

3D-Druck-Bauteile mit der Part Design Workbench von FreeCAD erstellen

Benötigt wird eine einigermaßen aktuelle Version von FreeCAD 0.17. Zu finden hier:

<https://github.com/FreeCAD/FreeCAD/releases>

Für dieses Tutorial habe ich die Version 0.17.10887 für Windows verwendet. Bei anderen Versionen kann es kleine Abweichungen geben. Das Grundprinzip sollte sich aber nicht verändern. Bevor man eine Entwicklerversion nutzt sollte man eine „offizielle Release“ installieren und mindestens einmal starten. Die Entwicklerversionen legen – zumindest auf Windows-Rechnern – die betriebssystemspezifischen Daten leider nicht an. Sie laufen aber dann problemlos ohne eigentliche Installation.

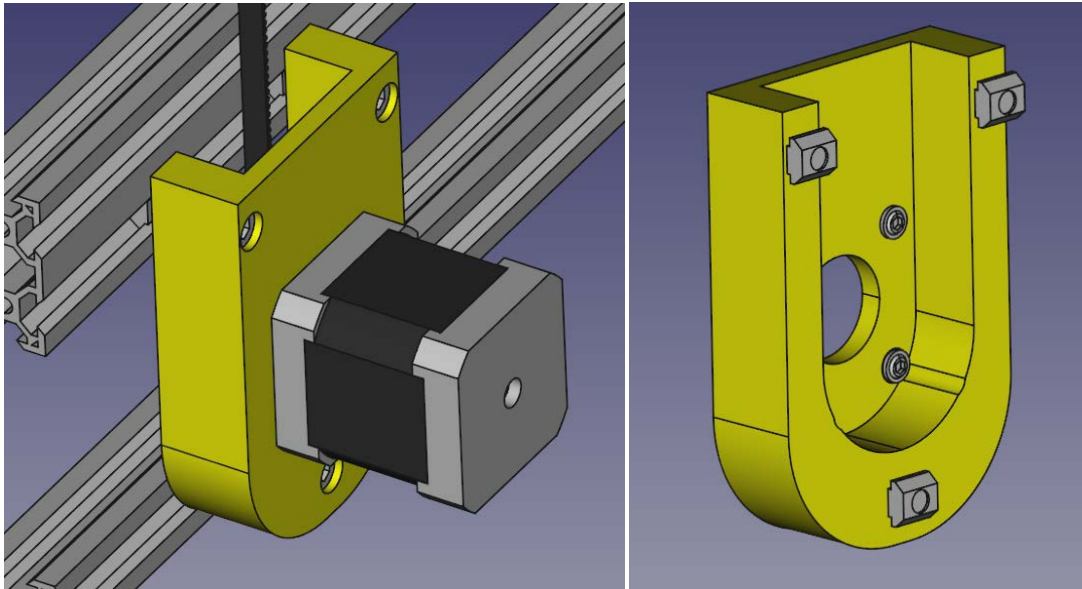
FreeCAD arbeitet mit unterschiedlichen Arbeitsumgebungen für einzelne Aufgaben, den sogenannte Workbenches. Die Workbenches stellen jeweils die nötigen Funktionen für einen bestimmten Arbeitsablauf zur Verfügung. Für die 3D-Konstruktion eignen sich die Workbenches „Part“ und „Part Design“. Die Part-Workbench arbeitet dabei mit geometrischen Primitiven wie Quader, Zylinder, Kugel, Torus etc. die dann mit booleschen Operationen kombiniert werden (Vereinigung, Durchschnitt, Differenz). Soweit ist das von anderen CAD-Programmen bekannt und sollte keine Probleme aufweisen.

Die Part-Design Workbench ist „anders“ und zunächst eher gewöhnungsbedürftig. Teile werden nicht direkt dreidimensional konstruiert sondern es werden zunächst zweidimensionale Zeichnungen erstellt – sogenannte Sketches – die dann mit verschiedenen Funktionen dreidimensional gemacht werden. Auch hier hat man die Möglichkeit nachträglich Teile „anzubauen“ oder Löcher in bestehende Objekte zu „bohren“.

Für dieses Tutorial verwenden wir neben der Part Design Workbench noch die Spreadsheet Workbench. Das ist eine eingebaute Tabellenkalkulation mit der man z.B. zentrale Abmessungen für seine Konstruktion verwalten kann. Das Besondere: Bei der Konstruktion kann man auf die Daten eines Spreadsheets zugreifen. Die Konstruktion ändert sich dabei automatisch. Wir werden das in diesem Tutorial auch gezielt nutzen um das konstruierte Bauteil – einen Nema17 Motorhalter – nachträglich auf einen Nema23-Motor anzupassen – ohne die eigentliche Konstruktion nochmal verändern zu müssen.

Vorüberlegungen

Man sollte im Vorfeld der Konstruktion schon eine gewisse Vorstellung davon haben was man konstruieren möchte. In diesem Fall einen Motorhalter der an zwei Alu-Extrusionsprofilen befestigt wird. Das fertige Teil soll im eingebauten Zustand ungefähr so aussehen (rechts nur ein Bild von einer frühen Vorversion):



Nicht allzu kompliziert für den Anfang – aber schon mit einigen Punkten auf die man achten muss. Der Motor wird mit 4 Schrauben befestigt. Im Falle eines NEMA17 sind das M3-Schrauben. Da der Motorhalter für unterschiedliche Motoren geeignet sein soll müssen wir die Daten der Schrauben parametrisieren. Für die Konstruktion brauchen wir dabei den Radius (oder Durchmesser) der Bohrungen für die Schrauben und den Radius (oder Durchmesser) der Schraubenköpfe.

Außerdem benötigen wir für die Befestigung an den T-Profilen M5 Schrauben – auch hier wieder Bohrungsradius und Kopfradius der Schrauben. Zusätzlich die Höhe des Schraubenkopfes weil wir die Schrauben versenken wollen.

Der Abstand der Befestigungsbohrungen am Motor ist ebenfalls ein sinnvoller Parameter, genauso wie die Höhendifferenz zwischen den T-Profilen. Insgesamt brauchen wir also folgende „Variablen“ für die Konstruktion:

- Radius Schraube Motorbefestigung - 1,5 mm für M3
- Radius Schraubenkopf Motorbefestigung - 2,66 mm für Innensechskant M3
- Radius Schraube Befestigung (die M5-Schrauben) – 2,5 mm für M5
- Radius Schraubenkopf Befestigung – 4,14 mm für M5
- Höhe Schraubenkopf Befestigung – 5 mm für M5
- Außenabmessungen des Motors (auf die Länge können wir verzichten) – 42,3 mm bei Nema17
- Abstand Befestigungsbohrungen am Motor – 31 mm für Nema 17
- Radius der Stufe vorne am Motor – 11 mm beim Nema 17
- Abstand T-Profile (Abstand von Mitte zu Mitte der verwendeten Nuten) – hier 80 mm

Los geht's – das Spreadsheet

Beginnen wir also mit FreeCAD. Nach dem Start öffnen wir ein neues Dokument und wechseln erstmal zur Spreadsheet Workbench. Über die Menüfunktion „Spreadsheet“->„Create Spreadsheet“ (oder über das entsprechende Symbol in der Iconleiste) erzeugen wir ein neues Spreadsheet.

Wir sollten neben der eigentlichen Arbeitsfläche folgende „Panels“ eingeschaltet haben:

- Combo View
- Tree View
- Property View
- Report View (optional)
- Python View (optional)

Ich arbeite mit zwei Bildschirmen und lege mir die Panels auf den zweiten Bildschirm. Das ist einfach übersichtlicher.

Im Tree view sehen wir nun das neu angelegte Spreadsheet. Als Erstes benennen wir es um. Einfach draufklicken und F2 drücken (oder mit der rechten Maustaste klicken und „Rename“ auswählen). Nennen wir das Blatt „Daten“.

Wichtig ist: Der Name darf KEINE Leerzeichen enthalten – sonst klappt das später mit der Nutzung der Daten bei der Konstruktion nicht.

Mit einem Doppelklick öffnen wir nun die Tabelle und tragen unsere gewünschten Parameter ein. Damit wir später noch wissen was wir da gemacht haben sollte man IMMER eine aussagekräftige Beschreibung der Parameter hinzufügen. Das könnte dann so aussehen:

	A	B
1	1.5 mm	Radius für Bohrung Motorbefestigung
2	2.66 mm	Radius für Schraubenkopf Motorbefestigung
3	2.5 mm	Radius Befestigungsbohrung
4	4.14 mm	Radius Schraubenkopf Befestigung
5	5 mm	Höhe Schraubenkopf Befestigung
6	42.3 mm	Breite/Höhe Motor
7	31 mm	Abstand Schrauben Motorbefestigung
8	11 mm	Durchmesser der Stufe am Motor vorne
9	80 mm	Abstand der Nuten in den Profile (Mitte untere Nut bis Mitte obere Nut)

Wichtig hierbei ist es das wir nicht einfach nur Zahlen eintragen sondern immer auch die Maßeinheit (hier mm) angeben. FreeCAD kann mit unterschiedlichen Maßeinheiten wunderbar umgehen. Wir können z.B. auch sowas wie „=10 mm + 1 in“ in eine Zelle eintragen. Ergebnis: 35,4 mm. Wenn wir keine Formeln verwenden können wir auf das Gleichheitszeichen am Anfang der Eingabe verzichten.

Im nächsten Schritt müssen wir unseren Parametern nun Namen geben. Auch hier gilt wieder: KEINE LEERZEICHEN! Und sinnvoll kürzen sollte man die Namen auch *g* Führen wir also zunächst ein paar Kürzel ein:

MS = Schraube Motorbefestigung

PS = Profilbefestigung

M = Motor

Rad = Radius

H = Höhe

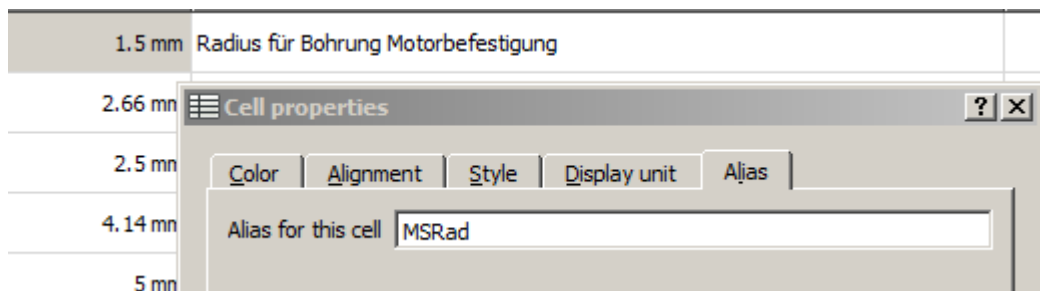
Dist = Abstand

Head = Schraubenkopf

Damit basteln wir uns nun folgende Parameternamen (von oben nach unten im Spreadsheet):

MSRad	Motorschraube Radius
MSHeadRad	Motorschraube Radius Schraubenkopf
PSRad	Profilbefestigungsschraube Radius
PSHeadRad	Profilbefestigungsschraube Radius Schraubenkopf
PSHeadH	Profilbefestigungsschraube Höhe Schraubenkopf
MotorH	Motorabmessungen Höhe/Breite
MotorStufeRad	Durchmesser der Stufe an der Motorfront
MSDist	Abstand der Motorbefestigungsschrauben
NutDist	Nutabstand der T-Nut Profile

Diese Parameternamen können wir nun den Zahlenwerten im Spreadsheet zuweisen. Dazu klicken wir mit der rechten Maustaste auf die Zelle und wählen „Properties“ (oder einfacher Strg-Shift-A, ein Symbol für die Namensvergabe findet sich auch in der Toolbar). Im letzten Tab „Alias“ können wir nun den Namen vergeben:



FreeCAD färbt die Zellen gelb ein die einen Alias haben.

Contents		
	A	B
1	1.5 mm	Radius für Bohrung Motorbefestigung
2	2.66 mm	Radius für Schraubenkopf Motorbefestigung
3	2.5 mm	Radius Befestigungsbohrung
4	4.14 mm	Radius Schraubenkopf Befestigung
5	5 mm	Höhe Schraubenkopf Befestigung
6	42.3 mm	Breite/Höhe Motor
7	31 mm	Abstand Schrauben Motorbefestigung
8	11 mm	Durchmesser der Stufe am Motor vorne
9	80 mm	Abstand der Nuten in den Profile (Mitte untere Nut bis Mitte obere Nut)

Im Property View, Reiter „Data“ werden außerdem die Parameter auch aufgelistet:

Property view	
Property	Value
Base	
Label	Daten
MSDist	31,00 mm
MSHead Rad	2,66 mm
MSRad	1,50 mm
Motor H	42,30 mm
Motor Stufe ...	11,00 mm
PSHead H	5,00 mm
PSHead Rad	4,14 mm
PSRad	2,50 mm
Taschentiefe	10,00 mm
Nut Dist	80,00 mm

Damit ist das Spreadsheet erst mal fertig. Wir können es schließen (und das Projekt sicherheitshalber mal kurz speichern).

In einem „realen“ Projekt wird man meist erst während der Konstruktion merken welche Parameter man braucht. Das ist aber kein Problem. Wir können ja jederzeit neue Parameter in das Spreadsheet einfügen. Wir können sogar mehrere Spreadsheets verwenden. Das mache ich häufiger. Hierzu habe ich z.B. ein Spreadsheet mit Standardabmessungen von Schrauben M3-M10 welches ich bei Bedarf in das aktuelle Projekt kopieren und ein zusätzliches Spreadsheet mit den eigentlichen Parametern des Bauteils. Für dieses einfache Beispiel wäre das aber zu viel Aufwand.

Durch die Namensvergabe (Alias) können wir neue Parameter auch „zwischenrin“ einfügen wenn das sinnvoll ist um mehr Übersichtlichkeit zu erreichen. Man sollte auch nicht mit Texten in den

Spreadsheets geizen. Eine Aufteilung in sinnvolle Gruppen von Parametern – mit Zwischentexten/Erläuterungen in eigenen Zeilen – macht sich später fast immer bezahlt, wenn man doch noch mal an die Konstruktion ran will und nachvollziehen muss was man da eigentlich gebastelt hat. Insbesondere wenn man im Spreadsheet Formeln verwendet macht es sehr viel Sinn diese ausführlich zu beschreiben.

Nach diesen Vorarbeiten können wir endlich mit der eigentlichen Konstruktion beginnen.

Endlich dreidimensional – oder?

Wechseln wir also zur Part Design Workbench.

Als Erstes sollten wir IMMER ein „Part“ anlegen. Das dient dann als Gruppierung für die – ggf. mehreren – Komponenten unserer Konstruktion. Erstellen wir also das „Part“. Im Tree view können wir es auch schnell umbenennen in „Motorhalter“. Ein Part ist eine abstrakte Gruppe und hat selbst erstmal keine Geometrie. Es gibt allerdings ein paar Default-Inhalte (Koordinatenachsen und Ebenen) – die wir aber geflissentlich ignorieren.

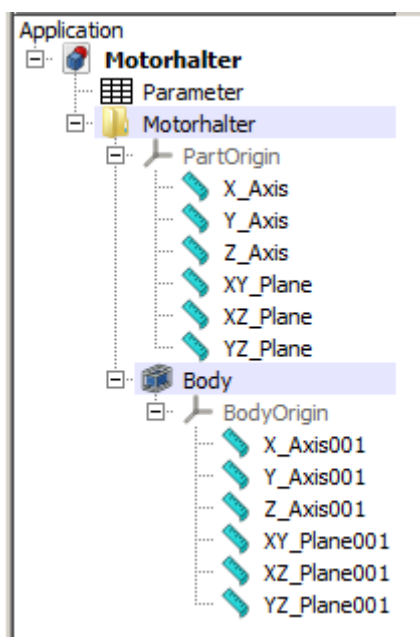
Direkt anschließend erstellen wir nun einen „Body“ – das wird das eigentliche Bauteil. Bodies sind die eigentlichen 3D-Bauteile im Part-Design. Auch Bodies haben Default-Elemente – identisch zu den Parts. Beim Erstellen eines Bodies wird dieser immer im aktuell aktiven Part erstellt. Ist kein Part aktiv, dann direkt „oben“ im Teilebaum. Man kann ein Part aktivieren indem man doppelt auf das Part im Tree View klickt. Nochmal Doppelklick deaktiviert das Part wieder. Aktive Parts und Bodies werden im TreeView neuerdings hellblau (oder hellviolett?) hinterlegt. Alternativ hilft auch die Python Console (Menü View->Panels->Python Console). Wurde mit einem Doppelklick ein Part aktiviert steht darin sowas wie:

```
Gui.getDocument('Unnamed').ActiveView.setActiveObject('part',  
App.getDocument('Unnamed').getObject('Part'))
```

Wurde ein Part deaktiviert steht sowas da:

```
Gui.getDocument('Unnamed').ActiveView.setActiveObject('part', None)
```

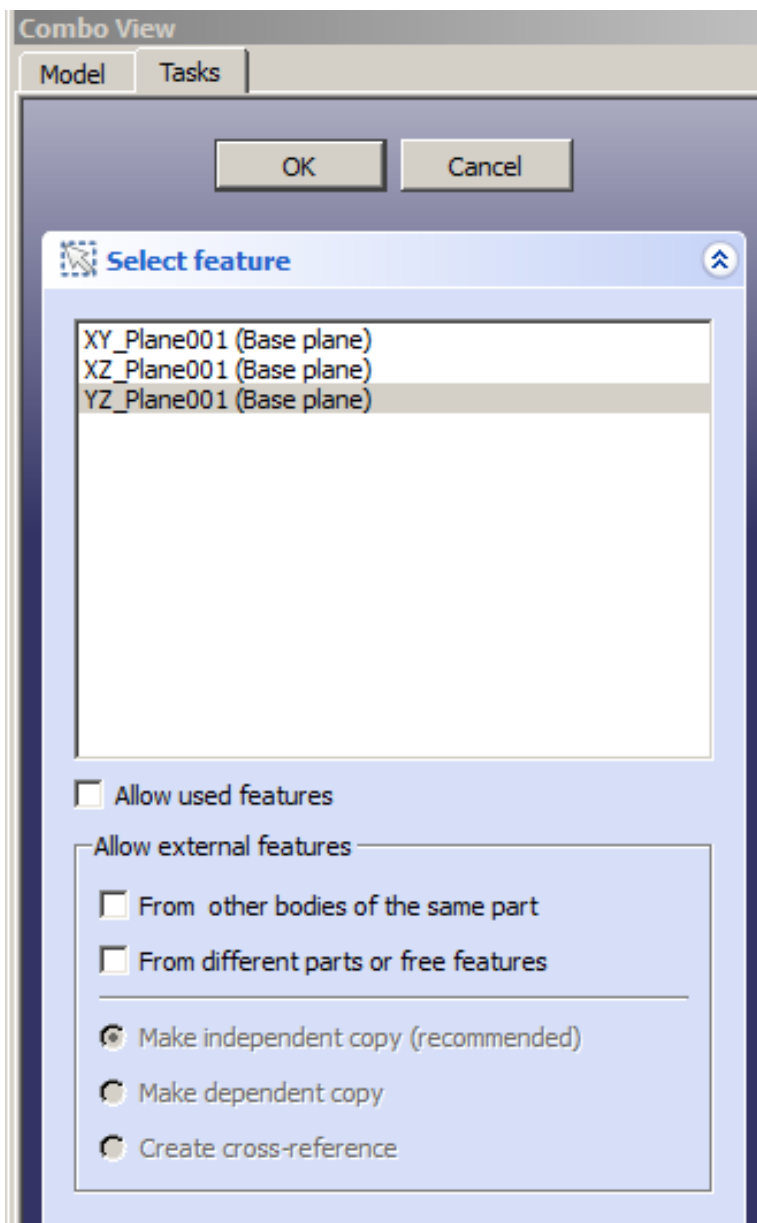
Das gilt für Bodies analog genauso. Beim Body gibt es aber im Tree View (rechte Maustaste) auch die Möglichkeit „Toggle active body“ als Alternative zum Doppelklick. Unser Tree View sollte nun – aufgeklappt – ungefähr so aussehen:



Für den nächsten Schritt muss das Objekt „Body“ aktiviert sein (direkt nach der Erstellung ist es automatisch aktiv). Wir erzeugen nun den ersten Sketch – den Außenumriß unseres Bauteils – indem wir auf „Create Sketch“ klicken. Wahlweise im Menü „Part Design“, in der Symbolleiste oder im

Combo View – alles dieselbe Funktion. FreeCAD blendet nun die Bezugsebenen des aktuell aktiven Bauteils ein – inkl. der Namen der Bezugsebenen. Da unser Motorhalter links an einem Rahmen befestigt werden soll wählen wir die YZ_Plane001 als Bezugsebene für unsere Skizze.

Den eigentlichen Alu-Profilrahmen haben wir im CAD nicht vorher erstellt – daher können wir die Checkboxen leer lassen. Hier kann man die Möglichkeit schaffen bei der Konstruktion Bezug auf andere Teile der Konstruktion zu nehmen. Das ist aber Stoff für weiterführende Tutorials. Das Combo View sieht also nun so aus:



Flugs auf „OK“ geklickt und wir finden uns in der Sketcher Workbench wieder. Im Hauptfenster prangt ein Koordinatenkreuz und die Benutzeroberfläche (Menüs und Icons) sieht plötzlich ganz anders aus. Es gibt zwei auf den ersten Blick recht ähnlich aussehende Symbolleisten.

Das sind die Symbole für die Erstellung von (2D-)Elementen:

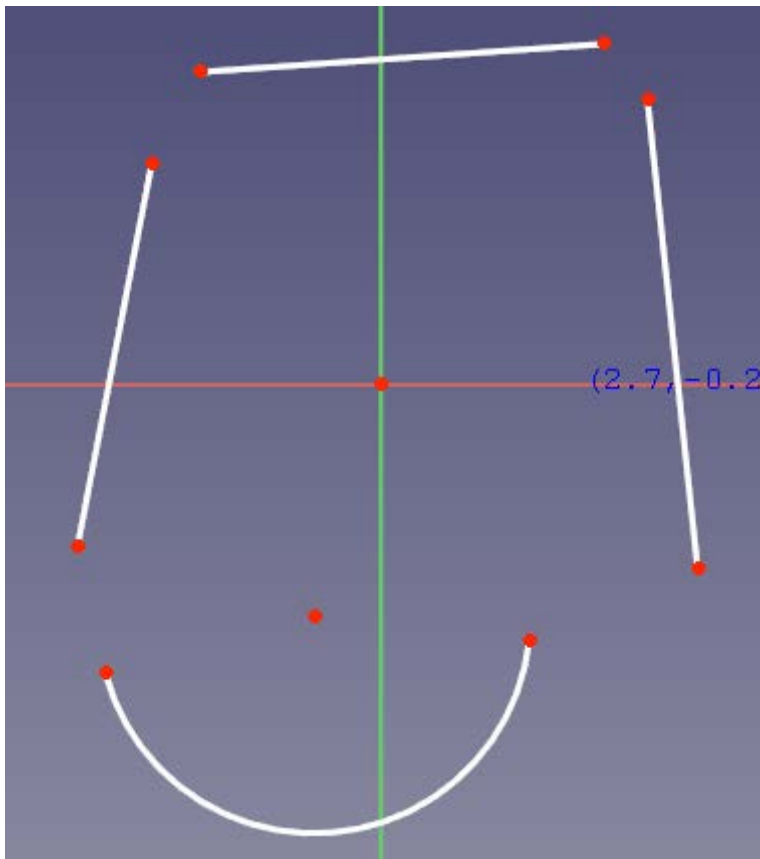


Und das sind die Icons für die Constraints mit denen wir unsere Zeichnung auf bestimmte Abmessungen festlegen:



Zunächst schalten wir im Combo View die Auto Constraints aus – die behindern zu Anfang mehr als sie nützen. Die Einstellung findet sich im Bereich „Edit Controls“ den man ggf. erst aufklappen muss.

Als Erstes sollten wir uns nun Gedanken darüber machen wo im Koordinatensystem wir gerne den Nullpunkt hätten. Ich habe in diesem Beispiel die untere Befestigungsschraube gewählt. Entsprechend skizzieren wir mit 3 Linien und einem Kreisbogen (2. und 3. Symbol aus der Elementleiste) folgende Grafik:



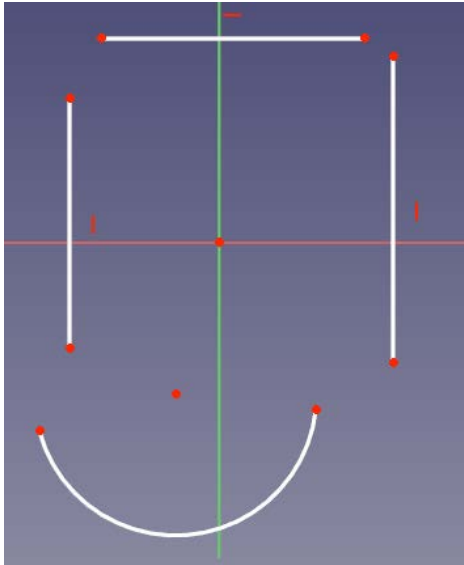
Ich habe die Einzelnen Elemente bewußt erstmal „schief“ und ohne Verbindung untereinander angelegt um im weiteren Verlauf besser zeigen zu können wie man die Constraints benutzt. Wenn man etwas genauer „freihand“ zeichnet erzeugt die Sketcher Workbench selbstständig „offensichtliche“ Constraints. Das kann am Anfang manchmal verwirrend sein. Man kann das im Combo View abschalten, wenn man die „Edit controls“ aufklappt und dort die Checkbox bei „Auto constraints“ deaktiviert.

Im ersten Schritt werden wir nun bei der oberen Linie einen waagerechten Verlauf erzwingen. Hierzu wählen wir das „Horizontal Constraint“ Werkzeug:



Entweder draufklicken oder auf der Tastatur die h-Taste drücken. Nun einfach die obere Linie anklicken und wie von Zauberhand ist die Linie nun völlig waagrecht. Entsprechend verfahren wir nun mit den beiden anderen Linien – nur nehmen wir nun natürlich das „Vertical constraint“-Werkzeug (Taste v).

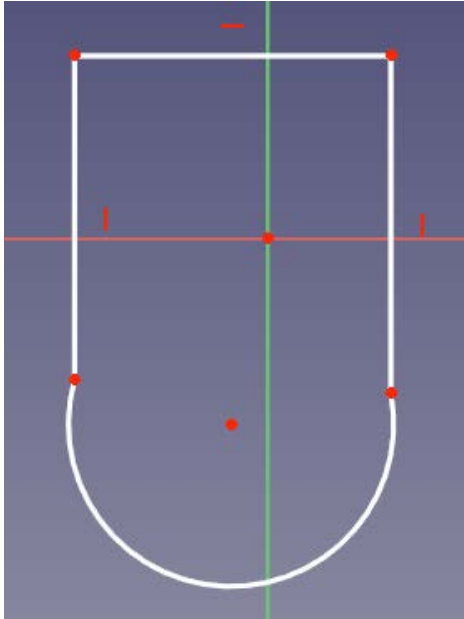
Nun sieht unsere Grafik schon gleich (geringfügig) besser aus:



Wir sollten die „Ecken“ irgendwie verbinden... Kein Problem mit dem „Coincident“ Werkzeug (Taste c):



Einfach nacheinander auf zwei Punkte klicken und die Linie verbinden sich sofort. Das klappt auch mit dem Kreisbogen. Beim Klicken immer darauf achten das die Punkte/Linien auf die man zielt vor dem Klick auch eingefärbt sind. Manchmal funktioniert die Selektion nicht sonderlich präzise. Gelegentlich hilft es dann den Sketch zu schließen und neu zu öffnen. Wenn wir die richtigen Punkte erwischt haben sieht unsere Zeichnung nun so aus:

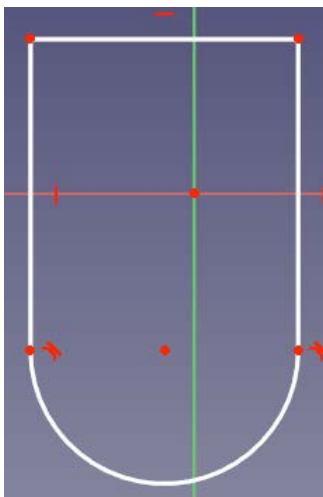


Langsam wird's doch. Der Kreisbogen soll natürlich „gerade“ in die Seitenlinien übergehen. Die Linien sollen also Tangenten am Kreisbogen sein. Natürlich gibt es auch dafür ein Constraint-Werkzeug:

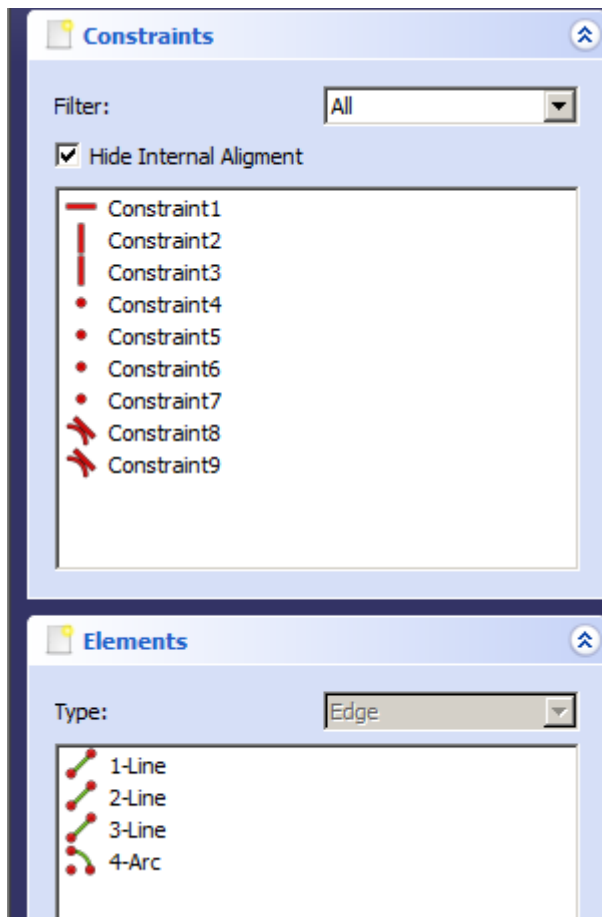


Welche Taste? Richtig: t

Damit klicken wir nun zunächst auf den Kreisbogen, dann auf die linke, senkrechte Linie (umgekehrt geht auch). Dann dasselbe für die rechte Seite. Und schon sieht unser Teil deutlich besser aus:



Man sieht auch das FreeCAD die verwendeten Constraints in der Zeichnung einblendet. Außerdem werden sowohl Elemente/Linien als auch Constraints im Combo-View angezeigt.



Jetzt kommen wir zu etwas abstrakteren Constraints. Zunächst soll unser Teil ja symmetrisch zur vertikalen Achse liegen (in diesem Fall die Z-Achse). Hierzu gibt es das Symmetrie Constraint Werkzeug (Taste s):



Wir klicken damit zunächst auf den Punkt links oben, dann auf die vertikale (grüne) Achse und dann auf den Punkt rechts oben. Sofort richtet sich die Zeichnung symmetrisch um die Mittelachse aus. Später werden wir noch eine andere Anwendung für dieses Werkzeug kennenlernen.

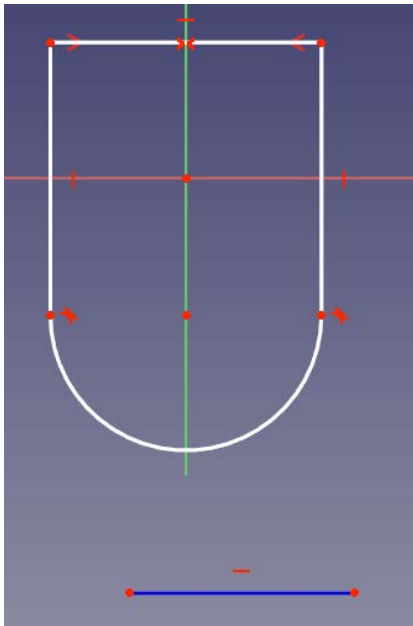
Für den nächsten Schritt brauchen wir eine Hilfslinie. Das sind Linien die ganz normal mit Constraints konstruiert werden aber keinen Anteil an der eigentlich zu erzeugenden Form haben. Wir schalten also um auf „Construction Mode“:



Die Icons für die Linien verändern sich farblich:



Nun ziehen wir eine horizontale Linie unter unserer Grafik. Diesmal gleich „richtig“ Waagrecht – Sketcher erstellt dann automatisch einen „Horizontal Constraint“:



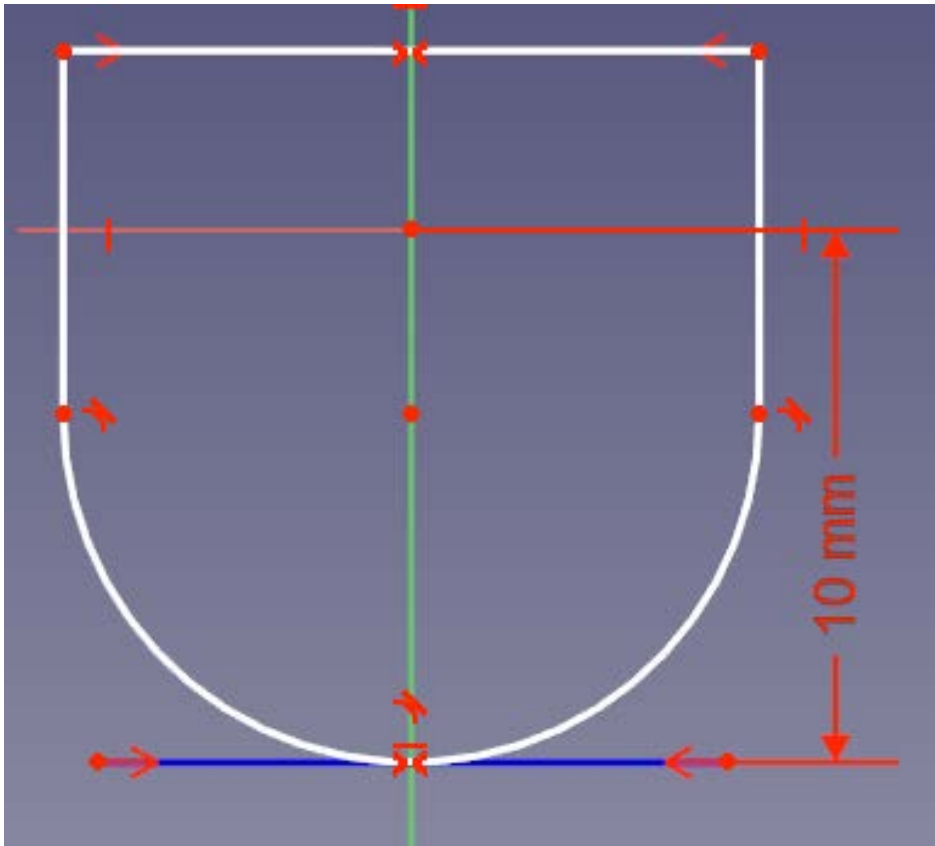
Damit unser Sketch später vollständig definiert ist müssen wir diese Linie nun genauer festlegen. Die Hilfslinie soll die Unterkante des unteren T-Profils darstellen und dient uns hier dazu den Kreisbogen genauer zu definieren.

Der Motorhalter ist für 20er T-Profile gedacht. Der Kreisbogen soll bündig mit dem unteren Nutprofil abschließen. Da wir den Nullpunkt des Bauteils bei der unteren Befestigungsschraube haben möchten liegt diese Unterkante 10 mm unter diesem Nullpunkt (halbes Profil eben). Wir stellen also nun mit dem „Vertical Distance“ Werkzeug (Taste Shift-v) den Abstand dieser Linie auf 10 mm vom Nullpunkt des Koordinatensystems ein:



Dazu klicken wir zuerst auf den Nullpunkt und dann auf einen der Endpunkt der neuen Hilfslinie. Es öffnet sich ein kleiner Dialog. Dort können wir „10 mm“ eingeben und diesem Maß auch einen eigenen Namen geben. Dieser Name erscheint später im Property View und dort kann man diesen Wert auch jederzeit direkt ändern.

Zusätzlich erzwingen wir auch bei der Hilfslinie eine Symmetrie zur senkrechten Achse. Wie das geht wissen wir ja schon. Nun noch den Kreisbogen und die Hilfslinie mit einem „Tangential constraint“ (Taste t) verbinden. Die Zeichnung sollte nun ungefähr so aussehen:



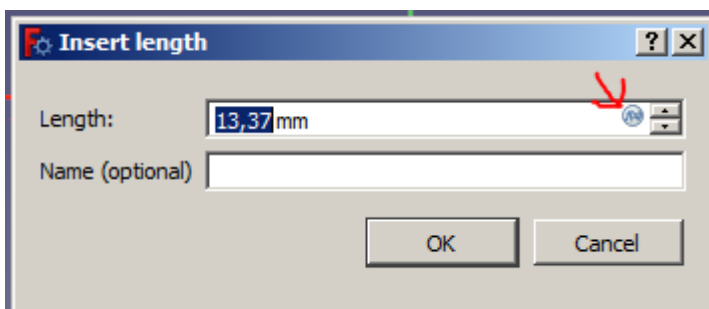
Nun müssen wir uns Gedanken um die übrigen Abmessungen machen. Zunächst die Gesamthöhe. Die wird bestimmt durch die beiden Nutprofile bzw. deren Abstand und deren Abmessungen. Den Abstand der Befestigungsnuten haben wir bereits im Spreadsheet: NutDist. Dazu kommen noch zwei halbe Profilbreiten – jeweils 10 mm. Insgesamt also:

$10 \text{ mm} + \text{Nutabstand (Parameter NutDist aus dem Spreadsheet)} + 10 \text{ mm}$

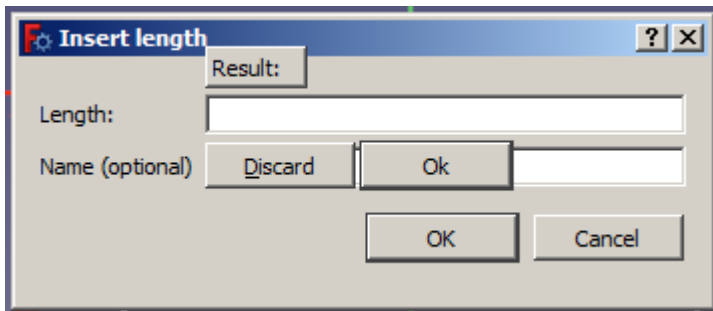
oder

$20 \text{ mm} + \text{NutDist}$

Das legen wir auch wieder mit dem Vertical Constraint Werkzeug fest. Damit klicken wir z.B. auf den linken oberen Punkt und auf den linken Punkt der Hilfslinie. Es öffnet sich der bereits bekannte Dialog zur Eingabe des Abstands:



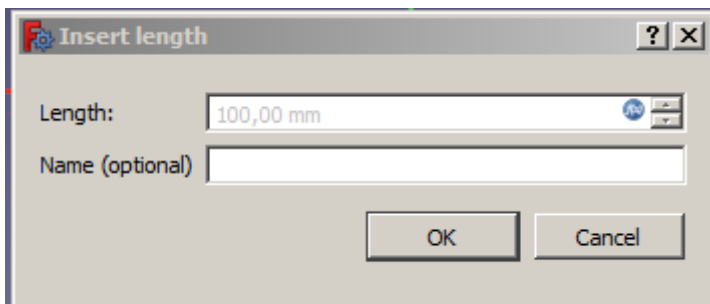
Diesmal geben wir aber keinen Zahlenwert ein sondern klicken auf das unscheinbare Funktionssymbol rechts. Der Dialog verändert sich nun und sieht etwas merkwürdig aus (kann man vielleicht „Bug“ nennen):



In das Eingabefeld geben wir (unterstützt von einer Autocomplete-Funktion) nun ein:

Daten.NutDist+20mm

und klicken auf den oberen Ok-Button. „Daten“ ist der Name unseres Spreadsheets. NutDist der im Spreadsheet festgelegte Parameter für den Abstand der Nuten. Groß-/Kleinschreibung spielt übrigens eine Rolle. Parameternamen und der Name des Spreadsheets müssen genau richtig eingegeben werden. Der Dialog ändert sich schon wieder:



Der berechnete Wert ist nun ausgegraut und kann nicht mehr direkt verändert werden. Was ja auch korrekt ist – er hängt ja nun von unserem Spreadsheet ab. Ein Klick auf „OK“ und unsere Zeichnung verändert sich wieder.

Fehlt noch die Breite. Wie breit muss das Teil sein? Im Innenraum brauchen wir Platz für die Befestigungsschrauben. Deren Abstand steht im Spreadsheet (MSDist). Zusätzlich brauchen wir links und rechts Platz für die Schraubenköpfe. Also kommen 2 mal MSHeadRad hinzu. Der Innenraum (die Tasche) wird als $MSDist + 2 * MSHeadRad$ breit. Oder? Moment. In der Tasche soll ja auch noch ein Pulley laufen. Die Tasche darf auf keinen Fall kleiner als der Durchmesser dieses Pulley sein. Besser etwas (2 mm) größer. Also tragen wir im Spreadsheet zunächst mal den Durchmesser des Pulles nach und benennen den Parameter „PulleyDia“. Die Taschenbreite ergibt sich damit zu

$$=\max(MSDist + 2 * MSHeadRad; PulleyDia + 2mm)$$

Leider klappt das in der aktuellen Version von FreeCAD nicht und das Ergebnis ist 0... Kein Problem machen wir direkt darunter zwei Behelfszeilen (bei mir Zeile 16 und 17) und tragen dort ein:

$$=MSDist + 2 * MSHeadRad$$

Und

$$=PulleyDia + 2mm$$

In das Feld für die Taschenbreite schreiben wir dann:

=max(A16;A17)

Vielleicht brauchen wir dieses Maß später nochmal. Vielleicht macht es Sinn diese Formel im Spreadsheet zu „verewigen“. Wir tragen dort also in ein Feld die Formel $MSDist + 2 * MSHeadRad$ ein, schreiben einen Kommentar dahinter und vergeben den Alias Taschenbreite und tragen auch noch einen Parameter für die Taschentiefe (names „Taschentiefe“) zusätzlich ein. Wir erhalten etwa folgendes Spreadsheet:

	A	B
1	1.85 mm	Radius für Bohrung Motorbefestigung
2	3 mm	Radius für Schraubenkopf Motorbefestigung
3	2.7 mm	Radius Befestigungsbohrung
4	4.7 mm	Radius Schraubenkopf Befestigung
5	5 mm	Höhe Schraubenkopf Befestigung
6	42.3 mm	Breite/Höhe Motor
7	31 mm	Abstand Schrauben Motorbefestigung
8	11 mm	Durchmesser der Stufe am Motor vorne
9	80 mm	Abstand der Nuten in den Profile (Mitte untere Nut bis Mitte obere Nut)
10	10 mm	Tiefe der Tasche für Motorritzen und Riemen
11	14 mm	Durchmesser Zahnscheibe
12		
13	Abgeleitete Daten	
14	37 mm	Breite der Tasche
15	37 mm	Taschenbreite basierend auf Verschraubung
16	16 mm	Taschenbreite basierend auf Pulleydurchmesser

„Außen rum“ brauchen wir auch noch Material. Hier muss Platz für die Schrauben zur Befestigung am Rahmen sein. Da wir die Schraubenköpfe hier versenken möchten brauchen wir mindestens den Durchmesser der Schraubenköpfe. Zwei mal zwei mal $PSHeadRad$. Vergrößern wir diesen beidseitig um eine Düsenbreite des 3D-Druckers (bei mir 0,4 mm) damit die Senklöcher nicht direkt an den Rand des Bauteils stoßen und dazwischen mindestens eine Druckbahn liegt. Es macht sicher Sinn den Düsendurchmesser auch ins Spreadsheet einzutragen. Bitte selbstständig durchführen und als Alias „DuesenDia“ nutzen.

Zusammen ergibt sich nun also:

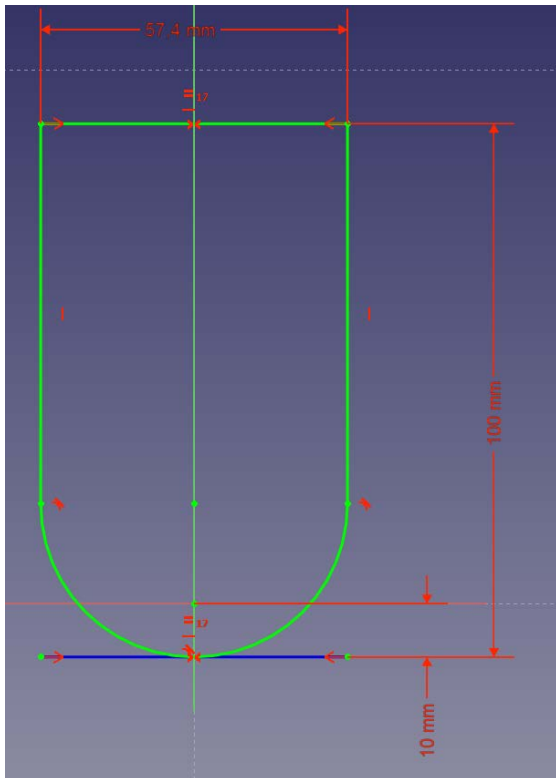
$Daten.Taschenbreite + Daten.PSHeadRad * 4 + Daten.DuesenDia * 4$

Diese Formel können wir nun mit dem „Length or Distance“ Werkzeug (Shift-d) an die obere vertikale Linie binden. Werkzeug auswählen, Linie anklicken, Funktionssymbol im Dialog anklicken. Formel eingeben – siehe oben.


Eigentlich sind wir nun fertig. Allerdings ist der Sketch noch nicht vollständig mit Constraints versehen. Unsere Hilfslinie hat noch keine festgelegte Länge. Das können wir ändern. Der Einfachheit halber nutzen wir wieder eine andere Constraint-Funktion. Equality Constraint. Damit können wir zwei Linie „gleichlang“ machen (oder zwei Radien gleichgroß):




Klicken wir damit auf die obere horizontale Linie und auf die Hilfslinie, dann färben sich unsere Linien grün. Damit signalisiert FreeCad das es nun genug von uns hat und keine weiteren Constraints mehr nötig sind:



Noch eine Anmerkung zur aktuellen Version des Sketchers: Wenn eine Zeichenfunktion ausgewählt ist (beispielsweise Linie) dann kann man keine Elemente mehr „frei“ selektieren. Das braucht man aber öfter mal um z.B. Maßzahlen „aus dem Weg“ zu schieben. Man kann dann einfach die rechte Maustaste drücken – dann ist der Cursor wieder „frei“ und man kann Elemente wieder einfach anklicken (auch zusammen mit der Strg-Taste für eine Mehrfachauswahl) und sie z.B. verschieben oder löschen (DEL-Taste).

Wir können den Sketch nun schließen (Close im Combo View,  in der Icon-Leiste oder Menü->Sketch->Leave sketch) und landen automatisch wieder in der Part Design Workbench.

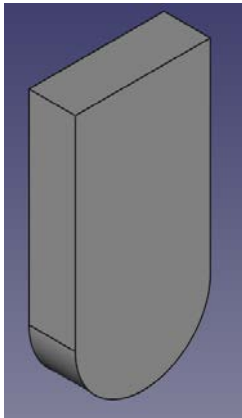
Mit der Funktion „Pad“ () können wir das Teil nun aufpolstern. Einfach mal ausprobieren. Standardmäßig wird eine Dicke von 10 mm eingestellt. Das ändern wir aber natürlich.

Wie „dick“ muss unser Teil nun sein? Zunächst die 10 mm Taschentiefe aus dem Spreadsheet. Dann brauchen wir Material um den Schraubenkopf versenken zu können: PSHeadH. Ein wenig mehr schadet nicht – soll ja alles stabil sein. Also packen wir noch zwei Millimeter drauf. Zusammen:

$$\text{Daten.Taschentiefe} + \text{Daten.PSHeadH} + 2 \text{ mm}$$

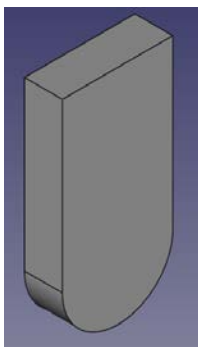
In der Combobox fällt uns beim Pad-Befehl sofort auf das es auch hier einen Funktion-Button gibt. Drauf klicken und die Formel eingeben wie wir das bei den Constraints auch schon gemacht haben.

Wunderbar *g* Schnell auf OK klicken und das erste „greifbare“ Ergebnis bewundern:

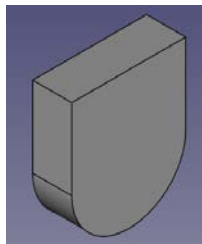


Das hat doch schon viel Schönes...

Jetzt wollen wir aber erstmal testen ob das mit dem Spreadsheet auch tatsächlich funktioniert. Also öffnen wir das Spreadsheet und ändern einfach mal den Abstand der Nuten auf 40 mm. Spreadsheet schließen und staunen:

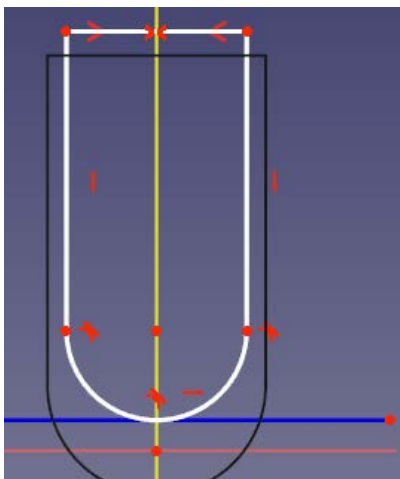


Vorher:



Nachher:

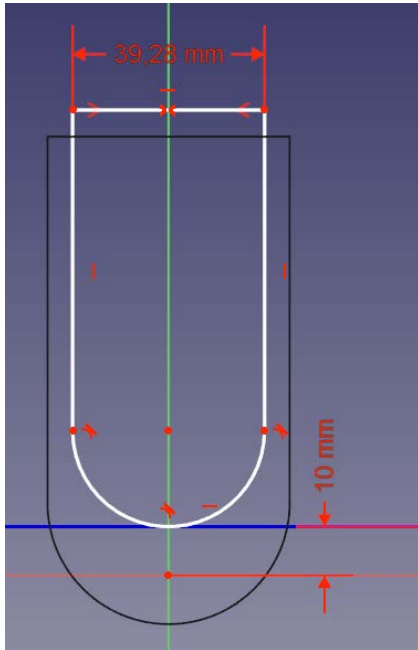
Es funktioniert! Schnell wieder auf 80 mm zurückändern und weiter geht es mit der „Tasche“. Hierzu brauchen wir wieder einen Sketch. Auch dieser Sketch liegt auf der YZ-Ebene und auch hier brauchen wir keine Referenzen zu externen Geometrien. Also: Sketch erzeugen (darauf achten das unser Body aktiv ist). Im View-Menü schalten wir nun am besten auf Wireframe um (View->Draw style->Wire frame) – sonst sehen wir wenig weil das Bauteil unsere Konstruktionsebene verdeckt. Wir bauen nun im Prinzip dieselbe Geometrie wie beim ersten Sketch bis zur Hilfslinie – nur ein wenig kleiner und etwas verschoben:



Der Kreisbogen bei der Tasche soll unten der Oberkante des Nutprofils entsprechen. Anders ausgedrückt: Der tiefste Punkt des Kreisbogens soll 20 mm über dem unteren Rand des Bauteils liegen. Das lösen wir wieder mit einem vertikalen Abstandsconstraint. Wir wissen ja noch dass der Nullpunkt 10 mm über dem unteren Rand des Bauteils liegt. Wir müssen also diesmal 10 mm nach oben (vom Nullpunkt aus).

Die Länge der oberen Linie steht schon im Spreadsheet: Daten.Taschenbreite

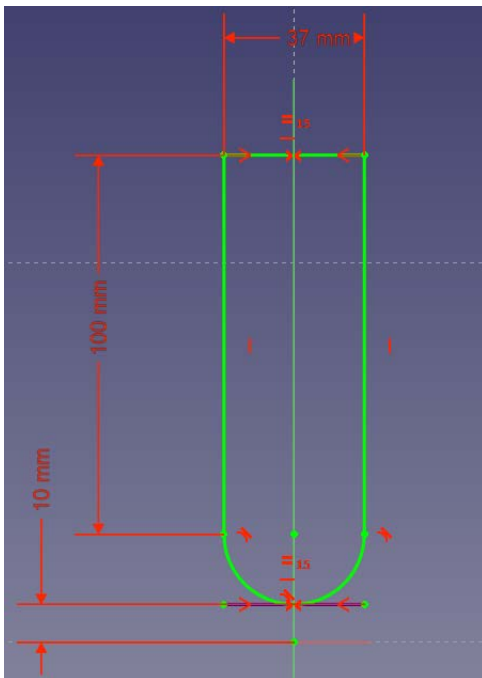
Das Ergebnis bis hierher:




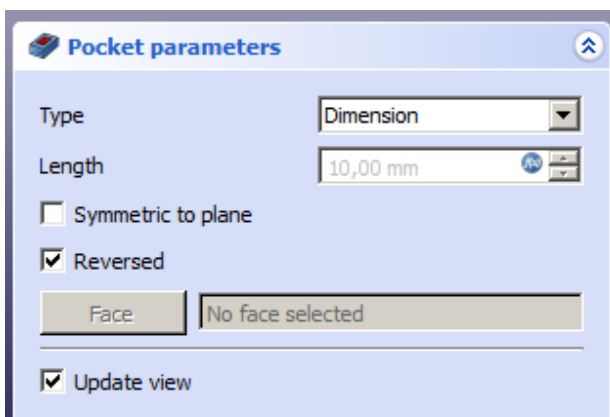
Auch hier können wir die Hilfslinie wieder genauso lang machen wie die obere Linie und die Hilfslinie symmetrisch zur grünen Achse machen. Fehlt noch die Höhe. Hier sollten wir großzügig sein. Beim „durchlöchern“ von Teilen ist es immer besser, wenn man nicht „Kante auf Kante“ konstruiert – manchmal kommt es dann zu Rechenfehlern. Daher setzen wir die Höhe der linken senkrechten Linie einfach auf die Gesamthöhe des Bauteils. Wir erinnern uns:

Daten.NutDist+20mm

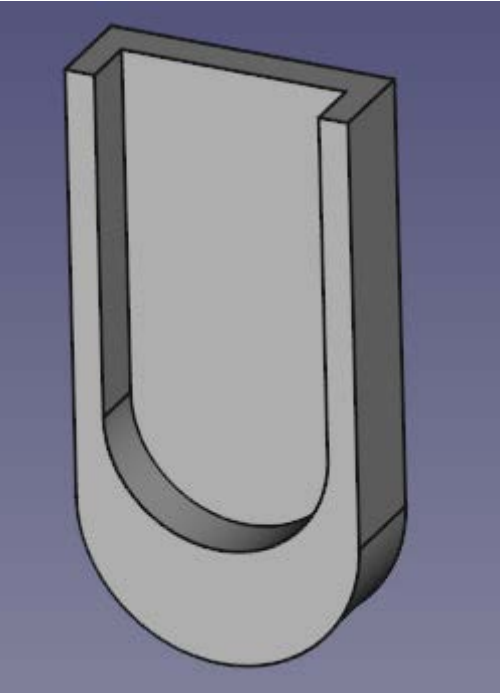
Damit ist der Sketch komplett konstruiert:



Wir schließen nun den Sketch und rufen die Funktion „Pocket“ auf ( , Menü Part Design->Pocket) und geben als „Length“ die Taschentiefe (Daten.Taschentiefe) ein (Vorher: Funktionssymbol). Bevor wir auf OK klicken schauen wir uns die Vorschau des Bauteils schonmal an. ??? Klappt nicht... Klar: Taschen funktionieren „andersrum“. Ausgehend von unserem Sketch wurde „nach oben“ aufgepolstert. Taschen arbeiten „nach unten“. Wir können das aber ändern indem wir den Haken bei „Reversed“ setzen:



Nun klappts:



Als Nächstes konstruieren wir die Öffnung für die Stufe vorne am Motor. Wieder brauchen wir einen Sketch. Diesmal bauen wir uns aber zuerst eine neue Bezugsebene. Datum plane genannt.



Findet sich natürlich auch im Part-Design Menu als „Create a datum plane“.

In der Combo View erscheint daraufhin ein fürchterlich kompliziertes Formular:

Combo View

Model Tasks

OK Cancel

Plane parameters

Not attached

Selecting... No reference selected

Reference2 No reference selected

Reference3 No reference selected

Reference4 No reference selected

Attachment mode:

- Translate origin
- Object's XY
- Object's XZ
- Object's YZ
- Plane face
- Tangent to surface
- Normal to edge
- Frenet NB
- Frenet TN
- Frenet TB
- Concentric
- Revolution Section
- Plane by 3 points
- Normal to 3 points

Extra placement (inactive - not attached):

X: 0,00 mm

Y: 0,00 mm

Z: 0,00 mm

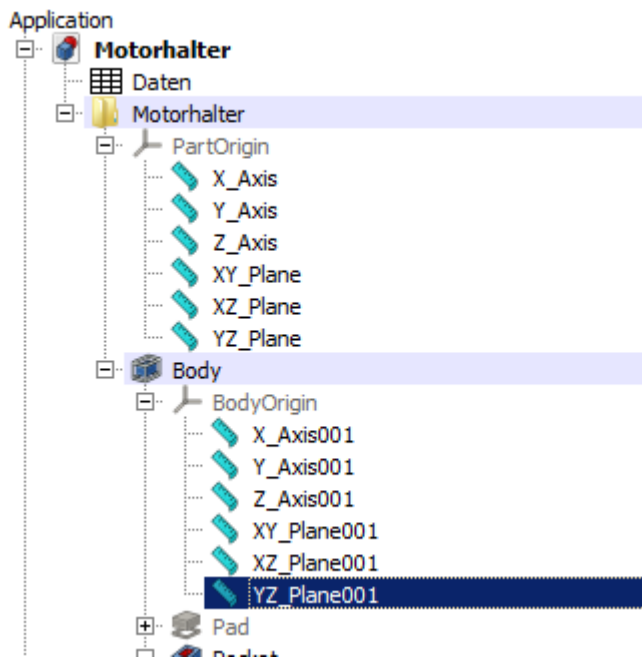
Yaw: 0,00 °

Pitch: 0,00 °

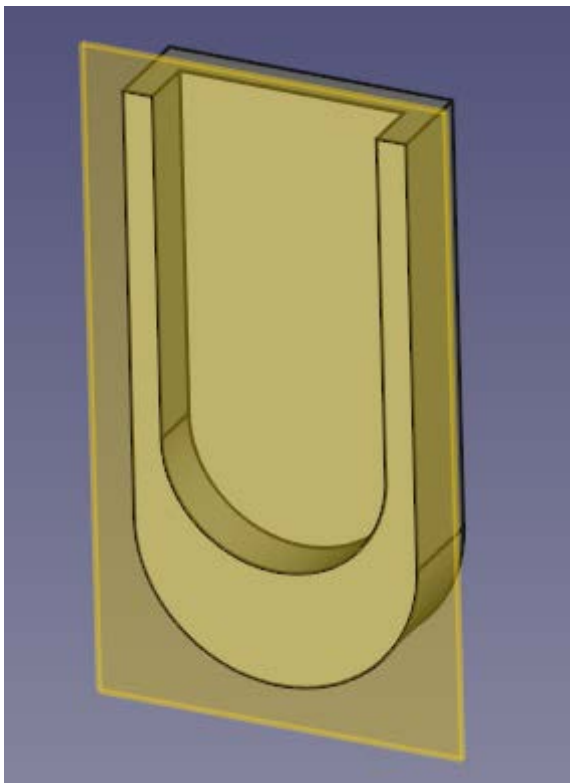
Roll: 0,00 °

Flip sides

Am besten sofort auf „Cancel“ drücken... Das ist viel zu kompliziert für den Anfang. Was genau wollen wir eigentlich? Wir brauchen eine Ebene die Parallel zur bereits vorhandenen YZ_Plane001 ist. Also klicken wir im Tree View auf diese Ebene:



Und dann nochmal auf „Create a datum plane“. Das Hauptfenster sieht nun so aus:



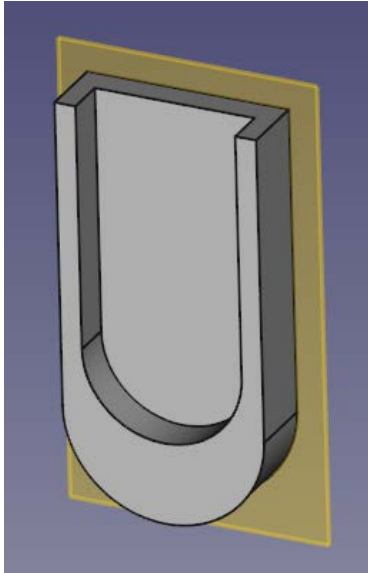
Das Beige-Braune Ding ist unsere neue Ebene. Sie ist an die vorher ausgewählte Ebene gekoppelt, soll aber einen gewissen Abstand davon haben. Den Abstand kennen wir noch vom Pad-Befehl beim ersten Sketch (da wir diese Daten nun schon zum zweiten Mal brauchen könnten wir dafür wieder eine Formel im Spreadsheet nachtragen – das sparen wir uns aber diesmal):

Daten.Taschentiefe + Daten.PSHeadH + 2 mm

Ein Formelsymbol gibt es in dem Formular auch bei „Extra Placement“ für X, Y und Z. Da wir einen Versatz senkrecht zur Ebene brauchen müssen wir die Formel bei Z eintragen:

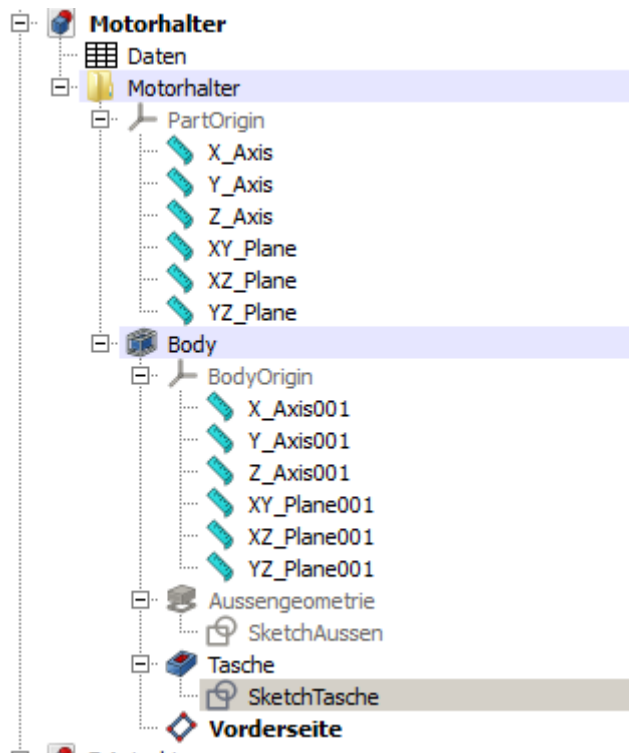
Daten.Taschentiefe + Daten.PSHeadH + 2 mm

Im Grafikfenster sieht das dann so aus:



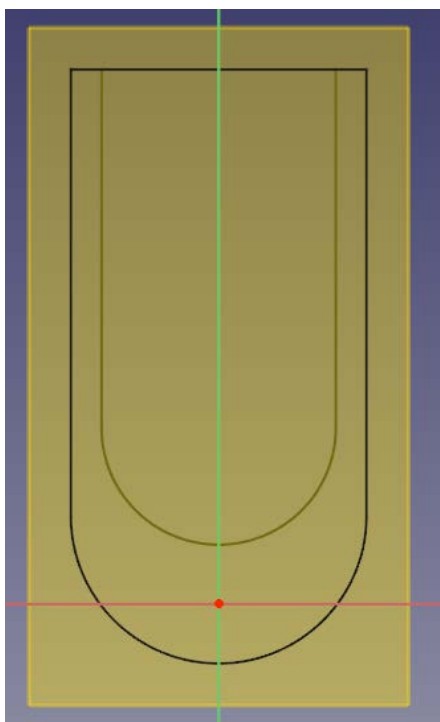
Scheint zu passen. Schnell auf OK klicken. Die neue Ebene erscheint im Tree view unter unserer „Pocket“ mit dem nichtssagenden Namen „DatumPlane“. das kann so nicht bleiben. Benennen wir sie um in „Vorderseite“.

Bei der Gelegenheit können wir auch gleich „Pad“ und „Pocket“ sinnvollere Namen geben und auch die Sketche können „bessere“ Namen gebrauchen. Das erhöht – besonders bei komplexen Konstruktionen die Übersichtlichkeit deutlich:




Grundsätzlich sind diese Ebenen ein sehr mächtiges Feature. Allerdings aktuell auch noch stark in der Entwicklung. Da gibt es unendlich viele Möglichkeiten weitere Ebenen zu definieren. Aber das wäre Stoff für ein eigenes Tutorial.

Nun geht es aber weiter mit der Öffnung für die „Stufe“ vorne am Motor. Wieder brauchen wir einen neuen Sketch. Diesmal klicken wir vorher im Tree View auf die neue Ebene „Vorderseite“ und dann erst auf „Create a new sketch“. Es ergibt sich folgendes Bild (Wireframe):



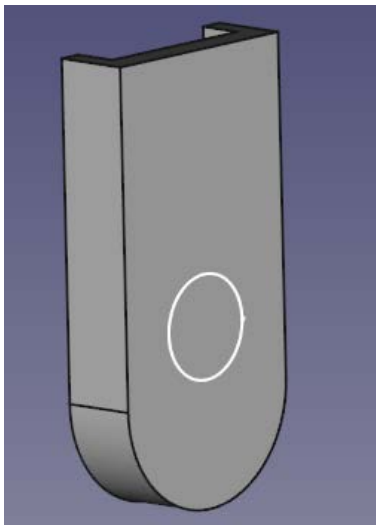
Im Treeview markieren wir „Vorderseite“ und blenden die Ebene aus – die stört nur. Das geht am einfachsten mit der Leertaste – rechte Maustaste->Hide selection geht aber auch.

Wir konstruieren nun einen Kreis und „constrain“ den Mittelpunkt. Dieser muss „10mm+Daten.Taschenbreite/2“ +über dem Nullpunkt des Objekts sitzen und außerdem auf der grünen Achse sitzen.

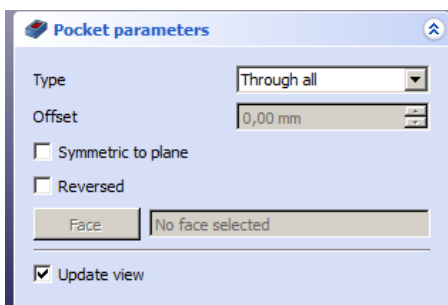
Den Radius des Kreises setzen wir mittels „Radius constraint“  (Taste Shift-r) auf den Wert aus unserem Spreadsheet (plus 0,5 mm für etwas „Spiel“):

Daten.MotorStufeRad+0,5 mm

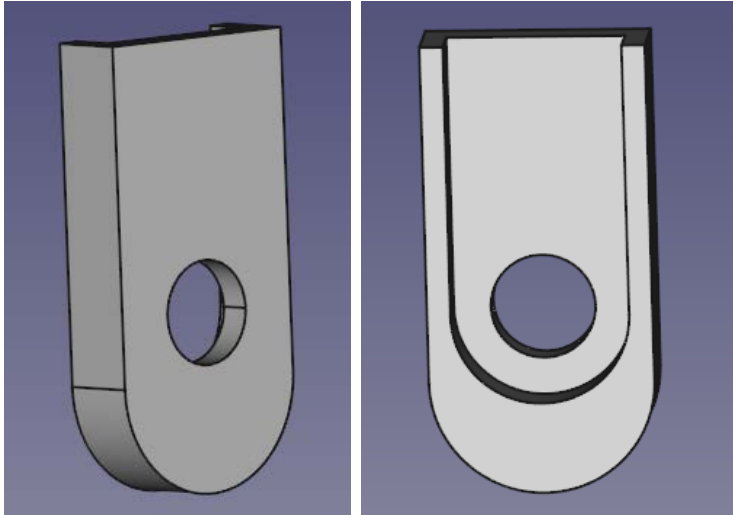
Alles grün? Dann können wir den Sketch schließen und uns das Zwischenergebnis ansehen:





Wir brauchen ein Loch. Pocket? Pocket! Erst den Sketch markieren (z.B. im Tree View – geht aber auch direkt in der Grafik) und dann die Pocket-Funktion starten. Default ist hier 5 mm. Reicht nicht. Egal. Diesmal wollen wir ohnehin keine Zahlen eingeben. Formeln auch nicht. Bei „Type“ im Combo View können wir aber auch „Through all“ auswählen:

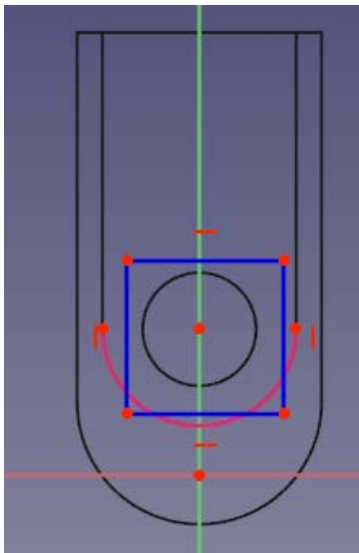



Das passt. Schnell auf OK klicken und – voila:



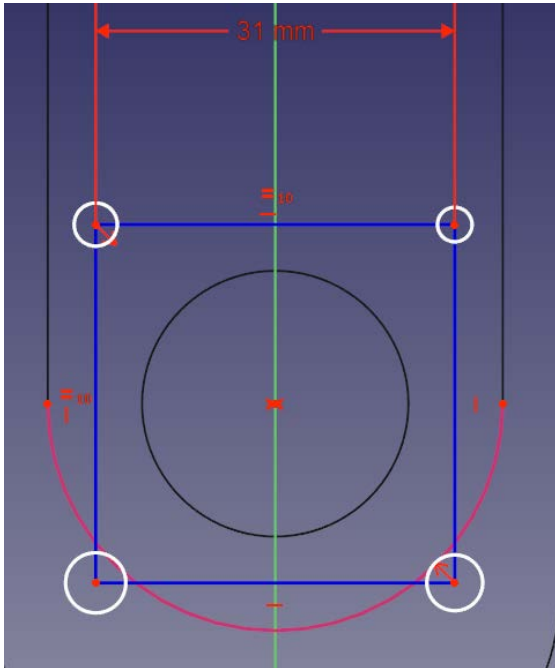
Hmm. Die Löcher für die Schrauben zur Motorbefestigung brauchen wir ja auch noch. Also schnell noch einen Sketch auf der Ebene „Vorderseite“ erstellen. Wieder die Ebene ausblenden. Und erstmal einen Mittelpunkt konstruieren (Hilfslinienmodus, Punkt ). Der Punkt soll die Motorachse darstellen. Also wieder $10 + \text{Taschenbreite}/2$ über dem Nullpunkt liegen und auf der grünen Linie.


Nun konstruieren wir ein Hilfslinien-Quadrat :

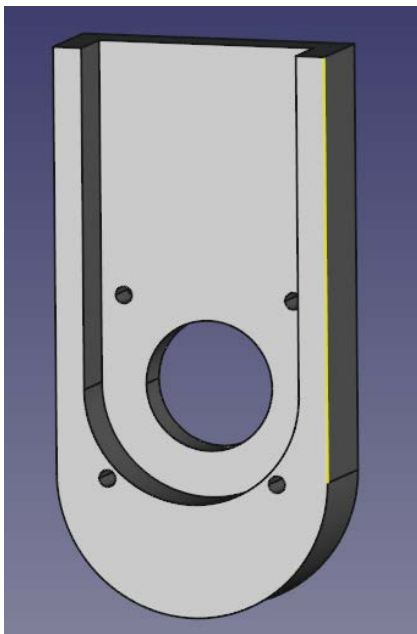


Mit der Symmetriefunktion klicken wir diesmal der Reihe nach auf die Ecke links oben, den eben erstellten „Motorachsenpunkt“ und die Ecke rechts unten. Damit bauen wir eine „PunktSymmetrie“ um den Kreismittelpunkt. Mit dem Equality Constraint machen wir aus dem Rechteck ein Quadrat (eine horizontale und eine vertikale Seite anklicken). Dann noch die Länge einer Seite auf den Bohrungsabstand aus dem Spreadsheet setzen (Daten.MSDist). Nun schalten wir zurück auf „richtige Linien“  und setzen Kreise auf die Ecken des Quadrats. Die Mittelpunkte entweder per „Auto

constraint“ (Combo view) oder manuell per „Coincident constraint“ auf die Ecken des Quadrats „zwingen“. Das sieht dann ungefähr so aus:




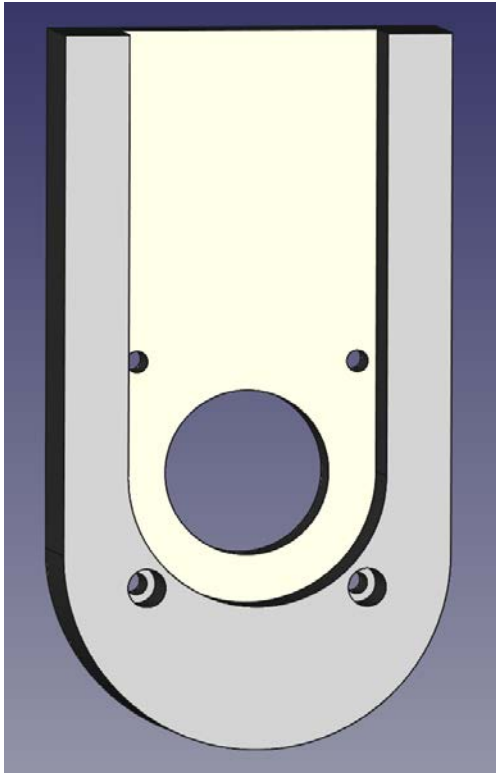
Um nun die Kreise zunächst alle gleich groß zu machen hilft wieder der Equality Constraint . Noch schnell den Radius eines der Kreise auf Daten.MSRad aus dem Spreadsheet setzen und die Kreise werden (hoffentlich) grün. Sketch schließen und wieder eine Pocket aus dem Sketch erstellen (dran denken: Vorher Sketch markieren) mit „Through all“:





Damit die unteren Schrauben nicht so lang sein müssen (und weil die Schraubenköpfe den Profilen im Weg wären) müssen wir noch „Taschen“ für die Schraubenköpfe machen. Wieder ein neuer Sketch. Diesmal mit YZ_Plane001 als Basis – denn wir wollen ja wieder von der anderen Seite aus „schneiden“. Damit wir etwas sehen müssen wir wieder auf „Wire Frame“ umschalten. Die Mittelpunkte der Taschen könnten wir von den bereits vorhandenen Bohrungen übernehmen. Das ist

aber unzuverlässig da die interne „Nummerierung“ der Kanten sich manchmal ändert. Besser wir konstruieren unser Quadrat inkl. Motorachspunkt nochmal. Geht ja schnell.

Auf den unteren beiden Punkten zwei Kreise zeichnen. Die Kreise mit  „gleich machen“ und als Radius für einen Kreis Daten.MSHeadRad eintragen. Pocket Funktion (diesmal wieder „Reversed“). Als Länge aus dem Spreadsheet Daten.Taschentiefe einstellen. Ergibt ungefähr das:



Weiter geht's wieder mit der Ebene „Vorderseite“ und den Löchern für die Befestigungsschrauben. Wir brauchen 3 Kreise. Den ersten ziehen wir direkt um den Nullpunkt. Die beiden anderen Kreise legen wir oben links und rechts der Mittellinie an. Alle Kreise „gleich machen“ . Die Mittelpunkte der oberen Kreise symmetrisch zur Mittellinie ausrichten . Den Radius eines der Kreise auf Daten.PSRad setzen. Vertikaler Abstand zwischen oberen Kreisen und unterem Kreis (jeweils Mittelpunkt zu Mittelpunkt) auf Daten.NutDist setzen. Beim horizontalen Abstand müssen wir uns wieder an die Formel zur Breite aus dem ersten Sketch erinnern

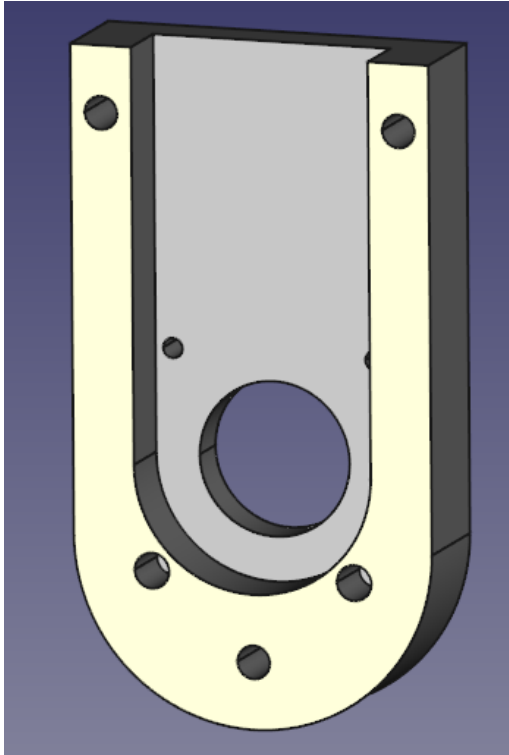
$$\text{Daten.Taschenbreite} + \text{Daten.PSHeadRad} * 4 + \text{Daten.DuesenDia} * 4$$

Diesmal brauchen wir aber nur

$$\text{Daten.Taschenbreite} + \text{Daten.PSHeadRad} * 2 + \text{Daten.DuesenDia} * 2$$

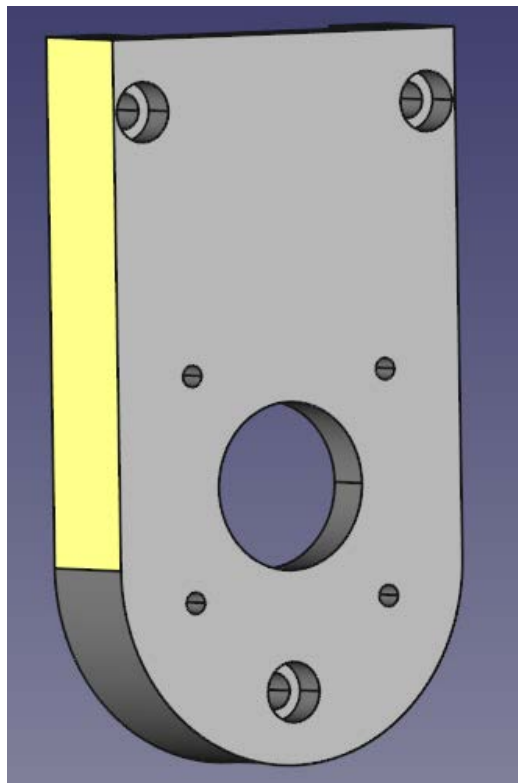
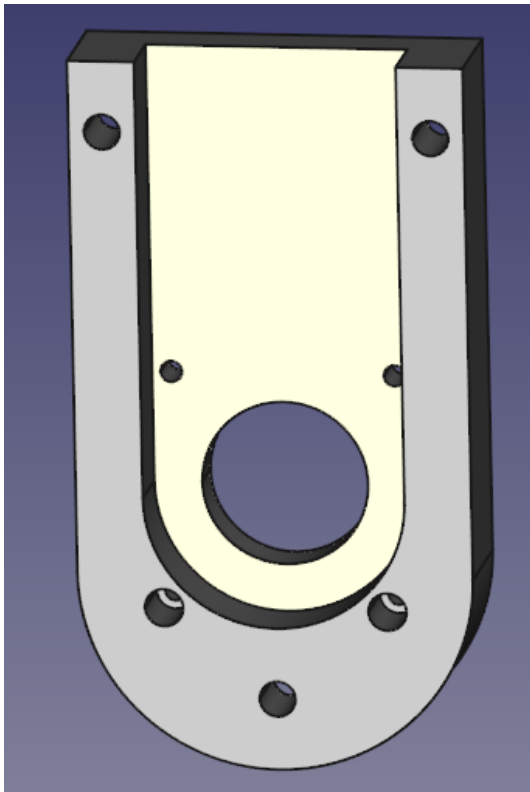
weil die Bohrungen ja um jeweils Daten.Duesenbreite + Daten.PSHeadRad nach innen versetzt sind.

Fertig. Wieder „Pocket“ mit „Through all“.



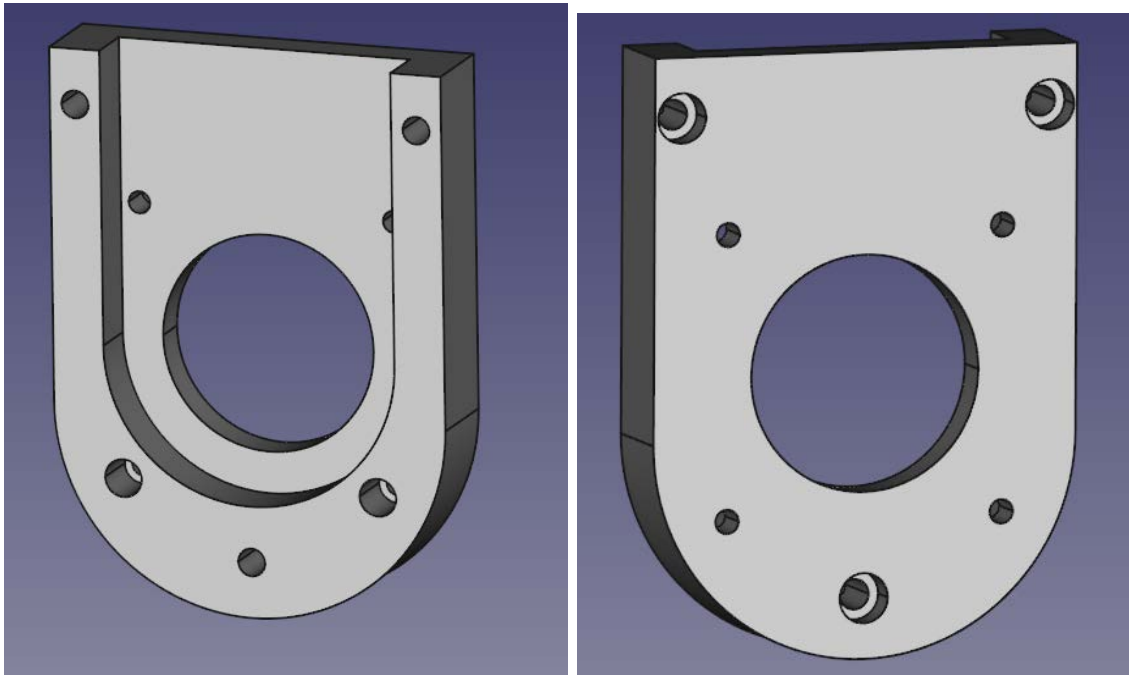
Und auf zum nächsten Sketch um die Schraubenköpfe zu versenken. Wieder brauchen wir „Vorderseite“ als Basisebene. Die Mittelpunkte genauso konstruieren wie bei den Löchern. Radius gleichschalten. Einen Radius auf Daten.PSHeadRad setzen. Sketch schließen. Pocket. Length = Daten.PSHeadH.

Fertig ist das gute Stück:



Schauen wir mal ob die Veränderung über das Spreadsheet funktioniert und bauen das Teil auf Nema23 um. Es ändern sich folgende Maße:

- Radius der Stufe: 19 mm
- Breite/Höhe des Motors: 56,4 mm
- Abstand Schrauben Motorbefestigung: 47,14
- Befestigungsschrauben: M3->M4
Das heißt: MSRad = 2 mm, MSHeadRad=3,4 mm



Deutlich bulliger geworden das Teil... aber offenbar hat alles so funktioniert wie es soll.

Nun ist die Umstellung von Nema 17 auf Nema 23 ein ziemlich großer Umbau. Viel häufiger wird man andere Änderungen brauchen. Besonders im 3D-Druck. Bekanntermaßen werden Kreise ja beim 3D-Druck gerne mal kleiner als geplant weil die Software Kreise durch Vielecke annähert und das Filament sich beim Druck nach innen zieht.

Man muss also insbesondere die Durchmesser der Schrauben, Löcher für Schraubenköpfe etc. anpassen.

Wir haben hier im Spreadsheet zum Beispiel die tatsächlichen Durchmesser der Schrauben eingetragen. Dummer Fehler... Bei meinem Drucker und meinem aktuell genutzten Filament passt das ganz und gar nicht. Da muss ich für die Nema17 Version folgende Werte einstellen:

	A	B
1	1.7 mm	Radius für Bohrung Motorbefestigung
2	3 mm	Radius für Schraubenkopf Motorbefestigung
3	2.7 mm	Radius Befestigungsbohrung
4	4.7 mm	Radius Schraubenkopf Befestigung
5	5 mm	Höhe Schraubenkopf Befestigung
6	42.3 mm	Breite/Höhe Motor
7	31 mm	Abstand Schrauben Motorbefestigung
8	11 mm	Durchmesser der Stufe am Motor vorne
9	80 mm	Abstand der Nuten in den Profile (Mitte untere Nut bis Mitte obere Nut)
10	10 mm	Tiefe der Tasche für Motorritzen und Riemen
11		

Bei den Befestigungsschrauben für den Motor würde ich beim realen Teil sogar auf 1,8 oder 1,9 mm gehen damit Ungenauigkeiten und Toleranzen ausgeglichen werden. Aber das ist mit dieser Konstruktion ja kein Problem. Passt etwas nicht kann man es häufig direkt im Spreadsheet anpassen. Die gesamte Konstruktion ändert sich automatisch.

Oder ein häufiger Fall: Mir sind die Hammermuttern mit M5-Gewinde ausgegangen. Bestellung aus China dauert... Kein Problem. Ändern wir die Befestigung auf M4 - da habe ich noch einen Beutel Hammermuttern.

Zugegeben:

So zu konstruieren ist nicht ganz einfach. Man muss viel mit Formeln arbeiten. Iiiiih! Mathematik! Man muss sich sehr viele Gedanken machen um Teile sauber zu konstruieren. Andererseits: Gerade das macht das „Konstruieren“ ja eigentlich aus. Wir „basteln“ uns ja nicht einfach Teile zusammen sondern das soll ja alles irgendwie sinnvoll sein. Bei einer guten Konstruktion hat „Alles“ seinen Sinn. Wenn kein Teil mehr dicker ist als nötig. Kein Maß mehr einfach willkürlich.

ANTOINE DE SAINT-EXUPÉRY: Perfektion ist nicht dann erreicht, wenn man nichts mehr hinzufügen, sondern nichts mehr weglassen kann

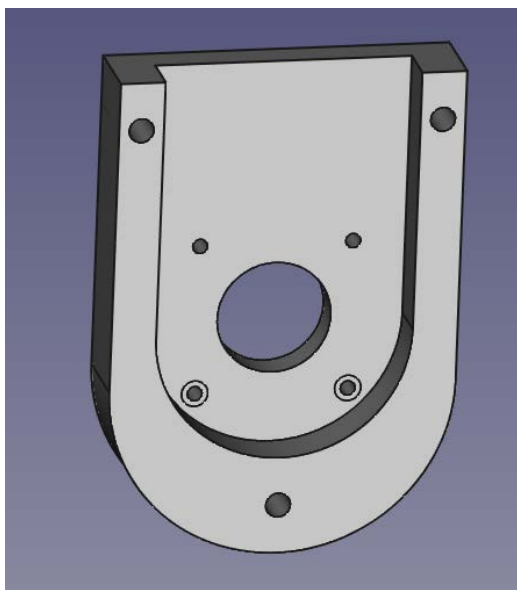
Man muss sich auch bei nicht-parametrischer Konstruktion überlegen: Warum will ich diese Kante nun 20 mm lang haben? Woher kommt diese Zahl? Einfach „nach Gefühl“? Da läuft man ganz schnell vor die Wand.

Parametrische Konstruktion zwingt den Konstrukteur dazu über Zusammenhänge und Maße genauer nachzudenken. Ist das ein festes Maß? Hängt es von anderen Maßen der Konstruktion ab? Oder ist es nur ein Mindestmaß das ich aus statischen oder optischen Gründen möchte?

Der Lohn der Mühe ist dann eine Konstruktion die – im Idealfall – voll parametrisiert ist. Alle wesentlichen Daten finden sich im Spreadsheet zusammengefasst und dann ist eine Änderung z.B. des Motors in diesem Fall eine Sache von Sekunden.

Oder wir wollen den üblichen GT2-Zahnriemen gegen eine anderen Riemen tauschen. Der ist breiter (Taschentiefe ändert sich) und wir finden nur Pulley mit 50 mm Durchmesser. Geht das überhaupt?

Klar! Über die MAX-Funktion bei der Konstruktion der Taschenbreite haben wir das bereits „vorhergesehen“. Das Ergebnis sähe so aus:



Die Senkungen für die unteren Motorbefestigungsschrauben sind jetzt zwar überflüssig weil die Tasche viel größer geworden ist. Macht aber nichts. FreeCAD macht nun halt zwei „Löcher“ in die Luft – man erkennt es noch an den Ringen um die Bohrungen. Die Konstruktion funktioniert trotzdem.

Man sieht: Ein sauber durchdachtes und durch**konstruiertes** Bauteil hat gegenüber einer „freien“ Konstruktion einige Vorteile. Natürlich kann man nicht immer im Vorfeld alle Eventualitäten berücksichtigen und muss Teile gelegentlich trotzdem neu konstruieren. Einfache Änderungen von Schraubendurchmessern lassen sich aber in der Regel sehr einfach „abfangen“ so dass in solchen Fällen keine Neukonstruktion nötig ist.

Je mehr „Verstand“ man in die Konstruktion einfließen lässt, desto besser wird das Teil. Man überlegt sich am besten schon im Vorfeld welche Varianten für ein Bauteil ggf. in Frage kommen und stimmt die Konstruktion darauf ab.