```
1
2
3   #Výpočet harmonického zkreslení pomocí Fourierovy trafsformace,
4   #tady z následujících dvou typů textových souborů (txt nebo csv):
5   #
6   #
7   # 1a)Textový soubor osciloskopu Hantek DSO4102C a jiných osciloskopů nebo
8   #    souborů jinak vytvořených, jejichž výstupní filé začíná takto:
9   %
10  %      #timebase=400000000(ps)
11  %      ,#voltbase=1000000(mv/100)
12  %      #size=20064
13  %      1.00E-06,-2.560
14  %      2.00E-06,-2.560
15  %      řádky s čísly pokračují a mají stejnou syntaxi až do konce filé
16  %
17  #    Pozn.: protože v oktáávě je znak pro poznámky taky čert (#) a       ↵
    první 3 řádky
18  #      ho mají v souboru na začátku, použil jsem zde u výpisu začátku    ↵
    souboru
19  #      pro poznámky znak %, v souboru oné řádky začínají znaky # a ,#
20  #      anebo čísly a žádný znak % s mezerama na začátku neobsahují.
21  #
22  #    Zde program předpokládá JEN záznam JEDNOHO kanálu!!
23  #
24  #    U souboru z osciloskopu Hantek DSO4102C program předpokládá, že     ↵
    obsahuje
25  #    VŽDY jen záznam JEDNOHO kanálu, i když samozřejmě osciloskop může   ↵
    mít
26  #    dva nebo čtyři kanály !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
27  #
28  #    Filé obsahuje na začátku 3 řádky začínající znaky # a ,# mají       ↵
    různé hodnoty
29  #    a používám (zatím) jen druhý, začínající ,#voltbase=...  to je
30  #    svislý rozsah osciloskopu (tedy přes celé stínítko) a po vynásobení
31  #    číslem 1e-5 je ve Voltech (ve filé je to v setinách milivoltu).
32  #
33  #    Další řádky odsahují data a jsou tu dva jako příklad, a zase v      ↵
    souboru
34  #    jsou až za mizerama s % označujícím zde poznámky tj. začínají       ↵
    číslicema.
35  #    Oddělovač hodnot je čárka, desetinnný oddělovač je tečka.
36  #    První číslo je čas vzorkování od začátku [v sevteřinách] a druhé    ↵
    číslo
37  #    (za čárkou) je změřená hodnota.
38  #
39  #    Pozn. k časům vzorkování: páč a jelikoš výstup do souboru u         ↵
    osciloskopu
40  #      Hantek DSO4102C asi dělali čičmundi a máťalové, udělali ho        ↵
    jenom na
41  #      3 platné číslice, což je konina, páč se v tom exponenciálním      ↵
    tvaru brzy
42  #      ty 3 číslice zaplní a pak má fůra hodnot stejný čas, takže z toho
43  #      kreslený gřaf je konina !!!!!!
44  #      Abych to odstranil dělám to zde tak, že z prvních dvou časů       ↵
    vypočítám
```

```
#      interval (tam to je ještě v pořádku, ovšem nevím co by to dělalo
#      kdyby interval byl větší než ty významné 3 číslice - musel by být
#      nějak na víc číslic než ty 3, a to nevím jak by bylo) a s tím
#      vypočteným intervalem počítám časy od prvního času znovu a vypočítané
#      časy používám místo časů přečtených ze souboru.
#
#      ANEBO gdyby někdo soubor z osciloskopu třeba v Poznámkovém bloku
#      apod. upravil tak, že by ořezal začátek až tam, gde už se výše
#      popsané omezení vyskytuje - to by se potom tento program nedopočítal
#      časování: když budou první dvě hodnoty času stejné tak program
#      špatně spočítá interval vzorkování (bude = 0) a to by byla nějaká
#      sranda!!!!!!
#
#
#
# 1b)Textový soubor generovaný jiným programem na základě souborů osciloskopu
#    Hantek DSO4102C, např. s vypočítaným signálem pro zkoušky programu.
#    Takový soubor začíná např. takto:
%
%      #timebase=19000000000(ps)
%      ,#voltbase=2.5(V); je simulováno rozlišení AD převodníku 12 bitů z rozsahu ±10 V
%      #size=19000
%      0,0.002442
%      1E-06,0.01709
%      2E-06,0.03175
%      řádky s čísly pokračují a mají stejnou syntaxi až do konce filé
%
#    anebo takto:
%
%      #timebase=1000000000(ps)
%      ,#voltbase=10(V); je simulováno výpočtem s max. možnou přesností simul. programu.
%      #size=19000
%      0,0
%      5.26315789473684E-08,0.00330693957508402
%      1.05263157894737E-07,0.0066138787885261
%      řádky s čísly pokračují a mají stejnou syntaxi až do konce filé
%
#    Poznámky k ukázce jsou stejné jako v bodě 1a), a může zde být také jen
#    jeden kanál.
#
#    Filé obsahuje na začátku 3 řádky začínající znaky # a ,# podobně jako
#    v bodě 1a), rozdíl je v textu přidaném za středníkem v druhém řádku.
#    Já jsem program na výrobu takového souboru udělal v MS Visual studiu,
#    text tam přidává automaticky.
#
#    Program simulaci hledá podle klíčového slova "simulováno" (bez uvozovek)
#    v druhém řádku a tedy pokud to je osciloskopem neměřený, ale jinak
```

```
92  #    generovaný signál MUSÍ tam toto slovo být včetně diakritiky, a potom
93  #    program předpokládá, že to simulace je.
94  #
95  #    Příklady simulace jsou tu dva:
96  #    - první, kde za klíčovým slovem je text obsahující mj. dvě čísla: ↵
    první
97  #         definuje na kolik bitů je kvantován (rozdělen) vypočítaný      ↵
    signál,
98  #         tedy jaké je simulováno rozlišení AD (analogově - digitálního)
99  #         převodníku, a druhé číslo udává rozsah, ze kterého se kvantuje
100 #         (rozsah AD převodníku). Tyto dvě čísla ve v příkladě           ↵
    uvedeném pořadí
101 #         (jejich pořadí je podmínka správného určení) program ze        ↵
    souboru čte
102 #         a vypisuje do výstupu. Jak je v příkladu viděti, první         ↵
    zaznamenaná
103 #         hodnota signálu není nula, a hodnoty jsou zaznamenány na       ↵
    menší počet
104 #         platných číslic než v druhém příkladě, protože v tomto         ↵
    případě bylo
105 #         simulováno rozlišení 12 bitů, tj. rozsah AD převodníku je      ↵
    rozdělen
106 #         na 4096 úrovní. Pozn.: při často používaném rozlišení AD       ↵
    převodníků
107 #         levnějších osciloskopů 8 bitů má signál jen 256 úrovní.
108 #
109 #    - druhý, kde za klíčovým slovem žádné číslo není, a proto program
110 #         předpokládá, že jde čistě o vypočítaný signál na max. přesnost
111 #         programu, který jej spočítal. Max. přesnost je vidět i na počtu
112 #         číslic, na který jsou hodnoty zapisovány.
113 #
114 #
115 #
116 #
117 # 2) Textový soubor osciloskopů OWON řady ADS800/900 a jiných           ↵
    osciloskopů nebo
118 #    souborů jinak vytvořených, jejichž výstupní filé začíná takto:
119 %
120 %      Model,ADS802A
121 %      Firmware Version,V1.0.1.7.1
122 %
123 %      Horizontal Units,s
124 %      TimeBase,0.0000100
125 %      Horizontal,0.0
126 %      Sample Interval,2.0E-7
127 %      Record Length,100000
128 %
129 %      Channel,CH1,CH2
130 %      Probe attenuation,1.0X,1.0X
131 %      Vertical Offset,0.00div,0.00div
132 %      vertical Scale,50.00mV,100.0mV
133 %      Label,?,?
134 %      Frequency,1.000kHz,872.7Hz
135 %      Period,1.000ms,1.145ms
136 %      PK-PK,316.5mV,409.0mV
137 %      Average,335.9µV,375.0µV
```

```
%      Vertical pos,0,0
%      Per ADC value,50.000000,100.000000
%
%      TIME(Units:s),CH1(Units:V),CH2(Units:V),
%      -0.0100000,0.003,-0.003
%      -0.0099998,0.004,-0.003
%      -0.0099996,0.003,-0.003
%      -0.0099994,0.002,-0.006
%      řádky s čísly pokračují a mají stejnou syntaxi až do konce filé
%
#    Pozn.: zde jsem opět použil pro poznámky u výpisu začátku souboru znak %,
#      v souboru řádky začínají textem anebo čísly a žádný znak % s mezerama
#      na začátku neobsahují.
#
#    Použitelnost programu pro celou řadu ADS 800/900 jen předpokládám,
#    protože mám osciloskop ADS802A a soubory z jiných osciloskopů této řady
#    jsem (zatím) neměl možnost vyzkoušet, ale snad budou stejné.
#
#    Zde může být jeden až deset kanálů, i když osciloskop tohoto typu
#    může mít max. čtyři analogové kanály. Program předpokládá, že data
#    jsou do souboru uložena jako typ "Wave" a jako zdroj byly nastaveny
#    libovolné kombinace kanálů, ale NE typy MATH, DIR a FFT, ty jsem zatím
#    nezkoušel. V příkladu je soubor se dvěma kanalízacemi.
#
#    Filé obsahuje na začátku 22 řádků s textem popisujícím typ a nastavení
#    a některé změřené hodnoty osciloskopu, jejichž význam je ze souboru
#    celkem jasný. Já (zatím) používám jen hodnoty "Channel,..." což je
#    identifikace zaznamenaných kanálů, "Model," což je model osciloskopu,
#    "(Units:" za tím jsou jednotky pro jednotlivé zaznamenané data
#    a  "vertical Scale,...", což je nastavená vertikální citlivost osciloskopu
#    v uvedených jednotkách na 1 vertikální dílek stínítka, a protože stínítko
#    má celkem 8 dílků vertikálně, tak tuto hodnotu po vynásobení 8 uvádím
#    ve výstupu programu pro osciloskopy OWON jako rozsah osciloskopu.
#
#    Další řádky vypadají následovně (jsou tu čtyři jako příklad, a zase
#    v souboru jsou až za mizerama s % označujícím zde poznámky tj. začínají
#    číslicema). Oddělovač hodnot je čárka, desetinnný oddělovač je tečka.
#    První číslo je čas vzorkování od začátku [v sevteřinách] a druhé a případná
#    další čísla (za čárkou) jsou změřené hodnoty jednotlivých kanálů.
#
#    Protože osciloskop umožňuje do souboru i zápis více kanálů ve formě dalších
#    přidaných hodnot na každém řádku (ADS802A/ADS812A/ADS822A jsou
#    dvoukanálové a ADS804A ADS814A ADS824A jsou čtyřkanálové), program
```

```
#    v případě, že zjistí v souboru kanálů víc, se zeptá který má zpracovat
#    a tento kanál se bude dále zpracovávat.
#
#
#    Pozn. k ZÁPISU DO .CSV FILÉ: jestliže jsem u osciloskopu HANTEK psal něco
#      o máťalech a pod., tak tady nevím jak to nazvat, protože se zápisem
#      do těchto filé (ale také do .zip a .mat) tady byly daleko větší potíže
#      které jsem reklamoval, a sice po dodání osciloskop do těchto souborů
#      zaznamenal dobře jen prvních 2047 dat (časy byly v pořádku), a potom
#      už to byly téměř konstantní náhodné hodnoty (konstantní ve smyslu
#      dotyčného souboru, v jiném souboru byly zase jiné). Z toho vyplynulo,
#      že "jedlý" byl JEN záznam do nejkratšího souboru o 1000 hodnotách
#      protože se to do uvedeného počtu zkrátka "vlezlo", delší soubory měly
#      "jedlých" jen prvních 2047 dat, což mě bylo nanic. Takže jsem asi
#      do týdne po dodání (tj. 17.11.2025) závadu reklamoval na dvou místech
#      (v Číně, protože jsem osciloskop koupil u firmy Banggood):
#
#       1) přímo u firmy Owon (i když tam jsem ho nekoupil, ale co kdyby...)
#          na obecné adrese z jejich stránek:
#              info@owon.com.cn
#          ovšem, jak jsem předpokládal, naprosto BEZ jakékoliv odezvy...
#          (tedy do 5.1.2026)
#
#       2) u firmy Banggood na stránce, kde jsem osciloskop koupil, samozřejmě
#          po přihlášení k mému účtu u nich:
#              https://www.banggood.com/cs/...
#          a tady byl přístup dost vstřícný, v podstatě když pominu některá
#          specifika lidí i nastavení jejich stránek, tak díky jejich firmě
#          jsem za dobu delší než jeden měsíc (24.12.2025) fungující upgrade
#          dostal. Ze "zvláštností" je asi nejdůležitější, že MUSÍ být
#          ze strany zákazníka nějaká aktivita během 7 DNÍ a když nebude,
#          (jako třeba v mém případě když jsem čekal na jejich reakci a to
#          i přesto, že psali, že mám počkat až pošlou upgrade a že to bude
#          chvíli trvat) tak KONVERZACI ZAVŘOU...  Pravda, můj osciloskop
#          měl při koupení verzi firmwaru osciloskopu V1.0.1.0.17
#          (s nefungujícím zápisem do csv) a v prvním kole mě poslali upgrade
#          na verzi V1.0.1.7.1 které se mi nepovedlo hned nainstalovat qůli
#          chybnému pojmenování zaslaného souboru (jak jsem soubor
```

```
dostal, tak
#          jsem ho zkusil instalovat, osciloskop upgrade poznal, ale po jeho
#          spuštění hlásil "Parse error") no a když se to po jeho přejmenování
#          povedlo, tak zápis do csv také nefungoval... Nakonec mě z Banggood
#          poslali emajlovou adresu na konkrétní paní z firmy OWON (nevím,
#          jestli ji tady můžu uvést), a této paní jsem 11. prasince 2025
#          poslal popis závady znovu a paní napřed, i když ne hned po poslání
#          emajlu odpověděla (16.12.2025), že to prošetří a že musím počkat,
#          no a potom toho 24.12.2025 mě poslala (jako dárek??) upgrade
#          na verzi V1.0.1.7.2 a to FUNGUJE, hurrrááá... (i když zase to má
#          nějaké "muchy", například, jak je vidět ze zde výše uvedeného výpisu
#          začátku csv souboru, je tam pořád hrdě verze firmwaru V1.0.1.7.1
#          a né V1.0.1.7.2 kterou to bylo nahráno...)
#
#
#
#  Typ osciloskopu program určuje JEN z počtu textových řádků na začátku filé,
#  mj. proto, že osciloskop HANTEK tam žádný model jako OWON neuvádí, takže
#  doufám, že případné upgrade osciloskopu toto neovlivní, ale vzhledem k tomu,
#  že za celou dobu co mám osciloskop HANTEK (asi 5 let) jsem nikde žadné
#  upgrade pro něj neviděl, a vzhledem k výše popsaným potížím se záznamem
#  do csv filé u osciloskopu OWON moc nepředpokládám, že by pro oba nějaké
#  up grade bylo v budoucnosti k dispozici.
#
#  Program obsahuje všechny potřebné funkce v sobě, takže nevolá žádné
#  funkce uložené v dalších souborech *.m
#
#  Pro svou činnost potřebuje mít zaveden balík signal protože z něho používá
#  některé funkce, já mám balík, tedy filé "signal-1.4.6.tar.gz" stažené z:
#          https://gnu-octave.github.io/packages/signal/
#
#  Program jsem začal vyrábět v Oktávě verze 10.2.0, tedy GNU Octave, je na:
#          https://octave.org/
#  vyzkoušen a dodělán byl (zatím) v ledenci 2026 v Oktávě verze 10.3.0.
#  Podle mého vyzkoušení NENÍ kompatibilní s MATLABem, je tam dost rozdílů...
#
#
#
```

```
global VerzeProg;
VerzeProg = '2. února 2026';  ###  Luděk Ruffer        lruffer@volny.cz
```

```
#KDYŽ na konci příkazu není střevník, oktáva zobrazí výstup do příkaz. ↵
qokna!!!
```

```
# Pozn NUF: aby jedle chodilo otevírání filé přes UIgetfilé, je potřeba
#    v hl. menu oktávy dát: EDIT/Preferences a tam v uchu Generál ↵
zafajfkovat
#    Use native filé dialogs. Jinak trvá načítání neimplicitních ↵
adresářů dost
#    dlouho.
```

```
pkg load signal;       #bez zavedení balíku signal není v Oktávě ani ↵
přístupná
                       #ani nápověda z něho, takže pro její ↵
                       zpřístupnění se musí
                       #tento program aspoň jednou spustiti......

close all;             #smaže fšecky figůůry
clear; clc;
output_precision(7);  #implicitně je na 5 míst
```

```
global Cestafile = ''; # určení typu jde snad jen přiřazením hodnoty, ↵
cocoti
                        # a viz poznámka tady u globál UzJePsFile = ...
```

```
global Iniplk = struct(); #vytvoření prázdné (staré) struktúry jako ↵
zobální
#a obsazení této staré struktúúry
Iniplk.zpracdat = 'Způsob zpracování dat = ';
Iniplk.cestafile = 'Poslední zpracovávaná cesta k souboru dat = ';
Iniplk.amplharmveV = 'Amplituda kreslených harmonických ve V (1 je V, ↵
0 je dB) = ';
Iniplk.pouzBT = 'Používat Butterworthovu dolní propust (ano, jiné je ↵
ne) = ';
```

```
303   Iniplk.fsmikBT = 'Kmitočet zlomu Butterworthovy DP [Hz] = ';
304   Iniplk.radBT = 'Řád Butterworthovy DP (rozsah 1 až 8, při jiném bude
      použit 2. řád) = ';
305   Iniplk.PocKrHarm = 'Počet kreslených harmonických = ';
306   Iniplk.PoziceFig = 'Pozice figury (okna) při ukončení programu = ';
307
308
309
310
311   global Jminifile AmplHarmV Velpistisk;
312
313
314
315
316   # proměnnná ZpracDat může mít následující hodnoty:
317   #
318   #   1 = "Data zpracovat jako celé kmity od prvního průchodu nulou"
319   #   2 = "Data zpracovat jako celé kmity přibližně od prvního maxima"
320   #   3 = "Data zpracovat jako celé kmity od začátku filé"
321   #   4 = "Data zpracovat jako celé kmity od začátku filé"
322   #
323   # předvolená hodnota (i při čtení inifilé) je 1
324   global ZpracDat = 1;
325   global VsechnyVybery = false;
326
327
328
329
330   global Fsmik RadBTfiltru JeBTfiltr;     #více proměnnných v řádku MUSÍ
      být BEZ
331                                           #oddělovačů (oddělených jen
                                              mizerami)
332
333
334   global Monitory Hlmonitor;
335
336
337   global Cestafile Jmfile hgrf1 hgrf2 men Velpis;
338
339
340   global rel_pos_leg rel_pos_leg_BT;
341
342
343   global Osciloskop = "";
344   global OscHAN = "HANTEK DSO4102C";
345   global OscOWO = "OWON ADS800/900";
346
347
348   # následuje vytvoření řetězcového vektoru - musí být ve složených
      zátvorkách
349   global plknastpod;
350   plknastpod = {"Data zpracovat jako celé kmity od prvního průchodu
      nulou" , ...
351                 "Data zpracovat jako celé kmity od prvního maxima" , ...
352                 "Data zpracovat jako celé kmity od začátku filé" , ...
353                 "Data zpracovat z celého filé"};
```

```
354
355
356  global PocKresHarm PocHarm FposlHarm Fvypis;
357  PocKresHarm = 200;  # preparát, dyby v ini filé nebylo
358
359  global JednHlWosyX JednHlWosyY JednWY;
360
361
362  global HlavPlk = 'Výpočet harmonického zkreslení (THD) pomocí diskrétní Fourierovy transformace';
363  global hfig1;
364
365
366  global TlacimFiguru;
367  TlacimFiguru = false;
368
369
370  global ps_file;
371
372  # Gdyš je takto deklarováno a hned v deklaraci je přiřazena hodnota,
373  # tak se hodnota přiřadí JEN při spuštění oktávového prostředí a při
374  # dalším spouštění přes F5 se už tady nepřiřazuje, ale kdyby byla hodnota
375  # tady přiřazována až v dalším řádku tak se hodnota přiřadí při každém
376  # spuštení (třeba přes F5) - tak jako jinde v programu při jeho běhu:
377  global UzJePsFile = false; # takto schválně, páč PS filé se maže při ukončování
378                             # celé oktáávy, aby zbytečně nepsalo na HD
379
380
381
382  global PocVolTam PocVolVen;
383  PocVolTam = 0;
384  PocVolVen = 0;
385
386
387  global ZaklVelFig = struct(); #vytvoření prázdné (staré) struktúry jako zobální
388  #a obsazení staré struktúúry - násobky pro výpočet zákl. velikosti figúry
389  ZaklVelFig.X = 0.01;
390  ZaklVelFig.Y = 0.06;
391  ZaklVelFig.Sir = 0.97;
392  ZaklVelFig.Vys = 0.87;
393
394
395
396
397
398
399
400
401
402
403
404
405
406  #++++++ funkce volané z událostí
```

```
++++++++++++++++++++++++++++++++++++++++++++++
407
408
409
410 #.....
.......
411 function Oktavasezavira;
412    # Funkce, se volá při zavírání celé oktáávy.
413    # Je tu qůli výmazu filé Póówlšelllu, které se používá k indikaci,
   jeli
414    # aktiwní vokno maxima lízováno. Filé mazat AŽ PŘI UKONČOVÁNÍ OKTÁÁVY,
415    # protože kdyby se mazalo při ukončování figúúry, bylo by to jednak
416    # zbytečně často, a druhak a HLAVNĚ by se taky musela nějak vypucovat
417    # inicializační proměnná v podprogramu
   je_maximalizov_okno_s_focusem(),
418    # páč bez výpucu zústane true a při dalším otevření figůry se filé
   nevytvoří
419    # a podprogram ho taky nenajde, takže podprogram projde, bude hlásit
420    # chybu a tím pádem se maximalízace okna bude určovat JEN z
   vodorovného
421    # rozměru, což nemusí vždy odpovídat...
422
423
424    global ps_file;
425
426
427    try
428      delete(ps_file);
429    end_try_catch
430
431 endfunction#.....
.......
432
433
434
435
436
437
438 #.....
.......
439 function Figurasezavira(src);
440    # Funkce, se zavolá při zavírání objektu s handlem src
441
442
443    #POZOR:
444    # gdyš se tato fuňkce volala z definice příkazem set(... a měla jako
445    # předávané párametry ještě nějaké jiné než src (byly to Jminifile,
   velobr, xrozl
446    # tak se sem předávaly jejich hodnoty které byly v okamžiku volání
447    # přikazu set(... a nebralo to ohled na jejich pozdější změny!!!!!
448    # Tady to vadilo u xrozl, páč ta se může měniti, ostatní dvě asi néé.
449    # Takže jsem je udělal všechny preparát zobální a funguje to.
450    #pohof.
451
452
453    global Jminifile;
```

```
454
455
456
457      UlozInifile (Jminifile);
458
459
460
461      #POZOR:
462      #  jeli volání této udalosti aktívováno, MUSÍ se tady objekt s handle src
463      #  taky zavříti, když tu zavření nebude tak zůstane otevřený a v Oktávě
464      #  potom nejde jinak zavřít než s celou Oktávou !!!!!
465      #pohof.
466
467      try
468        delete(src); # Zavření oného, musí být delete, close zde jaksi nefunguje
469      catch
470        # praparát, ale dyš to zblbne tak zmenšenou figůru stejně nesmaže...
471        close all;            #smaže fšecky figůůry
472      end
473
474
475    endfunction#.....................................................
       ........
476
477
478
479
480    #.....................................................................
       ........
481    function volani_tlac_plkboxu(~, ~)  #prázdný vstupy tam musí být, jinak řve botu
482
483        #Zabudovaná fuňkce gcbf ():
484        #Vrátí handle k figure obsahují objekt, jehož zpětné volání se právě provádí.
485
486
487        # Zavření figuury při stisknutí tlačítka
488        uiresume(gcbf); # Pokračuje v provádění po stisknutí tlačítka
489
490    endfunction#.....................................................
       ........
491
492
493
494
495
496    #.....................................................................
       ........
497    function omezeni_rozmeru(hzdroj,~)
498
499      # Omezení rozměrú figúry s handle hzdroj tak aby byly v nějakých mezích
500      # kdy ještě nejsou plky přes sebe (miminum) anebo aby nebylo přes dva
```

```
501     # monitory, dyš sou (maximumo). Funkce se volá následovně:
502     #     set(hFig, 'sizechangedfcn', @(h,ev) omezeni_rozmeru(h));
503     # kde hFig je handle dotyčné figúry (tady je to hzdroj) a to způsobí,
504     # že je volána (většinou vícekrát) když probíhá změna velikosti oné
        figúry.
505     # L. Ruffer
506
507
508
509     global TlacimFiguru Hlmonitor;
510     global PocVolTam PocVolVen;
511
512
513     PocVolTam++;
514
515     # Funkce volá sama sebe (rekurze) třeba při provedení příkazu k
        nastavení
516     # velikosti figúry set(hzdroj,'position',pos) a možná i při změně
        jednotek
517     # a při takové rekurzi hrozí zacyklování, takže aby se funkce
        nezacyklovala
518     # používám tady (po pokusech s více všelijakými možnostmi) počítání
519     # vstupů do funkce a výstupú z funkce do dvou proměnných PocVolTam
520     # a PocVolVen deklarovaných jako global v hlavním programu (aby byly
        jen
521     # jednou v paměti a viděly na ně všechny instance volané funkce)
522     # a vyskakuje se až při rovnosti vstupů a výstupú (u výstupu se
        proměnná
523     # samozřejmě zvýší když už je jasné, že bude následovat výskok a
        poté se
524     # rovnost vyhodnocuje). K provedení více čekajících instancí
        podprogramu
525     # používám příkaz pause(jeho popis je kousek dáál), ten způsobí mj.
        provedení
526     # nějakých čekajících volání tohoto podprogramu, ale nevím jak zjistit
527     # příkazem oktávy kolik volání čeká, a kolík už vyskočilo na pause a
        pod.
528     # a když jsem tady začal vyskakovat pro zbytečná volání dříf,
        nedařilo se
529     # mě dopočítat stejného počtu výskoků protože jsem měl příkaz pause
        ještě
530     # před přidáním do proměnné počítající vstupy. Takže přidávání počtu
        vstupů
531     # (proměnná PocVolTam) MUSÍ být před voláním příkazu pause() a
        jiných na
532     # které to vyskakuje (mj. taky drawnow), páč na ně to vyskakuje což
533     # samozřejmě způsobí další zavolání tohoto podprogramu, ve kterém by se
534     # počet vstupů už nepřičetl a byl by v tom boordéél...
535     if (PocVolTam - PocVolVen) > 1
536       PocVolVen++;
537       return;
538     endif
539
540
541     # Protože výstup (tisknutí) figúry do .jpg volá taky změnu rozměrů
542     # a provedení oného dál dělalo boordéél (nevytiskla se figura do
```

```
      obrazového
543   # filé ve fuňkci Ulozdofile - tamější příkaz print), tak je tu
      následující
544   # výskok přes TlacimFiguru když je tisk do .jpg
545   if TlacimFiguru
546     PocVolVen++;
547     return;
548   endif
549
550
551
552
553
554   # Dyš je maxima líízace tak véén bez ničeho a hned, aby maxima blízace
555   # zústala a funkce nechala figúúru maximalízovanou.
556   [Bota, JeMaximaBliizace] = je_maximalizov_okno_s_focusem();
557   if length(Bota)>0
558     # Ošetření případné obuvi při zjišťování maxima blízace
      PowerodeŠelem.
559     #
560     # Tady aspoň zjištěním jeli šířka wokna zhruba stejná jako šířka
      monitoru,
561     # oboje v logických pixelech a jen šířka je tu proto, že aspoň u W11
562     # jakštakš sedí (W11 by už neměly mít možnost přemísťovat hlavní
563     # panel), ale u W10 to bude fungovat jen když bude hl. panel doole,
564     # no aspoň ňěco...
565     posmaxim = round(get(hzdroj,'position'));
566
567     # Zdá se, že stačí tole rance 7 pikslí:
568     JeMaxSir = abs(posmaxim(3) - Hlmonitor.sirkaLogic) <= 7;
569     JeMaximaBliizace = JeMaxSir;
570
571
572     # a výstup chybových plků aspoň do Commmand okna
573     printf('\n\nPři zjišťování, jeli figure maximalízována se vyskytla
      chyba:');
574     printf('\n\n%s', Bota);
575     printf('\n\nKonec textu chyby ----------------------------%s',
      newline);
576   endif
577   if JeMaximaBliizace
578     PocVolTam = 0;
579     PocVolVen = 0;
580     return;
581   endif
582
583
584
585
586
587
588
589   # Protože jsem nikde nenašel možnost (ani přes IÁÁ) říct oktávě, že
      má fčíil
590   # udělat všechny příkazy ve frontě (stack?) je to tu uděláno přes
      fuňkci
```

```
591     # pause(sevteřiny), protože v dokumentaci oktávy se k pause praví:
        "I když
592     # je program pozastaven, Octave stále zpracovává vykreslování obrázků
593     # a provádění grafických zpětných volání". Ovšem podle mých pokusů
        musí mít
594     # pauza zadaný nějaký čas aby se to provedlo, při kratším neudělá nic.
595     # A otázka je, jestli ten čas není závislý taky na rychlosti daného
596     # počítače, to by pak někde můj čas tady použitý nemusel stačit, ale
        zase
597     # to nechci mít zapauzované dlouho, aby to zbytečně neotravovalo...
598     pause(0.2);
599
600
601
602
603
604
605
606     # Načtení limitů uložených v figúre v ÚserData.
607     # Je to v PIKSLÍCH , protože v programu jsou jednotky figúry
608     # nastaveny na PIKSLE.
609     # Tedy kromě přepnutí normalized - piksle při posledním volání, jehož
610     # důvod je popsán u něj.
611     s = get (hzdroj, 'UserData');
612
613
614     # Pozici pro nastavení zjistiti až tady (a znovu), aby zjištění bylo
615     # až po výskocích - může se lišit od zjištění na začátku podprogramu...
616     pos = round(get(hzdroj,'position'));
617     sir = pos(3); vys = pos(4);
618
619
620     # Omezení hodnot na povolený rozsah (round proto, že piksely jsou
        celá čísla
621     # a nejsou možné desetiny, pak by se s celým číslem v pos() dále
        nerovnalo)
622     Omez_sir = round(min(max(sir, s.minSir), s.maxSir));
623     Omez_vys = round(min(max(vys, s.minVys), s.maxVys));
624
625
626     # qůli níže popisovanému blbnutí oktáávy (nevím proč, možná blbnu já)
627     # je přidání a vynulování následujících oných tady, ocaď to už stejně
628     # vyskočí
629     PocVolVen++;
630     if PocVolTam == PocVolVen
631       PocVolTam = 0;
632       PocVolVen = 0;
633
634       # Někdy se stane (asi gdyš je myš při tažení pro změnu rozměrů mimo
635       # miminální rozměry figůůry - nahoře a v levo) že se figůůra
636       # nenastaví i když příkaz set má pole rozměrů pos() jedlé a zůstane
637       # malá (někdy v obou rozměrech a někdy jen v jednom podle pozice
          myši)
638       # no a zdálo se mě (i kdyš jsem nespal) že by mohla pomoct
          následující
639       # změna jednotek (asi musí být obě změny), ale nepomůže vždy...
```

```
640       set (hzdroj, 'units','normalized');   # normalízace
641       set (hzdroj, 'units','pixels');
642     endif
643
644
645
646     MimoVobrazofku = false;
647     # korekce začátku na výšku dyš by wrch byl někde mimo vobražofku
        nebo dyby
648     # bylo víc jak 3/4 wokna pod wobra žofkou
649     if (pos(2) + pos(4) > s.maxVys) || (pos(2) + pos(4)*0.75 < s.zacY)
650       MimoVobrazofku = true;
651     endif
652
653     #korekce začátku na šířku dyš by bylo víc jak 3/4 okna mimo vobražofku
654     if (pos(1) + pos(3)*0.25 > s.maxSir) || (pos(1) + pos(3)*0.75 < s.zacX)
655       MimoVobrazofku = true;
656     endif
657
658
659
660     if Omez_sir ~= sir || Omez_vys ~= vys || MimoVobrazofku
661
662       if Omez_sir ~= sir || Omez_vys ~= vys
663         pos(3) = Omez_sir;
664         pos(4) = Omez_vys;
665       endif
666
667
668
669       if pos(2) + pos(4)*0.75 < s.zacY
670         pos(2) = s.zacY;
671       elseif (pos(2) + pos(4) > s.maxVys)
672         pos(2) = s.maxVys - pos(4);
673       endif
674
675       if pos(1) + pos(3) * 0.75 < s.zacX
676         pos(1) = s.zacX;
677       elseif (pos(1) + pos(3) * 0.25 > s.maxSir)
678         pos(1) = s.maxSir - pos(3);
679       endif
680
681
682       set(hzdroj,'position',pos);
683       drawnow ("expose");
684
685
686       # protože změna rozměrů figuury někdy umístí legendy gřafu doprčic
          (někam
687       # mimo původní umístění nebo i mimo gřaf a pak nejsou viděti) je tu
688       # volání následující fuňkce aby se umístění legend předělalo
689       try
690         umisti_legendy(hzdroj)
691       end
692
693     endif
```

```
694
695
696 endfunction#.....
    ........
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726 #+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    +++++++
727 #++++++ funkce volané ocaď                                              
    ++++++++++++++++++++++++++++++++++++++++++++++++++
728
729
730
731
732
733
734 #.....
    ........
735 function [Bota, vystup] = je_maximalizov_okno_s_focusem();
736     # od IÁÁ Claude Haiku 4.5 ze dne 29.1.2026 - upr. ŇUF
737
738
739     % Funkce zjišťuje, zda je AKTIVNÍ okno (s fokusem) maximalizováno
740     % (optimalizovaná verze - soubor se vytvoří jen jednou)
741     %
742     % Syntaxe:
743     %  [Bota, vystup] = je_maximalizov_okno_s_focusem()
744     %
745     % Výstup:
```

```
746       %   vystup - logická hodnota (true/false)
747       %            true = okno je maximalizováno
748       %            false = okno není maximalizováno
749       #   Bota -  v případě průběhu bez chyby je prázdné (length(Bota) = 0)
750       #           v případě chyby je tam chybové hlášení (text)
751       %
752       % Poznámka:
753       %   Soubor se vytvoří jen při prvním volání a zůstane v tempdir
754       %   Vhodné pro časté volání (např. z resize eventů)
755       #
756       # Aby v temp adresáři soubor nesmrďéél tak na konci hl. programu se má
757       # smazat, tady je to uděláno na začátku hl. programu definicí funkce,
758       # která se spustí při ukončování celé oktáávy, a to příkazem:
759       #   atexit('jméno oné funkce')
760
761       global ps_file UzJePsFile;
762
763
764
765       Bota = '';
766
767       # Inicializace jen při prvním volání po spuštění Oktááve
768       # (je zajištěno globáál deklarací na začátku v hl. programu)
769       if UzJePsFile == false
770           % Vytvoření dočasného souboru (jen jednou)
771           temp_dir = tempdir();
772           ps_file = fullfile(temp_dir, 'Octave-je_maximalizov_okno_s_focusem.ps1');
773
774           % PowerShell skript
775           ps_code = sprintf([...
776               'Add-Type @"\n' ...
777               'using System;\n' ...
778               'using System.Runtime.InteropServices;\n' ...
779               'public class WindowState {\n' ...
780               '    [DllImport("user32.dll", SetLastError = true)]\n' ...
781               '    public static extern IntPtr GetForegroundWindow();\n' ...
782               '    [DllImport("user32.dll", SetLastError = true)]\n' ...
783               '    public static extern bool IsZoomed(IntPtr hWnd);\n' ...
784               '}\n' ...
785               '"@\n' ...
786               '$hWnd = [WindowState]::GetForegroundWindow()\n' ...
787               'if ($hWnd -ne 0) {\n' ...
788               '    if ([WindowState]::IsZoomed($hWnd)) {\n' ...
789               '        Write-Host "1"\n' ...
790               '    } else {\n' ...
791               '        Write-Host "0"\n' ...
792               '    }\n' ...
793               '} else {\n' ...
794               '    Write-Host "0"\n' ...
795               '}\n']);
796
797           % Zápis skriptu do souboru (jen jednou)
798           try
```

```
799             fid = fopen(ps_file, 'w');
800             fprintf(fid, '%s', ps_code);
801             fclose(fid);
802             UzJePsFile = true;
803         catch exception
804             Bota = cstrcat('Při vytváření souboru je chyba fuňkce:
                je_maximalizov_okno_s_focusem %s', exception.message);
805             UzJePsFile = false;
806             vystup = false;
807             return;
808         end
809     end
810
811     try
812         % Spuštění PowerShell skriptu (bez zápisu - jen čtení a spuštění)
813         cmd = sprintf('powershell.exe -NoProfile -ExecutionPolicy
            Bypass -File "%s"', ps_file);
814         [status, output] = system(cmd);
815
816         % Zpracování výstupu
817         if length(strtrim(output)) > 1
818           # je obuf, jedlá odpověď je 0 nebo 1, při chybě je tam
              chybový plk
819           Bota = strtrim(output);
820           vystup = false;
821         else
822           output = strtrim(output);
823           vystup = strcmp(output, '1');
824         endif
825
826     catch exception
827         % V případě chyby vrátit false
828         Bota = cstrcat('Chyba fuňkce: je_maximalizov_okno_s_focusem
            %s', exception.message);
829         vystup = false;
830     end
831
832 endfunction#.....................................................
    ........
833
834
835
836
837
838
839
840
841
842
843
844 #.....................................................................
    ........
845 function [Bota, Monitory] = Zjisti_monitory()
846
847   # načtení hodnot o monitourech jinak, dyš to oktááva neumí, ale jen
      we Voknech
```

```
848
849     # deklarace prázdného oného
850     Monitory = struct('primarni', {}, 'sirkaSkut', {}, 'vyskaSkut', {}, ...
851                       'zvetseni_proc', {}, 'sirkaLogic', {},
                          'vyskaLogic', {},...
852                       'rucni_vstup', {});
853
854
855     Bota='';
856     [Bota, monLogicke] = Zjisti_monitory_PS();
857     if length(Bota)>0, return; end     #je obuf
858
859     [Bota, DPI_ef] = Zjisti_DPI_ef();
860     if length(Bota)>0, return; end     #je obuf
861
862
863     for i=1 : numel(monLogicke)
864       Monitory(i).sirkaLogic = monLogicke(i).widthLogic;
865       Monitory(i).vyskaLogic = monLogicke(i).heightLogic;
866       Monitory(i).zvetseni_proc = (DPI_ef(i,1)/96)*100;
867       Monitory(i).primarni = monLogicke(i).primary;
868
869       # oboje zaokrouhlit podle pravidel zaokrouhlování, round kladné
          hodnoty
870       # okrrouhlí takto: 2.4 = 2; 2.6 = 3, záporné tu snad nebudou
871       Monitory(i).sirkaSkut = round(Monitory(i).sirkaLogic * ...
872                                Monitory(i).zvetseni_proc/100);
873       #výšku preparát brát z Y
874       Monitory(i).vyskaSkut = round(Monitory(i).vyskaLogic *
          (DPI_ef(i,2)/96));
875
876       Monitory(i).rucni_vstup = 0;
877     endfor
878
879   endfunction#.....................................................
    ........
880
881
882
883
884
885   #.....................................................................
    ........
886   function [Bota, monitors] = Zjisti_monitory_PS()
887
888     # Funkce pro přečtení rozlišení displejů počítače vytvořená IÁÁ, éé
        pardon AI
889     # GPT-5 mimi a voláním příkazu Poweršellu 11.1.2026.
890
891     # Vrátí pole struktur s poli: deviceName, widthLogic, heightLogic,
        x, y, primary
892
893     # Vrací rozlišení v tzv. "logických" pikslech, a pokud je we voknech
894     # na oném (nebo oných) displjejích nastaveno nějaké zvětšení, tedy
895     # měřítko větší jak 100% bude rozlišení v logických pikslech
        přiměřeně menší.
```

```
896     # Wokenice používají jako 100% rozlišení 96 DPI, tedy mělo by být:
897     #   96 DPI = 100%, 120 DPI = 125%, 144 DPI = 150% a to tady čte fuňkce
898     # Zjisti_DPI_ef().
899     # Já mám na hl. monitoru we woknech nastaveno zvětšení 150%, takže
        pro tento
900     # monitor tato funkce vrátí X rozlišení 3840/1,5 = 2560 logických
        pikslí
901     # (což vrací) a pro druhý monitor mám zvětšení 125%, takže 2560/1,25
        = 2048
902     # log. pikslí atd.
903
904
905     # Výpis z podprogramu do Command wokna woktáwy se tam, odkud se volá
906     # (NEE TADY) dá udělat třeba takto (tady MUSÍ být zapoznámkováno):
907     #    [Bota, mon] = Zjisti_monitory_PS();
908     #      for i = 1:length(mon)
909     #        fprintf('Monitor %d (%s): %dx%d at (%d,%d) primary=%d\n',
        i, ...
910     #                mon(i).deviceName, mon(i).widthLogic,
        mon(i).heightLogic, ...
911     #                mon(i).x, mon(i).y, mon(i).primary);
912     #      end
913
914
915     Bota = '';
916
917     monitors = struct('deviceName', {}, 'widthLogic', {}, 'heightLogic',
        {}, ...
918                       'x', {}, 'y', {}, 'primary', {});
919
920     % PowerShell příkaz: použije System.Windows.Forms.Screen
921     ps_cmd = [
922       'powershell -NoProfile -Command "Add-Type -AssemblyName
          System.Windows.Forms; ', ...
923       '$s = [System.Windows.Forms.Screen]::AllScreens | ForEach-Object {
          ', ...
924       '  [PSCustomObject]@{ deviceName=$_.DeviceName;
          widthLogic=$_.Bounds.Width; heightLogic=$_.Bounds.Height;
          x=$_.Bounds.X; y=$_.Bounds.Y; primary=$_.Primary } ', ...
925       '}; $s | ConvertTo-Json -Compress"'
926     ];
927     [st, out] = system(ps_cmd);
928     if st ~= 0
929       Bota = sprintf('Podprogram Zjisti_monitory_PS vrací:\n Chyba
          PowerShellu: %s', out);
930     end
931     if length(Bota)>0, return; end     #je obuf
932
933
934
935     % Parse JSON (handle single object vs array)
936     try
937       j = jsondecode(out);
938     catch
939       % Pokud jsondecode selže, vypiš surový výstup pro diagnostiku
940       Bota = sprintf('Podprogram Zjisti_monitory_PS vrací:\n Nepodařilo
```

```
        se dekódovat JSON z PowerShellu: %s', out);
941     end
942     if length(Bota)>0, return; end      #je obuf
943
944
945
946
947     % Normalize to array
948     if isstruct(j) && ~isscalar(j)
949       arr = j;
950     elseif isstruct(j) && isscalar(j)
951       arr = j; % ConvertTo-Json returns single object for 1 screen; keep
        as scalar
952     else
953       Bota = sprintf('Podprogram Zjisti_monitory_PS vrací:\n Neočekávaný
        formát JSON: %s', out);
954     end
955     if length(Bota)>0, return; end      #je obuf
956
957
958     % Build monitors structure array
959     if isscalar(arr)
960       n = 1;
961     else
962       n = numel(arr);
963     end
964     monitors(1:n) = struct('deviceName', '', 'widthLogic', 0,
        'heightLogic', ...
965                            0, 'x', 0, 'y', 0, 'primary', false);
966     for k = 1:n
967       if isscalar(arr)
968         item = arr;
969       else
970         item = arr(k);
971       end
972       monitors(k).deviceName = item.deviceName;
973       monitors(k).widthLogic = double(item.widthLogic);
974       monitors(k).heightLogic = double(item.heightLogic);
975       monitors(k).x = double(item.x);
976       monitors(k).y = double(item.y);
977       monitors(k).primary = logical(item.primary);
978     end
979
980   endfunction#.....................................................
      ........
981
982
983
984
985
986
987   #.....................................................................
      ........
988   function [Bota, dpi] = Zjisti_DPI_ef()
989
990     # Funkce pro přečtení efektivního DPI odpovídajícího Windows škálování
```

```
991     # pro všechny displeje (monitory) počítače vytvořená IÁÁ, éé pardon AI
992     # GPT-5 mimi a voláním příkazů Poweršellu 11.1.2026.
993
994     # Stručný popis podprogramu od IAA:
995     #   Podprogram před čtením DPI nastaví proces na per-monitor DPI aware
996     #   (pokud API dostupné) a pak volá GetDpiForMonitor pro každý monitor
997     #    — vrací efektivní DPI odpovídající Windows škálování,
998     #   např. 144 pro 150%, 120 pro 125%.
999     # Podprogram volá SetProcessDpiAwareness(2) (per-monitor) pokud je
        funkce
1000    # dostupná. Pokud není (velmi staré Windows), volání je ignorováno.
1001
1002    # Pro každý monitor vystoupí dvě hodnoty:
1003    #     dpi(i,1)   je DPI ve směru X (horizontální)
1004    #     dpi(i,2)   je DPI ve směru Y (vertikální)
1005    # Hodnoty budou většinou stejné, rozdíl může být např. když by
        monitor měl
1006    # různý počet bodů na palec horizontálně a vertikálně.
1007
1008    # Podprogram tedy např. pro nastavené wokenní zvětšení (škálování) 150%
1009    # vrátí hodnotu 144. To vychází z toho, že we vokenicích se jako
        výchozí
1010    # DPI používá hodnota 96 DPI, tedy 96 DPI = 100%. Vrácené hodnoty se
        tedy
1011    # na jedlá procenta musí přepočítat, např. pro vrácených 144 v
        případě 150%:
1012    # (144 / 96) * 100 = 150, pro vrácených 120: (120 / 96) * 100 = 125
        atp.
1013
1014
1015    Bota = '';
1016
1017    if ispc == 0
1018      Bota = sprintf('Podprogram Zjisti_DPI_ef vrací:\n Tento podprogram
          funguje pouze na Windows.');
1019    end
1020    if length(Bota)>0, return; end     #je obuf
1021
1022
1023
1024    ps_script = [
1025      'Add-Type -TypeDefinition @"' char(10) ...
1026      'using System; using System.Runtime.InteropServices; using
          System.Collections.Generic;' char(10) ...
1027      'public static class DPIHelper {' char(10) ...
1028      '  private const int MDT_EFFECTIVE_DPI = 0;' char(10) ...
1029      '  [DllImport("shcore.dll")] public static extern int
          GetDpiForMonitor(IntPtr hmonitor, int dpiType, out uint dpiX, out
          uint dpiY);' char(10) ...
1030      '  [DllImport("shcore.dll")] public static extern int
          SetProcessDpiAwareness(int value);' char(10) ...  %
          0=Unaware,1=System,2=PerMonitor
1031      '  [DllImport("user32.dll")] public static extern bool
          EnumDisplayMonitors(IntPtr hdc, IntPtr lprcClip,
          MonitorEnumDelegate lpfnEnum, IntPtr dwData);' char(10) ...
1032      '  public delegate bool MonitorEnumDelegate(IntPtr hMonitor,
```

```
      IntPtr hdcMonitor, ref RECT lprcMonitor, IntPtr dwData);' char(10) ...
      '  [StructLayout(LayoutKind.Sequential)] public struct RECT { public int left, top, right, bottom; }' char(10) ...
      '  public static List<string> GetAllDpi() {' char(10) ...
      '    try { SetProcessDpiAwareness(2); } catch { /* ignore if not available */ }' char(10) ...
      '    var list = new List<string>();' char(10) ...
      '    MonitorEnumDelegate d = (IntPtr hMon, IntPtr hdc, ref RECT r, IntPtr dd) => {' char(10) ...
      '      uint x=0,y=0; int res = GetDpiForMonitor(hMon, MDT_EFFECTIVE_DPI, out x, out y);' char(10) ...
      '      if (res == 0) list.Add(string.Format("({0},{1})", x, y));' char(10) ...
      '      return true; };' char(10) ...
      '    EnumDisplayMonitors(IntPtr.Zero, IntPtr.Zero, d, IntPtr.Zero);' char(10) ...
      '    return list; }' char(10) ...
      '}' char(10) ...
      '"@' char(10) ...
      '$list = [DPIHelper]::GetAllDpi();' char(10) ...
      'if ($list -eq $null -or $list.Count -eq 0) { Write-Output "" ; exit }' char(10) ...
      'Write-Output ($list -join ";" )' ];

    tf = [tempname '.ps1'];
    fid = fopen(tf,'w');
    if fid == -1
      Bota = sprintf('Podprogram Zjisti_DPI_ef vrací:\n Nelze vytvořit dočasný PS program.');
    endif
    if length(Bota)>0, return; end      #je obuf

    fwrite(fid, ps_script);
    fclose(fid);

    cmd = ['powershell -NoProfile -ExecutionPolicy Bypass -File "' tf '"'];
    [status, out] = system(cmd);

    delete(tf);

    if status ~= 0
      Bota = sprintf('Podprogram Zjisti_DPI_ef vrací:\n PowerShell volání selhalo: %s', out);
    end
    if length(Bota)>0, return; end      #je obuf


    out = strtrim(out);
    if isempty(out)
      dpi = zeros(0,2);
      return;
    end

    parts = strsplit(out, ';');
    n = numel(parts);
```

```
1078      dpi = zeros(n,2);
1079      for i = 1:n
1080        s = regexprep(parts{i}, '[\(\)\s]', ''); % odstranit závorky/mezery
1081        v = strsplit(s, ',');
1082        dpi(i,1) = str2double(v{1});
1083        dpi(i,2) = str2double(v{2});
1084      end
1085
1086    endfunction#.....................................................
          ........
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096    #.....................................................................
          ........
1097    function VobsadDilkyWos (handl, Miminum, Maximum, Wosa)
1098
1099      # obsluha kreslení wos
1100      # ve Wosa je, pro kterou wosu se budou dílky a rozsah předělávat:
1101      #     když první znak bude "x" nebo "X" (s vyjímkou předchozích
            mizer, ty se
1102      #        ignorují) bude to wosa X (vodorovná),
1103      #        musí tedy Miminum, Maximum být hodnoty vykreslované na wosu X
1104      #     když první znak bude jiný bude to wosa Y (svislá)
1105      #        musí tedy Miminum, Maximum být hodnoty vykreslované na wosu Y
1106      #     (třetí wosu Z to zatím neobhospodařuje)
1107      #
1108      # handl je handle na objekt wos (gca) pro který se to dělá
1109
1110
1111
1112
1113      Wosa = upper(strtrim(Wosa));  # může se upravovat vstupní prom.
            Wosa, protože
1114                                    # tady neleze veeen
1115
1116      KratkaWosa = false;
1117      if Cmuchret(Wosa(1:1),"Y")==1
1118        KratkaWosa = true;
1119      endif
1120
1121      [Hod] = IntervOs (Miminum , Maximum, KratkaWosa);
1122
1123
1124      # Pořadi následujících příkazů MUSÍ být takovéto:
1125      #      1. nastavení rozsahu,
1126      #      2. mood maňuál
1127      #      3. nastavení velkých dílků
1128      #      4. nastavení malých dílků
1129      # jinak si může některé hodnoty přeonačit (třeba dyš se nastaví malé
```

```
      dílky
1130  # dříf jak velké...
1131
1132
1133  if Cmuchret(Wosa,"X")==1;          #takto nemusí být ve Wosa jen
      jeden znak
1134
1135    # wosa X (vodorovná)
1136    xlim(handl, [Hod(6), Hod(7)]);  #nastavení vypočteného rozsahu
        wosy handl
1137    set(handl,'xtickmode','manual');
1138    set(handl,"xtick",(Hod(1):Hod(4):Hod(2)));
1139    set(handl,"xminortickvalues",(Hod(6):Hod(5):Hod(7)));
1140
1141  else
1142    # wosa Y (svislá)
1143
1144    # Případné trochu zvětšení rozsahu wosy dyš se kryje s max nebo
        min dat
1145    # zatím jen u Y wosy, aby byl líip vyděti nasnímaný průběch.
1146    # A u takto nastavovaných wos MUSÍ být tady a takto, aby se použilo
1147    # a už nezměnilo, poáč dyš jsem je měl za čtveřicí příkazů pro
        nastavení
1148    # tak tam provedený ylim() způsobuje nové přepočítání dílků podle
        Octave
1149    # a to tady nechcu...
1150    MaxGr = Hod(7);
1151    MinGr = Hod(6);
1152    if MaxGr <= Maximum*1.025
1153      MaxGr = MaxGr*1.025;
1154      if MaxGr > Hod(2) + Hod(4)
1155        #ještě enev. úprava maxima podle vel. dílků wos
1156        Hod(2) = Hod(2) + Hod(4);
1157      endif
1158    endif
1159    if MinGr >= Miminum*1.025
1160      MinGr = MinGr*1.025;
1161      if MinGr < Hod(1) - Hod(4)
1162        #ještě enev. úprava mimina podle vel. dílků wos
1163        Hod(1) = Hod(1) - Hod(4);
1164      endif
1165    endif
1166
1167
1168    ylim(handl, [MinGr, MaxGr]);  #nastavení vypočteného rozsahu wosy
        handl
1169    set(handl,'ytickmode','manual');
1170    set(handl,"ytick",(Hod(1):Hod(4):Hod(2)));
1171    set(handl,"yminortickvalues",(Hod(6):Hod(5):Hod(7)));
1172
1173  endif
1174  drawnow;
1175
1176
1177
1178 endfunction#.....................................................
```

```
    ........
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188  #.....................................................................
    ........
1189  function [Hod] = IntervOs (Min, Max, KratkaWosa)
1190    # Převzatý a upravený podprogram z programu Zpracování měření z VB6.
1191    #
1192    # Vypočte z minima a maxima hodnoty pro vosy, Max a Min by měly být
      skutečné
1193    # maximální a miminální hodnoty z kreslených dat dotyčné wosy.
1194    #
1195    # Když bude KratkaWosa = true, tak se malé dílky udělají s větším
      odstupem,
1196    # (4 nebo 5 na velký dílek místo 10 na velký dílek) aby na kratší wose
1197    # nebylo dílkování moc husté...
1198    #
1199    # Ve výstupu jsou výsledky uloženy takto:
1200    #      Hod(1)   miminum  ("kulaté" podle velkých dílků)
1201    #      Hod(2)   maximum  ("kulaté" podle velkých dílků)
1202    #      Hod(3)   průsečík (podle velkých dílků)
1203    #      Hod(4)   velký dílek
1204    #      Hod(5)   malý dílek
1205    #      Hod(6)   miminum  ("kulaté" podle malých dílků)
1206    #      Hod(7)   maximum  ("kulaté" podle malých dílků)
1207    #      Hod(8)   průsečík (podle malých dílků)
1208    #
1209    #  Tady spočítá všechny možnosti najednou a vybrat z nich se musí jinde
1210
1211
1212
1213
1214    if Max == Min
1215      Max = Max + 0.9;    # aby osy byly rozdílné o 1
1216    endif
1217
1218
1219    P = 10 ^ (floor(log10(abs(Max - Min))));
1220
1221    Hod(1) = floor(Min / P) * P;
1222    if Min > 0
1223      if Hod(1) > Min
1224        Hod(1) = (floor(Min / P) - 1) * P;
1225      endif
1226    endif
1227
1228    Hod(2) = (floor(Max / P)) * P;
1229    if Max > 0
1230      if Hod(2) < Max
```

```
1231          # Toto je tady proto, že fuňkce floor() (ve VB Int()) funguje tak,
1232          # že např. z -2.7 udělá -3, ale z +2.7 udělá 2, takže by se to tam
1233          # pokud bylo Max trochu větší než Hod(2) nevlezlo, a proto tam
              původně
1234          # bylo +1 furt, ale v tom případě, pokud bylo Max = Hod(2) tak to
1235          # zbytečně přidávalo velký dílek (u mimina výše je to kladných
              hodnot
1236          # obdobně, jen by se mělo ubírati).
1237          Hod(2) = (floor(Max / P) + 1) * P;
1238        endif
1239      endif
1240
1241
1242
1243
1244      P = log10(abs(Hod(2) - Hod(1)));
1245
1246      if (P - floor(P)) < 0.30102999567
1247         #10^(0.30102999567)=2.000000000027717
1248         Hod(4) = 0.2;
1249         if KratkaWosa
1250           Hod(5) = 0.05;
1251         else
1252           #toto tady dát jen pro delší wosu páč na krátké wose je to moc
              jemné...
1253           Hod(5) = 0.02;
1254         endif
1255       elseif (P - floor(P)) > 0.69897000434
1256         #10^(0.69897000434)=5.000000000045835
1257         Hod(4) = 1;
1258         if KratkaWosa
1259           Hod(5) = 0.2;
1260         else
1261           #toto tady dát jen pro delší wosu páč na krátké wose je to moc
              jemné...
1262           Hod(5) = 0.1;
1263         endif
1264       else
1265         Hod(4) = 0.5;
1266         Hod(5) = 0.1;
1267      endif
1268
1269      P = floor(P);
1270      Hod(4) = Hod(4) * 10 ^ P;
1271      Hod(5) = Hod(5) * 10 ^ P;
1272
1273
1274
1275
1276
1277      # & je and    | je or    ~ je not   ~= je není rovno   cocoti
1278      if (Min <= 0) && (Max >= 0) #chce to zkratku &&
1279        # úprava mimina a maxima tak, aby dyš je wosa přes nulu tam nula
            byla dycky,
1280        # v tomto případě se to musí podle malých dílků upravit
1281        # tam, kde se v Oktááve nastavuje ylim, páč je potřeba nastavit
```

```
      limity
1282  # gřafu, a to bez změny zde spočítaného jde jen tam a v tam
      uvedeném pořadí
1283  Hod(1) = 0;
1284  while Min <= Hod(1) - Hod(4)
1285    Hod(1) = Hod(1) - Hod(4);
1286  endwhile
1287
1288  Hod(2) = 0;
1289  while Max >= Hod(2) + Hod(4)
1290    Hod(2) = Hod(2) + Hod(4);
1291  endwhile
1292
1293  # miminum, maximum a prusečík pro malé dílky
1294  Hod(6) = Hod(1);
1295  while Min < Hod(6)          # uprava minima na malý dílek
1296    Hod(6) = Hod(6) - Hod(5);
1297  endwhile
1298  Hod(7) = Hod(2);
1299  while Max > Hod(7)          # uprava maxima na malý dílek
1300    Hod(7) = Hod(7) + Hod(5);
1301  endwhile
1302
1303 else
1304
1305  # miminum, maximum a prusečík pro velké dílky
1306  while Min > Hod(1) + Hod(4)  # uprava minima na interval popisu
1307    Hod(1) = Hod(1) + Hod(4);
1308  endwhile
1309
1310  while Max < Hod(2) - Hod(4)  # uprava maxima na interval popisu
1311    Hod(2) = Hod(2) - Hod(4);
1312  endwhile
1313
1314  # miminum, maximum a prusečík pro malé dílky
1315  Hod(6) = Hod(1);
1316  while Min > Hod(6) + Hod(5)  # uprava minima na malý dílek
1317    Hod(6) = Hod(6) + Hod(5);
1318  endwhile
1319  Hod(7) = Hod(2);
1320  while Max < Hod(7) - Hod(5)  # uprava maxima na malý dílek
1321    Hod(7) = Hod(7) - Hod(5);
1322  endwhile
1323
1324 endif
1325 Hod(3) = Hod(1);
1326 Hod(8) = Hod(6);
1327
1328
1329
1330 endfunction#.....................................................
      ........
1331
1332
1333
1334
```

```
1335
1336
1337  #.....................................................................
      ........
1338  function indeks = Cmuchret(vcem, co)
1339    # fuňkce by měla vrátiti první výskyt retězce co v retezci vcem,
        nebo nulu
1340    # gdyš tam nejni
1341
1342    indeksy = strfind(vcem, co);
1343    if isempty(indeksy)
1344      indeks = 0;
1345    else
1346      indeks = indeksy(1);
1347    end
1348  endfunction;#.....................................................
      ........
1349
1350
1351
1352
1353
1354
1355  #.....................................................................
      ........
1356  function aktivuj_omez_rozm(hFig)
1357    # Připojí ResizeFcn k figuře, aby volala funkci omezeni_rozmeru
1358    # při každé změně velikosti.
1359
1360    # Pokud není hFig zadána, použije se gcf.
1361    if nargin < 1 || isempty(hFig)
1362      hFig = gcf;
1363    end
1364
1365    set(hFig, 'sizechangedfcn', @(h,ev) omezeni_rozmeru(hFig));
1366
1367  endfunction;#.....................................................
      ........
1368
1369
1370
1371
1372  #.....................................................................
      ........
1373  function nastavPozLegendyRelatkOse(hOsy, hleg, rel_pos);
1374    # Funkce pro nastavení pozice legendy relativně k ose
1375    # Vstupní párametry MUSÍ být jedlé, nekontrolují se tady
1376    # Podprogram je od GPT-5 mimi
1377
1378    # kde je:
1379    # hOsy:     handle oné osy
1380    # hleg:     handle oné legendy
1381    # rel_pos:  [x y sirka výška] v normalizovaných osových jednotkách
1382
1383
1384    # Získat pozice v normalízovaných jednotkách proti figúře pro osu a
```

```
      pro legendu
1385    puv_jedn_osy = get(hOsy, "Units");
1386    puv_jedn_legendy = get(hleg, "Units");
1387    set(hOsy, "Units", "normalized");
1388    set(hleg, "Units", "normalized");
1389    PosOsy = get(hOsy, "Position");  # [x y sirka vyška] normalízováno
        vzhledem k figúře
1390
1391    # Spočítat legend pozici v normalizovaných (vzhledem k figúře)
        souřadnicích
1392    pos_leg_prepoc = [ PosOsy(1) + rel_pos(1)*PosOsy(3), ...
1393                       PosOsy(2) + rel_pos(2)*PosOsy(4), ...
1394                       rel_pos(3)*PosOsy(3), ...
1395                       rel_pos(4)*PosOsy(4) ];
1396    set(hleg, "Position", pos_leg_prepoc);
1397
1398    # vrátit původní jednotky
1399    set(hOsy, "Units", puv_jedn_osy);
1400    set(hleg, "Units", puv_jedn_legendy);
1401
1402  endfunction;#.....................................................
        ........
1403
1404
1405
1406
1407  #.....................................................................
        ........
1408  function umisti_legendy(fig);
1409    # znovu umístí obě legendy tady použité podle proměnnných uložených
        ve figúře s handlem fig
1410
1411
1412    # legenda hl. gřafu
1413    hleg = getappdata(fig, "hLegHL");
1414    ax = getappdata(fig, "hOsyXHL");
1415    rel_pos = getappdata(fig, "RelPosLegHL");
1416    if isempty(hleg) || ~ishandle(hleg) || isempty(ax) || ~ishandle(ax)
1417      return;
1418    end
1419
1420    % a umístiti
1421    nastavPozLegendyRelatkOse(ax, hleg, rel_pos);
1422
1423
1424
1425    # legenda BT gřafu
1426    JeKresBT = getappdata(fig, "KresSmikHarm");
1427
1428    if JeKresBT
1429      hleg = getappdata(fig, "hLegBT");
1430      ax = getappdata(fig, "hOsyXBT");
1431      rel_pos = getappdata(fig, "RelPosLegBT");
1432      if isempty(hleg) || ~ishandle(hleg) || isempty(ax) || ~ishandle(ax)
1433        return;
1434      end
```

```

    % a umístiti
    nastavPozLegendyRelatkOse(ax, hleg, rel_pos);
  endif

  drawnow;

endfunction;#.......................................................................












#..............................................................................
function [y] = rms(x);

  #výpočet efektivní hovnoty, RMS - root mean square

  n = length(x);
  y = sqrt(sum(x.^2)/n);

endfunction;#.......................................................................






#..............................................................................
function UlozInifile (Jminifile);

  #xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  #uložení do ini filé s stejným jménem jako toto spuštěné .m filé
  #a s rozšířením .NUF (snad ještě není)


  #proměnné předávané zobálně
  global Iniplk;  #aby fuňkce znala zobální proměnnné, musí se v ní
                  #znovu takto deklarovat (samozřejmě pod stejným jménem)

  global ZpracDat;
  global Cestafile;
  global AmplHarmV;
  global Fsmik RadBTfiltru JeBTfiltr;
  global PocKresHarm;
  global VerzeProg;
  global Monitory;
```

```
1485    global hfig1;
1486
1487
1488
1489    # Otevření filé pro zápis
1490    fid = fopen(Jminifile, 'w');  # 'w' je pro zápis, minulý obsach bude přepsán
1491
1492    # Kontrola, zda se filé otevřelo správně
1493    if fid == -1
1494       #  filé se neotevřelo
1495
1496       plkbox(cstrcat('Nejde otevřít soubor s nastaveními: ',Jminifile,newline, ...
1497              'nastavení nebude uloženo !!!'), 'Uložení do filé s nastavením');
1498
1499    else
1500
1501      plk = cstrcat('Soubor s nastaveními pro program Výpočet harmonického zkreslení,');
1502      plk = cstrcat(plk, ' verze z ',VerzeProg, newline);
1503      fprintf(fid, plk);   # Zápis textového řetězce do filé
1504      plk = cstrcat('vyrobený v prostředí GNU Octave, autor Luděk Ruffer  lruffer@volny.cz' ...
1505                    ,newline,newline );
1506      fprintf(fid, plk);   # Zápis textového řetězce do filé
1507
1508
1509
1510      plk = cstrcat(newline,'Hodnoty způsobu zpracování dat znamenají:',newline );
1511      fprintf(fid, plk);
1512      plk = cstrcat('1 = Data zpracovat jako celé kmity od prvního průchodu nulou',newline );
1513      fprintf(fid, plk);
1514      plk = cstrcat('2 = Data zpracovat jako celé kmity od prvního maxima',newline );
1515      fprintf(fid, plk);
1516      plk = cstrcat('3 = Data zpracovat jako celé kmity od začátku filé',newline );
1517      fprintf(fid, plk);
1518      plk = cstrcat('4 = Data zpracovat z celého filé',newline );
1519      fprintf(fid, plk);
1520      plk = cstrcat('Při načtení jiné nebo žádné hodnoty je nastaveno = 1',newline,newline );
1521      fprintf(fid, plk);
1522
1523
1524      plk = cstrcat(Iniplk.zpracdat, num2str(ZpracDat, "%d"),newline,newline);
1525      fprintf(fid, plk);
1526
1527
1528      plk = cstrcat(newline, Iniplk.PocKrHarm, num2str(PocKresHarm, "%d"),newline );
```

```
1529        fprintf(fid, plk);
1530
1531
1532        plk = cstrcat(newline, Iniplk.amplharmveV, num2str(AmplHarmV,
            "%d"),newline );
1533        fprintf(fid, plk);
1534
1535
1536
1537        plk = cstrcat(newline, Iniplk.cestafile);
1538        fprintf(fid, plk);
1539        #cesta se zpět. lomítky MUSÍ být uložena jako řetěz pomocí
            následujícího
1540        #formátování, jinak z ní oktáva ty zpětná lomítka vyháže!!!!!!
1541        fprintf(fid, '%s\n', Cestafile);
1542
1543
1544        plk = cstrcat(newline, Iniplk.PoziceFig);
1545        fprintf(fid, plk);
1546        fprintf(fid, '%d  %d  %d  %d\n', round(get(hfig1, 'position')));
1547
1548
1549        if JeBTfiltr
1550          plk1 = 'ano';
1551        else
1552          plk1 = 'ne';
1553        endif
1554        plk = cstrcat(newline, Iniplk.pouzBT, plk1, newline);
1555        fprintf(fid, plk);
1556
1557        plk = cstrcat(Iniplk.fsmikBT, num2str(Fsmik, "%d"),newline );
1558        fprintf(fid, plk);
1559
1560        plk = cstrcat(Iniplk.radBT, num2str(RadBTfiltru, "%d"),newline );
1561        fprintf(fid, plk);
1562
1563
1564
1565
1566        # následují informační hodnoty, při spuštění programu se nečtou:
1567
1568        plk = cstrcat(newline,newline, 'Další hodnoty jsou jen informační,');
1569        plk = cstrcat(plk, ' při spuštění programu se
            nečtou.',newline,newline);
1570        fprintf(fid, plk);
1571
1572        plk = cstrcat('Počítač, na kterém program běží má tyto
            monitory:',newline );
1573        fprintf(fid, plk);
1574        for i = 1:length(Monitory)
1575          fprintf(fid,'  Monitor %d%s',i,':   ');
1576          if Monitory(i).primarni
1577            fprintf(fid,cstrcat('  je primární;  ',char(9)));
1578          else
1579            fprintf(fid,cstrcat('  není primární;',char(9)));
1580          endif
```

```
1581          fprintf(fid,'  šířka %d %s',Monitory(i).sirkaSkut,';');
1582          fprintf(fid,'  výška %d %s',Monitory(i).vyskaSkut,';');
1583          fprintf(fid,'  zvětšení %d%% %s',Monitory(i).zvetseni_proc,';');
1584          fprintf(fid,'  logická šířka %d %s',Monitory(i).sirkaLogic,';');
1585          fprintf(fid,'  logická výška %d %s',Monitory(i).vyskaLogic,';');
1586
1587          if Monitory(i).rucni_vstup
1588          fprintf(fid,cstrcat('  hodnoty vloženy ručně'));
1589          else
1590          fprintf(fid,cstrcat('  hodnoty přečteny programem'));
1591          endif
1592
1593          fprintf(fid,newline);
1594        endfor
1595
1596
1597        plk = cstrcat(newline, 'Soubor byl uložen ');
1598        fprintf(fid, plk);
1599        # nápo věda k formátování času je u time
1600        fprintf(fid, strftime ("%d.%m.%Y v %T hodin.", localtime (time
          ())), newline);
1601
1602
1603        plk = cstrcat(newline, newline,'Konec souboru s
          nastaveními.',newline );
1604        fprintf(fid, plk);
1605
1606        # Zavření filé
1607        fclose(fid);
1608
1609      endif
1610    #xonec ukládání
      xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
1611    endfunction#.....................................................
      ........
1612
1613
1614
1615
1616
1617
1618    #.....................................................................
      ........
1619    function Jemriz = FunNufmriz (Jemriz, hgrf1, hgrf2);
1620      #V Oktáávě jsou v zátvorce za jménem fuňkce JEN vstupní párametry do
        fuňkce,
1621      #výstupní musí být před rovnítkem a dyš je jich víc tak v hranatejch
1622      #zátvorkách odděleny čárkama.
1623
1624
1625      if (Jemriz == true);
1626        grid (hgrf1, 'off');
1627        grid (hgrf2, 'off');
1628        Jemriz = false;
1629      else
1630        #u obouch gřafúú je to děláno takto, tak tady to tak musí býti taky
```

```
1631        #aby to obnovovalo mřížky u malejch dílků
1632        set (hgrf1, "xgrid", "on","ygrid", "on");
1633        set (hgrf1, "xminorgrid", "on","yminorgrid", "on");
1634
1635        set (hgrf2, "xgrid", "on","ygrid", "on");
1636        set (hgrf2, "xminorgrid", "on","yminorgrid", "on");
1637
1638        Jemriz = true;
1639      endif
1640
1641    endfunction#.....................................................
    ........
1642
1643
1644
1645
1646
1647
1648
1649    #.....................................................................
    ........
1650    function plkbox(message, title)
1651        #užívatelský msg box, ale velikost by byla potřeba upravit podle
        vloženého plku...
1652
1653
1654        figuury = findobj('Type', 'figure');      #získání všech
        otevřených figur
1655        pocet_figur = length(figuury);            #počet otevřených figur
1656
1657        if pocet_figur > 0
1658           cisla_figur = get(figuury, 'Number');  #čísla otevřených figuur
1659           maximumo = max(cisla_figur);
1660        else
1661           maximumo = 98;    # nějaké číslo které snad hned nebude
1662        endif
1663        Cfig = maximumo + 1;
1664
1665        sirplku = 600;
1666
1667        h = figure (Cfig, 'Name', title, 'NumberTitle', 'off', 'MenuBar',
        'none', ...
1668                    'position', [500, 500, sirplku, 150]);
1669
1670
1671        uicontrol('parent',h , 'Style', 'text', 'String', message,
        'FontSize', 12, ...
1672                  'FontWeight', 'normal', 'position', [20, 70, sirplku -
                  25, 50], ...
1673                  'HorizontalAlignment', 'center', 'backgroundcolor', [1 1
                  1]);
1674
1675        sirtlac = 100;
1676        uicontrol('parent',h ,'Style', 'pushbutton', 'String', 'OK',
        'position', ...
1677                  [sirplku/2 - sirtlac/2, 20, sirtlac, 30], ...
```

```
1678                'Callback', @volani_tlac_plkboxu);
1679
1680
1681        #Čekání na stisk tlačítka
1682        uiwait(h);  # Pozastaví provádění, dokud není figure zavřena nebo tlačítko stisknuto
1683
1684        try
1685          # dyš se figure s plkem zavře křížkem v rohu je tady už zavřená
1686          # a close(h) nemá co zavírat tak udělá botu
1687          close(h);
1688        end
1689
1690
1691    end#.....................................................................
        ........
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701    #.....................................................................
        ........
1702    function [hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile, hgrf1,
        hgrf2, men, Velpis);
1703
1704    [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2, men,
        Velpis, '-r300', ', menší rozlišení');
1705
1706    endfunction#.....................................................................
        ........
1707
1708
1709
1710    #.....................................................................
        ........
1711    function [hgrf1, hgrf2] = UlozdofileVetsi (Cestafile, Jmfile, hgrf1,
        hgrf2, men, Velpis);
1712
1713    [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2, men,
        Velpis, '-r600', ', větší rozlišení');
1714
1715    endfunction#.....................................................................
        ........
1716
1717
1718
1719
1720
1721    #.....................................................................
        ........
1722    function [hgrf1, hgrf2] = Ulozdofile (Cestafile, Jmfile, hgrf1, hgrf2,
```

```
men, Velpis, Rozlis, plkroz);

  #nakreslí se znovu s písmem jedlým pro tisk, vytiskne se a překreslí se
  #s písmem pro vobra žofku


  global Velpistisk;
  global TlacimFiguru;


  plk = cstrcat("Zadej jméno pro uložení do souboru (obrazové formáty)", ...
                plkroz);
  [fjmeno, fcesta, fpocet] = uiputfile ({"*.jpg","Obraz jpg"; "*.gif", "Obraz gif"; ...
                         "*.tif","Obraz tif"; "*.png", "Obraz png" }, ...
                         plk, Cestafile);


  if fpocet < 1
    odp = msgbox('Nebylo vybráno žádné filé pro uložení, nic nebude uloženo.' , ...
                'Uložení do filé');
    return;
  endif


  mysisko = get (gcf, 'pointer');
  set (gcf, 'pointer', 'watch');
  #čekatiti''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
  set (men.zaklovl, "enable", 'off');
  set (men.zakledit, "enable", 'off');

  Tlacenice = cstrcat(fcesta , fjmeno);

  plkfig = get(gcf, 'name');
  plk = cstrcat(plkfig, ' - Moment, tlačím do filé: ',Tlacenice);
  set (gcf, 'name', plk);




  TlacimFiguru = true;

  ZaklPosFig = getappdata(gcf,'ZaklPosFig');
  VcilJednFig = get (gcf, 'units');
  set (gcf, 'units', 'normalized');
  VcilPosFig = get (gcf, 'position');

  # úprava pozic legend pro výstup do obrazového filé
  ZaklPosLegHL = getappdata(gcf,'RelPosLegHL');
  ZaklPosLegBT = getappdata(gcf,'RelPosLegBT');
  ZaklPosLegHLup = ZaklPosLegHL;
  ZaklPosLegBTup = ZaklPosLegBT;
  ZaklPosLegHLup(1) = ZaklPosLegHL(1) - 0.008;   # úprava počátku výšky umístění
```

```
1773      ZaklPosLegHLup(2) = ZaklPosLegHL(2) - 0.035;   # úprava počátku
          výšky umístění
1774      ZaklPosLegBTup(1) = ZaklPosLegBT(1) - 0.045    # úprava počátku
          šířky umístění
1775      ZaklPosLegBTup(2) = ZaklPosLegBT(2) - 0.075;   # úprava počátku
          výšky umístění
1776      setappdata(gcf,'RelPosLegHL', ZaklPosLegHLup);
1777      setappdata(gcf,'RelPosLegBT', ZaklPosLegBTup);
1778
1779
1780      MeniloTuRozmery = false;
1781      if all(ZaklPosFig == VcilPosFig) == false
1782        # při takto udělané podmínce se sem dostane když nebudou všechny
          prvky
1783        # dotyčných polí stejné (stačí jeden různý)
1784
1785        MeniloTuRozmery = true;
1786        set (gcf, 'position', ZaklPosFig);
1787      endif
1788
1789
1790
1791      #některé výstupní proměnnné tady nejsou potřeba
1792      [ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,
          men, ...
1793                                          Velpistisk, [], false, true, true);
1794
1795      print (gcf,Tlacenice , Rozlis);
1796
1797      # vrácení původních pozic tady, aby v následujícím VypDFT překreslilo
1798      setappdata(gcf,'RelPosLegHL', ZaklPosLegHL);
1799      setappdata(gcf,'RelPosLegBT', ZaklPosLegBT);
1800
1801      [ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2,
          men, ...
1802                                          Velpis , [], false, true, false);
1803
1804
1805      if MeniloTuRozmery
1806        # eště dyš jsou rozměry normálized a je TlacimFiguru tak vrátiti
          rozměry
1807        # jaký byly při vstupu sem
1808        set (gcf, 'position', VcilPosFig);
1809      endif
1810
1811
1812
1813
1814      TlacimFiguru = false;
1815
1816
1817
1818      set (gcf, 'units', VcilJednFig);
1819      set (gcf, 'name', plkfig);
1820      set (men.zaklovl, "enable", 'on');
1821      set (men.zakledit, "enable", 'on');
```

```
1822      set (gcf, 'pointer', mysisko);
          #nečekatiti''''''''''''''''''''''''''''''''''''''''
1823
1824
1825    endfunction#.....................................................
        ........
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836    #.....................................................................
        ........
1837    function PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis,
        NoveZprac);
1838
1839    #přepočítá a překreslí výstup (figúúru) pro nově zadaný způsob zpracování
1840    # (od průchodu nulou, od maxima, celé filé a tak)
1841
1842
1843    global ZpracDat
1844
1845
1846
1847    #dyš je stejné jak minulé tak vééén
1848    if ZpracDat == NoveZprac
1849      return;
1850    end
1851
1852
1853    mysisko = get (gcf, 'pointer');
1854    set (gcf, 'pointer', 'watch');
        #čekatiti''''''''''''''''''''''''''''''''''''''''''''
1855    set (men.zaklovl, "enable", 'off');
1856    set (men.zakledit, "enable", 'off');
1857
1858
1859
1860    plkfig = get(gcf, 'name');
1861    plk = cstrcat(plkfig, ' - Moment, znovu zpracovávám data');
1862    set (gcf, 'name', plk);
1863
1864
1865    ZpracDat = NoveZprac;
1866
1867
1868    # zafajfkování aktuálního menu pro zpracování dat z filé
1869    for i = 1: length(men.nastpod)
1870      set(men.nastpod(i), "checked", "off")
1871    endfor
1872    set(men.nastpod(ZpracDat), "checked", "on")
```

```
#přeježení pro nové ZpracDat - některé výstupní proměnnné tady nejsou potřeba
[ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2, men, ...
                                    Velpis , [], false, false, false);

set (gcf, 'name', plkfig);
set (men.zaklovl, "enable", 'on');
set (men.zakledit, "enable", 'on');
set (gcf, 'pointer', mysisko); #nečekatiti''''''''''''''''''''''''''''''''''''''''

endfunction#...........................................................................

#...............................................................................
function ZmenKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);

global AmplHarmV;

mysisko = get (gcf, 'pointer');
set (gcf, 'pointer', 'watch'); #čekatiti''''''''''''''''''''''''''''''''''''''''

plk = cstrcat("Přepínač [dB - V] kreslení Y osy grafu harmonických: teď je ");
if AmplHarmV == 0
  AmplHarmV = 1;
  plk = cstrcat(plk, 've [V]');
else
  AmplHarmV = 0;
  plk = cstrcat(plk, 'v [dB]');
end

set (men.nastpodharm, 'text' , plk);
set (men.kontmenu.nastharm, 'text' , plk);
```

```
1922
1923  #přeježení pro nové kreslení harmonických - některé výst. prom. nejsou potřeba
1924  [ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2, men, ...
1925                                      Velpis , [], false, true, false);
1926
1927  set (gcf, 'pointer', mysisko);  #ne čekatiti''''''''''''''''''''''''''''''''''
1928
1929
1930  endfunction#.....................................................................
1931
1932
1933
1934
1935
1936
1937
1938
1939  #.....................................................................
1940  function ZmenPocKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);
1941
1942
1943    global PocKresHarm PocHarm FposlHarm Fvypis;
1944
1945
1946    mysisko = get (gcf, 'pointer');
1947    set (gcf, 'pointer', 'watch');  #čekatiti''''''''''''''''''''''''''''''''''
1948
1949
1950
1951
1952    plk = cstrcat('Vlož počet harmonických, které se budou kreslit v jejich grafu' ...
1953        ,newline, 'teď je počet kreslených harmonických:'...
1954        ,newline, '    ', num2str(PocKresHarm,'%d'), char(9), char(9)...
1955        ,'tj. po kmitočet ',num2str(Fvypis/1000, "%d"),' kHz'...
1956        ,newline, 'maximální možný počet kresl. harmonických pro zpracovávaný soubor je:'...
1957        ,newline, '    ', num2str(PocHarm,'%d'), char(9), char(9)...
1958        ,'tj. po kmitočet ',num2str(FposlHarm/1000, "%d"),' kHz'...
1959        ,newline, newline, 'Při zadání hodnoty menší naž 20 bude vstup upraven na 20,'...
1960        ,newline, 'při zadání hodnoty větší než obsahuje výpočet ze zpracovávaného'...
1961        ,newline, 'souboru bude vstup upraven na vypočítaný počet harmonických.');
1962
1963
1964    plktit = 'Zadání počtu kreslených harmonických v grafu.';
1965    defaults = {PocKresHarm};
1966    rowscols = [1,10];
```

```
1967      vlozeno = inputdlg (plk ,plktit ,rowscols , defaults);
1968
1969      if isempty(vlozeno)
1970        # vstup zrušen Cancel => vééééén
1971        set (gcf, 'pointer', mysisko);  #ne čekatiti''''''''''''''''''''''''''''''''
1972        return;
1973      endif
1974
1975      if isnan(str2double(vlozeno{1}))
1976        #je to NáN (Not a Number)
1977        vstup = PocKresHarm;
1978      else
1979        #je to muňkové pole(může obsahovat v muňkách různé
1980        #typy dat, tady řitěz), musí se převést na číslo...
1981        vstup = fix(str2double(vlozeno{1}));
1982      endif
1983
1984
1985      if vstup == PocKresHarm
1986        set (gcf, 'pointer', mysisko);  #ne čekatiti''''''''''''''''''''''''''''''''
1987        return;    #nic se nezměnilo => vééééén
1988      endif
1989
1990
1991      if vstup < 20
1992        vstup = 20;  #miminum pro kreslení
1993      endif
1994
1995      if vstup > PocHarm
1996        vstup = PocHarm;
1997      endif
1998
1999      PocKresHarm = vstup;
2000
2001      plk = cstrcat("Změna počtu kreslených harmonických ");
2002      plk = cstrcat(plk, '(teď je ',num2str(PocKresHarm, "%d"), ' tj. po ');
2003      plk = cstrcat(plk, num2str(Fvypis/1000, "%d"),' kHz)');
2004      set (men.nastPocHarm, 'text' , plk);
2005      set (men.kontmenu.nastPocHarm, 'text' , plk);
2006
2007
2008      #přeježení jen nové kreslení harmonických - některé výst. prom. nejsou potřeba
2009      [ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2, men, ...
2010                                          Velpis , [], false, false, false);
2011
2012      set (gcf, 'pointer', mysisko);  #ne čekatiti''''''''''''''''''''''''''''''''
2013
2014
2015    endfunction#.....................................................................
2016
```

```
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026  #.....................................................................
      ........
2027  function [Velpis] = NastavVelPis (monitor);
2028
2029
2030    global Velpistisk;
2031
2032
2033    Velpis = 0.00438 * monitor.sirkaLogic + 4.32;
2034    Velpistisk = 0.00078 * monitor.sirkaLogic + 5.98;
2035
2036
2037  endfunction#..........................................................
      ........
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050  #.....................................................................
      ........
2051  function [signalfilt] = DPButerworth (rad_filtru, signal, Fvzorkovani,
      kmitocet_zlomu);
2052
2053  #dolní Butterworthova propust řádu rad_filtru:
2054  # první řád je -6 dB/oktáávu
2055  # druhý řád je -12 dB/oktáávu
2056  # třetí řád je -18 dB/oktáávu  atd.
2057
2058
2059  if kmitocet_zlomu == 0
2060    smikfrekv = Fvzorkovani/100;
2061  else
2062    smikfrekv = kmitocet_zlomu;
2063  end
2064
2065
2066
2067  if smikfrekv/(Fvzorkovani/2) > 1;
2068    signalfilt = signal; # smikfrekv je mimo povolený interval, nefiltrovat
```

```
2069    else
2070      % návrh Butterworthova filtru:
2071      [b, a] = butter(rad_filtru, smikfrekv/(Fvzorkovani/2), 'low');
2072
2073      signalfilt = filtfilt(b, a, signal); # aplikace oného filtru bez zkreslení
2074    end
2075
2076
2077
2078    endfunction#.....................................................................
2079
2080
2081
2082
2083
2084
2085    #.....................................................................
2086    function NastavButt (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis);
2087
2088
2089      global Fsmik RadBTfiltru JeBTfiltr;
2090
2091
2092      RFtu = RadBTfiltru;
2093      KZtu = Fsmik;
2094      JFtu = JeBTfiltr;
2095
2096
2097
2098      plk1 = cstrcat('Zpracovávaný signál zde bude upraven průchodem přes DP, a potom bude vypočítáno THD'...
2099                  ,newline, '    (vystoupí na zvláštním řádku). Je to kvůli možnosti potlačení vlivu nežádoucích vyšších'...
2100                  ,newline, '    harmonických na THD. THD přes vhodně zvolenou DP by mělo být menší. Butterworthův filtr'...
2101                  ,newline, '    byl zvolen kvůli jeho maximálně ploché (v rámci možností filtrů) charakteristice v propustném pásmu.'...
2102                  ,newline, newline, 'Používat dolní propust (1 = ano, jiné = ne):');
2103
2104
2105
2106
2107
2108      plk2 = cstrcat(newline, newline, 'Řád filtru zde může být v rozsahu 1 až 8,' ...
2109                  ,newline, '    pro 1. řád má DP pokles -6 db/oktávu (-20 dB/dekádu)'...
2110                  ,newline, '    pro 2. řád má DP pokles -12 db/oktávu (-40 dB/dekádu)'...
2111                  ,newline, '    pro 3. řád má DP pokles -18 db/oktávu (-60 dB/dekádu), atd.'...
```

```
2112                ,newline, 'Řád filtru (rozsah 1 až 8):');
2113
2114
2115
2116
2117
2118
2119    plk3 = cstrcat(newline, newline,'Kmitočet zlomu (mezní frekvence,
        Fo) je kmitočet pro pokles o přibližně -3 dB. Zde se zadává v Hz a
        není' ...
2120                ,newline, '    nijak omezován, ovšem aby byly výpočty
                    THD přes DP tzv. jedlé, měl by být vyšší nebo roven
                    alespoň'...
2121                ,newline, '    3. harmonické, tedy např. pro měřený
                    kmitočet 20 kHz by měl být miminálně 60 kHz. Pro nižší
                    Fo bude'...
2122                ,newline, '    vypočítané THD vyšší, což nemusí
                    odpovídat požadavku na potlačení nežádoucích vyšších
                    harmonických.'...
2123                ,newline, '    V opačném případě, když bude zvolené Fo
                    vyšší než nejvyšší harmonické které je možno z daného
                    signálu'...
2124                ,newline, '    spočítat, se dolní propust neuplatní a
                    THD vypočítané přes ni bude stejné, jako THD signálu bez
                    DP.'...
2125                ,newline, 'Kmitočet zlomu [Hz]:');
2126
2127
2128
2129    plkcel = {plk1, plk2, plk3};
2130    plktit = 'Zadání párametrů pro Butterworthovu dolní propust (DP).';
2131
2132    defaults = {JeBTfiltr, RadBTfiltru, Fsmik};
2133    rowscols = [1,10; 1,10; 1,10];
2134    vlozeno = inputdlg (plkcel ,plktit ,rowscols , defaults);
2135
2136
2137    Prekreslit = false;
2138    Blbyvstup = false;
2139    if isempty(vlozeno)
2140      #vstup byl zrušen
2141      Blbyvstup = true;
2142    endif
2143
2144
2145
2146    if Blbyvstup
2147      # když byl vstup zrušen vrátiti původní hodnoty
2148      RadBTfiltru = RFtu ;
2149      Fsmik = KZtu;
2150      JeBTfiltr = JFtu;
2151    else
2152      if isnan(str2double(vlozeno{1}))
2153        JeBTfiltr = JFtu;
2154      else
2155        vstup = fix(str2double(vlozeno{1}));
```

```
2156
2157        if vstup == 1
2158          JeBTfiltr = true;
2159        else
2160          JeBTfiltr = false;
2161        endif
2162
2163        if JeBTfiltr ~= JFtu  # neni ho... pardon, rovno
2164          Prekreslit = true;
2165        endif
2166      endif
2167
2168
2169
2170      if isnan(str2double(vlozeno{2}))
2171        RadBTfiltru = RFtu;
2172      else
2173        vstup = fix(str2double(vlozeno{2}));
2174
2175        if (vstup > 0) & (vstup < 9)
2176          RadBTfiltru = vstup;
2177        else
2178          RadBTfiltru = RFtu;
2179        endif
2180
2181        if RadBTfiltru ~= RFtu
2182          Prekreslit = true;
2183        endif
2184      endif
2185
2186
2187
2188      if isnan(str2double(vlozeno{3}))
2189        Fsmik = KZtu;
2190      else
2191        Fsmik = str2double(vlozeno{3});
2192      endif
2193
2194      if Fsmik ~= KZtu
2195        Prekreslit = true;
2196      endif
2197    endif
2198
2199
2200
2201    if Prekreslit
2202      mysisko = get (gcf, 'pointer');
2203      set (gcf, 'pointer', 'watch');                                    ⏎
      #čekatiti''''''''''''''''''''''''''''''''''''''''
2204      set (men.zaklovl, "enable", 'off');
2205      set (men.zakledit, "enable", 'off');
2206      set (men.hlnast, "enable", 'off');
2207
2208      #některé výstupní proměnnné tady nejsou potřeba
2209      [ ~, ~,  ~, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1,     ⏎
      hgrf2, ...
```

```
                                        men, Velpis, [], false, false,
                                        false);

    set (men.zaklovl, "enable", 'on');
    set (men.zakledit, "enable", 'on');
    set (men.hlnast, "enable", 'on');
    set (gcf, 'pointer', mysisko);
    #nečekatiti''''''''''''''''''''''''''''''''''
  endif


endfunction#.....................................................
........


#.....................................................................
........
function [FTFazeGrf] = faze(realna, imagin)


FTFaze = atan(imagin / realna);    #fáze  v Rad
FTFazeSt = FTFaze * 180.0 / pi;    #fáze  ve °


# doježení fáze na kruh 360° podle kwadrantů, páč arctan je jen +- 90°
# (= průmět na wosy Y) a přepočet fáze ve ° jako průmět na wosu Y,
# pro gřaf - v FTFazeGrf()
if sign(realna) >= 0
  #od 0° do 180°
  FTFazeGrf = 90.0 + FTFazeSt;

else
  #od 180° do 360°
  FTFazeGrf = 270.0 + FTFazeSt;
  if sigfig(FTFazeGrf, 6) == 360.0  #zaokr. na 6 plat. číslic, esi
  bude stačiti
    FTFazeGrf = 0;
  endif

endif

endfunction#.....................................................
........


#.....................................................................
........
function [formplk] = Formnamista(cislo, pocmist)
```

```
2259
2260
2261   #jeli pocmist <=0 tak véén
2262   if pocmist <= 0
2263     formplk = '%4.2f';
2264     return
2265   endif
2266
2267
2268
2269   #jeli cislo = 0 tak véén
2270   if (sigfig(cislo, pocmist) == 0)
2271     formplk = cstrcat('%1.',num2str(pocmist-1,'%d'),'f');
2272     return
2273   endif
2274
2275
2276
2277   mista = log10(abs(cislo));
2278   znaminko = sign(mista);
2279   mista = fix(mista);
2280
2281
2282   if mista>=0
2283     #cislo je >= 0,1
2284
2285     if mista>=pocmist
2286      pred = mista;
2287      za = 0;
2288     else
2289      if znaminko >=0
2290        za = pocmist - mista - 1;  #pred musí být +1 místo pro x,xx
2291      else
2292        za = pocmist - mista;      #ale né pro 0,xx
2293      endif
2294      pred = pocmist - za;
2295
2296     endif
2297   else
2298     #cislo je < 0,1
2299
2300     pred = 1; #nula celá, (0,xx)
2301     za = abs(mista) + pocmist;
2302
2303     if za > 7   #pro víc jak zde daný počet míst za des. tečkou dát ee
      formát
2304       formplk = cstrcat('%1.',num2str(pocmist-1,'%d'),'e');
2305       return
2306     endif
2307
2308   endif
2309
2310
2311   formplk = cstrcat('%',num2str(pred,'%d'),'.',num2str(za,'%d'),'f');
2312
2313
```

```
endfunction#.....................................................
........
#.....................................................................
........
function y = sigfig(x,n,do_round=1)
  #NUF: toto jsem si dovolil tady použít (snad jsem si nedovolil moc...)

  %---------------------------------------------------------------------
  -------
  % Jonathan R. Senning <senning@gordon.edu>
  % Gordon College
  % October 22, 2000
  % January 2015: Revised and renamed from rnd()
  %
  % Usage: y = sigfig( x, n [, do_round] )
  %
  % Returns:
  %       y:    The value of x rounded or chopped to n significant
  figures
  %
  % Parameters:
  %       x:    The value to round or chop
  %       n:    The number of significant decimal places to keep.
  %  do_round:  non-zero for rounding (default), 0 for truncation
  %
  % Rounds or chops x to n significant decimal places
  %---------------------------------------------------------------------
  -------

  % Since we can't take the log of a negative number, we need to
  % strip any negative sign from x and reapply it at the end.  In
  % the event that x is zero, we just want to return zero.

  s = 1;
  if ( x < 0 )
    x = -x;
    s = -1;
  elseif ( x == 0 )
    y = x;
    return
  end

  % Compute the proper power of 10 to multiply x by so that n decimal
  % places are left to the left of the decimal point.

  a = -( floor( log10( x ) ) - ( n - 1 ) );

  % Now multiply x by 10^a, round to the nearest integer, and then
```

```
2363    % multiply by 10^(-a).  Also, don't forget to reapply the sign of x.
2364
2365    if ( do_round ~= 0 )
2366      y = s * round( x * 10^a ) * 10^(-a);
2367    else
2368      y = s * fix( x * 10^a ) * 10^(-a);
2369    end
2370
2371  endfunction#.....................................................
        ........
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381  #.....................................................................
        ........
2382  function [Indprvmax, Indposmax, pockmitu] = UprnaPruchNulou (signal,
        Fvzorkovani);
2383
2384    # úprava na celé kmity od 1. průchodu nulou
        \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2385
2386
2387    # odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2388    [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2389
2390
2391
2392
2393    # podle gpt-4o mimi
2394    # Detekce křížení nuly
2395    pruch_nulou = find(diff(sign(signalfilt)));
2396
2397
2398
2399    if length(pruch_nulou) > 3
2400      #pokus o úpravu na průchody nulou
2401
2402      Indprvmax = pruch_nulou(1);
2403      if mod(length(pruch_nulou), 2) == 0
2404        #sudy
2405        Indposmax =  pruch_nulou(length(pruch_nulou) - 1);
2406       pockmitu = length(pruch_nulou)/2 - 1;
2407      else
2408        #lichy
2409        Indposmax = pruch_nulou(length(pruch_nulou));
2410        pockmitu = (length(pruch_nulou) - 1)/2;
2411      end
2412
2413    else
2414      #preparát zařadit celé filé
```

```
2415
2416        [Indprvmax, Indposmax, pockmitu] = UprnaCeleFile (signal,
            Fvzorkovani)
2417      end
2418
2419
2420    endfunction#.....................................................
        ........
2421
2422
2423
2424
2425
2426
2427
2428    #.....................................................................
        ........
2429    function [Indprvmax, Indposmax, pockmitu] = UprnaCeleodZac (signal,
        Fvzorkovani);
2430
2431    #úprava na celé kmity od začátku filé
        \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2432
2433
2434    #odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2435    [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2436
2437
2438    poc_bodu_sign = length(signal);                    #z celého signálu
2439    Trvani_signalu = poc_bodu_sign / Fvzorkovani;      # v sevteřinách
2440    # Detekce kmitů, tady z filtrovaného signálu
2441    praach=0.15*abs(max(signalfilt)); # prahová hodnota pro detekci, zde
        15% maxima
2442
2443    # indeksy nulových průsečíkúú (takto celých cyklů)
2444    Ind_0prus = find(signalfilt(1:end-1) < praach & signalfilt(2:end) >=
        praach);
2445
2446    Prum_vzd_ind = mean(diff(Ind_0prus));  #průměrný rozdíl indeksú průsečíkú
2447    Cas_1kmitu = Prum_vzd_ind * (1/Fvzorkovani);  # v sevteřinách
2448    pockmitu_skut = Trvani_signalu / Cas_1kmitu;
2449
2450
2451    wocas_kmitu = pockmitu_skut - fix(pockmitu_skut);  #fix ušmikne celé
        číslo
2452    if abs(1 - wocas_kmitu) < 0.0075
2453      wocas_kmitu = 0;
2454      pockmitu_skut = fix(pockmitu_skut) + 1;
2455    end
2456
2457    wocas_v_indexech = fix((Cas_1kmitu * wocas_kmitu) / (1/Fvzorkovani));
2458
2459
2460    Indprvmax = 1;
2461    Indposmax = poc_bodu_sign - wocas_v_indexech ;
2462    pockmitu = fix(pockmitu_skut);  #tady na celé kmity
```

```
2463
2464  endfunction#.....................................................
      ........
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474  #.....................................................................
      ........
2475  function [Indprvmax, Indposmax, pockmitu] = UprnaCeleodMax (signal,
      Fvzorkovani);
2476
2477  #úprava na celé kmity od 1. maximuma
      \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2478
2479
2480
2481  #odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2482  [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2483
2484  MinVzds_picek=fix(length(signalfilt)*0.002);
2485
2486  #takto najde kladné i záporné maximuma, takže do jednoho kmitu jsou DVĚ
2487  [hodn_maxim, ind_maxim] = findpeaks(signalfilt, 'DoubleSided', ...
2488                                      "MinPeakDistance",MinVzds_picek);
2489
2490
2491  #průměrný rozdíl indeksú, zde maximum, a ty jsou na celý kmit DVĚ
2492  Prum_vzd_ind = mean(diff(ind_maxim)) * 2;
2493
2494  poc_bodu_sign = length(signal) - ind_maxim(1);    # tady od prvního
      maximuma
2495  Trvani_signalu = poc_bodu_sign / Fvzorkovani;     # v sevteřinách
2496  Cas_1kmitu = Prum_vzd_ind * (1/Fvzorkovani);      # v sevteřinách
2497  pockmitu_skut = Trvani_signalu / Cas_1kmitu;
2498
2499
2500  Ind_vsech_kmitu = fix(Prum_vzd_ind * fix(pockmitu_skut));
2501
2502
2503  Indprvmax = ind_maxim(1);
2504  Indposmax = Ind_vsech_kmitu + Indprvmax;
2505  pockmitu = fix(pockmitu_skut);  #tady na celé kmity
2506
2507  endfunction#.....................................................
      ........
2508
2509
2510
2511
2512
```

```
2513
2514  #.....
      ........
2515  function [Indprvmax, Indposmax, pockmitu_skut] = UprnaCeleFile
      (signal, Fvzorkovani);
2516
2517  #úprava na celé filé - jen qůli počtu cyklúú
      \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
2518
2519  #pro výpočet počtu kmitů jsou zde pou žita maximuma
2520
2521
2522  #odfiltrování vyšších kmitočtů, zde dolní Butterworthova propust
2523  [signalfilt] = DPButerworth (4, signal, Fvzorkovani,0);
2524
2525
2526  #takto najde kladné i záporné maximuma, takže do jednoho kmitu jsou DVĚ
2527  [hodn_maxim, ind_maxim] = findpeaks(signalfilt, 'DoubleSided');
2528
2529
2530  #průměrný rozdíl indeksú, zde maximum, a ty jsou na celý kmit DVĚ
2531  Prum_vzd_ind = mean(diff(ind_maxim)) * 2;
2532
2533  poc_bodu_sign = length(signal);                    # tady od začátku
2534  Trvani_signalu = poc_bodu_sign / Fvzorkovani;      # v sevteřinách
2535  Cas_1kmitu = Prum_vzd_ind * (1/Fvzorkovani);       # v sevteřinách
2536  pockmitu_skut = Trvani_signalu / Cas_1kmitu;
2537
2538
2539  Indprvmax = 1;
2540  Indposmax = length(signal);
2541
2542  endfunction#.....
      ........
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552  #.....
      ........
2553  function [THD, ss_slozka, pockmitu, index_zakl_harm, f_vektor,
      jednostr_spektrum, ...
2554            fft_signalu_jednostr, Indexy_harmonickych] = FFTNufsTHD
              (signal, Fvzorkovani, kmity_z_fun);
2555
2556
2557  # Výpočet Fourierovy transformace
      ////////////////////////////////////////////
2558  #    částečně s pomocí gpt-4o mimi
2559
2560  poc_bodu_sign = length(signal);                    # ze zpracovávané části
```

```
Trvani_signalu = poc_bodu_sign / Fvzorkovani;     # v sevteřinách


fft_signalu = fft(signal);


dvoustr_spektrum = abs(fft_signalu/poc_bodu_sign);          # dvoustr. spektrum
jednostr_spektrum = dvoustr_spektrum(1:poc_bodu_sign/2+1); # jednostr. spektrum
jednostr_spektrum(2:end-1) = 2 * jednostr_spektrum(2:end-1); # úprava amplitudy
fft_signalu_jednostr = fft_signalu(1:poc_bodu_sign/2+1);   #jednostr. komplexní


# Frekvenční vektor
f_vektor = Fvzorkovani*(0:(poc_bodu_sign/2))/poc_bodu_sign;




# Identifikace dominantních frekvencí, prej jsou > 10%, ale tady je 30%
praach = 0.3 * max(jednostr_spektrum); % nastavení prahu pro identifikaci

# najít vrcholy:
[s_picky, ind_s_picek] = findpeaks(jednostr_spektrum, 'MinPeakHeight', praach);

index_zakl_harm = ind_s_picek(1);


zakladni_frekvence = f_vektor(index_zakl_harm); # základní frekvence


# výpočet počtu kmitů tady, sedí líp pro hodně zkreslené signály,
# pro ně se bude víc lišit od kmity_z_fun
pockmitu = zakladni_frekvence * Trvani_signalu;

if ~ isempty(kmity_z_fun)
  if (kmity_z_fun > 1) & (abs(pockmitu - kmity_z_fun) < 1)
    pockmitu = kmity_z_fun;
  end
endif




# amplituuda základní harmonické
zakl_harmonicka = jednostr_spektrum(index_zakl_harm);

#ss složka
ss_slozka = jednostr_spektrum(1);
```

```
2612
2613    # Výpočet z ostatních harmonických
2614    % obsažené vyšší harmonické:
2615    Indexy_harmonickych = 2:(floor(poc_bodu_sign/2/index_zakl_harm));
2616    Indexy_harmonickych = Indexy_harmonickych .* (index_zakl_harm-1)+1;
2617
2618
2619    souc_harm_na2 = sum(jednostr_spektrum(Indexy_harmonickych).^2);
2620
2621    # Výpočet THD v %
2622    THD = 100 * sqrt(souc_harm_na2) / zakl_harmonicka;
2623
2624
2625    #konec výpočtu FFT                                                    ↵
        ////////////////////////////////////////////////////////
2626    endfunction#.....................................................↵
        ........
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639    #*****************************************************************↵
        ********
2640    #*************Hl. program jako fuňkce qůli volání z                ↵
        menu***********************
2641    function [Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT        ↵
        (Cestafile, ...
2642              Jmfile, hgrf1, hgrf2, men, Velpis, pocfile, VolitFile,   ↵
              JenPrekres, Jedojpg);
2643
2644    #POZOR: pokud se dá VolitFile = true a filé bude zvoleno (= vybráno a ↵
        pou žito)
2645    #       bude programem nastavena proměnnná JenPrekres na false, aby se ↵
        změny
2646    #       provedly. (Nezáleží tedy na její hodnotě při vstupu do fuňkce.)
2647
2648
2649
2650    global ZpracDat VsechnyVybery;
2651    global AmplHarmV;
2652
2653    global Fsmik RadBTfiltru JeBTfiltr;
2654
2655    global Osciloskop OscHAN OscOWO;
2656    global plknastpod;
2657    global PocKresHarm PocHarm FposlHarm Fvypis;
2658    global VerzeProg;
2659    global JednHlWosyX JednHlWosyY JednWY;
```

```
2660
2661   global hfig1 HlavPlk;
2662
2663
2664
2665
2666   persistent  THD  ss_slozka  pockmitu  index_zakl_harm  f_vektor;
2667   persistent  jednostr_spektrum  fft_signalu_jednostr Indexy_harm;
2668   persistent  OknoBlack  signBlack  thdnufBlack  jednostr_spektrum_black;
2669   persistent  Indprvmax  Indposmax  kmity_z_fun;
2670
2671   persistent  Popisfile;
2672
2673   persistent  THDF jednostr_spektrumF signalsmik THD2a3;
2674   persistent  KresSmikHarm PrubButter KresPrubBT;
2675
2676   persistent  procint IndVypis;
2677
2678
2679
2680
2681
2682   if  VolitFile
2683
2684
2685     if length(Cestafile) == 0
2686       Cestafile = pwd();     #nynější pracovní adresář
2687     else
2688       #zkusiti, jestli oná cesta ještě je
2689       if exist(Cestafile,'dir') ~= 7;    #dyš vrátí 7 tak je to adresář
2690         Cestafile = pwd();     #nejni, tak dát nynější pracovní adresář
2691       endif
2692     endif
2693
2694
2695
2696
2697     # dobrý popis uuuiiigetfilé je na (sice pro matlab...)
2698     #   https://uk.mathworks.com/help/matlab/ref/uigetfile.html
2699
2700
2701
2702     if compare_versions(version, '10.2.0','==')
2703        # takle to chodilo ve verzi oktáávy 10.2.0 a jeto v ní potřeba,
          jinak
2704        # uíigetfilé po spuštění oktávy občas předvolí adresář prostředí
          oktávy
2705        # a NÉÉ ten, co tomu posílám v Cestafile....
2706        [Jmfile, Cestafilevyb, pocfile] = uigetfile ({'*.csv';'*.txt'},...
2707                        "Vyber vstupní filé (csv nebo txt)",
                            cstrcat(Cestafile,'\'));
2708     else
2709        # a takle to chodí ve verzi oktáávy 10.3.0, preparát je to i pro
          jiné verze
2710        [Jmfile, Cestafilevyb, pocfile] = uigetfile ({'*.csv';'*.txt'},...
2711                        "Vyber vstupní filé (csv nebo txt)",
```

```
                        cstrcat(Cestafile));
2712
2713      endif
2714
2715
2716
2717      if pocfile < 1;
2718        odp = msgbox('Nebylo vybráno žádné filé, můžete ho vybrat z menu
          později.' , 'Výběr filé');
2719        return;        #ve fuňkci je východ takto
2720      else
2721        Cestafile = fullfile(Cestafilevyb);
2722      endif
2723
2724      JenPrekres = false; # je nové filé => překreslit
2725
2726      set(gcf, 'pointer', 'watch');
          #čekatiti''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
2727
2728    endif
2729
2730
2731
2732
2733    if isempty(Jmfile);
2734      set (gcf, 'pointer', 'arrow');  #ne
          čekatiti''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
2735      return;
2736    endif
2737
2738
2739
2740    JmfilesCest = fullfile(Cestafile,Jmfile);
2741
2742
2743    Plk = cstrcat(HlavPlk, ' pro soubor:   ', Jmfile);  #tady stačí jen
          méno filé
2744    set(hfig1, 'name', Plk);
2745
2746
2747
2748
2749    bota = false;
2750    try
2751      datata = importdata (JmfilesCest);  # takto to (snad) načte jedle
          pro různé
2752                                          # palice, tedy pro HANTEK i PRO
                                              OWOUN
2753                                          # ale pro OWOUN jedle jen pro
                                              záznam
2754                                          # jednoho kanálu
2755    catch
2756      bota= true;
2757    end_try_catch
2758
2759
```

```
if bota
  bota = false;
  try
    datata = importdata (JmfilesCest, ',',22); # takto to (snad) načte jedle pro
                      # dvoukanálový záznam OWOUN, ale taky pro simulaci
    if Cmuchret(datata.textdata{1}, 'Model,ADS802A') == 0
      bota= true;
    endif

  catch
    bota= true;
  end_try_catch
endif

if bota
  # záznam OWOUN nebo HANTEK filé to nejni, jestli to nejni simulace

  bota=false;
  try
    datata = importdata (JmfilesCest,',');
  catch
    bota= true;
  end_try_catch
endif

Osciloskop = "";
if bota == false
  if length(datata.textdata) == 3
    # Hantek, má jen 3 řádky palice
    Osciloskop = OscHAN;
  elseif length(datata.textdata) == 22
    # OWOUN má 22 řádků palice
    Osciloskop = OscOWO;
  endif
endif

if length(Osciloskop) == 0;
   plk = cstrcat('Vybrané filé neodpovídá žádnému ze dvou osciloskopů,',newline );
   plk = cstrcat(plk, 'kterých soubory umí tento program číst.',newline);
   plk = cstrcat(plk, '( ',OscHAN,' a ', OscOWO, ' )',newline);
   plk = cstrcat(plk, 'Filé můžete vybrat z menu později.',newline);
   odp = msgbox(plk , 'Výběr filé');

   set(gcf, 'pointer', 'arrow'); #nečekatiti''''''''''''''''''''''''''''''''''''
```

```
2812      return;
2813    endif
2814
2815
2816
2817
2818    # Vytažení signálu a casování ze sloupců v .data:
2819
2820    PlkyOsc = datata.textdata;    # je to muňkové pole
2821    RozsachOscProKan(1) = 0;      # první obsazení, nula by nikdy být nemělo
2822    RozsachOsc = 0;               # první obsazení, nula by nikdy být nemělo
2823    JeSimulace = false;       # jestli data jsou ze simulace (výpočtu)
        nebo z měření
2824    bityAD = 0;                   # simulovaný počet bitů AD převodníku
2825
2826
2827
2828
2829
2830
2831    CislaKanalu = 1;       # u Hantek dycky 1, kanalízace tam ve výpise nejsou
2832    KanalyVeFileVse = '1';
2833    MaxPocKanalu = 10;  # snad bude 10 stačit
2834    KanalyVeFilePole = cell(1,MaxPocKanalu);
2835    KanalyVeFilePole{1} = KanalyVeFileVse;
2836    JednVeFilePole = cell(1,MaxPocKanalu);
2837    JednVeFilePole{1} = KanalyVeFileVse;
2838    OsciloskopModel = '';
2839
2840    JednHlWosyX = 'Čas [';         #přednastavení pro nové filé, dyby tam
        nebylo
2841    JednHlWosyY = 'Amplituda [';  #ale tady bez jednotek, doplní se dáále
2842    JednWY = 'V';                 #natwrrdo jednotky wosy Y
2843
2844
2845
2846    # čtení palice osciloskopu Hantek
        ******************************************
2847    if Cmuchret(Osciloskop, OscHAN) > 0
2848      OsciloskopModel = OscHAN;
2849
2850      for i = 1:length(PlkyOsc);
2851
2852        Jeprekodovani = false;
2853        PlkOscUTF8 = sprintf('%c', unicode2native(PlkyOsc{i}, 'cp1250'));
2854        if Cmuchret(PlkOscUTF8, ',#voltbase=') > 0
2855          try
2856            RozsachOscProKan(1) = str2double(regexp(PlkOscUTF8, ...
2857                      '[-+]?\d*\.?\d+([eE][-+]?\d+)?', 'match', 'once'));
2858            Jeprekodovani = true;
2859          catch
2860            RozsachOscProKan(1) = str2double(regexp(PlkyOsc{i}, ...
2861                      '[-+]?\d*\.?\d+([eE][-+]?\d+)?', 'match', 'once'));
2862            Jeprekodovani = false;
2863          end
2864          PozRoz = i;
```

```
2865        break;
2866      endif
2867    endfor
2868
2869    if RozsachOscProKan(1) > 0
2870       if Jeprekodovani
2871         PlkOscUTF8 = sprintf('%c', unicode2native(PlkyOsc{PozRoz},
           'cp1250'));
2872         Cmuchplk = 'simulov?no';
2873       else
2874         PlkOscUTF8 = PlkyOsc{PozRoz};
2875         Cmuchplk = 'simulováno';
2876       endif
2877       if Cmuchret(PlkOscUTF8, Cmuchplk) > 0
2878         # je simulace
2879         Hausnumera = str2double(regexp(PlkOscUTF8,...
2880                     '[-+]?\d*\.?\d+([eE][-+]?\d+)?', 'match'));
2881         JeSimulace = true;
2882         if length(Hausnumera) >= 3
2883           #je simulováno qantování podle bitů AD převodníku
2884           RozsachOscProKan (1) = Hausnumera(3);
2885           bityAD = Hausnumera(2);
2886         endif
2887       else
2888         # převod ze setin mV Hanteku na V:
2889         RozsachOscProKan(1) = RozsachOscProKan(1) * 1e-5;
2890       endif
2891    endif
2892  endif
2893
2894
2895
2896  # čtení palice osciloskopú OWOUN
     ********************************************
2897  UzJeObs = 0;  # spočtení kolík oných obsadilo, zatím se dáál nepoužívá
2898  if Cmuchret(Osciloskop, OscOWO) > 0
2899    for i = 1:length(PlkyOsc);
2900      PlkOscUTF8 = PlkyOsc{i};
2901
2902
2903      if Cmuchret(PlkOscUTF8, 'Model,') > 0
2904        #přečtení modelu osciloskopu
          ******************************************
2905        OsciloskopModel = cstrcat('OWON
          ',PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:end));
2906        UzJeObs = UzJeObs + 1;
2907      endif
2908
2909
2910      if Cmuchret(PlkOscUTF8, 'vertical Scale,') > 0
2911        #přečtení vertikálního rozsahu
          ***************************************
2912
2913        RozsachOscProKan = str2double(regexp(PlkOscUTF8, ...
2914                                    '[-+]?\d*\.?\d+([eE][-+]?\d+)?',
                                      'match'));
```

```
2915
2916
2917        for i = 1:length(RozsachOscProKan);
2918          #můžou být volty nebo mV nebo µV a nic jinšího ?? - byl by rozsach = 0!!
2919          Nasrozs(i) = 0;
2920          if Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:length(PlkOscUTF8)),'mV') > 0
2921            # mV musí být dříf, páč V je tam taky
2922            Nasrozs(i) = 0.001;     #násobitel na V z mV
2923          elseif Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:length(PlkOscUTF8)),'µV') > 0
2924            # µV musí být dříf, páč V je tam taky
2925            Nasrozs(i) = 0.000001;  #násobitel na V z µV
2926          elseif Cmuchret(PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:length(PlkOscUTF8)),'V') > 0
2927            Nasrozs(i) = 1;         #násobitel na V z V
2928          endif
2929          #stínítko má 8 dílků:
2930          RozsachOscProKan(i) = RozsachOscProKan(i) * 8 * Nasrozs(i);
2931        endfor
2932        UzJeObs = UzJeObs + 1;
2933      endif
2934
2935
2936      if Cmuchret(PlkOscUTF8, 'Channel,') > 0
2937        #přečtení počtu zaznamenaných kanalizací
          ******************************
2938
2939        CislaKanalu = str2double(regexp(PlkOscUTF8, ...
2940                    '[-+]?\d*\.?\d+([eE][-+]?\d+)?', 'match'));
2941
2942        KanalyVeFileVse = PlkOscUTF8(Cmuchret(PlkOscUTF8, ',')+1:end);
2943        KanalyVeFile = KanalyVeFileVse;
2944
2945        Carka =  Cmuchret(KanalyVeFile,",");
2946        i = 1;
2947        if Carka > 0
2948          KanalyVeFilePole{i} = KanalyVeFile(1:Carka-1);
2949          do
2950            i= i + 1;
2951            if i > 10   # podle deklarace pole KanalyVeFilePole(1,x)
2952              break;
2953            endif
2954            KanalyVeFile = KanalyVeFile(Carka+1:end);
2955            Carka = Cmuchret(KanalyVeFile,",");
2956            if Carka > 0
2957              KanalyVeFilePole{i} = KanalyVeFile(1:Carka-1);
2958            else
2959              KanalyVeFilePole{i} = KanalyVeFile;
2960            endif
2961          until (Carka == 0) | (Carka == length(KanalyVeFile))
2962        else
2963          KanalyVeFilePole{i} = KanalyVeFile;
2964        endif
2965        UzJeObs = UzJeObs + 1;
```

```
2966      endif
2967
2968
2969      if Cmuchret(PlkOscUTF8, '(Units:') > 0
2970        #přečtení jednotek zaznamenaných dat
          **********************************
2971
2972        JednVeFileVse = PlkOscUTF8;
2973        JednVeFile = JednVeFileVse;
2974
2975        Carka =  Cmuchret(JednVeFile,",");
2976        i = 1;
2977        if Carka > 0
2978          JednVeFilePole{i} = JednVeFile(1:Carka-1);
2979          do
2980            i= i + 1;
2981            if i > 10   # podle deklarace pole JednVeFilePole(1,x)
2982              break;
2983            endif
2984            JednVeFile = JednVeFile(Carka+1:end);
2985            Carka = Cmuchret(JednVeFile,",");
2986            if Carka > 0
2987              JednVeFilePole{i} = JednVeFile(1:Carka-1);
2988            else
2989              JednVeFilePole{i} = JednVeFile;
2990            endif
2991          until (Carka == 0) | (Carka == length(JednVeFile))
2992        else
2993          JednVeFilePole{i} = JednVeFile;
2994        endif
2995        UzJeObs = UzJeObs + 1;
2996      endif
2997
2998
2999    endfor
3000  endif
3001
3002
3003
3004
3005
3006  ZprcKanal = 1;
3007  RozsachOsc = RozsachOscProKan(1);
3008  if length(CislaKanalu) > 1
3009    #je víc kana lízací, dát dotas kterou zpracovat
3010
3011    CelkPocKan = length(CislaKanalu);
3012    plk = cstrcat('Ve zpracovávaném souboru:' ...
3013        ,newline, '   ', JmfilesCest...
3014        ,newline, 'jsou kanály    ',KanalyVeFileVse ...
3015        ,'    tedy celkem ', num2str(CelkPocKan,'%d'), ' kanály.'...
3016        ,newline, 'Vlož číslo kanálu (bez CH, jen číslo),'...
3017        ,' který se bude zpracovávat z tohoto seznamu.'...
3018        ,newline, newline, 'Při zadání hodnoty která není ve výpisu
          kanálů se bude'...
3019        ,' zpracovávat PRVNÍ kanál uvedený ve výpisu.'...
```

```
3020        , newline, 'Když je v souboru více kanálů než ',num2str(MaxPocKanalu,'%d')...
3021        ,' (což nepředpokládám), je sem zařazeno jen prvních '...
3022        ,num2str(MaxPocKanalu,'%d'),' kanálů.', newline);
3023
3024
3025
3026    plktit = 'Zadání měřeného kanálu pro zpracování.';
3027    defaults = {CislaKanalu(1)};
3028    rowscols = [1,5];
3029    vlozeno = inputdlg (plk ,plktit ,rowscols , defaults);
3030
3031    if isempty(vlozeno)
3032      #vstup byl zrušen
3033      ZprcKanal = 1;
3034
3035    elseif (unicode2native(upper(strtrim(vlozeno{1}))(1:1))<48) | ...
3036           (unicode2native(upper(strtrim(vlozeno{1}))(1:1))>57)
3037      #první znak nejni číslo
3038      ZprcKanal = 1;
3039
3040    else
3041      ZprcKanal = fix(str2double(vlozeno{1}));  #číslo zprc. kanálu
3042      if (ZprcKanal < CislaKanalu(1)) | (ZprcKanal > CislaKanalu(length(CislaKanalu)))
3043        ZprcKanal = 1;
3044      else
3045        Priradilo = false;
3046        for i = 1:length(CislaKanalu);
3047          if CislaKanalu(i) == ZprcKanal
3048            ZprcKanal = i;    #toto je pořadí oného v poli CislaKanalu
3049            Priradilo = true;
3050            break;            #výskok z Do, While nebo For smyček
3051          endif
3052        endfor
3053        if ~ Priradilo  # to je NOT Priradilo
3054          ZprcKanal = 1;
3055        endif
3056      endif
3057      RozsachOsc = RozsachOscProKan(ZprcKanal);
3058    endif
3059  endif
3060
3061
3062  signal = datata.data(:,ZprcKanal+1);
3063  casovaniOsc = datata.data(:,1);
3064
3065
3066
3067  if strcmp(Osciloskop, OscOWO)
3068    # doplnění jednotek do popisu wos hl. gřafu podle filé
3069
3070    PorDwojtecky = Cmuchret(JednVeFilePole{1},':');
3071    JednWY = JednVeFilePole{1}(PorDwojtecky + 1:PorDwojtecky + 1);
3072    if Cmuchret(upper(JednWY), 'S') > 0
3073      JednHlWosyX = cstrcat(JednHlWosyX, 'sevteřiny]');
```

```
3074      else
3075        JednHlWosyX = cstrcat(JednHlWosyX, JednWY, ']');
3076      endif
3077
3078      PorDwojtecky = Cmuchret(JednVeFilePole{ZprcKanal+1},':');
3079      JednWY = JednVeFilePole{ZprcKanal+1}(PorDwojtecky+1:PorDwojtecky+1);
3080      JednHlWosyY = cstrcat(JednHlWosyY, JednWY, ']');
3081    else
3082      # doplnění jednotek do popisu wos hl. gřafu natwrdo
3083
3084      JednHlWosyX = cstrcat(JednHlWosyX, 'sevteřiny]');
3085      JednHlWosyY = cstrcat(JednHlWosyY, 'V]');
3086      JednWY = 'V';
3087    endif
3088
3089
3090
3091
3092    CasKrok = casovaniOsc(2)-casovaniOsc(1);    %v sevteřinách
3093    Fvzorkovani = 1/CasKrok;                    %kmitočet vzorkování
3094
3095
3096    if strcmp(Osciloskop, OscHAN)
3097      # osciloskop hantek
3098      casovani(1,:) = 0;
3099      for i = 2:length(signal);
3100        casovani(i,:) = casovani(i - 1,:) + CasKrok;
3101      endfor;
3102
3103    elseif strcmp(Osciloskop, OscOWO)
3104      # osciloskop OWOUN - jeho časování má nulu u prostřed, takže použití
3105      #      úpravy pro hantek i pro OWOUN to udělá od nuly...
3106      casovani = casovaniOsc;
3107    endif
3108
3109
3110
3111
3112
3113
3114
3115    if ~                                                                    ↵
       JenPrekres#------------------------------------------------------------ ↵
       --
3116
3117
3118      #zjištění, jeli možnost použiti všechny výběry pro zpracování filé
3119
3120      #zmenšit na miminální velikost řekněme 2048 dat
3121      vel_cel_sig = length(signal);
3122      koesmik = 2048 / vel_cel_sig;
3123      VsechnyVybery = false;
3124
3125      if koesmik < 1
3126        #redukovat
3127
```

```
3128        redukrok = fix(vel_cel_sig / (vel_cel_sig * koesmik));
3129        redukovany_signal = signal(1:redukrok:end); % každá redukrok hodnota
3130
3131        if abs(max(redukovany_signal)) == 0
3132          # BOTA!!!!!!!!!
3133          plk = cstrcat('Při zpracovávání souboru:' ...
3134            ,newline, '   ', JmfilesCest...
3135            ,newline, 'se vyskytla obuf (redukovaný signál obsahuje samé
              nuly)!!'...
3136            ,newline,newline, 'Tento soubor nelze nyní zpracovat, nejspíš
              nemá '...
3137            ,'pro zpracování vhodný průběh (pulsy, obdélník...).' ...
3138            ,newline, newline, 'Po zavření tohoto upozornění bude program
              UKONČEN !!');
3139
3140          plktit = 'CHYBA ZPRACOVÁNÍ';
3141          handlplk = msgbox(plk , plktit, 'modal');
3142          uiwait(handlplk);     # čeká na zavření
3143
3144          close all;            # smaže fšecky figůůry => ukončí i program
3145        endif
3146
3147        redukovane_casovani = casovani(1:redukrok:end);
3148        Fvzred = 1/(redukovane_casovani(2)-redukovane_casovani(1));
3149
3150        [THDred , ~, pockmitured, ~, ~, ~, ~, ~] = FFTNufsTHD
              (redukovany_signal, Fvzred, []);
3151
3152      else
3153        #je málo dat, redukce nemá smysl, THD spočítat z celého signálu
3154
3155        [THDred , ~, pockmitured, ~, ~, ~, ~, ~] = FFTNufsTHD (signal,
            Fvzorkovani, []);
3156      end
3157
3158      if THDred < 25  #jen při takovém THD mají výběry průběhů smysl,
          jinak kecají
3159        if pockmitured > 2  #a jen dyš je víc kmitů jak aspoň 2
3160          if (vel_cel_sig / pockmitured) > 300
3161          #a jen dyš je víc jak 300 bodů na kmit, jinak neumí jedle vybrat
              začátky
3162            VsechnyVybery = true;
3163          endif
3164        endif
3165      endif
3166
3167
3168
3169
3170
3171
3172
3173      #určení, jak se bude filé zpracovávati
3174      if ~ VsechnyVybery     #to je (if not VsechnyVybery)
3175        # jenom celé filé
3176        ZpracDat = 4;
```

```
3177
3178        if ~ isempty(men)
3179          #znepřístupnění ostatních výběrů, kromě celého filé
3180          set (men.nastpod(1), "enable", 'off');
3181          set (men.nastpod(2), "enable", 'off');
3182          set (men.nastpod(3), "enable", 'off');
3183
3184          set (men.nastpod(1), "checked", 'off');
3185          set (men.nastpod(2), "checked", 'off');
3186          set (men.nastpod(3), "checked", 'off');
3187          set (men.nastpod(4), "checked", 'on');
3188
3189          set (men.nastpod(1), "text", cstrcat('NELZE - ', plknastpod{1}));
3190          set (men.nastpod(2), "text", cstrcat('NELZE - ', plknastpod{2}));
3191          set (men.nastpod(3), "text", cstrcat('NELZE - ', plknastpod{3}));
3192        endif
3193
3194      else
3195        #zpřístupnění fšech výběrů
3196        if ~ isempty(men)
3197          for i = 1: length(men.nastpod)
3198            set(men.nastpod(i), "enable", 'on')
3199            set (men.nastpod(i), "text", cstrcat(plknastpod{i}));
3200          endfor
3201        endif
3202
3203      endif
3204
3205
3206      kmity_z_fun = -999;
3207      switch ZpracDat
3208
3209        case (1)
3210          #nastavení na celé kmity od nuly                                    ↵
            \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3211          [Indprvmax, Indposmax, kmity_z_fun]=UprnaPruchNulou (signal,       ↵
            Fvzorkovani);
3212
3213        case(2)
3214          #nastavení na celé kmity zhruba od maxima                          ↵
            \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3215          [Indprvmax, Indposmax, kmity_z_fun]=UprnaCeleodMax (signal,        ↵
            Fvzorkovani);
3216
3217        case(3)
3218          #nastavení na celé kmity od začátku filé                           ↵
            \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
3219          [Indprvmax, Indposmax, kmity_z_fun]=UprnaCeleodZac (signal,        ↵
            Fvzorkovani);
3220
3221        case(4)
3222          #pro celé filé
3223          [Indprvmax, Indposmax, kmity_z_fun] = UprnaCeleFile (signal,       ↵
            Fvzorkovani);
3224
3225      endswitch
```

```
endif#-------------------------------------------------------------------
--------


#a přestavení podle zvoleného intervalu
signalcast = signal(Indprvmax:Indposmax);
casovanicast = casovani(Indprvmax:Indposmax);









#-------------------------------------------------------------------
--------
# nakreslení změřených dat ve vybraném intervalu
------------------------------


### nastavení figure 1, je v hl programu !!!!!!!!!!!!!!!!!!!!!!!!!!!


# v příkazu subplot je pozice VŽDY normována mezi 0 a 1!!!!!
hgrf1 = subplot("position",[0.05,0.57,0.9,0.38]);  #[x, y, width, height]

plot(casovanicast, signalcast, 'linewidth', 0.4);




set (gca, 'fontsize', Velpis)                %velikost písma v oném
gřafu
set (gca, "xminortick", "on", "yminortick", "on");
set (gca, "xgrid", "on","ygrid", "on");
set (gca, "gridcolor","black");
#"-" plná, "--" čárk., ":" tečk., "-." čerch., "" nic:
set (gca, "gridlinestyle", "-");
grid minor on;
grid on;
ti_tulek = cstrcat('Změřená data ze souboru: ', JmfilesCest);


# z IÁÁ: Octave (při vykreslování textu v grafech) interpretuje určité
znaky
#        jako formátovací sekvence pro subscripts/superscripts.
```

```
Podtržítko "_"
3277  #        je v Octave/gnuplot/Qt interpretováno jako začátek dolního indexu.
3278  #        Takže aby se zobrazilo jedle vypnout interpretaci formátování.
3279  title(ti_tulek,'Interpreter','none');
3280
3281
3282  xlabel(hgrf1 , JednHlWosyX);
3283  ylabel(hgrf1, JednHlWosyY);
3284  hold on # udržení gřafu pro další kreslení do něj
3285
3286
3287
3288
3289  Efhodn = rms(signalcast);  #efektivní hovnota, RMS - root mean square
3290
3291
3292
3293
3294
3295
3296  if ~ JenPrekres#-------------------------------------------------------------
      --
3297    #je to: if NOT JenPrekres
3298
3299    # Výpočty THD a Fourierovy transformace ////////////////////////////////////
3300
3301
3302    [THD, ss_slozka, pockmitu, index_zakl_harm, f_vektor, jednostr_spektrum, ...
3303            fft_signalu_jednostr, Indexy_harm] = FFTNufsTHD (signalcast, ...
                ...
3304            Fvzorkovani, kmity_z_fun);
3305
3306
3307    THD2a3 = 100 * sqrt(jednostr_spektrum(Indexy_harm(1))^2 ...
3308                  + jednostr_spektrum(Indexy_harm(2))^2 ) ...
3309                  / jednostr_spektrum(index_zakl_harm);
3310
3311    PocHarm = length(Indexy_harm);
3312
3313    FposlHarm = f_vektor(Indexy_harm(PocHarm));
3314
3315
3316
3317
3318    if PocHarm < PocKresHarm
3319      PocKresHarm = PocHarm;
3320    endif
3321    IndVypis = round(Indexy_harm(PocKresHarm));
3322    procint = (PocKresHarm/PocHarm) * 100;
3323    Fvypis = round( f_vektor(IndVypis));
3324    if ~ isempty(men)
3325      # změnit taky plky v menu pro zadání počtu kreslených harmounických
```

```
3326        plk = cstrcat("Změna počtu kreslených harmonických ");
3327        plk = cstrcat(plk, '(teď je ',num2str(PocKresHarm, "%d"), ' tj. po
            ');
3328        plk = cstrcat(plk, num2str(Fvypis/1000, "%d"),' kHz)');
3329        set (men.nastPocHarm, 'text' , plk);
3330        set (men.kontmenu.nastPocHarm, 'text' , plk);
3331      endif
3332
3333
3334
3335      KresPrubBT = false;
3336      KresSmikHarm = false;
3337      if JeBTfiltr
3338
3339        signalsmik = DPButerworth (RadBTfiltru, signalcast, Fvzorkovani,
            Fsmik);
3340
3341
3342        [THDF, ~, ~, ~, ~, jednostr_spektrumF, ...
3343                 ~, Indexy_harmF] = FFTNufsTHD (signalsmik, Fvzorkovani,
                    kmity_z_fun);
3344
3345
3346
3347        Fnorm = Fsmik/(Fvzorkovani/2);
3348
3349        if (Fnorm <= 1) && (Fnorm >= 0);  #zdvojení operátoru na && je
            zkratka
3350
3351          % Výpočet frekvenční odezvy Butterwortha, PrubButter je kompleksní
3352          [b, a] = butter(RadBTfiltru, Fsmik/(Fvzorkovani/2), 'low');
3353          [PrubButter, we] = freqz(b, a, f_vektor, Fvzorkovani);
3354
3355          KresPrubBT = true;
3356
3357        endif
3358
3359
3360
3361
3362
3363        #kresliti když je průběh harmonických přes filtr Butterworth aspoň
            trochu
3364        #jiný než průběh harmonických bez filtru, páč dyš se blíží nebo je
            stejné
3365        #tak kreslení průběhu přes filtr přerazí průběh bez filtru a
            vypadá to,
3366        #jakodyby tam nebyl...
3367        KresSmikHarm = false;
3368
3369        prumkrfilt = mean(jednostr_spektrumF(1:IndVypis));
3370        prumkr = mean(jednostr_spektrum(1:IndVypis));
3371        if abs(1 - prumkrfilt/prumkr) * 100 >= 1  #rozdíl v %
3372            KresSmikHarm = true;
3373        endif
3374      endif
```

```
3375
3376
3377
3378
3379
3380
3381
    #.....................................................................
    ........
3382    #výpočet THD pro vstupní signál přes Blackmanovo okno (jako příklad)
3383
3384
3385    poc_bodu_sign = length(signalcast);
3386
3387    #výpočet koe ficyjentů wokenic
3388    OknoBlack = blackman(poc_bodu_sign);
3389
3390    #přepočet fft koe fycyjentama wokenic
3391    #s tečkou (.*) je to násobení element * element, BEZ (jen *) MATICOVÉ
3392    signBlack = signalcast .* OknoBlack;
3393    [thdnufBlack, ~, ~, ~, ~, jednostr_spektrum_black, ~, ~] =
    FFTNufsTHD ...
3394                                    (signBlack, Fvzorkovani, kmity_z_fun);
3395
3396
3397
3398  #konec výpočtů FFT
    ////////////////////////////////////////////////////////
3399
3400  endif#-------------------------------------------------------------------
    --------
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411  plot(casovanicast, signBlack, 'linewidth', 0.4, 'linestyle', ':',
    'color','black');
3412
3413
3414  # přidání
    legendy++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3415  hlegendy = legend('změřená data' , 'změř. data přes okno Blackman');
3416
3417
3418  set(hlegendy, 'Box', 'off');  # vypnout rámeček a možná i pozadí
3419
3420
3421  if ~ Jedojpg
3422    # výstup na vobra žofku
3423    set(hlegendy, 'fontsize', Velpis*1.04); # 0.97);
```

```
3424   else
3425     # výstup do obrazového filé
3426     set(hlegendy, 'fontsize', Velpis*0.95);
3427   endif
3428
3429   # nastavení pozice legendy relativně k X wose:
3430   nastavPozLegendyRelatkOse(gca, hlegendy, getappdata(gcf,'RelPosLegHL'));
3431   drawnow;
3432
3433   # uložení do appdat figúry potřebných handlů pro HL gřaf
3434   setappdata(gcf, "hLegHL", hlegendy);
3435   setappdata(gcf, "hOsyXHL", gca);
3436
3437
3438   hold off;  #zrušení udržení pro další kreslení
3439   #                                                                    ↵
       ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ ↵
       ++++++
3440
3441
3442   VobsadDilkyWos (hgrf1, min(casovanicast), max(casovanicast),"X");
3443   VobsadDilkyWos (hgrf1, min(signalcast), max(signalcast),"Y");
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455   if                                                                   ↵
       VolitFile#-------------------------------------------------------- ↵
       -
3456     #vymazání starého gřafu harmonických, aby tam při dotazu nesmrďéél:
3457     delete(hgrf2);
3458
3459     plk = cstrcat('Vlož popis (identifikaci) zpracovávaného souboru:');
3460     PopisfileVst = inputdlg (plk ,'Vstup popisu tohoto výpočtu',[1,60],  ↵
       {Popisfile});
3461
3462     if length(PopisfileVst)>0
3463       Popisfile = PopisfileVst{1};
3464     else
3465       if length(Popisfile)==0
3466         Popisfile = "";
3467       endif
3468     end
3469   end                                                                  ↵
       #-------------------------------------------------------------------- ↵
       ---
3470
3471
3472
```

```
#přidání plkú @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
#jsou umísťovány v souřadnicích prvního obrázku, musí být za ním





krajeos = axis();  #krajní body os gřafu
odstrad = 0.05;    #odstup řádků v násobiteli y souřadnice
prvrad = 0.15;     #první řádek v násobiteli y souřadnice qůli nyqvistovi
tabnuf = 0.25;     #něco jako tabulátor - posunuje plk
pocharvyp = 10;    #počet harmonických na výpis
pocplatcislic = 5; #počet platných číslic ve výpise



# trochu posun plků do leva qůli jejich šířce, MUSÍ být uděláno jako poměr
# rozsahů x, páč krajeos(2) se u každého kreslení jaksi mění, čert ví proč..
krajeos(1) = krajeos(1) - 0.015*(krajeos(2)-krajeos(1));



# první řádek
radek = 0;         #číslo vystupovaného řádku (první řádek je 0)

#násl. príkazy text vystoupí text do aktivní figury, v souřadnicích
#os právě kresleného gřafu


plk = cstrcat(Popisfile);
# prvrad-0.01 je zvýšení popisu aby byl trochu oddělen aspoň mizerou
# dyš tu nejde podtržení
text (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad-0.01+odstrad*radek), ...
      plk, 'fontsize', Velpis, 'fontweight', 'bold');


radek++;
if JeSimulace
  plk = cstrcat('Výpočet harmonického zkreslení (THD) pro {\bfsimulovaný signál}:');
else
  if strcmp(Osciloskop, OscOWO)
    plk = cstrcat('Výpočet harmonického zkreslení (THD) pro osciloskop ','{\bf',OsciloskopModel,'}:');
  else
```

```
3522        plk = cstrcat('Výpočet harmonického zkreslení (THD) pro typ dat
            osciloskopu ','{\bf',OsciloskopModel,'}:');
3523      endif
3524    endif
3525    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        plk, 'fontsize', Velpis);
3526
3527
3528
3529
3530
3531
3532    radek++;
3533    plk = cstrcat('Vst. soubor je zpracován od záznamu ',
        num2str(Indprvmax, "%d"),...
3534          ' do ',num2str(round(Indposmax), "%d"), ' a má celkem ',...
3535          num2str(length(signal), "%7.0i"),' záznamů.');
3536    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        plk, 'fontsize', Velpis);
3537
3538
3539
3540
3541    radek++;
3542    plk = cstrcat('Nejvyšší použitelná F (Nyquist) je ');
3543    plk = cstrcat(plk, num2str(sigfig(Fvzorkovani/2000,4),
        Formnamista(Fvzorkovani/2000,4)), ' kHz');
3544    if JeSimulace
3545      if bityAD <= 0
3546         plk = cstrcat(plk, ', je simulace s max. rozlišením.');
3547      else
3548         plk = cstrcat(plk, ', je simulace ');
3549         plk = cstrcat(plk, num2str(bityAD, "%d"), ' bitů z ');
3550         plk = cstrcat(plk, num2str(RozsachOsc, "%d"), ' V.');
3551      endif
3552    else
3553      plk = cstrcat(plk, ', rozsah osciloskopu ');
3554      plk = cstrcat(plk, num2str(RozsachOsc, "%d"), ' V');
3555      if strcmp(Osciloskop, OscOWO)
3556        plk = cstrcat(plk, ', kanál ', KanalyVeFilePole{ZprcKanal});
3557      endif
3558    endif
3559    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        plk, 'fontsize', Velpis);
3560
3561
3562    radek++;
3563    plk = cstrcat('Jsou zpracovány ');
3564    switch ZpracDat
3565      case 1
3566        plk = cstrcat(plk,'celé kmity od prvního průchodu nulou; ');
3567      case 2
3568        plk = cstrcat(plk,'celé kmity od prvního maxima; ');
```

```
3569      case 3
3570        plk = cstrcat(plk,'celé kmity od začátku filé; ');
3571      case 4
3572        plk = cstrcat('Je zpracováno celé filé; ');
3573    endswitch
3574    plk = cstrcat('{\bf', plk ,'}');   #tučný výstup plku
3575
3576
3577
3578    if (pockmitu == 1)
3579      #plkzac = 'Je zpracován ';
3580      plkzac = 'celkem ';
3581      plkkon = ' kmit.';
3582    elseif ((pockmitu > 1) & (pockmitu < 5))  # & je and
3583      plkzac = 'celkem ';
3584      plkkon = ' kmity.';
3585    else
3586      plkzac = 'celkem ';
3587      plkkon = ' kmitů.';
3588    endif
3589    plk = cstrcat(plk,' ', plkzac, num2str(pockmitu, "%d"),plkkon);
3590    text                                                              ↵↵
       (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),  ↵↵
       plk, 'fontsize', Velpis);
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601    radek++;radek++;
3602    plk = cstrcat('Efektivní hodnota (RMS) celého zpracovaného signálu:');
3603    text                                                              ↵↵
       (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),  ↵↵
       plk, 'fontsize', Velpis);
3604    plk = cstrcat(num2str(sigfig(Efhodn,pocplatcislic),               ↵↵
       Formnamista(Efhodn,pocplatcislic)), ' ',JednWY,'rms');
3605    text (krajeos(1) +                                                ↵↵
       (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra↵↵
       d+odstrad*radek), plk, 'fontsize', Velpis);
3606
3607
3608
3609    radek++;
3610    plk = cstrcat('Stejnosměrná složka:');
3611    text                                                              ↵↵
       (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),  ↵↵
       plk, 'fontsize', Velpis);
3612    plk = cstrcat(num2str(sigfig(ss_slozka,pocplatcislic),            ↵↵
       Formnamista(ss_slozka,pocplatcislic)), ' ', JednWY);
3613    text (krajeos(1) +                                                ↵↵
       (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra↵↵
```

```
    d+odstrad*radek), plk, 'fontsize', Velpis);
3614
3615
3616
3617    radek++;
3618    plk = cstrcat('THD zprac. průb. ',num2str(sigfig(PocHarm,8)),'
        harmonických, posl. ',num2str(sigfig(FposlHarm/1000,4)),' kHz:');
3619    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        plk, 'fontsize', Velpis);
3620    plk = cstrcat(num2str(sigfig(THD,pocplatcislic),
        Formnamista(THD,pocplatcislic)), ' %');
3621    if sigfig(THD, pocplatcislic) <= sigfig(thdnufBlack, pocplatcislic)
3622      text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
        rad+odstrad*radek), plk, 'fontsize', Velpis, 'fontweight', 'bold');
3623    else
3624      text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
        rad+odstrad*radek), plk, 'fontsize', Velpis);
3625    endif
3626
3627
3628
3629    radek++;
3630    plk = cstrcat('THD zpracovaný průběh přes okno Blackman:');
3631    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        plk, 'fontsize', Velpis);
3632    plk = cstrcat(num2str(sigfig(thdnufBlack,pocplatcislic),
        Formnamista(thdnufBlack,pocplatcislic)), ' %');
3633    if sigfig(THD, pocplatcislic) >= sigfig(thdnufBlack, pocplatcislic)
3634      text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
        rad+odstrad*radek), plk, 'fontsize', Velpis, 'fontweight', 'bold');
3635    else
3636      text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
        rad+odstrad*radek), plk, 'fontsize', Velpis);
3637    endif
3638
3639
3640
3641
3642    if JeBTfiltr
3643      radek++;
3644      plk = cstrcat(' THD zprac. průb. přes DP Butterworth
        ',num2str(sigfig(RadBTfiltru,8)),'.řádu, Fo
        ',num2str(sigfig(Fsmik/1000,4)),' kHz:');
3645      text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
         plk, 'fontsize', Velpis);
3646      plk = cstrcat(num2str(sigfig(THDF,pocplatcislic),
        Formnamista(THDF,pocplatcislic)), ' %');
3647      text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prv
```

```
        rad+odstrad*radek), plk, 'fontsize', Velpis);
3648    endif
3649
3650
3651
3652    radek++;
3653    plk = cstrcat(' THD vypočítané jen z  2. a 3. harmonické:');
3654    text                                                                ↵
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),  ↵
        plk, 'fontsize', Velpis);
3655    plk = cstrcat(num2str(sigfig(THD2a3,pocplatcislic),                 ↵
        Formnamista(THD2a3,pocplatcislic)), ' %');
3656    text (krajeos(1) +                                                  ↵
        (krajeos(2)-krajeos(1))*tabnuf,krajeos(3)-(krajeos(4)-krajeos(3))*(prvra ↵
        d+odstrad*radek), plk, 'fontsize', Velpis);
3657
3658
3659
3660
3661
3662
3663
3664    radek++;radek++;
3665    plk = cstrcat('{\bfVýpis harmonických} (zde jen do                  ↵
        ',num2str(pocharvyp,"%d"),'. harmonické):');
3666    text                                                                ↵
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),  ↵
        plk, 'fontsize', Velpis)  #, 'fontweight', 'bold');
3667
3668
3669
3670
3671    tabnuf1 = 0.1;
3672    tabnuf2 = 0.175;
3673    tabnuf3 = 0.25;
3674
3675
3676
3677
3678
3679    radek++;
3680    text (krajeos(1) +                                                  ↵
        (krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
        ad+odstrad*radek), 'Frekvence', 'fontsize', Velpis, 'fontweight',       ↵
        'bold');
3681    text (krajeos(1) +                                                  ↵
        (krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
        ad+odstrad*radek), 'Amplituda', 'fontsize', Velpis, 'fontweight',       ↵
        'bold');
3682    text (krajeos(1) +                                                  ↵
        (krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr ↵
        ad+odstrad*radek), 'Fáze' , 'fontsize', Velpis, 'fontweight', 'bold');
3683
3684
3685
3686
```

```
3687
3688
3689    radek++;
3690    text
        (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
        'Zákl. harmonická:', 'fontsize',Velpis);
3691    # freqence
3692    plk = cstrcat(num2str(sigfig(f_vektor(index_zakl_harm),pocplatcislic),
        Formnamista(f_vektor(index_zakl_harm),pocplatcislic)), ' Hz');
3693    text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr
        ad+odstrad*radek), plk, 'fontsize', Velpis)
3694    # ampli túúda
3695    plk =
        cstrcat(num2str(sigfig(jednostr_spektrum(index_zakl_harm),pocplatcislic)
        , Formnamista(jednostr_spektrum(index_zakl_harm),pocplatcislic)),'
        ',JednWY);
3696    text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr
        ad+odstrad*radek), plk, 'fontsize', Velpis)
3697    # fááze
3698    fazev = faze(real(fft_signalu_jednostr(index_zakl_harm)),
        imag(fft_signalu_jednostr(index_zakl_harm)));
3699    plk = cstrcat(num2str(sigfig(fazev, pocplatcislic),
        Formnamista(fazev,pocplatcislic)), ' °');
3700    text (krajeos(1) +
        (krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(prvr
        ad+odstrad*radek), plk , 'fontsize', Velpis)
3701
3702
3703
3704
3705
3706
3707
3708
3709    for i = 2:pocharvyp
3710      radek++;
3711      plk = cstrcat(num2str(i,"%d"),'. harmonická:');
3712      text
          (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek),
           plk, 'fontsize',Velpis);
3713      # freqence
3714      plk =
          cstrcat(num2str(sigfig(f_vektor((index_zakl_harm-1)*i+1),pocplatcislic
          ), Formnamista(f_vektor((index_zakl_harm-1)*i+1),pocplatcislic)), '
          Hz');
3715      text (krajeos(1) +
          (krajeos(2)-krajeos(1))*tabnuf1,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
          vrad+odstrad*radek), plk, 'fontsize', Velpis)
3716      # ampli túúda
3717      plk =
          cstrcat(num2str(sigfig(jednostr_spektrum((index_zakl_harm-1)*i+1),pocp
          latcislic),
          Formnamista(jednostr_spektrum((index_zakl_harm-1)*i+1),pocplatcislic))
          ,' ',JednWY);
```

```
3718      text (krajeos(1) +
          (krajeos(2)-krajeos(1))*tabnuf2,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
          vrad+odstrad*radek), plk, 'fontsize', Velpis)
3719      # fáze
3720      fazev = faze(real(fft_signalu_jednostr((index_zakl_harm-1)*i+1)),
          imag(fft_signalu_jednostr((index_zakl_harm-1)*i+1)));
3721      plk = cstrcat(num2str(sigfig(fazev, pocplatcislic),
          Formnamista(fazev,pocplatcislic)), ' °');
3722      text (krajeos(1) +
          (krajeos(2)-krajeos(1))*tabnuf3,krajeos(3)-(krajeos(4)-krajeos(3))*(pr
          vrad+odstrad*radek), plk , 'fontsize', Velpis)
3723    endfor
3724
3725
3726
3727
3728
3729    radek++;
3730    plk = '© Luděk Ruffer       lruffer@volny.cz';
3731    plk = cstrcat(plk, '       verze programu z ', VerzeProg);
3732    # v příkazu color [r,g,b] se míchají světla: třeba 0,0,0, je černá..
3733    # třeba [0, 0.35, 0.65] dál je tmafší modrá
3734    text
      (krajeos(1),krajeos(3)-(krajeos(4)-krajeos(3))*(prvrad+odstrad*radek)*1.
      02, plk, 'fontsize', Velpis, 'color', [0, 0.35, 0.65]);
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748    # 2. gřaf harmonických a
      Butterwortha!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
3749
3750
3751    if Jedojpg
3752      #trochu výš a nižší, aby se pod oné vešla legenda pro DP Butterworth
3753      # v příkazu subplot je pozice VŽDY normována mezi 0 a 1!!!!!
3754      hgrf2 = subplot("position",[0.37,0.1,0.58,0.35]);  #[x, y, width,
          height]
3755    else
3756      hgrf2 = subplot("position",[0.37,0.08,0.58,0.37]);
3757    endif
3758
3759
3760
3761
3762
3763
```

```
3764
3765  # pro rozhodnutí o wosách (lineární - logaritmické) je rozhodující první
3766  # kreslení, další do stejného gřafu už můžou být plotkem, drží to v
      tom případě
3767  # vosy jak byly nakresleny po prvé...
3768
3769  if AmplHarmV > 0
3770    # ve V
3771    semilogy(f_vektor(1:IndVypis)/1000,
      abs(jednostr_spektrum(1:IndVypis)), 'linewidth', 0.4);
3772  else
3773    # v dB
3774    jednostr_spektrum_dB = 20 *
      log10(jednostr_spektrum/jednostr_spektrum(index_zakl_harm));
3775    plot(f_vektor(1:IndVypis)/1000, (jednostr_spektrum_dB(1:IndVypis)),
      'linewidth', 0.4);
3776  end
3777
3778
3779
3780
3781  set (gca, 'fontsize', Velpis)        %velikost písma v oném gřafu
3782  set (gca, "xminortick", "on")
3783  set (gca, "xgrid", "on","ygrid", "on");
3784  set (gca, "gridcolor","black");
3785  set (gca, "gridlinestyle", "-");
3786
3787  # vlastnost xlimmode MUSÍ být "auto"
3788  set (gca, "xlimmode", "auto");
3789  set (gca, "xlimitmethod", "tight");
3790
3791  grid minor on;
3792  grid on;          #když je až po malé mřížži, udělá velký mřížkový
      čáry plný
3793
3794
3795
3796
3797  plk = cstrcat('Vypočtené harmonické (kresleno do ',...
3798        num2str(sigfig(Fvypis/1000,5), '%d'),' kHz',', tj. prvních ', ...
3799        num2str(PocKresHarm, '%d'),' harmonických a ',...
3800        num2str(sigfig(procint,3), '%d'),'% spočteného intervalu).');
3801  title(plk);
3802
3803
3804
3805  xlabel(hgrf2, 'Frekvence [kHz]');
3806
3807  if AmplHarmV > 0
3808    ylabel(hgrf2, 'Amplituda [V]');
3809  else
3810    ylabel(hgrf2, 'Amplituda [dB]');
3811  end
3812  hold on # udržení gřafu pro další kreslení do něj
      ..............................
3813
```

```
3814
3815
3816
3817
3818  #přidání Blackmannna a harm. přes DP Butterworth, dyš se kreslí
3819
3820  barvaprubBT = [0.45, 0.2, 0.666];
3821
3822  if AmplHarmV > 0
3823    # ve V
3824
3825    plot(f_vektor(1:IndVypis)/1000,                                    ↵
       abs(jednostr_spektrum_black(1:IndVypis)), ...
3826                      'linewidth', 0.5, 'linestyle', ':', 'color','black');
3827
3828    if KresSmikHarm
3829      handlBT = semilogy(f_vektor(1:IndVypis)/1000,                    ↵
       (jednostr_spektrumF(1:IndVypis)), ...
3830                     'linewidth', 0.35, 'linestyle', '-.', 'color','red');
3831
3832      Ywosyrozs = ylim;  # zapamatování si rozsachu wosy y teď
3833
3834
3835      hlegendyBT = legend(handlBT, 'harmonické přes DP Butterworth');
3836
3837      if KresPrubBT
3838        hav = abs(PrubButter);  #prevod na vektor z kompleksu a ve V
3839
3840        #přidání další proměnnné do legendy takto:
3841        plot(f_vektor(1:IndVypis)/1000, hav(1:IndVypis), ...
3842               'linewidth', 0.35, 'linestyle', '--', 'color',          ↵
               barvaprubBT,  ...
3843               "displayname", "charakteristika DP Butterworth");
3844      endif
3845
3846
3847      ylim(Ywosyrozs); # vrácení rozsahu wosy y na hodnoty před        ↵
       kreslením průběhu,
3848                       # aby ho průběh nezvětšoval
3849
3850    endif
3851
3852
3853  else
3854    # v dB
3855
3856    jednostr_spektrum_black_dB = 20 *                                  ↵
       log10(jednostr_spektrum_black/jednostr_spektrum_black(index_zakl_harm) ↵
       );
3857    plot(f_vektor(1:IndVypis)/1000,                                    ↵
       (jednostr_spektrum_black_dB(1:IndVypis)), ...
3858                     'linewidth', 0.5, 'linestyle', ':', 'color','black');
3859
3860
3861    if KresSmikHarm
3862
```

```
3863        jednostr_spektrum_F_dB = 20 *
            log10(jednostr_spektrumF/jednostr_spektrumF(index_zakl_harm));
3864        handlBT = plot(f_vektor(1:IndVypis)/1000,
            (jednostr_spektrum_F_dB(1:IndVypis)), ...
3865                       'linewidth', 0.35, 'linestyle', '--', 'color','red');
3866
3867        Ywosyrozs = ylim;  # zapamatování si rozsachu wosy y teď
3868
3869
3870        hlegendyBT = legend(handlBT, 'harmonické přes DP Butterworth');
3871
3872        if KresPrubBT
3873          hav = 20*log10(abs(PrubButter));  #prevod na vektor z kompleksu
              a v dB
3874
3875          #přidání další proměnnné do legendy takto:
3876          plot(f_vektor(1:IndVypis)/1000, hav(1:IndVypis), ...
3877                 'linewidth', 0.35, 'linestyle', '--', 'color', barvaprubBT,
3878                 "displayname", "charakteristika DP Butterworth");
3879        endif
3880
3881        ylim(Ywosyrozs);  # vrácení rozsahu wosy y na hodnoty před
            kreslením průběhu,
3882                          # aby ho průběh nezvětšoval
3883      endif
3884
3885    end
3886
3887
3888
3889    # a nastavení legendy pro Butterwortha dyš je, jen jednou tu
3890    if KresSmikHarm
3891
3892        set(hlegendyBT, 'Box', 'off');
3893        if Jedojpg
3894          set(hlegendyBT, 'fontsize', Velpis*1.04); # 0.97);
3895        else
3896          set(hlegendyBT, 'fontsize', Velpis*0.97);
3897        endif
3898
3899        nastavPozLegendyRelatkOse(gca, hlegendyBT, getappdata(gcf,
            'RelPosLegBT'));
3900        drawnow;
3901
3902        # uložení do appdat figúry potřebných handlů pro BT gřaf
3903        setappdata(gcf, "hLegBT", hlegendyBT);
3904        setappdata(gcf, "hOsyXBT", gca);
3905
3906    endif
3907
3908    # uložení do appdat figúry proměnné KresSmikHarm pro BT gřaf (jestli
        je legenda)
3909    setappdata(gcf, "KresSmikHarm", KresSmikHarm);
3910
3911
3912    hold off # konec udržení gřafu pro další kreslení do něj
```

```
    ......................
3913
3914
3915    #freqenční wosa by měla být dycky od 0
3916    VobsadDilkyWos (hgrf2, 0, max(f_vektor(1:IndVypis)/1000),"X");
3917    if AmplHarmV == 0
3918      # jen pro dB, zde maximumo by vždy měla být 0
3919      # a miminumo vybrat, je tam víc hodnot
3920
3921      # následující 2 hodnoty se kreslí vždycky, další (snad) miminum nepřesáhnou
3922      minBlack = min(jednostr_spektrum_black_dB(1:IndVypis));
3923      minSig = min(jednostr_spektrum_dB(1:IndVypis));
3924      if KresSmikHarm
3925        minHarm = min(jednostr_spektrum_F_dB(1:IndVypis));
3926      else
3927        minHarm = 9999.9;   #nějaké maximumo které miminumo nemá šanci přelést
3928      endif
3929
3930      VobsadDilkyWos(hgrf2, min([minBlack, minSig, minHarm]), 0,"Y");
3931    endif
3932
3933
3934
3935
3936
3937
3938
3939
3940    #ovládání položek menu
3941    try
3942      #gdyš poprvé nejni men tady ještě obsazené tak následující způsobí botu...
3943      set (men.editpod1, "enable", 'on');
3944      set (men.ovlpod2, "enable", 'on');
3945      set (men.ovlpod3, "enable", 'on');
3946      set (men.nastPocHarm, "enable", 'on');
3947
3948      #a z kontext. menu taky
3949      set (men.kontmenu.pod2, "enable", 'on');
3950      set (men.kontmenu.pod3, "enable", 'on');
3951      set (men.kontmenu.nastPocHarm, "enable", 'on');
3952
3953      # Přiřazení kontextového menu ke FŠEM prfkúm v figúúře, musí tu být znovu,
3954      # páč to nějak zapomene
3955      set (findall(gcf), "uicontextmenu", men.kontmenu.hlav);
3956    end
3957
3958
3959
3960    if VolitFile
3961      #gdyš je překreslení tak si kurzor ovládají oné fuňkce
3962      set(gcf, 'pointer', 'arrow');
      #nečekatiti''''''''''''''''''''''''''''''''''''''''
```

```
3963  endif
3964
3965
3966
3967  endfunction
3968
3969  # konec funkcí ↵
      **********************************************************************
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985  #********************************************************************** ↵
      ********
3986  # hlavní ↵
      program----------------------------------------------------------------
3987  #********************************************************************** ↵
      ********
3988
3989
3990
3991  # Tady, rači hned na začátku (i když to tak nevypadá) je definovaná ↵
      fuňkce,
3992  # která se zavolá při ukončení celé oktávy - je to qůli výmazu filé
3993  # Póówlšelllu, které se používá k indikaci, jeli aktiwní vokno maxima ↵
      lízováno.
3994  atexit ("Oktavasezavira");
3995
3996
3997
3998
3999
4000
4001  #velikost vobra žofky 0 v pikslích
4002  set (0, 'units','pixels')
4003
4004  #Oktaava takto monitoruje jen PRIMÁRNÍ monitor, no je zadarmo...
4005  [velobr] = get(0,"monitorpositions");
4006  dpi = get(0, 'screenpixelsperinch');  #toto celkem odpovídá a je to ↵
      fyzické DPI
4007  obrcmvypsir = 2.54* (velobr(3)-velobr(1))/dpi; #skut. rozměr šířky ↵
      monitoru v cm
4008  obrcmvypvys = 2.54* (velobr(4)-velobr(2))/dpi; #skut. rozměr výšky ↵
      monitoru v cm
4009
```

```
4010
4011
4012
4013
4014
4015
4016
4017  #xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      xxxxxxx
4018  #pokus o načtení ini filé s stejným jménem jako toto spuštěné .m filé
4019  #a s rozšířením .NUF (to snad ještě není)
4020
4021
4022  Cestaulozena = '';
4023  AmplHarmV = 0;
4024  Fsmik = 50000;
4025  RadBTfiltru = 2;
4026  JeBTfiltr = true;
4027  PoziceMinula = [-999, -999, -999, -999];
4028
4029
4030  Jminifile = cstrcat(mfilename,'.nuf');
4031
4032  Pocradfile = 0;
4033  radkyfile = [''];
4034
4035  # Otevření souboru pro čtení
4036  fid = fopen(Jminifile, 'r');
4037
4038
4039  # Kontrola, zda se soubor otevřel správně, kdyby néé, tak fid bude = -1
4040  # a když se neotevřel tak nedělat nic
4041  if fid ~= -1    # není rovno
4042
4043    # Čtení z filé po řádcích
4044    Pocradfile = 0;
4045    while ~feof(fid)
4046      Pocradfile++;
4047      % Načtení jednoho řádku, takto celý řádek (s :):
4048      radkyfile {Pocradfile,1} = fgetl(fid);
4049    end
4050
4051    # Zavření filé
4052    fclose(fid);
4053
4054
4055    # zpracování filé
4056    Cestaulozena = '';  #dyby ve filé nebylo
4057    if Pocradfile > 0
4058      for i = 1 : Pocradfile
4059
4060
4061        pozice =  strfind(radkyfile {i,1}, Iniplk.amplharmveV);
4062        if pozice > 0
4063          pozice = strfind(radkyfile {i,1}, '=');
4064          AmplHarmV = str2num(radkyfile {i,1}(pozice + 1:end));
```

```
4065
4066            if AmplHarmV ~ 0
4067              AmplHarmV = 1;  # V
4068            end
4069          end
4070
4071
4072          pozice =  strfind(radkyfile {i,1}, Iniplk.PocKrHarm);
4073          if pozice > 0
4074            pozice = strfind(radkyfile {i,1}, '=');
4075            PocKresHarm = str2num(radkyfile {i,1}(pozice + 1:end));
4076            if isempty(PocKresHarm)
4077              PocKresHarm = 200;
4078            endif
4079          endif
4080
4081
4082          pozice =  strfind(radkyfile {i,1}, Iniplk.zpracdat);
4083          if pozice > 0
4084            pozice = strfind(radkyfile {i,1}, '=');
4085            ZpracDat = str2num(radkyfile {i,1}(pozice + 1:end));
4086            if isempty(ZpracDat)
4087              ZpracDat = 1;
4088            else
4089              ZpracDat = fix(ZpracDat);
4090              if ZpracDat < 1 | ZpracDat > 4 # & je and; | je or; ~ je       ↵
                  not, cocoti
4091                ZpracDat = 1;
4092              endif
4093            endif
4094          endif
4095
4096
4097          pozice =  strfind(radkyfile {i,1}, Iniplk.cestafile);
4098          if pozice > 0
4099            pozice = strfind(radkyfile {i,1}, '=');
4100            Cestaulozena = strtrim (radkyfile {i,1}(pozice + 1:end));
4101          endif
4102
4103
4104          pozice =  strfind(radkyfile {i,1}, Iniplk.PoziceFig);
4105          if pozice > 0
4106            pozice = strfind(radkyfile {i,1}, '=');
4107            PoziceMinula = str2num (radkyfile {i,1}(pozice + 1:end));
4108          endif
4109
4110
4111          pozice =  strfind(radkyfile {i,1}, Iniplk.pouzBT);
4112          if pozice > 0
4113            pozice = strfind(radkyfile {i,1}, '=');
4114            plk = strtrim (radkyfile {i,1}(pozice + 1:end));
4115            if strcmp(upper(plk), 'ANO')
4116              JeBTfiltr = true;
4117            else
4118              JeBTfiltr = false;
4119            endif
```

```
4120          endif
4121
4122          pozice =  strfind(radkyfile {i,1}, Iniplk.fsmikBT);
4123          if pozice > 0
4124            pozice = strfind(radkyfile {i,1}, '=');
4125            Fsmik = str2num (radkyfile {i,1}(pozice + 1:end));
4126            if isempty(Fsmik)
4127              Fsmik = 50000;
4128            else
4129              Fsmik = fix(Fsmik);
4130            endif
4131          endif
4132
4133          pozice =  strfind(radkyfile {i,1}, Iniplk.radBT);
4134          if pozice > 0
4135            pozice = strfind(radkyfile {i,1}, '=');
4136            RadBTfiltru = str2num (radkyfile {i,1}(pozice + 1:end));
4137            if isempty(RadBTfiltru)
4138              RadBTfiltru = 2;
4139            else
4140              RadBTfiltru = fix(RadBTfiltru);
4141              if RadBTfiltru < 1 | RadBTfiltru > 8
4142                RadBTfiltru = 2;
4143              endif
4144            endif
4145          endif
4146
4147
4148
4149        endfor
4150
4151      endif
4152
4153    endif
4154    #xonec čtení filé
     xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
4155
4156
4157    if exist(Cestaulozena,'dir') == 7    #dyš vrátí 7 tak je to adresář
4158      Cestafile = Cestaulozena;
4159    endif
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170    # načtení hodnot o všech  monitourech jinak, dyš to oktááva neumí,
4171    # až po přečtení ini filé
4172    # ale funguje to jen we Voknech
4173    [Bota, Monitory] = Zjisti_monitory();
4174
```

```
4175
4176   if length(Bota)>0
4177     #ošetření případné obuvi při zjišťování monitourů PowerodeŠelem
4178
4179     plk1 = cstrcat('Při zjišťování monitorů připojených k počítači prostředky mimo Octave se vyskytla chyba:'...
4180                    ,newline, Bota ...
4181                    ,newline,newline, 'Opravte následující hodnoty podle skutečnosti, protože'...
4182                    ,' Oktáva asi neumí přečíst (nebo nevím jak ho zjistit) zvětšení displeje'...
4183                    ,newline, 'v procentech zadané v nastavení Windows - tam se tomu česky we W11'...
4184                    ,' říká Měřítko a je to v Nastavení/Systém/Obrazovka'...
4185                    ,newline, 'a zkontrolujte rozlišení hlavního monitoru, jestli je přečteno dobře.'...
4186                    ,newline, ' (Pokud by zadané hodnoty neodpovídaly skutečnosti, nemusí'...
4187                    ,' se správně zobrazovat hlavní okno programu.)'...
4188                    ,newline, newline, 'Vodorovné rozlišení hlavního displeje v pixelech:');
4189
4190     plk2 = cstrcat(newline, newline, 'Svislé rozlišení hlavního displeje v pixelech:');
4191
4192     plk3 = cstrcat(newline, newline, 'Zvětšení displeje v procentech (v nastavení woken je to Měřítko):');
4193
4194     plkcel = {plk1, plk2, plk3};
4195     plktit = 'Ruční zadání párametrů pro velikost základního okna';
4196
4197
4198     defaults = {velobr(3), velobr(4), 100};
4199     rowscols = [1,10; 1,10; 1,10];
4200     vlozeno = inputdlg (plkcel ,plktit ,rowscols , defaults);
4201
4202
4203     Blbyvstup = false;
4204     if isempty(vlozeno)
4205       #vstup byl zrušen
4206       Blbyvstup = true;
4207     endif
4208
4209
4210     # deklarace pole Hlmonitor
4211     Hlmonitor = struct('sirkaSkut', -999, 'vyskaSkut', -999, 'zvetseni_proc', ...
4212                        -999, 'sirkaLogic', -999, 'vyskaLogic', -999);
4213
4214
4215     if Blbyvstup
4216       # když byl vstup zrušen dáti načtené
4217       Xrozlmon = velobr(3);
4218       Yrozlmon = velobr(4);
4219       ZvetsMon = 100;
4220     else
```

```
4221      if isnan(str2double(vlozeno{1}))
4222        Xrozlmon = velobr(3);
4223      else
4224        Xrozlmon = str2double(vlozeno{1});
4225      endif
4226
4227
4228      if isnan(str2double(vlozeno{2}))
4229        Yrozlmon = velobr(4);
4230      else
4231        Yrozlmon = str2double(vlozeno{2});
4232      endif
4233
4234
4235      if isnan(str2double(vlozeno{3}))
4236        ZvetsMon = 100;
4237      else
4238        ZvetsMon = str2double(vlozeno{3});
4239      endif
4240
4241    endif
4242
4243    # no a nastavení pole Hlmonitor podle vstupu:
4244    Hlmonitor.sirkaLogic = Xrozlmon/ZvetsMon*100;
4245    Hlmonitor.vyskaLogic = Yrozlmon/ZvetsMon*100;
4246    Hlmonitor.zvetseni_proc = ZvetsMon;
4247    Hlmonitor.sirkaSkut = Xrozlmon;
4248    Hlmonitor.vyskaSkut = Yrozlmon;
4249
4250    # a taky pole Monitory pro jeden monitor podle vstupu:
4251    Monitory(1).primarni = 1;
4252    Monitory(1).sirkaLogic = Hlmonitor.sirkaLogic;
4253    Monitory(1).vyskaLogic = Hlmonitor.vyskaLogic;
4254    Monitory(1).zvetseni_proc = Hlmonitor.zvetseni_proc;
4255    Monitory(1).sirkaSkut = Hlmonitor.sirkaSkut;
4256    Monitory(1).vyskaSkut = Hlmonitor.vyskaSkut;
4257    Monitory(1).rucni_vstup = 1;
4258
4259  else
4260    # dyš není bota tak vybrání hlavního monitoru, aby se nemusel dále
       hledati
4261
4262    for i=1 : numel(Monitory)
4263      if Monitory(i).primarni
4264        Hlmonitor.sirkaLogic = Monitory(i).sirkaLogic;
4265        Hlmonitor.vyskaLogic = Monitory(i).vyskaLogic;
4266        Hlmonitor.zvetseni_proc = Monitory(i).zvetseni_proc;
4267        Hlmonitor.sirkaSkut = Monitory(i).sirkaSkut;
4268        Hlmonitor.vyskaSkut = Monitory(i).vyskaSkut;
4269        break;
4270      endif
4271    endfor
4272
4273  #konec ošetření případné obuvi při zjišťování monitourů PowerodeŠelem
4274  endif
4275
```

```
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285    [Velpis] = NastavVelPis (Hlmonitor);
4286
4287
4288
4289    hfig1 = figure (1, 'numbertitle', 'off' , 'name', HlavPlk, 'visible',
       'off');
4290
4291
4292
4293    # nastavení velikosti figúúry v pikslích podle logické velikosti
       displayjeje
4294    # a zaokrrouhlit na celá čísla, páč piksle jsou celé
4295    set (hfig1, 'units', 'pixels' , 'position', [  ...
4296         round(Hlmonitor.sirkaLogic * ZaklVelFig.X), ...
4297         round(Hlmonitor.vyskaLogic * ZaklVelFig.Y), ...
4298         round(Hlmonitor.sirkaLogic * ZaklVelFig.Sir), ...
4299         round(Hlmonitor.vyskaLogic * ZaklVelFig.Vys)]);
4300    PozfigPiksle = get (hfig1, 'position');
4301
4302
4303    #nastavení normalizace qůli zjištění normalizačních jednotek
4304    set (hfig1, 'units', 'normalized');
4305    Pozfig = get (hfig1, 'position');
4306
4307
4308    # Uložení základní normalízované pozice figuury po spuštění do appdat
       figúry
4309    # (tady aby byla normálízed)
4310    setappdata(hfig1,'ZaklPosFig',Pozfig);
4311
4312
4313    #a dál je to zase v PIKSLÍCH:
4314    set (hfig1,
       'units','pixels','papertype','a4','paperorientation','landscape');
4315    set (hfig1, 'paperunits','centimeters');
4316    #menubar none zruší jejich menu, bude jen moje:
4317    set (hfig1, 'resize','on','menubar','none');
4318
4319    set (hfig1, 'visible', 'on');
4320
4321
4322
4323
4324
4325    # nastavení min a max rozměrů figúry tady V PIKSLÍCH a zaokr. na celá
       čísla:
4326    # nasobSirky je úprava podle mého zkoušení aby se plky v min. okně
```

```
    nepřekrývaly
4327  nasobSirky =  2.2189E-08 * Hlmonitor.sirkaLogic^2 - 1.7899E-04 ...
4328                * Hlmonitor.sirkaLogic + 1.0591E+00;
4329  minSir = round(PozfigPiksle(3)*nasobSirky);
4330  maxSir = round(PozfigPiksle(3));
4331  minVys = round(PozfigPiksle(4)*0.82);
4332  maxVys = round(PozfigPiksle(4));
4333
4334  % a jejich uložení do ÚserData od figúry
4335  set (hfig1, 'UserData', struct('zacX',PozfigPiksle(1),'minSir',minSir,
    ...
4336                                                                    ↵
       'maxSir',maxSir,'zacY',PozfigPiksle(2),'minVys',minVys,'maxVys',max ↵
       Vys));
4337
4338
4339  # Relativní pozice legend vzhledem k x ose (v rel. jednotkách [0 až 1])
4340  #
4341  # Relativní pozice legendy v rámci osy pro gřaf hlav. průběhu
4342  rel_pos_leg = [0.83, -0.23, 0.2, 0.15];
4343  setappdata(hfig1,'RelPosLegHL',rel_pos_leg);  #a její uložení do
    appdat figúry
4344
4345  # Relativní pozice legendy v rámci osy pro gřaf harmonických (je menší)
4346  rel_pos_leg_BT = [0.79, -0.225, 0.2, 0.15];
4347  setappdata(hfig1,'RelPosLegBT',rel_pos_leg_BT); #a její uložení do
    appdat figúry
4348
4349
4350
4351
4352
4353  # a nastavení pozice figúry jaká byla při minulém ukončení programu, jeli
4354  if  all(PoziceMinula > 0)
4355    if PoziceMinula(3) <= PozfigPiksle(3) && PoziceMinula(4) <=
      PozfigPiksle(4)
4356      # Minulou pozici nastavit jen dyš program minule neskončil s
      maximalízovaným
4357      # woknem, páč příkaz k maksima lízaci wokna woktáwe nemá, a
      uložené rozměry
4358      # jsou pro okno maximalízované, a kdyš by se tu nastavily, bude
      wokno větší
4359      # a nepújde se dostat na horní lištu (nebude viděti)
4360      #
4361      # Ale taky kontrolovat počátek Y a výšku figúúry, páč nevím číím,
      ale i tak
4362      # se vobčas zobrazí s horní lištou nahoře mimo vobražofku, což je
      blbé
4363
4364
4365
4366      if PoziceMinula(4) + PoziceMinula(2) > maxVys
4367
4368        if PoziceMinula(4) > maxVys
4369          PoziceMinula(4) = maxVys;
4370        endif
```

```
4371          if PoziceMinula(4) + PoziceMinula(2) > PozfigPiksle(4) +
              PozfigPiksle(2)
4372            PoziceMinula(2) = PozfigPiksle(2);
4373          endif
4374
4375        endif
4376
4377        set (hfig1, 'position', PoziceMinula);
4378
4379      endif
4380    endif
4381
4382
4383
4384
4385
4386
4387    # groot je default graphics systém
4388    # třeba:  get(groot, 'ScreenSize')
4389    #
4390    # gcf   je current figúre
4391    # gca   je currenx axes
4392    # gco   je current objekt
4393
4394    # třeba nastavení defaultního písma:
4395    # set(groot, 'DefaultTextFontName', 'Times New Roman')
4396    # set(groot, 'DefaultTextFontName', 'Arial')
4397    # set(groot, 'DefaultTextFontSize', Velpis)
4398
4399
4400
4401
4402
4403
4404    Jemriz = true;  #po spuštění mřížky jsou, tak takto
4405
4406
4407    # první zavolání ocaď
4408    # při prvním volání ignorovat zatím neobsazene a nefinované vst.
      proměnné pomocí
4409    # prázdného argumentu []
4410    [Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT (Cestafile, [],
      [], [], ...
4411                                                 [], Velpis, [], true, false,
                                                    false);
4412
4413
4414
4415
4416
4417
4418    #*****************************************************************
      ********
4419    #menu, až nakonec, aby už byly figuury a jejich handleové
      ----------------------
4420
```

```
4421  #hl menu - !! v hl menu akcelerátory nějak nefungujou !!!
4422  men.zaklovl = uimenu ("text", "Ovládání DFT");  #, "accelerator", "u");
4423  men.hlnast = uimenu ("text", "Nastavení");
4424  men.zakledit = uimenu ("text", "Editace grafů"); #, "accelerator", "e");
4425
4426
4427
4428
4429
4430
4431
4432  #podmenu
4433  plk = cstrcat(plknastpod{1});
4434  men.nastpod(1) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4435              "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis,
              1)");
4436
4437  plk = cstrcat(plknastpod{2});
4438  men.nastpod(2) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4439              "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis,
              2)");
4440
4441  plk = cstrcat(plknastpod{3});
4442  men.nastpod(3) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4443              "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis,
              3)");
4444
4445  plk = cstrcat(plknastpod{4});
4446  men.nastpod(4) = uimenu (men.hlnast, "text", plk, "menuselectedfcn", ...
4447              "PrejezZprac (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis,
              4)");
4448
4449
4450
4451
4452
4453  plk = cstrcat("Volba výpočtu přes DP Butterworth a její párametry");
4454  men.nastDPButt = uimenu (men.hlnast, "text", plk, "separator", "on", ...
4455              "menuselectedfcn", "NastavButt (Cestafile, Jmfile, hgrf1,
              hgrf2, men, Velpis)");
4456
4457
4458
4459
4460
4461  plk = cstrcat("Přepínač [dB - V] kreslení Y osy grafu harmonických:
      teď je ");
4462  if AmplHarmV == 0
4463    plk = cstrcat(plk, 'v [dB]');
4464  else
4465    plk = cstrcat(plk, 've [V]');
4466  end
4467  men.nastpodharm = uimenu (men.hlnast, "text", plk, "separator", "on", ...
4468             "menuselectedfcn", "ZmenKreslHarm (Cestafile, Jmfile, hgrf1,
             hgrf2, men, Velpis)");
4469
```

```
4470
4471
4472   plk = cstrcat("Změna počtu kreslených harmonických ");
4473   plk = cstrcat(plk, '(teď je ',num2str(PocKresHarm, "%d"), ' tj. po ');
4474   plk = cstrcat(plk, num2str(Fvypis/1000, "%d"),' kHz)');
4475   men.nastPocHarm = uimenu (men.hlnast, "text", plk, ...
4476             "menuselectedfcn", "ZmenPocKreslHarm (Cestafile, Jmfile,
                  hgrf1, hgrf2, men, Velpis)");
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486   men.ovlpod1 = uimenu (men.zaklovl, "label", "Načíst vstupní sou&bor", ...
4487             "accelerator", "b", "menuselectedfcn",...
4488             "[Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT
                  (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, pocfile,
                  true, false, false)");
4489
4490   #vono to de i bez parenta, třeba viz men.nastpod1
4491   men.ovlpod2 = uimenu ("parent", men.zaklovl, "label", "&Uložit jako
      obrázek s menším rozlišením", ...
4492             "accelerator", "u", "separator", "on" , "menuselectedfcn", ...
4493             "[hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile,
                  hgrf1, hgrf2, men, Velpis)");
4494
4495   men.ovlpod3 = uimenu ("parent", men.zaklovl, "label", "U&ložit jako
      obrázek s větším rozlišením", ...
4496             "accelerator", "l", "menuselectedfcn",  "[hgrf1, hgrf2] =
                  UlozdofileVetsi (Cestafile, Jmfile, hgrf1, hgrf2, men,
                  Velpis)");
4497
4498   men.ovlpod4 = uimenu ("parent", men.zaklovl, "label", "Zavřít &toto
      okno", ...
4499             "accelerator", "t","separator", "on" , "menuselectedfcn",
                  "close (hfig1)");
4500
4501
4502
4503   men.editpod1 = uimenu (men.zakledit, "text", "&Mřížky v gřafech",
      "accelerator", "m", ...
4504             "menuselectedfcn", "[Jemriz] = FunNufmriz (Jemriz, hgrf1,
                  hgrf2)");
4505
4506
4507
4508
4509
4510
4511
4512
4513   # a kontekstové menu
```

```
4514
4515  # Vytvoření kontextového menu
4516  men.kontmenu.hlav = uicontextmenu(hfig1);
4517
4518  # Přidání položek do kontextového menu
4519  men.kontmenu.pod1 = uimenu(men.kontmenu.hlav, 'Label', 'Načíst vstupní soubor', ...
4520            "menuselectedfcn", ...
4521            "[Cestafile, Jmfile, pocfile, hgrf1, hgrf2] = VypDFT (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis, pocfile, true, false, false)");
4522
4523  men.kontmenu.pod2 = uimenu(men.kontmenu.hlav, 'Label', 'Uložit jako obrázek s menším rozlišením', ...
4524            "separator", "on" , "menuselectedfcn", ...
4525            "[hgrf1, hgrf2] = UlozdofileMensi (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4526
4527  men.kontmenu.pod3 = uimenu(men.kontmenu.hlav, 'Label', 'Uložit jako obrázek s větším rozlišením', ...
4528            "menuselectedfcn", ...
4529            "[hgrf1, hgrf2] = UlozdofileVetsi (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4530
4531
4532
4533  plk = cstrcat("Volba výpočtu přes DP Butterworth a její párametry");
4534  men.kontmenu.nastDPButt = uimenu (men.kontmenu.hlav, "text", plk, "separator", "on", ...
4535             "menuselectedfcn", "NastavButt (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4536
4537
4538
4539  plk = cstrcat("Přepínač [dB - V] kreslení Y osy grafu harmonických: teď je ");
4540  if AmplHarmV == 0
4541    plk = cstrcat(plk, 'v [dB]');
4542  else
4543    plk = cstrcat(plk, 've [V]');
4544  end
4545  men.kontmenu.nastharm = uimenu (men.kontmenu.hlav, "text", plk, "separator", "on", ...
4546            "menuselectedfcn", "ZmenKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4547
4548
4549
4550  plk = cstrcat("Změna počtu kreslených harmonických ");
4551  plk = cstrcat(plk, '(teď je ',num2str(PocKresHarm, "%d"), ' tj. po ');
4552  plk = cstrcat(plk, num2str(Fvypis/1000, "%d"),' kHz)');
4553  men.kontmenu.nastPocHarm = uimenu (men.kontmenu.hlav, "text", plk, ...
4554             "menuselectedfcn", "ZmenPocKreslHarm (Cestafile, Jmfile, hgrf1, hgrf2, men, Velpis)");
4555
4556
```

```
4557
4558
4559
4560  men.kontmenu.ZavrWokno = uimenu("parent", men.kontmenu.hlav, 'Label',
      'Zavřít toto okno', ...
4561            "separator", "on" , "menuselectedfcn", "close (hfig1)");
4562
4563
4564  # Přiřazení kontextového menu k figúúře a ke FŠEM jejím onýšům
4565  set (findall(hfig1), "uicontextmenu", men.kontmenu.hlav);
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577  #přednastavení menu pro zpracování dat z filé
4578  set(men.nastpod(ZpracDat), "checked", "on")
4579  if ~ VsechnyVybery     #to je (if not VsechnyVybery)
4580    #znepřístupnění ostatních výběrů, kromě celého filé
4581    #tady taky, páč když se obsazuje VsechnyVybery tak menu "men" ještě
      neexistuje
4582    set (men.nastpod(1), "text", cstrcat('NELZE - ', plknastpod{1}));
4583    set (men.nastpod(2), "text", cstrcat('NELZE - ', plknastpod{2}));
4584    set (men.nastpod(3), "text", cstrcat('NELZE - ', plknastpod{3}));
4585
4586    set (men.nastpod(1), "enable", 'off');
4587    set (men.nastpod(2), "enable", 'off');
4588    set (men.nastpod(3), "enable", 'off');
4589
4590  endif
4591
4592
4593
4594
4595  # následující poslouchač pomocí addlistener (od Duck.ia) nějak nefunguje
4596  # addlistener(hFig, 'CloseRequestFcn', @(src, event) closeFigure(src));
4597
4598  # Přidání poslouchače události zavření figure s hfig1, qůli uložení
      nastavení
4599  # do filé k předávání proměnných viz plk ve fuňkci !!!!!!!
4600  set(hfig1, 'CloseRequestFcn', @(src, event) Figurasezavira(src));
4601
4602
4603
4604
4605
4606  # Aktívace omezení rozměrů figůry MUSÍ být až tady (dál od nastavení
      pozice
4607  # figúry podle minula), páč dyš je těsně za nastavením oné pozice tak
4608  # se program s takto nastavenou figúrou spustí, ALE při první změně
```

```
4609  # rozměrů ZDECHNE celá Oktáva !!!!!!!!!!! a čert ví proč...
4610  aktivuj_omez_rozm(hfig1)
4611
4612
4613
4614
4615
4616
4617
4618
4619  if (pocfile < 1);  # znepřístupnění menu pro hotový výstup
4620                     #(třeba editace mříšky) dyš poprvé nejni filé
4621
4622    set (men.editpod1, "enable", 'off');
4623    set (men.ovlpod2, "enable", 'off');
4624    set (men.ovlpod3, "enable", 'off');
4625    set (men.nastPocHarm, "enable", 'off');
4626
4627    #a z kontext. menu taky
4628    set (men.kontmenu.pod2, "enable", 'off');
4629    set (men.kontmenu.pod3, "enable", 'off');
4630    set (men.kontmenu.nastPocHarm, "enable", 'off');
4631
4632  endif
4633
4634
4635
4636  #konec menu----------------------------------------------------------------
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
```