

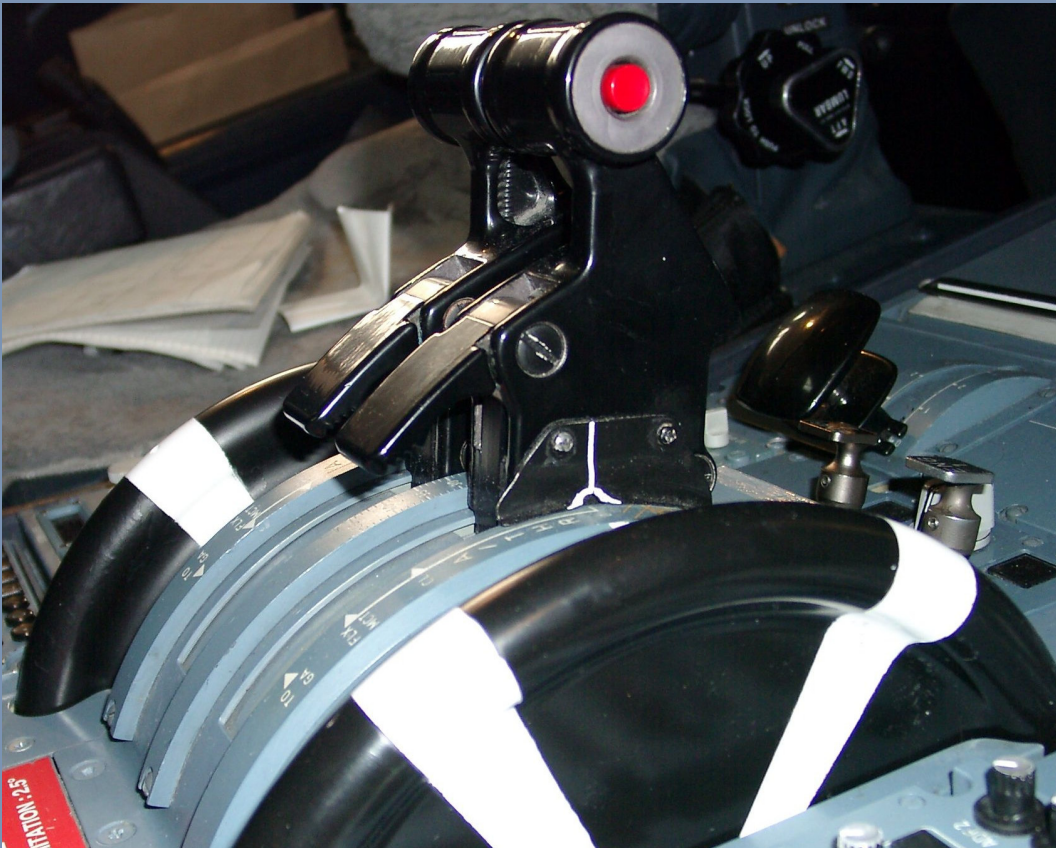
Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Thrust Lever

Der Nachbau des Thrust Levers war eines der aufwendigsten Projekte. Da weder eine Drehbank noch eine Fräse zur Verfügung standen mussten alle Teile "aus den Vollen geschnitzt" werden.

Doch zunächst mal das Original...



Der Thrust Lever besteht aus folgenden Komponenten:

- Unabhängig voneinander laufende Schubachsen und Reverserachsen
- Ein A/THR Disconnect Schalter je Schubhebel
- Motorisiertes Trimmrad mit Trim/CG Scala, beweglich

Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die Schubachsen verfügen über sog. Detents (Rasten) für CL (Climb), FLX/ MCT (Flex und max. Continues) und TOGA (Takeoff/ Go around).

Der Bereich zwischen Idle und CL wird als manueller Schubbereich verwendet und arbeitet annähernd linear bis zum vom FADEC errechneten Climb Thrust.

Autothrust steht nur in den Detents zur Verfügung und wird anhand der Performance Daten vom FADEC errechnet und bereitgestellt, mal ausgenommen Reduced Thrust.

Aktiviert wird das Autothrust System entweder durch das Bewegen des Schubhebels auf die FLEX oder TOGA Position oder durch Betätigen des ATHR Schalters in der Mitte der FCU.

Autothrust disconnect findet statt, wenn der Schubhebel auf Idle gezogen wird, der ATHR Schalter betätigt wird oder einer der ATHR Disconnect Schalter zweimalig am Schubhebel betätigt wird.

Zum Einstellen der Trimmung wird vor dem Start entsprechend dem Load und Trimsheet das Trimmrad in die gewünschte Richtung gedreht, bis der Trim entweder auf der Scala des Thrust Levers oder der auf der F/CTL Page des unteren ECAM's korrekt angezeigt wird.

Nach dem Start wird automatisch das Autotrimmsystem aktiv, der Pilot muss dann nicht mehr die entsprechende Fluglage nachtrimmen.

Wird bei aktivem Autotrim am Trimmrad gedreht, disconnected der Autopilot (wenn eingeschaltet) und Autotrimmsystem wird deaktiviert.

Das Autotrimmsystem schaltet wieder zu, sobald das Trimmrad losgelassen wird. Der Autopilot muss dann aber manuell wieder zugeschaltet werden.

In dem Projekt wurden folgende Funktionen umgesetzt:

- Zwei Getrennte Schubhebelachsen (beim Wilco funktioniert nur eine Achse)
- Reverser Funktion mit Arretierhebel auf beiden Achsen
- ATHR Disconnect Schalter an beiden Schubhebeln
- Zuordnung der Detents und deren Funktionen (Autothrust)
- Trimmung über die Trimmräder mit einem Drehencoder
- AP und Autotrim disconnect über die Trimmräder

Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Der Nachbau



Projekt Airbus A320 das virtuelle Cockpit

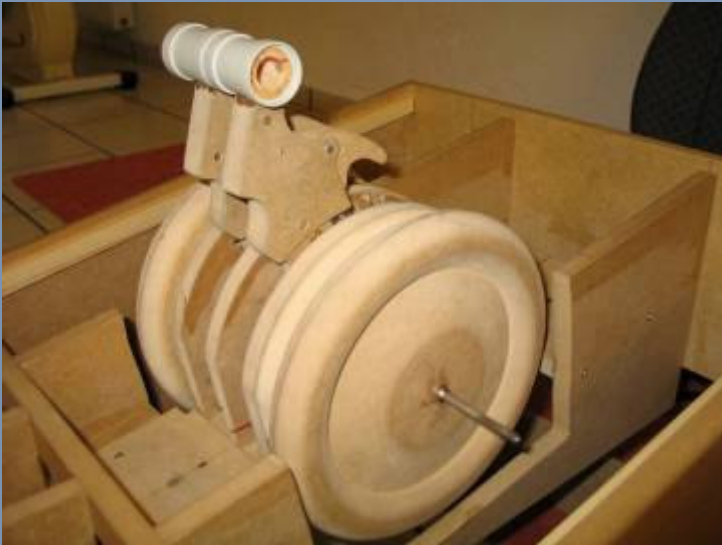
Ein Projektbericht von: Frank Sommer

Um die Frage nach Zeichnungen vorab zu klären – **nein, es existieren keine Zeichnungen**. Alle Teile wurden grob anhand der Maße des Originals abgemessen, hergestellt und eingepasst.

Eine gute Seite mit Originalbildern und Maßen findet ihr hier:

<http://www.simpit.de/>

Die Konstruktion besteht im Wesentlichen aus MDF und wurde in das Pedestal integriert.



Die Wellen (6 mm) der beiden Schubhebel sind jeweils in Kugellager gelagert, so dass jeder Schubhebel sich separat bewegen lässt und die Welle rechts und links hervorsteht.



Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die Detents wurden auf die Innenseiten des mittleren Steges gebohrt. An den Schubhebeln befinden sich je zwei 7 mm Stahlkugeln, federbelastet. Die zweite dient zur Erhöhung des Reibungswiderstandes.



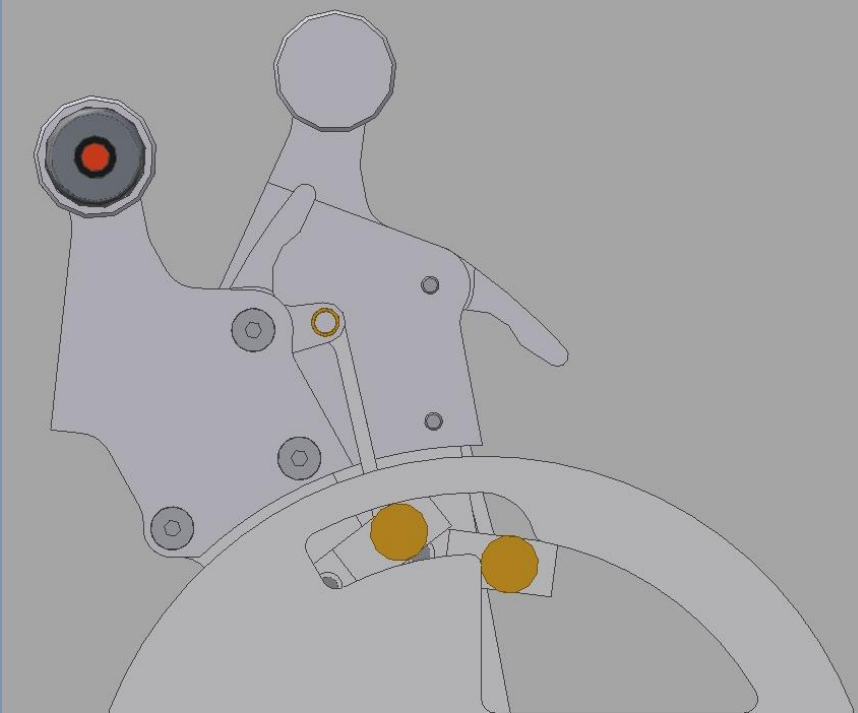
Die Reverser Mechanik läuft über eine Führung in den beiden Außenstegen und kommt in der Idle Position zur Anlage. Mit dem Hebel hebt sich die Rolle und wird in eine darüberliegende Bahn geführt.



Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die Führungen für die Schubumkehr wurden aus dieser Zeichnung abgemessen und so lange vergrößert bis die originalen Maße übereinstimmten.



Die Potis der beiden Schubachsen befinden sich rechts und links befestigt an den überstehenden Wellen der Schubhebel. Das Trimmrad wird auf einer 7 mm starken Messinghülse ebenfalls auf dieser Welle gelagert.

Die beiden Trimmräder waren ursprünglich mit einer Welle und Plastikzahnradern verbunden. Leider hielt das nicht lange und die Zahnräder bekamen heftige Zahnschmerzen. Mit dem Zahnrad des rechten Trimmrades ist der Drehencoder für die Trimmung verbunden.



Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die elektronische Einbindung der Schubachsen wurde durch einen 2-Achsen Billigjoystick bewerkstelligt. Davon habe ich vor einiger Zeit über Ebay 20 defekte Geräte für 5 € Gesamtpreis erworben.

Was benötigt wird sind lediglich die Potentiometer und der USB Controller. Die X und Y-Achse wird dann mit den jeweiligen Schubhebeln verbunden und über FSUIPC kalibriert.

Wichtig hierbei ist die Position der Detents. Die mussten bei dem Wilco fest definiert werden in Relation zur gesamten Bewegung des Schubhebels einschl. Reverser.

Grundsätzlich lassen sich die Schubhebelachsen auch auf andere Modell kalibrieren. Am einfachsten erscheint die Kalibrierung bei variablen Detents, was der Wilco leider nicht hat.

Die ATHR Disconnect Schalter sind separat an eine FSBUS IO Platine angeschlossen und über ein Mausevent in der Syntax mit der FCU verknüpft.

Grundsätzlich könnte man über das Basicprogramm auch die zweifache Betätigung der ATHR Disconnect Schalter bewerkstelligen, wir haben das allerdings bislang noch nicht umgesetzt.

Kalibrierung und Systemlogik

Der Wilco unterstützt leider nur eine Schubachse und bedient sich der DirectX Eingabe eines Gamecontrollers.

Alle Versuche die Achsen zu separieren endeten mit einem nicht synchron laufendem Engine Display. Die weißen Dots (Donats) liefen einfach nicht mehr synchron mit dem tatsächlichen Schub.

Eine weitere Aufgabe die es zu bewältigen galt ist die Umsetzung der Schubumkehr. Auch hier war es nicht möglich die von FSUIPC vorgegebene Reverser Achse zu verwenden.

Der gesamte positive Schub von Idle bis TOGA wurde über die FSUIPC folgendermaßen definiert:

```
[JoystickCalibration.Feelthere A320 CFM Condor VA]  
Throttle=-7471,16383
```

Wobei -7471 die Idle- und 16383 die maximale Schubposition definiert. In diesem Bereich läuft die Schubregelung absolut synchron mit dem Engine Display und FSUIPC generiert daraus 100 % DirectX Ausschlag.

Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Auf die X-Achse des Schubhebels wurde nun Schrittweise die Reverserfunktion in Form von Tastaturbefehlen auf bestimmte Bereiche der freien Achsenbewegung gelegt.

[Axes.Feelthere A320 CFM Condor VA]

```
0=1X,256
1=1X,U,9380,11381,65602,0
2=1X,D,9380,11381,65598,0
3=1X,U,11382,12381,65602,0
4=1X,D,11382,12381,65598,0
5=1X,U,12382,13381,65602,0
6=1X,D,12382,13381,65598,0
7=1X,U,13381,14382,65602,0
8=1X,D,13381,14382,65598,0
9=1X,U,14381,15382,65602,0
10=1X,D,14381,15382,65598,0
11=1Y,256
12=1Y,D,-8413,-7232,x01000810,x00
13=1Y,U,15018,16383,x01000810,x01
```

Unterteilt in jeweils einen Up- und Down Event unter Axis Assignment werden die EventID`s 65602 und 65598 für Reverser schrittweise ein bzw. ausfahren zugeordnet. Somit ergibt sich anstelle einer linearen Funktion das schrittweise Betätigen der Schubumkehr.

Eine andere Lösung kam bei der Verwendung des Wilco`s leider nicht in Frage.

Auf die Y-Achse (**gelb**) wurde entsprechend die Steuerung des Autothrust Systems gelegt. Als Downevent wird ab -8413 bis zum Idle (-7232) der FSUIPC Offset 0x0810 auf x00, also ATHR Arm, deaktiviert.

Aktiviert wird ATHR bei 15018, was ungefähr die FLEX Position des Schubhebels darstellt. Hier wird der Byte des Offsets 0x0810 auf x01 gesetzt.

Die ATHR Disconnect Schalter sind mit einer IO Platine direkt mit FSBUS verbunden und steuern einen Mausevent auf den ATHR Schalter auf der FCU. Das Panel des Wilco wurde entsprechend in einzelne Fenster zerlegt und fixiert, so dass immer der richtige Mausevent an der richtigen Stelle sichergestellt wird.

' Cockpit Objects Thrustlever

'C Name	Type	CID	RID	FLAGS	P1	P2	P3
C, CSATHRDISCL,	SWITCH,	3,	2,	,	0,	0,	0
C, CSATHRDISCR,	SWITCH,	3,	3,	,	0,	0,	0

Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die dazugehörige Syntax:

```
sub OnCSATHRDISCL
  DIM ATHRDISCL
  if (Get("CLATHR")=1)
  then
  ExtSysEvent (ATHRDISCL, "L+1340,740;L-1340,740")
  Set ("CLATHR", 0)
  end if
END SUB
```

```
sub OnCSATHRDISCR
  DIM ATHRDISCR
  if (Get("CLATHR")=1)
  then
  ExtSysEvent (ATHRDISCR, "L+1340,740;L-1340,740")
  Set ("CLATHR", 0)
  end if
END SUB
```

Bleibt zum Schluss noch die Trimmung. Leider hat wie oben bereits erwähnt das Zahnrad nicht das gehalten, was es versprochen hat. Macht aber nichts, geht auch nur mit dem rechten Trimmrad.

Der am Rad befindliche Drehencoder ist ebenfalls mit einer FSBUS IO Platine verbunden.

```
' Cockpit Objects Thrustlever
'C Name      Type      CID RID  FLAGS          P1 P2 P3
'-----
C, CRTRIM,      ROTARY,  3, 0,          , 0, 0, 0
```

Projekt Airbus A320 das virtuelle Cockpit

Ein Projektbericht von: Frank Sommer

Die dazugehörige Syntax:

```
sub OnCRTRIM
  DIM AP
  DIM CRTRIM
  if value=1
  then
  SysEvent (CRTRIM, "u")
  end if
  if value=-1
  then
  SysEvent (CRTRIM, "d")
  end if
  if Value=1 or -1
  then
  DIM AP_VALUE
  AP_VALUE=ReadFSDirect (0x07BC, U32,1,1)
  end if
  if AP_VALUE=1
  then
  SysEvent (AP, "b")
  Set ("CLAP1",0)
  Set ("CLAP2",0)
  end if
END SUB
```

Die Bewegung des Encoders ist so definiert, dass der Wert -1 den Key "U" und +1 den Key "D" auslöst. Über das FS Menü werden dann die beiden Buchstaben der Trimmung up und down zugeordnet.

Egal bei welcher Betätigung des Encoders (**gelb**) wird grundsätzlich der Autopilotenstatus über den Offset 0x07BC abgefragt. Ist dieser an, wird ein Systemevent generiert und der Key "b" ausgelöst. Dieser schaltet dann über das FS Menü definiert den AP aus.

Zusätzlich werden im FSBUS die Leuchten des AP1 und AP2 in der FCU deaktiviert.

Den Disconnect des Autotrimmsystems habe ich bislang noch nicht hinbekommen. Das liegt aber an dem Wilco selber. Funktionieren könnte das allerdings über eine Trimmachse. Einen solchen Effekt habe ich wahrgenommen als ich mit FSCommunicator auf einem Client PC experimentiert habe.