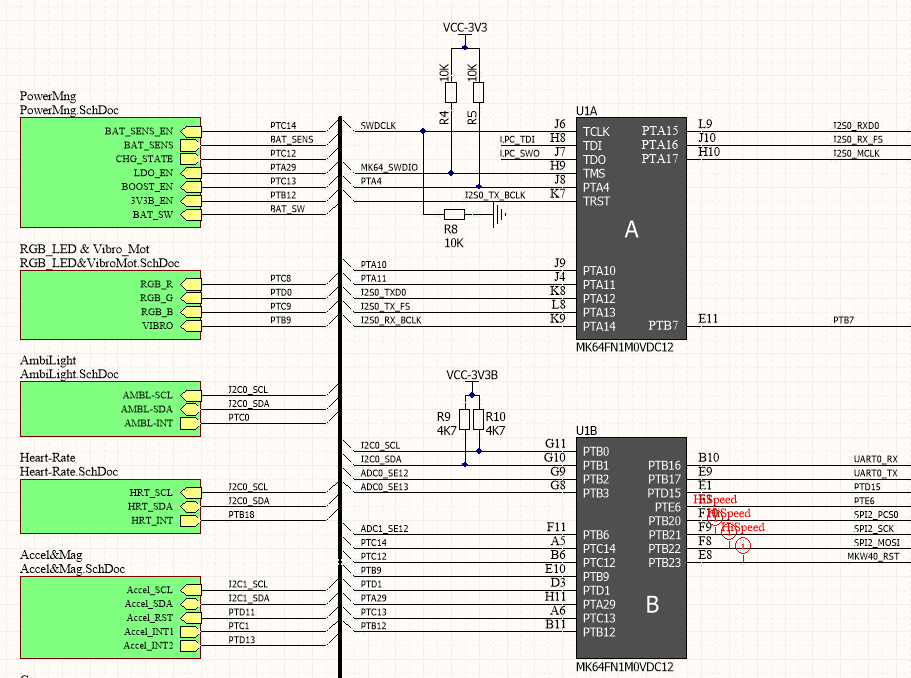


Несмотря на то, что Altium Designer поставляется с огромными библиотеками компонентов по-прежнему остается необходимость создания в нем своих схемных компонентов. Особенно это актуально для крупных компонентов с большим количеством выводов и атрибутов выводов. Это могут быть FPGA, микроконтроллеры, процессоры, чипы памяти и т.д. Здесь я представлю свою технологию генерации компонентов экстрагируя информацию из PDF файлов.

<cut />

Возьмем для примера даташиты на микроконтроллеры Kinetis, скажем серию K66. Нет труда извлечь схемные компоненты этих микроконтроллеров из многочисленных референс-дизайнов предоставляемых фирмой производителем. К счастью многие из них представлены в формате Altium Designer. Скачиваем [отсюда](http://www.nxp.com/products/reference-designs/hexiwear-complete-iot-development-solution:HEXIWEAR?tab=Design_Tools_Tab) архив "Hexiwear-Design-Files", находим там схему, а в ней вот такое представление компонента:



Здесь схемный компонент микроконтроллера разбит на несколько логических частей, так как было удобно автору схемы. Остается только завидовать профессионализму автора, но обычно такое скупое представление компонента затрудняет понимание работы схемы, а в последствии и поиск ошибок.

Микроконтроллер на своих выводах может поддерживать до 7-и альтернативных функций. Ошибись схемотехник в назначении функции хотя бы одного вывода и плату придется мучительно тюнинговать вручную после изготовления или даже выкинуть, если корпус - BGA.

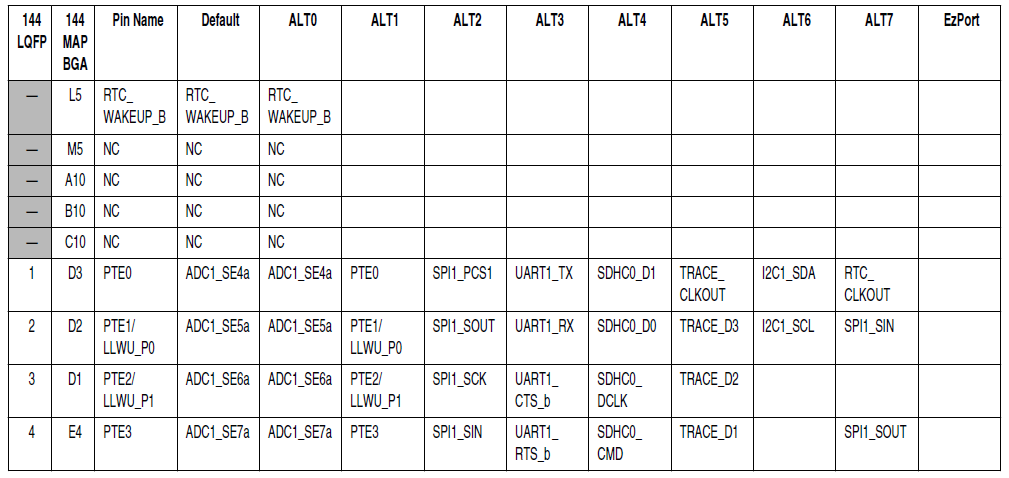
Поэтому такой компонент мы не можем позаимствовать. Он к тому же представлен только для одного корпуса, а корпуса могут быть и другие, с другой распиновкой.

Не лучше обстоят дела и у компонентов микроконтроллеров, найденных в сторонних библиотеках. В них также не указываются альтернативные функции.

Я нашел выход в автоматизации генерации компонентов из pdf даташитов.

**Шаг 1.**

Определяем какими таблицами в даташите представлена распиновка. Для K66 она представлена в виде такой таблицы простирающейся на несколько листов.

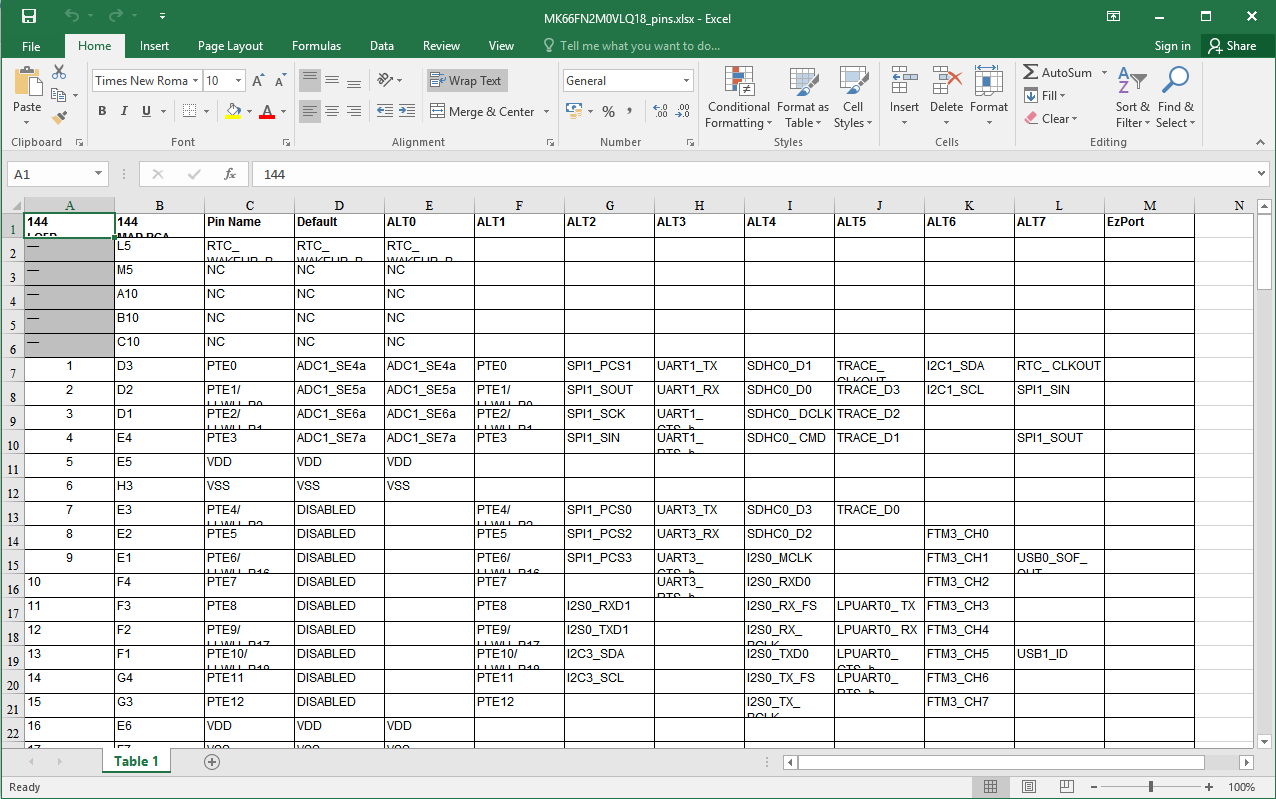


Это удобное представление. В этой таблице сразу сведены и номера выводов и названия всех их функций. Но скажем, для микроконтроллеров STM32 ситуация будет сложнее, там есть отдельно таблица соответствия номеров выводов их базовым названиям и таблица соответствия базовых названий и всех альтернативных функций. Это тоже несложно решаемо.

**Шаг 2.**

Из PDF файла переносим таблицы в MS Excel. Я использовал для этого программу Adobe Acrobat. У нее существует бесплатная триальная версия.

Получаем таблице в Excel такого вида:



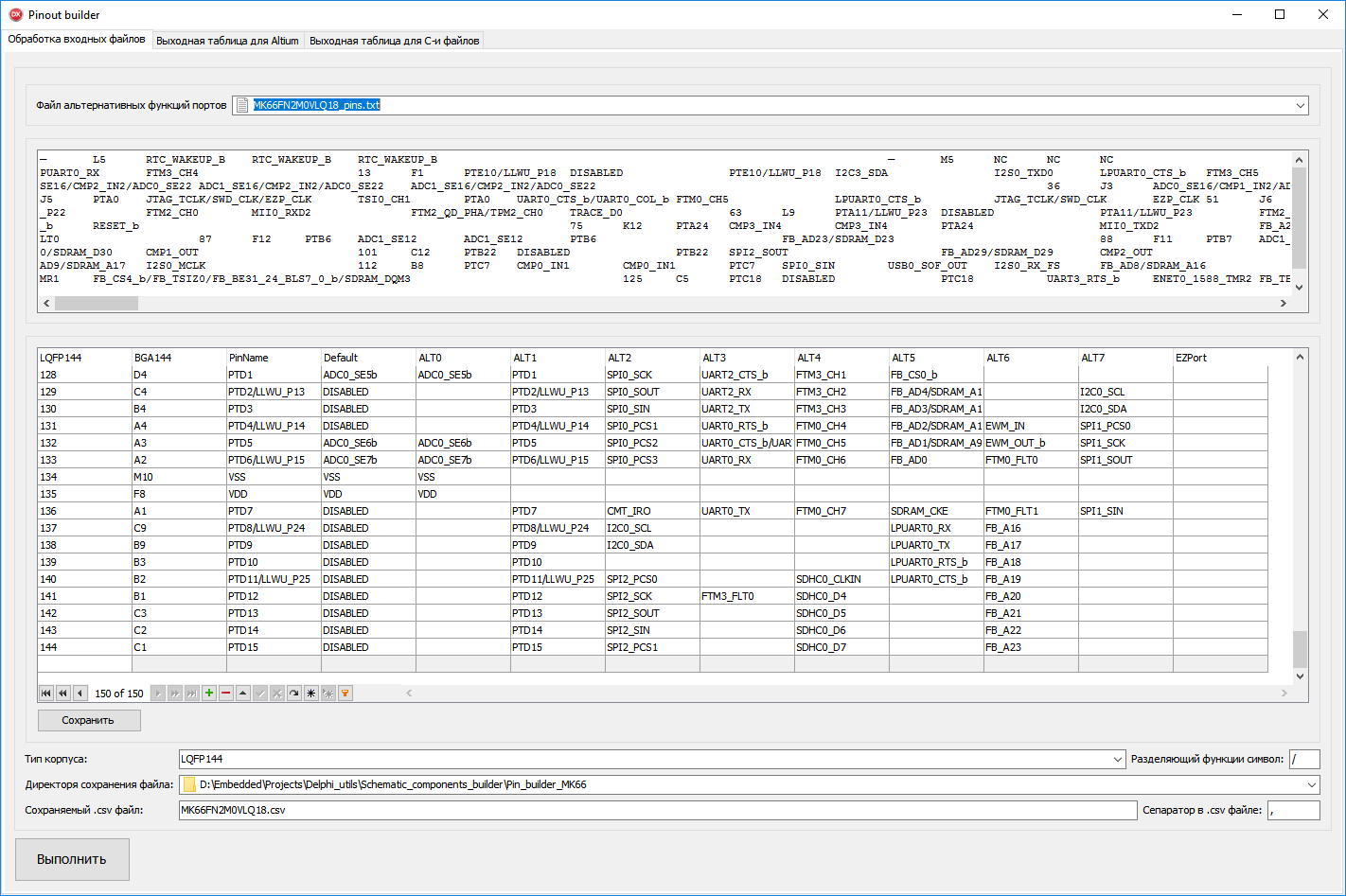
**Шаг 3.**

Экспортирую таблицу из Excel в текстовый файл где поля таблицы разделены символом табуляции (0x09).

**Шаг 4.**

Полученный нами файл таблицы наполнен всяческим мусором, унаследованным от форматирования в PDF файле. Это и ненужные переносы строк, и пробелы, и другие ненужные символы.

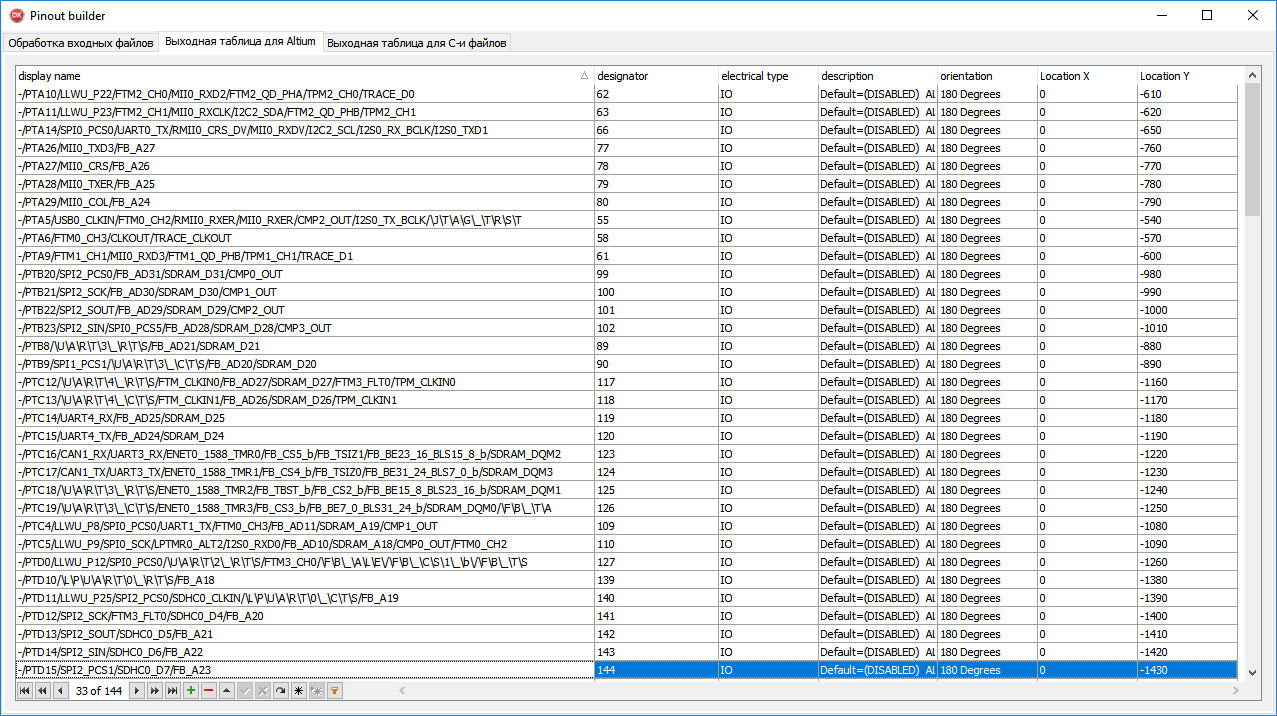
Поэтому я написал программу в Delphi которая импортирует файл и фильтрует мусор.



В окне программы указывается путь к файлу альтернативных функций портов (это экспортированная из Excel наша таблица), указывается тип корпуса микросхемы (список заполнен в программе на Delphi вручную), указывается директория и файл куда будет сконвертирована таблица в формат пригодный для последующего импорта в Altium (это должен быть файл с расширением .csv). Сепаратором для csv файла должна быть запятая. А разделяющий функции символ может быть произвольный, такой чтобы удобно читались перечисления функций в описании вывода.

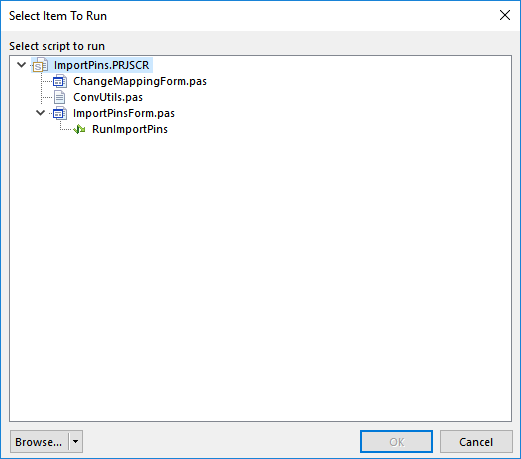
Все правильно настроив нажимаем "Выполнить".

После успешного выполнения увидим в закладке "Выходная таблица для Altium" таблицу, предназначенную для обработки скриптом Altium Designer. Таблица сохранена в указанном ранее csv файле.

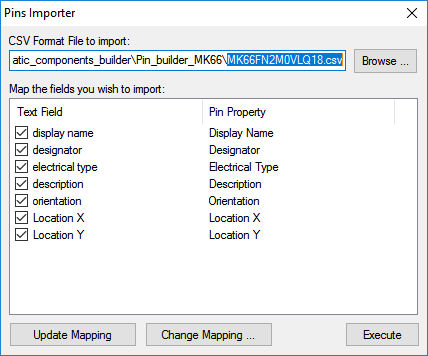


**Шаг 5.**

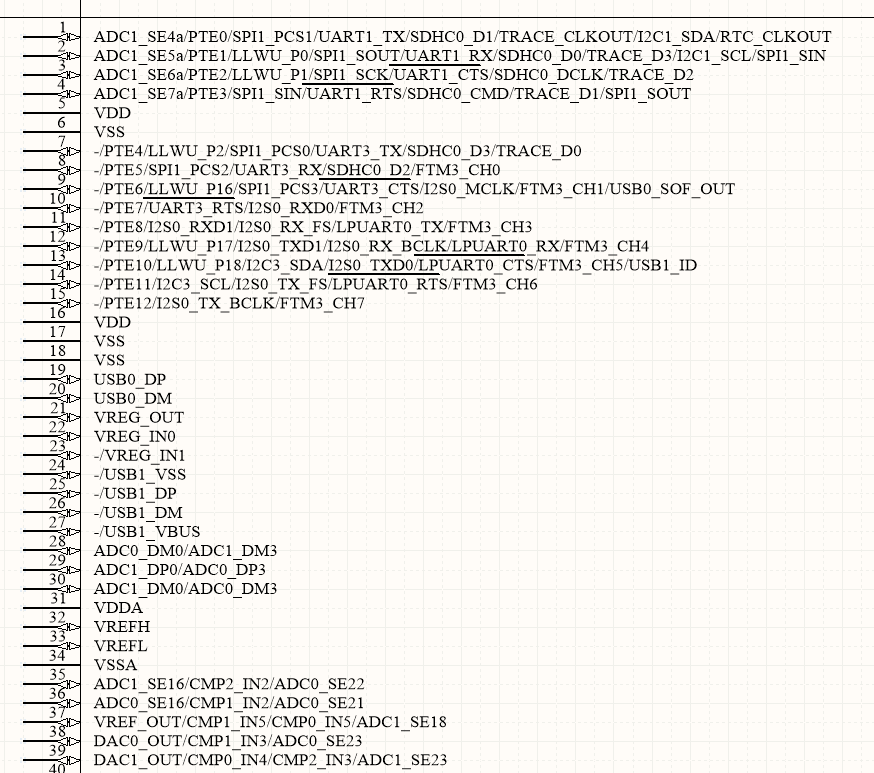
Открываем Altium Designer. Открываем библиотеку схемных компонентов где хотим создать новый компонент. Щелкаем последовательность DXP -> Run Script. Указываем путь к скрипту ImportPins.PRJSCR. Появляется такое окно:



В нем щелкаем RunImportPins. В появившемся диалоге указываем путь к нашему csv файлу и щелкаем Update Mapping. Получаем окно со следующим содержанием.



Щелкаем Execute и в окне редактора схемного компонента получаем изображение всех выводов с присвоенными им названиями и номерами:



*Все! Работа сделана.*

Может показаться, что это немного не то, что ожидалось от обещанной автоматизации. Но на самом деле точное соответствие номеров и названий выводов это самая ответственная и тяжелая работа при разработке схемных компонентов.

Далее можно вполне быстро отредактировать длину выводов, расположить их по полю, нарисовать контуры, разделить на функциональные группы и т.д. Это рутинная легкая работа, которая не приведет к фатальным ошибкам. Далее в редакторе футпринтов также без проблем сделать корпус микросхемы используя имеющиеся визарды. Поскольку корпуса у микроконтроллеров достаточно типовые.

Репозиторий проекта находится [здесь](https://github.com/Indemsys/Schematic_components_builder).

В директории Pin\_builder\_MK66 находятся все файлы для повторения шагов и их результаты, описанные в этой статье. Там же исходные файлы конвертера на Delphi. В директории Import\_pins\_Altium\_script находится проект скрипта для Altium Designer.

В файле FunctionsMapping.xlsx содержится исходная таблица, экспортированная из даташита.

csv файл для конвертации называется MK66FN2M0VLQ18.csv