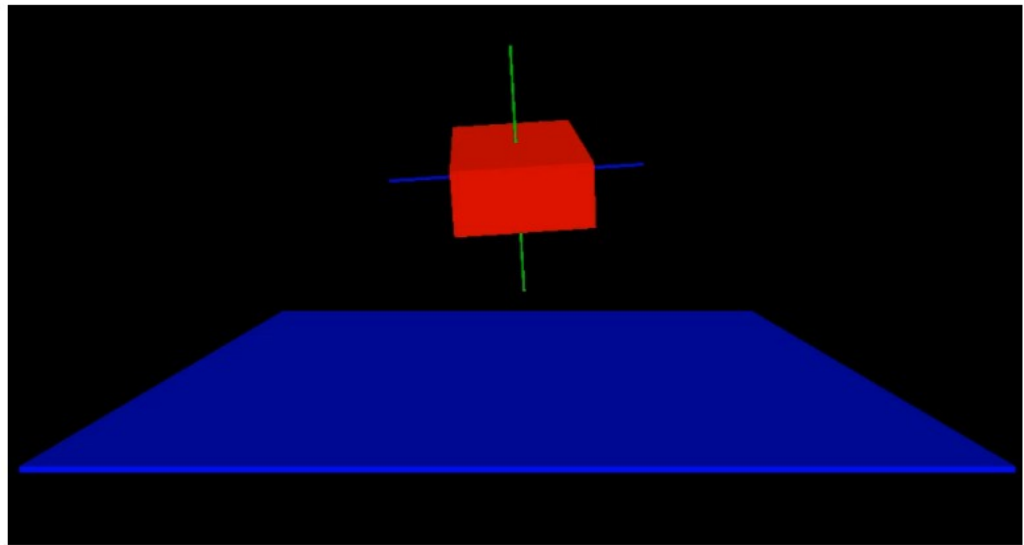


## Proyecto #2 INFO1157

### By Alberto Caro

**Aplicación Sensor: WitMotion Bluetooth 2.0 Mult-Connect BWT901CL 9 Axis IMU Sensor Angle Inclinator + Acceleration + Gyro + Mag on PC/Android/MCU**

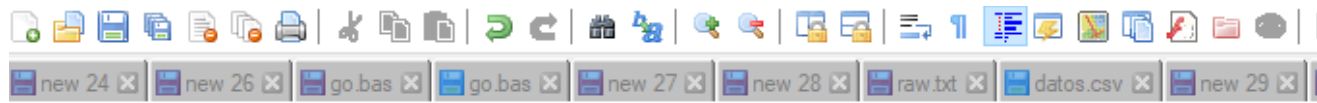


Se pide implementar una simulación mediante la utilización del sensor inercial **BWT901CL** que se conecte con **visual python** y los momentos del **Giróscopo** sean visualizados en una interfaz **3D**. Para tal efecto, se dispone del archivo **inercial.bin** donde se encuentra los paquetes de datos que fueron enviados por este sensor.

Usted deberá decodificar la trama de datos de este archivo y utilizarla para hacer los movimientos del **cujo rojo** en **3D**. En plataforma está disponible un video para que entienda lo que debe realizar. Se adjunta un código de ejemplo, el cual mediante ingeniería inversa lo puede utilizar para dar solución a este interesante desafío tecnológico.

La estructura de datos del archivo **inercial.bin** es la siguiente. Se utiliza el módulo **ctype** de python.

```
2
3 class eTrama(ct.Structure):
4     _fields_ = [
5         ('B0',ct.c_ubyte), ('B1',ct.c_ubyte), ('B2',ct.c_ubyte),
6         ('B3',ct.c_ubyte), ('B4',ct.c_ubyte), ('B5',ct.c_ubyte),
7         ('B6',ct.c_ubyte), ('B7',ct.c_ubyte), ('B8',ct.c_ubyte),
8         ('B9',ct.c_ubyte), ('BA',ct.c_ubyte)
9     ]
```



```
1 # By Alberto Caro
2 # Ingeniero Civil Informatico
3 # Universidad Catolica de Temuco
4 # Dr.(c) Ciencias de la Ingenieria - PUC
5 #-----
6 from __future__ import division, visual import *
7 import math, time as ti, serial as RS, random as ra, struct as ST
8
9 #-----
10 # Constantes utiles globales
11 #-----
12 nMAX_G = 16 ; nOld_R = nOld_P = nOld_Y = 0; nROLL = 0 ; nPITCH = 1 ; nYAW = 2
13 #-----
14 # Definicion del Mundo 3D y sus Objetos.-
15
16 #-----
17 WinM = display(title='Kinect',x=50,y=0,width=1000,height=1000,center=(0,0,0))
18 Base = box(pos=(0,-150,0),size=(500,3,500),color=color.blue)
19
20 #-----
21 # Retorna el Frame Base donde se acoplan todos los objetos
22 #-----
23 def Get_Frame(tFXYZ,nClr):
24     oFr_B = frame(pos=tFXYZ)
25     oFr_B.ob1=box(frame=oFr_B,pos=(0,0,0),size=(100,060,160),color=nClr)
26     oFr_B.ob2=box(frame=oFr_B,pos=(0,0,0),size=(200,002,002),color=color.blue)
27     oFr_B.ob3=box(frame=oFr_B,pos=(0,0,0),size=(002,200,002),color=color.green)
28     return oFr_B
29
30 #-----
31 # Realiza las rotaciones por los ejes (X,Y,Z)
32 #-----
33 def Rota(nEje,nAng,xObj):
34     global nOld_R,nOld_P,nOld_Y
35     if (nEje == nROLL): # Eje X
36         nDif = nAng - nOld_R
37         xObj.rotate(angle = -1*radians(nDif),axis=(0,0,1),origin =(0,0,0))
38         nOld_R = nAng # Guardamos el nuevo Roll
39     if (nEje == nYAW):# Eje Z
40         nDif = nAng - nOld_Y
41         xObj.rotate(angle = 1*radians(nDif),axis=(0,1,0),origin =(0,0,0))
42         nOld_Y = nAng # Guardamos el nuevo YAW
43     if (nEje == nPITCH): # Eje Y
44         nDif = nAng - nOld_Y
45         xObj.rotate(angle = -1*radians(nDif),axis=(1,0,0),origin =(0,0,0))
46         nOld_Y = nAng # Guardamos el nuevo YAW
47     return
```

```
52
53
54 #-----
55 # Codigo Main
56 #-----
57 sLine = '%03d,%03d,%03d,%03d,%03d,%03d,%03d,%03d,%03d,%03d,%03d\n'
58 cnx = RS.Serial('COM9'); cnx.baudrate = 115200
59 aObj = [Get_Frame((0,0,0),color.red)]
60 aP = [0,0,0,0,0,0,0,0,0,0,0] ; nF = open('inercial.txt','w')
61 while 1:
62     sT = cnx.read(2) # leemos 2 bytes cabecera
63     aD = ST.unpack('B'*len(sT),sT)
64     if (aD[0] == 0x55):
65         if (aD[1] == 0x53): # Inclination = (Roll, Pitch, Yaw)
66             sT = cnx.read(9) # Leemos Paquete de datos restante
67             aP[0:2] = aD # Sliding Array
68             aD = ST.unpack('B'*len(sT),sT)
69             aP[2:] = aD # Sliding Array
70             Rx = ((aD[1] << 8 | aD[0])/32768.0) * 180 # Roll
71             Py = ((aD[3] << 8 | aD[2])/32768.0) * 180 # Pitch
72             Yz = ((aD[5] << 8 | aD[4])/32768.0) * 180 # Yaw
73             Te = ((aD[7] << 8 | aD[6])/340.0 ) + 36.53 # Temperatura
74             nRol = Rx # Roll ; nPit = Py # Pitch ; nYaw = Yz # Yaw
75             Rota(nROLL,nRol,aObj[0]) ; Rota(nPITCH,nPit,aObj[0])
76             #Rota(nYAW ,nYaw,aObj[0]) -> No utilizamos Rotacion Eje Z Yaw)
77             nF.write(sLine %(aP[0],aP[1],aP[2],aP[3],aP[4],aP[5],aP[6],\
78                             aP[7],aP[8],aP[9],aP[10]))
79             rate(10)
80
81 s.close()
82
```

#### Observaciones:

- Trabajo grupo de 3 personas o individual
- Prohibido compartir códigos. Cualquier leve evidencia de copia los grupos o personas involucradas reprobarán el curso de manera inmediata.
- Informe impreso y en PDF
- Fecha de entrega y defensa: **Jueves 20 de Octubre desde las 15:00 AM hasta 17:00 PM**
- En plataforma se encuentra un **video** aclaratorio + archivo **inercial.bin**

