

Panel Hardware

Arduino-Hardware Ethernet

Aufbauend auf den Test-Aufbauten mit dem Arduino, die [hier](#) beschrieben sind, habe ich Leiterplatten für Avionics-Panel entwickelt. Das Panel ist nur 50 x 80 mm groß.

Hier ist der [Systemaufbau](#) dargestellt.



Das Panel wird über Ethernet mit dem FS-PC verbunden. Der Arduino-Sketch verarbeitet die Taster-/Drehgeber-Ereignisse, die über UDP (Ethernet) zu einer Schnittstellensoftware auf dem PC übertragen werden, der hieraus FS-Kommandos generiert. Die aktualisierten Variablen werden an das Arduino-Panel übertragen und dort auf dem LC-Display angezeigt. Als Kommunikationssoftware kommen SimConnect-Clients für MSFS (FSX, FS2020) und P3D sowie X-Plane-Plugins für X-Plane zum Einsatz. Weitere Software ist für den Betrieb der Panel nicht notwendig.

Für den Erwerb von Hardware und der Kommunikationssoftware kontaktieren Sie mich bitte per E-Mail. Die Hex-Files und Arduino-Quellen sind für Selbstbauprojekte ebenfalls auf Anfrage erhältlich.

Für die Programmierung des Arduino wird ein USB-Seriell-Adapter wie z.B. [dieser](#) benötigt.

Variante 1

Die Leiterplatte wird bestückt mit Tastern und einem Drehimpulsgeber. Weiterhin werden handelsüblichen Module mit der Leiterplatte verbunden. Durch den relativ einfachen Aufbau und die freie Verfügbarkeit der Leiterplatte ist das Modul auch für der Selbstbau geeignet. Es wird bei der bevorzugten Bestückung dieser Baugruppe keine SMD-Bestückung verwendet.

An die Leiterplatte können zwei unterschiedliche Ethernetmodule angeschlossen werden. Beim ENC28J60 muss der Arduino den IP-Stack in Software verwenden, was deutlich mehr Ressourcen benötigt. Außerdem wird für die gefundenen Module der 3,3V-Regler benötigt. Dem WizNET-Moduls W5500 ist deswegen der Vorzug zu geben:

Es besteht die Möglichkeit einen preiswerten Drehimpulsgeber mit einer Achse zu bestücken, mit dem dann mit einem Druck auf die Drehgeberachse zwischen Vor- und Nachkommastelle umgeschaltet wird. Die andere Möglichkeit ist die Bestückung eines deutlich teureren Drehimpulsgebers mit zwei Wellen. Problematisch ist das Finden von geeigneten Knöpfen dafür. In der Stückliste ist eine Quelle für 3D-Druckdaten eines geeigneten Knopf-Paares aufgeführt.

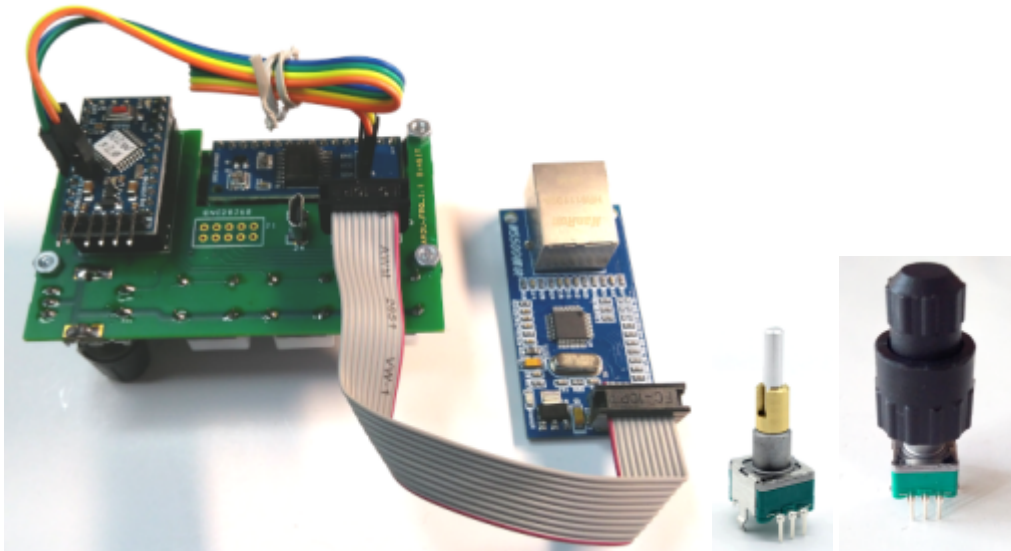
Für beide Varianten wird jeweils ein separater Sketch verwendet. Die Kennung, die beim Verbindungsaufbau gesendet wird, kann dann im Client/Plugin ausgewertet werden.

- Leiterplatte ARDU-FRQ 5cm x 8cm
- Arduino Pro Mini (5V, 16 MHz) in Fassung U1, der Pro Mini hat zwei separate Pins für A4/A5 = I²C-SDA/SCL
- LCD-Modul 2x16 Zeichen, I²C-SDA/SCL an A4/A5, direkt am Arduino-Modul, Spannung über J2
- Ethernet-Modul WizNET W5500 an Steckverbinder J3
- Drehimpulsgeber an A2/A3, **alternativ** Dual-Drehimpulsgeber mit zweitem Ausgang an A0/A1, Push-Button an D4
- drei Taster an D5 .. D7

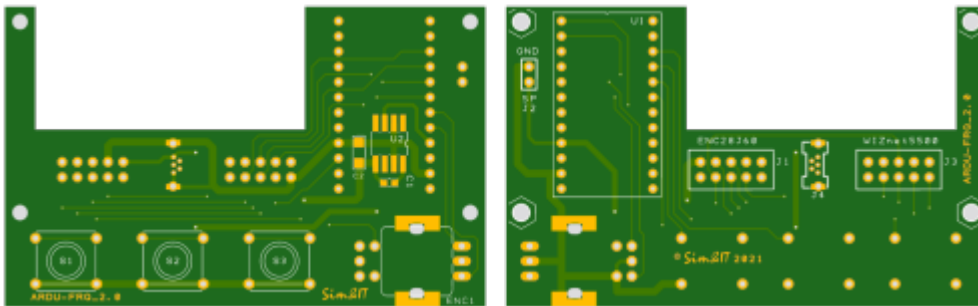
alternative Bestückung, nicht für Dual-Encoder verwendbar, **nicht empfohlen**:

- zusätzlich zu bestücken: 3,3V LDO für Ethernet-Modul mit ENC28J60
- Ethernet-Modul ENC28J60 an Steckverbinder J1

Die Spannungsversorgung (+5V) erfolgt über eine Mini-USB-Buchse J4 oder über die Stiftleiste des Arduino Pro Mini.



Leiterplatte



Die Bestandteile und die Entwicklung des Panels sind [hier](#) beschrieben.

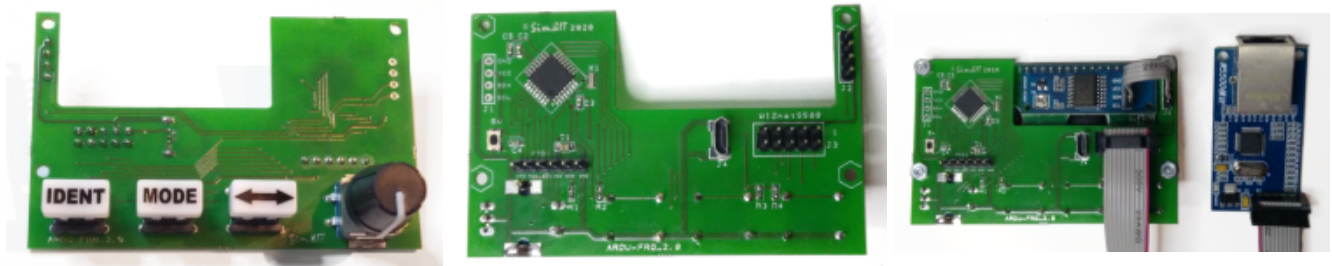
[Schaltplan](#)
[Bestückung](#)
[Shared PCB bei Aisler](#)
[Stückliste](#)

[Frontplatte](#)
[Abdeckung](#)

Variante 2

Bei dieser Variante ist der ATmega328 auf der Leiterplatte integriert. Die Anschlussbelegung entspricht dem Arduino Pro Mini, 5V, 16MHz. Zusätzlich sind die Taster und der Drehimpulsgeber auf der Leiterplatte enthalten.

- Leiterplatte ARDU-FRQ-II 5cm x 8cm
 - mit integriertem und programmierten ATmega328 (Boot Loader und Sketch)
 - Drehimpulsgeber an A2/A3, **alternativ** Dual-Drehimpulsgeber mit zweitem Ausgang an A0/A1, Push-Button an D4
 - drei Taster an D5 .. D7
- LCD-Modul 2x16 Zeichen, I²C-SDA/SCL an A4/A5, Anschluss an J1 bzw. J2
- Ethernet-Modul WizNET W5500 an Steckverbinder J3
- J3 wird auch für die [Erstprogrammierung des Bootloaders](#) auf den ATmega328 verwendet.
- die Spannungsversorgung (+5V) erfolgt über eine Mini-USB-Buchse J4 oder über die FTDI-Stiftleiste
- am Anschluss FTDI wird der USB-Seriell-Wandler für die Programmierung über die Arduino-IDE angeschlossen



[Schaltplan](#)
[Bestückung](#)
[Stückliste](#)

Diese Leiterplatten-Variante wird wegen der SMD-Bestückung und Erstprogrammierung des IC nur bei der Version für die programmierte [Baugruppe](#) eingesetzt.

Alternative Displays

Es stehen auch rote LC-Displays und OLED-Displays in ähnlicher Bauform zur Verfügung. Der I²C-zu-Parallel-Wandler muss jeweils separat beschafft und montiert werden. Bei den roten LC-Displays ist der Blickwinkel jedoch nicht sehr groß, der Kontrast nimmt dabei ab, die OLEDs sind sehr gut lesbar, bei der Ansteuerung aber etwas anders, insbesondere die Initialisierung. Die OLED-Displays sind deutlich preisintensiver.



In eine Acrylplatte wurden M2,5 Stehbolzen eingesetzt, an die die Leiterplatte montiert wird.

[Frontplatte](#)

Auf die Acrylplatte wurde eine Abdeckung aus Fotopapier geklebt.

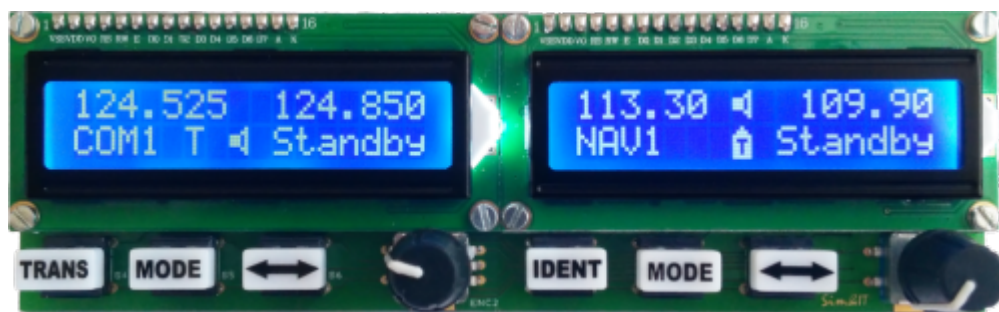
[Abdeckung](#)

NAVCOM-Panel

Aufbauend auf dem Avionics-Panel der [Variante 2](#) wurde dieses um eine zweite Anzeige, drei Taster und einen Encoder erweitert. Damit kann man mit dem NAVCOM-Panel die COM-Frequenzen und die NAV-Funktionen einstellen bzw. anzeigen. Der ATmega328 ist auf der Leiterplatte integriert.

Die Funktionen entsprechen den NAV- und COM-Einzelanzeigen, es ist jedoch nur ein Netzwerkanschluss notwendig. Die Modis sind [unten](#) beschrieben.

Es sind zwei Funktions-Versionen möglich, COM1 und NAV1 oder COM2 und NAV2. Die Funktion kann beim Einschalten/Booten durch Festhalten einer der beiden SWAP-Tasten <-> umprogrammiert werden.



- Leiterplatte ARDU-NAVCOM 5cm x 16cm
 - mit integriertem und programmierten ATmega328 (Boot Loader und Sketch)
 - zwei Drehimpulsgeber
 - sechs Taster
- zwei LCD-Module 2x16 Zeichen, I²C-Ansteuerung, die I²C-Adressen müssen unterschiedlich konfiguriert sein
- Ethernet-Modul WizNET W5500 an Steckverbinder J3



- J3 wird auch für die [Erstprogrammierung des Bootloaders](#) auf den ATmega328 verwendet.
- Die Spannungsversorgung (+5V) erfolgt über eine Mini-USB-Buchse J4 oder über den FTDI-Steckverbinder.
- Am Anschluss FTDI kann ein USB-Seriell-Wandler für die Programmierung über die Arduino-IDE angeschlossen werden.

Leiterplatte

[Schaltplan](#)
[Bestückungsplan](#)
[Stückliste](#)

[Frontplatte](#)
[Abdeckung](#)

NAVCOM Version Dual-Encoder



Für die Version mit Dual-Encoder reichen die Eingänge des Arduino nur noch für den Anschluss von 4 Tasten, das sind dann die beiden Swap-Tasten, der Push-Button am Drehimpulsgeber und beim NAV die Mode-Taste. Die Audio-Funktionen TRANS, RECV, IDENT werden über das [Audio-Panel](#) realisiert. Für diese Version habe ich auch LC-Displays mit rotem Hintergrund verwendet. Diese Displays haben einen eingeschränkten Sichtwinkel. Der

Kontrast nimmt deutlich ab. Aber es sieht realistischer aus. Alles wurde an eine Acryl-Frontplatte montiert, auf welche dann ein bedrucktes Fotopapier geklebt wurde.

Eine Variante mit rotem OLED-Display ist ebenfalls möglich.

[Schaltplan](#)
[Bestückungsplan](#)
[Stückliste](#)

[Frontplatte](#)
[Abdeckung](#)

Multi-Funktions-Panel (MFP)



Das Multi-Funktions-Panel vereint sechs Funktionen, die mit dem oben beschriebenen Avionics-Modul möglich sind. Mit den Tasten an der Seite kann zwischen den Funktionen gewechselt werden. Es stehen die Funktionen COM1, COM2, NAV1, NAV2, ADF und XPDR zur Verfügung.

Ich habe versucht, die Funktionen mit dem Flugsimulator zu synchronisieren, aber diese Kommandos werden oft nicht vom Entwickler der Flugzeug-Panel unterstützt.

Die Modis sind [unten](#) beschrieben.

Das Prinzip ist das gleiche wie bei den oben beschriebenen Avionics-Modulen. Auf der Trägerplatte befindet sich eine Fassung für den Arduino Pro Mini. LCD und Ethernetmodul werden an Stiftheisten angeschlossen.

Nur das Ethernet-Modul WIZnet W5500 wird unterstützt.

- Leiterplatte ARDU-MFP 5cm x 12cm
- Arduino Pro Mini in Fassung U1, der Pro Mini hat zwei separate Pins für A4/A5 = I²C-SDA/SCL
- LCD-Modul 2x16 Zeichen, I²C-SDA/SCL an A4/A5, direkt am Arduino-Modul, Spannung über J2
- Ethernet-Modul WIZNET W5500 an Steckverbinder J3
- Drehimpulsgeber an A2/A3, alternativ Dual-Encoder an A2/A3 und A0/A1
- Push-Button und Taster in 3x4-Matrix an D2..D4/D6..D9
- Die Leiterplatte ist 5cm x 12cm groß.

Die Spannungsversorgung (+5V) erfolgt über eine Mini-USB-Buchse J4 oder über den Arduino-Steckverbinder.

[Schaltplan](#)
[Bestückung](#)

[Shared PCB bei Aisler](#)
[Stückliste](#)

Beispiel mit Frontplatte



Für den abgebildeten Aufbau wurde eine schwarze Acryl-Platte mit Laser bearbeitet. In die Platte wurden sechs 2,5 mm Stehbolzen eingesetzt, an die die Leiterplatten montiert werden.

[Frontplatte](#)

Auf die Acryl-Platte wird eine Abdeckung aus Fotopapier geklebt.

[Abdeckung](#)

Wie bereits bei den Avionics Modulen erfolgte auch hier eine Integration des ATmega328 auf der Leiterplatte. Diese Variante bei der Version für die programmierte [Baugruppe](#) eingesetzt

[Schaltplan](#)
[Bestückung](#)
[Stückliste](#)

Auch hier ist eine OLED-Version mit Dual-Encoder möglich.



ESP8266-Hardware WIFI (Test-Version)

Dies ist eine experimentelle Entwicklung

Diese Variante hat das gleiche Design, wie die oben beschriebene erste Variante, wird aber nicht von einem Arduino-Pro-Mini angesteuert, sondern als Prozessor dient der ESP8266. Dieser enthält ein WIFI-Modul mit komplettem IP-Stack. Die Programmierung erfolgt weiterhin über die Arduino-Oberfläche, die ESP-Hardware kann ergänzt werden. Grundlegendes dazu ist [hier](#) beschrieben.

Das WeMos-D1-Mini-Modul enthält einen USB-Seriell-Wandler und kann somit direkt an den PC angeschlossen und aus der Oberfläche programmiert werden.

Das Programm konnte bei fest einprogrammierter SSID/Kennwort fast vollständig übernommen werden nur durch den Austausch von Bibliotheken und Anpassung der Ports.

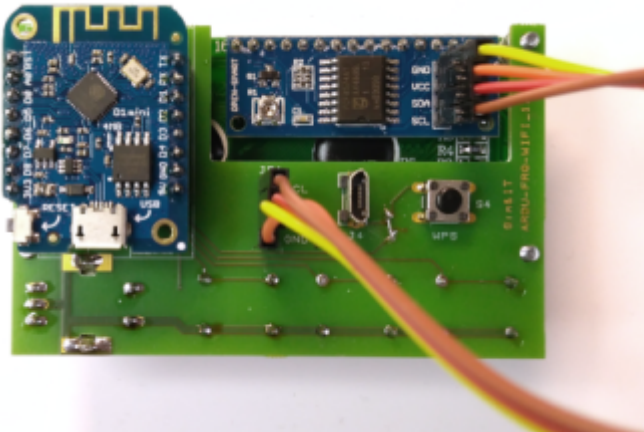
Größere Änderungen grachte die Unterstützung von WPS mit sich. Mit dieser Funktion handeln der Router (muss

WPS unterstützen) und das Gerät innerhalb eines kurzen Zeitraumes nach dem Drücken des WPS-Knopfes am Router und dann des WPS-Knopfes am Gerät die Verbindungsdaten aus.

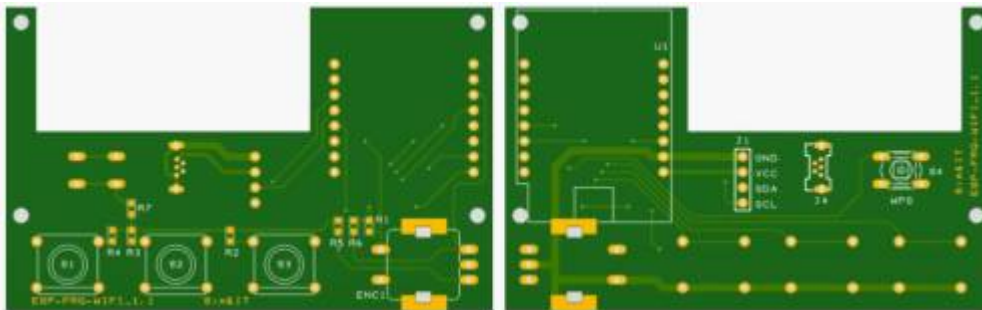
Die Leiterplatte enthält einen Sockel, um ein WeMos-D1-Mini-Modul aufzunehmen

- Leiterplatte Ardu-FRQ-WIFI 5cm x 8cm
- [WeMos-D1-Mini](#)
- LCD-Modul 2x16 Zeichen, I²C-SDA/SCL an D1/D2, Versorgung und I²C über 4pol Steckverbinder
- Drehimpulsgeber an D0/D3, Push-Button an D4
- drei Taster an D5 .. D7
- WPS-Taste an D8

Die Spannungsversorgung (+5V) erfolgt über eine Mini-USB-Buchse J4 oder über die USB-Buchse des WeMos-Moduls.



Leiterplatte



[Schaltplan](#)

[Bestückung Top](#)

[Bestückung Bottom](#)

Software

Arduino Sketch

Das Avionics-Panel mit dem Arduino-Sketch stellt ein Frontend zum SimConnect-Client bzw. X-Plane-Plugin dar. Das Panel verbindet sich mit der Kommunikations-Software auf dem PC, Ereignisse werden an diese übertragen und die gelieferten Variablen-Werte werden angezeigt.

Das Ethernet-Modul wird mit dem Netzwerk verbunden und muss über DHCP eine Netzwerk-Adresse zugewiesen bekommen. Die WIFI-Variante muss einmalig über WPS mit dem Router die Verbindungsdaten austauschen und

erhält dann ebenfalls vom DHCP-Server eine Netzwerk-Adresse. Für die Datenverbindung wird das UDP-Protokoll genutzt. Hier ist zu beachten, dass immer eine eindeutige MAC-Adresse und ein eindeutiger Port zugewiesen wird. Beide sind im Sketch definiert.

Im COM-Panel werden Active- und Standby-Frequenzen unterstützt, mit den Tasten kann der Transmit-Mode und der Receive-Both-Mode aktiviert werden.

Das NAV-Panel ist komplexer programmiert. Neben den Active- und Standby-Frequenzen ist auch eine einfache Skalendarstellung des CDI möglich. Diese Funktion wird durch die Möglichkeit von benutzerdefinierten Zeichen des LCD-Panels ermöglicht. Die Anzeigefunktionen werden vom SimConnect-Client bzw. dem X-Plane-Plugin realisiert.

Beim Starten bzw. beim Reset des Controllers kann über das Drücken der drei Tasten während des Startens/Rücksetzens die Modulfunktion geändert werden. Diese Änderung wird im EEPROM-Speicher des Controllers abgelegt, so dass diese Einstellung bis zu einer erneuten Änderung erhalten bleibt.

Tastenvarianten zum Ändern der Funktion beim Avionics-Panel:

rechte Taste	COM1
mittlere Taste	NAV1
rechte und mittlere Taste	COM2
linke Taste	NAV2
linke und rechte Taste	ADF
linke und mittlere Taste	XPDR

Tastenvarianten zum Ändern der Funktion beim NAVCOM-Panel:

<-> beim NAV-DISPLAY	NAVCOM1
<-> beim COM-Display	NAVCOM2

Die gestartete Software zeigt die Funktion und kurz die zugewiesene Netzwerk-Adresse an und wartet dann auf den Verbindungsaufbau durch die Kommunikations-Software. Durch den Verbindungsaufbau weiß die Arduino-Software, wohin sie die Daten senden soll.

- Anzeige der zugewiesenen **IP-Adresse** für ca. 3s
- Anschließend Warten auf Verbindungsaufbau durch ein Kommunikationsprogramm eines Flugsimulators (FSX/P3D-SimConnect-Client, X-Plane-Plugin)
- Anzeige des Connect-Status



Nach dem Verbindungsaufbau können Tasten- und Encoder-Ereignisse gesendet und Daten angezeigt werden, die von der Kommunikationssoftware in definierter Format übertragen werden.

Kommunikations-Software

Die Kommunikations-Software stellt die Verbindung zwischen dem Flugsimulator und dem Avionics-Panel her. Wenn ein PC mehr als eine Netzwerkkarte besitzt, dann muss über die Konfigurationsdatei des Clients/Plugins dieser definiert werden. Beim Start werden dann die gefundenen Netzwerkadapter mit ihrer ID aufgelistet.

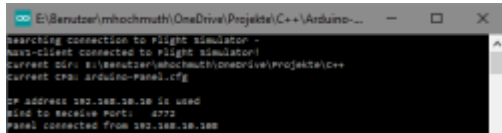
Über den Netzwerkadapter wird mit einer Anfrage auf der Broadcast-Adresse des Netzwerksegmentes und dem der jeweiligen Panel-Funktion zugeordneten UDP-Port (dieser muss mit dem Sketch übereinstimmen) der Avionics-Controller gesucht. Der Avionics-Controller empfängt diese Anfrage und kennt dadurch die IP-Adresse der sendenden Kommunikations-Software. An diese IP-Adresse wird dann die Bestätigung gesendet, wodurch der Kommunikations-Software die Controller-IP-Adresse übermittelt wird. Damit ist die Verbindung aufgebaut. Der

Controller kennt jetzt die IP-Adresse des Flugsimulators und der Flugsimulator die IP-Adresse des Controllers.

Jetzt können Ereignisse des Avionics-Panels über UDP zum Client/Plugin gesendet werden, der diese dann in Flugsimulator-Ereignisse umsetzt bzw. Anzeige-Varianten einstellt. Die aktualisierten Variablen werden in Anzeige-Zeichenketten umgewandelt und in definierter Form zurück an das Avionics-Panel gesendet und dort auf dem Display angezeigt.

Zyklisch wird abgefragt, ob das Panel noch im System ist. Wenn innerhalb einer bestimmten Zeit auf diese Anfrage keine Antwort empfangen wird, dann geht der Client/das Plugin wieder in die Verbindungsaufbau-Phase und sendet Such-Anfragen auf der Broadcast-Adresse.

SimConnect-Client



Der SimConnect-Client (FSX, P3D, FS2020) ist ein eigenständiges Programm und kann direkt aus einem Ordner, z.B. dem Installationsordner, aufgerufen werden. Die Konfigurationsdatei kann in diesem Ordner angelegt sein oder in einem Unterordner des AppData-Bereiches.

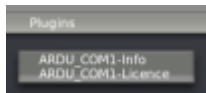
. \AppData\Roaming\Sim&IT\SimConnect\Avionics-Panel.cfg

→ Weitere Informationen zu den SimConnect-Clients und deren Konfiguration sind [hier](#) zu finden.

[Client-Setup FSX](#)

[Client-Setup FS2020](#)

X-Plane-Plugin



Das X-Plane-Plugin entspricht einer DLL. Diese sind im Unterordner . \resources\plugins abgelegt. Das gefundene Plugin kann im gestarteten Flugsimulator über das Menü angezeigt werden. Im Info-Bereich wird entweder die Verbindung zum Avionics-Panel oder die Auswahl der Netzwerk-Adapter angezeigt, wenn mehr als ein Adapter gefunden wurde. Dieser muss in der Konfigurationsdatei definiert werden.

. \AppData\Roaming\Sim&IT\X-Plane-Plugins\Avionics-Panel.cfg

→ Weitere Informationen zu den X-Plane-Plugins und deren Konfiguration sind [hier](#) zu finden.

[Plugin-Setup](#)

Funktionen

Die Funktion der Panels kann durch das Drücken von Tasten beim Einschalten der Betriebsspannung wie im [Arduino-Sketch](#) beschrieben, geändert werden:

Das **COM-Panel** zeigt die Active-/Standby-Frequenzen an. Folgende Funktionen sind möglich:

- **ACTIVE/STANDBY** - Mode ⇒ Anzeige der Active/Standby Frequenzen
 - Drehgeber ändert die Standby-Frequenz, Push Button wechselt zwischen Vor- und Nachkomma
 - Dual-Drehgeber - außen Vorkomma, innen Nachkomma
 - TRANS aktiviert das Panel als Transmitter, wird durch T angezeigt
 - <-> wechselt die Standby- und aktive Frequenz
 - MODE schaltet den Both-Receive-Mode ein/aus, wird durch Lautsprechersymbol angezeigt

Die Funktionen des **NAV-Panels** sind an die des KX155A/KX165A angelehnt.

Folgende Zustände sind dargestellt:



- **ACTIVE/STANDBY** - Mode ⇒ Anzeige der Active/Standby Frequenzen
 - Drehgeber ändert die Standby-Frequenz, Push Button wechselt zwischen Vor- und Nachkomma
 - Dual-Drehgeber - außen Vorkomma, innen Nachkomma
 - IDENT aktiviert den jeweiligen NAV-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die Standby- und aktive Frequenz
 - MODE schaltet in den ACTIVE/CDI-Mode
- **ACTIVE/CDI** - Mode ⇒ Anzeige der aktiven Frequenz, des Course Deviation Indicators (CDI) und des OBS-Kurses
 - Drehgeber ändert die aktive Frequenz, Push Button wechselt zwischen Vor- und Nachkomma
 - Dual-Drehgeber - außen Vorkomma, innen Nachkomma
 - IDENT aktiviert den jeweiligen NAV-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die versteckte Standby- und aktive Frequenz
 - Skale hat +/- 16 Stellungen (15 Striche entsprechen den 5 Dots des Rund-Instrumentes)
 - bei einem VOR wird TO bzw. FROM in der Mitte der Skale angezeigt
 - bei TO kann durch ca. 2s Druck auf die Drehgeberachse der aktuelle Bearing-TO-Kurs übernommen werden
 - Durch Doppeldruck auf die Drehgeberachse kann zur OBS-Einstellung gewechselt werden, dies wird durch ein > signalisiert
 - Im OBS-Mode schaltet ein weiterer Druck auf die Drehgeberachse den Fast-Mode (+/-5°) ein bzw. aus
 - Nach ca. 2s ohne Aktivität wird auf die Frequenzeinstellung zurückgeschaltet
 - bei einem Localizer wird eine Raute in der Mitte der Skale angezeigt und LOC anstelle des OBS-Kurses
 - Durch Doppeldruck auf die Drehgeberachse kann zur OBS-Einstellung gewechselt werden, dies hat keine Auswirkung auf das CDI, ist rein informativ
 - Ist kein Signal vorhanden, ist in der Mitte der Skale ein Strich, der Zeiger ist ebenfalls in der Mitte
 - MODE schaltet in den ACTIVE/BEARING-Mode
- **ACTIVE/BEARING TO** - Mode ⇒ Anzeige der aktiven Frequenz und des Kurses zur Station (Bearing TO)
 - Ungültiges Signal wird durch drei Minus-Zeichen angezeigt
 - Drehgeber ändert die aktive Frequenz, Push Button wechselt zwischen Vor- und Nachkomma
 - Dual-Drehgeber - außen Vorkomma, innen Nachkomma
 - IDENT aktiviert den jeweiligen NAV-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die versteckte Standby- und aktive Frequenz
 - MODE schaltet in den ACTIVE/RADIAL-Mode
- **ACTIVE/RADIAL FROM** - Mode ⇒ Anzeige der aktiven Frequenz und des Kurses weg von der Station (Radial FROM)
 - Ungültiges Signal wird durch drei Minus-Zeichen angezeigt
 - Drehgeber ändert die aktive Frequenz, Push Button wechselt zwischen Vor- und Nachkomma
 - IDENT aktiviert den jeweiligen NAV-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die versteckte Standby- und aktive Frequenz
 - MODE schaltet in den ACTIVE/DME-Mode
- **ACTIVE/DME** - Mode ⇒ Anzeige der aktiven Frequenz und der DME-Informationen
 - Ungültige Informationen werden durch Minus-Zeichen angezeigt
 - Drehgeber ohne Funktion, Push Button selektiert/deselektiert das DME, angezeigt durch sel
 - IDENT aktiviert den jeweiligen DME-Sound (nur im selektierten Mode), angezeigt durch Lautsprechersymbol
 - ↔ wechselt die versteckte Standby- und aktive Frequenz

- MODE schaltet in den ACTIVE/STANDBY-Mode

Das **ADF-Panel** kann folgende Zustände darstellen:

- **ACTIVE/STANDBY** - Mode ⇒ Anzeige der Active/Standby Frequenzen
 - Drehgeber ändert die Standby-Frequenz, Push Button wechselt zwischen den Stellen
 - IDENT aktiviert den ADF1-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die Standby- und aktive Frequenz
 - MODE schaltet in den ACTIVE/RAD-Mode
- **ACTIVE/RAD** - Mode ⇒ Anzeige der aktiven Frequenz, des ADF Zeigers und des ADF-HDG
 - Drehgeber ändert die aktive Frequenz, Push Button wechselt zwischen den Stellen
 - IDENT aktiviert den ADF1-Sound, wird durch Lautsprechersymbol angezeigt
 - ↔ wechselt die versteckte Standby- und aktive Frequenz
 - Ist kein Signal vorhanden, ist in der Mitte der Skale ein Strich, der Zeiger ist ebenfalls in der Mitte
 - Ein aktives Signal wird durch eine Raute angezeigt
 - die Skale stellt +/- 50° dar, größere Abweichungen werden durch < und > am Skalenende angezeigt
 - Durch Doppeldruck auf die Drehgeberachse kann zur ADF-HDG-Einstellung gewechselt werden, dies wird durch ein > signalisiert
 - Im HDG-Mode schaltet ein weiterer Druck auf die Drehgeberachse den Fast-Mode (+/- 5°) ein bzw. aus
 - Nach ca. 2s ohne Aktivität wird auf die Frequenzeinstellung zurückgeschaltet
 - MODE schaltet in den ACTIVE/ET-Mode
- **ACTIVE/ET** - Mode ⇒ Anzeige der aktiven Frequenz und eines Elapsed Timers
 - Drehgeber ändert die aktive Frequenz, Push Button wechselt zwischen den Stellen
 - Dual-Drehgeber - außen Vorkomma, innen Nachkomma
 - IDENT startet/stoppt/setzt den Timer
 - ↔ wechselt die versteckte Standby- und aktive Frequenz
 - MODE schaltet in den ACTIVE/STANDBY-Mode

Das **Transponder-Panel XPDR** zeigt den Flight-Level und den Transponder-Code an.

- Drehgeber ändert den Transponder-Code, Push Button wechselt zwischen den Stellen, nach ca. 5s wird zurück auf linke Stelle gesetzt
- durch ca. 2s Druck auf die Drehgeberachse wird auf den in der Konfigurationsdatei eingestellten Wert (default 1200) gesetzt
- **nur bei X-Plane und MSFS2020:**
 - IDENT aktiviert den Transponder-IDENT
 - ↔ wechselt zwischen ON und ALT, Drehgeber ändert kurzzeitig Mode
 - MODE wechselt zwischen OFF und Standby, Drehgeber ändert kurzzeitig Mode
 - beim Dual-Drehgeber ändert der äußere Drehgeber den Mode

Links

Installations-Anleitungen:

- [SimConnect-Client](#)
- [X-Plane-Plugin](#)

[Entwicklung des Flightsim-Panels](#)

[Audio-Panel](#)

[Encoder-Controller](#)

[Autopilot](#)

[Cockpit-Panel-PC](#)

From:

<http://simandit.de/simwiki/> - Wiki

Permanent link:

<http://simandit.de/simwiki/doku.php?id=hardware:anleitungen:arduino-panel:hardware>

Last update: **2025/09/26 21:01**

