```
import time
import ui_plot
import sys
import numpy
from PyQt4 import QtCore, QtGui
import PyQt4.Qwt5 as Qwt
import spidev

#Open SPI Bus
spi = spidev.SpiDev()
spi.open(0,0) #1st Param is Channel

def readChannel(chan):
    adc = spi.xfer2([1, (8+chan)<<4, 0]) #sends 3 bytes
    print(adc)
    data = ((adc[1]&3)<<8) + adc[2]
    print(data)
    return data


#Sampling the input wave
numPoints=100
xs=numpy.arange(numPoints)
ys = [None] * numPoints #Creating an empty array of size numPoints
i = 0
while i<numPoints:
    time.sleep(0.0001)
    ys[i] = readChannel(0)
    i = i+1

#Calculating frequency based on sample
valley1y = 0
valley1x = 0
for x in range (1, numPoints - 1):
    if ys[x] < ys[x - 1] and ys[x] <= ys[x + 1]:
        valley1y = ys[x]
        valley1x = x
        break

valley2y = 0
valley2x = 0
for x in range (valley1x + 1, numPoints - 1):
    if ys[x] < ys[x - 1] and ys[x] <= ys[x + 1]:
        valley2y = ys[x]
        valley2x = x
        break

wavelength = (valley2x - valley1x) / 10000.0
frequency = (1.0 / wavelength)/3 #Calcuation with error and time constant
print("\nFREQUENCY:")
print(frequency)

#Plotting the sample that was calculated in xs and ys
def plotSomething():
    global ys
    ys=numpy.roll(ys,-1)
    #print "PLOTTING"
    c.setData(xs, ys)
    uiplot.qwtPlot.replot()

if __name__ == "__main__":
##    readChannel(0);

    app = QtGui.QApplication(sys.argv)

    ### SET-UP WINDOWS
```

```
67
68      # WINDOW plot
69      win_plot = ui_plot.QtGui.QMainWindow()
70      uiplot = ui_plot.Ui_win_plot()
71      uiplot.setupUi(win_plot)
72      c=Qwt.QwtPlotCurve()
73      c.attach(uiplot.qwtPlot)
74
75      uiplot.timer = QtCore.QTimer()
76      uiplot.timer.start(100.0)
77
78      win_plot.connect(uiplot.timer, QtCore.SIGNAL('timeout()'), plotSomething)
79
80
81      ### DISPLAY WINDOWS
82      win_plot.show()
83
84      #WAIT UNTIL QT RETURNS EXIT CODE
85      sys.exit(app.exec_())
```