

## ОБРАБОТКА РЕЗУЛЬТАТОВ ИСПЫТАНИЙ НА НАДЁЖНОСТЬ ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ С ПОМОЩЬЮ БИБЛИОТЕК PYTHON

*Жданович В.П., Голубева И.А., Бруй Н.М.*

*Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь*

*Научный руководитель: Шнейдеров Е.Н. – канд.техн.наук*

**Аннотация.** При сборе большого количества данных об электрических параметрах приборов важно рационально осуществлять анализ получаемых значений. Выводы удобно делать при оценке графических представлений результатов, однако вручную строить графики для каждого параметра является нецелесообразным, в этом могут помочь библиотеки языка программирования Python.

**Ключевые слова:** обработка данных, анализ данных, испытания на надёжность.

**Введение.** В рамках проекта по прогнозированию эксплуатационной надёжности мощных полупроводниковых приборов были получены большие объёмы данных об измерении различных электрических параметров. Необходимо произвести первичную статистическую обработку значений.

**Основная часть.** Для момента времени  $t = 0$  (до начала эксплуатации) получены значения контролируемых электрических параметров исследуемых полупроводниковых приборов. Измерения электрических параметров проводились на изображённых на рисунке 1 сертифицированных измерительных установках Испытательного Центра Филиала «Транзистор» ОАО «ИНТЕГРАЛ» – управляющей компании холдинга «ИНТЕГРАЛ».



Рисунок 1 – Установки для измерения параметров полупроводниковых приборов

Результаты проведённых измерений электрических параметров транзисторов сведены в таблицы для визуализации. Измерения параметров проводились транзисторов двух типов, однако в целях сокращения объёма статьи будут приведены данные только о транзисторах типа КТ819В. Фрагменты таблиц измерений некоторых параметров для одного из используемых типов транзисторов приведены далее. С целью исключения случайных воздействующих факторов каждое измерение выполнялось 3 раза с временным промежутком между ними.

Таблица 1 – Значения электрических параметров КТ819В.

№	$U_{кн}$ (проверка контакта)				$I_{кэ}$				$U_{кн}$			
	Режим 100 мА		Пределы		Режим 30 В		Пределы		Режим 5 А		Пределы	
	1	2	0	8	1	2	0	200	1	2	0	1,9
				В				мкА				В
				СР								СР
1	0,101	0,100	0,099	0,1000	1,00	1,00	1,00	1,00	0,778	0,770	0,764	0,7707
2	0,100	0,103	0,099	0,1007	1,00	0,90	-0,30	0,53	0,782	0,772	0,770	0,7747
3	0,118	0,123	0,120	0,1203	0,80	0,80	0,90	0,83	0,806	0,799	0,796	0,8003
4	0,100	0,117	0,114	0,1133	-0,40	0,90	-0,30	0,07	0,788	0,786	0,783	0,7857
5	0,101	0,102	0,100	0,1010	0,90	-0,40	-0,40	0,03	0,770	0,763	0,759	0,7640
...												
200	0,121	0,121	0,116	0,1193	1,00	1,00	0,70	0,90	0,824	0,820	0,817	0,8203

По данным таблиц необходимо построить графики для начальной статистической обработки полученных значений. Для этого воспользуемся библиотеками Scikit-learn и pandas для Python. Для визуализации данных используем библиотеку Matplotlib и основанную на ней библиотеку seaborn. Использование данных библиотек позволяет сосредоточиться на статистических данных, используя готовые решения и настраивая базовые параметры рисунка [1].

Приведём код для построения гистограммы значения электрического параметра транзисторов для трёх измерений.

```
def histplot(data, param, unit):
    fig, axs = plt.subplots(1, 3, figsize = (14, 4))
    fig.suptitle('Значения {param}, {unit}'.format(param=param, unit=unit))
    sb.histplot(data=data, x='first', bins=10, kde=True, ax=axs[0])
    sb.histplot(data=data, x='second', bins=10, kde=True, ax=axs[1])
    sb.histplot(data=data, x='third', bins=10, kde=True, ax=axs[2])
    axs[0].set_title('Первое измерение')
    axs[1].set_title('Второе измерение')
    axs[2].set_title('Третье измерение')
    for ax in axs:
        ax.set_xlabel(unit)
        ax.set_ylabel('Количество, шт.')
```

Приведём код для построения диаграмм рассеяния каждого из измерений значения электрического параметра транзисторов.

```
def scatplot(data, param):
    fig, axs = plt.subplots(1, 3, figsize = (14, 4))
    fig.suptitle('Зависимости измерений {param} друг от друга'.format(param=param))
    x = np.linspace(data['first'].min(), data['first'].max(), 50)
    sb.scatterplot(data=data, x='first', y='second', edgecolors='white', alpha=0.9, s=40, ax=axs[0])
    sb.scatterplot(x = x, y = x, color='red', alpha=0.7, s=10, ax=axs[0])
    sb.scatterplot(data=data, x='first', y='third', edgecolors='white', alpha=0.9, s=40, ax=axs[1])
    sb.scatterplot(x = x, y = x, color='red', alpha=0.7, s=10, ax=axs[1])
    x = np.linspace(data['second'].min(), data['second'].max(), 50)
    sb.scatterplot(data=data, x='second', y='third', edgecolors='white', alpha=0.9, s=40, ax=axs[2])
    sb.scatterplot(x = x, y = x, color='red', alpha=0.7, s=10, ax=axs[2])
```

На рисунке 2 приведён пример статистической обработки значений электрического параметра  $U_{обр}$  выборки из 200 шт. транзисторов КП744А.

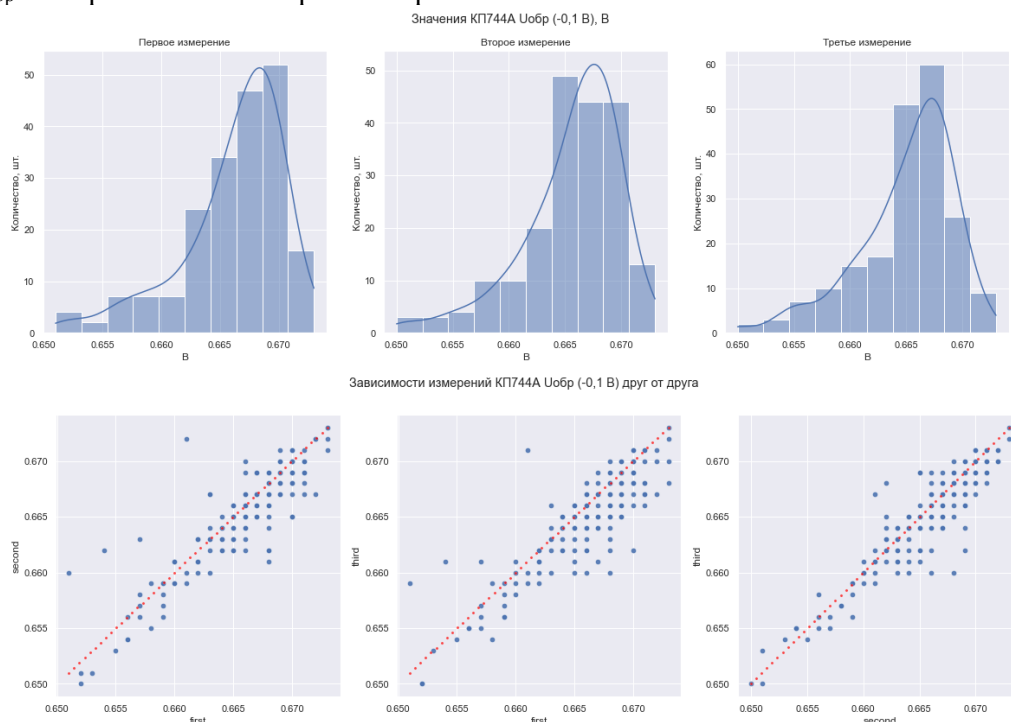


Рисунок 2 – Гистограммы распределения параметра  $U_{кн}$  (первая строка), диаграммы рассеяния измерений (вторая строка)

Во втором ряду (рисунок 2) в виде точечных полей изображены отклонения каждого измерения от среднего. Смысл их в том, что чем ближе точка измерения к диагональной линии, тем меньше разница между значениями разных измерений, тем больше вероятность отсутствия случайных факторов.

**Заключение.** Библиотеки Python позволяют удобно находить и визуализировать зависимости электрических параметров для последующего анализа.

### Список литературы

1. Matplotlib: Python plotting – Matplotlib 3.3.4 documentation [Electronic resource]. – Mode of access: <https://matplotlib.org/>. – Date of access: 18.03.2021.

UDC 621.382.33–027.45

## THE RESULTS OF RELIABILITY TESTS OF SEMICONDUCTOR DEVICES ANALYSIS USING PYTHON LIBRARIES

Zhdanovich V.P., Golubeva I.A., Brui N.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Shneiderov E.N. – PhD in Technical Science

**Annotation.** Collecting a large amount of data on the electrical parameters of devices, it is important to analyze the obtained values rationally. Conclusions can be done easily by a graphical presentation of the results; however, it is impractical to manually build graphs for each parameter. Python programming language libraries can help in this.

**Keywords:** data processing, data analysis, reliability tests.