To write a code using Python and Flask to read ion collection from an oscilloscope and plot a GUI graph between ion count, you would need to interface the oscilloscope with the Python code, which can be done using libraries like PyVISA or instrument drivers provided by the oscilloscope manufacturer. Here is a sample code to give you an idea of how to implement this:

```
from flask import Flask, render_template, json
import pyvisa
import time

app = Flask(__name__)

# Connect to the oscilloscope
rm = pyvisa.ResourceManager()
oscilloscope = rm.open_resource('GPIB0::22::INSTR')

# Query the oscilloscope for ion collection data
def get_ion_data():
    oscilloscope.write(':MEASure:SOURce CH1;:MEASure:DATA? ION')
    data = oscilloscope.read()
    return [float(x) for x in data.split(',')]

# Plot the ion data using Flask and Plotly
@app.route('/plot')
def plot():
    ion_data = get_ion_data()
    return render_template('plot.html', ion_data=ion_data)

if __name__ == '__main__':
    app.run(debug=True)
```

In this code, we use the PyVISA library to connect to the oscilloscope and query it for ion collection data. The data is then passed to a Flask template called `plot.html`, which uses the Plotly library to plot the data.

Here is an example of what the `plot.html` template might look like:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ion Collection Data</title>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
  </head>
  <body>
    <div id="plot"></div>
    <script>
      var data = [{
        x: [{{ ion_data|safe }}],
        y: [1, 2, 3, 4, 5],
        type: 'scatter'
      }];

      Plotly.newPlot('plot', data);
    </script>
  </body>
</html>
```