Making connections into python from the XML gui .ui file using linuxcnc QTvcp.

By Johannes P. Fassotte – auto-mation-assist, Fairbanks, Alaska 10/25/2018

In by note here I will refer to items, items could be a button, action button, or any other widget.

The the QT designers signal/slot editor is used with QTvcp its has has several modes. First if you have a simple gui and everything is placed on the MainWindow and you are in the signal/slot editor for editing you are in what I call easy mode.

In easy mode when you click on a item that will be linked into linuxcnc python and move the mouse, then the mouse pointer will always change to the electrical ground symbol. Configuration can now be done in the Configure Connection pop up appears and selecting edit.

Then under slots click on +, select the slot that pops up and rename it to your desired name making sure it ends with (). Then click OK to save it. Now you will see the new name in the list on the right window. From the left list click on the desired item, then click on the newly named item in the right list and push OK. Now everything is done that in going to describe in my notes below. However things are not always that simple depending on the gui design and complexity.

If we add a frame below the item mentioned above we find out that we now have to move the mouse outside of the frames limits to get the same result. Now if we add more and more frame that overlap all the way to the edge of a gui we will never be able land the mouse pointer on the most lower MainWindow. The result is not entry into easy mode available due to a inability to get to the slot list being on the linuxcnc slot list being on the MainWindow. Perhaps the slot list could be but in a slot handler.ui in the future. But for now read my notes they could be useful to you since it is what it is.

All right lets start;

If you have a complex gui with lots of items and layers the normal way of pointing to a specific button or item in the qt designer signal/slot editor may not be the best choice. By normal I mean getting the electrical ground symbol to appear after the normal qt pointer arrow is shown.

The purpose of the electrical ground symbol for the mouse pointer is to generate a mode that allows building a slot list in the XML file <class>MainWindow</class> and make a entry there for the name of the item that was clicked on and generate a slot name for that item that python can understand. like this as an example:

<slot>my_button_name_toggled()</slot>

Normally qt designer would make the this:

 <slot>toggle()</slot>   which linuxcnc python will not be happy with.

The slot list that is generated is located near the very bottom of the gui XML .ui file, just above the XML file end :

</ui>.

The main MainWindow class is defined at the top of the XML file. There are likely many thousands of items between MainWindow and the slot list. As a example if we were to eliminate those thousands of items everything would line up like the below sample from top of the XML .ui file to its end:

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <slots>
  <slot>my_button_name_toggled()</slot>
 </slots>
</ui>
```

So you can see that the slot list is actually part of the MainWindow and that its alignment with it is important for being able to access the slots list. As items are added to the slot list it grows like shown below. Items can be added manually with a text editor taking care to keep everything aligned and starting each new slot with <slot> start and ending it with</slot> as below.

```
<slots>
  <slot>my_button_name_toggled()</slot>
  <slot>pbtn_jog_step_1_toggle()</slot>
  <slot>pbtn_jog_step_2_toggle()</slot>
  <slot>pbtn_jog_step_3_toggle()</slot>
 </slots>
```

It is much faster for me to add slots for desired names manually with a text editor then when using the designers signal and slot editor for complex layered designs. However before entries that are made manually in the slot list are be able to be used by a specific item or button added there needs to be connection parameters added to the XML .ui file. This can also be done manually with a text editor if the attributes for the item is known. If not then I suggest using the designer signal/slot editor in its normal mode with just the arrow mouse pointer. It will make the required entries that can be manually edited later to get them working.

A typical entry in the XML .ui file may look like this when using the normal none ground symbol mouse pointer in the signal/slot editor but it allowed choosing an available signal type (attribute) of toggled(bool). This is not compatible with a gui’s handler.py file due to name errors in receiver and slot. Hits are not really needed at all but convenient when viewing the signal/tool edit in the designer if they do not get to cluttered, if so they can be removed starting at and including <hints> up to and including /hints> with no ill effect.

```
 <connection>
   <sender>my_button_name</sender>
   <signal>toggled(bool)</signal>
   <receiver>my_button_name</receiver>
   <slot>toggle()</slot>
   <hints>
    <hint type="sourcelabel">
     <x>233</x>
     <y>438</y>
    </hint>
    <hint type="destinationlabel">
     <x>233</x>
     <y>437</y>
    </hint>
   </hints>
  </connection>
```

This has added everything that we need and thus is thus available for editing. Note that <connection> start to </connection> connection end should not be confused with </connections > start <connections> end. In the later case is the segment in the XML .ui file that contains all available items that can be linked to linuxcnc python handler. Text editor search is your friend.

In looking at the added items:

<sender>my_button_name</sender>	Contains the name of the item: my_button_name
<signal>toggled(bool)</signal>	Contains the function: toggled
<slot>toggle()</slot>	Contains the name to be linked to python.

Python will not understand "toggle()" and it will cause a segmentation fault error. Therefore the slot name needs to be corrected and the signal name changed to MainWindow.

Example:

```
<signal>MainWindow</signal>
<signal>my_button_name_toggle()</signal>
```

Three things needed to form the name, "my_button_name" from <sender> plus "toggle" from <slot>plus "()" brackets. You actually name it anything you want but be sure to add the () on the end and be sure that it is the same as the name in the slot list.

Now we can reduce the clutter if needed by making sure that the labels are on top of each other or delete them completely if the editor screen get really cluttered up. Hints make it easy to see what has been added to be linked with python but this is not really needed.

```
<hints>
    <hint type="sourcelabel">
     <x>233</x>
     <y>438</y>
    </hint>
    <hint type="destinationlabel">
     <x>235</x>
     <y>437</y>
```

Make destinationlabel x Y postion the same as the sourcelabel x y location.

```
<hint type="destinationlabel">
     <x>233</x>
     <y>438</y
```

Hints make it easy to see what has been added to be linked with linuxcnc python but this is not really needed and can removed> so we would end up with this size enry in the XML .ui file instead making sure alignment is correct:

```
  <connection>
   <sender>my_button_name</sender>
   <signal>toggled(bool)</signal>
   <receiver>my_button_name</receiver>
   <slot>my_button_name_toggle()</slot>
  </connection>
```

Thus if you did not want to use the designer editor at all and enter information manually just make an copy template like this and enter the correct information and paste it in the correct area of the XML .UI file like this in the correct area and alignment and enter the information between the >< pointers.

```
  <connection>
   <sender></sender>
   <signal></signal>
   <receiver></receiver>
   <slot></slot>
  </connection>
```

Search for the below at the same time and you will just below the last entry in the connection stack. Paste it just above these:

```
 </connections>
 <slots>
```

So for a new button or any item to be linked to linuxcnc python can be added with the below if hints are not wanted and you know the names to be entered.

This show the last part of a XML .ui file but the slots area will likely have a lot more entries.

```
 </connection>  **** end of the prior item
  <connection>  **** start of our new item
   <sender>my_button_name</sender>
   <signal>toggled(bool)</signal>
   <receiver>my_button_name</receiver>
   <slot>my_button_name_toggle()</slot>
  </connection> **** end of our new item
 </connections> **** present connection list
  <slots>
   <slot>my_button_name_toggled()</slot>  **** added slot name
  </slots
 </ui>
```

The desinger editor lets us choose a avaliabel signal type.

Please edit the file as required to eliminate errors.

The end