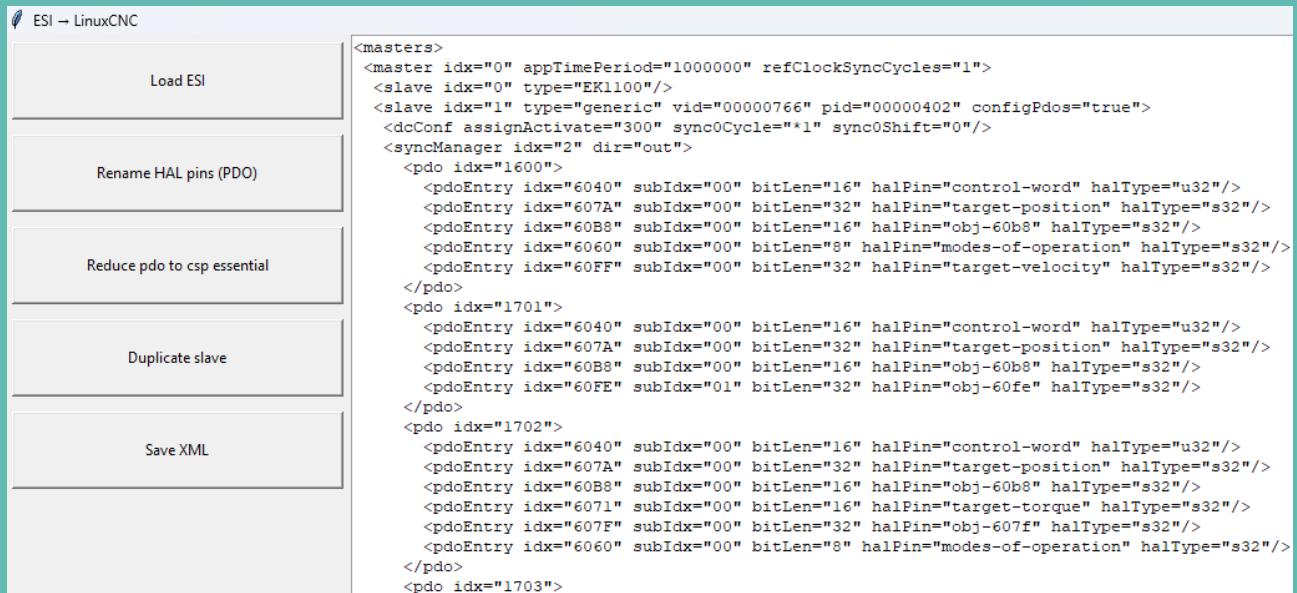


|   |            |
|---|------------|
| <b>1. XML Generator</b>                 | <b>p.2</b> |
| <b>2. HAL Generator</b>                 | <b>p4</b>  |
| <b>3. INI Generator</b>                 | <b>p.6</b> |
| <b>4. DosStyle → UTF-8 LF Conventer</b> | <b>p.8</b> |

To use the following programs on Windows/Linux, Python 3.14 with tkinter is required.

## 1. XML Generator

### 1.1 Load the .xml ESI file from the manufacturer.

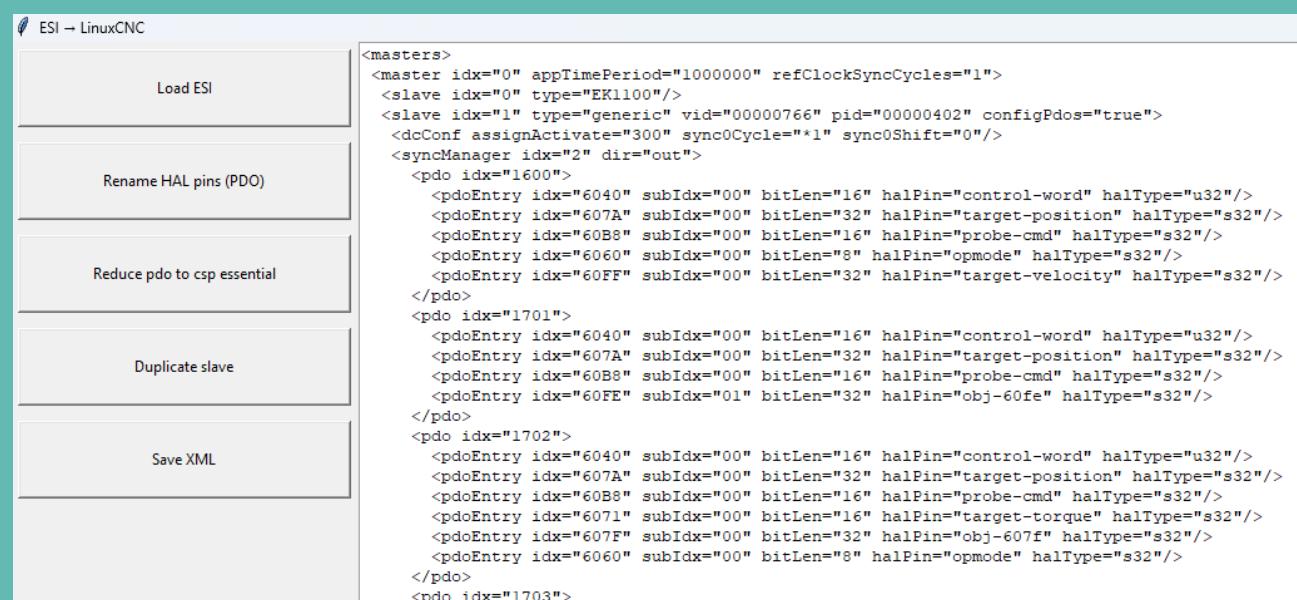


The screenshot shows the ESI → LinuxCNC software interface. On the left, there is a vertical list of five buttons: "Load ESI", "Rename HAL pins (PDO)", "Reduce pdo to csp essential", "Duplicate slave", and "Save XML". The "Load ESI" button is highlighted with a blue border. To the right of the buttons, the XML code is displayed. The XML starts with a <masters> tag and includes several <slave> and <pdo> tags, defining pin configurations like control-word, target-position, probe-cmd, opmode, and target-velocity.

```
<masters>
  <master idx="0" appTimePeriod="1000000" refClockSyncCycles="1">
    <slave idx="0" type="EK1100"/>
    <slave idx="1" type="generic" vid="00000766" pid="00000402" configPdos="true">
      <dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
      <syncManager idx="2" dir="out">
        <pdo idx="1600">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="obj-60b8" halType="s32"/>
          <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="modes-of-operation" halType="s32"/>
          <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>
        </pdo>
        <pdo idx="1701">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="obj-60b8" halType="s32"/>
          <pdoEntry idx="60FE" subIdx="01" bitLen="32" halPin="obj-60fe" halType="s32"/>
        </pdo>
        <pdo idx="1702">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="obj-60b8" halType="s32"/>
          <pdoEntry idx="6071" subIdx="00" bitLen="16" halPin="target-torque" halType="s32"/>
          <pdoEntry idx="607E" subIdx="00" bitLen="32" halPin="obj-607f" halType="s32"/>
          <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="modes-of-operation" halType="s32"/>
        </pdo>
      </syncManager>
    </slave>
  </master>
</masters>
```

### 1.2 Replace the names of the HAL pins (PDO).

It replaces the name modes of operation → opmode and a few others; all pin names can be found in the servodriver manual.

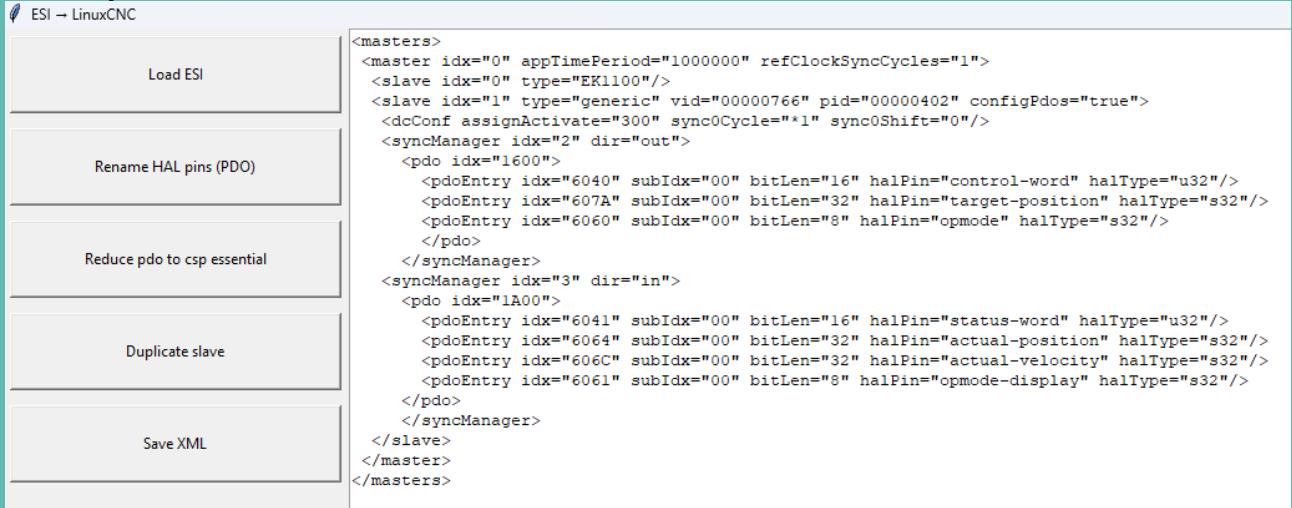


The screenshot shows the ESI → LinuxCNC software interface. The buttons on the left are the same as in the previous screenshot. The XML code on the right has been modified. The <pdo> tag for index 1600 now includes a <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/> entry. The <pdo> tag for index 1701 now includes entries for <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>, <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>, <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="probe-cmd" halType="s32"/>, <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>, and <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>. The <pdo> tag for index 1702 now includes entries for <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>, <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>, <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="probe-cmd" halType="s32"/>, <pdoEntry idx="6071" subIdx="00" bitLen="16" halPin="target-torque" halType="s32"/>, <pdoEntry idx="607F" subIdx="00" bitLen="32" halPin="obj-607f" halType="s32"/>, and <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>. The <pdo> tag for index 1703 remains the same as in the first screenshot.

```
<masters>
  <master idx="0" appTimePeriod="1000000" refClockSyncCycles="1">
    <slave idx="0" type="EK1100"/>
    <slave idx="1" type="generic" vid="00000766" pid="00000402" configPdos="true">
      <dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
      <syncManager idx="2" dir="out">
        <pdo idx="1600">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="probe-cmd" halType="s32"/>
          <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
          <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>
        </pdo>
        <pdo idx="1701">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="probe-cmd" halType="s32"/>
          <pdoEntry idx="60FE" subIdx="01" bitLen="32" halPin="obj-60fe" halType="s32"/>
        </pdo>
        <pdo idx="1702">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60B8" subIdx="00" bitLen="16" halPin="probe-cmd" halType="s32"/>
          <pdoEntry idx="6071" subIdx="00" bitLen="16" halPin="target-torque" halType="s32"/>
          <pdoEntry idx="607F" subIdx="00" bitLen="32" halPin="obj-607f" halType="s32"/>
          <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
        </pdo>
      </syncManager>
    </slave>
  </master>
</masters>
```

### 1.3 Reduce PDOs to CSP essential.

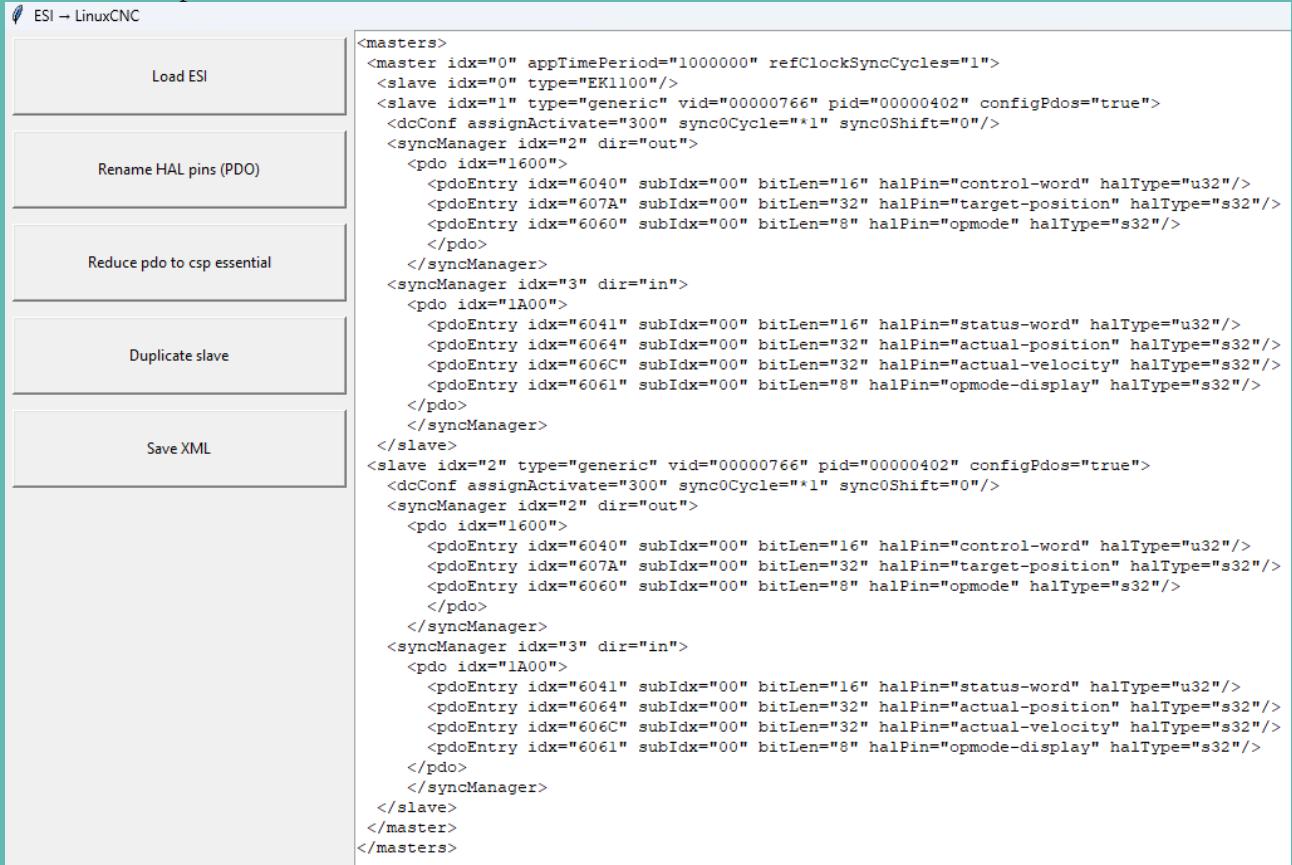
Keeps the first output group and the first input PDO, keeps essential PDOs for CSP mode; if you want to use CSV mode, target velocity is kept instead of target position. The text can be edited manually.



```
<master idx="0" appTimePeriod="1000000" refClockSyncCycles="1">
<slave idx="0" type="EK1100"/>
<slave idx="1" type="generic" vid="00000766" pid="00000402" configPdos="true">
<dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
<syncManager idx="2" dir="out">
<pdo idx="1600">
<pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
<pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
<pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
</pdo>
</syncManager>
<syncManager idx="3" dir="in">
<pdo idx="1A00">
<pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="status-word" halType="u32"/>
<pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
<pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
<pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
</pdo>
</syncManager>
</slave>
</master>
</masters>
```

### 1.4 Duplicate the slave.

Each click duplicates the text </slave... </slave> and increments the slave index in numerical order.



```
<master idx="0" appTimePeriod="1000000" refClockSyncCycles="1">
<slave idx="0" type="EK1100"/>
<slave idx="1" type="generic" vid="00000766" pid="00000402" configPdos="true">
<dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
<syncManager idx="2" dir="out">
<pdo idx="1600">
<pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
<pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
<pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
</pdo>
</syncManager>
<syncManager idx="3" dir="in">
<pdo idx="1A00">
<pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="status-word" halType="u32"/>
<pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
<pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
<pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
</pdo>
</syncManager>
</slave>
<slave idx="2" type="generic" vid="00000766" pid="00000402" configPdos="true">
<dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
<syncManager idx="2" dir="out">
<pdo idx="1600">
<pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="control-word" halType="u32"/>
<pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
<pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
</pdo>
</syncManager>
<syncManager idx="3" dir="in">
<pdo idx="1A00">
<pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="status-word" halType="u32"/>
<pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
<pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
<pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
</pdo>
</syncManager>
</slave>
</master>
</masters>
```

### 1.5 Save Xml

|               |   |
|---------------|---|
| ethercat-conf | ▼ |
| XML           | ▼ |

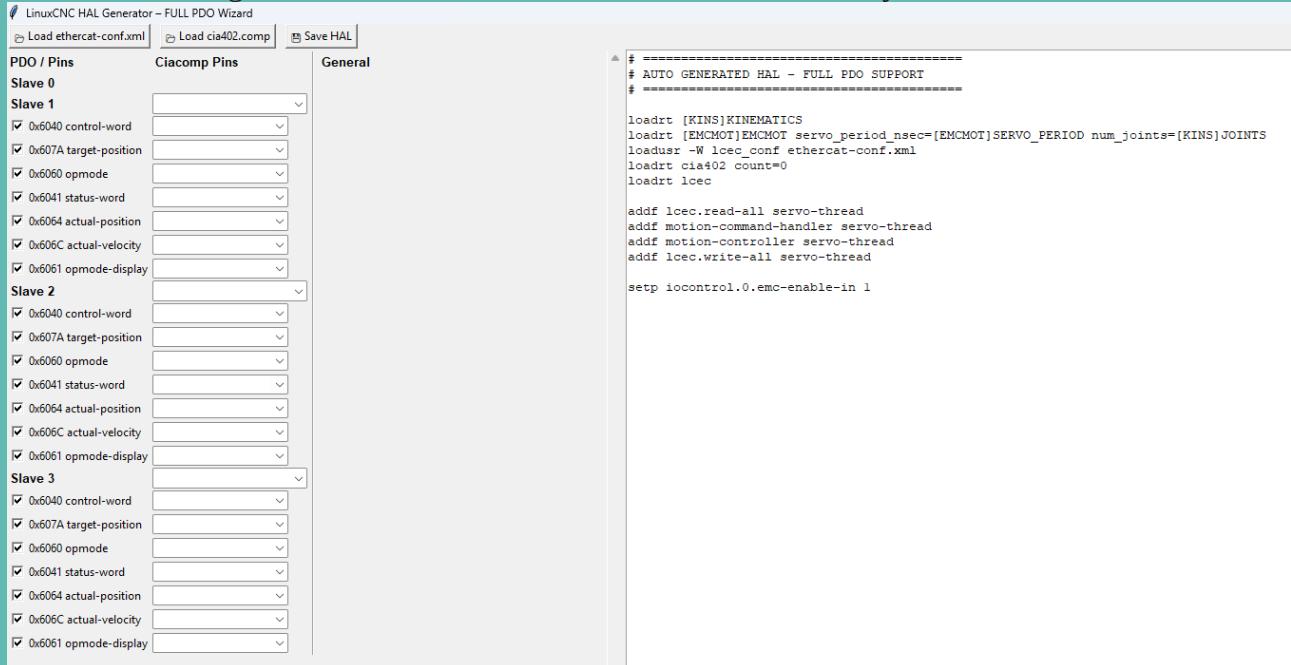
## 2. HAL Generator

### 2.1 Load ethercat-conf.xml

After loading ethercat-conf.xml, all PDOs available in the selected file will appear, and in the HAL generation window on the right side, the configuration initialization section will appear.

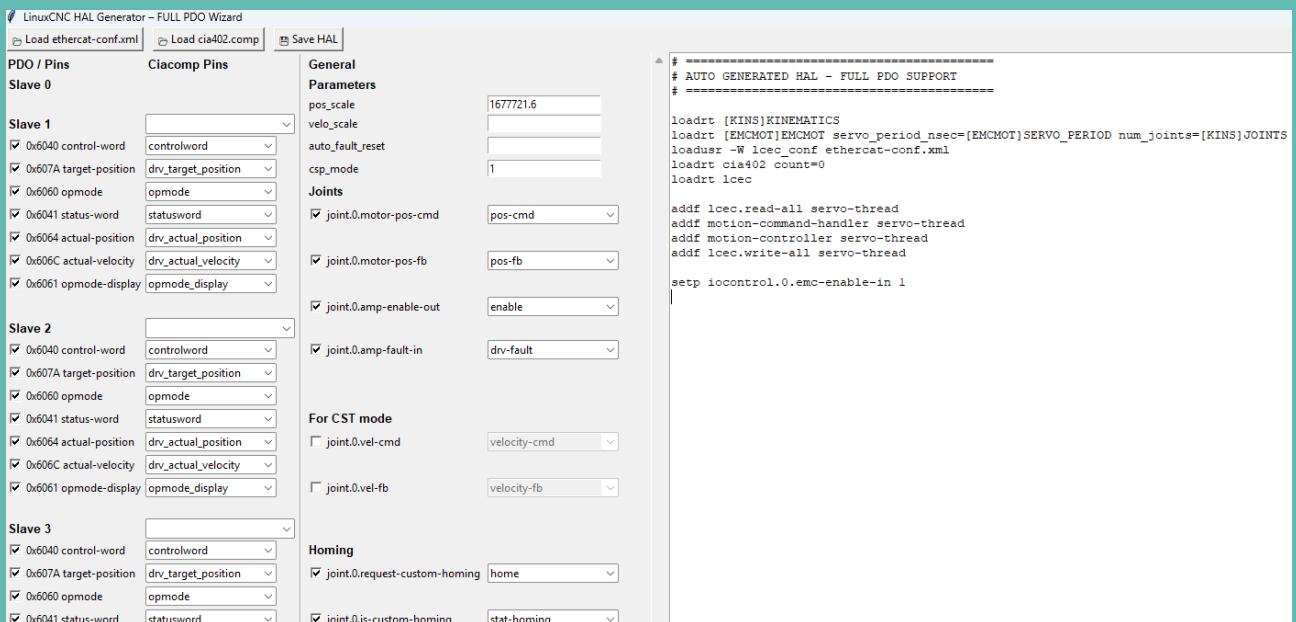
LinuxCNC executes the configuration line by line, initializing modules, setting parameters, and preparing the system for motion control.

Among others, in this line: "loadusr -W lcec\_conf ethercat-conf.xml", the PDOs from the file with the prefix lcec will be loaded into the HAL module running in LinuxCNC's internal memory in real time, where all signal connections between devices and the control system are created.



### 2.2. load cia402.comp

The .comp file is parsed into pins and parameters; most of the suggested pins are automatically matched. The pos\_scale value should be selected based on the encoder and the stroke per revolution



## 2.3. Axis selection

During selection, a specific machine axis (X, Y, Y2, Z, etc.) is assigned to the selected EtherCAT slave, and parameters common to all axes are set. Based on this, the program automatically connects the selected axis to the appropriate control input.

The HAL file preview is updated in real time, making it easy to understand which value is responsible for what and how the individual settings are related to each other while changing parameters.

The screenshot shows the LinuxCNC HAL Generator interface. On the left, there are three sections for Slave 1 (X), Slave 2 (Y), and Slave 3 (Z). Each section contains dropdown menus for various EtherCAT parameters like control-word, target-position, and actual-position. To the right of these sections is a 'General Parameters' group and a 'Joints' group. The 'General Parameters' group includes fields for pos\_scale (1677721.6), velo\_scale, auto\_fault\_reset, and csp\_mode (1). The 'Joints' group includes dropdowns for joint.0.motor-pos-cmd (pos-cmd), joint.0.motor-pos-fb (pos-fb), joint.0.amp-enable-out (enable), and joint.0.amp-fault-in (drv-fault). Below these groups is a 'For CST mode' section with dropdowns for joint.0.vel-cmd (velocity-cmd) and joint.0.vel-fb (velocity-fb). At the bottom is a 'Homing' section with dropdowns for joint.0.request-custom-homing (home), joint.0.is-custom-homing (stat-homing), and joint.0.custom-homing-finished (stat-homed). At the top right, there are buttons for 'Load ethercat-conf.xml', 'Load cia402.comp', and 'Save HAL'. On the far right, the generated HAL code is displayed, showing the configuration for three axes (X, Y, and Z) with their respective servo-thread definitions and homing logic.

```
# =====
# AUTO GENERATED HAL - FULL PDO SUPPORT
#
loadit [KINS]KINEMATICS
loadit [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[KINS]JOINTS
loadusr -W lcec_conf ethercat-conf.xml
loadit cia402 count=3
loadit lcec

addf lcec.read-all servo-thread
addf cia402.0.read-all servo-thread
addf cia402.1.read-all servo-thread
addf cia402.2.read-all servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf cia402.0.write-all servo-thread
addf cia402.1.write-all servo-thread
addf cia402.2.write-all servo-thread
addf lcec.write-all servo-thread

setp ioccontrol.0.emc-enable-in 1

# ----- AXIS X / joint.0 / cia402.0 / slave.1 -----
setp cia402.0.pos-scale 1677721.6
setp cia402.0.csp-mode 1

net X-pos-cmd joint.0.motor-pos-cmd => cia402.0.pos-cmd
net X-pos-fb cia402.0.pos-fb => joint.0.motor-pos-fb
net X-enable joint.0.amp-enable-out => cia402.0.enable
net X-amp-fault cia402.0.drv-fault => joint.0.amp-fault-in
net X-custom-home joint.0.request-custom-homing => cia402.0.home
net X-is-custom-homing cia402.0.stat-homing => joint.0.is-custom-homing
net X-custom-home-done cia402.0.stat-homed => joint.0.custom-homing-finished

net X-control-word cia402.0.controlword => lcec.0.1.controlword
net X-target-position cia402.0.drv-target-position => lcec.0.1.target-position
net X-omode cia402.0.omode => lcec.0.1.omode
net X-actual-position cia402.0.drv-actual-position => lcec.0.1.actual-position
net X-actual-velocity cia402.0.drv-actual-velocity => lcec.0.1.actual-velocity
net X-omode-display cia402.0.omode-display => lcec.0.1.omode-display
net X-status-word lcec.0.1.status-word => cia402.0.statusword

# ----- AXIS Y / joint.1 / cia402.1 / slave.2 -----
setp cia402.1.pos-scale 1677721.6
setp cia402.1.csp-mode 1

net Y-pos-cmd joint.1.motor-pos-cmd => cia402.1.pos-cmd
net Y-pos-fb cia402.1.pos-fb => joint.1.motor-pos-fb
net Y-enable joint.1.amp-enable-out => cia402.1.enable
net Y-amp-fault cia402.1.drv-fault => joint.1.amp-fault-in
net Y-custom-home joint.1.request-custom-homing => cia402.1.home
net Y-is-custom-homing cia402.1.stat-homing => joint.1.is-custom-homing
net Y-custom-home-done cia402.1.stat-homed => joint.1.custom-homing-finished
```

## 2.4. Save HAL

The screenshot shows a list interface titled 'HAL files'. It contains a single item: 'HAL files'. This interface is likely used for saving or managing the generated HAL files.

## 3.INI Generator

### 3.1 Load HAL

The HAL file is treated here as a description of logical connections. In LinuxCNC, an abstract logical model is created that corresponds to how LinuxCNC sees the machine.

Here it is similar, except that the file is only analyzed. The program checks whether the servos share common signals and whether the set of signals is consistent.

The machine moves within a closed space, and homing in this configuration is intended for absolute encoders, which most often use soft limits [MIN\_LIMIT] [MAX\_LIMIT] often without the use of physical limit switch inputs and I/O modules. For this reason, before using automatic configuration, proper installation of the CIA402 homing component is required.

(linuxcnc-dev, cia402\_homecomp.comp, cia402\_homecomp.h, basecomp.comp)

The screenshot shows the HAL -> INI Generator interface. The left side features a tree view of detected elements under sections like JOINTS, SERVOS, and VALIDATION. The right side shows the generated INI file with various configuration parameters. The generated INI file includes sections for [EMC], [DISPLAY], [RS274NGC], [AXIS], [JOINT], [TRAJ], [HAL], [EMCIO], and [RS274NGC].

```

[EMC]
MACHINE = Generated_EtherCAT
DEBUG = 0
VERSION = 1.1

[DISPLAY]
DISPLAY = axis
EDITOR = gedit
POSITION_OFFSET = RELATIVE
POSITION_FEEDBACK = ACTUAL
ARCDIVISION = 64
GRIDS = 10mm 20mm 50mm 100mm lin 2in 5in 10in
MAX_FEED_OVERRIDE = 1.2
DEFAULT_LINEAR_VELOCITY = 50
MAX_ANGLULAR_VELOCITY = 50
MIN_LINEAR_VELOCITY = 0
MAX_LINEAR_VELOCITY = 50
CYCLE_TIME = 0.100
INTRO_GRAPHIC = linuxcnc.gif
INTRO_TIME = 1
INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm

[KIN]
JOINTS = 4
KINEMATICS = trivkins coordinates=XYZA

[TASK]
TASK = milltask
CYCLE_TIME = 0.010

[EMCMOT]
EMCMOT = motmod
COMM_TIMEOUT = 1.0
SERVO_PERIOD = 1000000
HOMEMOD = cia402_homecomp

[HAL]
HALFILE = hahahal.hal
HALUI = halui

[TRAJ]
COORDINATES = XYZA
LINEAR_UNITS = mm
ANGULAR_UNITS = degree
DEFAULT_LINEAR_VELOCITY = 5
MAX_LINEAR_VELOCITY = 50

[JOINT]
TYPE = LINEAR
HOME = 0
MIN_LIMIT = -1000
MAX_LIMIT = 1000
MAX_ACCELERATION = 100
MAX_VELOCITY = 50
FERROR = 1000
MIN_FERROR = 1000
HOME_ABSOLUTE_ENCODER = 2

[EMCIO]
EMCIO = io
CYCLE_TIME = 0.100

[RS274NGC]
PARAMETER_FILE = gcodeparam.var

[AXIS_X]
MAX_VELOCITY = 50
MAX_ACCELERATION = 100
MIN_LIMIT = -1000
MAX_LIMIT = 1000

[JOINT_0]

```

### 3.2.Gantry mode

In other words, using two motors in one plane.

After switching to gantry mode, the following arrangement is created:

1 : 1

Axis X → joint.0

Gantry

Axis Y → joint.1

Axis X → joint.0

Axis Z → joint.2

Axis Y → joint.1 , joint.2

Axis A → joint.3

Axis Z → joint.3

Axis A → joint.3

And the parameters in [KINS] and [TRAJ] change dynamically and later define the recognition of the mode by LinuxCNC during loading.

The screenshot shows the configuration interface for a gantry setup. On the left, there are several tabs: JOINTS, SERVOS, and VALIDATION. Under JOINTS, it lists four servos (cia402.0, cia402.1, cia402.2, cia402.3) with their respective joint assignments. The SERVOS tab shows the servo model (Generated\_EtherCAT). The VALIDATION tab includes sections for Essential signals and Axis cohesion.

**EMC** section contains parameters like Enable, MACHINE (Generated\_EtherCAT), DEBUG (0), and VERSION (1.1).

**DISPLAY** section contains parameters like Enable, DISPLAY (axis), EDITOR (gedit), POSITION\_OFFSET (RELATIVE), POSITION\_FEEDBACK (ACTUAL), ARCDIVISION (64), GRIDS (10mm 20mm 50mm 100mm), MAX\_FEED\_OVERRIDE (1.2), DEFAULT\_LINEAR\_VELOCITY (5), MAX\_ANGULAR\_VELOCITY (50), MIN\_LINEAR\_VELOCITY (0), MAX\_LINEAR\_VELOCITY (50), CYCLE\_TIME (0.100), INTRO\_GRAPHIC (linuxcnc.gif), INTRO\_TIME (1), and INCREMENTS (5mm 1mm .5mm .1mm).

**KINS** section contains parameters like Enable, JOINTS (4), and KINEMATICS (trivkins kinstype=both).

**TASK** section contains parameters like Enable, TASK (milltask), and CYCLE\_TIME (0.010).

**RS274NGC** section contains parameters like Enable, PARAMETER\_FILE (gcodeparam.var).

**EMCMOT** section contains parameters like Enable, EMCMOT (motmod), COMM\_TIMEOUT (1.0), SERVO\_PERIOD (1000000), and HOMEMOD (cia402\_homecomp).

**HAL** section contains parameters like Enable, HALFILE (hahahal.hal), and HALUI (halui).

**AXIS** section contains parameters like Enable, MAX\_VELOCITY (50), MAX\_ACCELERATION (100), MIN\_LIMIT (-1000), and MAX\_LIMIT (1000).

**TRAJ** section contains parameters like Enable, COORDINATES (X Y Y Z), LINEAR\_UNITS (mm), ANGULAR\_UNITS (degree), DEFAULT\_LINEAR\_VELOCITY (5), and MAX\_LINEAR\_VELOCITY (50).

**JOINT** section contains parameters like Enable, TYPE (LINEAR), HOME (0), MIN\_LIMIT (-1000), MAX\_LIMIT (1000), MAX\_VELOCITY (50), MAX\_ACCELERATION (100), FERROR (1000), MIN\_FERROR (1000), and HOME\_ABSOLUTE\_ENCODER (2).

**EMCIO** section contains parameters like Enable, EMCIO (io), CYCLE\_TIME (0.100).

**[KINS]** section contains parameters like JOINTS (4), KINEMATICS (trivkins kinstype=both), and COORDINATES (X Y Y Z).

**[TASK]** section contains parameters like TASK (milltask), CYCLE\_TIME (0.010), and HOMEMOD (cia402\_homecomp).

**[EMCMOT]** section contains parameters like EMCMOT (motmod), COMM\_TIMEOUT (1.0), SERVO\_PERIOD (1000000), and HOMEMOD (cia402\_homecomp).

**[TRAJ]** section contains parameters like COORDINATES (X Y Y Z), LINEAR\_UNITS (mm), ANGULAR\_UNITS (degree), DEFAULT\_LINEAR\_VELOCITY (5), and MAX\_LINEAR\_VELOCITY (50).

**[HAL]** section contains parameters like HALFILE (hahahal.hal), HALUI (halui), and EMCIO (io).

**[EMCIO]** section contains parameters like CYCLE\_TIME (0.100).

**[RS274NGC]** section contains parameters like PARAMETER\_FILE (gcodeparam.var).

**[AXIS\_X]** section contains parameters like MAX\_VELOCITY (50), MAX\_ACCELERATION (100), MIN\_LIMIT (-1000), and MAX\_LIMIT (1000).

**[JOINT\_0]** section contains parameters like TYPE (LINEAR), HOME (0), MIN\_LIMIT (-1000), MAX\_LIMIT (1000), MAX\_VELOCITY (50), MAX\_ACCELERATION (100), FERROR (1000), MIN\_FERROR (1000), and HOME\_ABSOLUTE\_ENCODER (2).

**[AXIS\_Y]** section contains parameters like MAX\_VELOCITY (50), MAX\_ACCELERATION (100), MIN\_LIMIT (-1000), and MAX\_LIMIT (1000).

**[JOINT\_1]** section contains parameters like TYPE (LINEAR), HOME (0), MIN\_LIMIT (-1000), MAX\_LIMIT (1000), and MAX\_VELOCITY (50).

### 3.3.Save INI

The screenshot shows a file selection dialog with the following path: C:\Users\Public\Documents\lct\ini\INI files. The dialog has a search bar at the top and a list of files below it.

---

#### **4. DosStyle → UTF-8 LF Conventer**

---

**4.1** When files are saved in the programs presented above on Windows, DosStyle line endings are created. It is enough to enable the Converter, enter the given folder with the files, or load the files individually, or paste the file path, and press convert.

