

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Інститут телекомунікаційних систем

Кафедра Телекомунікацій

«До захисту допущено»

Завідувач кафедри

_____ В.С. Явіся

«___» _____ 2020

р.

Дипломна робота

на здобуття ступеня бакалавра

Спеціальність 172 Телекомунікації та радіотехніка,

**на тему: «Використання безсерверних хмарних технологій для цифрової
обробки сигналів.»**

Виконав:

студент IV курсу, групи ТЗ-61

Пилипенко Д.Т. _____

Керівник:

професор, д.т.н. проф.

Ільченко М.Ю. _____

Рецензент:

д.т.н., с.н.с.

Скулиш Марія Анатоліївна _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Інститут телекомунікаційних систем

Кафедра Телекомунікацій

Рівень вищої освіти – бакалавр

Спеціальність 172 Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ

_____ В.С.Явіся

«___» _____ 2020 р

ЗАВДАННЯ

на дипломну роботу студенту

Пилипенко Данііл Тарасович

1. Тема роботи « **Використання безсерверних хмарних технологій для цифрової обробки сигналів.** », керівник роботи професор, д.т.м проф Ільченко М.Ю. , затверджені наказом по університету від «30» березня 2020 р. № 924-с.
2. Термін подання студентом роботи 04.06.2020 р.
3. Вихідні дані до роботи: 1. Спеціальна література 2. Матеріали мережі інтернет
4. Зміст роботи:
 - 1) SDR приймачі
 - 2) Аналіз широкосмугового приймача RTL2832U
 - 3) Хмарні технології.

- 4) Робота AWS з мовами програмування для візуалізації та відтворення даних.
 - 5) Створення робочої версії програми
5. Перелік ілюстративного матеріалу (зі зазначенням плакатів, презентацій тощо)
- 1) Тема та предмет дослідження. Мета роботи
 - 2) Історія виникнення SDR
 - 3) Переваги та недоліки SDR
 - 4) SRD RTL2832U
 - 5) Хмарні технології
 - 6) Класифікація хмарних технологій
 - 7) Переваги та недоліки хмарних технологій
 - 8) Сучасні мови програмування та їх комунікація з хмарними технологіями
 - 9) Практична реалізація приймача
 - 10) Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання

07.10.2019

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	SDR приймачі	07.10 — 10.02	
2.	Аналіз широкосмугового приймача RTL2832U	10.02 — 24.02	
3.	Хмарні технології	24.02 — 19.05	
4.	Обробка даних в AWS та візуалізація їх	19.05 — 22.05	
5.	Створення робочої програми	22.05 — 01.06	
6.	Висновки	01.06 — 02.06	

Студент

Пилипенко Д.Т

Керівник роботи

Ільченко М.Ю.

РЕФЕРАТ

Робота містить 60 сторінки, 32 рисунків та 0 таблиць. Було використано 5 джерел.

Метою роботи створення робочої програми з використанням широкопasmугового приймача та хмарних технологій для візуалізації даних.

В даній роботі розглядається мова програмування Python та бібліотеки для взаємодії з приймачем на основі SDR. Для обробки даних буде застосовані сервіси хмарного обчислення AWS.

Ключові слова: хмарні технології, Python, AWS, IoT, широкопasmуговий приймач.

ABSTRACT

The work contains 60 pages, 32 figures and 0 tables. 5 sources have been used.

Goal: developing a working application with using broadband receiver and cloud data visualization.

This paper discusses the Python programming language and the library for interacting with the SDR-based receiver. AWS cloud services will be used for data processing.

Keywords: cloud technologies, Python, AWS, IoT, broadband receiver.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1	11
SDR технологія. Принцип роботи	11
1.1 Історія створення SDR	11
1.2 Принцип роботи SDR	13
1.3 Переваги SDR	14
1.4 Недоліки SDR	16
1.5 Різновид SDR	17
1.6 SDR приймач RTL2832U+R820T	18
1.7 Підтипи приймачів SDR	20
1.8 Сфери використання.....	21
Висновок	25
РОЗДІЛ 2	26
Хмарні технології	26
1.1 Поняття “хмарні технології”	26
1.2 Принцип роботи та сфери застосування	28
1.3 Класифікація моделей обслуговування	32
1.4 Типи хмарних технологій	34
1.5 Характеристики хмарних технологій та обчислень. Хмарна телефонія..	34
1.6 Переваги та недоліки	38
1.6.1 Переваги	37
1.6.2 Недоліки	38
1.7 Межі керованості для різних варіантів системи	40
Висновок	42
РОЗДІЛ 3	43
Взаємодія сучасних мов програмування з хмарними сервісами	43
Основні відомості про мову програмування Python	43

AWS та Python. Аналіз вимог	44
Висновок	45
РОЗДІЛ 4	46
Практична реалізація хмарного середовища з сучасною мовою програмування на базі SDR приймач	46
Висновок	57
Загальні висновки по роботі.....	58

ПЕРЕЛІК СКОРОЧЕНЬ

AWS	Amazon Web Services
IoT	The internet of things
EDGE	<i>Enhanced Data rates for GSM Evolution</i>
LTE	Long-Term Evolution
ПЗ	Програмне забезпечення
АПЦ	1. Аналого-цифровий перетворювач
ІРА	Internet Protocol address
CPU	Central processing unit
GSM	Global System for Mobile Communications

2. ВСТУП

2020 рік — рік бездротових технологій, хмарних сервісів, різних платформ, використання IoT, “розумних” технологій та пристроїв.

Майже кожен пристрій зараз використовує мережу інтернет, кожний сучасний комп'ютер підключений до мережі інтернету, кожний смартфон має можливість використовувати EDGE, 3G, LTE.

Компанії переходять на хмарні технології, сервери айти гігантів, пишуть програми з використанням сучасних мов програмування, використовують технології для обробки великої кількості інформації, створюють нові штучні алгоритми для обробки даних, нові нейронні мережі.

Розробка і перехід на сучасні хмарні технології та використання IoT в усі сфери життя надасть величезну кількість переваг людству та значно спростить повсякденну клопіт.

Мета даної дипломної роботи полягає в створенні такого програмного забезпечення для широкопasmового приймача з використанням сучасних технологій.

РОЗДІЛ 1

SDR технологія. Принцип роботи.

1.1 Історія створення SDR

За останні 100 років радіо досить сильно змінилося і це означає що тодішній інженер скоріше за все не розбереться у сучасних пристроях. Історія SDR базується на 2 принципах:

- Демодуляція сигналу — перетворення високочастотних модульованих коливань для виділення низькочастотного сигналу. Тобто все те, що звичайний радіоприймач робить у своєму обладнанні — то у SDR все відбувається на комп'ютері за допомогою математичних алгоритмів. Це означає, що радіоприймач тільки частково потрібен (тільки та частина, що приймає сигнал).
- Передача широкосмугового сигналу з радіоприймача до комп'ютера. Ширина смуги обробки від 48 КГц (наприклад звукова система)

Термін “цифровий приймач” - був придуманий в 1970 році науковцями в лабораторії Міністерства оборони США. Лабораторія під назвою “Golden Room” у Каліфорнії створила програмний інструмент аналізу базової смуги з назвою Midas, що визначав функціонування ПО.

У 1984 та ж сама лабораторія створила термін “програмне радіо” для позначення цифрового приймача базової смуги і було опубліковано в бюлетені компанії E-Team.

1995 рік став ключовим для терміну SDR. А саме завдяки Стівену Бласту, який опублікував запит на інформацію від Bell South Wireless на першому засіданні форуму модульних багатофункціональних систем передачі інформації (MMITS) у 1996 році.

Найперша розробка була ICNIA - це об'єднана конструкція декількох радіостанцій, тобто колекція декількох одноцільових радіостанцій, що використовуються як одне обладнання.

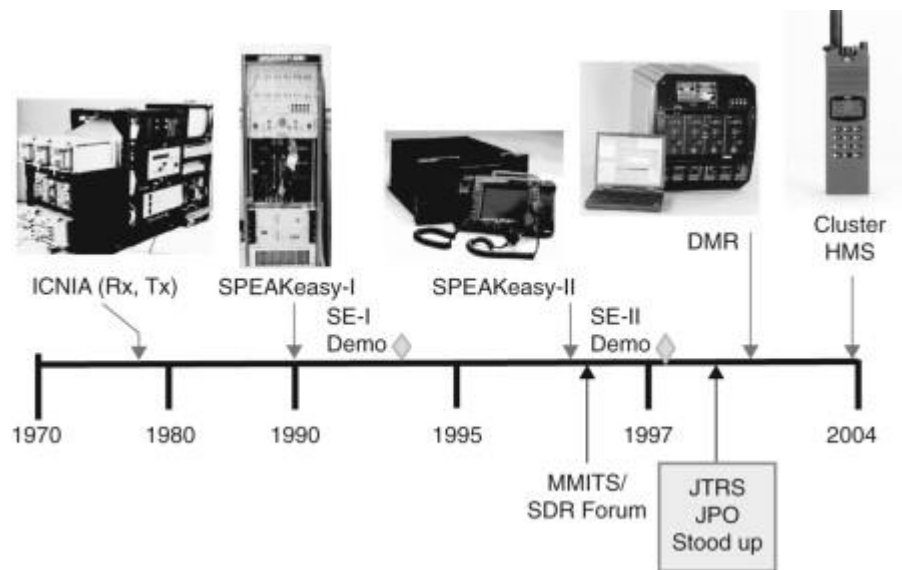


Рис 1.1 Історія розвитку SDR

Потім з'явився SDR SpeakEasy. Це розробка американських військових, з ціллю використання програмної обробки для емуляції більш ніж десяти існуючих військових радіосистем, які функціональні в діапазоні від 2 до 20 МГц. Іншою ціллю була можливість підтримки будь-яких нових схем кодування і модуляції та можливість їх покращення.

Мета такого підходу — розробити систему, яка зможе як і приймати так і передавати майже будь-які радіосигнали за допомогою ПО, що є адаптивним і гнучким.

1.2 Принцип роботи SDR

Принцип роботи базується на оцифруванні прийнятого радіосигналу та подальшій обробці його у цифровій формі. Ця технологія дозволяє отримати максимально високі характеристики приймального тракту. Велике навантаження прийнятого сигналу фактично лягає на комп'ютер. Робоча частота прийому програма автоматично визначає.

Ідеальним випадком роботи була би схема з АЦП, який би підключався

безпосередньо до антени, без аналогових виборчих приладів звичайного радіоприймача. Цифровий сигнальний процесор буде зчитувати сигнал з перетворювача і програмно представляти його в необхідному вигляді.

Схожим був би і ідеальний передавач. Цифровий сигнальний процесор генерує потік чисел які поступають на вхід цифро-аналогового перетворювача, вихід якого підключається до антени.

Ідеальна схема неможливо реалізувати із-за технічних обмежень. Основною причиною є важкість перетворення сигналу з аналогової форми в цифрову і зворотнього перетворення. Одночасно і з високою швидкістю, і з високою точністю, без появи перешкод і без допомоги електромагнітного резонансу.



Рис. 1.2 Спрощена архітектура типового SDR.

Як видно з Рис. 1.2 схема включає в себе блоки аналогово-цифрового, цифро-аналогового перетворювача, антену, ланцюг обробки цифрових

сигналів і інші допоміжні блоки. Як правило, окрім цифрового сигнального процесора, радіо з архітектурою SDR включає в себе мікроконтролер.

Одним із найважливіших вузлів є АЦП. В реальності АЦП підключається до антени напряму. Тобто перетворює безперервний в часі сигнал в дискретну двійникову форму. Очевидно, що характеристика АЦП буде в багатьох випадках визначити і параметри приладу в цілому. Тому потрібно звернути увагу на такі відносини як сигнал-шум, число біт за виборку та динамічний діапазон при відмові паразитних складових.

1.3 Переваги SDR

- SDR показує панораму радіоефіру як вона є.

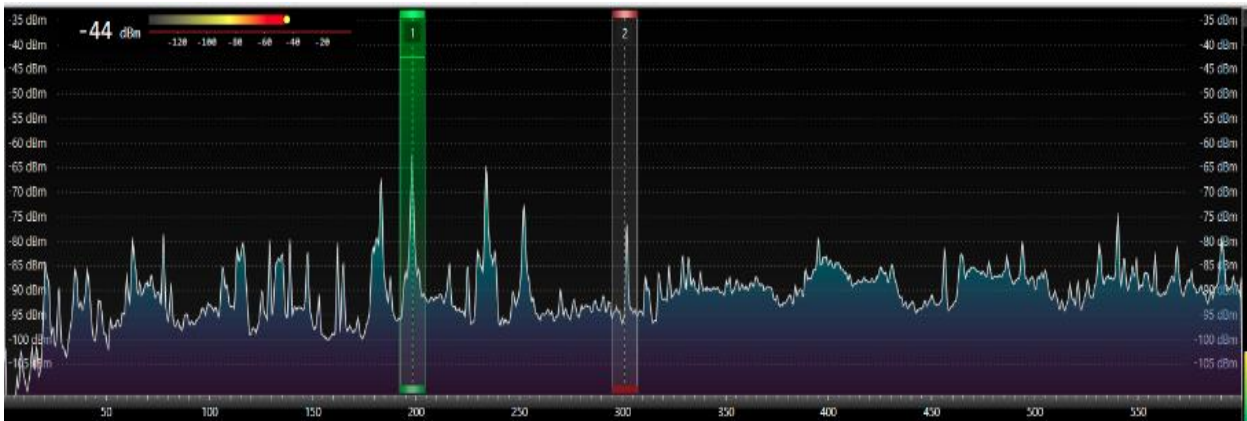


Рис 1.3 панорама радіоефіру в програмі SDR

Це доволі зручно, адже на екрані можна одразу побачити які станції і де працюють, зручно шукати нові сигнали, можна побачити різні завади, їх параметри, потужність сигналу, діапазон і тд.

І все це відбувається прямо у браузері. Не потрібно ніяких особливих підключень, потужних антен

Для прикладу буде використано голландський SDR.

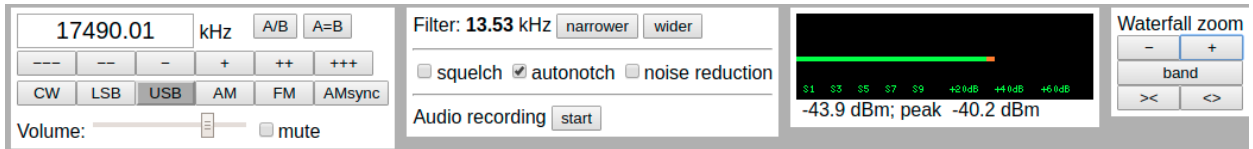


Рис 1.4 Налаштування голландського SDR на стацію CNN

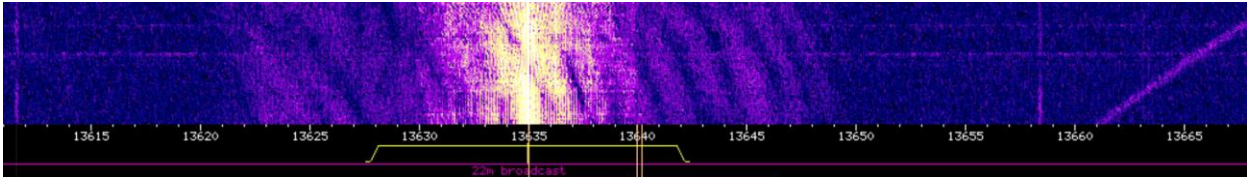


Рис 1.5 Графічне відображення потужності сигналу

Це CNN China Radio. Швидко знаходиться завдяки можливості простого налаштування, а також графічного інтерфейсу.

- Можливість зробити заміри
SDR доволі точний прилад для замірів. Починаючи від відображення різних станцій до оцінки завад, якості сигналу. Все недоліки свого або чужого сигналу добре видно на спектрі.
- Можливість робити широкосмугові записи та відтворення
У SDR все данні пишуться у цифрі - а це означає, що всю смугу можна записати суцільно, щоб прослухати і проаналізувати. Також є можливість записувати декілька радіостанцій в одному записі.
- Можливість віддаленої праці
Деякі SDR приймачі та трансивери мають можливість віддаленої праці — так як приймач доступний по ІРА. Тобто немає принципової різниці між тим, що стоїть десь поруч або в 1000 км від Вас. Виходячи з того, що завади у міст все більші і більші — то це може стати доволі актуальним.

1.4 Недоліки SDR

- Ціна

Проаналізувавши ринок SDR, можна впевнено сказати, що хороший SDR коштує дорого. Але можна знайти альтернативу, яка б підходила для більшості любительських цілей.

- Тільки стаціонарна робота

Основа проблема більшості SDR — це стаціонарний настільний прибор, який з собою брати дуже не зручно. А оскільки в містах є доволі сильний рівень завад, то портативності не вистачає.

Хоча з'являються альтернативи які можна було би компактно та легко переміщувати та взяти з собою — але ці прибори ще на ранніх стадіях і коштують чималих грошей.

- Енергоспоживання та потреби ПЗ

SDR мають високі вимоги до відеокарт та CPU. Радіоприймач також може працювати певну кількість часу від батарейки.

- Складні алгоритми цифрової обробки

Велика кількість функцій говорить про те, що за всім цим знаходяться складні алгоритми обробки сигналів. Звичайно, спільнота користувачів вже написала велику кількість різного ПЗ для SDR і дописувати нічого не прийдеться. Але при бажанні щось додати доведеться витратити немалу кількість часу.

1.7 Різновид SDR

- Старі моделі з використанням звукових карт — оцифровка сигналу в них відбувається безпосередньо в комп'ютері. Сигнал передається на лінійний вхід по кабелю. На даний момент вони вже вийшли з виробництва, але деякі можна все таки знайти десь на радіо-ринку
- SDR з вбудованим АЦП які передають сигнал у комп'ютер. Формат передачі — цифровий. Загалом це сучасні прилади середнього діапазону цін. Вони базуються на принципі гетеродинного прийому. Але після переносу частот замість низько частотного блоку стоїть АЦП. Такі приймачі працюють на смузі частот 2-10МГц.
- DDC SDR — Найсучасніша технологія на сьогоднішній день. Основна концепція є у тому, що гетеродин не потрібен. Це йде за рахунок надшвидкого АЦП з частотою оцифровки порядку 100 мільйонів семплів за секунду, що оцифрує вхідний сигнал з ефіру, що надає можливість (згідно теоремі Котельникова) мати прийом до частоти, рівній половині частоти дискретизації. 30-50МГц — верхня межа частоти цих приймачів.

Також доволі важливий параметр є тип підключення. Адаже майже всі SDR підключаються через USB, але є виключення — LAN-порт. Це зручно для організації якихось віддалених прийомів або передач. Приймач можна розмістити десь за межами міста і використовувати його вдома. Одним із виробників таких приймачів є KiwiSDR. Данні приймачі мають підключення до веб-інтерфейсу — звідки і відбувається управління.

1.6 SDR приймач RTL2832U+R820T.



Рис 1.6 Приймач RTL2832U+R820T у базовій комплектації

Даний приймач використовується у дипломній роботі, так як він має досить широкий спектр можливостей.

Сам приймач доволі дешевий та купити його можна майже на будь-якому сайті радіо-користувачів. Цей приймач має можливість роботи у SDR. Все що потрібно зробити для цього — це замінити оригінальний драйвер спеціальним ПЗ. Такий приймач може приймати всі радіостанції, які працюють у діапазонах частот від 24 МГц до 2,2 ГГц, у тому числі і СІ-БІ діапазон радіоприхильників діапазону 70 сантиметрів, 2 метри, 10 метрів. Також авіа діапазон, різного виду рацій, спектри GSM та інших з CW, AM, ЧМ, SSB, NFM, WFM модуляціями.

Для роботи з цим приймачем не потрібно окремої звукової карти. Достатньо підключити його через USB порт ноутбуку або комп'ютеру або планшету. Встановити драйвер і запустити будь яку програму яка вміє працювати з SDR.

Технічні характеристики цього приймача:

- ➔ 24-1750 МГц — діапазон частот
- ➔ АМ, ЧМ, NFM, CW, LSB і тд. — модуляції
- ➔ 0.22 мКв (438МГц в режимі NFM) — чутливість
- ➔ 50 ом — вхідний опір приймача
- ➔ 250 кГц до 3 МГц — смуга огляду
- ➔ 50 дб — динамічний діапазон
- ➔ 340 мсек — затримка приймального сигналу
- ➔ Діапазонні фільтри тільки зовнішнім
- ➔ USB 2.0 — інтерфейси
- ➔ 8 біт — Розрядність АЦП

Операційна система — будь-яка.

Далі буде розглянуто технічна реалізація обв'язки R820T

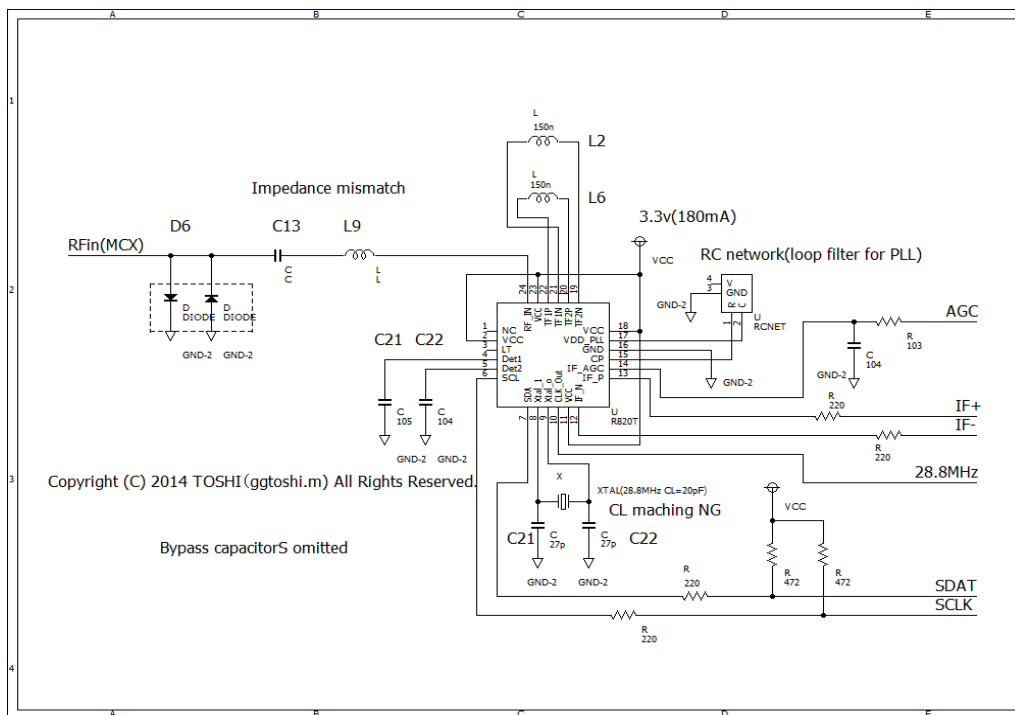
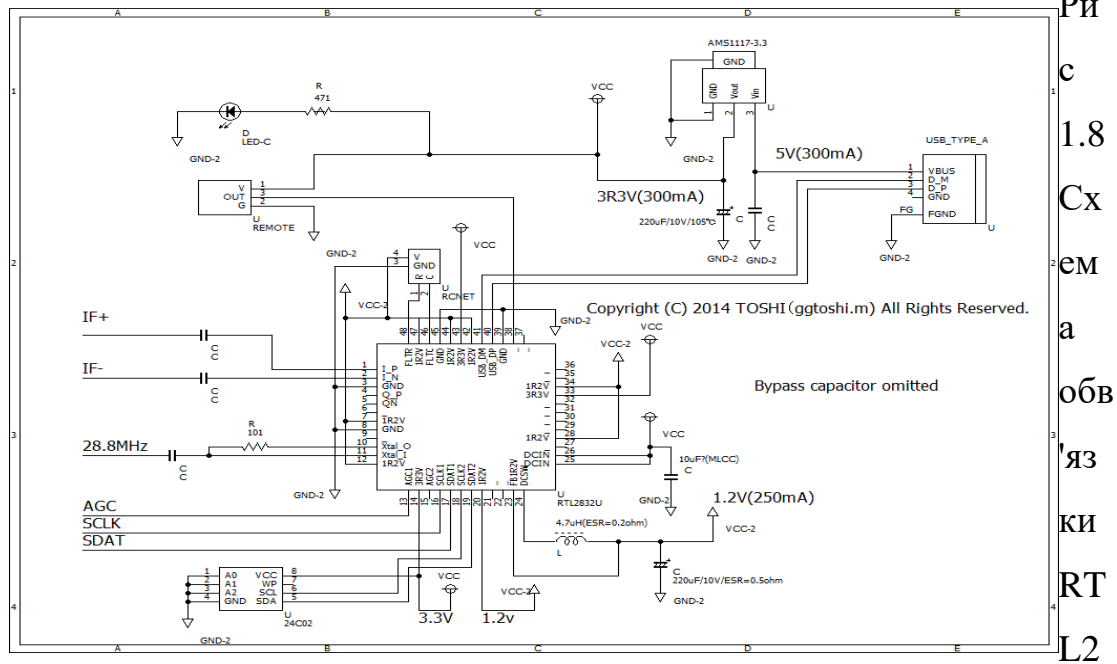


Рис 1.7 Схема обв'язки R820T

Та обв'язка RTL2832U



832U

Вхідний ланцюг включає в себе захисний діод BAV99(до 1 pF), конденсатор C13 (R820T — 330pF), індуктивність L9(8.2nH) і вхідна ємність R820T(3pF). Конденсатор C13 обрізає частоту до 10 МГц.

1.7 Підтипи приймачів SDR

- RTL-SDR

Ерік Фрай виявив, що деякі звичайні недорогі USB-ключі DVB-T з контролером Realtek RTL2832U і тюнером, наприклад, Elonics E4000 або Rafael Micro R820T можна використовувати як широкосмуговий (3 МГц) приймач SDR. Недавні експерименти довели можливість цієї установки аналізувати персеїди зливи за допомогою радіолокаційних сигналів могил. Цей проект зараз підтримується в Osmocom.



Рис 1.9 Приймач RLT2832U без зовнішнього корпусу

- USRP

З недавніх пір радіо GNU, що використовує передусім периферійне програмне забезпечення Universal Software (USRP), використовує інтерфейс USB 2.0, FPGA та високошвидкісний набір аналого-цифрових та цифро-аналогових перетворювачів у поєднанні з вірно-налаштованим безкоштовним програмним забезпеченням . Її смуга дискретизації та синтезу (30-120 МГц) в тисячу разів більша від звукових карт ПК, що дозволяє широкосмуговій роботі.

- HPSDR

Проект HPSDR (високоєфективне програмне забезпечення, визначений програмним забезпеченням) використовує 16-бітний аналого-цифровий перетворювач MSPS 135, який забезпечує продуктивність в діапазоні від 0 до 55 МГц, порівнянну з частотою звичайного аналогового високочастотного радіочастотного радіо. Приймач також буде працювати в діапазоні УКХ та УВЧ, використовуючи або зображення змішувача, або відповіді псевдоніму. Інтерфейс до ПК забезпечується інтерфейсом USB 2.0, хоча Ethernet також може бути використаний. Проект є модульним і містить опорну площину, на яку вставляються інші дошки. Це дозволяє експериментувати з новими методами та пристроями без необхідності заміни всього набору дощок.

Збуджувач забезпечує 1/2 Вт ВЧ в тому ж діапазоні або в діапазоні УКХ та УВЧ, використовуючи вихідні зображення або псевдоніми.

- WebSDR

WebSDR - це проект, ініційований Пітером-Терком де Буром, що забезпечує доступ через браузер до декількох приймачів SDR по всьому світу, що охоплюють повний спектр короткохвильових хвиль. Нещодавно він проаналізував сигнали передавача чирпу, використовуючи зв'язану систему приймачів.

1.8 Сфери використання

- Військове застосування:

Об'єднана тактична радіосистема (JTRS) - це програма американських військових для створення радіостанцій, які забезпечують гнучку та сумісну взаємодію. Приклади радіотерміналів, які потребують підтримки, включають радіостанції з ручним, автомобільним, повітряним транспортом та з демонтажем, а також базові станції (стаціонарні та морські).

Ця мета досягається завдяки використанню систем SDR на основі міжнародно схваленої відкритої архітектури програмного забезпечення (SCA). Цей стандарт використовує CORBA на POSIX операційних системах для координації різних програмних модулів.

Програма пропонує новий гнучкий підхід для задоволення різноманітних потреб в солдатській комунікації через програмно програмовані

радіотехнології. Вся функціональність та розширюваність базується на SCA.

Гнучкість SDR призводить до дорогої складності, неможливості оптимізації, уповільнення здатності застосовувати новітні технології та рідко потребує тактичного користувача (оскільки всі користувачі повинні вибрати та залишитися з одним і тим же радіо, щоб спілкуватися).

SCA, незважаючи на своє військове походження, перебуває під оцінкою комерційних виробників радіо здатності на предмет застосування у своїх сферах. Однак прийняття загальнообов'язкових рамок SDR поза військовими, розвідувальними, експериментальними та аматорськими напрямками суттєво гальмує той факт, що цивільні користувачі можуть легше розраховуватися з фіксованою архітектурою, оптимізованою для конкретної функції та як такою більш економічною у застосуваннях на масовому ринку. Тим не менш, властива гнучкістю радіо, визначена програмним забезпеченням, може принести істотні переваги в довгостроковій перспективі, як тільки постійні витрати на його впровадження знизяться достатньо, щоб перевершити витрати на повторне проектування цільових систем, побудованих за призначенням. Це потім пояснює зростаючий комерційний інтерес до технології.

Інфраструктурне програмне забезпечення на базі SCA та інструменти швидкого розвитку для освіти та досліджень SDR забезпечуються проектом Впровадження відкритих джерел SCA - Embedded (OSSIE). Форум бездротових інновацій фінансував проект впровадження довідкової програми SCA, впровадження з відкритим кодом для специфікації SCA. (SCARI) можна завантажити безкоштовно.

- В аматорському:

Типовий радіоаматорський програмний радіо використовує приймач прямого перетворення. На відміну від приймачів прямого перетворення більш віддаленого минулого, використовувани змішувальні технології базуються на детекторі квадратурного відбору проб і збуднику квадратурного відбору проб.

Продуктивність приймача цієї лінії SDR безпосередньо пов'язана з динамічним діапазоном використовуваних аналого-цифрових перетворювачів (АЦП). Радіочастотні сигнали перетворюються вниз в смугу звукових частот, яка відбирається за допомогою високопродуктивної АЦП звукової частоти. SDR першого покоління використовували звукову карту ПК 44 кГц, щоб забезпечити функціональність АЦП. Нові радіостанції, визначені програмним забезпеченням, використовують вбудовані високоефективні АЦП, які забезпечують більш високий динамічний діапазон і більш стійкі до шуму та інтерференції.

Швидкий ПК виконує операції цифрової обробки сигналів (DSP), використовуючи програмне забезпечення, специфічне для радіоапаратури. Декілька зусиль радіо програмного забезпечення використовують бібліотеку SDR з відкритим кодом DttSP.

Програмне забезпечення SDR виконує всю демодуляцію, фільтрацію (і радіочастоту, і звукову частоту), і посилення сигналу (вирівнювання та бінауральне подання). Використання включає всі поширені аматорські модуляції: код морзе, одностороння модуляція, частотна модуляція,

амплітудна модуляція, а також різноманітні цифрові режими, такі як радіотелетип, телевізор із повільним скануванням та пакетне радіо. [18] Аматори також експериментують з новими методами модуляції: наприклад, проєкт DREAM з відкритим кодом розшифровує техніку COFDM, яку використовує Digital Radio Mondiale.

Існує широкий спектр апаратних рішень для радіоаматорів та домашнього користування. Існують приймачі-приймачі професійного рівня, наприклад Zeus ZS-1 або Flex Radio, рішення для домашнього приготування, наприклад, Приймач PicAStar, комплект SDR SoftRock і рішення для початківців або професійних приймачів, наприклад, FiFi SDR для короткохвильового або когерентного багатоканального SDR приймача Quadrus для короткохвильового або УКХ / УВЧ в прямому цифровому режимі роботи.

Висновок

SDR приймачі мають дуже великий набір функціональності. І казати що SDR — це прийом тільки на персональний комп'ютер буде дуже звуженою ідеєю.

В ході дослідження проаналізовано як плюси так і мінуси цих приймачів.

Ця технологія відкриває широкий діапазон частот. Що дозволяє працювати з нею як і новачкам так, і радіо шанувальника, і науковцям.

Цінова політика доволі велика. Можна знайти як і дешеві приймачі з великими можливостями так і професійні приймачі для досліджень.

Приймачі мають достатньо ПЗ яке розвивається, має свої графічні інтерфейси, що дозволяє значно спростити роботу. Можна як і записувати звук, так і відтворювати — тобто значне спрощення. Потужний функціонал в ПЗ також значно спрощує роботу з цими приймачами та доповнює функціонал.

Також виходячи з аналізу приймача RTL2832+R820T — можна впевнено сказати, що він підходить для практичної реалізації дипломної роботи, має дуже великий спектр можливостей, особливо за свою ціну.

РОЗДІЛ 2. Хмарні технології.

2.1. Поняття “хмарні технології”

Останнім часом все більше популярність набирає термін “хмарні технології”, виникає все більше і більше статей, проводяться конференції, виникають нові рішення на ринках технологій.

З технічної точки зору хмарні технології — це спосіб організації програмних та фізичних коштів, а також набір інструментів за допомогою яких користувач отримує певну кількість обчислювальної потужності для виконання поставленим перед ним завданням.

Взагалі термін “хмарні технології” мають загальноприйняте графічне представлення.

Основа піраміди «інфраструктура» – це набір фізичних пристроїв (сервери, тверді диски), над нею надбудовується «платформа» – набір послуг і верхівка – програмне забезпечення, що доступне за запитом користувачів.



Рис 2.1 Графічне представлення хмарних технологій

Хмарні обчислення — це технологія, який дозволяє користувачам отримувати комп'ютерні ресурси і потужності у вигляді інтернет-сервісу.

Тобто для обробки і обчислень користувач не використовує потужності свого комп'ютеру, а сторонні. Для прикладу буде наведено такі популярні хмарні технології:

- Поштові клієнти — gmail, hotmail, mail, outlook.
- Збереження даних — Dropbox, OneDrive, Google Drive, SharePoint, iCloud.
- Онлайн-магазини додатків: App Store, Google Play.
- Ігрові сервіси — Xbox Live.
- Сервіси для сумісної роботи — Jira, Trello
- Хмарний хостинг — будь-який, що дозволяє розміщувати сайт на платформі сервісу.

Перераховувати можна довго, адже ця технологія набирає все більше можливостей, потужніший функціонал, простіше використання.

Можна переглянути малюнок для складання більш абстрактного уявлення цієї технології. На ній зображені великі компанії які по суті мають сервери для користувачів і надають всім послуги.

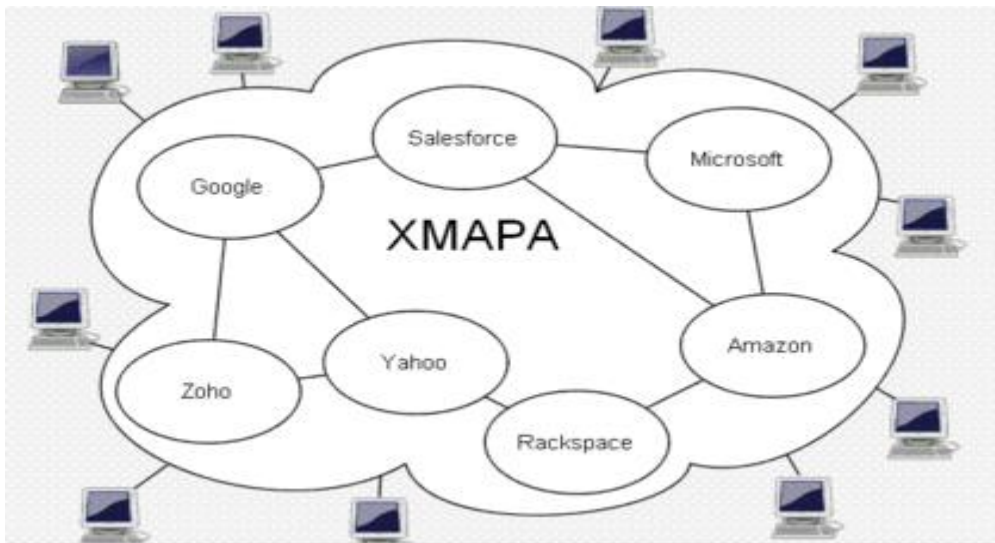


Рис 2.2 Реалізація хмарних технологій

2.2 Принцип роботи та сфери застосування

Хоча термін трактує дещо абстрактне, але за ним стоїть своя архітектура, реальне залізо та ПЗ.

Все об'єднано за допомогою програмного рішення в якому є певний інтерфейс для користувачів для управління. Все це стоїть на основі мережевого обладнання та серверів.

Для більш детального уявлення можна уявити спеціальне приміщення де стоять шафи. Це приміщення добре вентилується для підтримання певної температури.

Ці приміщення добре ізольовані для підтримання постійної напруги, температури. Мають декілька джерел напруги, запасні генератора для забезпечення максимальної стійкості. Також обладнанні спеціальними



датчиками для підтримання безпеки у разі надзвичайних ситуацій.

Рис 2.3 Серверна

В шафах знаходяться потужні комп'ютери які мають велику кількість пам'яті, дисків для зберігання даних. Тобто потужне обладнання для обчислення та зберігання даних. Щоб користувачі отримали доступ для цього обладнання приміщення знаходиться у мережі з купою мережевих комутаторів, маршрутизаторів.

Все це обладнання працює само по собі за допомогою різної підтримки. Кожна одиниця має свої задачі. Але в той же момент, хмарні системи — це все єдине ціле, яке працює ідеально, як налагоджений

механізм.

Кожен користувач використовує вже готовий продукт — використання сервісами на сайті. Наприклад — перевірити пошту, встановити програму на телефон.

Хмарні технології все більше проникають в усі галузі економіки, бізнес-процеси і сфери людського життя.

Перш за все в хмари для зберігання переміщуються величезні масиви даних, в тому числі інформація користувачів різних сервісів, персональні дані абонентів телеком-операторів, дані по платежах тримачів банківських карт, дані по покупкам в інтернет-магазинах, інформація з Інтернету речей (IoT) і геолокаційні сервісів, а також медіа-контент (відео-, аудіо- та фотоматеріали). Власне, систематизація, обробка і аналіз цих даних сьогодні теж відбувається в хмарах (Big Data), в тому числі для вирішення ситуаційних бізнес-завдань.

В хмарах сьогодні розгортають безліч інтерактивних сервісів для користувачів, наприклад, онлайн-перекладачі, онлайн-ігри, миттєві платежі, відеозв'язок, навігаційні сервіси, онлайн-бібліотеки, музичні агрегатори, держпослуги, а також звичні програми для роботи з текстами, таблицями, фотографіями та т.д.

Що стосується бізнесу, то зараз ми знаходимося на етапі зміни парадигми: від побудови ІТ всередині компанії до споживання всіляких сервісів для задоволення практично будь-якої потреби компанії з хмарної середовища. Уже сьогодні в корпоративних хмарах розміщують call-центри, поштові сервіси, сайти, форми замовлень, а також програмні продукти для

електронного документообігу, автоматизації виробництва, логістики та торгівлі, бухгалтерські та податкові системи компанії, і навіть самі робочі столи співробітників для віддаленої роботи (VDI).

Якщо говорити про окремі галузі, то в більшій мірі в хмари перемістилися розробка і тестування ПО, електронні торгові майданчики, телеком, логістика, дизайн і проектування, а також ціле бізнес-напрямок інтернет-стартапів, наукові дослідження (обробка метаданих) і дистанційна освіта. У самий найближчий час можна буде спостерігати значний перенос бізнес-критичних додатків в хмари компаній банківської та фінансової сфери, організацій охорони здоров'я, паливно-енергетичного комплексу та державних структур.

Самі віртуалізовані середовища будуть все більше зв'язуватися в глобальні гео-розподільні хмарні інфраструктури. Таким чином, експортні і мережеві підприємства можуть замовляти ІТ-сервіси по всьому світу з однієї локальної точки, маючи при цьому якісний рівень технічної підтримки від сервіс-провайдера.

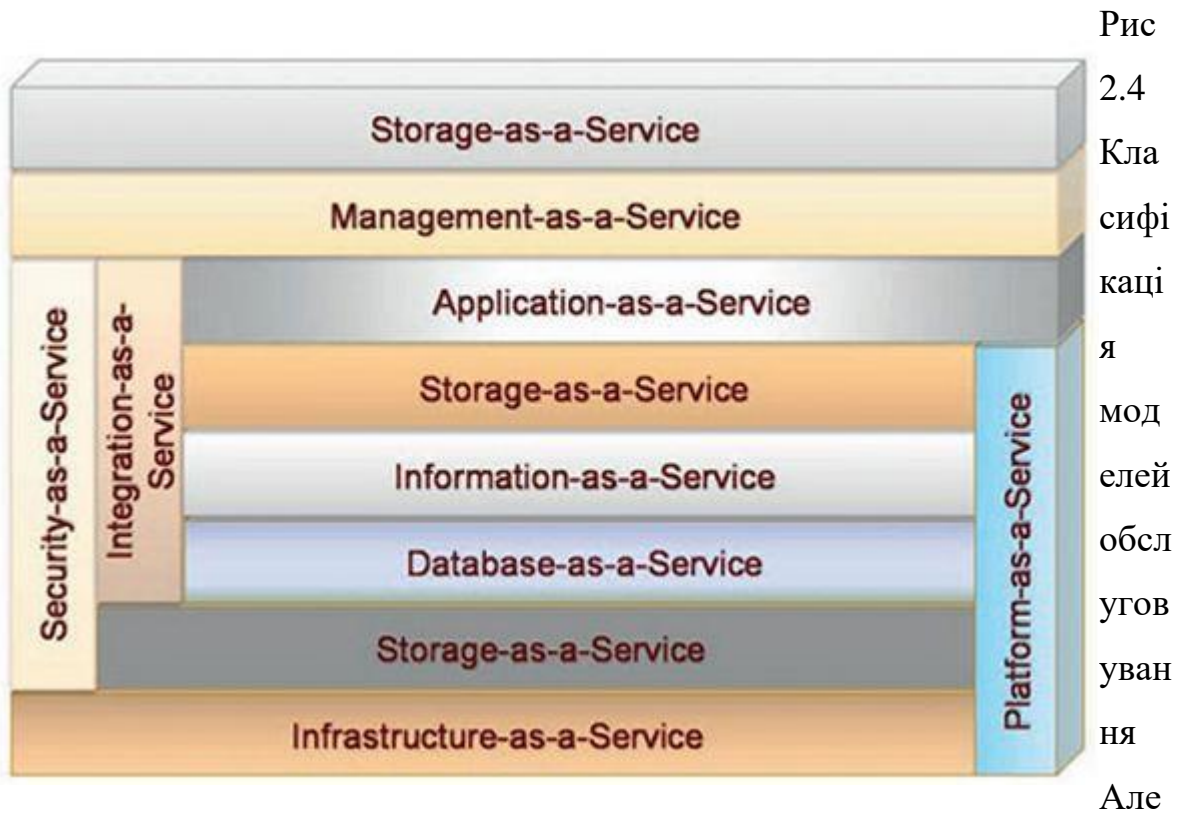
2.3 Класифікація моделей обслуговування

aaS — так прийнято називати все, що стосується хмарних технологій. Розшифровується це просто — як сервіс або у вигляді сервісу.

Хмарні технології нині включаються в себе такі технології:

- SaaS — програмне забезпечення як послуга.
- PaaS — платформа як послуга.
- IaaS — інфраструктура як послуга
- DaaS — дані як послуга

- WaaS — робоче місце як послуга
 - AaaS — усе як послуга
- Насправді їх більше.



достатньо для розуміння 3 основні з них:

SaaS — клієнт використовує ПЗ провайдера, яке працює у хмарному середовищі. Користувач не купує додаток SaaS, а орендує його — тобто платить за його використання певну суму в місяць.

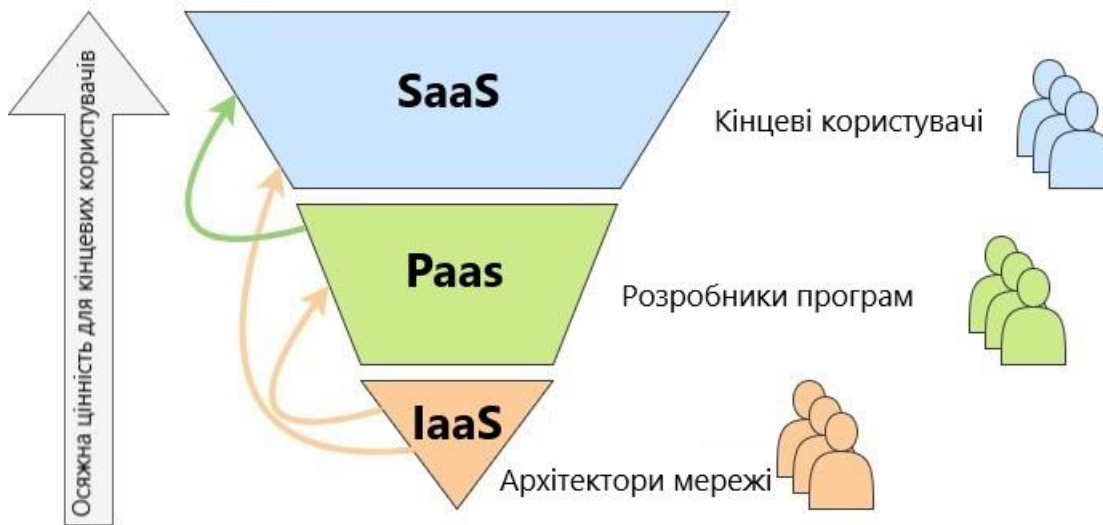
Також SaaS можна трактувати як Storage as a Service — тобто зберігання як сервіс. Хмарні ресурси можна використовувати для зберігання даних. Наприклад Dropbox, OneDrive, Google Drive.

PaaS — модель в якій користувач отримує доступ до використання інформаційно-технологічних платформ таких як операційна система, зв'язного ПЗ, управління базами даних. Провайдер керує над зберіганням системи, серверами. Також провайдер визначає набір доступів в платформі. В той же час користувачу надається можливість використовувати ці

платформи, створювати їх віртуальні екземпляри, видаляти, встановлювати, розробляти, тестувати та експлуатувати на них ПЗ, при цьому змінюючи параметри у динамічному режимі за допомогою споживаних обчислювальних ресурсів.

IaaS – модель яка надає можливість використання хмарної інфраструктури для самостійного управління ресурсами обробки, зберігання, мережами та іншими фундаментальними обчислювальними ресурсами, наприклад, споживач може встановлювати і запускати довільне програмне забезпечення, яке може включати в себе операційні системи, платформи і прикладне програмне забезпечення. Споживач може контролювати операційні системи, віртуальні системи зберігання даних і встановлені програми, а також володіти обмеженим контролем за набором доступних мережевих сервісів (наприклад, фаєрволем, DNS). Контроль і управління основним фізичною і віртуальною інфраструктурою хмари, в тому числі мережі, серверів, типів використовуваних операційних систем, систем зберігання здійснюється хмарним провайдером.

Рис 2.5 Основний принцип роботи при взаємодії з користувачами



Основні моделі для роботи з хмарними технологіями

2.4 Типи хмарних технологій



Рис 2.6
Моделі
хмарних
технологій
Публічні(за
гальнодосу

пні) — це ІТ-інфраструктура, яка використовується водночас багатьма компаніями і сервісами. Доступу у користувачів не вистачає для управління і обслуговуванням хмарного середовища, а уся відповідальність по цим питанням покладається на власника ресурсу. Прикладами виступають Amazon , Google Apps і так далі.

Приватні — це безпечні інфраструктури, яка контрольована і експлуатована в інтересах однієї та єдиної організації. Організація може

управляти цим хмарним середовищем самостійно або поручити задачу комусь з підрядчиків. Інфраструктура може розміщатися, або в приміщеннях замовника, або у зовнішнього оператора.

Гібридні — це інфраструктура, яка використовує якості публічного і приватного хмарного середовища базуючись на задачі. Часто такий тип використовується, коли організація має сезонні активності.

Громадське(корпоративна) — це інфраструктура, підготовлена для використання конкретним спільнотою споживачів, що мають спільні проблеми (наприклад, місії). Тобто ця інфраструктура може знаходитись в якійсь спільній власності або управлінні.

2.5 Характеристики хмарних технологій та обчислень. Хмарна телефонія

Доступність через інтернет мережу — основний спосіб з'єднання — це мережа інтернет. Весь доступ до механізмів, які забезпечують використання різновидностей клієнтських платформ.

Індивідуальне самообслуговування — споживач самостійно використовує обчислювальні можливості такі як час на сервері, мережеві сховища даних. Тобто майже в автоматичному режимі, без будь-якої взаємодії з командою розробників або персоналом зі сторони постачальника послуг.

Швидка адаптація — обчислювальні можливості дуже швидко можуть резервуватися(дуже часто буває у автоматичному режимі), щоб була змога оперативно робити масштабування для будь-яких завдань замовника, гнучко розпоряджати ресурси, швидко звільняти ті ж самі ресурси. З точки зору замовника(клієнта) - це можливість придбання в будь-який момент часу будь-де та в будь-якій кількості всі ці споживання.

Вимірюваність та масштабованість хмарної послуги — автоматизація

над контролем та використанням ресурсів через виміри деяких параметрів. В залежності від типу послуги ці параметри варіюються. Всі події відстежуються, чітко та пильно контролюються, а потім заносяться до звітів. За допомогою цього і постачальник, і клієнт отримують звітovanу інформацію про обсяг спожитих послуг.

Зі сторони постачальника, об'єднання ресурсів і доволі періодичному характеру використання зі сторони споживача, хмарні технології, а саме обчислення, дозволяють економити на масштабах, за допомогою використання меншого апаратного обладнання, ресурсів, ніж би кожен користувач отримував виділені апаратні потужності. Та автоматизація всіх процедур допомагає модифікувати виділення ресурсів які сильно зменшують витрати на абонентське обслуговування.

Зі сторони клієнта, всі ці характеристики надають можливість отримувати послуги з дуже високим показником поступності і малими показниками непрацездатності. А отже — забезпечити швидке масштабування всіх систем обчислення завдяки еластичності. Як висновок - Необхідність в модернізації власного обладнання та інфраструктури відпадає

За допомогою широкої доступності послуг забезпечується універсальність, а головне і зручність доступу, а також підтримкою різних класів пристроїв(термінальних), таких як ПК, мобільний телефон.

Хмарна телефонія

На відміну від аналогової телефонії, зв'язок між абонентами при такому способі організації встановлюється через інтернет. АТС, звичайна або віртуальна, знаходиться у провайдера. При цьому користувач має доступ до управління АТС (за умови замовлення послуги).

Саме на хмарних технологіях зараз побудовано більшість багатоканальних телефонних ліній з можливістю прийому і переадресації безлічі дзвінків одночасно. Дзвінки можна здійснювати не тільки по стаціонарному або мобільному телефону, але і за допомогою планшета, ноутбука або звичайного комп'ютера. При підключенні хмарної АТС, як правило, стають доступні функції голосового меню і автоматичної переадресації.

Хмарна телефонія дозволяє значно скоротити витрати на штат call-центру, підвищуючи при цьому ефективність роботи з клієнтами. Можна обійтися навіть без покупки телефонних апаратів, досить встановити на робочі комп'ютер, телефон або планшет спеціальне програмне забезпечення. Фірми, що мають філії в різних містах або працюють із зарубіжними партнерами, отримують швидку, комфортну і недорогу зв'язок. Для того щоб почати користуватися послугою хмарної АТС, досить просто зареєструватися на сайті провайдера і вибрати необхідні функції в особистому кабінеті.

Зараз послуги хмарної телефонії пропонують практично всі великі мобільні провайдери, причому компанія може зберегти свої старі номери, який би провайдер їх ні обслуговував, досить скористатися функцією переадресації дзвінків

2.6 Переваги та недоліки

2.6.1 Переваги

Спочатку буде розглянуті переваги хмарних обчислювань :

- Доступ до особистої інформації з будь-якого ПК який підключений до мережі.
- Можливість праці з різного типу пристроїв.
- Немає різниці з якої ОС працювати, адже все що потрібно — це браузер. А браузери є на будь-яких ОС.
- Можливість редагувати, переглядати будь-яку інформації з будь-яких пристроїв водночас.
- В разі будь-яких технічних проблем з пристроєм не буде втрати даних, адже все знаходиться на віддаленому сховищі.
- Можливість передавати інформації іншим користувачам.
- Актуальний ринок ПЗ
- З недавніх пір — можливості двуфакторної безпеки, що надають більше захисту інформації

Щодо користування хмарних серверів:

- великим плюсом є те, що вони доступні на будь-яких платформах пристроїв. Будь то телефон, чи планшет, чи ПК — хмарним серверам все одно. Це забезпечує економію високопродуктивні програмно-апаратні комплекси на стороні користувачів хмарних сервісів. Клієнти отримують підключення та доступ до інформації до особистої робочої області в будь-якій точці підключення мережі інтернету.
- Також надійність хмарних сервісів базується на тому, що програмно-апаратна платформа заходиться в спеціальних центрах обробки даних, які 100% включаються резервування мережевої і апаратної інфраструктури.
- І насамперед це економічна вигідність. Наявність хмарних платформ не потребує у придбанні ліцензованого ПЗ, його

налаштуванням та оновленням.

Насправді переваг дуже багато, але в ході роботи були розглянуті тільки основні.

2.6.2 Недоліки

Оскільки кожна система не ідеальна, то є і недоліки

- Необхідність у постійному підключенні до мережі — оскільки все підключення потребує мережі інтернету, то для доступу також потрібен інтернет
- Недостатня кастомізація ПЗ — оскільки є обмеження по ПЗ яке знаходиться на хмарних серверах, то і кастомізація цього обладнання іноді недостатня для цілей які хоче здійснити користувач. Тобто недостатній рівень налаштування під особисті потреби.
- Конфіденційність — ця проблема нині викликає дуже багато сперечань, адже це зберігання на публічних серверах. Більшість експерти не рекомендує зберігати найбільш цінні документи для компаній на публічних серверах, так як нині, захисту який гарантував би на всі 100% конфіденційності не існує.
- Безпека — хмарне середовище саме по собі являє достатньо надійну систему, але проникнення в неї зловмисника надасть доступ до всього сховище даних. Ще один мінус в безпеці — використання віртуалізації. Тобто використання ядер стандартних ОС, що дозволяють використовувати віруси і вразливості системи.
- Дороге обладнання — для будування особистого хмарного

середовища потрібно зробити чималі внески.

- Подальша монетизація — є велика вірогідність того, що компанії будуть брати більше плату за користування їх послугами. Це зумовлено підвищенням популярності в цій галузі, а отже ріст попиту на дані послуги.

До недоліків хмарних сервісів входять:

- Необхідність до підключення у мережу інтернет. Це повинно бути стабільне підключення. В населених пунктах, особливо містах, є підвищений ризик до технічних та технологічних проблем доступу до мережі інтернет. З іншої сторони — з розвитком мобільного інтернету та підключенням LTE, недолік дещо спрощується.
- Обмеженість щодо масштабування ПЗ також є недоліком. Багато хмарних сервісів надають користувачам мінімальний набір інструментів для конфігурацій та налаштувань, що звужує можливості користувача у використанні. Відповідно — користувач найчастіше за все не може оптимально налаштувати свою роботу область.
- Недоступність впровадження даних технологій малим компаніям через дорожнечу програмно-апаратних засобів «хмари».

Тобто можна скласти невеличку діаграму



Рис 2.8 Діаграма необхідності покращення хмарних технологій

2.7 Межі керуваності для різних варіантів системи



Рис 2.9 Детальна структура

З малюнку видно, що оптимальний контроль всіх компонентів системи виникає у випадку локальної інфраструктури. Тут є контроль над всім. Від мережевих ресурсів до програм виконання. Інші варіанти, як IaaS, керування можливе лише половиною від повного списку послуг. І при використанні PaaS — всі компоненти надаються у вигляді послуг з

певними обмеженнями можливостей керування. Це зроблено, щоб користувач мав доступ до оптимальних налаштувань його потреб, які не потребують додаткових налаштувань. SaaS — користувач не має контролю. Йому лише надається крайовий продукт, а саме програма або ПЗ.

Так само виходячи з опису малюнку, можна переконатись, що дана технологія має широкий спектр можливостей, націлений на роботу з клієнтами, орієнтуючись на всі групи користувачів.

Проаналізувавши всі галузі застосування можна скласти візуалізований малюнок використання всіх хмарних сервісів



2.10 Хмарні технології у повсякденному житті

Висновок:

Хмарні технології мають дуже великий потенціал. Аналіз показує, що це значно спрощує роботу компаній. Тобто з цього виникає висновок —

великий потенціал.

Звичайно, базуючись на одному із основних плюсів хмарних технологій — це зниження витрат і на тому, що всі айти-гіганти переходять на ці технології, а це робиться не просто так, абсолютно зрозуміло що, якщо навіть вони зайшли на цю незвідану територію, то явно не збираються звідти виходити, адже концепція хмарного середовища здавалась лише гарною ідеєю. Але сьогодні, переваги хмарних технологій відчують абсолютно всі люди.

Концепція має декілька напрямків, які дозволяють розвиватися як малим компанія, так і великим гігантам. Безпека для компаній має значну перевагу, а хмарні технології це дозволяють. До того ж, це звільнює від вимог тримати все необхідне в небезпечному середовищі, як диски або приводи, або від чималих витрат на ПК, ПЗ.

Абсолютно впевнено це можна сказати і про розробників хмарних сервісів. Це зручно, економить як час і гроші, так і дає великі можливості і сферах розробки.

РОЗДІЛ 3.

Взаємодія сучасних мов програмування з хмарними сервісами

Для обробки даних та взаємодії з хмарними технологіями доволі часто

використовують мову програмування Python.

3.1 Основні відомості про мову програмування Python

Python - це високорівнева мова програмування загального призначення, який використовується в тому числі і для розробки веб-додатків. Мова орієнтований на підвищення продуктивності розробника і читання коду.

Використання цієї мови програмування має широкий діапазон. На ньому пишуть плагіни і сценарії. З його допомогою сисадміни автоматизують виконання серверних завдань. Мова допомагає створювати додатки, сайти, ботів, онлайн-ігри. Є ще багато прикладів того, що можна писати на Python.

3.2 AWS та Python. Аналіз вимог.

Проаналізувавши функціонал AWS Lambda можна впевнено сказати, що можна запуснути Python код за допомогою цього сервісу. Lambda забезпечує час запуску для коду та його подій. Тобто це фактично спрощує роботу над оптимізацією та слідкуванням часу на виконання всього. Отже, можлива економія ресурсів.

Код запускається у спеціальному середовищі що включає в себе платформу для взаємодії з цією мовою.

Є підтримка різних версій мови програмування, а отже ця платформа підходить для виконання практичного завдання. ОС- Amazon Linux.

Для розміщення коду програми потребується використання IAM консолі на сайті платформи.

Висновок

Виходячи з аналізу сучасної мови програмування, можна впевнитись, що Python може використовуватись у різних сферах діяльності. Це надає йому

своєрідну динамічність розробки.

Дослідивши можливість реалізації скрипту для отримання сигналу від супутнику використовую хмарні технології компанії Amazon, можна впевнитись, що реалізація можлива, адже є можливості для взаємодії платформи з мовою програмування, тобто — з приймачем також.

РОЗДІЛ 4.

Практична реалізація хмарного середовища з сучасною мовою програмування на базі SDR приймача

Для коректної роботи спочатку складається схема або діаграма в якій буде сплановано всі етапи роботи програми.

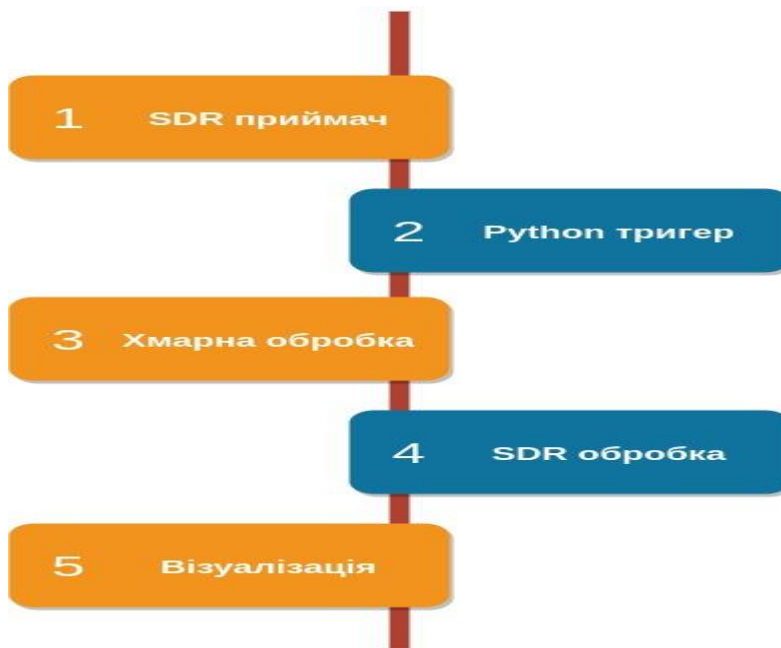


Рис 4.1 План реалізації

Як видно з діаграми, буде створено 5 основних етапів:

1) SDR-приймач — оскільки ми будемо отримувати сигнал зі супутника, то треба чимось його прийняти. Для цього підійде RTL2832U який був розглянутий до цього.

2) Оскільки супутник знаходиться на орбіті, то треба знайти оптимальне місце та час для отримання сигналу. Для цього потрібно використати онлайн сервіси з прорахуванням траєкторії його проходження над містом Київ.

3) Щоб грамотно прорахувати сигнал, потрібно налаштувати всі параметри приймання та написати скрипт для цього.

4) Обробка сигналу базуючись на прийнятих даних та записати дані.

5) Візуалізація всіх даних за допомогою спеціальних програм для взаємодії з SDR файлами.

Алгоритм складемо. Тепер буде розглянуто звідки буде поступати сигнал. Для цієї задачі використано супутник NOAA-19. Це космічний апарат національного управління океанічних та атмосферних досліджень США, метеосупутник серії POES. Цей супутник був запусканий 6 лютого 2009 року. До запуску він позначався як NOAA-N.



Рис 4.2 Метео-супутник NOAA-N

Супутник несе на собі прибори, які збирають дані для передбачення погоди та клімату. Він забезпечує отримання даних зображення хмар і деталей поверхні, а також вертикальні профілі температури та вологості. Використовуюся вони в цифрових моделях передбачення погоди. Також збираються дані о розподілу озону у верхніх шарах атмосфери і о стану біля-земного простору.

Щоб зробити розрахунки щодо розташування спершу треба знайти координати міста Києва. 50.45 — 30.52 є координати географічного центру міста. Тепер, для знаходження часу та дати проходження буде використано онлайн сервіс.

Object name	NOAA 19 Live tracking More info
Catalog #	33591 ⓘ, 2009-005A ⓘ
Observing location	79.110.131.15
Observing coord.	Lat: 50.45°, Lng: 30.52° Change
Local time zone	GMT +3 ⓘ

Рис 4.3 Налаштування для супутнику

Це данні які можливо було б вибрано для перегляду траєкторії цього супутнику, але на жаль, в найближчий час не буде збігів траєкторії з містом. Тому можна використати схожий супутник NOAA-15.

Start		Max altitude			End		Visible passes	
Date, Local time	Az	Local time	Az	EI	Local time	Az	Mag	Info
3-Jun 21:10	SW 220°	21:16	W 275°	13°	21:22	NW 329°	+7.7	Map and details
4-Jun 20:43	SSW 205°	20:50	W 269°	21°	20:57	NNW 334°	+7.2	Map and details
7-Jun 21:09	SW 220°	21:15	W 274°	13°	21:21	NW 329°	+7.7	Map and details
11-Jun 21:08	SW 219°	21:14	W 274°	14°	21:20	NW 329°	+7.7	Map and details

Legend: Not visible Marginal Good Excellent

Рис 4.4 Графічне зображення результату пошуку NOAA-15

З малюнку видно, що 3.06 буде проліт з 21-16 до 21-22. Але сигнал доволі слабкий, а у комплекті який використовується доволі слабка антена. Тому перевіримо NOAA-18

Start		Max altitude			End		Visible passes	
Date, Local time	Az	Local time	Az	EI	Local time	Az	Mag	Info
2-Jun 21:54	S 177°	22:02	W 262°	54°	22:10	NNW 342°	+5.6	Map and details
3-Jun 21:42	S 171°	21:50	WSW 251°	66°	21:58	NNW 343°	+5.4	Map and details
3-Jun 23:26	SW 226°	23:32	W 277°	11°	23:38	NW 326°	+7.4	Map and details
4-Jun 21:31	S 166°	21:38	WSW 240°	80°	21:46	NNW 344°	+5.2	Map and details
4-Jun 23:14	SW 219°	23:20	W 275°	14°	23:26	NW 330°	+7.2	Map and details
5-Jun 21:19	SSE 160°	21:27	NE 54°	84°	21:34	N 345°	+5.2	Map and details
5-Jun 23:01	SW 212°	23:08	W 272°	18°	23:15	NNW 332°	+7.0	Map and details
6-Jun 21:07	SSE 154°	21:15	ENE 68°	70°	21:22	N 347°	+5.3	Map and details
6-Jun 22:49	SSW 205°	22:56	W 269°	22°	23:03	NNW 334°	+6.8	Map and details
7-Jun 20:55	SE 148°	21:03	ENE 70°	57°	21:11	N 348°	+5.5	Map and details
7-Jun 22:37	SSW 198°	22:44	W 266°	27°	22:51	NNW 335°	+6.5	Map and details
8-Jun 22:25	S 192°	22:32	W 266°	32°	22:39	NNW 338°	+6.3	Map and details
9-Jun 22:13	S 186°	22:20	W 261°	40°	22:28	NNW 339°	+6.0	Map and details
10-Jun 22:01	S 180°	22:08	W 260°	49°	22:16	NNW 341°	+5.7	Map and details
11-Jun 21:49	S 174°	21:56	W 260°	60°	22:04	NNW 342°	+5.5	Map and details

Legend: Not visible Marginal Good Excellent

Рис 4.5 Таблиця з даними для супутника NOAA-18

Тут ситуація значно краща. Отже, виходячи з даних — найоптимальніший

час для зчитування даних буде — 03.06 з 21-50 до 21-58, або 04.06 з 21-38 до 21-46.

Тепер, буде використовуватись приймач RTL2832+820T який у стандартній комплектації та базовою антеною

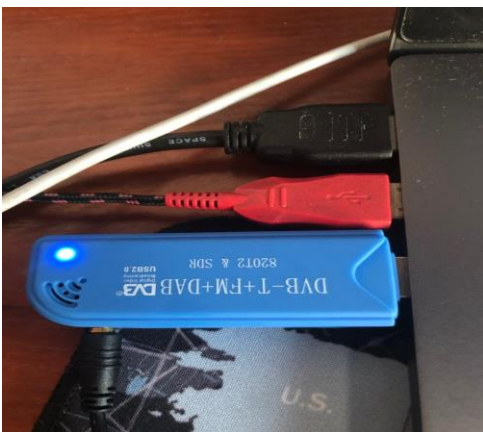


Рис 4.6 Підключення приймача в базовій комплектації

Для взаємодії з приймачем потрібна бібліотека SoapySDR яка вміє спілкуватися з мовою програмування Python та приймачем. ОС — Linux.

SoapySDR - це узагальнений API з відкритим кодом та бібліотека часу виконання для взаємодії з пристроями SDR. За допомогою SoapySDR можна створювати інстанцію, конфігурувати та передавати потоки за допомогою пристрою SDR в різних середовищах. Більшість апаратних

платформ SDR на полиці підтримуються SoapySDR, і багато програм з відкритим кодом для підтримки інтеграції апаратних засобів покладаються на SoapySDR. Крім того, SoapySDR має прив'язки для каркасів розвитку, таких як GNU Radio і Pothos.

Принципом роботи цієї бібліотеки є:

- 1) На вхід поступає Девайс модуль який має певні данні в бібліотеці девайс модуля. Або імпортований модуль що має підтримку gr-osmosdr вращпер девайсів для цієї бібліотеки
- 2) C++ оброблює данні, і якщо є підтримка на API — то передає дані далі. Це дає змогу робити доволі багато речей використовую різні мови програмування.
- 3) gr-osmosdr впроваджує права на використання osmosdr програми. Або використання різних платформ.

Оскільки приймач має свої драйвери — потрібна перевірка, яка б показувала наявність робочого приймача — тобто певний вивід у разі правильного відпрацювання скрипту при підключеному приймачі.

Весь код написаний на Python 2.7.

```
from __future__ import print_function
import SoapySDR

print("SDR:")
for x in SoapySDR.Device.enumerate("):
    print(x)
```

Вивід команди:

SDR:

Detached kernel driver

Found Rafael Micro R820T tuner

Reattached kernel driver

```
{available=Yes, driver=rtlsdr, label=Generic RTL2832U OEM :: 00000001,  
manufacturer=Realtek, product=RTL2838UHIDIR, rtl=0, serial=00000001,  
tuner=Rafael Micro R820T}
```

Тобто все працює грамотно. Приймач сигналізує свою готовність до роботи. Також можна побачити всю інформації щодо виробника, серії та тюнеру. Вся ця інформація записана безпосередньо на приймачі.

Маючи інформацію щодо часу, траєкторії та дню коли це буде можливо з максимальним коефіцієнтом корисної дії, треба мати скрипт для запуску у цей час. Маючи на увазі тригер, що запустить скрипт у цей час, роблячи якісь перевірки час від часу та сповідуючи користувача щодо часу до старту.

```
def wait_for_start(start_time):  
    while True:  
        now = datetime.datetime.now()  
        diff = int((start_time - now).total_seconds())  
        print("{:02d}::{:02d}::{:02d}: Recording will be started after {}m {:02d}s...".format(now.hour, now.minute, now.second, int(diff / 60), diff % 60))  
        time.sleep(10)  
        if diff <= 1:  
            break
```

Для цього написано деяку функцію, яка запускає скрипт у вказаний час.

Це означає, що кожні 10 секунд буде перевірка часу з відповідністю яку буде відведено у функцію

```
wait_for_start(datetime.datetime(2020, 06, 03, 21, 42, 0))
```

Час встановлено на відповідний.

Все готово для триггеру скрипту у відведений час. Для запису ефіру добре підійде формат WAVE або WAV

Цей формат є короткою формою до запису Wave Audio File Format. Він являється стандартом для зберігання потоку на персональному комп'ютері.

Він являється сферою для додатків формату формату RIFF для зберігання аудіо в «ланцюжках», це дуже нагадує формати 8SVX і AIFF, використовувані комп'ютерами Amiga і Macintosh відповідно. Це також основний формат на системах Windows для зберігання звичайного не стислого звуку. Як правило, для цього застосовується кодування методом лінійної імпульсна-кодової модуляції.

Як WAV, так і AIFF сумісні з операційними системами Windows, Macintosh або Linux. Формат також враховує деякі відмінності процесорів Intel, такі як порядок байтів від молодшого до старшого. RIFF формат виконує роль обгортки для різних кодеків аудіо стиснення.

Незважаючи на те, що WAV файл може містити стислий аудіо, найбільш звичайним його застосуванням є зберігання якраз не стислого аудіо в форматі лінійної імпульсна-кодової модуляції (LPCM). Стандартним форматом Audio-CD, наприклад, є аудіо в форматі LPCM, з 2 каналами, частотою дискретизації 44 100 Гц і 16 біт на семпл. Так як формат LPCM зберігає нестиснене аудіо, яке абсолютно ідентично оригіналу, це дозволяє використовувати його професійним користувачам і аудіо експертам для отримання максимальної якості звучання. WAV аудіо файл також може бути

змінений практично в будь-якому аудіо редакторі. WAV формат працює зі стисненим звуком в системах сімейства Windows через Audio Compression Manager (ACM). Будь-який ACM кодек може бути використаний для стиснення WAV файлу. Інтерфейс для ACM може бути доступний через безліч програм, включаючи і стандартну програму для запису звуку в деяких версіях Windows.

Для цілей запису ефіру з супутнику цей формат підходить.

Конфігурація для запису файлу потрібна складатися з назви:

HSDR_частота_дата.wav

```
file_name = "HSDR_%s_%dkHz_RF.wav" % (datetime.datetime.utcnow().strftime("%Y%m%d_%H%M%S"), frequency/1000)
```

Маючи цей запис - {available=Yes, driver=rtlsdr, label=Generic RTL2832U OEM :: 00000001, manufacturer=Realtek, product=RTL2838UHIDIR, rtl=0, serial=00000001, tuner=Rafael Micro R820T}, можна зробити висновок, що драйвер має назву rtlsdr

```
device = "rtlsdr"  
device_sdr = SoapySDR.Device({"driver": device})  
print(device_sdr)
```

Кінцевий формат повинен вийти АРТ

Automatic Picture Transmission (АРТ) - це система аналогової передачі зображень Землі, розроблена для використання на метеорологічних супутниках. Була впроваджена в 60-х роках і вже більше п'яти десятиліть (на 2019 р.) забезпечує отримання метеоснімков на відносно простих приймальних станціях по всьому світу. Прийом локальних даних на будь-якій станції можливий як мінімум двічі на добу з кожного супутника, при його проході над станцією. В АРТ супутник передає безперервну стрічку зображення тієї місцевості, над якою пролітає. У зображенні всього два

спектральних каналу, тому зображення не є кольоровим в звичайному розумінні. Фарбування знімка і накладення контурів великих орієнтирів іноді виробляються при пост-обробці для спрощення читання знімка.

Передача здійснюється за рядком. У кожен рядок горизонтального розгорнення входять дані двох каналів зображення (зазвичай позначаються Video A і Video B), дані телеметрії, спеціальні послідовності для синхронізації. Довжина кожного рядка зображення - 2080 пікселів.

Знаючи формат, треба встановити константні параметри.

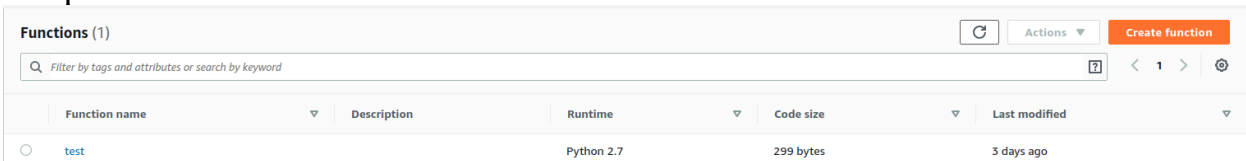
```
sdr_record(device, frequency=137912500, sample_rate=250000, gain=35, blocks_count=2100)
```

А також стандартний сетап даних для приймачів

З чого видно, що скрипт завантажує на приймач вхідні дані для

```
device.setFrequency(SoapySDR.SOAPY_SDR_RX, channel, "RF", frequency)
device.setGain(SoapySDR.SOAPY_SDR_RX, channel, "TUNER", gain)
device.setGainMode(SoapySDR.SOAPY_SDR_RX, channel, False)
device.setSampleRate(SoapySDR.SOAPY_SDR_RX, channel, sample_rate)
```

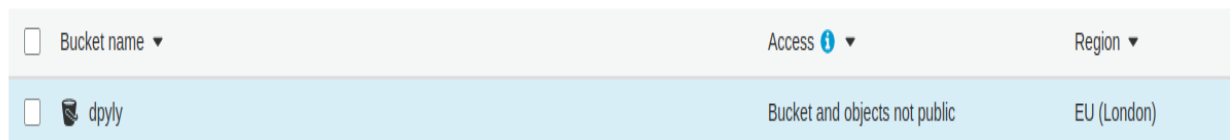
подальшого використання їх у роботі. Далі потрібно переходити на хмарні сервіси такі як AWS Lambda та AWS S3.



Function name	Description	Runtime	Code size	Last modified
test		Python 2.7	299 bytes	3 days ago


Код розміщений на сервісі Lambda і тепер сконфігуровано.

Тепер потрібне якесь сховище для даних. Використовуватись буде AWS S3.



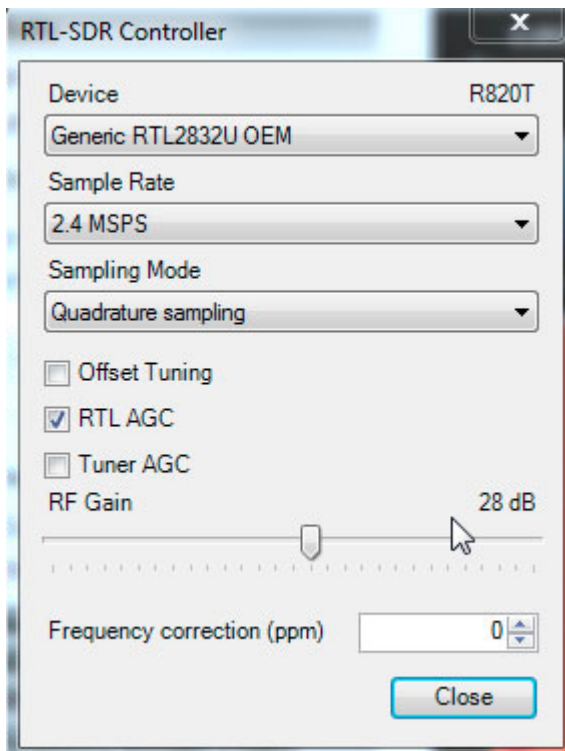
Bucket name	Access	Region
dpyly	Bucket and objects not public	EU (London)

Після відпрацювання скрипту дані будуть покладені в таблицю

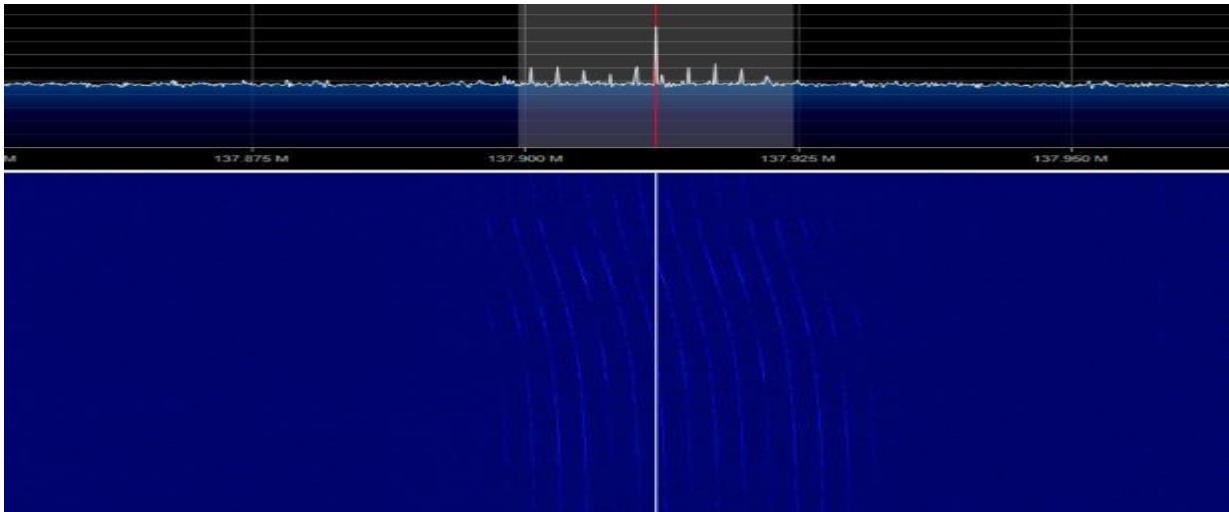
Viewing 1 to 1			
Name ▼	Last modified ▼	Size ▼	Storage class ▼
 HDSDR_20200530_182008Z_137912kHz_RF.wav	Jun 2, 2020 9:35:23 PM GMT+0300	55.5 MB	Standard

Viewing 1 to 1

Тепер потрібно якось відобразити всі дані. Для цього потрібна програма SDR.



На зображенні показані стандартні налаштування для приймача SDR. Далі встановимо потрібний формат.



Виходячи з інформації малюнка можна сказати, що помітний допреровський зсув на спектрі, який пройшов від того, що супутник пройшов мимо міста. Сигнал помітний, але не сильно. Для більш точних даних потрібно використовувати спеціальні антенни для приймання NOAA.

Висновок

Перш за все була сформована діаграма, яка покладена за основу для цієї реалізації.

Досліджено метеосупутники NOAA, їх принцип роботи та деякі історичні відомості. А також його основну систему передачі зображень АРТ.

Практична реалізація показала, що завдяки звичайному приймачу, сучасній мові програмування, а також хмарних технологій, можливо отримати сигнал з метео-супутника, записати його та навіть відобразити.

SD

Р приймачі мають потужні програми для обробки, візуалізації та вирахування сигналів, без застосувань сучасних лабораторій та дорогої техніки.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Були розглянуті різновиди SDR приймачів, сфери їх використання історичні відомості, можливості застосування, а також принцип роботи, ідеальні варіанти побудови, а саме головне — проведено докладний аналіз Переваг та недоліків даної технології. Було обрано оптимальний варіант для практичної реалізації програми. Як висновок отримано, що за невеликі гроші та мінімальну комплектації можливо зібрати потужний приймач з доволі об'ємним спектром можливостей, та використанням у різних цілях. А також, обширні можливості для цього напряму.

Був проведений аналіз характеристик SDR приймача RTL2832U, його характеристик, та схем реалізації. Доведено, що даний приймач підходить для реалізації практичної частини у базовій комплектації.

Було розглянуто основні поняття про хмарні технології, їх сфери застосування. Проведений детальний розгляд класифікації даної технології. Більш детально розглянуті межі керованості для різних видів діяльності. Охарактеризовані основні типи, переглянуто основні переваги та недоліки. Та зроблено висновок, що дана технологія має величезний потенціал, а її

використання виходить на новіший рівень щорічно.

Також в роботі була реалізована практичний скрипт для перевірки працездатності роботи звичайного SDR приймача в аматорських цілях, для отримання сигналу від метео-супутнику. За результатами досліджень можна зробити висновки:

1) Принцип роботи SDR приймача доволі простий і це надає можливості у легкому використанні на різних ОС, та для різних цілей.

2) Сучасні мови програмування чудово спілкуються з SDR приймачами, а також мають можливість взаємодіяти з хмарними технологіями.

3) Існує дуже багато бібліотек для радіоприймачів, які чудово взаємодіють з ними. А також дуже багато програм, які надають можливість працювати без спеціальних пристроїв.

4) WAVE — чудовий формат для запису радіосигналу.

5) Можливість за допомогою SDR приймача та SDR програми переглянути панораму радіоефіру як він є. Зробити заміри.

6) SDR відкриває можливості широкому діапазону, а також дуже далекому зв'язку.

7) SDR доволі вимогливі. SDR приймач та робота програми SDR дуже багато займають CPU. Тобто при віддаленій роботі — мала ефективність із-за вимог до підтримки ПЗ. Виходячи з загального положення і спираючись на сукупність всіх раніше перерахованих вище і згаданих фактів робиться висновок, що батареї пристрою, без постійної зарядки, довго не вистачить.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. John Rhoton. Cloud Computing Explained: Implementation Handbook for Enterprises 2-nd Edition, 2009
2. Mark Pilgrim Dive into Python, 2004
3. Фокин Г.А Технологии программно-конфигурируемого радио
4. Software Defined Radio: Architectures, Systems and Functions (Markus Dillinger, Kambiz Madani, Nancy Alonistioti) Page xxxiii (Wiley & Sons, 2003)
5. Using DVB USB Stick as SDR Receiver
<https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr>