



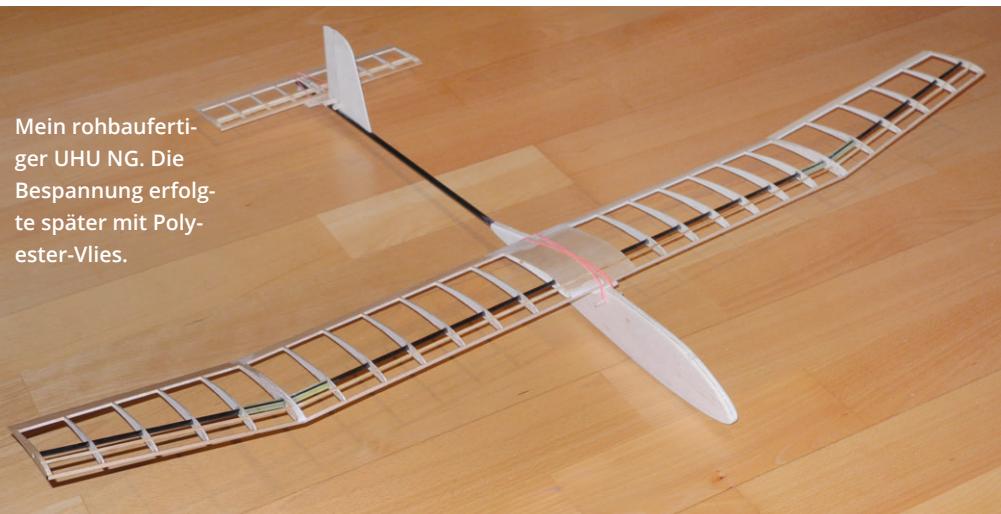
Vor zwei Jahren habe ich einen Freund, der mit seinen Enkeln zum UHU-Wettbewerb in Schorndorf ging, als Helfer begleitet. Am Start waren nicht nur UHUs, sondern auch ausgefeilte Freiflugmodelle der Klasse F1H. Am meisten

# BACK

erstaunt war ich darüber, dass nicht nur Jugendliche, sondern auch einige Leute gesetzteren Alters mit von der Partie waren. Für sie gab es eine getrennte Seniorenwertung. Es herrschte eine sehr lockere Atmosphäre, jeder half jedem, sei es als Starthelfer beim Hochstart oder als Zeitnehmer.

# TO THE ROOTS

## Timer für Freiflugmodelle



### Angefixt

Nachdem ich jahrzehntelang nur noch RC- und seit etwa zehn Jahren auch Fesselflugmodelle geflogen habe, war ich regelrecht begeistert, wie souverän die kleinen Segler völlig autark und geradezu majestätisch durch die Lüfte glitten. Es war nicht nur eine reine Freude, dem zuzuschauen, sondern teilweise auch richtig spannend. Vor allem

dann, wenn es darum ging, ob ein Modell nicht doch noch an einem Baum oder Telegrafenmasten hängen blieb oder in einen Maisacker flog. Ab und zu war es zwar knapp, aber soweit ich mich erinnere, ist es in allen Fällen gut oder zumindest äußerst glimpflich ausgegangen. Ganz besonders hat mich auch beeindruckt, wie selbstverständlich und einmütig Jung und Alt, vereint durch ein gemeinsames Hobby zusammen-

gespielt oder besser gesagt zusammengearbeitet haben.

### UHU NG

Für mich war völlig klar, nächstes Jahr werde ich da mit einem eigenen Modell mitfliegen. Jetzt muss ich allerdings zugeben, dass das Unterfangen vor dem Hintergrund einiger anderer größerer Modellbauprojekte doch wieder etwas in Vergessenheit geriet. Erst zwei Wochen vor dem Wettbewerb ist die Sache bei mir wieder auf dem Radar erschienen. Anfangs dachte ich noch, dass das jetzt einfach zu kurzfristig sei, um noch ein Modell zu bauen. Aber dann habe ich mich doch noch durchgerungen und losgelegt. Und so entstand in kürzester Zeit ein ganz spezielles, komplett selbst gebautes Freiflugmodell, ähnlich eines UHUs. Ganz pur und ohne Thermikbremse oder weitere Funktionen. Mir ging es einzig und allein um den sprichwörtlichen Spaß an der Freude, angefeuert durch den tollen Nachmittag beim vorjährigen UHU-Wettbewerb in Schorndorf.





Lange habe ich hin und her überlegt, wie ich die Kurvensteuerung realisieren könnte und bin schließlich auf die gezeigte Lösung gekommen. Auf dem Wettbewerb konnte ich dann feststellen, dass ich das Rad neu erfunden hatte. Diese Ansteuerung wurde wohl schon lange vor mir auch von anderen Leuten genauso erdacht und genutzt.

## Der Wettbewerb

Den ersten Hochstart habe ich dann erst beim Wettbewerb gemacht. Er war beinahe perfekt. Das einzige Manko war, dass der Hochstarthaken etwas weiter nach hinten gehört hätte. Bei Windstille habe ich es

mit meinen Laufkünsten und 70 Jahren in den Beinen gerade so hingebracht, dass das Modell einigermaßen ordentlich weg stieg. Bei leichtem Gegenwind war die Sache aber ideal. Der UHU ging jedes Mal ohne Ausbrechtdenz oder Tendenz zum Pumpen aus dem Seil.

Gestartet wurde mit einem 20-Meter-Seil. Die Maximal-Zeit war mit 60 Sekunden festgelegt. Diese habe ich mit meinem Modell leider nicht ganz erreicht. Als Ursache hierfür vermute ich das durch den massiven Rumpf doch etwas zu hohe Gesamtgewicht und die verschenkte Flügelfläche in Kombination mit einem zu engen Kurvenflug.

## Freiflug-Fieber

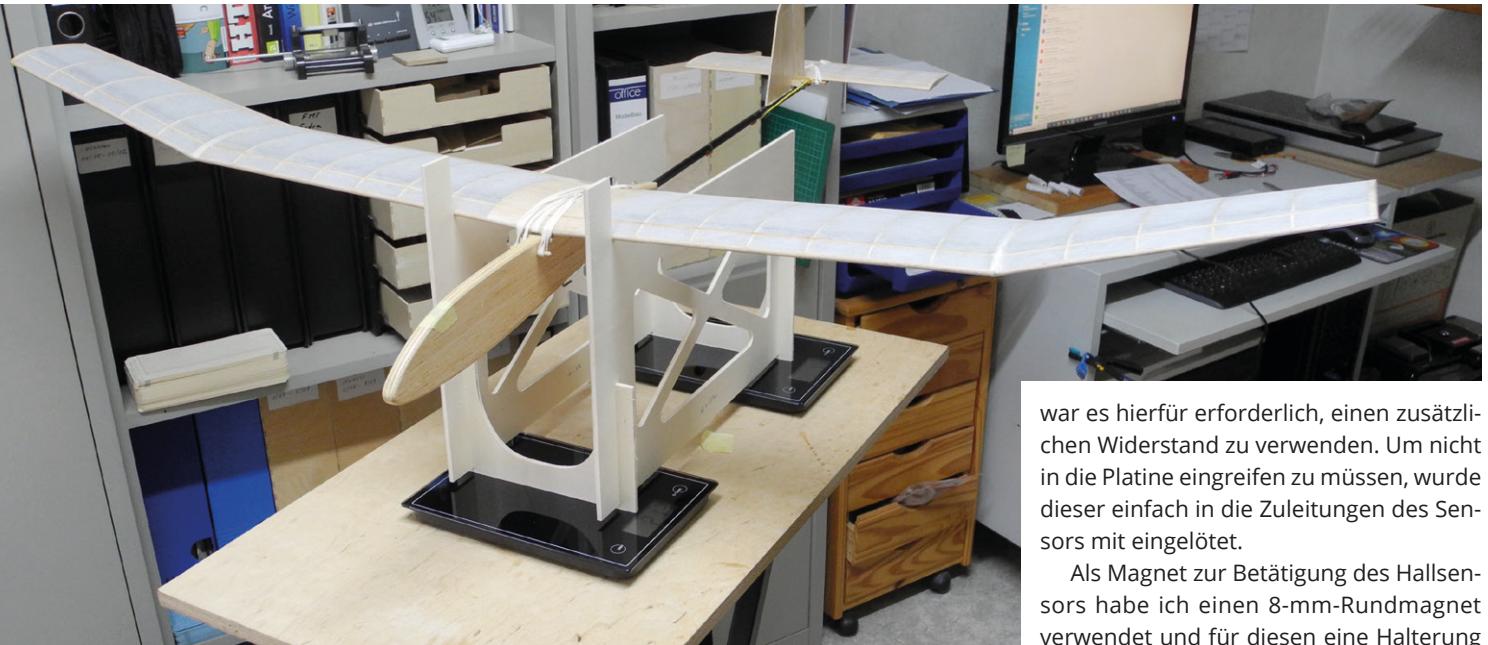
Jetzt hatte mich also das Freiflugfieber gepackt. Und natürlich wollte ich nun den Einbau einer Thermikbremse nachholen. Und damit begann für mich ein ganz anderes Kapitel der Freifliegerei. Da ich gerne auch mal einen Lötkolben und einen Mikroprozessor in die Hand nehme und Letzteren auch gerne programmiere, schwebte mir eine Thermikbremse vor, die über ein Servo ausgelöst wird, welches gleichzeitig auch zur Kurvensteuerung benutzt wird. Einfache, weil in Basic zu programmierende Mikroprozessoren sind die PICAXE Typen. Ein fertig in SMD-Technik bestücktes Board misst gerade einmal  $20 \times 13 \times 3,2$  mm.

## Freiflug-Timer pur

Die erste Variante, als einfacher Freiflug-Timer, war schnell erstellt. Dafür musste das Prozessorboard lediglich mit Stift- und Buchsenleisten versehen werden, um die peripheren Komponenten anschließen zu können. Vorgesehen wurden folgende Anschlüsse:

- ein Servo für Thermikbremse und Kurvensteuerung
- ein Schalter zum Aktivieren der Kurvensteuerung nach dem Hochstart
- eine LED als Signalgeber für die Programmierung
- ein Potenziometer zum Programmieren der Servostellungen für das Auslösen der Bremse und der Kurvensteuerung
- ein Taster zum Programmieren der Laufzeit
- ein Buchsenleiste für den Programm-download

Damit war das Board voll ausgereizt. Der Taster und das Potenziometer zum Programmieren der Servostellungen und der Laufzeit wurden in eine kleine Programmierbox eingebaut. Die Box wird nur zum Programmieren des Timers an diesen an gesteckt. Die eingestellten Werte werden bei jeder Programmierung im nicht flüchtigen Speicher des Mikroprozessors abgelegt und bleiben bis zur erneuten Programmierung erhalten.



Das fertige Modell beim Auswiegen des Schwerpunktes mittels zweier Digitalwagen. Die Berechnung des Schwerpunktes erfolgt über entsprechende Formeln in einem Excelsheet, welches von der vth-Seite heruntergeladen werden kann.

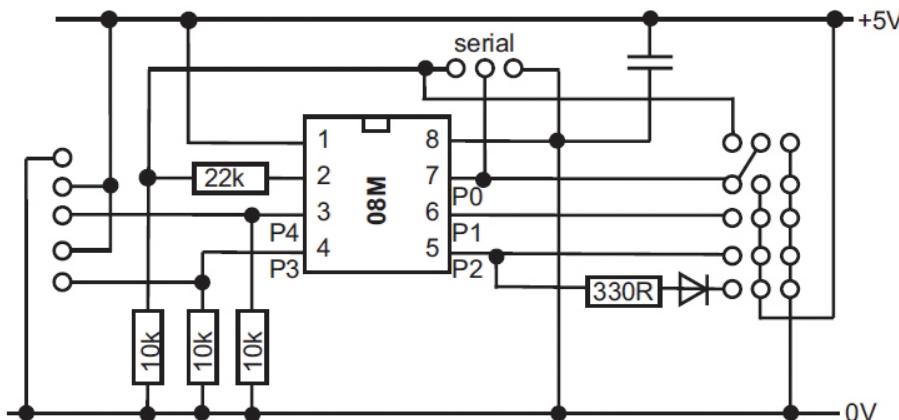


Das für den F1-Timer verwendete, mit SMD bestückte Prozessorboard mit einem PICAXE-08M2-Prozessor. Einmal im Lieferzustand und einmal mit den notwendigen Anschlussleisten.

## Die Kurvensteuerung

Mein Modellbaufreund Markus und ich kamen auf die Idee, einen Hallsensor zu verwenden, der anstelle eines Schalters die Kurvensteuerung auslösen sollte. Dieser wurde auf einem Metallplättchen ins Modell eingeklebt. Von außen kann so ein kleiner Magnet aufgesetzt werden, der über ein Stück Nylonfaden mit dem Hochstartseil verbunden ist. Nach dem Ausklinken des Hochstartseils fällt der Magnet mit dem Hochstartseil ab und der Schaltvorgang für die Kurvensteuerung wird ausgelöst. Leider

- Die Beschaltung des Prozessorboards als Auszug aus dem Datenblatt des Herstellers Revolution Education Ltd. Hier allerdings noch mit dem Prozessortyp 08M statt mit 08M2 gezeigt. Der 08M2 hat genau gleiche Anschlüsse und Funktionen, bietet aber unter anderem deutlich mehr Speicherplatz für Programme.



war es hierfür erforderlich, einen zusätzlichen Widerstand zu verwenden. Um nicht in die Platine eingreifen zu müssen, wurde dieser einfach in die Zuleitungen des Sensors mit eingelötet.

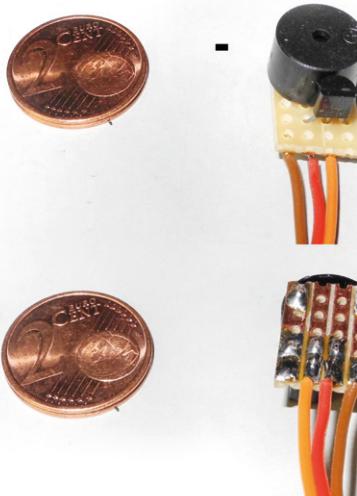
Als Magnet zur Betägigung des Hallsen-sors habe ich einen 8-mm-Rundmagnet verwendet und für diesen eine Halterung gefräst, die mit einem dünnen Nylonfaden mit dem Hochstartring verbunden wurde. Die Halterung wird so angesetzt, dass der Magnet nach außen schaut. Dadurch ist die Haftung nicht so hoch, wie wenn der Mag-net direkt aufgelegt wird. Durch Abschlei-fen der Holzseite kann die Haltekraft sehr feinfühlig eingestellt werden. Sie sollte so groß sein, dass der Magnet sicher hält, aber trotzdem leicht abgezogen werden kann.

Ganz wichtig: Der verwendete Hallse-  
sor ist ein unipolarer. Man muss unbedingt  
vorher ausprobieren, wie herum der Mag-  
net in die Halterung einzukleben ist. Da-  
mit war die erste Version des Timers fer-  
tiggestellt. Zum Einsatz kam er so aber nie.

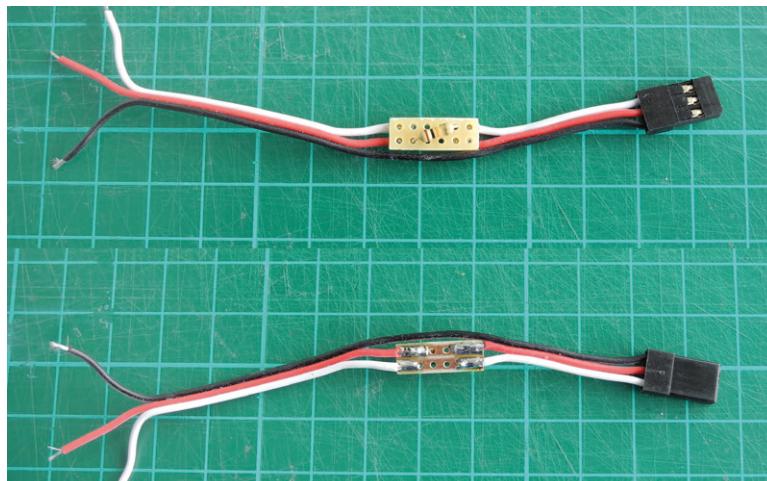
# Freiflug-Timer mit RC-Bremse

Im Laufe der Arbeiten kamen immer mehr Ideen für Erweiterungen auf. Die erste war die, den Timer an einen RC-Empfänger anzuschließen, um die Thermikbremse per Funk auslösen zu können. Damit könnte man dann auch unmittelbar bevorstehende Baumlandungen verhindern. Das sollte doch ohne viel Mehraufwand möglich sein. Dachte ich zumindest. Und im ersten Anlauf war es das auch. Ein beliebiger Empfängerausgang wurde an die Buchse für das Poti angeschlossen. Statt nun die Bremse zeitgesteuert auszulösen, wurde das Empfängersignal abgefragt und ausschließlich zum Auslösen der Bremse genutzt.

Beim Einschalten des Timers wird zuerst geprüft, ob ein Servosignal anliegt. Falls nicht, geht der Timer in den normalen Standardbetrieb. Ist ein Signal vorhanden, wechselt er in den Modus RC-Bremse. Die Funktion der Kurvensteuerung ist in beiden Betriebsmodi gleich.



Der vom Timer als Suchhilfe verwendete Piezo-Summer samt einem NPN-Transistor auf einer kleinen Mini-Streifenplatine.

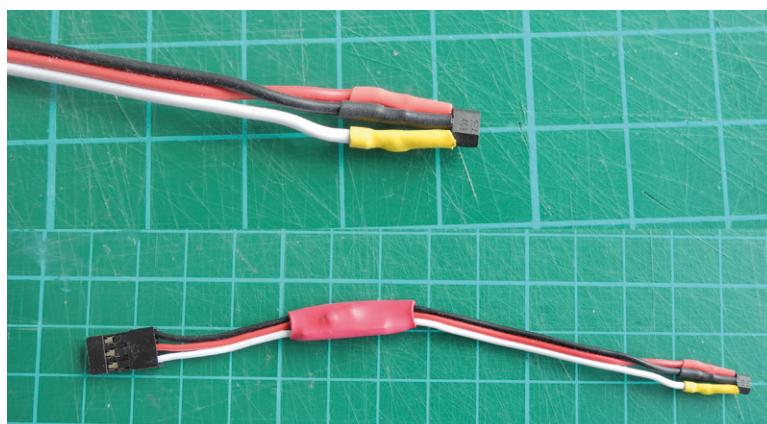


Der verwendete Hall-Sensor 95A 408 für die Aktivierung der Seitensteuerung erfordert einen Widerstand (10 K) zwischen der Signal- und Plus-Leitung.

Der zusätzliche Aufwand zur Ergänzung des Programms beschränkte sich auf einige wenige Befehle. Bei intensiven Tests ist mir dann aber aufgefallen, dass das Servo immer wieder ein wenig zuckte. Zunächst war mir das egal, weil sich die Servostellung nur ganz kurz und kaum sichtbar verändert hat. Leider hat sich das dann aber mit einer weiteren Funktionserweiterung sehr deutlich verschlechtert.

## Freiflug-Timer mit RC-Steuerung

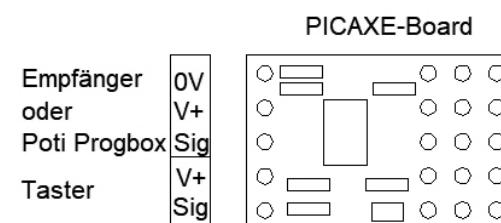
Da der Modellflugplatz unseres Vereins sehr nahe an einer stark befahrenen Landstraße liegt, sind dort keine Hochstarts im Freiflug möglich. Zumal nicht, wenn der Wind in Richtung Straße bläst. Das brachte mich auf die Idee, die ja schon vorhandene Fernsteuerung so zu nutzen, dass man das Modell damit nur mit dem Seitenruder wenigstens ab und zu in die richtige Richtung gängeln könnte. So wären auch in diesem Umfeld Trainingsstarts möglich, bei denen man das Hochstartverhalten optimieren könnte. Ziel war also nicht, den UHU richtig fernzusteuern, sondern ihn nur von der Straße fernzuhalten.



Der obere Teil zeigt die genaue Verdrahtung des Hallsensors, der untere das fertige Sensorkabel.



Meine gefrästen Magnethalter aus Holz sind durch Abschleifen fein in der Haltekraft einzustellen.



Und so werden die einzelnen Komponenten an das mit den Anschlüsse versehene PICAXE-08M2-Board angeschlossen.

des ganzen Chaos war der, dass der Befehl zum Abfragen des vom Empfänger kommenden Servosignals und der zum Stellen des Servos am Timer den gleichen internen Timer des Mikroprozessors benutzen, um die Framerate von 20 ms der Signale zu überwachen und zu generieren. Damit ist es leider nicht möglich, mit diesen sehr einfach zu handhabenden Befehlen gleichzeitig einen Servoausgang des Empfängers abzufragen und das Signal an ein an den Ti-

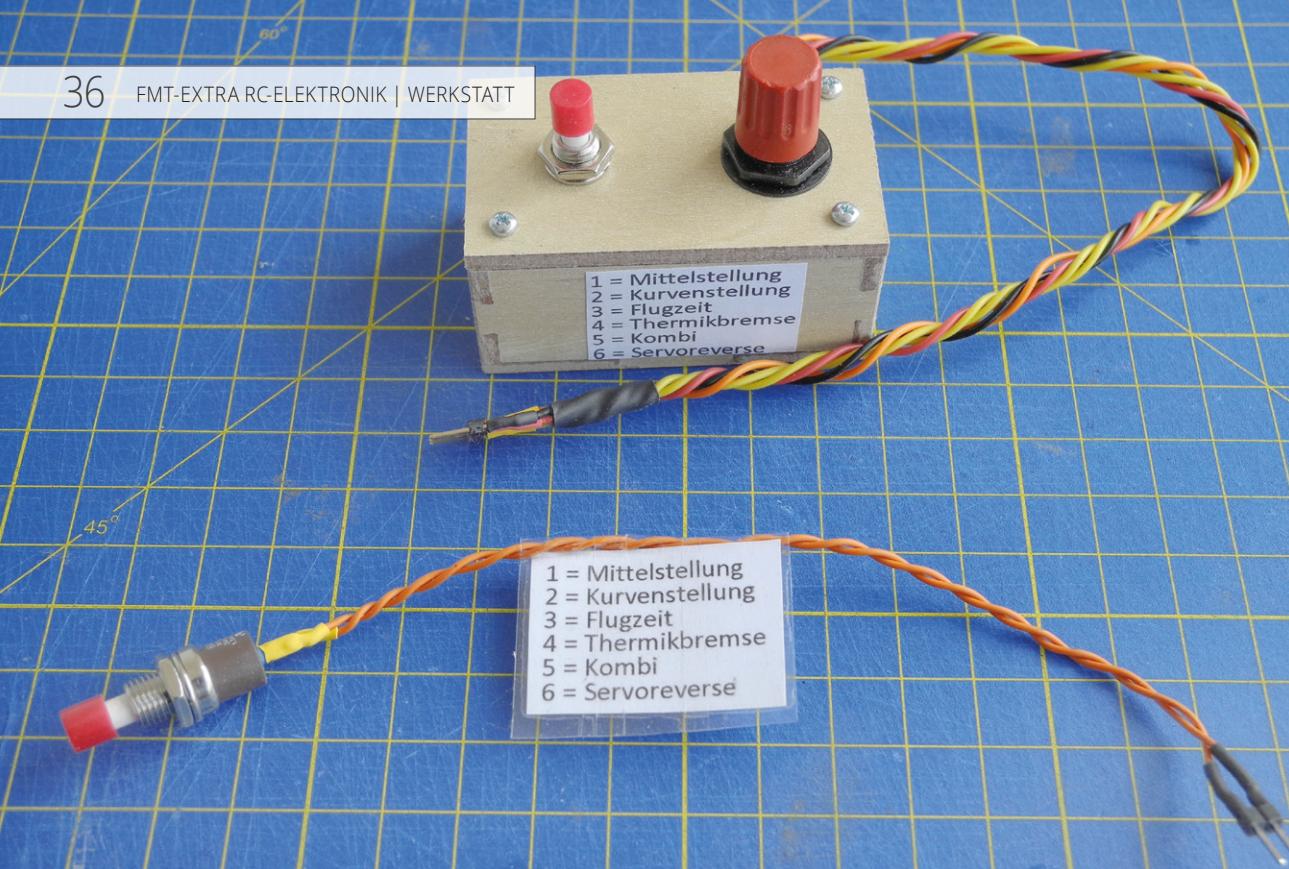
mer angeschlossenes Servo weiterzugeben. Das ist sehr schade und eigentlich ein deutliches Manko dieses kleinsten PICAXE-Typs.

## Die Lösung

Es gab nun drei Optionen. Erstens hätte man einen getrennten Servotreiber-Chip verwenden können. Dies hätte jedoch zu einer deutlich komplexeren Hardware geführt, da ein Zusatzboard erforderlich ge-

## Rückschlag im Projekt

Wieder mussten nur ein paar wenige Befehle ins Programm eingefügt werden, um den neuen Betriebsmodus zu realisieren. Doch damit begann dann eine kleine Katastrophe. Das Servozittern wurde stärker und bei bestimmten Steuerstellungen spielte das Servo regelrecht verrückt. Und das, obwohl ich selbst nach langem Suchen keinen Fehler im Programm erkennen konnte. Nach fast zwei Tagen intensiver Fehlersuche wollte ich schon aufgeben. Dann habe ich mich aber doch noch mal in das Handbuch der PICAXE-Programmierung gestürzt und bin tatsächlich fündig geworden. Grund



Für die Programmierung des Timers wurde eine kleine Programmierbox gefertigt. Wird der Timer mit einer Fernsteuerung betrieben, reicht es, zum Programmieren einen Taster anzuschließen. Die Servostellungen werden dann über die Fernsteuerung vorgegeben.

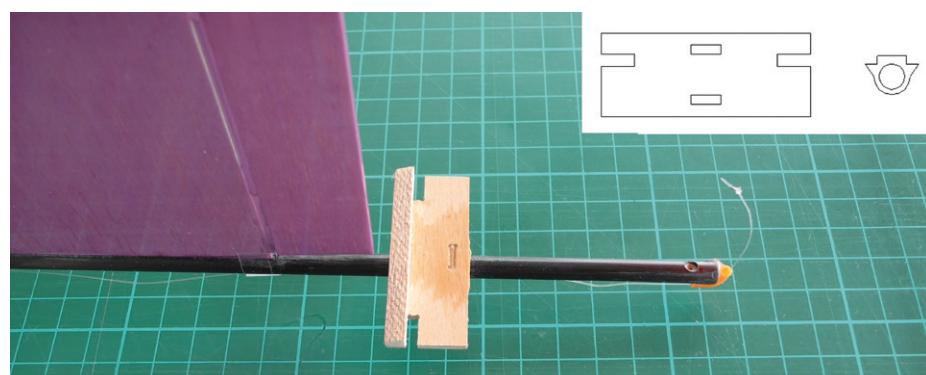
worden wäre. Zweitens hätte ich vom PICAXE etwa auf ein Arduino-Board wechseln können. Damit wäre das Board aber deutlich größer geworden und das Programm hätte völlig neu in der Programmiersprache C/C++ erstellt werden müssen. Die dritte Option war, die Kontrolle der Framerate im Programm selbst zu übernehmen. Damit könnte das kleine PICAXE-Board weiterver-

wendet werden, lediglich die Programmierung würde etwas komplizierter werden.

Letztlich habe ich mich für die dritte Option entschieden, was dann zwei bis drei Tage Programmierarbeit erforderte, bis alles richtig aufeinander abgestimmt war. Lohn der Arbeit war schließlich ein astrein und ohne die geringsten Zuckungen arbeitendes Servo.



Der Auslösemechanismus für die Thermikbremse macht vielleicht einen eher rustikalen Eindruck. Ich wollte aber den Aufwand, alles in den Rumpf zu integrieren, vermeiden. Die Technik funktioniert einwandfrei und sehr zuverlässig ist sie auch.



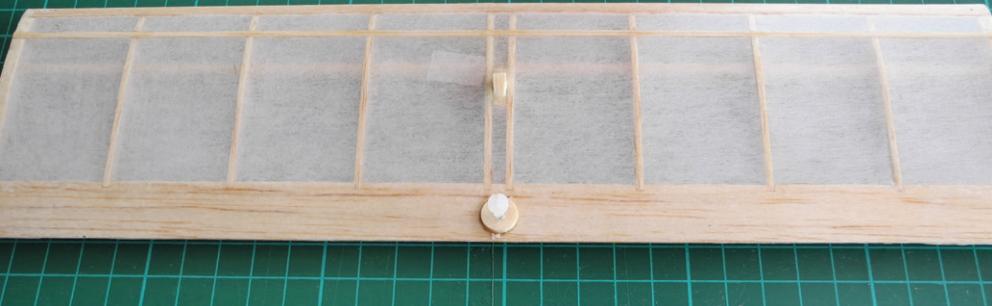
Die Höhenleitwerks-Auflage besteht aus den rechts oben gezeigten Frästeilen und wurde einfach auf den Rumpfstab aufgesteckt und mit Sekundenkleber fixiert. Am Ende des Rumpfstabes wurde ein 3-mm-Loch gebohrt, in das die Einstellschraube des Höhenleitwerks eingreift und das seitliche Verschieben des Höhenleitwerks verhindert.

## Freiflug-Timer mit Suchhilfe

Und kaum war der Timer so weit fertig, kam schon die nächste Idee auf. Man könnte statt der LED auch einen Piezo-Summer anschließen und diesen dann einige Zeit nach dem Auslösen der Thermikbremse aktivieren. Damit wäre das Modell in einem stark bewachsenen Landefeld nach der Landung besser wieder zu finden. Folge war, dass ich verschiedene Summer ausprobiert habe, bis ich einen gefunden hatte, der ein genügend lautes Signal erzeugen konnte. Leider benötigte dieser Summer einen etwas höheren Strom, sodass er nicht einfach nur anstelle der LED an den Ausgangs-Pin des Mikroprozessors angeschlossen werden konnte. Es wurde ein zusätzlicher Transistor erforderlich, den ich über eine Mini-Streifenplatine direkt neben dem Summer platziert habe. Auf diese Weise konnte das Prozessorboard weiterhin unverändert beibehalten werden.

## Die letzte Version des Timers

Da es mittlerweile Winter war und bis zum nächsten Freiflugtag sicher einige Zeit vergehen würde, habe ich noch eine zusätzliche Erweiterung des Timer-Programmes vorgenommen. Ich hatte irgendwo gelesen, dass das Thema Freiflug aufgrund der neuesten Gesetzesvorhaben der EU im Bereich Luftfahrt durch die steuerungsfreien Flugmodelle möglicherweise gefährdet werden könnte. Also dachte ich mir, man könnte das Modell ja bis zur jeweils vorgegebenen Maximalzeit frei und ungesteuert



Die Stellschraube im Höhenleitwerk wurde mit UHU Por schwegängig gemacht, sodass sie sich im Betrieb nicht verstellen kann. Falls notwendig, kann die Schraube mit einem Tropfen Reinigungsbrenzin wieder gängig gemacht und neu eingestellt werden.

fliegen lassen und danach, statt die Thermikbremse auszulösen, die Kontrolle an die Fernsteuerung geben. Damit könnte man anschließend zum Startplatz zurückfliegen und, falls erforderlich, erst hier dann die Thermikbremse auslösen. Und schon gab es einen neuen Betriebsmodus für den Timer. Damit war die Entwicklung zunächst einmal abgeschlossen, sodass die Flugerprobung beginnen konnte.

## Rudimentäres RC-Fliegen

Das Steuern des Modells nur mit Seitenruder ist etwas gewöhnungsbedürftig. Steile Kurven sind natürlich tabu, da man das Absacken nicht mit dem Höhenruder verhindern kann. Beim letzten Flug habe ich dann in etwa 10 m Höhe die Thermikbremse erprobt, die ebenfalls wie erwartet funktioniert und das Modell souverän nur wenige Meter vom Piloten entfernt mitten auf der Piste „abgelegt“ hat.

Die spätere Erprobung im Freiflug erbrachte erwartungsgemäß keine Überraschungen mehr. Fürs Programmieren des Timers wurde einfach der vorgesehene Taster angeschlossen und die mit der Fernsteuerung erflogenen Werte von dieser per entsprechender Tastenbedienung übernommen.

Da es diesen Artikel bei Weitem sprengen würde, nun alle Funktionen des Timers im Detail zu beschreiben, kann die Beschreibung und Bedienungsanleitung des Timers im VTH-Shop unter [shop.vth.de](http://shop.vth.de) kostenlos heruntergeladen werden. Gleicher gilt auch für das benötigte Programm.

## Wie geht's weiter?

Drittes Kapitel in meinen Freiflugaktivitäten: Was mir nicht so richtig zum Thema Freiflug passen will ist, dass ich für das Auslösen der Thermikbremse via RC bisher meinen großen Jeti-Pultsender DC-16 nutzen muss. Und da kommt nun ein ganz besonderes Schmankerl zum Tragen. Die Hardware des Freiflug-Timers kann in genau dieser Form unverändert auch als Fernsteuersender genutzt werden. Es müssen

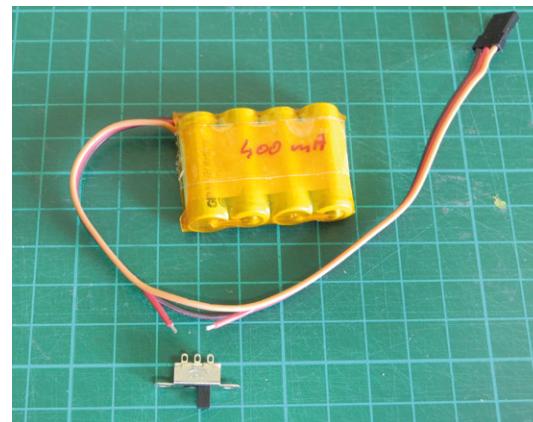
lediglich ein anderes Programm aufgespielt und andere peripheren Komponenten angeschlossen werden. Einen so aufgebauten Mini-Sender, den ich mir zum Fliegen einfach hinters Handgelenk schnalle, habe ich bereits für meine Fesselflugmodelle in Betrieb, um den Motor jederzeit abstellen zu können, falls jemand in den Flugkreis hineinlaufen würde.

## Der Freiflugsender

Für den Freiflugsender werden zusätzlich zum genau gleich bestückten Prozessorboard für den Freiflug-Timer ein Kreuzknüppel, ein einfacher Ein-Aus-Schalter, eine LED und ein handelsübliches HF-Modul benötigt. Da ich eine Jeti-Anlage habe, werde ich natürlich wieder ein HF-Modul von Jeti verwenden. Preislich deutlich günstiger als das spezielle HF-Modul ist der Empfänger R3L, der nicht nur als Empfänger, sondern nach Aufspielen einer anderen Firmware auch als HF-Modul genutzt werden kann.

Er wird in dieser Funktion auch in den auf dem Markt angebotenen, primitiven Jeti-kompatiblen Sendern verwendet. Leider wird dieser Empfänger nicht mehr hergestellt. Man kann ihn aber gelegentlich noch online bekommen. Damit und mit dem PICAXE-Board kann man einen speziellen, sehr kleinen Freiflugsender mit drei Kanälen bauen. Kanal eins und zwei werden über einen einfachen Kreuzknüppel und Kanal drei über einen Schalter bedient. Kanal eins werde ich zur Steuerung des Seitenruders und Kanal zwei zur Auslösung der Thermikbremse nutzen. Letzteres erfolgt tatsächlich nicht über einen zweiten Kanal, sondern dadurch, dass bei Betätigung des Gebers von Kanal zwei eine Zumischung auf Kanal eins so erfolgt, dass die Thermikbremse ausgelöst wird. Die Software, die dieses ermöglicht, habe ich bereits fertiggestellt.

Momentan hat der Eigenbausender drei Modellspeicher, wobei die Anzahl bei Bedarf mit geringstem Aufwand erweitert werden kann. Um den Sender selbst so einfach wie möglich zu halten, werden die Programmspeicher nicht über den Sender, sondern über ein PC-Programm verwaltet.



Statt Bleiballast liegt dieser aus vier Zellen der Größe 2/3 AAA zusammengelöste NiMH-Akku ganz vorne in der Rumpfspitze. Um zwei zusätzliche Steckverbindungen zu vermeiden, wurde der kleine Schalter einfach in die Plusleitung des Akkus eingelötet.

Eingestellt werden können Servo-Wege und -Drehrichtung und einige andere Parameter, die fürs Freifliegen jedoch nicht benötigt werden.

## Resümee

Alles in allem hat es mir sehr viel Spaß gemacht, wieder in eine für mich fast schon vergessene und früher nur rudimentär betriebene, aber doch äußerst spannende Sparte des Modellflugs einzusteigen. Natürlich bin ich mir bewusst, dass man sowohl ein Modell als auch wahrscheinlich deutlich komplexere Elektronik dazu fertig kaufen kann. Wenn auch vielleicht nicht mit exakt den gleichen, aber doch mit den wesentlichen und höchstwahrscheinlich auch mit zusätzlichen Funktionen. Das ist für mich aber völlig uninteressant. Vor allem deshalb, weil es aus meiner Sicht nicht nur um die letzte Stufe, nämlich das Fliegen eines Freiflugmodells, geht. Allein der Weg dahin war und ist für mich auch schon ein sehr spannendes Ziel, bei dem sich einmal mehr die außerordentliche Vielfalt unseres Hobbies gezeigt hat und sicher auch zukünftig immer wieder zeigen wird.

## Bezug

**PICAXE-Board 08M2** [www.picaxeshop.de](http://www.picaxeshop.de)

**Alle weiteren Komponenten** Elektronik Online-Shops

**Datenpaket F1-Timer inkl. DXF-Daten UHU NG** [shop.vth.de](http://shop.vth.de), ArtNr.: 7302

**Excel-Tabelle Schwerpunktberechnung** FMT-CAD-Bibliothek, [www.vth.de/fmt-cad-bibliothek](http://www.vth.de/fmt-cad-bibliothek)