AndFlmsg User's Manual **For Version BETA-0.4.0**

Flmsg with integrated Fldigi modems for Android

John Douyere (VK2ETA) on 21-February-2015

Quick start

To install, download the .apk file to the device and select from a file manager. Note that "Allow unknown sources" needs to be selected in the security or application section of the device settings.

When first launched, you will be prompted for a location of the NBEMS.files folder. You can select to use the default option (typically on "sdcard") or choose another location. If choosing your own location, navigate **to inside** the desired folder then click OK.

On subsequent launches, the application will directly displays the terminal screen. Navigating to the other screens is done by a left or right swipe movement on the screen.

The menu button (earlier phones) or menu icon on the screen (tablets) brings a menu to exit the application and setup the preferences.

Before using the application on air, please set these key preferences:

a. The User's call sign (in the User Data section)
b. In the Modem section, check the audio frequency (default 1500Hz)

The standard list of forms is installed at first launch in the NBEMS.files directory. This currently includes ALL ICS and HICS forms plus Plaintext, Radiogram, IARU and Blank forms.

At subsequent App releases, the new forms can be added with the menu option "Init Forms".

To use **custom forms**, simply download and copy them in both folders "DisplayForms" and "EntryForms" under the NBEMS.files folder.

If you want to handle files manually, I recommend the ES-File explorer on Android available for free in the Playstore for moving files around from the Download directory in the device to the desired location.

The three screens are:

- Terminal (where error message and key information about message reception are displayed)
- Modem (To change modem, start/stop modem, send a tune, view the waterfall...)
- Messages screen, further divided in folders:
    - Inbox
    - Forms (for creating new messages from a given form)
    - Drafts
    - Outbox
    - Sent items
    - Logs

Important Note: that as long as AndFlmsg is running (I.e has not been "exited"), it will use the microphone as its input and therefore will prevent other apps like Skype for example to use it.

Always press Menu then Exit when you are finished with AndFlmsg.

An “Fldigi” icon is now displayed as a reminder at the top or bottom of the screen while the modem is running.

- **What is new in version B0.4.0**

  - Added DominoEx mode (NON FEC for the time being as THOR seems better than DOMINOEX + FEC at the same speed).

  - Bug fix: From Android JellyBeans(MR1), web-page buttons would often (or sometimes!) not work on devices with API 17 onwards. This prevented the saving of the form and the auto fill-in of the date and time fields.

  - Added progress indicator of percent TXed characters in title bar. Title bar example: “AndFlmsg-Transmitting: 2/3: 37% - PSKR125X10”, meaning we are currently Txing the 2nd of three messages in the Outbox and we are 37% through the TX processing of that 2nd message, using the 10 carriers PSKR125 mode.

  - Added Blank Form plus auto save for forms with checksum errors (similar to Flmsg). Bad check-sums files will be saved in the unique file: “Last_Bad_CrC_Rx”.

  - Added text display of raw data when message is not complete / corrupted.

  - Added Android support library for Notifications, action bar and soft-menu button. Notifications of the modem running are now displayed in the system bar in all versions of Android from 2.2 onwards.

  - Changed to standard style across Android versions and adjusted for consistent display between various device screen sizes.

  - Removed display of the windows titles and replaced with modern actions bar (opens-up easy access to future functions).

  - Re-instated the clipboard select/copy on terminal and modem screens (was temporarily taken off in version B0.3.0).

  - Reformatted the overflow/menu button on top action bar to match action bar height (horizontal instead of vertical).

  - Fixed Bug: if a "send all messages" process was stopped in the middle, the "Send All Msg" button did become inactive.

  - Added update mode info in title when we change mode inside an Outbox form view.

  - Sort Forms either by name (in Forms and Templete views) or by reverse date in Inbox, Drafts, Outbox and Sent views.

  - Added message Rx timeout to abort message reception. Adjustable from 1 to 10 Seconds in preferences as per Fldigi.

  - Restored back button from messages displays, but with warning and double back-key

press safety.

- Bug fix: option Modem/Modem Volume now controls modem ouput also. Was only affecting tunes volume until now.

- **Acknowledgements**

I want to thank David Freese (W1HKJ) and team for the Fldigi and Flmsg programs which code under GPL was the basis of the modem's and message processing implementation in AndFlmsg. David's vision of a GPL set of software for Amateur Radio is an inspiration.

Thank you to the NBEMS community for their ongoing feedback and support.

- **Introduction**

The main objective of this development is to provide a low power highly-portable, as in mobile or backpack, Flmsg/Fldigi solution.

Rigs like the Yaesu's FT-817, Icom's 703 and Elecraft's KX3 would be the typical companions of the Android phone or tablet running AndFlmsg, thereby providing a low weight, low energy requirement, access to most Flmsg services.

The integration of Internet sharing/forwarding makes this solution attractive as a relay option, possibly a mobile or portable station in a pertinent geographical location.

The inclusion of modes like MT-63 and long interleave MFSK modes allows audio coupling to be used effectively without the need for an audio interface.

In a more developed configuration, for example with a 10” tablet connected to a Bluetooth or USB keyboard plus mouse, and maybe also with an audio interface, the operation we get, including data input, nears the performance and practicality of a laptop configuration but without the power consumption and cost of the later.

Also, the data entry forms can be re-modeled independently of the display forms in order to match the screen real-estate of the target devices.

- **Why Android?**

The key considerations where the re-use of the graphical interface code developed for the AndPskmail application, together with the Java Native Interface (JNI) which allowed the re-use of a large section of the Fldigi modems C++ code.

Also, the plethora of built-in services in the Android OS and it's open source development kit approach makes it a very attractive platform to develop a portable Flmsg/Fldigi solution.

It is designed to run on on various devices (phones with limited screen space, tablets and Media players/TV Boxes) from Android version 2.1 onwards.

- **Typical hardware configuration**

  Android phone or tablet running Android 2.1 onwards.

  Simple audio coupling (speaker to microphone) is giving very good results.

  Both hard-wired and Bluetooth audio interfaces are described later in this manual.

  There is also to my knowledge at least one commercial interface for Android devices (Google "Wolphi-link interface").

  This is a homebrew interface similar to the Wolphi-Link (3 parts blog):
  http://waynemerry.wordpress.com/2012/08/16/designing-a-phoneradio-interface/
  http://waynemerry.wordpress.com/2012/08/22/ft-817-phone-audio-interface-part-2/
  http://waynemerry.wordpress.com/2012/11/02/ft-817-ft-897-phone-audio-interface-part-3/

  Application compatibility: has been tested on HTC EVO 4G, HTC Desire, HUAWEI Sonic, DELL Streak (Mini 5), Motorola XOOM 10.1" 3G, Medion Lifetab 10.1" 3G, HTC Velocity 4G, HTC One XL, Asus Memo Pad 7 (X86 CPU), Samsung Galaxy Tab 7 v2. It should run on any device with Android 2.1 or later.

- **What does it do?**

  Basically most of what the fully featured version on the PC does.

  - Receive Flmsg formatted messages (any Form).

  - Forward any form message ***as-is*** over the radio in Flmsg format OR over the internet via email, instant messaging (Facebook, Tweeter, ...) or cloud services (Dropbox, Google drive...), print services (E.g. Google Cloud Print).

  - Copy and edit received messages for replying or further forwarding (all custom forms plus the following "hard coded forms": Blank form, Radiogram, IARU message, Plaintext, all ICS and HICS forms at present).

  - Create new messages using any custom forms or the current set of "hard coded" forms.

  - Send created messages over the radio or internet.

  - Define and save templates for facilitating the creation of new messages.

  - Consult the activity log and the messages as-sent.

  - The implemented modems are all the FLMSG modems (including extra modes not normally listed), that is all PSK/PSKR modes, including multi-carrier modes, THOR, MFSK, OLIVIA, MT-63 and Domino-Ex modes.

- **The limitations**

  At this moment:

  - The encoding options are base 64 and base 128 only. Base 256 will be added later.
  - There is no Transfer/Receive of custom forms over the radio yet (this restriction is for the sending/receiving of the form itself, not the messages).
  - There is no Export/Import to WRAP or Q-forms files yet.
  - There is no Auto-Send button, but there is a "Send all Messages in Outbox" button which works in the background.

- **Installation**

  Download the AndFlmsg apk file from the website at:

  http://www.w1hkj.com/vk2eta/

  And save it to the SD card, either from within the phone or on a PC to which the Android device is connected to and has the USB driver installed for that device to allow access to the SD card content from the PC.

  From there use a file explorer to install the .apk file.

  You have to allow non-Android market application to be installed for this to work. This is found under Settings / Security / Unknown Sources OR Setting / Applications / Unknown Sources, for older versions of Android.

  You will be asked to accept a series of permissions that are explained below:

  a. Your location: to provide GPS time when out of cellular range (available but not displayed yet).
  b. Storage: Read and write to the SD card. For storing and accessing the working files under the "NBEMS.files" directory.
  c. Your Personnal Information (log data): to allow reporting of the context of an error in case of an application crash (the user can decide to send or not the information after reviewing it).
  d. Network Communication as in Bluetooth and Full Internet access: to allow connection to the Bluetooth handsfree kits or headsets used as audio interfaces AND to share / Forward messages over the internet.
  e. Phone calls (as in reading phone status): to disconnect the Bluetooth interface when AndFlmsg is running through the Bluetooth interface AND a phone call comes in. This is to prevent the call's audio from going to the radio.
  f. Hardware controls (audio settings and recording): required for receiving and transmitting sound.
  g. Bluetooth administration: to redirect the sound via Bluetooth.

  In your applications list you should now have an Flmsg logo called AndFlmsg.

- **Usage**

I have tried to maintain a similar philosophy of operation with the PC version. The key difference is the organization by folders like an e-mail system.

One other difference with a PC environment is the fact that the sound devices cannot be shared between applications. Therefore if AndFlmsg is running and the modem is active (which is the default when the application starts), the microphone will not be available for other applications like Skype and others.

The microphone (or Bluetooth interface if selected) will be released on either toggling of the modem OFF (in the Modem screen) OR when the application is explicitly exited from the menu.

If the application is just send in the background by calling another application, the modem will keep running and will still keep the microphone input for itself.

A notification Icon now appears in the top or bottom system status bar to remind the user when the modem of the application is ON. Clicking on the notification brings the AndFlmsg application back to the foreground. This is only available on devices with Android version 3.0 and above.

When you launch the AndFlmsg application you are presented with the Terminal screen which displays key functional messages and errors.

The other screens available are:

- Message screen, where the content of the folders inbox, forms, templates, drafts, outbox, sent and logs are displayed.

- Modem screen, where all the received and transmitted characters are shown and where the modem can be turned ON and OFF to conserve batteries. The squelch is also adjusted there and the waterfall is accessible from that screen. RSID Tx and Rx are enabled/disabled here and the modes are cycled through their list.

Movement between these screens is done by a "swipe" movement on the screen either towards the left or the right.

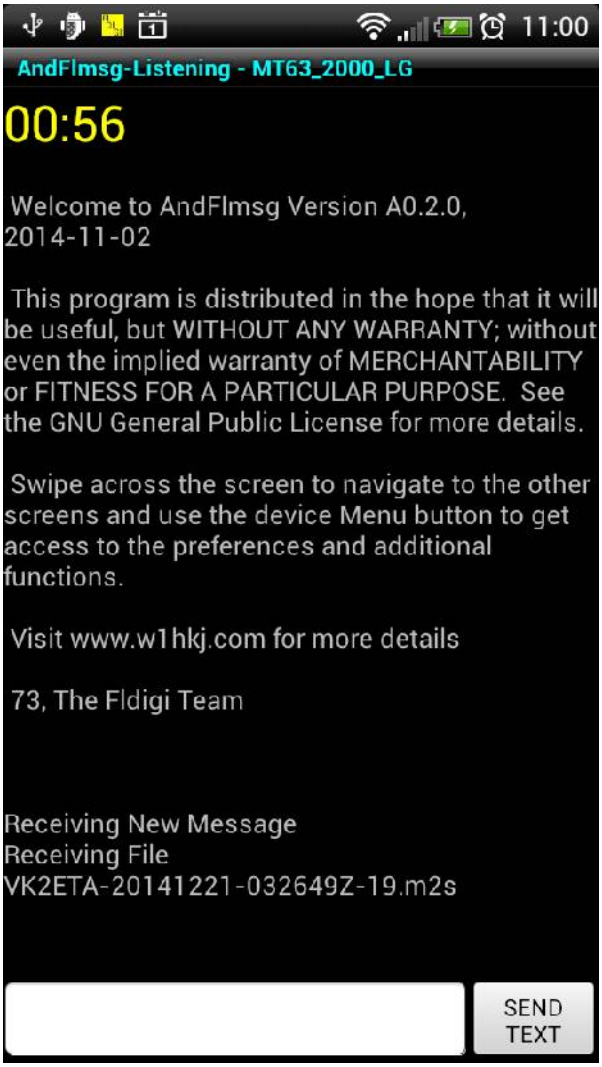
The menu is called by pressing the "Menu" button on the device (phones typically) or the "three vertical dots" screen button on tablets.
From the menu we access the exit function, the user's preferences and the switching of the Bluetooth interface on/off.

There is automatic bug reporting, but with user's control: in case of a program error causing a "Forced close" the user is presented with the option to email the stack trace (where the fault occurred) and a very brief history of the event log prior to the crash. The user can review the email and decide to sent it if happy with the content.

Screens description

1. **Terminal Screen**

Note the small yellow “FLDIGI” icon in the notification area at the top of the screen to remind the user that the modem is running. Pulling down the notifications and clicking on the FLDIGI notification brings the application back to the foreground. Only on Android version 3.0 and above.



At the top of the application window, the status line shows the session's status: “Listening”, “Transmitting” and “Modem OFF”.

In the title bar we also display the current Rx/Tx mode and the RX / TX status (Light Blue = RX, Yellow = TX, White = Modem OFF).

Below is the time, displayed as MM:SS.

This is used when working for long periods outside of cellular network coverage as the clocks on these devices tend to drift quite a lot if not resynchronised regularly.

The colour yellow indicates that only the internal clock of the device is used to calculate the current time.

If you select “Use GPS Time” in the preferences AND the GPS is active (see device “Location” settings), then the time's colour will turn green when the GPS time is used instead. This is a very accurate time.

The received text can be scrolled up and down.

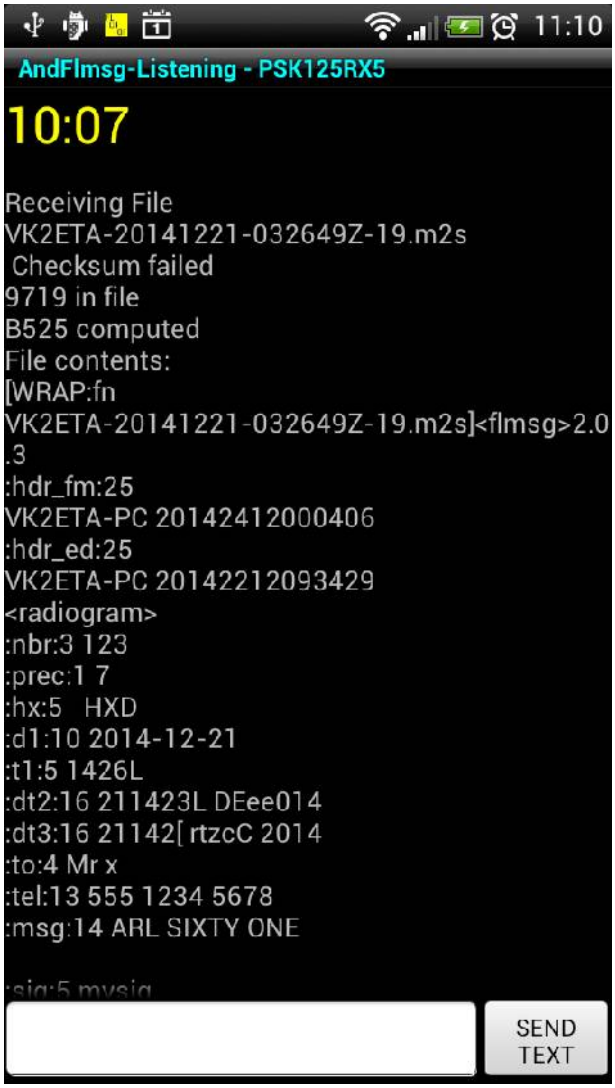
The received text can be selected for copy and paste in another application like a text editor. Just “Long Press” on the text to bring the select/copy dialogue on the screen. This is only available on devices with Android version 3.0 and above.

The single line field and the “Send Text” button are for sending short unformatted messages over the radio. Since this is not a “ragchew” or contesting PSK31 application it does not have macros and other QSO logging facilities.

When messages are being received, the details appear in the terminal, including the latest received message and if they were received correctly or not.

If the checksum of a received message is wrong, the received text will be displayed here. It can be copied by a long press on the window and pasted in a text editor if required.

Example on the right:

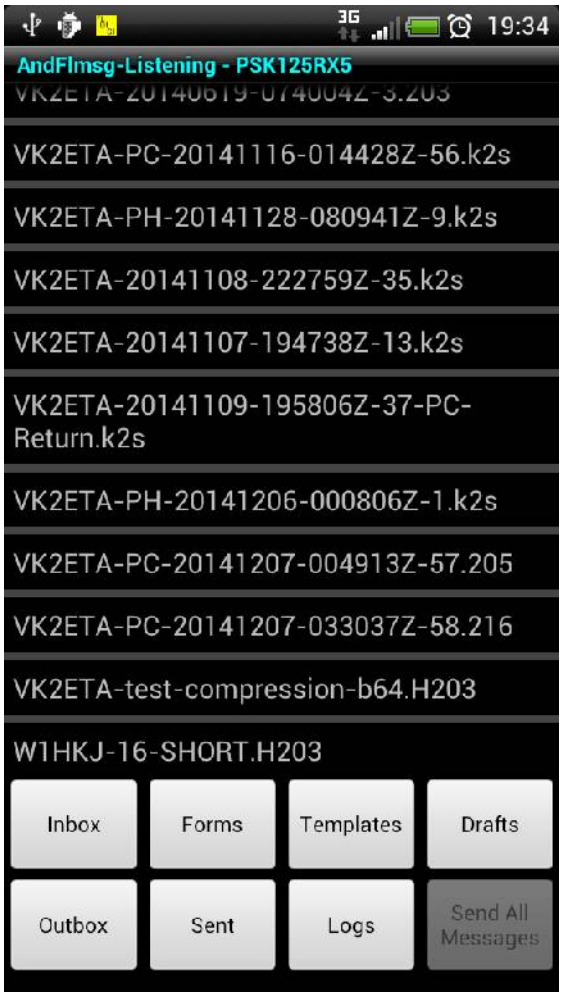


2. Message Screen

The buttons Inbox, Forms, Templates, Drafts, Outbox, Sent and Logs cycle the view between their respective folders.

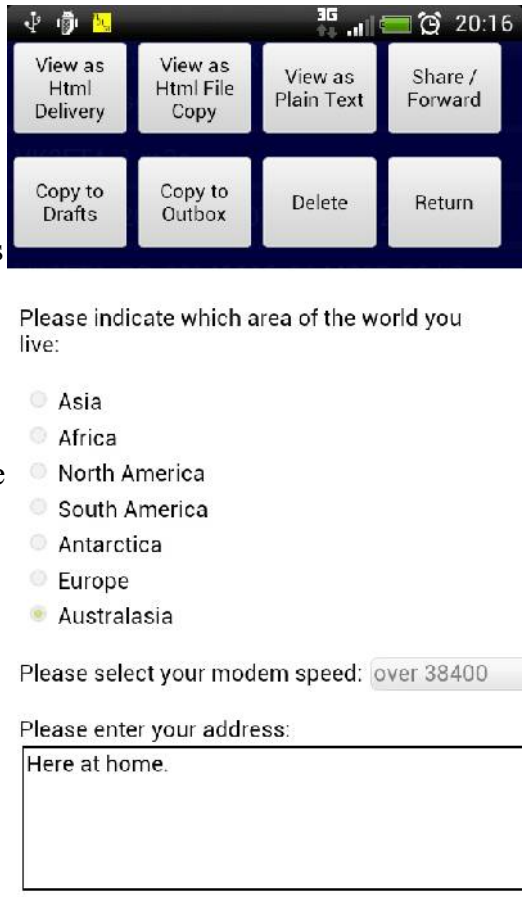
Shown here is the Inbox view which lists all received messages. The list is can be scrolled up and down.

To select an entry, long-press on the message. Once selected, the message is displayed, together with the actions available for that message (see screen copy below).



4. Inbox View, message display and associated actions:

Buttons behavior is as follow:



- View as Html, Html delivery and plan text. If the required form is not available, the raw message will be displayed and a warning issued.
- Share/Forward is for sending the message over the internet using the Android Share facility. This means any installed application that can handle the "sharing" will be proposed for the action. This includes email client, cloud services, instant messaging and file transfer facilities.
- Copy to Drafts is typically used when a message needs editing before being sent or forwarded. The file name becomes "Copy_of_" then the old file name.
- Copy to Outbox is for direct forwarding without changes. The file name is preserved as received.
- Delete will popup a verification window.
- Return is the same as ***pressing the hardware return button twice (NEW)*** and reverts to the message list.

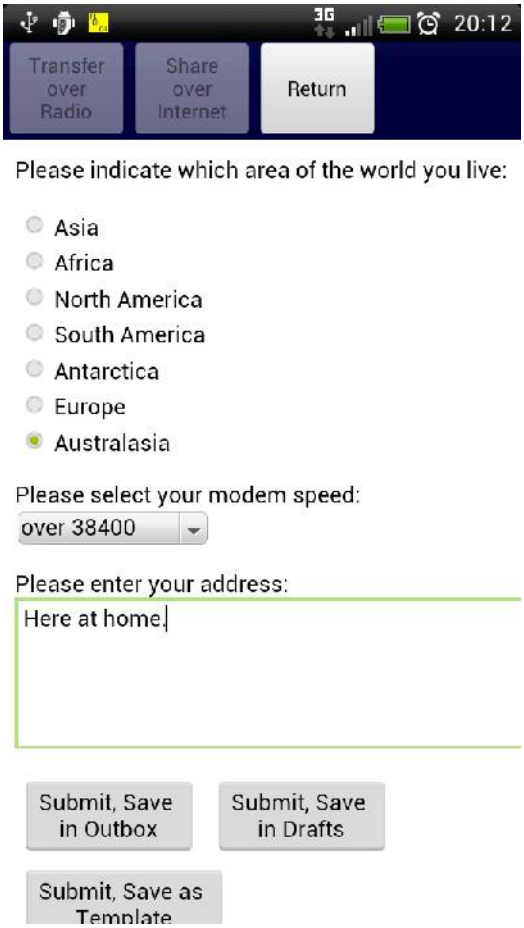
5. Forms View

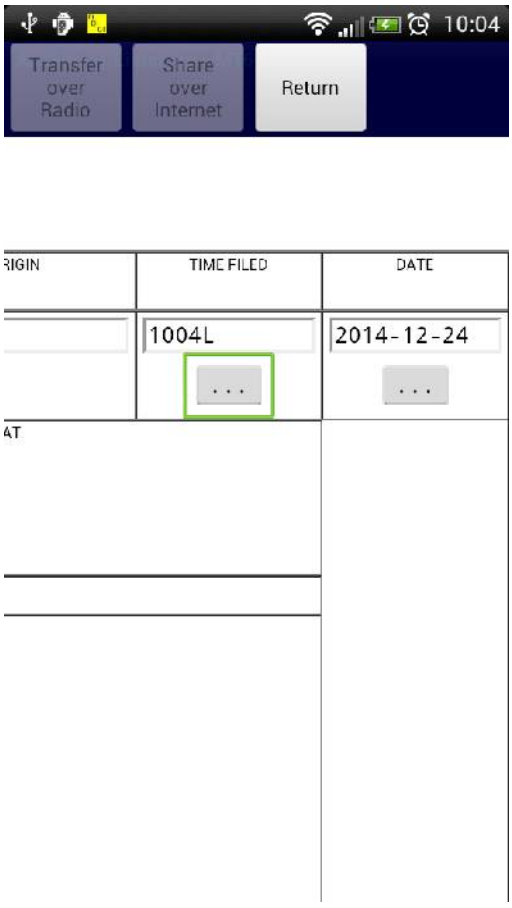
To Create a new Message using a listed form, long-press on the desired form. A new blank message will be created and will automatically display.

The “Save” buttons are located at the bottom of the form and include saving to Outbox, ready for sending, saving to Drafts for further editing and saving as as Template (in the Flmsg sense). See screen copy below left.

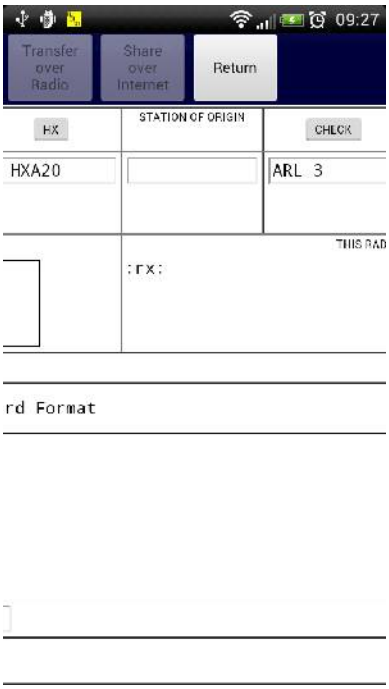
Date and time, or date/time combined fields with auto-fill buttons will use the preferences options to format the time and date (screen copy below-right).

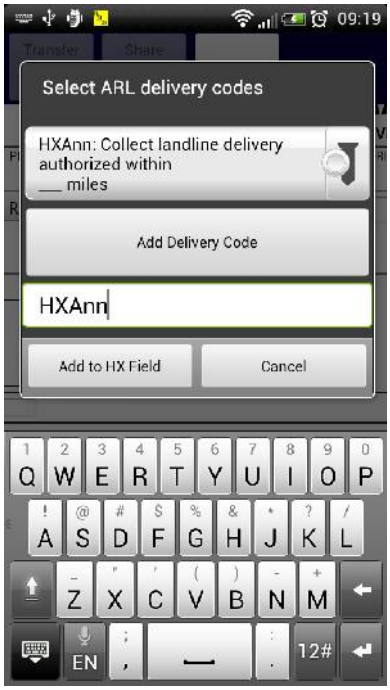
The file name is currently fixed as defined in the preferences options. When saving a message as a Template, a file name dialogue will be proposed for choosing a meaningful file name. In this case the extension is not mandatory (E.g. the Template name can be “Ics-213_for_Net123”).





Example of Radiogram below-right: form specific popups are triggered by the the buttons HX and ARL MSG. The Check button will reformat the message and update the check field.

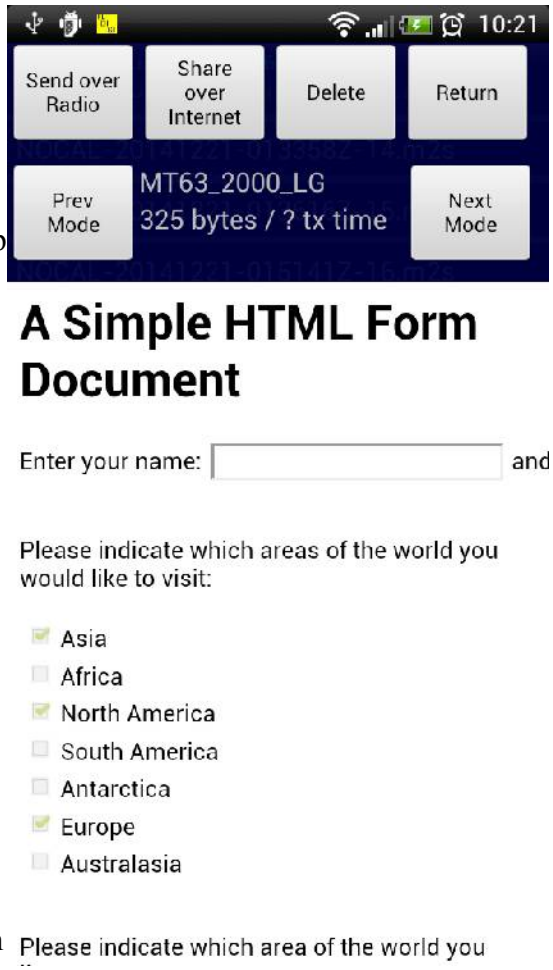




6. The **Template View** is identical to the Forms view except that saved Templates files will be displayed instead of blank forms.

7. The **Drafts View** is again similar to the Forms View with the addition of a Delete button if the draft needs to be discarded.

8. Outbox View



Note that in the main Message Screen, the "Send All Messages" button is only enabled when the Outbox list of messages is shown.

This will queue all messages in the Outbox using the current mode. If TxRSID is enabled, it will be sent before each message. The Status line at the top of the screen will display the progress of the Txing process.

When selecting a specific message, the message is displayed and the following actions are available: "Send Over Radio" sends only that message, "Share Over Internet" performs as described before, Prev and Next Mode cycles through the list of modes and, if available for that mode, will display the calculated TX time for the message based on the message content and the compression settings.

When a message is sent, it will automatically be moved to the Sent folder, UNLESS the STOP TX button in the Modem Screen was pressed before the end of transmission.

In that later case it is assumed that the transmission did not take place entirely and therefore the message is kept in the Outbox.

When the message is sent, the Outbox list is refreshed if it is the currently displayed screen.

- **The Modem Screen**

To monitor the data received and sent by the software modem and see an indicator of CPU load.

The top bar represents the CPU load with zero load on the left and 100% on the right. This is an indication of how much reserve the CPU has, rather than a true load indicator.

RX and TX RSID can be selected here. They are remembered between AndFlmsg sessions and so is the last mode used.

The bottom bar (2 overlaying bars in fact) represents both the squelch level and the current received signal quality. The brighter one is the signal quality and the darker one the squelch. The signal quality moves in a comparable way to the Fldigi green bar on the right hand side of the waterfall.

The squelch up and down increases/decreases the squelch value and stores it for later use so that it is set to the same value the next time the application starts.

The Next Mode and Prev Mode buttons will cycle either through your custom list of modes as set in the preferences OR through the whole list of modes available if the “Use custom list of modes” preferences is not set.

Modem ON/OFF stops/restarts reception and processing of incoming audio. This is to conserve processing and therefore batteries as well as freeing the microphone input for other applications.

Note: Digital Modem reception is very CPU intensive and the battery drain increases quite a lot compared with a device in standby, particularly on phones. Use the Modem On/Off button to conserve battery.

The Tune button produces a 3 seconds tune at the current set level of the modem's volume. Handy for testing SWR and tuning.

The Stop TX button is for immediate silencing of the modem in case of unwanted transmission.

The decoded RSID mode and center frequency are also displayed in the Modem screen.

The received modem data can be selected for copy and paste in another application like a text editor. Just “Long Press” on the text to bring the select/copy dialog on the screen. This is only available on devices with Android version 3.0 and above.
Note: The text can only be selected when the Modem is Off (Modem On/Off Button) as it makes the selection much easier and prevents the flickering of the screen when data is

received.

The waterfall can be displayed by pressing the W.FALL on/off button. See screen copy below.

Be aware **that the CPU load increases when displaying the waterfall and on some (old and slow) devices doing so while on multi-carrier PSKR or any of the THOR modes can lead to losses of data AND making RSID not detectable.**

Use this waterfall display sparingly on these devices.

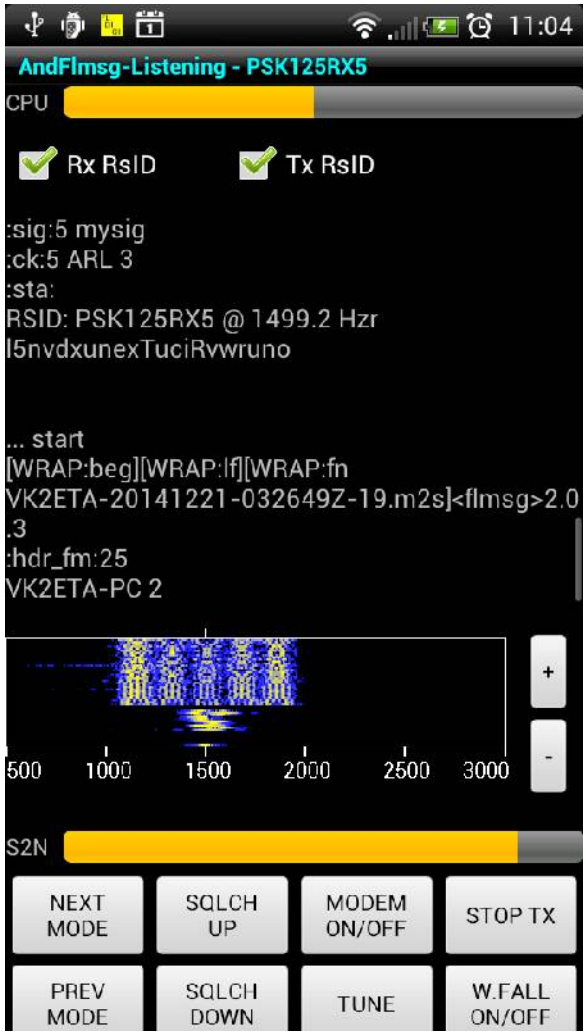
On most modern devices there is no significant load difference with or without waterfall display.

Modem screen with waterfall:

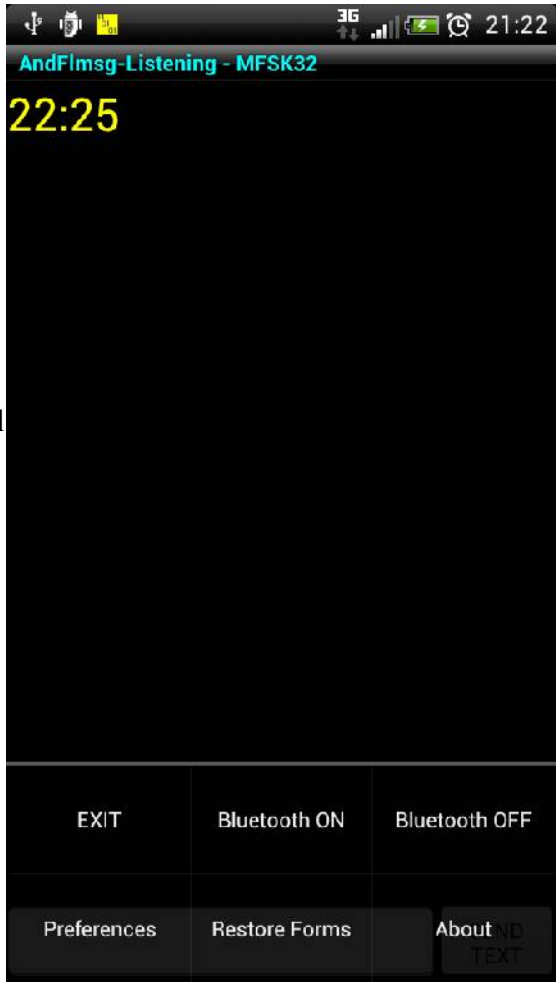
Note the tick mark on the top side of the waterfall which represents the current signal centre frequency.

It is recommended to use a custom list of modes in the preferences to make the cycling through the desired modes faster.

There is a slight delay when changing modes as the previous audio buffer needs to be cleared completely before switching mode.



- **The Navigating Menu**



Upon pressing the menu button on the device (phone) OR the menu icon on the screen (tablet), the menu is displayed as per the left screen copy here:

The **EXIT** option is used to returns to the previous displayed application or main device screen**, and stops the software modem (which is important for the power consumption of the device and for freeing the microphone for other applications)**.

It also completely removes the application from the device's memory, therefore the next launch will be from a cleanly restarted application.

This section also accesses the preferences for the application.

The Bluetooth ON must be selected AFTER the Bluetooth (headset) interface has connected to the phone or tablet. From then-on, and until the Bluetooth OFF option is selected, the audio in and out will be through the Bluetooth interface.

If you are using Android 2.2 (Froyo) and **only in that version**, there is a bug (corrected in 2.2.1) that causes the phone to reset when the Bluetooth is turned off, after having used programs that require an audio channel to a headset. This is not a bug in AndFlmsg, and the workaround is to exit AndFlmsg with the option/exit action, then kill the process using one of the many "memory cleaners" available on Android Market. I haven't found a workaround that I can include in the application.

For Android 2.3 and above there is also an automatic disconnect of the Bluetooth interface when a phone call is received. This is to prevent the phone call audio from being routed to the radio via the Bluetooth interface.

The "Init. Forms" option is to restore the entry and display forms as per the default version as supplied with the application ***plus copy any new forms*** included with the application.

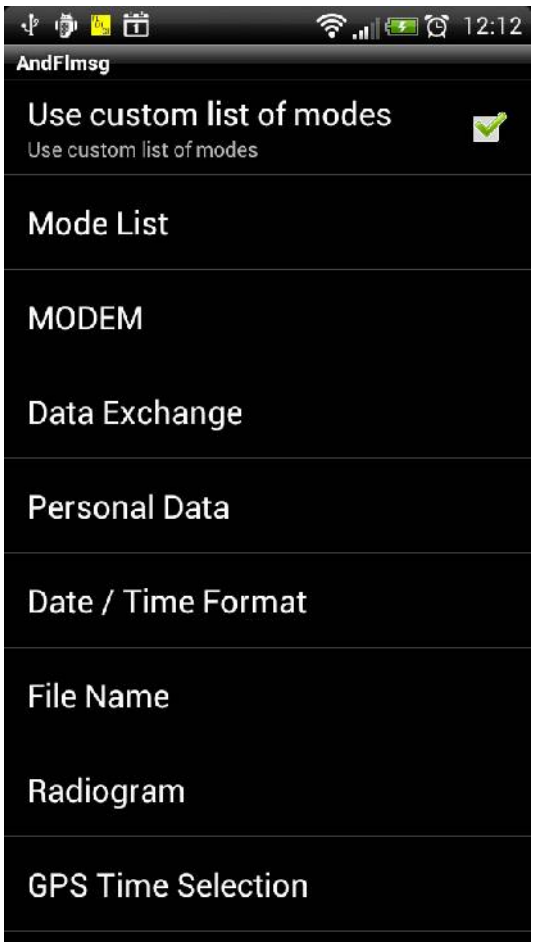
Therefore if you have modified forms on the device after installation (see next section) this will overwrite the changes.

This is also used useful to add new forms while keeping all the application setting after an application upgrade, as a full uninstall/re-install would also wipe out your settings.

If the application was uninstalled then re-installed, the program assumes it is dealing with a fresh installation and will override all existing forms with the original ones at first run.

There is also an About screen which displays the version and some information as usual.

- **The Preference screen**



Preferences are now grouped in preference groups accessed by clicking on the item.

This is a scrollable list of options. In a nutshell:

- "Use Custom list of modes" makes active the Mode List described below.
- "Mode List" brings a multi-selectable list to reduce the cycling of modes through a smaller sample of modes. Tick the modes wanted.
- Modem is a reduced version of the Fldigi options. See description below.
- Data Exchange, Personal Data, Date/Time Format, File Name and Radiogram mimic most of the Flmsg options. ***NEW: added Extract Timeout for message reception.***
- GPS time Selection is for using the GPS time when not in mobile reception area.

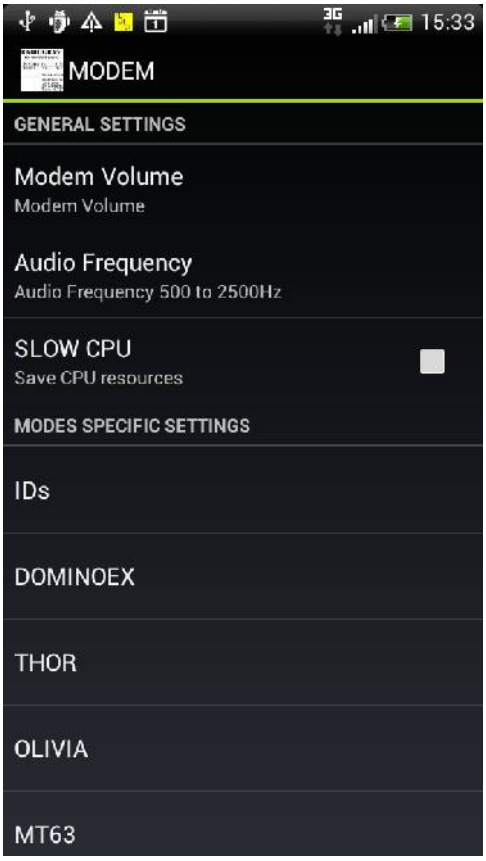
Please note that THOR modes in particular but also multi-carrier PSKR modes and MFSK Long Interleave modes can be very CPU demanding. If your device CPU load stays above 75% more than half of the time when in these modes (the top bar in the Modem screen), and you have already ticked the "slow CPU" box in the MODEM preferences, then it is better not to use them as it will result in missed characters AND possibly non-decode of incoming RSIDs.

The **MODEM** preferences are split between General setting and Modes Specific settings.

- Modem volume and audio frequency since the modem is integrated in the application. A normal value for the Volume is around 60 to 80%.

- Slow CPU selection: if your CPU load bar in the Modem screen stays above 75% select that option (only affects THOR modes at present).

Th modes specific options bring up similar choices as the Fldigi options for these modes.
Only the options relevant to Flmsg have been brought over.



Notes on “Gps Time Selection” preferences:

In this section we can elect to display the GPS time (as sent by the GPS satellites). The “Leap Seconds Correction” is normally not needed and should be left at zero as the devices GPS engines automatically correct the GPS time to bring it to the UTC time despite regular (almost yearly) increases in the leap-seconds.

On some devices (e.g. Motorola Xoom), the current leap seconds are not included, and the value 15 (as of 2014) needs to be entered in that field.

Note that the devices own clock is not changed, only the time displayed in the Terminal screen. When using Auto-Fill buttons in forms the device own clock is used and the value needs to be corrected is required.

- **Storage, Folders and Forms**

Android devices normally provide an “sdcard” storage area which can be either internal or mapped to a real external SD card if present.

The root folder NBEMS.files of AndFlmsg is created in that memory space by default, but can now be included in any other folder or sub-folder.

The sub-folders under that root folder are Inbox, Drafts, Templates, EntryForms, DisplayForms, Outbox, Sent, Temp and Logs.

The content of the folders is pretty evident, with the particularity of having display and entry forms split in two separate folders.

This has been done to cater for devices with reduced screen real-estate as it allows to have say a “standard” display form, but a customized entry form (narrower and taller for example) to facilitate data entry on smaller devices.

AndFlmsg will handle any custom form, plus the set of per-defined forms as supplied with the application.

But unlike the PC version, the entry and display forms can be customized even if they are part of the pre-defined set. The limitation to ensure compatibility with the PC version is that all the field names MUST be kept as-is (the names between the two “:” like for example “:from:”).

But the display format (E.g. number of columns per line, column width, font size, column title, etc...) can be changed while still allowing exchange with the PC version of Flmsg.

Please note that the file name of the form must be in lower case for NON-custom forms(E.g. ics213, radiogram, etc..).

In the “DisplayForms” folder there can be up to three files for the same form: xyz.html, xyz.fc.html and xyz.txt. They relate respectively to the HTML display form, the HTML File Copy form and the plain text form.

If the requested display form is not present, AndFlmsg will instead display the raw data file and alert the operator.

This is handy for using the app as a relay station as any message from any form can be relayed even if it cannot be displayed under the “standard” format on the Android device.

In the “EntryForms” folder there can be multiple files for the same custom or “hard-coded” form to accommodate different screen size or orientation preferences (as in portrait or landscape). For example we can have radiogram.html radiogram.portrait.html and radiogram.landscape.html.

The only musts are: the prefix which defines the form (E.g. “radiogram”), terminated by a dot, and the suffix made up of a dot then “html”. What is between the first and the second dot is of no significance to the application.

- **Making custom forms and modifying existing form**

Existing forms, either custom or hard-coded can be modified both for data entry and display purposes.

For data entry forms, the typical data entry field will be a one line text field which in the html form file will be something like (E.g. the “Station of Origin” field in the radiogram form):

<INPUT NAME=":org:" TYPE="text" SIZE=20 MAXLENGTH=40>

This means that (And)Flmsg will record this as the “org” field with its value being text. It can be 40 characters maximum in length (limited at time of data entry), with only 20 characters being displayed in a sliding window on screen.

A multi-line text field is described in the html file as follow:

<TEXTAREA NAME=":to:" ROWS=4 COLS=34></TEXTAREA>

In AndFlmsg there are a number of pre-defined buttons that can be associated with a field to allow faster data entry like for time and dates. Some are form specific and their usage would probably not make sense outside of the intended form like for the radiogram ARRL predefined messages.

Here is the list as their appear in the html data entry forms and an examples of their usage:

1. Combined now date and time formatted as per preferences (local/zulu etc...). Replace datetime1 by your field name in both places. Replace “...” by whatever description you want to show for the button.

   <INPUT NAME=":datetime1:" TYPE="text" SIZE=11 MAXLENGTH=20><INPUT TYPE="button" onclick="datetimeprefill(':datetime1:')" VALUE="...">

2. Now date formatted as per preferences (local/zulu etc...). Replace date1 by your field name in both places. Replace “...” by whatever description you want to show for the button.

   <INPUT NAME=":date1:" TYPE="text" SIZE=11 MAXLENGTH=20><INPUT TYPE="button" onclick="datetimeprefill(':date1:')" VALUE="...">

3. Now time formatted as per preferences (local/zulu etc...). Replace time1 by your field name in both places. Replace “...” by whatever description you want to show for the button.

   <INPUT NAME=":time1:" TYPE="text" SIZE=11 MAXLENGTH=20><INPUT TYPE="button" onclick="timeprefill(':time1:')" VALUE="...">

4. Radiogram specific button, delivery field (:hx:). Brings up a dialog popup and updates the specified field (here “:hx:”).

   <INPUT TYPE="button" onclick="arlhxdialog(':hx:')" VALUE="HX">

5. Radiogram specific button, checks the message specified in first paramter (here “:msg:”), re-format it as required. The field in the second parameter (here “:ck:”) is updated with the work “ARL” if ARRL messages are included in the first field value, plus the word count of that same field. The third parameter (here “:std:”) is the check-box field which indicates if

the message must be reformatted (changed to all capital letters and limited in words per line). If the third parameter is blank (as in open quote close quote), then it is assumed TRUE and reformatting always occurs. This is used for the IARU message form.

<INPUT TYPE="button" onclick="arlmsgcheck(':msg:',':ck:',':std:')" VALUE="CHECK">

6. Radiogram specific button. Brings up a dialog popup to select ARRL messages and their parameters if any, then updates the specified field (here ":msg:").

<INPUT TYPE="button" onclick="arlmessagesdialog(':msg:')" VALUE="ARL MSG">

- **Know bugs and quirks in version Beta 0.4.0:**

- Remember this is BETA software and is released for testing purposes mainly and although fully functional has most likely bugs, some of which that may be critical to the application.

- Bug: when using a custom form, the password fields are not always displayed correctly. I haven't worked out in which circumstances. If you work out the conditions of this bug please let me know.

- ***This seems to have been fixed with new screen handling.*** Some devices (e.g. HTC Velocity 4G) seem to have an issue with the keyboard prediction window which pops up just above the on-screen keyboard.
This means that when entering a form it may crash the application. The current work-around is to disable the word prediction in the device keyboards preferences (not a desirable solution really). Using a USB or Bluetooth keyboard resolves the issue since the built-in on-screen keyboard then remains hidden. I will investigate if there are application (AndFlmsg based) work-arounds for that device specific issue.

- **What is in the development pipeline?**
    - Add choice of format for Sharing function. Emails for example could have either Html formatted message or the Flmsg formatted file message attached. Maybe PDF format as well.

    - When calculating the transmission time of a message, will add the single or double RSID TX time to the calculated time.

    - More Hard-Coded forms as required or modified forms for small screen devices (phones), more modems (E.G. 8PSK modes).

    - Add Forms transfer (as in html file) over radio/share.

    - When using Share function, add option for sending raw data file (and maybe PDF) together with/instead of just HTML formatted message.

    - Add Base256 to compression option/processing.

    - Have a "very slow CPU" option for PSKR, Thor and maybe MFSK (solution from AndPskmail java modems) for older/cheaper platforms.

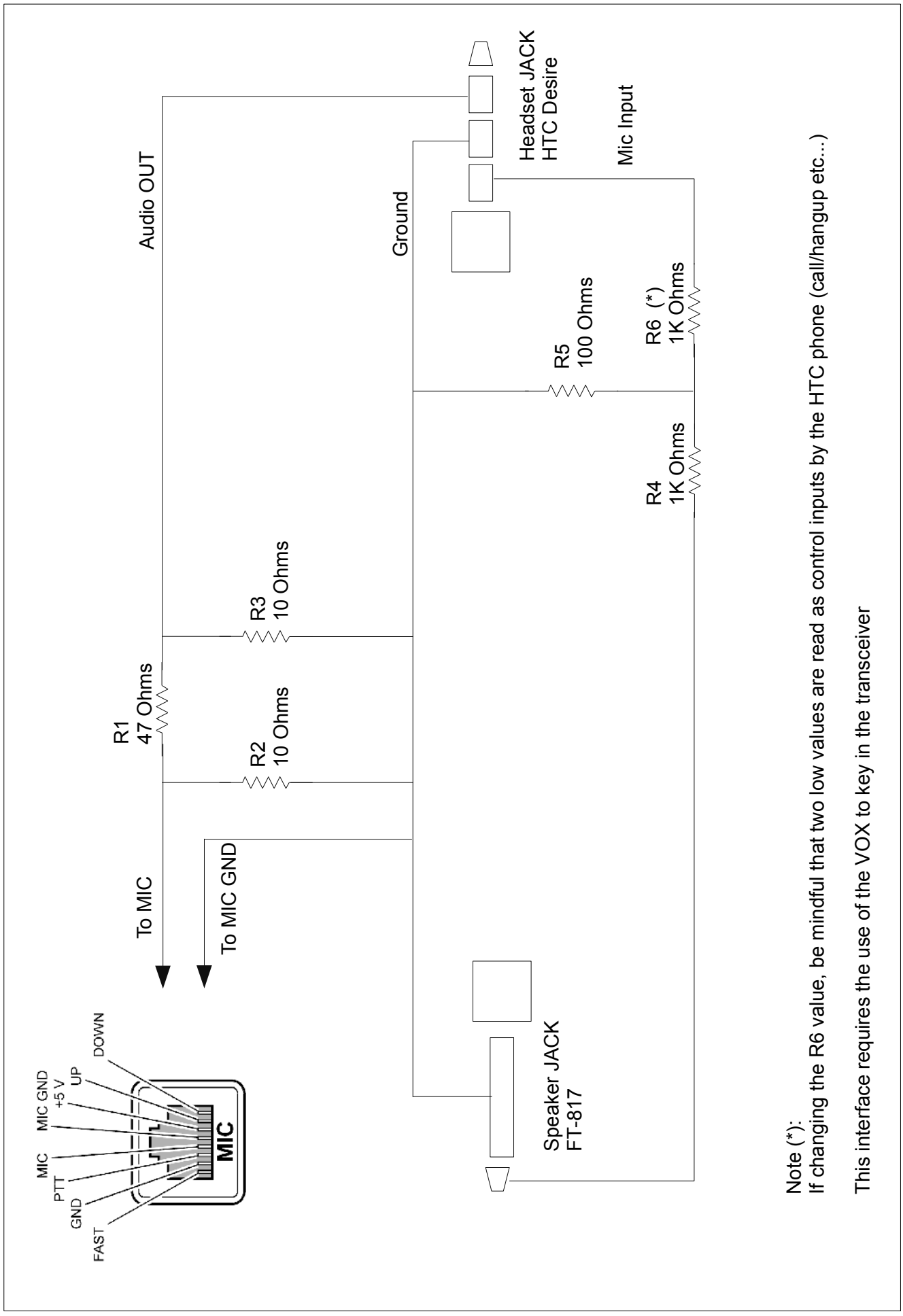
    - Import the AndPskmail adjustable TX tune option which including toggle ON/OFF of indefinite time.

    - That's all for now :)

Enjoy, 73 John (VK2ETA)

P.S: Circuit diagram for a simple audio cable and two Bluetooth interfaces below.

Simple audio interface cable for FT-817 (857/897/450)



Using a Bluetooth HandsFree car-kit Adapter with Android Phones running AndFlmsg

The price ranges from $15 to around $250 for the top of range units. Low cost units seem to work well.

By removing the internal speaker I was able to fit all the signal level adaptation and the VOX inside the unit rather than have a separate box, but that is a personal choice as keeping the loud-speaker can provide extra feedback.

These units typically clips on the sun-visor of the car and have a built-in rechargeable battery giving around 10 hours of run time, are small (approx in cm 10 x 5 x 1.7 or in inches: 4 x 2 x 0.7) and light weight (around 100g originally).

The minimal device specifications are that it handles either the hands-free or the headset Bluetooth profile. The A2DP profile is of no use in that case as it is for one way high resolution stereo audio.

The thicker the unit, the easier it will be to fit the electronics inside.

The model I used is an Advantalk Nova, but I am sure a lot of other units would fit the bill. The key criterion is that these units pair with your phone or tablet and can be used in a phone call as interface for audio in and out.

Required adaptation:

I basically left all the electronics of the units as is which is a plus as it is all small surface mount components. Only the integrated speaker and microphone were removed.

The two functions required to be added are signal level and impedance adaptation for the input and output to/from the transceiver and the VOX circuit if necessary.

It is worth noting that most modern radios integrate a digital mode VOX and therefore only need the signal adaptation. But for portable work with the FT-817 I wanted both so that I could use the data plug at the back of the unit.

Luckily there is ample signal strength to allow simple resistive circuits for the signal adaptation both way.

I wanted the VOX circuit to be simple and not pull energy from the battery of the unit or the transceiver.

The original audio output circuit of the Bluetooth unit is based on a H circuit as there is not much supply voltage to go around at 3.7V (single cell integrated lithium battery) and therefore it is not referenced to ground. So I used a transformer to isolate it and at the same time boost the voltage so that the VOX has ample voltage to work with.

A small output transformer normally used on the final stage of a low power audio amplifier is used for that purpose. It has an 8 Ohms secondary and a 2 x 500 Ohms primary. I use it “in reverse” so that I can get 10-15Vpep on the secondary at full volume. Ample to trigger the VOX circuit even at low volume settings which is a great plus.

A few decoupling capacitors are required for signal cleanliness and stability.

I added for good measure some ferrite sleeves on the short cable between the unit and the radio just in case.

The circuit diagram is shown with the pictures further down. Please note that it can be adapted to most other rigs quite easily.

Operation:

Only on devices running Android 2.1 (skip this step if you have Android 2.2 and above): download and install the program “testinband.apk” found at Google code.

Link: http://code.google.com/p/android/issues/detail?id=7906 search for “testinband.apk”

Every time you want to use the Bluetooth interface, perform in that sequence:
Start Bluetooth on the phone
Assuming the handsfree kit was paired before, start the handsfree kit and ensure it is connected (This is typically done by waiting 10 to 30 seconds or by a short press on the multifunction button of the handsfree kit).
If you are using Android 2.1, launch the “testinband” application.

For all versions of Android after 2.1, start here:

Launch AndFlmsg
Call the menu in AndFlmsg, select “Bluetooth ON”
Ensure that the modem's volume in the Preferences is at 60 to 80%.
From now on, the audio received should be from the radio and not the microphone, and the sound output should be redirected towards the transceiver
To stop redirecting the audio to the Bluetooth and return to the speaker/Audio plug, select “Bluetooth OFF” from the AndFlmsg menu.

Note that the Bluetooth volume need to be around 50% and above to provide enough signal output voltage to trigger the VOX (dependent on your VOX circuit and Bluetooth unit).

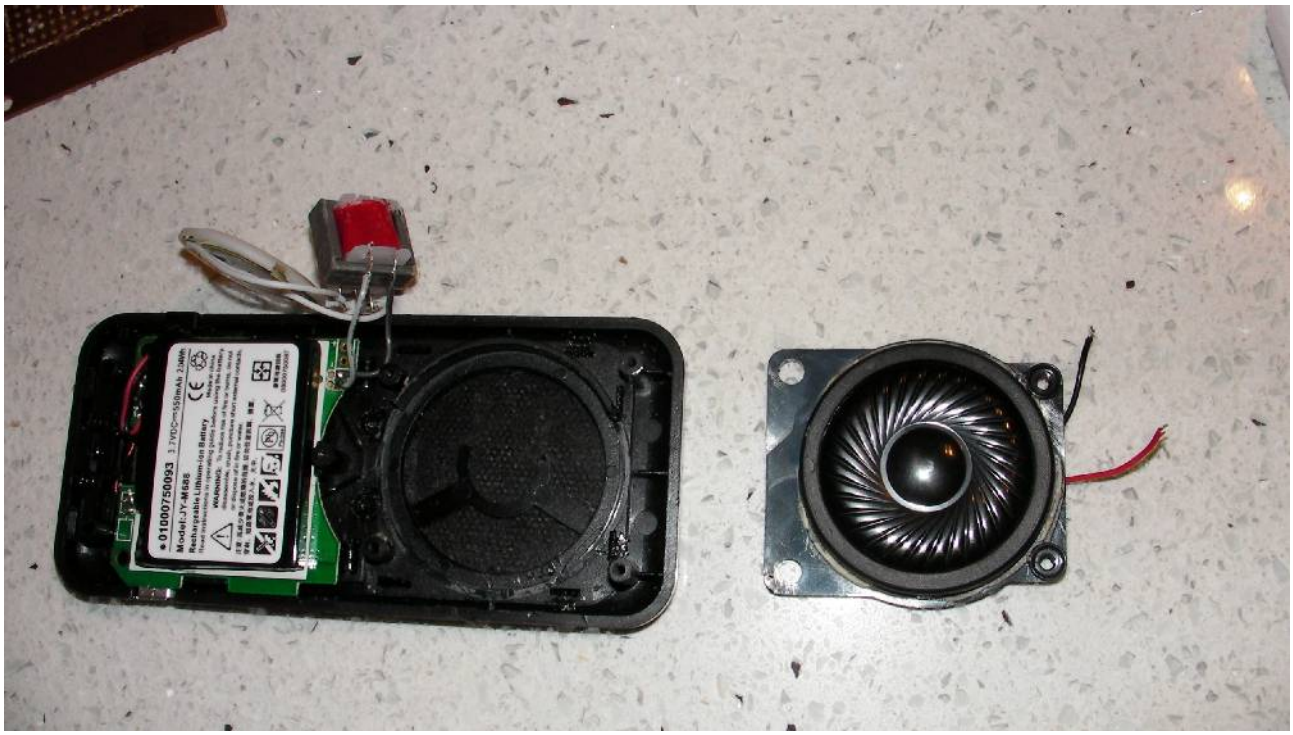
Adjusting the volume up and down on the Bluetooth unit or on the phone should display a “Bluetooth volume” bar on the phone.

Also I noted on my phone that the media playback level also had an impact, although not very large, on the final output volume.

So in summary there is: the modem volume in the AndFlmsg preferences (normally set and left at 50 to 80%), the Bluetooth volume (adjustable when the Bluetooth device is connected from the handsfree kit or the phone) and the media volume (to a lesser extent).

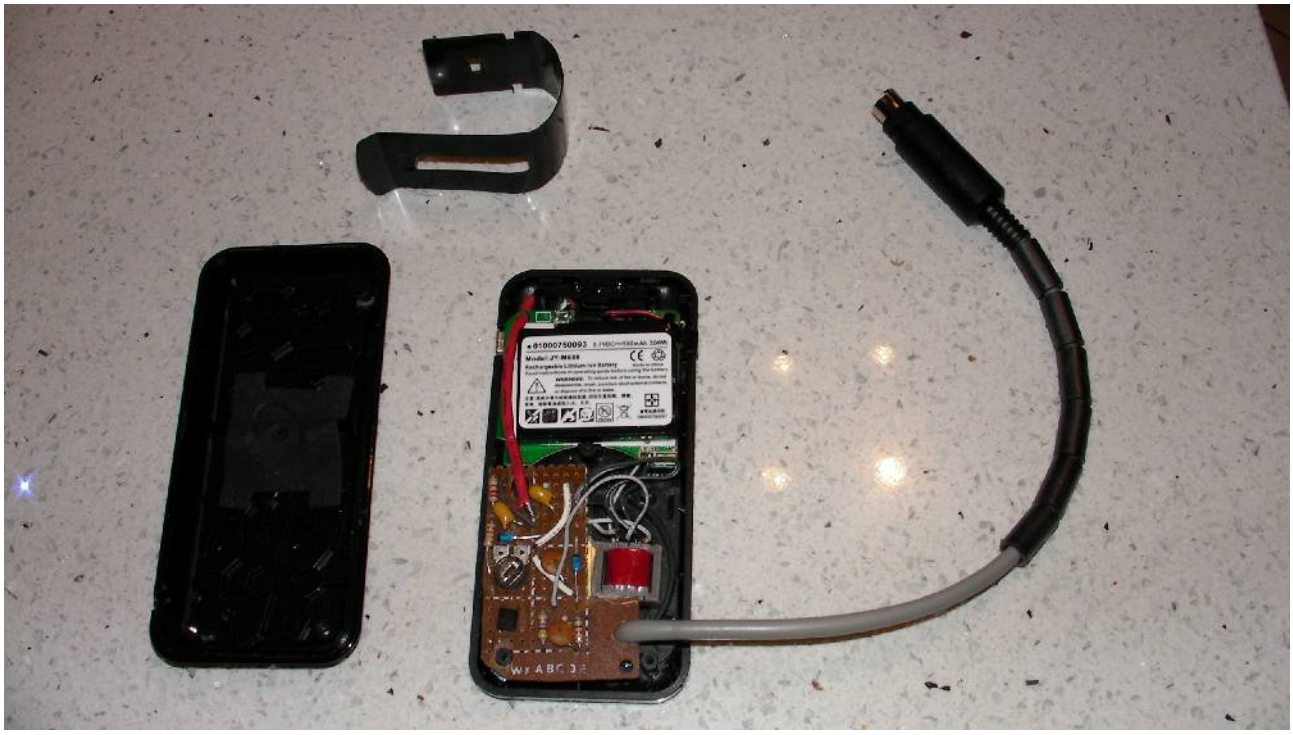
Pictures and schematics:

Original unit, opened, with the internal speaker removed, transformer soldered.



Veroboard cut to shape. Note the usage of a very small speaker that was removed after initial testing.

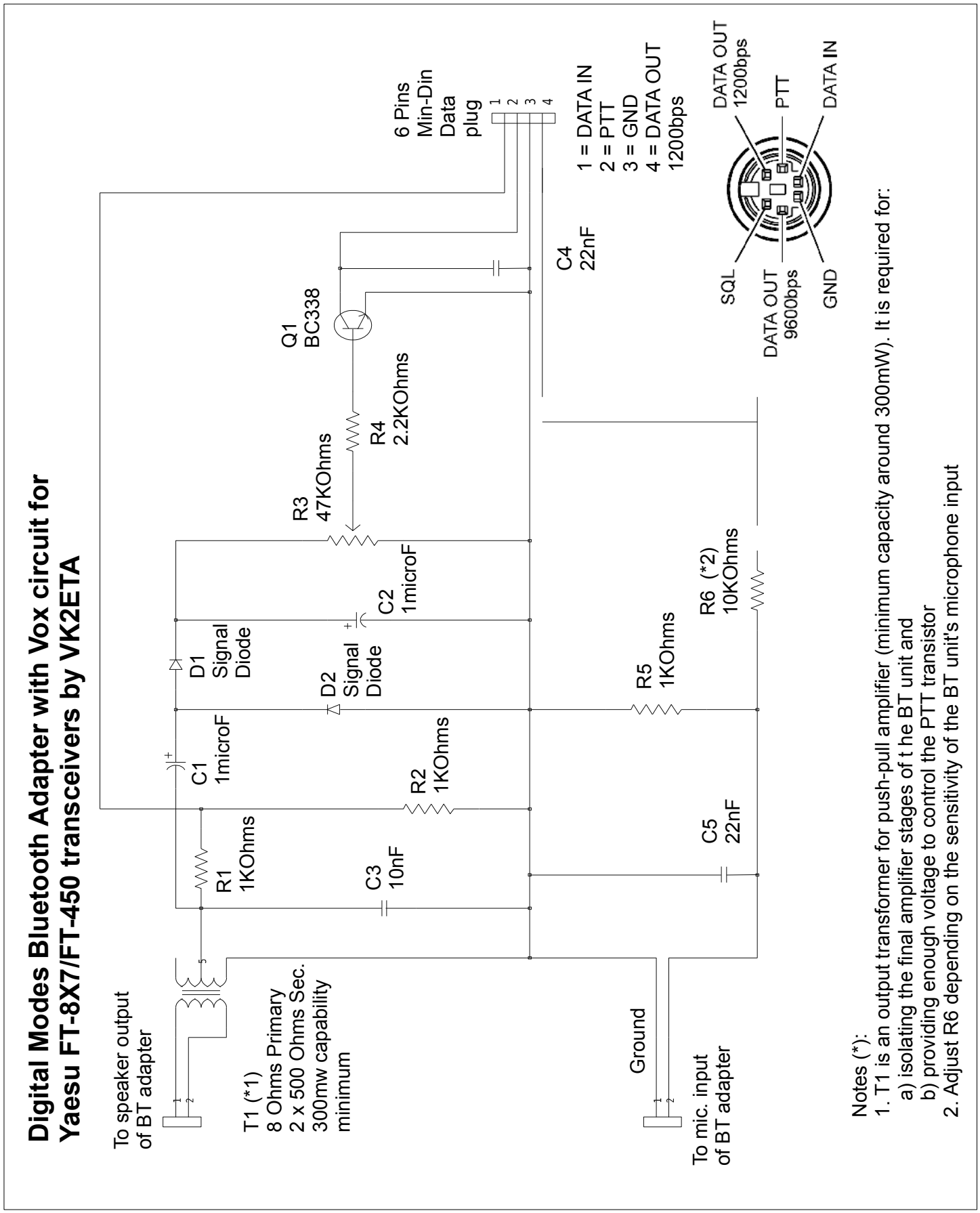
Unit wired ready to plug in the transceiver. Note the six ferrite sleeves on the audio cable to the TRX.



Connected, clipped on the FT-817

The whole portable set-up: 41M wire and tuner for end-fed Fuchs antenna, FT-817 with Bluetooth adapter, Android phone, 5Ah battery in sealed box.

Circuit Diagram: note that the VOX circuit is only required for the FT-817 as the other units have a Digital Modes VOX that triggers the PTT on reception of an audio input.



Mini Bluetooth Adapter

The headset is a cheap eBay headset at around $10 delivered, but I strongly suspect most version would work as well. In my case the total cost of the project comes to around $13 - $15.

The circuit is very simple, consisting of four resistors and one capacitor.

For the FT-8x7 series of transceivers I used the end of an Ethernet network cable and simply attached the Bluetooth headset at the end of the plug. The arrangement is given mechanical strength with a blob of Polymorph (Polycaprolactone is it's real name) or any other potting solution which encapsulates the 5 components located between the plug and the headset.

Required adaptation:

Once I opened the Bluetooth headset, I basically de-soldered and removed the microphone and the earpiece speaker. I then extended the wires through where the earpiece speaker was, and closed the unit back.

I then simply wired the five components from the microphone plug to the headset and enclosed the lot in a blob as mentioned above.

The circuit is shown at the bottom of this document.

Operation is identical to the Bluetooh handsfree car-kit described above.

Results:

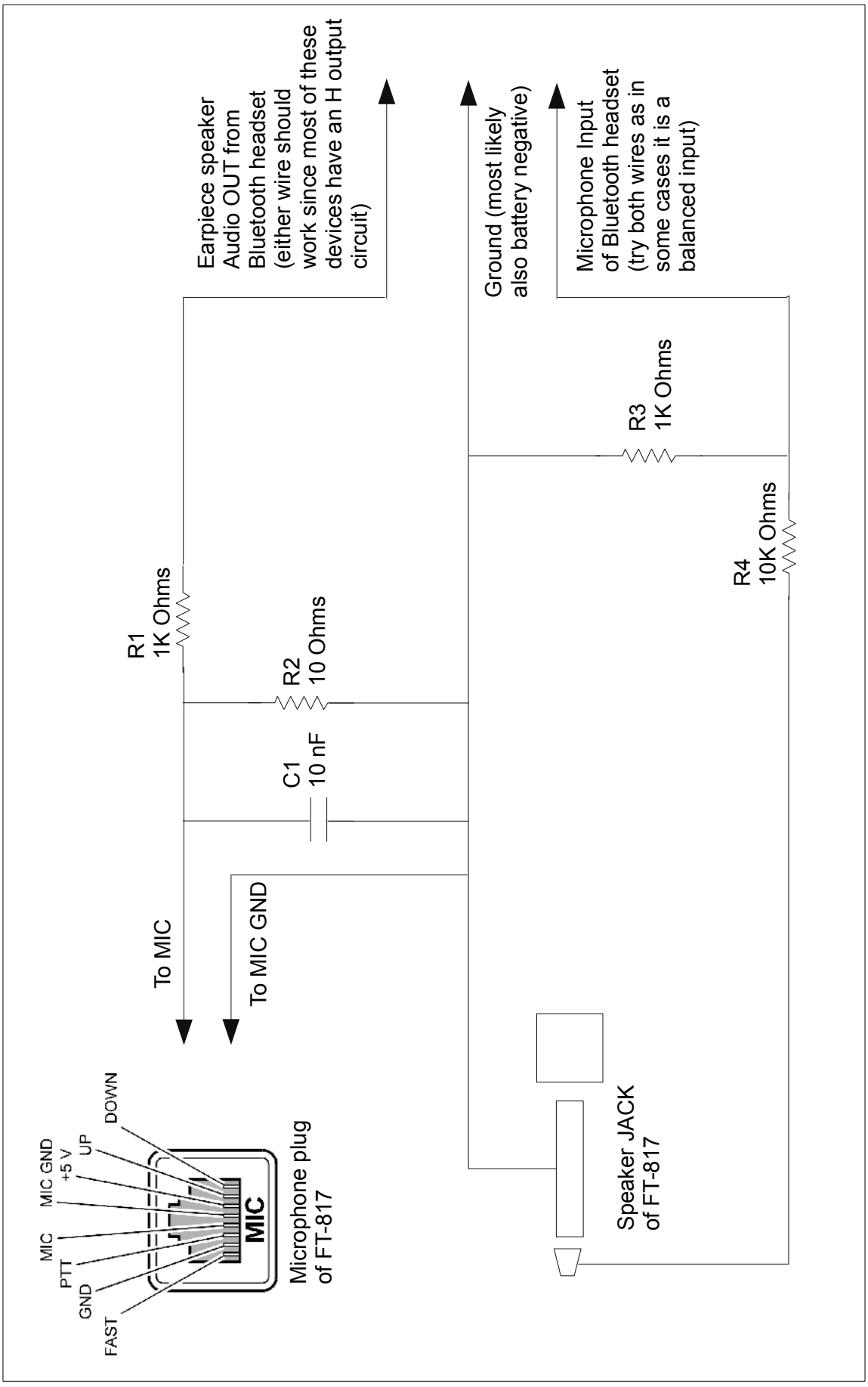
They have been excellent and in fact from a radio to Android device direction, better than the larger car kit unit simply because there does not seem to be any noise reduction function in these little (cheap) headsets. So the signal quality is excellent both ways. I get the receiving Fldigi reporting between -36 and -30dB of IMD in the PSK modes which is very good for an on-air test.

No hash or birdies were detected in my initial tests.

Comparison with the Handsfree Car Kit version

Advantages and disadvantages of the mini version:
+ Size and weight of course.
+ Simpler circuit (although the NO-VOX circuit here can be used with a handsfree car kit too).
+ Better audio towards the phone/tablet since there is no noise reduction applied to the signal.
+ Reduced power consumption (not measured but implied due to the size of the batteries and the comparative talk times)
- Reduced “talk time”: 3-4Hours versus 10 hours.
- Does not allow charging while being turned ON.
- Cuts the speaker out unless a splitter plug is used.

Schematic:


Earpiece speaker
Audio OUT from
Bluetooth headset
(either wire should
work since most of these
devices have an H output
circuit)
Ground (most likely
also battery negative)
Microphone Input
of Bluetooth headset
(try both wires as in
some cases it is a
balanced input)
R1
1K Ohms
R2
10 Ohms
C1
10 nF
R3
1K Ohms
R4
10K Ohms
To MIC
To MIC GND
Speaker JACK
of FT-817
Microphone plug
of FT-817
MIC
MIC GND
+5 V
UP
DOWN
PTT
GND
FAST