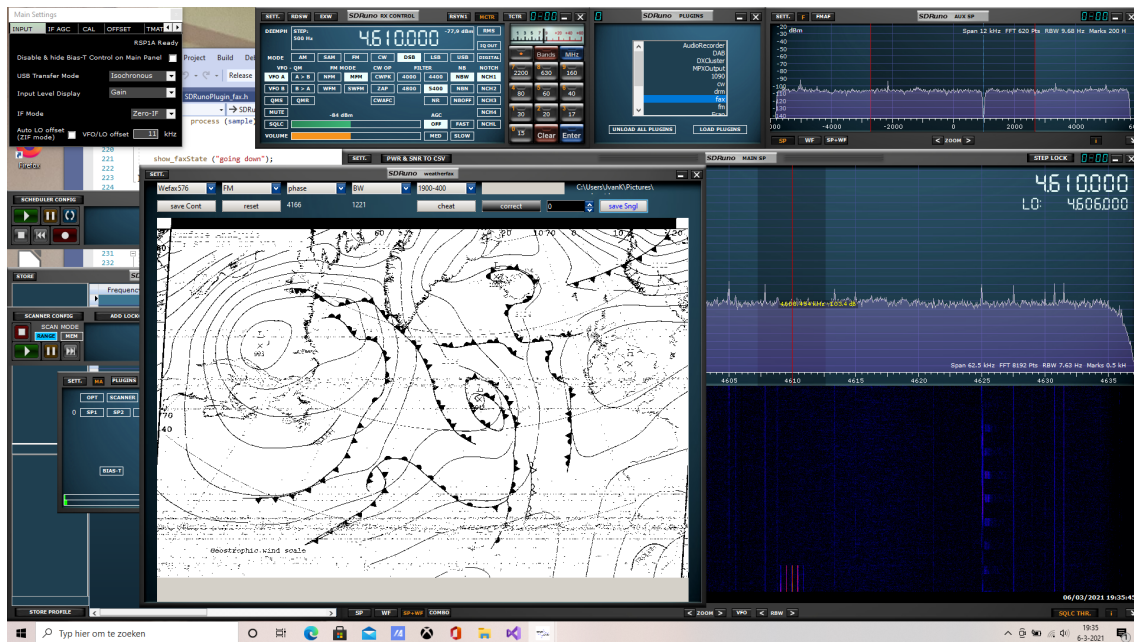


A simple weatherfax plugin for SDRuno

Jan van Katwijk
Lazy Chair Computing
The Netherlands
J.vanKatwijk@gmail.com

September 19, 2022



1 Introduction

The SDRuno weatherfax plugin is a simple plugin to decode weatherfax signals.

On shortwave (between 3 and 16 Mhz) there are transmissions of weatherfax charts on a regular base. Here, in western Europe, 3588 KHz, 4610 and 7880 KHz are frequencies where (almost) continuously weather charts are transmitted that can be received (where I live people are extremely climate aware, so the number of solar panels is overwhelming, and so is the amount of jamming signals, from time to time making it virtually impossible to receive a chart noise free).

The most common format for transmitting weather charts is *Wefax576*, a format with an IOC (Index of Cooperation) of 576 and 1200 lines charts. The so-called IOC results in a width of the chart of app 1800 pixels (the plugin reduces this to 900 pixels).

Transmission speed is 2 lines a minute, a chart has 1200 lines so, with some header information preceding the chart, transmission of a single chart takes more than 10 minutes.

Modulation of the signal is by phase shifting, with a signal deviation of app +400 Hz for white, and -400 Hz for black.

A transmission starts with a predefined signal, for Wefax576 a signal at precise 300 Hz, followed by a number of *phase lines* with which the receiver can synchronize with the transmission. Such a *phase line* starts and ends with 2.5 % of the linelength with a pure white signal, and 95 % of the linelength with pure black (the picture of the chart shows in the top lines just a few of these phase lines).

At the end of the transmission of a weatherchart, again, a tone with predefined frequency is transmitted, just for a few seconds. For Wefax576 this tone is at 450 Hz.

2 SDRuno settings

2.1 Setting the samplerate

The implementation uses a sample frequency of *12 KHz*, similar to e.g. the navtex and the rtty plugin.

This implementation select the so-called *IQ-OUT* option, so decoding basically uses the unprocessed input, down-sampled to 192000. One should realize that the SDRuno spectrum display shows default a wide band, the advantage is that one sees a lot of signals, the disadvantage is that precise tuning, based on the view on the spectrum is not easy.

The plugin generates an audiotone of 800 Hz + the tuning offset, so, when a transmission is active, two tones are heard, one for white and one for black.

2.2 Setting the frequency

Transmission frequencies for weatherfax are predefined, so just select a frequency from one of the lists of frequencies. In

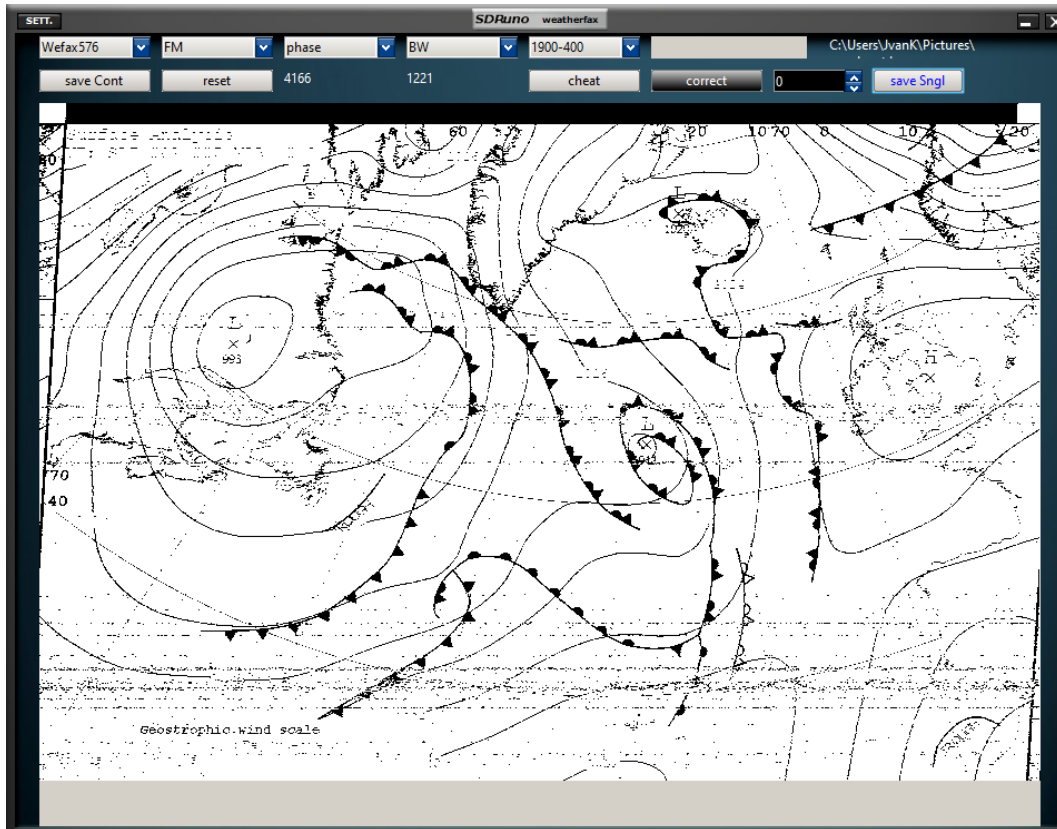
<https://www.weather.gov/media/marine/rfax.pdf>

one finds a list of stations.

Note that - different from some other implementations - tuning is *precisely* on the specified frequency, not 1900 Hz less!

3 The plugin

The widget for the plugin is shown in the picture below



The widget is large, the weather chart is displayed on it. The size of the widget is such that a wefax576 chart is shown on precisely one quarter of its size (app 900 pixels wide, 600 lines).

The top two lines of the widget are reserved for the controls, the *top one* contains (from left to right)

- a selector for the kind of charts, default Wefax576, alternatively (but untested) Wefax288;
- the *modulation mode*, default FM, alternatively AM;
- the *phase*, by default the higher frequency of the signal is used for *white*, the lower for *black*. Selecting "invers" reverses this interpretation;
- *BW*, a choice between Black and white, gray and color (only black-white is used in the weathercharts and other settings are not tested);
- the deviation. While in Europe the deviation of the modulation is 400 Hz, literature states that the US uses 450 Hz.
- a blank field (for later use);
- a label displaying the *state* of the decoder.

– *APTSTART* is - as the name suggests - the start state. The software will read incoming signals until a few seconds a signal of 300 Hz is received;

- *PHASING* is the state where the software is trying to synchronize. If - during a longer time - no reliable synchronization can be realized, the assumption is that the detection of the 300 Hz was erroneous, and the APTSTART state is entered;
- *ON SYNC* is the state when a reliable synchronization is detected, and in this state the data lines are processed. The lines in the picture will be displayed on the widget.
- *FAX_DONE* is - as the name suggests - the state entered when processing the picture finishes. If *saving* was set, the picture will be stored in a file and the APTSTART state will be entered again. If *saving* was not set, the software will wait in the final state until a *reset* is given.

The second line contains 8 elements

- the *save Cont* button. When set, the software will continuously run the sequence to decode a picture and store each picture in a file. The files are stored in the home folder of the user, and the name will contain the time. If set the text on the button shows as text *saving*, otherwise *Save*.
- the *reset* button, which does what can be expected from a reset button;
- a label on which - while in the *APTSTART* state - the frequency of the incoming decoded signal is displayed.
- a label on which - when in sync - the line number of the line currently being decoded is displayed;
- a *cheat* button. As said, processing a whole chart in mode Wefax576 takes well over 10 minutes. The cheat button cheats the system by forcing it into state *ON SYNC*. I.e. one can see some of the weatherchart, even in the middle of the transmission.
- a *correct* button. Due to clock offsets, the picture might be slightly leaning (as does the weatherchart on the pictures). When the picture is complete, it can be changed by touching this button. The correction factor is then given in the spinbox to the right.
- a *spinbox* telling the correction factor. Since the error is usually small, the correction factor indicates the *amount of samples added or subtracted from a group of 10 lines*. Experience shows that a correction of "2" (i.e. 2 samples per 10 lines), will correct the picture.
- a *save Single* button. After completing the decoding of a single picture, the picture can be stored in a file. When selected, the user will be requested to specify a filename.