import sys

from PySide import QtGui, QtCore

import RPi.GPIO as gpio

import spidev

import os

from time import sleep

spi = spidev.SpiDev()

spi.open(0,0)

spi.max\_speed\_hz = 2000000

gpio.setmode(gpio.BCM)

gpio.setup((5,6,20,13,19), gpio.OUT)

gpio.output((19,13,6,5,20), True)

y=1

class Window(QtGui.QMainWindow):

def \_\_init\_\_(self):

super(Window, self).\_\_init\_\_()

self.setGeometry(200, 200, 300, 150)

self.setWindowTitle("ADC Reader")

self.setWindowIcon(QtGui.QIcon('Celldrifter-Muku-Style-Sys-Command.ico'))

self.statusBar()

self.home()

def home(self):

self.tmbl\_1 = QtGui.QPushButton(self)

self.tmbl\_1.setText("Baca ADC")

label\_1 = QtGui.QLabel("Hasil ADC", self)

self.tmbl\_1.move(180, 60)

label\_1.move(130, 25)

self.tmbl\_1.clicked.connect(self.tmbl1)

self.tmbl\_2 = QtGui.QPushButton(self)

self.tmbl\_2.setText("Stop")

self.tmbl\_2.move(20, 100)

self.tmbl\_2.clicked.connect(self.tmbl2)

label\_2 = QtGui.QLabel("Device Mode : 1", self)

label\_2.move(190, 100)

label\_3 = QtGui.QLabel("Volt ", self)

label\_3.move(20, 25)

menu = self.menuBar()

QuitAct = QtGui.QAction('&Quit', self)

QuitAct.setShortcut('Ctrl+X')

QuitAct.setStatusTip("Keluar")

QuitAct.triggered.connect(self.close\_apps)

fileMenu = menu.addMenu("&File")

fileMenu.addAction(QuitAct)

EditorAct = QtGui.QAction('&Editor', self)

EditorAct.setShortcut('Ctrl+X')

EditorAct.setStatusTip("Keluar")

EditorAct.triggered.connect(self.close\_apps)

editMenu = menu.addMenu("&Edit")

editMenu.addAction(QuitAct)

self.comboBox = QtGui.QComboBox(self)

self.comboBox.move(20,60)

self.comboBox.addItem("Pilih Panjang Gelombang")

self.comboBox.addItem("1310")

self.comboBox.addItem("1552")

About = QtGui.QAction('&About', self)

About.setShortcut('Ctrl+X')

About.setStatusTip("About")

About.triggered.connect(self.msgbox)

AboutMenu = menu.addMenu("&About")

AboutMenu.addAction(About)

self.show()

def tmbl2(self):

gpio.cleanup()

def tmbl1(self):

About = QtGui.QAction('&Reading ADC', self)

About.setStatusTip("Reading ADC")

value = int(self.comboBox.currentText())

konv = self.konversi()

if (0.5>konv)&(y==1):

gpio.output(19, False)

self.label\_2.setText('Device Mode 2')

y=2

else:

if (0.5>konv)&(y==2):

gpio.output(13, False)

self.label\_2.setText('Device Mode 3')

y=3

else :

if (0.5>konv)&(y==3):

gpio.output(6, False)

self.label\_2.setText('Device Mode 4')

y=4

else:

if (0.5>konv)&(y==4):

gpio.output(5, False)

self.label\_2.setText('Device Mode 5')

y=5

else:

if (0.5>konv):

self.label\_2.setText('Device Outranged')

if (konv>4.5)&(y==2):

gpio.output(19, True)

self.label\_2.setText('Device Mode 1')

y=1

else:

if (konv>4.5)&(y==3):

gpio.output(13, True)

self.label\_2.setText('Device Mode 2')

y=2

else :

if (konv>4.5)&(y==4):

gpio.output(6, True)

self.label\_2.setText('Device Mode 3')

y=3

else:

if (konv>4.5)&(y==5):

gpio.output(5, True)

self.label\_2.setText('Device Mode 4')

y=4

else:

if (konv>4.5):

self.label\_2.setText('Device Outrange')

sleep(1)

konv = self.konversi()

self.label\_3.setText(str(konv)+" Volt")

if (y==5)&(value==1310):

x = (log((konv/14429)))/0.225

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==4)&(value==1310):

x = (log((konv/1577)))/0.229

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==3)&(value==1310):

x = (log((konv/158.7)))/0.229

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==2)&(value==1310):

x = (log((konv/14.67)))/0.223

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==1)&(value==1310):

x = (log((konv/1.588)))/0.231

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==5)&(value==1552):

x = (log((konv/14825)))/0.224

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==4)&(value==1552):

x = (log((konv/1745)))/0.230

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==3)&(value==1552):

x = (log((konv/176.1)))/0.231

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==2)&(value==1552):

x = (log((konv/16.83)))/0.229

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

elif (y==1)&(value==1552):

x = (log((konv/1.638)))/0.225

x = round(x, 2)

self.label\_1.setText(str(x)+" dBm")

def msgbox(self):

self.msgbox1 = QtGui.QMessageBox()

self.msgbox1.setWindowTitle('About')

self.msgbox1.setText("Software ini dibuat dalam rangka PKL")

self.msgbox1.exec\_()

def close\_apps(self):

sys.exit()

def read(channel):

if (channel < 0) | (channel > 1):

return -1

adc = spi.xfer2([6,(channel)<<6,0])

data = ((adc[1]&15)<<8) + adc[2]

return data

def konversi(self):

data = read(0)

volt = (data\*5)/float(4095)

volt = round(volt, 3)

return volt

def main():

app = QtGui.QApplication(sys.argv)

app.setStyle('Plastique')

GUI = Window()

sys.exit(app.exec\_())

try :

main()

except :

KeyboardInterrupt()

gpio.cleanup()