FrSky Telemetry details:

To run the Frsky build of er9x you will need to make some modifications to the hardware in your radio. These modifications require soldering of small tracks on the radio circuit board. Be warned that it is possible that damage the board permanently if care is not taken.

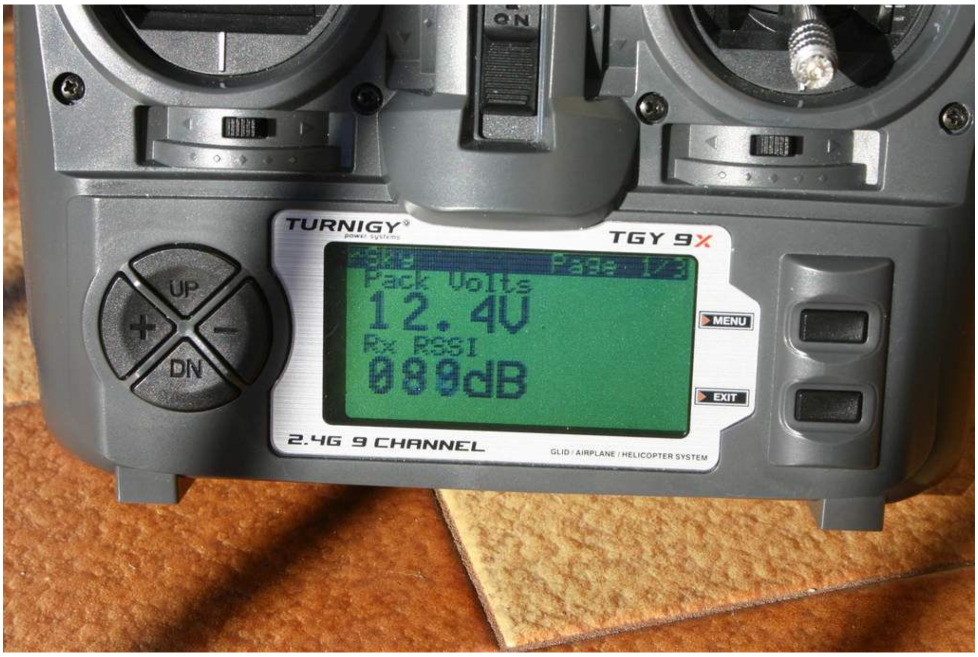
Gruvin has written an excellent description of the modifications needed here: http://code.google.com/p/gruvin9x/wiki/FrskyInterfacing so I won't repeat it in this document. Note: I believe that the emitter and collector of the BC857 (Q3) are reversed on Gruvin's diagram. The emitter should connect to Vcc and the collector should connect to R75 and R76.

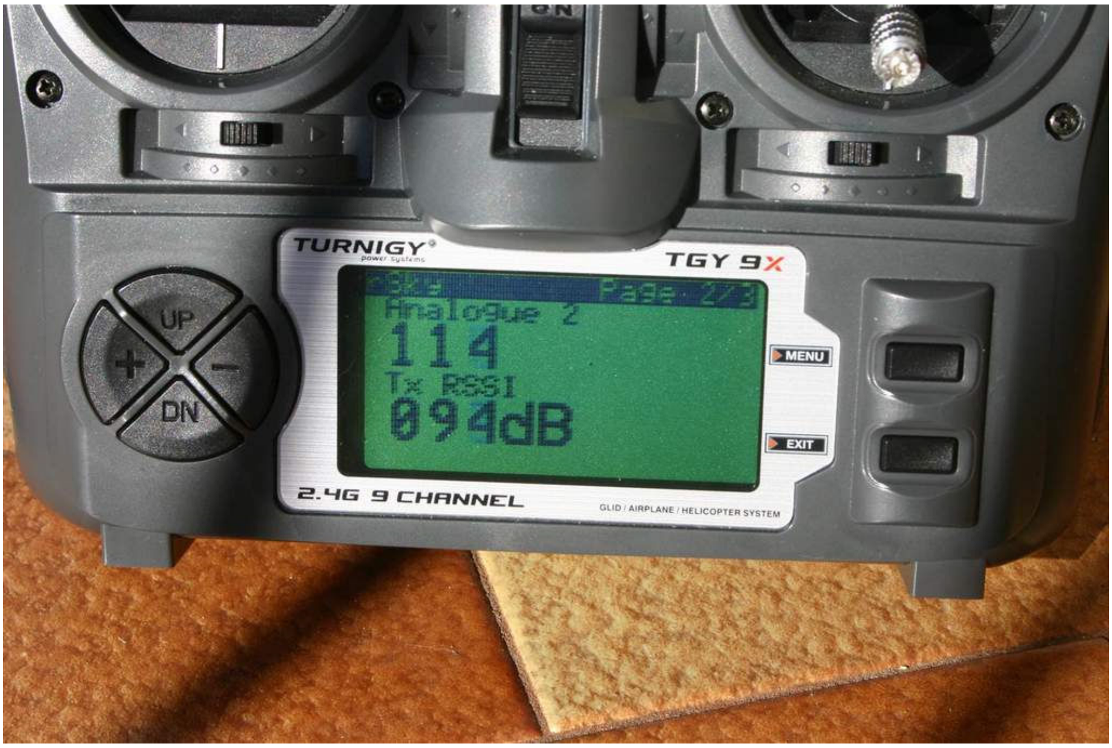
An alternative to the two transistor circuit is to use a Maxim MAX232 IC. This is the approach I have taken. It simply requires the IC and four capacitors.

Nb. Gruvin is actually writing a side branch of er9x called gruvin9x specifically dedicated to FrSky telemetry. If you are serious about getting more in depth with the FrSky telemetry functions you might want to try it out. This er9x build provides basic telemetry support and has the beauty of incorporating all of Erazz's changes and bug fixes as he makes them. Gruvin has now deviated from er9x a bit so the er9x functions may take a while to make it to his code.

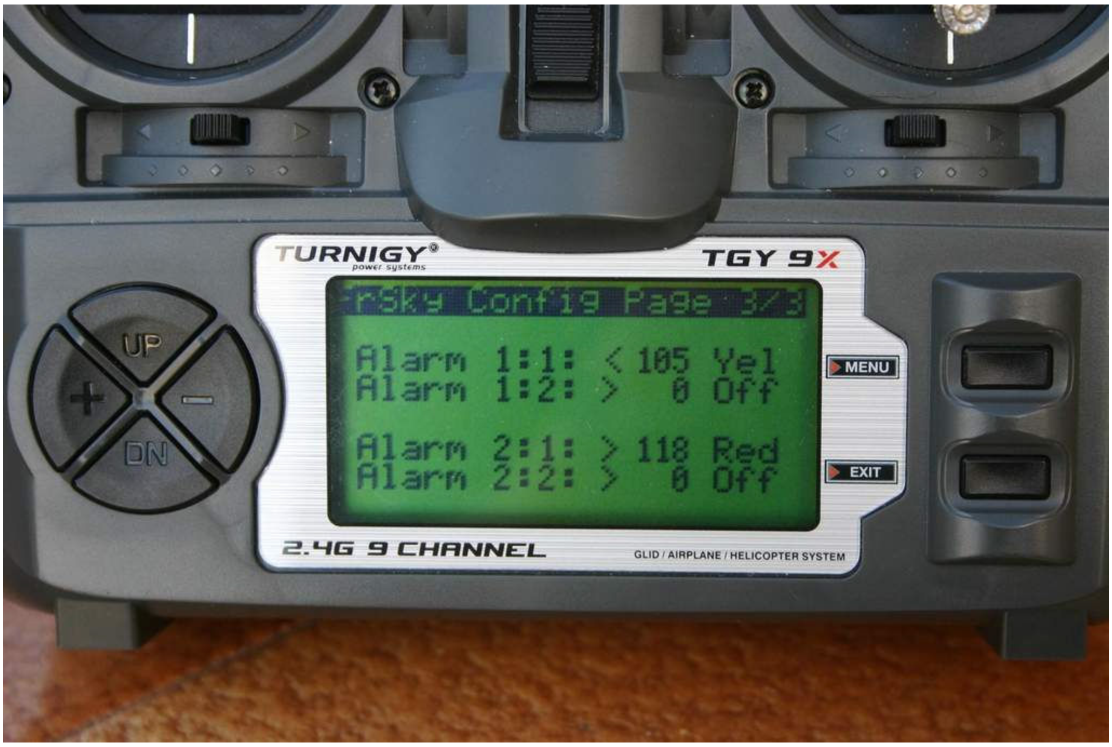
Currently there are 3 screens (see pictures below), the first displays the battery voltage connected to A1 and Rx RSSI. The second has a decimal representation of the analog reading on A2 (0-254) and the Tx RSSI, and the third is a configuration screen for setting the built in alarm points for the Tx module.

To access the FrSky screens, from the main er9x display press and hold the down arrow button. From there you can scroll through the screens by using the up and down buttons.





```
rSky           Page 2/3
Analogue 2
114
Tx RSSI
094dB
```



```
rSky Config Page 3/3

Alarm 1:1: < 105 Yel
Alarm 1:2: >   0 Off

Alarm 2:1: > 118 Red
Alarm 2:2: >   0 Off
```

The config screen:

In the config screen, pressing menu will allow editing of the fields. To save changes, navigate to the bottom of the config screen until 'save' is highlighted and press menu again. To discard changes, simply long-press end to return to the er9x main screen or press menu with any other field highlighted to turn off edit mode. Doing this, your changes will still be temporarily shown in the config screen but will not be saved.

What the display means:

Take the first line, Alarm 1:1: means the first alarm for the A1 analog input (Alarm 1:2 is the second alarm for A1).

The 105 is the alarm point, as my A1 is a voltage sensor cable this means my alarm is set to 10.5V. The < sign means the alarm goes off if the reading is below 105.

The last field is the alarm level. As per FrSky's documentation there are three alarm levels: Red Orange and Yellow. This corresponds to how many beeps the alarm will sound. Red is 3 beeps, Orange is 2 beeps and Yellow is One beep.

Current Limitations:
(Simple):
The voltage reading at the moment requires the voltage sensor cable described here: http://www.rcgroups.com/forums/showpost.php?p=16889744&postcount=2403 to be connected to the A1 input. This is not possible with the V2 receiver without performing a modification on the receiver as described here: http://www.rcgroups.com/forums/showpost.php?p=16900284&postcount=2441 . It is something I am working on and will hopefully work around soon.

(Technical):
As far as voltage readings go, the code was written around a V1 receiver without the built-in sensor on A1 and was configured to use a very convenient resistor ratio giving you a full scale reading of 25.4V. This meant that one type of voltage sensor cable could be used for anything up to 6s lipo and give an accurate reading. It also meant that it did not need any scaling in the firmware code.

FrSky have messed things up a little by hard wiring the voltage sensor in the V2. For one it is useless to anyone using a BEC and second, the ratio is different to the one I have coded for. The solution is to introduce a setting in the config menu to set the scaling for the voltage sensor. It will require some changes to the way the voltage is currently displayed and I will need to use EEPROM to be able to save the value. I'll work with Erez to get this working as soon as I can.