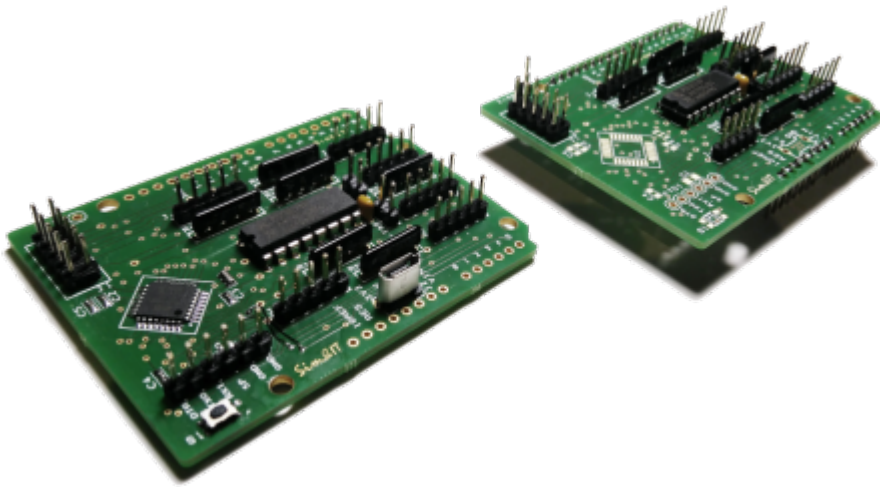


Controller Hardware

Arduino Hardware



Aufbauend auf den ersten Tests, die [hier](#) beschrieben sind, habe ich eine Leiterplatte erstellt.

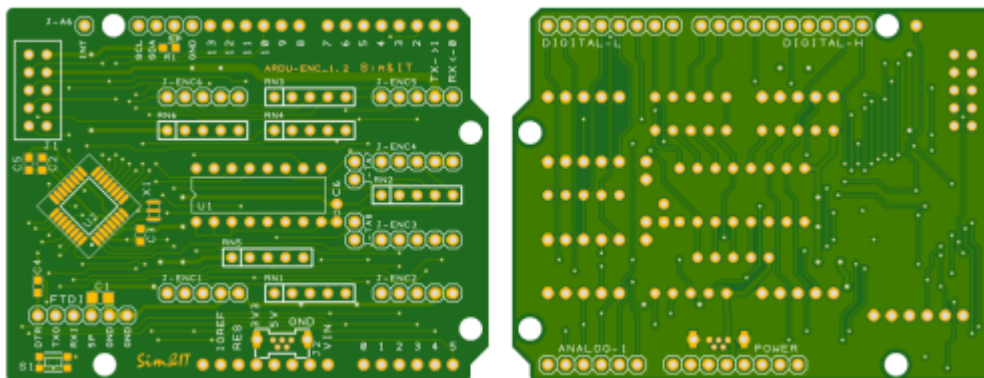
Die Leiterplatte kann in zwei Bestückungsvarianten aufgebaut werden. Die einfacher aufzubauende Variante „Shield“ ist als Zusatz für einen Arduino Uno R3 SMD gedacht. Voraussetzung ist, dass der Anschluss A6, der nur in der SMD-Variante des ATmega328 verfügbar ist, auch über die Leiterplatte zugänglich ist. Dies ist nicht bei allen Ausführungen der Fall. Alle anderen Anschlüsse werden für die Drehimpulsgeber und die Peripherie benötigt. Die je zwei Impulsausgänge der Drehimpulsgeber werden an Ports des Arduino geführt. Die Drucktaster werden an einen I²C-Portexpander PCF8574 geführt. Der Interruptausgang initiiert über A6 des Arduino die Portabfrage beim Zustandswechsel.

An diese Leiterplatte werden angeschlossen:

- ein [Ethernet-Modul](#) und die
- sechs [Drehimpulsgeber](#)

Leiterplatte

Die Leiterplatte ist im Arduino-Uno-Shield-Format gestaltet. Diese enthält bei voller Bestückung einen Arduino Pro Mini (ATmega328) und kann dann separat verwendet werden. Die abgerüstete Variante (Shield) wird auf einen Arduino Uno R3 SMD gesteckt. Angeschlossen werden dann das Ethernet-Modul und die 6 Drehimpulsgeber. Es können noch zwei einzelne Tasten angeschlossen werden.



[Schaltplan \(Vollausbau\)](#)
[Stückliste](#)

Leiterplatte bei Aisler

Variante Encoder Shield

Die Encoder-Shield-Variante ist nur mit THT-Bauteilen bestückt, so dass sich der Aufbau recht einfach gestaltet.

Die Stiftleisten auf der Unterseite dienen der Verbindung zum Arduino Uno R3, über die auch die Spannungsversorgung erfolgt. J-A6 muss separat mit A6 des Arduino verbunden werden. An die Steckverbinder J-ENC1 bis J-ENC6 werden die Drehimpulsgeber angeschlossen und an J1 das Ethernetmodul.

- Arduino UNO R3 (mit SMD-Variante des ATmega328 und herausgeführtem Anschluss A6), siehe [hier](#)
- Leiterplatte ARDU-ENC im Arduino-Uno-Format mit Shield-Bestückung
- Ethernet-Modul WizNET W5500 an Steckverbinder J3, siehe [hier](#)
- Drehimpulsgeber mit Push-Button (gleiche Anzahl Pulse wie Rastungen), siehe [hier](#)

Bestückungsplan Shield

Variante Encoder Controller

Die Encoder-Controller-Variante enthält neben den THT-Bauteilen der Encoder-Shield-Variante auch die Bauteile eines Arduino Pro Mini. Die Stiftleisten auf der Unterseite sowie J-A6 werden nicht benötigt. Die Spannungsversorgung kann über eine Micro-USB-Buchse erfolgen, aber auch über den Programmierstecker FTDI ist dies möglich. An die Steckverbinder J-ENC1 bis J-ENC6 werden die Drehimpulsgeber angeschlossen und an J1 das Ethernetmodul.

- Leiterplatte ARDU-ENC im Arduino-Uno-Format mit Controller-Bestückung
- Ethernet-Modul WizNET W5500 an Steckverbinder J3, siehe [hier](#)
- Drehimpulsgeber mit Push-Button (gleiche Anzahl Pulse wie Rastungen), siehe [hier](#)

Bestückungsplan Controller

Für den Aufbau dieser Variante ist es notwendig, im ATmega328 einen Bootloader zu programmieren, damit die IDE die Sketche hochladen und der ATmega diese dann starten kann. Dies kann mit einem Arduino Uno erfolgen. Grundlegende Hinweise zur Erstprogrammierung sind hier zu finden:

- [ArduinoISP](#)
- [ArduinoToBreadboard](#)

Die konkrete Vorgehensweise habe ich [hier](#) beschrieben.

Für die Sketch-Programmierung ist dann ein USB-Seriell-Wandler am Anschluss FTDI notwendig, wie z.B. [hier](#) gezeigt.

Steckerbelegung

J2 - Spannungsversorgung

Die Baugruppe kann über USB mit Spannung versorgt werden.

J-ENC1 bis J-ENC6 (Drehimpulsgeber)

Für die 6 Drehimpulsgeber des Controllers sind fünfpolige Steckverbinder vorgesehen.

1 2 3 4 5

```

o  o  o  o  o
|  |  |  |  |
|  |  |  Masse für Taster und Encoder
|  |  in Uhrzeigerrichtung folgendes Signal
|  Taster Encoderachse
in Uhrzeigerrichtung führendes Signal

```



Encoder-Anschlussmöglichkeiten

Als Encoder können Typen mit 11mm Breite zu Einsatz kommen.

[Hier](#) sind einige Beispiele aufgeführt. Es müssen Encoder mit der gleichen Anzahl Pulse wie Rastungen sein, z.B. 20 Pulse und 20 Rastungen.

Es besteht die Möglichkeit des Anschlusses über eine Leiterplatte mit/ohne Steckverbinder oder bei sehr engen Bauräumen direkt am Drehimpulsgeber.

Die Anschlussbelegung auf dem Kabel resultiert aus der ursprünglichen Verwendung eines 16mm-Drehimpulsgebers und ist an die eine Reihe der Belegung des 10poligen Steckverbinders bei ITRA-Drehimpulsgebern angelehnt.

Beispiel 11mm-Encoder [Alpha RE111F-41B3N-20F-20P](#) (Blick von oben)

```

A  o  ----- (3) in Uhrzeigerrichtung folgendes Signal
PB          o  --- (4) Masse Taster
C  o  ----- (5) Masse Encoder
PB          o  --- (2) Taster Encoderachse
B  o  ----- (1) in Uhrzeigerrichtung führendes Signal

```

Wenn die Drehrichtung verkehrt herum ist, dann hilft es, die Anschlüsse A und B zu tauschen.

J1 - Ethernet



Das Ethernet-Modul wird über SPI angeschlossen, die zugeordneten Arduino-Ports sind D10 ... D13. Weitere Informationen zum Modul gibt es [hier](#)

```

      10   9
      o   o 5P
MISO  o   o GND
MOSI  o   o RES
CS     o   o
SCK   o   o
      2   1

```

Dieser Anschluss wird auch für die [Erstprogrammierung des Bootloaders](#) auf den ATmega328 verwendet.

Anschluss FTDI



Über diesen Steckverbinder kann der Controller mit einem USB-Seriell-Adapter verbunden und dann programmiert werden. Die Belegung entspricht der Belegung eines Arduino Pro Mini. Rx, TX sind für die Datenübertragung, das DTR-Signal steuert das Reset des Pro Mini. Es gibt Ausführungen u.a. mit einem FTDI-Chip oder auch mit einem CH340G-Chip.

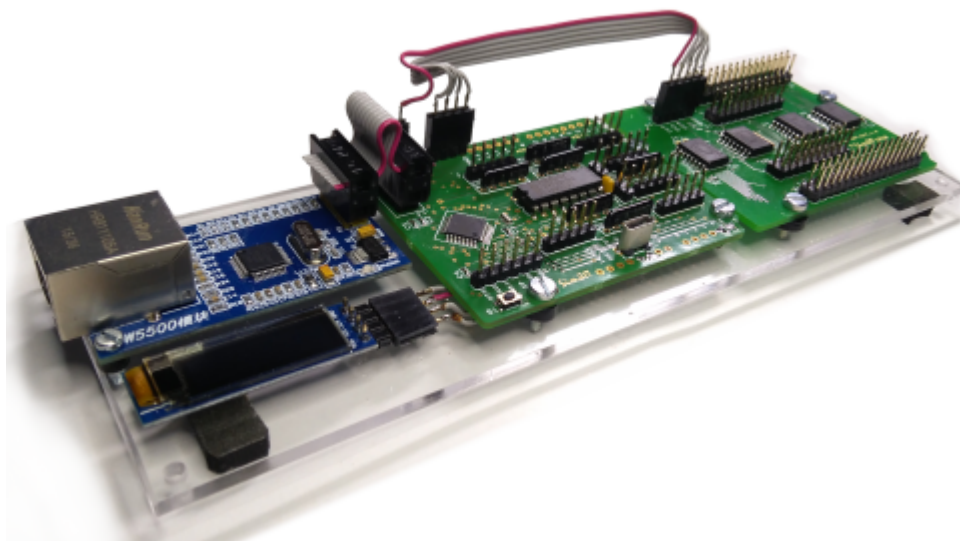
1	2	3	4	5	6
0	0	0	0	0	0
				Vss (GND)	
			Vcc (5P)		
		Rx (Eingang des ATmega)			
	Tx (Ausgang des ATmega)				
DTR (für Reset)					

Erweiterungen

Port-Erweiterung für Schalter

Die Port-Erweiterung ist nur für die Controller-Version geeignet.

Für die Abfrage von Schalter-Ereignissen wurde eine Leiterplatte erstellt, die die gleiche Baugröße wie der Arduino-Uno hat, aber es sind nicht die Steckverbinder eines Arduino-Shields vorhanden. Als Verbindung zur Encoder-Controller-Baugruppe ist ein 5-Poliger Steckverbinder vorgesehen.



Die Abbildung zeigt den kompletten Aufbau des Controllers mit der Port-Erweiterung und dem Ethernet-Modul. Mit dieser Erweiterung ist es möglich, Schalter des Panels eines Kleinflugzeuges nachzubilden. Dazu zählen u.a. Starter, Lichtschalter, Fuehpump, Tank, Flaps, Gear.

Weitere Informationen hierzu sind im Abschnitt [Erweiterungen](#) zu finden.

Anwendungsbeispiel

[Bau eines Cockpit-Panel-PCs](#)



Software

Arduino Sketch

Der Encoder Controller soll sich mit einer PC-Software verbinden und Ereignisse an diese übertragen.

Das Ethernet-Modul wird mit dem Netzwerk verbunden und muss über DHCP eine Netzwerk-Adresse zugewiesen bekommen. 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.

Nach dem Verbindungsaufbau können Tasten- und Encoder-Ereignisse an die PC-Software gesendet werden.

Kommunikations-Software

Die Kommunikations-Software stellt die Verbindung zwischen dem Flugsimulator und dem Encoder Controller 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.

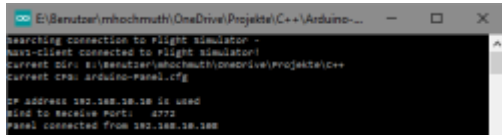
Über den Netzwerkadapter wird mit einer Anfrage auf der Broadcast-Adresse des Netzwerksegmentes und dem der jeweiligen Panel-Funktion zugeordneten UDP-Port (dieser muss mit dem Sketch übereinstimmen) der Encoder-Controller gesucht. Der Encoder-Controller empfängt diese Anfrage und kennt dadurch die IP-Adresse 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 der Drehimpulsgeber über UDP zum Client/Plugin gesendet werden, der diese dann in

Flugsimulator-Ereignisse umsetzt.

Zyklisch wird abgefragt, ob der Controller 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 erneut Such-Anfragen auf der Broadcast-Adresse.

SimConnect-Client



Der SimConnect-Client (FSX, FS2020, P3D) 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\Encoder-Control.cfg

→ Weitere Informationen zum SimConnect-Client und dessen Konfiguration sind [hier](#) zu finden.

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 Encoder Controller 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\Encoder-Control.cfg

→ Weitere Informationen zum X-Plane-Plugin und dessen Konfiguration sind [hier](#) zu finden.

Funktionen

Der Encoder Controller realisiert folgende Funktionen, wenn nicht andere Einstellungen über die Konfigurationsvariablen eingestellt werden.

Encoder1

Der Encoder stellt den OBS1. Ein kurzer Druck auf die Drehgeberachse schaltet in einen Schnellverstell-Modus (+/- 5°) und zurück. Nach einigen Sekunden ohne Eingabe wird auf den Normal-Modus zurückgestellt.

Encoder2

Der Encoder stellt den OBS2. Ein kurzer Druck auf die Drehgeberachse schaltet in einen Schnellverstell-Modus (+/- 5°) und zurück. Nach einigen Sekunden ohne Eingabe wird auf den Normal-Modus zurückgestellt.

Encoder3

Der Encoder stellt den ADF-Kurs. Ein kurzer Druck auf die Drehgeberachse schaltet in einen Schnellverstell-Modus (+/- 5°) und zurück. Nach einigen Sekunden ohne Eingabe wird auf den Normal-Modus zurückgestellt.

Encoder4

Der Encoder stellt den Heading-Bug. Ein kurzer Druck auf die Drehgeberachse schaltet in einen Schnellverstell-Modus (+/- 5°) und zurück. Nach einigen Sekunden ohne Eingabe wird auf den Normal-Modus zurückgestellt.

Encoder5

Der Encoder stellt den barometrischen Druck ein (QNH). Ein kurzer Druck auf die Drehgeberachse stellt auf 29,92inHg bzw. 1013,2hPa (STD).

Encoder6

Ein kurzer Druck auf die Drehgeberachse aktiviert den Encoder. Der Encoder stellt den Kurskreisel. Ein langer Druck synchronisiert den Kurskreisel mit dem magnetischen Kompass.

Für andere Anwendungen sind die Funktionen über Konfigurationsvariablen veränderbar. Dies ist im Abschnitt zu der [Kommunikationssoftware](#) beschrieben.

Links

Installations-Anleitungen:

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

[Entwicklung des Encoder Shields](#)

[Bootloader programmieren](#)

[Bau eines Cockpit-Panel-PCs](#)

From:

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

Permanent link:

<https://simandit.de/simwiki/doku.php?id=hardware:anleitungen:ardu-enc:hardware>

Last update: **2025/09/23 20:09**

