wellhausen & marquardt Mediengesellschaft

Der folgende Bericht ist in Ausgabe 10+11 – Oktober/November 2025 des Magazins FlugModell erschienen.

www.flugmodell-magazin.de

RC-UMSTIEG VON FUTABA AUF POWERBOX

Systemwechsel

TEXT UND FOTOS: *Rainer Strobel*

Nach mehr als drei Jahrzehnten Modellflug mit Futaba-Anlagen stand bei FlugModell-Autor Rainer Strobel die Anschaffung eines neuen Fernsteuersystems an. Aber wie hoch ist der Aufwand bei der Umstellung der bislang mit Futaba betriebenen Modelle auf das neue RC-System Core von PowerBox-Systems und welche neuen Möglichkeiten bringt das mit sich? Zwei konkrete Modell-Beispiele verdeutlichen es hier.

Eine neue Anlage sollte her, nicht weil es Probleme mit meiner sehr zuverlässigen T18MZ gab, sondern weil die meisten heutigen Fernsteuersysteme in Bezug auf Programmiermöglichkeiten und Telemetrieausgabe deutlich mehr zu bieten haben. Leider hat Futaba aus meiner Sicht in den letzten Jahren den Anschluss an die Entwicklungen der Konkurrenz verloren, sodass ein Herstellerwechsel anstand.

Beim Vergleich der aktuell verbreiteten Systeme war mir neben der Zuverlässigkeit der Funkstrecke vor allem eine möglichst hohe Kompatibilität zu den in meinen Modellen verbauten elektronischen Komponenten wichtig. Da ich ausschließlich Akkuweichen und die dazu passenden Peripheriekomponenten von PowerBox-Systems verwende, war die Anschaffung einer Core oder Atom naheliegend. Ich habe mich auch mit Sendern anderer Hersteller beschäftigt, aber nicht zuletzt, weil der Aufbau der Modellprogrammierung am besten zu meinen persönlichen Gewohnheiten (möglichst wenig vorgegebene Modelltypen, Mischer und Funktionen) passte, fiel die Entscheidung schließlich auf die Core.

Im Folgenden möchte ich über meine Erfahrungen beim Umstieg von Futaba auf PowerBox berichten und werde die Umrüstung zweier sehr unterschiedlicher Modelle beschreiben. Ich starte mit meiner im letzten Jahr fertiggestellten Mirage, über deren Bau ich in **FlugModell** 10+11/2024 bis 1+2/2025 berichtet habe, und gehe dann auf die Umrüstung in meiner Hawker SeaFury ein, deren Erstflug bereits 13 Jahre zurückliegt und die unverändert mit den ursprünglichen Komponenten ausgerüstet ist.

Der Umstieg vom alten Futaba-RC-System erfolgt auf die moderne Core von PowerBox-Systems

Umrüstung in der Mirage

Der Jet ist mit einer Mercury SR2 ausgerüstet, an deren FastTrack-Port das GPSIII und der iGyro-Sat angeschlossen sind – alles Komponenten von PowerBox. Bislang erfolgte die Kommunikation mit den zwei Futaba-Empfängern R7003SB über das S-BUS2-Protokoll, das neben den Steuersignalen auch die Telemetriedaten der Weiche und deren Peripherie übertrug. An einem der Empfänger war zusätzlich ein ECU-Konverter von VSpeak über ein V-Kabel am SBUS2-Port angeschlossen. In der T18MZ standen somit die Daten der Turbine, der Akkuweiche sowie die von den Empfängern selbst gelieferten Werte der Eingangsspannung und der Signalstärke des Rückkanals zur Verfügung. Die insgesamt 27 Telemetriedaten waren auf drei Display-Seiten des Senders verteilt und deren praktische Nutzung entsprechend umständlich. Während des Startvorgangs der Turbine habe ich Abgastemperatur, Drehzahl, Pumpenspannung und Turbinen-Status beobachtet, im Flug konnte ich die Restkraftstoffmenge per Taster ansagen lassen und die Versorgungsspannung der Empfänger war mit einem Alarm bei Mindestwertunterschreitung versehen. Mit der Umstellung auf die Core würden deutlich mehr Telemetrieinformationen bereitgestellt und dennoch erwartete ich eine wesentlich übersichtlichere Darstellung für mich relevanter Werte im Display des Senders.

Aber zunächst galt es, die Mirage im neuen Sender Core als Modell anzulegen und alle Steuerfunktionen zu definieren. Ich habe erst gar nicht nach möglicherweise im Netz verfügbaren Tools gesucht, die Modelldateien von der T18MZ in ein von der Core lesbares Format konvertieren können. Meiner Erfahrung nach funktioniert so etwas selbst zwischen den verschiedenen Sendertypen desselben Herstellers meist nicht zufriedenstellend. Entsprechend würde ich dem Ergebnis einfach nicht vertrauen und wollte deshalb lieber alles neu anlegen. Da ich ohnehin für jedes meiner Modelle eine Excel-Liste mit Funktionen, Geberzuordnungen, Kanalbelegungen, Servolaufrichtungen und mehr angelegt habe, war diese die beste Grundlage für die Übertragung der Mirage in die Core. Und zur Kontrolle gab es ja auch noch die Menü-Displays der Futaba.

Umsetzung

Abzubilden waren ein Deltaflügel mit vier Rudern, das Seitenruder, das lenkbare Bugrad mit definierter Servoposition im eingefahrenen Zustand, die Gasfunktion mit Turbinenabschaltung über einen Schalter, das Einziehfahrwerk mit Klappensteuerung, die Radbremse mit Feststellfunktion, die Nachbrenner- und Lichtsteuerung mit Abschaltung der Landescheinwerfer bei eingefahrenem Fahrwerk, die Empfindlichkeitsverstellung des Kreisels, die abschaltbare Ansteuerung des Starthilfevektors sowie ein Reset der Akku-Kapazitätsanzeige und schließlich die Abfrage der Restkraftstoffmenge per Taster. Abbildungen 1 und 2 zeigen, wie das Ganze im Funktionsmenü der T18MZ aussah. Die Delta-Mischung wurde in der Futaba durch Auswahl eines entsprechenden Modelltyps generiert, bei dem die Kanäle 1 und 6 mit Quer und Kanal 16 automatisch mit Höhe belegt wurden.

In der Core kann man zwar auch einen Deltaflügel als Modelltyp vorgeben, aber genauso schnell geht das auch durch manuelle Definition der Funktionen. Abbildungen 3 bis 5 zeigen die in der Core angelegten Funktionen. Die Spalte „Servo“ entspricht genau genommen dem Ausgangskanal des Empfängers und ist somit vergleichbar mit der Spalte „KA“ bei der T18MZ. Und hier liegt ein wesentlicher Unterschied zwischen

den beiden Anlagen. Die Core erlaubt die Zuordnung desselben Servos (Kanals) zu mehreren Funktionen, was man beispielsweise bei Servo 1 und 6 für Quer und Höhe oder bei Servo 11 für Bremse und Handbremse sieht. Bei Futaba kann ein Kanal nur einer Funktion und damit nur einem Geber zugeordnet werden. Soll er eine weitere Funktion erfüllen, muss ein Mischer verwendet werden. So wurden bei der Mirage in der Futaba Kanal 11 und 13 gemischt, um die Handbremsfunktion abzubilden, bei der Core hingegen einfach das Servo 11 der Funktion „Bremse“ mit Geber ST-A (Höhe) und der Funktion „Handbremse“ mit Geber SW-A (Schalter) zugeordnet. Durch Nutzung dieser Möglichkeit sind in der Core bei meiner Mirage sämtliche Mischer entfallen, was das Modellprogramm deutlich „aufgeräumter“ macht.

Als Nächstes galt es, die Servomitten, -wege und -laufrichtungen einzustellen. Während dazu bei Futaba drei getrennte Menüs (siehe Abbildungen 6 bis 8) aufgerufen werden müssen, werden diese Einstellungen in der Core in je einem Menü pro Servo erledigt. Ein weiterer Nutzen der oben beschriebenen Mehrfachzuordnung von Servos zu den Funktionen wird nun deutlich. In Abbildung 9 ist Servo 1 für die Funktion „Quer“ mit +-25 % Weg, in Abbildung 10 dasselbe Servo für die Funktion „Höhe“ mit +-75 % Weg und invertierter Laufrichtung zu sehen. Damit wird der geringere Ruderweg für Quer im Vergleich zu Höhe übersichtlich direkt in den Servoeinstellungen möglich, während dies bei Futaba separat in den Gebereinstellungen erfolgen muss. Die geänderte Laufrichtung zwischen Quer und Höhe ersetzt die Delta-Mischfunktion. Bei den Servomitten und -wegen ist zu beachten, dass die einzustellenden Prozentwerte bei Futaba und PowerBox unterschiedlich sind. Dies liegt daran, dass die Impulslängen und deren Variationsbreiten (Impulsweiten) nicht übereinstimmen. Die Tabelle verdeutlicht die Unterschiede.

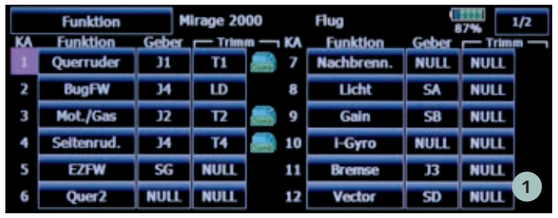
Daraus ergibt sich, dass die Servomitten in der Core um zirka 4 % und die Wege auf zirka 84 % des Futaba-Werts angepasst werden müssen. Mit dieser Regel habe ich alle Servomitten und -wege in wenigen Minuten eingestellt und anschließend ergaben sich bei der Messung der effektiven Ruderausschläge am Modell nur minimale Abweichungen zu den ursprünglichen Werten. Der letzte Schritt bei den Rudereinstellungen war nun noch die Einstellung der Expo-Kurven. Abbildungen 11 sowie 12 zeigen die Kurven der T18MZ und die Abbildungen 13 sowie 14 die entsprechenden Kurven der Core. Man sieht, dass die Expo-Prozentwerte ohne Anpassung übernommen werden können.

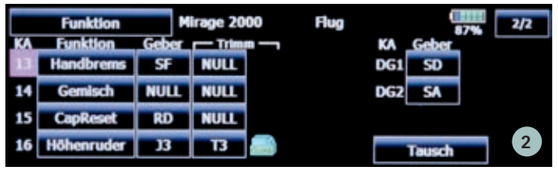
Ohne Mischer

Damit waren die Funktionen definiert und alle Rudereinstellungen vorgenommen. Es blieben nun noch die Sonderfunktionen, wie Turbinenabschaltung, Abschaltung der Landescheinwerfer und Bugradlenkung bei eingefahrenem Fahrwerk und die Deaktivierung der Vektorlippe. Diese vier Aufgaben lassen sich in der Core sehr elegant mit dem Feature „Servo Cut-Off“ lösen (Abbildung 15). Damit lässt sich für das gewählte Servo eine feste Position vorgeben, die es nach Betätigen des hierfür

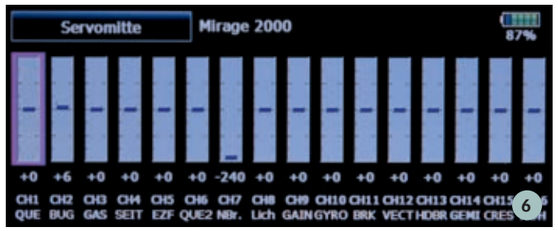
Unterschiede Impulsweiten

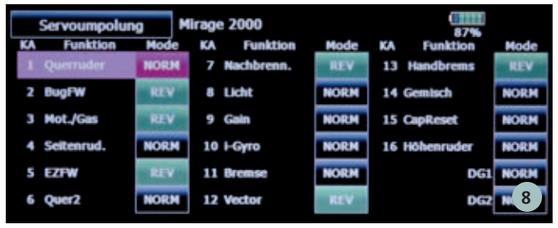
	-100 %	0 %	+100 %
Futaba	1100 µs	1520 µs	1940 µs
PowerBox	1000 µs	1500 µs	2000 µs

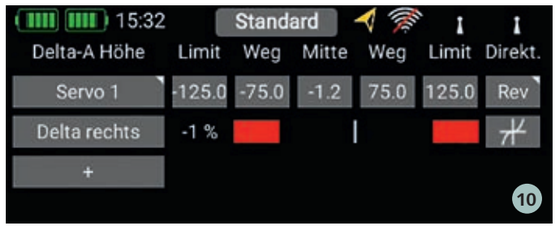


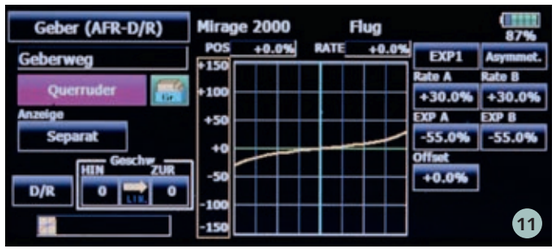


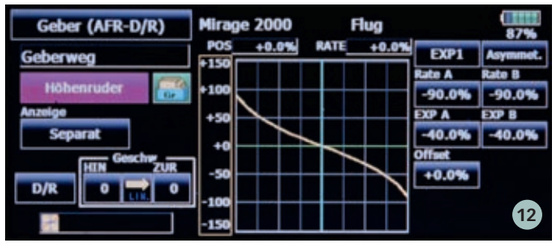
In der Futaba kann jedem Empfängerausgang (Kanal) nur genau eine Funktion zugeordnet werden (Abbildungen 1 und 2), in der Core hingegen ist es möglich, jeden Empfängerausgang (Servo) mehreren Funktionen, und damit Gebern, zuzuordnen. So wurden beispielsweise die Ausgänge 1 und 6 je einmal der Funktion Quer und der Funktion Höhe zugeordnet (Abbildung 3). Das Gleiche gilt für das Bremsventil an Ausgang 11 (Abbildung 4), das damit sowohl proportional über Tiefenruder als auch zu 100 % mit einem Schalter für die „Handbremsfunktion“ angesteuert wird – alles ohne Mischer und dadurch mit weniger Kanälen als bei der T18MZ (Abbildung 5)

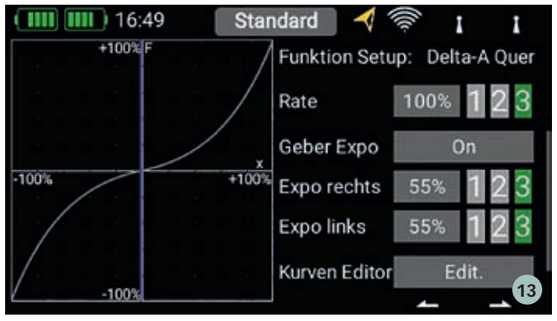


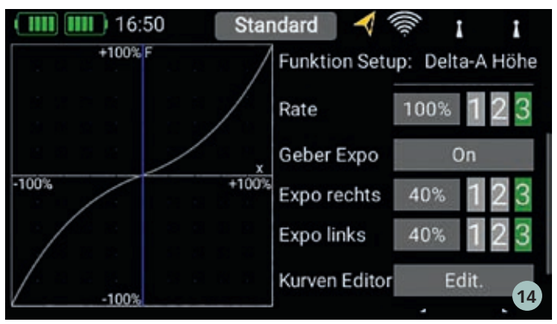










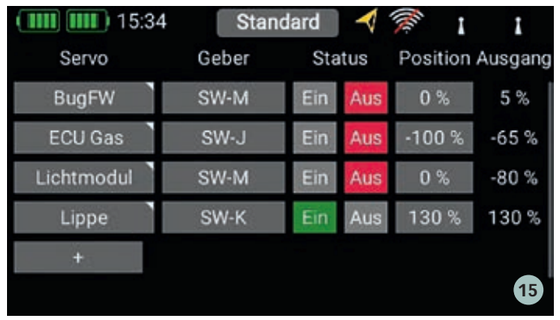


Servomitten, -endpunkte und -laufrichtungen erfordern in der Core andere Einstellungen als bei Futaba, da die Impulslängen und deren Variationsbreite unterschiedlich sind. Während die jeweiligen Werte bei Futaba für jeden Kanal fest eingestellt werden (Abbildungen 6 bis 8), sind bei der Core in jeder Funktion unterschiedliche Werte hierfür möglich. Das wurde genutzt, um für Quer niedrigere Ruderausschläge als für Höhe einzustellen (Abbildungen 9 und 10), ohne die Geberwege zu ändern (Abbildungen 11 bis 14)

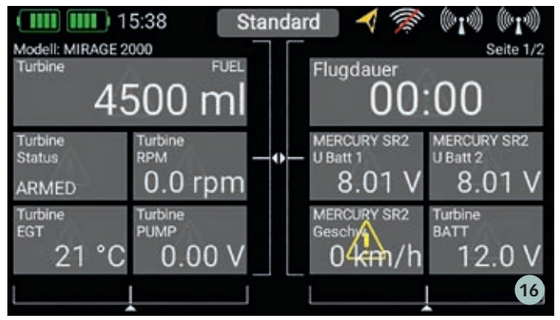
definierten Gebers einnimmt. Und das unabhängig von den Vorgaben aus anderen Funktionen. So habe ich beispielsweise den Weg in der Funktion „Gas“ zwischen -65 % und +100 % eingestellt, das über einen Sicherheitsschalter auszulösende Servo Cut-Off auf -100 %. Die Turbine kann auf diese Weise bei jeder beliebigen Stellung des Gasknüppels (not-)ausgeschaltet werden. Die Position „Einfahren“ des Fahrwerksschalters bewirkt mit dem Servo Cut-Off die feste Neutralposition des Lenkservos für das Bugfahrwerk sowie das Abschalten der Landescheinwerfer, und schließlich wird die beim Start mit dem Höhenruder gekoppelte Vektorlippe mit einem Schalter aus dem Abgasstrahl gefahren und deaktiviert – alles wie gesagt ohne Mischer. Und auch ohne Flugphasen. „Keep it simple“ at it’s best.

Da ich alle weiteren Einstellungen, wie den Door-Sequencer und die Feineinstellung der Servopositionen (Matching) in der Mercury vorgenommen hatte, war die Programmierung der Mirage in der Core abgeschlossen. Statt der zwei R7300SB musste ich nur die PowerBox-Empfänger (zwei PBR26D) ins Modell einbauen, den ECU-Konverter auf den P²Bus umstellen und in der ECU sicherheitshalber den Gasknüppel neu einlernen. In der Mercury waren keine Änderungen erforderlich, da diese das angeschlossene RX-System automatisch erkennt. Nach nur zwei Nachmittagen war damit die Umstellung meines ersten Modells bis auf die Konfiguration der Alarmgrenzwerte

Nach erfolgreichem Wechsel von Futaba auf Core in der Mirage wurde als zweites Modell die SeaFury umgerüstet



Ein sehr nützliches Feature der Core ist das Servo Cut-Off. Jeder Empfängerausgang (Servo) kann damit in eine vordefinierte Position gebracht werden (Abbildung 15)



Die wichtigsten Telemetriewerte und der Timer passen auf nur eine Displayseite der Core (Abbildung 16)

und der Telemetrieanzeigen im Display des Senders erledigt. Letzteres lässt sich nach persönlichen Vorlieben völlig frei durch das Erstellen und Anordnen von Widgets einrichten – ganz wie bei einem Smartphone (Abbildung 16). Über diese Widgets gelangt man dann auch in die entsprechenden Untermenüs, in denen Alarme und mittels Text-to-Speech-Funktion frei vorgebbare Ansagen konfiguriert werden können. Das gab es in der Form bei der T18MZ nicht, weshalb ich einen weiteren Nachmittag investiert habe, um alles in Ruhe auszuprobieren und nach meinen Wünschen einzustellen.

Testflüge Mirage

Auf dem Platz habe ich zunächst den obligatorischen Reichweitentest durchgeführt, wie erwartet ohne Probleme. Diese gab es auch nicht bei den folgenden Testflügen. Die Ruderausschläge erwiesen sich als absolut passend, ein Nachtrimmen war nicht erforderlich und die Mirage flog wie gewohnt ausgewogen in allen Geschwindigkeitsbereichen; die Kreisel- und Expo-Einstellungen passten also auch. Nun war ich sehr gespannt auf die neu programmierte Ansage der Fluggeschwindigkeit auf Basis der GPS-Werte. In Landekonfiguration sollte jeweils eine Ansage erfolgen, wenn sich die Geschwindigkeit um mehr als 10 km/h ändert. Das geschah zwar, aber mit deutlichem Zeitversatz und war deshalb als Unterstützung beim Landeanflug nicht praktisch nutzbar. Bei der späteren Auswertung der LOG-Dateien sah es aber so aus, als ob die GPS-Daten zeitlich doch weitgehend zu den „Echtzeitdaten“, wie beispielsweise den Knüppelstellungen, passen würden. Offenbar kam also nur die Ansage verspätet, was durch einen Blick eines Helfers auf das Display, während eines der folgenden Flüge, bestätigt werden konnte. Vielleicht lag es ja daran, dass ich bei der Sprachansage „Wert und Einheit“ eingestellt hatte und Ansagen wie „Einhundertsiebenundzwanzig Kilometer pro Stunde“ einfach zu viel Zeit in Anspruch nehmen. Leider kann ich das aktuell in der Mirage nicht testen, da ein undichter Pneumatikzylinder weitere Flüge verhindert – auf das Thema selbst komme ich gleich nochmals zurück. Dennoch ist mein Fazit zur Umrüstung meines ersten Modells äußerst positiv. Schließlich war der Zeitaufwand sehr überschaubar und es hat technisch auf Anhieb alles prima funktioniert. Nun war ich sehr motiviert, das nächste Modell umzubauen – nur hierin waren deutlich ältere RC-Komponenten verbaut und ich war gespannt, wie gut die Umstellung gelingen würde.

Pläne für die SeaFury

Der Erstflug meiner SeaFury aus dem Hause Airworld war im April 2012. Bis heute bereitet mir dieser Warbird aufgrund der guten Flugeigenschaften und vor allem wegen des genialen Sounds des Moki 250 immer noch viel Freude. Ich hatte das Modell beim Bau mit damals aktuellen RC-Komponenten ausgerüstet. So kamen beispielsweise durchweg hochpreisige Futaba-Servos zum Einsatz, die bis heute absolut zuverlässig arbeiten. Das Gleiche gilt für die Cockpit-SRS von PowerBox-Systems, in der ich alle wesentlichen modellspezifischen Einstellungen, einschließlich der des Doorsequencers vorgenommen hatte. Telemetrie hatte ich zu der Zeit noch für überflüssig gehalten und deshalb auch lediglich Futaba-Empfänger mit FAST-Protokoll eingesetzt, die via S-BUS mit der Cockpit verbunden waren. Das sollte sich im Rahmen der Umrüstung auf die Core ändern. Doch zunächst lautete die Frage, ob ich die Weiche überhaupt weiter verwenden könnte und wenn ja, mit welchem Bus-System das möglich wäre. Erfreulicherweise ist die Cockpit-SRS updatefähig und mit der aktuell verfügbaren Firmware steht im TX-Menü auch der P²Bus zur Verfügung, der somit auch die von der Weiche bereitgestellten Telemetriedaten überträgt. Abwärtskompatibilität über mehr als 14 Jahre – toll!

Wie bei der Mirage, habe ich auch in der SeaFury die Futaba-Empfänger durch zwei PBR26D ersetzt und ohne weitere Telemetriesensoren wäre der mechanische Umbau damit auch schon erledigt gewesen. Aber nur die Spannungen der Empfängerakkus und deren Restkapazität war mir dann doch zu wenig. In Gesprächen mit Vereinskollegen kamen gleich sehr viele Empfehlungen zu Sensoren, die ich unbedingt in meiner SeaFury brauchen würde: Für den Moki Temperatur der fünf Zylinder und Drehzahl, Druck im Pneumatiksystem, True Air Speed, Spannung des Zündakkus, um nur die zu nennen, die ich mir merken konnte. Also erstmal kurz innehalten und überlegen, was davon für mich von echtem Nutzen sein könnte.

Die rechtzeitige Erkennung eines undichten Pneumatiksystems erschien mir am wichtigsten. Auch habe ich mir bislang oft die Frage gestellt, ob der Zündakku nach mehreren Flügen noch genug „Saft“ hatte. Diese beiden Werte wollte ich also zukünftig haben. Und nach den beschriebenen Erfahrungen mit der Mirage fand ich auch die Fluggeschwindigkeit interessant, zumal die SeaFury es gefühlt mit dem einen oder anderen Jet am Platz aufnehmen konnte und man das ja mal quantifizieren könnte. Als Scale-Detail hatte ich ohnehin an der Außenseite der Tragfläche ein Pitot-Rohr verbaut, natürlich aus rein optischen Gründen. Dieses würde ich einfach durch ein echtes Staurohr ersetzen und, solange die Mirage gegroundet war, weitere Erfahrungen mit der Echtzeitansage der Core sammeln.

Technische Daten

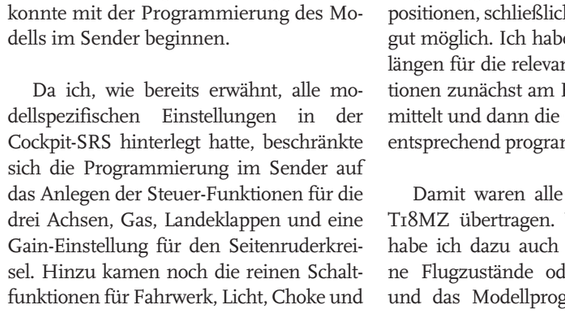
Core von PowerBox-Systems

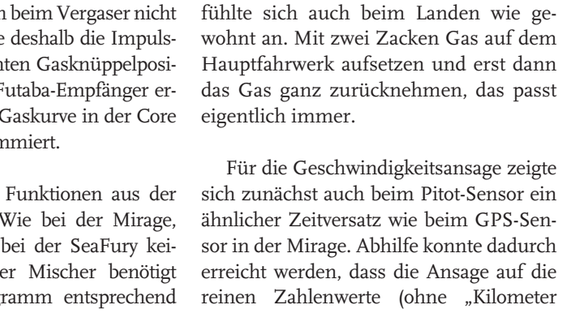
Preis:	ab 2.490,– Euro
Bezug:	Direkt
Internet:	Fachhandel
Kanäle:	26
Sensoren:	bis 250
Display:	TFT, Touchscreen

Konkrete Umsetzung

Die drei benötigten Sensoren wurden bei PowerBox bestellt und im Modell verbaut. Beim Drucksensor PBS-P16 und beim Spannungssensor PBS-T60 war das in wenigen Minuten erledigt. Etwas mehr Aufwand machte der Einbau des

Zur verzögerungsfreien Ansage der Fluggeschwindigkeit empfiehlt sich die Einstellung auf ein festes Intervall und das Weglassen der Einheiten (Abbildung 17). Auch bei der SeaFury finden alle wichtigen Informationen aus dem Modell auf nur einer Displayseite Platz. Druck im Pneumatiksystem, Spannung des Zündakkus und die Fluggeschwindigkeit werden mit separaten Sensoren erfasst, auf der rechten Displayhälfte unter dem Timer einige Daten, die direkt aus der Cockpit-SRS stammen (Abbildung 18)





Zum nachträglichen Einbau des Geschwindigkeitssensors wurde ein Ausschnitt in der Flächenunterseite geschaffen (Abbildung 19). Darstellung der erfassten Werte des Sensors PBS-TAV. Die SeaFury ist wirklich schnell unterwegs (Abbildung 20)

Airspeedsensors PBS-TAV. Hierzu habe ich auf der Unterseite der Tragfläche einen Ausschnitt gemacht, den eigentlichen Differenzdrucksensor dort eingebracht und ein Servokabel zur Wurzelrippe geführt. Zur Halterung des abnehmbaren Staurohrs war nach Entfernung der Scale-Attrappe das entsprechende Aufnahmestück einzuharzen und schließlich eine Befestigung des ausgeschnittenen Deckels zu schaffen. Nur gut, dass ich die alte Dose mit dem Basislack noch nicht entsorgt hatte. Mit etwas Geduld und Verdünnung ließ sich aus dem eingetrockneten Inhalt noch ein wenig pinselfähige Farbe für die Nacharbeit erzeugen. Nach Anpassung der Verkabelung im Rumpf war schließlich hardwareseitig alles erledigt und ich konnte mit der Programmierung des Modells im Sender beginnen.

Da ich, wie bereits erwähnt, alle modellspezifischen Einstellungen in der Cockpit-SRS hinterlegt hatte, beschränkte sich die Programmierung im Sender auf das Anlegen der Steuer-Funktionen für die drei Achsen, Gas, Landeklappen und eine Gain-Einstellung für den Seitenruderkreisel. Hinzu kamen noch die reinen Schaltfunktionen für Fahrwerk, Licht, Choke und die bewegliche Kabinenhaube. Das war im Vergleich zur Mirage deutlich einfacher und mit den geschilderten Erfahrungen zur Übertragung der Servowege von Futaba zu PowerBox recht schnell erledigt. Lediglich der Definition der beim Moki erforderlichen, extrem nichtlinearen Gaskurve habe ich mehr Aufmerksamkeit gewidmet. Dies ist deshalb wichtig, weil erfahrungsgemäß eine saubere Landung der SeaFury nur mit passender Gaskurve im unteren Viertel gelingt. Ein nützliches Hilfsmittel für derartige Feineinstellungen ist die Jeti-Box, mit der die Impulslänge eines Kanals in ms angezeigt werden kann. Das ist noch genauer als die Umrechnung von Prozentwerten und ein optischer Check der Einstellungen ist, anders als bei den Ruderpositionen, schließlich beim Vergaser nicht gut möglich. Ich habe deshalb die Impulslängen für die relevanten Gasknüppelpositionen zunächst am Futaba-Empfänger ermittelt und dann die Gaskurve in der Core entsprechend programmiert.

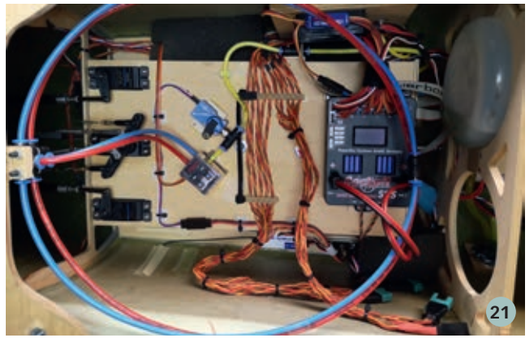
Damit waren alle Funktionen aus der T18MZ übertragen. Wie bei der Mirage, habe ich dazu auch bei der SeaFury keine Flugzustände oder Mischer benötigt und das Modellprogramm entsprechend einfach halten können. Vor dem ersten Testflug blieb somit nur noch die Einrichtung der Telemetriedarstellung, der Alarme, des Timers und der Geschwindigkeitsansage. Letztere sollte nur in der Landekonfiguration aktiv sein. Diese habe ich über einen virtuellen Schalter definiert, der die Ansage auslöst, wenn das Fahrwerk ausgefahren und die Positionslichter eingeschaltet sind.

Testflüge SeaFury

Auch bei der SeaFury verliefen sämtliche Testflüge mit der Core völlig unproblematisch und die Einstellungen der Ruderwege und Expo-Werte passten auf Anhieb. Der Aufwand mit der Gaskurve zahlte sich nun aus, denn das Modell fühlte sich auch beim Landen wie gewohnt an. Mit zwei Zacken Gas auf dem Hauptfahrwerk aufsetzen und erst dann das Gas ganz zurücknehmen, das passt eigentlich immer.

Für die Geschwindigkeitsansage zeigte sich zunächst auch beim Pitot-Sensor ein ähnlicher Zeitversatz wie beim GPS-Sensor in der Mirage. Abhilfe konnte dadurch erreicht werden, dass die Ansage auf die reinen Zahlenwerte (ohne „Kilometer pro Stunde“) und auf ein festes Intervall von zwei Sekunden eingestellt wurde (Abbildung 17). Damit kommt die Ansage nahezu in Echtzeit und ist nun ein nützliches Hilfsmittel beim Landeanflug mit wechselnden Windgeschwindigkeiten. Ebenfalls nützlich ist die Überwachung des Drucks im Pneumatiksystem. Zwar kann diese kein gut abgedichtetes System ersetzen, aber wenn doch einmal eine Leckage während des Flugs auftritt, kann man rechtzeitig reagieren und das Fahrwerk noch sicher ausfahren. Ich habe übrigens lediglich einen Alarm, nicht jedoch das automatische Ausfahren des Fahrwerks bei Unterschreitung eines unteren Druck-Grenzwerts programmiert, weil solch eine Aktion bei Vollgas mit Sicherheit zum Verlust der Restabdeckungen führen würde.

Die Spannungsanzeige für den Zündakku hat ergeben, dass dieser eigentlich überdimensioniert ist. Ich weiß also nun, dass ich mir diesbezüglich keine Gedanken mehr machen muss. Abbildung 18 zeigt das Homedisplay der SeaFury. Alle relevanten Informationen sind auf nur einer Seite dargestellt und es wäre sogar auf der linken Seite noch Platz für zwei weitere Werte. Aber weniger ist ja bekanntlich manchmal auch mehr. Mit einem Doppelklick auf ein Widget können nach jedem Flug die Min- und Max-Werte angezeigt werden, in Abbildung 20 ist dies für den True Air Speed Sensor zu sehen. Die Spitzengeschwindigkeit des Warbirds lag beim zweiten Testflug immerhin bei 254 km/h. Zum Vergleich erreicht meine Mirage mit Centertank und der 190-N-Turbine von Frank laut GPS etwas über 280 km/h. Die SeaFury ist also mit ihrem Kolbenantrieb tatsächlich recht schnell unterwegs – ganz wie das Original. ■



Die 14 Jahre alte Cockpit-SRS konnte nach einem Update weiter verwendet werden. Links im Bild der Drucksensor zur Überwachung des Pneumatiksystems (Abbildung 21)

Mein Fazit

Die Umstellung der beiden recht unterschiedlichen Modelle von meiner guten alten T18MZ auf die moderne Core war absolut problemlos und mit überschaubarem Zeitaufwand machbar. Meine über viele Jahre mit der Futaba entwickelten Programmiergewohnheiten konnten auch in der Core weitestgehend und intuitiv übernommen werden. Lediglich Features der Core, die es in der Form bei der T18MZ nicht gibt, erfordern etwas Eindenken. Lobenswert ist die per Update erreichbare volle Kompatibilität zwischen der mehr als 14 Jahre alten Cockpit-SRS und der aktuellen P²Bus-Technologie.

Rainer Strobel