```
from sonar import Sonar
from threads import thrd

import time
import threading
import matplotlib.pyplot as plt
import numpy as np

#sonar1=Sonar(3,4)
#sonar2=Sonar(14,15)
#sonar3=Sonar(17,18)
sonars=[Sonar(3,4),Sonar(14,15),Sonar(17,18)]

plt.figure("distance")
#plt.figure("velocity")
plt.ion()
plt.show()
plt.hold(False)

t=np.arange(-len(sonars[0].timeArray),0,1)
dist=np.zeros((3,len(sonars[0].distanceBuffer)))
vel=np.zeros((3,len(sonars[0].velocityBuffer)))

for c in range(5):  # Pre-populate buffers
    for s in range(3):
        sonars[s].measureDistance()
        sonars[s].computeVelocity()

while True:

    for s in range(3):
        sonars[s].measureDistance()
        sonars[s].computeVelocity()

    print "%.3fs> Sonar 0: %s [m]   %s [m/s]" % (sonars[0].timeArray[0],sonars[0].distance, sonars[0].velocity)
    print "%.3fs> Sonar 1: %s [m]   %s [m/s]" % (sonars[1].timeArray[0],sonars[1].distance, sonars[1].velocity)
    print "%.3fs> Sonar 2: %s [m]   %s [m/s]" % (sonars[2].timeArray[0],sonars[2].distance, sonars[2].velocity)
    print ""


    plt.figure("distance")
    plt.plot(sonars[0].timeArray,sonars[0].distanceBuffer,sonars[1].timeArray,sonars[1].distanceBuffer,sonars[2].timeArray,sonars[2].distanceBuffer)
    plt.axis([min(min(sonars[0].timeArray),min(sonars[1].timeArray),min(sonars[2].timeArray)),max(max(sonars[0].timeArray),max(sonars[1].timeArray),max(sonars[2].timeArray)),0,4])
    plt.legend(["Sonar 0","Sonar 1","Sonar 2"])

    plt.figure("velocity")
    plt.plot(sonars[0].timeArray,sonars[0].velocityBuffer,sonars[1].timeArray,sonars[1].velocityBuffer,sonars[2].timeArray,sonars[2].velocityBuffer)
    plt.axis([min(min(sonars[0].timeArray),min(sonars[1].timeArray),min(sonars[2].timeArray)),max(max(sonars[0].timeArray),max(sonars[1].timeArray),max(sonars[2].timeArray)),-1,1])
    plt.legend(["Sonar 0","Sonar 1","Sonar 2"])

    plt.pause(1e-6)

```