

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій та радіотехніки

КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

Бакалавр

Освітній рівень

Спеціальність 172 Телекомунікації та радіотехніка

Шифр і назва спеціальності

на тему: Проєктування систем віддаленого керування на основі міні комп'ютера Raspberry PI

КПТР. 2017017.01.09 ПЗ

Виконав:

студент 4 курсу, група ТР-17-1



підпис

Мокрицький А.Б.

Ініціали, прізвище

Керівник:

д-р техн.наук, проф.



підпис

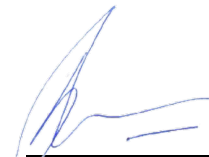
Мясіщев О.А.

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри ТР

д-р техн. наук, проф.



підпис

Бойко Ю.М.

Ініціали, прізвище

«20» травня 2021р.

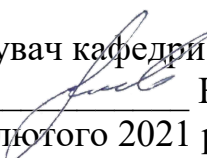
Хмельницький, 2021

Хмельницький національний університет

Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра телекомунікацій та радіотехніки
Спеціальність 172 – Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри ТР

 Юлій БОЙКО
«10» лютого 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНИЙ ПРОЄКТ**

дипломнику Мокрицькому Андрію Богдановичу
(Прізвище, ім'я, по батькові студента)

1 Тема проєкту Проектування систем віддаленого керування на основі міні комп'ютера Raspberry PI

Затверджено наказом по університету від «05» лютого 2021р. № 11.

2 Строк подання студентом роботи на кафедру: «20» травня 2021р.

3 Вихідні дані (характеристика об'єкта, умов дослідження та ін.)

- 1) Розглянути загальну структуру схему, принципи роботи плати Raspberry PI.
- 2) Порівняти існуючі мікрокомп'ютери та визначити основні переваги та недоліки застосування їх.
- 3) Розглянути питання підключення окремих вузлів до плати Raspberry PI.
- 4) Проаналізувати схемотехнічні рішення, що використано в платі Raspberry PI.
- 5) Виконати розробку та проектування плати розширення.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

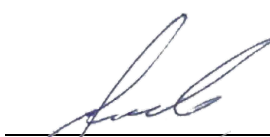
1. Вступ.
2. Аналіз існуючих рішень.
3. Електричний розрахунок пристрою
4. Проектування схеми

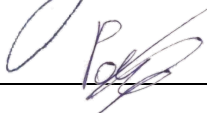
5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

- 1) Плакат 1. Огляд технології PLC.
- 2) Плакат 2. Сигнали технології PLC.
- 3) Схема структурна пристрою збору та передачі інформації за технологією PLC.
- 4) Креслення А1 Схема електрична принципова.

6 Термін подання роботи до захисту 20 травня 2021 року

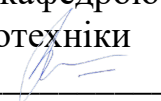
7 Дата видачі завдання 10 лютого 2020 року

Науковий керівник 

Завдання отримав 

ЗАТВЕРДЖУЮ

Зав. кафедрою телекомунікації та
радіотехніки

 Юлій БОЙКО

«10» лютого 2021 р.


ПЛАН ІНДИВІДУАЛЬНОЇ РОБОТИ

№ п/п	Найменування виду роботи	Форма звітності, термін виконання	Відмітка наукового керівника
1	Огляд літератури	Вступ 14.02	Виконано
2	Огляд мікро-ЕОМ Raspberry PI	Розділ 1 20.02	Виконано
3	Розробка структури пристрою	Розділ 2 1.03	Виконано
4	Розробка схеми електричної принципової	Розділ 3 15.03	Виконано
5	Плакат. Плати Raspberry PI.	Плакат 5.04	Виконано
6	Схема електрична принципова	Креслення 29.04	Виконано
7	Висновки по роботі	Висновки 6.05	Виконано

Примітки:

1. _____
2. _____


Науковий керівник



(підпис, дата)

О.А. Мясіщев

Студент






(підпис, дата)

А.Б. Мокрицький

Зміст

ВСТУП.....	6
Розділ 1 ОГЛЯД ПИТАННЯ ТА АНАЛІЗ ВІДОМИХ РІШЕНЬ	8
1.1 Аналоги одноплатного мини-ПК Raspberry PI	8
1.2 Порівняння Raspberry PI з Orange Pi	9
1.2.1 Переваги Orange Pi.....	9
1.2.2 Недоліки Orange Pi.....	9
1.3 Відмінності Raspberry E14 і RS	10
1.4 Відмінності Banana Pi від Raspberry	11
1.5 Огляд плат Raspberry PI.....	12
1.6 Операційні системи.....	15
1.6.1 Установка Raspbian.....	17
1.7 Аналіз схемотехнічних рішень плати Raspberry PI	18
1.7.1 Структура портів GPIO.....	18
1.7.2 Порт USB 3.0	19
1.7.3 Перефрійне обладнання	20
1.7.4 Аудіо канали.....	25
1.7.5 Джерело живлення 1.0 В	27
1.7.6 Ethernet канал.....	28
1.7.7 Центральний процесор системи	29
1.7.8 Тактова частота системи, керування частотою та терморегулювання	30
Розділ 2 ЕЛЕКТРИЧНИЙ РОЗРАХУНОК ПРИСТРОЮ.....	32
2.1 Робота з GPIO	32
2.2 Програмування GPIO.....	36

					КПТР. 2017010.01.07 ПЗ			
		№докум.	Підпис	Дата				
Виконав	Мокрицький А.Б.				Проектування систем видаленого керування на основі міні комп'ютера Raspberry PI Пояснювальна записка	Літера	Аркуш	Аркушів
Перевір.	Мясіщев О.А.					у	4	
Т.Контр.						ТР-17-1, ФПКТС,		
Н.контр.								
Затвер.								

2.3	Перспективи використання Raspberry PI.....	39
2.3.1	Чому вибирають Raspberry PI?	40
2.3.2	Тривалість життєвого циклу та обрання системи на чіпові	41
2.3.3	Запуск операційної системи (ОС) на SD -карті	42
2.3.4	Наявність USB-інтерфейсів	43
2.3.5	Зовнішнє джерело живлення?.....	44
2.3.6	Проблема відведення тепла	44
2.3.7	Потреби масштабованості	46
Розділ 3 ПРОЄКТУВАННЯ СХЕМИ ПРИСТРОЮ		47
3.1	Існуючі рішення щодо застосування Raspberry PI	47
3.1.1	Плата розширення портів введення / виводу Raspberry PI відкриває можливості навігації та автоматизації	47
3.2	Розробка крос-плати для Raspberry PI	53
3.2.1	Вхідний каскад	54
3.2.2	Джерело живлення	54
3.2.3	Релейний модуль	55
3.3	Програмна частина системи контролю та віддаленого керування	56
3.3.1	Мова програмування для Raspberry PI	57
3.3.2	Середовище розробки програмного коду Thonny	57
3.3.3	Мережеві налаштування для Raspberry Pi.....	59
3.3.4	Встановлення фреймворка Webіорі	60
3.3.5	Файла-скрипт на питоне script.py.....	66
3.3.6	Побудова графіка температури	66
3.4	Висновки	67
Перелік літературних джерел.....		68

ВСТУП

Ідея про створення дешевого комп'ютера з'явилася в 2006 році у групи колег: Ебена Аптона, Роба Маллінс, Джека Ланга і Алана Майкрофта. Вони створили кілька прототипів, після чого до них приєднався Девід Бребен, вони заснували фонд Raspberry PI Foundation [en] і почали роботу над комп'ютером.

У травні 2011 року Бребен представив перший концепт Raspberry PI розміром з USB-флеш-накопичувач [8] [9].

В кінці липня 2011 року було закінчено і відправлена у виробництво альфа-версія плати, а вже 12 серпня Raspberry PI Foundation отримала першу партію пристроїв. Альфа-версія комп'ютера містила деякі тестові функції і дорогі деталі, які прибрали з фінальної версії. Також кінцева версія плати на 20% менше і складається з чотирьох шарів, а не з шести. 14 липня 2014 року розробники проекту випустили Raspberry PI «B +».

26 листопада 2015 року розробники проекту випустили новий мікрокомп'ютер Raspberry PI Zero. Основні відмінності - ціна в п'ять доларів і незмонтовану роз'єм GPIO [14]. В подальшому був представлений Raspberry PI 3. Головні відмінності: 64-бітний процесор, наявність Wi-Fi, Bluetooth, а також 100 / 10Мбіт / с Ethernet.

Протягом наступного 2017 року розробники випустили Raspberry PI Zero W. Головні відмінності: наявність Wi-Fi і Bluetooth, а трохи згодом розробники випустили Raspberry PI 3B +. Головні відмінності: більш потужний процесор, 1 Гбіт / с Ethernet (підключений через «віртуальну» мережеву карту поверх USB 2.0, тому реальні швидкості нижче заявлених), двохдіапазонний Wi-Fi і Bluetooth 4.2.

У червні 2019 року була представлена модель Raspberry PI 4B [18]. Вона отримала новий чотирьохядерний процесор з ARM Cortex-A72 1.5 ГГц,

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		6

доступна в 4 варіантах з 1, 2, 4 або 8 Гб оперативної пам'яті, має повно швидкісну 1 Гбіт / с Ethernet, Bluetooth 5.0. З 4 портів USB два тепер мають формат USB 3.0. Для підключення моніторів є 2 порти micro HDMI (2 по 4K 30 fps або 1 на 4k 60 fps). З боку графіки використовується VideoCore VI (OpenGL ES 3.x) і доданий апаратний декодер 4K для HEVC-відео.

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		№докум.	Підпис	Дата		7

Розділ 1 ОГЛЯД ПИТАННЯ ТА АНАЛІЗ ВІДОМИХ РІШЕНЬ

1.1 Аналоги одноплатного мини-ПК Raspberry PI

Свого часу через відсутність аналогів Raspberry PI зумів притягнути увагу споживачів і принести своїм творцям колосальний прибуток. Проте відтоді пройшло багато часу і на ринку з'явилася безліч подібних пристроїв.

Порівняємо аналоги Raspberry і порівняємо їх можливостей.



Рисунок 1.1 – Плата HiKey

					КПТР. 2017010.01.07 ПЗ			
		№докум.	Підпис	Дата				
Виконав	Мокрицький А.Б.				Проектування систем видаленого керування на основі міні комп'ютера Raspberry PI Пояснювальна записка	Літера	Аркуш	Аркушів
Перевір.	Мясищев О.А.					у	8	
Т.Контр.						ТР-17-1, ФПКТС,		
Н.контр.								
Затвер.								

1.2 Порівняння Raspberry Pi з Orange Pi

У відкритий продаж Orange Pi поступив у кінці 2015 року і був покликаний витіснити Raspberry своїми характеристиками при ціні до 15\$. На відміну від оригіналу, виробництво якого концентрується в Англії, Orange робиться виключно в Китаї. Тому давайте порівняємо характеристики пристроїв і з'ясуємо, що краще: Raspberry Pi або Orange Pi.

1.2.1 Переваги Orange Pi

1. Ціна варіюється від 8,5 до 15\$, тоді як Raspberry Pi стоять від 10 до 35\$ (якщо не рахувати Zero, вартість якого не перевищує 5\$).

2. 4-ядерний процесор на базі Allwinner H3, який забезпечує високу продуктивність при запуску повноцінних застосувань через операційні системи Android або Windows.

3. Для обробки графічної інформації використовується Mali - 400 MP2, з тактовою частотою в 600 MHz. Аналогічний відеоприскорювач функціонує в мобільних пристроях і зарекомендував себе програванням відео високої якості при стабільній частоті кадрів.

4. Об'єм оперативної пам'яті налічує 1Gb. Цього цілком достатньо для роботи більшості операційних систем і додатків.

5. 3-и USB-порти для підключення пристроїв, microUSB-OTG для забезпечення пристрою електрикою або підключення периферії, виходи для HDMI і AV, для підключення відеопристроїв, і Ethernet для підключення до локальної мережі.

1.2.2 Недоліки Orange Pi

1. Для живлення МІНІ-ПК необхідно використати блоки живлення з потужністю 2-3А (залежно від кількості підключеної периферії).

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		9

2. Необхідність в оснащенні пристрою додатковим охолодженням у вигляді радіатора. Якщо Ви відмовитися від охолодження — плата прослужить не більше місяця при середніх навантаженнях або пара днів при високих. Краще придбати мініатюрні радіатори для Raspberry, які відмінно себе почувають на Orange.

3. Для стабільної роботи пристрою з картою пам'яті звертайте увагу на швидкість її роботи. Ідеальну продуктивність демонструють накопичувачі з класом швидкості Class 10 UHS - 1 і вище. У іншому випадку робота більшості операційних систем супроводжуватиметься постійними фризами і зависаннями.

4. У продажу відсутні моделі з Wi — Fi. Якщо Ви хочете використати безпроводне підключення — доведеться докупати модуль (підключається до пристрою через USB і коштує біля 4\$).

Недоліків трохи, проте, вони істотні. Тому перш, ніж купувати Raspberry Pi або Orange Pi, визначитеся, для чого вам потрібний МІНІ-ПК, а вже потім оформляйте замовлення.

Проведемо порівняння Raspberry Pi від різних компаній.

1.3 Відмінності Raspberry E14 і RS

1. Різниця між найменуваннями полягає у виробниках плат, які виступають і продавцем міні-ПК.

2. Відмінність Raspberry Pi RS від Element14 полягає в тому, що усі виробничі потужності RS розташовані в Англії, тоді як виробництво E14 розділене між Китаєм і Англією. Через це і може спостерігатися незначна різниця в продуктивності і терміні служби різних плат.

3. Процесор Raspberry Pi Element14 у більшості тестів демонструє продуктивність на 0,5-1% вище, ніж його аналог.

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		10

посібник по інформатиці. Але сама ідея виявилася настільки вдалою, що за декілька років міні-комп'ютер став популярний в дуже широких кругах. З часом Raspberry PI пережила декілька модифікацій, кожна з яких відрізнялася від попередника яким-небудь параметром. Такий підхід дозволив регулювати вартість виробу залежно від потреб користувача, що також позитивно позначилося на популярності пристрою. Уся лінійка Raspberry PI застосовує процесори з Арм-архітектурою, яка зарекомендувала себе з кращого боку. На малюнку №1 показаний зовнішній вигляд однієї з популярних плат Raspberry PI B+.

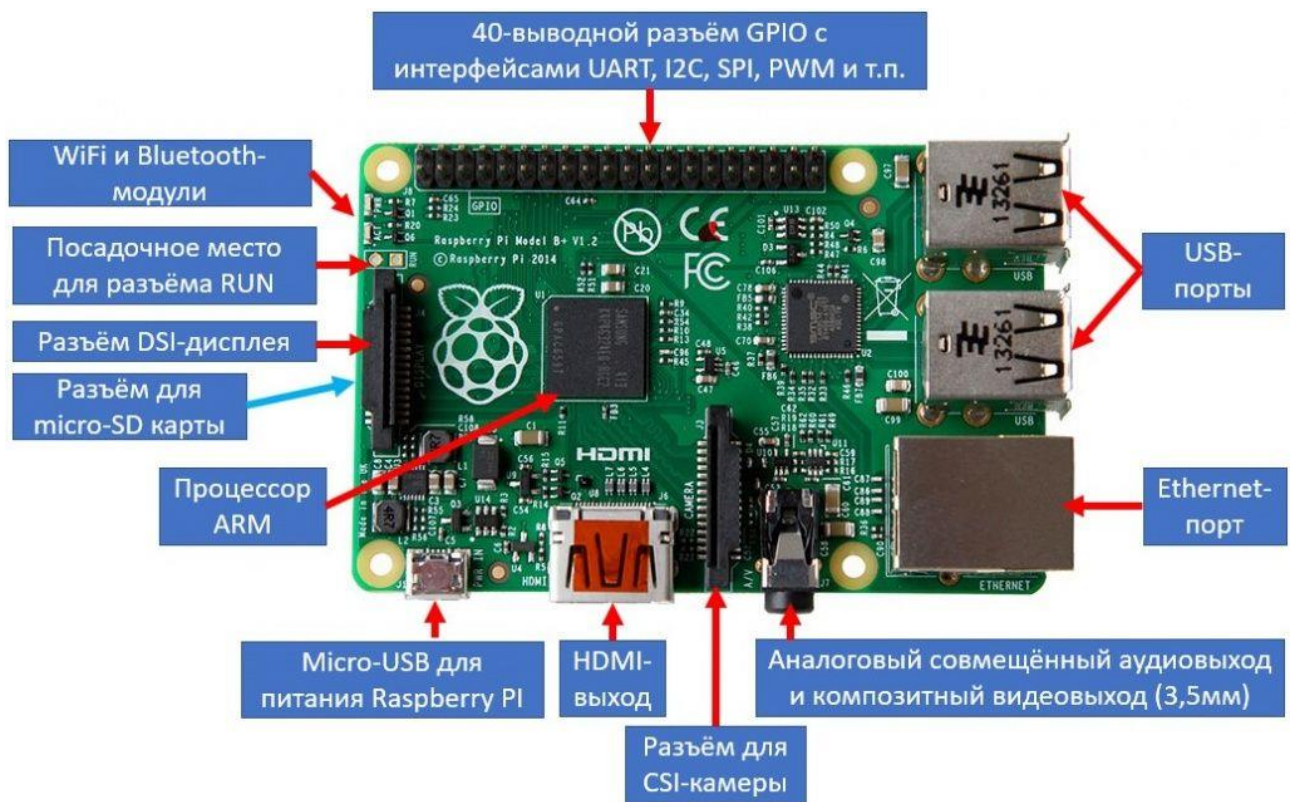


Рисунок 1.4 – Огляд складових елементів Raspberry PI

На (період 2012-2019гг.) сьогодні існує 11 різновидів Raspberry PI. Останні версії оснащені безпроводними WiFi і Bluetooth модулями, що

розширюють межі застосування міні-ПК в області Ethernet -технологій. Нижче приведена порівняльна таблиця, в якій відбиті особливості кожної модифікації з вказівкою деяких технічних даних.

Таблиця 1.1 – Таблиця порівняння параметрів плат Raspberry PI

Модифікація	Процесор	Тактова частота	Кількість ядер	Об'єм ОЗУ	Кількість GPIO	Кількість USB	Піддержка Ethernet	Піддержка WiFi	Піддержка Bluetooth	Год выпуска
B	ARM1176JZ-F	700 МГц	1	512 МБ	26	2	√			2012
A	ARM1176JZ-F	700 МГц	1	256 МБ	26	1				2013
B+	ARM1176JZ-F	700 МГц	1	512 МБ	40	4	√			2014
A+	ARM1176JZ-F	700 МГц	1	256 МБ	40	1				2014
2B	ARM Cortex-A7	900 МГц	4	1 ГБ	40	4	√			2015
Zero	ARM1176JZ-F	1 ГГц	1	512 МБ	40	1				2015
3B	Cortex-A53 (ARM v8)	1,2 ГГц	4	1 ГБ	40	4	√	802.11n	4.1	2016
Zero W	ARM1176JZ-F	1 ГГц	1	512 МБ	40	1		802.11n	4.0	2017
3B+	Cortex-A53 (ARM v8)	1,4 ГГц	4	1 ГБ	40	4	√	802.11n	4.2	2018
3A+	Cortex-A53 (ARM v8)	1,4 ГГц	4	512 МБ	40	1		802.11n	4.2	2018
4B	Cortex-A72 (ARM v8)	1,5 ГГц	4	1, 2, 4 ГБ	40	4	√	802.11n	5.0	2019

Як видно з наведеної вище таблиці, навіть сама молодша модель в лінійці має цілком серйозні характеристики, враховуючи те, що це одноплатний комп'ютер розміром трохи більше кредитної карти.

На рис. 1.5 зображена остання на момент написання статті модифікація Raspberry PI 4B, запущена в продаж в червні 2019г. Вона оснащена додатковим графічним процесором VideoCore VI (OpenGL ES 3.x), а також апаратним декодером 4Kp60 для відтворення HEVC відео. Два порти мікроHDMI з можливістю пропускати сигнал до 4K, дозволяють підключити одночасно два монітори.

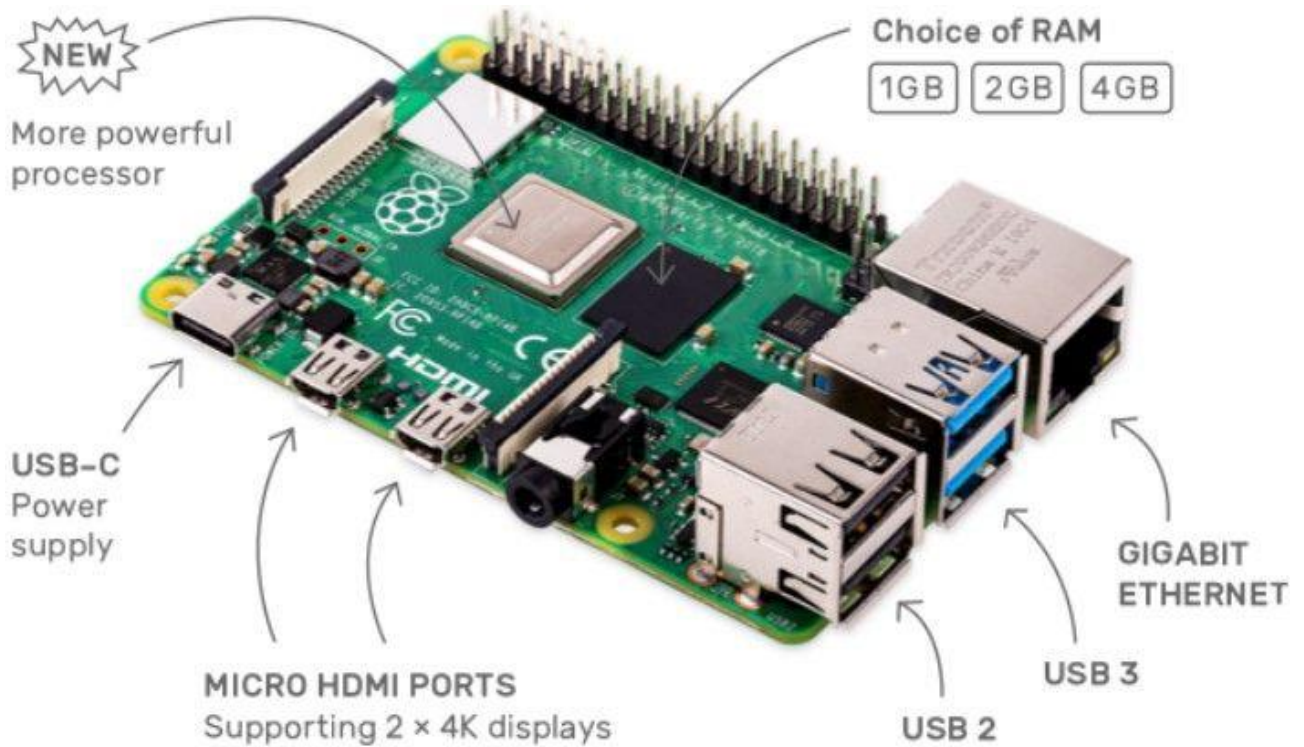


Рисунок 1.5 – Зовнішній вигляд Raspberry PI 4B

Основною відмінною рисою Raspberry PI від звичайних комп'ютерів, являється наявність програмованих портів введення-виведення GPIO. За допомогою їх можна управляти різними пристроями і приймати телеметрію з різного роду датчиків.

1.6 Операційні системи

Хоч Raspberry PI зовні може нагадати нам Arduino, він все-таки використовує кардинально інший метод функціонування. Ця плата, як і звичайний ПК, працює під управлінням однієї із спеціалізованих операційних систем. Залежно від сфери застосування або особистих симпатій, кожен може вибрати для себе свою. Нижче приведений перелік найбільш популярних «операціонок» для Raspberry PI з їх коротким описом.

Raspbian - ця операційна система в 2015 році була представлена як основна для Raspberry PI. Вона по максимуму оптимізована для процесорів з Арм-архітектурою і досить активно продовжує розвиватися. Основою операційної системи є Debian GNU/Linux. Середовище робочого столу складається з LXDE (середовище для UNIX і інших POSIX –сумісних систем типу Linux і BSD), а також менеджера вікон Openbox (безкоштовний менеджер для X Window System). До складу дистрибутива входять: програма комп'ютерної алгебри Mathematica; модифікована версія Minecraft PI; урізана версія Chrome.

Debian - операційна система з відкритим початковим кодом. До складу Debian входить більше 59000 пакетів вже скомпільованого ПО. Система використовує ядро Linux або FreeBSD. У стандартний дистрибутив включені: середовище робочого столу GNOME з набором найбільш популярних програм, таких як Firefox, LibreOffice, Evolution, і інший набір для роботи з мультимедіа. Також є можливість установки образів з використовуваними середовищами робочих столів KDE, Xfce, LXDE, MATE і Cinnamon.

Ubuntu - система заснована на Debian GNU/Linux. За популярністю Ubuntu займає перше місце серед дистрибутивів Linux, призначених для web - серверов. До складу дистрибутива входять: програма для перегляду Інтернет; офісний пакет, програми для комунікації і так далі

Fedora - ця операційна система заснована на дистрибутиві Linux від відомої фірми Red Hat. До складу дистрибутива входять LibreOffice, Mozilla Firefox, а також інше ПО, яке можна додатково встановити через Цент Додатків GNOME.

Arch Linux - це вільно поширюваний дистрибутив GNU/Linux загального призначення. Особливістю цієї системи є відсутність графічного установника, що може неабияк потренувати навички затятих дослідників Linux.

Gentoo Linux - один з популярних дистрибутивів GNU/Linux з гнучкою технологією управління пакетами. У системі передбачена можливість максимальної оптимізації під конкретне апаратне рішення. Алгоритм управління пакетами дає можливість легко реалізувати як робочу станцію, так і сервер.

RISC OS - операційна система спеціально розроблялася для процесорів з архітектурою ARM. Особливості ядра RISC OS дозволяють системі робити прискорений запуск за рахунок зберігання даних в ПЗП. Такий підхід також допомагає захистити дані різного роду збоїв і впливи шкідливого ПО.

OpenELEC - це програмний комплекс для організації домашнього кінотеатру під управлінням GNU/Linux.

OSMC - ще один комплекс для реалізації домашнього кінотеатру.

У мережі Інтернет, окрім перерахованих операційних систем, можна знайти ще безліч модифікацій для найрізноманітніших призначень. Але оскільки Raspbian є основним середовищем для Raspberry PI, то надалі спиратимемося саме на неї.

1.6.1 Установка Raspbian

Для установки операційної системи необхідно підключити до плати Raspberry PI мінімальний набір периферії, а саме: монітор, клавіатуру і мишу. Далі, потрібна SD -карта із записаним чином Raspbian. Саме з неї і буде робиться установка.

Для того, щоб записати образ на карту пам'яті, її необхідно вставити в комп'ютер і відформатувати в системі FAT32. Зробити це можна як стандартними засобами Windows, так і сторонніми програмами, наприклад - SD Memory Card Formatter. Після чого, викачуємо дистрибутив операційної системи з офіційного сайту Raspberry. Для недосвідчених користувачів,

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		Нодокум.	Підпис	Дата		17

доступна спрощена версія установника NOOBS. Після закінчення завантаження, архів необхідно розпакувати в корінь карти пам'яті. На цьому підготовчий етап закінчений.

Вставляємо карту пам'яті в плату Raspberry PI (клавіатура, миша і монітор вже підключені) і подаємо живлення через роз'єм micro - USB. Починається установка Raspbian, яка тривати близько 10 хвилин. В цей час від користувача практично нічого не потрібно окрім найпростіших і інтуїтивно зрозумілих дій, таких як вибір мови, введення пароля і тому подібне. На завершуючому етапі з'явиться меню, в якому можна вибрати тип призначеного для користувача інтерфейсу (консольний або графічний). Вибираємо графічний і завершуємо установку натисненням кнопки Finish. Система попросить перезавантажитися і як наслідок запуститися вже в привабливішому вигляді.

1.7 Аналіз схемотехнічних рішень плати Raspberry PI

Розглянемо окремі складові плати Raspberry PI на прикладі плат версій 3 та 4. Це найбільш потужніші версії плат та вони містять найбільшу кількість елементів вводу-виводу.

1.7.1 Структура портів GPIO

Для підключення до зовнішніх периферійних елементів використовується порт обміну GPIO.

Сам порт GPIO, як видно з рисунку 1.6, має порти з функцією SDA/SCL – обмін з пристроями за протоколом I2C.

Також наявні виводи з функціями MOSI – Master Out Slave In, MISO – Master In Slave Out, SCLK – Serial Clock. Ці виводи забезпечують обмін з пристроями, що включено ланцюгом – один за одним.

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		№докум.	Підпис	Дата		18

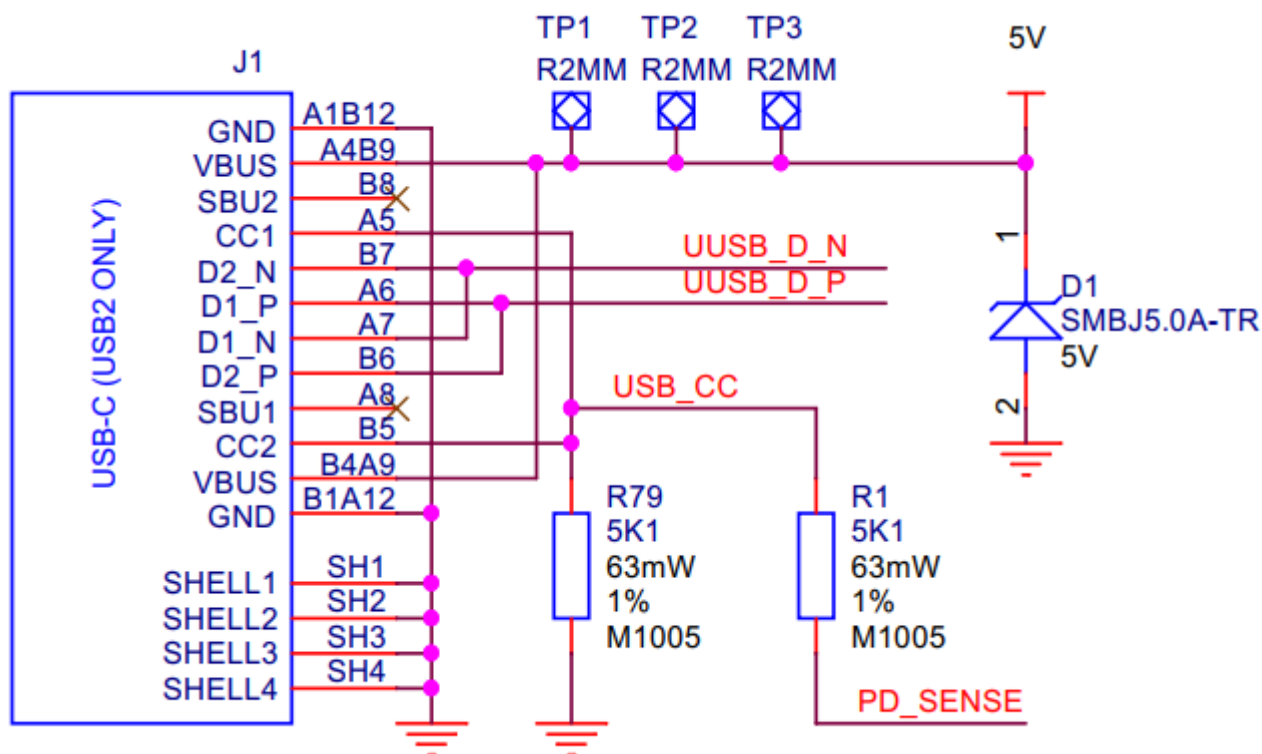


Рисунок 1.7 – Порт USB та живлення

1.7.3 Периферійне обладнання

Плата Raspberry PI дозволяє розширити можливості мікросистеми шляхом підключення такого звичного елемента як дисплей та камера. Використовуються стандартні цифрові інтерфейси підключення, тому плата надає типову схемотехніку.

Зм.	№докум.	Підпис	Дата	

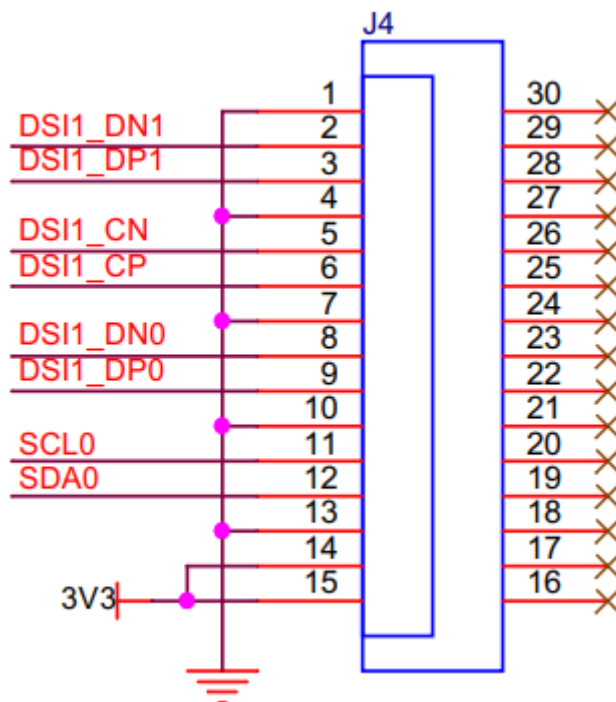


Рисунок 1.8 – Дисплей

Паралельний RGB-інтерфейс до 24 біт доступний на всіх платах Raspberry PI із 40-смуговим заголовком та обчислювальними модулями. Цей інтерфейс дозволяє приєднувати паралельні RGB-дисплеї до Raspberry PI GPIO у форматі RGB24 (8 біт для червоного, зеленого та синього) або RGB666 (6 біт для кожного кольору) або RGB565 (5 біт червоного, 6 зелених та 5 синіх).

Цей інтерфейс контролюється прошивкою графічного процесора і може бути запрограмований користувачем за допомогою спеціальних параметрів config.txt та ввімкнення правильного накладання дерева пристроїв Linux.

GPIO контакти

Однією з альтернативних функцій, яку можна вибрати на банку 0 Raspberry PI GPIO, є DPI (Display Parallel Interface), який є простим тактовим паралельним інтерфейсом (до 8 бітів R, G і B; годинник, увімкнути, hsync та vsync). Цей інтерфейс доступний як альтернативна функція 2 (ALT2) на GPIO-банку 0:

GPIO	ALT Func2
GPIO0	PCLK
GPIO1	DE
GPIO2	LCD_VSYNC
GPIO3	LCD_HSYNC
GPIO4	DPI_D0
GPIO5	DPI_D1
GPIO6	DPI_D2
GPIO7	DPI_D3
GPIO8	DPI_D4
GPIO9	DPI_D5
GPIO10	DPI_D6
GPIO11	DPI_D7
GPIO12	DPI_D8
GPIO13	DPI_D9
GPIO14	DPI_D10
GPIO15	DPI_D11
GPIO16	DPI_D12
GPIO17	DPI_D13
GPIO18	DPI_D14
GPIO19	DPI_D15
GPIO20	DPI_D16
GPIO21	DPI_D17
GPIO22	DPI_D18
GPIO23	DPI_D19
GPIO24	DPI_D20
GPIO25	DPI_D21
GPIO26	DPI_D22
GPIO27	DPI_D23

Рисунок 1.9 – Опис функцій на GPIO порту

Всі SoC, що використовуються в лінійці Raspberry PI, мають два інтерфейси камер, які підтримують джерела CSI-2 D-PHY 1.1 або CCP2 (Compact Camera Port 2). Цей інтерфейс відомий під кодовою назвою "Unicam". Перший екземпляр Unicam підтримує 2 смуги передачі даних CSI-2, тоді як другий підтримує 4. Кожна смуга може працювати зі швидкістю до 1 Гбіт / с (DDR, тому максимальна частота зв'язку становить 500 МГц).

Однак звичайні варіанти Raspberry PI виявляють лише другий екземпляр і спрямовують лише 2 смуги даних до роз'єму камери. Діапазон обчислювального модуля виводить усі смуги з обох периферійних пристроїв.

зображення. Його можна використовувати через MMAL, OpenMAX IL та V4L2 за допомогою модуля ядра bcm2835-v4l2. Через цей інтерфейс підтримуються лише камери Raspberry Pi.

Компонент MMAL rawcam

Це було тимчасовим варіантом до того, як був доступний драйвер V4L2. Компонент MMAL vs.ril.rawcam дозволяє отримувати необроблені дані CSI2 так само, як і драйвер V4L2, але вся конфігурація джерела повинна виконуватися користувачем через будь-який інтерфейс, який вимагає джерело. Додаток raspiraw доступний на github. Він використовує цей компонент та стандартні набори регістрів I2C для OV5647, IMX219 та ADV7282M для підтримки потокової передачі.

V4L2

Для блоку Unicam доступний повністю відкритий драйвер ядра; це модуль ядра під назвою bcm2835-unicam. Це інтерфейси до драйверів підпристроїв V4L2 для джерела для доставки необроблених кадрів. Цей драйвер bcm2835-unicam керує датчиком і налаштовує приймач CSI-2 таким чином, що периферійний пристрій запише необроблені кадри (після Debayer) в SDRAM для V4L2 для доставки в додатки. За винятком цієї можливості розпакувати формати CSI-2 Bayer до 16 біт / піксель, немає обробки зображень між джерелом зображення (наприклад, датчиком камери) та bcm2835-unicam, розміщуючи дані зображення в SDRAM.

Також плата містить на борту HDMI-вихід.

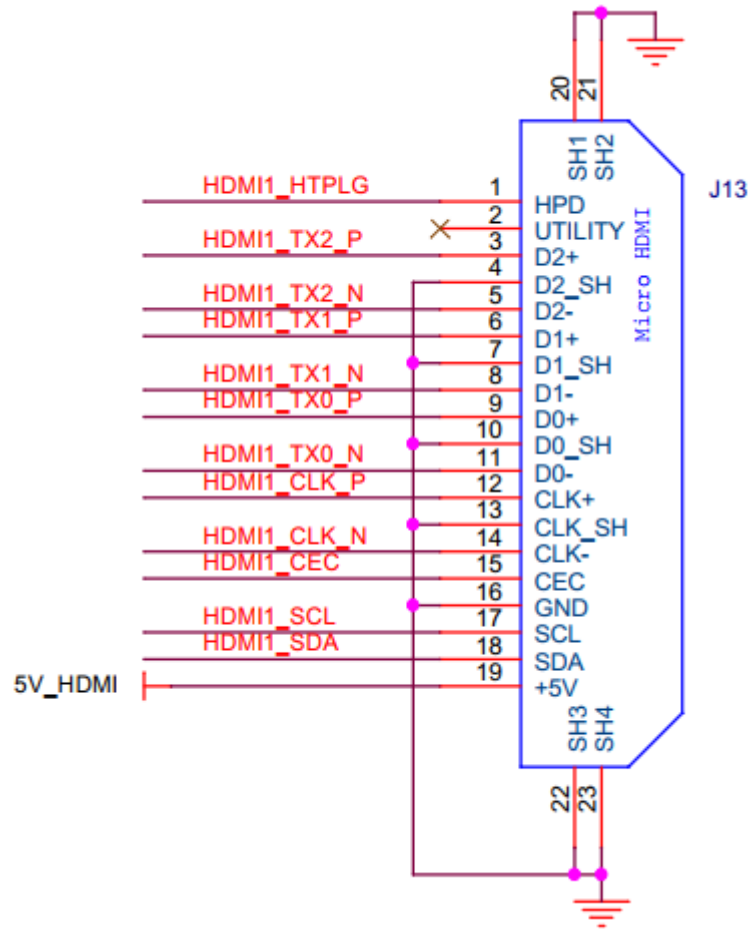


Рисунок 1.11 – HDMI-роз'язтя плати

1.7.4 Аудіо канали

Для роботи зі звуком, для його створення, Raspberry PI не використовує цифро-аналоговий перетворювач. Його не має. Проте функцію ЦАП виконує звичайний НЧ-фільтр, через який пропускається цифровий сигнал з широтно-імпульсною модуляцією.

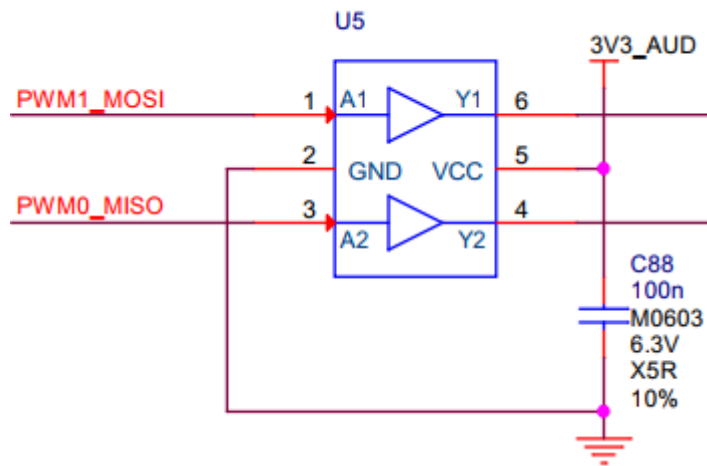


Рисунок 1.12 – Буферний підсилювач

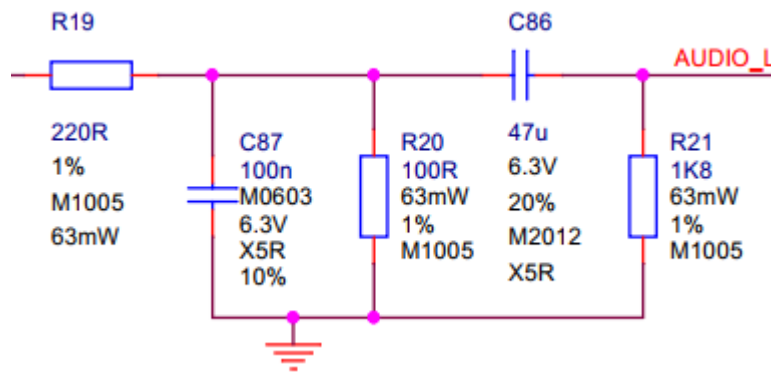


Рисунок 1.13 – Фільтр НЧ (Лівий канал)

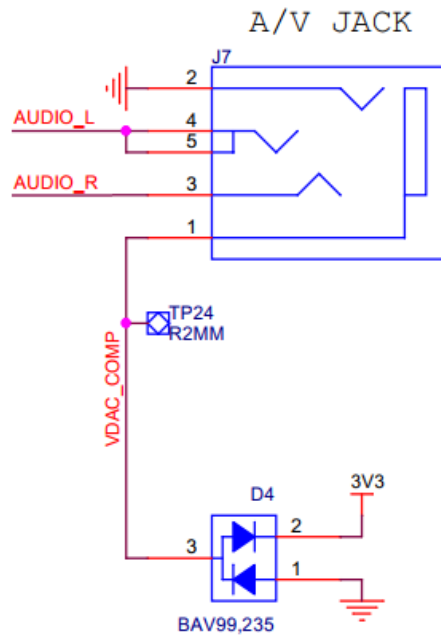


Рисунок 1.14 – Вихідне розняття

1.7.5 Джерело живлення 1.0 В

Живлення 1.0 В забезпечує живлення ядра системи. Для побудови джерела живлення використано одночіпний перетворювач U3, який містить необхідні елементи для побудови такого джерела живлення.

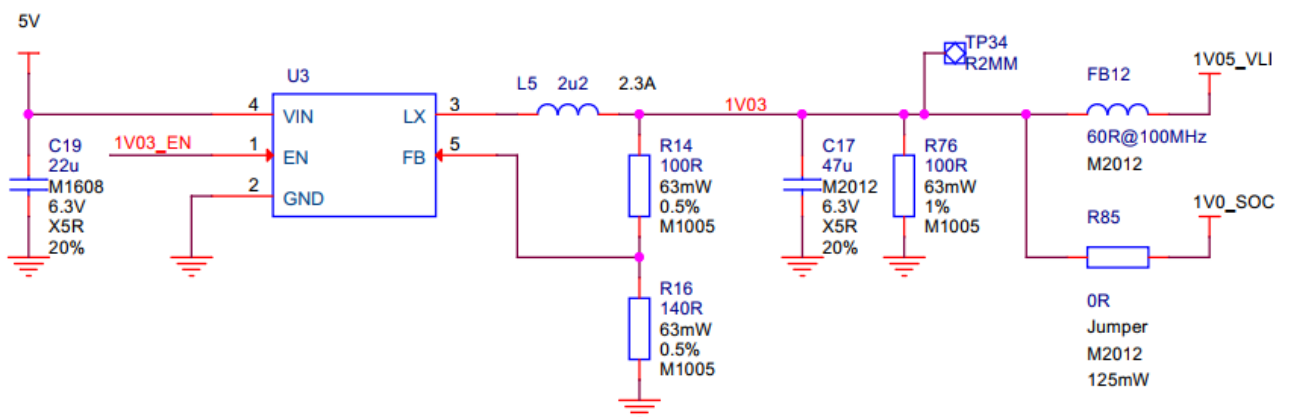


Рисунок 1.15 – Джерело живлення 1.0 В

оптичних і бездротових (радіо, наприклад) сигналів, а на другому формування кадрів (фреймів). І тут ми робимо висновок: Ethernet - це набір описів способів фізичної передачі сигналів (електрика) на першому рівні моделі OSI і формування кадрів (фреймів) на другому рівні моделі OSI всередині локальних мереж LAN.

1.7.7 Центральний процесор системи

BCM2711 – це мікросхема фірми Broadcom, яка використовується в моделі Raspberry PI 4 Модель В.

Архітектура BCM2711 є значним вдосконаленням у порівнянні з тією, що використовувалася SoC у попередніх моделях Pi. Він продовжує дизайн чотирьохядерних процесорів BCM2837, але використовує більш потужне ядро ARM A72. Він має значно вдосконалений набір функцій графічного процесора з набагато швидшим введенням / виведенням завдяки вбудованому з'єднанню PCIe, що з'єднує порти USB 2 та USB 3, та вбудованому контролеру Ethernet. Він також може адресувати більше пам'яті, ніж SoC, що використовувались раніше.

Ядра ARM здатні працювати на частоті до 1,5 ГГц, завдяки чому Pi 4 приблизно на 50% швидше, ніж Raspberry PI 3B +. Зараз новий блок VideoCore VI 3D працює на частоті до 500 МГц. Ядра ARM є 64-розрядними, і хоча VideoCore є 32-розрядним, є новий блок управління пам'яттю, що означає, що він може отримати доступ до більшої кількості пам'яті, ніж попередні версії.

Деякі технічні деталі

Процесор: чотириядерний Cortex-A72 (ARM v8) 64-бітний SoC @ 1,5 ГГц. Детальніше див. На сторінці Вікіпедії на A72.

Пам'ять: Доступ до 8 Гб LPDDR4-2400 SDRAM (залежно від моделі)

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		29

Кеші: 32 КБ даних + 48 КБ кеш-інструкцій L1 на ядро. 1 Мб кеш-пам'яті L2.

Мультимедіа: H.265 (декодування 4Кр60); H.264 (декодування 1080р60, кодування 1080р30); OpenGL ES, 3.0 графіка

I/O: шина PCIe, вбудований порт Ethernet, 2 × DSI-порти (лише один на Raspberry PI 4B), 2 × CSI-порти (лише один на Raspberry PI 4B), до 6 × I2C, до 6 × UART (в поєднанні з I2C), до 6 × SPI (лише п'ять виставляються на Raspberry PI 4B), подвійний вихід відео HDMI, композитний вихід відео.

Попередні версії Raspberry PI обладнанно такими чипами:

- Raspberry PI 3+ chip BCM2837B0
- Raspberry PI 3 chip BCM2837
- Raspberry PI 2 chip BCM2836
- Raspberry PI 1 chip BCM2835

1.7.8 Тактова частота системи, керування частотою та терморегулювання

Усі моделі Raspberry PI працюють у певному температурному режимі, щоб уникнути перегріву під великим навантаженням. SoC мають внутрішній датчик температури, програмне забезпечення на опитуваннях графічного процесора забезпечує перевищення температури заздалегідь визначеною межею – це 85 ° C на всіх моделях. Можна встановити це значення на більш низьке, але не на більш високе. Коли пристрій наближається до межі, різні частоти, а іноді і напруги, що використовуються на мікросхемі (ARM, GPU), зменшуються. Це зменшує кількість виробленого тепла, підтримуючи температуру під контролем.

Коли температура в ядрі знаходиться в діапазоні від 80 ° C до 85 ° C, відобразатиметься піктограма попередження, що показує червоний наполовину

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		30

заповнений термометр, а ядра ARM будуть поступово повертатися назад. Якщо температура досягає 85 ° С, відобразатиметься значок із повністю заповненим термометром, а ядра ARM і графічний процесор будуть зупинятись.

Для Raspberry PI 3 Model B + технологія друкованої плати була змінена для забезпечення кращого відведення тепла та збільшення теплової маси. Крім того, запроваджено м'яку температурну межу, метою якої є максимізація часу, протягом якого пристрій може «спринтувати», перш ніж досягати жорсткого обмеження при 85 ° С. При досягненні м'якого обмеження тактова частота зменшується з 1,4 ГГц до 1,2 ГГц, робоча напруга також трохи зменшується. Це зменшує швидкість підвищення температури. Фактично ми "стрибаємо" між коротким періодом на частоті 1,4 ГГц, а на довший період на 1,2 ГГц. За замовчуванням обмежене обмеження становить 60 ° С, і це можна змінити за допомогою налаштування temp_soft_limit у config.txt.

Raspberry PI 4 Model B продовжує використовувати ту саму технологію друкованих плат, що і Raspberry PI 3B +, щоб допомогти розсіювати надлишкове тепло. На даний момент не визначено м'яких обмежень.

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		31

Розділ 2 ЕЛЕКТРИЧНИЙ РОЗРАХУНОК ПРИСТРОЮ

Як вже говорилося раніше, основною відмінною рисою Raspberry Pi від звичайного ПК, являється наявність на платі портів загального призначення GPIO (General - purpose input/output). Користувачеві доступна можливість управління цими виводами, а це означає, що до Raspberry Pi можна підключати дисплеї, кнопки, датчики, реле і інші електронні модулі, якими можна маніпулювати на власний розсуд.

2.1 Робота з GPIO

Зовні GPIO виконаний у вигляді дворядної штирьової колодки з кроком 2,54мм, яка розташована на краю плати. Ранні моделі, такі як B і A містять 26 виводів, а сучасніші - 40. На малюнку №3 показаний зовнішній вигляд портів загального призначення для плати Raspberry Pi 3B+ з вказівкою нумерації виводів.



Рисунок 2.1 –Виводи GPIO на платі

					КПТР. 2017010.01.07 ПЗ			
		№докум.	Підпис	Дата				
Виконав	Мокрицький А.Б.		<i>[Signature]</i>		Проектування систем видаленого керування на основі міні комп'ютера Raspberry Pi Пояснювальна записка	Літера	Аркуш	Аркушів
Перевір.	Мясищев О.А.		<i>[Signature]</i>			у	32	
Т.Контр.						ТР-17-1, ФПКТС,		
Н.контр.								
Затвер.			<i>[Signature]</i>					

Але для того, щоб повноцінно використати GPIO, знати їх нумерацію недостатньо. Необхідно точно розуміти де розташований той або інший вивід, як він називається і за що він відповідає. На рисунку нижче приведена повна розпіновка роз'ємів GPIO для різних модифікацій Raspberry Pi.

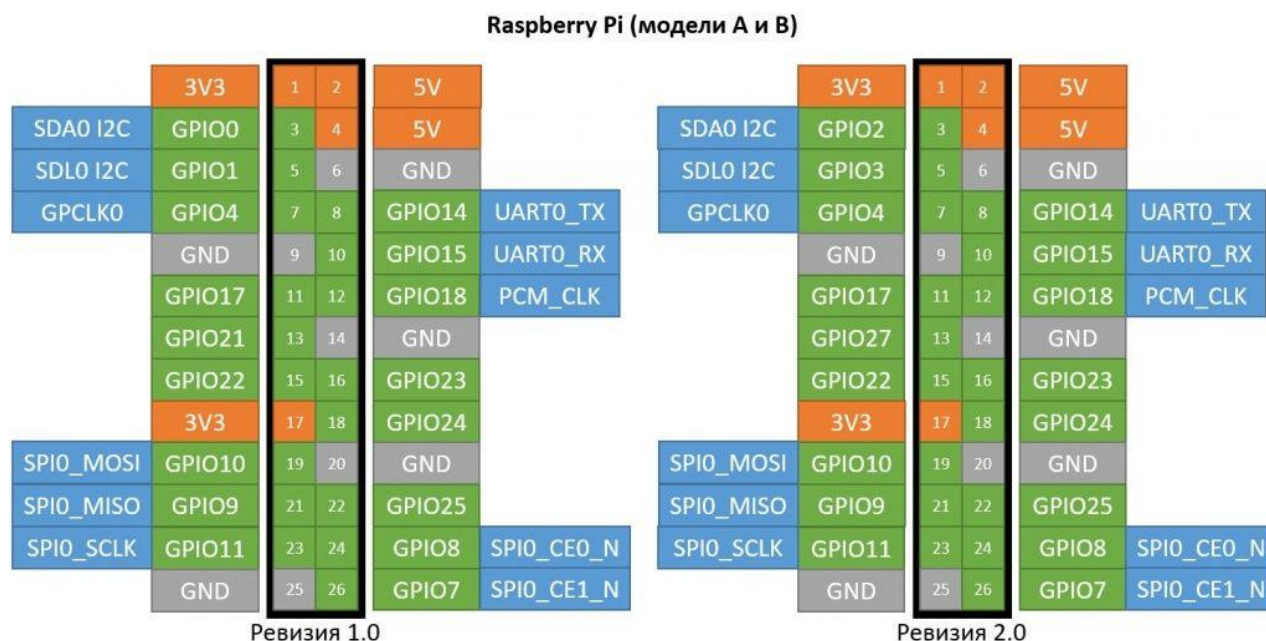


Рисунок 2.2 – Роз'язтя GPIO для різних модифікацій Raspberry Pi

Як видно з наведеного вище малюнка, на колодці окрім самих GPIO виведені штирки з напругою 3,3V, 5V, а також виводи GND. Деякі GPIO мають альтернативні функції, призначення яких вказане в синіх блоках. До того ж не можна порушувати здібності навантажень порту, щоб не вивести Raspberry Pi з ладу.

Слід пам'ятати, що GPIO працює з напругою 3,3V і максимальним струмом навантаження 50mA на один вивід. Це означає, що будь-яке перевищення вказаних параметрів негативно позначиться на працездатності

плати, тому краще використати гальванічну розв'язку між GPIO і зовнішнім виконавчим пристроєм.

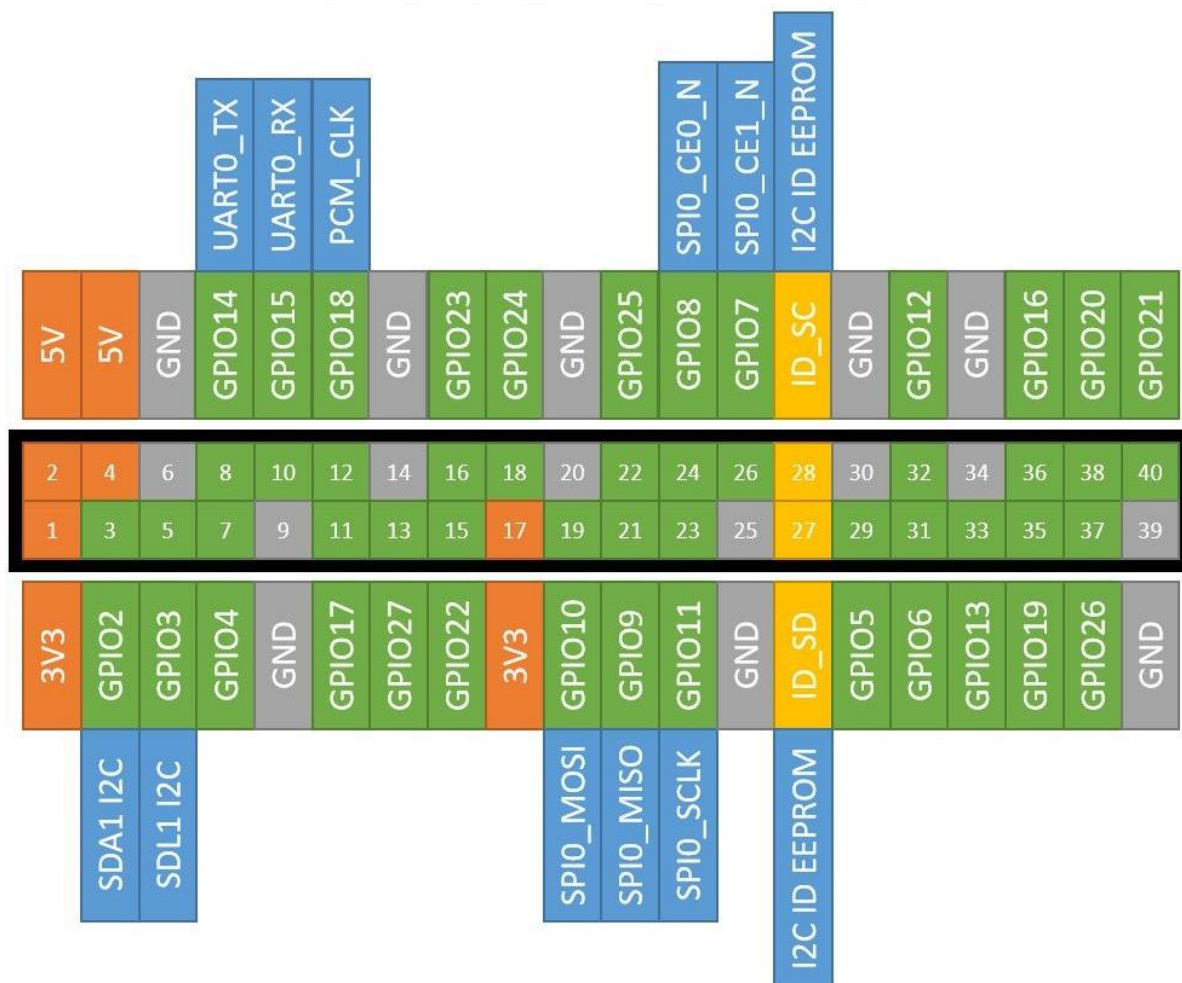
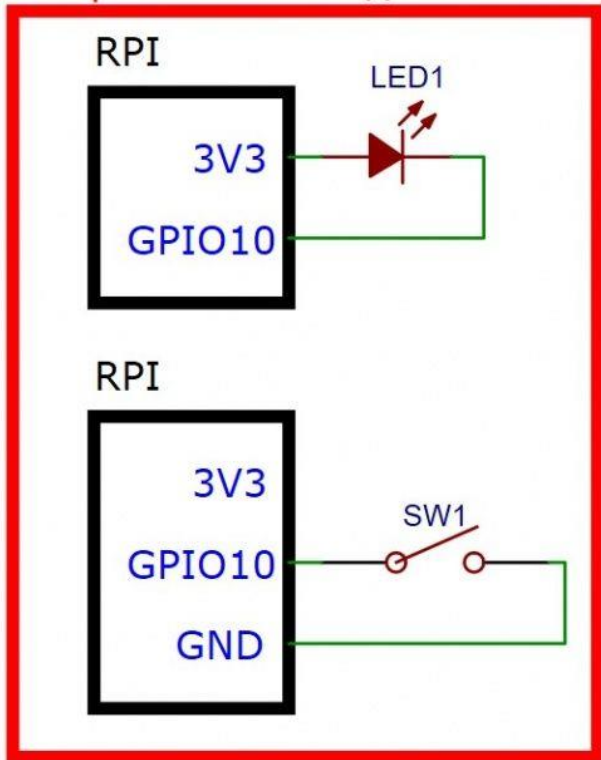


Рисунок 2.3 – Піни GPIO в платах Raspberry PI

Те ж саме відноситься і до вхідних ланцюгів, до яких застосовуються резистивні ділянки і всілякі перетворювачі рівнів. На рис. показаний приклад правильного і неправильного підключення базових елементів.

Неправильное подключение



Правильное подключение

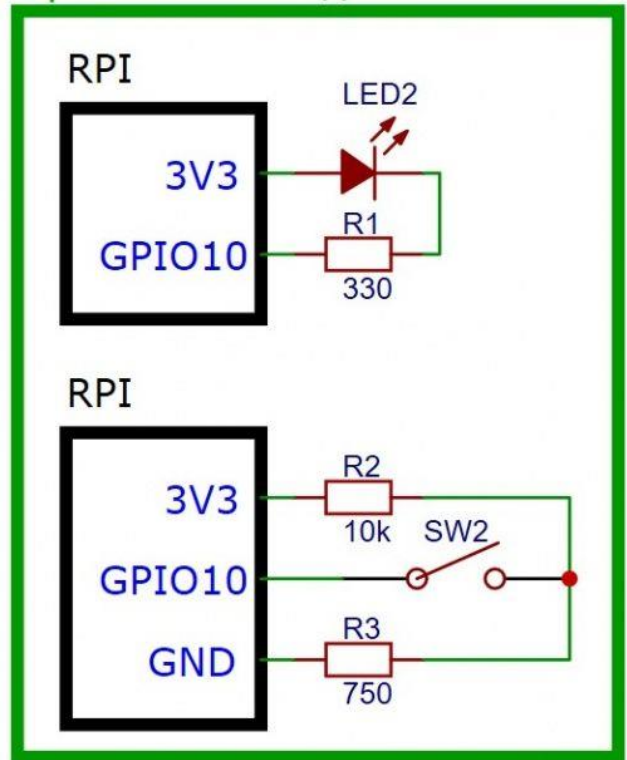


Рисунок 2.4 – Приклади правильного і неправильного підключення навантаження

У лівій частині малюнка пряме підключення світлодіода приведе до перевищення максимально допустимого струму 50mA. Це, у свою чергу, виведе GPIO10 з ладу. У правій частині малюнка доданий обмежувальний резистор, який утримуватиме струм в допустимих рамках. Що стосується кнопки, то може виникнути ситуація, коли GPIO10 помилково буде конфігурований на вихід, і її натиснення приведе до прямого з'єднання 3,3V і GND. При додаванні резисторів R2 і R3 усі виводи будуть гарантовано захищені від перевантажень. Виходячи з вищевикладеного можна зробити висновок, що є дріб'язковий в елементах захисту не коштує, оскільки працездатність Raspberry PI набагато важливіша за купку дешевих резисторів.

Зм.	№докум.	Підпис	Дата	

2.2 Програмування GPIO

Операційна система Raspbian пропонує користувачам зручний модуль для програмного управління GPIO. Називається він RPi.GPIO і є стандартним застосуванням. Перед його застосуванням, модуль рекомендується відновити. Зробити це можна набравши в консолі наступні рядки:

```
sudo apt-get update  
sudo apt-get install python-rpi.gpio
```

Щоб мати практичне уявлення про роботу з GPIO, створимо невеликий проект, який змусить Raspberry PI блимати світлодіодом один раз в секунду, а при натисненні на кнопку збільшувати частоту мигання в 5 разів. Схема майбутнього проекту показана на рисунку.

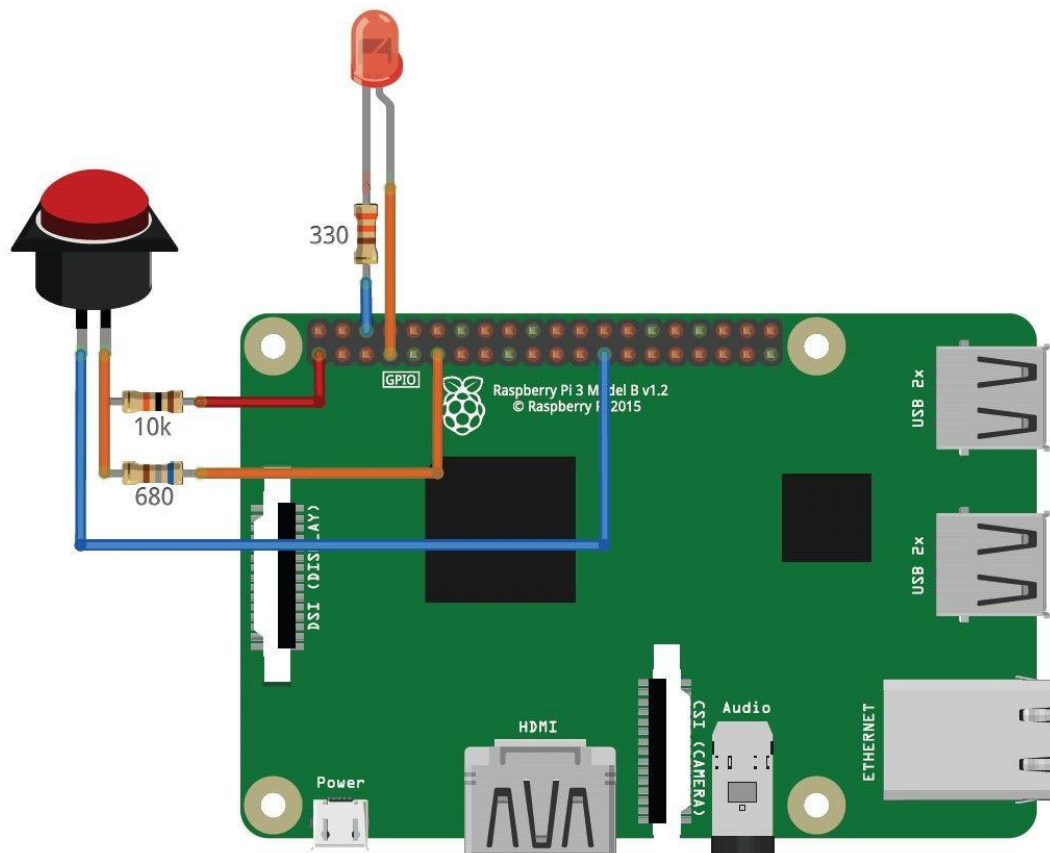


Рисунок 2.5 –схема управління світлодіодом

Зм.	Нодокум.	Підпис	Дата	

За управління світлодіодом відповідатиме GPIO4, а за читання стану кнопки GPIO17.

По класиці, програми для Raspberry PI пишуться скриптовою мовою програмування Python. Особливість його в тому, що для запуску програми не потрібно компілятор. Скрипт запускається і починає роботу відразу, але його необхідно зберегти у файл з подальшим завантаженням в плату. Для цього відкриваємо термінал і прописуємо наступний рядок:

```
nano /home/pi/led_key_test.py
```

Тим самим ми створюємо файл «led_key_test.py» в директорії «/home/pi». Як наслідок відкриється редактор, в якого необхідно написати нижченаведений код.

```
#!/usr/bin/env python
# -*- coding: utf - 8 -* -
# Підключення бібліотек для роботи з GPIO і організації затримок за
часом
import time
import RPi.GPIO as GPIO

# Визначення виводів GPIO, до яких підключені світлодіод і кнопка
LED = 4
KEY = 17

# Скидання портів (усі виводи налаштовуються на вхід - INPUT)
GPIO.cleanup()
# Режим нумерації пінів - по назві (не по порядковому номеру на
колотці)
GPIO.setmode(GPIO.BCM)
# Налаштування піну LED на вихід (OUTPUT)
GPIO.setup(LED, GPIO.OUT)
# Установка логічного (0) на виводі LED
GPIO.output(LED, GPIO.LOW)
# Налаштування піну KEY на вхід (INPUT)
GPIO.setup(KEY, GPIO.IN)
# Виведення вітання на екран
print 'Hello Raspberry PI'
```

```

# Перевірка на переривання програми по натисненню (CTRL+C) на
квіатури
try:
    # Нескінченний цикл
    while True:
        # Якщо кнопка натиснута (на піну KEY логічний 0)
        if GPIO.input(KEY) == False:
            # Виставляємо затримку 0,1 сек. і виводимо повідомлення
            timeout = 0.1
            print 'Key is pressed.'
        else:
            # інакше затримка - 0,5 сек.
            timeout = 0.5
        # Включаємо світлодіод
        GPIO.output(LED, GPIO.HIGH)
        # Затримка
        time.sleep(timeout)
        # Гасимо світлодіод
        GPIO.output(LED, GPIO.LOW)
        time.sleep(timeout)
# Якщо CTRL+C була натиснута - скидаємо порт і завершуємо виконання
програми
except KeyboardInterrupt:
    GPIO.cleanup()

```

Ще однією важливою особливістю програми Python є суворе дотримання відступів (табуляцій) при написанні програм. Враховуйте це правило при створенні свого коду.

Отже, переходимо до останнього етапу. Щоб вийти з редактора тиснемо CTRL+X і зберігаємо програму натисненням «у» + ENTER. Залишилося тільки зробити скрипт виконуваним. Для цього вводимо в терміналі наступні рядки:

```

chmod +x /home/pi/led_key_test.py
/home/pi/led_key_test.py

```

Бачимо на екрані вітання і миготливий 1 раз в секунду світлодіод. Тепер перевіряємо працездатність кнопки, і якщо усе зроблено правильно, то при її натисненні частота мигання зростає і повернується до колишньої при її відпуску.

2.3 Перспективи використання Raspberry PI

Плати Raspberry PI виявилися надзвичайно успішними і корисними, особливо для початкуючих молодих інженерів, але існує відразу декілька причин того, чому вони не підходять для реалізації професійних проектів.

У проведених опитуваннях близько 20% респондентів заявили про те, що використовують вбудовувані плати від різних виробників. Застосовуються такі плати, у тому числі Raspberry PI, в якості складових елементів в складніших пристроях. Серед заявлених застосувань були відмічені і професійні проекти.

Попри те, що у багатьох випадках вбудовувані плати стають важливим інструментом, наприклад, при STEM –навчанні(science, technology, engineering and mathematics) і при створенні прототипів, я виділю деякі недоліки цих плат, які заважають їх використанню в комерційних і промислових застосуваннях. Як приклад розглядатимуться одноплатні комп'ютери Raspberry PI.

Без сумніву, плати Raspberry PI є прикладом хорошої реалізації відмінної ідеї. Ці одноплатні комп'ютери ідеально підходять для навчання і прототипування. Існують деякі недоліки, які можуть завадити використанню Raspberry PI в комерційних продуктах, особливо якщо в цих продуктах Raspberry PI виступає основним обчислювальним модулем.



Рисунок 2.6 –Зовнішній вигляд Raspberry PI

2.3.1 Чому вибирають Raspberry PI?

Давайте спочатку спробуємо зрозуміти, чому багато інженерів використовують Raspberry PI у своїх проектах.

У Raspberry PI є безліч переваг :

- Низька вартість (~ 35 \$);
- Величезна обчислювальна потужність при компактних габаритах;
- Підтримка різних інтерфейсів (HDMI, USB, Ethernet, вбудовані Wi - Fi і Bluetooth, безліч GPIO, живлення від USB і т. д.);
- Підтримка Linux, Python (полегшує створення додатків);

Зм.	№докум.	Підпис	Дата	

- Безліч готових прикладів і підтримка співтовариства розробників;
- Мінімальна вартість розробки кінцевого пристрою.

Коли йдеться про хобі або створення невеликого проекту для власних потреб, усі перераховані пункти, виявляються дуже до речі. Але якщо ви створюєте промисловий або комерційний пристрій, то треба подумати про безліч інших важливих параметрів, перш ніж використати яку-небудь готову вбудовувану плату. Отже, про що слід згадати, перш ніж вибрати вбудовану плату для свого проекту?

2.3.2 Тривалість життєвого циклу та обрання системи на чіпові

Доступність компонентів, є найбільш важливим чинником, який необхідно враховувати при створенні більшості проектів. В середньому, розробка комерційного застосування, займає близько 18-24 місяців, і тільки після цього продукт може бути випущений на ринок.

Як правило, тільки після трьох-чотирьох років успішних продажів розробник планує модернізацію і перехід на сучасніший одноплатний комп'ютер. Отже, якщо ви розглядаєте можливість використання готової вбудовуваної плати, то, на мою думку, тривалість її життєвого циклу повинна складати сім і більше років.

Тепер давайте перевіримо плани випуску Raspberry PI, опубліковані на офіційному сайті :

- Raspberry PI 3 Model B + наявність, до січня 2023 року;
- Raspberry PI 3 Model A + наявність, до січня 2023 р.;
- Raspberry PI Zero, до січня 2022 р..

При цьому немає гарантії, що потрібна Вам кількість плат буде доступна у будь-який слушний час. Таким чином, перед Вами як перед розробником

встає питання: чи слід мені вибрати вбудовувану плату, яка буде доступна тільки впродовж наступних трьох-чотирьох років? На додаток до усього, ви повинні подумати про те, як забезпечити безперебійне постачання необхідної кількості плат в мить, коли вони вам знадобляться?

2.3.3 Запуск операційної системи (ОС) на SD -карті

Raspberry PI працює під управлінням операційної системи Linux, що запускається з SD -карти. Це означає, що у додатків з Raspberry PI можуть виникнути проблеми з вібрацією, оскільки із-за механічних коливань електричний контакт з картою пам'яті може порушуватися. При цьому плата не має вбудованих способів боротьби з такими ситуаціями. В якості захисту від випадання карти пам'яті можна використати обхідні шляхи (двосторонній скотч, клей і т. д.), які, втім, не можна вважати досить надійними.



Рисунок 2.7 – SD -карта

Зм.		№докум.	Підпис	Дата

Інша проблема з SD-картою полягає в тому, що вона може бути пошкоджена, наприклад, якщо живлення плати було відключене в процесі запису файлу. Для вирішення цієї проблеми знадобиться зовнішня батарея або джерело живлення на основі суперконденсатора. За наявності батареї операційна система встигне безпечно завершити роботу після виявлення виключення основного джерела живлення.

2.3.4 Наявність USB-інтерфейсів

На платах Raspberry PI є присутніми традиційні USB-роз'язки, але штирьові роз'язки з виведеними лініями USB -інтерфейсів відсутні, що обмежує можливості підключення датчиків, модемів і інших пристроїв. Фактично розробники можуть підключати зовнішні USB-пристрої тільки за допомогою USB-кабелів, що є не кращим варіантом для промислових застосувань.

Незважаючи на великі можливості, в платах Raspberry PI відсутні багато важливих функцій, які потрібні для створення комерційних продуктів.

У більшості випадків, якщо ви хочете контролювати події в режимі реального часу, вам знадобиться годинник реального часу (RTC). На жаль, в платах Raspberry PI цей важливий елемент відсутній. Ви можете самостійно додати зовнішню плату розширення з RTC, але це зажадає додаткових витрат часу і засобів. Далі приведемо лише невеликий перелік функцій/ інтерфейсів, які можуть Вам знадобитися, але які відсутні в платах Raspberry PI.

1) АЦП: якщо у вашому застосуванні потрібно АЦП (з дозволом 8/10/12 біт), вам потрібно буде підключити додатковий зовнішній модуль АЦП, наприклад, по I2C або SPI.

2) EEPROM/ FRAM/ SPI Flash : у багатьох застосуваннях дуже важливо мати вбудовану енергонезалежну пам'ять (EEPROM або FRAM або Flash) для зберігання даних, що поступають. Нині у Raspberry PI такої можливості немає.

3) UART: в платах Raspberry PI на штирьовий роз'єм виведений тільки один UART. Зазвичай для додатків вимагається два або три UART.

Обмежена кількість входів/ виходів : в платах Raspberry PI на штирьове роз'єм виведений 28 GPIO, але цього недостатньо для багатьох комерційних застосувань.

Із-за перерахованих недоліків плат Raspberry PI користувачеві може знадобитися підключення додаткової плати (чи плат) розширення.

2.3.5 Зовнішнє джерело живлення?

При створенні комерційних пристроїв на базі Raspberry PI Вам знадобиться додаткове джерело живлення. Це може бути АС - DC -джерело (якщо пристрій живиться від мережі) або DC - DC -перетворювач (якщо пристрій живиться від акумулятора або зовнішнього адаптера 6/ 12/ 24 В).

Для живлення слід використати надійний роз'єм. На жаль, роз'єм мікро-USB не рекомендується для застосування в комерційних застосуваннях. Крім того, щоб захистити Raspberry PI від збоїв, необхідно, щоб джерело живлення мало вбудований захист від зворотної напруги, викидів напруги і перевантаження по струму.

2.3.6 Проблема відведення тепла

Процесор Raspberry PI виділяє досить багато тепла, яке необхідно відводити, щоб уникнути проблем з перегріванням і надійністю. На даний момент розробники можуть використати тільки невеликий радіатор, який

передбачається встановлювати за допомогою клею. Спеціалізовані монтажні отвори для надійного кріплення радіатора поблизу процесора відсутні.

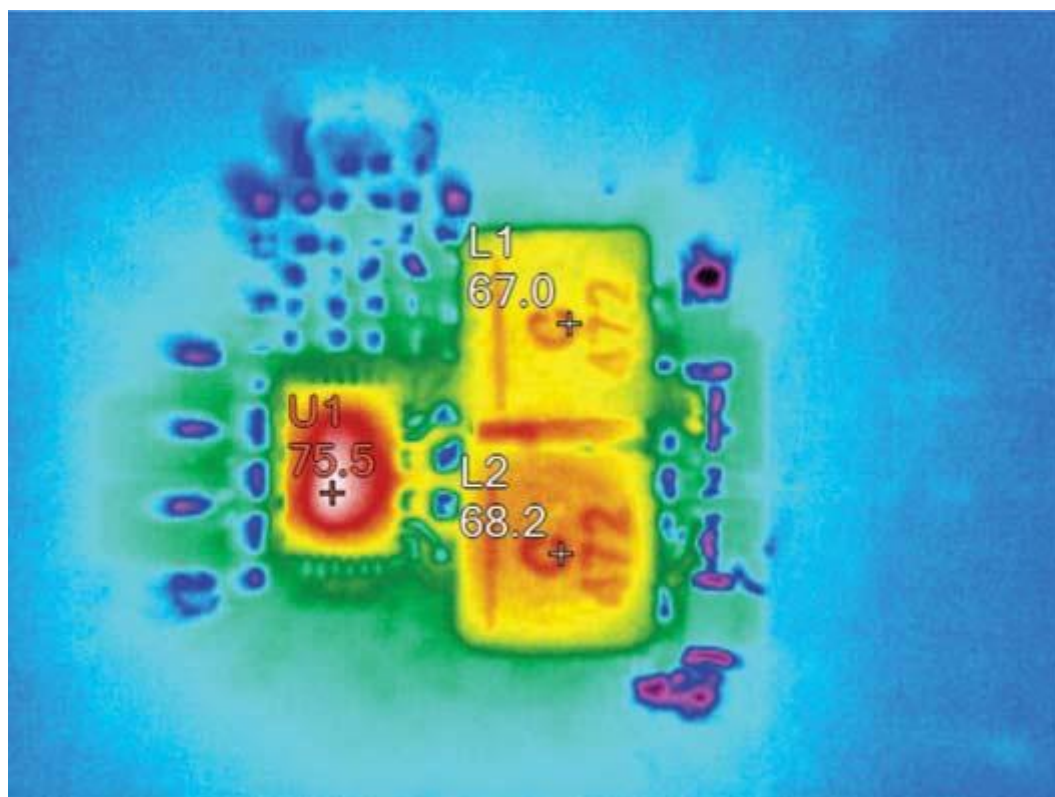


Рисунок 2.8 – Тепловий режим роботи схеми від ADI

Кінець кінцем, якщо ви використовуєте обчислювальну потужність Raspberry Pi на 70-90%, то вам обов'язково знадобиться потужний радіатор, який не завжди є в наявності. Крім того, для кріплення доведеться використати монтажні отвори, розташовані по краях плати. В результаті може знадобитися виготовлення радіатора по індивідуальному замовленню, а це знову-таки додаткові витрати часу і засобів.

Плати Raspberry Pi мають чудову підтримку у вигляді онлайн-форумів і різних співтовариств користувачів. Проте, потужна підтримка з боку співтовариств зовсім не є підтримкою з боку професійних розробників. Чи

зможете ви запустити продукт в стислі терміни, використовуючи підтримку співтовариства, або вам знадобиться спеціальна професійна підтримка?

2.3.7 Потреби масштабованості

Плати Raspberry PI не орієнтовані на комерційні застосування і представлені у вигляді набору окремих модулів, а не у вигляді впорядкованої уніфікованої лінійки моделей, що відрізняються обчислювальною потужністю. Іншими словами, якщо Вам потрібно буде збільшити продуктивність, то Ви не зможете вибрати потужнішу плату з тим же форм-фактором. Замість цього Вам потрібно буде переробляти увесь конструктив.

Що Ви робитимете, якщо після закінчення розробки Ви раптом зрозуміли, що у споживачів є попит на інші специфікації пристрою? Надзвичайно важливо враховувати можливості масштабування і розширення функціонала при переході від прототипів до реальних серійних моделей.

В деяких випадках можливостей, Raspberry PI, що надаються, буде більш ніж достатньо. Raspberry PI можна сміливо використати для розробки, у тому числі і комерційних пристроїв.

Розділ 3 ПРОЄКТУВАННЯ СХЕМИ ПРИСТРОЮ

3.1 Існуючі рішення щодо застосування Raspberry PI

3.1.1 Плата розширення портів введення / виводу Raspberry PI відкриває можливості навігації та автоматизації

Плата розширення RIO (для портів введення / виводу одноплатного комп'ютера Raspberry) компанії Roboteq, що встановлюється над Raspberry, призначена для створення потужного вбудованого керування та навігації комп'ютера, має малі розміри, всього лише 65x85x20 мм.

RIO значно розширює обмежені можливості введення / виведення Raspberry, додаючи 13 вхідних портів, які можна конфігурувати як цифрові входи, як аналогові входи напругою 0-5 В з 12-бітовим дозволом або як імпульсні входи з налаштуванням ширини імпульсу, коефіцієнта заповнення імпульсу або захоплення частоти. Вісім цифрових виходів призначені для управління навантаженням до 1 А на кожен вихід при напрузі до 24 В.

RIO включає 32-бітний мікропроцесорний контролер ARM Cortex M4 для обробки і буферизації портів введення / виводу, і забезпечення безперебійної зв'язку з комп'ютером Raspberry. RIO процесор може програмуватися користувачем за допомогою простого, але потужного мови програмування на базі Бейсика, для виконання логічних, що погоджують і інших операцій введення / виводу в режимі реального часу, знижуючи навантаження на комп'ютер Raspberry при виконанні таких операцій. Для операційної системи Linux RIO поставляється з драйверами і функціональної бібліотекою для швидкої конфігурації і доступу до портів введення / виводу, а також безперебійного обміну даними з Raspberry.

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		Нодокум.	Підпис	Дата		47

Плата RIO також має кілька комунікаційних інтерфейсів, включаючи послідовний порт RS232 для зв'язку зі стандартними послідовними прилаштувати, послідовний порт TTL для зв'язку з Arduino та іншими мікроконтроллерами, необладнаних RS232 трансиверами, а також CANbus інтерфейс.



Рисунок 3.1 – Плата розширення RIO

Версія плати RIO доступна з датчиками руху, включаючи 3-осьовий акселерометр, 3-осьовий гіроскоп, 3-осьовий магнетометр і алгоритм fusion для створення курсової навігаційної системи AHRS. На додаток до великого асортименту контролерів електродвигуна Roboteq, RIO-AHRS відкриває нові можливості застосування в автоматичних робототехнічних системах, які можуть працювати в морі, на землі або в повітрі. Плата RIO включає DCDC перетворювач для живлення Raspberry, RIO і зовнішніх датчиків, напругою від 8 до 30 В постійного струму.

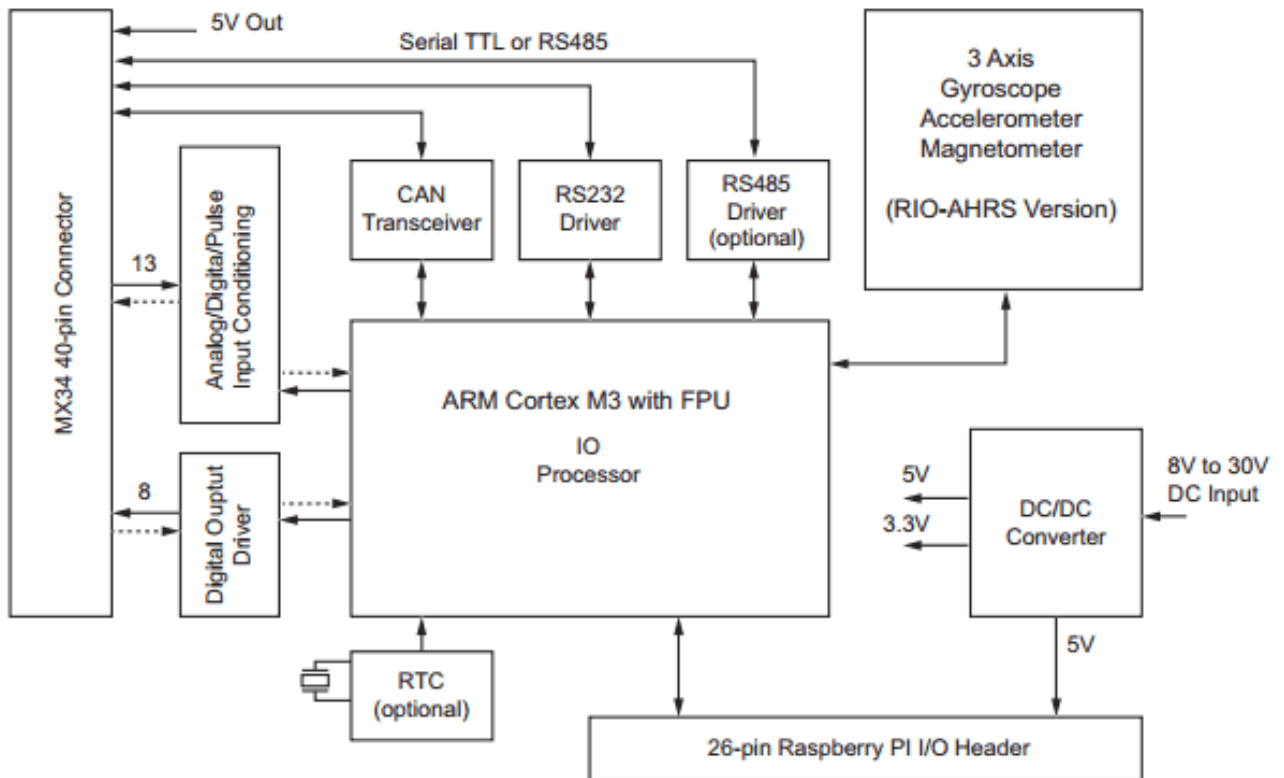


Рисунок 3.2 – Структурна схема плати RIO

Вже згадана нами система домашньої автоматизації, як ви пам'ятаєте з попередніх публікацій, складається з двох основних сегментів - централізованого і розподіленого. Розподілений сегмент - це контролери, рознесені територіально і підключаються до Raspberry PI через інтерфейс RS485. Централізований сегмент - це релейні модулі, модулі дискретних входів, годинник реального часу, різні датчики температури, тиску і т.д, що підключаються безпосередньо до портів GPIO Raspberry PI.

Так як компоненти централізованого сегмента системи автоматизації згруповані фактично в одному місці, то для них була розроблена об'єднавча плата (крос-плата), що дозволило виконати всі «міжблочні» з'єднання цього сегмента на рівні друкованих провідників. Частина компонентів розміщується безпосередньо на крос-плати, а інша частина в вигляді окремих модулів підключається через роз'ємні з'єднання.

До складу крос-плати входять:

- Роз'єм 2x20 для підключення портів GPIO Raspberry B + (або Pi 2);
- Блок живлення 5В / 2А на імпульсному стабілізаторі LM2576 для харчування Raspberry PI і деяких датчиків. Наявність вихідної напруги 5В відображається світлодіодом VD4;
- Вихід для підключення в буферному режимі резервного акумулятора;
- 8 гальванічно розв'язаних дискретних входів, виконаних на оптронах PC817;
- 8 релейних виходів, реалізованих на двох чотирьохканальних релейних модулях. Конструкція плати дозволяє використовувати як модулі з харчуванням реле 5В, так і модулі з харчуванням 12В. Вибір напруги харчування проводиться за допомогою перемичок JM1-JM2 і JM3-JM4;
- Датчик атмосферного тиску і температури BMP085 (BMP180), що працює по шині I2C;
- Годинник реального часу, виконані на мікросхемі DS1307, що працюють по шині I2C;
- клемних роз'єм для підключення шини 1-wire;
- клемних роз'єм для підключення шини I2C.

Крім того, на платі передбачені штирові роз'єми для підключення додаткових пристроїв через UART і SPI.

Raspberry PI, релейні модулі, датчик тиску і температури BMP180 підключаються до крос-плати через роз'ємні з'єднання. Для надійності і додання зафіксованого стану, що встановлюються на плату компоненти і модулі кріпляться до неї через невеликі монтажні стійки з різьбленням M2,5.

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		50

Перед тим як почати розглядати установку програмного забезпечення для системи домашньої автоматизації, кілька слів про апаратну частину проекту. Розробка апаратного забезпечення (крос-плати) для системи домашньої автоматизації була викликана прагненням упорядкувати розрізнені елементи централізованого сегмента в єдине ціле і в той же час зробити його і окремим пристроєм, і «ядром» всієї системи. Крім того, це пов'язано і з ідеологією побудови самої системи - для невеликих об'єктів цілком може бути достатньо тільки централізованого сегмента, а для великих об'єктів система додатково «посилюється» розподіленим сегментом - тобто мережею віддалених контролерів, що підключаються по шині RS485 або по радіоканалу.

Розміщення на сайті раніше матеріали дозволяють в повній мірі «вдихнути життя» в крос-плату, але для повної реалізації її можливостей розроблена спеціалізована програмна збірка. Поряд з напрацюваннями, про які вже розповідалося, так само передбачається можливість створення користувачем нескладних сценаріїв через інтерфейс розширеного налаштування - тобто без втручання у внутрішні програмні коди виконуваних скриптів і алгоритмів.

На поточний момент на базі крос-плати реалізовані і підтримуються наступні функції:

- управління 8 релейними виходами;
- контроль 8 дискретних входів;
- шина 1-wire для підключення DS18B20;
- шина I2C для підключення DS1307, BMP085; TSL2561, SHT21, MPL115A2, MMA7660
- два входи для підключення датчиків температури і вологості DHT22;

- можливість реалізації на реле №1 і №2 двох незалежних термостатів по командам датчиків температури (з точністю до десятих часток градуса як в області позитивних, так і негативних температур);

- можливість реалізації на реле №3 і №4 двох незалежних добових таймерів (крок установки - 1 хвилина);

- можливість реалізації на реле №5 вимірювача освітленості (люксметра);

- можливість реалізації на реле №7 і №8 двох незалежних каналів управління по зміні стану входів №7 і №8;

- відправку повідомлень на e-mail при замиканні / розмиканні будь-якого з дискретних входів;

- побудова графіків змін в часі параметрів, одержуваних від датчиків;

- ведення журналів (ретроспективи) подій на входах і виходах з часу і з «конкретизацією» причин подій, а так само можливість віддаленого перегляду і очищення журналів;

- самостійне конфігурація користувачем системи через web-інтерфейс (створення описаних вище сценаріїв) безпосередньо в процесі роботи без перезавантаження сервера.

В принципі, для невеликого обсягу автоматизації зазначених функцій цілком достатньо. Однак, можливості системи можна значно розширити за допомогою контролерів, що підключаються по шині RS485 (це так званий розподілений сегмент системи). Використовуючи ці контролери можна організувати додатковий збір свідчень температури і вологості з датчиків DHT11 і DHT22, бездротове управління кондиціонером і освітленням, стан бездротових датчиків, управління додатковими навантаженнями і контроль

додаткових дискретних входів. Всю необхідну інформацію для самостійної збірки даних контролерів, а так само їх опис, ви можете знайти тут.

Як можна помітити, в порівнянні з попередньою версією, на поточний момент система домашньої автоматизації отримала багато додаткових функцій. Наприклад, можливість організації охоронної та аварійно-попереджувальної сигналізації з оповіщенням на e-mail (відповідно, після отримання e-mail можна налаштувати повідомлення про це на SMS - багато стільникові оператори підтримують таку функцію). Термостати, таймери і люксометр дозволять реалізувати сценарії управління навантаженнями по температурі, за встановленим часом і по освітленості.

Далі розглянемо web-інтерфейс. Зазначу, що всі web-сторінки самомасштабуючі - тобто автоматично підлаштовуються під розмір вікна браузера. На головній сторінці передбачені три додаткові вкладки - «Метеостанція», «Радіокерування» і «Виходи і входи GPIO» з кнопками їх розгортання і згортання. Це дозволяє оперативним чином перемикатися між вкладками і в той же час не захищувати сторінку зайвою інформацією. За замовчуванням при відкритті web-інтерфейсу всі вкладки згорнуті. Періодично змінюють колір смужки (червоний / зелений) показують, що йде обмін даними по шині RS485 Raspberry PI з метеостанцією і радіомодулем.

3.2 Розробка крос-плати для Raspberry PI

Для створення крос-плати обрано вирішення задачі забезпечення більшої потужності вихідних сигналів. А також забезпечення надійності введення цифрових сигналів широкого діапазону напруг.

3.2.1 Вхідний каскад

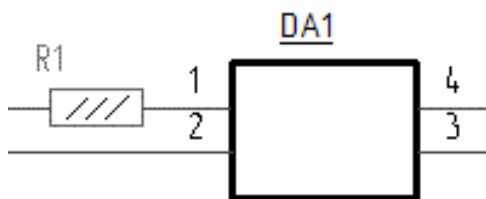


Рисунок 3.3 – Схема вхідного каскаду введення

На рис. 3.3 показано фрагмент схеми для введення вхідного сигналу. DA1 – оптопара типу PC817.

Оптопара PC817 складається з оптодіоду та фототранзистора. Оптична розв'язка дозволяє розділити вхід та вихід гальванічно. Якщо на вхід попаде велика напруга, що може зруйнувати оптопару PC817, то в такому випадку вихідна частина залишиться непошкодженою.

Резистор R1 обирається за умов обмеження струму, що йде на оптодіод. Для цієї оптопари струм обмежується на рівні 5 мА.

3.2.2 Джерело живлення

Для живлення використовується перетворювач напруги LM2576. LM2576 є на фіксовану вихідну напругу 3.3В, 5В, 12В, 15, а також з регульованим вихідним напругою. У регульованій версії вихідна напруга змінюється від 1.23В до 37В, а у LM2576HV до 57 Вольт.

Вхідна напруга може досягати 40В, а у LM2576HV до 60В. Максимальний вихідний струм 3 А. Температура, яку може витримати кристал, становить 150 градусів Цельсія.

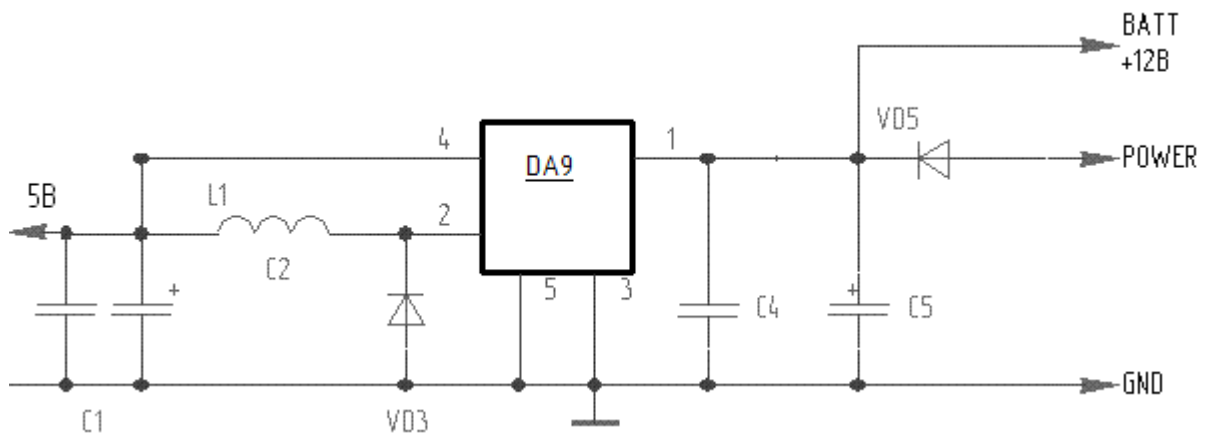


Рисунок 3.4 – Стабілізатор живлення

3.2.3 Релейний модуль

Для роботи у якості вихідного елемента використовуються готові модулі на 4 канали кожний.

Релейний модуль

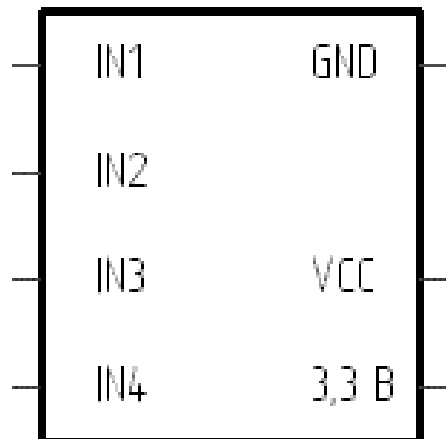


Рисунок 3.5 – Модуль на 4 виходи

Застосування реле дозволяє відділити навантаження від керуючих виходів.

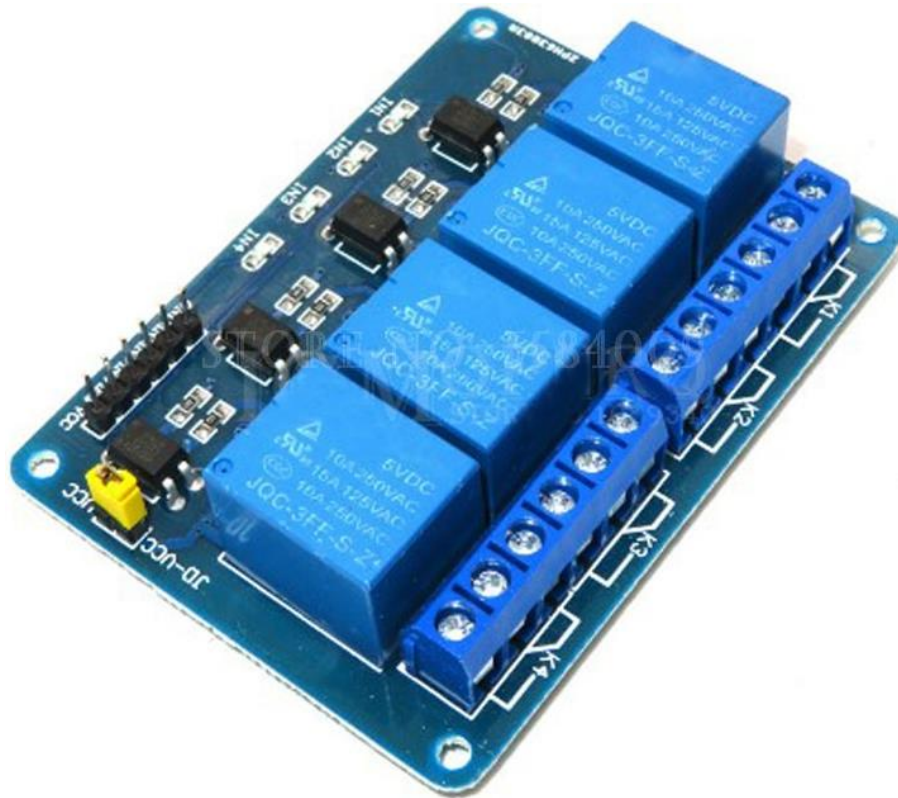


Рисунок 3.6 – Зовнішній вигляд релейного модуля

3.3 Програмна частина системи контролю та віддаленого керування

Отже, представлений в роботі модуль дозволяє мікрокомп'ютеру включати та виключати до 8 незалежних навантажень. Наприклад: електродвигун, лампа освітлення, сервопривід, насос або нагрівач. Також, дозволяє вводити сигнали. Такими джерелами сигналів можуть виступати: кнопки (на замикання або на розмикання), перемикачі, датчики комутаційного типу та багато іншого.

Окрім представленого переліку, мікрокомп'ютер залишається обладнаним цифровими портами типу I2C, 1-wire.

Тому, для роботи у якості віддаленого елемента керування нам потрібно провести створення також і програмної частини.

Зм.	№докум.	Підпис	Дата	

3.3.1 Мова програмування для Raspberry Pi

В Raspberry основною мовою є "Пайтон". Тому власнику Raspberry, який поки не знає ніякого програмування, рекомендується вивчати саме цей. Причин, чому Raspberry Pi з програмування на Python найбільш краще рішення, існує безліч.

Ось лише деякі з них:

- робота з коробки в Raspbian;
- наявність великої, добре документованої офіційної бібліотеки, яка вже включена в пакет;
- простота мови і т. Д.

3.3.2 Середовище розробки програмного коду Thonny

Thonny: найкраща середа для починаючих Python-розробників

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		Нодокум.	Підпис	Дата		57

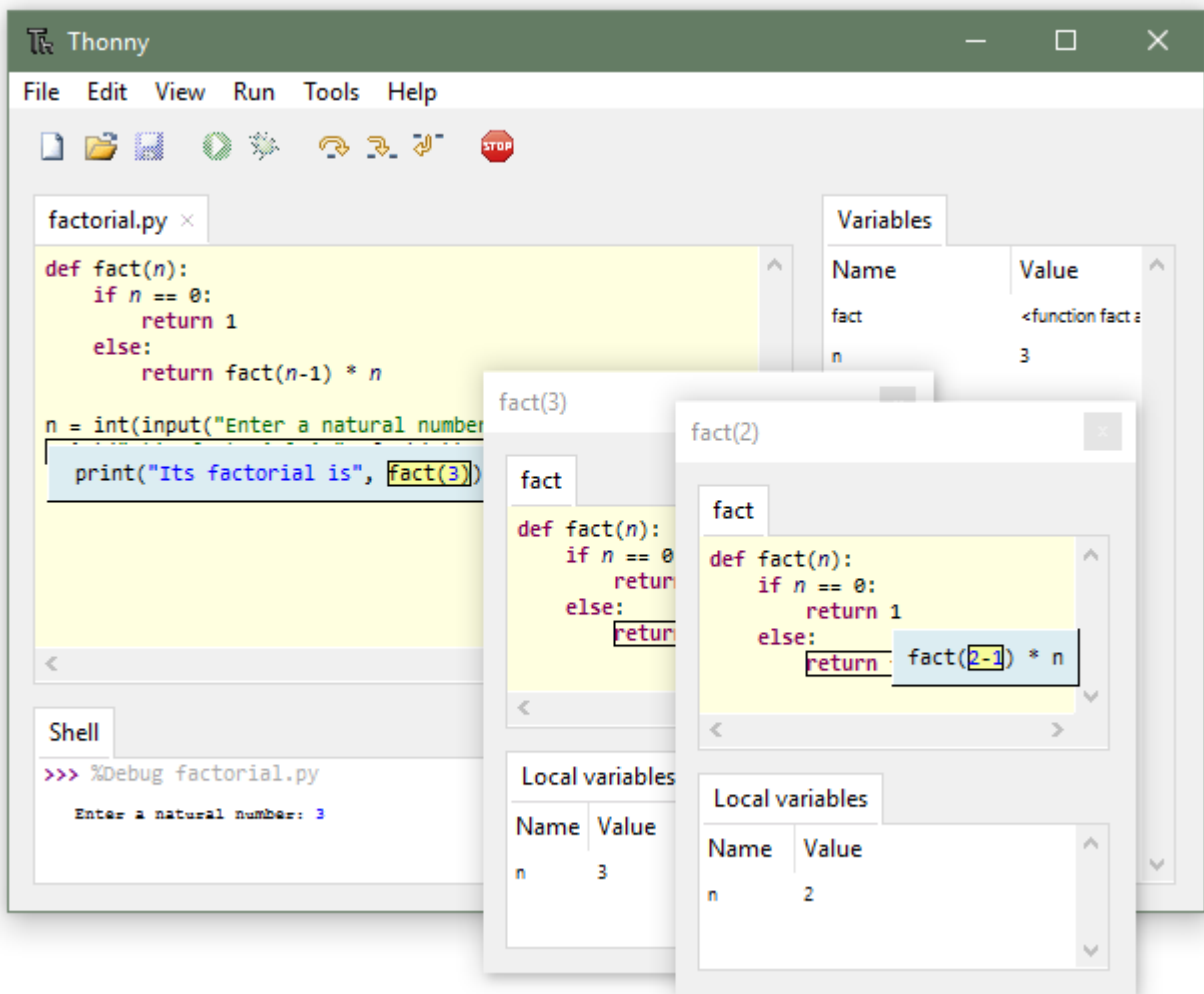


Рисунок 3.7 – Середовище Thonny

Thonny - це просунута Python-IDE, яка добре підходить для новачків. Хоча користуватися їй цілком можуть і професіонали, деякі риси цієї IDE говорять про те, що вона особливо хороша для початківців пітоністів. Вона дає в розпорядження програміста можливості по покрокового виконання виразів, засоби візуалізації стека викликів і безліч інших корисних дрібниць. Якщо новачок візьме все це на озброєння - він не тільки поліпшить свої навички Python-програмування, але і буде краще розуміти те, що відбувається під час виконання коду.

Отже, головні властивості Thonny:

- Підтримка Python.
- Простий у використанні відладчик.
- Зручність для новачків.
- Обчислення виразів.
- Підсвічування синтаксичних помилок.
- Допоміжні засоби, що допомагають розібратися в коді.

3.3.3 Мережеві налаштування для Raspberry Pi

Оскільки Raspberry Pi буде пристроєм, яким ми плануємо керувати через мережу Інтернет, тому для пристрою слід призначити статичну IP-адресу. Статична адреса дозволяє знайти пристрій у всесвітній мережі Інтернет за його адресом. Слід звернути увагу, що адреса має бути надана провайдером напряму.

Так, нехай адреса в мережі для Raspberry Pi відома і буде 108.201.50.127. Виконаємо первинне налаштування Raspberry Pi на етапі, коли плата підключена до ЕОМ. Отже, змінюємо вміст файлу `/etc/network/interfaces` на мікрокомп'ютері.

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
    address 108.201.50.127
    netmask 255.255.0.0
    gateway 108.201.50.1
    dns-nameservers 8.8.8.8
```

- повністю видаляємо з системи `dhcpcd5`, виконавши команду

```
sudo apt-get purge dhcpcd5
```

3.3.4 Встановлення фреймворка Webіорі

Webіорі представляє пакет програм, спеціально розроблений для Raspberry Pi для віддаленого управління пристроями. Спільно з Raspberry Pi він реалізує технологію Internet of Things (Інтернет речей). Webіорі дозволяє створювати різні користувальницькі додатки. На рисунку 3.8 показані основні можливості цього фреймворка відповідно до його офіційним сайтом <http://webiopi.trouch.com/>.

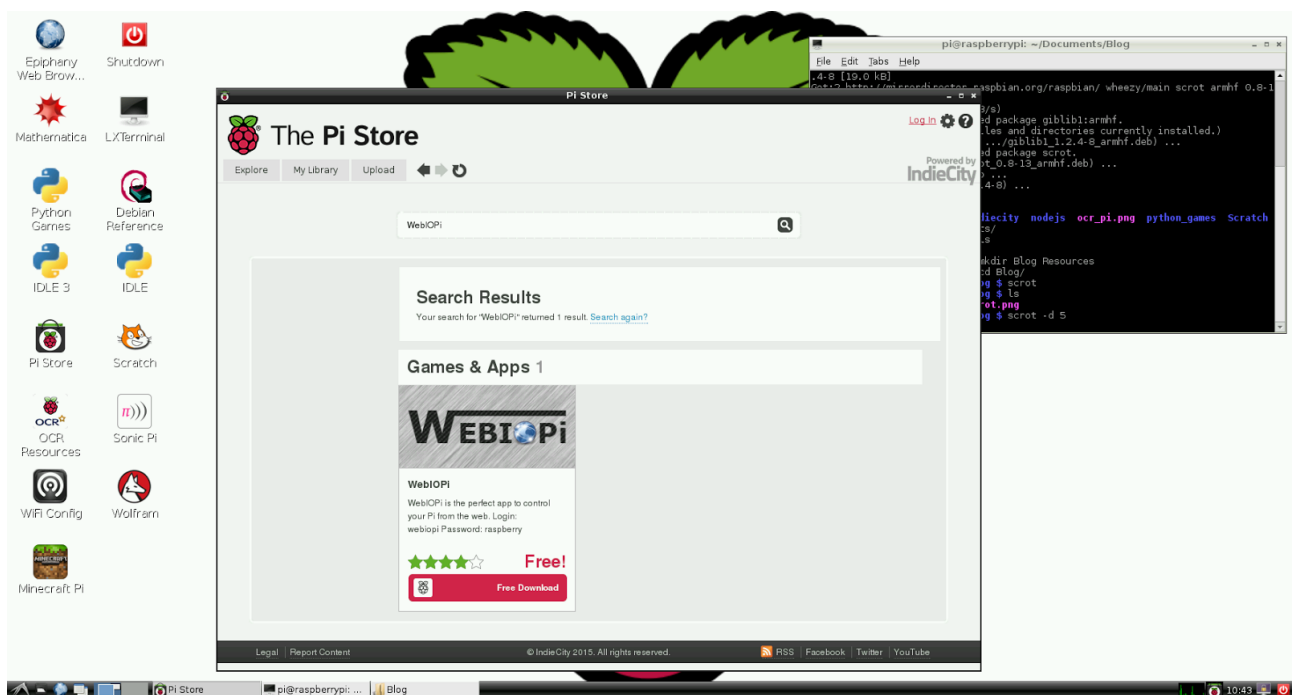


Рисунок 3.8 – Webіорі

Таким чином Webіорі має наступні можливості:

- Вбудований Web - сервер, реалізований на мові Python
- Вбудована підтримка більш ніж 30 пристроїв з інтерфейсами UART, SPI, I2C, 1-Wire
- Бібліотеки Javascript / HTML для створення Web-інтерфейсу
- Бібліотеки Python / Java для створення додатків під Android

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		Нодокум.	Підпис	Дата		60

- Підтримує протокол CoAP призначений для управління і взаємодії між простими електронними пристроями через мережу.

Webіорі має відкритий код, який може бути змінений користувачем. Це дозволяє збільшити кількість завдань для вирішення. Для настройки пакета під конкретну задачу змінюється файл конфігурації. Наприклад, в цей файл записуються GPIO pins, до яких підключені пристрої. Якщо використовуються датчики, їх також заносять в конфігураційний файл. Однак необхідно в деяких випадках включити драйвер пристрою (наприклад для температурного датчика ds18b20). Розглянемо установку версії 0.6 Webіорі. Ця версія добре підтримує Raspberry Pi model B, що має 26 пінів порту GPIO. Для установки WebІОPi, заходимо в Raspberry Pi через програму Putty (логін - pi, пароль - raspberry) і в терміналі вводимо по черзі наступні команди:

```
$ wget http://webiopi.googlecode.com/files/WebIOPi-0.6.0.tar.gz
$ tar xvzf WebIOPi-0.6.0.tar.gz
$ cd WebIOPi-0.6.0
$ sudo ./setup.sh
```

Після завершення установки необхідно активувати автозапуск WebІОPi. Автозапуск WebІОPi потрібен для того, що б кожен раз після включення Raspberry Pi не виконувати запуск програми вручну. Для цього виконуємо команду:

```
$ sudo update-rc.d webiopi defaults
```

Після чого перезапускаємо Raspberry Pi:

```
sudo reboot
```

Тепер можна перевірити роботу Webіорі. З будь-якого комп'ютера в локальній мережі набираємо мережеву адресу, присвоєний Raspberry Pi із зазначенням порту 8000. Наприклад:

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		61

<http://108.201.50.127:8000/app/gpio-header>

Для доступу до WebIOPi необхідно в формі, що відкрилася ввести логін і пароль. За замовчуванням логін «webіорі», пароль - «raspberrу». На малюнку 6 представлений інтерфейс програми Webіорі, на якому представлені номери пинов порту GPIO і їх призначення.

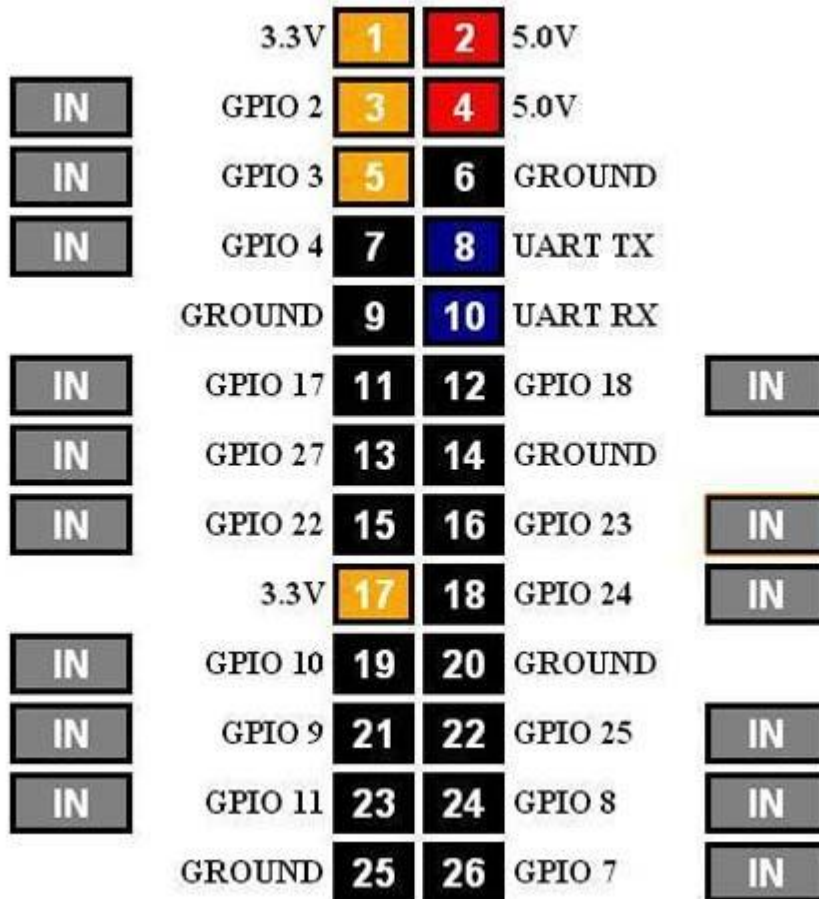


Рисунок 3.9 – Інтерфейс налаштування

З малюнка видно, що за замовчуванням всі Піни GPIO працюють на вхід (IN). Для перемикання на вихід, необхідно мишкою натиснути на IN. Його значення зміниться на OUT. Чорний колір Піна означає низький рівень порту, жовтий - високий. Якщо клікнути мишкою на номер порту, його значення переключитися з низького на високий і чорний колір зміниться на жовтий. В

цьому випадку відбудеться включення (або виключення) пристрою, підключеного до цього піну.

Для настройки Webіорі під задачу необхідно вибрати Піни порту GPIO для підключення вихідних реле і датчика температури DS18B20 і прописати їх в конфігураційному файлі Webіорі:

GPIO23 – реле 1;

GPIO9 – реле 2;

GPIO11 – реле 3.

Датчик температури підключений до GPIO4, який реалізує шину 1-wire.

Підключаємося до Raspberry Pi через термінальну програму Putty (логін - pi, пароль - raspberry) і відкриваємо для редагування конфігураційний файл config:

```
sudo nano /etc/webiopi/config
```

Налаштовуємо порти GPIO 9, 11, 23 як виходи з високим стартовим рівнем. У секції [GPIO] записуємо:

```
9 = OUT 1
```

```
11 = OUT 1
```

```
23 = OUT 1
```

Для того, щоб працював датчик температури в секції [DEVICES] прописуємо:

```
tmp0 = DS18B20
```

У секції [SCRIPTS] вказуємо ім'я та розташування файлу - скрипта на Python, необхідний для зчитування даних з датчика температури і контролю настання включення або виключення обігрівача в заданий час:

```
myproject = /home/pi/myproject/python/script.py
```

У секції [HTTP] вказуємо ім'я та розташування html файлу:

```
doc-root = / home / pi / myproject / html
```

і номер порту роботи вбудованого в Webіорі web - сервера:

```
port = 80
```

У секції [REST] прописуємо активні порти (тільки вони будуть працювати):

```
gpio-export = 9, 11, 23
```

і дозволяємо вводити зміна з браузера величин GPIO (низький і високий рівень Піна) але забороняємо міняти функції GPIO (не можна міняти OUT на IN):

```
gpio-post-value = true
```

```
gpio-post-function = false
```

Для того, щоб працював датчик DS18B20 необхідно в файл /boot/config.txt вставити рядок

```
dtoverlay = w1-gpio
```

Якщо до GPIO4 підключено кілька датчиків DS18B20, то кожен з них повинен мати своє ім'я. Для визначення імені переходимо в каталог:

```
cd / sys / bus / w1 / devices
```

Роздруковуємо файли в цьому каталозі. Назва файлу відповідає за серійний номер датчика DS18B20:

```
ls
```

```
28-0000067b503c 28-00000d3a5f18 w1_bus_master1
```

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		Нодокум.	Підпис	Дата		64

Перші два імені визначають два датчика DS18B20. Заносимо їх в / etc / webiopi / config в секцію [DEVICES]:

```
tmp0 = DS18B20 slave: 28-0000067b503c
```

```
tmp1 = DS18B20 slave: 28-00000d3a5f18
```

Зміна пароля Webiopi виконується командою

```
$ Sudo webiopi-passwd
```

Далі виконуємо перевантаження комп'ютера командою reboot.

Підключитися до web - сервера webiopi тепер можна за адресою

```
http://108.201.50.127/app/gpio-header
```

У браузері з'являться зображення аналогічне рис.3.9, але активними будуть тільки порти gpio 9, 11, 23.

Перевірити працездатність датчика температури можна, підключившись за адресою:

```
http://108.201.50.127/app/devices-monitor
```

У браузері має з'явитися значення температури від датчика.

Для перевантаження webiopi після внесення змін до конфігураційний файл, Скрипт на пітона, html файл, необхідно виконати команду:

```
/etc/init.d/webiopi restart
```

Повідомлення про помилки при запуску Webiopi знаходяться в файлі /var/log/webiopi. Його можна роздрукувати:

```
cat /var/log/webiopi
```

Файл index.html представлено у додатку. За основу взяті стандартні файли, що є на сайті <http://webiopi.trouch.com>.

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		65

Цей html файл з WebIOPi Javascript library повинен сформувати web - сторінку, передавати на сервер команди при натисканні на кнопки, одержувати від сервера інформацію про стан кнопок, виводити значення дати, температури і взаємодіяти зі скриптом на Python. Файл index.html розмістимо в каталозі /home/pi/myproject/html, який попередньо необхідно створити. Цей файл створювався за прикладами, які можна знайти в посиланнях: http://webiopi.trouch.com/Tutorial_Basis.html, http://webiopi.trouch.com/Tutorial_Macros.html, http://webiopi.trouch.com/Tutorial_Devices.html

3.3.5 Файл-скрипт на пайтоні script.py

Цей файл повинен знаходитися в каталозі /home/pi/myproject/python, який повинен бути попередньо створений. Завданням скрипта є отримання даних з температурного датчика і передача їх в файл index.html, запис файлу /home/pi/myproject/html/data_18B20.txt з значеннями температур і часу їх визначення для побудови графіка, аналізу часу включення і виключення обігрівача і його включення або відключення. Файл також представлено у додатку.

3.3.6 Побудова графіка температури

Графік температури будуватися за допомогою бібліотеки dygraph, файл якої dygraph-combined-dev.js необхідно скопіювати в каталог /home/pi/myproject/html з сайту <http://dygraphs.com>. Ця бібліотека для відображення графіка використовує файл /home/pi/myproject/html/data_18B20.txt, який у свою чергу формує кожні 100 секунд скрипт на Python. Побудова графіка буде виконуватися браузером при зверненні за адресою: <http://108.201.50.127/temp.html> .

3.4 Висновки

Таким чином, плата Raspberry PI дозволяють будувати інтерактивні системи керування, що програмуються із застосуванням простих мов програмування, наприклад Python.

Представлена в проекті робота показує простоту та зручність побудови плати розширення, що дозволяють приймати цифрові сигнали та керувати потужним навантаженням.

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		№докум.	Підпис	Дата		67

Перелік літературних джерел

5. FAQs. Raspberry PI. Архів оригіналу за 2013-07-16. Процитовано 2011-11-03. «What Linux distros will be supported at launch? Debian, Fedora and ArchLinux will be supported from the start.»

6. <http://www.raspberrypi.org/archives/2180>

7. Начались продажи 25-долларового компьютера Raspberry PI // <http://lenta.ru/news/2013/02/04/raspi/>

8. Jim Salter (2020-10-27). Ubuntu Groovy Gorilla adds Raspberry PI as a “first class citizen”. Ars Technica.

9. Raspberry PI Zero - Raspberry PI (en-GB). Raspberry PI. Процитовано // <https://www.raspberrypi.org/blog/raspberry-pi-zero/>.

10. PETER BRIGHT (2018-03-14). Raspberry PI 3 B+ has faster CPU, Wi-Fi, and easier compliance testing. ARS Technica.

11. Michael Larabel (2018-03-22). Raspberry PI 3 Model B+ Benchmarks. Phoronix // <https://www.phoronix.com/scan.php?page=article&item=raspberrypi-3-bplus&num=1>

12. Michael Larabel (2018-03-23). Raspberry PI 3 Model B+ WiFi Performance. Phoronix // <https://www.phoronix.com/scan.php?page=article&item=raspberry-pi3bplus-wifi&num=1>

13. Paunski, Angelov, Yassen, Georgi (2019). IFAC-PapersOnLine (English). с. 425.

14. Eric Brown (2018-08-25). Raspberry PI's PoE HAT ships for \$20, tosses in a free fan. // <http://linuxgizmos.com/raspberry-pis-poe-hat-ships-for-20-tosses-in-a-free-fan/>

15. Nick Heath (2019-06-23). The Raspberry PI 4 is a true next-generation Pi, with dual-screen 4K support, USB 3.0, a new CPU and GPU, and up to 4GB RAM. Tech Republic.

16. Ron Amadeo (2019-07-09). Raspberry PI admits to faulty USB-C design on the Pi 4. Ars Technica.

17. Jim Salter (2020-11-03). Raspberry PI 400 review—the under-\$100 desktop PC you didn't know you needed. Ars Technica.

18. Eben Upton (2020-11-02). Raspberry PI 400: the \$70 desktop PC. Raspberry PI Blog.

19. Romain Dillet (2021-01-21). Raspberry PI Foundation launches \$4 microcontroller with custom chip. Tech Crunch.

20. Raspberry PI Pico за \$4 — на что способна новая плата от разработчиков “малинок”, 22 січня 2021 // <https://habr.com/ru/company/selectel/blog/538778/>

21. <https://www.raspberrypi.org/blog/raspberry-pi-silicon-pico-now-on-sale/>

22. Shimpi, Anand Lal. The iPhone 3GS Hardware Exposed & Analyzed. www.anandtech.com. Цитувано 2018-12-18.

23. Introducing turbo mode: up to 50% more performance for free. Raspberry PI (en-GB) // <https://www.raspberrypi.org/blog/introducing-turbo-mode-up-to-50-more-performance-for-free/>

24. Bradbury, Alex (2018-12-07). Configuration tool for the Raspberry PI. Now maintained at <https://github.com/RPi-Distro/raspi-config>: asb/raspi-config. Процитовано 2018-12-18.

25. CFL Podcast: Eben Upton (Raspberry PI Foundation). Consortium of Foundation Libraries. Архів оригіналу за 2013-07-16.

26. Wong, George (2011-10-24). Build your own prototype Raspberry PI minicomputer. ubergizmo // <http://www.ubergizmo.com/2011/10/build-raspberry-pi-minicomputer/>

27. Humphries, Matthew (2011-07-28). Raspberry PI \$25 PC goes into alpha production. Geek.com. Архів оригіналу за 2013-07-16. Процитовано 2011-08-01.

28. Q&A with our hardware team. Raspberry PI Foundation. // <https://www.webcitation.org/6IA8dythT?url=http://www.raspberrypi.org/archives/169>

29. Holwerda, Thom (31 October 2011). Raspberry PI To Embrace RISC OS. OSNews // http://www.osnews.com/story/25276/Raspberry_Pi_To_Embrace_RISC_OS

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		№докум.	Підпис	Дата		70

ДОДАТОК А

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Temperatura 1-wire DS18B20</title>
<style>
  a {
    color: #FF0000; /* Колірт звичайного посилання */
    text-decoration: none; /* видаляємо підкреслення у посилань */
  }
  a:visited {
    color: #800080; /* Колір відвіданого посилання */
  }
  a:hover {
    color: #800000; /* Колір посилання при наведені на неї курсором миші */
    text-decoration: underline; /* Додавляем підкреслення */
  }
</style>
  <script type="text/javascript" src="/webiopi.js"></script>
  <script type="text/javascript">

    // Виклик макросу зміни температуру з інтервалом 5 сек.
    setInterval ("callMacro_getTmp0()", 5000);{
      }
    // Запит температури
    function callMacro_getTmp0(){
      webiopi().callMacro("getTmp0", [], macro_getTmp0_Callback);
    }
    // Отримання температури
    function macro_getTmp0_Callback(macro, args, data) {
      $("#celsius_0").text("Температура: "+data+" °C");
    }
setInterval ("callMacro_tihour()", 2000);{
}
function callMacro_tihour(){
webiopi().callMacro("tihour", [], macro_tihour_Callback);
}
function macro_tihour_Callback(macro, args, data) {
$("#hr").text(data);
}
  webiopi().ready(function() {
    // Following function will process data received from set/getLightHours
    macro.
      var updateLightHours = function(macro, args, response) {
        var hours = response.split(";");
        // Following lines use jQuery functions
        $("#inputOn").val(hours[0]);
        $("#inputOff").val(hours[1]);
        $("#inputOnm").val(hours[2]);
        $("#inputOffm").val(hours[3]);
      }
    // Immediately call getLightHours macro to update the UI with current values
    // "getLightHours" refers to macro name
    // [] is an empty array, because getLightHours macro does not take any argument
    // updateLightHours is the callback function, defined above

    webiopi().callMacro("getLightHours", [], updateLightHours);
  });
</script>
</body>
</html>
```

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		71

```

    // Create a button to call setLightHours macro
    var sendButton = webiopi().createButton("sendButton", "Передати", function() {
        // Arguments sent to the macro
        var hours = [$("#inputOn").val(), $("#inputOff").val(), $("#inputOnm").val(),
            $("#inputOffm").val()];
        // Call the macro
        webiopi().callMacro("setLightHours", hours, updateLightHours);
    });

    // Append the button to the controls box using a jQuery function
    $("#send_contr").append(sendButton);

    // Create a "Обігрівач" labeled button for GPIO 23
    var button = webiopi().createGPIOButton(23, "Обігрівач");

    // Append button to HTML element with ID="controls" using jQuery
    $("#controls").append(button);

    // Create a "Розетка 1" labeled button1 for GPIO 9
    var button1 = webiopi().createGPIOButton(9, "Розетка 1");

    // Append button to HTML element with ID="controls1" using jQuery
    $("#controls1").append(button1);

    // Create a "Розетка 2" labeled button2 for GPIO 11
    var button2 = webiopi().createGPIOButton(11, "Розетка 2");

    // Append button to HTML element with ID="controls2" using jQuery
    $("#controls2").append(button2);

    // Refresh GPIO buttons
    // pass true to refresh repeatedly of false to refresh once
    webiopi().refreshGPIO(true);
});
</script>

<style type="text/css">
#celcius_0, #hr {
margin: 5px 5px 5px 5px;
width: 350px;
height: 75px;
background-color: Blue;
font-size: 24pt;
font-weight: bold;
color: white;
padding: 5px 5px 5px 5px;
}
</style>
<style type="text/css">
#send_contr {
display: block;
margin: 5px 5px 5px 5px;
width: 360px;
height: 85px;
font-size: 34pt;
font-weight: bold;
color: white;
}
</style>

<style type="text/css">
button, button1, button2 {

```


ДОДАТОК Б

```
import webiopi
import datetime
import time
from webiopi import deviceInstance
from time import strftime
GPIO = webiopi.GPIO

RELAY1 = 23 # GPIO pin

HOUR_ON = 13 # Turn Relay 1 ON at 13:00
HOUR_OFF = 14 # Turn Relay 1 OFF at 14:00
MIN_ON = 0
MIN_OFF = 0
celsius_0 = 0,0
cel_0=0,0
num=1

# ця функція автоматично запускається при запуску WebIOPi
def setup():
    # set the GPIO used by the light to output
    GPIO.setFunction(RELAY1, GPIO.OUT)

    # retrieve current datetime
    now = datetime.datetime.now()
    # test if we are between ON time and tun the light ON
    if ((now.hour >= HOUR_ON) and (now.hour <= HOUR_OFF)) and ((now.minute >=
MIN_ON) and (now.minute <= MIN_OFF)):
        GPIO.digitalWrite(RELAY1, GPIO.LOW)

@webiopi.macro
def getTmp0():
    global celsius_0
    tmp0 = webiopi.deviceInstance("tmp0")
    celsius_0 = tmp0.getCelsius() # получение температуры
    print (celsius_0)
    return "%.2f" % celsius_0 # возврат данных температуры в HTML
    # с округлением до сотых

@webiopi.macro
def tihour():
    now = datetime.datetime.now()
    return "Час {0}, {1}".format("%d:%d %d:%d" % (HOUR_ON, MIN_ON, HOUR_OFF,
MIN_OFF), strftime("%Y-%m-%d %H:%M"))

def loop():
    global cel_0
    global num
    # retrieve current datetime
    now = datetime.datetime.now()
    # toggle light ON all days at the correct time
    if ((now.hour == HOUR_ON) and (now.minute == MIN_ON) and (now.second == 0)):
        if (GPIO.digitalRead(RELAY1) == GPIO.HIGH):
            GPIO.digitalWrite(RELAY1, GPIO.LOW)

    # toggle Relay 1 OFF
    if ((now.hour == HOUR_OFF) and (now.minute == MIN_OFF) and
```

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		74

```

        (now.second == 0)):
            if (GPIO.digitalRead(RELAY1) == GPIO.LOW):
                GPIO.digitalWrite(RELAY1, GPIO.HIGH)
    if (num==100):
        tmp0 = webiopi.deviceInstance("tmp0")
        cel_0 = tmp0.getCelsius()
        f = open('/home/pi/myproject/html/data_18B20.txt', 'a')
        data_entry = "{0},{1}\n".format(strftime("%Y-%m-%d %H:%M:%S"), "%0.2f" %
cel_0)
        f.write(data_entry)
        f.close()
        num=0
        num+=1
        # gives CPU some time before looping again
        time.sleep(1)

@webiopi.macro
def getLightHours():
    return "%d;%d;%d;%d" % (HOUR_ON, HOUR_OFF, MIN_ON, MIN_OFF)

@webiopi.macro
def setLightHours(on, off, onm, offm):
    global HOUR_ON, HOUR_OFF, MIN_ON, MIN_OFF
    HOUR_ON = int(on)
    HOUR_OFF = int(off)
    MIN_ON = int(onm)
    MIN_OFF = int(offm)
    return getLightHours()

# destroy function is called at WebIOPi shutdown
def destroy():
    GPIO.digitalWrite(RELAY1, GPIO.HIGH)

```

					<i>КПТР. 2017010.01.07 ПЗ</i>	Арк.
Зм.		№докум.	Підпис	Дата		75

ДОДАТОК В

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Temperature_graph</title>

<script type="text/javascript"
src="dygraph-combined-dev.js"></script>

<!=====ВИВІД ГРАФІКА НА СТОРІНКУ>

<div id="graph_temperatura"
style="width:1200; height:600px;"></div>

<!=====СКРИПТ ГРАФІКА>

<script type="text/javascript">
  g3 = new Dygraph(
    document.getElementById("graph_temperatura"),

    "data_18B20.txt",
      "DATA, Temperature\n" +           // the data series
      "2021-01-01 12:00,15.00\n" +
      "2021-01-01 12:15,23.50\n" +
      "2021-01-01 12:30,23.00\n",

    {
      title: 'Температура',
      legend: 'always',
      ylabel: 'Temp',
      xlabel: 'Date',
      rollPeriod: 5,
      color: "#FF0000",
      showRoller: true
    }
  );
</script>
</head>
</html>
```

3

					КПТР. 2017010.01.07 ПЗ	Арк.
Зм.		№докум.	Підпис	Дата		76

РЕЦЕНЗІЯ
на кваліфікаційний проєкт Мокрицького Андрія Богдановича
“ Проєктування систем видаленого керування на основі міні комп'ютера
Raspberry PI ” , за спеціальністю 172 Телекомунікації та радіотехніка

В роботі представлено проведені опрацювання актуальної тематики щодо застосування сучасних систем SoC – систем на кристалі. Тенденції розвитку телекомунікаційних мереж і систем передачі інформації передбачають взаємодію з широким обсягом різноманітних послуг, таких як доступ до аудіо, відео контентів, інтерактивна взаємодія з обладнанням.

Raspberry PI - це мініатюрний одноплатний комп'ютер, який з легкістю поміститься на долоні дорослої людини. Незважаючи на свої скромні розміри, плата має високу продуктивність, що дозволяє їй вийти на один рівень із стаціонарними ПК.

Основною відмінною рисою Raspberry PI від звичайних комп'ютерів, являється наявність програмованих портів введення-виведення GPIO. За допомогою їх можна управляти різними пристроями і приймати телеметрію з різного роду датчиків.

У представленій бакалаврській роботі викладені результати, що пов'язані із аналізом існуючих мікрокомп'ютерних плат. Проаналізовано причини, чому для плат Raspberry PI припадає велика частина запитів на створення користувацьких рішень.

Показано основні складові плати Raspberry PI зі сторони наявного обладнання, такого як: USB 3, Gigabit Ethernet port, порти камери та дисплею. Основним елементом є універсальний порт GPIO, який розширяє можливості плати.

В роботі автором показано схематичні аспекти створення такої плати розширення із застосуванням простої елементної бази, яка одночасно забезпечить захист плати Raspberry PI.

До недоліків роботи слід віднести такі зауваження:

1. В роботі не має математичних моделей та розрахунків.
2. Розглянута в третьому розділі спроектована схема є дещо абстрактною, без конкретного впровадження.

Незважаючи на зазначені вище зауваження та недоліки, робота Мокрицького Андрія Богдановича є закінченим дослідженням, яке за предметом досліджень, метою, вирішеним завданням, висновками відповідає вимогам до бакалаврських кваліфікаційних робіт за спеціальністю 172 Телекомунікації та радіотехніка.

Результати роботи мають практичну цінність для задач з побудови електронних плат розширення функцій існуючого мікропроцесорного обладнання

Автор роботи, Мокрицький Андрій Богданович, заслуговує на оцінку «добре».

к.т.н., доц. кафедри



Мішан В.В.

Завідувачу кафедри телекомунікацій
та радіотехніки
д.т.н., професору БОЙКУ Юлію
здобувача вищої освіти
МОКРИЦЬКОГО Андрія
ФПКТС, гр. ТР-17-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

25 травня 2021 р.
дата


підпис

Ім'я користувача:
Kafedra telekom_radiotechniki

ID перевірки:
1008314710

Дата перевірки:
16.06.2021 20:03:49 EEST

Тип перевірки:
Doc vs Internet

Дата звіту:
16.06.2021 20:04:50 EEST

ID користувача:
100005656

Назва документа: Мокрицький

Кількість сторінок: 59 Кількість слів: 8991 Кількість символів: 66794 Розмір файлу: 2.23 MB ID файлу: 1008381940

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.18%
Схожість

Найбільша схожість: 4.78% з Інтернет-джерелом (https://uk.wikipedia.org/wiki/Raspberry_Pi)

6.18% Джерела з Інтернету

148

Сторінка 61

Пошук збігів з Бібліотекою не проводився

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

8

Підозріле форматування

14
сторінок

РІШЕННЯ КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОТЕХНІКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Проектування систем видаленого керування на основі міні комп'ютера RASPBERRY PI

Автор: Мокрицький Андрій Богданович

Спеціальність: 172 Телекомунікації та радіотехніка

Науковий керівник: д.т.н., професор Мясіщев Олександр Анатолійович

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	-
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
5	Інше:	-

Підтвердження: Виявленні запозичення не є плагіатом так як розміщені в розділах, які не описують безпосередньо авторське дослідження (є власні терміни, визначення тощо), складають 2,38% та мають посилання на приведений список літературних джерел.

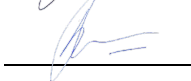
«16» червня 2021 р.

Науковий керівник



Олександр МЯСІЩЕВ

Завідувач кафедрою ТР



Юлій БОЙКО