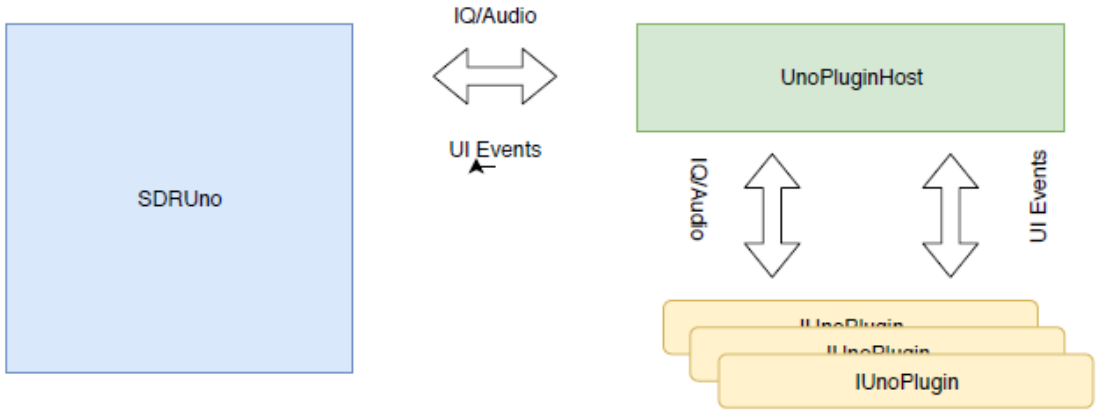
SDRuno Plugin System

Developer Pack

5th November 2020

SDRuno Plugin API Architecture

- SDRuno delivers and reacts to events to/from the plugin host via callbacks
- The main application has no inherent concept of plugins

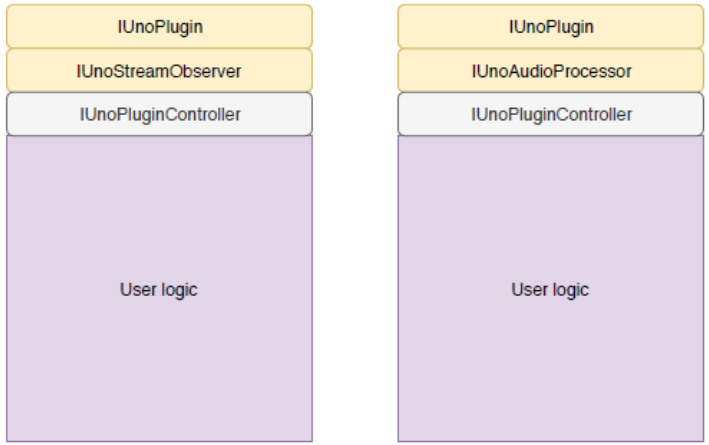


UnoPluginHost

The UnoPluginHost is the translation layer between the main application and plugins

It is loaded dynamically by SDRuno to remove dependency. SDRuno needs only to understand how to talk to the plugin host.

It is a bi-directional proxy. The decoupling is intended to provide scope to make interactions asynchronous for the avoidance of performance problems.



IUnoPlugin

Plugins implement well defined interfaces from the API. A plugin controller object is provided by the plugin host to each individual plugin.

Plugins can optionally subscribe to the output of various stages of the DSP processing pipeline.
A callback in the plugin defined as part of the overridden interface is the entry point.

Plugin Development Files

GitHub – https://github.com/SDRplay/plugins

/include – Plugin System Header Files
/nana – Nana UI Library* (see below)
/resources – SDRplay image files for plugin background
/SDRunoPlugin_Template – Visual Studio 2017 solution
/SDRunoPlugin_DXCluster – Working example plugin
/SDRunoPluginsDeveloperPack.pdf – This document

* Nana UI is a cross platform open source UI library and included for convenience. Due to github file size limit, after cloning the repository, go to /nana/build/bin and uncompress the nana_v141_Release_x86.zip to extract the nana_v141_Release_x86.lib file

Main Concepts

- Read SDRuno's IQ stream
- Read/Modify SDRuno's audio stream
- Set/Get SDRuno properties
  - Including using SDRuno ini file for settings (Set/GetConfigurationKey)
- Annotate on the SP1 Panel spectrum display
- Receive events from SDRuno

Register/Unregister Functions

- RegisterStreamObserver
- RegisterStreamProcessor
- RegisterAudioObserver
- RegisterAudioProcessor
- RegisterAnnotator

- UnregisterStreamObserver
- UnregisterStreamProcessor
- UnregisterAudioObserver
- UnregisterAudioProcessor
- UnregisterAnnotator

Get Functions

- GetDemodulatorType
- GetVfoFrequency
- GetCenterFrequency
- GetFilterBandwidth
- IsStreamingEnabled
- GetSampleRate
- GetAudioSampleRate
- GetSquelchLevel
- GetSquelchEnable
- GetAgcMode
- GetAgcThreshold
- GetNoiseReductionLevel
- GetCwPeakFilterThreshold
- GetFmNoiseReductionEnable
- GetFmNoiseReductionThreshold
- GetWfmDeemphasisMode
- GetAudioVolume
- GetAudioMute
- GetSNR
- GetPower
- GetConfigurationKey
- GetVRXCount
- GetVRXEnable
- GetStepSize
- GetVFOSelect
- GetSP1MinFrequency
- GetSP1MaxFrequency
- GetSP1MinPower
- GetSP1MaxPower

Set Functions

- SetDemodulatorType
- SetVfoFrequency
- SetCenterFrequency
- SetFilterBandwidth
- IsStreamingEnabled
- SetSquelchLevel
- SetSquelchEnable
- SetAgcMode
- SetAgcThreshold
- SetNoiseBlankerLevel
- SetNoiseReductionLevel
- SetCwPeakFilterThreshold
- SetFmNoiseReductionEnable
- SetFmNoiseReductionThreshold
- SetWfmDeemphasisMode
- SetAudioVolume
- SetAudioMute
- SetConfigurationKey
- SetVRXEnable
- SetVFOSelect

Additional Functions

- RequestUnload

Events

- DemodulatorChanged
- BandwidthChanged
- FrequencyChanged
- CenterFrequencyChanged
- SampleRateChanged
- StreamingStarted
- StreamingStopped
- SquelchEnableChanged
- SquelchThresholdChanged
- AgcThresholdChanged
- AgcModeChanged
- NoiseBlankerLevelChanged
- NoiseReductionLevelChanged
- CwPeakFilterThresholdChanged
- FmNoiseReductionEnabledChanged
- FmNoiseReductionThresholdChanged
- WfmDeemphasisModeChanged
- AudioVolumeChanged
- AudioMuteChanged
- SavingWorkspace
- VRXCountChanged
- VRXStateChanged
- StepSizeChanged
- VFOChanged
- ClosingDown
- SP1MinFreqChanged
- SP1MaxFreqChanged
- SP1MinPowerChanged
- SP1MaxPowerChanged

Annotator Structure

- Array of structs
- Struct contents
  - frequency (x location)
  - power (y location)
  - text (to be displayed)
  - style (see below)
  - rgb (RGB colour)
  - lineToFreq (x location for the end of the line) – if empty, no line will be drawn
  - lineToPower (y location for the end of the line) – if empty, no line will be drawn
- Style
  - AnnotatorStyleFlag
  - AnnotatorStyleBox
  - AnnotatorStyleMarker
  - AnnotatorStyleMarkerAndLine
- If the annotator has been registered and the array of structs is not null then periodically the array will be cycled through. If frequency[i] and power[i] are visible in the SP1 panel spectrum display, then the specified text[i] will be displayed.
- For a line to be drawn, frequency[i], power[i], lineToFreq[i] and lineToPower[i] will all need to be within the visible spectrum.

Our Plugin Roadmap

- Available Now
  - DAB/DAB+
  - Audio Recorder – PCM WAV and MP3 output
  - DX Cluster
  - MPX Output – RDS output for applications such as RDS Spy

- Planned for future releases
  - ADS-B (dump1090)
  - Direction Finding – to work with the RSPduo