

W celu ułatwienia konfiguracji do niniejszej instrukcji przygotowany został plik tekstowy **LinuxCNC – CDN.txt** w którym znajdują się wszystkie listingi programów i konfiguracji które można skopiować i wklejać zgodnie z instrukcją. Plik **LinuxCNC – CDN.txt** jest dostępny do pobrania pod adresem:

[http://www.machmaker.pl/1\\_5\\_pliki-do-pobrania.html](http://www.machmaker.pl/1_5_pliki-do-pobrania.html)

Plik składa się z sekcji oznaczonych kolejno w następujący sposób

[----1----]

Listing1

[----2----]

Listing2

[----3----]

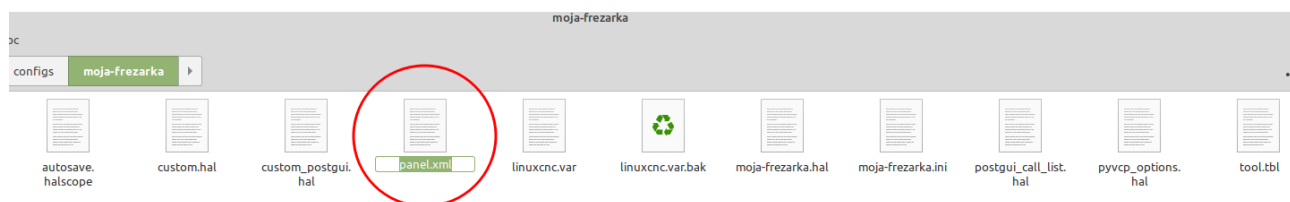
Listing3

itd.

Kopiować należy jedynie Listing, bez oznaczenia sekcji [----1----],[----2----],[----3----] itd.

## 1. Utworzenie przycisku do pomiaru długości narzędzia

W katalogu z ustawieniami linuxcnc (tam gdzie między innymi znajduje się plik .ini) tworzymy nowy dokument (np. poprzez edytor tekstu) i zapisujemy go pod nazwą **panel.xml**

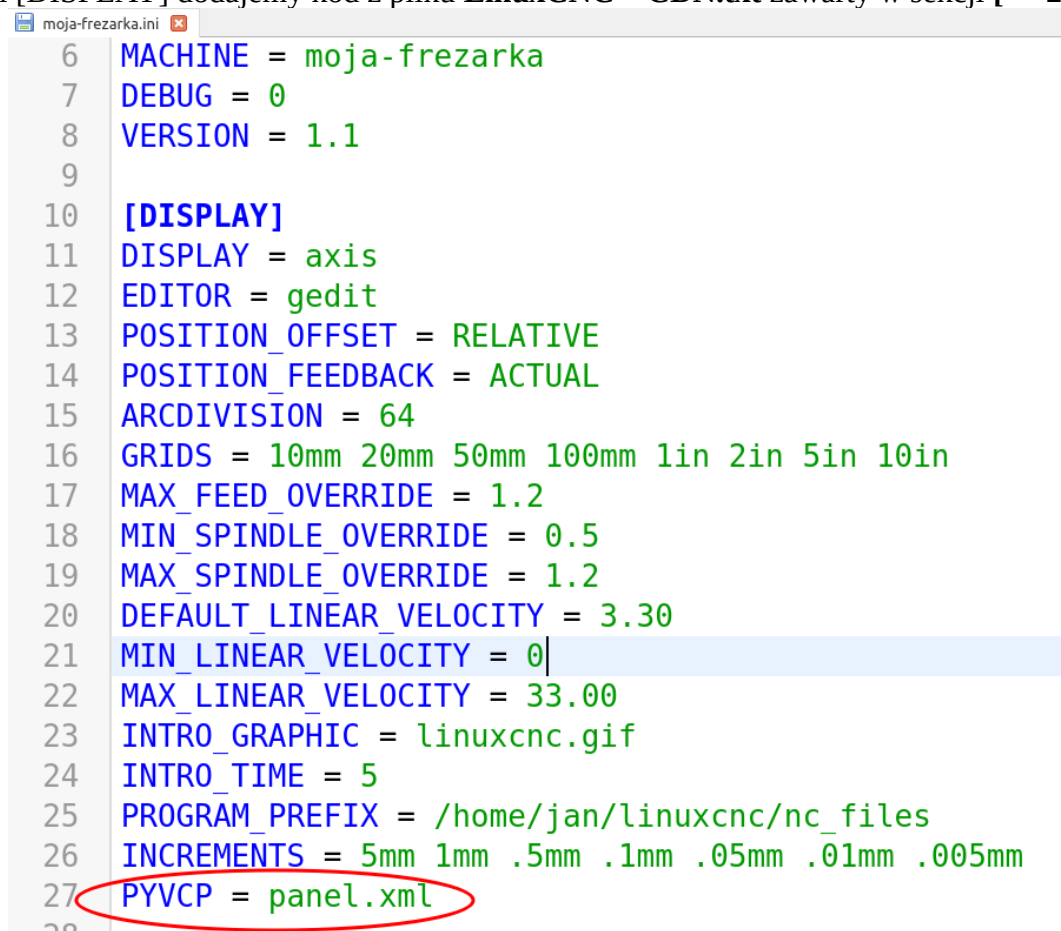


Otwieramy dokument w dowolnym edytorze tekstu i tworzymy własny przycisk wklejając kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [----1----]

```
panel.xml x
1 <?xml version='1.0' encoding='UTF-8'?>
2 <pyvcp>
3 <button>
4   <halpin>"wymiana_narzedzia"</halpin>
5   <text>"Nowe narzędzie"</text>
6   <font>('fixed',10)</font>
7 </button>
8 </pyvcp>
```

Tekst w cudzysłowie w linii 4 oznacza nazwę pliku z programem który będzie odpowiedzialny za obsługę pomiaru długości narzędzia. Plik stworzymy później.  
Tekst w linii 5 to nazwa jaka będzie wyświetlana na przycisku.  
Zapisujemy plik i zamykamy.

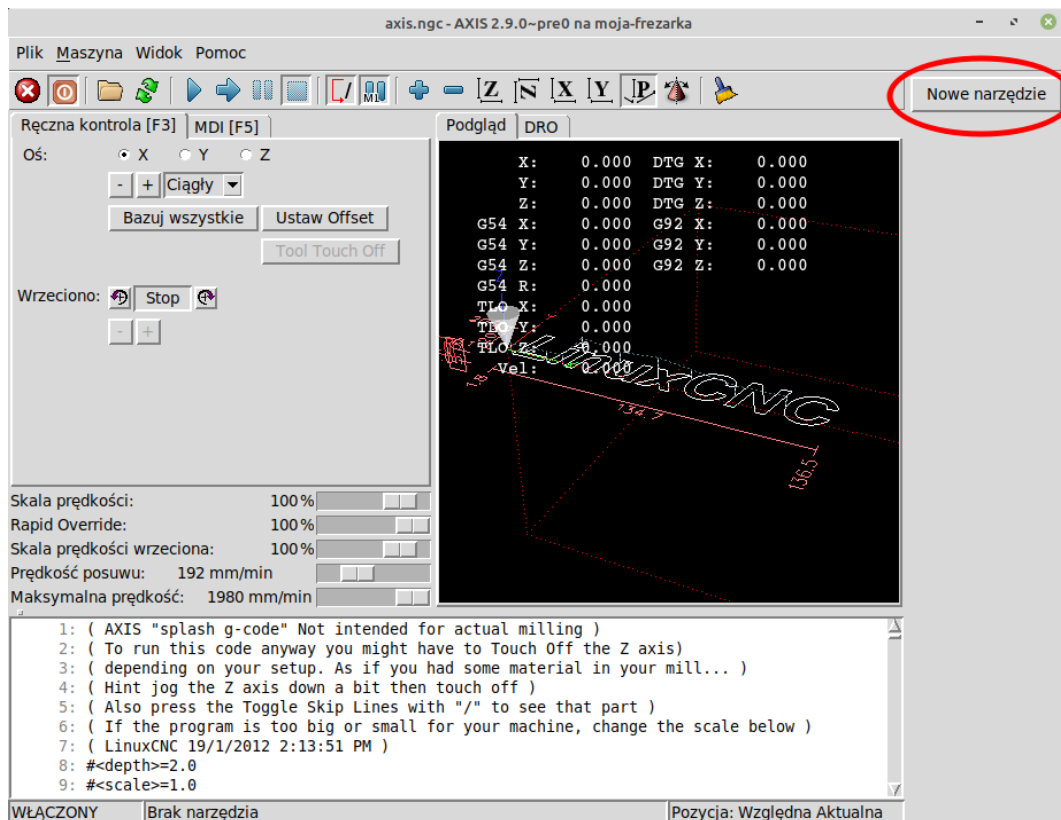
W edytorze tekstu otwieramy plik z ustawieniami linuxcnc czyli plik z rozszerzeniem .ini  
W sekcji [DISPLAY] dodajemy kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [----2----



```
6 MACHINE = moja-frezarka
7 DEBUG = 0
8 VERSION = 1.1
9
10 [DISPLAY]
11 DISPLAY = axis
12 EDITOR = gedit
13 POSITION_OFFSET = RELATIVE
14 POSITION_FEEDBACK = ACTUAL
15 ARCDIVISION = 64
16 GRIDS = 10mm 20mm 50mm 100mm 1in 2in 5in 10in
17 MAX_FEED_OVERRIDE = 1.2
18 MIN_SPINDLE_OVERRIDE = 0.5
19 MAX_SPINDLE_OVERRIDE = 1.2
20 DEFAULT_LINEAR_VELOCITY = 3.30
21 MIN_LINEAR_VELOCITY = 0
22 MAX_LINEAR_VELOCITY = 33.00
23 INTRO_GRAPHIC = linuxcnc.gif
24 INTRO_TIME = 5
25 PROGRAM_PREFIX = /home/jan/linuxcnc/nc_files
26 INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm
27 PYVCP = panel.xml
```

Zapisujemy plik i zamykamy.

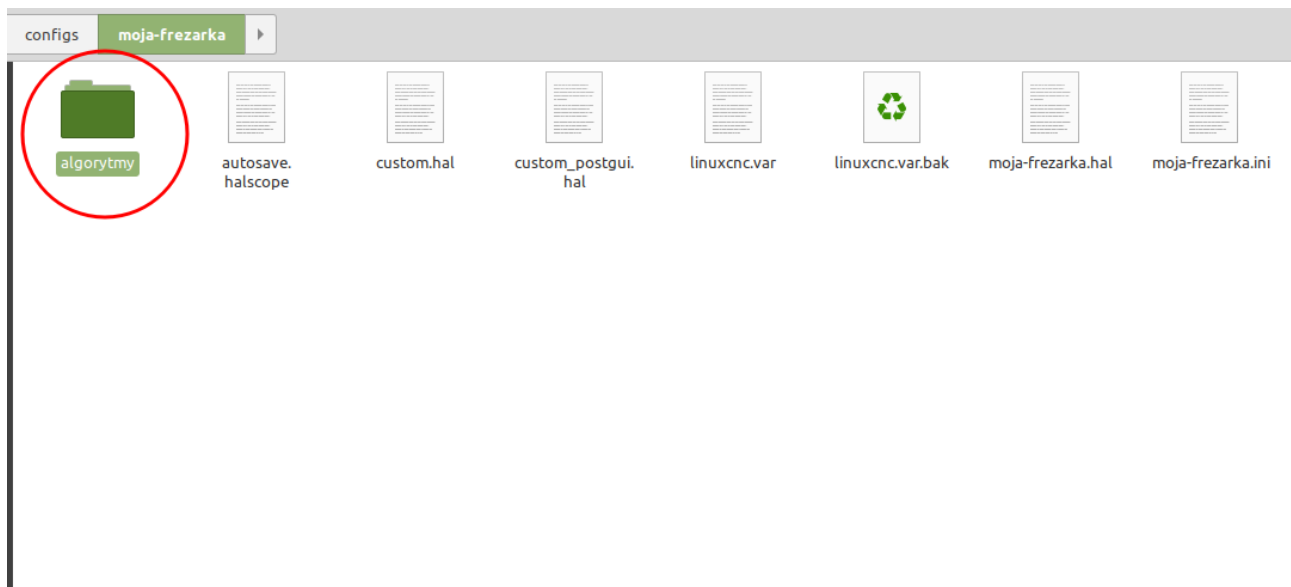
Uruchamiamy LinuxCNC, na panelu powinien być widoczny nasz nowy przycisk.



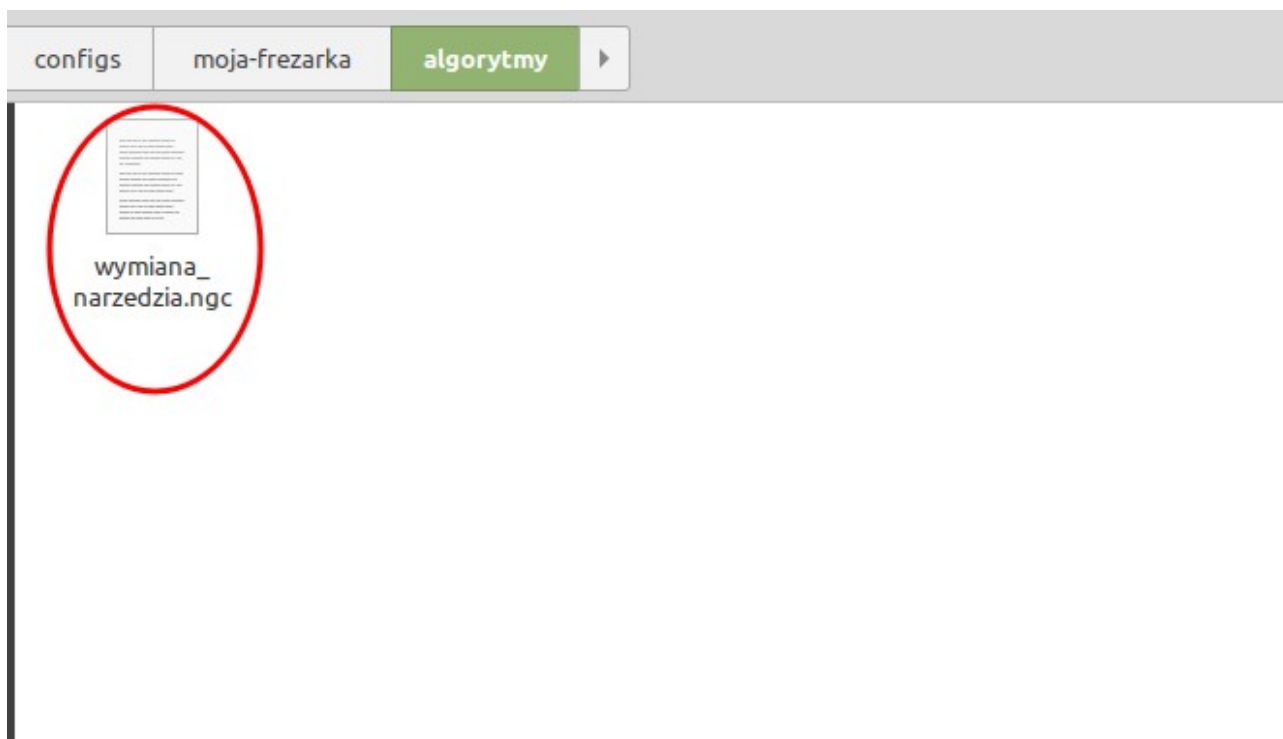
Zamykamy LinuxCNC.

## 2. Stworzenie pliku z programem wykonującym procedurę wymiany i pomiaru długości narzędzia

W katalogu z ustawieniami maszyny tworzymy katalog w którym znajdować się będą nasza algorytmy (czyli na początek program realizujący obsługę wymiany i pomiaru narzędzia)



Przechodzimy do katalogu algorytmy i tworzymy w nim plik z programem (za pomocą edytora tekstu) o nazwie jaką podaliśmy w pliku panel.xml



Teraz kilka słów o tym jak zaplanowałem swoją procedurę wymiany i pomiaru długości narzędzia:

- 1) Maszyna ustawia się do wymiany narzędzia na zdefiniowanej pozycji wymiany i wyświetla pierwszy komunikat aby zamontować/wymienić narzędzie. Po zamontowaniu nowego narzędzia operator wciska przycisk kontynuuj.
- 2) Maszyna ustawia się na zdefiniowanej pozycji w której znajduje się czujnik długości narzędzia i wyświetla drugi komunikat aby zamontować czujnik. (a to dlatego, że nie każda maszyna posiada czujnik zamontowany na stałe. Niektóre posiadają również czujniki dotykowe, które wymagają aby do narzędzia podłączyć przewód. To jest właśnie ten czas aby zamontować czujnik i/lub zamontować przewód). Po zamontowaniu czujnika operator wciska przycisk kontynuuj.



3) Maszyna obniża narzędzie w osi Z ze zdefiniowaną prędkością aż do dotknięcia narzędziem czujnika. Po dotknięciu czujnika oś Z podnosi się bardzo powoli do góry aż do momentu gdy narzędzie przestanie dotykać czujnika i ta pozycja używana jest do ustalenia długości narzędzia. Po zakończeniu procedury narzędzie podnosi się do góry w osi Z do zdefiniowanej pozycji czujnika długości narzędzia i wyświetla trzeci komunikat aby zdemontować czujnik. Po zdemontowaniu czujnika operator klika przycisk kontynuuj i maszyna ustawia się na zdefiniowanej pozycji po wymianie czujnika i jest gotowa do pracy z nowym narzędziem.

A teraz program:

Otwieramy plik wymiana\_narzedzia.ngc w edytorze tekstu wklejamy kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [---3---]

```
0 <wymiana_narzedzia> sub
#<_ActualMeasureZ> = 0

0100 IF [ EXISTS[#<_FirstTool>] EQ 0 ]
    #<_FirstTool> = 0
    G49
0100 ENDIF

G43
M70
G21
G94
G90
G53 G0 Z0
G53 G0 X#<_ini[TOOL_CHANGE_POS]> Y#<_ini[TOOL_CHANGE_POS]>
G53 G0 Z#<_ini[TOOL_CHANGE_POS]>
M100 P0
G53 G0 Z0
G53 G0 X#<_ini[TOOLSENSOR]X> Y#<_ini[TOOLSENSOR]Y>
G53 G0 Z#<_ini[TOOLSENSOR]Z>
M100 P1
G91

0200 IF [ #<_FirstTool> EQ 0 ]
    G38.3 Z-#<_ini[TOOLSENSOR]MAX_CHANGE> F50
    G38.4 Z10 F1
    #<_FirstMeasureZ> = #5063
    (MSG, Zmierzone pierwsze narzędzie)
0200 ELSE
    G38.3 Z-#<_ini[TOOLSENSOR]MAX_CHANGE> F50
    G38.4 Z10 F1
    #<_ActualMeasureZ> = #5063
    G43.1 Z[#<_ActualMeasureZ> - #<_FirstMeasureZ>]
    (MSG, Zmierzone kolejne narzędzie)
0200 ENDIF

G90
G53 G0 Z#<_ini[TOOLSENSOR]Z>
M100 P2
G53 G0 Z #<_ini[AFTER_CHANGE_POSITION]Z>
G53 G0 X #<_ini[AFTER_CHANGE_POSITION]X> Y #<_ini[AFTER_CHANGE_POSITION]Y>
M72

0300 IF [ #<_FirstTool> EQ 0 ]
    #<_FirstTool> = 1
0300 ELSE
    G43.1 Z[#<_ActualMeasureZ> - #<_FirstMeasureZ>]
0300 ENDIF

0 <wymiana_narzedzia> endsub
m2
```

(Tworzy parametr <\_ActualMeasureZ>)

(Jeśli nie istnieje parametr flaga <\_FirstTool>)  
(Tworzy flagę <\_FirstTool> i przypisuje jej wartość 0)  
(Anuluje kompensację długości narzędzia)

(Używa kompensacji długości narzędzia z tablicy)  
(Zapisuje aktualny stan)  
(Jednostką odległości są [mm])  
(Jednostką ruchu jest [jednostka/min])  
(Programowanie absolutne)  
(Podnosi oś Z do bezpiecznej wysokości - w tym przypadku maksymalna wysokość = 0)  
(Przesuwa w osi X i Y do zdefiniowanego punktu wymiany narzędzia)  
(Przesuwa w osi Z do zdefiniowanego punktu wymiany narzędzia)  
(Wyświetla komunikat aby zamontować/zmienić narzędzie. Czekaj na kliknięcie "Kontynuuj")  
(Podnosi oś Z do bezpiecznej wysokości - w tym przypadku maksymalna wysokość = 0)  
(Przesuwa w osi X i Y do zdefiniowanego punktu pomiaru długości narzędzia)  
(Przesuwa w osi Z do zdefiniowanego punktu pomiaru długości narzędzia)  
(Wyświetla komunikat aby zamontować czujnik. Czekaj na kliknięcie "Kontynuuj")  
(Programowanie przyrostowe)

(Jeśli pomiar pierwszego narzędzia)  
(Obniża narzędzie do momentu kontaktu)  
(Podnosi narzędzie do utracenia kontaktu)  
(Zapisuje wynik pierwszego pomiaru w osi Z)

(Jeśli pomiar kolejnego narzędzia)  
(Obniża narzędzie do momentu kontaktu)  
(Podnosi narzędzie do utracenia kontaktu)  
(Zapisuje wynik aktualnego pomiaru w osi Z)  
(Ustawia dynamicznie offset długości aktualnego narzędzia - działa do momentu wystąpienia M72)

(Programowanie absolutne)  
(Podnosi oś Z do zdefiniowanego punktu czujnika)  
(Wyświetla komunikat aby zdemontować/zabezpieczyć czujnik)  
(Przesuwa oś Z do zdefiniowanego punktu po wymianie narzędzia)  
(Przesuwa oś X,Y do zdefiniowanego punktu po wymianie narzędzia)  
(Przywraca zapisany stan)

(Jeśli pomiar pierwszego narzędzia)  
(Ustawia flagę <\_FirstTool>)

(Ponownie ustawia dynamicznie offset długości aktualnego narzędzia)

Zapisujemy i zamykamy plik **wymiana\_narzedzia.ngc**

### 3. Zdefiniowanie współrzędnych punktu wymiany narzędzia, punktu pomiaru narzędzia, punktu po wymianie narzędzia oraz zakresu pomiaru długości narzędzia

Wartości definiujemy w pliku z ustawieniami czyli otwieramy w edytorze tekstu plik z rozszerzeniem .ini i dodajemy definicje wartości wykorzystywanych w programie wymiany długości narzędzia wklejając kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [---4---]

```
moja-frezarka.ini
55 [HAL]
56 HALFILE = moja-frezarka.hal
57 HALFILE = custom.hal
58 POSTGUI_HALFILE = postgui_call_list.hal
59
60 #Współrzedne punktu wymiany narzędzia (maszynowe)
61 [TOOL_CHANGE_POS]
62 X = 120
63 Y = 0
64 Z = 0
65
66 #Współrzedne czujnika długości narzędzia (maszynowe)
67 [TOOLSENSOR]
68 X = 220
69 Y = 13
70 Z = -48
71 MAX_CHANGE = 32
72
73 #Współrzedne punktu po zmianie i pomiarze narzędzia (maszynowe)
74 [AFTER_CHANGE_POSITION]
75 X = 120
76 Y = 0
77 Z = 0
78
79 [TRAJ]
```

Linia 62-64: Współrzedne maszynowe na których ustawia się maszyna do wymiany narzędzia

Linia 68-70: Współrzedne maszynowe na których ustawia się maszyna do pomiaru długości narzędzia. Tak jak pisałem wyżej, na osi Z musi to być wartość ok 2-3mm nad czujnikiem długości narzędzia dla najdłuższego narzędzia jakie montujemy w maszynie.

Linia 71: Wartość zakresu pomiaru (jaki zakres w osi Z obniża się narzędzie w oczekiwaniu na zadziałanie czujnika długości narzędzia). Wartość ta po dodaniu do wartości jaką w osi Z pokonała maszyna od pozycji najwyższej do punktu nad czujnikiem nie może być większa od całego zakresu ruchu w osi Z. Przykład:

Moja maszyna ma zakres ruchu w osi Z równy 90mm.

Najwyższa pozycja w osi Z we współrzędnych maszynowych wynosi 0.

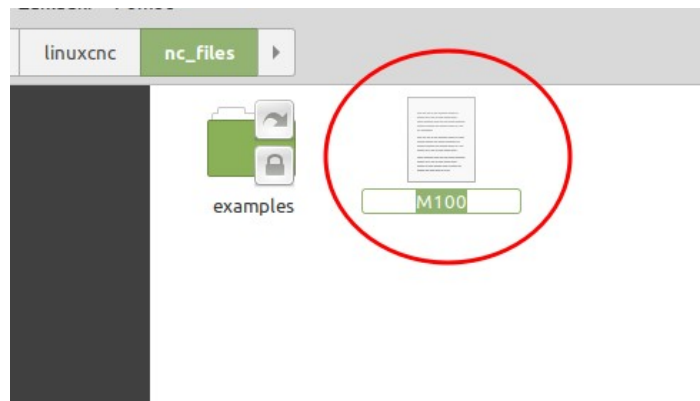
Najniższa pozycja w osi Z we współrzędnych maszynowych wynosi -90.

A więc, jeśli wartość w osi Z na której maszyna ma zdefiniowany czujnik jest równa -48, to znaczy że maszyna może się obniżyć w osi Z jeszcze jedynie o 32mm ponieważ taki ma zakres ruchu, dlatego u mnie MAX\_CHANGE = 32

Linia 75-77: Pozycja na jaką maszyna ustawi się po wymianie i zmierzeniu długości narzędzia.

#### 4. Przygotowanie wyświetlania komunikatów

Przechodzimy do katalogu **nc\_files** i tworzymy nowy plik (za pomocą edytora tekstu) i zapisujemy go pod nazwą **M100**



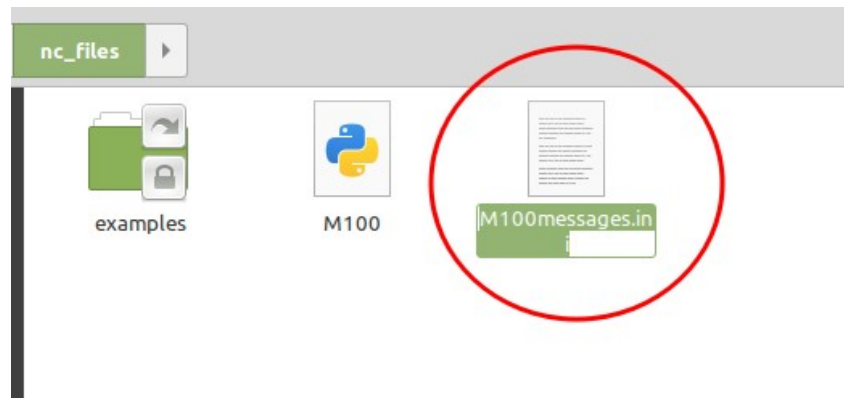
Otwieramy plik za pomocą edytora tekstu i wklejamy kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [----5----]

```

1  #!/usr/bin/python
2
3  title="Komunikat LinuxCNC"
4
5  fontsize = 15
6
7  import sys
8  import os
9  import ConfigParser
10 import Tkinter as tk
11
12 MessageFilePath = __file__ + "messages.ini"
13
14 proceed = "Y"
15
16 if not os.path.isfile(MessageFilePath):
17     instruction = MessageFilePath + " does not exist"
18     proceed = "N"
19 else:
20
21     if len(sys.argv) == 1:
22         instruction = "M100 was called with no P value"
23         proceed = "N"
24     else:
25         P=sys.argv[1]
26
27         if len(P)<7:
28             ps=P
29         else:
30             ps=P[0:len(P)-7]
31
32         cfg = ConfigParser.ConfigParser()
33
34         try:
35             cfg.read(MessageFilePath)
36             instruction = cfg.get("Messages", ps)
37             instruction = instruction.decode("unicode_escape")
38         except (ConfigParser.MissingSectionHeaderError), Argument:
39             proceed = "N"
40             instruction = "First line of\n" + MessageFilePath + "\nmust be:\n[Messages]"
41             pass
42         except (ConfigParser.MissingSectionHeaderError, ConfigParser.NoOptionError), Argument:
43             proceed = "N"
44             instruction = "There is no message for number\n" + P + "\n in the file\n" + MessageFilePath
45
46 if instruction == "":
47     instruction = "The message for P" + P + " is blank"
48     proceed = "N"
49
50 mess = os.linesep + instruction.replace("\n", os.linesep) + os.linesep
51
52 window = tk.Tk()
53
54 window.title(title)
55 window.geometry("300x200")
56
57 msg = tk.Message(window, text = mess,width=1000)
58 msg.config(fg="Black", font=("times", fontsize, "bold"))
59
60 def close(event):
61     exit(1)
62 window.bind("<Escape>",close)
63
64 if proceed == "Y":
65
66     def go_on(event):
67         exit(0)
68         window.bind("<Return>",go_on)
69
70     def Continue_clicked():
71         exit(0)
72
73     btnContinue = tk.Button(window,text="Kontynuuj",command=Continue_clicked,width=60,height=1,font=("Helvetica", "20"))
74     btnContinue.config(bg = "gray")
75
76 def Stop_clicked():
77     exit(1)
78     msg.pack()
79
80 if proceed == "Y":
81     btnContinue.pack()
82
83 window.mainloop()

```

W tym samym katalogu tworzymy plik z komunikatami, zapisujemy go pod nazwą M100messages.ini

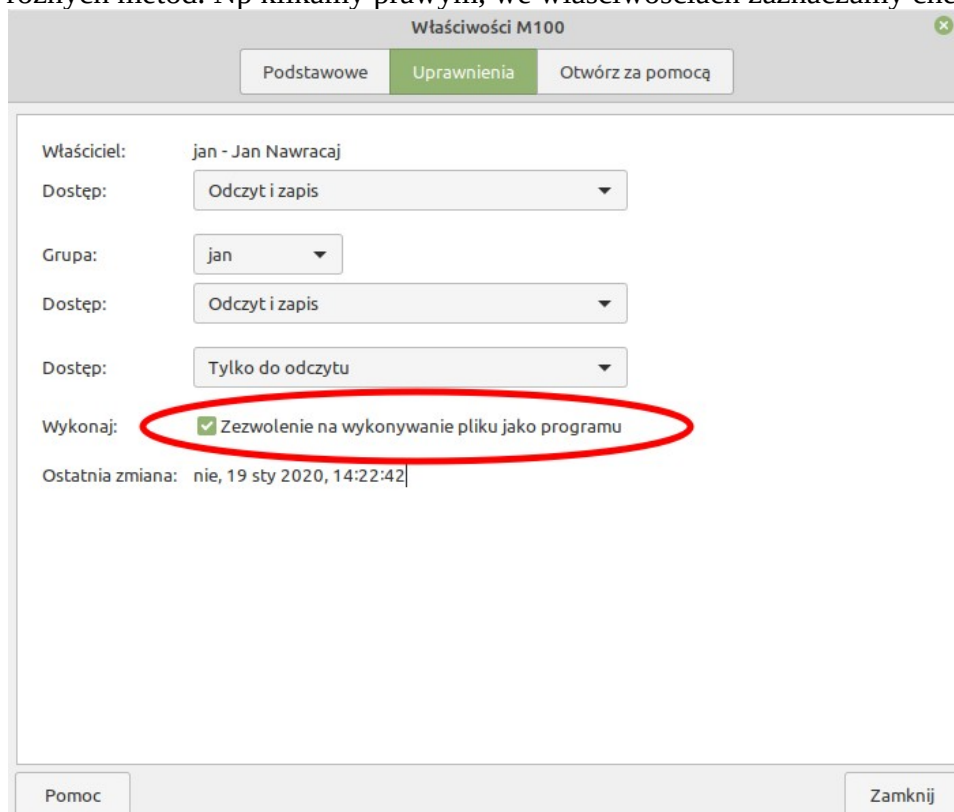


Otwieramy plik w edytorze tekstu i wklejamy komunikaty z pliku **LinuxCNC – CDN.txt** zawarte w sekcji [----6----]

```
M100messages.ini
1 [Messages]
2 # uzyj \n do zasygnalizowania nowej linii
3
4 0=Zamontuj nowe narzedzie \n Kontynuuj gdy gotowe
5
6 1=Pomiar dlugosci narzedzia \n Podlacz kabel czujnika\n Kontynuuj gdy gotowe
7
8 2=Zakonczone pomiar narzedzia \n Odlacz kabel czujnika \n kontynuuj gdy gotowe
9
```

Zapisujemy i zamykamy plik.

Nadajemy jeszcze prawa do uruchamiania pliku m100 jako aplikacji. W zależności od systemu, można użyć różnych metod. Np klikamy prawym, we właściwościach zaznaczamy checkbox:





## 5. Dodanie ścieżek do plików i remapowanie komend wymiany narzędzia

Aby LinuxCNC wiedział gdzie ma szukać naszych algorytmów należy go o tym poinformować. W tym celu otwieramy plik z rozszerzeniem .ini w edytorze tekstu.

Do sekcji [RS274NGC] dodajemy następujące linie z pliku **LinuxCNC – CDN.txt** zawarte w sekcji [----7----]

```
46 [RS274NGC]
47 PARAMETER FILE = linuxcnc.var
48 SUBROUTINE_PATH = ./algorytmy
49 FEATURES = 12
50 REMAP=M6 modalgroup=6 ngc=wymiana_narzedzia
51 REMAP=M06 modalgroup=6 ngc=wymiana_narzedzia
52
```

Jak widać remapujemy również komendę wymiany narzędzia M6 M06 tak aby po wywołaniu jej z dowolnego programu wywołała się nasza procedura wymiany zawarta w pliku **wymiana\_narzedzia.ngc**

## 6. Pozostała konfiguracja w pliku .ini

Aby włączyć interfejs HALUI w sekcji [HAL] dodajemy linię z pliku **LinuxCNC – CDN.txt** zawarte w sekcji [----8----]

```
59 [HAL]
60 HALUI = halui
61 HALFILE = moja-frezarka.hal
62 HALFILE = custom.hal
63 POSTGUI_HALFILE = postgui_call_list.hal
64
```

i dodajemy sekcję HALUI wklejając kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [----9----]

```
65 [HALUI]
66 # dodaj polecenia halui MDI (maks 64)
67 MDI_COMMAND = 0<wymiana_narzedzia> call
68
```

Zapisujemy i zamykamy plik .ini

## 7. Konfiguracja w pliku custom\_postgui.hal

Otwieramy w edytorze tekstu plik **custom\_postgui.hal** i dodajemy kod z pliku **LinuxCNC – CDN.txt** zawarty w sekcji [----10----]

```
custom_postgui.hal x
1 # Include your custom_postgui HAL commands here
2 # Ten plik nie zostanie nadpisany gdy uruchomisz ponownie Stepconf
3
4 net remote-wymiana_narzedzia halui.mdi-command-00 <= pyvcp.wymiana_narzedzia
5
```

Zapisujemy i zamykamy plik.

Po skonfigurowaniu obsługi wymiany narzędzia nie uruchamiamy kreatora konfiguracji stepconf ponieważ nadpisze nam edytowane pliki konfiguracyjne i ustawienia znikną.

Jeśli nie mieliśmy wcześniej skonfigurowanego wejścia sondy narzędzia, należy to teraz zrobić manualnie dodając w pliku **nazwa\_konfiguracji.hal** linię z pliku **LinuxCNC – CDN.txt** zawartą w sekcji [----11----]

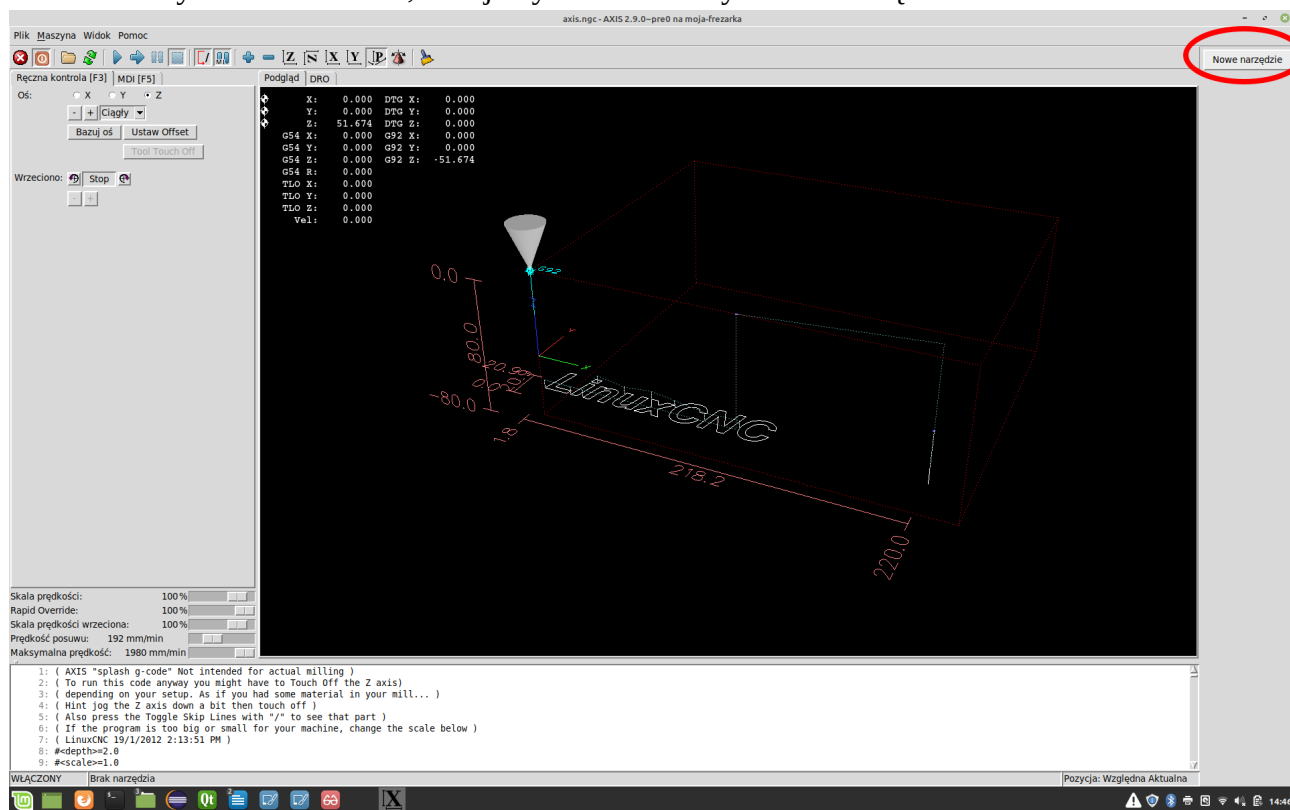
```
54 net home-x          <= parport.0.pin-10-in
55 net home-y          <= parport.0.pin-11-in
56 net home-z          <= parport.0.pin-12-in
57 net home-a          <= parport.0.pin-13-in
58 net probe-in        <= parport.0.pin-15-in
59
```

15 to numer wejścia na porcie LPT. Wpisujemy tam własny numer. Jeśli chcemy zanegować wejście to dodajemy końcówkę not np.:

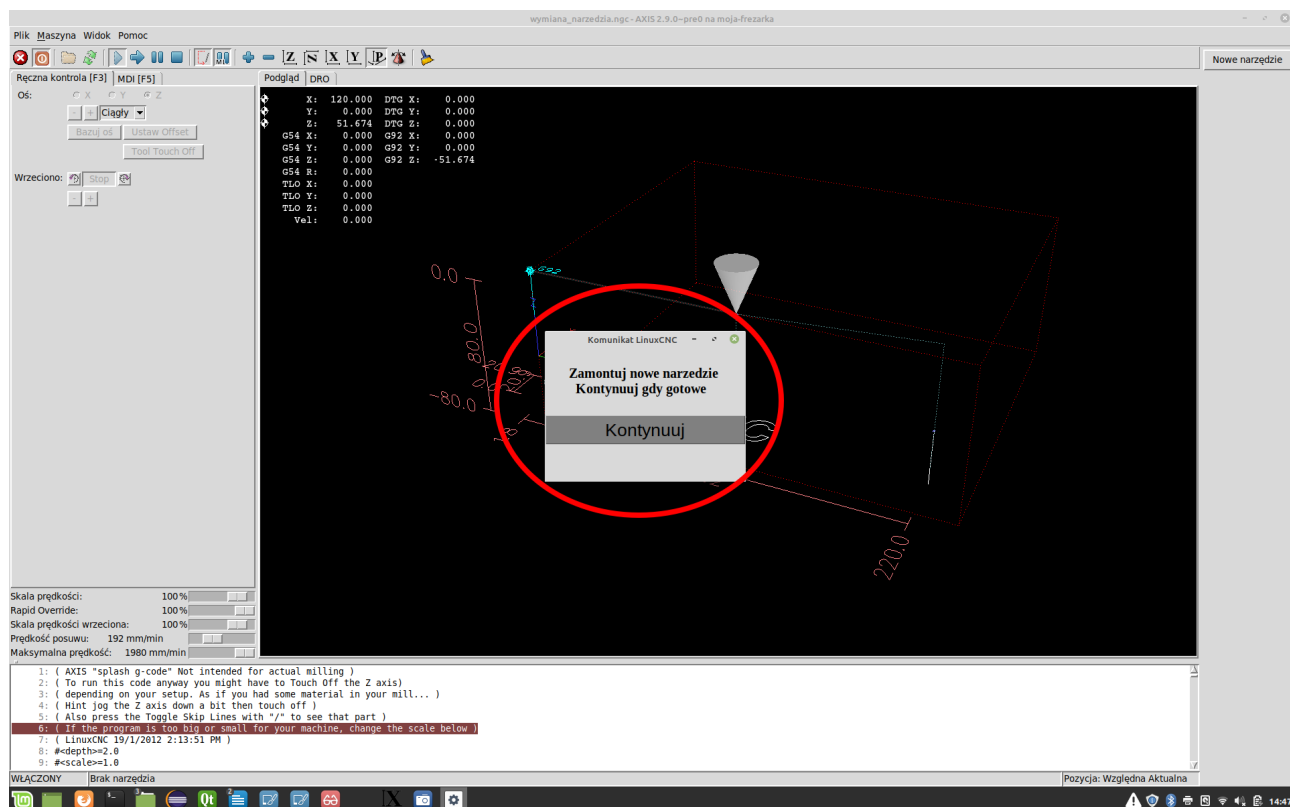
```
net probe-in    <= parport.0.pin-15-in-not
```

## 8. Jak korzystać z procedury wymiany i pomiaru długości narzędzia

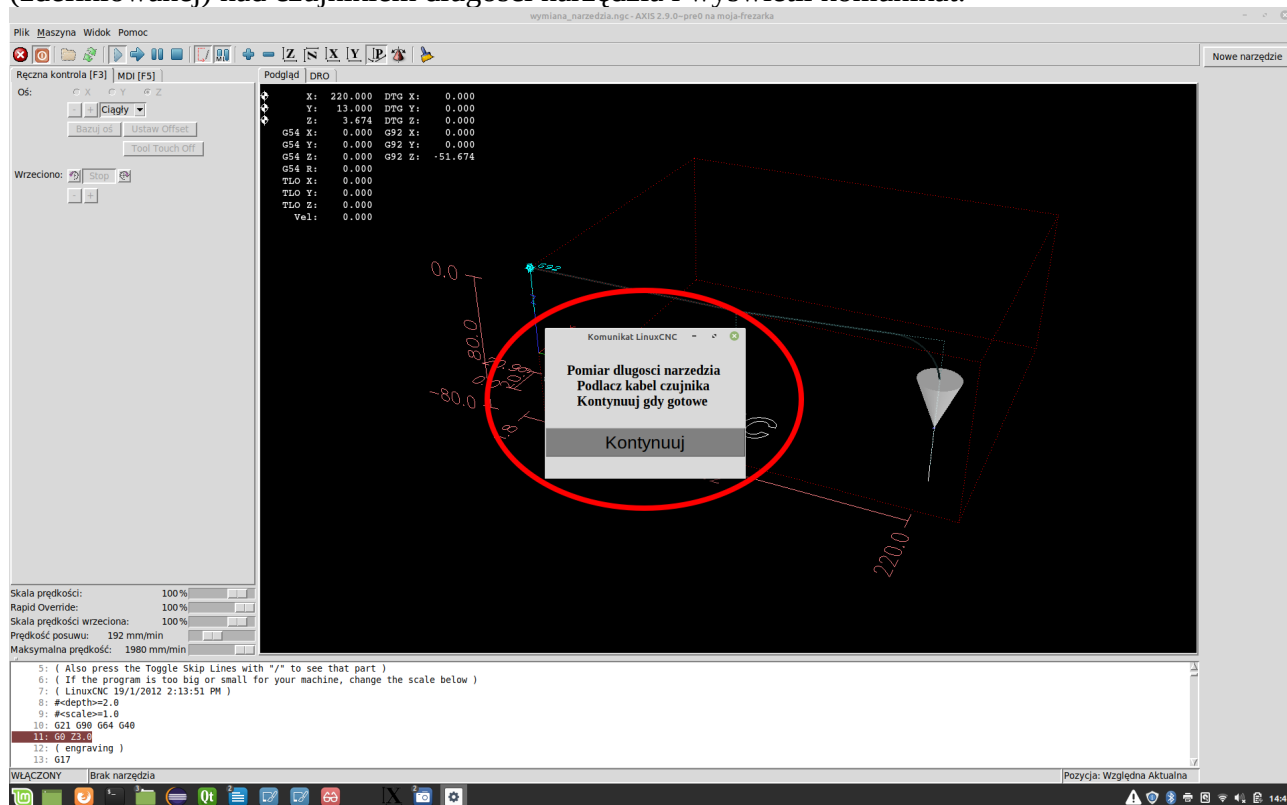
Uruchamiamy nasz LinuxCNC, bazujemy osie i klikamy Nowe Narzędzie



Maszyna ustawia się w pozycji jaką zdefiniowaliśmy jako pozycja wymiany narzędzia i wyświetla komunikat:



Montujemy pierwsze narzędzie i klikamy **Kontynuuj**. Maszyna ustawi się w pozycji (zdefiniowanej) nad czujnikiem długości narzędzia i wyświetli komunikat:

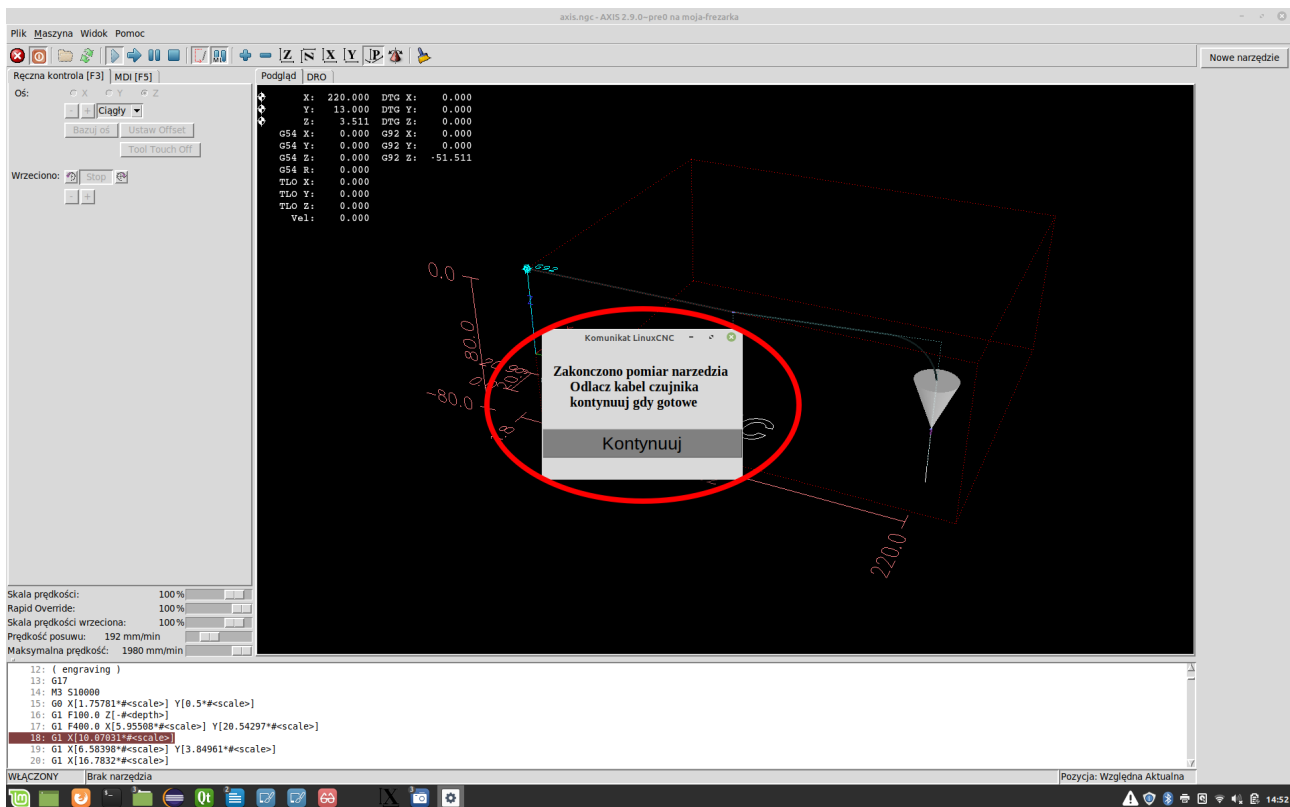


Jeśli czujnik nie jest montowany na stałe to teraz należy go ustawić w miejscu pomiaru. Jeśli posiadamy czujnik z koniecznością podłączenia przewodu to podłączamy go do narzędzia



Jeśli czujnik jest gotowy to klikamy **Kontynuuj**.

Maszyna dokona teraz pomiaru i z powrotem ustawi się w pozycji nad czujnikiem i wyświetli komunikat:



Odłączamy i / lub zabezpieczamy czujnik jeśli to konieczne i klikamy **Kontynuuj**. Maszyna ustawi się w zdefiniowanej pozycji po wymianie narzędzia i jest gotowa do pracy.

Jeśli mamy już zmierzone narzędzie to możemy ustawić offsety do materiału. W przypadku gdy zajdzie potrzeba wymiany narzędzia klikamy przycisk nowe narzędzie i po całej procedurze można pracować dalej. (offsety ustawiamy do materiału tylko po zamontowaniu pierwszego narzędzia, po wymianie narzędzia offsetów już nie ruszamy).

## 9. Jak korzystać z wymiany i pomiaru narzędzia z własnego programu

W pliku konfiguracyjnym zrobiliśmy już remapowanie komend M6 / M06 a więc aby z poziomu programu wywołać tą procedurę wystarczy wpisać kod wymiany narzędzia, np. T1M06. Należy pamiętać, że wypadałoby zatrzymać wrzeciono przed wywołaniem wymiany narzędzia i uruchomić je po zakończeniu procedury. Przykładowy kod wymiany narzędzia w naszym programie może wyglądać tak:

```
M05
G04 P10
(MSG, Zamontuj narzędzie <t>)
T1M06
M03 S<s>
G04 P20
```

Gdzie <t> to numer narzędzia, <s> to obroty wrzeciona po wymianie narzędzia. Ja mam ustawione w postprocesorze że każda wymiana narzędzia wywołują taki właśnie kod.