

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра телекоммуникационных систем и вычислительных средств
(ТС и ВС)

Отчет по производственной практике
по дисциплине
SDR

по теме:
Архитектура Adalm Pluto SDR. GNU Radio. Построение радио-приёмника

Студент:
Группа ИА-331

К.А Любимов

Преподаватели:
Лектор
Практик
Практик

Калачиков А.А
Ахнашев А.В
Попович И.А

Новосибирск 2025 г.

1 ПРАКТИКА

АРХИТЕКТУРА SDR СИСТЕМЫ

УСТАНОВКА ПО, НАСТРОЙКА УСТРОЙСТВА

I [Ссылка на GitHub](#)

Цель практики:

Узнать, что такое SDR, изучить принципы его работы и внутреннюю архитектуру на базовом уровне. Познакомиться с инструментом GNU Radio и создать с его помощью программу для SDR, позволяющую принимать радио.

Краткие теоретические сведения

Что такое SDR?

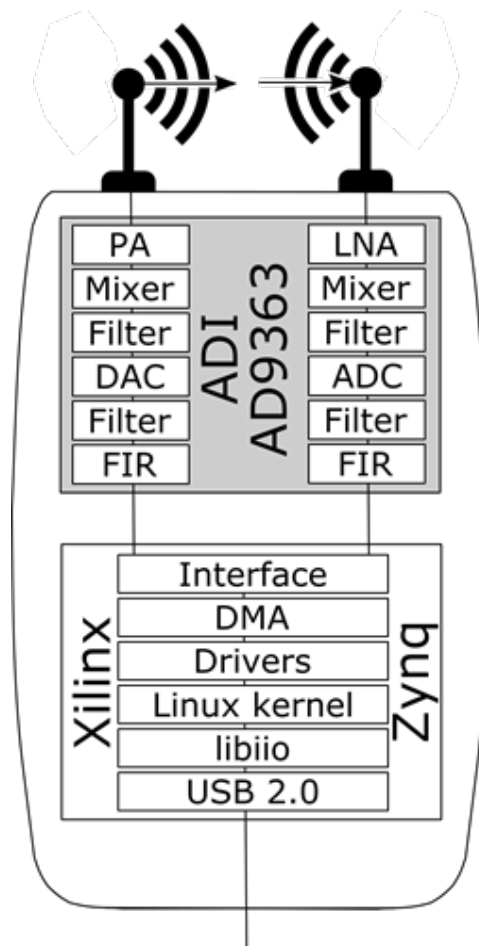


Рис. 1 — ADALM Pluto

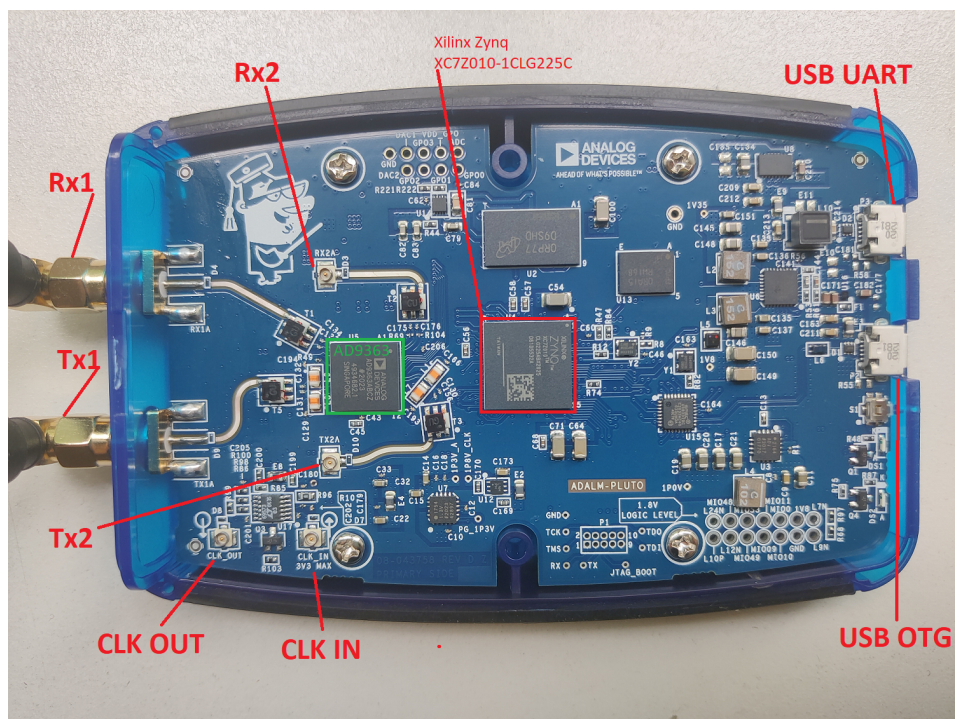


Рис. 2 — ADALM Pluto

Обучающая платформа PlutoSDR может взаимодействовать с:

1. Matlab, Simulink
2. GNU Radio
3. C, C++ при помощи дополнительных библиотек
4. C#
5. Среда языка Python

В начале данного курса мы наладим взаимодействие PlutoSDR с языком Python. Далее напишем программы под другие платформы, сравним разницу по времени обработки сигналов, простоте написания кода и редактирования.

Чип AD9363

Программируемый РЧ передатчик, возможности которого позволяют использовать его для построения микро- (фемто-) сот мобильной связи 3G, 4G и 5G (в некоторых конфигурациях).

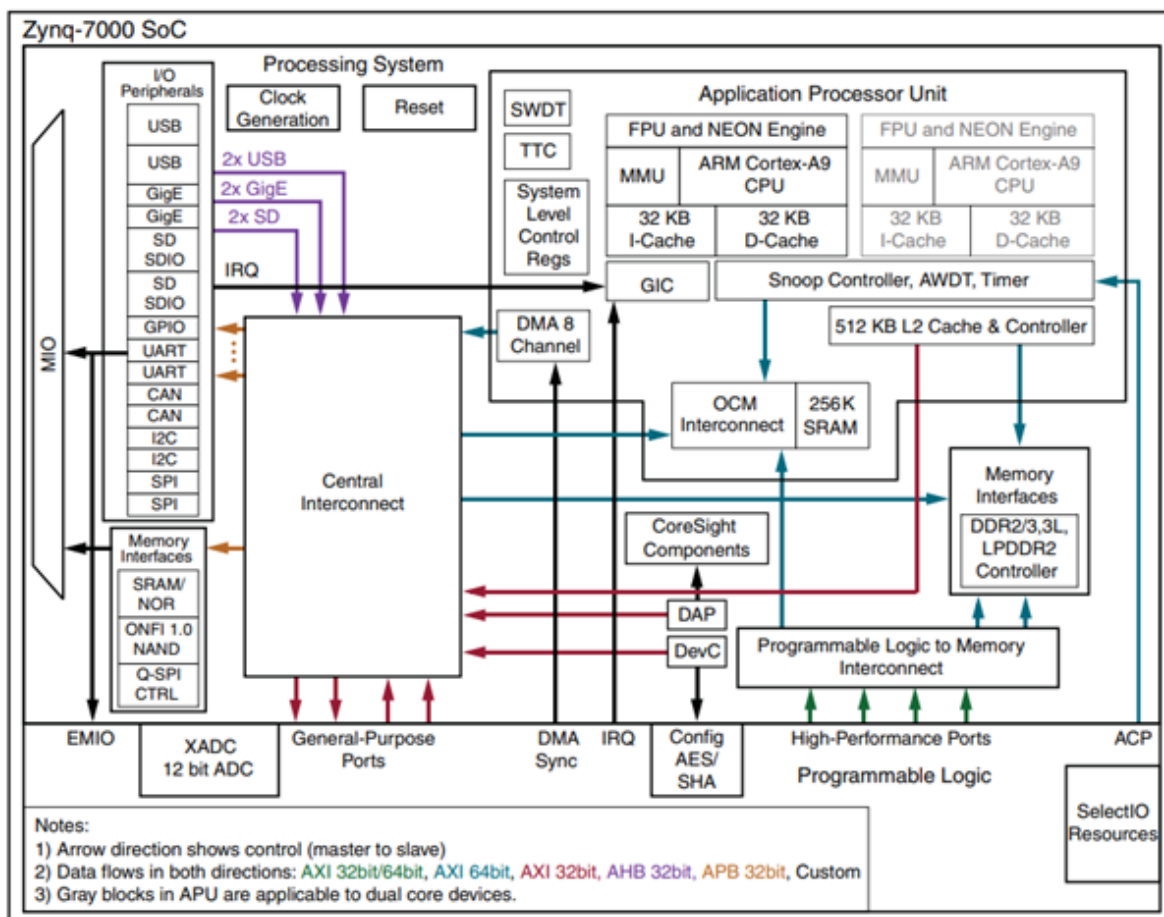


Рис. 3 — ПЛИС Xilinx Zynq

Основные характеристики

- **12-битный** ЦАП/АЦП (Цифро-аналоговый/Аналого-цифровой Преобразователь)
- Поддерживаемые несущие частоты от 90 [МГц] до **3.8** [ГГц]
- Поддерживает временной и частотный дуплексы (**TDD, FDD**)
- Ширина полосы частот: **20** [МГц]
- Шумы в приемнике: **3** [dB]
- EVM (Error Vector Magnitude): **-34** [dB]
- Tx noise: **< -157** [dBm/Hz]
- **2 Rx, 2 Tx**

Структурная схема Zynq

Каждый Zynq состоит из одного или двух ядер ARM Cortex-A9 (ARM v7), кэш L1 у каждого ядра свой, кэш L2 общий. Поддерживаемая оперативная память имеет стандарты DDR3, DDR3L, DDR2, LPDDR-2. Максимальный объем оперативной памяти равен 1 Гбайт (2 микросхемы по 4 Гбит). Максимальная тактовая частота оперативной памяти 525 МГц. Операционные системы: Standalone (bare-metal) и Petalinux. Процессорный модуль общается с внешним миром и программируемой логикой с помощью портов, объединенных в группы:

- MIO (multiplexed I/O)
- EMIO (extended multiplexed I/O)

- GP (General-Purpose Ports)
- HP (High-Performance Ports)
- ACP (Accelerator coherency port)

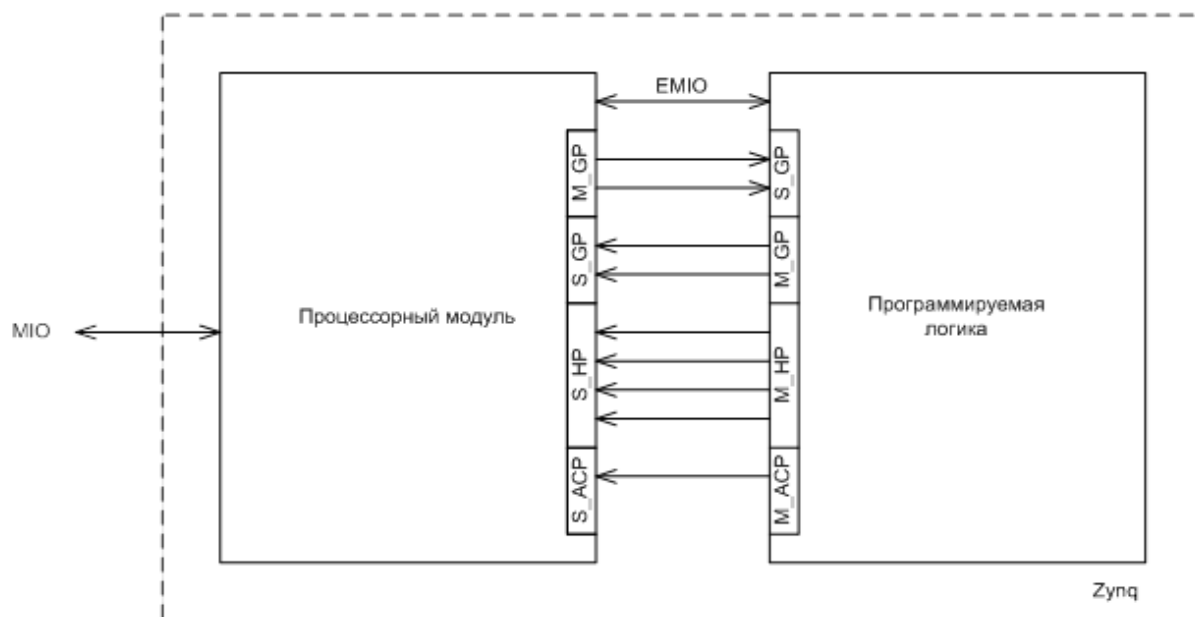


Рис. 4 — Структурная схема Zynq

Схема интерфейсов Zynq

**Буквы S и M у порта обозначают соответственно Slave и Master.*

Так как, в одном корпусе Zynq реализованы и процессорный модуль и программируемая логика, есть выводы, которые относятся к процессорному модулю и выводы, которые относятся к программируемой логике.

Порты

Порты MIO представляют собой многофункциональные порты ввода-вывода, непосредственно подключенные к выводам процессорной системы (Processing System, PS). Ключевые характеристики:

- **Количество:** 54 порта (в большинстве конфигураций)
- **Назначение:** подключение периферийных устройств PS к внешним выводам кристалла
- **Особенности:** мультиплексирование функций на одних и тех же физических выводах

MIO

Порты MIO подключены к выводам процессора. С помощью MIO могут быть подключены следующие периферийные устройства процессорного модуля:

- USB-контроллер – 2 шт
- Gigabit Ethernet контроллер – 2 шт
- SD/SDIO контроллер – 2 шт

- UART – 2 шт
- CAN – 2 шт
- I2C – 2 шт
- SPI – 2 шт
- GPIO. Все выводы можно использовать как выводы общего назначения

Так же, к MIO могут быть подключены следующие устройства памяти процессорного модуля:

- QSPI контроллер
- ONFI контроллер
- SRAM/NOR контроллер

Количество MIO портов равно 54 (за исключением некоторых микросхем в корпусе CLG225, там еще меньше). Поэтому все сразу включить не удастся. Для решения этой проблемы существует группа портов EMIO.

Выполнение

Практическая работы выполнялась в программе GNU Radio. GNU Radio - это открытая платформа для разработки программных решений в области SDR. Её ключевое преимущество — визуальное проектирование радиоэлектронных систем с помощью готовых блоков, что избавляет от необходимости программировать. Собранные проекты работают непосредственно с SDR-оборудованием, таким как Adalm-Pluto или LimeSDR.

Библиотека GNU Radio содержит обширный набор компонентов для цифровой обработки сигналов (ЦОС). Сами модули разработаны на C++ для обеспечения высокой производительности, а их соединение и управление проектом осуществляется с помощью Python. Разработку можно вести как программно, используя API, так и визуально — в графической среде GNU Radio Companion (GRC).

Создание схемы в GNU Radio

Блок Options (Параметры проекта)

Данный блок определяет основные настройки всего проекта. Наиболее важные параметры:

- **Output Language (Язык выходного кода):** определяет язык генерации исполняемой программы (Python или C++)
- **Generate Options (Опции генерации):** задаёт тип используемого интерфейса

Блок Variable (Переменная)

Служит для объявления переменных. Переменная имеет имя (ID) и значение. В проекте часто определяется:

Variable ID: `samp_rate`
Value: 2.4M

Блок QT GUI Range (Ползунок)

Создает регулируемый ползунок для динамического изменения параметров во время работы программы.

- **Default Value:** значение при запуске
- **Start/Stop:** диапазон значений
- **Step:** шаг изменения

Блок PlutoSDR Source (Источник)

Обеспечивает подключение к SDR-модулю ADALM-Pluto и управление его параметрами.
Основные параметры:

- **IO context URI:** IP-адрес устройства
- **Sample Rate:** частота дискретизации АЦП
- **LO Frequency:** частота гетеродина
- **Buffer size:** размер буфера

Блок Low Pass Filter (ФНЧ)

Фильтр нижних частот для ограничения полосы пропускания сигнала.
Параметры:

- **Decimation:** коэффициент уменьшения частоты дискретизации
- **Cutoff Freq:** частота среза
- **Sample Rate:** исходная частота дискретизации

Блок QT GUI Frequency Sink (Спектр)

Обеспечивает визуализацию спектра сигнала в реальном времени.

Блок QT GUI Time Sink (Осциллограф)

Отображает сигнал во временной области.

Блок WBFM Receive (FM-демодулятор)

Выполняет демодуляцию широкополосного FM-сигнала.

Блок Audio Sink (Выход на звуковую карту)

Направляет аудиопоток на звуковую карту компьютера.

Сборка системы

Все блоки соединяются в графическом редакторе GRC, формируя законченную схему FM-приёмника. Программа позволяет в реальном времени:

- Принимать радиопередачи
- Наблюдать спектр и форму сигнала
- Регулировать частоту настройки

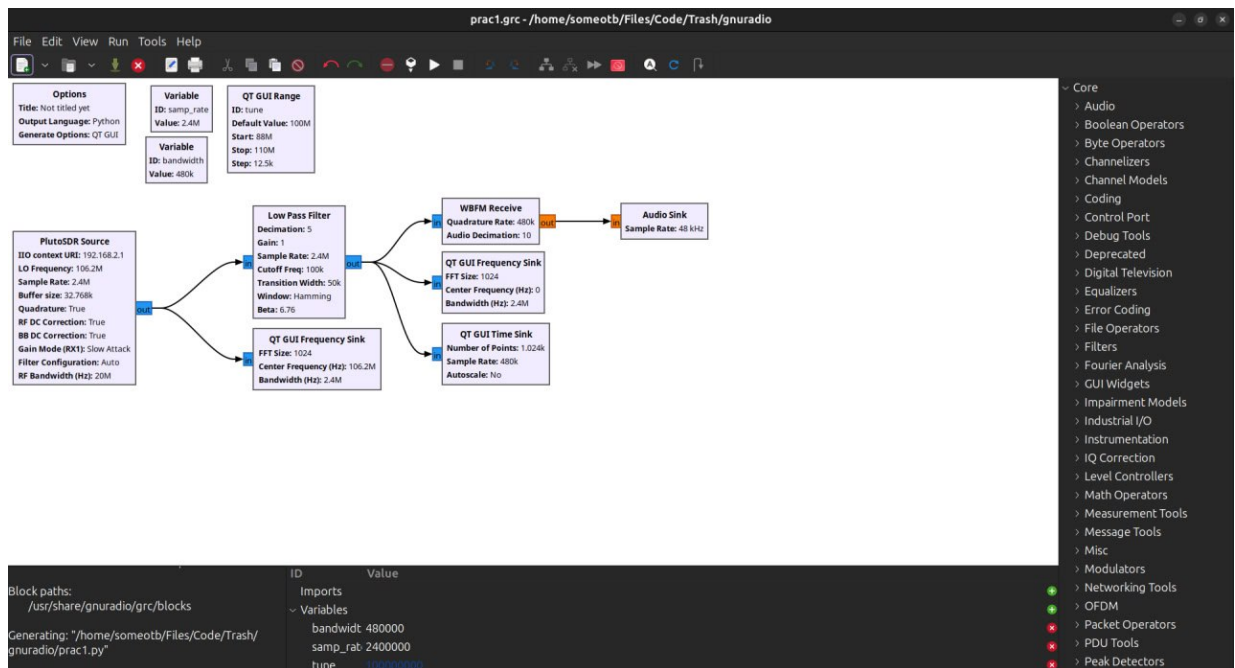


Рис. 5 — Собранная модель

Код соберется автоматически исходя из модели собранной из блоков в интерфейсе GNU Radio. И с помощью данной модели получилось поймать сигнал одной радио станции, которую можно было слушать, на других частотах звук был сильно искажен. На радио станции пока слушал, понравилась одна песня: **Девочка в платице белом - Мюзикол**.

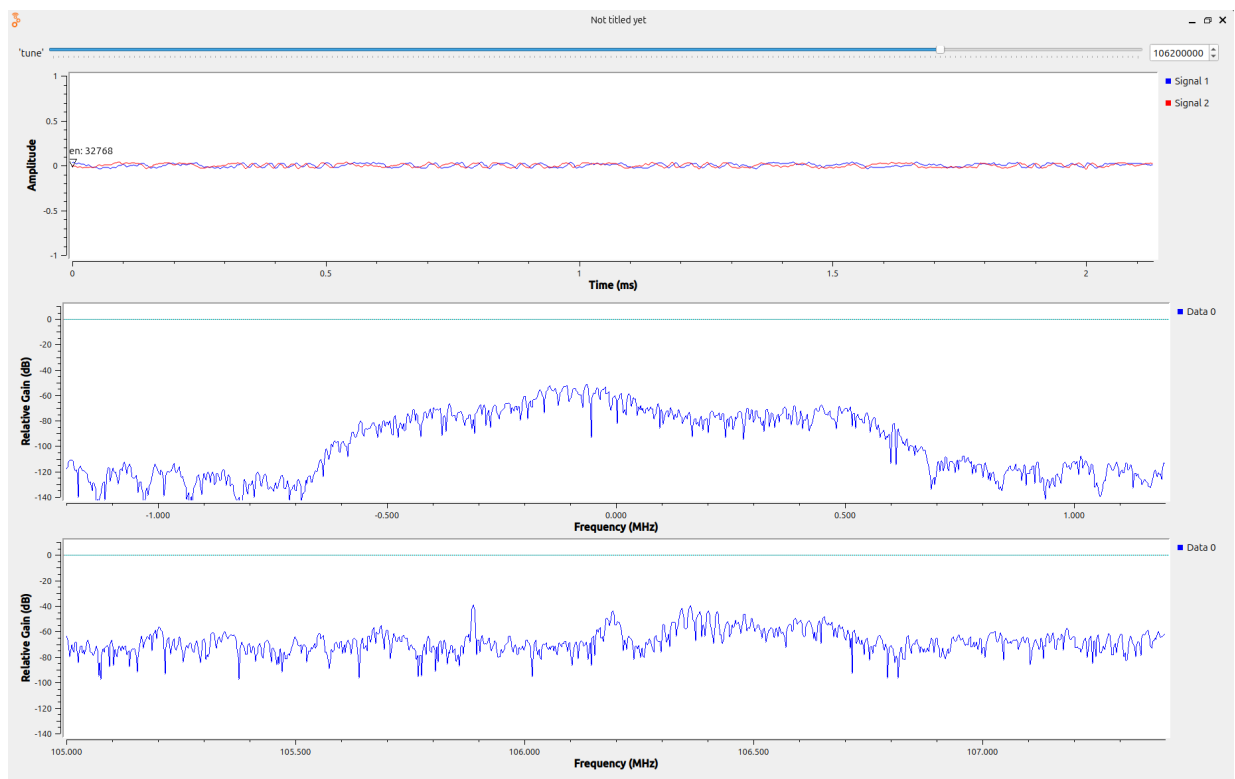


Рис. 6 — Результаты

Вывод

В результате работы были изучены принципы SDR, исследована архитектура ADALM Pluto и освоен инструмент GNU Radio. Разработана программа для приёма и воспроизведения передач в FM-диапазоне.