

Харківський національний університет радіоелектроніки

Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В.

**ПРОЕКТУВАННЯ МОБІЛЬНИХ РОБОТІВ  
НА БАЗІ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ  
(Raspberry Pi и мови Python 3.6)**

Харків

2020

УДК 621.396.6:658.51

Н40

**РЕЦЕНЗЕНТИ:**

доктор технічних наук, професор,  
завідувач кафедри «Автоматизації виробничих процесів»  
Центральноукраїнського національного технічного університету  
**ОСАДЧИЙ СЕРГІЙ ІВАНОВИЧ**

доктор технічних наук, професор,  
завідувач кафедри «Автоматизації та комп'ютерно-інтегрованих технологій»  
Харківського національного автомобільно-дорожнього університету  
**НЕФЬОДОВ ЛЕОНІД ІВАНОВИЧ**

доктор технічних наук, професор,  
професор кафедри «Теоретичної та прикладної інформатики»  
Харківського національного університету імені В. Н. Каразіна  
**ФРОЛОВ В'ЯЧЕСЛАВ ВІКТОРОВИЧ**

**Невлюдов І.Ш., Андрусевич А.О., Євсєєв В.В.**

Проектування мобільних роботів на базі одноплатних комп'ютерів (Raspberry Pi и мови Python 3.6): Підручник. – Харків: 2020.-257

У підручнику розглянуті питання проектування і розробки мобільних платформ на базі одноплатного комп'ютера Raspberry Pi model 3 plus. Проведено аналіз сучасних одноплатних комп'ютерів Asus Tinker Board, Raspberry Pi, Orange Pi, LattePanda, Odroid, Pine64 і їх характеристик. Приділення уваги платам розширення і модулів які можуть бути використовуватися при рішень задач розробки мобільних робототехнічних платформ, описані основні характеристики сенсерія і датчиків. На базі одноплатного комп'ютера Raspberry Pi model B plus показана архітектура і електричні характеристики, послідовні інтерфейси I2C, SPI, UART і особливості шини GPIO. Наведено послідовність установки і настройки OS Raspbian для Raspberry Pi model B для реалізацій завдань віддаленого адміністрування і розробки систем управління мобільною платформою з використанням бібліотеки Open CV для реалізації системи комп'ютерного зору і передачу відео потоку по технологи IoT. Наведено покроковий приклад розробки мобільного робототехнічної платформи на базі 4WD від фірми Elecrow яка включає розробку схеми управління, розрахунок і вибір преобразователів напруги. Показана програмна реалізація підключення датчиків і серводвигунів, а також скрипти на мові Python реалізацій системи управління робототехнічної мобільною платформою.

Посібник призначений для студентів закладів вищої освіти в галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи».

Для зручності сприйняття інформації і акцентування уваги група авторів пропозиції наступні варіанти написання:

*sudo apt-get update*

*sudo apt-get upgrade*

Команди які виконуються послідовно в терміналі Raspbian OS.

Написання *interface wlan0* – розглядається як варіант представленого вище опису при інших параметрах настройки або назва програмного забезпечення рекомендованого для роботи.

Так само є всі необхідні наступні графічні позначення:



*Текст подано на замітку, який не критичний, має загальні положення але необхідний для розуміння роботи*



*Важливий інформаційний опис без якого може бути критична помилка.*

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	7
ВСТУП .....	11
1. АНАЛІЗ СУЧАСНИХ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ ЯКІ ПРИМЕНЯЮТЬСЯ ПРИ ПРОЕКТИРОВАНИЙ МОБІЛЬНИХ РОБОТІВ .....	14
1.1 Основні поняття і визначення .....	14
1.2 Asus Tinker Board .....	21
1.3 Raspberry Pi Foundation .....	24
1.4 Orange Pi .....	31
1.5 LattePanda .....	36
1.6 Odroid .....	38
1.7 Pine64 .....	43
1.8 Плати розширення, модулі .....	45
1.9 Датчики .....	65
1.10 Контрольні запитання .....	92
2. АРХІТЕКТУРА ОДНОПЛАТНИХ КОМП'ЮТЕРІВ RASPBERRY PI MOBEL B PLUS .....	94
2.1 Основні елементи Raspberry Pi 3 model B plus .....	94
2.2 Електричні характеристики Raspberry Pi 3 - Model B Plus .....	96
2.3 GPIO .....	101
2.4 Послідовний інтерфейс I <sup>2</sup> C .....	105
2.5 Послідовний інтерфейс SPI .....	109
2.6 Послідовний порт UART .....	112
2.7 Контрольні запитання .....	114
3. УСТАНОВКА І НАЛАШТУВАННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ RASPBIAN .....	116
3.1 Установка OS Raspbian за допомогою NOOBS .....	116

3.2 Базові команди Linux для Raspberry Pi .....	120
3.3 Налаштування OS Raspbian .....	126
3.3.1 Установка екранної клавіатури .....	126
3.4 Налаштування віддаленого доступу .....	128
3.4.1 Налаштування віддаленого робочого столу VNC Viewer .....	128
3.4.2 Віддалений робочий стол через Internet .....	136
3.5 Налаштування samba на OS Raspbian .....	137
3.5.1 Налаштування доступу до існуючих папок OS Raspbian .....	140
3.6 Створення резервної копії OS Raspbian .....	142
3.7 Контрольні запитання .....	145
4. УСТАНОВКА І НАСТРОЙКА PYTHON ДЛЯ РОЗРОБКИ СИСТЕМ УПРАВЛІННЯ МОБІЛЬНИМИ РОБОТАМИ .....	147
4.1 Установка бібліотеки OpenCV .....	147
4.2 Установка бібліотеки для роботи с GPIO.....	155
4.3 Контрольні запитання .....	159
5. ПРОЕКТУВАННЯ МОБІЛЬНОЇ ПЛАТФОРМИ НА БАЗІ RASPBERRY PI .....	160
5.1 Збірка мобільної платформи 4WD от фирмы Elecrow .....	160
5.2 Розробка системи управління DC моторами на базі модуля L298 .....	162
5.3 Вибір перетворювача напруги для управління живленням мобільної платформи .....	170
5.4. Розробка схеми живлення мобільної платформи.....	172
5.5 Розрахунок часу автономної роботи мобільної платформи ....	174
5.6 Контрольні запитання .....	177
6. ПІДКЛЮЧЕННЯ ТА ПРОГРАМУВАННЯ ДАТЧИКІВ НА МОБІ РАУТЕН .....	178
6.1 Підключення ультразвукового датчика HC-SR04 .....	178
6.2 Підключення та програмування датчика температури і вологості DHT22 .....	181

6.3 Підключення та програмування управління сервоприводами для системи комп'ютерного зору .....	188
6.4 Контрольні запитання .....	191
7. РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНОЮ ПЛАТФОРМОЮ НА БАЗІ МОВИ РАУТЕН .....	192
7.1 Скрипт управління DC двигунами на мобільній платформі ...	192
7.2 Підключення і настройка відео стріму з камери Raspbian Pi ...	195
7.3. Розробка алгоритмів керування мобільного робота .....	202
7.4 Розробка програми віддаленого управління мобільним роботом .....	212
7.5 Розробка клієнтської програми удаленого управління мобільним роботом .....	214
7.6 Встановлення необхідних пакетів та налаштувань .....	217
7.7 Контрольні запитання .....	221
ПЕРЕЛІК ПОСИЛАНЬ .....	223
Додаток А. Розташування виводів Asus Tinker Board 2Гб на GPIO .....	239
Додаток Б. Розташування виводів Raspberry Pi 2 &3 на GPIO .....	240
Додаток В. Порівняльні характеристики основних моделей Raspberry Pi .....	241
Додаток Г. Розташування виводів SB LattePanda 2Гб .....	243
Додаток Д. Порівняльна таблиця ODROID-C2 з аналогічними міні комп'ютерами .....	244
Додаток Е. Програма сервісної частини дистанційного керування мобільним роботом .....	245
Додаток Є. Програмний засіб серверної частини системи дистанційного керування мобільного робота .....	247
Додаток Ж. Скрипт Python для ультразвукового далекоміра .....	252
Додаток З. Скрипт Python для одночасного управління двома сервомоторами .....	254

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- SBC - (single-board computer) Одноплатні комп'ютери в ультра-маленькому форм-факторі;
- PoE - (Power over Ethernet) технологія, що дозволяє передавати віддаленого пристрою електричну енергію разом з даними через стандартну виту пару в мережі Ethernet;
- USB - (Universal Serial Bus) послідовний інтерфейс для підключення периферійних пристроїв до обчислювальної техніки;
- GPIO - (general-purpose input/output) інтерфейс для зв'язку між компонентами комп'ютерної системи, наприклад мікропроцесором і різними периферійними пристроями. Контакти GPIO можуть виступати як в ролі входу, так і в ролі виходу - це, як правило, конфігурується. GPIO контакти часто групуються в порти;
- MicroUSB - мініатюрні роз'єми для застосування USB в переносних і мобільних пристроях;
- HDMI - (High Definition Multimedia Interface) інтерфейс для мультимедіа високої чіткості, дозволяє передавати цифрові відеодані високої роздільної здатності та багатоканальні цифрові аудіосигнали із захистом від копіювання;
- CSI - інтерфейс для CSI камери;
- DSI - конектор для LCD дисплею;
- PXE - (Preboot eXecution Environment) завантаження ОС через мережевий інтерфейс використання локальних носіїв даних;

- ZigBee - специфікація мережевих протоколів регламентованих стандартом IEEE 802.15.4. ZigBee і IEEE 802.15.4 описують бездротові персональні обчислювальні мережі (WPAN) ;
- GPS - (Global Positioning System) супутникова система навігації, яка забезпечує вимірювання відстані, часу та визначає місце розташування у всесвітній системі координат WGS 84;
- GSM - (Global System for Mobile Communications) глобальний стандарт цифрового мобільного стільникового зв'язку з розділенням каналів за часом (TDMA) і частоті (FDMA) ;
- ШИМ (PWM) - (pulse-width modulation) процес управління потужністю методом пульсуючого включення і виключення приладу. Розрізняють аналогову ШИМ і цифрову ШИМ, двійкову (дворівневу) ШИМ і трійкову (трирівневу) ШИМ;
- SPI - (Serial Peripheral Interface) послідовний синхронний стандарт передачі даних в режимі повного дуплексу, призначений для забезпечення простого і недорогого високошвидкісного сполучення мікроконтролерів і периферії;
- SD - (Secure Digital Memory Card) формат карт пам'яті (флеш-пам'ять), розроблений SD Association (SDA) для використання в портативних пристроях;
- TFT - (thin-film transisto) рідкокристалічний дисплей з активною матрицею;
- I<sup>2</sup>C - (Inter-Integrated Circuit) послідовна асиметрична шина для зв'язку між інтегральними схемами всередині електронних приладів;

- SSH - (Secure Shell) мережевий протокол прикладного рівня, що дозволяє виробляти віддалене управління операційною системою і тунелювання TCP-з'єднань;
- SDRAM - (Synchronous Dynamic Random Access Memory) тип пристрою, що запам'ятовує, та використовується в комп'ютері і інших цифрових пристроях як ОЗУ;
- DDR3 - (double-data-rate three synchronous dynamic random access memory) це тип оперативної пам'яті, що використовується в обчислювальній техніці в якості оперативної і відеопам'яті;
- IoT - (internet of things) концепція обчислювальної мережі фізичних предметів, оснащених вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем, яка розглядає організацію таких мереж як явище, здатне перебудувати економічні та суспільні процеси, що виключає з частини дій і операцій необхідність участі людини;
- LPDDR3 - стандарт JESD209-3 «Low Power Memory Device Standard» пропонується більш висока швидкість обміну даними, збільшена енергоефективність і велика щільність пам'яті;
- CPU - (central processing unit) інтегральна схема (мікропроцесор), виконуюча машинні інструкції (код програм), головна частина апаратного забезпечення комп'ютера або програмованого логічного контролера;
- GPU - (graphics processing unit) інтегральна схема (мікропроцесор), для обробки і вирішення графічних завдань;
- FPS - (Frames Per Second) кількість кадрів в секунду на екрані монітора;

- eMMC - (embedded Multimedia Memory Card) це одночіповий накопичувач, найчастіше вживаний в компактних пристроях;
- DSI - (Display Interface Specifications) Специфікація визначає послідовну шину і протокол зв'язку між хостом (джерело зображення) і пристроєм (одержувачем зображення) ;
- DHCP - (Dynamic Host Configuration Protocol) протокол динамічної настройки вузла, що використовується для автоматичного динамічного присвоєння IP-адрес пристроїв мережі;
- КЗ - комп'ютерний зір;
- SMB/CIFS - (Server Message Block) мережевий протокол прикладного рівня для віддаленого доступу до файлів, принтерів та інші ресурси мережі, а також для взаємодії між процесами;
- RPi - Raspberry Pi ;
- ПО - програмне забезпечення.

## ВСТУП

Технологічний вектор постіндустріального суспільства визначається переходом на повністю автоматизоване цифрове виробництво із застосуванням кіберфізических систем. Важливою частиною таких систем є автономні мобільні промислові роботи, які вже стали економічно вигідною альтернативою повагою до людської природи в розширенні спектрі галузей. За оцінкою McKinsey Global Institute (MGI), економія операційних витрат від автоматизації в цілому може становити від 15% до 90% в залежності від галузі. У світі робототехніки вже більше десятиліття використовується досить корисний індикатор зростання ринку - щільність роботизації. Він вимірюється як число роботів на 10 000 працівників, зайнятих в промисловості. Цей показник знаходиться в хорошій кореляції як зі станом економічного розвитку, так і темпами зростання. За даними IFR, в 2019 р. на 10000 зайнятих в промисловості людей по всьому світу доводилося в середньому 120-125 роботів .

Підручник являє собою логічну послідовність розділів, яка дозволяє вивчити і поглибити знання проектування мобільних робототехнічних платформ на базі одноплатних комп'ютерів з використанням сучасних технічних і програмних підходів для їх реалізації.

Перший розділ присвячений рассмотренню сучасних одноплатних комп'ютерів які приміняються при проектуванні мобільних робототехнічних платформ. Проведено аналіз одноплатних комп'ютерів Asus Tinker Board, Raspberry Pi, Orange Pi, LattePanda, Odroid, Pine64 їх основних характеристик, переваг і недоліків з точки зору застосування для проектування мобільних робототехнічних платформ. Приділено увагу сучасним платам розширення і модулів для одноплатних комп'ютерів, рассмотреного їх характеристики і рекомендації щодо їх застосування. Детально розглянуті датчики які можуть застосовуватися при

проектирований мобільних робототехнічних платформ. Визначено основні поняття і визначення концепцій Industry 4.0 і кіберфізических систем з точки зору застосування мобільних робототехнічних платформ.

У другому розділі рассмотрено архітектуру одноплатного комп'ютера Raspberry Pi 3 - Model B Plus, призначення основних елементів і його електричні характеристики. Детально описано методи підключення і роботи послідовних інтерфейсів I<sup>2</sup>C, SPI, UART. Приділено більшої уваги 40-бітній шині GPIO на базі якої здійснюється взаємодія Raspberry Pi 3 з датчиком і виконавчим пристроєм на мобільній робототехнічній платформі.

Третій розділ присвячено питанням установки і налаштування OS Raspbian, детально описані базові команди і особливості налаштування. Показана реалізація створення віддаленого доступу до Raspberry Pi 3 на базі VNC і Internet, а також налаштування сервера Samba. Розглянуто питання щодо створення резервної копії OS Raspbian під OS Windows 10 і iOS.

Особливості установки і налаштування бібліотек OpenCV і CPIO наведені в четвертому розділі. Показана покрокова інструкція компіляції бібліотеки OpenCV для мови Python і перевірки її роботоздатності для реалізації завдань комп'ютерного зору.

Окремо в п'ятому розділі наведено приклад проектування та збирання мобільної робототехнічної платформи, розробки системи управління DC моторами на базі модуля L298. Детально розглянуто методи управління, схема складання і принципи роботи модуля L298. Обґрунтовується вибір преобразувача напруги, розроблена схема підключення і проведені розрахунки автономної роботи мобільної робототехнічної платформи.

У шостому розділі наведені схеми підключення датчиків і виконавчих пристроїв через шину GPIO. Рассмотрено програмні скрипти на мові Python для їх управління за допомогою Raspberry Pi 3. Приділено увагу до програмної реалізації управління сервоприводами за допомогою яких

реалізується переміщення камери комп'ютерного зору в горизонтальних і вертикальних площинах щодо мобільного робототехнічної платформи.

Розробка системи віддаленого управління мобільного робототехнічної платформи наведені в цьому розділі. Наведені приклади реалізацій і настройки відео стріму за допомогою технологій Internet. Розроблено загальні алгоритми управління і фрагменти скриптів на мові Python клієнтської і серверної частини мобільного робототехнічної платформи.

Для закріплення вивченого матеріалу в кінці кожного розділу запропоновані контрольні питання які допоможуть перевірити рівень знань. Підручник написано таким чином, що дозволяє на базі представленого матеріалу самостійно провести збірку і розробку системи управління макетом мобільного робота на базі одноплатного комп'ютера Raspberry Pi 3 і мови програмування Python з використанням сучасних бібліотек. Розроблена мобільна робототехнічна платформа може використовуватися для досліджень систем комп'ютерного зору, ідентифікації об'єктів за допомогою нейронних мереж і систем штучного інтелекту.

У підручнику використовується сучасні матеріали опубліковані у відкритих джерелах Євросоюзу і світових виробників одноплатних комп'ютерів, модулів та датчиків.

Підручник призначений для студентів закладів вищої освіти в галузі знань 15 Автоматизація та приладобудування за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи».

# 1. АНАЛІЗ СУЧАСНИХ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ ЯКІ ЗАСТОСОВУТЬСЯ ПРИ ПРОЕКТУВАННІ МОБІЛЬНИХ РОБОТІВ

## 1.1 Основні поняття та визначення

Одноплатний комп'ютер (SBC, англ. Single-board computer) - самодостатній комп'ютер, зібраний на одній друкованій платі, на якій встановлені мікропроцесор, оперативна пам'ять, системи введення-виведення і інші модулі, необхідні для функціонування комп'ютера. Одноплатні комп'ютери виготовляються в ролі промислових або вбудованих комп'ютерів. Які можуть використовуватися для рішення великого спектра завдань, починаючи від ролі навчального макета до систем на яких реалізовані технології IoT і Industry 4.0[1-3].

**IoT** - Інтернет речей (Internet of Things - IoT) включає в себе відразу кілька явищ. Це самі пристрої, які вийшли в мережу і взаємодіють між собою. Це і спосіб підключення - M2M - тобто машини-до-машині, без участі людини. Це і великі дані, які тепер генерують пристрої. Дані, які можна (і потрібно) збирати, аналізувати і надалі використовувати для підвищення комфорту або прийняття бізнес-рішень. Особливо активно IoT розвивається в аграрному секторі, логістиці, Smart City і виробництві. Тобто там, де є потреба у віддаленому моніторингу стану об'єктів або зборі великих даних з метою подальшого аналізу. IoT дає можливість економити на обслуговуванні обладнання: датчики збирають інформацію про його стан, тому техобслуговування і ремонт виробляються саме тоді, коли це потрібно. Профілактика - завжди дешевше ремонту. Україна не стоїть осторонь від світових трендів. Великі міста поступово стають «розумними», флагмани руху - Київ і Львів, Харків. Хоча при комплексному підході і масштабному розгортанні технологій Smart City поки рано. Аграрний сектор, логістика також використовують рішення для IoT.

Хоча в цілому в Україні розвиток відбувається повільно. За словами Андріса Гайлітіса, CEO оператора дата-центрів DEAC - ринок інтернет речей і хмарних технологій в Україні нагадує країни Балтії кілька років тому. Тобто зачатки є, але ще є куди рости. Як приклад можемо навести застосування IoT в містах:

- моніторинг і управління рухом міським транспортом;
- сміттєві баки з датчиками наповнення;
- планування маршрутів транспорту на підставі даних про переміщення людей по місту;
- відеоспостереження;
- контроль за рівнем води в водоймах;
- датчики шуму і забруднення роблять міста зручніше і безпечніше.

А великі дані, які збираються в результаті роботи датчиків, дають можливість владі міста краще розуміти потреби жителів.

В аграрному секторі IoT дозволяє проводити моніторинг щодо стану полів:

- датчики шуму і забруднення роблять міста зручніше і безпечніше.
- дрони ведуть запис з неба і передають їх інженерам;
- в оцінці стану ґрунтів інженерам можуть допомагати нейромережі.

В наслідку чого більше не потрібно фізично проводити моніторинг кожного поля або ділянки, щоб контролювати урожай і стежити за кожним метром землі. У Нідерландах невеликій країні з високою щільністю населення, є одним зі світових лідерів з вирощування продуктів живлення - це стало можливим завдяки впровадженню і використанню технологій IoT.

Завдяки IoT доставка будь-яких товарів з виробництва або зі складів до магазинів стала набагато простіша - що важливо як для кінцевого споживача, так і для бізнесу. Транспортні компанії можуть відстежити, де перебуває автомобіль або в який момент йому пора під'їжджати на завантаження. Крім вантажних автомобілів система застосовується також у водному транспорті IoT в вантажних судах, наприклад: датчики моніторять стан і стан

автомобіля IoT - це позбавляє власників логістичних компаній від зайвих дзвінків і витрат на ремонт.

Smart home є невід'ємною частиною технологій IoT, дане рішення дозволяє контролювати витрату електрики, газу, дистанційно керувати побутовими приладами та системами опалення.

**Industry 4.0** - цей термін був вперше представлений в 2011 році в Німеччині на виставці в Ганновері, як позначення високотехнологічної стратегії уряду Німеччини. Принцип Industry 4.0 полягає в тому, що шляхом з'єднання машин, деталей і систем, а так само інтелектуальних мереж створюються ланцюжки, елементи яких можуть керувати один одним автономно. Це призведе до переходу від традиційної п'ятирівневої піраміди автоматизації (що складається з CNC, PLC, SCADA, MES і ERP) до надзвичайно гнучкого підключення інтелектуальної фабрики на основі хмари.

Об'єднання двох основних елементів дозволяють виробничому середовищу, що змінюється використовувати, так звану, **Industry 4.0: M2M** (Машина-к-Машині) + **IoT**. Зв'язок M2M використовується для автоматичної передачі та вимірювання даних між механічними або електронними пристроями. Типовими компонентами системи M2M є розгорнуті на місці бездротові пристрої з вбудованими датчиками або RFID-бездротові мережі зв'язку.

Кіберфізичні системи (CPS) - забезпечують технології, які об'єднують віртуальні та фізичні світи для створення мережевого світу, в якому інтелектуальні об'єкти спілкуються і взаємодіють один з одним. CPS в обробній промисловості, наприклад, включає в себе всілякі датчики, пристрої та механізми, які стають інтерактивними завдяки вбудованому програмному забезпеченню та підключенню до мережі для моніторингу та управлінню фізичними процесами за допомогою контурів зворотного зв'язку. CPS збирає, зберігає і аналізує дані датчиків через свою локальну бізнес-логіку для надання та використання даних і послуг. Такий децентралізований інтелект

створює інтелектуальну мережу об'єктів і незалежне управління процесами, взаємодіючи з реальним і віртуальним світами. Це являє собою зовсім новий аспект промислових і виробничих процесів. CPS не засновані на новій технології, а використовують і поєднують існуючі та сталі тенденції розумним чином:

- завдяки інноваційній ІТ-індустрії, комунікаційні технології, відповідні апаратні засоби і програмне забезпечення будуть знижуватися в ціні і ставати ефективніше з кожним роком. Це буде дуже популярним серед користувачів у багатьох галузях, оскільки вони дозволяють їм отримувати більше цінної інформації з пристроїв, машин і відповідних процесів, таких як статус, діагностика, параметри процесу і т.п;

- з вищевикладеного випливає, що пристрої, системи і машини - як фізичні об'єкти в реальному світі - будуть все більше і більше підключатися один до одного в заводських мережах, а також в Інтернеті. Після цього, існуючі в мережі «точки даних», апаратура та пристрої будуть зберігати все більше і більше конкретних даних для створення «об'єкта даних» в мережі, який з часом буде зростати до кіберзаснованої другої ідентичності або «кібероб'єкта», крім фізичного існування пристрою в реальному світі;

- фізичні об'єкти будуть продовжувати зберігати дані, які можуть включати специфічні ноу-хау, досвід і т.п. Це призведе до того, що їх кіберідентичність стане більш розумною і в певній мірі, автономною.

У виробництві CPS машини можна розглядати як дзеркальне зображення реальної машини, яка може безперервно записувати і відслідковувати свій стан під час використання. Завдяки високій можливості підключення, пропонованої технологією хмарних обчислень, дзеркальна модель може забезпечувати, наприклад, дані для оптимізації виробництва краще, ніж реальна машина.

Підхід Advantech до Industry 4.0 структурований як тришарова будова, що дозволяє використовувати Інтелектуальні роботи і машини (Рівень 1), підключені Рішення iFactory (Рівень 2) і Інтернет послуги (Рівень 3). Завдяки

інтеграції інформації, швидкому реагуванню та гнучкому виробництву, загальна ефективність і ефективність виробничих процесів буде значно поліпшена. Ця тришарова концепція поєднується з інтенсивним розвитком вертикальних ринків.

Рівень 1: Інтелектуальних машин і робототехніки Advantech пропонує повний спектр продуктів і технологій, включаючи контролери, управління рухом, машинний зір, обчислювальні системи автоматизації та інтегрований контролер верстатів і роботів. Завдяки цим сучасним інструментам і технологіям, традиційні машини стануть більш інтелектуальним (розумним) обладнанням. Мета Advantech - перетворити машини в кіберфізичні системи.

Рівень 2: У середовищі Industry 4.0 всі машини і обладнання підключені до мережі і постійно забезпечують інформацію про стан машини в системі управління технологічним процесом. Таким чином, система може виконати негайний аналіз і швидко вжити необхідних заходів. Крім того, ця інформація може бути інтегрована в великі корпоративні системи, такі як Системи виробничих процесів (MES) і Система планування ресурсів підприємства (ERP). Щоб досягти вищезазначених цілей, Advantech надає рішення, що дозволяють повністю підключити iFactory до мережі. Advantech's WebAccess Integrated IoT Software Suite і Solution Platform можуть підключати машини роботів та обладнання до заводської мережі та інтегруватися з такими системами, як MES і ERP.

Рівень 3: Статус заводського обладнання і виробничої інформації можна безперервно створювати за допомогою повного аналізу даних і розвитку інтернет-послуг. Нескінченний потік інформації - відмінне джерело для розробки нових додатків і творчих послуг. Наприклад, традиційне обслуговування машин залежить від людського судження, яке часто суб'єктивно і не дуже точне. При повному аналізі даних, такому як вібрація підшипників верстата, ми можемо налаштувати модель виникнення збою і, отже, розробити прогностичну систему обслуговування. Таким чином, дія техобслуговування може бути виконана в неробочий час і це не вплине на

регулярну роботу, а також буде покращена ефективність. Інші послуги включають серіалізацію продукту, виробниче виконання або масову настройку. На даний момент часу впровадження технологій Industry 4.0 зачіпає все більше і більше провідних корпорацій і компаній світу у всіх сферах наукової та виробничої діяльності людини. Фахівці сайту IOT Analytics (<https://iot-analytics.com>) представили список міжнародних компаній (рисунок 1.1) які є лідерами впровадження і використання технологій Industry 4.0 в 2019 році.

# Leading Industry 4.0 Vendors 2019

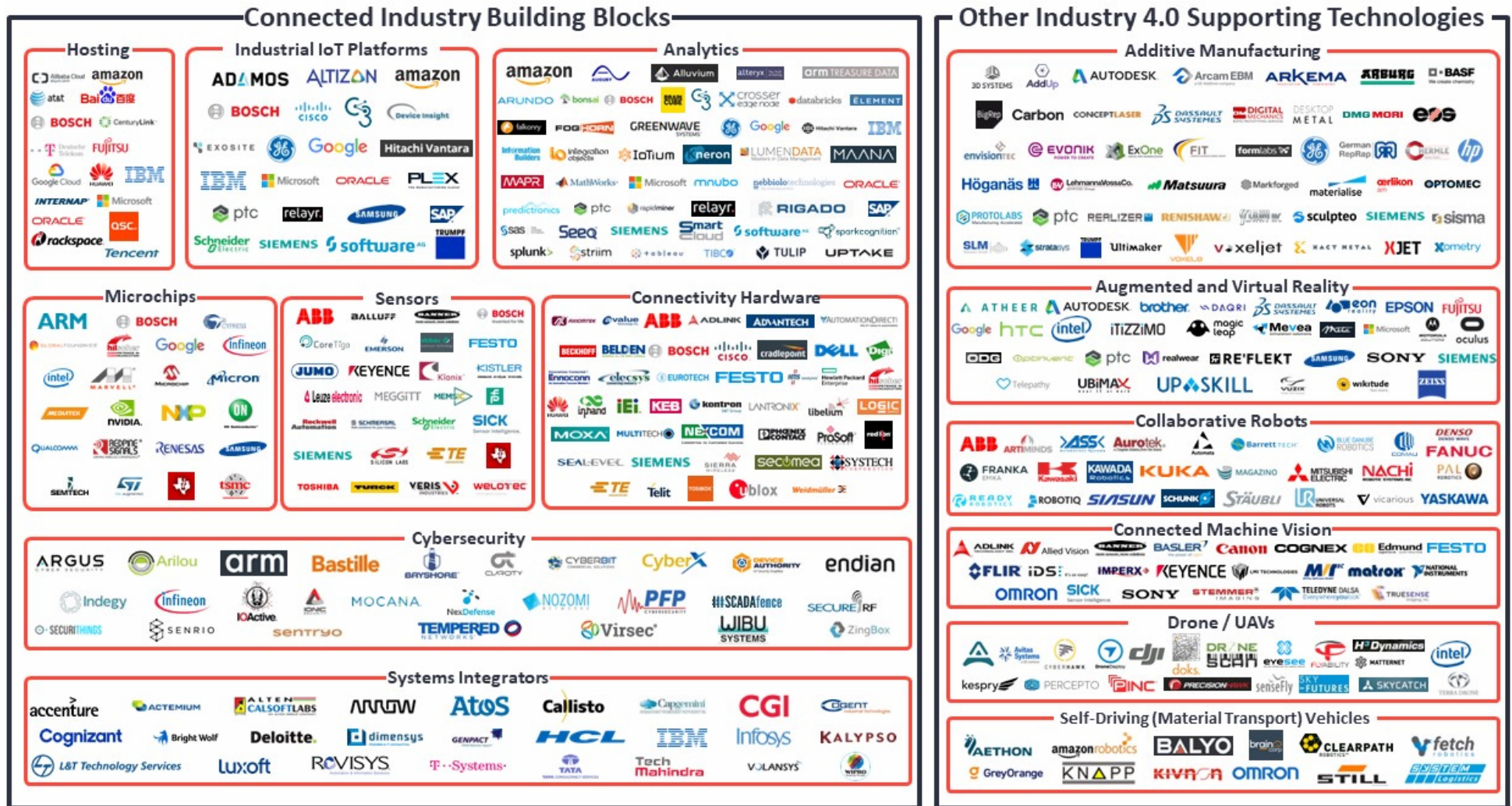


Рисунок 1.1 - Лідери по впровадженню і використанню технологій Industry 4.0 у 2019 році

## 1.2 Asus Tinker Board

**Asus Tinker Board 2Гб** - являє собою одноплатний комп'ютер в ультра-малому форм-факторі, який пропонує досить високу продуктивність в своєму класі. Плата має на борту потужний ARM Cortex-A17 з тактовою частотою 1,8 ГГц, гігабітний Ethernet, порт HDMI 2.0 для моніторів і телевізорів з роздільною здатністю до 4К і H.264, 2 Гб оперативної пам'яті, звук 192К / 24 біта, SDIO 3.0 і ще кілька бонусів. Крім того до плати можна підключити різні модулі розширення, які є у продажу. Загальний вигляд Asus Tinker Board 2Гб представлений на рисунку 1.2, а основні характеристики в таблиці 1.1

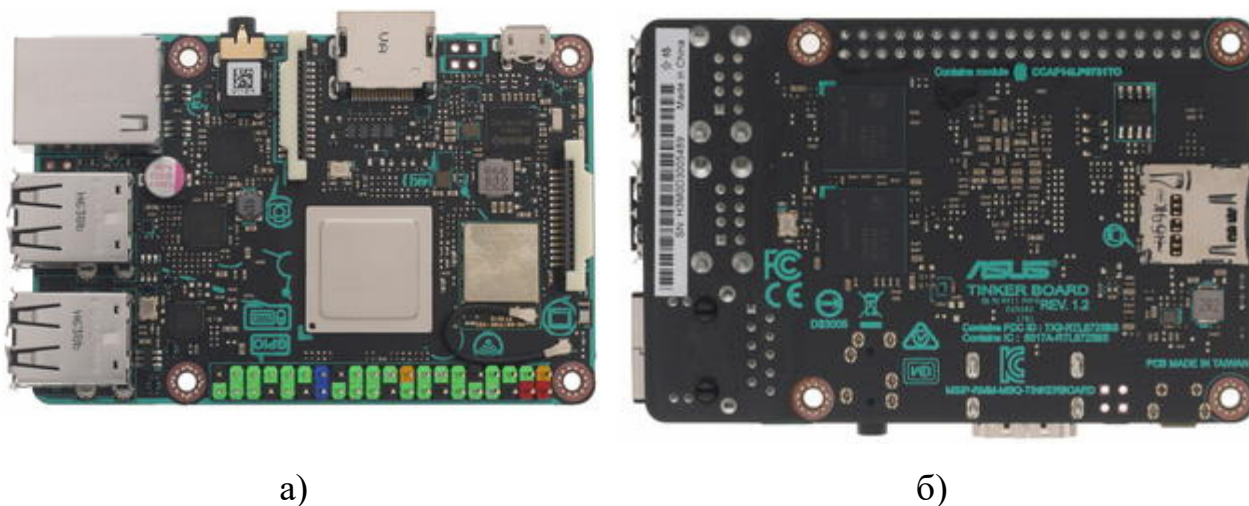


Рисунок 1.2 - Мікрокомп'ютер ASUS Tinker Board

Основні характеристики ASUS Tinker Board представлені в таблиці

Таблиця 1.1 - Основні характеристики ASUS Tinker Board[49-50]

Параметри	Значення
1	2
Процесор	
Виробник процесора	Rockchip

Продовження таблиці 1.1

1	2
Модель процесора	Rockchip RK3288
Кількість ядер процесора	4
Частота процесора	1800 МГц
Оперативна пам'ять	
Тип оперативної пам'яті	LPDDR3
Розмір оперативної пам'яті	2 Гб
Накопичувачі даних	
Об'єм вбудованої флеш-пам'яті	немає
Відеокарта	
Виробник відеочипа	Mali
Модель відеочипа	Mali T764
Обсяг відеопам'яті	виділяється з оперативної
Інтерфейси, роз'єми	
Відео інтерфейси	HDMI
Аудіо інтерфейси	3.5 мм jack (аудіо)
Інтерфейси периферії	display serial interface, camera serial interface, 40 pin GPIO, USB 2.0 x4
Додаткове вбудоване обладнання	Bluetooth 4.0
Особливості	HDMI 2.0 з підтримкою 4К, слот Micro SD з підтримкою UHS-I
Комунікації	
Вид доступу в Інтернет	Wi-Fi, Ethernet (RJ-45)
Швидкість мережевого адаптера	1000 Мбіт/с
Габарити (Д*Ш)	85.6 мм * 54 мм

### **Переваги:**

- в порівнянні з іншими одноплатними комп'ютерами Orang Pi, Bana Pi і т.п. тут є драйвера, працює апаратне прискорення відео. Open GL ES краще ніж нічого. Простий доступ до GPIO за допомогою WiringPi.

- по залізу добре збалансований, тобто швидкодія всіх компонентів, а не окремих елементів, картридер читає 65м / с, ОЗУ працює в двоканальному режимі, мережа працює набагато швидше ніж у Raspberry Pi, GPU 80Gflops майже як на ноутбуці 2010 року, на Raspberry Pi тільки 10Gflops. CPU витримує заявлену чистоту, і при цьому не поганий звук при наявності розв'язки з колонкою.

- завантаження стандартної TinkerOS кілька секунд (з флешки на 65мб / с відповідно) теж саме компіляція невеликих скетчів на QT 2-3с. YouTube в HD можна дивитися з браузера (без фризів) думаю з охолодженням можна і FULL HD, а так може іноді пригальмовувати. VLC плеєр 4K відео заявлені в характеристиках ASUS Tinker Board відпрацьовує на мінімальних FPS.

- 15- контактний роз'єм DSI и CSI

### **Недоліки:**

- гріється до 60-70град тому throttling (механізм захисту процесора від перегріву) ріже частоту під навантаженням, штатний радіатор краще замінити на модуль активного охолодження;

- живлення потрібне якісне 2-3А, шнур короткий інакше можливе самовільне перезавантаження;

- активну систему охолодження жити через GPIO не бажаном

- драйвера на TinkerOS Debian дуже «чутливі» в наслідок чого оновлювати образ ОС некваліфікованими користувачами (apt get) Не рекомендується;

- TinkerOS Android потребує передвстановлений Google Play Store;

- підтримка API GPIO все ще знаходиться в «зародковому» стані.

### 1.3 Raspberry Pi Foundation

**Raspberry Pi 3 Model A+ 512MB** - міні-комп'ютер з процесором ARM на якому може працювати операційна система Linux. Raspberry Pi 3 model A + 512Mb є міні-комп'ютер з 512 МБ ОЗУ, HDMI і аудіовиходом з композитним відеосигналом RCA (через гніздо 3,5 мм аудіовиходу), один порт USB і 40-контактним роз'ємом з кроком 2,54мм, який забезпечує доступ до входів і виходів загального призначення (GPIO). Модель А Raspberry Pi менше і дешевше, ніж модель В Raspberry Pi. На платі відсутній Ethernet інтерфейс але додані бездротові інтерфейси Wi-Fi і Bluetooth. Для запуску комп'ютера потрібна карта microSD з операційною системою і блок живлення (не входять в комплект). Мінікомп'ютер Raspberry Pi користується великою популярністю і величезним співтовариством шанувальників. Безліч різноманітних прикладів проектів та інформації доступні в Інтернеті у вільному доступі. Загальний вигляд Raspberry Pi Model A + 512MB представлений на рисунку 1.3.

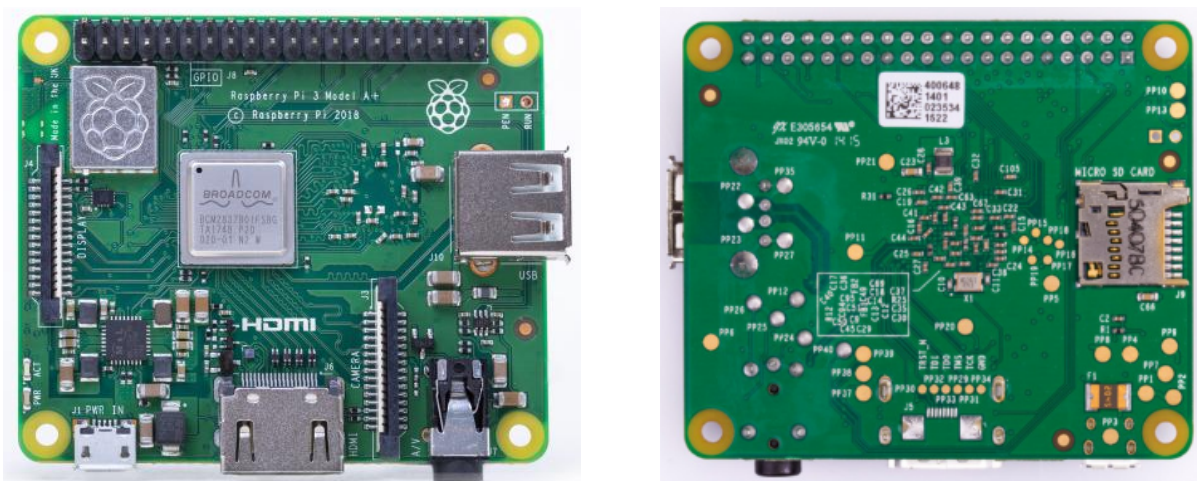


Рисунок 1.3 - Raspberry Pi Model A+ 512MB

Плата відрізняється малим споживанням і скромними розмірами, що робить її незамінною для створення мобільних пристроїв з автономним живленням.[51-52]

Основні характеристики Raspberry Pi Model A + 512MB представлені нижче:

- процесор BCM2837B0 64-розрядний Cortex-A53 з тактовою частотою 1.4GHz;
- графічний процесор VideoCore IV;
- ОЗУ 512 МБ;
- пам'ять програм: micro-SD карта пам'яті;
- один USB-порт 2.0;
- повнорозмірний вихід HDMI;
- чотирьох-контактний роз'єм 3,5 мм з аудіовиходом і композитним відеовиходом (PAL або NTSC) ;
- вбудований двохдіапазонний Wi-Fi (2.4GHz / 5GHz) стандарту IEEE 802.11 b / g / n / ac (чіп Cypress CYW43455) ;
- вбудований Bluetooth 4.2 LE (чіп Cypress CYW43455) ;
- 40-контактний GPIO-роз'єм з 0,1-дюймовими штирями, які сумісні з роз'ємами 2 x 20;
- інтерфейс камери (CSI) ;
- інтерфейс дисплею (DSI) ;
- слот для карт Micro SD;
- напруга живлення: 5В;
- напруга логічних рівнів: 3,3В;
- споживаний струм: до 2,5А.

**Raspberry Pi 3 Model B+** - от Raspberry Pi Foundation. Нова плата помітно краща за попередню, зібрана на більш потужному 4х-ядерному 64-бітному SoC Broadcom BCM2837B0 процесорі, що працює на частоті 1.4ГГц. Так само компанія оновила бездротові адаптери: Wi-Fi став двох-діапазонним стандартом IEEE 802.11ac, а Bluetooth - 4.2 BLE. Гігабітний мережевий адаптер Gigabit Ethernet забезпечує передачу даних до 300 Мб в секунду. Плата виглядає більш "закінчено" і вже не нагадує інженерний зразок. Оперативної пам'яті так і залишилося 1Гб, але збільшилася частота

процесора, швидкість мережевих інтерфейсів і додалася функція мережевого завантаження операційної системи. Загальний вигляд Raspberry Pi 3 Model B+ представлений на рисунку 1.4.[53-54]

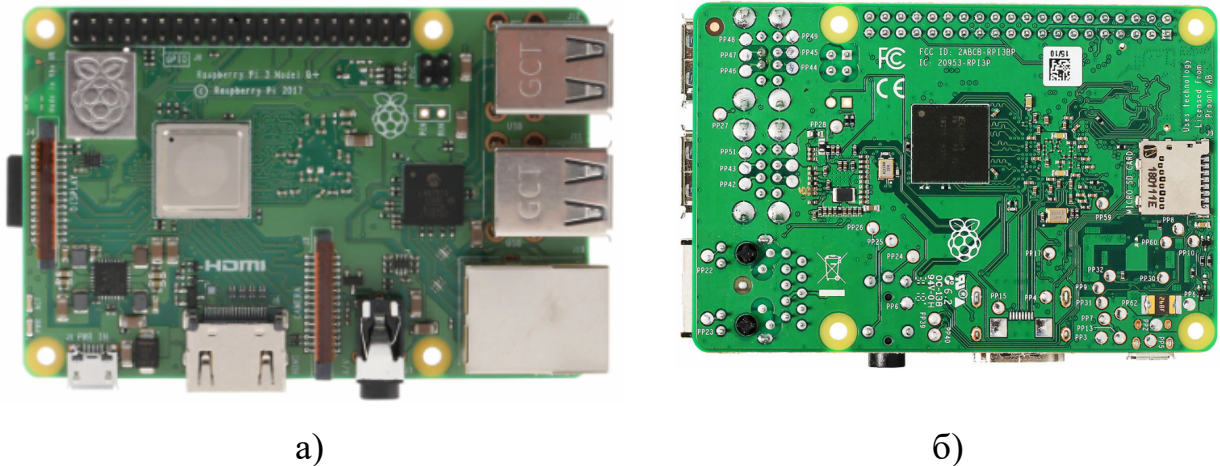


Рисунок 1.4 - Загальний вид Raspberry Pi 3 Model B+

*Для живлення Raspberry обов'язково потрібно застосовувати тільки якісний блок живлення 5В 3А! Використання неякісних блоків живлення або зарядних пристроїв мобільних телефонів призводить до виходу з ладу внутрішнього джерела живлення і не є гарантійним випадком! Максимально значення вхідної напруги живлення 5,5В. При вхідній напрузі 6В (навіть імпульсному) мікросхема перетворювача напруги гарантовано вийде з ладу. До подібного результату призведе замикання джерела напруги 3,3 В на землю.*



Основні відмінності від попередньої моделі:

- металева кришка з логотипом Raspberry Pi, що закриває модуль бездротової передачі даних;
- металевий радіатор на процесорі для поліпшеної тепловіддачі;
- 4-піновий роз'єм для підключення PoE.



*Процесор Raspberry Pi 3 B + при виконанні навіть не дуже важких обчислювальних задач сильно гріється. Забороняється використовувати міні-комп'ютер без додаткового активного охолодження.*

Характеристики:

- процесор: SoC Broadcom серії BCM2837B0 (чотирьохядерний ARM Cortex-A53 1.4GHz з 64-бітної архітектурою);
- оперативна пам'ять: 1 Гб;
- порти USB: 4 USB 2.0 порта;
- Ethernet: Гігабітний;
- підтримка живлення по мережі: PoE;
- GPIO: Стандартний 40-піновий роз'єм (з них 26 GPIO);
- роз'єм живлення: MicroUSB;
- двохдіапазонний Wi-Fi (2.4GHz и 5GHz) 802.11ac;
- Bluetooth 4.1 (Classic и Low Energy);
- інтерфейс монітору: HDMI;
- інтерфейс для CSI камери;
- DSI-конектор для LCD дисплею;
- 3.5 мм роз'єм для виведення звуку і композитного відео;
- 2-pin роз'єм для апаратного скидання;
- Micro SD Слот.

Основні відмінності від попередньої моделі:

- збільшена тактова частота процесора (до 1,4 ГГц) ;
- підтримка PoE через окремий роз'єм;
- гігабітний Ethernet
- покращений двохдіапазонний WiFi модуль;
- підтримка PXE (Preboot eXecution Environment) - завантаження ОС через мережевий інтерфейс використання локальних носіїв даних;

- поліпшене керування температурою і потужністю (на базі MaxLinear MxL7704) ;
- поліпшене пасивне охолодження процесора;
- бездротовий модуль, виконаний відповідно до стандарту FCC і захищений металевією кришкою.[54]

**Raspberry Pi A + 256Mb** - це спрощена версія моделі Raspberry Pi Model B +, у якій відсутній порт Ethernet, є тільки один порт USB і 256 мегабайтів оперативної пам'яті. Ці спрощення дозволили ще більше здешевити плату. Маленьке енергоспоживання робить її ідеальною для застосування в проектах, що вимагають від батарей, акумуляторів або сонячних елементів - роботи, сенсори, вилучені застосування або, наприклад, в якості WiFi-репитера. Загальний вигляд Raspberry Pi A + 256Mb представлений на рисунку 1.5.

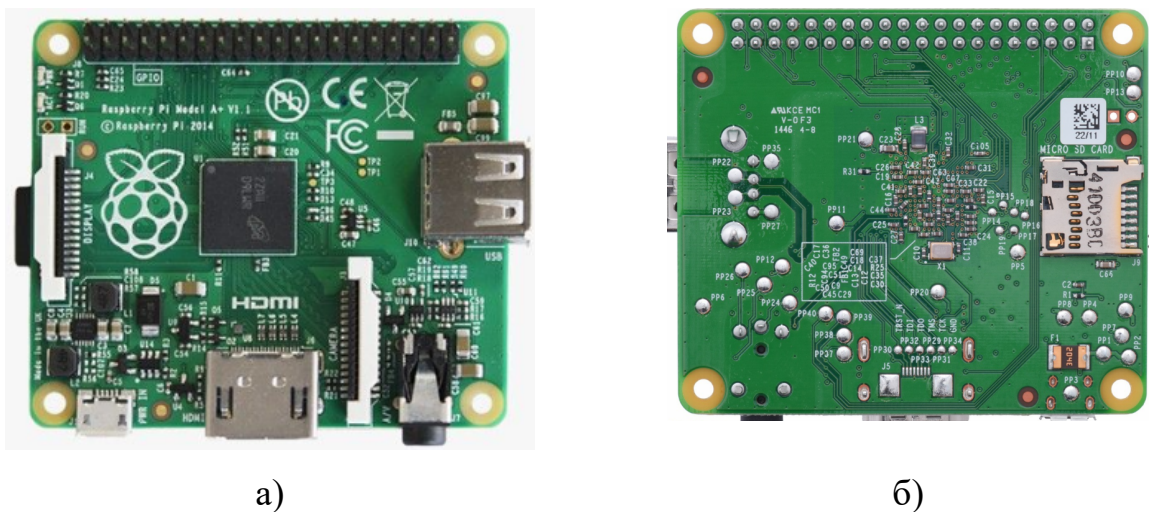


Рисунок 1.5 - Загальний вигляд Raspberry Pi A+ 256Mb

Основні характеристики представлені нижче:

- процесор 700MHz Broadcom BCM2835 з ОЗУ 256 Мб;
- 40-піновий роз'єм GPIO;
- 1 порт USB 2.0;

- комбінований 4-контактний роз'єм для стерео-аудіовиходу і композитного відео;

- HDMI повного формату;

- порт камери CSI для підключення камери Raspberry Pi;

- цифровий порт дисплея DSI для підключення сенсорного дисплея Raspberry Pi;

- роз'єм Micro SD для завантаження ОС і зберігання даних;

- роз'єм живлення Micro USB;

Основні відмінності від Model A:

- ефективніше енергоспоживання;

- поліпшена розводка плати з підтримкою розширеного 40-Голковий GPIO;

- порт Micro SD захищає вашу карту від пошкоджень;

- можливість підключення багатоспоживаючих USB пристроїв.

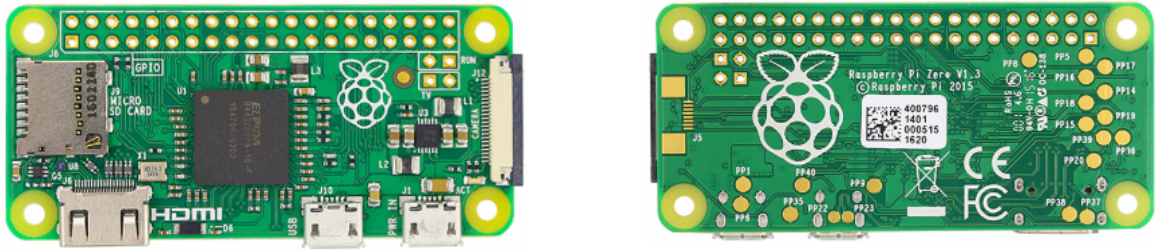
Покращення в Raspberry PI Model A +:

- менший розмір плати і повна сумісність з розширеннями НАТ 40-контактний GPIO;

- можливість підключення камери і сенсорного дисплея;

- потокове відео високої роздільної здатності (1080p).[55-56]

**Raspberry Pi Zero V1.3** - ідеально підходить для створення вбудованих інтернет-проектів Things (IoT). На платі Pi Zero V1.3 компактно розміщені основні міні-роз'єми і нерозпаяні контакти GPIO, що дозволяють використовувати тільки те, що вимагає Ваш проект. В оновленій версії (1.3) додали роз'єм CSI для підключення Pi-відеокамер.[57-58] Загальний вигляд представлений на рисунку 1.6



а)

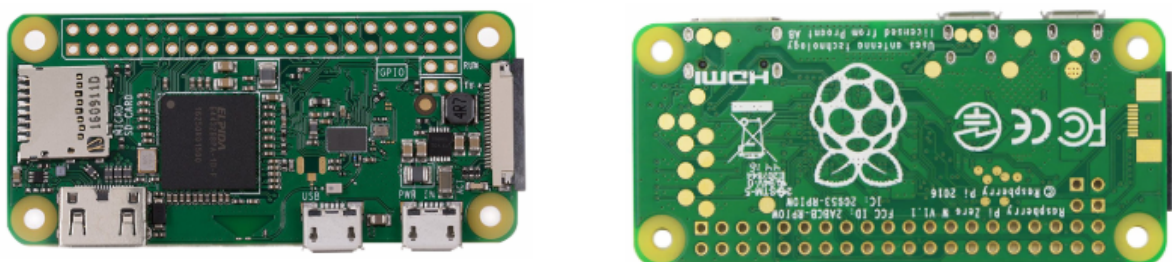
б)

Рисунок 1.6 – Загальний вид Raspberry Pi Zero V1.3

Основні характеристики:

- CPU: 1 ГГц процесор ARM11 на чіпі Broadcom BCM2835;
- ОЗУ: 512 МБ DDR2 ELPIDA;
- зовнішня пам'ять: слот карт пам'яті micro-Sd;
- міні-HDMI: з вихідним дозволом до 1080@p60;
- Usb-Порт: micro-usb з підтримкою On-The-Go (OTG) ;
- живлення: рознімання USB micro-b для джерела на 5В/2А;
- GPIO: 40-контактне рознімання;
- додаткові рознімання: композитне відео, камера CSI.

**Raspberry Pi Zero W V1.1** – був розроблений як можна більш гнучким і компактним із гніздами-з'єднувачами-міні-розніманнями й нерозпаяним 40-контактним GPIO, що дозволяють використовувати тільки те, що вимагає Ваш проект. [59] Загальний вид представлено на рисунку 1.7



а)

б)

Рисунок 1.7 - Загальний вид Pi Zero W V1.1

Основні характеристики: [60]

- CPU: 1 ГГц процесор ARM11 на чіпі Broadcom BCM2835;

- ОЗУ: 512 МБ DDR2 ELPIDA;
- зовнішня пам'ять: слот карт пам'яті micro-SD;
- міні-HDMI: з вихідним дозволом до 1080@p60;
- USB-Порт: micro-USB з підтримкою On-The-Go (OTG) ;
- живлення: рознімання USB micro-b для джерела на 5В/2А;
- GPIO: 40-контактне рознімання;
- доп.рознімання: композитне відео, камера CSI;
- бездротова мережа: Wi-Fi 802.11 b/g/n;
- Bluetooth: BLE 4.1 на чипі Cypress CYW43438.



*Робота Raspberry Pi Zero W V1.1 і Raspberry Pi Zero V1.3 виходить небагато складніше через нестандартні рознімання. Для початку знадобиться кабель Mini HDMI-HDMI або адаптер для підключення дисплея. Також знадобиться кабель USB OTG для Usb-Обладнань, і ще спеціальний кабель для камери CSI*

У додатку А приведена зведена таблиця основних параметрів одноплатних комп'ютерів сімейства Raspberry, за допомогою якої можна порівняти і вибрати не обходжену модифікацію для розроблювального проекту.

## 1.4 Orange Pi

**Orange Pi Win Plus A64 Quad-core 2GB WIFI** - основна особливість якої - готовність до установки операційної системи Windows 10 Iot на додаток до вже звичних Android, Armbian, Ubuntu, Debian і Raspbian. Як і кожне наступне поповнення комп'ютерів плата доповнена могутнішим процесором Soc Allwinner A64 з 4 64-бітними ядрами ARM Cortex-A53 і GPU Mali-400MP2, розширеним до 2Гб обсягом оперативної пам'яті DDR3 і гнучкою системою живлення, що дозволяє підключати батарейний відсік для забезпечення максимальної автономності.

Головна відмінність міні-комп'ютера Orange Pi Win Plus від Orange Pi Win полягає в збільшеному до 2Гб обсязі оперативної пам'яті. На платі розведене місце для модуля емтс пам'яті, але сам модуль на платі не розпаяний. У якості пам'яті зберігання програм використовується microsd карта пам'яті з максимальним обсягом до 64Гб.

Для вивіду зображення можна використовувати як MIPI DSI рознімання для підключення ЖК панелі так і вихід HDMI 1.4, а для підключення камери дозволом до 5 Мп призначене рознімання MIPI CSI.

У якості мережних інтерфейсів доступні: мережний порт Gigabit Ethernet, контролер Ампак AP6212 з підтримкою Wi-Fi 802.11n і Bluetooth 4.2. Для підключення додаткових обладнань можна використовувати один microusb порт із підтримкою OTG і чотири порти USB 2.0. 40-контактний GPIO рознімання сумісне з Raspberry Pi і три UART (один з них використовується для налагодження), ІК-приймач, кнопки живлення й перезавантаження. Окремий джампер відповідає за вибір джерела живлення: порт micro-USB OTG або зовнішній адаптер з вихідною напругою 5 В, 2 А. [61-62] Orange Pi Win Plus A64 Quad-core 2GB WIFI представлено на рисунку 1.8

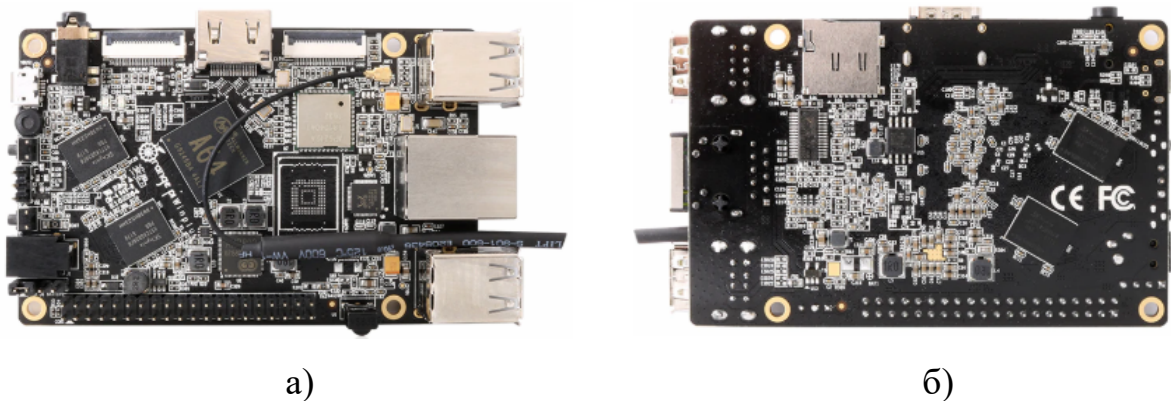
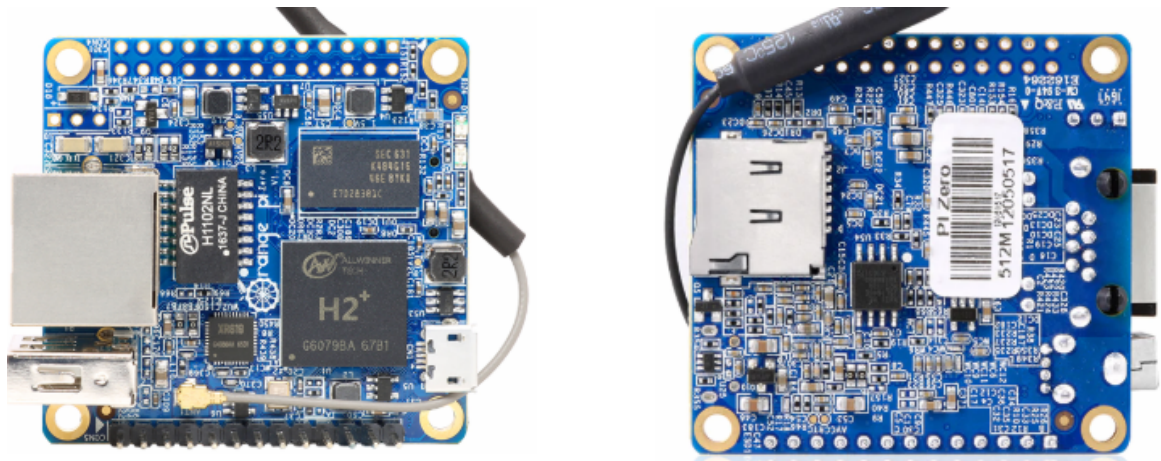


Рисунок 1.8 - Orange Pi Win Plus A64 Quad-core 2GB WIFI

Основні характеристик Orange Pi Win Plus A64 Quad-core 2GB WIFI представлені нижче:

- процесор: Allwinner A64 Quad-core, сімейство процесора: Cortex A53, розрядність процесора: 64-bit, робоча частота: 1,2ГГц - 1,6ГГц;
- графічний процесор: Mali400MP2 GPU, підтримка: Opengl ES 2.0 Opengl 1x, DX9\_3, Openvg, 1080p45 H.264, апаратна підтримка декодування: H.265/HEVC 4K;
- оперативна пам'ять: 2 ГБ на Win Plus DDR3 ( разом з GPU) ;
- пам'ять програм: microsd ( до 64 ГБ), MMC слот, 2M Spi Flash;
- мережні інтерфейси: Ethernet 10/100/1000 Мб/сек Ethernet RJ45, Wi-Fi+Bluetooth Ампак AP6212, IEEE 802.11 b/g/n + Bluetooth 4.2;
- відеовхід: Роз'єм 1 x CSI;
- відеовихід: рознімання HDMI 1.4 з підтримкою HDCP і CEC, рознімання MIPI LCD;
- аудіовхід: вбудований мікрофон;
- аудіовихід: HDMI, mini-jack (3,5 мм) ;
- USB інтерфейси: USB 2.0 x 4 порту, USB 2.0 OTG x 1 порт;
- інтерфейс дистанційного керування: ІК-Порт, інтерфейс уведення-вивід: 40-контактний GPIO сумісний з Raspberry Pi GPIO, UART, I2C, SPI або PWM, UART Інтерфейс (Debug TTL UART) ;
- живлення: напруга живлення: 5В, споживаний струм: 2А;
- тип рознімання живлення: 4.0 мм/1.7 мм.

**Orange Pi Zero 256Mb** - це мініатюрний одноплатний комп'ютер на процесорі Allwinner H2+, у який входять чотири обчислювальні ядра Cortex A7 з тактовою частотою до 1,2 ГГц. з відкритим вихідним кодом. Він може працювати під керуванням Android 4.4, Ubuntu, Debian. Він має 256 МБ DDR3 оперативної пам'яті SDRAM. Також цей міні-комп'ютер оснащений графічним прискорювачем Mali-400MP2, який дозволяє декодувати 4 К-відео формату H.265/HEVC із частотою відтворення до 30 кадрів у секунду й підтримує microSD карти пам'яті ємністю до 64 гігабайт.[63] Загальний вид Orange Pi Zero 256Mb представлено на рисунку 1.9.



а)

б)

Рисунок 1.9 - Загальний вид Orange Pi Zero 256Mb

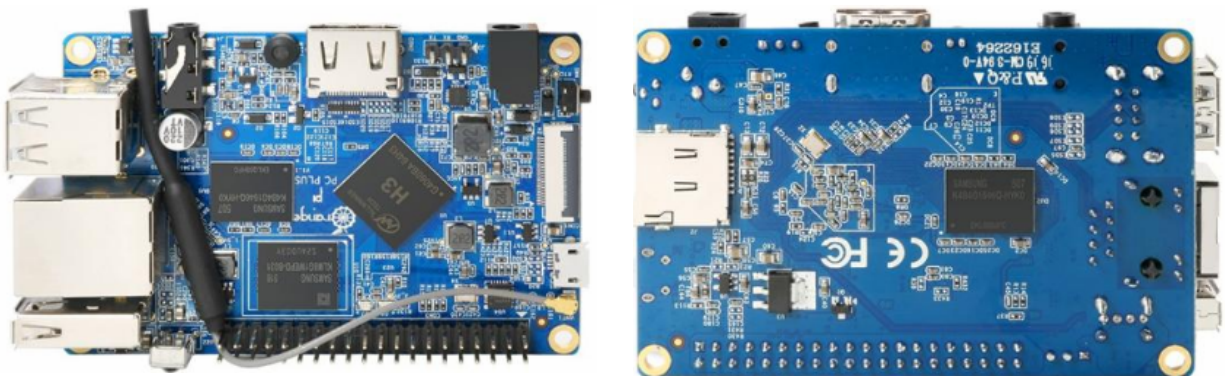
Основні характеристик Orange Pi Zero 256Mb представлені нижче:

- процесор: Soc: Allwinner H2+, Quad-Core Cortex A7@1,2 ГГц;
- графічний процесор: GPU: Mali400MP2(5)600Mhz, Підтримка OpenGL ES2.0;
- оперативна пам'ять: DDR3 RAM: 256M;
- пам'ять програм: microSD;
- мережний інтерфейс: 10/100M Ethernet port;
- USB порти: Host 1x USB 2.0 і 2 порту на додатковій розніманні розширення, Micro USB порт: OTG порт, передача даних і вхід живлення;
- відео вихід: композитний;
- інтерфейс камери: відсутній;
- налагоджувальний послідовний порт: 3-х контактний із кроком 2.54ммv
- GPIO входи/виходи: 2.54мм 24-контактний і 12-контактний рознімання, включаючи USB, UART, SPI, I2C, PWM і т.д. ;
- напруга живлення: 5В;
- максимальний споживаний струм: 2А.[64]



*Міні-комп'ютер **Orange Pi Zero** відмінно підходить для побудови систем, що вбудовуються, керування й систем контролю. Підтримує наступні операційні системи: Ubuntu MATE, Debian, Armbian і інші. Живиться комп'ютер від блоку живлення 5В 2А через microUSB рознімання.*

**Orange Pi PC Plus** - це багатофункціональний і потужний одноплатний міні-комп'ютер з відкритим вихідним кодом. Він використовує Allwinner H3 SOC, і має 1 ГБ DDR3 SDRAM. Він може працювати під керуванням Android 4.4, Ubuntu, Debian, Raspberry OS. На платі комп'ютера є 8Гб flash EMMC пам'яті програм і слот для підключення microSD карти пам'яті. Відмінною рисою комп'ютера є те, що при збереженні функціональності й сумісності вийшов комп'ютер дуже малих розмірів. Міні-комп'ютер дуже вимогливий до якості живлення, тому рекомендуємо ретельно підійти до питання живлення й здобувати якісні й надійні блоки живлення.[65] Загальний вид Orange Pi PC Plus представлено на рисунку 1.10



а)

б)

Рисунок 1.10 - Загальний вид Orange Pi PC Plus

Основні характеристики Orange Pi PC Plus:[66]

- процесор: H3 чотирьохядерний процесор Cortex-A7 H.265/HEVC 4 ДО 1.6Ггц;
- графічний процесор: Mali400MP2 GPU @ 600 Мгц із підтримкою Opengl ES 2.0;

- оперативна пам'ять: SDRAM 1 ГБ DDR3 (використовується разом з GPU)
- вбудована пам'ять програм: 8 ГБ EMMC Flash пам'ять;
- зовнішня пам'ять програм: TF карта (Макс. 64 ГБ) ;
- мережний інтерфейс: 10/100/1000 М Ethernet RJ45;
- бездротовий інтерфейс: WI-FI Realtek RTL8189ETV, IEEE 802.11 b/g/n;
- відео вхід: CSI рознімання Камери з підтримкою 8-бітних YUV422 CMOS камер CCIR656 протоколу для NTSC і PAL;
- дозвіл відео-захоплення: до 1080p @ 30fps;
- аудіо вхід: вбудований мікрофон;
- відео вихід: HDMI з HDCP і підтримкою HDMI CEC і інтегрований інтерфейс CVBS. Виходи можна використовувати одночасно;
- аудіо вихід: 3.5 мм рознімання й HDMI;
- USB інтерфейс: три USB 2.0 HOST порту й один USB 2.0 OTG;
- низкорівнева периферія: 40 Pins GPIO інтерфейс, сумісний з Raspberry Pi B +.

Підтримувані OS: Android, Lubuntu, Debian, Raspbian

## 1.5 LattePanda

**LattePanda 2GB/32GB** - це повнофункціональний одноплатний комп'ютер під керуванням операційної системи Windows 10 на 4-х ядерному процесорі Intel Cherry Trail Z8350 з обсягом оперативної пам'яті 2 ГБ і із внутрішньої emmc пам'яттю 32ГБ. Має всю необхідну периферію що й звичайний ПК і сполучимо практично з усіма розповсюдженими обладнаннями: принтери, джойстики, камери й багато чого іншого. Будь-які периферійні обладнання, які працюють на ПК, будуть працювати на LattePanda. LattePanda поставляється з попередньо встановленим з повним виданням Windows 10 Home Edition, що дозволяє запускати потужні

інструмент, такі як Visual Studio, Nodejs, Java, обробка й багато інших. LattePanda унікальний тим, що для зв'язку із зовнішнім миром комп'ютер містить інтегрований контролер Arduino Leonardo. Мінікомп'ютер знайде широке застосування в портативних системах, що вбудовуються, системах вилученого збору інформації й керування, 3D принтерах і т.д.[67] Загальний вид одноплатного ПК LattePanda 2GB/32GB представлено на рисунку 1.11

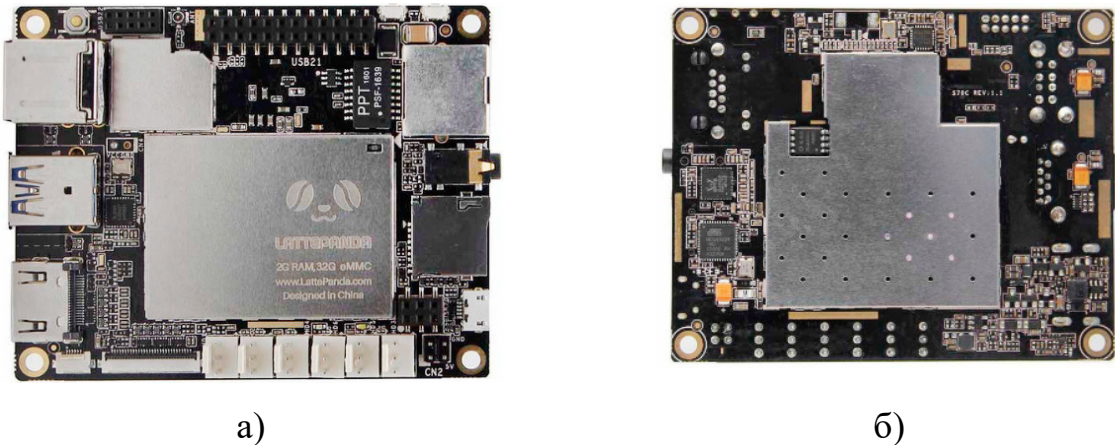


Рисунок 1.11 - Загальний вид одноплатного ПК LattePanda 2GB/32GB

Основні характеристики LattePanda 2GB/32GB:[68]

- процесор: Intel Cherry Trail Z8350 Quad Core;
- базова частота процесора: 1.44ГГц (1.92Ghz максимальна) ;
- операційна система: Windows 10 Home Edition (неактивована) ;
- оперативна пам'ять: 2GB DDR3L;
- еммс пам'ять: 32GB;
- графічний прискорювач Intel HD Graphics, 12 Eus @200-500Mhz;
- USB інтерфейси: USB 3.0 x 1, USB 2.0 x 2;
- бездротовий інтерфес: Wi-Fi 802.11n 2.4G;
- Bluetooth: 4.0;
- Arduino інтерфейс: Atmega32u4 (Arduino Leonardo) ;
- відео вихід: HDMI і MIPI-DSI;
- рознімання підключення сенсорною панеллю;
- Ethernet: 100Mbps;
- GPIO входи/виходи;

- GPIO процесора Intel x 6;
- GPIO контролера Atmega32U4 x 20;
- інтерфейсні рознімання x 6;
- напруга живлення: 5В 2А - рекомендуємо даний блок живленням
- розміри плати: 88 x 70мм;
- вага: 55g.

## 1.6 Odroid

**ODROID-XU4Q** - це покоління обчислювальних обладнань із могутнішим, більш енергоефективним устаткуванням і меншим форм-фактором. Тепер це ще й самий тихий продуктивний міні-комп'ютер. Пропонуючи підтримку з відкритим вихідним кодом, плата може працювати з різними версіями Linux, включаючи останню версію Ubuntu 16.04 і Android 4.4 Kitkat, 5.0 Lollipop і 7.1 Nougat. Завдяки реалізації інтерфейсів EMMC 5.0, USB 3.0 і Gigabit Ethernet, ODROID-XU4 може похвалитися неймовірною швидкістю передачі даних – функцією, яка усе більше потрібна для підтримки поліпшеної обчислювальної потужності на обладнаннях ARM. Це дозволяє користувачам по-справжньому випробувати новинки в області обчислювальної техніки, особливо з більш швидким завантаженням, переглядом веб-сторінок, мережними й 3D-іграми. Дана модель є самою тихою й продуктивною у всій лінійці ODROID міні-комп'ютерів.

ODROID-XU4Q, оснащено чотирма продуктивними ядрами (ARM® Cortex® -A15™ до 2,0 ГГц) і чотирма невеликими ядрами (ARM® Cortex® -A7™ до 1,4 ГГц), забезпечує поліпшені можливості обробки при збереженні найбільш ефективною споживаною потужності. Завдяки розв'язку big.LITTLE™ HMP Exynos-5422 може використовувати максимум усі вісім ядер для керування обчислювальними завданнями. рафічений процесор ARM® Mali™-T628 MP6 пропонує повну підтримку API для OpenGL ES 1.1, OpenGL ES 2.0 і OpenGL ES 3.0, OpenCL 1.2 Full Profile і Google Renderscript. Mali-T628

– це кращий графічний процесор для використання в останніх поколіннях лідируючих на ринку обладнань, оптимізований для створення приголомшливих графічних ефектів для таких споживчих додатків, як 3D-Графіка, візуальні обчислення, доповнена реальність, генерація процедурних текстур і розпізнавання голосу.

Для завантаження ОС потрібна карта microSD або модуль EMMC. Рекомендуємо використовувати модуль emmc для більш швидкого завантаження ОС, більш швидкого запуску додатків, гладкої багатозадачності й ефективного доступу до хмари.[69-70] Загальний вид ODROID-XU4Q представлено на рисунку 1.12



Рисунок 1.12 - Загальний вид ODROID-XU4Q

Характеристик ODROID-XU4Q:

- процесори Samsung Exynos5422 Cortex™-A15 з тактовою частотою 2 ГГц і Cortex™-A7 Octa;
- Mali-T628 MP6 (повний профіль OpenGL ES 3.1 / 2.0 / 1.1 і OpenGL 1.2) ;
- 2 Гб оперативної пам'яті LPDDR3 ОЗУ (750 МГц, пропускна здатність пам'яті 12 ГБ / з, шина 2x32 біт) ;
- emmc5.0 HS400 Flash Storage. Рознімання для модуля emmc: флеш-накопичувач emmc 5.0 ( до 64 ГБ) Слот для карти Microsd ( до 128 ГБ)м
- 2 x USB 3.0 Host, 1 x USB 2.0 Host;
- порт Gigabit Ethernet 10/100/1000 Мбит/із із розніманням RJ-45 (підтримка Auto-mdix) ;

- Wi-Fi: IEEE 802.11 AC / B / G / N 1T1R WLAN з антеною (зовнішній Usb-Адаптер) ;

- HDMI 1.4a для підключення дисплея або телевізора;

- розмір: приблизно 83 x 58 x 20 мм ( без кулера) ;

- живлення: 5 В / 4 А.

Підтримувані операційні системи:

- Ubuntu 16.04 + Opengl ES + Opencil в Linux Ядро 4.14 LTS;

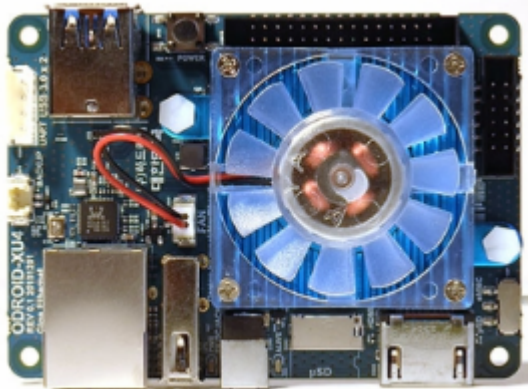
- Android 4.4.2 у ядрі LTS 3.10;

- Android 7.1 доступна в якості розробки ОСм

- розміри: 83 x 58 x 20 мм (вага: 38 грам) без кулера.

**ODROID-XU4** - побудований на процесорі ARM® big.LITTLE™ technology, із застосуванням Heterogeneous Multi-Processing (HMP) технології. ODROID-XU4 нове покоління потужних, енергоефективних міні-комп'ютерів. Підтримується програмне забезпечення з відкритим вихідним кодом. На платі можна запустити різні клони Linux, включаючи Ubuntu 15.04 і Android 4.4 Kitkat і 5.0 Lollipop.

Вбудовані EMMC 5.0, USB 3.0 і Gigabit Ethernet інтерфейс. ODROID-XU4 має дивно велику швидкість передачі даних. Користувачі по гідності оцінять можливості комп'ютера при перегляді інтернет сторінок у мережних додатках і 3D іграх.[71-72] Загальний вид ODROID-XU4 представлено на рисунку 1.13



а)



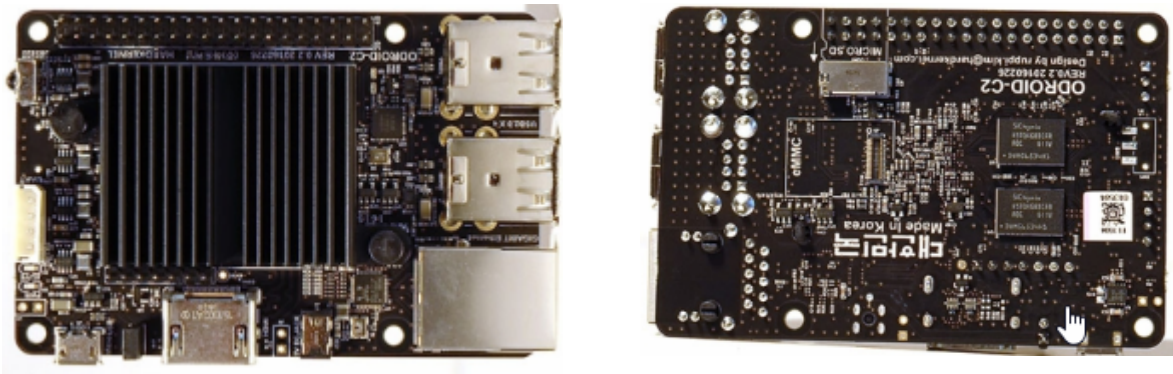
б)

Рисунок 1.13 - Загальний вид ODROID-XU4

Основні характеристики ODROID-XU4:

- центральний процесор Samsung Exynos5422 Cortex™-A15 2Ghz і 8-мі ядерний процесор Cortex™-A7;
- Mali-T628 MP6(підтримка Opengl ES 3.0/2.0/1.1 і Opencil 1.1 повний профіль)м
- 2 Гбайта LPDDR3 RAM PoP;
- флеш накопичувач EMMC5.0 HS400;
- 2 x USB 3.0 Host, 1 x USB 2.0 Host;
- гігабітний Ethernet порт;
- інтерфейс HDMI 1.4a;
- розмір: 82 x 58 x 22 мм (включаючи кулер процесора).

**ODROID-NC2** - це 64-розрядна платформа на 4-ядерному процесорі Amlogic ARM® Cortex®-A53(Armv8), що працює на частоті 1,5 ГГц. Платформа призначена для побудови високопродуктивних систем автоматизації й обробки даних, домашніх кінотеатрів, прототипування обладнань, що вбудовуються. Платформа ODROID-C2 підтримує такі операційні системи як Ubuntu, Android, Archlinux, Debian і інші портовані 64-розрядні операційні системи з відкритим вихідним кодом. У комп'ютері сполучається мале споживання, розміри й висока продуктивність із можливістю підключення зовнішніх датчиків, обладнань і виконавчих механізмів. По багатьом параметрам ODROID-C2 більш ніж в 2 рази перевершує такий популярний міні-комп'ютер як Raspberry Pi 2 B.[73-74] Загальний вид представлено на рисунку 1.14



а)

б)

Рисунок 1.14- Загальний вид представлений ODROID-C2

Основні характеристики ODROID-C2 представлені нижче:

- процесор: Amlogic ARM® Cortex®-A53(Armv8) 1,5 Ghz quad core;
  - графічний процесор: Mali™-450 GPU (3 Pixel-processors + 2 Vertex shader processors) ;
  - оперативна пам'ять: 2Gbyte DDR3 SDRAM;
  - мережний інтерфейс: 10/100/1000Mbps Ethernet with RJ-45;
  - бездротовий інтерфейс: USB IEEE 802.11b/g/n WLAN (опційно) ;
  - відео вихід: HDMI 2.0 4K/60Hz;
  - вхід камери: USB 720p (опційно) ;
  - підтримувані формати відео: H.265 4K/60FPS and H.264 4K/30FPS
- capable VPUм
- порти введення/вивіду: 40pin Gpios + 7pin I2S;
  - зберігання програм і даних: emmc5.0 HS400 і UHS-1 SDR50 microSD;
  - порти USB: USB 2.0 Host x 4, USB OTG x 1 (power + data capable) ;
  - додаткові порти: Інфрачервоний порт;
  - операційні системи: Ubuntu 16.04 або Android 5.1 Lollipop based on Kernel 3.14LTS;
  - розміри: 85 x 56 x 18 мм;
  - вага: 56 з радіатором.

## 1.7 PINE64

**PINE A64+ Board 1Гб** - 1,2 ГГц 64- розрядному чотирьохядерному процесорі ARM Cortex A53. Комп'ютер обладнано оперативною пам'яттю DDR3 обсягом 1Гбайт. Графічна підсистема посилена двухядерним графічним співпроцесором Mali 400 MP2, що дозволяє без проблем переглядати відео з дозволом до 4К.

На платі розміщено два GPIO рознімання - один Raspberry Pi2 сумісний, другий формату Euler Bus. Наявність такої кількості цифрових портів робить Pine64 гідним конкурентом Raspberry для систем, що вбудовуються, керування, де потрібна велика кількість GPIO входів/виходів. Передбачені рознімання для підключення батарейного живлення (є перемички, що перемикають живлення комп'ютера як від зовнішніх 5В так і від батарей 3.6В) комп'ютера й резервного живлення годин реального часу. До плати можна підключити дисплей із цифровою шиною, відеокамеру й сенсорну панель.[75-76] Загальний вид PINE A64+ Board 1Гб представлено на рисунку 1.15

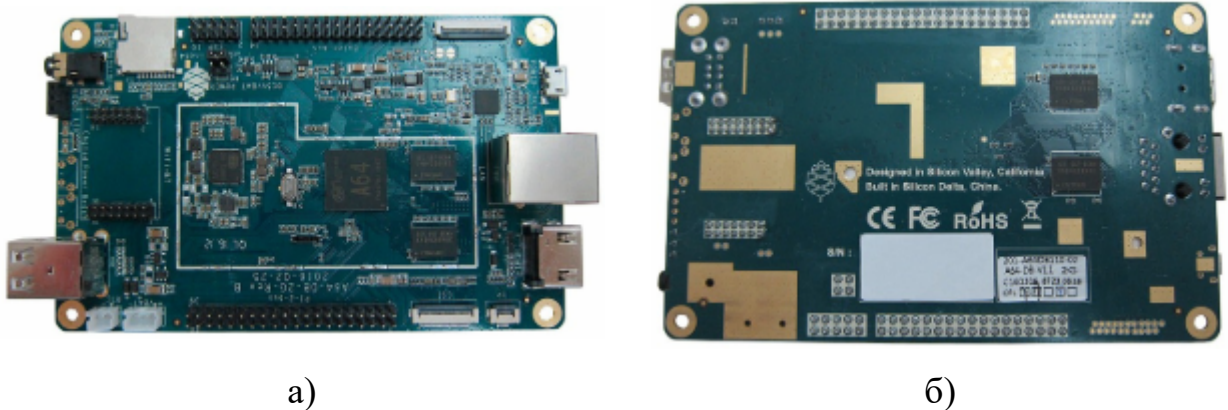


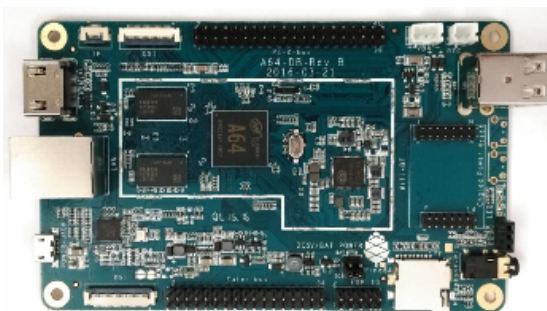
Рисунок 1.15 - PINE A64+ Board 1Гб

Основні характеристики PINE A64+ Board 1Гб:

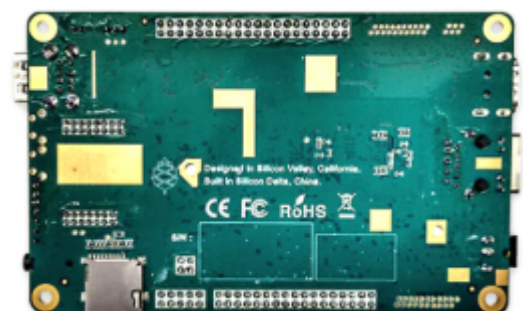
- процесор: 1.2 ГГц Quad-Core ARM Cortex A53 64-біт процесор з виконанням як 32 так 64-бітних команд;
- цифрове введення/вивід: два рознімання Pi2Bus і Euler Bus;

- графічний процесор: двухядерний Mali 400 MP2;
- оперативна пам'ять: 1 Гбайт;
- графічна підсистема: інтегрована з HDMI 1.4 виходом з якістю до 4К
- мережні обладнання: 10/100/1000 Ethernet порт;
- бездротові обладнання: опційно Bluetooth 4.0 і 802.11BGN бездротовий модуль;

**PINE A64-LTS** - комп'ютер на потужному 64-бітному процесорі Quad-Core ARM Cortex A53 від Allwinner «R18». Комп'ютер оснащений оперативною пам'яттю DDR3 обсягом 2Гбайта й завантажувальною Flash-Пам'яттю 128 МБ SPI. Графічна підсистема посилена двухядерним графічним співпроцесором Mali 400 MP2, що дозволяє без проблем переглядати відео з дозволом до 4К.[77-78] Загальний вид міні комп'ютера PINE A64-LTS представлено на рисунку 1.16



а)



б)

Рисунок 1.16 - Загальний вид PINE A64-LTS

Основні характеристики PINE A64-LTS представлені нижче:

- процесор: 64-бітного процесора Quad-Core ARM Cortex A53 від Allwinner «R18» процесор з виконанням як 32 так 64-бітних команд;
- цифрове введення/вивід: два рознімання Pi2Bus і Euler Bus;
- графічний процесор: двухядерний Mali 400 MP2;
- оперативна пам'ять: 2 Гбайтм
- графічна підсистема: інтегрована з HDMI 1.4 виходом з якістю до 4К

- мережні обладнання: 10/100/1000 Ethernet порт;
  - бездротові обладнання: опційно Bluetooth 4.0 і 802.11BGN
- бездротовий модульм
- підтримка Android, Linux, Remix OS, Windows 10 IoT;
  - живлення: 5V 2A (рознімання 3.5x1.35мм, але підходить і 4.0x1.7мм) ;
  - розміри: 133 x 80 x 19mm.

Інтерфейси:

- 2 порту USB 2.0;
- рознімання HDMI 1.4a;
- рознімання 3.5 jack;
- 3 х-контактне рознімання для підключення літєвої батареї;
- 2 х-контактне рознімання для підключення RTC батареї;
- рознімання для підключення камери, дисплея й тачскрина;
- вбудований мікрофон;
- вбудований ІК-Приймач;
- 3 набору GPIO пінів: 2x20 RPi пінів, 2x17 Euler пінів і 2x5 EXP;
- рознімання для підключення Wi-Fi або Bluetooth модуля.

## 1.8 Плати розширень, модулі

### *Arduino Expansion Shield om DFRobot (www.DFRobot.com)*

Для Arduino немає нічого простіше, чим управляти двигунами з реакцією на датчики перешкоди. Тепер переваги обох обладнань об'єднані в новому shield для Raspberry Pi. На shield виведені контакти введення/вивід в стандартному форм-факторові Arduino, що дозволяє використовувати готові шилди й плати для контролера Arduino. Додатково виведені контакти GPIO Raspberry, що дозволяє максимально використовувати ресурси як Raspberry Pi так і Arduino.

Shield підключається до Raspberri Pi за допомогою GPIO рознімання й USB перехідника й дозволяє використовувати Arduino у якості віртуального

послідовного порту, віртуальної клавіатури або віртуальної миші. На платі shield так само встановлений годинник реального часу для Raspberry на мікросхемі DS1307 з автономним живленням.[79] Загальний вид Arduino Expansion Shield від Dfrobot представлено на рисунку 1.17.

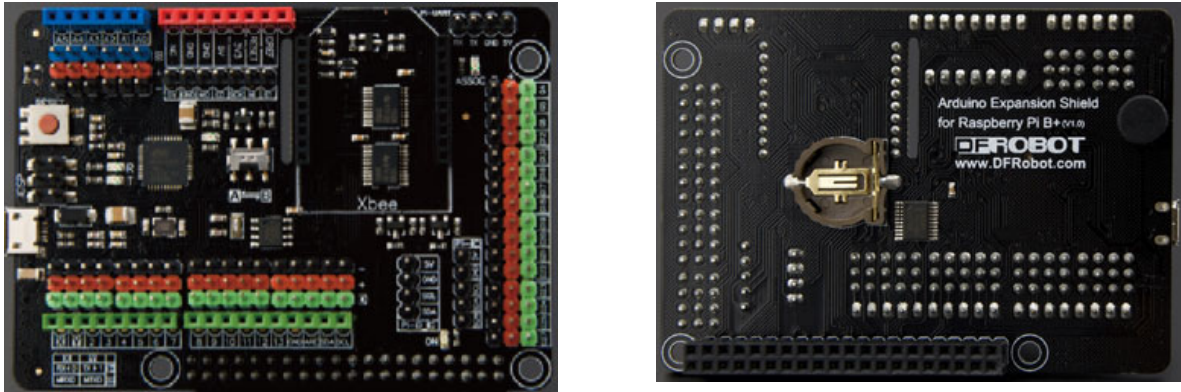


Рисунок 1.17 - Загальний вид Arduino Expansion Shield от DFRobot

Основні характеристики Arduino Expansion Shield от DFRobot:

- бортовий мікроконтролер Atmega32u4;
- Arduino-Сумісне розміщення контактів;
- сполучимо з усіма стандартними екранами й датчиками Arduino;
- підтримка стандарту інтерфейсів Gravity для UART і I<sup>2</sup>C інтерфейсів;
- напруга живлення Arduino 5 В;
- цифрових входів/виходів Arduino 20;
- аналогових входів Arduino 6;
- інтерфейси Raspberry Pi: Raspberry Pi B + GPIO(16); Raspberry Pi B + I2C(1); Raspberry Pi B + ID\_I2C(1); Raspberry Pi B + SPI(1); Raspberry Pi B + TTL UART(1).
- розмір: 86 мм x 61 мм x 26 мм.



*Роз'єм ICSP призначений тільки для внутрішньосхемного програмування мікроконтролера й не має доступу як Arduino введення/виводу. Перемикач на платі дозволяє підключити Xbee модуль як до Raspberry Pi так і до контролера Arduino.*

Розташування й призначення висновків на Arduino Expansion Shield від Dfrobot представлено на рисунку 1.18.

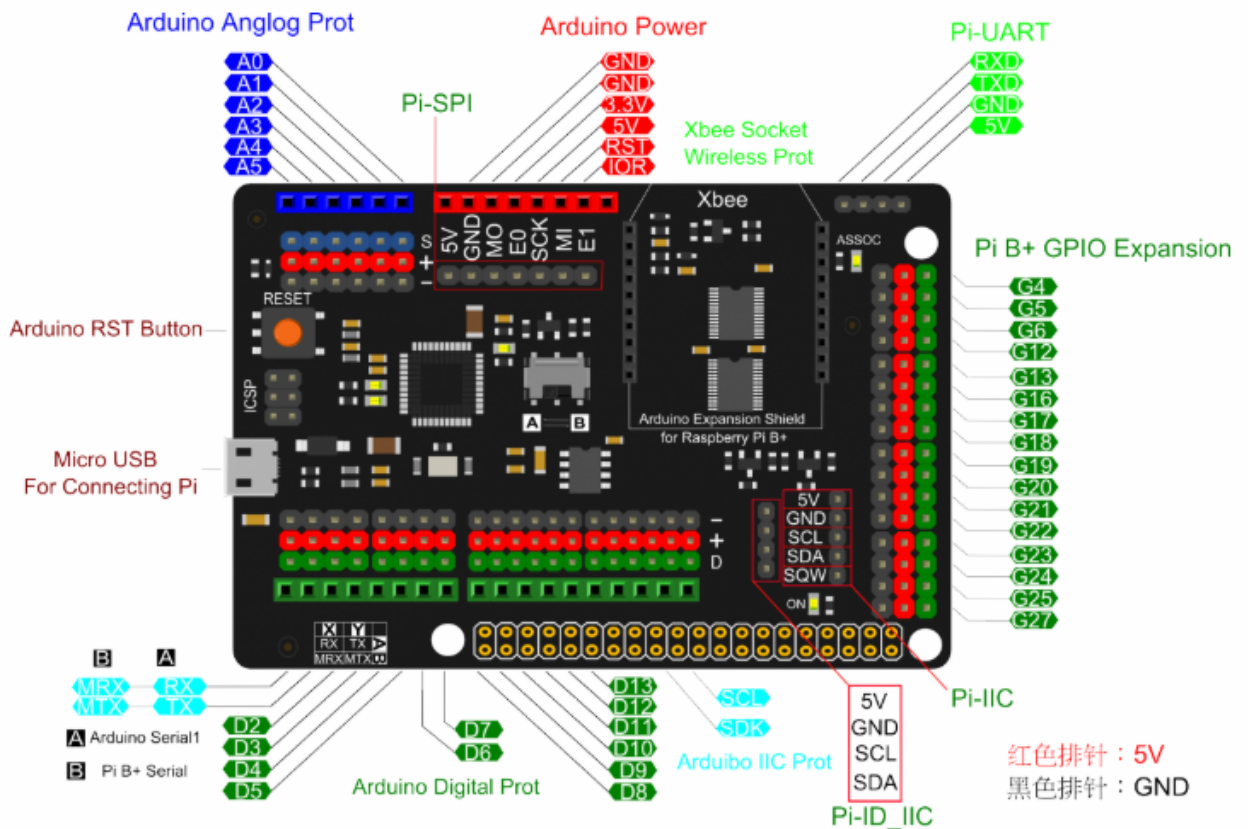


Рисунок 1.18- Розташування й призначення висновків на Arduino Expansion Shield

*I<sup>2</sup>C* модуль розширення на MCP23017 від Robotdyn ([www.robotdyn.com](http://www.robotdyn.com))

Модуль I<sup>2</sup>C адаптера графічного дисплея на мікросхемі MCP23017 призначений для спрощення підключення графічного дисплея 128x64 крапки (рекомендується на чипі KS0108) з паралельним інтерфейсом до контролера Arduino або міні-комп'ютеру Raspberry за допомогою послідовного I<sup>2</sup>C інтерфейсу. Модуль так само можна використовувати як звичайний розширник квазідвохнаправлених портів введення-вивід.

I<sup>2</sup>C адреса обладнання вибирається перемичками на платі. За замовчуванням виставлена адреса 0x20, яку, при необхідності, можна

змінити на кожній з діапазону 0x20-0x27. Дана операція вимагає застосування паяльника й навичок монтажу.

Вмикання й вимикання підсвічування контролюється так само через шину I<sup>2</sup>C. Якщо Вам потрібно управляти яскравістю підсвічування мікроконтролером, необхідно перерізати перемичку Backlight control на платі й подати керуючий ШІМ сигнал на висновок LCD+ рознімання.[80] Загальний вид модуля представлено на рисунку 1.19

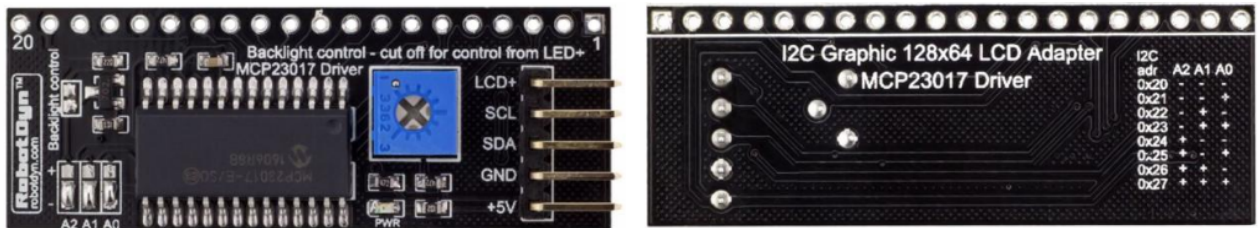


Рисунок 1.19 - Загальний вид модуля I2C адаптера графічного дисплея

Основні характеристики I2C модуль розширення на MCP23017:

- напруга живлення від 1,8В до 5,5В ( при напрузі живлення нижче 4,5В знижується яскравість підсвічування й контрастність дисплея ) ;
- максимальний споживаний струм 150 мА;
- інтерфейс I<sup>2</sup>C;
- режими роботи інтерфейсу 100кГц/400кГц/1,7МГц;
- I2C адреса обладнання від 0x20 до 0x27;
- I2C адреса за замовчуванням 0x20;
- розміри плати: 19x51 мм.

Швидкість роботи I<sup>2</sup>C модуль розширення на MCP23017:

- очищення екрана: 600 мс ( тобто 0.6 секунди ) \;
- установка курсору: 33 мс;
- печатка тексту: приблизно 4 мс/символ;
- малювання графіки: приблизно 2 мс/байт;
- очищення прямокутної області: приблизно 120 мс для області розміром 20 x 50 крапок з вирівнюванням по 8 крапок по вертикалі;

- заповнення прямокутної області 20 x 50 крапок: 5230 мс;
- малювання рамки прямокутника в одну крапку 20 x 50 крапок: 730 мс;
- малювання рамки прямокутника у дві крапки 20 x 50 pixels: 1430 мс;

Розташування й призначення висновків на I<sup>2</sup>C модуль розширення на MCP23017 від Robotdun представлено на рисунку 1.20.

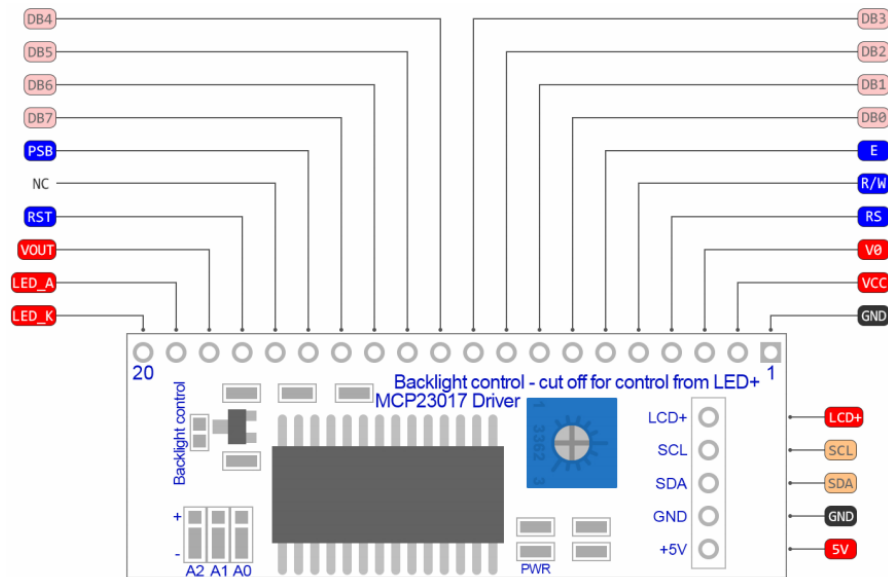


Рисунок 1.20 - Розташування й призначення висновків на I<sup>2</sup>C модуль розширення на MCP23017



*Для виводу тексту, що повільно змінюється або статичного або I2C модуль прекрасно справляється. Для високошвидкісної роботи із графікою необхідне тільки підключення через паралельний і SPI інтерфейси.*

**Модуль PoE живлення POE\_BOARD для Raspberry Pi 3 B+ ([raspberrypi.org](http://raspberrypi.org))**

Модуль PoE живлення для міні-комп'ютера Raspberry Pi 3 B+. Дозволяє заживлювати через Ethernet кабель, що дуже актуально для місць, де є кабель але немає можливості додатково підключити або встановити блок живлення. Плата відрізняється малими розмірами, високою якістю виготовлення й наявністю додаткового керованого вентилятора для

примусового охолодження компонентів комп'ютера. Модуль комплектується елементами кріплення до плати міні-комп'ютера.[81] Загальний вид Модуль PoE живлення POE\_BOARD представлено на рисунку 1.21



Рисунок 1.21 - Загальний вид модуля PoE живлення POE\_BOARD

Основні характеристики модуля PoE:

- сумісність із Raspberry Pi 3 B+;
- стандарт 802.3 af PoE;
- вхідна напруга від 36В до 56В;
- вихідна напруга 5В;
- максимальний вихідний струм 2,5А;
- керування охолодженням: мікроконтролерне;

Послідовність підключення модуля PoE до Raspberry Pi:

- перед установкою PoE HAT необхідно прикріпити проставлення, які йдуть у комплекті, у чотирьох кутах модуля;
- підключити Raspberry Pi PoE HAT до контактів GPIO і PoE (40 + 4), які встановлені на Raspberry Pi;

Після підключення, плата PoE HAT готова до використання й Raspberry Pi може житися через мережне рознімання RJ45.

***Модуль живлення Li-ion 14500 5В для Raspberry Pi від Waveshare  
([www.waveshare.com](http://www.waveshare.com))***

Модуль живлення для міні-комп'ютерів сімейство Raspberry Pi побудований на мікросхемі керування електроживленням SW6106, забезпечує вихідну напругу живлення 5 В від Вашого акумулятора 14500, що робить Raspberry Pi автономним портативним обладнанням. Він також заряджає акумулятор, підтримує двохнаправлену швидку зарядку. Цей модуль може використовуватися як універсальний міні-блок живлення й джерело безперебійного живлення, а також працює з іншими обладнаннями 5В.[82] Загальний вид модуль живлення Li-ion 14500 5В для Raspberry Pi представлено на рисунку 1.22

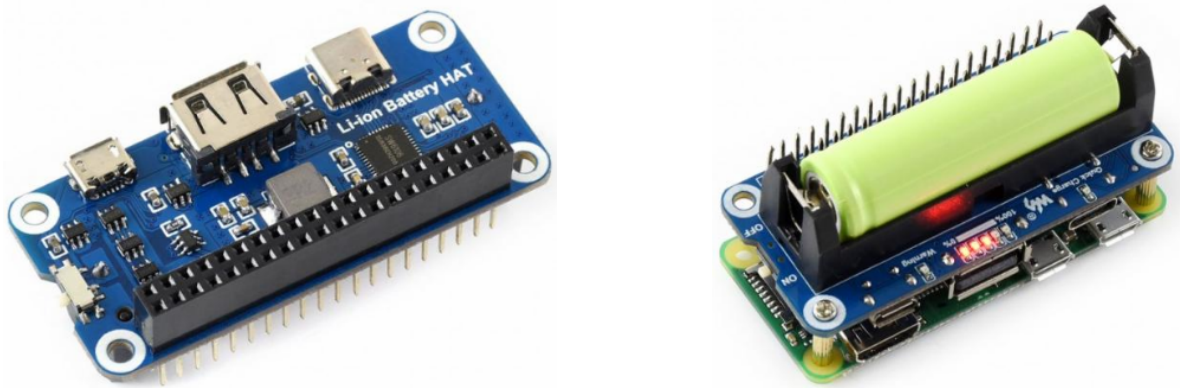


Рисунок 1.22 - Модуль живлення Li-ion 14500 5В для Raspberry Pi

Основні характеристики модуля живлення Li-ion 14500 5В для Raspberry Pi:

- тип батареї Li-іо;
- типорозмір 14500;
- напруга батареї 3.7В (4.2У при повній зарядці) ;
- мікросхема перетворювача SW6106;
- вихідна напруга 5 В;
- рознімання для Raspberry Pi GPIO;
- мікросхема керування живленням SW6106;
- рознімання USB Type-C заряд батареї / вихідна напруга;
- рознімання USB Type-A вихідна напруга;
- рознімання Micro USB заряд батареї;

- вимикач живлення Raspberry Pi.

Додаткові функції модуля живлення Li-ion 14500 5В для Raspberry Pi:

- вбудований чип SW6106 підтримує двонаправлену швидку зарядку, підтримує протоколи з декількома швидкими зарядами, такі як PD/QC/FCP/PE/SFCP;

- вбудований захист акумулятора від зворотної полярності, захист від перезаряду / перерозряду, захист від перевантаження по струму й захист від короткого замикання;

- вбудовані індикатори: швидкої зарядки, інформація про ємність акумулятора й неправильної полярності акумулятора.

***Модуль розширення DVK512 для Raspberry Pi від Waveshare  
([www.waveshare.com](http://www.waveshare.com))***

Плата розширення DVK512 від Waveshare - це плата, розроблена для лінійки міні-комп'ютерів Raspberry Pi, що поєднує в собі різні компоненти й інтерфейси для підключення зовнішніх аксесуарів. Загальний вид модуль розширення DVK512 представлено на рисунку 1.23.

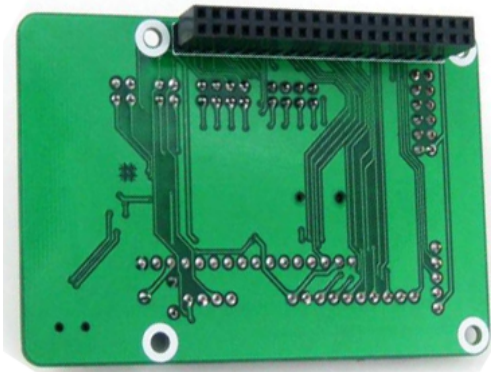
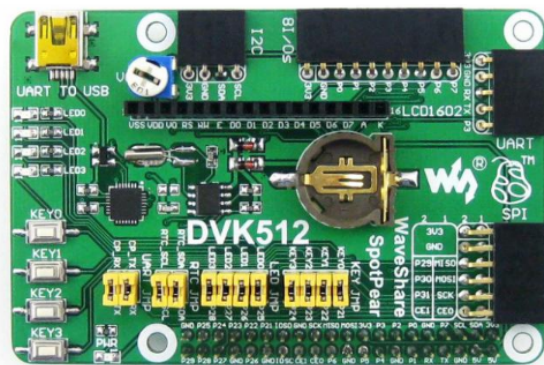


Рисунок 1.23 - Загальний вид модуль розширення DVK512 от Waveshare

Модуль розширення DVK512 сполучимо з наступними моделями Raspberry Pi:[83]

- Raspberry Pi 1 Model A+;

- Raspberry Pi 1 Model B+;

- Raspberry Pi 2 Model B;
- Raspberry Pi 3 Model B;
- Raspberry Pi 3 Model B+.

Розміщення компонентів на платі модуля розширення DVK512 представлено на рисунку 1.24.

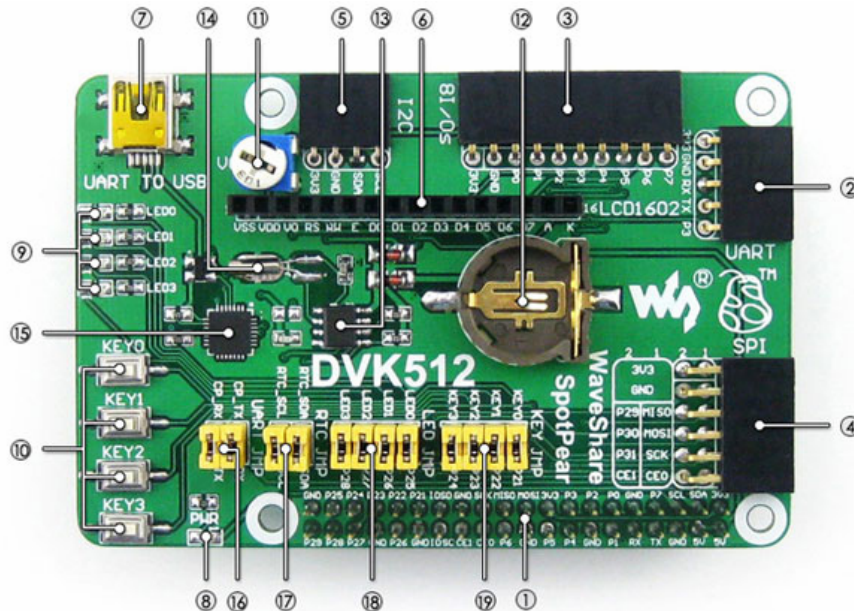


Рисунок 1.24- - Розміщення компонентів на платі модуля розширення DVK512

1. Рознімання для підключення до Rpi;
2. Інтерфейс UART: підключається до модулів UART, таким як RS232, RS485, USB-UART і т.д. ;
3. Інтерфейс 8I / O: з'єднується з модулями, керованими входами/виходами, такими як 8 кнопок Push, Logic Level Converter, Mix Board і т.д. ;
4. Інтерфейс SPI: підключається до Spi-Модулям, таким як AT45DBXX Dataflash, плата L3G4200D і т.д. ;
5. Інтерфейс I2C: підключається до модулів I<sup>2</sup>C, таким як плата розширення PCF8574, плата PCF8563 RTC, плата LSM303DLHC і т.д. ;
6. Інтерфейс ЖК-дисплея: для підключення ЖК -дисплеїв, таких як LCD1602;

7. USB-Рознімання: USB-UART, підтримуваний вбудованим перетворювачем CP2102;

8. Індикатор живлення;

9. Світло діоди призначені для користувача;

10. Призначені для користувача кнопки;

11. Потенціометр: для регулювання контрастності LCD1602;

12. Власник батареї RTC;

13. Мікросхема годин PCF8563;

14. Кварцовий резонатор 32.768К годин реального часу;

15. Мікросхема перехідника USB-UART CP2102;

16. Перемичка включення CP2102v

17. Перемичка включення годин реального часу;

18. Перемичка світлодіодів користувача;

19. Перемички для кнопок користувача.

### ***Модуль 3-х реле для Raspberry Pi от Waveshare ([www.waveshare.com](http://www.waveshare.com))***

Модуль реле для Raspberry Pi дозволяє програмно управляти потужним навантаженням з користувацької програми на ОС Raspberry. На платі встановлено три реле з максимальним струмом комутації до 10А і напругою до 250В. [84] Для індикації стану статусу реле в модулі передбачені світлодіоди. Загальний вид модуля представлено на рисунку 1.25

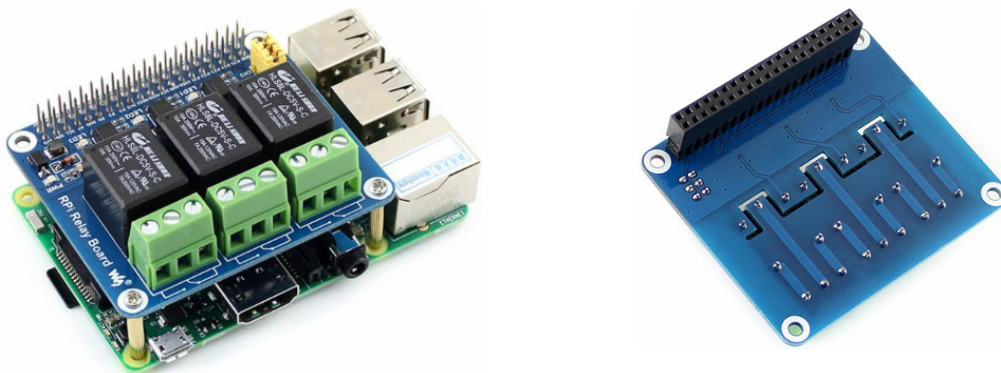


Рисунок 1.25 – Загальний вид модуля 3-х реле для Raspberry Pi от Waveshare

Особливості модуля реле для Raspberry Pi:

- сумісність із Raspberry Pi A+/B+/2B/3B/3B+;
- якісні реле з комутацією навантаження 250VAC/5A, 30VDC/5A;
- оптоізоляція для забезпечення якісної високовольтної ізоляції;
- вбудовані індикатори стану реле;
- перемички керування реле з можливістю підключення зовнішнього сигналу керування;
- підтримка засобами розробки в Wiringpi, Webiopi, Shell, Python і Vsc2835.

Розташування основних елементів та їх призначення представлені на рисунку 1.26

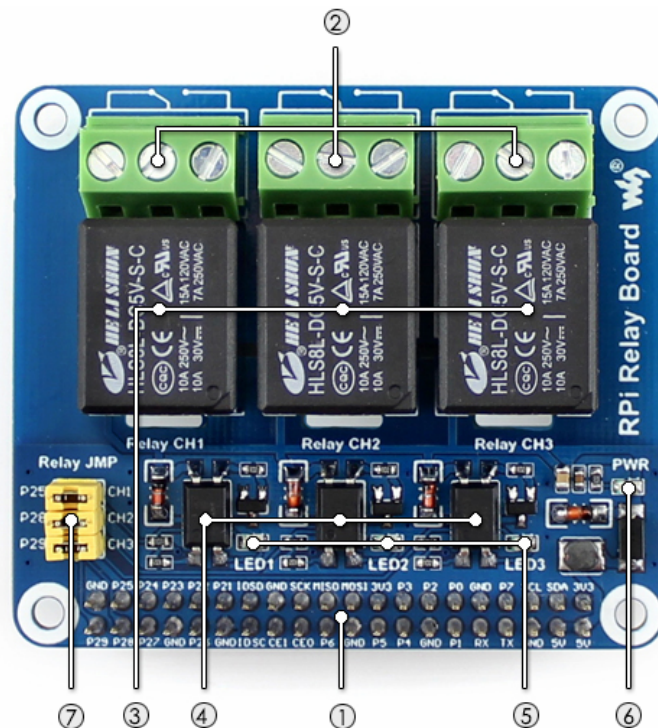


Рисунок 1.26- Розташування основних елементів модуля 3-х реле для Raspberry Pi

Призначення елементів на модулі для 3-х реле для Raspberry Pi представлено на рисунку 1.26.

1. Raspberry Pi GPIO інтерфейс для підключення до Raspebby Pi;

2. Рознімання контактів реле;
3. Реле;
4. Оптопара PC817;
5. Індикатор стану реле;
6. Індикатор горить: реле включене;
7. Індикатор не горить: реле виключене;
8. Індикатор живлення;
9. Перемичка керування реле;
10. Установлена: програмне керування реле;
11. Знята: керування реле перемичками або додатковими.

***Модуль колекторних двигунів для Raspberry Pi на MC33886 від Waveshare ([www.waveshare.com](http://www.waveshare.com))***

Модуль драйвера двох колекторних моторів для Raspberry Pi Motor Driver Board на мікросхемах MC33886. Сполучно практично з усією лінійкою міні-комп'ютерів Raspberry Pi і прекрасно підходить для побудови автономних робо-платформ і роботів. На платі розміщений потужний стабілізатор напруги, що дозволяє жити Raspberry Pi від силових акумуляторів напругою до 40В.[85] Загальний вид модуля на MC33886 представлено на рисунку 1.27.

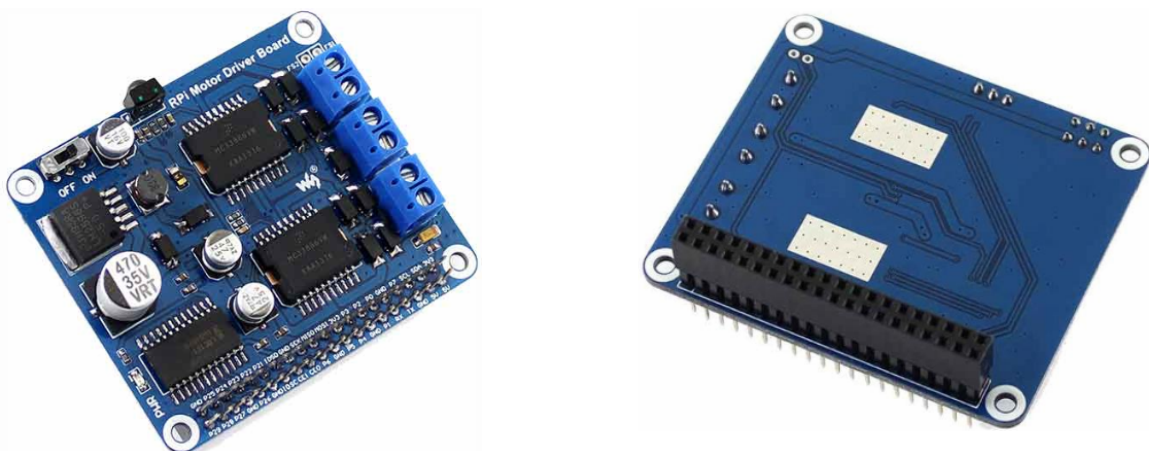


Рисунок 1.27 - Модуль драйвера двох колекторних моторів для Raspberry Pi Rpi Motor Driver Board на мікросхемах MC33886

Особливості модуля Motor Driver Board:

- сумісність із моделями Raspberry Pi A+/B+/2B/3B/3B+;
- мікросхема мостового драйвера двигуна: MC33886;
- вбудований стабілізатор напруги 5В для живлення Raspberry Pi;
- вбудований IR приймач для керування роботом;
- кілька захисних ланцюгів для стабільної роботи;
- запобіжник, що самовідновлюється;
- Driver chip features Short-Circuit Shutdown for Large Output Current, Undervoltage Disable Function;
- Reversed polarity protection for motor output;
- захист від неправильної полярності джерела живлення;
- діапазон вхідної напруги: 7В~40В;
- максимальне споживання одного мотора: до 5А;
- максимальний струм для живлення Raspberry Pi: до 2А.

Призначення основних елементів модуля Motor Driver Board і їх опис представлено на рисунку 1.28.

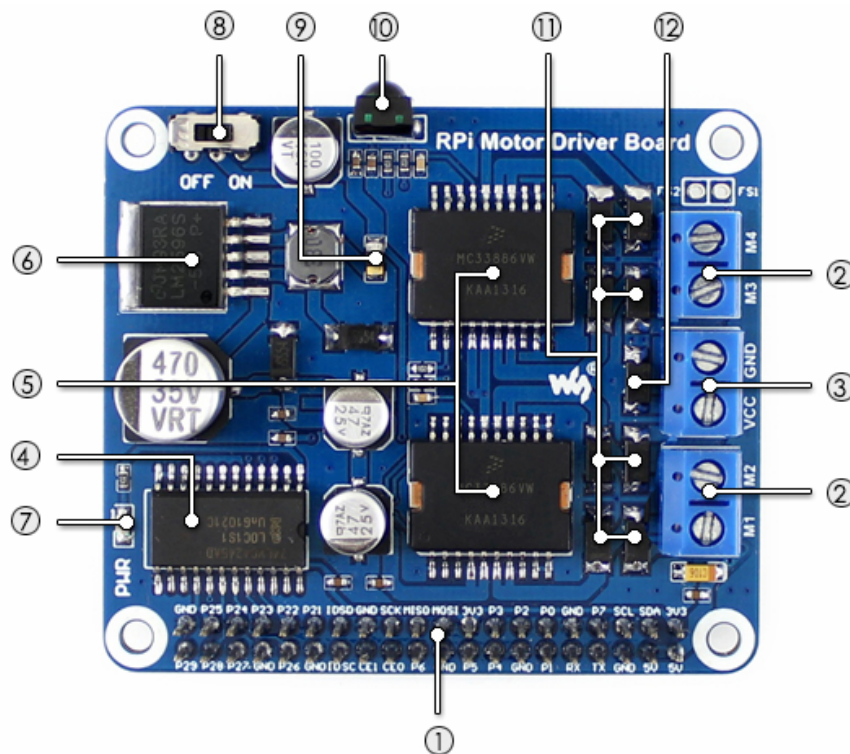


Рисунок 1.28 - Призначення загальних елементів модуля Motor Driver Board

1. Raspberry Pi GPIO інтерфейс для підключення до Raspberry Pi;
2. Рознімання для підключення двигунів;
3. Вхід в зовнішнього джерела живлення;
4. Перетворювач рівня логічних сигналів 74LVC4245AD;
5. Мікросхема драйвера двигунів MC33886;
6. Мікросхема стабілізатора LM2596-5.0;
7. Індикатор живлення;
8. Перемикач живлення Raspberry Pi:
  - OFF: Raspberry Pi живить драйвер двигуна;
  - ON: Драйвер двигателя живить Raspberry Pi
9. Запобіжник що самовідновлюється 2А;
10. ІР приймач;
- 11 Захисний діод Шоттки драйвера двигуна;
- 12 Захисний діод джерела живлення.

***Модуль прецизійного АЦП/ЦАП для Raspberry Pi від Waveshare  
([www.waveshare.com](http://www.waveshare.com))***

При всіх гідностях міні-комп'ютерів у них є один істотний недолік, що стримує їхнє застосування - відсутність аналогових входів і виходів. І, якщо з останніми простіше - їх можна замінити ШІМ виходами, то із входами усе більш сумно. Тепер ця проблема вирішена. Пропонований модуль АЦП/ЦАП Raspberry Pi High-Precision AD/DA Expansion Board дозволяє підключити 8 вхідних аналогових сигналів, оцифровуючи їх з 24-розрядним дозволом і частотою оцифровки до 30кГц, і 2 аналогових виходу з дозволом 16 розрядів і часом установлення напруги 10мкс. Аналогові входи можна використовувати як 4 диференціальних входу. Джерело опорної напруги вбудоване, але є можливість підключення зовнішнього джерела. Загальний вид модуля прецизійного АЦП/ЦАП для Raspberry Pi представлено на рисунку 1.29.

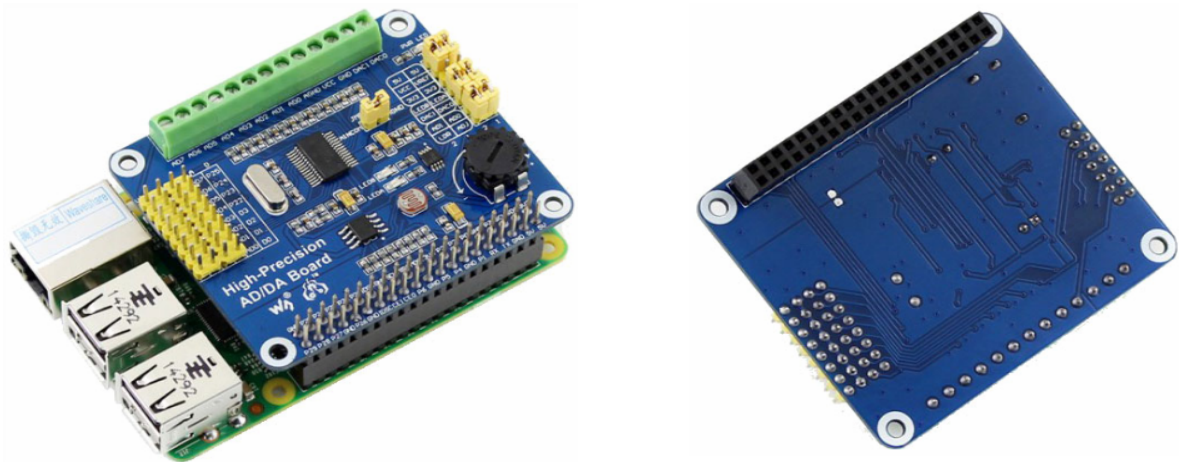


Рисунок 1.29- Загальний вид модуля прецизійного АЦП/ЦАП

Підтримувані моделі Raspberry Pi:

- Raspberry Pi 1 Model A+;
- Raspberry Pi 1 Model B+;
- Raspberry Pi 2 Model B;
- Raspberry Pi 3 Model B;
- Raspberry Pi 3 Model B+;

Особливості модуля прецизійного АЦП/ЦАП:[86]

- вбудований ADS1256 8-канальний 24-розрядний прецизійний АЦП (4 диференціальних входу), 30ksps частота оцифровки;
- вбудований DAC8532 2-канальний 16-розрядний прецизійний ЦАП;
- вбудований інтерфейс із розніманнями для підключення аналогових сигналів;
- розміщення контактів сумісне з Waveshare стандартом датчиків;
- вбудований інтерфейс для підключення аналогових датчиків " під гвинт";
- вбудовані датчики й індикатори для демонстрації обладнання.

Розміщення основних елементів і їх опис представлено на рисунку 1.30.

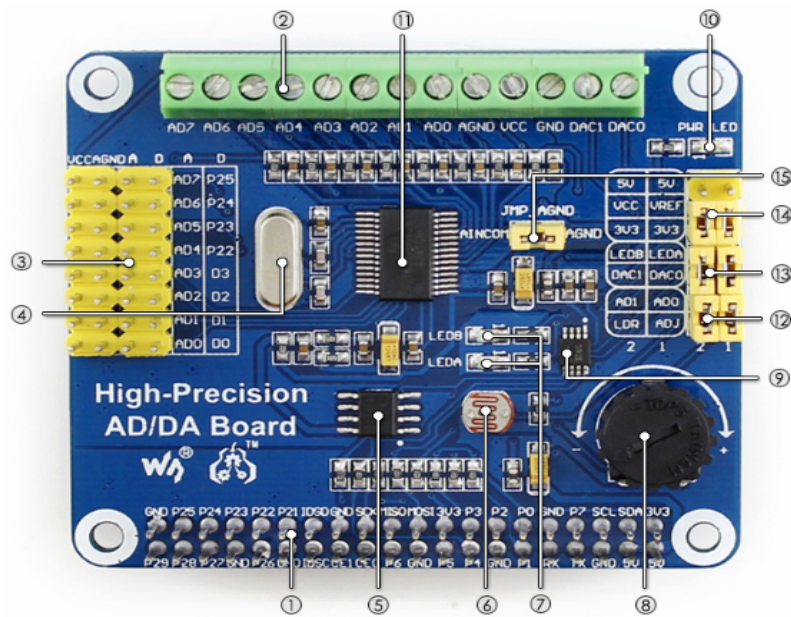


Рисунок 1.30 - Розміщення основних елементів на модулі прецизійного АЦП/ЦАП

1. Роз'єм розширення RPі для підключення до міні-комп'ютера;
2. АЦП/ЦАП входи/виходи;
3. АЦП входи. Розміщення контактів сумісне з Waveshare стандартом датчиків;
4. Кварцовий резонатор 7.68МГц;
5. Мікросхема джерела опорної напруги LM285-2.5 для АЦП;
6. Фоторезистор;
7. Світлодіоди стану виходів ЦАП;
8. Змінний резистор 10КОм;
9. Мікросхема прецизійного ЦАП DAC8532;
10. Світлодіодний індикатор живленням
11. Мікросхема прецизійного АЦП ADS1256;
12. Тестові перемички АЦП;
13. Тестові перемички ЦАПм
14. Перемички вибору джерела живлення;
15. Перемикач конфігурації ADC: використання вбудованого джерела напруги або зовнішнього, що підключається до контакту AINCOM;

***Комунікаційний модуль GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi від Waveshare (www.waveshare.com)***

Комунікаційний модуль для Raspberry Pi GSM/GPRS/GNSS/Bluetooth HAT на модулі SIM868 з підтримкою GSM, GPRS, GNSS і Bluetooth, що дозволяє легко перетворити Raspberry у мобільний телефон, меседжер, GPS трекер, сервер збору й передачі інформації з датчиків і багато чого іншого. Даний шилд сумісний з Raspberry Pi 3B+ / 3B / 2B / Zero/Zero W. За допомогою комунікаційного шилду можна управляти об'єктом на будь-якій відстані, одержувати й передавати дані. Управляється й налагоджується шилд за допомогою вбудованого USB - UART CP2102. Загальний вид представлено на рисунку 1.31

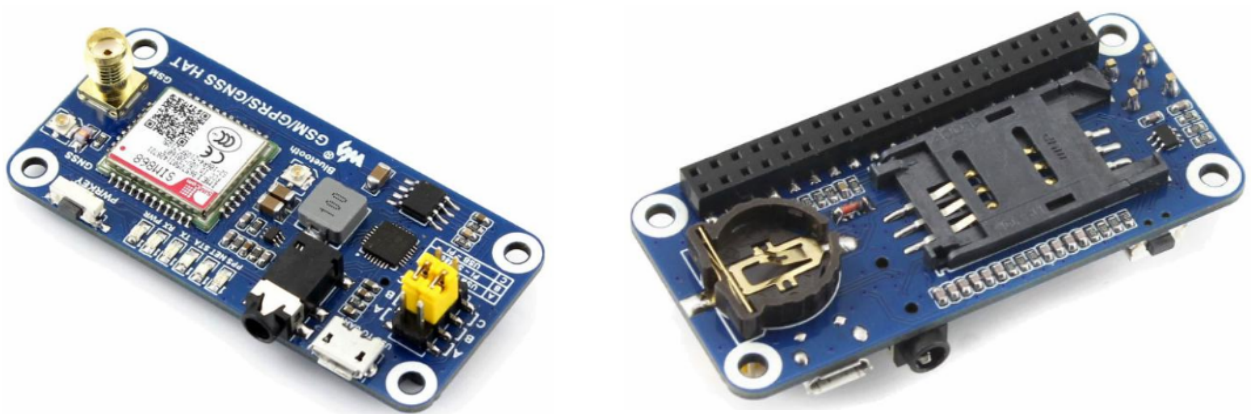


Рисунок 1.31 - Комунікаційний модуль GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi

Основні можливості комунікаційного модуля GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi:[87]

- повна сумісність із інтерфейсом Raspberry Pi 2B/3B/3B+/Zero/Zero W;
- підтримка приймання й відправлення SMS, телефонних дзвінків, приймання й передачі даних, GPRS, DTMF, HTTP, FTP, MMS, email і т.д.;
- підтримка GPS, COMPASS, Glonass, LBS позиціонування по базових станціях;

- підтримка передачі даних через Bluetooth 3.0;
- USB-UART конвертер на CP2102 для налагодження програм;
- 6 світлодіодів режиму роботи модуля;
- слот для SIM карти 1.8В/3В;
- наявність батарейки годин реального часу;
- автовизначення швидкості послідовного порту ( від 1200бод до 115200бод) ;
- керування AT командами (3GPP TS 27.007, 27.005 і розширені SIMCOM AT команди) ;
- підтримка додатків SIM application toolkit: GSM 11.14 Версія 99;
- підтримка ресурсами для розробки із прикладами програм Raspberry Pi/Arduino/STM32.

Призначення основних елементів комунікаційного модуля GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi представліні на рисунку 1.32 і описані нижче.

1. Модуль SIM868;
2. Стабілітрон ZMM5V1;
3. Діодна складання SMF05C;
4. USB-UART конвертер CP2102;
5. Перетворювач напруги MP1482;
6. Перетворювач рівнів NDC7002N;
7. Індикатор стану GPS;
8. Індикатор стану мобільної GSM мережі:
  - швидке миготіння - початок роботи модуля;
  - повільне миготіння - успішна реєстрація в мобільній GSM мережі;
9. Індикатор стану STA модуля;
10. Індикатор стану SIM868 UART Tx/Rx;
11. Індикатор живлення;

12. SIM868 control button: press the button and hold for 1s, to startup/shutdown the SIM868;
13. Рознімання підключення Raspberry Pi GPIO;
14. Рознімання Sim-Карти;
15. USB-UART інтерфейс;
16. 3.5мм рознімання підключення навушників і мікрофона;
17. Рознімання підключення GNSS антени;
18. Рознімання підключення Bluetooth антени;
19. Рознімання підключення GSM антени;
20. CR1220 battery holder: for RTC backup battery;
21. Перемикач режиму роботи UART інтерфейсу.

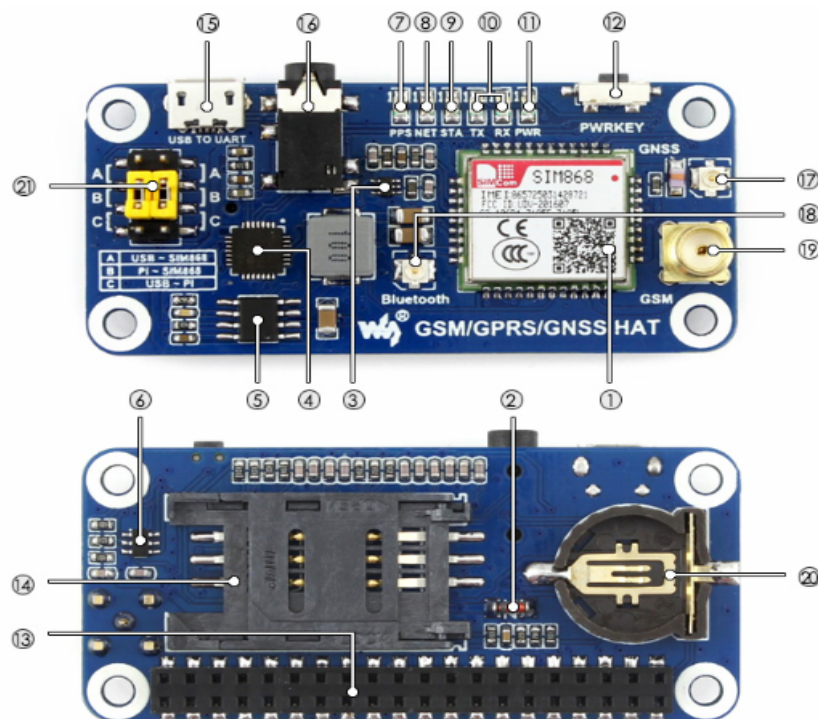


Рисунок 1.32. - Основні елементи комунікаційного модуля GSM/GPRS/GNSS/Bluetooth HAT для Raspberry Pi

### *Модуль годин реального часу для Raspberry Pi та Arduino*

Модуль має відмінну точність ходу, тому що кварцовий резонатор розміщений безпосередньо в корпусі мікросхеми й застосована цифрова

корекція точності ходу годин залежно від температури навколишнього середовища. Модуль можна використовувати й з контролерами Arduino, підключивши його перемичками до контролера. Загальний вид модуля представлено на рисунку 1.33



Рисунок 1.33 - Модуль годин реального часу для Raspberry Pi

Основні характеристики модуля годин реального часу для Raspberry Pi:[88]

- мікросхема годин: DS3231;
- напруга живлення: від 2.3 У до 5.5 В;
- робоча температура: від -40 ° С до + 85 ° С;
- точність ходу годин:  $\pm 5\text{ppm}$  ( $\pm 0.432$  сек/день) ;
- резервне живлення: Lipo акумулятор;
- параметри часу: годинник і календар. Включаючи секунди, хвилини, годинник, дні, дату, місяць і рік до 2100 з урахуванням високосного року;
- сигнали календаря: два сигнали;
- цифрові виходи: 1 Гц і 32.768 кГц;
- інтерфейс: I2C 400 кГц
- адреса обладнання: 0x68;
- точність датчика температури:  $\pm 3^{\circ}\text{C}$ .



*Наведені в даному розділі модулі мають завдання показати свою різноманітність і можливість розширювати базові функції одноплатних комп'ютерів сімейства Raspberry Pi.*

## 1.9 Датчики

### *3-осьовий акселерометр GY-291 ADXL345*

Акселерометр ADXL345 - це крихітний мікропотужний трьохосьовий акселерометр високого дозволу (13 біт). Діапазоном виміру положення коливається до  $\pm 16$  g. Результат виміру віддається у вигляді 16-розрядних чисел у додатковому коді й через цифрові інтерфейси SPI/I<sup>2</sup>C. Загальний вид 3-осьовий акселерометр GY-291 ADXL345 представлено на рисунку 1.34.

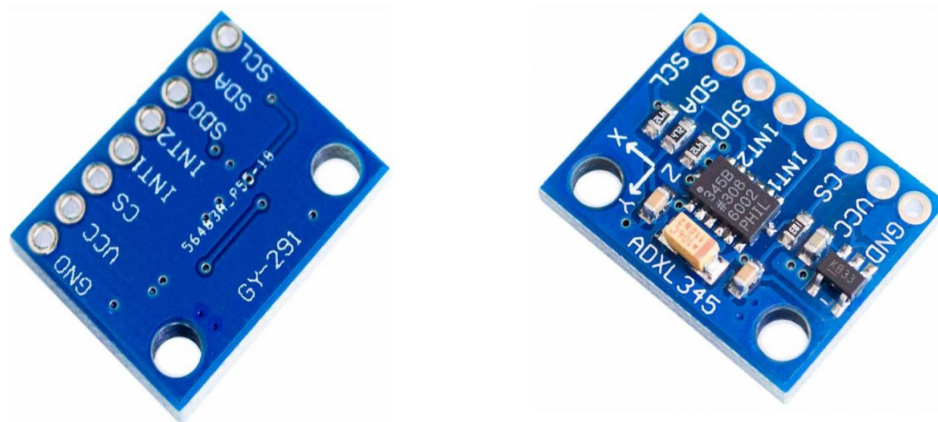


Рисунок 1.34 - Загальний вид 3-осьовий акселерометр GY-291 ADXL345

Даний акселерометр ADXL345 ідеально підходить для використання в мобільних обладнаннях - він вимірює статичне прискорення (викликане гравітацією) у завданнях визначення відхилення, або динамічне прискорення, викликане рухом або ударами. Заявлений високий дозвіл акселерометра ( $4 \cdot 10^{-3}$  g/LSB) дозволяє точно відслідковувати зміни відхилення менш ніж на  $1.0^\circ$ . Режим зниженого енергоспоживання датчика дозволяє реалізовувати інтелектуальне управління живленням системи.[89]

Основні характеристики 3-осьовий акселерометр GY-291 ADXL345:

- живлення акселерометра: 3-5В;
- підтримувані інтерфейси: I<sup>2</sup>C, SPI;
- дозвіл: 13bit, 4 мг / LSB;
- діапазон вимірів: + / - 16g;

- розміри: 20x15мм.



*Особливості одноплатних комп'ютерів сімейства Raspberry Pi полягає в тому що його можливостей вистачить на роботу з 90% датчиків. У цьому розділі ми розглянемо тільки ті датчики які будуть не обходжені для розробки мобільних робототехнічних платформ.*

### ***IR датчик відстані (10-80 см) Sharp GP2Y0A21YK0F.***

Сенсор із цифровою обробкою інформації й аналоговим виходом, що вимірює відстань у діапазоні від 10 до 80 см. Відрізняється високою точністю й маленьким часом відгуку. Загальний вид IR датчик представлено на рисунку 1.35.



Рисунок 1.35 - IR датчик відстані Sharp Sharp GP2Y0A21YK0F

Основні характеристики IR датчик відстані Sharp GP2Y0A21YK0F:

- робоча напруга: 4,5В - 5,5В;
- середній струм споживання: 30 мА;
- вимірювана відстань: 10 - 80 див;
- тип вихідного сигналу: аналоговий;
- середній час відновлення інформації на вивід OUT: 16,5 мс;
- розміри: 29,5 x 13 x 13,5 мм.[90]

### ***IR датчик відстані (4-30 см) Sharp GP2Y0A41SK0F***

Маленький сенсор із цифровою обробкою інформації й аналоговим виходом, що вимірює відстань у діапазоні від 4 до 30 см. Відрізняється високою точністю, маленьким часом відгуку, компактним розміром і низьким споживаним струмом. Загальний вид представлено на рисунку 1.36

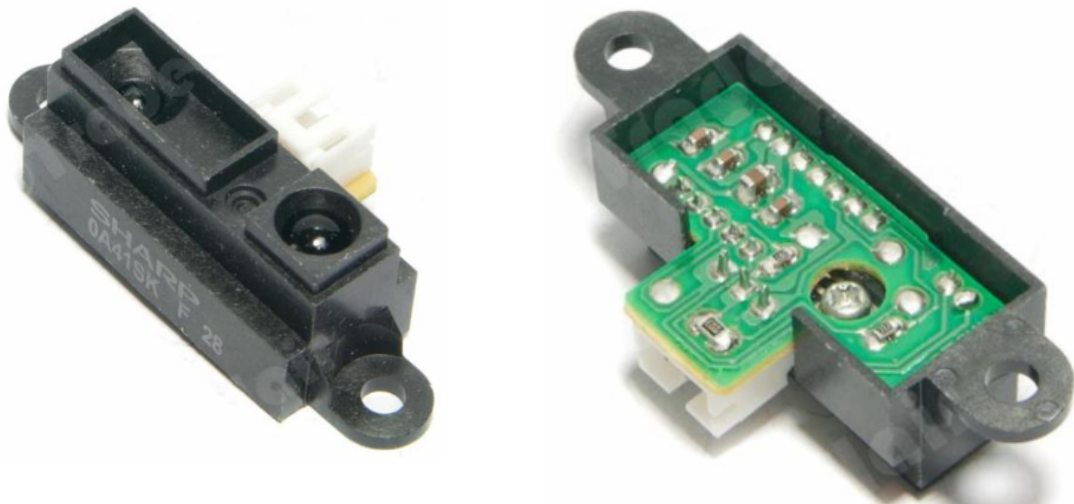


Рисунок 1.36 - IR датчик відстані (4-30 см) Sharp GP2Y0A41SK0F

Основні характеристики IR датчик відстані GP2Y0A41SK0F:

- робоча напруга: 4,5В - 5,5В;
- середній струм споживання: 12 мА;
- вимірювана відстань: 4 - 30 см;
- тип вихідного сигналу: аналоговий;
- середній час відновлення інформації на вивід OUT: 16,5 мс;
- розміри: 29,5 x 13 x 13,5 мм.[91]

***Аналоговий датчик ультрафіолету на GUVA-S12SD від Adafruit (www.adafruit.com).***

Модуль датчика ультрафіолету на фотодіоді GUVA-S12SD з піком чутливості в діапазоні випромінювання 240-370нм (UVB і більша частина UVA) ідеально підходить для конструкцій, що вимірюють рівень

ультрафіолетового випромінювання. Робочий струм фотодіода дуже малий - наноампери, тому, для підключення його до аналогового входу контролерів, потрібне застосування підсилювача на операційній системі. Загальний вид представлено на рисунку 1.37



Рисунок 1.37 - Загальний вид аналогового датчика ультрафіолету на GUVA-S12SD

Вихідна напруга датчика вважається по формулі:  $V_{out} = 4,3 \cdot I$  (де  $I$  - струм діода). Так при струмі у  $1\mu A$  ( $9\frac{mW}{cm^2}$ ) вихідну напругу складе  $4,3V$ . При перерахуванні напруги в індекс УФ випромінювання потрібно розділити вихідну напругу на  $0,1V$ . т.б. якщо вихідна напруга складе  $0,5V$  при індексі УФ випромінювання 5. Зверніть Вашу увагу на те, що УФ світлодіоди, як правило, мають інший діапазон випромінювання. Для реєстрації випромінювання й перевірки датчика краще підійдуть сонячне випромінювання й лампи для засмаги.

Основні характеристики аналогового датчика ультрафіолету на GUVA-S12SD:[92]

- тип фотодіода: GUVA-S12SD;
- тип виходу: аналоговий;
- робочий діапазон УФ випромінювання: 240-370nm (UVB і більша частина UVA);
- напруга живлення: від 2.7V до 5.5V;
- розміри: 10мм x 19мм x 2мм.

### **Водонепроникний ультразвуковий далекомір DYP-ME007 (30см - 3.5м)**

DYP-ME007 - водонепроникний ультразвуковий модульний далекомір, що забезпечує вимір відстані в діапазоні від 30 см до 3.5 м. Далекімір застосовується для виміру таких відстаней як: автомобільний вимірник відстані при паркуванні, відхилення від перешкод роботів, системи безпеки й промислового контролю й багатьох інших.[93] Загальний вид ультразвукового далекоміра DYP-ME007 представлено на рисунку 1.38.

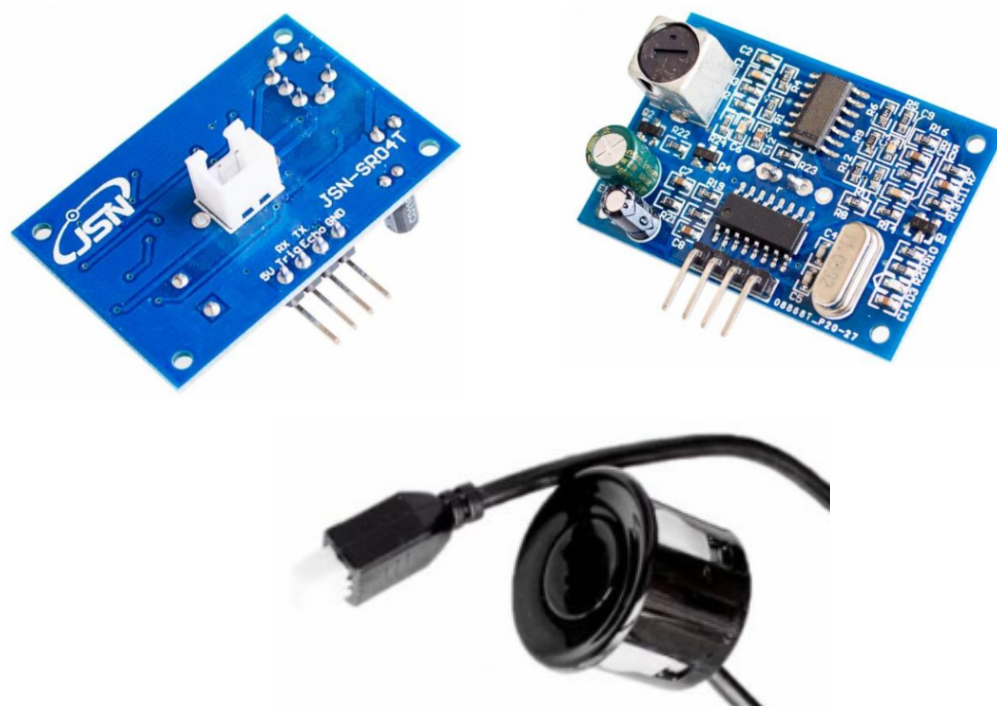


Рисунок 1.38 - Загальний вид ультразвукового далекоміра DYP-ME007

Основні технічні характеристики ультразвукового далекоміра DYP-ME007:

- робоча напруга: DC 5 В;
- робочий струм: <15ma;
- частота: 40 кгцм
- діапазон виміру: 30 см - 3.5 м;
- вихідний сигнал: 5 Урівень TTL;
- час відгуку: 100 мс;

- точність: 1 см;
- робоча температура: -10 - 60 град.С;
- температура зберігання: -20 - 80 град.С;
- розмір (Д x Ш): прибіл. 4x3 см;
- довжина кабелю: прибіл. 200 см;
- діаметр випромінювача 12мм, корпуса 25мм, висота корпуса 20м.

***Датчик вологості й температури SHT10 від Robotdyn (www.robotdyn.com)***

Цифровий датчик температури й вологості SHT1x простий у використанні, точний і підключається стандартним інтерфейсом I2C. Якісне калібрування забезпечує високу точність, відмінну стабільність і довгочасне використання. Загальний вид датчик вологості й температури SHT10 представлено на рисунку 1.39.

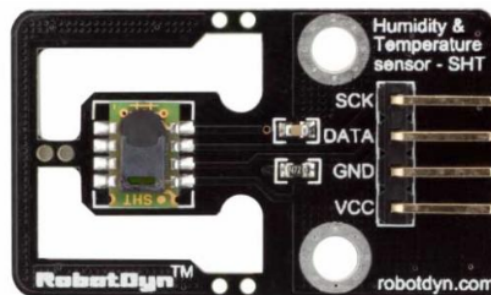


Рисунок 1.39 - Загальний вид датчик вологості й температури SHT10

Технічні характеристики датчик вологості й температури SHT10:

напруга живлення: 3,5 - 5 В

- низьке енергоспоживання 30 мВт
- вимірювана вологість: 0 ... 100% відносної вологості
- абсолютна точність виміру вологості +/- 2% відносної вологості (відносна вологість 10-90%)
- визначається температурою: -40 - + 128°S ± 0,3%
- точність: +/- 0,3 ° С при 25 ° С

- частота дискретизації:  $\leq 1$  Гц.[94]

### ***Датчик руху мікрхвильовий RCWL-0516***

Датчик руху RCWL-0516 працюючий на ефекті Доплера з можливістю виявлення рухів минаючи перешкоди, використовується з Ардуїно або іншими платформами, також його можна застосовувати прямо з релейними модулями. Загальний вид датчика руху RCWL-0516 представлено на рисунку 1.40



Рисунок 1.40 - Загальний вид датчика руху RCWL-0516

В основу роботи датчика закладений ефект Доплера – зміна частоти відбитої хвилі, внаслідок руху випромінювача, приймача або відбивача. У даному модулі частота випромінюваної їм радіохвилі міняється внаслідок руху відбивача (перешкоди). Модуль побудований на базі чипа RCWL-9196 який оснащений передавачем і приймачем. Датчик спрацює, якщо приймач прийме сигнал, частота якого незначно відрізняється від частоти сигналу передавача:

- якщо в зоні дії датчика немає об'єктів здатних відбивати радіохвилі, то приймач нічого не прийме й датчик не спрацює;

- якщо в зоні дії датчика є нерухливі об'єкти здатні відбивати радіохвилі, то приймач прийме радіохвилю передавача, відбиту від цих об'єктів, але частота прийнятої радіохвилі буде дорівнювати частоті сигналу передавача й датчик не спрацює;

- якщо в зоні дії датчика є об'єкт здатний відбивати радіохвилі, який наближається до датчика (рухається), то приймач прийме відбиту від об'єкта

радіохвилю, частота якої буде вище чому в сигналу передавача й датчик спрацює;

- якщо в зоні дії датчика є об'єкт здатний відбивати радіохвилі, який віддаляється від датчика (рухається), то приймач прийме відбиту від об'єкта радіохвилю, частота якої буде нижче ніж в сигналі у передавача й датчик спрацює.

Коли датчик спрацює, на його виході «OUT» устанавлюється рівень логічної «1». Датчик постачений тригером, який утримує рівень логічної «1» на виході «OUT» у плині 2 сек  $\pm 30\%$ , після припинення руху.

Якщо датчик багаторазово спрацює, наприклад, постійно фіксує руху в плині 10 секунд, то рівень логічної «1» на виході «OUT» буде встановлений на 12 секунд із моменту першого спрацювання (10 секунд під час фіксації рухів + 2 секунди після їхнього припинення, поки не «скинеться» тригер). Використання датчиків руху заснованих на ефекті Доплера дозволяє фіксувати руху через об'єкти, що не відбивають радіохвилі (дерево, пластик, гіпс і т.д.), чого не можуть зробити датчики руху засновані на піроелектричному ефекті (такі як HC-SR501).[95]



*Увага! Не підключайте живлення до виходу «3V3» модуля.*

Підключення датчика руху RCWL-0516:

**OUT** - вихід датчика (установлюється в «1» при наявності рухів + затримується на 2 секунди після їхнього припинення).

**VIN** - вхід напруги живлення, від +4 до +28 В постійного струму.

**GND** - вхід живлення (загальний).

**3V3** - вихід стабілізованої напруги живлення 3,3 В (можна використовувати для живлення мікроконтролерів).

**COS** - вхід дозволу (підтягнутий внутрішнім опором чіпа). Якщо на даному виводі встановити рівень логічного «0», то після скидання тригера, він не буде встановлюватися (датчик перестане реагувати на рухи).

Вихід «OUT» підключається до будь-якого вивід Arduino (на схемі підключений до виводу D4). Вхід «COS» можна залишити не підключеним, тому що він підтягнутий внутрішнім опором чипу, отже, дозволяє роботу тригера. Доп.висновок CDS – можна використовувати для дистанційної установки фоторезистора, а так само для добору змінним резистором ступені освітленості при якій починає працювати датчик, потім цей номінал припаюється до R-CDS.

Основні параметри датчика руху RCWL-0516:

- напруга живлення (VIN): 4-28 В постійного струму
- споживаний струм: до 3 мА (номіально 2,8 мА)
- дальність виявлення: до 9 м (номіально до 5 м)
- кут виявлення: 120 гради.
- потужність передавача: до 30 мВт (номіально до 20 мВт)
- частота передавача: 5,8 ГГц
- час затримки до скидання тригера: 2 сек  $\pm$ 30%
- вихідна напруга (3V3): 3,2...3,4 В (номіально 3,3 В)
- максимальний струм на виході «3V3»: до 100 мА
- робоча температура: -20 до +80 °С
- температура зберігання: -40 до +100 °С

#### ***Датчик лінії для Arduino від Robotdyn ([www.robotdyn.com](http://www.robotdyn.com))***

Сенсор відстеження лінії для Arduino може виявляти білі лінії на чорному фоні й чорні лінії на білому фоні. Якщо датчик відстеження лінії виявляє чорний колір, сигнальна лінія (DO) датчика переходить у стан HIGH (1), у той же час загоряється світлодіод (синій) на датчику. Якщо датчик відстеження лінії виявляє білий колір, сигнальна лінія (DO) переходить у стан LOW (0). За допомогою потенціометра Ви можете налаштувати чутливість датчика лінійного спостереження й відстань до лінії. Загальний вид датчик лінії представлено на рисунку 1.41



Рисунок 1.41 - Датчик лінії для Arduino від Robotdyn

Основні характеристики датчик лінії:

- напруга живлення: 3.3В / 5В;
- споживаний струм: 20 мА при 5 В;
- вага: 2,8 г.[96]

### ***Датчик освітленості GY-49 MAX44009 I<sup>2</sup>C 3.3В***

Датчик освітленості на мікросхемі MAX44009 (GY-49) призначений для виміру рівня навколишнього висвітлення наближеного до чутливості людського ока. Модуль має цифровий вихід через інтерфейс I<sup>2</sup>C. Датчик по своїх характеристиках відмінно підходить для різних портативних обладнань або промислового використання. Наявність на платі стабілізатора напруги живлення й перетворювача рівнів інтерфейсу I<sup>2</sup>C дозволяє використовувати модуль із мікроконтролерами як з 5В так і з 3,3В логікою. Загальний вид датчика GY-49 MAX44009 представлений на рисунку 1.42

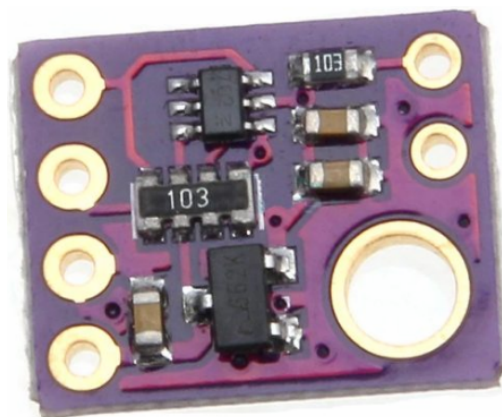


Рисунок 1.42 - Загальний вид датчика  
GY-49 MAX44009

Спектральний відгук чипа оптимізований робити коректні виміри в режимі природнього висвітлення й містить у собі Ік-Фільтр і Уф-Фільтр для усунення засвічення при штучнім висвітленні. Блок адаптивного посилення автоматично вибирає коректний діапазон освітленості для оптимізації значень. Програмувальний вихід переривання виконаний у вигляді виходу з відкритим стоком і може обслуговувати керуючу логіку з будь-яким діапазоном напруг живлення.[97]

Основні характеристики датчика GY-49 MAX44009:

- інтерфейс: I<sup>2</sup>C
- діапазон освітленості: 0.045 - 188000 Люкс
- динамічний діапазон: 22 біт
- живлення: 3,3 - 5 В
- струм споживання: 0.65 мка
- розмір плати: 2 x 1 див
- робоча температура: - 40 до +85 °С

#### *Датчик пульсу MAX30102 (Датчик серцевого ритму)*

Датчик серцевого ритму (пульсу) MAX30102 – інтегральний датчик пульсу й насичення крові киснем. У датчику зібрано оптимізовану оптику, два світлодіода, фотодетектор, високоточний аналоговий підсилювач і перетворювач, цифровий оброблювач і інтерфейсний модуль. Для підключення датчика до контролера використовується послідовний інтерфейс I<sup>2</sup>C. Загальний вид датчика MAX30102 представлено на рисунку 1.43

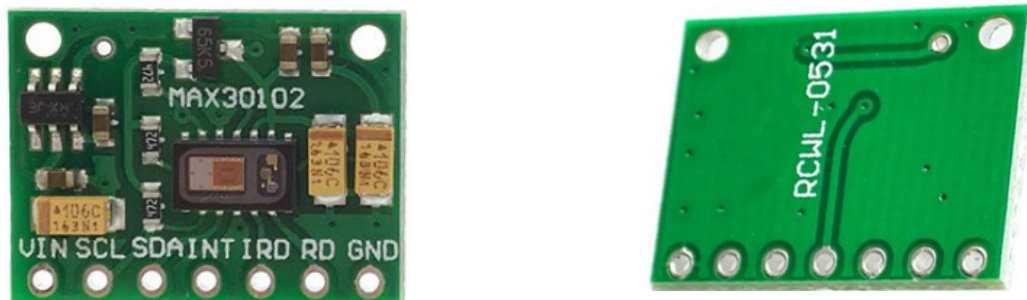


Рисунок 1.43 - Загальний вид датчика MAX30102

У датчику використовується два світлодіода: червоного й інфрачервоного спектра. Є можливість управляти як струмом через світлодіоди ( від 0мА до 50мА) так і тривалістю імпульсів ( від 200мкс до 1,6мс) для забезпечення максимальної точності вимірів. Для калібрування датчика є вбудований датчик температури.[98]

Характеристики датчика МАХ30102:

- вимірювані параметри: частота пульсу й насичення крові й SpO2
- напруга живлення: 3,3В (внутрішній стабілізатор на 1,8В)
- споживаний струм у режимі виміру: 1.2 мА
- споживаний струм у режимі сну: до 10 мкА
- інтерфейс: I2C
- максимальна частота інтерфейсу: 400 кГц.

#### ***Датчик температури й вологості SHT31-D ІІС.***

Модуль цифрового датчика температури й вологості SHT-31D відкриває новий рівень у технології вимірів. У функціонал датчика входить схема обробки й посилення сигналу, блок пам'яті калібрування, АЦП і схема скидання по живленню, а також широкий діапазон напруги живлення від 2.5 до 5.5В. Даний модуль може використовуватися з різними мікроконтролерами на базі Arduino або Raspberry Pi.[99] Загальний вид представлено на рисунку 1.44



Рисунок 1.44 - Датчик температури й вологості SHT31-D ІІС

Основні характеристики датчика температури й вологості SHT31-D ІС:

- діапазон виміру відносної вологості: 0 - 100 %
- точність діапазону відносної вологості: 2 %
- тип інтерфейсу: I2C
- дозвіл: 16 біт
- точність: +/- 0.3 С
- напруга живлення: 2.5 - 5.5 В
- робочий струм: 800  $\mu$ а
- робоча температура: -40 до +125 град.С

### *Датчик струму з аналоговим виходом на MAX471.*

Модуль аналогового типу для виміру сили струму споживаного навантаженням від джерела живлення. Принцип виміру заснований на контролі напруги, що знімається із прецизійного дільника напруги. Для виміру використовується мікросхема MAX471 із вбудованим вимірювальним шунтом. Загальний вид представлено на рисунку 1.45



Рисунок 1.45 - Загальний вид датчик струму з аналоговим виходом на MAX471

Максимальний вимірюваний струм для одного модуля 3А, для збільшення цієї межі, схема дозволяє застосовувати паралельне включення. Або можна використовувати зовнішній струмовий резистор.

Основні характеристики Датчика струму з аналоговим виходом на MAX471:

- напруга живлення MAX471: від 3 до 36В;
- вимірюваний струм: від 0А до 3А;
- точність виміру струму: 2%;
- напрямок виміру струму: двонаправлений;
- споживаний струм: 100 мкА;
- споживаний струм у режимі очікування: 18 мкА;
- корпус MAX471: 8-вивідний SOIC;
- розмір друкованої плати: 22x21x7 мм.[100]

### ***Датчик ультрафіолету на GUVA-S12SD***

Модуль датчика ультрафіолету на GUVA-S12SD ідеально підходить для конструкцій, що вимірюють рівень ультрафіолетового випромінювання. Лінійна залежність датчика вимірів дозволяє обійтися без перетворень, що знижують швидкість і точність вимірів. Загальний вид датчика ультрафіолету на GUVA-S12SD представлено на рисунку 1.46



Рисунок 1.46 - Загальний вид датчика ультрафіолету на GUVA-S12SD

Основні характеристики датчика ультрафіолету на GUVA-S12SD:

- датчик: GUVA-S12SD;
- діапазон вимірів: 240nm-370nm

- великий кут реєстрації: 130 градусів
- тип світлочутливих елементів: діоди Шоттки
- напруга живлення: 2.5 В ~ 5 В;
- робочий струм становить: 1мА;
- тип характеристики: лінійна.[101]

### ***ІК-термомодуль (4-pin)***

ІК-термомодуль призначений для виміру температури повітря й/або предметів. Маленькі розміри, висока точність і можливість прямо підключати його до контролера Arduino або модулю реле дозволяють широко використовувати його в конструкціях де потрібно точно й швидко вимірювати температуру: термометри, термостати, системи вентиляції й кондиціонування повітря. Загальний вид представлено на рисунку 1.47.

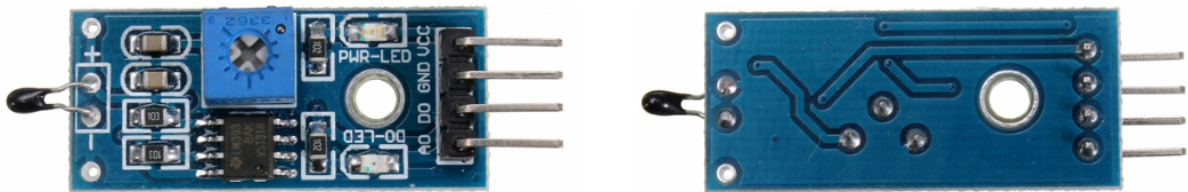


Рисунок 1.47 - Загальний вид ІК-термомодуля (4-pin)

Основні характеристики й особливості модуля:

- використовується NTC чуттєвий термістор;
- діапазон температури, що вимірюється від 20 до 80 °С;
- використовується NTC чутливий термістор;
- діапазон температури, що вимірюється т від 20 до 80 °С;
- цифровий вихід зі струмом до 15 мА;
- аналоговий вихід сигналу датчика;
- поріг, що настроюється, спрацьовування цифрового виходу;
- вбудований індикатор стану цифрового виходу й напруги живлення;
- робоча напруга від 3В до 5В;
- маленький розмір: 3.2см x 1.4 см;

- використовується розповсюджений компаратор LM393 із широким діапазоном робочої напруги.

### *Лазерний датчик відстані GY-530 на VL53L0X*

GY-530 VL53L0X - мініатюрний модуль датчика відстані й розпізнавання жестів. Швидкий і точний вимір відстані до 2 м. Модуль підключається через розповсюджений послідовний інтерфейс I2C для керування обладнанням і передачі даних. Напруга живлення модуля 5В.

VL53L0X оснащений передовою матрицею на основі високочутливих однофотонних лавинних діодів. Принцип роботи заснований на запатентованій Stmicroelectronics технології Flightsensetm. Загальний вид представлено на рисунку 1.48

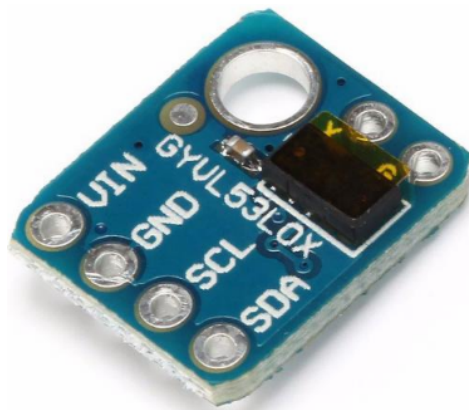


Рисунок 1.48 - Лазерний датчик відстані  
GY-530

Поверхнево-випромінюючий лазер VCSEL з довжиною хвилі 940 нм, виконує в датчику відстані VL53L0X роль джерела оптичного сигналу. Він оснащений вбудованим інфрачервоним фільтром. Його світіння повністю невидимо для людського ока, і забезпечує більшу дистанцію виміру при меншій чутливості до рівня зовнішнього висвітлення й більш стійкий до перехресних перешкод, що викликано скляними поверхнями. Датчик VL53L0X має вкрай низьке енергоспоживання.

Можливі області застосування лазерного датчика відстані GY-530:

- датчики присутності персональних ПК / ноутбуків / планшетних ПК і обладнань Інтернету речей (автоматичні вмикачі висвітлення)
- робототехніка (датчики виявлення перешкод)
- побутова техніка (автоматичні крани, вентиля, дозатори мила і т.д.)
- одномірні датчики розпізнавання жестів
- лазерні системи автофокусування. Поліпшує й прискорює роботу автофокуса камери, особливо в несприятливих умовах (низька освітленість, низька контрастність) або при зйомці на високій швидкості руху.[102]

### ***Модуль датчика RGB і жестів APDS-9960 від Robotdyn***

Датчик жестів кольору RGB APDS-9960 - це цифровий датчик RGB висвітлення, що оточує, датчик наближення й жестів. Датчик оснащений інтерфейсом, сумісним з I<sup>2</sup>C, із червоним, зеленим, синім, прозорим (RGBC) сенсорами наближення й жестів з IR-світлодіодом. Функція виміру RGB і навколишнього світла визначає інтенсивність світла при різних умовах висвітлення й різних матеріалах, включаючи затемнене скло. Крім того, вбудований УФ-IR-Фільтр блокує паразитну IR і УФ засвічення й корелює передачу кольору. Загальний вид модуля датчика RGB і жестів APDS-9960 представлено на рисунку 1.49

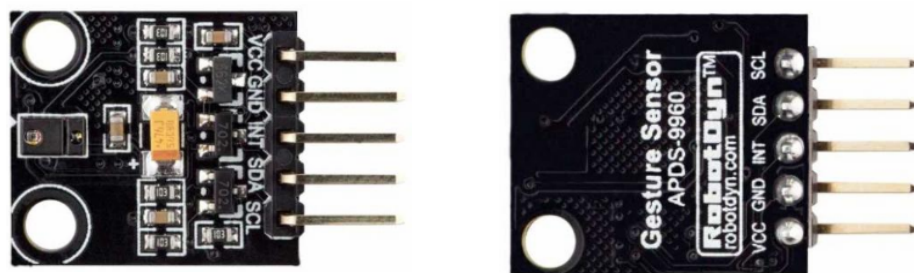


Рисунок 1.49 - Загальний вид модуля датчика RGB і жестів APDS-9960

Функція наближення й жестів відкалібрована до 100 мм відстані виявлення наближення, не вимагаючи додаткового калібрування датчика.

Виявлення жесту використовує чотири спрямовані фотодіоди, інтегровані з видимим фільтром, для точного визначення простих жестів униз-вправо-уліво або більш складних жестів. Додавання мікрооптичних лінз у модуль забезпечує високу ефективність передачі й приймання інфрачервоного випромінювання. Внутрішній контролер переводить датчик у режим зниженого енергоспоживання між вимірами RGBC, наближення й жестів, забезпечуючи дуже низьке енергоспоживання.

Особливості модуля датчика RGB і жестів APDS-9960:

- датчик світла RGBC, детектор наближення й жестів з IR-Індикатором в оптичному модулі;
- інтерфейс сумісний з I<sup>2</sup>C з висновком спеціального переривання;
- висока чутливість, що забезпечує роботу за темним склом;
- світлочутливість RGBC із вбудованим блоковим фільтром UV-IR;
- геометрично впорядковані фотодіоди RGBC, що забезпечують рівномірний кутовий відгук;
- калібрування до 100-міліметрової відстані виявлення;
- чотири окремі фотодіоди, чутливі до різних напрямків;
- вимір наближення й жест із інтегрованим блоковим фільтром;
- запатентована конструкція екрана;
- низьке енергоспоживання: 1,0 мкА, типовий у режимі сну.[103]

### ***Модуль датчика звуку від Waveshare***

Модуль датчика звуку від Waveshare побудований на звуковому підсилювачі мікрофона LM386 і компараторі LM393, служить для одержання інформації про звуковий фон навколишнього середовища в мікроконтролер або використання для підключення до звукового підсилювача через цифровий або аналоговий виходи для оцінки й виміру його інтенсивності. Легко підключається до контролера перемичками, які входять у комплект модуля. Коефіцієнт підсилення й поріг спрацьовування регулюються змінними резисторами на платі. Загальний вид представлено на рисунку 1.50

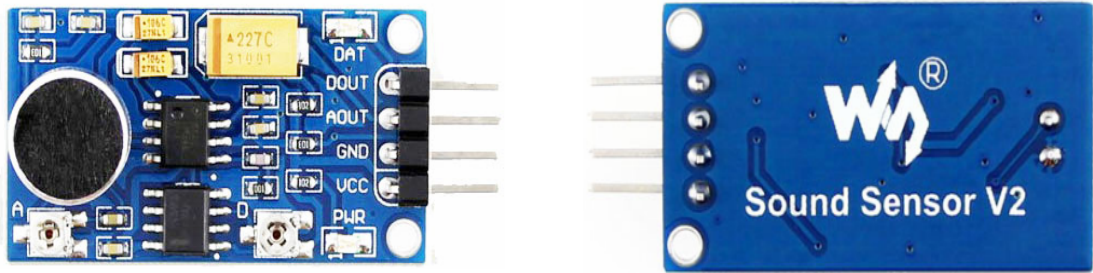


Рисунок 1.50 - Загальний вид модуля датчика звуку від Waveshare

Загальні характеристики модуля датчика звуку:

- чутливість мікрофона 52 db;
- частотний діапазон 50Гц ~ 20 КГц;
- операційний підсилювач LM386;
- коефіцієнт підсилення 200;
- напруга живлення 3.3 ~ 5.3 В;
- діаметр монтажних отворів: 2.0 мм.[104]

### ***Модуль датчика наближення VL6180X***

Модуль унікального датчика наближення й освітленості VL6180X, що використовує технологію Flightsense від Stmicroelectronics, здатний визначати відстань до об'єктів у діапазоні відстані 1-100 мм із міліметровою точністю незалежно від характеристик об'єкта, а також вимірювати освітленість у широкому динамічному діапазоні. Гідності нового датчика дозволяють зробити прорив в області побудови безконтактних систем керування як у побутовій і портативній електроніці, так і в промислових системах. Загальний вид представлено на рисунку 1.51

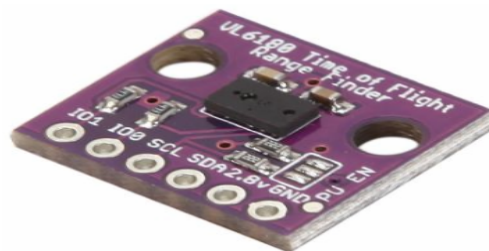


Рисунок 1.51 - Загальний вид модуль датчика наближення  
VL6180X

Загальні характеристики датчика наближення VL6180X:

- тип випромінювача інфрачервоний;
- довжина хвилі випромінювача 850нм;
- вимірювана відстань 0-50 см;
- тип інтерфейсу I<sup>2</sup>C;
- адреса обладнання 0x29 ( 7-біт) ;
- датчик температури навколишнього середовища від 0.038 до 50 кlux;
- розрядність вихідного коду 16.[105]

### ***Модуль датчика температури й вологості DHT22 від Waveshare***

Модуль датчика температури й вологості підвищеної точності DHT22. Містить усі необхідні додаткові компоненти для підключення датчика до мікроконтролера. Підходить для підключення як до контролерів Arduino так і до інших мікроконтролерів з напругою логічних рівнів 5V и 3.3V. Для зменшення споживаного струму модулем необхідно відключити світлодіод індикатору напруги живлення на платі модуля. Датчик модуля вимірює вологість і температуру точніше свого попередника - датчика вологості й температури DHT11. Загальний вид представлено на рисунку 1.52



Рисунок 1.52 - Загальний вид датчика температури й вологості DHT22

Основні характеристики датчика температури й вологості DHT22:

температура:

- дозвіл виміру: 0.1°C
- точність виміру : ±0.5°C
- вимірюваний діапазон : -40°C ~ 80°C

вологість:

- дозвіл виміру: 0.1%RH
- точність виміру:  $\pm 2\%$ RH (25°C)
- вимірюваний діапазон: 0%RH ~ 99.9%RH

робоча напруга: від 3.3У до 5.5 В

рекомендовані умови зберігання:

температура: 10°C ~40°C

вологість: 60%RH або нижче.[106]

### ***Модуль датчика струму на ACS709 від -75А до +75А від Pololu.***

Модуль датчика струму ACS709 є лінійним струмовим датчиком на мікросхемі Allegro ACS709LLFTR-35BB-T  $\pm 75$ A на основі ефекту Холу з аналоговим виходом з низькоомним внутрішнім опором ( $\sim 1,1$  мом) і електричною ізоляцією до 2,1 кВ RMS. Датчик оптимізований для точного виміру струмів від -37,5 А до 37,5 А, а аналоговий вихід напруги лінійний для значень струму до 75 А. Вихідну напругу центровано по  $VCC / 2$  і має типову погрішність  $\pm 2\%$ . Датчик працює в діапазоні живлячої напруги від 3 В до 5,5 В, тому він може прямо взаємодіяти із системами 3,3 У и 5 В. Загальний вид представлено на рисунку 1.53

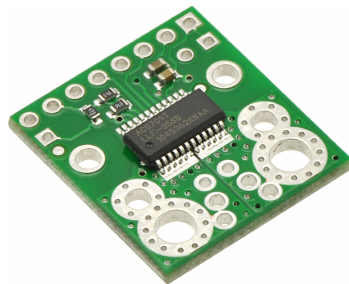


Рисунок 1.53 - Загальний вид датчика струму ACS709

Ключові функції датчика струму на ACS709:

- оптимізована точність для двонаправленого вхідного струму від -37,5 А до 37,5 А с лінійним діапазоном чутливості від -75 А до 75 А;
- внутрішній опір провідного шляху звичайно становить 1,1 мом, а друкована плата виконана зі стовщеної міді, тому на платі виділяється дуже мало тепла;

- вбудована ізоляція значно зменшує ємнісний зв'язок від токопроводу до вимірювальної матриці й запобігає зсуву дрейфу в додатках;
- використання датчика на ефекті Холу означає, що ІС здатна електрично ізолювати електроніку від датчика ( до 2.1 кВ RMS), що дозволяє вставляти датчик у будь-якому місці провідника струму й використовувати в додатках, що вимагають електричну ізоляцію;
- смуга шириною 120 кГц, яка може бути додатково зменшена шляхом додавання конденсатора через штирьки плати з написом «FILT»;
- висока точність і надійність: середня загальна помилка виходу 2% при кімнатній температурі із заводським калібруванням, надзвичайно стійка вихідна напруга спокою;
- робочий діапазон робочої температури від -40 °С до 150 °С;
- установлене користувачем граничне значення максимального струмового навантаження: вихідні дані виходу FAULT затримуються на низькому рівні, якщо струм перевищує встановлений поріг протягом часу, який може бути встановлений шляхом додавання зовнішнього конденсатора.



*Збільшений діапазон від -75 А до 75 А датчика повинен бути обмежений піковими перехідними струмами. При тестуванні виявлено, що мікросхема датчика може витримувати струм 50 А протягом 20 секунд або 37,5 А протягом 150 секунд, перш ніж перевищити максимальну температуру 150 °С. Тому, якщо Ви не вживаєте спеціальних заходів відводу тепла від мікросхеми датчика, ми рекомендуємо обмежити безперервні струми до рівня 30 А. Навіть при низькому опорі датчика 1,1 мом плата може нагріватися настільки, щоб досягтися високих температур, тому, для тривалої роботи з більшими струмами слід ухвалювати додаткові заходи по відводу тепла.*

Розташування й призначення висновків модуля датчика презентовано на рисунку 1.54

Єдиними підключеннями, які необхідні для використання цього датчика, є вхідний струм ( $IP +$  і  $IP-$ ), напруга живлення ( $VCC$  і  $GND$ ) і вихід датчика ( $VIOUT$ ). Усі інші контакти є необов'язковими, як і два зовнішні конденсатори, показаних на рисунку праворуч.

Датчик вимагає напруги живлення від 3 В до 5,5 В для підключення через контакти  $VCC$  і  $GND$ , які позначені на платі шовкографією. Датчик виводить аналогову напругу, лінійно пропорційну вхідному току. Вихідна напруга спокою становить  $VCC / 2$  і змінюється на 28 мВ на ампер вхідного струму (при  $VCC = 5$  В), при цьому позитивний струм збільшує вихідну напругу а негативний струм зменшує вихідну напругу.

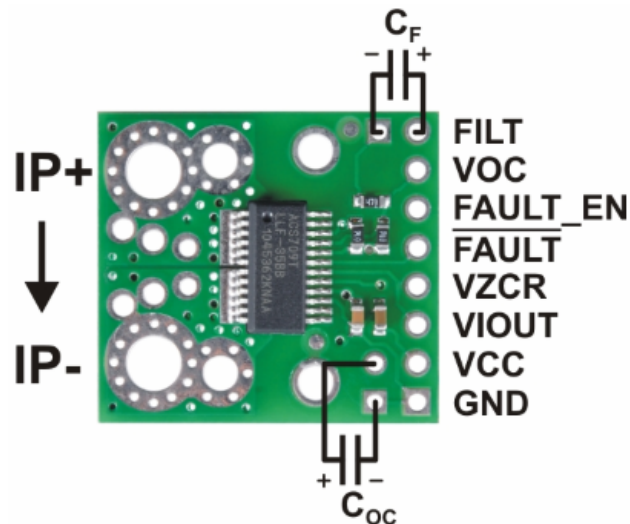


Рисунок 1.54 - Розташування й призначення висновків модуля датчика струму на ACS709

Для довільного вхідного струму  $i$  (в амперах) вихідна напруга датчика може бути в більш загальному виді представлена як:

$$VIOUT = (0,028B/A \cdot i + 2,5B) \cdot VCC / 5V$$

Висновок  $VZCR$  є виходом опорної напруги й може використовуватися як опорний сигнал нульового струму (0A). Він буде приблизно рівний  $VCC/2$

і дозволить Вам більш точно обчислити вхідний струм від вихідної напруги VIOUT.

Висновок FILT дозволяє Вам регулювати пропускну здатність плати, додаючи конденсатор CF до землі ( для зручності була доданий контакт заземлення поруч із контактом FILT). Без якого-небудь зовнішнього фільтруючого конденсатора ширина смуги становить 120 кГц. У технічному паспорті втримується додаткова інформація про те, як конденсатор зовнішнього фільтра впливає на пропускну здатність.

Висновок FAULT перебуває в низькому рівні, коли струм перевищує поріг спрацьовування від перевантаження по струму несправності ( $I_{oc}$ ). Ця крапка перемикавання задається напругою, що подається на вивід VOC, і залежить від дільника напруги, показаного на принциповій схемі. За замовчуванням  $I_{oc}$  устанавлюється на значення струму в 57А, але його можна змінити, додавши зовнішні резистори до дільника напруги, щоб змінити напругу VOC. Після того, як контакт FAULT зафіксований на низькому рівні, його можна скинути, якщо вхідний сигнал FAULT\_EN низький (цей вхід за замовчуванням піднятий на платі). Зовнішній конденсатор, СОС, може бути доданий, щоб збільшити час відгуку по перевантаженню по струму. Без цього конденсатора час відгуку, як правило, менше 2 мкс. Для одержання докладної інформації про використання функції аварії перевантаження по струму, у тому числі про деякі обмеження на перевищення максимального струму. Вхідний струм можна підключати до плати різними способами. Отвору із кроком 0,1", 3,5 мм і 5 мм доступні, як показано на картинці вище, для підключення штирьков або клемних колодок. Для потужнострумівих застосувань ви можете припаяти проведення безпосередньо до наскрізних отворів, які щонайкраще відповідають Вашим проводам або Ви можете використовувати безконтактні клемні колодки, як показано на рисунку вище. Більші наскрізні отвори достатні для гвинтів М6.

Принципова схема модуля датчика струму на ACS709 представлена на рисунку 1.55



й простий у підключенні й використанні. Загальний вид представлено на рисунку 1.56.

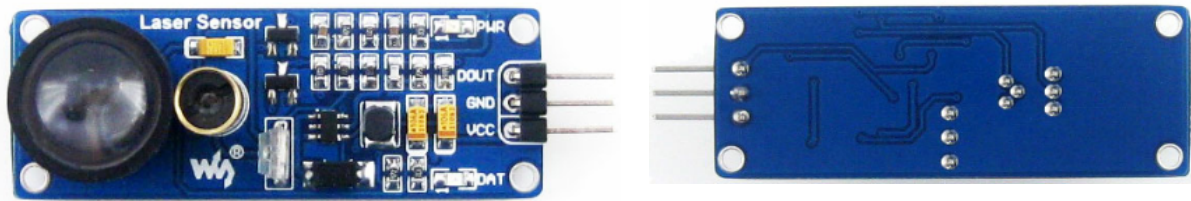


Рисунок 1.56 - Загальний вид модуля лазерного датчика перешкоди

Загальні характеристики модуля лазерного датчика перешкоди:

- відстань виявлення: 0.8м, 1.5м (максимальне);
- напруга живлення: від 2.5У до 5.0В;
- розміри: 47.7мм \* 17.9мм;
- діаметр монтажного отвору: 2.0мм. [108]

### ***H206-модуль оптичного датчика швидкості обертання***

Може використовуватися для виміру швидкості обертання за допомогою диска із прорізами або як датчик перешкоди. Принцип – визначення об'єкта (перешкоди) між прорізами. Так само цей модуль можна використовувати в якості високоточного кінцевого датчика положення (ENDSTOP). На платі встановлений граничний компаратор напруги для одержання вихідного сигналу із крутими фронтами. Від інших аналогічних датчиків відрізняється компактними розмірами. Загальний вид оптичного датчика представлено на рисунку 1.57.



Рисунок 1.57 - Загальний вид оптичного датчика

Основні характеристики оптичного датчика:

- напруга живлення 3,3В - 5В;
- ширина паза датчика 6 мм;
- тип виходу аналоговий і цифровий;
- індикатор стану виходу.[109]

### ***Модуль камери Raspі CAM v2 для Raspberry Pi***

Raspі CAM v2 камера с 8MP сенсором від Sony IMX219, поліпшеним фіксованим фокусом. Камера дозволяє робити статичні фото з дозволом 3280 x 2464 пікселів, знімає відео в трьох режимах — 1080p @ 30fps, 720p @ 60 fps і 640 x 480p @ 90fps. Загальний вид камери Raspі CAM v2 представлено на рисунку 1.58

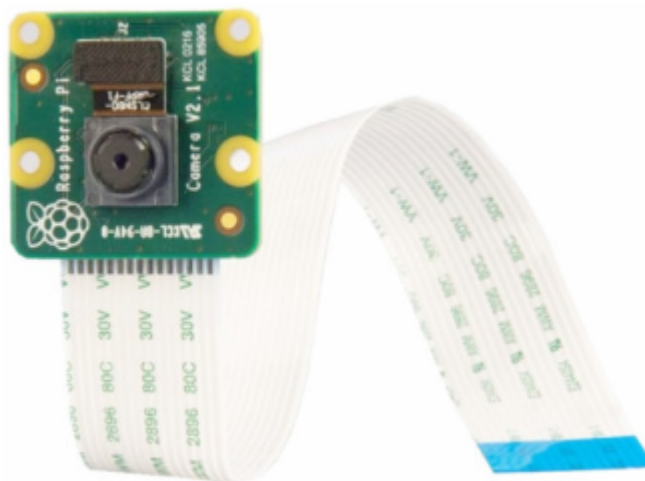


Рисунок 1.58 - Камера для Raspberry Pi - Raspі CAM v2.

Основні характеристик Raspі CAM v2:

- 8-мі мегапіксельний сенсор Sony IMX219;
- поліпшений фіксований фокус;
- дозвіл фото: 3280 x 2464 пікселів;
- підтримувані формати відео: 1080p @ 30fps, 720p @ 60 fps і 640 x 480p @ 90fps;
- розмір матриці CCD : ¼ дюйма;

- 1.4 $\mu$ m x 1.4 $\mu$ m пікселів з технологією Omnibsi для високої продуктивності (висока чутливість, низький рівень перехресних перешкод, низький рівень шуму);

- розміри: 25 x 23 x 9mm.[110]

## 1.8 Контрольні питання

1. Дайте визначення поняттю Internet of Things;
2. Опишіть основні концепції терміну Industry 4.0;
3. Що таке кіберфізическі системи (CPS)?;
4. Рівні Industry 4.0 ?;
5. Що таке одноплатний комп'ютер ?;
6. Проведіть порівняння одноплатних комп'ютерів Asus Tinker Board 2Гб і Raspberry Pi 3 Model B +
7. Опишіть основні достоинства і недоліки одноплатних комп'ютерів;
8. Що таке PoE?
9. Дайте загальні характеристики лінійки олноплатних комп'ютерів Orange Pi;
10. Перерахуйте особливості LattePanda 2GB / 32GB від інших одноплатних комп'ютерів ?;
11. Дайте опис основних характеристик PINE A64 + Board 1Гб;
12. Для чого потрібні плати расшеренія для одноплатних комп'ютерів?
13. Призначення модуля Arduino Expansion Shield ?;
14. Особливості використання модуля PoE для Raspberry Pi 3 B +;
15. Для вирішення яких завдань не обходимо використовувати модуль харчування Li-ion 14500 5В ?;
16. До яких одноплатним комп'ютерів підходить модуль розширення DVK512 ?;
17. Призначення модулів 3-х реле ?;
18. Дайте визначення скорочення АЦП / ЦАП;

19. Принцип роботи ІR датчик відстані ?;
20. Для чого використовується датчик лінії ?;
21. Призначення датчика ультрафіолету ?;
22. Принцип роботи лазерного датчика відстані GY-530 ?;
23. Привид загальні характеристики лазерного датчика перешкоди;
24. Основні параметри модуль камери Raspі CAM v2;

## 2. АРХИТЕКТУРА ОДНОПЛАТНОГО КОМП'ЮТЕРА RASPBERRY PI 3 MODEL B PLUS

### 2.1 Основні елементи Raspberry Pi 3 model B plus

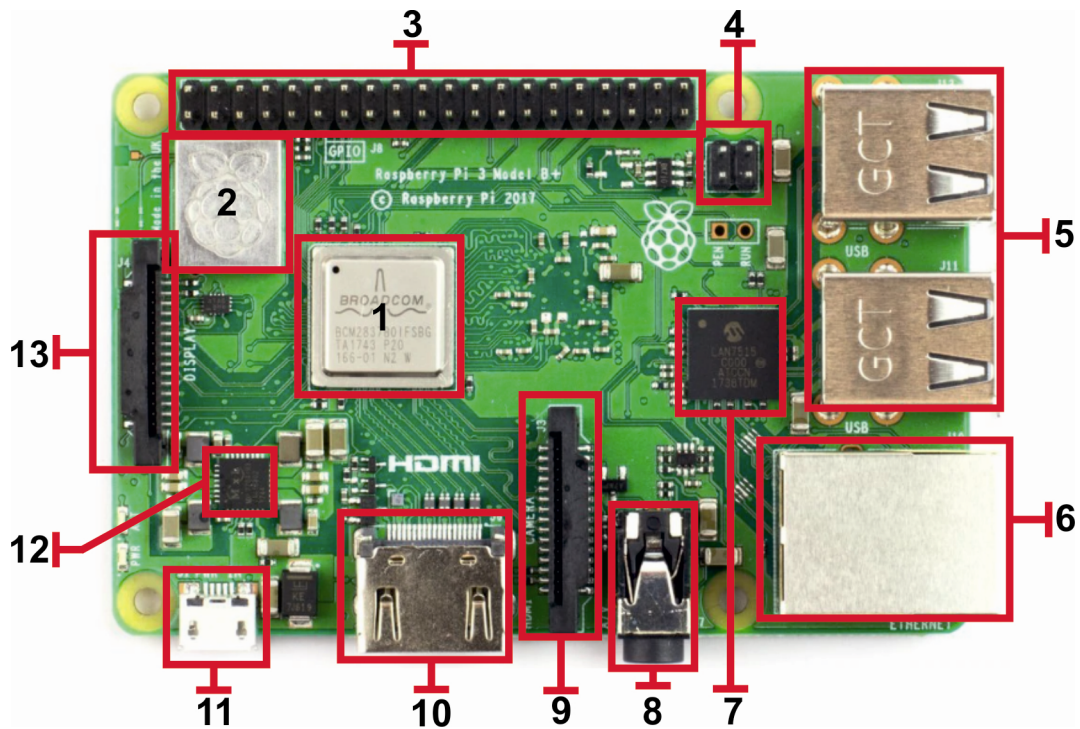


Рисунок 2.1 – Розташування основних елементів на платі  
Raspberry Pi 3 - Model B Plus.[110]

1 – чип Broadcom BCM2837B0, Cortex-A53, 64-bit, quad-core Soc @ 1.4Ghz with 1GB LPDDR2 SDRAM (Поліпшене керування температурою й продуктивністю);

2 – чип Cypress CYW43455, IEEE 802.11 b/g/n/ac 2.4Ghz/5Ghz Wireless LAN (WLAN) ;

3 – Extended 40-pin GPIO header;

4 – Power over Ethernet (PoE) header (requires separate PoE HAT);

5 – 4xUSB 2.0 ports and Faster Ethernet over USB 2.0 (maximum throughput 300Mbps);

6 – роз'єм для конектора RJ45 (кручена пара);

7 – чип Microchip LAN7515;

8 – роз'єм 3,5 mm (jack), 4 pole stereo output and composite video port;

9 – MIPI CSI camera port;

10 – Full size HDMI;

11 – microUSB power connector 5V/2.5 A;

12 – НАВ мікросхема;

13 – MIPI DSI display port;

Raspberry Pi 3 - Model B Plus підтримує наступні периферійні обладнання й протоколи:[111]

– 48x GPIO;

– 2x I<sup>2</sup>C;

– 2x SPI;

– 2x UART;

– 2x SD/SDIO<sub>M</sub>

– 1x HDMI 1.3a;

– 1x USB2 HOST/OTG;

– 1x DPI (Parallel RGB Display);

– 1x NAND interface (SMI);

– 1x 4-lane CSI Camera Interface (up to 1Gbps per lane);

– 1x 2-lane CSI Camera Interface (up to 1Gbps per lane);

– 1x 4-lane DSI Display Interface (up to 1Gbps per lane);

– 1x 2-lane DSI Display Interface (up to 1Gbps per lane).

Взаємодію між основними елементами Raspberry Pi 3 - Model B Plus представлено на рисунку 2.2 виді блок схеми.

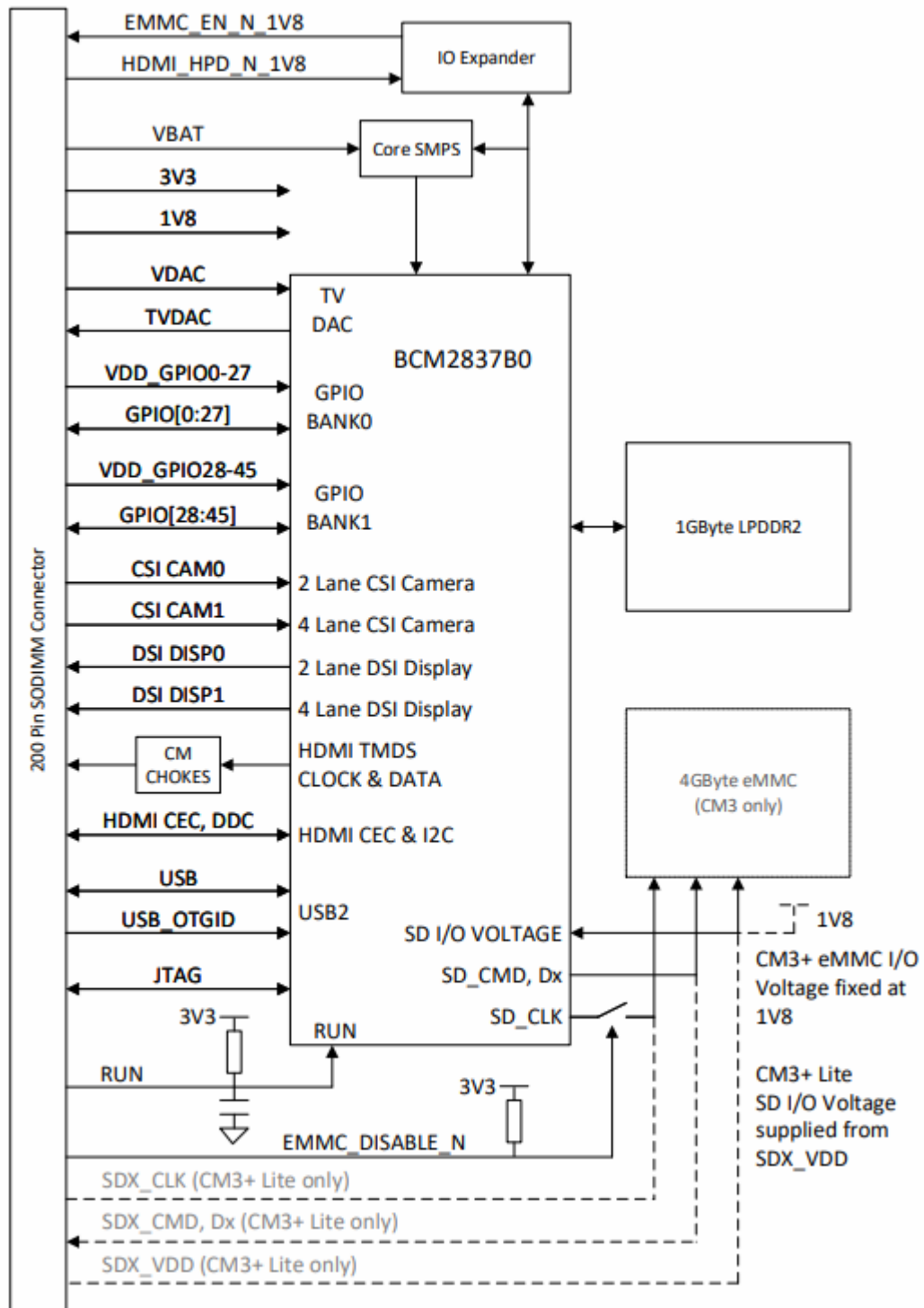


Рисунок 2.2 – Блок схема взаємодії основних елементів  
Raspberry Pi 3 - Model B Plus.[111]

## 2.2 Електричні характеристики Raspberry Pi 3 - Model B Plus

Одноплатний міні-комп'ютер Raspberry Pi 3 - Model B Plus має шість окремих джерел живлення, які повинні працювати постійно й не можна

залишити жодного з них без живлення, навіть якщо певний інтерфейс або банк GPIO не використовується:[112]

- VBAT is used to power the BCM2837 processor core. It feeds the SMPS that generates the chip core voltage;

- 3V3 powers various BCM2837 PHYs, IO and the eMMC Flash;
- 1V8 powers various BCM2837 PHYs, IO and SDRAM;
- VDAC powers the composite (TV-out) DAC;
- GPIO0-27 VREF powers the GPIO 0-27 IO bank;
- GPIO28-45 VREF powers the GPIO 28-45 IO bank.

Робочій діапазон живлення Raspberry Pi 3 - Model B Plus представлено в таблиці 2.1

Таблиця 2.1 - Робочій діапазон живлення Raspberry Pi 3 - Model B Plus

Supply	Description	Minimum	Typical	Maximum	Unit
VBAT	Core SMPS Supply	2.5	-	5.0 + 5%	V
3V3	3V3 Supply Voltage	3.3 - 5%	3.3	3.3 + 5%	
1V8	1V8 Supply Voltage	1.8 - 5%	1.8	1.8 + 5%	
VDAC	TV DAC Supply <sup>α</sup>	2.5 - 5%	2.8	3.3 + 5%	
GPIO0-27 VDD	GPIO0-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	
GPIO28-45 VDD	GPIO28-45 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	
SDX VDD	Primary SD/eMMC Supply Voltage	1.8 - 5%	-	3.3 + 5%	

<sup>α</sup>- Requires a clean 2.5-2.8V supply if TV DAC is used, else connect to 3V3

Основні електричні характеристики Raspberry Pi 3 - Model B Plus наведено в таблиці 2.2.

Таблиця 2.2 - Основні електричні характеристики Raspberry Pi 3 - Model B Plus

Symbol	Parameter	Minimum	Maximum	Unit
VBAT	Core SMPS Supply	-0.5	6.0	V
3V3	3V3 Supply Voltage		4.10	
1V8	1V8 Supply Voltage		2.10	
VDAC	TV DAC Supply		4.10	
GPIO0-27 VDD	GPIO0-27 I/O Supply Voltage			
GPIO28-45 VDD	GPIO28-45 I/O Supply Voltage			
SDX VDD	Primary SD/eMMC Supply Voltage			



*Напруги вище перераховані у таблицях 2.1-2.2 можуть привести до необоротного ушкодження Raspberry Pi 3 - Model B Plus. Це тільки рейтинг максимального стресу який не впливає на функціонування обладнання в тих або інших умовах..*

Основні характеристики Raspberry Pi 3 - Model B Plus по постійному струму (DC) представлено в таблиці 2.3.

Таблица 2.3 - DC Characteristics

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
1	2	3	4	5	6	7
$V_{IL}$	Input low voltage <sup>α</sup>	VDD IO = 1.8V	-	-	0.6	V
		VDD IO = 2.7V	-	-	0.8	
		VDD IO = 3.3V	-	-	0.9	
$V_{IH}$	Input high voltage <sup>α</sup>	VDD IO = 1.8V	1.0	-	-	V
		VDD IO = 2.7V	1.3	-	-	
		VDD IO = 3.3V	1.6	-	-	
$I_{IL}$	Input leakage current <sup>α</sup>	TA = +85C	-	-	5	μA
$C_{IN}$	Input capacitance	-	-	5	-	pF
$V_{OL}$	Output low voltage <sup>β</sup>	VDD IO = 1.8V, IOL = -2mA	-	-	0.2	V
		VDD IO = 2.7V, IOL = -2mA	-	-	0.15	
		VDD IO = 3.3V, IOL = -2mA	-	-	0.14	
$V_{OH}$	Output high voltage <sup>β</sup>	VDD IO = 1.8V, IOH = 2mA	1.6	-	-	V
		VDD IO = 2.7V, IOH = 2mA	2.5	-	-	
		VDD IO = 3.3V, IOH = 2mA	3.0	-	-	
$I_{OL}$	Output low current <sup>ξ</sup>	VDD IO = 1.8V, VO = 0.4V	12	-	-	mA
		VDD IO = 2.7V, VO = 0.4V	17	-	-	
		VDD IO = 3.3V, VO = 0.4V	18	-	-	
$I_{OH}$	Output high current <sup>ξ</sup>	VDD IO = 1.8V, VO = 1.4V	10	-	-	mA
		VDD IO = 2.7V, VO = 2.3V	16	-	-	
		VDD IO = 3.3V, VO = 2.3V	17	-	-	
$R_{PU}$	Pullup resistor	-	50	-	65	kΩ
$R_{PD}$	Pulldown resistor	-	50	-	65	

<sup>α</sup> - Hysteresis enabled;

<sup>β</sup> - Default drive strength (8mA)

<sup>ξ</sup> - Maximum drive strength (16mA).

Основні характеристики цифрових входів / виходів по змінному струму наведено в таблиці 2.4

Таблиця 2.4 - Характеристики цифрових входів/виходів змінного струму

Pin Name	Symbol	Parameter	Minimum	Typical	Maximum	Unit
Digital outputs	$t_{rise}$	10-90% rise time <sup>α</sup>	-	1.6	-	ns
Digital outputs	$t_{fall}$	90-10% fall time <sup>α</sup>	-	1.7	-	
GPCLK	$t_{JOSC}$	Oscillator-derived GPCLK cycle-cycle jitter (RMS)	-	-	20	ps
GPCLK	$t_{JPLL}$	PLL-derived GPCLK cycle-cycle jitter (RMS)	-	-	48	

<sup>α</sup>- Default drive strength, CL = 5pF, VDD IOx = 3.3V

Загальний вид цифровий сигнал входу/виходу представлено на рисунку 2.3



Рисунок 2.3 - Цифровий сигнал входу/виходу

## 2.3 GPIO

GPIO (General Purpose Input Output) - це низькорівневий інтерфейс ведення-вивід прямого керування який дозволяє підключати різні модулі й датчики (рисунок 2.1 обз.3). Raspberry Pi 3 має на борті 40-пинову рейку GPIO але так як 12 з них представляють із себе піни живлення 3.3 В, 5 У і загальні піни GND. [113] Призначення пінов роз'єма GPIO представлено на рисунку 2.4

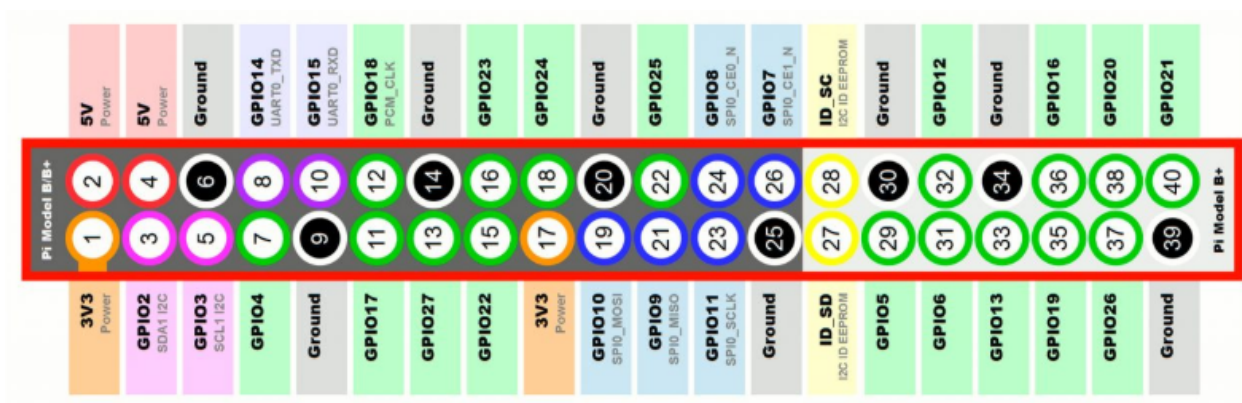


Рисунок 2.4- Призначення пінів роз'єму GPIO на Raspberry Pi 3

Також 27 (BCM 0) і 28 (BCM 1) піни використовуються для конфігурації EEPROM Raspberry Pi 3 для роботи з Нат-Обладнаннями (Hardware Attached on Top — обладнання поверхневого монтажу, по суті — звичайні плати розширення) і використання цих пінів у край не рекомендується. Проте вони є повноцінними Gpio-Пинами.

Існує кілька варіантів нумерації пінів Raspberry Pi 3: Board (фізична нумерація один по одному) і BCM (нумерація із чипа). Також з бібліотекою Wiringpi використовується своя нумерація яка представлено на рисунку 2.5



*Для запобігання проблем, при написанні програм, необхідно явно вказувати який режим нумерації буде використаний.*

WiringPi		BCM		Board		WiringPi	
3,3 V Power		1	2			5 V Power	
WiringPi 8	BCM 2	3	4			5 V Power	
WiringPi 9	BCM 3	5	6			5 V Power	
WiringPi 7	BCM 4	7	8	BCM 14	WiringPi 15	GND	
GND		9	10	BCM 15	WiringPi 16		
WiringPi 0	BCM 17	11	12	BCM 18	WiringPi 1		
WiringPi 2	BCM 27	13	14			GND	
WiringPi 3	BCM 22	15	16	BCM 23	WiringPi 4		
3,3 V Power		17	18	BCM 24	WiringPi 5		
WiringPi 12	BCM 10	19	20			GND	
WiringPi 13	BCM 9	21	22	BCM 25	WiringPi 6		
WiringPi 14	BCM 11	23	24	BCM 8	WiringPi 10		
GND		25	26	BCM 7	WiringPi 11		
WiringPi 30	BCM 0	27	28	BCM 1	WiringPi 31		
WiringPi 21	BCM 5	29	30			GND	
WiringPi 22	BCM 6	31	32	BCM 12	WiringPi 26		
WiringPi 23	BCM 13	33	34			GND	
WiringPi 24	BCM 19	35	36	BCM 16	WiringPi 27		
WiringPi 25	BCM 26	37	38	BCM 20	WiringPi 28		
GND		39	40	BCM 21	WiringPi 29		

Рисунок 2.5- Варіант нумерації пінів Raspberry Pi 3: Board и BCM

Кожний з 28 пінів може бути встановлений у режим цифрового виходу OUTPUT, і в режим цифрового входу INPUT як показано на рисунку 2.6.

Максимальний вихідний струм кожного піна не повинен перевищувати 16 мА.

$$\max I_{out} = 16mA$$

Сумарний вихідний струм усіх пінів не повинен перевищувати 50 мА

$$\max \sum I_{out} = 50mA$$

5-вольтові піни можуть давати більший струм, який залишається після живлення Raspberry Pi 3 і інших периферійних обладнань (клавіатури, миші) — до 500 мА.

$$\max \sum I_{5v-out} = 500mA$$

За замовчуванням усі піни (крім BCM14 і BCM15) перебувають у режимі INPUT, причому піни BCM0-BCM8 і BCM15 підтягнуті до одиниці резисторами, що підтягують (pullup). Саме із цієї причини мультиметр покаже напругу на цих пинах. Інші піни стягнуті до нуля. Кожний з 28 пінів має резистори, що підтягують (pullup) і стягують (pulldown), завдяки чому, у режимі INPUT може бути підтягнутий до логічної одиниці, або стягнутий до нуля.

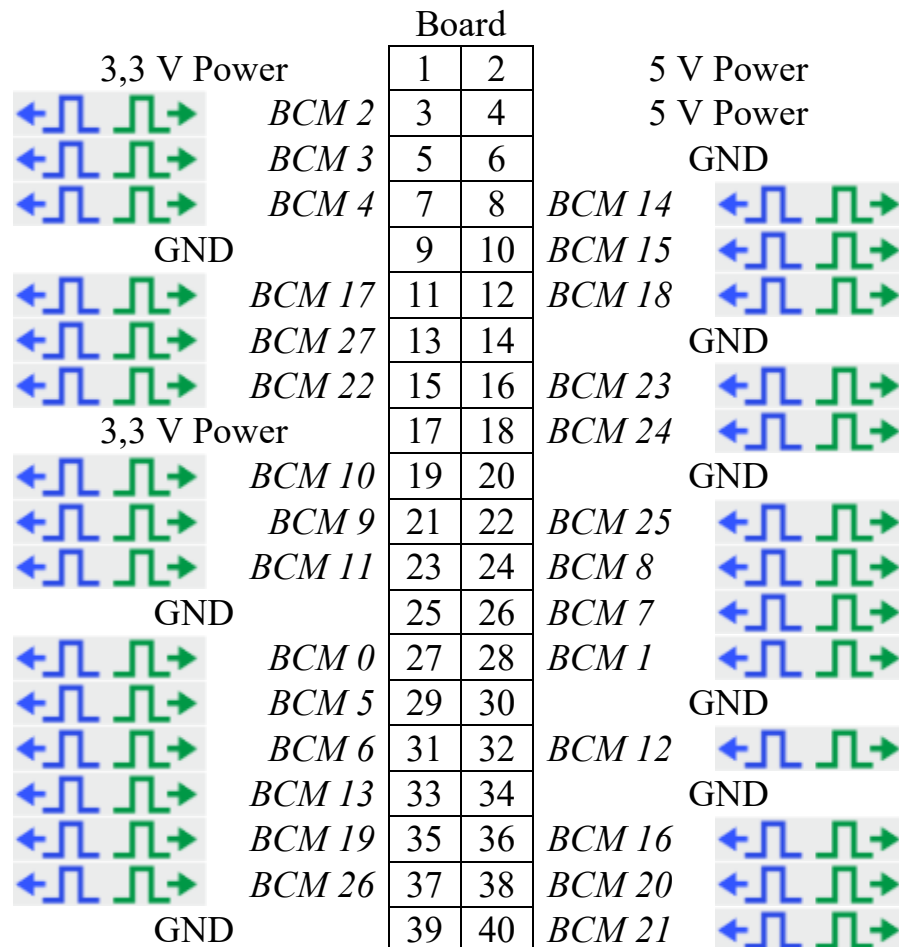


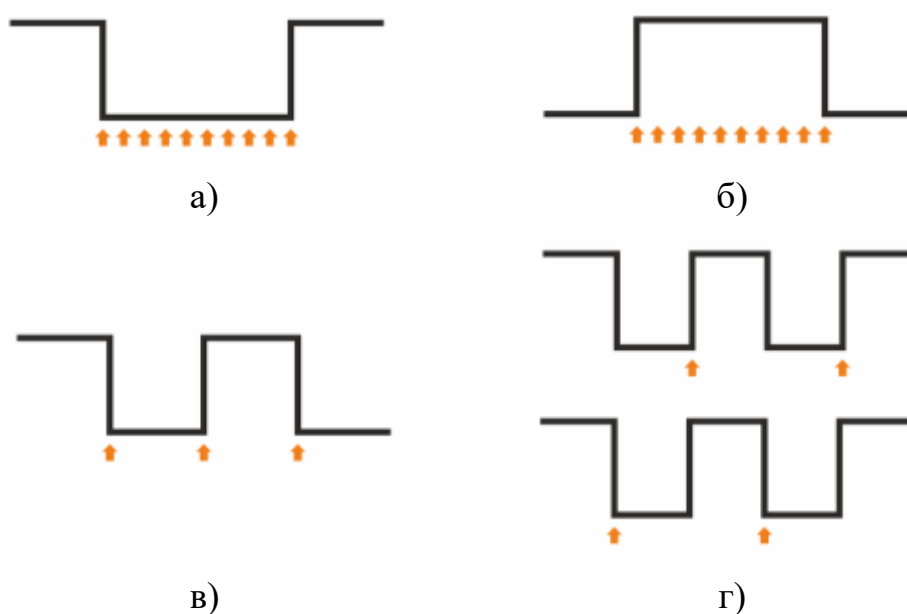
Рисунок 2.6 - Режими роботи роз'ємів GPIO

Номінали опорів не постійні й рівні:

- для резистора, що підтягує 50 КОм — 65 КОм;
- для стягуючого резистора 50 КОм — 60 КОм.

Номінал опору, що підтягує/стягуючого резистора для пінів ВСМ2 і ВСМ3 -1.8 Ком.

Кожний з 28 пінів у режимі INPUT може генерувати переривання - по спаду, по фронту, по одиниці, по нулю, по зміні сигналу, а також в асинхронному режимі по фронту й по спаду як показано на рисунку 2.7



а) low; б) high; в) change; г) async.

Рисунок 2.7. – Режими роботи пінів зі станів INPUT

Також усі піни в режимі INPUT мають вхідну фільтрацію на тригері Шмитта (перетворюють аналоговий сигнал у цифровий з різкими переходами між станами) як показано на рисунку 2.8.



*Вхідний реальний сигнал до триггера Шмітта*

*Вхідний сигнал після триггера Шмітта*

Рисунок 2.8 – Приклад фільтрацій на триггери Шмітта

Raspberry Pi 3 не має аналогових входів/виходів. Для реалізації подібного функціонала потрібно використовувати зовнішні АЦП/ЦАП, наприклад, АЦП ADS1115 (I<sup>2</sup>C) або АЦП MCP3008

## 2.4 Послідовний інтерфейс I<sup>2</sup>C

Інтерфейс I<sup>2</sup>C (I2C або TWI) був розроблений компанією Philips для внутрішнього зв'язку периферії в побутовій електроніці. Для передачі даних і адресації використовуються всього два провідення. При першому, швидкому погляді на структуру кадра й опис в Datasheet, видається що інтерфейс I<sup>2</sup>C дуже складний і відразу в структурі його кадра не розібратися. Також думав і я, коли вперше з ним познайомився, але в підсумку полюбив працювати із цим інтерфейсом більше чому з SPI.[114]

У порівнянні з інтерфейсом SPI, в I<sup>2</sup>C є такі позитивні моменти:

- більш наочна структура кадра;
- можливість відстеження виникаючих помилок при прийманні й передачі;
- можливість використання переривання, при вступі даних по шині I2C у той час, коли мікроконтролер перебуває в сплячому режимі;
- для вибору обладнання не потрібно задіяти додаткові висновки мікроконтролера;
- повільні обладнання самостійно знижують швидкість шини.

Так і негативні моменти:

- низька швидкість передачі даних, до 400 кбит/з (а якщо врахувати ще, що спочатку передається адреса обладнання, те реальна швидкість ще нижче);

- обмеження по числу однакових обладнань, до адресуються (звичайно виробники периферії із шиною I<sup>2</sup>C залишають 3 біта адреси обладнання на вибір користувача, а значить максимум можна використовувати 8 однакових обладнань).

До двухпровідної шини I<sup>2</sup>C усі обладнання підключаються паралельно й оба дрота підтягуються до шини живлення. По лінії SDA передаються дані, а по SCL здійснюється тактування. Приклад підключення по шині I<sup>2</sup>C представлено на рисунку 2.9

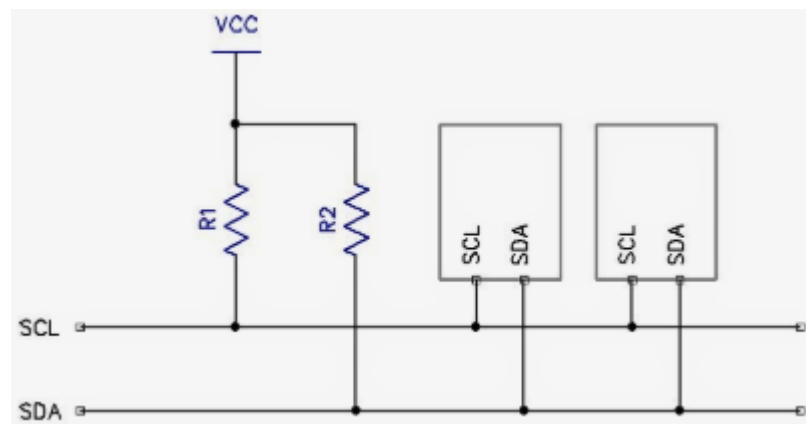


Рисунок 2.9 - Схема підключення резисторів, і обладнань що підтягують, до шини I<sup>2</sup>C

Номінал резисторів, що підтягують, залежить від довжини шини, а також від кількості підключених обладнань. Якщо використовувати пару обладнань, що підключаються до шини, не вважаючи мікроконтролер, і довжину лінії до 30 см, то можна використовувати резистори номіналом 10 кому.

Щоб почати передачу обладнання, що веде (Raspberі Pi) повинен послати на шину I<sup>2</sup>C сигнал "START" - спочатку низький рівень установлюється на лінії SDA, а потім на лінії SCL. Далі впливає передача

адреси й даних. По закінченню передачі на шину посилає сигнал "STOP" - спочатку високий рівень установлюється на шині SCL, а потім на шині SDA. Форма сигналу "START" і "STOP" представлено на рисунку 2.10

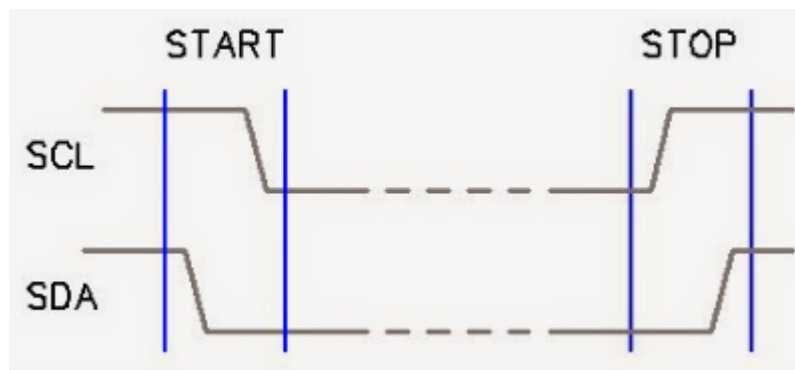
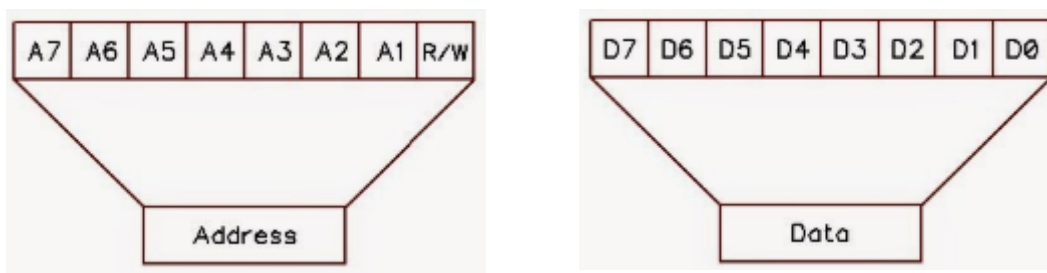


Рисунок 2.10 - Сигнали "START", "STOP"

Адреса й дані передаються пакетами по 8 біт. У пакеті адреси 1 біт використовується для інформування веденого обладнання, чи будемо ми з нього зчитувати (1) дані або ж записувати (0). На рисунку 2.11 представлений формат пакету адрес і даних.



а)

б)

а) фрагмент пакету адрес;

б) фрагмент пакету даних;

Рисунок 2.11 - Формат пакетів адрес і даних.

До кожного пакету додається ще один біт - підтвердження (ACK) або не підтвердження (NACK). При передачі адреси, а також записи даних, цей біт передає ведене обладнання. При читанні - провідне (Raspberry Pi).

У режимі *Master Write* мікроконтролер після команди "START" передає адреса обладнання, у яке прагне записати дані (установлюючи біт читання/запису в 0). Якщо на шині присутнє обладнання з такою адресою, то воно посилає відповідь "ACK", і далі впливає пересилання даних від провідного обладнання. Якщо обладнання з такою адресою немає, то передача закінчується. Після закінчення передачі даних провідне обладнання посилає в шину команду "STOP". Режим Master Write представлено на рисунку 2.12

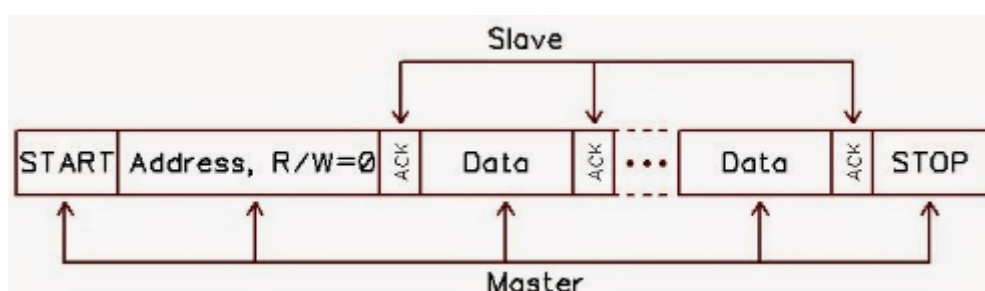


Рисунок 2.12 - Режим Master Write.

У режимі *Master Read* мікроконтролер після послу команди "START" передає адресу обладнання, з якої прагне вважати дані (установлюючи біт читання/запису в 1). Якщо на шині присутнє обладнання з такою адресою, то воно посилає відповідь "ACK", і далі впливає пересилання даних від веденого обладнання, яке підтверджується провідним обладнанням. Після прийняття останнього байта даних від веденого обладнання, що веде посилає відповідь "NACK" Якщо обладнання з такою адресою немає, то передача закінчується. Після закінчення передачі даних провідне обладнання посилає в шину команду "STOP". Режим Master Read представлено на рисунку 2.13

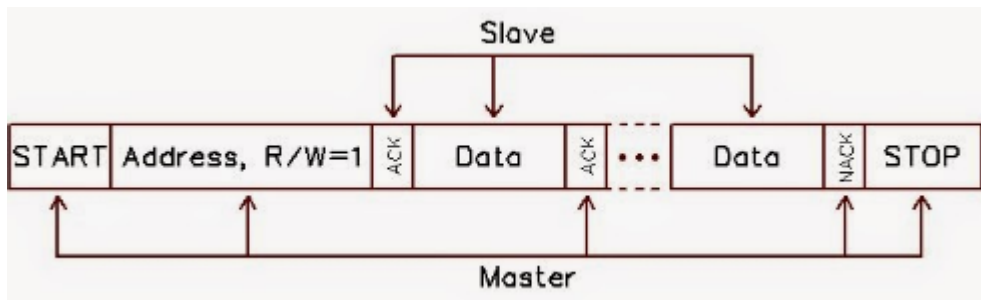


Рисунок 2.13 - Режим Master Read



Однією із цікавих особливостей передачі даних по протоколу I<sup>2</sup>C є здатність ведених обладнань знижувати швидкість передачі даних. Якщо швидкість висока, то ведене обладнання втримує лінію тактування (SCL) у низькому стані, і відпускає її коли готове прийняти наступний біт

## 2.5 Послідовний інтерфейс SPI

SPI (Serial Peripheral Interface) or (послідовний периферійний інтерфейс) - це синхронна чотирьохпровідна шина. Вона являє собою з'єднання двох синхронних зсувних регістрів, які є центральним елементом будь-якого SPI обладнання. Для з'єднання використовується конфігурація ведучий/ведений. Тільки ведучий може генерувати імпульси синхронізації. У схемі завжди тільки один ведучий ( на відміну від тієї ж шини I<sup>2</sup>C, де можливий варіант із більш ніж одним ведучим), кількість ведених може бути по-різному. У загальному випадку вихід ведучого з'єднується із входом веденого, і навпаки, вихід веденого з'єднується із входом ведучого. При подачі імпульсів синхронізації на вихід SCK, дані виштовхуються ведучим з виходу MOSI, і захоплюються веденим по входу MISO. У такий спосіб якщо подати кількість імпульсів синхронізації відповідно до розрядності сдвигового регістру, то дані в регістрах обмінюються місцями. Звідси випливає що SPI завжди працює в повнодуплексному режимі.[115]

Контролер SPI, як правило, реалізується периферійним блоком в MCU або етпру. У більшості чипів він може працювати як у режимі провідного, так і в режимі веденого. Існує кілька способів включення SPI обладнань, але в даному розділі ми розглянемо найпоширеніший «Каскадна схема» підключення SPI обладнань який представлено на рисунку 2.14.

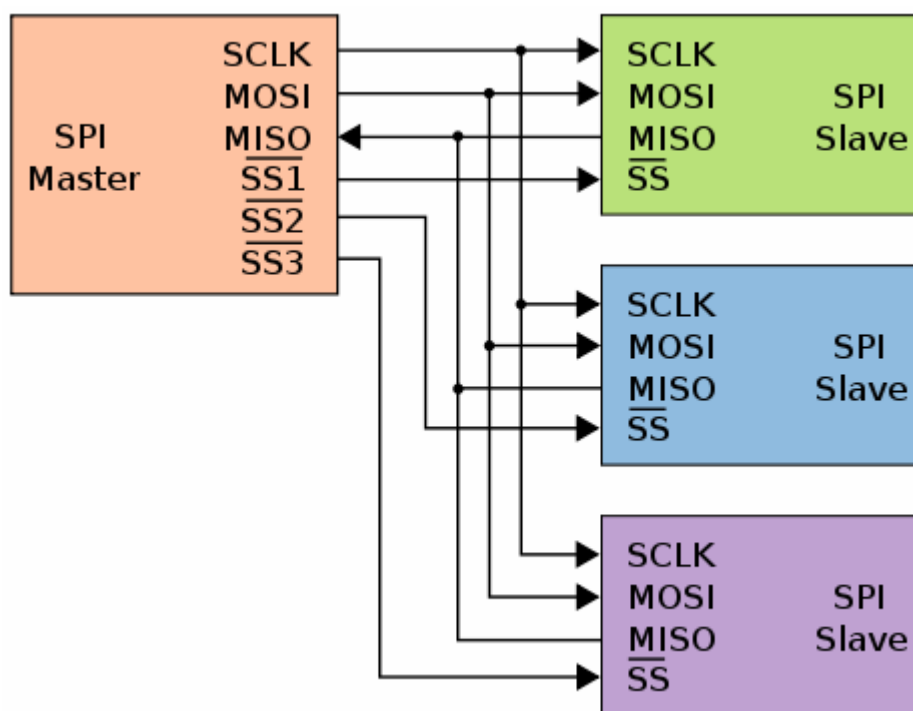


Рисунок 2.14 - Каскадна схема підключення SPI обладнань.

У цьому випадку до провідного всі ведені підключаються паралельно, за винятком сигналу вибору веденого ( $\sim CS$ ). Для кожного веденого необхідний окремий сигнал вибору веденого (на рисунку вони позначені як  $SSx$ ). Для сигналів вибору веденого можуть використовуватися як спеціально призначені для цього виходи Spi-Контролера, так і порти введення/вивід загального призначення (GPIO) Raspberry Pi.

Два провідники використовуються для передачі даних, один для подачі тактових імпульсів і по одному сигналу вибору веденого для кожного з ведених. У каскадній схемі підключення SPI обладнань використовується наступний набір сигналів:

MOSI - Master Output, Slave Input (вихід провідного, вхід веденого). Даний сигнал призначений для послідовної передачі даних від ведучого до веденого. Також може називатися SDO, DO і т.п.;

MISO - Master Input, Slave Output (вхід провідного, вихід веденого). Даний сигнал призначений для послідовної передачі даних від веденого до ведучого. Може називатися SDI, DI і т.п.;

SCK - Serial Clock (сигнал синхронізації). Використовується для синхронізації при передачі даних. Також може мати назва SCLK, CLK і ін.;

$\sim$ CS - Chip Select (вибір мікросхеми). За допомогою даного сигналу відбувається активація веденого обладнання. Звичайно він є інверсним, тобто низький рівень вважається активним. Іноді його називають  $\sim$ SS (Slave Select, рос. «вибір веденого»).

Окремим випадком незалежного підключення є варіант із одним єдиним веденим. У такому випадку може виникнути бажання підтягти сигнал  $\sim$ CS до землі, щоб обладнання завжди було в активному стані. Але робити це вкрай не рекомендується, тому що ведене обладнання може використовувати сигнал CS для ініціалізації або для інших службових цілей.

Існує чотири режими роботи SPI обладнань. Дані чотири режими являють собою комбінацію двох біт:

CPOL (Clock Polarity) — визначає початковий рівень (полярність) сигналу синхронізації. CPOL=0 показує, що сигнал синхронізації починається з низького рівня, так що передній фронт є наростаючим, а задній — падаючим. CPOL=1, сигнал синхронізації починається з високого рівня, у такий спосіб передній фронт є падаючим, а задній — наростаючим;

CPHA (Clock Phase) — фаза синхронізації, визначає по якому із фронтів синхронізуючого сигналу робити вибірку даних. CPHA=0 показує що необхідно робити вибірку по передньому фронту, а CPHA=1 показує що вибірку даних необхідно робити по задньому фронту



*Ці два біти й утворюють номер режиму. CPOL є старшим бітому, а CPHA — молодшим. Іноді в документації до обладнання явно не вказують номер режиму, але його завжди можна легко визначити по тимчасових діаграмах.*

## 2.6 Послідовний порт UART

UART (Universal Asynchronous Receiver/Transmitter - універсальний асинхронний приймач/передавач) - протокол послідовного зв'язку, у якому дані передаються послідовно, тобто побітно. При такому зв'язку за один раз передається байт даних.

Існує велика кількість різноманітних протоколів зв'язку, призначених для використання з UART. Найбільш відомі:

- MODBUS - сімейство протоколів типу «запит-відповідь», популярне в промисловій автоматичі. Орієнтовані на управління обладнанням за допомогою коротких команд;

- AT-Команди телефонних модемів. Набір текстових команд, що дозволяють управляти роботою модему при встановленні з'єднання;

- PPP – протокол, який широко використовувався при підключеннях до інтернету через модем. Дозволяє виконувати аутентифікацію користувача в провайдера, шифрувати дані тощо;

- IRDA - сімейство протоколів фізичного рівня для оптичного бездротового керування.

Протокол UART використовує певну структуру кадрів для передачі байту даних. Структура в асинхронному повідомленні наступна:

- біт START - вказує, що послідовний зв'язок почався, і цей біт завжди має низький рівень сигналу;

- пакет бітів даних - бітами даних можуть бути пакети від 5 до 9 біт. Як правило, використовується 8-бітний пакет даних, який завжди надсилається після START-біта;

- біт STOP - це, зазвичай, один або два біти. Відправляється після пакетів бітів даних, щоб вказати кінець кадру. Біт STOP завжди має високий логічний рівень.[116]

На рисунку 2.15 представлена структура основного кадру протокол UART.

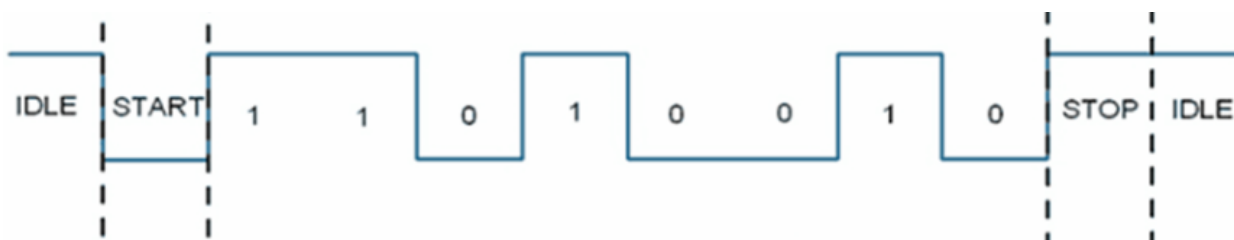


Рисунок 2.15 - Структура основного кадру протокол UART

Кадр може також мати біт PARITY, який передається після STOP-біта. UART зазвичай використовується в Raspberry Pi (RPi) як зручний спосіб керування через GPIO або для доступу до завантажувальних повідомлень ядра з послідовної консолі (включено за замовчуванням). Він також може бути використаний як спосіб для взаємодії з Arduino, завантаженого ATmega, ESP8266 та інших мікроконтролерів з RPi.



*Будьте обережні з рівнями логіки між пристроями, бо, наприклад, RPi має 3,3 В, а Arduino – 5 В. Підключивши їх разом ви можете їх пошкодити.*

Raspberry Pi має два вбудованих UART:

- PL011 UART;
- mini UART.

PL011 UART - це UART на базі ARM. Він має кращу пропускну здатність, ніж mini UART. Хоча RPi має два UART, але ми маємо лише одну пару вивідів TXD і RXD для роботи.

UART PL011 є основним UART для моделей без функції Bluetooth і зв'язаний безпосередньо з консоллю Linux. Це означає, що через цей UART ви можете надсилати команди Linux з вашого ПК на RPi. З іншого боку, mini UART є Linux консоллю UART для моделей з Bluetooth, такими як Raspberry

Pi 3 і Raspberry Pi Zero W. Для цих моделей UART PL011 зв'язаний безпосередньо з модулем Bluetooth.

UART PL011 є більш надійним, ніж міні UART, оскільки останній має менші за обсягом FIFO, не може контролювати потік, а швидкість передачі даних залежить від тактової частоти GPU. Оскільки частота ядра графічного процесора змінюється, то змінюється і частота UART, що, в свою чергу, змінює швидкість передачі даних через UART. Це робить міні UART нестабільним, що може призвести до втрати даних або їх пошкодження. Щоб зробити міні UART стабільним, треба змінити основну частоту. Також міні UART не підтримує перевірку парності.



*Для кращого та ефективного зв'язку рекомендується використовувати PL011 UART замість міні UART.*

## 2.7 Контрольні питання

1. Перерахуйте основні елементи Raspberry Pi 3 Model B Plus;
2. Які Периферійная пристрої та протоколи підтримує Raspberry Pi 3 Model B Plus ?;
3. Характеризується робочий Діапазони харчування Raspberry Pi 3 Model B Plus;
4. Намалюйте загальний вигляд цифровий сигнал входу / виходу;
5. Що таке шина GPIO ?;
6. Які типи пинов і їх призначення разьма GPIO ?;
7. Як проводиться нумерації пинов GPIO ?;
8. У яких режимах може бути встановлено піни в GPIO?;
9. Принцип фільтарції на тригер Шміта ?;
10. Дайте характеристику послідовному інтерфейсу I<sup>2</sup>C;
11. Опишіть формат пакетів адрес і даних для I<sup>2</sup>C;
12. Як працює режим Master Write для послідовного інтерфейсу I<sup>2</sup>C;

13. Як працює режим Master Read для послідовного інтерфейсу I<sup>2</sup>C ?;
14. Дайте характеристику послідовному інтерфейсу SPI;
15. Для чого потрібен імпульс синхронізації (SCK) в SPI ?;
16. Намалюйте і опишіть схему каскадного підключення через SPI;
17. Дайте характеристику послідовного порту UART;
18. Привид приклад структури основного кадру протокол UART;
19. Скільки портів UART має Raspberry Pi 3 mobil B +.

### 3. УСТАНОВКА Й НАЛАШТУВАННЯ ОПЕРАЦІЙНОЇ СИСТЕМИ RASPBIAN

#### 3.1 Установка OS Raspbian за допомогою NOOBS

Raspberry Pi (RPi) — це одноплатний мікрокомп'ютер, у якого є всі ті ж ознаки, що й у звичайних персональних комп'ютерів і ноутбуків. До нього можна підключити монітор, клавіатуру, мишу, аудіо колонку, а також інтернет кабель. Як і персональний комп'ютер, RPi працює під керуванням повноцінних операційних систем (ОС), таких як: Raspbian (Debian), Android і навіть Windows 10(IoT). операційна система в Rpi зберігається на карті пам'яті формату microSD. Там же зберігаються й усі користувацькі файли. Щоб почати роботу з Raspberry Pi, нам необхідно встановити на карту пам'яті операційну систему. Поділимо даний підрозділ на *основні кроки*. [117]

**Крок 1** - рекомендується використовувати карту пам'яті розміром не менш 8 Гб. Самої популярної ОС для Raspberry Pi вважається Raspbian. Це по суті модифікована Debian. Є два варіанти установки Raspbian на карту пам'яті:

- копіювання образу карти пам'яті, із передвстановленим Raspbian; робиться це за допомогою Win32Diskimager.

- копіювання спеціального установника NOOBS, який в автоматичному режимі встановить Raspbian або іншу ОС.

**Крок 2** – вантажимо архів (<https://www.raspberrypi.org/downloads/noobs/>) файлами з образами NOOBS Можна вибрати NOOBS або NOOBS Lite. У першому випадку ми завантажимо установник у комплекті з Raspbian. У другому випадку установник буде порожній, але можна буде «дозавантажити» будь-яку ОС із інтернету. Вибираємо перший варіант.

**Крок 3** - Розпаковуємо завантажений архів.

**Крок 4** - Копіюємо всі файли з архіву на порожню карту пам'яті.

**Крок 5** - Запуск Raspberry Pi. Усі Raspberry Pi починають своє завантаження з райдужного екрана як показано на рисунку 3.1

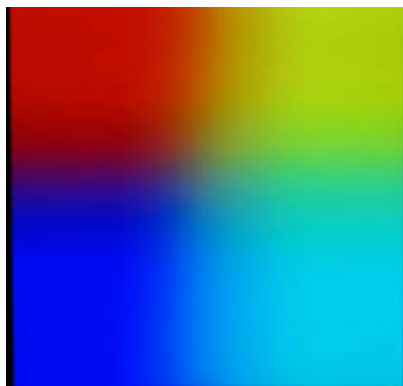


Рисунок 3.1- Райдужний екран завантаження.

**Крок 6** - Запуск установника NOOBS. Після завантаження, установник просить вибрати потрібну операційну систему. Вибираємо Raspbian як показано на рисунку 3.2.

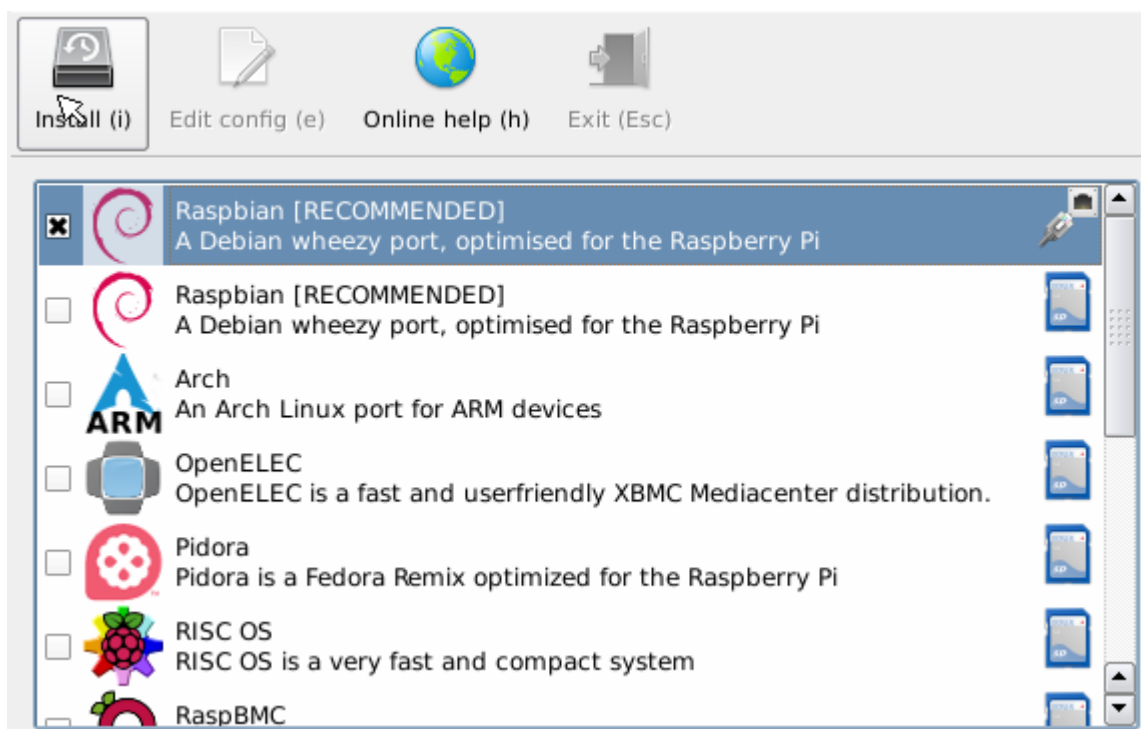


Рисунок 3.2.- Установник NOOBS.

Підтверджуємо установку ОС Raspbian натисканням кнопки «Yes».

**Крок 6** - Копіювання файлів операційної системи (рисунок 3.3). На цьому кроці нам потрібно просто дочекатися, поки всі копіюються. Це займає хвилин 15-20.

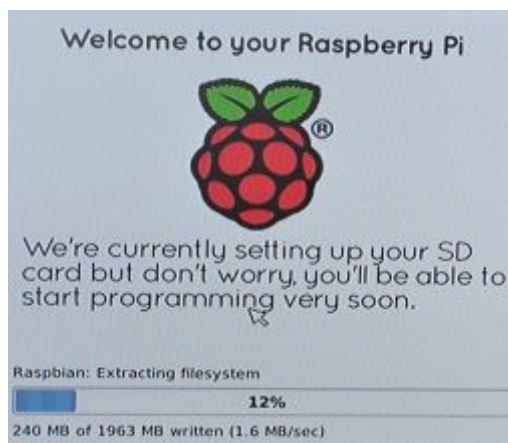


Рисунок 3.3.- Копіювання файлів ОС Raspbian

**Крок 7** – По завершенню копіювання й установки на екрані буде виведене вікно представлене на рисунку 3.4. Натисніть «ОК».



Рисунок 3.4 - Вікно успішної установки ОС Raspbian.

Після виконання всіх кроків, на Raspberry Pi успішно встановлений ОС Raspbian і можна працювати з робочим столом, інтерфейс якого представлено на рисунку 3.5.

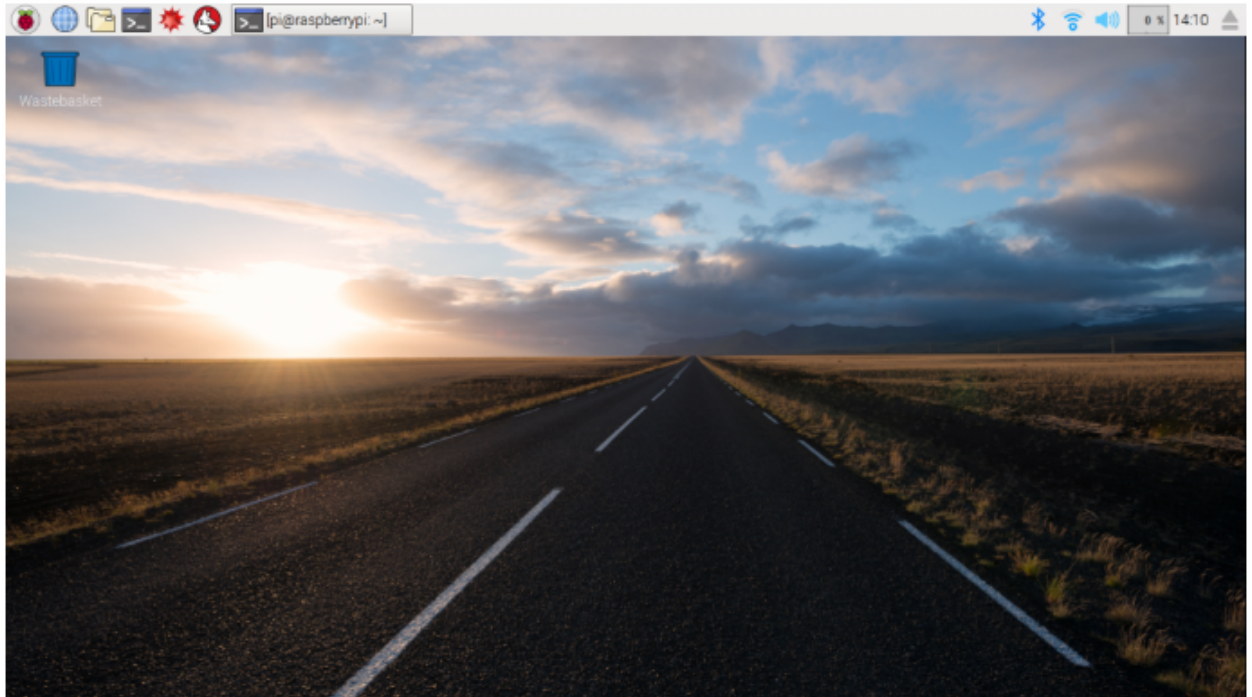


Рисунок 3.5 – Інтерфейс робочого стола ОС Raspbian

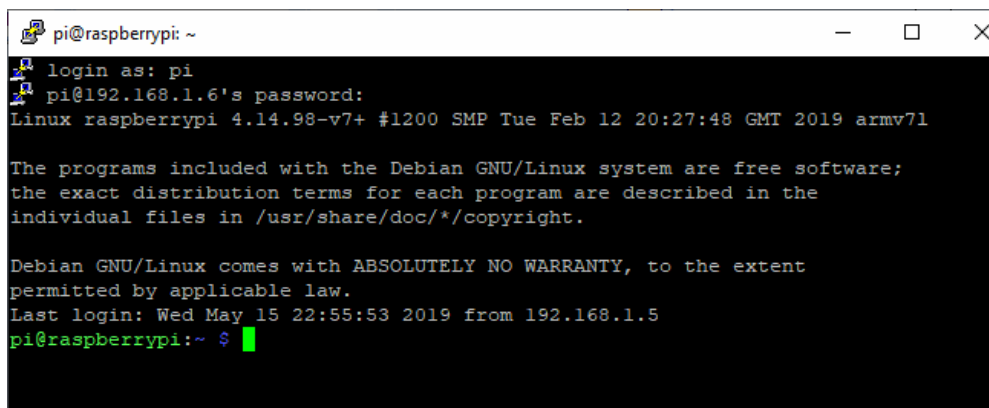
Властиво, по завершенню процедури установки ОС Raspbian, Raspberry Pi повністю готовий до роботи. У системі вже встановлені різні програмні додатки й середовища розробок, серед них є:

- мова програмування python версій 2 і 3;
  - візуальна мова програмування Scratch;
  - засоби розробки Java додатків: Bluej Java IDE, Greenfoot Java IDE;
  - засіб розробки Geany Programmer's Editor;
  - пакет Mathematica;
  - пакет офісних додатків Libreoffice;
  - Vncviewer;
  - браузер Chromium;
  - поштовий клієнт Claws Mail;
- і навіть гру Minecraft Pi.

## 3.2 Базові команди Linux для Raspberry Pi

При настроюванні й роботі з Raspberry Pi по необхідності прийдеться часто користуватися командним рядком або *Terminal*. Більшість початківців користувачів Raspberry Pi стараються мінімізувати використання *Terminal* представлено на рисунку 3.6 і даний розв'язок є в корені невірним.[118]

Для того, щоб користуватися терміналом, досить знати кілька базових команд. Після того, як ви встановите підключення по протоколу SSH, ці команди дозволять вам прямо з комп'ютера встановлювати на Pi нове ПО, створювати файли/папки й запускати будь-які скріпти без використання графічної оболонки Raspbian OS.



```
pi@raspberrypi: ~
login as: pi
pi@192.168.1.6's password:
Linux raspberrypi 4.14.98-v7+ #1200 SMP Tue Feb 12 20:27:48 GMT 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 15 22:55:53 2019 from 192.168.1.5
pi@raspberrypi:~ $
```

Рисунок 3.6 - Terminal на базі вилученого з'єднання через Putty

На першому кроці вивчимо команду *pwd*. У результаті термінал напише */home/pi*. Прямі слеші завжди використовуються для позначення папок і файлів, що перебувають усередині папок. У цьому випадку поточною робочою директорією є *pi*, яка перебуває в директорії *home*, яка перебуває в корені файлової системи.

Слово *pi*, що розміщується перед *@* – це ім'я користувача, під яким пользователь зайшов у систему.



Команди Linux чутливі до регістру, і це значить, що **PWD**, **Pwd**, **pwd** та інші варіації не будуть ідентичні **pwd**. Те ж саме стосується й усіх інших команд Linux.

Найбільш часто використовувані команди – це *ls* і *cd*:

*ls* - використовується, щоб показати вміст директорії;

*cd* - дозволяє переміститися з однієї директорії в іншу.

Завантаживши термінал у перший раз, користувач опиниться в домашній директорії (/home/pi). Отже, щоб надрукувати список файлів і папок, що перебувають у робочій директорії, необхідно в термінал задати наступну команду (*ls*):

```
pi@raspberrypi:~ $ ls
```

Результат виконання даної команди представлено на рисунку 3.7

```
pi@raspberrypi:~ $ ls
Cam_Py          myvideo.h264          teamviewer-host_armhf.deb.1
Desktop         oldconffiles          Templates
Documents       Pictures              timestamped.h264
Downloads       Public                Videos
filename.jpg    python_games          картинка.jpg
LCD-show        teamviewer-host_14.2.8352_armhf .deb  фильм.h264
Music           teamviewer-host_armhf.deb
pi@raspberrypi:~ $
```

Рисунок 3.7 - Результат виконання команди *ls* у директорій /home/pi.

Для створення нової директорії необхідно ввести команду *mkdir* синтаксис команди представлений нижче:

```
mkdir NewFolder
```

*NewFolder* – ім'я директорій

Для навігації по файловій системі використовується команда *cd*, після якої через пробіл вказується назва директорій, у яку необхіднопереміститися.

```
pi@raspberrypi:~ $ cd NewFolder
```

```
pi@raspberrypi:~/NewFolder $
```

Ця команда дає можливість перемістити користувача в папку *NewFolder*.



При роботі з терміналом можна користуватися одним хитрим прийманням, що працює за принципом автодоповнення тексту. Для цього використовується клавіша “Tab” яка підсвічує назвами директорій і файлів по початкових символах.

Для створення нового файлу використовується команда *touch*, синтаксис написання даної команди представлений нижче.

```
pi@raspberrypi:~/Newfolder $ touch Newfile.txt
```

Окремі файли можна копіювати за допомогою команди *cp*, після якої через пробіл пишеться ім'я файлу. Крім того, цією командою можна користуватися для перейменування файлів:

```
pi@raspberrypi:~/Newfolder $ cp Newfile.txt Otherfile.txt
```

Видалення файлу здійснюється за допомогою команди *rm*, після якої через пробіл пишеться ім'я файлу, що віддаляється:

```
pi@raspberrypi:~/Newfolder $ rm Newfile.txt
```

Переміщення файлів здійснюється за допомогою команди *mv*

```
pi@raspberrypi:~/Newfolder $ mv Otherfile.txt /home/pi
```

За замовчуванням в OS Raspbian і багато інші дистрибутивів Linux вбудований простий текстовий редактор за назвою *nano*. Для відкриття файлу в редакторі *nano* використовується наступний синтаксис:

```
pi@raspberrypi:~ $ nano namefile.txt
```

У результаті *nano* перейде по зазначеному шляхові й відкриє файл, якщо той існує. Якщо його не існує, *nano* запустить у цій директорії новий буфер із цією назвою файлу. Приклад виконання команди представленої вище показано на рисунку 3.8.

Рисунок 3.8 - Приклад виконання команди *nano namefile.txt*

У верху редактора *nano* написана версія програми, назва й розширення файлу, що редагується, а також інформація, чи був цей файл модифікований після останнього збереження.



*Якщо ви працюєте з новим файлом, який ще не був збережений, то замість назви файлу побачите напис «New Buffer». Далі буде перебувати вміст вашого файлу.*

В двох самих нижніх рядках перебувають підказки для деяких гарячих клавіш які наведено на рисунку 3.9

Рисунок 3.9 - Підказка гарячих клавіш редактори *nano*

Редактор *nano* підтримує різні функції начебто збереження, виходу, пошуку і т.п. Гарячі клавіші для найпоширеніших функцій показані в нижній частині екрана (див. рисунок 3). Символ «^» означає клавішу *Ctrl*.



Клавіша *Shift* для гарячих клавіш не використовується. Крім того, усі гарячі клавіші використовують нижній регістр і незмінні цифрові клавіші, так що *Ctrl+G* – це не *Ctrl+Shift+G*. Для виклику підказки в редакторі *nano* натисніть комбінацію клавіш *Ctrl+G*.

При необхідності завантаження й установки нового програмного забезпечення на Raspberry Pi і його аналоги працюючі на основі OS Linux відбувається методом «*пакетів*» – програм, які можна завантажити з інтернету й установити, просто написавши в терміналі невелику команду.

Для завантаження й установки цих пакетів використовується, як правило, пакетний менеджер, який завантажує й установлює не тільки саме ПО, але й допоміжні програми, які називаються «*залежностями*».

Дистрибутив OS Raspbian використовує пакетний менеджер за назвою **apt**. Щоб управляти установкою/видаленням ПО, користувачеві необхідно авторизуватися із правами адміністратором (його також називають «суперкористувачем»). Для цього на початку команди потрібно вписати *sudo* (це скорочення від «superuser do»).

Щоб установити ПО, необхідно використовувати наступний синтаксис:  
*sudo apt-get install <назва ПО>*

При необхідності запуску ПО прямо з термінала, використовується наступний синтаксис (*nameprogram* – назва ПО):

```
pi@raspberrypi:~ $ nameprogram
```

Щоб вилучити ПО з Raspberry Pi, використовується пакетний менеджер *apt*. Синтаксис написання даної команди представлений нижче:

```
sudo apt-get remove <назва ПО>
```

Ця команда, втім, не видаляє файли, пов'язані з вилученням ПО – начебто логів і конфігураційних файлів. При відсутності необхідності використовувати їх надалі, то можете скористатися командою *purge*, яка видаляє й саме ПО, і допоміжні файли:



Raspbian OS. Основний інструмент *raspi-config* дозволяє міняти наступні налаштування:

- Change User Password – зміни пароля користувача;
- Boot Options – вибір інтерфейсу, який увімкнеться при завантаженні обладнання;
- Internationalisation Options – тут настраюються годинний пояс, розкладка клавіатури, мова й регіональна налаштування;
- Overclock – зміна тактової частоти Raspberry Pi;
- Advanced Options – додаткові опції;
- Update -
- About raspi-config – інформація про інструмент raspi-config.

Перезавантаження Raspberry Pi виконується командою :

```
sudo reboot
```

Вимикання Raspberry Pi:

```
sudo poweroff
```



*Зверніть увагу, фізичне відключення живлення ( від джерела) рівнозначно тому, як якби ви вимкнули свій комп'ютер натисканням на кнопку Power або навіть відключенням від розетки що може привести до ушкодження файлів.*

### 3.3 Налаштування OS Raspbian

#### 3.3.1 Установка екранної клавіатури

Якщо ви використовуєте Raspberry PI разом із сенсорним дисплеєм, зручно використовувати екранну клавіатуру. Для її установки виконаємо наступну послідовність етапів.[119]



*Варто помітити що екранна клавіатура може бути вже предустановлена. І крок установки пакета може бути не потрібний.*

На першому етапі оновимо OS Raspbian через Terminal наступними командами:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

потім за допомогою команди

```
sudo apt-get install matchbox-keyboard
```

– проведемо установку пакета екранної клавіатури.

Однією з головних умов необхідних для відображення в *Menu* екранної клавіатури це підтвердити її статус *Visible*. Для цього необхідно викликати *Main Menu Editor* через *Menu* як презентовано на рисунку 3.11 Потім як на рисунку 3.12 обираємо виберіть пункт меню *Accessories* і активуємо «галочку» напроти *Keyboard*.

Після цього віртуальна клавіатура буде доступна всім користувачам на всіх вилучених робочих столах по наступним шляхам *Menu -> Accessories -> Keyboard*.

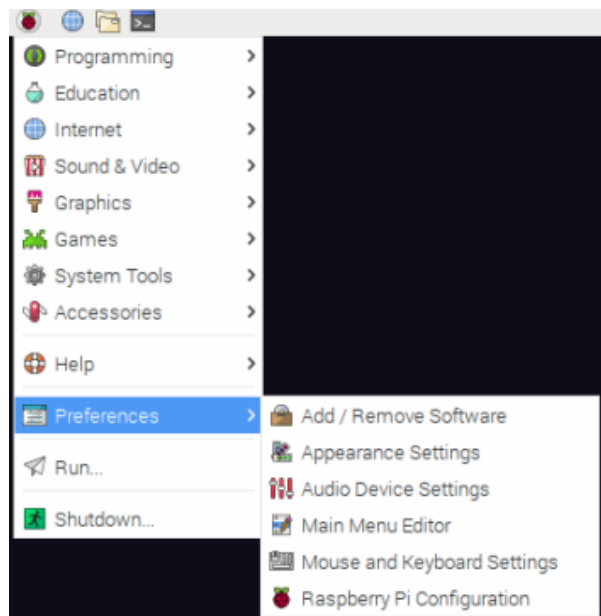


Рисунок 3.11 – Виклик *Main Menu Editor* через *Menu*.

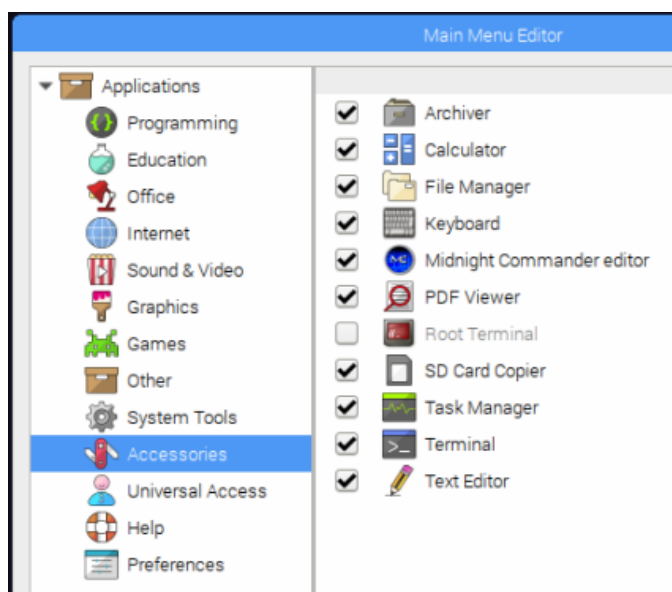


Рисунок 3.12 - Вікно *Main Menu Editor*.



*У підменю Accessories можна відразу налаштувати відображення потрібних програм для роботи з OS Raspbian. Це актуально коли використовуються сенсорні екрани з невеликим розширенням .*

### 3.4 Налаштування віддаленого доступу

#### 3.4.1 Налаштування віддаленого доступу через VNC

*Метод, представлений у данному підрозділі, дозволить здійснювати обмежений перелік дій, що включає роботу в терміналі. Він не дозволить запускати повноекранні графічні додатки, такі як ігри або медіаплеєри. Проте, він забезпечить повноцінне оточення, таке ж, як якби воно було виведено прямо на монітор.*

При розробці систем керування мобільними роботами, виникне необхідність управляти Raspberry Pi віддалено, будь це транслявання відео потоку з web-камери для спостереження за навколишнім простором, програмування елементів обробки й прийняття розв'язків на Python, або налаштування OS Raspbian. Отже на даний момент часу не існує більш зручного методу чому використання вилученого робочого стола. Наведена

послідовність дозволить налаштувати зв'язування Raspberri Pi 3 + ПК на Windows, що перебувають в одній локальній мережі. [120]

На першому етапі оновимо OS Raspbian:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Опцією *update* утиліта *apt-get* запускає перевірку в репозиторіях нових версій програм, доступних для завантаження й оновляє кеш у системі. Опцією *upgrade* запускається відновлення системи до останньої версії. Як правило обидві команди використовуються разом. Можна використовувати обидві команди в одному рядку, для послідовного виконання обох команд, для цього потрібно розділити їх крапкою з комою — *sudo apt-get update; sudo apt-get upgrade* або амперсандом — *sudo apt-get update & sudo apt-get upgrade*. Це досить довга процедура якщо це перше відновлення після установки OS Raspbian.



*Для коректного виконання важливих команд використовується комбінація && — наступна команда буде виконана, тільки якщо попередня завершена без помилок: **sudo apt-get update && sudo apt-get dist-upgrade***

Перед маніпуляціями із сервером необхідно налаштувати статичну IP-адресу обладнання, для того щоб VNC-Клієнт міг звертатися до обладнання по конкретній адресі. Наступними діями, ми скасовуємо режим DHCP (англ. Dynamic Host Configuration Protocol — протокол динамічного налаштування вузла, використовується для автоматичного динамічного присвоєння IP-Адрес обладнанням мережі) і всі параметри мережі встановлюємо вручну.



*Можна використовувати автоматичний режим DHCP, якщо використовується локальна домашня або лабораторна мережа, але при цьому варто врахувати, що в будь-який момент IP-Адреса може бути змінена.*

Для цього спочатку потрібно виконати команду (відкрити файл `/etc/dhcpd.conf` для редагування в текстовому редакторі nano):

```
sudo nano /etc/dhcpd.conf
```

І в самому кінці файлу під рядком `nohook lookup-hostname` указати дані своєї мережі й бажаний Ір-Адрес обладнання (він повинен бути вільний, щоб не було конфліктів, визначити вільний Ір-Адрес можна за допомогою програми Advanced IP Scanner (<http://www.advanced-ip-scanner.com>)) у мережі:

```
interface eth0
static ip_address=192.168.1.6
static routers=192.168.1.1
static domain_name_servers=192.168.1.1 192.168.1.2
```



Зверніть увагу, що налаштування для *Wi-Fi* і *Ethernet* відбувається роздільно. Для *Wi-Fi* у першому рядку коду потрібно вказати ***interface wlan0***.

Після того як усе налаштовано, потрібно перезавантажити систему й переконатися що, налаштовані параметри набули чинності, як презентовано на рисунку 3.13

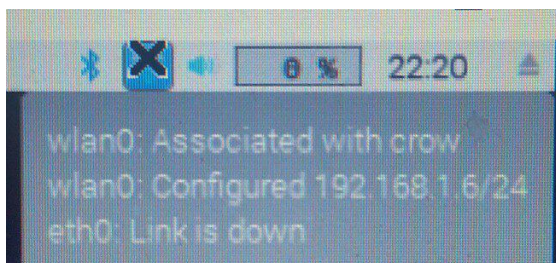


Рисунок 3.13 – Прийняті зміни у файлі *dhcpd.conf* при підключенні *wlan0*.

Установлену Ір-Адресу можна завжди довідатися командою (параметр -І чутливий до регістру):

```
hostname -I
```

Наступним етапом необхідно налаштувати **VNC Server**. Починаючи з версії 4.8 (Raspbian Jessie With Desktop) VNC Server установлений у системі. Якщо його з якихось причин немає, його можна встановити командою:

```
sudo apt-get install realvnc-vnc-server
```

Після цього його потрібно включити в налаштуваннях Start → Preferences → Raspberry Pi Configuration, на вкладці Interfaces, установити галочку Enabled напроти VNC як представлено на рисунку 3.14.

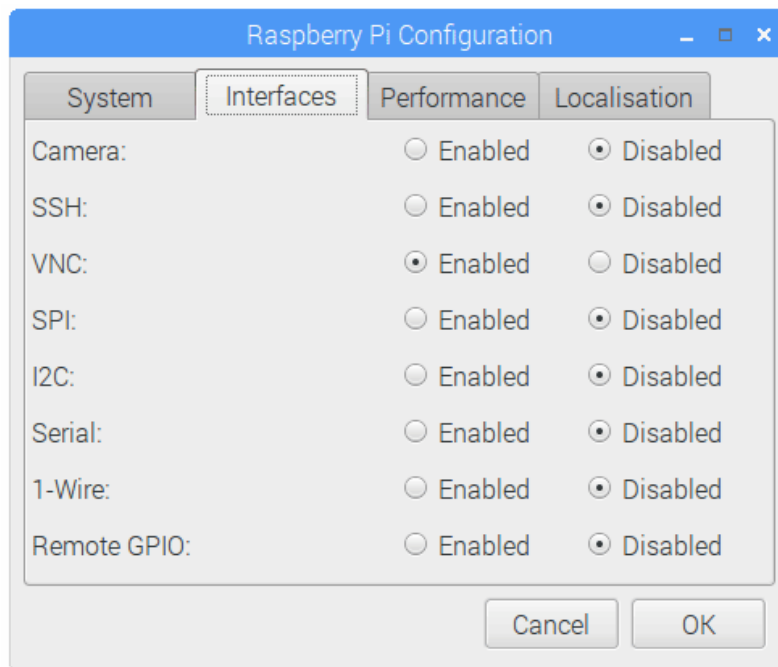


Рисунок 3.14 - Вікно Raspberry Pi Configuration.



*Якщо для налаштування VNC використовується термінал через протокол SSH на базі Putty Вікно Raspberry Pi Configuration можна викликати командою: **sudo raspi-config** ->(меню *Interfacing Options*)*

VNC-Сервер буде запущений автоматично, якщо цього не відбулося, його можна запустити вручну:

```
vncserver
```

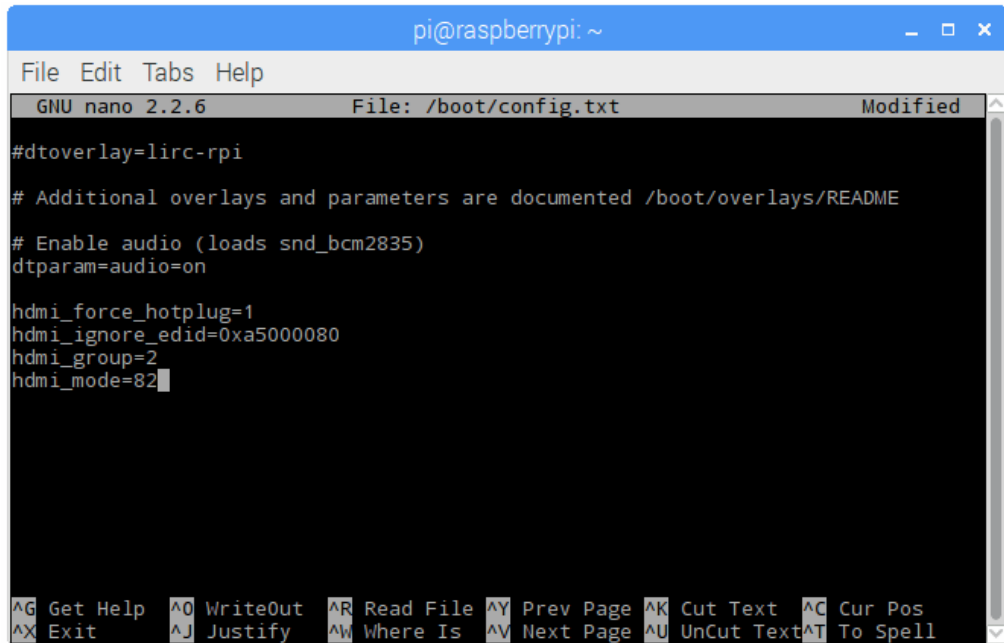
Перед запуском сервера необхідно налаштувати зручний для роботи дозвіл віддаленого робочого стола. Для цього необхідно внести зміни у файл /boot/config.txt за допомогою редактора nano.

```
sudo nano /boot/config.txt
```

Необхідно додати рядок

```
hdmi_force_hotplug=1
hdmi_ignore_edid=0xa5000080
hdmi_group=2
hdmi_mode=82
```

У цьому випадку `hdmi_mode=82` - 1920×1080, 60Hz, 1080p, `hdmi_force_hotplug = 1`. Приклад наведено на рисунку 3.15.



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /boot/config.txt Modified
#dtoverlay=lirc-rpi
# Additional overlays and parameters are documented /boot/overlays/README
# Enable audio (loads snd_bcm2835)
dtparam=audio=on

hdmi_force_hotplug=1
hdmi_ignore_edid=0xa5000080
hdmi_group=2
hdmi_mode=82
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Рисунок 3.15 – Налаштування дозволу віддаленого робочого стола

Після цього зберігаємо внесені зміни `config.txt` за допомогою комбінацій клавиць **Ctrl+X, Y(yes)**.



*Raspberry Pi завжди вважає, що до Hdmi- роз'єму підключений монітор, якщо монітор підключається по Rca- роз'єму, ці рядки необхідно вилучити*

Альтернативно можна задати дозвол віддаленому столу за допомогою команди:

```
tightvncserver :1 -geometry 1360x768 -depth 24
```

як можна бачити дана команда створює робочий стіл під номером 1 з дозволом 1360x768 на 24 години, протягом даного часу не буде приєднань налаштування скидається.



При першому створенні віддаленого робочого стола в терміналі необхідно ввести *login: \*\*\*\** і *password:\*\*\*\*\** (не менше 8 символів).

На наступному етапі перейдемо до налаштування **VNC Viewer** на клієнтській стороні OS Windows ( на прикладі Windows 10), але насправді вона може бути будь-яка, у тому числі й Android. Для початку необхідно скачати VNC Viewer (<https://www.realvnc.com/en/connect/download/viewer/windows/>) і встановити. Першим кроком необхідно створити нове підключення як презентовано на рисунку 3.16.

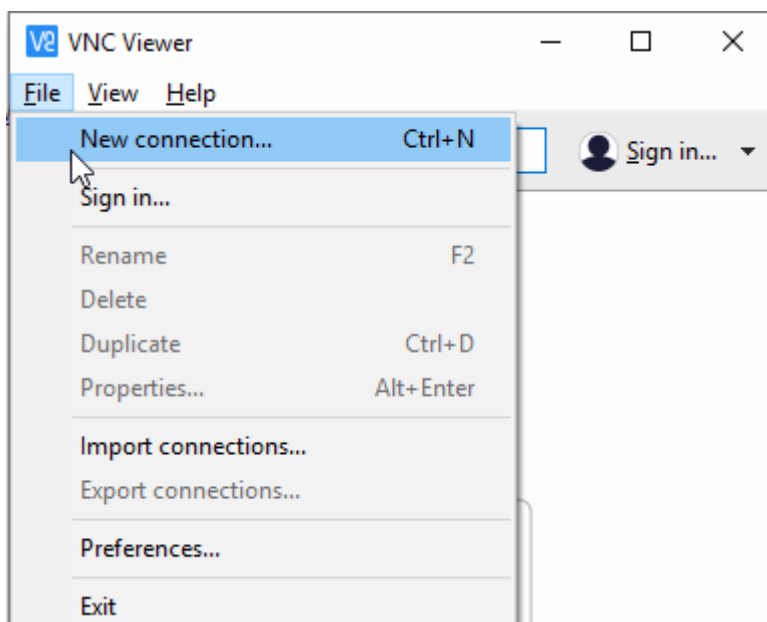


Рисунок 3.16 – Створення нового підключення.

У налаштуваннях з'єднання потрібно вказати тільки IP-Адресу обладнання до якого буде здійснене підключення — вона задається у файлі `dhcpcd.conf` або впізнається динамічно через програму Advanced IP Scanner. Через двокрапку вказуємо номер дисплея, заданий при запуску VNC-Сервера (за замовчуванням 0 — можна не вказувати). У полі `name` можна вказати ім'я з'єднання, наприклад, Raspberry Pi 3. Приклад наведено на рисунку 3.17.

Якщо планується працювати тільки з одним робочим столом, то можна вказувати IP-Адресу (у полі VNC Server) без його номера (192.168.1.6). Але якщо робочих столів буде декілька (кожна нова команда `vncserver` створює новий робочий стіл), підключатися до кожного з них потрібно буде із вказівкою його номера після Ip-Адреси через двокрапку: 192.168.1.6:1, 192.168.1.6:2 і т.д.

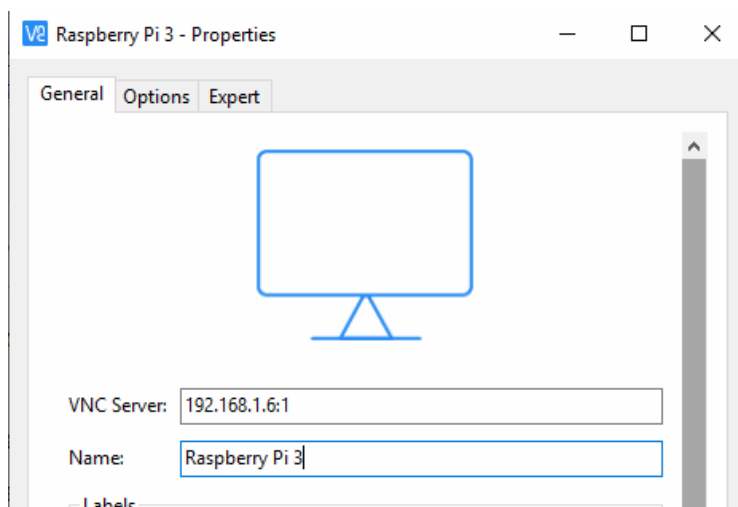


Рисунок 3.17 – Налаштування VNC.

Знову створене з'єднання з'являється в головному вікні VNC Viewer (рис.3.18)

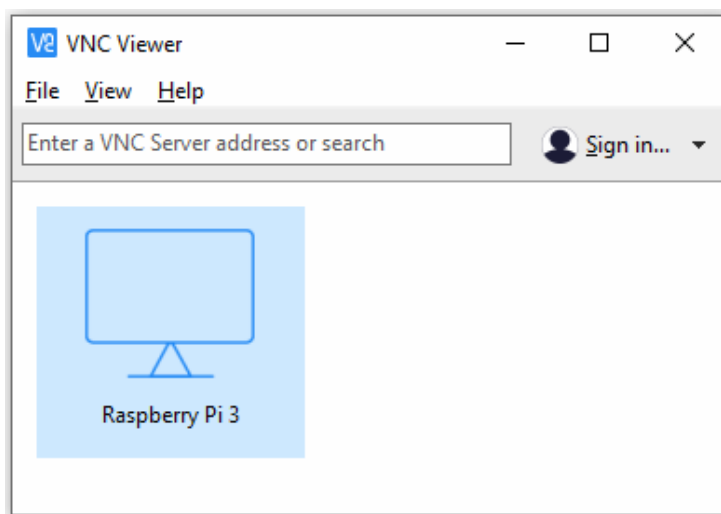


Рисунок 3.18 - Головне вікно VNC Viewer.

Якщо всі налаштування зроблені коректно, то при подвійному кліку повинне з'явитися вікно із запитом імені користувача й пароля як на рисунку 3.19



за замовчуванням логін *pi*, пароль — *raspberry*

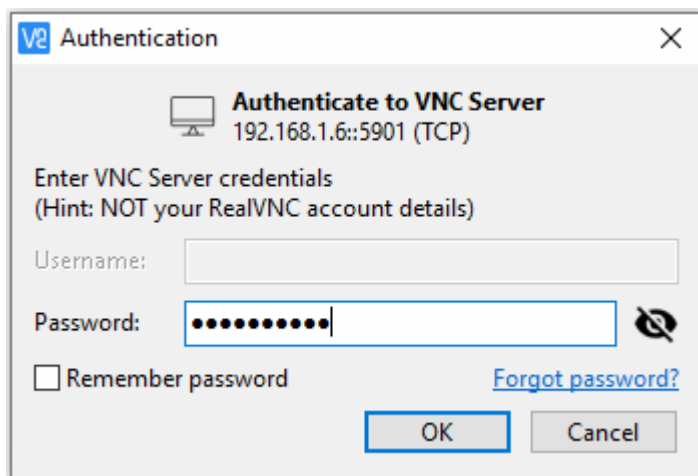


Рисунок 3.19 - Вікно авторизацій.

Можна встановити галочку Remember Password — у цьому випадку вводити пароль більше не буде потрібно.

Після введення пароля, з'являється віддалений робочий стіл системи (рис 3.20) із заданим дозволом.

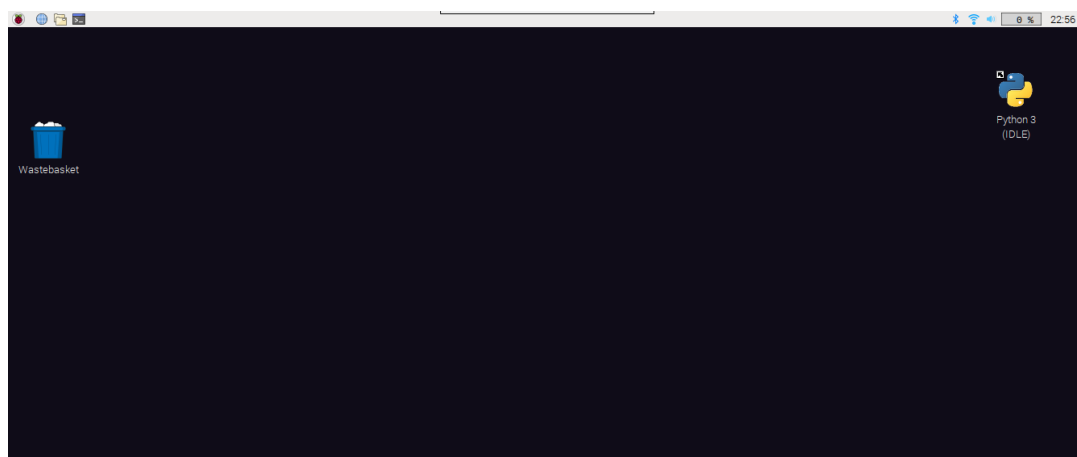


Рисунок 3.20 - Створений віддалений робочий стіл

Віддалений робочий стіл зберігається доти, поки ви його явно не знищите. Запустите наступну команду, якщо ви певен, що він більше не потрібно:

```
vncserver -kill :<display-number>
```



*Зверніть увагу, що ця команда перерве всі поточні підключення без попередження для цих користувачів*

Для зупинки Vnc-Сервера потрібно виконати команду:

```
sudo systemctl stop vncserver-x11-serviced.service
```

Докладніше про команди можна подивитися на офіційній сторінці <https://www.realvnc.com/en/docs/raspberry-pi.html>.

### 3.4.2 Віддалений робочий стол через Internet

Іноді виникає необхідність дистанційного доступу до RPi через межі вашої домашньої мережі. Існує 2 методу:

- через виділений статичний IP-адрес;
- за допомогою програми TeamViewer.

Перший спосіб складний і тут буде опущений. Другий спосіб дуже простий і ефективний. Для його реалізації необхідно скачати TeamViewer Host для Debian (armv7 32bit) зі сторінки наочування[121]. І встановити додаток за допомогою утиліти прямої установки додатків dpkg з файлу .deb, попередньо скачавши настановний файл:

```
wget https://download.teamviewer.com/download/linux/teamviewer-host_armhf.deb
```

```
sudo dpkg --install teamviewer-host_armhf.deb
```

Але оскільки недоліком використання утиліти dpkg є те, що вона працює з окремими пакетами й не обробляє залежності автоматично, установка припиниться. Але при цьому утиліта надасть список залежностей, які можна встановити командою:

*sudo apt-get install -f*

Для налаштування роботи Teamviewer скористаємося вже налаштованим VNC Viewer. На рисунку 3.21 представлений фрагмент віртуального робочого стола зі шляхом виклику Teamviewer 14 Host.

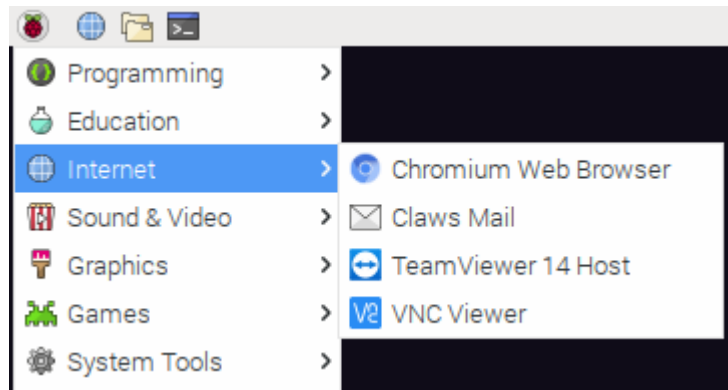


Рисунок 3.21 - Фрагмент віртуального робочого стола зі шляхом виклику Teamviewer 14 Host.

### 3.5 Налаштування samba на OS Raspbian

При розробці проектів пов'язаних з мобільними роботами, часто виникає необхідність взаємодії з файловою системою Raspbian OS через робочий комп'ютер на базі OS Windows. Для того, щоб реалізувати Raspberry Pi 3 у файловий сервер з дистанційним доступом до файлів по локальній мережі, знадобиться встановити файловий сервер Samba. Він дозволяє здійснювати кроссплатформну файлову взаємодію в мережі й працює по протоколу SMB/CIFS.[122]

На першому етапі зробимо відновлення системи за допомогою команд:

*sudo apt-get update*

*sudo apt-get dist-upgrade*

Тепер після відновлення OS Raspbian проводимо установку Samba. Samba - пакет програм, які дозволяють звертатися до мережних дисків і принтерів на різних операційних системах по протоколу SMB/CIFS. Має клієнтську й серверну частини. Є вільним програмним забезпеченням,

випущена під ліцензією GPL. Samba працює на більшості Unix-Подібних систем, таких як Linux, Posix-Сумісних Solaris і Mac OS X Server, на різних варіантах BSD; в OS/2 портовано Samba-Клієнт, що є плагином до віртуальної файлової системи Netdrive. Samba включена практично в усі дистрибутиви Linux. Установку Samba проводимо через термінал SSH або вертуальний робочий стіл за допомогою команди.

```
sudo apt-get install samba samba-common-bin
```

як представлено на рисунку 3.22.

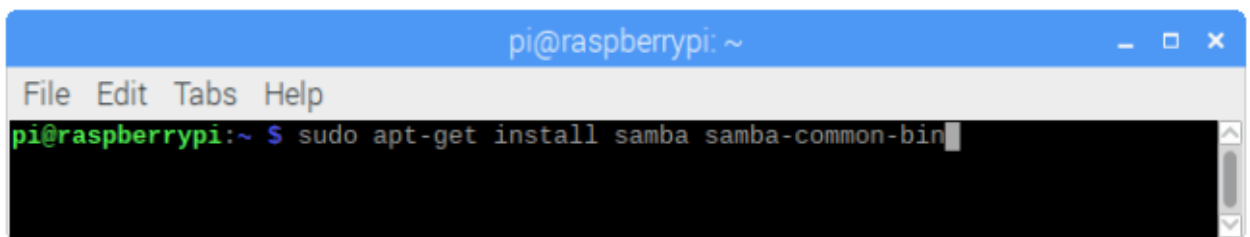


Рисунок 3.22 – Команда для установки Samba у терміналі.

Після установки Samba необхідно створити папку на жорсткому диску Raspberry Pi (Microsd з OS), доступ до якої хотілося б одержати. У нашій випадку це буде папка share у кореневій директорії:

```
sudo mkdir -m 1777 /share
```

Ця команда встановлює Sticky bit (1) (липкий біт — sticky bit використовується в основному для каталогів, щоб захистити в них файли. З такого каталогу користувач може вилучити тільки ті файли, власником яких він є. Прикладом може служити каталог /tmp, у якому запис відкритий для всіх користувачів, але небажане видалення чужих файлів.), щоб запобігти випадковому видаленню каталогу й дає кожному користувачеві права на читання/запис/виконання (777).

Тепер Samba потрібно налаштувати. Для цього відкриваємо файл конфігурації:

`sudo leafpad/etc/samba/smb.conf` – через текстовий графічний редактор по віддаленому робочому столу або на пряму як показано на рисунку 3.23.

*sudo nano/etc/samba/smb.conf*- через термінальний текстовий редактор



*Зверніть увагу, на необхідність створення резервної копії файлу конфігурації Samba за допомогою команди*

І додаємо в кінець файлу наступні 10 рядків налаштування:

*[share]*

*Comment = Pi shared folder*

*Path = /share*

*Browseable = yes*

*Writeable = yes*

*only guest = no*

*create mask = 0777*

*directory mask = 0777*

*Public = yes*

*Guest ok = yes*

У такий спосіб ми дозволяємо всім користувачам, у тому числі й гостям вашої локальної мережі, мати повний доступ до читання, запису й виконання файлів у загальному каталозі /share (рис. 3.23).

Щоб внесені зміни набули чинності необхідно перезавантажити Samba:

*sudo /etc/init.d/samba restart*

```

*smb.conf
File Edit Search Options Help
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[share]
Comment = Pi shared folder
Path = /share
Browseable = yes
Writeable = Yes
only guest = no
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes

```

Рисунок 3.23. – Налаштування Samba сервера до доступу до папки *Share* для всіх користувачів.

### 3.5.1 Налаштування доступу до існуючих папок OS Raspbian

У більшості випадків при роботі із проектами в команді може перебувати велика кількість людей для яких необхідно відкрити доступ уже до існуючих папок, як на прикладі `/home/pi`. Для цього необхідно внести зміни у файл конфігурації `smb.conf` для цього відкриває його за допомогою:

`sudo leafpad/etc/samba/smb.conf` – через текстовий графічний редактор по віддаленому робочому столу або на пряму як показано на рисунку 3.23;

`sudo nano/etc/samba/smb.conf` – через термінальний текстовий редактор та додаємо в кінець файлу наступні налаштування, при чому параметру `Path` привласнюємо шлях до папки, яку прагнемо зробити доступною - `/home/pi`:

```

[share]
Comment = Pi shared folder
Path = /home/pi
Browseable = yes
Writeable = yes
only guest = no

```

```
create mask = 0777
directory mask = 0777
Public = yes
Guest ok = yes
```

Після перезавантаження Samba за допомогою команди:

```
sudo /etc/init.d/samba restart
```

Папка стане доступною всім користувачам мережі, але тільки для читання. Це пов'язане з тим, що всі папки й файли були створені раніше й для того, щоб відкрити повний доступ до них, потрібно перепризначити права на всі елементи розширеної папки:

```
sudo chmod -R 777 /home/pi
```



Тут необхідно мати на увазі, що файли, створені безпосередньо в Raspbian OS (наприклад, скетчі Python) після відкриття доступу, не будуть мати дозволу на зміни з боку зовнішніх комп'ютерів у мережі. Для того, щоб призначити права на зміну цих файлів, потрібно знову повторити процедуру зміни прав: `sudo chmod -R 777 /home/pi`.

Якщо необхідно надати доступ до кореневої папки ( до всіх папок), отже у файлі настроювань необхідно змінити рядок `Path = /any_path` на `Path = /`. При цьому прийдеться повторити процедуру зміни прав на папку (атрибут `-R` указує, що команду слід застосувати й до всіх вкладених елементів).

Для обмеження доступу до відкритих ресурсів Raspberry Pi3, необхідно по-перше, заборонити доступ гостям — для цього з файлу конфігурації потрібно вилучити рядок `Guest ok = yes`. По-друге, потрібно задати пароль Samba для доступу до папки `pi` за допомогою команди:

```
sudo smbpasswd -a pi
```

Для застосування змін необхідно перезавантажити сервер Samba за допомогою команди:

```
sudo /etc/init.d/samba restart
```

*У випадку зміни прав на всі папки:*

***sudo chmod -R 777 /***

*автоматично будуть переписані права й вашого root-користувача (папка `usr/bin/sudo`). Після цієї процедури, виконання будь-якої команди під суперюзером `sudo` буде неможливо! Будь-яка спроба використання команди `sudo` буде завершена помилкою:*



***sudo: /usr/bin/sudo повинен належати користувачеві з uid 0 і мати біт setuid***

*Тому, наполегливо рекомендується точно вибирати папки для зміни прав:*

***sudo chmod -R 777 /one\_folder***

### 3.6 Створення резервної копії OS Raspbian

Створення резервної копії (“бэкапа”) з Raspberry Pi представляє із себе образ, знятий з microSD-картки із встановленою й сконфігурованою OS Raspbian і набором додатків. Отже користувач створює власну версію дистрибутива Raspbian або робить унікальне складання з оригінального OS Raspbian із встановленим у цьому випадку набору сторонніх додатків необхідних для розв'язків певних завдань поставлених перед користувачем. Для створення резервної копії необхідні наступні елементи:

- Raspberry Pi 3 Model B;
- картридер або здатне його замінити обладнання;
- комп'ютер під керуванням Windows, Linux або macOS.

Розглянемо послідовність дій для створення резервної копії під OS Windows буде потрібна безкоштовна програма Win32 Disk Imager, скачати яку можна сайті Sourceforge (<https://sourceforge.net/projects/win32diskimager/>).[123] Загальний вид Win32 Disk Imager представлено на рисунку 3.24

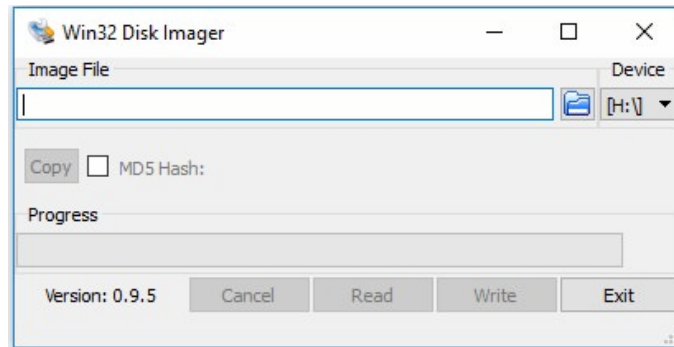


Рисунок 3.24 – Загальний вид програми Win32 Disk Imager.

При встановленій у картридер (або его обладнання, що заміняє) microSD-картці з Raspberry Pi у *поле Image File* уводимо назву файлу, у який буде записаний образ, і натискаємо кнопку “Read”: Програма повністю зчитає вміст microSD в Img-Файл. Тому не вигідно використовувати в якості системного накопичувача Raspberry Pi карти пам'яті більших розмірів – вони всі будуть зчитуватися цілком, а тримати на комп'ютері диску “бекапи” вагою по 32-64Гб кожний, досить не зручно. Оптимальний розмір microSD під Raspberry Pi для розв'язку завдань розглянуті в даній книзі – 8Гб. Максимум 16Гб. Більше – уже немає особливої необхідності, для зберігання файлів краще використовувати USB-Накопичувачі.

Відновлення Raspberry Pi з бекапа відбувається через ту ж програму Win32 Disk Imager. Потрібно вказати шлях до збереженого файлу образу й натиснути кнопку “Write”.



*Після відновлення системи з бекапа злітають створені з'єднання з Bluetooth-Обладнаннями – потрібно заново проводити сполучення, але всі інші налаштування зберігаються.*

Розглянемо послідовність дій для створення резервної копії під OS Linux. В Linux-Системах зробити “бекап” Raspberry Pi можна за допомогою системної утиліти *dd*.

Для цього потрібно ввести в терміналі наступну команду:

```
sudo dd if=/dev/mccblk0 of=/home/Username/Desktop/raspberrypi-
backup.img
```

Де в “*if=*” прописується шлях до вставленої в картридер microsd-карти, а в “*of=*” прописується шлях до файлу образу, у який уміст цієї карти буде вписуватися.

Відповідно, замість ***/dev/mccblk0*** і ***/home/Username/Desktop/raspberrypi-backup.img*** потрібно ввести свої шляхи.

Шлях до змонтованої microsd-карті можна подивитися виконанням консольної команди:

```
ls -l /dev/disk/by-uuid
```

Відновлення Raspberry Pi з бекапа проводиться за допомогою тієї ж утиліти *dd*, тільки міняються місцями значення “*of*” і “*if*”:

```
sudo dd if=/home/Username/Desktop/raspberrypi-backup.img
of=/dev/mccblk0.
```

Розглянемо послідовність дій для створення резервної копії *macos*. Оскільки *macos* теж є \* nix-системою, процес бекапа Raspberry Pi у цій операційній системі не буде кардинально відрізнятися від вищеописаної процедури “бекапа” під Linux.

Тільки шлях до змонтованої microsd-карті виглядає через програму Disk Utility (Дискова утиліта) яку представлено на рисунку 3.25.[124]

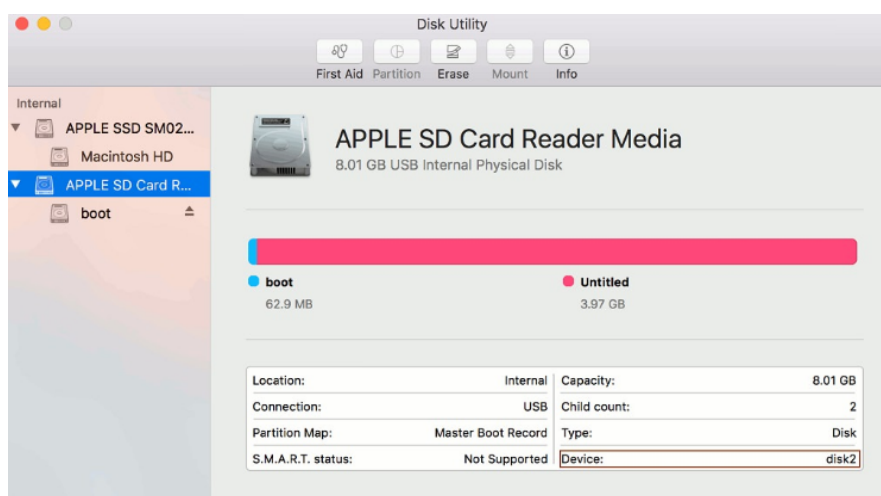


Рисунок 3.25 – Загальний вид програми Disk Utility для macOS

А потім виконується команда `dd` для запису образу карти пам'яті у файл:

```
dd if=/dev/disk2 of=raspberrypi-backup.img
```

Для відновлення карти пам'яті зі знятого образу OS Raspbian необхідно ввести наступну команду:

```
dd if=raspberrypi-backup.img of=/dev/disk2.
```



Перший “бекан” рекомендується зробити відразу після завершення установки й налаштування OS Raspbian, і надалі робити бекапи не рідше, чим раз в 2-4 тижні.

### 3.7 Контрольні питання

1. Які OS піддерживаються Raspberry Pi?;
2. Які мови програмування піддерживаються за замовчуванням OS Raspbian ?;
3. Для чого необхідний *Terminal* в Raspbian OS ?;
4. Перерахуйте найбільш часто використовувані команди користувачем в терміналі?;
5. Особливості текстового редактора *nano*;
6. Яка команда дозволяє встановити вбрання ПЗ ?;
7. Назвіть команду для видалення обраного ПЗ?;
8. За допомогою якої команди можна поміняти базові настройки Raspberry Pi ?;
9. Перерахуйте основні інструмент *raspi-config*;
10. За допомогою яких команд проводиться поновлення OS Raspbian через Terminal;
11. Що таке DHCP ?;
12. Для чого використовується VNC ?;
13. З допомогою якої команди можна задати дозволу віддаленого столу ?;

14. Якими методами можна створити віддалений робочий стіл через Internet ?;
15. Для чого потрібен сервер Samba на OS Raspbian?;
16. Перерахуйте права які може використовувати адміністратор для обмеження доступу до певних ресурсів?;
17. Для чого не обходимо робити резервну копію OS?;

## 4. УСТАНОВКА ТА НАЛАШТУВАННЯ PYTHON ДЛЯ РОЗРОБКИ СИСТЕМ КЕРУВАННЯ МОБІЛЬНИМИ РОБОТАМИ

### 4.1 Установка бібліотеки OpenCV

У цьому розділі книги розглянемо особливості установки бібліотеки комп'ютерного зору OpenCV і використання її на Raspberry Pi 3 за допомогою мови Python.[125] Це бібліотека з відкритим вихідним кодом, дозволяє реалізовувати різні алгоритми обробки відео і зображень, за допомогою стандартних типів даних, функцій і класів.



*Варто зауважити що принцип установки бібліотеки OpenCV для мови Python (OS Raspbery) і принцип установки бібліотеки OpenCV для мови Python (OS Windows famely) неоднакові.*

Для зручності опис процесу установки і компіляцій ядра, даний матеріал буде представлений вигляді кроків.

#### ***Крок 1. Підготовка перед установкою.***

Бібліотека OpenCV для своєї установки вимагає значний обсяг вільного дискового простору - приблизно 6,5-7 Гб (з них близько 6 Гб можна буде видалити після установки). Тому, для використання OpenCV рекомендується використовувати SD-карту ємністю не менше 16 Гб.[126]

Перевірити наявність вільного місця на SD-карті (рисунок 4.1), можна, скориставшись утилітою *df* з параметром -h:

```
df -h
```

Якщо з якихось причин, на момент початку установки бібліотеки OpenCV, файлова система не розширена на всю SD-карту, це необхідно зробити як описано в главі вище.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        15G   5.7G   7.9G   42% /
devtmpfs        434M   0    434M   0% /dev
tmpfs           438M   0    438M   0% /dev/shm
tmpfs           438M  12M   427M   3% /run
tmpfs           5.0M   4.0K   5.0M   1% /run/lock
tmpfs           438M   0    438M   0% /sys/fs/cgroup
/dev/mmcblk0p1  42M   21M   21M   51% /boot
tmpfs           88M   0    88M   0% /run/user/1000
pi@raspberrypi:~ $

```

Рисунок 4.1 – Результати аналізу вільного місця

Для того, щоб звільнити додаткове місце на SD-карті, можна видалити непотрібні пакети, наприклад Wolfram Mathematica (близько 700 Мб) або LibreOffice (близько 200 Мб) наступними командами:

```
sudo apt-get purge wolfram-engine
```

```
sudo apt-get purge libreoffice*
```

Те ж саме можна зробити за допомогою вікна Add/Remove Software як показано на рисунок 4.2 ситуації для нумерації.

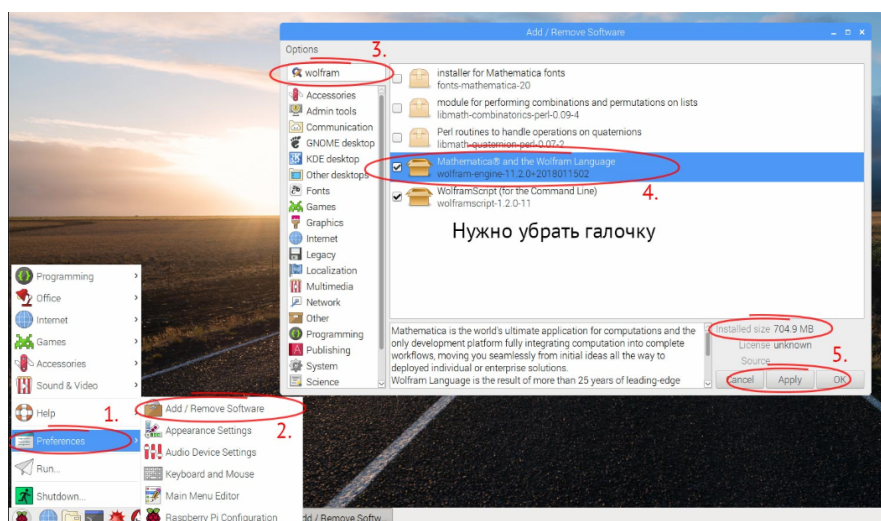


Рисунок 4.2 – Видалення програм через вікно Add/Remove Software

За допомогою команди *clean* звільняється дисковий простір за рахунок видалення отриманих *.deb*-пакунку пакетів в локальній репозиторії (очищення локального кеша):

```
sudo apt-get clean
```

Командою *autoremove* автоматично будуть видалені встановлені як залежності інших, але більше не потрібні, «осиротілі» пакети. Це дозволить вивільнити ще близько 700-800 Мб:

```
sudo apt-get autoremove
```



*Перед усіма маніпуляціями, як зазвичай, необхідно оновити систему командами `sudo apt-get update` і `sudo apt-get dist-upgrade` а також **презагрузити операційну систему.***

### **Крок 2. Встановлення залежних пакетів.**

Перед початком установки, потрібно встановити необхідні для роботи бібліотеки OpenCV пакети, включаючи додаткові пакети для роботи з зображеннями, відео і т.д. Назва додаткових пакетів і їх опис представлено в таблиці 4.1.

Таблиця 4.1 – Опис додаткових пакетів для роботи з бібліотеки OpenCV

Назва пакета	Опис пакета
1	2
cmake	Утиліта для автоматичного складання програми з вихідного коду
cmake-curses-gui	Пакет GUI (графічний інтерфейс) для cmake
libgtk2.0-dev	GTK+ інструментарій для створення графічних користувацьких інтерфейсів
libjpeg-dev	Бібліотеки для роботи з різними форматами зображень

Продолжение таблицы 4.1

1	2
libjasper-dev	JasPer - набір програм для кодування та обробки зображень
libavcodec-dev	Бібліотека кодеків от Libav (аудіо / відео)
libswscale-dev	Бібліотека для виконання високооптимізовані збільшувати або зменшувати зображення
libv4l-dev	Набір бібліотек, для роботи з пристроями video4linux2
libx264-dev	Библиотека кодирования для создания видеопотоков H.264 (MPEG-4 AVC)
libxvidcore-dev	Відеокодек MPEG-4 (Xvid)
gfortran	Матричні перетворення
libatlas-base-dev	Підпрограми лінійної алгебри

Для роботи з бібліотекою OpenCV мінімально необхідно встановити наступний набір пакетів за допомогою команд:

```
sudo apt-get install cmake cmake-curses-gui libgtk2.0-dev
```

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libx264-dev libxvidcore-dev
```

```
sudo apt-get install libjpeg-dev libpng12-dev libtiff5-dev libjasper-dev
```

```
sudo apt-get install gfortran libatlas-base-dev
```

Не буде зайвим встановити додатковий пакети для роботи з відео, закодованими різними кодеками:

```
sudo apt-get install -y libdc1394-22-dev libavresample-dev libgphoto2-dev ffmpeg libgtk-3-dev
```

Якщо планується 3D-візуалізація або 3D-моделювання (в т.ч. складних процесів), знадобиться пакет VTK (Visualization ToolKit, - (інструментарій візуалізації) - відкрита кроссплатформенная програма для тривимірного моделювання, обробки зображень і прикладної візуалізації.)

```
sudo apt-get install -y libvtk6-dev
```

Далі потрібно встановити python-пакет NumPy. NumPy - це розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами:

```
sudo pip3 install numpy
```



Установка **numpy** займає тривалий час без ознак життя - це не зависання.

### ***Шаг 3. Скачивание и распаковка исходников***

Створимо в папці `/home/pi` робочу папку `opencv` і перейдемо в неї за допомогою наступних команд:

```
cd /home/pi
```

```
mkdir opencv
```

```
cd opencv
```

Тепер необхідно завантажити архіви з вихідними кодами бібліотеки і додаткових модулів (рекомендується використовувати однакові релізний версії бібліотеки і додаткових модулів) і розпакувати їх командою **wget**:



На момент написання книги релізний версія бібліотеки 3.4.1, додаткових модулів 3.4.1:

```
https://github.com/opencv/opencv/releases
```

```
https://github.com/opencv/opencv_contrib/releases
```



Використання нерелізних версій може викликати помилки, в тому числі при створенні конфігурації збірки, за допомогою утиліти **cmake**

```
wget https://github.com/opencv/opencv/archive/3.4.0.zip -O  
opencv_source.zip  
unzip opencv_source.zip
```

Оскільки є можливість використання додаткових модулів, те ж саме доведеться зробити і для них:

```
wget https://github.com/opencv/opencv_contrib/archive/3.4.0.zip -O
opencv_contrib.zip
```

```
unzip opencv_contrib.zip
```

Після того, як файли будуть розархівовані, їх можна видалити:

```
rm opencv_source.zip
```

```
rm opencv_contrib.zip
```

Створюємо папку для конфігурації:

```
cd /home/pi/opencv/opencv-3.4.0
```

```
mkdir build
```

```
cd build
```

#### **Крок 4. Налаштування конфігурації**

На даному етапі необхідно налаштувати параметри збірки, за допомогою утиліти **cmake**, яка була встановлена на початку. Для цього, всі значущі параметри передаються в якості змінних утиліти, разом з присвоюють значеннями:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE \
cmake -D CMAKE_INSTALL_PREFIX=/usr/local \
cmake -D INSTALL_C_EXAMPLES=OFF \
cmake -D INSTALL_PYTHON_EXAMPLES=ON \
cmake -D OPENCV_EXTRA_MODULES_PATH=/home/pi/opencv
/opencv_contrib-3.4.0/modules \
cmake -D BUILD_EXAMPLES=ON \
cmake -D BUILD_DOCS=ON \
cmake -D ENABLE_NEON=ON ..
```

Якщо створення конфігурації завершилося повідомленнями *Configuring done* і *Generating done* (рисунок 4.3), то настройка параметрів пройшла успішна.

```

pi@raspberrypi:~/opencv-
File Edit Tabs Help
--
-- Install to:
-----
-- Configuring done
-- Generating done
-- Build files have been written to:
pi@raspberrypi:~/opencv-

```

Рисунок 4.3 – Повідомлення про успішне завершення створення конфігурацій параметрів

Тут же, в звіті, перед початком збирання бінарників, потрібно переконатися, що додаткові модулі (наприклад *face*) також будуть включені в збірку:

*--OpenCV modules:*

*--To be built: aruco bgsegm bioinspired calib3d ccalib core datasets dnn dpm face features2d flann freetype fuzzy highgui img\_hash imgcodecs imgproc line\_descriptor ml objdetect optflow phase\_unwrapping photo plot python2 python3 python\_bindings\_generator reg rgbd saliency shape stereo stitching structured\_light superres surface\_matching text tracking ts video videoio videostab xfeatures2d ximgproc xobjdetect xphoto*

*--Disabled: js world*

*--Disabled by dependency: -*

*--Unavailable: cnn\_3dobj cudaarithm cudabgsegm cudacodec cudafeatures2d cudafilters cudaimgproc cudalegacy cudaobjdetect cudaoptflow cudastereo cudawarping cudev cvv dnn\_modern hdf java matlab ovis sfm viz*

*--Applications: tests perf\_tests examples apps*

*--Documentation: NO*

*--Non-free algorithms: NO*

Після завершення кофігурірованія, настройки конфігурації можна змінити за допомогою утиліти, що надає графічний інтерфейс (рисунок 4.4) - *csmake*:

*csmake* ..

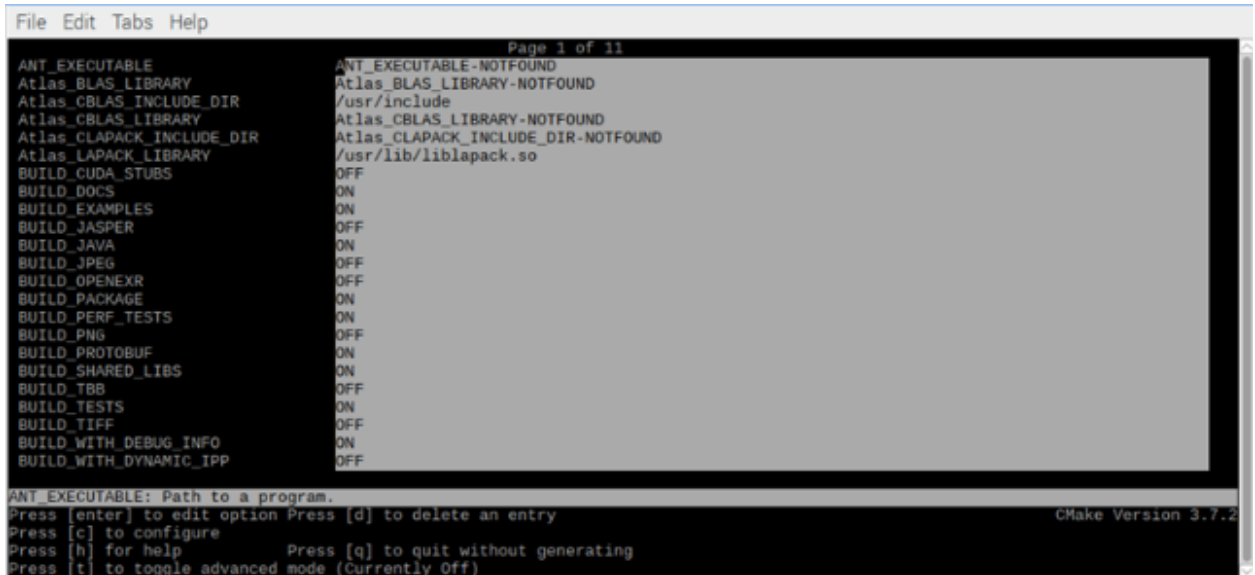


Рисунок 4.4 - Графічний інтерфейс *csmake*

### ***Крок 5. Компіляція бібліотеки OpenCV***

Після настройки конфігурації, можна приступати до складання. Параметр *-jn*, задає кількість задіяних ядер процесора (*n*) - для Raspberry Pi 3 можна задіяти 4 ядра:

*make -j4*



*В процесі компіляції процесор дуже сильно гріється. При відсутності радіатора процесор буде знижувати тактову частоту, що дуже ймовірно призведе до зависання системи.*

*З метою усунення можливості подібних неприємностей, рекомендується використовувати меншу кількість ядер, наприклад 1 або 2 (-j1 або -j2).*

У разі зависання, рекомендується, після перезавантаження, відновити процедуру компіляції з залученням одного ядра:

*cd /home/pi/opencv/opencv-3.4.0/build*

```
make -j1
```

### **Крок 6. Установка бібліотеки OpenCV**

Після успішної компіляції, бібліотеку OpenCV необхідно встановити. Робиться це за допомогою тієї ж утиліти *make*:

```
sudo make install
```

Після того, як бібліотека OpenCV інстальована, утилітою *ldconfig*, створюємо необхідні зв'язки і формуємо кеш встановлених динамічних бібліотек:

```
sudo ldconfig
```

Для перевірки, в оболонці Shell (Python 3) можна внести пару рядків:

```
>>> import cv2  
>>> cv2.__version__
```

Як результат виконання команд Python видасть версію бібліотеки OpenCV.

### **Шаг 7. Видалення початкових кодів**

Після того, як бібліотека встановлена, всі вихідні файли можна видалити (папка *opencv* - приблизно 6 Гб) наступною командами:

```
cd ~  
rm -rf opencv
```

## **4.2 Установка бібліотек для роботи з GPIO**

Бібліотека RPi.GPIO написана спеціально для управління портами GPIO на мові Python. Ці порти дозволяють збирати даних з датчиків, управляти двигунами, ключами, реле та іншими пристроями за допомогою програми. Опис призначення всіх 40 pin порту GPIO описано в розділі 2.3.



При роботі з портом GPIO не обходимо бути дуже уважним, великий відсоток «згорілих» *Raspberry Pi* були з вини користувачів що не правильно зробили підключення.

На першому кроці не обходимо завантажити архів з офіційного Web-сайту [sourceforge.net](https://sourceforge.net) (<https://sourceforge.net/projects/raspberry-gpio-python/files/? Source = navbar>).[127] На сторінці можна побачити велику кількість версій бібліотеки, не обходимо вибрати саму последню версію. На малюнку 4.5 представлені скріншот сайту з версією бібліотеки 0.7.0 від 22.07.2019.

Name	Modified	Size	Downloads / Week
raspbian-buster	2019-07-22		18
raspbian-stretch	2018-11-16		1
old releases	2018-11-16		29
raspbian-jessie	2016-10-30		0
raspbian-wheezy	2016-02-29		0
RPi.GPIO-0.7.0.tar.gz	2019-07-22	30.8 kB	64
Totals: 6 Items		30.8 kB	112

Рисунок 4.5 – Скріншот Web - сайту [sourceforge.net](https://sourceforge.net) з архівом бібліотекою RPi.GPIO-0.7.0.tar.gz

Архив будет загружен в папку загрузки по пути `/home/pi/Downloads`.

На другому кроці не обходимо перейти в папку Downloads де знаходиться скачаний архів, натиснути праву кнопку миші і вибрати у спливаючому меню Extract Here, як показано на малюнку 4.6 і розархівувать його в дану папку.

Ця команда розархівуйте бібліотеку в папку, в яку поміщений архів. Поруч з ним буде додана нова папка з назвою RPi.GPIO-0.7.0.

Третім кроком потрібно встановити *python-dev* за допомогою терміналу і команди

*sudo apt-get install python-dev*

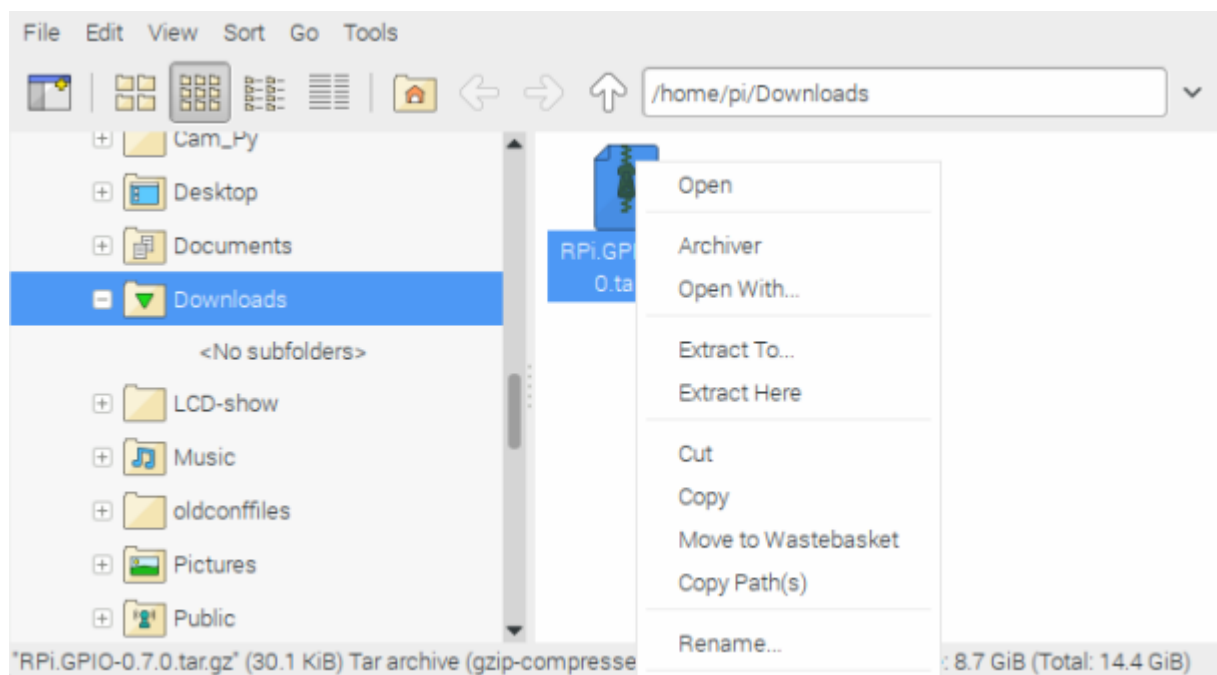


Рисунок 4.6 – Розархівація архіву бібліотеки RPi.GPIO-0.7.0.tar.gz

В ході установки буде поставлено питання «Do you want to continue? [Y/n]» (рисунок 4.7), потрібно натиснути «Yes »і Enter.

```
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get install python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libpython-dev libpython2.7-dev python2.7-dev
The following NEW packages will be installed:
  libpython-dev libpython2.7-dev python-dev python2.7-dev
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.2 MB of archives.
After this operation, 25.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Рисунок 4.7 – Установка *python-dev*

Підтвердженням коректної установки у вікні терміналу буде видано повідомлення про успішну установки як показано на рисунку 4.8.

```

File Edit Tabs Help
Get:3 http://mirrordirector.raspbian.org/raspbian/ jessie/main libpython-dev arm
hf 2.7.9-1 [19.6 kB]
Get:4 http://mirrordirector.raspbian.org/raspbian/ jessie/main python2.7-dev arm
hf 2.7.9-2+deb8u1 [287 kB]
Fetched 18.2 MB in 49s (367 kB/s)
Selecting previously unselected package libpython2.7-dev:armhf.
(Reading database ... 121836 files and directories currently installed.)
Preparing to unpack ../libpython2.7-dev_2.7.9-2+deb8u1_armhf.deb ...
Unpacking libpython2.7-dev:armhf (2.7.9-2+deb8u1) ...
Selecting previously unselected package libpython-dev:armhf.
Preparing to unpack ../libpython-dev_2.7.9-1_armhf.deb ...
Unpacking libpython-dev:armhf (2.7.9-1) ...
Selecting previously unselected package python2.7-dev.
Preparing to unpack ../python2.7-dev_2.7.9-2+deb8u1_armhf.deb ...
Unpacking python2.7-dev (2.7.9-2+deb8u1) ...
Selecting previously unselected package python-dev.
Preparing to unpack ../python-dev_2.7.9-1_armhf.deb ...
Unpacking python-dev (2.7.9-1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libpython2.7-dev:armhf (2.7.9-2+deb8u1) ...
Setting up libpython-dev:armhf (2.7.9-1) ...
Setting up python2.7-dev (2.7.9-2+deb8u1) ...
Setting up python-dev (2.7.9-1) ...
pi@raspberrypi:~ $

```

Рисунок 4.8 - Підтвердження коректної установки *python-dev*

Для установки бібліотеки RPi.GPIO, не обходимо в даному терміналі перейти в папку (за замовчуванням /home/pi/Downloads/RPi.GPIO-0.7.0) в якій був разархіфровано бібліотека RPi.GPIO-0.7.0.tar.gz з допомогою простої консольної команди *cd*

```
cd /home/pi/Downloads/RPi.GPIO-0.7.0
```

і запустити установку бібліотеки за допомогою команди

```
sudo python setup.py install
```

Після успішної установки на екрані з'явиться новий звіт як представлено на рисунку 4.9.

```

pi@raspberrypi: ~/Downloads/RPi.GPIO-0.7.0
relro -Wdate-time -D_FORTIFY_SOURCE=2 -g -fdebug-prefix-map=/build/python2.7-Ub3
vap/python2.7-2.7.13=. -fstack-protector-strong -Wformat -Werror=format-security
  build/temp.linux-armv7l-2.7/source/py_gpio.o build/temp.linux-armv7l-2.7/source
/c_gpio.o build/temp.linux-armv7l-2.7/source/cpuinfo.o build/temp.linux-armv7l-2
.7/source/event_gpio.o build/temp.linux-armv7l-2.7/source/soft_pwm.o build/temp.
linux-armv7l-2.7/source/py_pwm.o build/temp.linux-armv7l-2.7/source/common.o bui
ld/temp.linux-armv7l-2.7/source/constants.o -o build/lib.linux-armv7l-2.7/RPi/G
PIO.so
running install_lib
creating /usr/local/lib/python2.7/dist-packages/RPi
copying build/lib.linux-armv7l-2.7/RPi/_GPIO.so -> /usr/local/lib/python2.7/dist
-packages/RPi
copying build/lib.linux-armv7l-2.7/RPi/__init__.py -> /usr/local/lib/python2.7/d
ist-packages/RPi
creating /usr/local/lib/python2.7/dist-packages/RPi/GPIO
copying build/lib.linux-armv7l-2.7/RPi/GPIO/__init__.py -> /usr/local/lib/python
2.7/dist-packages/RPi/GPIO
byte-compiling /usr/local/lib/python2.7/dist-packages/RPi/__init__.py to __init
_.pyc
byte-compiling /usr/local/lib/python2.7/dist-packages/RPi/GPIO/__init__.py to __
init__.pyc
running install_egg_info
Writing /usr/local/lib/python2.7/dist-packages/RPi.GPIO-0.7.0.egg-info
pi@raspberrypi:~/Downloads/RPi.GPIO-0.7.0 $ █

```

Рисунок 4.9 - Приклад звіту про успішну установки бібліотеки RPi.GPIO-0.7.0

Крім вищеописаної бібліотеки RPi.GPIO існують і інші. Для доступу до GPIO контактам використовується також бібліотека WiringPi. Ця бібліотека написана на мові C, використовується в середовищах C, C ++ і RTB (BASIC) та інших. Вона створювалася схожою з бібліотекою Wiring на Arduino.

### 4.3. Контрольні питання

1. Що таке комп'ютерний зір і для чого воно використовується ?;
2. Опишіть призначення бібліотеки OpenCV?;
3. Перерахуйте необхідні пакет для коректної роботи бібліотеки OpenCV;
4. Для чого використовують графічний інтерфейс *cmake*?;
5. Як перевірити правильність установки бібліотеки OpenCV?.

## 5. ПРОЕКТУВАННЯ МОБІЛЬНОГО РОБОТА НА БАЗІ RASPBERRY PI

### 5.1 Збірка мобільної платформи 4WD від фірми Elecrow

Оригінальна і зручна робо-платформа на 4 колесах від фірми Elecrow (<https://www.elecrow.com>) може послужити як основа для побудови робота на управлінні за допомогою Bluetooth або WiFi, за допомогою якого ви зможете реалізувати будь-які експериментальні дослідження. На рисунку 5.1 представлений стандартний комплект робо-платформи за ціну 350 грн.[128]

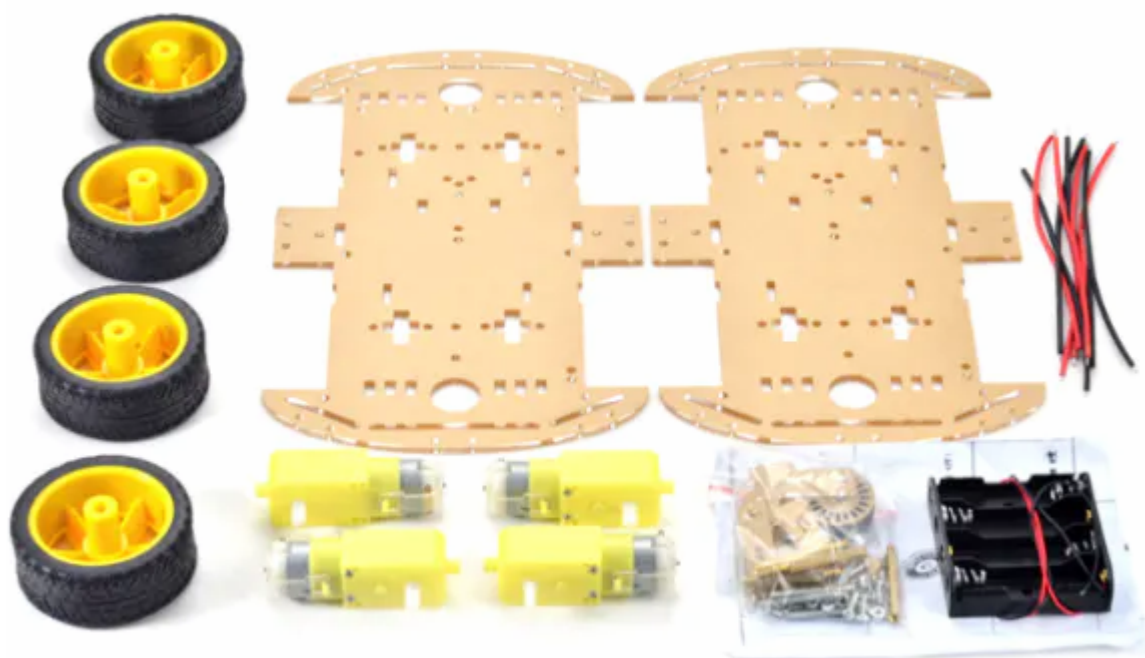


Рисунок 5.1 – Комплект мобільної платформи 4WD

До складу комплекту мобільно платформи 4WD входять: акрилова платформа (2 пластини) - 1 шт; колеса-4 шт; диск з прорізами для оптич.датчика- 2 шт .; мотор з редуктором- 4 шт (рис. 5.2); кронштейн для кріплення мотора- 8 шт; вимикач харчування - 1 шт .; набір латунних стійок з гвинтами: -1 шт .; набір гвинтів і гайок: -1 шт .; батарейний відсік: 1 шт.

Особливості платформи:

- розміри платформи 235 x 155 мм;
- проста конструкція зручна в збірці;
- оснащена енкодер на два колеса;
- можливість розширення і дообладнання платформи.



Рисунок 5.2 - DC мотор з редуктором

Характеристики моторів:

- напруга живлення: 6В;
- сила струму: 120 мА;
- редуктор: 48: 1;
- швидкість обертання: 240 об / хв;
- діаметр колеса: 66 мм;
- швидкість: 48 м / хв;
- вага мотора: 50 г;
- розміри мотора: 70 x 22 x 18 мм.

Після складання ви отримаєте таку модель як представлена на рисунку

5.3



*Максимальний крутний момент: 800 г / см (при 3 В),  
 передавальне число 1:48 (при 3 В), струм навантаження: 70мА  
 @ 3V; напруга живлення: 3-8 Вольт; швидкість обертання без  
 навантаження: 170 об / хв (при напрузі 3,6 В).*



Рисунок 5.3 – Зібрана модель мобільної платформи 4WD від фірми Elecrow

Наступним кроком необхідно вибрати драйвер (модуль) управління 4м DC моторами

## **5.2 Управління кроковими двигунами та DC моторами на базі L298**

При проектуванні верстатів, роботів та інших автоматизованих пристроїв виникає необхідність управляти електродвигуном постійного струму або ж котушками крокової движка. Для того, щоб мати можливість керувати обмоткою двигуна і змусити його вал обертатися в різні боки, необхідно виконувати комутацію з переполюсовкой. Для такої мети використовується так званий "H-міст".

Основним елементом схеми є електронний ключ. Принципова схема моста нагадує латинську букву «H», звідси назва пристрою. У схему входять 4 ключа розташованих попарно, зліва і справа, а між ними включена навантаження (рис. 5.4).

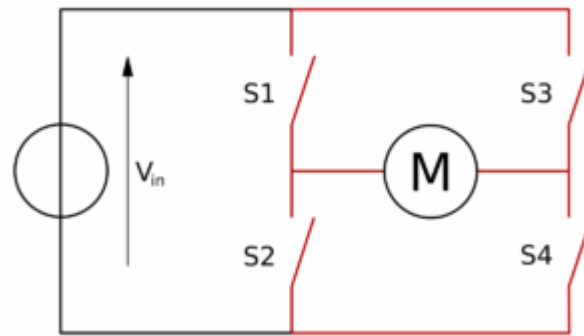


Рисунок 5.4 - Принципова схема Н-моста.

На схемі видно, що перемикачі повинні включатися попарно і по діагоналі. Коли включений 1 і 4 ключ, електродвигун обертається за годинниковою стрілкою. 2 і 3 ключі забезпечують роботу двигуна в протилежному напрямку. При включенні двох ключів по вертикалі зліва чи справа відбудеться коротке замикання. Кожна пара по горизонталі закорачиває обмотки двигуна і обертання не відбудеться. На рисунку 5.5 показано, що відбувається, коли якщо міняти положення перемикачів..

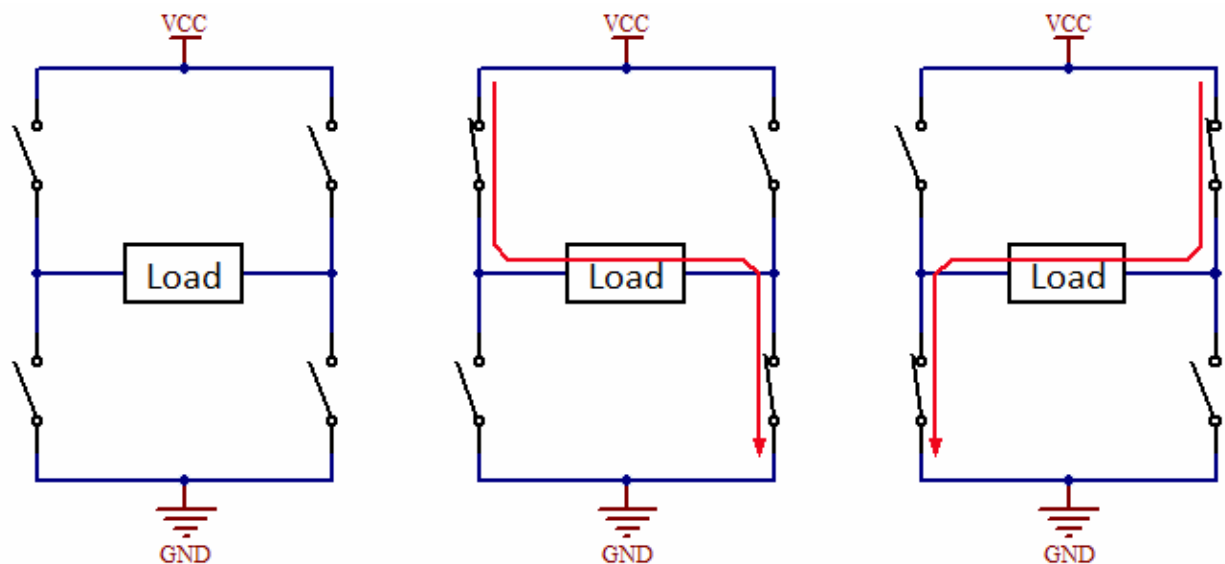


Рисунок 5.5 - Принцип комутації двигуна для обертання в різні боки Н-моста

Як бачимо, за допомогою 4-х перемикачів ми можемо підключати мотор до джерела живлення в різній полярності, що в свою чергу змусить

обертатися його вал в різні боки. Перемикачі можна замінити на реле, або ж на потужні електронні ключі на транзисторах.



*Важливо зауважити що НЕ МОЖНА допускати замикання двох ключів на одній стороні H-моста, оскільки вийде коротке замикання, при проектуванні схеми моста потрібно закласти це правило в логіку і таким чином реалізувати захист.*

Зібрати простий драйвер двигуна постійного струму (або для обмотки крокової двигуна) можна на поширених кремнієвих транзисторах (рис. 5.6).

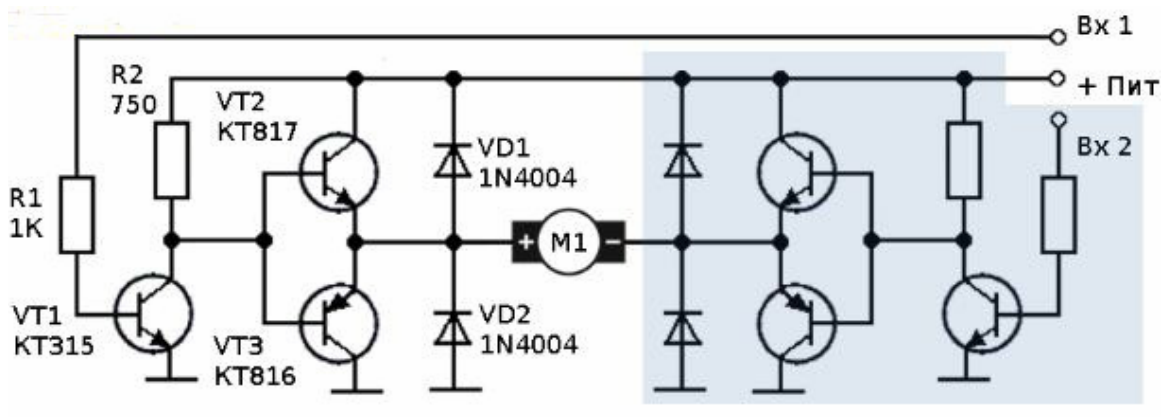


Рисунок 5.6 - Принципова схема простого драйвера електродвигуна на кремнієвих транзисторах.

Такий драйвер дозволяє управляти електродвигуном постійного струму з напругою до 25В (для KT817A, KT816A) і до 45В (для KT817Б-Г, KT816Б-Г) з струмом не більше 3А. При великому робочому і нагрузочному струмі двигуна вихідні транзистори KT817 і KT816 повинні бути встановлені на радіатори достатнього розміру.

Установка діодів VD1-VD2 обов'язкове, вони потрібні для захисту вихідних транзисторів від зворотного струму. На їх місце можна поставити вітчизняні КД105А або інші на більший струм.

Зібравши дві такі схемки (2х6 транзисторів) можна також управляти кроковим двигуном або ж двома двигунами постійного струму.

Інтегральна мікросхема L298 - це потужний універсальний бруківці драйвер для управління двигунами постійного струму, кроковими двигунами, електромагнітними реле і електромагнітами (соленоїдами). У мікросхемі міститься два Н-моста, виконаних на потужних транзисторах, а також логіка сумісна з TTL типи корпусів представлені на рисунку 5.7.



а)



б)

а) Multiwatt15(L298N);

б) PowerSO20 (L298P).

Рисунок 5.7 - Мікросхема L298 в корпусах Multiwatt15 PowerSO20

Основні технічні характеристики:

- робоче напруга - до 46В;
- Максимальний постійний струм - 4А (з радіатором);
- Низьке напруга насичення;
- захист від перегріву;

логічний "0" = напруга до 1,5 В.

Де можна застосувати драйвер на мікросхемі L298:

- управління кроковим двигуном;
- управління двома двигунами постійного струму (DC motors);
- комутація котушок потужних реле;
- управління соленоїдами (електромагнітами).

Нижче на рисунок 5.8 представлена проста схема для драйвера двигунів на мікросхемі L298N. Управління здійснюється за чотирма

проводам (замість шести у L298) завдяки використанню додаткових інверторів в мікросхемі CD4011.

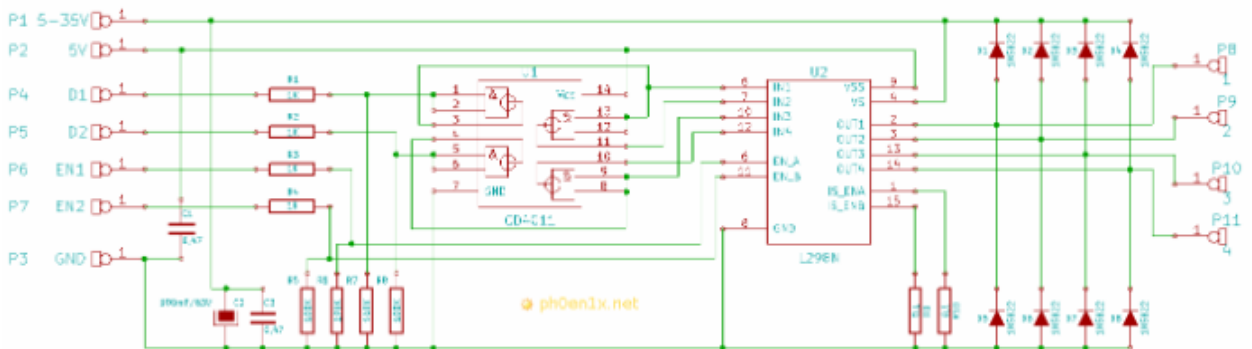


Рисунок 5.8 - Принципова схема драйвера електродвигунів на мікросхемі L298N

Для харчування логіки обох мікросхем потрібно стабілізовану напругу +5 В (P2), можна використовувати інтегральний стабілізатор, наприклад L7805 або ж живити логіку від наявної лінії живлення + 5В. Для подачі напруги живлення на двигуни використовується окрема лінія харчування P1.

Висновки P4, P5 використовуються для установки полярності кожного з каналів, а висновки P6, P7 - дозволяють подачу харчування на каскади (ключі) внутрішнього Н-моста для кожного каналу.

Мікросхему CD4011 можна замінити на вітчизняну К176ЛА7. Діоди Шоттки можна поставити іншого номіналу, на 35В / 4А і більш. Якщо не планується обмежувати струм обмоток двигуна (двигунів) то низькоомні обмежують резистори R9-R10 можна виключити зі схеми, замінивши їх на перемички.

В Internet-е можна замовити готовий модуль на L298, правда в ньому буде 6 входів для управління, на рисунку 5.9 представлені основні модулі Motor Shield розроблений на базі мікросхеми L298N.[129]

*Якщо для живлення двигунів використовується напруга більша ніж 12В то перемичку потрібно прибрати і подавати окремо 5В на виділений для цього коннектор.*



*Якщо харчування двигунів буде здійснюватися від напруги 5-12В то перемичку потрібно утсановити і додаткове харчування 5В не знадобиться.*

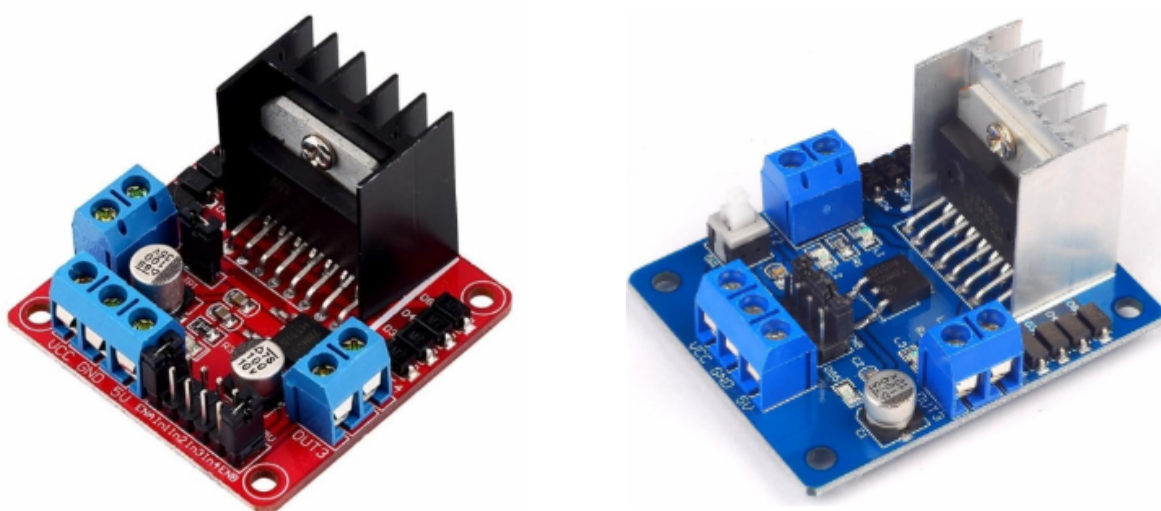


Рисунок 5.9 - Готові модулі на L298

Motor Shield розроблений на базі мікросхеми L298N. Дозволяє управляти двома моторами постійного струму, або одним кроковим двигуном. Максимальний споживаний струм не повинен перевищувати 2 Ампера.

На відміну від відомої мікросхеми L293D, в більш потужний драйвер L298N не вбудовані захисні діоди. Їх необхідно встановлювати в обв'язку мікросхеми додатково. На розглядаємому модулі ми можемо побачити ці діоди (4 штуки зліва і 4 праворуч від мікросхеми).

По виконуваних функцій L298N і L293D схожі трохи більше ніж повністю. Є ті ж керуючі висновки - IN1 і IN2, IN3 і IN4, ENA і ENB. Чергування сигналів різного рівня (високий і низький логічний рівень) на парах висновків задає напрямок обертання моторів.

Висновки ENA (прив'язаний до IN1 і IN2) і ENB (прив'язаний до IN3 і IN4) відповідають за роздільне управління каналами. Можуть використовуватися в двох режимах:

1. **Активний режим.** Управління покладається на контролер (рис.5.10а). Високий логічний рівень дозволяє обертання моторів, низький забороняє, при цьому рівні сигналу на висновках "IN" не має значення. Подаючи на висновки "EN" ШІМ (PWM) сигнал, можна керувати швидкістю обертання моторів.

2. **Пасивний режим.** Підключення висновків "EN" до високого рівня (+5 Вольт). Для цього на платі за допомогою джамперів замикаємо штирі +5V і EN, як показано на (рис.5.10б). У такому режимі ми вже не зможемо регулювати швидкість двигунів, вони завжди будуть працювати на повній швидкості. Зате ми економимо два висновки мікроконтролера. Напрямок обертання і раніше задається станом висновків "IN". А ось щоб зупинити двигун, на парні висновки "IN" подаємо сигнал одного рівня.

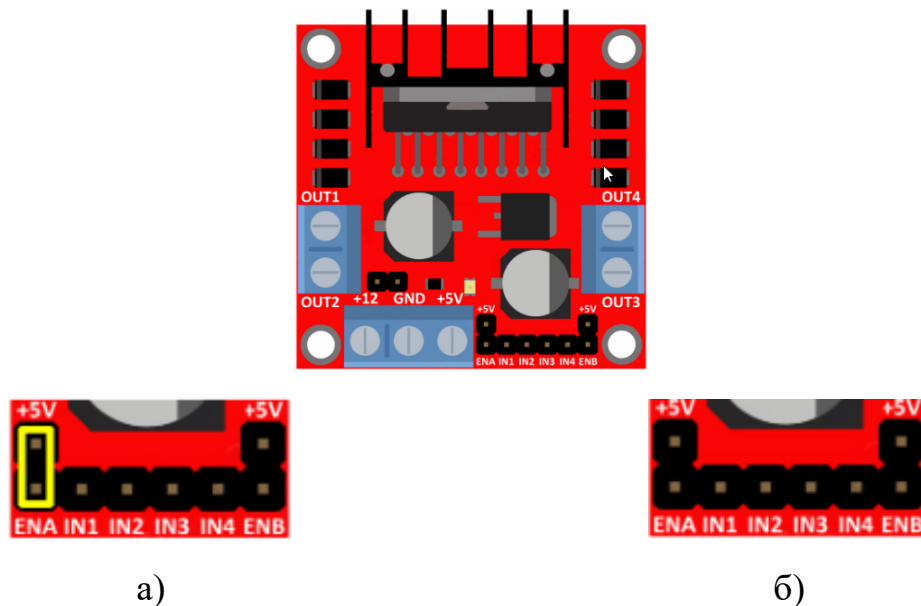


Рисунок 5.10 – Вибір режиму використання модулі на L298N

Підключення живлення на модулі L298N можливий за двома схемами представленим на рисунку 5.11 .



### 5.3. Вибір перетворювача напруги для управління живленням мобільної платформи.

Для живлення Raspberry Pi Mobil 3 пропонується використовувати DC-DC підвищує / понижуючий перетворювач, XL6009 або його аналог.[130] Пристрій зібрано на монолітному ШІМ-регуляторі напруги XL6009 (LM2577) з частотою перемикання 400кГц. Має вбудований тепловий захист і обмеження робочого струму. Регулювання вихідної напруги здійснюється за допомогою високоточної потенціометра. Загальний вигляд DC-DC преобразователя представлений на малюнку 5.12



Рисунок 5.12 - DC-DC підвищує / понижуючий перетворювач,  
XL6009

Застосовується в автомобільних адаптерах для ноутбуків, автомобільних і промислових пристроях, що вимагають напруги вище рівня напруги мережі живлення, портативній електроніці.

DC-DC перетворювач ідеальне рішення для живлення схем з великим падінням напруги (різницею між вхідним і стабілізується напругою). На відміну від класичних стабілізаторів напруги практично не виділяє тепло, має низький струм споживання.

Основні технічні характеристики:

- номінальна вхідна напруга: постійне 5,0 - 32,0 (В);
- Максимальна вхідна напруга: постійне 36,0 (В);
- діапазон регулювання вихідної напруги: постійне 1,25 - 35,0 (В);

- максимальний струм навантаження: 3,0 (А);
- обмеження робочого струму навантаження: 4,0 (А);
- частота перемикачів: 400 (кГц);
- пульсація вихідної напруги: <50 (мВ);
- нестабільність вихідної напруги: <2,5%;
- ККД: 94% (макс);
- робоча температура: -45 °С - + 85 °С;
- захист від перегріву: 125 °С;
- габарити (ШхВхГ): 48х25х14 (мм);
- вага: 24 (г).



*При живленні навантаження з номінальним струмом споживання більше 1,0А необхідно встановити радіатор зверху на мікросхему або плату на радіатор, дотримуючись ізоляцію струмопровідних частин від радіатора.*

Розрахунок максимального струму навантаження здійснюється за формулою:

$$I_{\max} = (3,0A \times U_{\text{input}}) / U_{\text{output}}, \text{ при } U_{\text{output}} \leq 10 \times U_{\text{input}}.$$



*Аналіз форумів і статей показав що даний преобразователь не можна використовувати в зв'язці з одним акумулятором 18650.*

Виходячи з цього на ринку на даний момент існує безліч DC-DC підвищує/понижуючий преобразователів в наслідок чого ви можете вибрати будь-який для вирішення поставлених задач.

#### 5.4. Розробка схеми живлення мобільної платформи

Для управління і розподілом харчування на мобільному роботі пропонується наступна схема підключення основних модулів, представлена на рисунку 5.13

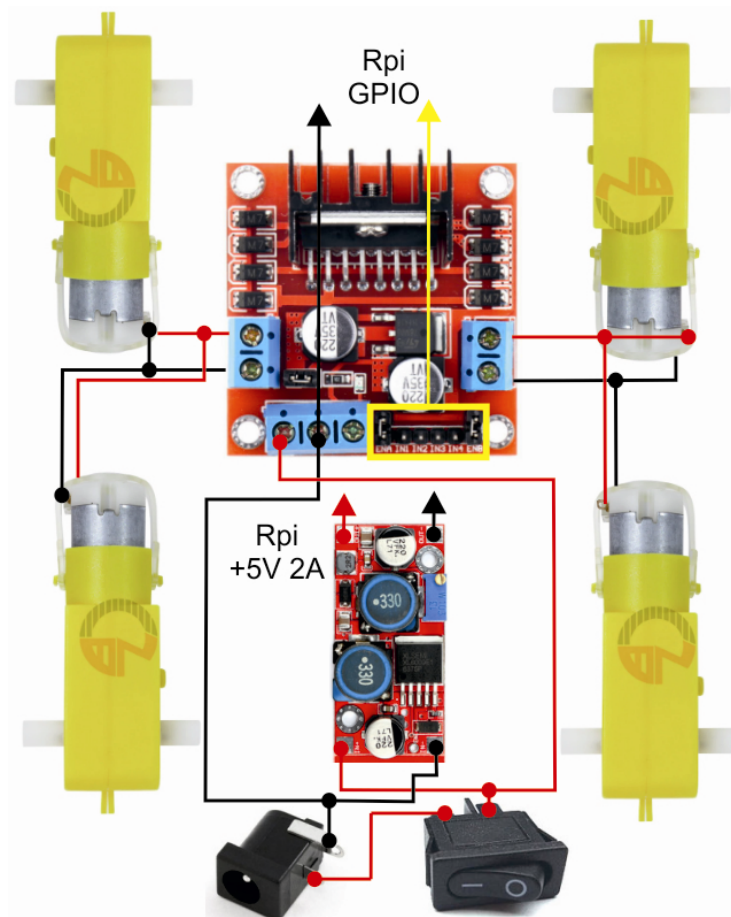


Рисунок 5.13 – Схема живлення мобільного робота

Запропонована схем розподілу живлення, можна споя на макетної платі, і внести в неї корективні в залежності від кількості модулів або датчиків не обхідних для рішень завдань мобільного робота. Для початкового етапу розробки мобільного робота на базі Raspberry Pi дана схемо дозволить виробляти живлення DC двигунів і системи управління за допомогою 3x18500 акумуляторів які можна приєднати через роз'єм або під час розробки та тестування програмного забезпечення використовувати блок живлення 5В

2-3А. На рисунку 5.14 представлено компоновка і спаяна макетна плата на базі 4WD платформи.

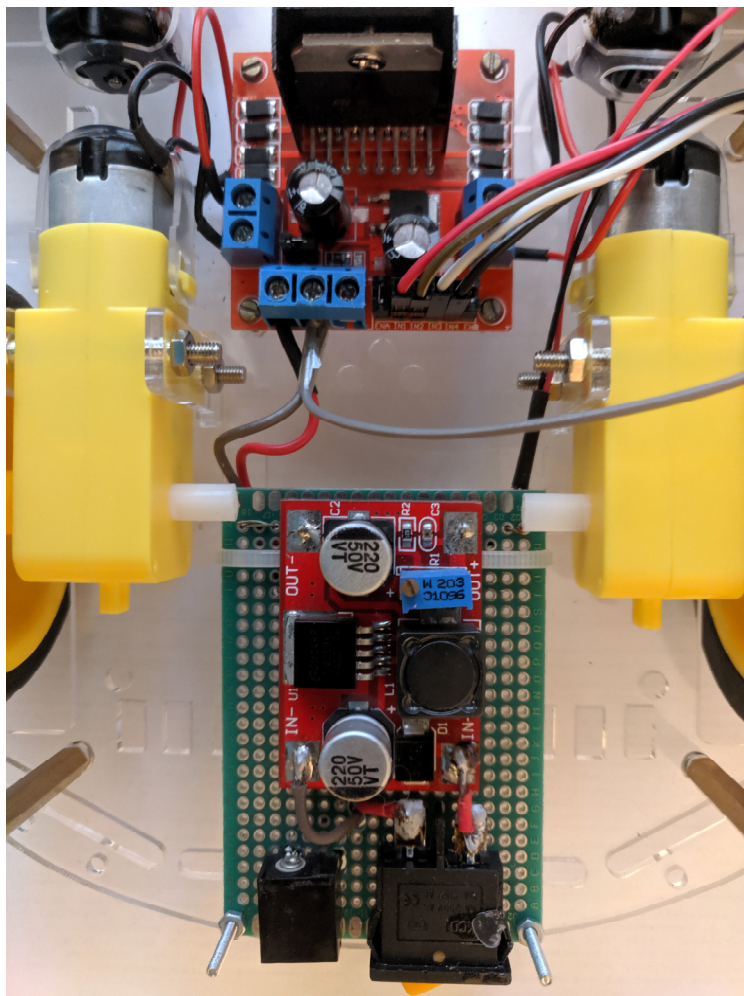


Рисунок 5.14 – Варіант компоновання елементів на макетної платі

На останньому кроці не обходимо зробити підключення модуля L298N до GPIO шині на Raspberry Pi. Схема підключення представлений на рисунку 5.15.

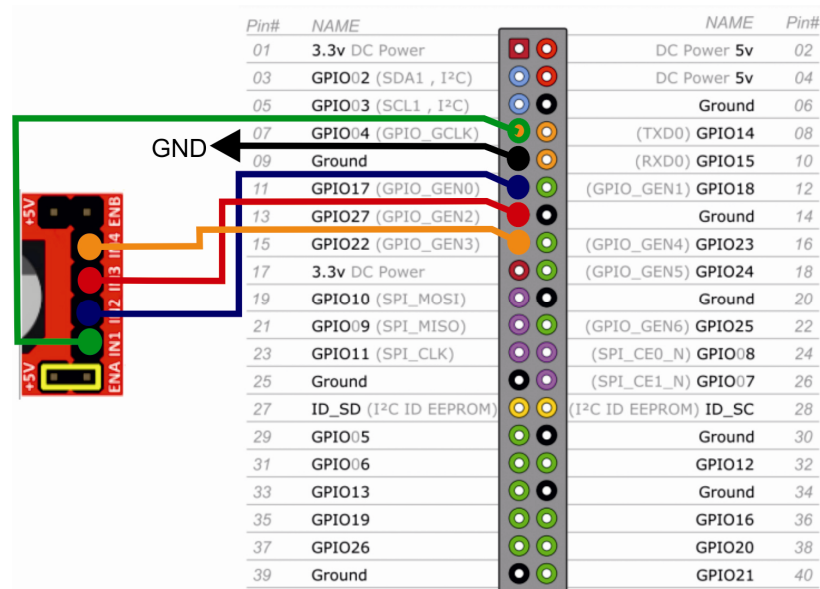


Рисунок 5.15 - Схема підключення модуля L298N до шини GPIO

## 5.5 Розрахунок часу автономної роботи мобільного робота

Розрахунок ємкості акумулятора це дуже важливий етап для проведення експерименту, тому що необхідно знати який період часу може працювати МР автономно. Розрахунок необхідної ємкості здійснюється таким чином:

$$E = P \cdot t \cdot 100, \quad (5.1)$$

де  $P$  – потужність навантаження (кВт);

$t$  – час автономної роботи (години).

Для розрахунку потужності навантаження використовується формула 5.2:

$$P = U \cdot I, \quad (5.2)$$

де  $U$  – номінальна напруга;

$I$  – сила струму.

У таблиці 5.1 приведено необхідні характеристики модулів для розрахунку необхідної ємкості акумулятору.

Таблиця 5.1 – Технічні характеристики модулів МР

Назва модуля	Сила струму $I$ , А	Номінальна напруга $U$ , В	Потужність навантаження $P$ , кВт
Raspberry Pi 3 Model B [27]	0,15	5	$7,5 \cdot 10^{-4}$
L298N (двигуни)	2	16	$2,4 \cdot 10^{-2}$
L298N (логіка)	0,36	5	$11,8 \cdot 10^{-4}$

Виходячи з таблиці 5.1 необхідно підрахувати сумарну потужність навантаження по формулі 3.3:

$$P_{\text{сум}} = \sum P_i. \quad (5.3)$$

Отже  $P_{\text{сум}}$  дорівнює  $336,6 \cdot 10^{-4}$  кВт. Підставивши усі необхідні змінні у формулу (5.2) отримаємо [28]:

$$E = 336,3 \cdot 10^{-4} \cdot 3 \cdot 100 = 10,089 \text{ А} \cdot \text{г}.$$

Маючи 4 акумулятори стандарту 18650 із ємкістю  $3,4 \text{ А} \cdot \text{г}$ , сумарна ємкість становить  $13,6 \text{ А} \cdot \text{г}$ , що забезпечить стабільну роботу МР упродовж більш ніж 3 годин.

Розрахуємо максимальний час роботи МР із ємкістю  $13,6 \text{ А} \cdot \text{г}$ . Виходячи з формули (5.2) отримаємо формулу (3.4):

$$t = \frac{E}{P \cdot 100}, \quad (5.4)$$

де  $t$  – максимальний час роботи МР.

Підставивши змінні у формулу (5.4) отримаємо:

$$t = \frac{13,6}{336,3 \cdot 10^{-4}} = 4,044 \text{ год.}$$

Отримавши час роботи від акумулятора фактичної ємкості, можна зробити висновок, що при ємкості акумуляторного блоку, що дорівнює 13,6 А·г, МР може стабільно працювати до 4 годин.

У загальному випадку поки на базі запропонованої послідовності зборки не обходимо конструктивно дабавить сервомотор Tower Pro 9g SG90 для Arduino PIC ARM AVR для установки на нього Web камери для Raspberі Рі, що в майбутньому дасть можливість горизонтально управляти її кутами огляду. Рисунок вже із зібраним мобільним роботом на базі колісної 4WD платформи представлений на 5.16.

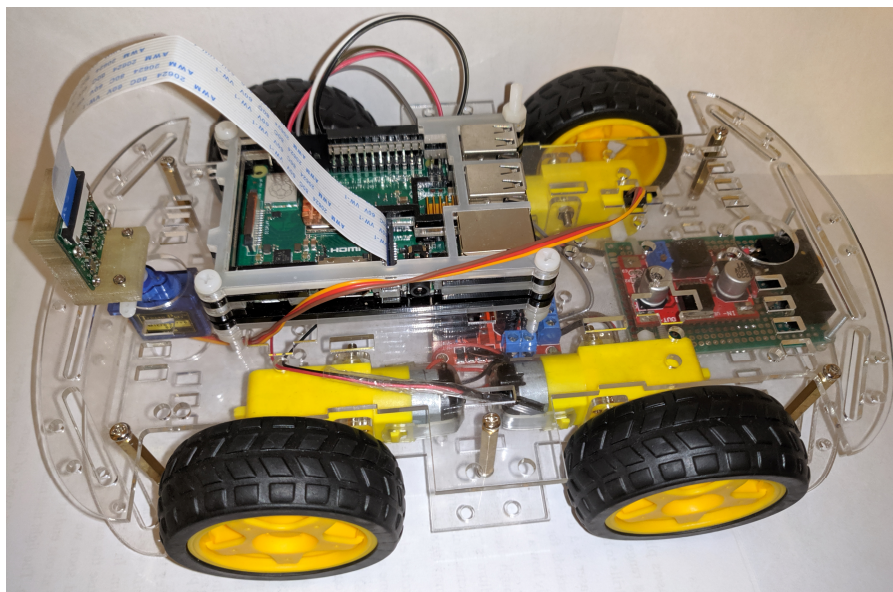


Рисунок 5.16 – Мобільний робот на базі 4WD платформи.

## 5.5 Контрольні питання

1. Для чого потрібен редуктор в DC моторі ?;
2. Поясніть принцип роботи схеми H-моста;
3. Перерахуйте параметри модуля на базі мікросхеми L298N ?;
4. Чим отличаються драйвери L293D і L298N ?;
5. Перелічіть назва керуючих висновків в драйвері L293D і L298N.
6. У яких режимах можна управляти драйвером L293D ?;
7. Для чого використовується DC-DC підвищує / понижуючий перетворювач ?;
8. Перерахуйте основні характеристики преобразователя XL6009;
9. За якою формулою проводиться розрахунок максимального струму навантаження ?;
10. Як можна розрахувати час автономної роботи мобільного робота ?;
11. Де можна застосувати драйвер на мікросхемі L298 ?;
12. Чому дорівнює логічний "0" в драйвері на базі мікросхеми L298 ?;
13. Розкажіть принцип комутації двигуна для обертання в різні боки на базі H-моста ?.

## 6. ПІДКЛЮЧЕННЯ ТА ПРОГРАМУВАННЯ ДАТЧИКІВ МОВОЮ RAYTON

### 6.1 Підключення ультразвукового датчика HC-SR04

Особливістю роботи даного датчика є, то що після запуску вимірювання, відправки на пін Trig імпульсу в 10мк.сек, сенсор генерує ту саму серію імпульсів, чекає відповіді і потім відправляє на пін Echo сигнал, тривалість якого дорівнює часу який буде потрібний щоб отримати свій відправлений сигнал, тобто подвоєне відстань до об'єкта.

HC-SR04 датчик має 4 піна:

- VCC - живлення сенсора. 5 В -напрямою з піна номер 2 мого Raspberry;

- Grd - пін 6 на Raspberry Pi;

- Trig. - на цей пін необхідно надіслати сигнал тривалістю 10 мікросекунд для того щоб сенсор заробив. Сенсор спрацьовує і посилає звуковий ряд з 8 імпульсів в 40кГц і приймає їх на свій датчик;

- Echo з цього піна ми зчитуємо інформацію з датчик. Особливістю роботи даного датчика вляється, то що після запуску вимірювання, відправки на пін Trig імпульсу в 10мк сек, сенсор генерує ту саму серію імпульсів, чекає відповіді і потім відправляє на пін Echo сигнал, тривалість якого дорівнює часу який буде потрібний щоб отримати свій відправлений сигнал, тобто подвоєне відстань до об'єкта.[131]

З цих даних можна порахувати на скільки часто ми можемо отримувати дані про найближчі об'єкти для цього не обходимо врахувати:

- мінімальна затримка між вимірами повинна бути 50 мс;

- 10 мікросекунд на генерацію збудливого сигналу (можна опустити);

- $1/340$  секунди = 3мс на вимір максимально доступного відстані.

Припустимо 500см максимально доступне відстань. Звук буде проходити в

обох напрямках, тобто йому потрібно пройти 1м. Швидкість звуку приблизно 340мс. Отримує звук пролетить цей метр за 1/340-ву секунди або приблизно 3мс.

- ще 3мс. цей час вимагає датчику щоб передати сигнал на пін Echo.

В підсумку.  $50 + 3 + 3 = 56$ мс. Це мінімально дозвіл між одержуваної інформації про відстань до об'єкту. Наприклад, якщо МР рухається зі швидкістю 20 км/год це дорівнює 5,5мс, то за зазначений час 56мс вона проїде 30см. Якщо швидкості у вас не високі, точності вимірювань буде достатньо. Схем підключення датчика HC-SR04 представлений на рисунку 6.1

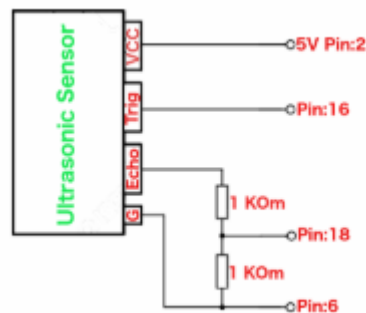


Рисунок 6.1 – Схем підключення датчика HC-SR04

Додаткова особливість. Живлення схеми йде від 5В і вихідна напруга вона видає 5В з Echo Піна, з цього потрібно його зменшити перед подачею на вхідний пін Raspberry Pi. Це можна зробити найпростішим дільником напруги складається з двох резисторів. У схемі представленої на рисунку 6.1 є два резистора в 1 кому. Знімаючи сигнал з одного з них получется 2,5 вольтової сигнал. Даного рівня достатньо для розпізнавання сигналу. Мінімальний рівень сигналу для визначення наявності сигналу і визначення його як High це 1,3В. Максимальний рівень 3,3. Тому 2,5 В більш ніж достатньо.

Для написання програми для обробки інформації з ультрозвуковго датчика HC-SR04 не обходимо підключити дві бібліотеки.

```
inport Rpi.GPIO as GPIO
```

```
inport time
```

У коді визначаємо два пін. Один як вхідний, інший як вихідний. З останнього будемо відправляти імпульс для збудження датчика, а з першого будум зчитувати інформацію з сенсора.

```
GPIO.setmode(GPIO.BOARD)
```

```
TRIG = 16
```

```
ECHO = 18
```

```
GPIO.setup(TRIG, GPIO.OUT, initial=0)
```

```
GPIO.setup(ECHO, GPIO.IN)
```

Далі необхідно почекати мінімум 50мс - мінімальна затримка між вимірами. В цьому скрипті вона не особливо потрібна, але якщо використовувати цей скрипт як модуль у великій програмі, яка буде його викликати, то краще використовувати цю затримку при цьому краще використовувати зацикленість вимірювань відстані.

Після це необхідно надіслати сигнал на пін Trig сенсора. Тривалість сигналу 10мк сек:

```
GPIO.output(TRIG,1)
```

```
time.sleep(0.00001)
```

```
GPIO.output(TRIG,0)
```

Далі необхідно чекати поки не почне надходити сигнал на порт Echo. Потрібно відзначити потрібний проміжок часу сигналу. Як тільки сигнал почне надходити, необхідно зрозуміти коли він закінчиться. Як тільки він перерветься - відзначити коли він закінчився. Робиться це за допомогою змінних start та stop, плюс модуль часу time.

```
while GPIO.input(ECHO) == 0:
```

```
pass
```

```
start = time.time()
```

```
while GPIO.input(ECHO) == 1:
```

```
pass
```

```
stop = time.time()
```

Знаючи початковий і кінцевий час сигналу можна знайти його тривалість і отже відстань пройдене сигналом. Не забуваємо, що сигнал проходить подвійну відстань і результат (Відстань = Швидкість / Час) потрібно ділити на 2:

```
print "Distance = ",(stop - start) * 17000,"sm"
```

Приклад програмного коду на наведений в додатку.

## **6.2 Підключення та програмування датчика температури і вологості DHT22**

DHT22 - це універсальний і недорогий датчик вологості, який також може розраховувати температуру в довкілля МР. Цей датчик має відносно велику відстань передачі, що дозволяє датчику передавати дані по дротах на відстані до 20 м від Raspberry Pi.[132]

Як бонус, DHT22 - це цифровий датчик з вбудованим аналого-цифровим перетворювачем. Конвертер значно полегшує підключення датчика до Raspberry Pi, оскільки вам не потрібно мати справу з будь-якими додатковими чіпами. Самым большим недостатком датчиков DHT11 и DHT22 является то, что они являются довольно медленными датчиками. Они имеют частоту дискретизации один раз в секунду для DHT11 и один раз каждые 2 секунды для DHT22.

DHT22 - це універсальний і недорогий датчик вологості, варто зауважити що підключення DHT22 до Raspberry Pi аналогічно з датчиками вологості DHT11 і AM2302, оскільки всі вони використовують однакові терморегулятори.

Щоб спростити складання ланцюга датчика вологості, ми включили терморегулятори датчика DHT22. Терморегулятори датчика DHT22 наведено на рисунку 6.2

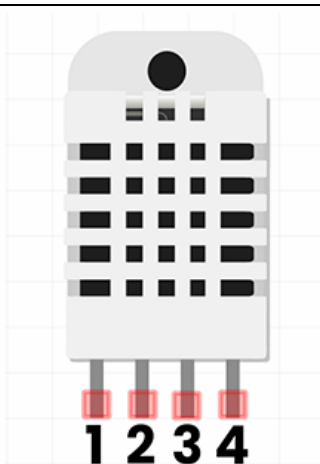


Рисунок 6.2 - Распінровка датчика DHT22

Опишемо кожен пін і його призначення:

- пін 1 - це VCC (джерело живлення);
- пін 2 - це дані (сигнал даних);
- пін 3 - недійсний (не з'єднувати);
- пін 4 - Gnd;

Для підключення датчика DHT22 не обходимо виконати наступний алгоритм підключення:

- помістіть резистор 10 кОм між контактом 1 і контактом 2 DHT22;
- підключіть контакт 1 DHT22 до фізичного контакту 1 (3v3) на Rpi;
- підключіть контакт 2 DHT22 до фізичного контакту 7 (GPIO4) на Rpi;
- підключіть контакт 4 DHT22 до фізичного контакту 6 (GND) на Rpi;

Перш ніж ми почнемо програмувати скрипт для датчика вологості Raspberry Pi, не обходимо оновити Raspberry Pi. Для цього не обходимо виконати наступні дві команди.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Встановіть обидва python3-dev і python3-pip, виконавши команду нижче.

```
sudo apt-get install python3-dev python3-pip
```

Перш ніж продовжити установку бібліотеки DHT, необхідно виконати наступну команду, щоб переконатися, що у нас є останні версії пакетів `setuptools`, `wheel` і `pip python`.

```
sudo python3 -m pip install --upgrade pip setuptools wheel
```

Тепер, використовуючи `pip`, проведемо установку DHT-бібліотеку Adafruit на Raspberry Pi. Дана бібліотека Python дозволить реалізувати взаємодію з датчиком DHT22.

Як бонус бібліотека також підтримує датчики вологості / температури DHT11 і AM2302, що робить її відмінною бібліотекою для вивчення того, як її використовувати.

Виконайте наступну команду, щоб встановити бібліотеку DHT на Raspberry Pi.

```
sudo pip3 install Adafruit_DHT
```

У цьому файлі нам потрібно ввести такі рядки коду. Нижче кожен розділ коду буде прокоментований і пояснений у міру написання скрипта.

```
import Adafruit_DHT
```

Цей рядок імпортує бібліотеку «Adafruit\_DHT», яка була проінсталірована за допомогою `pip`.

Дана бібліотека буде використана для спілкування і взаємодії з датчиком DHT22. Бібліотека дозволяє нам дуже легко отримати температуру і вологість від датчика за допомогою кількох рядків коду Python.

```
DHT_SENSOR = Adafruit_DHT.DHT22
```

У цьому рядку определяється константу для датчика, яку будемо використовувати для бібліотеки DHT. У бібліотеці Adafruit DHT є три варіанти вибору, в залежності від типу використовуваного датчика.

В даному навчальному посібнику використовується DHT22, тому определили нашу константу «DHT\_SENSOR» як «Adafruit\_DHT.DHT22».

Потім визначаємо константу, яка називається «DHT\_PIN», яка призначена для зберігання номера контакту GPIO, до якого підключений DHT22 на Raspberry Pi.

```
DHT_PIN = 4
```

Що використовується число відноситься до нумерації ВСМ висновків GPIO. В даному випадку це означає, що нам потрібно використовувати число «4», так як буде використовуватися висновок GPIO4 для з'єднання для передачі даних. Якщо буде використовуватися інший висновок GPIO, переконайтеся, що його номер буде вказано замість цей номер відповідним чином.

```
while True:
```

```
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,
DHT_PIN)
```

На початку кожного циклу використовується бібліотека Adafruit DHT для вилучення як вологості, так і температури з датчика DHT22, через константу DHT\_SENSOR і DHT\_PIN.

Функція «read\_retry» буде постійно намагатися отримати дані від датчика вологості.

```
if humidity is not None and temperature is not None:
```

```
    print("Temp={0:0.1f}*C    Humidity={1:0.1f}%"
format(temperature,
humidity))
```

```
else:
```

```
    print("Failed to retrieve data from humidity sensor")
```

Необхідно перевірити, щоб бібліотека DHT повертала не обхідні дані.

Якщо змінні «humidity» або «temperature» рівні «None», то буде повертатися повідомлення, яке повідомить кінцевому користувачеві, що не вдалося отримати дані з датчика вологості.

Якщо у нас значення в обох цих змінних, то ми виводимо повідомлення, що показує кінцевому користувачеві температуру і вологість, які отримані з датчика. Повний текст сткріпта наведено нижче.

```
import Adafruit_DHT
```

```
DHT_SENSOR = Adafruit_DHT.DHT22
```

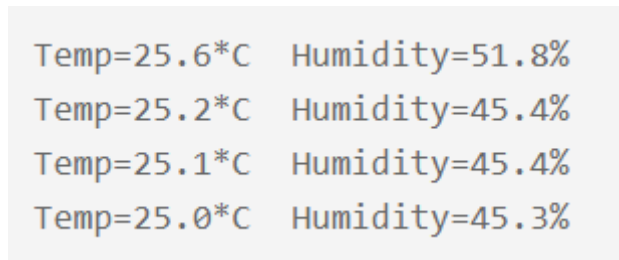
```
DHT_PIN = 4
```

```

while True:
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,
DHT_PIN)
    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}*C    Humidity={1:0.1f}%"
.format(temperature,
humidity))
    else:
        print("Failed to retrieve data from humidity sensor")

```

При виконань прведення вище скрепта користувачеві повинно відобразитися температура і вологість в командному рядку. Приклад результату роботи скрипта наведені на рисунку 6.3.



```

Temp=25.6*C    Humidity=51.8%
Temp=25.2*C    Humidity=45.4%
Temp=25.1*C    Humidity=45.4%
Temp=25.0*C    Humidity=45.3%

```

Рисунок 6.3 – Відображення значень температура і вологість в командному рядку.

Нижче наведемо приклад скрипт для збору інформації з датчика DHT22 для майбутнього аналізу на базі його записи в файл. Файл зможе бути прочитаний як Microsoft Excel або Google Spreadsheets що дозволить реалізувати облочний контроль за навколишнім середовищем навколо МР.

Вдосконалюємо вище наведений скрипт за допомогою добовленням нових рядків програмного коду. Нижче пояснюються тільки нові доповнення до існуючому скрипту.

```

import os
import time
import Adafruit_DHT

```

Додамо бібліотеки «os» і «time», які будуть перевірити перевірку існування CSV-файл, перш ніж проводити записати в нього даних з датчика.

Бібліотеку «time», щоб ставити мітки часу в кожній новій рядку з поточною датою і часом коли були отримані дані з датчика.

```
DHT_SENSOR = Adafruit_DHT.DHT22
```

```
DHT_PIN = 4
```

Використовуємо фрагмент скрипта з попереднього прикладу і додамо новий фрагмент:

```
try:
```

```
f = open('/home/pi/humidity.csv', 'a+')
```

```
if os.stat('/home/pi/humidity.csv').st_size == 0:
```

```
    f.write('Date,Time,Temperature,Humidity\r\n')
```

```
except:
```

```
    pass
```

Реалізуємо все в операторі try для обробки будь-яких можливих помилок. Реалізуємо откритіє дескриптора файлу «/home/pi/humidity.csv» з застосованим тегом «+». Тег + означає, що будь-які дані, записані в файл, будуть додані в його кінець.

Далі використовуємо бібліотеку «os», щоб дізнатися, чи були записи будь-коли в цей файл раніше. Якщо розмір повертається як 0, записуємо початкову рядок в файл. Цей рядок буде містити заголовки наших стовпців, щоб ви користувачеві було простіше аналізувати дані.

```
while True:
```

```
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,  
DHT_PIN)
```

Цикл while дозволяє реалізувати використання бібліотеки DHT для повторного зчитування даних з датчика вологості DHT22.

Нижче наведений фрагмент скрипта вигляді функції дозволяє зберігати дані, що зчитуються з датчиків, в змінні «humidity» і «temperature».

```
if humidity is not None and temperature is not None:
```

```
    f.write('{0},{1},{2:0.1f}*C,{3:0.1f}%\r\n'.format(time.strftime('%m/%d/%y'),  
time.strftime('%H:%M'), temperature, humidity))
```

*else:*

```
print("Failed to retrieve data from humidity sensor")
time.sleep(30)
```

Дана зміна в цьому фрагменті коду полягає в тому, що замість виведення температури і вологості на консоль робитися записи в файл «*humidity.csv*» за допомогою дескриптора файлу, який був відкритий раніше. Далі проводимо форматування тексту, записаний в файл, щоб включити поточну дату, поточний час і отформатовану температуру і вологість з датчика. В кінці циклу встановлюємо час виконання сценарію на 30 секунд, якщо не обходимо змінити час опрашівння датчика, не обходимо змінити це число в залежності від вимозі.

Доповнений скрипт має наступний загальний вид.

```
import os
import time
import Adafruit_DHT

DHT_SENSOR = Adafruit_DHT.DHT22
DHT_PIN = 4

try:
    f = open('/home/pi/humidity.csv', 'a+')
    if os.stat('/home/pi/humidity.csv').st_size == 0:
        f.write('Date,Time,Temperature,Humidity\r\n')
except:
    pass

while True:
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR,
DHT_PIN)
    if humidity is not None and temperature is not None:
        f.write('{0},{1},{2:0.1f}*C,{3:0.1f}%\r\n'.format(time.strftime('%m/%d/%y'),
time.strftime('%H:%M'), temperature, humidity))
```

*else:*

```
print("Failed to retrieve data from humidity sensor")
```

```
time.sleep(30)
```

На рисунку 6.4 наведено приклад вмісту файлу «*humidity.csv*» який має наступний архітектуру.

```
Date,Time,Temperature,Humidity
05/04/19,05:47,22.3*C,50.6%
05/04/19,05:48,22.2*C,50.4%
05/04/19,05:48,22.2*C,50.6%
05/04/19,05:49,22.2*C,50.4%
05/04/19,05:49,22.2*C,50.4%
05/04/19,05:50,22.2*C,50.3%
05/04/19,05:50,22.2*C,50.3%
05/04/19,05:51,22.2*C,50.2%
```

Рисунок 6.4 - Вміст файлу «*humidity.csv*»

### **6.3. Підключення та програмування управління сервоприводами для системи комп'ютерного зору**

Для управління поворотом камери на МР не обходимо використовувати сервопривід моделі TowerPro SG90.[133] Загальний вигляд сервоприводу представлена на рисунку 6.5.



Рисунок 6.5 - Загальний вигляд сервоприводу моделі TowerPro SG90

Параметри сервоприводу моделі TowerPro SG90:

Робоча напруга: 4.8 DC;

Швидкість: 60 градусів за 0.1с;

Крутний момент: 1.6 кг см;

Діапазон ширини імпульсу: 0.5 мс - 2.5 мс.

Підключення проводиться наступним способом: Від сервоприводу йде шлейф з трьох проводів:

- червоний - живлення підключається до контакту 5V (пін 2) або безпосередньо до джерела живлення;

- коричневий або чорний - земля GND (пін 6 Raspberry Pi)

Дане серво не може обертатися на 360 градусів. Воно має нейтральну позицію (Neutral) і кордони в нуль і 180 градусів. Для встановлення серво в нейтральну позицію, необхідно подати високий сигнал тривалістю 1.5 мс, в 0 градусів - 0.5 мс, в 180 градусів - 2.5 мс. Як представлено на рисунку 6.6.

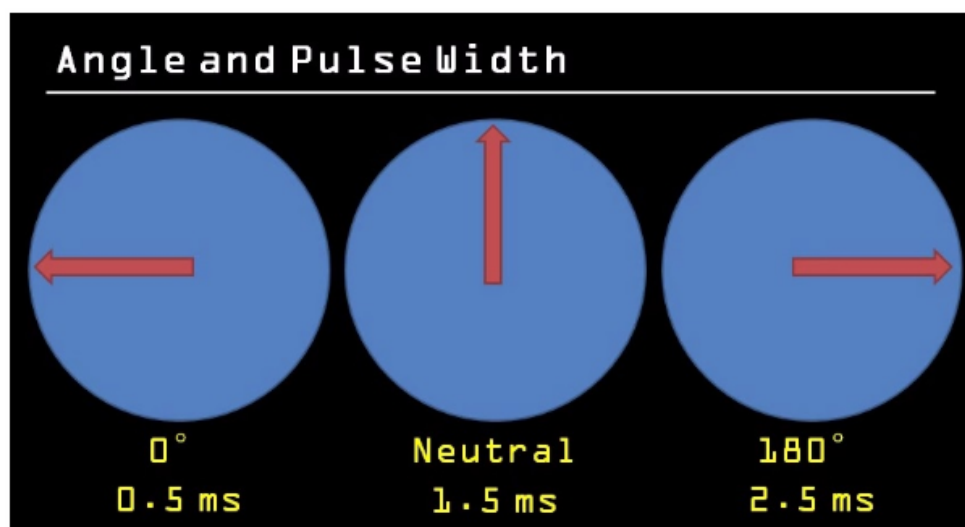


Рисунок 6.6 - Тривалість високого сигналу на кут повороту сервоприводу

Raspberry Pi не має аналогового виходу, але ми можемо імітувати його, використовуючи підхід ШІМ (широотно-імпульсна модуляція). Ми будемо

робити цифровий сигнал з фіксованою частотою, де ми змінимо ширину послідовності імпульсів як наведено на рисунку 6.7.

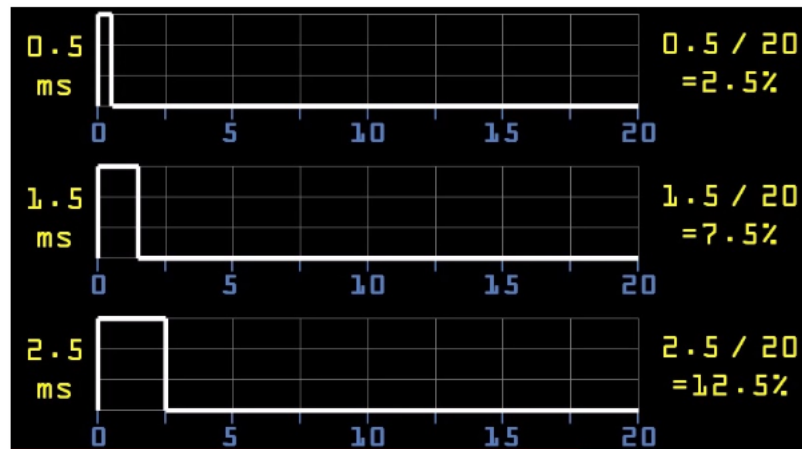


Рисунок 6.7 – Графік залежності довжини імпульсів на положення сервоприводу

Напишем следующий скрипт на языке Python:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7,GPIO.OUT)

p = GPIO.PWM(7,50)
p.start(7.5)

try:
    while True:
        p.ChangeDutyCycle(7.5)
        time.sleep(1)
        p.ChangeDutyCycle(12.5)
        time.sleep(1)
        p.ChangeDutyCycle(2.5)
        time.sleep(1)
```

*except KeyboardInterrupt:*

*p.stop()*

*GPIO.cleanup()*

Наведений скрипт дозволяє управляти сервоприводом через сьомий пін за допомогою ШІМ з частотою 50 Герц: GPIO.PWM (7,50). В додатків наведено скрипт для управління двома сервоприводами що дозволить забезпечити горизонтальном і вертикальний кут повороту камерою.

#### **6.4. Контрольні питання**

1. Перерахуйте Піни датчика HC-SR04;
2. За якою формулою рассчитиветься відстань до об'єкта для датчика HC-SR04;
3. Опишіть призначення пинов датчика DHT22;
4. Для чого використовується резистор 10 кОм при підключень датчика DHT22 ?;
5. Перелічіть номери пинов на шині GPIO до яких можна підключть датчик DHT22 ?;
6. Яка бібліотека в Pyuten дозволяє працювати з датчиком DHT22 ?;
7. Поясніть принцип роботи ультрозвукового датчика HC-SR04 ?;
8. Чим за призначенням отличаються піни Echo і Trig ?;
9. Чому дорівнює мінімальное дозвіл між одержуваної інформації про відстань до об'єкту датчика HC-SR04 ?;
10. Перерахуйте основні характеристики сервоприводу моделі TowerPro SG90;
11. Чому використовується ШІМ для управління сервоприводу ?;
12. Як залежить довжини імпульсів на положення сервоприводу ?.

## 7. РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ МОБІЛЬНОЇ ПЛАТФОРМИ НА БАЗІ МОВИ РАУТЕН



*Для успішної реалізації управління мобільною платформою на базі Raspber Pi автори рекомендую уважно ознакоміться з розділами 2 і 3. даної книги.*

### 7.1 Скрипт управління DC двигунами на мобільною платформою

На першому етапі тесірованія управління DC двигунами на мобільною платформою 4WD напишемо програму на мові Python без використання PWM для управління швидкістю руху.

```
import time
import RPi.GPIO as GPIO

# Подготавливаем пины GPIO.
GPIO.cleanup()
GPIO.setmode(GPIO.BCM)
#Hiden errors
GPIO.setwarnings(False)
#колеса(all) install
GPIO.setup(4,GPIO.OUT)
GPIO.output(4,GPIO.LOW)
GPIO.setup(17,GPIO.OUT)
GPIO.output(17,GPIO.LOW)
GPIO.setup(27,GPIO.OUT)
GPIO.output(27,GPIO.LOW)
GPIO.setup(22,GPIO.OUT)
GPIO.output(22,GPIO.LOW)
```

*#-----Forfard -----*

*GPIO.output(17,GPIO.HIGH)*

*GPIO.output(27,GPIO.HIGH)*

*# чекаємо 5 секунд.*

*time.sleep(5)*

*GPIO.output(17,GPIO.LOW)*

*GPIO.output(27,GPIO.LOW)*

*#----Down-----*

*GPIO.output(4,GPIO.HIGH)*

*GPIO.output(22,GPIO.HIGH)*

*# чекаємо 5 секунд.*

*time.sleep(5)*

*GPIO.output(4,GPIO.LOW)*

*GPIO.output(22,GPIO.LOW)*

*# ---left -----*

*GPIO.output(27,GPIO.HIGH)*

*GPIO.output(4,GPIO.HIGH)*

*# чекаємо 5 секунд.*

*time.sleep(5)*

*GPIO.output(27,GPIO.LOW)*

*GPIO.output(4,GPIO.LOW)*

*#----righet-----*

*GPIO.output(17,GPIO.HIGH)*

```
GPIO.output(22,GPIO.HIGH)
```

```
# чекаємо 5 секунд.
```

```
time.sleep(5)
```

```
GPIO.output(17,GPIO.LOW)
```

```
GPIO.output(22,GPIO.LOW)
```

```
#----swich of all gpio -----
```

```
GPIO.output(17,GPIO.LOW)
```

```
GPIO.output(27,GPIO.LOW)
```

```
GPIO.output(4,GPIO.LOW)
```

```
GPIO.output(22,GPIO.LOW)
```

Широтно-імпульсна модуляція (PWM (pulse-width modulation) або ШІМ) - процес управління потужності методом пульсуючого включення і виключення приладу це спосіб контролювання харчування. У даних дослідженнях ми будемо застосовувати PWM, щоб контролювати кількість струму, який проходить через мотор і, як наслідок, швидкість обертання мотора. Кожну 1/500 секунди роз'єм PWM подає імпульс. Довжина цього імпульсу контролює кількість енергії, яке отримує мотор. Якщо імпульс не надходить, мотор не буде обертатися. Якщо імпульс короткий, мотор буде обертатися повільно. Якщо імпульс довжиною в половину проміжку між імпульсами, мотор отримає половину від енергії, яку він міг би отримати, якби імпульс тривав весь проміжок між імпульсами. На рисунку 7.1 приведена на діаграма сигналу PWM на Raspberry Pi.

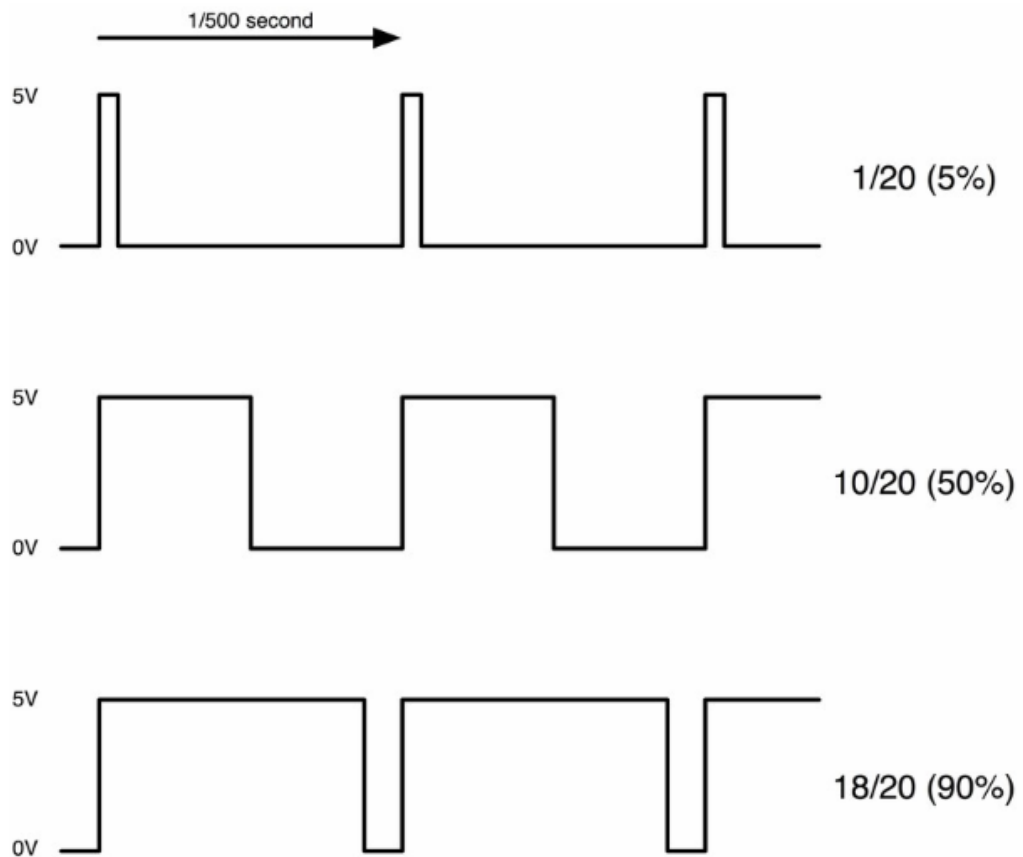


Рисунок 7.1 - Діаграма сигналу ШІМ на Raspberry Pi.

## 7.2 Підключення і настройка відео стріму з камери Raspbian Pi

Цей підрозділ присвячений вирішенню завдання забезпечити розробляемому мобільному роботу можливість отримувати інформацію про навколишнє його середовищі. Для цього на початковому етапі будемо використовувати потокову передачу відео за допомогою Raspberry Pi та камери, яка буде транслювати відео в системі реального часу на віддалену Web-сторінку. Таке рішення дозволить забезпечити доступ з будь-якого пристрою, що має браузер і підключеного до нього мережі Internet. Отже дасть можливість розробляти системи ідентифікації і розпізнання об'єктів з використанням різних математичних методів (нейронні мережі, машинне навчання, і.т.д.).



Якщо ви використовуєте модуль камери Raspberry Pi, вам необхідно включити програмне забезпечення камери в вашому Raspberry Pi, щоб використовувати його. У середовищі робочого столу перейдіть у вікно «Конфігурація Raspberry Pi» в меню «Налаштування», відкрийте вкладку «Інтерфейси» і включіть камеру, як показано на малюнку нижче. Як представлено на рисунку 7.2

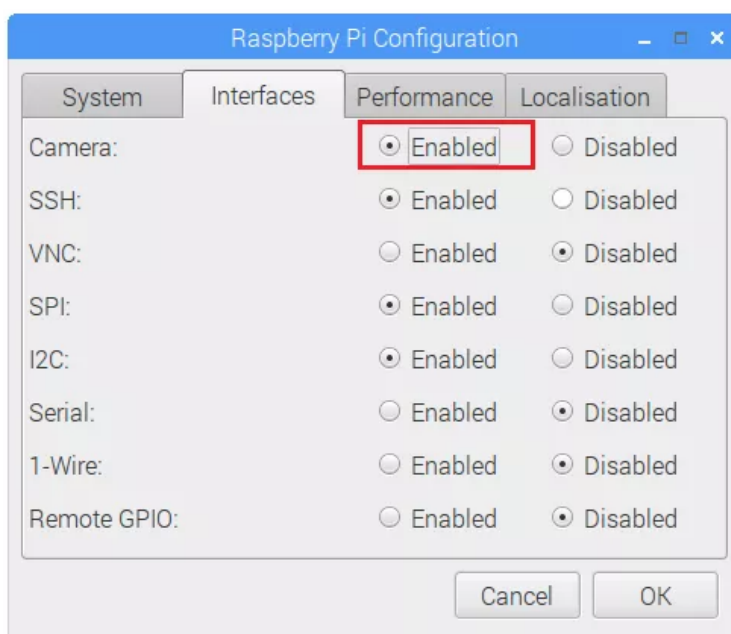


Рисунок 7.2 - Вікно Raspberry Pi Configuration

Або скористатися *Terminal*, де необхідно ввести наступну команду  
`sudo raspi-config`

Ви повинні побачити інструмент налаштування програмного забезпечення Raspberry Pi. Виберіть Interfacing Options і активуйте (Enable) P1 Camera за допомогою <Select> як показано на рисунку 7.3

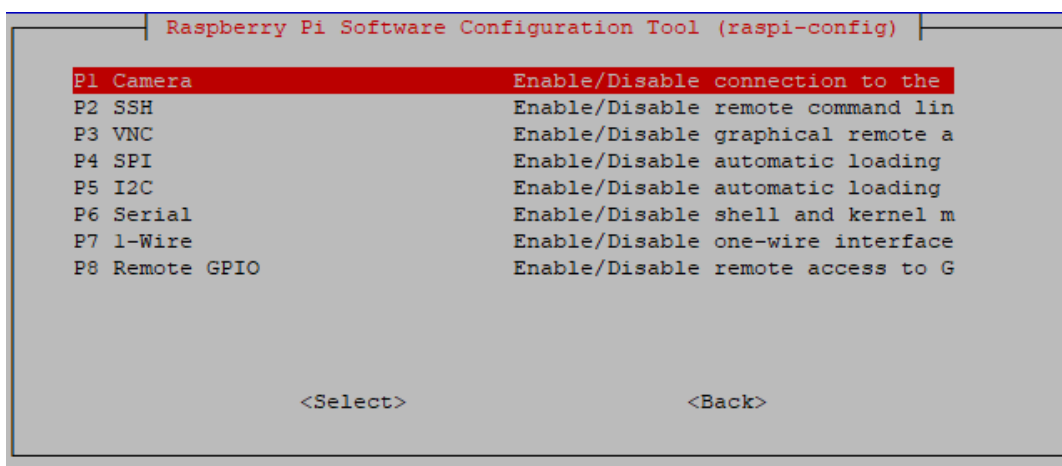


Рисунок 7.3 – Вікно настройки raspi-config.

Щоб отримати доступ до веб-сервера потокового відео, вам необхідно знати свій IP-адресу Raspberry Pi. Для цього використовуйте наступну команду:

```
ifconfig
```

Вам буде надана купа інформації, включаючи ваш IP-адреса Raspberry Pi. У моєму випадку IP-адреса RPi - 192.168.1.5. Це можна побачити на рисунку 7.4

```
pi@raspberrypi: ~
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 72 bytes 27368 (26.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 72 bytes 27368 (26.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.5 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::a739:182c:a5b:182f prefixlen 64 scopeid 0x20<link>
<global>
ether b8:27:eb:8b:cb:48 txqueuelen 1000 (Ethernet)
RX packets 469809 bytes 22298074 (21.2 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 957732 bytes 1472552875 (1.3 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```

Рисунок 7.4 – Визначення IP-адреси Raspberry Pi.

Підключити модуль камери Raspberry Pi дуже просто. Після відключення Pi підключіть камеру до порту Pi CSI, як показано на наступному рисунку 7.5 Переконайтеся, що камера підключена в правильній орієнтації так, щоб сині літери на стрічці були спрямовані вгору.

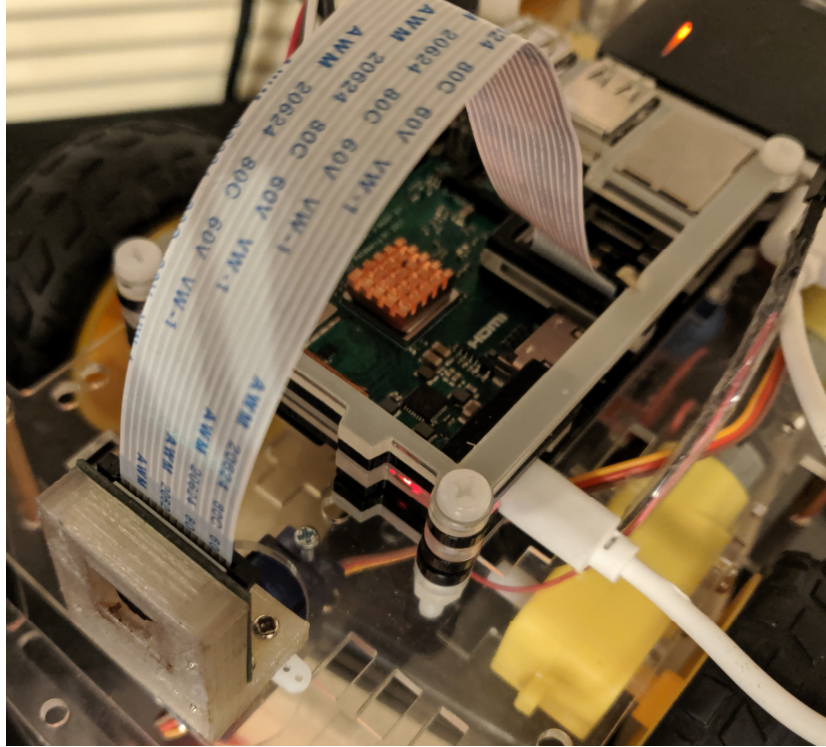


Рисунок 7.5 - Підключити модуль камери до порту CSI на Raspberry Pi

Скрипт для потокового відео буде в собі поєднувати HTML і Payten. На першому етапі створимо новий файл з ім'ям `rpi_camera_surveillance_system.py` з використанням редактора nano за допомогою наступної команди:

```
nano rpi_camera_surveillance_system.py
```

і ставимо в нього наступний код складається з 3 основних блоків.

**Блок 1:** Підключення бібліотек:

```
import io
import picamera
import logging
import socketserver
from threading import Condition
```

```
from http import server
```

**Блок 2:** Блок HTML коду:

```
PAGE=""
<html>
<head>
<title>Raspberry Pi - Surveillance Camera</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance Camera</h1></center>
<center></center>
</body>
</html>
```

**Блок 3:** Блок Python коду і робота з бібліотекою *picamera*:

```
class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()
    def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
```

```
def do_GET(self):
    if self.path == '/':
        self.send_response(301)
        self.send_header('Location', '/index.html')
        self.end_headers()
    elif self.path == '/index.html':
        content = PAGE.encode('utf-8')
        self.send_response(200)
        self.send_header('Content-Type', 'text/html')
        self.send_header('Content-Length', len(content))
        self.end_headers()
        self.wfile.write(content)
    elif self.path == '/stream.mjpg':
        self.send_response(200)
        self.send_header('Age', 0)
        self.send_header('Cache-Control', 'no-cache, private')
        self.send_header('Pragma', 'no-cache')
        self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
        self.end_headers()
        try:
            while True:
                with output.condition:
                    output.condition.wait()
                    frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
```

```

        self.wfile.write(b'\r\n')
    except Exception as e:
        logging.warning(
            'Removed streaming client %s: %s',
            self.client_address, str(e))
    else:
        self.send_error(404)
        self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()

```

Щоб зберегти файл, натисніть Ctrl + X, введіть Y і Enter.

Після написання скрипта можна запустити його за допомогою Python 3 використовуючи наступну команду:

```
python3 rpi_camera_surveillance_system.py
```

Після запуску сценарія ви можете отримати доступ до веб-сервера потокового відео по адресі:

*http: // <Your\_Pi\_IP\_Address>:8000.*

Замініть своїм IP-адресою Raspberry Pi, в даному випадку `http://192.168.1.5:8000`. Ви можете отримати доступ до потокового відео через будь-який пристрій, яке має браузер і підключено до тієї ж мережі, що і ваш RPі.

### **7.3. Розробка алгоритмів керування мобільного робота**

Перед розробкою програми необхідно розробити алгоритм керування мобільного робота. Але перед самою розробкою алгоритму, необхідно визначити систему команд, які повинен обробляти та виконувати MR. Як вже було визначено у постановці задачі, робот повинен рухатися вперед, назад, ліворуч, праворуч, а також робити зупинку двигунів. Отже, виходячи з цього, система команда наступна:

- forward для руху вперед;
- backward для руху назад;
- left для руху ліворуч;
- right для руху праворуч;
- stop для зупинки двигунів.

Розробивши систему команд, перейдемо до розробки алгоритмів програми MR. Перший алгоритм, який необхідно розробити та представити, це алгоритм запуску усіх необхідних процесів для можливості керування мобільним роботом. На рисунку 7.6 приведена схема алгоритму завантаження процесів для керування. Цей алгоритм є лінійний без жодного розгалуження. На ньому лише представлено, у якій послідовності буде завантажено усі необхідні програмні компоненти.

Наступним кроком, необхідно розробити схему алгоритму програми керування. Для більшої зручності, одну велику схему було розбито на декілька маленьких, а саме узагальнену схему, схему обробки даних, та схему виконання дій у випадку збігу команд з необхідними. На рисунку 7.7, приведено узагальнену схему алгоритму серверної програми керування на

MP. Нескінченний цикл необхідний для постійного прийому даних від клієнту та їх постійної обробки.



Рисунок 7.6 – Схема алгоритму завантаження процесів для керування



Рисунку 7.7 – Узагальнена схема алгоритму серверної програми керування на MR

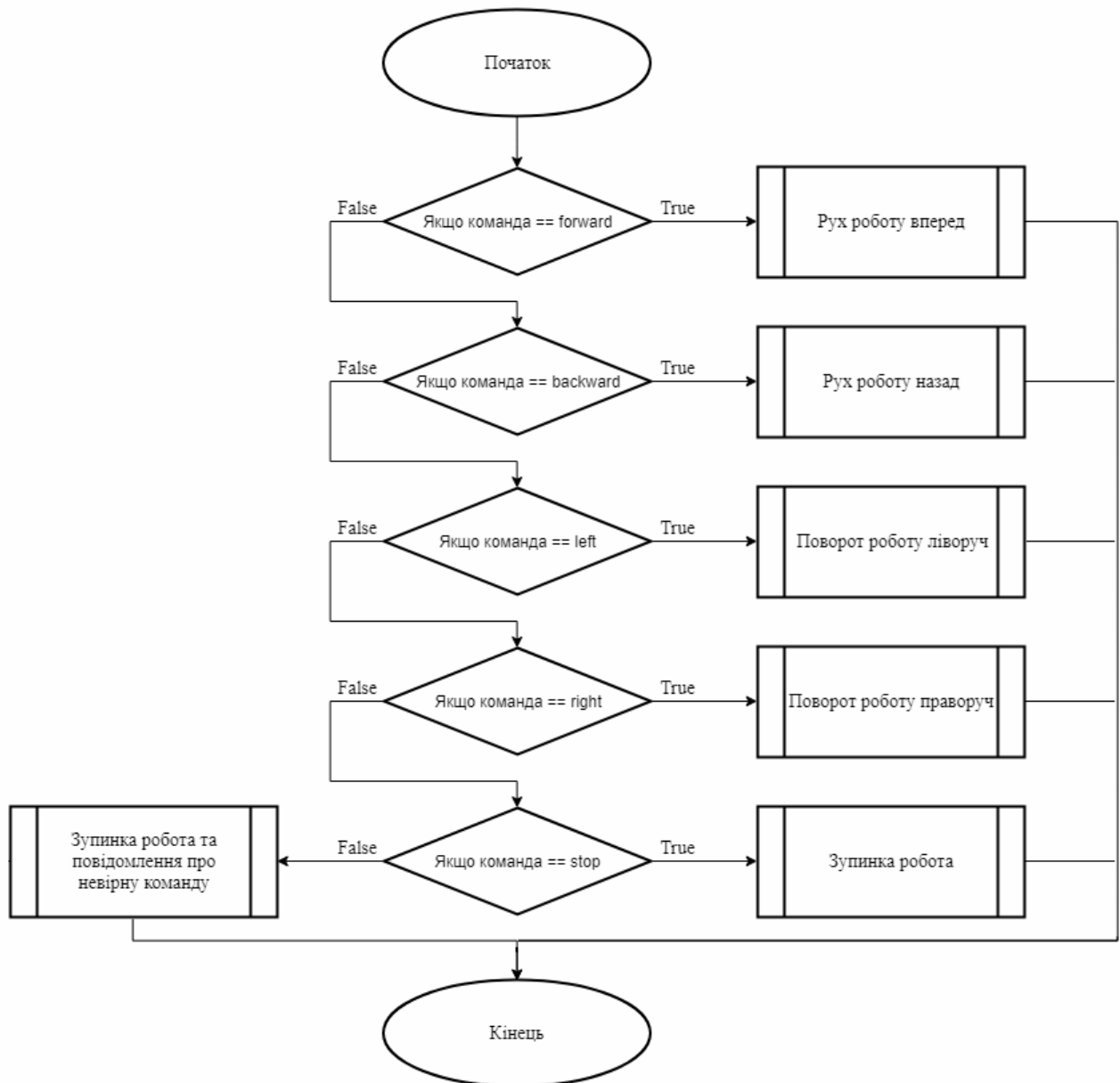


Рисунок 7.8 – Схема алгоритму прийому команд та їх обробки

Далі, слід розробити схему алгоритму обробки самих даних. Для реалізації цієї ділянки програми буде використано повне розгалуження. Схему приведено на рисунку 7.8. Також, в алгоритмі враховано випадок, якщо команда невідома.

Розробивши схеми великих ділянок програми, приступимо до маленьких, але не за своїм значенням, ділянок програми, які будуть виконувати дії на самому роботі. На рисунку 7.9 приведено схему алгоритму ділянки програми, у якій виконується рух роботу вперед.

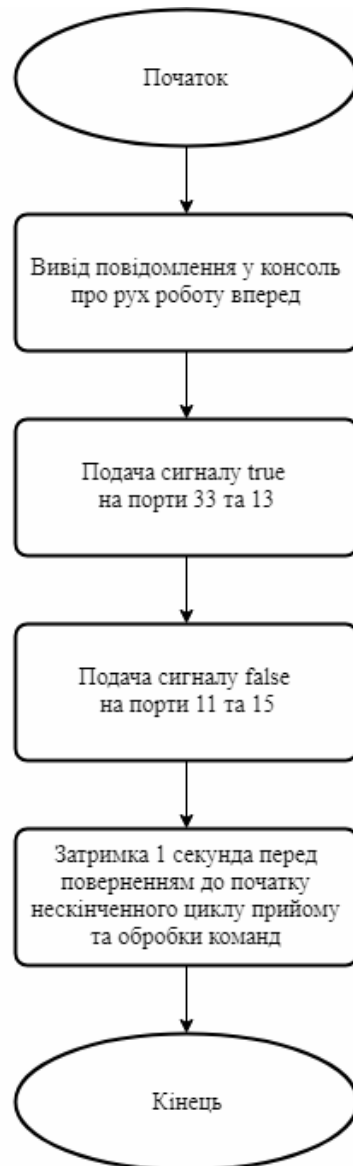


Рисунок 7.9 – Схема алгоритму підпрограми руху робота вперед

Як можна побачити, керування виконується завдяки подачі сигналів з плати керування, а саме міні комп'ютера RPі на драйвер двигунів. Таким чином, передається інструкція до цього самого драйверу, щодо полярності двигунів. Таким чином можна задати напрямок руху МР. На рисунку 7.10 приведена схема алгоритму руху МР назад.

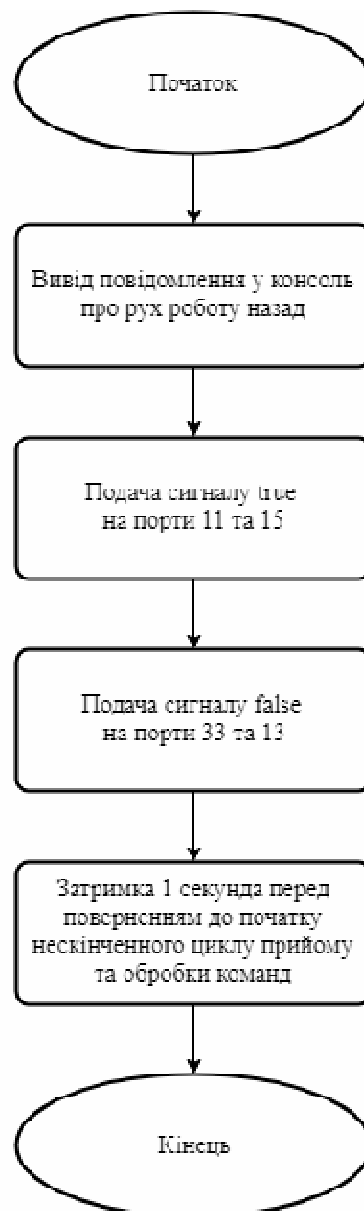


Рисунок 7.10 – Схема алгоритму підпрограми руху робота назад

Як видно зі схеми алгоритму, полярність двигунів змінюється, таким чином і змінюється напрямок руху МР на протилежний, на відміну від алгоритму, коли МР рухається вперед. На рисунку 7.11 приведена схема алгоритму повороту МР ліворуч.

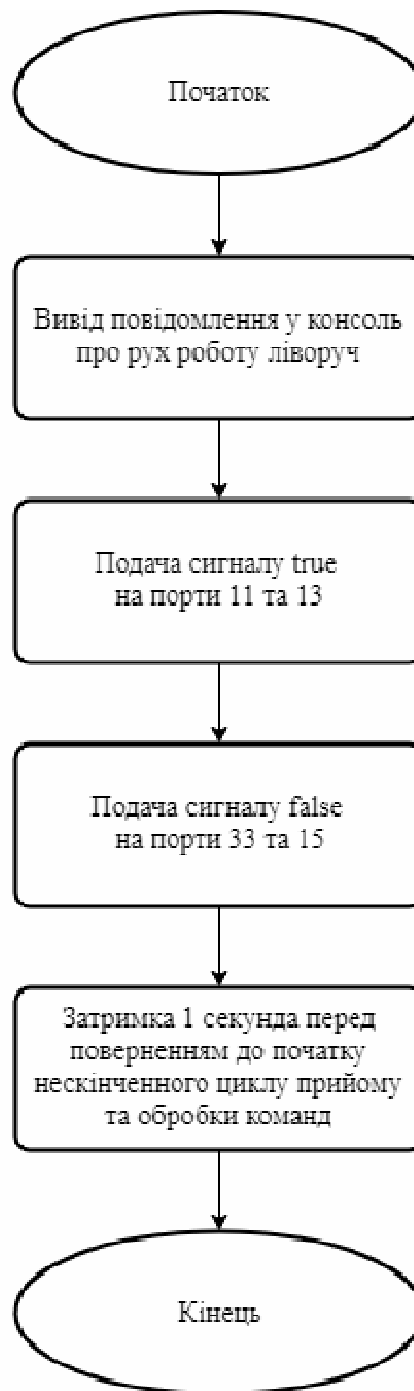


Рисунок 7.11 – Схема алгоритму підпрограми повороту робота ліворуч

Алгоритм руху МР ліворуч дозволяє робити поворот МР на місці. Це є дуже точний поворот, тому що робот виконує поворот без відхилень. Для цього встановлюється різна полярність на два двигуни, що є досить ефективним. На рисунку 7.12 приведено схему алгоритму повороту МР праворуч.

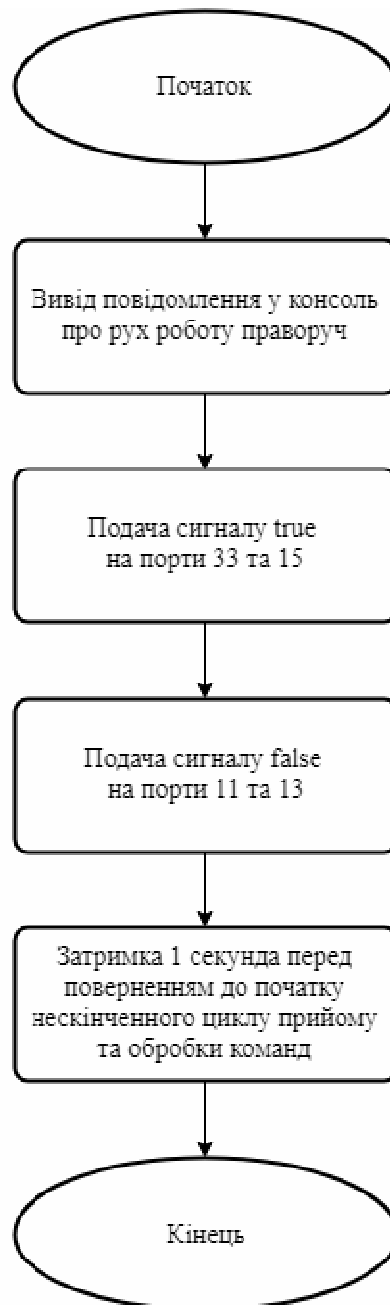


Рисунок 7.12 – Схема алгоритму підпрограми повороту робота праворуч

Також, як і при руху ліворуч, поворот здійснюється на місці, але полярність двигунів протилежна, на відміну від повороту ліворуч. Також, у кожній ділянці програми, яка виконує дії з роботом присутня затримка 1 секунду, перед обробкою наступної команди. Це зроблено для запобігання утворення перешкод під час прийому великої кількості команд. На рисунку 7.17 приведено схему алгоритму зупинки роботи.

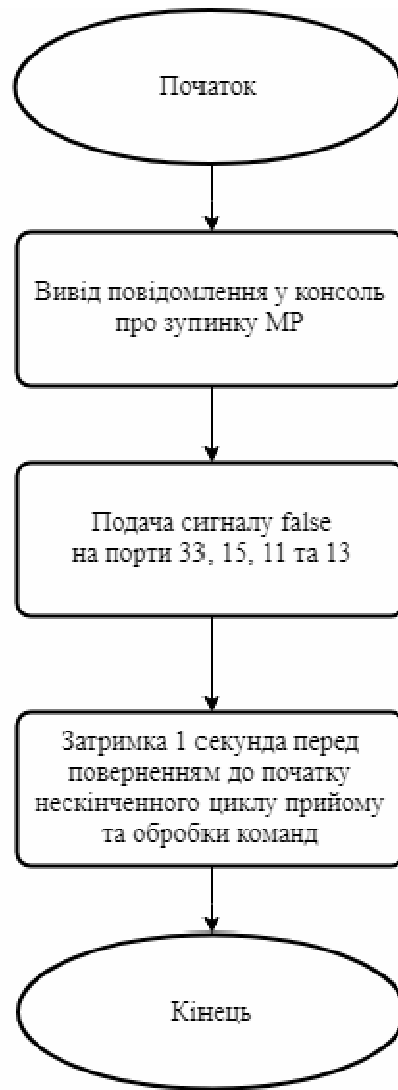


Рисунок 7.17 – Схема алгоритму підпрограми зупинки робота

Для зупинки робота, керуючий комп'ютер подає команду на драйвер двигунів вимкнути напругу. Таким чином, без напруги, робот негайно зупиняється. На рисунку 7.18, приведено схему алгоритму обробки невідомої команди.

Під час прийому невідомої команди роботом, комп'ютер одразу дає команду на драйвер відключити напругу на драйвер, та виводить у термінал повідомлення про помилку.

Розробивши схему алгоритму програми робота, також необхідно розробити схему алгоритму роботи програми, яка передає команди на МР.

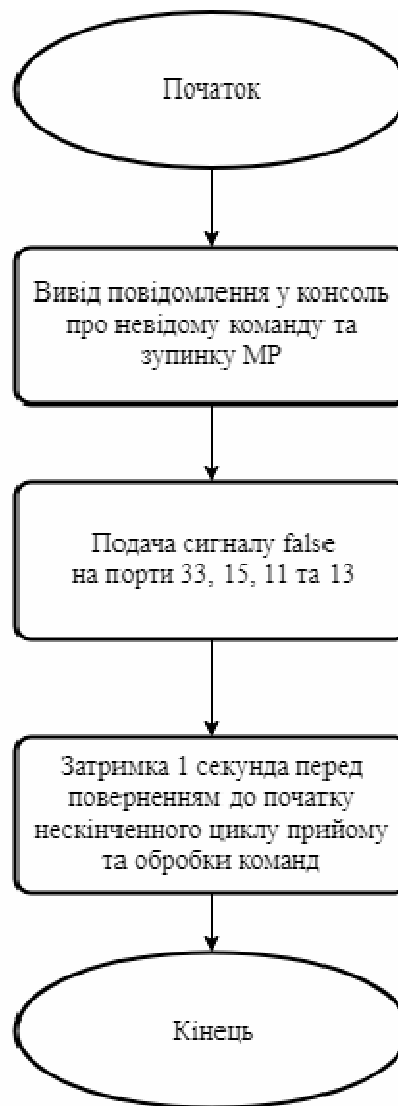


Рисунок 7.18 – Схема алгоритму підпрограми обробки невідомої команди

Програма повинна робити з'єднання з сокетом на сервері MP. Данні підключення потрібно вводити у форму, та після цього натиснути кнопку з'єднання. У випадку невдалого з'єднання, необхідно відправити повідомлення про помилку. В іншому випадку, програма починає транслювати з веб сервера відео зображення з камери MP за протоколом HyperText Transfer Protocol (HTTP). В інтерфейсі програми потрібно створити кнопки, які будуть відправляти необхідні команди на MP, а саме: Forward; Backward; Left; Right; Stop.

## 7.4 Розробка програми удаленого управління мобільним роботом

Розробка програми робота є важливою частиною, тому що вона повинна приймати данні з клієнтської програми та обробляти їх. В якості мови програмування, буде використано мову Python 3, тому що вона дуже поширена у робототехніці, а тому знайти інформацію щодо використання потрібних функції не є складною задачею. Спочатку необхідно підключити бібліотеки, функції з яких будуть використані у програмі:

```
import RPi.GPIO as GPIO  
import socket  
import time
```

Бібліотека RPi.GPIO використовується для керування портами керуючого міні комп'ютера для його зв'язку з драйвером двигунів. Бібліотека socket використовується для програмного створення каналу зв'язку між програмою-клієнтом та програмою-сервером. Також бібліотека time використовується для створення затримки між операціями.

Наступним кроком необхідно налаштуємо режими роботи портів наступним кодом:

```
GPIO.setwarnings(False)  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(33, GPIO.OUT)  
GPIO.setup(11, GPIO.OUT)  
GPIO.setup(13, GPIO.OUT)  
GPIO.setup(15, GPIO.OUT)  
GPIO.setup(29, GPIO.OUT)  
GPIO.setup(31, GPIO.OUT)  
GPIO.output(29, True)  
GPIO.output(31, True)
```

Таким чином було налаштовано режим звернення до портів, та їх

режими роботи. Наступним кроком необхідно відкрити сокет для обміну даними.

```
UDP_IP = "0.0.0.0"
UDP_PORT = 5050
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
```

Далі ініціалізуємо нескінченний цикл прийому та обробки даних. Дані на робота приходять у вигляді кортежу, тому необхідно записати команди, які потрібно обробляти також у вигляді кортежу, через неможливість порівняння строки із кортежем:

```
while True:
    raw=sock.recvfrom(1024)
    rawnew = ('forward', 'stop', 'backward', 'left', 'right')
```

Наступним кроком необхідно описати обробку прийнятого сигналу. В якості прикладу приведено умову, якщо прийнята команда на пересування роботу вперед. Якщо прийнята саме ця команда, робот рухається вперед, та робить затримку на 1 секунду перед обробкою наступної команди. Також у консоль виводиться налагоджувальна інформація у вигляді повідомлення, яка команда виконується:

```
if raw[0] == rawnew[0]:
    print 'robot moves forward'
    GPIO.output(33, True)
    GPIO.output(11, False)
    GPIO.output(13, True)
    GPIO.output(15, False)
    time.sleep(1)
```

Також, може статися, що робот прийняв невідому команду, тому слід обробити і її. Якщо робот прийняв команду, яка не задекларована у програмі, програма виведе повідомлення про це:

```
else:
```

```

print "message error"
GPIO.output(33, False)
GPIO.output(11, False)
GPIO.output(13, False)
GPIO.output(15, False)
time.sleep(1)

```

## 7.5 Розробка клієнтської програми удаленого управління мобільним роботом

Розробивши програму роботу, яка приймає команди, також необхідно розробити програму, яка буде відправляти ці команди. В якості мови програмування було обрано мову C# [17] та .NET Framework 4.5 [18], тому що цей фреймворк та мова програмування дуже зручні, мають необхідні інструменти для роботи з віконними додатками. Також при розробці додатків на цій мові програмування дуже легко масштабувати розробляємий додаток. І ще, ця мова програмування має досить велике ком'юніті та авторитетного розробника мови, а саме, корпорацію Microsoft, яка у наш час дуже відома своїми продуктами, у тому числі Microsoft windows, Visual studio, Visual studio code та інші.

Почнемо розробку зі створення з'єднання по сокету з MP:

```

public Presenter(Form1 form, int port, string host){
    this.form = form;
    this.socket_port = port;
    socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
    ProtocolType.Udp);
    try{
        socket.Connect(host, port);
        this.form.ShowMessage("Connecting");
    }
}

```

```

catch(Exception ex){
    this.form.ShowMessage(ex.ToString());
}
}

```

В якості адреси з'єднання, програма бере дані з форми, у яку користувач повинен ввести адресу та порт підключення. Далі, розробимо ділянку програми, яка оброблює подію натискання кнопки з'єднання. Після успішного з'єднання програми з МР, додаток одразу починає трансляцію з камери роботу методом створення простого браузеру та об'єкту мови розмітки HyperText Markup Language (HTML) 5, `<iframe>` [19], який створено саме для трансляції зі сторонніх веб-документів.

```

private void bn_appy_configs_Click(object sender, EventArgs e){
    if (tb_port.Text.Length == 4 && tb_host.Text.Length > 5) {
        presenter = new Presenter.Presenter(this,
            Convert.ToInt32(tb_port.Text), tb_host.Text);
        var page = @"
<iframe src='http://'+ tb_host.Text + "
width='100%' height='100%'> </iframe > ";
        webBrowser.DocumentText = page;
        webBrowser.ScriptErrorsSuppressed = true;
        MessageBox.Show("Configs has been saved");
    }else{
        MessageBox.Show("Uncorrect value");
    }
}
}

```

Також, слід розробити у додатку обробник кнопок відправки команд на МР:

```

private void bn_KeyPress(object sender, KeyPressEventArgs e) {
    switch (((Button)sender).Name) {
        case "bn_command_1":
            presenter.Send_command("value",

```

```

        Commands.FORWARD);
        break;
    case "bn_command_2":
        presenter.Send_command("value",
            Commands.BACKWARD);
        break;
    case "bn_command_3":
        presenter.Send_command("value", Commands.LEFT);
        break;
    case "bn_command_4":
        presenter.Send_command("value", Commands.RIGHT);
        break;
    case "bn_command_5":
        presenter.Send_command("value", Commands.STOP);
        break;
    default:
        break;
}

```

Після створення обробнику, також необхідно описати інтерфейс, який буде відправляти команди, в залежності від натиснутої кнопки на MP:

```

public void Send_command(string value, Commands command) {
    if (socket != null){
        byte[] send_buf = new byte[200];
        switch (command){
            case Commands.FORWARD:
                send_buf = Encoding.ASCII.GetBytes("forward");
                break;
            case Commands.BACKWARD:
                send_buf =
                Encoding.ASCII.GetBytes("backward");

```

```
        break;
    case Commands.LEFT:
        send_buf = Encoding.ASCII.GetBytes("left");
        break;
    case Commands.RIGHT:
        send_buf = Encoding.ASCII.GetBytes("right");
        break;
    case Commands.STOP:
        send_buf = Encoding.ASCII.GetBytes("stop");
        break;
    default:
        break;
}
socket.Send(send_buf);}}
```

## 7.6 Встановлення необхідних пакетів та налаштувань

Встановлення, та налаштування компонентів системи, є важливою частиною, щоб система керування працювала. Перше, що необхідно зробити, це встановити операційну систему, на базі якої буде працювати серверна частина MP. В якості операційної системи, потрібно встановити ОС Linux Raspbian 10 buster lite, яка є безкоштовною. Наступним кроком, за допомогою програми Win32 Disk Imager потрібно розгорнути завантажений образ на флешці microSD. Вікно програми з усіма налаштуваннями приведено на рисунку 7.19.

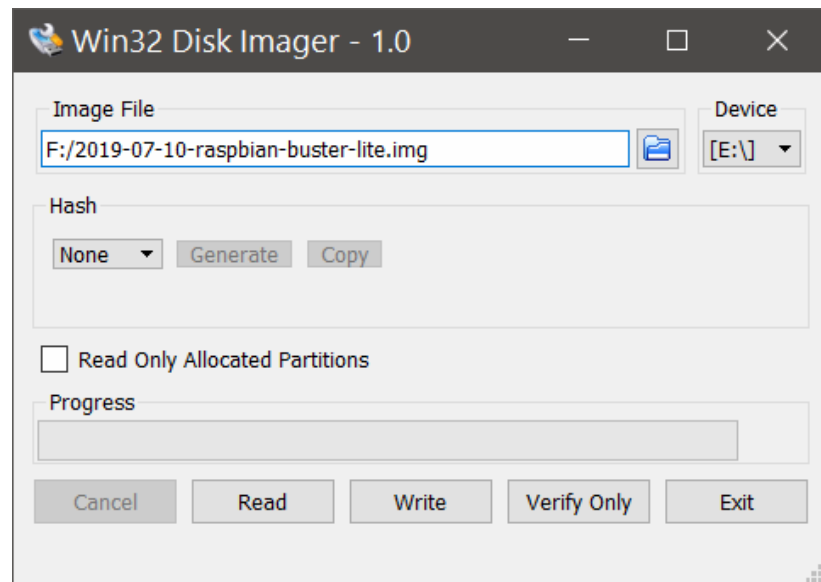


Рисунок 7.19 – Вікно програми Win32 Disk Imager

Після цього, необхідно встановити карту пам'яті у міні комп'ютер, подати напругу та з'єднатися з ним за протоколом SSH. Стандартно, з'єднатися можна за адресом raspberrypi. Логін pi, пароль доступу raspberry. Перша команда, яку необхідно виконати, це `sudo su`, щоб отримати усі привілеї для виконання наступних операцій.

Наступним кроком, необхідно запуснути утиліту `raspi-config`, обрати у розділі `Interfacing options` пункт `Camera`, та включити інтерфейс. Вікно утиліти `raspi-config` приведено на рисунку 7.20.

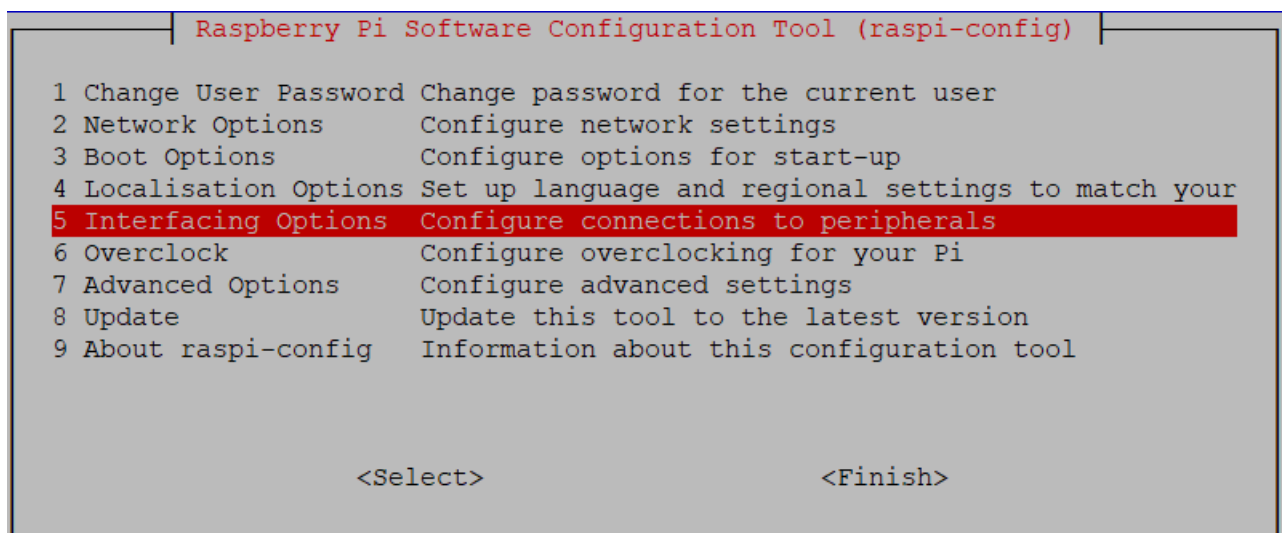


Рисунок 7.20 – Вікно утиліти `raspi-config`

Далі, слід оновити список джерел для встановлення необхідних пакетів, виконавши команду `apt-get update`. Після цього, встановимо усі необхідні пакети командою:

```
apt-get install hostapd dnsmasq git python-pip
```

За допомогою вже встановленої утиліти `pip`, встановимо необхідні бібліотеки `python` для роботи програми роботу:

```
pip install sockets times RPi.GPIO
```

Для встановлення утиліти трансляції відео, встановимо пакет з репозиторія за допомогою утиліти `git` [20]:

```
git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface.git
```

Далі, слід перейти до директорії `RPi_Cam_Web_Interface` за допомогою команди `cd` та встановити утиліту:

```
cd /RPi_Cam_Web_Interface  
./install.sh
```

Після установки пакету, трансляція відео відбувається на веб сервері, та стартує автоматично, коли завантажується ОС.

Наступним кроком, необхідно встановити програму керування. Для облегшення установки, програму завантажено на видалений репозиторій, тому встановлюється за допомогою утиліти `git`:

```
git clone https://github.com/aleksey-smx/robot_server_socket.git
```

Також, для можливості виконання програми, необхідно надати права на запуск командою:

```
chmod +x /home/pi/robot_server_socket/robot.py
```

Все необхідне зроблено, тепер потрібно зробити автоматичне виконання програми, під час завантаження ОС. Для цього необхідно додати у файл `/etc/rc.local` наступну команду:

```
python /home/pi/robot_server_socket/robot.py
```

Останнє, що необхідно зробити, це налаштувати мережу `wifi`, до якої повинен підключатись керуючий оператор. Для цього, перо налаштуванням, необхідно відключити пакети [21]:

```
systemctl stop dnsmasq
```

```
systemctl stop hostapd
```

Наступним кроком, необхідно додати у файл `/etc/dhcpd.conf`:

```
interface wlan0
```

```
static ip_address=192.168.7.1/24
```

```
nohook wpa_supplicant
```

Далі слід перезапустити пакет `dhcpd`:

```
service dhcpd restart
```

Також потрібно відредагувати файл `/etc/dnsmasq.conf`, де задамо діапазон адрес клієнтів [22]:

```
interface=wlan0
```

```
dhcp-range=192.168.7.2,192.168.8.20,255.255.255.0,24h
```

Налаштуємо параметри бездротової мережі у файлі `/etc/hostapd/hostapd.conf`:

```
interface=wlan0
```

```
driver=nl80211
```

```
ssid=Назва мережи
```

```
hw_mode=g
```

```
channel=7
```

```
wmm_enabled=0
```

```
macaddr_acl=0
```

```
auth_algs=1
```

```
ignore_broadcast_ssid=0
```

```
wpa=2
```

```
wpa_passphrase=пароль
```

```
wpa_key_mgmt=WPA-PSK
```

```
wpa_pairwise=TKIP
```

```
rsn_pairwise=CCMP
```

У файлі `/etc/default/hostapd` додаємо завантаження конфігурації бездротової мережі:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Запустимо пакет `hostapd`:

```
systemctl unmask hostapd
```

```
systemctl enable hostapd
```

```
systemctl start hostapd
```

Додамо у файл `/etc/sysctl.conf` наступний параметр:

```
net.ipv4.ip_forward=1
```

А також створимо правило для фаєрвола та збережемо його у пам'яті [23]:

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Також, додаймо у файл `/etc/rc.local` автоматичне встановлення правил при перезавантаженні:

```
iptables-restore < /etc/iptables.ipv4.nat
```

На цьому, встановлення та налаштування програмного забезпечення завершується. Достатньо перезавантажити міні комп'ютер, щоб усі параметри вступили в силу.

## 7.7 Контрольні питання

1. Що означає рядок `GPIO.setup` в скрипті управління DC двигунами ?;
2. Як в `PyuTen` прописується рівень сигналу на пінє шини `GPIO` ?;
3. Принцип роботи сигналу ШІМ на `Raspberry Pi` ?;
4. Перерахуйте набір бібліотек для реалізацій відео стріму на мові `PyuTen`;
5. За допомогою якої команди на мові `PyuTen` можна перевернути зображення отриманої з камери ?;

6. В якому форматі траслірується відео стрім з використанням бібліотеці PiCamera ?;
7. Чим може бути викликані артефакти і затримка при трансляції відео стріму ?;
8. За допомогою якого протоколу транслюєте відео стрім з камери MP ?;
9. Як можна організувати структуру управління MP через інтерфейс користувача ?.

**ПЕРЕЛІК ПОСИЛАНЬ**

1. Klaus-Dieter Thoben; Stefan Wiesner; Thorsten Wuest "Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples// IJAT (International Journal of Automation Technology): IJAT 11, 2017 pp. 4–16. (DOI: [doi.org/10.20965/ijat.2017.p0004](https://doi.org/10.20965/ijat.2017.p0004))
2. Giovanna Culot, Guido Nassimbeni, Guido Orzes, Marco Sartora Behind the definition of Industry 4.0: Analysis and open questions // International Journal of Production Economics Available online 10 January 2020 (<https://doi.org/10.1016/j.ijpe.2020.107617>)
3. Tommaso Dreossi, Alexandre Donzé, Sanjit A. Seshia Compositional Falsification of Cyber-Physical Systems with Machine Learning Components // Systems with Machine Learning Components. J Autom Reasoning 63, 1031–1053 (2019) (doi:10.1007/s10817-018-09509-5)
4. Al-Emran M., Malik S.I., Al-Kabi M.N. (2020) A Survey of Internet of Things (IoT) in Education: Opportunities and Challenges. In: Hassanien A., Bhatnagar R., Khalifa N., Taha M. (eds) Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications. Studies in Computational Intelligence, vol 846. Springer, Cham ([https://doi.org/10.1007/978-3-030-24513-9\\_12](https://doi.org/10.1007/978-3-030-24513-9_12))
5. Paul Brous, Marijn Janssen, Paulien Herder The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations // International Journal of Information Management Volume 51, April 2020, 101952 (<https://doi.org/10.1016/j.ijinfomgt.2019.05.008>)
6. Dey A.J., Sarma H.K.D. (2020) Routing Techniques in Internet of Things: A Review. In: Sarma H., Bhuyan B., Borah S., Dutta N. (eds) Trends in Communication, Cloud, and Big Data. Lecture Notes in Networks and Systems, vol 99. pp 41-50 ([https://doi.org/10.1007/978-981-15-1624-5\\_5](https://doi.org/10.1007/978-981-15-1624-5_5))

7. Bajaj RK, Rao M, Agrawal H (2018) Internet of things (IoT) in the smart automotive sector: a review. In: IOSR Journal of Computer Engineering (IOSR-JCE), conference on recent trends in computer engineering (CRTCE), pp 36–44
8. Durga, S.; Nag, R.; Daniel, E. Survey on machine learning and deep learning algorithms used in internet of things (IoT) healthcare. In Proceedings of the 3rd International Conference on Computing Methodologies and Communication—ICCMC, Erode, India, 27–29 March 2019; pp. 1018–1022.
9. hosal, P.; Das, D.; Das, I. Extensive survey on cloud-based IoT-healthcare and security using machine learning. In Proceedings of the 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks—ICRCICN, Kolkata, India, 22–23 November 2018; pp. 1–5.
10. Wang, C.; Redmond, S.J.; Lu, W.; Stevens, M.C.; Lord, S.R.; Lovell, N.H. Selecting Power-Efficient Signal Features for a Low-Power Fall Detector. *IEEE Trans. Biomed. Eng.* 2017, 64, 2729–2736.
11. Anagnostopoulos, T.; Ntalianis, K.; Skourlas, C.; Jino Ramson, S.R. IoT-enabled fall verification of elderly and impaired people in Smart Cities. In Proceedings of the Pan-Hellenic Conference on Informatics, Athens, Greece, 29 November–1 December 2018; pp. 88–92.
12. Ghasemi, F.; Rezaee, A.; Rahmani, A.M. Structural and behavioral reference model for IoT-based elderly health-care systems in smart home. *Int. J. Commun. Syst.* 2019, 32, 4002.
13. Balakrishna, S.; Thirumaran, M.; Solanki, V.K. IoT sensor data integration in healthcare using semantics and machine learning approaches. *Intell. Syst. Ref. Libr.* 2020, 165, 275–300.
14. Takahashi, Y.; Nishida, Y.; Kitamura, K.; Mizoguchi, H. Handrail IoT sensor for precision healthcare of elderly people in smart homes. In Proceedings of the IEEE 5th International Symposium on Robotics and Intelligent Sensors, Ottawa, ON, Canada, 5–7 October 2017; pp. 364–368.

15. Klaib, A.F.; Alsrehin, N.O.; Melhem, W.Y.; Bashtawi, H.O. IoT smart home using eye tracking and voice interfaces for elderly and special needs people. *J. Commun.* 2019, 14, 614–621.
16. B. Chatterjee, N. Cao, A. Raychowdhury, S. Sen, "Context-aware intelligence in resource-constrained IoT nodes: Opportunities and challenges", *IEEE Design Test*, vol. 36, no. 2, pp. 7-40, Apr. 2019.
17. M. Zorn, S. Pelc, M. Koderman, "Natural disasters and less developed countries" in *Nature Tourism and Ethnicity as Drivers of (De)Marginalization*, Cham, Switzerland:Springer, vol. 3, 2018.
18. O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, M. H. D. N. Hindia, "An overview of Internet of things (IoT) and data analytics in agriculture: Benefits and challenges", *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758-3773, Oct. 2018.
19. S. Rimer, "An IoT architecture for financial services in developing countries", *Proc. IST-Africa Week Conf. (IST-Africa)*, pp. 1-10, May 2017
20. A. T. Abagissa, A. Behura, S. K. Pani, "IoT based smart agricultural device controlling system", *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, pp. 26-30, Apr. 2018
21. R. S. Cabrera, A. P. de la Cruz, "Public transport vehicle tracking service for intermediate cities of developing countries based on ITS architecture using Internet of things (IoT)", *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, pp. 2784-2789, Nov. 2018.
22. J. Fell, J. Pead, K. Winter, "Low-cost flow sensors: Making smart water monitoring technology affordable", *IEEE Consum. Electron. Mag.*, vol. 8, no. 1, pp. 72-77, Jan. 2019
23. R. I. S. Pereira, I. M. Dupont, P. C. M. Carvalho, S. C. S. Jucá, "IoT embedded Linux system based on raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant", *Measurement*, vol. 114, pp. 286-297, Jan. 2018.
24. C. Dupont, T. Bures, M. Sheikhalishahi, C. Pham, A. Rahim, "Low-cost IoT big data and cloud platform for developing countries" in *BT—Economics of*

Grids Clouds Systems and Services, New York, NY, USA:UNCDF, pp. 285-299, 2017.

25. A. Benešová, J. Tupa Requirements for education and qualification of people in industry 4.0 //Procedia Manufacturing, 11 (2017), pp. 2195-2202

26. D. Gorecky, K. Mura, F. Arlt A vision on training and knowledge sharing applications in future factories // IFAC Proceedings, 46 (14) (2013), pp. 90-97

27. Kamlesh Tiwari, Mohammad Shadab Khan Sustainability accounting and reporting in the industry 4.0 // Journal of Cleaner Production Volume 258, 10 June 2020, 120783 (<https://doi.org/10.1016/j.jclepro.2020.120783>)

28. Shohin Aheleroff, Xun Xua, Yuqian Lu, Mauricio Aristizabal, Juan Pablo Velásquez, Benjamin Joab, Yesid Valenciab IoT-enabled smart appliances under industry 4.0: A case study // Advanced Engineering Informatics Volume 43, January 2020, 101043(<https://doi.org/10.1016/j.aei.2020.101043>)

29. D. Arunachalam, N. Kumar, J.P. Kawalek Understanding big data analytics capabilities in supply chain management: Unravelling the issues, challenges and implications for practice //Transportation Research Part E: Logistics and Transportation Review, 114 (6) (2018), pp. 416-436, (10.1016/j.tre.2017.04.001)

30. Requirements for education and qualification of people in industry 4.0// Procedia Manufacturing, 11 (2017), pp. 2195-2202, (10.1016/j.promfg.2017.07.366)

31. R. Boselli, M. Cesarini, F. Mercorio, M. Mezzanzanica Classifying online job advertisements through machine learning// Future Generation Computer Systems, 86 (9) (2018), pp. 319-328, (10.1016/j.future.2018.03.035)

32. M. Elhoseny, A. Abdelaziz, A.S. Salama, A.M. Riad, K. Muhammad, A.K. Sangaiah A hybrid model of internet of things and cloud computing to manage big data in health services applications// Future generation computer systems, 86 (2018), pp. 1383-1394, (10.1016/j.future.2018.03.005)

33. F. Imran, J. Kantola Review of industry 4.0 in the light of sociotechnical system theory and competence-based view: A future research agenda for the evolutive approach // J. Kantola, S. Nazir, T. Barath (Eds.), International Conference on applied human factors and ergonomics (118-128). Advances in human factors, business management and society, AHFE 2018. Advances in intelligent systems and computing, 783, Springer, Cham, CH (2018), (10.1007/978-3-319-94709-9\_12)
34. Š. Kozák, E. Ružický, J. Štefanovič, F. Schindler Research and education for industry 4.0: Present development // Proceedings of the 29th International Conference 2018 Cybernetics & Informatics (K&I) (2018), pp. 1-8
35. P. Macurova, L. Ludvik, M. Žwakova The driving factors, risks and barriers of the industry 4.0 concept // Journal of Applied Economic Sciences, 12 (7) (2017), pp. 2003-2011
36. A.Y. Alqahtani, S.M. Gupta, K. Nakashima Warranty and maintenance analysis of sensor embedded products using internet of things in industry 4.0// Int. J. Prod. Econ., 208 (2019), pp. 483-499
37. H. Fatorachian, H. Kazemi A critical investigation of Industry 4.0 in manufacturing: theoretical operationalisation framework // Prod. Plann. Contr., 29 (8) (2018), pp. 633-644
38. A. Frank, L. Dalenogare, N. Ayala Industry 4.0 technologies: implementation patterns in manufacturing companies // Int. J. Prod. Econ., 210 (2019), pp. 15-26
39. Y. Liao, F. Deschamps, E. Loures, L. Ramos Past, present and future of Industry 4.0-a systematic literature review and research agenda proposal //Int. J. Prod. Res., 55 (12) (2017), pp. 3609-3629
40. Y. Lu Industry 4.0: a survey on technologies, applications and open research issues // Journal of Industrial Information Integration, 6 (2017), pp. 1-10
41. Reza Arghandeha, Alexandravon Meier, Laura Mehrmanesh, LamineMilic On the definition of cyber-physical resilience in power systems //

Renewable and Sustainable Energy Reviews Volume 58, May 2016, Pages 1060-1069 (<https://doi.org/10.1016/j.rser.2015.12.193>)

42. Tommaso Dreossi, Alexandre Donzé, Sanjit A. Seshia Compositional Falsification of Cyber-Physical Systems with Machine Learning Components // Systems with Machine Learning Components. J Autom Reasoning 63, 1031–1053 (2019) (doi:10.1007/s10817-018-09509-5)

43. Zhipeng Cai ; Xu Zheng A private and efficient mechanism for data uploading in smart cyber-physical systems// IEEE Transactions on Network Science and Engineering ( Early Access ) 2018, Page(s): 1 – 1 (DOI: 10.1109/TNSE.2018.2830307)

44. Sascha Julian Oks, Albrecht FritzscheKathrin M. Möslin An application map for industrial cyber-physical systems // Industrial Internet of Things 2016 – Springer, pp 21-46 ([https://doi.org/10.1007/978-3-319-42559-7\\_2](https://doi.org/10.1007/978-3-319-42559-7_2))

45. O Penas, R Plateaux, S Patalano, M Hammadi Multi-scale approach from mechatronic to Cyber-Physical Systems for the design of manufacturing systems // Computers in Industry Volume 86, April 2017, Pages 52-69 (<https://doi.org/10.1016/j.compind.2016.12.001>)

46. F Yuan, SE OOI, LIM Yuto, TAN Yasuo Time task scheduling for simple and proximate time model in cyber-physical systems// Computational Science and Technology. Lecture Notes in Electrical Engineering, vol 481. Springer, Singapore (2019) pp 185-194 (DOI: [https://doi.org/10.1007/978-981-13-2622-6\\_19](https://doi.org/10.1007/978-981-13-2622-6_19))

47. Peter Gorm Larsen, John Fitzgerald,Jim Woodcock,René Nilsson,Carl Gamble, Simon Foster Towards semantically integrated models and tools for cyber-physical systems design// International Symposium on Leveraging Applications of Formal Methods ISoLA 2016: Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications, Lecture Notes in Computer Science, vol 9953. Springer, Cham, pp 171-186 (DOI: [https://doi.org/10.1007/978-3-319-47169-3\\_13](https://doi.org/10.1007/978-3-319-47169-3_13))

48. KR Varshney, H Alemzadeh On the safety of machine learning: Cyber-physical systems, decision sciences, and data products // Published Online:10 Sep 2019 (DOI: <https://doi.org/10.1089/big.2016.0051>)

49. ASUS TINKER BOARD [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/boards/482-asus-tinker-board.html>. (дата звернення: 23.03.2020)

50. Asus Tinker Board — как Raspberry Pi, только для искусственного интеллекта. [Електронний ресурс]. Режим доступу до інформації: <https://www.ixbt.com/news/2019/11/14/asus-tinker-board-raspberry-pi.html>. (дата звернення: 23.03.2020)

51. RASPBERRY PI 3 MODEL A+ [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/boards/809-raspberry-pi-3-model-a.html>. (дата звернення: 23.03.2020).

52. Одноплатный компьютер Raspberry Pi 3 Model A+ оценен в \$25 [Електронний ресурс]. Режим доступу до інформації: <https://megaobzor.com/Odnoplatnii-komputer-Raspberry-Pi-3-Model-A-ocenен-v-25.html>. (дата звернення: 23.03.2020).

53. RASPBERRY PI 3 MODEL B+ [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/boards/693-raspberry-pi-3-model-b.html>. (дата звернення: 23.03.2020).

54. Вышел обновлённый Raspberry Pi 3 Model B+ (URL: <https://habr.com/ru/post/357968/>) Published Online:29 Feb 2020.

55. [www.arduino.ua](http://www.arduino.ua) Raspberry Pi Model A+256Mb [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod3000-raspberry-pi-a-256mb>. (дата звернення: 23.03.2020).

56. Raspberry Pi Model A 256MB RAM\* [Електронний ресурс]. Режим доступу до інформації: <https://botland.com.pl/en/modules-and-kits-raspberry-pi-3-a-b/1287-raspberry-pi-model-a-256mb-ram.html>. (дата звернення: 23.03.2020).

57. Raspberry Pi Zero [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/products/raspberry-pi-zero/>. (дата звернення: 23.03.2020).

58. Raspberry Pi Zero [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/products/raspberry-pi-zero/>. (дата звернення: 23.03.2020).

59. Мини-комп'ютер Raspberry Pi Zero W (v.1.1) [Електронний ресурс]. Режим доступу до інформації: <https://raspberrypi.in.ua/product/mini-kompyuter-raspberry-pi-zero-w/>. (дата звернення: 23.03.2020).

60. Raspberry Pi Zero W [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>. (дата звернення: 23.03.2020).

61. What's Orange Pi Win? [Електронний ресурс]. Режим доступу до інформації: [http://www.orangepi.org/OrangePiWin\\_WinPlus/](http://www.orangepi.org/OrangePiWin_WinPlus/). (дата звернення: 23.03.2020).

62. Мини-комп'ютер Orange Pi Win Plus A64 Quad-core 2GB WIFI [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod2782-mini-komputer-orange-pi-win-plus-a64-quad-core-2gb-wifi>. (дата звернення: 23.03.2020).

63. What's Orange Pi Zero? [Електронний ресурс]. Режим доступу до інформації: <http://www.orangepi.org/orangepizero/>. (дата звернення: 23.03.2020).

64. Мини-комп'ютер Orange Pi Zero 256Мб [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod1894-mini-komputer-orange-pi-zero-256mb>. (дата звернення: 23.03.2020).

65. What's Orange Pi Pc Plus? [Електронний ресурс]. Режим доступу до інформації: <http://www.orangepi.org/orangepipcplus/>. (дата звернення: 23.03.2020).

66. Миникомпьютер Orange Pi PC Plus [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod1780-minikomputer-orange-pi-pc-plus>. (дата звернення: 23.03.2020)

67. LattePanda V1.0 - A Powerful Windows 10 Mini PC 2GB/32GB (Unactivated) [Електронний ресурс]. Режим доступу до інформації: <https://www.dfrobot.com/product-1498.html>. (дата звернення: 23.03.2020)

68. LattePanda 2G/32G [Електронний ресурс]. Режим доступу до інформації: <https://www.latterpanda.com/products/1.html>. (дата звернення: 23.03.2020).

69. ODROID-XU4Q Special Price [Електронний ресурс]. Режим доступу до інформації: <https://www.hardkernel.com/shop/odroid-xu4q-special-price/>. (дата звернення: 23.03.2020).

70. Odroid XU4Q [Електронний ресурс]. Режим доступу до інформації: [https://www.odroid.co.uk/index.php?route=product/product&product\\_id=813](https://www.odroid.co.uk/index.php?route=product/product&product_id=813). (дата звернення: 23.03.2020).

71. ODROID-XU4 [Електронний ресурс]. Режим доступу до інформації: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>. (дата звернення: 23.03.2020).

72. ODROID-XU4 [Електронний ресурс]. Режим доступу до інформації: <https://ameridroid.com/products/odroid-xu4>. (дата звернення: 23.03.2020)..

73. Мини-компьютер ODROID-НС2 Home Cloud Two с блоком питания 12В [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod2574-odroid-hc2-home-cloud-two>. (дата звернення: 23.03.2020).

74. ODROID-НС2 [Електронний ресурс]. Режим доступу до інформації: <https://ameridroid.com/products/odroid-hc2>. (дата звернення: 23.03.2020).

75. Мини-компьютер PINE A64+ Board 1Гб [Електронний ресурс]. Режим доступу до інформації: <http://arduino.ua/prod1668-mini-komputer-pine-a64-board-1gb>. (дата звернення: 23.03.2020).

76. PINE A64+ 1GB BOARD [Електронний ресурс]. Режим доступу до інформації: <https://store.pine64.org/?product=pine-a64-board-1gb>. (дата звернення: 23.03.2020).

77.. The Long Term Supply Version of the PINE A64 (+) [Електронний ресурс]. Режим доступу до інформації: <https://www.pine64.org/devices/single-board-computers/pine-a64-lts/>. (дата звернення: 23.03.2020).

78. PINE A64-LTS [Електронний ресурс]. Режим доступу до інформації: <https://www.pine64.org/devices/single-board-computers/pine-a64-lts/>. (дата звернення: 23.03.2020).

79. Arduino Expansion Shield for Raspberry Pi model B [Електронний ресурс]. Режим доступу до інформації: <https://www.dfrobot.com/product-1148.html>. (дата звернення: 23.03.2020).

80. I2C PCF8574 розширювач портів 8шт GPIO порту расширитель [Електронний ресурс]. Режим доступу до інформації: <http://ardu.net/uk/moduli/155-i2c-pcf8574-rozshiryuvach-portiv-8sht-gpio-portu-rasshiritel-139773705.html>. (дата звернення: 23.03.2020).

81. Модуль PoE питания POE\_BOARD для Raspberry Pi 3 B+ [Електронний ресурс]. Режим доступу до інформації: [https://arduino.ua/prod2846-modul-poe-pitaniya-poe\\_board-dlya-raspberry-pi-3-b](https://arduino.ua/prod2846-modul-poe-pitaniya-poe_board-dlya-raspberry-pi-3-b). (дата звернення: 23.03.2020).

82. LI-ION BATTERY HAT ДЛЯ RASPBERRY PI [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/platy-rasshireniya/790-li-ion-battery-hat-dlya-raspberry-pi.html>. (дата звернення: 23.03.2020).

83. Плата розширення для Raspberry Pi DVK512 [Електронний ресурс]. Режим доступу до інформації: <https://diylab.com.ua/p110170081-plata-rozshirennya-dlya.html>. (дата звернення: 23.03.2020).

84. Шилд 3-х реле для Raspberry Pi от Waveshare [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod2535-shild-3-h-rele-dlya-raspberry-pi>. (дата звернення: 23.03.2020).

85. Плата расширения моторшилд для Raspberry Pi [Электронный ресурс]. Режим доступа до інформації: <https://miniboard.com.ua/platy-rasshireniya/359-plata-rasshireniya-motorshild-dlya-raspberry-pi.html>. (дата звернення: 23.03.2020).

86. Шилд прецизионного АЦП/ЦАП для Raspberry Pi от Waveshare [Электронный ресурс]. Режим доступа до інформації: <https://arduino.ua/prod2534-raspberry-pi-high-precision-adda-expansion-board>. (дата звернення: 23.03.2020).

87. GSM/GPRS/GNSS/Bluetooth HAT for Raspberry Pi [Электронный ресурс]. Режим доступа до інформації: <https://www.waveshare.com/gsm-gprs-gnss-hat.htm>. (дата звернення: 23.03.2020).

88. Модуль RTC годинника реального часу DS3231 [Электронный ресурс]. Режим доступа до інформації: <https://diylab.com.ua/p706968182-modul-rtc-godinnika.html>. (дата звернення: 23.03.2020).

89. 3-осевой акселерометр GY-291 ADXL345 [Электронный ресурс]. Режим доступа до інформації: <https://arduino.ua/prod2516-gy-291-adxl345>. (дата звернення: 23.03.2020).

90. Инфракрасный датчик расстояния [Электронный ресурс]. Режим доступа до інформації: [http://home.roboticlab.eu/ru/examples/sensor/ir\\_distance](http://home.roboticlab.eu/ru/examples/sensor/ir_distance). (дата звернення: 23.03.2020).

91. Датчик расстояния инфракрасный SHARP GP2Y0A41SK0F 4-30см [Электронный ресурс]. Режим доступа до інформації: <https://www.minitech.com.ua/datchik-rasstoyaniya-sharp-gp2y0a41sk0f-4-30-sm>. (дата звернення: 23.03.2020).

92. Аналоговый датчик ультрафиолета на GUVVA-S12SD от Adafruit [Электронный ресурс]. Режим доступа до інформації: <https://arduino.ua/prod1644-analogovii-datchik-ultrafioleta-na-guvas12sd-ot-adafruit>. (дата звернення: 23.03.2020).

93. Водонепроницаемый ультразвуковой дальномер 30см - 3.5м [Электронный ресурс]. Режим доступа до інформації:

<https://arduino.ua/prod1243-vodonipronicaemii-yltrazvykovoii-datchik-30-sm-3-5-m>. (дата звернення: 23.03.2020)..

94. Модуль датчика температуры и влажности SHT1X ROBOTDYN [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/sensors/701-modul-datchika-temperatury-i-vlazhnosti-sht1x-robotdyn.html>. (дата звернення: 23.03.2020).

95. Микроволновый датчик движения RCWL-0516 [Електронний ресурс]. Режим доступу до інформації: [https://3v3.com.ua/product\\_7642.html](https://3v3.com.ua/product_7642.html). (дата звернення: 23.03.2020)..

96. Модуль трассировки пути Robotdyn [Електронний ресурс]. Режим доступу до інформації: <https://miniboard.com.ua/sensors/699-modul-trassirovki-puti-robotdyn.html>. (дата звернення: 23.03.2020).

97. Датчик освещенности ТЕМТ6000 для Ардуино [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod2560-datchik-osveshhenosti-temt6000-dlya-ardyino>. (дата звернення: 23.03.2020).

98. МАХ30102 - Цифровой датчик пульса I2C [Електронний ресурс]. Режим доступу до інформації: <https://1wire.com.ua/max30102-cifrovoy-datchik-pulsa-i2c.html>. (дата звернення: 23.03.2020).

99. Adafruit Sensiron SHT31-D Temperature & Humidity Sensor Breakout [Електронний ресурс]. Режим доступу до інформації: <https://xcraft.com.ua/adafruit-sensiron-sht31-d-temperature-and-humidity-sensor-breakout>. (дата звернення: 23.03.2020).

100. Датчик тока с аналоговым выходом на МАХ471 [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod1909-datchik-toka-s-analogovim-vihodom-na-max471>. (дата звернення: 23.03.2020).

101. Grove - UV Sensor [Електронний ресурс]. Режим доступу до інформації: <https://uamper.com/>. (дата звернення: 23.03.2020).

102. Лазерный датчик расстояния GY-530 на VL53L0X [Електронний ресурс]. Режим доступу до інформації: <https://arduino.ua/prod2144-lazernii-datchik-rasstoyaniya-gy-530-na-vl53l0x>. (дата звернення: 23.03.2020).

103. Модуль датчика RGB и жестов APDS-9960 (GY-9960) [Электронный ресурс]. Режим доступа до інформації: [https://diyshop.com.ua/index.php?route=product/product&product\\_id=408](https://diyshop.com.ua/index.php?route=product/product&product_id=408). (дата звернення: 23.03.2020).

104. Модуль датчика звука от Waveshare [Электронный ресурс]. Режим доступа до інформації: <https://arduino.ua/prod2619-modul-datchika-zvuka>. (дата звернення: 23.03.2020).

105. VL6180X - Датчик приближения и освещенности [Электронный ресурс]. Режим доступа до інформації: <https://1wire.com.ua/vl6180x-pximity-ambient-light.html>. (дата звернення: 23.03.2020).

106. Модуль датчика температуры і вологості DHT22 [Электронный ресурс]. Режим доступа до інформації: <https://diylab.com.ua/p111396915-modul-datchika-temperaturi.html>. (дата звернення: 23.03.2020).

107. Модуль датчика тока ACS712 [Электронный ресурс]. Режим доступа до інформації: [https://www.rcscomponents.kiev.ua/product/modul-datchika-toka-acs712\\_103100.html](https://www.rcscomponents.kiev.ua/product/modul-datchika-toka-acs712_103100.html). (дата звернення: 23.03.2020).

108. Модуль лазерного датчика Waveshare [Электронный ресурс]. Режим доступа до інформації: <https://miniboard.com.ua/sensors/274-modul-lazernogo-datchika-waveshare.html>. (дата звернення: 23.03.2020)..

109. H206 - модуль оптического датчика скорости вращения [Электронный ресурс]. Режим доступа до інформації: <https://controller.in.ua/infrakrasnij-shhelevoj-opticheskij-datchik>. (дата звернення: 23.03.2020).

110. Камера для Raspberry Pi [Электронный ресурс]. Режим доступа до інформації: <https://raspberrypi.com.ua/p/raspberrypi-camera-board/>. (дата звернення: 23.03.2020).

110. Raspberry Pi 3 Model B+ [Электронный ресурс]. Режим доступа до інформації: <https://raspberrypi.in.ua/product/mini-kompyuter-raspberrypi-3-model-b-plus/>. (дата звернення: 23.03.2020).

111. Raspberry Pi 3 Model B+ Specifications [Електронний ресурс]. Режим доступу до інформації: [https://raspberrypi.in.ua/wp-content/uploads/2018/03/datasheet\\_raspberry\\_pi\\_3\\_model\\_b.pdf](https://raspberrypi.in.ua/wp-content/uploads/2018/03/datasheet_raspberry_pi_3_model_b.pdf). (дата звернення: 23.03.2020).

112. Обмеження потужності Raspberry Pi [Електронний ресурс]. Режим доступу до інформації: <https://qastack.com.ua/raspberrypi/51615/raspberry-pi-power-limitations>. (дата звернення: 23.03.2020).

113. GPIO [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/documentation/usage/gpio/>. (дата звернення: 23.03.2020).

114. Последовательный интерфейс I2C [Електронний ресурс]. Режим доступу до інформації: <http://www.schematic.net/page-27.html>. (дата звернення: 23.03.2020).

115. Библиотека SPI [Електронний ресурс]. Режим доступу до інформації: <https://doc.arduino.ua/ru/prog/SPI>. (дата звернення: 23.03.2020).

116. UART и USART. COM-порт. Часть 1. [Електронний ресурс]. Режим доступу до інформації: [http://www.rotr.info/electronics/mcu/arm\\_usart.htm](http://www.rotr.info/electronics/mcu/arm_usart.htm). (дата звернення: 23.03.2020).

117. NOOBS [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/documentation/installation/noobs.md>. (дата звернення: 23.03.2020).

118. Raspberry Pi: Основы Linux/Базовые команды Linux (памятка для Raspberry Pi) [Електронний ресурс]. Режим доступу до інформації: [http://wikihandbk.com/wiki/Raspberry\\_Pi](http://wikihandbk.com/wiki/Raspberry_Pi). (дата звернення: 23.03.2020).

119. Экранная клавиатура [Електронний ресурс]. Режим доступу до інформації: <https://www.raspberrypi.org/forums/viewtopic.php?t=196824>. (дата звернення: 23.03.2020).

120. Налаштування VNC на Raspberry Pi [Електронний ресурс]. Режим доступу до інформації:

<http://isearch.kiev.ua/uk/searchpractice/methodsinstruments/1789-setting-up-vnc-on-raspberry-pi>. (дата звернення: 23.03.2020).

121. TeamViewer for Linux [Електронний ресурс]. Режим доступу до інформації: <https://www.teamviewer.com/ru/%D1%81%D0%BA%D0%B0%D1%87%D0%B0%D1%82%D1%8C/linux/>. (дата звернення: 23.03.2020).

122. Как настроить Samba для обмена файлами между компьютерами Ubuntu и Windows [Електронний ресурс]. Режим доступу до інформації: <http://blog.sedicomm.com/2018/07/25/kak-nastroit-samba-dlya-obmena-fajlami-mezhdu-kompyuterami-ubuntu-i-windows/>. (дата звернення: 23.03.2020).

123. Win32 Disk Imager. A Windows tool for writing images to USB sticks or SD/CF cards [Електронний ресурс]. Режим доступу до інформації: <https://sourceforge.net/projects/win32diskimager/>. (дата звернення: 23.03.2020).

124. Disk Utility User Guide [Електронний ресурс]. Режим доступу до інформації: <https://support.apple.com/guide/disk-utility/welcome/mac>. (дата звернення: 23.03.2020).

125. Python Source Releases [Електронний ресурс]. Режим доступу до інформації: <https://www.python.org/downloads/source/>. (дата звернення: 23.03.2020).

126. OpenCV Tutorials [Електронний ресурс]. Режим доступу до інформації: [https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html). (дата звернення: 23.03.2020).

127. Raspberry-gpio-python A Python module to control the GPIO on a Raspberry Pi [Електронний ресурс]. Режим доступу до інформації: <https://sourceforge.net/projects/raspberry-gpio-python/files/?%20Source%20=%20navbar>. (дата звернення: 23.03.2020).

128. Конструктор Шасси платформа четырехколесная Умный робот 4WD [Електронний ресурс]. Режим доступу до інформації: <https://izi.ua/p-1355687-konstruktor-shassi-platforma-chetyrekhkolesnaya-umniy-robot-4wd-ar>. (дата звернення: 23.03.2020).

129. Драйвер шагового двигателя L298N Dual Bridge DC Stepper Motor Controller Module [Електронний ресурс]. Режим доступу до інформації: <https://www.mini-tech.com.ua/drayer-dlya-dvigatelye-l298n>. (дата звернення: 23.03.2020).

130. DC-DC конвертер підвищуючий 3.5...32В в 5...35В на XL6009 [Електронний ресурс]. Режим доступу до інформації: <https://www.mini-tech.com.ua/dc-dc-konverter-povyshayushchij-xl6009>. (дата звернення: 23.03.2020).

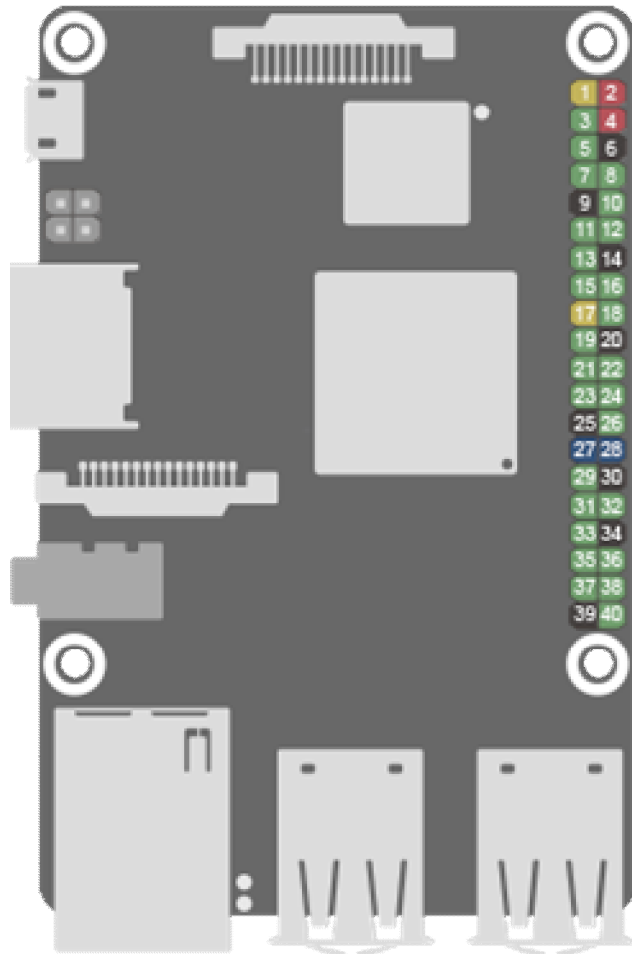
131. Raspberry Pi Distance Sensor using the HC-SR04 [Електронний ресурс]. Режим доступу до інформації: <https://pimylifeup.com/raspberry-pi-distance-sensor/>. (дата звернення: 23.03.2020).

132. Raspberry Pi Humidity Sensor using the DHT22 [Електронний ресурс]. Режим доступу до інформації: <https://pimylifeup.com/raspberry-pi-humidity-sensor-dht22/>. (дата звернення: 23.03.2020).

133. Сервопривод SG90 [Електронний ресурс]. Режим доступу до інформації: [https://autohome.org.ua/market/moto\\_servo/servo-sg90-9g-detail](https://autohome.org.ua/market/moto_servo/servo-sg90-9g-detail). (дата звернення: 23.03.2020).

## ДОДАТОК А

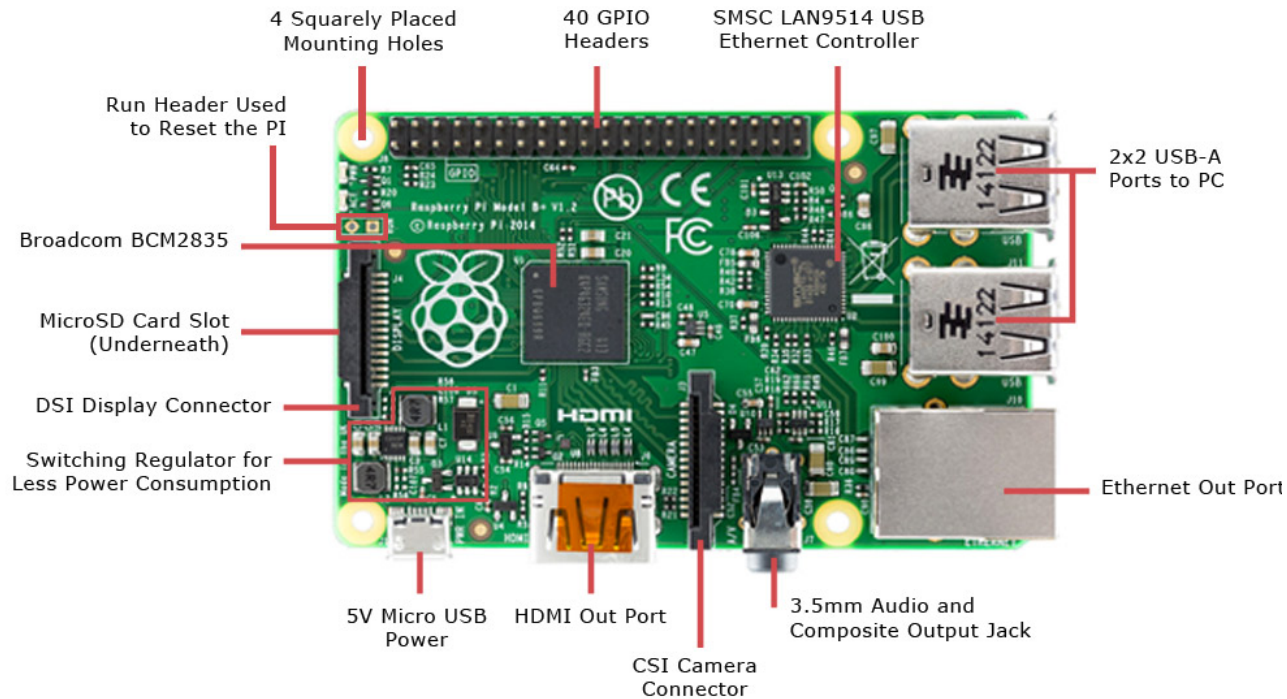
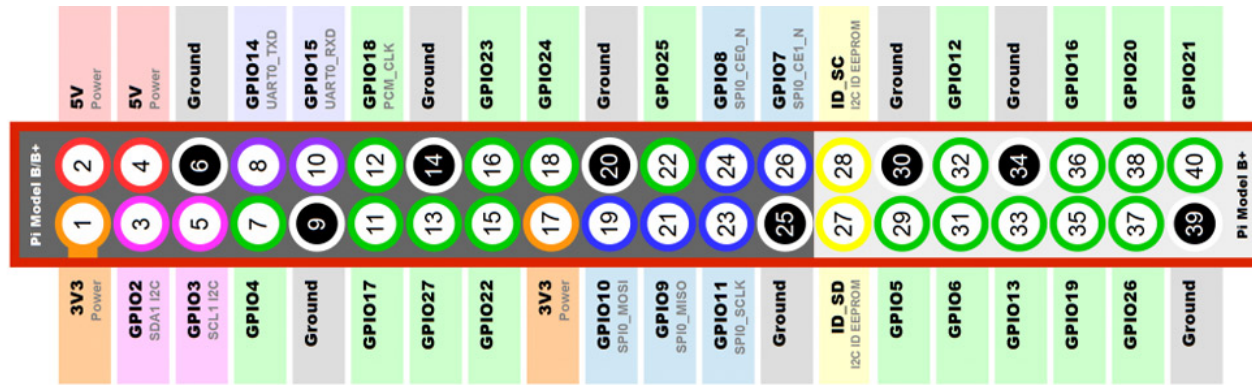
Розташування виводів Asus Tinker Board 2Гб на GPIO



1	VCC3.3V_IO	2	VCC5V_SYS
3	GP8A4_I2C1_SDA	4	VCC5V_SYS
5	GP8A5_I2C1_SCL	6	GND
7	GP0C1_CLKOUT	8	GP5B1_UART1TX
9	GND	10	GP5B0_UART1RX
11	GP5B4_SPI0CLK_UART4CTSN	12	GP6A0_PCM/I2S_CLK
13	GP5B6_SPI0_TXD_UART4TX	14	GND
15	GP5B7_SPI0_RXD_UART4RX	16	GP5B2_UART1CTSN
17	VCC3.3V_IO	18	GP5B3_UART1RTSN
19	GP8B1_SPI2TXD	20	GND
21	GP8B0_SPI2RXD	22	GP5C3
23	GP8A6_SPI2CLK	24	GP8A7_SPI2CSN0
25	GND	26	GP8A3_SPI2CSN1
27	GP7C1_I2C4_SDA	28	GP7C2_I2C4_SCL
29	GP5B5_SPI0CSN0_UART4RTSN	30	GND
31	GP5C0_SPI0CSN1	32	GP7C7_UART2TX_PWM3
33	GP7C6_UART2RX_PWM2	34	GND
35	GP6A1_PCM/I2S_FS	36	GP7A7_UART3RX
37	GP7B0_UART3TX	38	GP6A3_PCM/I2S_SDI
39	GND	40	GP6A4_PCM/I2S_SDO








## ДОДАТОК Б

### Розташування виводів Raspberry Pi 2 &3 на GPIO



## ДОДАТОК В

## Порівняльні характеристик основних моделі Raspberry Pi

	Model A	Model B	Model A +	Model B+	Raspberry Pi 2	Raspberry Pi 3	Raspberry Pi Zero
							
1	2	3	4	5	6	7	8
SoC	Broadcom BCM2835				Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2835
CPU	700 MHz ARM11 ARM1176JZF-S core				4-ех ядерный 900MHz ARMv7 Cortex-A7	4-ех ядерный 1.2GHz 64-bit ARMv8 Cortex-A53	1000MHz ARM1176JZ-F c LE
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264, 250 MHz					Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p60 H.264, 400 MHz	Broadcom VideoCore IV
SDRAM	256MB	512MB*	256MB	512 MB	1024 MB		512 MB
USB порты	1 USB 2.0	2 USB 2.0	1 USB 2.0	4 USB 2.0			1 Micro USB OTG
Video Out.	Композитний відеовихід RCA   HDMI (одночасне виведення зображення через RCA і HDMI не підтримується)		HDMI   Композитний відеовихід, поєднаний з 3.5мм аудіовиходом (для підключення RCA екрану необхідно використовувати спеціальний кабель)				HDMI **
Audi Out.	3.5 mm джек, HDMI						Багатоканальний HD звук через HDMI

1	2	3	4	5	6	7	8
Audi In.	Немає вбудованого аудіовходу, але підтримуються USB звукові карти з аудіовходом						
Карта пам'яті	SD / MMC / SDIO		MicroSD				
Мережа	-	10/100 Ethernet RJ45	-	10/100 Ethernet RJ45	10/100 Ethernet RJ45, встроений 802.11n Wi-Fi и Bluetooth 4.1		-
Низькорівнева периферія	26 вивідний контактний GPIO роз'єм, що включає в себе загальні порти введення / виводу, а також інтерфейси SPI, I <sup>2</sup> C, I <sup>2</sup> S, UART		40 вивідний контактний GPIO роз'єм, що включає в себе загальні порти введення / виводу, а також інтерфейси SPI, I <sup>2</sup> C, I <sup>2</sup> S, UART				40 вивідний контактний GPIO роз'єм, що включає в себе загальні порти введення / виводу, а також інтерфейси SPI, I <sup>2</sup> C, I <sup>2</sup> S, UART. ***
RTC	Вбудовані RTC відсутня ****						
Енергоспоживання	300 мА, (1.5 Вт)	700 мА, (3.5 Вт)	600мА - 1.2А	~650мА, (3.0 Вт)		800мА (4.0 Вт) - 2.5А	160мА
Живлення	5V (DC) через роз'єм MicroUSB або GPIO						
Габарити	65x56x15мм	85x56x17мм	65x56x12мм	85x56x17мм	85x56x17мм	85.6x56.5x17мм	65.0x30.0x5мм
Вага	31 г	40 г	23 г	40 г	40 г	45 г	9 г

\* - 256 МВ (до 15.10.2012) і 512МВ (з 15.10.2012)

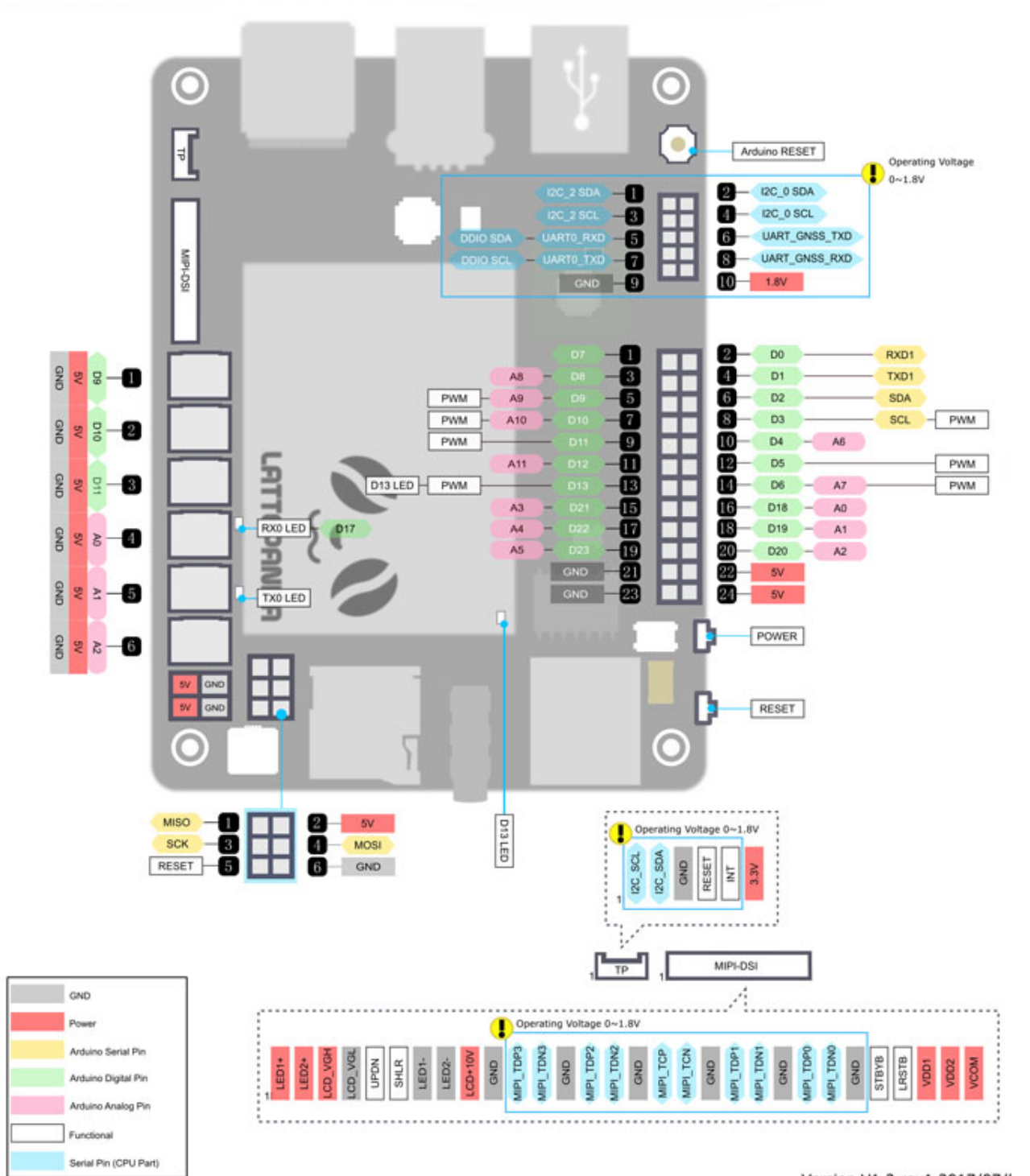
\*\* - композитний відеовихід не припаяний, передбачені 2 отвори для його монтажу

\*\*\* - Роз'єм не припаяний, потрібно самостійний монтаж

\*\*\*\* - Можна підключити RTC модуль по I<sup>2</sup>C

# ДОДАТОК Г

## Розташування виводів Asus Tinker Board 2Гб



## ДОДАТОК Д

Порівняльна таблиця ODROID-C2 з аналогічними мінікомп'ютерами

	<b>ODROID-C2</b>	<b>ODROID-C1+</b>	<b>RPi 2 Model B</b>
CPU	Amlogic S905 SoC 4 x ARM Cortex-A53 2GHz 64bit ARMv8 Architecture @28nm	Amlogic S805 SoC 4 x ARM Cortex-A5 1.5GHz 32bit ARMv7 Architecture @28nm	Broadcom BCM2836 4 x ARM Cortex-A7 900MHz 32bit ARMv7 Architecture @40nm
GPU	3 x ARM Mali-450 MP 700MHz	2 x ARM Mali-450 MP 600MHz	1 x VideoCore IV 250MHz
RAM	2GB 32bit DDR3 912MHz	1GB 32bit DDR3 792MHz	1GB 32bit LP-DDR2 400MHz
Flash Storage	Micro-SD UHS-1 @83Mhz/SDR50 or eMMC5.0 storage option	Micro-SD UHS-1 @78Mhz/SDR50 or eMMC4.5 storage option	Micro-SD @ 50Mhz/SDR25 No eMMC storage option
USB2.0 Host	4 Ports		
USB2.0 Device /OTG	1 Port for Linux USB Gadget device or USB host		No
Ethernet / LAN	10 / 100 / 1000 Mbit/s		10 / 100 Mbit/s
Video Output	HDMI 2.0 4K / 60Hz	HDMI 1.4	HDMI 1.4 / RCA / DSI
Audio Output	HDMI / I2S		MDMI / 3.5mm Jack / I2S
Camera Input	USB 720p		MIPI CSI 1080p
Real Time Clock	No (unless using an add-on module)	Yes (on-board RTC)	No (unless using an add-on module)
IR Receiver	Yes (on-board IR sensor)	Yes (on-board IR sensor)	No (unless using an add-on module)
IO Expansion	40 + 7 pin port GPIO / UART / I2C / I2S / ADC	40 + 7 pin port GPIO / UART / SPI / I2C / I2S / ADC	40 pin port GPIO / UART / SPI / I2S
ADC	10 bit SAR 2 channels		No (unless using an add-on board)
Heat sink	Included		Optional
Size	85 x 56 mm (3.35 x 2.2 inch)	85 x 56 mm (3.35 x 2.2 inch)	85 x 56 mm (3.35 x 2.2 inch)
Weight	40g	40g	42g

## ДОДАТОК Е

Програма сервісної частини дистанційного керування мобільним роботом

```
import RPi.GPIO as GPIO
import socket
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(33, GPIO.OUT)
GPIO.setup(11, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
GPIO.setup(29, GPIO.OUT)
GPIO.setup(31, GPIO.OUT)
GPIO.output(29, True)
GPIO.output(31, True)

UDP_IP = "0.0.0.0"
UDP_PORT = 5050

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))

while True:
    raw=sock.recvfrom(1024)

    #print type(raw)
    rawnew = ('forward', 'stop', 'backward', 'left', 'right')

    print type(rawnew)

    if raw[0] == rawnew[0]:
        print 'robot moves forward'
        GPIO.output(33, True)
        GPIO.output(11, False)
        GPIO.output(13, True)
        GPIO.output(15, False)
        time.sleep(1)

    elif raw[0] == rawnew[1]:
        print 'robot stop'
        GPIO.output(33, False)
```

```
GPIO.output(11, False)
GPIO.output(13, False)
GPIO.output(15, False)
time.sleep(1)

elif raw[0] == rawnew[2]:
print "robot moves backward"
GPIO.output(33, False)
GPIO.output(11, True)
GPIO.output(13, False)
GPIO.output(15, True)
time.sleep(1)

elif raw[0] == rawnew[3]:
print "robot moves left"
GPIO.output(33, False)
GPIO.output(11, True)
GPIO.output(13, True)
GPIO.output(15, False)
time.sleep(1)

elif raw[0] == rawnew[4]:
print "robot moves right"

GPIO.output(33, True)
GPIO.output(11, False)
GPIO.output(13, False)
GPIO.output(15, True)
time.sleep(1)
else:
print "message error"
GPIO.output(33, False)
GPIO.output(11, False)
GPIO.output(13, False)
GPIO.output(15, False)
time.sleep(1)
```

## ДОДАТОК Є

Програмний засіб серверної частини системи дистанційного керування  
мобільного робота

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using RoboControl.Presenter;

namespace RoboControl
{
    public partial class Form1 : Form
    {
        Presenter.Presenter presenter;

        public Form1()
        {
            InitializeComponent();
        }

        public void ShowMessage(string text)
        {
            if (text != null)
            {
                MessageBox.Show(text);
            }
        }

        private void bn_appy_configs_Click(object sender, EventArgs e)
        {
            if (tb_port.Text.Length == 4 && tb_host.Text.Length > 5)
            {
                presenter = new Presenter.Presenter(this, Convert.ToInt32(tb_port.Text),
                tb_host.Text);

                var page = @"

```

```

        <iframe src='http://'+ tb_host.Text + "' width='100%' height='100%'"> </
        iframe > ";
        webBrowser.DocumentText = page;
        webBrowser.ScriptErrorsSuppressed = true;

        MessageBox.Show("Configs has been saved");
    }
    else
    {
        MessageBox.Show("Uncorrect value");
    }
}

private void bn_event(object sender, EventArgs e)
{
    switch( ((Button)sender).Name )
    {
        case "bn_command_1":
            presenter.Send_command("value", Commands.FORWARD);
            break;
        case "bn_command_2":
            presenter.Send_command("value", Commands.BACKWARD);
            break;
        case "bn_command_3":
            presenter.Send_command("value", Commands.LEFT);
            break;
        case "bn_command_4":
            presenter.Send_command("value", Commands.RIGHT);
            break;
        case "bn_command_5":
            presenter.Send_command("value", Commands.STOP);
            break;
        default:
            break;
    }
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    MessageBox.Show("connection has been closed");
    if (presenter != null)
    {
        presenter.CloseConnection();
    }
}

```

```
private void bn_KeyDown(object sender, KeyEventArgs e)
{
    switch (((Button)sender).Name)
    {
        case "bn_command_1":
            presenter.Send_command("value", Commands.FORWARD);
            break;
        case "bn_command_2":
            presenter.Send_command("value", Commands.BACKWARD);
            break;
        case "bn_command_3":
            presenter.Send_command("value", Commands.LEFT);
            break;
        case "bn_command_4":
            presenter.Send_command("value", Commands.RIGHT);
            break;
        case "bn_command_5":
            presenter.Send_command("value", Commands.STOP);
            break;
        default:
            break;
    }
}
```

```
private void bn_KeyPress(object sender, KeyPressEventArgs e)
{
    switch (((Button)sender).Name)
    {
        case "bn_command_1":
            presenter.Send_command("value", Commands.FORWARD);
            break;
        case "bn_command_2":
            presenter.Send_command("value", Commands.BACKWARD);
            break;
        case "bn_command_3":
            presenter.Send_command("value", Commands.LEFT);
            break;
        case "bn_command_4":
            presenter.Send_command("value", Commands.RIGHT);
            break;
        case "bn_command_5":
            presenter.Send_command("value", Commands.STOP);
            break;
        default:
            break;
    }
}
```

```

}
}
}
class Presenter
{

    Form1 form;
        int socet_port;
        Socket socet;

    public Presenter(Form1 form, int port, string host)
    {
        this.form = form;
        this.socet_port = port;

        socet = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
            ProtocolType.Udp);

    try
    {
        socet.Connect(host, port);
        this.form.ShowMessage("Connecting");
    }
    catch(Exception ex)
    {
        this.form.ShowMessage(ex.ToString());
    }
    }

    public Bitmap GetFrame()
    {
        var width = 410;
        var height = 325;

        Bitmap bmp = new Bitmap(width, height);

        if (socet != null)
        {
            using (Graphics gfx = Graphics.FromImage(bmp))
            using (SolidBrush brush = new SolidBrush(Color.FromArgb(0, 255, 0)))
            {
                gfx.FillRectangle(brush, 0, 0, width, height);
            }
        }
    }
}

```

```

return bmp;
}

public void Send_command(string value, Commands command)
{
    if (socet != null)
    {
        byte[] send_buf = new byte[200];

        switch (command)
        {
        case Commands.FORWARD:
            send_buf = Encoding.ASCII.GetBytes("forward");
            break;
        case Commands.BACKWARD:
            send_buf = Encoding.ASCII.GetBytes("backward");
            break;
        case Commands.LEFT:
            send_buf = Encoding.ASCII.GetBytes("left");
            break;
        case Commands.RIGHT:
            send_buf = Encoding.ASCII.GetBytes("right");
            break;
        case Commands.STOP:
            send_buf = Encoding.ASCII.GetBytes("stop");
            break;
        default:
            break;
        }
        socet.Send(send_buf);
    }
}

public void CloseConnection()
{
    if (socet != null)
    {
        socet.Close();
    }
}

enum Commands
{
    FORWARD, BACKWARD, LEFT, RIGHT, STOP
}
}

```

## ДОДАТОК Ж

Скрипт Python для ультразвукового далекоміра

```

import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
TRIG = 16
ECHO = 18
GPIO.setup(TRIG, GPIO.OUT, initial=0)
GPIO.setup(ECHO, GPIO.IN)

# Minimum delay between measurements is 50ms. 100ms is fine:
time.sleep(0.1)

# Send a probe signal to the sensor. Signal should be 10 micro sec long.
GPIO.output(TRIG,1)
time.sleep(0.00001)
GPIO.output(TRIG,0)

while GPIO.input(ECHO) == 0:
    pass
start = time.time()

while GPIO.input(ECHO) == 1:
    pass stop = time.time()

# Print distance to object in santimeters. Sound speed = 340 m/s
print "Distance = ",(stop - start) * 17000,"sm"
print "start time = ", start
print "stop time = ", stop

GPIO.cleanup()

-----

$ cat 5_ultrasonic_distance_02_infinet-loop.py
import RPi.GPIO as GPIO
import time #GPIO.cleanup()

GPIO.setmode(GPIO.BOARD)
TRIG = 16
ECHO = 18

```

```
GPIO.setup(TRIG, GPIO.OUT, initial=0)
GPIO.setup(ECHO, GPIO.IN)

try:
    while True:
        # Minimum delay between measurements is 50ms. 100ms is fine:
        time.sleep(0.1)

        # Send a probe signal to the sensor. Signal should be 10 micro sec long.
        GPIO.output(TRIG,1)
        time.sleep(0.00001)
        GPIO.output(TRIG,0)

        while GPIO.input(ECHO) == 0:
            pass start = time.time()

        while GPIO.input(ECHO) == 1:
            pass stop = time.time()

        # Print distance to object in santimeters. Sound speed = 340 m/s
        print "Distance = ",(stop - start) * 17000,"sm"
        print "start time = ", start
        print "stop time = ", stop

except KeyboardInterrupt:
    GPIO.cleanup()
```

## ДОДАТОК 3

Скрипт Python для одночасного управління двома сервомоторами

```

from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

pan = 27
tilt = 17

GPIO.setup(tilt, GPIO.OUT) # white ==> TILT
GPIO.setup(pan, GPIO.OUT) # gray ==> PAN

def setServoAngle(servo, angle):
    assert angle >=30 and angle <= 150
    pwm = GPIO.PWM(servo, 50)
    pwm.start(8)
    dutyCycle = angle / 18. + 3.
    pwm.ChangeDutyCycle(dutyCycle)
    sleep(0.3)
    pwm.stop()

if __name__ == '__main__':
    import sys
    if len(sys.argv) == 1:
        setServoAngle(pan, 90)
        setServoAngle(tilt, 90)
    else:
        setServoAngle(pan, int(sys.argv[1])) # 30 ==> 90 (middle point)
        ==> 150
        setServoAngle(tilt, int(sys.argv[2])) # 30 ==> 90 (middle point) ==>
150
    GPIO.cleanup()

```

Коли скрипт виконується, Ви повинні ввести параметри, кут повороту і кут нахилу. наприклад:

```
sudo python3 servoCtrl.py 45 120
```

Вище зазначена команда буде позиціонувати механізм повороту/нахилу з 45 градусами за «азимуту» (кут повороту) і 120 градусів "висота" (кут нахилу). Зверніть увагу, що, якщо параметри не було введено, за замовчуванням буде встановлено обидва кута повороту і нахилу до 90 градусів.