

Sistemas de Comunicaciones basados en Radio Definida por Software (SDR)

Dr. Ing. Alejandro José Uriz

Conceptos Introductorios

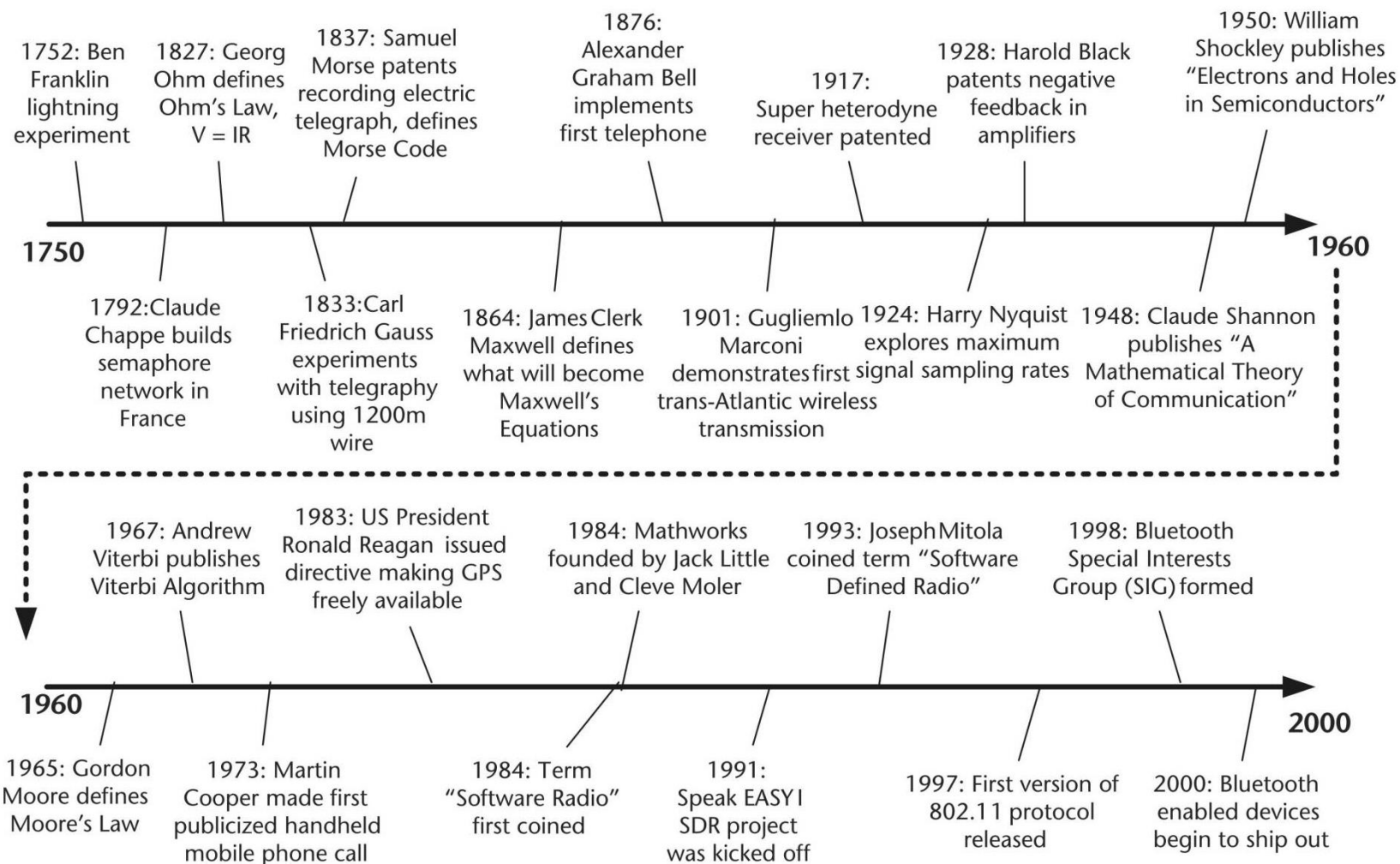
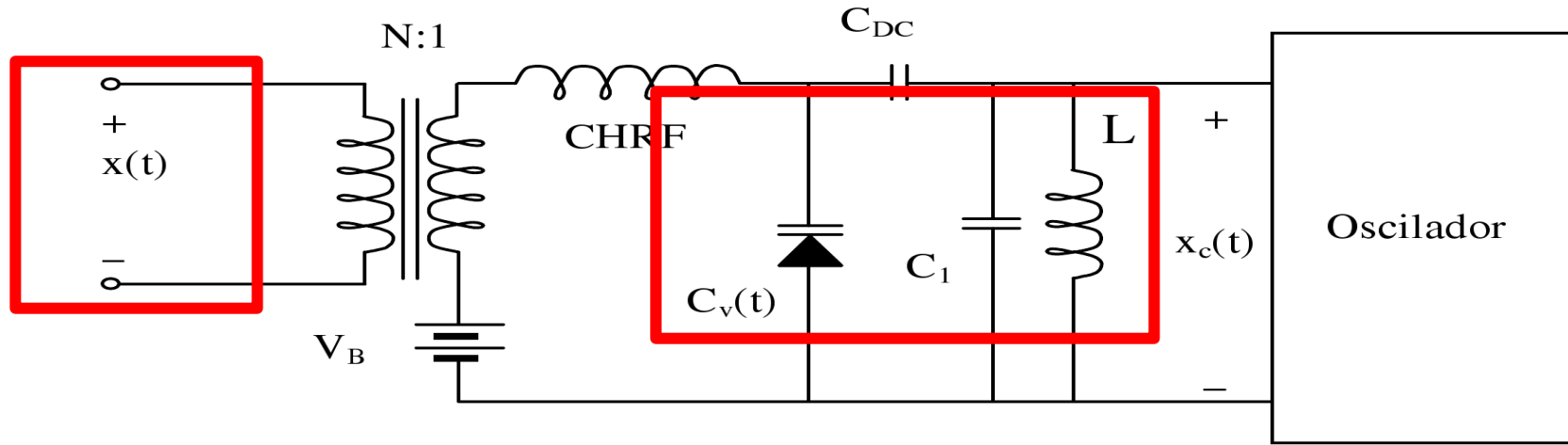


Figure 1.1 Timeline of several key milestones in communications.

Método directo de generación de FM



Este circuito consiste en un VCO. Su frecuencia de resonancia depende de L y C . La capacidad total del sistema es controlada mediante $x(t)$.

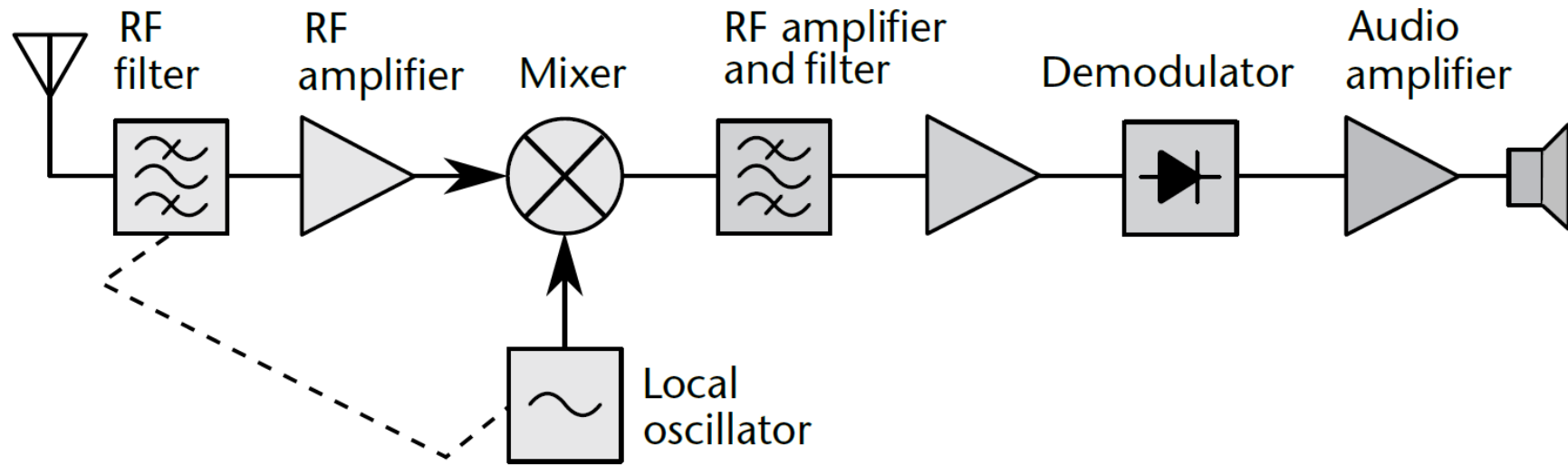
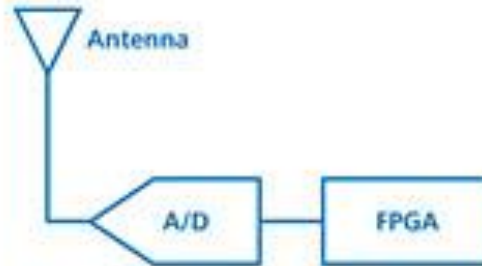
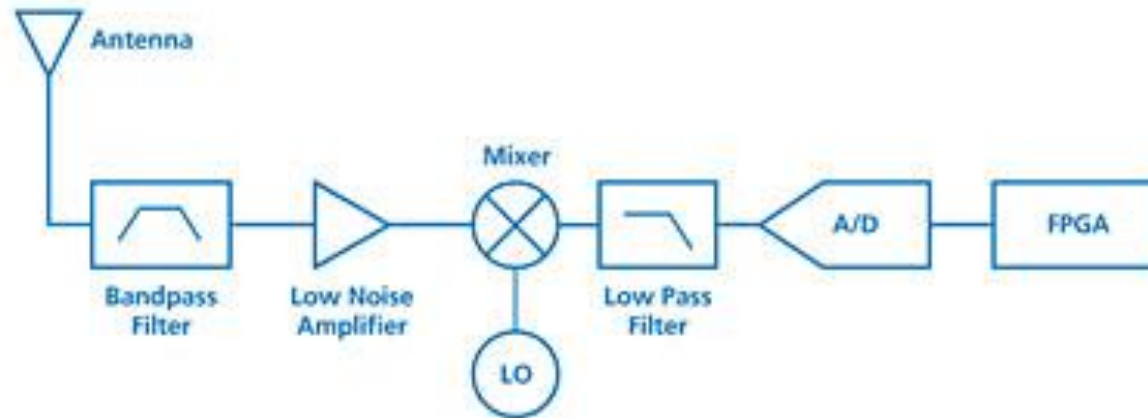
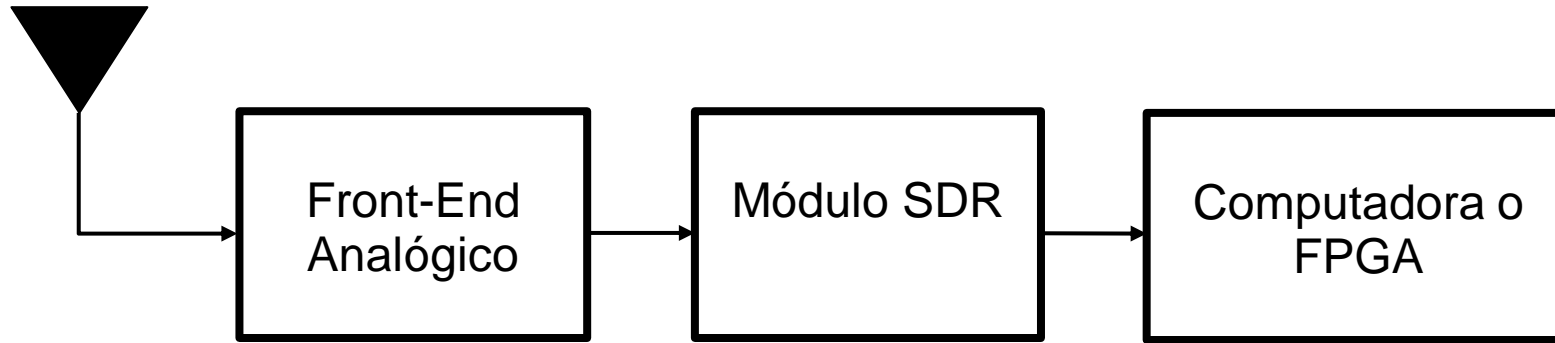
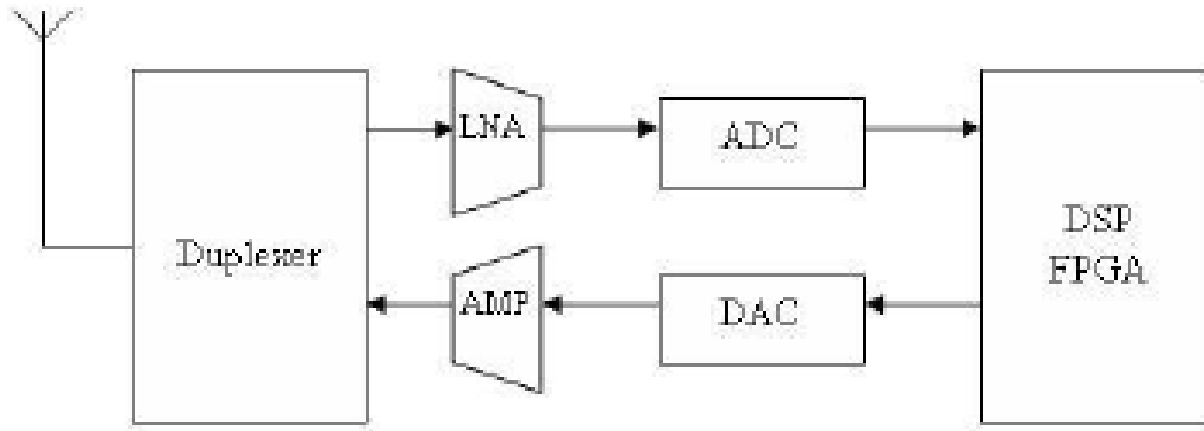
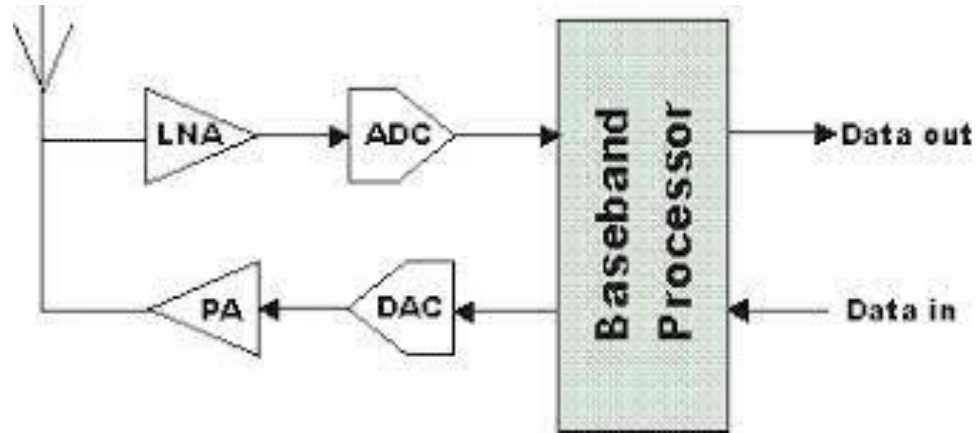


Figure A.3 Single-conversion superheterodyne radio receiver. The incoming radio signal from the antenna (left) is passed through an RF filter to attenuate some undesired signals, amplified in a radio frequency (RF) amplifier, and mixed with an unmodulated sine wave from a local oscillator. The result is a beat frequency or heterodyne at the difference between the input signal and local oscillator frequencies, a lower frequency called the IF. The IF signal is selected and strengthened by several IF stages that bandpass filter and amplify the signal. The IF signal is then applied to a demodulator that extracts the modulated audio signal. An audio amplifier further amplifies the signal, and the speaker makes it audible.

FIGURE 1**THE SOFTWARE-DEFINED RADIO RECEIVER
IN ITS IDEAL AND PRACTICAL FORMS****(a) Ideal SDR****(b) Practical SDR**





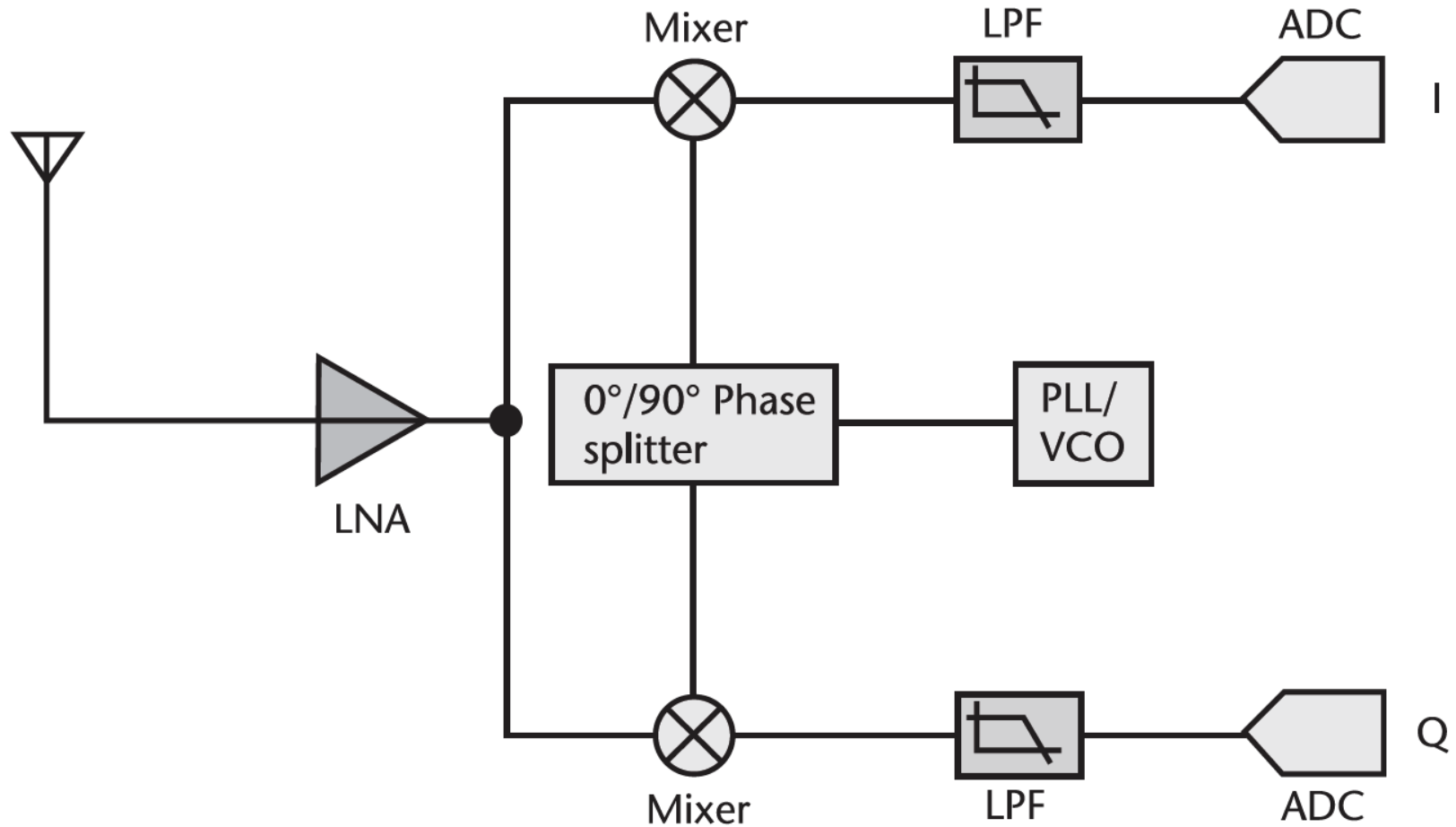
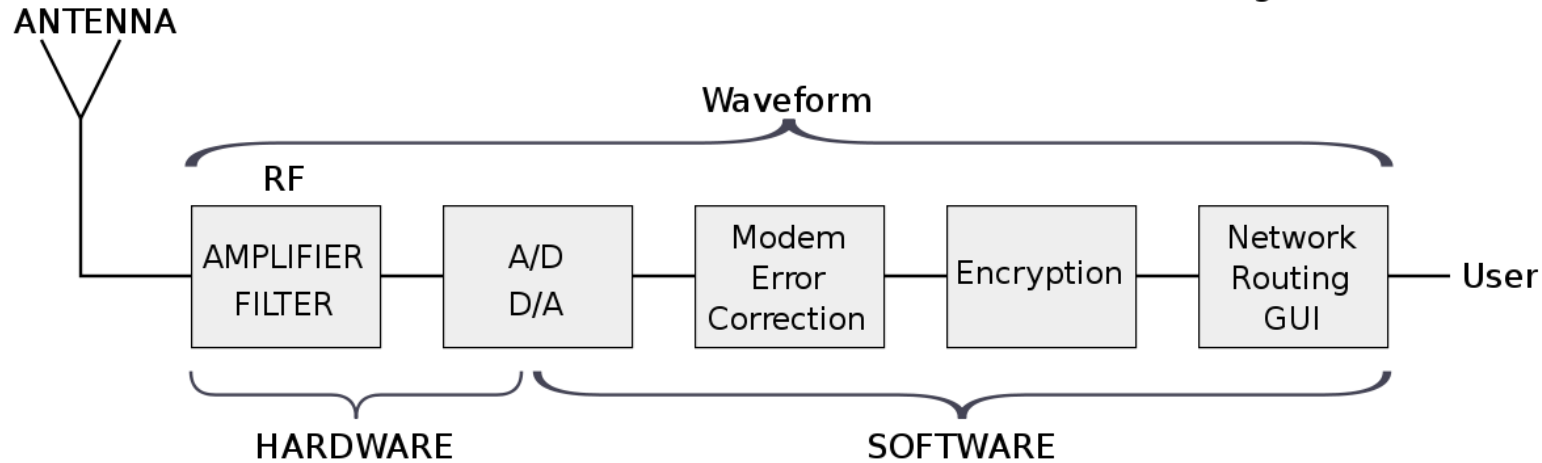
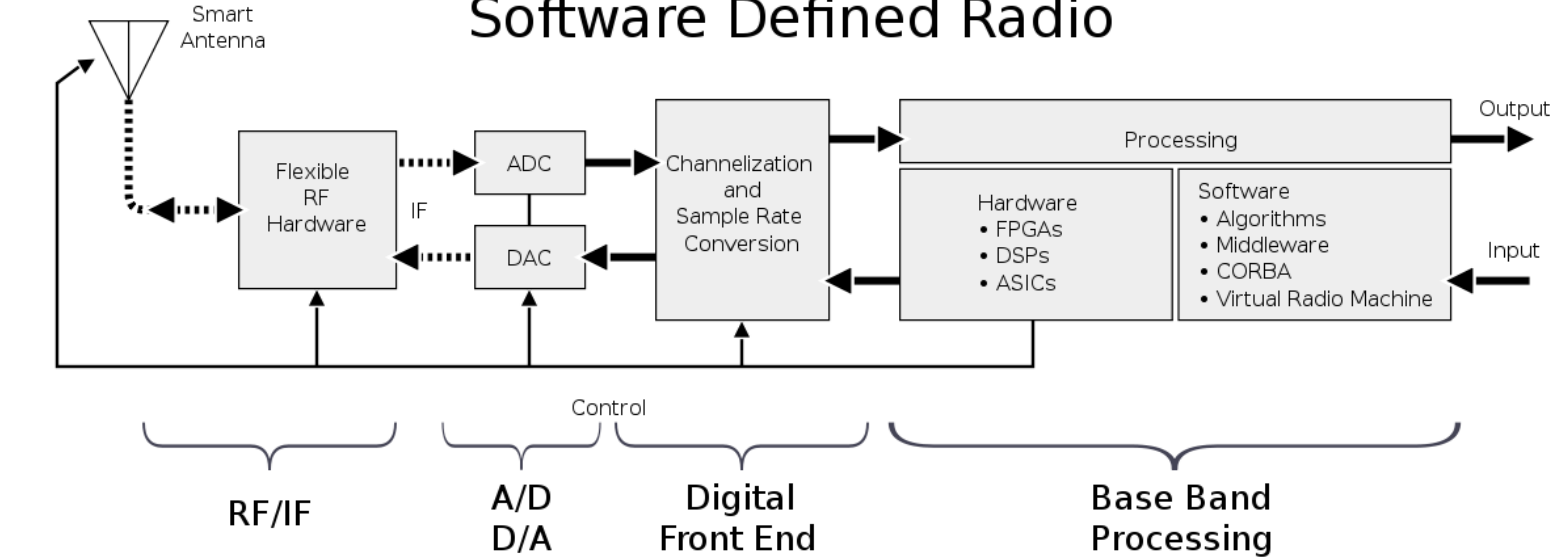


Figure 1.7 Zero IF architecture [4].

Software Defined Radio



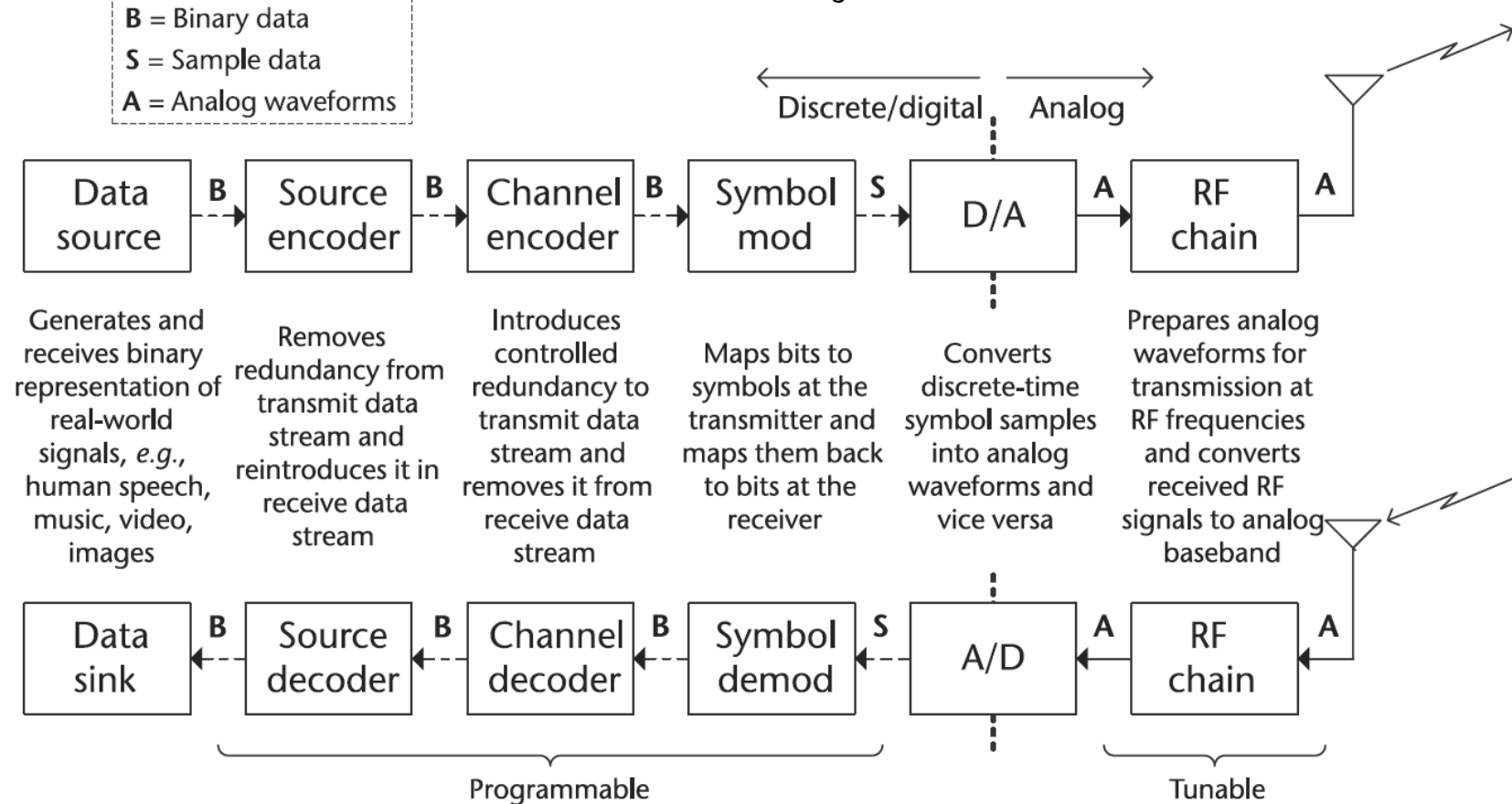
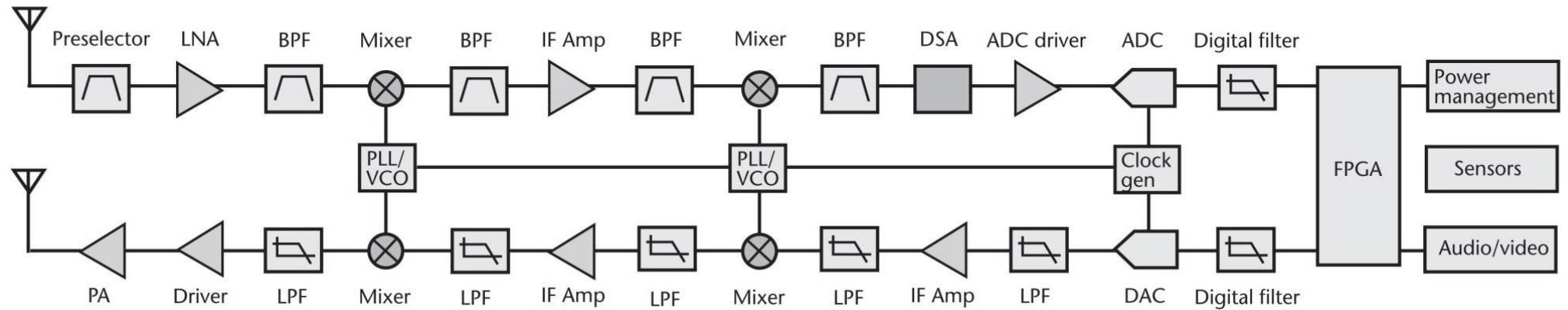
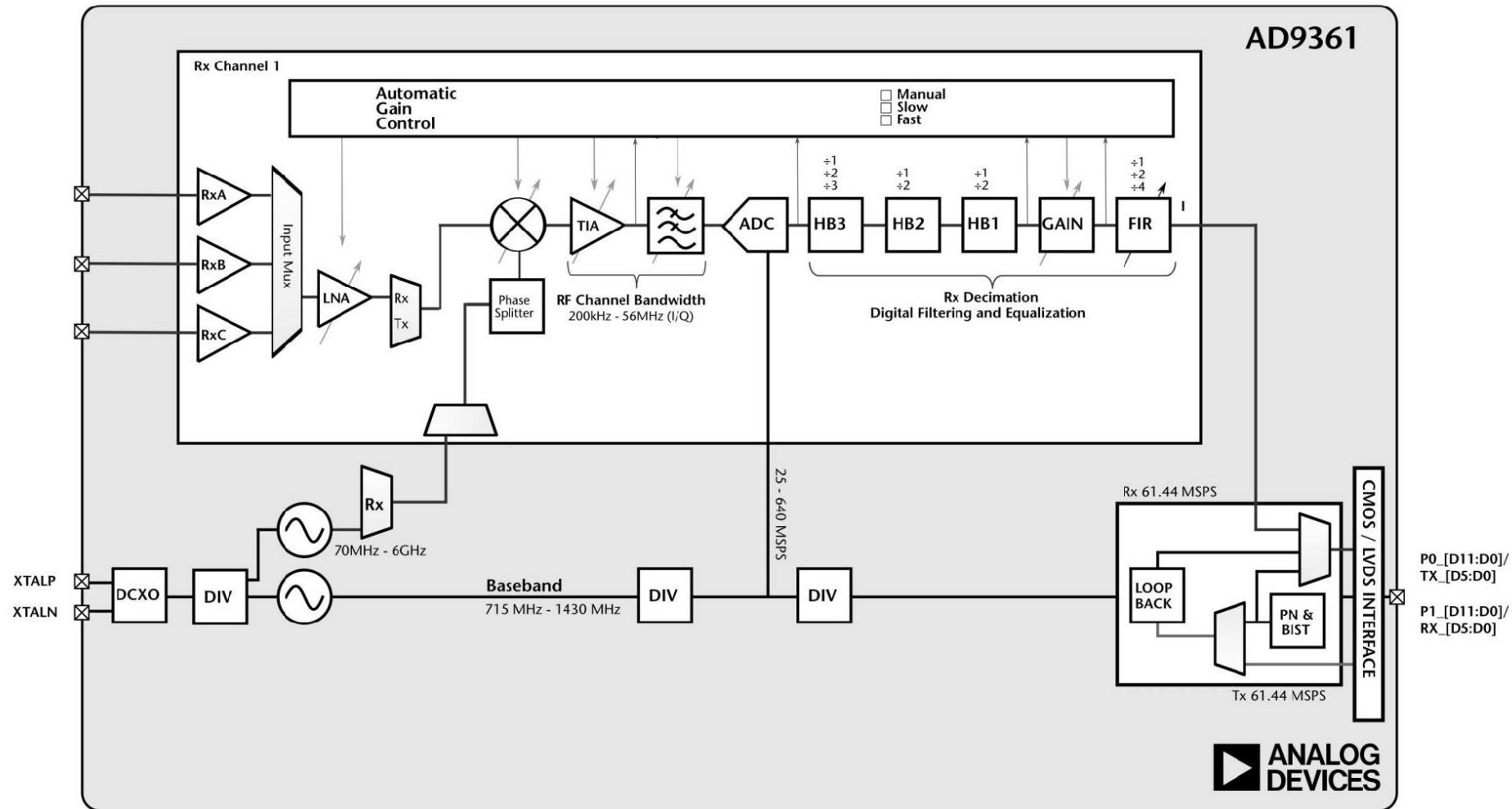
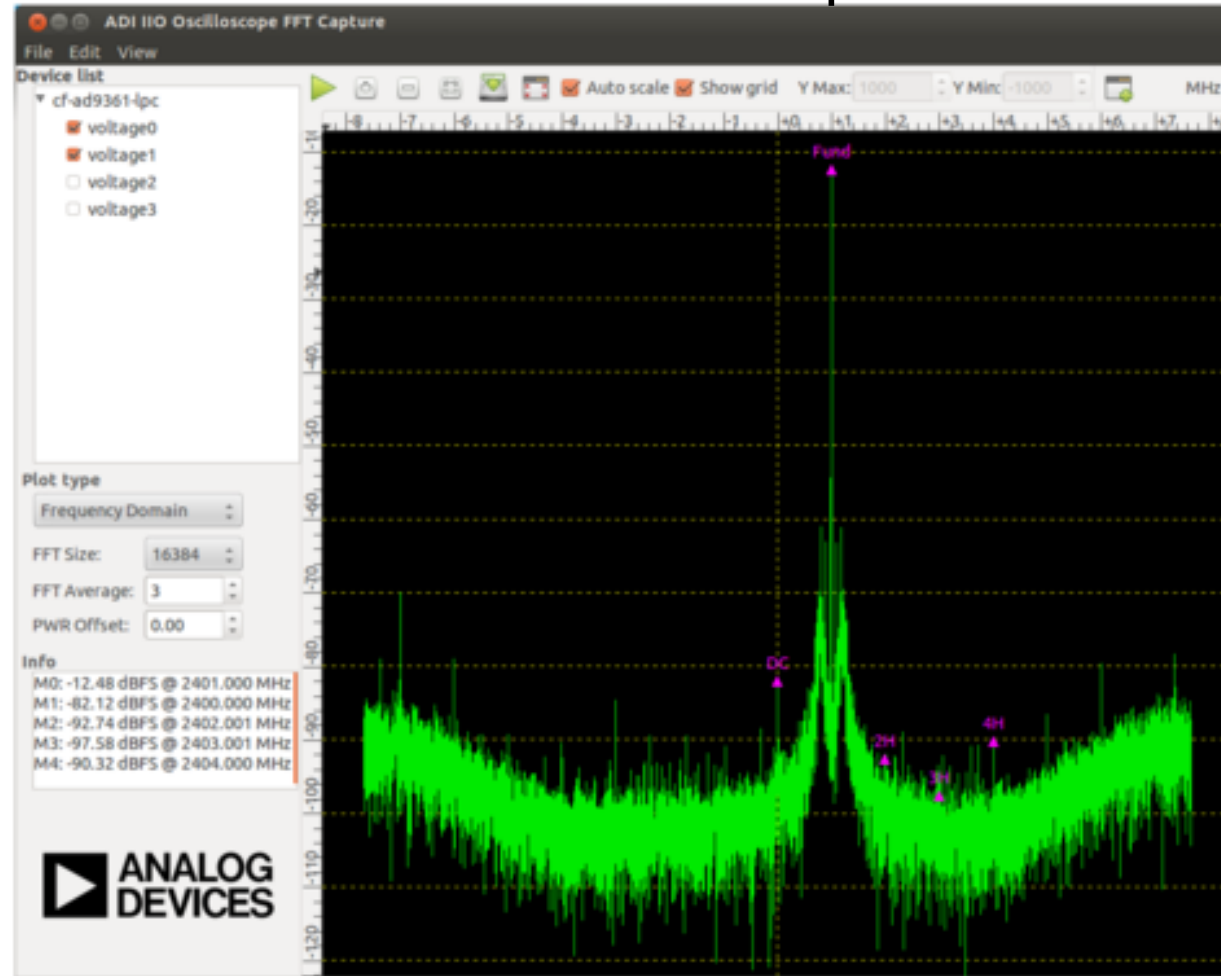


Figure 1.3 An illustration describing some of the important components that constitute a modern digital communications system. Note that for a SDR-based implementation, those components indicated as programmable can be realized in either programmable logic or software.



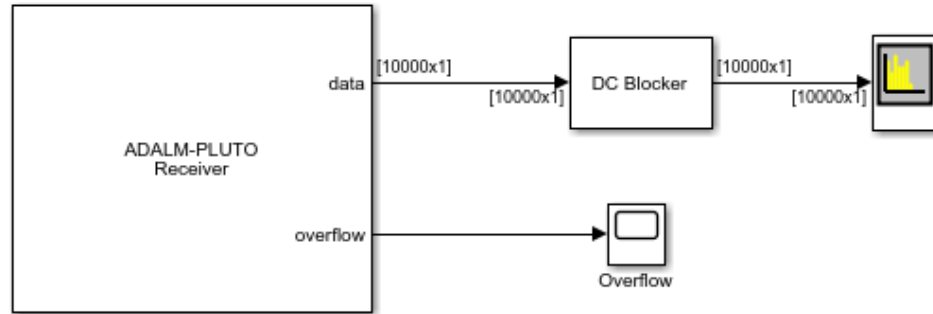


ADI IIO Oscilloscope



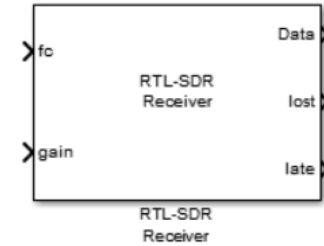
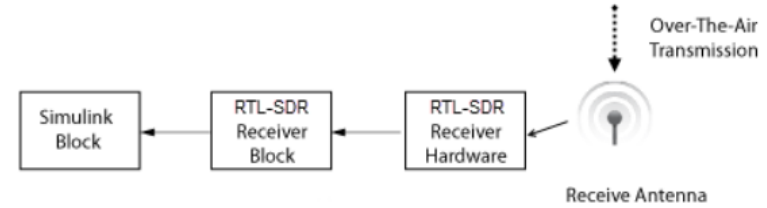
MatLab

Spectrum Analysis with ADALM-PLUTO Radio



Copyright 2016-2017 The MathWorks, Inc.

Info

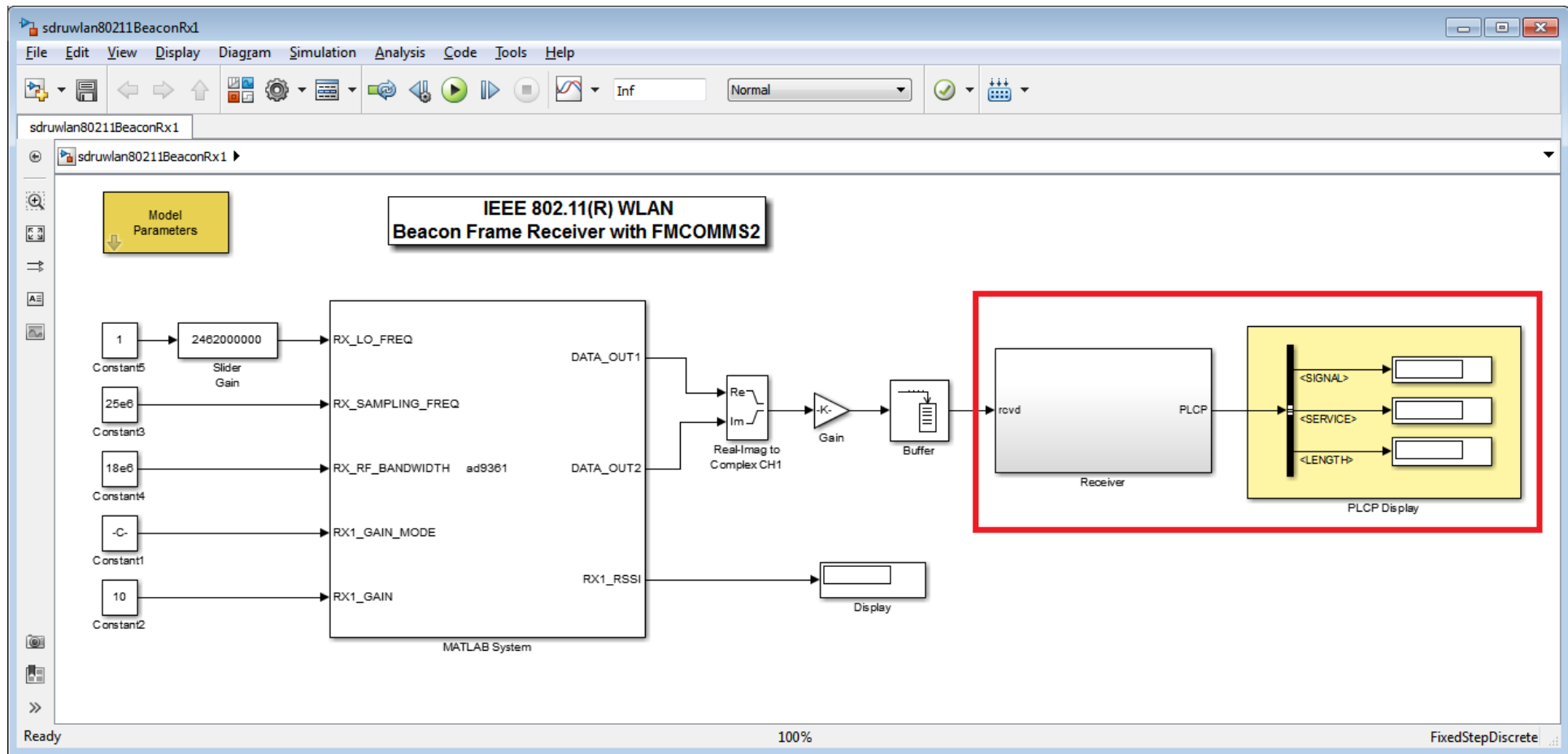


Ports

Input

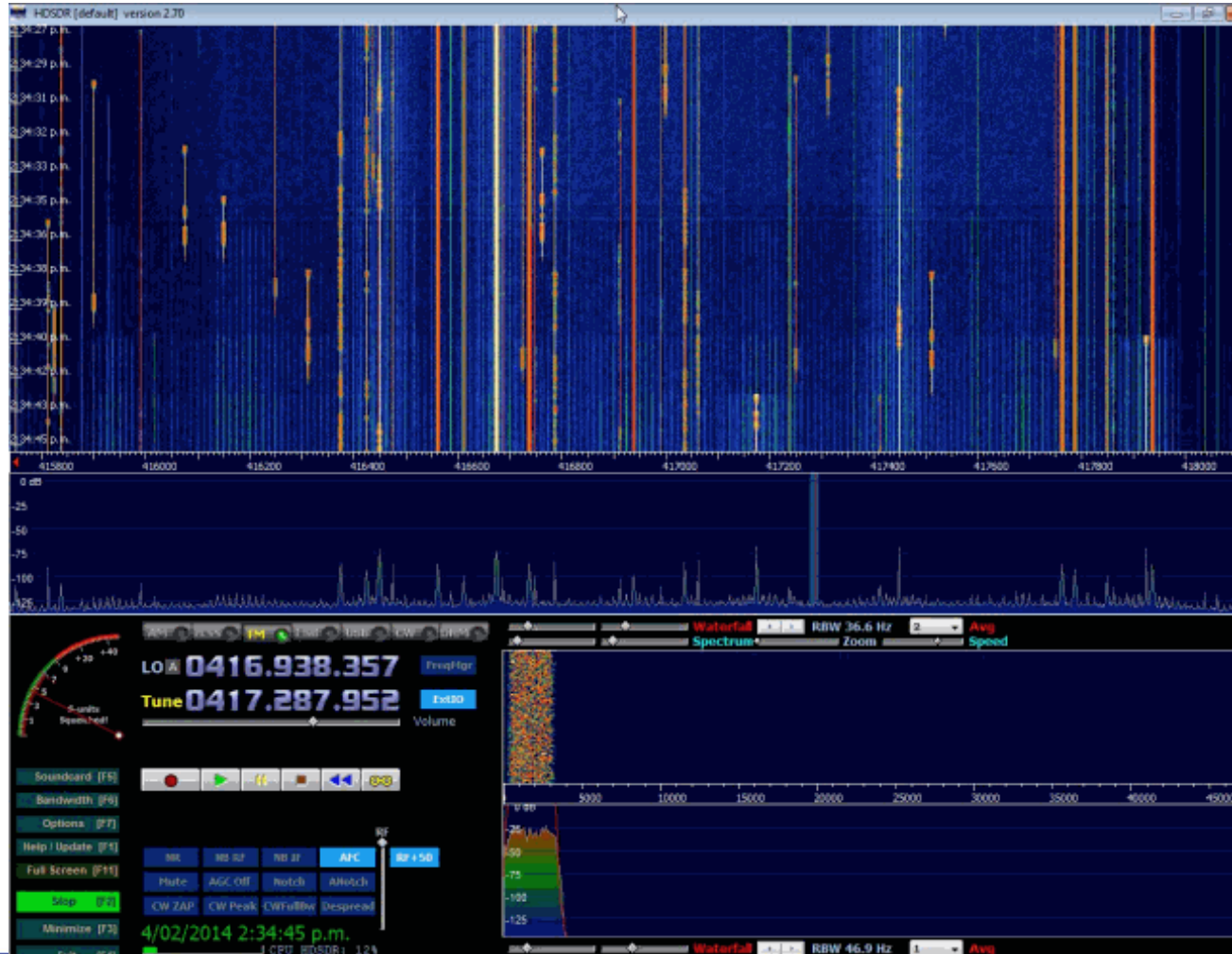
> **fc** – Center frequency
positive scalar

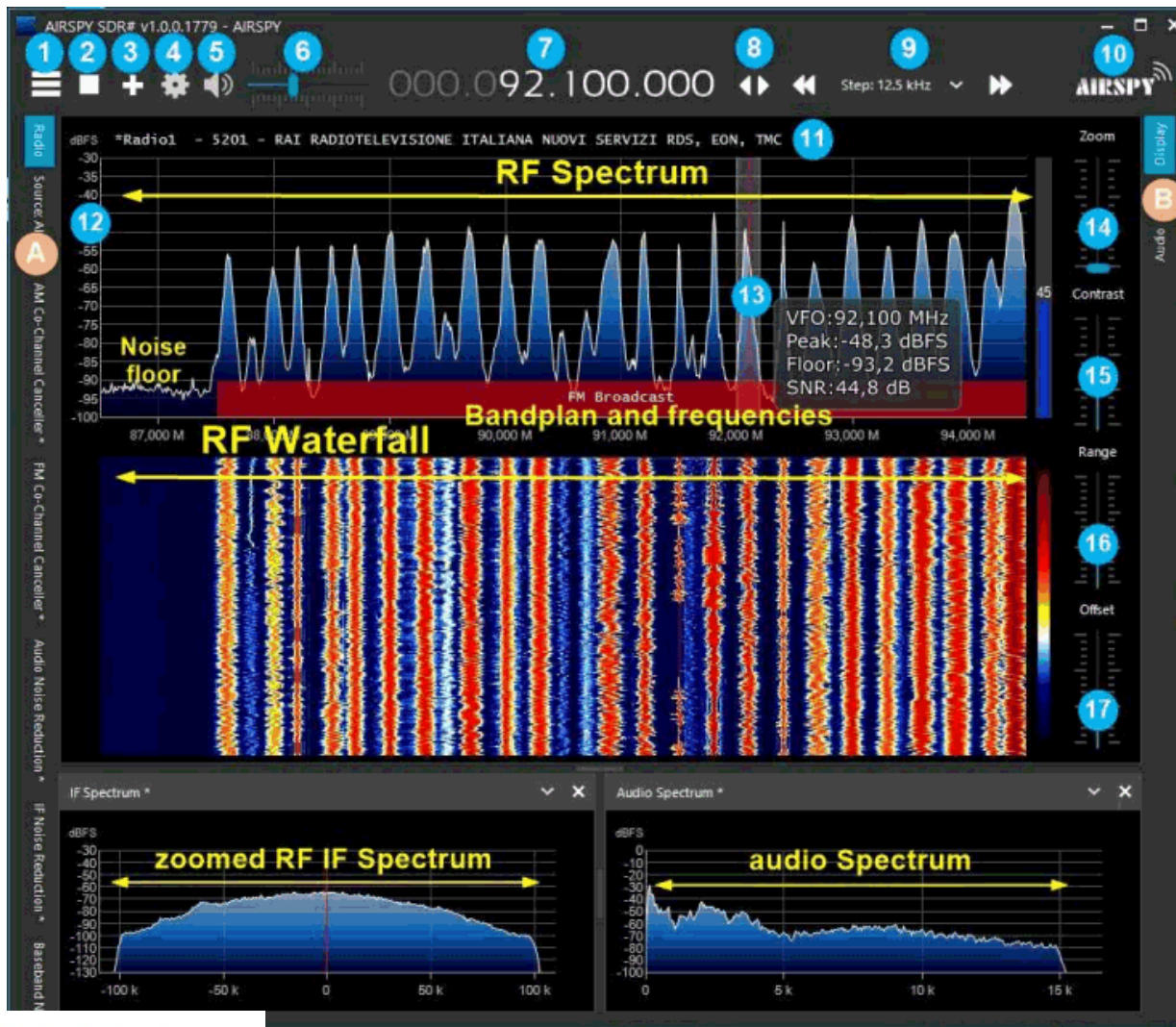
> **gain** – Receiver gain
scalar

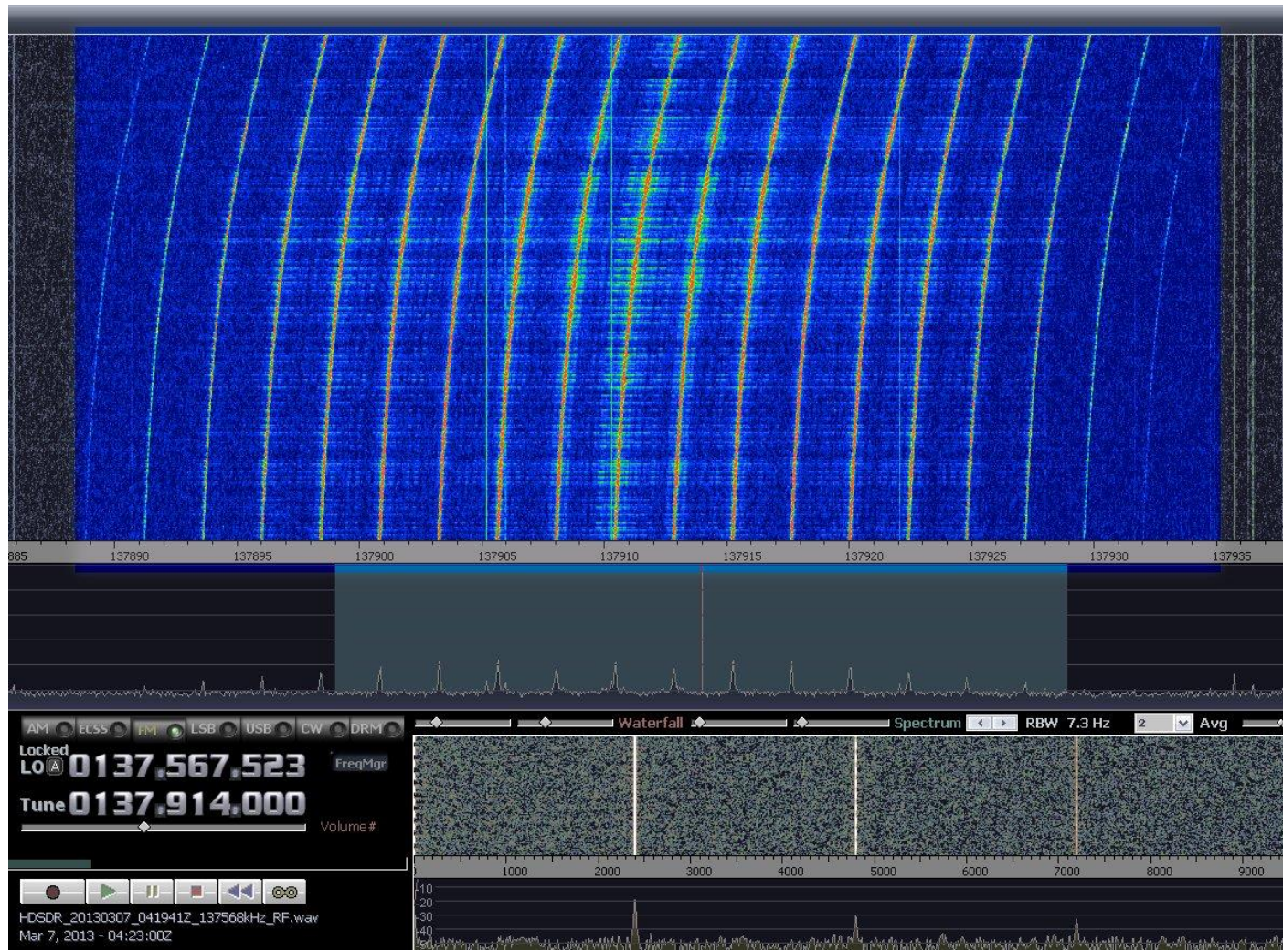


Diseño basado en modelos

HDSDR







Google Collab



Eye Diagram and ISI.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se editó por última vez: 25 de abril](#)

Comentar Compartir Configuración Perfil

+ Código + Texto

Conectar Editando

```
[ ] #----- Transmitter configuration parameters -----

num_bits = 2**13 # Number of transmitted bits
sps      = 8     # Samples per symbol
span     = 12    # The filter is truncated to span symbols
beta     = 1     # Excess-bandwidth parameter
sample_rate = 4e6 # Sample rate RX and TX paths[Samples/Sec]


[ ] #----- SDR Parameter Configuration -----

Uri          = "ip:10.0.0.71"
SamplingRate = sample_rate # Sample rate RX and TX paths[Samples/Sec]
Loopback     = 1           # 0=Disabled, 1=Digital, 2=RF

TxLOFreq     = 910e6       # Carrier frequency of TX path [Hz]
TxAtten      = -40         # Attenuation applied to TX path, valid range is -90 to 0 dB [dB]
TxRFBw       = 2e6         # Bandwidth of front-end analog filter of TX path [Hz]

RxLOFreq     = TxLOFreq    # Carrier frequency of RX path [Hz]
GainControlModes = "slow_attack" # Receive path AGC Options: slow_attack, fast_attack, manual
RxHardwareGain = 0         # Gain applied to RX path. Only applicable when gain_control_mode is set to 'manual'
RxRFBw       = TxRFBw      # Bandwidth of front-end analog filter of RX path [Hz]
RxBufferSize = 2**20-1

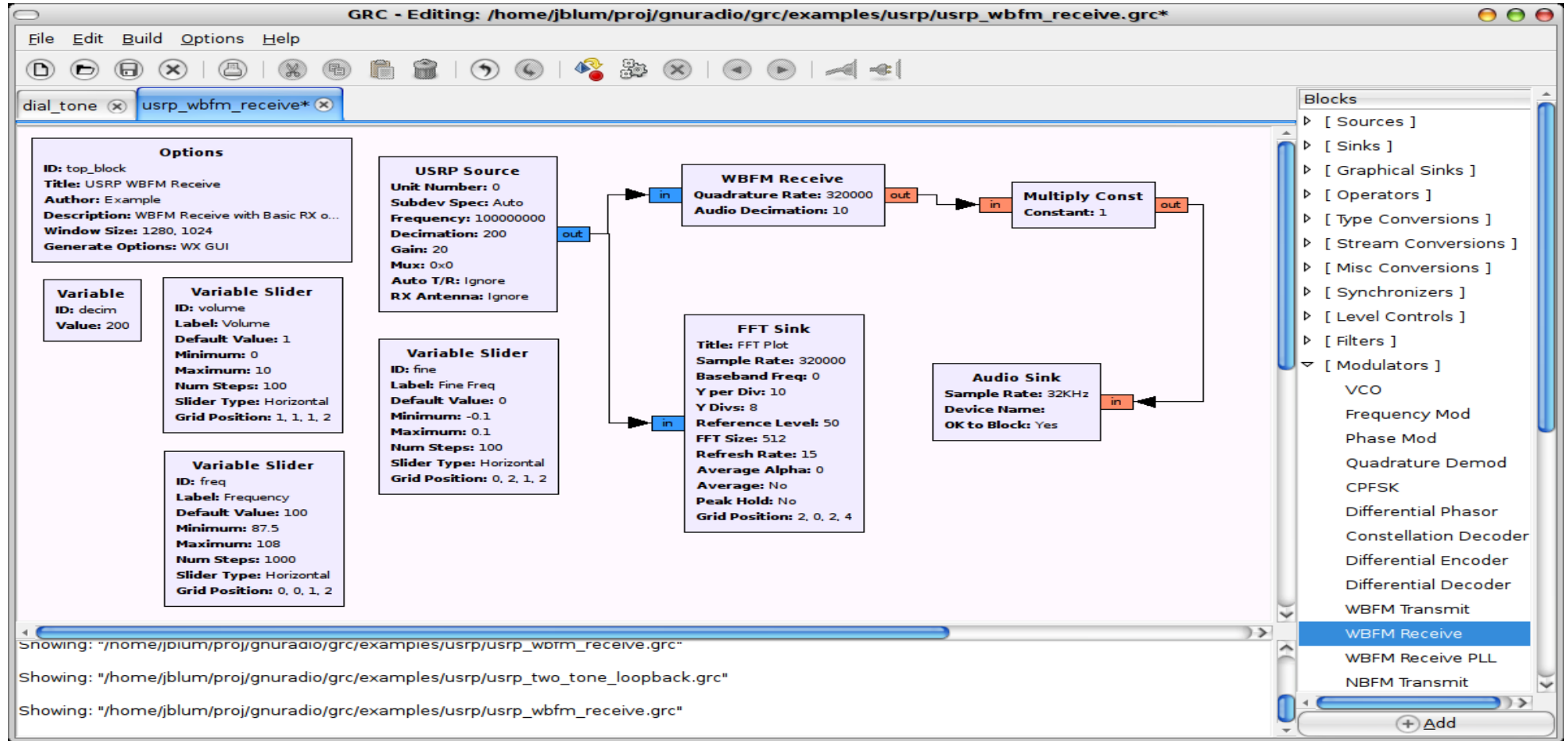

[ ] #----- Root Raised Cosine Filter Function -----

def rcosdesign(beta,span,sps):
    index = np.arange(-(span*sps)/2,(span*sps)/2,1)
    Ts    = sps
    rrcFilter = np.array([])

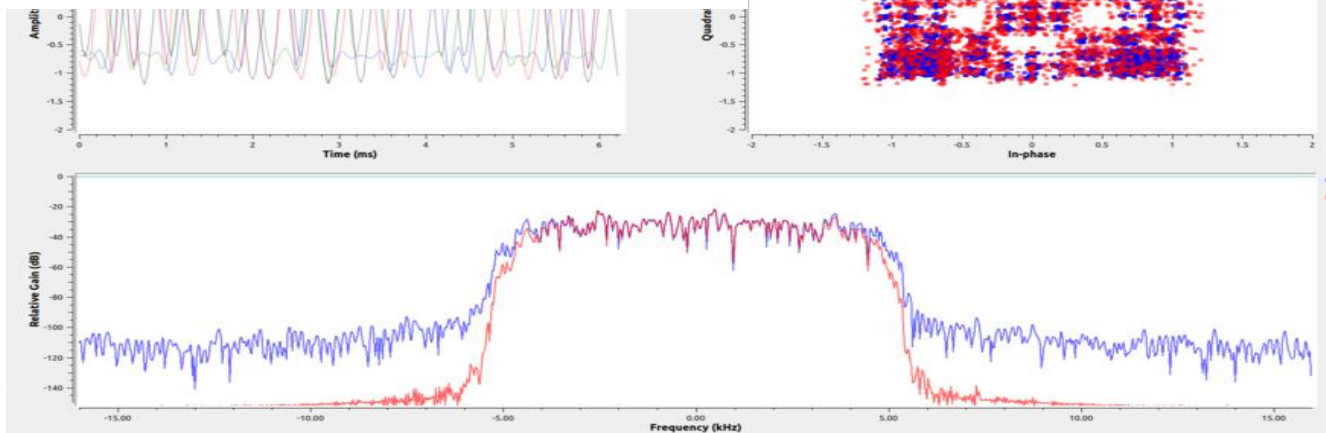
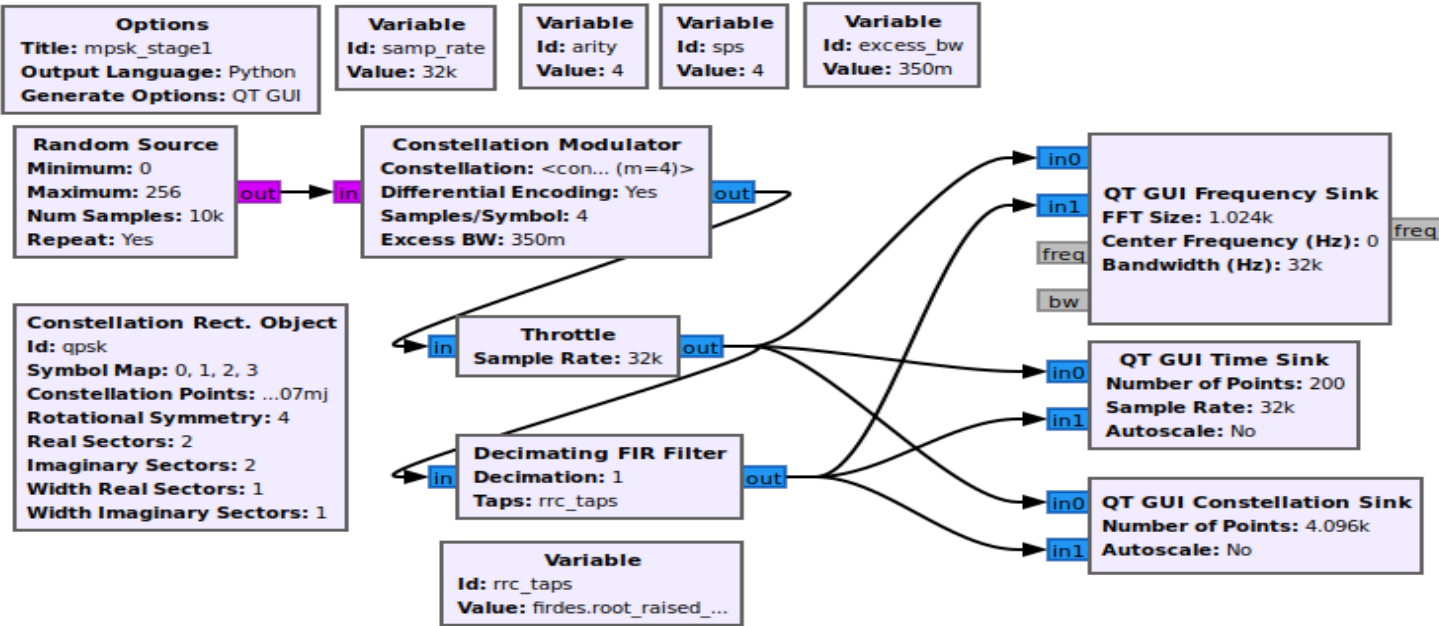
    for n in index:
        if n == Ts/(4*beta) or n == -Ts/(4*beta):
            aux = beta*((np.pi+2)*np.sin(np.pi/(4*beta))+(np.pi-2)*np.cos(np.pi/(4*beta)))/(np.pi*np.sqrt(2))
            rrcFilter = np.append(rrcFilter,aux)
```



GNU RADIO



Ejemplo QPSK



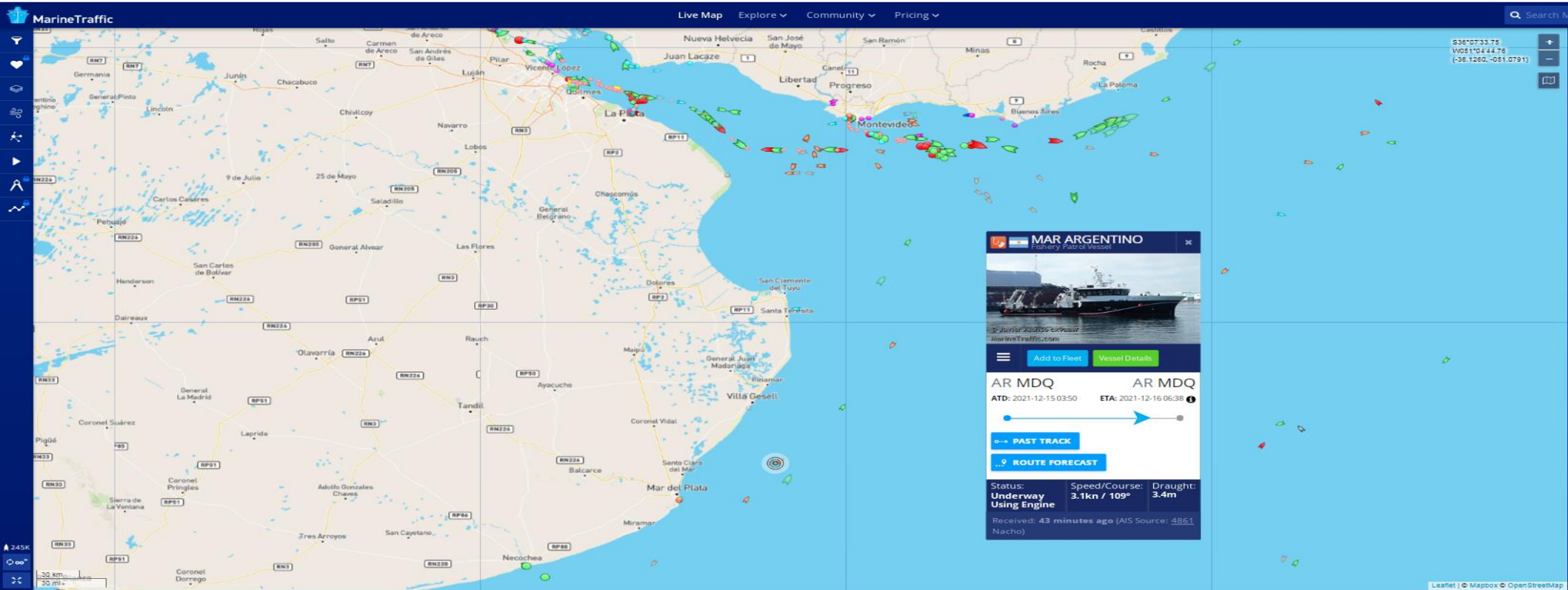
Algunos módulos de GNU Radio Companion

Algunos ejemplos de módulos para instalar en GNU Radio son los siguientes:

- gr-satellites
- gr-adsb
- gr-IEEE802-15-4
- gr-lora
- gr-gsm
- gr-isdtv
- gr-bluetooth
- gr-iridium
- gr-IEEE802-11

Se pueden encontrar más en: <https://www.cgran.org/>

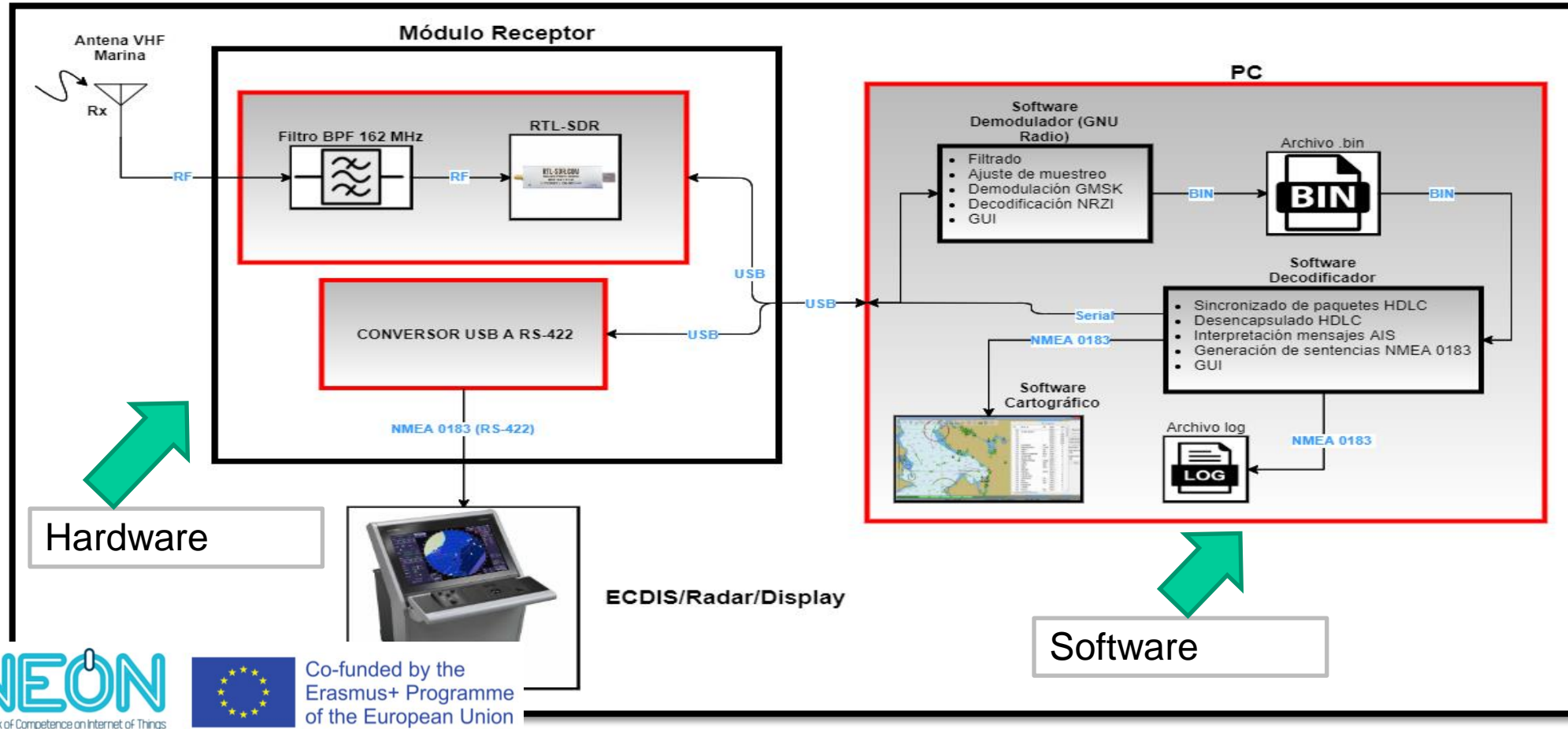
Ejemplo avanzado: Receptor AIS

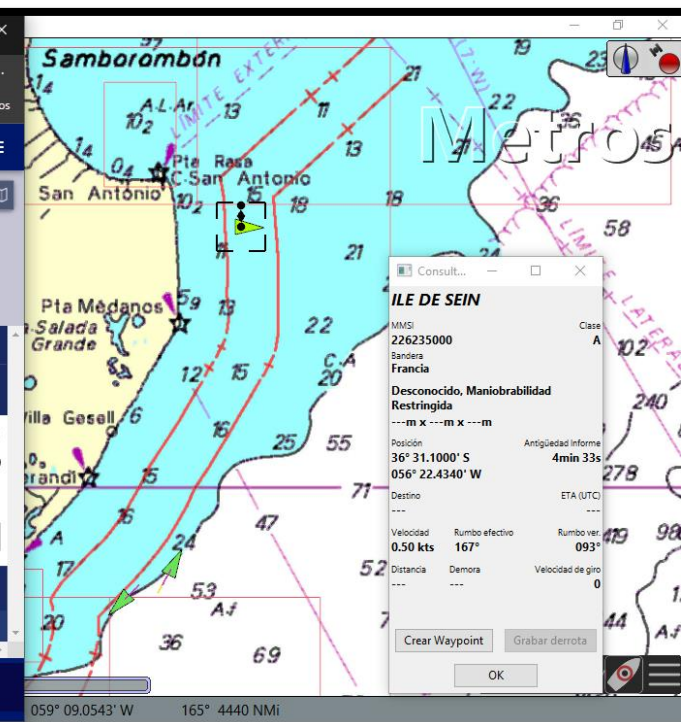
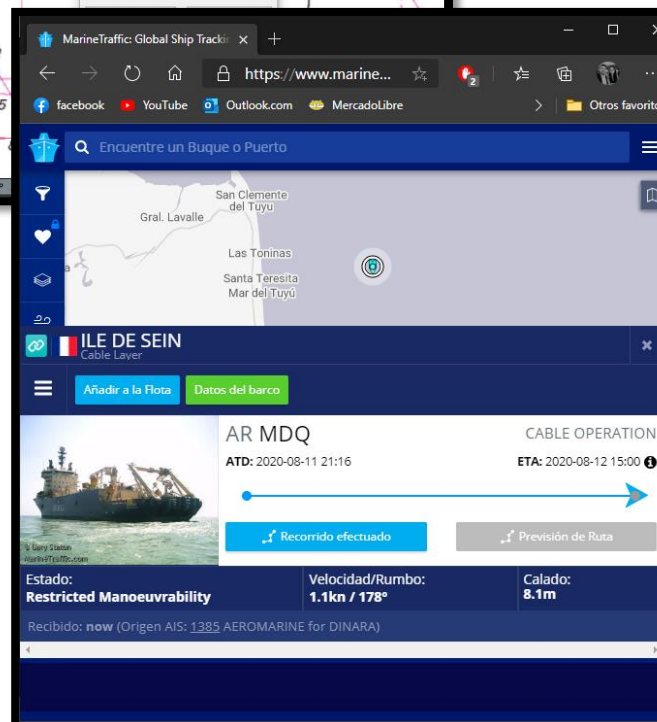
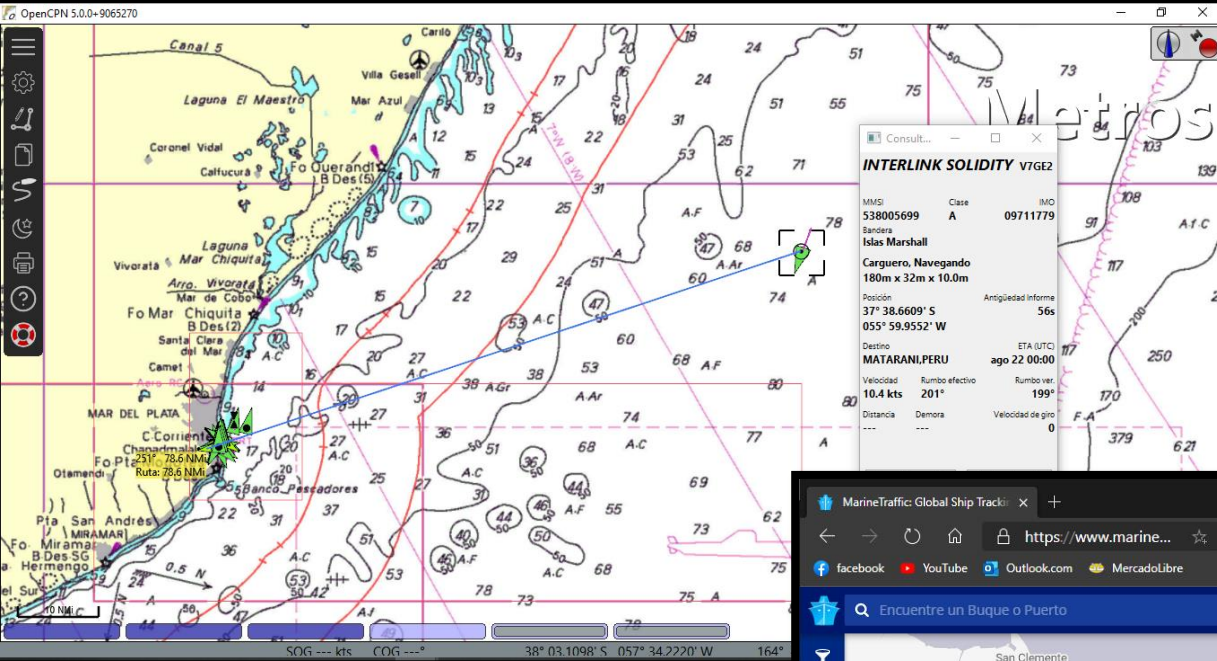


Captura del sitio <https://marinetraffic.com>

Ejemplo avanzado: Receptor AIS

Basado en Receptor RTL-SDR
Uso de software libre y gratuito.



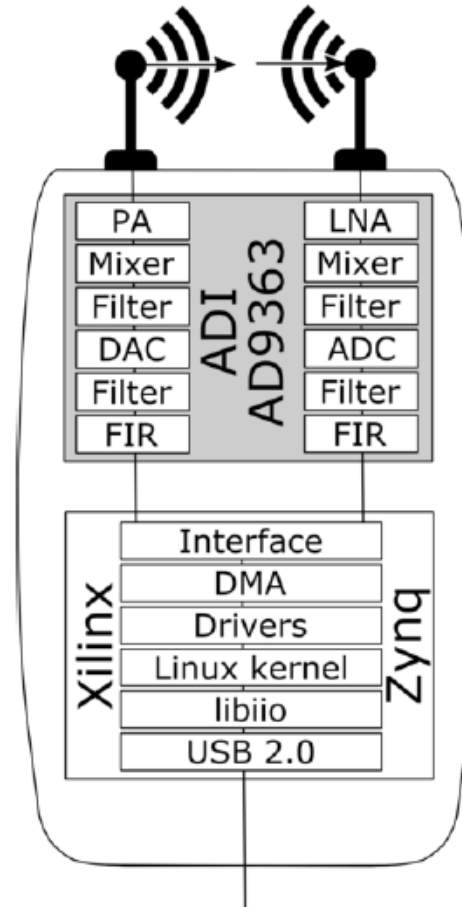


ADALM PLUTO



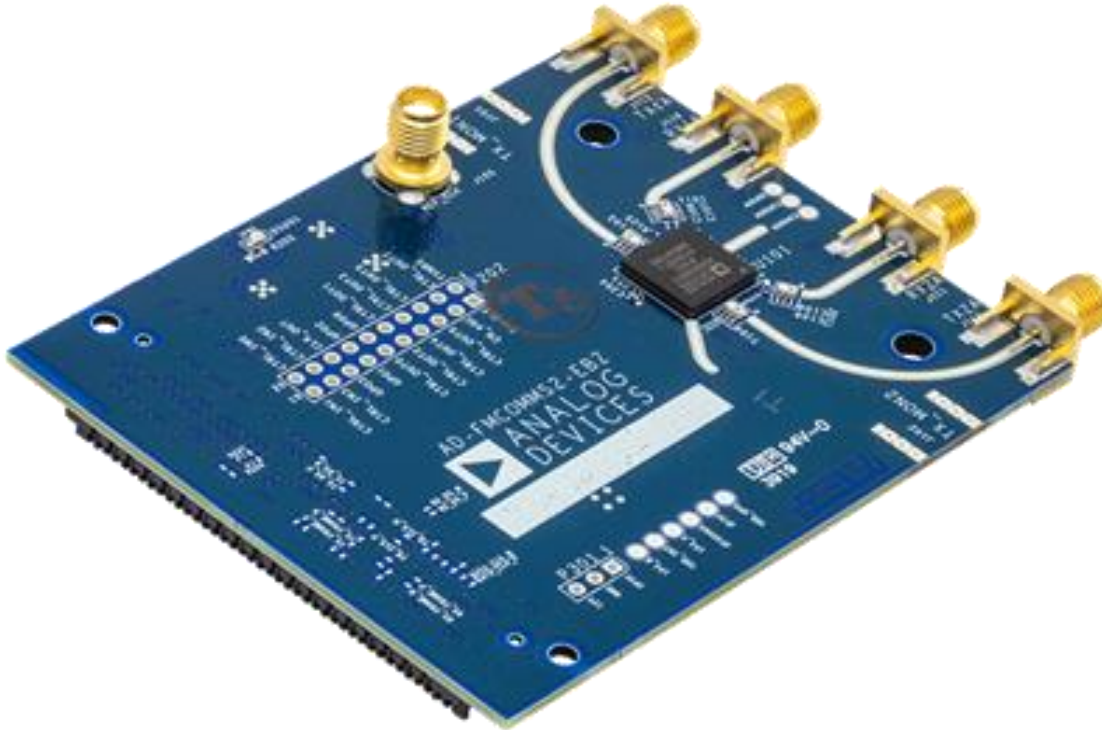
- Full duplex / Half duplex.
- 325MHz - 3,8GHz (extendible a 60MHz - 6GHz)
- ADC/DAC Sample Rate 65.2 kSPS to 61.44 MSPS
- ADC/DAC Resolution 12 bits
- Frequency Accuracy ± 25 ppm
- RBW= máximo 20MHz (puede estar limitado a máx 5MHz).
- Impedancia de entrada de 50 Ohms.

ADALM PLUTO



- ▶ Runs Linux inside the device
- ▶ Uses Linux's IIO framework to expose I/Q data and control
- ▶ Multi-Function Device
 - Native IIO over USB
 - Serial over USB
 - Ethernet over USB
 - Mass Storage
 - Device Firmware Update
- ▶ Host
 - USB dongles

FSCOMMS4



- **Dos salidas y dos entradas full duplex**
- **Rango de operación: 70 MHz – 6,0 GHz)**
- **RBW= <200 kHz to 56 MHz**
- **12 bits de resolución**
- **Superior receiver sensitivity with a noise figure < 2.5 dB**
- **RX gain control**
- **Real-time monitor and control signals for manual gain**
- **Independent automatic gain control**

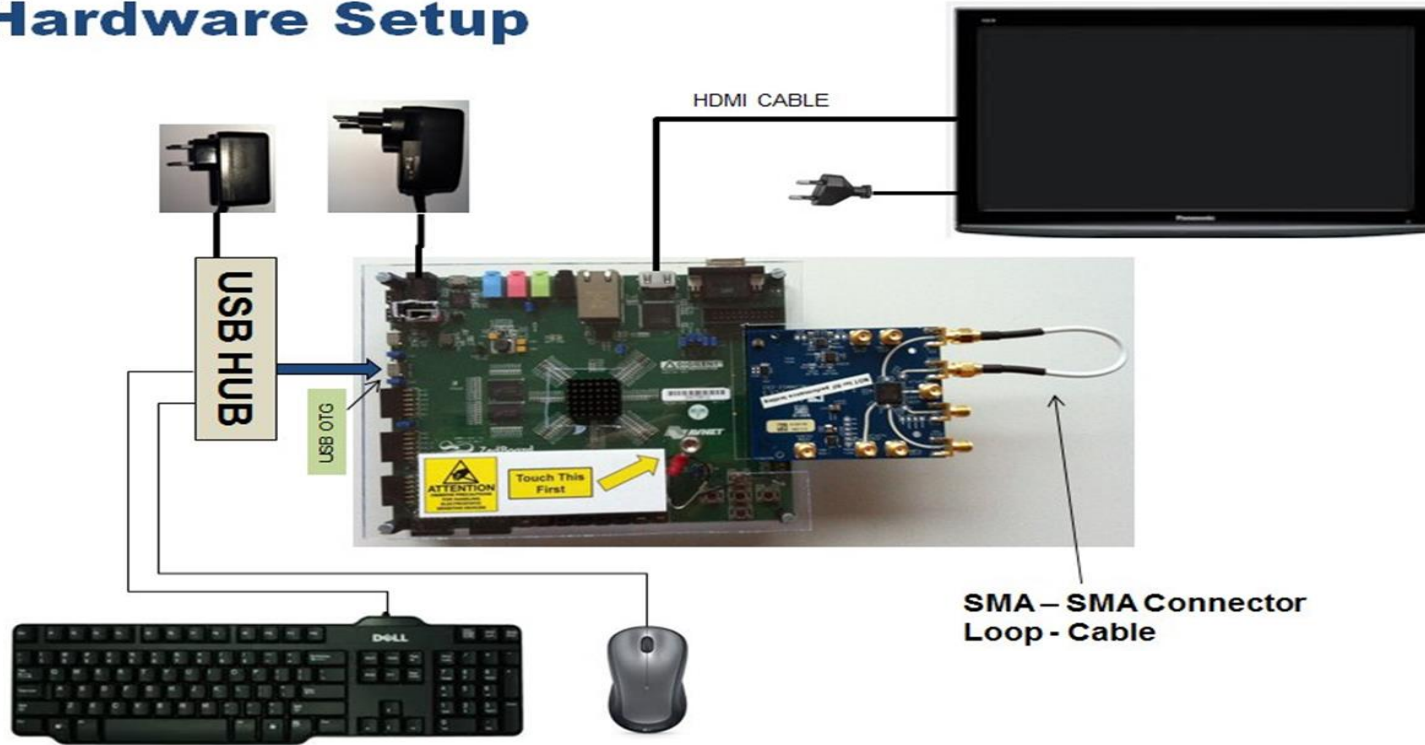
FSCOMMS2 + Zedboard



Plataforma SDR: Xilinx Zedboard + Analog Devices

FSCOMMS4 (60MHz - 6GHz)

Hardware Setup



RTL2832U



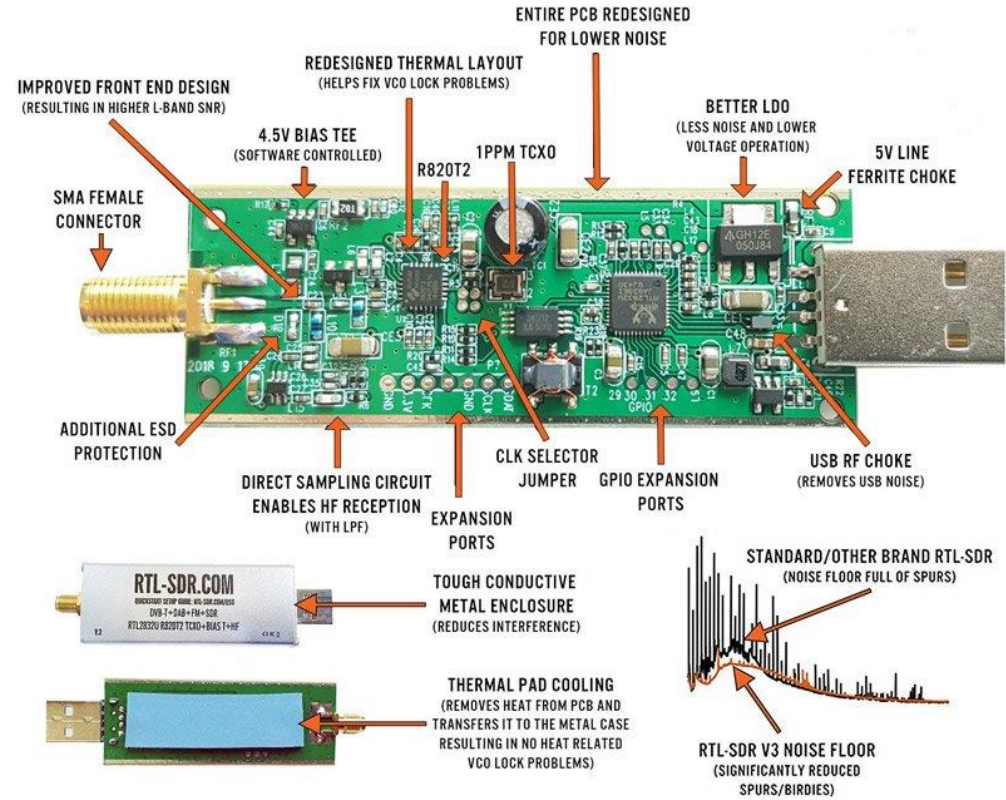
- Solo recepción.
- Rango de operación: 35MHz - 1,8GHz
- RBW=
- 8 bits de resolución
- Encapsulado de aluminio (disipador pasivo)
- Disponible con impedancia de entrada de 50 Ohms y de 75 Ohms.
- <1 PPM temperature compensated oscillator (TCX)

RTL2832U

CHOOSE A GENUINE RTL-SDR BLOG V3



<https://www.rtl-sdr.com/>





Original RTL-SDR Blog V3

- Rounded enclosure
- Full website URL written on body
- Two diagonally offset screws on each side
- Newer units have logo on the back
- Green PCB with thermal pad on bottom
- NSY production QC sticker on back
- Newer units say R860 instead of R820T



Fake RTL-SDR Blog V3 Clones:

- Flat enclosure
- May say "RTL.SDR", "RTL-SDR V3 Pro", or be unmarked
- Four screws per side panel
- May not have bias tee, HF or TCXO features despite advertising
- No SMA nut, or nut without washer
- PCB sits loosely inside enclosure
- May have significantly more spurs + noise
- No logo on the back
- Yellow double stacked PCB, or blue PCB
- May not have thermal pad
- Signals may be distorted with mysterious high pitched whine in the audio spectrum

Clone sellers may also use images of the original
Please try to order from reputable sellers if not ordering directly from our stores.



New Sophisticated Fake V3 Clones

- Looks exactly like an original V3 except for minor differences
- Side panel screws are not diagonally offset
- No NSY QC sticker
- Listings may use our original graphics



<https://www.rtl-sdr.com/>

Hack RF ONE



- 1 MHz to 6 GHz operating frequency
- Half-duplex transceiver
- Up to 20 million samples per second
- 8-bit quadrature samples (8-bit I and 8-bit Q)
- SMA female antenna connector
- SMA female clock input and output for synchronization

Especificaciones relevantes

- Frecuencia mínima y máxima de operación.
- Solo recepción, half- duplex o full-duplex.
- Ancho de banda de tiempo real (RBW).
- Cantidad de bits ADC/DAC.
- Conectores de entrada y salida de RF.
- Encapsulado.