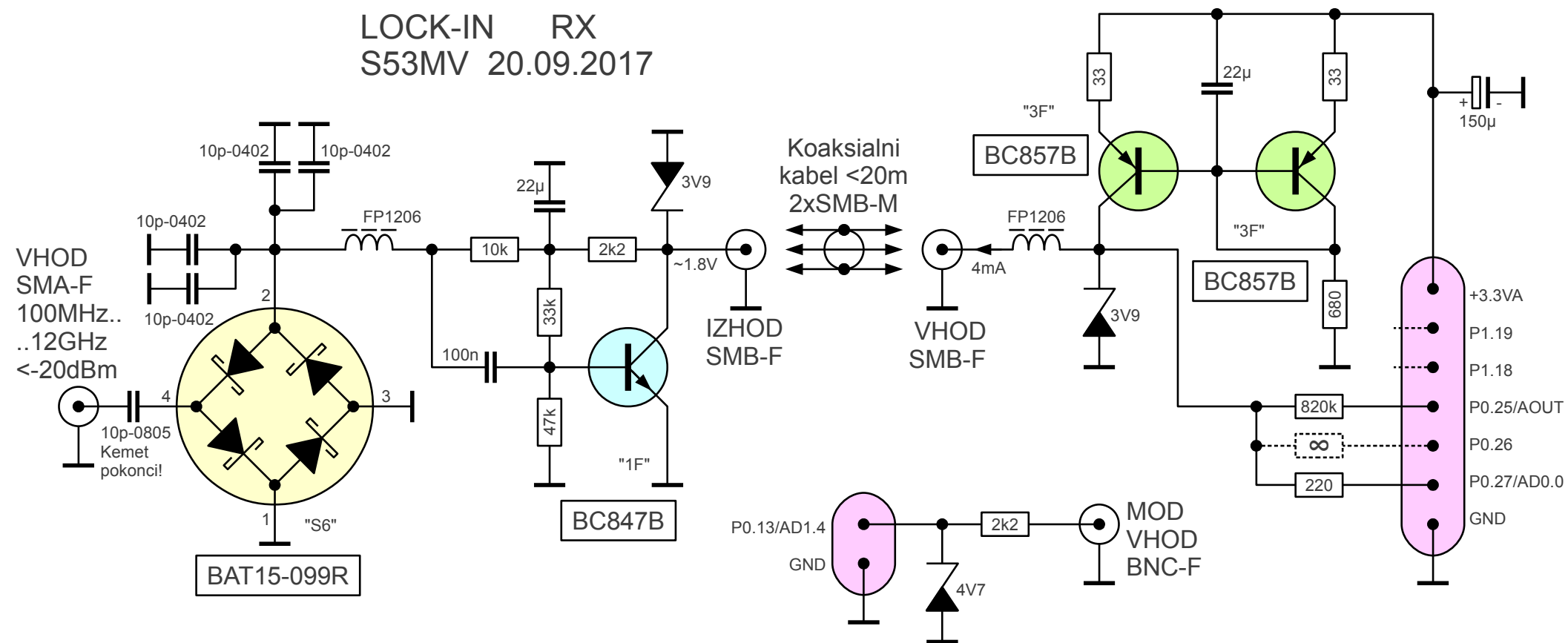


15 – Ulomkovni merilni izvor

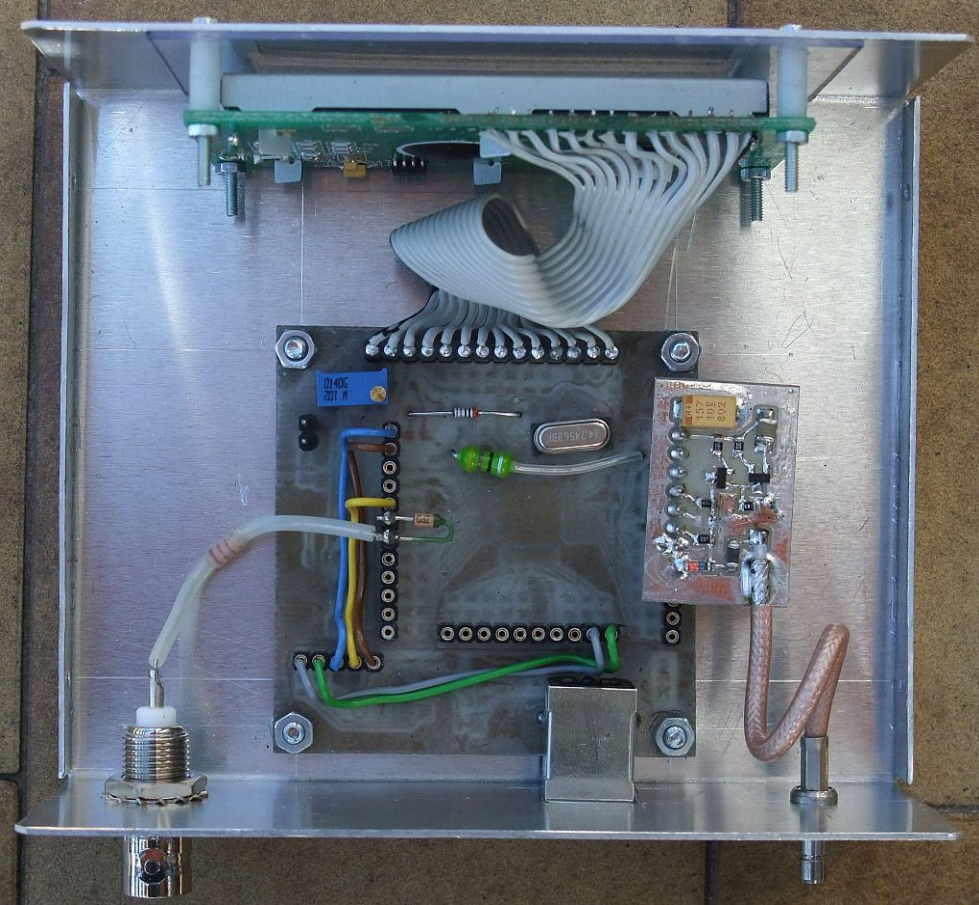
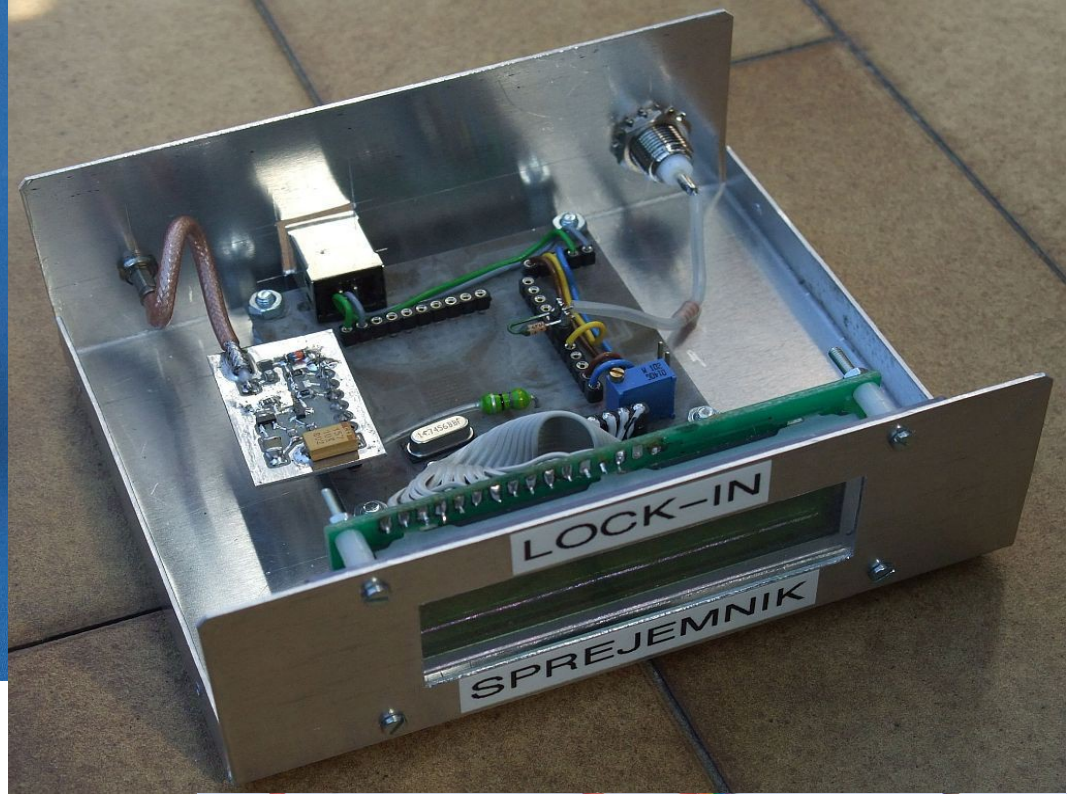
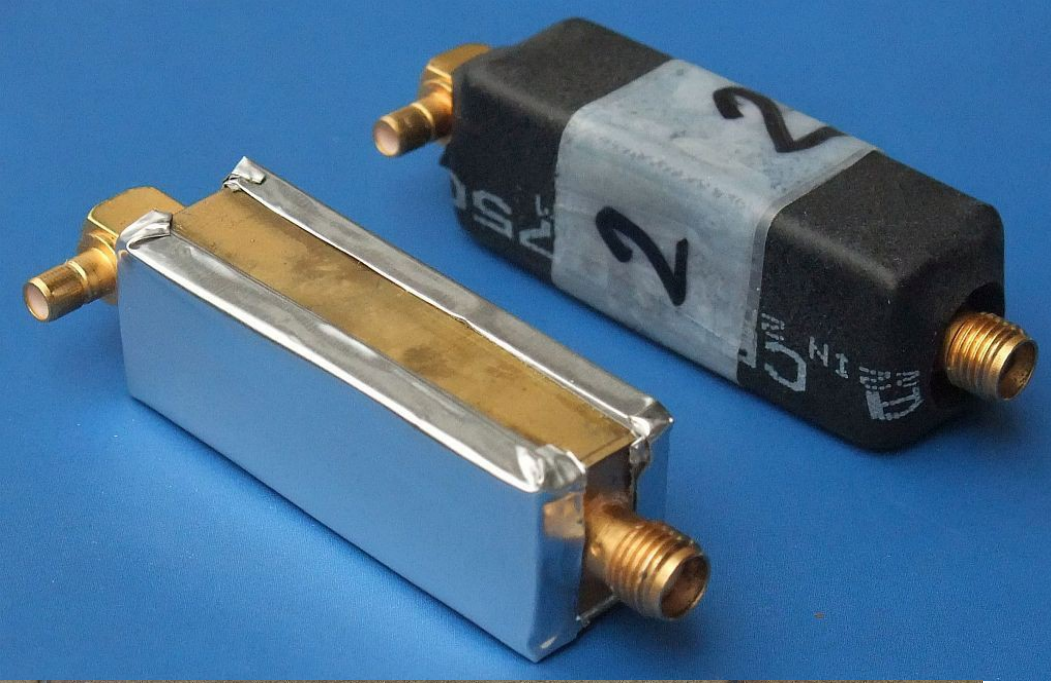


17 - Izvor 8..12GHz

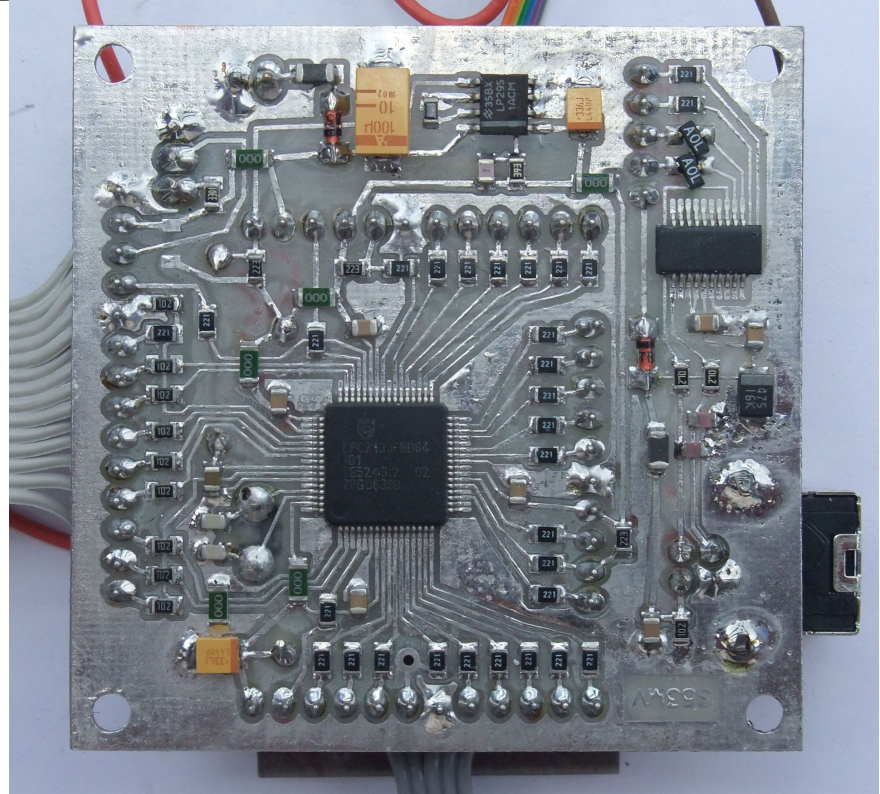
LOCK-IN RX
S53MV 20.09.2017



18 - 12GHz merilni detektor za lock-in sprejemnik



19 – Lock-in sprejemnik



Frekvenca prekinitev 230.4kHz

PRN izvor za tresenje

10bit ADC

javljanje napak (BITE)

HPF 1xIIR

sinhroni usmernik

LPF 2xIIR

Merilno območje
-70dBm .. -20dBm

Detektor s kvadratnim
odzivom zahteva
~100dB dinamike!

```
@*** Prekinitev FIQ *** (R7,R8-13.FIQ)
ADDS R13,R13,R13 @PRBS 1+x^28+x^31 v R13
MOVEQ R13,#-1
TST R13,#0x10000000
EORNE R13,R13,#1
TST R13,#0x80000000
EORNE R13,R13,#1
MOV R9,R13,LSR#22 @ocisti 10 bitov za DAC
MOV R9,R9,LSL#6
LDR R8,=0xE006C000 @DACR
STR R9,[R8]
LDR R8,=0xE0034004 @AD0GDR dioda
LDR R9,[R8]
ANDS R8,R9,#0x0000E000 @prekoracitev navzdol v R7.30?
ORREQ R7,R7,#0x40000000
EORS R8,R8,#0x0000E000 @prekoracitev navzgor v R7.31?
ORREQ R7,R7,#0x80000000
MOV R9,R9,LSL#16 @ocisti 10 bitov meritve
MOV R9,R9,LSR#22
MOV R9,R9,LSL#9 @mnozenje s 512
SUB R9,R9,R10,LSR#8 @visokoprepustno sito R10.unsigned
ADD R10,R10,R9
LDR R8,=0xE0060004 @AD1GDR lockin
LDR R8,[R8]
MOVS R8,R8,LSL#16 @lockin sinhroni usmernik z MSB(AD1)
RSBPL R9,R9,#0
ORRPL R7,R7,#0x10000000 @prisotnost nizko v R7.28?
ORRMI R7,R7,#0x20000000 @prisotnost visoko v R7.29?
SUB R9,R9,R11,ASR#12 @nizkoprepustno sito R11.signed
ADD R11,R11,R9
LDR R8,=0xE0034008 @ADGSR ponovno pozeni pretvorbo AD0 in AD1 hkrati
LDR R9,=0x01000000 @BURST=0,START=001,EDGE=0
STR R9,[R8]
MOV R9,R11,ASR#12 @nadaljevanje nizkoprepustno sito R11.signed
SUB R9,R9,R12,ASR#12 @nizkoprepustno sito R12.signed
ADD R12,R12,R9
MOV R9,R12,ASR#12
MOVS R9,R9 @nenegativen rezultat meritve v R7
MOVMI R9,#0
AND R7,R7,#0xF0000000 @ocisti spodnjih 28 bitov
ORR R7,R7,R9
LDR R8,=0xE0004000 @T0IR=0xFF reset zahtev TIMER0
MOV R9,#-1
STRB R9,[R8]
SUBS PC,LR,#4 @izhod iz prekinitve FIQ
```

21 – Programski jezik Python

- počasen interpreterski jezik (100...1000-krat počasnejši od "C")
- preprosto, kratko in jedrnato strukturirano programiranje
- brez deklaracije spremenljivk kjerkoli v programu
- odlična dokumentacija brez čarobnih formul kjerkoli v programu
- zelo širok nabor razširitev s preprosto vgradnjo, na primer knjižnice "matplotlib" za risanje, "pyserial" za RS-232 (tudi USB VCP)
- isti uporabniški program se enako izvaja na Windows/Linux/MAC
- enakovreden interpreter napisan v "C" za Windows/Linux/MAC
- odprtokodna zasnova (interpreter in vse razširitve so zastonj)
- dokazana dolgoročna podpora združljivosti
- naraščajoča podpora s strani proizvajalcev merilne opreme
- sodobna zamenjava za programski jezik BASIC za inženirje

```
#!/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: #iscis maksimum
            m=r
            kml=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)] #3601 vrstic podatkov: kot+jakost
    i=0
    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: #iscis maksimum
            m=r
            kml=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=(kml+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))+' dB'
    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:
                #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

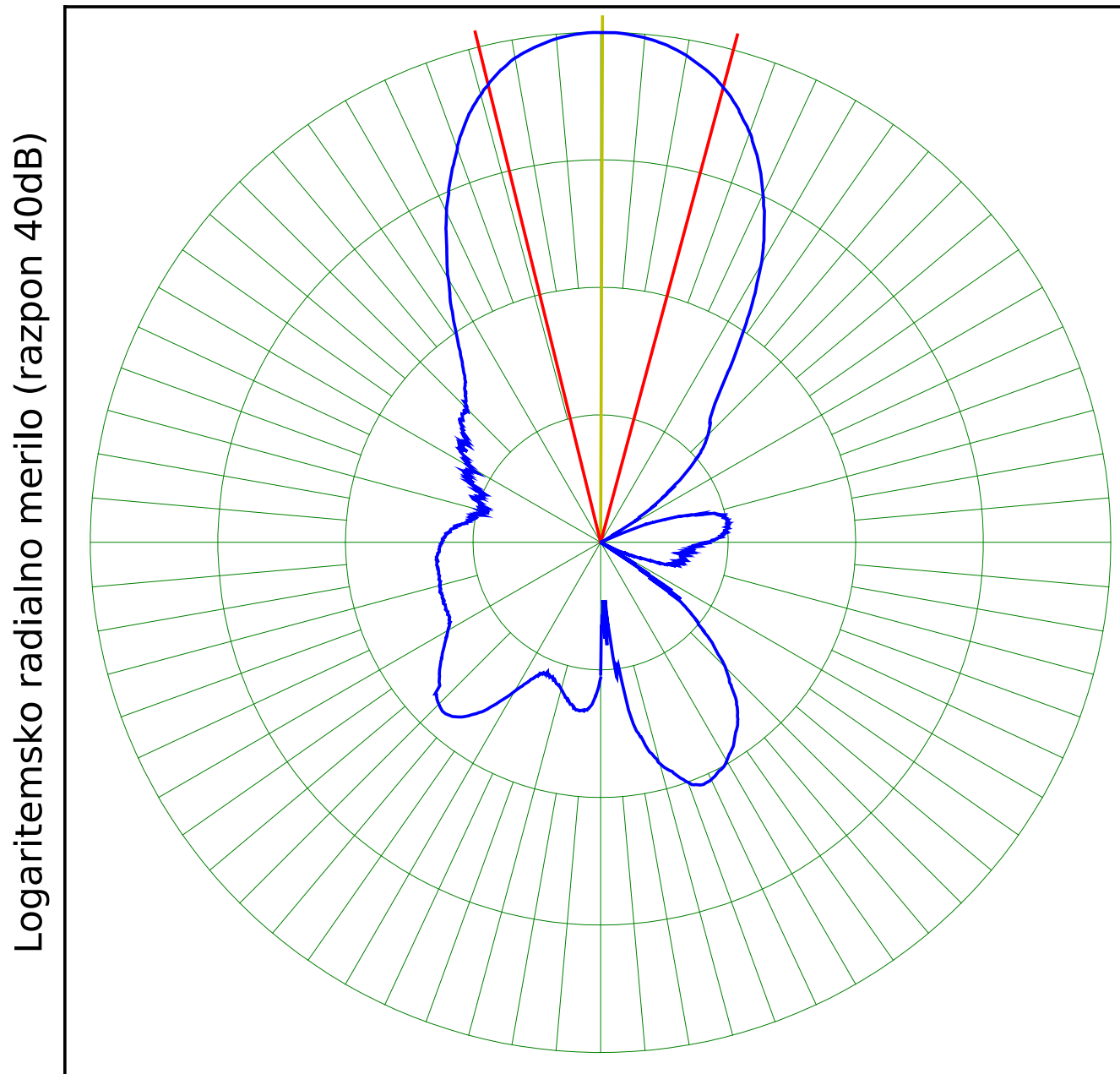
r=1.03 #narisi položaj maksimuma in sirino lista
plt.plot([0.0,math.sin(math.radians(km))*r],[0.0,-math.cos
(math.radians(km))*r], 'y-')
plt.plot([0.0,math.sin(math.radians(km1))*r],[0.0,-math.cos
(math.radians(km1))*r], 'r-')
plt.plot([0.0,math.sin(math.radians(km2))*r],[0.0,-math.cos
(math.radians(km2))*r], '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

```

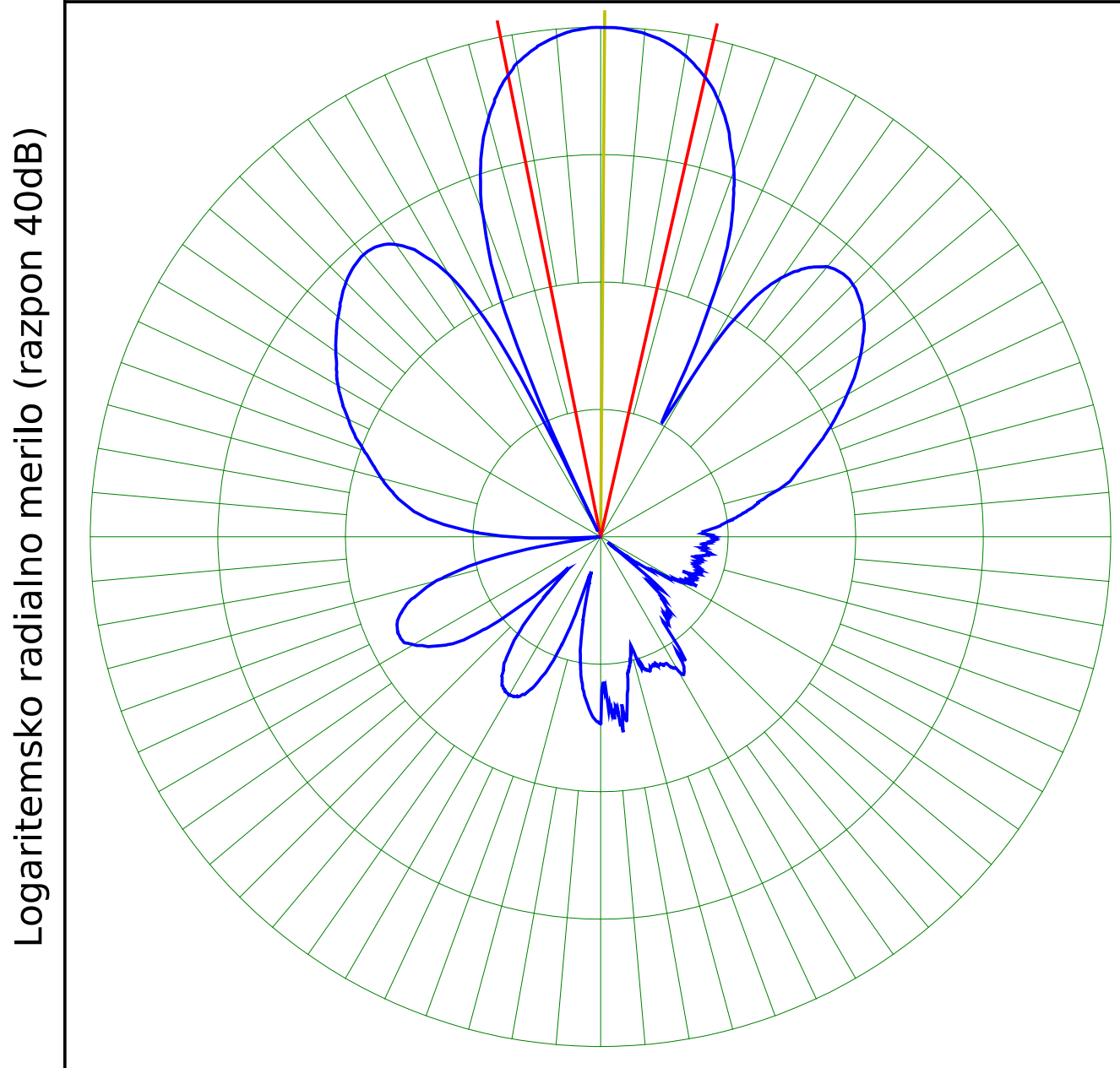
Sat Nov 25 19:46:46 2017
SBFA13 2360MHz ravnina E



Sirina -3dB: 28.9 Odklon: -0.2 Smernost: 44.8 = 16.51 dBi

24 – Izmerjeni smerni diagram v ravnini E

Sat Nov 25 19:37:17 2017
SBFA13 2360MHz ravnina H



Sirina -3dB: 24.1 Odklon: -0.45 Smernost: 41.4 = 16.17 dBi

25 – Izmerjeni smerni diagram v ravnini H