

STT-Baumappe

von Thomas Schiller - DC7GB

Vorbereitung

Das STT-Shield wird als doppelseitig SMD-bestückte Baugruppe geliefert. Es müssen lediglich noch die Spindelpotis, Steckleisten, Buchsen und der DCDC-Wandler eingelötet werden. Dazu benötigt man einen temperatur-geregelten LötKolben und einen Seitenschneider. Man beginnt am besten mit dem DCDC-Wandler W und den Spindelpotis P1, P2, P4, P5 und P6. **ACHTUNG:** Nicht die Werte verwechseln! Sie sind oben auf der Gehäuseoberseite als 3-stellige Zahl aufgedruckt: 102 = 1k, 103 = 10k und 104 = 100k.

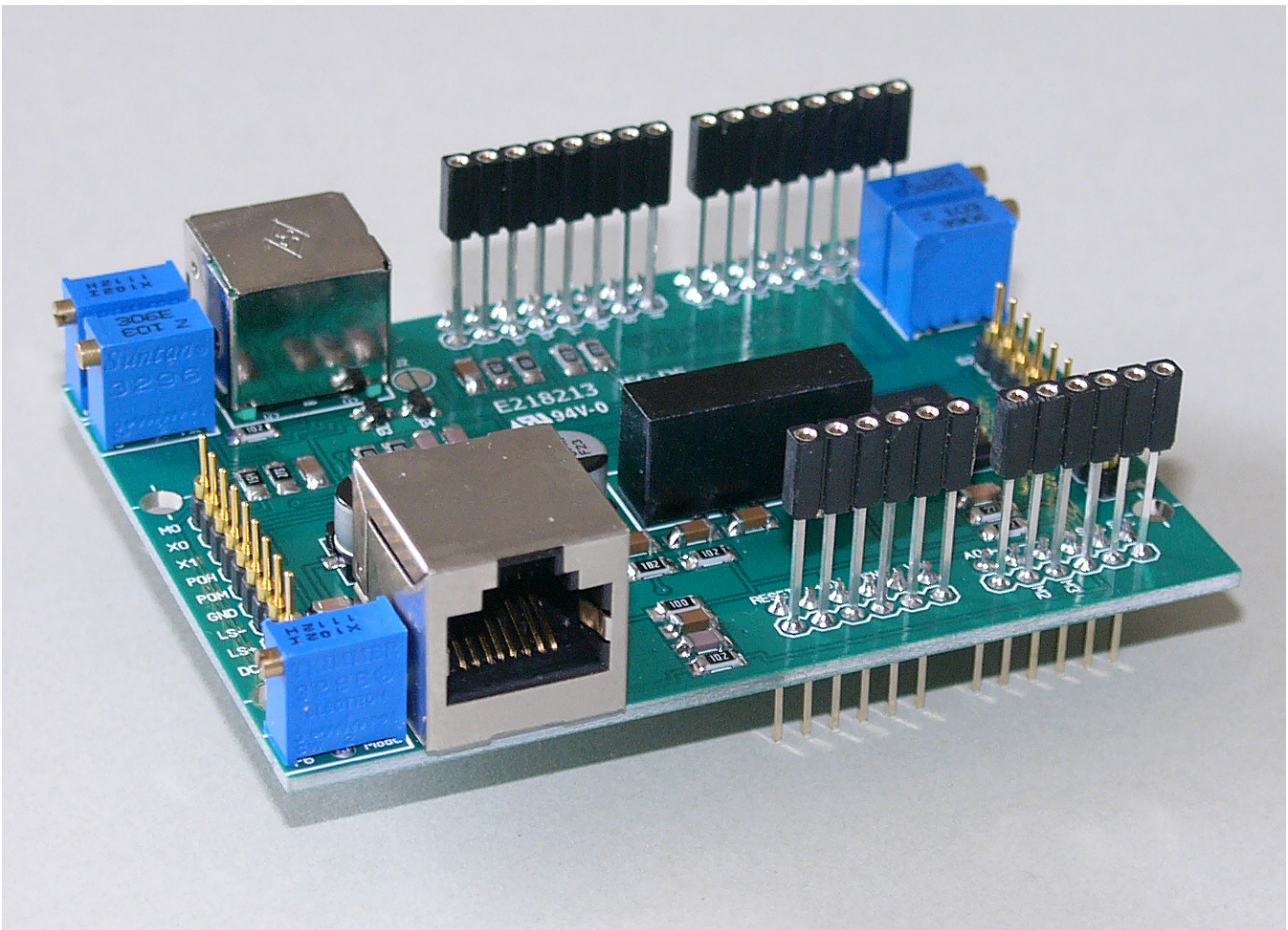


Bild 10: STT-Shield nach dem Einlöten der Potis, Buchsen, Stecker und des DC-Wandlers.

Abgeschirmte Buchsen

Die DATA-Buchse (mini-DIN6) am oberen Rand der Leiterplatte einstecken. Die Massefahnen am Häuserand brauchen nur von unten angelötet zu werden. Die Massefahnen der Mic-Buchse (RJ45-8P8C) um 90° umbiegen und auf etwa 2 mm kürzen. Sie müssen auf der Oberseite angelötet werden!

Hinter der Mic-Buchse befindet sich das Jumperfeld J1, mit dem nach dem Aufkratzen der Drahtbrücken auf der Oberseite der Leiterplatte auch beliebige andere Belegungen des Mic-Steckers realisiert werden können. Dazu müssen die vier blanken Leiterbahnen auf der Oberseite von J1 mit einem scharfen Messer durchtrennt und mit Drahtbrücken individuell neu verlegt werden. Bild 11 zeigt die Belegung beim Blick auf die Oberseite der Leiterplatte. Die quadratischen schwarzen Bereiche sind direkt mit der RJ45 Buchse

verbunden. Die Ziffern entsprechen der Reihenfolge von links nach rechts, beim Blick auf die Steckkontakte. **ACHTUNG:** Die Reihenfolge der Nummerierung ist nicht bei allen Herstellern identisch! Bitte achten Sie bei der Nutzung anderer als Yaesu-Mikrofone daher auf eventuelle Abweichungen. Für spätere Erweiterungen kann das Jumperfeld J1 von der Unterseite mit einer zusätzlichen Stiftleiste bestückt werden. Für den Arduino Uno wird sie jedoch nicht benötigt.

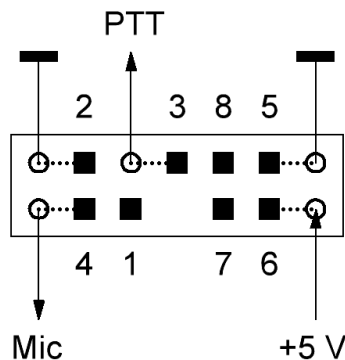


Bild 11: Blick auf das Jumperfeld J1 hinter der Mic-Buchse. Die gestrichelt eingezeichneten Brücken sind durch blanke Leiterbahnen für das Yaesu-Mikrofon MH-31 vorgegeben.

Federleiste Arduino

Dies sind die vier, etwa 19 mm langen 8-poligen Buchsenleisten. Zwei davon müssen auf nur noch sechs Kontakte verkürzt werden. **ACHTUNG:** Es ist nicht empfehlenswert dies mit einem Seitenschneider zu tun, weil dann oft mehr kaputt geht, als nötig. Eine deutlich schonendere Lösung ist mit einer kleinen Mini-Trennscheibe oder einer Säge möglich (Bild 12). Dazu spannt man die Leisten in einen Schraubstock und schleift oder sägt genau auf der zweiten Buchse vom Rand (siehe Bild). Beim Abtrennen zwischen zwei Kontakten geht zu viel Material verloren! Anschliessend feilt man den 6-poligen Teil am Rand sauber.

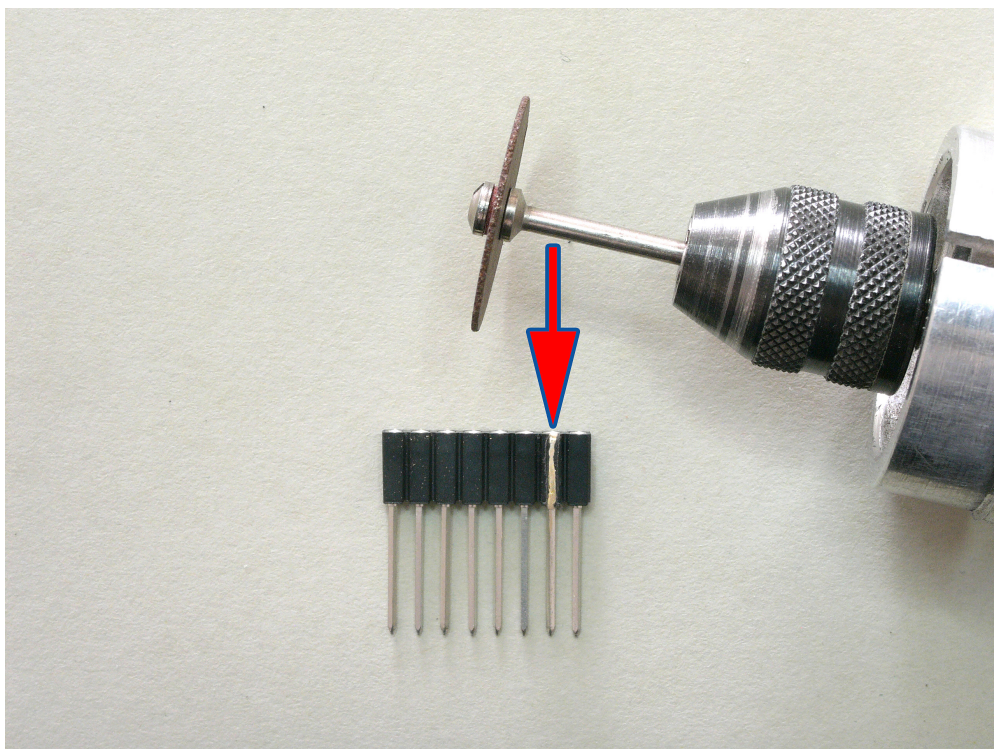


Bild 12: Die 8-polige Buchsenleiste an der Pfeilposition bis zum Metall aufschleifen, mit einer Flachzange abbrechen und die Kanten sauber feilen.

Die Buchsenleisten werden von oben in die Leiterplatte gesteckt. **ACHTUNG:** Die Positionen sind mit einem weißen Rand auf der Bestückungsseite markiert. So kommen die 6-poligen Leisten also z.B. in die unterste Reihe! Die Leisten höchstens 1 mm auf der Lötseite überstehen lassen und zunächst nur einen Pin zum fixieren anlöten. Die Leisten ragen dann auf der Oberseite etwa 1 - 2 mm über die obere Kante der Mic-Buchse. Das sind etwa 18 mm von der Bestückungsseite aus gemessen. Danach die korrekte rechtwinklige Lage in allen Ebenen und die gleiche Höhe überprüfen. Ggf. die erste Lötstelle noch einmal erwärmen und die Leiste neu ausrichten. Sie kann im endgültig eingelöteten Zustand mechanisch kaum noch verbogen werden! Wenn alles optisch einwandfrei aussieht, wird die Leiste vom entgegengesetzten Pin beginnend vollständig angelötet, ohne dass sie dabei noch einmal verrückt wird.

Stiftleiste

Aus der 40-poligen Stiftleiste müssen je zwei 6- und 8-polige Stücke mit einer Flachzange abgebrochen werden. **ACHTUNG:** Diese Stiftleiste ist mit 13,8 mm etwas länger als üblich, damit beim Zusammenstecken ein kleiner Abstand bleibt! Sicherheitshalber kann man auch auf die USB-Buchse noch etwas Isolierband kleben. Bei normal-langen Stiftleisten kann die USB-Buchse auf dem Arduino Uno in ungünstigen Fällen einen Kurzschluss verursachen.

Die vier Stiftleisten werden mit dem kurzen Ende von unten ganz in die verbliebenen weiß umrandeten Felder gesteckt, ausgerichtet und von der Bestückungsseite aus angelötet. **ACHTUNG:** Auf rechtwinklige Lage und gleiche Höhe achten, da man die eingelöteten Leisten nachträglich praktisch nicht mehr ausrichten kann!

Inbetriebnahme

Wer ein 5 V Netzteil hat, kann ohne die beiden anderen Boards zwischen Vin und GND probeweise Spannung anlegen und den aufgenommenen Strom kontrollieren. Er liegt bei etwa 85 mA. Nun kann alles zusammen gesteckt werden. Auf das Arduino Uno Board wird das STT-Shield gesteckt und darauf das LCD-1602-Shield. Alle Boards zusammen nehmen etwa 140 mA auf.

Der Betrieb ohne Netzteil, nur via USB ist möglich, sollte aber nur dann angewendet werden, wenn nur die standardmäßige SQL-Funktion (siehe unten) verwendet wird. Bei Verwendung der SSI-Funktion muss unbedingt ein externes Netzteil mit nicht mehr als 7,5 V verwendet werden, da die 5 V durch den Stabilisator auf dem Arduino-Board dann wirklich sichergestellt sind. Bei USB-Versorgung liegt die Betriebsspannung meist deutlich darunter, was im SSI-Betrieb unter ungünstigen Bedingungen zu Fehlkalibrierungen führen kann. **ACHTUNG:** Bei höheren Spannungen wird der Spannungsregler auf dem Arduino Uno sehr heiß. Eine Kühlung oder ein externer 5 V-Regler ist dann dringend erforderlich!

Firmwareinstallation

Vor dem Abgleich muss die STT-Shield-Firmware ins Arduino-Board programmiert werden. Andernfalls ist der Nullabgleich des Modulationsausgangs nicht möglich. In einigen Fällen muss auf dem PC noch ein USB-Treiber installiert werden. Das Arduino Uno-Board wird dazu ohne weitere Steckkarten mit dem PC über ein USB-Kabel mit dem PC verbunden. Auf dem Board blinkt danach eine gelbe LED.

Fehlt der erforderliche USB-Treiber auf dem PC, so startet unter Windows-XP der Hardware-Assistent (Bild 12). Die automatische Installation und eine Verbindung zum Internet wird dabei empfohlen. Der Suchvorgang kann einige Minuten dauern.



Bild 13: Von links nach rechts: Einstecken der USB-Verbindung und Fortschritt bei der Installation des USB-Treibers.

Während der Installation des USB-Treibers wird dem Arduino Uno automatisch ein COM-Port zugewiesen, über dem die weitere Kommunikation abläuft. Den Namen dieses COM-Ports erhält man über Einstellungen > Systemsteuerung > System > Hardware > Gerätemanager > Anschlüsse (COM und LPT). **ACHTUNG:** Der COM-Port kann bei jedem neuen Anstecken des Arduino-Board an den PC anders lauten!

Zur Programmierung der Firmware in das Arduino-Board benötigt man nur noch das kleine Freeware-Programm Xloader [1]. Es läuft stabil unter Windows-XP. Das Arduino-Entwicklungssystem [2] muss dazu nicht installiert sein! Xloader übernimmt die Kommunikation mit dem Bootloader, der sich bereits im Controller eines frisch gekauften Arduino Uno befindet (Sonderfälle siehe nächsten Abschnitt: SPI-Programmierung). Ausser der Eingabe des COM-Ports und der Auswahl des HEX-Files (die STT-Shield-Firmware) muss man i.d.R. nichts ändern. Die Einstellung der Baudrate ist egal, da sie bei der USB-Programmierung mit dem Bootloader keine Rolle spielt. Nach dem Anklicken von „Upload“ leuchten die TXD- und RXD-LEDs auf dem Arduino-Board, bis die Programmierung nach etwa 5 s beendet wurde. Am unteren Ende des Xloader-Fensters steht dann eine Info-Meldung (Bild 14).

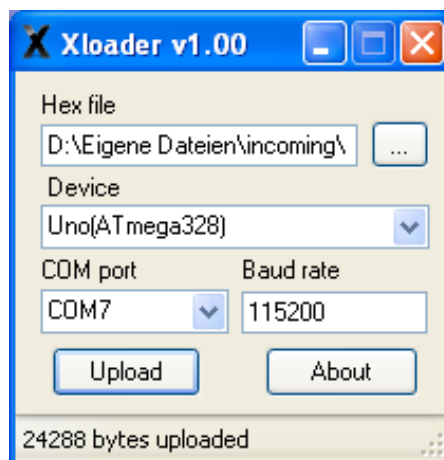


Bild 14: Start von Xloader. Der COM-Port kann vom hier angegebenen Wert abweichen! Die Anzahl der übertragenen Bytes hängt von der Firmwareversion des STT-Shields ab und kann variieren.

SPI-Programmierung

Dieser Abschnitt kann normalerweise übersprungen werden. Wird ein ladenfrischer Controller ATmega328P (z.B. als Ersatz) verwendet, so lässt er sich mit Xloader i.d.R. nicht direkt programmieren, da der Controller im Auslieferungszustand keinen Bootloader enthält. In diesem Fall benötigt man mindestens einen SPI-Adapter (siehe [5] und [6]). Mit Hilfe des Arduino-Entwicklungssystems [2] lässt sich dann der Bootloader in den Arduino Uno programmieren.

Sollte das scheitern, weil z.B. einige spezielle Fuses im Controller aus Versehen oder durch eigene Experimente verändert wurden, so kommt man um den Einsatz eines Hardware-Programmiergeräts meist nicht herum. In diesen Fällen muss man die in Tabelle 1 angegebenen Einstellungen im Programmiergerät

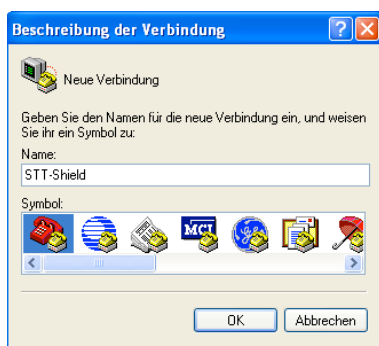
verwenden und kann die STT-Firmware dann auch direkt, ohne den Umweg über einen Bootloader einprogrammieren. Das hat nebenbei auch noch den Vorteil, dass das STT-Shield nach dem Einschalten ein paar Sekunden schneller startet.

Fuse-Name	Einstellung	Erklärung
CLKDIV	aus	Systemtakt-Vorteiler deaktiviert
CLKSEL	1111	Quarzbetrieb > 8 MHz
SUT	11	slow rising Vcc, RESET intern 65 ms
BOD	111	keine Spannungsüberwachung

Tabelle 1: Einstellungen der Programmierfuses bei SPI-Programmierung

Terminalprogramm

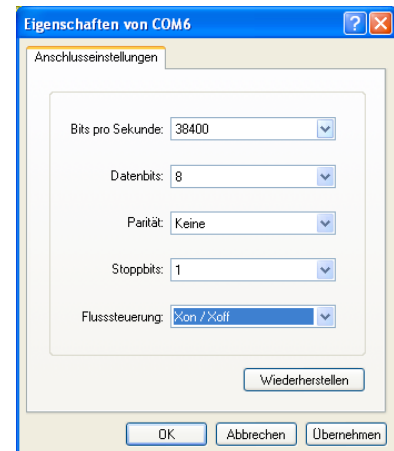
Windows enthält ein Terminalprogramm, das zum Eingeben von Befehlen und zur Anzeige von Statusmeldungen oder empfangenen Paketen eingesetzt werden kann. Unter Windows-XP klickt man „Start“ und dann „Ausführen...“, trägt in das Eingabefeld den Namen „Hyperterm“ ein und startet das Programm mit „OK“. Die weitere Einstellung erfolgt in drei Schritten, wie in Bild 15 angegeben ist. Der im Schritt 2 angezeigte COM-Port kann bei jedem PC anders lauten (siehe Abschnitt: Firmwareinstallation)! Nach Klick auf „Übernehmen“ öffnet sich nach dem Schritt 3 ein Terminalfenster. Es ist sinnvoll die Einstellungen unter „Datei“ und dann „Speichern unter...“ für den nächsten Start von Hyperterm in einer Datei mit passenden Namen zu speichern.



Schritt 1



Schritt 2



Schritt 3

Bild 15: Einstellungen bei Hyperterm unter Windows-XP

Nun kann auf den Arduino Uno erst das STT-Shield und darauf das LCD-1602-Shield gesteckt werden. Die Spannungsversorgung sollte ab jetzt mit einem kleinen Steckernetzteil mit maximal 7,5 V @ 300 mA erfolgen. Höhere Spannungen führen zu einer erheblichen Erwärmung des 5 V Reglers auf dem Arduino-Board und sind zu vermeiden!

Nach einem Druck auf die RST-Taste sollte auf dem nach Bild 14 eingestellten Terminal eine Startmeldung ausgegeben werden. Je nach Voreinstellung des Kontrast-Potis kann das LCD-Display ggf. noch nichts oder nur weiße Blöcke anzeigen.

Kurzanleitung

Der Controller startet beim Anlegen der Versorgungsspannung i.d.R. automatisch. Sollte die Firmware nicht von selbst anlaufen oder sich in einem undefinierbaren Zustand befinden, so kann der Controller durch Drücken von RST jederzeit manuell neu gestartet werden. Bei einem neuen System oder wenn das EEPROM manuell gelöscht wurde, wird nach der Anzeige der Startmeldung automatisch das Setup-Menü aufgerufen. Hier muss zumindest das (i.d.R. eigene) MyCall eingegeben werden, mit dem die STT-Pakete ausgesendet werden sollen. Die Eingabe erfolgt mit den Tasten auf dem LCD-Shield (Bild 16).

Mit den Tasten UP und DOWN kann an der blinkenden Cursorposition ein Buchstabe, eine Ziffer oder ein SPACE ausgewählt werden. Mit RIGHT und LEFT wechselt man die Cursorposition. Ein Call darf am Anfang und in der Mitte kein SPACE und am Ende keine SSID enthalten! Die Eingabe wird durch einen kurzen Tastendruck auf SELECT abgeschlossen. Bei fehlerfreier Eingabe wechselt die Software dann zur nächsten Eingabemöglichkeit.

Alle nachfolgenden Einstellungen sind optional und weitgehend selbsterklärend. Sie können in der Bedienungsanleitung [7] nachgelesen werden. Das Setup kann, ausser bei der Eingabe des MyCalls, an jeder anderen Stelle des Setups durch einen langen SELECT -Tastendruck abgebrochen werden. Der letzte eingegebene Wert bleibt dabei erhalten und das Programm wechselt dann in den Betriebsmodus über.

TX-Abgleich

Wenn die Terminalausgabe erfolgreich war, so läuft die Firmware einwandfrei und das LCD-Display wurde initialisiert. Es müssen nun folgende Abgleichsschritte mit den Trimpotis (Bild 16) durchgeführt werden:

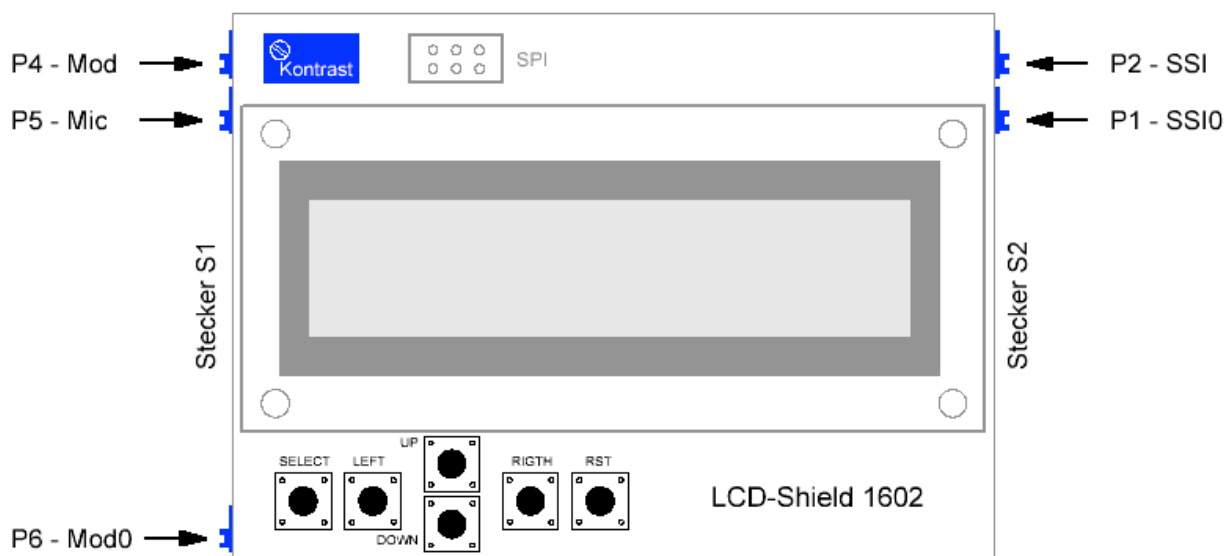


Bild 16: Anordnung der Eingabetasten und Trimpotis am STT-Shield

- Durch Ändern der Einstellung des Kontrast-Trimpotis eine gut lesbare LCD-Anzeige einstellen. Die Spannung am Schleifer des Potis liegt dann zwischen 0,4 V bis 1 V.
- An M1 (Unterseite des STT-Shields) einen Draht anlöten, da der Messpunkt im Betrieb nicht zugänglich ist. Mit einem analogen Zeigerinstrument die Gleichspannung messen und mit P6 (Mod0) auf 0 V einstellen. **ACHTUNG:** Bei galvanisch gekoppelten Modulatoren (meist nur im Selbstbau vorhanden) kann hier auch eine ggf. nötige Diodenvorspannung und damit die Sendefrequenz fein eingestellt werden.
- Das Amplitudenverhältnis von 20:1 (26 dB) des Mikrofondesignals zum STT-Signal an M1 mit einem Oszilloskop messen. Dazu wird das auch später verwendete Mikrofon in die Mic-Buchse gesteckt und in

dem Abstand und mit der Lautstärke besprechen, mit der später auch gefunkt wird. Zischlaute sollte man bei der Messung vermeiden, da sie an M1 durch die Preemphasis dazu führen, dass die Mic-Verstärkung zu gering eingestellt wird! Auch das sonst so beliebte „Ahhh...“ ist meist viel lauter, als man später tatsächlich spricht. Mit P5 (Mic) wird der Mikrofonpegel so eingestellt, dass er an M1 im Mittel $10 V_{ss}$ beträgt. Spitzen und Zischlaute dürfen diese Marken überschreiten! Bild 17 zeigt ein typisches, dabei auftretendes Oszillogramm.

- Voreinstellung der Modulationsspannung: An Pin 1 der DATA-Buchse sollte mit dem Summenregler P4 (Mod) ein NF-Signal von etwa $1 V_{ss}$ als Ausgangsbasis für alle weiteren Einstellungen der Modulation eingestellt werden. Mit diesem Poti wird nach der Auswahl des Transceiver-Betriebsmodus im Betrieb der tatsächliche FM-Hub eingeregelt. Die endgültige Einstellung sollte man mit einem Vergleichsempfänger und/oder einer Gegenstation durchführen. **ACHTUNG:** Das STT-Shield enthält keinen NF-Kompressor! Die Modulation kann daher klarer und dynamischer klingen, als über den Mic-Eingang des Funkgerätes. Die Sprachanteile unter 290 Hz sind zudem deutlich abgesenkt!

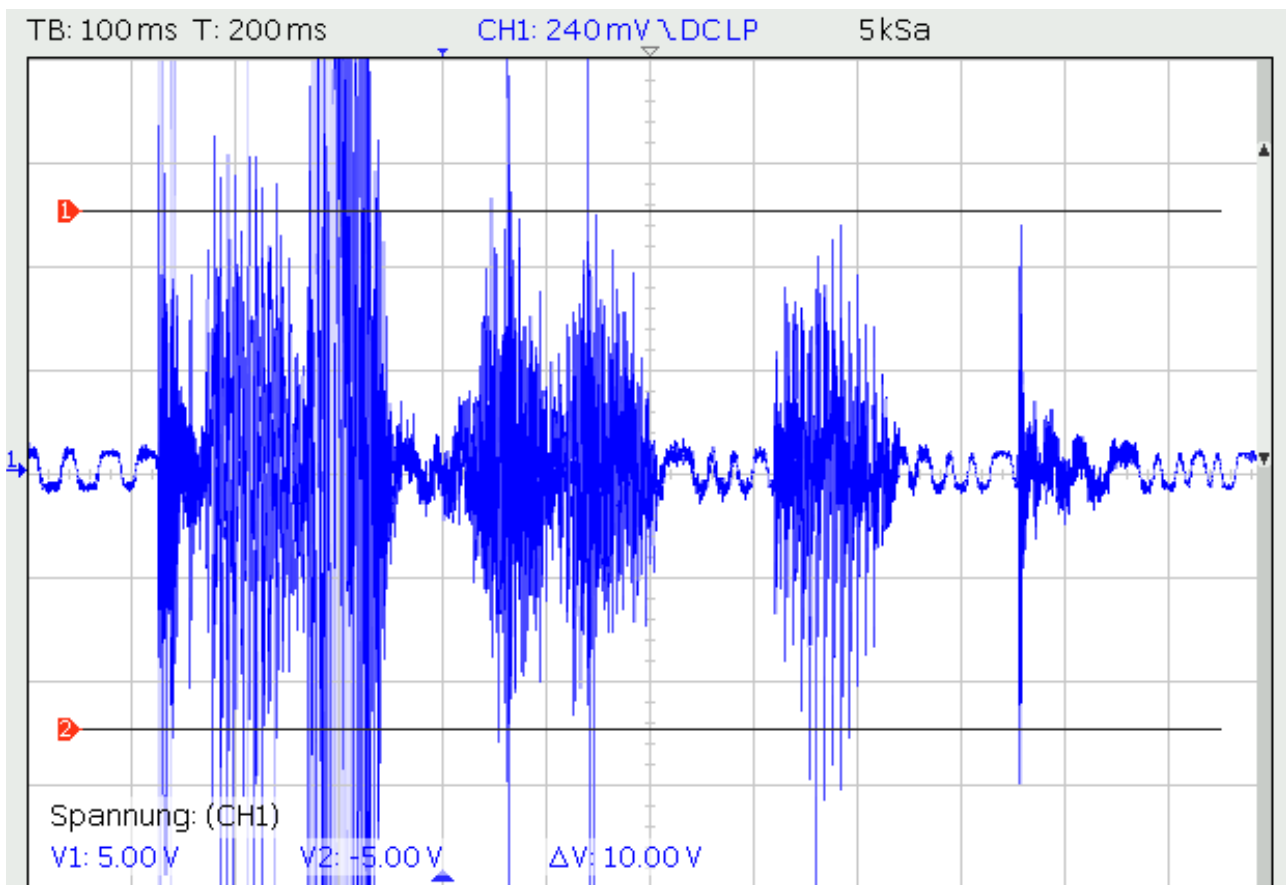


Bild 17: Oszillogramm der NF an M1 beim Sprechen des Wortes: „Testsendung“. Am Anfang erkennt man das STT-Signal mit ca. $500 mV_{ss}$. Die horizontalen Linien markieren den Bereich von $10V_{ss}$. Hinweise im Text beachten!

RX-Abgleich

Im Empfangspfad braucht nur der Pegelkonverter für das Squelch- bzw. SSI-Signal abgeglichen zu werden. Im Originalzustand liegt an Pin 6 der DATA-Buchse ein Squelch-Signal, das beim Öffnen bzw. Schliessen der Rauschsperr zwischen 0 V und 5 V wechselt. Die Polarität ist jedoch bei jedem Hersteller unterschiedlich (siehe Tabelle 2). Liegt wie bei Yaesu bei geschlossener Rauschsperr ein 0 V Pegel an Pin 6, so wird im Setup-Menü des STT-Shields die Einstellung „normal“ benötigt. Dies ist auch die Standard-Einstellung nach der Installation der Firmware. Zumindest bei einigen iCom-Geräten muss vor dem weiteren Abgleich im Setup-Menü die SQL/SSI-Richtung auf „invers“ gesetzt werden [7].

Im Folgenden gibt es zwei Varianten, die davon abhängen, ob man nur die bereits vorhandene SQL-Funktion nutzen möchte oder nach einem kleinen Eingriff in den RX an Pin 6 eine feldstärkeabhängige SSI-

Spannung angelegt hat. Dieser Umbau ermöglicht das Einmessen des RX und eine auf 1 dB genaue Feldstärkeanzeige auf dem LCD-Display, die auch via Telemetrie zur Gegenstation als Rapport übertragen wird. Jeder RX wird dadurch zu einem kalibrierten Messempfänger und viele endlose Diskussionen um Leistungsgewinne bekommen dadurch eine solide Grundlage.

Der Abgleich wird am Beispiel von Yaesu-Pegeln beschrieben, Bei inversen Pegeln (iCom) müssen die 0 V und 5 V-Angaben im Text vertauscht werden. **ACHTUNG:** Die Spannung an M3 kann im ersten Kalibrierschritt auf den LCD-Display angezeigt werden (siehe [7] unter Wartung). Anzeigen < 0 V und > 5V sind jedoch bei dieser Messhilfe nicht möglich! Im Zweifel messen Sie mit einem Digitalvoltmeter an M3. Alle Spannungen müssen auf mindestens 0,1 V genau eingestellt werden.

Alternative 1 (nur SQL-Funktion)

- Messen der Gleichspannung an M3 (SSI).
- Schliessen der Rauschsperrung am RX. Es liegen nun 0 V am Pin 6 der DATA-Buchse. Mit P1 eine Spannung von 0 V an M3 einstellen.
- Öffnen der Rauschsperrung am RX. Es liegen nun ca. +5 V am Pin 6 der DATA-Buchse. Mit P2 eine Spannung von +5 V an M3 einstellen.
- Da sich beide Potis stark gegenseitig beeinflussen, müssen beide Abgleichsschritte mehrfach wiederholt werden!

ACHTUNG: Wird Alternative 1 ohne Umbau am RX verwendet, so hat die Feineinstellmöglichkeit „Rauschsperrung“ im Setup-Menü keinen Einfluss, sondern weiterhin nur der Squelch-Regler am RX! Eine flatternde Rauschsperrung kann in diesem Fall auch nur am RX korrigiert werden. Die Rauschsperrung ist dann auch nur so schnell, wie die bereits vorhandene Rauschsperrung im RX.

Alternative 2 (optionale SSI-Funktion)

Für den Abgleich benötigt man einen kalibrierten Messender mit Dämpfungsglied. Es müssen HF-Spannungen bis hinunter zu -138 dBm (etwa 28 nV) einstellbar sein. An Pin 6 der DATA-Buchse muss eine von der Feldstärke abhängige Spannung liegen. Beim FT817 kann sie z.B. vom FM-ZF-Verstärker (Signal FM-S an TP1070, Q1060, Pin 12) abgenommen werden. Beim Durchfahren des HF-Eingangsspannungsbereichs sollte der Spannungshub an Pin 6 mindestens 1 V betragen, da sonst Störungen und Restauschen die Funktion der Rauschsperrung zu stark beeinflussen. Es sind auch negative Spannungen und inverse Spannungsverläufe möglich!

Über das Wartungs-Menü der Firmware (siehe Betriebsanleitung [7]) ruft man die CAL-Funktion auf. Im ersten CAL-Schritt wird die Spannung an M3 als Messhilfe ständig auf dem LCD-Display angezeigt. Jetzt werden die Extremwerte des SSI-Signals mit P1 und P2 in einem Vorabgleichsschritt auf den erforderlichen Pegelbereich an M3 (0,2...4,8 V) eingestellt:

- HF am RX abschalten und mit P1 etwa 0,2 V einstellen.
- HF einschalten und das HF-Maximum bestimmen. Es liegt bei FM-Geräten meist bei knapp 1 mV (um etwa -50 dBm). Bei höheren HF-Spannungen steigt das SSI-Signal nicht mehr (Begrenzung) oder geht leicht zurück (Übersteuerung). Am Maximum muss mit P2 eine Spannung von 4,8 V eingestellt werden.
- Auch hier beeinflussen sich beide Potis, so dass die Einstellungen des Vorabgleichs mehrfach (etwa fünf Mal) wiederholt werden müssen.
- Wenn der Spannungsbereich stimmt, kann der zweite Kalibrierschritt durch kurzes Drücken der Taste SEL aufgerufen werden. Die Kalibrieroutine führt dann durch die weiteren Abgleichsschritte und fordert zum Anlegen definierter HF-Eingangsspannungen auf.
- Für den nächsten Schritt ist immer die SEL-Taste zu betätigen und die angezeigte HF-Spannung an den RX zu legen. Die Routine erkennt das Ende selbstständig und speichert die kalibrierten Werte mit errechneten Zwischenwerten im EEPROM ab.

In den seltenen Fällen, in denen die SSI-Spannung bei steigendem HF-Eingangssignal nicht wie hier vorausgesetzt zunimmt, sondern abnimmt, müssen die beiden Spannungen von 0,2 V (keine HF) und 4,8 V (maximale HF) beim Vorabgleichsschritt miteinander vertauscht werden. Eine manuelle Einstellung der SQL/SSI-Richtung wie bei der Alternative 1 ist nicht erforderlich. Die Kalibrieroutine erkennt dies selbstständig.

ACHTUNG: Die Potis P1 und P2 dürfen nach der Kalibrierung beider Alternativen nicht mehr verstellt werden! Andernfalls sind im Betrieb Fehlfunktionen möglich und der Abgleich muss wiederholt werden.

Transceiver-Betriebsmodus

Die Belegung der DATA-Buchse wurde zwischen den Firmen iCom, Yaesu und Kenwood als ISO-Standard festgelegt. Dennoch gibt es bei jedem Hersteller leichte Abweichungen. Die Aktivierung der DATA-Buchse und weitere Einstellungen variieren bei jedem Hersteller (Tabelle 2). Das SQL-Signal kann z.B. bei iCom invers vorliegen. Verwendet man nur die SQL-Funktion (anstatt nach einem Umbau des RX die SSI-Funktion), so muss dies im Setup-Menü des STT-Shields unter „SQL/SSI-Richtung“ eingegeben werden. Andernfalls funktioniert die eingebaute Rauschsperrung nicht richtig.

Hersteller/Gerät	Einstellungen am Funkgerät	Status	SQL
iCom IC703	<ul style="list-style-type: none"> Mit MODE-Taste RTTY auswählen, Taste länger als 1s gedrückt halten: SSB fängt an zu blinken. Weiter schalten auf FM 	nicht erprobt	invers
Yaesu FT817: FT857 + FT897:	Auswahl der Betriebsart: PKT, zusätzlich... <ul style="list-style-type: none"> Einstellen von Menü 40: PKT RATE auf 9600 Einstellen von Menü 73: PKT RATE auf 9600 	erprobt	normal
Kenwood TM733	<ul style="list-style-type: none"> per Menü (Taste F, dann Step) 9600 wählen. Der offizielle Mic-Eingang wird automatisch bei PTT via DATA-Buchse deaktiviert. 	nicht erprobt	normal

Tabelle 2: Erforderliche Einstellungen zur Nutzung des STT-Shields bei verschiedenen Geräteherstellern

Stecker S0 und S1

An den Positionen S1 (links) und S2 (rechts), können am STT-Shield Stiftheileiten eingelötet werden. Es lassen sich dann weitere Signale abnehmen, ein Lautsprecher anschliessen und je vier individuelle Datenports verwenden (siehe Tabelle 3 und 4). Die Funktion des Digital-Ports an S2 ist von der gewählten Betriebsart (User/Relais) abhängig. Der Relais-Mode wird nur vom SysOp eines FM-Relais benötigt, der das STT-Shield zur Telemetrieübertragung von Relaisdaten und Zuständen der Relaissteuerung verwenden möchte. Die Zuordnung der Bits zum Relaisstatus kann der Beschreibung der Shield-Codes in [8] entnommen werden. Für den Normaluser hat diese Auswahl keine Bedeutung. Jeder Ausgang an S2 kann im Low-Zustand 25 mA pro Pin treiben. Im High-Zustand stehen nur etwa 300 μ A zur Verfügung.

PIN	Funktion in jeder Betriebsart
S1-X0	digitaler Messausgang im Wartungsmodus
S1-X1	digitaler Messausgang im Wartungsmodus
S1-POH	NF-Ausgang zum heißen Poti-Anschluss
S1-POM	NF-Eingang vom Poti-Schleifer
S1-GND	NF-Masse zum/vom kalten Poti-Ende
S1-LS-	Lautsprecher (GND)
S1-LS+	Lautsprecher (heißes Ende)
S1-DCF	DCF77-Eingang, aktiv low

Tabelle 3: Belegung der linken Stiftheileite S1

An S1-DCF kann das low-aktive Empfangssignal eines DCF77-Empfängers angelegt werden. Die Software dekodiert das Protokoll und stellt die eingebaute Uhr, wenn die Übernahme im Setup-Menü frei gegeben wurde. Nur beim Vorliegen einer gültigen Uhrzeit können Funktionen mit Zeitbezug (QST- QTC-, QTR- und QTH-Pakete) ausgeführt werden. Die Uhr kann aber auch durch den Empfang eines QTR-Pakets gestellt werden. Da der Systemquarz i.d.R. nicht stabil genug ist, weicht die Uhrzeit bereits nach einigen Stunden im Sekundenbereich ab. Hier kann jedoch durch eigenen, individuellen Umbau des 16 MHz-Oszillators ein beliebiger Aufwand zur Steigerung der Ganggenauigkeit getrieben werden.

PIN	User-Mode	Relais-Mode (nur Eingabe)
S2-D7	4 Bit User-Datenausgabe	Freigabe RelaisLink
S2-D6		Freigabe ECHOLINK
S2-D5		6 Bit Relaisstatus (Zuordnung siehe [8])
S2-D4		
S2-D3		
S2-D2		
S2-D1		
S2-D0		
S2-GND	Masse	

Tabelle 4: Belegung des Digital-Ports auf der rechten Stiftleiste S2

Automatische Rauschsperrre

Wird an Pin 6 der DATA-Buchse - wie in der Abgleich-Alternative 2 beschrieben - eine der Feldstärke proportionale SSI-Spannung vom ZF-Verstärker des Empfängers angelegt, so steht eine schnelle Rauschsperrre zur Verfügung, die in nur 3 - 5 ms reagieren kann. Dies ist mindestens 10x schneller, als die meisten Geräte-Rauschsperrren sind. Die Rauschsperrre justiert sich selbstständig und kann langsamen Änderungen folgen. Die Justierung erfolgt nur bei RX-Signalen $< 0,5 \mu\text{V}$ und dauert etwa 2 - 10 Minuten. Ein schwacher Dauerträger wird von der automatischen Rauschsperrre also wirksam unterdrückt. Längere QSO-Durchgänge einer sehr schwachen Station können dadurch allerdings auch unterdrückt werden. Entfällt das schwache Dauersignal, so justiert sich die Rauschsperrre innerhalb von 2 - 10 Minuten wieder empfindlicher ein. In der Praxis ist die Nachregelzeit optimal eingestellt, so dass im Betrieb kaum Probleme auftreten. Die Rauschsperrre kann während der Betriebsanzeige auch manuell durch Drücken der Taste LEFT geöffnet werden.

Nach dem Einschalten des STT-Shields ist die SSI-Rauschsperrre mit Absicht etwas zu fest auf etwa $0,5 \mu\text{V}$ eingestellt. Der RX erscheint dadurch vorübergehend etwas unempfindlicher zu sein, da schwächere Signale zunächst ausgesperrt bleiben. Dies ist aber wesentlich angenehmer, als nach dem Einschalten eine offene Rauschsperrre hören zu müssen, die sich erst nach einigen Minuten auf den richtigen, aber stets individuellen Wert eingestellt hat. Im Betrieb justiert sich innerhalb weniger Minuten immer eine optimale Schwelle weit unter $0,5 \mu\text{V}$ (beim FT817 etwa $0,08 \mu\text{V}$) ein. Mit einem manuellen Squelch ist es über eine längere Zeit kaum möglich eine ähnlich hohe Ansprechempfindlichkeit einzustellen und konstant zu halten!

Sollte sich nach etwa 10 Minuten ein Prasseln oder Knacken der Rauschsperrre bemerkbar machen, so kann die Rauschsperrre im Setup-Menü individuell in Schritten zu 1 dB feinjustiert werden. Es hat keinen Sinn hier um 1 dB zu feilschen, denn die STT-Rauschsperrre ist weitaus empfindlicher, als jeder einfacher Squelch!

Call-Squelch

Diese Funktion kann im Setup-Menü des STT-Shields aktiviert werden. Sie gibt die STT-Rauschsperre nur dann frei, wenn eine der ausgewählten Optionen aktiviert wurde. So lassen sich z.B. Selektivrufsysteme oder Fernsteuerungen bzw. Fernabschaltungen für Relais aufbauen, wenn als UrCall eine nahezu beliebige 6-stellige Buchstaben-Zahlenkombination eingegeben wurde. Insgesamt sind über 2 Milliarden Kombinationen möglich.

Der Data-Port und alle anderen stationsabhängigen Daten werden nur beim Öffnen des Call-Squelchs aufgefrischt. Der Call-Squelch schliesst etwa 3 s nach dem Ausfall der PLL-Synchronisation des Dekoders und kann so kurze Signalunterbrechungen überbrücken. QSP-Frames mit dem entsprechenden Call halten den Call-Squelch sogar für 10 s offen. Unabhängig davon arbeitet die signalabhängige Rauschsperre natürlich weiter, so dass kurze Zwischenrufe einer anderen Station ohne oder mit anderer STT-Kennung ebenfalls erkannt werden können. Auch der Empfang und die Anzeige von QST-/QTC-Meldungen wird durch den Call-Squelch nicht beeinflusst.

Lötbrücken

Auf der Oberseite des STT-Shields befinden sich zwei Lötbrücken J2 und J3, die im Normalfall offen sind. Beim Schliessen von J2 wird an den Mic-Eingang eine Gleichspannung gelegt, um Elektretmikrofone zu versorgen. J3 liegt an der DATA-Buchse. Beim Schliessen kann hier eine Hilfsspannung von 5 V angelegt werden, um kleine Zusatzschaltungen (z.B. Logikinvertierer für PTT) betreiben zu können. Über diesen Pin 5 lässt sich das ganze STT-Shield auch über ein abgeschirmtes PS2-Kabel komplett fernspeisen, wenn man eine Drahtbrücke zwischen Pin 5 der DATA-Buchse und VIN legt. Die Baugruppe kann dann z.B. im Fahrzeug mit Lautsprecher vom Funkgerät abgesetzt betrieben werden.

Sonstige Funktionen

Das STT-Shield kann auch als Modem zur binären Datenübertragung (DATA) benutzt werden, obwohl das bei maximal 140 Bit/s sicher keine Hauptanwendung sein kann. Ausserdem lassen sich über die USB-Schnittstelle individuelle Text-Nachrichten an alle (QST) oder an ein Call (QTC) eingeben.

Der Betrieb einer APRS-Bake, die nur während des aktiven QSOs laufend die eigenen Koordinaten sendet, ist mit STT ebenfalls möglich. Für diesen Betrieb muss das RXD-Signal einer RS232 GPS-Maus im NMEA-0183 Format (4800 bps) über eine kleine Zusatzschaltung (Bild 17) an RXD des STT-Shields angeschlossen werden. An diesem Pin liegt auf dem Arduino Uno Board der Widerstand RN4B in Reihe zum Ausgang M8RXD des USB-Controllers. Wird der USB-Controller nicht zur Dateneingabe an das STT-Shield benutzt, so wirkt RN4B als Arbeitswiderstand für den Open-Collector-Ausgang der Zusatzschaltung, da der USB-Controller im inaktiven Zustand ein H-Signal (5 V) ausgibt. Die Firmware wertet nur \$GPRMC-Datenfelder aus.

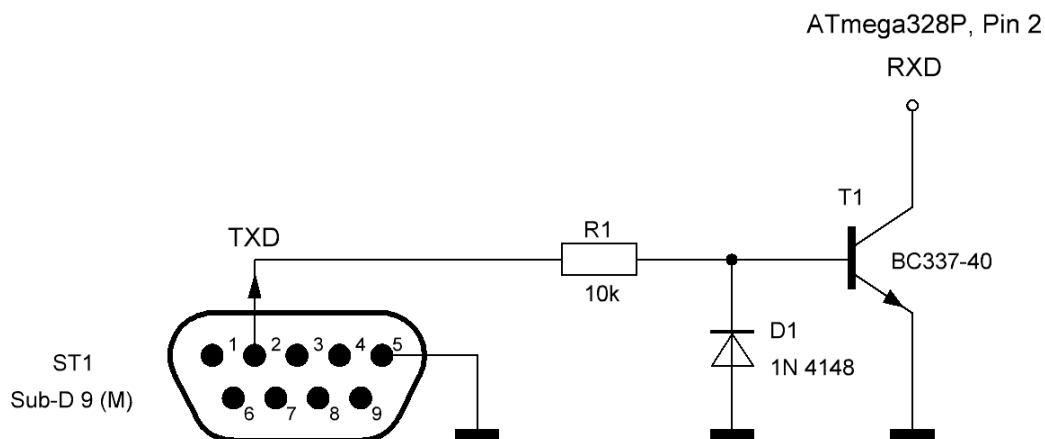


Bild 17: Simpler V24-TTL-Konverter zum Anschluss einer RS232 GPS-Maus an das STT-Shield

Das Yaesu-Mikrofon MH-31 überträgt einige Stellungen seiner Schalter direkt ohne Codierung an den RJ45-Stecker. Durch manuelle Verdrahtung an J1 lassen sich einige der Tasten des LCD-Shields mit den Tasten des Mikrofons verbinden. So kann man dann z.B. die LEFT-Taste zum Öffnen der Rauschsperrung auch direkt am Mikrofon nutzen. Es bietet sich auch die Fernbedienung der UP- und/oder DOWN-Tasten an, weil damit die untere Zeile des LCD-Displays umgeschaltet werden kann (siehe [7]). Damit kann bequem z.B. zwischen Uhrzeit und zur Anzeige von INFO, QTC, QST, S-Meter und Statistik gewechselt werden.

Betriebstechnik

STT sendet im User-Mode das MyCall, den eingegebenen INFO-Text, die Telemetriebits D0...D3 und - sofern vorhanden - DATA-Pakete. Wenn die Uhrzeit via DCF77, GPS oder durch Empfang eines QTR-Pakets bereits vorliegt, so wird Zeit und Datum beim Senden ebenfalls mit eigenen QTR-Paketen ausgegeben. Erst dann können auch QST- und QTC-Meldungen ausgesendet werden. Ein kompletter Durchlauf aller Pakete dauert bei der ersten Aussendung nach Programmstart etwa 15 Sekunden. Nach zwei kompletten Durchläufen (etwa nach 30 s) dauert eine Runde nur noch etwa 3 bis 8 s, je nachdem ob und wie viel Text als QTH und INFO eingegeben wurde. Erst wenn ein neues QTC, QST oder DATA via RXD/USB manuell nachgeladen wurde, verlängert sich die Durchlaufzeit einer Runde wieder. Telemetrieübertragungen benötigen also im Laufe eines QSOs weniger Zeit, bis sie beim Dekoder ankommen.

Ist ein GPS-RX an RXD angeschlossen und die GPS-Eingabe aktiviert, so wird ein QTH-Paket mit Koordinaten anstelle des eingegebenen QTH-Kenners ausgesendet. Wenn eine Gegenstation diese Daten sicher empfangen und verfolgen soll, so muss der Sendedurchgang länger als etwa 15 s dauern. Kurze Durchgänge wie etwa: „Ist hier jemand QRV?“ reichen dazu natürlich nicht aus! In einem normalen QSO-Durchgang sollte es aber keine Probleme geben.

Relaisfunk

Die meisten FM-Relais sind nicht dafür ausgelegt, eine empfangene STT-Sendung ohne lineare Verzerrungen wieder abzustrahlen. Dazu sind in vielen Fällen Anpassungen im Relais-Modulator und im NF-Zweig erforderlich. Z.Z. laufen Arbeiten an einem RX-Modem, das als eine Art Framesammler empfangene User-STT zu einem neuen QSP-Paket zusammen stellt, so dass es zusammen mit der Relais-Telemetrie wieder abgestrahlt werden kann. STT kann momentan also i.d.R. noch nicht zuverlässig über die meisten FM-Relais repeated werden. Ausnahmen sind dennoch möglich!

Fehler

Bitte beachten Sie, dass die Übertragung mit STT zwar sehr sicher ist, sie ist aber nicht auf Geschwindigkeit, sondern auf die Coexistenz mit der gleichzeitigen analogen Sprachübertragung optimiert. Die Sprachübertragung hat im Zweifel immer Vorrang. STT drängelt sich in keiner Situation durch akustische Störungen auf! Es kann daher im ungünstigen Fall auch schon mal bis zum nächsten kompletten Paket-Durchgang (siehe oben) dauern, bis die Daten beim Empfänger vollständig angekommen sind!

Obwohl die STT-Firmware ausgiebig getestet wurde, lassen sich in einem Hobbyprojekt bei dem Umfang der Möglichkeiten nicht alle Anwendungsfälle restlos durchprobieren. Firmwareänderungen und Erweiterungen „in letzter Sekunde“ können leider auch zu unerwarteten Effekten führen. Besonders lästig ist das bei Programmzuständen, die erst nach Stunden oder bei einem Datumswechsel auftreten. Bitte verwenden Sie deshalb immer nur die letzte unter [3] abgelegte Firmware! Sollte im Betrieb ein Fehlverhalten auftreten, so bitte ich um eine Rückmeldung mit einer möglichst genauen Beschreibung, wie es dazu kam. Sofern möglich, hilft mir auch ein Mitschnitt über die DEBUG-Funktion weiter, den ich als ASCII-Textdatei zur Analyse mit der Fehlerbeschreibung bekomme. Testen Sie bereits bitte selbst aus, ob der Fehler nur bei bestimmten gesteckten Kabeln, Netzteilen, PCs, GPS-Empfängern und/oder Funkgeräten auftritt, denn ich kann im Zweifel den ganz speziellen Einzelfall selbst nur angenähert und erst nach vielen Nachfragen

rekonstruieren, was die Fehlersuche verlängert. Ich werde mich schnellstens darum kümmern das Problem zu beheben, sofern es sich um eine ungewollte Funktion handelt.

Problembehandlung

Mit der ASCII-Ausgabe [7] und den Auswahlen „alles ungeprüft“, „alle RX-Pakete“ oder „Logbuch“ lässt sich das Einlaufen von Informationen gut beobachten. Wenn die Übertragung im Standard-Mode auch bei rauschfreien Signalen $> 1 \mu\text{V}$ und normalem FM-Hub nicht zuverlässig funktioniert, so liegt das Problem sehr wahrscheinlich am Modulator des STT-sendenden Funkgerätes. PLL-Aufbereitungen sind für STT meistens ungeeignet, weil sie Modulationssignale $< 100 \text{ Hz}$ oft einfach weg regeln. Das gilt leider auch für einige Mobiltransceiver mit 9k6-FSK-Eingang, die ansonsten für Packet-Radio problemlos funktionieren: Ein Kenwood TM541 ist z.B. zur Aussendung von STT ohne Umbau unbrauchbar!

Auf dem Arduino Uno befindet sich ein USB-Controller, der in der Referenzschaltung [9] über C5 (100 nF) mit dem Signal DTR ein RESET des STT-Shields auslösen kann. Dies kann z.B. dann geschehen, wenn nachträglich am PC ein Terminalprogramm gestartet wird, um z.B. STT-Daten anzeigen oder eingeben zu können. Man verhindert diesen oft unerwünschten RESET, indem man die goldene Lötbrücke RESET-EN auf der Oberseite des Arduino Uno, etwa in der Mitte links zwischen den beiden Quarzen, mit einem scharfen Messer auftrennt.

HF-Einstreuungen sollte es wegen der Abblockung aller Ein- und Ausgänge am STT-Shield nicht geben. Wackelkontakte sind bei Präzisionskontakten meist auch auszuschliessen. Das metallische Gehäuse der Mic- und USB-Buchsen kann jedoch unter ungünstigen Umständen zu Kurzschlüssen auf den darüber liegenden Leiterplatten führen. Hier hilft im Zweifel etwas Isolierband. Für eine stabile Funktion ist der mechanisch feste Einbau des Leiterplattenstapels in ein abgeschirmtes Gehäuse sehr zu empfehlen.

Zusammenfassung

Die neusten Informationen, alle Erweiterungen, bekannte Fehler und Einschränkungen lassen sich unter [11] nachlesen. Unter [12] findet man ein kleines Video, in dem einige Funktionen des STT-Shields an einem Prototypen demonstriert werden. Antworten zu häufigen Fragen zum Betrieb des STT-Shields werden unter [10] gesammelt. Die technischen Daten von STT bei Schmalband-FM Nutzung fasst Tabelle 5 zusammen.

Eigenschaft	Daten
Pegelabsenkung zur Nutz-NF @ 1 kHz	26 dB
Bitfehlerrate bei FM @ $U_e > 0,22 \mu\text{V}$	$\text{BER} < 10^{-6}$
Modulationsart	4-DPSK
PSK-Signalformung	geglättet / optimiert
Synchronwort	0b010101111110
Fehlersicherung	8 Bit CRC nach CCITT
Übertragungsrate brutto	70 Bit/s oder 140 Bit/s
maximale Paketgröße	70 Bytes
minimale Paketzeit	514 ms @ 70 Bit/s
maximale Paketzeit	8,51 s @ 70 Bit/s
Textkomprimierung	auf bis zu 66% via RX37
empfohlene CTCSS-Nutzung	ab 141 Hz

Tabelle 5: Technische Daten des STT-Systems bei Schmalband-FM

Thomas Schiller - DC7GB@vfdb.org

Literatur

- [1] Xloader; <http://xloader.russeotto.com>
- [2] Arduino-Entwicklungssystem; <http://arduino.cc/en/Main/Software>
- [3] Schiller, Th., DC7GB: STT-Firmware; <http://www.mydarc.de/dc7gb/stt/Daten/STT-Shield.hex>
- [4] Philip Hayton; <http://gotbadger.org.uk/post/flashing-arduino-bootloader-using-usbtinyisp-on-windows>
- [5] AVR In System Programmer; http://www.mikrocontroller.net/articles/AVR_In_System_Programmer
- [6] DH8GHH: Ohne viel Aufwand - Programmieradapter für AVR's; FA 7/13, Seite 740-741
- [7] Schiller, Th., DC7GB: STT-Bedienungsanleitung; <http://www.mydarc.de/dc7gb/stt/Daten/Bedienungsanleitung.pdf>
- [8] Schiller, Th., DC7GB: RX37-Codierung; Shield-Code, Seite 12; <http://www.mydarc.de/dc7gb/projekte/daten/Telemetrie/Telemetrie-Codierung.pdf>
- [9] Arduino Uno Reference Design; http://arduino.cc/en/uploads/Main/Arduino_Uno-schematic.pdf
- [10] Schiller, Th., DC7GB: Fragen und Antworten zum STT-Shield; <http://www.mydarc.de/dc7gb/stt/fua.html>
- [11] Schiller, Th., DC7GB: Alles über STT; <http://www.mydarc.de/dc7gb/stt/index.html>
- [12] Schiller, Th., DC7GB: STT-Shield für Arduini-uno; <http://www.youtube.com/watch?v=Xt5HzOJZXPm>