

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,

канд. техн. наук, доцент

_____ Я. М. Крайник

«__» _____ 20__ р.

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА
АПАРАТНО-ПРОГРАМНИЙ МОДУЛЬ ДЛЯ ДИСТАНЦІЙНОГО
НАВЧАННЯ НА БАЗІ ОДНОПЛАТНОГО КОМП'ЮТЕРУ

Спеціальність «Комп'ютерна інженерія»

123 – КМР.1 – 605.21610101

Студент

_____ А. С. Абрамова
підпис

«__» _____ 20__ р.

Керівник канд. техн. наук, доцент

_____ Я. М. Крайник
підпис

«__» _____ 20__ р.

Миколаїв – 2022

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП.....	5
РОЗДІЛ 1 АНАЛІТИЧНА ЧАСТИНА. ДОСЛІДЖЕННЯ СУЧАСНИХ ЗАСОБІВ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	7
1.1 Дистанційне навчання як одна з форм організації навчального процесу	8
1.2 Роль викладача у корпоративному викладацькому середовищі.....	14
1.3 Сучасні технології навчання: труднощі та шляхи їх вирішення	17
Висновки до розділу 1.....	18
РОЗДІЛ 2 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	19
2.1 Raspberry Pi.....	19
2.1.1 Основні складові одноплатного комп'ютеру Raspberry Pi.....	20
2.1.2 Переваги та недоліки використання одноплатного комп'ютера Raspberry Pi.....	27
2.2 PyCharm	29
2.2.1 Ключові особливості	31
2.3 Microsoft Visual Studio.....	32
2.4 Anaconda	35
2.5 Python	40
2.5.1 Ключові терміни.....	43
2.6 Flask.....	44
2.7 C++	46

2.7.1 Переваги та недоліки.....	50
2.8 NumPy	51
2.9 “libc++” C++ Standard Library	52
Висновки до розділу 2.....	54
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ.....	55
3.1 Logitech HD Pro Webcam C920	55
3.2 Перелік основних аналогів використаного одноплатного комп'ютера	58
3.3 Типи (архітектури) систем відеоконференцій	61
3.4 Принцип роботи апаратно-програмного модулю.....	64
Висновки до розділу 3.....	70
ВИСНОВКИ	72
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	74
ДОДАТОК А Лістинг коду файлу setup.py.....	78
ДОДАТОК Б Лістинг коду файлу video-streaming.pyx	79
ДОДАТОК В Лістинг коду файлу sender.py.....	83
ДОДАТОК Г Лістинг коду файлу receiver.py.....	85
ДОДАТОК Д Лістинг коду файлу VideoDecoder.cpp.....	87
ДОДАТОК Е Лістинг коду файлу VideoDecoder.h	91
ДОДАТОК Ж Лістинг коду файлу VideoEncoder.cpp.....	92
ДОДАТОК И Лістинг коду файлу VideoEncoder.h	96

ПЕРЕЛІК СКОРОЧЕНЬ

ADSL – Asymmetric Digital Subscriber Line

CSI – Scalable Coherent Interface

GPIO – General Purpose Input Output

HDMI – High Definition Multimedia Interface

I2C – Inter-Integrated Circuit

IDE – Integrated Drive Electronics

NTP – Network Time Protocol

RCA – Radio Corporation of America

RTC – Real-Time Clock

SBC – Session Border Controller

SDHC – Secure Digital High Capacity

SDXC – Secure Digital eXtended Capacity

SoC – System-on-a-Chip

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver-Transmitter

VCS – Version Control System

VoIP – Voice over IP

АЦП – Аналогово-Цифровий Перетворювач

ВСТУП

Актуальність теми

У ситуації пандемії COVID-19 освіта опинилася у становищі, коли віддалені інструменти та технології стають незамінними для всього освітнього процесу. Он-лайн платформи з безліччю функцій, інструменти для онлайн-зустрічей набули помітної популярності за останній період.

Метою цієї роботи є створення рішення для віддаленої організації навчального процесу, що дозволяє використовувати одноплатний комп'ютер під час віддалених сесій.

Характерними рисами дистанційної освіти є:

– *Гнучкість* – учні в системі дистанційної освіти працюють у зручному місці та в зручному темпі, у зручний для себе час, де кожен може навчатися стільки, скільки йому особисто необхідно для освоєння предмета та отримання необхідних іспитів за вибраними курсами;

– *Модульність* – кожен курс створює цілісне уявлення про певну предметну область, що дозволяє формувати навчальну програму за індивідуальними та груповими потребами;

– *Спеціалізований контроль якості навчання* – використовуються дистанційно організовані іспити, співбесіди, практичні, курсові та проєктні роботи, комп'ютерні інтелектуальні системи тестування;

Метою роботи є розробка апаратно-програмного модулю для дистанційного навчання на базі одноплатного комп'ютеру. Завдяки тому, що у Чорноморському національному університеті ім. Петра Могили (Миколаїв, Україна) є дві бездротові мережі, що відповідають обом описаним випадкам, з'єднання було підготовлено для обох мереж.

Доступ до SBC було надано студентам, які вивчають вбудовувані комп'ютерні системи реального часу (рівень магістра). У ході курсу їм вдалося провести лабораторні роботи з внутрішньою інфраструктурою SBC, підключеними компонентами, програмуванням та налагодженням програмного забезпечення.

Об'єкт: процес програмування та створення апаратно-програмного модулю для дистанційного навчання на базі одноплатного комп'ютеру.

Предмет: комп'ютер Raspberry Pi та плата Arduino з Ethernet-екранами, інтегрованими в локальну мережу.

Система має складний API для зв'язку і величезний набір пристроїв, що підключаються. Система управління реалізована з використанням гіпертекстового протоколу та доступна у вигляді веб-проекту. Проте проект ArPi Lab переважно орієнтований працювати з контролерами Arduino. Raspberry Pi грає роль сервера і може бути запрограмований під час курсу. Оскільки одноплатні комп'ютери є потужнішими пристроями, бажано мати додатковий доступ до таких пристроїв.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- виконати аналіз літератури та патентної інформації з аналогів пристроїв, які можуть забезпечити стабільне та захищене з'єднання для забезпечення процесу дистанційного навчання;
- змодельовати та спроектувати прототип під керуванням комп'ютеру Raspberry Pi та плати Arduino з Ethernet-екранами, інтегрованими в локальну мережу;
- виконати тестування роботи створеної бета-версії застосунку;
- розробити питання з охорони праці та безпеки життєдіяльності.

Кваліфікаційна магістерська робота присвячена вирішенню цієї проблеми з використанням наявної інфраструктури університету.

РОЗДІЛ 1 АНАЛІТИЧНА ЧАСТИНА. ДОСЛІДЖЕННЯ СУЧАСНИХ ЗАСОБІВ ОРГАНІЗАЦІЇ ДИСТАНЦІЙНОГО НАВЧАННЯ

Головним фактором тривалого успіху будь-якого бізнесу є ефективний та результативний менеджмент. В умовах існуючих рівнів та темпів розвитку технологій та економіки, коли ринок переповнений альтернативними, практично ідентичними постачальниками, продуктами, потенційними партнерами, а найважливішу роль в організації починає відігравати її інтелектуальний потенціал, стратегічний менеджмент стає найважливішим фактором успіху бізнесу, забезпечуючи оптимальне використання всіх ресурсів сьогоднішній день і в майбутньому. Стратегічний менеджмент є безпосереднім завданням топ-менеджменту організації, що пред'являє до керівництва досить жорсткі вимоги щодо кваліфікації та володіння певними навичками. Сучасні технології дозволяють знаходити рішення в галузі управління організацією та персоналом як найважливішим виробничим ресурсом.

Ефективність роботи топ-менеджменту залежить насамперед від результативності його комунікацій зі своїми контактними групами. Діяльність сучасного топ-менеджера складається насамперед із регулярних комунікацій зі своїми контактними групами – керівниками владних структур, стратегічними партнерами, інвесторами, персоналом організації. Ефективність роботи топ-менеджера залежить, насамперед, від результативності цих комунікацій, тобто. досягнення та підтримання партнерських відносин із представниками контактних груп [5].

Навчання персоналу робочому місці є важливою частиною корпоративної системи розвитку компетенцій співробітників. Цей аспект діяльності організації заслуговує на особливу увагу, оскільки є одним з найбільш значущих елементів ведення та розвитку бізнесу. З кожним роком 2022 р.

все більшої популярності в бізнес-спільноті набувають корпоративні тренінги, мета яких – підвищення кваліфікації та покращення навичок співробітників організації.

Ми живемо в епоху, коли знання та безперервне навчання є найважливішими компонентами успіху. Тому лідерами нової економіки виявляються ті організації, які навчилися ефективно управляти знаннями (аналізувати, систематизувати, інтегрувати їх у нові продукти та послуги) швидше, ніж конкуренти [13].

Сьогодні незаперечний те що, з основних і найважливіших капіталів, якими володіє організація, є кваліфіковані фахівці, які мають певними знаннями, навичками і технологіями, тобто. людський капітал. І дійсно, кожна організація, - це самостійна і самонавчається система: співробітники постійно навчаються у своїх керівників, один у одного, у партнерів, у своїх конкурентів. Іноді це відбувається стихійно, несвідомо.

Що стосується системності, то організація, що поставила за мету бути першою, успішною, швидко реагувати на зміни у зовнішньому середовищі, приймати випереджаючі рішення, впроваджувати нові розробки та технології, неминуче має стати безперервно учнями. Розвиток та навчання персоналу – це невід'ємна та важлива інвестиція, яка при грамотному та системному підході дасть високі результати. Саме наставництво є тим комплексом заходів, що дозволяє сформувати ефективний та постійний бізнес-процес передачі досвіду, знань, розвитку навичок у співробітників організації [1].

1.1 Дистанційне навчання як одна з форм організації навчального процесу

Залежно від способу комунікації викладачів та учнів виділяють такі методи дистанційного навчання:

– Метод навчання за допомогою взаємодії того, хто навчається, консультується або взаємодіє з освітніми ресурсами за мінімальної участі викладачів, репетиторів, консультантів, наукових та технічних керівників (самонавчання). Для здійснення цього методу викладачами, репетиторами створюються та підбираються різні освітні ресурси: друковані, аудіо- та відеоматеріали, а також навчальні посібники, що доставляють телекомунікаційними мережами (інтерактивні бази даних, електронні видання та комп'ютерні навчальні системи).

– Метод індивідуалізованого викладання та навчання, для якого характерні взаємини одного учня, консультованого студента чи школяра, клієнта, який потребує науково-технічних послуг, здобувача наукового ступеня з одним викладачем, репетитором, консультантом або науковим та технічним керівником (навчання «один до одного»). Цей метод може реалізуватися в дистанційному навчанні переважно за допомогою таких технологій, як телефон, голосова пошта, факс, електронна пошта, система скайп.

– Метод, в основі якого лежить виклад навчального матеріалу викладачем, при цьому учні не відіграють активної ролі в комунікації (навчання «один до багатьох»). Даний метод використовується педагогом, репетитором, консультантом, коли навчається і консультується ціла група, вони приблизно однаково підготовлені і для всіх однаковий кінцевий результат. Наприклад, це відбувається при підготовці школярів репетитором до ЄДІ, або при консультуванні студентів з різних дисциплін. Цей метод, властивий традиційної освітньої системі, набуває нового розвитку з урахуванням сучасних інформаційних технологій. Так, лекції, записані на аудіо- або відеокасети, що читаються по радіо або телебаченню, доповнюються в сучасному дистанційному навчанні так званими

електронними лекціями, що розповсюджуються по комп'ютерних мережах за допомогою систем дошок оголошень. Електронна лекція, яку готують та підбирають викладачі, репетитори, консультанти може являти собою добірку статей або витягів із них, а також навчальних матеріалів, які готують учнів до майбутніх дискусій.

– Метод, для якого характерна активна взаємодія між усіма учасниками навчального процесу (навчання «багато хто до багатьох»). Цей метод орієнтований на групову роботу студентів та становить найбільший інтерес для дистанційного навчання. Він передбачає широке використання дослідницьких та проблемних способів навчання. Роль викладача при такому навчанні зводиться до того, що він ставить тему для студентів, школярів або для претендентів на наукові ступені (ставить навчальне завдання), а далі він повинен створити і підтримувати таке сприятливе середовище спілкування і психологічний клімат, при яких учні могли б працювати в співробітництво. Викладач відповідає за координацію, управління ходом дискусії, і навіть за підготовку матеріалів, розробку плану роботи, обговорюваних питань, і тем.

– Метод проєктів передбачає комплексний процес навчання, що дозволяє учню проявити самостійність у плануванні, організації та контролі своєї навчально-пізнавальної діяльності, результатом якої є створення будь-якого продукту чи явища. У основі методу проєктів лежить розвиток пізнавальних, творчих інтересів учнів, умінь самостійно формувати знання.

– Метод проблемного навчання заснований на розгляді складних пізнавальних завдань, вирішення яких становить суттєвий практичний чи теоретичний інтерес. У процесі проблемного навчання увагу учнів фокусується на важливих проблемах, вони стимулюють пізнавальну активність, сприяють розвитку умінь та навичок у вирішенні цих проблем. Роль викладача зводиться до спостереження та підтримки, але не більше.

– Дослідницький метод навчання характерний наявністю чітко поставлених актуальних та значущих для учасників цілей, продуманої та обґрунтованої структури, широкого використання арсеналу методів дослідження, використання наукових методів обробки та оформлення результатів [2].

Останні кілька років в освітній сфері стали активно застосовуватися онлайн-семінари або вебінари. Вебінар – це форма веб-конференції, яка включає проведення онлайн-нарад, демонстрацію презентацій через мережу Інтернет в режимі реального часу та інші інтерактивні можливості. Під час вебінару кожен учасник використовує свій персональний комп'ютер, зв'язок між учасниками підтримується через Інтернет за допомогою спеціалізованого програмного забезпечення, встановленого на комп'ютері кожного учасника, або через веб-додаток. Останній спосіб проведення вебінару дуже зручний, тому що не вимагає розгортання на робочих станціях учасників конференції спеціалізованого програмного забезпечення [3].

Вебінари можуть використовувати різні інтерактивні взаємодії, включати сеанси голосувань і опитувань, що забезпечує повну взаємодію між аудиторією і викладачем. Визначення «інтерактивна взаємодія» часто використовується у спеціальній літературі. Інтерактивна взаємодія – це спілкування студента з іншими учасниками вебінару через програмний комплекс. За більш просунутих можливостей ведення спілкування реалізується можливість вибору варіантів змісту навчального контенту та режиму навчання. Чим більше можливостей у програмного комплексу, тим активніше студент бере участь у процесі навчання та спілкуванні з іншими учасниками вебінару, тим вище інтерактивність. У загальному сенсі інтерактивна взаємодія передбачає спілкування будь-яких учасників, залучених до навчального процесу, один з одним із використанням

доступних їм можливостей. При проведенні вебінарів дуже часто застосовується технологія «електронна дошка». Насамперед цей інтерактивний інструмент призначений для коментарів, він дозволяє викладачеві та слухачам залишати позначки або коментувати пункти слайдової презентації [4].

Завдяки використанню глобальної мережі Інтернет, викладач та студенти вебінару можуть перебувати на величезних відстанях один від одного, та брати участь в одному освітньому процесі. Без використання сучасних технологій зібрати аудиторію разом було б досить складно чи взагалі неможливо. Не слід розуміти вебінар як односторонню трансляцію. Ведучий, як правило, надає право учасникам вебінару поставити запитання, що їх цікавлять, або висловити свою точку зору з обговорюваної теми. Завдяки використанню веб-камер співрозмовники можуть бачити один одного, що також позитивно впливає на освітній процес. Під час проведення занять часто застосовуються засоби віддаленого контролю комп'ютерів, за допомогою яких викладач може безпосередньо допомогти студенти, якщо у останнього виникли якісь труднощі при виконанні практичного завдання. Такі засоби як JoinMe та TeamViewer дозволяють ефективно керувати програмами на стороні віддаленого комп'ютера [14].

Зазвичай, форма дистанційної освіти передбачає самостійне виконання студентами практичних завдань. Для того, щоб централізовано зберігати виконані завдання, навчальні матеріали та інші документи, необхідні для навчального процесу, часто використовуються хмарні сховища даних. Хмарне сховище даних – це вид онлайн-сховища, в якому інформація та дані користувачів зберігаються на кількох розподілених у мережі Інтернет серверах, що надаються в користування клієнтам. Використання хмарного сховища найчастіше набагато зручніше, ніж використання власних виділених

серверів. З погляду користувача, хмарне сховище представляє їх цілісний віртуальний сервер. Прикладами хмарних сховищ є послуги DropBox і GoogleDrive. Слід зазначити, що більшість хмарних сховищ надають певний обсяг свого дискового простору для безкоштовного використання. Декілька гігабайт зазвичай вистачає для освітніх потреб. Додатковим (а в деяких випадках навіть основним) засобом комунікації між учасниками освітнього вебінару можуть бути VoIP-програми [15]. Voice over IP або VoIP, – загальна назва мережевих протоколів та програмних засобів, що дозволяють передавати по IP-мережах аудіо інформацію у цифровому вигляді. VoIP-програми надають користувачам надійний і недорогий канал зв'язку, за допомогою якого можна здійснювати аудіо та відеодзвінки, організувати конференції. Популярними програмами VoIP в даний час є Skype, Google Talk, Twinkle та інші. Більшість засобів IP-телефонії також підтримують шифрування даних, що дозволяє успішно їх використовувати і для передачі закритої внутрішньокорпоративної інформації [16].

Часто, при дистанційному навчанні у вищих навчальних закладах від студентів не потрібно постійно перебувати в аудиторії. У більшості робочих програм вищих навчальних закладів, що реалізують дистанційне навчання, все ж таки проходять очні заняття вечорами або вихідними [6].

Ці заняття, як правило, не обов'язкові для відвідування, однак, часто бувають корисні для вироблення у студентів різноманітних практичних навичок. Велика кількість великих компаній підтримують власні центри дистанційного навчання, щоб уніфікувати та покращити якість підготовки власних співробітників. У внутрішньокорпоративній сфері освіти та підвищення кваліфікації працівників застосовуються спеціальні системи дистанційного навчання, які забезпечують найефективнішу побудову

освітніх програм. До таких систем відносяться WebTutor, Competentum, Moodle та інші.

Великі ІТ-корпорації створюють великі навчальні портали для підготовки співробітників або зовнішніх покупців своїх продуктів. При цьому деякі курси надаються безкоштовно або входять до комплексу програмного забезпечення. Можна сказати, що у сфері інформаційних технологій дистанційні методи навчання вже поширені повсюдно та міцно зайняли свою нішу [9].

1.2 Роль викладача у корпоративному викладацькому середовищі

Наставництво – це одна з форм навчання на робочому місці, акцент при цьому робиться на практичній діяльності. У процесі такого навчання досвідченіший і кваліфікований співробітник передає новачкові знання та допомагає йому формувати навички, необхідні для ефективного виконання професійних обов'язків. Іншими словами, наставництво спрямоване на розвиток прикладних професійних компетенцій працівника та адресоване лінійним спеціалістам. Наставництво дозволяє співробітнику професійно вдосконалюватись, не залишаючи робочого місця. Процес розвитку професійних навичок у цьому випадку є постійним та планомірним, а як навчальний посібник виступають виробничі завдання. Функція наставника полягає у навчанні чи демонстрації того, як потрібно виконувати ті чи інші операції, обов'язковому контролю діяльності підопічного, в оцінці її результатів. Наставництво покликане допомогти новому співробітнику вникнути у специфіку виробництва та (або) надання послуг, отримати уявлення про основні етапи цих процесів. Навчальний повинен розуміти, що треба робити, як потрібно робити (засоби досягнення результату) і для чого це робити (мета діяльності) [7].

Грамотно вибудована та функціонуюча система наставництва дозволяє досягти наступних ключових результатів:

- скорочення часу виходу плановий рівень виробництва;
- підтримання високої якості продукції та (або) послуги, що надається, відповідність корпоративним стандартам;
- підвищення рівня мотивації співробітників;
- профілактика професійного «вигорання»;
- підвищення лояльності співробітників, їх прихильність до цінностей організації, зміцнення командного духу [10].

Таблиця 1. Профіль компетенції викладача

Компетенція	Опис	Поведінкові індикатори
Комунікбельність	Зацікавлення у спілкуванні. Уміння вступати у розмову, підтримувати бесіду, зрозуміло викладати думки в усній та письмовій формах. Вміння бути тактовним та делікатним.	Створює приємне враження, сприяє своєму; легко встановлює контакт із співрозмовниками; ввічливий і делікатний; чітко та ясно викладає свої думки; мова барвиста, емоційна, використовує яскраві образи; знаходить компромісні рішення у спірних ситуаціях.
Навички командної роботи	Прагнення співпраці з іншими задля досягнення загального результату. Надання допомоги колегам у вирішенні професійних завдань.	Надихає членів команди, робить відчутний внесок у загальний результат; охоче ділиться інформацією, завжди готовою до діалогу; відсуває на другий план власні цілі заради цілей команди; враховує у своїй роботі побажання та запити суміжних підрозділів;

<p>Педагогічні здібності</p>	<p>Бажання передати накопичений професійний та життєвий досвід. Прояв ініціативи у громадській діяльності. Готовність показати гнучкість у поведінці.</p>	<p>Спостерігає за діями нового працівника; надає зворотний зв'язок новому співробітнику; виявляє терпіння під час навчання, показуючи ті чи інші дії, дохідливо викладає матеріал; вміє слухати, розуміє потреби та бажання учня; використовує ефективні методи та способи навчання нового співробітника; гнучко реагує на зміни (на зміну емоційного фону в ході спілкування тощо)</p>
<p>Орієнтація на розвиток</p>	<p>Прагнення отримання нових знань і досвіду. Прагнення до професійного та особистісного розвитку. Удосконалення професійних компетенцій, пов'язаних із новою та більш складною роботою.</p>	<p>Має довгостроковий план особистого та професійного розвитку; береться за вирішення суміжних завдань із метою подальшого професійного розвитку; відстежує останні досягнення у своїй професійній галузі; підтримує професійний розвиток колег; застосовує отримані знання практично; самостійно визначає свої слабкі місця.</p>
<p>Корпоративна ідентичність</p>	<p>Дотримання корпоративних стандартів та правил. Орієнтація на місію та цілі організації. Розуміння свого індивідуального внеску до загального результату.</p>	<p>Має уявлення про місію та цілі організації; планує свою роботу відповідно до стратегічних та тактичних бізнес-цілей організації; слідує корпоративним стандартам і правилам; знає зміст внутрішніх нормативних документів; розуміє, як його особистий внесок узгоджується з місією організації; виявляє лояльність до організації, розглядає компанію як єдине можливе місце роботи.</p>

1.3 Сучасні технології навчання: труднощі та шляхи їх вирішення

В рамках регіонального та вузівського компонентів навчального закладу спільно з організаціями-роботодавцями та їх асоціаціями необхідно актуалізувати вміння та навички, що відповідають вимогам до працівників конкретної організації чи галузі. Варто також сформувати набір компетенцій, які вимагаються від випускника. При цьому цей підхід може бути реалізований для підготовки фахівців за програмами як вищої, так і додаткової професійної освіти, при цьому слід враховувати, що в рамках програм різного рівня підготовки можливе формування різних груп компетенцій:

– для програм вищої освіти, – фундаментальна підготовка спеціаліста, націлена на формування комплексу універсальних (загальнонаукових, інструментальних, соціально-особистісних та загальнокультурних), а також професійних (загальнопрофільних, предметно-спеціалізованих) компетенцій;

– для програми професійної перепідготовки; – орієнтація на переважне формування професійних (загальнопрофесійних, предметно-спеціалізованих) компетенцій;

– для програми підвищення кваліфікації; – орієнтація на переважне формування професійних, насамперед предметно-спеціалізованих, компетенцій [8].

Програми підготовки фахівців у галузі інноваційної діяльності, що розробляються нині в рамках освітнього проєкту, будуть засновані на формуванні карт компетенцій фахівців даної сфери, що визначаються спільно з організаціями-роботодавцями. В рамках проєкту планується проведення додаткових досліджень для уточнення карт компетенцій менеджерів у галузі інноваційної діяльності (залежно від специфіки сфери роботи). Сформовані характеристики компетентнісних профілів визначатимуть траєкторії

2022 р. Абрамова А. С. 123 – КМР.1 – 605.21610101

навчання, а також формуватимуть структуру та зміст конкретних дисциплін та модулів [12].

Викладач дисциплін повинен не тільки досконало знати теорію предмета, але й володіти практичними аспектами тих чи інших бізнес-процесів, мати досвід роботи у відповідній галузі [11].

Висновки до розділу 1

Вебінар – це форма веб-конференції, яка включає проведення онлайн-нарад, демонстрацію презентацій через мережу Інтернет в режимі реального часу та інші інтерактивні можливості. Під час вебінару кожен учасник використовує свій персональний комп'ютер, зв'язок між учасниками підтримується через Інтернет за допомогою спеціалізованого програмного забезпечення, встановленого на комп'ютері кожного учасника, або через веб-додаток. Останній спосіб проведення вебінару дуже зручний, тому що не вимагає розгортання на робочих станціях учасників конференції спеціалізованого програмного забезпечення.

Грамотно вибудована та функціонуюча система наставництва дозволяє досягти наступних ключових результатів:

- скорочення часу виходу плановий рівень виробництва;
- підтримання високої якості продукції та (або) послуги, що надається, відповідність корпоративним стандартам;
- профілактика професійного «вигорання»;
- підвищення лояльності співробітників, їх прихильність до цінностей організації, зміцнення командного духу.

РОЗДІЛ 2 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Raspberry Pi

Raspberry Pi, ефективний та економічний комп'ютер розміром із кредитну картку, представлений компанією United Kingdom-Raspberry Pi з метою просвітити та розширити можливості викладання інформатики у школах та інших країнах, що розвиваються. З моменту свого створення різні спільноти відкритого вихідного коду зробили величезний внесок у розвиток відкритого програмного забезпечення.

Raspberry Pi з моменту свого запуску регулярно перебуває у процесі постійного розвитку та вдосконалення як апаратного, так і програмного забезпечення, що, у свою чергу, робить Pi «Fully Fledged Computer» з можливістю одночасного вирішення багатьох завдань, що вимагають великих обчислювальних ресурсів [17].

2.1.1 Основні складові одноплатного комп'ютеру Raspberry Pi

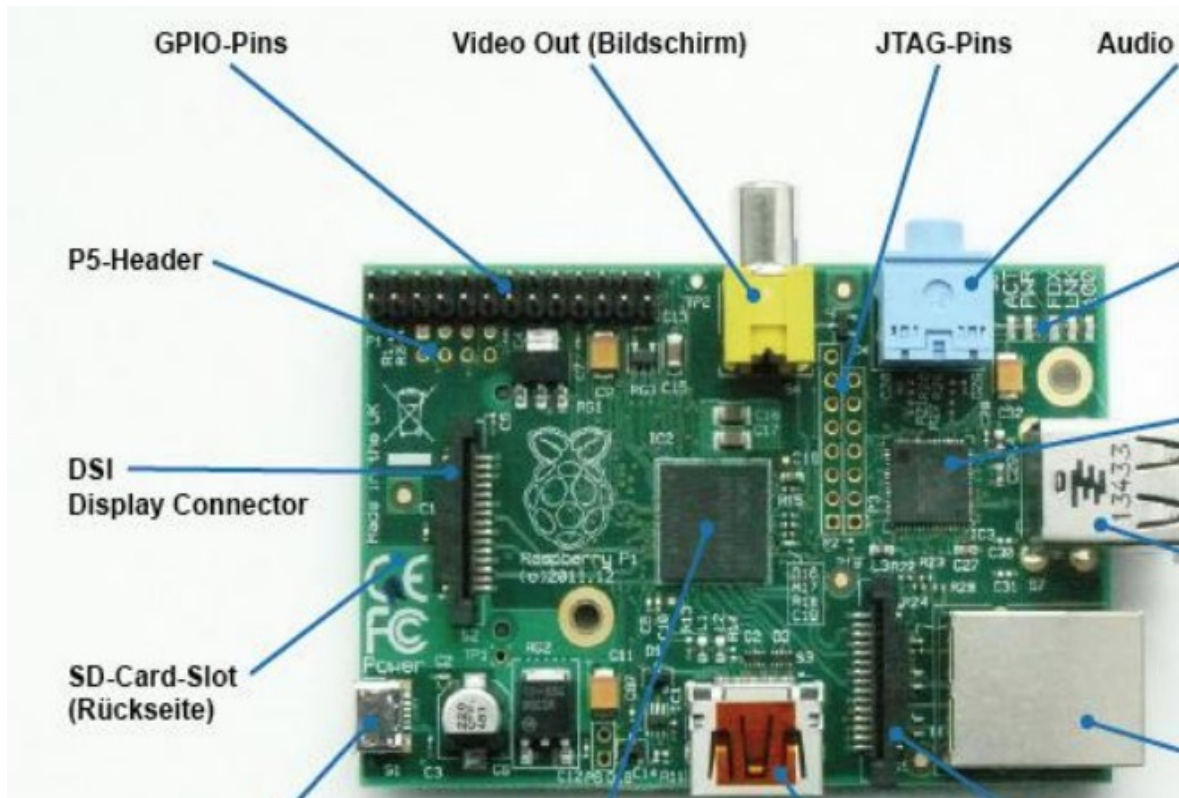


Рисунок 2.1 – Зображення Raspberry Pi та основних його компонентів

Джерело живлення micro-USB: для Raspberry Pi потрібно 5 вольт (В) +/- 5% для стандарту USB 2.0. Перелічення компонентів основних моделей Raspberry Pi: модель B: 700 мА при 3,5 Вт (Вт); модель A: 500 мА при 2,5 Вт. Печатна плата Raspberry Pi є інтерфейсом Micro-USB типу B, тому Pi-сумісний блок живлення використовує стандартний USB A з одного боку роз'єм, а з іншого – роз'єм MicroUSB B.



Рисунок 2.2 – Джерело живлення USB Type B

Роз'єм для SD-карти: слот для цифрової карти (SD-карта) — це твердотільний знімний пристрій зберігання даних, який необхідний для запуску операційних систем на Raspberry Pi, оскільки Raspberry Pi не має вбудованої пам'яті та інших пристроїв для зберігання. Raspberry Pi підтримує як SDHC (Secure Digital High Capacity), так і SDXC (Secure Digital eXtended Capacity).



Рисунок 2.3 – SD карта у роз'ємі одноплатного комп'ютера

Порти USB і порт Ethernet: Raspberry Pi Model B складається з 2 портів USB 2.0, тоді як модель B+ складається з 4 портів USB 2.0. Порти USB дозволяють підключати зовнішні периферійні пристрої, такі як клавіатура, миша, USB-концентратор, ключ Wi-Fi тощо.



Рисунок 2.4 – Порти USB і порт Ethernet

Порт Ethernet: щоб увімкнути підключення до Інтернету та оновити програмне забезпечення або встановити останні пакети з онлайн-сховищ, Raspberry Pi підтримує Ethernet Connection. Raspberry Pi (кожна модель) складається з гнізда Ethernet RJ45, який підтримує кабелі CAT5/6. Це дозволяє Raspberry Pi підключатися до бездротового маршрутизатора, моделі ADSL або будь-якого іншого пристрою для спільного доступу до Інтернету.

HDMI (High Definition Multimedia Interface): порт HDMI дозволяє підключати Raspberry Pi до HDTV за допомогою кабелю HDMI. Raspberry Pi

підтримує максимальну роздільну здатність 1920x1200. За допомогою HDMI Full HD MPEG-4 можна транслювати через HDMI.



Рисунок 2.5 – HDMI порт на Raspberry Pi

Відеовихід (кабель RCA): на додаток до підключення HDMI, яке полегшує підключення HD, Raspberry Pi також має можливість підключення до стандартного монітора або телевізора за допомогою відеокабелю RCA. Кабель RCA менш дорогий у порівнянні з HDMI, але разом із кабелем RCA користувач повинен придбати стереокабель 3,5 мм.

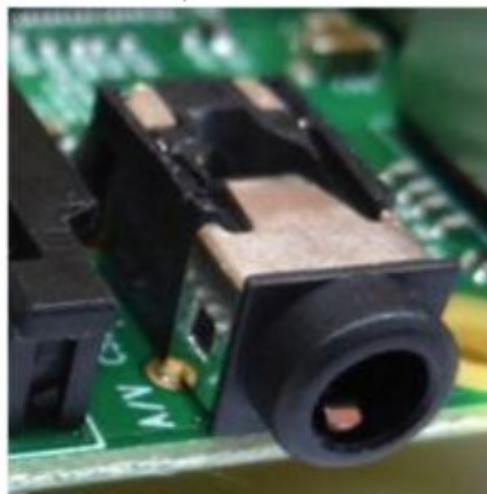


Рисунок 2.6 – Відеовихід (кабель RCA)

Світлодіоди стану: Raspberry Pi складається з 5 основних світлодіодів, які виконують такі функції:

– АСТ: (зелений): Основна функція світлодіода АСТ – показувати статус карти. Зазвичай блимає під час будь-якої дії з SD-карти, яку виконує кінцевий користувач.

– PWR: (червоний): Основною функцією світлодіода PWR є живлення. Цей світлодіод постійно світиться, коли Raspberry Pi увімкнено, і продовжує горіти до вимкнення.

– FDX: (оранжевий): Основна функція світлодіода FDX – повний дуплекс. Цей світлодіод вмикається, коли з'єднання Ethernet має дуплексний тип.

– LNK: (оранжевий): функція, яку виконує світлодіод LNK, – це посилення. Цей індикатор світиться, коли встановлено з'єднання Ethernet і починається передача пакетів.

– 100: (оранжевий): ціль 100 світлодіодів – показувати з'єднання зі швидкістю 100 Мбіт/с. Коли будь-яке з'єднання встановлено через порт Ethernet, цей світлодіод загоряється лише тоді, коли швидкість з'єднання становить 100 Мбіт/с, і вимикається, коли швидкість підключення становить 10 Мбіт/с.

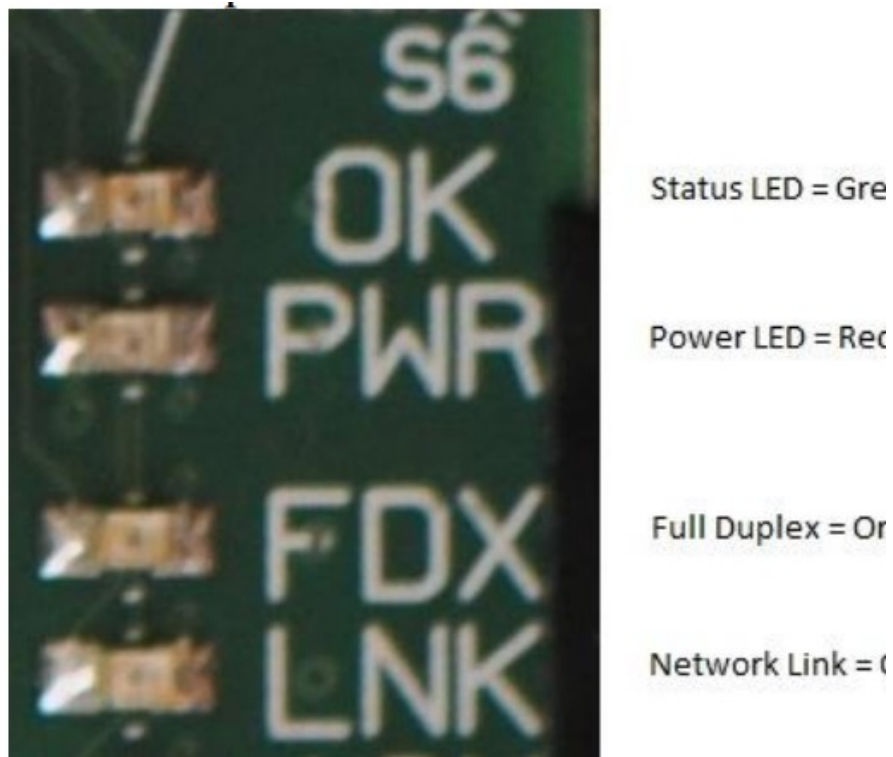


Рисунок 2.7 – Світлодіоди стану

GPIO (General Purpose Input Output): GPIO полегшує підключення будь-яких периферійних пристроїв до Raspberry Pi. Raspberry Pi має вбудований GPIO з 40 контактами, 26 з яких використовуються як цифрові входи або виходи. Що ще важливіше, 9 з 14 нових контактів GPIO є виділеними входами/виходами, це також полегшує вбудовану шину UART, I2C, SPI Bus і все ще існує велика кількість безкоштовних контактів GPIO для додаткових контактів.

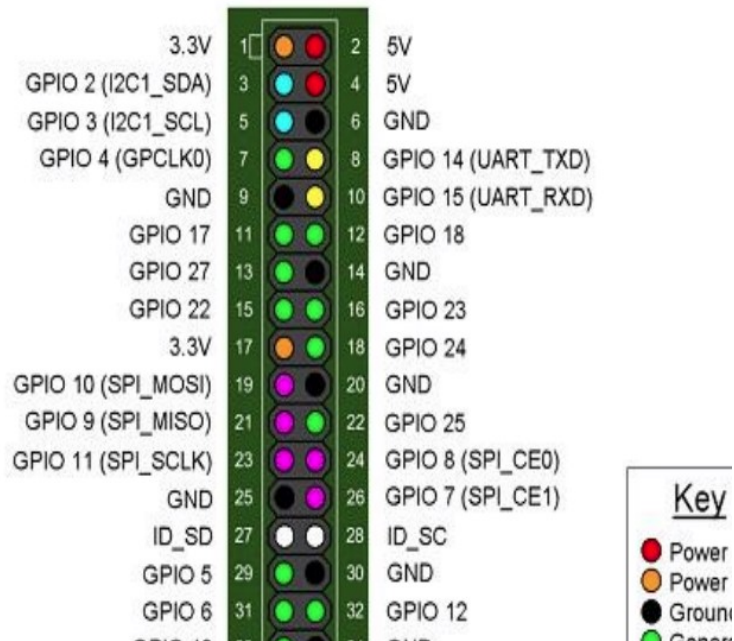


Рисунок 2.8 – General Purpose Input Output

Роз'єм камери CSI: Raspberry Pi має інтерфейс мобільного промислового процесора (MIPI) Camera Serial Interface Type 2 (CSI-2). CSI-2 полегшує підключення невеликої камери до процесора Broadcom BCM 2835. Функція цього інтерфейсу — стандартизувати підключення модулів камери до процесорів для індустрії мобільних телефонів. MIPI CSI-2 версії 1.01 підтримує до 4 смуг передачі даних, і кожна смуга має пропускну здатність 1 Гбіт/с. Специфікація DPHY уточнює інтерфейс фізичного апаратного рівня між камерою та процесором, щоб сприяти швидкому обміну даними.



Рисунок 2.9 – Роз'єм камери CSI

Система на чіпі (SoC): Raspberry Pi (система на чіпі) SoC є ARM на базі Broadcom Technologies. Процесор ARM працює від 700 МГц до 1 ГГц. SoC також забезпечує відеоядро 4 GPU, а також підтримує швидке тривимірне ядро, OpenGL і підтримує відтворення відео H.264 [29].



Рисунок 2.10 – Система на чіпі (SoC)

2.1.2 Переваги та недоліки використання одноплатного комп'ютера Raspberry Pi

Нижче наведено плюси/переваги використання Raspberry Pi:

1. Raspberry Pi вважається невеликим, потужним і ефективним комп'ютером компактного форм-фактора, за порівняно невелику ціну. Різні

підприємства малого та середнього бізнесу можуть використовувати Raspberry Pi для виконання завдань малого та середнього рівня, таких як запуск у якості веб-серверу, серверу баз даних та медіа-серверу.

2. Raspberry Pi може діяти як єдина платформа для широкого досвіду програмування. Pi підтримують різні мови програмування, і користувачі можуть встановити відповідний компілятор для правильного виконання коду. Python, основна мова програмування, яку використовує Pi, менш складна і легка для розуміння. Він забезпечує кращу розробку коду, і забезпечує функцію автоматичного керування пам'яттю.

3. Продукт підтримує операційну систему з відкритим кодом і програми з відкритим кодом. Саме тому, у розпорядженні Raspberry Pi доступні безліч операційних систем у різних варіантах Linux і мільйони програм у цій операційній системі.

4. Raspberry Pi підтримує додаткове обладнання, таке як камера, комплект модулів компонентів, плата Gertboard і NAT, що полегшує підключення тисяч варіантів стороннього обладнання, такого як світлодіоди, і користувач може виконувати різні операції на Pi.

5. Продукт є дуже енергоефективним. Цей невеликий продукт розміром із кредитну картку легко переробляється та заощаджує багато грошей на рішеннях для охолодження [28].

Нижче наведено обмеження/мінуси Raspberry Pi:

1. Він не може діяти як повноцінний комп'ютер, оскільки порт Ethernet і процесор не настільки швидкі для обробки багатозадачних обчислювальних циклів.

2. Несумісний з повнофункціональною операційною системою Windows.

3. Продукт обмежений навіть для малих і середніх підприємств і не дуже корисний для великих організацій/підприємств.

4. Не має годинника реального часу (RTC) із резервним акумулятором. Час можна визначити лише за допомогою NTP-сервера, і більшість операційних систем виконує це завдання автоматично.

5. Не підтримує Bluetooth, Wi-Fi із коробки і навіть різні ключі на основі USB також не підтримуються для бездротового підключення.

6. Не має вбудованого АЦП-перетворювача. Зовнішній зарядний пристрій використовується для АЦП [19].

2.2 PyCharm

PyCharm — це спеціальне інтегроване середовище розробки Python (IDE), що надає широкий спектр важливих інструментів для розробників Python, тісно інтегрованих для створення зручного середовища для продуктивної розробки Python, Інтернету та науки про дані.

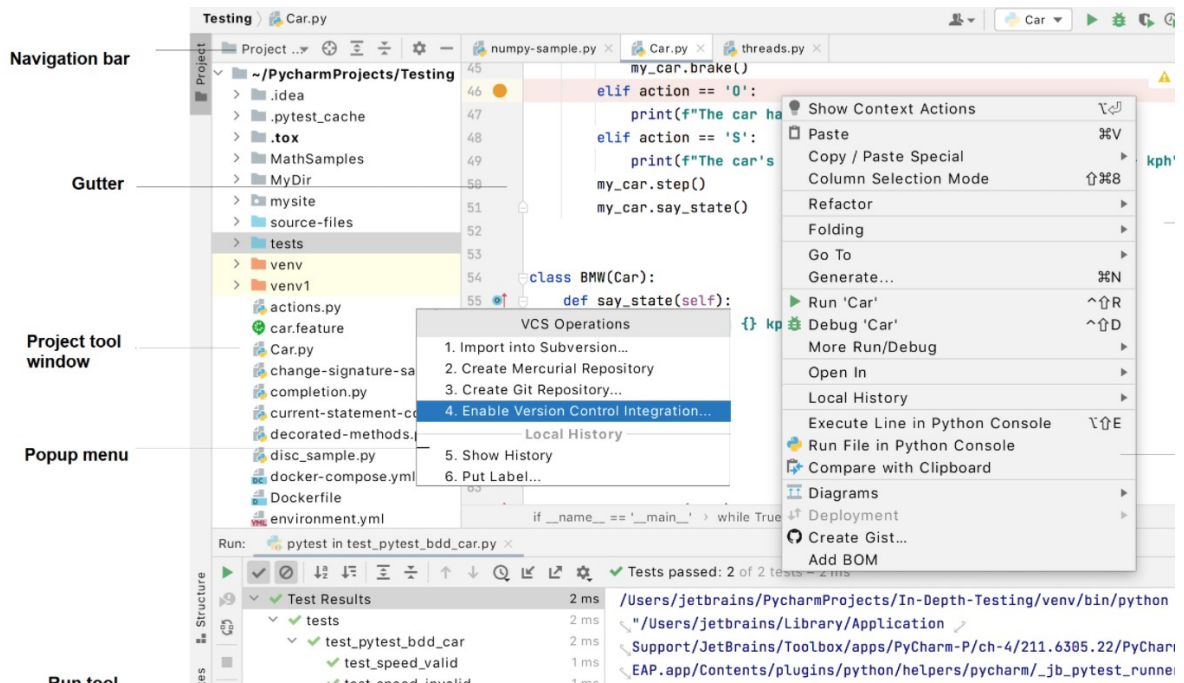


Рисунок 2.11 – Основні UI елементи середовища PyCharm

У вікні інструментів проєкту зліва відображаються файли проєкту.

Редактор з правого боку для безпосередньо написання коду. Він має вкладки для легкої навігації між відкритими файлами.

Панель навігації над редактором додатково дає змогу швидко запускати та налагоджувати програму, а також виконувати основні дії VCS.

Gutter, вертикальна смуга поруч із редактором, показує наявні точки зупинки та забезпечує зручний спосіб навігації по ієрархії коду, як-от перехід до визначення/оголошення. Він також показує номери рядків та історію VCS для кожного рядка.

Смуга прокрутки, на правій стороні редактора. PyCharm постійно контролює якість коду і завжди показує результати перевірок свого коду в стокі: помилки, попередження тощо. Індикатор у верхньому правому куті показує загальний стан перевірок коду для всього файлу.

Вікна інструментів — це спеціалізовані вікна, прикріплені знизу та з боків робочого простору і забезпечують доступ до типових завдань, таких як керування проєктами, пошук вихідного коду та навігація, інтеграція з системами контролю версій тощо.

Рядок стану вказує стан проєкту та всієї IDE, а також показує різні попередження та інформаційні повідомлення, як-от кодування файлів, роздільник рядків, профіль перевірки тощо. Він також забезпечує швидкий доступ до налаштувань інтерпретатора Python.

Крім того, у нижньому лівому куті вікна PyCharm, у рядку стану розташована кнопка Показати панелі вікна інструментів або Сховати вікно інструментів. Ця кнопка перемикає відображення панелей вікна інструментів. Якщо навести вказівник миші на цю кнопку, відобразиться перелік доступних на даний момент вікон інструментів.

2.2.1 Ключові особливості

- Потужний і функціональний редактор коду з підсвічуванням синтаксису, автоматичним форматуванням і автоматичним відступом для підтримуваних мов.
- Проста та потужна навігація по коду.
- Допомога в кодуванні, включаючи автозаповнення, автоматичний імпорт, шаблони коду, перевірку сумісності версії інтерпретатора мови тощо.
- Швидка документація для будь-якого елемента безпосередньо у вікні редактора, перегляд зовнішньої документації через браузер, підтримка docstring – генерування, виділення, автозаповнення та багато іншого.
- Багато оглядів коду.
- Потужний рефакторинг коду, який надає широкі можливості для швидких глобальних змін у проєкті.
- Повна підтримка останніх версій фреймворку Django.
- Підтримка Google App Engine.
- Підтримка IronPython, Jython, Cython, PyPy wxPython, PyQt, PyGTK тощо.
- Підтримка фреймворку Flask, мов Mako та Jinja2.
- Редактор Javascript, Coffescript, HTML/CSS, SASS, LESS, HAML.
- Інтеграція з системами контролю версій (VCS).
- Діаграми класів UML, діаграми моделей Django та Google App Engine.
- Інтегроване модульне тестування.
- Інтерактивні консолі для Python, Django, SSH, відлагоджувача та баз даних.
- Повнофункціональний графічний відлагоджувач (Debugger).

- Підтримка схем для більшості популярних IDE/редакторів. таких як Netbeans, Eclipse, Emacs, емуляція редактора VIM.
- Підтримувані мови: Python (версії: 2.x, 3.x), Jython, Cython, IronPython, PyPy, Javascript, CoffeScript, HTML/CSS, Django/Jinja2 шаблони, Gql, LESS/SASS/SCSS/HAML, Мако, Puppet , RegExp, Rest, SQL, XML, YAML.
- PyCharm має кілька кольірних схем, а також підсвічування синтаксису спеціального коду.
- Інтеграція з трекерами помилок/проблем, такими як JIRA, Youtrack, Lighthouse, Pivotal Tracker, GitHub, Redmine, Trac...
- Величезна, постійно оновлювана колекція плагінів.
- Кросплатформенність (Windows, Mac OS X, Linux) [20].

2.3 Microsoft Visual Studio

Microsoft Visual Studio — це IDE, створена Microsoft і яка використовується для різних типів розробки програмного забезпечення, наприклад комп'ютерних програм, веб-сайтів, веб-програм, веб-сервісів та мобільних додатків. Він містить інструменти завершення, компілятори та інші функції для полегшення процесу розробки програмного забезпечення.

Visual Studio існує вже понад 20 років. Його першою версією була Visual Studio 97. З тих пір було багато різних версій, нинішня – Microsoft Visual Studio 2022.

Visual Studio IDE (інтегроване середовище розробки) — це програмна програма для розробників для написання та редагування свого коду. Його інтерфейс користувача використовується для розробки програмного забезпечення для редагування, налагодження та створення коду. Visual Studio містить редактор коду, який підтримує IntelliSense (компонент завершення

коду), а також рефакторинг коду. Інтегрований налагоджувач працює і як налагоджувач на рівні джерела, і як налагоджувач на рівні машини. Інші вбудовані інструменти включають профайлер коду, конструктор для створення додатків із графічним інтерфейсом користувача, веб-дизайнер, конструктор класів і дизайнер схем баз даних [21].

Переваги Visual Studio:

- Для C++ це одна з найкращих IDE і підтримує багато мов
- Багато розширень
- Має безкоштовну версію
- Використовується багатьма розробниками

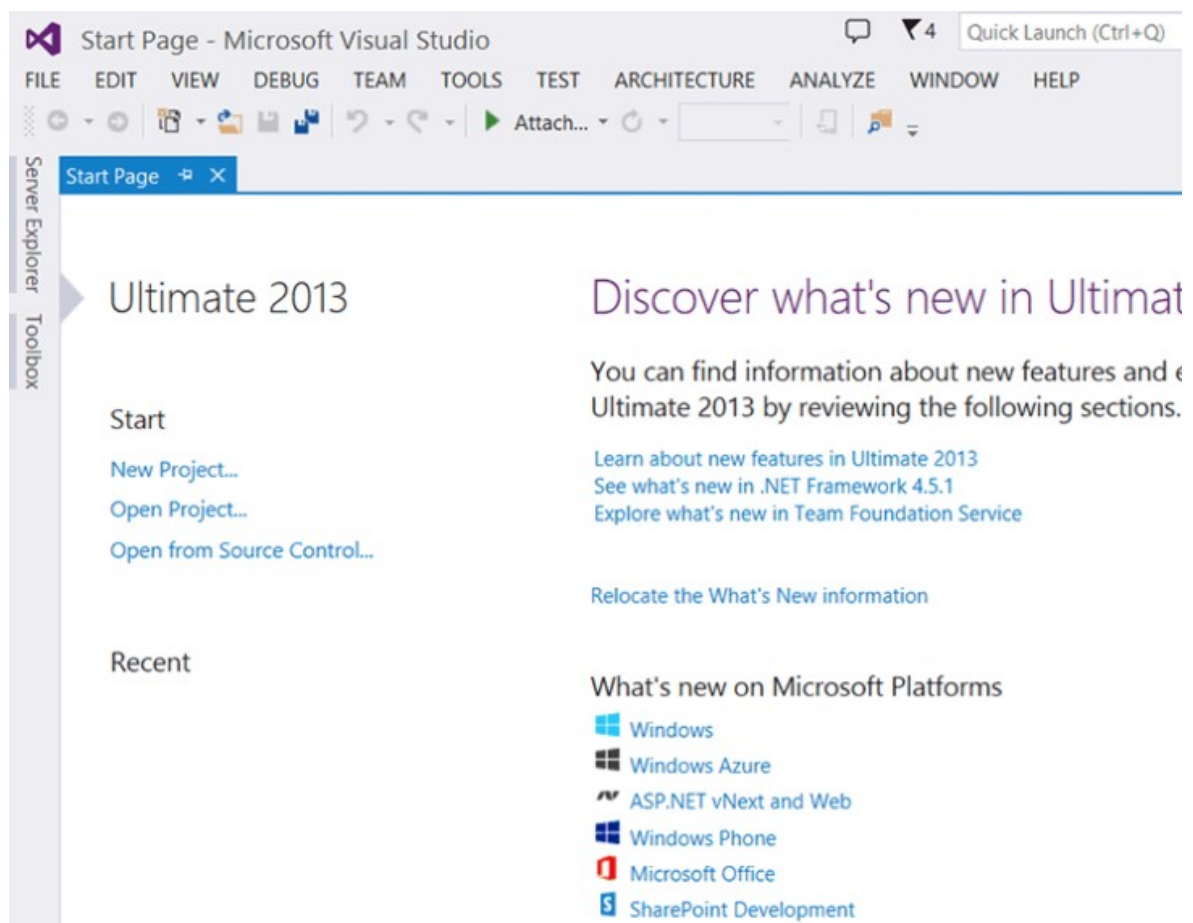


Рисунок 2.12 – Стартова сторінка Microsoft Visual Studio

Створюючи або відкриваючи застосунок або просто окремий файл, система Visual Studio 2013 використовує концепцію рішень (solution) для зв'язування всіх компонентів в єдине ціле. Як правило, рішення складається з одного або кількох проєктів, кожен із яких, у свою чергу, містить безліч елементів, пов'язаних з ним. У минулому такими елементами були звичайні файли, але в даний час всі проєкти створюються з елементів, які утворені з кількох файлів або взагалі не включають в себе ні один файл.

Інструментальне вікно Solution Explorer (Ctrl+Alt+L) забезпечує зручне візуальне представлення рішень, проєктів та елементів, як показано на малюнку нижче. На цьому рисунку представлені три проєкти, що утворюють дерево: WPF-приклад на мові C#, службова WCF-бібліотека та бібліотека класів.

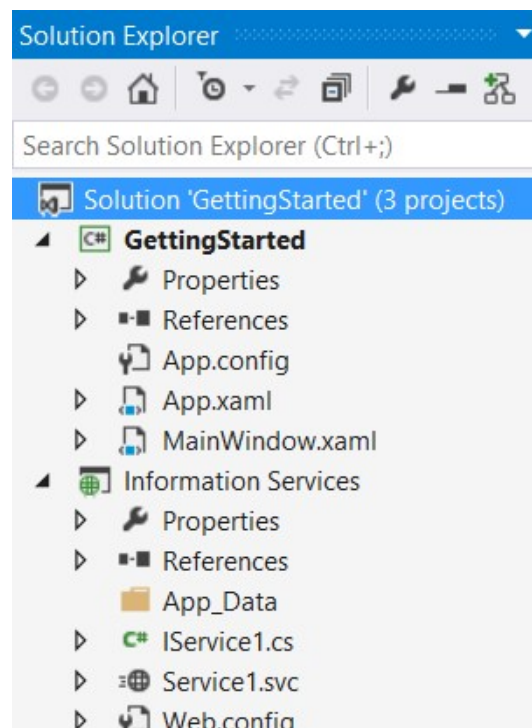


Рисунок 2.13 – Інструментальне вікно Solution Explorer

Проблеми, вирішені в версії 17.0.6:

- Оновлено програму встановлення Visual Studio, щоб правильно записувати метадані каналу.
- Додано інформацію про кінцеву дату до каналу 17.0 LTSC.
- Випущені бюлетені в інсталяторі Visual Studio для встановлення або оновлення.
- Оновлено програму встановлення Visual Studio, щоб покращити роботу віддаленого розгортання на підприємстві.
- Оновлено додавання облікового запису GitHub, щоб підвищити його рівень користування [22].

2.4 Anaconda

Anaconda — це дистрибутив Python і R. Він надає все необхідне для вирішення завдань з аналізу та обробки даних (з використанням Python).

Anaconda — це набір бінарних систем, що включає в себе Scipy, Numpy, Pandas та їх залежності.

Scipy — це пакет статистичного аналізу.

Numpy — це пакет числових розрахунків.

Pandas — рівень абстракції даних для об'єднання та перетворення даних.

Anaconda Navigator — це нативний графічний інтерфейс користувача на робочому столі (GUI), включений у дистрибутив Anaconda, який дозволяє запускати додатки та легко керувати пакетами, засобами та каналами conda без використання командної строки. Навігатор може шукати пакети в Anaconda Cloud або в локальній репозиторії Anaconda. Він доступний для Windows, MacOS та Linux.

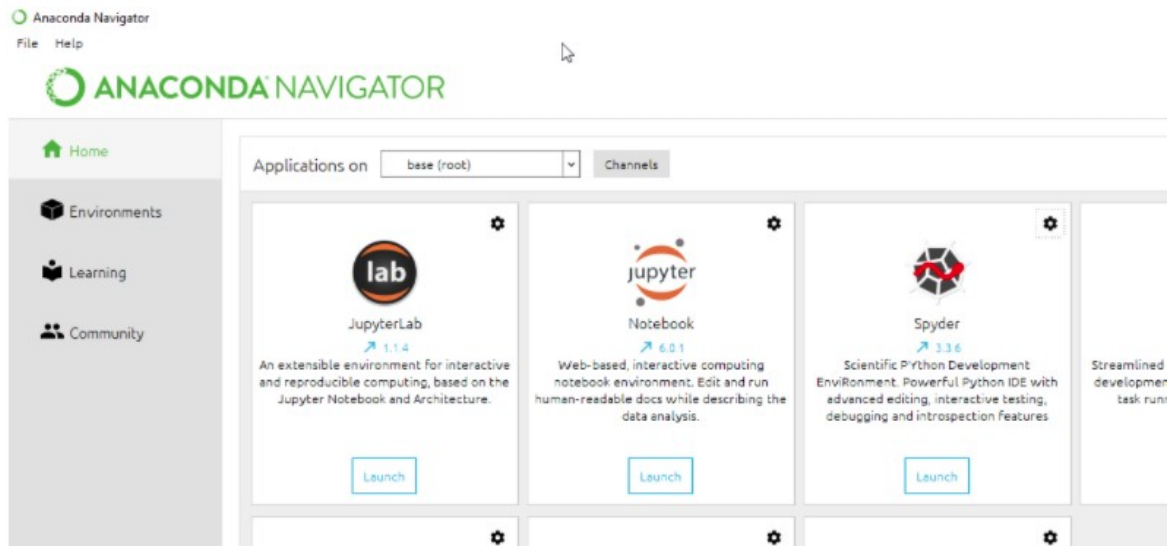


Рисунок 2.14 – Стартова сторінка Anaconda Navigator

JupyterLab — це інтерактивна середа розробки для роботи з блокнотами, кодом і даними.

Jupyter Notebook — зручний інструмент для створення гарних аналітичних звітів, що дозволяє зберігати разом коди, зображення, коментарі, формули та графіки. Робота ведеться у вікні браузера.

Spyder — інтерактивна IDE для навчальних розрахунків на мові Python. Дана IDE дозволяє писати, редагувати та тестувати код. Spyder пропонує перегляд і редагування змінних за допомогою графічного інтерфейсу, динамічну інтроспекцію коду, виявлення помилок у реальному часі та багато іншого. Також, за необхідності, можна інтегрувати Anaconda з іншими Python IDE, включаючи PyCharm і Atom.

VS Code — це оптимізований код редактора з підтримкою таких операцій розробки, як відладка, запуск задач і контрольна версія.

Glueviz — використовується для візуалізації багатьох даних у файлах. Він досліджує відносини всередині і між пов'язаними наборами даних.

Orange 3 — це заснована на компонентах структури інтелектуального аналізу даних. Це може бути використано для візуалізації даних та аналізу

даних. Робочі процеси в Orange 3 дуже інтерактивні та надають великий набір інструментів.

RStudio — це набір інтегрованих інструментів, призначених для підвищення продуктивності роботи з R. Він є основою R і Notebooks.

Для запуску багато навчальних пакетів залежать від конкретних версій інших пакетів. Дослідники даних часто використовують кілька версій множини пакетів і використовують кілька засобів для розділення цих різних версій.

Програма командного рядку conda є одночасно менеджером пакетів і менеджером засобів. Це допомагає спеціалістам за даними гарантування, що версія кожного пакета має всі необхідні умови і працює правильно.

Navigator — це простий і зручний спосіб роботи з пакетами та засобами без необхідності вводити команду conda в вікні терміналу. Ви можете використовувати його, щоб знайти потрібні вам пакети, встановити їх у середу, запустити пакети та оновити їх — все в Navigator.

Виконані файли в середовищі Conda:

python.exe — виконуваний файл Python для доданої командної строки. Так, наприклад, якщо ви перебуваєте в каталозі прикладної програми, ви можете виконати його: `python.exe exampleapp.py`

pythonw.exe — виконуваний файл Python для доданого з графічним інтерфейсом або додатка без інтерфейсу користувача

\Scripts — виконувані файли, які є частиною встановлених пакетів. Після активації середи цей каталог додається в системний шлях, тому виконані файли стають доступними без їх повного шляху.

\Scripts\activate.exe — активує навколишню середу

Управління пакетами, залежностями та засобами для будь-якої мови — Python, R, Ruby, Lua, Scala, Java, JavaScript, C / C ++, FORTRAN та інших.

Conda — це система управління пакетами з відкритим вихідним кодом і системою керування середою, що працює в Windows, macOS та Linux. Conda швидко встановлюється, запускає та оновлює пакети та їх залежності. Conda легко створює, зберігає, завантажує і перемикається між засобами на вашому локальному комп'ютері. Він був створений для програми Python, але він може упаковувати та розширювати програмне забезпечення для будь-якої мови.

Conda як менеджер пакетів допоможе знайти і встановити пакети. Якщо потрібен пакет, для якого потрібна інша версія Python, не потрібно переключатися на інший менеджер засобів, тому що conda також є менеджером засобів. За допомогою всього лише кількох команд є можливість побудувати повністю відокремлену середу для запуску іншої версії Python, продовжуючи при цьому завантажувати звичайну версію Python у звичайному середовищі.

За замовчуванням conda може встановлювати й керувати тисячами пакетів на repo.anaconda.com, які створюються, перевіряються та підтримуються Anaconda.

Conda може бути об'єднана з системами неперервної інтеграції, такими як Travis CI та AppVeyor, щоб забезпечити часте автоматичне тестування програмного коду.

Пакет conda і менеджер середи включені у всі версії Anaconda і Miniconda.

Conda також включена в Anaconda Enterprise, яка забезпечує керування корпоративними пакетами та засобами для Python, R, Node.js, Java та інших стеків додатків. Conda також доступна на [conda-forge](https://conda-forge.org).

Основні команди Conda:

– `conda search package_name` — пошук пакета через conda

- `conda install package_name` — встановлення пакета через `conda`
- `conda install` — встановлення всього стандартного набору пакетів — більше 150, близько 3 Гб
- `conda list` — список встановлених пакетів
- `conda update conda` — оновлення `conda`
- `conda clean -t` — видалення кеша — архівів `.tar.bz2`, які можуть займати багато місця та можуть бути не потрібні.

Завдяки службі Anaconda Repository для IBM Cloud Pak for Data підприємство може керувати пакетами з відкритим кодом, які науковці даних можуть використовувати в ноутбуках Jupyter і JupyterLab в аналітичних проєктах Watson Studio.

За допомогою послуги є можливість отримувати оновлення пакетів Conda в режимі реального часу в міру їх випуску, і отримати доступ до:

- Більше 7500 пакетів Anaconda на Python і R
- Численні пакети з відкритим кодом у Conda-Forge, CRAN і PyPI
- Також можна використовувати Anaconda Repository для IBM Cloud Pak for Data для зберігання пакетів.
- Репозиторій Anaconda для IBM Cloud Pak for Data дає контроль:
- Блокувати, вилучати та додавати пакети відповідно до стандартів підприємства
- Вказувати, які пакунки команда може завантажувати і хто має до них доступ
- Не допускати вразливостей та ненадійного програмного забезпечення у конвєсах науки про дані та машинне навчання
- Керувати залежними пакетами та надавати користувачам швидший доступ до програмного забезпечення з відкритим кодом

Дослідники даних в аналітичних проєктах можуть створювати власні визначення середовища, які включають канали conda та пакети зі сховища, а потім використовувати ці середовища для запуску блоків і сценаріїв Jupyter.

2.5 Python

Python — це потужна мова програмування, яка легко вивчається. Він має ефективні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічне введення Python разом з його інтерпретованою природою роблять його ідеальною мовою для написання сценаріїв і швидкої розробки додатків у багатьох областях на більшості платформ.

Інтерпретатор Python і велика стандартна бібліотека є вільно доступними у вихідній або двійковій формі для всіх основних платформ з веб-сайту Python, і може вільно поширюватися.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих на C або C++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для програм, що налаштовуються.

Хоча існує одна реалізація Python, яка є найпопулярнішою, є кілька альтернативних реалізацій, які представляють особливий інтерес для різної аудиторії.

Відомі реалізації включають:

CPython

Це оригінальна і найбільш підтримувана реалізація Python, написана на C. Нові можливості мови зазвичай з'являються тут першими.

Jython

Python реалізований на Java. Ця реалізація може використовуватися як мова сценаріїв для програм Java, або може використовуватися для створення

додатків за допомогою бібліотек класів Java. Він також часто використовується для створення тестів для бібліотек Java.

Python для .NET

Ця реалізація фактично використовує реалізацію CPython, але є керованою програмою .NET і робить доступними бібліотеки .NET. Його створив Браян Ллойд.

IronPython

Альтернативний Python для .NET. На відміну від Python.NET, це повна реалізація Python, яка генерує IL і компілює код Python безпосередньо в збірки .NET. Його створив Джим Хугунін, оригінальний творець Jython. Для отримання додаткової інформації дивіться веб-сайт IronPython.

PyPy

Реалізація Python повністю написана на Python. Він підтримує кілька розширених функцій, яких немає в інших реалізаціях, як-от підтримка без стека та компілятор Just-in-Time. Однією з цілей проєкту є стимулювання експериментів із самою мовою, полегшуючи модифікацію інтерпретатора (оскільки він написаний на Python).

Інтерфейс інтерпретатора нагадує інтерфейс оболонки UNIX, але передбачає деякі додаткові методи виклику:

При виклику зі стандартним введенням, підключеним до пристрою tty, він запитує команди та виконує їх, доки не буде EOF (символ кінця файлу, який можна створити за допомогою Ctrl-D в UNIX або Ctrl-Z, Enter у Windows) читати.

При виклику з аргументом імені файлу або з файлом як стандартним введенням він читає та виконує сценарій із цього файлу.

При виклику з аргументом імені каталогу він читає та виконує сценарій із відповідною назвою з цього каталогу.

При виклику з командою `-c` він виконує оператор(и) Python, подані як команда. Тут команда може містити кілька операторів, розділених новими рядками.

При виклику з `-m` ім'я-модуля даний модуль знаходиться на шляху до модуля Python і виконується як сценарій.

У неінтерактивному режимі перед виконанням аналізується усі вхідні дані.

`-c <команда>`

Команда може бути одним або кількома операторами, розділеними новими рядками, зі значним пробілом на початку, як у звичайному коді модуля.

Якщо вказано цю опцію, першим елементом `sys.argv` буде «-c», а поточний каталог буде додано до початку `sys.path` (що дозволить імпортувати модулі в цьому каталозі як модулі верхнього рівня).

Викликає подію аудиту `cpython.run_command` з командою аргументу.

`-m <ім'я-модуля>`

Знайти `sys.path` для названого модуля та запустити його вміст як модуль `__main__`.

Оскільки аргумент є ім'ям модуля, не потрібно вказувати розширення файлу (`.py`). Ім'я модуля має бути дійсним абсолютним ім'ям модуля Python, але реалізація не завжди може забезпечити це (наприклад, це може дозволити використовувати ім'я, яке містить дефіс).

Назви пакетів (включаючи пакети простору імен) також дозволені. Коли замість звичайного модуля надається ім'я пакета, інтерпретатор виконає `<pkg>.__main__` як основний модуль. Ця поведінка навмисно схожа на обробку каталогів і zip-файлів, які передаються інтерпретатору як аргумент сценарію.

2.5.1 Ключові терміни

Pip є бажаною програмою встановлення. Починаючи з Python 3.4, він включено за замовчуванням у двійкові інсталятори Python.

Віртуальне середовище — це напівізольоване середовище Python, яке дозволяє встановлювати пакети для використання певною програмою, а не встановлювати їх по всій системі.

Venv є стандартним інструментом для створення віртуальних середовищ і є частиною Python з Python 3.3. Починаючи з Python 3.4, він за замовчуванням встановлює pip у всі створені віртуальні середовища.

Virtualenv є альтернативою (і попередником) третьої сторони venv. Він дозволяє використовувати віртуальні середовища у версіях Python до 3.4, які або не надають venv взагалі, або не можуть автоматично встановлювати pip у створені середовища.

Індекс пакетів Python — це загальнодоступне сховище ліцензованих пакетів з відкритим кодом, доступне для використання іншими користувачами Python.

Python Packaging Authority — це група розробників і авторів документації, відповідальних за підтримку та розвиток стандартних інструментів пакування та відповідних стандартів метаданих і форматів файлів. Вони підтримують різноманітні інструменти, документацію та засоби відстеження проблем як на GitHub, так і на Bitbucket.

Distutils — це оригінальна система збирання та розповсюдження, вперше додана до стандартної бібліотеки Python у 1998 році. Хоча пряме використання distutils поступово припиняється, вона все ще закладає основу для поточної інфраструктури пакування та розповсюдження, і вона не лише залишається частиною стандарту бібліотека, але її назва існує іншими

способами (наприклад, назва списку розсилки, який використовується для координації розробки стандартів пакування Python).

2.6 Flask

Flask — це невеликий і легкий веб-фреймворк, написаний на мові Python, який пропонує корисні інструменти та функції для упорядкування процесу створення веб-додатків із використанням Python. Він забезпечує гнучкість і є більш доступним фреймворком для нових розробників, так як дозволяє швидко створювати веб-додатки, використовуючи лише один файл Python. Flask — це розширювана система, яка не зобов'язує використовувати конкретну структуру директорій і не вимагає складного шаблонного коду перед початком використання.

Flask використовує механізм шаблонів Jinja для динамічного створення HTML-сторінок із використанням знайомих початків у Python, таких як змінні, цикли, списки тощо.

Flask залежить від деяких зовнішніх бібліотек — таких, як Werkzeug і Jinja2.

Werkzeug — це інструментарій для WSGI, стандартного інтерфейсу Python між веб-прикладами та різними серверами. Він призначений як для розробки, так і розвитку.

Інтегровані функції:

- Сервер розробки и відладник
- Інтегрована підтримка модульного тестування
- Відправка запитів RESTful
- Використання шаблонізатора Jinja2
- Підтримка безпечних файлів cookie (сеанси на стороні клієнта)
- 100% відповідність WSGI 1.0

- Unicode
- Обширна документація
- Сумісність з Google App Engine
- Розширення для покращення бажаних функцій

Використання Flask

Спільнотою підтримується серія пакетів-розширень для Flask. Їх назву зазвичай починається з flask-, наприклад, flask-login, flask-sqlalchemy.

Flask є конструктором, тому всі додаткові функції придатні для встановлення мануально. Фреймворк дуже добре підходить для невеликих проєктів, які працюють в основному зі статичним контентом. Також підходить для створення мікросервісів. Із-за відсутності вбудованого ORM дозволяє вибрати відповідний під ваші потреби.

Flask 2.0

З травня 2021 року Flask почав оновлюватися та впроваджувати асинхронний режим роботи. Синтаксис:

```
from flask import Flask
app = Flask(__name__)
@app.get('/')
async def ping():
    return {'message': 'pong'}

if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

Коли варто використовувати Flask:

- Якщо потрібен невеликий бекенд.
- Якщо необхідно зробити власний pet-проєкт.
- Якщо немає навичок роботи з асинхронністю.
- Якщо необхідно отримати більше досвіду та можливостей для

навчання.

– Якщо потрібно більше контролю над використовуваними компонентами.

2.7 C++

C++ — компілована, статично типізована мова програмування загального призначення.

Підтримує такі парадигми програмування як процедурне програмування, об'єктно-орієнтоване програмування, забезпечує узагальнене програмування, модульність, роздільну компіляцію, обробку виключень, абстракцію даних, оголошення типів (класів) об'єктів, віртуальні функції. Стандартна бібліотека включає, в тому числі, загальноживані контейнери та алгоритми. C++ поєднує властивості як високорівневих, так і низькорівневих мов. Порівнюючи з його попередником — мовою C, — найбільшу увагу уділено підтримкою об'єктно-орієнтованого й узагальненого програмування.

C++ широко використовується для розробки програмного забезпечення, являючись одним із самих популярних мов програмування. Область його застосування включає створення операційної системи, різноманітних прикладних програм, драйверів пристроїв, додатків для вбудовуваних систем, високовиробничих серверів, а також різноманітних додатків. Існує безліч реалізованих мов C++, як безкоштовних, так і комерційних і для різних платформ.

Типи даних що використовуються:

- Символьні: `char`, `wchar_t` (`char16_t` та `char32_t`, в стандарті C++11).
- Цілочисельні знакові: `signed char`, `short int`, `int`, `long int` (та `long long int`, в стандарті C++11).

- Цілочисельні беззнакові: unsigned char, unsigned short int, unsigned int, unsigned long int (та unsigned long long int, в стандарті C++11).
- С плаваючою крапкою: float, double, long double.
- Логічний: bool.

Операції порівняння повертають тип bool. Вирази в лапках після if, а приводяться до типу bool.

Функції можуть приймати аргументи по посиланню. Функції можуть повертати результат по посиланню. Посилання схожі з показниками, а також мають наступні особливості: перед використанням посилання має бути ініціалізованим; посилання завжди вказує на одну і ту саму адресу; у вираженні посилання обумовлює безпосередньо цей об'єкт або ту функцію, на яку вона вказує, обертання же до об'єкта або функції через вказівник вимагає розіменування вказівника.

Об'єктно-орієнтовані особливості

C++ додає до C об'єктно-орієнтовані можливості. Він вводить класи, які забезпечують три найважливіших властивості ООП: інкапсуляцію, спадкування та поліморфізм.

Методи класу — це функції, які можна застосовувати до примірників класу. Грубо говорячи, метод — це функція оголошення всередині класу і призначена для роботи з його об'єктами. Методи об'єднання в тілі класу. Описуватися можуть там же, но можуть і за межами класу (всередині класу в такому випадку достатньо представити прототип методу, а за межами класу визначити метод поставив перед його ім'ям — ім'я класу та оператора ::). Методи і поля, що входять у склад класу, називаються членами класу. При цьому методи часто називають функціями-членами класу.

Наслідування

У C++ при спадкоємстві одного класу від другого настає реалізація класу, плюс клас-наслідник може додати свої поля і функції або перевизначити функцію базового класу. Множинне наслідування дозволено.

Конструктор-спадкоємець викликає конструктори базових класів, а потім конструктори нестатичних членів-даних, які є екземплярами класів. Деструктор працює в зворотному порядку.

Наслідування буває публічним, захищеним і закритим.

Поліморфізм

Метою поліморфізму, применшено до об'єктно-орієнтованого програмування, є використання одного імені для завдання загальних для класу діяльності. Виконання кожного конкретного дії буде вирішуватися типом даних.

Перевагою поліморфізму є те, що він допомагає знижувати складність програми, дозволяючи використовувати той самий інтерфейс для завдання єдиного класу дій. Вибір же конкретної дії, в залежності від ситуації, покладається на компілятор. Поліморфізм може застосовуватися також і к операторам.

Інкапсуляція

Основними способами організації інформації в C++ є класи. На відміну від структури мови C, яка може складатися тільки з поля і складених типів, клас C++ може складатися з полів, складених типів і функцій-членів. Інкапсуляція в C++ реалізується через вказівку рівня доступу до членів класу: вони бувають публічними (загальнодоступними), захищеними і закритими (приватними). У C++ структури відрізняються від класів тем, що за замовчанням члени і базові класи у структури публічні, а у класу — власні.

Стандартна бібліотека

У мові програмування C++ термін Стандартна Бібліотека означає колекцію класів і функцій, написаних на базовій мові. Стандартна Бібліотека підтримує кілька основних контейнерів, функції для роботи з цими контейнерами, об'єктами-функціями, основними рядками та потоками (включно інтерактивний та файлове введення-виведення), підтримку деяких мовних особливостей та часто використовуються функції для виконання таких завдань, як, наприклад, обчислення квадратного корню числа. Стандартна Бібліотека мови C++ також включає в себе специфікації стандартів ISO C90 стандартної бібліотеки мови C. Функціональні особливості Стандартної Бібліотеки об'єднуються всередині простору імені `std`.

Стандартна бібліотека шаблонів (STL) — підмножина стандартної бібліотеки C++ і містить контейнери, алгоритми, ітератори, об'єкти-функції та ін.

Заголовні файли стандартної бібліотеки C++ мають розширення «.h».

Стандартна бібліотека C++ містить останні розширення C++ стандарту ANSI (у тому числі бібліотеку стандартних шаблонів і нову бібліотеку `iostream`). Вона являє собою набір файлів заголовків.

Переваги і недоліки

C++ — мова, що формувалася еволюційно. Кожен елемент C++ запозичувався з усіх інших мов окремо та незалежно від інших елементів (нічого із запропонованого C++ для історії його розвитку не було новим у інформатиці), що зробило мову надзвичайно складною, з безліччю дублюючих та взаємосуперечних елементів, блоки яких засновані на різних формальних базах.

Критики C++ не протиставляють йому будь-яку конкретну мову, а навпаки, підтверджують, що для будь-якого випадку застосування C++ завжди існує альтернативний інструментарій, який дозволяє вирішити цю задачу більш ефективно та якісно. У своїй черзі сторони C++ вважають некоректною порівнюючи різні аспекти C++ з різними мовами, так як загальний набір засобів і можливостей C++ істотно ширше, чим у більшості мов, з якими порівнюється, і сама по собі широта можливостей, на їх погляд, є вагомим виправданням недосконалої кожної окремо взятої можливості. Більше того, за їх думками, висока сумісність із C є однією з принципових характеристик мови, і тому всі недоліки C++ виправдовуються перевагами, наданими цією сумісністю.

2.7.1 Переваги та недоліки

Переваги:

- Висока сумісність з мовою C
- Обчислювальна продуктивність
- Підтримка різних стилів програмування: структурне, об'єктно-орієнтоване, узагальнене програмування, функціональне програмування, породжуюче метапрограмування.
- Автоматичний виклик деструкторів об'єктів (в порядку зворотного виклику конструкторів) спрощує і підвищує надійність управління пам'яттю та іншими ресурсами (відкритими файлами, мережевими з'єднаннями, т. п.).
- Перевантаження операторів
- Шаблони (дають можливість побудови зведених контейнерів і алгоритмів для різних типів даних)

- Можливість розширення мови для підтримки парадигм, які не підтримуються компіляторами напряму
- Доступність. Для C++ існує величезна кількість навчальної літератури, переведеної на усі можливі мови

Недоліки:

- Погано продуманий синтаксис звужує спектр застосовності мови
- Мова не містить багатьох важливих можливостей
- Мова містить небезпечні можливості
- Продуктивність праці програмістів на мові виявляється занадто низька
- Громіздкість синтаксису
- Важка наслідуваність
- Необхідність слідкувати за пам'яттю

2.8 NumPy

NumPy це бібліотека Python яка використовується в цілому для роботи з масивами.

Він також має функції для роботи в області лінійної алгебри, перетворення Фур'є та матриць. NumPy був створений в 2005 році Тревісом Оліфантом. Це є Open Source Project і він може використовуватися безкоштовно. NumPy розшифровується як числовий Python.

У Python є списки, які служать меті масивів, але вони повільно обробляються. NumPy має на меті надати об'єкт масиву до 50 разів швидше, ніж традиційні списки Python. Об'єкт масиву в NumPy називається ndarray, він надає безліч допоміжних функцій, які роблять роботу з ndarray дуже легкою.

Масиви дуже часто використовуються в науці про дані, де швидкість і ресурси дуже важливі.

На відміну від списків, масиви NumPy зберігаються в одному безперервному місці в пам'яті, тому процеси можуть дуже ефективно отримувати доступ до них і маніпулювати ними.

Така поведінка в інформатиці називається локальністю посилання.

Це основна причина, чому NumPy швидше, ніж списки. Також він оптимізований для роботи з останніми архітектурами ЦП.

2.9 “libc++” C++ Standard Library

libc++ — це нова реалізація стандартної бібліотеки C++, орієнтована на C++11 і вище.

Особливості та цілі:

- Коректність, як визначено стандартом C++11.
- Швидке виконання.
- Мінімальне використання пам'яті.
- Швидкий час компіляції.
- Сумісність ABI з libstdc++ gcc для деяких низькорівневих функцій, таких як об'єкти винятків, rtti та виділення пам'яті.
- Розширені модульні тести.

Проектування та реалізація:

- Розширені модульні тести
- Внутрішню модель компоувальника можна скидати/читати в текстовий формат
- Додаткові функції підключення можна підключити як «пропуски»
- Специфічний для ОС і процесора код враховано

Після його початкового введення багато людей запитують: «Чому створювати нову бібліотеку замість того, щоб робити внесок у існуючу?» (наприклад, `libstdcxx` Apache, `libstdc++` GNU, `STLport` тощо). Існує багато причин, які сприяють цьому, але деякі з основних:

З багаторічного досвіду (включно з впровадженням стандартної бібліотеки раніше) ми дізналися багато речей про впровадження стандартних контейнерів, які вимагають зламу ABI та фундаментальних змін у тому, як вони реалізуються. Наприклад, загальноновизнано, що створення `std::string` за допомогою «оптимізації короткого рядка» замість використання `Copy On Write` (COW) є кращим підходом для багатоядерних машин (зокрема в C++11, який має посилання на значення `rvalue`). Порушення сумісності ABI зі старими версіями бібліотеки було визначено як критичне для досягнення цілей продуктивності `libc++`.

Mainline `libstdc++` перейшов на GPL3, ліцензію, яку розробники `libc++` не можуть використовувати. `libstdc++ 4.2` (остання версія GPL2) можна було б незалежно розширити для підтримки C++11, але це було б розгалуженням кодової бази (що часто вважається гіршим для проекту, ніж запуск нового незалежного). Інша проблема `libstdc++` полягає в тому, що вона тісно інтегрована з розробкою G++, як правило, досить тісно пов'язана з відповідною версією G++.

`STLport` і бібліотека Apache `libstdcxx` є двома іншими популярними кандидатами, але обидва не мають підтримки C++11. Досвід користувачів (і досвід розробників `libstdc++`) свідчить про те, що додавання підтримки C++11 (зокрема, посилання на `rvalue` та типи, які лише переміщують) вимагає змін майже кожного класу та функції, що по суті становить перепис. Зіткнувшись із переписуванням, ми вирішили почати з нуля та оцінити кожне дизайнерське рішення з перших принципів на основі досвіду. Крім того,

обидва проекти, очевидно, залишені: STLport 5.2.1 був випущений у жовтні 2008 року, а STDCXX 4.2.1 – у травні 2008 року.

Однак libc++ прагне забезпечити високоякісну реалізацію стандартної бібліотеки C++, особливо коли йдеться про коректність. Таким чином, ми прагнемо забезпечити тестове покриття для всіх платформ і компіляторів, які, як ми стверджуємо, підтримуємо. Якщо платформа або компілятор не вказано тут, вони офіційно не підтримуються. Можливо, це спрацює, і на практиці відомо, що бібліотека працює на деяких платформах, не перелічених тут, але ми не даємо жодних гарантій. Якщо ви хочете, щоб ваш компілятор та/або платформа були офіційно підтримані та вказані тут, будь ласка, співпрацюйте з командою libc++, щоб налаштувати тестування для вашої конфігурації.

Висновки до розділу 2

Було проведено аналіз існуючих технологій, та для реалізації апаратно-програмного модулю для реалізації віддаленого навчання на базі одноплатного комп'ютера було обрано саме такий стек технологій через їх найвищу сумісність та відповідність усім вимогам встановленим для апаратно-програмного модулю для реалізації віддаленого навчання на базі одноплатного комп'ютера. Перелічені технології було використано у наступному розділі кваліфікаційної магістерської роботи, практичній реалізації апаратно-програмного модулю.

РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ

3.1 Logitech HD Pro Webcam C920

Logitech HD Pro Webcam C920 здатна записувати відео в режимі 1080p з частотою 30 к/сек та використанням популярного формату H.264, тоді як C910 була обмежена 15 кадрами в секунду і не найкращим кодеком Windows Media Video.

Як обіцяє виробник, завдяки технології Logitech Fluid Crystal, на додаток до плавного відеоряду, без дратівливих ривків, користувачі отримують більш чітку картинку, насичені кольори та чистий звук. Якщо порівнювати Logitech HD Pro Webcam C920 з рядовими веб-камерами та моделями вбудованими в мобільні пристрої, то чіткість картинки, кольору та звук записані камерою Logitech справді вражають. Але і тут розробникам є ще куди прагнути. Крім відео, Logitech C920 вміє робити і фотографії з інтерполірованою роздільною здатністю 15 Мп.

Декілька слів про комплектне ПЗ. Logitech Webcam Software дозволяє записувати відео та фото максимально можливої якості на жорсткий диск. Налаштування дуже гнучкі і дозволяють змінювати безліч параметрів, які знайомі людям, що захоплюються обробкою фотографій і відео – задається значення експозиції, посилення, яскравості, контрасту, насиченості кольорів та балансу білого. За допомогою таких важелів управління можна досягти набагато цікавіших результатів, ніж зі стандартними налаштуваннями. Звичайно, в налаштуваннях можна налаштувати рівень стиснення звуку і відео, а також рівень сигналу мікрофона.

Ще одна цікава особливість програми – функція стеження за особою, яка працює дуже ефективно. Вона доступна тільки в режимі 480p і при цьому

сама камера, природно, не обертається, використовується цифровий зум та запас роздільної здатності.

За допомогою режиму Motion Detection можна стежити за вашим ПК, коли ви відходите від нього на якийсь час. Потрібно лише задати ступінь чутливості сенсора, час запису після того, як веб-камера помітила рух і натиснути кнопку запису. Як тільки в полі зору об'єктива з'явиться об'єкт, що рухається, почнеться запис і ви зможете дізнатися, хто саме підходив до вашого робочого місця. Працює детектор тільки в роздільній здатності 800x600 пікселів, зате дуже ефективно і з функцією стеження за обличчям.

Програма Logitech Vid HD покликана поєднати власників веб-камер цього бренду під спільний прапор. Вона безкоштовна, і дозволяє швидко та зручно виконувати відеодзвінки у форматі 720.

Якщо говорити про різницю між C920 та 910 (попередня модель) у повній роздільній здатності – вона помітна неозброєним оком. Дворазове збільшення частоти дається взнаки. При цьому в нижчій роздільній здатності відмінності вже не такі істотні – в режимі 720р обидві моделі видають 30 кадрів на секунду. Адже саме в цьому режимі швидше за все і будуть використовуватися камери [30].



Рисунок 3.1 – Веб камера Logitech HD Pro Webcam C920



Рисунок 3.2 – Веб камера Logitech HD Pro Webcam C920 (фронтальний
вигляд)

Плюси та мінуси обраної моделі:

Плюси:

- підтримка режиму 1080p із частотою 30 к/сек у Skype та при записі на жорсткий диск;
- використання кодека H.264;
- якість мікрофона;
- універсальне кріплення під будь-які типи дисплеїв;
- функція стеження за особою;
- автофокусування;
- макрорежим.

Мінуси:

- досить висока вартість;
- глянсовий верх;
- великі габарити.

Таблиця 2 – Технічні характеристики веб-камери Logitech HD Pro Webcam C920

Роздільна здатність відео / Частота кадрів (кадрів/с)	1920x1080/30, 1280x720/30
Формати відео	H.264
Мікрофон	стерео
Сенсор	CMOS
Фокусування	авто
Фото	15 Мп

3.2 Перелік основних аналогів використаного одноплатного комп'ютера

На даний момент Raspberry Pi найпопулярніший одноплатний комп'ютер. Його можна використовувати для невеликих саморобних

пристроїв та особистих проектів, або як дешеву систему для навчання програмування, або навіть зібрати домашній медіа-сервер.

З Raspberry Pi можна багато чого зробити, але це не абсолютне рішення для будь-яких завдань. Для чогось можна використовувати більш дешеву плату, а для чогось потрібна потужніша. Саме тому у даному підрозділі магістерської кваліфікаційної роботи було розглянуто основні аналоги одноплатного комп'ютера Raspberry Pi.

ONION OMEGA2+

Omega2+ це найдешевший, який тільки можна знайти, одноплатний комп'ютер для роботи у сфері IoT (internet of things – інтернет речей). Він працює на LEDE (Linux Embedded Development Environment) – дистрибутиві Linux на основі OpenWRT.

Його форм-фактор, ціна, і гнучкість, що з роботи на Linux, роблять його ідеальним варіантом для практично будь-якого застосування у сфері інтернету речей.

Характеристики:

- MT7688 SoC.
- 4 GHz IEEE 802.11 b/g/n WiFi.
- 128 MB DDR2 RAM.
- 32 MB вбудованої флеш-пам'яті.
- MicroSD.
- USB 2.0
- 12 GPIO pins.

NVIDIA JETSON NANO DEVELOPER KIT

Якщо є необхідність у технологіях штучного інтелекту та глибокому навчанні, то саме цей варіант може бути обрано, іта саме він може ідеально виконати усі вимоги.

Характеристики:

- Процесор: чотириядерний ARM A57 @ 1.43 ГГц.
- Графічний процесор: 128-core Maxwell.
- Оперативна пам'ять: 4 Гб 64-бітний LPDDR4 25.6 Гбіт/с.
- Відео вихід: HDMI 2.0
- 4 x USB 3.0 та eDP 1.4.

ASUS TINKER BOARD S

ASUS Tinker Board S можливо не найдоступніша альтернатива Raspberry Pi за її \$82, але безумовно досить потужна. Вона включає в себе той же 40-pin конектор, що і Raspberry Pi 3, але пропонує більш потужний процесор і відеокарту. Будучи при цьому того ж розміру, що Raspberry Pi 3.

Основна особливість цієї плати це наявність 16 GB eMMC (простою мовою, щось на зразок маленького вбудованого SSD, що збільшує швидкість роботи).

Характеристики:

- Чотирьохядерний процесор Rockchip RK3288.
- 2GB оперативної пам'яті DDR3.
- Вбудований графічний процесор.
- ARM® Mali™-T764 GPU.
- 16 GB eMMC.
- Слот для карт MicroSD.
- 11 b/g/n, Bluetooth V4.0+EDR.
- USB 2.0
- 28 GPIO pins.
- HDMI порт.

ARDUINO MEGA 2560

Якщо є необхідність використати одноплатний комп'ютер у робототехніці або потрібна плата для 3D принтера, Arduino Mega 2560 буде непоганою заміною Raspberry Pi. На відміну від Raspberry Pi, цей одноплатник заснований на мікроконтролері, замість мікропроцесора.

Характеристики:

- Мікроконтролер: ATmega2560.
- Тактова частота: 16 МГц.
- Цифрові I/O pins: 54.
- Аналогові input pins: 16.
- Флеш пам'ять: 256 КВ, 8 КВ із якої відведено під завантажувач.

3.3 Типи (архітектури) систем відеоконференцій

Щоб згладити технічні обмеження терміналів, обмін даними завжди здійснюється через якогось медіуму — сервер ВКС.

Очевидно, ефективність такої системи залежить від:

- способу представлення даних при передачі між терміналами та сервером;
- поділу етапів обробки даних між терміналами та сервером.

Раніше було прийнято ділити їх на програмні та апаратні, але приблизно з 2014 це стало не актуально, оскільки з'явилися і програмні, і апаратні рішення з нетиповою для таких архітектурою. Крім того, всі провідні виробники намагаються перекласти свою ВКС інфраструктуру на віртуалізовані середовища, щоб поставляти як програмне забезпечення.

Таблиця 3 – Різниця між основними архітектурами

	MCU	SFU	Simulcast	SVC
Розкладка формується на боці терміналу	Ні	Так	Так	Так

Якість відео в реальному часі адаптується під можливості терміналу	Так	Ні	Так	Так
Ефективне використання каналів для сформованої розкладки	Так	Ні	Ні	Так

Архітектура відеоконференцій на основі мікшування (MCU)

Вся обробка даних відбувається на стороні сервера. Після збору вихідних відеопотоку з усіх терміналів сервер окремо для кожного терміналу:

- Склеює зменшені зображення учасників в єдиний потоковий канал.
- Стискає відеопотік до якості, що відповідає смузі пропускання до терміналу в даний момент часу.
- Відправляє відеопотік терміналу.

Така архітектура називається мікшуючою або MCU (від англ. Multipoint Control Unit). Системи на основі MCU вимагають великої обчислювальної потужності та погано масштабуються, навіть з урахуванням можливої віртуалізації. До того ж, вартість підключення нового абонента в подібній інфраструктурах вкрай висока.

Архітектура відеоконференцій на основі мультиплексування (SFU, Switching)

Це класичний підхід до побудови програмних систем ВКС, за таким принципом, наприклад, працює Skype. На відміну від MCU, сервер ВКС у разі не обтяжує себе перекодуванням, створює копії вхідних потоків і пересилає їх іншим учасникам «як є». Виходить, що кожен із терміналів отримує відразу кілька відеопотоків у повній якості, які він просто не може відобразити одночасно. Терміналу доводиться зменшувати розмір кожного

відео від кожного з учасників на своїй стороні, або просити зменшувати його перед відправкою, що погіршує якість відео для всіх інших учасників.

Плюс у цієї схеми один: інфраструктура невибаглива до ресурсів і навіть рядовий ПК може витримати сотні таких конференцій одночасно. Але мінусів значно більше: терміналу (зазвичай це простий ПК) доводиться декодувати не один, а відразу кілька потоків, а серверу ВКС потрібно в кілька разів більша вихідна ширина каналу, щоб вмістити в себе всі створені ним копії потоків.

Архітектура відеоконференцій на основі паралельної передачі (Simulcast)

Simulcast поєднує в собі переваги MCU та SFU, частково адаптуючи мережеве навантаження під можливості терміналів, але не обтяжуючи сервер обробкою відео.

У цій архітектурі сервер отримує від кожного терміналу не один, а кілька відеопотоків, що копіюють зображення камери в різній роздільній здатності та якості для різних смуг пропускання. Далі, як і в SFU, кожен термінал отримує набір окремих відеопотоків учасників, але якість знижується вже без додаткового перекодування, а за рахунок вибору потрібної копії.

Але навантаження, пов'язане з одночасною підтримкою кількох рівнів якості відео, від цього нікуди не зникає – воно переміщається на початок процесу і лягає на ВКС-термінал користувача.

На практиці кодування терміналом трьох вихідних відеопотоків різної якості вимагає багато ресурсів, а за логікою речей воно і надмірно – адже інформація (зображення) у всіх потоках передається одна і та ж, просто з різним ступенем деталізації.

Архітектура відеоконференцій на основі масштабованого відеокодування (SVC)

Дана архітектура поєднує у собі всі переваги мікшуючого підходу і при цьому позбавлена недоліків систем на основі мультиплексування. Вона дешева, миттєво масштабується та працює на будь-яких платформах. Це стало можливим завдяки розвитку технологій обробки сигналів та стиснення даних.

Суть полягає в тому, що термінал стискає свій відеопотік шарами: кожен додатковий шар підвищує роздільну здатність відео, його якість і кількість кадрів в секунду. Якщо канал між терміналом і сервером ВКС хороший, то термінал відправляє максимально можливу кількість шарів. Варто зауважити, що шар – це не окремий відеопотік меншої якості, а повноцінна різниця між ним та попереднім шаром. Тим самим, SVC потік всього на 15-20% відрізняється по ширині каналу від SVC-потіку, і значно менше необхідної суми смуги пропускання незалежних потоків.

Сервер ВКС, отримавши SVC-потік з шарами, легко відсікає зайві без перекодування, тільки за рахунок викидання з нього пакетів з даними за певними правилами. Тим самим дозволяючи на льоту створювати індивідуальні набори відеопотоків для кожного з учасників групової відеоконференції залежно від його реальних умов зв'язку.

3.4 Принцип роботи апаратно-програмного модулю

Апаратно-програмний модуль для організації процесу віддаленого навчання працює у режимі, у якому всі учасники (учні) бачать і чують лише одного мовника (викладача), а він бачить і чує всіх учнів. Завдяки саме такій організації процесу віддаленого навчання, учні не відволікаються, а викладач їх контролює.

Описуючи принцип роботи апаратно-програмного модулю, можна зазначити дві основні сторони-учасники створеної відео-конференції:

- Відправник (вчитель)
- Отримувач (учень)

Відправник (вчитель) за допомогою веб-камери, що оснащена вбудованим мікрофоном, збирає аудіо та відео потоки. Одноплатний комп'ютер виступає у ролі серверу, що приймає та відправляє аудіо та відео потоки користувачам системи. У нього йде канал зі сторони відправника сигналу, та з нього виходить канал, що розповсюджується на всіх учасників процесу віддаленого навчання (студентів).

Схематично це можна зобразити наступним чином:

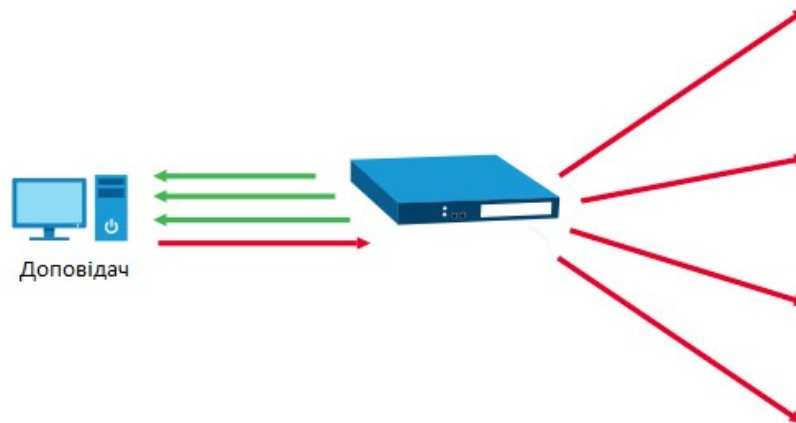


Рисунок 3.3 – Схематичне зображення роботи апаратно-програмного модулю

Реалізація апаратно-програмного модулю була здійснена на мові програмування Python. Нижче наведено короткий опис кожного з файлів, скріншот візуальної частини застосунку, а також частина лістингу коду.

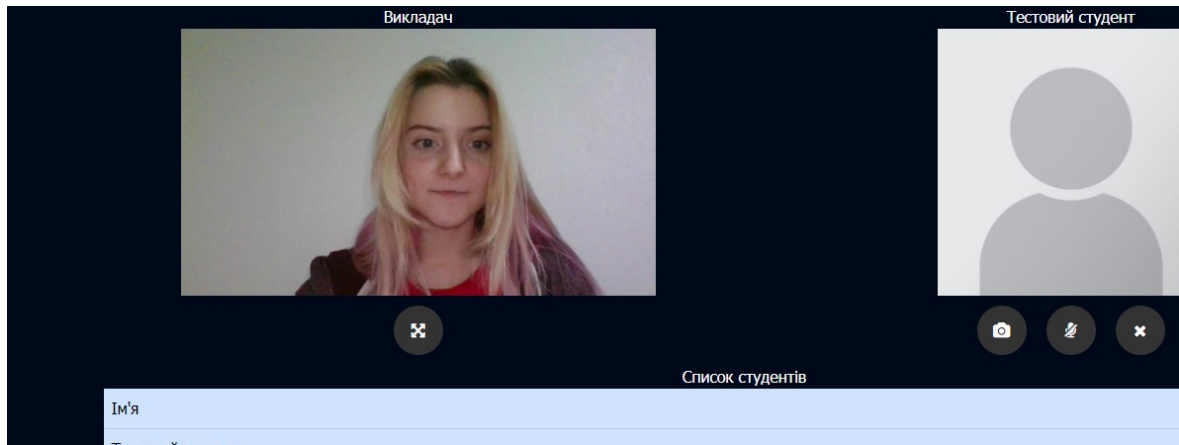


Рисунок 3.4 – Програмний інтерфейс підключення до тестового каналу
Файл setup.py, у якому виконується підключення та встановлення усіх
необхідних пакетів.

```
import numpy
from Cython.Distutils import build_ext
include_dir = '/usr/local/include'
lib_dir = '/usr/local/lib/'

def configuration(parent_package="", top_path=None):
    """ Function used to build our configuration.
    """
    from numpy.distutils.misc_util import Configuration

    # The configuration object that hold information on all the files
    # to be built.
    config = Configuration("", parent_package, top_path)
    config.add_extension('video_streaming',
                        sources=['video_streaming.pyx', './src/VideoEncoder.cpp',
                                './src/VideoDecoder.cpp'],
                        libraries=['avcodec', 'avutil', 'avformat', 'swresample',
                                'swscale', 'z', 'lzma', 'pthread'],
```

```
        depends=['./src/VideoEncoder.cpp',  
                './src/VideoEncoder.h', './src/VideoDecoder.cpp', './src/VideoDecoder.h'],  
        language = "c++",  
        extra_compile_args=['-std=c++11'],  
        include_dirs=[numpy.get_include(), include_dir],  
        library_dirs= [lib_dir])  
  
    return config
```

Повний лістинг коду наведено у Додатку А.

Частина файлу `video_streaming.py`, у якому відбувається декодування відео та аудіо файлів з основного потоку.

```
cdef VideoEncoder * thisptr  
    def __cinit__(self, int width, int height, int bitrate, string codec_name,  
int fps, int gop_size, int max_b_frames):  
        """"  
        Parameters  
        -----  
        width : int  
            Frame width  
        height : int  
            Frame height  
        bitrate : int  
            Video bitrate. The lower is the bitrate, the higher is compression.  
        codec_name: str  
            The name of the codec to use. Note that the encoder and the  
decoder must use the exact same codec. Tested with libx264 and mpeg1video.  
        fps: int  
            Frames per second  
        gop_size: int  
            Group of pictures size. Please see  
https://en.wikipedia.org/wiki/Group\_of\_pictures  
        max_b_frames: int
```

Maximum quantity of consecutive B frames. Please see
https://en.wikipedia.org/wiki/Video_compression_picture_types

```
"""
    self.thisptr = new VideoEncoder(width, height, bitrate, 0,
    codec_name, fps, gop_size, max_b_frames)
    def __dealloc__(self):
        del self.thisptr
    def encodeFrame(self, frame):
        """
        frame: np.array
            RGB frame
        bytes
            a byte message
        """
        cdef np.ndarray[np.uint8_t, ndim=3, mode = 'c'] np_buff =
np.ascontiguousarray(frame, dtype = np.uint8)
        cdef uint8_t * im_buff = <uint8_t *> &np_buff[0, 0, 0]
        cdef vector[uint8_t] frame_bytes
        frame_bytes = self.thisptr.encodeFrame(im_buff, frame.strides[0],
frame.strides[1], frame.strides[2])
        return bytes(frame_bytes)
```

Повний лістинг коду наведено у Додатку Б.

Частина файлу sender.py, у якому відбувається процес відправлення аудіо та відео пакетів у канал.

```
class FrameSender:
    def __init__(self, width, height, bitrate=1000000,
    codec_name="libx264", fps=25, gop_size=10, max_b_frames=1):
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.frame_queue = Queue()
        self.image_sending_thread =
threading.Thread(target=FrameSender._encode_send_frame, args=(self,))
```

```
        self.encoder = PyVideoEncoder(width, height, bitrate,
        codec_name.encode('utf-8'), fps, gop_size, max_b_frames)
    def connect(self, ip, port):
        self.socket.connect((ip, port))
        self.image_sending_thread.start()
    def _encode_send_frame(self):
        while True:
            frame, pts = self.frame_queue.get()
            frame_bytes = self.encoder.encodeFrame(frame)
            self.socket.sendall(frame_bytes)
    def send_frame(self, frame, pts):
        self.frame_queue.put((frame, pts))
def parse_args():
    parser = argparse.ArgumentParser()
    parser.add_argument('--ip', type=str, default="0.0.0.0")
    parser.add_argument('--port', type=int, default=17098)
    parser.add_argument('--frame-width', type=int, default=640)
    parser.add_argument('--frame-height', type=int, default=480)
    parser.add_argument('--bitrate', type=int, default=10000000)
    parser.add_argument('--codec', type=str, default="libx264")
    args = parser.parse_args()
    return args
```

Повний лістинг коду наведено у Додатку В.

Частина файлу receiver.py, у якому відбувається процес отримання аудіо та відео пакетів з каналу.

```
class FrameReceiver:
    def __init__(self, port, width, height, codec="h264"):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.bind(('0.0.0.0', port))
        self.socket = None
```

```
self.frame_queue = Queue()
self.decoder = PyVideoDecoder(width, height, codec.encode('utf-8'))
self.image_receiving_thread =
threading.Thread(target=FrameReceiver._receive_images, args=(self,))
self.image_decoding_thread =
threading.Thread(target=FrameReceiver._decode_images, args=(self,))
def _decode_images(self):
    while True:
        frame, num = self.decoder.getFrame()
        self.frame_queue.put_nowait((num, frame))
def _receive_images(self):
    while True:
        bytes_chunk = self.socket.recv(BUFSIZE)
        self.decoder.supplyBytes(bytes_chunk)
def run(self):
    print('awaiting connection')
    self.sock.listen(5)
    self.socket, address = self.sock.accept()
    print('connection established')
    self.image_receiving_thread.start()
    self.image_decoding_thread.start()
    print("receiver is running")
    while True:
        pts, frame = self.frame_queue.get()
        cv2.imshow('frame',frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

Повний лістинг коду наведено у Додатку Г.

Висновки до розділу 3

У даному розділі було оглянуто використану веб-камеру для забезпечення якісного та чіткого відео сигналу Logitech HD Pro Webcam 2022 р.

C920. Вона здатна записувати відео в режимі 1080p з частотою 30 кадр/сек та використанням популярного формату H.264, тоді як C910 була обмежена 15 кадрами в секунду і не найкращим кодеком Windows Media Video. Було обрано саме цей варіант через його надійність та якість корпусу та високі показники основних характеристик. У деяких випадках кращим варіантом буде обрати варіант за більш низькою ціною, адже вартість обраної моделі є основним її недоліком.

Також було розглянуто основні технології організації відеоконференцій, а саме:

- MCU
- SFU
- Simulcast
- SVC

Було розглянуто їх сильні та слабкі сторони, наведено опис кожної з технологій. Було проаналізовано увесь перелік та обрано найкращий з наведених варіантів для організації принципу роботи апаратно-програмного модулю.

Було описано принцип роботи апаратно-програмного модулю та наведено лістинги коду, з повним переліком яких можна ознайомитися у розділі Додатки. Також було наведено скріншот користувацької частини застосунку, що демонструє тестове з'єднання.

ВИСНОВКИ

Дистанційне навчання – це спосіб навчання на відстані, при якому викладач та учні фізично знаходяться в різних місцях. При такому виді навчання у людей, які обтяжені сімейними та діловими турботами та не мають можливості відвідувати традиційні заняття, з'являється шанс отримати якісні послуги з навчання. Дистанційне навчання відповідає вимогам сучасного життя, особливо якщо врахувати не тільки транспортні витрати, але й витрати на організацію всієї системи очного навчання. Звідси все більший інтерес до дистанційного навчання, до його різних форм, необхідних протягом усього життя людини.

Практичне значення аналіз актуальних навчальних програм провідних вузів у галузі комп'ютерної інженерії показує, що вони передбачають роботу з одноплатними комп'ютерами як один із варіантів апаратних платформ. SBC забезпечують необхідний баланс зручної програмної інфраструктури та низькорівневого доступу до апаратних інтерфейсів. Більш того, оскільки вони використовують операційні системи на основі Linux переважно як основний програмний рівень, перехід від розробки програмного забезпечення для настільних комп'ютерів до розробки програмного забезпечення для периферійних пристроїв повинен бути плавним і плавним.

Під час виконання магістерської роботи було використано наступні технології та бібліотеки:

Використані технології:

- Raspberry PI (Orange Pi)
- Python
- Anaconda
- Flask

- C++

Бібліотеки:

- NumPy
- CPython
- libc++ C++ Standard Library

IDE:

- PyCharm
- VisualStudio

У ході виконання роботи було розроблено апаратно-програмний модуль для організації дистанційного навчання на базі одноплатного комп'ютеру. Завдяки тому, що у Чорноморському національному університеті ім. Петра Могили (Миколаїв, Україна) є дві бездротові мережі, що відповідають обом описаним випадкам, з'єднання було підготовлено для обох мереж.

Для досягнення поставленої мети було вирішено такі завдання:

- виконано аналіз літератури та патентної інформації з аналогів пристроїв, які можуть забезпечити стабільне та захищене з'єднання для забезпечення процесу дистанційного навчання;
- змодельовано та спроектовано прототип під керуванням комп'ютеру Raspberry Pi та плати Arduino з Ethernet-екранами, інтегрованими в локальну мережу;
- виконано тестування роботи створеної бета-версії застосунку;
- розроблено питання з охорони праці та безпеки життєдіяльності.

Результат виконання кваліфікаційної магістерської роботи є повністю функціонуюча система, що може бути впроваджена у навчальний процес Чорноморського національного університету імені Петра Могили.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Сагиндикова, А. С. Актуальність дистанційного навчання / А. С. Сагиндикова, М. А. Тугамбекова. — 2015. — № 20 (100). — С. 495-498. — URL: <https://moluch.ru/archive/100/20703/> (дата обращения: 22.12.2021).
2. Толстобоков, Олег Николаевич Т 529 Сучасні методи та технології дистанційного навчання. Монографія – М.: Мир науки, 2020. – Онлайн ресурс. Режим доступу: <https://izdmn.com/PDF/37MNNPM20.pdf>
3. Андреев А.А., Солдаткин В.И. Дистанційне навчання: суть, технологія, організація. – М.: Видавництво МЭСИ, 2016. – 196 с.
4. Зайченко Т.П. Основи дистанційного навчання: теоретико-практичний базис: навчальний посібник. – СПб.: Изд-во РГПУ им. А.И. Герцена, 2014. – 167 с.
5. Занковский А.Н. Організаційна психологія. – М.: Флинта: МПСИ, 2002.
6. Иванченко Д.А. Системний аналіз дистанційного навчання: монографія. – М.: Союз, 2015. – 192 с.
7. Подласий И.П. Педагогика. Новый курс: підручник для студ. пед. вузів: в 2 кн. – М.: Владос-пресс, 2016. – Кн. 1: Загальні основи. Процес навчання. – 576 с.
8. Полат Е.С. Дистанційне навчання: навчальний посібник / Е.С. Полат, М.В. Моисеева, А.Е. Петров, М.Ю. Бухаркина, Ю.В. Аксенов, Т.Ф. Горбунькова. – М.: Владос, 1998. – 192 с.
9. Теорія та практика дистанційного навчання: навчальний посібник для студ. вищ. пед. навч. закладів / Е.С. Полат, М.Ю. Бухаркина, М.В. Моисеева; під ред. Е.С. Полат // М.: Видавничий центр «Академія», 2014. – 416 с.

10. Cantor J. A. (1992). Delivering Instruction to Adult Learners. Wall & Emerson, Toronto.

11. Crotty P. T. and Soule A. J. (1997). Executive education: yesterday and today, with a look at tomorrow. Journal of Management Development, Vol. 16 (1), pp. 4–21.

12. Goodwin, J. and Fulmer, R.M. (1995), «The systems dynamics of executive education», Executive Development, Vol. 8, No. 4, pp. 9–14.

13. Effect of covid-19 on school education system: challenges and opportunities to adopt online teaching and learning URL: https://www.researchgate.net/publication/347835298_effect_of_covid-19_on_school_education_system_challenges_and_opportunities_to_adopt_online_teaching_and_learning (дата звернення: 22.12.2021).

14. 4th IFAC Workshop on Advanced Maintenance Engineering, Services and Technologies - AMEST 2020: Cambridge, UK, 10–11 September 2020

15. 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA

16. Remote and Virtual Labs for Engineering Education 4.0: Achievements of the ELLI project at the TU Dortmund University URL: <https://www.sciencedirect.com/science/article/pii/S2351978918308023> (дата звернення: 22.12.2021).

17. International Educational Technology Conference, IETC 2014, 3-5 September 2014, Chicago, IL, USA

18. Remote Access for Education and Control of Mechatronics Systems URL: <https://www.sciencedirect.com/science/article/pii/S187704281500614X> (дата звернення: 22.12.2021).

19. ArPi Lab: A Low-cost Remote Laboratory for Control Education
URL: <https://www.sciencedirect.com/science/article/pii/S1474667016430430>
(дата звернення: 22.12.2021).
20. An internet-based real-time remote automatic control laboratory for control education URL: <https://www.sciencedirect.com/science/article/pii/S1877042810008992> (дата звернення: 22.12.2021).
21. IFAC Workshop on Internet Based Control Education IBCE15: Brescia (Italy), November 4-6, 2015
22. Facilitating the Creation of Virtual and Remote Laboratories for Science and Engineering Education URL: <https://www.sciencedirect.com/science/article/pii/S2405896315024702> (дата звернення: 22.12.2021).
23. Effects of remote labs on cognitive, behavioral, and affective learning outcomes in higher education URL: <https://www.sciencedirect.com/science/article/abs/pii/S0360131519301435> (дата звернення: 22.12.2021).
24. Learning in the age of SARS-COV-2: A quantitative study of learners' performance in the age of emergency remote teaching URL: <https://www.sciencedirect.com/science/article/pii/S2666557320300021> (дата звернення: 22.12.2021).
25. Weick, K. (1995), Sense making in Organizations, Sage, Beverly Hills, CA.
26. Russell R. F. and Stone A. G. (2002). A review of servant leadership attributes: developing a practical model. Leadership & Organization Development Journal, Vol. 23, pp. 145–157.

27. Goodwin, J. and Fulmer, R.M. (1995), «The systems dynamics of executive education», Executive Development, Vol. 8, No. 4, pp. 9–14.

28. Саймон Монк (2016), «Книга Raspberry Pi. Сборник рецептов. Решение программных и аппаратных задач», 528

29. Eben Upton, Gareth Halfacree (2012), «Raspberry Pi User Guide 1st Edition», Wiley; 1st edition

30. Logitech URL: <https://www.logitech.com/> (дата звернення: 01.02.2022)

ДОДАТОК А

Лістинг коду файлу `setup.py`

```
import numpy
from Cython.Distutils import build_ext

include_dir = '/usr/local/include'
lib_dir = '/usr/local/lib/'

def configuration(parent_package="", top_path=None):
    """ Function used to build our configuration.
    """
    from numpy.distutils.misc_util import Configuration

    # The configuration object that hold information on all the files
    # to be built.
    config = Configuration("", parent_package, top_path)
    config.add_extension('video_streaming',
                        sources=['video_streaming.pyx', './src/VideoEncoder.cpp',
                                './src/VideoDecoder.cpp'],
                        libraries=['avcodec', 'avutil', 'avformat', 'swresample',
                                'swscale', 'z', 'lzma', 'pthread'],
                        depends=['./src/VideoEncoder.cpp', './src/VideoEncoder.h',
                                './src/VideoDecoder.cpp', './src/VideoDecoder.h'],
                        language = "c++",
                        extra_compile_args=['-std=c++11'],
                        include_dirs=[numpy.get_include(), include_dir],
                        library_dirs= [lib_dir])
    return config

if __name__ == '__main__':
    # Retrieve the parameters of our local configuration
    params = configuration(top_path="").todict()

    # Override the C-extension building so that it knows about '.pyx'
    # Cython files
    params['cmdclass'] = dict(build_ext=build_ext)

    # Call the actual building/packaging function (see distutils docs)
    from numpy.distutils.core import setup
    setup(**params)
```

ДОДАТОК Б

Лістинг коду файлу video-streaming.pyx

```
from libcpp.string cimport string
from libcpp.vector cimport vector
from libc.stdint cimport uint8_t
from libc.stdlib cimport free
from cpython cimport PyObject, Py_INCREF
import numpy as np
cimport numpy as np
np.import_array()

cdef extern from "./src/VideoEncoder.h":
    cdef cppclass VideoEncoder:
        VideoEncoder(int, int, int, int, string, int, int, int) except +
        vector[uint8_t] encodeFrame(uint8_t *, int, int, int) except +

    cdef class PyVideoEncoder:
        cdef VideoEncoder * thisptr
        def __cinit__(self, int width, int height, int bitrate, string codec_name, int
fps, int gop_size, int max_b_frames):

            """
            Parameters
            -----
            width : int
                Frame width
            height : int
                Frame height
            bitrate : int
                Video bitrate. The lower is the bitrate, the higher is compression.
            codec_name: str
                The name of the codec to use. Note that the encoder and the
decoder must use the exact same codec. Tested with libx264 and mpeg1video.
            fps: int
                Frames per second
            gop_size: int
                Group of pictures size. Please see
https://en.wikipedia.org/wiki/Group\_of\_pictures
            max_b_frames: int
                Maximum quantity of consecutive B frames. Please see
https://en.wikipedia.org/wiki/Video\_compression\_picture\_types
            """

            self.thisptr = new VideoEncoder(width, height, bitrate, 0, codec_name,
fps, gop_size, max_b_frames)

        def __dealloc__(self):
```

```
    del self.thisptr
def encodeFrame(self, frame):
    """
    Parameters
    -----
    frame: np.array
           RGB frame

    Returns
    -----
    bytes
           a byte message
    """
    cdef np.ndarray[np.uint8_t, ndim=3, mode = 'c'] np_buff =
np.ascontiguousarray(frame, dtype = np.uint8)
    cdef uint8_t * im_buff = <uint8_t *> &np_buff[0, 0, 0]
    cdef vector[uint8_t] frame_bytes
    frame_bytes = self.thisptr.encodeFrame(im_buff, frame.strides[0],
frame.strides[1], frame.strides[2])
    return bytes(frame_bytes)

cdef extern from "./src/VideoDecoder.h":
    cdef struct VideoFrame:
        int width
        int height
        int nc
        int pts
        uint8_t * data

cdef extern from "./src/VideoDecoder.h" nogil:
    cdef cppclass VideoDecoder:
        VideoDecoder(int, int, string) except +
        void supplyBytes(uint8_t *, int) except +
        VideoFrame getFrame() except +
    cdef class ArrayWrapper:
        cdef uint8_t* data_ptr
        cdef int width
        cdef int height
        cdef int nc
        cdef set_data(self, int width, int height, int nc, uint8_t* data_ptr):
            self.data_ptr = data_ptr
            self.width = width
            self.height = height
            self.nc = nc

    def __array__(self):
        cdef np.npy_intp shape[3]
        shape[0] = <np.npy_intp> self.height
```



```

        shape[1] = <np.npy_intp> self.width
        shape[2] = <np.npy_intp> self.nc

        ndarray = np.PyArray_SimpleNewFromData(3, shape, np.NPY_UINT8,
        <uint8_t *> self.data_ptr)

        return ndarray

    def __dealloc__(self):
        free(<uint8_t*>self.data_ptr)

cdef class PyVideoDecoder:

    """
    Decodes video

    ...

    Methods
    -----
        supplyBytes(data)
            Accepts a byte message
        getFrame()
            Blocks until a frame is available. Returns a frame and a presentation
timestamp
    """

cdef VideoDecoder * thisptr

    def __cinit__(self, int width, int height, string codec_name):

        """
        Parameters
        -----
        width : int
            Frame width
        height : int
            Frame height
        codec_name : str
            The name of the codec to use. Note that the encoder and the
decoder must use the exact same codec. Tested with h264 and mpeg1video.
        """

        self.thisptr = new VideoDecoder(width, height, codec_name)

    def __dealloc__(self):

        del self.thisptr

```

```
def supplyBytes(self, data):  
    """  
    Parameters  
    -----  
    data: bytes  
        Bytes message  
    """  
  
    size = len(data)  
    cdef uint8_t * data_ptr = data  
    self.thisptr.supplyBytes(data_ptr, size)  
  
def getFrame(self):  
    """  
    Blocks until a frame is available  
  
    Returns  
    -----  
    ndarray  
        Decoded frame  
    pts  
        Presentation timestamp  
    """  
  
    with nogil:  
        video_frame = self.thisptr.getFrame()  
        cdef np.ndarray ndarray  
        array_wrapper = ArrayWrapper()  
        array_wrapper.set_data(video_frame.width, video_frame.height,  
video_frame.nc, video_frame.data)  
        ndarray = np.array(array_wrapper, copy=False)  
        ndarray.base = <PyObject*> array_wrapper  
        Py_INCREF(array_wrapper)  
        return ndarray, video_frame.pts
```

ДОДАТОК В

Лістинг коду файлу sender.py

```
import socket
from queue import Queue
import threading
from threading import Lock
import argparse
import cv2
from video_streaming import PyVideoEncoder

class FrameSender:

    def __init__(self, width, height, bitrate=10000000,
codec_name="libx264", fps=25, gop_size=10, max_b_frames=1):

        self.socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

        self.frame_queue = Queue()

        self.image_sending_thread =
threading.Thread(target=FrameSender._encode_send_frame, args=(self,))

        self.encoder = PyVideoEncoder(width, height, bitrate,
codec_name.encode('utf-8'), fps, gop_size, max_b_frames)

    def connect(self, ip, port):

        self.socket.connect((ip, port))
        self.image_sending_thread.start()

    def _encode_send_frame(self):

        while True:
            frame, pts = self.frame_queue.get()
            frame_bytes = self.encoder.encodeFrame(frame)
            self.socket.sendall(frame_bytes)

    def send_frame(self, frame, pts):
        self.frame_queue.put((frame, pts))
```

```
def parse_args():
    parser = argparse.ArgumentParser()
    parser.add_argument('--ip', type=str, default="0.0.0.0")
    parser.add_argument('--port', type=int, default=17098)
    parser.add_argument('--frame-width', type=int, default=640)
    parser.add_argument('--frame-height', type=int, default=480)
    parser.add_argument('--bitrate', type=int, default=1000000)
    parser.add_argument('--codec', type=str, default="libx264")

    args = parser.parse_args()
    return args

if __name__ == "__main__":

    args = parse_args()

    frame_sender = FrameSender(args.frame_width, args.frame_height,
    bitrate=args.bitrate, codec_name=args.codec)
    frame_sender.connect(args.ip, args.port)

    cap = cv2.VideoCapture(0)

    pts = 0
    while True:

        ret, frame = cap.read()

        if not ret:
            continue

        frame_sender.send_frame(frame, pts)
        pts += 1
```

ДОДАТОК Г

Лістинг коду файлу receiver.py

```
import socket
import threading
from queue import Queue
from video_streaming import PyVideoDecoder
import numpy as np
import argparse
import cv2

BUFSIZE = 4 * 1024

class FrameReceiver:

    def __init__(self, port, width, height, codec="h264"):

        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.bind(('0.0.0.0', port))
        self.socket = None

        self.frame_queue = Queue()

        self.decoder = PyVideoDecoder(width, height, codec.encode('utf-8'))

        self.image_receiving_thread =
threading.Thread(target=FrameReceiver._receive_images, args=(self,))
        self.image_decoding_thread =
threading.Thread(target=FrameReceiver._decode_images, args=(self,))

    def _decode_images(self):

        while True:
            frame, num = self.decoder.getFrame()
            self.frame_queue.put_nowait((num, frame))

    def _receive_images(self):

        while True:
            bytes_chunk = self.socket.recv(BUFSIZE)
            self.decoder.supplyBytes(bytes_chunk)

    def run(self):

        print('awaiting connection')

        self.sock.listen(5)
        self.socket, address = self.sock.accept()
```

```
print('connection established')

self.image_receiving_thread.start()
self.image_decoding_thread.start()
print("receiver is running")

while True:
    pts, frame = self.frame_queue.get()
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

def parse_args():
    parser = argparse.ArgumentParser()
    parser.add_argument('--port', type=int, default=17098)
    parser.add_argument('--frame-width', type=int, default=640)
    parser.add_argument('--frame-height', type=int, default=480)
    parser.add_argument('--codec', type=str, default="h264")

    args = parser.parse_args()
    return args

if __name__ == "__main__":

    args = parse_args()

    receiver = FrameReceiver(args.port, args.frame_width,
args.frame_height, args.codec)
    receiver.run()
```

ДОДАТОК Д

Лістинг коду файлу VideoDecoder.cpp

```
#include "VideoDecoder.h"
#include <stdexcept>

extern "C"{
    #include <libswscale/swscale.h>
    #include <libavcodec/avcodec.h>
}
#define INBUF_SIZE 4096

#include <condition_variable>
#include <mutex>
#include <queue>
#include <cstring>

struct VideoDecoder::Impl{

    Impl(int width, int height, std::string codec_name): _width(width),
    _height(height){

        _pkt = av_packet_alloc();
        if (!_pkt){

            throw std::runtime_error("Could not allocate packet\n");

        }

        memset(_inbuf + INBUF_SIZE, 0,
AV_INPUT_BUFFER_PADDING_SIZE);

        _codec = avcodec_find_decoder_by_name(codec_name.c_str());

        if (!_codec) {
            throw std::runtime_error("Codec not found\n");
        }

        _parser = av_parser_init(_codec->id);

        if (!_parser) {
            throw std::runtime_error("Parser not found\n");
        }
        _codec_context = avcodec_alloc_context3(_codec);

        if (!_codec_context) {

            throw std::runtime_error("Could not allocate video codec
context\n");
```

```

    }
    if (avcodec_open2(_codec_context, _codec, NULL) < 0) {
        throw std::runtime_error("Could not open codec\n");
    }
    _frame = av_frame_alloc();
    if (!_frame) {
        throw std::runtime_error("Could not allocate video frame\n");
    }
    _sws_context = sws_getContext(width, height, AV_PIX_FMT_YUV420P,
        width, height, AV_PIX_FMT_RGB24,
        0, NULL, NULL, NULL);
    if(!_sws_context){
        throw std::runtime_error("Could not allocate sws context\n");
    }
}
void decode(){

    int ret;
    ret = avcodec_send_packet(_codec_context, _pkt);
    if (ret < 0) {

        throw std::runtime_error("Error sending a packet for decoding\n");
    }
    while (ret >= 0) {

        ret = avcodec_receive_frame(_codec_context, _frame);

        if (ret == AVERROR(EAGAIN) || ret == AVERROR_EOF)
            return;

        else if (ret < 0) {

            throw std::runtime_error("Error during decoding\n");
        }

        VideoFrame video_frame;
        video_frame.width = _width;
        video_frame.height = _height;
        video_frame.nc = 3;
        video_frame.data = (uint8_t *) std::malloc(_width * _height * 3);
        video_frame.pts = _codec_context->frame_number;

        int out_linesize[1] = { 3 * _width };
        sws_scale(_sws_context, _frame->data, _frame->linesize, 0,
            _height, &video_frame.data, out_linesize);

    }
}

```



```
        std::lock_guard<std::mutex> lock(_mutex);
        _queue.push(video_frame);
    }
    _inputCondition.notify_one();
}

}

void supplyBytes(uint8_t * input_bytes, int size){

    uint8_t * input_bytes_alias = input_bytes;
    int num_available_bytes = size;

    while(num_available_bytes > 0){

        int num_bytes_to_copy = num_available_bytes >= INBUF_SIZE ?
INBUF_SIZE : num_available_bytes;

        std::memcpy(_inbuf, input_bytes_alias, num_bytes_to_copy);

        num_available_bytes -= num_bytes_to_copy;
        input_bytes_alias += num_bytes_to_copy;

        int data_size = num_bytes_to_copy;

        uint8_t * data = _inbuf;

        while (data_size > 0) {

            int ret = av_parser_parse2(_parser, _codec_context, &_pkt-
>data, &_pkt->size,
                                data, data_size, AV_NOPTS_VALUE,
AV_NOPTS_VALUE, 0);
            if (ret < 0) {

                throw std::runtime_error("Error while parsing\n");
            }

            data += ret;
            data_size -= ret;
            if (_pkt->size)
                decode();
        }
    }

}

VideoFrame getFrame(){
```

```
std::unique_lock<std::mutex> lock(_mutex);
_inputCondition.wait(lock, [&]{ return !_queue.empty(); });
VideoFrame frame = _queue.front();
_queue.pop();
return frame;
}
~Impl(){
    av_parser_close(_parser);
    avcodec_free_context(&_codec_context);
    av_frame_free(&_frame);
    av_packet_free(&_pkt);
    sws_freeContext(_sws_context);
}
private:
    int _width;
    int _height;

    AVCodec          *_codec          = NULL;
    AVCodecParserContext *_parser      = NULL;
    AVCodecContext   *_codec_context = NULL;
    AVPacket         *_pkt           = NULL;
    AVFrame          *_frame         = NULL;
    struct SwsContext *_sws_context  = NULL;

    uint8_t _inbuf[INBUF_SIZE + AV_INPUT_BUFFER_PADDING_SIZE];

    std::condition_variable _inputCondition;
    std::mutex              _mutex;
    std::queue<VideoFrame> _queue;
};
VideoDecoder::VideoDecoder(int width, int height, std::string codec_name):
_self( new Impl(width, height, codec_name)){

VideoDecoder::~VideoDecoder(){

    delete _self;
}
void VideoDecoder::supplyBytes(uint8_t * data, int size){

    _self->supplyBytes(data, size);
}

VideoFrame VideoDecoder::getFrame(){

    return _self->getFrame();
}
}
```

ДОДАТОК Е

Лістинг коду файлу VideoDecoder.h

```
#include <string>
#include <stdint.h>
#include <vector>

struct VideoFrame{
    int width = 0;
    int height = 0;
    int nc = 0;
    int pts;
    uint8_t * data = NULL;
};

class VideoDecoder{

    struct Impl;
    Impl * _self;

public:

    VideoDecoder(int width, int height, std::string codec_name);
    ~VideoDecoder();

    void supplyBytes(uint8_t * data, int size);

    VideoFrame getFrame();

    //void terminate();

};
```

ДОДАТОК Ж

Лістинг коду файлу VideoEncoder.cpp

```
#include "VideoEncoder.h"
#include <stdexcept>

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctime>
#include <iostream>

extern "C"{
    #include <libswscale/swscale.h>
    #include <libavcodec/avcodec.h>
    #include <libavutil/opt.h>
    #include <libavutil/imgutils.h>
}

struct VideoEncoder::Impl{

    Impl(int width, int height, int bitrate, int pixel_fmt, std::string
codec_name, int fps, int gop_size, int max_b_frames): _width(width),
_height(height), _pixel_fmt(pixel_fmt){

        //TODO release resources before every throw

        _codec = avcodec_find_encoder_by_name(codec_name.c_str());

        if (!_codec) {

            throw std::runtime_error(("Codec " + codec_name + "not
found\n").c_str());
        }

        _codec_context = avcodec_alloc_context3(_codec);

        if (!_codec_context){

            throw std::runtime_error("Could not allocate video codec
context\n");
        }

        _pkt = av_packet_alloc();
        if (!_pkt){

            throw std::runtime_error("Could not allocate packet\n");
        }
    }
};
```

```
_codec_context->bit_rate = bitrate;

_codec_context->width = _width;
_codec_context->height = _height;

_codec_context->time_base = (AVRational){1, fps};
_codec_context->framerate = (AVRational){fps, 1};

_codec_context->gop_size = gop_size;
_codec_context->max_b_frames = max_b_frames;

//TODO assign actual pixel format
_codec_context->pix_fmt = AV_PIX_FMT_YUV420P;

if (_codec->id == AV_CODEC_ID_H264)
    av_opt_set(_codec_context->priv_data, "preset", "slow", 0);

int ret = avcodec_open2(_codec_context, _codec, NULL);

if (ret < 0) {

    throw std::runtime_error("Could not open codec\n");
}

_frame = av_frame_alloc();

if (!_frame) {

    throw std::runtime_error("Could not allocate video frame\n");
}

_frame->format = _codec_context->pix_fmt;
_frame->width = _codec_context->width;
_frame->height = _codec_context->height;

ret = av_frame_get_buffer(_frame, 32);

if (ret < 0) {

    throw std::runtime_error("Could not allocate the video frame
data\n");
}

_sws_context = sws_getContext(width, height, AV_PIX_FMT_RGB24,
                             width, height, AV_PIX_FMT_YUV420P,
                             0, NULL, NULL, NULL);
if(!_sws_context){
```

```
        throw std::runtime_error("Could not allocate sws context\n");
    }
}

~Impl(){
    avcodec_free_context(&_amp;_codec_context);
    av_frame_free(&_amp;_frame);
    av_packet_free(&_amp;_pkt);
}

void encode(std::vector<uint8_t> & bytes) {
    int ret;

    ret = avcodec_send_frame(_amp;_codec_context, _amp;_frame);

    if (ret < 0) {
        throw std::runtime_error("Error sending a frame for encoding\n");
    }

    while (ret >= 0) {
        ret = avcodec_receive_packet(_amp;_codec_context, _amp;_pkt);

        if (ret == AVERROR(EAGAIN) || ret == AVERROR_EOF)
            return;

        else if (ret < 0) {
            throw std::runtime_error("Error during encoding\n");
        }

        //TODO possibly use memcpy
        for(int i = 0; i < _amp;_pkt->size; ++i){
            bytes.push_back(_amp;_pkt->data[i]);
        }

        av_packet_unref(_amp;_pkt);
    }
}

std::vector<uint8_t> encodeFrame(uint8_t * data, int stride1, int stride2,
int stride3){
    uint8_t * inData[1] = { data }; // RGB24 have one plane
```

```
int inLinesize[1] = { 3*_width };

sws_scale(_sws_context, inData, inLinesize, 0, _height, _frame->data,
_frame->linesize);

_frame->pts = _pts++;

std::vector<uint8_t> bytes;
encode(bytes);

return bytes;
}
```

private:

```
AVCodec* _codec = NULL;
AVCodecContext* _codec_context = NULL;
AVFrame* _frame = NULL;
AVPacket* _pkt = NULL;
struct SwsContext * _sws_context = NULL;
```

```
int _pts = 0;
```

```
int _width;
int _height;
int _bitrate;
int _pixel_fmt;
```

```
};
```

```
VideoEncoder::VideoEncoder(int width, int height, int bitrate, int pixel_fmt,
std::string codec_name, int fps, int gop_size, int max_b_frames):
    _self(new Impl(width, height, bitrate, pixel_fmt, codec_name, fps,
gop_size, max_b_frames)){}
```

```
VideoEncoder::~~VideoEncoder(){
```

```
    delete _self;
}
```

```
std::vector<uint8_t> VideoEncoder::encodeFrame(uint8_t * data, int
stride1, int stride2, int stride3){
```

```
    return _self->encodeFrame(data, stride1, stride2, stride3);
}
```

ДОДАТОК И

Лістинг коду файлу VideoEncoder.h

```
#include <string>
#include <stdint.h>
#include <vector>

class VideoEncoder{

    struct Impl;
    Impl * _self;

public:

    VideoEncoder(int width, int height, int bitrate, int pixel_fmt, std::string
codec_name, int fps, int gop_size, int max_b_frames);
    ~VideoEncoder();

    std::vector<uint8_t> encodeFrame(uint8_t * data, int stride1, int stride2,
int stride3);

    //void terminate();

};
```