

GNU OCTAVE CODE TO DESIGN HELICAL ANTENNA

```
function HelicalAntennaGUI

% Create figure and components

fig = figure('Position', [100, 100, 1000, 620], 'Name', 'Helical Antenna Generator', 'NumberTitle',
'off', 'Resize', 'off');

% Labels and Edit Fields

uicontrol('Style', 'text', 'String', ' Turns:', 'Position', [20, 520, 75, 22], 'HorizontalAlignment', 'right');

turnsEdit = uicontrol('Style', 'edit', 'Position', [130, 520, 100, 22]);

uicontrol('Style', 'text', 'String', 'Diameter', 'Position', [50, 485, 70, 22], 'HorizontalAlignment',
'right');

diameterEdit = uicontrol('Style', 'edit', 'Position', [130, 485, 100, 22]);

uicontrol('Style', 'text', 'String', 'Spacing:', 'Position', [50, 450, 70, 22], 'HorizontalAlignment',
'right');

spacingEdit = uicontrol('Style', 'edit', 'Position', [130, 450, 100, 22]);

uicontrol('Style', 'text', 'String', 'Radius:', 'Position', [50, 415, 70, 22], 'HorizontalAlignment', 'right');

radiusEdit = uicontrol('Style', 'edit', 'Position', [130, 415, 100, 22]);

uicontrol('Style', 'text', 'String', 'Frequency:', 'Position', [40, 380, 80, 22], 'HorizontalAlignment',
'right');

frequencyEdit = uicontrol('Style', 'edit', 'Position', [130, 380, 100, 22]);

% Generate Button

generateButton = uicontrol('Style', 'pushbutton', 'String', 'Generate Antenna', 'Position', [50, 320,
150, 30], 'Callback', @generateAntenna);

% Result Label

resultLabel = uicontrol('Style', 'text', 'String', '', 'Position', [20, 150, 300, 120], 'FontWeight', 'bold');
```

```
% 3D Plot
```

```
ax = axes('Parent', fig, 'Position', [0.35, 0.1, 0.6, 0.8]);
```

```
% Save handles in the figure's UserData
```

```
handles = guihandles(fig);
```

```
handles.turnsEdit = turnsEdit;
```

```
handles.diameterEdit = diameterEdit;
```

```
handles.spacingEdit = spacingEdit;
```

```
handles.radiusEdit = radiusEdit;
```

```
handles.frequencyEdit = frequencyEdit;
```

```
handles.generateButton = generateButton;
```

```
handles.resultLabel = resultLabel;
```

```
handles.ax = ax;
```

```
guidata(fig, handles);
```

```
% Make the figure visible
```

```
set(fig, 'Visible', 'on');
```

```
function generateAntenna(~, ~)
```

```
try
```

```
    % Retrieve handles from UserData
```

```
    handles = guidata(fig);
```

```
    % Get user input
```

```
    turns = str2double(get(handles.turnsEdit, 'String'));
```

```
    diameter = str2double(get(handles.diameterEdit, 'String'));
```

```
    spacing = str2double(get(handles.spacingEdit, 'String'));
```

```
    radius = str2double(get(handles.radiusEdit, 'String'));
```

```
    frequency = str2double(get(handles.frequencyEdit, 'String'));
```

```

% Calculate pitch angle (alpha)

pitchAngle = atan(spacing / (2 * pi * radius)) * (180 / pi);

% Calculate Half Power Beam Width (HPBW) and First Null Beam Width (FNBW)

wavelength = 3e8 / frequency; % Speed of light divided by frequency gives the wavelength

HPBW = (52 * (wavelength)^(1.5)) / ((2*pi*radius)*sqrt(turns * spacing));
FNBW = (115 * (wavelength)^(1.5)) / ((2*pi*radius)*sqrt(turns * spacing));

% Use the parameters to calculate something (this is just a placeholder)

length = turns * spacing;

% Generate 3D plot of the helical antenna

theta = linspace(0, turns * 2 * pi, 10000);
x = radius.* cos(theta);
y = radius.* sin(theta);
z = linspace(0, length, 10000);

plot3(handles.ax, x, y, z, 'LineWidth', 2);
grid(handles.ax, 'on');
xlabel(handles.ax, 'X-axis (mm)');
ylabel(handles.ax, 'Y-axis (mm)');
zlabel(handles.ax, 'Z-axis (mm)');
title(handles.ax, '3D Plot of Helical Antenna');

% Add labels for clarity

text(handles.ax, x(1), y(1), z(1), 'Start', 'HorizontalAlignment', 'right');
text(handles.ax, x(end), y(end), z(end), 'End', 'HorizontalAlignment', 'right');

```

```

    % Display the results

    hpbwText = sprintf('HPBW: %.2f degrees', HPBW);
    fnbwText = sprintf('FNBW: %.2f degrees', FNBW);
    pitchAngleText = sprintf('Pitch Angle (alpha): %.2f degrees', pitchAngle);
    set(handles.resultLabel, 'String', {hpbwText, fnbwText, pitchAngleText});

    % Display the result (this is just a placeholder, you'll replace this with your actual antenna
creation code)

    resultText = sprintf('Helical Antenna Length: %.2f mm', length);
    set(handles.resultLabel, 'String', {hpbwText, fnbwText, pitchAngleText, resultText});

catch

    % Handle invalid input

    errorText = 'Invalid input. Please enter numerical values.';

    set(handles.resultLabel, 'String', errorText);

end

end

end
end

```