

Řízení mikroposuvů v rovině

Tomáš Košina

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Poděkování doc. RNDr. Vojtěch Křesálek CSc. za vedení bakalářské práce a za zodpovězení všech otázek.

Prohlašuji, že jsem na celé diplomové práci pracoval/a samostatně a použitou literaturu jsem citoval/a.

Ve Zlíně, 02. 06. 2005

.....

Košina Tomáš

Abstrakt

Abstrakt česky

Cílem této práce bylo propojit počítač s analogovým plotrem a ten pak použít k účelům skenování. Následně navrhnout a vypracovat program k řízení plotru, naskenovat hodnoty magnetického pole a toto magnetické pole zobrazit.

V teoretické část této práce byl popsán vývoj programů v Borland Delphi k řízení plotru a zobrazení oskenované oblasti.

Praktické části je ukázka naskenovaného a zobrazeného magnetického pole pomocí vytvořených programů.

Abstrakt ve světovém jazyce

Purposes these work was connect computer with analog plotter and this aparature apply for scanning. After it design and maked program for directed plotter, scanning value magnetic frighten and this magnetic frighten to show.

In teoretical part these work was describe development programs in Borland Delphi to direction plotter and show scannings area.

Practics parts this work is example scanning and show magnetics frighten iv made programs.

OBSAH

OBSAH.....	5
ÚVOD.....	6
I. TEORETICKÁ ČÁST.....	6
1 SEZNÁMENÍ S POUŽITÝMI PŘÍSTROJI, PROGRAMOVACÍM JAZYKEM.....	7
1.1 POUŽITÝ PC	7
1.2 LABORATORNÍ KARTA OD FIRMY ADVANTECH PCL812PG.....	7
1.3 ANALOGOVÝ PLOTTER	9
1.4 PŘÍSTROJ PRO MĚŘENÍ INTENZITY MAGNETICKÉHO POLE LAKESHORE421	11
2 SKENOVÁNÍ, ZÁKLADNÍ POJMY.....	14
3 ZÁKLAD K PROGRAMOVÁNÍ V BORLAND DELPHI	19
4 STRUKTURA POUŽITÝCH DATOVÝCH SOUBORŮ DT1 A DT2.....	21
5 SKENOVÁNÍ MAGNETICKÉHO POLE (PROGRAMOVÁNÍ)	22
5.1 SKENOVÁNÍ, NAČÍTÁNÍ HODNOT PŘES ANALOGOVÝ VÝSTUP.....	22
5.2 SKENOVÁNÍ, NAČÍTÁNÍ HODNOT PŘES RS232.....	34
6 ZOBRAZOVACÍ PROGRAM (PROGRAMOVÁNÍ).....	37
6.1 ÚVOD DO PROGRAMOVÁNÍ V OPENGL	37
6.2 PROGRAMOVÁNÍ ZOBRAZOVACÍHO PROGRAMU	40
7 MANUÁL KE SKENOVÁNÍ PŘES ANALOGOVÝ VSTUP	47
8 MANUÁL KE SKENOVÁNÍ PŘES RS232	56
9 MANUÁL K PROGRAMU ZOBRAZENÍ DAT	57
10 INSTALACE SKENOVACÍHO A ZOBRAZOVACÍHO PROGRAMU	63
1.1 MINIMÁLNÍ POŽADAVKY SYSTÉMU	63
1.2 INSTALACE PROGRAMU.....	63
II. PRAKTICKÁ ČÁST.....	64
11 NASKENOVANÁ DATA A PRÁCE S NIMI V ZOBRAZOVACÍM PROGRAMU	65
11.1 DATA NASKENOVANÁ PŘES ANALOGOVÝ VSTUP	65
11.2 DATA NASKENOVANÁ PŘES RS232:	67
12 POROVNÁNÍ NASKENOVANÝCH DAT ANALOGOVĚ/RS323	77
ZÁVĚR.....	80

ÚVOD

Tato práce měla být zaměřena na mikrovlnné mikroposuvy a spektrální analyzátor FSH3. Bohužel na těchto přístrojích již zpracovávali dva studenti 5. ročníku svou diplomovou práci. Po konzultaci s vedoucím práce (doc. RNDr. Vojtěch Křesálek, CSc) byl pro posuv vybrán analogový plotr a k měřícím účelům přístroj LakeShore 421 pro měření magnetického pole, protože aplikace těchto přístrojů byla podobná.

Tyto programy pomohou lépe řešit konstrukce magnetických filtrů, magnetů pro elektromotory a konstrukce samotných jader elektromotorů a v neposlední řadě skenování a zobrazení nebezpečných magnetických polí vyzařovaných počítači a jinou, lidmi používanou elektronikou.

V této práci bylo použito pro řízení posuvů plotru laboratorní karty od firmy Advantech PCL812PG a měřící přístroj LakeShore421 na měření intenzity magnetického pole. Na vytváření zobrazovacího a řídicího programu bylo použito programovacího jazyku Borland Delphi. Pro komunikaci s přístrojem LakeShore421 přes RS232 bylo použito programu Agilent VEE pro.

Při programování zobrazovacího programu byla použita komponenta GL, která slouží pro komunikaci s knihovnami windowsu podporující rozšířené 3D zobrazování grafickými kartami. Tyto knihovny slouží převážně pro rychlé propočty 3D prostorových matic pro zobrazení 3D prostoru na 2D ploše (obrazovce) a obsahují je všechny systémy windows kromě windows3.1 a 3.11 (starších než windows95). V dnešní době se OpenGL používají pro programování 3D prostorových her.

Tímto programem jsem chtěl dokázat, že OpenGL není jen na programování her, ale také je mocným nástrojem pro vědu a výzkum.

I. TEORETICKÁ ČÁST

1 SEZNÁMENÍ S POUŽITÝMI PŘÍSTROJI, PROGRAMOVACÍM JAZYKEM

1.1 Použitý PC

Na vytváření ovládacího a zobrazovacího programu i na skenování byl použit počítač s procesorem 300MHz. Tento počítač nedoporučuji pro programování matic větších než 5000x5000 bodů. Výpočetní rychlost a kapacita paměti RAM počítače je již nedostačující.

1.2 Laboratorní karta od firmy Advantech PCL812PG

Pro ovládání plotru byla použita Laboratorní karta od firmy Advantech PCL812PG. Tato karta má 2x 12bit D/A výstupy a 16x 12bit A/D měřících vstupů.

Výstupní porty této karty lze nastavit na výstupu v rozsahu: 0 až +5V, 0 až +10V, 0 až -5V a 0 až -10V.

Vstupní porty na měření v oblasti: +/-10V, +/-5V, +/-2,5V, +/-1,25V, +/-0,625V, +/-0,3125V.

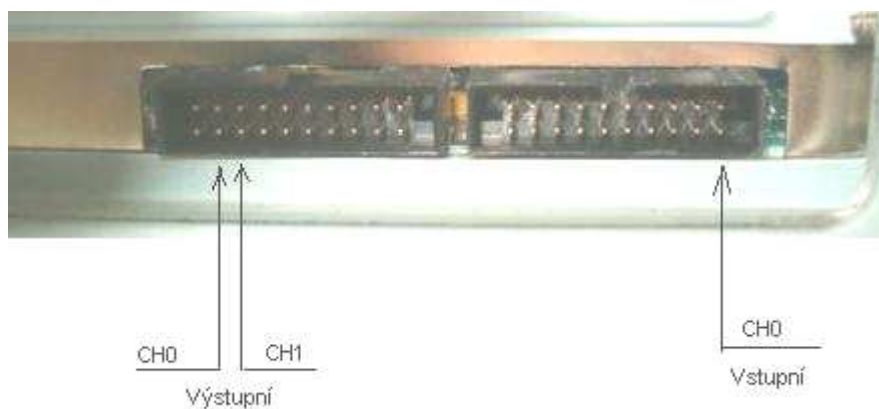
Pro naše účely byla použita nastavení:

- D/A výstup nastaven na 0 až +5V
- A/D vstup na +/-5V

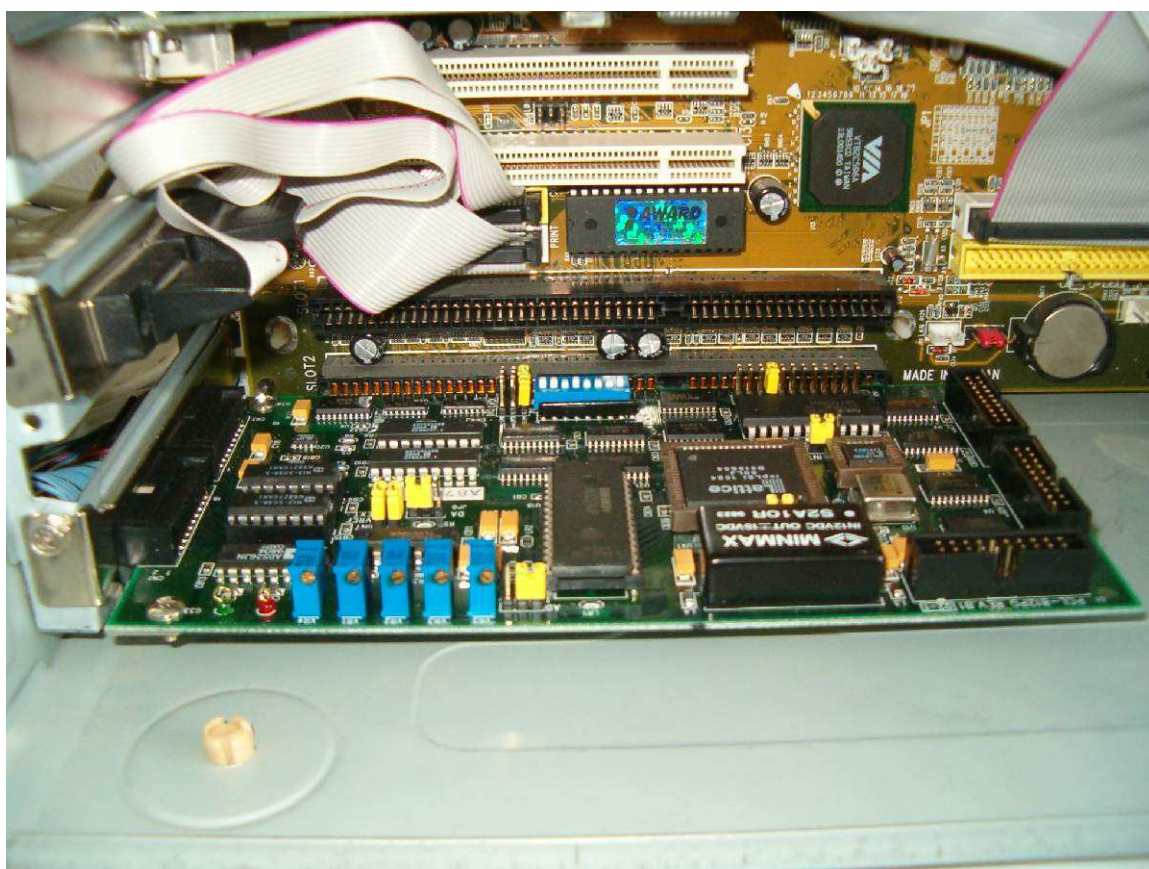
Při tomto rozlišení byl 1inkrement (posuv o 1 bit) na výstupu 0,0012V a rozlišení měření bylo také 0,0012V. (Karta byla ale dosti zašuměná. Vstup se dal měřit zhruba s rozlišením 0,5V). Viz skenování pomocí A/D vstupu. Zašumění karty bylo asi způsobeno neošetřenými vstupy.

Jako D/A výstup byly použity výstupy ch0 a ch1.

Pro měření byl použit vstup ch0.



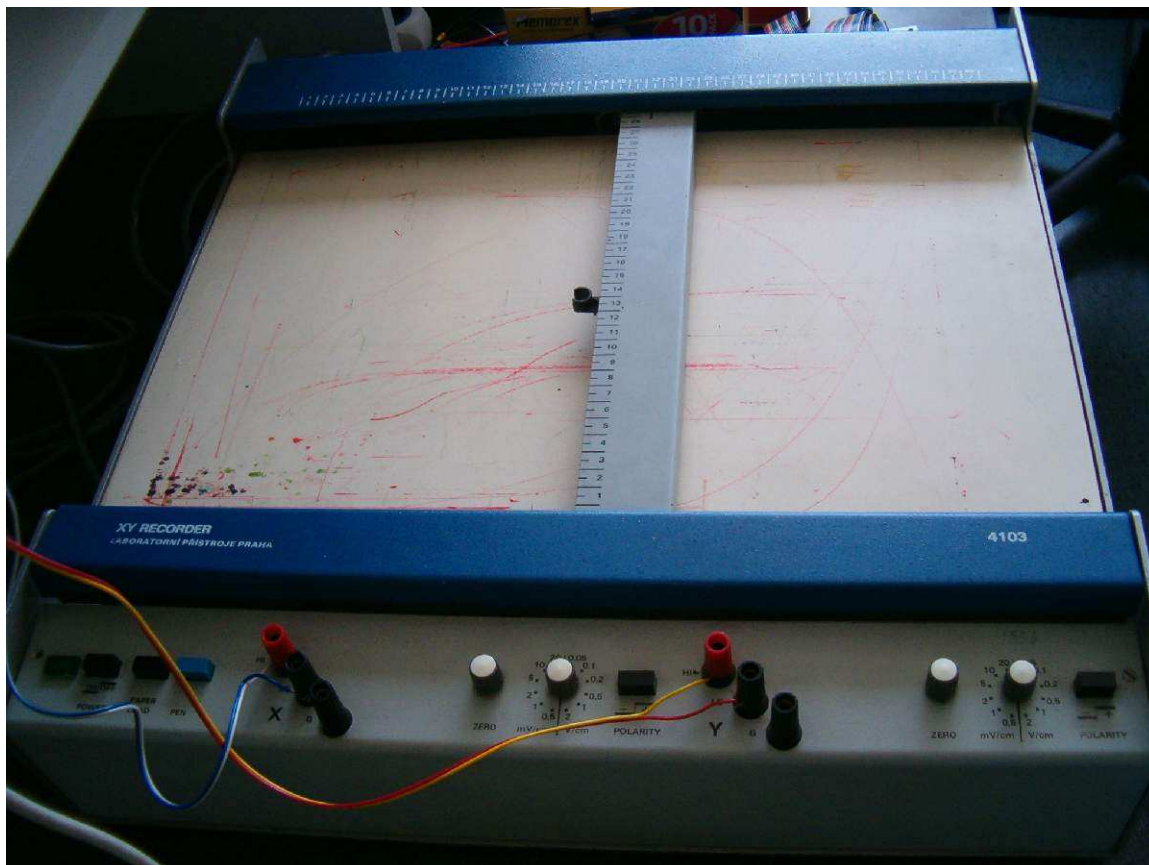
Obr. 1. použité vstupy/výstupy karty PCL812PG



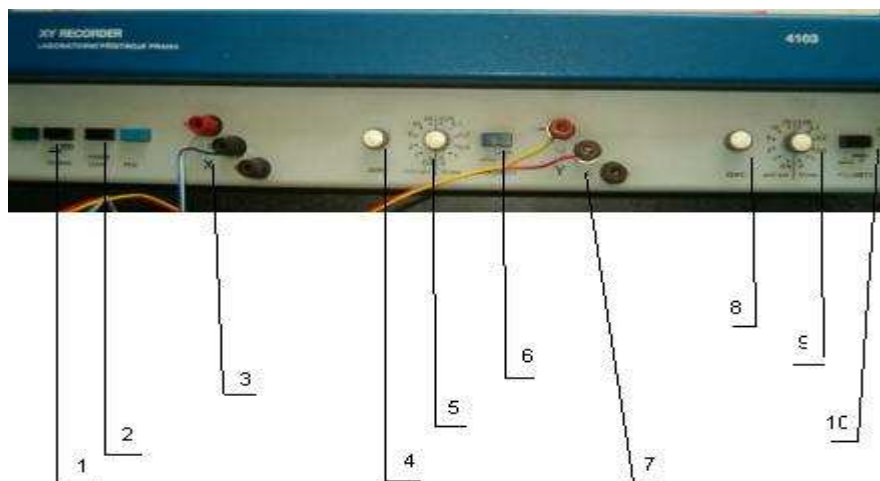
Obr. 2. laboratorní karta PCL812PG

1.3 Analogový plotter

Jako posuv o osách X a Y byl použit analogový plotr, dříve používaný pro vykreslování grafů.



Obr. 3. analogový plotter



Obr. 4. popis plotru

Popis plotru:

1. On/Off
2. Přichycení papíru
3. Analogový vstup osy x
4. Ruční nastavení pozice plotru v ose x
5. Rozlišení v ose x
6. Přepólování osy x
7. Analogový vstup osy y
8. Ruční nastavení pozice plotru v ose y
9. Rozlišení v ose y
10. Přepólování osy y

Rozlišení analogového plotru závisí na odporu vnitřních mechanismů. Experimentálně bylo zjištěno rozlišení přibližně 0,1 mm. Při menším rozlišení byl již posuv nepřesný.

1.4 Přístroj pro měření intenzity magnetického pole LakeShore421

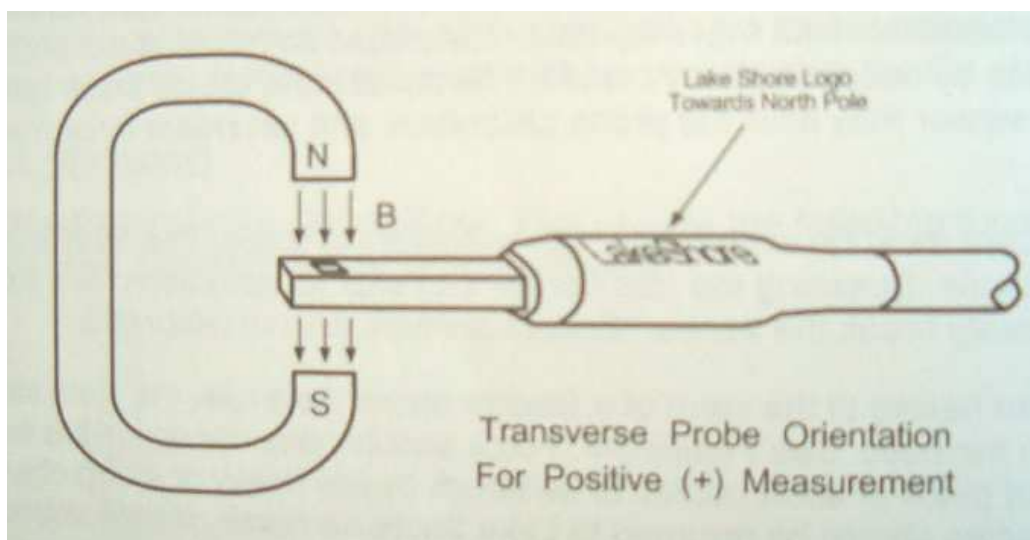
Pro měření intenzity magnetického pole byl použit přístroj LakeShore 421 .



Obr. 5. přístroj LakeShore 421

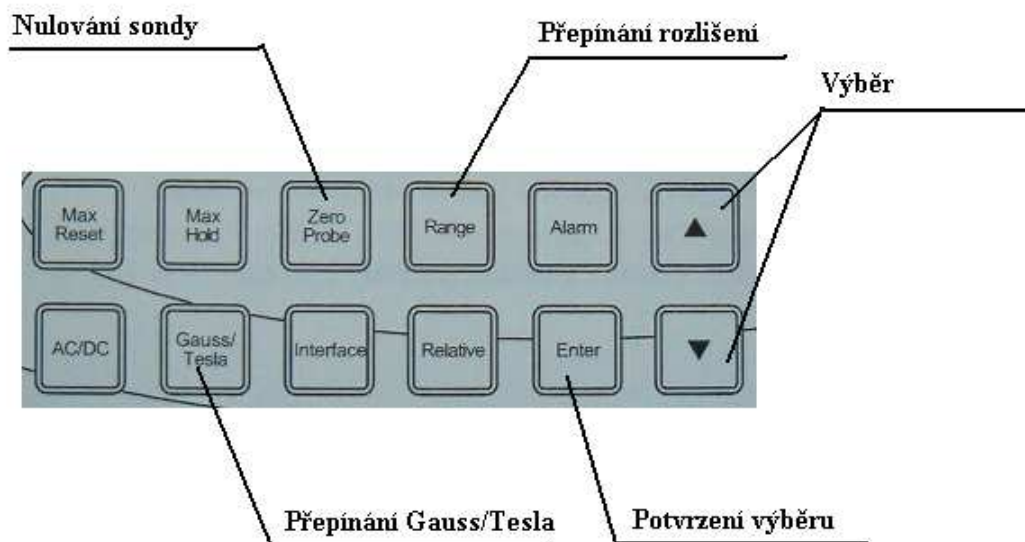
Použitá sonda:

Pro naše účely byla použita sonda pro měření magnetického pole kolmého k ose sondy.



Obr 6. popis použité sondy

Používaná tlačítka pro nastavení Gaussmetru



Obr. 7. používaná tlačítka přístroje LakeShore 421

Rozlišení přístroje LakeShore 421:

nastavené rozlišení	minimální rozlišení	minimální rozlišení analogového výstupu
300kG	0,01kG	0,1kG
30kG	0,001kG	0,01kG
3kG	0,0001kG	0,001kG
300G	0,01G	0,1G
30G	0,001G	0,01G
3G	0,0001G	0,001G
300mG	0,01mG	0,1mG

Tabulka I. přístroje LakeShore 421

Pozn.: Měření bylo prováděno v jednotkách Gauss, které nejsou standardními jednotkami k měření intenzity magnetického pole, protože měření v jednotkách Gauss bylo mnohem přesnější. Standardní jednotky jsou Tesla.

Převodní vztah mezi Gauss a Tesla:

$$1 \text{ [Gauss]} = 10^{-4} \text{ [Tesla]} \quad (1)$$

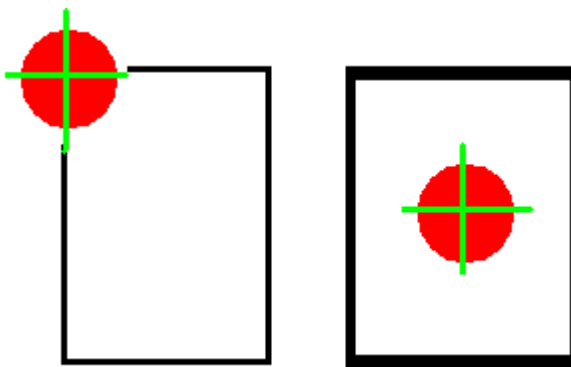
2 SKENOVÁNÍ, ZÁKLADNÍ POJMY

Skenování, je v podstatě pohyb (sondou, nebo pozorovaným předmětem) v jedné nebo dvou osách (X,Y; X,Z; Y,Z) a druhou (třetí) osou je námi skenovaná hodnota. Oskenováním hodnot v jednotlivých bodech nám vznikne dvourozměrný nebo třírozměrný profil pozorované hodnoty. V našem případě bylo použito třírozměrné skenování, pohyb byl řízen v osách XY a osa Z nám zobrazovala naskenovanou hodnotu.

Základní pojmy:

- a) Počáteční bod skenování
- b) Směr (styl) skenování
- c) Rastr x skenování
- d) Rastr y skenování
- e) Celkový počet skenovaných bodů

a) Počáteční bod skenování je bod, od kterého začínáme skenovat bod[0,0].



Obr. 8 ukázka počátečních bodů skenování

Pokud počítač není vybaven snímači polohy, nedokáže detekovat, ve kterém bodě se právě nachází měřící sonda. Proto je velice důležité najed na bod, o který nás počítač požádá. Jedná se o počáteční bod, který volí programátor a od kterého si program odečítá svůj vlastní souřadný systém. Kdyby bylo najetí na počáteční bod špatné, došlo by k oskenování jiné plochy.

b) Směr, styl skenování:

Styl skenování většinou volíme podle:

- konstrukce skenovacího přístroje
- zatížení skenovaného materiálu (vlivem skenování)
- přesnosti skenování
- nebo vlastní úvahy.

Typy třírozměrného skenování mohou být rozděleny do dvou typů:

- **skenování osově nesymetrických veličin**
 - **spirálové**
 - tvarové
 - obdélníkové
 - **přímé**
 - tvarové
 - obdélníkové
- **skenování osově symetrických veličin**
 - skenáž celého řezu
 - skenáž polovičního řezu

Skenování osově nesymetrických veličin

Jedná se o skenování, kdy profil skenované veličiny v ose X je jiný než v ose Y. Je tedy nutné oskenovat všechny body roviny XY.

Spirálové tvarové skenující hrot (měřidlo) se nám po povrchu pohybuje od středu ve tvaru spirály až do dosažení jakéhokoliv předem nadefinovaného tvaru skenované oblasti. Např. kruh, elipsa,..., obrázek.

Spirálové obdélníkové skenující hrot (měřidlo) se nám po povrchu pohybuje od středu ve tvaru spirály až do dosažení obdélníkového tvaru skenované oblasti.

Přímé tvarové skenující hrot (měřidlo) se nám po povrchu pohybuje od okraje ve tvaru přímky buď jednosměrně (měří pouze zleva doprava, nebo zprava doleva) nebo obousměrně (skenuje při pohybu oběma směry) až do dosažení jakéhokoliv předem nadefinovaného tvaru skenované oblasti. Např. kruh, elipsa,..., obrázek.

Přímé obdélníkové skenující hrot (měřidlo) se nám po povrchu pohybuje od okraje ve tvaru přímky buď jednosměrně (měří pouze zleva doprava, nebo zprava doleva) nebo obousměrně (skenuje při pohybu oběma směry) až do dosažení koncového bodu obdélníkové skenované oblasti. Např. kruh, elipsa,..., obrázek.

Skenování osově symetrických veličin

Jedná se o skenování, kdy profil skenované veličiny v ose X je stejný jako v ose Y. Stačí tedy oskenovat jen řez v ose symetrie (po ose X nebo ose Y) a tento řez zrotovat. Tím dosáhneme kompletní obraz skenované oblasti. Toto skenování je velice rychlé, ale musí být přesné najetí střed symetrie.

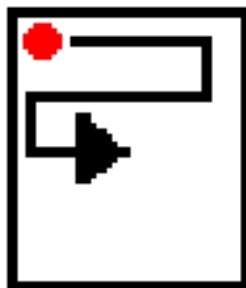
skenáž celého řezu

Oskenujeme celý řez v ose symetrie a ten pootáčíme až do 180° a tím získáme celý profil skenované hodnoty.

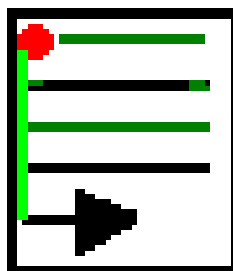
skenáž polovičního řezu

Oskenujeme jen poloviční řez v ose symetrie a ten pootáčíme až do 360° a tím získáme celý profil skenované hodnoty.

Pro naše účely (skenování magnetického pole) bylo zvoleno **Přímé obdélníkové** skenování **měřící oběma směry** a to dvěma způsoby, viz obr. 9 a 10.



Obr. 9
skenování
oboustranně



Obr. 10
skenování
jednostranně

Skenování oboustranně:

Výhody:

- krátký čas skenování

Nevýhody:

- může dojít k nepřesnostem vlivem vůle v posuvech
- nedochází ke kontrole hodnot.

Skenování jednostranně:

Výhody:

- Dochází ke zpětné kontrole; při posuvu $x+$ se hodnoty načítají a při posuvu $x-$ se naměřené hodnoty průměrují s hodnotami uloženými, což má za následek větší přesnost měření

Nevýhody:

- Dvojnásobná doba skenování

c) Rastr x skenování hrubost skenovaných bodů v ose x (i_x)

d) Rastr y skenování hrubost skenovaných bodů v ose y (i_y)

e) Celkový počet skenovaných bodů

$$I_p = i_x \cdot i_y \quad (2)$$

3 ZÁKLAD K PROGRAMOVÁNÍ V BORLAND DELPHI

Programovací jazyk Delphi je velice podobný Pascalu.

Základem programu je komponenta (např. tlačítko), tečka, za ní následuje vlastnost tlačítka, procedura apod.. Příkaz je ukončen středníkem. Příkazy jsou uzavřeny do programu začínající příkazem begin a končící příkazem end jako v Pascalu.

`:=` znamená přiřad

`=` porovnávací člen

Např.

Begin

Button1.caption:= 'OK';

End;

Tlačítku1 vlastnosti nápis předej text OK.(na tlačítko se napíše OK)

Pro přehlednost budu značit ***kurzívou text*** programu a **popis normálně**.

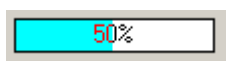
Přehled nejpoužívanějších komponent:

 label: je textové pole, sloužící pro výpis textu na formulář programu

 button: je tlačítko

 radiobutton: slouží pro bodové „vybírání“

 richedit: slouží pro výpis, zápis dlouhých textů, dat apod.

 gauge: procentový ukazatel



spinedit: slouží pro předávání celočíselných hodnot

Form: hlavní formulář programu (programová „deska“, na které jsou rozmístněny tlačítka, labely.... apod.), vizualizační část programu.

Unit: Textové jádro programu (zdrojový soubor).

4 STRUKTURA POUŽITÝCH DATOVÝCH SOUBORŮ DT1 A DT2

Protože při ukládání dat do maticového tvaru bylo zpětné načítání velice pomalé, bylo zvoleno ukládání tak, aby bylo každé číslo zvlášť na řádku. To mnohonásobně urychlilo načítání dat do zobrazovací matice, protože program nemusí na řádku detekovat oddělovací znak (tabelátor, mezeru...) pro další číslo, ale stačí, když načte celý řádek a ten potom převede na číslo.

DT1:

Jedná se o hlavní soubor dat.

```

2,563 1.hodnota z z[1,1]
2,358 2.hodnota z z[2,1]
5,453 3.hodnota z z[3,1]
2,145 5.hodnota z z[4,1]
|      oddělovač (další řada x), y+1
2.564 6.hodnota z z[1,2]
2.356 7.hodnota z z[2,2]
3.241 8.hodnota z z[3,2]
1.235 9.hodnota z z[4,2]
|      oddělovač (další řada x), y+1
....
....
%      konec programu

```

DT2

Jedná se o vedlejší soubor dat (informace o skenování)

```

0,42  krok v ose x
0,42  krok v ose y
40    rozměr v ose x
40    rozměr v ose y
15    max. hodnota v ose z
-10   min. hodnota v ose z
1     zvolený tip skenování (1,2)
Gauss jednotky přístroje

```

5 SKENOVÁNÍ MAGNETICKÉHO POLE (PROGRAMOVÁNÍ)

5.1 Skenování, načítání hodnot přes analogový výstup

Po spuštění programu analog_scan se po kliknutí na vybranou kartu tato karta inicializuje a vynulují se všechny výstupy.

Jako komunikační formulář jsem si vytvořil zobrazující formulář (tfrmrun).



Obr. 11 zobrazení vstupně/výstupních hodnot

Scrollbary jsem si pojmenoval vsclVoltage a vsclVoltage2.

Protože používáme 12bitovou kartu, je minimální hodnota nastavena na 0 a maximální na 4096.

Na změnu potom reagují změnou výstupu na kartě:

```
procedure Tfrmrun.vsclVoltageChange(Sender: TObject);
```

```
var
```

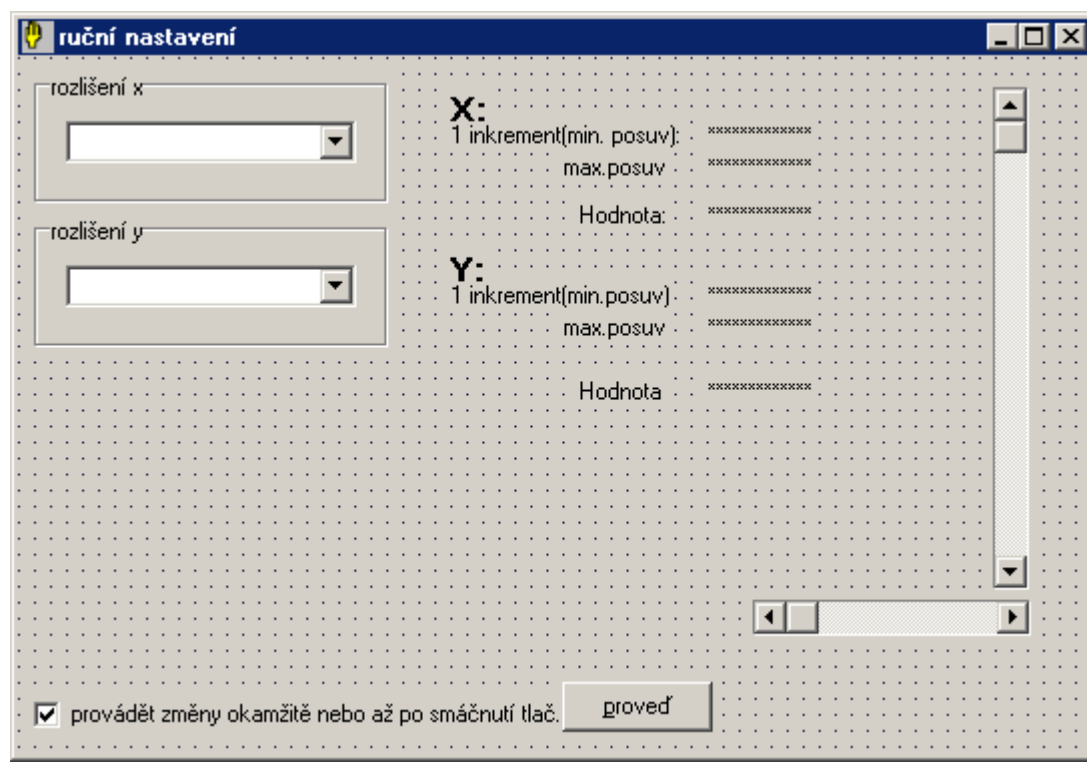
```
    lpAOVoltageOut : PT_AOVoltageOut;
```

```
VsclRange : Longint;  
VolRange : Single;  
VsclOffset : Longint;  
begin  
  VolRange := ptAOConfig.MaxValue - ptAOConfig.MinValue;  
  VsclRange := vsclVoltage.Max - vsclVoltage.Min;  
  VsclOffset := vsclVoltage.Max - vsclVoltage.Position;  
  lpAOVoltageOut.chan := 0;  
  lpAOVoltageOut.OutputValue := VsclOffset / VsclRange * VolRange +  
ptAOConfig.MinValue;  
  ErrCde := DRV_AOVoltageOut(DeviceHandle, lpAOVoltageOut);  
  If (ErrCde <> 0) Then  
    begin  
      DRV_GetErrorMessage(ErrCde, pszErrMsg);  
      Response := Application.MessageBox(pszErrMsg, 'Error!!', MB_OK);  
      DRV_DeviceClose(DeviceHandle);  
      Exit;  
    end;  
  txtVoltage.Text := FloatToStrF((VsclOffset / VsclRange * VolRange +  
ptAOConfig.MinValue), ffFixed, 5,4);  
end;
```

dojde k zadání kanálu, propočtu hodnoty a zápisu hodnoty do *lpAOVoltageOut.OutputValue* což má za následek zápis do knihovny a okamžitou změnu výstupního napětí.

Jestli dojde k chybě, vypíše se chybová hláška.

Nakonec se výstup ve voltech zapíše do editu (*txtVoltage*).

Ruční řízení:

Obr. 12. ruční řízení plotru

Na výběr rozlišení program reaguje:

```

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  if combobox1.Text='0,5 mV/cm' then begin
    label9.Caption:=FloatToStrF(((5/4095)/0.0005)*10,ffixed,5,10)+'mm';
    label10.Caption:=FloatToStrF((5/0.0005)*10,ffixed,5,10)+'mm';
    label11.Caption:=FloatToStrF((rucx.Position*((5/4095)/0.0005))*10,ffixed,5,10)+'mm';
  end;
  if.....
  .....
  .....

```

Jestliže je vybráno rozlišení 0,5mV/cm potom proved':

Label9(výpočet inkrementu)= $5(V)/4095(2^{12})/0,0005(\text{vybrané rozlišení}) \cdot 10(\text{cm} \rightarrow \text{mm})$ a to je zaokrouhleno na 10 míst.

Label10(max.posuv)= $5(V)/0,0005(\text{vybrané rozlišení}) \cdot 10(\text{cm} \rightarrow \text{mm})$ a to je zaokrouhleno na 10 míst.

Label11(momentální hodnota)=poloha posuvníku $x \cdot 1 \text{ inkrement}(\text{label9})$.

.....

.....

Stejně tak pro jiná vybraná rozlišení jen jsou změněna čísla 0,0005 na jiná (dle zvoleného rozlišení).

Následně při posuvu posuvníku: (změna hodnoty)

procedure TForm1.rucxChange(Sender: TObject);

var x:real;

begin

if combobox1.Text='0,5 mV/cm' then x:=0.0005; přiřadí x hodnotu dle zvoleného rozlišení

.....

.....

label11.Caption:=FloatToStrF((rucx.Position((5/4095)/x))*10,ffixed,5,10)+'mm';*

if checkbox1.Checked=true then frmrun.vsclvoltage.position:=4095-rucx.position;

end;

label11=pozice rucx*inkrement ve voltech na 1 bit(5/4095)/rozlišení

Jestliže je dolní políčko zaškrtnuto(změny provádět okamžitě), posuň na frmrun (form výstupů) ukazatel na hodnotu...To způsobí změnu výstupního napětí. Jestliže není zaškrtnuto, potom posuň až po zmáčknutí tlačítka.

Stejný princip je aplikován pro změnu v ose y.

Formulář skenování (1.)

Skenování 1/3

- Zadej rozměry zkoumané plochy

x: mm

y: mm

Zadej nastavené rozlišení

x:

y:

Rozlišení x je:

Rozlišení y je:

1 Inkrement v X je [mm]: Label21

1 Inkrement v Y je [mm]: Label22

Ujistěte se že plotr má dostatek místa k pohybu a zmáčkněte tlač. nuluj

Po dojetí plotru najedzte RUČNÍM POSUVEM NA PLOTRU na střed sledované plochy

-Typ pohybu PLOTU

- Jestliže jsou zadány všechny parametry klikněte na OK pro pokračování

Obr. 13 formulář skenování 1/3

Pro vyhodnocení, jestli jsou rozměry v mezích přístroje program reaguje na změnu vybraného rozlišení:

```
procedure TForm2.ComboBox1Change(Sender: TObject);
```

```
var x:real;
```

```
begin
```

```
if combobox1.Text='2 V/cm' then x:=2;
```

```
.....
```

```
.....
```

```
label21.Caption:=FloatToStrF(((5/4095)/x)*10,ffixed,5,10);  
if StrToFloat(edit1.Text)>StrToFloat(label21.Caption) then begin  
label14.Font.Color:=clgreen;  
label14.Caption:='zvoleno dobře';  
end  
else begin  
label14.Font.Color:=clred;  
label14.Caption:='zvolené rozlišení je nevyhovující,změňte velikost nebo rozlišení';  
end;  
  
if StrToFloat(edit1.text)>StrToFloat(label21.Caption)*4095 then begin  
label23.Font.Color:=clred;  
label23.Caption:='rozměr je příliš velký';  
end  
else begin  
label23.Font.Color:=clgreen;  
label23.Caption:='rozměr je vpořádku';  
end;  
form4.Label9.Caption:=FloatToStrF(((5/4095)/x)*10,ffixed,5,10);  
  
end;
```

Jestliže rozměr je větší než inkrement potom se vypíše, že je rozměr vpořádku. Jestliže tato podmínka nevyhovuje, potom vypíše 'zvolené rozlišení je nevyhovující,změňte velikost nebo rozlišení'.

Jestli je Rozměr větší než 4095*inkrement potom se vypíše 'rozměr je příliš velký'. Naopak vypíše 'rozměr je v pořádku'.

Po zmáčknutí tlačítka nuluj se spustí timer(časovač), pro plynulý posuv. Ten přičítá inkrement v osách x a y tak dlouho, až se výstupy rovnají 2,5V.

To bylo zvoleno jako střed pro počítač, od kterého se vše kalkuluje. Je to proto, abychom vymezili jakékoliv skreslení v oblasti minimálního nebo maximálního napětí karty.

Od této chvíle budou všechny posuvy plynulé, aby se souřadný systém vlivem prokluzů vedení plotru „nerozladil“.

Na tlačítko zvolit se zviditelní formulář pro výběr typu posuvů a po vybrání se zapíše typ:=1 nebo typ:=2.

Po zmáčknutí tlačítka OK se tento formulář zneviditelní a další formulář (skenování 2/3) se zviditelní.

Formulář skenování (2.)

Skenování 2/3

Rozměry obrobku:

X[mm]:

Y[mm]:

Rastr(1 inkrement pohybu):

X[mm]:

Y[mm]:

Krok X: inkrement(y) => 1 krok v X[mm]=

Krok Y: inkrement(y) => 1 krok v Y[mm]=

Rychlost posuvu (posuv o 1 krok ):

/1000 sek

!!rychlost skenování je ovlivněna rychlostí přenosu skenované hodnoty

Odhadovaný čas skenování:

Zade rozlišení skenovacího přístroje:

Zadej počet hodnot pro měření 1 bodu:

Zadej jednotky měřicího přístroje:

Obr. 14 formulář skenování 2/3

Při změně krok x,y se nám násobí rastr s vybranými kroky.

Přitom se nám stále kalkuluje čas

```
ii:=(StrToFloat(form4.Label5.Caption)/StrToFloat(form4.Label10.Caption))/form4.Spinedit3.value;  
i:=(StrToFloat(form4.Label4.Caption)/StrToFloat(form4.Label9.Caption))/form4.Spinedit2.value;  
x:=(((i*spinedit4.value*ii)*(form4.spinedit1.value/1000))/3600  
xx:=floor(x); //hod  
xxx:=ceil((x-floor(x))*60); //min  
form4.Label14.Caption:=FloatToStr(xx)+':'+FloatToStr(xxx)+' :00';  
ii=rozměrx/velikost inkrementu x/počet inkrementů na krok  
i=rozměry/velikost inkrementu y/ počet inkrementů na krok  
x=přepoče doby trvání v hodinách  
xx=čistý čas v hodinách  
xxx=čistý čas v minutách  
label14=hodiny:minuty:00;
```

Po zmáčknutí tlačítka najed' na počátek skenování se nám skener posune na pozice počátečního bodu.

Tento bod můžeme ještě doladit ručním posuvem plotru.

Po zmáčknutí tlačítka Vše nastaveno skenuj, se nám zobrazí 3. formulář a podle vybraného typu skenování se nám zapne buď timer1 nebo timer2.

Formulář skenování (3.)

Obr. 15. formulář skenování 3/3

Při výberu 1. typu skenování:



```

procedure TForm5.Timer1Timer(Sender: TObject);
begin
  i:=i+1;      počítadlo naměřených hodnot
  form5.Gauge1.MaxValue:=ceil(obly);      max. hodnota ukazatele
  label6.Caption:=TimeToStr(TdateTime(now));
  label10.Caption:=IntToStr(x-(2048-(ceil(oblx/2))))+'/' + IntToStr(ceil(oblx))+'
skenováno/z(skenových bodů)';
  label11.Caption:=IntToStr(y-(2048-(ceil(obly/2))))+'/' + IntToStr(ceil(obly))+'
skenováno/z(skenových bodů)';
  gauge1.Progress:=y-(2048-(ceil(obly/2)));

  if i>poc then begin      jestliže je hodnota i větší než počet skenovaných bodů

```

```

if pomm=1 then hodnoty[im]:=((h/poc))*inkr;          pomm=1(pojezd zleva doprava)
ukládej      hodnoty
if pomm=0 then richedit1.Lines.Add(FloatToStrF((((h/poc))*inkr),ffFixed, 5, 2)); pomm=1
(zprava do leva) zapisuj hodnoty
i:=-1;
h:=0;
  if pom2=0 then begin  jestli jsi jinde než nakonci
    x:=x+ix;          posuň se o jedno v x
    pomm:=0;
    im:=0;
    frmrun.vsclVoltage.Position:=ceil(4095-x); zapiš pro změnu na výstupu
    if x>2048+(oblx/2) then begin
      pom2:=1;  jestliže jsi na konci
      y:=y+iy;  posuň se o jedno v y
      richedit1.Lines.Add('|');

      end
    else pom2:=0;
    end;
  if pom2=1 then begin  to samé ale naopak
    x:=x-ix;
    pomm:=1;
    im:=im+1;
    frmrun.vsclVoltage.Position:=ceil(4095-x);
    if x<2048-(ceil(oblx/2)) then begin
      pom2:=0;
      y:=y+iy;
      timer1.Enabled:=false;
      timer3.Enabled:=true;
      end
    else pom2:=1;
    end;
  frmrun.vsclVoltage2.Position:=ceil(4095-y);

```

```

    if (y-(2048-(ceil(obly/2))))>(ceil(obly)) then begin   koncový bod, vypni timer1
        timer1.enabled:=false;
        richedit1.Lines.Add('%');
        konecny;
        end;
    end;
if i<>poc then begin
    frmrun.Button1.OnClick(self);
    h:=h+StrToFloat(frmrun.Edit1.Text);   zapiš hodnotu
    if h>zmax then zmax:=h;   detekuj maximum z
    if h<zmin then zmin:=h;   detekuj minimum z
    end;

end;

```

Při výběru 2. typu skenování:



```

procedure TForm5.Timer2Timer(Sender: TObject);

```

```

begin
    i:=i+1;
    form5.Gauge1.MaxValue:=ceil(obly);
    label6.Caption:=TimeToStr(TdateTime(now));
    label10.Caption:=IntToStr(x-(2048-(ceil(oblx/2))))+'/' + IntToStr(ceil(oblx))+'
skenováno/z(skenových bodů)';
    label11.Caption:=IntToStr(y-(2048-(ceil(obly/2))))+'/' + IntToStr(ceil(obly))+'
skenováno/z(skenových bodů)';

    gauge1.Progress:=y-(2048-(ceil(obly/2)));   ukazatel oskenovaných procent

```

```

if i>poc then begin
if pomm=1 then hodnoty[im]:=(h/poc)*inkr;  (zprava doleva)ukládej
if pomm=0 then richedit1.Lines.Add(FloatToStrF((((h/poc)*inkr)+hodnoty[im])/2,ffFixed,
5, 2));  (zleva doprava) průměruj hodnoty naměřené s uloženými
i:=-1;
h:=0;
  if pom2=0 then begin  ještě nejsi na konci
    x:=x+ix;           posuň se o jeden krok
    im:=im+1;
    frmrun.vsclVoltage.Position:=ceil(4095-x);
    if x>2048+(oblx/2) then begin    už jsi na konci
      pom2:=1;
      pomm:=1;
      richedit1.Lines.Add('|');    vlož oddělovací znaménko
    end
  else pom2:=0;
  end;
  if pom2=1 then begin    to samé ale opačným směrem
    x:=x-ix;
    im:=im-1;
    frmrun.vsclVoltage.Position:=ceil(4095-x);
    if x<2048-(ceil(oblx/2)) then begin
      pom2:=0;
      pomm:=0;
      y:=y+iy;
      richedit1.Lines.Add('|');
    end
  else pom2:=1;
  end;
  frmrun.vsclVoltage2.Position:=ceil(4095-y);
  if (y-(2048-(ceil(obly/2))))>(ceil(obly)) then begin    konečný bod, vypni timer2
    timer2.enabled:=false;
    richedit1.Lines.Add('%'); ukonči program
  end
end

```

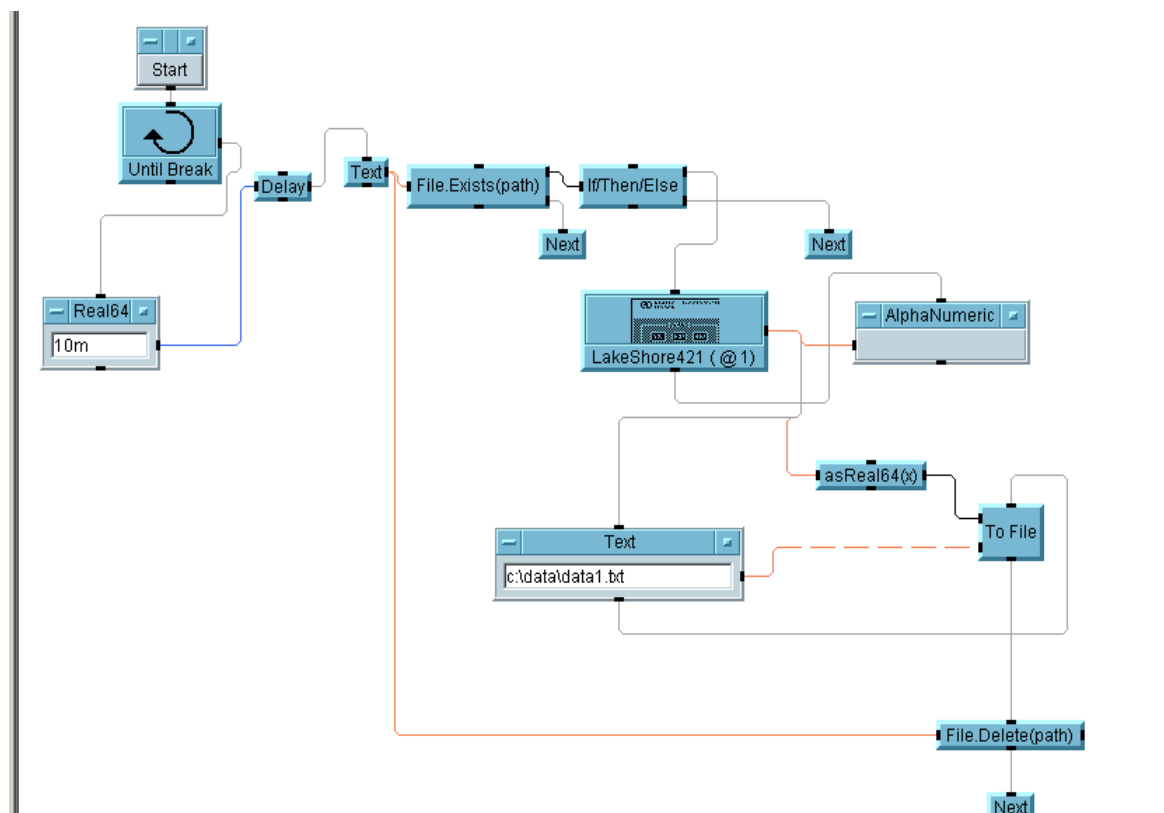
```
                                konecny;  
                                end;  
  
    end;  
  
    if i <> poc then begin  
        frmrun.Button1.OnClick(self);  
        h:=h+StrToFloat(frmrun.Edit1.Text);   načti hodnotu z  
        if h>zmax then zmax:=h;    detekuj max. z  
        if h<zmin then zmin:=h;    detekuj min z  
    end;  
  
end;
```

5.2 Skenování, načítání hodnot přes RS232

Tento program je velice podobný předcházejícímu programu, až na načítání hodnot.

Pokoušel jsem se pomocí různých komponent pro Borland delphi komunikovat s přístrojem LakeShore 421 po RS232, ale marně. Borland delphi umí totiž komunikovat na standardních 8 bitech a se standardně nastavenou paritou. Přístroj LakeShore 421 (z nezjištěných důvodů) komunikuje po 7 bitech a po jiné paritě. Komunikovat po jiných bitech a s jinou paritou mi umožnil program Agilen VEE pro verze 7.

V tomto programu jsem udělal schéma pro komunikaci: pokus.vee



Obr. 16. program pokus.vee v Agilen VEE pro 7

Program funguje následujícím způsobem:

Program stále cyklicky zkouší, je-li v adresáři c:\data přítomen soubor sys.txt. Jestli ano, pošle dotaz k přístroji LakeShore 421 na hodnotu, tuto hodnotu zapíše do c:\data\data1.txt a smaže soubor sys.txt.

Celkově to funguje tak, že:

Skenovací program najde na potřebnou souřadnici. Až je připraven, vytvoří soubor c:\data\sys.txt a čeká, až je soubor smazán. Smazání souboru je pro něj impuls, že hodnota magnetického pole v daném bodě byla zapsána a že může pokračovat v činnosti.

Programování je naprosto stejné, až na to, že hodnoty nejsou hned načteny. Po oskenování povrchu musíme vypnout program pokus.vee a otevřít soubor c:\data\data1.txt v libovolném textovém editoru a dát zaměnit tečku za desetinnou čárku. Je to kvůli znakové konvenci. Program Vee podporuje desetinnou tečku, zatímco Borland Delphi podporuje desetinnou čárku. Potom tento přepsaný dokument uložíme a dáme ve skenovacím programu vyhodnot'.

Proběhne jakoby skenování, mnohonásobně zrychlené, ale již bez posuvů, až na to, že program nenačítá hodnotu ze vstupu, ale ze souboru data1.txt.

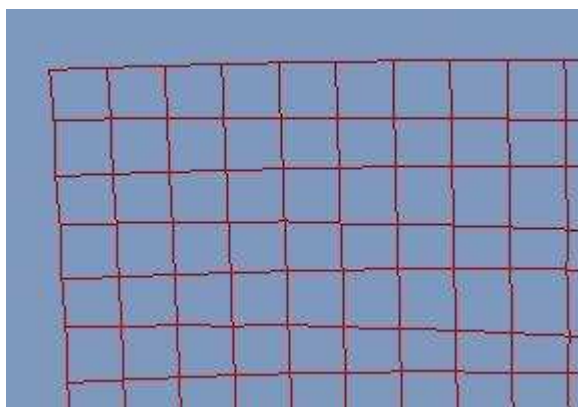
Nakonec dáme uložit data.

6 ZOBRAZOVACÍ PROGRAM (PROGRAMOVÁNÍ)

6.1 Úvod do programování v OpenGL

Cílem této kapitoly není vykopírovat celý zdrojový soubor a popsat striktně každý příkaz. Je to v podstatě nemožné, protože jenom hlavní zdrojový soubor main.pas má 1710 řádků a dalších 15 vedlejších souborů má každý přibližně 200 řádků.

Proto je každý příkaz jen popsán základními příkazy pro pochopení. Má-li někdo zájem, u této práce bude přiloženo CD, na kterém budou všechny tyto zdrojové soubory k nahlédnutí.



Obr. 17 vertex

Pokud programujeme plochu v OpenGL, tak ji definujeme pomocí tzv. vertexů (viz obr. 17). Jsou to nejmenší inkrementy plochy. Nejlepším vertexem je trojúhelník, čtverec, lichoběžník. V tomto programu byl použit čtverec. Základem je jádro mezi `glBegin` a `glEnd`. Obsahující barvu bodu a souřadnice, barvu bodu a souřadnice..... Tímto jádrem dochází k zápisu bodů do zobrazovací matice samotného jádra OpenGL. Vybarvení plochy mezi body propočítává již OpenGL. Barva má hodnotu 0-1 (ne jak je zvykem 0-255). Souřadnice mají libovolnou reálnou hodnotu.

Příkaz tvoří:

`glBegin`(jak vykreslovat)

`glColor3f(R,G,B);` barva 1. bodu

`glVertex3f(x,y,z);` souřadnice 1.bodu

`glColor3f(R,G,B);` barva 2. bodu

GLfloat(x,y,z); souřadnice 2.bodu

GLfloat(R,G,B); barva 3. bodu

GLfloat(x,y,z); souřadnice 3.bodu

GLfloat(R,G,B); barva 4. bodu

GLfloat(x,y,z); souřadnice 4.bodu

glEnd; ukončení zadávání bodů

Přechod barev mezi body již vyřeší *OpenGL*.

Jak vykreslovat: může mít parametry:

GL_POINTS

Všechny vertexy budou vykresleny jako samostatné body v prostoru.

GL_LINES

Každá dvojice po sobě jdoucích vertexů bude spojena čarou. př: Ze čtyř bodů (A, B, C, D) budou vytvořeny dvě čáry (AB) a (CD).

GL_LINE_STRIP

Nakreslí spojitou, lomenou čáru bod po bodu. Začne u prvního vertexu a skončí posledním. př: Ze čtyř bodů (A, B, C, D) budou vytvořeny tři čáry (AB), (BC) a (CD).

GL_LINE_LOOP

Nakreslí spojitou, uzavřenou lomenou čáru. Začne u prvního vertexu a taky jím skončí. př: Ze čtyř bodů (A, B, C, D) budou vytvořeny čtyři čáry (AB), (BC), (CD) a (DA).

GL_TRIANGLES

každá trojice po sobě jdoucích bodů bude spojena do plochy tvořící trojúhelník. př: Z šesti bodů (A, B..F) budou vytvořeny dva trojúhelníky (ABC) a (DEF).

GL_TRIANGLE_STRIP

každé tři po sobě jdoucí body budou spojeny a vytvoří plochu trojúhelníku. př: Z pěti bodů (A, B..E) budou vytvořeny tři trojúhelníky (ABC), (BCD) a (CDE).

GL_TRIANGLE_FAN

každé dva po sobě jdoucí body budou spolu s prvním bodem zadaným po volání procedury *glBegin* spojeny do trojúhelníku. př: Z pěti bodů (A, B..E) budou vytvořeny tři trojúhelníky (ABC), (ACD) a (ADE).

GL_QUADS

každá čtveřice po sobě jdoucích bodů bude spojena do plochy tvořící čtyřúhelník. Př:
Z osmi bodů (A, B..H) budou vytvořeny dva čtyřúhelníky (ABCD) a (EFGH).

GL_POLYGON

všechny body budou spojeny do plochy mnohoúhelníka

V tomto programu bylo použito budeme používat : **GL_LINE_STRIP** a **GL_POLYGON**

6.2 Programování zobrazovacího programu

Po zapnutí programu proběhnou inicializační příkazy OpenGL, které jsou v proceduře po spuštění programu a které nebudou podrobně popsány, protože nejsou momentálně důležité. Jde o nastavení barev pozadí a hlavně o vykreslení nejbližších a nejvzdálenějších bodů matice. Toto nastavení se používá hlavně při programování rozlehlých scén (krajín apod.), pro vytvoření reálného dojmu.

Protože je načítání z textového editu (richeditu) velice zdoluhavé, hlavně když potřebujeme překreslit scénu při otočení nebo po jiné práci s objektem, kdy je potřeba všechny body znovu načíst, vytvoříme si vlastní matici bodů, do které po otevření dat objektu, tyto data načteme.

Var:

```
z:array[0..5000,0..5000]of GLfloat;
```

Tento příkaz znamená: vytvoř matici čísel o velikosti 5000x5000 míst k zápisu reálné hodnoty.

Potom volání daného bodu bude:

```
Hodnota z:=z[x,y];
```

Dále musí být zajištěno, aby když je velikost povrchu například 100x100 bodů, tak aby body, přesahující hranici 100 nebyly vykreslovány.

To je uděláno tak, že před použitím matice program vynuluje, nahrajem hodnoty (musí být všechny kladné) tam, kde patří a pokud je hodnota menší než 0, tak ji nevykreslíme.

Pokud máme v matici menší hodnotu, než je 0, pak k celé matici přičteme uloženou minimální absolutní hodnotu, ale při zobrazování dat tuto hodnotu odečteme. Bodová matice je tak posunuta pro počítač nad nulu, kdežto pozorovatel sedící u počítače si ničeho nevšimne a bude mít k dispozici plusové i minusové hodnoty.

Po každé operaci vyvoláme proceduru

```
Glform.invalidate;
```

Když je provedena jakákoliv změna, tato změna je provedena ve zobrazovací matici OpenGL. Ale objekt byl dříve již vykreslen bez této změny. Tato procedura dá delphi příkaz, že se něco změnilo. Doslovný překlad této procedury je: vykreslovací plocha má

chybná data. To spustí proceduru překreslení plochy z matice OpenGL a to vede ke změně povrchu na obrazovce.

Procedura překresli:

Po vyvolání procedury překresli se znovu načtou dvě proměnné:

Sit – jestli je momentálně zatrhnut síťový (2) nebo vyplněný (1) model

Bar - (1-8) která je momentálně vybraná barva.

if sit=1 then if bar=1 then kresli11; síť ne, barva 1 spust' podprogram kresli11;

.....

if sit=2 then if bar=5 then kresli25; síť ano, barva 5 spust' podprogram kresli25;

.....

Podprogram kresli 11...:

glTranslatef(PolohaX, PolohaY, PolohaZ); Určí polohu objektu

glRotatef(uhelx,-1,0,0); Určí rotaci v ose x o úhel x

glRotatef(uhely,0,1,0); Určí rotaci v ose y o úhel y

glRotatef(uhelz,0,0,1); Určí rotaci v ose z o úhel z

glEnable(GL_COLOR_MATERIAL); Zapne používání barevného materiálu

kreslisit; Spustí podprogram pro vykreslení osového kříže

for y:=y00 to ymaxr do begin cykluje od y0 do y maximální hodnoty

for x:=x00 to xmaxr do begin cykluje od x0 do x maximální hodnoty

pom:='1';

if z[x,y]>-0 then za:=z[x,y]

else pom:='0';

if z[x+1,y]>-0 then zb:=z[x+1,y]

else pom:='0';

if z[x+1,y+1]>-0 then zc:=z[x+1,y+1]

else pom:='0';

if z[x,y+1]>-0 then zd:=z[x,y+1]

```
else pom:='0';
```

tyto čtyři cykly zajistí, že všechny 4 body vertexu jsou různé od nuly. Jestli tomu tak není, uloží do proměnné pom hodnotu 0, to znamená, že vertex již nebude vykreslen.

```
if za<z00 then pom:='0';
if za>zmaxr then pom:='0';
if zb<z00 then pom:='0';
if zb>zmaxr then pom:='0';
if zc<z00 then pom:='0';
if zc>zmaxr then pom:='0';
if zd<z00 then pom:='0';
if zd>zmaxr then pom:='0';
```

Těchto pár podmínek slouží pro řez v ose Z. Je-li bod menší než Z minimální nebo je-li bod větší než Z maximální, uloží do pom 0.

```
if pom='1' then begin      podmínky jsou splněny, vykresluj
glbegin(gl_polygon);      pro plný objekt(gl_polygon), pro sít'ový(gl_line_strip)
```

Program propočítá barevný inkrement pro zvolenou barvu, v tomto případě se jedná o odstín zelené.

```
Zi=1/zmax;
```

Potom

Bod A

```
glcolor3f(R,zi*momentální hodnota z,B);
```

```
Glvertex3f(X,Y,z[X,Y]);
```

Bod B

```
glcolor3f(R,zi*momentální hodnota z,B);
```

```
Glvertex3f(X+1,Y,z[X+1,Y]);
```

Bod C

```
glcolor3f(R,zi*momentální hodnota z,B);
```

```
Glvertex3f(X+1,Y+1,z[X+1,Y+1]);
```

Bod D

```
glcolor3f(R,zi*momentální hodnota z,B);
```

```
Glvertex3f(X,Y+1,z[X,Y+1]);
```

Tím dojde k zapsání 1. čtverec.

```
glEnd;
```

A tak dlouho program cykluje, až hodnota x a y nabydou hodnot xmax, ymax.

Dojde k vykreslení plochy.

Další barvy:

Předcházející případ byl pro odstín zelené barvy. Ostatní odstíny jsou podobné:

Odstín červené:

```
Zi:=1/zmax;
```

```
GlColor3f(zi* momentální hodnota z,G,B);
```

Odstín modré:

```
Zi:=1/zmax;
```

```
GlColor3f(R,G,zi* momentální hodnota z);
```

Odstín červené, přecházející do modré:

```
Zi:=1/zmax;
```

```
GlColor3f(1-(zi* momentální hodnota z),G, zi* momentální hodnota z);
```

S přibývajícím hodnotou z modré přibývá ale červené ubývá.

Odstín modré, přecházející do červené:

$Z_i = 1/z_{max};$

$GlColor3f(z_i * \text{momentální hodnota } „G”, 1 - (z_i * \text{momentální hodnota } z));$

S přibývajícím hodnotou z červené přibývá ale modré ubývá.

Odstín červené, přecházející do zelené

$Z_i = 1/z_{max};$

$GlColor3f(1 - (z_i * \text{momentální hodnota } z), z_i * \text{momentální hodnota } z, B);$

S přibývajícím hodnotou z zelené přibývá ale červené ubývá.

Odstín zelené, přecházející do zelené

$Z_i = 1/z_{max};$

$GlColor3f(z_i * \text{momentální hodnota } z, 1 - (z_i * \text{momentální hodnota } z, B));$

S přibývajícím hodnotou z červené přibývá ale zelené ubývá.

Duhový odstín:

Dokonale se vybarví jen na obrovském rozlišení. Je to mnou nadefinovaná matice barev.

*poznámka k rotaci:

Na začátku podprogramu kreslí je procedura pro rotaci objektu.

$glTranslatef(PolohaX, PolohaY, PolohaZ);$ Určí polohu objektu (kam se má kamera „dívat“)

$glRotatef(uhelx, -1, 0, 0);$ Určí rotaci v ose x o úhel x

$glRotatef(uhely, 0, 1, 0);$ Určí rotaci v ose y o úhel y

$glRotatef(uhelz, 0, 0, 1);$ Určí rotaci v ose z o úhel z

$glEnable(GL_COLOR_MATERIAL);$ Zapne používání barevného materiálu

Rotace kolem os probíhá vždy kolem bodu [0,0,0]. Takže pokud je potřeba rotovat kolem os středem objektu, jak je to v našem případě, pak musíme od os při zobrazování odečíst polovinu objektu v každé ose. Tím dostaneme bod [0,0,0] do středu objektu.

Rotace může také probíhat kolem více os například `glRotatef(uhel,1,1,1)` bude rotovat kolem všech tří os současně.

Proměnné Poloha X,Y se mění při pohybu myši se zamáčknutým pravým tlačítkem.

Proměnné Poloha Z se mění „lupou“.

Proměnné uhel X,Y,Z se mění při pohybu myši se zamáčknutým levým tlačítkem.

V OpenGL nelze rotovat o více než 270°:

Je to způsobeno propočtem matice v OpenGL. Náš povrch se skládá z velkého počtu bodů. Každý z nich je sám spočítán a zrotován. Při velké rotaci dojde k chybám a stávalo se v programu, že když nebylo v programu omezení, došlo náhle k absolutnímu převrácení matice a plocha se zobrazila „vzhůru nohama“. Z těchto důvodů jsem kontaktoval nejlepší programátory pro OpenGL na nehe tutorial a dozvěděl jsem se, že musím dát omezení, že je to chyba samotného výpočtu zobrazovací matice. Pokud chceme rotovat o více než 270°, musíme při přechodu přes 270° přepočítat matici a to tak, že ji v postatě vykreslíme od zadu (zrcadlově-nezobrazujeme $x_0, x_1, x_2, \dots, x_n$ ale $x_n, \dots, x_2, x_1, x_0$) Tím se nám obrazně otočí předmět o 180° a potom můžeme rotovat o dalších 180°.

Řez objektem:

Z předcházející části o vykreslování objektu je zřejmé, že vykreslování objektu v osách se provádí:

V ose x: od hodnoty proměnné x_{00} do hodnoty proměnné x_{max}

V ose y: od hodnoty proměnné y_{00} do hodnoty proměnné y_{max}

V ose z: od hodnoty proměnné z_{00} do hodnoty proměnné z_{max}

Řez objektem je tedy proveden velice jednoduše, a to tak, že je změněna hodnota

X_{00}, y_{00}, z_{00} : (řez od začátku ke konci)

$X_{max}, y_{max}, z_{max}$: (řez od konce k začátku)

A potom je zavolána procedura `glForm.invalidate`.

Tlačítko původní velikosti objektu načte do těchto 6ti proměnných zpět původní hodnoty a zavolá proceduru `glForm.invalidate`;

Časová rotace:

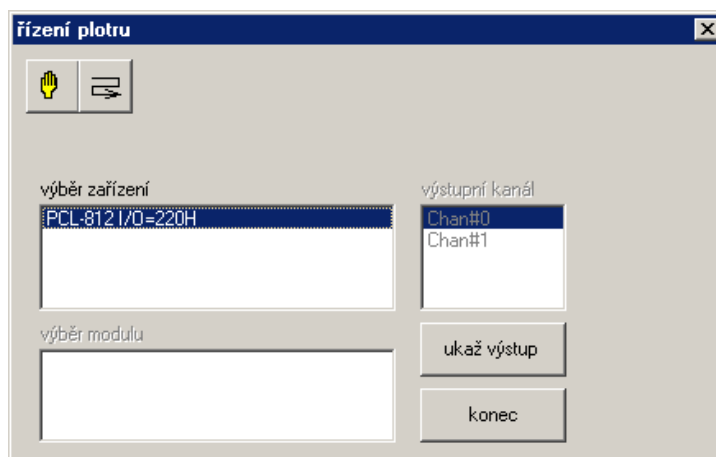
Efektní animace vznikne, vložíme-li do programu timer. Ten nám po zapnutí bude po desetině sekundy přičítat a zase odečítat inkrement úhlu ve zvolené ose.

Nalezení minima a maxima plochy:

Cyklus proběhne hodnoty matice Z a nalezne minima a maxima. Pokud je více stejných minim (maxim) zapíše pouze tu poslední hodnotu.

7 MANUÁL KE SKENOVÁNÍ PŘES ANALGOVÝ VSTUP

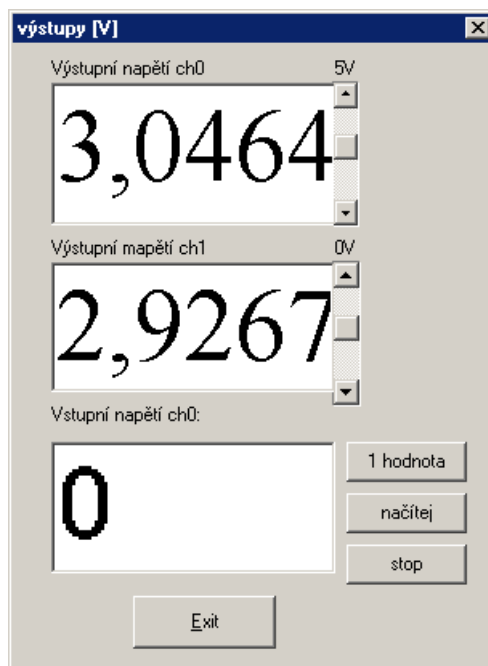
Spusťte program analog_scan.exe. Pokud se vám zobrazí chybové hlášky, je to způsobeno chybnými adresami ve windows. Po odkliknutí se Vám zobrazí následující okno:



Obr. 18 hlavní formulář řízení plotru

Po kliknutí na Vybranou kartu PCL_812 I/O=220H se Vám aktivují horní dvě tlačítka.

Při kliknutí na dolní tlačítko „ukaž výstup“ se Vám ukáže momentální hodnota výstupů ch0 , ch1 a vstupní hodnota na vstupu ch0.



Obr. 19 formulář výstupů a vstupů karty advantech

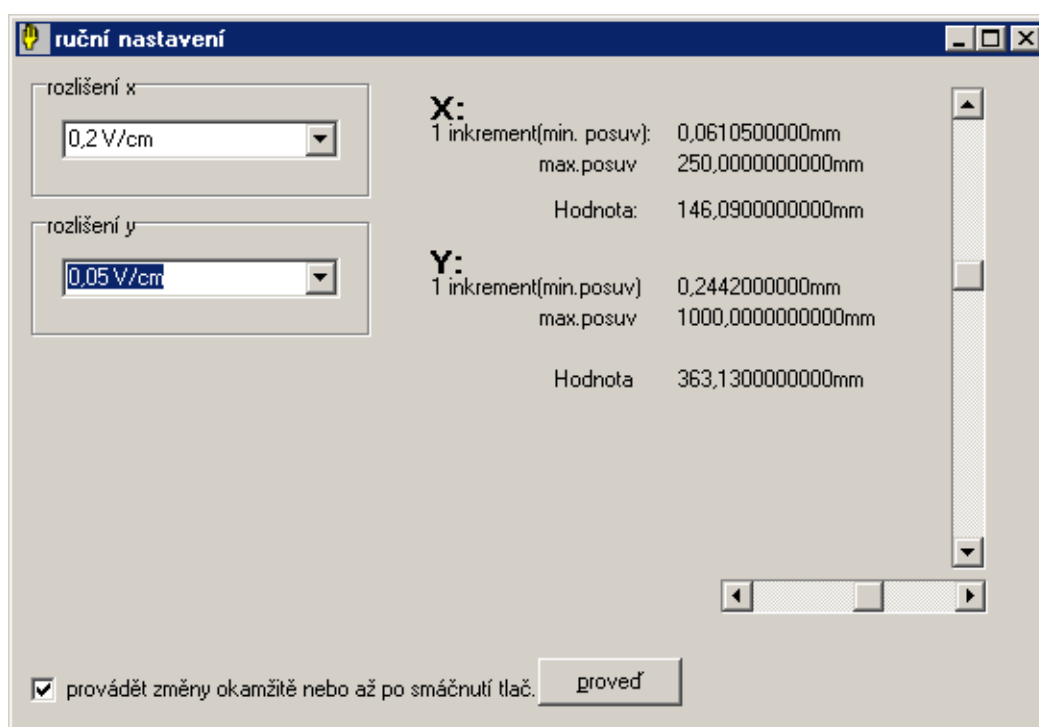
Při kliknutí na „1 hodnota“ se Vám zobrazí vždy jen vstupní hodnota v okamžiku kliknutí.

Při kliknutí na “načítej” se Vám bude obnovovat vstupní hodnota po 0,01s.

Tlačítkem „stop“ zastavíte načítání hodnot.

Posuvníkem na pravém kraji výstupních hodnot měníte tuto výstupní hodnotu.

Kliknutím na hlavním formuláři na ikonku ruky se nám zobrazí menu ručního posuvu



Obr. 20 formulář ručního řízení plotru

Vlevo nahoře vybereme rozlišení, na které máme nastavený plotr. Podle těchto hodnot se nám bude orientačně propočítávat hodnoty.

Horizontálním posuvníkem měníme aktuální hodnotu X.

Vertikálním posuvníkem měníme aktuální hodnotu Y.

Zobrazují se nám hodnoty:

1 inkrement: hodnota minimálního posuvu v dané ose

Max posuv: maximální posuv v dané ose (bez ohledu na rozměry plotru)

Hodnota: Hodnota, o kolik mm byl posunut posuv od nulového bodu.

Pokud je dolní políčko zaškrtnuté, změna v posuvu se projeví okamžitě.

Pokud je políčko nezaškrtnuté, změna se projeví až po kliknutí tlačítka „proved“

Po kliknutí na hlavním formuláři na tlačítko šipky „skenování“ se nám zobrazí první formulář skenování

Skenování 1/3

- Zadej rozměry zkoumané plochy

x: 50 mm
y: 50 mm

- Zadej nastavené rozlišení

x: 0,5 V/cm
y: 0,5 V/cm

1 Inkrement v X je [mm]: 0,0244200000
1 Inkrement v Y je [mm]: 0,0244200000

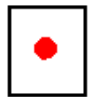
Rozlišení x je: zvoleno dobře
Rozlišení y je: zvoleno dobře
rozměr je v pořádku
rozměr je v pořádku

Ujistěte se že plotr má dostatek místa k pohybu a zmáčkněte tlač. nuluj



Po dojetí plotru najedzte RUČNÍM POSUVEM NA PLOTRU na střed sledované plochy

- Typ pohybu PLOTRU



- Jestliže jsou zadány všechny parametry klikněte na OK pro pokračování

Obr. 21 formulář skenování 1/3

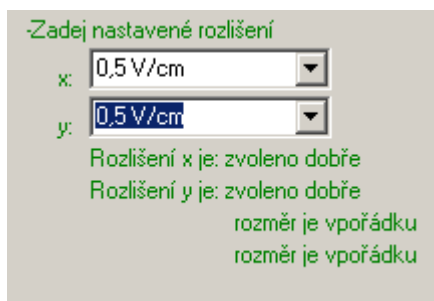
Do prvních dvou políček zadejte rozměry zkoumané plochy v mm

Do druhých dvou políček zadejte nastavené rozlišení plotru v obou osách.

Mohou nastat tři stavy:

1.

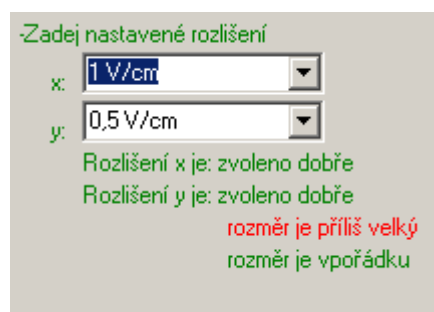
rozlišení je zvoleno vpořádku



Obr. 22. ukázka vyhodnocení

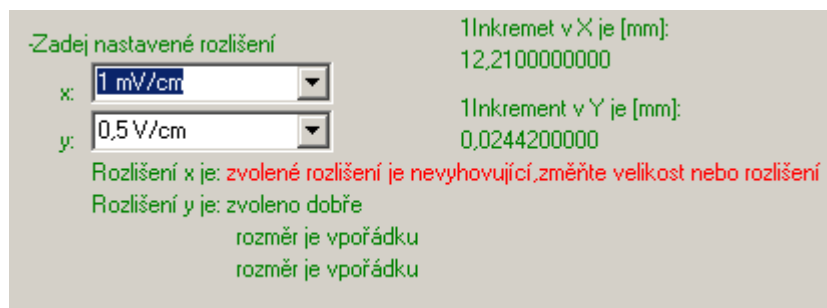
2.

Rozlišení je příliš malé, rozměr skenované plochy je příliš velký, skenování neobsáhne celou plochu. Musíte změnit rozlišení nebo zmenšit rozměr.



Obr. 23 ukázka vyhodnocení

3. Rozlišení je příliš malé, skenovaná oblast je menší než 1 inkrement pohybu. Musíte zvětšit skenovanou oblast nebo zmenšit rozlišení.



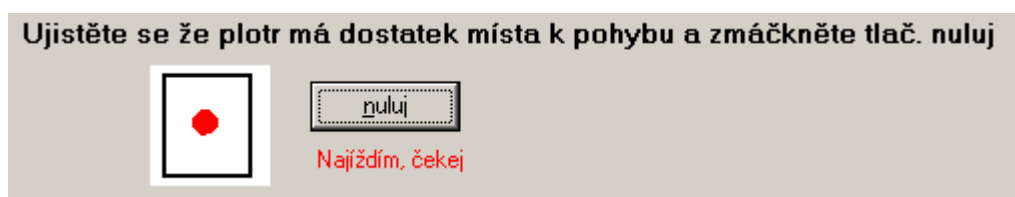
Obr. 24. ukázka vyhodnocení rozlišení

Se změnou rozlišení se Vám mění i inkrement v osách. Dle vlastní volby zvolte potřebnou jemnost rastru, doporučuji ten nejjemnější.

Dále postupujte dle nabídky programu.

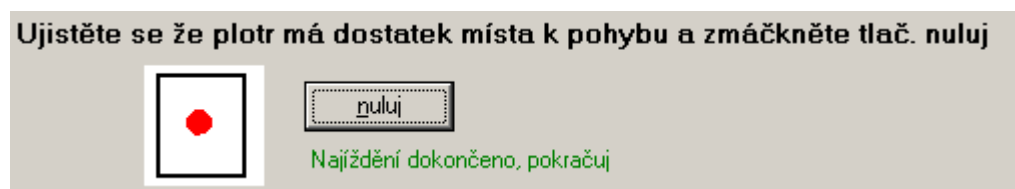
Ujistěte se, že plotr má dostatek místa a klikněte na tlačítko nuluj.

Plotr najede přesně na SVŮJ střed souřadného systému což je 2,5V a 2,5V.



Obr. 25 hláška najíždění

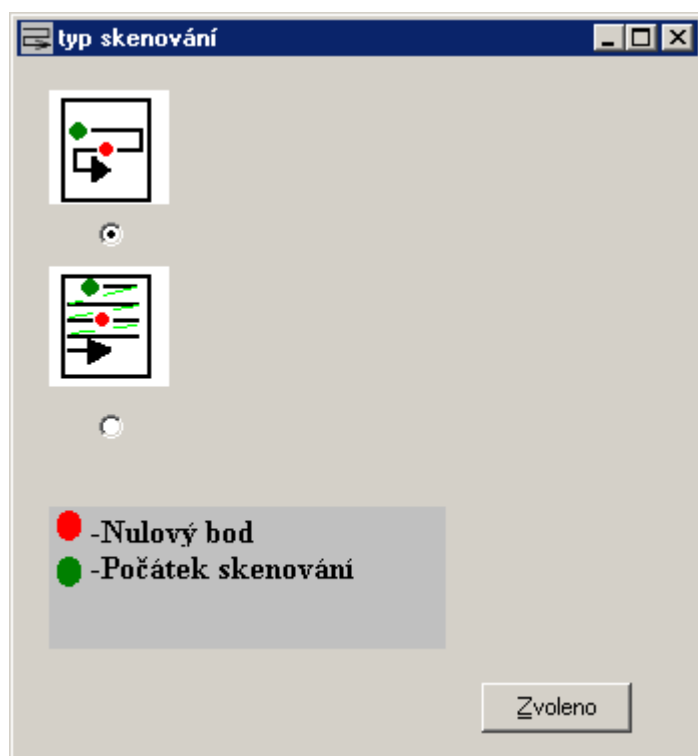
Dokud svítí červeně „najíždím, čekej“ počítač najíždí na střed souřadného systému, plotr by se měl pomalu pohybovat.



Obr. 26 hláška konec najíždění

Po rozsvícení „najíždění dokončeno, pokračuj“ najed'te dle příkazů programu **RUČNÍM POSUVEM PLOTU** na střed měřené oblasti povrchu.

Následně klikněte na tlačítko “<<<zvolit“ a zvolte typ skenování



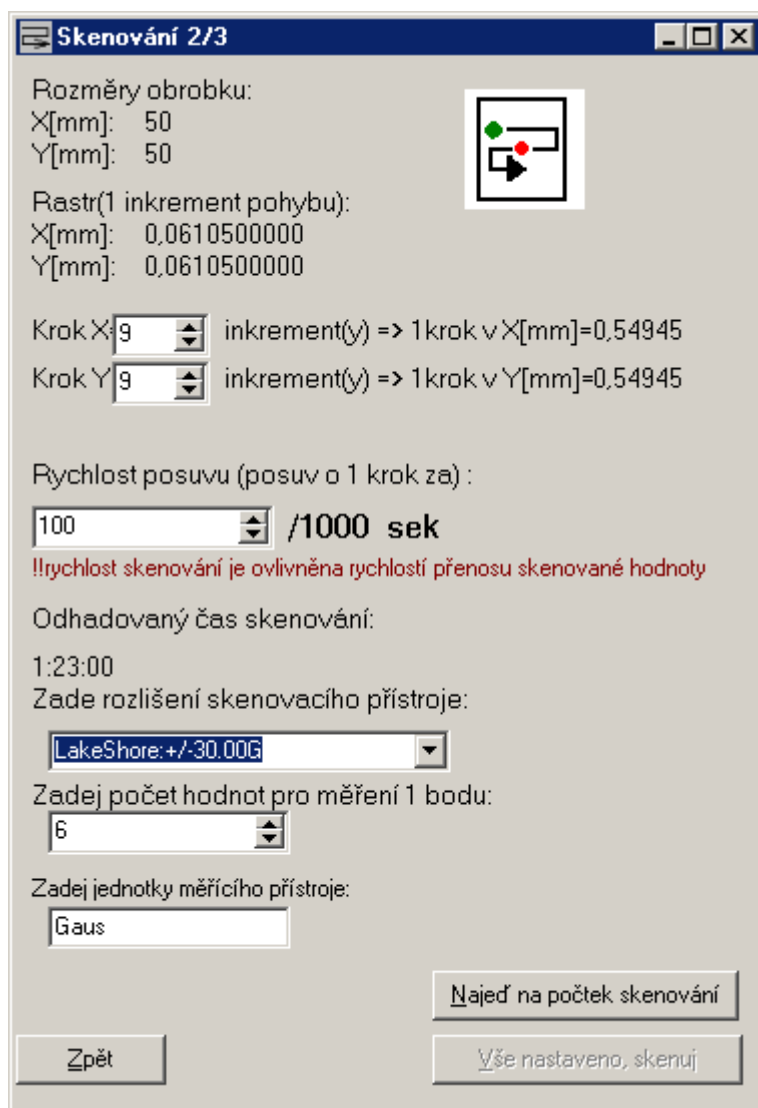
Obr. 27 typ skenování

První typ je rychlejší než druhý. Druhý je o polovinu rychlejší než první, ale průměruje naměřené hodnoty. Takže jsou odfiltrovány případné chyby.

Po zvolení klikněte na „zvoleno“. Vráťte se tím na první formulář skenování.

Potom klikněte na tlačítko „OK“, které Vás přesune na druhý formulář skenování.

Z tohoto formuláře se můžete vrátit zpět pomocí tlačítka „zpět“.



Skenování 2/3

Rozměry obrobku:
X[mm]: 50
Y[mm]: 50

Rastr(1 inkrement pohybu):
X[mm]: 0,0610500000
Y[mm]: 0,0610500000

Krok X: 9 inkrement(y) => 1 krok v X[mm]=0,54945
Krok Y: 9 inkrement(y) => 1 krok v Y[mm]=0,54945

Rychlost posuvu (posuv o 1 krok za):
100 /1000 sek
!! rychlost skenování je ovlivněna rychlostí přenosu skenované hodnoty

Odhadovaný čas skenování:
1:23:00

Zade rozlišení skenovacího přístroje:
LakeShore: +/-30.00G

Zadej počet hodnot pro měření 1 bodu:
6

Zadej jednotky měřicího přístroje:
Gaus

Najed' na počtek skenování

Zpět Vše nastaveno, skenuj

Obr. 28 formulář skenování 2/3

Na tomto formuláři se vám zobrazí:

Zadané rozměry obrobku

Typ skenování

Rastr, je to nejmenší možný kro, který je schopen počítač, s vaším nastavením rozlišení, udělat (bez ohledu na přesnost plotru!).

Krok X(Y): zde nastavíte, jaký počet rastrů budete chtít k udělení jednoho kroku. V našem případě jsme zvolili $\text{krok} = 9 \cdot \text{rastr}$ takže skenované body budou od sebe vzdáleny 0,54945mm.

Potom nastavíme rychlost skenování: jak rychle se skener dostane z jednoho bodu na druhý (záleží na přenosové rychlosti oskenované hodnoty). Při jakékoliv změně se nám zobrazuje odhadovaný čas skenování, podle kterého se také můžeme řídit.

Dále nastavíme rozlišení přístroje (pro propočet naskenované hodnoty)

Potom nastavíme počet hodnot pro 1 bod. Znamená to, že například při našem nastavení dojde k načtení 6 hodnot po sobě po daném intervalu. Tím se rychlost skenování sníží 6x. Hodnoty se zprůměrují. Aby se odfiltroval případný šum.

Zadejte jednotky přístroje (slouží pro zobrazovací program)

Je-li vše vyplněno, klikněte na „najed“ na počátek skenování“

Plotr by měl najet na levý horní okraj skenované oblasti.

Jestli tomu tak není, najed'te pomocí ručního posuvu plotru na tento bod.

Odklikněte hlášku, že je vše nastaveno a klikněte na tlačítko „vše nastaveno, skenuj“

Tím se zahájí skenování a zobrazí se 3.formulář.

skenování 3/3

Odhadovaná doba skenování:
může být ovlivněna rychlostí načítání dat
0:3:00
Čas zahájení skenování
10:00:36
Čas nyní
10:01:15

Oskenováno:
X: 28/40 skenováno/z(skenových bodů)
Y: 1/40 skenováno/z(skenových bodů)

Výstup
ch0: 2,5104
ch1: 2,4774

Ulož data

2%

Storno

Obr. 29. formulář skenování 3/3

V průběhu skenování se nám zobrazuje odhadovaná doba skenování, čas zahájení skenování a momentální čas.

Dále se nám zobrazuje: oskenováno kolik bodů z kolikati (rastru). Když máme například krok 9, pak nám budou čísla naskakovat po 9vítí.

Data: v prvním editu se nám zobrazí data: krok v x, krok v y, velikost v x a velikost v y.

V druhém editu se nám zobrazují oskenované hodnoty.

Skenování můžeme stornovat tlačítkem storno.

Po dokončení skenování uložíme klikneme na „uložit data“

Edit hodnot se nám uloží s příponou *.dt1 a edit rozměrů se nám uloží s příponou *.dt2.

Doporučuji jim dát stejné jméno.

Např. pokus.dt1 a pokus.dt2.

8 MANUÁL KE SKENOVÁNÍ PŘES RS232

Skenování přes RS232 je stejné jako přes Analogový vstup.

Je ale potřeba mít nainstalovaný program Agilent VEE Pro. Po vyplnění druhého formuláře je potřeba spustit c:\data\pokus.vee a až potom zpustit skenování.

Následně po dokončení skenování VYPNĚTE program pokus.vee otevřete soubor c:\data\data.txt a dejte zaměnit tečku za čárku (kvůli desetinné konvekci. Agilent používá des. Tečku, kdežto delphi vyžaduje čárku).

Potom dejte uložit tento soubor a následně klikněte na „vyhodnotit“. Vyhodnocená data uložte stejně jako u skenování přes Analogový vstup.

9 MANUÁL K PROGRAMU ZOBRAZENÍ DAT

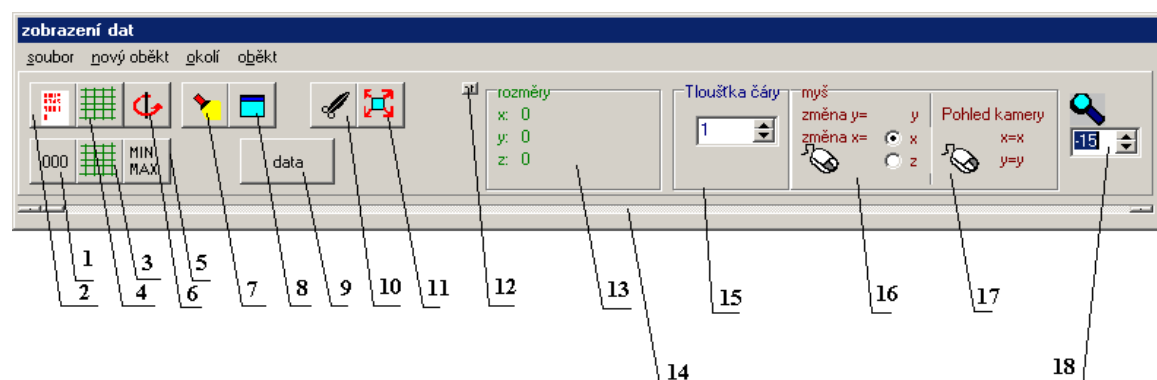
Spustíte program „zobraz“

Ovládání tohoto programu je převážně pomocí myši počítače.

Při pohybu po hlavní ploše se zmáčknutým levým tlačítkem myši budeme rotovat pozorovaným předmětem kolem os x a y. Pokud chceme rotovat kolem osy z, musíme přepnout na hlavním programovém panelu myš na z a rotace kolem osy x bude nahrazena rotací kolem z.

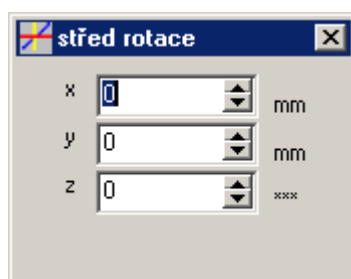
Pokud budeme mít zamáčkнутé pravé tlačítko, budeme pozorovaným předmětem pohybovat po osách y a x. Pohyb po z je lupa.

Prvním krokem je soubor -> otevřít a potom si vyzkoušet, co program umí.



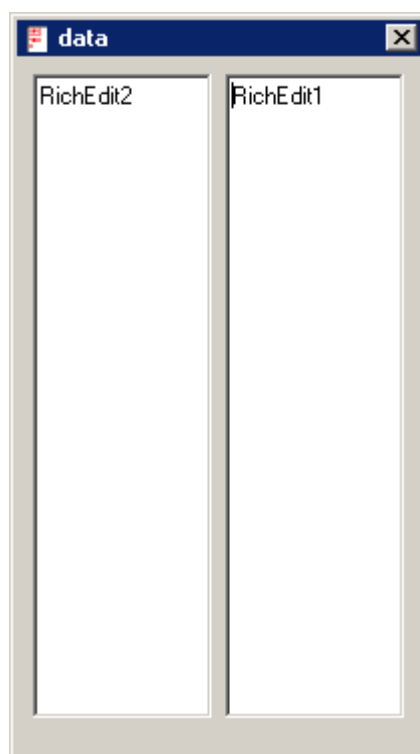
Obr. 30 popis ovládacího panelu zobrazovacího programu

- 1- zobrazí formulář změny středu rotace



Obr. 31 střed rotace

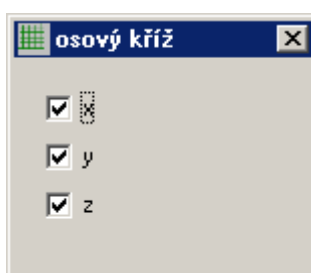
- 2- zobrazí formulář simulačních dat



The image shows a window titled "data" with a standard Windows-style title bar. Inside the window, there are two vertical text editors. The left editor is labeled "RichEdit2" and the right editor is labeled "RichEdit1". Both editors are currently empty.

Obr. 32. formulář dat

- 3- zobrazí formulář síťový/celistvý model



The image shows a window titled "osový kříž" with a standard Windows-style title bar. Inside the window, there are three checkboxes, each with a small icon to its right. All three checkboxes are checked. The icons appear to be a 3D coordinate system (x, y, z) and a 2D coordinate system (x, y).

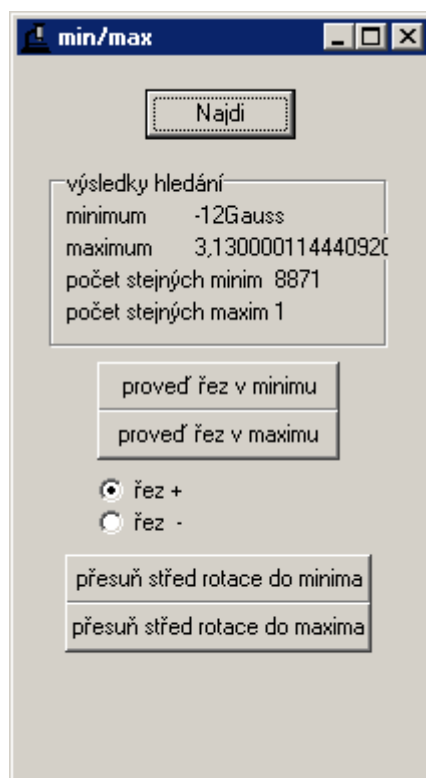
Obr. 33 zapnutí
osového kříže

- 4- zobrazí formulář orientační sítě



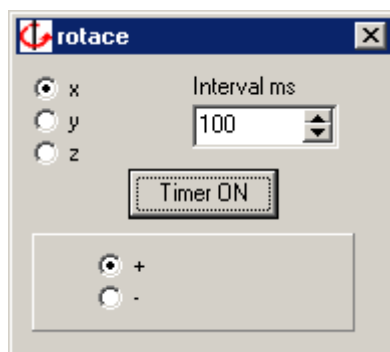
Obr. 34 výběr typu
plochy

- 5- zobrazí formulář vyhledání minima a maxima modelu



Obr. 35 formulář hledání
min. max.

- 6- zobrazí formulář rotace závislé na čase (animace)



Obr. 36 formulář časové
rotace

- 7- zobrazí formulář nastavení světla (nýbrž jsou barvy dost syté, změny se projeví jen nepatrně)

- 8- zobrazí formulář barvy pozadí



Obr. 37.
nastavení barvy
pozadí

- 9- zobrazí formulář data modelu

The 'Data' dialog box contains the following information:

- Velikost:**
 - X: 40mm
 - Y: 40mm
 - Z: 26Gauss
- Inkrement:**
 - X: 0,42735mm
 - Y: 0,42735mm
 - Z: dle přesnosti přístroje
- Jednotky:**
 - X: mm
 - Y: mm
 - Z: Gauss
- Z maximum:** 26Gauss
- Z minimum:** 12Gauss
- typ skenování:** (represented by a square icon with a green dot, a red dot, and a black arrow indicating a scanning path)
- Zavři** (button)

Obr. 38 data o skenovaném objektu

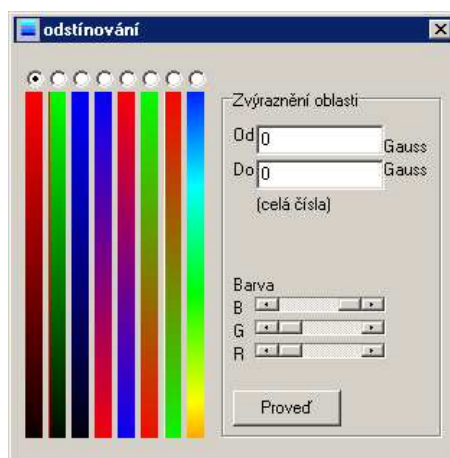
- 10- zobrazí formulář řezu oběktu

The 'řez' dialog box contains the following information:

- od** (from) and **do** (to) columns with input fields and up/down arrows.
- x:** 0 to 94 mm
- y:** 0 to 94 mm
- z:** -12 to 26 inkrementu*
- *1 inkrement viz data o povrchu** (footnote)

Obr. 39 řez objektem

- 11- zobrazí celý model
- 12- zobrazí formulář programátora
- 13- rozměry modelu
- 14- lupa (přiblížení a oddálení modelu)
- 15- tloušťka čáry při síťovém modelu
- 16- při změně polohy myši se zmáčknutým levým tlačítkem bude rotovat model kolem vybraných os
- 17- při změně polohy myši se zmáčknutým pravým tlačítkem budeme přesouvat model v osách x a y
- 18- lupa2 (přiblížení a oddálení modelu)
- 19-  zobrazí vybarvení objektu (boční lišta)



Obr. 40 vystínování objektu

Zvýraznění oblasti se používá při zobrazení nějaké oblasti (od-do) v ose Z, výraznou barvou nadefinovanou uživatelem RGB.

10 **INSTALACE SKENOVACÍHO A ZOBRAZOVACÍHO PROGRAMU**

1.1 Minimální požadavky systému

Procesor: Pentium 300MHz

RAM: min. 64Mb

Grafická karta: libovolná, na zobrazovací program: min.rozlišení 24Mil. Barev, na skenovací program jakékoliv.

1.2 Instalace programu

Zobrazovací program lze spouštět z jakéhokoliv místa na jakémkoliv disku.

Skenovací program

- potřebuje jako podporu nainstalovat program “Agilent VEE pro”
- skopírovat složku DADA na místní disk C:\

II. PRAKTICKÁ ČÁST

11 NASKENOVANÁ DATA A PRÁCE S NIMI V ZOBRAZOVACÍM PROGRAMU

11.1 Data naskenovaná přes analogový vstup

Jako zdroj magnetického pole byla pro začátek zvolena šachová figurka, která má ze spodu kruhový magnet.

Rozměry skenované oblasti byly zvoleny experimentálně a to tak, že se ručním posuvem posouvalo od šachové figurky, dokud se hodnota magnet. Pole neustálila.

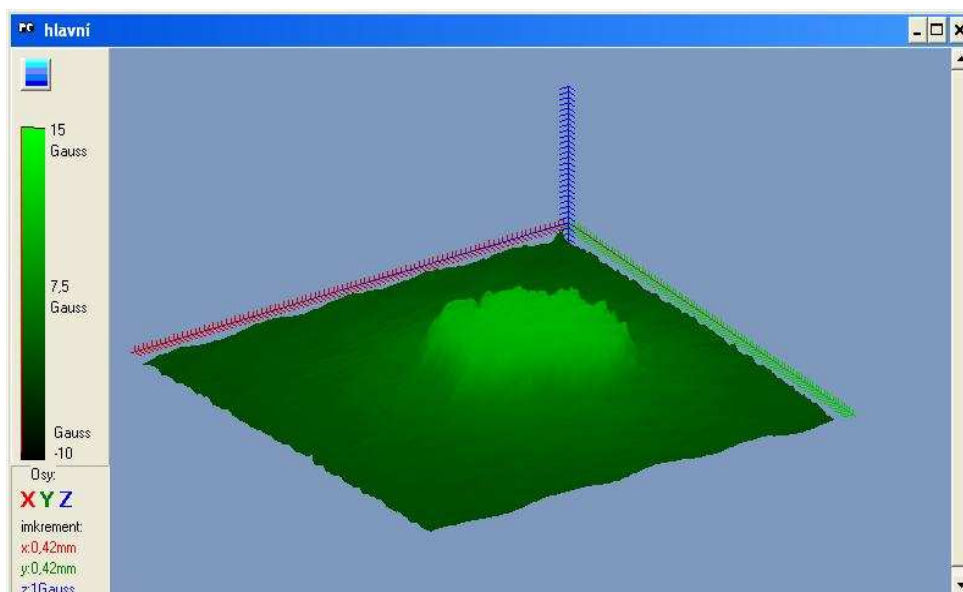
Experimentálně zvolené rozměry skenované oblasti: 40x40mm

Inkrement byl zvolen tak, aby byl dostačující pro podrobné vykreslení oblasti, ale čas skenování nebyl příliš dlouhý (i když na čase v tomto případě nezáleželo, protože magnetické pole je stálé.). Experimentálně byl zvolen inkrement 0,42 mm v obou osách. Čas byl přibližně 50 minut a pro náš rozměr skenované plochy to bylo 9070 bodů pro vykreslení plochy.

Typ skenování jsem vybral oboustranné skenování.



Obr. 41 Skenáž šachové figurky



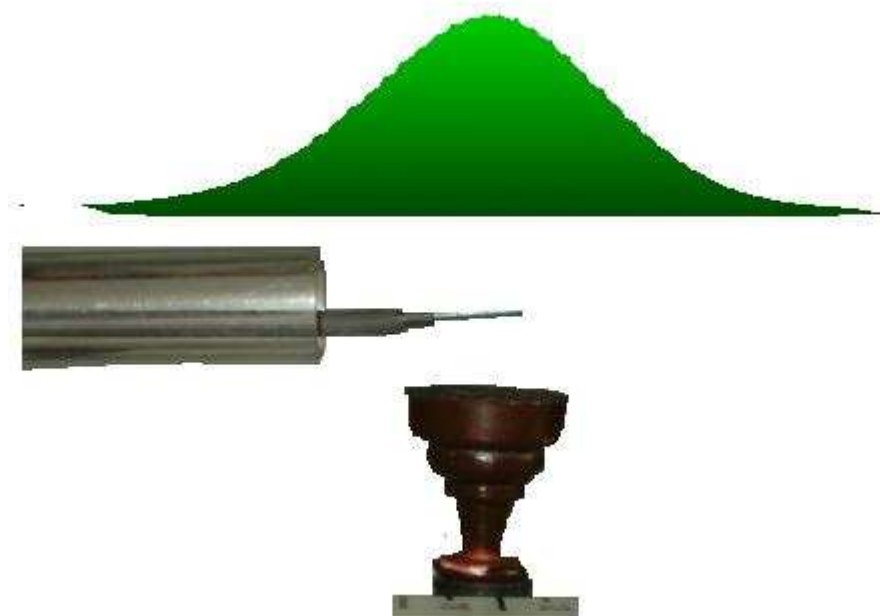
Obr. 42 Zobrazení oskenovaných hodnot pře analog. Vstup

Jde vidět, jak je analogový vstup karty zašuměný. Výstup z LakeShoru byl 0-3V a šum místy dosahoval až 0,5V! Šum jsem částečně odstranil naskenováním a zprůměrováním 6ti hodnot pro každý bod.

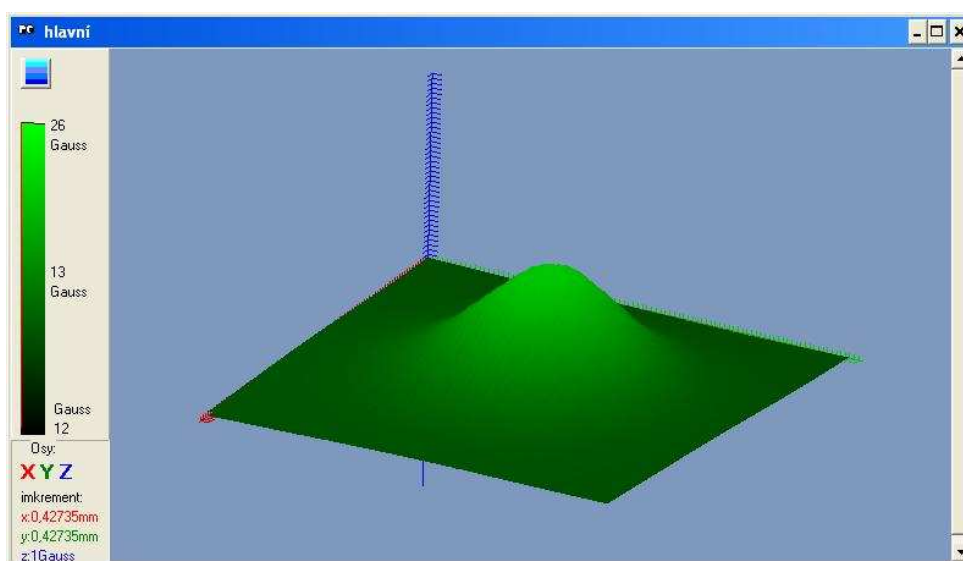
Šum karty byl pravděpodobně způsoben neošetřenými vstupy karty PCL812PG. Protože tu byla možnost načítat data po RS232, již jsme potom, jak ošetřit vstup nepátrali.

11.2 Data naskenovaná přes RS232:

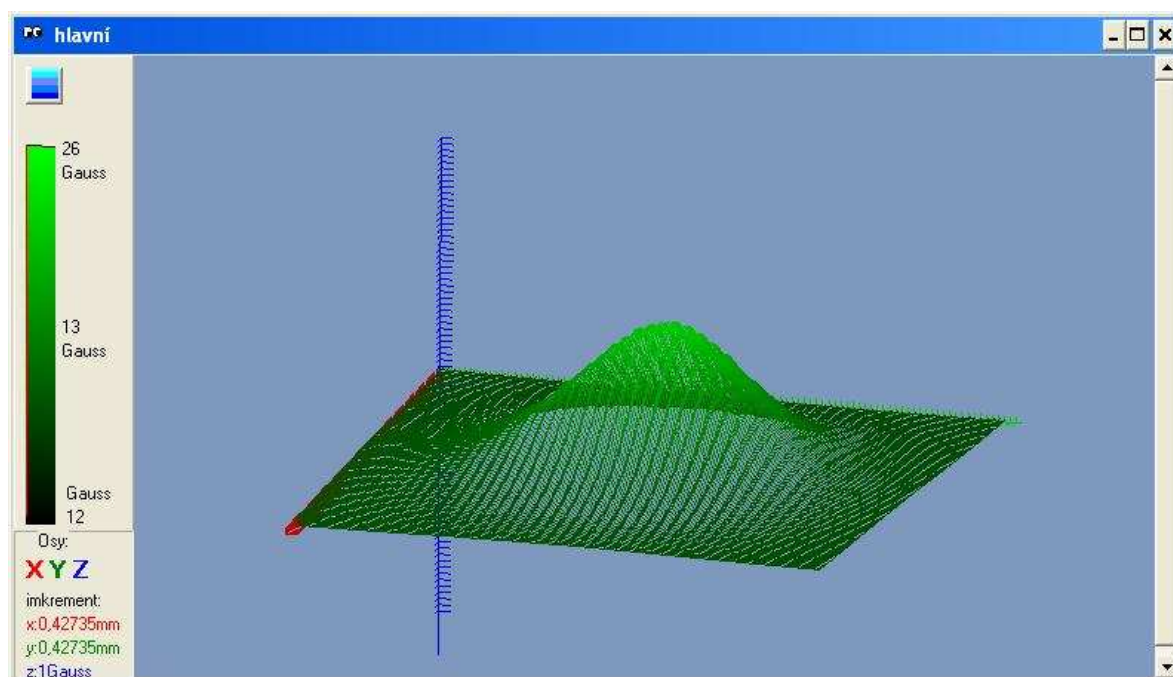
Rozměry skenované oblasti, inkrement a typ skenování jsou stejné jako u předcházejícího skenování, aby bylo možné tyto dvě varianty mezi sebou porovnat.



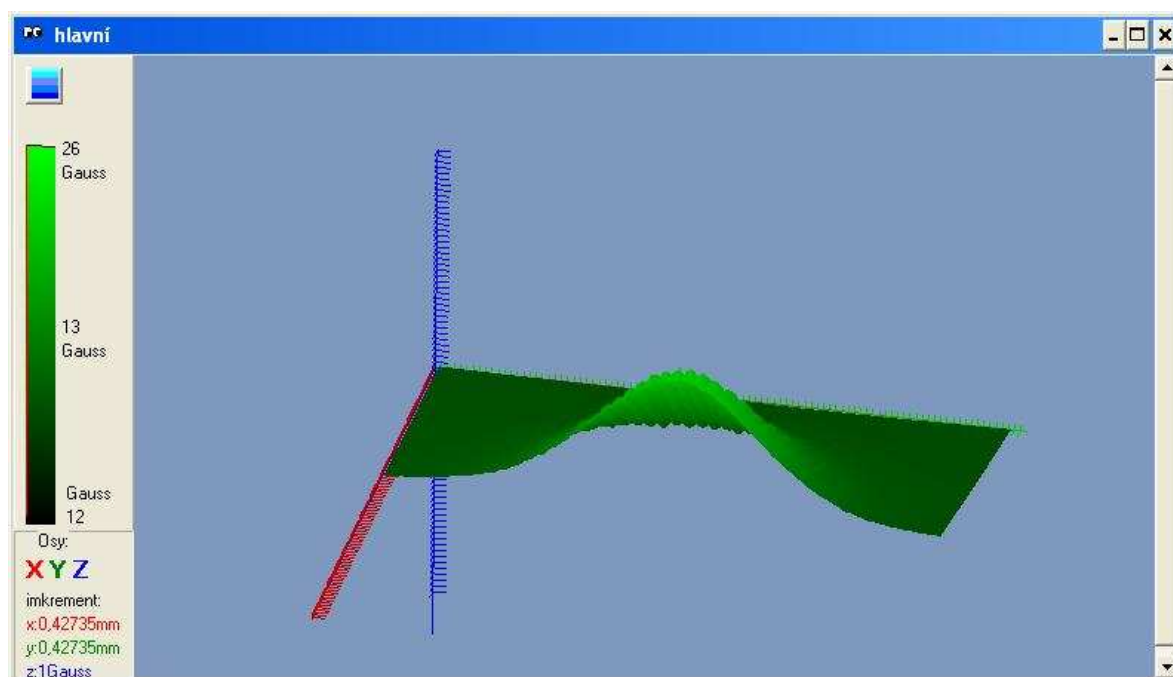
Obr. 43 Skenáž šachové figurky + rozložení magnetického pole.



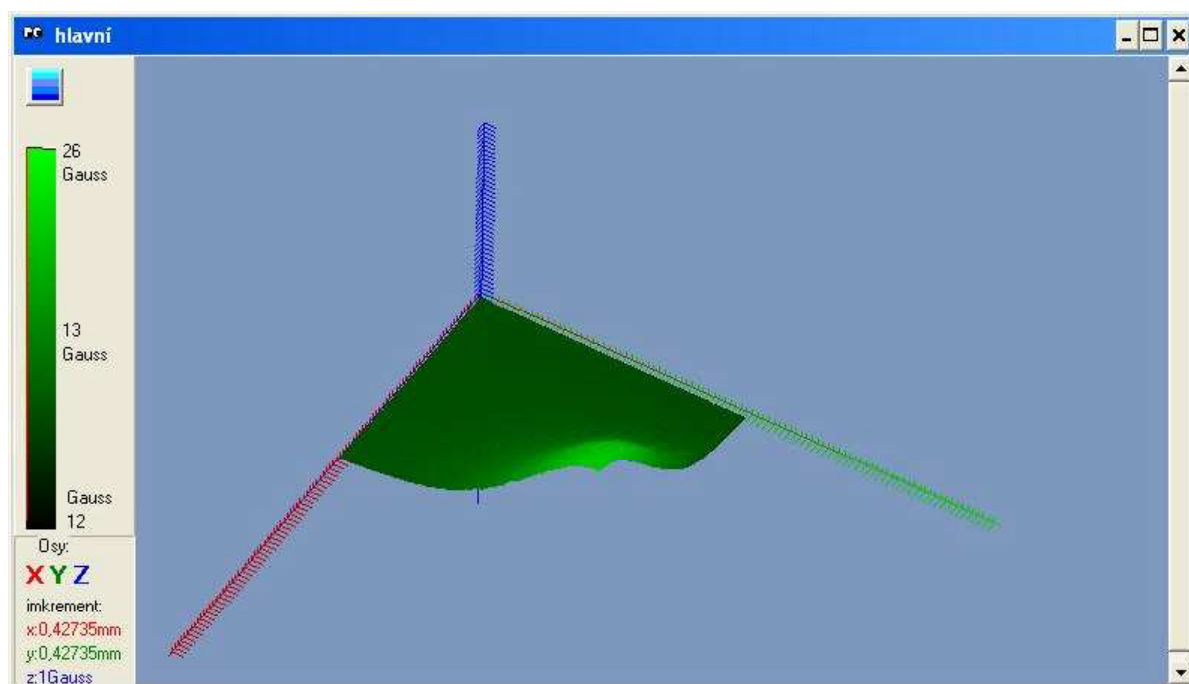
Obr. 44 Skenáž magnetického pole šachové figurky přes RS232



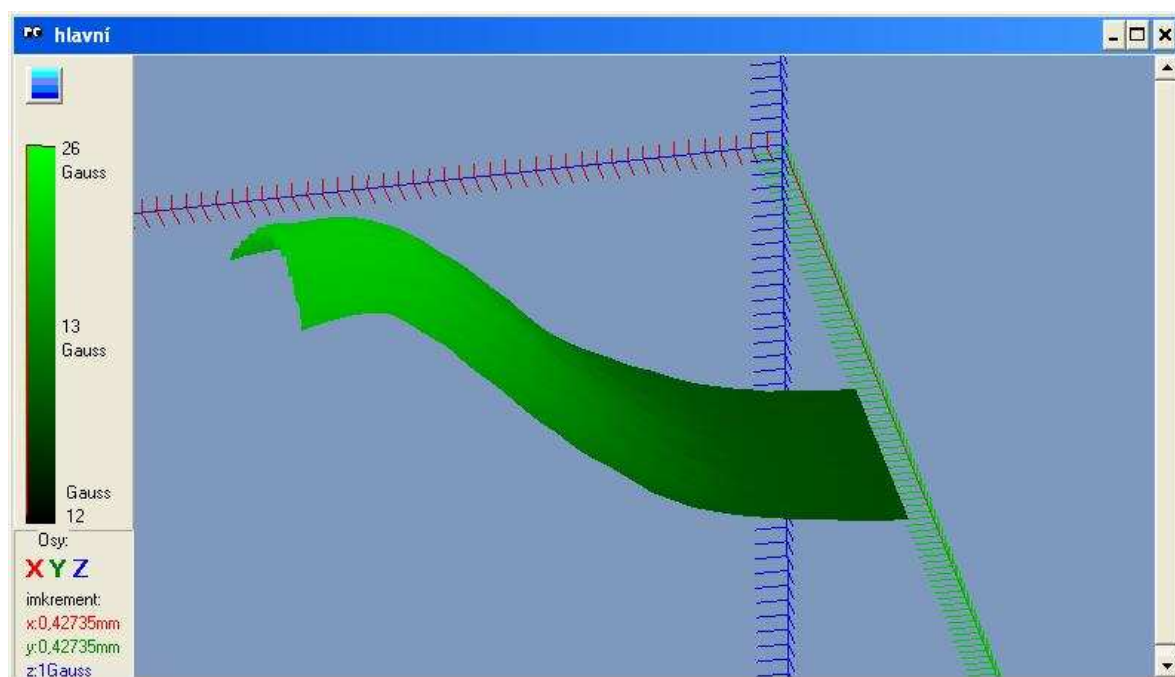
Obr. 45 Síťový model magnetického pole figurky



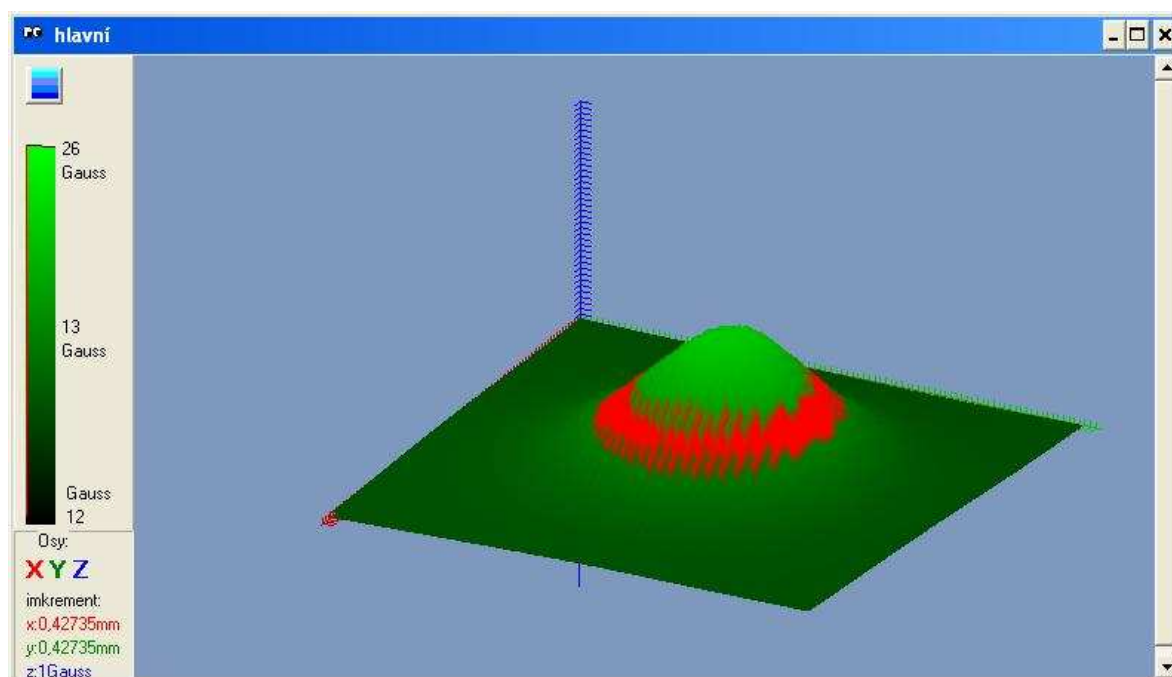
Obr. 46 Řez magnetickým polem šachové figurky v polovině osy X



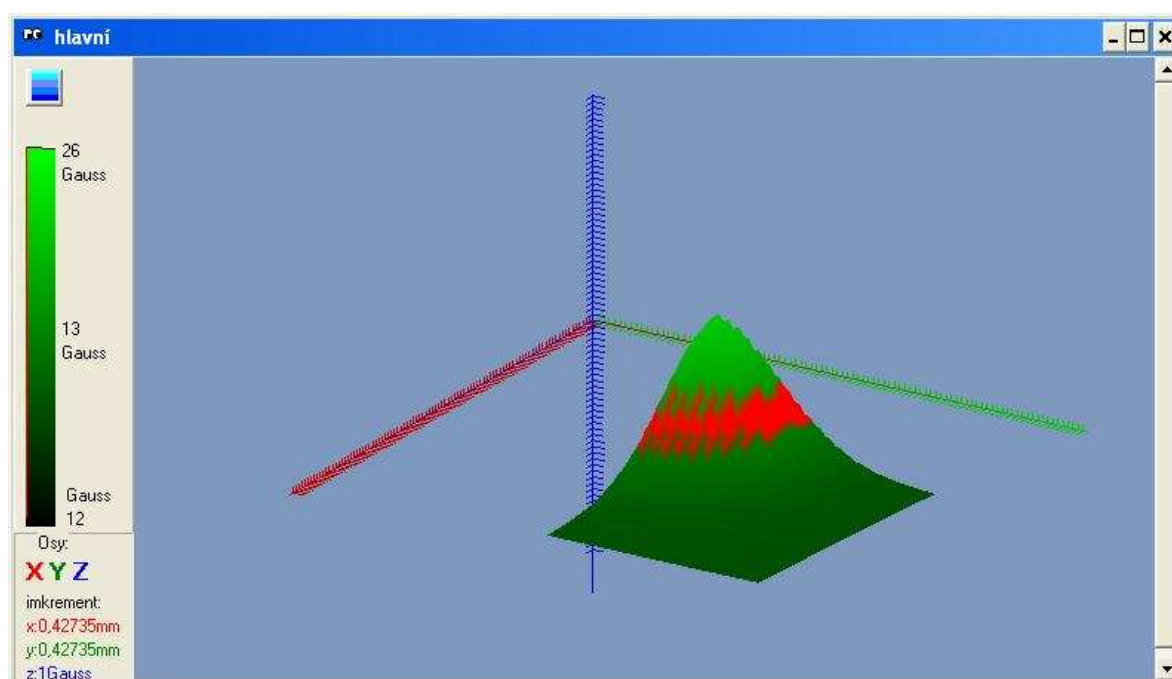
Obr. 47 Řez magnetickým polem šachové figurky v polovině osy X a Y



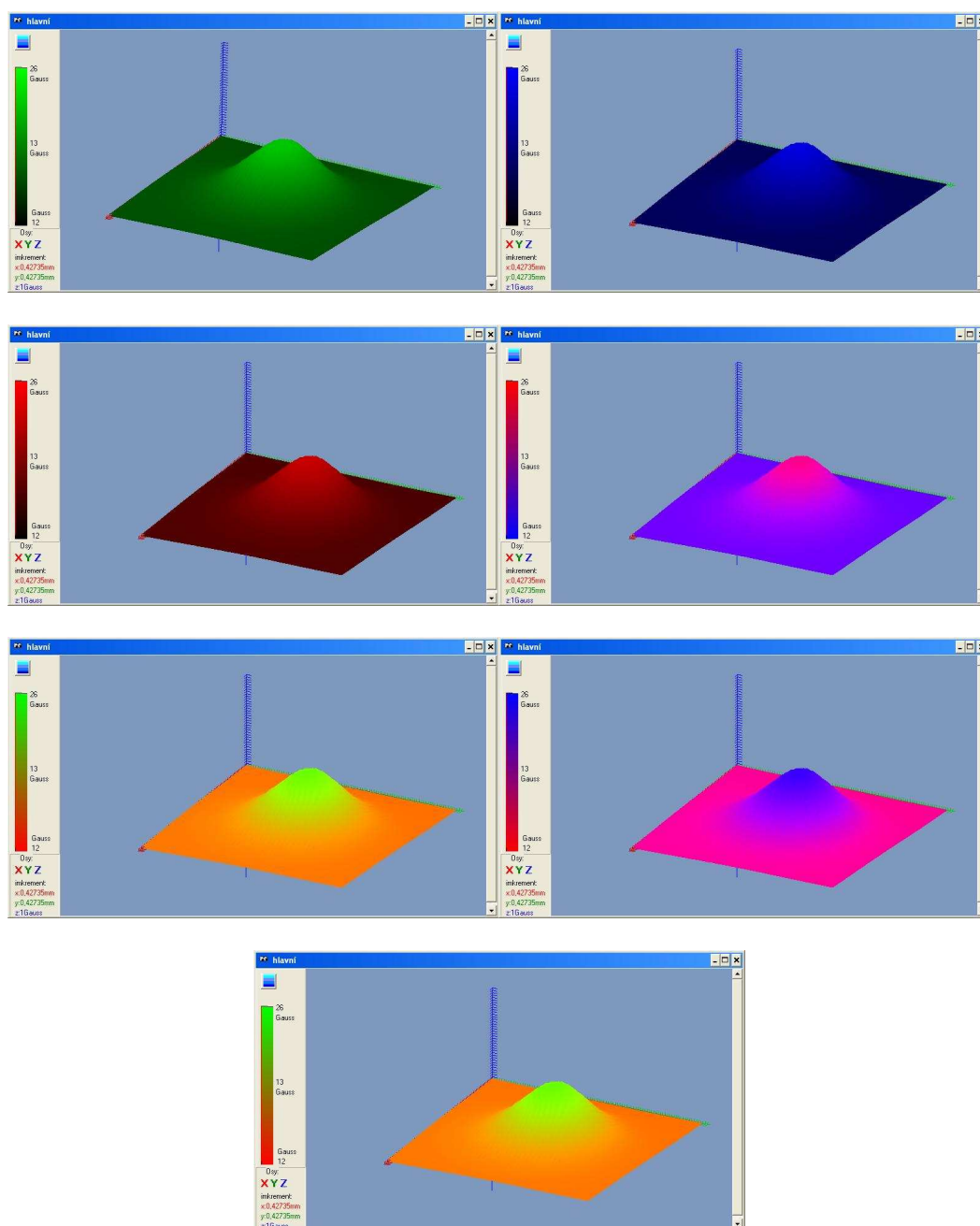
Obr. 48 Ukázka možností řezu



Obr. 49 Zobrazení kritické, uživatelem nadefinované oblasti



Obr. 50 Zobrazení kritické, uživatelem nadefinované oblasti + řez v osách



Obr. 51 Ukázky možností vybarvení povrchu

Oskenování magnetu rotoru z disketové mechaniky

Po zdařilém naskenování jednoduchého magnetického pole byl vybrán jako zdroj složitějšího magnetického pole rotor elektromotorku z disketové mechaniky. Je dosti lehký, aby ho plotr unesl, ale přesto má dosti složité magnetické pole. Na obvodní straně se nachází tenký feritový magnet na němž se po obvodě 5x vystřídá severní a jižní pól.

Průměr skenovaného magnetu je 60mm

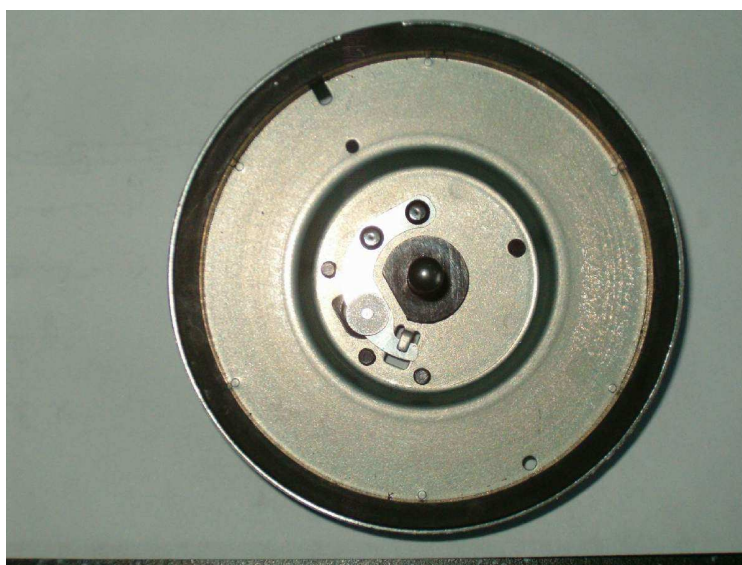
Velikost skenované oblasti byla experimentálně zvolena :120x120mm

Inkrement byl jako u předcházejícího případu zvolen 0,42mm v obou osách. Zobrazené magnetické pole se skládá z 81632 bodů.

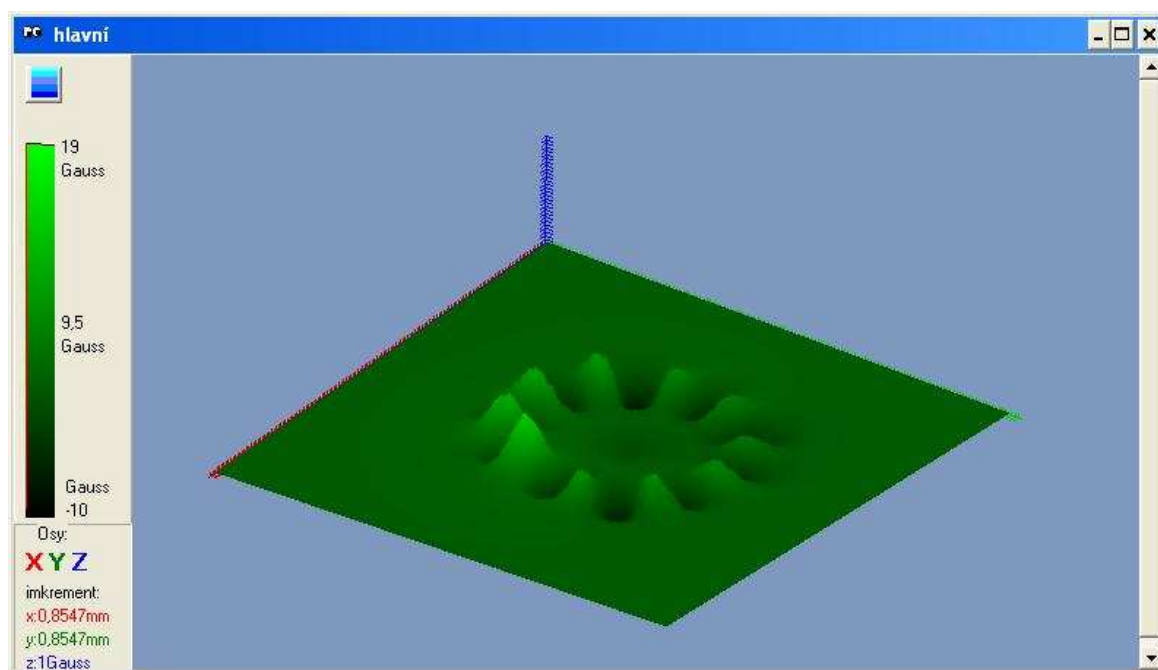
Typ skenování byl vybrán jako v předcházejících případech



Obr. 52 Skenování magnetu

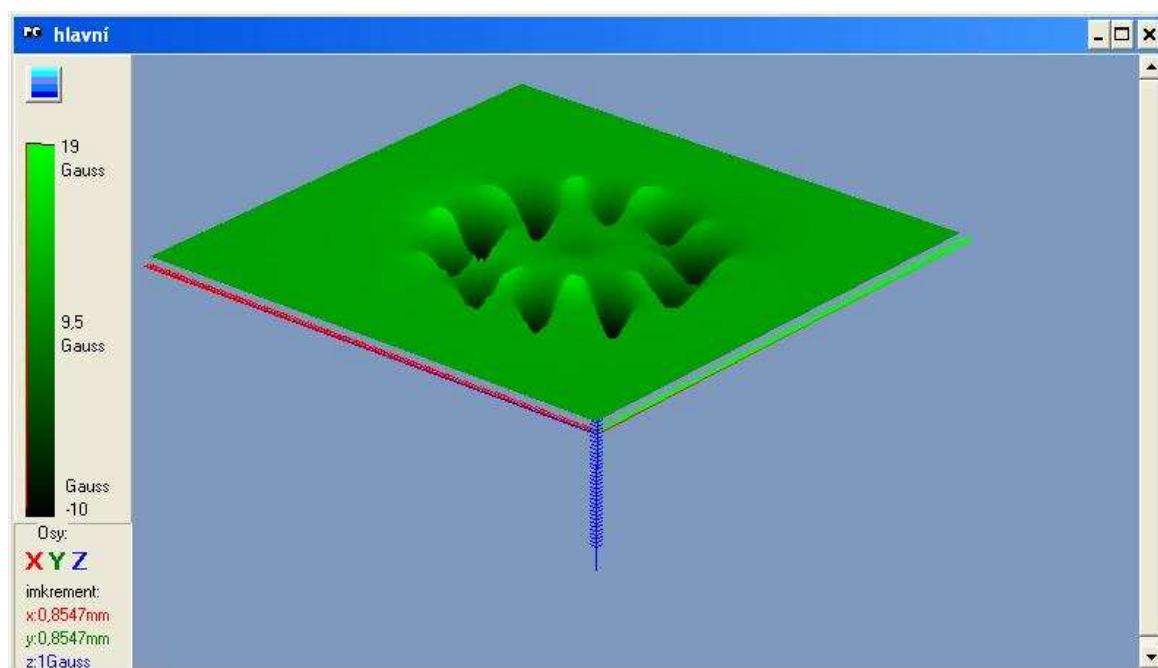


Obr. 53 Skenovaný magnet rotoru (průměr 6 cm)

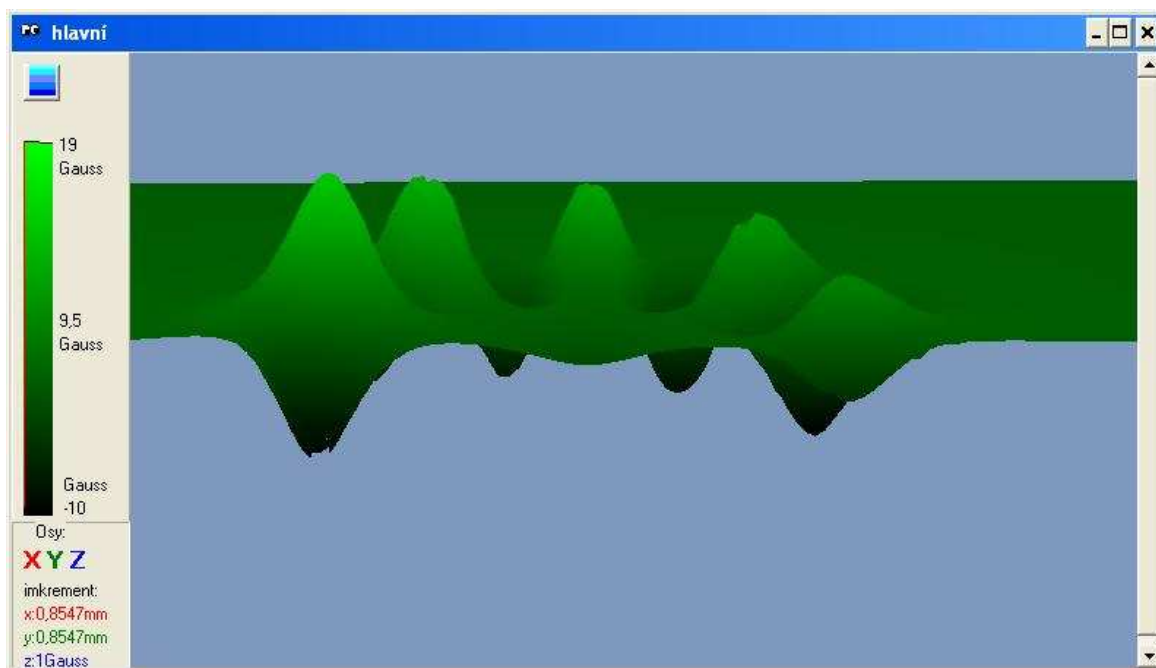


Obr. 54 Oskenované magnetické pole rotorku

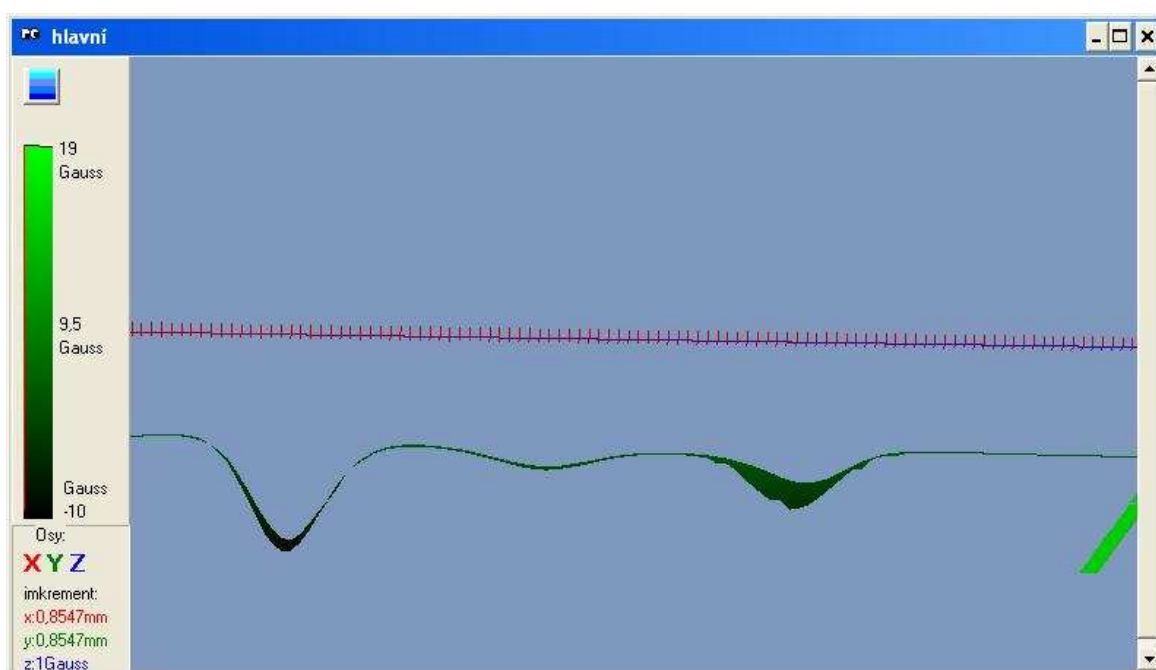
Na oskenovaném obrázku jde vidět, jak byl rotor nakloněný (nebyl položen vodorovně na plotr).



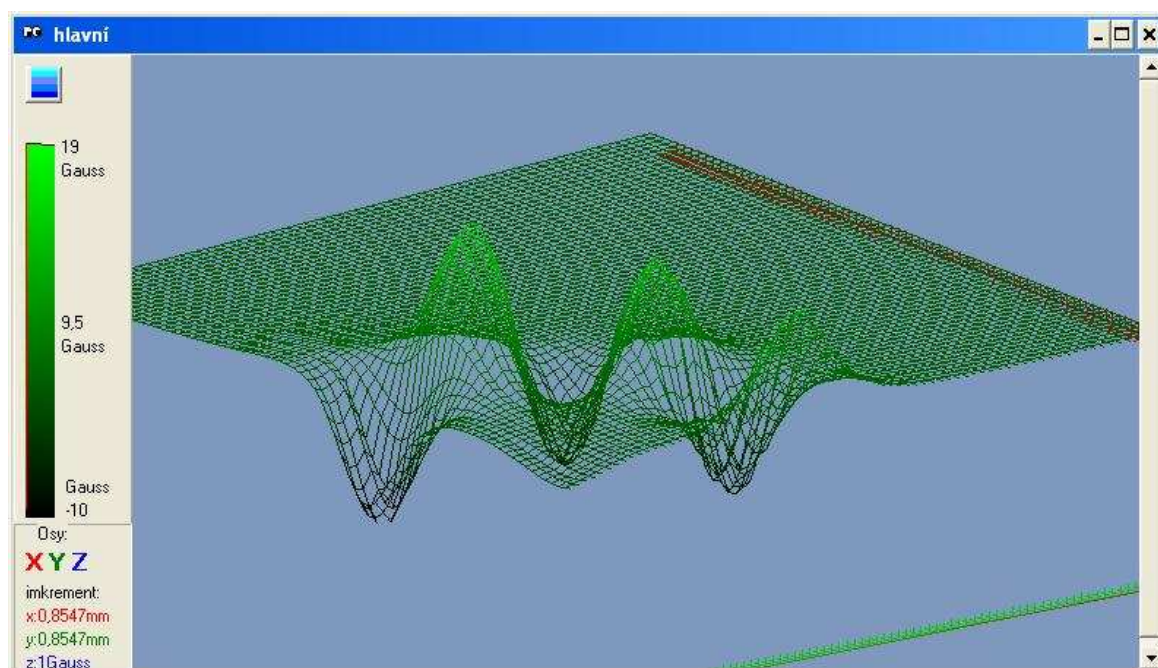
Obr. 55 Pohled na spodní stranu naskenovaného pole



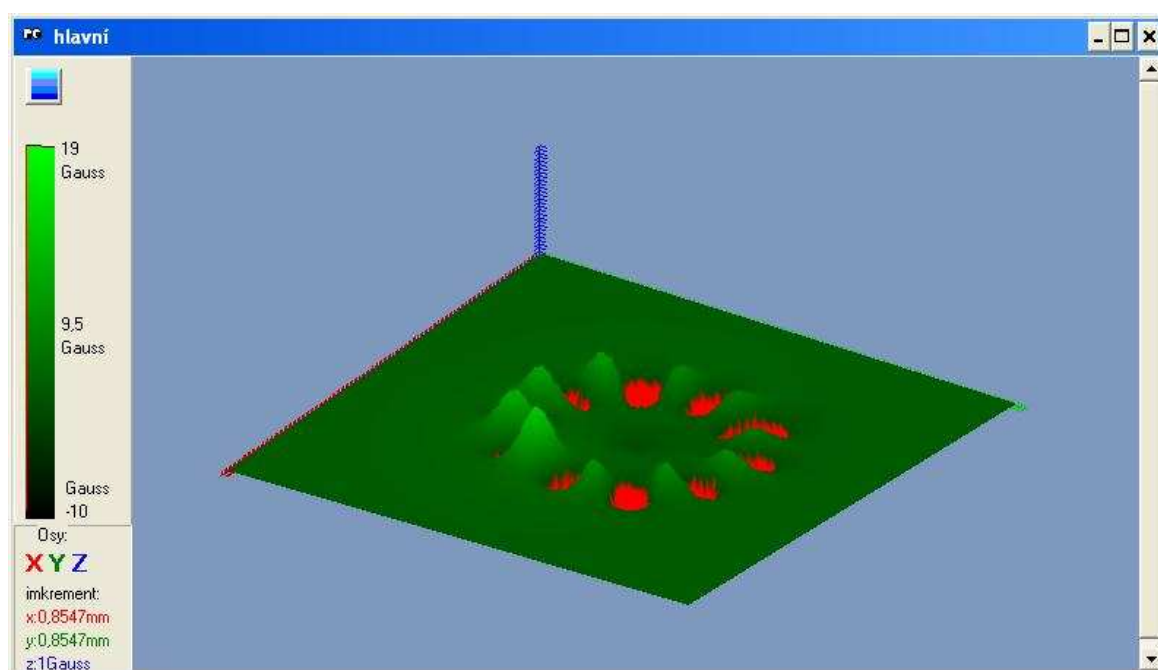
Obr. 56 Řez plochou v ose x



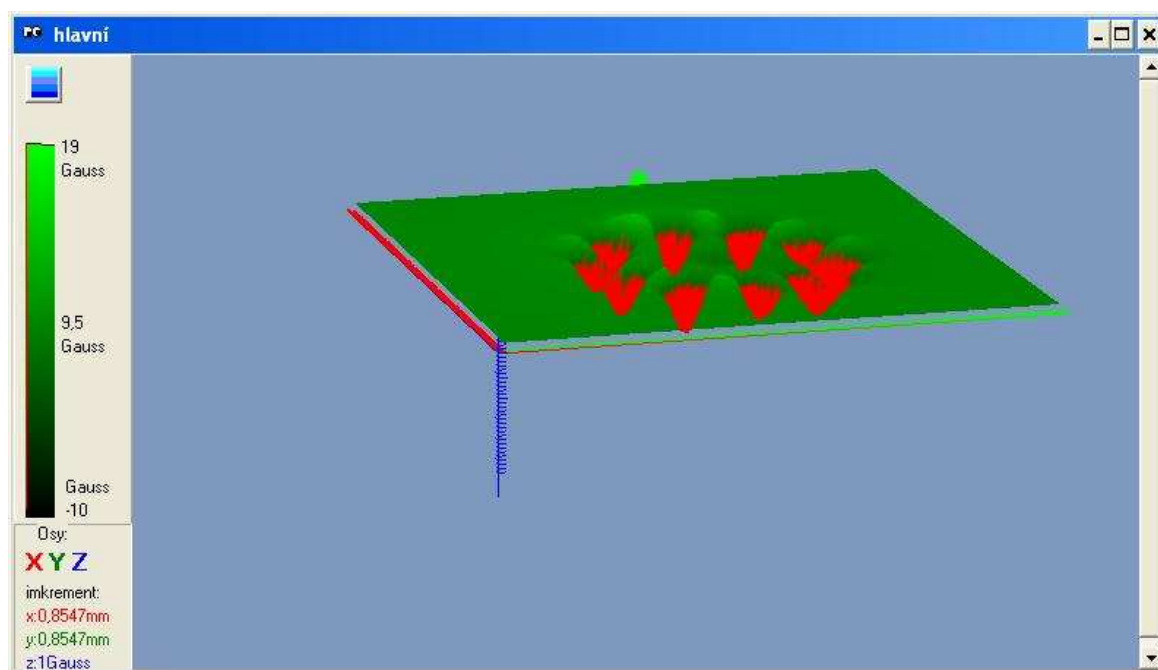
Obr. 57 Velice tenký řez, tloušťka v ose Y ve středu tloušťky 5 bodů
($5 \cdot 0,42\text{mm} = 2,1\text{mm}$)



Obr. 58 Řez v obou osách, síťový model



Obr. 59 Zobrazení severního a jižního pólu (zadáno krit.oblast od -10-0 vybarvy červeně)



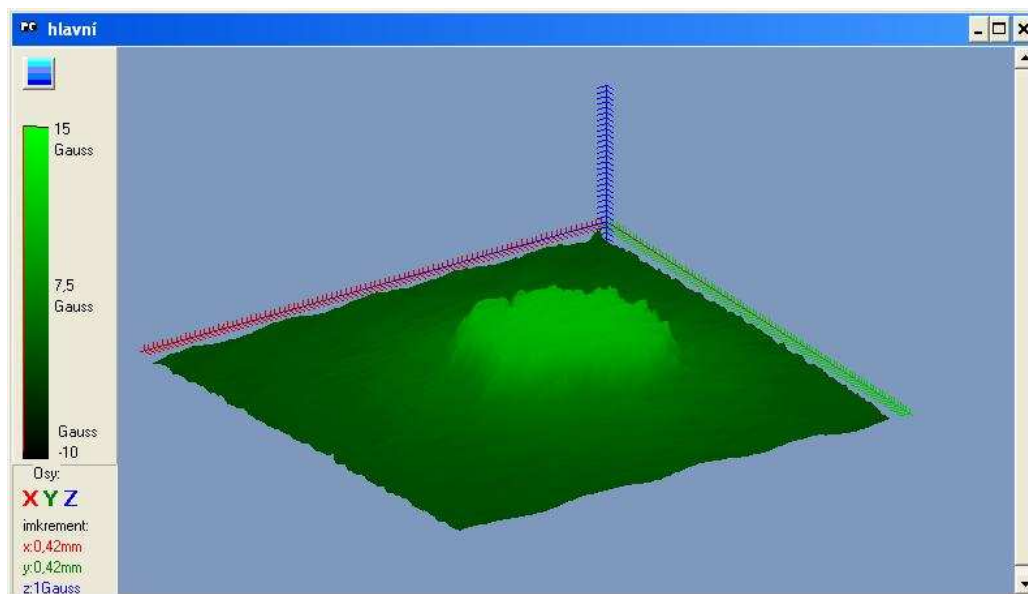
Obr. 60 Zobrazení severního a jižního pólu dolní pohled

12 POROVNÁNÍ NASKENOVANÝCH DAT ANALOGOVĚ/RS323

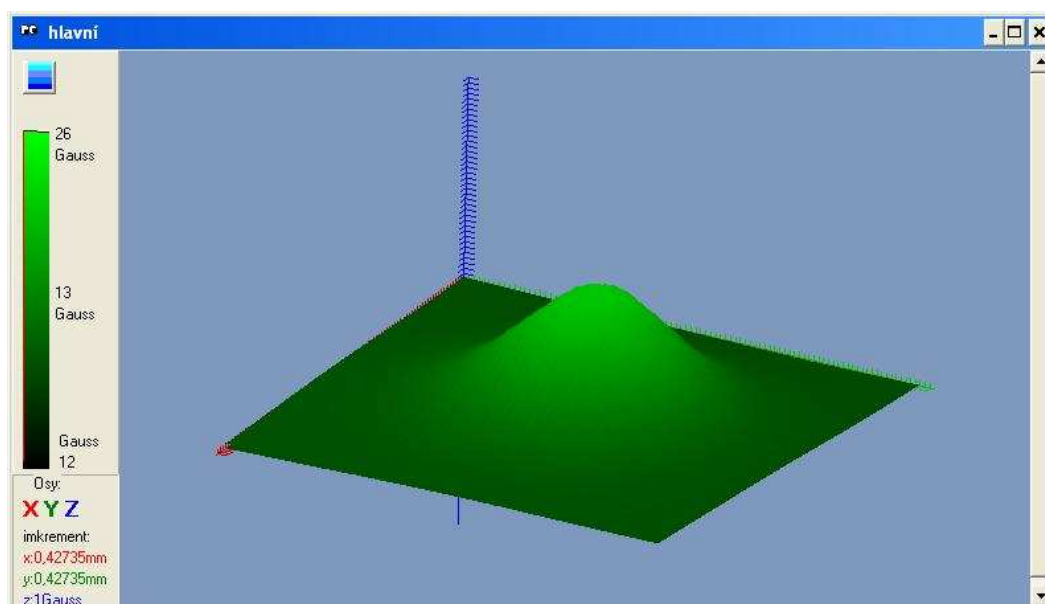
K porovnání byla použita skenáž šachové figurky skenované přes analogový vstup a pomocí komunikace po RS232.

Velikos skenované oblasti byla v obou případech 40x40mm

Inkrement byl v obou případech 0,42mm v obou osách.



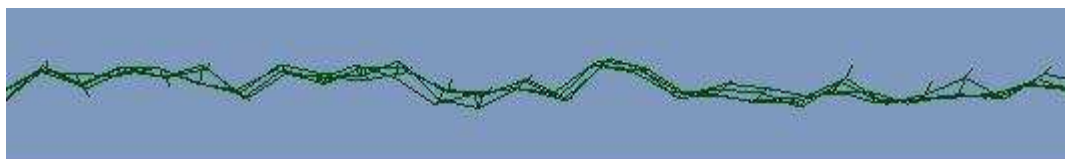
Obr. 61 Analogový sken



Obr. 62 Sken přes RS232

Jde vidět obrovských rozdílů vzniklých zašuměním vstupu analogové karty.

Pro porovnání velikosti šumu byl zvětšen okraj naměřených ploch.



Obr. 63 Zvětšený okraj skenáže přes analogový vstup



Obr. 64 Zvětšený okraj skenáže přes vstup RS232

Po zvětšení okraje naskenované oblasti vidět obrovské rozdíly vzniklé šumem.

Propojení počítače s měřícím přístrojem přes RS232 má za následek načítání přesných hodnot, bez šumu. Nemusíme hodnoty průměrovat a tím mnohonásobně urychlíme skenování.

Skenáž přes analogový vstup karty advantech trvala přes 2 hodiny vlivem načítání 6ti hodnot pro jeden bod. Celkový počet načtených hodnot se tak zvýšil 6x!

Skenáž přes port RS232 trvala 50 minut. Načítala se pouze jediná hodnota pro bod a i přes to je skenáž naprosto bezšumová.

Mírný šum na obrázku skenáže přes RS232 na pravé straně je způsoben tím, že tím směrem byl umístěn magnet a plocha se již v těchto místech začínala zakřivovat.

ZÁVĚR

V dnešní civilizované společnosti se stále více používá elektropřístrojů využívající vysokých a nebezpečných frekvencí. Tyto frekvence jsou vyzařovány formou elektromagnetického záření, které je pro člověka nebezpečné. Každý máme počítač, televizi, v kuchyni mikrovlnou troubu a v kapse mobilní telefon. Tyto přístroje jsou nám pořád nablízku. Vlivem své činnosti vyzařují elektromagnetické záření, které je potřeba měřit, mapovat a navrhovat dokonalé filtry k jeho odstranění.

Dalším využitím jsou mlékárny. Mléko, než se dostane na Váš stůl ve formě sýrů, jogurtů aj. projde spoustou šnekových, zubových a jiných čerpadel. Ty jsou vyráběny z kovu. Kov podléhá časem opotřebení a může dojít k uvolnění kousku kovu do mléka. Pokud by se tento kousek dostal do potravin, za nedlouho by tu byla žaloba od spotřebitele, tisk by to rozepsal a lidé by tyto výrobky přestali kupovat, což by vedlo k finančnímu kolapsu mlékárny. Tyto mlékárny uvolňují ročně spousty peněz na výrobu magnetických filtrů „lapačů“ těchto nečistot. Spoustu práce, při výrobě těchto filtrů, by bylo ušetřeno vizualizací průběhu magnetických polí nad nimi.

Při výrobě od drobných elektromotorků do počítačových mechanik až po obrovské elektromotory je velice důležité zmapovat rozložení magnetického pole jak nad statorem, tak rotorem pro přesné nastavení spínání cívek motoru. Tím dosáhneme maximální účinnosti při minimálním odběru proudu.

Tímto skenerem nemusíme skenovat jen magnetické pole. Můžeme sem umístit světelné čidlo a mapovat odrazivost světla od různých povrchů, umístit sem elektrostatická nebo jiná čidla a mapovat elektrostatické aj. pole a v neposlední řadě tento skener můžeme použít jako kapacitní mikroskop pro mikroskopická zkoumání. S touto prací bych rád pokračoval v příštích studiích a diplomové práci. V plánu na příští rok je již přichystán můstek s hrotem, který bude aplikován na plotter a data následně na zobrazovací program. Toto zapojení bude fungovat na principu kapacitního mikroskopu.

SEZNAM POUŽITÝCH ZDROJŮ

. Monografické publikace:

- 1) Manuál k Advantechu PCL812PG
- 2) Manuál k LakeShoru 421 gaussmetru

WWW stránky

- 3) <http://delphi3d.net/links.php>
- 4) <http://www.exsanity.freeola.com/links.html>
- 5) <http://www.savagesoftware.com.au/DelphiGamer/links.php?PHPSESSID=0715a7b945d7e1300539c0bb358a958f#OpenGL>
- 6) http://clootie.narod.ru/delphi/download_custom.html
- 7) <http://www.gametutorials.com/>
- 8) <http://www.planet-source-code.com/vb/default.asp?lngWId=7>
- 9) <http://www.delphi.cz/odkazy.asp?skupina=doc&order=>
- 10) <http://www.shadow3d.de.vu/>
- 11) <http://jca.developpez.com/opengl.php>
- 12) komponenty pro opengl www.builder.cz

Programy

- 1) soubor příkladů pro delphi ke kartě od firmy Advantech

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Aj. A jiné

Atd. a tak dál

Apod. a podobně

Form formulář programu (hlavní vizualizační jádro)

SEZNAM OBRÁZKŮ

Obr. 1. použité vstupy/výstupy karty PCL812PG	8
Obr. 2. laboratorní karta PCL812PG	8
Obr. 3. analogový plotter	9
Obr. 4. popis plotru	10
Obr. 5. přístroj LakeShore 421	11
Obr. 6. popis použité sondy	11
Obr. 7. používaná tlačítka přístroje LakeShore 421	12
Obr. 8 ukázka počátečních bodů skenování	14
Obr. 9 skenování oboustranně	17
Obr. 10 skenování jednostranně	17
Obr. 11 zobrazení vstupně/výstupních hodnot	22
Obr. 12. ruční řízení plotru	24
Obr. 13 formulář skenování 1/3.....	26
Obr. 14 formulář skenování 2/3.....	28
Obr. 15. formulář skenování 3/3.....	30
Obr. 16. program pokus.vee v Agilen VEE pro 7.....	35
Obr. 17 vertex	37
Obr. 18 hlavní formulář řízení plotru	47
Obr. 19 formulář výstupů a vstupů karty advantech.....	47
Obr. 20 formulář ručního řízení plotru	48
Obr. 21 formulář skenování 1/3.....	49
Obr. 22. ukázka vyhodnocení	50
Obr. 23 ukázka vyhodnocení	50
Obr. 24. ukázka vyhodnocení rozlišení	51
Obr. 25 hláška najíždění	51
Obr. 26 hláška konec najíždění.....	52
Obr. 27 typ skenování.....	52
Obr. 28 formulář skenování 2/3.....	53
Obr. 29. formulář skenování 3/3.....	54
Obr. 30 popis ovládacího panelu zobrazovacího programu	57
Obr. 31 střed rotace.....	57
Obr. 32. formulář dat	58
Obr. 33 zapnutí osového kříže	58
Obr. 34 výběr typu plochy	59
Obr. 35 formulář hledání min. max.	59
Obr. 36 formulář časové rotace.....	60
Obr. 37. nastavení barvy pozadí	60
Obr. 38 data o skenovaném objektu.....	61
Obr. 39 řez objektem	61
Obr. 40 vystínování objektu	62
Obr. 41 Skenáž šachové figurky	65
Obr. 42 Zobrazení oskenovaných hodnot pře analog. Vstup.....	66
Obr. 43 Skenáž šachové figurky + rozložení magnetického pole.	67
Obr. 44 Skenáž magnetického pole šachové figurky přes RS232	67
Obr. 45 Síťový model magnetického pole figurky	68
Obr. 46 Řez magnetickým polem šachové figurky v polovině osy X	68
Obr. 47 Řez magnetickým polem šachové figurky v polovině osy X a Y	69

Obr. 48 Ukázka možností řezu	69
Obr. 49 Zobrazení kritické, uživatelem nadefinované oblasti	70
Obr. 50 Zobrazení kritické, uživatelem nadefinované oblasti + řez v osách.....	70
Obr. 51 Ukázky možností vybarvení povrchu	71
Obr. 52 Skenování magnetu.....	72
Obr. 53 Skenovaný magnet rotoru (průměr 6 cm).....	72
Obr. 54 Oskanované magnetické pole rotorku	73
Obr. 55 Pohled na spodní stranu naskenovaného pole	73
Obr. 56 Řez plochou v ose x.....	74
Obr. 57 Velice tenký řez, tloušťka v ose Y ve středu tloušťky 5 bodů ($5 \cdot 0,42\text{mm} = 2,1\text{mm}$)	74
Obr. 58 Řez v obou osách, síťový model.....	75
Obr. 59 Zobrazení severního a jižního pólu (zadáno krit.oblast od -10-0 vybarvy červeně)	75
Obr. 60 Zobrazení severního a jižního pólu dolní pohled	76
Obr. 61 Analogový sken	77
Obr. 62 Sken přes RS232.....	77
Obr. 63 Zvětšený okraj skenáže přes analogový vstup.....	78
Obr. 64 Zvětšený okraj skenáže přes vstup RS232	78

SEZNAM TABULEK

Tab. 1 rozlišení přístroje LakeShore 421.....	12
---	----

;