

Tutorial relevant for users "Kompas-3D".

I don't know if they "Kompas-3D" is used in other countries, but I know "Kompas-3D" most popular in Russia, so further explanation will be in Russian. If you need English version, please comment tutorial, and I make translate.

Вместо предисловия

В данной заметке рассмотрено построение параметрической модели на примере корпуса микросхем типа QFP, модель строится в САПР «Компас-3D».

Я ни в коем случае не позиционирую себя как знатока работы в механических САПР, представленный ниже способ - лишь один из многих возможных. И я искренне надеюсь, что кому-то он поможет лучше разобраться 3D моделированием, а кто-то увидит, как оптимизировать процесс. Несмотря на то, что построение модели выполнялось в «Компас-3D», подобные механизмы могут присутствовать и в иных системах, так что возможно пригодится пользователям другого ПО.

Описанный ниже способ предполагает наличие определенного навыка работы с 3D моделями, и не содержит подробного описания последовательностей нажатия клавиш, скорее описана методика построения и взаимосвязь объектов.

Построение параметрической модели

Многие из нас сталкивались с ситуацией, когда необходимо создать много однотипных моделей, или же на этапе разработки постоянно надо вносить изменения в размеры или форму детали. Первое, что приходит в голову - сидеть и перерисовывать бесконечно одну модель. Но мы так делать не будем. САПР "Компас-3D" имеет механизм задания параметров через переменные, и при этом неплохо ладит с математикой, что позволяет вносить изменения нажатием пары клавиш. Однако, практика показала, что многие пользователи игнорируют сочетание параметризации с другими возможностями среды, что в итоге вызывает сложности при параметризации чего-то сложнее длины болта. Так попробуем же усложнить модель?

А попробуем мы создать модель корпуса для микросхем типа QFP. У этого семейства корпусов может быть разное количество выводов, разный шаг между ними, и конечно же разные габаритные размеры. Очень удобная для демонстрации модель, можно показать метод, но не лезть в дебри. Поехали.

Сначала можно построить сам пластиковый корпус, в котором смонтирован кремниевый кристалл. Так как модель планируется применять при работе в Altium Designer, то для удобства восприятия в меню "ориентация" выберем "Изометрия YZX", в таком положении плоскость, образованная осями X и Y оказывается в горизонтальном положении, Z направлена вверх. Плоскость XY является плоскостью платы для Altium Designer.

Так как сам корпус немного приподнят над плоскостью платы и стоит на выводах, мы построим вспомогательную плоскость, параллельную плоскости XY, на расстоянии 0,1 мм. в положительном направлении по оси Z, и уже от нее будем строить тело корпуса. Так как корпус квадратный, и стороны его одинаковы, можем задать длину стороны одной переменной: **body_width**. В построенной вспомогательной плоскости создаем эскиз, в нем чертим прямоугольник, и удобнее сделать это так, чтобы начало координат оказалось внутри него. Далее в меню "Размеры" выбираем размер от прямой до точки. Проставляем расстояние от каждой стороны прямоугольника до точки начала координат, при этом в поле "Выражение" во всех четырех случаях прописываем **(body_width/2)**, так мы получаем для всех вариантов ориентацию относительно центральной точки и более удобную привязку модели.

Теперь нам надо задать объем корпуса, для чего выдавливаем полученный эскиз вверх на расстояние, равное **(height-0.1)**, где **height** - высота всей итоговой детали над уровнем печатной платы, а **-0.1** необходим для компенсации зазора под корпусом. Заготовка пластикового корпуса готова (см. рисунок 1).

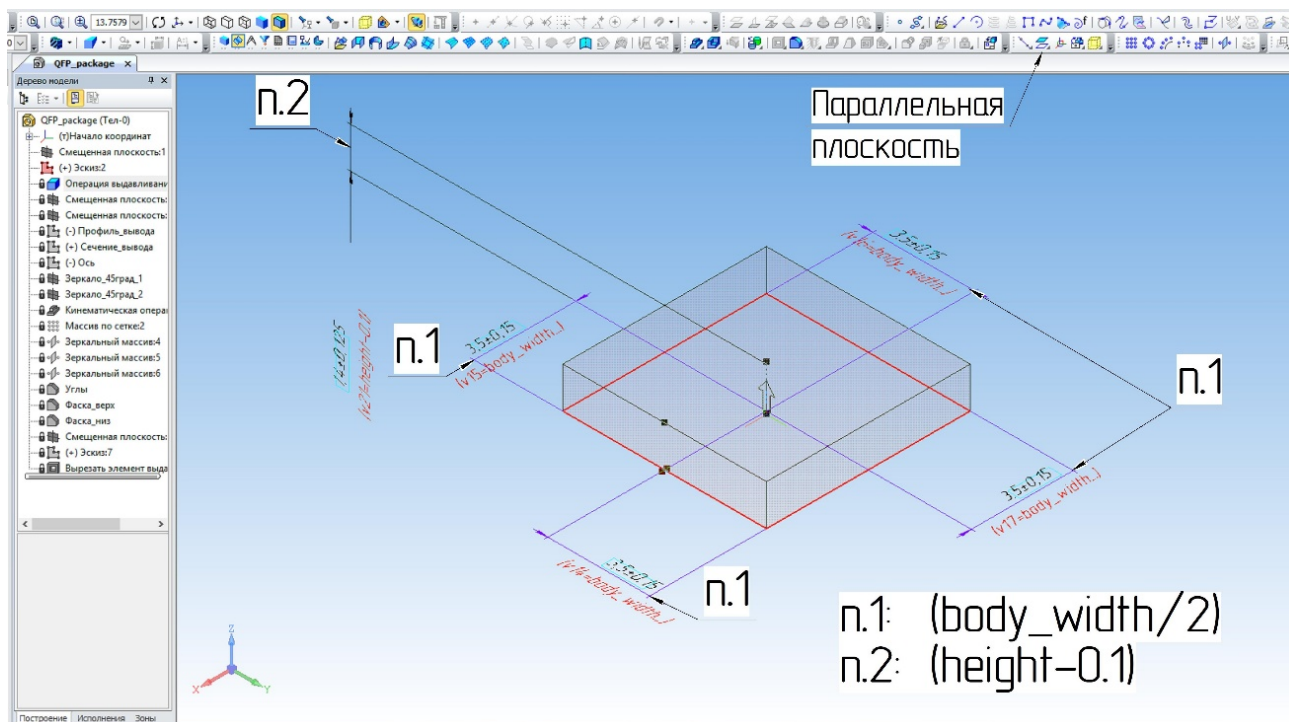


Рисунок 1 – задание объема корпуса QFP и привязка переменных.

QFP без выводов - как паук без лапок, только девчонок пугать. Так построим же один вывод.

Фактически, это прямоугольная в сечении пластина, имеющая два отгиба в противоположные стороны (разогнем немного букву "S", и вот, почти оно). При грубом приближении, нам необходимы следующие параметры вывода:

pin_thickness - толщина пластины;

pin_width - ширина пластины;

foot_length - длина участка, который будет взаимодействовать с платой при пайке;

pin_pitch - расстояние между соседними выводами;

overall_width - ширина корпуса вместе с выводами;

pin_number – количество выводов корпуса.

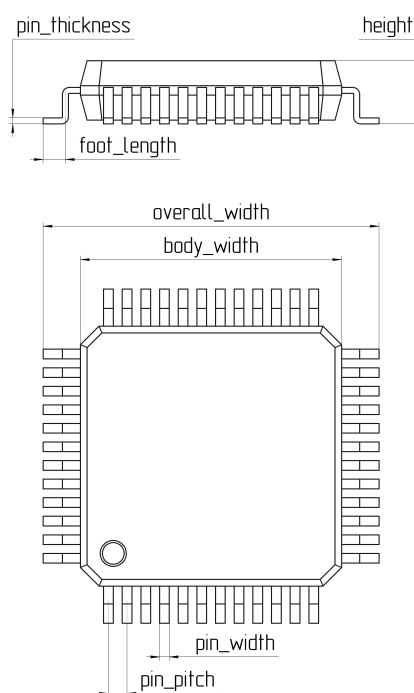


Рисунок 2 – назначение переменных.

Все эти переменные будут появляться в списке по мере их появления при построении. Исходя из того, что вывод имеет изогнутую форму, для простоты параметризации мы создадим два эскиза, в первом - сечение проволоки вывода, это будет прямоугольник шириной **pin_width** и высотой **pin_thickness**, во втором эскизе будет профиль формовки вывода (траектория изгиба). А потом с помощью кинематической операции выдадим первый эскиз с сечением по траектории, нарисованной во втором. Просто же?

А вот есть подводные камни. Тут снова не обойтись без вспомогательных параллельных плоскостей. Их полезная фишка в том, что когда мы создаем в них эскиз, он при перестроении перемещается вместе с плоскостью и привязан к ней. Казалось бы, самое простое - это построить эскиз сечения на одной из сторон полученного выше тела, но в дальнейшем возможны ситуации, которые приведут к разрушению модели, например при создании фаски исходная плоскость может быть разрушена и будет потеряна связь объектов.

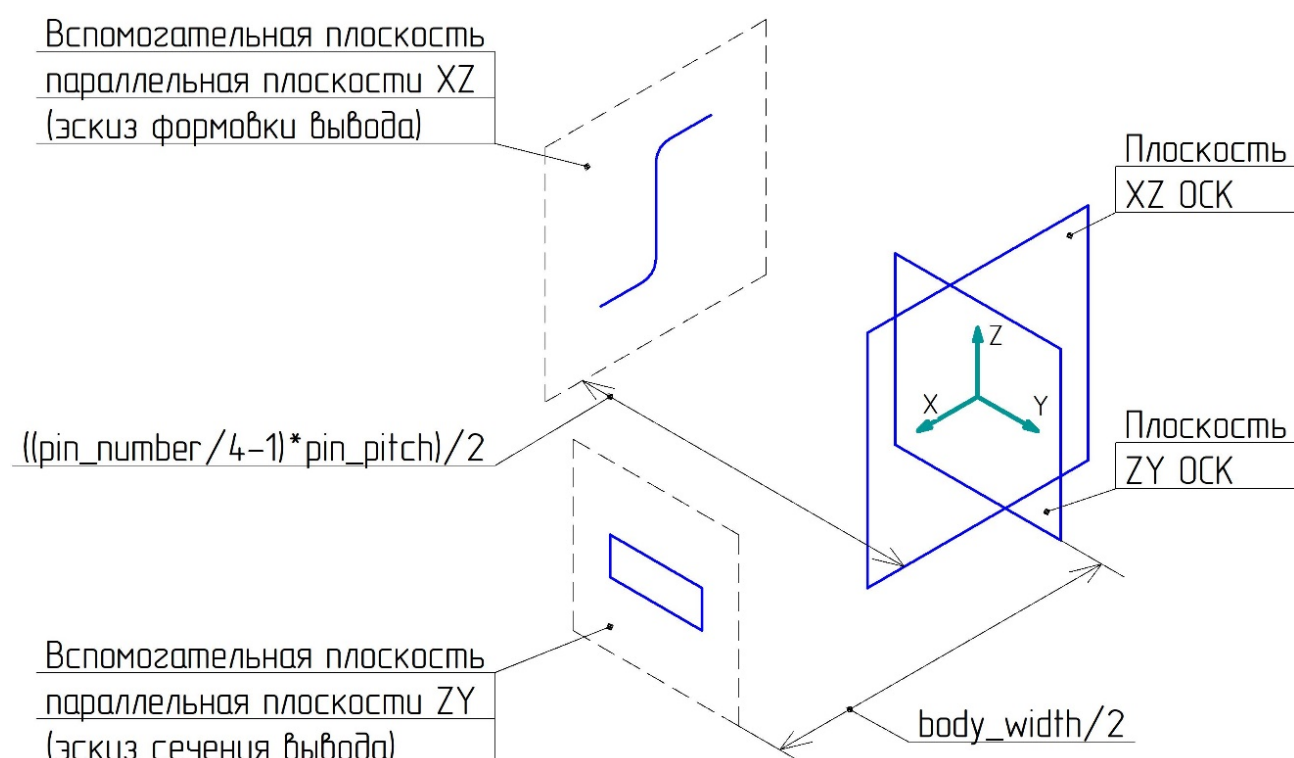


Рисунок 3 – расположение вспомогательных параллельных плоскостей.

Эскиз сечения вывода (рисунок 4а):

Как упоминалось выше, вывод в своем сечении представляет прямоугольник шириной **pin_width** и высотой **pin_thickness**. Исходя из этого можно вычислить расстояние от начала координат до сторон прямоугольника, определяющих ширину вывода, являющегося крайним на своей стороне:

$$(((pin_number/4-1)*pin_pitch)/2)-pin_width/2$$

$$(((pin_number/4-1)*pin_pitch)/2)+pin_width/2$$

И расстояние от начала координат, до сторон, определяющих нижний и верхний края указанного сечения:

$$(height+0.1)/2-(pin_thickness/2)$$

$$(height+0.1)/2+(pin_thickness/2)$$

Эскиз формовки вывода (рисунок 4б):

В изначальном варианте эскиз представляет три отрезка – участок, прилегающий непосредственно к плате, вертикальная составляющая и отвод непосредственно к корпусу. После простановки размеров выполняем скругление углов между отрезками и проставляем для них

радиальный размер для избежания изменений при перестроении, так же проставляем выравнивание точек примыкания дуг скругления по вертикали/горизонтали с соответствующими объектами. Радиус скругления примем равным 0,2 мм., как наиболее часто встречающийся, и так как данный параметр будет нести в нашей модели больше эстетическую составляющую, усложнять его вычислениями не будем.

Далее используя операцию кинематического выдавливания эскиза (рисунок 4а) по траектории (рисунок 4б), результат изображен на рисунке 4в.

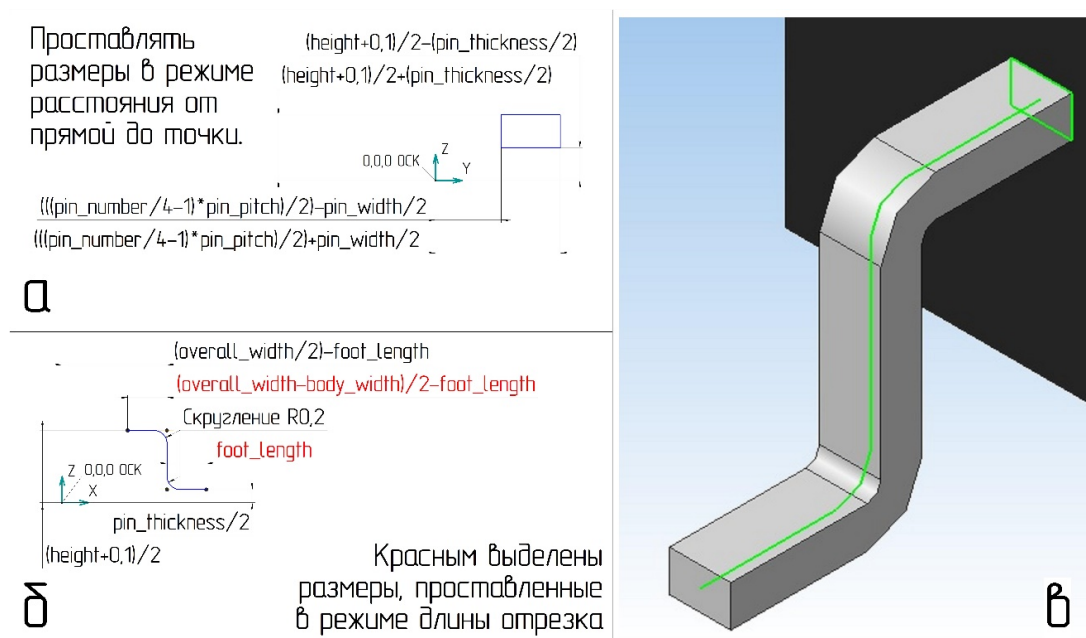


Рисунок 4 – эскизы для построения вывода микросхемы.

Один экземпляр вывода мы получили, теперь размножим его через создание массива по сетке. В качестве направления построения выберем ось Y, исходя из того, что на каждой стороне корпуса у нас одинаковое количество выводов, на одной стороне расположена лишь четверть из них. Это приводит нас к следующей формуле для вычисления количества элементов в массиве: **(pin_number/4)**. Шаг соответственно будет равен **pin_pitch**. Результат данного построения отображен на рисунке 5.

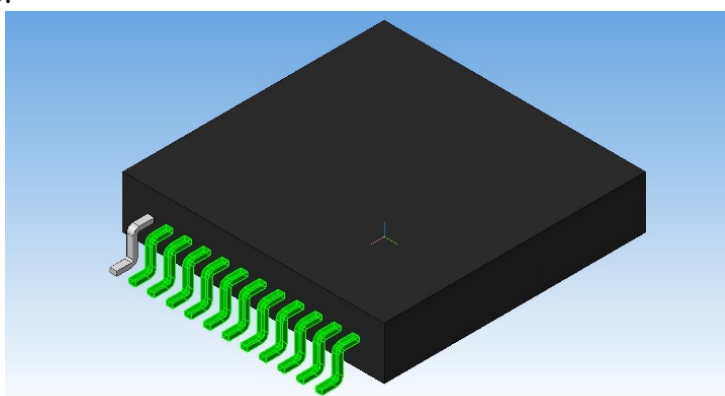


Рисунок 5 – базовый массив выводов.

Собственно, мы получили выводы вдоль одной стороны, теперь нужно сделать такие же массивы вдоль остальных трех сторон. Для этого очень удобен инструмент построения зеркального массива относительно плоскости. И снова мы строим вспомогательные плоскости, но на этот раз не параллельные, а под углом. Данное построение выполняется с указанием плоскости, по отношению к которой будет задан угол, и прямой, по которой эти плоскости в итоге будут пересекаться, а этого

ребра как раз у нас и нет, следовательно, необходимо его построить. Создадим эскиз в любой из двух плоскостей XZ или ZY, который будет представлять собой лишь один отрезок произвольной длины, совпадающий с осью Z. Имея этот эскиз строим две плоскости, под углом 45 градусов к плоскости XZ, при этом задавая поворот в противоположных направлениях, таким образом получая две новых взаимно перпендикулярных плоскости (рисунок 6).

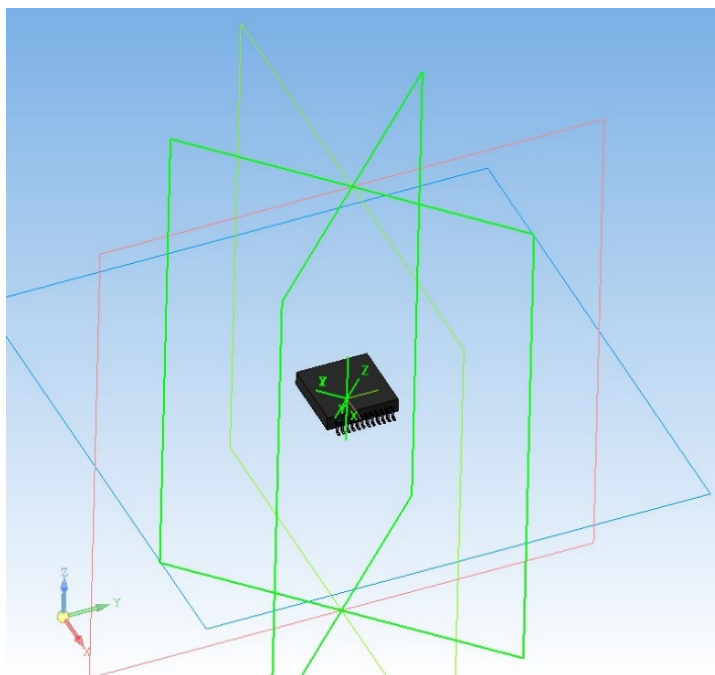


Рисунок 6 – вспомогательные плоскости под углом 45 град. к основным.

Теперь только остается трижды построить зеркальный массив, указывая в качестве отражаемого массив выводов, полученный выше, а в качестве плоскостей отражения указывая построенные нами вспомогательные плоскости и плоскость ZY. Так за малое количество действий мы получим массивы выводов по всем четырем сторонам микросхемы (рисунок 7).

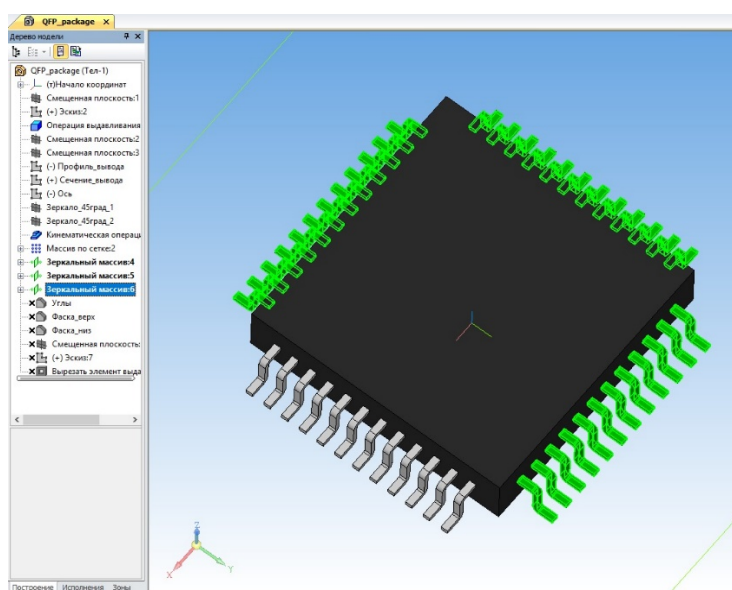


Рисунок 7 – построение зеркальных массивов вдоль оставшихся сторон.

Раз уж мы решили создать более-менее реалистичную модель, неплохо бы дорисовать фаску на углах и уклоны, возникающие при литье, ну и выдавить маркер первого вывода. Начнем с углов.

Данный параметр так же не нормирован стандартами, и отличается в зависимости от производителя, однако надо задать его размеры так, чтобы при перестроении не получилось ситуации, в которой крайние выводы «висели в воздухе». Применим к углам операцию создания фаски, с углом 45 градусов, глубину выреза зададим формулой, которая в целом не требует объяснений:

$$(\text{body_width} - ((\text{pin_number}/4) * \text{pin_pitch}))/4$$

Деление на 4, а не на 2, применено исключительно для того, чтобы срез углов не начинался слишком близко к выводам и модель была ближе к реальным образцам.

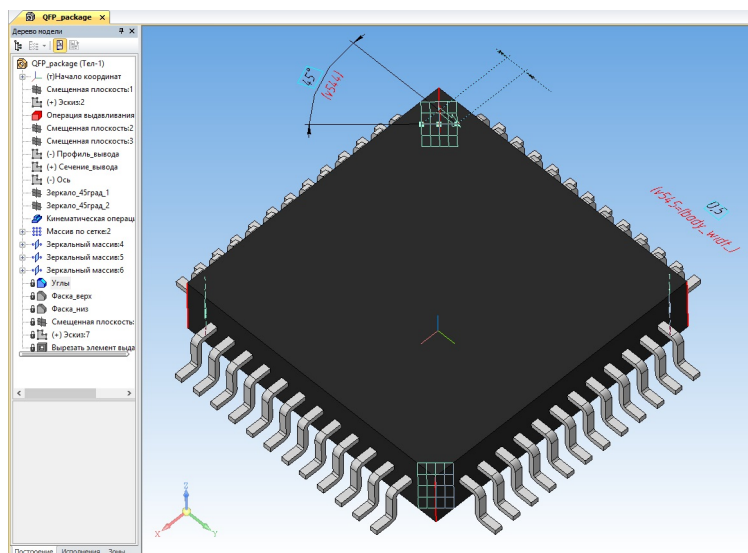


Рисунок 8 – фаска по углам.

Следующим ходом будет создание уклонов, возникающих при отливке корпуса. Угол откосов так же меняется в зависимости от производителя, обычно колеблется от 5 до 15 градусов, мы возьмем величину в 15 градусов. Создадим две фаски, по ребрам верхней и по ребрам нижней поверхностей. Наша задача сводится к тому, чтобы эти фаски при построении не сталкивались и не наползали на плоскость вхождения выводов в корпус, так что их параметры тоже привяжем к переменным, задающим высоту корпуса и толщину вывода.

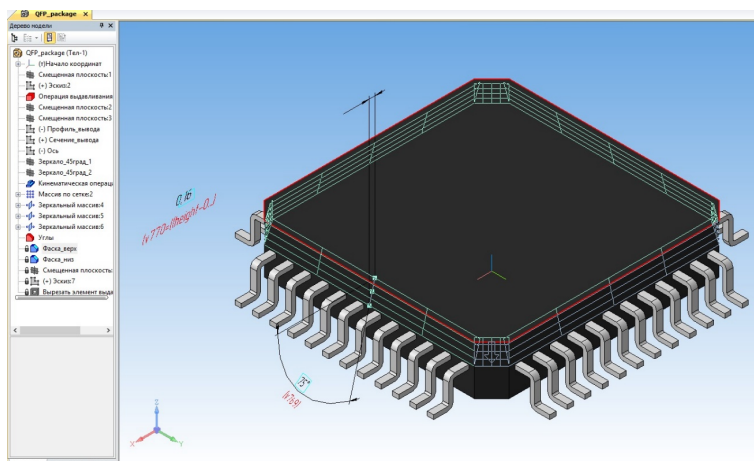


Рисунок 9 – Построение фаски от ребер верхней плоскости.

Построим фаску от верхней поверхности, под углом 75 градусов, на глубину, описанную формулой:

$$((\text{height} - 0.1 - \text{pin_thickness})/2) * 0.25$$

Думаю, тут в пояснении нуждается только умножение на 0,25. Это экспериментальная величина, введена в следствие упрощения построения (при построении выбираем плоскость, а не

фаску от каждого ребра по отдельности), что приводит к заданию угла в 75 град., из-за чего указанная формула нуждается в пропорциональном коэффициенте, который подобран так, чтобы фаска не «наползала» на выводы. Конечно, это число является достаточно грубым, но так как оно не влияет на важные параметры модели, а данное построение несет скорее эстетическую функцию, то полученной погрешностью можно пренебречь. В целом, в этом плане ситуация схожа с приближением при построение углов.

Построение для нижней плоскости является аналогичным, и используется та же формула.

Следующим шагом на пути к реализму будет метка первого вывода. Для этого построим вспомогательную плоскость, параллельную плоскости XY, со смещением на расстояние **height**, в положительном направлении по оси Z, в ней мы построим окружность диаметром 0,7 мм., и проставим расстояние от центра этой окружности до начала координат, используя простановку размеров по горизонтали и вертикали. В качестве расстояния зададим формулу:

$$\text{body_width}/2-(((\text{height}-0.1\text{-pin_thickness})/2)*0.25)-0.7$$

Данные зависимости прописаны для предотвращения попадания метки на край и столкновения с остальными построениями, призванными придать реализм модели. Данный эскиз выдадим на глубину 0,1 мм, с углом откоса 30 градусов для получения конуса, сужающегося к низу.

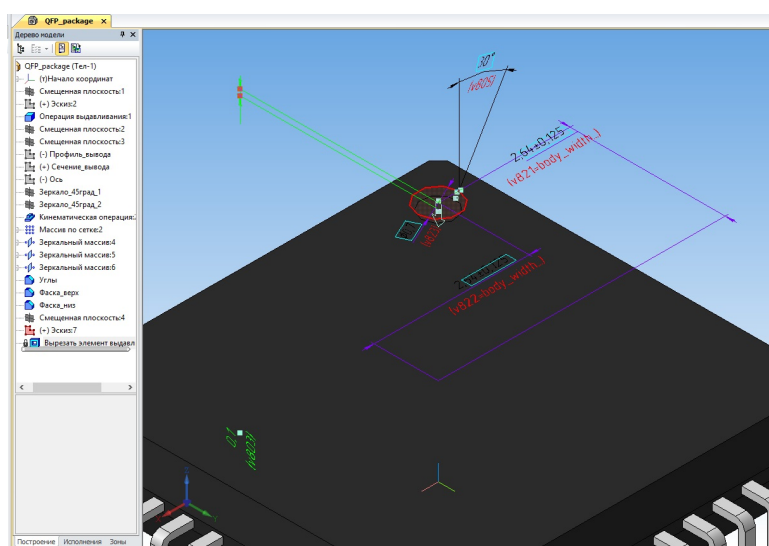


Рисунок 10 – построение метки первого вывода.

В результате всех описанных выше действий мы получили модель, в которой основные критичные параметры заданы через переменные, что предполагает возможность быстрого изменения конфигурации. Для этого во вкладке «Переменные» имеется возможность создания таблицы с разными значениями, зайдя в которую можно добавить строки с нужными значениями переменных и переставить модель одним щелчком мыши. Однако, чтобы это работало, необходимо всем переменным, используемым в модели присвоить атрибут «Внешняя», это можно сделать, нажав на правую кнопку мыши на названии переменной в их списке, и установить галочку напротив параметра «Внешняя» (рисунок 11).

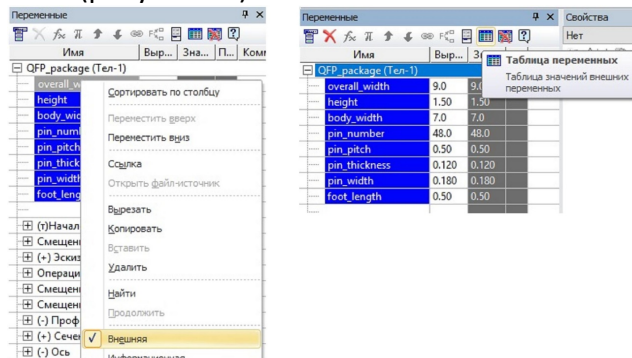


Рисунок 11 – список полученных переменных.

Комментарий	overall_width	height	body_width	pin_number	pin_pitch	pin_thickness	pin_width	foot_length
QFP208_FPT-208P-M02	30.6	3.8	28	208	0.5	0.15	0.2	0.5
QFP208_FPT-208P-M04	30.6	3.9	28	208	0.5	0.15	0.2	0.5
QFP232_FPT-232P-M01	44	4.4	40	232	0.65	0.15	0.3	0.8
QFP256_FPT-256P-M06	30.6	3.7	28	256	0.4	0.12	0.16	0.5
QFP296_FPT-296P-M01	34.6	4.4	32	296	0.4	0.12	0.16	0.5
TQFP64_FPT-64P-M04	12	1.1	10	64	0.5	0.15	0.18	0.5
TQFP80_FPT-80P-M15	14	1.2	12	80	0.5	0.15	0.22	0.6
TQFP100_FPT-100P-M09	16	1.1	14	100	0.5	0.15	0.18	0.5
TQFP100_FPT-100P-M18	14	1.2	12	100	0.4	0.12	0.18	0.6
TQFP120_FPT-120P-M17	16	1.1	14	120	0.4	0.12	0.14	0.5
LQFP32_FPT-32P-M21	9	1.5	7	32	0.8	0.12	0.3	0.5
LQFP48_FPT-48P-M05	9	1.5	7	48	0.5	0.12	0.18	0.5
LQFP64_FPT-64P-M03	12	1.5	10	64	0.5	0.12	0.18	0.5
LQFP64_FPT-64P-M09	14	1.5	12	64	0.65	0.15	0.3	0.5
LQFP80_FPT-80P-M05	14	1.5	12	80	0.5	0.12	0.18	0.5
LQFP80_FPT-80P-M11	16	1.5	14	80	0.65	0.15	0.3	0.5
LQFP100_FPT-100P-M05	16	1.5	14	100	0.5	0.12	0.18	0.5
LQFP120_FPT-120P-M05	16	1.5	14	120	0.4	0.12	0.16	0.5
LQFP144_FPT-144P-M08	22	1.7	20	144	0.5	0.12	0.2	0.5
LQFP176_FPT-176P-M02	26	1.5	24	176	0.5	0.12	0.2	0.5
LQFP176_FPT-176P-M03	22	1.7	20	176	0.4	0.12	0.18	0.5
LQFP208_FPT-208P-M06	30	1.7	28	208	0.5	0.12	0.22	0.6

Рисунок 12 – таблица переменных.

Так же можно импортировать данные из .xls таблиц, в которых заголовками столбцов будут являться названия переменных, в первом столбце можно разместить комментарий.

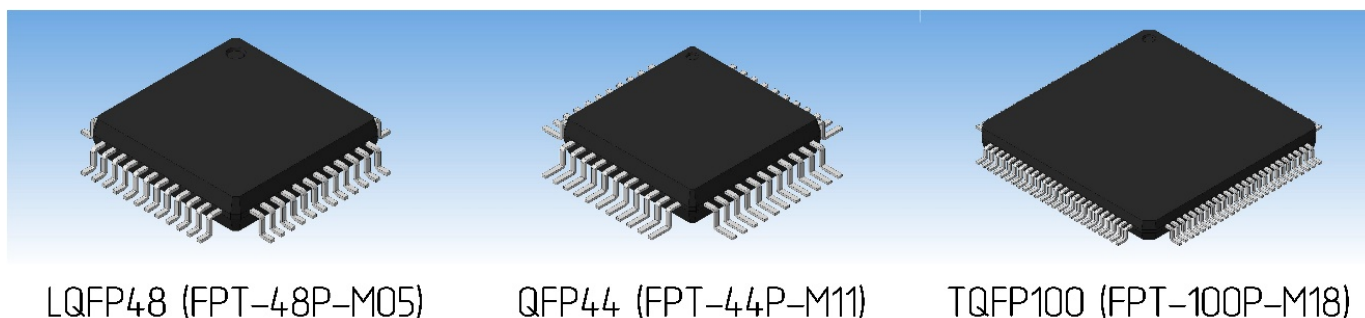
Что ж, модель построена, можно попробовать менять переменные и посмотреть, не развалится ли вся наша конструкция. Для примера попробуем построить корпуса:

LQFP48 (FPT-48P-M05);

QFP44 (FPT-44P-M11);

TQFP100 (FPT-100P-M18).

Перечисленные корпуса имеют разные габариты, число выводов, расстояние между ними, и неплохо подходят для проверки вносимых изменений.



LQFP48 (FPT-48P-M05)

QFP44 (FPT-44P-M11)

TQFP100 (FPT-100P-M18)

Рисунок 13 – построение моделей с разными переменными.

Как мы можем видеть, модели собрались, параметры меняются на табличные, наша задумка работает. Естественно, работает она только в разумных пределах, ограниченных заданием реальных значений, например если мы пропишем ширину корпуса больше чем общая ширина, или ширину вывода больше чем расстояние между соседними, то результат будет непредсказуем, но это не имеет практического применения.

Послесловие

Как я подметил в начале, в данной заметке рассмотрен лишь один из возможных способов. И как можно видеть по иллюстрациям, была взята готовая модель, в которой я исключал операции, чтобы продемонстрировать использованные методы. Готовая модель была использована исключительно по той причине, что после ее создания она была опробована при создании соответствующей библиотеки для Altium Designer, и показала себя работоспособной. Она конечно имеет ряд допущений, касающихся декоративных, если так можно выразиться, составляющих, однако параметры, влияющие непосредственно на технические моменты определяются с должной точностью. Так же я не делал акцент на таких деталях, как задание цвета операции выдавливания и подобных нюансах, не имеющих отношения к методу построения.

Преимущества параметрической модели очевидны – построение модели заняло примерно 20 минут, не сильно больше чем построение модели без привязки к переменным, однако для внесения изменений теперь достаточно лишь ввести значения использованных переменных, дать команду на перестроение модели и получить результат.

Так как Altium Designer использует модели в формате .step, отличным дополнением к описанной модели будет скрипт, который будет перебирать последовательно все строки таблицы переменных, перестраивать модель и сохранять в указанном формате, это значительно ускорило бы процесс, но в данный момент это только задумки, мои навыки в программировании для ПК недостаточны.

Сохраненные в .step модели самых ходовых корпусов и саму параметрическую модель можно скачать здесь: <https://grabcad.com/library/qfp-package-collection-1>