```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

#Parameters
n=10000 #number of sample points
thick=1.0 #slice thickness in mm
w=32 #number of slabs/slices
k=w  #number of helix repetitions/loops ??
L_total=k*thick  #total length along z axis
HP=21 #hellical pitch (**not sure how to use**)

# Plot a helix along the z-axis


a=1 #radius of each helix circle

#just some thinking on other ways to define b by using HP
#L=N*np.sqrt((a**2)+(b**2)) #arc length
#L_total=k*L+(k-1)*HP #total length of slab
#A=1-(1/(k**2))
#B=(2*(k-1)*HP)/k
#C=(a**2)-((k-1)*HP/(N*k))**2
#b=np.sqrt(((B**2)-4*A*C)/(2*A))

b=L_total/N #since by definition of z it should be that for theta max (N) we should get z max
(L_total)

N = k*2*np.pi
theta= np.linspace(0, N, n)
x = a*np.cos(theta)
z =  b*theta
y =  a*np.sin(theta)
ax.plot(x, y, z, 'b', lw=2)


# An line through the centre of the helix
ax.plot((0,0), (0,0), (np.min(z), np.max(z)), color='k', lw=2)
# sin/cos components of the helix (e.g. electric and magnetic field
# components of a circularly-polarized electromagnetic wave
#ax.plot(x, y, 0, color='r', lw=1, alpha=0.5)
#ax.plot(x, [0]*n, z, color='m', lw=1, alpha=0.5)

plt.xlabel('mm')
plt.ylabel('mm')
# Remove axis planes, ticks and labels
#ax.set_axis_off()
plt.show()
```