

УДК 681.515.8

Н.В. Здолбіцька, А.П. Здолбіцький, Б.Ю. Калінін
Луцький національний технічний університет

КОМП'ЮТЕРНА СИСТЕМА ВІЗУАЛЬНИХ ЕФЕКТІВ

Н.В. Здолбіцька, А.П. Здолбіцький, Б.Ю. Калінін. Комп'ютерна система візуальних ефектів. Розроблено програмно-апаратний комплекс для керування 6-а незалежними каналами RGB світлодіодної стрічки.

Ключові слова: RGB, світлодіод, ATmega, PyQt, ШИМ.

Рис. 5. Літ. 10

Н.В. Здолбицкая, А.П. Здолбицкий, Б.Ю. Калинин. Компьютерная система визуальных эффектов. Разработано программно-аппаратный комплекс для управления 6-ю независимыми каналами RGB светодиодной ленты.

Ключевые слова: RGB, светодиод, ATmega, PyQt, ШИМ.

N.V. Zdolbitska, A.P. Zdolbitsky, B.Yu. Kalinin. The computer system of visual effects. Developed software and hardware to control the sixth independent channels RGB LED strip.

Keywords: RGB, LED, ATmega, PyQt, PWM.

Постановка проблеми. Сучасною промисловістю виробляється велика кількість різноманітних систем "розумного" освітлення, проте переважна більшість бюджетних пристроїв виконує лише одну функцію, що призводить до нераціонального використання наявних освітлювальних пристроїв.

Актуальність. Розробка бюджетного багатофункціонального пристрою з відкритим кодом є актуальною, оскільки наразі на ринку ця ніша майже не зайнята.

Формулювання цілі. Розробити бюджетну багатофункціональну систему візуальних ефектів, яка буде реалізовувати мультимедійні функції.

Сучасні системи "розумного" освітлення. Сучасний будинок важко уявити без електричних пристроїв – від вже звичайної "економічної" освітлювальної лампи до складних комп'ютерних систем. Для спрощення керування ними та об'єднання в одну централізовану мережу розроблені системи "розумний" будинок. Це така система, яка об'єднує усі побутові пристрої для можливості простого централізованого, дистанційного чи віддаленого управління. Найпростіші електроприлади, на кшталт ламп освітлення, обігрівачів та електричні розетки, керуються простим вмиканням-вимиканням за допомогою реле [1]. Складніші пристрої, які містять в собі мікроконтролер чи мікропроцесор, отримують керуючі сигнали по одному із протоколів обміну та виконують вказану дію [2]. Найпоширеніші протоколи: RS 485, RS 232, LAN, Wi-Fi та інфрачервоний зв'язок [3].

Основними завданнями системи "розумного" будинку є керування світлом, кліматом, забезпечення безпеки, сенсорне управління, віддалене керування, голосове управління. Найпростішим в реалізації та найнеобхіднішим з точки зору частоти взаємодії користувача з системою є система освітлення.

Більшість існуючих на ринку систем інтелектуального освітлення мають або дуже обмежений функціонал [4], або дуже велику вартість [5]. Розроблений комплекс займає нішу багатофункціональних бюджетних DIY пристроїв, що забезпечується завдяки використанню платформи Arduino [6], поширеної компонентної бази, та вільного ПЗ. Ліцензування розробленого комплексу за допомогою відкритої ліцензії дозволить будь-кому додати потрібний функціонал чи вдосконалити існуючий [7].

Опис розробленої системи. Програмно-апаратний комплекс складається з таких основних блоків (рисунок 1): керуючий пристрій, ключі, джерела світла, джерело живлення, управляюча програма з 2-ма інтерфейсами (консольний та графічний). В якості керуючого пристрою було обрано плату Arduino Nano, як недорогий та поширений контролер з достатньою швидкістю. Для ключів використовуються польові N-канальні транзистори, що дозволяє підібрати необхідні параметри під джерела світла та комутувати навантаження, які живляться більшою напругою, ніж логічний рівень контролера. В якості джерела світла обрано повноколірну світлодіодну стрічку, що дає можливість встановити систему на об'єктах з будь-якою кривизною поверхні та забезпечує енергоефективність всього пристрою. Джерело живлення обирається у відповідності до параметрів джерела світла, тому на платі розташовано 2 різних роз'єми для БЖ. Arduino має вбудований стабілізатор напруги на 5 В, тому може бути заживлено від одного джерела живлення із джерелом світла.

© Здолбіцька Н.В., Здолбіцький А.П., Калінін Б.Ю.

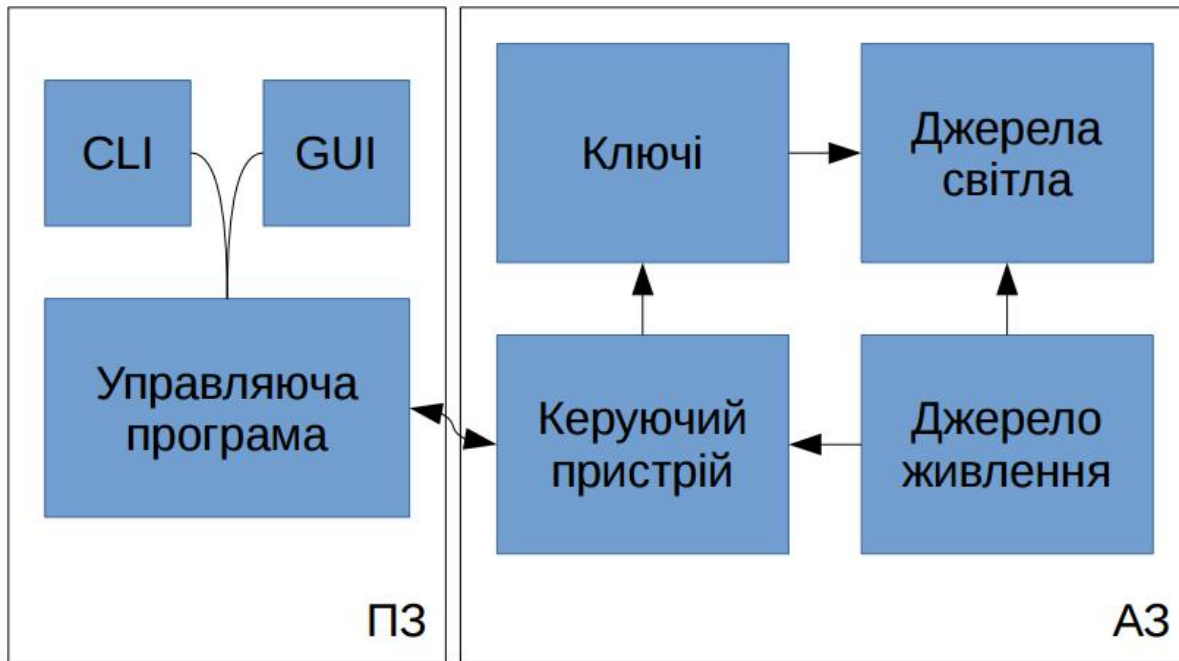


Рисунок 1 – Основні складові комплексу

Принципова схема пристрою показана на рисунку 2. ARDUINO NANO являється центральним керуючим пристроєм, він отримує живлення +12 В із зовнішнього джерела через роз'єми J1 або X7 та конденсаторний фільтр C2, C3. Через вбудовані лінійні стабілізатори Arduino видає напругу +5 В та +3.3 В. +5 В необхідні для підтяжки кнопки U\$2 через резистор R37 та живлення інфрачервоного приймача IR1. Конденсатор C1 служить для згладжування пульсацій при замиканні чи розмиканні механічних контактів, так званий "брязкіт". Ключі на МДН-транзисторах T1...T18 керуються ШІМ сигналом із виводів Arduino A0...A5 та D2...D13. Резистори R1...R36 використовуються в якості обв'язки для транзисторів, парні – для стікання наведеного заряду із затвору на землю в простій, непарні – для захисту портів мікроконтролера від надмірного струму при виході з ладу ключа та для зменшення пікового струму переключення транзистора. Підключення світлодіодної стрічки із спільним анодом відбувається за допомогою роз'ємів J2...J7.

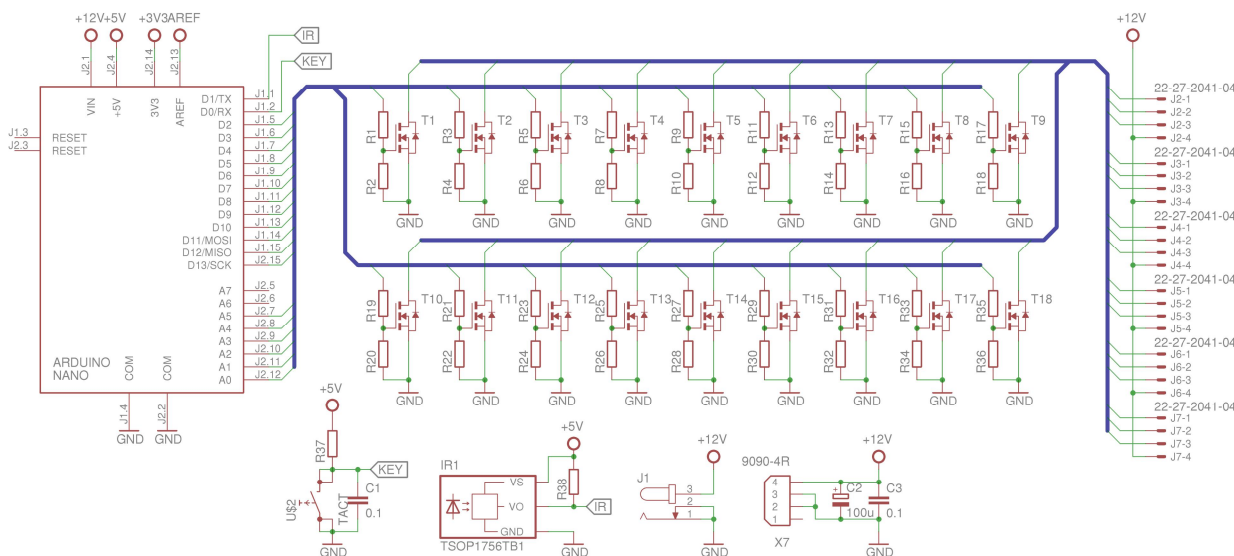


Рисунок 2 – Принципова схема пристрою

Друковану плату для пристрою було розроблено в САПР Eagle CAD (рисунок 3). Кола живлення (роз'єми та фільтри) розташовано в нижній частині плати. Від них відходять широкі

доріжки – шини +12 В та землі. По лівому і правому краю розміщено 6 роз'ємів для світлодіодної стрічки та відповідні ключі біля них. У середній частині – платформа Arduino та захисні резистори.

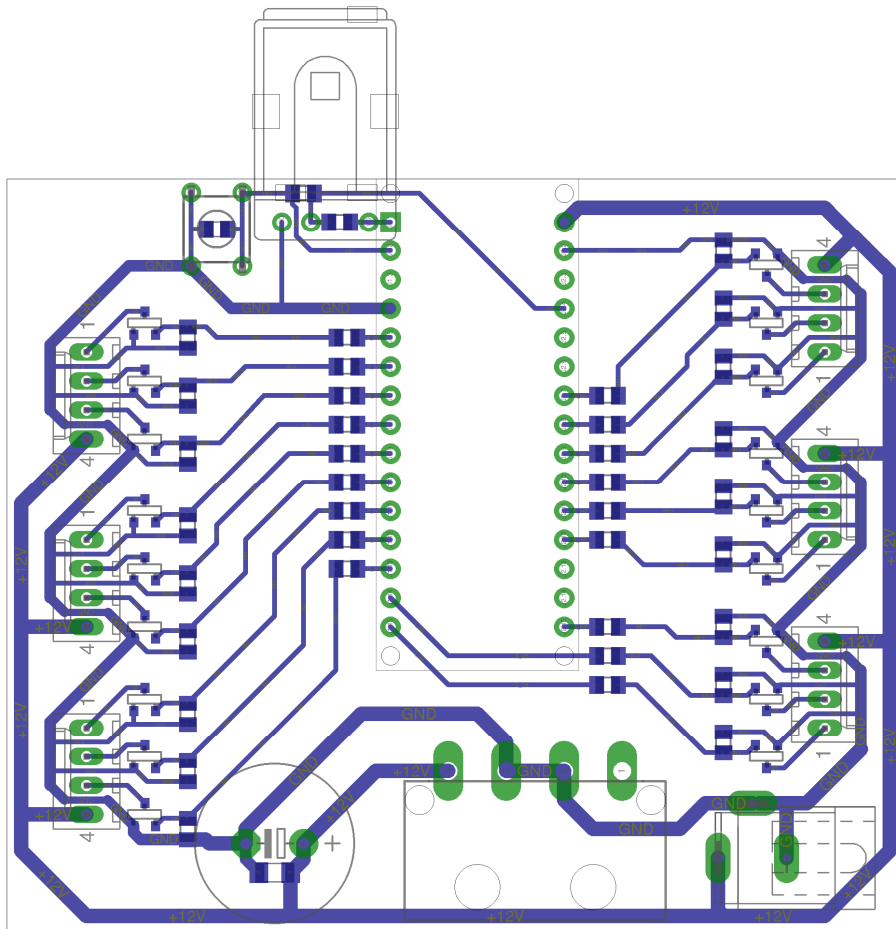


Рисунок 3 – Друкована плата пристрою

Керуюча програма для мікроконтролера написана на мові C з використанням компілятора avr-gcc та стандартної бібліотеки avr-libc [8]. Структуру програми мікроконтролера показано на рисунку 4. При ініціалізації запускаються переривання від лічильника-таймера T/C0, та при отриманні чи надсиланні байта через USART0. В обробниках переривань USART0 реалізований кінцевий автомат для прийому/надсилання повідомлень через послідовний порт, тобто в програмі викликається лише функція прийому/передачі і одразу повертається управління в основну програму, а усі необхідні дії виконуються незалежно і без участі основної програми – це значно спрощує розробку і масштабування коду. На перериванні від лічильника-таймера T/C0 реалізовано програмний ШІМ на 18 незалежних каналах розрядністю 9 біт. При зменшенні кількості каналів можна збільшити розрядність ШІМ, що забезпечить плавнішу зміну кольорів на низькій яскравості.

Основна програма ж виконується у “вічному” циклі, очікуючи приходу команди по послідовному інтерфейсу. Цим забезпечується завершеність автомата світлових ефектів. Усі “інтелектуальні” функції покладені на управляючу програму на комп'ютері.

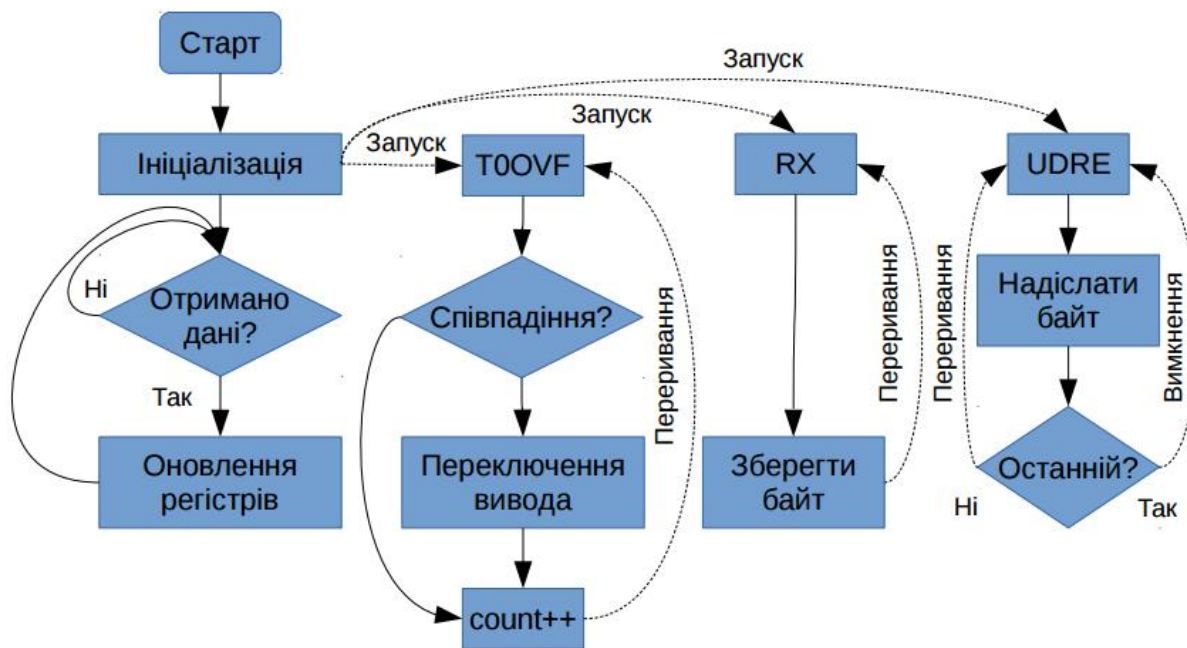


Рисунок 4 – Структура роботи мікроконтролера

Розробка управляючої програми з графічним інтерфейсом для комп'ютера проводилася на мові Python з використанням бібліотек зі стандартного репозиторію [9]. В якості графічної бібліотеки використано PyQt5, оскільки вона забезпечує кросплатформність та зручні інструменти для розробки графічного інтерфейсу. Структура програми показана на рисунку 5.

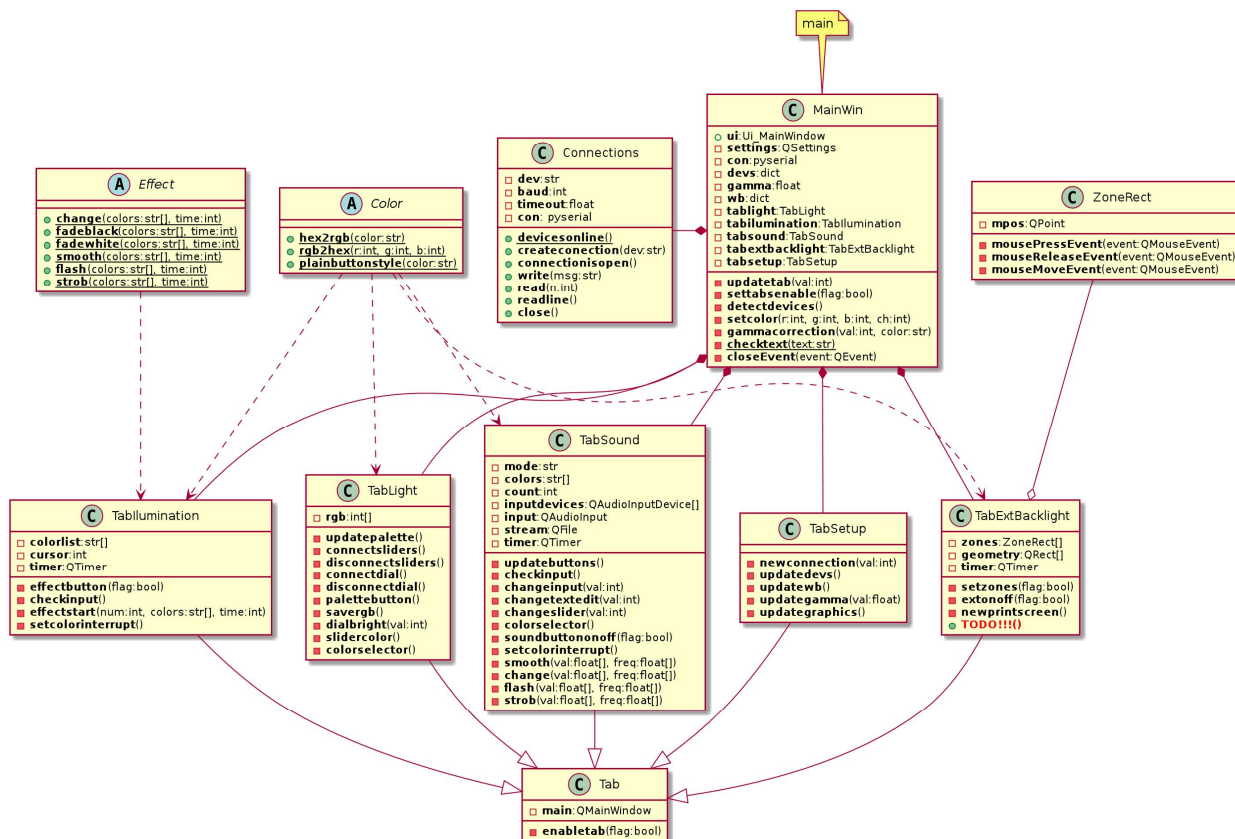


Рисунок 5 – Структура управляючої програми

Клас MainWin є головним класом програми, він керує усіма компонентами форми через об'єкт ui. Також в ньому створюються такі об'єкти як:

- settings – для збереження та відновлення усіх налаштувань та параметрів вікна;
- con – для обміну даними через послідовний порт;
- tablight, tabillumination, tabsound, tabextbacklight, tabsetup – для взаємодії з відповідними вкладками головного вікна.

Клас Tab є батьківським для усіх інших Tab*-класів.

У класі TabLight реалізовано інтерфейс взаємодії для реалізації функцій простого світильника. Він дозволяє обрати колір з палітри або ввести RGB-значення та відрегулювати яскравість.

Клас TabIllumination реалізує динамічну зміну кольорів за наперед заданим користувачем законом. Реалізовані різні варіанти переходів: просте переключення, перехід через чорне, вицвітання до білого, плавна зміна кольору, ефект спалаху та ефект стробоскопу. На вкладці реалізовано 4 секції для зберігання різних наборів кольорів та переходів між ними.

У класі TabSound реалізовано можливість вибору вхідного каналу звукових даних із усіх наявних в системі пристроїв, вибір кольору для низьких, середніх та високих частот, їхні смуги, рівень шуму та регулювання загальної яскравості. Реалізовані 4 режими роботи: звичайний – кількість кольору залежить від амплітуд частот, зміни – програма реагує на низькочастотні біти та змінює колір на наступний на кожному з них та режими спалаху і стробоскопу – реакція на біти із одинарним або подвійним спалахом на них.

Клас TabExtBacklight реалізує функціонал схожий до технології Ambilight фірми Philips[10]. Суть її полягає в аналізі зображення на екрані і підсвіченні зон позаду екрану відповідним кольором, що збільшує занурення в мультимедійний контент. На вкладці реалізовано можливість вибору кількості зон, інтервалу оновлення та вибору розміру і розташування зон на екрані.

У класі TabSetup реалізовано вибір доступних пристроїв, налаштування балансу білого та гама-корекції.

Клас ZoneRect реалізує прозорі прямокутні вікна зі зміною розмірів для вибору зон задньої підсвітки. Клас Connection реалізує інтерфейс взаємодії із послідовним портом. У класі Color реалізовані функції для перетворення хеш-коду кольору в значення R, G, B та навпаки і, також, генерації стилю кнопки в залежності від її кольору. Клас Effects реалізує генерацію таблиць переходів кольорів з одного в інший в залежності від різних ефектів.

Висновки. Здійснено реалізацію програмно-апаратного комплексу на базі платформи Arduino Nano (ATmega162). Проект повністю розроблений “з нуля” з використанням відкритого інструментарію. У системі реалізовано режим освітлення, ефектів, світломузики та фонові підсвітки зображення.

1. Gerhart James Home Automation and Wiring//McGraw-Hill Professional, 31 March 1999. – ISBN 0-07-024674-2.
2. Harper et al, 2003. pp. 18–19
3. Mann William C. The state of the science//Smart technology for aging, disability and independence. – John Wiley and Sons, 7 July 2005. – ISBN 0-471-69694-3.
4. <http://www.theverge.com/2016/8/23/12560024/best-smart-lights-home-gadgets-philips-hue>
5. <https://www.control4.com/solutions/smart-lighting>
6. <https://www.arduino.cc/en/Main/ReleaseNotes>
7. Richard Stallman. Why «Open Source» misses the point of Free Software
8. Артур Гриффитс. GСС. Настольная книга пользователей, программистов и системных администраторов. – Диасофт, 2004. – С. 624. – ISBN 966-7992-34-9.
9. http://docs.linux.org.ua/Програмування/Python/Підручник_мови_Python/
10. <http://www.philips.ru/c-m-so/televisions/p/ambilight>