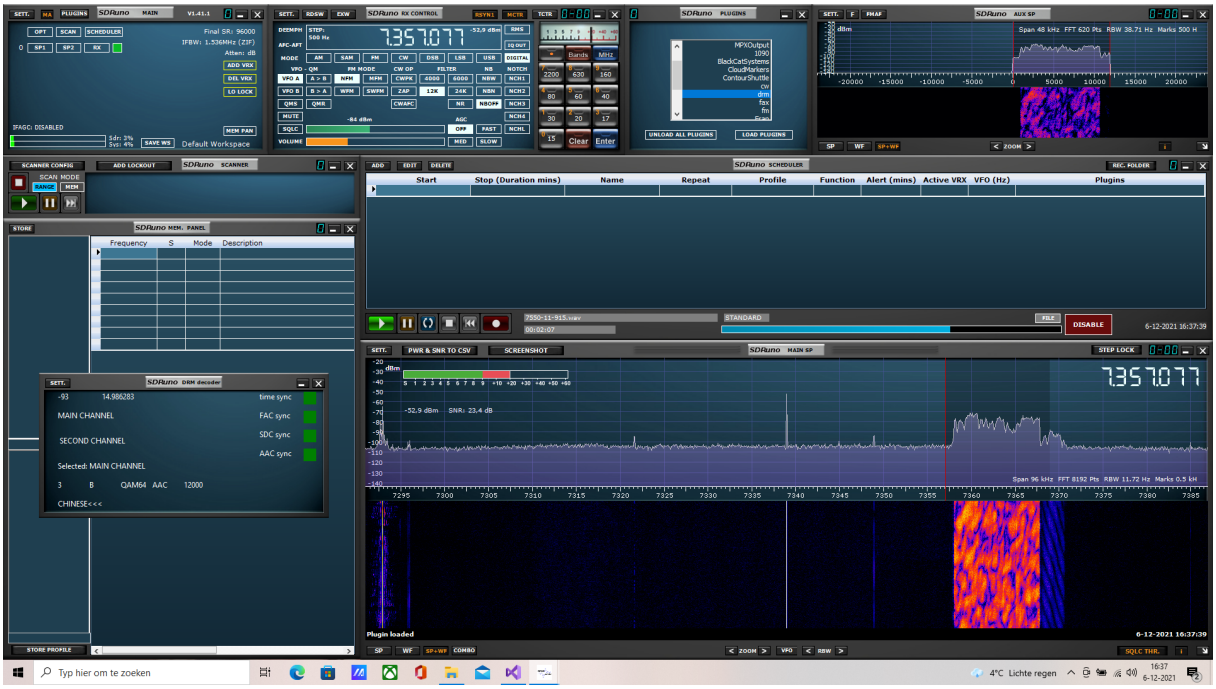
A DRM plugin for SDRuno

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

December 7, 2021



1 Introduction

DRM, Digital Radio Mondiale, is a system for digital radio on medium and short waves. DRM is not very popular in western Europe, I receive during daytimes the transmission from Radio Kuwait, and in the evening transmissions from Radio Romenia International. In the past there were transmissions from e.g. Nigeria and India, but I do not hear them anymore.

This document described a plugin for decoding DRM, developed for the SDRuno environment.

2 Installing

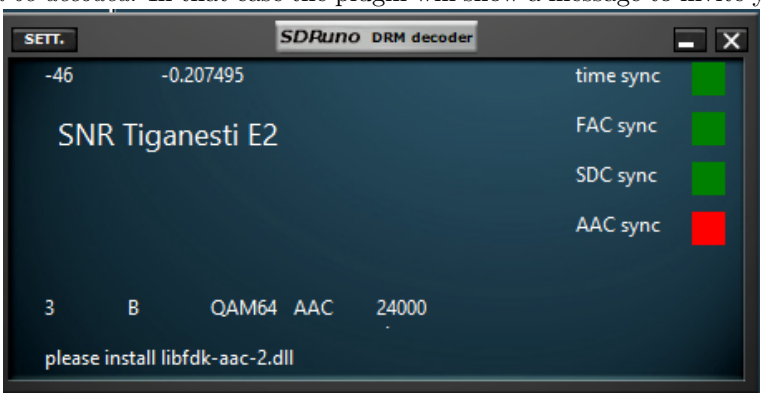
As other plugins, the plugin is implemented as a dll, i.e. a *dynamic load library*, and should be installed in the folder for community plugins.

The recent standard for DRM supports, next to AAC encoded audio, xHE-AAC encoded audio. The libfdk-aac-2 decoder is capable of handling both. The plugin uses this one for AAC and xHE-AAC decoding.

- *libfdk-aac-2.dll*, for the decoding of the AAC and xHE-AAC data is the main one. However, running this dll requires two additional dll's to be installed
- *libgcc_s_dw-1.dll*, and
- *libwinpthread-1.dll*

As said, these libraries are needed for convertings the AAC (or xHE-AAC) blocks to PCM samples, the last step in the plugin. The PCM samples are sent to the SDRuno system for the actual sound output.

If (one of) these libraries is NOT installed, the plugin will work with limited functionality: no sound will be decoded. In that case the plugin will show a message to invite you to install the library (libraries)



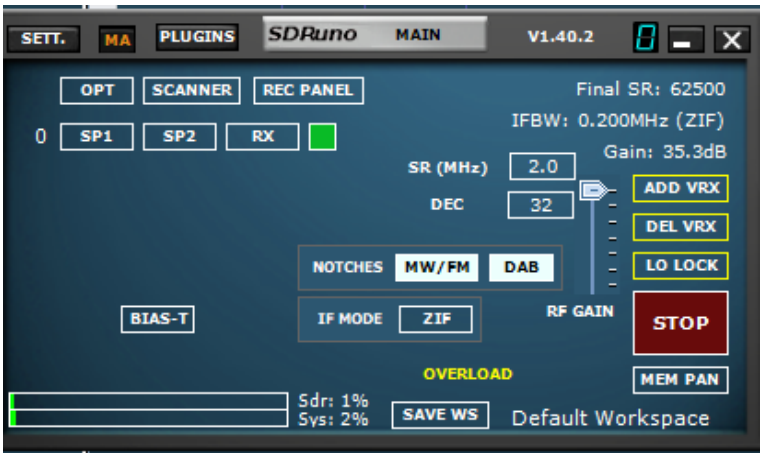
Copies of the libraries can be found in the "the-dll" folder in the repository. Being a Linux developer, I do not understand much of Windows, but I installed these libraries in the folder where the Uno stuff is stored:

```
C:\\Program Files(x86)\\SDRuno
```

3 Running the plugin

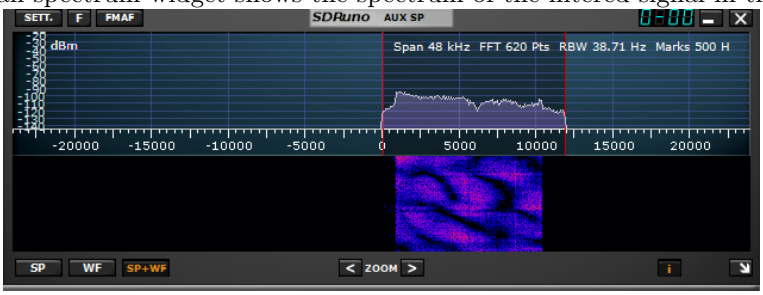
The DRM signal (in the common modes) has a spectrum with a width of 10 KHz. SDRuno might show a much wider spectrum. On a 2 MHz wide spectrum 10 KHz takes 1/200-th part and is therefore not (hardly) visible.

It seems therefore wise to select a smaller samplerate. *On the main widget select samplerate 2000000, and a suitable decimation factor* (I use a decimation factor 32, leading to a spectrum with a width of 63 KHz). The picture at the top of this paper shows a spectrum width of 192 KHz, it is made from a recording made by my SW software.



To use the plugin, select on the main receiver widget a bandwidth of 12, after all, the DRM signal has a spectrum with a width of 10 KHz. The software operates on a rate of 48K for both input and output, and the software will choose the mode *digital* for that.

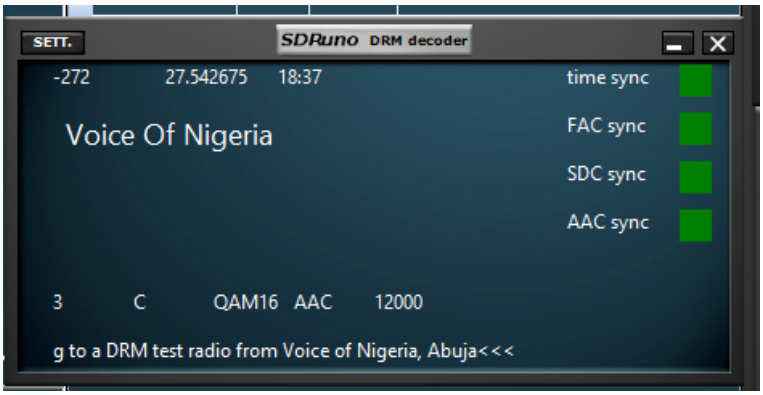
Note that - as can be seen on the picture on top - the filtered 12 k band should cover the drm signal, *the marker is on the left side of the selected spectrum part, NOT in the middle as for e.g. AM.* The small spectrum widget shows the spectrum of the filtered signal in the range 0 .. 12 KHz.



The widget shows that a 12k filter is selected and it shows that the software has selected the *digital* output.

4 The widget

On selecting the plugin, the software starts reading in samples, trying to reach synchronization, etc. The detected *mode* (one of Modes *A, B, C*) and the *spectrum* (one of 1 .. 3) are displayed, as is the way the bits of the audio content are encoded (usually QAM16 or QAM64), followed by the decoding of the audio (either AAC or xHE-AAC) and the output sample rate.



At the top left, one sees two numbers, telling the offset of the selected frequency. The first number indicates the so-called *coarse* offset, which is (should be) reasonably stable, the second one tells the so-called *fine* offset, which ranges between -30 .. 30 and is - in most cases - continuously changing. A number to the right might be displayed, which tells the time of the transmission (in UTC).

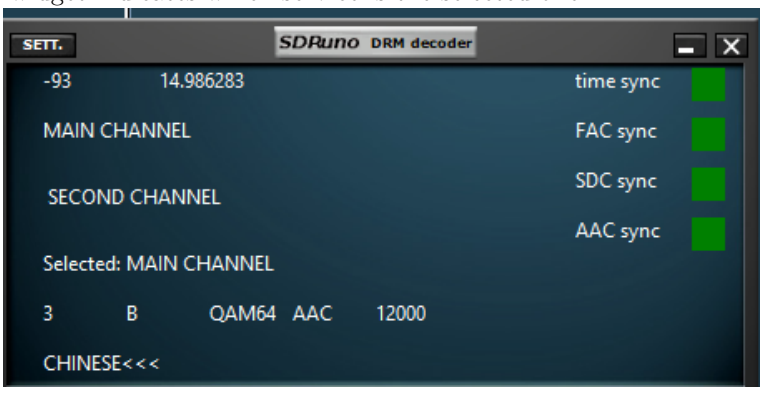
Processing the input takes 4 steps

- trying to reach time synchronization;
- then, trying to decode the FAC (Fast Access Channel) that contains some general information on the transmission;
- trying to decode the SDC (Service Description Channel), that contains detailed information on the service(s) and their decoding;
- trying to decode the AAC frames of the selected service.

The state of each of these "processes" is shown at the right side, from top to bottom an indicator for *time synchronization*, success in *FAC* decoding, success in *SDC* decoding and, finally, success with *AAC* decoding. If all these indicators are *green*, there will be sound, if one or mode of the indicators are red, some process parts fail with the given input and no sound can be emitted.

Services Most DRM transmissions carry a single service (as shown by the picture in the paragraph). The DRM technique, however, allows carrying more than one service (actually, up to 4). If a single service is carried, that service is - by default - the selected one and its content is decoded. If the transmission carries two services, both service names are shown and - by default - the first one is selected.

Clicking with the mouse on any of the two services will select that service, and to avoid ambiguity, the widget indicates which service is the selected one.



In some cases the transmission carries some additional text, this text is displayed at the bottom of the widget.

5 Limitations of the implementation

The DRM plugin has some limitations, when compared to the standard:

- the plugin does not provide support for data transmissions, other than texts appearing as program associated data;
- the plugin supports up to two audio services;
- the plugin - in its current form - does not support spectra larger than 10 KHz;