In [1]:
```
import scipy.io
import seaborn as sns
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
```

In [2]:
```
# Using os.path.join to create the path
data_dir = "C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        path = os.path.join(root, file_name)
        print(path)
```

```
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA007_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA014_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA021_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR007_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR014_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR021_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR007_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR014_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR021_1.mat
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\rN_1.mat
```

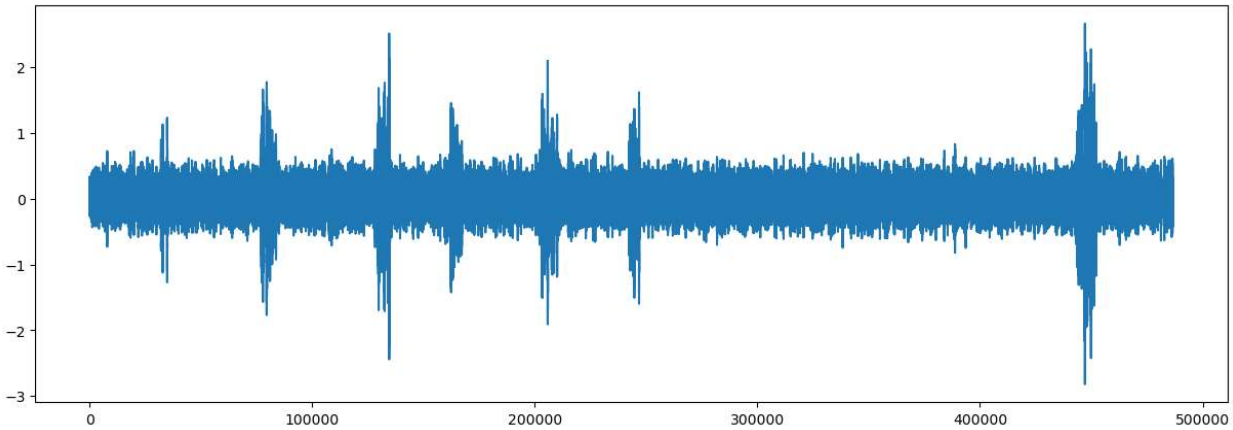
In [3]:
```
# Using f-strings for path
path = f'C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR021_1.mat'
mat = scipy.io.loadmat(path)
```

In [4]:
```
# Using mat.keys() directly
key_name = list(mat.keys())[3]
```

In [5]:
```
# Simplifying fault creation
fault = np.full((len(mat[key_name]), 1), file_name[:-4])
```

In [6]:
```
# Using DataFrame initialization directly
df_temp = pd.DataFrame({'DE_data': np.ravel(mat[key_name]), 'fault': np.ravel(fault)})
```

In [7]:
```
# Plotting directly without creating a variable
plt.figure(figsize=(15, 5))
plt.plot(df_temp.iloc[:, 0])
plt.show()
```



In [8]:
```
# Initializing df with data directly
df = pd.DataFrame(columns=['DE_data', 'fault'])
```

In [9]:
```
# Using f-strings for path
data_dir = "C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp"
for root, dirs, files in os.walk(data_dir, topdown=False):
    for file_name in files:
        # Check if the file has a .mat extension
        if file_name.endswith('.mat'):
            path = os.path.join(root, file_name)
            print(path)

            try:
                mat = scipy.io.loadmat(path)
                key_name = list(mat.keys())[3]
                DE_data = mat.get(key_name)
```

```
                # Simplifying fault creation
                fault = np.full((len(DE_data), 1), file_name[:-4])

                # Concatenating directly without creating df_temp
                df = pd.concat([df, pd.DataFrame({'DE_data': np.ravel(DE_data), 'fault': np.ravel(fault)})],
                print(df['fault'].unique())

            except Exception as e:
                print(f"Error processing file {file_name}: {e}")
```

```
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA007_1.mat
['BA007_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA014_1.mat
['BA007_1' 'BA014_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\BA021_1.mat
['BA007_1' 'BA014_1' 'BA021_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR007_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR014_1.mat
Error processing file IR014_1.mat: All arrays must be of the same length
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\IR021_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1' 'IR021_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR007_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1' 'IR021_1' 'OR007_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR014_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1' 'IR021_1' 'OR007_1' 'OR014_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\OR021_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1' 'IR021_1' 'OR007_1' 'OR014_1'
 'OR021_1']
C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp\rN_1.mat
['BA007_1' 'BA014_1' 'BA021_1' 'IR007_1' 'IR021_1' 'OR007_1' 'OR014_1'
 'OR021_1' 'rN_1']
```

In [10]:
```
# Save the resulting DataFrame to a CSV file
df.to_csv('C:/FAULT_DIAG_PROJ/CWRU_dataset/48k_drive_end/1hp/1hp_all_faults.csv', index=False)
```

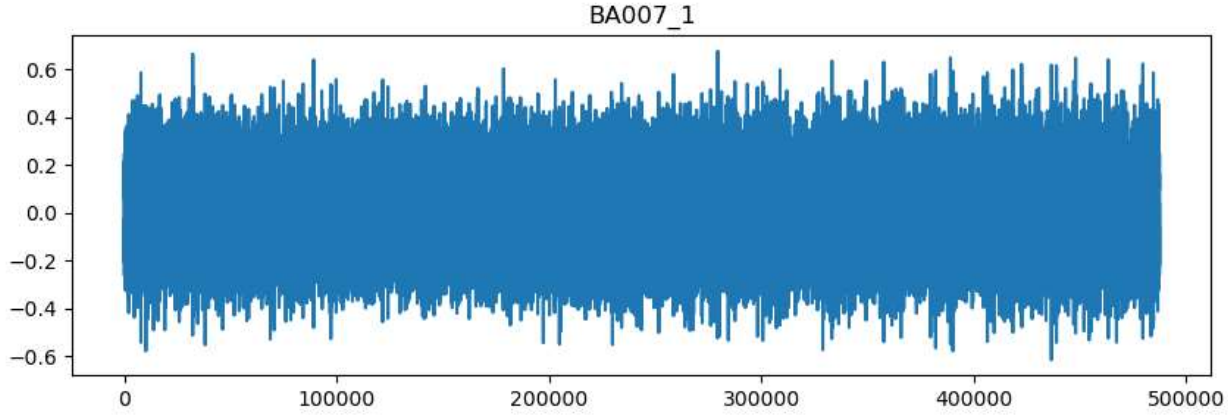
In [11]:
```
# Display the DataFrame
df
```

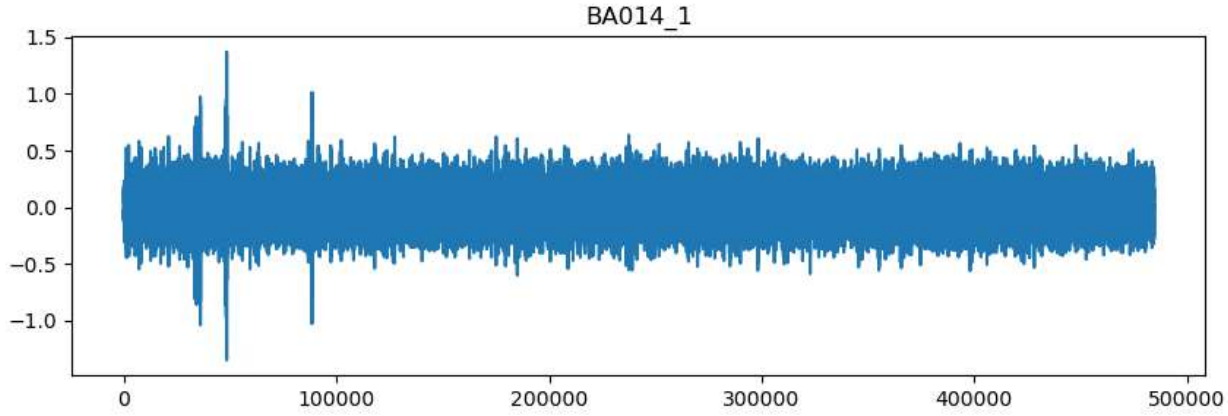
Out[11]:

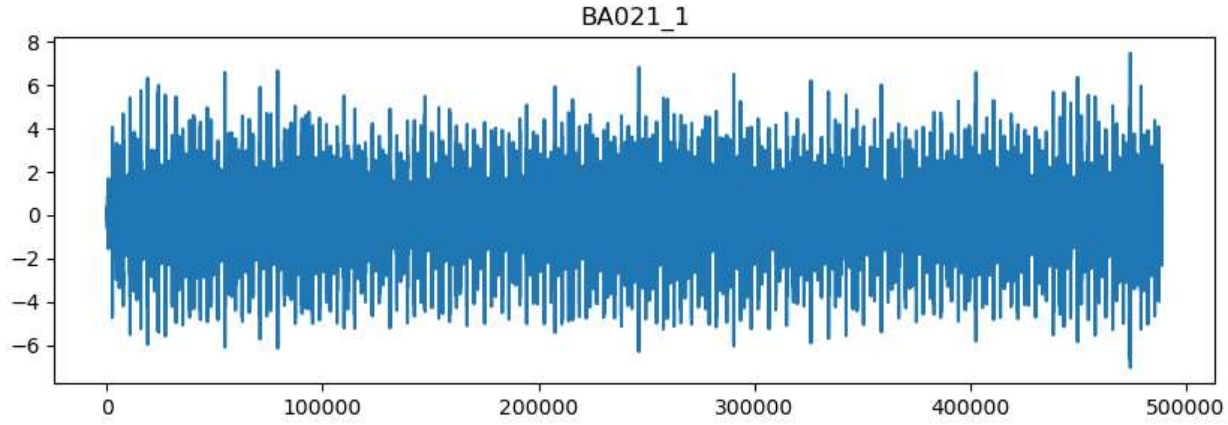
	DE_data	fault
0	-0.041097	BA007_1
1	-0.046104	BA007_1
2	-0.028372	BA007_1
3	0.001252	BA007_1
4	0.033796	BA007_1
...	...	...
483898	-0.043183	rN_1
483899	-0.067383	rN_1
483900	-0.099092	rN_1
483901	-0.108271	rN_1
483902	-0.070929	rN_1

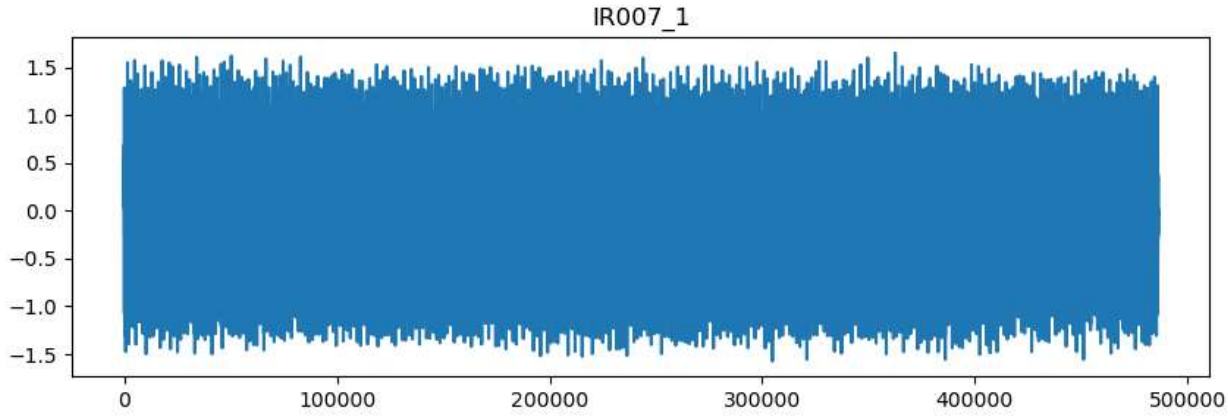
4376014 rows × 2 columns

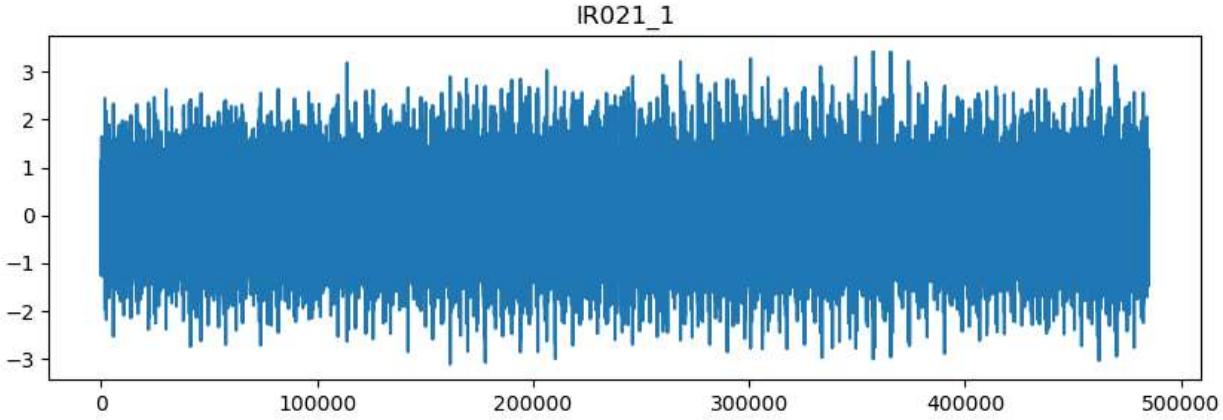
In [12]:
```
# Simplifying the faults loop
for f in df['fault'].unique():
    plt.figure(figsize=(10, 3))
    plt.plot(df[df['fault'] == f].iloc[:, 0])
    plt.title(f)
    plt.show()
```

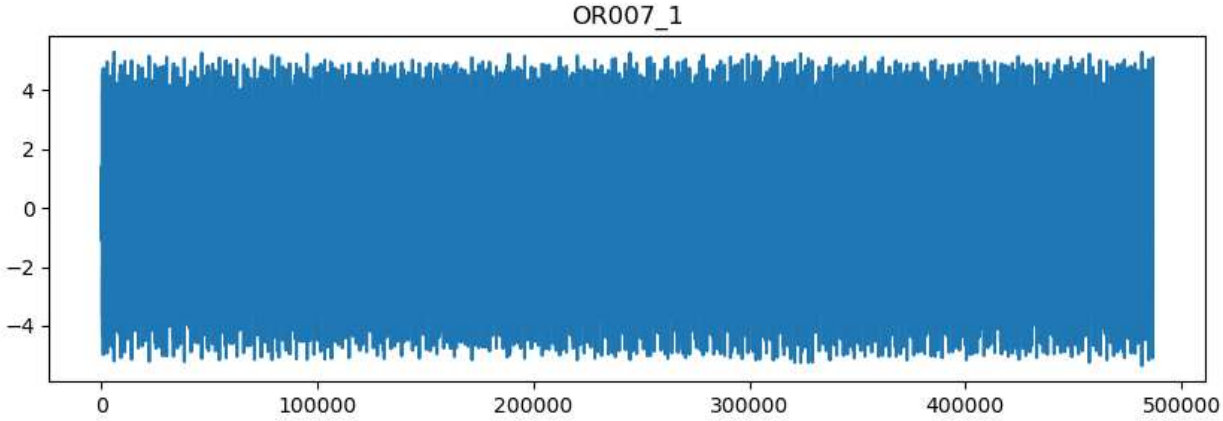

BA007_1
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
100000
200000
300000
400000
500000

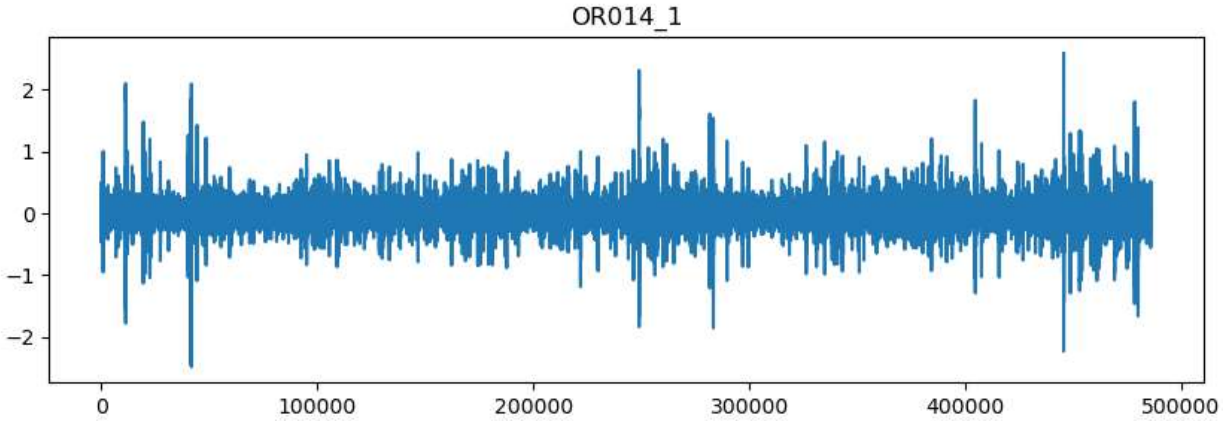

BA014_1
1.5
1.0
0.5
0.0
−0.5
−1.0
0
100000
200000
300000
400000
500000

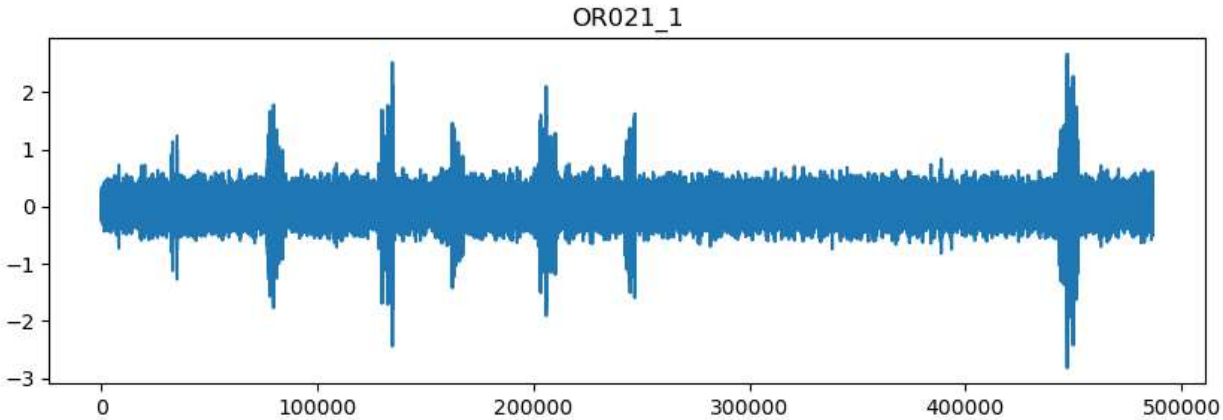

BA021_1
8
6
4
2
0
−2
−4
−6
0
100000
200000
300000
400000
500000

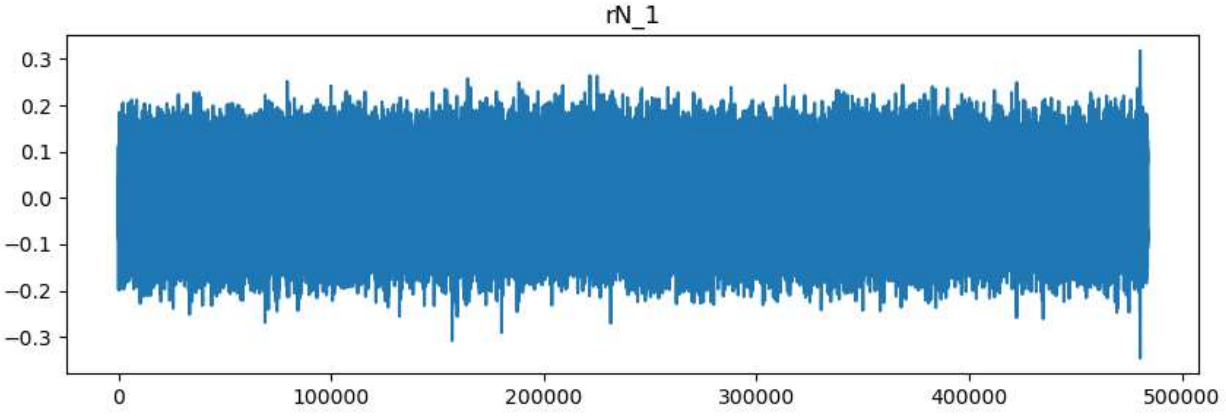

IR007_1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
100000
200000
300000
400000
500000


IR021_1
3
2
1
0
−1
−2
−3
0
100000
200000
300000
400000
500000

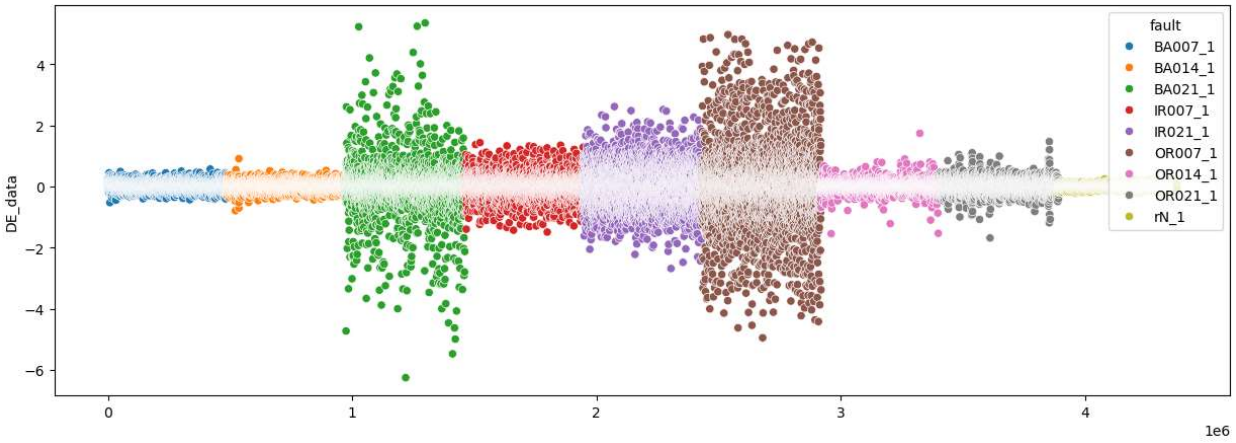

OR007_1
4
2
0
−2
−4
0
100000
200000
300000
400000
500000


OR014_1
2
1
0
−1
−2
0
100000
200000
300000
400000
500000


OR021_1
2
1
0
−1
−2
−3
0
100000
200000
300000
400000
500000



In [13]:
```
# Plotting directly without creating a variable
plt.figure(figsize=(15, 5))
sns.scatterplot(data=df.iloc[::100, :], y='DE_data', x=np.arange(0, len(df), 100), hue='fault')
plt.show()
```



In [ ]: