```python
#!/usr/bin/env python
##################################################
# Gnuradio Python Flow Graph
# Title: Tcp Server
# Generated: Sun Mar 19 13:47:53 2017
##################################################

from gnuradio import eng_notation
from gnuradio import gr
from gnuradio import wxgui
from gnuradio.eng_option import eng_option
from gnuradio.filter import firdes
from gnuradio.wxgui import scopesink2
from grc_gnuradio import blks2 as grc_blks2
from grc_gnuradio import wxgui as grc_wxgui
from optparse import OptionParser
import wx

class tcp_server(grc_wxgui.top_block_gui):

    def __init__(self):
        grc_wxgui.top_block_gui.__init__(self, title="Tcp Server")
        _icon_path = "/usr/share/icons/hicolor/32x32/apps/gnuradio-grc.png"
        self.SetIcon(wx.Icon(_icon_path, wx.BITMAP_TYPE_ANY))

        ##################################################
        # Variables
        ##################################################
        self.samp_rate = samp_rate = 32000

        ##################################################
        # Blocks
        ##################################################
        self.wxgui_scopesink2_0 = scopesink2.scope_sink_f(
          self.GetWin(),
          title="Scope Plot",
          sample_rate=samp_rate,
          v_scale=0,
          v_offset=0,
          t_scale=0,
          ac_couple=False,
          xy_mode=False,
          num_inputs=1,
          trig_mode=wxgui.TRIG_MODE_AUTO,
          y_axis_label="Counts",
        )
        self.Add(self.wxgui_scopesink2_0.win)
        self.blks2_tcp_source_0 = grc_blks2.tcp_source(
          itemsize=gr.sizeof_float*1,
          addr="0.0.0.0",
          port=12345,
          server=True,
        )

        ##################################################
        # Connections
        ##################################################
        self.connect((self.blks2_tcp_source_0, 0), (self.wxgui_scopesink2_0, 0))


    def get_samp_rate(self):
        return self.samp_rate

    def set_samp_rate(self, samp_rate):
        self.samp_rate = samp_rate
        self.wxgui_scopesink2_0.set_sample_rate(self.samp_rate)

if __name__ == '__main__':
    import ctypes
    import sys
    if sys.platform.startswith('linux'):
        try:
            x11 = ctypes.cdll.LoadLibrary('libX11.so')
            x11.XInitThreads()
        except:
            print "Warning: failed to XInitThreads()"
    parser = OptionParser(option_class=eng_option, usage="%prog: [options]")
    (options, args) = parser.parse_args()
    tb = tcp_server()
    tb.Start(True)
    tb.Wait()
```