Z0 procedure

The first intervention consists of renaming the "Common" pad to "Z-Sensor" to make the interface more intuitive, The "TLO" field (which has no place in this part of the interface) is replaced by a "Sensor Height" field. The modified interface now has only one role: to enter and store the sounding parameters (approach speed, measurement speed, height of the measuring sensor and sounding process)

The second intervention removes ambiguous variables to make the code easier to read: a beautiful duplicate CNC.vars["TLO"] in CNC.py class CNC which will be replaced by CNC.vars["Zsensor"] and creates a specific variable to memorize the height of the sensor: "sensor". Any reference to "TLO" in the definition part of the working repository is removed; she has no place there. For the record, the term TLO in the world of the CNC is not a generic term, it exclusively designates a tool length offset. It should only be used in the context of a tool change and never to indicate a sensor height or a work frame offset.

Below are the changes to be made. They have been tested with bCNC version 0.9.14.317 Attention with bCNC 0.9.14.318 the passing of arguments has been modified:

ainsi cmd += "Z%s"%(v) prend la forme cmd += f"Z{v}"

For those who use an interface in a language other than English, the language module must be updated with the Poedit utility available for download.

Side effect of this change: The status tile now displays the current tool number and its TLO.

Z0 Specific variable creation for H-Sensor

CNC.py file

class CNC:

replace duplicate (CNC.vars["TLO"] with (CNC.vars["Zsensor"] : 0.0
Create variable "sensor" : ""

ProbePage.py file

Change : from CNC import CNC, Block
with : from CNC import CNC, Block, WCS

class ProbeCommonFrame(CNCRibbon.PageFrame):

change : lframe = tkExtra.ExLabelFrame(self, text=_("Common"),foreground="DarkBlue")
with : lframe = tkExtra.ExLabelFrame(self, text=_("Z-Sensor"),foreground="DarkBlue")

change : Label(frame, text=_("TLO")).grid(row=row, column=col, sticky=E)
with : Label(frame, text=_("Sensor Height")).grid(row=row, column=col,
sticky=E)

tool offset
…. is deleted

change with :

```
# Probe sensor
row += 1
col        = 0
Label(frame, text=_("Zsensor")).grid(row=row, column=col, sticky=E)
col += 1
self.probeSensorVar = StringVar()
self.probeSensorVar.trace("w", lambda *_: ProbeCommonFrame.probeUpdate())
ProbeCommonFrame.probeSensor = tkExtra.FloatEntry(frame,
background=tkExtra.GLOBAL_CONTROL_BACKGROUND, width=5,
                                                  textvariable=self.probeSensorVar)
ProbeCommonFrame.probeSensor.grid(row=row, column=col, sticky=EW)
tkExtra.Balloon.set(ProbeCommonFrame.probeSensor, _("Set sensor height for probing"))
self.addWidget(ProbeCommonFrame.probeSensor)
```

def tloSet(self, event=none) ... is deleted

change with :

```
#---------------------------------------------------------------------
```

```
def sensorSet(self, event=None):
    try:
        CNC.vars["Zsensor"] = float(ProbeCommonFrame.probeSensor.get())
    except:
        pass
    self.app.mcontrol.viewParameters()
```

```
def probeUpdate():

        Add : CNC.vars["Zsensor"]= float(ProbeCommonFrame.probeSensor.get())
```

```
def saveConfig(self)

        change : Utils.setFloat("Probe", "tlo", ProbeCommonFrame.tlo.get())
        with     : Utils.setFloat("Probe", "sensor",ProbeCommonFrame.probeSensor.get())
```

```
def loadConfig(self)

        Change : ProbeCommonFrame.tlo.set(Utils.getFloat("Probe","tlo"))
        with  :    ProbeCommonFrame.probeSensor.set(Utils.getFloat("Probe","sensor"))
```

This change makes the input of the height of the sensor permanent. If the sensor is changed, the new value must be entered (eg touch plate vs surfave Pcb...). The entered value is written in the .bCNC initialization file, which makes it available as soon as the software is launched.
H Sensor Compensation

ProbePage.py file

```
#==============================================================================
# Probe Frame
#==============================================================================
class ProbeFrame(CNCRibbon.PageFrame):
        #----------------------------------------------------------------------

def updateProbe(self):
        try:
                self._probeX["text"] = CNC.vars.get("prbx")
                self._probeY["text"] = CNC.vars.get("prby")
                self._probeZ["text"] = CNC.vars.get("prbz")
        except:
                return

        if self.probeautogotonext:
                self.probeautogotonext = False
                self.gotoAfterProbe()  >>> replace goto2probe

#----------------------------------------------------------------------
# Probe one Point
#----------------------------------------------------------------------
def probe(self, event=None)
        ...
        ...
        if ok:
                self.gcode(cmd)
        else:
                tkMessageBox.showerror(_("Probe Error"),
                                       _("At least one probe direction should be specified"))

becomes :
        if ok:
                self.sendGCode(cmd) # search switch sensor
                p = WCS.index(CNC.vars["WCS"]) + 1  # Actual Work Space
                cmd = "G10 L20 P%d Z%s"%((p), ProbeCommonFrame.probeSensor.get())   # Sensor height
                self.sendGCode(cmd)
                cmd = "G4 P1"   # Wait 1s
                self.sendGCode(cmd)
        else:
                tkMessageBox.showerror(_("Probe Error"),
                                       _("At least one probe direction should be specified"))



create :

#----------------------------------------------------------------------
# Rapid move after probe
#----------------------------------------------------------------------
def gotoAfterProbe(self, event=None):
        try:
                cmd = "G91 G0 Z5"   # Sensor clearence  5 mm
                self.sendGCode(cmd)
        except:
                return
                cmd = "G90" # default value
                self.sendGCode(cmd)
```