Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“____” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
Програмне забезпечення мікропроцесорної системи обробки даних серцебиття на платформі Arduino

КбПЗ_2024

Виконав здобувач вищої освіти
IV курсу, групи КІ21-3СК
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____Пересунько Д.В.
«____» _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
Босько В.В
«____» _____ 2024 р.
Рецензент _____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *бакалавр*
Галузь знань ._12 _____*"*Інформаційні технології*"*_____
Спеціальність *123 "*Комп'ютерна інженерія*"*
Освітньо-професійна (освітньо-наукова) програма _*"*Комп'ютерна інженерія*"*_

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
"____" _____2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Пересуньку Данилу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення мікропроцесорної системи обробки даних серцебиття на платформі Arduino*
2. Керівник роботи *Босько Віктор Васильвич, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 01.04.2024 року № 136-02
3. Строк подання роботи до захисту *18.05.2024 р.*
4. Мета та завдання випускної кваліфікаційної роботи. *Метою роботи є розробка програмного мікропроцесорної системи обробки даних серцебиття на платформі Arduino*
5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Призначення та область використання.
2.Перегляд аналогічних існуючих систем.
3. Опис і обґрунтування проектних рішень.
4. Етапи програмування системи.
5. Впровадження системи в промислову експлуатацію
6. Висновки.
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи	*1 аркуш*
Функціональна схема системи	*1 аркуш*
Діаграма процесів	*1 аркуш*
Блок-схема алгоритму роботи системи	*2 аркуша*

7. Дата видачі завдання « » 20 р.

КПЗ 2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв’язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.05.2024 р.	
8.	Попередній захист роботи	18.05.2024 р.	

КОПЗ_2024

Дата видачі завдання
«__»_____2024р.

Підпис керівника

_____ _____
(прізвище та ініціали)

Завдання прийнято до виконання
«__»_____2024р.

Підпис здобувача

_____ _____
(прізвище та ініціали)

АНОТАЦІЯ

Пересунько Д.В. Програмне забезпечення мікропроцесорної системи обробки даних серцебиття на платформі Arduino

123 Комп'ютерна Інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

Кваліфікаційна бакалаврська робота присвячена розробці пристрою, що реєструє зміну різницю потенціалів між двома точками при роботі серця. За отриманими даними проводиться розрахунок пульсу пацієнта, а також відображаються отримані дані на моніторі ПК.

Також детально досліджена платформа Arduino, її технічні можливості та особливості програмування вбудованого мікроконтролера.

Досліджено методи дискретизації аналогового сигналу в цифровий.

Розроблено алгоритм вимірювання пульсу за отриманим сигналом.

Дану роботу рекомендовано використовувати у якості основи для подальшого вивчення технологій та розробки нових систем.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10/11/ та Android.

Програму розроблено в середовищі C++.

Ключові слова: Arduino, мікроконтролер, дискретизація, швидке перетворення Фур'є, вимірювання пульсу.

КбПЗ_2024

ANNOTATION

Peresunko D.V. Software of a microprocessor-based heartbeat data processing system on the Arduino platform

123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

The qualifying bachelor's work is devoted to the development of a device that registers the change in the potential difference between two points during the operation of the heart. Based on the received data, the patient's heart rate is calculated, and the received data is also displayed on the PC monitor.

The Arduino platform, its technical capabilities and features of programming the built-in microcontroller are also studied in detail.

The methods of discretization of an analog signal into a digital one have been studied.

An algorithm for pulse measurement based on the received signal has been developed.

This work is recommended to be used as a basis for further study of technologies and development of new systems.

The program can be used on personal computers of the IBM PC architecture with Windows XP/Vista/7/8/10/11/ and Android.

The program was developed in the C++ environment.

Keywords: Arduino, microcontroller, sampling, fast Fourier transform, pulse measurement.

КОПЗ – 2024

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ 2
ВСТУП..... 3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ 5
1.1 Призначення системи..... 10
1.2 Область застосування..... 11
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ 13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти) 13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування..... 20
2.3 Розгорнута постановка завдання 22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ 24
3.1 Опис функціонування системи 24
3.2 Розробка структурної схеми..... 38
3.3 Розробка функціональної схеми 41
3.4 Розробка діаграми процесів..... 43
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ..... 46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи..... 59
4.2 Захист розробленого програмного забезпечення..... 64
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ..... 68
6 ОСНОВНІ ВИСНОВКИ..... 73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 74

КПЗ – 2024

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
BIOS	–	Basic Input-Output System
LINQ	–	Language Integrated Query
МК	–	Мікроконтролер
МДВВ	–	Універсальний вимірювач
ОЗП	–	оперативний запам’ятовуючий пристрій
ПЗМ	–	постійний запам’ятовуючий пристрій
ПЛК	–	Програмований логічний контролер
ТП	–	Технологічний процес
ПЗзО	–	пристрій зв'язку з об'єктом
АЦП	–	аналогово-цифровий перетворювач
ШІМ	–	широко-імпульсна модуляція
АСУ	–	автоматизована система управління
IoT	–	(Interner of Things) Інтернет речей

КбПЗ-2024

ВСТУП

Актуальність теми. Сьогоднішня ера цифровізації охоплює широкий спектр сфер життєдіяльності людини, а особливо виразно це проявляється у медицині та охороні здоров'я. Використання цифрових технологій дозволяє значно підвищити ефективність медичного обслуговування за рахунок автоматизації збору даних, їх аналізу та моніторингу стану пацієнтів. Одним з важливих напрямків у сфері здоров'я є моніторинг серцевої активності, який відіграє ключову роль у виявленні та запобіганні кардіологічних захворювань. Традиційно, для моніторингу серцебиття використовують складне медичне обладнання, доступ до якого є обмеженим через високу вартість та потребу у спеціальних навичках для його використання.

В цьому контексті розробка доступних і ефективних технологій, які б могли використовуватися на повсякденній основі без спеціальних медичних знань, стає актуальною та важливою задачею. Саме тому, метою цієї бакалаврської роботи є розробка програмного та апаратного забезпечення для мікропроцесорної системи обробки даних серцебиття на платформі Arduino, яка дозволить широкому колу користувачів самостійно моніторити своє здоров'я без потреби використання дорогого обладнання.

Мета і завдання дослідження. Метою роботи є програмне забезпечення та розробка пристрою для постійного моніторингу здоров’я.

Важливістю цієї теми є забезпечення доступності медичних технологій для населення, зниження економічного навантаження на систему охорони здоров'я та підвищення обізнаності людей щодо їхнього здоров'я. Дослідження в рамках даної роботи охоплює аналіз існуючих рішень у сфері моніторингу серцебиття, розробку власної системи на базі відкритих технологій і оцінку її ефективності через серію тестів і експериментів.

КОПЗ_2024

Практична цінність отриманих результатів. Робота має практичне значення, оскільки запропоноване рішення може бути використане не тільки в медичних установах, але і вдома, що робить процес моніторингу здоров'я більш незалежним і доступним. Структура бакалаврської роботи передбачає огляд теоретичних аспектів мікропроцесорних систем, аналіз вимог до системи обробки даних серцебиття, проектування і реалізацію програмного та апаратного забезпечення, тестування розробленої системи, а також аналіз отриманих результатів та формулювання висновків. В даний час спостерігається стрімкий розвиток різноманітної медичної техніки, включаючи засоби та технології для дослідження організму. Ідея здається досить очевидною: ми можемо відстежувати різні параметри організму за допомогою приладів, таких як біоструми, біопотенціали серця, мозку, м'язів та інших органів. У ідеалі людина могла б бути "покрита" сенсорами, подібно до новорічної ялинки. Потім, застосовуючи різні методи лікування, від ліків до різних фізичних, розумових, емоційних та дихальних вправ, змінюючи режим праці та відпочинку, харчування тощо, ми могли б відслідковувати, як ці дії впливають на організм.

Ці технології використовуються не лише для лікування, а й для розвитку фізичних та розумових здібностей. Наприклад, спортсмени, відстежуючи біоструми своїх м'язів, можуть зробити висновки про корисність різних вправ для цих м'язів. Той же принцип застосовується до тренування серцевого м'яза за допомогою електрокардіографів. Зараз широко відомі так звані майнд-машини, де різноманітні стимуляції, такі як світлові, звукові та тактильні, спрямовані на поліпшення пам'яті, навчальної здатності, емоційного настрою та інше. Для відстеження ефективності майнд-машин застосовуються прилади для фіксації енцефалограм мозку - енцефалографи.

КОПЗ_2024

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

Сучасний світ охоплений пошуками і реалізаціями інноваційних технологій, що допомагають зберігати та покращувати здоров'я людини. Однією з ключових задач у медичній інформатики є розробка надійних та доступних систем для моніторингу життєво важливих показників організму, зокрема ритму серцебиття. Вимірювання та аналіз серцебиття має велике значення для вчасного виявлення та попередження різних серцево-судинних захворювань.

Ринок пропонує багато варіантів технологій для моніторингу серцевої діяльності, проте багато з них мають високу вартість або потребують спеціалізованого обладнання та програмного забезпечення. Це створює бар'єри для їх широкого розповсюдження та доступності серед населення, особливо у регіонах з обмеженими медичними ресурсами.

З огляду на це, існує необхідність створення економічно ефективної та легко доступної системи для моніторингу серцебиття, що може бути використана як в медичних установах, так і вдома. Платформа Arduino пропонує гнучкість та відкритість, що робить її ідеальною базою для розробки подібних медичних пристроїв. Використання мікропроцесорних систем на базі Arduino може допомогти в розробці недорогого, але ефективного рішення для обробки та аналізу даних серцебиття, доступного для широкого кола користувачів.

Таким чином, проблема, яка стоїть перед нами, полягає у необхідності розробки програмної та апаратної частин мікропроцесорної системи, здатної збирати, обробляти, відображати та записувати дані про серцебиття в реальному часі. Система має бути простою у використанні, доступною за вартістю та мобільною, щоб користувачі могли використовувати її без спеціалізованих навичок чи знань.

Електрокардіограма має перевагу в тому, що надає можливість швидко отримати результат оцінки роботи серця. Під час дослідження дані про серцеву

КПЗ–2024

активність негайно фіксуються на паперовій стрічці, яка повільно проходить через систему апарату ЕКГ. На сучасному обладнанні значення можуть відображатися на моніторі комп'ютера і потім друкуватися за допомогою принтера. Однак, незалежно від способу, коли ви покидаєте процедурний кабінет, в ваших руках вже є результат електрокардіограми, який ви бажаєте прочитати якнайшвидше. Розшифрування ЕКГ дозволить зробити висновок щодо наявності або відсутності причин для занепокоєння.

Азбука електрокардіограм

Схема роботи серця представляє собою складну криву безперервну лінію, схожу на синусоїду, з численними відмітками і позначеннями в буквеному і цифровому вираженні. Спочатку може здатися, що розшифрувати та дати висновок по ЕКГ може лише висококваліфікований медичний фахівець - професор медичного інституту, доктор наук або кардіолог з великим досвідом роботи. Однак це не зовсім так. Аналіз ЕКГ дійсно потребує високого рівня уваги, концентрації, точності та знання алгебраїчних основ і алгоритмів. Проте, якщо розібратися і навчитися, то процес розшифрування може стати цікавим.

Навички читання кардіограми та формування висновків необхідні не тільки для кардіологів. Звичайно, спеціалісти цієї галузі можуть витягнути набагато більше інформації з кардіограми щодо роботи серця. Але навчитися проводити дослідження та читати кардіограму повинні і лікарі загальної практики, особливо фельдшери швидкої допомоги. Проведення дослідження та інтерпретація ЕКГ перед наданням допомоги в лікарні дозволяє своєчасно надати ефективну допомогу, наприклад, при інфаркті, і врятувати життя пацієнта.

Цікавість, підвищена обізнаність про власний стан здоров'я та навіть недовіра до лікаря часто підштовхують людей до бажання самостійно навчитися читати ЕКГ. Проте перше звернення до медичного довідника зазвичай розгублює, оскільки велика кількість термінів і абревіатур здаються надзвичайно складними для розуміння. Однак це не причина відмовлятися від ідеї зазирнути за лаштунки кардіології. В першу чергу важливо зрозуміти, що саме

КПЗ_2024

відображається на кардіограмі.

Що відображено на малюнку ЕКГ

Робота серця з точки зору фізики відображає автоматичний перехід від фази деполяризації до фази реполяризації серцевого м'яза. Іншими словами, відбувається постійна зміна станів скорочення і розслаблення м'язової тканини, під час яких збудження клітин міокарда чергується з їх відновленням.

Для того щоб навчитися інтерпретувати схеми ЕКГ, важливо знати, з яких елементів вони складаються:

-зубець: опукла або увігнута відносно горизонтальної осі частина кривої;

-сегмент: прямий відрізок лінії між двома сусідніми зубцями;

-інтервал: сукупність зубця та сегмента.

Аналіз окремих елементів кардіограми

При формулюванні висновку за результатами ЕКГ, зубці оцінюються за амплітудою на вертикальній осі та тривалістю на горизонтальній. Кожному з зубців в рамках одного циклу присвоєна своя буква латинського алфавіту, що характеризує проходження електричного імпульсу через певну частину серця:

- зубець P: описує відповідь передсердь на поширення електричного імпульсу в них. У здоровому стані має позитивне значення, округлений вершину, спрямований вгору, з висотою до 2,5 мм і тривалістю не більше 0,1 с. Патологічним вважається загострена форма Р-зубця, характерна для гіпертрофії правого передсердя або роздвоєна вершина при гіпертрофії лівого;

- зубець Q: характеризує поширення імпульсу в міжшлуночковій перегородці. У нормі має слабко виражене від'ємне значення та тривалість всього 0,03 с;

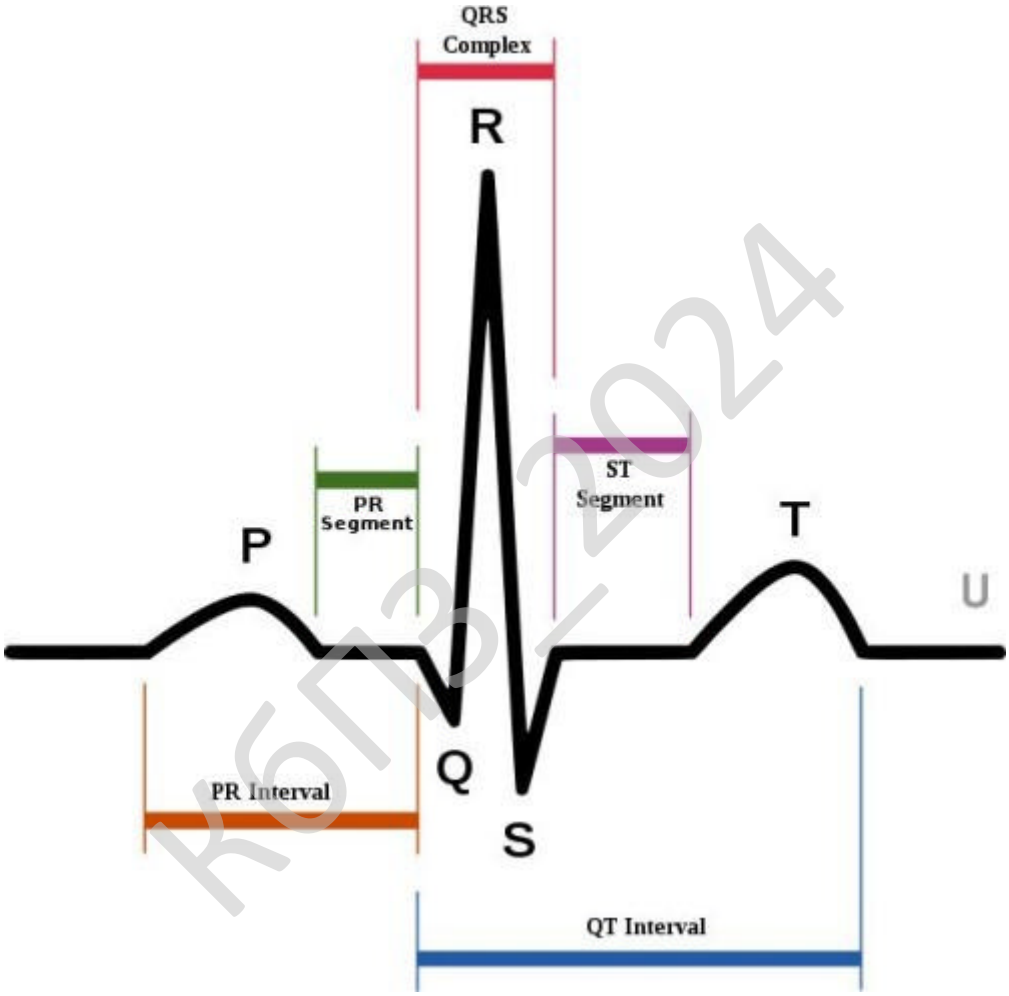
- зубець R: описує проходження електричного сигналу по міокарду шлуночків. За амплітудою найбільший, але тривалість в нормі не перевищує значення Q;

- зубець S: визначає завершення збудження в шлуночках серця. Має негативне значення і невелику глибину - всього 2 мм;

КПЗ-2024

- зубець Т: показник відновлення потенціалу в м'язовій тканині серця. У нормі має позитивне значення і підноситься над горизонтальною віссю не більше як на третину від амплітуди R-зубця. Високий Т-елемент свідчить про вегетативні порушення серцевої активності, наприклад, при гіперкаліємії;

- зубець U: рідко реєструється на ЕКГ. У нормі має висоту до 2 мм і може вказувати на брадикардію або відображати фізичне навантаження, наприклад, у спортсменів.



КПЗ 2024

Рисунок 1.1 – Період роботи серця

Висновок по роботи серця включає в себе перевірку сегментів на ЕКГ-смузі. Кожен сегмент вимірюється від закінчення одного піку до початку наступного. Особливу увагу приділяють сегментам PQ та ST. Їх оцінка включає вимірювання довжини та висоти підйому над ізоелектричною лінією, яка є

горизонтальною основою. Зазвичай, цей підйом не має перевищувати 1 мм. Тривалість сегментів корелює з частотою пульсу, що може вказувати на аномалії в серцевому ритмі.

Робота серцевого м'яза в тимчасових інтервалах

Для коректного аналізу інтервалів на ЕКГ, особливий акцент слід зробити на їх тривалість, адже вона відображає швидкість розповсюдження електричних сигналів у різних частинах серцевого м'яза та реакцію м'язової тканини на ці імпульси. Зокрема, нормальна тривалість інтервалу QT становить 0,45 секунди, і його збільшення може свідчити про такі проблеми, як ішемія або атеросклероз. Також тривалість інтервалів безпосередньо пов'язана з функціональністю серцевого м'яза протягом часу.

Вивчення ЕКГ для визначення серцевого ритму також не є складним. Основним показником тут є інтервал RR, який вимірюється як відстань між двома послідовними вершинами позитивних зубців. Для здорової дорослої людини в спокійному стані цей інтервал зазвичай варіює від 70 до 80 ударів на хвилину, і коливання цієї відстані не повинні перевищувати 10%. Правильний, регулярний ритм вказує на синусову природу ритму на ЕКГ. Наявність інших типів ритму може вказувати на патологічні зміни у серці, у цих випадках лікарі детально аналізують максимальні та мінімальні частоти серцевих скорочень для виявлення потенційних джерел порушень ритму.

План інтерпретації малюнка ЕКГ

Отримати розуміння результатів ЕКГ можна шляхом використання спеціального плану, який спрощує процес інтерпретації, так само, як це роблять фахівці. Основні кроки цього плану включають:

- оцінка серцевого ритму і провідності;
- визначення показника "електрична вісь серця";
- аналіз роботи передсердь за Р-зубцем та PQ інтервалом;
- характеристика показників комплексу QRS-T;
- кардіографічний висновок.

КПЗ-2024

У план аналізу ЕКГ також потрібно включити перевірку правильності реєстрації кардіограми. Це важливо, оскільки контрольний сигнал на початку дослідження, що відображає стандартну напругу в один мілівольт, повинен мати відхилення на 10 мм. Без цієї перевірки запис кардіограми вважається неправильним.

Однак правильна інтерпретація результатів ЕКГ неможлива без знання фізіологічних особливостей людини, таких як вік, стать, статура, зріст та наявність хронічних захворювань. Наприклад, показник "електрична вісь" може відрізнятися у людей з різною фізичною конституцією, і це вважається нормальним. Крім того, для глибокого розуміння кардіограми потрібні знання медичної термінології, яка характеризує ознаки патологій, таких як миготлива аритмія, екстрасистолія та інші.

Висновки

Опис кардіограми - ціле мистецтво! Це дає можливість розуміти, як працює ваше серце та як воно функціонує в нормальних умовах. Навчитися розпізнавати здорову кардіограму — це не лише корисний навичка, але й стимул для здорового способу життя. Збереження здоров'я серця і кардіоваскулярної системи може стати однією з ваших пріоритетних мет заради тривалого та активного життя.

КбПЗ-2024

1.1 Призначення системи

Основною метою даної бакалаврської роботи є розробка та реалізація мікропроцесорної системи для збору, обробки та відображення даних про серцебиття на базі платформи Arduino. Ця система повинна бути доступною, ефективною та легкою у використанні для звичайних користувачів, не маючи медичних знань та навичок. Через використання відкритих технологій і недорогих компонентів, така система може стати в нагоді для широкого кола осіб, що прагнуть контролювати своє здоров'я на регулярній основі.

Для досягнення цієї мети передбачається виконання наступних завдань:

- дослідження існуючих методів та технологій моніторингу серцебиття: Аналіз комерційно доступних систем і вивчення наукових робіт в цій галузі допоможе виявити основні тренди і можливості для покращення;

- проектування мікропроцесорної системи: Розробка схеми з'єднань та архітектури системи, вибір необхідних компонентів та їх інтеграція з платформою Arduino;

- програмування системи: Написання коду для збору, обробки та відображення даних серцебиття. Особлива увага буде приділена оптимізації алгоритмів для забезпечення точності та швидкодії системи;

- тестування та налаштування системи: Проведення випробувань для перевірки надійності, точності та ефективності системи в реальних умовах;

- аналіз даних та висновки: Оцінка результатів тестування для визначення відповідності системи встановленим цілям та рекомендацій для її покращення.

Реалізація цієї системи не тільки дозволить краще контролювати здоров'я, але й сприятиме популяризації персонального здоров'я використовуючи доступні та відкриті технології.

1.2 Область застосування

Серцево-судинні захворювання залишаються однією з головних причин смертності у світі, згідно зі звітами Всесвітньої організації охорони здоров'я. Регулярний моніторинг серцевої діяльності важливий для раннього виявлення ознак цих захворювань, що може значно покращити прогноз і ефективність лікування. Однак, доступ до сучасних медичних технологій часто обмежений високою вартістю та складністю обладнання, що робить їх недоступними для значної частини населення, особливо в країнах з низьким та середнім рівнем доходу.

КБПЗ_2024

У цьому контексті, розробка дешевих, ефективних та легкодоступних технологій для моніторингу серцебиття є дуже актуальною.

Платформа Arduino, яка є основою для цієї бакалаврської роботи, надає можливість створення низьковартісних медичних приладів завдяки своїй відкритості, гнучкості та широкій підтримці спільноти. Це відкриває двері для демократизації медичних технологій, забезпечуючи базу для незалежного моніторингу здоров'я.

Також, актуальність теми підсилюється постійним зростанням інтересу до персонального здоров'я, збереження та самодіагностики серед населення. Люди прагнуть мати більше контролю над своїм здоров'ям і шукають доступні засоби для цього.

Система моніторингу серцебиття на базі Arduino може служити надійним інструментом для щоденного використання, забезпечуючи користувачів важливою інформацією про їхній стан здоров'я.

Впровадження інновацій у сфері медичного моніторингу за допомогою мікропроцесорних систем, які є предметом цієї роботи, має потенціал змінити підходи до здоров'я, зробивши його більш інклюзивним і доступним для широких верств населення. Це створює важливу основу для наукових досліджень та практичних застосувань у сфері здоров'я, що робить дану тему вкрай актуальною та значущою в сучасних умовах.

КОПЗ_2024

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)

Комерційні продукти

Комерційний сектор також не стоїть осторонь. Продукти, як Fitbit або Apple Watch, використовують передові технології для моніторингу серцебиття і активно впроваджують їх у повсякденне життя людей. Вивчення цих рішень допомагає ідентифікувати ключові функціональні можливості та ускладнення, з якими можуть зіткнутися користувачі.

Fitbit Charge 2 та інші моделі Fitbit - ці пристрої широко використовуються для моніторингу фізичної активності, включаючи кількість кроків, витрачені калорії, і, звісно, серцебиття. Однак, дослідження показали, що точність вимірювань серцебиття може знижуватися під час інтенсивних фізичних навантажень (MDPI).

Apple Watch - Вважається одним із найточніших пристроїв для моніторингу серцебиття серед споживчих гаджетів. Згідно з дослідженнями, Apple Watch демонструє високу узгодженість з електрокардіографічними (ЕКГ) вимірюваннями, особливо в спокійному стані (PLOS) (Cardiovascular Diagnosis and Therapy).

Garmin Vivosmart HR та інші пристрої Garmin - ці пристрої також популярні серед спортсменів та людей, що ведуть активний спосіб життя. Вони забезпечують високу точність вимірювань під час бігу і ходьби, але можуть виявитися менш точними при низьких швидкостях або під час інших специфічних умов використання (PLOS).

Ці та інші комерційні пристрої часто інтегрують сучасні технології, такі

КПЗ_2024

як оптичне вимірювання серцебиття, аналіз варіабельності серцебиття (HRV), що може служити важливим індикатором фізіологічного стану користувача. Також вони часто включають можливості для зв'язку з мобільними додатками та хмарними сервісами для збору та аналізу даних (MDPI).

Пропозиції ринку

Частота серцевих ударів та її варіабельність вважаються ключовими індикаторами взаємодії між аутономною нервовою системою та ризиком серцево-судинних захворювань. Аналізуючи ці динамічні та нестабільні параметри, можна точно відстежувати різні фізіологічні процеси, що впливають на ритм серця. Наприклад, аналіз бінарної символіки RR-інтервалів допомагає краще зрозуміти механізми регулювання серцебиття, варіації серцевої частоти можуть служити відображенням контролю за дихальною та кардіосистемами. Часозалежний спектральний аналіз, зокрема вейвлет-перетворення, корисний для виявлення патернів регуляції серцевої частоти, особливо під час реперфузії. Такі аналізи є важливими для виявлення взаємодії між симпатичною та парасимпатичною частинами нервової системи, що може сигналізувати про майбутні серцеві проблеми та раптову зупинку серця. Монітор серцевого ритму (HRM) є особистим приладом для моніторингу, який дозволяє реєструвати або відображати серцеві скорочення у реальному часі або зберігати дані для подальшого аналізу. Він широко застосовується для відстеження серцевої активності під час фізичних навантажень. Електричні сигнали від серця вимірюються за допомогою електрокардіографії (ЕКГ).

У медичних установах HRM зазвичай використовується з кількома датчиками і називається Холтеровим монітором. Повсякденні споживчі монітори серцевого ритму зручні для щоденного використання, оскільки не вимагають проводів. Перший бездротовий монітор серцевого ритму EKG був розроблений компанією Polar Electro у 1977 році як тренувальний пристрій для фінської національної команди з лижних гонок.

З поширенням концепції "тренувань з високою інтенсивністю" у

КОПЗ-2024

спортивних колах в середині 1980-х, у 1983 році стартували роздрібні продажі бездротових особистих моніторів серця. Сучасні пульсометри часто використовують один з двох основних методів для збору даних про серцеву діяльність — електричний та оптичний. Ці методи забезпечують аналогічну базову інформацію про серцевий ритм за допомогою автоматизованих алгоритмів для вимірювання частоти серцевих скорочень. Електрокардіографічні (ЕКГ) датчики фіксують біопотенціали, створювані електричними імпульсами, які стимулюють скорочення та розслаблення серцевих камер, і зазвичай використовуються в медичних пристроях. Натомість, датчики, які працюють на принципі фотоплетизмографії (PPG), застосовують світлові технології для вимірювання змін обсягу крові, що насосується серцем.

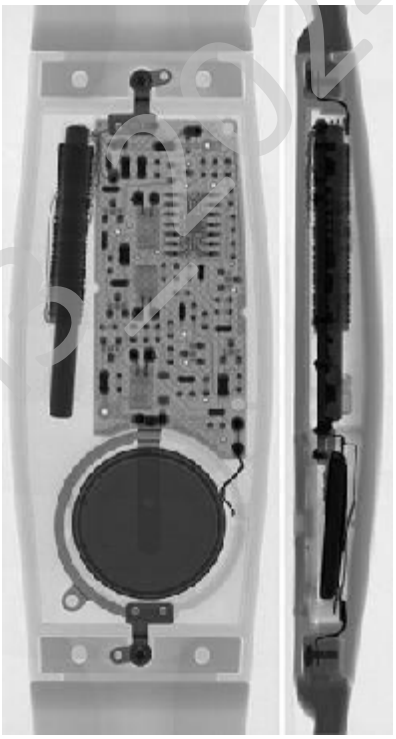


Рисунок 2.1 - Електричний монітор

Електричні монітори серця складаються з двох компонентів: монітора/передавача, що кріпиться до грудної стрічки, та приймача. Під час фіксації серцебиття, передавач відсилає радіосигнал, який приймач використовує для відображення чи визначення поточного ритму серця. Сигнал

КПІ-2024

може бути простим радіоімпульсом або унікальним кодованим сигналом, наприклад, через Bluetooth, ANT, або інший радіосигнал малої потужності. Сучасні технології забезпечують захист від перехресних перешкод або небажаного перехоплення сигналів іншими приймачами поблизу. Зазначимо, що старіша радіопередавальна технологія Polar 5,1 кГц може використовуватися під водою, на відміну від Bluetooth та Ant +, які працюють на частоті 2,4 ГГц і не передають сигнали під водою. Новітні пристрої використовують оптичні технології для вимірювання серцевого ритму, просвічуючи шкіру світлом від світлодіода та аналізуючи його розсіювання через кровоносні судини. Ці технології також можуть вимірювати насичення крові киснем. Сучасні оптичні датчики також здатні передавати зібрані дані. Більшість новітніх гаджетів, таких як смартфони або годинники, використовуються для відображення або збирання інформації про серцевий ритм, насичення киснем, а також інші параметри здоров'я, включаючи швидкість, місцезнаходження та відстань завдяки вбудованим акселерометрам, гіроскопам та GPS. Це зробило такі пристрої, особливо смарт-годинники, дуже популярними на ринку. Останнім часом монітори серцевого ритму на зап'ясті демонструють майже таку ж точність, як і традиційні грудні ремені, з точністю до 95% за даними незалежних тестів, хоча під час інтенсивних фізичних навантажень або під водою можливі помилки вимірювань.

Монітори серцевого ритму, відомі також як пульсометри, є ключовими пристроями для моніторингу частоти серцевих скорочень або пульсу в реальному часі. Частота серцевих скорочень вказує на кількість серцебиттів за одну хвилину, що в середньому становить 72 удари на хвилину для здорової дорослої людини. Ця частота може збільшуватися під час фізичної активності та зменшуватися під час відпочинку. Монітори серцевого ритму є надзвичайно корисними для спортсменів та людей, які регулярно займаються фізичними вправами. Вони дозволяють користувачам визначати інтенсивність тренування та моніторити їхній загальний фізичний стан. Ці пристрої допомагають

КОПЗ-2024

розуміти, чи допомагають їхні поточні тренування досягати поставлених цілей у сфері фізичного здоров'я.

Монітори серцевого ритму зазвичай складаються з двох основних частин: передавача та приймача. Передавач є датчиком, який кріпиться до грудей і зчитує електричні сигнали серця, а потім передає цю інформацію. Цей елемент часто виконаний у вигляді водонепроникного ременя, який дозволяє використовувати його навіть під час плавання або тренувань у вологих умовах. Приймач, зазвичай розташований на зап'ясті, відображає актуальну інформацію про частоту серцевих скорочень, що дозволяє користувачу легко відстежувати свої показники під час активності.

Ці пристрої дозволяють отримати детальні дані про серцевий ритм, що є важливим для ефективного планування та коригування тренувань, вимірювання фізичної витривалості та загального здоров'я.

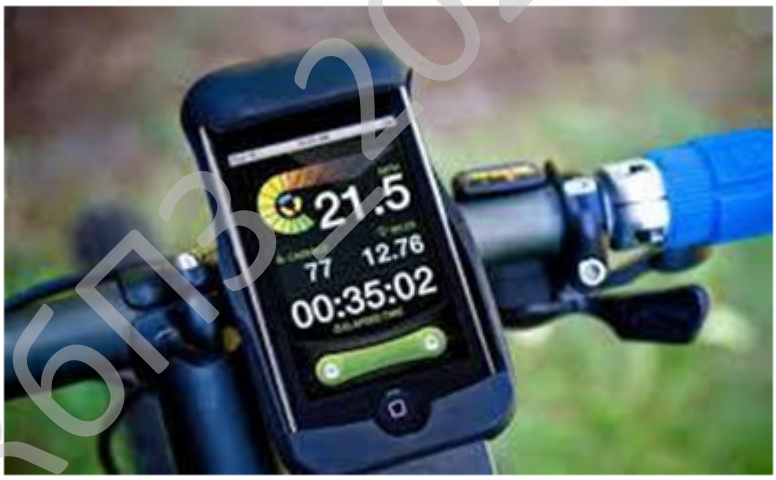


Рисунок 2.2 - Монітор серцевого ритму для велосипедистів

Fitbit Inspire HR — це прекрасний приклад сучасних пристроїв для відстеження здоров'я, які використовують оптичні монітори серцевого ритму (HRM) для надання користувачам глибокого аналізу їхнього фізичного стану та прогресу у фітнесі. Цей пристрій може цілодобово відстежувати частоту серцевих скорочень, забезпечуючи важливу інформацію для осмислення загального здоров’я та рівня фізичної придатності користувача.

КБПЗ_2024

Однією з ключових функцій Fitbit Inspire HR є здатність вимірювати частоту серцевих скорочень у спокої, що є важливим показником загального серцево-судинного здоров’я. Частота серцевих скорочень у спокої може дати зрозуміти, наскільки добре серце працює під час відпочинку, і як воно адаптується до фізичних навантажень.

Fitbit також використовує дані HRM для оцінки Cardio Fitness Score — показника, що вимірює максимальний об'єм споживання кисню (VO2 max). Цей показник допомагає зрозуміти, наскільки добре ваше тіло використовує кисень під час інтенсивних вправ і є одним із головних індикаторів серцево-судинного здоров'я.

Інша важлива функція Fitbit Inspire HR — відстеження сну. Це дозволяє користувачам отримувати деталізовану інформацію про свої сонові цикли, якість сну та загальний час сну, що є критично важливим для відновлення організму та здоров'я.

Також пристрій забезпечує порівняння цих даних із статистикою інших користувачів Fitbit того ж віку, допомагаючи користувачам оцінити свій прогрес у контексті ширшої групи. Це може слугувати мотивацією для поліпшення або підтримки фізичної форми.

Отже, Fitbit Inspire HR — це інноваційний інструмент, який виходить за рамки простого відстеження кроків, надаючи глибокі інсайти про здоров'я та фізичну активність, що робить його ідеальним для щоденного моніторингу та підтримки здоров'я.

Рисунок 2.3 - Fitbit Inspire HR

КБПЗ-2024

Ремінь для грудної клітки Polar H10

Polar H10 є одним із провідних виборів серед серцевих моніторів, який поєднує високу точність з зручністю користування. Цей монітор серцевого ритму пропонує унікальну можливість підключення до двох пристроїв одночасно через Bluetooth, ANT+ та 5 кГц радіосигнали. Це робить його ідеальним для багатьох видів спорту, включаючи плавання, де необхідне під водою зв'язок.

Великий термін служби батареї, що становить 400 годин, забезпечує довготривале використання без необхідності частої заміни батареї, що є великою перевагою для серйозних спортсменів і активних людей, які не хочуть часто займатися обслуговуванням свого обладнання. Заміна акумулятора CR2025 є недорогою та зручною, що є додатковим бонусом.

Щодо комфорту, Polar H10 вважається одним із найзручніших грудних ременів на ринку. Особливі пряжки на ремені підвищують комфорт і зменшують ймовірність подразнення шкіри, яке часто трапляється з іншими грудними ременями. Використання H10 для щоденних пробіжок протягом тривалого часу підтверджує його високий рівень комфорту, оскільки він не викликає подразнення шкіри навіть при тривалому використанні.

Таким чином, якщо ви шукаєте надійний, точний та зручний монітор серцевого ритму, який підходить як для повсякденних активностей, так і для інтенсивних тренувань, Polar H10 може бути чудовим варіантом.

Рисунок 2.4 - Polar H10

КОПЗ_2024

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Принципи обробки даних серцебиття

Принципи обробки даних серцебиття включають кілька ключових аспектів, які дозволяють точно і ефективно аналізувати сигнали серцебиття для медичних і дослідницьких цілей.

Збір даних. Серцебиття зазвичай реєструється за допомогою електрокардіограм (ЕКГ), яка вимірює електричну активність серця через електроди, розміщені на тілі пацієнта.

Попередня обробка. Перед тим як проаналізувати дані, важливо видалити шуми та артефакти, які можуть виникнути внаслідок руху, дихання або інших електричних джерел. Це досягається за допомогою фільтрації сигналів.

Детекція R-піків. Однією з ключових функцій в обробці даних серцебиття є ідентифікація R-піків на ЕКГ, які представляють собою точки найвищої амплітуди в циклі серцебиття. Вони використовуються для визначення частоти серцебиття та інших параметрів здоров'я серця.

Аналіз варіабельності серцевого ритму (HRV). HRV — це міра, що відображає варіації у часі між послідовними серцевими циклами. Вона є важливим індикатором нервової системи та загального стану здоров'я серця.

Фінальний аналіз. Отримані дані можуть бути проаналізовані для отримання інформації про такі показники, як середня частота серцебиття, максимальна та мінімальна частота, а також патерни ритму серця, які можуть вказувати на потенційні проблеми зі здоров'ям.

Ці принципи є основою для створення надійних систем моніторингу серцебиття, які можуть використовуватися як у клінічних, так і в домашніх умовах. Це дозволяє здійснювати своєчасне та точне виявлення можливих серцевих захворювань або забезпечувати постійний моніторинг для людей із вже діагностованими станами.

КПЗ-2024

Давайте детально розглянемо кожен із пунктів, пов'язаних з обробкою даних серцебиття.

Збір даних

Збір даних серцебиття зазвичай виконується за допомогою електрокардіограми (ЕКГ), яка фіксує електричну активність серця. Для цього на тілі пацієнта розміщуються електроди, які вловлюють малі електричні зміни на шкірі, що виникають від електричної активності серця. Ці сигнали відображають роботу різних частин серцевого м'яза.

Попередня обробка

Перед тим як сигнали можна буде аналізувати, вони мають бути очищені від шумів та артефактів. Шуми можуть виникати через рухи пацієнта, електричні інтерференції від інших пристроїв або неправильне підключення електродів. Для фільтрації сигналів використовують різні види фільтрів, такі як низькочастотний фільтр для усунення "дріботи" та високочастотний фільтр для зниження впливу повільних змін напруги.

Детекція R-піків

R-піки на ЕКГ представляють моменти найбільшої амплітуди сигналу, що відповідають вентрикулярній деполяризації серця. Виявлення цих піків критично важливе для аналізу HRV і визначення частоти серцебиття. Існують численні алгоритми для детекції R-піків, включаючи алгоритми на основі порогових значень, алгоритми, що використовують перетворення вейвлетів, та інші методи машинного навчання.

Аналіз варіабельності серцевого ритму (HRV)

HRV — це міра варіації інтервалів між послідовними серцебиттями. Варіабельність може бути аналізована в часовому та частотному діапазонах для оцінки впливу вегетативної нервової системи на серце. Методи аналізу HRV включають розрахунок стандартного відхилення всіх інтервалів RR, аналіз спектральної щільності потужності та багато інших показників.

КПІ – 2024

Фінальний аналіз

Після обробки та аналізу, дані можуть бути використані для визначення загальної картини серцевої активності пацієнта. Остаточний аналіз може включати:

– оцінка середньої частоти серцебиття (HR): Визначення середніх значень HR протягом певного періоду часу для оцінки загальної серцевої активності;

– аналіз екстремальних значень: Ідентифікація максимальних та мінімальних значень HR для визначення потенційних аномалій;

– оцінка ритмів серця: Визначення регулярності або нерегулярності серцевих ударів, що може вказувати на різні серцеві умови, такі як фібриляція передсердь;

Довгостроковий моніторинг. Аналіз даних, зібраних протягом тривалого часу, може допомогти виявити тенденції або зміни у серцевій активності, які можуть вимагати медичного втручання.

Ці дані можуть бути представлені у вигляді графіків, таблиць та інших візуальних форматах, які полегшують їх інтерпретацію медичними фахівцями та пацієнтами. Наприклад, візуалізація серцевого ритму на графіку може швидко показати періоди нерегулярного серцебиття, які можуть потребувати подальшої уваги.

Використання передових аналітичних методів, таких як машинне навчання та штучний інтелект, може далі покращити точність та корисність обробки даних серцебиття, дозволяючи робити більш точні прогнози та рекомендації на основі великих обсягів даних.

КБПВ-2024

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке продемонструє роботу і побудову

датчика серцебиття для щоденного моніторингу здоров’я людини.

В процесі розробки роботи необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран повідомлень про некоректні дії користувача та нестандартні ситуації;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) сформувати висновки про виконаний обсяг робіт та одержані результати.

КбПЗ_2024

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Цифрова частина приладу представлена мікроконтролером Arduino Leonardo.

Рисунок 3.1 - Микрокотролер Arduino Leonardo Mega

Arduino - це багатофункціональний електронний конструктор і платформа для розробки пристроїв, що користуються популярністю як у початківців, так і серед досвідчених спеціалістів. Його зручність полягає у простоті мови програмування та відкритій архітектурі і програмному коді. Arduino програмується через USB, не потребує використання програматорів.

Arduino дозволяє комп'ютеру взаємодіяти з фізичним світом, зчитувати дані від різних датчиків та керувати виконавчими пристроями. Мікроконтролер програмується мовою Arduino, що базується на мові Wiring, у середовищі розробки Arduino, яке базується на середовищі Processing. Проекти на базі Arduino можуть працювати самостійно або взаємодіяти з програмним забезпеченням на комп'ютері.

КПЗ-2024

Плати Arduino можуть бути зібрані користувачем або придбані готовими до використання. Програмне забезпечення доступне для безкоштовного скачування, а вихідні креслення схем є загальнодоступними. У 2006 році Arduino отримала визнання на фестивалі Ars Electronica Prix у категорії Digital Communities.

Апаратна частина платформи Arduino

Існує кілька версій платформи Arduino, кожна з яких базується на різних мікроконтролерах. Ось короткий огляд основних версій:

Arduino Uno і Duemilanove: Ці версії базуються на мікроконтролері Atmel ATmega328. Uno є однією з найпопулярніших версій Arduino.

Arduino Leonardo: Ця версія побудована на мікроконтролері ATmega32u4. Вона має деякі відмінності у функціональності порівняно з Uno, такі як можливість емуляції клавіатури та миші за допомогою USB.

Arduino Mega2560: Ця версія використовує мікроконтролер ATmega2560 і має більше виводів та розширені можливості порівняно з Uno.

Старі версії: Серед старіших версій можна відзначити Diecimila і ранні версії Duemilanove, які використовували мікроконтролери ATmega168 та ATmega8 відповідно.

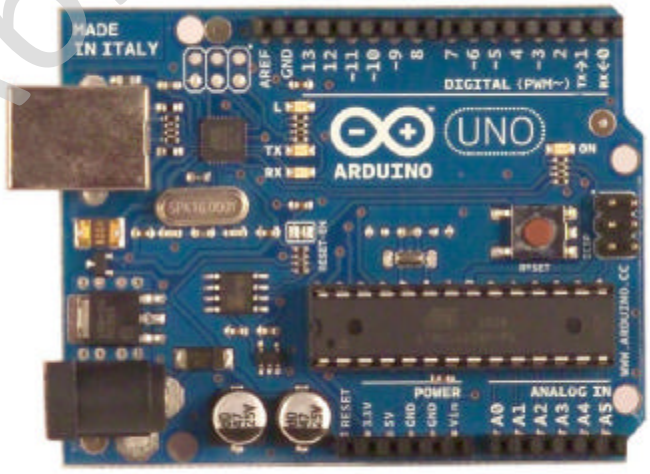


Рисунок 3.2 – Arduino Uno

КПЗ-2024

Версії платформи Arduino

Нижче представлені основні версії плат Arduino:

Due: Це нова плата, яка базується на ARM мікропроцесорі 32-біт Cortex-M3 ARM SAM3U4E. Вона обладнана потужним процесором та має ряд передових можливостей.

Leonardo: Остання версія Arduino, яка використовує мікроконтролер ATmega32u4. Ця версія відрізняється наявністю роз'єму microUSB та компактним розміром, що схоже на UNO.

Yun: Ця нова плата має вбудовану підтримку WiFi і базується на ATmega32u4 та Atheros AR9331. Вона призначена для проектів, які вимагають бездротового зв'язку.

Micro: Це нове компактне рішення, яке також використовує мікроконтролер ATmega32u4.

Uno: Це одна з найпопулярніших версій Arduino, яка має стандартний порт USB та використовує чіп ATMega8U2 для USB-підключення. Uno може бути легко розширена за допомогою різних плат розширення.

Arduino Ethernet: Цей контролер має вбудовану підтримку мережі та можливість живлення через мережу за допомогою модуля POE (Power over Ethernet).

Duemilanove: Це передостання версія Arduino, яка також має стандартний USB-порт і може бути розширена за допомогою різних плат розширення.

Diecimila: Це попередня версія базової платформи Arduino USB. Вона має свої характеристики та можливості, але була попередньою до новіших версій.

Nano: Це компактна платформа, яка часто використовується для прототипування. Nano підключається до комп'ютера за допомогою кабелю USB Mini-B.

Mega ADK: Це версія плати Mega 2560 з підтримкою USB host інтерфейсу, призначена для зв'язку з Android-телефонами та іншими пристроями з USB інтерфейсом.

КПЗ_2024

Mega2560: Це нова версія плати серії Mega, побудована на базі Atmega2560. Вона використовує чіп ATMega8U2 для послідовного з'єднання через USB порт.

Arduino BT: Це платформа з модулем Bluetooth для бездротового зв'язку і програмування, яка сумісна з платами розширення Arduino.

LilyPad: Це платформа, яка розроблена для носіння та може бути зашита в тканину. Вона має пурпурний колір.

Fio: Це платформа, яка призначена для бездротових застосувань. Fio містить роз'єм для радіо XBee, роз'єм для батареї LiPo і вбудовану схему підзарядки.

Mini: Це найменша платформа Arduino, яка працює як макетна модель або в проектах, де місце обмежене. Вона підключається до комп'ютера за допомогою адаптера Mini USB.

Адаптер Mini USB: Це плата, яка конвертує з'єднання USB в лінії 5 В, GND, TX і RX для з'єднання з платформою Arduino Mini або іншими мікроконтролерами.

Pro: Це платформа, розроблена для досвідчених користувачів і може бути частиною більшого проекту. Вона дешевша, ніж Diecimila, але вимагає додаткової збірки і компонентів.

Pro Mini: Ця платформа, подібно до версії Pro, розроблена для досвідчених користувачів, яким потрібна низька ціна, компактні розміри та додаткова функціональність.

Serial: Це базова платформа з інтерфейсом RS232 для зв'язку та програмування. Плата легко збирається, навіть початківцями користувачами, і містить схеми і файли CAD.

Serial Single Sided: Ця платформа розроблена для ручної збірки. Вона трохи більшого розміру, ніж Diecimila, але сумісна з платами розширення Arduino.

USB Serial Light Adapter: Це адаптер, який дозволяє підключати плати

КОПЗ_2024

Arduino до комп'ютера для обміну даними та завантаження скетчів. Він зручний для програмування таких плат, як Arduino Mini, Arduino Ethernet та інших, які не мають власного USB-роз'єму.

Ці версії платформи Arduino також мають свої унікальні характеристики та властивості, які відповідають різним потребам користувачів.

Плати розширення

Ось декілька розширень для платформи Arduino, які розширюють її можливості:

– плата розширення WiFi: Вона призначена для підключення до бездротових мереж стандарту 802.11 b/g, щоб Arduino міг взаємодіяти та обмінюватися даними з іншими пристроями через Wi-Fi;

– плата розширення Xbee Shield: Це дає можливість для бездротового зв'язку між кількома пристроями Arduino за допомогою модуля Maxstream Xbee Zigbee. Зв'язок може бути встановлений на відстані до 35 метрів всередині будівель та до 90 метрів на відкритому повітрі;

– плата розширення Motor Shield: Ця плата дозволяє керувати двигунами постійного струму та отримувати дані від датчиків положення. Вона дозволяє створювати проекти, що включають рухомі механізми;

– плата розширення Ethernet Shield: Це дозволяє підключати Arduino до Інтернету через Ethernet, що розширює можливості пристрою у зборі та обробці даних, а також взаємодії з іншими пристроями та веб-серверами.

Arduino Leonardo

Рисунок 3.3 – Arduino Leonardo

КОПІЯ 2024

Загальні відомості

Arduino Leonardo - це мікроконтролер, що базується на чіпі ATmega32u4, і він відзначається рядом можливостей, що роблять його важливим і універсальним для електронних проектів. Основні особливості Arduino Leonardo включають:

Широкий набір вхідно-вихідних портів: У Leonardo є 20 цифрових вхідно-вихідних портів, із них 7 можуть працювати як виходи ШІМ (широтно-імпульсна модуляція), що дозволяє регулювати яскравість світлодіодів, швидкість обертання моторів і т. д., і 12 портів можуть використовуватися як аналогові входи для зчитування аналогових датчиків.

Мікро-USB роз'єм: Для зручності підключення до комп'ютера або інших пристроїв Leonardo використовує мікро-USB роз'єм.

Силовий роз'єм і кнопка перезавантаження: У Leonardo є силовий роз'єм для живлення від адаптера AC/DC або батареї, а також кнопка перезавантаження для швидкого перезапуску програми або пристрою.

Вбудована підтримка USB з'єднання: Однією з особливостей Leonardo є вбудована підтримка USB з'єднання, що дозволяє програмувати його для взаємодії з комп'ютером як клавіатура, миша або віртуальний послідовний/COM порт, відкриваючи широкі можливості для створення різноманітних проектів.

Завдяки цим характеристикам Arduino Leonardo стає важливим інструментом для розробників, які шукають потужний, гнучкий і легко програмований контролер для своїх проектів.

КОПЗ-2024

Характеристики

Таблиця 3.1 – Характеристики Arduino Leonardo

Мікроконтролер	ATmega32u4
Робоча напруга	Робоча напруга
Вхідна напруга (рекомендований)	7-12 В
Вхідна напруга (граничне)	6-20 В
Цифрові Входи / Виходи	20 (7 з яких можуть використовуватися як виходи ШІМ)
Аналогові канали	12
Постійний струм через вхід / вихід	40 мА
Постійний струм для виведення 3.3	50 мА
Флеш-пам'ять	32 кБ (ATmega32u4) з яких 4 КБ
ОЗУ	2 кБ (ATmega32u4)
Тактова частота	16 МГц

Живлення

Arduino Leonardo може живитися через підключення до USB або зовнішнього джерела живлення. У випадку, якщо обидва джерела підключені, пріоритет надається зовнішньому живленню.

Зовнішнє живлення може бути забезпечене через блок живлення AC/DC або акумуляторну батарею. Для цього перетворювач напруги підключається за допомогою роз'єму з центральним позитивним полюсом діаметром 2.1 мм. Проводи від батареї підключаються до висновків Gnd і Vin роз'єму живлення.

Платформа може працювати при зовнішньому живленні від 6 В до 20 В. Проте, напруга живлення нижче 7 В може призвести до зниження напруги на висновку 5V, що може призвести до нестабільної роботи. При використанні напруги вище 12 В, регулятор напруги може перегрітися і пошкодити плату. Рекомендований діапазон живлення - від 7 В до 12 В.

КПЗ_2024

Виводи живлення

На платі Arduino Leonardo є кілька важливих виводів:

VIN: Цей вивід використовується для подачі живлення від зовнішнього джерела,оли немає напруги 5 В від роз'єму USB або іншого джерела. Живлення подається через цей вивід.

5V: Цей вивід є регульованим джерелом напруги, яке живить мікроконтролер і компоненти на платі. Він може отримувати живлення з виводу VIN через регулятор напруги, з роз'єму USB або з іншого регульованого джерела напруги 5 В.

3V3: На цьому виводі знаходиться напруга 3.3 В, яка генерується вбудованим регулятором на платі. Максимальне споживання струму з цього виводу - 50 мА.

GND: Це вивід заземлення, який використовується для з'єднання землі.

IOREF: Цей вивід показує робочу напругу вхідно-вихідних виводів плати. Для Arduino Leonardo це 5 В. Він призначений для використання платами розширення для правильного вибору робочої напруги.

Входи і Виходи

Кожен з 20 цифрових виводів на Arduino Leonardo може бути налаштований як вхід або вихід за допомогою функцій pinMode(), digitalWrite() і digitalRead(). Виводи працюють при напрузі 5 В та мають навантажувальний резистор (за замовчуванням відключений) 20-50 кОм і можуть пропускати до 40 мА.

Деякі виводи мають особливі функції:

Послідовна шина: 0 (RX) і 1 (TX). Вони використовуються для отримання (RX) і передачі (TX) даних TTL. Ці виводи підключені до відповідних висновків мікросхеми послідовної шини ATmega32U4 USB-to-TTL. Зверніть увагу, що у Leonardo, клас Serial відноситься до послідовного з'єднання USB CDC. Послідовне з'єднання через виводи 0 і 1 здійснюється через клас Serial1.

TWI: 2 (SDA) і 3 (SCL). Ці виводи використовуються для зв'язку I2C

КОПЗ_2024

(TWI), для чого використовується бібліотека Wire.

Зовнішнє переривання: 2 і 3. Ці виводи можуть бути налаштовані на виклик переривання або на молодшому значенні, або на передньому чи задньому фронті, або при зміні значення. Детальна інформація знаходиться в описі функції attachInterrupt().

ШІМ: 3, 5, 6, 9, 10, 11 і 13. Будь-який з виводів забезпечує ШІМ з роздільною здатністю 8 біт за допомогою функції analogWrite().

SPI: на роз'ємі ICSP. Ці виводи використовуються для зв'язку SPI, для чого використовується бібліотека SPI. Зверніть увагу, що в Leonardo виводи SPI не розлучені на цифрові вхід/вихід, як це було в попередніх версіях Arduino контролерів.

LED: 13. Це вбудований світлодіод, підключений до цифрового виводу 13. Якщо значення на виводі має високий рівень, то світлодіод світиться.

Аналогові входи: A0-A5, A6-A11 (на цифрових виводах 4, 6, 8, 9, 10 і 12). Leonardo має 12 аналогових входів, позначених від A0 до A11. Всі аналогові входи можуть працювати в режимі цифрових вхід/вихід. Вхідні виводи з A0 по A5 збігаються з аналоговими входами UNO. Виводи з A6 по A11 на цифрових виводах 4, 6, 8, 9, 10 і 12 відповідно. Розширення аналогових входів - 10 біт, тобто 1024 різних значень. За замовчуванням значення на аналогових входах вимірюються від землі (0) до 5 Вольт, верхня межа діапазону може бути змінена за допомогою входу AREF і функції analogReference().

Додаткові виходи платформи Arduino Leonardo включають:

AREF: Це опорна напруга для аналогових входів. Використовується з функцією analogReference() для встановлення внутрішнього джерела опорної напруги для аналогових перетворювачів;

Reset: Низький рівень сигналу на виводі Reset перезавантажує мікроконтролер. Зазвичай використовується для підключення кнопки перезавантаження на платі розширення, що закриває доступ до кнопки на самій платі Arduino. Це дає можливість перезавантажувати мікроконтролер в разі

КБПЗ-2024

необхідності.

Зв'язок

На платформі Arduino Leonardo зв'язок можна встановити з комп'ютером, іншими пристроями Arduino або мікроконтролерами кількома способами. ATmega32U4 підтримує послідовний інтерфейс UART TTL (5 В), який реалізується за допомогою виводів 0 (RX) і 1 (TX). Крім того, ATmega32U4 може організувати послідовне з'єднання з програмами на комп'ютері через USB, щоб вони взаємодіяли з платою через віртуальний COM порт. Arduino Leonardo може підключатися як USB 2.0 пристрій за допомогою стандартних драйверів USB COM, що дає можливість моніторити послідовну шину (Serial Monitor) в середовищі розробки Arduino для надсилання та отримання текстових даних. Світлодіоди RX і TX на платформі мигатимуть при передачі даних через USB підключення, але не під час використання послідовної передачі через виводи 0 і 1. Для створення послідовної передачі даних через будь-який з цифрових виводів Leonardo можна використовувати бібліотеку SoftwareSerial.

ATmega32U4 також підтримує інтерфейси I2C (TWI) і SPI. Для зручного використання шини I2C у Arduino включена бібліотека Wire, а для SPI можна використовувати бібліотеку SPI.

Arduino Leonardo може розпізнаватися при підключенні до комп'ютера як пристрій миші або клавіатури. Управління цим режимом здійснюється за допомогою класів Keyboard і Mouse.

Програмування

Платформу програмують за допомогою програмного забезпечення Arduino. При виборі "Arduino Leonardo" з меню Tools > Board обирається відповідна конфігурація для мікроконтролера, який встановлено на платформі. Для детальної інформації можна звернутися до довідника та інструкцій.

Мікроконтролер ATmega32U4 на платформі Leonardo вже має записаний загрузчик, що спрощує завантаження нових програм без необхідності використання зовнішніх програматорів. Зв'язок здійснюється за допомогою

КПІ – 2024

протоколу AVR109.

Також є можливість програмувати мікроконтролер через виводи ICSP (внутрішнє програмування), при цьому не використовуючи загрузчик. Більш докладна інформація щодо цього наведена у відповідних інструкціях.

Автоматичне (програмне) перезавантаження

Платформа Leonardo розроблена з урахуванням автоматичного перезавантаження перед записом нового коду, що здійснюється програмою Arduino на комп'ютері, а не потребує натискання кнопки на самій платформі. Цей процес спрацьовує, коли віртуальний CDC COM порт відкривається зі швидкістю 1200 бод, а потім закривається. Після цього мікропроцесор автоматично переходить на перезавантаження, розриваючи USB з'єднання. Після перезавантаження стартує завантажувач (бутлодер), який залишається активним протягом приблизно 8 секунд. Крім того, завантажувач також можна ініціювати натисканням кнопки Reset. Важливо зауважити, що при подачі живлення контролер відразу переходить до виконання завантаженої користувальницької програми без виконання завантажувача.

Струмовий захист роз'єму USB

У Leonardo також вбудований запобіжник (автомат), який захищає порт USB комп'ютера від струмів короткого замикання і надструмів. Цей запобіжник надає додатковий бар'єр, хоча практично всі комп'ютери мають подібний захист. Він спрацьовує при проходженні струму більше 500 мА через USB порт і розмикатиме ланцюг до тих пір, поки нормальні значення струму не будуть відновлені.

Фізичні характеристики

Плата Leonardo має розміри 6.9 на 5.3 см. Роз'єми USB і силовим роз'єм виступають за межі цих розмірів. Чотири отвори в платі дозволяють закріпити її на поверхні. Відстань між цифровими висновками 7 і 8 становить 0,4 см, тоді як між іншими висновками вона складає 0,25 см.

КПЗ_2024

Вибір дисплея

SSD1306 є потужним драйвером для OLED/PLED дисплеїв, що надає розробникам гнучкість і ефективність у реалізації графічних відображень. Його основні особливості дозволяють використовувати його у широкому спектрі приладів завдяки високій розширеності та інтеграції з різними мікроконтролерами.

SSD1306, драйвер для OLED/PLED дисплеїв, включає підтримку 128 сегментів і 64 спільних лінії, створюючи матрицю 128x64 пікселів. Цей чіп ідеально підходить для використання в мобільних телефонах, MP3-плеєрах, калькуляторах та інших портативних пристроях завдяки своїм невеликим розмірам та низькому енергоспоживанню.

Основні особливості SSD1306 включають.

Висока роздільна здатність: Дисплей має роздільну здатність 128 на 64 пікселі, що забезпечує чітке та деталізоване зображення.

Енергоефективність: Чіп має дві окремі лінії живлення: VDD для логічних цепей від 1,65 до 3,3 В та VCC для керування панеллю від 7 до 15 В, що дозволяє ефективно управляти енергоспоживанням.

Підтримка різних інтерфейсів: SSD1306 може взаємодіяти з головними мікроконтролерами через 8-бітний паралельний інтерфейс 6800/8080 або через 3/4 провідний послідовний периферійний інтерфейс, забезпечуючи велику гнучкість у підключенні.

Управління яскравістю: Інтегрований 256-ступеневий контроль яскравості дозволяє точно регулювати контрастність дисплею, адаптуючи його до різних умов освітлення.

Безпека використання: Максимальний струм сегмента становить лише 100 мкА, а загальний максимальний струм панелі — 15 мА, що робить його безпечним для тривалого використання.

Використання SSD1306 дозволяє розробникам створювати компактні, високоефективні графічні дисплеї з високим рівнем інтеграції та низьким

КОПЗ-2024

споживанням енергії.

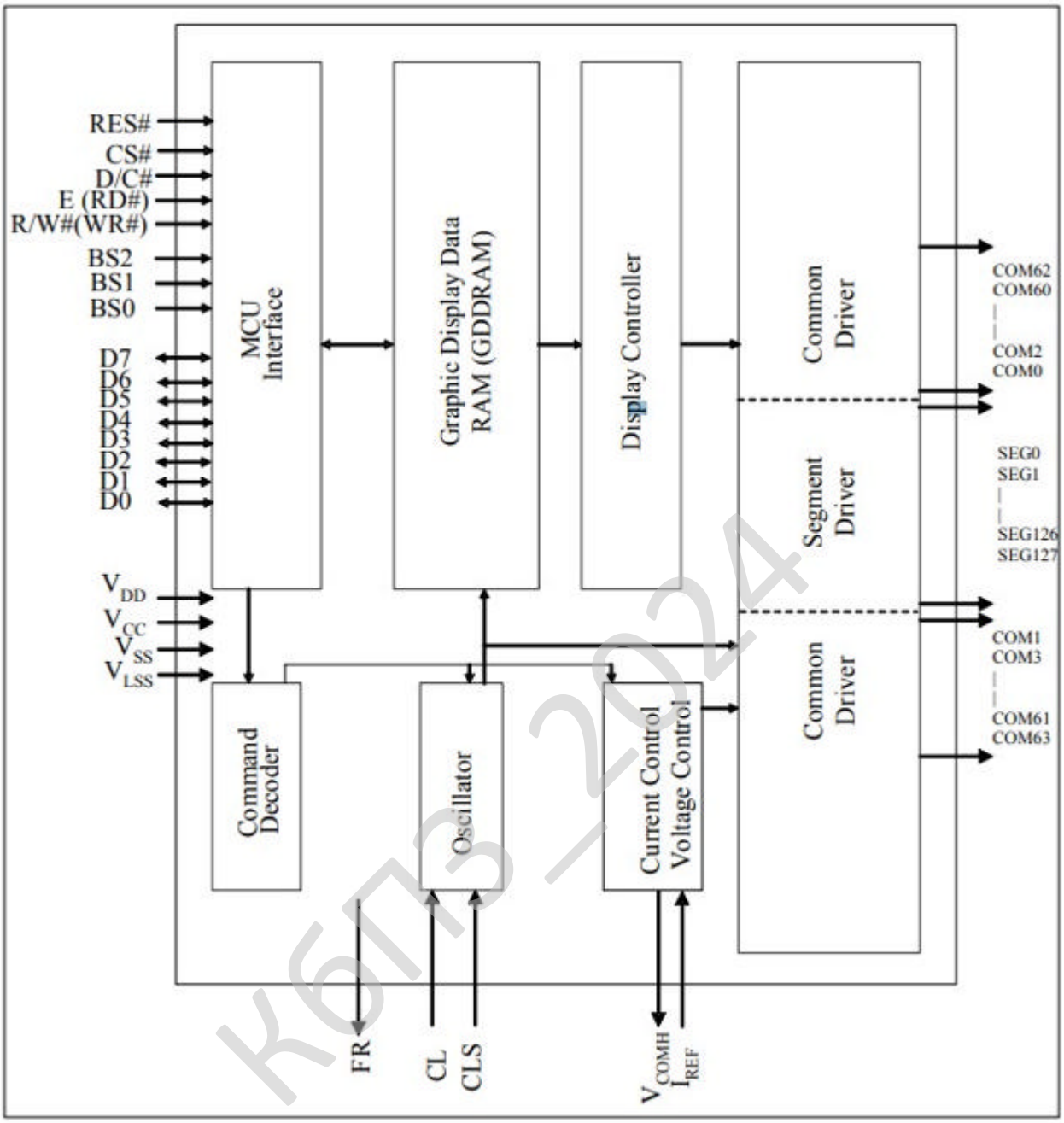


Рисунок 3.4 - Блок-схема SSD1306

Pulse Sensor – це високофункціональний датчик серцебиття, призначений для інтеграції з Arduino, який знайшов своє застосування в освітніх, спортивних, розважальних та розробницьких проектах. Він легко кріпиться на кінчик пальця або мочку вуха і під'єднується до Arduino через гнучкі перемички, що не вимагають пайки, забезпечуючи зручність використання і швидке налаштування.

КБПЗ 2024

Комплектація датчика включає:

- довгий 24-дюймовий кабель зі з'єднувачами, готовими до використання;
- затискач для вуха, який легко кріпиться до датчика і забезпечує стійкість на вуху;
- дві стрічки на липучках, які можна використовувати для зручного кріплення датчика;
- велкро-ремінець для надійної фіксації датчика на пальці;
- три захисні прозорі наклейки, які захищають поверхню датчика від поту та жирних слідів.

Дизайн передньої панелі датчика включає логотип у формі серця, під яким розташований світлодіод, що просвічує через шкіру, та фотодіод для зчитування відбитого світла, що дозволяє визначати серцебиття. Задня частина містить необхідні електронні компоненти, що підсилюють сигнал і передають дані до Arduino. Система забезпечує точні та надійні показники, ідеально підходячи для проектів, де важливий контроль серцевого ритму.

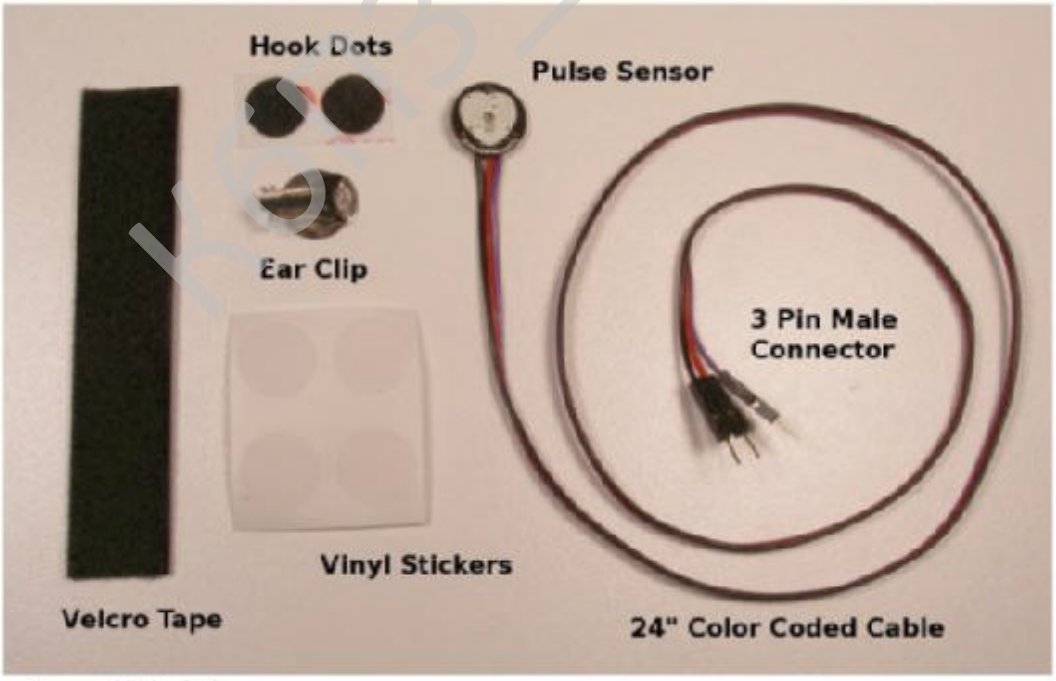


Рисунок 3.5 - Компоненти датчика пульсу Pulse Sensor

КБІЗ 2024

На передній частині датчика пульсу, де розміщено логотип у формі серця, відбувається безпосередній контакт зі шкірою. На цій же стороні є невеликий круглий отвір для світлодіода, який променює світло через шкіру, та невеликий квадратний датчик, розташований під світлодіодом. Цей квадратний датчик слугує для зчитування рівня освітленості, схожий на ті, що використовуються в мобільних телефонах, планшетах і ноутбуках для адаптації яскравості екрану в залежності від умов освітлення. Світлодіод освітлює тканини, такі як кінчик пальця або мочка вуха, де капіляри відображають світло назад до датчика. Всі інші компоненти розташовані на задній панелі датчика, включаючи спеціальний світлодіод зі зворотним кріпленням.

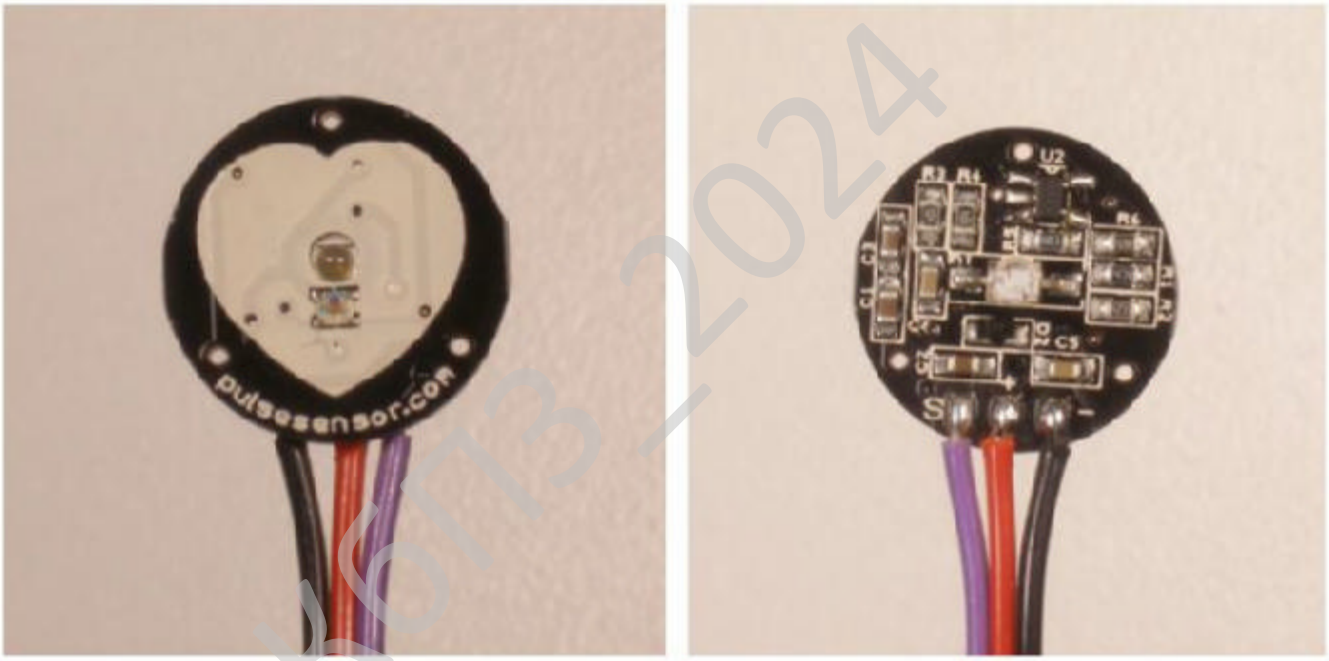


Рисунок 3.6 - Вигляд датчика з обох сторін

3.2 Розробка структурної схеми

Структурна схема надає інформацію про взаємодію майбутніх частин программи, за якій функціонал вони будуть відповідати, та як будуть взаємодіяти між собою. Не можна сказати що вони є інформативними, оскільки на них не можна відстежити як дані передаються та змінюються. Тому в залежності від технічного завдання, структурні схеми розробляють для окремих частин

программи.

Для розробки структурної схеми зазвичай використовують метод покрокової деталізації, щоб вказати всі компоненти з яких складається схема, а також набори підпрограм та підключені бібліотеки.

Структурними компонентами програми можуть називатися підсистеми, бази даних, бібліотеки і т.д. А за допомогою зв'язків можна продемонструвати як вони взаємодіють між собою, обмінюються інформацією, підключати нові бібліотеки. При розробці структурної схеми програмного забезпечення були розглянуті основні модулі програми, та їх взаємодія.

Дуже важливо під час розробки, особливо якщо проект великий і має складну архітектуру, поділити програму на структури, щоб відображати логіку програми, показати місця де підключаються бібліотеки, та за що вони відповідають.

В наведеній схемі, взаємодія модулів, відображена в схемі ієрархії. Де до головного модуля підключаються розроблені модулі, однак схема не відображає порядок функціонування системи. Схема доповнюється розшифрову функцій які виконують підключення модулів.

Ця система обробки даних серцебиття на платформі Arduino складається з декількох ключових компонентів, що інтегровані для збору, обробки та відображення інформації про серцеву активність користувача.

Збір даних. Датчик серцебиття збирає аналогові сигнали, які представляють електричну активність серця або пульсові хвилі крові. Ці сигнали передаються до Arduino.

Попередня обробка сигналів. Сигнали, які надходять з датчика, можуть містити шум або бути слабкими. Використовуючи аналогові фільтри та підсилювачі, система фільтрує та підсилює сигнал перед його цифровою обробкою.

Аналого-цифрове перетворення. Arduino перетворює аналогові сигнали від датчика в цифрові дані для обробки за допомогою вбудованого АЦП (аналого-

КПЗ_2024

цифрового перетворювача).

Цифрова обробка даних. Мікроконтролер обробляє цифрові дані, виявляючи R-піки, розраховуючи інтервали між ними для аналізу HRV, та виконуючи інші необхідні аналітичні функції.

Відображення та передача даних. Оброблені дані можуть бути відображені на LCD дисплеї або передані бездротово через Bluetooth на мобільний додаток для подальшого моніторингу або аналізу.

Збереження даних. Для довгострокового моніторингу або детального аналізу дані можуть зберігатися на зовнішньому носії, як-от SD картці або в EEPROM.

Функціональні можливості

Система здатна забезпечити реальний час моніторингу серцебиття, надаючи користувачеві актуальну інформацію про його серцевий ритм та інші життєво важливі показники. Це може включати сповіщення про аномалії або потенційні здоров'я загрожуючі умови. Система також може бути налаштована для специфічних потреб користувача, наприклад, для спортсменів або осіб з певними медичними станами.

У структурній схемі (рисунок 3.7) позначені зовнішні специфікації програми, які дають розуміння функціональній частині програми, за що вони відповідають, та як вони взаємодіють з вхідними та вихідними даними.

КПЗ_2024

Рисунок 3.7 - Структурна схема

КбГЗ_2024

3.3 Розробка функціональної схеми

Функціональна схема пристрою для вимірювання сердечного ритму зазвичай складається з кількох основних компонентів:

Датчик серцебиття. Зазвичай використовують електрокардіографічні (ЕКГ) датчики або оптичні датчики (PPG). ЕКГ датчики реєструють електричну активність серця через електроди, які контактують зі шкірою. Оптичні датчики працюють за принципом фотоплетизмографії, вимірюючи зміни об'єму крові через світлові хвилі, що відбиваються від шкіри.

Підсилювач сигналу. Цей компонент підсилює слабкий сигнал, отриманий від датчика, до рівня, який можна ефективно обробляти. Для ЕКГ сигналів це зокрема важливо, оскільки сигнали мають низьку амплітуду.

Фільтр сигналу. Фільтруються шуми та інші небажані сигнали, щоб

поліпшити якість даних. Це можуть бути фільтри низьких частот для усунення шуму від руху, високих частот для усунення інтерференції з електричним обладнанням тощо.

Аналого-цифровий перетворювач (АЦП). Перетворює аналоговий сигнал від датчика та підсилювача у цифрову форму, що необхідно для подальшої обробки мікроконтролером.

Мікроконтролер або мікропроцесор. Обробляє цифрові дані для визначення характеристик серцебиття, таких як частота і ритм. Може також використовувати алгоритми для виявлення аномалій у серцевій діяльності.

Інтерфейс користувача. Може включати дисплей для відображення інформації про серцебиття або інтерфейс для підключення до комп'ютера чи смартфона для детальнішого аналізу.

Джерело живлення. Батареї або аккумулятори, які забезпечують енергією всі компоненти пристрою.

Ця схема дозволяє не тільки вимірювати сердечний ритм, але й аналізувати різні аспекти серцевої діяльності, що може бути корисним для медичного моніторингу та діагностики.

КбПЗ – 2024

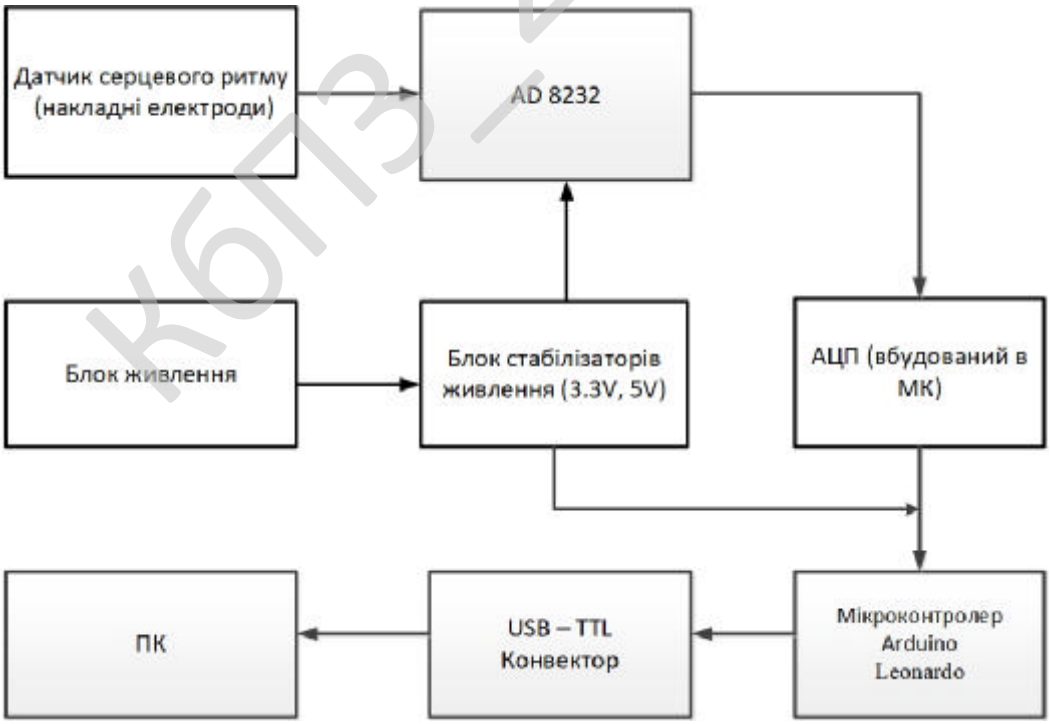


Рисунок 3.8 - Функціональна схема системи

3.4 Розробка діаграми процесів

Є різні методи побудови діаграм, які допомагають відстежити логіку та послідовність дії в процесах, але найкраще себе зарекомендував метод покрокового алгоритму побудови діаграм.

З усіх вище перелічених методів, та технічних рішень при проектуванні, та розробки програмного забезпечення, дивлячись на результати, можна зробити висновки, що я обрав оптимальні проектні рішення, які задовольняють головні потреби, які були поставлені переді мною на початку проекту. Оскільки з результатів можна побачити, що програма виконує всі поставленні задачі.

Для створення діаграми взаємодії процесів пристрою вимірювання сердечного ритму, ми можемо поділити процеси на основні кроки, кожен з яких відповідає певному компоненту або дії. Далі я опишу кожен з цих елементів та їх взаємодію.

Компоненти і процеси

Датчик серцебиття

Задача: збір фізіологічних даних, таких як електричні сигнали серця (ЕКГ) або зміни у кровообігу (PPG).

Взаємодія: передає аналогові сигнали до підсилювача сигналу.

Підсилювач сигналу

Задача: підсилення аналогових сигналів від датчика до рівнів, придатних для подальшої обробки.

Взаємодія: передає підсилені сигнали до фільтра сигналу.

Фільтр сигналу

Задача: фільтрація шумів і небажаних частот з сигналу, для поліпшення якості даних.

Взаємодія: передає очищений аналоговий сигнал до АЦП.

Аналого-цифровий перетворювач (АЦП)

Задача: перетворення аналогового сигналу в цифровий формат для обробки мікроконтролером.

КОПЗ – 2024

Взаємодія: передає цифрові дані до мікроконтролера.

Мікроконтролер

Задача: обробка цифрових даних для визначення характеристик серцебиття та інших параметрів.

Взаємодія: може відправляти інформацію на інтерфейс користувача або зовнішні пристрої.

Інтерфейс користувача

Задача: відображення інформації про серцебиття та інші виміряні параметри.

Взаємодія: отримує дані від мікроконтролера і показує їх користувачу.

Джерело живлення

Задача: забезпечення енергією всіх компонентів системи.

На діаграмі взаємодії кожен компонент з'єднаний стрілками, що показують напрямок потоку даних. Від датчика серцебиття сигнали спочатку йдуть до підсилювача, далі через фільтр до АЦП, який передає цифрові дані мікроконтролеру. Мікроконтролер обробляє отримані дані, визначає параметри серцебиття та може виконувати додатковий аналіз для виявлення можливих аномалій. Після обробки, мікроконтролер передає відповідну інформацію на інтерфейс користувача, де вона відображається для зручності спостереження і аналізу.

Інтерфейс користувача може включати дисплеї, індикатори, або інші засоби зв'язку (наприклад, Bluetooth або Wi-Fi модулі), які дозволяють передавати дані на зовнішні пристрої, такі як смартфони або комп'ютери, для додаткового зберігання та аналізу. Це розширює можливості моніторингу і використання даних у медичних або фітнес-додатках.

Джерело живлення забезпечує стабільну роботу всіх компонентів, і його вибір залежить від потреб у мобільності та тривалості автономної роботи пристрою. Використання батареї або акумуляторів дозволяє зробити пристрій портативним і зручним для щоденного носіння.

КОПЗ-2024

Застосування діаграми

Така діаграма взаємодії допомагає інженерам та розробникам зрозуміти, як компоненти працюють разом, ідентифікувати потенційні слабкі місця в проекті та оптимізувати загальну ефективність та надійність пристрою. Вона також корисна для налагодження та тестування системи, оскільки дає змогу перевірити кожен етап обробки даних і взаємодію між компонентами.

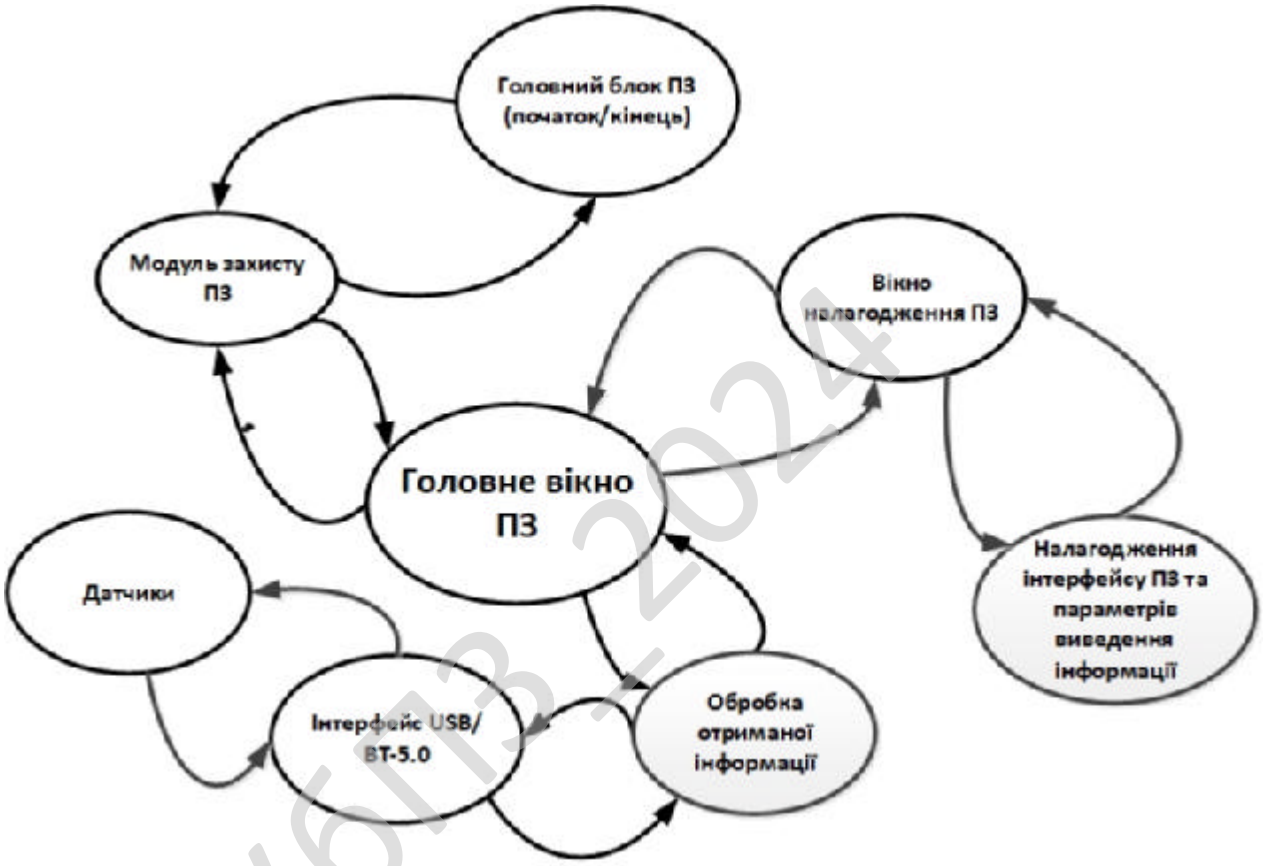


Рисунок 3.9 – Діаграма взаємодії процесів

КПІ – 2024

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

Розробка програми для мікроконтролеру

Для того щоб отримати графік серцевого ритму нам потрібно побудувати свій осцилограф. Роль вимірювального та оцифровуючого приладу буде грати мікроконтролер Arduino. Для того щоб отримати гарний графік нам треба робити 1000 замірів у секунду, що дорівнює 1кГц.

Arduino виконує операцію зчитування за 100 мікросекунд, тобто максимально можливим є 10 000 вимірювань у секунду, що повністю задовольняє нашу потребу. Також дуже важливо робити заміри з однаковим періодом. Для цього ми будеме використовувати внутрішню пам’ять Arduino як буфер. Arduino Leonardo має 2кБ ОЗУ. Це дозволяє нам зберігати 1000 значень типу Integer, займаючи 80% доступної пам’яті. Таким чином ми будемо передавати дані з буфера раз у секунду.

Чому не передавати дані відразу після вимірювання ? Передача даних через USB порт потребує деякого часу на відкриття порту та з’єднання. Таким чином дані надходитимуть повільно та з різною частотою. А у буфер дані надходять с однаковою переодичністю, яка обмежена лише апаратною частиною контролеру. За ту секунду, яку ми будеме зчитувати дані, перший пакет даних піде на обробку на ПК.

Для виконання цього функціоналу був розроблений наступний простий код :

```
int values[1000]; int count = 0; void setup() {
Serial.begin(9600);
}
void loop() {
```

КПЗ_2024

```
if(count < 1000){
values[count] = analogRead(1); count++;
delay(1);
}
else{
for(int i = 0; i < count; i++){ Serial.print(values[i]);
}
count = 0;
}
}
```

Тепер детальніше по важливим функціям

void setup() – блок коду, який виконується одноразово

void loop() – блок коду, який виконується у безкінечному циклі Serial.begin(9600) – встановлює швидкість передачі даних послідовно порту. Може приймати такі значення: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200,

28800, 38400, 57600, 115200

Serial.print() – представляє дані у ASCII кодуванні та посилає у послідовний порт як послідовність байтів

Підсилювач сигналу

Перейдемо до аналогової частини схеми для отримання графіку серцевого ритму. Якщо у вас є схема чи діаграма, я можу допомогти з її аналізом та розглядом.

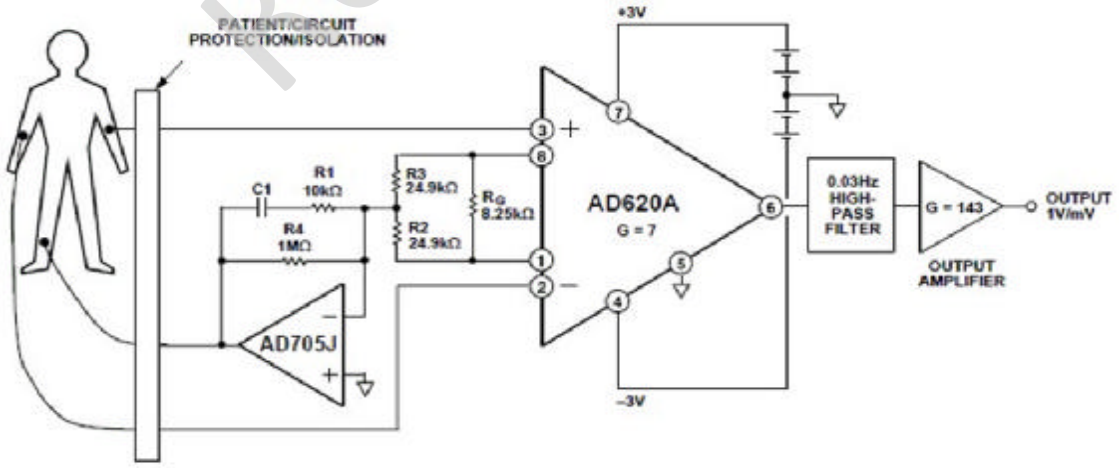


Рисунок 4.1 – Схема з даташиту AD620

КБПЗ_2024

Це дало розуміння того як він працює. Далі я розробив свою схему.

Реєстрація різниці потенціалів на тілі людини, що утворюються між двома точками при роботі серцевого м'яза, дійсно є важливою для медичних досліджень та діагностики. Цей сигнал, вимірюваний за допомогою електродів, може надати інформацію про роботу серця, відображаючи його електричну активність. Як ви вже зазначили, найбільший рівень цього сигналу можливо отримати між внутрішньою стороною зап'ястя лівої і правої руки або з внутрішньої сторони ліктьових згинів рук, де розташовані великі артерії, що є провідниками мікрострумів серцевого м'яза. Шар шкіри фактично ізолює ці артерії від зовнішнього середовища, що робить їх ідеальними точками для отримання сигналу серцевої активності.

Точно так, електрокардіографія (ЕКГ) є важливим інструментом для оцінки різних аспектів функціонування серця. Деякі з основних характеристик, які можна оцінити за допомогою ЕКГ, включають:

Ритм серця. ЕКГ дозволяє визначити ритмічність скорочень серця. За допомогою аналізу інтервалів між QRS комплексами можна виявити різні ритмічні порушення, такі як тахікардія (прискорене серцебиття) або брадикардія (сповільнене серцебиття).

Частота серцебиття. ЕКГ дозволяє точно виміряти частоту серцевих скорочень, що є важливим параметром для оцінки функції серця.

Аритмії. Аналіз ЕКГ може виявити різні аритмії, такі як фібриляція передсердь, фібриляція шлуночків, екстрасистолія та інші аномалії серцевого ритму.

Стан серцевих структур. Характеристики форми та розмірів QRS комплексів можуть допомогти в оцінці стану різних частин серця, таких як передсердя та шлуночки.

ЕКГ є необхідним інструментом для виявлення, моніторингу та лікування різних серцевих захворювань, і використовується як у клінічній медицині, так і в наукових дослідженнях.

КПЗ-2024

Так, інструментальні підсилювачі дійсно широко використовуються для реєстрації різниці потенціалів між двома сигналами в електрофізіологічних дослідженнях, таких як ЕКГ.

У схемі, яку ви описали, використовуються три операційних підсилювача. Перший операційний підсилювач (DA1) та другий (DA2) можуть використовуватися для збільшення вхідного опору підсилювача та для налаштування коефіцієнта посилення. Однак, основною функцією третього операційного підсилювача (DA3) є реалізація диференційного підсилення.

У цій схемі операційний підсилювач DA3 разом з резисторами R1, R2, R3 та R4 створює диференційний підсилювач, який вимірює різницю потенціалів між двома входами. Ця різниця потенціалів може бути використана для реєстрації біопотенціалів, таких як ЕКГ.

Важливим аспектом цієї схеми є те, що вона забезпечує симетричне підсилення сигналу, а також можливість регулювання коефіцієнта посилення за допомогою одного резистора Rg, що дозволяє легко налаштувати параметри підсилення відповідно до потреб дослідження.

Така схема реалізована в мікросхемі INA333 фірми Texas Instruments (Micro-Power (50μA), Zerø-Drift, Rail-to-Rail Out Instrumentation Amplifier).

Мікросхема INA333, окрім входів підсилювача та висновків для підключення зовнішнього резистора, що встановлює посилення, має висновок REF. У разі використання двунапрямленої схеми живлення цей висновок з'єднується з загальним входом схеми. Оскільки ми плануємо використовувати підсилювач разом з мікроконтролером типу "Arduino", який має однонапрямлений аналоговий вхід АЦП (від 0В до +5В), а також вихід +5В для живлення зовнішніх пристроїв, доцільно побудувати схему для однонапрямленого джерела живлення з напругою +5В.

КПЗ–2024

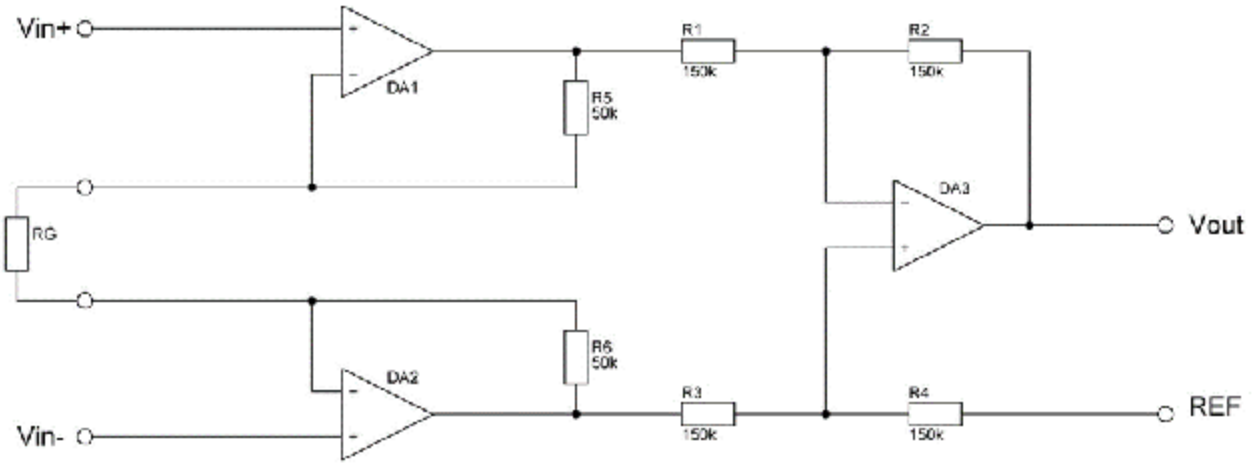


Рисунок 4.2 Схема електрична інструментального підсилювача.

Для цього потрібно створити опорну напругу, яка дорівнює половині напруги живлення. Простий спосіб отримати опорну напругу - використати дільник напруги, який складається з двох резисторів однакового номіналу. (рисунок. 4.2).

У такому випадку, якщо використовувати джерело живлення з однонапрямленою схемою, опір подільника, який дорівнює опору паралельно підключених резисторів, додався б до R4 на схемі, що порушило б баланс схеми і змінило коефіцієнт ділення подільника напруги. Це в свою чергу зменшило б коефіцієнт приглушення спільного режектора і точність коефіцієнта посилення. Якщо б резистор R4 був доступний, то можна було б включити опір подільника в його опір. Однак в такому випадку якість підсилювача погіршилася б, оскільки зовнішні резистори матимуть інший температурний коефіцієнт опору порівняно з інтегрованими в схему резисторами. Це призвело б до можливого посилення спільного режектора при зміні температури навколишнього середовища. Використання низькоомного подільника, який має мінімальний вплив на опір R4, призвело б до значного збільшення споживання струму живлення, що неприйнятно для пристрою з автономним живленням. Ці проблеми можна вирішити за допомогою застосування повторювача напруги на основі операційного підсилювача (див. рис. 4.2). Застосування конденсатора з великою

КОПЗ_2024

ємністю, наприклад, С = 100 мкФ, дозволить зменшити вплив флуктуацій джерела живлення на підсилювач і зберегти коефіцієнт приглушення відхилень напруги джерела живлення на високому рівні, приблизно 100 дБ. Недоліком цієї схеми буде значний час на встановлення опорної напруги, але для схеми кардіографа це не принципово, оскільки для зняття кардіограми пацієнт повинен заспокоїтися і не рухатися протягом певного часу, приблизно 10 секунд. Якщо потрібно швидше включення схеми в роботу, то повторювач напруги можна виконати за схемою активного фільтра 2-го чи вищого порядку. В такому разі можна використовувати конденсатори малої ємності і отримати необхідний час для стабілізації схеми.

Сигнали, зняті з рук пацієнта, мають рівень до одного мілівольта. Щоб підвищити цей сигнал до рівня одного вольта, потрібне посилення в тисячу разів. Проте, якщо задати таке посилення на першому етапі, це призведе до перевантаження виходу підсилювача через збільшення постійної складової сигналу. Тому перший каскад посилює сигнал лише в десять разів. Опір Rg в інструментальному підсилювачі встановлюється згідно з документацією на мікросхему. Основне посилення, приблизно в сто разів, забезпечує другий каскад. Постійна складова при цьому відсікається за допомогою розділового конденсатора (С3), а конденсатор С5 зменшує підсилення для високочастотних перешкод, які можуть бути на високоомному вході.

Оскільки АЦП «Ардуіно» працює в діапазоні від 0 В до 5 В, вихідний підсилювач отримує зсув на 2,5 В, що дозволяє сигналу коливатися в межах ±2,5 В від цієї постійної складової. Щоб знизити рівень перешкод на вході підсилювача, використовуються екрановані провідники довжиною до одного метра. При використанні довших провідників на входах необхідно встановити повторювачі напруги для зменшення наведень, але також потрібно додатковий провід для живлення цих пристроїв.

КПЗ-2024

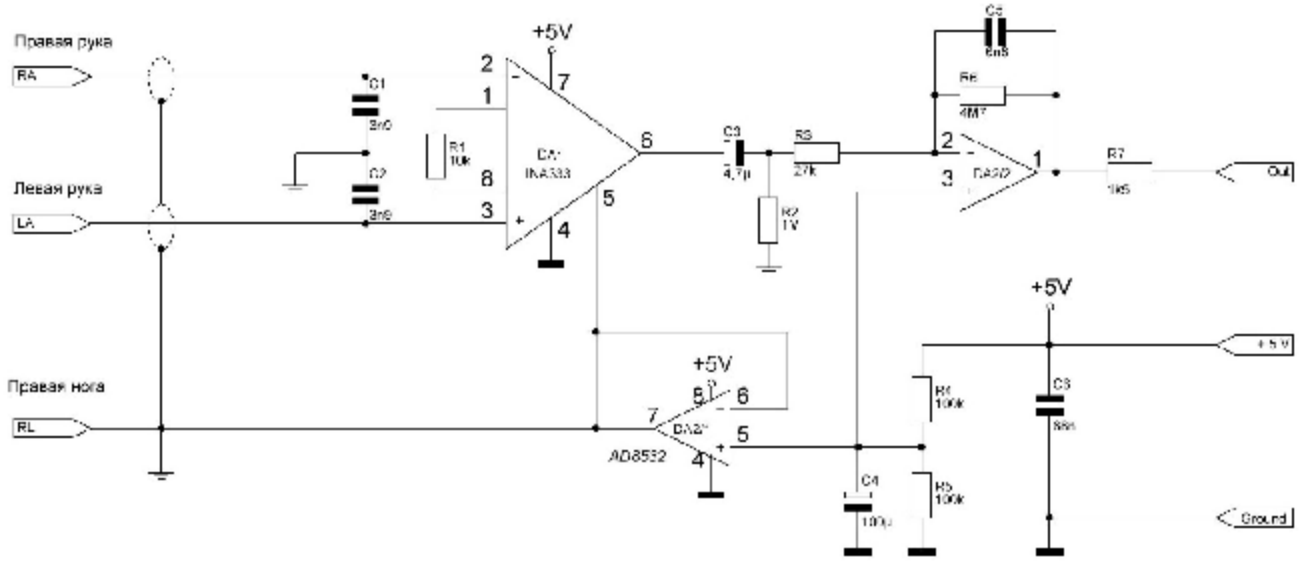


Рисунок 4.3 – Результуюча схема підсилювача

Приклади кардіограм, знятих за допомогою описаного вище підсилювача на цифровий осцилограф.

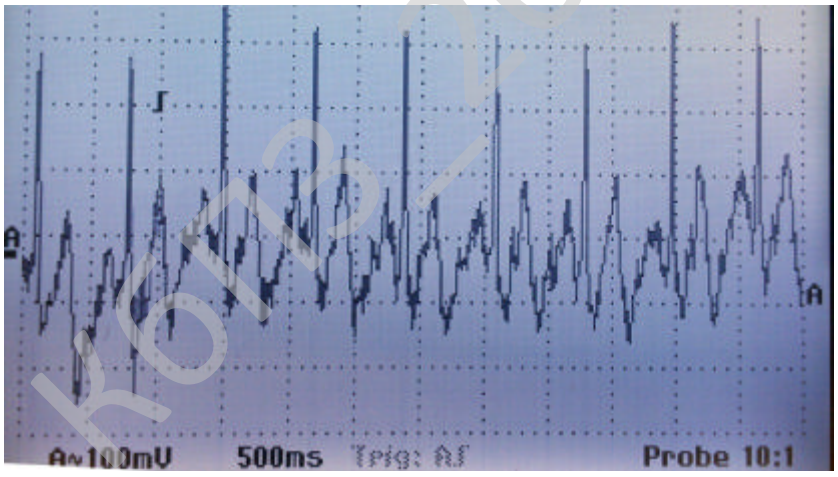


Рисунок 4.4 – Приклад роботи схема на екрані цифрового осцилографу

КПІ 2024

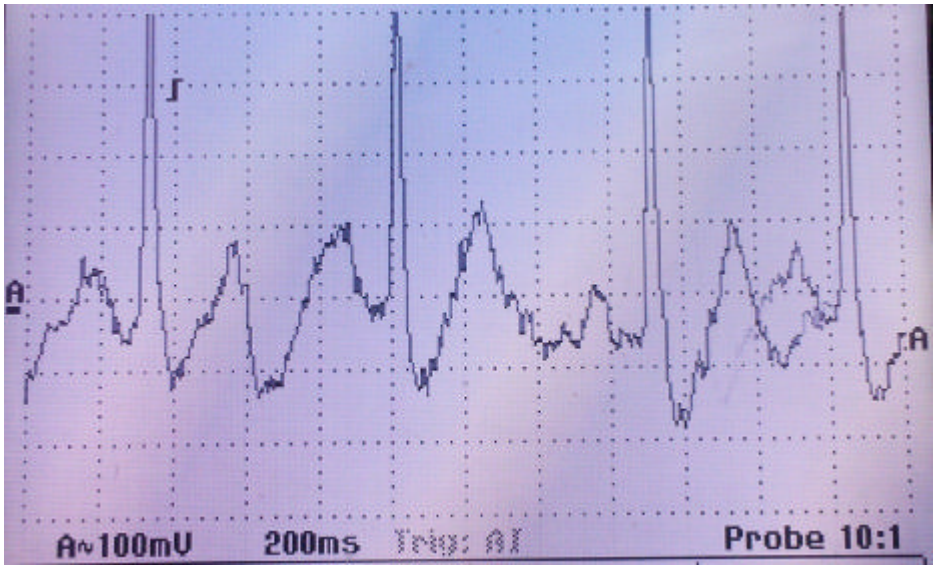


Рисунок 4.5 - Приклад роботи схема на екрані цифрового осцилографу

Програмний інтерфейс

Arduino може передавати дані на комп'ютер через USB-інтерфейс. USB, що розшифровується як Universal Serial Bus, це універсальна послідовна шина, яка використовується для підключення периферійних пристроїв до комп'ютера. Логотип USB складається з чотирьох геометричних фігур: квадрата, трикутника, великого кола і малого кола.

USB-шину використовують як послідовний інтерфейс для передачі даних від середньо- і низькошвидкісних периферійних пристроїв. Шина FireWire раніше вважалася кращим вибором для високошвидкісних пристроїв, однак після появи пристроїв з підтримкою USB 3.0 ця перевага стала обговорюваною.

USB-кабель складається з двох пар звитих дротів: одна пара передає дані у двох напрямках завдяки диференціальному з'єднанню, тоді як друга пара забезпечує електроживлення периферійних пристроїв з напругою +5 В та силою струму 500 мА. Ця функція живлення дозволяє використовувати USB-пристрої без зовнішніх джерел живлення та заряджати акумулятори мобільних пристроїв, таких як камери та плеєри, які споживають до 500 мА. У новітньому стандарті USB 3.0 максимальне навантаження на лінії живлення збільшено до 900 мА.

КОПЗ_2024

Підключення за допомогою USB-кабелів створює інтерфейс між USB-пристроями і USB-хостом. Хостом слугує USB-контролер, керований операційною системою, який включає в себе USB-концентратор, званий також хабом. Цей хаб служить стартовою точкою для формування мережі пристроїв у топології "зірка". Його називають кореневим концентратором. До портів цього хабу можна підключати інше USB-обладнання та додаткові зовнішні хаби. Загальна кількість пристроїв у мережі може досягати 127, організованих не більше ніж у п'ять рівнів каскаду, не враховуючи рівень кореневого хаба.

Дизайн USB-конекторів створений для того, щоб можна було безпечно під'єднувати та від'єднувати пристрої в режимі роботи, без вимкнення хоста. Це досягається завдяки подовженому контакту заземлення GND у порівнянні з іншими контактами. Така особливість дозволяє вирівняти потенціали корпусів перед замиканням сигнальних контактів, забезпечуючи захист електроніки пристрою від статичної електрики.

Бібліотека на Java для роботи с послідовним портом

JSSC, або Java Simple Serial Connector, це бібліотека Java, яка спрощує роботу з COM-портом і була вперше представлена у 2010 році. Заснована через відсутність ефективних інструментів для роботи з COM-портами, jSSC розповсюджується під ліцензією LGPL і стала популярною альтернативою іншим бібліотекам, таким як javax.comm та rxtx.

Основні компоненти jSSC включають:

-SerialNativeInterface: Клас, який забезпечує доступ до "нативних" методів jSSC.

-SerialPort: Клас, який використовується для безпосередньої роботи з потрібним COM-портом.

-SerialPortEventListener: Інтерфейс, який слід реалізувати для реагування на події порту.

КПЗ-2024

Ось як може виглядати приклад роботи з jSSC. Припустимо, ви маєте пристрій, що використовує апаратне керування потоком і ви хочете отримувати відповіді на запити від цього пристрою.

Ось приклад реалізації.

```
import jssc.SerialPort; import jssc.SerialPortEvent;
import jssc.SerialPortEventListener; import jssc.SerialPortException;
public class Test {
private static SerialPort serialPort; public static void main(String[] args) {
// serialPort = new SerialPort("COM1"); try {
// serialPort.openPort();
//
serialPort.setParams(SerialPort.BAUDRATE_9600,
SerialPort.DATABITS_8, SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
serialPort.setFlowControlMode(SerialPort.FLOWCONTROL_RTSCTS_IN | SerialPort.FLOWCONTROL_RTSCTS_OUT);
serialPort.addEventListener(new PortReader(), SerialPort.MASK_RXCHAR);
//Отправляем запрос устройству
serialPort.writeString("Get data");
}
catch (SerialPortException ex) { System.out.println(ex);
}
}
private static class PortReader implements SerialPortEventListener {
public void serialEvent(SerialPortEvent event) { if(event.isRXCHAR() && event.getEventValue() > 0){
try {
}
catch (SerialPortException ex) { System.out.println(ex);
}
}
}
}
}
```

КБПЗ_2024

Розроблена мною програма представлена нижче.

```
import  java.io.BufferedWriter; import java.io.FileWriter;
import java.io.IOException;
import  java.nio.charset.Charset; import java.util.ArrayList;
import java.util.List;
import jssc.SerialPort;
import jssc.SerialPortEvent;
import  jssc.SerialPortEventListener;
import  jssc.SerialPortException; public class Starter {
private static SerialPort serialPort;
private static final String filePath =
"/Users/Etherlord/Desktop/text.txt";
public static List<String> values = new ArrayList <>(); public
static void main(String[] args) {
serialPort = new SerialPort ("/dev/tty.usbmodem1411");
try {
serialPort.openPort ();
serialPort.setParams (SerialPort.BAUDRATE_9600 ,
SerialPort.DATABITS_8 ,
SerialPort.STOPBITS_1 ,
SerialPort.PARITY_NONE );
serialPort.setEventsMask (SerialPort.MASK_RXCHAR );
serialPort.addEventListener (new ComPortListener ());
} catch (SerialPortException ex) { System.out .println (ex);
}
}
private static class ComPortListener implements
SerialPortEventListener {
public void serialEvent(SerialPortEvent event) { if (event.isRXCHAR
()) {
try {
FileWriter (
BufferedWriter wr = new BufferedWriter (new
filePath , true )); byte [] buf =
serialPort.readBytes (event.getEventValue ());
if (buf.length > 0) {
```

КОПЗ_2024

```
        String msg = new String (buf,
        Charset.forName("utf-8"));
        }
        wr.write (msg); System.out .print (msg); wr.close ();
        } catch (SerialPortException | IOException ex) { System.out .println
(ex);
        }
        }
        }
```

Обробка сигналу

Цифрова обробка сигналів (ЦОС), яка включає в себе як апаратні, так і програмні компоненти, активно розвивається у сфері обчислювальної техніки. Ця галузь тісно пов'язана з теорією інформації, зокрема, з теоріями оптимального прийому сигналів і розпізнавання образів. Основне завдання у першому напрямку — виділення сигналу серед шумів і перешкод, у другому — автоматична класифікація і ідентифікація сигналів.

У процесі цифрової обробки сигнали представляються послідовностями чисел або символів. Ціль обробки може бути спрямована на аналіз характеристик сигналу або його перетворення в більш зручну форму. Класичні чисельні методи, такі як інтерполяція та інтегрування, є основою алгоритмів ЦОС. Прогрес у розробці інтегральних схем і використання швидкодіючих комп'ютерів дозволяє створювати все більш ефективні алгоритми.

Цифрова обробка сигналів використовується у багатьох галузях, включаючи біомедицину, акустику, радіолокацію, сейсмологію, зв'язок і навіть ядерну техніку. Основні переваги ЦОС порівняно з аналоговою обробкою включають високу точність, програмованість, гнучкість, адаптивність і технологічність.

Відкриття ефективних алгоритмів для обчислення перетворень Фур'є у 1965 році значно прискорило розвиток цифрової обробки сигналів. Ці алгоритми стали відомі як швидке перетворення Фур'є (ШПФ). ШПФ зробило великий вплив, оскільки дозволило значно скоротити час обчислення, необхідний для

КБПЗ-2024

спектрального аналізу, який є критичним елементом у багатьох алгоритмах обробки сигналів. Завдяки цьому, розробка складних алгоритмів обробки сигналів у реальному часі стала можливою, а також відкрилася можливість реалізації цих алгоритмів на спеціалізованих цифрових пристроях.

Цифрова обробка сигналів включає математичні методи та алгоритми, які здійснюють обчислення над цифровими сигналами. Серед них — алгоритми цифрової фільтрації, спектрально-кореляційний аналіз, алгоритми модуляції та демодуляції, а також методи адаптивної обробки. Ці методи використовуються у широкому спектрі застосувань, включаючи зв'язок, радіолокацію, медичну діагностику та інші технологічні області.

Для реалізації цифрової обробки сигналів (ЦОС) використовуються різноманітні технології та пристрої. Серед них — жорстка логіка, програмовані логічні інтегральні схеми (ПЛІС), загальнопризначені мікропроцесори, мікроконтролери, персональні комп'ютери для комп'ютерної обробки сигналів, та спеціалізовані цифрові сигнальні процесори (ЦСП). Ці пристрої надають різні рівні гнучкості, потужності та ефективності обробки, відповідаючи конкретним потребам застосування.

Термін "сигнал" у технічних дисциплінах може мати широкий спектр значень, відсутність строгої термінології дозволяє його використовувати у багатьох контекстах. Сигнал може визначатися як матеріальний носій для передачі та використання інформації — будь то електричний, магнітний, або оптичний сигнал. Також це може бути фізичний процес, який відображає інформацію через зміни певних параметрів носія, таких як напруга, частота, або інтенсивність світлового потоку, що може змінюватися у часі, просторі, або відповідно до інших змінних.

КБПЗ_2024

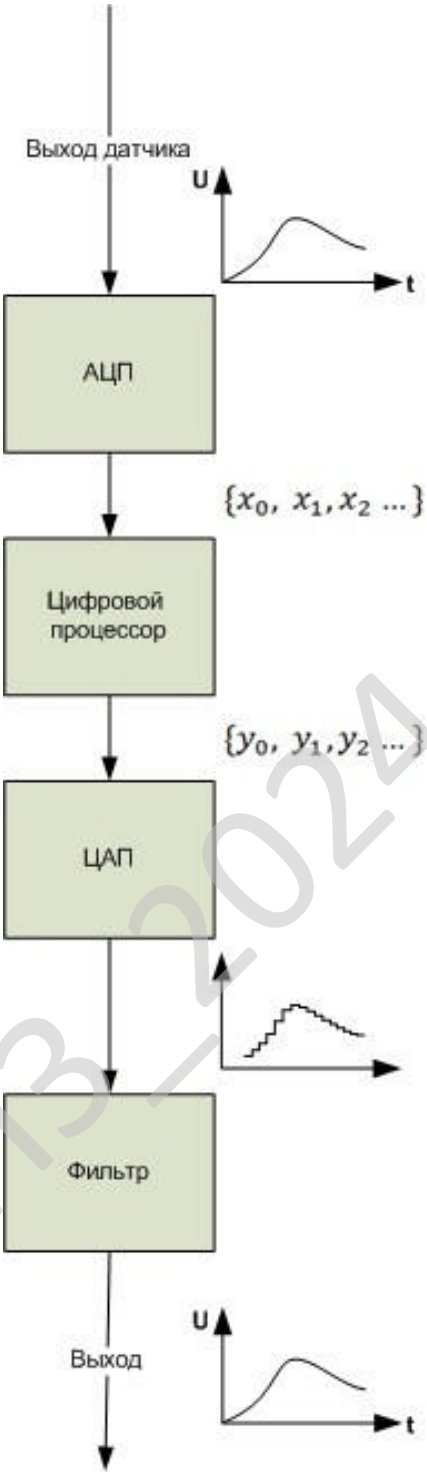


Рисунок 4.6 - Структурна схема цифрової обробки сигналу

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Нижче наведено алгоритм роботи реєстрації пацієнта в системі моніторингу:

КбПЗ_2024

Початок процесу: Пацієнт або медичний працівник запускає процес реєстрації нового пацієнта в системі моніторингу.

Збір інформації: Спочатку збирається базова інформація про пацієнта, така як:

- ПІБ пацієнта;
- дата народження;
- адреса проживання;
- контактні дані.

Інші особисті дані, які можуть бути важливими для моніторингу (наприклад, стани здоров'я, хронічні захворювання тощо).

Створення облікового запису: На основі зібраної інформації створюється обліковий запис пацієнта в системі моніторингу. Це може включати створення унікального ідентифікатора користувача та пароля для доступу до системи.

Введення медичної інформації: Після створення облікового запису додаткова медична інформація може бути додана до профілю пацієнта. Це може включати медичну історію, результати аналізів, інформацію про призначені ліки тощо.

Призначення пристроїв моніторингу (за потреби): Якщо пацієнт підлягає моніторингу з використанням медичних пристроїв (наприклад, моніторів тиску, датчиків пульсу тощо), ці пристрої можуть бути призначені і пов'язані з обліковим записом пацієнта.

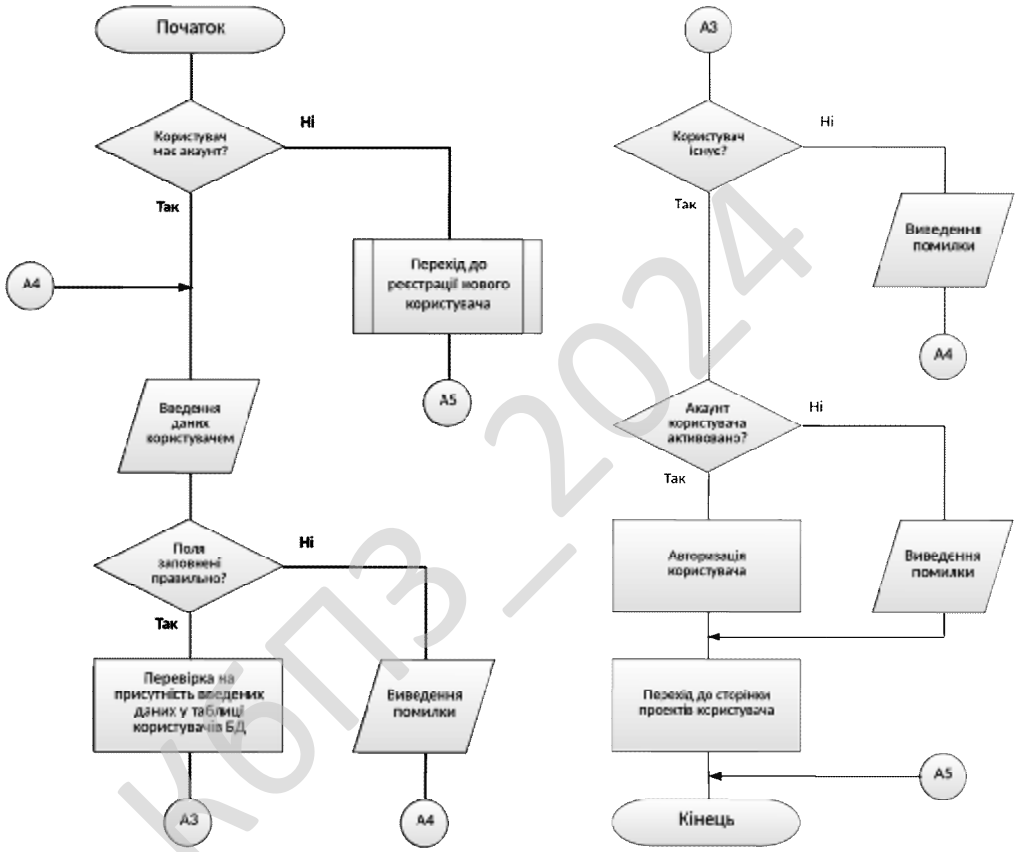
Підтвердження і завершення реєстрації: Перед завершенням процесу реєстрації пацієнт може підтвердити або оновити інформацію, а також надати згоду на умови використання системи моніторингу.

Повідомлення про успішну реєстрацію: Після завершення процесу пацієнт та/або медичний персонал отримують повідомлення про успішну реєстрацію в системі моніторингу.

КПЗ_2024

Доступ до системи моніторингу: Пацієнт може увійти до системи моніторингу, використовуючи свій обліковий запис та пароль, для перегляду своїх даних та отримання необхідної медичної допомоги.

Підтримка та оновлення: Система моніторингу може надавати підтримку та оновлення для пацієнтів, включаючи нагадування про прийом ліків, заплановані консультації з лікарем, а також інші корисні функції.



КПЗ_2024

Рисунок 4.7 - Блок-схема авторизації в додатку

Наступним кроком було розробка алгоритму роботи системи.

Початок: Початок блок-схеми.

Початок вимірювання: Старт вимірювання серцевого ритму.

Зчитування сигналу серцевого ритму: Сигнал зчитується з сенсора або електрокардіографа.

Попередній аналіз сигналу: Операція попереднього аналізу сигналу для фільтрації шумів та підготовки до обробки.

Обробка сигналу: Обробка сигналу для виявлення серцевих скорочень та визначення інтервалів між ними.

Вимірювання часу інтервалів: Вимірювання часу між послідовними серцевими скороченнями для визначення серцевого ритму.

Аналіз результатів: Операція аналізу виміряних інтервалів для визначення серцевого ритму, включаючи частоту та регулярність.

Відображення результатів: Відображення виміряного серцевого ритму на екрані або іншому пристрої візуалізації.

Завершення вимірювання: Зупинка процесу вимірювання.

Кінець: Кінець блок-схеми.

Розроблений алгоритм представлено на рисунку 4.8.

КбПЗ_2024

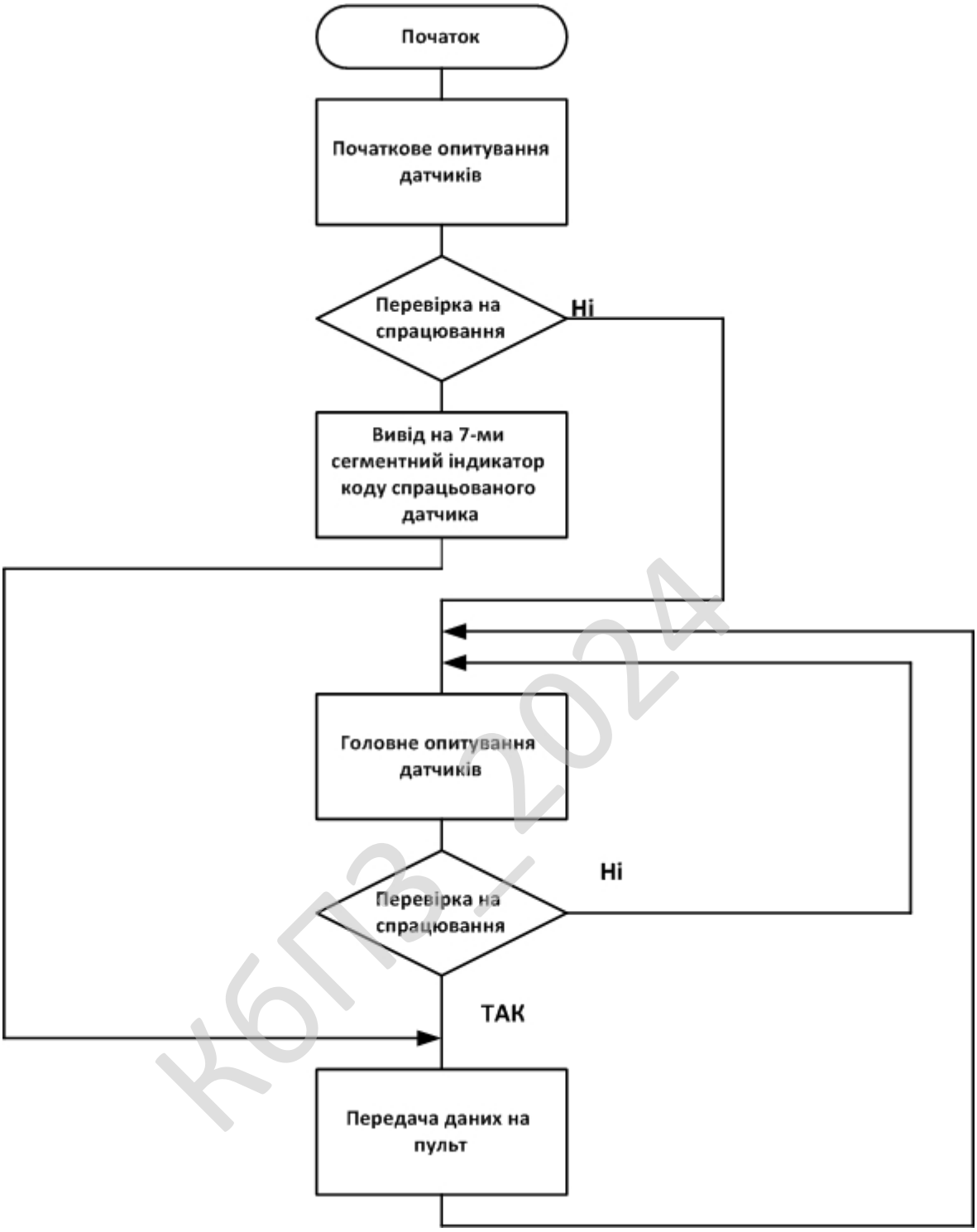


Рисунок 4.8 - Блок-схеми алгоритму роботи системи моніторингу

Висновки

Зростаюча популярність платформи Arduino є стимулом для розробників створювати все нові і нові прострої, розширючи ринок доступних рішень.

КбПЗ-2024

Простота використання платформи дає змогу будь-якій людині використовувити плати розширення, модулі та датчики до Arduino.

4.2 Захист розробленого програмного забезпечення

Захист програмного забезпечення (ПЗ) для систем, розроблених на платформі Arduino, може бути доволі проблемним, оскільки Arduino є відкритою платформою з легким доступом до апаратного забезпечення і коду. Проте, існує кілька методів, які можуть допомогти підвищити рівень безпеки проекту:

Мінімізація програмного коду на Arduino.

Залишайте на Arduino лише найнеобхідніший код для забезпечення функціональності. Це зменшує можливості для експлойтів. Вся бізнес-логіка або обробка даних, якщо можливо, повинні виконуватись на зовнішньому сервері або в захищеній системі.

Шифрування зв'язку

Якщо ваша система Arduino комунікує через мережу або інтернет, використовуйте зашифровані протоколи зв'язку, такі як TLS/SSL, для захисту даних від перехоплення або маніпуляції.

Обмеження фізичного доступу

Захистіть апаратну частину від несанкціонованого доступу. Це може включати корпус з замком для вашої Arduino плати, особливо якщо вона використовується у публічно доступних місцях.

Захист від запису на платі

Деякі моделі Arduino дозволяють встановити захист від запису на мікроконтролері, що може запобігти змінам в програмному забезпеченні без спеціального скидання цього захисту.

Автентифікація та контроль доступу

Реалізуйте системи автентифікації та контролю доступу для керування взаємодією з пристроєм. Це може включати паролі, токени доступу або інші

КБПЗ-2024

методи верифікації.

Використання цифрових підписів

Використовуйте цифрові підписи для перевірки справжності коду, завантаженого на Arduino. Це дозволяє переконатися, що програмне забезпечення не було змінено.

Оновлення та патчі

Регулярно оновлюйте програмне забезпечення Arduino, включаючи бібліотеки та скетчі, для виправлення будь-яких відомих уразливостей.

Моніторинг стану системи

Розгляньте можливість впровадження моніторингу для виявлення та реагування на нестандартну поведінку або спроби несанкціонованого доступу.

Захист даних пацієнтів

bcrypt є однією з найпопулярніших хеш-функцій для захисту паролів, особливо веб-додатків. Основною особливістю bcrypt є використання алгоритму "Eksblowfish", модифікованої версії шифру Blowfish, яка включає дорогу процедуру підготовки ключів. Ця особливість робить bcrypt особливо стійким до атак злому за допомогою "brute force" (перебором), адже час необхідний для генерації одного хешу може бути налаштований (шляхом збільшення кількості раундів в підготовці ключів), зробивши процес більш часозатратним.

Хешування паролів

За допомогою bcrypt здійснюється наступним чином:

- Salt: Для кожного пароля генерується унікальний випадковий рядок, званий сіллю, щоб забезпечити, що однакові паролі не будуть мати однаковий хеш;
- конфігурація: bcrypt дозволяє вибрати кількість раундів для генерації хешу. Більше раундів означає більшу складність і більший час обробки, що підвищує безпеку.
- генерація хешу: використовуючи пароль, сіль та вибрану кількість раундів, bcrypt створює хеш, який потім може бути збережений у базі даних.

КПЗ-2024

Безпека bcrypt полягає в тому, що він може адаптуватися до зростаючих обчислювальних можливостей, просто збільшуючи кількість раундів. Це робить алгоритм дуже надійним у випадках, коли потрібно зберігати паролі безпечно і ефективно протягом тривалого часу.

Bcrypt є частиною стандартних бібліотек багатьох операційних систем і використовується для забезпечення високої стійкості до атак, що робить його важливим інструментом в захисті паролів і особистих даних користувачів.

Заходи безпеки

Завжди уникайте використання облікових записів адміністратора або власника для підключення до бази даних. Натомість, створюйте спеціалізовані користувацькі акаунти з строго обмеженими дозволами для забезпечення безпеки. Використовуйте підготовлені запити з параметрами для зменшення ризику SQL-ін'єкцій, такі можливості пропонують такі бібліотеки, як PDO та MySQLi.

Переконайтесь, що всі вхідні дані відповідають очікуваному формату, використовуючи різні вбудовані функції PHP для перевірки типу даних, такі як is_numeric() та ctype_digit(), або використання регулярних виразів. Для валідації вводу можна використати ctype_digit(), або жорстко задати тип за допомогою settype(), або перетворити дані у числове представлення за допомогою sprintf().

Якщо ваша система баз даних не підтримує підготовлені вирази з параметрами, завжди використовуйте функції для екранування рядків, специфічні для вашої СУБД, наприклад, mysql_real_escape_string() або sqlite_escape_string(). Уникайте використання загальних функцій типу addslashes(), оскільки вони можуть бути ненадійними в деяких контекстах.

Не відображайте інформацію про структуру або налаштування бази даних у вашому додатку. Користуйтесь процедурами, що дозволяють абстрагувати взаємодію з даними, це додає додатковий рівень абстракції і безпеки.

Хоча моніторинг сам по собі не може запобігти всім видам атак, він є важливим інструментом для виявлення та аналізу безпекових інцидентів.

КПЗ-2024

Логування є ключовим елементом у цьому процесі, а інформація, яку ви зберігаєте у лог-файлах, може бути надзвичайно корисною при розслідуванні зламів.

КБПЗ_2024

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Проектування та налаштування аналогових схем - це трудомісткий та часовитратний процес, який вимагає глибоких знань. Однією з основних складнощів у розробці таких схем є непередбачуваність результатів. Для досягнення потрібних параметрів часто доводиться багаторазово коригувати помилки та вносити зміни.

Підключення елементів

На етапі створення прототипу обрано мікроконтролер Arduino, але вибір конкретної моделі не є критичним для функціональності проекту. Arduino Nano було вибрано через його компактні розміри, що є важливим для мініатюризації кінцевого пристрою. Діаграма підключення модуля AD8232 до Arduino UNO представлена на рисунку 5.1.

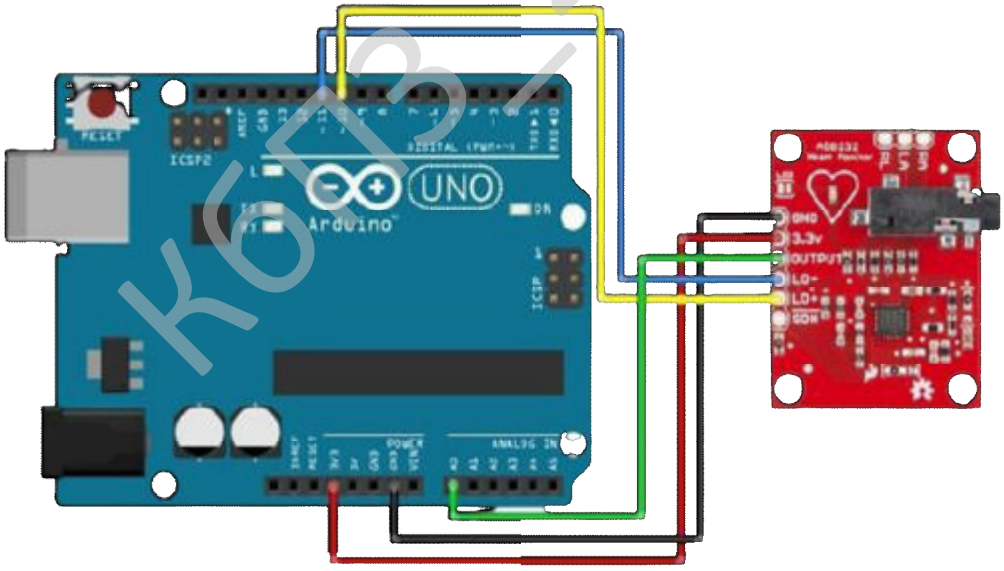


Рисунок 5.1 - Схема підключення модуля AD8232 до плати Arduino

КБПВ-2024

Таблиця 5.1 – підключення AD8232 до плати Arduino

Вивід модуля	Функція	Вивід Arduino
GND	Загальний	GND
3.3B	Живлення	3.3B
OUTPUT	Вихідний сигнал	A0
LO-	Leads-off Detect -	11
LO+	Leads-off Detect –	10
SDN	Shutdown	не використовується

Розробка корпусу приладу

Під час розробки корпусу для пристрою та створення 3D моделі використовувався програмний продукт SolidWorks, один з лідируючих інструментів у сфері автоматизованого проектування та 3D моделювання. СолідВоркс широко відомий своєю здатністю до точного моделювання комплексних деталей та зручністю у підготовці моделей до 3D друку, що знижує ризики помилок, які можуть виникнути при традиційному кресленні. У програмі можна ефективно працювати з різними типами геометрії, від простих до складних форм. Моделювання у SolidWorks засноване на використанні базових геометричних форм і виконанні маніпуляцій із ними для досягнення потрібної конфігурації. Початковий етап моделювання включав визначення розмірів майбутнього корпусу, що мав вміщати всередині лише дві невеликі плати, тому великий корпус не був потрібен. Враховуючи численні переваги SolidWorks, було вирішено використовувати цю систему для проектування корпусу.

Конструкція корпусу задумана як легка у монтажі та розбірці, складається з шести ключових елементів: основи, задньої панелі, передньої панелі, двох бічних стінок та рухомої кришки, що забезпечує доступ до внутрішніх компонентів пристрою. Враховано також наявність отворів для кріплення плат, що полегшує збірку. Кожна частина корпусу спочатку проектувалася окремо у SolidWorks шляхом створення двовимірних креслень, а потім перетворювалася у

КБПЗ-2024

тривимірні моделі. Завдяки функціональності програми можливо зібрати всі деталі у єдину збірку та перевірити її на можливість монтажу і виявити та швидко усунути помилки креслень. В процесі складання збірки можна визначити, які елементи будуть рухомими, що дозволяє симулювати їх рух у готовій моделі. Ілюстрації різних стадій збірки корпусу і детальні креслення кожної частини зображено на супутніх рисунках.

Виготовлення деталей корпусу

При виборі методу та матеріалу для виробництва деталей корпусу, особлива увага була звернена на умови експлуатації приладу. Оскільки пристрій не буде використовуватися в екстремальних умовах, необхідність у використанні металевих сплавів або металевого корпусу було відкинуто. Розглядались варіанти з використанням деревини та пластику. З огляду на економічні та технічні аспекти, зокрема час і вартість виготовлення на 3D принтері, було вирішено виготовляти корпус із тонкої фанери.

Для роботи з деревиною вибрано метод лазерної різки. Програма SolidWorks дозволяє експортувати креслення в форматі DXF (Drawing Exchange Format), який використовується в автоматизованих системах проектування для обміну графічними даними, і який придатний для ЧПУ обладнання з лазерною різкою. Лазерна обробка фанери має переваги, такі як висока точність, чистота країв різу і відсутність відходів, оскільки матеріал на краях різу одразу згорає, забезпечуючи акуратність обробки та зниження втрат матеріалу.

КПЗ-2024

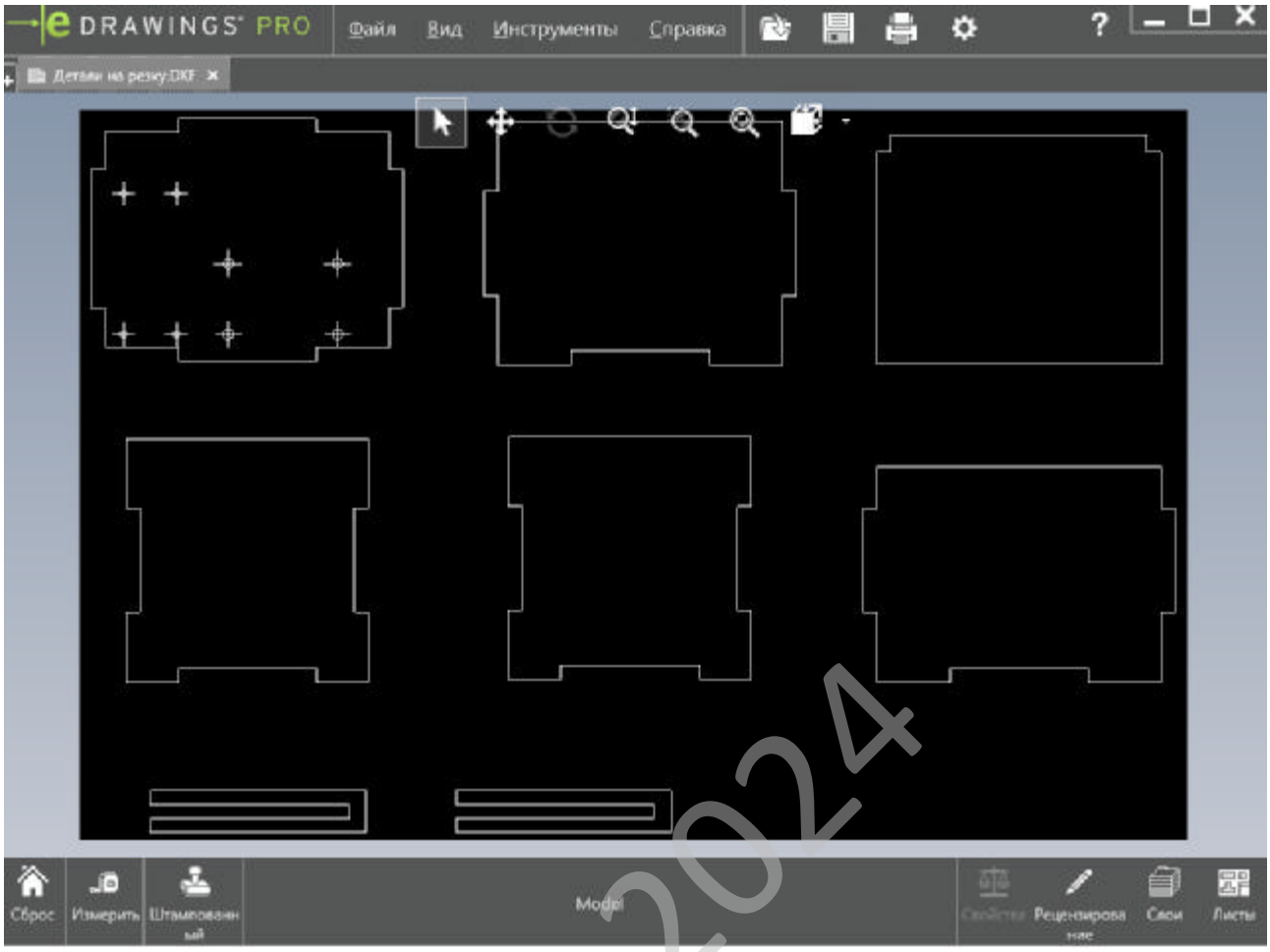


Рисунок 5.2 – Деталі конструкції в CAD форматі DXF

Середовище програмування мікроконтролера

В цьому проекті в якості основи для мікроконтролера обрано ATmega328, інтегрований в плату Arduino Leonardo. Програмування мікроконтролера здійснюється через Arduino IDE, що є стандартним розробницьким середовищем для плат Arduino.

КОПЗ_2024

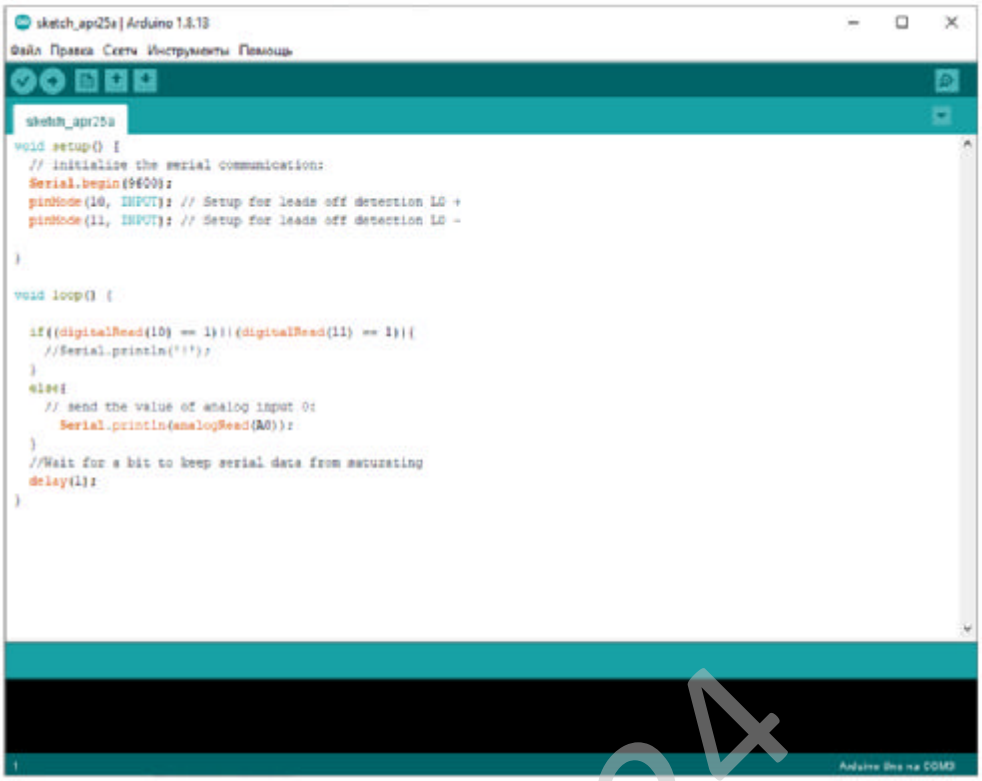


Рисунок 5.3 – Зовнішній вигляд середовища програмування Arduino IDE

Arduino IDE, інтегроване середовище розробки, використовує мову програмування, що нагадує спрощений варіант C++. Це середовище підтримує розробку на платах Arduino та інших сумісних системах. Мова, яка використовується в Arduino IDE, подібно до C++, вимагає строгого дотримання типізації і компіляції коду, а також слідування специфічним правилам написання коду.

КПІЗ 2024

6 ОСНОВНІ ВИСНОВКИ

З величезною швидкістю розвитку технологій складно встигати за новинками. Ринок пропонує різноманітні рішення як для домашнього, так і для професійного використання. Втім, на ринку медичних приладів все ще спостерігається дефіцит, особливо щодо доступних домашніх апаратів. Високі ціни роблять неможливим їх придбання для звичайних споживачів. Незважаючи на це, прості медичні пристрої, такі як ЕКГ, стають доступнішими завдяки платформам на кшталт Arduino. Ця платформа є доступною і потужною, дозволяючи задовольнити потреби користувачів. Встановлення модуля і завантаження програми не вимагають особливих навичок, що робить процес доступним кожному. Розроблений мною модуль ЕКГ коштує у 100 разів дешевше за аналоги з медичних магазинів, що робить його унікальним рішенням. Цей проект демонструє можливість створення функціонального прототипу ЕКГ на простій платі Arduino, що відкриває можливості для розробки біомедичних станцій для автоматичної діагностики та попередження серцево-судинних захворювань.

Сподіваюся, цей пристрій допоможе покращити медичне обслуговування Україні та світі.

КОПЗ-2024

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

2. Yasin, J.N., Mohamed, S.A.S., Haghbayan, MH. et al. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int. j. inf.tecnol. 13, 97–107 (2021). https://doi.org/10.1007/s41870-020-00513-w.

3. Sabry F. Self Driving Car: Solving Full Self-driving Need Solving Real-worldartificial Intelligence. One Billion Knowledgeable. 2022 . 293 p.

4. Shapiro D. G. Three Anecdotes from the DARPA Autonomous Land Vehicle Project. AI Magazine, 2008. 29(2), 40. https://doi.org/10.1609/aimag.v29i2.2108

5. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates SixScientific Studies from 2015-2018 SAE International. 2021. 252 p

6. Рудик А. В. Наукові основи та принципи побудови приладової системи вимірювання прискорення мобільного робота : дис. ... д-ра техн. наук : 05.11.01– Прилади та методи вимірювання механічних величин. Київ, 2018. – 460 с.

7. Поліщук М. М., Ткач М.М. Робототехнічні системи: проектування і моделювання: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.

8. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.

9. Staron M. Automotive Software Architectures: An Introduction. Second Edition Springer International Publishing A&G. 2022. 274 p

10. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.

11. Wevolver 2020 Autonomous Vehicle Technology Report. URL: https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehicec

КПІ – 2024

hnologyReportpdf/ (accessed 2023 June 1)

12. React.js [Електронний ресурс] – Режим доступу до ресурсу: https://https://uk.reactjs.org/

13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm

14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: https://vuejs.org/

15. A Look Inside ADAS Modules. URL: https://amkor.com/semiconductor-story/alook.

16.AV and ADAS Sensors. URL https://community.sw.siemens.com/s/article/AVand- ADAS-Sensors (accessed 2023 June 1)

17. Гуржій А. М. Основи автоматики та робототехніки: Навчальний посібник/ А.М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:–Дніпро:«Гарант СВ», 2021.- 243с.

18. Lim, B.S. et al. Autonomous Vehicle Ultrasonic Sensor Vulnerability and Impact Assessment. In Proceedings of the IEEE World Forum on Internet of Things (WFIoT), Singapore, 5–8 February 2018; pp. 231–236.

19. Саліхов М. М. Arduino – перспективний інструмент протитопування у робототехніці // Концептуальні шляхи розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції м. Львів, 9-10 червня 2023 року.– Львів: Львівський науковий форум, 2023. – с. 73-79.

20. Bräunl T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with RaspberryPi and Arduino Springer Nature, 2022. 519 p.

21. Навігаційні системи [Електронний ресурс] : навч. посіб. для студ. Спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / С.Л. Лакоза; КПІ ім. Ігоря Сікорського, 2021. — 80 с.

22. Imad M. et al. Navigation system for autonomous vehicle: A survey. JCSTS., vol. 2, no. 2, pp. 20–35, 2020.

23. Ковалець І. В. та ін. Технологія планування траєкторій руху

КОПЗ-2024

мобільних об’єктів з урахуванням перешкод на складній місцевості. Енергетика і автоматика. - 2017. - № 1. - С. 110-122.

24. HTML [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/uk/docs/Web/HTML

25. Aguilo I. et al. Artificial Intelligence Research and Development Hardcover by IOS Press. 2003. 500 p.

26. Гребенюк Б. А. Розробка підсистеми управлення інтелектуальним роботом / Б. А. Гребенюк // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National Universitety of Radio Elektronics [electronic edition], 2023. – Part 1. –336p. P. 263-269

28. Becker M. et al Obstacle avoidance procedure for mobile robots. ABCM Symposium series in Mechatronics. Vol. 2, 2006. P. 250-257.

29. CORS, XSS [Електронний ресурс] – Режим доступу до ресурсу: https://dev.to/maleta/cors-xss-and-csrf-with-examples-in-10-minutes-35k3

30. AJAX [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/AJAX

31. Fetch API [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/API/Fetch_API

33. Адамів, О.П. Моделі та інтелектуальні засоби адаптивного керування автономним мобільним роботом. Дис. канд.техн. наук. Одеса, 2007. 124 с.

34. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS’2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

35. npm [Електронний ресурс] – Режим доступу до ресурсу: https://www.npmjs.com/

36. Install MongoDB [Електронний ресурс] – Режим доступу до ресурсу: https://docs.mongodb.com/manual/administration/install-on-linux/

37. Як встановити Node.js [Електронний ресурс] – Режим доступу до

КПЗ_2024

ресурсу: https://www.digitalocean.com/community/tutorials/node-js-ubuntu-18-04-ru

38. Linux [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Linux

39. npm-audit [Електронний ресурс] – Режим доступу до ресурсу: https://docs.npmjs.com/cli/audit

40. Kojima T. et al. Intelligent Technology for More Advanced Autonomous Driving. 2018. Hitachi Review Vol. 67, No. 1. p. 58–63

41. Кардиограф на Arduino своїми руками. Режим доступу : http://www.prointellekt.ru/EKG1.php - Дата доступу : 03.02.2024

42. MongoDB Compass [Електронний ресурс] – Режим доступу до ресурсу: https://www.mongodb.com/products/compass

43. Postman [Електронний ресурс] – Режим доступу до ресурсу: https://www.postman.com/

44. Саліхов М. М. Від автоматизованих до автономних транспортних засобів // Наукові досягнення та інновації: шлях до успіху. X Всеукраїнська мультидисциплінарна науково-практична Інтернет-конференція, 31 травня 2023, Україна, Київ : зб. матеріалів — Електрон. дан. — Київ : Ярочéнко Я. В., 2023.

КбПЗ_2024

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....2
2 Підстава для розробки.....2
3 Мета та призначення розробки.....2
4 Джерела розробки.....2
5 Технічні вимоги.....2
5.1 Вміст проекту.....2
5.2 Показники призначення.....3
5.3 Вимоги до функціональних характеристик.....3
5.4 Вимоги до архітектури.....3
5.5 Вимоги до надійності.....3
5.6 Умови експлуатації.....4
5.7 Вимоги до складу та параметрів технічних засобів.....4
5.8 Вимоги до інформаційної і програмної сумісності.....4
5.8.1 Обладнання.....4
5.8.2 Мова програмування.....4
5.8.3 Вхідні дані.....5
5.8.4 Вихідні дані.....5
6 Вимоги до програмної документації.....5
7 Перелік документів, що розробляються.....5
8 Етапи розробки.....6
9 Порядок контролю та приймання.....6

КОПЗ-2024

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку Програмне забезпечення клієнт-серверної моделі керування мікроконтролерними системами

Підставою для розробки служить завдання на випускну кваліфікаційну роботу, видане на кафедрі програмування та захисту інформації (нак. №132-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної бакалаврської роботи є розробка програмного забезпечення клієнт-серверної моделі керування мікроконтролерними системами.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської дипломної роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

– вибір і обґрунтування методів реалізації проекту;

– розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

КПЗ_2024

5.2 Показники призначення

Система повинна забезпечувати:

– роботу клієнт-серверної моделі керування мікроконтролерними системами;

– цілісність даних у процесі роботи та при зберіганні;

– простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

– температура повітря: 19-20 град. по Цельсію;

– відносна вологість повітря до 80%;

КПЗ_2024

– атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

При розробці ПЗ потрібно використовувати наступні технології та мови програмування: C++, Arduino IDE. Для програмування мікроконтролера мова низького рівня С. Середовище програмування – PHP-Shtorm.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

КОПЗ – 2024

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

– Структурна схема системи керування – 1 аркуш.
– Функціональна схема системи – 1 аркуш.
– Діаграма процесів – 1 аркуш.
– Блок-схема авторизації в додатку – 1 аркуш.
– Блок схема алгоритму роботи системи моніторингу – 1 аркуш.
– Пояснювальна записка – 77 аркуша.

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи.

Постановка задачі на виконання кваліфікаційної роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної роботи.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

КбПЗ-2024

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання кваліфікаційної роботи на попередній захист 18.05.2024 р.

9.2 Подання кваліфікаційної роботи на захист 05.06.2024 р.

КБПЗ_2024

Додаток Б
(обов’язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник кваліфікаційної бакалаврської роботи
_____ Босько В.В

Програмне забезпечення мікропроцесорної системи обробки даних серцебиття на платформі Arduino

Лістинг програми

Код документу 12
Носій: CD/DVD-диск

Загальна кількість аркушів: 13

Літера: РП

КбПЗ_2024

Кропивницький – 2024 року

Скетч програми Arduino Drive_version_1.3.ino

```
#include <Bridge.h>
#include <YunServer.h>
#include <YunClient.h>
//#include <Servo.h>
#include <PID_v1.h>
//#include <Ultrasonic.h>
#define D1 4 //
#define M1 5 //
#define D2 6 //
#define M2 7
#define Trig 8
#define Echo 9
#define ledPin 13
double Setpoint, Input, Output;
PID myPID(&Input, &Output, &Setpoint,15,5,1, DIRECTbool autoMode;
bool autoReverse;
bool ON=false;
bool directionr = 0;
63
bool directionl = 0;
int valuel = 0;
int valuer = 0;
YunServer server;
//Servo servol;
//Servo servor;
//Ultrasonic ultrasonic(12, 13);
unsigned int impulseTime=0;
unsigned int distance_sm=0;
void setup() {
 autoMode = false;
 autoReverse = false;
 pinMode(Trig, OUTPUT); //
 pinMode(Echo, INPUT); //
 pinMode(ledPin, OUTPUT);

 Serial.begin(9600);

 digitalWrite(ledPin, HIGH);
 Bridge.begin();
 digitalWrite(ledPin, LOW);
 server.begin();
 Setpoint = 30;
64
```

КБПЗ_2024

```
 myPID.SetOutputLimits(0, 255);
 myPID.SetSampleTime(100);
 myPID.SetMode(AUTOMATIC);

 pinMode(D1, OUTPUT);
 pinMode(D2, OUTPUT);

 analogWrite(M1,0);
 analogWrite(M2,0);
}
int counter = 0;
YunClient client;
void loop() {
 if(client){
 client.connect("192.168.240.240", );
 client.print("Counter = "); client.println(counter);
 }
 YunClient tempClient = server.accept();

 counter++;

 if (tempClient) {
 client = tempClient;
 process(client);
 client.print("Counter = "); client.println(counter);
 client.stop();
65
 }else if(autoMode){
 Serial.print("Setpoint: "); Serial.println(Setpoint);
 Input = distance_sm;
 Serial.print("Input: "); Serial.println(Input);
 // it can only go forward or backward at once
 if((distance_sm - Setpoint) < 2 || (distance_sm - Setpoint) > 2){
 if(distance_sm > Setpoint){
 if(!autoReverse){
 myPID.SetControllerDirection(DIRECT);
 Serial.print(myPID.Compute());
 directionl = 0;
 valuel = 255 - (int)Output;
 directionr = 0;
 valuer = 255 - (int)Output;
 }else{
 if(directionr == 1 && valuer > 220){
 autoReverse = false;
 myPID.SetControllerDirection(DIRECT);
 Serial.print(myPID.Compute());
```

КбПЗ_2024

```
directionl = 0;
valuel = 255 - (int)Output;
directionr = 0;
valuer = 255 - (int)Output;
}else{
myPID.SetControllerDirection(REVERSE);
Serial.print(myPID.Compute());
directionl = 1;
valuel = (int)Output;
66
directionr = 1;
valuer = (int)Output;
}
}
}else{
if(autoReverse){
myPID.SetControllerDirection(REVERSE);
Serial.print(myPID.Compute());
directionl = 1;
valuel = (int)Output;
directionr = 1;
valuer = (int)Output;
}else{
if(directionr == 0 && valuer < 20){
autoReverse = true;
myPID.SetControllerDirection(REVERSE);
Serial.print(myPID.Compute());
directionl = 1;
valuel = (int)Output;
directionr = 1;
valuer = (int)Output;
}else{
myPID.SetControllerDirection(DIRECT);
Serial.print(myPID.Compute());
directionl = 0;
valuel = 255 - (int)Output;
directionr = 0;
valuer = 255 - (int)Output;
67
}
}
}
}
Serial.print("Output: "); Serial.println(Output);
Serial.print("valuer: "); Serial.println(valuer);
Serial.print("valuel: "); Serial.println(valuer);
```

КОПЗ_2024

```
 }
 // float dist_cm=ultrasonic.Ranging(CM);
 // Serial.print("Dist:");
 // Serial.println(dist_cm);
// if(dist_cm > 10) {
// directionl = directionr = 0;
// valuer = valuel = 0;
// }

 digitalWrite(D1, directionl); //
 digitalWrite(D2, directionr);
 //servol.write(valuel);
 //servor.write(valuer);
 analogWrite(M1,valuel);
 analogWrite(M2,valuer);

 // Serial.print("directr: "); Serial.println(directionr);
 // Serial.print("directl: "); Serial.println(directionl);
 /* Serial.print("valuer: "); Serial.println(valuer);
 Serial.print("valuel: "); Serial.println(valuel);*/
 // client.println(valuel);
68
 // Serial.println(valuel);
 // client.println(valuer);
 // Serial.println(valuer);


 getDistance();
 delay(100);
}
void getDistance(){
 digitalWrite(Trig, HIGH);
delayMicroseconds(10); //
 digitalWrite(Trig, LOW); //
 impulseTime=pulseIn(Echo, HIGH); //
 distance_sm=impulseTime/58; //
 if(autoMode && distance_sm > (Setpoint + 100)){
 distance_sm = Setpoint + 100;
 }
 Serial.println(distance_sm); //
 if (distance_sm<30) //
 {
 digitalWrite(ledPin, HIGH); //
 }
 else
 {
```

КОПЗ_2024

```
 digitalWrite(ledPin, LOW); //
 }
69
}
void process(YunClient client) {
 String command = client.readStringUntil('/');
 if (command == "digital") {
 digitalCommand(client);
 }
 if (command == "analog") {
 analogCommand(client);
 }
 if (command == "mode") {
 modeCommand(client);
 }
 if (command == "drive") {
 driveCommand(client);
 }

}
void driveCommand(YunClient client){
 String mode = client.readStringUntil('/');
 Serial.println(autoMode);
 if(mode == "power"){
 ON = (bool)client.parseInt();
70
 client.println(ON);
 Serial.println(ON);
 if(ON){
// if(!servol.attached() || !servor.attached()){
 //servol.attach(M2);
 //servor.attach(M1);
 client.println("Servos were attached");
 // Serial.println("Servos were attached");
 // }
 }else{
 //servor.detach();
 //servol.detach();
 analogWrite(M1,0);
 analogWrite(M2,0);
 }
 }else if (mode == "pursuit"){
 int tempDist = client.parseInt();
 if(tempDist){
 Setpoint = tempDist;
 autoMode = true;
```

КОПЗ_2024

```
}
}else{
if(client.readStringUntil(':') == "P"){
if(client.parseInt() == 1){
bool directionr = 0;
bool directionl = 0;
int valuel = 0;
int valuer = 0;
71
ON = ON ^ 1;
if(ON){
client.println("Power turned ON");
// Serial.println("Power turned ON");
}else{
client.println("Power turned OFF");
// Serial.println("Power turned OFF");
}
}
}
if(ON){
int X = 0, Y = 0, Z = 0;
bool correct = false;
client.read();
if(client.readStringUntil(':') == "Y"){
Y = client.parseInt();
client.read();
if(client.readStringUntil(':') == "X"){
X = client.parseInt();
client.read();
if(client.readStringUntil(':') == "Z"){
Z = client.parseInt();
correct = true;
}
}
}
if(correct){
Y = dataCorrect(Y);
72
X = dataCorrect(X);
Z = dataCorrect(Z);

if(Y >= 0){
directionl = 0;
valuel = Y*0.255;
directionr = 0;
valuer = Y*0.255;
```

КОПЗ_2024

```
 }else{
 directionl = 1;
 valuel = 255 + Y*0.255;
 directionr = 1;
 valuer = 255 + Y*0.255;
 }
 if(X >= 0){
 if(Y >= 0){
 valuer = valuel - (int)((float)X*valuel/1000);
 }else{
 valuer = valuel + (int)((float)X*(255-valuel)/1000);
 }
 }else{
 if(Y >= 0){
 valuel = valuer + (int)((float)X*valuer/1000);
 }else{
 valuel = valuer - (int)((float)X*(255-valuer)/1000);
 }
 }
 if(X == 0 && Y == 0){
73
 if(Z >= 0){
 directionl = 0;
 valuel = Z*0.255;
 directionr = 1;
 valuer = 255 - Z*0.255;
 }else{
 directionr = 0;
 valuer = 0 - Z*0.255;
 directionl = 1;
 valuel = 255 + Z*0.255;
 }

 }
 }
);

 }
 }
}
int dataCorrect(int val){
 if(val < 200 && val > -200){ return 0; }
 if(val > 1000) return 1000;
 if(val < -1000) return -1000;
 return val;
}
```

КОПЗ_2024

```
void digitalCommand(YunClient client) {
74
 int pin, value;
 pin = client.parseInt();
 if (client.read() == '/') {
 value = client.parseInt();
 digitalWrite(pin, value);
 }
 else {
 value = digitalRead(pin);
 }
 client.print(F("Pin D"));
 client.print(pin);
 client.print(F(" set to "));
 client.println(value);
 String key = "D";
 key += pin;
 Bridge.put(key, String(value));
}
void analogCommand(YunClient client) {
 int pin, value;
 pin = client.parseInt();
 if (client.read() == '/') {
75
 value = client.parseInt();
 analogWrite(pin, value);
 client.print(F("Pin D"));
 client.print(pin);
 client.print(F(" set to analog "));
 client.println(value);
 String key = "D";
 key += pin;
 Bridge.put(key, String(value));
 }
 else {
 value = analogRead(pin);
 client.print(F("Pin A"));
 client.print(pin);
 client.print(F(" reads analog "));
 client.println(value);
 String key = "A";
 key += pin;
 Bridge.put(key, String(value));
 }
}
void modeCommand(YunClient client) {
```

КбПЗ_2024

```
 int pin;
76
 pin = client.parseInt();
 if (client.read() != '/') {
 client.println(F("error"));
 return;
 }
 String mode = client.readStringUntil('\r');
 if (mode == "input") {
 pinMode(pin, INPUT);
 client.print(F("Pin D"));
 client.print(pin);
 client.print(F(" configured as INPUT!"));
 return;
 }
 if (mode == "output") {
 pinMode(pin, OUTPUT);
 client.print(F("Pin D"));
 client.print(pin);
 client.print(F(" configured as OUTPUT!"));
 return;
 }
 client.print(F("error: invalid mode "));
 client.print(mode);

// Зчитування даних з датчика

#include <OneWire.h>
OneWire ds (10); // чіп DS18S20 на 10-му виводі
void setup (void) {
 // запускаємо послідовний порт:
 Serial.begin (9600);
}
void loop (void) {
 byte i;
 byte present = 0;
 byte data [12];
 byte addr [8];
 ds.reset_search ();
 if (! ds.search (addr)) {
 Serial.print ( "No more addresses. \ N"); // "Адрес більше
немає. \ N")
 ds.reset_search ();
 return;
 }
 Serial.print ( "R =");
```

КБПЗ_2024

```
 for (i = 0; i <8; i ++) {
 Serial.print (addr [i], HEX);
 Serial.print ( "");
 }
 if (OneWire :: crc8 (addr, 7)! = addr [7]) {
 Serial.print ( "CRC is not valid! \ N"); // "CRC не
коректне! \ N")
 return;
 }
 if (addr [0] == 0x10) {
 Serial.print ( "Device is a DS18S20 family device. \ N");
// "Пристрій належить сімейству DS18S20. \ N")
 }
 else if (addr [0] == 0x28) {
 Serial.print ( "Device is a DS18B20 family device. \ N");
// "Пристрій належить сімейству DS18B20. \ N")
66
 }
 else {
 Serial.print ( "Device family is not recognized: 0x"); //
"Сімейство пристрою не розпізнано. \ N")
 Serial.println (addr [0], HEX);
 return;
 }
 ds.reset ();
 ds.select (addr);
 ds.write (0x44,1); // запускаємо конверсію і включаємо
паразитное живлення
 delay (1000); // 750 мілісекунд може вистачити, а може і ні
 // тут можна використовувати ds.depower (), але про це подбає
скидання
 present = ds.reset ();
 ds.select (addr);
 ds.write (0xBE); // зчитуємо scratchpad-пам'ять
 Serial.print ( "P =");
 Serial.print (present, HEX);
 Serial.print ( "");
 for (i = 0; i <9; i ++) {// нам потрібно 9 байтів
 data [i] = ds.read ();
 Serial.print (data [i], HEX);
 Serial.print ( "");
 }
 Serial.print ( "CRC =");
 Serial.print (OneWire :: crc8 (data, 8), HEX);
 Serial.println ();
}
```

КПЗ_2024

```
const int trigPin = 12;
const int echoPin = 11;
void setup() {
 pinMode(trigPin, OUTPUT); // тригер - вихідний пін
 pinMode(echoPin, INPUT); // ехо - вхідний
 Serial.begin(9600); // ініціалізація послід. порта
}
void loop() {
 long distance = getDistance(); // отримуємо дистанцію з
датчика
 Serial.println(distance); // виводимо в послідовний порт
 delay(100);
}
// Визначення дистанції до об’єкта в см
long getDistance() {
 long distacne_cm = getEchoTiming() * 1.7 * 0.01;
 return distacne_cm;
}
// Визначення часу затримки
long getEchoTiming() {
89
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH); // генеруємо імпульс запуску
 delayMicroseconds(10);
 digitalWrite(trigPin, LOW);
 // визначення на піні echoPin довжині рівня HIGH, мксек:
 long duration = pulseIn(echoPin, HIGH);
 return duration;

#include <NewPing.h>
#define TRIGGER_PIN 12 // Arduino pin прив'язаний до тригера
на ультразвуковому датчику..
#define ECHO_PIN 11 // Вивід Arduino прив'язаний до echo виводу на
ультразвуковому датчику.
#define MAX_DISTANCE 200 // Максимальна відстань, з якою ми
хочемо працювати, тобто пінгувати (у см).
//Максимальна відстань датчика оцінюється в 400-500 см.
// Нлаштування контактів і максимальної відстані.
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
void setup() {
 Serial.begin(115200); // Відкриття послідовного порту 115200.
}
void loop() {
 delay(50); // Чекаємо 50 мс між пінгами (близько 20
пінг/с).
```

КОПЗ_2024

```
//29мс має бути найкоротшою затримкою між пінгами.
 ……
}
```

КбПЗ_2024