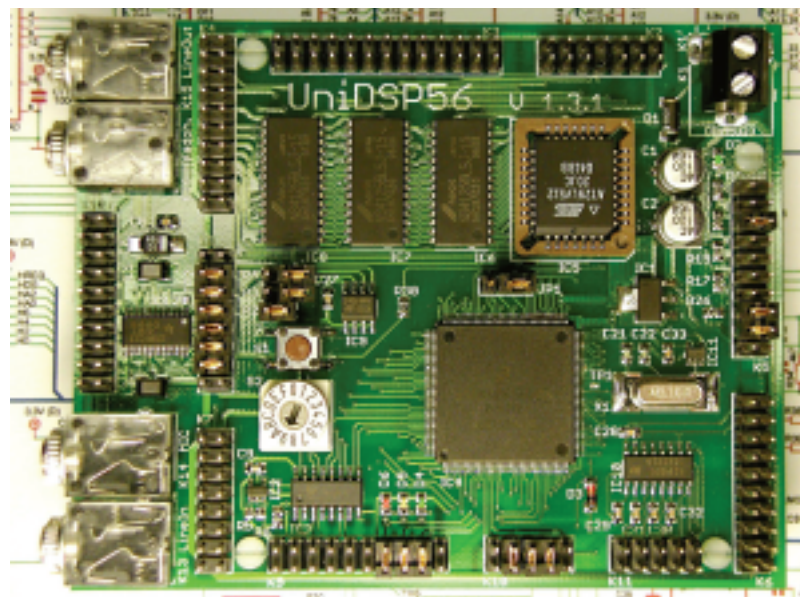


# UniDSP56 – Universelle DSP-Plattform für den Amateurfunk

Wann immer man in Amateurfunkkreisen diskutiert, wie man mit zeitgemäßer Technologie unser Hobby bereichern könnte, kommt man auf digitale Signalverarbeitung. Der Beitrag beschreibt eine DSP-Platine, die vielfältig einsetzbar ist.

Vor einigen Jahren gab es sehr preiswerte DSP Starter Kits (DSK) oder Evaluation Modules (EVM), die als Plattform für eigene Versuche dienten. Die damaligen sind mittlerweile nicht mehr im Angebot, und deren Nachfolger sind nicht nur sehr teuer, sondern werden fast nur noch an kommerzielle Kunden verkauft. Vor diesem Hintergrund reifte der Entschluss, eine möglichst flexible DSP-Plattform für Amateurfunkzwecke zu designen, die folgende Eigenschaften aufweisen sollte:

- Geringer Anschaffungspreis der Hardware,
  - Manuelle Lötbarkeit aller Bauteile (keine BGA-Gehäuse),
  - Einsatz von Bauteilen, die problemlos in kleinen Stückzahlen beschaffbar sind (RS, Reichelt, Conrad & Co.).
- Gerade der letzte Punkt stellt heutzutage die wohl höchste Hürde dar. Nicht nur die Auswahl ist sehr gering, zum Zeitpunkt der Fertigstellung des Designs waren auch schon die ersten eingesetzten Bauteile wieder abgekün-



- Alle wichtigen Funktionsblöcke auf einer Leiterplatte (DSP, Audio-Codex, RAM, Flash, RS-232, ...),
- Ausreichend Ressourcen hinsichtlich Rechenkapazität und Speicher,
- Flexible Erweiterungsmöglichkeiten durch massives Herausführen aller Signale,
- Betrieb auch als Stand-Alone-Gerät (für die, die „nur“ Programme nutzen, aber nicht programmieren wollen),
- Verfügbare Entwicklungswerkzeuge zum „Null-Tarif“ (Assembler, C-Compiler, Debugger, Simulator),

dig, und es mussten neue Quellen aufgetan werden.

## Die Lösung

Herzstück der nun für den Amateurfunk entwickelten Lösung ist der DSP56309 von Motorola bzw. nun Freescale. Es handelt sich dabei um einen 24-Bit-Festkomma-DSP mit 100 MHz Taktrate, der relativ viel internen Programm- und Datenspeicher bietet. Er ist einer der letzten Freescale-DSPs in der aufsteigenden Leistungsskala, den man auch noch im QFP-



## Autor

**Gerrit Buhe, DL9GFA**  
Jahrgang 1971. Studium der Elektrotechnik, Fachrichtung Nachrichtentechnik.

Sechs Jahre in der Entwicklung von Mobilfunk-Basisstationen bei Siemens tätig. Jetzt in Forschung und Entwicklung von Sennheiser zuständig für professionelle Drahtlossysteme. Amateurfunkgenehmigung seit 1985 als Y39FA, ab 1991 DL9GFA. Veröffentlichungen in Amateurfunkmedien zu Software-Defined-Radio.

Gerrit Buhe, DL9GFA  
Eichenkamp 4  
30900 Wedemark  
dl9gfa@darcl.de

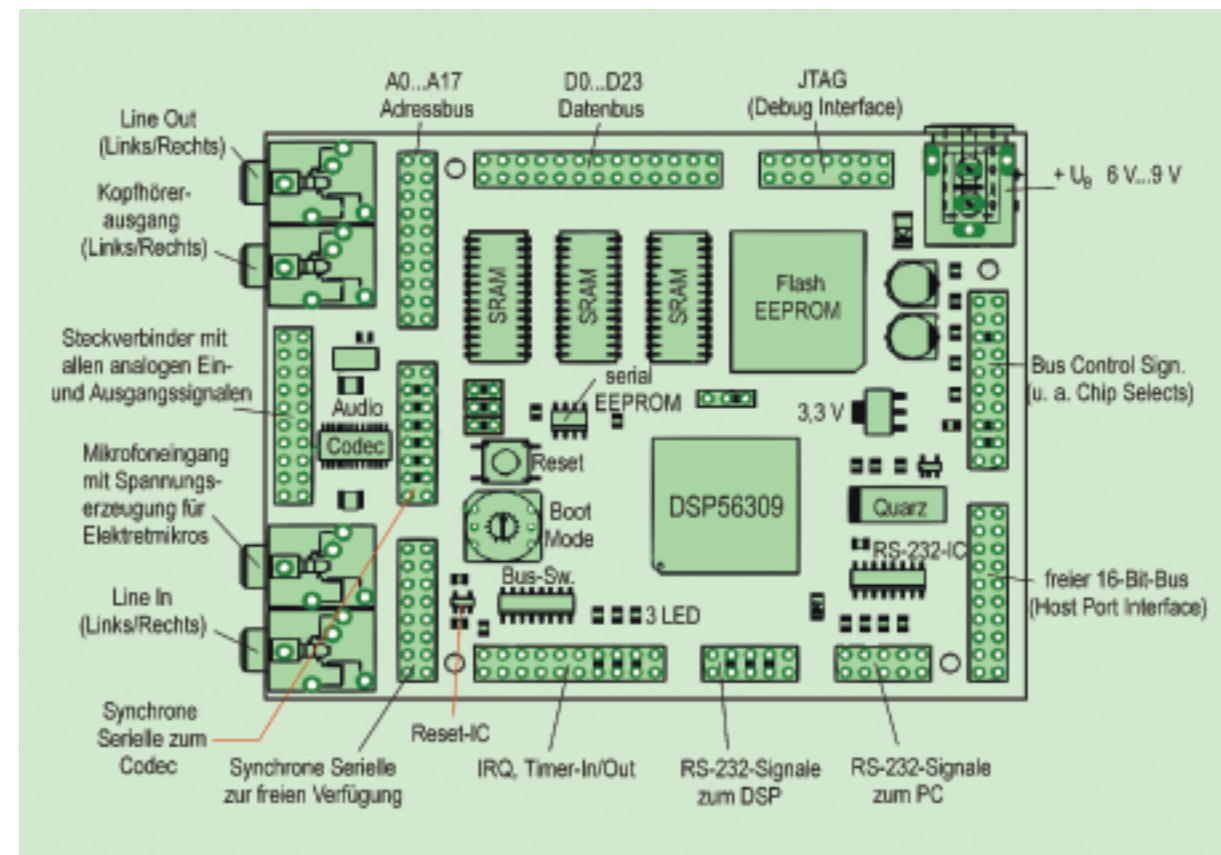
Gehäuse anbietet und der somit handlötbar bleibt. Weitere wesentliche Pluspunkte sind die freie Verfügbarkeit von Assembler, C-Compiler, Debugger und Simulator, wobei die Qualität professionelles Arbeiten erlaubt, was man von einigen DSK- und EVM-Versionen nicht behaupten kann.

Erweitert wird die Funktionalität durch einen Audio-Codex, der neben zwei analogen Ein- und Ausgängen (NF-In Links/Rechts, NF-Out Links/Rechts) zusätzlich noch Mikrofoneingang und Kopfhörerausgang samt Verstärker bietet. Die Speicherbestückung kann optional um schnellen SRAM, Flash und seriellen EEPROM für nichtflüchtige Applikationsdaten (z.B. Rufzeichen, letzte Programmeinstellungen) ergänzt werden.

Zur Außenwelt hält das DSP-Board durch RS-232, zwei synchrone serielle Schnittstellen, ein 16-Bit-paralleles Host-Port-Interface und durch die herausgeführten Adress-, Daten- und Steuerbusleitungen Kontakt. **Bild 1** gibt eine Übersicht über die Zusammensetzung der entwickelten Hardware. Die technischen Daten bei Vollbestückung der 80 mm x 100 mm großen Leiterplatte kann man der **Tabelle** entnehmen.

## Belegung der Hardware

Einer der Vorteile des DSP56309 ist die Code-Kompatibilität mit dem



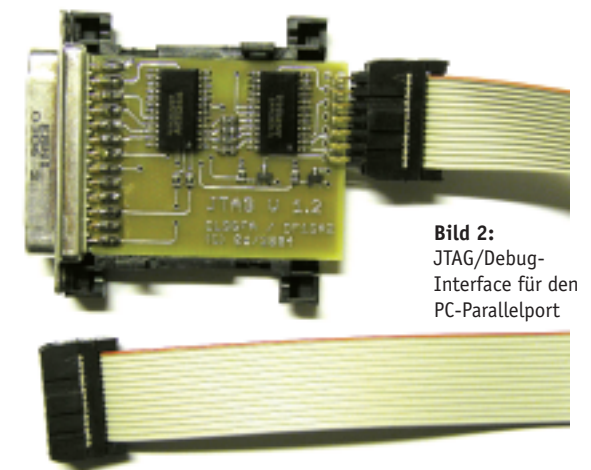
**Bild 1:** Schnittstellen und Bestückung des UniDSP56

DSP56002, für den es bereits eine Menge Amateurfunk-Software aus Zeiten des entsprechenden EVMs gibt. Die Programme bedürfen nur kleiner Anpassungen an den verwendeten Audio-Codex und ggf. konkrete Speicherverhältnisse. Im Falle der Nutzung spezieller ROM-Daten des DSP56002 (Sinus-Tabelle, Kompressionskennlinien  $\mu$ -Law, A-Law) muss geringfügig mehr Portierungsaufwand spendiert werden. Um selbst geschriebene Software mit Hilfe des Debuggers auf den DSP zu laden und im Einzelschritt zu testen, benötigt man einen JTAG-Adapter, der das On-CE-Interface (On-Chip Emulation-Interface) mit dem Parallellport des PCs verbindet. Dieser Adapter besteht neben einigen Kleinteilen nur aus zwei Bustreibern und findet auf einer kleinen Leiterplatte im 25-poligen Sub-D-Stecker Platz (**Bild 2**).

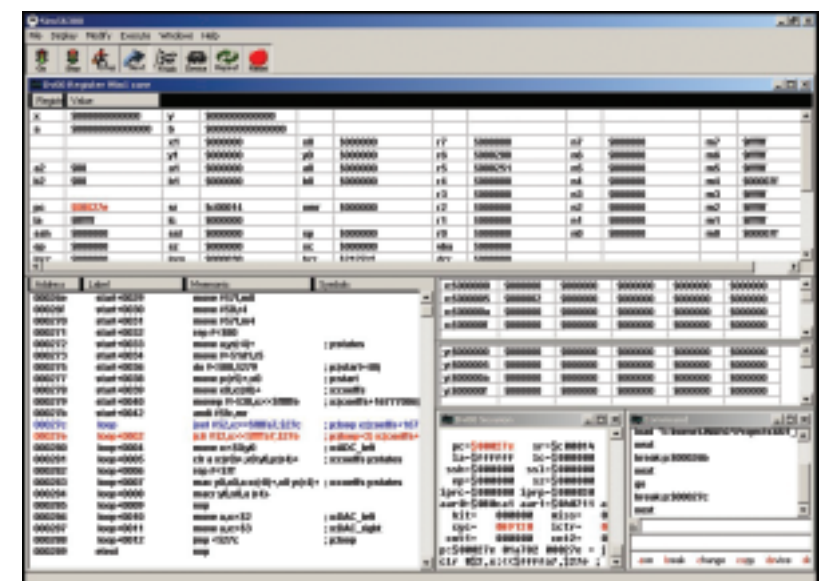
Für den Anwender, der nicht programmieren und Debuggen möchte, gibt es die Möglichkeit, bereits verfügbare und getestete Software über die serielle Schnittstelle (RS-232) eines PCs auf das UniDSP56-Board zu laden und auszuführen. Es ist auch ohne weitere Hilfsmittel möglich, mehrere dieser Applikationsprogramme in den Flash-Speicher zu brennen und dann über einen Programmwählschalter zu laden, sodass ein multifunktionales „Stand-Alone-

Gerät“ entsteht. Über weitere digitale Potentiometer ist sogar die Parametrierung der Funktionen möglich (z.B. Mittenfrequenz und Bandbreite eines digitalen Filters). Das Herunterladen von Programmen und Brennen der Applikationen in den Flash-EEPROM wird durch einen speziellen Boot-Loader ermöglicht, der im Auslieferungszustand des UniDSP56-Boards bereits in den ersten Sektoren des Flash-Bausteins programmiert wurde.

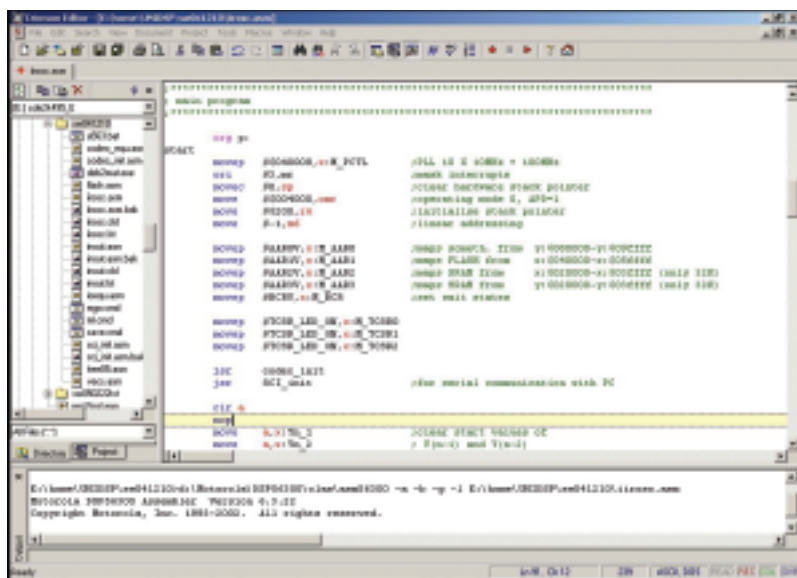
Will man diesen nutzen, muss der Boot-Mode des DSPs mit Hilfe von S2 (Hex-Potentiometer) auf „9“ eingestellt



**Bild 2:** JTAG/Debug-Interface für den PC-Parallellport



**Bild 3:** Arbeiten mit dem Suite56-Simulator



**Bild 4:** Crimson-Editor beim Editieren und Assemblieren eines DSP-Programmes

werden. In diesem Fall bootet der DSP aus dem Flash den (genau genommen Second-Stage-) Boot-Loader, der einen Programmwahlschalter an Port B (Steckverbinder K6) auswertet und je nach dessen Einstellung einen PC auf der RS-232 anspricht.

**Der RS-232-Loader**

Um I/O-Pins zu sparen, ist als Programmwahlschalter ein Hex-Potentiometer eingesetzt. Ist dieses nicht installiert, oder in Stellung „0“ oder „F“ gebracht, erwartet das UniDSP56 einen PC an seiner seriellen Schnittstelle mit 9600 Baud (1 Start-, 8 Daten-, 1 Stop-Bit, keine Flusskontrolle, DTR = Reset).

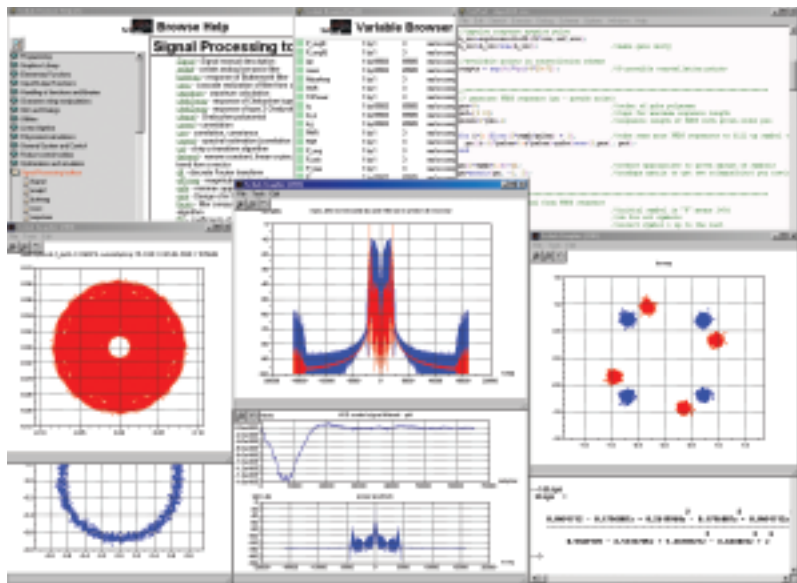
Die initiale Kommunikation ermöglicht das Verändern der RS-232-Geschwindigkeit hoch bis zu 500 kBit/s und die Konfiguration des Speicher-Mappings. Letzteres weist den Speicher-ICs auf der Platine (Flash, zwei Bänke SRAM) die Adressräume und ggf. nötige Wait-States zu. Da-

nach kann mit dem implementierten Befehlssatz auf alle Adressen im internen und externen Speicher zugegriffen werden, um diese mit Daten und Programmen vom PC zu laden. Der letzte Befehl ist in der Regel der Startbefehl zum Abarbeiten des geladenen Programm-Codes ab einer vorgegebenen Adresse.

Dieser Boot-Loader dient neben dem Herunterladen und Ausführen von Programmen auch dem Brennen der verbleibenden 14 wählbaren Applikationen in den Flash-EEPROM.

**Weitere Schnittstellen**

Neben der RS-232-Verbindung gibt es zwei synchrone serielle Schnittstellen (ESSI, Enhanced Synchronous Serial Interface) wie sie typisch für DSPs sind. Sie können mit 50 MBit/s sehr hohe Datenraten erreichen und werden genutzt, um z.B. mehrere DSPs untereinander zum verbinden oder Audio-Codex anzuschließen.



**Bild 5:** Oberfläche des Programms Scilab

Auch beim UniDSP56 ist eine zum Anschluss des Audio-Codex eingesetzt, wobei mit den sechs verfügbaren Signalleitungen der ESSII neben dem Daten-Interface für A/D- und D/A-Wandler auch die Steuerungsschnittstelle zur Konfiguration zu bedienen ist.

Da die analogen Schnittstellen für den Einsatz im Amateurfunk von besonderer Bedeutung sind, soll kurz auf sie eingegangen werden. Es handelt sich beim eingesetzten Audio-Codex um einen TLV320AIC23 von Texas Instruments mit folgenden Merkmalen:

- Zwei 90-dB-SNR-Multibit Sigma-Delta A/D-Wandler mit bis zu 96 kHz Abtastrate,
- Zwei 100-dB-SNR-Multibit Sigma-Delta D/A-Wandler mit bis zu 96 kHz Abtastrate
- Mikrofoneingang mit Elektret-Vorspannung und einstellbarem Verstärker (Stumm, 0 dB, 20 dB),
- Stereo-NF-Eingänge mit programmierbarem Verstärker (12...34,5 dB in 1,5-dB-Schritten)
- Stereo-NF-Ausgänge
- Stereo-Kopfhörer-Ausgänge mit programmierbarem Verstärker (6...76 dB in 1-dB-Schritten).

Die analogen Ein- und Ausgänge sind zur einfachen Adaptierbarkeit sowohl auf NF-übliche 3,5-mm-Klinkenbuchsen geführt, wie auch auf Pfostensteckverbinder, falls eine weitere Platine mit Basisbandfiltern und Quadratur(de)modulator angeschlossen werden soll.

**UniDSP56-Anwendungen**

Die Anwendungen für ein universelles DSP-Board im Amateurfunk dürften sehr breit gestreut sein, sodass nur einige Beispiele aufgeführt werden sollen:

- Universelle NF-Box: NF-Filter (TP, HP, BP) beliebig parametrisierbar in Mittenfrequenz und Bandbreite, manuelles und adaptives (Multi)-Notch-Filter, Noise-Blanker, adaptive Rauschunterdrückungsfilter, Sprachkompressor und -expander, Tx- und Rx-Equalizer,
- Zweikanal-NF-Messinstrument mit bis zu 96 kHz Abtastrate: NF-Signalgenerator (Sinus, Rechteck, Dreieck, Puls und auch völlig „arbitrarily“), NF-Analysator (Zeit-/Frequenzbereich), Ausgabe über LC-Display oder PC (RS-232).
- Modem für digitale Sprach- und Datenübertragung, Sprach-Komprimierung, -Dekomprimierung, Kanalkodierung/FEC (Forward Error Correction), Modulation/Demodulation digitaler Modulationsverfahren,

- Modem für PSK31, RTTY, Packet Radio, SSB, AM, FM, PM

Die ersten beiden Punkte sind vor allem für den Einstieg in die Programmierung und digitale Signalverarbeitung geeignet, weil sie skalierbar und weniger komplex sind.

Die Basisroutinen, z.B. zur Initialisierung und Nutzung des Audio-Codex und der Schnittstellen, sowie komplette Beispielprogramme werden vom Autor frei zur Verfügung gestellt, sodass jeder die ersten Schritte durch Modifikation vorhandener Programme gehen kann.

**Literatur und Bezugsquellen**

- [1] Bezug von Werkzeugen für die Motorola-DSP563xx-Familie: [www.metrowerks.com](http://www.metrowerks.com)
- [2] Editor: [www.ultraedit.com](http://www.ultraedit.com)
- [3] Editor: [www.crimsoneditor.com](http://www.crimsoneditor.com)
- [4] Webseite des Software-Herstellers zu MATLAB: [www.mathworks.de](http://www.mathworks.de)
- [5] Gerrit Buhe, DL9GFA: „Werkzeuge der digitalen Signalverarbeitung“, CQ DL 12/03, S. 848
- [6] Webseite des Simulationsprogramms: [www.octave.org](http://www.octave.org)
- [7] Webseite des Simulationsprogramms: [www.scilab.org](http://www.scilab.org)

**DSP-Entwicklungsumgebung**

Besondere Bedeutung für den Erfolg bei der Programmierung und Erprobung eigener Software haben die Entwicklungswerkzeuge.

Neben dem eigentlichen Assembler und C-Compiler gehört dazu auch ein Debugger, mit dem man den erstellten Code auf die Zielplattform laden und im Einzelschritt testen kann (**Bild 3**). Auch ein Simulator kann im Vorfeld gute Dienste leisten und die Entwicklungszeit verkürzen. Für die Motorola-DSP563xx-Familie kann man diese Werkzeuge frei aus dem Internet laden [1], was ein wesentliches Kriterium bei der Auswahl des DSPs war. Die Erstellung der Quelltexte sollte in einem geeigneten Editor erfolgen, der zusätzlich zum Syntax-Highlighting auch die Einbindung von Assembler- und Compiler-Aufrufen, sowie die Ausgabe ihrer Warn- und Fehlermeldungen unterstützt. Empfehlenswert ist die Shareware UltraEdit [2] oder auch der freie CrimsonEditor [3]. Beide erfüllen genannte Anforderungen und bieten noch eine Menge mehr (**Bild 4**).

**Simulation von Algorithmen**

Vor jedem Programmieren einer Signalverarbeitungsapplikation ist die zu Grunde liegende Mathematik zu entwerfen

und verifizieren. Dazu dienen numerische Simulationsprogramme, von denen MATLAB [4] eines der populärsten sein dürfte. Da es sich dabei um teure kommerzielle Software handelt, haben die wenigsten Leser Zugriff darauf. Aber es gibt auch hier sehr gute freie Alternativen, die hinsichtlich Leistungsfähigkeit kaum nachstehen [5].

Allen voran sind hier Octave [6] und Scilab (**Bild 5**), [7] zu erwähnen. Aufgrund der großen Anzahl an mitgelieferten Signalverarbeitungsfunktionen ist Scilab besonders empfehlenswert. MATLAB-Nutzer mögen hingegen lieber auf Octave zurückgreifen, da man die Syntax nachempfunden hat.

Mit Hilfe der genannten Software können interaktiv Datenströme generiert, bearbeitet und ausgewertet oder sogar ganze Signalverarbeitungsketten (z.B. gesamter Sender und Empfänger) programmiert und simuliert werden. Erst wenn ein Algorithmus hier in der Simulation funktioniert, ist das Implementieren auf einem DSP sinnvoll.

Um die korrekte Funktionalität in der Praxis zu verifizieren, sind Daten vom DSP-Board über den Debugger in die Simulationsumgebung ladbar, um die realen Daten im Zeit- und Frequenzbereich auszuwerten und ggf. mit den Sollwerten der Simulation zu vergleichen.

Selbst das Generieren und Herunterladen von „künstlichen“ Eingangsdaten für das DSP-Board ist möglich, was zum Test der programmierten Funktionen sehr hilfreich ist. Beispielskripte dafür und zum Design und der Simulation von einfachen Filtern bis hin zu gesamten Übertragungstrecken sind ebenfalls vom Autor erhältlich.

**Status und Ausblick**

Es wurden bisher zwölf UniDSP56-Boards und JTAG-Adapter von Beta-Testern aufgebaut. Bei entsprechender Nachfrage kann der Autor weitere Teilbausätze anbieten, die neben der Leiterplatte alle Halbleiter enthalten (außer optionale Speicherweiterung wie SRAM und serieller EEPROM). Die Steckverbinder, Widerstände, Kondensatoren und die beiden Spulen sind über die üblichen Versender selbst zu beschaffen. Listen und Bestellnummern dafür liegen vor.

Der Preis für den Teilbausatz beträgt etwa 99 € und kann bei entsprechender Nachfrage weiter sinken. Im Moment ist der Autor dabei, ein LC-Display und eine PS2-Tastatur direkt an das UniD-



**Bild 6:** Test der DSP-Baugruppe für PSK31-Betrieb

SP56 zu adaptieren, um digitale Betriebsarten auch ohne PC zu ermöglichen (**Bild 6**). Ein HF-Frontend für ein Multi-Mode-Digital-Radio, das neben den herkömmlichen Betriebsarten auch eine neue für digitale Sprachübertragung bei sehr kleinen Signal/Rausch-Abständen ermöglicht, liegt mittlerweile ebenfalls vor, bedarf aber noch einiger Zuwendung und vor allem einer Dokumentation.

An dieser Stelle möchte der Autor dem OM Markus Tröbensberger, DF1SAZ, danken, der das Layouten der Platine übernommen hat.

**Technische Daten der UniDSP56-Platine**

- DSP56309 (24 Bit, Festkomma, 100 MHz Takt, 34 k x 24 Bit interner Speicher, drei Timer/Counter, sechs DMA),
- 32 k x 24 Bit externer SRAM,
- Bis zu 256 kB externer Flash-Speicher für Applikationen,
- 8 kB serieller EEPROM für Applikationsdaten,
- Interface für externen DRAM,
- Audio-Codex mit Line-In L/R, Line-Out L/R, Mic in, Kopfhörer out, ≤96 kHz, 100 dB,
- Spannungsversorgung aller Bauteile mit 3,3 V,
- Reset-Baustein, manueller Reset-Taster, Reset über RS-232 möglich,
- RS-232 mit Pegelwandler, Datenraten bis zu 500 kBit/s,
- Zwei synchrone serielle Schnittstellen (ESSI) mit bis zu 50 MBit/s Datenrate (eine für Audio-Codex),
- Pfostensteckverbinder mit allen wichtigen Signalen,
- JTAG-Interface zum Debuggen (On-Chip Emulation Interface),
- Wahlschalter für Boot-Mode (Flash, Seriell, parallel via Host Port),
- Vier Interrupt-Eingänge, drei Benutzer-LEDs zur Signalisierung.