

Ім'я користувача:
Volodymyr Donchenko

Дата перевірки:
04.01.2024 14:37:29 EET

Дата звіту:
04.01.2024 14:39:44 EET

ID перевірки:
1016045142

Тип перевірки:
Doc vs Internet

ID користувача:
100012947

Назва документа: Моргун_маг – копія (1)

Кількість сторінок: 52 Кількість слів: 6240 Кількість символів: 50370 Розмір файлу: 12.08 MB ID файлу: 1015743036

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

5.51%
Схожість

Найбільша схожість: 0.75% з Інтернет-джерелом (<https://studfile.net/preview/5010347>)

5.51% Джерела з Інтернету

42

Сторінка 54

Пошук збігів з Бібліотекою не проводився

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0.02%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 9 слів та 0%)

0.02% Вилучення з Інтернету

83

Сторінка 55

Немає вилучених бібліотечних джерел

Модифікації

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

Замінені символи

11

Підозріле форматування

8
сторінок

Міністерство освіти і науки України
Державний заклад
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

Моргун Данило Вікторович

**ПРОЄКТУВАННЯ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ
КОНФІГУРАЦІЇ РОЗДАВАЛЬНИКІВ ЛІКІВ**

**кваліфікаційна робота
здобувача вищої освіти другого (магістерського) рівня
освітньої програми «Комп'ютерні мережі»
за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис _____ Данило МОРГУН

Науковий керівник _____ Володимир ДОНЧЕНКО,
старший викладач
кафедри інформаційних технологій
та систем

В.о. завідувача кафедри _____ Микола СЕМЕНОВ,
кандидат педагогічних наук, доцент
кафедри інформаційних технологій
та систем

Полтава – 2024

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ПЕРЕДУМОВИ ДЛЯ РОЗРОБКИ.....	5
1.1. Класифікація пігулок.....	5
1.2. Конструкції роздавальників ліків та пристроїв «розумних» контейнерів.....	9
1.3. Інтерфейси різних систем.....	13
1.4. Висновки до розділу.....	20
РОЗДІЛ 2. КОНСТРУКЦІЯ РАЗДАВАЛЬНИКІВ ЛІКІВ.....	21
2.1. Кінематична схема роздавальника ліків.....	21
2.2. Дизайн роздавальника ліків.....	22
2.3. Принципова схема роздавальника ліків.....	23
2.4. Синтез структурно-функціональної схеми пристрою.....	24
2.5. Основні характеристики і можливості ATmega16.....	29
2.6. Принципова схема пристрою.....	31
2.7. Вибір елементної бази.....	32
2.8. Розробка програми.....	33
2.9. Розробка друкованої плати.....	37
2.10. Висновки до розділу.....	40
РОЗДІЛ 3 РОЗРОБКА ДОДАТКА ДЛЯ ПРИСТРОЇВ НА ПЛАТФОРМІ ANDROID.....	41
3.1. Вибір інструментарію розробки.....	41
3.2. Висновки до розділу.....	48
ВИСНОВКИ.....	49

ВСТУП

У 1839-му використовувати желатинові капсули запропонував Мозес, і вже до 1841 року їх виробництво було порівняно непогано поставлено у Франції. Значним досягненням в технології лікарських форм стало винахід Вільямом Брокдоном в 1843 році пігулок.

У 1847 році в Німеччині желатинові капсули були розроблені фармацевтом Сімоненом, який готував їх за допомогою воскових форм. Уже в 1853 році фармацевт Штейнбрехер для цієї мети почав застосовувати металеві "шпильки".

Перші розчинні пігулки з'явилися в 1931 році. Незадовго до цього, в розпал епідемії грипу та застуд в січні 1928 року, глава американської фармацевтичної компанії Dr. Miles Laboratorie Ендрю Бердслей, зайшовши до редакції місцевої газети, з подивом виявив, що весь штат співробітників на робочих місцях. Редактор газети охоче поділився з ним секретом винайденого ним еліксиру з аспірину, соди і лимонного соку. Саме ці компоненти увійшли до складу розчинних пігулок, які позиціонують як засіб від застуди, грипу, головного болю і шлункових нездужань. Популярності пігулок сприяв і виявлений незабаром "побічний ефект" - зняття дискомфорту після прийому алкогольних напоїв. Саме він згодом став основним доведенням дії препарату. За минулі 80 з невеликим років він став міжнародним символом боротьби з симптомами похмілля і загальновідомим методом швидко прийти в себе вранці після вдалої вечірки.

Крім того, водорозчинні лікарські форми дають можливість доставляти в організм великі дози препаратів, що дозволяє виробляти засоби пролонгованої дії. Так, муколітик ацетилцистеїн зручний тим, що дає можливість прийняти ліки один раз на добу, полегшити симптоми хвороби і займатися справами. Те ж можна сказати і про численні "шипучі" вітаміни, що містять необхідну добову дозу в одній пігулці.

Сьогодні відомо безліч лікарських форм, і список цей постійно зростає. Розширюється асортимент препаратів в нових лікарських формах та упаковках:

шарові пігулки та драже, різні капсули, спеціальні форми для дітей (від кашлю гранули для приготування сиропу), мазі в тубах, спреї та аерозолі в балончиках, упаковках з полімерних та інших матеріалів і т. п.

З величезною кількістю лікарських препаратів оформлених у вигляді пігулок існує проблема організації прийому даних препаратів пацієнтами, особливо в похилому віці.

З метою вирішення цієї проблеми багато фірм випускають пристрої призначені для організації прийому препаратів. Але при проектуванні даних пристроїв виникає інша проблема, це вибір людино-машинного інтерфейсу для конкретного пацієнта.

Об'єкт дослідження - роздавальники ліків.

Предмет дослідження - інтерфейси роздавальників ліків.

Мета роботи – розробка підходів до проектування людино-машинного інтерфейсу конфігурації роздавальників ліків.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- провести аналіз теоретичних основ для розробки пристрою роздавальників ліків та класифікацію інтерфейсу між людиною та машиною;
- провести розробку роздавальника ліків оснащеного кнопочним виконавчим механізмом інтерфейсу;
- виходячи з аналізу виконаного інтерфейсу розробити універсальний інтерфейс на базі ANDROID приладу та BlueTooth.

Методи дослідження - теоретико-експериментальна розробка людино-машинного інтерфейсу.

РОЗДІЛ 1.**ТЕОРЕТИЧНІ ПЕРЕДУМОВИ ДЛЯ РОЗРОБКИ****1.1. Класифікація пігулок**

За способом отримання розрізняють два класи пігулок [1]:

1. Пресовані, одержувані шляхом пресування лікарських порошків на пігулкових машинах різної продуктивності. Даний спосіб є основним.
2. Формовані або тритураційні пігулки, одержувані формуванням пігульчастої маси. Вони складають приблизно 1-2% від усього обсягу виробництва пігулок. Тритураційні пігулки містять невеликі дози лікарських і розбавляючих речовин: маса їх може становити до 0,05 м.

Пігулки класифікують також за конструктивною ознакою:

1. За складом: прості (однокомпонентні) та складні (багатокомпонентні).
2. За структурною будовою: каркасні, одношарові і багатошарові (з не менше ніж 2 шарами), з покриттям або без нього.

Каркасні (або скелетні) пігулки (дурули) мають нерозчинний каркас, порожнечі якого заповнені лікарськими речовинами. Кожна окрема пігулка може бути порівняна з губкою, що просочена ліками. Під час прийому каркас не розчиняється, зберігаючи свою геометричну форму, а лікарська речовина поступово дифундує в шлунково-кишковий тракт.

Одношарові пігулки складаються з пресованої суміші лікарських і допоміжних речовин і однорідні по всьому об'єму лікарської форми.

У багатошарових пігулках лікарські речовини розташовуються пошарово. При використанні багатошарових пігулок з хімічно несумісними речовинами це призводить до мінімізації їх взаємодії.

За характером покриття: дражированне, плівкове і пресоване сухе покриття.

Вироблені хіміко-фармацевтичною промисловістю форми пігулок є дуже різноманітними, включаючи циліндри, кулі, куби, трикутники, чотирикутники

та інші. Найпоширенішими є плоско-циліндрична форма з фаскою та двоопукла форма, яка є зручною для ковтання. Крім того, пуанسونи і матриці, використовувані у виробництві пігулок, є досить простими і не викликають особливих труднощів при їх установці на таблеткові машини.

Більшість наявних упаковочних автоматів також придатні для роботи з плоско-циліндричними і двоопуклими пігулками.

Плоско циліндрична без фаски форма пігулок для виробництва не рекомендується, так як при розфасовці і транспортуванні спостерігається руйнування гострих країв пігулок, в результаті чого втрачається їх товарний вигляд.

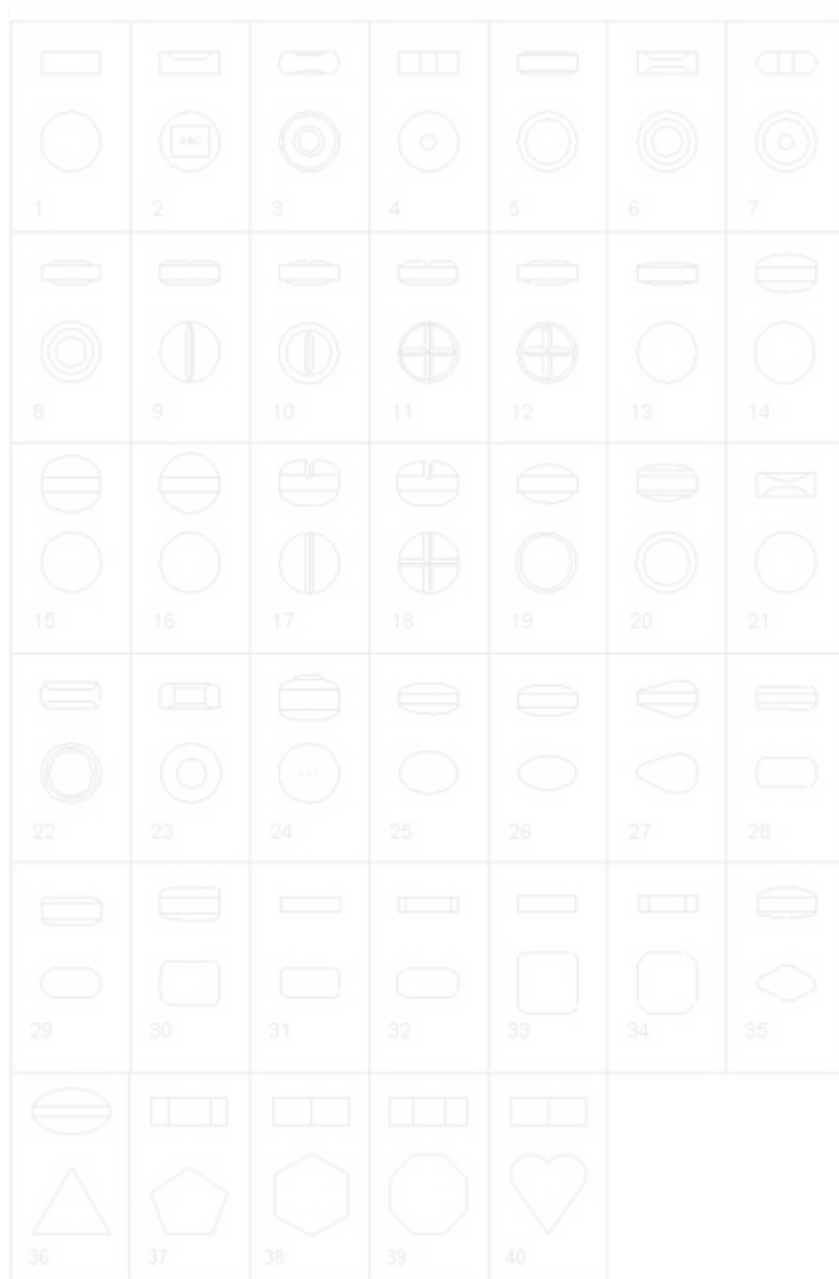
Розмір пігулок коливається від 4 до 25 мм в діаметрі. Пігулки діаметром понад 25 мм називаються брикетами. Найбільш поширеними є пігулки діаметром від 4 до 12 мм. Таблетки з діаметром понад 9 мм можуть мати одну або дві ризи, які розташовані перпендикулярно одна одній. Це дозволяє розділити таблетку на дві або чотири частини і, таким чином, регулювати дозу лікарської речовини.

Вага пігулок, в основному, коливається від 0,05 до 0,8 г і залежить від дозування лікарської речовини та кількості допоміжних речовин, що входять до їх складу.

Важливо, щоб таблетки мали правильну форму, були цілими, без подряпин на краях, і мали гладку й однорідну поверхню. Пігулки повинні мати достатню міцність і не повинні кришитися. Геометрична форма і розміри пігулок визначаються стандартом - ОСТ 64-072-89 «Засоби лікарські. Пігулки. Типи і розміри». Він передбачає, в основному, випуск двох типів пігулок: плоско циліндричні без фаски і з фаскою, двоопуклих без покриття та з покриттям: плівковим, напресованим і дражированим. За кордоном є більш широкий вибір форм пігулок (табл. 1.1).

Таблиця 1.1

Типорозмірний ряд пігулок, вироблених за кордоном



1. плоско циліндрична, проста;
2. плоско циліндрична з поглибленої панеллю;

8

3. плоско циліндрична з поглибленими центрами;
4. плоско циліндрична з вирізаним центром;
5. плоско циліндрична з фаскою;
6. плоско циліндрична з фаскою і поглибленими центрами;
7. плоско циліндрична з фаскою і вирізаним центром;
8. плоско циліндрична з посиленою фаскою;
9. плоско циліндрична з фаскою і однієї рисою;
10. плоско циліндрична з посиленою фаскою і однієї рисою;
11. плоско циліндрична з фаскою і двома ризиками;
12. плоско циліндрична з посиленою фаскою і двома ризиками;
13. плоско циліндрична з дрібної сферою;
14. плоско циліндрична з нормальною сферою;
15. плоско циліндрична з глибокої сферою;
16. плоско циліндрична куляста;
17. круга з нормальною сферою і однієї рисою типу «А»;
18. круга з нормальною сферою і двома ризиками типу «А»;
19. дражеподібна, проста;
20. круга з фаскою і сферою;
21. круга з поглибленими центрами;
22. круга плоска з обідком;
23. круга з обідком і вирізаним центром;
24. круга з нормальною сферою і написом;
25. сферична еліпсоїдна;
26. сферична овальна;
27. сферична мигдалевидна;
28. сферична капсулевидна;
29. сферична капсулевидна з товарним знаком;
30. сферична кулевидна;
31. плоска прямокутна із заокругленими кутами;
32. плоска прямокутна з ромбовидними кутами;

- 33.плоска квадратна із закругленими кутами;
- 34.плоска квадратна з ромбовидними кутами;
- 35.сферична ромбовидна;
- 36.сферична трикутна;
- 37.плоска п'ятикутна;
- 38.плоска шестикутна;
- 39.плоска восьмикутна;
- 40.плоска серцевидна.

1.2. Конструкції роздавальників ліків та пристроїв «розумних» контейнерів

Розглянемо універсальний пристрій у вигляді кришки, який складається з корпусу з технологічними отворами для розміщення кнопок, екрану, світлодіода і щілини для поширення звукових хвиль. Усередині корпусу розташована двостороння електронна плата зі схемою. Цей пристрій дозволяє користувачеві встановлювати кількість і періоди прийомів, а також відображає інформацію про проведені та прийоми що залишилися на екрані. Щоб допомогти користувачеві орієнтуватися в часі приймання препаратів, в пристрої вбудований годинник.

Кришка стає універсальною завдяки набору знімних різьбових вкладишів, які фіксуються в корпусі ззаду плати і призначені для різних типів ємностей. Система світлової та звукової сигналізації часу прийому активується за вказаним користувачем часом і керується відкриванням горловини флакона або іншої ємності, підтверджуючи тим самим факт прийому препарату [2].

Суть винаходу роз'яснюється через креслення, які зображено на рисунку 1.1. На цих малюнках відображено збірку пристрою з розташованими компонентами. На передній частині корпусу кришки (означеної як 1) розміщені: електронна плата (2) із вбудованим мікроконтролером (3), дисплей (4), кнопки переміщення по цифровому ряду (5), кнопка налаштування часу і дати (6), кнопка для установки прийомів (7), а також система сигналізації у вигляді

одного двобарвного світлодіода (8) і динаміка (9). Під номером 16 позначено знімні вкладиші (умовно показані два вкладиші).

На рисунку 1.2. представлено збірку пристрою із розміщеними компонентами, включаючи донну частину корпусу кришки 1, задню частину електронної плати 2, на якій розташовані елементи оптичної пари, що включає джерело 10 та приймач 11 світлового випромінювання, мікросхему годинника реального часу 12, елемент живлення для годин реального часу CR 13, джерело живлення всього пристрою 14 і кнопку включення 15. Знімні вкладиші позначені під номером 16 на рисунку (представлено умовно, зображено два вкладиші).

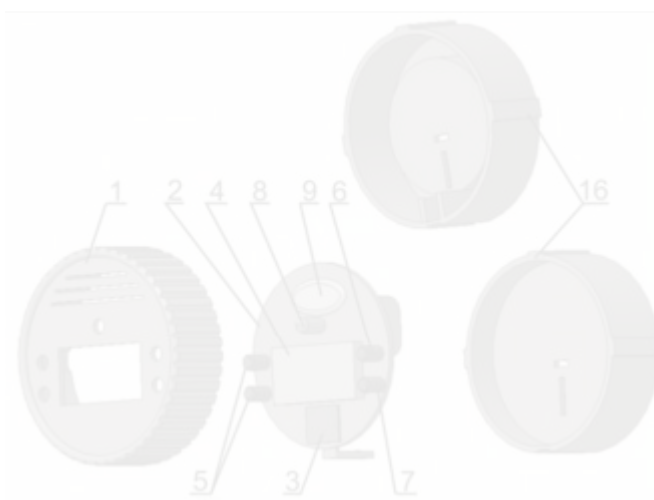


Рис. 1.1. Зборка пристрою з рознесеними компонентами

Робота пристрою виглядає наступним чином: користувач вмикає пристрій за допомогою кнопки 15 і закриває посудину з препаратом. У цей час на дисплеї 4 відображається поточний час і дата. Налаштування часу і дати виконуються за допомогою кнопок для налаштування часу і дати 6, а також кнопок переміщення по цифровому ряду 5. Подальше визначення кількості та часу проведення прийомів відбувається шляхом натискання кнопки налаштування прийомів 7 та кнопок переміщення по цифровому ряду 5 користувачем. Після введення часу останнього прийому, натисканням кнопки для установки 7

користувач отримує на екрані 4 інформацію про поточний час і дату та кількість прийомів, що залишилися.

Коли настане час для першого прийому, активується сигналізація у вигляді миготіння світлодіода 8 і звуку з динаміка 9. Після цього користувач повинен відкрити посудину з препаратом, щоб вимкнути сигналізацію, і тоді інформація про проведений прийом відображатиметься на екрані 4. У випадку, якщо користувач не відкриє кришку протягом сигналізації, через хвилину сигналізація автоматично вимикається, і світлодіод 8 починає миготіти жовтим світлом, позначаючи пропущений прийом. Якщо користувач вирішить передчасно взяти препарат, відкриття кришки також викличе сигнал про прийом препарату і перейде до очікування часу для наступного прийому.

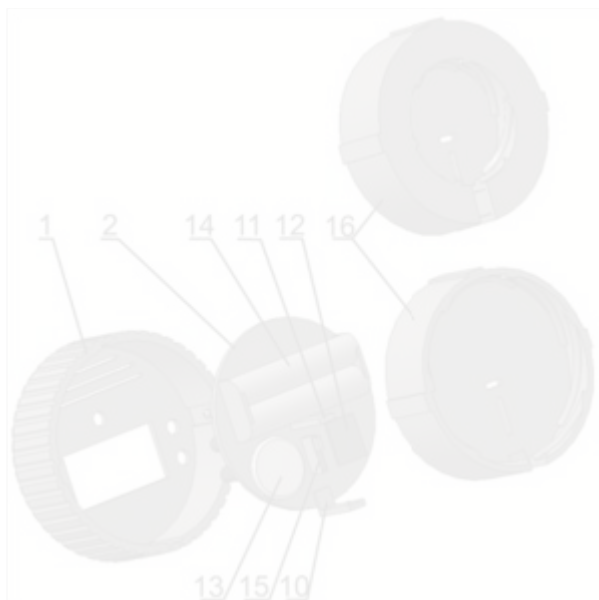


Рис.1.2. Збірка пристрою із розташованими окремо компонентами, на якій відображена нижня частина корпусу кришки 1.

GlowCap - це спеціальна кришка, яка підходить до більшості контейнерів для ліків і починає миготіти помаранчевим світлом, коли приходить час взяти ліки. Якщо це відкладається, через годину починає відтворюватися мелодія. Кришка постачається з каганцем, що бездротово з'єднується з нею і також

починає миготіти в потрібний час. Внутрішній чіп кришки визначає, коли контейнер відкритий, і висилає відповідне повідомлення через бездротову мережу в додаток, що дозволяє лікарю (або родичам) відстежувати точність виконання режиму прийому ліків [3].



Рис. 1.3. GlowCap

Система розподілу (дозатор) пігулок MedReady 1700 (рис 1.4.) схожа на звичайну семиденну коробочку для ліків, яку іноді можна придбати і в нашій країні. Це простий пристрій, який гарантує, що користувач отримає правильну дозу ліків в потрібний час. Дозатор містить 28 осередків для тижневого набору ліків і може містити десятки різних видів пігулок.

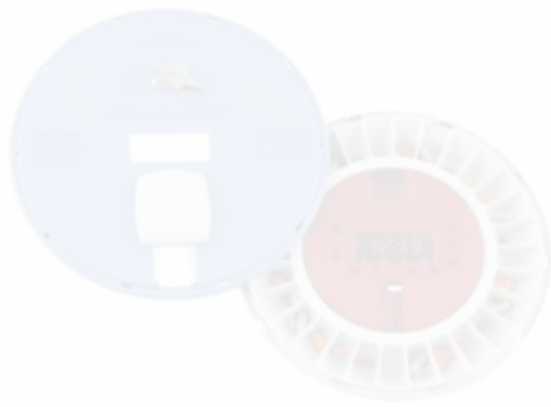


Рис. 1.4. MedReady 1700

Коли підходить час приймати ліки, лоток всередині пристрою автоматично повертається до потрібного осередку, і пролунає звуковий і світловий сигнал. Потрібні пігулки виявляються під невеликою кришкою у верхній частині дозатора, і щоб їх взяти досить зрушити кришку. Якщо, незважаючи на сигнали, ліки не використовуються, пристрій через бездротову мережу негайно надсилає відповідне повідомлення лікарю або особі, яка доглядає за пацієнтом. Для цього MedReady забезпечена модемом мобільного зв'язку, що дозволяє зв'язуватися з системою на центральному комп'ютері, що містить інформацію про правила прийому пігулок пацієнтом та їх виконання. Ця інформація міститься на захищеному порталі і доступна для використання лікарем або, наприклад, родичем.

1.3. Інтерфейси різних систем

Поняття «інтерфейс» відображає форми, засоби та можливості забезпечення взаємодії двох або більше систем (їх компонентів) між собою незалежно від їх фізичної або ментальної природи в процесі досягнення їхніх цілей.

Людино-машинний інтерфейс - це засоби та методи, що забезпечують пряму взаємодію між оператором та технічною системою, дозволяючи оператору керувати системою та контролювати її роботу. Зазвичай цей термін використовується для опису взаємодії між оператором та програмним забезпеченням ЕОМ, яким він користується.

Управління складною технікою висуває вимоги до добре продуманого призначеного для користувача інтерфейсу, що забезпечує ефективну і безпечну експлуатацію об'єкта управління.

Класифікуючи ЛМІ, слід розрізняти способи уявлення та принципи відбору інформації (Рис. 1.5).



Рис.1.5. Класифікація інтерфейсів оператора

За способом подання інформації можна виділити наступні типи:

1. Приладовий інтерфейс (рис. 1.6) – інформація представляється у вигляді аналогових і цифрових приладів, індикаторних лампочок, табло тощо. Як органи управління використовуються ключі, важелі і кнопки. Позитивною стороною є те, що всі прилади одночасно доступні оператору, вони містять динамічні елементи і дають наочне уявлення про систему.



Рис. 1.6. Приладовий інтерфейс

2. Схематичний інтерфейс (рис. 1.7). Характерною рисою даного інтерфейсу є висока ступінь концентрації елементів управління і візуалізації. Стан обладнання відображається або кольором, або спеціальними символами.

Сильною стороною такого інтерфейсу є те, що вся інформація по системі зібрана в одному місці, що полегшує її візуальний пошук.



Рис. 1.7. Схематичний інтерфейс

3. Екологічний інтерфейс (рис. 1.8) складається не тільки з окремих елементів, але і з образів вищого порядку, що відбивають, наприклад, фізичні закономірності середовища.



Рис. 1.8. Екологічний інтерфейс

Візуалізація процесів є найбільш популярним додатком ідеї екологічного інтерфейсу. Мета такої візуалізації - прискорити зіставлення величин.

4. Функціональний інтерфейс (рис. 1.9) заснований на функціонально-орієнтованій філософії управління. Даний підхід дозволяє переключити увагу оператора з контролю над станом різних підсистем на контроль над виконанням загальних для цих підсистем функцій. Сильною стороною функціонального

інтерфейсу є можливість швидко оцінити стан технологічного процесу і достатність задіяного обладнання.



Рис. 1.9. Фрагмент відеокдру з функціональним інтерфейсом

5. Імерсивний інтерфейс (рис. 1.10) - максимально правдоподібне відтворення реального світу, зване також віртуальної реальністю. Мнемосхема, що відтворює умовний образ об'єкта, також є елементом імерсійного середовища.



Рис. 1.10. Імерсивний інтерфейс

Для комп'ютеризованого інтерфейсу важливою характеристикою є принцип відбору інформації для наповнення одного відеокадру. З відбору інформації існують три основних подання:

- оглядові інтерфейси вирішують задачу швидкої оцінки стану системи. Інформація, що відображається проходить ранжування і фільтрацію;
- задачні інтерфейси містять інформацію, релевантну поточному завданню;
- системні інтерфейси призначені для представлення інформації за обраною ізольованою системою.

За підсумками розгляду різних типів інтерфейсів було прийнято рішення розробляти комбінований інтерфейс, який містить в собі елементи функціонального та імпресивного інтерфейсів. За принципом відбору інформації інтерфейс є системним.

Етапи проєктування людино-машинного інтерфейсу можна розглядати наступним чином:

1. Проєктування:

- **функціональні вимоги:** визначення мети розробки та вихідних вимог.
- **аналіз користувачів:** визначення потреб користувачів, розробка сценаріїв, оцінка відповідності сценаріїв очікуванням користувачів.
- **концептуальне проєктування:** моделювання процесу, для якого розробляється програма.
- **логічне проєктування:** визначення інформаційних потоків в додатку.
- **фізичне проєктування:** вибір платформи, на якій буде реалізований проєкт, і оцінка витрат на розробку.

2. Реалізація:

- **прототипування:** розробка паперових та/або інтерактивних макетів екранних форм.

- **конструювання:** створення додатку з урахуванням можливості зміни його дизайну.

3. Тестування:

- **юзабіліті-тестування:** тестування додатка різними користувачами, включаючи користувачів з обмеженими можливостями (англ. accessibility testing).

Проектування є процесом визначення архітектури, компонентів, інтерфейсів та інших характеристик системи або її частини (ISO 24765).

Результатом цього процесу є проект - цілісна сукупність моделей, властивостей або характеристик, описаних у формі, придатній для реалізації системи. Поряд з аналізом вимог, проектування є частиною великої стадії життєвого циклу системи, відомої як визначення системи (англ. System definition). Результати цієї стадії служать вхідною інформацією для стадії реалізації (втілення) системи (англ. system realization).

Проектування системи направлено на уявлення системи, відповідно до передбаченої мети, принципам і задумам. Воно включає оцінку і прийняття рішень по вибору таких компонентів системи, які відповідають її архітектурі і укладаються в запропоновані обмеження.

Функціональні вимоги визначають той функціонал чи поведінку, яку має мати програмна система, щоб надавати користувачам можливість виконання своїх завдань відповідно до технічного завдання та заданих вимог користувачів. Програма, що розробляється повинна реалізувати наступні функціональні можливості:

- обмін даними в реальному часі через інтерфейси;
- обробка інформації в реальному часі;
- логічне управління об'єктом;
- відображення інформації на екрані;
- ведення архіву;
- аварійна сигналізація і управління повідомленнями про несправності;

- контроль команд оператора і заборона їх виконання, якщо команда не відповідає алгоритму функціонування об'єкта.

З моменту запуску програма з заданим інтервалом зчитує з пам'яті дані отримані від модулів введення/виведення. Прийняті дані перевіряються на валідність і аналізуються. Залежно від цього визначаються: колір, текст, що зображують на індикаторах, і положення елементів малюнків на активних сторінках. При відсутності поновлення даних за заданий проміжок часу, відбувається оповіщення оператора сполученням «Ні обміну».

При запуску програма запускає графічний інтерфейс на початковій сторінці. Отримані дані відображаються на індикаторах графічної частини програми. Оператор вибирає «Режим», «Операцію» і «Вид операції». Залежно від його вибору програма виводить обрану сторінку на екран.

У режимі «Автоматичний» оператор може управляти об'єктом за допомогою елементів управління - фізичної клавіатури на передній панелі пульта управління, а також віртуальних органів управління. Для кожної сторінки створено спеціальний набір елементів, що забезпечує виконання даної технологічної операції.

У режимі «Налаштування» передбачаються спеціальні технологічні операції, захищені кодом доступу. Вони дають можливість налаштуванню окремих виконавчих механізмів, при цьому частина блокувань відключається. Режим «Резервний» призначений для безпосереднього управління обмеженою кількістю виконавчих механізмів в разі відмови пристроїв СУ. Команди, які обираються оператором за допомогою елементів управління, передаються в СУ.

Технічним завданням визначено, що управління об'єктом має здійснюватися одним оператором. Для виконання цієї вимоги частину СУ являє собою програмовані контролери, що управляють в автоматичному чи напівавтоматичному режимі роботою всіх виконавчих механізмів на підставі аналізу показників датчиків (тиску, кута, положення і т. п.).

На етапі логічного проектування потрібно побудувати модель, в якій будуть показані логічні потоки, які використовуються в роботі програмного

обладнання (ПО), що розробляється. Для кожного потоку зазвичай визначається приблизний тип даних (Рис. 1.11).

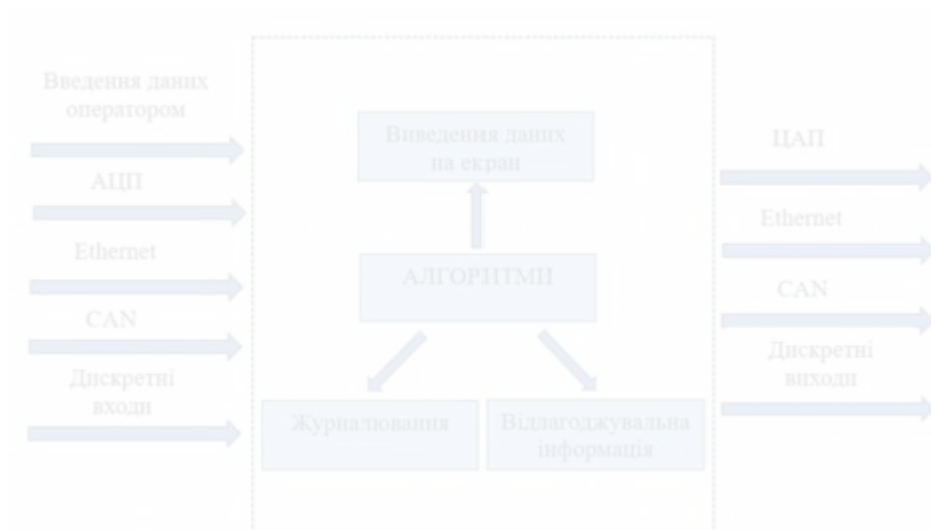


Рис. 1.11. Концептуальне проектування

Після побудови логічних схем слід перейти до фізичного проектування, де для реалізації кожного потоку буде запропоновано апаратне рішення, визначена апаратна частина виробу. Вона вибирається для реалізації потоків, необхідних для роботи системи управління, виходячи зі схем, отриманих на етапі логічного проектування. Програмна частина повинна обиратися виходячи з вимог до системи управління.

1.4. Висновки до розділу

Виходячи з вище наведеної інформації можна зробити наступний **ВИСНОВОК**:

- найбільш технологічним рішенням для роздавальників ліків є пристрій основою якого є лоток, який обертається і який поділено на ту кількість частин яка забезпечує необхідну автономність;
- інтерфейс може бути простий кнопочний та реалізований за допомогою бездротової передачі даних.

РОЗДІЛ 2.**КОНСТРУКЦІЯ РАЗДАВАЛЬНИКІВ ЛІКІВ****2.1. Кінематична схема роздавальника ліків**

Розглянемо кінематичну схему роздавальника ліків.

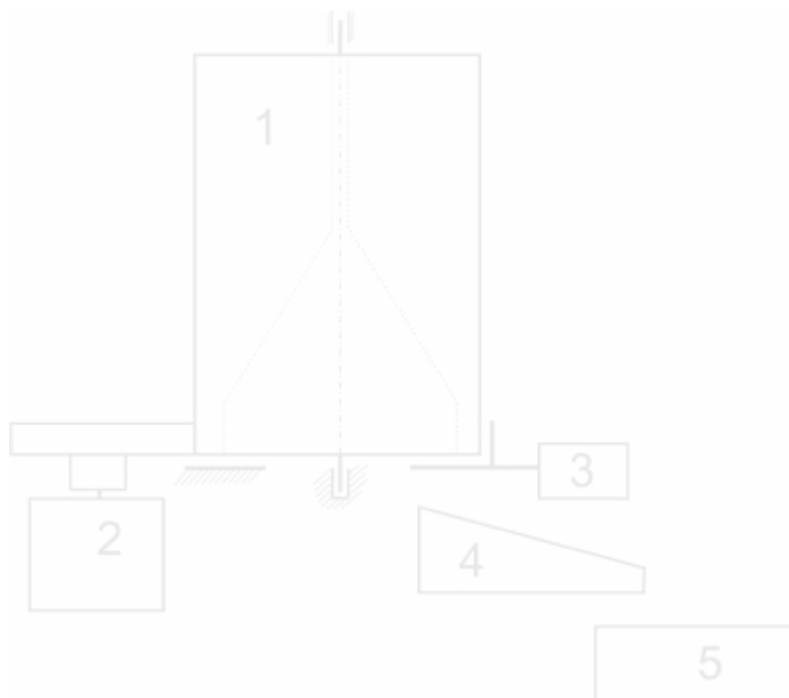


Рис. 2.1 . Кінематична схема роздавальника ліків: 1-бункер з лікарськими препаратами; 2-кроковий привід повороту бункера; 3-дозуючий механізм; 4 -приймальний похилий лоток; 5-лоток для передачі пацієнтові

2.2. Дизайн роздавальника ліків

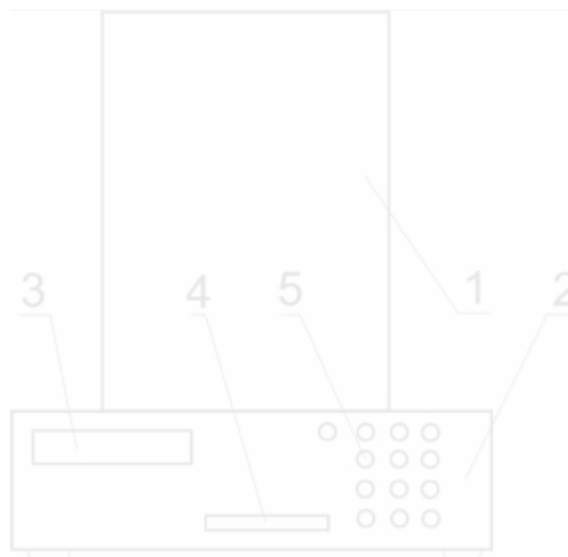


Рис. 2.2. Дизайн пристрою роздавальника ліків: 1 - бункер з лікарськими препаратами; 2 - корпус пристрою; 3 - LCD 16 * 2 індикатор; 4 - вихід лотка роздавальника; 5 - кнопкова панель введення програми роздачі ліків

Корпус і бункер пропонується виготовляти з не дорогого, поширеного пластика (наприклад, полікарбонату). Технологію пропонується використовувати не пов'язану з розробкою і використанням оснащення для термопласт автоматів і засновану на використанні не складних ручних операцій (розкрій по шаблонах, гнуття з використанням ручних пристроїв) останнє дозволить використовувати працю людей з вадами зору для забезпечення останніх роботою.

2.3. Принципова схема роздавальника ліків



Рис. 2.3. Принципова схема роздавальника ліків:

1 блок живлення; 2 центральна система управління; 3-індикатор 16 * 2; 4 кнопкова панель; 5 години реального часу; 6 .драйвер крокової двигуна і виконавчого механізму; 7- кроковий двигун; 8- виконавчий механізм

Принцип роботи роздавальника ліків полягає в тому, що програму прийому лікарських препаратів вводять, погоджуючи з часом здійснення роздачі. У пам'яті мікроконтролера зберігається розклад з необхідною кількістю імпульсів крокового двигуна і з подальшою кількістю спрацювань механізму роздачі (випадок видачі декількох таблеток одного найменування). При настанні часу роздачі генерується переривання, за якими подаються сигнали на спрацювання крокового двигуна і виконавчого механізму (що представляє собою реле) після чого управління передається основній програмі і триває перевірка годин реального часу до наступного збігу.

2.4. Синтез структурно-функціональної схеми пристрою



Рис. 2.4. Структурно-функціональна схема пристрою роздавальників ліків

З рисунку 2.4 видно, що блок живлення повинен бути представлений електротехнічним пристроєм, що забезпечує перетворення вхідної змінної напруги 220V в постійну стабілізовану напругу 5V і 12V, необхідну для живлення, та входить до складу системи мікросхем і приводу крокової двигуна.

В якості блоку живлення пропонується використовувати блок живлення від принтера Epson Stylus Color 600 (рис. 2.5).



Найбільш відповідним рішенням проблеми відображення є GLCD WG12864A індикатор. Цей дисплей виробляє багато компаній, але більшість

основних китайських виробників WINSTAR. Дисплей є монохромною матрицею рідкокристалічних елементів WG12864A з роздільною здатністю 128 x 64 (128 по вертикалі, по горизонталі, 64), внутрішні контролери, KS0108 і внутрішній відео 1 КБ пам'яті (рис. 2.6). В експозиції представлено дві перемички SMD J1 і J2 і їхнє становище залежить від полярності блоку живлення дисплея.

Розводка дисплею відображена в таблиці 2.1.



Рис. 2.6. Дисплей WG12864A

Таблиця 2.1.

26

Таблиця розпіновки підключення дисплея

Pin No.	Symbol	Level	Description
1	GND	0V	Ground
2	V _{DD}	5.0V	Supply voltage for logic
3	V _e	(Variable)	Operating voltage for LCD
4	D/I	H/L	H: Data , L: Instruction
5	R/W	H/L	H: Read (MPU→Module) , L: Write (MPU→Module)
6	E	H	Enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	CS1	H	Select Column 1~ Column 64
16	CS2	H	Select Column 65~ Column 128
17	RST	L	Reset signal
18	Vout	—	Negative Voltage
19	A	—	Power Supply for LED backlight (+)
20	K	—	Power Supply for LED backlight (-)

Розпіновка підключення дисплея представлена в таблиці 2.1.

Для управління дисплеєм потрібно дві шини: - ШД - шина даних 8 біт (8 виходів МК); - ШУ - шина управління 6 біт (6 виходів МК).

На платі дисплея встановлено два контролера KS0108, ці контролери мають внутрішню пам'ять по 512Байт і керують кожен своєю половиною дисплея. Тобто, програмно і апаратно дисплей розбитий на дві частини по 64х64. Звертатися до кожного з них можна як разом, так і окремо. Виходи, які відповідають за це звернення мають номери 15, 16 і зветься CS1 і CS2. Для вибору потрібної половини дисплея необхідно подати високий сигнал на відповідний вихід дисплея. Тип передачі даних про спробу готовності, тому на платі дисплея є окремий вихід під номером 6 і під назвою E. Момент часу, коли дисплей зчитує показання з ШД - це той момент, коли на виході E змінюється

сигнал з високого на низький. Однак для виконання внутрішніх процесів після цієї зміни сигналів необхідно чекати не менше ніж 10 us. Залежно від сигналу на виводі 4, який називається D / I, контролер дисплея розуміє який байт йому був переданий. Вихід D / I визначає дані або інструкції, якщо подати високий сигнал на цей вихід, то дисплей буде розуміти, що на вихід потрапили дані, які необхідно зобразити на матриці за потрібною адресою, а якщо подати низький, то дисплей зрозуміє, що це звернення до адреси або читання стану дисплея. Для того щоб опитати дисплей про стан будь-яких даних або станів, використовується вихід 5 який називається R / W. Цей вихід позначає напрямки даних, якщо на вихід R / W подати низький сигнал, то дисплей буде приймати дані з ШД, а якщо подати високий, то дисплей буде видавати на ШД (дисплей передає дані тільки на ШД при цьому ШУ не змінює) дані про внутрішній стан або про байти даних під певною адресою. Вихід під номером 17 називається RST і коментарів не потребує. При подачі низького сигналу, дисплей скине свою внутрішню пам'ять і вимкнеться. Для його повторного запуску необхідно заново форматувати і записувати нову інформацію. ШД розташована на проводах від 7 до 14, особисто я рекомендую підключити до одного цілого порту МК для того, щоб було простіше виводити дані за один машинний цикл.

Виходи живлення: 1 - загальний і 2 - позитивний вихід.

Живлення необхідно подавати в межах від 4.75 - 5.25, в принципі можна подати напругу нижче, але тоді необхідно збільшувати час ініціалізації і час стрибка. Виходи 3 і 18 призначені для регулювання контрастності зображення апаратно.

Для забезпечення бездротової передачі параметрів налаштування був обраний Bluetooth модуль HC-06. Цей модуль представляє собою плату розмірами 2.7x1.4 см, з 34 виходами з кроком 1.5 мм, розташованих по периметру плати, та має на одному з торців антену. Зовнішній вигляд HC-06 представлений на рис. 2.7, а характеристики в таблиці 2.2.



Рис. 2.7. Зовнішній вигляд HC-06

На платі розташований чіп BC417 від компанії Cambridge Silicon Radio, який забезпечує апаратну підтримку Bluetooth 2.0 + EDR (Enhanced Data Rate). Крім того, є флеш-пам'ять ES29LV800DB-70WGI від Excel Semiconductor на об'єм 8 Мбіт (1 МБ), що використовується для зберігання прошивки та налаштувань. Модуль вміє працювати в двох режимах: UART і управління AT-командами. Модуль має 4 роз'єми для підключення:

- VCC - живлення +5 В;
- GND - заземлення;
- TXD, RXD - UART інтерфейси для спілкування з контролером.
- Bluetooth модуль HC-06 має напругу логічної одиниці (TXD, RXD) 3.3В, а mega - 5В.

Таблиця 2.2

Характеристики HC-06

Характеристики	Показники
Напруга живлення, В	3,3 - 6
Максимальна входна напруга логічної одиниці, В	5
Вихідна напруга логічної одиниці, В	3,3
Максимальний струм споживання, мА	45
Швидкість передачі даних, бод	1200 - 1382400
Аналогові входи	16
Дальність зв'язку в прямій видимості, м	30

2.5. Основні характеристики і можливості ATmega16

В роботі використовується ATmega16 - високопродуктивний 8-розрядний AVR-мікроконтролер з низьким енергоспоживанням (рис. 2.8). Мікроконтролер виготовлено за технологією КМОП, яка, спільно з RICS архітектурою, дозволяє досягти оптимального балансу між продуктивністю та витратою енергії.

Основні переваги Atmega16:

- мікроконтролер може обробляти 131 потужну інструкцію, причому більшість з них виконуються за один машинний цикл.
- має 32 8-розрядних регістра загального призначення і регістри управління вбудованою периферією.
- забезпечує повністю статичну роботу.
- має продуктивність до 16 мільйонів операцій в секунду при тактовій частоті 16 МГц.
- вбудований множувальний пристрій виконує множення за 2 машинних циклу;
- 16 Кбайт внутрішньої системної програмованої флеш-пам'яті;
- пам'ять даних (ОЗУ) - 1 Кбайт;
- пам'ять даних (EEPROM) - 512 байт, розрахована на 100000 циклів читання / запису;
- 32 програмовані лінії введення / виведення (рис. 2.9).

Рис. 2.8. Блок-схема ATmega16

Рис. 2.9. Розташування виходів ATmega16

2.6. Принципова схема пристрою

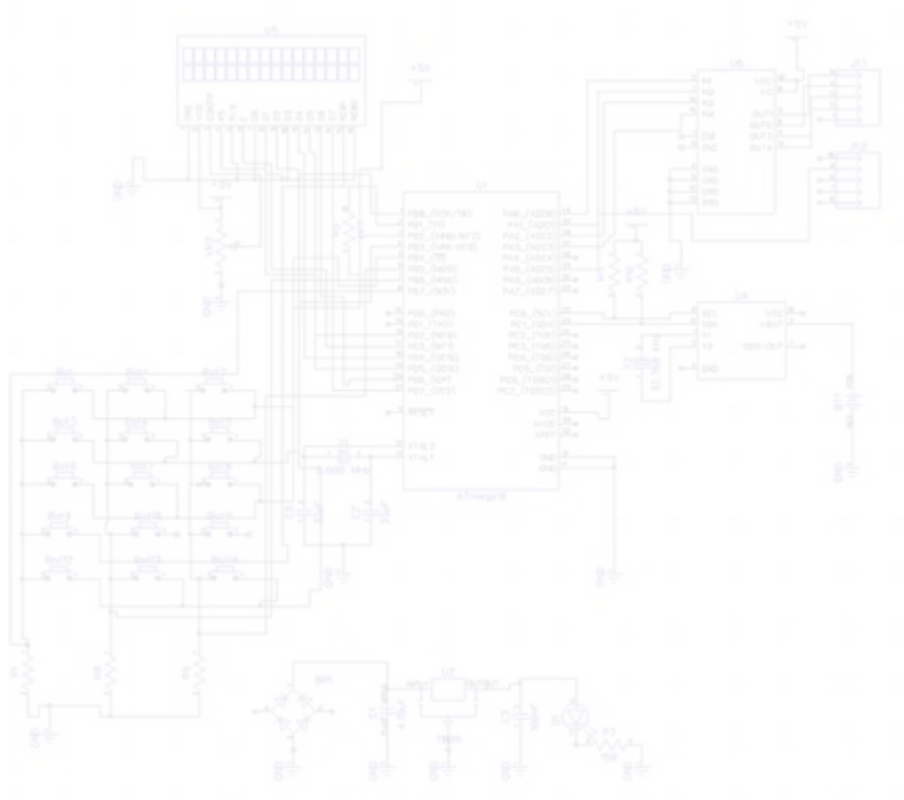


Рис. 2.10. Принципова схема пристрою з кнопочним інтерфейсом

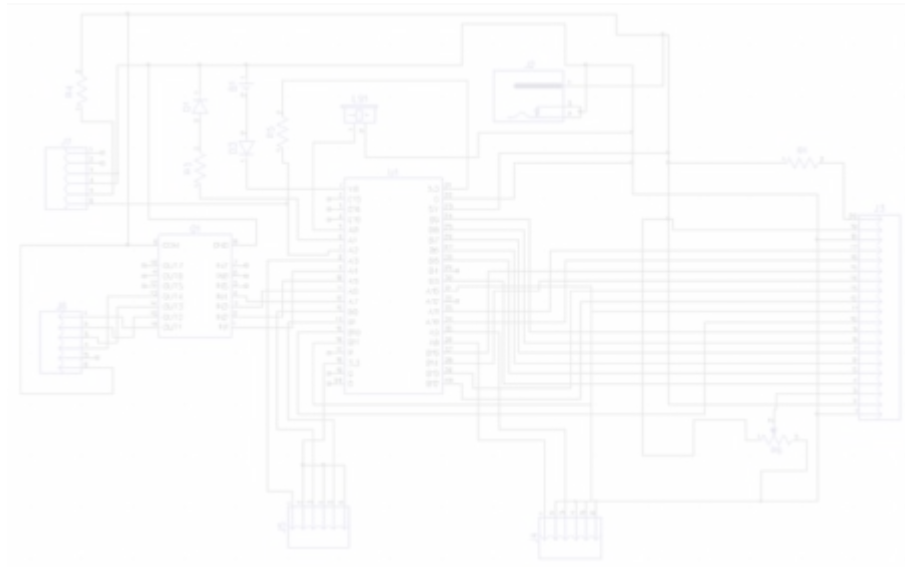


Рис. 2.11. Принципова схема пристрою с інтерфейсом побудованим на енкодері та можливістю підключенням блоку BlueTooth

2.7. Вибір елементної бази

При виборі елементів для ділянок схеми використовувалися критерії, такі як досягнення достатньої швидкодії та низьке споживання потужності.

В якості приладів керування інтерфейсів використовувались:

- кнопочна панель 4 x 4 (Рис.2.12)
- проміжковий варіант з використанням енкодеру ECW1J-b24-BC0024c (Рис. 2.13)



Рис. 2.12. Кнопочна панель 4 х 4



Рис. 2.13. Енкодер ECW1J-b24-BC0024c

2.8. Розробка програми

Вибір середовища розробки.

Для вирішення поставленого завдання був обраний крос компілятор C

[CodeVisionAVR](#).

CodeVisionAVR - це крос-компілятор C, інтегроване середовище розробки (IDE - Integrated Development Environment) та автоматичний генератор програм

(CodeWizardAVR), що розроблені для сімейства AVR-мікроконтролерів фірми Atmel [5].

Програма є 32-бітовим додатком, який працює під операційними системами Windows.

CodeVisionAVR забезпечує виконання майже всіх елементів мови C, які дозволені архітектурою AVR, з деякими доданими характеристиками, які реалізують свої переваги специфіки архітектури AVR.

Компілятор призначений для використання разом з відладчиком AVR Studio від Atmel версії 4.06 або більш пізньої. AVR Studio можна безкоштовно завантажити з веб-сайту: www.atmel.com.

IDE володіє вбудованим програмним забезпеченням для внутрішньої схеми програматора мікросхем AVR, яке дозволяє автоматично передавати скомпільовані програми у мікроконтролерний чіп. Це програмне забезпечення внутрішньої схеми програматора може співпрацювати з пристроями Atmel STK500 / AVRISP / AVRProg, Kanda Systems STK200 + / 300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR і розробницькою платою MicroTronics ATCPU / Mega2000.

Для налагодження розроблюваних систем, які застосовують послідовний зв'язок, IDE має вбудований термінал.

Крім стандартних бібліотек C, компілятор C CodeVisionAVR має бібліотеки для:

- алфавітно-цифрових LCD-модулів;
- шини I2C від Philips;
- температурного датчика LM75 від National Semiconductor;
- годин реального часу PCF8563, PCF8583 від Philips і DS1302, DS1307 від Dallas Semiconductor;
- протоколу 1-Wire від Dallas Semiconductor;
- температурного датчика DS1820 / DS18S20 від Dallas Semiconductor;
- термометра / термостата DS1621 від Dallas Semiconductor;

- EEPROM DS2430 і DS2433 від Dallas Semiconductor;
- SPI;
- управління живленням;
- затримок;
- перетворення коду Грея.

CodeVisionAVR також містить автоматичний генератор програм - CodeWizardAVR, який дозволяє написати за кілька хвилин весь код, необхідний для виконання наступних функцій:

- установка доступу до зовнішньої пам'яті;
- ідентифікація джерела скидання чіпа;
- ініціалізація порту введення / виведення;
- ініціалізація зовнішніх переривань;
- ініціалізація таймерів / лічильників;
- ініціалізація сторожового таймера;
- ініціалізація UART і переривань, керуючих буфером послідовного зв'язку;
- ініціалізація аналогового компаратора;
- ініціалізація АЦП;
- ініціалізація інтерфейсу SPI;
- ініціалізація шини I2C, температурного датчика LM75, термометра / термостата DS1621 і годин реального часу PCF8563, PCF8583, DS1302, DS1307;
- ініціалізація шини 1-Wire і температурного датчика DS1820 / DS18S20;
- ініціалізація LCD-модуля.

Вікно CodeVisionAVR IDE (рис. 2.14) включає численні стандартні елементи Windows, в тому числі: рядок заголовка, де вказані назви програми і відкритого в даний момент файлу проєкту; кнопку мінімізації, що дозволяє тимчасово згорнути вікно програми; кнопку розгортання, що дозволяє розгорнути вікно програми на весь екран і / або повернути в первісний стан;

кнопку закриття, що дозволяє закрити вікно програми (вийти з програми); рядок меню, що дозволяє вибирати команди з різних меню; панель інструментів складається з кнопок, які забезпечують зручний доступ до різноманітних команд, а також є рядок стану, що відображає інформацію про поточний стан програми.

Вікно програми розділене на три області: область редагування, призначена для перегляду і редагування файлів проєкту; вікно Navigator, за допомогою якого здійснюється зручна навігація по всьому проєкту, і вікно Messages, в якому виводяться різні повідомлення про помилки і попередження.



Рис. 2.14. Вікно IDE CodeVisionAVR

Загальна схема програми відображена на рис. 2.15.

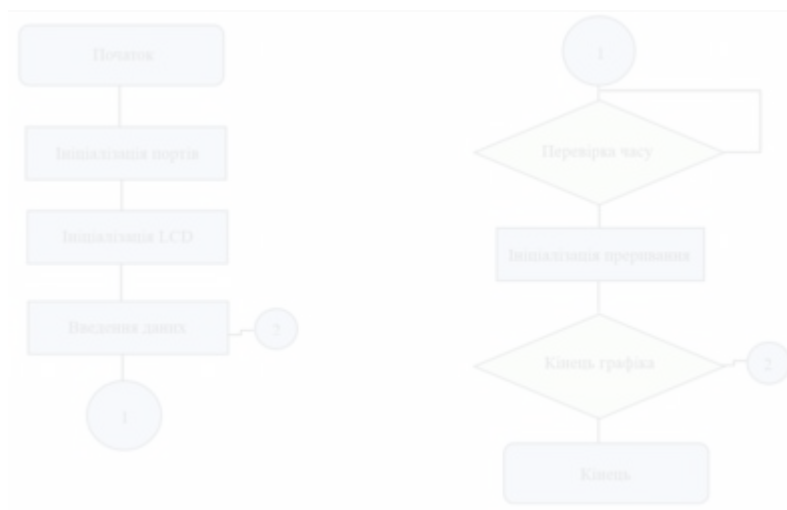


Рис. 2.15. Загальна схема програми

2.9. Розробка друкованої плати

Для розробки друкованої плати було використано додаток Dipe Trace.

Друкована плата зображена на рис 2.16.

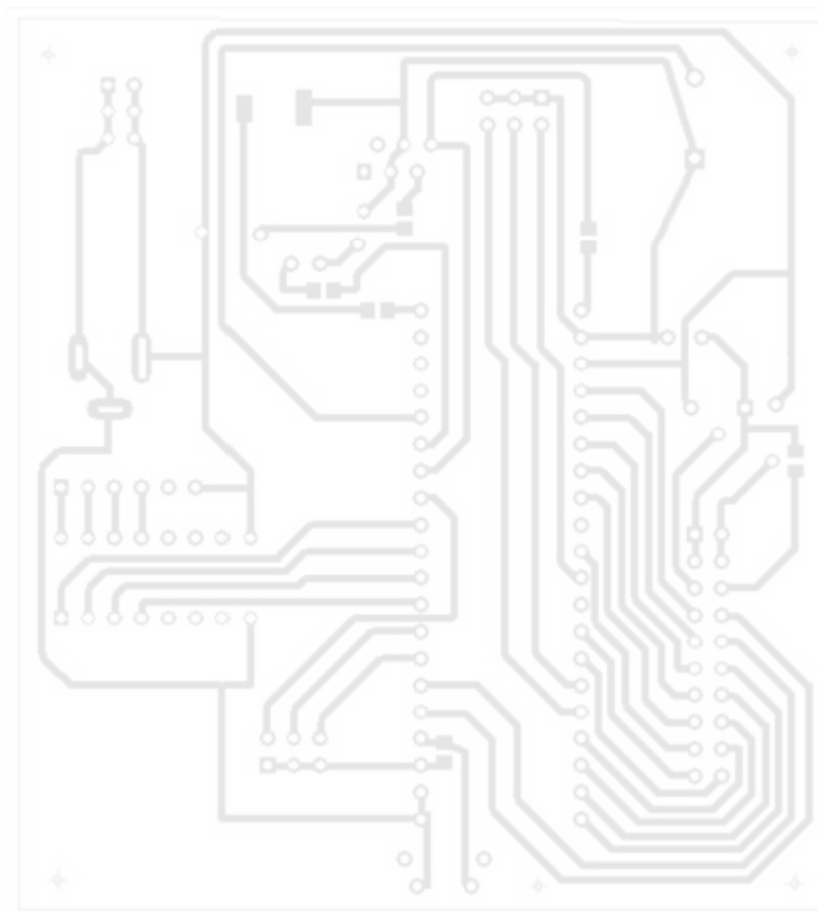


Рис. 2.16. Друкована плата



Рис. 2.13. Пристрій вимірювання в зборі

Розглянемо інтерфейс приладу та його налаштування (рис. 2.14).

Інтерфейс включає екран GLCD для відображення введеної інформації, три нормально розімкнуті кнопки 1.3.4 та екодер 2.

По перше треба розглянути базові характеристики самого роздавальника ліків.

Так в барабані роздавальника присутні 16 відділень 2 з яких є технологічними. Таким чином його можливо використовувати для автоматичної видачі груп препаратів два рази на добу в продовж тижня. Для цього потрібно ввести в програму видачі два часи, це час першого прийому та час другого прийому.

Також при першому включенні потрібно ввести поточний час.



Рис. 2.14. Елементи керування інтерфейсом приладу

Налаштування роздавальника ліків проводиться наступним чином.

Після підключення пристрою до живлення необхідно розмістити механізм барабану роздавальника в відповідне до часу та дня тижня положення, для цього використовуються кнопка 4, після цього потрібно встановити поточний час використовуючи кнопку 1 та енкодер 2, установити два часу прийому ліків використовуючи кнопку 3 та енкодер 2.

Після цього треба одночасним натиском на кнопку 1 та 3 включити робочій режим при цьому на екрані з'явиться напис «прийом» (рис 2.15) і це значить що включився робочій режим та при досягненні вказаного часу барабан повернеться. Ліки які розташовані в комірці барабану випадуть до сосуду який для цього призначений, та пролунає сигнал який сигналізує про необхідність прийому ліків.

Також після включення робочого режиму, використовуючи кнопку 4 пацієнт, може самостійно в інтервалі одного часу до необхідного отримати ліки натиснувши на кнопку.



Рис. 2.15. Екран GLCD після налаштування

2.10. Висновки до розділу

Розглядаючи кнопочий ввід даних для налагодження часу прийому ліків та аналізуючи досвід експлуатації приладу можемо відзначити, що такий спосіб потребує достатньо великого числа вільних виводів мікроконтролера та потребує достатньо високих вмінь обслуговуючого персоналу та часу на налагодження.

Ці факти підтверджують всесвітню тенденцію використання для керування побутовими пристроями бездротових технологій та говорить про необхідність проектування інтерфейсу керування роздавальником ліків за допомогою ANDROID пристрою та BlueTooth модулю.

РОЗДІЛ 3

РОЗРОБКА ДОДАТКА ДЛЯ ПРИСТРОЇВ НА ПЛАТФОРМІ ANDROID

3.1. Вибір інструментарію розробки

Android - це комплексне програмне забезпечення для мобільних пристроїв, яке охоплює операційну систему, проміжний шар (middleware) і основні користувацькі додатки, такі як електронна пошта, календар, карти, браузер, контакти та інші.

Загальна архітектура системи Android приведена на рисунку 3.1.



Рис. 3.1. Архітектура операційної системи Android

- Рівень додатків (Applications):

У склад Android включено набір основних додатків, які включають клієнти електронної пошти та SMS, календар, різноманітні карти, браузер, програму для управління контактами та інші функціональність. Всі додатки, які працюють на платформі Android, розроблені на мові програмування Java.

- Рівень каркаса додатків (Application Framework):

У Android існує можливість використовувати повний функціонал API, який використовується в основних додатках. Структура платформи організована

таким чином, що будь-який додаток може користуватися функціональністю інших програм, якщо ці програми надають доступ до своєї функціональності. Таким чином, архітектура платформи сприяє багаторазовому використанню компонентів операційної системи та додатків.

Всі додатки ґрунтуються на наборі систем та служб.

1. Система уявлень (View System) - це розширений набір ідей, який містить багатий набір функціональності для створення зовнішнього вигляду додатків. Він включає різноманітні компоненти, такі як списки, таблиці, поля введення, кнопки та інші елементи.
2. Контент - провайдери (Content Providers) - це сервіси, які дозволяють додаткам отримувати доступ до даних, що належать іншим додаткам, а також надавати доступ до своїх власних даних.
3. Менеджер ресурсів (Resource Manager) призначений для отримання доступу до різноманітних ресурсів, включаючи текстові, графічні та інші типи.
4. Менеджер сповіщень (Notification Manager) забезпечує можливість будь-якому додатку відображати повідомлення для користувача у рядку статусу.
5. Менеджер дій (Activity Manager) керує життєвим циклом додатків і забезпечує систему навігації, що дозволяє відстежувати історію взаємодії з додатками та їх діями.

- Рівень бібліотек (Libraries):

У платформі Android включено набір бібліотек C/C++, які використовуються різними компонентами операційної системи. Розробникам надається доступ до функцій цих бібліотек через використання Application Framework. Ось кілька прикладів цих бібліотек:

1. System C library - BSD - стандартна системна бібліотека C (libc) для вбудованих пристроїв, які працюють на базі операційної системи Linux, реалізована в платформі Android.

2. Media Libraries - бібліотеки, які базуються на PacketVideo's OpenCORE, використовуються для забезпечення підтримки програвання та запису популярних аудіо- та відеоформатів, таких як MPEG4, H.264, MP3, AAC, AMR, JPG, PNG та інші.
3. Менеджер поверхонь (Surface Manager) регулює доступ до графічних шарів 2D і 3D у підсистемі відображення.
4. LibWebCore - це сучасний движок веб-браузера, який надає всю функціональність вбудованого Android-браузера.
5. SGL - це движок, призначений для роботи з 2D-графікою.
6. Бібліотеки для 3D (3D libraries) - це движок, спрямований на роботу з 3D-графікою, заснований на API OpenGL ES 1.0.
7. FreeType - це бібліотека, створена для роботи з шрифтами.
8. SQLite - потужний легкий движок для операцій з реляційними базами даних.

- Рівень середовища виконання (Android Runtime):

Android має в своєму складі набір бібліотек ядра, які надають значну частину функціональності, аналогічну бібліотекам ядра мови Java. У платформі використовується віртуальна машина Dalvik, яка оптимізована для роботи з реєстром, на відміну від стандартної стек-орієнтованої віртуальної машини Java. Кожен програмний продукт виконується в власному процесі з власним екземпляром віртуальної машини. Dalvik використовує формат Dalvik Executable (*.dex), який оптимізований для ефективного використання пам'яті додатками. Це досягається завдяки базовим функціям ядра Linux, таким як організація потокової обробки та низькорівневе керування пам'яттю. Байт-код Java, на якому написані ваші програми, компілюється в формат dex за допомогою утиліти dx, яка входить до складу набору інструментів розробки (SDK).

- Рівень ядра Linux (Linux Kernel):

Android базується на версії 2.6 операційної системи Linux, що дає платформі доступ до різних системних служб ядра, наприклад, управління

пам'яттю та процесами, забезпечення безпеки, мережеві функції та взаємодія з драйверами. Крім того, ядро виступає як шар абстракції між апаратним і програмним забезпеченням.

В даний час для розробки Android-додатків використовується два середовища розробки: Eclipse IDE і Android Studio.

Ми обираємо середовище розробки Android Studio, На першому етапі будемо додати додаток (Рис. 3.2).

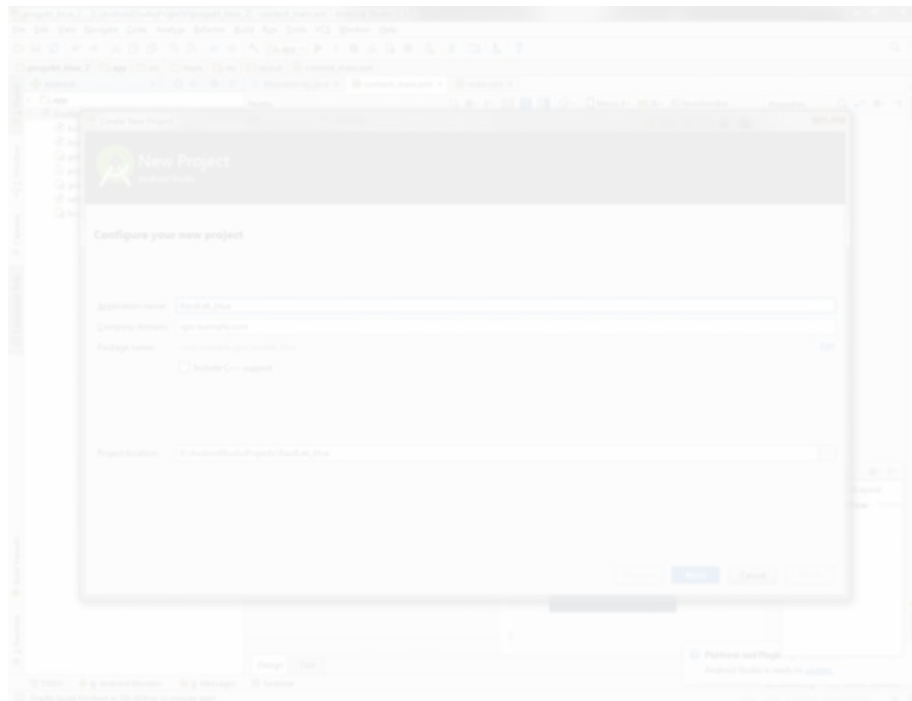


Рис. 3.2. Введення ім'я

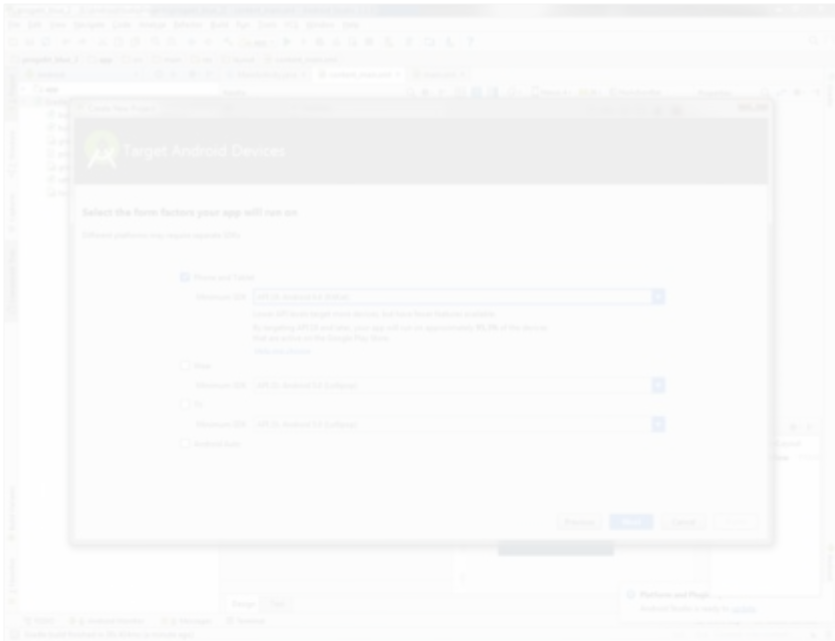


Рис. 3.3. Вибір версії API

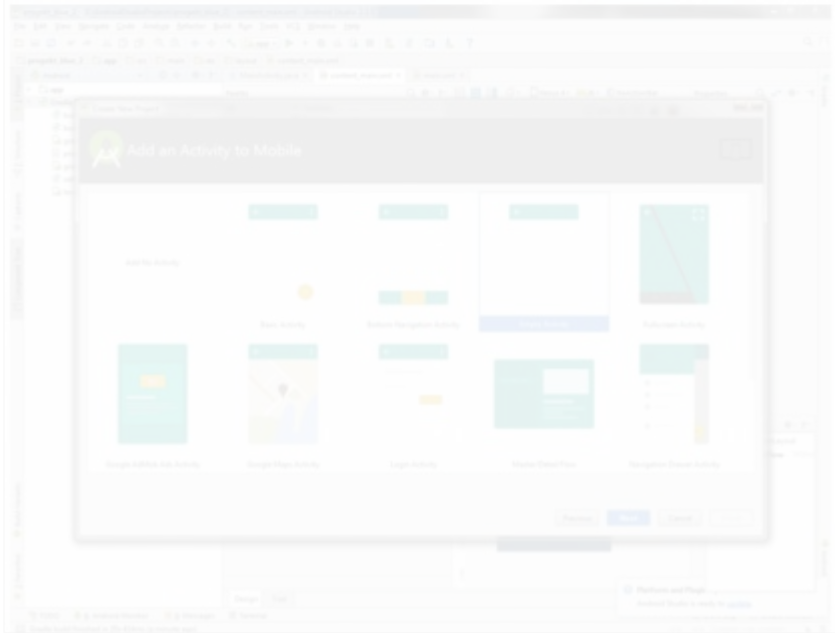


Рис. 3.4.Вибір формиActivity

Рис. 3.5. Фіксація вибраних параметрів та відкриття вікон проєктування

Рис. 3.6. Проектування форми

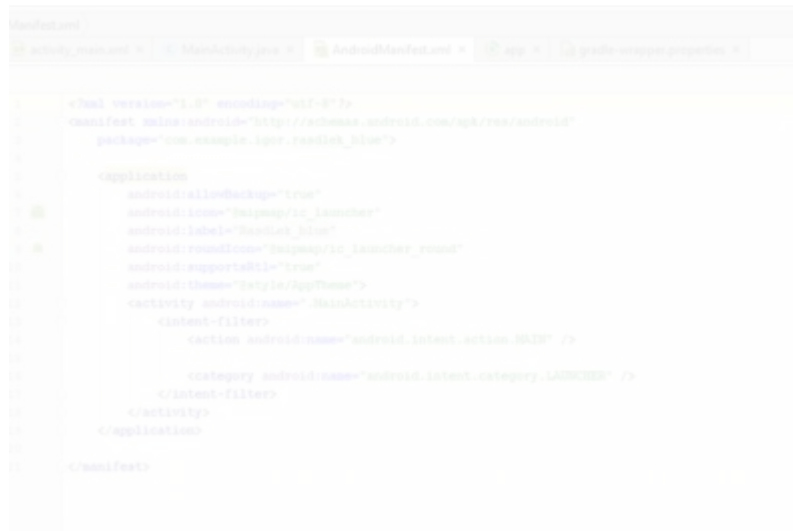


Рис. 3.7. Підключення BLUETOOTH

Після рядка package = «com.example.dima.blueardu» додаємо дозвіл на використання bluetooth:

```
Дл<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission
android:name="android.permission.BLUETOOTH_ADMIN"/>
```

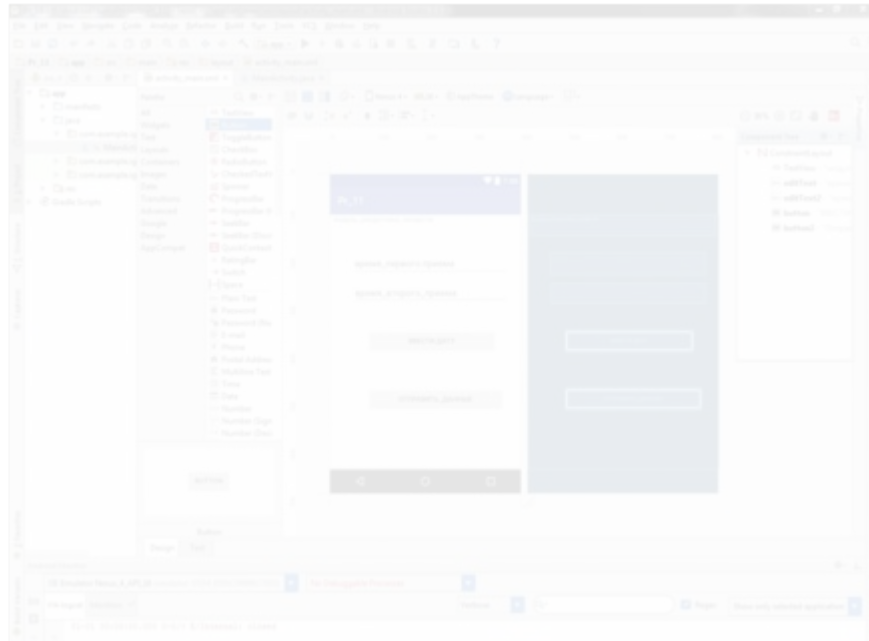


Рис. 3.8. Розроблена форма

3.2. Висновки до розділу

Оцінюючи процес проектування налагоджування інтерфейсу з використанням бездротової технології, яка базується на використанні Bluetooth та ОС Android можна зробити висновок що цей процес не вельми складний але саме використання такого підходу потребує наявності мобільного пристрою. Але маючи на увазі практично повсюдне використання цих приладів дає змогу вважати що такий підхід дуже перспективний в майбутньому. Також використання цієї технології дозволяє зменшити навантаження кількості виходів мікроконтролера, що іноді дуже актуально.

ВИСНОВКИ

У магістерській роботі було розглянуто проєктування людино-машинного інтерфейсу конфігурації роздавальників ліків.

Для досягнення поставленої мети, яка стосувалась підходів до проєктування людино-машинного інтерфейсу конфігурації роздавальників ліків було виконано наступні кроки:

було проаналізовано теоретичні передумови для розробки пристроїв роздавальників лік, типи пристроїв та класифікацію людино-машинних інтерфейсів;

було розроблено та виготовлено пристрій роздавальника ліків, проведено його дослідна експлуатація;

було розроблено додаток до мобільно приладу з ОС Android для реалізації бездротового налагодження інтерфейсу налаштування роздавальника ліків.

На основі корисної моделі вирішено завдання створення зручного інтерфейсу для відображення графіку прийому, можливості накопичення і відображення інформації про виконання графіку прийому, а також можливості використання пристрою з різними ємностями для препаратів.

Це завдання вирішено шляхом розробки універсальної кришки, що складається з корпусу з технологічними отворами для розміщення кнопок, екрана, світлодіода і щілини для поширення звукових хвиль. У корпусі кришки розташована електронна двостороння плата з вбудованою схемою. Цей пристрій дозволяє користувачеві встановлювати кількість та періоди прийому, а також виводить на екран інформацію про проведені та залишкові прийоми. Щоб точно визначити час прийняття препаратів, пристрій оснащено годинником.

Універсальність кришки забезпечена набором знімних різьбових вкладишів, які фіксуються позаду плати і призначені для різних ємностей. Пристрій має систему світлової та звукової сигналізації для інформування про час прийому, яка активується вказаним користувачем часом та контролюється відкриттям горловини флакона, яке свідчить про факт прийому препарату.

Оцінюючи процес проєктування та налагодження інтерфейсу, який використовує бездротову технологію на базі Bluetooth та операційної системи Android, можна прийти до висновку, що цей процес не є надто складним. Проте важливо враховувати, що для використання такого підходу потрібно наявність мобільного пристрою. Однак, враховуючи практично всеосяжне використання

51

цих пристроїв, можна зазначити, що такий підхід має великий потенціал у майбутньому.

Схожість

Джерела з Інтернету

42

1	https://studfile.net/preview/5010347	0.75%
2	https://dl.sumdu.edu.ua/textbooks/109154/414708/index.html	0.69%
3	https://er.knutd.edu.ua/bitstream/123456789/19448/1/Dyplom122_Yehorov_Astistova.pdf	4 джерела 0.51%
4	https://divovo.in.ua/referat-magisterseka-disertaciya-skladayetesya-zi-vstupu-pyati.html?page=4	0.45%
5	http://inmad.vntu.edu.ua/portal/static/4FAC1A77-F524-4F1B-963B-3A98EE6AFC41.pdf	0.43%
6	https://essuir.sumdu.edu.ua/bitstream-download/123456789/82229/1/Fedirka_mag_rob.pdf	0.43%
7	https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2347/1/401%20%D0%94%D0%B0%D1%80%D1%96%D0%B9%20%D0%90...	0.42%
8	https://uapatents.com/5-90548-pristriij-dlya-organizaci-prijjomu-riznikh-preparativ-za-zadanim-rozkladom.html	2 джерела 0.42%
9	http://repository.kpi.kharkov.ua/bitstream/KhPI-Press/62424/1/Book_2023_Ananieva_Zahalna_tekhnolohiia.pdf	0.4%
10	https://ela.kpi.ua/handle/123456789/49798	2 джерела 0.38%
11	https://essuir.sumdu.edu.ua/handle/123456789/90977	2 джерела 0.35%
12	https://123dok.com/document/nq77mdq6-perancangan-alat-kadar-alkohol-menggunakan-sensor-berbasis-atmega.htm	6 джерел 0.3%
13	http://dspace.luguniv.edu.ua/xmlui/bitstream/handle/123456789/4258/Metodrekom%20vasin.pdf?isAllowed=y&sequence=1	0.26%
14	http://dspace.luguniv.edu.ua/jspui/bitstream/123456789/9204/1/%D0%A8%D0%B0%D1%85%D0%B1%D0%B0%D0%B7%27%D1...	0.26%
15	http://www.vxsecurity.sg/author/jacobsoo/page/3	0.24%
16	http://eir.nuos.edu.ua/xmlui/bitstream/handle/123456789/3203/Ryabenkyi%207.pdf?isAllowed=y&sequence=1	4 джерела 0.21%
17	http://ni.biz.ua/3/3_9/3_9256_tverdie-lekarstvennie-formi.html	0.18%
18	http://elar.khmnu.edu.ua/jspui/bitstream/123456789/12849/1/2022_%D0%94%D0%A0%D0%9C_%D0%9C%D0%BE%D1%80%D0...	0.16%
19	http://ir.stu.cn.ua/handle/123456789/6961/browse?locale-attribute=uk&type=title	0.16%
20	https://www.bibliofond.ru/view.aspx?id=669921	0.14%

21	https://docplayer.net/55420572-Diplomna-robota-pershogo-bakalavrskogo-rivnya-vishchoyi-osviti.html	0.14%
22	https://ela.kpi.ua/handle/123456789/28931	7 джерел 0.14%

