



ADALM-PLUTO培训

北京威视锐科技有限公司

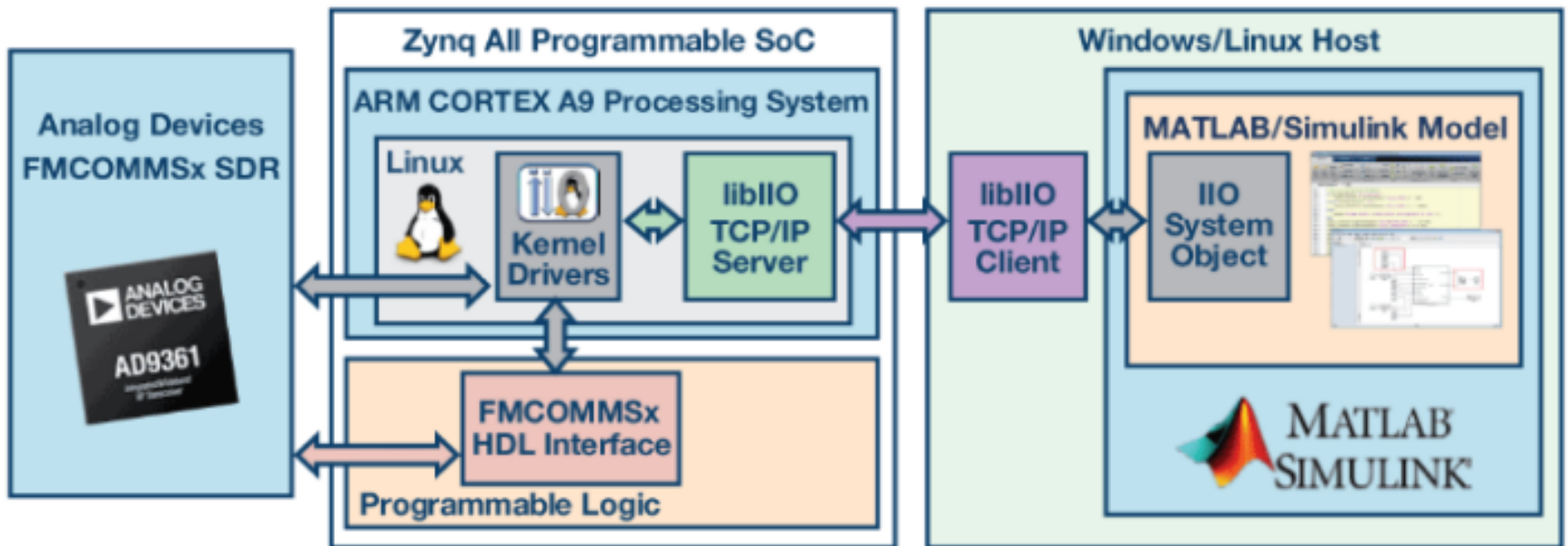
V3 Technology (Beijing) Ltd

威视锐旗下品牌:



专题二 PlutoSDR软件配置

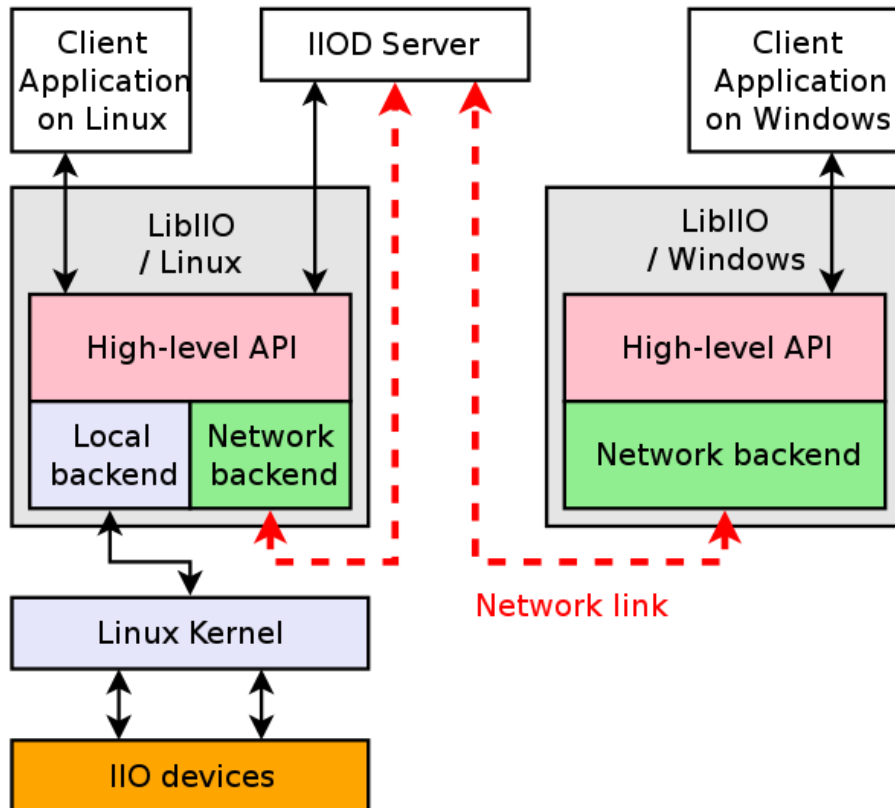
-
- 软件交互流程
- libiio
- MATLAB与PlutoSDR的数据交互
- SIMULINK与PlutoSDR的数据交互
- 无线通信原型的设计



2.1 libiio

libiio库是硬件底层细节的抽象，提供了简单但完整的编程接口，可用于绑定各种语言（C、C++、C#、Python）的高级项目。





Analog Devices 开发了 Libiio 库，可简化用于连接 Linux Industrial I/O (IIO) 器件（例如 Pluto 上的 AD9363）的软件开发工作。开源 (GNU Lesser General Public License V2.1) 库抽象出硬件的低层细节，并提供一个可用于高级项目的简单但完整的编程接口。

您可以使用 Libiio 在项目原型设计阶段与 Pluto SDR 进行接口连接，以便向或从 MATLAB、Simulink 或 GNURadio 等工具的模式发送或接收样本流

2.2 MATLAB与PLUTO的数据交互

这一部分讨论ADI公司提供的软件工具，其支持MATLAB和Simulink模型与SDR平台直接互动。此外还会说明如何利用这些工具验证无线通信模型。



PLUTO通过libiio向上层控制机（MATLAB）提供了简单易用的设备配置和数据通道接口，通过通用的代码配置格式，可以方便的对下层硬件进行快速配置并可实时修改。通过TCP/IP网络协议确保数据的可靠传输。

工作模式：

➤ 单工

只作为发射机或者接收机

➤ 半双工

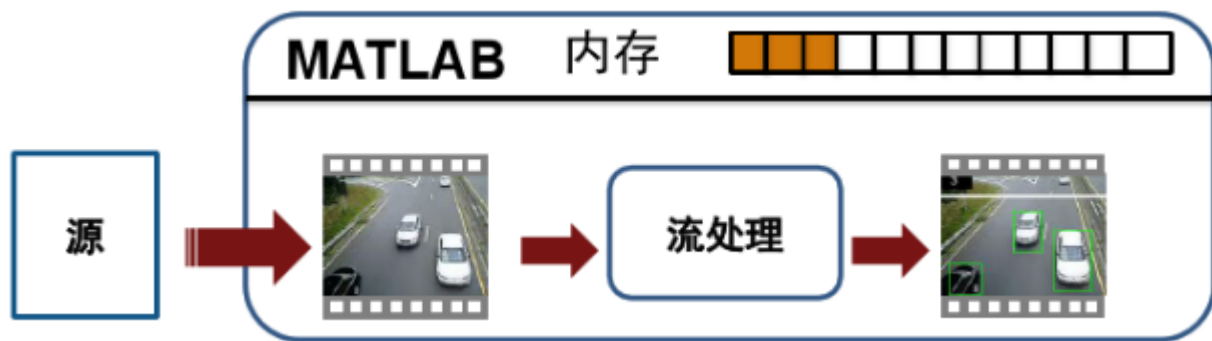
发送通道作为信号源，实时接收数据

➤ 全双工

实时发射和接收数据

2.2.1 系统对象

- 支持流处理的一类MATLAB对象
简化了流应用的数据访问过程
- 管理来自文件或网络的数据流
- 包含大量的流数据处理算法
- 支持信号处理、通信和相控阵应用



2.2.2 IIO系统对象

- 基于MathWorks系统对象规范
- 通用型IIO系统对象
- 通过libiio服务器/客户端基础设施来完成
- 专门针对用于处理大数据流的迭代计算而优化，能够在环内无线配置中自动实现 Pluto SDR 与 MATLAB 和 Simulink环境之间的流数据



2.2.3 代码配置

创建IIO系统对象：

```
% System Object Configuration
s = iio_sys_obj_matlab; % MATLAB libiio Constructor
s.ip_address = ip;
s.dev_name = 'ad9361';
s.in_ch_no = 2;
s.out_ch_no = 2;
s.in_ch_size = length(tx_data);
s.out_ch_size = length(tx_data)*4;
```

- 构建IIO系统对象
- IP地址
- 目标设备名称
- 输入、输出通道的大小和采样点数



利用IIO系统对象设置AD9363属性：

```
% Configure the FIR filter on AD9361
s.writeFirData(fir_data_file);

input = cell(1, s.in_ch_no + length(s.iio_dev_cfg.cfg_ch));
output = cell(1, s.out_ch_no + length(s.iio_dev_cfg.mon_ch));

% Set the attributes of AD9361
input{s.getInChannel('RX_LO_FREQ')} = 2.45e9;
input{s.getInChannel('RX_SAMPLING_FREQ')} = samplingrate;
input{s.getInChannel('RX_RF_BANDWIDTH')} = bandwidth;
input{s.getInChannel('RX1_GAIN_MODE')} = 'slow_attack';
input{s.getInChannel('RX1_GAIN')} = 0;
% input{s.getInChannel('RX2_GAIN_MODE')} = 'slow_attack';
% input{s.getInChannel('RX2_GAIN')} = 0;
input{s.getInChannel('TX_LO_FREQ')} = 2.45e9;
input{s.getInChannel('TX_SAMPLING_FREQ')} = samplingrate;
input{s.getInChannel('TX_RF_BANDWIDTH')} = bandwidth;
```

- 频点
- 采样率
- 带宽
- 增益模式
- 数字FIR滤波器

•

发射与接收数据：

发送基带I/Q数据tx_data（16bit量化）到pluto，并接收基带I/Q数据到rx_data（12bit量化）：

```
input{1} = real(tx_data);  
input{2} = imag(tx_data);  
output = stepImpl(s, input);  
I = output{1};  
Q = output{2};  
rx_data = I+1i*Q;
```

2.2.4 配置文件

ad9361.cfg:

将系统对象接口链接到IIO数据通道
或IIO属性

```
1 data_in_device = cf-ad9361-dds-core-lpc
2 data_out_device = cf-ad9361-lpc
3 ctrl_device = ad9361-phy
4 channel = RX_LO_FREQ, IN, out_altvoltage0_RX_LO_frequency,
5 channel = RX_SAMPLING_FREQ, IN, in_voltage_sampling_frequency,
6 channel = RX_RF_BANDWIDTH, IN, in_voltage_rf_bandwidth,
7 channel = RX1_GAIN_MODE, IN, in_voltage0_gain_control_mode,
8 channel = RX1_GAIN, IN, in_voltage0_hardwaregain,
9 channel = RX1_RSSI, OUT, in_voltage0_rssi,
10 channel = TX_LO_FREQ, IN, out_altvoltage1_TX_LO_frequency,
11 channel = TX_SAMPLING_FREQ, IN, out_voltage_sampling_frequency,
12 channel = TX_RF_BANDWIDTH, IN, out_voltage_rf_bandwidth,
```

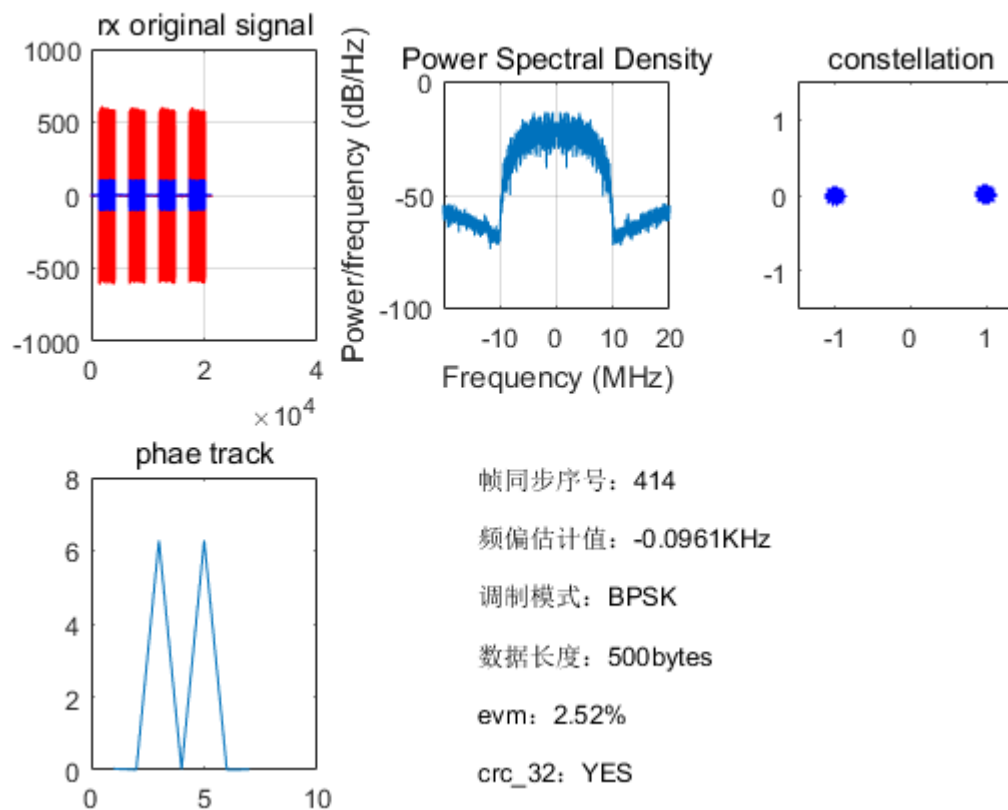
- **data_in_device**: 用于向设备发送数据的linux驱动程序的名称
- **data_out_device**: 用于从设备读取数据的linux驱动程序的名称
- **ctrl_device**: 用于控制和监视设备的linux驱动程序的名称
- **channel**: 通道配置
<通道名称, 通道类型, Linux属性>

2.2.5 个性化开发

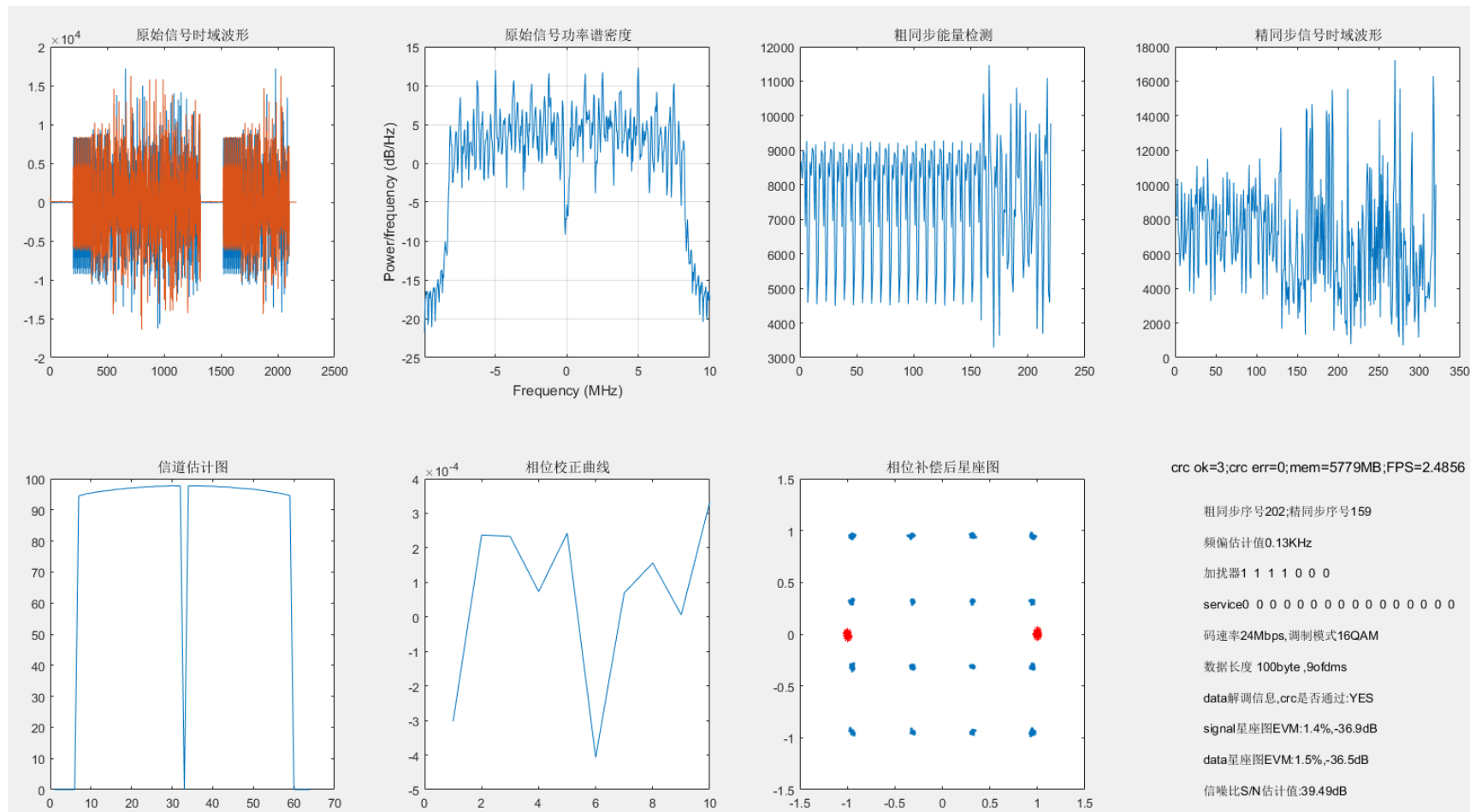
- 利用实时射频信号进行空中实验和现场测试
- 自定义无线功能的快速原型
- 通过动手实践，学习无线通信的概念和设计技巧



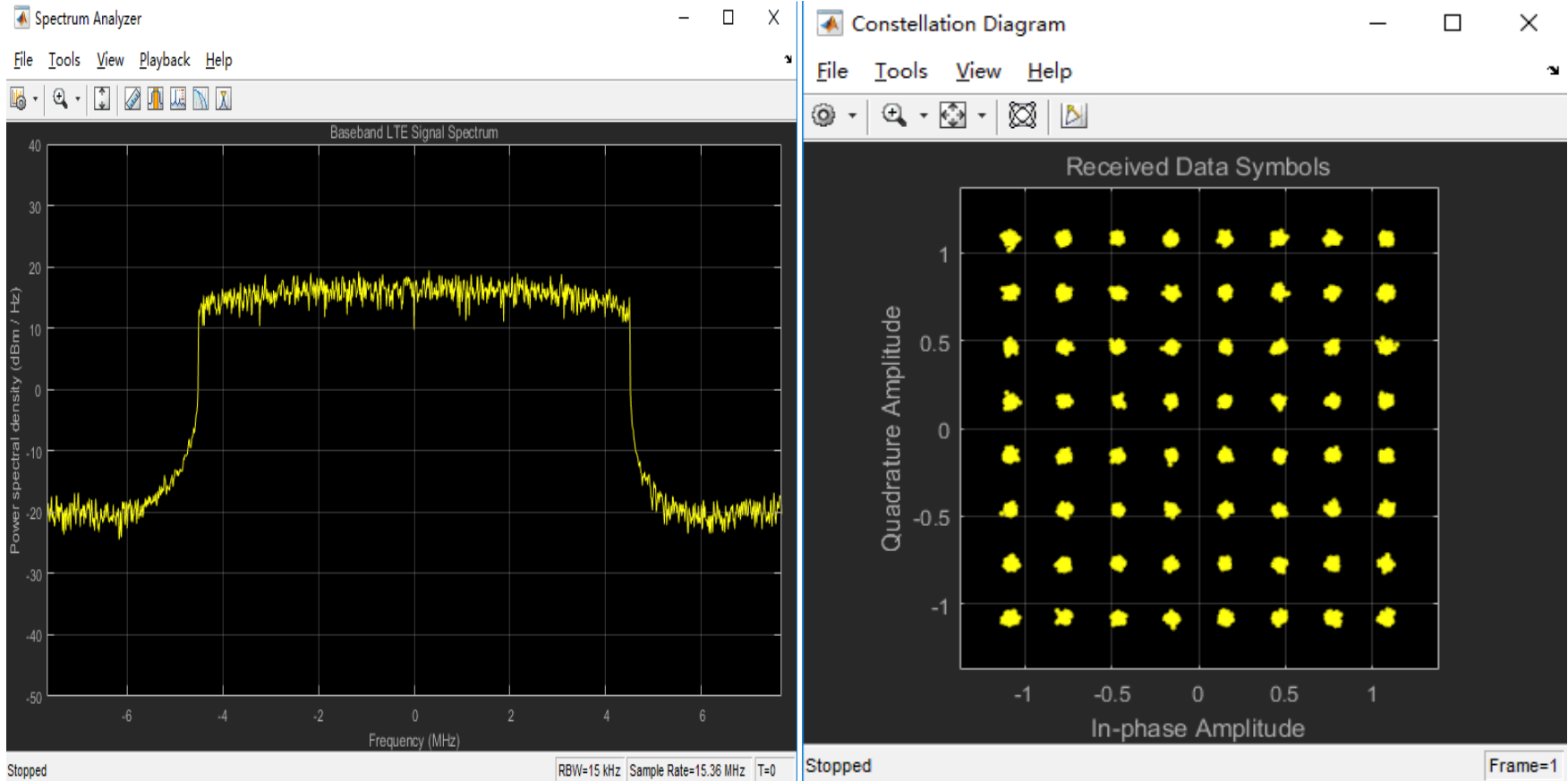
自定义单载波BPSK无线通信系统：



IEEE802.11a标准物理层实现:



LTE:

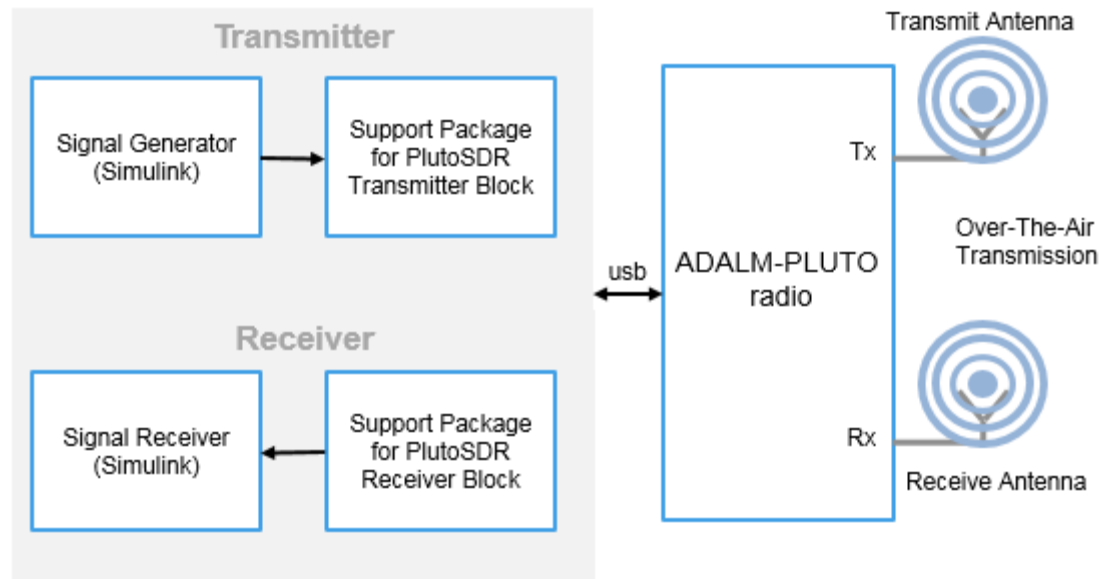


2.2.6 SIMULINK

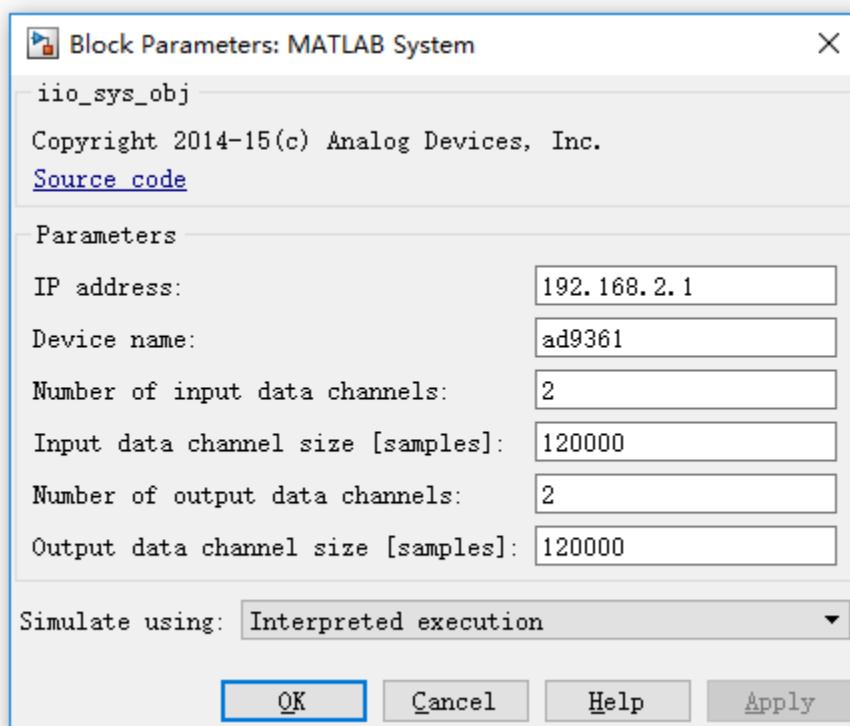
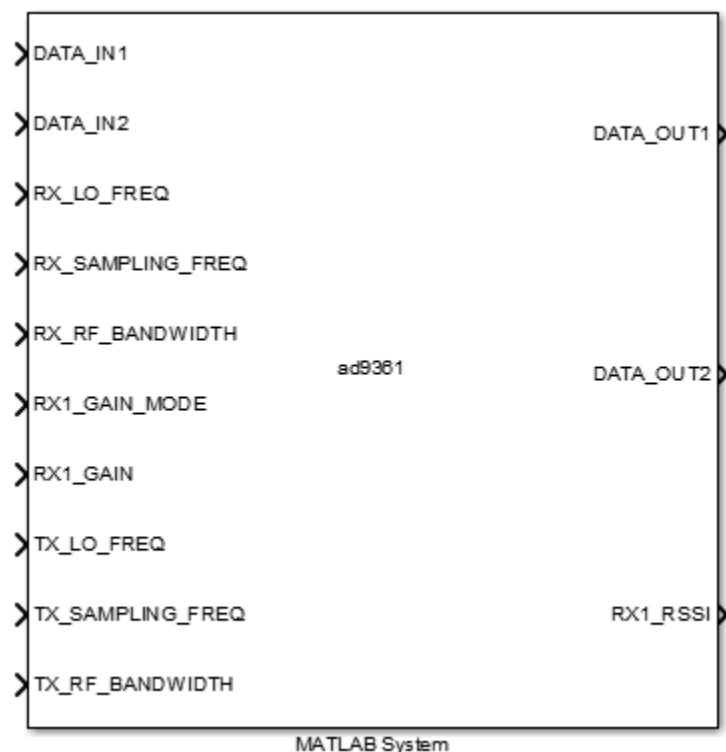
基于模型的通信系统的设计，无需编写代码或只需要少量代码，轻松实现stream模式的通信：

- 建模仿真
- 自动代码生成
- 模型验证



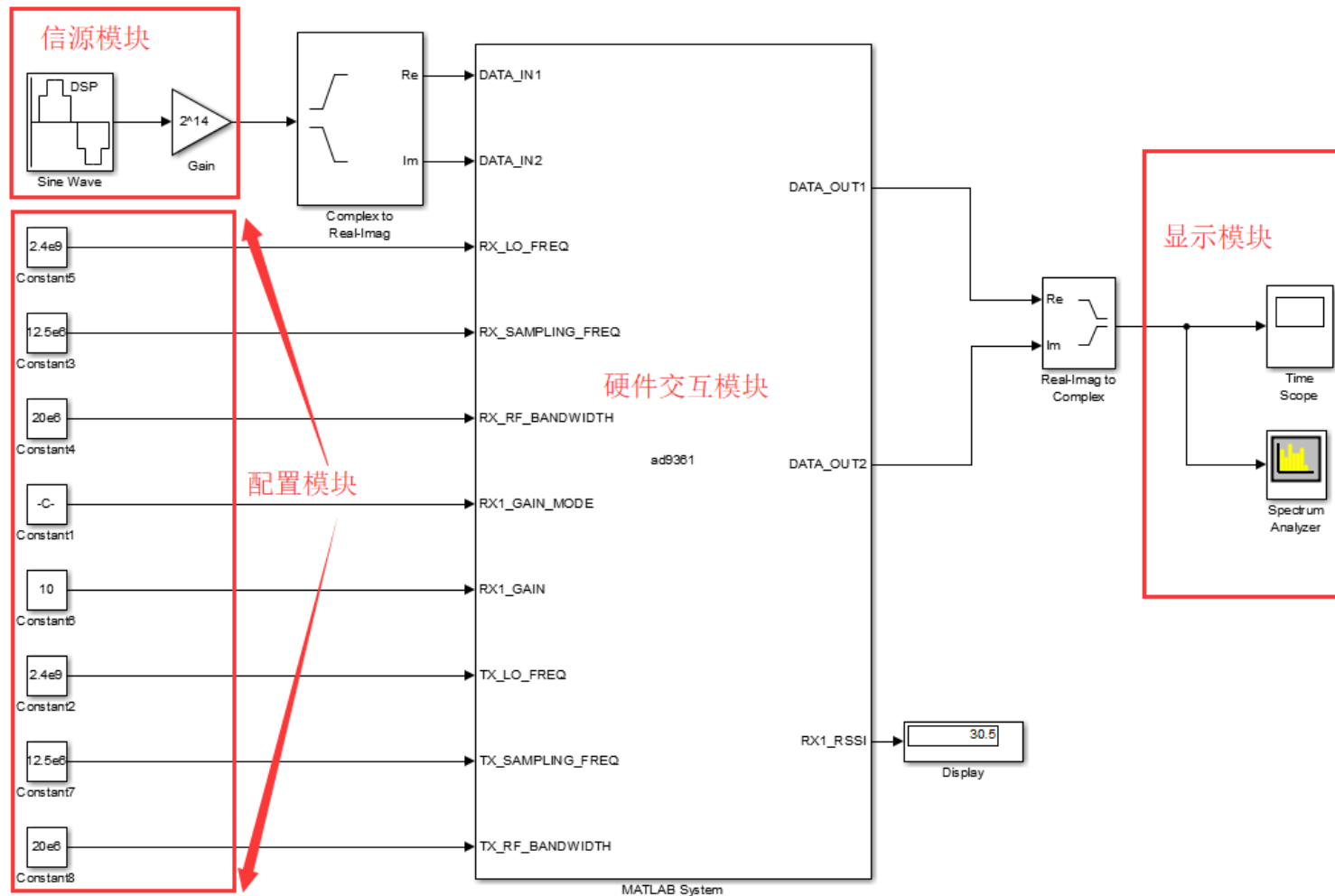


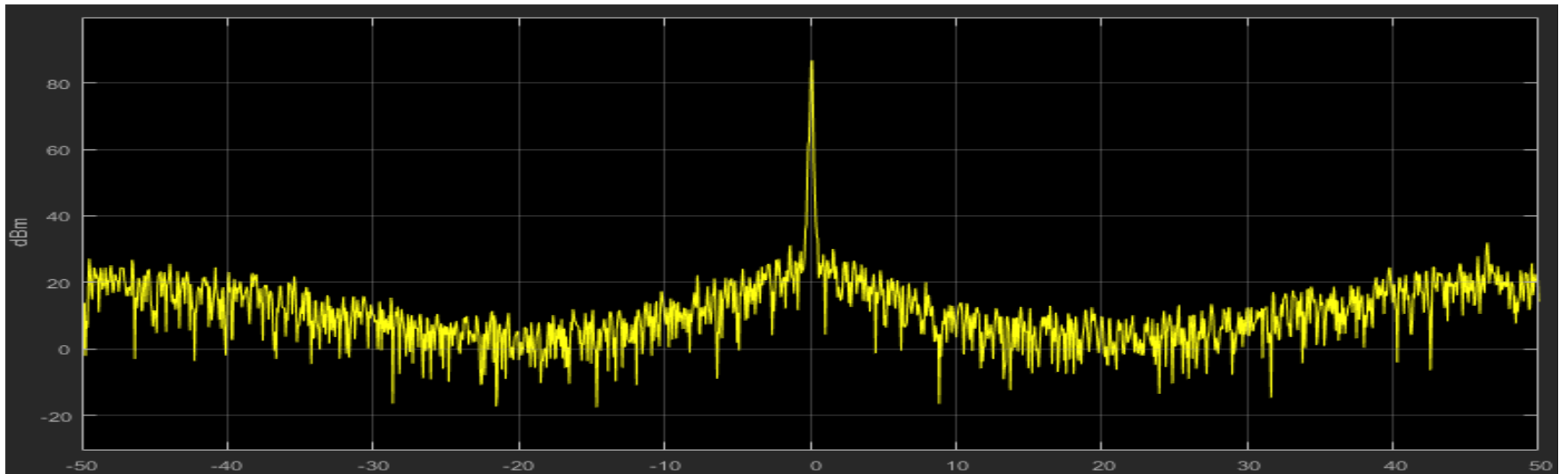
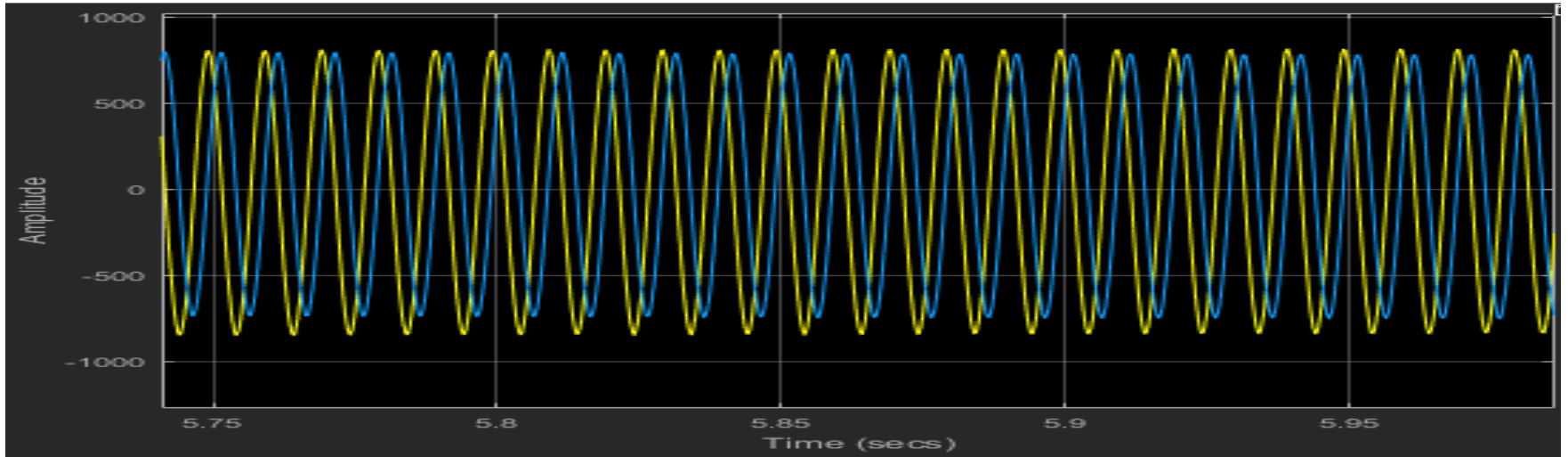
PLUTO提供封装好的SIMULINK硬件配置以及数据交互模块：



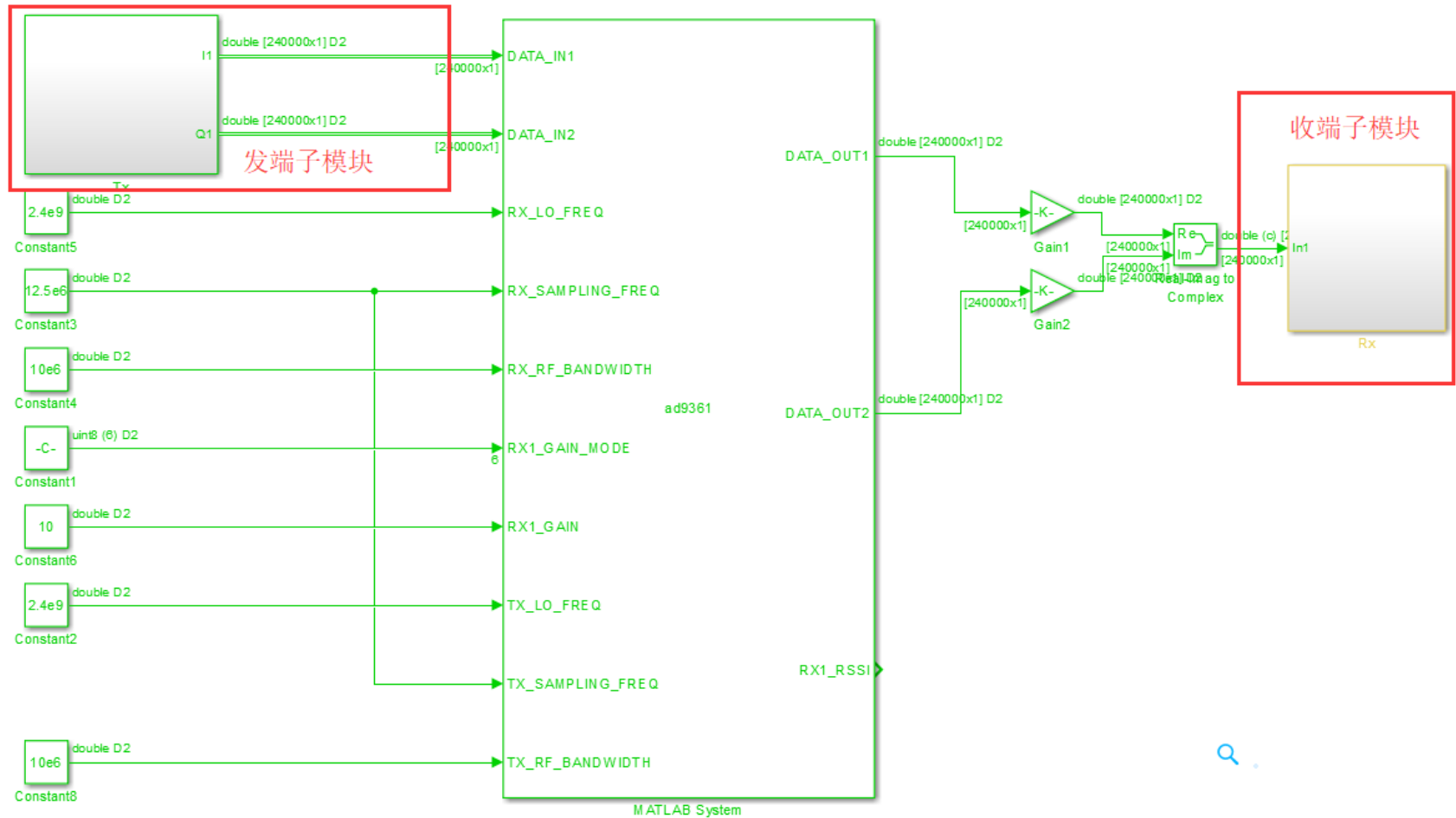
A screenshot of the 'Block Parameters: MATLAB System' dialog box. The dialog has a title bar with a close button (X). The main content area is divided into sections. The first section contains the text 'iio_sys_obj', 'Copyright 2014-15(c) Analog Devices, Inc.', and a link to 'Source code'. The second section, titled 'Parameters', contains several input fields: 'IP address:' with the value '192.168.2.1', 'Device name:' with the value 'ad9361', 'Number of input data channels:' with the value '2', 'Input data channel size [samples]:' with the value '120000', 'Number of output data channels:' with the value '2', and 'Output data channel size [samples]:' with the value '120000'. The third section, titled 'Simulate using:', has a dropdown menu set to 'Interpreted execution'. At the bottom, there are four buttons: 'OK', 'Cancel', 'Help', and 'Apply'.

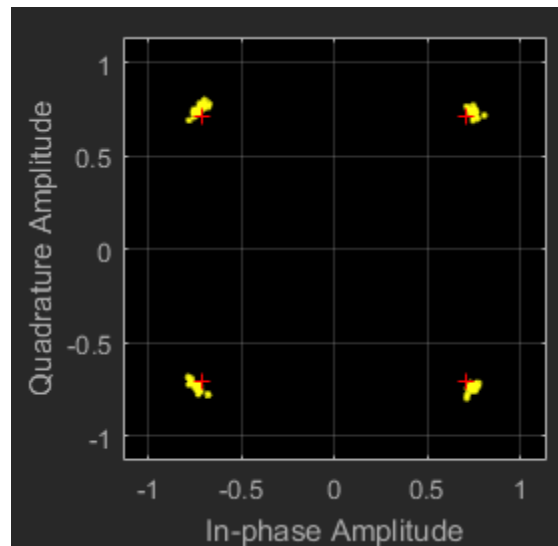
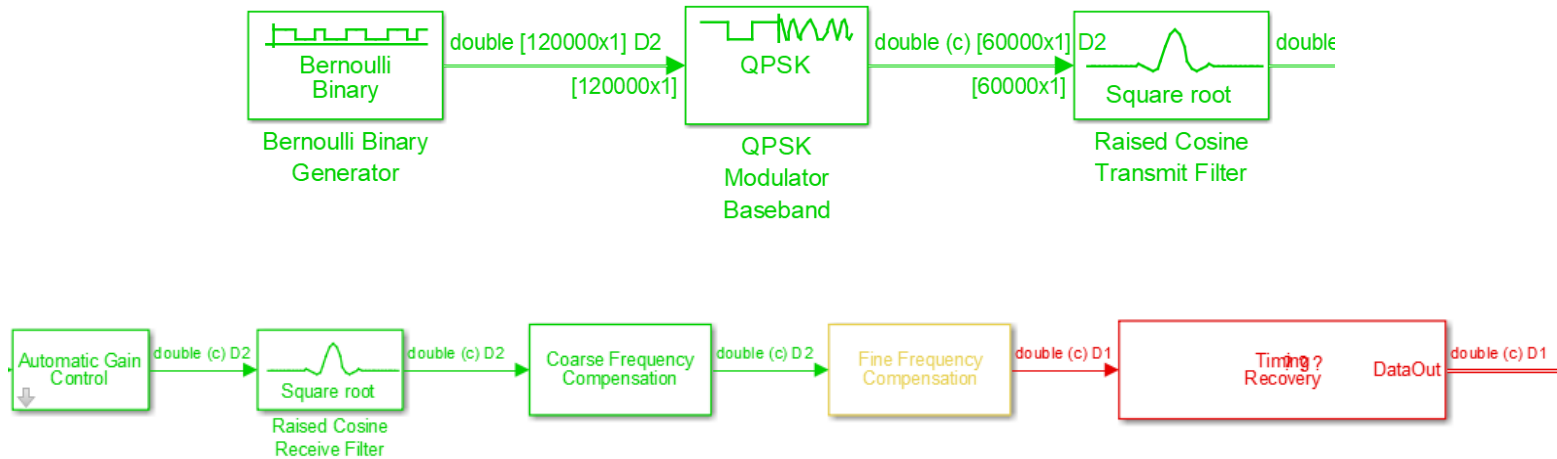
2.2.6.1 单音信号测试





2.2.6.2 QPSK通信系统





总结

- libiio实现对设备的通用抽象
- iio系统对象实现高效的数据流管理
- 通用的代码配置格式
- 灵活的可扩展设计

