SOFTWARE DEFINED RADIO
USR SDR WORKSHOP, SEPTEMBER 2017
PROF. MARCELO SEGURA

SESSION 1: SOFTWARE TOOLS
**BRIEF HISTORY**

- **USRP**
  - Libusrp
  - Libusrp-gnuradio
  - Python dboard code
  - C++ dboard code
  - Usrp_* examples and utils

- **USRP2**
  - Libusrp2 (linux only)
  - libusrp2-gnuradio
  - C dboard code in FW
  - Usrp2_* examples and utils

- **NOT SCALE**
UHD is an C++ API, that allow to interface a host with any USRP boards.

- Instantiate device object on Matlab or GNURadio.
  - SET/GET FUNCTION
  - SEND/ RECEIVE SAMPLES (HOST: Ring buffer, SDR: BRAM FIFO)
NEXT GEN WIRELESS DEVELOPMENT

REQUIRES AT LEAST 7 DIFFERENT SKILLS!!!
We primarily use **Comm & DSP** toolbox

### Algorithms, Waveforms, Measurements
- Communications System Toolbox
- Phased Array System Toolbox
- LTE System Toolbox
- WLAN System Toolbox

### RF Front End
- RF Toolbox
- RF Blockset

### Antennas, Antenna Arrays
- Antenna Toolbox
- Phased Array System Toolbox

### System Architecture

- **DSP Algorithms**
  - Baseband
  - Digital Front End

- **Digital PHY**
  - Baseband
  - Digital Front End

- **RF Front End**
  - DAC
  - PA
  - ADC
  - LNA

- **Antenna Design**
  - Antenna

- **Mixed-signal**
  - Simulink
  - DSP System Toolbox
  - Control System Toolbox

- **Communications System Toolbox**
- Phased Array System Toolbox
- LTE System Toolbox
- WLAN System Toolbox

---

**DR ING MARCELO SEGURA**

**USC SDR WORKSHOP**
DESIGN WORKFLOW

System Architecture

DSP/Algorithms

Digital PHY

Software

Digital Hardware

HDL and C code generation

Instrument Control Toolbox

SDR Support Packages

Communications System Toolbox

Fixed-Point Designer

HDL Coder

Embedded Coder

RF Test Instruments

Software-Defined Radio

Multi-vendor hardware support
PROTOTYPING WORKFLOW

1. Algorithm design: Simulation Model
2. SDR testing with live signal I/O
3. Generate HDL Code: Implementation Model
4. Verify on SDR hardware

SDR Hardware

- Analog Front-End
- Pre-configured FPGA Algorithms
- User-Designed Algorithm
- User-Designed Algorithms

MATLAB & Simulink

RF I/O

Streaming I/O

Desktop Design and Simulation

Information
• Start with base HDL reference design from ADI
PROTOTYPING WORKFLOW

- Create your own IP core
- Performance
- Interface speed
- Hardware accelerators
Matlab System objects are the base building block to communicate with SDR.

System objects are designed specifically for implementing and simulating dynamic systems with inputs that change over time.

Matlab objects have a predefined structure that allow to control and send/receive data from SDR.
SUPPORT PACKAGE

- UHD ON A SYSTEM BLOCK
The sampling rates should be matched with the master clock rate.
SUPPORT PACKAGE

- SDRu RX block:
- Serial Number of B200 mini
- Channel Mapping: 1 SISO, [1 2] MIMO
- Center Freq: LO frequency
- LO Offset: offset freq to avoid LO leakage or self interference.
- Gain: overall gain including both analog and digital components.
- Transport data: 16 bit default
- Samples per frame: to optimize Ethernet packet.
- Burst mode: to test systems that cannot run on real time, avoiding overrun or Underrun.
rx = comm.SDRuReceiver('Platform','B210','SerialNum','ECR042DBT')

rx =

System: comm.SDRuReceiver

Properties:
- Platform: 'B210'
- SerialNum: 'ECR042DBT'
- ChannelMapping: 1
- CenterFrequencySource: 'Property'
- CenterFrequency: 2450000000
- ActualCenterFrequency: 2450000000
- LocalOscillatorOffsetSource: 'Property'
- LocalOscillatorOffset: 0
- ActualLocalOscillatorOffset: 0
- GainSource: 'Property'
- Gain: 8
- ActualGain: 8
- ClockType: 'Internal'
- MasterClockRate: 32000000
- ActualMasterClockRate: 32000000
- DecimationFactorSource: 'Property'
- DecimationFactor: 512
- ActualDecimationFactor: 512
- TransportDataType: 'int16'
- OverrunOutputPort: false
- SampleRate: 1
- OutputDataType: 'Same as transport data type'
- FrameLength: 362
- EnableBurstMode: false

tx = comm.SDRuTransmitter('Platform','B210','SerialNum','ECR042DBT')

tx =

comm.SDRuTransmitter with properties:
- Platform: 'B210'
- SerialNum: 'ECR042DBT'
- ChannelMapping: 1
- CenterFrequencySource: 'Property'
- CenterFrequency: 2.4500e+09
- ActualCenterFrequency: 0
- LocalOscillatorOffsetSource: 'Property'
- LocalOscillatorOffset: 0
- ActualLocalOscillatorOffset: 0
- GainSource: 'Property'
- Gain: 8
- ActualGain: 0
- ClockSource: 'Internal'
- MasterClockRate: 32000000
- ActualMasterClockRate: 0
- InterpolationFactorSource: 'Property'
- InterpolationFactor: 512
- ActualInterpolationFactor: 0
- TransportDataType: 'int16'
- UnderrunOutputPort: false
- EnableBurstMode: false
Lab 1: INSTALLING B200 on Matlab

Setup and Configuration

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sdrunload</td>
<td>Load FPGA and firmware images for USRP® radio</td>
</tr>
</tbody>
</table>

Hardware Discovery

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>findsdru</td>
<td>Find and report status for all USRP® devices connected to host computer</td>
</tr>
<tr>
<td>getsDRuDriverVersion</td>
<td>Report UHD™ version number for the support package</td>
</tr>
<tr>
<td>probesdru</td>
<td>Provides detailed USRP® radio information</td>
</tr>
</tbody>
</table>

Radio Management

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setsdrup</td>
<td>Set USRP® radio IP address</td>
</tr>
<tr>
<td>comm.SDRuReceiver</td>
<td>Receive data from USRP® device</td>
</tr>
<tr>
<td>comm.SDRuTransmitter</td>
<td>Send data to USRP® device</td>
</tr>
</tbody>
</table>