# Digital Communications Laboratory

## (Chapter One - Introduction and Background)

**Laboratory Manual**

**Session One**

**Website:** https://bit.ly/dclabAUT

**Email:** digitalcomslab[at]gmail[dot]com

Pouya Agheli & Mohammad J. Emadi

Electrical Engineering Department

Amirkabir University of Technology (Tehran Polytechnic)

Autumn 2020

# Outline

# Section One

An Introduction to Communication Systems

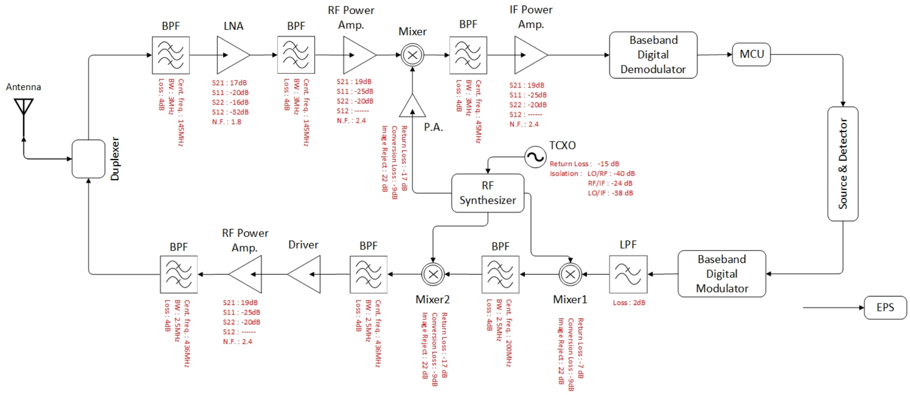
# Communication Systems

**What is a communication system?!**
A system that aims to transfer information from a node to the other one within a communication network.

**Essential Elements of a communication system:**

- Data Source
- Detector
- Source encoder
- Source decoder
- Modulator
- Demodulator
- Mapper
- Demapper
- Up-converter
- Down-converter
- Channel encoder
- Channel decoder
- Generic filter
- Pulse shaping filter
- Matched filter
- Amplifier
- Medium
- Channel

# Transmission and Reception Chains



# Section Two

Different Communication Mediums

# Communication Channels

1. **Wired channels:**
   - Twisted copper wires: Telephone and xDSL
   - Coaxial cable: Internet, broadcast, multicast, and TV
   - Optical fiber: FTTx and VDSL

2. **Wireless channels:**
   - Radio frequency (RF)
   - Microwave (mmWave)
   - Satellite radio
   - Infrared (IR)

**Note:** No matter the type of modulation and demodulation, whether analog or digital, all communication channels are analog.

# Section Three

Design Challenges of Communications Systems

# Design Challenges

1. **Channel effects:**
   - Interference
   - Propagation delay
   - Frequency and phase shift
   - Large-scale fading (in wireless):
     1. Path-loss
     2. Shadowing
   - Small-scale fading (in wireless):
     1. Multi-path delay spread: Flat or frequency-selective fading
     2. Doppler spread: Slow or fast fading

2. **Noise:**
   - Thermal noise at receiver

$$N = k_B \times T_0 \times B_s \times NF,$$

where $k_B = 1.381 \times 10^{-23}$ [Joule/Kelvin], $T_0 = 290$ [Kelvin], $B_s$, and $NF$ are Boltzmann constant, noise temperature, system bandwidth and noise figure, respectively.

# Section Four

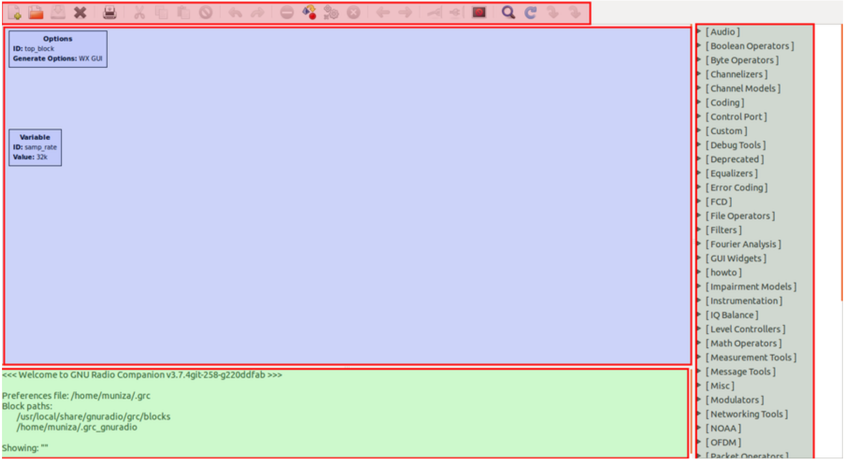
Overview of GNU Radio Software

# An Overview of GNU Radio

- GNU Radio is a framework that enables users to design, simulate, and deploy highly capable real-world radio systems. It is a highly modular, "flow-graph"-oriented framework that comes with a comprehensive library of processing blocks that can be readily combined to make complex signal processing applications
- GNU Radio has been used for a huge array of real-world radio applications, including audio processing, mobile communications, tracking satellites, radar systems, GSM networks, Digital Radio Mondiale (DRM), and much more - all in computer software
- GNU Radio, by itself, not a solution to talk to any specific hardware. Nor does it provide out-of-the-box applications for specific radio communications standards (e.g., IEEE802.11, ZigBee, LTE, etc.,), but it can be (and has been) used to develop implementations of basically any band-limited communication standard

**Check: https://wiki.gnuradio.org/index.php/Tutorials**

- GRC has five parts: Library, Toolbar, Terminal, and Workspace.



# Flow-graph, Block, and Item

**What do flow-graph and block mean in GRC?!**

- Flow-graphs are graphs (as in graph theory) through which data flows
- Many GNU Radio applications contain nothing other than a flow-graph
- The nodes of such a graph are called blocks, and the data flows along the edges,
- Any actual signal processing is done in the blocks
- A Ideally, every block does exactly one job - this way GNU Radio stays modular and flexible. Blocks are usually written in C++ (might also be Python); writing new blocks is not very difficult

**What does item mean in GRC?!**

- In general, we call whatever a block outputs an item
- An item can be anything that can be represented digitally
- The most common types of samples are real samples (as before), complex samples (the most common type in software-defined radio), integer types, and vectors of these scalar types
- An item can be anything, a sample, a bunch of bits, a set of filter coefficients or whatever

# Simulations with GNU Radio

**Is GR a tool for simulation?!**

- GNU Radio is not primarily intended for simulations, but often these are an important step in the development of signal processing code
- Using GNU Radio can even be advantageous at times, since the simulation code and the code to actually transmit over the air is always the same
- There never is a reason that completely rules out GNU Radio as a simulation tool. However, there are situations when other tools might be suited better for the task, in particular when development time is an issue
- If you're not planning to ever take your code live on the air, and just need simulations results e.g. to create graphs for a research paper, other tools might be more suitable

# Summary

Here's what you should know by now:

- All signal processing in GNU Radio is done through flow-graphs
- A flow-graph consists of blocks. A block does one signal processing operation, such as filtering, adding signals, transforming decoding, hardware access or many others,
- Data passes between blocks in various formats, complex or real integers, floats or basically any kind of data type you can define
- Every flow-graph needs at least one sink and source

# Section Five

Overview of Software-Defined Radio Modules and Dongles

# Software-Defined Radio (SDR)

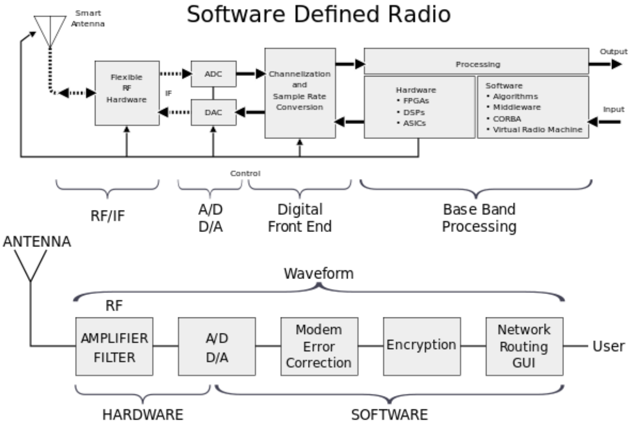
- A software-defined radio (SDR) is defined as a radio in which some or all of the physical layer functions are software-defined, that is, they use software processing within the radio system or device to implement operating (but not control) functions

  From: Short-range Wireless Communication(Third Edition), 2019

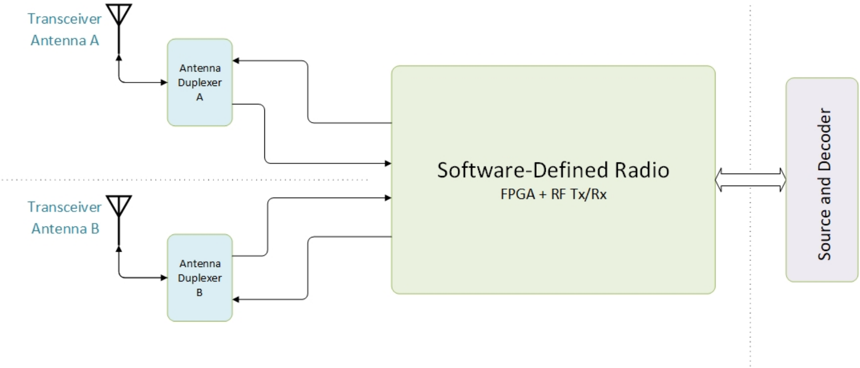
- Software-defined radio (SDR) is a radio communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators and demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system

  From: www.sdr-radio.com

# SDR Concept


Software Defined Radio
Smart Antenna
Flexible RF Hardware
IF
ADC
DAC
Channelization and Sample Rate Conversion
Processing
Hardware
• FPGAs
• DSPs
• ASICs
Software
• Algorithms
• Middleware
• CORBA
• Virtual Radio Machine
Output
Input
Control
RF/IF
A/D D/A
Digital Front End
Base Band Processing
ANTENNA
Waveform
RF
AMPLIFIER FILTER
A/D D/A
Modem Error Correction
Encryption
Network Routing GUI
User
HARDWARE
SOFTWARE

# Transceiver System based on SDR

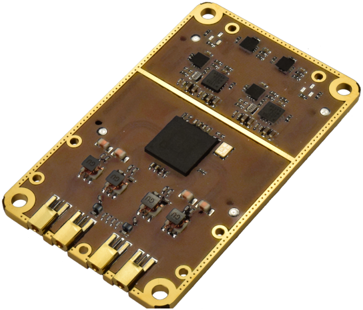


# SDR Transceiver Modules - NanoCom SR2000

- High-speed QPSK transceiver for point-to-point S-band communications
- Frequency and time division duplex support
- Symbol rate 500 kBd to 2 MBd
- S-band operation (1980-2290 MHz)
- Dual-modem support allows two simultaneous RF links
- Adjustable output power
- PC104 form factor
- Interfaces: CAN, LVDS, I2C and SSMCX antenna connectors
- Operational temperature: -40C to +85C

# SDR Transceiver Modules - NanoCom TR600

- AD9361 Transceiver
- Band: 70 MHz to 6 GHz
- Channel bandwidth is tunable from 200 KHz to 56 MHz
- RF $2 \times 2$ transceiver with integrated 12-bit DACs and ADCs
- Supports TDD and FDD operation
- Multichip synchronization
- LVDS/single-ended digital BB interface
- Requires a NanoDock SDR
- Flight proven
- Precision milled anodized aluminum heat sink to control thermal load and provide EMI shielding
- Temperature and current sensors,
- EEprom for persistent configuration storage



# SDR Transceiver Modules - HackRF 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)
- Compatible with GNU Radio, SDR, and more
- Software-configurable RX and TX gain and baseband filter
- Software-controlled antenna port power (50 mA at 3.3 V)
- SMA female antenna connector
- SMA female clock input and output for synchronization
- Convenient buttons for programming
- Internal pin headers for expansion
- High-Speed USB 2.0
- open source hardware

# SDR Transceiver Modules - USRP B210

- First fully integrated, two-channel USRP device with continuous RF coverage from 70 MHz – 6 GHz
- Full duplex, MIMO (2 Tx & 2 Rx) operation with up to 56 MHz of real-time bandwidth (61.44MS/s quadrature)
- Fast and convenient SuperSpeed USB 3.0 connectivity
- GNURadio and OpenBTS support through the open-source USRP Hardware Driver$^{TM}$ (UHD)
- Open and reconfigurable Spartan 6 XC6SLX150 FPGA (for advanced users)
- Early access prototyping platform for the Analog Devices AD9361 RFIC, a fully integrated direct conversion transceiver with mixed-signal baseband

# SDR Receiver Dongles - RTL-SDR


- Each type of RTL-SDR dongle has somewhat different features compared to others.

# Assignment

1. Install GNU Radio on your computer
2. Find and briefly introduce two new usages of SDR in industry performed after 2018
3. Concisely describe all available data types in GRC

**Note:**

- Due: Oct. 6, 2020 (12 PM)
- Upload your project here.