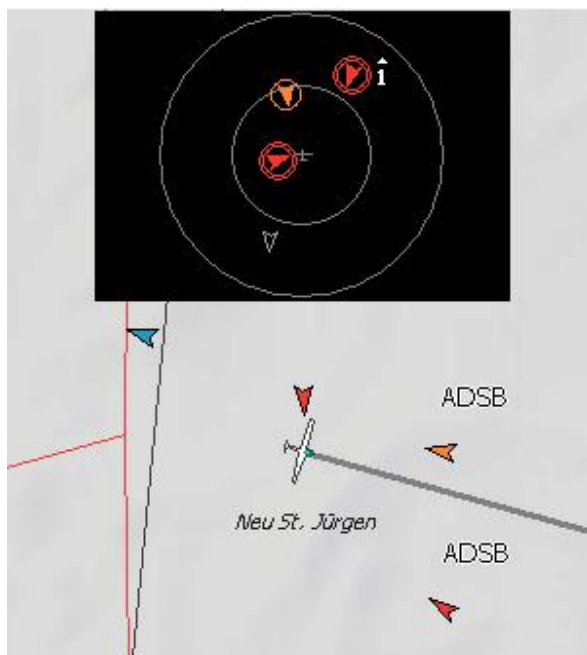


## ADS-B per preiswertem DVB-T -Stick in die XCSoar-Moving Map bringen, oder ...

### 1.)

Das nebenstehende Bild zeigt die Signalisierung von vier Verkehrszielen in XCSoar.  
Das harmlose Ziel links (blau) fliegt von uns weg, das direkt von oben kommende (rot) ist eine sehr ernste Gefahr. Beides sind Flarm-Verkehrsziele.



### 3.)

Es handelt sich um eine provozierte Vektor-Simulation zum Test der Visualisierung. Es möge bitte niemand in eine derartige Situation kommen!

*Das hier vorgestellte System ist bereits in der realen Welt seit 2016 in Betrieb, die folgende ADSB-Abhandlung ist keine Simulation!*

### 2.)

Die beiden von rechts kommenden Ziele (gelb + rot) sind sich sehr schnell annähernde Flugzeuge, mit wachsender und ernster Kollisionsgefahr. Beide Ziele sind in den bestehenden Flarm-Datenstrom eingemischte ADSB-Verkehrsziele. Die „ADSB“-Kennung kann anstelle der Original-ID eingeblendet werden.

## ... pimp your Flarm mit ADS-B

Das Programm "**adsbVektor**" (in Verbindung mit **ADSBSharp.exe = ADSB#**) kann mit sehr geringem zusätzlichen Hardware-Aufwand ADS-B -Verkehrsziele in jede bestehende Flarm-Installation bringen, die mit einem Windows-PC und XCSoar zusammen arbeitet. Die genannten Programme sind, genau wie eventuell weitere notwendige Tools, alle kostenlos nutzbar.

### Vorab-Hinweise:

- J Dieses Dokument ist die FAQ Nr. 5.1.12 (F: Kann man einen DVB-T -Stick zum Empfang von ADS-B- Transpondern an den IPX anschließen?) aus dem Manual: <http://www.schrandt-electronic.de/Dokumente/IPX-Manual.pdf>
- J Im Folgenden wird davon ausgegangen, dass als Bordrechner oder Bodenstation ein Windows-PC arbeitet.
- J Der Begriff "DVB-T" (Digital Video Broadcasting - Terrestrial) wird im folgenden "DVBT" geschrieben.
- J Der Begriff "ADS-B" (Automatic Dependent Surveillance - Broadcast) wird im folgenden "ADSB" geschrieben.

*Im ersten Teil dieses Dokumentes wird der (ein möglicher) Weg zum bordeigenen ADSB-Empfang mittels Low-Cost-USB-DVBT-Stick (SDR-Prinzip) beschrieben.*

*Im zweiten Teil wird die praktische Umsetzung in einer typischen Flarm-Avionik-Umgebung erklärt.*

### Erster Teil, die Funktionsweise:

Es folgt die Beschreibung zweier, seit Anfang 2016 mit ADSB „aufgerüsteten“ Bordrechner, sowie die schon langjährig zuvor in Betrieb befindliche Grund-Hardware und deren Installation (3x im Segelflugzeug und einem Motorsegler).

- J Flarm-Grundgeräte (Classic-Flarm bzw. Flarm-Core),
- J externes (V3+)LED-Display (die Entfernungsanzeige entlastet enorm, ist sehr zu empfehlen!),

- J Windows-PC als Bordrechner (der IPX mit Win-10 wird in dieser Beschreibung genannt) mit XCSoar als Navigations-Software,
- J die Bereitstellung von aerodynamischen Daten erfolgt über eine externe Datenquelle (Cambridge 302, LX-1600, XCSensorbox etc.).

Diese Installationen haben sich mehrfach langjährig bewährt und ist mit sehr überschaubarem Aufwand um ADS-B erweitert worden. Die dazu zusätzlich notwendigen Komponenten:

- J Ein DVBT-Stick mit R820T-Chipsatz,
  - J das Programm **adsbVektor** und etwas Freeware,
  - J noch etwas Software-Installations-Geschick,
  - J und nur bei Verwendung mit externem LED-Display, auf dem zusätzlich noch die ADSB-Ziele dargestellt werden sollen (V2-4), ist noch ein externer Multiplexer/Gewichter+COM (im Folgenden „**adsbMuxer**“ genannt) nötig, siehe „Schematisches Schaltbild-1“.
- Die Funktionsweise und Arbeits-Logik des **adsbMuxer**'s ist überschaubar, wer will, kann (als versierter Elektroniker) sich diesen auch mit eigenen Mitteln selber aufbauen.

In entsprechender Abwandlung wird manch vergleichbares Flarm-System ebenso mit ADSB-Daten aufzurüsten sein. Man erhält eine Signalisierung von drohenden Kollisionen, welche durchaus mit einem originalen Power-Flarm oder dem Core-Pure mit integrierter ADSB-Option vergleichbar ist.

Die Nachteile der DVBT-Sticklösung gleich vorweg:

- J Es lassen sich vielleicht nicht auf jedem PC die Programme automatisch nacheinander starten, so dass das alternative ADSB-System eventuell erst nach einigen Mausklicks voll betriebsbereit ist.
- J Eine ADSB-Verkehrswarnung auf dem externen Flarm-LED-Displays ist zurzeit sicherlich nicht so perfekt (siehe **Hinweis-A**) realisiert wie beim Power-Flarm. Wie endgültig zu verfahren ist, z.B. wenn ein annäherndes Luftfahrzeug gleichzeitig Flarm- UND ADSB-Daten sendet, muß noch die Praxis zeigen. Irgendwer meinte mal "Die Optimierung ist nur noch ein bisschen Änderung in der Software ...". Stimmt, man muss die Optimierung aber auch machen. Die Gewissheit, ADSB-Ziele in der unmittelbaren Umgebung überhaupt angezeigt zu bekommen, kann schon einen erheblichen zusätzlichen Sicherheitsgewinn bringen.
- J Für die Signalisierung von ADSB-Verkehrszielen auf einem externen Display ist eine kleine weitere MUX-Hardware (ca. 15€ Material) unumgänglich.

Nun zu den wesentlichen Vorteilen:

- J Es findet keine räumliche Hardware-Verschmelzung von Flarm- und ADSB-Empfängern statt. In den hochpreisigen Endgeräten führt dies immer wieder zu Interferenzen, was letzt endlich eine geringere Empfangs-Reichweite beim (sehr empfindlichen) Flarm bedeutet als bei einer soliden Solo-Version. Aus der jahrelang zurück liegenden Erfahrung muß weiterhin empfohlen werden, nur Original-Flarm-Geräte einzusetzen, die ausschließlich die grundlegende Flarm-Funktionalität bereitstellen, und auch nicht irgendwo in andere Geräte mit Lizenz verbaut wurden.
- J Es kann auf „einfache“ altbewährte, weit verbreitete und damit gut supportete Flarm-Hardware zurückgegriffen werden.
- J Kostenpunkt? Ein seit Frühjahr 2017 „veralteter“ DVBT-Stick mit R820T-Chipsatz ist (auch nach Abschaltung der ersten DVBT-Generation) für weniger als 20€ zu beziehen.
- J Man besitzt selber die volle Kontrolle über die zusätzlichen ADSB-Daten, ohne Beeinflussung von Flarm, denn Flarm arbeitet in einem ADSB/PC-Fehlerfall/Ausfall ohne Einschränkung weiter.
- J Die Signalisierung von ADSB-Zielen ist auch auf externen LED-Displays möglich, wobei diese zusätzlich noch von Flarm-Zielen unterscheidbar („Trick“ per TX-LED) sind.
- J Bei alternativen, externen ADSB-Empfängern ist man auf das angewiesen, was das betreffende Gerät von Haus aus leistet, und um eine weitere/zusätzliche, individuelle Verkabelung kommt man dabei auch nicht herum. Um manche externe Geräte, genau wie bei der vorgestellten **adsbVektor**-Software-Lösung in eine bestehende Avionik zu integrieren, ist sogar höherer Aufwand nötig. Zwei käufliche, externe, fix und einbaufertige ADSB-Empfänger, konnten allein schon aus den Datenblättern ersichtlich nicht praxisgerecht mit der hier vorgestellten **adsbVektor**-Lösung mithalten. Und zu hoffen, dass für persönliche Belange nachgerüstet wird, auf so was sollte man nie setzen. Der vielfach höhere Preis der bereits käuflichen ADSB-Fertiglösung gegenüber der Stick-Lösung spielte bei den Überlegungen zur Entwicklung von **adsbVektor** überhaupt keine Rolle. Noch einmal ganz deutlich: Volle Kontrolle über die ADSB-Integration zu haben und sich zusätzlich vom Flarm-Monopol abzukoppeln waren die wesentlichen Motivationen für die Entwicklung dieser ADSB-Lösung. Zudem war schon zu Projektbeginn klar, dass diese ADSB-Lösung nicht nur in einem einzigen Flugzeug installiert wird.

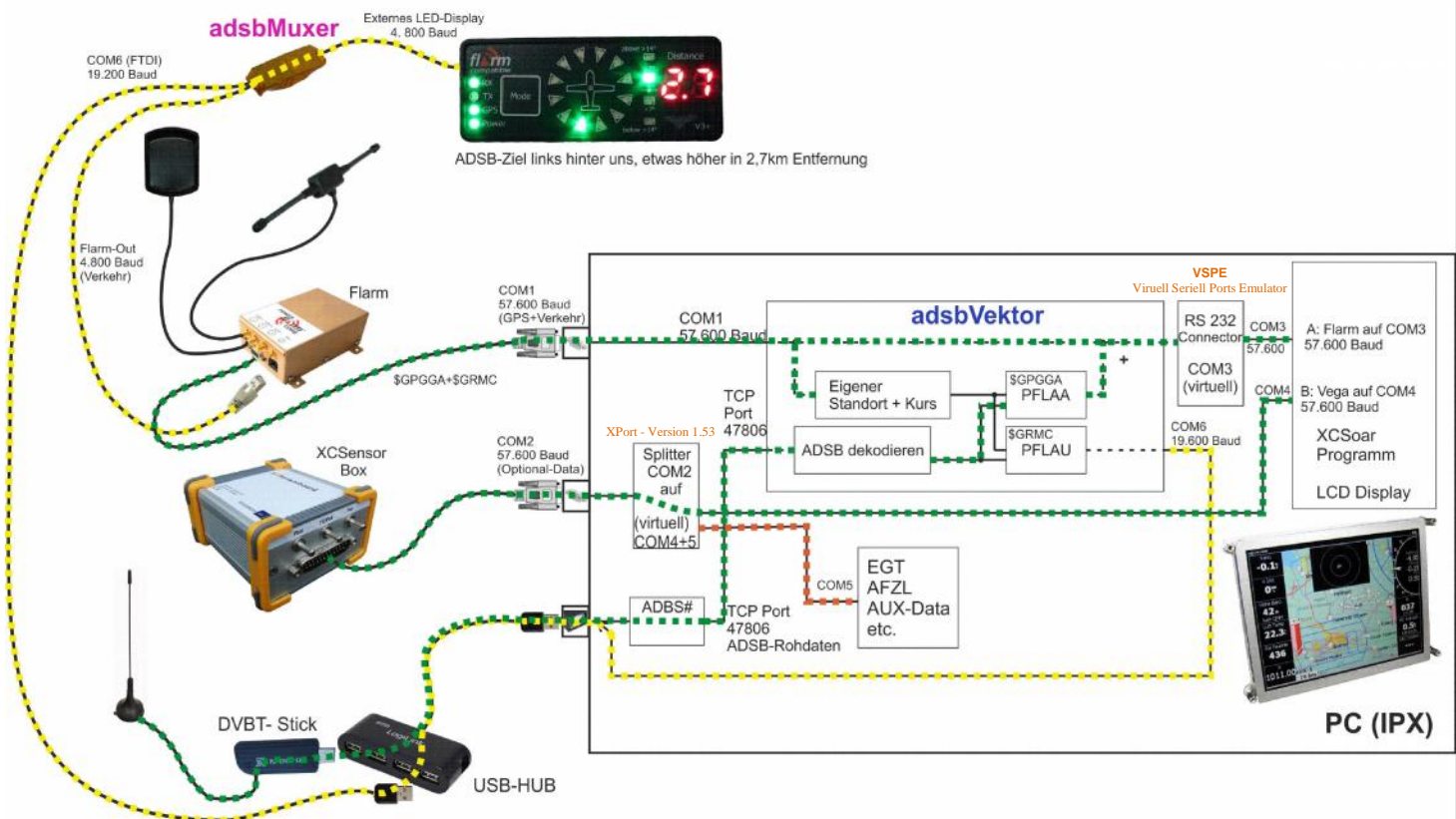
Die "Kosten-Ersparnis" steht in keinerlei Verhältnis zur eingesetzten Entwicklungs-Zeit !!!

Es wird davon abgeraten, dieses System NUR wegen des wesentlich günstigeren Preises als ein Power-Flarm zu installieren. Wer sich für den Einsatz der hier vorgestellten Lösung entscheidet, sollte dieses Manual vollständig gelesen, und halbwegs verstanden haben. Weiterhin muss er bzw. jeder Nutzer mit den Inhalten dieses Dokumentes uneingeschränkt einverstanden sein.

Das schematische Schaltbild-1 zeigt farblich unterstützt, wie die eingangs beschriebenen Grund-Komponenten für die ADSB-Integration mittels DVBT-Stick verschaltet wird.

Die **grünen** Verbindungen zeigen den mindestens notwendigen Aufbau, die **gelben** Verbindungen zeigen die notwendige Erweiterung mit dem **adsbMuxer** für eine ADSB-Signalisierung auf externen LED-Displays. Die **orange** Signalpfade sind über eine Kollisionswarnung hinaus gehende Funktionalitäten und Komponenten, diese werden nur bei individueller Software für eine Triebwerks- oder Flugüberwachung, Aufzeichnung von externen Logger- oder Wetterdaten, sowie bei der erweiterten Verwendung der XCSensorbox benötigt.

### Schematisches Schaltbild-1:



### Was leistet das Programm **adsbVektor** ?

Der ADSB-Rohdatenstrom wird von dem zuvor zu startenden Programm ADSB# mit der Anwendung **adsbVektor** abgegriffen. Der Daten-Übergabepunkt ist eine PC-interne virtuelle TCP-Schnittstelle mit der Adresse 47806. Die aktuellen nmea-Daten erhält **adsbVektor** vom Flarm per seriellen Datenstrom, der über die echte (physikalische) COM1-Schnittstelle an den PC geliefert wird.

) Aus den ADSB-Rohdaten und den aktuellen GPS-Daten werden zwei Flarm-identische Verkehrsdaten-Vektoren (\$PFLAU und \$PFLAA) generiert, **ein kurzer Satz, mit sehr (sehr) viel Programm-Entwicklung!**

Warum denn zwei Verkehrs-Vektoren?

Da die externen LED-Flarm-Displays (wie V2/V3/V3+/V4) einen anderen Verkehrsdaten-Vektor (\$PFLAU) benötigen bzw. auswerten als XCSoar (\$PFLAA), stellt **adsbVektor** gleich für beide Fälle diese zwei typischen Datensätze bereit.

Der PFLAA-Datensatz für XCSOar wird an eine fest voreingestellte, interne virtuelle (Erklärung folgt) Schnittstelle (COM3) übergeben und der PFLAU-Datensatz für die externen LED-Hardware-Displays über einen zusätzlich physikalisch realisierten COM-Port ausgegeben. Dieser zusätzliche physikalische COM-Port (COM6) wird über einen USB-Steckplatz mit FTDI-Chipsatz realisiert, der schon einbaufertig und für den Anwender „unsichtbar“ im **adsbMuxer** integriert ist.

Die von **adsbVektor** generierten Datensätze enthalten, wie das originale Flarm-Datenformat, alle relevanten Verkehrs-Informationen. So sind diese Datensätze von der empfangenden Soft- oder Hardware (XCSOar und z.B. einem V4-LED-Display) nicht von "originalen" Flarm-Datensätzen zu unterscheiden, bis auf das zweckentfremdete TX-Bit im PFLAU-Datensatz, siehe später.

Weiterhin lassen sich mit ein paar einfachen Konfigurationen in der XCSOar Moving-Map "normale" Flarm-Ziele eindeutig von neu dazu gemischten ADSB-Zielen unterscheiden bzw. kennzeichnen.

Jetzt muss man (zur Zeit zumindest) bei XCSOar folgende Eigenschaft beachten: Das Programm XCSOar verarbeitet Verkehrsvektoren (\$PFLAA) ausschließlich über EINE einzige, als Flarm-Gerätetreiber deklarierte Schnittstelle. Wird eine zweite Flarm-Geräte-Schnittstelle als empfangener Treiber geöffnet und mit zusätzlichen Verkehrsvektoren versorgt, so werden diese "weiteren" Verkehrsziele leider ignoriert. In Der Praxis ist es ja auch nicht sinnvoll ZWEI Flarms an einen Rechner anzuschließen.

Somit bleibt vorerst nur die Möglichkeit, XCSOar die originalen Flarm Daten (wie bisher auch) über eine einzige physikalische Schnittstelle zuzuführen, und diesen Datenstrom zuvor IM PC-System mit den von **adsbVektor** zusätzlich generierten ADSB-Verkehrszielen zu "Muxen" -> beide Datenströme werden also vom Benutzer unbemerkt in einen einzigen überführt und dieser dann (wie bisher) an XCSOar an EINE Schnittstelle übergeben.

Genau mit diesem (reinen Software-) Kunstgriff können in XCSOar nun Flarm- UND ADSB-Ziele dargestellt werden. Die genaue Verschaltung der Datenströme wird aus dem Schaltbild-1 ersichtlich.

Nebenbei, es ist damit auch sehr einfach ermöglicht, vorab eine Gewichtung der beiden voneinander unabhängigen Datenströme vorzunehmen. Sonst müsste XCSOar die Gefahren-Gewichtung erledigen!

Wer selber ADSB aussendet, muss seine eigene ADSB-ID (einmaliger 24-Bit-Code) in **adsbVektor** eingeben, dann wird diese (eigene) ID ausgeblockt. Es muss hoffentlich nicht weiter erklärt werden warum das nötig ist.

Die Nutzung des Systems in mehreren Flugzeugen erfolgt nun seit etwa Mitte 2016. Im Anhang werden die wesentlichen Eingabe-Möglichkeiten und Indikatoren der Benutzeroberfläche von **adsbVektor** beschrieben.

Im Gegensatz zu mancher stationären ADSB-Raspberry-Lösung, wird die eigene Position per GPS-Daten in **adsbVektor** permanent aktualisiert. Somit sind die Ausgabe-Zielvektoren auch bei sich änderndem Standort und Bewegungsrichtung immer korrekt. Das ist natürlich eine Grund-Voraussetzung für die Nutzung im Flugzeug! Für stationäre Anwendungen ohne GPS-Anschluss kann in **adsbVektor** eine feste Position vorgegeben werden.

Kurzfassung: Im Programm **adsbVektor** können diverse individuelle Einstellungen vorgenommen werden:

- ) Diverse Filter für uninteressant weit entfernte oder sich in der Flughöhe unterscheidende Ziele,
  - ) gezielte Warn-Signalisierungen und Hervorhebungen in der XCSOar-Moving-Map,
  - ) dabei können für den PFLAU- und den PFLAA-Datensatz völlig getrennte Einstellungen vorgenommen werden,
  - ) ggf. Filterung der eigenen ADSB-Aussendung, Eingabe ortsfester GPS-Standortdaten, etc.
- Siehe „Die **adsbVektor** – Programmbeschreibung“

### Das "zurück"-Rüsten auf "ohne"-ADSB ist mit wenigen Klicks möglich:

Für den fliegen Wechsel braucht nur das Programm **adsbVektor** beendet werden und XCSOar kann mit einer "originalen" Benutzer-Profil-Datei (\*.prf) neu gestartet werden. In dieser ist die COM-Schnittstelle auf die ursprüngliche Nummer, typischer Weise COM-1 (mit **adsbVektor** auf COM-3), eingestellt. Das USB-Kabel vom **adsbMuxer** kann, muss aber nicht vom PC abgezogen werden.

### Was ist eine "virtuelle Schnittstelle"?

Eine virtuelle Schnittstelle ist eine tief im System sitzende, emulierte (nicht physikalisch vorhandene) Schnittstelle, auf die wie eine tatsächlich physikalisch vorhandene Schnittstelle zugegriffen werden kann. Es kann sich um COM- oder TCP/UDP-Schnittstellen handeln.

Die virtuellen Schnittstellen werden per Software erzeugt, benötigen also überhaupt keine zusätzliche (physikalische) Hardware.

### Wozu braucht man diese virtuellen Schnittstellen?

Die in dieser ADSB-Lösung verwendeten Programme stehen standardmäßig über physikalische COM-Schnittstellen mit der physikalischen Außenwelt in Verbindung.

Für jedes einzeln genutzte Programm werden die am PC physikalisch vorhandenen COM-Schnittstellen ausreichen. Auch COM-Schnittstellen, die über einen USB-to-RS232-Konverter realisiert sind, gelten als physikalische Schnittstelle.

Bei der in diesem Bericht vorgestellten ADSB-Variante werden mehrere Programme per **adsbVektor** miteinander kombiniert, so dass wohl immer einige physikalische Schnittstellen am PC fehlen würden. Nebenbei, bis auf genau einen einzigen Fall in dieser Beschreibung, ist die Heraus-Führung der Datensignale in die reale Welt absolut überflüssig, da die



erwähnten Schnittstellen nur als Daten-Übergabe-Punkte dienen. Daher auch die Namensgebung: SCHNITT-Stelle = Daten-Übergabe-Punkt von/zu Programmen.

Es mag sich vielleicht noch herausstellen, daß diese virtuellen Schnittstellen später viel eleganter durch andere Lösungen ersetzt werden können.

Für den Entwickler dieses ADSB-Systems, der als Hardware-Elektronik-Entwickler viel näher an der realen Außenwelt arbeitet als ein reiner Software-Entwickler, war die Wahl der virtuellen Schnittstelle die vernünftigste Lösung, um das gesetzte „ADSB-Ziel“ so schnell wie möglich (mal-eben-fix) erreichen zu können.

## Wie funktioniert die Warnung vor Verkehrszielen in den Anzeige-Geräten überhaupt?

XCSOar warnt grundsätzlich nur dann vor nahendem Verkehr, wenn es den entsprechend gekennzeichneten Datenvektor dafür zugeführt bekommt. Genau so verhalten sich auch die externen LED- Displays.

Also bitte verstehen: Beide Anzeigesysteme berechnen bezüglich drohender Annäherungen selber gar nichts, sondern zeigen "stumpf" das an, was sie an Daten zugeführt bekommen. Die Warn-Berechnung von Flarm-Verkehrs-Zielen nimmt das Flarm-Hauptgerät gänzlich alleine vor. Nichts ist ohne Ausnahme: Nur die LEDs, die die relative Höhe des signalisierten Zieles darstellen, werden tatsächlich vom LED-Display eigenständig aus rel. Höhe und Entfernung des PFLAU's berechnet und entsprechend angezeigt.

Die an die Anzeigesysteme geschickten Datenvektoren (\$PFLAU & \$PFLAA) sind dabei immer auf den eigenen Standort bezogen, enthalten also keine absolute Position. Da bei der sehr kleinen Flarm-Sendeleistung auch öfter mal ein Funk-Datensatz nicht beim Empfänger ankommt, reicht schon ein einziger empfangener Datensatz für eine exakte Kollisionswarnung aus (ein guter, kontinuierlicher Flarm-Empfang ist dennoch sehr wichtig). Auch ein schlecht sendendes oder kurzzeitig abgeschattetes Flarm wird dann bei Kollisionsgefahr noch wenigstens 1x empfangen werden können. Mit dem Flarm-Core und zwei Antennen ist man schon gut bedient. Wünschenswert wäre den(die) Funkchip(s) mit auf kürzestem Weg abgeschlossener Antenne direkt ins Seitenleitwerk (etc.) zu setzen, und per Datenbus mit dem Hauptgerät zu verbinden. So als kleine Anregung, das kann jetzt auch schon jeder für sich selber erledigen 😊.

Die hochselektiven DVBT-Empfänger alleinig für den Flarm-Empfang zu nutzen, oder damit den Nordic-Funkchip zu unterstützen wird bereits in einer sehr kleinen Testrunde umgesetzt.

**Hinweis-A:** Bei einem Flarm MIT integriertem(!) ADSB-Empfang werden BEIDE Annäherungs-Berechnungen (Flarm & ADSB) vorgenommen und miteinander(!) abgestimmt („gewichtet“) und ggf. kombiniert ausgegeben. Wie die Abstimmung genau passiert ist eine undokumentierte Grauzone, wird aber sicherlich praxisgerecht funktionieren. Einen jahrelang optimierten Annäherungs-Warn-Algorithmus und eine langjährig in der Praxis erprobte und angepaßte Funktionsweise läßt sich nicht mal so eben nebenbei nachbilden. Aber wir sind da nahe dran 😊.

Es sei angemerkt, dass jede (!) externe ADSB-Einspeisung denselben Nachteil aufweist, so wie im Power-Flarm-Original lässt sich ADSD in externer Form kaum nachrüsten. Wie "perfekt" die ADSB-Integration im originalen Flarm vorgenommen wurde, ist noch eine ganz andere Frage.

## Die Signalisierung von ADSB-Warnungen, bei der hier vorgestellten DVBT-Lösung per [adsbVektor](#) in der XCSOar-Moving-Map mittels \$PFLAA, ist zurzeit (... mit der Zeit beliebig bis zur original Flarm-Warnberechnung zu verbessern) folgender:

Sobald ein ADSB-Ziel näher kommt als in [adsbVektor](#) eingestellt UND dabei weniger als den ebenfalls dort eingestellten vertikalen Abstand einnimmt, wird dieses Ziel als Warn-Level-1 („orange“) ausgegeben.

Bei Unterschreitung des zweiten in [adsbVektor](#) eingestellten horizontalen und vertikalen Abstandes wird der zweite Warnlevel („rot“) aktiviert.

Auswirkung in der XCSOar-Moving-Map:

Sobald ein sich nähernder ADSB-Verkehr eine erste Mindestdistanz & Höhe (z.B. 5000m & 500m) unterschreitet, wird der zuvor **grünlich** (noch ohne jede Gefahr) dargestellte ADSB-Verkehrsziel-Pfeil in XCSOar **orange** dargestellt. Bei einer zweiten Distanz & Höhe (z.B. 2500m & 250m) in **rot**.

Üblicher Weise sind ADSB-Ziele sich sehr schnell und geradlinig bewegend Objekte, weshalb eine fehlende Warnberechnungs-Zeit (in den drei bekannten Stufen) in der Praxis gar nicht so negativ auswirken könnte wie im ersten Moment angenommen.

Damit das XCSOar-Radar nicht am Rand komplett voll mit ADSB-Verkehrszielen erscheint, kann auch für den Warn-Level-0 (**grünlich**) eine Distanz- und Höhenfilterung (z.B. 20 km/2000m) in [adsbVektor](#) definiert werden.

Auswirkung im XCSOar-Flarm-Radar (kleines in Moving-Map eingeblendetes oder Vollbild):

Alle Ziele, Flarm und/oder ADSB, werden gleichzeitig/parallel dargestellt, wenn kein Ziel einen Warnlevel größer Null beinhaltet. Sowie XCSOar allerdings ein Verkehrsziel mit einem Warnlevel 1,2 oder 3 zugeführt bekommt (gelbe oder rote Verkehrsziele), werden (aktuell in Version 6.8) alle (noch) grünlichen Verkehrsziele ausgeblendet, dieses fördert die Übersichtlichkeit in der Radaranzeige erheblich. Es bleibt also nur noch das eine Gefahrenziel sichtbar. Sollten nun weitere Ziele (Flarm und/oder ADSB) auch noch zur Gefahr werden, erscheinen diese zusätzlich in gelb/rot auf dem Bildschirm, siehe das einleitendes Bildchen.

Der tiefe Blick dazu ins XCSOAR-Manual schadet übrigens nicht: Man kann XCSOAR so konfigurieren, dass bei irgendeinem eintreffenden Warnvektor ein kleines, aber deutlich sichtbares Radarfeld aufpoppt, genau wie im einleitenden Bild dargestellt. Über die „xcsoar-flarm.txt“-Datei in XCSOAR ist es weiterhin möglich, den signalisierten ADSB-Kennungen in der Moving Map eine spezielle Kennung wie „!ADSB!“ ... oder wie auch immer man mag, zuzuordnen. Mit einer hinterlegten Liste aller öffentlich zugänglichen Kennungen, ließe sich sogar der Flugzeugtyp anzeigen.

**Die Signalisierungen von ADSB-Warnungen, bei der hier vorgestellten DVBT-Lösung per *adsbVektor* in Kombination mit dem *adsbMuxer* auf den externen LED-Displays mittels (\$PFLAU), sind zur Zeit folgende:**

- J Grundsätzlich: Originale FLARM-Datensätze werden inhaltlich niemals verändert!
- J Treten NUR Flarm-Ziele auf, verhält sich der *adsbMuxer* absolut passiv (macht gar nichts). Genau so verhält er sich, wenn der USB-Stecker abgezogen wird, das externe LED-Display arbeitet dann unbeeinflusst weiter.
- J Gibt es NUR Flarm-Ziele (wie oben) oder NUR ADSB-Ziele, wird das jeweils nächste Ziel angezeigt.
- J Wenn ein ADSB-Ziel per Dreieck und Entfernung angezeigt wird, ist die "TX"-LED abgeschaltet, so kann eindeutig zwischen Flarm- und ADSB-Verkehrszielen unterschieden werden.
- J Gibt es GLEICHZEITIG Flarm- und ADSB-Ziele, wird das nächst gelegene Ziel angezeigt. Ist es ein Flarm-Ziel, leuchtet die „TX“-LED wie bekannt. Bei einem ADSB-Ziel wird sie zur Unterscheidung aus-geschaltet.
- J Sobald ein Flarm-Ziel in einen Gefahren-Level (1, 2 oder 3) wechselt oder so erstmalig empfangen wird, wird ein eventuell näher vorhandenes ADSB-Ziel nicht mehr angezeigt. Flarm hat immer Vorrang!
- J ADSB-Ziele können vorerst keinen Alarmlevel erlangen, da diese zurzeit noch nicht wie bei Flarm perfektioniert sind. Vielleicht wird das Wagnis auch gar nicht erst unternommen.  
Weitere Anmerkung: Es wurde eine Saison praktisch erfahren, wie es sich anfühlt, wenn durch einen Softwarefehler im externen LED-Display (V3+) selber, auch nur ein einziges Mal (bei einem stundenlangem Flug) FALSCH gewarnt wird (es gab definitiv kein Ziel weit und breit). Daher wird bis auf weiteres lieber etwas "zu wenig" als "zu viel" vor ADSB gewarnt. Selbst eine perfekt funktionierende Kollisionswarnung darf nicht überbewertet werden, kurzum Luftraumbeobachtung steht an erster Stelle!
- J Die Entfernung im zugemischten PFLAU-Datensatz ist auf 99km begrenzt, mehr können die LED-Displays eh nicht anzeigen. Ein original-Flarm-HF-Datensatz wird übrigens kaum mehr als 40 km überwinden können.
- J Die Anzahl der Datensätze zur Verkehrs-Datenausgabe ist beim Flarm-Classic und Flarm-Core nicht immer identisch, um es abzukürzen, damit der *adsbMuxer* mit diesen beiden typischen Flarm-Grundgeräten funktional identisch arbeitet, wird NUR der PFLAU-Datensatz nutzbar durch-geschleift bzw. ausgegeben!

Jede Kritik oder Anmerkung, Wünsche oder Ideen zu den zuvor genannten Signalisierungs-Verfahren bitte an [ipx@schrandt-electronic.de](mailto:ipx@schrandt-electronic.de) schicken.

#### **Bitte immer bedenken:**

Diese Signalisierungen alle zu kennen ist eine Sache, danach im Ernstfall zu handeln eine ganz andere. Das ganze bunte Zeug im Gefahrenfall ist niemals durch aufmerksame Luftraumbeobachtung zu ersetzen.

So ungünstig wie manche Flarm-Displays in einer unteren Ecke im Instrumenten-Brett (wo eben gerade noch so Platz war) verbaut sind, werden diese bei schneller Annäherung einer Gefahr kaum helfen können.

**ACHTUNG:** Je nach Windverhältnissen kann es vorkommen, dass ein auf dem Bildschirm oder auf dem externen Display dargestelltes Ziel in einem ganz anderen zu erwartetem Winkel zur Flugzeugslängsachse signalisiert wird.

Denn, Flarm & Co -Visualisierungen gehen immer davon aus, daß die GPS-basierte Bewegungsrichtung mit der Flugzeugslängsachse EXAKT überein stimmt. Das tut sie zum Beispiel bei der Landung mit ordentlichem Seitenwind, oder bei deutlich vorhaltendem Welle-Fliegen aber NICHT. Da wird dann also z.B. das andere Flugzeug rechts voraus signalisiert, obwohl es sich tatsächlich links voraus befindet!!!

Auf dem Flarm-LED-Display (V2/V3/V3+/V4), kann, bei Ansteuerung durch ein originales Power-Flarm, definitiv NICHT unterschieden werden ob ein signalisiertes Verkehrsziel von einem Flarm- oder ADSB-Sender stammt.

Auf Butterfly-Displays KANN das anders sein, das Data-Port-Manual hätte im \$PFLAA noch ein paar Bits für zusätzliche Informationen frei. Da diese Bits allerdings nicht dokumentiert bzw. selber analysiert sind, konnten diese (möglichen) Zusatzinformationen noch nicht in *adsbVektor* integriert werden.

Wie schon erwähnt, mit dem *adsbMuxer* wird eine Unterscheidung (Flarm- oder ADSB-Datensatz) auf „einfachen“ LED-Displays ermöglicht, indem die TX-Status-LED des Displays etwas "zweckentfremdet" geschaltet wird (s.o.).

Der *adsbMuxer* nimmt weiterhin eine Gewichtung nach Dringlichkeit beider Datensätze vor, damit der dringlichere Datensatz ausgegeben=angezeigt wird. Logisch, das externe LED-Display kann nur EINEN Datensatz zur Zeit anzeigen! Womit spätestens jetzt klar wird, dass der *adsbMuxer* nicht einfach nur beide Datensätze „addiert“ oder eine Anpassung der Baudraten vornimmt.

Bedenkliche Erfahrung: ADSB-Ziele können sehr (sehr) schnell sein, die huschen bei typischem Moving-Map-Ausschnitt in wenigen Augenblicken gradlinig quer über den Bildschirm. Ein Segelflug-Flarmziel bewegt sich deutlich anders. In den uns betreffenden Höhen dürfen knapp über 1km in 10 Sekunden zurück gelegt werden. Wenn sich so ein ADSB-Flieger flott von hinten nähert? Das sollte jedem bewusst sein, der diese ADSB-Lösung anstrebt. Dennoch ... vielleicht besser als gar keine ADSB-Signalisierung, man erschrickt dann nicht ganz so stark.

Und, wer als Motorflieger gar kein Flarm hat, bekommt hiermit zumindest ADSB „für fast umsonst“ auf den vorhandenen PC-Bildschirm oder ein externes LED-Display.

Nie vergessen: Niemand darf sich wegen in der Nähe signalisierten ADSB-Zielen von Flarm-Zielen ablenken lassen! Flarm ist das seit vielen Jahren optimierte Anti-Kollisions-System!

### **Außer dem DVBT-Stick noch weitere Hardware?**

Wenn in Anzeigeräte außerhalb des PCs bzw. XCSOAR eingespeist werden soll, also zusätzlich z.B. noch in eines dieser LED-Displays, dann ... (schon logischer Weise) JA! Da davon ausgegangen wird, daß dieses externe Display (V2-4) direkt und ohne weitere Verzweigungen an das Flarm-Hauptgerät angeschlossen ist: Es bleibt, um dem Display zusätzlichen ADSB-Daten überhaupt zugänglich zu machen, nur die Möglichkeit diese Daten auf dem Kabelweg zum Display hin per externem Multiplexer (mit Gewichtung der Datensätze, der schon mehrfach zuvor erwähnte „**adsbMuxer**“) einzuspeisen. Das sollte jedem einleuchten der sich das schematische Schaltbild-1 angesehen hat.

Erinnerung: Für die physikalische Ausgabe zum externen Display aus dem PC heraus ist eine zusätzliche USB-COM-Schnittstelle notwendig. Der „gewichtende“ Multiplexer in Form eines Y-Kabels (kombiniert mit der bereits erwähnten FTDI-Schnittstelle) ist in einer zusammen hängenden externen Hardware, dem **adsbMuxer** kombiniert. Diese Einheit ist fix und fertig als Blockschaltbild, Schaltung oder ggf. als Fertigteil zu beziehen.

Der Anschluss ist wie im Schaltbild-1 zu sehen, ganz einfach: Das originale Verbindungskabel zwischen Flarm-Grundgerät und externem Display (es wird vom 8-poligen RJ45-Kabel ausgegangen) wird entfernt. Der **adsbMuxer** verbindet entsprechend der gekennzeichneten RJ-Enden nun wieder das Grundgerät mit dem Display.

Über dieses RJ-Anschlusskabel wird vom Flarm kein Strom für den **adsbMuxer** abgegriffen, dieser wird ausschließlich über den USB-Stecker versorgt.

Der **adsbMuxer** gibt nur dann ADSB-Daten an das Display weiter, wenn er mit einem eingeschalteten Flarm-Gerät verbunden ist, welches PFLAU-Datensätze auf 4.800 Baud ausgibt. Die an das Flarm anzuschließende Leitung des **adsbMuxer**'s darf also nicht offen liegen bleiben, was auch gar keinen Sinn machen würde.

Achtung, bei Anschluss eines V2-LED-Displays muss ein Adapter von „6 auf 4“ Pole dazwischen gesteckt werden, die beiden äußeren Adern des 6pol-Kabels werden beim V2 nicht benötigt, siehe ggf. entsprechende Flarm-Dokumentationen.

Die beiden 6-poligen RJ12 Stecker passen nicht nur in RJ12-Buchsen, sondern auch in 8-polige RJ45 Buchsen ("Netzwerk-Verbinden). Zum „Glück“ sind diese Buchsen bezüglich der "inneren 6 Pins" beim Flarm-Classic identisch wie beim Flarm-Core belegt. Bei der 8poligen Variante sind jeweils die beiden äußeren Leitungen identisch (parallel geschaltet) - Bitte erst verstehen (oder in Original-Flarm-Doku nachlesen) dann erst stecken!

Außen liegt also 2x die positive Betriebsspannung und 2x GND/Masse. Die 8-polige Variante kann also mehr Strom liefern (durch 2 zusätzliche Adern), wenn das Flarm den denn auch hergeben darf. Und sie kann über die Buchse noch eine zusätzliche Schirmung führen.

Wer also beim Core an die RJ45-Buchse ein V3 anschließen will, kann das (billigere) 6-pol-Telefonflachkabel mit den Western-Steckern verwenden. Oder er nimmt besser ein Netzkabel und wechselt erst auf den letzten cm zum 6-pol-Kabel mit Stecker. Ohne Löten oder klobigem Adapter ist das leider nicht umsetzbar.

Der verbleibende USB-Stecker des **adsbMuxer**'s wird in einen USB-Port des PCs gesteckt, falls noch einer fehlen sollte kann ein USB-HUB verwendet werden.

Nicht vergessen, der DVBT-Stick muß auch noch einen freien USB-Port finden, der Stick kann mit in den Erweiterungs-HUB eingesteckt werden. Wie immer gilt, alle Kabelverbindungen sind so kurz wie möglich zu halten!

Anmerkung: Es wäre auf Grund der gewählten PC-internen Datenflüsse möglich, auf die zum Flarm gehende Zuleitung vom **adsbMuxer** ganz zu verzichten. Der PC könnte die Gewichtung von Flarm zu ADSB noch eleganter vornehmen als der externe µ-Controller im **adsbMuxer**.

Abgesehen davon, dass dann einige LED-Displays von woanders mit +12V versorgt werden müssten, würde das LED-Display dann nicht mehr ohne den „arbeitenden“ PC, also unabhängig / ausfallsicher betrieben werden können. Daher die Entscheidung für ein sicheres Y-Kabel und gegen den 2fach-Stecker-Muxer.

Sehr reizvoll erscheint die Entwicklung eines FLARM-LED-Displays, mit mehrfarbiger, hochauflösender richtungs-anzeigender Rose, sowie Entfernungs- und Höhenunterschieds-Anzeige in Metern. Der **adsbMuxer** wäre gleich mit in der Display-Einheit enthalten. Vom Material incl. Fertigung her mit Sicherheit preiswerter als jedes zurzeit käufliche LED-Display. Ob der Aufwand allerdings lohnt ... eher nicht :-)

Ganz wichtig für diejenigen, die immer auf Nummer sicher gehen möchte: Ist der PC ausgeschaltet, oder irgendein Programm in der Signalkette klemmt, dann arbeitet das externe Display völlig unbeeindruckt weiter, dafür sorgt die ausgeklügelte Schaltungstechnik im **adsbMuxer**.

Damit man den **adsbMuxer** typisch am Flarm-Classic & Flarm-Core mit identischer Firmware betreiben kann, arbeitet der **adsbMuxer** am Ausgang fest auf 4.800 Baud. Diese Baudrate ist auch Voreinstellung bei den (meisten) externen LED-Displays.

Diese (sehr geringe, noch aus den GPS-Anfangszeiten stammende) Baudrate stellt in diesem speziellen Fall keinen Nachteil dar, da das LED-Display über den PFLAU-Datensatz eh nur ein Mal pro Sekunde aktualisiert wird.

Dieser Umstand macht die Arbeit der "Gewichtung" im **adsbMuxer** nicht unbedingt einfacher, ebenso, dass ein V3+ (SW 3.5) die Checksumme der PFLAU-Datensätze tatsächlich ignoriert!

**Wer auf die Anzeige von ADSB-Zielen auf dem externen Display verzichten möchte bzw. dem die Signalisierung in XCSoar reicht, der braucht den eingangs vorausgesetzten **adsbMuxer** nicht, sowie die Verkabelung nicht zu verändern.**

Die gesamte Kabel-Länge vom DVBT-Stick bis zum PC-Port sollte so kurz wie möglich sein. Abgesehen von möglicher Störstrahlung funktioniert der Stick bei einer zu langen Leitung nicht richtig (wird nicht richtig im System erkannt), das haben mehrere Versuche oder Installationen gezeigt.

Alle zusätzlichen Hardwarekomponenten sollten gut geschirmt installiert werden, da der Datenstrom und damit der Störeinfluß des DVBT-Sticks erheblich sein kann. Bei den bisherigen Installationen hat die ADSB-Nachrüstung in Sachen EMV erstaunlicher Weise praktisch keinen Einfluss (Rauschpegel-Anhebung im Funk) gehabt. Es ist eher so, dass die etwa 5 Watt COM-Sendeleistung bei nicht korrekter Stehwelle manches USB-Device lahmlegen können, darauf wird noch eingegangen.

#### **Wie ist die ADSB-Antenne zu installieren?**

Die Antenne des DVBT-Sticks ist senkrecht ausgerichtet zu befestigen. Dabei ist die genaue Ausrichtung und Länge der Antenne und deren „Standort“ im Gegensatz zur Flarm-Antenne extrem unkritisch. Die originale DVBT-Antenne erzielt am Boden, senkrecht ohne Gegengewicht auf einen Instrumentenpilz gestellt, immer über 100 km Empfangsreichweite, auch weit über 250 km sind keine Seltenheit.

Jedem, der sich schon etwas mit ADSB auseinander gesetzt hat, wird die enorm hohe Sendeleistung der ADSB-Sender in Form eines teuren Transponders bekannt sein ... diese ist etwa um den Faktor 10.000 höher als beim Flarm.

Ein kurzer Antennenstummel direkt am DVBT-Stick angebracht, wird typischer Weise im relevanten Entfernungsbereich (unter 10km) voll ausreichen, gute 50-Watt auf 1GHz ausgestrahlt, sicher zu empfangen.

Es macht sogar Sinn, die DVBT-Antenne „NICHT optimal“ zu positionieren, damit uninteressant weit entfernte ADSB-Ziele erst gar nicht den Daten-Verarbeitungsprozess beanspruchen. Denn 99,7% der empfangenen Verkehrsziele wird der Nutzer später per **adsbVektor** sowieso ausblenden wollen. Weiterhin werden so auch Überlagerungen bzw. Ausblockungen stark verringert, auch wenn ein einzelner ADSB-Extended-Squitter zeitlich recht kurz ist. Es sind allerdings sehr (sehr!) viele Squitter unterwegs.

Da es gut an diese Stelle passt: Auch wenn die gesendeten Datenpakete nur sehr (112+8 µs) kurz sind, je mehr Teilnehmer zufällig variierend einmal pro Sekunde aussenden (ohne vorher zu lauschen ob die Frequenz frei ist, so wird es gemacht), steigt logischer Weise die Wahrscheinlichkeit von gegenseitigen Ausblockungen/Überlagerungen in Empfangsreichweite. Dieses gilt für Flarm und ADSB-Squitter gleichermaßen, natürlich nur auf deren entsprechenden Frequenzen.

*Das deckt sich insbesondere mit der (Flarm-) Erfahrung, immer gegen sehr späten Nachmittag die größten Reichweiten zu erzielen: Die relativ wenigen, die noch in der Luft sind, sind auch noch relativ hoch. "Wenig" und "Hoch" ... sind die entscheidenden Worte :-)*

Der DVBT-Stick kann mit seiner Antenne also irgendwo im Instrumentenpilz untergebracht (untergestopft) werden. Bei geschlossenen Metall- oder Kohlerümpfen muss die Antenne natürlich etwas ins Freie schauen können und darf nicht durch größere metallische Gegenstände an einer Rundumsicht gehindert werden.

Der DVBT-Stick ist ein HF-High-Tech-Produkt in aller-billigster Verpackung, also ohne jeden wirklichen mechanischen Schutz von Antennen- und USB-Anschluss. Es empfiehlt sich je nach Einbauort, den Stick mit ultrakurzer Verlängerung in eine Buchse einzustecken, damit man eine mechanische Entkopplung schafft. Der Antennenanschluss ist ebenfalls vor mechanischen Angriffen oder Vibrationen zu schützen.



Hier der Vorschlag einer mehrfach gut bewährten Selbstbau-Antenne (Lambda/4-Koaxial-Antenne) für unseren ADSB-Anwendungsfall:

Das Original DVBT-Kabel ggf. kürzen und die äußere Isolierung auf einer Länge von **65mm** frei legen, die innere "Isolierung" nicht entfernen. Dann den „Schirm“ umkrempeln .....

... und das ganze mit Schrupfschlauch überziehen, fertig!



Auf gleiche Weise lässt sich eine erstaunlich gute Flarm-Antenne herstellen, die Strahler-Länge beträgt dann **86mm**. Nach dem gleichen Prinzip geht auch eine Flugfunkantenne mit **555mm** Stab-Länge.

Anmerkungen: Die Längen sind etwas verkürzt, da die Signallaufzeit in Leitern kleiner als die Lichtgeschwindigkeit ist. Die Flugfunk-Version arbeitet allerdings nicht mehr gut, da das typische RG58-Kabel am Strahler-Ende zu dünn ist.

### Stört der ADSB-Empfang die vorhandene Elektronik?

Der eigentliche „Störenfried“ für die restliche Avionik kann der DVBT-Stick selber und dessen USB-Zuleitung oder der HUB sein. Also, wenn möglich ganz auf eine USB-Verlängerung oder den HUB verzichten und alles an Kabeln so kurz wie möglich halten.

Eher ist es beim empfangsempfindlichen DVBT-Stick so herum, die "Störquelle" kann das Funkgerät selber sein:

Eine hohe HF-Feldstärke oder eine schlecht angepasste COM-Antenne kann für manch unangenehme Überraschung bei in der Nähe (bis zu einigen Metern) arbeitenden Elektroniken sorgen.

Auch wenn ein installiertes Funkgerät schon immer ohne Beanstandungen funktioniert hat, bedeutet das noch lange nicht, dass es perfekt im Sinne des Erfinders gewerkelt hat.

Sollte beim Drücken des Funk-Knopfes das E-Vario immer deutlich ausschlagen, wird das schon ein Anzeichen für eine nicht perfekte Anpassung der Antenne sein.

Diesem Anpassungs-Fehler kann man sehr einfach mit einer Stehwellenmessung in der Mitte- und den Randfrequenzen auf die Schliche kommen.

Das „gemeine“ daran ist, dass sich bei einer anderen Frequenz ein bestimmter, sogar reproduzierbarer Störeffekt leider gänzlich anders (oder gar nicht mehr) äußern kann. HF geht eben sonderbare Wege, die oft genug an Magie grenzen.

Es wäre nicht das erste mal, dass eine neu installierte elektronische Komponente schon ewig bestehende „Unzulänglichkeiten“ zu Tage fördert. Die fehlerhaften Auswirkungen werden dann gerne mit einem „ging ja vorher einwandfrei“ auf die neu hinzu gekommene Komponente geschoben!

Mehrfach hat sich bei Funkgeräte-Tests gezeigt:

Aus einer großen geschlossenen Metall-Flugzeughalle kann die COM-Sendeleistung (ca. 5 Watt) kaum entweichen! Die rundherum wieder nach innen / zurück reflektierte HF-Strahlung kann an zufälligen Positionen durch das Überlagerungsprinzip erhebliche Feldstärke-Maxima/Minima mit unvorhersehbaren elektronischen Beeinflussungen hervor rufen.

Diese Phänomene sind leider kaum reproduzierbar, am nächsten Tag sind sie plötzlich nicht mehr vorhanden ... da das Flugzeug vielleicht nur ein paar cm verrückt wurde.

Also sollten derartige "Endkontrollen" möglichst ganz weit im freien stattfinden, kaum zu glauben, aber folgende Fälle waren eindeutig zu reproduzieren:

Die Antennen-Anpassung wird sogar noch erheblich beeinflusst,

- ) wenn das Flugzeug einige Meter VOR einer Metall-Halle steht, auch ist es ein Unterschied für die Stehwelle, ob die Funkanlage auf freiem Erdboden steht oder sich das Flugzeug in der Luft befindet!
- ) Selbst der Anhänger aus Alu, an dem das Segelflugzeug aufgerüstet wird, hat einen erkennbaren Einfluß auf das Messergebnis. Die Sperrtopfantenne sitzt meist im Seitenleitwerk, das Funkgerät selber sehr nahe am Anhänger auf der Rampe! Also sollte der Rumpf für Messungen oder Anpassungs-Optimierungen der COM-Stehwelle nicht nur aus dem Anhänger rausgezogen werden, sondern das Flugzeug aufgerüstet (Metallgestänge in den Flächen sind auch Gegengewichte) auf möglichst freier Fläche vermessen werden. Wie bei der Kompaßkompensierung auch :-)

**So vorgenommen, kommt die Stehwellenmessung dem später vorgesehenem Betriebspunkt, also dem Flug, am nächsten.**

Zurück zum Thema, es macht Sinn, den DVBT-Stick von vorn herein zu schirmen, das funktioniert z.B. mit einer antistatischen Tüte, Alufolie oder am saubersten mit einem Metallgehäuse (das muss nicht unbedingt geerdet werden).

Es ist leider so, dass der DVBT-Stick in Einzelfällen gerne, bei typischer COM-Feldstärke in der Nähe, mit temporärer Aufgabe seiner Tätigkeit oder Abmeldung im System reagiert. Diese „Einzelfälle“ unterscheiden sich leider kaum von

anderen Installationen, die fast identisch eingebaut sind, und ohne jegliche Schirmung oder perfekt kurze Kabel bestens funktionieren. Aber logo, kein Einbau ist exakt wie ein anderer, gerade was das resultierende HF-Design angeht. Welche Komponenten genau die „DVBT-Stick-Aufhänger“-Phänomene auslösen oder daran beteiligt sind, konnte bisher leider noch nicht geklärt werden. Wer genau dazu Erfahrungen oder Tips beisteuern kann, möge sich bitte melden!

#### Wer sich nur eine ADSB-Bodenstation aufbauen möchte:

Mit einem einfachen Windows-PC kann sich mit **adsbVektor** und weiterer, unten genannter Freeware, ein offline (!) ADS-B-Radar in Echtzeit auf den XCSoar-Moving-Map/Radar-Bildschirm geholt werden. Eine Überwachung im relevanten Höhenband des Luftraumes am Startplatz ist somit möglich.

Für diesen Anwendungs-Fall ist keine aktive GPS-Quelle nötig, da der eigene Standort in **adsbVektor** fest vorgegeben werden kann.

Mal abgesehen davon, dass zur Zeit noch viel weniger Motorflugzeuge Flarm-Signale aussenden als ADSB, wird gerade Flarm wegen seiner sehr geringen Sendeleistung auch bei kleiner Entfernung nicht immer sicher am Boden empfangen werden können.

Ausgesendete ADSB-Signale von sich tief (und schnell) annähernden Flugzeugen hingegen, werden dagegen auch schon mit einer kurzen Stummelantenne aus einigen km Entfernung gut empfangen. Für Empfangs-Reichweiten von über 350km (!) gibt es im Internet unzählige Nachbauanleitungen für optisch schon abenteuerlich anmutende Antennen-Konstruktionen.

Auf einem Startschreiber-Computer kann **adsbVektor** im Hintergrund laufen und ein riesiges LED-Display (die Idee zu einer Eigenentwicklung kam schon) könnte den Verkehr deutlich erkennbarer als ein typischer LCD-Monitor anzeigen.

Ein möglicher Weise durch die Winden-Schleppstrecke führender ADSB-Flugweg kann zwar nicht verhindert, aber vom Bildschirmbeobachter erkannt und als Warnung bekannt gegeben werden. Bleibt zu hoffen, dass militärischen Jets auch ab und zu ziviles ADSB aussenden.

Der Empfang und die Decodierung von ADS-B-Daten auf 1090 MHz sind laut einer Publikation im "Funkamateure" übrigens erlaubt.

#### Zukunftsmusiken:

1.)

Für einen versierten Software-Entwickler ist es sicherlich möglich, ADSB#, **adsbVektor** und die nötigen Schnittstellen-Manipulationen alle in einem einzigen Programm zu verschmelzen.

Als hauptamtlicher Elektronik-Hardware-Entwickler programmiert man meist sehr nahe orientiert an der Hardware. Soll heißen: Das Programm **adsbVektor** ist ganz sicher noch an vielen Stellen optimierbar. Wer sich daran beteiligen möchte kann sich gerne unter [ipx@schrandt-electronic.de](mailto:ipx@schrandt-electronic.de) melden 😊.

Wenn irgendwer aus dem XCSoar-Team jemals zu viel Langeweile haben sollte, könnte derjenige den hier vorgestellten umständlichen Software-Parcours zur ADSB-Integration ... auch komplett sauber in die PC-Version von XCSoar implementieren. So als kleine (bzw. ganz große) Anregung ;-)

2.)

Die Version des Flarm-Core bietet die Option einer zweiten Flarm-Antenne.

) Die erste Antenne sendet und empfängt.

) Die zweite Antenne sendet und empfängt ab V6.4 ebenso, laut Dokumentation konnte die zweite Antenne in der Vergangenheit NUR empfangen.

Dieses Konzept ist technisch sinnvoll und wird nicht in Frage gestellt. Die Antennen-Hardware bietet nur vier Empfangs-Szenarien:

) Beide Antennen liefern jeweils einen gültigen und korrekten (identischen) Datensatz, einer davon wird genommen.

) NUR die erste Antenne liefert einen gültigen Datensatz, dieser wird dann verwendet.

) NUR die zweite Antenne liefert einen gültigen Datensatz, dann wird jener verwendet, logo.

) Keine der beiden Antennen empfängt einen Datensatz mit korrekter Checksumme, dann bleibt das Flarm-Radar eben ohne Inhalt.

Nun aber: Der zweite Empfangs/Sende-Chip wird durch einen Mikrocontroller emuliert, der sein Antennen-Signal von einem empfangsstarke DVBT-Stick bezieht. Der Nichtelektroniker stelle sich einfach vor, es würde eine kleine Schaltung (der „DVBT-Channel“) anstelle des zweiten verbauten Empfangs/Sende-Chip eingelötet, die sich genau so verhält wie der originale Empfangs/Sende-Chip.

Wie erwartet hat sich in der Praxis gezeigt, dass der empfangsstarke "DVBT-Channel" immer noch ein gültiges Datenpaket empfangen kann, wenn die erste und auch die zweite (nun ersetzte) Flarm-Antennen-Empfangs-Einheit schon lange nichts gültiges mehr empfangen würden.

Zusammenfassung: Der erste Flarm-Funk-Kanal bleibt unverändert. Der zweite wird auf DVB-T-Empfänger umgerüstet, das ergibt eine oberstarke Empfangsleistung ohne in die elementare Funktionsweise des Flarms eingreifen zu müssen -> Ja, es funktioniert, mit etwas höherem Aufwand sogar beim Standard-Flarm :-)))

Super: Eine Änderung des Funkprotokolls kann dieser Lösung rein gar nichts anhaben.

Allerdings nur von Ober-Spezial-Spezialisten umzusetzen, mit ein paar Risiken verbunden, was das "Darf ich das denn überhaupt" angeht. Achso, Zukunft ist schon da .... es funktioniert wie erwartet!

Mist ist allerdings, dass die zweite Antenne ohne weiteren Spezi-Aufwand nun nicht mehr sendet ... . Das tat sie aber in den ersten Core-Software-Versionen auch nicht.

3.)

Mit einem weitem DVBT-Stick können bekannter Maßen sehr weitreichend Flarm-Ziele empfangen und auf der XCSoar-Moving-Map angezeigt werden. Ob schon jemand 50-100 km entfernt in der Welle hängt, ist damit offline zu erkennen.

Das Programm **adsbVektor** ist dafür bereits vorbereitet. Diese Integration ist erst dann möglich, wenn das Flarm-Protokoll dauerhaft dekodiert werden kann ;-)))

4.)

Mit dem immer wieder erwähnen DVBT-Stick und einer weiteren 50€-Hardware (868MHz-ISM-Sender, wie im Flarm) läßt sich ein von Flarm völlig unabhängiges Antikollisions-System aufbauen, ein „originales“ Flarm ist gar nicht mehr nötig.

Dieses ginge abwärtskompatibel nur, wenn wie unter 3.) angesprochen, das Flarm-Protokoll bekannt wäre. Ansonsten würde man ein neues System aufbauen, welches zum bestehenden Flarm inkompatibel wäre. Die Übergangszeit von "alt" nach "neu" wäre viel zu lang und damit inakzeptabel gefährlich.

Ob jemand (s)eine feste ID aussendet, oder einen zufälligen Platzhalter, könnte er selber entscheiden. Für die Kollisionswarnung ist keine ID nötig. Die Verwendung eines kostenlos nutzbaren ISM-Bandes macht die Frequenz allerdings immer zu einer Spielwiese jeglicher Manipulationen. In dieser Hinsicht hat die umstrittene Verschlüsselung der vom Flarm übermittelten Daten-Bits schon einen Sinn.

*Es sollte zumindest jedem klar sein, daß ein original-Flarm mit einem DVBT-Stick & einem ISM-Sender vollständig ersetzt werden kann:*

*Letzt endlich braucht nur jeder Flieger die wesentlichen Inhalte des \$GPGGA und \$GRMC-Datensatzes auszusenden, also Position, Geschwindigkeit, Kurs etc., möglichst in einem nicht ganz so ASCII-Zeichen-verschleuderndem Protokoll. Dazu reicht schon Hardware für ein paar Euro.*

*Jeder in der Nähe des Sendenden könnte somit seine eigene Berechnung vornehmen, wie das empfangene Ziel zu bewerten ist. Also, ist eigentlich ganz simpel!*

*Letztendlich benötigt man an Hardware ...*

) einen DVBT-Stick für den Flarm-Empfang auf 868MHz und

) einen zweiten für ADSB auf 1090MHz.

) Dazu kommt nur noch ein 868MHz-ISM-Sender.

) Alles an einen einzigen USB-Hub angeschlossen.

*Kostenpunkt ... keine 100€. Rest ist „nur noch etwas Software“.*

Auch hier ist die Zukunftsmusik seit Anfang 2017 schon Realität geworden: Die genannten Punkte sind keine Hirngespinnste, es sind entsprechende Prototypen schon im Einsatz. Sie überfordern in der praktischen Vervielfältigung „für alle“ aber zeitlich leider eine einzeln daran (nebenbei aus Freude) werkende Person, die auch noch ein Berufs- Privat- und Fliegerleben hat.

Also, wer sich mit Freude an der Technik bei der Weiter-Entwicklung in jeglicher Form (außer unbegründet meckern) beteiligen möchte, sei gebeten sich unter [ipx@schrandt-electronic.de](mailto:ipx@schrandt-electronic.de) zu melden.

### **Der erste Testflug ...**

... war anfangs ernüchternd: Kein Ziel weit und breit! Na gut, das war schon beim Start absehbar, dann wird eben der Filter-Umkreis, ab dem Ziele überhaupt erst signalisiert werden auf 50 km erhöht. Immer noch nix. 100, 150km ... nix.

Da stimmt doch was nicht?

Genau: Die Höhenausblendung ist auf +/- 100m eingestellt. Dafür gibt es doch den "Unlimited"-Knopf -> oh, nun ist der Radar-Bildschirm voll :-)

Auf der 8,4-Zoll-Moving-Map ist allerdings kein Verkehrsziel-Pfeil zu sehen ... aha, wenn der Bildschirmausschnitt in XCSoar über etwa 60km beträgt, wird kein Verkehrspfeil mehr angezeigt. Jetzt zum Testen zwar doof, aber voll praxisgerecht.

Das Flarm-Radar zeigt nun also diverse Ziele, alle liegen auf dem äußeren out-of-range-Ring des XCSoar-Flarm-Radars. Die Bewegungs-Richtungen sind durch die extrem weit außerhalb des Darstellungsbereiches liegenden Ziele entsprechend verzerrt. Den radialen Filter unter 50km (ohne Höhenfilter) zu legen brachte im Bereich einer Kontrollzone endlich das, was man an visuellen Eindrücken erwartet hat. Auf einen praktisch sinnvollen Höhen- und Horizontalabstand zur Kollisionswarnung gesetzten Filter ... war wieder gar kein Ziel mehr vorhanden.

Naja, so soll es doch auch sein?

Nun haben bereits unzählige Flüge gezeigt: Es funktioniert praxisgerecht!!! Eine Gewöhnung an die geradlinig vorbeiziehenden ADSB-Ziele ist allerdings notwendig ... wenn man selber kurbelt.

## **Abspann,**

### **und weitere gesammelte Gedanken, ob das jemand wirklich bis zum Ende ließt?**

Die Nutzung von **adsbVektor** mit den dazu gehörigen Beschreibungen ist ein seit Anfang November 2015 in der Praxis genutzte Möglichkeit zum Empfang von ADSB, für den keinerlei Gewähr übernommen wird. Es gilt sinngemäß all das, was an Einschränkungen und Warnungen auch im Flarm-Manual auf vielen Seiten des "Kleingedruckten" steht. Wer einfach nur einen Schalter umlegen will, greift bitte zum Power-Flarm & Co.

Es haben Praxis-Flüge in ausreichend relevanter Anzahl, auf mehreren Flugzeugen und Besatzungen mit Erfolg statt gefunden, so dass mit gutem Gewissen das Programm **adsbVektor** (Version 20160309) seit Ende 2016 per Emailanfrage zugeschickt wird.

### **Auf spezielle Nachfrage: Was bringt mir überhaupt ein ADSB-Empfang?**

Dazu muss man erst einmal etwas weiter erklärend ausholen. Der folgende Abschnitt kann nur einen Bruchteil dessen wiedergeben, was für eine vollständig richtige Darstellung und Erklärung nötig wäre! Los geht der Erklärungsversuch:

Ein **Primärradar** wertet die vom Boden ausgesendeten und vom Flug-Objekt **reflektierten Radarstrahlen** aus. Idealer Weise erfolgt das von mehreren Standorten aus.

Mittels Abstrahlwinkel und Signal-Laufzeit(en) wird dann die Position des zu erfassenden Objektes berechnet, die Höhenbestimmung funktioniert dabei nur sehr eingeschränkt.

Diese Art der Positionsbestimmung ist recht störanfällig (kontrastarm) und hängt stark von der Oberflächenbeschaffenheit und dem Material des zu erfassenden Objektes ab.

Das Objekt verhält sich also passiv, es ist also "Glückssache", ob der Controller jemanden so sieht. Wolken, Vogelschwärme oder ein Pulk Segelflugzeuge können von einem Primär-Radar erfasst werden.

Es kommt eher selten vor, dass man überhaupt von einer Bodenstelle betreut oder beobachtet wird, wenn man keinen aktiven Transponder an Bord hat. Nicht jeder FIS hat uneingeschränkten oder überhaupt Zugriff auf ein Primärradar.

Beim **Sekundär-Radar** sendet ein im Flugzeug eingebauter **Transponder aktiv ein ungerichtetes Antwortsignal** (Downlink) mit mindesten 75 Watt (!) aus, sobald es einen Radar-Abfragestrahl (Uplink) vom Boden aus empfängt.

Dieses Verfahren ist weit präziser als das Primär-Radar und in der zivilen Luftfahrt daher Standard.

Leider erfordert es einen aktiven/eingeschalteten Transponder (XPDR) der in der Anschaffung richtig teuer ist. Daher haben dieses Gerät oft auch nur diejenigen eingebaut, die es wirklich nutzen (oder das müssen).

Sogenannte Mode-C Transponder liefern mit angeschlossenem und aktiviertem Höhendecoder im wesentlichen nur die Flughöhe und einen selber einstellbaren 4stellige Dezimalzahl (Code - SQUAK).

Seit etwa 2008 ist in vielen Lufträumen (z.B. über 5000ft immer) der Mode-S -Transponder erforderlich. Die Technik ist prinzipiell mit dem Sekundär-Radar identisch.

Es wird bei einer Antwort dieser neueren Geräten zusätzlich (mit vielen weiteren Daten) eine weltweit eindeutige ID mit gesendet. Damit ist dieser Transponder identifizierbar und bei Bedarf sogar einzeln ansprechbar.

So, und nun endlich der Zusammenhang mit ADSB: Die "neuen" Mode-S Transponder können überhaupt erst ADSB-Daten (die die "GPS"-Position enthalten) aussenden. Und weil (die meisten) Mode-S-Geräte es prinzipiell ermöglichen, kann diese Möglichkeit eben genutzt werden.

Kann genutzt werden???

Ein typischer Mode-S Transponder wird überhaupt nur dann GPS-behaftete ADSB-Daten aussenden, wenn er mit einem GPS speziell gekoppelt ist! Von Haus aus tut er das nicht.

Selber fremde ADSB-Daten empfangen und als "Annäherungswarnung" ausgeben kann ein typischer Mode-S Transponder erst echt nicht!

Was das ganze nun noch mit Flarm zu tun hat? Erst einmal ... gar nichts.

### **Das "Radar" ist in erster Linie für die Verfolgbarkeit von Luftfahrzeugen vom Boden aus entwickelt worden.**

Die zugehörigen Frequenzen (Up/Downlink) sind aus gutem Grund nicht frei nutzbar, NUR der Empfang ist, sagen wir mal „geduldet“.

### **Flarm entstand dagegen ganz gezielt für eine reine Air-to-Air Kollisionswarnung.**

Das Flarm-System hat daher von Anfang an Sender, Empfänger und Kollisionsberechnung incl. Warnkonzept als Komplettlösung angeboten. Nur so war Einbau und Nutzung als Kollisions-Warngerät für jedermann ohne Avionik-Kenntnisse oder wesentliche Zulassungen möglich.



Das verwendete Funkprotokoll ist nicht verfügbar, daher ist der typische Flarm-Nutzer auf den Kauf von entsprechender kompatibler Flarm-Hardware angewiesen. Flarm arbeitet allerdings auf öffentlicher ISM-Frequenz. Darauf darf jeder mit kleinen Einschränkungen machen was er will, genau deshalb ist diese Frequenzwahl etwas riskant.

Hätte-hätte-Fahrradkette:

Mit der Wahl eines geschützten (= kostenintensiven) Frequenzbandes zum Projektstart von Flarm wäre Flarm heute nicht mehr nur eine (schon offiziell akzeptierte) "Bastellösung". Genau aus diesem geschichtlichen Hintergrund gibt es das Flarm-System in der momentan technisch realisierten Funk-Ausführung!

Anders geschrieben: Ohne den preiswertesten Projektstart auf dem kostenfreien ISM-Band über IDoCAS wäre Flarm sehr wahrscheinlich nicht ansatzweise so verbreitet wie es jetzt ist.

Mit diesem Monopol müssen wir Flarm nun nehmen wie es ist. Wer etwas in seinem Sinne programmieren kann und möchte ... nutzt dafür einfach die schon fertige Hardware ;-)

Durch Nutzung diverser technischer Möglichkeiten fangen Flarm und ADSB seit einiger Zeit an zu verschmelzen. Mit **adsbVektor** zum Beispiel.

Denn, obwohl es mindestens 2000€ kostet ADSB per typischem Transponder auszusenden („ADSB-out“), ist der völlig autarke ADSB-Empfang („ADSB-in“) durch das bekannte ADSB-Funk-Protokoll mittels einfachster Hardware fast umsonst.

### **Eine langfristige und praxiserichte Anti-Collisionseinrichtung?**

Wer ADSB sendet UND empfängt, kann die anderen Verkehrsteilnehmer für sich auswerten und teilt auch ändern seine Bewegungs- bzw. Positionsdaten mit.

Wenn sich alle so eingerichtet haben, ist Flarm als Kollisionssystem sofort überflüssig.

Der Empfang anderer ADSB-Verkehrsteilnehmer ist preislich nicht mehr der Rede wert. Daher empfangen mittels verschiedensten Hardware-Lösungen schon sehr viele „Kleinflieger“ die möglichen ADSB-Ziele und visualisieren diese.

Eine ADSB-Aussendung ist zur Zeit nur per teurem Mode-S-Transponder (und Willen! des Nutzers) legal möglich.

Und genau das ist der Punkt, deshalb senden kaum Kleinflieger ADSB aus.

Wenn fast alle also nur empfangen ... dann bringt das in Sachen Kollisionswarnung erst einmal fast gar nichts. Verstanden!?

### **Lösung, ADSB-Mini-Sender?**

Es müsste also ein legales preiswertes Gerät im Handel zu erstehen sein, welches wie ein Mode-S-Transponder ADSB-Signale in einer sicherer Kollisionsgefahren-Reichweite aussendet, als Leistung wären 1 Watt (auf nicht-öffentlicher XPDR-Frequenz) voll ausreichend. Idealer Weise ist das Gerät gleich ergänzt durch einen integrierten ADSB-Empfänger.

Somit ließe sich ein externes Kollisions-Warnsystem in fast beliebiger Konstellation mit relevanten Daten versorgen. Flarm wäre damit technisch, insbesondere wegen seiner zu geringen Sendeleistung und Manipulierbarkeit, sofort Geschichte.

Solange es noch keine Transponder-Pflicht (mit Mode-S) für alle Luftfahrzeuge gibt, wäre eine solche nur ADSB-Daten auf Transponder-Frequenz (1090 MHz) aussendende Lösung, eine relativ preiswerte, stromsparende und Reichweiten-sichere Alternative zum echten Transponder, noch einmal:

Diese "Lösung" sendet ADSB (exakt wie ein Mode-S-XPDR) aus, nur mit erheblich geringerer Leistung als ein XPDR, aber mit weit mehr Sendeleistung als Flarm. Der ADSB-Empfang geschieht nebenher.

Einen solchen ADS-B-Sender preislich attraktiv (unter 1000€) und als hervorragende Alternative (Ablösung) zu Flarm anzubieten wird ausschließlich den etablierten Transponder-Herstellern vorbehalten bleiben.

Denn nur diese werden den kostspieligen Weg der offiziellen Zulassung erfolgreich angehen können/wollen. Auf 1090 MHz zu senden ... ist für jeden Bastler strafbar.

Abgesehen davon erfordert die Implementierung des ADS-B-Sende-Protokolls eine absolut fehlerfreie und sichere Hard- und Software.

Die technische Umsetzung ist keinerlei Herausforderung, die bürokratische um so mehr. Wer jetzt schon einen zugelassenen Mode-S-Transponder entwickelt hat, der bräuchte dieses Gerät nur um einen erheblichen Teil kostspieliger Bauteile erleichtern und erhielte damit ohne nennenswerten Hard- und Software-Aufwand den angesprochenen „ADSB-Mini-Sender“.

Ein GPS-Empfänger könnte gleich noch mit integriert werden, damit keine externe GPS-Quelle mehr angeschlossen werden müsste.

Ob dieser Aufwand insgesamt lohnt ist die erste Frage, ob ein Hersteller das überhaupt will die zweite, soll der Kunde doch den bereits zugelassenen Mode-S-Transponder kaufen, der ist fertig :-)

Und ob ein solcher reiner Zukunfts-ADS-B-Sender in einer Kontrollzone jemals wie ein Mode-S-(Sekundär-)Transponder akzeptiert wird? Lassen wir uns überraschen.

## Flugweg-Tracking und Privatsphäre

Denjenigen, die sich schon tiefer mit ADSB beschäftigt haben wird klar sein, daß sich bewegte, echte ADSB-Verkehrsziele nicht ansatzweise so einfach wie Flarm-Verkehrsziele per HF „simulieren“ lassen. Vom „Dürfen“ ganz zu schweigen.

Erinnerung: Nicht alle (kaum ein) Transponder in einem privaten Kleinflugzeug sendet zurzeit ADSB-Daten aus. Es sind dem Autor mehrere Vereins-Motorsegler bekannt, die zwar ein Power-Flarm eingebaut haben, deren Mode-S-Transponder selber aber KEIN ADSB aussendet. So mancher fliegt auch grundsätzlich mit nicht eingeschaltetem Mode-S-Transponder durch die Gegend.

Wie war noch die Frage, ach ja, warum dann überhaupt ADSB empfangen?

Es gibt genügend Luftfahrzeuge, die verpflichtet sind, ADSB auszusenden. Und immer mehr, die es nicht müssen, tun es freiwillig.

*Wer seinen Transponder dazu befähigt ADSB auszusenden, der sollte wissen z.B. auch auf flightradar24 gesehen zu werden.*

Zur Verdeutlichung: Wer mit einem Mode-S-Transponder auf "On/Alt" fliegt, dem ist völlig klar, fast flächendeckend von der DFS bzw. dem FIS gesehen zu werden.

Spätestens bei aufgeschaltetem ADSB, wird jedem (!) per Internetzugang ermöglicht, den exakten Flugweg und das Luftfahrzeugmuster eindeutig (!) zu verfolgen. Mit der entsprechenden Hardware ... sogar offline im Umkreis von bis zu 400km. Wem es gerade entfallen ist, eine solche Hardware wird in diesem Dokument beschrieben.

Für die Öffentlichkeit „heimlich“ und dennoch offiziell und sicher mit Führung der DSF unterwegs zu sein, ist nicht mehr möglich.

Interessant ist weiterhin zu wissen, dass ein dazu befähigter sowie eingeschalteter Mode-S-Transponder auch dann regelmäßig, etwa in Sekunden-Abständen ADSB-Daten aussendet, wenn er gar keinen abfragenden Radarstrahl (übrigens auf 1030 MHz) von einer Bodenstation empfängt. Der Fluglotse sieht den betreffenden Flieger auf seinem Sekundärradar-Empfänger nicht, wir mit ADSB-Empfang dagegen schon!

Noch etwas zum Flarm-System, jeder der mit einer originalen Flarm-Software im Flieger unterwegs ist, sollte wissen:

Sein Flug kann im Internet zu Zeit noch ohne jede Barriere "getrackt" werden, wenn er sich im Empfangsbereich einer OGN-Empfangsstation befindet UND er das No-Tracking-Flag in der Flarm-Voreinstellung gelassen hat.

Die dafür notwendigen OGN-Empfangsstationen kann wiederum jeder bereit stellen. So mancher Segelflugverein tut dieses sogar, ohne vorher seine Mitglieder darüber zu informieren oder gar gefragt zu fragen.

Praxis: Der an einer solchen Station in Reichweite vorbei-Fliegende mit original installiertem Flarm und Tracking-Flag=0 hat überhaupt keine Chance zu verhindern, dass sein Flugweg im Internet von jedermann nachvollziehbar wird.

Aktuell mit Flarm-Software 6.4 kann man zwar eine zufällige ID konfigurieren, so wird es vom Hersteller zumindest beschrieben. In Wirklichkeit wird immer noch in regelmäßigen Abständen die originale, unveränderbare ID ausgegeben. Wir werden von manchen schon für ganz schön blöd gehalten bzw. belogen.

Der private Flieger, der das Tracking komplett ausschließen will, muss sein Flarm ausschalten, womit er selber keine Flarm-Kollisionswarnungen mehr erhält und auch potentiellen Kollisionspartnern dieses Feature verwehrt. Es war nicht der Sinn von Flarm, dieses auszuschalten!

Der versierte Elektroniker ... würgt den Sendebefehl zu den Funkchips ganz gezielt ab (der noch versiertere Programmierer ... macht das viel eleganter), womit andere Flarm-Verkehrsteilnehmer NICHT mehr von einem selber gewarnt werden. Das sollte also nicht jeder so machen!

Was jedem klar sein sollte: Fliegen zwei ADSB-aussendende ohne weitere Komponenten aufeinander zu ... erhält noch keiner der beiden eine Warnung. Genau wie beim Mode-S-Transponder auch nicht!

Der Airliner weicht nur deshalb einem anderen Transponder, aus weil der Lotze entweder die Information dazu gibt, oder das TCAS (als völlig separat verbautes, fast unbezahlbares Anti-Collisions-Empfangsgerät im Airliner) vor einer gefährlichen Annäherung warnt.

Somit muss auch ein entsprechender ADSB-Empfänger im „Ideal-Gerät“ integriert werden, oder extern betrieben werden, deren Daten dann erst, für eine Kollisionswarnung verarbeitet werden.

Mit "mal eben so über Nacht" wird auch ein etablierter Avionik-Hersteller die Aufgabe nicht lösen. Und ein Elektronik-Einzelkämpfer schon gar nicht.

*Bei dieser Teil-Entwicklung steckte, wie so oft, als erstes das nötige Technikinteresse dahinter, der finanzielle Sinn schon abgrundtief nicht mehr :-)*

## Eine Bitte an alle Leser:

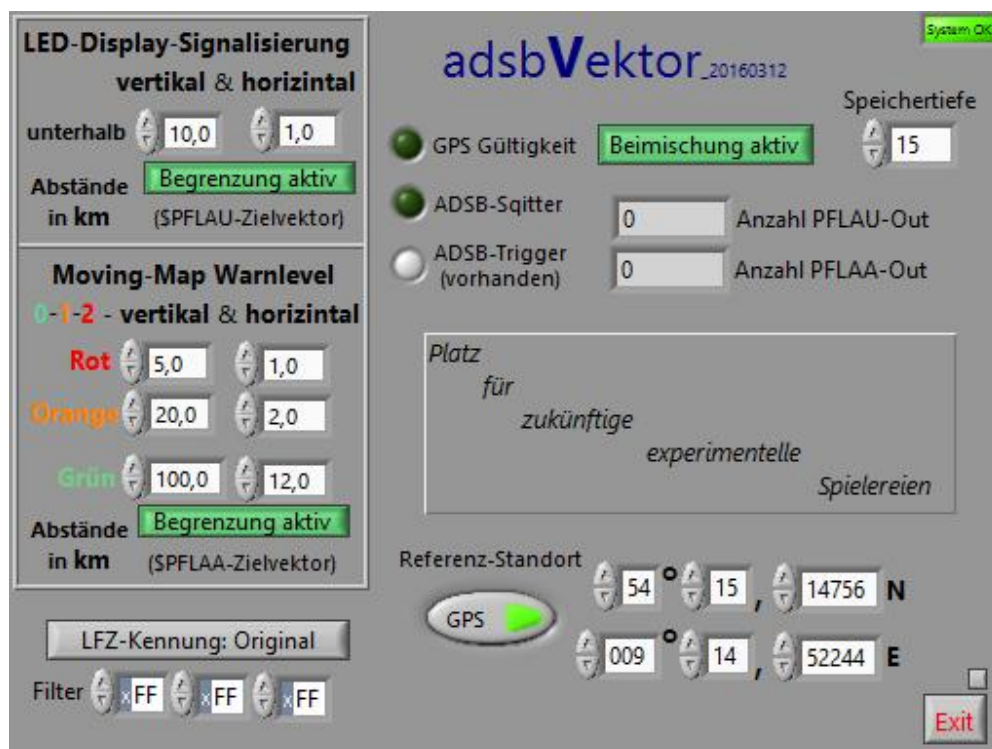
Nicht bei jeder kleinen Dokumenten-Änderung kann der gesamte Text erneut auf vollständigen inhaltlichen Abgleich mit allen ggf. verlinkten Dokumenten überprüft werden. Das ganze Projekt IPX, ADS-B & Co ist zwar ein technisch sehr interessantes Thema, was vom Autor mit Freude, aber NICHT aus kommerziellen Gründen betrieben wird.

Eine gewisse Betriebsblindheit stellt sich ebenfalls bei vielfachem Durchlesen eigener seitenlanger Dokumentationen ein. Daher ist bitte jede Unstimmigkeit an [ipx@schrandt-electronic.de](mailto:ipx@schrandt-electronic.de) mitzuteilen, Danke!

## Anhang

### Die **adsbVektor** – Programmbeschreibung:

Die Anleitung zur Benutzeroberfläche von **adsbVektor** wird entsprechend der zukünftigen Versionen permanent angepasst.



Es können folgende, wesentliche Einstellungen vorgenommen werden:

#### Movin-Map Warnlevel - Einstellung der ADSB-Entfernungen zur Aktivierung der Warnlevel 0/1/2 im PFLAA-Datenvektor

) Einstellung "**Rot** - vertikal & horizontal":  
ADSB-Ziele werden bei Unterschreitung dieser beiden Abstände in **rot (Warnlevel-2)** auf der XCSoar-Moving-Map oder dem XCSoar-Flarmradar angezeigt. Mit der Einstellung „0,0“ kann die „Umfärbung“ unterbunden werden.

) Einstellung "**Orange** - vertikal & horizontal":  
ADSB-Ziele werden bei Unterschreitung dieser beiden Abstände in **orange (Warnlevel-1)** auf der XCSoar-Moving-Map oder dem XCSoar-Flarmradar angezeigt. Mit der Einstellung „0,0“ kann die „Umfärbung“ unterbunden werden.

) Einstellung "**Grün** - vertikal & horizontal":  
ADSB-Ziele werden überhaupt erst bei Unterschreitung dieser beiden Abstände in **grün (Warnlevel-0)** auf der XCSoar-Moving-Map oder dem XCSoar-Flarmradar angezeigt. Ansonsten sind diese Ziele ausgeblendet. Durch die hohe ADSB-Reichweite ist diese Begrenzung in der Praxis äußerst sinnvoll (notwendig).  
Mit dem Knopf "Keine Begrenzung / Begrenzung aktiv" können beide vorherigen Eingaben außer Kraft gesetzt werden. Dieser Knopf hat keinen Einfluss auf die obigen Orange/Rot-„Einstellungen“.

**LED-Display-Signalisierung** - Einstellung der ADSB-Entfernungen zur Signalisierung von ADSB-Zielen auf externem Display mittels PFLAU-Datenvektor, in Verbindung mit dem **adSBMuxer**

ADSB-Ziele werden erst bei Unterschreitung dieser beiden Abstände auf dem LED-Display angezeigt. Alle weiter entfernten Ziele (nur ADSB!) werden ausgeblendet.

Mit dem Knopf "Keine Begrenzung / Begrenzung aktiv" können beide vorherigen Eingaben außer Kraft gesetzt werden.

Achtung: Bei geringen Geschwindigkeiten über Grund (!) ist die LED-Richtungsanzeige naturgemäß zunehmend ungenau! Wahrscheinlich aus diesem Grunde wird beim original-Flarm am Boden NUR die RX-LED aktiviert, es findet trotz Ziel-Empfangs keine Anzeige von Richtung und Entfernung statt.

Da gerade die Entfernung zu ADSB-Zielen wichtig erscheint, ist deren Signalisierung dennoch bei geringen GND-Geschwindigkeiten bei diesem DVBT-System voreingestellt.

Im Betrieb als Bodenstation mit festgelegter GPS-Koordinate, wird ein externes Display "Nord"-orientiert angesteuert.

Mit dem Knopf "**LFZ-Kennung: Original/DD-ADSB**" kann jede originale ADSB-Kennung (24-Bit-ID) gegen ein festes „DDADSB“ umbenannt werden

Damit wird in der Moving-Map oder im Radar ein ADSB-Ziel als „ADSB“ markiert. Dazu ist es noch nötig, in der Datei „xcsoar-flarm.txt“ die Zeile <DDADSB=ADSB> zu ergänzen.

Die in der „**Filter**“- Einstellung gewählte ADSB-Kennung (24-Bit-ID) wird bei Empfang vollständig ignoriert

So wird die möglicher Weise selber ausgesendete ADSB-Kennung nicht auf dem LED-Display oder der Moving-Map dargestellt.

Mit dem Schalter „**Referenz/Standort**“ wird entweder die aktuelle GPS-Position ausgewählt oder die darunter eingetragene Koordinate.

Wird der Schalter als „GPS“ in grün dargestellt, werden die aktuellen GPS-Koordinaten als Standort verrechnet. Diese Einstellung ist im Fluge zu verwenden!!! Daher ist dies die grün untermalte Voreinstellung.

Der Schalter in roter Darstellung führt den ADSB-Berechnungen die FEST eingestellte Koordinate zu!

Mit dem Schalter „**Beimischung aktiv**“ oder „Keine Beimischung“ werden dem Flarm-Datenstrom entweder ADSB-Daten hinzugefügt oder nicht.

Mit „**Exit**“ wird das Programm nach 3 Sekunden Betätigung beendet.

Mit „**Speichertiefe**“ wird gewählt, wie viele ADSB-Ziele gleichzeitig verarbeitet werden sollen.

Mit „**Beimischung aktiv**“ werden dem original-Datenstrom die generierten ADSB-Vektoren hinzu gefügt (Voreinstellung)

#### Statusanzeigen

- J „GPS Gültigkeit“, Status des GPS-Empfangs:
  - o Bei grün (dauerleuchtend) ist ausreichender GPS-Empfang vorhanden, ansonsten erfolgt keine ADSB-Ausgabe, außer es ist ein fester Referenz-Standort gewählt
- J „ADSB-Squitter“ (aufblitzend)
  - o Es werden relevante ADSB-Sequenzen empfangen
- J „ADSB-Trigger vorhanden“ (aufblitzend)
  - o Es sind berechnete ADSB-Verkehrsziele als \$PFLAA und \$PFLAU vorhanden. Diese werden je nach den Filtereinstellungen ausgegeben oder geblockt.
- J „System-OK“ (dauerleuchtend)
  - o Bei „System OK“ arbeitet das Programm korrekt, ansonsten erfolgt keine ADSB-Ausgabe
- J Die Felder „PFLAU-Out“ und „PFLAA-Out“ geben die Anzahl der bisher tatsächlich ausgegebenen Datenvektoren wieder.



## Zweiter Teil, die Installation:

### PC-Systemvoraussetzungen:

- J Betriebssystem: Win7 und Win10 (WinXP sollte auch gehen, nicht getestet), das verwendete Betriebssystem sollte ohne zu klemmen auf dem PC laufen. Die folgende Beschreibung wurde auf mehreren Win7 und Win10-Rechnern (32+64 Bit) erfolgreich angewendet.
- J Die Installation auf Win10/32-Bit IPX'en ist bereits vielfach problemlos vorgenommen worden, diese Anleitung ist darauf hin optimiert.
- J Achtung: Der zurzeit verwendete virtuelle Schnittstellen-Treiber VSPE [„Virtual Serial Ports Emulator“ von „eterlogic“] ist nur auf einem 32-Bit-System Freeware! Ggf. muss auf eine funktional vergleichbare Alternative gewechselt werden.

### Hinweise:

- J Als erstes: PC ausschalten!
- J Alle im Vorfeld des Dokumentes gegebenen Hinweise sind weiterhin zu beachten.
- J Es wird dringend empfohlen, vor dieser umfangreichen Installation ein komplettes 1:1 Festplatten-Abbild zu erstellen, auf das man im Fall der Fälle zurück greifen kann.
- J Die gesamte Verkabelung ist soweit wie möglich laut Schaltbild-1 vorzubereiten. Wer von irgendwelchen Nummerierungen, Bezeichnungen, Einstellungen usw. abweicht, muss die darauf abgestimmten Vorgaben entsprechend anpassen. Als Untermauerung dieser Anleitung dient das Schaltbild-1, dessen Inhalte für den "Installateur" verständlich und nachvollziehbar sein sollten.
- J Sind genug USB-Ports vorhanden? Ist abzusehen, dass z.B. ein USB-Hub wegen der Anzahl erforderlicher USB-Anschlüsse notwendig ist, dann ist dieser als erstes anzuschließen. Dessen Installation erfolgt in Windows automatisch, die erfolgreiche Integration ist mit einem Speicher-Stick einfach zu überprüfen. Auf bewährte Qualität und Funktionsweise ist zu achten, also nicht das aller billigste Ding einsetzen. In einen USB-HUB sollten eher USB-Devices angesteckt werden, die eher langsam Daten mit dem PC austauschen, wie Mäuse oder Tastaturen und wenig Strom verbrauchen.
- J Einmal vorgenommene USB-Verbindungen sollten im weiteren Verlauf nicht mehr in andere USB-Ports umgesteckt werden.
- J Es wird davon ausgegangen, dass ein PC-versierter Nutzer die ADSB-Integration vornimmt oder zu Seite steht. Der späteren Übersicht halber sollten alle Programme geordnet und in wieder-erkennbar erstellte Verzeichnisse installiert werden. Der Installateur und auch der spätere Nutzer werden nicht völlig drum herum kommen, sich etwas mit dieser ADSB-Lösung zu beschäftigen, nur mit einem Grundverständnis wird man das System vollständig und sinnvoll nutzen können.
- J Der PC ist vom Internet zu trennen.
- J Alle im Folgenden erwähnten Programme und Tools sind (erst nach Testphase) in einer einzigen Zip-Datei zusammen gefasst und per Emailanfrage erhältlich. Damit erspart sich der einzelne viel Sucherei, da die verwendeten Programme oder Treiber nicht immer gleich auf der ersten Google-Seite stehen.
- J ACHTUNG, den USB-Anschluss des **adsbMuxer**'s noch NICHT in den PC einstecken, USB-Port aber einplanen!
- J Es kann Sinn machen, diese Anleitung vorab einmal durchzulesen.

Nun geht's also los, der PC darf eingeschaltet werden:

### 1.) Den DVBT-Stick installieren

Folgender Stick wird in diesem Dokument verwendet und wird auch allgemein für den ADS-B-Empfang empfohlen:



Entscheidend ist der darin enthaltene Chipsatz: **RTL2832U/R820T chip/tuner**

Schrandt electronic / IBS (IchBinSegelflieger)  
Hermann Löns Weg 3a - D-27412 Tarmstedt  
Copyright (c) 2014 Schrandt electronic. Alle Rechte vorbehalten.



Der DVBT-Stick ist in einen freien USB-Port einzustecken. Ein originaler, nicht per HUB aufgeteilter Port ist immer erste Wahl.

Es darf nun KEINER der beim Stick mitgelieferten Treiber installiert werden, auch wenn Windows dazu auffordert! Letzt endlich muss der DVBT-Stick im System bzw. im Gerätemanager so wie folgt erkannt und angemeldet sein:

Diese durch den Stick hervorgerufene Signalisierung ...  ... ist kein Problem!

Wichtig ist dagegen, ob der Stick zusätzlich noch so ...  ... unter „USB-Geräte“ erscheint. Das „(Interface 0)“ in der untersten Zeile kann je nach Installation auch fehlen.

Ist der Stick nicht wie beschrieben im System aufgeführt? Sondern als ZWEI Ausrufezeichen unter „Andere Geräte“ gekennzeichnet, ganz oben? Meist ist es so, in diesem Fall wird „Zadig“ weiterhelfen.

) [www. http://zadig.akeo.ie/](http://zadig.akeo.ie/)

Mit dem Programm „Zadig“ muss der DVBT-Stick gesondert installiert werden. Der Stick wird im Zadig-Auswahlfenster als „Bulk-In, Interface (Interface 0)“ zur Auswahl auftauchen. Dieses „Interface 0“ also auswählen und „Install Driver“ anklicken. Auf kleinen Bildschirmauflösungen muss eventuell unter dem Reiter „Options“ noch „Advanced Mode“ ausgewählt werden, damit man den „Install-Driver“-Button überhaupt sieht! Diesen drücken, ohne vorher sonstige Einstellungen zu ändern. Nach ein paar Augenblicken erscheint die frohe Botschaft: „The Driver was installed successfully“. Zumindest unter Win10. Dann „Close“ drücken das und Programm beenden.

Ansonsten, folgendes gilt nur unter Win7:

Nun wird „Installing Driver“ als durchlaufender Balken angezeigt. Da war ein kreativer Programmierer am Werk, es werden ein paar lustige Kommentare zur gefühlten Verkürzung der Wartezeit über dem Balken angezeigt.

Auch wenn die Installation dann nach zig Minuten laut Kommentar fehl schlägt („The driver installation failed“), war der DVBT-Stick dann nach dem „Close“ und einem Neustart bisher immer genau "richtig" wie oben vorgegeben im System angemeldet zu finden.

Wenn sich „Zadig“ nach der Fehlermeldung aufhängt wird der PC zur Not gewaltsam ausgeschaltet. Nach dem nächsten booten ist der Stick dann in 9 von 10 Fällen wie gewünscht „installiert“.

Eventuell sind mehrere Versuche und weiter helfendes googeln nötig, es funktioniert auf jeden Fall und die Vorgehensweise der korrekten Stick-Anmeldung wird auch für ein ADSB-Heimradar an vielen Stellen im Internet beschrieben. Die richtige Einbindung des DVBT-Sticks ins System ist der einzig nicht durchschaubare Punkt bei der ADSB-Integration.

Erst wenn der DVBT-Stick als „Interface 0“ im Gerätemanager (wie oben abgebildet) zu finden ist, kann zum nächsten Schritt gegangen werden.

Beim einem Neustart des PCs oder erneutem Einstecken des Sticks: Meldungen wie „Neue Hardware gefunden“ und das aufpoppende „Assistent für das Suchen neuer Hardware“-Fenster einfach wegklicken. Auch die Meldung „Neue Hardware gefunden ... Beim Installieren ist ein Fehler aufgetreten ...“ einfach ignorieren.

Für alle Betriebssysteme gilt: Der Stick muss immer wieder (in etwa) so wie oben abgebildet im System zu finden sein: Im oberen Teil des Gerätemanagers unter „Andere Geräte“ mit EINEM gelben Ausrufezeichen, im unteren Bereich unter „USB-Geräte“ als „Bulk-In, Interface“ oder „Bulk-In, Interface (Interface 0)“.

Dieses seltsames Phänomen ist ein (einziges) Mal aufgetreten: Nach dem Anschluss bzw. Nutzung von USB-HUBs ist es vorgekommen, dass auf einmal andere, völlig unbeteiligte USB-Ports lahmgelegt wurden und eventuell erst nach einem Rechner-Neustart wieder funktionierten. Hier half auf Dauer nur, die USB-Geräte an anderen Steckplätzen als zuvor anzuschließen, oder alles über einen einzigen USB-HUB laufen zu lassen. Das geht also auch, selbst wenn es eingangs als „nicht optimal“ angegeben wurde. Da muss man als Windows-User eben manchmal durch.

Wenn auf einem PC der DVBT-Stick auf einmal nicht mehr von ADSB# angesprochen werden kann (Device used, found, but not ...) dann hilft:

- ) Den unter „Audio-, Video- und Gamecontroller“ plötzlich als „Audio Realtech“ ... erkannten DVBT-Stick mitsamt Treiber (Häckchen bei „Die Treibersoftware für dieses Gerät löschen“ setzen ) deinstallieren,
- ) Stick abziehen ...
- ) ... wieder anstecken ...

... und der DVBT-Stick sollte wieder ansprechbar sein.

Folgendes hat sich bisher in unzähligen PC Kalt- und Warmstarts über Monate hinweg auf diversen Rechnern gezeigt: Ist der DVBT-Stick wenige Male nacheinander wie beschrieben von ADSB#, auch nach einem System-Neustart ansprechbar, wird sich das zukünftig auch nicht ändern!

Selbst wenn USB-Ports getauscht oder nachträglich mit einem HUB erweitert wird, bleibt der Stick immer wie gewünscht ansprechbar, selbst wenn dieser durch einen baugleichen ersetzt wird. An diesem stabilen, praxisgerechten Zustand ändert sich auch nichts nach einem Backup/Restore, selbst auf einen anderen PC mit gänzlich neu verteilten USB-Ports.

Sehr beruhigend!

Der Rechner darf zwischenzeitlich allerdings NICHT mit dem Internet verbunden werden, sonst könnten fälschlicher Weise automatisch "geeignere" Treiber nachgeladen werden!

Die DVBT-Installation sollte also geklappt haben, dann kann es weitergehen:

## 2.) Das Rohdaten-Empfangsprogramm ADSB# (gelesen: ADSB-Sharp) installieren:

Das Programm ADSB# (= ADSBSharp.exe) ist per Copy-Paste in ein passend angelegtes Verzeichnis zu "installieren". Ein Doppelklick auf ADSBSharp.exe startet das Programm (eventuelle Anfrage der Firewall wegklicken).

Sollte die Fehlermeldung kommen, dass „.NET Framework 3.5 (enthält .NET 2.0 und 3.0)“ noch installiert werden muss bzw. fehlt ... bitte erledigen. Das geht sogar offline(!) mit dem Tool „NET-Framework-3.5-Offline-Installer-v2.3 exe“, in Verbindung mit einer Win10-Installations-CD.

Ist ADSBSharp.exe per Doppelklick erfolgreich gestartet, muss nichts weiter eingestellt oder verändert werden.

Wenn jetzt, nachdem der "Start"-Knopf gedrückt wurde, im rechten Bereich eine sich permanent ändernde Anzahl von Frames angezeigt wird, dann werden ausgesendete ADSB-Rohdaten (Squitter) empfangen, erster Glückwunsch!

Wenn „Frames/sec“ auf Null bleibt, kann es daran liegen, dass in den DVBT-Stick noch gar keine Empfangs-Antenne eingesteckt wurde, oder sich diese in einem gut abgeschirmten Bereich befindet!?! Draußen, bei freier Antennen-Rundumsicht wird man immer einige ADSB- Ziele empfangen können.

So sieht das Ergebnis dann in etwa aus, wobei „154 Frames/sec“ schon recht viele Daten sind.



Es ist ratsam, das PC-System nun ein bis zweimal komplett neu zu starten, um dann nach wenigen Klicks (ggf. fehlende Treiber ignorieren) und der Doppelklick auf ADSBSharp#.exe + Start-Knopf, sofort wieder an der ersten Glückwunsch-Stelle anzukommen.

Das muss auch dann noch funktionieren, wenn der PC komplett stromlos gemacht wurde, also alle USB-Devices wirklich abgeschaltet wurden. Daher ist ein Neustart NICHT mit einem „Komplett abschalten“ zu vergleichen!

Wenn das wiederholt reibungslos klappt, kann es mit besten Erfolgsaussichten weiter gehen.

Und falls nicht, hier noch ein paar Starthilfen:

<http://www.rtlsdr.com/2013/07/adsb-windows-adsb-decoder-for-use-with-a-rtl-sdr/>

<http://www.pennula.de/spezialthema-flugzeug-radar-mode-s-adsb-ads-b-decoder-air-flugfunk-sdr-radio-rtl-usb.htm>

<http://www.sprut.de/electronic/pic/projekte/adsb/adsb.htm>

Der versierte PC-Nutzer hat nun schon die ersten Verknüpfungen auf den Desktop gesetzt oder gleich eine entsprechende \*.bat angelegt ;-)

## 3.) Den virtuellen Schnittstellentreiber VSPE installieren:

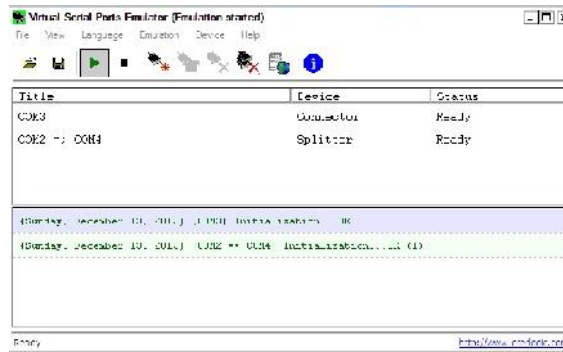
Nun wird der Gerätetreiber für die virtuellen, und ggf. mehrfach abzugreifenden Schnittstellen namens „VSPE“ installiert:

-> <http://www.eterlogic.com/Products.VSPE.html>

Nach der Installation wird die vorbereitete Datei c3-c24.vspe oder c3.vspe (im Download Zip-Paket) nach dem Start von VSPE geladen. Es wird dann folgendes Programmfenster zu sehen sein:

Schrandt electronic / IBS (IchBinSegelflieger)  
Hermann Löns Weg 3a - D-27412 Tarmstedt  
Copyright (c) 2014 Schrandt electronic. Alle Rechte vorbehalten.





Fehlermeldungen gibt es nur, wenn die Schnittstellen-Nummern schon zuvor von irgendeinem anderen Programm belegt sind, ggf. sind die belegten COM-Port-Nummern in andere zu ändern.

Die Bedeutung der beiden Zeilen unter Titel / Device / Status:

„COM3“:

COM3 ... ist die absolut notwendige zu erzeugende virtuelle Schnittstelle, an der das Programm **adsbVektor** die originalen Daten aus dem Flarm-Gerät, gemuxt mit den zusätzlich generierten ADSB-Vektoren an die erste Schnittstelle von XCSoar übergibt. Siehe Schaubild!

COM2 - COM4/5:

Die echte Schnittstelle COM2 wird auf eine virtuelle COM4/5 als Splitter gelenkt, auf COM4/5 kann nun mit mehreren Programmen zugegriffen werden. Siehe Schaubild!

Hinweis, die Splitter-Funktion ist bei langsameren Rechner-Systemen u.U. nur bei relativ geringen Baudraten zu nutzen.

Gutes und bewährtes Ausweichprogramm: XPort

Wer die Splitter-Funktion von COM2 auf COM4/5 nicht nutzen möchte, kann (und sollte!) die COM 2->4/5 -Funktion löschen und kann dann zum Beispiel jegliche Datenquelle an COM2 anschließen und diese Schnittstelle wie gewohnt direkt in XCSoar oder jedem anderen Programm als COM2 abgreifen.

#### 4.) Das Programm **adsbVektor** und die dafür notwendigen Libarys installieren:

Für die Ausführung von **adsbVektor\_Ausgabedatum.exe** per Verknüpfung oder Doppelklick sind wieder spezielle Treiber nötig, damit Windows etwas mit dem Programm anfangen kann.

Die per Doppelklick zu startenden Installationsroutinen für **adsbVektor** sind zusammen mit dem Programm **adsbVektor** als kostenloser Download erhältlich sobald ein paar erste Nutzer (Versuchskaninchen) diese Anleitung ohne jegliche Rückfrage erfolgreich nutzen konnten. Und wie erwähnt erst ab Mitte 2016.

-> Die entscheidende Datei „**adsbVektor\_Ausgabedatum.exe**“ wird in ein passend angelegtes Verzeichnis kopiert.

Eine detaillierte Beschreibung für das Programm **adsbVektor** selber, mit all seinen Konfigurationsmöglichkeiten, ist in der zukünftig per Download bereit gestellten \*.zip-Datei enthalten.

#### 5.) Vorgehensweise zur Hard- und Software-Installation des **adsbMuxer**'s

Soll der gelbe Signalpfad mit dem **adsbMuxer** genutzt werden, dann muss der darin physikalisch enthaltene Schnittstellenbaustein FTDI232RL im System angemeldet werden und auf COM6 eingestellt werden.

Wer KEINE Ausgabe auf einem V2/3/4er LED-Display beabsichtigt, kann sofort zu Punkt 6.) gehen.

Zum Verständnis, es muss gar nicht der Muxer selber installiert werden, sondern der im **adsbMuxer** eingesetzte USB-RS232-Konverter Chip von FTDI (FTDI232RL).

Die Installation des entsprechenden Schnittstellentreibers kann, wie im das PDF-Dokument "AN\_119\_FTDI\_Drivers\_Installation\_Guide\_for\_Windows7.pdf" beschrieben, vorgenommen werden.

Dennoch bitte die folgende Kurzfassung wegen der späteren COM-Nummern-Einstellung zu Ende lesen!

Installation des FTDI-Chips für PC-Anwender OHNE Internetverbindung auf einem **Win10-32Bit-System**:



- J Die nötigen FTDI-Treiber-Dateien werden mit dem Muxer mitgeliefert, diese sind auch unter <http://www.ftdichip.com> zu finden.
- J Das FTDI-Kabel bzw. den **adsbMuxer** mit seinem USB-Anschluss an den PC anschließen.
- J In den Gerätemanager gehen, dort wird unter „Andere Geräte“ ein "TTL232R-3V3" mit einem **gelben Ausrufezeichen** signalisiert werden, sofern der Treiber nicht schon zuvor vollständig installiert wurde. Vorweg, es sind zwei Gerätetreiber = Installations-Schritte nötig!
  - 1.)
    - J Auf die TTL232R-3V3 -Zeile gehen, rechte Maustaste und "Treibersoftware aktualisieren" auswählen.
    - J „Manuell auf dem Computer selber suchen“, und von das Verzeichnis ...
      - ➔ „FTDI232RL - CDM v2.12.10 WHQL Certified“
    - ... anwählen (Unterordner einbeziehen).
      - ➔ „Weiter“
      - ➔ ... die Treibersoftware wird „Erfolgreich“ installiert.
      - ➔ „Schließen“.
  - 2.)
 

Nun wird im Gerätemanager ein "USB Serial Port" mit **gelbem Ausrufezeichen** angezeigt.

    - J Auf diese Zeile gehen, rechte Maustaste und wiederum "Treibersoftware aktualisieren" auswählen.
    - J Erneut "manuell suchen" auswählen, Pfad wie zuvor auswählen und auf "Weiter" Treiber .... wird installiert und dann "Schließen".

Wenn die Aufforderung nach einem Neustart kommt, diesen sofort erledigen.

Dann wieder in den Gerätemanager gehen, nun sollte unter „Anschlüsse (COM & LPT)“ ein "USB Serial Port (COMx)" erscheinen. Das kleine "x" steht für eine automatisch vergebene COM-Port-Nummer.

- J Nun die rechte Maustaste auf die "USB Serial COM Port"-Zeile, und auf "Eigenschaften klicken.
- J Dann in "Eigenschaften, den Reiter "Anschluss-Einstellungen" anklicken, und dort dann auf "Erweitert" gehen.
- J Dort wird nun wird die "COM-Anschluss-Nummer:" ... auf "COM6" geändert (ggf. erneut versuchen). Wird die Nr.-6 als „schon verwendet“ gekennzeichnet, das Ändern dennoch bestätigen. Je nach Vorgeschichte des Rechners kann die Nr.6 schon verwendet worden sein. Natürlich darf jetzt kein weiteres Programm den COM6 nutzen!
- J OK, OK, und den Gerätemanager verlassen.

## 6.) ... nun ist es ... fast ... geschafft :-)

- J In XCSoar wird die erste nmea-Schnittstelle auf „COM3“ mit „57.600“ Baud und „Flarm-Treiber“ eingestellt.
- J Die Flarm-Datenquelle selber ist wie folgt einzustellen:
  - o Der den grünen Signal-Pfad speisende Datenstrom sendet (logischer Weise auch) auf
    - J 57.600Baud
    - J ... konfiguriert mit GPS & Verkehrsdaten
  - J Der den gelben Signal-Pfad speisenden Datenstrom vom Flarm arbeitet auf
    - J 4.800 Baud
    - J ... und dort nur Verkehrsdaten
    - J Hinweis: Der „gelbe“ aus dem PC stammende Datenstrom arbeitet Hardware-bedingt auf 19.200 Baud, dies ist für den Nutzer aber ohne jede Bedeutung.
- J Die weitere(n) Schnittstellen in XCSoar sind optional entsprechend der ggf. zusätzlich angeschlossenen Datenquelle(n) zu konfigurieren.
- J Das externe Display (V2/3/4er LED-Display) muss bei Nutzung des **adsbMuxer**'s auf 4.800 Baud eingestellt werden. Warum Festlegung auf 4.800? Das neue Flarm-Core kann zwar höher, aber die alte Zigarettenschachtel über den Extension-Port leider nicht. Damit nicht zwei **adsbMuxer**-Firmware-Versionen nötig sind, wird die Baudrate gewählt, die beide Flarm-Geräte können, also 4.800 Baud. Die geringe Baudrate ist auf den LED-Displays wegen der nur sekundlichen Aktualisierung kein Nachteil.

## 7.) Fertig, soll heißen: Die meisten sind (fix und ?) fertig :-)))

Hier die grundsätzliche Reihenfolge der zu öffnenden Programme, die auch durch eine .bat-Datei automatisch mit einigen Sekunden Sicherheitsabstand nacheinander gestartet werden können:

Nach dem Hochfahren des Rechners ...

- J adsb#.exe + "Start"-Button
  - o erste Frames laufen ein?
- J VSPE.exe (mit der Datei „c3.vspe“, siehe VSPE-Hilfe, minimierend starten!)
  - o ... Staus „Ready“?
- J adsbVektor.exe
- J XCSoar.exe
- J ... ggf. noch Triebwerks/Flugzeitlogger.exe

... starten. Im einfachsten Falle zu ersten Tests geht das per Doppelklick. Später dann vollautomatisiert mit einer Batch-Datei in zeitlichem Abstand.

#### WICHTIG:

Damit serielle Daten vom Flarm (oder der GPS-Quelle) für jegliche Navigation und Verkehrsinformationen auf der XCSoar-Moving-Map dargestellt werden können, muss VSPE und **adsbVektor** noch VOR XCSoar gestartet werden (siehe Signalfloss in schematischem Schaltbild-1), und zwar in genau dieser Reihenfolge, jeweils mit einem kleinen zeitlichem Sicherheitsabstand!

Das Programm adsb# muss nicht unbedingt ausgeführt (oder gestartet) werden, nur dann gibt es auch keine ADSB-Verkehrsdaten.

Das Flarm bzw. die GPS-Quelle muss unbedingt eingeschaltet sein und gültige GPS-Daten liefern, damit **adsbVektor** ADSB-Vektoren berechnen und ausgegeben kann. Ohne ein eingeschaltetes Flarm (oder GPS-Daten von anderer Quelle, die mindestens \$GPGGA und \$GPRMC liefert) werden keine ADSB-Ziele ausgegeben, es sei denn es wird für eine feste Bodenstation der Standort in **adsbVektor** eingegeben.

Man kann Windows noch dazu bewegen, dass alles obige automatisch und unauffällig nacheinander startet, sowie die letzt endlich gewünschte Oberfläche auf dem Bildschirm stehen bleibt. Mit den unzähligen Feinheiten, ob man nun die original ADSB-Kennung oder pauschal ADS-B auf der Moving-Map sehen möchte ... kann man sich so manch verregneten Tag beschäftigen.

Es kann weitere Abende dauern ... bis man alles so richtig perfekt für sich persönlich eingestellt und konfiguriert hat. Und sei es die immer noch vergessene (aktivierte!) Energiespar-Option: Diese bewirkt urplötzlich einen blanken IPX-Bildschirm und höheren Puls ;-)

Sobald die Modalitäten geklärt sind, wird ein fix und fertiges Festplatten-Backup zur Verfügung gestellt. Das bedeutet, man muss nach dem Einschalten nur noch die aller-notwendigsten Einstellungen (wie die Polare) vornehmen.

#### **Es klappt etwas überhaupt nicht, auch nicht beim fünften Versuch am übernächsten Tag?**

Auf jede Frage zum alternativen ADSB-Konzept, die per Email an [ipx@schrandt-electronic.de](mailto:ipx@schrandt-electronic.de) verständlich und ausreichend formuliert gestellt wird, wird so bald wie möglich geantwortet, bei Notwendigkeit sofort als Aktualisierung dieses Dokuments.

Was ganz deutlich NICHT (mehr!) unterstützt wird, ist die Dokumentations-Politik der Firma Flarm:

Ein eindeutig kommerziell orientiertes Unternehmen ist für eine Endnutzer-verständliche und anwendbare Dokumentation ihre Produkte selber verantwortlich.

Ein vielleicht lästiger und kosten-intensiver Support darf nicht denen aufgehast werden, die nicht an deren Geschäft beteiligt sind.

Es werden von dieser Seite also keine Informationen oder Auskünfte gegeben, wie man z.B. nun genau eine Flarm-Schnittstelle zu konfigurierten hat. Die Flarm-Dokumentationen sind für einen „Fachmann“ ausreichend, für so manchen Endbenutzer oder jemanden mit dem Einbau beauftragten allerdings nicht. Gerade dann, wenn der Einbau irgendwie über die Minimal-Installation hinaus geht. Bitte wendet Euch an den Flarm-Support, sonst wird deren Dokumentation nie verständlicher.

Auf Fragen, die die Integration der vorgestellten ADSB-Lösung in einem Android- oder Linux-System betreffen, kann leider keine helfende Auskunft werden, da für diese Betriebssysteme wahrscheinlich zu wenig Erfahrung vorliegt.

Wer es erfolgreich geschafft hat das per DVBT-Stick eingespeiste ADSB mittels **adsbVektor** zu integrieren, darf als kleinen Dank für diese Anleitung gerne die eventuell noch vorhandenen Stolpersteine mitteilen! Die nächsten „Installateure“ haben es dann entsprechend leichter :-)

Dieses Dokument wird ... je nach Notwendigkeit und der zukünftig gestellten Fragen ... laufend aktualisiert. Der aktuelle Stand ist in der Fußzeile dieses Dokumentes ersichtlich.