Eine Konfigurationsanleitung

Aus Raspberry Pi wird APRS-Tracker

Dipl.-Ing. Raphael Mayer-Wegelin, DK7MW

In diesem Beitrag wird aus dem preiswerten Kleinrechner ein APRS-Tracker. Eine externe GPS-Maus empfängt die Standort-Daten, die Ausgabe der AX.25-Pakete erfolgen über den bordeigenen Soundausgang.

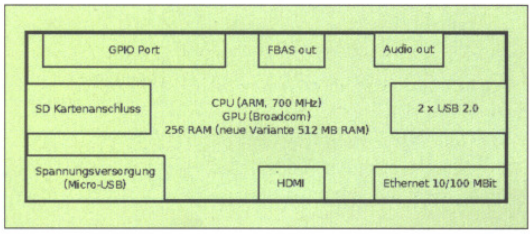


Bild 1: Raspberry Pi mit allen Ein- und Ausgängen

Auf diesen kleinen Minirechner wurde ich erstmals aufmerksam durch einen Artikel der letzten Ausgaben der CQ DL [1]. Begeistert von der Flexibilität für einen geringen Preis und der niedrigen Stromaufnahme habe ich mir den Raspberry Pi besorgt. Verwendet wird als Betriebssystem Linux, das frei zur Verfügung steht und sich in den letzten Jahren rasant weiterentwickelt hat. Auf der Platine befindet sich ein ARM-Prozessor, der mit 700 MHz arbeitet und 256 MB RAM Speicher zur Verfügung hat. Neue Varianten des Minirechners besitzen 512 MB RAM.

Bestückt mit zwei USB-Ports, einem Ethernet (10/100 MBit) und HDMI-Anschluss sowie Audio Out und FBAS-Out-Ausgängen ist er als Minirechner für verschiedene Aufgaben gut geeignet. Darüber hinaus besitzt er ein GPIO-Port, der angesteuert werden kann **(Bild 1)**. Der Preis liegt bei 40 €. Eine genaue Beschreibung findet man im Internet [2].

Das Einstellen und Einrichten des Rechners erfolgt mit Hilfe der vielen Webseiten zu diesem Thema. Dort findet man viele Tipps und Antworten. In [3] gibt es beispielsweise eine englischsprachige Beschreibung für Einsteiger und die für den Rechner vorkonfigurierten Betriebssysteme zum Download.

Zur Person

Dipl.-Ing. Raphael Mayer-Wegelin, DK7MW
Jahrgang 1975, Amateurfunkgenehmigung seit 2008, Studium der Nachrichtentechnik, beruflich in der Automobilbranche tätig.
Besondere Interessen: Antennentheorie, APRS, Programmierung von Mikrocontrollern

Anschrift:
Birkenweg 13
61381 Friedrichsdorf
dk7mw@darc.de

Nachfolgend eine Beschreibung über die weiteren Einstellungen und die Erfahrungen, die ich gemacht habe. Diese sollen als Beispiel und nicht als allgemeine Anleitung dienen, da nicht alle Variationen durch die Wechselwirkung und Weiterentwicklung der einzelnen Software-Komponenten erfasst werden können.

Als Grundlage dient das Raspbian „wheezy"-Betriebssystem, das auf Debian basiert. Die grafische Oberfläche ist so konfiguriert, dass sie entweder bei jedem Hochfahren automatisch oder durch Eingabe von „startx" manuell gestartet werden kann. Das Speichermedium ist bei mir eine 8 GB SD-Karte.

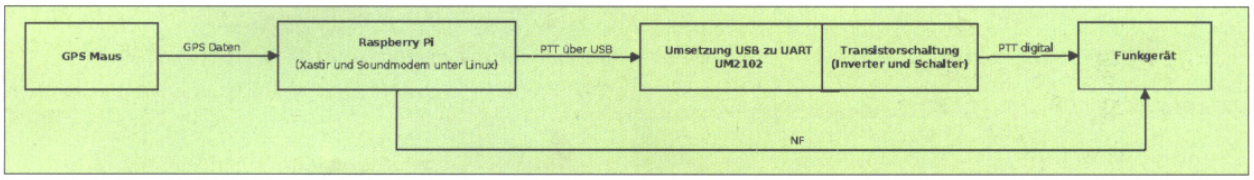


Bild 2: Übersicht der einzelnen Hardware-Komponenten

Als Minimum empfehle ich 4 GB, da bei einer 2-GB-Karte der Speicherplatz schnell zu knapp sein wird. Die Partition muss entsprechend erweitert sein. Nach dieser Grundkonfiguration sollte die Vorinstallation auf den neuesten Stand aktualisiert werden durch Öffnen von LXTerminal und Eingabe von

```
sudo apt-get update
sudo apt-get upgrade
```

Ein Neustart sollte erfolgen, beispielsweise durch

```
sudo reboot
```

Hardware

Zu der Konfiguration ist für den Kleinrechner natürlich auch eine bestimmte Hardwareumgebung notwendig **(Bild 2)**. So ist zum Empfang der aktuellen GPS-Daten eine entsprechende Maus erforderlich, die für wenig Geld gebraucht zu erwerben ist. Ich setze beispielsweise die USB-Variante GM-200 von HOLUX ein. Man kann aber ebenso einen anderen GPS-Empfänger verwenden. Dabei sollte man sich allerdings vorher erkundigen, ob diese auch wirklich mit dem Rechner funktioniert. Im Netz findet man hierzu weitere Informationen. Zum Testen der GPS-Maus empfiehlt sich das kleine und einfache Terminal-Programm Minicom [4], das gegebenenfalls installiert werden muss. Bei stationärem Einsatz kann natürlich auf den GPS-Empfang verzichtet werden.

Zudem benötigen wir eine Möglichkeit zur Ansteuerung der PTT-Tastung. Dies ist einfach durch ein Mini-USB-Modul realisiert, das eine Umsetzung von USB auf die serielle UART-Schnittstelle vornimmt. Dieses Modul UM2102 gibt es als Komplettbausatz bei ELV [5]. Die Ansteuerung erfolgt über dessen RTS-Ausgang. An diesem sind zwei Transistorschaltungen angeschlossen, die aus einer Inverter- und einer Schalterstufe bestehen. Diese wiederum kontrollieren den PTT-Eingang des Funkgerätes. Ein Schaltungsbeispiel ist in [1] zu finden. Um die NF übertragen zu können, muss der Soundausgang des Raspberry Pi mit dem Funkgerät angeschlossen werden. Falls das Funkgerät die VOX-Funktion unterstützt, so kann das Umsetzungsmodul USB zu UART und die Transistorstufe entfallen, und man spart sich die entsprechende Software-Konfiguration. Gleichzeitig wird ein USB-Port für andere Aufgaben frei.

Möchte man ihn mobil betreiben, so benötigt man ein entsprechendes Kfz-Kabel zur Stromversorgung, das über den üblichen Elektronikhandel zu beziehen ist.

USB-Ports

Bevor mit der eigentlichen Konfiguration begonnen werden kann, müssen zuerst die zugewiesenen Ports herausgefunden werden mit

```
dmesg | grep ttyUSB
```

Ist man sich nicht sicher über die Zuordnung, so entfernt man eines der Geräte vom USB-Port und schließt es gleich wieder an. Durch nochmaligen Aufruf des Kommandos sehen wir anhand der Änderungen, wie die Ports zugewiesen sind. Bei mir ist beispielsweise die GPS-Maus dem Port ttyUSB0 zugeordnet **(Bild 3)**, da ich diese kurz entfernt hatte. Folglich ist der USB-zu-UART-Konverter an Port ttyUSB1 angeschlossen.

Die Zuordnung sollte immer geprüft werden, sobald sich Änderungen am USB-Anschluss ergeben, da falsch zugewiesene Ports zu unvorhersehbarem Verhalten führen können, sobald die entsprechenden Programme ohne Anpassung gestartet werden.

Soundmodem

Um die digitalen Daten ausgeben zu können, wird der Audio-Ausgang des Raspberry Pi benutzt. Hierzu wird das Soundmodem [6] installiert.

```
sudo apt-get install soundmodem
```

Die Konfiguration hat bei mir nur mit dem File Manager richtig funktioniert. Man findet ihn unter Accessories **(Bild 4)**. Zuerst müssen wir jedoch das Verzeichnis mit Admin-Rechten öffnen. Dies geschieht mit der Funktion „Open Current Folder as Root" unter Tools **(Bild 5)**. Die Einstellungen für das Soundmodem werden mit soundmodemconfig vorgenommen, welches sich in /usr/bin befindet. Nach dem Öffnen mit dem File Manager legen wir eine neue Konfiguration mit dem Namen AX25 an **(Bild 6)**. Gemäß **Bild 7** stellen wir den Mode auf alsa und den ALSA Audio Driver auf plughw:0,0 ein. Unter PTT Driver steht der Portname, der zur PTT-Ansteuerung verwendet wird. Normalerweise muss hier eine serielle Schnittstelle angegeben werden. Wie wir bereits wissen, ist es bei mir allerdings der USB-Port /dev/ttyUSB1,

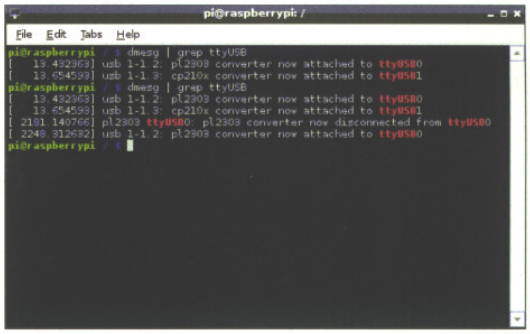


Bild 3: Zugewiesene USB-Ports herausfinden

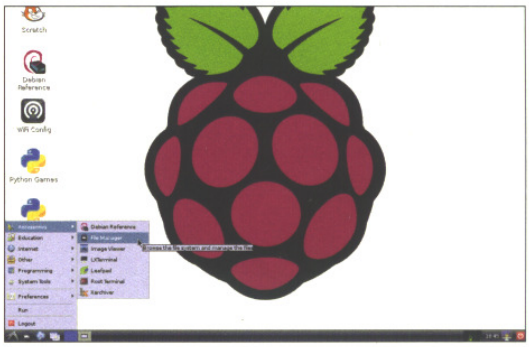


Bild 4: Aufruf des File Managers

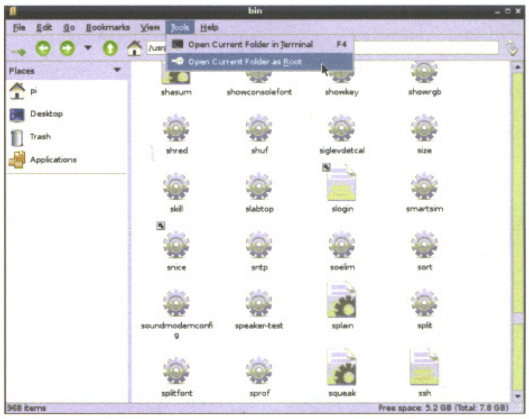


Bild 5: Das Verzeichnis mit Root-Rechten wird geöffnet, um auch die Schreibberechtigung zu erhalten

an dem der USB-zu-UART-Konverter angeschlossen ist. Dies muss gegebenenfalls angepasst werden. Alle anderen Einstellungen bleiben unverändert.

Als Nächstes muss ein Kanal erstellt werden **(Bild 8)**. Wir klicken auf Channel0, und unter Modulator wählen wir als Mode afsk. Bei Demodulator lassen wir bewusst den Mode auf off, da wir keinen Mikrofon- oder LineIn-Eingang zum Demodulieren besitzen. Unter PacketIO wird KISS als Mode ausgewählt und als File lassen wir die Voreinstellung /dev/soundmodem0 stehen **(Bild 9)**. Alle anderen Parameter werden auch hier nicht verändert.

Bezug des **Raspberry Pi's** auch über:

Lindenallee 6
34225 Baunatal
Tel. (05 61) 9 49 88 73
Fax (05 61) 9 49 88 55

www.darcverlag.de
verlag@darcverlag.de

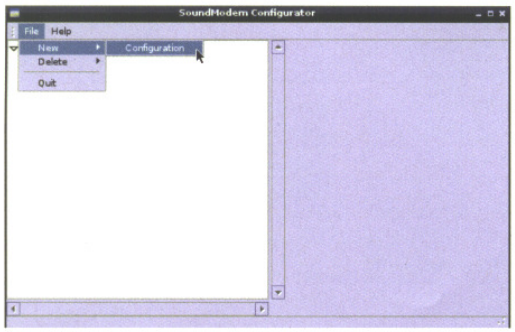


Bild 6: Anlegen einer neuen Konfiguration

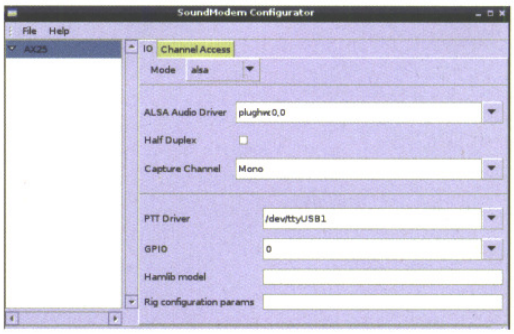


Bild 7: Schnittstelleneinstellung

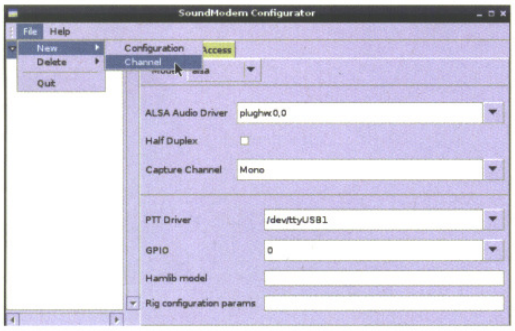


Bild 8: Hinzufügen eines neuen Kanals

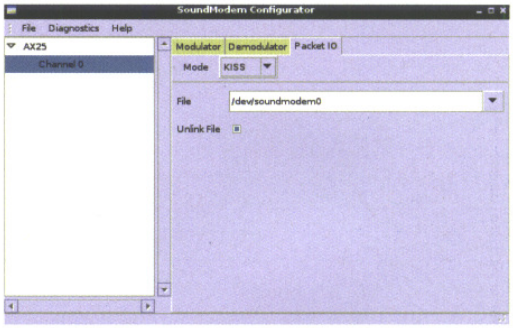


Bild 9: Änderungen unter PacketIO

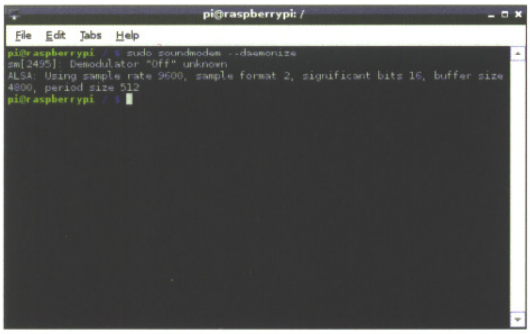
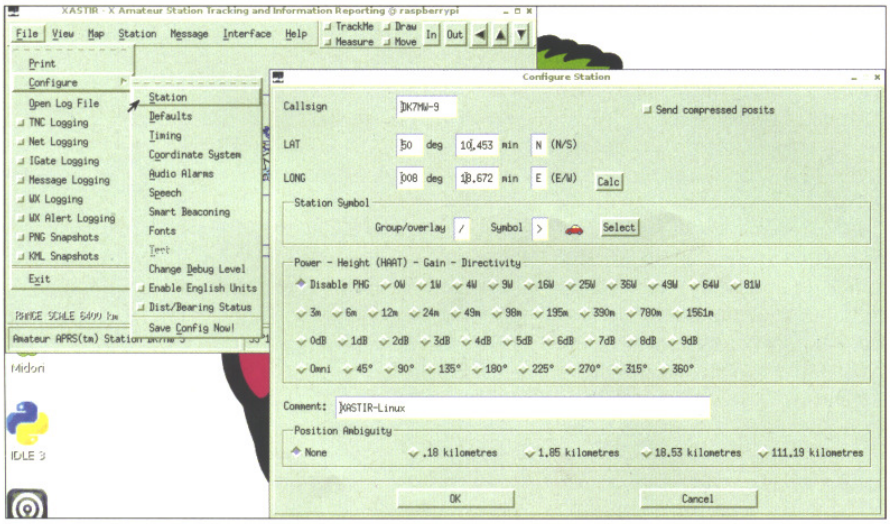


Bild 10: Erfolgreiche Rückmeldung des Soundmodems

Bild 11: Anpassungen des Rufzeichens und des Symbols



Die Einstellungen werden automatisch nach dem Schließen gespeichert, vorausgesetzt das Programm wurde auch tatsächlich mit Admin-Rechten aufgerufen. Zur Sicherheit können die Änderungen durch erneuten Programmstart kontrolliert werden.

Ein direkter Test – wie sonst üblich – mit der Modemfunktion ist mit dem Soundmodemkonfigurator nicht möglich. Wird es trotzdem versucht, so erhält man die Fehlermeldung „Cannot start audio IO", da wir zuvor wissentlich nur einen Audioausgabekanal und keinen Audioeingabekanal konfiguriert haben. Beides ist aber für diese Funktion notwendig. Wir benötigen dies allerdings nicht. Das Soundmodem können wir nun starten und geben in die Kommandozeile folgendes ein

```
sudo soundmodem --daemonize
```

Der zusätzliche Parameter bewirkt, dass das Programm als Daemonprozess parallel im Hintergrund läuft, und wir weitere Eingaben machen können. Falls keine Fehler festgestellt wurden und das USB-zu-UART-Modul angeschlossen ist, bekommt man eine Antwort wie in **Bild 10** dargestellt.

Xastir

Als APRS-Client ist das frei verfügbare Xastir [7] geeignet. Die Installation erfolgt durch

```
sudo apt-get install xastir
```

Damit das Programm richtig läuft, sollten die entsprechenden Rechte eingerichtet werden mit

```
sudo chmod 4755 /usr/bin/xastir
```

Die Kommandozeile für den Aufruf des Programms ist einfach der Programmname

```
xastir
```

Nach dem Start nehmen wir die allgemeine Konfiguration vor. Unter File,

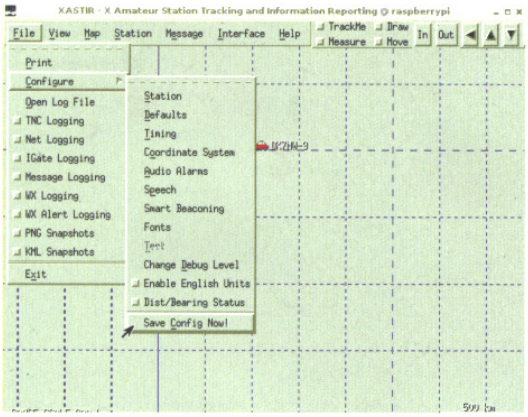


Bild 12: Damit werden alle Einstellungen sofort gespeichert

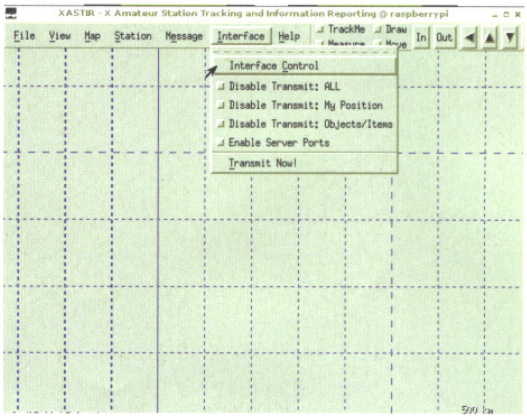


Bild 13: Schnittstelle zum Soundmodem

Configure und Station **(Bild 11)** passen wir das Rufzeichen an und stellen das entsprechende Symbol ein. Beim Start und unplausiblen Daten wird dieses Fenster immer automatisch geöffnet.

Bei stationärer Verwendung ohne Empfang der GPS-Daten wird die statische Position in dieses Fenster eingetragen. Die noch folgende Konfiguration der GPS-Maus entfällt dann.

Unter File und Configure kann man für den späteren Betrieb weitere relevante Einstellungen machen. Weiterführende Informationen dazu findet man im Netz. Nach den Eingaben führen wir Save Config Now! aus, damit alle Änderungen sofort übernommen werden und nicht erst nach Neustart des Programms **(Bild 12)**. Nun konfigurieren wir die Schnittstelle zum Soundmodem. Dies erfolgt durch das Interface Control **(Bild 13)**. Mit Add fügen wir ein neues Interface hinzu und wählen Serial KISS TNC aus.

Wir prüfen und aktivieren gegebenenfalls die Funktion Activate on Startup, damit bei jedem Hochfahren automatisch das Interface gestartet wird. In TNC Port geben wir den Schnittstellennamen von unserem Soundmodem /dev/soundmodem0 ein. Dies entspricht der Eingabe, die wir bereits bei der Konfiguration des Soundmodems gemacht haben **(Bild 14)**.

Als Nächstes starten wir unter Interface Control unser neu erzeugtes Interface Serial KISS TNC, indem es markiert und mit Start manuell aktiviert wird. Der Status ändert sich von DOWN auf UP **(Bild 15)**, sobald das Interface erfolgreich hochgefahren ist.

Vorausgesetzt das UM2102-Modul ist am USB-Port angeschlossen, können wir das Senden und die Ansteuerung der PTT-Taste prüfen. Zudem können wir testen, ob die Soundlautstärke ausreichend ist. Dies erfolgt durch Transmit Now! unter Interface **(Bild 16)**. Falls notwendig, beenden wir wieder Xastir und ändern den Audiopegel durch Aufruf des Mixers **(Bild 17)**:

```
sudo alsamixer store
```

Jetzt starten wir wieder Xastir und prüfen die Lautstärke. Gegebenenfalls rufen wir den Mixer nochmal auf. Sobald der Pegel den richtigen Wert hat, wechseln wir erneut zu Xastir und konfigurieren das Interface zu unserer GPS-Maus. Mit Add wählen wir Serial GPS aus und führen die nötigen Einstellungen aus.

Die GPS-Maus ist an /dev/ttyUSB0 verbunden. Da ich sie mit einer Datenrate von 4800 Baud und den Parametern 8 Datenbits, keine Parität und ein Stopbit benutze, sieht meine Konfiguration aus wie in **Bild 18**. Auch hier aktivieren wir die Funktion Active on Startup.

Mit angeschlossener GPS-Maus am USB-Port aktivieren wir das Interface manuell wie zuvor mit dem Soundmodem, um zu prüfen, ob der Status von DOWN auf UP wechselt **(Bild 19)**. Falls der Interface-Start fehlschlägt, so kann die Verbindung zur GPS-Maus nochmals mit dem Terminal-Programm Minicom getestet werden.

Automatisierung

Sobald der Raspberry Pi hochfährt, sollen die notwendigen Programme wie Soundmodem und Xastir automatisch starten. Hierzu sind aber noch bestimmte Einstellungen vorzunehmen. Das Soundmodem wird beim Hochfahren des Rechners mit Hilfe eines Skriptes gestartet. Erstellt wird es mit

```
sudo nano /etc/init.d/soundmodem
```

Damit erzeugt der Texteditor Nano die neue Datei soundmodem. Nun gibt man den Skriptcode im **Kasten** ein. Nach der Eingabe wird der Texteditor mit der Tastenkombination Strg+X und der Bestätigung mit der Taste Y, dass man es abspeichern möchte, beendet.

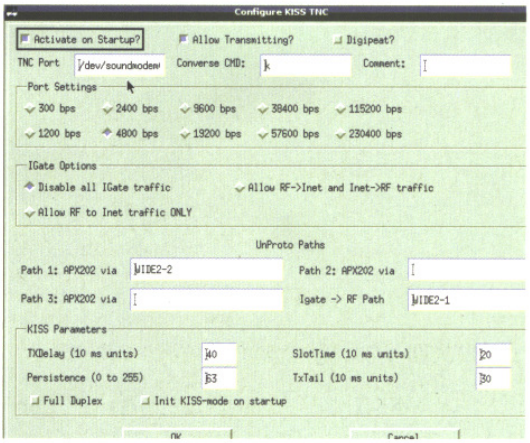


Bild 14: Serial KISS TNC-Interface

/etc/init.d/soundmodem

```
#!/bin/sh -e
### BEGIN INIT INFO
# Provides:          soundmodem
# Required-Start:    $syslog
# Required-Stop:     $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start/stop of soundmodem
### END INIT INFO

case „$1“ in
  start)
    sudo soundmodem —daemonize
    rm /home/pi/.xastir/xastir.pid 2>/dev/null
    ;;
  stop)
    sudo killall soundmodem
    ;;
  *)
    echo „Usage: /etc/init.d/soundmodem {start|stop}“
    exit 1
    ;;
esac

exit 0
```

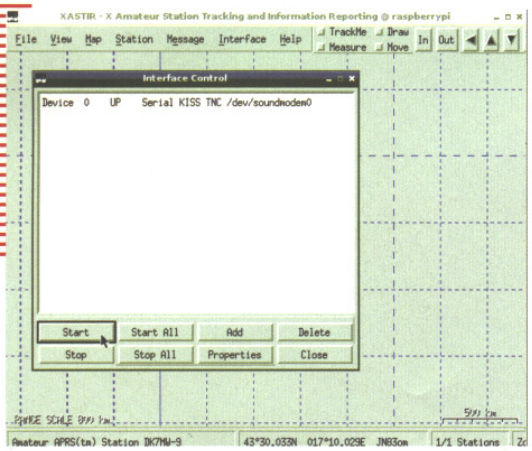


Bild 15: Soundmodem Interfacestatus

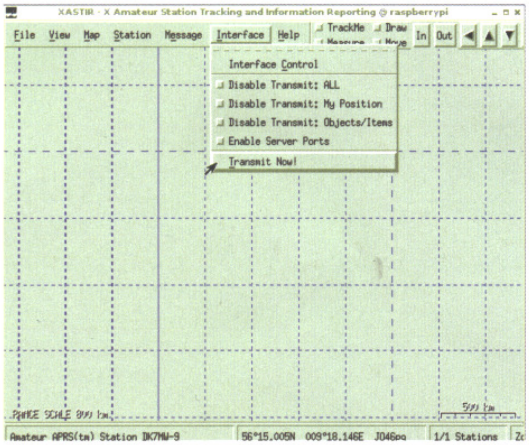


Bild 16: Manueller Sendetest

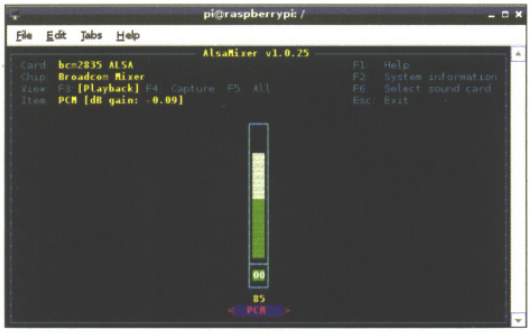


Bild 17: Anpassung des Audiopegels

Literatur und Bezugsquellen

[1] Stefan Hüpper, DH5FFL: „Ein Echolink-Rechner mit <3 W Leistungsaufnahme", CQ DL 10/12, S. 695

[2] Raspberry Pi, http://de.wikipedia.org/wiki/Raspberry_Pi

[3] Aktuelle Informationen zum Raspberry Pi, www.raspberrypi.org

[4] Textbasiertes Terminalemulationsprogramm Minicom, http://alioth.debian.org/projects/minicom/

[5] Mini-USB-Modul UM2102, ELV: www.elv.de/mini-usb-modul-um2102-komplettbausatz.html

[6] Thomas Sailer, HB9JNX/AE4WA: Multiplatform Soundcard Packet Radio Modem: ww.baycom.org/~tom/ham/soundmodem/

[7] APRS client Programm Xastir: www.xastir.org/wiki/Main_Page

Nun müssen wir noch das Skript ausführbar machen mit

```
sudo chmod +x /etc/init.d/soundmodem
```

Das Skript kann manuell gestartet werden

```
/etc/init.d/soundmodem start
```

Nach korrekten Eingaben erhalten wir die gleiche erfolgreiche Antwort wie beim manuellen Aufruf vom Soundmodem, und können es zusammen mit Xastir testen. Falls nötig, lässt sich das Soundmodem wieder beenden durch

```
/etc/init.d/soundmodem stop
```

Im Fehlerfall sollte man das Skript nochmals mit dem Texteditor kontrollieren und gegebenenfalls korrigieren. Man ruft dabei den Editor genauso auf wie in den beschriebenen Schritten zuvor bei der Neuerstellung der Skriptdatei.
Nach erfolgreichen Tests wird das Skript für den automatischen Start beim Hochfahren im System angemeldet mit

```
sudo update-rc.d soundmodem defaults
```

Falls erforderlich kann der automatische Aufruf auch wieder deaktiviert werden

```
sudo update-rc.d -f soundmodem remove
```

Der APRS-Client Xastir soll natürlich ebenfalls nach dem Start der grafischen Benutzeroberfläche automatisch aufgerufen werden. Dies erfolgt einfach durch Eingabe von @xastir am Ende der Textdatei und Aufruf dieser mit

```
sudo nano /etc/xdg/lxsession/LXDE/autostart
```

An dieser Stelle empfehle ich zu prüfen, wie sich der Raspberry Pi nach dem Start von Xastir verhält, wenn keine Spannungsversorgung mehr anliegt. Dies wird nämlich sicherlich während des Mobilbetriebes oft passieren, wenn keine Möglichkeit zum Auslösen der Herunterfahrprozedur besteht.
Dies erzeugte bei mir keine negativen Auswirkungen. Allerdings sollte man beachten, dass Xastir beim Start eine xastir.pid Datei in /home/pi/.xastir anlegt und diese beim Beenden wieder löscht. D.h. wenn keine Spannungsversorgung mehr anliegt, wird diese Datei noch existieren und muss vor dem erneuten Start von Xastir entfernt werden. Dies wird bereits automatisch mit rm /home/pi/.xastir/xastir.pid 2>/dev/null im Soundmodemskript erledigt. Es läuft auch fehlerfrei durch, wenn die Datei nicht vorhanden ist.

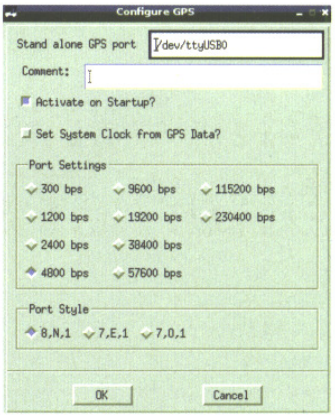


Bild 18: Schnittstellendaten der GPS-Maus

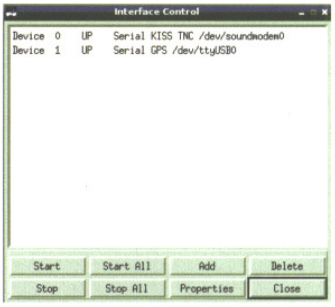


Bild 19: Aktive USB-Schnittstellen zum UM2102-Modul und zur GPS-Maus

Falls allerdings während des Hochfahrens ein Fehler beim Start von Soundmodem erfolgt, so wird die pid-Datei nicht gelöscht, sodass Xastir nicht starten wird. Auf diese Weise wird man auf den Fehler mit dem Soundmodem aufmerksam.

Fazit

Das Ergebnis ist ein kostengünstiger APRS-Tracker, der durch die Verwendung des Raspberry Pi sehr flexibel und weiter ausbaufähig ist. Daraus kann man viele Varianten mit anderen oder zusätzlichen Funktionen entwickeln.
Man kann beispielsweise durch die Einsparung eines USB-Ports, indem die VOX-Funktion zur PTT-Tastung benutzt wird, eine kleine externe USB-Soundkarte mit einem Soundeingang verwenden, um damit APRS-Daten zu empfangen und diese auf einem kleinen Monitor über den FBAS-Ausgang auszugeben. Dazu ist kein zusätzlich großer Aufwand mehr nötig. Individuelle Anpassungen der einzelnen Programme sind aufgrund der im Netz frei verfügbaren Quellcodes ebenso möglich. Bei jeder Änderung sollte man allerdings nicht vergessen, immer Sicherheitskopien zu erstellen. Empfehlenswert ist ein komplettes Image der SD-Speicherkarte. CQ DL