

MachMaker
www.machmaker.pl

INSTRUKCJA INSTALACJI ZADAJNIKA XHC-WHB04B W LINUXCNC NA SYSTEMIE DEBIAN 7 WHEEZY



Do instalacji Zadajnika WHB04B potrzebujemy komputer z zainstalowanym i działającym LinuxCNC na systemie Debian7 i ze stabilnym połączeniem z internetem.

Z urządzeniem firma MachMaker dostarcza spakowany katalog xhc-whb04b-6 DEB7 LCNC2_9.zip. Należy go rozpakować w wygodnym miejscu (np. na pulpicie).

Otwieramy terminal i wpisujemy

```
sudo nano /etc/apt/sources.list
```

Usuwamy zawartość otwartego pliku i wklejamy

```
deb http://archive.debian.org/debian wheezy main contrib non-free
deb-src http://archive.debian.org/debian wheezy main contrib non-free
# deb http://archive.debian.org/debian wheezy/updates main contrib non-free
# deb-src http://archive.debian.org/debian wheezy/updates main contrib non-free
deb http://archive.debian.org/debian-security wheezy/updates main contrib non-free
deb-src http://archive.debian.org/debian-security/ wheezy/updates main contrib non-free
```

Zapisujemy (Ctrl + o i enter) i zamykamy (Ctrl + x)

Następnie wpisujemy

```
sudo apt-get clean
```

```
sudo apt-get install debian-archive-keyring
```

```
sudo apt-get update -o Acquire::Check-Valid-Until=false
```

```
sudo apt-get install dirmngr software-properties-common gnome-commander git usbutils
```

Podłączamy odbiornik USB naszego zadajnika do gniazda USB komputera. W terminalu wpisujemy następujące polecenie:

```
lsusb
```

```
jannawracaj@jannawracaj:~$ lsusb
Bus 003 Device 003: ID 090c:1000 Silicon Motion, Inc. - Taiwan (formerly Feiya Technology Corp.) Flash Drive
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 002: ID 10ce:eb93 Silicon Labs
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 002: ID 89e5:101b
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 003: ID 046d:09b2 Logitech, Inc. Fujitsu Webcam
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 002: ID 1199:6832 Sierra Wireless, Inc. MC8780 Device
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 002: ID 0c24:000f Taiyo Yuden Bluetooth Device (V2.0+EDR)
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
jannawracaj@jannawracaj:~$
```

Wyświetli się lista urządzeń a między innymi nasze urządzenie (zaznaczone w czerwonej ramce). Jeśli urządzenie się nie wyświetla, to należy sprawdzić czy jest poprawnie podłączone lub ewentualnie uruchomić ponownie komputer i powrócić do polecenia lsusb)

W ramce żółtej znajduje się adres naszego urządzenia. W tym przypadku jest to adres 10ce:eb93. Jeśli wyświetli się taki właśnie adres, to nic nie trzeba zmieniać ponieważ taki zapisany jest w pliku konfiguracyjnym. Jeśli natomiast znajduje się tam inny adres, to należy wejść do dostarczonego katalogu xhc-whb04b-6 DEB9 LCNC2_9 i w edytorze tekstu otworzyć plik 99-xhc-whb04b-6.rules.

Należy zmienić wartości odpowiedzialne za adres i zapisać plik:

```
ATTR{idProduct}=="eb93", ATTR{idVendor}=="10ce", MODE="0666", OWNER="root",  
GROUP="plugdev"
```

Ponieważ na LinuxCNC 2.7 występują problemy z załadowaniem modułu zadajnika XHC, czeka nas teraz instalacja wersji 2.9.0 LinuxCNC. Wcześniej jeszcze zapisujemy np. w formie archiwum aktualną konfigurację (ustawienia) dla swojej maszyny. Następnie otwieramy nowy terminal i wpisujemy:

```
sudo apt-key adv --keyserver hkp://keys.gnupg.net --recv-key EF1B07FEE0EE663E  
sudo nano /etc/apt/sources.list.d/linuxcnc-buildbot.list
```

i wklejamy:

```
deb http://buildbot.linuxcnc.org/wheezy master-rt  
deb-src http://buildbot.linuxcnc.org/wheezy master-rt
```

Zapisujemy (Ctrl + o i enter) i zamykamy (Ctrl + x)

```
sudo apt-get update -o Acquire::Check-Valid-Until=false
```

```
sudo apt-get install linuxcnc
```

Uruchom LinuxCNC ze swoją działającą konfiguracją.

UWAGA!

Jeśli uruchomimy wbudowany w LinuxCNC kreator np. Stepconf Wizard to nadpisze on plik .ini i trzeba go zaktualizować na nowo.

Uruchamiamy menadżer plików z uprawnieniami administratora wpisując w terminal:

```
sudo gnome-commander
```

Kopiujemy dostarczony plik 99-xhc-whb04b-6.rules do katalogu /etc/udev/rules.d/

Zamykamy gnome-commander

Następnie w terminalu uruchom polecenie

```
sudo udevadm trigger
```

Zamykamy terminal

Uruchamiamy nowy terminal w katalogu użytkownika i wpisujemy:

`sudo git clone https://github.com/LinuxCNC/linuxcnc linuxcnc-dev`

W katalogu użytkownika powstanie nowy katalog `linuxcnc-dev`

Uruchamiamy menadżer plików z uprawnieniami administratora wpisując w terminal:

`sudo gnome-commander`

Kopiujemy dostarczony katalog `hc-whb04b-6` do katalogu `linuxcnc-dev/src/hal/user_comps/`

Przez `gnome-commander` wchodzimy do katalogu `linuxcnc-dev/src/`

W edytorze tekstu otwieramy plik `Makefile`

Odszukujemy sekcję `SUBDIRS` i wklejamy ścieżkę do naszego `hal`

`hal/user_comps/xhc-whb04b-6 \`

```
124 SUBDIRS := \  
125     libnml/linklist libnml/cms libnml/rcs libnml/inifile libnml/os_intf \  
126     libnml/nml libnml/buffer libnml/posemath libnml \  
127     \  
128     rtapi/examples/timer rtapi/examples/semaphore rtapi/examples/shmem \  
129     rtapi/examples/extint rtapi/examples/fifo rtapi/examples rtapi \  
130     \  
131     hal/components hal/drivers hal/drivers/mesa-hostmot2 \  
132     hal/user_comps/devices hal/user_comps/mb2hal \  
133     hal/user_comps hal/user_comps/vismach hal/user_comps/vfs11_vfd hal/classicladder hal/utils \  
134     hal/user_comps/vfdb_vfd hal/user_comps/wj200_vfd \  
135     hal/user_comps/huanyang-vfd \  
136     hal/user_comps/xhc-whb04b-6 \  
137     \  
138     emc/usr_intf/axis emc/usr_intf/touchy emc/usr_intf/stepconf emc/usr_intf/pncconf \  
139     emc/usr_intf/gremlin emc/usr_intf/gscreen emc/usr_intf/pyui emc/usr_intf/qtvcpl \  
140     emc/usr_intf/gmoccapy \  
141     emc/usr_intf emc/nml_intf emc/task emc/iotask emc/kinematics emc/tp emc/cantermpl \  
142     emc/motion emc/ini emc/rs274ngc emc/sai emc emc/pythonplugin \  
143     emc/motion-logger \  

```

Zapisujemy i zamykamy plik `Makefile`.

Zamykamy `gnome-commander`

Otwieramy nowy terminal i wklejamy kolejno polecenia, zatwierdzamy enterem, potwierdzamy jeśli konieczne.

`sudo apt-get install autoconf automake pkg-config libgtk-3-dev autoconf autogen libmodbus-dev libusb-1.0-0-dev`

`sudo apt-get update -o Acquire::Check-Valid-Until=false`

`sudo apt-get install libboost-python-dev netcat libmodbus-dev yapps2 libudev-dev tcl8.5-dev tk8.5-dev`

`libreadline-gplv2-dev asciidoc dblatex docbook-xsl dvipng graphviz groff imagemagick inkscape`

`python-lxml source-highlight texlive-extra-utils texlive-font-utils texlive-fonts-recommended texlive-`

`lang-cyrillic texlive-lang-french texlive-lang-german texlive-lang-polish texlive-lang-spanish texlive-`

`latex-recommended w3c-linkchecker xsltproc python-dev libxmu-dev libglu1-mesa-dev libgl1-mesa-dev`

`libgtk2.0-dev intltool python3-tk libusb-1.0-0-dev`

`sudo apt-get install linux-headers-$(uname -r)`

`cd linuxcnc-dev/src`

`sudo ./autogen.sh`

`sudo ./configure`

`sudo make`

`sudo make setuid`

Uruchamiamy menadżer plików z uprawnieniami administratora wpisując w terminal:

```
sudo gnome-commander
```

Kopiujemy wygenerowany plik bin `xhc-whb04b-6` z katalogu `/linuxcnc-dev/bin/` do katalogu `/usr/bin/`

zamykamy `gnome-commander`

Zamykamy wszystko i otwieramy nowy terminal

W terminalu wpisujemy:

```
sudo udevadm trigger
```

Może to spowodować chwilowe przyblokowanie systemu, należy cierpliwie poczekać.

Następnie wklejamy w terminal:

```
halrun
```

Jeśli `halrun` będzie działać, wklejamy następnie

```
loadusr xhc-whb04b-6
```

```
jannawracaj@jannawracaj:~$ sudo udevadm trigger
[sudo] hasło użytkownika jannawracaj:
jannawracaj@jannawracaj:~$ halrun
halcmd: loadusr xhc-whb04b-6
```

i klikamy Enter.

Uruchamiamy nasz zadajnik i kręcimy kółkiem lub wciskamy przyciski, powinny się pojawiać w terminalu sygnały od zadajnika.

```
ode=0x00}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button pressed event metaButton={key={code=0x0b text='s-on-off' altText=
macro-8'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button released event metaButton={key={code=0x0b text='s-on-off' altText=
macro-8'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button pressed event metaButton={key={code=0x09 text='safe-z' altText='m
cro-6'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button released event metaButton={key={code=0x09 text='safe-z' altText='m
cro-6'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button pressed event metaButton={key={code=0x03 text='start-pause' altTe
t='macro-13'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
pndnt button released event metaButton={key={code=0x03 text='start-pause' altTe
t='macro-13'} modifier={code=0x00 text='' altText=''}}
pndnt handwheel total counts {counts=0 activeCounter=0 isLeadActive=0}
```

Naciskamy `ctrl + c` i zamykamy terminal.

Kopiuujemy dostarczony plik *xhc-whb04b-6.hal* do swojego katalogu gdzie mamy ustawienia frezarki (tam gdzie między innymi znajduje się plik *.ini*)

Otwieramy w edytorze tekstu swój plik *.ini* i do sekcji [HAL] wklejamy

HALFILE = xhc-whb04b-6.hal

oraz

HALUI = halui

```
58 [HAL]
59 HALUI = halui
60 HALFILE = moja-frezarka.hal
61 HALFILE = custom.hal
62 HALFILE = xhc-whb04b-6.hal
63 POSTGUI_HALFILE = custom_postgui.hal
```

Do sekcji [HALUI] wklejamy

```
MDI_COMMAND=(debug,00)
MDI_COMMAND=(debug,macro1)
MDI_COMMAND=(debug,macro2)
MDI_COMMAND=(debug,macro3)
MDI_COMMAND=(debug,macro4)
MDI_COMMAND=G1 G53 X0 Y0 Z0 F4000
MDI_COMMAND=(debug,macro6)
MDI_COMMAND=G1 X0 Y0 Z0 F4000
MDI_COMMAND=(debug,macro8)
MDI_COMMAND=(debug,macro9)
MDI_COMMAND=(debug,macro10)
MDI_COMMAND=(debug,macro11)
MDI_COMMAND=(debug,macro12)
MDI_COMMAND=(debug,macro13)
MDI_COMMAND=(debug,macro14)
MDI_COMMAND=(debug,macro15)
MDI_COMMAND=(debug,macro16)
MDI_COMMAND=G1 G53 Z0 F4000
MDI_COMMAND=(debug,macro17)
MDI_COMMAND=(debug,macro18)
MDI_COMMAND=(debug,macro19)
```



```

66 [HALUI]
67 MDI_COMMAND=(debug,00)
68 MDI_COMMAND=(debug,macro1)
69 MDI_COMMAND=(debug,macro2)
70 MDI_COMMAND=(debug,macro3)
71 MDI_COMMAND=(debug,macro4)
72 MDI_COMMAND=G1 G53 X0 Y0 Z0 F4000
73 MDI_COMMAND=(debug,macro6)
74 MDI_COMMAND=G1 X0 Y0 Z0 F4000
75 MDI_COMMAND=(debug,macro8)
76 MDI_COMMAND=(debug,macro9)
77 MDI_COMMAND=(debug,macro10)
78 MDI_COMMAND=(debug,macro11)
79 MDI_COMMAND=(debug,macro12)
80 MDI_COMMAND=(debug,macro13)
81 MDI_COMMAND=(debug,macro14)
82 MDI_COMMAND=(debug,macro15)
83 MDI_COMMAND=(debug,macro16)
84 MDI_COMMAND=G1 G53 Z0 F4000
85 MDI_COMMAND=(debug,macro17)
86 MDI_COMMAND=(debug,macro18)
87 MDI_COMMAND=(debug,macro19)

```

Następnie należy zaktualizować sekcję [DISPLAY]

Jeśli nie wiesz co ustawić, na początek po prostu zamieść to co masz w sekcji DISPLAY na następujący kod:

[DISPLAY]

DISPLAY = axis

POSITION_OFFSET = RELATIVE

POSITION_FEEDBACK = ACTUAL

MIN_FEED_OVERRIDE = 0.000000

MAX_FEED_OVERRIDE = 1.200000

MAX_SPINDLE_OVERRIDE = 1.5

MIN_SPINDLE_OVERRIDE = 0.1

INTRO_GRAPHIC = linuxcnc.gif

INTRO_TIME = 5

PROGRAM_PREFIX = /home/cnc/linuxcnc/nc_files

INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm

POSITION_FEEDBACK = ACTUAL

DEFAULT_LINEAR_VELOCITY = 6.000000

MAX_LINEAR_VELOCITY = 83.3333

MIN_LINEAR_VELOCITY = 0.500000

DEFAULT_ANGULAR_VELOCITY = 12.000000

MAX_ANGULAR_VELOCITY = 180.000000

MIN_ANGULAR_VELOCITY = 1.666667

EDITOR = gedit

GEOMETRY = xyz

Sprawdzamy jeszcze, czy mamy ustawioną sekwencję bazowania. W pliku ini definiuje się te w dodając parametr HOME_SEQUENCE dla każdej osi. Jeśli w ustawieniach twojej maszyny jest już takie ustawienie to nie musisz nic robić. Jeśli natomiast nie masz ustawionych sekwencji bazowania wklej następującą wartość dla każdej z osi:

HOME_SEQUENCE = 0

```
[JOINT_0]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 25.0
MAX_ACCELERATION = 750.0
STEPGEN_MAXACCEL = 937.5
SCALE = 80.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 200.0
HOME_OFFSET = 0.0
HOME_SEQUENCE = 0
```

```
[AXIS_Y]
MIN_LIMIT = -0.001
MAX_LIMIT = 200.0
MAX_VELOCITY = 25.0
MAX_ACCELERATION = 750.0
```

```
[JOINT_1]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 25.0
MAX_ACCELERATION = 750.0
STEPGEN_MAXACCEL = 937.5
SCALE = 80.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 200.0
HOME_OFFSET = 0.0
HOME_SEQUENCE = 0
```

```
[AXIS_Z]
MIN_LIMIT = -100.0
MAX_LIMIT = 0.001
MAX_VELOCITY = 25.0
MAX_ACCELERATION = 750.0
```

```
[JOINT_2]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 25.0
MAX_ACCELERATION = 750.0
STEPGEN_MAXACCEL = 937.5
SCALE = 80.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -100.0
MAX_LIMIT = 0.001
HOME_OFFSET = 0.0
HOME_SEQUENCE = 0
```

Zapisz i zamknij plik .ini.

Uruchom LinuxCNC, zadajnik powinien już działać prawidłowo.