

## АНОТАЦІЯ

У курсовому проекті розроблено шаховий годинник на мікроконтролері. Проведено аналіз сучасного стану розвитку шахових годинників, виявлено їх переваги та недоліки. Розроблено структурну схему шахового годинника. Запропоновано схемотехнічне рішення даного пристрою та проведено моделювання в програмному середовищі Proteus. На основі схеми електричної принципової розроблена друкована плата пристрою та складальне креслення.

## ANNOTATION

The course project developed a chess clock on a microcontroller. The analysis of the current state of development of chess clocks is carried out, their advantages and disadvantages are revealed. The structural scheme of a chess clock is developed. The circuit solution of this device is offered and modeling in the Proteus software environment is carried out. The printed circuit board of the device and the assembly drawing are developed on the basis of the scheme of electric principle.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

ВСТУП .....	5
РОЗДІЛ 1. ШАХОВІ ГОДИННИКИ ТА СУЧАСНІ МІКРОКОНТРОЛЕРИ....	7
1.1 СХЕМА ШАХОВОГО ГОДИННИКА НА ОСНОВІ КВАРЦОВИХ БУДИЛЬНИКІВ.....	7
1.2 ОСНОВНІ ВИДИ МІКРОКОНТРОЛЕРІВ ТА ЇХ АРХІТЕКТУРА.....	9
1.3 СЕРЕДОВИЩЕ ДЛЯ ПРОЕКТУВАННЯ - PROTEUS.....	11
1.4 ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ .....	12
РОЗДІЛ 2. ВИБІР КОМПОНЕНТІВ ТА РОЗРОБКА СТРУКТУРНОЇ І ЕЛЕКТРИЧНОЇ СХЕМИ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА .....	13
2.1 ВИБІР КОМПОНЕНТІВ .....	13
2.2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА.....	15
2.3 РОЗРОБКА ЕЛЕКТРИЧНОЇ СХЕМИ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА.....	16
2.4 РОЗРОБКА СХЕМИ В СЕРЕДОВИЩІ PROTEUS ТА НАПИСАННЯ КОДУ .....	17
2.4.1 Обробка переривань.....	19
2.4.2 Виведення на індикатори .....	20
2.4.3 Обробка натискань кнопок.....	22
2.5 МОДЕЛЮВАННЯ РОБОТИ СХЕМИ .....	24
РОЗДІЛ 3 РОЗРОБКА ПЛАТИ ТА ЕЛЕМЕНТІВ КОНСТРУКЦІЇ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА .....	28
3.1 ВИБІР ТИПУ ТА МАТЕРІАЛУ ДРУКОВАНОЇ ПЛАТИ.....	28
3.2 ВИБІР КЛАСУ ТОЧНОСТІ ТА МЕТОДУ ПРОЕКТУВАННЯ ДРУКОВАНОЇ ПЛАТИ....	30
3.3 РОЗРАХУНОК ШИРИНИ ПРОВІДНИКІВ.....	31
3.4 МОДЕЛЮВАННЯ ПРОЕКТУ В ARES PCB LAYOUT.....	32
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	36

## ВСТУП

Шахові годинники почали використовувати для гри в шахи відносно недавно - тільки в другій половині 19 століття. До цього шахи грали без годинника, і часто шахові ігри вкрай затягувалися[1].

Шаховий годинник був вперше розроблений в 1900 році. Годинник мав два циферблати (для кожного з гравців), стрілки, що фіксують падіння прапорця і прапорець. Годинники заводилися вручну. Форма і марки годинників з плином років змінювалися. За радянських часів дуже популярною маркою були шахові годинники Янтар. В останні роки кращими серед механічних годинників вважаються INSA [1].

Поряд з механічним годинником дуже популярними стали кварцові шахові годинники. Вони відрізняються не функціональністю і зовнішнім виглядом від механічних, але більш міцними і точними. Для їх роботи на рік вистачить одної пальчикової батареї (поставляється в комплекті). Найпопулярнішим кварцовим шаховим годинником є класичний кварц. Ці годинники продаються по всьому світу. У Сполучених Штатах, наприклад, вони є стандартом годинника Федерації шахів США.

**Актуальність** теми полягає в тому, що шахи та шашки не зважаючи на свою довгу історію, все ще залишаються досить популярною настільною грою, а для ефективного контролю часу потрібен шаховий годинник, який зможе собі дозволити кожен гравець в шахи.

**Метою** курсової роботи є створення недорогого шахового годинника з функцією будильника на базі мікроконтролера.

**Об'єктом** дослідження є процес перетворення електричних сигналів мікроконтролером.

**Предметом** дослідження є шаховий годинник на основі мікроконтролера.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### **Методи дослідження:**

- аналіз літературних джерел;
- розробка схеми електричної принципової, її моделювання та розробка плати і складального креслення у відповідних програмних продуктах, що входять до пакету PROTEUS.

### **Структура роботи**

Курсовий проект складається з трьох розділів. В першому розділі проведено аналіз існуючих шахових годинників, відзначені переваги та недоліки тих чи інших варіантів їх реалізації, а також особливості їх конструктивного виконання. В другому розділі проведено розробку структурної схеми пристрою, на її основі розробку схеми електричної принципової, наведено їх опис, призначення та особливості функціонування основних вузлів, а також розроблений програмний код для прошивки мікроконтролера. Третій розділ роботи присвячений розробці топології друкованої плати, вибору матеріалу та способу її виготовлення, а також розробці складального креслення виробу.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1. Шахові годинники та сучасні мікроконтролери

### 1.1 Схема шахового годинника на основі кварцових будильників

При проведенні змагань з шахів необхідний облік часу, який бере на роздуми перед ходом кожен гравець. Для цього існують спеціальні шахові годинники. У них два циферблати і загальний механізм, ще є дві кнопки - одна для одного гравця, інша для другого. Зробивши хід, гравець натискає свою кнопку, який вимикає свій годинниковий механізм і включає годинниковий механізм противника. Таким чином, можна буде за показаннями циферблатів визначити, скільки часу витратив на роздуми кожен гравець [1].

На жаль, шахові годинники в продажу зустрічаються нечасто, але їх можна зробити з двох недорогих і доступних китайських кварцових будильників. Здавалося б, все просто, - треба з'єднати будильники механічно разом і встановити який-небудь двухкнопочний перемикач з залежною фіксацією [1].

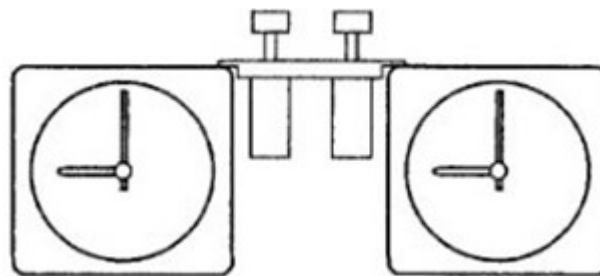


Рисунок 1.1 - Кнопки між годинниками

Але, проблема в тому, що у популярних перемикачів типу П2К або ПКН кнопки розташовані дуже близько, і такий перемикач доводиться розташовувати між корпусами двох будильників (рисунок 1.1). Це дуже незручно, так як, натискаючи один перемикач легко зачепити другий. Необхідно, щоб кнопки розташовувалися на максимальному віддаленні так, як показано на рис.1.2. Але тут знову виникає проблема з механікою [1].

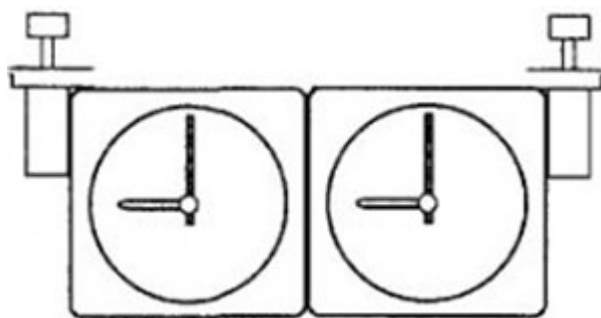


Рис. 1.2 - Кнопки з боків годинника

А що якщо взагалі відмовитися від механіки і згадати стару схему коридорних вимикачів? У цьому випадку те, який з будильників включений, а який вимкнений буде залежати не від конкретного положення перемикача, а від того, перемикачі знаходяться в однакових положеннях або в різних (рисунок 1.3). S2 і S3 - це модульні однокнопочні П2К з незалежною фіксацією. S1 - загальний вимикач [1].

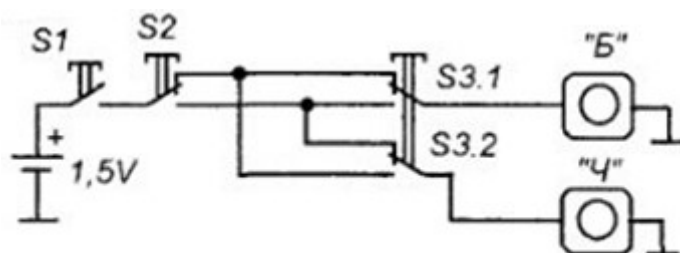


Рисунок 1.3 - Схема перемикання будильників

Перед початком гри S2 і S3 відтиснуті, S1 - вимкнений. Оголошуючи про початок гри, суддя включає S1. Відразу ж починає працювати будильник «Б» (для гравця, що грає білими). Зробивши хід, гравець білими натискає кнопку перемикача S2. Тепер S2 і S3 в різних положеннях.

Живлення відключається від будильника «Б» і переключається на будильник «Ч» (для гравця чорними). Зробивши хід, гравець чорними натискає S3. Тепер обидва перемикача S2 і S3 знову опиняються в однаковому положенні. Будильник «Ч» вимикається, але включається «Б». Зробивши хід, гравець білими знову натискає S2, на цей раз кнопка S2 віджимається і

перемикачі виявляються в різному становищі. Чи включається будильник «Ч». І так далі.

Таким чином, коли кнопки перемикачів S2 і S3 в однакових положеннях (не важливо, натиснуті або відтиснуті, важливо, що в однакових положеннях) включений будильник «Б», а будильник «Ч» вимкнений. Коли кнопки перемикачів S2 і S3 в різних положеннях (все одно в будь, але тільки б в різних), включається «Ч», але вимикається «Б».

Часто сучасні шахові годинники робляться на мікроконтролерах різними виробниками, правда, коштують вони дуже недешево. В даному проекті реалізовано власний шаховий годинник на мікроконтролері [2].

## 1.2 Основні види мікроконтролерів та їх архітектура

Якщо уявити всі типи сучасних мікроконтролерів (МК), то можна здивуватися величезною кількістю різноманітних приладів цього класу, доступних споживачеві. Однак всі попередні можна розділити на наступні основні типи:

- Вбудовувані (embedded) 8-розрядні МК;
- 16- і 32-розрядні МК;
- Цифрові сигнальні процесори.

Промисловістю випускаються дуже широка номенклатура вбудованих МК. У них всі необхідні ресурси (пам'ять, пристрої введення-виведення і т.д.) розташовуються на одному кристалі з процесорним ядром [3]. Якщо подати живлення і тактові імпульси на відповідні входи МК, то можна сказати, що він як би «оживе» і з ним можна буде працювати. Зазвичай МК містять значну кількість допоміжних пристроїв, завдяки чому забезпечується їх включення в реальну систему з використанням мінімальної кількості додаткових компонентів.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

До складу цих МК входять:

- Схема початкового запуску процесора (Reset);
- Генератор тактових імпульсів;
- Центральний процесор;
- Пам'ять програм (Е (Е) PROM) і програмний інтерфейс;
- Засоби введення / виводу даних;
- Таймери, що фіксують число командних циклів.

Загальна структура МК показана на рис. 1.4. Ця структура дає уявлення про те, як МК зв'язується із зовнішнім світом.

Більш складні МК можуть додатково реалізовувати такі можливості:

- Вбудований монітор / відладчик програм;
- Внутрішні засоби програмування пам'яті програм (ROM);
- Обробка переривань від різних джерел;
- Аналоговий введення / виведення;
- Послідовний введення / виведення (синхронний і асинхронний);
- Паралельний введення / виведення (включаючи інтерфейс з комп'ютером);
- Підключення зовнішньої пам'яті (мікропроцесорний режим).

Всі ці можливості значно підвищують гнучкість МК додатків і полегшують розробку систем на їх основі.



Рисунок 1.4 - Структура мікроконтролера

Деякі МК (особливо 16- і 32-розрядні) використовують тільки зовнішню пам'ять, яка включає в себе як пам'ять програм (ROM), так і деякий обсяг



пам'яті даних (RAM), необхідний для даного застосування. Вони застосовуються в системах, де потрібен великий обсяг пам'яті і відносно не велика кількість пристроїв (портів) введення / виведення. Типовим прикладом застосування такого МК із зовнішньою пам'яттю є контролер жорсткого диска (HDD) з буферної кеш-пам'яттю, що забезпечує проміжне зберігання і розподіл великих обсягів даних (порядку декількох мегабайт). Зовнішня пам'ять дає можливість такому мікроконтролеру працювати з більш високою швидкістю, ніж вбудований МК [4].

Цифрові сигнальні процесори (DSP) - відносно нова категорія процесорів. Призначення DSP полягає в тому, щоб отримувати поточні дані від аналогової системи, обробляти дані і формувати відповідний відгук у реальному масштабі часу. Вони зазвичай входять до складу систем, використовуючись в якості пристроїв управління зовнішнім устаткуванням, і не призначені для автономного застосування [5].

### 1.3 Середовище для проектування - Proteus

Proteus є середовищем для проектування та налагодження електронних пристроїв, включаючи мікроконтролери різних сімей. Надає можливості введення схеми в графічному редакторі, моделювання її роботи і розробки друкованої плати, включаючи тривимірну візуалізацію її складання. Унікальною особливістю середовища Proteus є можливість ефективно моделювати роботу різних мікроконтролерів (PIC, 8051, AVR, HC11, ARM7/LPC2000 і т.д.) і налагодження прошивки [10].

Середовище PROTEUS має величезну бібліотеку електронних компонентів, а відсутнє - можна зробити самостійно. Передбачена підтримка SPICE-моделей, які часто надаються виробниками електронних компонентів.

Професійна версія включає в себе USBCONN інструменти для підключення імітації ланцюга до реального USB-порту комп'ютера і COMPIR для підключення до COM-порту ПК [10].

					ВНТУ 153.КП.МПТ.140.01	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Середовище PROTEUS сумісне з популярними середовищами розробки прошивки, включаючи:

- CodeVisionAVR (тільки МК AVR)
- IAR (будь які МК)
- ICC (МК AVR, msp430, ARM7)
- WinAVR (МК AVR)
- Keil (МК 8051 і ARM)
- HiTECH (МК 8051 і PIC)

- Найпотужніша автоматизована система проектування, що дозволяє практично імітувати роботу величезної кількості аналогових і цифрових пристроїв.

- Програмний пакет Proteus VSM дозволяє зібрати схему будь-якого електронного пристрою і симулювати його роботу, виявляючи помилки, допущені на стадії проектування і трасування. Програма складається з двох модулів. ISIS - редактор електронних схем з наступною імітацією їх роботи. ARES - редактор друкованих плат, оснащений автотрасувальником Electra, вбудованим редактором бібліотек і автоматичною системою розміщення компонентів на платі. Крім цього ARES може створити тривимірну модель друкованої плати [10].

#### 1.4 Висновки до першого розділу

Аналіз існуючих шахових годинників показав, що вони мають такі недоліки, а саме: близьке розташування кнопок та висока ціна, тому в другому розділі розроблено ефективну та недорогу схему шахового годинника.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. ВИБІР КОМПОНЕНТІВ ТА РОЗРОБКА СТРУКТУРНОЇ І ЕЛЕКТРИЧНОЇ СХЕМИ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА

### 2.1 Вибір компонентів

На початку процесу розробки було визначено, що в якості дисплея буде використовуватися два 7-сегментних діодних індикатора, представлених на рис.2.1 [17]:



Рисунок 2.1 – семи сегментний чотири-бітний світлодіодний індикатор

Для управління достатньо 5 кнопок - «Гравець 1», «Гравець 2», «Скидання», «Налаштування» і «Пауза» (рисунок 2.2) [5]:



Рисунок 2.2 - Кнопка "Чотири контакти"

Одним з основних елементів схеми являється мікроконтролер. З міркувань економічності, надійності і габаритів обрано мікроконтролер AVR архітектури ATmega8 (рис. 2.3).

Як бачимо він має 23 лінії вводу – виводу (рис.2.4). Для розробленого пристрою їх вистачає, одні порти для виводу інформації на семи сегментний індикаиор, а інші для роботи з рештою зовнішніх елементів.



Рисунок 2.3 - Мікроконтролер Атмега8-16ПУ

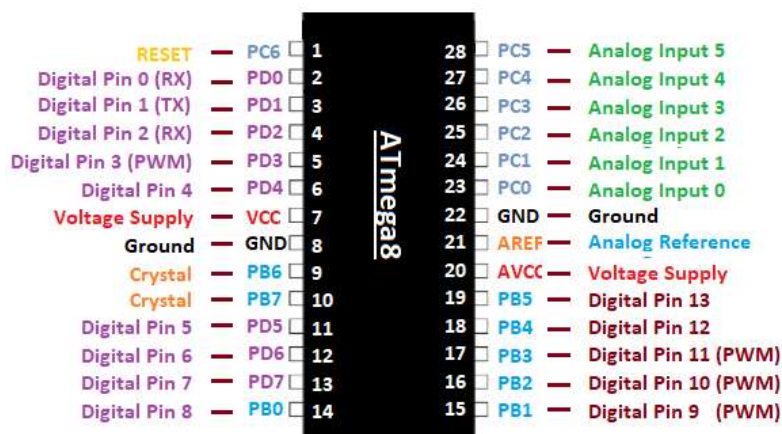


Рисунок 2.4 – Вигляд мікроконтролера АТМега 8 з описом виводів

Основні характеристики даного мікроконтролера:

1. Розрядність – 8 біт;
2. 32 8-бітних робочих регістра;
3. 8 кБ програмованої Flash пам'яті;
4. 512 байта енергонезалежної пам'яті EEPROM;
5. 1 кБ оперативної пам'яті SRAM;
6. Корпус PDIP 28 вивідний;
7. 23 програмованих ліній вводу-виводу;

Споживання в активному режимі при частоті 1 МГц – 230мкА [6].

## 2.2 Розробка структурної схеми цифрового шахового годинника

Відповідно до визначених задач розроблено структурну схему пристрою, яку показано на рис. 2.6.

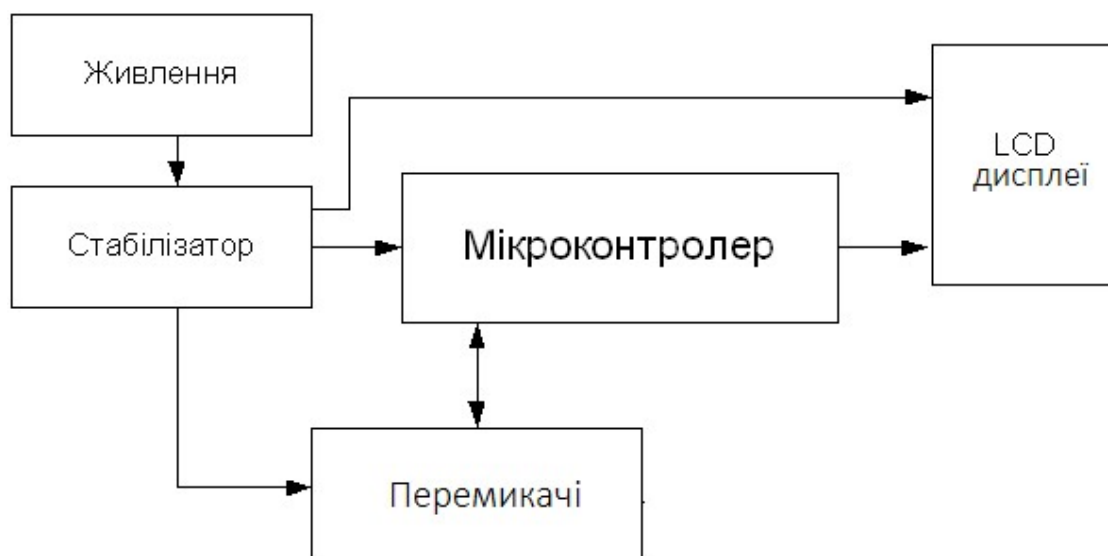


Рисунок 2.6 – Структурна схема шахового годинника

До розробленої структурної схеми входять наступні елементи:

- блок живлення;
- стабілізатор;
- мікроконтролер;
- LCD дисплеї;
- кнопки;

Призначення компонентів наступне. Від блоку живлення до усіх блоків цифрового шахового годинника надходить необхідна напруга, яка стабілізується на рівні 5 Вольт у стабілізаторі. З кнопок надходить сигнал до МК, потім мікроконтролер обробляє данні та відображає інформацію на LCD дисплеї.

## 2.3 Розробка електричної схеми цифрового шахового годинника

На основі структурної схеми розроблено електричну схему пристрою, яка зображена на рис. 2.7.

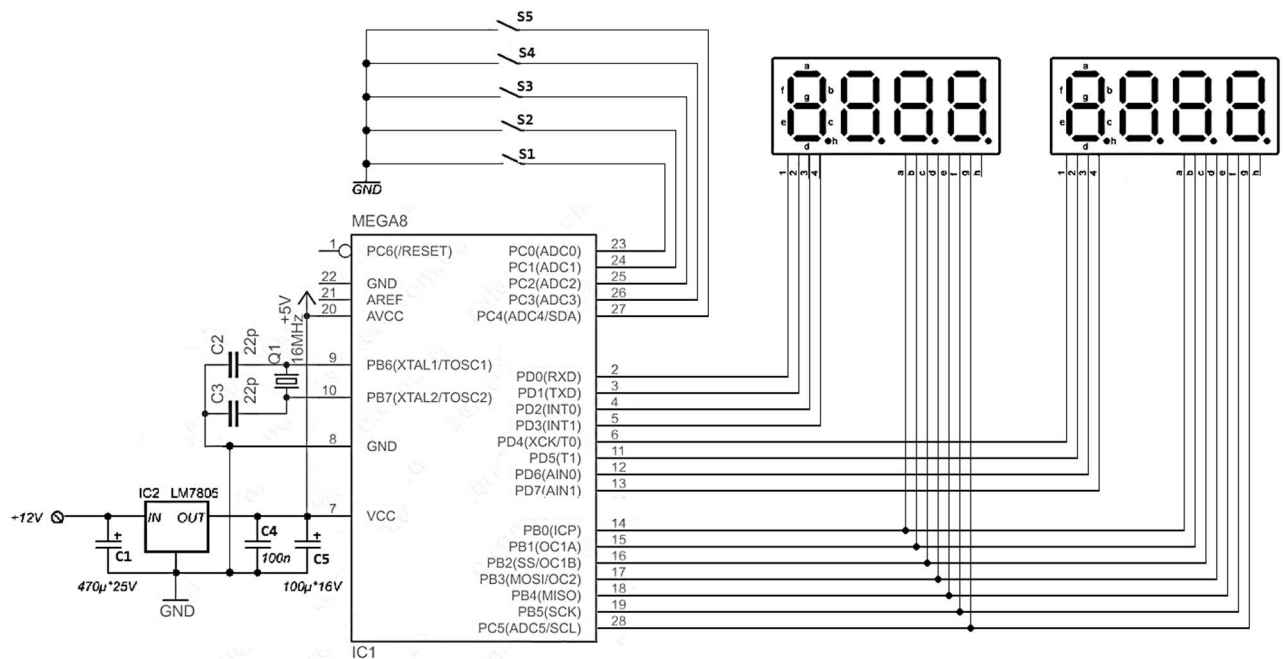


Рисунок 2.7 – Схема електрична цифрового шахового годинника

Принцип роботи пристрою полягає в наступному. Основним елементом цифрового шахового годинника є мікроконтролер. Він обробляє натискання кнопок, та виводить на індикатори відповідні значення. Для роботи мікроконтролера потрібне живлення певного рівня. Це здійснюється за рахунок мікросхеми лінійного стабілізатора напруги LM5806, який встановлює напругу до рівня 5В. Для забезпечення стабільної роботи мікроконтोलера, до його виводів підключено кварцевий резонатор, який задає і підтримує тактову частоту на певному рівні.

## 2.4 Розробка схеми в середовищі Proteus та написання коду

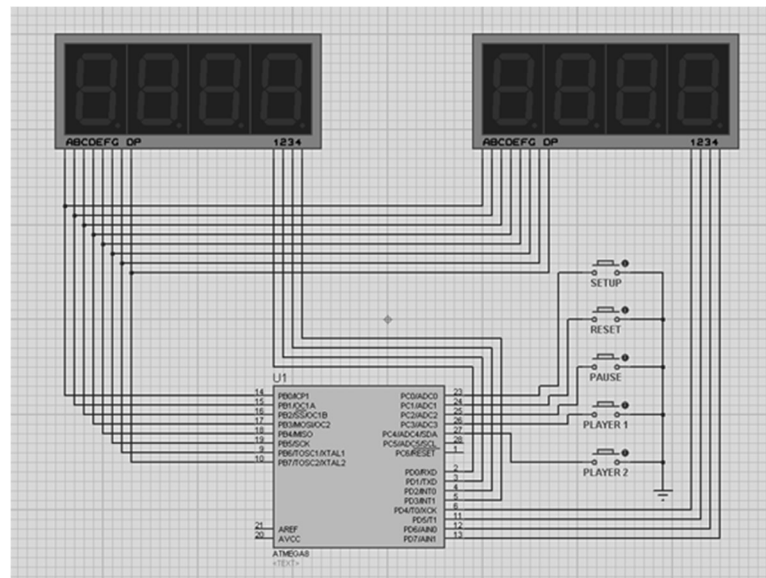


Рисунок 2.8 - Схема шахового годинника в середовищі Proteus

Після створення схеми в середовищі Proteus потрібно запрограмувати мікроконтролер ATmega8.

Програмувати AVR контролери можна як на асемблері (AVR assembler), так і на C. Тут кожен повинен зробити свій вибір сам в залежності від конкретного завдання і своїх переваг. При програмуванні на асемблері архітектура пристрою стає зрозуміліше і з'являється відчуття, що розбираєшся безпосередньо в нутрошах контролера. До того ж в особливо критичних за розміром і продуктивності програмах знання асемблера може дуже стати в нагоді. Вибір був зроблений на користь C.

Починати слід з точки входу програми - функції main. Насправді нічого примітного в ній немає - настройка портів, ініціалізація даних і нескінченний цикл обробки натискань. А також виклик sei () - дозвіл обробки переривань, про них трохи пізніше [10].

```

int main(void)
{
    init_io();
    init_data();
    sound_off();
    sei();
    while(1)
    {
        handle_buttons();
    }
    return 0;
}

```

Розглянемо кожну функцію в окремо [17].

```

void init_io()
{
    // set output
    DDRB = 0xFF;
    DDRD = 0xFF;
    // set input
    DDRC = 0b11100000;
    // pull-up resistors
    PORTC |= 0b00011111;
    // timer interrupts
    TIMSK = (1<<OCIE1A) | (1<<TOIE0);
    TCCR0 |= (1 << CS01) | (1 << CS00);
    TCCR1B = (1<<CS12|1<<WGM12);
    //OCRn = (clock_speed / prescaler) * seconds - 1
    OCR1A = (F_CPU / 256) * 1 - 1;
}

```

					ВНТУ 153.КП.МПТ.140.01	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		



Налаштування портів введення - виведення відбувається дуже просто - в регістр DDRx (де x - літера, що позначає порт) записується число, кожний біт якого означає, чи буде відповідний пін пристроєм введення (відповідає 0) або виведення (відповідає 1). Таким чином, заставши в DDRB і DDRD число 0xFF, ми зробили B і D портами виводу. Відповідно, команда DDRC = 0b11100000; перетворює перші 5 пинів порту C у входні піни, а що залишилися - у вихідні. Команда PORTC |= 0b00011111; включає внутрішні підтягує резистори на 5 входах контролера. Згідно зі схемою, до цих входів підключені кнопки, які при натисканні замкнуті їх на землю. Таким чином, контролер розуміє, що кнопка натиснута [10].

Далі йде налаштування двох таймерів, Timer0 і Timer1. Перший використовується для поновлення індикаторів, а другий - для зворотного відліку часу, попередньо налаштувавши його на спрацьовування кожну секунду. Детальний опис всіх констант і методу налаштування таймера на певний інтервал можна знайти в документації до ATmega8.

#### 2.4.1 Обробка переривань

Для коректної роботи пристрою, необхідно організувати важливу функцію - переривання.

```
ISR (TIMER0_OVF_vect)
{
    display();
    if (_buzzer > 0)
    {
        _buzzer--;
        if (_buzzer == 0)
            sound_off();
    }
}

ISR(TIMER1_COMPA_vect)
```

```

{
if (ActiveTimer == 1 && Timer1 > 0)
{
Timer1--;
if (Timer1 == 0)
process_timeoff();
}
if (ActiveTimer == 2 && Timer2 > 0)
{
Timer2--;
if (Timer2 == 0)
process_timeoff();
}
}

```

При спрацьовуванні таймера - управління передається відповідному обробнику переривання. У нашому випадку, це обробник `TIMER0_OVF_vect`, який викликає процедуру виведення часу на індикатори, і `TIMER1_COMPA_vect`, який обробляє зворотний відлік [10].

#### 2.4.2 Виведення на індикатори

Для того щоб виводити інформацію на індикатори, потрібно правильно налаштувати, виведення інформації.

```

void display()
{
display_number((Timer1/60)/10,0b00001000);
_delay_ms(0.25);
display_number((Timer1/60)%10,0b00000100); _delay_ms(0.25);
display_number((Timer1%60)/10,0b00000010); _delay_ms(0.25);
display_number((Timer1%60)%10,0b00000001); _delay_ms(0.25);
display_number((Timer2/60)/10,0b10000000); _delay_ms(0.25);

```

```

display_number((Timer2/60)%10,0b01000000);_delay_ms(0.25);
display_number((Timer2%60)/10,0b00100000);_delay_ms(0.25);
display_number((Timer2%60)%10,0b00010000);_delay_ms(0.25);
PORTD = 0;
}

void display_number(int number, int mask)
{
PORTB = number_mask(number);
PORTD = mask;
}

```

Функція `display` використовує метод динамічної індикації. Справа в тому, що кожен окремо взятий індикатор має 9 контактів (7 для управління сегментами, 1 для точки і 1 для живлення). Для управління 4 цифрами знадобилося б 36 контактів. Занадто багато. Тому вивід розрядів на індикатор з декількома цифрами організований за наступним принципом (рис. 19) [10]:

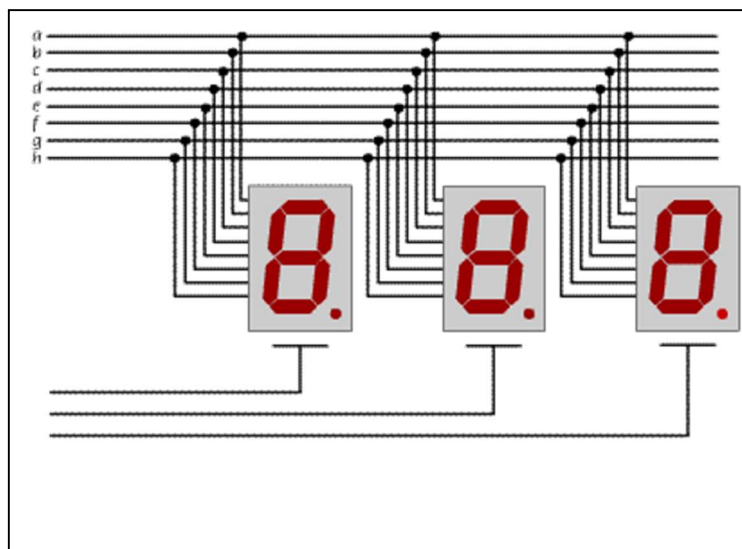


Рисунок 2.9 – Принцип підключення індикаторів

Напруга по черзі подається на кожен із загальних контактів, що дозволяє висвітлити на відповідному індикаторі потрібну цифру за допомогою одних і тих же 8 керуючих контактів. При досить високій частоті виведення це виглядає для ока як статична картинка. Саме тому всі 8 контактів живлення

обох індикаторів на схемі підключені до 8 виходів порту D, а 16 керуючих сегментами контактів з'єднані попарно і підключені до 8 виходів порту В. Таким чином, функція display з затримкою в 0.25 мс поперемінно виводить потрібну цифру на кожен з індикаторів. Під кінець відключаються всі виходи, які подають напругу на індикатори (команда PORTD = 0;). Якщо цього не зробити, то остання виводиться цифра буде продовжувати горіти до наступного виклику функції display, що призведе до її більш яскравого світіння в порівнянні з іншими [10].

### 2.4.3 Обробка натискань кнопок

Для роботи натискання кнопок необхідно написати код для кожної кнопки, щоб мікроконтролер знав яку дію треба виконати при натисканні кожної кнопки.

```
void handle_buttons()
{
    handle_button(KEY_SETUP);
    handle_button(KEY_RESET);
    handle_button(KEY_PAUSE);
    handle_button(KEY_PLAYER1);
    handle_button(KEY_PLAYER2);
}

void handle_button(int key)
{
    int bit;
    switch (key)
    {
        case KEY_SETUP: bit = SETUP_BIT; break;
        case KEY_RESET: bit = RESET_BIT; break;
        case KEY_PAUSE: bit = PAUSE_BIT; break;
        case KEY_PLAYER1: bit = PLAYER1_BIT; break;
```

```

case KEY_PLAYER2: bit = PLAYER2_BIT; break;
default: return;
}
if (bit_is_clear(BUTTON_PIN, bit))
{
if (_pressed == 0)
{
_delay_ms(DEBOUNCE_TIME);
if (bit_is_clear(BUTTON_PIN, bit))
{
pressed |= key;
// key action
switch (key)
{
case KEY_SETUP: process_setup(); break;
case KEY_RESET: process_reset(); break;
case KEY_PAUSE: process_pause(); break;
case KEY_PLAYER1: process_player1(); break;
case KEY_PLAYER2: process_player2(); break;
}
sound_on(15);
}
}
}
else
{
pressed &= ~key;
}
}

```

Ця функція по черзі опитує всі 5 кнопок і обробляє натискання, якщо

					ВНТУ 153.КП.МПТ.140.01	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

таке сталося. Натискання реєструється перевіркою `bit_is_clear (BUTTON_PIN, bit)`, тобто кнопка натиснута в тому випадку, якщо відповідний їй вхід з'єднаний з землею, що і відбудеться, згідно зі схемою, при натисканні кнопки. Затримка тривалістю `DEBOUNCE_TIME` і повторна перевірка потрібна, щоб уникнути множинних зайвих спрацьовувань через брязкіт контактів. Збереження статусу натискання в відповідних бітах змінної `_pressed` використовується для виключення повторного спрацьовування при тривалому натисканні на кнопку.

Функції обробки натискань досить тривіальні і, гадаю, що в додаткових коментарів не потребують.

## 2.5 Моделювання роботи схеми

Запускаючи схему, час на індикаторах відразу встановлюється в 0 і чекає, поки поставлять час. Це продемонстровано на рис.2.10.

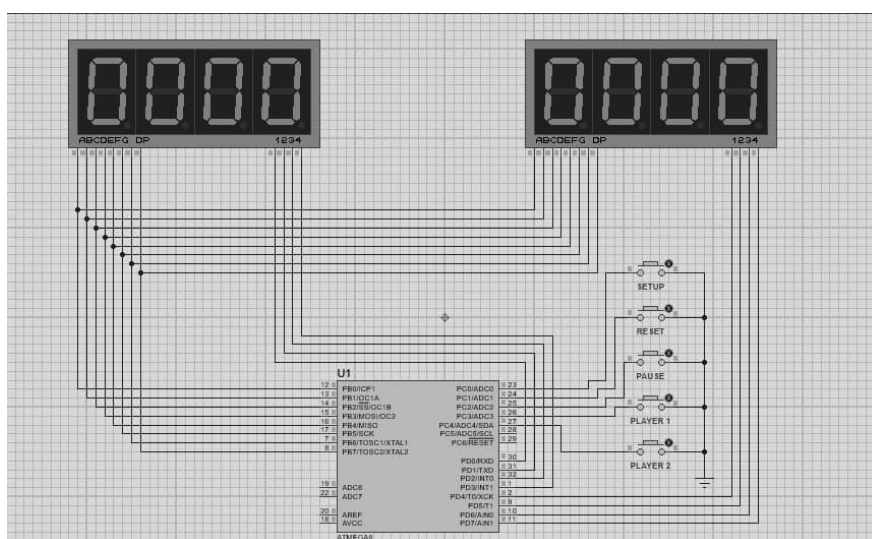


Рисунок 2.10 – Включена схема, індикатори встановлені в 0

Натиснувши три рази на кнопку «SETUP», час на індикаторах було встановлено на 3 хвилини. Це видно на рисунку 2.11:

Натиснувши один раз на кнопку «RESET», час на індикаторах встановився в 0 хвилин. Це можна спостерігати на рисунку 2.12:

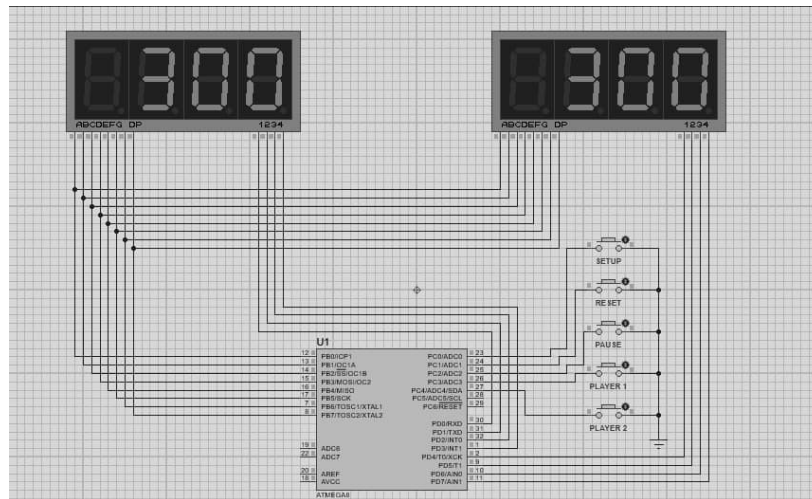


Рисунок 2.11 – Включена схема. Індикатори показують 3 хвилини

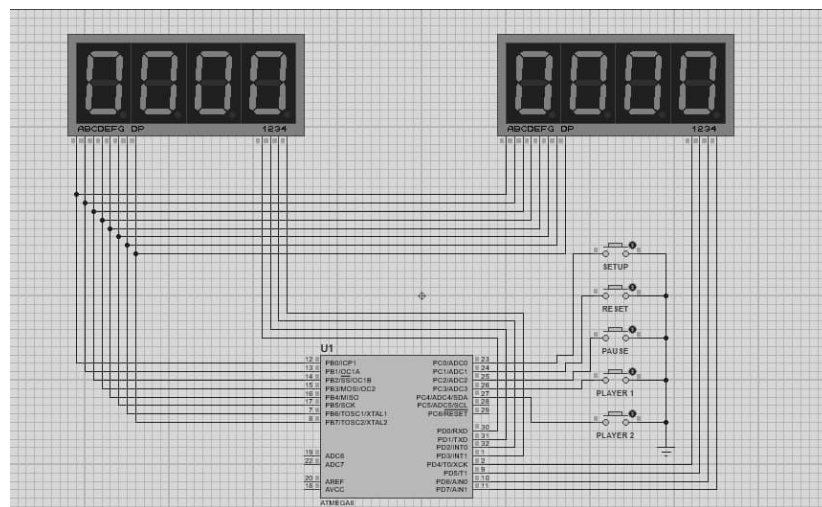


Рисунок 2.12 – Включена схема після натискання кнопки «RESET»

Черговий раз, натиснувши кнопку «SETUP» один раз, було встановлено час в 1 хвилину. Це можна спостерігати на рисунку 2.13:

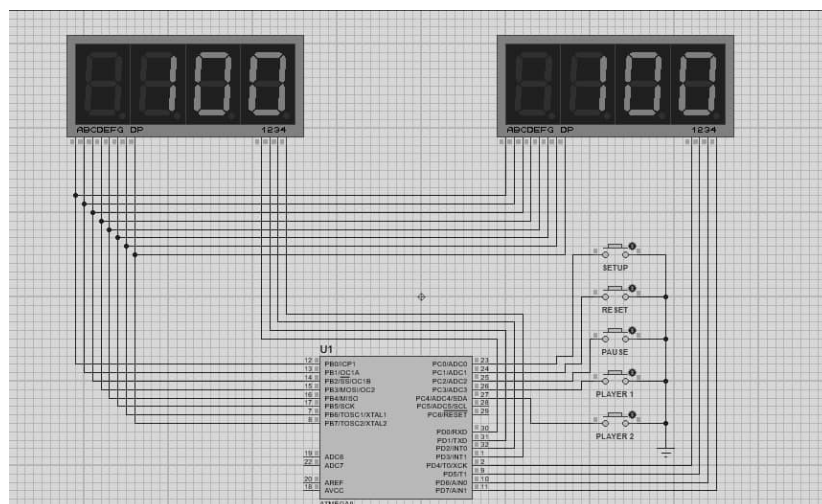


Рисунок 2.13 – Включена схема після одноразового натискання кнопки «SETUP»

Натиснувши кнопку «Гравець 1», час на першому індикаторі пішов в зворотньому порядку, що показано на рисунку 2.14:

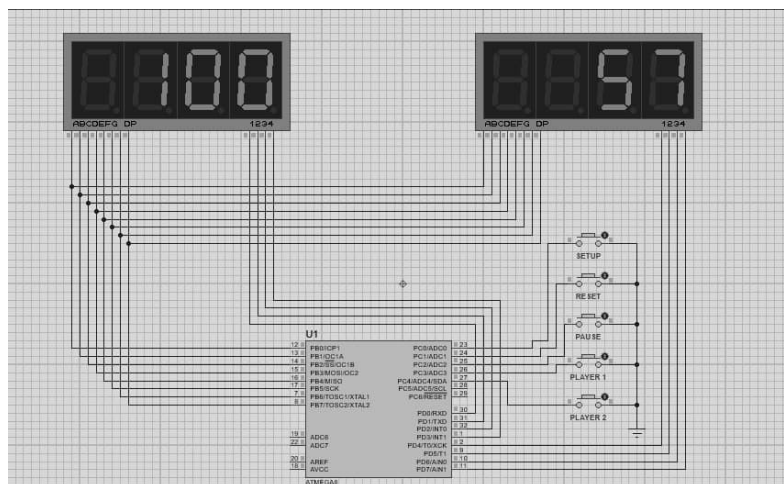


Рисунок 2.14 – Включена схема після натискання кнопки «Гравець 1»

Натиснувши кнопку «Гравець 2», час на першому індикаторі зупинився, а на другому пішов у зворотньому напрямку. Це показано на 2.15:

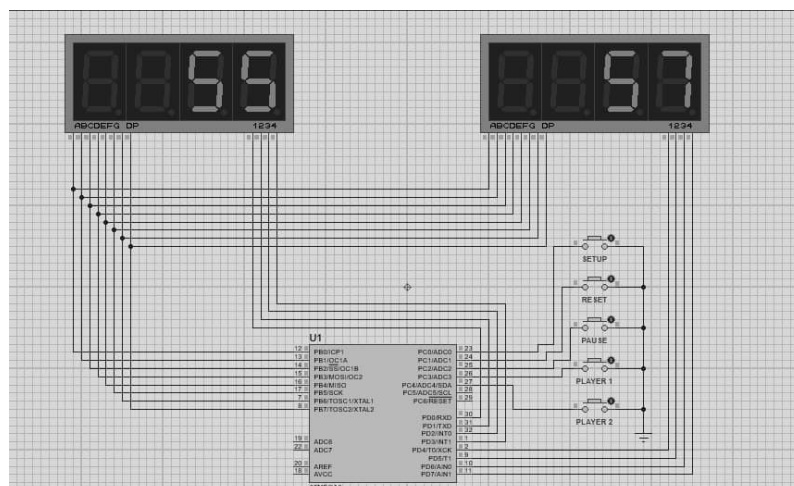


Рисунок 2.15 – Включена схема після натискання кнопки «Гравець 2»

Після багаторазових натискань на кнопки «Гравець 1» та «Гравець 2» натиснемо кнопку «PAUSE» після чого час на обох індикаторах зупиниться. Це показано на рисунку.2.16:



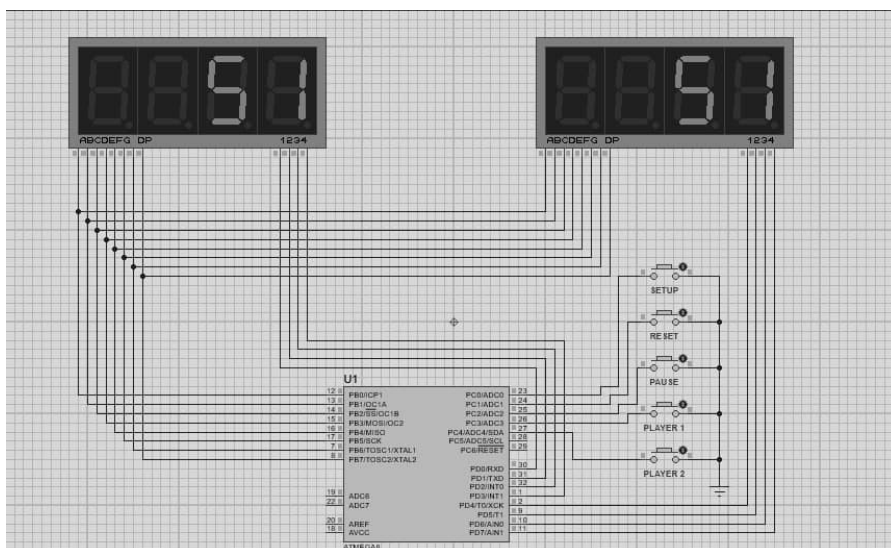


Рисунок 2.16 – Включена схема після натискання кнопки «PAUSE»

Таким чином, отримані результати моделювання показують, що дана схема є працездатною.

## 2.6 Висновки до другого розділу

Розроблено структурну схему цифрового шахового годинника та описано призначення її основних складових компонентів.

Розроблено на базі структурної електричну схему та описано принцип її роботи.

Проведено перевірку та налагодження всіх вузлів схеми з покроковим описання виконаних дій.

Промодельовано розроблену електричну схему в різних режимах роботи, що підтверджує можливість її практичної реалізації.

## РОЗДІЛ 3 РОЗРОБКА ПЛАТИ ТА ЕЛЕМЕНТІВ КОНСТРУКЦІЇ ЦИФРОВОГО ШАХОВОГО ГОДИННИКА

Проведено обґрунтування вибору типу та матеріалу друкованої плати, класу точності та кроку координатної сітки, на основі чого розроблено друковану плату та складальне креслення цифрового шахового годинника.

### 3.1 Вибір типу та матеріалу друкованої плати

При виборі типу друкованої плати для конструкції потрібно враховувати техніко-економічні показники. Модель даного приладу не є досить складною за своєю конструкцією і складається з доступних радіоелементів. Вибираємо двосторонню друковану плату, яка характеризується високими комутаційними якостями за рахунок металізації в отворах, підвищеною міцністю з'єднань виводів навісних елементів з рисунком плати та низькою вартістю.

Для виготовлення друкованої плати використовують фольговані та нефольговані діелектрики. Вихідними матеріалами для їх виготовлення є папір або склотканина, просякнуті синтетичними смолами, або полімерні плівки з лавсану, фторопласту та ін. На поверхню фольгованих матеріалів мідна фольга приклеюється з однієї чи двох сторін листа в процесі виготовлення. Її товщина становить 20 – 50 мкм.

До матеріалу висуваються такі вимоги:

- висока технологічність;
- високі електрофізичні властивості;
- забезпечення високої адгезії;
- мінімальні механічні короблення.

В табл. 3.1 подано основні властивості матеріалів друкованих плат [11].

Найчастіше для виготовлення друкованих плат використовують склотекстоліт і гетинакс наступних марок:

					ВНТУ 153.КП.МПТ.140.01	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

- ГФ – фольгований гетинакс;
- СФ – фольгований склотекстоліт;
- ФГС – склотекстоліт фольгований травильний;
- СФПН – склотекстоліт фольгований нагрівостійкий;
- СТФ – склотекстоліт теплостійкий.

Таблиця 3.1 – Властивості матеріалів друкованих плат

Параметри	Гетинакс	Текстоліт	Склотекстоліт
Щільність без фольги, г/см <sup>3</sup>	1,3...1,4	1,3...1,45	1,6...1,8
Відносна діелектрична проникність	4,5...6	4,5...6	5...6
Питомий об'ємний опір, Ом·м	1012...1014	1012...1014	1014...1015
Діапазон робочих температур, °С	-60...+80	-60...+105	-60...+120
Коефіцієнт теплопровідності, Вт/(м×К)	0,25...0,3	0,23...0,34	0,34...0,74
Температурний коефіцієнт лінійного розширення, 10 <sup>-6</sup> К <sup>-1</sup>	22	22	8...9

Товщина друкованої плати визначається товщиною вихідного матеріалу і вибирається залежно від елементної бази та навантажень.

Найоптимальніші параметри для розробки шахового годинника, згідно табл. 3.1, має склотекстоліт. Тому обираємо двосторонній фольгований склотекстоліт марки СФ-2-35-1,5 ТУ16-503.271-86, який має товщину фольги 35 мкм, товщину матеріалу з фольгою 1,5 мм. та діапазон робочих температур від –60 до +120 °С. Даний матеріал, згідно з [11], володіє досить високим значенням ударної міцності, та здатен витримувати перевантаження в 25g та механічні удари з тривалістю імпульсу в 13 мілісекунд. Будучи вкритим захисним лаковим покриттям, фольгований склотекстоліт стає не чутливим до вологи у всьому робочому температурному діапазоні, що задовольняє умовам поставленого технічного завдання.

### 3.2 Вибір класу точності та методу проектування друкованої плати

До друкованих плат висуваються такі вимоги:

1. Висока точність розташування струмопровідного рисунка
2. Велика величина опору діелектрика
3. Висока механічна міцність
4. Добра здатність до паяння, особливо після тривалого зберігання.

Досягається шляхом вибору типу покриття та технологією його нанесення.

Точність розташування струмопровідного рисунка регламентується вимогами держстандартів. З цією метою введено поняття класу точності друкованих плат.

Таблиця 3.2 – Класи точності друкованих плат

Параметри конструкції плат, мм	Клас точності				
	1	2	3	4	5
1. Ширина друкованих провідників	0,75	0,45	0,25	0,15	0,1
2. Відстань між суміжними провідниками	0,75	0,45	0,25	0,15	0,1
3. Ширина гарантійного пояса	0,3	0,2	0,1	0,05	0,0025
4. Відносна товщина ДП	0,4	0,4	0,33	0,25	0,2

Для даного виробу (згідно таблиці 3.2), з огляду на не складне трасування оптимальним варіантом буде вибір 3-го класу точності друкованої плати (не вимагає використання високоякісних матеріалів, інструменту та обладнання).

Даний клас характеризується наступними параметрами конструкції:

- а) ширина провідника – 0,25 мм;
- б) відстань між сусідніми доріжками – 0,25 мм;
- в) ширина гарантійного пояса – 0,1 мм;
- г) відносна товщина ДП – 0,33 мм.

Установка навісних елементів на друкованій платі здійснюється згідно з ГОСТ 4ГО.010.030-81 та ГОСТ4.ГО.010.009. При розташуванні елементів необхідно враховувати наступні фактори: забезпечення високої надійності, мінімізація розмірів, забезпечення тепловідводу та ремонтпридатності.

Елементи розташовуємо з однієї сторони паралельно сторонам плати по принципу найкоротших зв'язків.

Вибираємо крок координатної сітки 1,25 мм. Координатну сітку на початок координат розташовуємо згідно з ГОСТ 2.417-78. Отвори та елементи друкованого малюнку розташовуємо на платі відносно початку координат [12].

Всі контактні площадки розміщуємо у вузлах сітки. Це ж стосується і виводів елементної бази.

### 3.3 Розрахунок ширини провідників

Ширина провідників визначається у розрахунку на струм, який проходить через них, та максимально можливе падіння напруги.

Для допустимого струму :

$$b_{\min} = \frac{I_{\max}}{j_{\text{доп}} \times t}, \quad (3.1)$$

де  $j_{\text{доп}}$  – допустима густина струму для мідного провідника,  $j_{\text{доп}} = 17 \text{ А/мм}^2$  ;

$t$  – товщина фольги,  $t = 35 \text{ мкм}$ ;

$I_{\max}$  – максимальний постійний струм, який проходить через провідник;

$I_{\max} = 0,03 \text{ А}$ , згідно індивідуального завдання.

$$b_{\min} = \frac{0,03}{17 \times 0,035} = 0,05(\text{мм}) = 50(\text{мкм}) < 250(\text{мкм}).$$

					ВНТУ 153.КП.МПТ.140.01	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Визначимо мінімальну ширину провідника з умови допустимого падіння напруги:

$$b_{\min} = \frac{I_{\max} \times \rho \times l_{\max}}{\Delta U_{\text{доп}} \times t}, \quad (3.2)$$

де  $\rho$  – питомий опір провідників,  $\rho = 0,0175 \text{ Ом} \times \text{мм}^2/\text{м}$ ;

$I_{\max}$  – струм, який проходить по провіднику,  $I_{\max} = 0,03 \text{ А}$ ;

$l_{\max}$  – максимальна довжина провідника,  $l_{\max} = 0,1 \text{ м}$ ;

$U_{\text{доп}}$  – допустиме падіння напруги (становить 5% від напруги живлення  $U_{\text{доп}} = 5 \times 0,05 = 0,25 \text{ В}$ );

$t$  – товщина фольги;

$$b_{\min} = \frac{0,03 \times 0,0175 \times 0,1}{0,25 \times 0,035} = 0,06(\text{мм}) = 60(\text{мкм}) < 250(\text{мкм}).$$

Розраховані значення ширини провідників не перевищують обраних для другого класу точності.

### 3.4 Моделювання проекту в ARES PCB LAYOUT

Розробка друкованої плати в ARES розпочинається з виклику команди Tools>NetlisttoARES в програмі моделювання ISISProteus [13]. Після виклику команди буде автоматично запущено вікно програми ARES з пропозицією вибору необхідного типу друкованої плати, що буде розроблятися. В якості друкованої плати вибираємо плату з двосторонньою металізацією, після чого зберігаємо проект.

В процесі завантаження списку з'єднань для кожного символу схеми в бібліотеках корпусів компонентів \*.LIB знаходиться відповідний корпус. Якщо необхідний корпус відсутній у бібліотеці, то за допомогою вкладки PackageMode створюємо новий корпус або проводимо вибірку одного із існуючих корпусів.

Перед початком процесу автоматичного трасування друкованої плати, за допомогою вкладки Technology>DesignRules проводимо необхідні

					ВНТУ 153.КП.МПТ.140.01	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

налаштування допустимого розміщення провідників, їх мінімальної та максимально можливої товщини, а також зазорів між ними. Для початку трасування виконуємо команду Tools>AutoRouter. Отримаємо розведену плату (рис. 3.1).

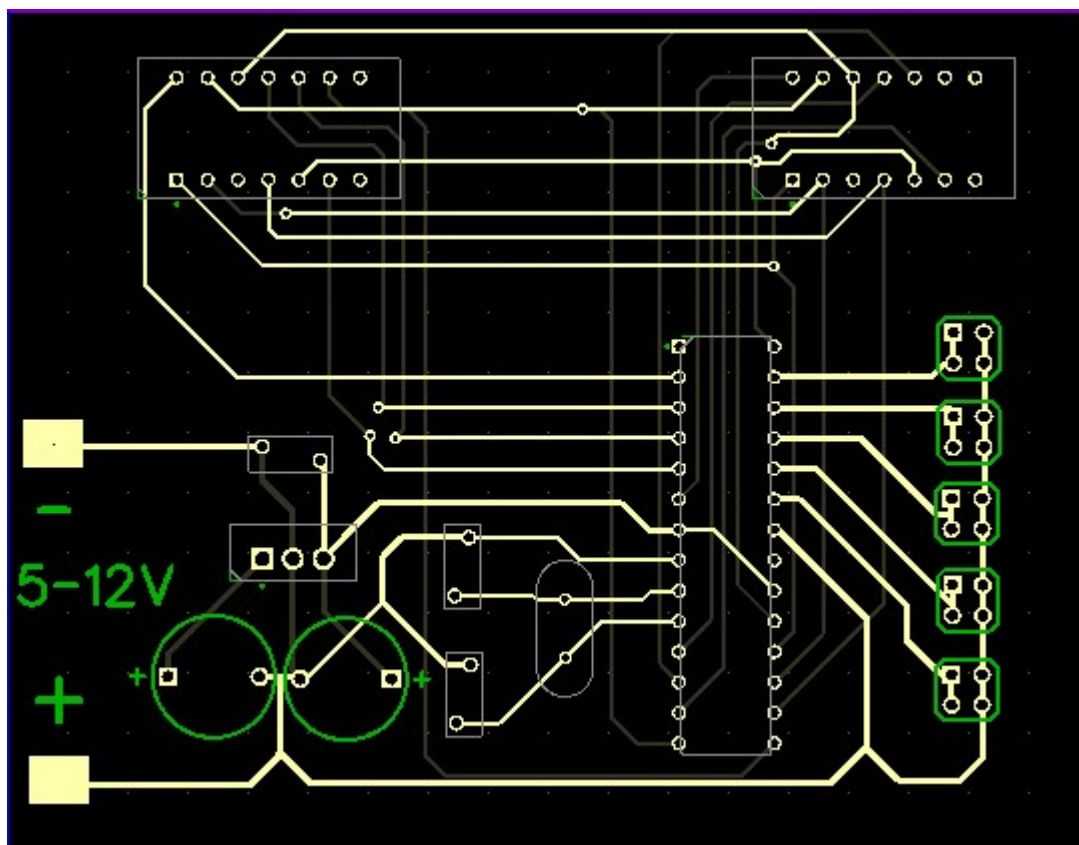


Рисунок 3.1 – Схема друкованої плати для шахового годинника на МК

Обрано Sprint-Layout, так як це простий, але в той же час дуже ефективний програмний пакет для проектування і розведення друкованих плат малої і середньої складності.

### 3.5 Висновки до третього розділу

Виконано розрахунок параметрів друкованої плати та обрано її тип – двостороння. Цей тип характеризується високими комутаційними властивостями, підвищеною міцністю з'єднань виводів навісних елементів з рисунком плати, високою щільністю розташування електронних компонентів та низькою вартістю.

Обрано матеріал друкованої плати – двосторонній фольгований склотекстоліт марки СФ-2-35-1,5 ТУ16-503.271-86 (ГОСТ 10316 – 78), який має товщину 1,5 мм. Для даного типу плати проведено розрахунок ширини друкованих провідників.

Використовуючи ARES PCB Layout створено проект та проведено моделювання пристрою. На основі проектів ISIS Proteus та ARES PCB Layout створено електричну принципову схему, друковану плату та складальне креслення, які наведені в додатках.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		



## ВИСНОВКИ

У даній роботі запропоновано цифровий шаховий годинник на основі мікроконтролера. Для цього виконано ряд робіт.

Виконано аналіз існуючих шахових годинників, який показав, що вони мають такі недоліки, а саме: коштують невиправдано великої кількості грошей, при цьому не можуть похвалитися високою технологічністю або зручністю.

Досліджено, що існуючі шахові годинники широко використовуються для контролю часу в настільних іграх, таких як шахи, шашки, нарди, го, рендзю, сянци, сьогі та інші. .

Розроблено структурну схему цифрового шахового годинника на основі використання мікроконтролера та описано призначення її основних елементів.

Розроблено схему електричну принципову з використанням мікроконтролера для керування процесом відліку часу та обробки натискань на кнопки. Промодельовано її в різних режимах роботи і тим самим підтверджено її працездатність.

Виконано розрахунок параметрів друкованої плати та обрано її тип. У якості матеріалу друкованої плати обрано двосторонній фольгований склотекстоліт марки СФ-2-35-1,5 ТУ16-503.271-86 (ГОСТ 10316 – 78), який має товщину 1,5 мм. Розроблено креслення друкованої плати та складальне креслення. Створено модель плати та навісних елементів в програмному пакеті ARES PROTEUS.

					ВНТУ 153.КП.МПТ.140.01	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Электронные схемы: Схема шахматных часов на основе кварцевых будильников. [Электронный ресурс] – Режим доступа: <https://bit.ly/2XsvDqh>.
2. Кравченко А.В. 10 практичних пристроїв на AVR-мікроконтролерах. Книга 1 -. М, Додека-XX 1, МК-Пресс, 2008 - 224с.
3. Предко, М. Руководство по микроконтроллерам. Том 1. / Пер. с англ. под ред. И. И. Шагурина и С. Б. Лужанского. – М.: Постмаркет, 2001. – 416 с.
4. Предко, М. Руководство по микроконтроллерам. Том 2. / Пер. с англ. под ред. И. И. Шагурина и С.Б. Лужанского. – М.: Постмаркет, 2001. – 488 с.
5. Белов А.В. Мікроконтролери AVR в радіоаматорського практиці - СПб, Наука і техніка, 2007 р. - 352с.
6. Мортон Дж. Мікроконтролери AVR. Вступний курс / Пер. з англ. - М, Додека-XX 1, 2006 -. 272с.
7. Трамперт В. Вимірювання, управління і регулювання за допомогою AVR-мікроконтролерів: Пер. з нім -. До, МК-Пресс, 2006 р. - 208с.
8. Медведев А. В. Печатные платы. Конструкции и материалы: монографія / А. В. Медведев. – М.: Техносфера, 2005. – 304 с.
9. Основы проектирования электронных средств. Часть 1: Учебное пособие /Муромцев Д.Ю., Тюрин И.В./ - 2011г. – 368 с.
10. Proteus. Редактор ISIS [Электронный ресурс] // radio-hobby. – Режим доступа: <http://radio-hobby.org/modules/instruction/instr.php?id=15>.