**Анотація**

Роботу викладено на 62 сторінках, вона містить 5 розділів , 24 ілюстрацій, 9 таблиць та 20 джерел в переліку посилань. Об’єктом розробки став багатофункціональний годинник зворотної дії з режимом детонації на основі мікроконтролера.

Метою даної роботи є розробка універсального годинника який має у своєму складі функцію звичайного відображення часу, будильника та режиму детонації. Ця розробка призначена для використання у домашньому приміщенні в якості креативного будильника та годинника. Також функція детонації активно використовується у страйкболі та тематичних квест кімнатах.

У ході виконання дипломного проекту ми провели пошук та аналіз схожих приладів до нашого годинника. Ми розробили схему електричну принципову, структурну схему, виконали вибір елементної бази, зробили ілюстрацію алгоритму роботи. Провели конструкторсько-технологічні та електричні розрахунки , що підтвердили правильність прийнятих рішень. Обрано матеріал, клас точності та тип ДП. Розроблений програмний код для роботи годинника. На основі коду спроектовано симуляцію в Proteus для відображення роботи приладу.

**Summary**

The work is presented on 62 pages, it contains 5 sections, 24 illustrations, 9 tables and 20 sources in the list of references. The object of development was a multifunctional clock with a detonation mode based on a microcontroller.

The purpose of this work is to develop a universal clock that has the function of the usual display of time, alarm clock and detonation mode. This design is designed for use in the home as a creative alarm clock and clock. The detonation function is also actively used in airsoft and themed quest rooms.

During the diploma project we searched and analyzed similar devices for our watch. We have developed an electrical schematic diagram, a structural diagram, made the choice of the element base, made an illustration of the algorithm. Carried out design and technological and electrical calculations, which confirmed the correctness of the decisions. The material, accuracy class and type of DP are selected. Developed software code for the clock. Based on the code, a simulation in Proteus is designed to display the operation of the device.

**Пояснювальна записка**

**до дипломного проекту**

на тему: **Багатофункціональний годинник зворотної дії з режимом детонації на основі мікроконтролера**

Київ – 2021

**ЗМІСТ**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

Розробив

Перевірив

Реценз.

Н. Контр.

Затвердив

Багатофункціональний годинник зворотної дії з режимом детонації на основі мікроконтролера

Літ.

Аркушів

Перелік скорочень…………………………………………………………….

ВСТУП………………………………………………………………………...

РОЗДІЛ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПОШУК СХОЖИХ АНАЛОГІВ…………………………………………………………………….

1.1 Проблематика виміру часу……………………………………………….

1.2 Аналіз роботи багатофункціональних годинників з режимом детонації..

1.3 Пошук схожих аналогів……………………………………………………

Висновки до розділу……………………………………………………………

РОЗДІЛ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ………………………..

2.1 Розробка структурної схеми………………………………………………

2.2 Обґрунтований вибір елементної бази……………………………………

2.2.1 Вибір МК………………………………………………………………

2.2.2 Вибір регістру зсуву ………………………………………………….

2.2.3 Вибір годинника реального часу ……………………………………

2.2.4 Вибір АЕЖ ……………………………………………………………

2.2.5 Вибір кварцового резонатора ……………………………………….

2.2.6 Вибір роз’єму ………………………………………………………..

2.2.7 Вибір світлодіодних індикаторів ……………………………………

2.2.8 Вибір транзисторів …………………………………………………...

2.2.9 Вибір інших елементів……………………………………………..

2.3 Розробка схеми електричної принципової ……………………………

Висновки до розділу…………………………………………………………

3.1 Вибір типу друкованої плати. 3.2 Вибір матеріалу друкованої плати.

3.3 Вибір класу точності друкованої плати. ……………………………….

3.4. Вибір методу виготовлення друкованої плати ………………………..

3.5 Проектування друкованої плати у середовищі Altium Designer ………

Висновки до розділу ………………………………………………………....

РОЗДІЛ 4. РОЗРАХУНКИ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ КОНСТРУКТОРСЬКИХ РІШЕНЬ……………………………………………

4.1. Конструкторсько-технологічні розрахунки ДП……………………….

4.1.1.Визначення мінімальної ширини друкованого провідника по постійному струму для ланцюгів живлення та землі……………………….

4.1.2.Визначення мінімальної ширини провідника з урахуванням допустимого падіння напруги на ньому ……………………………………

4.1.3.Розрахунок номінального діаметру монтажного отвору ………..

4.1.4. Розрахунок діаметра контактного майданчика …………………..

4.1.5.Розрахунок мінімальної та максимальної ширини провідника ….

4.1.6.Розрахунок мінімальної відстані між провідником та контактним майданчиком …………………………………………………………………...

4.1.7. Розрахунок мінімальної відстані між двома сусідніми провідниками

4.1.8.Розрахунок мінімальної відстані між двома контактними майданчиками………………………………………………………………….

4.2 Електричний розрахунок друкованої плати ……………………………

4.2.1. Визначення падіння напруги на найдовшому провіднику ………

4.2.2. Визначення ємності між двома сусідніми провідниками, які розташовані на одній стороні друкованої плати та мають однакову ширину.

4.2.3.Найбільша взаємна індуктивність двох паралельних друкованих провідників……………………………………………………………………

4.2.4. Визначення потужності втрат ……………………………………..

4.3. Розрахунок надійності друкованої плати ………………………………

4.4 Розрахунок віброміцності друкованої плати …………………………...

Висновки до розділу ………………………………………………………….

РОЗДІЛ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ……………..

5.1 Алгоритм роботи годинника реального часу з режимом детонації. …

5.2 Тестування роботи годинника з режимом детонації. …………………

5.3 Опис лістингу програми. ………………………………………………..

5.3.1 Опис циклу виводу цифр до індикаторів. ………………………..

5.3.2 Опис функцій send\_74HC595\_seg та send\_74HC595\_dig. ………

Висновки до розділу ………………………………………………………..

ВИСНОВКИ…………………………………………………………………..

Список використаної літератури…………………………………………….

Додаток А. Технічне завдання

Додаток Б. Код програми

**СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ**

I2C - Inter-Integrated Circuit

МК - Мікроконтролер

RTC - real time clock

SPI - Serial Peripheral Interface

ТЗ - Технічне завдання

ДП – Друкована плата

ОДП - Одностороння друкована плата

ДДП - Двостороння друкована плата

БДП - Багатошарова друкована плата

КМ - контактний майданчик

АЕЖ - автономний елемент живлення

**Вступ**

В наш час, коли годинники вже зайняли суттєву та важливу частину у нашому житті , їх різноманітні реалізації можна зустріти майже всюди. Зараз вже не зовсім доцільно казати що звичайний годинник ,який виконує свою пряму функцію показувати час , має якісь перспективи, окрім вже всесвітньо відомих торгових марок які заробили собі гарну репутацію. Зараз годинник вбудований у всі телефони, навіть самі гірші.

Тому доцільно стверджувати що цінність годинників має полягати в чомусь іншому, якщо їх все ще купляють та виготовляють у великій кількості, навіть якщо не враховувати іменні бренди, такі як Rolex та Patek Philippe. До цього можна застосувати кілька тез, основними з яких я можу відокремити високу функціональність з дотриманням габаритів , до чого я можу віднести смарт-годинники від компаній Apple, Huawei, Sumsung та інших[1]. Ще не можна не виокремити важливий для багатьох людей критерій зовнішнього вигляду , так як в багатьох випадках годинники використовуються як частина інтер'єру. Тому креативна реалізація скоріш за все буде сильно відокремлюватись з поміж інших. Та останній з основних, але в той же час дуже важливий критерій, це ціна. Тому, для впровадження годинників на ринок з більшою ймовірністю продажів необхідно дотримуватися цих кількох критеріїв.

Тож, з дотриманням цих критеріїв я взяв за мету цієї роботи створення багатофункціонального годинника зворотної дії з режимом “детонації” на основі мікроконтролера. Цей годинник може функціонувати як звичайний годинник, будильник та таймер зворотної дії з режимом детонації. Під режимом “детонації” мається на увазі функція зворотного відліку часу, стандартно 10 секунд, після чого, коли час дорахує до 0 буде світіння усіх світлодіодів та гучний звук від п'єзовипромінювача. Але користувачу не обов'язково чекати до спрацювання віртуального вибуху, так як схема надає змогу втрутитися в процес детонації, що стає можливим за рахунок 4 проводів різного кольору, кожний з яких при спрацюванні режиму детонації виконує різні ролі. Це може бути як миттєвий вибух при “перерізанні” проводу, так і спрацювання деактивації, чи провід може не виконувати ніяких функцій. Режим детонації не тільки робить прилад унікальним за виглядом та функціональністю , він також може бути гарним подарунком. Якщо додати до приладу якісь трубки,наприклад гарно підійдуть картонні трубки з під фольги(Рисунок 1.1) , то це буде нагадувати виконавчий пристрій для вибуху що так часто можна зустріти у фільмах. Але з цим потрібно завжди бути обережним, правоохоронні служби не розуміють жартів. В цілому функція детонації дуже активно використовується в страйкболі, що є досить відома гра , і основна з переваг це що гравці є досить платоспроможними, так як зброя та обладнання коштують досить багато грошей. Ще вона застосовується в квест кімнатах та як реквізити для фільмів. Та загалом це гарний декор.

Протягом виконання проекту поставлено за мету розглянути функції годинника. Також представлена структурна та електрична принципова схеми. Розглянута програма реалізованого проекту. Також розроблене технічне завдання по вибору компонентів, типу матеріалу ДП та класу точності. Для більш чіткого розуміння приведена блок-схема алгоритмів роботи приладу.

**РОЗДІЛ 1. АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ПОШУК СХОЖИХ АНАЛОГІВ.**

**1.1 Проблематика виміру часу.**

Одне з ключових питань яке ставить перед собою розробник, який вирішив створили свій унікальний годинник , це спосіб виміру часу який доступний для розробника. В цьому питанні, яке безпосередньо стосується важливих критеріїв для годинника як точність та ціна, потрібно точно вирішити що використовувати. Також важливо тримати у пам'яті що пристрій має свої умови експлуатації , при недотриманні яких пристрій швидко вийде з ладу. Так як пристрій розрахований на використання в домашніх умовах без контакту з водою, нехтування цим призведе до проблем .

Перший варіант який варто розглянути це внутрішній RC-генератор мікроконтролера. Вони використовуються в AVR-микроконтролерах та інших МК (AVR- відоме сімейство МК, яке випускає компанія Atmal). Як приклад можна згадати ATtiny12, Atmega8 та інших [2]. Цей спосіб є самий простий, але в той же час досить поганий. Головний недолік це досить велика похибка часу. Тобто час може відставати чи переганяти номінальний час в середньому на 15 хвилин за місяць [3]. До переваг можна віднести простоту налаштування та відсутність потреби в розробці, так як RC-генератор вже вбудований в МК.

Другий варіант це годинник реального часу DS1307 від компанії Dallas. Спосіб передачі інформації з МК відбувається за рахунок Inter-Integrated Circuit (I2C), незначна кількість обв’язки та ціна в межах 1-1.5$ [3]. Враховано можливість втрати живлення, при якому починає працювати батарейка. За деякою інформацією похибка роботи DS1307 становить 2 секунди[3] за день. Але цього можна уникнути якщо задати програмну корекцію.

Третій варіант це годинник реального часу DS3231 від компанії Dallas. Це більш нова версія вже відомого DS1307 , її основна перевага це більш висока точність, що становить похибку в 6 секунд [3] за місяць. Але ціна теж складає 2-3$[4].

Четвертий варіант є досить специфічним, але я вирішив його додати. Це Радіомодуль DCF-77 від Siemens. Це модуль, який приймає значення точного часу по радіо хвилям, тому він не потребує налаштування. Цей модуль використовують в багатьох годинниках, але найбільший недолік це ціна. Вона складає 20$[3]. Це в 20 разів більше ніж DS1307.

**1.2 Аналіз роботи багатофункціональних годинників з режимом детонації.**

Такий різновид пристроїв як годинники з режимом детонації у своєму складі мають деякі спільні риси. Хоча всі годинники відрізняються за габаритами , кількістю індикаторів, ціною, та функціями годинника, але можна виділити кілька шаблонів.

Основна спільна риса - це налагодження роботи приладу зосереджене на максимальному використанні кнопок. Під цим мається на увазі ,що якщо користувач хоче максимально використовувати функціональність годинника, наприклад встановити будильник чи таймер, або збільшити чи зменшити час, він матиме розібратись в кнопках. Недоліком виступає потреба користувачу витратити зайвий час щоб розібратися. Перевага це розширена кількість можливостей для налагодження.

Керувати всім цим буде МК, він буде виконувати роль основного блоку , він матиме здійснювати всі основні налагодження та ініціалізацію,також функції такі як світіння світлодіодів чи звучання п'єзовипромінювача при умовах детонації або будильника.

В основному використовується ще модулі годинника реального часу, вони виконують свою роль вимірювання часу. Передача між ним та МК проходить за допомогою використання I2C. Також годинник реального часу має змогу живитися від батарейки при відсутності живлення.

Також суттєву частину зовнішнього вигляду та сприйняття під час використання вносять світлодіоди. В нашому випадку це зелені світлодіоди. При роботі окремі світлодіоди блимають під час вибору різних режимів роботи, інші під час детонації або спрацюванні будильника.

Можна підвести підсумок, що робота годинника з режимом детонації полягає у спрацюванні самої функції детонації, що програмно налаштовано на спрацювання після натискання на відповідну кнопку детонації. Якщо зворотній відлік дорахує до 0 та не буде “деактивації”, то буде гучний звук з п'єзовипромінювача та блимання усіх світлодіодів. Після цього годинник повертається в звичайний режим годинника чи будильника. “Деактивація” - це запрограмована можливість витягнути один з проводів та зупинити відлік, чи витягнути не той провід , після чого буде миттєве спрацювання детонації. Дана функція є актуальною для квест кімнат, які є досить розповсюдженими у Києві, та страйкболі , що є досить реалістичною грою, тому вона потребує реалістичних пристроїв для використання. Інформація виводиться на семисегментні індикатори, зазвичай це два індикатори по 2 розряди , для годин та хвилин. При реалізації режиму детонації замість сегменту хвилин виводяться секунди, в цей час сегмент годин відключений чи працює як хвилини. Також цей пристрій можна використовувати як систему охоронної сигналізації. Якщо запрограмувати його , наприклад кожні 20 хвилин вмикати блимання світлодіодів та звук п'єзовипромінювача , то це буде створювати ілюзію присутності людини у домі.

**1.3 Пошук схожих аналогів.**

Виконаємо аналіз вже існуючих пристроїв, які подібні до нашого. Розглянемо тільки ті годинники, які мають у своєму складі функцію детонації, так як це є відмінною особливістю нашого проекту.

Першим аналогом буде Defusable Clock Game Timer Pro [5] (рисунок 1.1 ) від компанії Nootropic Design. Цей прилад був повністю спроектований для цілей використання у страйкболі, квест кімнатах та реквізитах для відео та фільмів. Цей годинник зібраний на базі МК Atmega328.

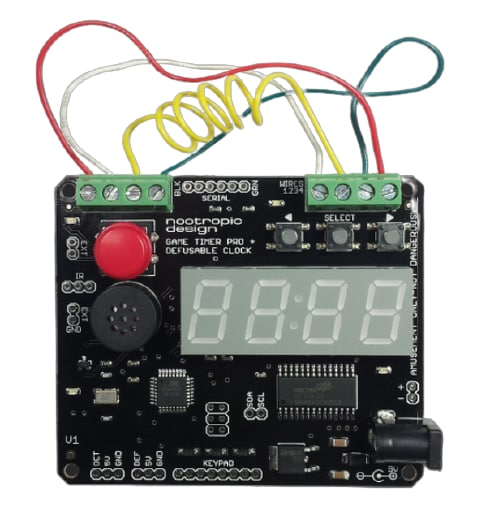
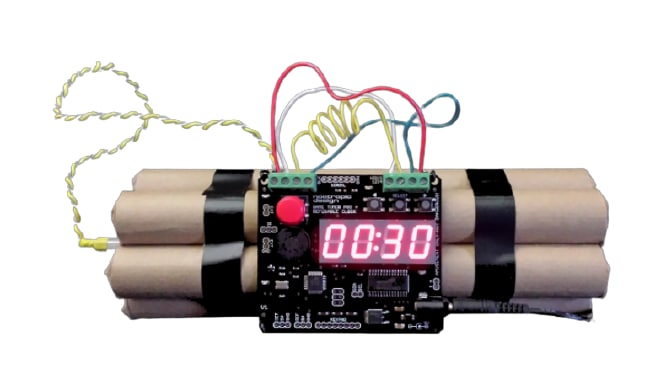
 

Рисунок 1.1 ‒ годинник Defusable Clock Game Timer Pro

Такий варіант має перевагу у великій кількості функцій та додатковій можливості додавання до приладу маленької клавіатури та п'єзосирени . Також він підтримує курування за допомогою мобільного телефону через інфрачервоний датчик на платі. Існує функція встановлення часу годинника, таймера для детонації та будильника.

Будильник має 3 режими :

1) На спрацювання один раз у визначений час;

2) Кожного дня в один час;

3) Відключений;

За показник включення якогось з режимів відповідає засвічення світлодіодів. Якщо світиться один, то перший режим увімкнений, якщо два то другий і якщо зовсім не світять то третій.

Основними недоліками можна вважати ціну. Ціна виробу складає 70$ [6]. Пульт дистанційного керування буде коштувати ще 10$ [6], клавіатура теж коштує 10$ [6]. Ще досить суттєвим недоліком є складність в налаштуванні роботи приладу.

Наступним аналогом є проект від автора wieliksiasty з повною назвою Bomb Watch[7] , який представлений на рисунку 1.2. Основним центром системи є МК Atmega8A-PU. Працює на годиннику реального часу DS1307 та термометрі DS1820 .

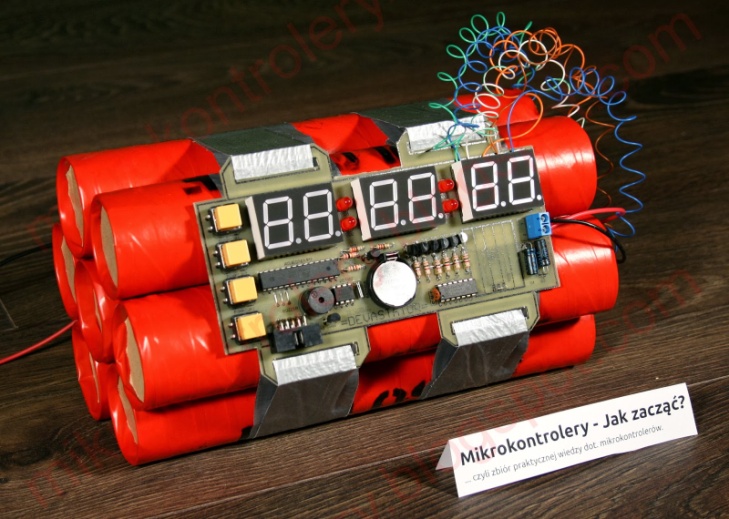
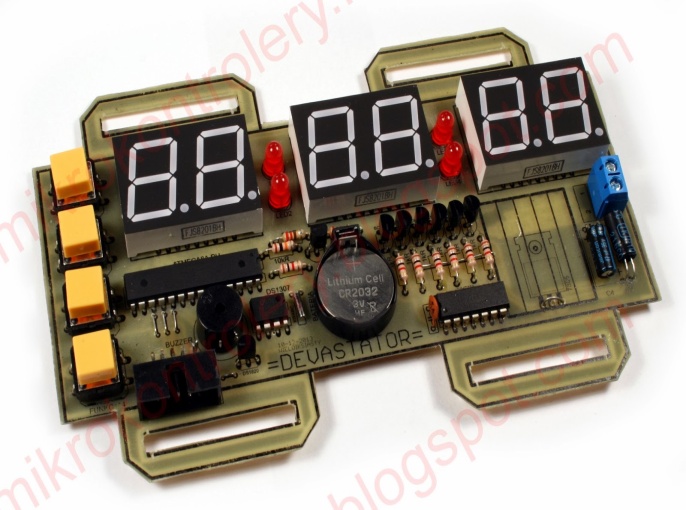


Рисунок 1.2 ‒ годинник Bomb Watch

Перевагою можна вважати наявність великої кількості режимів, таких як режим термометра, звичайного годинника, будильника та режиму детонації. Також не можна не відмітити гарний декор приладу. Досить зручне розташування кнопок та світлодіодів. Кнопки зручні за рахунок великого розміру та розташування з одного краю, що дає змогу для ліворуких людей зручно натискати. Недоліками можна вважати важку реалізацію програмної частини . В даному випадку через велику кількість режимів та малу кількість кнопок користувачу може знадобитися багато часу щоб розібратися та налаштувати все під себе. Також за недолік можна вважати великі габаритні розміри пристрою.

Останній аналог це проект від автора Igoryosha під назвою Detonation Clock[8]. Побудована на основі мікроконтролера Atmega8 з використанням годинника реального часу DS1307. В його складі є також 4 кнопки. Дві з яких відповідають за налагодження годин та хвилин, а інші дві для режимів детонації та годинника. Режими будильника схожі за принципом роботи з першим аналогом і мають в своєму складі 3 режими. Detonation Clock відрізняється від минулого аналога кількістю семисегментників. В цьому випадку в годиннику не передбачений сегмент для секунд. На рисунку 1.3 зображений годинник Detonation Clock.

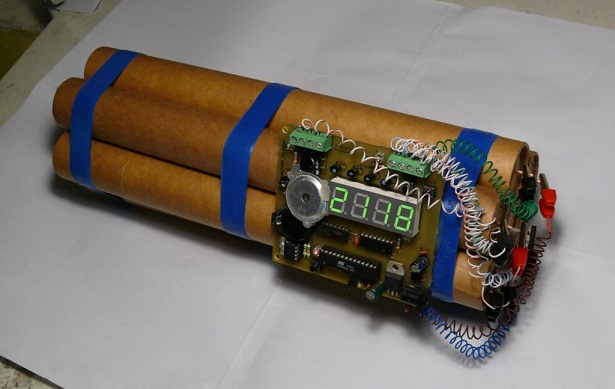
****

Рисунок 1.3 ‒ годинник Detonation Clock

Перевагою можна вважати зручний спосіб налаштування режимів та способів програмування. До того ж він досить невеликого розміру. Можна відзначити можливість налаштування будильника в три різних режими з виводом на світлодіоди. Основним недоліком я можу вважати незручне розташування кнопок. Так як розташування кнопок здійснено по різним кутам схеми, то це може призвести до складностей при налаштуванні годинника та його режимів.

**Висновки до розділу**

В цьому розділі розглянуто проблематику виміру часу, її недоліки та переваги для різних варіантів. Проаналізовано принципи роботи годинників з режимом детонації. Можна стверджувати, що даний прилад досить часто використовується, так як має необхідні функції та вигляд для сучасних розважальних заходів. Розглянуті інші доступні рішення аналогів від компаній та авторів на ринку. Розглянуті їх сильні та слабкі сторони. Сучасні існуючі пропозиції годинників з режимом детонації мають декілька спільних проблем, основна з яких це важкість для користувача розібратись в роботі всіх режимів, так як на схемах присутні в основному 4 кнопки та не менше трьох режимів, що ускладнює налаштування кожного з них. Якщо одна кнопка виконувала одну функцію в одному режимі, вона може виконувати зовсім іншу функцію в іншому режимі.

Найближчими аналогами за функціональністю та принципом роботи , які найближче відповідають бажаним вимогам можна вважати годинник від компанії Nootropic Design на МК Atmega328 та проект Detonation Clock на Atmega8. Обидва ці варіанти мають невеликі габарити та функції які найбільш близько стосуються годинника з режимом детонації. Але недоліком все ще являється важкість в налаштуванні. Detonation Clock також має недолік, який пов’язаний з незручним розташуванням кнопок. Тобто необачне розташування, вибір розміру та назви кнопки може призвести до складнощів при налаштуванні, що призведе до зайвих проблем. Користувач має інтуїтивно розуміти як працює та чи інша кнопка.

Тому є сенс в створенні приладу з зручним розташуванням кнопок та зрозумілим для кожного користувача налаштуванням програмної частини для режимів роботи годинника. Код має бути зрозумілим для користувача з незначними знаннями мови програмування для подальшого удосконалення чи зміни проекту для своїх цілей.

**Розділ 2. СХЕМОТЕХНІЧНЕ ПРОЕКТУВАННЯ**

Відповідно до описаного в попередньому розділі вимог до пристрою, в цьому розділі створено структурну схему та на її основі розроблено електричну принципову схему. Ці виконані дії мають відповідати вимогам ТЗ.

**2.1 Розробка структурної схеми**

Згідно з вимогами, представленими у ТЗ, створено структурну схему годинника з режимом детонації(рисунок 2.1)

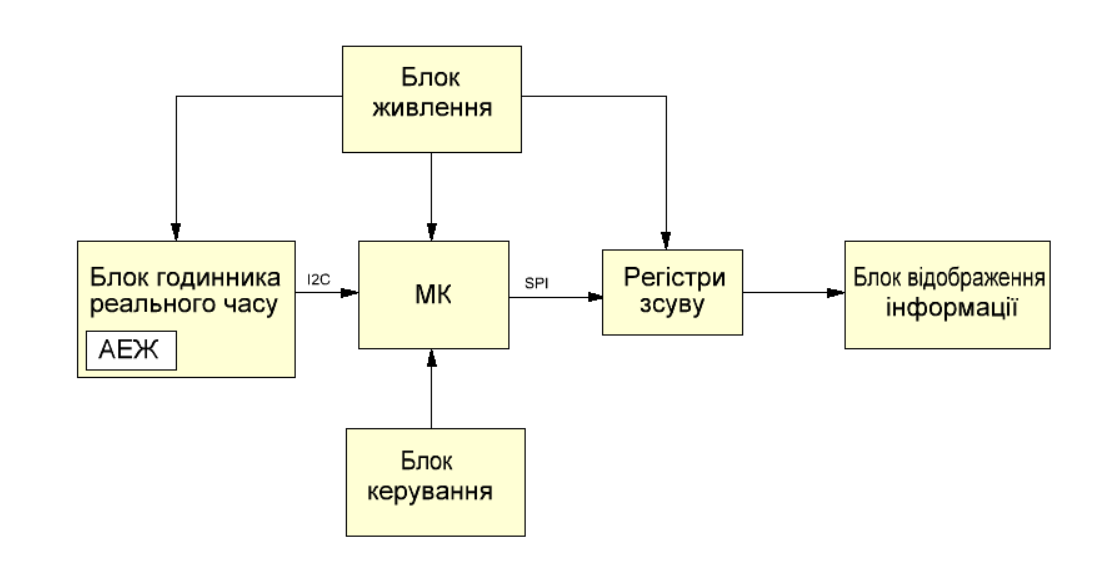


Рисунок 2.1 ‒ структурна схема пристрою

Структурна схема даного годинника на основі мікроконтролера зібрана з наступних блоків:

1) МК(Мікроконтролер).

2)Блок годинника реального часу.

3)Блок відображення інформації.

4)Блок керування.

5)Блок живлення.

6)Регістри зсуву.

**МК.** Отримує інформацію для налагодження режимів роботи з блоку керування. Приймає значення з годинника реального часу. Виводить необхідні данні. Підключений до джерела живлення та живиться від напруги 5В.

**Блок годинника реального часу.** Може відраховувати години, хвилини, секунди. Передача відбувається через I2C. Має змогу при падінні напруги нижче відповідного значення перемикатися на АЕЖ(автономний елемент живлення - це акумуляторна батарейка). Напруга живлення складає 5В.

**Блок відображення інформації.** Здійснює відображення відповідного значення часу. Вивід відбувається на семисегментні індикатори з використанням регістру зсуву. Також відображення роботи окремих режимів відтворюється на світлодіодах та звуці на п'єзовипромінювачі, спільно з роботою годинника.

**Блок керування.** Використовуючи кнопки, які під'єднані до МК , ми маємо можливість налаштовувати час чи режими роботи.

**Блок живлення.** Відбувається перетворення напруги від мережі 220 В в постійну напругу 5 В, так як для живлення інших блоків нам потрібно напруга 5 В.

**Регістри зсуву.** Збільшую кількість виходів МК з використаних трьох до восьми. Передача інформації відбувається через Serial Peripheral Interface (SPI) від МК до регістрів зсуву. У схемі є проміжним блоком між мікроконтролером та семисегментний індикатором.

**2.2 Обґрунтований вибір елементної бази**

До того як ми перейдемо до створення схеми, нам потрібно відібрати найкращі для нас складові нашого пристрою. Головним чином наш вибір подіє на важливі параметри, такі як ціна приладу, напруга живлення, точність та інші.

Далі представлений перелік наших електронних компонентів:

• Мікроконтролер

• Регістри зсуву

• Годинник реального часу

• АЕЖ

• Кварцовий резонатор

• Роз’єм

• Семисегментний індикатор

• Перемикач 4 позиції

• Транзистори

• Резистори

• Конденсатори

• Кнопки

• П'єзовипромінювач

• Світлодіоди

**2.2.1 Вибір МК**

Вибір мікроконтролера є ключовим вибором та є основою нашої розробки. Перше, на що ми маємо звернути увагу, це підтримка передачі по SPI та I2C, це досить важливо, тому що блок годинника реального часу підключений по I2C, та блок відображення по SPI. Наступним важливим критерієм є наявність великої кількості інформації у відкритому доступі та присутність вже існуючих приладів з тієї ж галузі що і наш проект . Також це вартість та доступність на ринку України. Та напруга живлення 5В. До всіх ціх цілей гарно підходить МК з сімейства AVR. Вибраний саме МК Atmega8, так як він відповідає нашим потребам в передачі по SPI та I2C, він досить розповсюджений на ринку та головне, має перевагу у великій кількості розроблених проектів у відкритому доступі у галузі яка пов’язана з моїм проектом.

Мікроконтролер Atmega8 в корпусі TQFP32(Рисунок 2.2) має наступні параметри[9]:

• Тактова частота 0-16 МГц

• Напруга живлення від 4.5 В до 5.5 В

• Можливість обробки зовнішніх і внутрішніх переривань

• Оперативна пам'ять (ОЗУ) 1 кб

• Обсяг флеш пам'яті 512 б

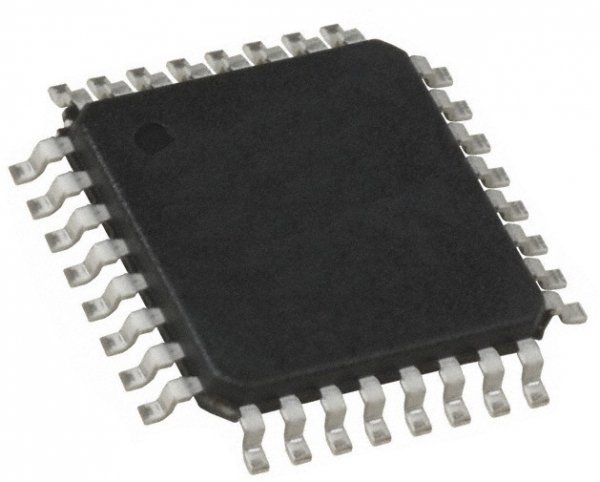


Рисунок 2.2 ‒ Мікроконтролер Atmega8 в корпусі TQFP32

**2.2.2 Вибір регістру зсуву**

У даному випадку нам потрібно вибрати регістр зсуву, тому що кількість виходів МК під світлодіодну індикацію буде велике, та набагато вигідніше використовувати замість 8 виходів МК всього 3. В такій ситуації з використанням великої кількості виходів на МК це досить корисно, бо досить швидко може настати момент коли виходи закінчаться, і варіант купляти МК з більшою кількістю виходів досить абсурдне при дешевій альтернативі регістру зсуву. До вимог можна віднести це передача через SPI , та напруга живлення 5В. Відповідно до вже існуючих варіантів на ринку ми зупинилися на 74HC595, так як він є основним представником регістрів зсуву на ринку, та має гарну репутацію, дешеву ціну, та є восьмирозрядним регістром зсуву з послідовним вводом інформації по SPI.

Регістру зсуву 74HC595 в корпусі SO16 (Рисунок 2.3) має наступні параметри[10]:

• Робоча частота до 100 МГц

• Працює по SPI

• Діапазон робочих температур -40°C …+ 125°C

• Напруга живлення від 2 В до 6 В

• 8 виходів які можна використовувати



Рисунок 2.3 ‒ Регістру зсуву 74HC595 в корпусі SO16

**2.2.3 Вибір годинника реального часу**

Оскільки дуже важливим є годинник реального часу, то наступним оберемо його. Порівняємо три подібних за функціями один до одного RTC за допомогою матриці параметрів.

Таблиця 2.1 – Характеристики годинників реального часу.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Серія | Напруга живлення,В | Напруга живлення батареї,В | Макс температура, °C | Ціна,  грн/шт |
| DS1307 | 4.5 В - 5.5 В | 2 В - 3.5 В | 70°C | 29 |
| M41T56M6E | 4.5 В - 5.5 В | 2.5 В - 3.5 В | 85°C | 40 |
| DS3231 | 2.7 В - 5.3 В | 2.3 В - 5.3 В | 85°C | 70 |
| Ваговий коеф. | 0.25 | 0.1 | 0.25 | 0.4 |

Після вибору значень вагових коефіцієнтів необхідно перевірити щоб сума всіх коефіцієнтів була рівна 1.

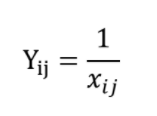
Оскільки характеристики усіх аналогів досить близькі один до одного, було прийняте рішення віддати ціні більше уваги, так як від неї залежить повна ціна нашого пристрою. Після того як ми склали таблицю 2.2, ми маємо скласти матрицю X відповідно до таблиці 2.2

Таблиця 2.2 – Матриця Х

|  |  |  |  |
| --- | --- | --- | --- |
| 5.5 | 3.5 | 70 | 29 |
| 5.5 | 3.5 | 85 | 80 |
| 5.3 | 5.3 | 85 | 45 |

X =

Наступним кроком буде задіяти формулу 2.1 . Після того як ми проаналізували матрицю Х , нам слід її перетворили таким чином, щоб найбільшому значення у матриці для відповідного параметра відповідала найкраща якість модуля.

 (2.1)

де i = 1,n – кількість обраних ІС;

j = 1,m – кількість параметрів ІС.

Виходячи з цього матриці Y буде виглядати так :

Таблиця 2.3 – Матриця Y

|  |  |  |  |
| --- | --- | --- | --- |
| 0.181 | 0.28 | 0.01 | 0.034 |
| 0.181 | 0.28 | 0.011 | 0.0125 |
| 0.188 | 0.188 | 0.011 | 0.022 |

Y =

Наступним кроком ми маємо перетворити матрицю Y у матрицю нормованих параметрів , це буде матриця A. Нормування параметрів виконуємо за наступною формулою:

(2.2)

maxYij — максимальний елемент в стовпці Y;

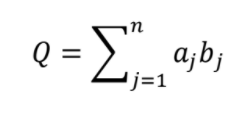
Yij – поточне значення елементу в стовпці Y.

Таблиця 2.4 – Матриця A

|  |  |  |  |
| --- | --- | --- | --- |
| 0.03 | 0 | 0.9 | 0 |
| 0.03 | 0 | 0 | 0.67 |
| 0 | 0.32 | 0 | 0.49 |

A =

Далі останнім кроком це необхідно ввести оціночну функцію Q, яка вираховується за наступною формулою:

 (2.3)

де bj - це ваговий коефіцієнт, aj - це значення з матриці A ;

Q1 = 0.03\*0.25 + 0\*0.1 + 0.9\*0.25 + 0\*0.4 = 0.224

Q2 = 0.03\*0.25 + 0\*0.1 + 0\*0.25 + 0.67\*0.4 = 0.275

Q3 = 0\*0.25 + 0.32\*0.1 + 0\*0.25 + 0.49\*0.4 = 0.228

Виходячи з результатів розрахунків найкоректнішим варіантом для вибору буде та серія, що має найменше значення Q. Отриманим варіантом є годинник реального часу DS1307.

DS1307[11] – досить відомий RTC, в цілому це модуль, який рахує час. Це можуть бути секунди, хвилини, години чи навіть день та місяць. Напруга живлення становить 5В. Передача відбувається за рахунок I2C. Діапазон робочих температур -40°C …+ 85°C.

Годинник реального часу DS1307 в корпусі SOIC8 (Рисунок 2.4) має наступний вигляд:



Рисунок 2.4 ‒ Годинник реального часу DS1307 в корпусі SOIC8

**2.2.4 Вибір АЕЖ**

ДоDS1307 треба підключити акумуляторну батарейку , так як годинник реального часу має вбудовану функцію спостереження за живленням , відповідно до якої під час перебоїв живлення DS1307 перемикається на живлення від акумуляторної батарейки та продовжую роботу. Одна з потреб які поставлені в документації можна віднести напругу в межах 2 В – 3.5 В. Це також має бути Літієва акумуляторна батарейка з ємністю більше 48 мА/ч. По цим вимогам гарно підходить батарейка CR2032, котрий має наступні характеристики[12]:

• Номінальна напруга 3 В

• Номінальна ємність 210 мА/ч

• Робочий температурний діапазон -30°C…+65°C

**2.2.5 Вибір кварцового резонатора**

DS1307 має виводи для кварцового резонатора з частотою 32.768 кГц. Повинен мати вхідну ємність 12.5 пФ. Тому обираємо MC-206 CRYSTAL 32768 Гц [13]. Всі основні параметри точно відповідають до поставлених вимог.

**2.2.6 Вибір роз’єму**

Далі необхідно вибрати роз’єм. Це має бути DC роз’єм. Наступним вже буде стояти питання ціни та розміру. Після огляду існуючих варіантів я вирішив обрати KLS1-DC-005A-2.5 [14]. Володіє наступними характеристиками:

• Виробник : KLS.

• Габарити 11 х 9 х 14.4 мм.

• Кількість контактів 3.

**2.2.7 Вибір світлодіодних індикаторів**

При виборі світлодіодних індикаторів першою чергою необхідно визначитися буде він зі спільним анодом чи спільним катодом. Згідно з поставленою задачею буде простіше працювати зі спільним катодом. Наступне питання полягаю у наявності чи ні двокрапок між сегментами, що не є бажаним у нашому проекті. Та бажано щоб монтаж був SMD. Тому обрав KCDC04-102[15] від компанії KINGBRIGHT ELECTRONIC в SMD корпусі зі спільним катодом та індикацією зеленого світла.

**2.2.8 Вибір транзисторів**

Далі необхідно обрати транзистор. Спочатку я обрав КТ3102, але вирішив замінити на зарубіжний SMD аналог BC847 . Це біполярний транзистор в корпусі SOT-23. Далі представлені деякі параметри:

• Структура n-p-n.

• Максимальна допустима напруга колектор-емітер : 45 В.

• Максимальна допустима напруга колектор-база : 50 В.

• Максимальна допустима напруга емітер-база : 6 В.

• Температурний діапазон -65..+150°С.

• Гранична частота коефіцієнта передачі струму 150 МГц.

**2.2.9 Вибір інших елементів**

Нам залишилося обрати тільки резистори, конденсатори, кнопки, світлодіоди та п'єзовипромінювач. Почнемо з резисторів. Номінали,які ми використовуємо це 470 Ом, 1 кОм, та 10 кОм. Оскільки струм в схемі не можна вважати значним, тому можна обмежитися потужністю в 0,125 Вт. Виходячи з цього можна підсумувати та вибрати резистор від Yageo в корпусі 0805 з точні 5% та робочою температурою від -55 до +155°С.

Далі ми обираємо світлодіод. В якості світлодіода вибрано LTST-C150GKT [16] від Liteon . Тип корпусу SMD 1206, робоча температура -55℃ - 85℃ , зелений колір випромінення.

В якості кнопки вибрана B3U-1000P [17] від компанії Omron. Це вертикальна SMD кнопка , яка відноситься до кнопок без фіксації. Максимальна робоча напруга становить 12 В.

В якості п'єзовипромінювача ми обрали KPEG242 [18] від компанії Kingstate Electronics Corp. Рівень звуку 70 дБа.

Наступним необхідно обрати перемикач на 4 позиції, гарним варіантом для цього виступає Rocker Actuator DIP Switch SPST [19], так як цей перемикач є досить доступним та не дорогим.

**2.3 Розробка схеми електричної принципової**

Ми розроблюємо схему електричну принципову спираючись на схему структурну, яку ми склали та описали на початку цього розділу. Основний блок, який є центром нашої схеми – це мікроконтролер. Ми обрали МК ATmega8, який живиться від напруги 5 В. Всі інші блоки теж живляться від постійної напруги 5 В, тому ми обрали можливість перетворення вхідної напруги 220 В у стабільну напругу 5 В через блок живлення. Оскільки нам не потрібно використовувати іншої напруги ми не будемо перетворювати її в 3.3 В чи інші величини. На рисунку 2.5 представлений роз’єм для живлення пристрою:

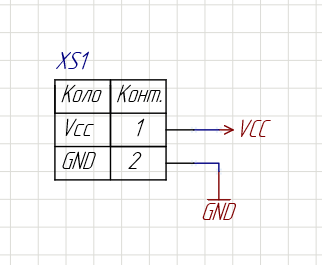


Рисунок 2.5 ‒ Схема підключення роз’єму для живлення пристрою

Наступним важливим кроком буде підключити семисегментні індикатори, з'єднані через регістри зсуву з сегментами індикаторів та їх розрядів задля зменшення використовуваних виводів МК. Перше що ми зробили це обрати виходи МК які будуть підключені до регістрів зсуву 74HC595 , передача даних до якого проходить по SPI. Ми використовуємо два регістри зсуву для керування сегментами індикатора та керування розрядів. Для спрощення ми використовуємо окремі виходи порту B МК для керування сегментами, це PB2,PB4 ,PB5 та виходи порту C для розрядів, це PC0,PC1,PC2. Ми обрали по 3 виходи МК для підключення до входів регістру DS, SHCP, STCP. DS – це вхід даних, SHCP – вхід тактових імпульсів та STCP – вхід замикання даних. Відповідно після підключення до регістру зсуву потрібно подати живлення на регістри, що складає постійну напругу 5В, та додати землю. Після цього ми вже підключили відповідні виходи регістру зсуву до семисегментного індикатора зі спільним катодом. Загальний вигляд блоку відображення інформації можна побачити на рисунку 2.6 :

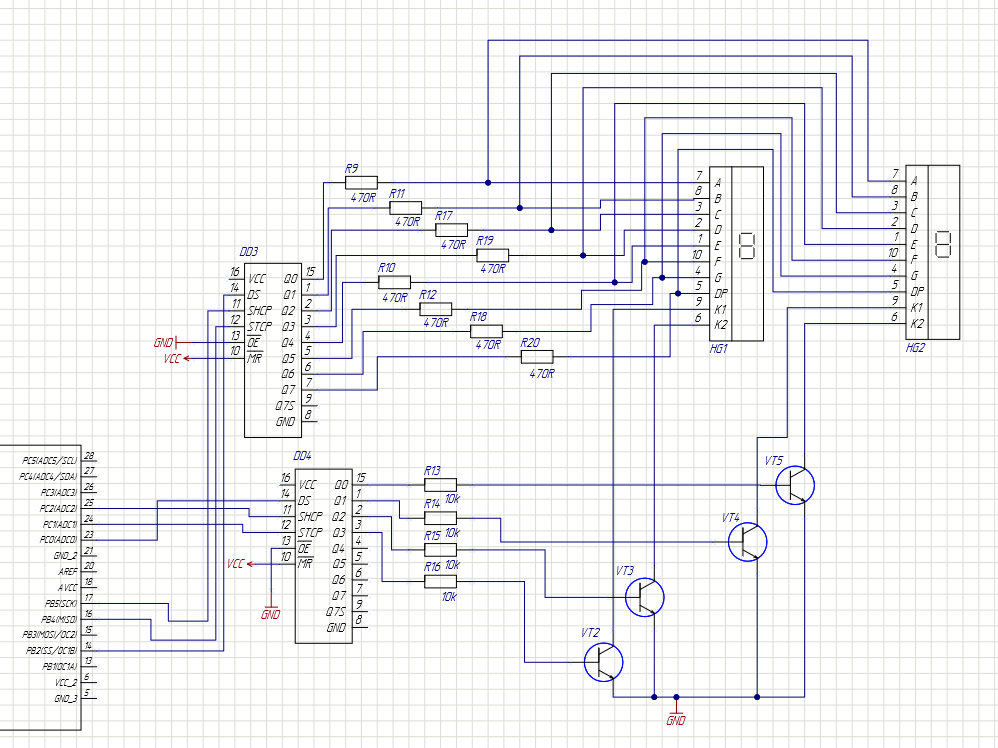


Рисунок 2.6 ‒ Схема підключення світлодіодних індикаторів до МК

Далі необхідно додати годинник реального часу DS1307 який буде рахувати час. Передача інформації між DS1307 та МК відбувається по шині I2C. Для цього ми виділили 2 виходи шини B, це PD2 та PD3. До DS1307 ще підключені АЕЖ та кварцовий резонатор, відповідно до [11].

Оскільки ми не використовуємо вихід SQW/OUT, то необхідності підключати його нема. Також потрібно підключити виходи SDA та SCL відповідно до PD2 та PD3. Схема підключення до МК зображена на рисунку 2.7:

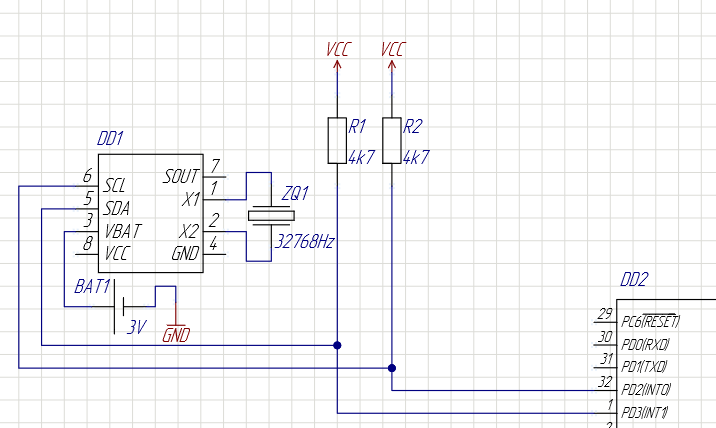


Рисунок 2.7 ‒ Схема підключення до МК

Наступними ми додали кнопки та світлодіоди. Кнопки у нас використовуються для налагодження роботи годинника та контролю за режимами роботи. Світлодіоди виконують в основному дві функції, це відображення режиму роботи при вмиканні режимів будильника та світіння при спрацюванні режиму детонації. Підключення світлодіодів та кнопок зображене на рисунку 2.8.

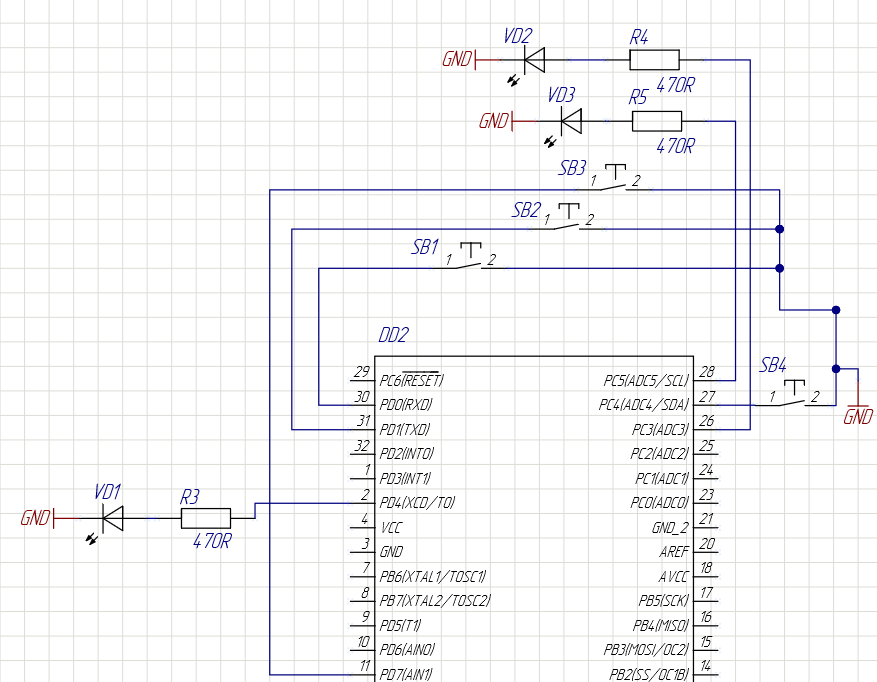
****

Рисунок 2.8 ‒ Схема підключення світлодіодів та кнопок

Далі ми підключаємо п'єзовипромінювач до порту B виходу PB3. Гучність звуку досить суттєва за рахунок схеми посилення напруги на транзисторі VT1 та резисторах R6-R8. На рисунку 2.9 зображена схема підключення п'єзовипромінювача.

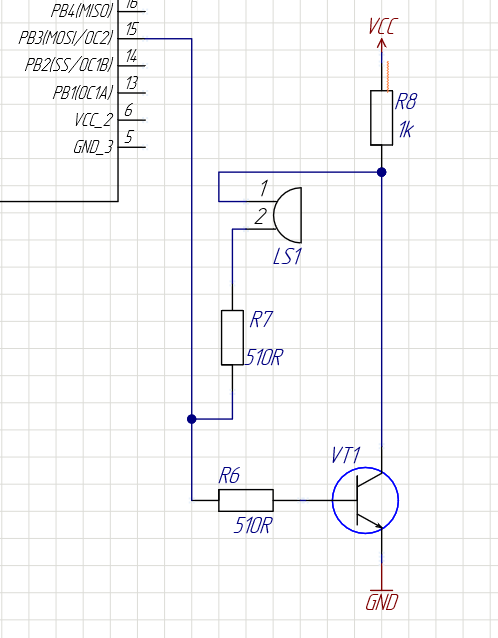
****

Рисунок 2.9 ‒ Схема підключення п'єзовипромінювача

Останнім необхідно додати перемикачі на 4 позиції, хоча використання пристрою можливе без додавання перемикача, але робота буде не повноцінно. На рисунку 2.10 зображена схема підключення перемикача.

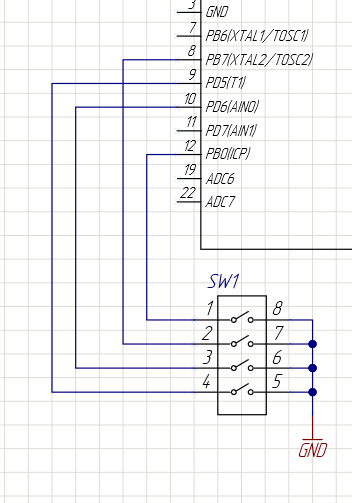
****

Рисунок 2.10 ‒ Схема підключення перемикача на 4 позиції.

**Висновки до розділу**

В цьому розділі створено структурну схему пристрою, розділену на 6 блоків. Описано роботу кожного з блоків та вибрано найкращі компоненти для нашої розробки. Створено схему електричну принципову годинника зворотної дії з режимом детонації. Схему приладу, яку ми створили повністю базується на створених блоках у структурній схемі.

Першим та основним рішенням яке дозволить нам далі розробляти проект це вибір мікроконтролера. Моє рішення вибору ATmega8 базується на достатньо дешевій ціні, великій кількості матеріалів які стосуються самого МК та інформації про створені годинника на базі цього МК. Відповідно до напруги живлення та підтримки SPI та I2C це для нас найкращий варіант. Після вибору МК ми перейшли до вибору ключових елементів інших блоків, таких як блок годинника реального часу в якому ми обрали RTC DS1307 та блок відображення інформації для якого ми обрали регістр зсуву 74HC595 та світлодіодний індикаторKCDC04-102. Встановлено кнопки для керування роботою будильника та режиму детонації . Та також одна із функцій кнопок це налаштовувати час самого годинника. Наступним було додано світлодіоди, що грають важливу роль, так як використовуються при виборі режимів будильника та при спрацюванні режиму детонації. Разом зі спрацюванням світлодіодів при вмиканні режиму детонації для ефекту вмикається шум із п'єзовипромінювача.

**РОЗДІЛ 3. ПРОЕКТУВАННЯ ДРУКОВАНОЇ ПЛАТИ**

**3.1 Вибір типу друкованої плати.**

Друкована плата (ДП) — пластина, виконана з діелектрика (склотекстоліт, текстоліт, гетинакс, ситал і т.д.), на якій або всередині якої сформовано хоча б один шар з провідними доріжками. На ДП монтуються електронні компоненти, які з'єднуються своїми виводами з елементами провідного рисунка паянням, у результаті чого складається електронний модуль — змонтована друкована плата [20].

Друковані плати за конструктивним виконанням розділяються на такі класи:

• односторонні(одношарові) - це ОДП;

• двосторонні(двошарові) - це ДДП;

• багатошарові - це БДП;

Одностороння друкована плата(ОДП) - це пластина, на якій провідники розміщуються лише з одного боку. ОДП прості по виробництву та конструкції, але монтажні і трасувальні здібності цих плат низькі . Можна відзначити невисоку надійність односторонніх друкованих плат та невисоку механічну міцність елементів. Також цей тип плати має слабке кріплення елементів на платі. ОДП переважно використовується для виробництва бюджетної побутової техніки.

Двостороння друкована плата(ДДП) - даний різновид плат має розташування провідників з двох сторін. ДДП мають підвищену щільністю монтажу. Також мають більшу надійність з'єднань порівняно з односторонніми друкованими платами. Такі друковані плати дають можливість реалізувати більш складні схеми та мають широке застосування.

Багатошарова друкована плата(БДП) - коли розведення з'єднань на двосторонній друкованій платі стає занадто складною задачею, використовують БДП. Кількість шарів залежить від складності поставленої перед розробником задачі, відповідно до якої розробник обирає оптимальну кількість шарів. У багатошаровій друкованій платі шари, які знаходяться всередині, призначенні для з’єднування компонентів один з одним. Усі елементи ДП монтуються по обидві сторони ДП. Багатошарова друкована плата характеризується високою надійністю і щільністю монтажу. Також характеризується підвищеною стійкістю до механічних впливів. Гарною перевагою можна вважати зменшення розмірів. До недоліків БДП відносяться важкість розробки, труднощі при виробництві та ремонті. Але головне це висока вартість.

Тому, спираючись на всі недоліки та переваги кожного з класів ДП , та зважаючи на не складну схему розробленого проекту, можна зупинити свій вибір на ДДП. Цей тип є не тільки дуже поширеним , але й має досить гарну ціну.

**3.2 Вибір матеріалу друкованої плати.**

При виробництві ДП застосовуються вітчизняні та імпортні матеріали різних виробників. Матеріали, що застосовуються в якості основи для ДП повинні мати:

• високі електроізоляційні властивості;

• достатню механічну міцність;

• бути стійкими до кліматичних впливів;

Цим вимогам задовольняють електротехнічні матеріали: склотекстоліт, гетинакс, текстоліт, сапфір і ін..

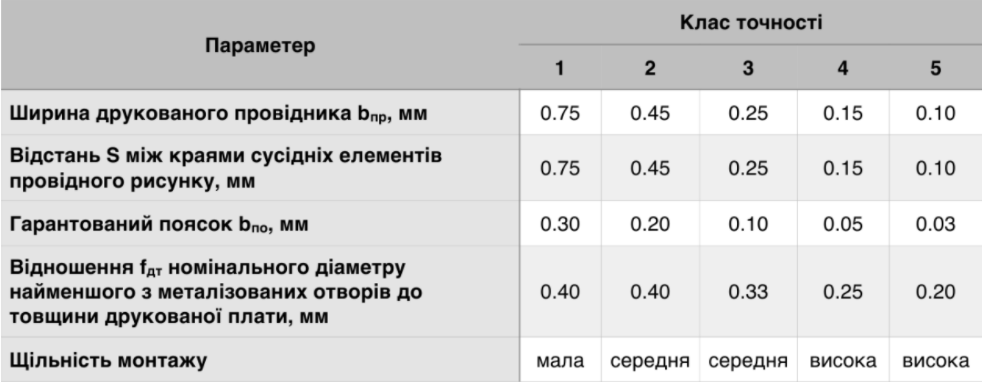
В якості матеріалу ДП доцільно вибирати склотекстоліт, оскільки він має кращі характеристики ніж у гетинаксу та текстоліту. Найбільш розповсюдженими типами можна вважати склотекстоліт FR4 та склотекстоліт СТФ-2. Суттєвої різниці між ними немає, перший тип – імпортний, другий вітчизняний. Але обираємо саме FR-4, так як він імпортний та більш надійний і сучасний, ніж його вітчизняний аналог. FR-4 це композитивний матеріал на основі склотекстоліти. FR-4 дає змогу отримати отвори високої якості, а це важливо для монтажу елементів в отвори. Для реалізації ДДП обрано FR4-2-35-1,5. Даний матеріал є фольгованим склотекстолітом з підвищеною нагрівостійкістю, товщиною 1,5 мм, облицьований з двох сторін мідною електролітичної фольгою товщиною 35 мкм.

**3.3 Вибір класу точності друкованої плати.**

Точність друкованої плати залежить від комплексу технологічних параметрів і з практичної точки зору визначає основні параметри елементів ДП. В першу чергу це відноситься до мінімальної ширини провідників, мінімального зазору між елементами провідного малюнка і до ряду інших параметрів.

Згідно ГОСТ 23751-86 передбачає п’ять класів точності ДП. Вибір класу точності завжди пов’язаний з конкретним виробництвом. Спроба вирішити цю задачу у зворотньому порядку може призвести до того, що проект не буде реалізований. Необхідні граничні значення елементів друкованого монтажу і допустимі похибки наведені в Таблиця 3.1 та Таблиця 3.2 відповідно:

Таблиця 3.1 — Граничні значення основних параметрів ПМ



Таблиця 3.2 — Допустимі похибки виконання елементів ПМ



Виготовлення ДП п’ятого класу точності вимагає застосування унікального високоточного обладнання, спеціальних (як правило, дорогих) матеріалів і навіть створення у виробничих приміщеннях “чистої зони”. Таким вимогам відповідає не кожне виробництво. Однак, Друкована плата невеликого розміру можуть виконуватися по п’ятому класу на обладнанні, що забезпечує виробництво плат четвертого класу.

ДП четвертого класу випускається на високоточному обладнанні, але вимога до матеріалів, обладнання і виробничих приміщень нижче, ніж для п’ятого класу.

ДП третього класу - найбільш поширені, оскільки, з одного боку, забезпечують досить високу щільність трасування і монтажу, а з іншого для їх виробництва достатньо звичайного спеціалізованого обладнання.

Випуск ДП другого і третього класів проводиться на звичайному не спеціалізованому обладнанні. Такі друковані плати призначені для недорогих пристроїв з малою щільністю монтажу.

Отже, спираючись на те, що у нас в схемі використовуються сучасні SMD мікросхеми, які мають досить малі габаритні розміри, ми зупинили свій вибір на 4-тому класі точності, оскільки 4-ий клас точності забезпечує достатню щільність трасування і монтажу з урахуванням необхідних габаритів пристрою.

**3.4. Вибір методу виготовлення друкованої плати**

При виборі необхідного методу виготовлення розглянули наступні методи:

-хімічний субтрактивний метод,

-комбінований позитивний метод,

-метод попарного пресування друкованих плат,

-метод пошарового нарощування,

З огляду на переваги та недоліки кожного з методів, прийнято рішення виготовити ДП комбінованим позитивним методом. Цей метод найкраще підходить для виготовлення ДДП, а також має ряд переваг порівняно з іншими. До того ж позитивний комбінований метод перспективніший, оскільки він дозволяє виготовляти ДП із підвищеною щільністю монтажу, високими електричними параметрами і високою міцністю зчеплення провідників.

**3.5 Проектування друкованої плати у середовищі Altium Designer**

Для створення друкованої плати використовувався САПР Altium Designer. Altium Designer(AD) – це середовище для автоматизованого проектування друкованих плат. Дана програма дає змогу розробляти креслення друкованих плат, схеми електричні принципові та всі інші необхідні складові для виготовлення плати.

Протягом написання дипломного проекту були розглянуті наступні можливості програми:

• Створення бібліотеки компонентів(умовних графічних позначень (УГП) окремих елементів електричної схеми) згідно з ГОСТ 2.702-2011;

• Розробка принципової електричної схеми;

• Трасування друкованої плати;

Для кращого візуального відображення вигляду розробленого проекта є можливість додавати 3D моделі компонентів та наглядно бачити вигляд схеми. AD дозволяють проводити трасування як вручну, так і автоматизовано на основі електричної принципової схеми.

На рисунку 3.1 зображена друкована плата годинника з режимом детонації:

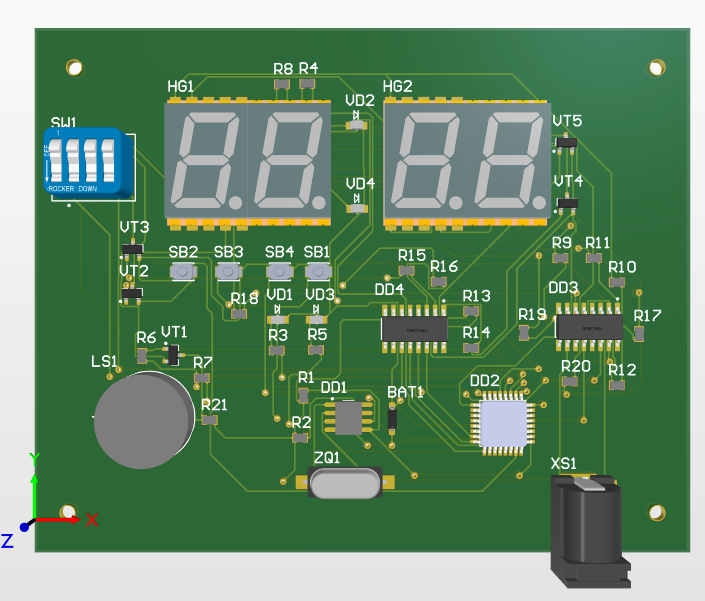


Рис. 3.1 - 3D вигляд ДП.

**Висновки до розділу**

В цьому розділі нами обрано матеріал ДП, метод виготовлення, а також клас точності, розброблено схему у Altium Designer. Наш ДП буде виготовлений комбінованим позитивним методом, оскільки цей метод широко використовується і є гарним рішенням для ДДП. Матеріалом ДП було обрано склотекстоліт FR402-35-1,5 так як він є досить розповсюдженим, та має гарні характеристики. При виборі типу ДП було обрано двосторонню друковану плату, так як він досить поширений та чудово відповідає потребам невисокої складності нашої схеми, також ціна не є високою. Ми розглянули п’ять класів ДП, та після цього обрали четвертий клас точності, оскільки для цього класу точності не потрібно надточне обладнання та дорогі матеріали, та це найбільша точність, яка потрібна для нашого приладу. Створено бібліотеку компонентів та 3D моделі для них. За допомогою цих компонентів ми створили електричну принципову схему та після трасували друковану плату.

**РОЗДІЛ 4. РОЗРАХУНКИ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ КОНСТРУКТОРСЬКИХ РІШЕНЬ**

Конструкторсько-технологічний розрахунок проводиться для підтвердження вибору класу точності виготовлення друкованої плати.

**4.1 Конструкторсько-технологічні розрахунки ДП.**

**4.1.1. Визначення мінімальної ширини друкованого провідника по постійному струму для ланцюгів живлення та землі.**

Мінімальна ширина друкованого провідника на постійному струмі bmin (мм) для ланцюгів живлення та землі визначається формулою 4.1

(4.1)

де *Imax* – пікове значення струму через провідник, А;

*j*доп – допустима щільність струму для ДП, що виготовлені комбінованим позитивним методом, *j*доп = 48 А/мм2 ;

*t*пров – товщина друкованого провідника, мм; Товщина друкованого провідника, що визначається формулою (4.2), мм Друкований провідник виготовлено комбінованим позитивним методом. Згідно методу виготовлення:

*t*пров = *h*ф + *h*гм + *h*хм, (4.2)

де *h*ф – товщина фольги матеріалу, *h*ф = 0,035 мм;

*h*гм – товщина фольги матеріалу *h*гм = 0,055 мм;

*h*хм – товщина фольги матеріалу *h*хм = 0,0065 мм.

Розраховуємо, підставляючи значення в формулу (4.2) та отримаємо:

*t*пров = 0,035 + 0,055+ 0,0065 = 0,097 мм

Для визначення максимального струму *Imax* важливо розглянути ситуацію при якій споживання струму пристроєм буде максимальним. Макимальний струм споживання для RTC DS1307 досить незначний , це 300 мкА, для 74HC595 це 40 мА та МК ще 200 мА . В результаті маємо:

*Imax* = 0.3 + 40 + 40 + 200 = 280.3 мА.

Вже можна розрахувати мінімальну ширину друкованого провідника на постійному струмі для ліній живлення і землі:

мм.

Отримали мінімальну ширину друкованого провідника – 0.06 мм. Отримане значення мінімальної ширини провідника повністю задовольняє четвертий клас точності при = 0,15 мм. Для підвищення надійності та покращення якості живлення компонентів приладу, прийнято рішення збільшити ширину друкованих провідників ліній живлення та землі до 0.3 мм.

**4.1.2. Визначення мінімальної ширини провідника з урахуванням допустимого падіння напруги на ньому:**

Мінімальна ширина провідника з урахуванням допустимого падіння напруги на ньому визначається наступним виразом:

(4.3)

Де ρ–питомий опір провідника, виготовленого комбінованим позитивним методом, ρ=0,0175 .

*L*пров – довжина найдовшого друкованого провідника ДП, *L*пров=0.274 м.

*U*доп – допустиме падіння напруги на друкованому провіднику,

*U*доп = 0,05×*U*жив =0,05×5=0,25

За формулою 4.3:

(м).

Як видно з розрахунків, обрана ширина в 0,15 мм цілком задовольняє вимогам.

**4.1.3. Розрахунок номінального діаметру монтажного отвору**

Номінальний діаметр монтажного отвору визначається за виразом (4.4):

, (4.4)

де dве – діаметр виводу елементів, для якого визначається діаметр монтажного отвору, dве = 0,5 мм;

∆d – нижнє граничне відхилення від номінального діаметру МО, ∆dмо=0,1 мм;

r– різниця між мінімальним діаметром МО та максимальним діаметром виводу елемента, r=0,1…0,2 мм;

Підставивши значення у формулу, маємо:

.

**4.1.4. Розрахунок діаметра контактного майданчика**

Мінімальний діаметр контактного майданчика може бути розрахований за формулою:

, (4.5)

де Dmin1 – мінімальний ефективний діаметр КМ, мм,

hф – товщина фольги, hф = 0,035 мм. Коефіцієнт 1,5hф враховує підтравлювання фольги друкованого провідника у ширину, 0,03 – КМ виготовляють комбінованим позитивним методом.

, (4.6)

де dmax– максимальний діаметр отвору в ДП, мм,

bпо - ширина пояска КМ, bпо =0,05 мм;

δо- похибка розташування центру отвору відносно вузла КС, δо=0,07 мм ,

δкм- похибка розташування центру КМ відносно вузла КС, δкм=0,05 .

Максимальний діаметр просвердленого отвору у друкованій платі розраховується за формулою:

dmax=d+∆d+(0,1…0,15), (4.7)

де d– номінальний діаметр МО, мм,

∆d- допуск на діаметр отвору, ∆d=0,05 мм

Максимальний діаметр КМ:

D max = Dmin+0,02, (4.8)

Для SB5,SB6:

dmax= *d* + *∆d* + (0,1…0,15) = 0,8 + 0,05 + 0,1 = 0,95 мм

*Dmax*=1,3725 + 0,02=1,3925 мм

**4.1.5. Розрахунок мінімальної та максимальної ширини провідника**

Для наступного розрахунку необхідно застосувати формулу:

bmin=, (4.9)

де –мінімальна ширина провідника. Визначаємо з таблиці класів точності (табл.3.1). Для 4-го класу точності .

Отже:

bmin=

Максимальна ширина провідника:

*bmax*= *bmin*+ 0,02, (4.10)

*bmax* = 0,2325 + 0,02 = 0,2525 мм

**4.1.6. Розрахунок мінімальної відстані між провідником та контактним майданчиком**



Рис. 4.1 – Схематичне зображення відстані між провідником і контактним майданчиком

Відстань між провідником і контактним майданчиком може бути розрахована за формулою 4.11

(4.11)

де L0– відстань між центром отвору та центром друкованого

провідника, що є кратним кроку КС, найгірший випадок, L0=1,25 мм.

Dmax - максимальний діаметр КП;

bmax - максимальна ширина провідника;

δкм - похибка зміщення контактного майданчика, δкм=0,05;

δсп - похибка, яка враховує зміщення провідника, δсп=0,05 мм;

**4.1.7 Розрахунок мінімальної відстані між двома сусідніми провідниками**

На Рис.4.2 зображено схематичне розташування двох провідників і

зображені відстані, які використовуються для розрахунку формули (4.12):



Рис. 4.2 – Відстань між двома провідниками

Мінімальна відстань між двома сусідніми провідниками може бути розрахована за формулою:

(4.12)

Так як для трасування друкованої плати необхідно проводити друковані провідники між виводами мікросхем, застосовано координатну сітку із кроком 1 мм, тоді = 1 мм.

Підставивши числа, маємо:

м

Отже, мінімальна відстань між двома друкованими провідниками має складати 0,6475 мм.

**4.1.8 Розрахунок мінімальної відстані між двома контактними майданчиками**



Рис. 4.3 – Схематичне зображення відстані між двома контактними майданчиками

Мінімальна відстань між двома контактними майданчиками може бути розрахована за формулою

, (4.13)

де L01- відстань між центрами сусідніх КП, L01=2,5 мм.

Підставивши числа,маємо:

Отримане значення задовольняє 4-ий клас точності.

**4.2 Електричний розрахунок друкованої плати**

Такі параметри як габаритні розміри, маса, та інші є дуже важливими при розробці електронних пристроїв . Але зі зменшенням розмірів, як самих елементів так і зі збільшенням щільності компонування цих елементів, виникають паразитні ємнісні та індуктивні зв’язки, які в свою чергу можуть впливати на компоненти та призводити до помилкового спрацювання і до некоректної роботи схеми. Тому для запобігання такого роду проблем потрібно обраховувати ці значення і переконатися що вони знаходяться в допустимих межах. Паразитна ємність та індуктивність проявляється у затримці розповсюдження сигналу.

**4.2.1. Визначення падіння напруги на найдовшому провіднику**

Падіння напруги на друкованому провіднику визначається наступною формулою:

(4.14)

Де ρ–питомий опір провідника, виготовленого комбінованим позитивним методом, ρ=0,0175 .

*L*пров – довжина найдовшого друкованого провідника ДП, *L*пров=0.274 мм.

bпр – мінімальна ширина провідника, для 4 класу точності це 0.15.

За формулою 4.14:

В.

Розраховане падіння напруги не перевищує 5% від напруги живлення (Uж = 5 В).

**4.2.2. Визначення ємності між двома сусідніми провідниками, які розташовані на одній стороні друкованої плати та мають однакову ширину.**

Ємність між двома сусідніми провідниками, які розташовані на одній стороні друкованої плати та мають однакову ширину визначаються за наступною формулою:

, (4.15)

де S – відстань між двома паралельними провідниками,

bпр - ширина друкованого провідника, мм

tпр - товщина друкованого провідника, мм

lпр - довжина взаємного перекриття двох паралельних провідників, мм

**4.2.3.Найбільша взаємна індуктивність двох паралельних друкованих провідників**

, (4.16)

де lпр – довжина перекриття паралельних провідників, см

Lо-відстань між осьовими лініями двох паралельних провідників, см

Гн

Отримане значення дозволяє стверджувати, що на працездатність схеми паразитна індуктивність не вплине.

**4.2.4. Визначення потужності втрат**

Визначемо потужність втрат двосторонньої друкованої плати. Потужність

втрат визначається за наступною формулою:

, (4.17)

де *f*нч=1 Гц для низьких частот та *f*вч=32768 Гц для високих частот.

*tg*σ – тангенс кута діелектричних втрат для матеріалу друкованої

плати, tgσ = 0,002 для матеріалу FR4

E – напруга живлення, у нас це E = 5В.

С – ємність друкованої плати, що вираховується за формулою:

, (4.18)

де ε – діелектрична проникність матеріалу, для FR4 ε = 4,5

Sm - площа металізації, мм2

h - товщина ДП, мм

== 30 пФ.

Для низьких частот:

Рпот = .

Для високих частот:

Рпот = .

**4.3. Розрахунок надійності друкованої плати**

Надійність – це одна з основних та найважливіших параметрів електронного пристрою та залежить від якості та кількості елементів , від умов експлуатації та інших причин. В цілому це можливість приладу виконувати всі свої функції та зберігати експлуатаційні показники протягом певного заданого часу .

Найбільш точна кількісна міра надійності кожного конструктивного елементу є його індивідуальне напрацювання на відмову. На практиці ж достатньо повна характеристика надійності - щільність розподілення часу безвідмовної роботи даного типу КЕ f(t) та інтенсивність відмов λ(t). Функції f(t) та λ(t) визначаються експерементально. При цьому період нормальної експлуатації для ІС характеризується високою надійністю. Тоді величини f(t) та λ(t) пов’язані відношенням:

f(t) = (4.19)

Знаючи f(t) та λ(t), можна визначити інші кількісні характеристики надійності протягом часу від 0 до t:

Q(t) = 1 – P(t) = 1 - (4.20)

Важливою характеристикою надійності також є середній час безвідмовної роботи. Cередній час напрацювання на відмову визначається за наступною формулою :

(4.21)

Інтенсивність відмов ЕРЕ є їх вихідною характеристикою надійності, яка залежить від режиму роботи та від зовнішніх чинників і їх тяжкості. До таких чинників відносяться температура, вологість, вібрації та ін. Тоді можна записати вираз:

(4.22)

де λое – інтенсивність відмов компоненту за нормальних увов роботи (н.у. – темп. навколишнього середовища 25 15%);

,,... – поправочні коефіцієнти, що враховують режими роботи та умови експлуатації.

Для врахування впливу режиму роботи та інтенсивності відмов ЕОА коефіцієнт навантаження (4.23) що дорівнює відношенню навантаження в робочому режимі до навантаження в номінальному режимі.

Коефіцієнт навантаження для резисторів:

Для усіх інших резисторів, розраховані Кн наведено у таблиці 4.1.

Таблиця 4.1– Коефіцієнти навантаження для усіх резисторів у схемі

|  |  |  |  |
| --- | --- | --- | --- |
| Резистор | Номінал, Ом | , В |  |
| R13-R16 | 10 kOm | 3.3 В | 0,008712 |
| R3-R5, R9-R12, R17-R20 | 470 Om | 3.3 В |  |
| R1-R2 | 4,7 kOm | 3.3 В |  |
| R21 | 1 kOm | 3.3 В | 0,08712 |
| R7-R8 | 510 Om | 3.3 В | 0.170 |

Визначимо результуючу інтенсивність відмов друкованого вузлу годинника з режимом детонації. Друкований вузол відноситься до наземної апаратури, експлуатується при температурі до Tp = 40°С, інші умови експлуатації нормальні. Вихідні дані для λp - схема принципова, перелік елементів, графіки залежності коефіцієнтів, що враховують зміну температур. В таблиці 4.2 нижче наведені: елементи, коефіцієнти навантаження, кількість елементів, базова інтенсивність відмов, коефіцієнти, що враховують зміну температур.

Таблиця 4.2 – Значення для визначення надійності

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Компонент | N | λ0e  \* 10-6, год-1 | Kн | at | ae | | λ0e \* N \* Kн\* at \* ae \* 10-6 |
| Резистори R13-R16 | 4 | 0.04 | 0,008712 | 0.2 | 10 | | 0.00209 |
| Резистори R3-R5, R9-R12, R17-R20 | 12 | 0.04 |  | 0.2 | 10 | | 0.1728 |
| Резистори R1-R2 | 2 | 0.04 |  | 0.2 | 10 | | 0.00296 |
| Резистор R21 | 1 | 0.04 | 0,08712 | 0.2 | 10 | | 0.00696 |
| Резистори R7-R8 | 2 | 0.04 | 0.170 | 0.2 | 10 | | 0.0272 |
| П'єзоелектричні  прилади | 2 | 0.03 | 1 | 1 | 10 | | 0.6 |
| Регістри зсуву | 2 | 0.02 | 1 | 1 | 10 | | 0.4 |
| Кнопка | 4 | 0.16 | 1 | 1 | 10 | | 6.4 |
| Кнопочний  перемикач | 1 | 0.16 | 1 | 1 | 10 | | 1.6 |
| Світлодіоди | 4 | 0.034 | 1 | 1 | 10 | | 1.36 |
| Семисегментні  індикатори | 2 | 0.42 | 1 | 1 | 10 | | 8.4 |
| Мікросхема  цифрова | 2 | 0.23 | 1 | 1 | 10 | | 4.6 |
| Транзистор  біполярний | 5 | 0.044 | 1 | 1 | 10 | | 2.2 |
| Друкована плата | 1 | 0.002 | 1 | 1 | 10 | | 0.02 |
| Контакти роз’ємів | 3 | 0.015 | 1 | 1 | 10 | | 0.45 |
| Пайка виводу | 96 | 0.00007 | 1 | 1 | 10 | | 0.0672 |
|  | | | Сумма | | |  | |

Де aе – поправочний коефіцієнт зовнішніх впливів(для наземної стаціонарної апаратури ае = 10);

at – поправочний температурний коефіцієнт.

Результуюча інтенсивність відмов дорівнює сумі інтенсивностей відмов компонентів:

(4.23)

Середній час напрацювання до першої відмови:

(4.24)

38008 год ≈ 4.5 років

Ймовірність безвідмовної роботи протягом року розраховується формулою 4.25 . У одному році 8760 годин. Тоді, маємо

= (4.25)

Ймовірність відмов протягом року:

= 1 - = 0.21 (4.26)

Графік залежності безвідмовної роботи ДВ та ймовірність відмов ДВ від часу(годин) представлені на рисунку 4.4, де:

Q(t) – ймовірність відмов,

Р(t) – ймовірність безвідмовної роботи.

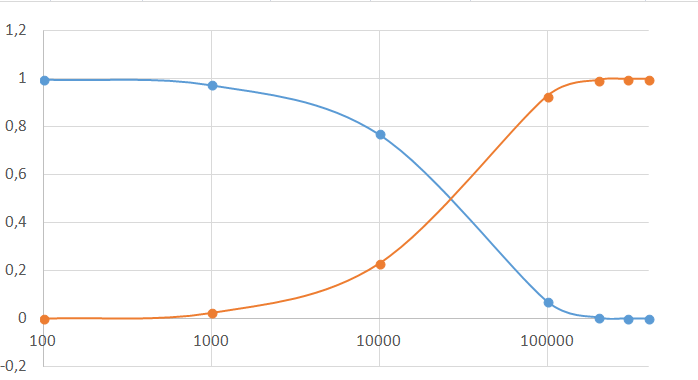


Рисунок 4.4 – Ймовірність відмови

**4.4 Розрахунок віброміцності друкованої плати**

Для виконання розрахунків віброміцності необхідно спочатку визначити масу усіх елементів та самої плати.

Таблиця 4.3– Маси елементів у ДП.

|  |  |  |  |
| --- | --- | --- | --- |
| Компонент | Кількість | Маса, г | Заг. Маса, г |
| Резистор 0805 | 20 | 0.03 | 0.6 |
| Регістр зсуву 74HC595 | 2 | 0.2 | 0.4 |
| RTC DS1307 | 1 | 0.8 | 0.8 |
| ІС AtMega8A-AU | 1 | 3.7 | 3.7 |
| Кнопка B3U-1000P | 4 | 0.12 | 0.48 |
| Світлодіоди LTST-C150GKT | 4 | 0.038 | 0.152 |
| Семисегментні індикатори KCDC04-102 | 2 | 3 | 6 |
| Транзистор BC847 | 5 | 0.018 | 0.09 |
| АЕЖ CR2032 | 1 | 2.8 | 2.8 |
| Кварцовий резонатор | 1 | 0.3 | 0.3 |
| Роз’єм KLS1-DC-005A-2.5 | 1 | 1.7 | 1.7 |
| Всього | | | 17.02 |

Сумарна маса електричних компонентів друкованої плати становить 17 г.

Друкована плата має наступні розміри a (мм) x b (мм) x h (мм) =

100 х 80 х 1,5. Маса елементів на друкованій платі наведена в таблиці 4.3.

Розмір ДП: a \* b \* δ = 100 \* 80 \* 1.5 мм.

Густина склотекстоліту FR4: ρ = 2,05 гр/м3

Маса ДП розрахововується за формулою:

m = ρ \* a \* b \* δ *=* 2,05 \*10\* 8,0 \* 0,15 = 23,06 г (4.27)

Параметри склотекстоліту FR4:

- часовий опір σтимч = 105 МПа;

- модуль Юнга E = 3.02 \*1010 Па;

- коефіцієнт Пуассона μ = 0.22;

- показник затухання ε = 0.06;

- питома щільність ν = 2.05 \* 104 Н/м3;

- коефіцієнт запасу міцності n1 = 2.

Обраний тип закріплення друкованої плати - спирання на 4 сторони. (Рис 4.5)



Рис. 4.5 Тип закріплення ДП.

Далі наведено розрахунки методу закріплення на 4 сторони:

(4.28)

= 0.76

У формулі 4.29 наведено розрахунку коефіцієнта α, що враховує обраний тип закріплення друкованої плати:

(4.29)

= 27.38

У формулі 4.30 наведено формулу розрахунку циліндричної жорсткості D:

(4.30)

Hм.

У формулі 4.31 наведено формулу власної частоти коливань друкованої плати *f*власн:

(4.31)

*f*власн > 250 Гц – що означає конструкція абсолютно жорстка. Тому розрахунок динамічного прогину можна не розраховувати.

**Висновки до розділу**

В цьому розділі ми провели розрахунки які підтвердили коректність прийнятих конструкторських рішень. Розрахували значення паразитної ємності та індуктивності, по результатам розрахунків яких ці величини не впливають на коректну роботу схеми так як вони лежать у допустимих межах. Розраховане значення надійності ДП. Так як розрахунки були спрощені до розрахунку надійності при гіршому варіанті, через це розрахована надійність є трохи заниженою. Середній час напрацювання до першої відмови склад 38000 годин, що є приблизно 4.5 років. Ймовірність безвідмовної роботи протягом року склала 0,79 , а ймовірність відмов протягом року склала 0,21. При розрахунку віброміцності ми виявили що друкована плата є абсолютно жорсткою, так як власна частота *f*власн = .

**Розділ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Для керування роботою пристрою використовується МК Atmega8. МК приймає данні часу від годинника реального часу DS1307 через інтерфейс I2C. Для налаштування роботи режимів використовуються кнопки, які підключені до МК. Інформація проходить до регістрів зсуву через SPI та виводиться на семисегментні індикатори. Для розробки програмної частини використовується Atmel Studio. Реалізацію роботи пристрою вирішено продемонструвати в середовищі Proteus.

**5.1 Алгоритм роботи годинника реального часу з режимом детонації.**

На основі поставленого завдання створено алгоритм роботи годинника з режимом детонації з послідовно продемонстрованими кроками роботи пристрою. Алгоритм продемонстрована на рисунку 5.1 :

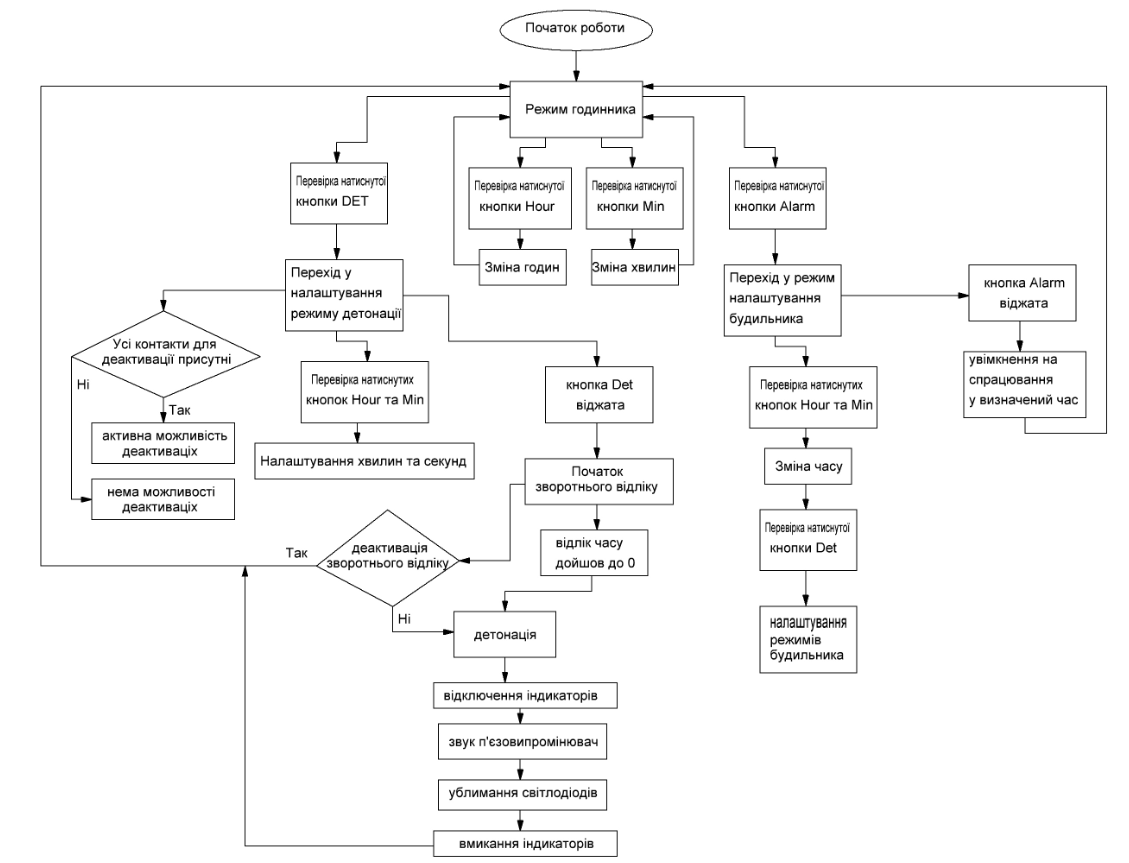


Рисунок 5.1 – Алгоритм роботи годинника з режимом детонації

**5.2 Тестування роботи годинника з режимом детонації.**

Для тестування роботи необхідно обрати середовище яке дасть змогу продемонструвати усі можливості проектованого пристрою. Так як нами обраний мікроконтролер Atmega8, найкращим середовищем для демонстрації роботи можна вважати Proteus. Це середовище не тільки дає нам змогу створити електричну принципову схему з вже існуючих компонентів у бібліотеці середовища, але й має у своєму складі можливість прошити віртуальний МК з програмою та перевірили працездатність пристрою. Так як обраною темою проекту є годинник, досить зручним та зрозумілим способом відображення роботи буде демонстрація у Proteus. На рішення обрати Proteus ще досить суттєво вплинула велика кількість відео уроків та створених проектів на цій платформі, тому розібратися в роботі середовища було досить легко, навіть зважаючи на те що ми не працювали в ньому під час навчання. Також до тестування роботи пристрою додано відеоматеріал, який повною мірою демонструю можливості розробленого проекта []. Схема зібраного годинника реального часу з режимом детонації в Proteus продемонстрована на рисунку 5.2, відеоматеріал за посиланням [].

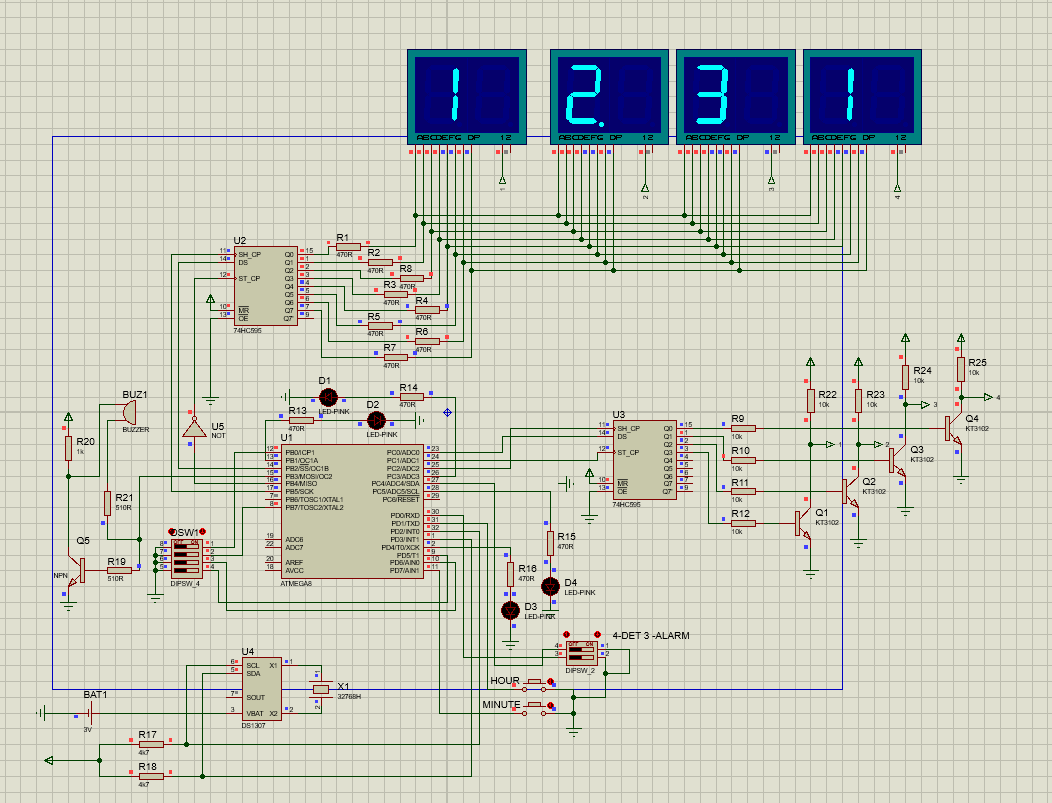


Рис 5.2 Схема годинника реального часу з режимом детонації в Proteus

Так як у версії Proteus яку я використовую є деякі проблеми з відображенням інформації на семисегментні індикатори, прийняті конструкторські рішення для усунення цих проблем у Proteus(Рис 5.2). При створенні реального прототипу ці моменти не будуть впливати на роботу приладу, тому створена електрична принципова схему у Altium Designer є повністю коректною та правильною.

**5.3 Опис лістингу програми.**

Перш за все програма написана на мові програмування C з використанням звичайних бібліотек для МК Атмега8 як mega8.h, delay.h та stdlib.h. В якості бібліотеки ініціалізації для RTC використовується бібліотека ds1307.h та для передачі інформації по інтерфейсу бібліотека i2c.h.

**5.3.1 Опис циклу виводу цифр до індикаторів.**

Цикл відображення інформації виконаний за допомогою функції switch. Використовує в свою черту аргумент count , який після кожного обходу циклу збільшую своє значення на 1 та якщо він більше 4 то обнуляється.

if (count>=4) count=0;

switch (count)

{

case 0: if(DIGIT\_IS\_ON(DIGIT\_1)) // Це макрос, який відповідає за управління індикацією відповідного розряду

{

send\_74HC595\_dig(1<<NUM4); // Підключення необхідного індикатора. Функція описана у розділі 5.3.2.

send\_74HC595\_seg(number[\*digit\_12/10]); break; // Функція відповідає за формування цифри. number[] це масив цифр від 0 до 9, використовується для відображення значення на індикатори. digit\_12 це змінна в якій лежить значення годин. Після ділення на 10 значення заноситься в старший розряд .

}

case 1: if(DIGIT\_IS\_ON(DIGIT\_2))

{

send\_74HC595\_dig(1<<NUM3); // Підключення необхідного індикатора

if(point) send\_74HC595\_seg((number[\*digit\_12%10])|(1<<SEG\_DP)); //Запалити точку якщо змінна point один. %10 це означає що значення годин буде ділитися з остачею на 10, та заносилися у молодший розряд годин.

else send\_74HC595\_seg(number[\*digit\_12%10]); break;

}

case 2: if(DIGIT\_IS\_ON(DIGIT\_3)) // Наступні дві операції збігаються з минулими двома та виконують ідентичну роботу , але для розрядів хвилин.

{

send\_74HC595\_dig(1<<NUM2);

send\_74HC595\_seg(number[\*digit\_34/10]); break;

}

case 3: if(DIGIT\_IS\_ON(DIGIT\_4))

{

send\_74HC595\_dig(1<<NUM1);

send\_74HC595\_seg(number[\*digit\_34%10]); break;

}

}

count++;

**5.3.2 Опис функцій send\_74HC595\_seg та send\_74HC595\_dig.**

Обидві ці функції працюють по одному принципу. Вони створені для можливості передачі інформації від МК до індикаторів через регістр зсуву 74HC595. Для розуміння роботи функції необхідно розуміти як працює регістр зсуву 74HC595 та які ролі виконує кожна з його ніжок на вихід та вхід. На рисунку 5.3 зображений вигляд регістру зсуву з позначенням виводів мікросхеми:

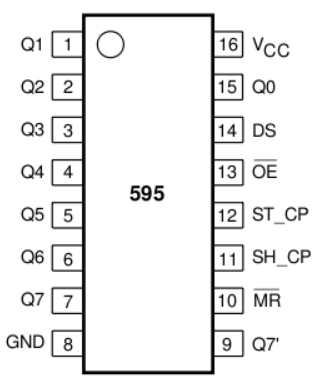


Рис 5.3 Регістр зсуву 74HC595

Q0 ... Q7 - виходи які будуть підключені до індикаторів та якими ми управляємо для виводу правильних значень.

MR - скидання регістра.

SH\_CP - вхід для тактових імпульсів.

ST\_CP - вхід «защелкиваючий» данних.

OE - вхід переводить виходи з HI-Z в робочий стан.

DS - вхід даних.

Для скидання регістру необхідно на MR подати низький рівень , але так як нам не потрібне скидання ми подаємо на цей вхід високий рівень. Якщо подати на вхід OE високий рівень то виходи регістру зсуву будуть у  HI-Z стані, але нам це не потрібно і ми подаємо логічний нуль після чого виходи будуть у робочому стані. Відповідно для кращого розуміння простіше розглянути рисунок 5.4.

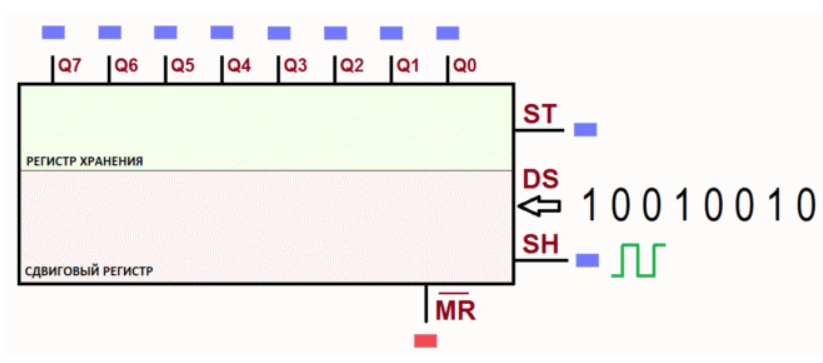


Рис 5.3 Логіка роботи вводу даних у 74HC595

Логіка роботи розглянута на прикладі вводу двійкової послідовності 10010010. Оскільки на сході DS вже є біти для подальшого вивода на індикатори, необхідно ці біти передати у регістр зсуву. Для заповнення 74HC595 ми подаємо імпульси логічної одиниці на вхід SH\_CP, після першого імпульсу перший біт з входу DS, який є одиницею, буде переданий у наймолодший розряд, після наступного імпульсу біт 0 буде переданий у наймолодший розряд та перший переданий біт буде зсунутий вліво на один розряд. Така послідовність буде виконуватися поки не будуть передані усі 8 бітів. Але всі ці значення зберігаються у регістрі зсуву, і не відображені на виходах Q0 ... Q7. Для того щоб дані з'явилися на виходах Q0 ... Q7 необхідно подати високий рівень на вхід ST\_CP. Логіка виводу даних зображена на рисунку 5.4:

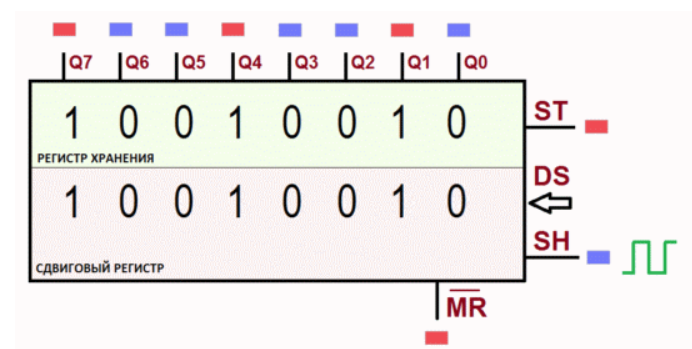


Рис 5.4 Логіка роботи виводу даних з 74HC595.

Тепер можна розібрати роботу коду. Відповідно до розглянутого циклу у розділі 5.3.1 стало зрозуміло що аргумент byte приймає значення number[] від RTC.

void send\_74HC595\_seg (char byte) // Функція формування сегменту

{

char x;

for(x=0; x<8; x++) // Проходження циклу для запису всіх 8 бітів.

{

if(byte&0x80) DS1=1; // Якщо значення старшого біта у аргументі функції один, то значення на DS встановлюється один.

else DS1=0; // В іншому випадку на вході DS формується 0

byte<<=1; // Аргумент побітово зсувається вліво.

SH\_CP1=1; // Подаємо імпульс логічної одиниці на вхід SH\_CP

SH\_CP1=0;

}

ST\_CP1=1; ST\_CP2=1; // Після проходження циклу для запису бітів за допомогою високого рівня на ST\_CP виводимо на виходи Q0 ... Q7 записані значення.

ST\_CP1=0; ST\_CP2=0;

DS1=0; DS2=0; // Обнуляємо вхідні значення.

}

При установці необхідного розряду через функцію send\_74HC595\_dig процес налаштування збігається з встановленими налаштуваннями для функції формування сегменту, за лиш одною різницею, це що в регістрі зсуву для установки розряду ми задіємо всього 4 виходи замість 7.

void send\_74HC595\_dig (char byte) // Функція установки необхідного розряду

{

char x=0;

for(x=0; x<8; x++)

{

if(byte&0x80) DS2=1;

else DS2=0;

byte<<=1;

SH\_CP2=1;

SH\_CP2=0;

}

}

**Висновки до розділу**

У цьому розділі ми оглянули блок-схему алгоритму роботи годинника з режимом детонації. Нами розроблена схема у Proteus та написаний програмний код для роботи годинника. Протестовано створену схему на працездатність та записано на основі цього відеоматеріал. Оглянуто декілька важливих функцій та циклів роботи з лістингу коду, основна частина описаних функцій була спрямована на розуміння роботи регістру зсуву та запису інформації в нього так як це найскладніша частина коду та потребує окремої уваги.

**Висновки**

В рамках дипломного проекту ми розробили годинник реального часу з режимом детонації. Виконане завдання повністю відповідає поставленому завданні у ТЗ.

В першому розділі розглянуто проблематику виміру часу. Проаналізовано принципи роботи годинників з режимом детонації. Розглянули інші існуючі рішення аналогів від компаній та авторів на ринку.

Розглянуті аналоги Nootropic Design на МК Atmega328 та проект Detonation Clock на Atmega8. Обидва мають невеликі габарити та функції які найбільш близько відповідають потребам годинника з режимом детонації. Але обидва цих пристоїв мають в своєму складі недоліки, такі як клеми з проводками. Це може бути не очевидно, але постійне висмикування чи перерізання проводів призводить до заміни проводів чи знову увімкнення їх до клем. Якщо пристрій використовується для ігор чи як реквізит, але якщо це домашній пристрій для власного використання заміна на перемикачі є кращим рішенням. Оскільки цей годинник використовується для домашнього використання було прийнято рішення також замінити колір індикаторів та світлодіодів на зелений з початкового червоного. Це рішення ґрунтується ще на тому, що цей годинник використовується ще як будильник, а червоний колір, який блимаю в обличчя зранку є досить дратівливим.

В другому розділі створено структурну схему пристрою, розділену на 6 блоків. Спираючись на схему структурно нами була розроблена схема електрична принципова. Ми описали роботу кожного з блоків та вибрано найкращі компоненти для нашого проекта. Відповідно до переваг та недоліків які розглянуті у розділі обрано МК, регістр зсуву, RTC та інші компоненти.

В третьому розділі ми обрали матеріал ДП, метод виготовлення, а також клас точності, після цього ми розбробили схему у Altium Designer. Проаналізувавши різні методи виготовлення ДП було обрано комбінований позитивний методом. Матеріалом ДП було обрано склотекстоліт FR4. При виборі типу ДП було обрано двосторонню друковану плату, порівняно з іншими варіантами ДДП є оптимальним рішенням. Ми розглянули п’ять класів ДП, та після цього обрали четвертий клас точності.

В четвертому розділі ми проводили розрахунки які підтвердили коректність прийнятих конструкторських рішень. Розрахували значення паразитної ємності та індуктивності, по результатам розрахунків яких ці величини не впливають на коректну роботу схеми. Розраховали значення надійності ДП. Середній час напрацювання до першої відмови склад 38000 годин, що є приблизно 4.5 років. Ймовірність безвідмовної роботи протягом року склала 0,79 , а ймовірність відмов протягом року склала 0,21. При розрахунку віброміцності ми виявили що друкована плата є абсолютно жорсткою, так як власна частота *f*власн = .

В п’ятому розділі ми створили блок-схему алгоритму роботи годинника з режимом детонації. Розробили робочу симуляцію у Proteus та на основі цього записали відеоматеріал з демонстрацією роботи. Ми оглянули декілька важливих функцій з лістингу коду та розібрати принцип їх роботи.

Усі вищеперераховані дії для створення нашого проекту дають нам змогу переконатися що нами виконані всі умови щоб вважати що цей дипломний проект виконаниний повноцінно.

**Список використаної літератури**

1. 15 кращих смарт-годинників [Електронний ресурс] – <https://www.expertcen.ru/article/ratings/luchshie-umnye-chasy.html>

2. Рекомендації щодо застосування 8-розрядних мікроконтролерів AVR [Електронний ресурс] – <http://www.gaw.ru/html.cgi/txt/app/micros/avr/AVR053.htm>

3. Саморобні електронні годинники, вимір часу [Електронний ресурс] – <https://habr.com/ru/post/383091/>

4. DS3231-BIG-MODUL [Електронний ресурс] – <http://www.kosmodrom.com.ua/el.php?name=DS3231-BIG-MODUL>

5. Аналог №1 Defusable Clock Game Timer Pro [Електронний ресурс] – <https://nootropicdesign.com/game-timer-pro/>

6. Ціни к Аналогу №1 [Електронний ресурс] – <https://nootropicdesign.com/store/product-category/defusable-clock/>

7. Аналог №2 Bomb Watch [Електронний ресурс] – <http://mikrokontrolery.blogspot.com/2011/03/DIY-Devastator-Bombowy-zegarek-ATmega8.html>

8. Аналог №3 Detonation Clock [Електронний ресурс] – <https://sl.webflasher.net/8370844-watch-bomb-or-detonator-clock>

9. Технічні характеристики микроконтролера Atmega8 [Електронний ресурс] – <https://radio-magic.ru/microcontrollers/381-mikrokontroller-atmega8>

10. Технічні характеристики регістру зсуву 74HC595 [Електронний ресурс] – <http://tec.org.ru/board/74hc595b1r_74hc595d/150-1-0-2710>

11. Технічні характеристики RTC DS1307 [Електронний ресурс] – <http://zelectro.cc/Content/Uploads/Common/datasheet/DS1307_rus.pdf>

12. Технічні характеристики АЕЖ CR2032 [Електронний ресурс] – <http://www.dart.ru/cataloguenew/others/accumulator/html/cr2032.shtml>

13. Технічні характеристики кварцового резонатора MC-206 [Електронний ресурс] – <https://datasheetspdf.com/pdf/559387/EpsonCompany/MC-206/1>

14. Технічні характеристики роз’єму KLS1-DC-005A-2.5 [Електронний ресурс] – <https://www.rcscomponents.kiev.ua/product/dc-140-005-kls1-dc-005a-2-5-dc-power-socket_56646.html>

15. Технічні характеристики семисегментного індикатора KCDC04-102 [Електронний ресурс] – <https://www.tme.eu/ru/details/kcdc04-102/displei-led-dvoinye/kingbright-electronic/>

16. Технічні характеристики світлодіодів LTST-C150GKT[Електронний ресурс] -<https://ru.mouser.com/ProductDetail/Lite-On/LTSTC150GKT?qs=kIhwFMorz7OTw%252BezAWlXhg%3D%3D>

17. Технічні характеристики кнопки B3U-1000P[Електронний ресурс] - <https://ru.mouser.com/ProductDetail/Omron-Electronics/B3U-1000P?qs=AO7BQMcsEu4ip80xyf2FwA%3D%3D>

18.Технічні характеристики п'єзовипромінювача KPEG242[Електронний ресурс] - <https://ru.farnell.com/kingstate/kpeg242/piezo-buzzer-pin-type/dp/1502726>

19. Технічні характеристики перемикача Rocker Actuator[Електронний ресурс] -<https://ie.rs-online.com/web/p/dip-sip-switches/7455151/>

20. Друкована плата [Електронний ресурс] – <https://uk.wikipedia.org/wiki/%D0%94%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D0%BD%D0%B0_%D0%BF%D0%BB%D0%B0%D1%82%D0%B0>

ГОСТ 23751-86 ПЛАТЫ ПЕЧАТНЫЕ Основные параметры конструкции.

ГОСТ 23665-79. Платы печатные. Обработка контура. Требования к типовым технологическим процессам.

ГОСТ 23770-86. Платы печатные. Типовые технологические процессы химической и гальванической метализации.

ГОСТ 2.417-9. Платы печатные. Правила выполнения чертежей.

ОСТ 4.010.022-85. Платы печатные. Методы конструирования и расчета.

ГОСТ 2.701-84 – ЕСКД. Правила выполнения схем.

ГОСТ 2.701-84 – ЕСКД. Правила выполнения электрических схем.

ГОСТ 2.701-84 – ЕСКД. Правила оформления схем.

ГОСТ 2.743-91 – УГО. Элементы цифровой техники.

# *Додаток А*

**Технічне завдання**

1. **Найменування та галузь використання**

Багатофункціональний годинник зворотної дії з режимом детонації на основі мікроконтролера. Використовується у страйк болі та квест кімнатах та як звичайний годинник з будильников у житловому приміщенні.

**2. Підстава для розробки**

Підставою для розробки дипломного проекту є завдання, що було дано за наказом по КПІ ім. Ігоря Сікорського № 1196-с від 25.05.2020р.

**3. Мета і призначення розробки**

Метою є створення пристрою, який буде виводити час та працювати в кількох режимах. Основна задача це створити легкий у використанні багатофункціональний пристрій.

**4. Технічні вимоги**

**4.1. Склад виробу й вимоги до пристрою, що розробляється**

Пристрій являє собою моноблочну конструкцію, має забезпечувати підключення до джерела живлення, можливість використання кнопок та виводу реального часу до індикаторів.

**4.2. Вимоги до конструкції**

Габаритні розміри готового виробу не повинні перевищувати 100x80 мм

Вага пристрою повинна бути не більше 0.4 кг.

**4.3. Вимоги до надійності**

Середній час напрацювання на відмову повинен бути не менше 30000 год.

**4.4. Вимоги до віброміцності**

Конструкція повинна бути абсолютно жорсткою.

**4.5. Вимоги безпеки обслуговування**

Керуватися загальними вимогами безпеки до апаратури низької напруги ГОСТ 12.2.007-75.

**4.6. Вимоги до складових частин виробу, сировини, вихідних й експлуатаційних матеріалів**

Мають використовуватись електронні компоненти, що знаходяться у легкому доступі та бажано імпортні.

**4.7. Умови експлуатації**

Кліматичне виконання УХЛ.4.2 по ГОСТ 15150-69.

**4.8. Вимоги до транспортування і зберігання**

Група умов зберігання Л1 згідно ГОСТ 15150-69. Зберігати у зачинених опалювальних вентильованих приміщеннях, при температурі повітря +5…+40oС.Транспортувати автомобільним, залізничним або авіаційним видами транспорту.

**5. Вимоги до документації**

Необхідна документація наводиться в пояснювальній записці до роботи. Додаткова документація не потрібна.

**6. Робота повинна містити в собі документи**

• Пояснювальну записку (формату А4, до 60 аркушів)

• Схему електричну принципову та перелік елементів (формату А1, А4 відповідно)

• Складальне креслення та специфікацію (формату А1, А4 відповідно)

• Креслення друкованої плати (формату А1)

**8. Економічні показники**

В умовах даного проекту не розглядаються.

**Додаток Б**

**Лістинг програми**

**Dangerous clock.c**

#include <mega8.h>

#include <delay.h>

#include <stdlib.h>

#include "buttons.h"

/\* the I2C bus is connected to PORTD \*/

/\* the SDA signal is bit 3 \*/

/\* the SCL signal is bit 2 \*/

.equ \_\_i2c\_port=0x12

.equ \_\_sda\_bit=3

.equ \_\_scl\_bit=2

#include <i2c.h>

//DS1307 Real Time Clock functions

#include <ds1307.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define SOUND\_FREQ 2600 //Резонансная частота пьезогенератора (в Гц)

//ИНДИКАЦИЯ

//Номера выводов, подключенных от регистра сдвига к сегментам индикатора

#define SEG\_A 0 //Q0

#define SEG\_B 1 //Q1

#define SEG\_C 2 //Q2

#define SEG\_D 3 //Q3

#define SEG\_E 4 //Q4

#define SEG\_F 5 //Q5

#define SEG\_G 6 //Q6

#define SEG\_DP 7 //Q7

//Используя десятичные цифры от 0 до 9 в качестве индекса массива, легко выводить в порт нужные коды

unsigned char number[] = {

(1<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(1<<SEG\_E)|(1<<SEG\_F)|(0<<SEG\_G), //0

(0<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(0<<SEG\_D)|(0<<SEG\_E)|(0<<SEG\_F)|(0<<SEG\_G), //1

(1<<SEG\_A)|(1<<SEG\_B)|(0<<SEG\_C)|(1<<SEG\_D)|(1<<SEG\_E)|(0<<SEG\_F)|(1<<SEG\_G), //2

(1<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(0<<SEG\_E)|(0<<SEG\_F)|(1<<SEG\_G), //3

(0<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(0<<SEG\_D)|(0<<SEG\_E)|(1<<SEG\_F)|(1<<SEG\_G), //4

(1<<SEG\_A)|(0<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(0<<SEG\_E)|(1<<SEG\_F)|(1<<SEG\_G), //5

(1<<SEG\_A)|(0<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(1<<SEG\_E)|(1<<SEG\_F)|(1<<SEG\_G), //6

(1<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(0<<SEG\_D)|(0<<SEG\_E)|(0<<SEG\_F)|(0<<SEG\_G), //7

(1<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(1<<SEG\_E)|(1<<SEG\_F)|(1<<SEG\_G), //8

(1<<SEG\_A)|(1<<SEG\_B)|(1<<SEG\_C)|(1<<SEG\_D)|(0<<SEG\_E)|(1<<SEG\_F)|(1<<SEG\_G) //9

};

//Номера выводов регистра, подключенные к индикаторам

#define NUM1 0

#define NUM2 1

#define NUM3 2

#define NUM4 3

#define DS1 PORTB.2 //вход данных

#define ST\_CP1 PORTB.4 //вход «защёлкивающий» данные

#define SH\_CP1 PORTB.5 //вход для тактовых импульсов

#define DS2 PORTC.0 //вход данных

#define ST\_CP2 PORTC.1 //вход «защёлкивающий» данные

#define SH\_CP2 PORTC.2 //вход для тактовых импульсов

volatile char point, count\_point; //Флаг разрешения включения точки и счётчик времени горения точки

volatile char blinking, count\_blink; //Флаг разрешения мигания светодиода таймера и счётчик времени горения светодиода таймера

int random\_led; //Переменная для сохранения псевдослучайного числа

char blink\_led; //Флаг разрешения имитации взрыва светодиодами

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//СВЕТОДИОДЫ

#define LED\_1 PORTB.1

#define LED\_2 PORTC.3

#define DET\_LED PORTC.5 // Вывод индикации включения таймера

#define ALARM\_LED PORTD.4 // Вывод индикации включения будильника

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//ФЛАГИ ДЛЯ УПРАВЛЕНИЯ ИНДИКАЦИЕЙ РАЗРЯДОВ

char digit;

#define DIGIT\_1 0x01

#define DIGIT\_2 0x02

#define DIGIT\_3 0x04

#define DIGIT\_4 0x08

//МАКРОСЫ ДЛЯ УПРАВЛЕНИЯ ИНДИКАЦИЕЙ РАЗРЯДОВ

#define DIGIT\_ON(t) digit|=(t)

#define DIGIT\_OFF(t) digit&=(~t)

#define DIGIT\_IS\_ON(t) ((digit & (t)) != 0)

#define ALL\_TUBE\_ON (digit|=(DIGIT\_1|DIGIT\_2|DIGIT\_3|DIGIT\_4))

#define ALL\_TUBE\_OFF (digit&=~(DIGIT\_1|DIGIT\_2|DIGIT\_3|DIGIT\_4))

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//ВРЕМЯ И БУДИЛЬНИК

volatile unsigned char \*digit\_12, \*digit\_34; //Указатели для вывода на индикатор

unsigned char hour, minute, second;

unsigned char last\_second;

unsigned char hour\_alarm, minute\_alarm; //Буферные переменные настройки будильника

eeprom unsigned char e\_hour\_alarm=0, e\_minute\_alarm=0;

unsigned char minute\_timer, second\_timer; //Буферные переменные настройки таймера

eeprom unsigned char e\_minute\_timer=0, e\_second\_timer=10;

unsigned char alarm\_on; //Переменные режима будильника

eeprom unsigned char alarm\_mode=0;

unsigned char count; //Счётчик для динамической индикации

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define ALARM PIND.0 //Кнопка 3

#define DET PINC.4 //Кнопка 4

unsigned char switch\_key; //Буферная переменная кода нажатой кнопки

unsigned char release; //Флаг нажатия кнопки

char starting\_countdown; //Разрешение запуска обратного отсчёта

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#define WIRE\_1 PINB.0

#define WIRE\_2 PINB.7

#define WIRE\_3 PIND.5

#define WIRE\_4 PIND.6

char defuse, detonate; //Флаги обезвреживания и детонации бомбы

char defuse\_bomb\_flag; //Флаг укомплектованности разъёмов 4-мя проводами

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//ПРОТОТИПЫ

void send\_74HC595\_seg (char byte);

void send\_74HC595\_dig (char byte);

void beep(unsigned int tone);

void check\_wire (void);

void defuse\_bomb (void);

void detonate\_bomb (void);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Timer1 overflow interrupt service routine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

interrupt [TIM1\_OVF] void timer1\_ovf\_isr(void) //Прерывание каждые 3 мс

{

TCNT1H=0xFE;

TCNT1L=0x89;

if (count>=4) count=0;

switch (count)

{

case 0: if(DIGIT\_IS\_ON(DIGIT\_1))

{

send\_74HC595\_dig(1<<NUM4); //Подключаю нужный индикатор

send\_74HC595\_seg(number[\*digit\_12/10]); break;

}

case 1: if(DIGIT\_IS\_ON(DIGIT\_2))

{

send\_74HC595\_dig(1<<NUM3);

if(point) send\_74HC595\_seg((number[\*digit\_12%10])|(1<<SEG\_DP)); //Зажечь точку

else send\_74HC595\_seg(number[\*digit\_12%10]); break;

}

case 2: if(DIGIT\_IS\_ON(DIGIT\_3))

{

send\_74HC595\_dig(1<<NUM2);

send\_74HC595\_seg(number[\*digit\_34/10]); break;

}

case 3: if(DIGIT\_IS\_ON(DIGIT\_4))

{

send\_74HC595\_dig(1<<NUM1);

send\_74HC595\_seg(number[\*digit\_34%10]); break;

}

}

count++;

//Опрос кнопок

BUTTON\_Scan();

//Выключение точки через определённый период времени

if(point)

{

if(++count\_point==150)

{

point=0;

count\_point=0;

}

}

//Мигание светодиода таймера

if(blinking)

{

if(++count\_blink==50) { DET\_LED^=1; count\_blink=0; }

}

//Имитация взрыва светодиодами

if(blink\_led)

{

if(++count\_point==26)

{

count\_point=0;

random\_led=rand();

if(!(random\_led%5)) LED\_1=1; else LED\_1=0;

if(!(random\_led%4)) LED\_2=1; else LED\_2=0;

if(!(random\_led%3)) DET\_LED=1; else DET\_LED=0;

if(!(random\_led%2)) ALARM\_LED=1; else ALARM\_LED=0;

}

}

if((!blink\_led) && (!starting\_countdown)){ ALARM\_LED=0; DET\_LED=0; }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Timer1 output compare A interrupt service routine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

interrupt [TIM1\_COMPA] void timer1\_compa\_isr(void)

{

//Пониженная яркость индикации включения будильника в режиме отображения реального времени

if((alarm\_mode==1)&&(!starting\_countdown))

{

ALARM\_LED=1;

}

if((alarm\_mode==2)&&(!starting\_countdown))

{

ALARM\_LED=1; DET\_LED=1;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Инициализация портов индикаторов и светодиодов

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void IND\_Init(void)

{

PORTB&=~((1<<1)|(1<<2)|(1<<4)|(1<<5));

DDRB|=((1<<1)|(1<<2)|(1<<4)|(1<<5));

PORTC&=~((1<<0)|(1<<1)|(1<<2)|(1<<3)|(1<<5));

DDRC|=((1<<0)|(1<<1)|(1<<2)|(1<<3)|(1<<5));

PORTD&=~(1<<4);

DDRD|=(1<<4);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Формирование цифры(сегментов)

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void send\_74HC595\_seg (char byte)

{

char x;

for(x=0; x<8; x++)

{

if(byte&0x80) DS1=1;

else DS1=0;

byte<<=1;

SH\_CP1=1;

//#asm("nop");

SH\_CP1=0;

}

ST\_CP1=1; ST\_CP2=1;

//#asm("nop");

ST\_CP1=0; ST\_CP2=0;

DS1=0; DS2=0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Установка нужного разряда

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void send\_74HC595\_dig (char byte)

{

char x=0;

for(x=0; x<8; x++)

{

if(byte&0x80) DS2=1;

else DS2=0;

byte<<=1;

SH\_CP2=1;

//#asm("nop");

SH\_CP2=0;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Режим СТС для формирования звукового сигнала

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void beep(unsigned int tone)

{

if(tone==0) {TCCR2=0;}

else {OCR2=(unsigned char)((8000000.0/(tone\*2.0\*128.0))-1.0); TCCR2=(1<<4)|(1<<3)|(1<<2)|(1<<0);}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Обезвреживание "бомбы"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void check\_wire (void)

{

if(WIRE\_1) { if(defuse==0) defuse\_bomb(); if(detonate==0) detonate\_bomb(); }

if(WIRE\_2) { if(defuse==1) defuse\_bomb(); if(detonate==1) detonate\_bomb(); }

if(WIRE\_3) { if(defuse==2) defuse\_bomb(); if(detonate==2) detonate\_bomb(); }

if(WIRE\_4) { if(defuse==3) defuse\_bomb(); if(detonate==3) detonate\_bomb(); }

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Функция обезвреживания "бомбы"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void defuse\_bomb (void)

{

beep(0);

blinking=0;

DET\_LED=0;

delay\_ms(4000);

starting\_countdown=0;

switch(alarm\_mode) //Обновить светодиоды будильника

{

case 0: ALARM\_LED=0; DET\_LED=0; break;

case 1: ALARM\_LED=1; break;

case 2: ALARM\_LED=1; DET\_LED=1; break;

}

defuse\_bomb\_flag=0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Функция детонации "бомбы"

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void detonate\_bomb (void)

{

blinking=0; //Запрет мигания светодиода DET\_LED

DET\_LED=0;

blink\_led=1; //Имитация светодиодами взрыва

beep(SOUND\_FREQ);

ALL\_TUBE\_OFF; //Выключить индикаторы

send\_74HC595\_dig(0x00); //Отключение всех индикаторов

ST\_CP2=1; //Импульс "защёлкивания" регистром

ST\_CP2=0;

DS2=0;

delay\_ms(6000);

beep(0);

blink\_led=0; count\_point=0;

ALARM\_LED=0; DET\_LED=0; LED\_1=0; LED\_2=0;

delay\_ms(3000);

starting\_countdown=0; //Отключаем обратный отсчёт

ALL\_TUBE\_ON; //Включить все индикаторы

switch(alarm\_mode) //Обновить светодиоды будильника

{

case 0: ALARM\_LED=0; DET\_LED=0; break;

case 1: ALARM\_LED=1; break;

case 2: ALARM\_LED=1; DET\_LED=1; break;

}

defuse\_bomb\_flag=0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main(void)

{

// Declare your local variables here

// Input/Output Ports initialization

// Port B initialization

PORTB=0x00; PORTB|=(1<<0)|(1<<7);

DDRB=0x00; DDRB|=(1<<3); //Выход звукового сигнала

// Port C initialization

PORTC=0x00; PORTC|=(1<<4);

DDRC=0x00;

// Port D initialization

PORTD=0x00; PORTD|=(1<<0)|(1<<5)|(1<<6);

DDRD=0x00;

// Timer/Counter 0 initialization

TCCR0=(1<<0)|(1<<2);

TCNT0=0x3D;

// Timer/Counter 1 initialization

TCCR1A=0x00;

TCCR1B=0x03;

TCNT1H=0xFE;

TCNT1L=0x89;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0xFF;

OCR1AL=0xF9;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=1;

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization

TIMSK=0x14;

// Analog Comparator initialization

ACSR=0x80;

SFIOR=0x00;

//Обновление переменных будильника

hour\_alarm=e\_hour\_alarm; minute\_alarm=e\_minute\_alarm;

//Обновить светодиоды будильника

switch(alarm\_mode)

{

case 0: ALARM\_LED=0; DET\_LED=0; break;

case 1: ALARM\_LED=1; break;

case 2: ALARM\_LED=1; DET\_LED=1; break;

}

// I2C Bus initialization

i2c\_init();

// DS1307 Real Time Clock initialization

// Square wave output on pin SQW/OUT: On

// Square wave frequency: 4096Hz

rtc\_init(0,0,0);

//Инициализация кнопок

BUTTON\_Init();

//Инициализация портов ввода-вывода и включение всех нндикаторов

IND\_Init();

ALL\_TUBE\_ON;

// Global enable interrupts

#asm("sei")

while (1)

{

if(!starting\_countdown) { digit\_12=&hour; digit\_34=&minute; } //Обычный режим отображения времени

switch\_key=BUTTON\_GetKey(); //Проверка нажатия кнопок, сохранил код нажатой кнопки в переменной switch\_key

if(switch\_key)

{

//УСТАНОВКА ЧАСОВ

if(switch\_key==KEY\_BUTTON\_1)

{

if(++hour==24) hour=0;

}

//УСТАНОВКА МИНУТ

if(switch\_key==KEY\_BUTTON\_2)

{

if(++minute==60) minute=0;

}

rtc\_set\_time(hour,minute,0);

}

//УСТАНОВКА БУДИЛЬНИКА

if((!ALARM)&&(!starting\_countdown))

{

delay\_ms(20);

hour\_alarm=e\_hour\_alarm; minute\_alarm=e\_minute\_alarm;

digit\_12=&hour\_alarm; digit\_34=&minute\_alarm; //Индикатор показывыает настройки будильника

while(!ALARM)

{

switch\_key=BUTTON\_GetKey();

if(switch\_key==KEY\_BUTTON\_1)

{

if(++hour\_alarm==24) hour\_alarm=0;

}

if(switch\_key==KEY\_BUTTON\_2)

{

if(++minute\_alarm==60) minute\_alarm=0;

}

if((!DET)&&(!release)) { delay\_ms(20); if(!DET) { if(++alarm\_mode==3) alarm\_mode=0; release=1;

switch(alarm\_mode) //Установки будильника

{

case 0: ALARM\_LED=0; DET\_LED=0; break; //Отключен

case 1: ALARM\_LED=1; break; //Будильник сработает один раз

case 2: ALARM\_LED=1; DET\_LED=1; break; //Будильник будет срабатывать каждый день в установленное время

}

}}

if(DET) release=0;

}

e\_hour\_alarm=hour\_alarm;

e\_minute\_alarm=minute\_alarm;

}

//УСТАНОВКА ТАЙМЕРА

if(!DET)

{

delay\_ms(20);

minute\_timer=e\_minute\_timer; second\_timer=e\_second\_timer;

digit\_12=&minute\_timer; digit\_34=&second\_timer; //Индикатор показывыает настройки таймера

//РАЗРЕШЕНИЕ ОБРАТНОГО ОТСЧЁТА

if(!starting\_countdown) starting\_countdown=1; //Включение режима обратного отсчёта

if((!WIRE\_1)&&(!WIRE\_2)&&(!WIRE\_3)&&(!WIRE\_4)) defuse\_bomb\_flag=1; //Проверить укомплектованность 4-х проводов

else defuse\_bomb\_flag=0;

defuse=rand()%4; //Установка флагов обезвреживания и детонации бомбы

do { detonate=rand()%4; } while(defuse==detonate);

while(!DET)

{

switch\_key=BUTTON\_GetKey();

if(switch\_key==KEY\_BUTTON\_1)

{

if(++minute\_timer==99) minute\_timer=0;

}

if(switch\_key==KEY\_BUTTON\_2)

{

if(++second\_timer==60) second\_timer=0;

}

}

e\_minute\_timer=minute\_timer;

e\_second\_timer=second\_timer;

}

//СЧИТЫВАНИЕ ВРЕМЕНИ

delay\_ms(20);

rtc\_get\_time(&hour,&minute,&second);

//Включение точки при изменении секунд

if(!starting\_countdown)

{

if(last\_second!=second)

{

last\_second=second;

point=1;

}

}

//"ОБЕЗВРЕЖИВАНИЕ БОМБЫ"

if(defuse\_bomb\_flag && starting\_countdown) check\_wire();

//ОБРАТНЫЙ ОТСЧЁТ ВРЕМЕНИ

if(starting\_countdown)

{

blinking=1; //Разрешение мигания светодиода DET\_LED

if(last\_second!=second)

{

last\_second=second;

ALARM\_LED=0;

beep(SOUND\_FREQ);

if(second\_timer==0) { second\_timer=60; minute\_timer--; }

second\_timer--;

delay\_ms(60);

beep(0);

if((!minute\_timer)&&(!second\_timer)) //Если достигли 0

{

detonate\_bomb();

}

}

}

//БУДИЛЬНИК

if((hour==hour\_alarm)&&(minute==minute\_alarm)&&(second==0)&&alarm\_mode&&!alarm\_on)

{

alarm\_on=1; //Будильник работает

if((!WIRE\_1)&&(!WIRE\_2)&&(!WIRE\_3)&&(!WIRE\_4)) defuse\_bomb\_flag=1; //Проверить укомплектованность 4-х проводов

else defuse\_bomb\_flag=0;

defuse=rand()%4; //Установка флагов обезвреживания и детонации бомбы

do { detonate=rand()%4; } while(defuse==detonate);

minute\_timer=0; second\_timer=10;

digit\_12=&minute\_timer; digit\_34=&second\_timer;

starting\_countdown=1;

//Начнётся обратный отсчёт с 10 секунд, после этого некоторое время будет звучать сигнал и мигать светодиоды

if(alarm\_mode==1)

{

alarm\_mode=0; //Отключили установку будильника

alarm\_on=0; //Сбросить флаг работы будильника

}

if(alarm\_mode==2)

{

alarm\_on=0; //Сбросить флаг работы будильника

}

}

};

}

**buttons.h**

#ifndef BUTTONS\_h

#define BUTTONS\_h

#pragma used+

//Порт, к которому подключены кнопки

#define PORT\_BUTTON PORTD

#define PIN\_BUTTON PIND

#define DDRX\_BUTTON DDRD

//Номера выводов, к которым подключены кнопки

#define BUTTON\_1 1

#define BUTTON\_2 7

//Коды, которые будут записываться в буфер

#define KEY\_BUTTON\_1 1

#define KEY\_BUTTON\_2 2

//Сколько циклов опроса кнопка должна удерживаться

#define THRESHOLD 20

#define THRESHOLD2 250

//ПРОТОТИПЫ

void BUTTON\_Init(void);

void BUTTON\_Scan(void);

unsigned char BUTTON\_GetKey(void);

//Макроопределения

#define MASK\_BUTTONS (1<<BUTTON\_1)|(1<<BUTTON\_2)

#define BitIsClear(port, bit) ((port & (1<<(bit))) == 0)

#define BitIsSet(port, bit) ((port & (1<<(bit))) != 0)

volatile unsigned char pressedKey = 0; // Переменная для передачи кода нажатой кнопки

unsigned char comp = 0; // Переменная для создания антидребезговой выдержки времени

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Инициализация клавиатуры

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void BUTTON\_Init(void)

{

DDRX\_BUTTON &= ~(MASK\_BUTTONS); // Установка выводов на вход

PORT\_BUTTON |= MASK\_BUTTONS; // Установка подтягивающих резисторов

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Функция опроса кнопок

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void BUTTON\_Scan(void) // Функция работает в прерывании таймера!

{

unsigned char key; // Временная переменная для хранения кода нажатой кнопки

// Последовательный опрос выводов мк

if (BitIsClear(PIN\_BUTTON, BUTTON\_1))

key = KEY\_BUTTON\_1;

else if (BitIsClear(PIN\_BUTTON, BUTTON\_2))

key = KEY\_BUTTON\_2;

else key = 0;

// Если во временной переменной что-то есть

if (key)

{ // и если кнопка удерживается долго

// записать ее код в буфер

if (comp > THRESHOLD2)

{

comp = THRESHOLD2 - 35;

pressedKey = key;

return;

}

else comp++;

if (comp == THRESHOLD)

{

pressedKey = key;

return;

}

}

else comp=0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Функция возврата кода нажатой кнопки

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned char BUTTON\_GetKey(void) // Функция работает в основной программе и возвращает код нажатой кнопки

{

unsigned char key = pressedKey; // Сохранил код нажатой кнопки

pressedKey = 0;

return key;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#pragma used-

#endif