



**Міжнародний гуманітарний університет
ФАХОВИЙ КОЛЕДЖ
Циклова комісія зі спеціальності «Комп'ютерна інженерія»**

Русу Олександр Петрович

ОФОРМЛЕННЯ ЗВІТУ З ПРАКТИКИ

Методичні рекомендації для самостійної роботи
здобувачів здобувачів фахової передвищої освіти
зі спеціальності 123 Комп'ютерна інженерія

Одеса 2023

Затверджено Педагогічною радою Фахового коледжу Міжнародного гуманітарного університету (протокол №1 від 29 серпня 2023 року).

Русу О.П.

Оформлення звіту з практики: методичні рекомендації для самостійної роботи здобувачів фахової передвищої освіти за спеціальністю 123 «Комп'ютерна інженерія» [Електронне видання] / Русу О.П. Циклова комісія «Комп'ютерна інженерія» Фахового коледжу Міжнародного гуманітарного університету. Одеса, 2023. – 15 с.

Методичні рекомендації «Оформлення звіту з практики» призначені для допомоги здобувачам фахової передвищої освіти за спеціальністю 123 «Комп'ютерна інженерія» в оформленні звітів з практики. Методичні рекомендації розроблені відповідно до робочих програм навчальної, технологічної та переддипломної практики.. Матеріали призначено для здобувачів Фахового коледжу Міжнародного гуманітарного університету.

1. Нормативна база

Дані вимоги сформовані відповідно до ДСТУ 3008:2015 Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання.

2. Структура звіту

Звіт з практики повинен мати:

- титульний аркуш;
- вступ;
- розділ «Відомості про компанію»
- розділ «Відомості про проект»
- розділи, в яких описано зміст вашої роботи;
- висновки;
- перелік джерел посилання;
- додатки (за необхідності).

3. Мова звіту

Звіт і усі його елементи (рисунки, таблиці) повинен бути написаний державною мовою. Винятком, у яких дозволяється використовувати іншомовні слова, є копії екрану та фотографії.

У звіті не бажано вживати іншомовних слів і термінів за наявності рівнозначних слів і термінів в державній мові. Наприклад, замість «скріншот» необхідно використовувати «копія екрану».

4. Загальне оформлення

Звіт друкують з використанням комп'ютера та принтера на одному боці аркуша білого паперу формату А4 (210 мм х 297 мм).

Рекомендовано на сторінках звіту використовувати береги такої ширини: верхній і нижній – 20 мм, лівий – 25 мм, правий – 10 мм. Береги повинні бути однакові на усіх сторінках звіту.

Звіт з практики повинен мати естетичний вигляд. Рекомендовано сторінки звіту скріплювати за допомогою «папки-планки».

Дозволяється звіт писати від руки розбірливим почерком ручкою чорного кольору із дотриманням усіх інших вимог до оформлення звіту. Рисунки, у даному випадку, повинні бути накреслені чорнилом або тушшю чорного кольору. Копії екрану та фотографії дозволяється вклеювати безпосередньо на сторінки.



Папка-планка – рекомендований варіант оформлення звіту

5. Оформлення тексту

Звіт друкують шрифтом Times New Roman чорного кольору прямого накреслення через півтора міжрядкові інтервали кеглем 14.

Основний текст повинен бути вирівняний по ширині сторінки.

Абзацний відступ повинен дорівнювати 1,25...1,27 см і бути однаковим протягом усього звіту.

Заголовки структурних елементів звіту «Зміст», «Вступ», «Висновки», «Перелік джерел посилання» та заголовки розділів та додатків треба друкувати великими літерами напівжирним шрифтом без крапки в кінці. Дозволено їх розміщувати або з абзацного відступу, або посередині рядка.

Заголовки підрозділів, пунктів і підпунктів звіту потрібно друкувати з абзацного відступу з великої літери без крапки в кінці.

Якщо заголовок складається з кількох речень, їх розділяють крапкою. Розривати слова знаком переносу в заголовках заборонено.

У назвах заголовків не рекомендується використовувати скорочення. Винятком можуть бути загальновідомі стандартизовані скорочення (ККД, ШІМ тощо).

Відстань між заголовком, рисунком, таблицею і подальшим або попереднім текстом має бути не менше одного рядка.

Не дозволено розміщувати назву розділу, підрозділу, а також пункту й підпункту на останньому рядку сторінки. Після заголовка, рисунка, таблиці повинно бути не менше двох рядків тексту.

6. Нумерація сторінок

Сторінки звіту нумерують наскрізно арабськими цифрами, охоплюючи додатки. Номер сторінки проставляють праворуч у верхньому куті сторінки без крапки в кінці.

Номер на сторінках «Титульний аркуш», «Зміст», «Вступ», «Висновки», «Перелік джерел посилання», а також на сторінках, що містять назви розділів та додатків, не проставляється.

7. Нумерація структурних елементів

Розділи, підрозділи, пункти, підпункти нумерують арабськими цифрами.

Структурні елементи: «Зміст», «Вступ», «Висновки», «Перелік джерел посилання» не нумерують.

Розділи звіту нумерують у межах викладення суті звіту і позначають арабськими цифрами без крапки, починаючи з цифри «1».

Підрозділи як складові частини розділу нумерують у межах кожного розділу окремо. Номер підрозділу складається з номера відповідного розділу та номера підрозділу, відокремлених крапкою. Після номера підрозділу крапку не ставлять, наприклад, 1.1, 1.2 тощо.

Пункти нумерують арабськими цифрами в межах кожного розділу або підрозділу. Номер пункту складається з номера розділу та порядкового номера пункту, або з номера розділу, порядкового номера підрозділу та порядкового номера пункту, які відокремлюють крапкою. Після номера пункту крапку не ставлять, наприклад, 1.1, 1.2 або 1.1.1, 1.1.2 тощо.

8. Оформлення рисунків

Усі графічні матеріали звіту (ескізи, діаграми, графіки, схеми, фотографії, рисунки, кресленики тощо) повинні мати однаковий підпис «Рисунок».

Рисунок подають одразу після тексту, де вперше посилаються на нього, або якнайближче до нього на наступній сторінці, а за потреби – в додатках до звіту. На кожен рисунок має бути посилання в тексті із зазначенням його номера (наприклад, «Рис. 1.1»).

Якщо рисунки створені не автором звіту, подаючи їх у звіті, треба дотримуватися вимог чинного законодавства України про авторське право. У даному випадку у назві рисунку слід вказати посилання на джерело, з якого цей рисунок взято (див. приклад).

Рисунки нумерують наскрізно арабськими цифрами, крім рисунків у додатках. Дозволено рисунки нумерувати в межах кожного розділу. У цьому

разі номер рисунка складається з номера розділу та порядкового номера рисунка в цьому розділі, які відокремлюють крапкою, наприклад, «Рисунок 3.2» — другий рисунок третього розділу.

Назва рисунка має відображати його зміст, бути конкретною та стислою.

Назву рисунка друкують з великої літери та розміщують безпосередньо під ним посередині рядка без абзацного відступу, наприклад, «Рисунок 2.1 – Схема устаткування».

Приклад оформлення рисунку, зробленого не автором звіту (у квадратних дужках розміщують номер джерела у переліку джерел посилання):



Рисунок 2.1 – Головна будівля ФОП «Інформаційні технології» [1]

9. Оформлення таблиць

Таблицю подають безпосередньо після тексту, у якому її згадано вперше, або на наступній сторінці. На кожену таблицю має бути посилання в тексті звіту із зазначенням її номера.

Таблиці нумерують наскрізно арабськими цифрами, крім таблиць у додатках. Дозволено таблиці нумерувати в межах розділу. У цьому разі номер таблиці складається з номера розділу та порядкового номера таблиці, відокремлених крапкою, наприклад, «Таблиця 2.1» — перша таблиця другого розділу.

Назва таблиці має відображати її зміст, бути конкретною та стислою.

Назву таблиці друкують з великої літери і розміщують над таблицею з абзацного відступу.

Якщо рядки або колонки таблиці виходять за межі формату сторінки, таблицю поділяють на частини, розміщуючи одну частину під іншою або поруч, чи переносять частину таблиці на наступну сторінку. У кожній частині таблиці повторюють її головку та боковик. У разі поділу таблиці на частини дозволено її головку чи боковик замінити відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами в першій частині таблиці. Слово «Таблиця» подають лише один раз над першою частиною таблиці. Над іншими частинами таблиці з абзацного відступу друкують «Продовження таблиці ____» або «Кінець таблиці ____» без повторення її назви.

Приклад оформлення таблиць:

Таблиця 1.1 – Основні характеристики транзисторів OptiMOS

Модель	V_{DS} , В	I_D , А	$R_{DS(on)}$, мОм	Корпус
IPTG007N06NM5ATMA1	60	454	0,75	PG-HSOG-8
IPTG011N08NM5ATMA1	80	408	1,1	PG-HSOG-8
IPTG014N10NM5ATMA1	100	366	1,4	PG-HSOG-8
IPTG111N20NM3FDATMA1	200	108	11,1	PG-HSOG-8
IPTG210N25NM3FDATMA1	250	77	21	PG-HSOG-8
IPTC012N08NM5ATMA1	80	396	1,2	PG-HDSOP-16
IPTC014N08NM5ATMA1	80	330	1,4	PG-HDSOP-16
IPTC015N10NM5ATMA1	100	354	1,5	PG-HDSOP-16
IPTC019N10NM5ATMA1	100	279	1,9	PG-HDSOP-16

10. Оформлення переліків

Переліки (за потреби) подають у розділах, підрозділах, пунктах і/або підпунктах.

Перед переліком ставлять двокрапку. Кожен пункт переліку повинен закінчуватися крапкою з комою окрім останнього – в останньому пункті переліку ставлять крапку.

Якщо подають переліки одного рівня підпорядкованості, то перед кожним пунктом переліку ставлять знак «тире». Для багаторівневих переліків для позначання пункту використовують спочатку малі літери української абетки, далі – арабські цифри, далі – знаки «тире».

Текст кожної позиції переліку треба починати з малої літери з абзацного відступу відносно попереднього рівня підпорядкованості.

Після цифри або літери певної позиції переліку ставлять круглу дужку.

Приклад оформлення однорівневого переліку:

Нові транзистори OptiMOS призначені для використання у наступних типових застосунках:

- імпульсні перетворювачі електричної енергії;
- електричні інструменти;

- електромобілі;
- зарядні пристрої.

Приклад оформлення багаторівневого переліку:

Нові транзистори OptiMOS призначені для використання у наступних типових застосунках:

- а) імпульсні перетворювачі електричної енергії;
- б) електротранспорт:
 - електромобілі;
 - електросамокати;
 - електроскутери;
- в) електричні інструменти:
 - шурупокрути;
 - дрилі;
 - лобзики;
- г) зарядні пристрої.

11. Оформлення формул та рівнянь

Формули та рівняння подають посередині сторінки симетрично тексту окремим рядком безпосередньо після тексту, у якому їх згадано. Відстань між формулою та попереднім або наступним текстом повинно дорівнювати одному рядку.

Формули та рівняння, крім формул і рівнянь у додатках, треба нумерувати наскрізно арабськими цифрами. Дозволено їх нумерувати в межах кожного розділу. наприклад, (2.1) перша формула другого розділу.

Номер формули чи рівняння друкують на їх рівні праворуч у крайньому положенні в круглих дужках, наприклад (3). У багаторядкових формулах або рівняннях їхній номер проставляють на рівні останнього рядка.

Пояснення познач треба подавати без абзацного відступу з нового рядка, починаючи зі слова «де» без двокрапки.

Приклад оформлення формули:

Таким чином, для багатоканальних ключів PROFET, потужність тепла, що виділяється на кристалі мікросхеми, можна обчислити за формулою (3.1):

$$P = V_{S_max} I_{GND_max} + \sum_{i=1}^N I_{L(NOM)i}^2 R_{DS(ON)i} ; \quad (3.1)$$

де N – кількість силових каналів мікросхеми PROFET.

12. Оформлення посилань

У тексті звіту можна робити посилання на структурні елементи самого звіту та інші джерела.

У разі посилання на структурні елементи самого звіту зазначають відповідно номери розділів, підрозділів, пунктів, підпунктів, позицій переліків, рисунків, формул, рівнянь, таблиць, додатків.

Посилаючись, треба використовувати такі вирази: «у розділі 4», «див. 2.1», «відповідно до 2.3.4.1», «(рисунок 1.3)», «відповідно до таблиці 3.2», «згідно з формулою (3.1)», «у рівняннях (1.23) – (1.25)», «(додаток Г)» тощо. Дозволено в посиланні використовувати загальноприйняті та стандартизовані скорочення згідно з ДСТУ 3582, наприклад, «згідно з рис. 10», «див. табл. 3.3» тощо.

Посилання на джерело інформації, наведене в переліку джерел посилання, рекомендовано подавати так: номер у квадратних дужках, за яким це джерело зазначено в переліку джерел посилання, наприклад, «у роботах [2] – [3]».

13. Перелік джерел посилання

Перелік джерел, на які є посилання в основній частині звіту, наводять у кінці тексту звіту перед додатками на наступній сторінці. У переліку джерел посилання бібліографічні описи подають у порядку, за яким джерела вперше згадують у тексті.

Джерела посилання нумеруються арабськими цифрами із крапкою після номера.

Порядкові номери бібліографічних описів у переліку джерел мають відповідати посиланням на них у тексті звіту (номерні посилання).

Якщо джерелом посилання є сайт, то у переліку потрібно привести назву сайту або назву сторінки із вказання адреси (URL).

Приклад оформлення переліку джерел посилання

1. Інформація з сайту ФОП «Інформаційні технології» URL: <https://www.infotech.com.ua/>.
2. Інформація з сайту IAR Systems URL: <https://www.iar.com/>.
3. Інформація з сайту STMicroelectronics URL: <https://www.st.com/>.

14. Оформлення додатків

Додатки позначають послідовно великими літерами української абетки, крім літер Г, Є, З, І, Ї, Й, О, Ч, Ъ, наприклад, ДОДАТОК А, ДОДАТОК Б.

Рисунки, таблиці, формули та рівняння в тексті додатків треба нумерувати в межах кожного додатка, починаючи з літери, що позначає додаток, наприклад, «рисунок Г.3» – третій рисунок додатка Г; «таблиця А.2» – друга таблиця додатка А; «формула (А.1)» – перша формула додатка А.

МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
ФАХОВИЙ КОЛЕДЖ
Циклова комісія «Комп'ютерна інженерія»

ЗВІТ

про проходження технологічної практики
здобувача фахової передвищої освіти
зі спеціальності 123 Комп'ютерна інженерія
на базі практики ФОП «Інформаційні технології»

Виконав: студент 3 курсу, групи KI-3.1

Коваленко Микола Степанович

Керівник Русу Олександр Петрович

Одеса – 2023 рік

ЗМІСТ

	Стор.
ВСТУП.....	13
1 ОПИС БАЗИ ПРАКТИКИ.....	14
1.1 Відомості про компанію	14
1.2 Опис робочого місця.....	15
1.3 Програмне забезпечення	16
1.4 Обов'язки інженера-програміста мікроконтролерів	18
1.5 Основні вимоги з техніки безпеки.....	20
2 ОПИС ПРОЄКТУ «ПЛАТФОРМА ПОБУТОВОЇ АВТОМАТИЗАЦІЇ».....	24
2.1 Загальні відомості	24
2. Мета проекту	25
2.2 Цільова аудиторія.....	25
4. Технічна реалізація	26
2.3 Головні переваги проекту.....	28
3 МОДУЛЬ РЕЛЕ	30
3.1 Електрична схема.....	30
3.2 Протокол обміну даними.....	31
3.3 Ініціалізація модуля UART	32
3.4 Обробка переривань модуля UART при передаванні даних	33
3.5 Обробка переривань модуля UART при прийомі даних.....	34
3.6 Практична реалізація та тестування.....	37
ВИСНОВКИ.....	39
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	40
ДОДАТОК А Вихідний код програмного модуля UART модуля реле.....	41

ВСТУП

У даному документі наведено звіт про проходження технологічної практики студента Коваленка Миколи Степановича, який проходив на базі практики ФОП «Інформаційні технології».

Технологічна практика проводилась відповідно до освітньо-кваліфікаційної програми «Комп'ютерна інженерія» для отримання кваліфікаційного рівня фахового молодшого бакалавра за спеціальністю 123 «Комп'ютерна інженерія» з метою набуття, закріплення й удосконалення отриманих знань, формування професійних умінь і навичок, а також для забезпечення потреби творчого їх застосування.

Технологічна практика проходила в одеському філіалі ФОП «Інформаційні технології», розташованому у місті Одеса за адресою вул. Велика Арнаутська 345 протягом 4 тижнів у період з 8 січня 2023 року по 2 лютого 2023 року. Керівником практики від Фахового коледжу Міжнародного гуманітарного університету був викладач вищої категорії Русу Олександр Петрович, керівником практики від бази практики – інженер відділу розробки програмного забезпечення Петренко Степан Миколайович.

Головним завданням практики було участь у розробці проекту «Платформа побутової автоматизації», в рамках якого я працював на посаді інженера-програміста мікроконтролерів та приймав участь у розробці та тестуванні модуля реле. Основним напрямком практичної роботи було розробка електричної схеми, друкованої плати та програмного забезпечення для мікроконтролерів STM8.

1 ОПИС БАЗИ ПРАКТИКИ

1.1 Відомості про компанію

ФОП «Інформаційні технології» (Рис. 1.1) є провідним виробником електронної техніки та ексклюзивним дистриб'ютором з продажу приладів обліку електроенергії, телекомунікаційного обладнання та інтелектуальних систем, як власного виробництва, так і інших провідних українських та закордонних компаній [1].



Рисунок 1.1 – Головна будівля ФОП «Інформаційні технології» [1]

Компанія ФОП «Інформаційні технології» працює на ринку інтелектуального обладнання сектора вже понад 5 років, має розгалужену мережу дилерів, що дозволяє оперативно реагувати на кон'юнктуру національного та світового ринку приладів обліку електроенергії, що змінюється.

1.2 Опис робочого місця

Одеська філія ФОП «Інформаційні технології» розташована у місті Одеса за адресою вул. Велика Арнаутська 345. У цьому будинку розташовані основні відділи, що працюють у рамках проекту «Smart Home», у тому числі і відділ, що займається розробкою апаратно-програмних модулів, де я проходив практику.

Безпосередньо на робочому місці (Рис. 1.2) мені було виділено окремий стіл, облаштований усім необхідним обладнанням, у тому числі:

- персональний комп'ютер, підключений до локальної інформаційної мережі, корпоративної інформаційної системи та мережі Інтернет;
- програматор ST-Link, призначений для завантаження програмного забезпечення у мікроконтролер;
- цифровий двоканальний осцилограф FNIRSI-1014D;
- два цифрових мультиметра YATO (YT-73080);
- лабораторний блок живлення ZHAOXIN RXN-305D;
- настільна світлодіодна лампа;
- набір інструментів (викрутки, бокорізи, ніж тощо).

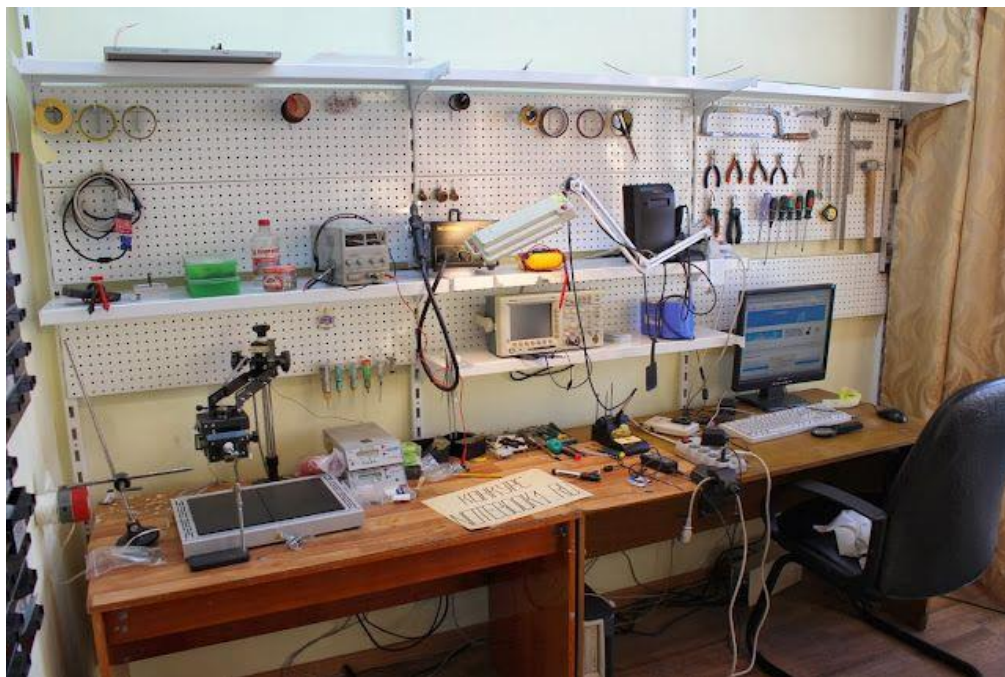


Рисунок 1.2 – Зовнішній вигляд робочого місця

1.3 Програмне забезпечення

Для розробки програмного забезпечення для мікроконтролера використовувалось програмне забезпечення IAR Embedded Workbench – інтегроване середовище розробки, випущене фірмою IAR Systems (Рис. 1.3) [2].

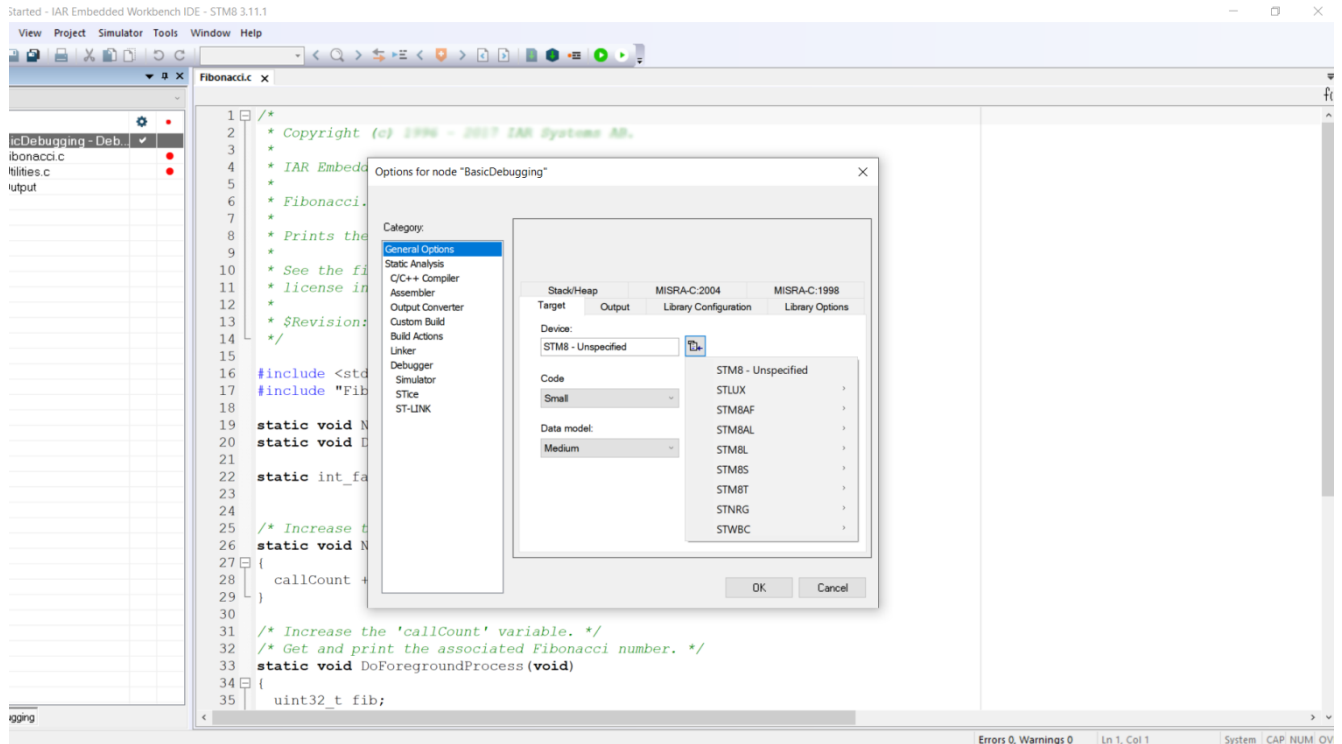


Рисунок 1.3 – Зовнішній вигляд головного вікна IAR Embedded Workbench [2]

Цей програмний продукт містить у собі зручний інтерфейс, оптимізовану CLIB/DLIB бібліотеку, підтримує різноманітні RTOS (Micrium uC/OS-II, OSECO RTI), а також JTAG- адаптери різних фірм (OLIMEX, Phyton, ASHLING). IAR Embedded Workbench підтримує широкий спектр 8-, 16-, 32-розрядних мікроконтролерів – ARM, Actel, Infineon, NEC, Cypress, Atmel, Micronas, Analog Devices, ZiLOG, Microchip, Luminary Micro, Maxim, OKI, NXP, Samsung, STMicroelectronics, Texas Instruments, Renesas, Freescale Semiconductor, SiLabs тощо. Кожній платформі відповідає своє середовище, наприклад платформи ARM відповідає IAR Embedded Workbench for ARM, платформи 8051 – IAR Embedded Workbench for 8051.

В комплект IAR Embedded Workbench входять: C/C++ компілятор, транслятор мови асемблера, компоновальник, підпрограми для роботи з бібліотеками, редактор, менеджер проектів, C-SPY відладчик.

STM8CubeMX – це графічний інструмент, який дозволяє дуже легко конфігурувати мікроконтролери STM8 і генерувати вихідний код, що відповідає створеній конфігурації.

Рисунок 1.4 – Зовнішній вигляд головного вікна STM8CubeMX [3]

Після цього користувач може налаштувати мікроконтролер за допомогою засобу розв’язання конфліктів контактів, помічника налаштування дерева годинника та калькулятора споживання електроенергії.

Результатом роботи із програмною є вихідний код, який інтегрується, наприклад, шляхом копіювання та вставки через буфер обміну, у вихідний код програмного модуля.

1.4 Обов’язки інженера-програміста мікроконтролерів

Відповідно до посадової інструкції, з якою я був ознайомлений перед початком роботи, інженер-програміст мікроконтролерів повинен знати:

- керівні і нормативні матеріали, що стосуються методів програмування і використання обчислювальної техніки в процесі оброблення інформації;
- техніко-експлуатаційні характеристики, конструктивні особливості, призначення і режими роботи устаткування, правила його технічної експлуатації;
- технологію механізованого оброблення інформації;
- види технічних носіїв інформації;
- методи класифікації і кодування інформації;
- формалізовані мови програмування;
- чинні стандарти, системи числення, шифрів і кодів;
- методи програмування;
- порядок оформлення технічної документації;
- передовий вітчизняний і світовий досвід програмування і використання обчислювальної техніки;
- основи економіки, організації праці і виробництва;
- правила та норми охорони праці.

Безпосередньо на робочому місці інженер-програміст мікроконтролерів повинен виконувати накази керівника проекту, а також керівників підприємства.

До головних обов'язків інженера-програміста мікроконтролерів відносяться:

- на основі аналізу математичних моделей і алгоритмів (постановок економічних та інших задач) розробка програм, які реалізують розв'язання задач;
- розробка технологій розв'язання задачі на усіх етапах;
- вибір мови програмування і перекладання на неї алгоритмів задач;
- пошук інформації, необхідної для реалізації задачі: її обсяг, структуру, макети і схеми вводу, оброблення, зберігання і видавання, методи її контролю;
- визначення обсягів і змісту даних тестових прикладів, які забезпечують найбільш повну перевірку відповідності програм їх функціональному призначенню;
- виконання робіт під час підготовки програм до налагодження і виконання їх налагодження;
- розробку інструкції на роботи з програмами, створення необхідної технічної документації;
- визначання можливості використання готових програмних засобів;
- супроводження впроваджених програм і програмних засобів;
- розробка і впровадження методів і засобів автоматизації програмування, типових і стандартних програмних засобів;
- приймання участі у проектних роботах;
- на основі логічного аналізу проведення камеральної перевірки програм;
- визначення сукупності даних, що забезпечують урахування максимального числа умов, які включені до програми, виконання робіт з її підготовки до налагодження;
- проведення налагодження розроблених програм, коригування їх у процесі доопрацювання;
- розробка інструкцій щодо роботи з програмами, оформлення необхідної технічної документації.

1.5 Основні вимоги з техніки безпеки

Відповідно до інструкції з охорони праці, з якою я був ознайомлений перед початком роботи, інженер-програміст мікроконтролерів повинен:

- знати правила пожежної безпеки і вміти користуватися первинними засобами пожежогасіння (вогнегасниками);
- мати навички надання першої (долікарської) допомоги;
- при виявленні несправності обладнання, устаткування, пристроїв, інші небезпечні прояви та нещасні випадки, які трапилися в приміщенні, повідомляти начальника відділу або особу, яка його заміняє.

Вимоги безпеки перед початком роботи інженер програміст повинен:

- оглянути своє робоче місце з метою виявлення небезпечних для життя та здоров'я факторів;
- у разі виявлення порушень або несправностей вжити заходів щодо їх усунення, а за необхідності — повідомити начальника відділу або інженера з охорони праці.

Вимоги безпеки під час виконання роботи:

- виконувати роботу згідно зі своїми функціональними обов'язками;
- не залишати без нагляду своє робоче місце, коли обладнання підключено до електромережі (комп'ютер, електроприлади тощо).

Вимоги до безпеки після закінчення роботи:

- перевірити своє робоче місце;
- відключити від електромережі електрообладнання;
- закрити вікна, погасити світло;
- про виявлені недоліки повідомити начальника відділу або особу, що його заміняє.

Вимоги до безпеки в аварійних ситуаціях:

- при виявленні небезпечної ситуації (пожежа, землетрус, радіаційна небезпека, неполадки в електрогосподарстві тощо) для його життя та життя

співробітників заспокоїтись і заспокоїти оточуючих, оцінити важкість аварійної ситуації;

- не усувати самому несправностей електромережі та електрообладнання, а вимкнути загальне електропостачання;

- при виявленні пожежі негайно викликати пожежну частину за телефоном 101, повідомити начальника відділу, директора або осіб, що їх заміняє;

- вжити заходів згідно з планом евакуації на випадок пожежі та вивести людей у безпечне місце, організувати роботу щодо збереження майна та цінних паперів;

- при появі сторонньої особи, яка застосовує протиправні дії щодо його безпеки або оточуючих, викликати міліцію за телефоном 102;

- у випадку травмування себе, або інших працівників необхідно звернутися до медичного працівника, викликати швидку допомогу 103 або за необхідності надати першу долікарську допомогу, повідомити начальника відділу або особу, що його заміняє.

Надання першої медичної допомоги треба починати з оцінки загального стану потерпілого і на підставі цього скласти думку про характер пошкодження. У разі різкого порушення або відсутності дихання, зупинки серця негайно приступити до проведення штучного дихання та зовнішнього масажу серця, негайно викликати за телефоном 103 швидку медичну допомогу.

Дії при ураженні електричним струмом:

- необхідно негайно звільнити потерпілого від дії електричного струму,
- відключивши електрообладнання від джерела живлення, а при неможливості відключення – відтягнути його від струмопровідних частин за одяг або застосувавши підручний ізоляційний матеріал;

- при відсутності у потерпілого дихання і пульсу необхідно зробити йому штучне дихання і непрямий (зовнішній) масаж серця, звернувши увагу на зіниці. Розширені зіниці свідчать про різке погіршення кровообігу мозку. При

такому стані оживлення починати необхідно негайно, після чого викликати швидку медичну допомогу (103).

Дії при пораненні:

- для надання першої допомоги при пораненні необхідно розкрити індивідуальний пакет і накласти стерильний перев'язувальний матеріал;
- якщо індивідуального пакета не буде, то для перев'язки необхідно використати чисту тканину (носову хустинку тощо). На те місце тканини, що приходить безпосередньо на рану, бажано накапати декілька капель розчину йоду, щоб одержати пляму розміром більше рани, а після цього накласти тканину на рану. Особливо важливо застосовувати розчин йоду зазначеним чином при забруднених ранах.

Дії при переломах, вивихах, ударах, розтягненні:

- при переломах і вивихах необхідно пошкоджену кінцівку закріпити шиною, фанерною пластинкою, палицею, картоном або іншим подібним предметом. Пошкоджену руку можна також підвісити за допомогою перев'язки або хустки до шиї і прибинтувати до тулуба;
- при передбачуваному переломі черепа (несвідомий стан після удару голови, кровотеча з вух або рота) необхідно прикласти до голови холодний предмет (грілку з льодом, снігом чи холодною водою) або зробити холодний компрес;
- при підозрі перелому хребта необхідно потерпілого покласти на дошку, не підіймаючи його, чи повернути потерпілого на живіт обличчям униз, наглядаючи при цьому, щоб тулуб не перегинався, з метою уникнення пошкодження спинного мозку;
- при переломі ребер, ознакою якого є біль при диханні, кашлі, чханні, рухах, необхідно туго забинтувати груди чи стягнути їх рушником під час видиху.

Дії при теплових опіках:

- при опіках вогнем, парою, гарячими предметами в жодному разі не можна відкривати пухирі, які утворюються, та перев'язувати опіки бинтом;

- при опіках першого ступеня (почервоніння) обпечене місце обробляють ватою, змоченою етиловим спиртом;
- при опіках другого ступеня (пухирі) обпечене місце обробляють спиртом, 3%-м марганцевим розчином або 5%-м розчином таніну;
- при опіках третього ступеня (зруйнування шкіряної тканини) накривають рану стерильною пов'язкою та викликають лікаря.

Дії при кровотечі:

- для того, щоб зупинити кровотечу, необхідно підняти поранену кінцівку вгору, кровоточиву рану закрити перев'язувальним матеріалом, складеним у клубочок, придавити її зверху, не торкаючись самої рани, потримати протягом 4-5 хв; якщо кровотеча зупинилася, то, не знімаючи накладеного матеріалу, поверх нього покласти ще одну подушечку з іншого пакета чи шматок вати і забинтувати поранене місце (з деяким натиском);
- при сильній кровотечі, яку не можна зупинити пов'язкою, застосовується здавлювання кровоносних судин, які живлять поранену область, за допомогою згинання кінцівок у суглобах, а також пальцями, джгутом або закруткою; при великій кровотечі необхідно терміново викликати лікаря.

2 ОПИС ПРОЕКТУ «ПЛАТФОРМА ПОБУТОВОЇ АВТОМАТИЗАЦІЇ»

2.1 Загальні відомості

Інтелектуальні автоматизовані системи, у тому числі і системи типу «Розумний будинок» зараз є одним із найпоширеніших напрямків сучасної електроніки. Проте якщо спробувати у домашніх умовах створити якусь автоматизовану систему то при цьому виникне ціла низка технічних труднощів. Зараз подібні системи можна створити або на основі дорогих готових рішень, або створювати їх власноруч на основі доступної елементної бази.

Зараз найбільш придатною платформою для домашньої автоматизації є платформа Arduino, проте вона має велику кількість недоліків, головними із яких є:

- необхідність знань в області електроніки;
- необхідність знань в області програмування.

Отже, для того, щоб створити якісь автоматизований пристрій на платформі Arduino потрібно самому скласти електричну схему та написати програмне забезпечення. Навіть якщо використовувати готові модулі та готове програмне забезпечення, то людині, що не має необхідної освіти, наприклад, фахівцю в галузі агротехніки (для того щоб створити, наприклад, автоматизовану теплицю) це буде важко. Наприклад, якщо необхідно визначити рівень освітленості, то можна використати модуль ВН-1750, але для цього слід спершу розібратися, що таке інтерфейс I2C, скласти самостійно електричну схему, розробити та завантажити програмне забезпечення на мові програмування Arduino. Зрозуміло, що не всі люди це зможуть самостійно зробити, тому платформа Arduino використовується у більшості випадків лише радіоаматорами. Крім того, вона зовсім не придатна для серійного копіювання пристроїв на її основі.

Наприклад, для того щоб використати датчик освітленості BH-1750 на платформі Arduino потрібно знати що таке інтерфейс I2C, скласти схему та завантажити програмне забезпечення (Рис. 2.1).

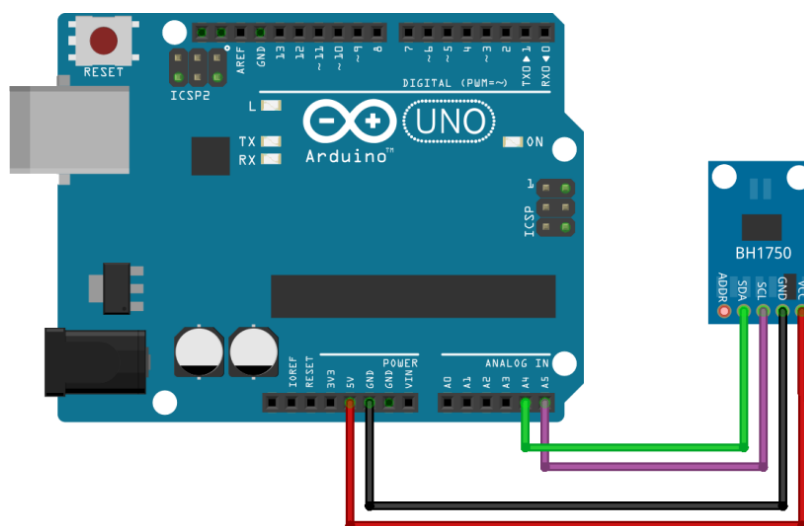


Рисунок 2.1 – Приклад використання датчика BH-1750 з платформою Arduino

Отже, якщо мова йде про прості на дешеві засоби домашньої автоматизації, доступні для широкого кола користувачів, то виявляється що подібної платформи зараз зовсім немає.

2. Мета проекту

Створити просту, зручну та дешеву платформу для домашньої автоматизації, яку б могли використовувати широке коло фахівців без спеціальної освіти у галузі електроніки та програмування.

2.2 Цільова аудиторія

Цільовою аудиторією проєкту «Платформа побутової автоматизації» є:

- люди, що зацікавлені в автоматизації домашніх та побутових процесів;
- радіоаматори;

- школярі;
- представники малого та середнього бізнесу, які не можуть дозволити собі придбання дорогих промислових автоматизованих систем;
- науковці та аматори, що працюють з пошуку нових технології виробництва.

4. Технічна реалізація

Платформа буде складатися із набору повністю готових до використання модулів (на кшталт Arduino), проте, на відміну від Arduino, ці модулі

- не потребують втручання в електричну схему (модулі вже містять усі компоненти необхідні для роботи);

- не потребують розробки програмного забезпечення (програмне забезпечення вже завантажено у мікроконтролер модуля, а для керування та налаштування системи також використовується готове програмне забезпечення).

З'єднання модулів у систему буде відбуватися наступним чином. Усі модулі підключаються у єдину шину, що містить 4 проводи: загальний провід, провід живлення, два проводи для передачі даних (використовується інтерфейс RS-485).

Перевагами такого рішення є:

- широкий діапазон напруги живлення – від 6 до 30 В (у базовому варіанті – 12 В), при цьому кожен модуль має власний стабілізатор напруги 5 В;
- велика відстань між модулями (до 1 км при використанні швидкості обміну даними 9600 бод);
- велика кількість модулів на одній шині (до 32).

Кожен модуль має власний адрес на шині, тому у системі може бути кілька однотипних модулів із різними адресами (адрес можна змінити у налаштуваннях).

Прикладами практичного використання платформи побутової автоматизації можуть бути автоматизована системи для вирощування рослин (Рис. 2.2), автоматизованої системи для керування освітленням в аудиторіях (Рис. 2.3) або автоматизованої системи подачі дзвінків на заняття (Рис. 2.4).

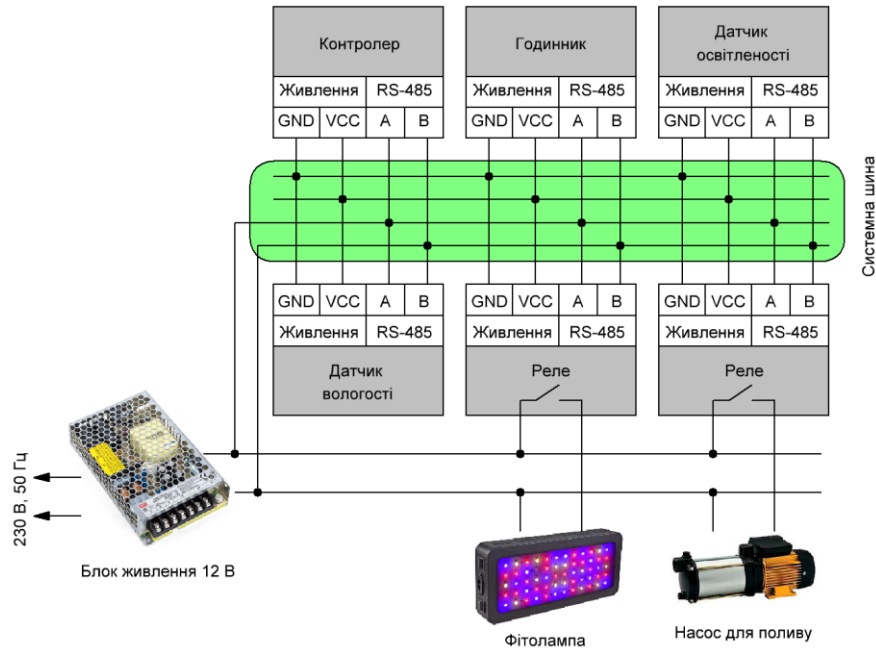


Рисунок 2.2 – Приклад інтелектуальної автоматизованої системи для вирощування рослин

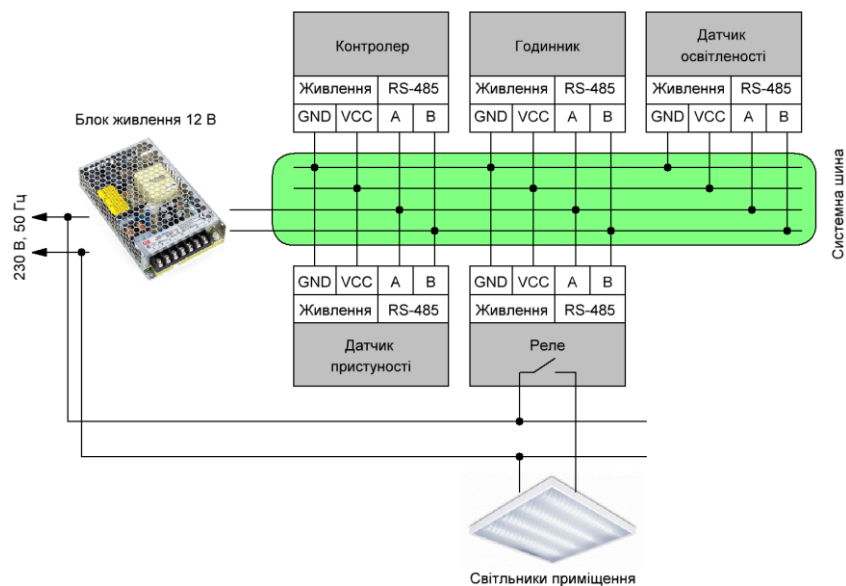


Рисунок 2.3 – Приклад інтелектуальної автоматизованої системи для керування освітленням в аудиторіях

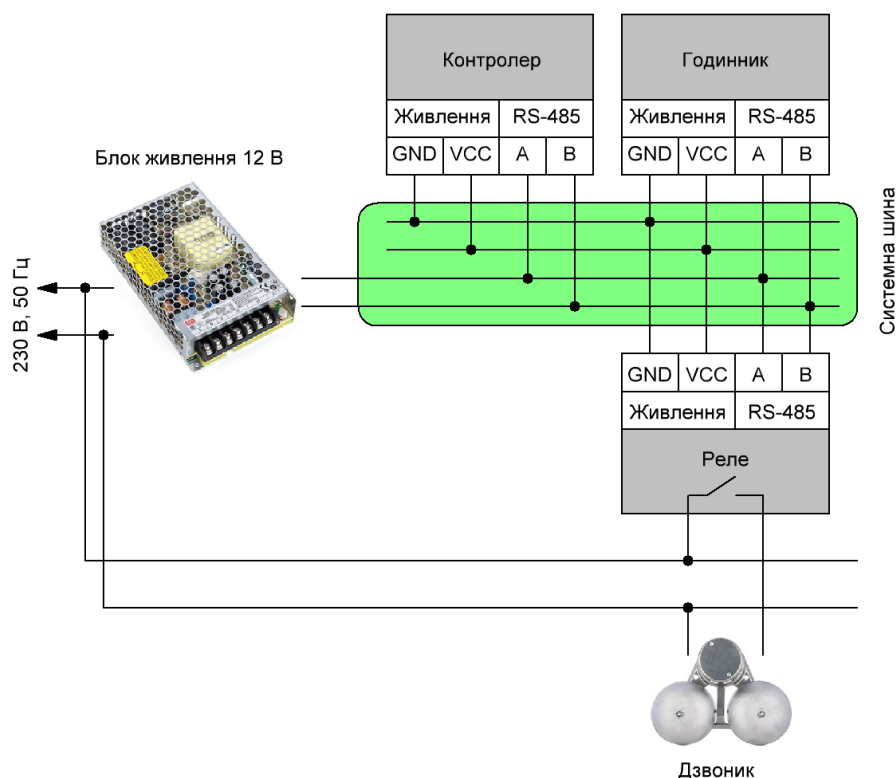


Рисунок 2.4 – Приклад інтелектуальної автоматизованої системи подачі дзвінків на заняття

2.3 Головні переваги проекту

1. Низька вартість модулів. Орієнтовна роздрібна ціна більшості модулів при серійному виробництві – приблизно 100 грн (ціни 2021 року)
2. Модулі можна замовляти у Китаї (ручна збірка через малі розміри компонентів дуже утруднена), на місці залишається завантажувати лише програмне забезпечення у мікроконтролер.
3. Відкритий та простий протокол обміну даними, що дозволить долучити до цього проекту сторонніх виробників.
4. Швидкий старт проекту – випуск першої дослідної партії можна розпочати вже через кілька місяців.
5. Можливість масштабування та модифікування системи – у будь яку систему можна добавляти нові модулі та розширювати її функціональність.

6. Можливість розширення переліку модулів без зміни вже існуючих проектів.

7. Гнучкість – на основі готових модулів можна створити будь яку систему.

8. Швидке створення нових модулів – від 1 тижня, програмне забезпечення (прошивка мікроконтролера) є типовою тому потребує лише мінімальної доробки та реалізації унікальних функцій.

3 МОДУЛЬ РЕЛЕ

3.1 Електрична схема

Електрична схема розробленого мною модуля реле показана на Рис. 3.1. Приймачем інтерфейсу RS-485 є мікросхема DA2 (SP485). Вона пов'язана з мікроконтролером DD1 (STM8S003F3P6) за допомогою трьох ліній, дві з яких з'єднані з інтегрованим модулем UART (висновки 2 і 3), а третя – з'єднана з висновком 1, налаштованим у режимі звичайного порту введення-виводу – використовується для управління приймальною частиною SP485.

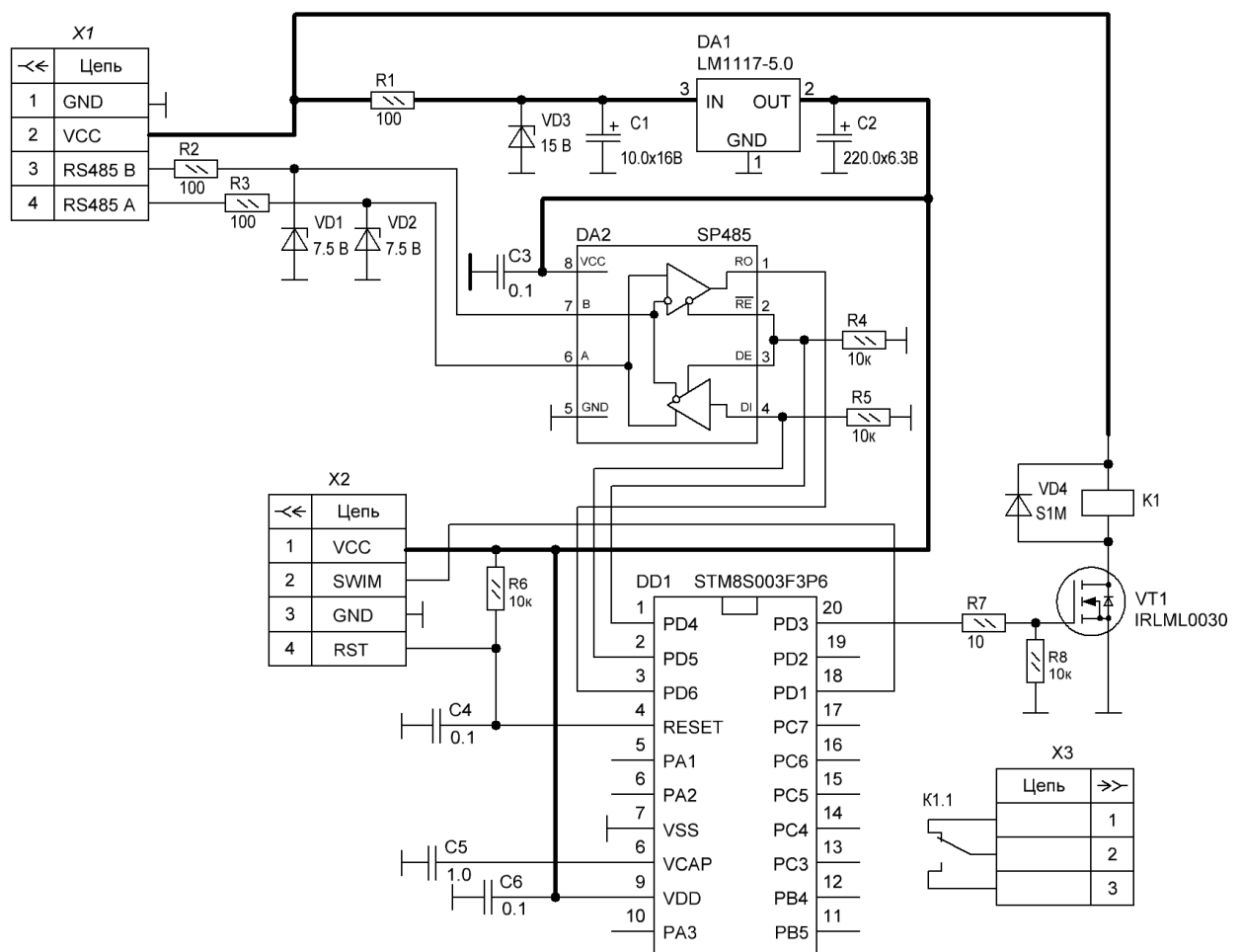


Рисунок 3.1 – Електрична принципова схема модуля реле

У нормальному режимі роботи мікроконтролер і приймач RS-485 живляться напругою 5 В, що формується стабілізатором DA1. Однак через те, що дешеві версії програматора ST-LINK можуть працювати тільки з сигналами, рівень яких дорівнює 3,3 В, під час програмування та налагодження програмного забезпечення живити мікроконтролер і основні мікросхеми доводиться напругою 3,3 В, що отримується від програматора через роз'єм X2. Однак це навіть виявилось зручним, оскільки при налагодженні всю систему можна живити від одного USB-порту і не використовувати лабораторні джерела живлення. При розробці можна було б використовувати і напругу живлення 3,3 В, проте мікросхеми, розраховані на роботу при такій напрузі, наприклад, MAX3485 або LM1117-3.3, чомусь коштують у кілька разів дорожче за свої 5-вольтові аналоги.

Інші елементи є «стандартною обв'язкою», встановленою відповідно до технічної документації на вибрані мікросхеми або на основі загальних принципів проектування радіоелектронних пристроїв.

3.2 Протокол обміну даними

Обмін даними у системі відбувається у формі діалогу. Мастер-пристрій (модуль контролера) періодично надсилає кожному модулю команду у певному форматі (Рис. 3.2).

Байт 0	Байт 1	Байт 2		Байт N
Розмір пакету	Номер модуля	Код команди	Дані (необов'язково)	Контрольна сума

Рисунок 3.2 – Формат команди від контролера до модулів

Під час роботи всі відомі модулі «слухають» інформаційну шину і при виявленні пакета, адресованого конкретному модулю (при збігу поля «Номер пристрою» з налаштуванням, записаним у власній енергонезалежній пам'яті), передають контролеру відповідь також у фіксованому форматі (Рис 3.3).

Байт 0	Байт 1	Байт 2		Байт N
Розмір пакету	Номер модуля	Код помилки	Дані (необов'язково)	Контрольна сума

Рисунок 3.3 – Формат відповіді модуля контролеру

Як очевидно з рисунка, структура відповіді модуля відрізняється від структури запиту контролера лише одним полем – кодом помилки. Код помилки є критерієм правильної роботи модуля та коректності переданих даних. Наприклад, якщо в модулі годинника, стався збій у роботі мікросхеми RTC (Real Time Clock), то при запиті часу технічно простіше передати останні дані, отримані від цього вузла, з кодом «Апаратна помилка», ніж інакше сигналізувати про наявність несправності. Контролеру цього буде цілком достатньо для того, щоб «зрозуміти», що отриманим даним про час вірити не можна, і вжити заходів щодо відновлення роботи системи (наприклад, перезавантажити аварійний модуль і, якщо помилка не зникла, активізувати аварійний світлодіод).

3.3 Ініціалізація модуля UART

Вихідний код розробленого програмного модуля приведений у Додатку А. Особливості апаратної частини модуля UART мікроконтролерів STM8 та його роботи у всіх можливих режимах досить добре описані як у технічній документації, так і в прикладах. При використанні бібліотеки StdPeriph_Lib для налаштування модуля найпростіше використовувати функцію UART1_Init, передавши в неї значення у вигляді наперед визначених констант. У цьому прикладі модуль UART переводиться в режим: асинхронний режим роботи (використовуються тільки висновки RX і TX), швидкість обміну даними – 9600 бод, довжина пакета – 8 біт, один стоповий біт, без біта парності. При ініціалізації модуля також відбувається включення і приймача, і передавача – за

це «відповідає» останній вхідний параметр, якому присвоюється значення UART1_MODE_TXRX_ENABLE.

3.4 Обробка переривань модуля UART при передаванні даних

На початку передачі ми активізували переривання від усіх подій, пов'язаних із процесом передачі, тому обробник цього переривання буде викликатися і за спустошенні буфера, і після закінчення передачі байту. Згідно з технічною документацією, прапор SR.TC (завершення передачі) може бути піднятий тільки при піднятому прапорі SR.TXE (реєстр DR порожній). Це означає, що подія SR.TC має відбутися лише один раз – наприкінці передачі пакета, коли буде сформовано стоп-біт останнього байт (Рис 3.4). Тому в цьому обробнику перевіряти прапори переривання за допомогою функції UART1_GetITStatus є сенс лише у випадку, коли переривання від UART можуть відключатися на тривалий час. У системі таке малоймовірно, тому подібна перевірка не реалізовувалася.

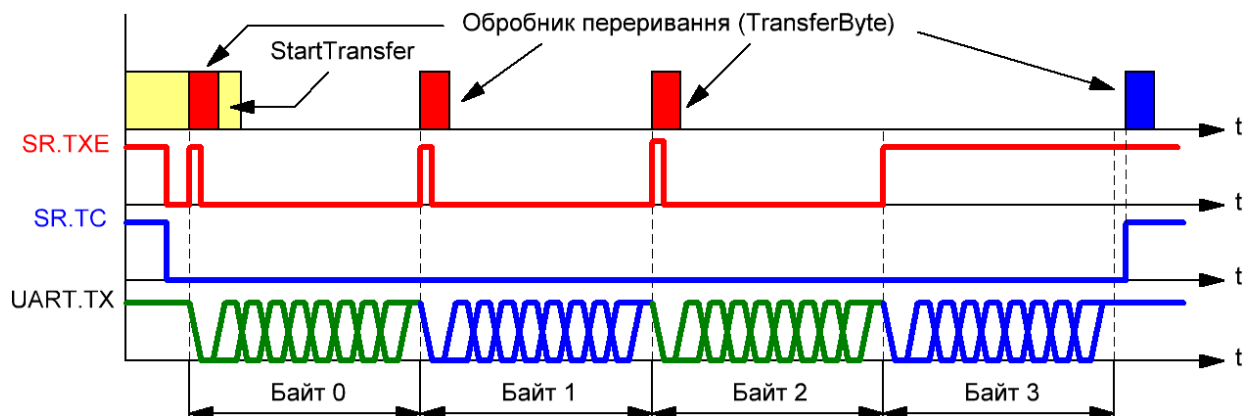


Рисунок 3.4 – Процеси, що відбуваються при передачі пакета (4 байти)

У цьому випадку алгоритм обробки переривання під час передачі даних визначається виключно кількістю вже переданих байт, що міститься в змінній DataCount. Якщо значення цієї змінної менше розміру пакета, що зберігається в

нульовому елементі буфера Tx, ми копіюємо в регістр DR наступний байт, після чого збільшуємо на одиницю змінну DataCount. Після цього нам потрібно знову порівняти її значення розміром пакета. Якщо виявилось, що ми відправили на передачу останній байт, то переривання по прапорі SR.TXE потрібно відключити, адже щоб його скинути наступного разу, потрібно буде щось записати в регістр DR, а ми вже записали все, що було потрібно. .

Після відключення переривання TXE обробник буде викликаний ще раз, але вже в кінці передачі пакета, коли буде сформований стоп-біт останнього байта (Малюнок 7). У цьому випадку переривання буде гарантовано викликане прапором SR.TC, а змінна DataCount дорівнюватиме розміру пакета, тому нам не залишається нічого іншого, як вимкнути передавач RS-485, відключити останнє активне переривання і, піднявши прапор IsTransferComplete, вказати високорівневим функціям, що передача пакета завершено.

3.5 Обробка переривань модуля UART при прийомі даних

Обробка переривання, що виникає під час прийому даних, починається зі скидання прапорів. Тут знову потрібно суворо дотриматись певної послідовності дій: спочатку читається регістр SR, а потім DR. Так само, як і в підпрограмі ResetReceiver, я зберіг значення цих регістрів у змінні UART_Flags і UART_Data, хоча в цьому випадку тип змінних – локальні чи глобальні – не є важливим.

Оброблювач переривання при прийомі даних може бути викликаний трьома подіями: прийомом байта, переповненням вхідного буфера та припиненням надходження даних. При перших викликах обробника – коли значення змінної DataCount дорівнює нулю – переривання при виявленні простою лінії відключено, тому, потрапивши у цей обробник, ми, у разі, щось прийняли.

Насамперед потрібно перевірити апаратні прапори SR.NF (зашумлена передача) та SR.FE (помилка кадру). Встановлення хоча б одного з них свідчить

про наявність проблем на апаратному рівні, наприклад, погану пайку або невідповідність швидкостей обміну приймача та передавача. Наявність подібних помилок є вагомою основою ігнорування всіх прийнятих даних, оскільки вони, з великою ймовірністю, є некоректними.

Не найкращою подією є також і встановлення прапора SR.OR (переповнення вхідного буфера). У технічній документації зазначено, що одразу після встановлення цього прапора дані ще можна отримати в повному обсязі, проте модуль UART не має жодних інструментів для визначення кількості вже втрачених байт. Тому під час встановлення прапора SR.OR найкраще проігнорувати все повідомлення. А якщо цей прапор часто підніматиметься, доведеться переглянути все програмне забезпечення – схоже, мікроконтролер «не встигає» обробляти потік даних з потрібною швидкістю.

І апаратні, і програмні проблеми призведуть до часткової втрати даних, через що подальша обробка отриманої інформації буде неможлива. Тому якщо хоча б один із прапорів SR.NF, SR.FE або SR.OR піднято, приймач переводиться у вихідний стан шляхом виклику підпрограми ResetReceiver.

Прийом першого байта ніяк не гарантує того, що ми взяли початок повідомлення, адже приймач може "включитися" у довільний момент, наприклад, на середині пакета, адресованого іншому модулю. Достовірно визначити, що байт є першим і містить інформацію про довжину інформаційної посилки неможливо. Навіть якщо ввести спеціальний маркер початку передачі, наприклад, 0xFF, і слідкувати за тим, щоб коди команд або параметрів не набували цього значення, все одно існує висока ймовірність наявності подібних байтів у секціях даних або контрольної суми. А використання складніших маркерів призведе до значного збільшення «накладних» витрат на обмін даними, адже за мінімальної довжини команди 4 байти додавання навіть однобайтного маркера призведе до збільшення довжини повідомлення на 25%.

Тому в нашому випадку визначити, що ми правильно синхронізувалися з передавачем можна лише наприкінці прийому – під час аналізу контрольної суми. При прийомі першого байта це принципово неможливо, тому на самому

початку - коли змінна DataCount дорівнює нулю - проводиться лише примітивна перевірка: мінімальна довжина повідомлення не може бути меншою за чотири байти, тому якщо перший байт більше трьох, то вважаємо, що передача почалася. У цьому випадку ми зберігаємо байт у приймальний буфер, збільшуємо на одиницю змінну DataCount і включаємо переривання при зупинці обміну.

Увімкнення переривання по прапорі SR.IDLE є дуже важливим. Воно дозволить приймачеві «не зависнути» у разі порушення загальної синхронізації. Наприклад, нехай приймач почав «слухати» шину наприкінці повідомлення і прийняв байт, зі значенням, рівним 255 (Рис 3.5). Нехай, відразу після передачі цього байта на інформаційній шині настала тиша, що може статися, наприклад, якщо пакет призначався цього пристрою і тепер ведучий чекає від нього відповіді. Однак, приймач, для того, щоб розпочати обробку до пакета, повинен прийняти ще 254 байти, яких за інформаційною шиною ніхто не передає. В цьому випадку через деякий час, зазвичай дорівнює тривалості одного байта, в модулі UART підніметься прапор SR.IDLE і відбудеться переривання, в даному випадку через припинення передачі інформації.

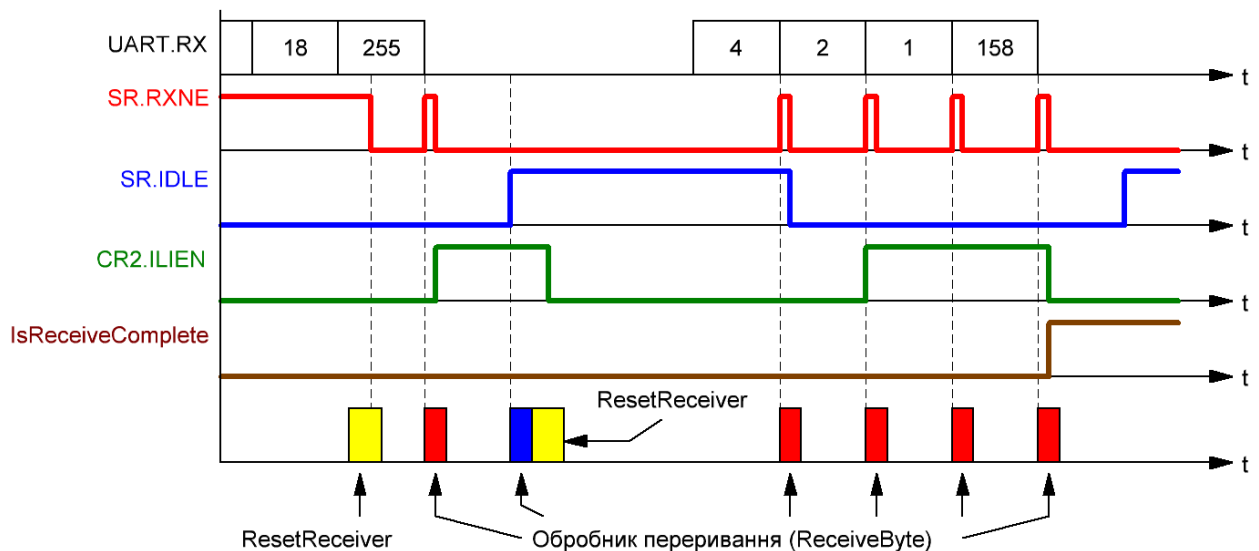


Рисунок 3.5 – Процеси, що відбуваються при прийомі пакета (4 байти)

Оскільки повідомлення було прийнято не до кінця, то приймач повернеться у вихідний стан і при прийомі наступного байта, який з ймовірністю виявиться початком нового пакета, інформація буде прийнята коректно. Але якщо навіть тиша на інформаційній шині не настане, і приймач все-таки прийме 254 байти, то повідомлення не матиме потрібну структуру і розрахункове значення контрольної суми з високою ймовірністю не збігається зі значенням, що зберігається в останньому байті.

Таким чином, синхронізація передавача та приймачів на рівні інформаційних пакетів рано чи пізно відбудеться у будь-якому випадку, просто активізація переривання при виявленні простою в лінії дозволить прискорити настання цієї події. Якщо приймач приймає інформацію синхронно з передавачем, то вимкнення всіх переривань і встановлення прапора `IsReceiveComplete` відбудеться відразу після прийому останнього байта – задовго до установки волога `SR.IDLE` і переведення приймача у вихідний стан.

3.6 Практична реалізація та тестування

Зовнішній вигляд друкованої плати модуля реле показаний на Рис. 3.6. Ця плата була інтегрована у дослідний зразок макета установки, призначеної для вирощування рослин (Рис 3.7). Результати тестування показали повну відповідність функціонування модуля поставленому технічному завданню.

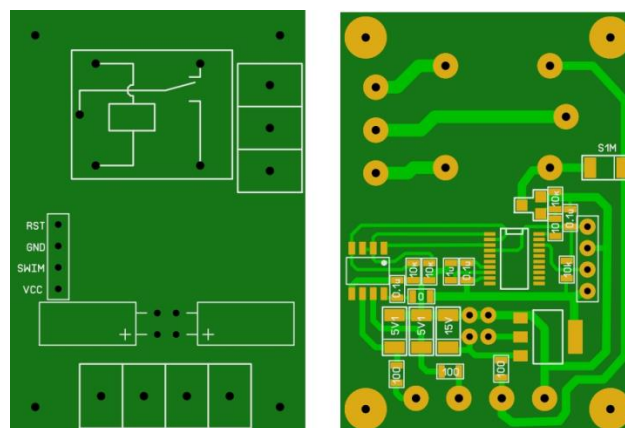


Рисунок 3.6 – Зовнішній вигляд друкованої плати модуля реле



Рисунок 3.7 – Дослідний зразок макета установки, призначеної для вирощування рослин

ВИСНОВКИ

Основними результатами проходження практики на підприємстві ФОП «Інформаційні технології» є наступні.

1. Проведено знайомство із підприємством, його структурою та особливостями функціонування.

2. Проведено знайомство із особливостями роботи інженера-програміста мікроконтролерів, у тому числі посадовими інструкціями та інструкціями з охорони праці.

3. Отримано практичні навички:

- розробки принципів схем електронних пристроїв;
- розробки друкованих плат;
- розробки програмного забезпечення для мікроконтролерів STM8;
- тестування та налаштування програмного забезпечення.

4. У результаті проходження практики:

– розроблена принципова схема модуля реле для проекту «Платформа побутової автоматизації»;

– розроблена друкована плата модуля реле для проекту «Платформа побутової автоматизації»;

– розроблено програмне забезпечення для мікроконтролерів STM8 для плати модуля реле проекту «Платформа побутової автоматизації».

5. Під час проходження практики також були отримані навички роботи в команді, спілкування державною мовою та роботи із технічною документацією.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Інформація з сайту ФОП «Інформаційні технології» URL: <https://www.infotech.com.ua/>.
2. Інформація з сайту IAR Systems URL: <https://www.iar.com/>.
3. Інформація з сайту STMicroelectronics URL: <https://www.st.com/>.

ДОДАТОК А

Вихідний код програмного модуля UART модуля реле

```
/*
*****
* @file      UART_Communicator.c
* @version   1.7
* @date      25.12.2022
*****
**
*/

#include "UART_Communicator.h"

extern bool GetDeviceInfo(TDeviceInfo* C, uint8_t Index);
extern bool GetSetting(TSetting* V, uint8_t Index);
extern bool GetVariable(TVariable* V, uint8_t Index);

extern int32_t DeviceNumber;
extern char ErrorMessage[64];

const uint8_t ER_OK = 0x00;
const uint8_t ER_HARD_ERROR = 0x01;
const uint8_t ER_SOFT_ERROR = 0x02;
const uint8_t ER_LOGICAL_ERROR = 0x03;
const uint8_t ER_INVALID_PARAM = 0x04;
const uint8_t ER_INVALID_COMMAND = 0xFF;

const uint8_t UNIVERSAL_DEVICE_NUMBER = 0xFF;
const uint8_t BROADCAST_DEVICE_NUMBER = 0x00;

uint8_t UART_Flags;
uint8_t UART_Data;

//
#####
##

#define LED_ON GPIO_WriteHigh(P->LED_Port, P->LED_Pin)
#define LED_OFF GPIO_WriteLow(P->LED_Port, P->LED_Pin)

void Communicator_InitVariables(TCommunicator* P)
{
    P->OperationMode = omSlave;

    P->LED_Port = NULL;
    P->LED_Pin = GPIO_PIN_0;

    P->BaudRate = 9600;

    P->RS485_Port = NULL;
    P->RS485_Pin = GPIO_PIN_0;

    Communicator_ResetReceiver(P);
}
```

```

//
#####

void Communicator_InitModules(TCommunicator* P)
{
    UART1_DeInit();

    UART1_Init(P->BaudRate,
                UART1_WORDLENGTH_8D,
                UART1_STOPBITS_1,
                UART1_PARITY_NO,
                UART1_SYNCMODE_CLOCK_DISABLE,
                UART1_MODE_TXRX_ENABLE);

    if (P->RS485_Port != NULL)
        GPIO_Init(P->RS485_Port, P->RS485_Pin, GPIO_MODE_OUT_PP_LOW_FAST);

    if (P->LED_Port != NULL)
        GPIO_Init(P->LED_Port, P->LED_Pin, GPIO_MODE_OUT_PP_LOW_SLOW);
}

//
#####

void Communicator_ResetReceiver(TCommunicator* P)
{
    if (P->RS485_Port != NULL)
        GPIO_WriteLow(P->RS485_Port, P->RS485_Pin);

    P->DataCount = 0;

    UART_Flags = UART1->SR;
    UART_Data = UART1->DR;

    UART1_ITConfig(UART1_IT_RXNE_OR, ENABLE);
    UART1_ITConfig(UART1_IT_IDLE, DISABLE);
}

//
#####

void Communicator_StartTransfer(TCommunicator* P)
{
    if (P->LED_Port != NULL)
        LED_ON;

    if (P->RS485_Port != NULL)
        GPIO_WriteHigh(P->RS485_Port, P->RS485_Pin);

    UART1->DR = P->Tx[0];
    P->DataCount = 1;

    UART1_ITConfig(UART1_IT_TC, ENABLE);
    UART1_ITConfig(UART1_IT_TXE, ENABLE);
}

```

Навчальне видання

Русу Олександр Петрович

ОФОРМЛЕННЯ ЗВІТУ З ПРАКТИКИ

Методичні рекомендації для самостійної роботи
здобувачів здобувачів фахової передвищої освіти
зі спеціальності 123 Комп'ютерна інженерія

Українською мовою