

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Катін П.Ю.

СИСТЕМИ РАДІОКЕРУВАННЯ ЛАБОРАТОРНИЙ ПРАКТИКУМ

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня магістра
Спеціальність – 172 Електронні комунікації та радіотехніка
За освітньою програмою
Радіотехнічні комп'ютеризовані системи

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензенти

Долголенко Олександр Миколайович, канд. техн. наук, доц. кафедри обчислювальної техніки, ФІОТ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Відповідальний редактор

Жук С.Я., д.т.н., проф., зав. каф. РТС РТФ

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 8 від 02.06.2023 р.) за поданням Вченої ради радіотехнічного факультету (протокол № 05/2023 від 28.04.2023 р.)

Електронне мережне навчальне видання

Автор:

Катін Павло Юрійович, канд. техн. наук, доц.

СИСТЕМИ РАДІОКЕРУВАННЯ

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Системи радіокерування. Лабораторний практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 172 «Електронні комунікації та радіотехніка» / КПІ ім. Ігоря Сікорського; уклад.: П.Ю. Катін. – Електронні текстові дані (1 файл: 2,67 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2023. – 77 с.

В навчальному посібнику розміщено завдання на лабораторні роботи, що включають пояснення програмної і апаратної частини систем радіокерування повітряними і наземними рухомими платформами (моделями, об'єктами). Надані базові пояснення щодо їх розв'язання. Визначений порядок оформлення результатів лабораторних робіт.

Для частини лабораторних робіт представлений інформаційний матеріал.

Для студентів всіх форм навчання, які навчаються у магістратурі за спеціальністю 172 «Електронні комунікації та радіотехніка» радіотехнічного факультету та інших споріднених спеціальностей.

© КПІ ім. Ігоря Сікорського, 2023

© Катін Павло Юрійович

ЗМІСТ

ПЕРЕДМОВА	5
1. ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ	7
2. ЛАБОРАТОРНА РОБОТА № 1 Проектування, розробка, налагодження, документування базових прототипів систем радіокерування	8
2.1 Основні завдання лабораторної роботи №1.	8
2.2 Теоретичний матеріал для лабораторної роботи №1.	10
2.3. Контрольні запитання до лабораторної роботи № 1	16
3. ЛАБОРАТОРНА РОБОТА № 2 Основи програмування клавіатури для пульта управління системи радіокерування	17
3.1. Основні завдання лабораторної роботи №2.	17
3.2. Теоретичний матеріал для лабораторної роботи №2.	19
3.3. Контрольні запитання до лабораторної роботи № 2	30
4. ЛАБОРАТОРНА РОБОТА № 3 Базові паттерни (шаблони) і архітектура програмної складової систем радіокерування. Програмування мікросхем радіомодулів	32
4.1. Основні завдання лабораторної роботи №3.	32
4.2. Теоретичний матеріал для лабораторної роботи №3.	34
4.3. Контрольні запитання до лабораторної роботи № 3	37
5. ЛАБОРАТОРНА РОБОТА № 4 Управління силовими елементами прототипів колісних роботів у режимі реального часу у системі радіокерування	38
5.1. Основні завдання лабораторної роботи №4.	38
5.2. Теоретичний матеріал для лабораторної роботи №4.	40
5.3. Контрольні запитання до лабораторної роботи № 4	44
6. ЛАБОРАТОРНА РОБОТА № 5 Використання АЦП і ЦАП для реалізації функціонального вузла для програмування джойстиків систем радіокерування	45
6.1 Основні завдання лабораторної роботи №5.	45

6.2 Теоретичний матеріал для лабораторної роботи №5.	47
6.3. Контрольні запитання до лабораторної роботи № 5	54
7. ЛАБОРАТОРНА РОБОТА № 6 Управління силовими елементами прототипів у плавному режимі зміни потужності. Управління серво у повітряних моделях систем радіокерування	56
7.1. Основні завдання лабораторної роботи №7	56
7.2. Контрольні запитання до лабораторної роботи № 6	58
8. ЛАБОРАТОРНА РОБОТА № 7 Налагодження системи радіокерування для повітряних моделей	59
8.1. Основні завдання лабораторної роботи № 7	59
8.2 Контрольні запитання до лабораторної роботи № 7	62
9. ЛАБОРАТОРНА РОБОТА № 8 Основи технології програмування, розробки і налагодження системи радіокерування у варіанті клієнт-сервер	63
9.1. Основні завдання лабораторної роботи № 8	63
9.2. Контрольні запитання до лабораторної роботи № 8	64
10. ЗАГАЛЬНІ РЕКОМЕНДАЦІЇ ДЛЯ ОФОРМЛЕННЯ ЗВІТУ	65
СПИСОК ЛІТЕРАТУРИ	67
ДОДАТКИ	71

ПЕРЕДМОВА

Метою створення даного посібника є підготовка фахівця, здатного вирішувати задачі розробки, тестування і супроводження систем радіокерування наземних (промислових) об'єктів та повітряних моделей.

Під час виконання лабораторних робіт досліджуються системи, що забезпечують керування наземними і повітряними об'єктами за допомогою програмованих радіозасобів. Особливістю курсу є використання систем управління, що містять програмну і апаратні складові. Даний навчальний посібник дозволяє, у ході лабораторних досліджень, отримати знання і закріпити навички з наступних питань:

- проектування розповсюджених та перспективних систем радіокерування на рівні структурної схеми;

- розробки типових структурних і функціональних схем систем радіокерування для промислових систем, наземних рухомих платформ, повітряних радіокерованих моделей;

- побудови архітектури програмної складової систем радіокерування;

- вдосконалення знань однієї з типових мов програмування для розробки програмної частини системи радіокерування (РК).

Дисципліну «Системи радіокерування» забезпечують дисципліни «Вища математика», «Основи програмування», та інші навчальні дисципліни, що зв'язані з програмуванням вбудованих пристроїв. Для засвоєння дисципліни потрібен базовий рівень володіння англійською мовою не нижче А2.

Лабораторні роботи № 1–5 є обов'язковими для виконання і входять в блок основних (базових) завдань для отримання допуску до екзамену (або до заліку у особливий період). Далі студенти можуть виконати ЛР № 6, 7, 8 для отримання додаткових балів. Студентам прискореного курсу навчання з досвідом в програмуванні клієнт-сервер пропонується ЛР № 8 для отримання

додаткових балів, при умові здачі лабораторних робіт, що входять в блок основних завдань.

Більша частина теоретичного матеріалу, програмного коду, рисунків, що наведені у посібнику, є авторськими, а інша частина - запозичена укладачами із різноманітних джерел. Зокрема, матеріал, що належить до основ програмування мікроконтролерів запозичено з [1]. В якості основного списку літератури використані матеріали [1-5]. Додаткова інформація з програмних рішень, апаратного обладнання міститься у списку [1-26 д].

Виконання лабораторних робіт у даному курсі дозволяється у різних інтегрованих середовищах розробки Integrated Development Environment (IDE), що забезпечують можливості розробки програмного забезпечення для мікроконтролерів. Це зроблено з метою підтримання студентів, які мають практичний досвід програмування, для забезпечення можливості обрання того середовища розробки, яке буде найбільш зручним для відповідного студента. Дозволяється використання різних компіляторів, мов програмування, парадигм і стандартів.

Мова програмування (C або C++) для даного посібника визначена фактом її широкого застосування при розробці програм для мікроконтролерів та мікропроцесорів.

Основою навчального посібника є курс лабораторних робіт (ЛР [1]), що виконувався на факультеті інформатики та обчислювальної техніки у 2018-2020 році під керівництвом к.т.н. Катіна П.Ю. Урахований також досвід проведення курсу систем радіокерування у 2022 році (Радіотехнічний факультет).

1. ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ

Для зручності і швидкого розроблення програм у теперішній час активно використовують інтегровані середовища розробки або Integrated Development Environment (IDE). Основним інтегрованим середовищем розробки для виконання курсу лабораторних робіт є Microcontroller Development Kit (MDK), а саме Keil uVision для мікроконтролерів архітектури Arm Cortex-M (<https://www.keil.com>). Крім нього можуть бути обрані інші інтегровані середовища розробки, наприклад IAR Systems (<https://www.iar.com>). Також є доступним безкоштовний компілятор для ARM мікроконтролерів GNU Arm Embedded Toolchain (<https://developer.arm.com/downloads/-/gnu-rm>).

У тому випадку, коли у якості апаратної системи радіокерування використовується мінікомп'ютер (наприклад Orange Pi), доступні всі засоби програмування, що відповідають операційній системі Orange Pi.

Середовище Keil у даному курсі пропонується “корпоративним стандартом”, що містить набір утиліт для розробки, налагодження і програмування функціональних вузлів системи радіокерування на базі мікроконтролера. Система Keil uVision дозволяє працювати з проектами будь-якого ступеня складності, починаючи з програмної складової функціональних вузлів і закінчуючи програмуванням всієї системи радіокерування.

Вихідний код, що представлений у навчальному посібнику, протестований з використанням Keil uVision.

2. ЛАБОРАТОРНА РОБОТА № 1

Проектування, розробка, налагодження, документування базових прототипів систем радіокерування

Дана лабораторна робота передбачає проведення досліджень для набуття навичок проектування, розробки і налагодження основних прототипів систем РК. Структура цих систем містить базові складові, що характерні для наземних рухомих платформ з дистанційним управлінням, промислових систем і повітряних моделей.

Під час виконання досліджень надаються навички розробки структурних і функціональних схем систем РК з використанням різних графічних редакторів.

Надається можливість вивчення основ користування IDE Keil uVision для розробки програмної частини РК. Дозволяється використовувати іншу IDE при наявності досвіду і навичок у студентів і по узгодженню з викладачем.

2.1 Основні завдання лабораторної роботи №1

Вхідні дані ЛР 1

Для підготовки ЛР № 1 може бути використаний будь-який сучасний ПК з операційною системою Windows, Linux або Mac OS (Macintosh Operating System) з інстальованим IDE Keil uVision або іншим IDE при наявності досвіду і навичок у студентів. Для виконання лабораторної роботи, крім структурної (функціональної) схем, потрібно розробити програму, що формує сигнал на відповідному апаратному виході. Він має бути відображений на структурній (функціональній) схемі. Вихідні дані для програми показані у Таблиці 1. Для створення таблиці використовувався комплект налагодження, що описаний у Додатках на базі STM32F103C8. При наявності іншого комплекту налагодження дозволяється застосовувати його

по узгодженню з викладачем. Програму можна реалізувати у довільній парадигмі програмування з використанням різних бібліотек, що вивчалися на протязі попередніх семестрів.

Таблиця 1 – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Порт і апаратний номер виходу сигналу	Частота слідування імпульсів Гц/ ширина імпульсу ms	Кількість радіоканалів для управління
1	A7	10/ 5	2
2	B8	100/2	3
3	B9	20/10	4
4	B10	150/1	2
5	B11	80/8	3
6	A5	70/7	4
7	A6	60/9	2
8	A0	40/4	3
9	A1	120/2	4
10	A2	110/2	2
11	A3	170/2	3
12	A4	150/1	4

Вихідні дані ЛР 1

1. Структурна схема системи РК (за потребою розробляється функціональна схема). Структурна схема розробляється у вигляді креслення з вибором потрібних компонентів для реалізації подальших лабораторних робіт.

2. Вихідний код і проект програми, що формує сигнал, характеристики якого вказані в Таблиці 1. Зображення сигналів з осцилографу або скріншоти налагоджувача Keil, що підтверджують працездатність вихідного коду і

відповідність характеристик сигналів на виходах мікроконтролера, наведеним у Таблиці 1 значенням. Висновки по розділам лабораторної роботи: розробка вихідного коду, підготовка проекту, налагодження і тестування.

Завдання

1. Розробити структурну схему (за потребою функціональну), що відображає реальні порти вводу-виводу загального призначення комплектів налагодження, які передбачається використовувати для розробки програми. На схемі відобразити порти мікроконтролера, його назву, кількість радіоканалів відповідно до варіантів у таб.1. Для кожного радіоканалу підібрати послідовний інтерфейс мікроконтролера для програмного управління. Підібрати частоту передачі і вказати її на структурній схемі, обґрунтувати прийняте рішення.

2. Розробити програму для комплекту налагодження (Додаток1) на базі нескінченного циклу опитування (Polled loop system), або іншому варіанті, що виводить на апаратний вихід сигнал заданий у таб.1.

3. Переконатися, що розроблена програма дійсно видає сигнал відповідно до часових характеристик, що записані у таб. 1 з використанням апаратного комплекту налагодження або налагоджувача Keil uVision.

2.2 Теоретичний матеріал для лабораторної роботи №1

Системи РК широко розповсюджені у промисловості, індустрії розваг, та активно використовуються у інших галузях. Яскравим прикладом промислової системи дистанційного РК, є системи управління вантажними кранами. Оператор управління краном може знаходитися на відстані до 90 метрів від вантажу і успішно управляти краном. У цьому варіанті система РК вантажним краном включає в себе наступні базові елементи:

- стандартний пульт управління (він звичайно містить 6 командних кнопок

управління вантажем, кнопку аварійної зупинки, кнопку включення-виключення та інші елементи захисту);

- блок приймача, який встановлюється безпосередньо на крані і підключається до його систем. Таким чином просту структурну схему радіокерування можна розподілити на дві базові частини: систему дистанційного управління і пульт дистанційного управління, що зв'язані між собою через радіоканал. Варіант реалізації структурної схеми РК показаний на рис.1.

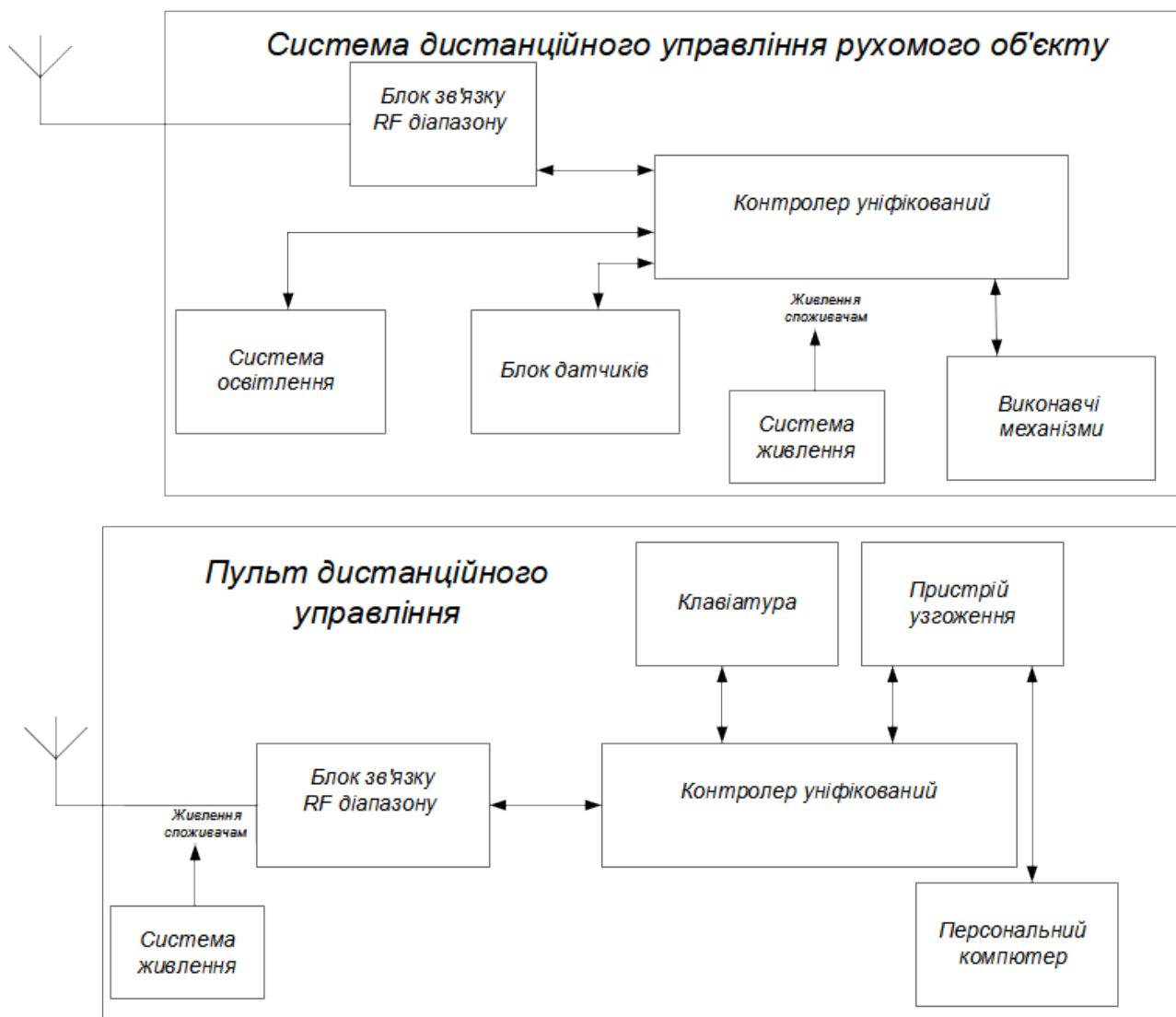


Рисунок 1 – Узагальнена структурна схема системи радіокерування (варіант)

На теперішній час подібні системи у більшості випадків реалізуються з використанням програмного управління. Тобто всю систему радіоуправління можна розділити на апаратну і програмну частину. Апаратна частина системи

РК показана на рис.1. Програмна складова системи РК призначена для управління блоком уніфікованого контролера, що зображений на рис.1. Більш докладно зміст уніфікованого контролера розкритий на функціональній схемі рис.2, що демонструє дуже спрощений модуль контролера, який містить сигнальний світлодіод і кнопку для управління. Всі ці елементи підключаються і управляються програмно через порти вводу-виводу загального призначення.

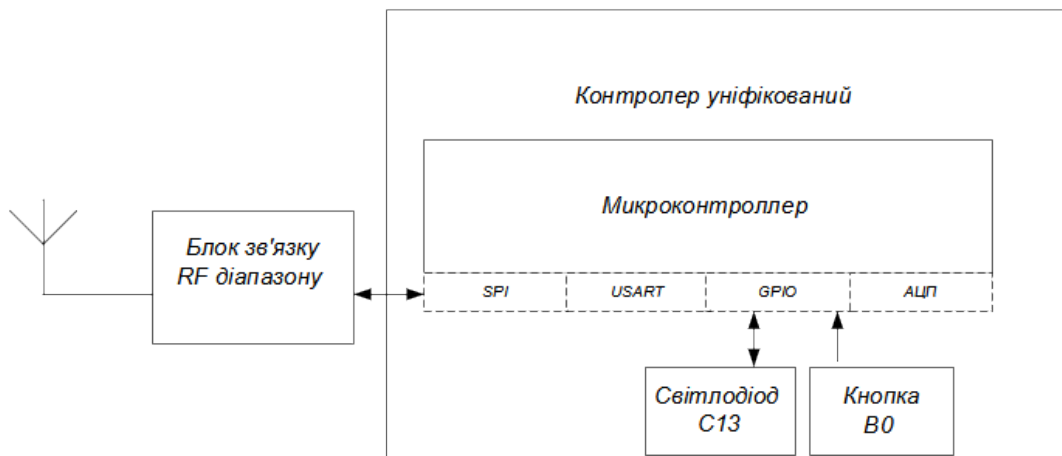


Рисунок 2 – Узагальнена структурна схема системи радіокерування (варіант)

Також на рис.2 показано, що блок радіодіапазону підключається до мікроконтролеру через порт SPI, який є найшвидшим портом управління у типовому стандартному мікроконтролері або мікрокомп'ютері. Системи РК призначені для забезпечення управління на основі одного або двохстороннього радіозв'язку. У більшості випадків радіоуправління промислових систем здійснюється на невеликих відстанях (short range device (SRD)), або на відстанях прямої видимості. Основна вимога до частотного діапазону таких систем полягає в тому, щоб вони не створювали перешкод іншим електронним системам і не заходили на ліцензійні частоти. Крім систем радіокерування наземними рухомими об'єктами, промисловими системами, повітряними пристроями такі системи використовуються для виконання наступних завдань:

- для дистанційного підключення датчиків,
- для роботи у системах охоронної сигналізації,

- у автомобільній промисловості, тощо.

На теперішній час актуальними для РК є частоти, що не ліцензуються, які можна використовувати для систем загального і промислового призначення без додаткового дозволу у спеціалізованих органах державного управління. До таких діапазонів відносяться:

- діапазон 13.56 МГц, що забезпечує зв'язок на малої відстані;
- діапазон 40 МГц, використовувався для роботи безшнурових телефонів;
- діапазон 433 МГц, розповсюджений діапазон, що забезпечував зв'язок на відстані 1...2 км і не вимагав особливих вимог для реалізації НВЧ пристроїв;
- діапазон 868/915 МГц, цей діапазон актуальний для Європи, США, Канади, Австралії;
- діапазон 2.4 ГГц, розповсюджений для стандарту IEEE 802.11 та Bluetooth і для промислової, на наукової та медичної галузі;
- діапазон 5.8 ГГц, для стандарту IEEE 802.11.

Звичайно, що кожний діапазон має певну смугу для переналаштування. Наприклад для захисту від побутових і промислових радіозавад у деяких діапазонах застосовується кроковий варіант перебудови частоти (frequency hopping (FH)). Для цього дозволений діапазон розподіляється на певну кількість частот (наприклад на 79). Далі передавач системи РК налаштовується на іншу частоту за псевдовипадковим алгоритмом з певною швидкістю. Діапазон 5.8 ГГц більш захищений з точки зору промислових радіозавад. Натомість на діапазоні 2.4 ГГц працює багато мікросхем, що реалізують програмоване радіокерування, мають невисоку вартість і добре керуються через інтерфейс SPI, що показано на рис. 2. Узагальнюючи діапазони радіокерування, що не ліцензуються і доступні для використання без спеціалізованого дозволу, можна відокремити клас Industrial, Scientific, Medical (ISM) до якого відносяться наступні діапазони:

- промисловий (Industrial), що має значення 902 – 928 MHz (ширина 26 MHz);
- науковий (Scientific), що має значення 2400 - 2500 MHz (ширина 100 MHz);

- медичний (Medical), що має значення 5725 - 5875 MHz (ширина 150 MHz). Всі перелічені діапазони можуть бути взаємозамінними. Вибір відповідних відкритих частотних діапазонів є одним із ключових завдань у ході проектування апаратних складових для радіоканалу системи управління. Для програмного управління модулем радіоканалу потрібно використовувати мікроконтролери. Далі представлений код для мікроконтролерів серії stm32f10x, що може бути основою для вирішення програмної складової лабораторної роботи 1 у варіанті нескінченного циклу опитування.

```
#include <stm32f10x.h>
#include <stm32f10x_gpio.h>
#include <stm32f10x_rcc.h>
//Out B15_, tay_ = 5, T_ = 20

void delayL1(int millisL1);
void initL1 (void);

int main(void)
{
    init();
    while (1) {

        GPIOB->ODR |= GPIO_Pin_15;
        delayL1 (5);
        GPIOB->ODR &= ~GPIO_Pin_15;
        delayL1 (15);
    }
}

void delayL1 (int millisL1) {
    for(int i=0;i< millisL1*0x010000;i++);
}

void initL1 (void) {

    GPIO_InitTypeDef GPIO_InitS;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    GPIO_InitS.GPIO_Pin = GPIO_Pin_15;
    GPIO_InitS.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitS.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, & GPIO_InitS);
    GPIO_SetBits(GPIOB, GPIO_Pin_15);
}
```

Далі представлений вихідний код, який демонструє програмне рішення, що реалізує побудову імпульсного сигналу на двох апаратних виходах з використанням переривань системного таймеру.

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

void delayL1(uint32_t nTimeL1);

int main(void)
{
    int iL1;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB|RCC_APB2Periph_GPIOA,
ENABLE);

    GPIO_InitTypeDef GPIO_InitL1;
    GPIO_InitL1.GPIO_Pin = GPIO_Pin_3|GPIO_Pin_11;
    GPIO_InitL1.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_InitL1.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &GPIO_InitL1);
    GPIO_Init(GPIOA, &GPIO_InitL1);

    if (SysTick_Config(SystemCoreClock / 1000))
    {
        GPIOB->ODR=GPIO_Pin_11;
        GPIOA->ODR=GPIO_Pin_3;
    }

    while (1)
    {
        // ON
        GPIOB->ODR=GPIO_Pin_11;
        GPIOA->ODR=GPIO_Pin_3;
        delayL1(5);
        //OFF
        GPIOB->ODR=~GPIO_Pin_11;
        delayL1(10);
        //OFF
        GPIOA->ODR=~GPIO_Pin_3;
        delay(30);
    }
}

static __IO uint32_t __TimingDelayL1;
```

```

void delayL1(uint32_t nTimeL1){
    __TimingDelayL1 = nTimeL1;
    while(__TimingDelayL1 != 0);
}

void SysTick_Handler(void){
    if (__TimingDelayL1 != 0x00)
        __TimingDelayL1 --;
}

```

2.3. Контрольні запитання до лабораторної роботи № 1

1. Які базові елементи структурної схеми мають сучасні програмовані системи радіокерування ?
2. Наведіть приклад технології розробки вихідного коду програмної складової системи радіокерування.
3. Які на теперішній час є частотні діапазони, що не ліцензуються і які можна використовувати для систем радіокерування загального і промислового призначення без спеціалізованого дозволу?
4. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 13.56 МГц.
5. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 40 МГц ?
6. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 433 МГц?
7. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 868/915 МГц?
8. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 2.4 ГГц ?
9. В чому полягають особливості, основне призначення, переваги і недоліки частотного діапазону 5.8 ГГц ?
10. Які архітектурні рішення можливі при створенні програми для мікроконтролера системи управління на основі ПК?

11. Які послідовні інтерфейси мікроконтролера і міні-комп'ютера найбільш часто використовуються для управління радіомодулем і чому?

3. ЛАБОРАТОРНА РОБОТА № 2

Основи програмування клавіатури для пульта управління системи радіокерування

Дана лабораторна робота реалізує проведення досліджень для набуття навичок проектування, розробки і налагодження систем на основі широтно-імпульсної модуляції (ШИМ) радіокерованих систем. Також розглядаються питання програмування клавіатури ПК систем, що є однаковими для наземних рухомих платформ з дистанційним управлінням, промислових систем радіоуправління.

Під час виконання досліджень надаються навички розробки структурних і функціональних схем РК з використанням ШИМ і клавіатури.

Продовжується вдосконалення навичок розробки програмної складової систем РК з питань використання IDE Keil uVision, або іншим IDE при наявності досвіду і навичок у студентів.

3.1. Основні завдання лабораторної роботи №2

Вхідні дані ЛР 2

Для підготовки ЛР № 2 може бути використаний сучасний ПК з операційною системою Windows, Linux або Mac OS. Для виконання лабораторної роботи потрібно доробити структурну (і функціональну) схему, що розроблена на лабораторній роботі 1. У ході доопрацювання схем потрібно додати клавіатуру і відповідні елементи, що передбачені у завданні. Вихідні дані для програми показані у Таблиці 2. Для створення таблиці використовувався комплект налагодження, що описаний у Додатку 1 на базі STM32F103C8. При наявності іншого комплекту налагодження дозволяється

застосовувати його по узгодженню з викладачем. Програму можна реалізувати у довільній парадигмі програмування з використанням різних бібліотек, що вивчалися на протязі попередніх семестрів.

Таблиця 2. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Порти і апаратні номери для виходу ШІМ сигналу	Вихідне значення сигналів. Частота слідування імпульсів Гц / ширина імпульсу мс	Діапазон зміни значень, (номер кнопки на клавіатурі)
1	A7, B8	50/ 5	10%-70% (1)
2	B8, B9	40/4	10%-80%(2)
3	B9, B10	60/6	10%-90%(3)
4	B10, B11	70/7	10%-70%(4)
5	B11, A5	80/8	10%-80%(5)
6	A5, A6	50/5	10%-90%(6)
7	A6, A0	60/6	10%-70%(7)
8	A0, A1	40/4	10%-80%(8)
9	A1, A2	50/5	10%-90%(9)
10	A2, A3	60/6	10%-70%(0)
11	A3, A4	70/7	10%-80%(1)
12	A4, C13	80/8	10%-90%(2)

Вихідні дані ЛР 2

1. Структурна схема системи РК (за потребою розробляється функціональна схема). Структурна схема містить клавіатуру і відображає назву апаратних виходів портів мікроконтролера (МК) до яких підключена клавіатура. Також відображаються вихідні порти МК для формування сигналів.

2. Вихідний код і проект програми, що призначений для опитування клавіатури і формування сигналів Таблиці 2.

3. Зображення сигналів з осцилографу або скріншоти налагоджувача Keil, у початковому стані і після натискання кнопки, що підтверджують зміну ширини імпульсу ШІМ сигналу і працездатність вихідного коду. Значення сигналів мають в цілому відповідати значенням, наведеним у Таблиці 2. Висновки по розділам лабораторної роботи: розробка вихідного коду, підготовка проекту, налагодження і тестування.

Завдання

1. Розробити структурну схему (за потребою функціональну), що відображає схему підключення клавіатури до комплекту налагодження. На схемі також відобразити порти мікроконтролера, на яких формуються вихідні ШІМ сигнали.

2. Розробити програму для комплекту налагодження. Для даного рішення пропонується обрати архітектуру програми на основі переривань таймерів загального призначення мікроконтролерів або на базі операційної системи реального часу.

3. Переконайтеся, що розроблена програма дійсно видає сигнал відповідно до часових характеристик, що записані у таблиці 2 і ширина ШІМ сигналів керується через клавіатуру. Реалізувати це з використанням комплекту налагодження і або налагоджувача Keil uVision.

3.2. Теоретичний матеріал для лабораторної роботи №2

Важливість широтно-імпульсної модуляції визначається тим, що у сучасних системах радіокерування ШІМ сигнал є основою для управління сервомеханізмами різного роду. Сервомеханізми бувають різних розмірів, конструкцій і потужності. Важіль на валу сервоприводу (сервомеханізму) може відхилитися від центральної точки в обидва боки, і це залежить від ширини

сигналу управління, що подається на керуючий вхід сервоприводу. Найбільш часто зустрічається конструкція сервоприводу, яка дає можливість відхилити вал на 60 градусів. Кут відхилення можна запрограмувати з використанням мікроконтролера і він може відхилитися експоненціально, лінійно, асиметрично відносно сигналу управління. Існують сервоприводи з постійним обертанням валу, що призначені для моделей яхт. Крім сервоприводів ШІМ сигнал застосовується для керування електромоторами різної конструкції. Також існують електронні регулятори для управління яскравістю лампочок, світлодіодів, елементів нагрівання тощо.

Як варіант розглянемо приклад сигналів ШІМ для управління сервоприводом, що показаний на рис.3.

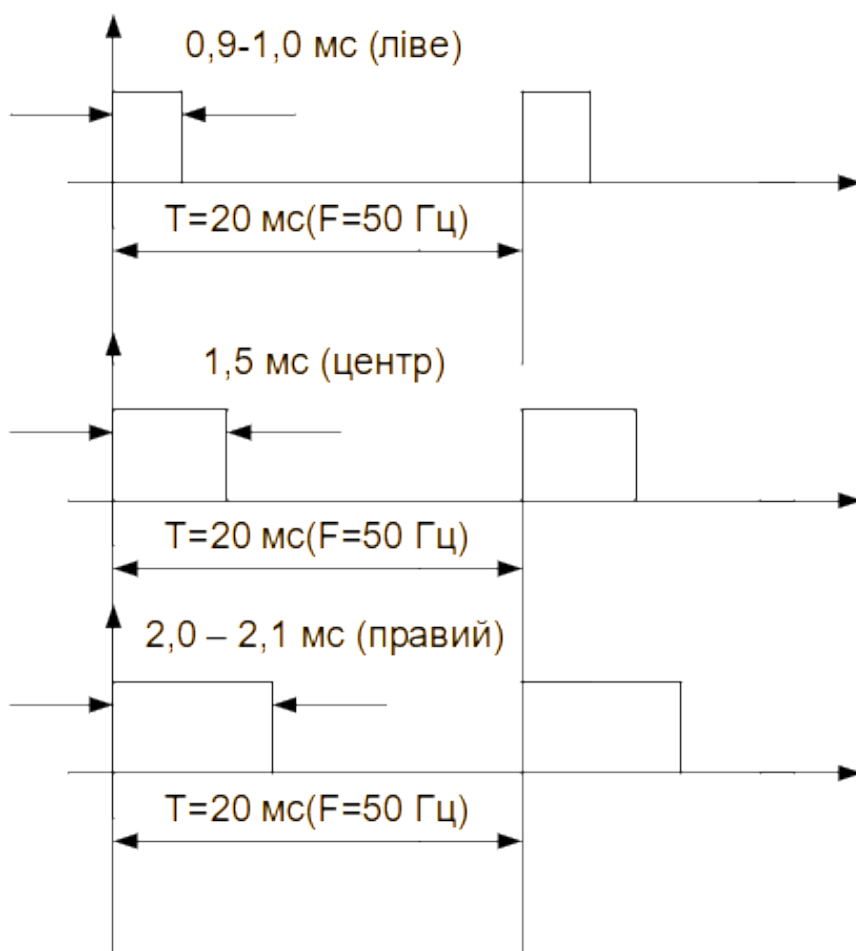


Рисунок 3 – Ширини сигналів для управління сервоприводом

На рисунку показані характеристики сигналів, частота слідування імпульсів та його ширина для повернення валу сервоприводу у відповідне положення. На рис.4 показана структурна схема найбільш простого пульта дистанційного управління на основі ШІМ сигналу для управління моделлю або технічною системою з використанням радіоканалу.

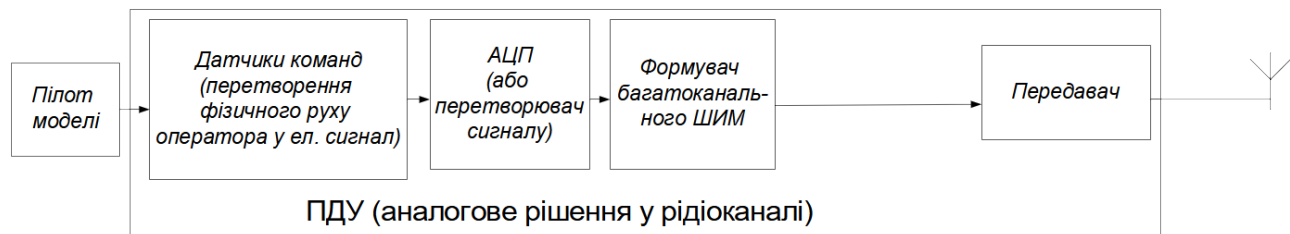


Рисунок 4 – Структурна схема пульта дистанційного управління на основі ШІМ

Як можна побачити з структурної схеми, що показана на рис.4 важливою частиною пульта дистанційного управління є формувач багатоканального ШІМ сигналу і датчик команд. У дискретному варіанті датчиком команд може бути звичайна клавіатура. Для вбудованих систем доволі часто використовують прості мембранні клавіатури. У Додатках показана принципова схема електронної складової мембранної клавіатури, що має 12 кнопок. Вихідний код для управління (драйвер) цей клавіатури показаний далі у прикладі коду. Програма складається з заголовного файлу ("keyboard.h") драйверу клавіатури, файлу реалізації функцій-драйверів ("keyboard.c") і тестового застосунку ("main.c"). Апаратні входи клавіатури позначаються символами: C1, C2, C3 та R1, R2, R3, R4. Кожний апаратний вивід клавіатури підключений до портів МК відповідно коментарів у файлі "main.c", як це показано у прикладі вихідного коду далі. Апаратні виходи клавіатури можуть підключатися до інших виводів мікроконтролера, що потребує переробленню вихідного коду драйверу. Вихідний код "main.c" наданий далі.

```

#include "keyboard.h"
//keyboard
//C3 GPIOB9
//C2 GPIOB10
//C1 GPIOB11
//R4 GPIOB12
//R3 GPIOB13
//R2 GPIOB14
//R1 GPIOB15

int state=0;

int main()
{
    keypad_init((unsigned int[3])
{GPIO_Pin_9, GPIO_Pin_10 ,GPIO_Pin_11},
(unsigned int[4])
{GPIO_Pin_12,GPIO_Pin_13,GPIO_Pin_14,GPIO_Pin_15});

while (1)
    {
        char c=keypadGetKey();
    }
}

```

У файлі вихідного коду потрібно прокоментувати особливості функції ініціалізації (драйверу) клавіатури `keypad_init (...)`. Функція має один параметр, що являє собою матрицю `keypad[numRow][numCol]`, яка має розміри 3x4, відповідно вихідному коду "keyboard.c". Таким чином, у вихідному коді `main.c` відразу відбувається ініціалізація рядків матриці у списку параметрів `keypad_init (...)`. Функція реалізації містить матрицю зі значенням клавіш для клавіатури і реалізацію двох функцій: функції налаштування і функції опитування клавіш.

```

#include "keyboard.h"

#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

char keypad[numRow][numCol]=
{
    '1','2','3',

```

```

    '4','5','6',
    '7','8','9',
    '*','0','#'
};

unsigned int row[numRow];
unsigned int col[numCol];
char oldKey=KEYPAD_NO_PRESSED;

void keypad_init(unsigned int cols[numCol],unsigned int rows[numRow]){
    unsigned int colSum=0;
    unsigned int rowSum=0;
    for(int i=0;i<numCol;i++)
    {
        col[i]=cols[i];
        colSum=colSum | col[i];
    }
    for(int i=0;i<numRow;i++)
    {
        row[i]=rows[i];
        rowSum=rowSum | row[i];
    }
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);// BUS is ON
    GPIO_InitTypeDef  KEYPAD_COL_B_STRUCT;
    //Init columns
    KEYPAD_COL_B_STRUCT.GPIO_Pin = colSum; // KEEP HARD PINS
    KEYPAD_COL_B_STRUCT.GPIO_Mode = GPIO_Mode_Out_OD;
    KEYPAD_COL_B_STRUCT.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &KEYPAD_COL_B_STRUCT);
    // Init rows
    GPIO_InitTypeDef KEYPAD_ROW_B_STRUCT;
    KEYPAD_ROW_B_STRUCT.GPIO_Pin = rowSum; // KEEP HARD PINS

    KEYPAD_ROW_B_STRUCT.GPIO_Mode = GPIO_Mode_IPU;
    KEYPAD_ROW_B_STRUCT.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOB, &KEYPAD_ROW_B_STRUCT);
}

char keypadGetKey(void)
{
    char res=KEYPAD_NO_PRESSED;
    for(int i=0;i<numCol;i++)
    {
        for(int k=0;k<numCol;k++)GPIO_SetBits(GPIOB, col[k]);
        //Set columns
        GPIO_ResetBits(GPIOB, col[i]);

        //Reset column which check now
    }
}

```

```

for(int k=0;k<numRow;k++)if(!GPIO_ReadInputDataBit(GPIOB, row[k]))res=keypad[k][i];
    //Check all rows
}
if(res != oldKey) //if key is difference with old key
{
    oldKey=res;
    return res;
}
return KEYPAD_NO_PRESSED;
}

```

Далі представлений вихідний код заголовного файлу драйверу клавіатури, що не потребує особливих пояснень і містить макровизначення і прототипи двох функцій – драйверів клавіатури.

```

#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

//=====
//
// Connect to keyboard as following:
//
// PORT B - 9 10 11 12 13 14 15
//           ^   ^   ^   ^
//
//=====

// GPIOB

#define KEYPAD_NO_PRESSED 0xFF
#define numRow 4
#define numCol 3

void keypad_init(unsigned int cols[numCol],unsigned int rows[numRow]);

char keypadGetKey(void);

```

Як показано на рис.4, крім управління клавіатурою потрібно сформувати ШІМ сигнал для передачі його на систему управління. Для передачі сигналу потрібно розробити програму, що дозволяє сформувати ШІМ сигнал і управляти ним з використанням сенсорів або клавіатури. Далі представлений вихідний код, що реалізує варіант налаштування і управління ШІМ сигналом у нескінченному циклі опитування. Сигнал формується з використанням таймеру

загального призначення, а управління відбувається з використанням основного потоку виконання програми.

```
#include "stm32f10x.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_tim.h"

#define PERIODL2 1000
int main(void)
{
    int TIM_PulseL2 = 0;
    int iL2;

    GPIO_InitTypeDef portL2;
    TIM_TimeBaseInitTypeDef timerL2;
    TIM_OCInitTypeDef timerPWML2;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);

    GPIO_StructInit(&portL2);
    portL2.GPIO_Mode = GPIO_Mode_IPU;
    portL2.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;
    portL2.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOA, &portL2);

    GPIO_StructInit(&portL2);
    port.L2GPIO_Mode = GPIO_Mode_AF_PP;
    port.L2GPIO_Pin = GPIO_Pin_6;
    port.L2GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &portL2);

    TIM_TimeBaseStructInit(&timerL2);
    timerL2.TIM_Prescaler = 720;
    timerL2.TIM_Period = PERIOD;
    timerL2.TIM_ClockDivision = 0;
    timerL2.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM4, &timerL2);

    TIM_OCStructInit(&timerPWML2);
    timerPWML2.TIM_OCMode = TIM_OCMode_PWM1;
    timerPWML2.TIM_OutputState = TIM_OutputState_Enable;
    timerPWML2.TIM_Pulse = 10;
    timerPWML2.TIM_OCPolarity = TIM_OCPolarity_High;
    TIM_OC1Init(TIM4, &timerPWML2);

    TIM_Cmd(TIM4, ENABLE);
```

```

while(1)
{
    if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 0) {
        if (TIM_PulseL2 < PERIODL2)
            TIM_PulseL2++;
        TIM4->CCR1 = TIM_PulseL2;
    }
    if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_1) == 0) {
        if (TIM_PulseL2 > 0)
            TIM_PulseL2--;
        TIM4->CCR1 = TIM_PulseL2;
    }
}
}
}

```

Вихідний код, що представлений нижче дозволяє сформувати ШІМ сигнал для передачі його на систему управління, що керується натисканням на кнопку. Від попередньої програми цей вихідний код відрізняє те, що налаштування таймеру загального призначення реалізовано як окремий драйвер. Сигнал також формується з використанням таймеру загального призначення, а управління відбувається з урахуванням частоти роботи мікроконтролера.

```

#include <stm32f10x.h>
#include <stm32f10x_rcc.h>
#include <stm32f10x_gpio.h>
#include <stm32f10x_tim.h>

#define PERIOD (uint16_t) (SystemCoreClock / 10000) //Amount of timer ticks for one period
#define FREQUENT 1 //Amount of periods in one second

int iL2 = 0;
uint16_t TIM_PulseL2;

void init(void);

int main(void)
{
    int directionL2 = 1; // increase, decrease
    init();
}

```

```

while (1) {

    //Button B6decrease/increase_pulse
    if (GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_6) == 0) {
        //if max/min change direction
        if (TIM_PulseL2 > PERIOD || TIM_PulseL2 < 0)
            directionL2 *= -1;
        TIM_PulseL2 += directionL2;
        TIM1->CCR3 = TIM_PulseL2;
    }
    // delay contact bounce
    for(iL2=0;iL2<0x1000;iL2++);
}

}

void init(void)
{

    GPIO_InitTypeDef port_;
    TIM_TimeBaseInitTypeDef timer_;
    TIM_OCInitTypeDef timerPWM_;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);

    GPIO_StructInit(&port_);
    port_.GPIO_Mode = GPIO_Mode_IPU;
    port_.GPIO_Pin = GPIO_Pin_6;
    port_.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &port_);

    GPIO_StructInit(&port_);
    port_.GPIO_Mode = GPIO_Mode_AF_PP;
    port_.GPIO_Pin = GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
    port_.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &port_);

    TIM_TimeBaseStructInit(&timer_);
    timer_.TIM_Prescaler = SystemCoreClock / PERIOD / FREQUENT;
    timer_.TIM_Period = PERIOD;
    TIM_TimeBaseInit(TIM1, &timer_);

    TIM_PulseL2 = (uint16_t) (PERIOD*0.85); //85% to get 15% on inverted pin
    TIM_OCStructInit(&timerPWM_);
    timerPWM_.TIM_OCMode = TIM_OCMode_PWM1;
    timerPWM_.TIM_OutputNState = TIM_OutputNState_Enable;
    timerPWM_.TIM_OutputState = TIM_OutputState_Enable;
    timerPWM_.TIM_Pulse = TIM_PulseL2;
}

```

```

timerPWM_.TIM_OCParity = TIM_OCParity_High;
timerPWM_.TIM_OCNPolarity = TIM_OCNPolarity_High;

TIM_OC1Init(TIM1, &timerPWM_);
TIM_OC2Init(TIM1, &timerPWM_);
TIM_OC3Init(TIM1, &timerPWM_);

TIM_Cmd(TIM1, ENABLE);
TIM_CtrlPWMOutputs(TIM1, ENABLE);

}

```

Далі представлена програма для управління безпосередньо сервоприводом з використанням двох кнопок, що підключені до порту А. Програма складається з двох драйверів, один з яких призначений для ініціалізації таймерів загального призначення, інший для ініціалізації кнопок управління, що працюють на введення інформації і управління приводом.

```

#include "stm32f10x.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_tim.h"

#define SYSCLK 72000000
#define PRESCALER 72

GPIO_InitTypeDef port;
TIM_TimeBaseInitTypeDef timer;
TIM_OCInitTypeDef timerPWM;

void servo_init(void);
void init_buttons(void);

int main(void)
{
    int TIM_Pulse;
    int i;

    servo_init();
    init_buttons();

    TIM_Pulse = timerPWM.TIM_Pulse;

    while(1)
    {
        if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0) == 0) {
            if (TIM_Pulse < 4000)

```

```

        {
            TIM_Pulse++;
            TIM4->CCR1 = TIM_Pulse;
            TIM4->CCR3 = TIM_Pulse;
        }
    }
    if (GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_1) == 0) {
        if (TIM_Pulse > 0)
            TIM_Pulse--;
            TIM4->CCR1 = TIM_Pulse;
            TIM4->CCR3 = TIM_Pulse;
        }
        // delay
        for(i=0;i<0x1000;i++);
    }
}

void servo_init(void) {
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
//init gpio
    GPIO_StructInit(&port);
    port.GPIO_Mode = GPIO_Mode_AF_PP;
    port.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_8;
    port.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOB, &port);
//init timer
    TIM_TimeBaseStructInit(&timer);
    timer.TIM_Prescaler = PRESCALER;
    timer.TIM_Period = SYSCLK / PRESCALER / 800;
    timer.TIM_ClockDivision = 0;
    timer.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM4, &timer);
//init pwm
    TIM_OCStructInit(&timerPWM);
    timerPWM.TIM_Pulse = 0;
    timerPWM.TIM_OCMode = TIM_OCMode_PWM1;
    timerPWM.TIM_OutputState = TIM_OutputState_Enable;
    timerPWM.TIM_OCPolarity = TIM_OCPolarity_High;
    TIM_OC1Init(TIM4, &timerPWM);
    TIM_OC3Init(TIM4, &timerPWM);

    TIM_Cmd(TIM4, ENABLE);
}

void init_buttons(void)
{
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
    GPIO_StructInit(&port);
    port.GPIO_Mode = GPIO_Mode_IPU;
    port.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;
    port.GPIO_Speed = GPIO_Speed_2MHz;
    GPIO_Init(GPIOA, &port);
}

```

Таким чином система радіокерування складається з передавача (пульта дистанційного управління) і контролера-приймача, що міститься на об'єкті управління. Для радіокерованих моделей у найбільш простому варіанті контролер містить приймач, що приймає радіосигнал з пульта управління на основі transmitter протоколу (TX). Далі контролер моделі обробляє сигнали і відправляє на пристрій, що керує процесом пілотування з використанням receiver протоколу (RX).

Найбільш поширені TX-протоколи, що перелічені далі: D8, D16, LR12 (Frsky); DSM, DSM2, DSMX (Spektrum); Flysky (Flysky); A-FHSS (Hitec); FASST (Futaba); Hi-Sky (Deviation).

У переліку RX-протоколів потрібно визначити універсальні протоколи, що є відкритими і активно використовуються для радіокерування, а саме:

Pulse Width Modulation (PWM (ШИМ)), протокол, відповідно якого кожний сервопривід керованої рухомої системи (моделі) управляється сигналом, що передається по окремому радіоканалу, імпульс аналоговий, від ширини імпульсу залежить кут відхилення валу сервоприводу або потужність двигуна;

Pulse Position Modulation (PPM) протокол, що має також інші назви (PPMSUM та CPPM), у цьому протоколі по одному радіоканалу відбувається передача декількох інформаційних каналів (стандартно до 8), відбувається часове ущільнення сигналів, сигнали передаються у аналоговій формі, недоліком може бути додаткова затримка сигналів, перевагою – велика економія апаратних елементів системи передачі радіокерування;

Pulse Code Modulation (PCM) протокол, що визначається як імпульсно-кодова модуляція, корисний сигнал кодується у цифрову форму, перевага полягає в можливості використання перешкодозахищеного коду, недоліком є висока вартість у реалізації.

3.3. Контрольні запитання до лабораторної роботи № 2

1. Чому ШІМ сигнал дуже часто використовується в сучасних системах радіокерування?
2. Навіщо потрібні прототипи функцій драйверів?
3. Які фізичні системи можуть керуватися з використанням ШІМ сигналу?
4. Як керується сервопривід з використанням ШІМ сигналу?
5. Наведіть приклад опису функції, що призначена для налаштування таймеру загального призначення для генерації керованого ШІМ сигналу.
6. Нарисуйте принципову схему мембранної клавіатури з 12 кнопками.
7. Як відбувається підключення мембранної клавіатури до портів мікроконтролеру?
8. Поясніть у загальних рисах роботу драйверу клавіатури.

4. ЛАБОРАТОРНА РОБОТА № 3

Базові паттерни (шаблони) і архітектура програмної складової систем радіокерування. Програмування мікросхем радіомодулів

Дана лабораторна робота містить завдання на дослідження архітектури програмного рішення для формування радіоканалу в системі радіоуправління. Також розглядаються практичні питання щодо програмування мікросхем радіомодулів.

Вхідні дані ЛР 3

Для підготовки ЛР № 3 потрібно використовувати сучасні ПК, операційні системи і інтегровані середовища розробки, аналогічно ЛР1, 2.

В якості базової мікросхеми пропонується використовувати nRF24L01, що працює на частоті 2.4ГГц і керується через інтерфейс SPI.

В якості керуючого пристрою мікросхеми пропонується використовувати мікроконтролер STM32F103C8 або міні-комп'ютер, наприклад Orange PI. При наявності іншого комплекту налагодження мікроконтролера або іншої мікросхеми радіомодулю, дозволяється застосовувати їх для побудови радіоканалу по узгодженню з викладачем.

Програму потрібно реалізувати у варіанті КА, відповідно до завдання, що вказані у Таблиці 3. Дозволяється варіант реалізації програми з використанням операційних систем реального часу.

Дається можливість програмування з використанням різних бібліотек і архітектури, що вивчалися на протязі попередніх семестрів.

Таблиця 3. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Номер порту SPI для підключення радіомодуля	Вихідне значення сигналів. Символи відповідно таблиці ASCII	Кількість станів у патерні КА (або кількість потоків у ОСРЧ)
1	1	A, b, c	3
2	2	D, e, f	4
3	1	G, I, e	2
4	2	S, p, i	3
5	1	W, q, t	4
6	1	U, I, o	5
7	2	A, s, d	2
8	1	J, l, x	3
9	2	V, b, n	4
10	1	G, I, e	5
11	2	S, p, i	2
12	1	W, q, t	3

Вихідні дані ЛР 3

1. Структурна схема системи РК (за потребою розробляється функціональна схема) з урахуванням підключення радіомодулю nRF24L01. Відображаються вихідні порти МК для управління радіомодулем.

2. Вихідний код і проект програми, що реалізує управління nRF24L01 для передавання серії сигналів, показаних у Таблиці 3. Вихідний код має містити кількість станів у кінцевому автоматі, як показано у Таблиці 3.

3. Підтвердження факту передавання сигналу і приймання його на приймаче. Можна обрати двійковий спосіб відображення сигналів або відповідні індикаторні пристрої для відображення.

Завдання

1. Розробити структурну схему (за потребою функціональну), що відображає схему підключення радіомодулю до комплекту налагодження. Реалізувати схему приймача і передавача.

2. Розробити програму для комплекту налагодження приймача і передавача. Протестувати програму.

3. Переконатися, що розроблена програма дійсно організує радіоканал і передає послідовність, що записана у таблиці 3 без викривлень. Показати, що програма реалізована у варіанті кінцевого автомату і має кількість станів, що показані у Таблиці 3. Реалізувати це з використанням комплекту налагодження або налагоджувача Keil uVision.

4.2. Теоретичний матеріал для лабораторної роботи №3

Одним з варіантів реалізації системи дистанційного управління є розробка програмно керованого радіоканалу на частоті 2.4ГГц, що не потребує ліцензії. Радіомодуль nRF24L01, що працює на цій частоті, керується через інтерфейс SPI з використанням мікроконтролера або міні-комп'ютера, наприклад Orange PI. Габаритні розміри і розташування виводів показано на рис.5.

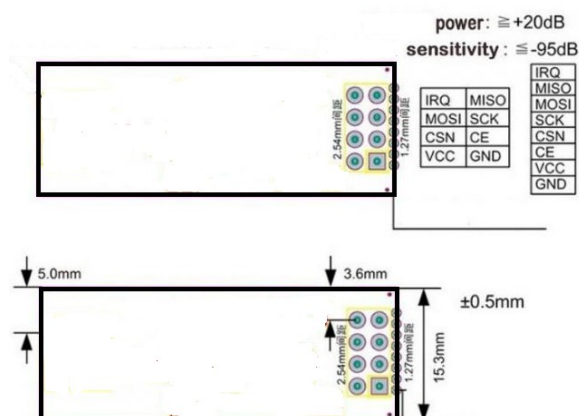


Рисунок 5 – Призначення виводів керованого радіомодуля і його габарити

Призначення виводів керованого радіомодуля наступне:

Serial Clock (SCK) – синхронізація;

Master Out Slave In (MOSI / MI) – вхід даних;

Master In Slave Out (MISO / MO) – вихід даних;

CE/SS – вибір підлеглого пристрою, якщо на шині SPI декілька пристроїв;

SCN – вибір режиму прийому/передача;

IRQ – сигнал переривання реакції мікроконтролера на рівні системи переривань;

GND – мінусова шина;

Vcc – вхід напруги живлення (+3,3В).

Паралельно живленню потрібно підключити електролітичний конденсатор до 100мкФ, і керамічний на 0,1 мкФ. У радімережу може бути об'єднано максимум до 6 радіомодулів. Максимальна довжина пакету дорівнює 32 байта.

Технічні характеристики радіомодуля nRF24L01:

– швидкість передачі до 2 Мбіт/сек;

– робоча напруга 2.7 - 3.6В;

– діапазон сигналів управління до 5В;

– діапазон робочих частот 2400 - 2524 МГц;

– модуляція Gaussian Minimum Shift Keying (GMSK);

– чутливість приймача: -95 дцБм;

– робочий діапазон температур, індустріальний стандарт -45...+85 град.

Структурна схема мікросхеми радіомодуля показана на рис.6.

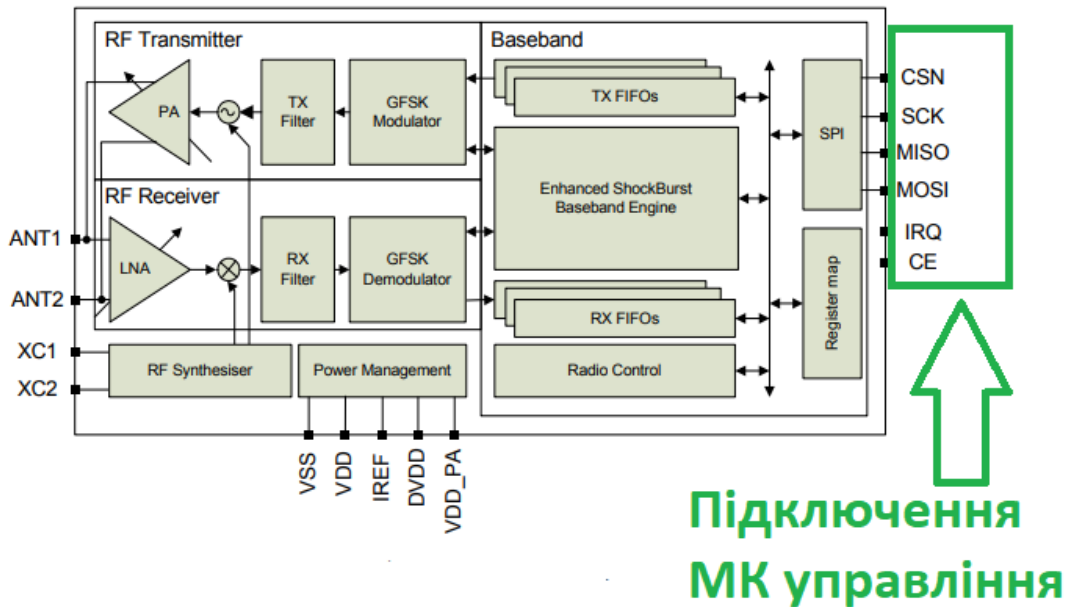


Рисунок 6 – Структурна схема мікросхеми радіомодуля nRF24L01

Багато програм для управління зовнішніми системами, у тому числі і для реалізації систем радіокерування, можна побудувати з використанням шаблону кінцевого автомату КА. Теорія кінцевого автомату у контексті використання для прикладного програмування систем РК показана в [5, 6 д]. Програма на основі КА може бути реалізована у процедурному варіанті. Для демонстрації прикладу КА доцільно розглянути найбільш простий вихідний код, що містить два стана. Інформація про стан зберігається у еnumerаторі, як показано у вихідному коді далі.

```
enum states {
    GPIO_PORT_ON,
    GPIO_PORT_OFF
};

enum states state = GPIO_PORT_OFF;

while(1) {
    if(state == GPIO_PORT_OFF) {
        led_on();
        state = GPIO_PORT_ON;
    } else {
```

```
led_off();
state = GPIO_PORT_OFF;
}
sleep(1); // sleep for a second
}
```

Більш складні варіанти вихідного коду з використанням КА, у тому числі у ООП варіанті, докладно розкриті у [1, 5д, 6д].

4.3. Контрольні запитання до лабораторної роботи № 3

1. Чому частота 2,4 МГц дуже часто використовується в сучасних системах радіокерування?

2. Навіщо потрібні реалізації програмної архітектури у вигляді кінцевого автомату?

3. Навіщо потрібні реалізації програмної архітектури у вигляді багатопоточності?

4. Як керується радіомодуль через послідовні інтерфейси?

5. Наведіть приклад структурної схеми для забезпечення радіоканалу, до використовує мікросхеми радіомодуля.

5. ЛАБОРАТОРНА РОБОТА № 4

Управління силовими елементами прототипів колісних роботів у режимі реального часу системи радіокерування

Проведення первинних досліджень для розробки програмної і апаратної частини радіокерованих колісних роботів (РККР) на основі дистанційного управління.

5.1. Основні завдання лабораторної роботи №4

Вхідні дані ЛР 4

Для виконання лабораторної роботи № 4 потрібно використовувати персональні комп'ютери, операційні системи і інтегровані середовища розробки, аналогічно ЛР1-3. В якості базової мікросхеми пропонується використовувати nRF24L01, що працює на частоті 2.4ГГц і керується через інтерфейс SPI.

Результати лабораторних робіт №2, 3 є основою для розробки структурної схеми апаратної частини і прототипу програмної частини.

В якості керуючого пристрою мікросхеми пропонується використовувати мікроконтролер STM32F103C8 або міні-комп'ютер. При наявності іншого комплекту налагодження мікроконтролера або іншої мікросхеми радіомодулю, дозволяється застосовувати їх для побудови радіоканалу, по узгодженню з викладачем.

Відповідно до Таблиці 4 потрібно виконати завдання лабораторної роботи.

Таблиця 4. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Тип управління	Кількість апаратних радіоканалів	Діапазон управління швидкістю
1	Зміна частоти обертання коліс лівого і правого боку	1	0-90%
2	Рульові колеса	2	0-80%
3	Зміна частоти обертання коліс лівого і правого боку	2	0-70%
4	Рульові колеса	1	0-90%
5	Зміна частоти обертання коліс лівого і правого боку	1	0-80%
6	Рульові колеса	2	0-70%
7	Зміна частоти обертання коліс лівого і правого боку	1	0-90%
8	Рульові колеса	2	0-80%
9	Зміна частоти обертання коліс лівого і правого боку	1	0-70%
10	Рульові колеса	2	0-50%
11	Зміна частоти обертання коліс лівого і правого боку	1	0-40%
12	Рульові колеса	2	0-80%

Вихідні дані ЛР 4

1. Структурна і функціональна схема апаратної частини РККР з урахуванням позначення портів вводу-виводу загального призначення.

2. Прототип програми для системи управління РККР.

Завдання

1. Розробити структурну схему (за потребою функціональну), що відображає апаратну частину системи управління РККР.

2. Розробити програму для обраного комплекту налагодження приймача і передавача РККР, з урахуванням вимог Таблиці 4. Передбачити потрібну кількість сигналів управління та їх склад.

3. Переконатися, що розроблена програма дійсно забезпечує управління РККР з урахуванням умов Таблиці 4. Реалізувати це з використанням апаратного комплекту налагодження або налагоджувача Keil uVision.

5.2 Теоретичний матеріал для лабораторної роботи №4

Радіокеровані колісні роботи і безпілотні наземні рухомі платформи активно використовуються у повсякденній діяльності. Вони призначені для спрощення процесу ліквідації наслідків природних чи техногенних катастроф. Також широко використовуються для зменшення рівня небезпеки для людини при проведенні робіт у небезпечних місцях і з небезпечними об'єктами. Прикладом може бути моніторинг території, небезпечної місцевості, що забруднена отруйними чи шкідливими речовинами.

Крім того на основі дистанційно керованих платформ розробляються роботи для огляду небезпечних місць у промисловості. Сферою застосування РККР є проведення моніторингу території під час ліквідації наслідків стихійних лих, робота у місцевостях де відбувається ліквідація наслідків техногенних катастроф або проведення огляду інших небезпечних для людини місць.

Прикладом РККР є автономний пристрій фірми Starship Technologies, що призначений для доставки маленьких вантажів по шляхам колективного користування. Платформа має шість коліс. Управління відбувається зміною швидкості трійки коліс лівого чи правого боку. Технічні характеристики платформи наступні:

- радіус дії до 3 кілометрів;
- час доставки 15-30 хвилин,
- вага пристрою близько 15 кг,
- корисний вантаж 7-8 кг;
- швидкість руху може досягати 6,5 км/г.

Пристрій працює на акумуляторних батареях Li-Ion, здатний переміщуватися автономно або автоматизовано. При переміщенні пристрою, на вимогу закону штату, за кожною платформою зобов'язаний спостерігати людина-оператор.

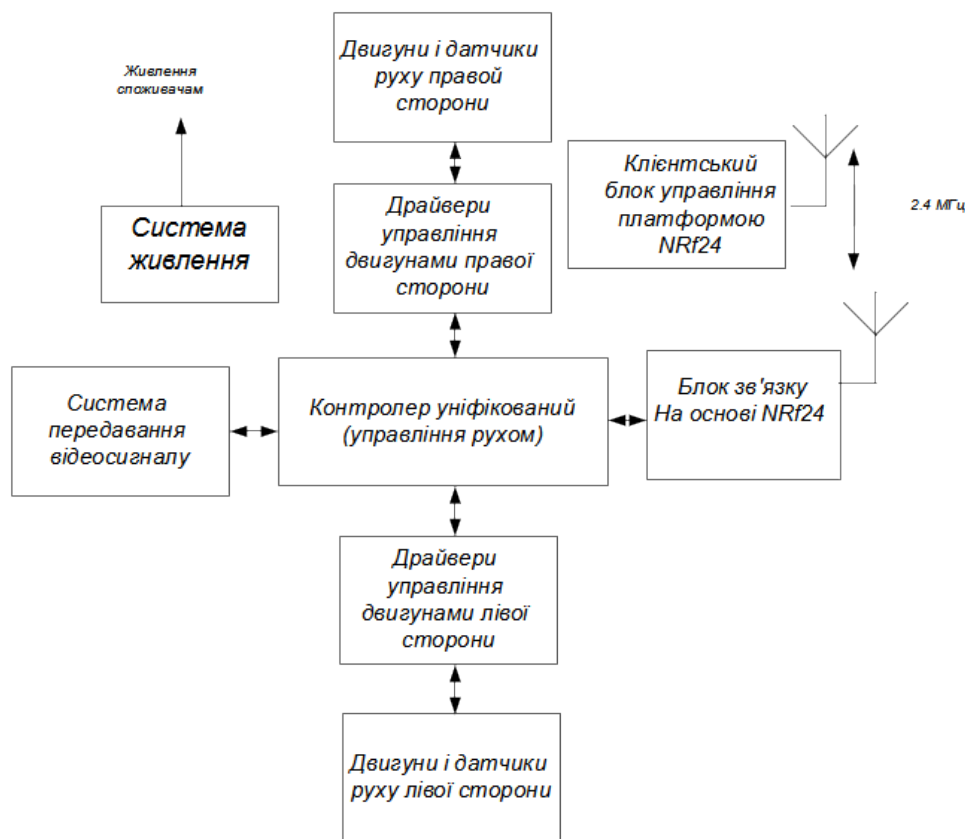


Рисунок 7 – Структурна схема управління рухомою платформою з використанням різниці швидкості обертання коліс лівого і правого боків

Другим варіантом прототипу РККР є апарат SpyRobot 4WD, що має 4 колеса особливої форми для кращого пересування по поверхні землі. Крім того вони дають можливість пересування SpyRobot 4WD на воді. Пружні "лопатки" цих коліс виконують роль своєрідної підвіски для амортизації нерівностей на високій швидкості.

Технічні характеристики апарату SpyRobot 4WD наступні:

- вага 5 кілограмів;
- швидкість до 10 кілометрів на годину;
- можливість подолання 45-градусного підйому;
- далькість керування до 400 метрів.

Платформа не має явно позначеного верху і низу і може рухатися у будь-якому положенні. Особливість рухомої платформи такого типу є те, що вона може керуватися зміною швидкості коліс лівого чи правого боку. Існує багато аналогічних платформ. Таким чином для узагальнення цього типу РККР, можна реалізувати структурну схему, що показана на рис. 7. Особливістю структурної схеми є те, що вона відображає систему управління двигунами лівого і правого боку для здійснення поворотів на місцевості. Для цього потрібно мінімум два окремих каналу управління. Потрібні також окремі апаратні частини драйверів двигунів і окремі двигуни лівого і правого боку, що також мають керуватися за швидкістю автономно.

Менш розповсюджені варіанти управління РККР можуть управлятися за рахунок повороту пари коліс відносно центральної вісі платформи. Колеса знаходяться на одній вісі. Аналогом такого рішення є РККР, що має наступні технічні характеристики для базової комплектації:

- розміри: (ш * д * в): 200 * 360 * 140 мм;
- швидкість руху: 0-11 м / с або 0-40 км / год;
- напруга живлення: 7,2 В;
- робоча частота радіоканалу : 50 або 450 МГц;
- час роботи акумуляторів: 2 год;

– стійкість до вертикальних прискорень: не менше 15 g.

В цьому варіанті, структурну схему рис.7, можна переробити на таку, що реалізує даний варіант РК і показана далі на рис.8.

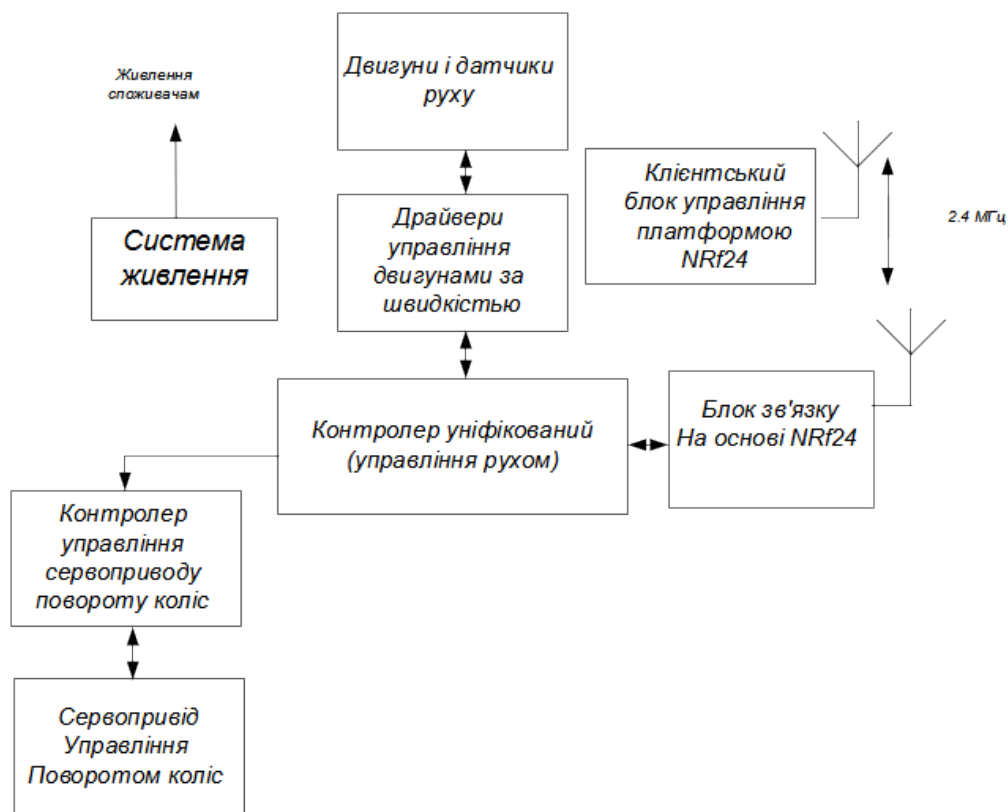


Рисунок 8 – Структурна схема управління рухомою платформою з використанням рульових коліс

Особливістю структурної схеми є те, що вона має окремі канали управління для повертання рульових коліс і для управління двигунами (двигуном), що керує швидкістю РККР. В обох каналах використовується сигнал ШІМ. Схема і принципи управління сервоприводом показані у лабораторній роботі 2.

5.3. Контрольні запитання до лабораторної роботи № 4

1. Поясніть призначення основних елементів структурної схеми РККР на основі зміни частоти обертання коліс лівого і правого боку для управління напрямом руху.
2. Поясніть призначення основних елементів структурної схеми РККР на основі обертання рульових коліс (колеса) відносно центральної вісі платформи.
3. Назвіть архітектурні варіанти програмного рішення для РККР.
4. Які варіанти протоколів передавання (прийому) ШІМ сигналів управління доцільно використовувати для реалізації РККР?
5. Як вирішувати проблему, коли кількість апаратних каналів менш ніж кількість сигналів управління?

6. ЛАБОРАТОРНА РОБОТА № 5

Використання АЦП і ЦАП для реалізації функціонального вузла для програмування джойстиків систем радіокерування

На відміну від попередніх лабораторних робіт, де управління відбувалося у дискретному режимі, у даній лабораторній роботі провадяться дослідження для управління у плавному варіанті з використанням джойстика.

Метою лабораторної роботи є проведення досліджень для розробки драйверу джойстику і інтеграції джойстика у загальну систему управління повітряною моделлю.

6.1 Основні завдання лабораторної роботи №5

Вхідні дані ЛР 5

В якості базової мікросхеми пропонується використовувати STM32F103C8 або міні-комп'ютер, наприклад Orange PI. При наявності іншого комплекту налагодження мікроконтролера або іншої мікросхеми радіомодулю, дозволяється застосовувати їх для побудови радіоканалу по узгодженню з викладачем.

Програму потрібно реалізувати у варіанті КА, відповідно до завдання, що вказані у Таблиці 5. Дозволяється варіант реалізації програми з використанням операційних систем реального часу.

Дається можливість програмування з використанням різних бібліотек і архітектури, що вивчалися на протязі попередніх семестрів.

В якості моделі датчика джойстика потрібно застосувати змінний резистор. Значення опору резистора вказані у таб.5

Таблиця 5. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Значення опору резистора для моделі датчика джойстика
1	10 КОм
2	1 КОм
3	10 КОм
4	1 КОм
5	10 КОм
6	1 КОм
7	10 КОм
8	1 КОм
9	10 КОм
10	1 КОм
11	10 КОм
12	1 КОм

Вихідні дані ЛР 3

1. Структурна схема пульта управління з джойстиком.
2. Вихідний код і проект програми, що реалізує управління однією лінією управління джойстику.
3. Підтвердження працездатності каналу АЦП на налагоджувачі або на реальному обладнанні.

Завдання

1. Розробити структурну схему (за потребою функціональну), що відображає схему підключення джойстика у загальній системі пульта управління з урахуванням каналу X і каналу Y.
2. Розробити програму для АЦП одного з каналів.

3. Переконалися, що розроблена програма дійсно видає сигнал, відповідно зміни положення джойстика.

6.2 Теоретичний матеріал для лабораторної роботи №5

Типовий пульт дистанційного управління моделлю літака або планера показаний на рис.9. Існує багато аналогічних пультів дистанційного управління, що виготовляється і використовується для управління повітряними моделями літаків або для планерів. В результаті аналізу пультів дистанційного управління такого типу можна визначити узагальнені їх узагальнені характеристики:



Рисунок 9 – Зовнішній вигляд типового пульта дистанційного управління для моделі літака або планера

- кількість каналів управління елементами моделі літака 5- 9;
- наявність польотних програм різного типу (Літак, Планер, Вертоліт, Літаюче крило);
- частотний діапазон: ISM 2,4 ГГц;
- різні протоколи передавання даних;
- підтримка протоколів PPM PS2;
- наявність дисплею;
- вага пульту до 1500 г, в залежності від виконання і реалізації;

вага приймача до 50 г.;

радіус дії: 500-2500 м.

Потрібно відмітити, що для управління моделями рухомих об'єктів, в багатьох випадках використовуються пульти дистанційного управління (ПДУ) з наявністю 1-2 джойстиків, що дозволяють керувати певною величиною повільно, плавно. Найбільш простий джойстик має 5 виводів, приклад якого показаний у Додатках, а принципова схема приведена на рис. 10. Даний джойстик дозволяє плавно управляти 2 параметрами і має одну кнопку для включення або виключення певного механізму. У інших джойстиків може бути значно багатий функціонал. Як можна побачити з рис. 10, основою джойстика є змінний резистор, що механічно з'єднаний з ручкою управління. Значення резисторів дорівнює 10 Ком. Один резистор забезпечує аналоговий сигнал X (для переміщення по осі X ручки джойстика). Інший резистор забезпечує аналоговий Y (для переміщення по осі Y ручки джойстика). На джойстику також є кнопка (Button). Отже, на виходах X і Y значення напруги повільно змінюється в залежності від нахилу ручки. Для мікроконтролера на входи джойстика подається напруга живлення 3,3 В, у вихідному стані на виводах X і Y буде половина напруги живлення. Звичайно потрібно враховувати допуски резисторів. В більшості випадків потрібна програмна корекція апаратної частини джойстика.

Для включення джойстика у систему управління потрібно використовувати аналого-цифровий перетворювач (АЦП), який є наявними на борту мікросхеми мікроконтролерів.

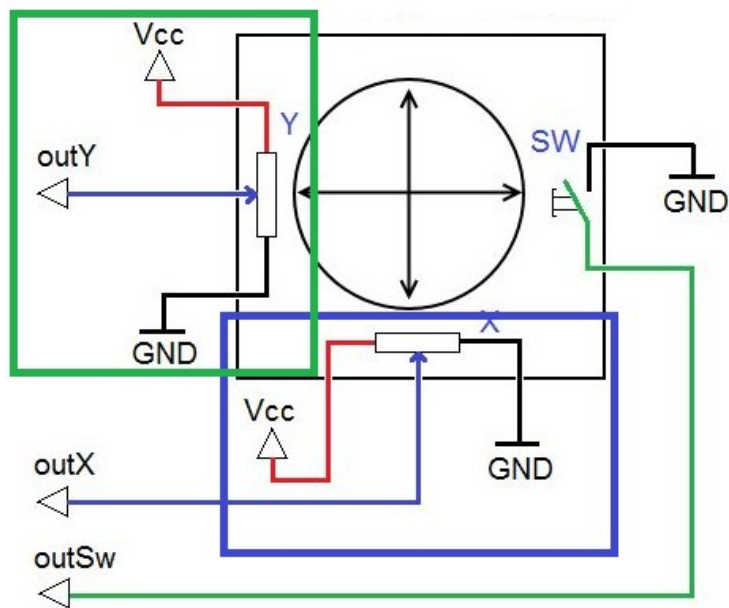


Рисунок 10 – Ескіз принципової схеми простого джойстика

Розкриємо основні поняття, що стосуються АЦП. Аналого-цифрові перетворювачі є пристрої, що приймають вхідні аналогові сигнали і генерують відповідні їм цифрові коди. Вони призначені для перетворення аналогових сигналів в МК та узгодження з іншими цифровими пристроями. Англійський термін, що відповідає АЦП - Analog-to-Digital Converter. Математична формалізація аналого-цифрового перетворення, є перетворення безперервної функції часу $X(t)$, у послідовність $\{X'(t_j)\}$, $j=0,1,2,\dots$, де t_j визначені фіксовані моменти часу. В більшості випадків вони є рівними. Умовно аналого-цифрове перетворення сигналів можна розділити на дві самостійні операції.

Перша називається дискретизацією і полягає у перетворенні безперервного сигналу $X(t)$ до дискретного аналогового сигналу. Це можна описати формальною моделлю у вигляді послідовності $\{X(t_j)\}$.

Друга операція називається квантуванням. Вона завершує перетворення елементів $\{X(t_j)\}$, до множини $\{X'(t_j)\}$. Елементами цієї множини є значення аналогового сигналу у вигляді цифрового коду. Отже на вході АЦП маємо аналоговий сигнал $X(t)$, на виході АЦП - цифровий код, що відповідає вхідному сигналу $\{X'(t_j)\}$.

Мікроконтролери STM32 [1] мають вбудований 12-разрядний АЦП з тактовою частотою максимум 14 МГц. Розрядність АЦП це характеристика, що визначає його точність. Крім того важливою характеристикою АЦП є діапазон опорних напруг. Діапазон опорних напруг визначає максимальний і мінімальний сигнал, що може перетворювати АЦП.

Розглянемо вищезгадані характеристики для абстрактного АЦП у мікроконтролері серії STM32. Типовий приклад це 12-разрядний АЦП з діапазоном опорних напруг від 0 до 3,3 В може забезпечувати перетворення сигналу з точністю до 8.05×10^{-4} В. Розглянемо більш докладно характеристики АЦП STM32. Вони визначаються наступними значеннями:

розрядність – 12 біт;

час перетворення мінімальний – 1 мкс;

кількість каналів – 16 зовнішніх;

калібрування автоматичне;

наявність віконного компаратору;

можливість запуску від зовнішнього джерела і робота через ПДП.

Доступні наступні режими перетворення: одноразове, безперервне, по тригеру, за таймером. Перетворювачі АЦП мікроконтролерів серії STM32 мають два класи: з регулярними каналами (regular) і з інжектованими каналами (injected). Звичайно перших налічується 18, два з яких вже задіяні для власних потреб мікроконтролера. Один канал для опорної напруги, другий канал для температурного датчика. Вимірювання регулярних каналів зберігаються у загальному для всіх каналів регістрі. Отже вони потребують збереження результатів у пам'яті мікроконтролера. Чотири інжектованих канала мають окремі регістри для зберігання результату.

При незалежній роботі АЦП мають працювати в наступних режимах:

-одноканальний (Single-channel) АЦП виконує перетворення одного каналу, записує у вихідний регістр, зупиняється;

- одноканальний тривалий (Single continuous) те саме, що попередній, тільки результат записується циклічно;

- багатоканальний Scan, декілька каналів виконують перетворення у заданій послідовності, однократне перетворення, після закінчення все зупиняється;

- багатоканальний тривалий (Scan continuous), те саме, що попередній, тільки виконується циклічно, результати записуються до одного окремого регістра і перезаписуються після кожного перетворення каналу;

- переривистий (Discontinuous), те саме, що попередні, тільки скановані не всі канали за раз, а ті що налаштовані програмно.

Звичайно АЦП має зовнішнє живлення від 2,4 В до 3,6 В. Розглянемо приклад роботи АЦП, що перетворює сигнал і передає його до послідовного інтерфейсу [1, 8] у повному обсязі.

Далі представлений вихідний код для одноканального тривалого режиму роботи, вхід АЦП на А1, на вхід подається напруга живлення від змінного резистору 10 КОм, напруга на входах резистора 3,3 В. Інформація передається до послідовного порту USART1, що має наступні конфігурації:

- частота передачі 115200 baud;
- для передавання використовується 8 Bits;
- один Stop Bit;
- не перевіряється паритет;
- дуплексний варіант передачі.

```
#include "stm32f10x.h"  
#include "stm32f10x_gpio.h"  
#include "stm32f10x_rcc.h"  
#include "stm32f10x_usart.h"  
#include "stm32f10x_adc.h"  
#include "misc.h"
```

```
volatile char L5buffer[50] = {'\0'};
```

```
void L5usart_init(void);  
void L5USARTSend(const unsigned char *pucBuffer, unsigned long ulCount);  
void L5ADC_init(void);
```

```

int main(void)
{
    const unsigned char L5mytext[] = " Hello World!\r\n";
    int L5adc_value;

    //USART1
    L5usart_init();
    L5USART_SendData(USART1, '\r');
    L5USARTSend(L5mytext, sizeof(L5mytext));
    L5ADC_init();

    while (1)
    {
        L5adc_value = ADC_GetConversionValue(ADC1);
        sprintf(L5buffer, "%d\r\n", L5adc_value);
        USARTSend(L5buffer, sizeof(L5buffer));
    }
}
//-----
void L5usart_init(void)
{
    /* Enable USART1 and GPIOA clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1 | RCC_APB2Periph_GPIOA,
    ENABLE);

    NVIC_InitTypeDef L5NVIC_IStr;
    L5NVIC_IStr.NVIC_IRQChannel = USART1_IRQn;
    L5NVIC_IStr.NVIC_IRQChannelPreemptionPriority = 0;
    L5NVIC_IStr.NVIC_IRQChannelSubPriority = 0;
    L5NVIC_IStr.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&L5NVIC_IStr);

    /* Configure the GPIOs */

    GPIO_InitTypeDef L5GPIO_IStr;

    /* Configure USART1 Tx (PA.09) as alternate function push-pull */
    L5GPIO_IStr.GPIO_Pin = GPIO_Pin_9;
    L5GPIO_IStr.GPIO_Mode = GPIO_Mode_AF_PP;
    L5GPIO_IStr.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &L5GPIO_IStr);

    /* Configure USART1 Rx (PA.10) as input floating */
    L5GPIO_IStr.GPIO_Pin = GPIO_Pin_10;
    L5GPIO_IStr.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOA, &L5GPIO_IStr);
}

```

```

/* Configure the USART1 */
//USART_Configuration();
USART_InitTypeDef L5USART_ISt;

L5USART_ISt.USART_BaudRate = 115200;
L5USART_ISt.USART_WordLength = USART_WordLength_8b;
L5USART_ISt.USART_StopBits = USART_StopBits_1;
L5USART_ISt.USART_Parity = USART_Parity_No;
L5USART_ISt.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
L5USART_ISt.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;

USART_Init(USART1, &L5USART_ISt);

/* Enable USART1 */
USART_Cmd(USART1, ENABLE);

/* Enable the USART1 Receive interrupt: this interrupt is generated when the
   USART1 receive data register is not empty */

}

void L5USARTSend(const unsigned char *pucBuffer, unsigned long ulCount)
{
    //
    // Loop while there are more characters to send.
    //
    while(ulCount--)
    {
        USART_SendData(USART1, *pucBuffer++); // Last Version USART_SendData(USART1,
        (uint16_t) *pucBuffer++);
        /* Loop until the end of transmission */
        while(USART_GetFlagStatus(USART1, USART_FLAG_TC) == RESET)
        {
        }
    }
}

void L5ADC_init(void)
{
    //ADC
    ADC_InitTypeDef L5ADC_ISt;
    GPIO_InitTypeDef L5GPIO_ISt;
    // input of ADC (it doesn't seem to be needed, as default GPIO state is floating input)
    L5GPIO_ISt.GPIO_Mode = GPIO_Mode_AIN;
    L5GPIO_ISt.GPIO_Pin = GPIO_Pin_1 ; // that's ADC1 (PA1 on STM32)
    GPIO_Init(GPIOA, &L5GPIO_ISt);
}

```

```

//clock for ADC (max 14MHz --> 72/6=12MHz)
RCC_ADCCLKConfig(RCC_PCLK2_Div6);
// enable ADC system clock
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);

// define ADC config
L5ADC_ISt.ADC_Mode = ADC_Mode_Independent;
L5ADC_ISt.ADC_ScanConvMode = DISABLE;
L5ADC_ISt.ADC_ContinuousConvMode = ENABLE; // we work in continuous sampling mode
L5ADC_ISt.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
L5ADC_ISt.ADC_DataAlign = ADC_DataAlign_Right;
L5ADC_ISt.ADC_NbrOfChannel = 1;

ADC_RegularChannelConfig(ADC1,ADC_Channel_1, 1,ADC_SampleTime_28Cycles5); //
define regular conversion config
ADC_Init ( ADC1, &L5ADC_ISt); //set config of ADC1

// enable ADC
ADC_Cmd (ADC1,ENABLE); //enable ADC1

// ADC calibration (optional, but recommended at power on)
ADC_ResetCalibration(ADC1); // Reset previous calibration
while(ADC_GetResetCalibrationStatus(ADC1));
ADC_StartCalibration(ADC1); // Start new calibration (ADC must be off at that time)
while(ADC_GetCalibrationStatus(ADC1));

// start conversion
ADC_Cmd (ADC1,ENABLE); //enable ADC1
ADC_SoftwareStartConvCmd(ADC1, ENABLE); // start conversion (will be endless as we are
in continuous mode)
}

```

6.3. Контрольні запитання до лабораторної роботи № 5

1. Які узагальнені характеристики мають типові пульти дистанційного управління повітряних об'єктів?
2. Назвіть базові характеристики аналого-цифрового перетворювача STM32F1xx.
3. Намалуйте і поясніть принципову схему простого джойстика з кнопкою.
4. Для чого потрібні АЦП у будь якому типі МК?
5. Які функції забезпечує DMA контролер?
6. Як організована архітектура DMA у родини МК STM32F1xx?
7. У чому полягає процес дискретизації у АЦП?

8. У чому полягає процес квантування у АЦП?
9. Яка характеристика визначає швидкодію АЦП?
10. Навіщо підключати АЦП до джойстика, який сигнал на нього надходить (вказіть мінімальну і максимальну амплітуду), який сигнал з нього виходить?
11. Якими параметрами визначається точність і швидкодія АЦП?

7. ЛАБОРАТОРНА РОБОТА № 6

Управління силовими елементами прототипів у плавному режимі зміни потужності. Управління серво у повітряних моделях систем радіокерування

Мета лабораторної роботи полягає у проведенні досліджень для набуття знань, умінь та навичок з технології розроблення структурної схеми системи РК повітряних моделей літаків або планерів та програмного рішення для окремих елементів.

7.1. Основні завдання лабораторної роботи № 6

Вхідні дані ЛР 6

В якості основи системи РК моделі літака або планера пропонується використовувати мікроконтролер STM32F103C8 або міні-комп'ютер, наприклад Orange PI. При наявності іншого комплекту налагодження мікроконтролера або іншої мікросхеми радіомодулю, дозволяється застосовувати їх для побудови радіоканалу по узгодженню з викладачем.

Програмно потрібно реалізувати систему РК моделі літака або планера відповідно до завдання, що вказані у Таблиці 6.

В пульті дистанційного управління пропонується використовувати два джойстика.

Вибір моделі сервомашинки не обмежено.

Вибір камери огляду і технології передавання відео не обмежений.

Вибір двигунів не обмежений.

Частота роботи каналу і мікросхема радіомодулю управління за вибором студента.

Таблиця 6. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Тип моделі	Кількість каналів для управління обладнанням моделі
1	Модель літака	4
2	Модель планера	5
3	Модель літака	4
4	Модель планера	5
5	Модель літака	4
6	Модель планера	5
7	Модель літака	4
8	Модель планера	5
9	Модель літака	4
10	Модель планера	5
11	Модель літака	4
12	Модель планера	5

Вихідні дані ЛР 6

1. Структурна схема системи управління моделі літака (планера) разом з пультом управління з джойстиками і додатковими каналами.
2. Вихідний код і проект програми, що реалізує управління однією лінією управління джойстиком і елементом авіоніки моделі (сервомеханізмом).
3. Перелік додаткового обладнання для розробки моделі літака (планера).

Завдання

1. Розробити структурну схему системи РК моделі літака або планера, з урахуванням вимог Таблиці.6
2. Розробити програму що реалізує один канал управління елементом авіоніки моделі (наприклад елеронами) з використанням одного каналу джойстика.

3. Переконатися, що розроблена програма дійсно видає сигнал, відповідно зміни положення джойстику. Реалізувати це з використанням комплексу налагодження або налагоджувача Keil uVision.

7.2. Контрольні запитання до лабораторної роботи № 6

1. В чому відмінність структурної схеми моделі літака і планера?

2. В чому відмінність каналу управління двигуном моделі літака і каналів управління його авіонікою?

3. Скільки радіоканалів потрібно для управління моделлю літака і планера мінімум? Чому?

3. Намалуйте і поясніть структурну схему простого пульта дистанційного управління літака (планера).

4. Намалуйте і поясніть структурну схему простої моделі літака з радіокеруванням.

5. Намалуйте і поясніть структурну схему простої моделі планера з радіокеруванням.

6. Яку функцію виконує польотний контроллер моделі літака (планера)? Намалуйте його узагальнену структурну схему.

7. Які складові потрібні для системи дистанційного РК моделі літака (планера)?

8. ЛАБОРАТОРНА РОБОТА № 7

Налагодження системи радіокерування для повітряних моделей

Мета лабораторної роботи полягає у проведенні досліджень для набуття знань, умінь та навичок з технології розроблення структурної схеми, програми і загального процесу налагодження РК повітряних моделей на приладі квадрокоптера.

8.1. Основні завдання до лабораторної роботи № 7

Вхідні дані ЛР 7

В якості основи моделі РК для процесу налагодження пропонується використовувати модель системи радіокерування відповідно Таблиці 7.

Програмно потрібно реалізувати систему РК моделі квадрокоптера відповідно до завдання, що вказані у Таблиці 7.

В пульті дистанційного управління пропонується використовувати два джойстика.

Вибір моделі сервомашинки не обмежений.

Вибір камери огляду і технології передавання відео не обмежений.

У системі радіокерування передбачено використання електродвигунів (зазвичай по одному на кожен гвинт).

Можливо використання колекторного двигуна. У двигунах цього типу для перемикання струму між обмотками використовується колектор. Перевагами даної конструкції є простота і не висока вартість. Недоліком колекторних електричних двигунів є те, що вони мають невисокий коефіцієнт корисної дії і невисоку надійність.

У безколекторних двигунів перемикання струму між обмотками здійснюється за допомогою електронних схем, що значно підвищує коефіцієнт корисної дії і надійність конструкції. Недоліком є висока вартість двигуна.

Передбачити, що реалізація радіоканалу буде відбуватися на частоті 2.4 GHz або 5.8 GHz допускається підтримка два діапазону. Частота роботи каналу і мікросхема радіомодулю управління за вибором студента.

Передбачається, що коптер вже має польотний контролер, для якого потрібно забезпечити радіоканали для зв'язку.

Для квадрокоптера передбачено управління від спеціалізованого пульта дистанційного управління.

Потрібно враховувати, що радіус дії коптера буде визначатися для ідеальних умов, в межах прямої видимості, без перешкод на шляху сигналу і перешкод в ефірі.

Передбачити додаткові датчики, в конструкції квадрокоптера, а саме:

- датчик висоти, на основі барометричного або ультразвукового принципу вимірювання висоти.

- відеокамера, що дозволяє квадрокоптеру оглядати навколишню обстановку в певних напрямках;

- датчики для стабілізації польоту;

- датчики положення (наприклад GPS) і напрямку руху;

- датчики температури, тощо.

Недоліком барометричних датчиків висоти є те, що барометричні датчики не дуже точні, проте вони добре працюють на великому діапазоні зміни висот, від десятків до сотень метрів.

Ультразвукові датчики дозволяють дуже точно вимірювати малі висоти, під час набору висоти їх ефективність зменшується.

Передбачити використання відразу двох типів датчиків.

Таблиця 7. – Варіанти виконання лабораторної роботи і параметри сигналу

Номер варіанту	Тип моделі	Кількість каналів для управління обладнанням моделі
1	Трикоптер	4
2	Квадрокоптер	5
3	Трикоптер	4
4	Квадрокоптер	5
5	Трикоптер	4
6	Квадрокоптер	5
7	Трикоптер	4
8	Квадрокоптер	5
9	Трикоптер	4
10	Квадрокоптер	5
11	Трикоптер	4
12	Квадрокоптер	5

Вихідні дані ЛР 7

1. Структурна схема системи управління моделі коптера разом з пультом управління з джойстиками і додатковими каналами.
2. Вихідний код і проект програми, що реалізує лінією управління джойстиком і елементом коптера.
3. Перелік додаткового обладнання для розробки моделі коптера.
4. Апаратна модель одного каналу з двигуном (або сервоприводом) і джойстиком, з програмою передбаченою п.2, для налаштування. Даний пункт актуальний під час роботи у аудиторії.

Завдання

1. Розробити структурну схему системи РК коптера, з урахуванням вимог Таблиці.7

2. Розробити програму, що реалізує управління в одній лінії квадрокоптера з використанням одного каналу джойстика.

3. Переконайтеся, що розроблена програма дійсно видає сигнал, відповідно зміни положення джойстика. Реалізувати це апаратно.

4. Провести налаштування лінії управління.

8.2. Контрольні запитання до лабораторної роботи № 7

1. В чому відмінність структурної схеми моделі літака від квадрокоптера?

2. В чому особливість каналу управління двигуном квадрокоптера?

3. Скільки радіоканалів потрібно для управління моделлю коптера?

Чому?

3. Намалюйте і поясніть структурну схему простого пульта дистанційного управління коптера.

4. Намалюйте і поясніть структурну схему простої моделі квадрокоптера з радіокеруванням.

5. Яку роль виконує польотний контролер коптера?

6. Намалюйте узагальнену структурну схему РК коптера.

7. Які складові потрібні для системи дистанційного РК коптера?

9. ЛАБОРАТОРНА РОБОТА № 8

Основи технології програмування, розробки і налагодження системи радіокерування у варіанті клієнт-сервер

Провести дослідження для реалізації моделі літака (планера, коптера) на основі мінікомп'ютера.

9.1. Основні завдання до лабораторної роботи № 8

Вхідні дані ЛР 8

Таблиці попередніх лабораторних робіт 6 (або 7).

Вихідні дані, що перелічені у попередніх лабораторних роботах 6 (або 7).

Вихідні дані ЛР 8

1. Структурна схема системи управління моделі літака або коптера, разом з пультом управління з джойстиками і додатковими каналами, з урахуванням того, що базові елементи системи РК побудовані на основі мінікомп'ютера.

2. Вихідний код і проект програми, що реалізує управління однією лінією управління джойстиком і елементом , з урахуванням того, що базові елементи системи РК побудовані на основі мінікомп'ютера.

3. Перелік додаткового обладнання для розробки моделі літака (планера, коптера).

4. Апаратна модель одного каналу з двигуном (або сервоприводом) і джойстиком, з програмою передбаченою п.2, для налаштування.

Завдання

1. Розробити структурну схему системи РК коптера (літака, планера), з урахуванням вимог Таблиц 6, 7, з урахуванням того, що базові елементи системи РК побудовані на основі мінікомп'ютера.

2. Розробити програму, що реалізує управління в одній лінії квадрокоптера з використанням одного каналу джойстика, з урахуванням того, що базові елементи системи РК побудовані на основі мінікомп'ютера.

3. Провести тестування результатів.

9.2. Контрольні запитання до лабораторної роботи № 8

1. В чому відмінність структурної схеми моделі літака (планера, коптера) на основі мікроконтролера від рішень на основі мінікомп'ютера?
2. В чому відмінність каналу управління двигуном на основі мікроконтролера від рішень на основі мінікомп'ютера?
3. Намалюйте і поясніть структурну схему простого пульта дистанційного управління на базі мінікомп'ютера.
4. Намалюйте і поясніть структурну схему простої моделі літака (планера, коптера) на основі мінікомп'ютера?
5. Як може бути реалізований польотний контролер з використанням мінікомп'ютера?
6. Які програмні технології доступні для створення польотного контролеру з використанням мінікомп'ютера?
7. Які апаратні технології доступні для створення польотного контролеру з використанням мінікомп'ютера?

10. ОФОРМЛЕННЯ ЗВІТУ ТА ПОРЯДОК ЙОГО ПОДАННЯ

Звіт можна оформлювати у варіанті проекту з використання вимог до конструкторської документації або у довільній формі. Дозволяється реалізувати звіт в електронній формі або роздрукованому вигляді. Затримка подання роздрукованого звіту не повинна перевищувати дві лабораторні роботи. У випадку більшої затримки можуть нараховуватися штрафні бали. Під час особового стану штрафні бали не нараховуються. Звіт має містити наступні елементи:

- файл з *вихідним кодом*, що не містить синтаксичних помилок і помилок на етапі виконання програми;

- у вихідному коді коментарі з прізвищем, номером групи, номером лабораторної роботи;

- здатність студента зробити зміни у програмі і налагодити після цього програму;

Для підтвердження результатів до програми додаються фотографії екранів (скріншот).

У звіті має бути висновок, що містить інформацію про ускладнення, що виникли під час розробки і налагодження програми, відповідність результатів теоретичним положенням.

Далі запропонований варіант титульного аркушу.



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
НАЗВА ФАКУЛЬТЕТУ
НАЗВА КАФЕДРИ

ЛАБОРАТОРНА РОБОТА № 1

З ДИСЦИПЛІНИ «»

Виконав:

студент групи РТ – 00

« _ » _____

Перевірив:

« _ » _____

СПИСОК ЛІТЕРАТУРИ

Основна література

1. Програмування мікроконтролерних систем: методичні рекомендації до виконання лабораторних робіт [Електронний ресурс] : для студентів спеціальності 121 «Інженерія програмного забезпечення» кафедри Автоматики та управління в технічних системах / КПІ імені Ігоря Сікорського ; уклад.: П. Ю. Катін, Я. Ю. Дорогий. – Електронні текстові дані (1 файл: 1,72 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 127 с. – Назва з екрана. https://ela.kpi.ua/bitstream/123456789/49043/1/PMS_LABS
2. Інформатика. Основи програмування та алгоритми. Мова програмування С. Лабораторний практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітніми програмами «Інтелектуальні технології радіоелектронної техніки», «Інформаційна та комунікаційна радіоінженерія», «Радіотехнічні комп'ютеризовані системи», «Інформаційне забезпечення роботехнічних систем» спеціальності 172 Телекомунікації та радіотехніки 126 Інформаційні системи та технології / КПІ ім. Ігоря Сікорського ; уклад. С. В. Вишневий, П. Ю. Катін, Є. В. Крилов. – Електронні текстові дані (1 файл: 3,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 221 с. – Назва з екрана. <https://ela.kpi.ua/handle/123456789/48158>
3. Харченко В.П. Авіоніка: навч. посіб. / В.П. Харченко, І.В. Остроумов. – К. : НАУ, 2013. – 272 с.
4. Синєглазов В.М. Філяшкін М.К. С382 Автоматизовані системи управління повітряних суден: Підручник – К.: НАУ, 2013. – 502 с.
5. Системи реального часу [Електронний ресурс] : методичні вказівки до виконання лабораторних робіт для студентів напряму підготовки 6.050102 «Комп'ютерна інженерія» кафедри обчислювальної техніки всіх форм навчання / НТУУ «КПІ»; уклад. В. Є. Мухін, А. М. Волокита, Я. І. Корнага ; відп. ред. В. П. Широчин. – Електронні текстові дані (1 файл: 487 Кбайт). – Київ : НТУУ «КПІ», 2013. - 40 с.– Назва з екрана. <https://ela.kpi.ua/handle/123456789/4826>

Додаткова література

1. Бичковський В. О. Оцінка точності та ефективності систем радіокерування рухомими об'єктами / Бичковський В. О., Реутська Ю. Ю. // Вісник НТУУ «КПІ». Радіотехніка, радіоапаратобудування : збірник наукових праць. – 2015. – Вип. 62. – С. 62–69. – Бібліогр.: 12 назв. <https://ela.kpi.ua/handle/123456789/15306>
2. Jörg F. Wollert. Wolfram Gebhardt. The Wireless Book Evolution and Communication. steute Schaltgeräte GmbH & Co. KG, Löhne, Germany, Wireless business division. First edition, November 2010, 1000 wd. E-mail info@steute.com.
3. Nick Hunn. Introducing Bluetooth® LE Audio A guide to the latest Bluetooth specifications and how they will change the way we design and use audio and telephony products. 2022. www.nickhunn.com.
4. Owen Bishop. Remote Control handbook. Bernard Babani LTD/ 1988.
5. П.Ю. Катін, О.А. Похиленко. Шаблиони типу Стан для створення інфраструктури системного програмного забезпечення мікроконтролерів архітектури Cortex-M у режимі реального часу для вбудованих систем. Електронне моделювання. Том 43, № 2 (2021) <https://www.emodel.org.ua/uk/archive-ukr/2021/43-2-ц> . Мова матеріалу: Українська кількість сторінок: 17 (51 — 67).
6. О.А. Похиленко, П.Ю. Катін. ПАТЕРН «СТАН» ДЛЯ ВБУДОВАНИХ СИСТЕМ З МОЖЛИВІСТЮ ДИНАМІЧНОГО СТВОРЕННЯ СТАНІВ. ТЕХНІЧНІ НАУКИ ТА ТЕХНОЛОГІЇ TECHNICAL. SCIENCES AND TECHNOLOGIES. № 1(23), 2021. <http://tst.stu.cn.ua/article/view/233571>. Мова матеріалу: Українська, кількість сторінок: 10 (118 – 127).
7. С.І. АЛЬПЕРТ, М.І. АЛЬПЕРТ, П.Ю. КАТІН, Н.О. ЛІТВІНОВА. ПРОГРАМНО-АПАРАТНА ІНФРАСТРУКТУРА НАЗЕМНОЇ АВТОНОМНОЇ ПЛАТФОРМИ З ЕЛЕМЕНТАМИ ШТУЧНОГО ІНТЕЛЕКТУ Науковий Центр аерокосмічних досліджень Землі ІГН НАН України, м. Київ, Україна 8 (24 – 31) 2021 Мова матеріалу: Українська кількість сторінок: 8 (24 — 31) 01_21_Alpert.pdf (immsp.kiev.ua).
8. Geoffrey Brown [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://github.com/geoffreybrown/STM32-Template> (дата звернення 01.09.2018) – Назва з екрана.
9. Andrey Koryagin (Avislab) [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://github.com/avislab/STM32F103> (дата звернення 01.09.2018) – Назва з екрана.

10. AVISLAB. Програмування STM32F103. [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу http://www.avislab.com/blog/stm32-nvic_ua/ STM32F103 (дата звернення 01.09.2018) – Назва з екрана.

11. P Katin, V Chmelov, V Shemaev. Development of Typical ‘State’ Software Patterns for CortexM Microcontrollers in Real Time. Eastern-European Journal of Enterprise Technologies, 2020. <http://journals.uran.ua/eejet/article/view/2053771>. MIT 16.07 Real Time Operating Systems. Lecture (RTOS - 16.070 Lecture 27) [Електронний ресурс] //– режим доступу : <http://web.mit.edu/16.070/www/year2001/RTOS27.pdf>, вільний . (дата звернення 05.01.2020) Загл. з екрану.

12. MIT 16.07 Real Time Operating Systems. Lecture (RTOS - 16.070 Lecture 28) [Електронний ресурс] //– режим доступу : <http://web.mit.edu/16.070/www/year2001/RTOS28.pdf>, вільний . (дата звернення 05.01.2020) Загл. з екрану.

13. Geoffrey Brown. Discovering the STM32 Microcontroller ©2012 June 5, 2015. [Електронний ресурс] //– режим доступу : <https://legacy.cs.indiana.edu/~geobrown/book.pdf>, вільний . – (дата звернення 05.01.2020) Загл. з екрану.

14. Geoffrey Brown STM32 tutorial implementation [Електронний ресурс] //– режим доступу : https://github.com/nalepae/stm32_tutorial, вільний . (дата звернення 2020) – Назва з екрана.

15. CMSIS-RTOS [Електронний ресурс]//–режим доступу: <https://os.mbed.com/handbook/CMSIS-RTOS> (дата звернення 08.01.2020) – Назва з екрана.

16. General-purpose timer cookbook AN4776 Application note [Електронний ресурс]// – режим доступу : www.st.com/resource/en/application_note/dm00236305.pdf, вільний . - Загл. з екрану. (20.02.2018).

17. RM0008 Reference manual for STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM®-based 32-bit MCUs [Електронний ресурс] // Keil, 2018. – режим доступу : www.keil.com/dd/docs/datashts/st/stm32f10xxx.pdf, вільний . - Загл. з екрану. (20.02.2018). [1]. 1133 с

18. STMicroelectronics. Reference manual. STM32F101xx, STM32F102xx, STM32F103xx, advanced Arm®-based 32-bit MCUs, March 2017. RM0008

19. STMicroelectronics. Programming manual: Stm32f10xxx/ 20xxx/ 21xxx/ 11xxxx cortex-m3 programming manual, March 2011. PM0056.

20. Geoffrey Brown [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://github.com/geoffreymbrown/STM32-Template> (дата звернення 01.09.2018) – Назва з екрана.
21. Andrey Koryagin (Avislab) [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://github.com/avislab/STM32F103> (дата звернення 01.09.2018) – Назва з екрана.
22. Yohanes Erwin. stm32f103-sw4stm32 [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://github.com/yohanes-erwin/stm32f103-sw4stm32> (дата звернення 01.09.2018) – Назва з екрана.
23. CMSIS [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу <https://developer.arm.com/tools-and-software/embedded/cmsis>(дата звернення 2019) – Назва з екрана.
24. Timer [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу https://i.ytimg.com/vi/aLCUDv_fgoU/maxresdefault.jpg(дата звернення 2019) – Назва з екрана.
25. Simple and Efficient Task Scheduler stm32f10x [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу GitHub: https://github.com/atakansarioglu/ME_KERN_demo_stm32f10x(дата звернення 2019) – Назва з екрана.
26. CMSIS RTOS [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу https://www.keil.com/pack/doc/CMSIS/RTOS/html/CMSIS_RTOS_Tutorial.pdf, (дата звернення 2019) – Назва з екрана.

ДОДАТКИ

ДОДАТОК 1. Базовий комплект налагодження

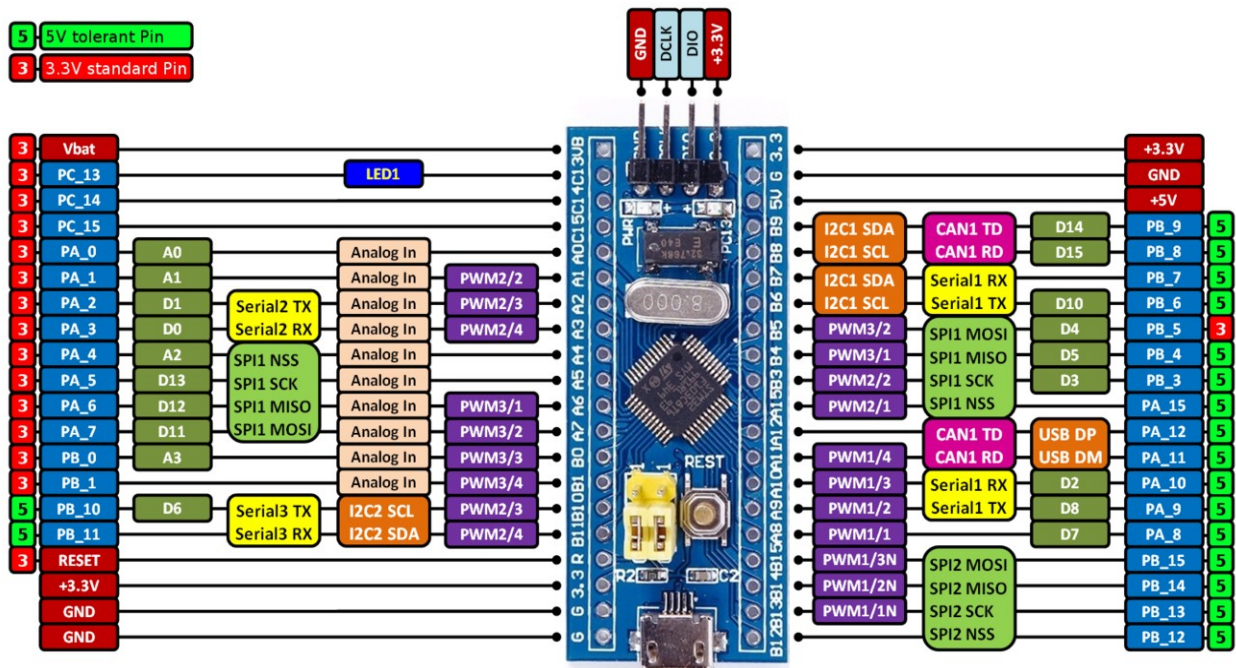


Рисунок 1 – Призначення виводів комплекту налагодження

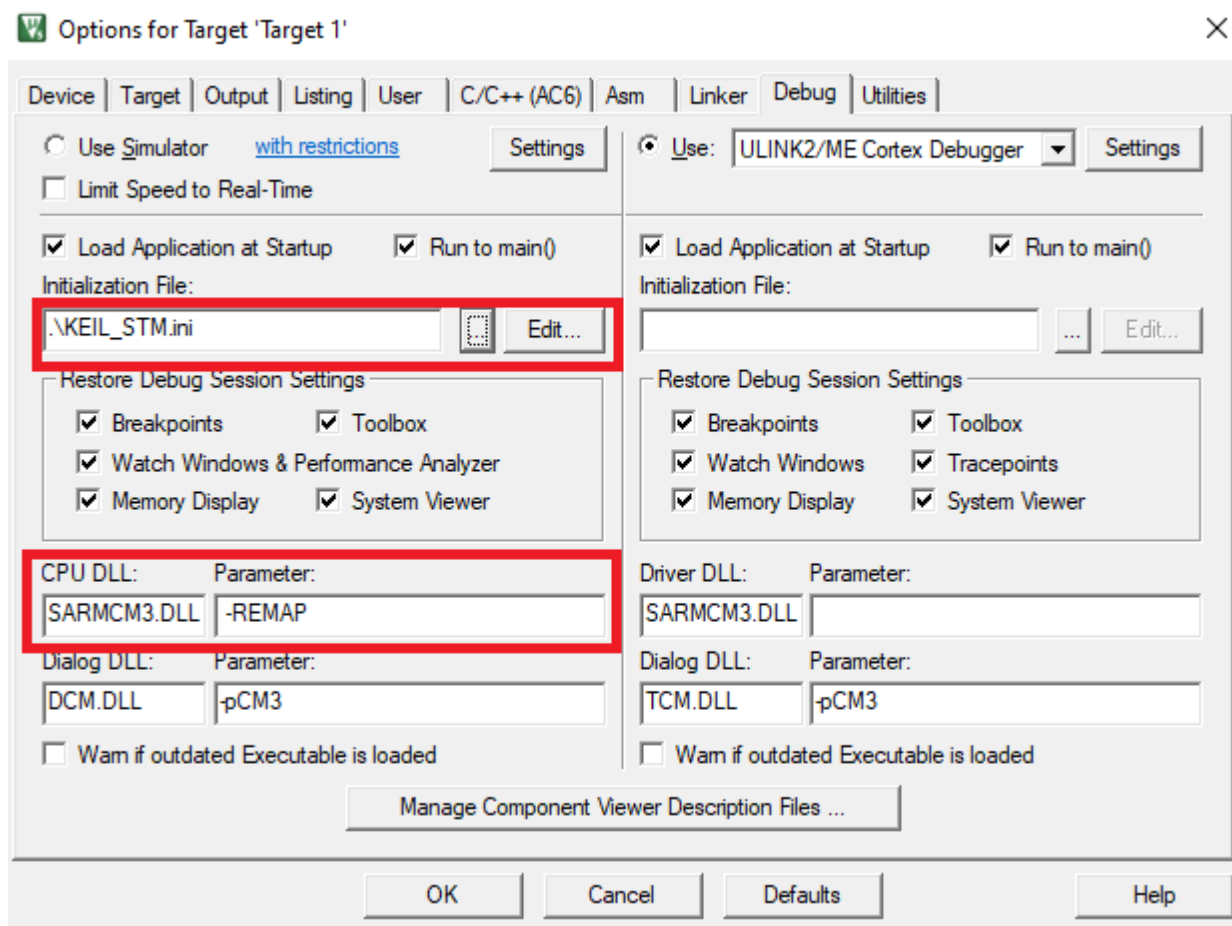


Рисунок 2 – Налаштування з використанням симулятора. Частина 1.

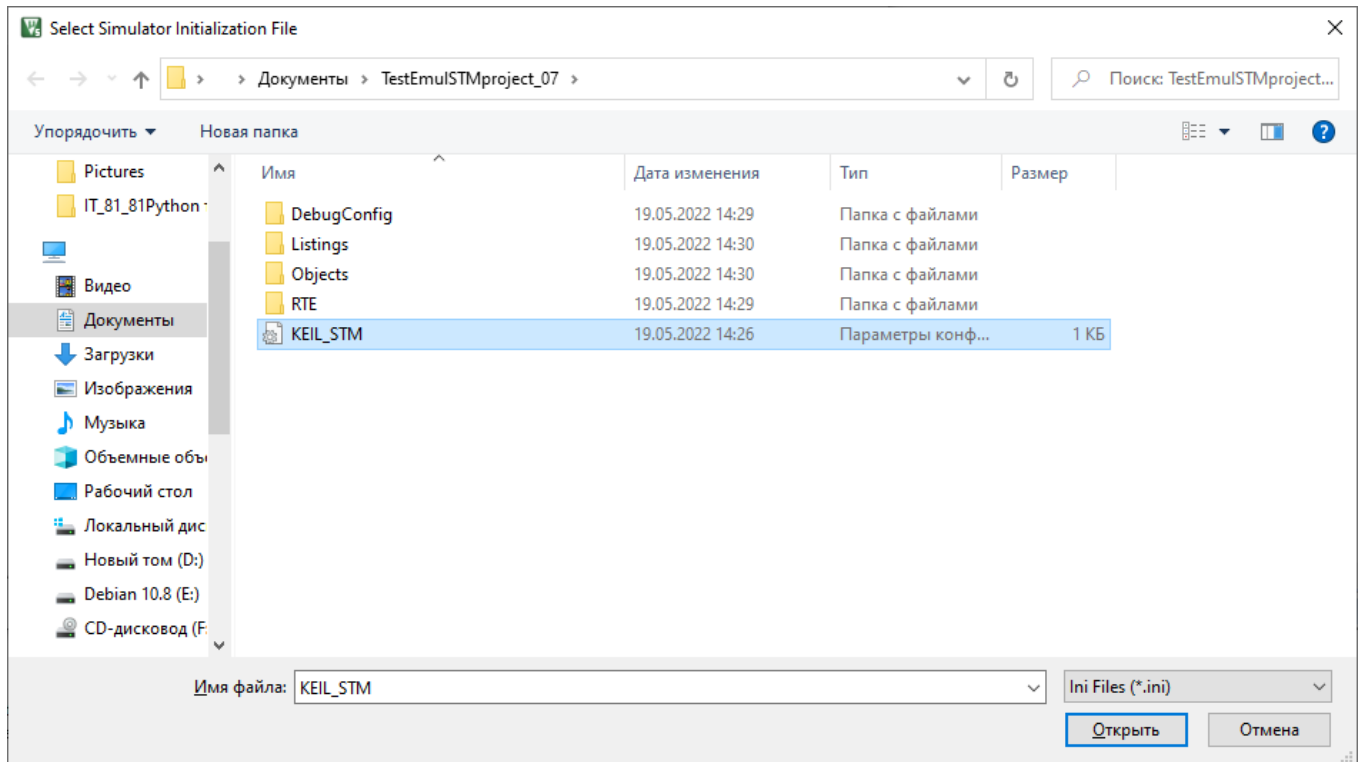


Рисунок 3 – Налagodження з використанням симулятора. Частина 2.

Зміст файлу KEIL_STM.ini

MAP 0x40000000, 0x400FFFFFF read write

Рис. Підготовка емулятору для налагодження прототипу функціонального вузла системи радіоуправління

ДОДАТОК 2. Принципова схема базового комплекту налагодження

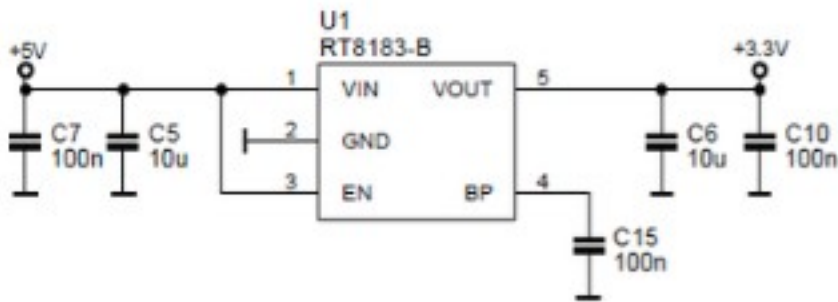


Рисунок 4 – Схема перетворювача напруги живлення

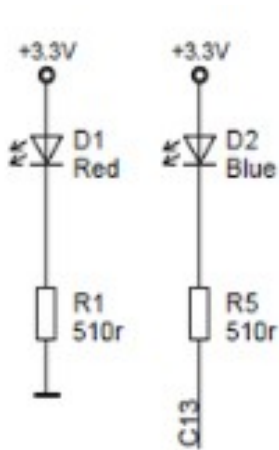


Рисунок 5 – Підсистема сигналізації живлення і програмування МК

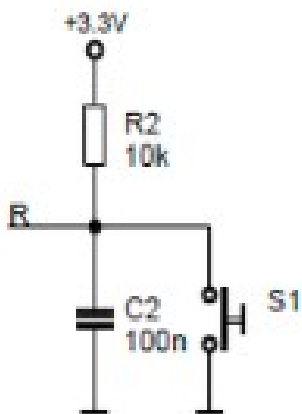


Рисунок 6 – Підсистема скидання (Reset) МК

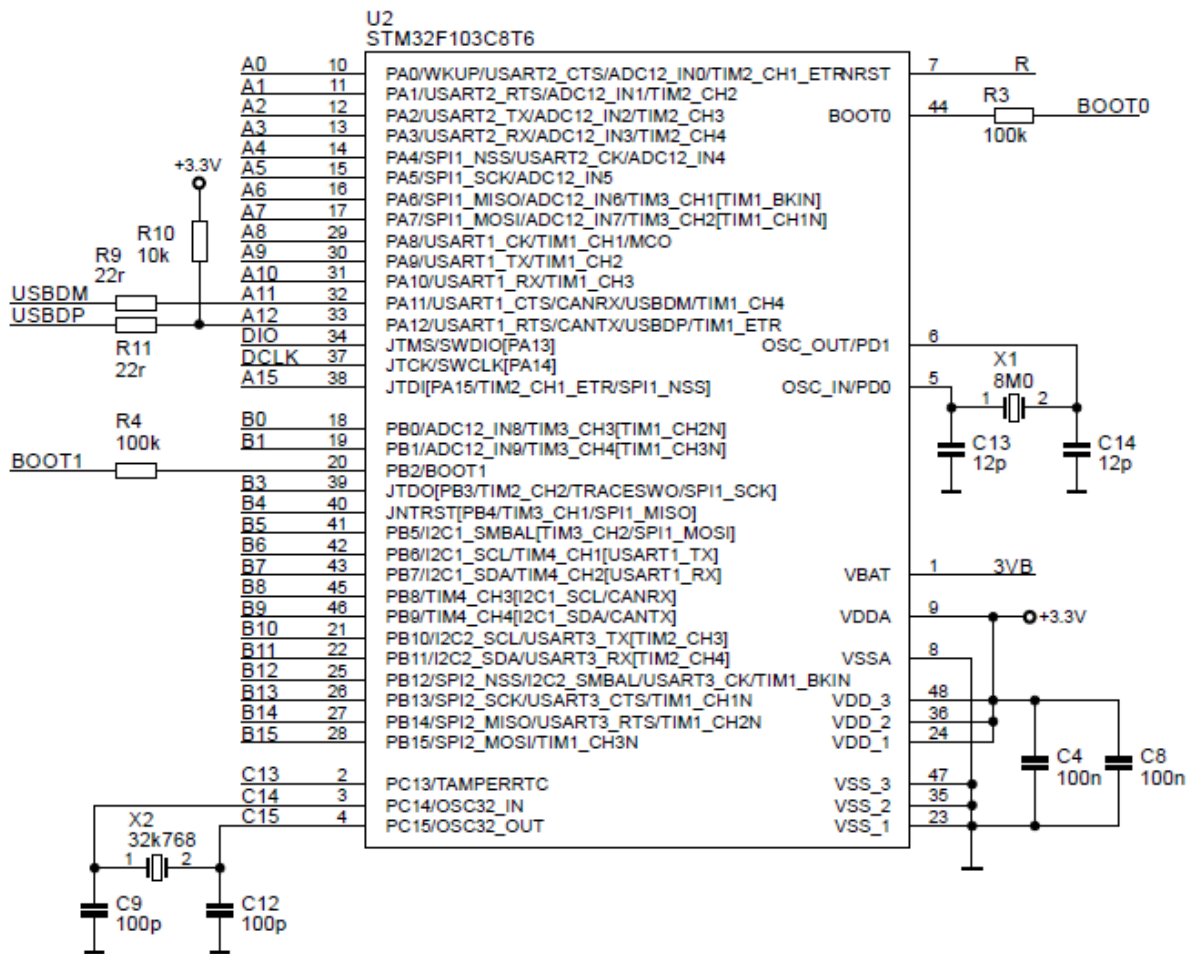


Рисунок 7 – Принципова схема комплекту налагодження



Рисунок 8 – Вигляд сервомашини

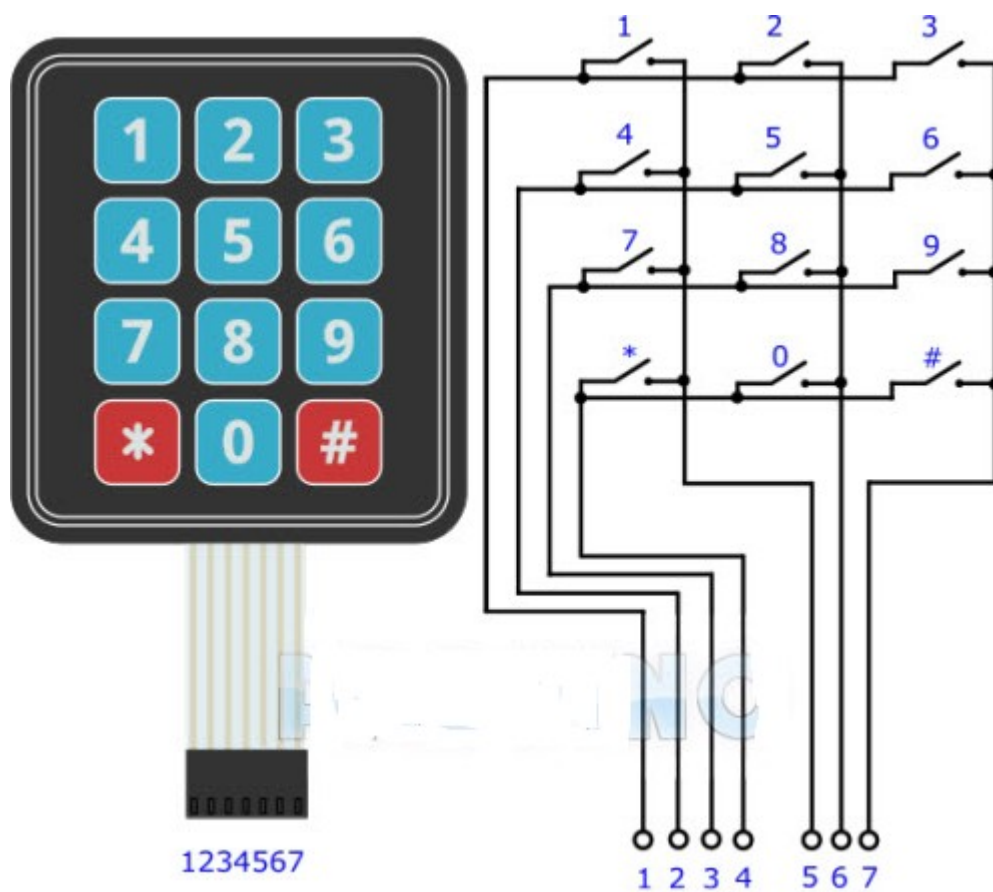


Рисунок 8 – Вигляд і принципова схема мембранної клавіатури на 12 кнопок



Рисунок 9– Вигляд простого джойстика

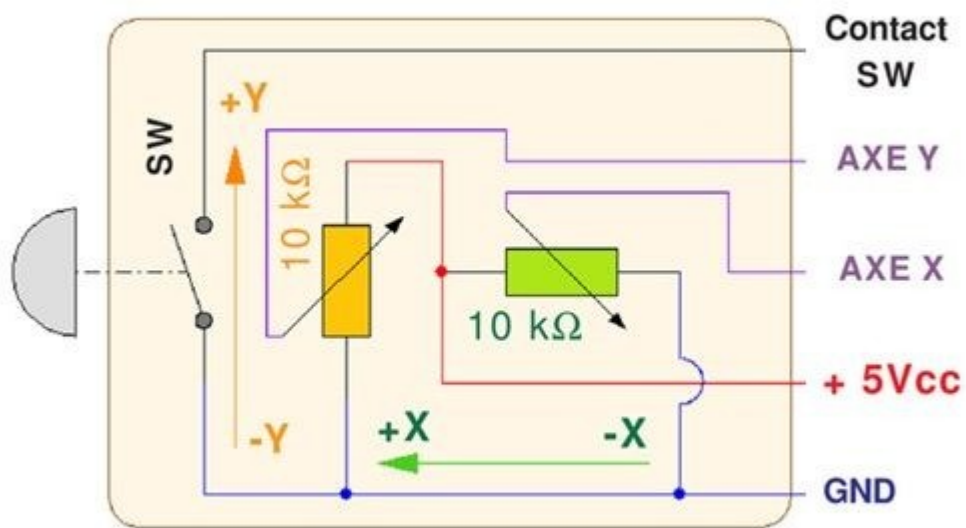


Рисунок 10– Ескіз принципової схеми джойстика