

Как пользоваться скриптом для построения графиков для лаб по электронике

Содержание

1	Что это?	1
2	Перед работой	2
3	Как пользоваться?	3
3.1	Получение файла с координатами	3
3.2	Построение графиков скриптом	5
3.3	Сохранение построенных графиков	6
4	Примечания	6

1. Что это?

Это руководство пользователя скрипта, который строит графики для лаб по электронике. Он работает, считывая данные файла, сгенерированного графопостроителем Воркбенча, и по его содержанию строит нужные графики. То есть по такой цепочке:

1. Воркбенч
2. Файл с координатами графиков
3. Скрипт
4. Графики

2. Перед работой

Для правильной работы скрипта необходимы:

1. Рабочий Electronics Workbench (тут использовалась версия 5.12).
2. Рабочий Python 3 с поддержкой Tk (если во время установки Python вы нажали «Install Now», он у вас есть).
3. Два установленных модуля для Python:
 - a) `matplotlib`
 - b) `numpy`

Если Python не установлен, его надо скачать и установить с официального сайта (<https://www.python.org/ftp/python/3.6.3/python-3.6.3.exe>). При установке установить галочку «Add Python 3.6 to PATH» и нажать кнопку «Install Now».

Желательно иметь Python в %PATH%, чтобы обращаться к нему прямо, не указывая полного пути к интерпретатору. Чтобы проверить, есть ли Python в %PATH%, достаточно в командной строке ввести команду из листинга 1.

```
python --version
```

Листинг 1: Команда для проверки наличия Python

Если Python установлен и указан в переменной среды %PATH%, вывод будет примерно таким:

```
Python 3.5.1
```

Так как `matplotlib` и `numpy` не идут в комплекте стандартной поставки Python, их надо установить. Для этого в командной строке запустим команду из листинга 2.

```
pip install matplotlib numpy
```

Листинг 2: Команда для установки необходимых модулей

Если модули установились успешно и все остальные необходимые составляющие на месте, скрипт готов к работе.

3. Как пользоваться?

3.1. Получение файла с координатами

Для начала необходимо получить файл с координатами графиков. Для этого в Воркбенче открываем лабу и выполняем всё по тексту лабы до тех пор, пока на осциллографе не будет показана нужная осциллограмма (рис. 1).

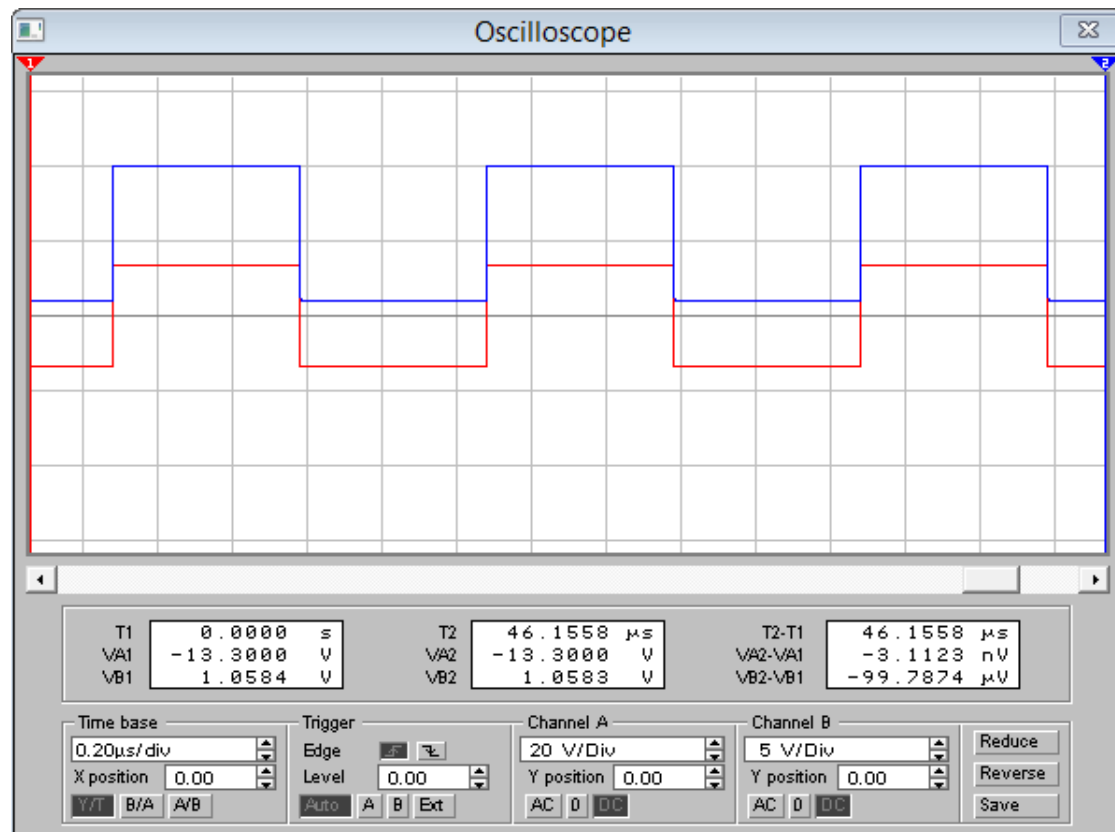


Рис. 1: На экране осциллографа появилась осциллограмма

Скрипту не важны никакие настройки масштаба (Time base, Channel A и Channel B) и положение визирных линий, так как он работает непосредственно с полученными данными. Главное, чтобы был правильно выставлен режим работы (Y/T, Auto, DC, DC) и сама схема была настроена так, как описано в тексте лабы.

Далее, чтобы получить файл с координатами, открываем графопостроитель и с его помощью экспортируем файл. Для этого нажимаем на кнопку «Display Graphs» (тулбар в главном окне, пиктограмма с двумя графиками). На экране откроется окно графопостроителя (рис. 2).

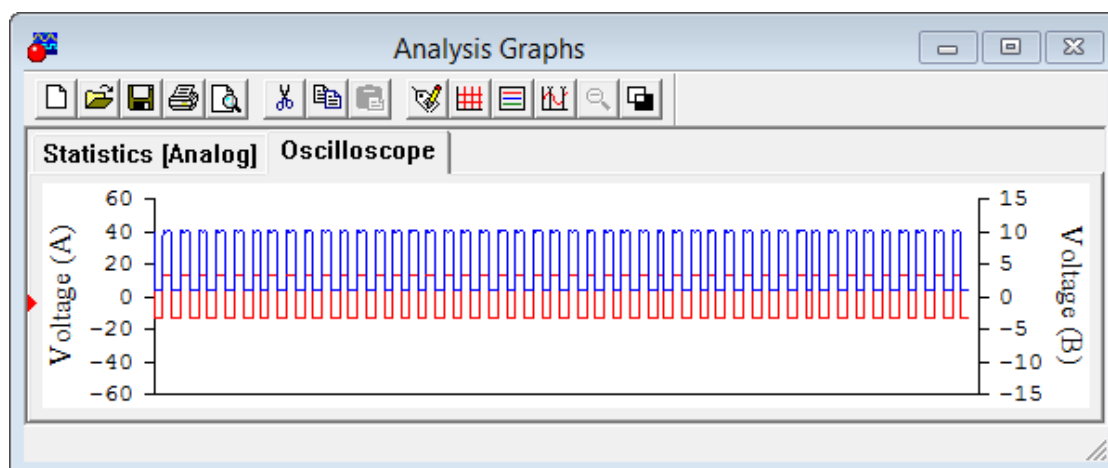


Рис. 2: Окно графопостроителя

Затем, в окне графопостроителя, нажимаем на кнопку «Save As» (дискета), и сохраняем файл в удобном месте (рис. 3). Желательно, в той же папке, что и сам скрипт. *Внимание!* Очень важно, чтобы тип файла был именно «Graph Data (.txt)», потому что скрипт писался для обработки именно этого типа файла.

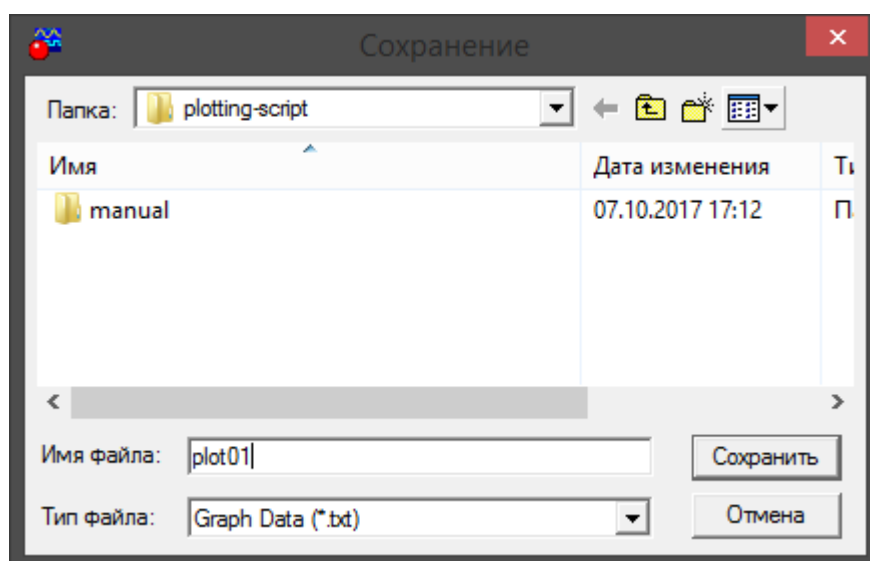
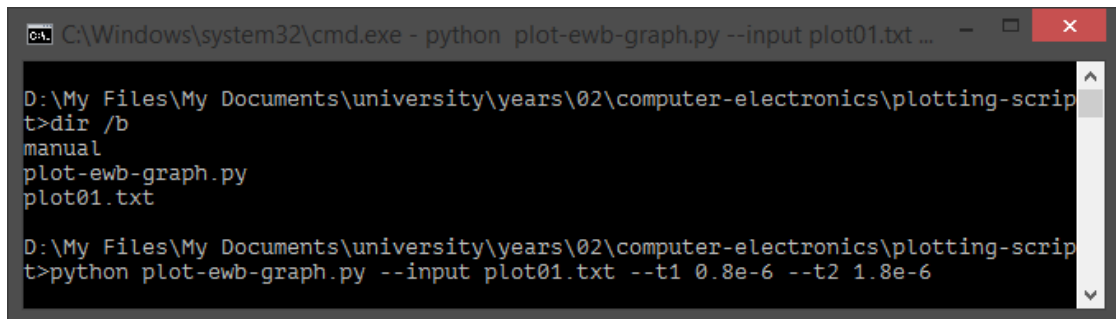


Рис. 3: Параметры сохранения

После сохранения в выбранной папке появится текстовый файл (в данном случае, plot01.txt). Именно этот файл нужен для построения графиков с помощью скрипта.

3.2. Построение графиков скриптом

Теперь необходимо передать полученный файл скрипту на обработку. Для этого открываем командную строку и переходим в папку, которая содержит сам скрипт и файл с графиком. Затем, выполняем команду с рис. 4.



```
C:\Windows\system32\cmd.exe - python plot-ewb-graph.py --input plot01.txt ...  
D:\My Files\My Documents\university\years\02\computer-electronics\plotting-scrip  
t>dir /b  
manual  
plot-ewb-graph.py  
plot01.txt  
D:\My Files\My Documents\university\years\02\computer-electronics\plotting-scrip  
t>python plot-ewb-graph.py --input plot01.txt --t1 0.8e-6 --t2 1.8e-6
```

Рис. 4: Запуск скрипта

После запуска скрипта должно появиться окно с построенными графиками (рис. 5).

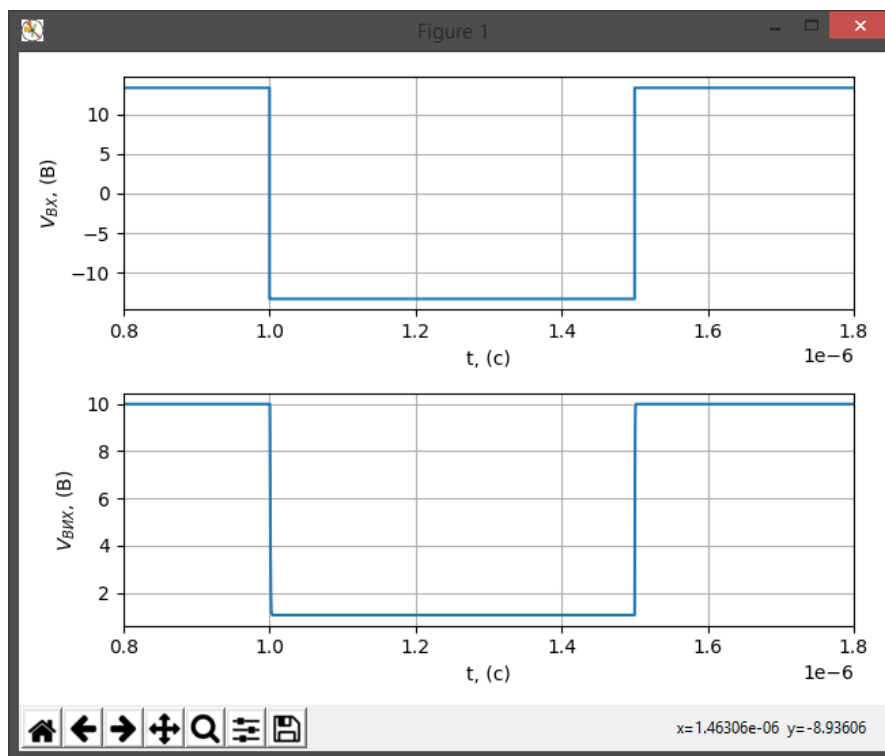


Рис. 5: Окно с построенными графиками

Скрипт запускается командой по типу листинга 3. Интерпретатор Python на Windows почему-то не хочет сразу подхватывать скрипт, поэтому скрипт необходимо запускать, указывая имя интерпретатора перед именем скрипта.

```
python plot-ewb-graph.py --input plot01.txt --t1 0.8e-6 --  
t2 1.8e-6
```

Листинг 3: Пример команды для построения графиков скриптом

Рассмотрим параметры, которые были указаны, чтобы получить графики с рис. 5:

1. `python plot-ewb-graph.py` — запуск скрипта.
2. `--input plot01.txt` — в качестве входного файла использовать `plot01.txt`.
3. `--t1 0.8e-6 --t2 1.8e-6` — построить графики для временного промежутка от $t_1 = 0.8 \times 10^{-6}$ с до $t_2 = 1.8 \times 10^{-6}$ с.

Временные параметры нужно выбирать так, чтобы график показывал как минимум один полупериод, то есть чтобы потом было удобно показать $t_{зд}$, $t_{НР}$, t_P , $t_{СП}$. Если вдруг параметры были выбраны неправильно, можно нажать на кнопку с четырьмя стрелочками и подвигать любой из графиков. Таким образом можно узнать удобный временной промежуток и перезапустить скрипт с новыми параметрами.

3.3. Сохранение построенных графиков

Для сохранения графиков достаточно в окне с построенными графиками нажать на кнопку «Save the figure». При сохранении можно выбрать любой удобный формат: JPEG, PNG, SVG и т.д. Советую сохранять в векторном формате (SVG), чтобы график можно было масштабировать без потерь и сжатия.

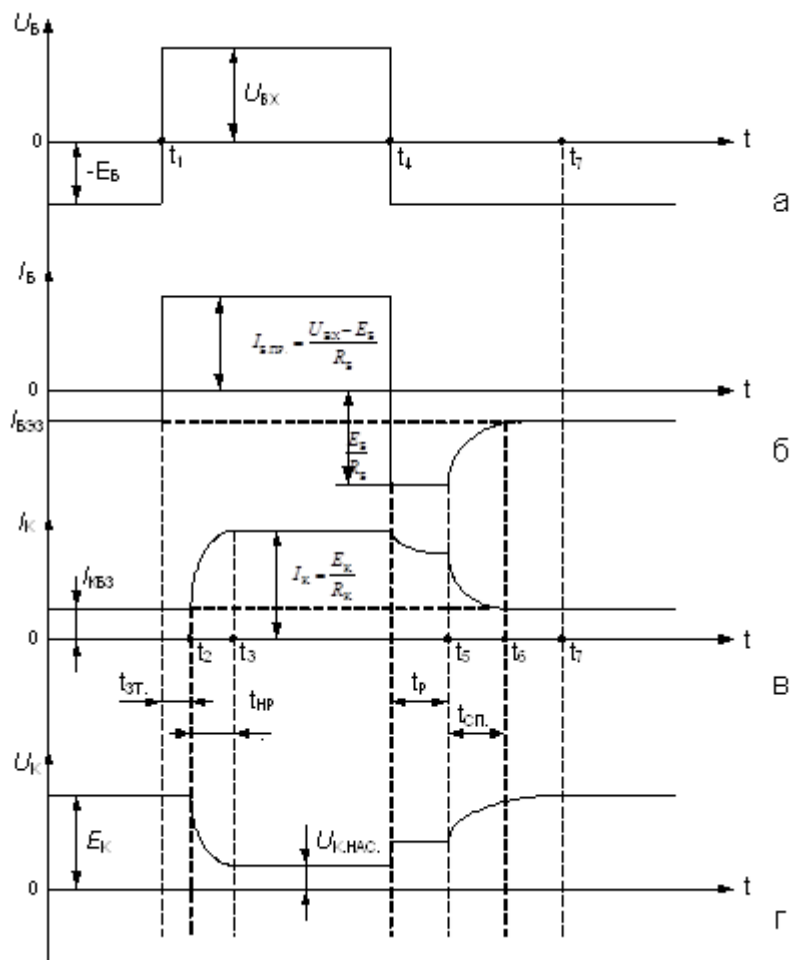
4. Примечания

Чтобы строить графики с другими настройками электрической схемы, наверное, удобнее всего будет каждый раз закрывать лабу, выставлять нужные настройки, строить осциллограмму по новой и сохранять в отдельный

файл. Это делается для того, чтобы не запутаться в том, какая диаграмма для каких настроек.

Допустим, открыли лабу и настроили на напряжение 10 В без подключенного диода. Построили осциллограмму, получили файл, назвали его 1.5v-nodiode.txt, построили график, экспортировали его. Потом закрыли лабу и настроили на то же напряжение, но теперь с диодом. Построили осциллограмму, получили файл, назвали его 1.5v-diode.txt, построили график, экспортировали его. И так для каждого нужного значения.

На полученных диаграммах нужно обозначить $t_{зд}$, $t_{нр}$, t_p , $t_{сп}$ по примеру с рис. 4 (взял из методички).



Единицы измерения временной оси OX — секунда, но на графике отмечены значения 0.8, 1.0, 1.2 и т.д. Это не секунды, а *микросекунды*. Дело в том, что на оси стоит обозначение $1e-6$, что означает, что любое обозначенное значение на оси OX мы должны умножить на 1×10^{-6} . Таким обра-

зом получаем микросекунды. Иначе говоря, что бы там препода не говорил, масштаб правильный, просто надо уметь читать обозначения. Или можно в любом редакторе закрасить $1e-6$, а на оси изменить t , (с) на t , (мкс).

Как и во всех нормальных графиках, на координатных осях нет стрелочек. Направление оси очевидно по значениям. Возможно, преподау это не понравится — можно дорисовать.