

1. АНАЛІТИЧНИЙ ОГЛЯД

Комп'ютерна система електронного органайзера є програмно-апаратний комплекс, який призначений для контролю за виконанням справ, відстеження користувачем подій. Основні функції системи складаються з відображення дати і часу, секундоміра, таймера та будильника. Також присутня функція автоматичного повідомлення користувача про події. Даний комплекс має відносно невеликий розмір, швидкий і зручний у використанні в повсякденному житті.

1.1 Призначення електронного органайзера

Органайзер, планувальник, ділової календар, записна книжка - це тільки частина назв комп'ютерних систем, які допомагають оптимізувати робочий час і позбавляють людину від необхідності записувати важливу інформацію на листках паперу, часто безповоротно губляться на робочому столі. Людина, яка захотіла впорядкувати свою роботу, на сьогоднішній день може вибрати програму з десятків систем-органайзерів. Тільки для початку варто визначитися, для яких конкретно завдань і як часто буде використовуватися ця система. З маси подібного комплексів, дійсно, не так легко знайти оптимальний варіант. Але, як то кажуть, хто шукає - той завжди знайде.

У нашому швидкому і метушливому світі дуже важливо все встигнути, не пропустити жодної задачі. Від правильно складеного розпорядку дня багато залежить. Адже головне - не забути нічого і зберегти такі головні ресурси, як сили і час. Допоможе в цьому органайзер для робочого столу (якщо говорити про комп'ютери) або побутової електронний органайзер, який не займає багато місця і простий у використанні.

На жаль, точна дата винаходу органайзера невідома. Напевно, це сталося незабаром після появи календаря, адже дата і подія - це нерозривні поняття. У ролі перших виступали зарубки на палицях (звідси і вираз «Зарубай собі на носі»). Проблема полягала в тому, що такий «органайзер» не містив інформації про всі завдання, які потрібно виконати. Люди часто ламали голову,

намагаючись пригадати, про що ж була карб. Це слово прийшло з англomовних країн, позначає предмет, який допомагає в організації справ і подій, нагадує про важливу дату. Спочатку це був блокнот або книга з календарем, в яку записувалися всі заплановані заходи, адреси, контакти і інша необхідна інформація. Головне призначення - домогтися ефективного розподілу часу з метою виконання всіх запланованих справ і завдань.

Час - одне з головних цінностей в нашому житті. Постійна його нестача є однією з причин стресу. Управляти часом цілком можливо завдяки такому помічникові, як електронний органайзер у себе в кишені. Успішно розподіляти час - його головне завдання. Такий планувальник користується величезною популярністю у ділових людей. Існує багато видів органайзерів, доступних для застосування в різних областях. Всі вони покликані впорядкувати і спростити наше життя. У будь-якій області призначення органайзера для робочого столу - це чітке структурування справ або речей.

Електронні планувальники з'явилися більше 25 років тому. Спочатку ці гаджети поєднували в собі функції калькулятора і календаря з безліччю незрозумілих функцій. Своім виглядом вони спантеличували людей, не знайомих з вищою математикою. З появою мобільних телефонів і портативної техніки все спростилося, зараз в багато електронні пристрої вбудований органайзер. Електронний органайзер на робочий стіл для Windows, Linux або для смартфона - це інструмент, який збирає і зберігає дані користувача з можливістю пошуку занесеної інформації. Він відстежує потрібні події, допомагає ставити завдання і контролює їх виконання. Можна створити простий органайзер на своїми руками на основі таблиці Excel - внести номери телефонів, важливі дати і іншу корисну інформацію. Але він мало в чому відрізняється від паперового носія. Це просто сховище інформації. Головна функція органайзера, нагадування, в цьому випадку відсутній. І завжди є ризик забути або пропустити щось важливе. Набагато зручніше використовувати спеціалізовані пристрої електронних органайзерів.

1.2 Мікроконтролери і мікрокомп'ютери

Якщо необхідно зібрати комп'ютерну систему, то доведеться зіткнутися з вибором апаратної платформи, існує широкий вибір плат і рішень для побудови ефективної і недорогой системи з використанням готових рішень. Деякі з доступних рішень для плат мікроконтролерів і мікрокомп'ютерів, які в даний час досить поширені і доступні на різних ринках. Мікрокомп'ютери використовують операційні системи, а мікроконтролери використовують мікропрограму на мові Асемблера або С, або будь-який інший мову, скомпільований під конкретний мікроконтролер.

Мікроконтролер - це мікрочіп або плата з мікрочіпом для вирішення комп'ютерних систем. Зазвичай можна програмувати мікроконтролери за допомогою мови програмування високого рівня, такого як С, С++, Python і т.д. Деякі комп'ютерні системи найпростіше програмувати на мікроконтролерах. Вони підтримують безліч стандартів введення і виведення, працюють з меншим енергоспоживанням і коштують дешевше в порівнянні з мікрокомп'ютерами. Недоліком є менша обчислювальна потужність і відсутність операційної системи за умовчанням. Найбільш популярними мікроконтроллерами є ESP8266, ESP32, Atmel, STM, ESP (або, наприклад, плати Arduino) і т.д.

Мікрокомп'ютер зазвичай являє собою систему на чіпі, включаючи класичну архітектуру фон Неймана з центральним процесором, відеокартою, оперативною пам'яттю, мережами WiFi / Bluetooth і портами введення-виведення. Сучасні мікрокомп'ютери використовують такі операційні системи, як Linux та Windows. Як правило, мікрокомп'ютери мають велику обчислювальну потужність, ніж мікроконтролери, відеовихід на HDMI, високошвидкісний Wi-Fi і Bluetooth, підключення до карт флеш-пам'яті і M.2 і т.д. Недоліком мікрокомп'ютерів є більш висока ціна і більш високе енергоспоживання в порівнянні з мікроконтроллерами. Мікрокомп'ютери використовуються в комп'ютерних системах, якщо необхідно виконувати завдання високого рівня, включаючи потокове відео, складні інформаційні системи, міні-сервери і т.д.

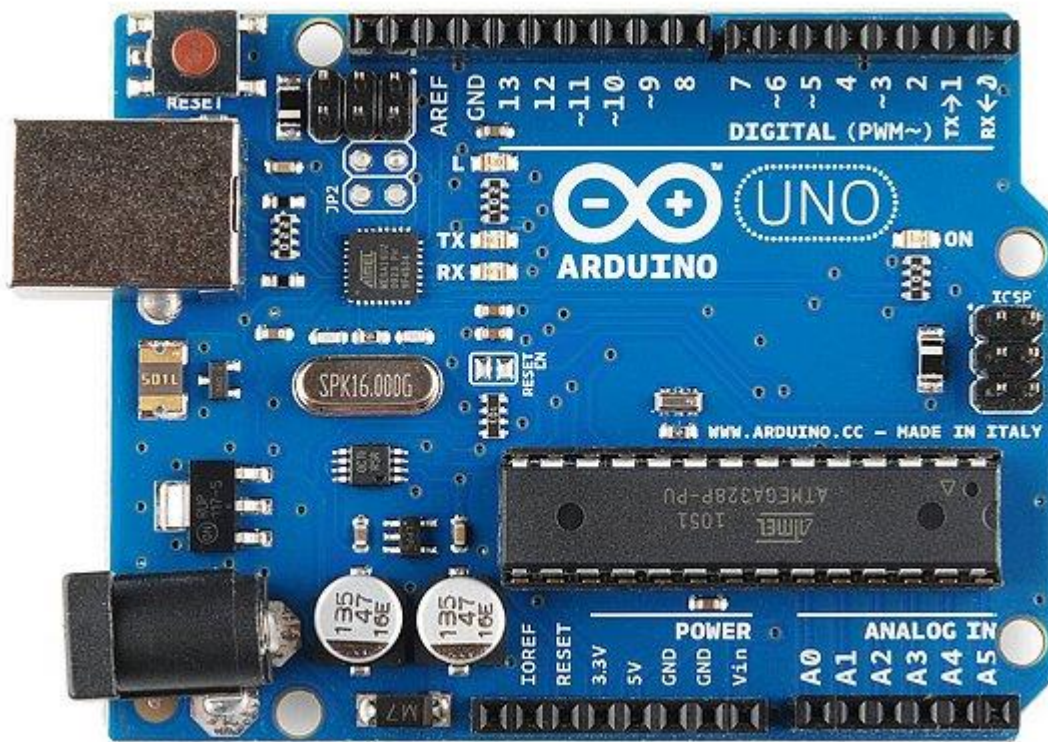
Як правильно вибрати апаратне забезпечення для побудови комп'ютерної

системи? Це рішення в першу чергу залежить від розміру завдань, які повинні бути виконані на стороні клієнта.

У будь-якому випадку необхідно зосередитися на кращій якості і дешевому рішенні для побудови клієнтської частини. В даний час на ринку представлені досить потужні мікропроцесорні плати, які можна легко інтегрувати навіть з протоколами Bluetooth і WiFi плюс HTTP.

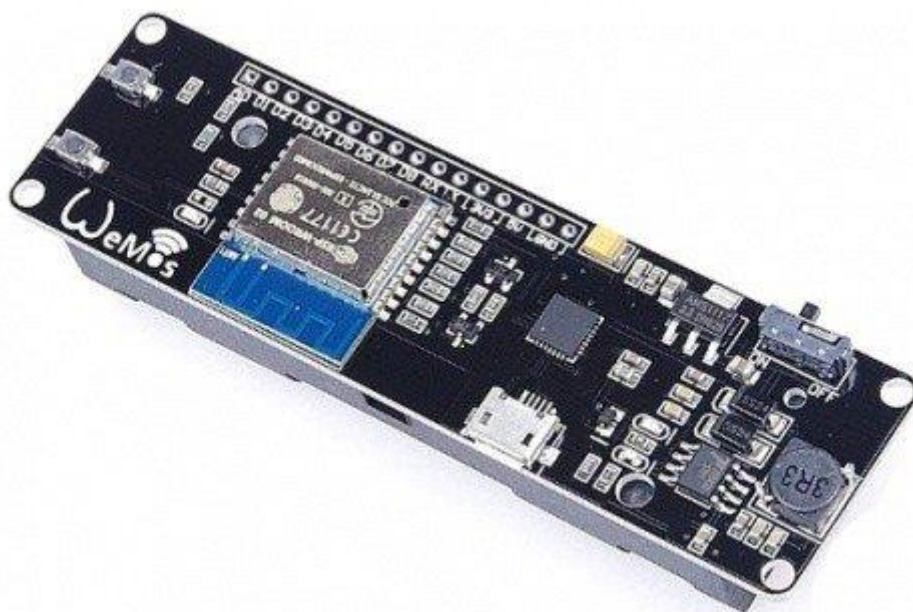
Ці рішення можуть варіюватися в ціні від кількох доларів за контролер до 35 доларів (або 50 доларів за версію з 4 ГБ пам'яті) на електронну плату.

Arduino - Найпопулярніша платформа DIY (Do It Yourself) для комп'ютерних систем. Існує багато типів плат Arduino, і особливою перевагою є набір різних щитів розширення з датчиками, інтерфейсами, реле управління і найширшим спектром додаткових пристроїв. Arduino Mega 2560 - одна з кращих плат мікроконтролерів від Arduino, заснована на потужному чипі ATmega2560 (16 МГц). Вона має 54 цифрових входу / виходу, 14 з яких можуть працювати в режимі ШІМ, 16 аналогових входів, 4 апаратних послідовних порту UART для зв'язку з комп'ютером та іншими пристроями, роз'єм USB і роз'єм зовнішнього живлення. Ще однією перевагою цього рішення є безкоштовна Arduino IDE, за допомогою якої ви можете писати прошивки контролера на мові програмування C. Крім того, він має величезну кількість нарощуваних плат розширення (GSM, GPS, PID контролери, датчики і т.д.).



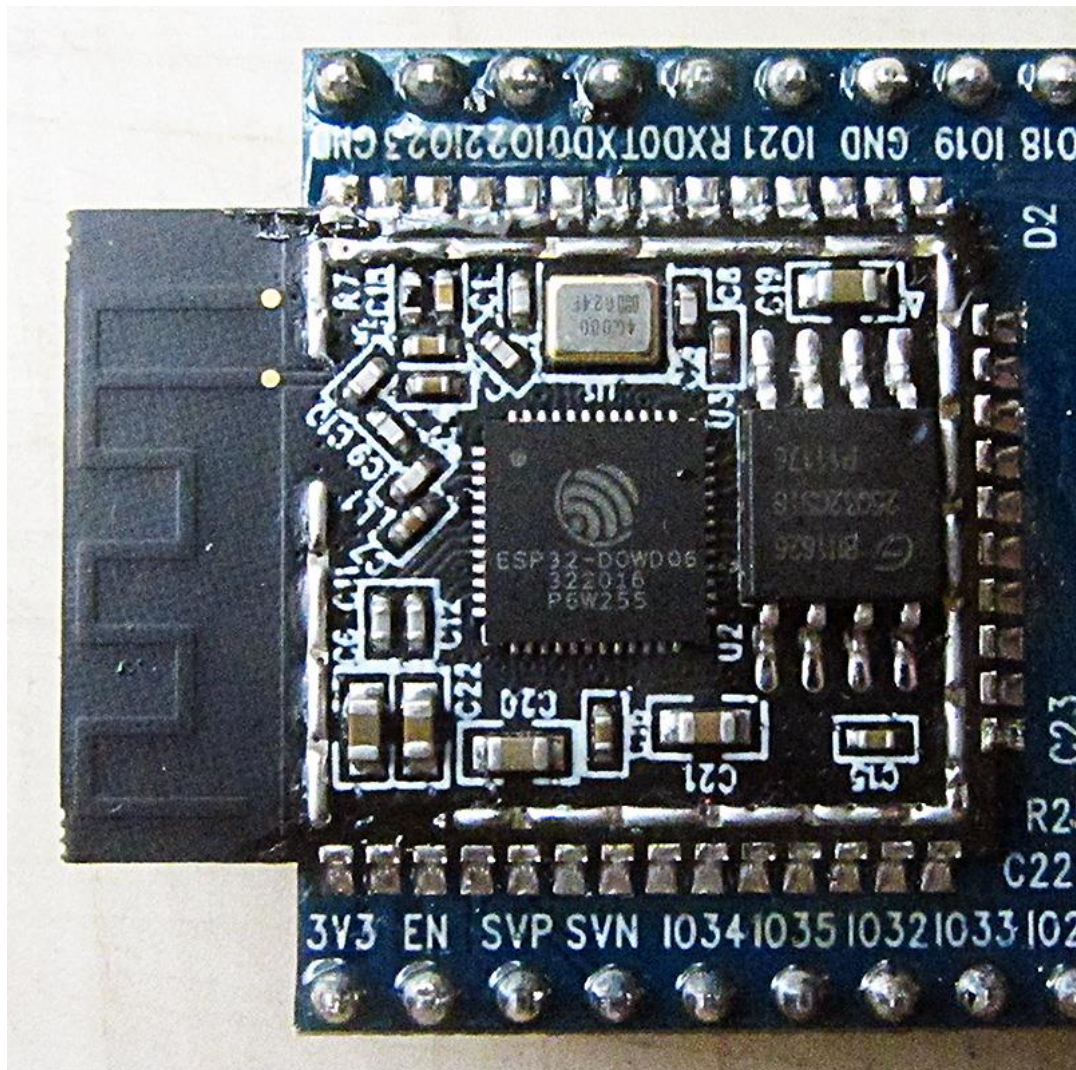
Малюнок 1 – Мікроконтролер Arduino Uno

ESP8266 - Найбільш доступним і цікавим рішенням є ESP8266. Це мініатюрний WiFi-модуль на основі чіпа ESP8266 з вбудованим стеком протоколів TCP / IP і командами управління AT. Чіп призначений для використання в інтелектуальних розетках, пористих мережах, IP-камерах, бездротових датчиках, що носиться електроніці і т.д. Цей модуль підтримує Wi-Fi-протоколи 802.11 b / g / n, Wi-Fi Direct (P2P), програмну точку доступу, SDIO 2.0, SPI, UART, STBC, 1x1 MIMO, 2x1 MIMO зі споживанням в режимі очікування до 1,0 мВт (DTIM3) і має розміри 24,5x14 мм. Плата успішно працює як клієнт Wi-Fi, може працювати як Soft-AP, і ви можете побудувати TCP-сервер на модулі для прийому і відправки даних або TCP-клієнт.



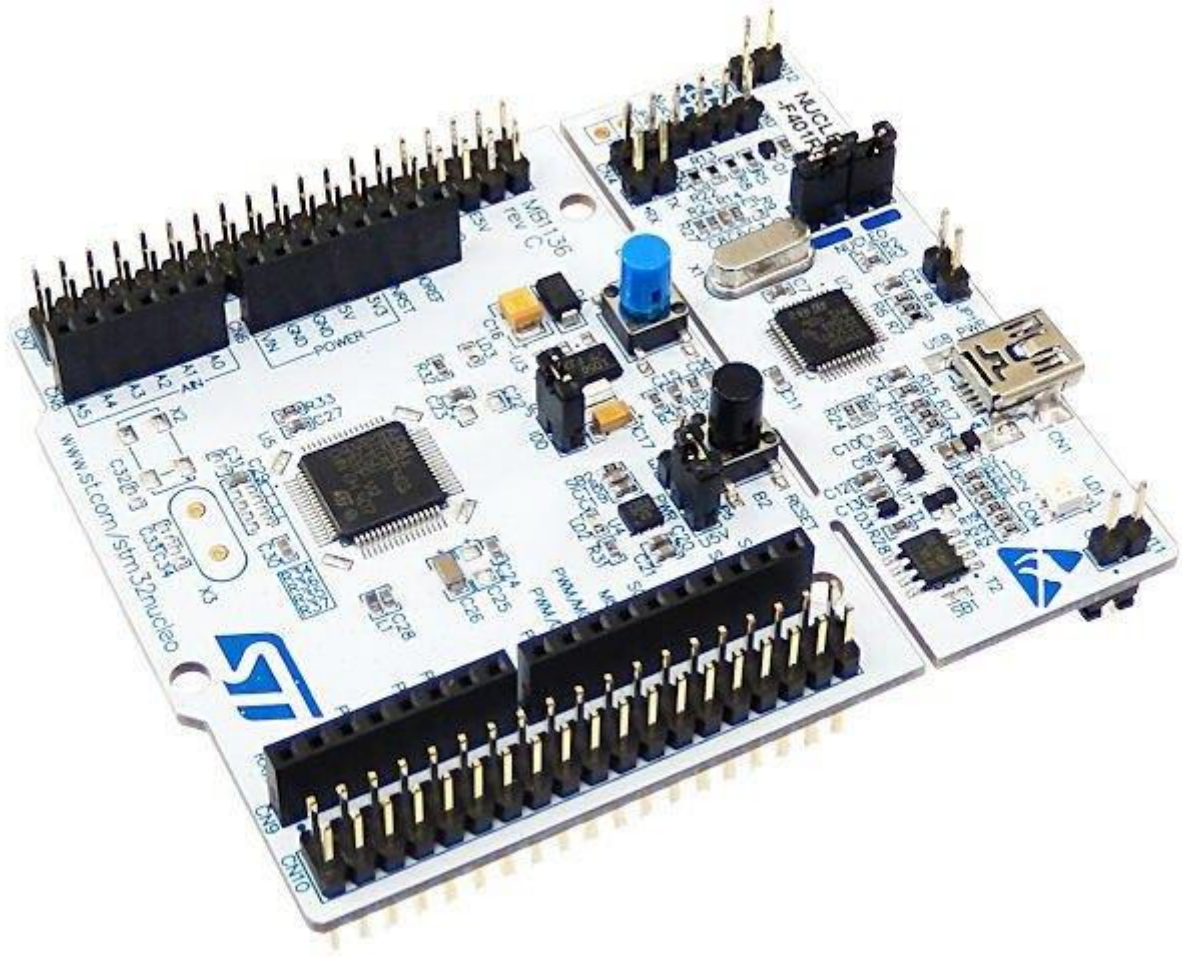
Малюнок 2 – Мікроконтролер ESP8266

ESP32 - це відмінний мікроконтролер з двоядерним 32-розрядним процесором, тактовою частотою 160 або 240 МГц і продуктивністю до 600 DMIPS. Цей мікроконтролер має підтримку дворежимного Bluetooth: «класичний» та BLE, швидкість Wi-Fi: 802.11 b / g / n до 150 Мбіт / с, підтримка режиму Wi-Fi: клієнт, точка доступу, Wi-Fi Direct з мінімальною чутливістю становить - 98 дБм, і широкий діапазон робочих температур: $-40^{\circ}\text{C} \dots +125^{\circ}\text{C}$, при якому споживана потужність ланцюга може впасти до 5 мКА (режим глибокого сну). Mongoose OS - це операційна система для плат мікроконтролерів ESP32, яка програмується на JavaScript або C. Це рекомендована платформа для систем Espressif, AWS IoT і Google Cloud IoT. Дуже цікаве рішення на основі ESP32 - плата M5Stack, яку можна нарощувати в стеці. Це дозволяє створити дуже цікавий додаток з сполуками GSM / Інтернет, дисплеями, блоком живлення і т.д. Воно працює як плати розширення Arduino, але розроблено спеціально для ESP32.



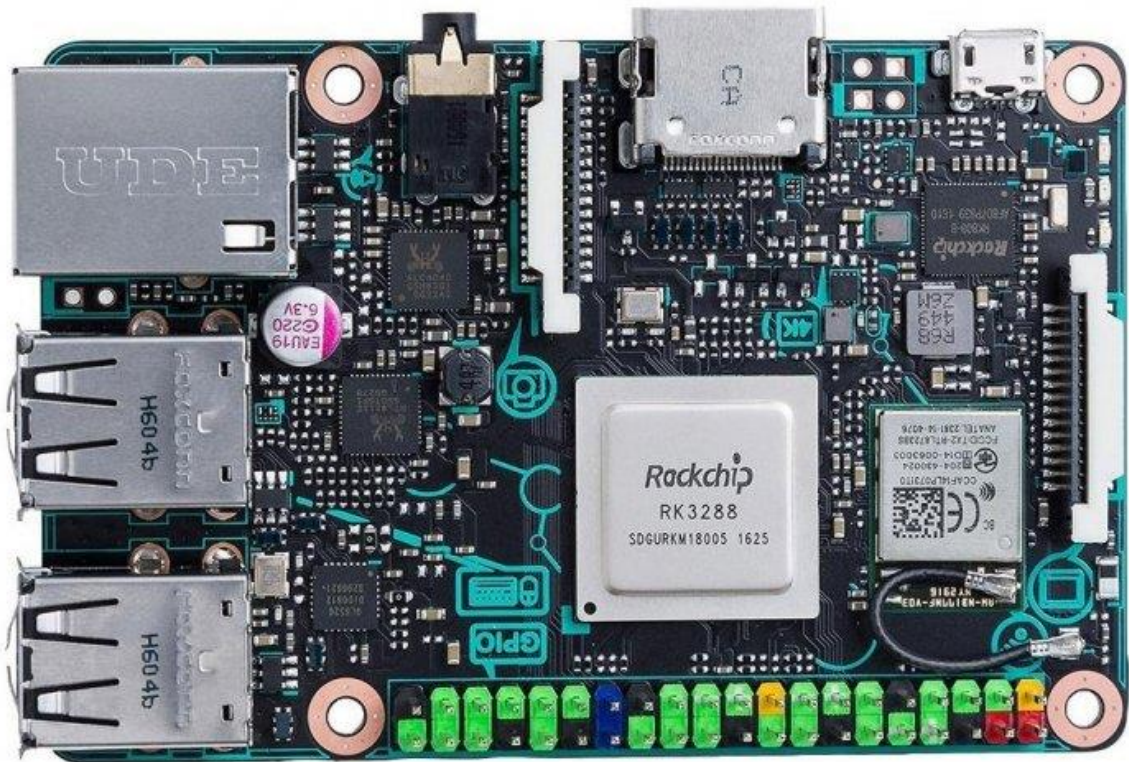
Малюнок 3 - Мікроконтролер ESP32

STM - Якщо потрібно більш потужне рішення, ніж Arduino, можна використовувати STM - сімейство мікроконтролерів з низьким енергоспоживанням. Наприклад, плата розробки STM32 Nucleo може підключати екрани від Arduino. Плата виконана на 32-бітному ARM мікроконтролері STM32F411. Nucleo - це сімейство плат, що підтримують середу онлайн-розробки mbed. Ця спеціальна плата складається з ядра контролера Cortex-M4 100 МГц, 512 КБ пам'яті програм, пам'яті даних 128 КБ і вбудованого програматора.



Малюнок 4 - Мікроконтролер STM32

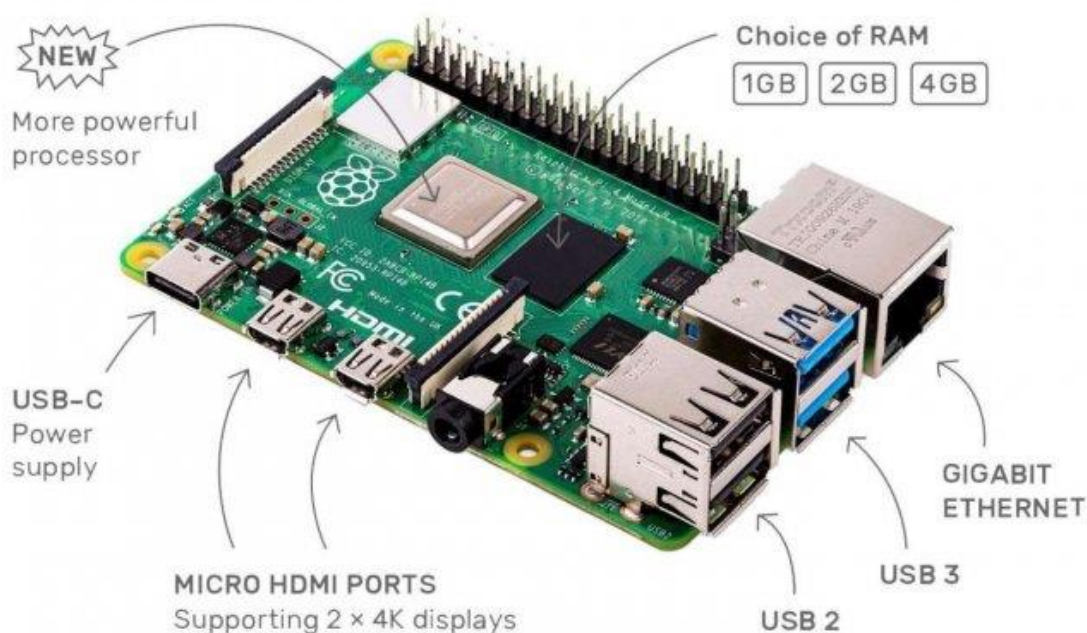
Asus Tinker Board S - високопродуктивний одноплатний комп'ютер від світового бренду. Побудований на основі потужного 4-ядерного Rockchip RK3288 з ядрами Cortex-A17, він має графіку Mali-T764. Плата Tinker оснащена 2 ГБ оперативної пам'яті LPDDR3, гігабітним Ethernet, вбудованим WiFi і Bluetooth 4.0. Плата підтримує апаратне прискорення при відтворенні відео в форматах H.264 і H.265 і має аудіокодек HD Audio з підтримкою форматів до 192 кГц / 24 біта. Завдяки своїм чудовим функцій ASUS Tinker Board може вирішувати багато завдань: стати висококласним медиacentром, ігровим автоматом, домашнім сервером, а також незамінним помічником мініатюри в промислових і розважальних програмах.



Малюнок 5 – Плата Asus Thinker

Raspberry Pi - це мікрокомп'ютер від Raspberry Pi Foundation, який працює на основі оновленого 4-ядерного 64-розрядної SoC Broadcom BCM2837B0 і збільшеною тактовою частотою 1,4 ГГц. Бездротовий модуль також отримав оновлення: Wi-Fi став дводіапазонним стандартом IEEE 802.11ac і Bluetooth 4.2 BLE. Нова плата має Gigabit Ethernet, що працює через шину USB 2.0, яка забезпечить швидкість передачі даних до 300 Мбіт / с. У першому збережений обсяг оперативної пам'яті - 1 Гб оперативної пам'яті, форм-фактор і розташування всіх портів, що забезпечує сумісність з більшістю існуючих аксесуарів. Raspberry Pi 3 Model B+, як і його попередники, відрізняється високим рівнем надійності, простотою настройки, величезним співтовариством і високою якістю виконання. Комп'ютер з розміром кредитної картки став ще менше в версії Raspberry Pi Zero V1.3 і все ще є платою, яку ви знаєте і любите, але зі зменшеним розміром до 65 мм в довжину і 30 мм в ширину і за дуже привабливою ціною. Raspberry Pi Zero ідеально підходить для створення вбудованого онлайн-проекту Things (IoT). На платі Pi Zero V1.3 основні міні-

роз'єми і непаяного висновки GPIO розташовані компактно, що дозволяє використовувати тільки те, що потрібно для вашого проекту. В оновленій версії (1.3) доданий CSI-роз'єм для підключення пі-відеокамер. Але це ще не все. Нещодавно вийшла мікрокомп'ютерних плата Raspberry Pi 4 з харчуванням USB-C, 4 ГБ пам'яті, можливістю підключення дисків M.2, потужним процесором, двома мікро HDMI, виведенням зображення на два монітори 4К, двома USB 3 і Гігабітний Ethernet. Версія з 1 ГБ пам'яті коштує 35 доларів, а версія з 4 ГБ пам'яті - 55 доларів. Це фантастична можливість використовувати Linux, Android або іншу операційну систему і замінити настільний комп'ютер або медіацентр.



Малюнок 6 – Плата Raspberry Pi 4

1.3 Операційна система Raspbian

Raspbian - заснована на Debian операційна система для Raspberry Pi. Існує кілька версій Raspbian, в тому числі Raspbian Stretch і Raspbian Jessie. З 2015 року Raspbian офіційно представлена Raspberry Pi Foundation в якості основної операційної системи для одноплатних комп'ютерів Raspberry Pi. Raspbian була створена Майком Томпсоном і Пітером Грінном в якості незалежного проекту. Первісна збірка була виконана в червні 2012 року. Операційна система знаходиться в стадії активної розробки. Raspbian оптимізована для

нізкопроїзводительних процесорів ARM, які використовуються в лінійці комп'ютерів Raspberry Pi.

В якості основної середовища робочого столу в Raspbian використовується PIXEL (Pi Improved Xwindows Environment, Lightweight). Вона складається з модифікованої середовища робочого столу LXDE і менеджера вікон Openbox з новою темою і декількома іншими змінами. PIXEL є відгалуженням робочого столу LXDE. Завдання розробників при його створенні була в тому, щоб дати користувачам легку і швидку оболонку, з якої буде просто працювати. Особливо це стосується новачків, які раніше не мали справ з Linux.

Raspbian заснована на ARM версії Debian 8 Jessie і містить такі програми за замовчуванням - офісний пакет LibreOffice, веб-браузер, поштовий клієнт - Claws Mail, легке оточення робочого столу, а також деякі інструменти для навчання програмуванню.

Рекомендованої операційною системою для Raspberry Pi 4 є Raspbian.

Raspbian - це модифікація популярного дистрибутива GNU / Linux Debian. Він має безліч відмінних рис від свого прабатька. З усіма ними можна ознайомитися в офіційній "Вікі" проекту. А тут є сенс перерахувати лише основні з них:

- можливість роботи на ARM-процесорах;
- наявність в комплекті всіх необхідних драйверів;
- гарна оптимізація під порівняно невеликий обсяг ОЗУ;
- підтримка GPIO з коробки.

Дистрибутив Raspbian не прив'язаний до конкретної моделі Pi, більше того, в комплекті з ним поставляються драйвери схвалених Raspberry PI Foundation пристроїв, включаючи Wi-Fi-донгл Pi USB, модулі камер (v1 і v2), плату розширення Raspberry Pi Sense HAT. Raspbian додатково комплектується перевіреним ПО та утилітами. Як приклади можна привести диспетчер WiFi-мережі, менеджер Bluetooth, утиліти для роботи з камерами і різними бібліотеками, які потрібні для програмування різних модулів.

За словами Уотсона, програми для Raspberry Pi відібрані і модифіковані розробниками. Це зроблено для того, щоб використовувати ресурси

малопотужного комп'ютера по максимуму. Власне, доробка програм для Raspberry Pi принесла свої плоди - продуктивність цього пристрою в зв'язці з Raspbian набагато вище (в деяких випадках - на порядки), ніж інших протестованих їм дистрибутивів Linux, пише автор огляду. Окремо відзначається стабільність дистрибутива: «П'ять років розробки, тестування і налагодження при тісній взаємодії з користувачами очікувано привнесли в роботу Raspbian стабільність. Вона сприймається як щось само собою зрозуміле: Raspbian практично ніколи не зависає і робить те, що від нього вимагається. Припустимо, існують деякі додатки або утиліти, які можуть викликати періодичні зависання або навіть крах системи, але мені такі зустрічалися».

Важливо підкреслити, що встановлення Raspbian відрізняється простотою - це стосується як самої завантаження (на відміну від інших дистрибутивів, де потрібно пройти реєстрацію або інші необов'язкові дії), так і «залівки» на карту пам'яті. Причому це стосується будь-яких моделей Pi - будь-то Pi Zero або Pi 4. Під час розпакування образу завантажувач в автоматичному режимі визначає, на яку модель встановлюється Raspbian і зробить відповідні налаштування і оптимізації.

1.4 Веб браузер і веб розробка

1.4.1 Веб Браузер

Браузер - це програма, що представляє в зручному для сприйняття вигляді інформацію, що отримується з Інтернету. Це інструмент для перегляду ресурсів Мережі і, в меншій мірі, для взаємодії з ними. Втім, говорити про те, що браузер потрібен тільки в Інтернеті, було б невірно. У наш час на технологіях Інтернету ґрунтуються багато локальних мереж, звані інтранетом або інтра мережами. Вони можуть і не мати виходу в Інтернет.

Дивне на перший погляд слово «браузер» походить від англійського слова browse, що означає «гортати» (це слово можна побачити на кнопках в діалогових вікнах англійських програм). Ця назва програма отримала за кордоном ще до виникнення системи World Wide Web в той час, коли, крім тексту, переглядати в Мережі було нічого, і користувачі «гортали» текстові файли на екранах

комп'ютерів.

Зараз Інтернет пропонує нам дивовижне розмаїття інформації - текст, графіку, анімацію, відео, звук, і все це пишність ми бачимо і чуємо завдяки браузеру.

Компанія Microsoft при локалізації своїх програм вводить нові терміни, які з часом можуть стати стандартом завдяки поширеності продуктів цієї фірми. Оскільки з деяких пір кнопка Browse у всіх україномовних програмах Microsoft стала іменуватися Огляд, програма перегляду Інтернет-ресурсів набула назву «оглядач».

Інформація, яку браузер передає на сервер в якості запиту і яку отримує від нього у відповідь, передається, як правило, по протоколу HTTP.

Web-сторінки, які ви бачите на екрані, блискучі різноманіттям кольорів і форм представлених на них об'єктів, на ділі є всього лише текстові файли, в яких містяться певні інструкції. Ці інструкції наказують браузеру, що і як слід показувати на даній сторінці. У найпростішому випадку вихідний текст Web-сторінки пишуться на мові HTML. Внаслідок своєї складної історії мову HTML неоднозначно трактується різними браузерами. Та й самі браузери можуть призначатися для різних цілей. В результаті в різних браузерах одні й ті ж сторінки можуть виглядати по-різному.

Виділяють наступні види браузерів:

- браузер режиму командного рядка. До цього типу відносяться самі ранні браузери. Вони не дають можливості переглядати текст і графіку. Такі браузери підтримують переміщення тільки з використанням цифрових адрес (IP). В даний час практично не використовуються;
- повноекранний браузер. Текстовий браузер без підтримки мультимедійних (картинки, анімація і т.п.) ресурсів мережі Інтернет. За допомогою нього можна переглядати тільки текст і посилання;
- браузер з підтримкою мультимедіа. Найпоширеніші і популярні браузери сьогодні. Дозволяють працювати практично з усіма видами інформації, представленої в Інтернеті;

- браузер-доповнення. Вони є надбудовами над повнофункціональними браузерами. Найчастіше розробниками додатків використовується Google Chrome і Mozilla Firefox. Надбудови використовують для відображення сайтів "движок" цього браузера. Тому їх можливості в цій області повністю ідентичні з движком браузера. Додатки всього лише змінюють інтерфейс і додають деякі функції, які розробники браузерів обійшли своєю увагою.

Також виділяють оффлайн браузери. Це програми, котрає автоматично завантажує інформацію з Інтернету і зберігають її на локальному диску комп'ютера для подальшого перегляду та аналізу. Ідея, що лежить в основі роботи оффлайн браузерів, досить проста. Користувач вказує адресу цікавить його Web-сайту, а програма у відповідь завантажує на його комп'ютер всі файли, які необхідні для автономного (тобто відключилися від мережі інтернет) перегляду цього сайту. Переваги такого способу роботи в наявності - в автономному режимі зовсім необов'язково мати доступ в інтернет, інформація з жорсткого диска завантажується дуже швидко, а локальна копія сайту повністю "оглядача" - зберігається структура і взаємозв'язок HTML-документів.

1.4.2 HTML

Термін HTML (Hyper Text Markup Language) означає "мова маркування гіпертекстів". Це поняття більш широке, включає в себе Інтернет і локальні мережі, редактори, браузери, різноманітні програмні продукти, компакт-диски, навчальні курси, дизайн і багато іншого. HTML - своєрідна протилежність складним мовам програмування, відомим тільки фахівцям.

HTML давно перестав бути просто мовою програмування. Людина, який вивчав цю мову, знаходить можливість робити складні речі простими способами і, головне, швидко, що в комп'ютерному світі не так вже й мало. Гіпертекст підходить для включення елементів мультимедіа в традиційні документи. Практично саме завдяки розвитку гіпертексту, більшість користувачів отримало можливість створювати власні мультимедійні продукти і поширювати їх на

компакт-дисках. Такі інформаційні системи, виконані у вигляді набору HTML-сторінок, не вимагає розробки спеціальних програмних засобів, так як всі необхідні інструменти для роботи з даними (WEB-браузери) стали частиною стандартного програмного забезпечення більшості персональних комп'ютерів. Від користувача потрібно лише ту роботу, яка відноситься до тематики продукту, що розробляється: підготувати тексти, намалювати малюнки, створити HTML-сторінки і продумати зв'язок між ними.

HTML, як основа створення WEB-сторінок, має пряме відношення і до нового напрямку образотворчого мистецтва - WEB-дизайн. Художнику в Інтернеті недостатньо просто намалювати красиві картинки, оригінальний логотип, створити новий фірмовий стиль. Він повинен ще помістити все це в мережі інтернет, продумати зв'язок між WEB-сторінками, щоб все рухалося, відгукувалася на дію користувача, вражало уяву, викликало бажання створити що-небудь своє, оригінальне в цій області.

Перша версія HTML була розроблена в 1989 році Тімом Бенерс-Лі для популярного в минулому браузера Mosaic. Але в той час ні для мови, ні для браузера не знайшлося гідного застосування. У 1993 році з'явився HTML+, і ця версія також залишилася практично непоміченою. Початок широкого використання гіпертексту дала версія 2.0, яка, з'явилася в червні 1994 року. Це був рік зростання популярності WWW по всьому світу. Елементи, включені в версію 2, в більшості своїй використовуються донині.

У версії 3.0 HTML, яка з'явилася через рік, була реалізована можливість промальовування математичних символів (знаків інтервалу, нескінченності, дробу, дужок і т.д.) за допомогою елементів мови. Під цю версію було розроблено браузер Arena. Але цей проект виявився тупиковим і не отримав подальшого поширення.

У 1996 році з'явився HTML 3.2. Це було новаторське рішення, в специфікацію мови були введені фрейми, які стали тепер дуже популярні у розробників WEB-сторінок. Навіть зараз на основі цієї специфікації можна реалізувати цікаві дизайнерські рішення. Практично всі сучасні браузери підтримують версію 3.2, тому автори WEB-сторінок впевнені в працездатності

всіх елементів.

Поряд з офіційними специфікаціями мови, які розроблялися організацією W3C (W3 Консорціум), компанії-виробники браузерів створювали власні елементи (розширення). Згодом, деякі з цих елементів, після отримання загального визнання включилися в специфікацію наступних версій мови. Але новаторське рішення - фрейми, які не були включені в специфікацію 3.2. Але браузери підтримували фрейми і багато книги, присвячені HTML, містили опис фреймів без згадки про те, що це нестандартні елементи. Надалі, фрейми стали стандартом де-факто. У версії 4 вони вже були включені на цілковитій підставі.

І навпаки, елементи APPLET і SCRIPT, необхідні для розширення HTML іншими програмними кодами версії 3.2, не зіграли тієї ролі, яку були покликані зіграти. Це пояснюється тим, що браузери різних версій по-різному інтерпретували програми на різних мовах JAVA, JAVASCRIPT, Visual Basic (VBScript). В результаті не вдалося отримати досить надійний працюючий код, і дані мови використовувалися любителями HTML в основному для експериментів.

Офіційна специфікація HTML 4 (Dynamic HTML) з'явилася в 1997 році. У цей час уже було очевидно, що подальший розвиток гіпертексту буде здійснюватися за рахунок скрипт - програмування. Це виявилось трохи більше ефективним, ніж вводити в мову всі нові елементи. З'явилися в той час браузери (Netscape Navigator 4, Microsoft Internet Explorer 4 і ін.) Вже досить надійно інтерпретували програмний код (був встановлений певний рівень стандартизації). Однак проблеми у розробників ще залишилися. Як приклад можна відзначити, що багато скрипти починаються з визначення версії браузера, щоб потім використовувати той чи інший фрагмент коду. Очевидно, що на програміста лягає обов'язок тестування сторінок на всіх популярних в теперішній час браузерах.

В результаті, використання усіх можливостей Dynamic HTML стало долею програмістів досить великих організацій, де є умови для розробки складних програм і всебічного їх тестування. Творцям особистих WEB-сторінок часом доводиться шукати компроміс між надійністю і новаторством, щоб отримати

досить грамотний HTML-код.

1.4.3 CSS

Вперше каскадні таблиці стилів CSS були реалізовані в браузері Internet Explorer 3.0. Однак в той час розвиток CSS знаходилося в зародковому стані, тому правила складання стильових шаблонів були вельми розрізненими.

З моменту свого виникнення структура CSS була кілька разів переглянута, в неї були додані нові елементи і прибрані (видозмінені) старі. Існують три рівні CSS, які визначаються наявністю завершеною редакції структури. Це: CSS 1 (перший рівень структури стильових шаблонів, остаточно затверджений 11 січня 1999 роки), CSS 2 (другий рівень стильових конструкцій, початок обговорення якого датується травнем 1998 року) і CSS 3 (третій рівень стильового оформлення електронних документів, прийнятий до обговорення 23 травня 2001 року, на момент написання книги знаходився в стадії опрацювання). Слід додати, що перехід від одного рівня до іншого, в основному, супроводжувався деякими видозмінами в структурі і в правилах стильового оформлення, технологічними доповненнями, а також спробами систематизувати застосування CSS.

Саме третій рівень (CSS 3) позиціонується розробниками як якоїсь єдиної системи подання стилів в електронному документі, заснованої на використанні спеціальних модулів.

Способи визначення таблиць стилів. Як вже було сказано, будь-яка таблиця стилів CSS повинна бути інтерпретована браузером для того, щоб правила CSS, позначені для конкретних елементів електронного документа, вступили в силу.

1.4.4 JavaScript

Гіпертекстова інформаційна система складається з безлічі інформаційних вузлів, безлічі гіпертекстових зв'язків, визначених на цих вузлах і інструментах маніпулювання вузлами і зв'язками. Технологія World Wide Web - це технологія ведення гіпертекстових розподілених систем в мережі інтернет, і, отже, вона повинна відповідати загальним визначенням таких систем. Це означає, що всі

перераховані вище компоненти гіпертекстової системи повинні бути і в Web.

Web, як гіпертекстову систему, можна розглядати з двох точок зору. По-перше, як сукупність відображуваних сторінок, пов'язаних гіпертекстовими переходами (посиланнями - контейнер ANCHOR). По-друге, як безліч елементарних інформаційних об'єктів, що становлять відображаються сторінки (текст, графіка, мобільний код і т.п.). В останньому випадку безліч гіпертекстових переходів сторінки - це такий же інформаційний фрагмент, як і вбудована в текст картинка.

При другому підході гіпертекстова мережа визначається на безлічі елементарних інформаційних об'єктів самими HTML-сторінками, які і грають роль гіпертекстових зв'язків. Цей підхід більш продуктивний з точки зору побудови відображуваних сторінок "на льоту" з готових компонентів.

При генерації сторінок в Web виникає дилема, пов'язана з архітектурою "клієнт-сервер". Сторінки можна генерувати як на стороні клієнта, так і на стороні сервера. У 1995 році фахівці компанії Netscape створили механізм управління сторінками на клієнтській стороні, розробивши мову програмування JavaScript.

Таким чином, JavaScript - це мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні клієнта. Якщо бути більш точним, то JavaScript - це не тільки мова програмування на стороні клієнта. Liveware, прабатько JavaScript, є засобом підстановок на стороні сервера Netscape. Однак найбільшу популярність JavaScript забезпечило програмування на стороні клієнта, але також код JavaScript може виконуватися на платформі NodeJS як на стороні сервера з деякими обмеженнями.

Основна ідея JavaScript полягає в можливості зміни значень атрибутів HTML-контейнерів і властивостей середовища відображення в процесі перегляду HTML-сторінки користувачем. При цьому перезавантаження сторінки не відбувається.

На практиці це виражається в тому, що можна, наприклад, змінити колір фону сторінки або інтегровану в документ картинку, відкрити нове вікно або видати попередження.

JavaScript стандартизований ECMA (European Computer Manufacturers Association - Асоціація європейських виробників комп'ютерів). Відповідні стандарти носять назви ECMA-262 і ISO-16262. Ці стандарти вимагають визначається мова ECMAScript, який приблизно еквівалентний JavaScript 1.1. Відзначимо, що не всі реалізації JavaScript на сьогодні повністю відповідають стандарту ECMA.

1.4.5 Платформа NodeJS

Програмна крос-платформа, вперше випущена в 2009 році, написана на C++, JavaScript та C. Вона працює на движку V8, тим самим перетворює його з вузькоспеціалізованого мови в мову загального призначення. До скриптова мова NodeJS додає можливість через API взаємодіяти з пристроями введення і виведення, підключати зовнішні бібліотеки, написані на різних мовах. В його основі закладено подієво-орієнтоване і асинхронне програмування з неблокуючим введенням та виведенням.

V8 - движок з відкритим програмним кодом для JavaScript, розроблений одним з відділень Google з використанням C++ і JavaScript. Розповсюджується за ліцензією BSD. Остання його версія 8.3.

Особливості V8:

- компіляція вихідного коду відбувається безпосередньо в машинний код;
- ефективна система управління пам'яті, яка призводить до невеликих пауз збірки сміття і швидкому виділенню об'єктів;
- введення вбудованих кешей, які прискорюють доступ до властивостей і викликам функцій;
- більш висока продуктивність в порівнянні з іншими (JScript, JavaScriptCore).

У 2014 році було створено незалежне відгалуження io.js. Головна його відмінність від NodeJS - оновлений V8. У 2015 було прийнято рішення про злиття io.js і NodeJS для подальшого спільного розвитку під заступництвом Node.js Foundation. На даний момент розробники надають 2 версії продукту: 12.16.3 -

версія з довгостроковою підтримкою і 14.2.0 - остання випущена версія.

Установка і вибірковий пошук додаткових пакетів проводиться менеджером npm (остання версія 6.14.5). Він має 2 утиліти: `install` для установки і `search` для пошуку.

Одним з переваг NodeJS є асинхронність. Вона надає можливість виконувати кілька завдань одночасно. Це дуже продуктивно, адже поки сервер робить запит до бази даних, він одночасно може і обробляти інші запити і повертати вже сформовані відповіді.

1.5 Вибір апаратної бази електронного органайзера

В даний час інженери-любители, які захоплюються конструюванням в дусі «зроби сам», не відчувають нестачі в спеціальних пристроях, що дозволяють обладнати будь-які вироби хорошою електронною начинкою. Одне з найпопулярніших рішень такого роду - це дешевий мікрокомп'ютер Raspberry Pi, система на кристалі (SoC), яка використовує повну версію ОС Linux (цей комп'ютер розроблявся в навчальних цілях). Також існує платформа Arduino - мікроконтролер, що володіє значною технічною підтримкою (ціле співтовариство розробників) і має сотні схем-розширень (так званих «ШІлд»).

Після анонса про появу комп'ютера Raspberry Pi багато могли подумати, що платформа Arduino морально застаріла, однак цілком імовірно, що це занадто поспішне судження. Насправді, ніші застосування обох пристроїв відрізняються, у кожного з них є свої переваги і недоліки, а також спектр завдань, які вирішуються по-різному. Крім того, якщо спиратися лише на сухі дані специфікацій, то пряме порівняння двох платформ виявляється дуже неоднозначним, враховуючи, що процесор Arduino з частотою 16 MHz явно поступається процесору Pi з частотою 900 MHz.

Raspberry Pi - Безумовно, Raspberry Pi є повнофункціональним комп'ютером. Він володіє всіма атрибутами справжнього комп'ютера: виділеним процесором, пам'яттю і графічним драйвером для виведення через HDMI. На ньому навіть працює спеціальна версія операційної системи Linux. Тому на Raspberry Pi легко встановити більшість програм для Linux. Варто трохи

попрацювати - і Raspberry Pi можна використовувати як повноцінний медіа-сервер або емулятор відеоігор.

Хоча в Pi і відсутня внутрішнє сховище даних, на цьому комп'ютері можна використовувати смарт-карти в якості флеш-пам'яті, яка обслуговує всю систему. Таким чином, можна швидко вивантажувати для налагодження різні версії операційної системи або програмних оновлень. Оскільки цей пристрій забезпечує незалежну з'єднуваність по мережі, його можна налаштувати і для доступу по SSH, або пересилати на нього файли по протоколу FTP.



Малюнок 7 – Плата Raspberry Pi

Плати Arduino - це мікроконтролери, а не повноцінні комп'ютери. На них немає операційної системи як такої, Arduino просто виконує код, що інтерпретується прошивкою. В даному випадку ви не маєте в розпорядженні базових інструментів, що надаються операційною системою, але, з іншого боку, таке безпосереднє виконання нескладного коду протікає простіше, а при роботі не виникає ніяких витрат, пов'язаних з операційною системою.

Основне призначення плати Arduino - взаємодія з сенсорами і пристроями, тому Arduino відмінно підходить для апаратних проектів, де потрібно просто реагувати на різні сигнали сенсорів і ручне введення. Може здатися, що в цьому немає нічого особливого, проте на ділі Arduino - складна вивірена система, значно полегшує управління пристроями. Вона відмінно підходить саме для зчленування інших пристроїв і виконавчих механізмів, де повновага операційна система просто не потрібно, так як мова йде просто про реєстрацію дій і реагування на них.



Малюнок 8 - Мікроконтролер Arduino Uno

Живлення. Вимоги до електроживлення для цих двох систем дуже відрізняються. Raspberry Pi для роботи потрібно постійна напруга 5V, більш того, робота Raspberry Pi завершується програмним процесом - як у звичайного комп'ютера. Arduino, в свою чергу, починає виконувати код відразу після включення і припиняє роботу, коли ви виймаєте штепсель з розетки. Щоб розширити функціонал пристрою з Arduino, ви підключаєте пристрій або безпосередньо до штирьковим контактам самої плати Arduino або до платам розширень (Шілд) для неї. Існують сотні різноманітних Шілд, кожен з яких призначений для вирішення специфічного завдання, може взаємодіяти з тими чи іншими сенсорами, а також з іншими Шілд, які разом утворюють повноцінний керуючий блок.

Raspberry Pi складно переносити з місця на місце, так як ви не зможете просто вставити в нього дві батарейки AA. Для роботи цього комп'ютера необхідно забезпечити безперебійне живлення, а також підключити додаткове обладнання, яке гарантує подачу постійного струму. У випадку з Arduino весь процес дещо спрощується, так як система вимагає всього лише комплекту батарей, що забезпечує напругу не нижче певного рівня; також потрібен простий Шілд для управління електроживленням. Навіть при аварійному відключенні струму на Arduino ви не ризикуєте ні пошкодити операційну систему, ні отримати будь-які програмні помилки. Варто заново підключити Arduino до джерела енергії - і прилад просто відновить роботу.

Мережа. У Raspberry Pi є вбудований Ethernet-порт, який забезпечує легкий доступ до будь-якої мережі і практично не вимагає налаштування. Провести бездротовий Інтернет на Raspberry Pi також не складає труднощів, сучасні плати Raspberry Pi мають вбудований модуль WiFi. Як тільки це зроблено, можете використовувати операційну систему для підключення до веб-серверів, обробляти HTML або просто що-небудь писати в Інтернеті. Можете навіть використовувати Raspberry Pi для створення віртуальної приватної мережі або в якості сервера друку.

На жаль, система Arduino без додаткових модифікацій не пристосована для роботи по мережі. З нею потрібно як слід повозитися, щоб встановити надійне з'єднання, але це цілком можливо. Вам потрібна додаткова схема, оснащена Ethernet-портом, також знадобиться підключити деякі кабелі і написати потрібний код, щоб все запрацювало. Загалом, це досить складна робота, тому деякі компанії випускають платформи Arduino з уже вбудованим функціоналом Ethernet.

Сенсори. Як Raspberry Pi, так і Arduino володіють набором інтерфейсних портів, проте аналогові сенсори набагато простіше підключати саме до Arduino. Мікроконтролер з легкістю інтерпретує різні сигнали на основі написаного вами коду і реагує на них. Тому Arduino відмінно підходить для тих випадків, коли ви плануєте виконувати серії команд або реагувати на показання сенсорів, в залежності від яких буде коригуватися робота сервоприводів і пристроїв.

У свою чергу, Raspberry Pi для ефективної взаємодії з такими пристроями вимагає спеціального програмного забезпечення - ймовірно, це зайві складності, якщо ви хочете просто автоматизувати поливання в саду або зберігати воду при потрібній температурі. У багатьох проектах Arduino і Raspberry Pi використовуються спільно, причому Arduino виступає в якості керуючої плати, на якій виконуються команди, що видаються ПО Raspberry Pi. Інформація з сенсорів подається на Raspberry Pi, де вона записується, або у відповідь на неї виконуються ті чи інші операції.

Таким чином, можна зробити висновок, що обидві платформи підходять для вибору за їх апаратну основу. Для комп'ютерної системи електронного органайзера цілком підійде мікрокомп'ютер Raspberry Pi, так як він має набагато більше можливостей для розширення, ніж плати Arduino, а також має підтримку повноцінних операційних систем, таку як Raspbian, яка в свою чергу має нативну підтримку веб браузерів, таких як Chromium та Mozilla Firefox.

В якості основної плати буде використаний мікрокомп'ютер Raspberry Pi 4 Model B.

Технічні характеристики плати:

- Процесор: SoC: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz;
- Оперативна пам'ять: 2GB LPDDR4-3200 SDRAM;
- Графічний процесор VideoCore VI з OpenGL ES 1.1, 2.0, 3.0;
- Порт Ethernet: Gigabit Ethernet (BCM54213PE);
- WiFi: Вбудований двохдіапазонний WiFi (2.4GHz / 5GHz) стандарту IEEE 802.11 b/g/n/ac;
- Bluetooth: Вбудований Bluetooth 5.0 BLE;
- Порти USB: USB 3.0 2шт, USB 2.0 2шт;
- Відеовихід: Два роз'єму microHDMI (дозволом до 4K при 60 FPS);
- Аудіовихід: Композитний 3.5 jack для виведення звуку / відео;
- Інтерфейс введення / виводу: 40 пінів GPIO (general-purpose input / output);
- Слот для карт microSD;
- Інтерфейс для підключення камери CSI;
- Інтерфейс для підключення дисплея DSI;
- Харчування 5V 3A через роз'єм USB Type-C;
- Габарити: 88x58 міліметрів.

Однак для повноцінної реалізації електронного органайзера знадобиться

додаткові компоненти:

- пристрій введення і виведення;
- захисний корпус;
- блок живлення;
- карта пам'яті формату MicroSD.

Як пристрій введення і виведення впав вибір на 3.5 дюймовий сенсорний TFT дисплей роздільною здатністю 320x480 пікселів, який може використовуватися з будь-якою платою Raspberry Pi. Він працює через SPI інтерфейс з частотою передачі даних в 125 МГц, що дозволяє оновлювати картинку з частотою в 50 кадрів в секунду.

Характеристики дисплея:

- Живлення: 5V / 0.13A;
- Роздільна здатність: 320x480;
- Частота оновлення кадру: 50 FPS;
- Кількість кольорів: 65535;
- Підсвічування: LED;
- Співвідношення сторін: 8.5;
- Розміри: 91x62x34 мм.

Захисний корпус необхідний для захисту плати Raspberry і TFT дисплея від різних пошкоджень або недбало поводження з системою. В якості захисного корпусу буде використаний корпус з ABS пластику, як показано на малюнку 9, який відмінно підходить для використання разом з TFT дисплеєм роздільною здатністю 320x480. Корпус має як гумові ніжки для портативного використання, так і спеціальні вирізи на нижній частині, які дозволяють кріпити його до різних поверхонь. Крім отворів для кріплення, в корпусі передбачені отвори для циркуляції повітря для охолодження плати.



Малюнок 9 – Захистний корпус Raspbian Pi Model 4

Так як плата Raspberry Pi не має автономного джерела живлення, будемо використовувати імпульсний блок живлення від розробників самої плати Raspberry Pi Foundation. Створено спеціально для міні-комп'ютерів Raspberry Pi 4.

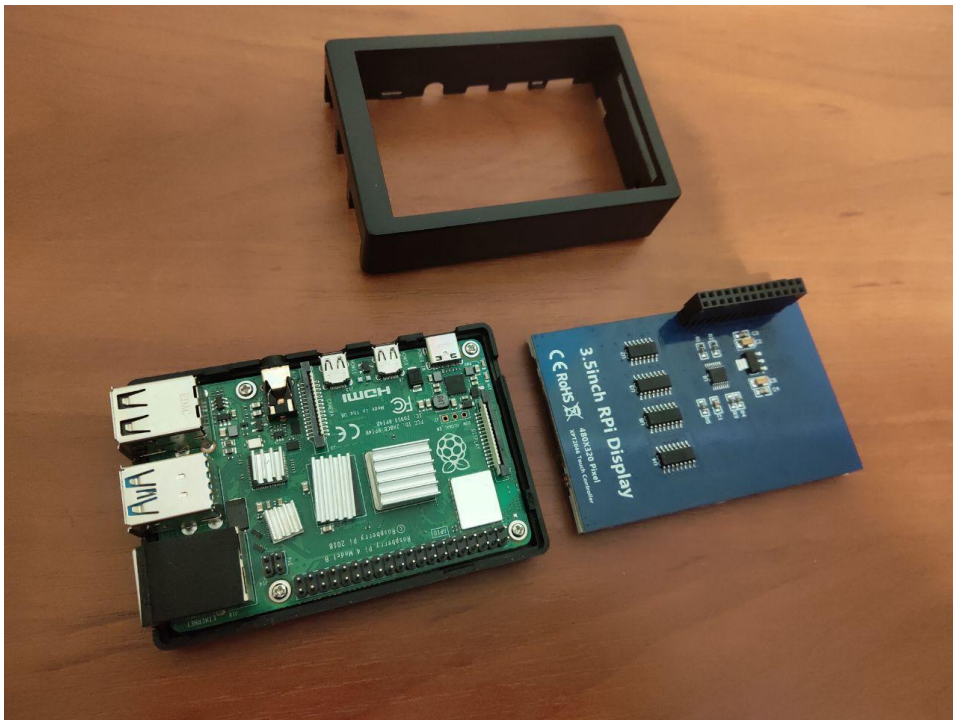
Блок живлення має наступні характеристики:

- Вихідна напруга: 5.1V;
- Вихідний струм, не більше: 3A;
- Вхідна напруга: 96 - 264 V;

- Захист від короткого замикання, перевантаження по струму і перегріву;
- Кабель: 1.5м 18 AWG з роз'ємом USB Type-C;
- Штекер: EU.

Як карти пам'яті і основного сховища даних системи будемо використовувати флеш карту ємністю 8 гігабайт, Такого розміру карти досить щоб встановити операційну систему Raspbian і також залишиться досить кількості вільного місця для програмного коду та додаткового програмного забезпечення.

Таких чином апаратна частина виглядає як зображено на малюнках 10, 11 та 12.



Малюнок 10 – Плата Raspberry Pi Model 4, захистний корпус, TFT дисплей



Малюнок 11 - Плата Raspberry Pi Model 4, захистний корпус, TFT дисплей



Малюнок 12 - Плата Raspberry Pi Model 4, захистний корпус, TFT дисплей

1.6 Вибір програмної частини електронного органайзера

Операційна система Raspbian, яка розроблена спеціально розроблена для одноплатних мікрокомп'ютерів Raspberry Pi, відмінно підходить для комп'ютерної системи електронного органайзера. Оскільки вона адаптована під

екран дозволом 320x480px і має підтримку сенсерного дисплея.

Як програмна частини буде використано мови веб розробки, такі як HTML, CSS і JavaScript. Виконуватися код буде в браузері Chromium, оскільки веб браузер не вимагає попередньої компіляції вихідного коду, а виконує його під час роботи. Це найоптимальніший вибір, оскільки така модель виконання програмного коду не вимагає установки додаткового програмного забезпечення, і досить використовувати штатний програмне забезпечення яке поставляється операційною системою Raspbian. HTML і CSS дозволить розробити графічний інтерфейс комп'ютерної системи, а JavaScript у спільній роботі з API браузера дозволить обробляти події приходять від користувача системи. API браузера дозволить використовувати такі функції як робота з часом і датою, обробка натискань від користувача системи, робота з запитами до веб сервера також асинхронне виконання JavaScript коду.

Однак потрібно веб сервер, який буде обробляти і відповідати на запити приходять від веб браузера, тому що веб браузер працює тільки з інтернет ресурсами. Веб сервер буде запускатися локально при запуску пристрою і операційних системи Raspbian. Як веб сервера буде використаний NodeJS, поскільки він одночасно є сервером і обробником запитів, до того ж легкий в налаштуванні і не вимагає додаткового програмного забезпечення, крім як інтерпретатора NodeJS встановлений в операційній системі.

2. ОСНОВНА ЧАСТИНА

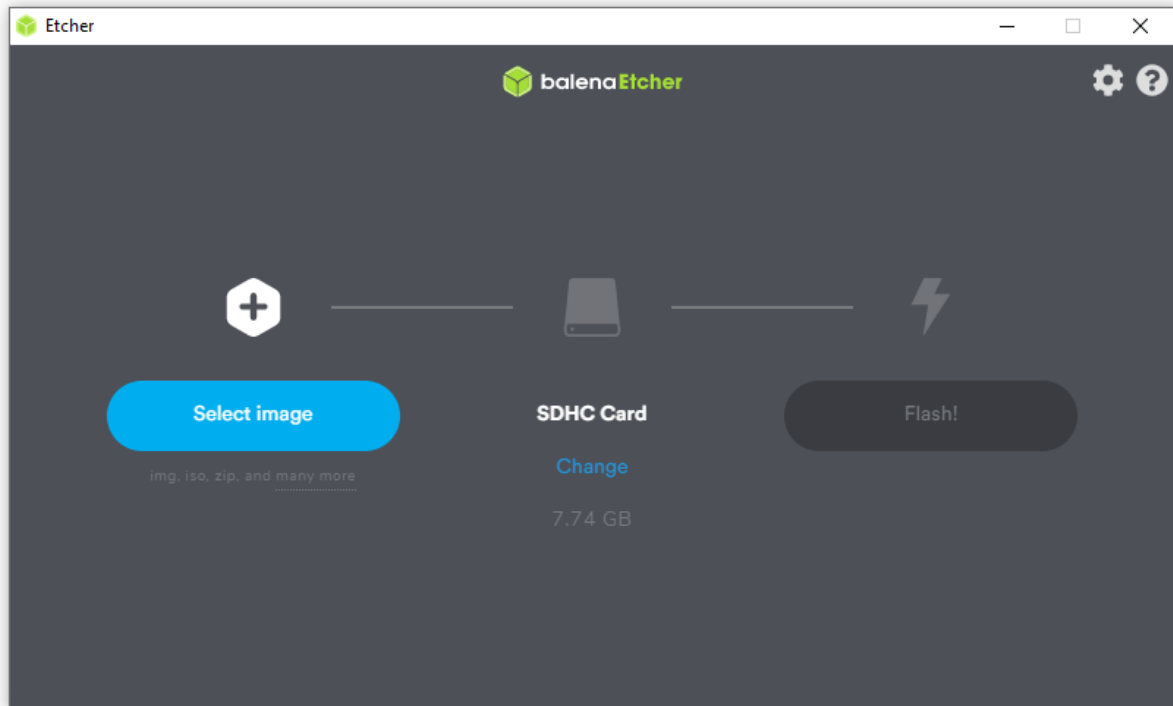
2.1 Підготовка апаратної частини Raspberry Pi та всіх необхідних компонентів

Для повноцінного запуску комп'ютерної системи електронного органайзера попередньо необхідно встановити операційну систему Raspbian встановивши її на карту пам'яті. Raspbian можна скачати на офіційному інтернет ресурсі <https://www.raspberrypi.org/downloads/>, але в цій ОС немає встановленого драйвера для роботи з сенсорним TFT дисплеєм. Розробники такого дисплея надають окрему збірку операційної системи Raspbian з попередньо встановленими і налаштованими драйверами для дисплея та сенсорного введення, яку можна завантажити на інтернет ресурсі http://www.lcdwiki.com/3.5inch_RPi_Display#Download_Resources.

Для встановлення операційної системи знадобиться додаткове програмне забезпечення balenaEtcher. Завантажити ПО можна по інтернет посиланням <https://www.balena.io/etcher/>. ПО підтримує популярні операційні системи, такі як Windows, macOS та багато інших дистрибутивів GNU / Linux.

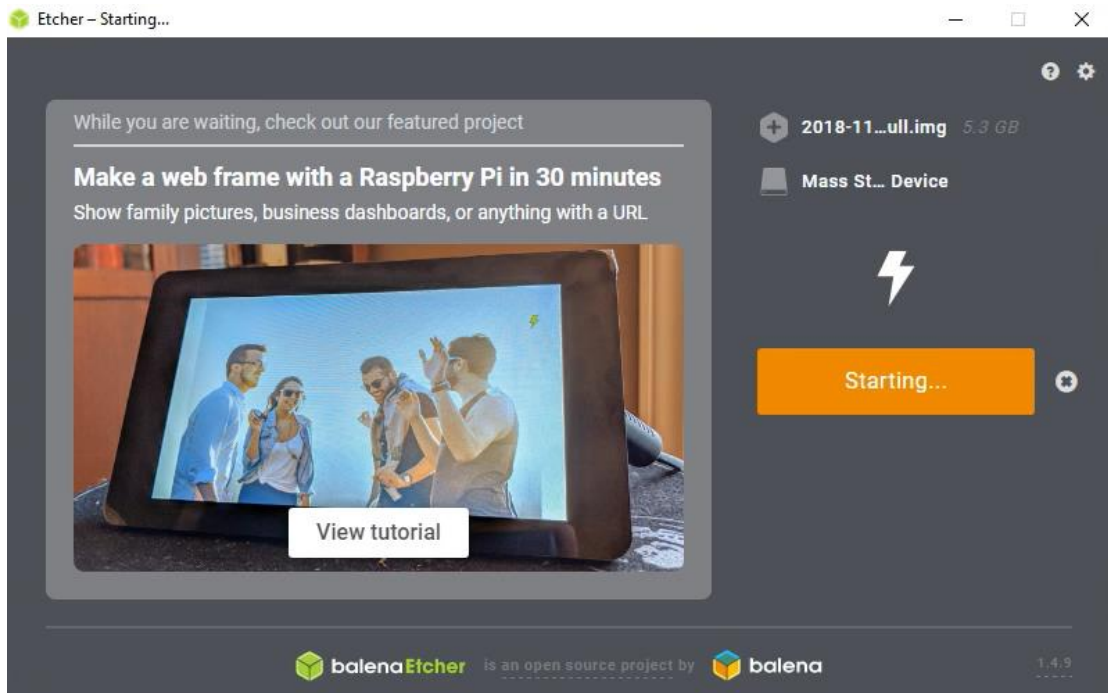
Процес встановлення ОС Raspbian на карту пам'яті:

1. Після завантаження та встановлення balenaEtcher на персональний комп'ютер необхідно підключити карту пам'яті в комп'ютер. Однак багато комп'ютерів не мають можливість для підключення карт пам'яті формату MicroSD, але якщо мають роз'єми для підключення карт формату SD або USB, то необхідно використовувати спеціальні адаптери формату MicroSD-to-SD або MicroSD-to-USB;
2. Після встановлення карти пам'яті в комп'ютер необхідно запустити balenaEtcher. Вікно програми має виглядати як показано на малюнку 13;



Малюнок 13 - Головний екран balenaEtcher

3. Далі необхідно вибрати образ ОС Raspbian натиснувши на кнопку "Select image" у вікні програми. Образ повинен бути заздалегідь завантажений з Інтернет ресурсу; http://www.lcdwiki.com/3.5inch_RPi_Display#Download_Resources.
4. Після вибору потрібного образу необхідно натиснути на кнопку "Flash!" і почнеться процес запису образу на карту пам'яті. Процес запису повинен виглядати як показано на малюнку 14.



Малюнок 14 – Процес завантаження образу ОС Raspbian на карту пам'яті

Після закінчення завантаження образу Raspbian на карту пам'яті її необхідно помістити в слот MicroSD на платі Raspberry Pi. Далі необхідно підключити дисплей в порти GPIO плати Raspberry Pi і встановити захисний корпус. Зібраний пристрій повинен виглядати як показано на малюнках 15 та 16.

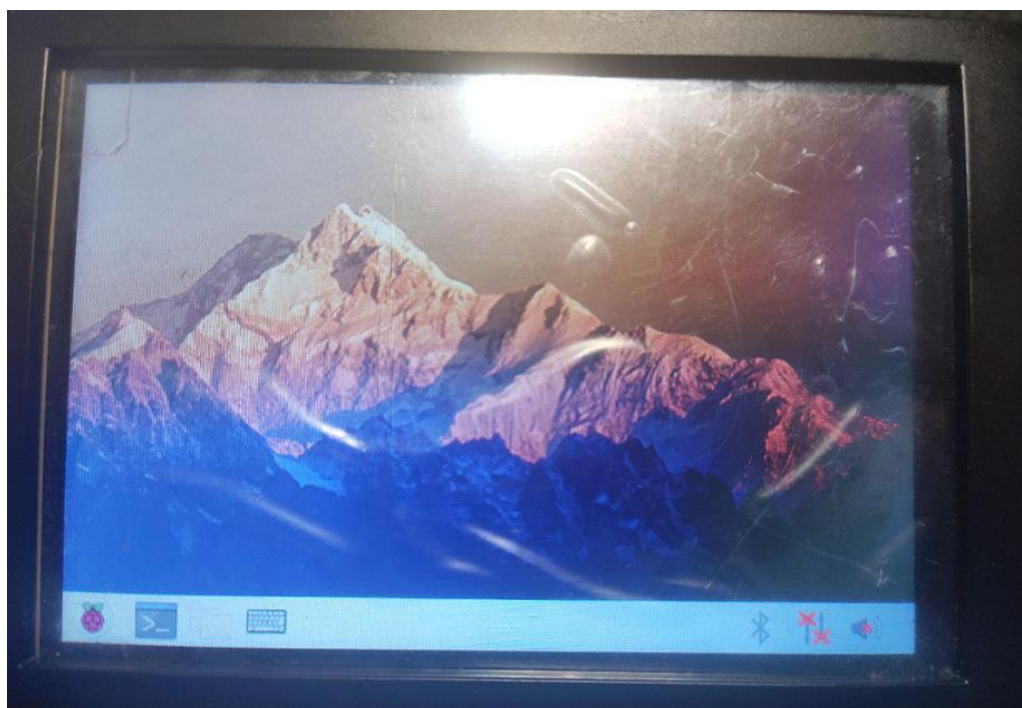


Малюнок 15 - Передня частина зібраного пристрою



Малюнок 16 - Задня частина зібраного пристрою

Після того як дисплей та карта пам'яті була підключена, корпус зібраний можна запускати пристрій підключивши блок живлення до самого пристрою і до електромережі. Через декілька секунд запуситься пристрій і операційна система Raspbian як показано на малюнку 17.



Малюнок 17 - Робочий запуск пристрою

На цьому етапі можна вважати підготовку до запуску пристрою закінченою і можна приступати до підготовки ОС Raspbian до використання.

2.2 Підготовка операційної системи Raspbian для роботи з веб браузером та NodeJS

Для повноцінної роботи комп'ютерної системи необхідно встановити NodeJS. Всі необхідні компоненти будуть завантажені через мережу інтернет. Raspberry Pi та ОС Raspberry підтримує підключення до інтернету через кабель Ethernet або через мережу WiFi. Для зручності налаштування пристрою можна підключити зовнішню клавіатуру і комп'ютерну мишу, або скористатися штатною екранною клавіатурою.

Встановлення NodeJS в систему проводиться командою `curl https://www.nodejs.com/install.sh | sudo sh`. Після цього можна перевірити версію NodeJS командою `node -v`. Версія повинна бути не нижче 12.0.0.

Після встановлення всіх необхідних пакетів необхідно створити папку `organizer` в домашній папці користувача `pi` (абсолютний шлях `/home/pi/organizer`), де будуть знаходитися все вихідні коди електронного органайзера. Після створення папки необхідно створити скрипти, які будуть запускатися під час запуску пристрою і операційної системи. Створені скрипти з назвами `start.sh` та `start-server.sh` необхідно помістити в папку `/home/pi/organizer` і додати їх в автозапуск операційної системи. Для автозапуску створених скриптів скористаємося планувальником завдань `cron` який доступний в багатьох дистрибутивах GNU/Linux та Unix, в тому числі і Raspbian. Командою `crontab -e` переходимо в режим редагування файлу конфігурації завдань і першим рядком необхідно ввести `@reboot /home/pi/organizer/start-server.sh & /home/pi/organizer/start.sh`. Цією командою скрипти будуть запускатися один раз відразу ж після завантаження операційної системи Raspbian.

Зміст скрипта `/home/pi/organizer/server-start.sh`. Він відповідає за запуск сервера NodeJS.

```
#!/bin/sh
cd /home/pi/organizer && node server.js
```

Зміст скрипта `/home/pi/organizer/start.sh`. Він відповідає за запуск вкладки

веб браузеру Chromium в повноекранному режимі, яка введе на локальну адресу сервера NodeJS.

```
#!/bin/sh
```

```
chromium-browser --kiosk --incognito --tab "http://localhost:8080/"
```

На цьому етапі підготовчі роботи автозапуску сервера і веб браузера можна вважати закінченою та можна приступати до програмної частини комп'ютерної системи.

2.3 Розробка програмного забезпечення для роботи системи на стороні веб сервера

Комп'ютерна система електронного органайзера в основі використовує веб браузер Chromium, але всі веб браузери працюють тільки з інтернет ресурсами. Для того щоб інтернет ресурс комп'ютерної системи був доступний локально, без доступу до мережі інтернет, необхідно встановити веб сервер і обробник HTTP запитів. Для створення веб сервера необхідно створити файл package.json в папці /home/pi/organizer, який буде мати в собі всі налаштування NodeJS проекту, з наступним змістом:

```
{  
  "name": "Electronic Organizer",  
  "version": "1.0.0",  
  "description": "",  
  "main": "server.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
    "start": "node server.js"  
  },  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "child_process": "^1.0.2",
```

```
"http": "0.0.1-security",  
"node-static": "^0.7.11"  
}  
}
```

У файлі `/home/pi/organizer/package.json` наведено код у форматі JSON, в якому описані залежності, такі як:

1. `child_process` відповідає за виклик системних команд операційної системи з використанням JavaScript коду;
2. `http` відповідає за роботу веб сервера;
3. `node-static` відповідає за проксінг статичних файлів, таких як `.css` або `.js`, а також інших ресурсів, таких як шрифти та зображення.

Далі необхідно встановити ці залежності одного разу командою `npm install`, для цього потрібне підключення до мережі інтернет. Після цього створиться папка `/home/pi/organizer/node_modules` в якій містяться всі залежності описані у файлі `/home/pi/organizer/package.json`.

Після встановлення всіх необхідних залежностей можна приступати до написання веб сервера і обробника HTTP запитів якід приходять від веб браузера. Необхідно створити файл `server.js` з наступним змістом:

```
var static = require('node-static'); // модуль який відповідає за проксінг  
статичних файлів  
var http = require('http'); // модуль який відповідає за роботу веб сервера  
var exec = require('child_process').exec; // модуль який відповідає за виклик  
системних команд ОС  
  
var file = new (static.Server)();  
  
// Запуск веб сервера і обробка HTTP запитів приходять на порт 8080  
http.createServer(function (req, res) {  
  // Якщо запит містить команду на вимикання пристрою  
  if (req.url === '/api/shutdown') {
```

```

    execute('shutdown -r now', function (callback) {});

    return;
}

// Інакше відповідь на запит буде статичний файл (якщо такий буде
знайдено в файлової системі)
file.serve(req, res);
}).listen(8080);

// Функція яка викликає системну команду ОС
function execute(command, callback) {
    exec(command, function (error, stdout, stderr) {
        callback(stdout);
    });
}

```

Після написання програмної частини веб сервера та обробника HTTP запитів можна запустити веб сервер командою `node server.js` в директорії `/home/pi/organizer`. Після цього стане доступний інтернет ресурс `http://localhost:8080/`, де `localhost` це адреса локальної машини, а `8080` це що прослуховується порт веб сервером.

2.4 Розробка програмного забезпечення для роботи системи в веб браузері

Комп'ютерна система електронного органайзера ділиться на декілька частин, таких як: відображення дати і часу, таймер, секундомір та будильник. Також буде потрібно функціонал установки дати і часу, на основі яких працює вся система. Повний вихідний код наведено у Додатку А.

Для розробки призначеного для користувача інтерфейсу буде використаний мову розмітки HTML в зв'язці з мовою опису стилів CSS (каскадні

таблиці стилів), а також деякі додаткові ресурси, такі як іконки UI елементів в форматі SVG і фонове зображення у форматі JPEG. Готовий варіант користувальницького інтерфейсу, а також додаткових вікон управління наведені на малюнках 18, 19, 20, 21, 22, 23, 24, 25.

На екранах показаних на малюнках 18, 19, 21, 22 присутні елементи управління, такі як перемикання активного екрану (екран дати часу, таймера, секундоміра і будильника) які знаходяться в центрі верхньої частини екрану, а також виклик налаштувань дати і часу знаходиться в правому верхньому кутку екрану (малюнок 18, 19, 21, 22) та кнопка для виключення пристрою знаходиться в лівому верхньому кутку екрану. Однак перед вимкненням телефону буде виведено вікно з підтвердженням вибору як показано на малюнку 25, адже користувач може випадково натиснути на цю кнопку.

Вся логіка пов'язана з обробкою подій користувача, а так само роботу всіх основних функцій написана на мові програмування JavaScript з використанням API веб браузеру Chromium.

Екран з відображенням поточної дати і часу є головним екраном як показано на малюнку 18. На ньому динамічно змінюється поточний час та дата у реальному часі. За замовчуванням дати і час має значення 01.01.2001 00:00:00 (початок третього тисячоліття), це значення можна змінити встановивши дату та час у вікні завдання налаштувань (малюнок 24).

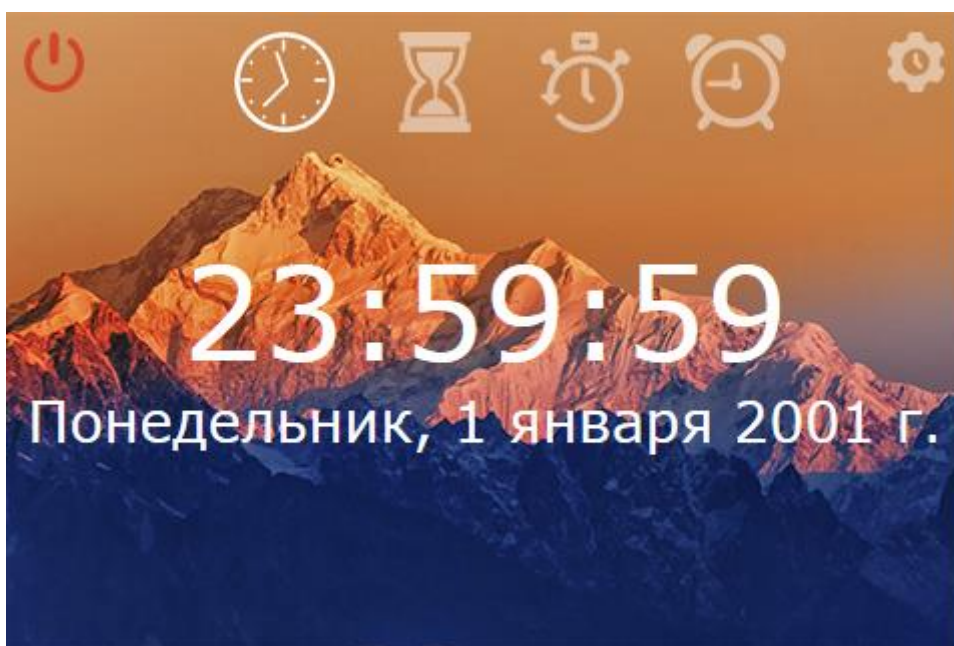
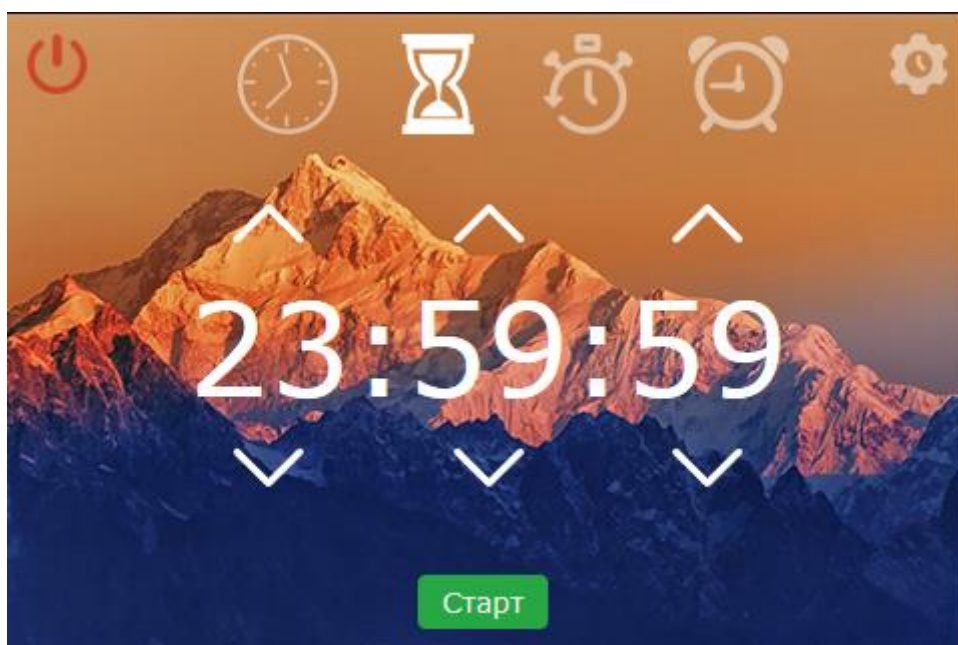
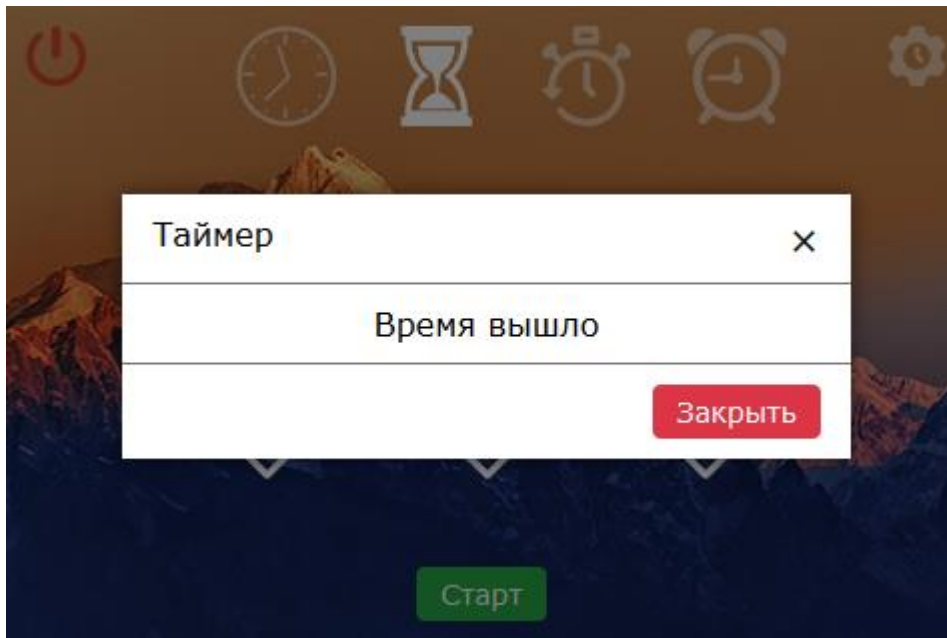


Рисунок 18 – Головний екран

Екран таймера зображений на малюнку 19. За допомогою елементів управління «стрілок» задається необхідний час таймера і після натискання на кнопку «Старт» почнеться посекундна зменшення цього часу. Після запуску таймера пропадають «стрілки» завдання часу і з'являються додаткова кнопка «Пауза» натискання на якій таймер зупиняється і пропонується вибір, або очистити таймер або продовжити. По закінченню таймера до відмітки часу в 00:00:00 користувачеві виводяться вікно з повідомленням «Время вышло», як показано на малюнку 20, незалежно від того який екран був активний.

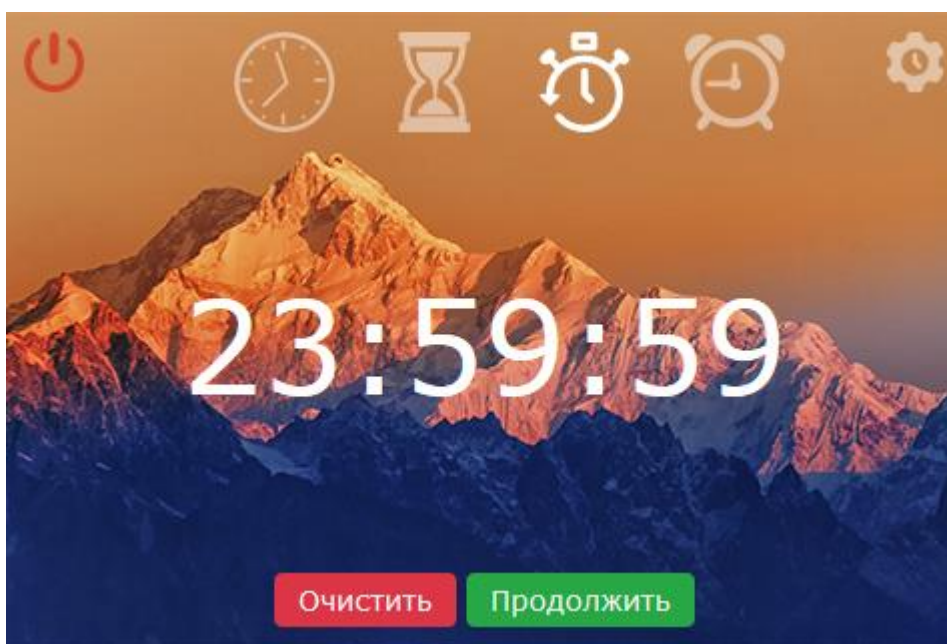


Малюнок 19 - Екран таймера



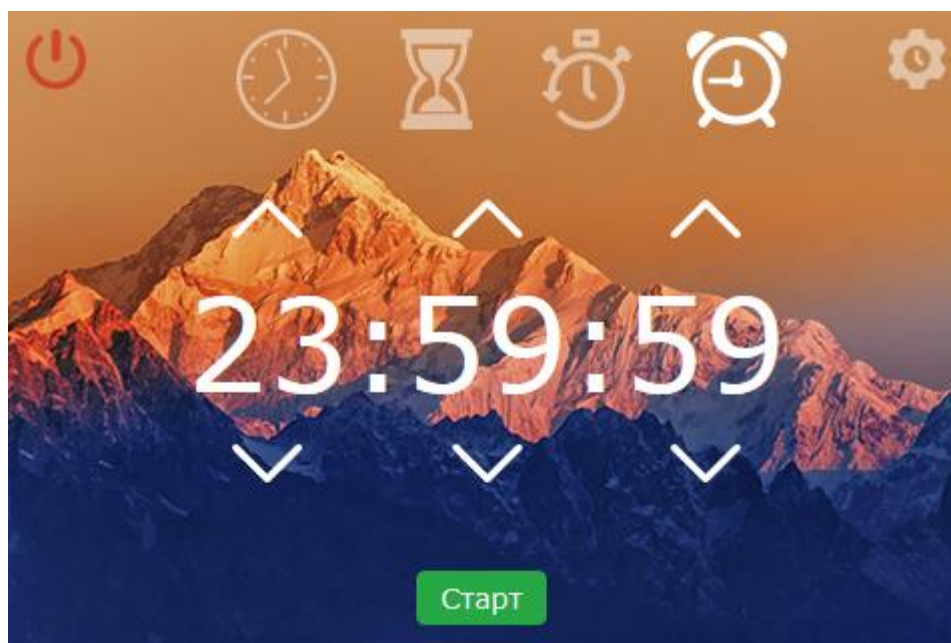
Малюнок 20 - Вікно з повідомленням таймера

Екран секундоміра зображений на малюнку 21. Після натискання на кнопку «Старт» секундомір почне свою роботу збільшуючи час секундоміра. Після запуску секундоміра кнопка «Старт» пропадає і замість неї з'являється кнопка «Пауза» при натисканні на якій секундомір не продовжує збільшувати час і пропонується Очистити або Продовжити роботу секундоміра. При досягненні позначки в 86400 секунд показники секундоміра обнуляються на час 00:00:00.

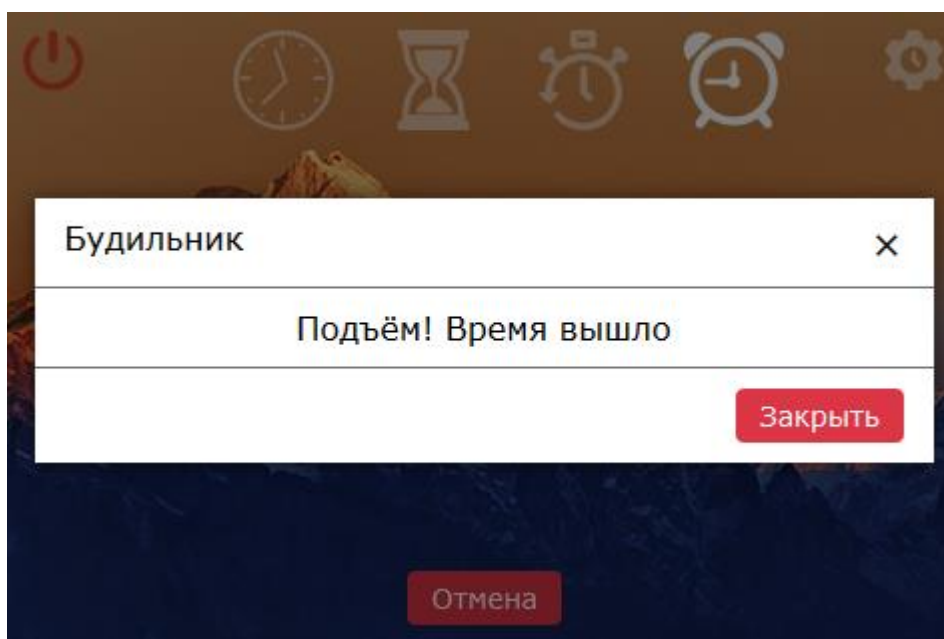


Малюнок 21 - Екран секундоміра

Екран будильника зображений на малюнку 22. За допомогою елементів управління «стрілок» задається бажаний час будильника і при натисканні на кнопку «Старт» будильник встановлюється в активний стан. Якщо ж загального системного часу досягне тієї позначки часу яка встановлена в будильнику, то буде виведено вікно з повідомленням «Подъём! Время вышло», як показано на малюнку 23, незалежно від того який екран був активний.

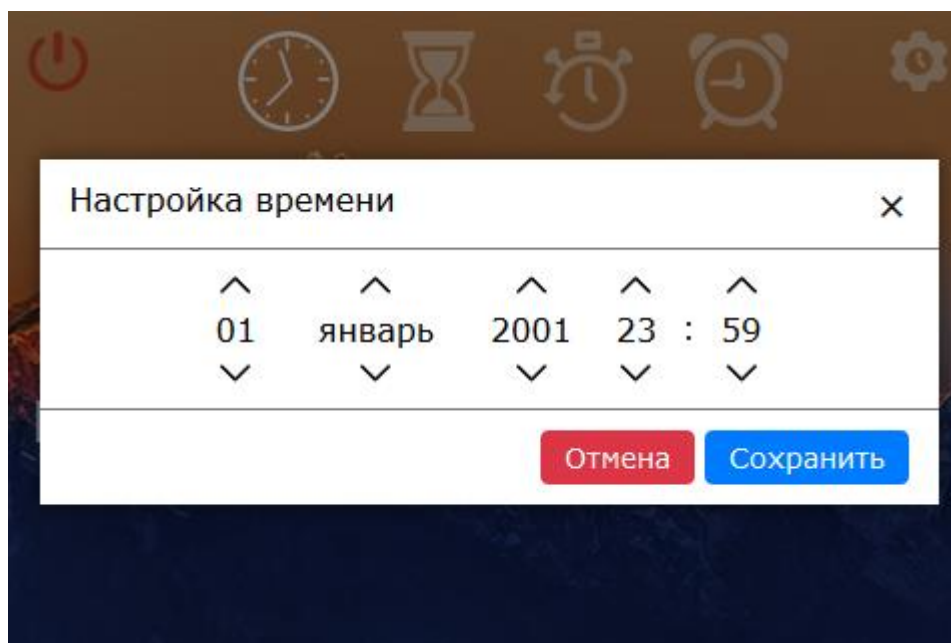


Малюнок 22 - Екран будильника



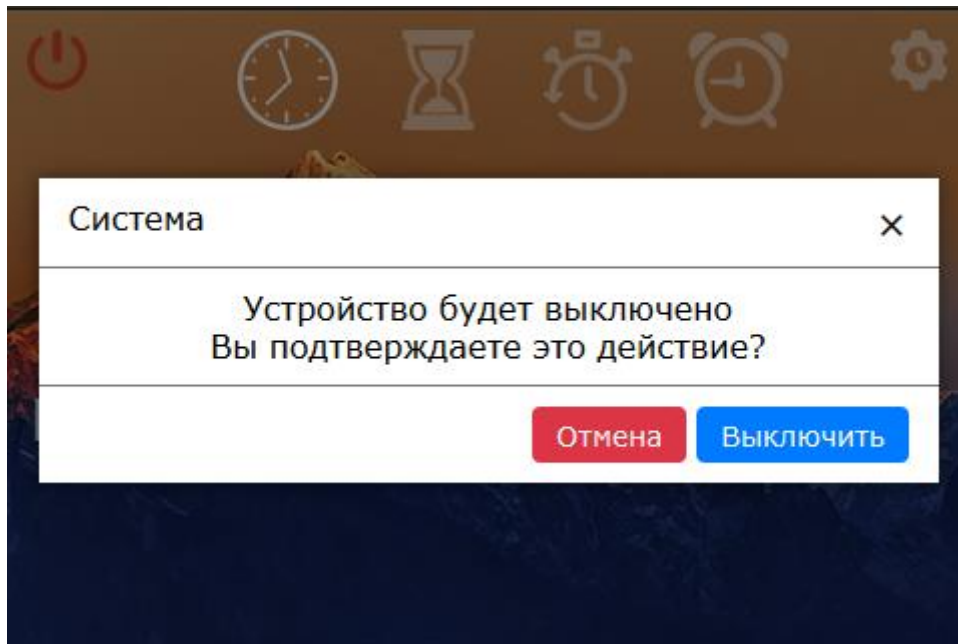
Малюнок 23 - Вікно з повідомленням будильника

Діалогове вікно настройки дати та часу зображено на малюнку 24. За допомогою елементів управління «стрілок» задається поточна дата і час і при натисканні на кнопку «Сохранить» значення зберігається в оперативну пам'ять. Дане значення впливає на екрани Дати і часу (рисунок 18) і функцію Будильника (рисунок 22). Це значення обнуляється на значення за замовчуванням 01.01.2001 00:00:00 при перезапуску пристрою.



Малюнок 24 - Вікно установки дати і часу

Діалогове вікно підтвердження вимикання пристрою зображено на малюнку 25. При натисканні на кнопку «Выключить» операційна система Raspbian коректно вимикається через кілька секунд. Розробники плат Raspberry Pi та ОС Raspbian не рекомендують вимикати свої плати методом різкого відключення живлення, так як це може пошкодити компоненти плати тим самим порушивши роботу всієї комп'ютерної системи.



Малюнок 25 - Вікно з підтвердженням вимкнення пристрою

ВИСНОВОК

У дипломній роботі розглянуті варіанти побудови комп'ютерних систем на базі мікроконтролерів і мікрокомп'ютерів, застосування веб технологій з використанням IoT технологій.

Було проведено аналітичний огляд вибору між мікроконтролерів і мікрокомп'ютерів. Вибір припав на мікрокомп'ютери, тому що вони більш продуктивні, підтримують повноцінні операційні системи, такі як різні дистрибутиви GNU/Linux, а також Windows. Також вибір на користь мікрокомп'ютерів дозволив абстрагуватися від апаратної і програмної частини комп'ютерної системи. Програмний код легко можна буде перенести і адаптувати під будь-який інший мікрокомп'ютер або комп'ютер, якщо він відповідає вимогам комп'ютерної системи.

Вибір на користь веб технологій допоміг максимально абстрагуватися від використовуваної платформи виконання коду і здатний працювати скрізь, де тільки можна запустити сучасний браузер. JavaScript з використанням веб браузера дозволив організувати легке паралельне виконання коду за рахунок використання асинхронної моделі мови JavaScript в зв'язці з API веб браузера, а також реагування на події користувача комп'ютерної системи. Мови HTML і CSS дозволили з легкістю реалізувати користувальницький інтерфейс.

Таким чином вибір на користь вибору мікрокомп'ютерів і веб технологій дозволив реалізувати комп'ютерну систему, яку з легкістю можна запустити на багатьох мікрокомп'ютерів або комп'ютерів не змінюючи програмний код системи, також немає необхідності в компіляції.