

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»  
КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ  
АПРАТУРИ

## **РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА**

**з курсу:**

**«Обчислювальні та мікропроцесорні засоби в РЕА»**

**тема: «Пристрій керування сервоприводом»**

Керівник:

доц. Корнєв В.П.

Допущено до захисту

“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Захищено з оцінкою

\_\_\_\_\_

Виконав:

Котов Болеслав Вадимович

студент III курсу ФЕЛ

групи ДК-81

**Київ – 2021**

Національний Технічний Університет України  
"Київський Політехнічний Інститут  
імені Ігоря Сікорського,,

Кафедра Конструювання електронно-обчислювальної апаратури  
Дисципліна Обчислювальні та мікропроцесорні засоби в РЕА  
Спеціальність 172 Телекомунікації та радіотехніка  
Курс 3 Група ДК- 81 Семестр VI

## ЗАВДАННЯ

до розрахунково-графічної роботи

Котов Болеслав Вадимович

(призвище, ім'я та по батькові)

1. Тема проекту Керування сервоприводом за допомогою далекоміра
2. Строк здачі студентом закінченого проекту (роботи) 30.05.2021
3. Вихідні данні до проекту (роботи)  
Зпроектувати пристрій керування поворотом вала сервопривода в залежності від відстані ультразвукового далекоміра до перешкоди.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що розробляються)
  1. Опис структури пристрою і його складових
  2. Обґрунтування вибору елементної бази
  3. Опис і розрахунок схеми електричної принципової
  4. Алгоритм роботи програми
5. Перелік графічного матеріала (з точним зазначенням обов'язкових креслень)
  1. Схема електрична принципова
  2. Перелік елементів
6. Дата видачі завдання 31.03.2021

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапу роботи	Строк виконання етапів роботи	Примітка
1	<i>Пошук літератури та аналіз існуючих рішень. Розробка технічного завдання</i>	<i>01.04 – 7.04</i>	
2	<i>Розділ 1. Розробка та опис структури пристрою і його окремих складових</i>	<i>8.04 – 14.04</i>	
3	<i>Розділ 2. Обґрунтування вибору елементної бази.</i>	<i>15.04 – 20.04</i>	
4	<i>Створення схеми електричної принципової та узгодження її з керівником</i>	<i>15.04 – 30.04</i>	
6	<i>Опис і розрахунок схеми електричної принципової</i>	<i>03.05 – 10.05</i>	
7	<i>Розділ 3. Створення структури програмного забезпечення пристрою та алгоритмів роботи програм.</i>	<i>20.04 – 30.04</i>	
8	<i>Розробка програмного забезпечення пристрою.</i>	<i>03.05 – 16.05</i>	
9	<i>Оформлення документації: ЕЗ, ПЕЗ, ПЗ</i>	<i>20.05 – 22.05</i>	
10	<i>Подання до захисту</i>	<i>30.05</i>	
11	<i>Захист РГР</i>	<i>26.05 – 3.06</i>	

Студент \_\_\_\_\_  
(підпис)

Керівник \_\_\_\_\_  
(підпис)

«\_\_\_» \_\_\_\_\_ 2021 р.

**Зміст**

ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ .....	2
ВСТУП.....	3
1. СТРУКТУРНА СХЕМА ПРИСТРОЮ ТА ПРИНЦИП РОБОТИ .....	4
1.1 Структурна схема пристрою.....	4
1.2 Принципи роботи блоку отримання даних з далекоміра.....	4
1.3 Принципи і засоби обробки даних .....	5
1.4. Принципи і засоби відображення вихідних даних і результату .....	6
2.СХЕМА ЕЛЕКТРИЧНА ПРИНЦИПОВА ПРИСТРОЮ.....	7
2.1 Вибір мікроконтролера.....	7
2.2 Проектування блоку отримання даних.....	8
2.3 Проектування блоку обробки даних .....	9
2.4 Проектування блоку виведення даних .....	9
3.ПРОГРАМА КЕРУВАННЯ ПРИСТРІЄМ.....	11
3.1 Структура програми .....	11
3.2 Структура даних.....	12
3.3 Опис алгоритмів окремих підпрограм.....	15
3.3.1 Опис роботи функції GPIO_Config: .....	15
3.3.2 Опис роботи функції TIM3_Config: .....	17
3.3.3 Опис роботи функції TIM4_Config: .....	18
3.3.4 Опис роботи функції TIM3_IRQHandler:.....	19
3.3.5 Опис роботи функції angle().....	20
3.3.6 Опис роботи безкінечного циклу:.....	21
ВИСНОВОК.....	22
ЛІТЕРАТУРА.....	23

					ДК-81460839.001 ПЗ				
Зм.		№ докум.							
Розробив	Котов Б.В.				Блок керування серво-двигуном за допомогою дальноміра	Літ.	Арк	Аркцив	
Перевірів	Корнєв В.П.						1	23	
Реценз.						«КПІ ім. І.Сікорського», ФЕЛ, ДК-81			
Н. Контр.									
Затв.	Корнєв В.П.								

## ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

### 1. Найменування та область використання.

Пристрій керування сервоприводом за допомогою далекоміра призначений для використання в навчальних цілях. Також можливою сферою застосування є автоматика процесів.

### 2. Підстава для розробки.

Підставою для виконання роботи є завдання, видане викладачем згідно навчального плану.

### 3. Мета і призначення роботи.

Метою даної роботи є вивчення принципів керування сервоприводом та ультразвуковим дальноміром та закріплення навичок проектування цифрової апаратури на основі мікроконтролера сімейства STM32. Макет прототипу пристрою може застосовуватися у якості лабораторного стенду при вивченні відповідної теми курсу.

### 4. Джерела розробки.

Технічна документація на мікроконтролер STM32F401RE, сервопривід MG995 та ультразвуковий далекомір HC-SR04.

### 5. Технічні вимоги.

#### 5.1. Функціональні можливості.

Ультразвуковий далекомір повинен визначати відстань від нього до перешкоди на відстані до 4 метрів, і в залежності від відстані керувати поворотом валу серводвигуна. Спосіб передачі сигналу - провідний. Вал може знаходитися в положеннях від 0° до 120°. 0° - якщо відстань до перешкоди мінімальна, 120° - якщо відстань до перешкоди максимальна.

#### 5.2 Технічні характеристики.

Напруга живлення серводвигуна: +5 В.

Напруга живлення далекоміра: +5 В.

Напруга живлення мікроконтролера: +3.3 В.

Довжина імпульсу вихідного ШІМ сигналу: 500-2500 мкс.

					ДК81.460839.001 ПЗ	Арк
						2
Зм.	Арк	№ док.	Підпис	Дата		

## ВСТУП

Серводвигун - це двигун з управлінням через негативний зворотний зв'язок, що дозволяє точно керувати параметрами руху. Сервомотором є будь-який тип механічного приводу, що має в складі датчик положення і плату управління. Простими словами, серводвигун - це механізм з електродвигуном, який може повертатися в заданий кут і утримувати поточний стан.

Ультразвуковий далекомір – пристрій, що здатен вимірювати відстань до перешкоди.

Для користування цими модулями використовують мікроконтролери.

Мікроконтролер — виконана у вигляді мікросхеми спеціалізована мікропроцесорна система, що включає мікропроцесор, блоки пам'яті для збереження коду програм і даних, порти вводу-виводу і блоки зі спеціальними функціями (лічильники, компаратори, АЦП та інші). Використовується для керування електронними пристроями. Використання однієї мікросхеми значно знижує розміри, енергоспоживання і вартість пристроїв, побудованих на базі мікроконтролерів.

Мікроконтролери можна зустріти в багатьох сучасних приладах, таких як телефони, пральні машини, вони відповідають за роботу двигунів і систем гальмування сучасних автомобілів, з їх допомогою створюються системи контролю і системи збору інформації. STM32 – популярна лінійка мікроконтролерів на основі сімейства мікропроцесорних ядер ARM-Cortex-M.

					ДК81.460839.001 ПЗ	Арк
						3
Зм.	Арк	№ док-м.	Підпис	Дата		

# 1. СТРУКТУРНА СХЕМА ПРИСТРОЮ ТА ПРИНЦИП РОБОТИ

## 1.1 Структурна схема пристрою

На рис 1.1 зображено структурну схему пристрою, яка складається з блоку отримання даних з далекоміра, блоку обробки даних та блок керування серводвигуном.

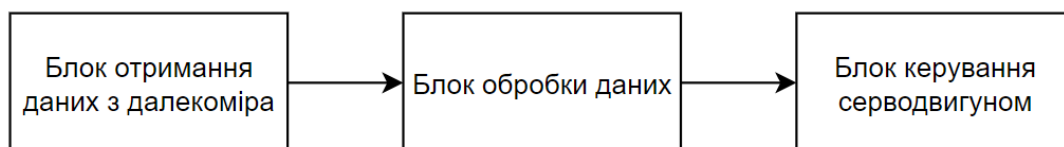


Рис.1.1 - Структурна схема пристрою

## 1.2 Принципи роботи блоку отримання даних з далекоміра

Робота модуля заснована на принципі ехолокації. Модуль посилає ультразвуковий сигнал і приймає його відображення від об'єкта.

Для того щоб ініціювати відправку сигналу далекоміром, необхідно подати високий сигнал тривалістю 10  $\mu$ s(мкс.) на пін Trig. Для цього ми використовуємо ШІМ (Широтно-Імпульсна Модуляція).

Після отримання високого сигналу тривалістю 10  $\mu$ s на пін Trig, модуль генерує пучок з восьми сигналів частотою 40 кГц і встановлює високий рівень на пині Echo. Після отримання відбитого сигналу модуль встановлює на пині Echo низький рівень. На рис. 1.2.1 наведено принцип роботи далекоміра, тобто надані сигнали керування.

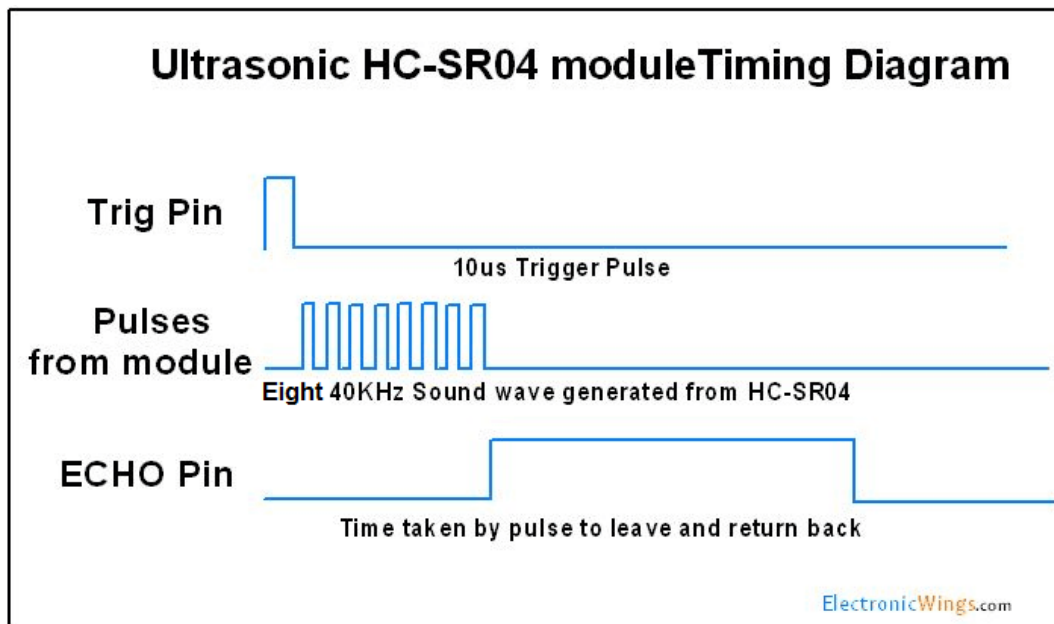


Рис. 1.2.1 Сигнали керування далекоміром

Знаючи тривалість високого сигналу на піні Echo можемо обчислити відстань, помноживши час, який витратив звуковий імпульс, перш ніж повернувся до модуля(duration), на швидкість поширення звуку в повітрі (340 м / с). Тепер обчислимо відстань перевівши швидкість з м/с в см/мкс:

$$distance = duration \cdot 340 \frac{\text{м}}{\text{с}} = duration \cdot 0.034 \frac{\text{м}}{\text{мкс}},$$

Беручи до уваги те, що звук подолав відстань до об'єкта і назад, поділимо отриманий результат на 2, тобто результуюча формула має наступний вигляд:

$$distance = duration \cdot \frac{0.034 \text{ м}}{2 \text{ мкс}}$$

### 1.3 Принципи і засоби обробки даних

Зчитане значення піну echo ультразвукового далекоміру аналізує мікроконтролер, робить обчислення яке було описано у п. 1.2, і в залежності від відстані до перешкоди, змінюється значення довжини імпульсу ШІМ сигналу, занесенням необхідного значення в регістр порівняння таймеру TIM3 каналу 3. Період імпульсів 20 мс завдається значенням, яке заноситься у регістр перезавантаження ARR таймеру TIM3.



#### 1.4. Принципи і засоби відображення вихідних даних і результату

Отриманий ШІМ сигнал з порта РС8 поступає на вхід керування серводвигуна, і в залежності від довжини імпульсу даного сигналу змінюється положення валу сервоприводу, а саме:  $0^\circ$  - довжина імпульсу 500 мкс,  $90^\circ$  - довжина імпульсу 1500 мкс,  $120^\circ$  - довжина імпульсу 2500 мкс (рис.1.4.1).

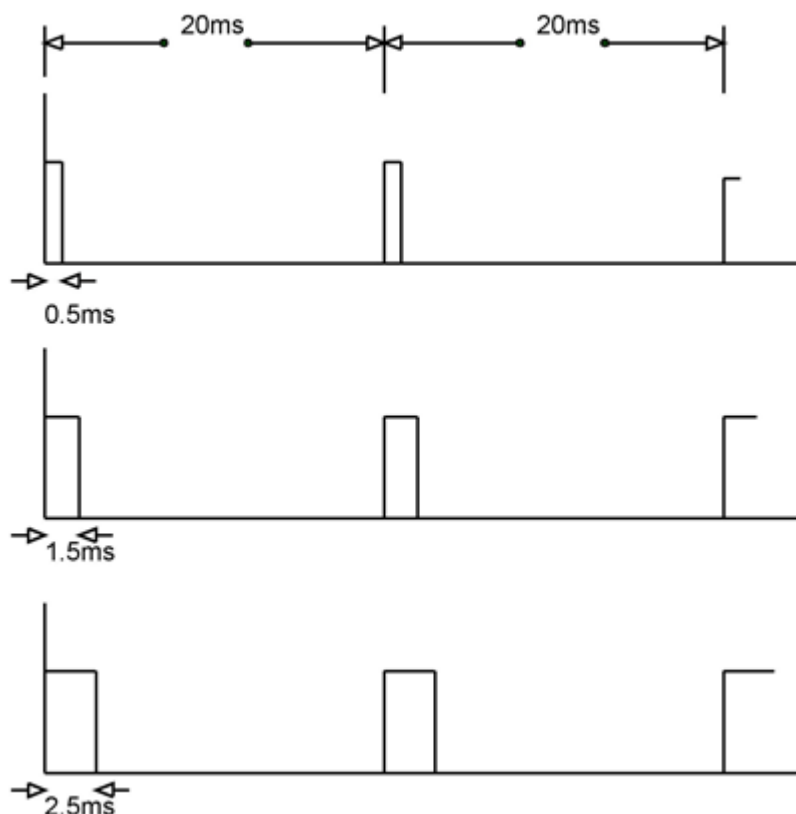


Рис.1.4.1 – Діаграма ШІМ сигналу для серводвигуна.

## 2.СХЕМА ЕЛЕКТРИЧНА ПРИНЦИПОВА ПРИСТРОЮ

Схема електрична принципова зображена на рисунку 2.1. В проекті використовується зовнішнє джерело постійної напруги яке формує стійку напругу 5В.

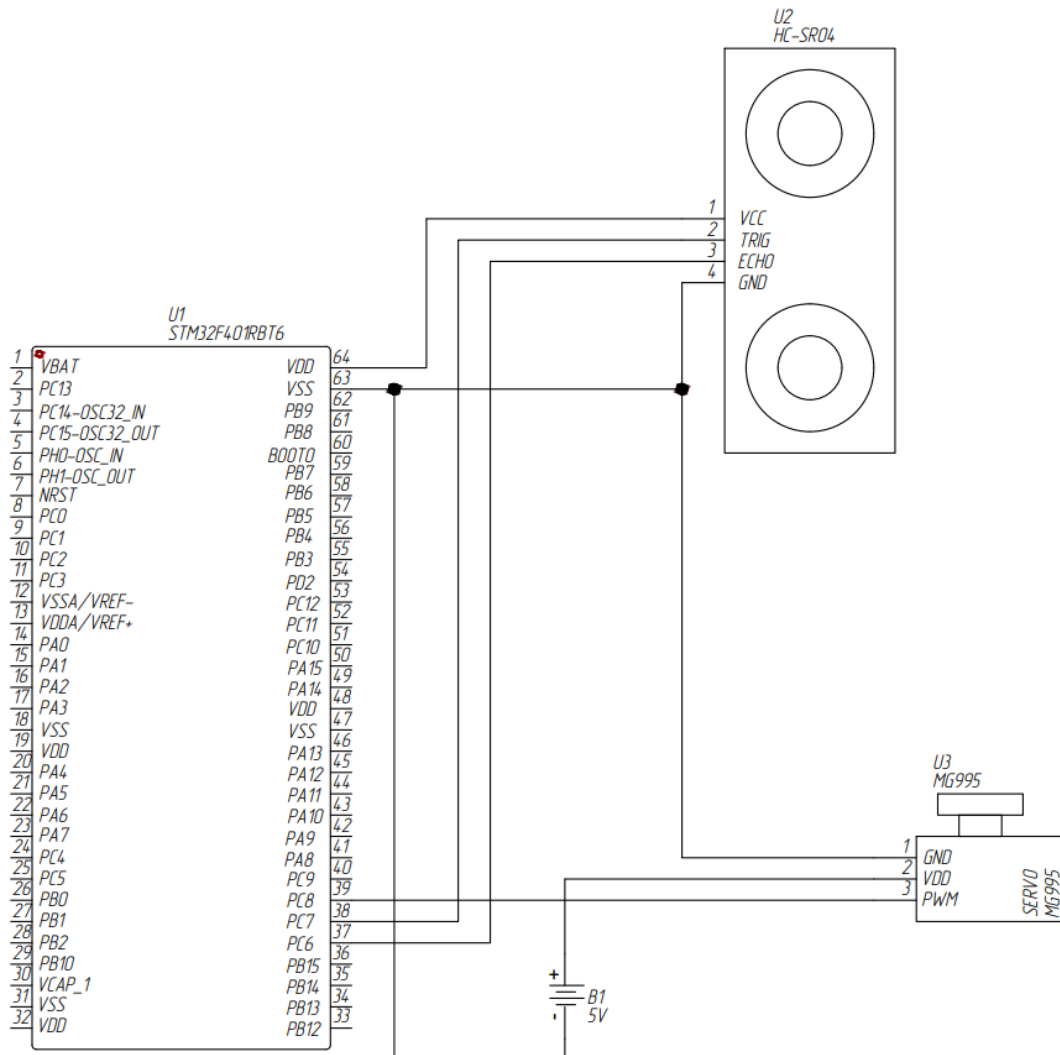


Рис. 2.1 Принципова схема приладу

### 2.1 Вибір мікроконтролера

Центральною ланкою в пристрою керування сервоприводом та далекоміра є блок, що оброблятиме дані, отримані з блоку введення даних та формуватиме дані для передачі на блок виведення даних. Згідно технічного завдання, це буде мікроконтролер STM32F401RE. Він ідеально підходить для поставленої задачі.

					ДК81.460839.001 ПЗ	Арк
Зм.	Арк	№ док.	Підпис	Дата		7

Мікросхему виконано у корпусі LQFP (рис. 2.1)

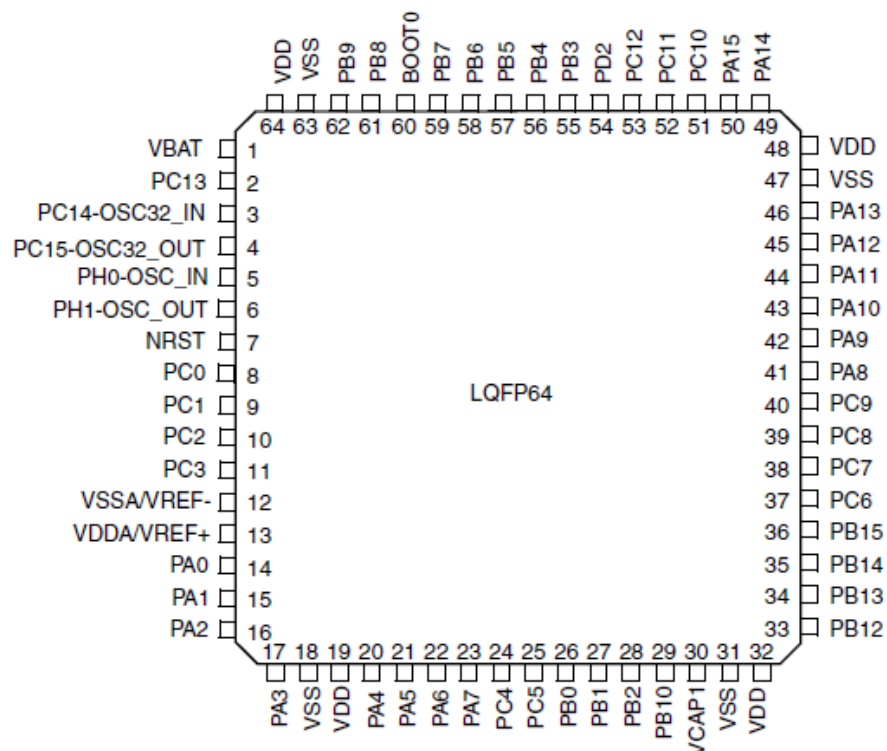


Рис 2.1.1 – мікроконтролер STM32F401RE у корпусі LQFP

Характеристики мікроконтролера:

- Core: ARM® 32-bit Cortex® -M4
- 512 KB Flash пам'ять
- 96 KB SRAM
- Частота вбудованого генератора синхроімпульсів HSI - 8 MHz
- Напруга живлення 1.7 В - 3.6 В

## 2.2 Проектування блоку отримання даних

Основою блоку отримання даних є далекомір. Для цього був використаний ультразвуковий далекомір HC-SR04, який живиться від STM32F401-Nucleo. Пін Trig підключений до PC7, пін Echo підключений до PC6. (Рис. 2.2)

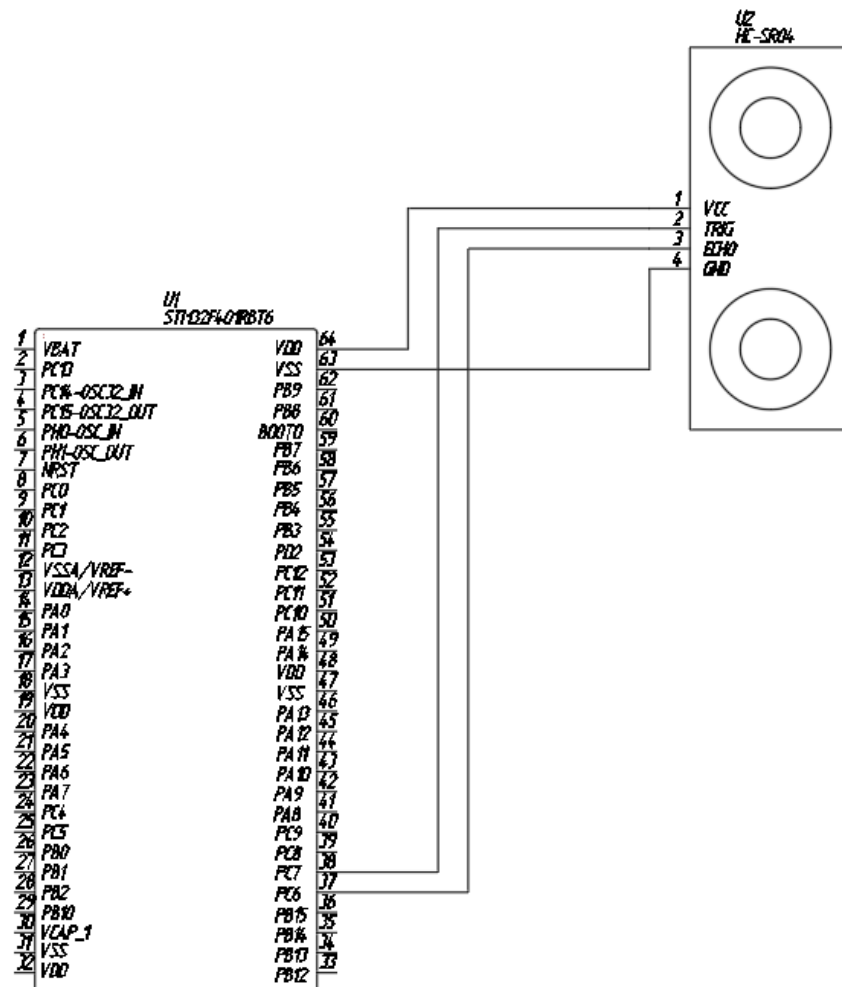


Рис. 2.2.1 Схема блоку отримання даних

## 2.3 Проектування блоку обробки даних

В якості схеми блоку обробки даних використовується плата STM32F401-Nucleo.

Тактування здійснюється підключенням внутрішнього тактового RC - генератора (HSI) з частотою 16 МГц, після проходження через переддільників (Prescaler), які в нашому випадку дорівнюють 1, надходить до шин без зміни, тобто шини тактуються з частотою 16 МГц.

## 2.4 Проектування блоку виведення даних

Блок виведення даних складається з сервоприводу, вхід керування якого підключається до порту PC8 мікроконтролеру. Сервопривод живиться від зовнішнього джерела з напругою 5 В. Схема блоку відображення даних зображено на рис. 2.4.

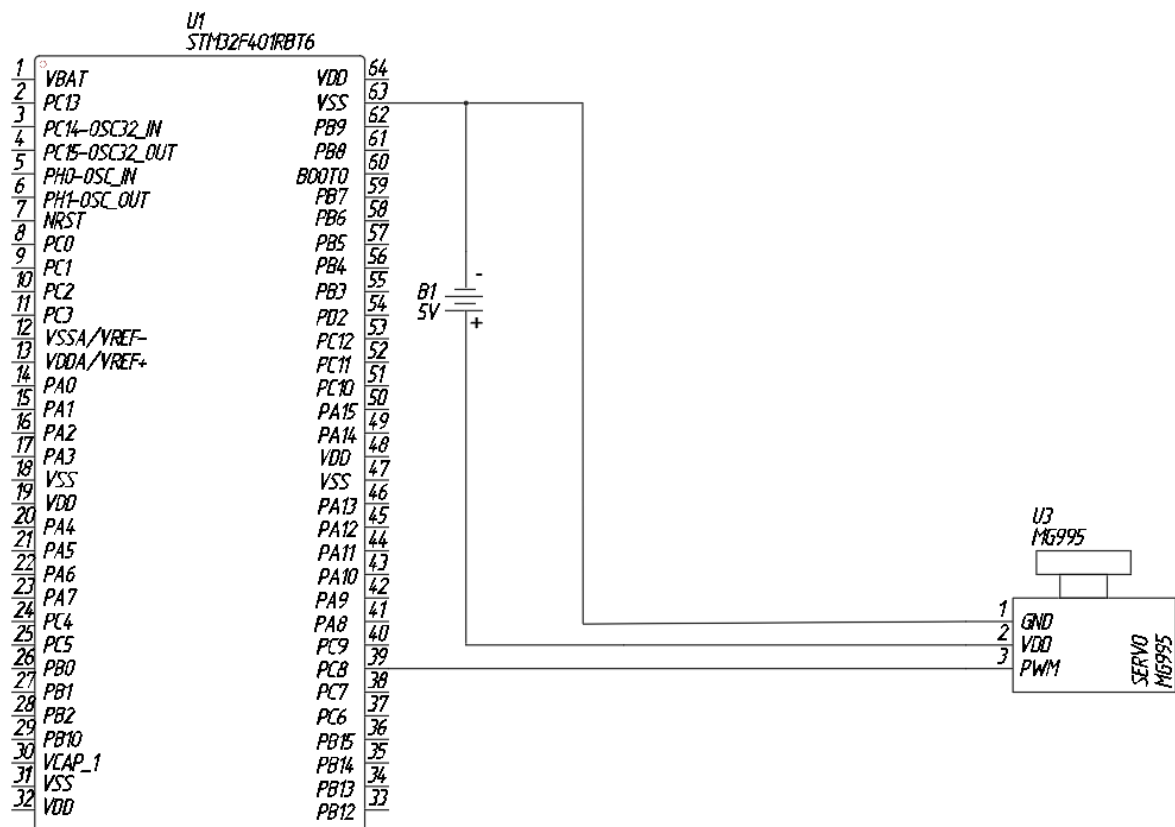


Рис. 2.4.1 Схема блоку виведення даних

### 3.ПРОГРАМА КЕРУВАННЯ ПРИСТРІЄМ

#### 3.1 Структура програми

Програма керування пристроєм написана на мові програмування C з використанням бібліотеки CMSIS. Виконання програми здійснюється у функції main(). Спочатку визивається функція GPIO\_Config(), яка конфігурує порти вводу-виводу, як альтернативні. Після цього викликається функція TIM3\_Config(), де налаштовується два канали. Перший канал відповідає за отримання сигналу з піну Echo(PC6) на далекомірі, тобто перший канал налаштовуємо як input capture. Другий канал використовується для піну Trig(PC7) далекоміра, і форму ШІМ на цьому піні. Далі дозволяємо таймеру почати відлік, а також дозволяємо переривання, у якому буде відбуватись зчитування значень з далекоміру і наступна обробка цих значень. Переривання буде відбуватись за обома фронтами сигналу з піну Echo(PC6), а також по переповненню таймера. Наступною функцією визивається TIM4\_Config(), яка відповідає за конфігурацію таймера 4 каналу 3. Цей таймер створює ШІМ сигнал для серводвигуна на піні PB8.

У вічному циклі викликається функція angle() у яку передається у якості аргументу змінну Distance, що містить в собі відстань до перешкоди. Функція angle() перетворює значення змінної з одного діапазону в інший. Тобто у загальному вигляді значення змінної з ім'ям «value», яке дорівнює «fromLow», буде перетворено в число «toLow», а значення «fromHigh» – у значення «toHigh». Всі проміжні значення «value» масштабуються відносно нового діапазону від «toLow» до «toHigh». У нашому випадку значення імпульсу для двигуна знаходиться у діапазоні від 500 мс. до 2500 мс. А значення з далекоміра може приймати діапазон від ~2 см. до 400 см. За допомогою функції angle ми перетворюємо значення далекоміра у діапазон значень для серводвигуна. Потім функція angle() повертає значення довжини імпульсу серводвигуна і ми змінюємо значення імпульсу для ШІМ.

					ДК81.460839.001 ПЗ	Арк
Зм.	Арк	№ док.м.	Підпис	Дата		11

### 3.2 Структура даних

Таблиця 3.2.1 Використані змінні

Назва	Тип змінної	Призначення
Distance	32 розрядна змінна типу unsigned long	Збереження значення відстані до перешкоди
pulse_width	32 розрядна змінна типу unsigned long	Довжина вхідного імпульсу далекоміра
current_captured	32 розрядна змінна типу unsigned long	Зберігається поточне значення фронту вхідного імпульсу далекоміра
last_captured	32 розрядна змінна типу unsigned long	Зберігається минуле значення фронту вхідного імпульсу далекоміра
signal_polarity	32 розрядна змінна типу unsigned long	Зберігається полярність сигналу для того щоб ми рахували тільки довжину сигналу у стані лог. «1»
temp	32 розрядна змінна типу unsigned long	Використовується для перетворення значення з діапазону відстані до діапазону серводвигуна

Таблиця 3.2.2 Використані виводів плати NUCLEO

Назва	Призначення
PB8	Комутація сервоприводу з мікроконтролером

PC6	Комутація мікроконтролера з піном Echo далекоміра
PB7	Комутація мікроконтролера з з піном Trig далекоміра

Таблиця 3.2.3 Регістри мікроконтролера, застосовані в програмі

Назва	Призначення
AHB1ENR	Ввімкнення тактування портів GPIOB та GPIOC
APB1ENR	Дозвіл тактування системної шини, TIM3 та TIM4
GPIOx (MODER)	Налаштування роботи PB8, PC6, PC7 на альтернативну функцію
GPIOx (AFRL)	Вибір альтернативної функції PB8, PC6, PC7
TIMx(PSC)	Задає значення переддільника частоти таймеру
TIMx(CR1)	Налаштування таймера (напрям рахунку, вирівнювання)
TIMx(ARR)	Задає значення до якого буде рахувати таймер (завдання періоду імпульсів)
TIM3 (CCMR1)	Вибір режиму input capture першого каналу та PWM1 другого каналу для таймера 3
TIM3(CCER)	Встановлюємо активний фронт. Встановлення полярності вихідного ШІМ сигналу та дозвіл ШІМ на вихід



TIM3(CCR2)	Задає довжину імпульсу ШІМ сигналу (разом із ARR визначає скважність)
TIM3 (BDTR)	Ввімкнення головного виходу каналу(MOE)
TIM3(CR1)	Ввімкнення таймеру
TIM3(DIER)	Дозвіл переривання по переповненню таймера
TIM3(CCER)	Встановлення полярності вихідного ШІМ сигналу та дозвіл ШІМ на вихід
NVIC_SetPriority	Встановлення пріоритету обробника переривань
NVIC_EnableIRQ	Вмикаємо переривання
TIM4(CCMR2)	Вибір режиму роботи каналу як PWM1, CCR1 встановлюється після встановлення UEV flag, тобто після переповнення таймеру
TIM4 (CCER)	Встановлення полярності вихідного ШІМ сигналу та дозвіл ШІМ на вихід
TIM4 (CCR3)	Встановлення довжини імпульсу ШІМ для таймеру 4 каналу 3
TIM3(CR1)	Ввімкнення таймеру

### 3.3 Опис алгоритмів окремих підпрограм

#### 3.3.1 Опис роботи функції GPIO\_Config:

В даній функції відбувається включення тактування системної шини АНВ1, порту GPIOB та GPIOC. Для цього необхідно налаштувати регістр АНВ1ENR з блоку регістрів RCC (Reset and Clock Control). Біт 1 (GPIOBEN) та біт 2 (GPIOCEN) необхідно встановити в 1 (Рис. 3.3.1).

#### 6.3.5 RCC АНВ1 peripheral reset register (RCC\_AHB1RSTR)

Address offset: 0x10

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2 RST	DMA1 RST	Reserved				
									rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCSRST	Reserved				GPIOH RST	Reserved	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST	
			rw					rw		rw	rw	rw	rw		

Рис. 3.3.1 Біти шини АНВ1

Далі йде конфігурування PB8, PC6, PC7 як альтернативної функції. Для цього у регістрі GPIOB(MODER) та GPIOC(MODER) з блоку регістрів GPIO необхідно біти 13 та 12 (MODER6) для PC6, біти 14 та 15 (MODER7) для PC7, і біти 16 та 17 (MODER8) для PB8 встановити відповідно у 1 та 0. (Рис. 3.3.2)

### 8.4.1 GPIO port mode register (GPIOx\_MODER) (x = A..E and H)

Address offset: 0x00

Reset values:

- 0x0C00 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y:2y+1 **MODERy[1:0]**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O direction mode.

00: Input (reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

Рис. 3.3.2 Встановлення режиму роботи портів вводу-виводу

Далі вибирається тип альтернативної функції. Для PC6 – TIM3\_CH1(Рис. 3.3.3), PC7 – TIM3\_CH2, PB8 – TIM4\_CH1(Рис. 3.3.4).

Table 9. Alternate function mapping (continued)

Port	AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
	SYS_AF	TIM1/TIM2	TIM3/TIM4/TIM5	TIM9/TIM10/TIM11	I2C1/I2C2/I2C3	SPI1/SPI2/I2S2/SPI3/I2S3/SPI4	SPI2/I2S2/SPI3/I2S3	SPI3/I2S3/USART1/USART2	USART6	I2C2/I2C3	OTG1_FS		SDIO			
PortB	PB0	-	TIM1_CH2N	TIM3_CH3	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PB1	-	TIM1_CH3N	TIM3_CH4	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PB2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PB3	JTDO-SWO	TIM2_CH2	-	-	SPI1_SCK	SPI3_SCK/I2S3_CK	-	-	I2C2_SDA	-	-	-	-	-	EVENT OUT
	PB4	JTRST	-	TIM3_CH1	-	SPI1_MISO	SPI3_MISO	I2S3ext_SD	-	I2C3_SDA	-	-	-	-	-	EVENT OUT
	PB5	-	-	TIM3_CH2	-	I2C1_SMBA	SPI1_MOSI	SPI3_MOSI/I2S3_SD	-	-	-	-	-	-	-	EVENT OUT
	PB6	-	-	TIM4_CH1	-	I2C1_SCL	-	USART1_TX	-	-	-	-	-	-	-	EVENT OUT
	PB7	-	-	TIM4_CH2	-	I2C1_SDA	-	USART1_RX	-	-	-	-	-	-	-	EVENT OUT
	PB8	-	-	TIM4_CH3	TIM10_CH1	I2C1_SCL	-	-	-	-	-	-	SDIO_D4	-	-	EVENT OUT
	PB9	-	-	TIM4_CH4	TIM11_CH1	I2C1_SDA	SPI2_NSS/I2S2_WS	-	-	-	-	-	SDIO_D5	-	-	EVENT OUT
	PB10	-	TIM2_CH3	-	-	I2C2_SCL	SPI2_SCK/I2S2_CK	-	-	-	-	-	-	-	-	EVENT OUT
	PB12	-	TIM1_BKIN	-	-	I2C2_SMBA	SPI2_NSS/I2S2_WS	-	-	-	-	-	-	-	-	EVENT OUT
	PB13	-	TIM1_CH1N	-	-	-	SPI2_SCK/I2S2_CK	-	-	-	-	-	-	-	-	EVENT OUT
	PB14	-	TIM1_CH2N	-	-	-	SPI2_MISO	I2S2ext_SD	-	-	-	-	-	-	-	EVENT OUT
	PB15	RTC_REFN	TIM1_CH3N	-	-	-	SPI2_MOSI/I2S2_SD	-	-	-	-	-	-	-	-	EVENT OUT

Рис. 3.3.3 Таблиця альтернативних функцій для GPIOB

Table 9. Alternate function mapping (continued)

Port		AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
		SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/SPI2/ I2S2/SPI3/ I2S3/SPI4	SPI2/I2S2/ SPI3/ I2S3	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS		SDIO			
Port C	PC0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC2	-	-	-	-	-	SPI2_ MISO	I2S2ext_SD	-	-	-	-	-	-	-	-	EVENT OUT
	PC3	-	-	-	-	-	SPI2_ MOSI /I2S2_SD	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC6	-	--	TIM3_CH1	-	-	I2S2_MCK	-	-	USART6_ TX	-	-	-	SDIO_ D6	-	-	EVENT OUT
	PC7	-		TIM3_CH2	-	-	-	I2S3_MCK	-	USART6_ RX	-	-	-	SDIO_ D7	-	-	EVENT OUT
	PC8	-	-	TIM3_CH3	-	-	-	-	-	USART6_ CK	-	-	-	SDIO_ D0	-	-	EVENT OUT
	PC9	MCO_2	-	TIM3_CH4	-	I2C3_SDA	I2S_CKIN	-	-	-	-	-	-	SDIO_ D1	-	-	EVENT OUT
	PC10	-	-	-	-	-	-	SPI3_SCK/ I2S3_CK	-	-	-	-	-	SDIO_ D2	-	-	EVENT OUT
	PC11	-	-	-	-	-	I2S3ext_ SD	SPI3_MISO	-	-	-	-	-	SDIO_ D3	-	-	EVENT OUT
	PC12	-	-	-	-	-	-	SPI3_MOSI/ I2S3_SD	-	-	-	-	-	SDIO_ CK	-	-	EVENT OUT
	PC13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PC15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT

Рис. 3.3.4 Таблиця альтернативних функцій для GPIOC

Функція обирається шляхом встановлення бітів 27, 26, 25, 24 (AFRL6) для PC6, регістра GPIOC(AFRL) у комбінацію 0010 (AF2). Далі робимо теж саме для портів PC7 та PB8, тобто для PC7 встановлюємо біти, 31, 30, 28, 29 (AFRL) регістра GPIOC(AFRL) у комбінацію 0010 (AF2). Для PB8 встановлюємо біти 3, 2, 1, 0 (AFRH) регістра GPIOC(AFRH) у комбінацію 0010.

### 3.3.2 Опис роботи функції TIM3\_Config:

В даній функції виконується налаштування таймеру 4, каналу 1, який буде у режимі Input capture та буде вимірювати тривалість імпульсу далекоміра, та каналу 2, який буде налаштований як ШІМ для далекоміра. Спочатку вмикаємо тактування таймеру встановивши біт у регістрі APB1ENR. Далі встановлюємо переддільник у значення 15 шляхом занесення у регістр PSC значення 16-1(тобто 15). Після встановлення переддільника, частота буде розраховуватись так:

$$F_{ClockCNT} = \frac{F_{ClockPSC}}{Prescaler + 1} = \frac{16 \cdot 10^6}{15 + 1} = 1 \cdot 10^6 \text{ Гц} = 1 \text{ МГц}$$

Тобто тактова частота буде дорівнювати 1 МГц, звідси період одного такту буде дорівнювати 1 мкс. Далі встановлюємо максимальне значення до якого буде рахувати лічильник шляхом занесення у регістр ARR значення 0xffff. Тобто лічильник буде рахувати до значення 65535. Розрахувати період лічильника можна наступним чином:

$$T_{CNT} = (1 + ARR) \cdot \frac{1}{F_{ClockCNT}} = (1 + 65535) \cdot \frac{1}{1 \cdot 10^6} \approx 0.065 = 65 \text{ мс.}$$

Отже період лічильника буде дорівнювати 65 мс. Далі налаштовується перший канал у режим input capture. Для того щоб вибрати режим input capture, встановлюємо лог. 1 у біт 0 і лог. 0 у біт 1 регістру CCMR1. Так як фільтрація нам не потрібна, ми її вимикаємо шляхом занесення лог.0 у відповідний біт регістру CCMR1. Далі обираємо активний фронт по якому буде виконуватись переривання. У нашому випадку це обидва фронти, тобто «both edges». Для цього заносимо лог. 1 у два відповідних біта регістру CCER, які відповідають за передній і задній фронти. Потім вимикаємо режим input capture і дозволяємо переривання. Далі налаштовується канал 2 того ж самого таймеру 3. Його налаштовуємо як ШІМ, режим 1. Тобто коли в нас лічильник менший за значення регістру CCR2, то на виході буде сигнал високого рівню, а коли лічильник буде мати більше значення ніж CCR2, то на виході буде сигнал низького рівню. Для цього встановлюємо відповідні біти в лог. 1 у регістр CCMR1 (OC2M = 110). Далі встановлюємо Preload біт для того щоб значення у регістрі CCR2 не змінювалось відразу, а змінювалось тільки після переповнення лічильнику. Після цього у регістрі CCER налаштовуємо полярність сигналу і вмикаємо вихід каналу 2. Вмикаємо головний вихід(MOE) у регістрі BDTR. Встановлюємо регістр CCR2 у значення 10. Тобто тривалість високого сигналу ШІМ буде становити 10 мкс. Далі включаємо таймер, встановлюємо пріоритетність переривання і вмикаємо переривання.

### 3.3.3 Опис роботи функції TIM4\_Config:

Цей таймер використовується як ШІМ для серводвигуна. Період ШІМу повинен бути 20 мс, а тривалість імпульсу повинна бути від 500 до 2500 мкс.

					ДК81.460839.001 ПЗ	Арк
Зм.	Арк	№ док.	Підпис	Дата		18

Спочатку вмикається тактування таймеру TIM4 встановленням у 1 біта 2 (TIM4EN) регістра APB1ENR з блоку регістрів RCC. Задається значення переддільника частоти – 16-1, занесенням даної константи у регістр PSC. Значення ARR встановлюємо у 0x4E20, тобто 20 000. Порахувавши за формулою наведеною у п. 3.3.2 період сигналу буде дорівнювати 20 мс. Встановлюємо канал як ШІМ з режимом 1 (OC3M = 110). Для цього встановлюємо відповідні біти в лог. 1 у регістр CCMR2. Далі встановлюємо Preload біт у регістрі CCMR2. Після цього у регістрі CCER налаштовуємо полярність сигналу і вмикаємо вихід каналу 3. Вмикаємо головний вихід(MOE) у регістрі BDTR. Встановлюємо регістр CCR2 у значення 1520. Тобто тривалість високого сигналу ШІМ буде становити 1520 мкс. Далі включаємо таймер. Отримано ШІМ з періодом 20 мс.

### 3.3.4 Опис роботи функції TIM3\_IRQHandler:

Ця функція є обробником переривань. Тобто коли трапляється переривання з таймеру 3 каналу 2, то ми потрапляємо у цю функцію.

У нашому випадку ми потрапляємо у цю функцію, коли приходить передній або задній фронт з далекоміра. Щоб визначити тривалість імпульсу ми зберігаємо значення таймеру щойно надійшовшого фронту у змінну current\_captured. У змінну last\_captured зберігаємо минулий фронт з current\_captured у кінці всіх дій. Тобто у current\_captured записуємо значення регістру CCR1. Далі ми змінюємо полярність сигналу у змінній signal\_polarity, для того щоб рахувати довжину сигналу тільки коли він знаходиться у стані лог. 1 і ігнорувати довжину коли сигнал знаходиться у стані лог. 0. Наступним кроком ми робимо перевірку стану полярності (signal\_polarity), якщо полярність дорівнює 0, то буде розраховуватись довжина імпульсу. Також тут потрібно зробити механізм захисту коли передній фронт прийшов на одному періоді таймера, а задній фронт прийшов коли вже наступний період таймеру. Для цього після перевірки на полярність ми також перевіряємо чи більше current\_captured за last\_captured. Якщо так то все добре і фронти знаходяться на одному періоді і тоді ми просто робимо різницю між ними і отримаємо довжину імпульсу.

					ДК81.460839.001 ПЗ	Арк
Зм.	Арк	№ док.	Підпис	Дата		19

$$Pulse_{width} = CurrentCaptured - LastCaptured$$

А якщо current\_captured менший за last\_captured, тобто у current\_captured було записано значення зафіксованого заднього фронту на наступному періоді таймеру, тоді ми беремо максимальне значення таймеру, яке було записано у регістр ARR, і від нього віднімаємо last\_captured а потім додаємо current\_captured, і ми отримаємо коректне значення довжини імпульсу.

$$Pulse_{width} = 0xffff - lastCaptured + CurrentCaptured$$

Наприклад передній фронт прийшов коли значення лічильнику дорівнювало 0xffffe, а задній фронт прийшов коли значення лічильнику вже переповнилось і стало дорівнювати 5. Тоді згідно нашої формули будемо мати наступне:

$$Pulse_{width} = 0xffff - fffe + 5 = 6 \text{ мкс}$$

Після обрахунку довжини, зберігаємо current\_captured у змінній last\_captured. Далі робимо перетворення до сантиметрів:

$$Distance(см) = Pulse_{width} \cdot \frac{0.034}{2}$$

Тобто для того щоб перевести у відстань, нам потрібно виміряну довжину сигналу(у мкс) помножити на швидкість звуку, яка дорівнює 340 м/с. З урахування того, що довжина імпульсу вимірюється в мкс і результат ми хочемо отримати в см, нам потрібно швидкість звуку перевести з м/с у см/мкс, тобто 340 м/с = 0.034 см/мкс. Також треба урахувати, що імпульс повинен досягти перешкоди і потім повернутися назад, тому потрібно результат поділити на 2.

Тепер у змінній Distance знаходиться коректне значення відстані до перешкоди.

### 3.3.5 Опис роботи функції angle()

Функція angle, як вже було сказано раніше перетворює значення змінної з одного діапазону в інший. В нашому випадку вона перетворює вхідну змінну Distance з далекоміра у діапазон з 500 до 2500. Формула для перетворення наступна:

$$VALUE_{RETURN} = \frac{(value - in_{MIN}) \cdot (Out_{MAX} - Out_{MIN})}{(In_{MAX} - In_{MIN})} + Out_{MIN}$$

$$= (value - 1) \cdot \frac{(2500 - 500)}{(400 - 1)} + 500$$

### 3.3.6 Опис роботи безкінечного циклу:

У безкінечному циклі передається значення Distance у функцію angle, яка повертає значення в діапазоні від 500 до 2500. Далі це значення записуємо у регістр CCR3 таймеру 4, в результаті чого змінюється коефіцієнт заповнення ШІМ, і серводвигун зміщується на відповідний кут.



## ВИСНОВОК

В ході виконання лабораторної роботи було створено пристрій повертання серводвигуна на заданий кут в залежності від відстані ультразвукового далекоміра до перешкоди.

Було досліджено серводвигун MG995, який обертається на заданий кут за допомогою ШІМ сигналу. Також було досліджено ультразвуковий далекомір HC-SR04.

Розроблено схему електричну принципову для створення пристрою.

Під час виконання роботи було створено прототип пристрою з наведеними вище модулями, якими керував мікроконтролер STM32F401RE.

У ході роботи було досліджено роботу з мікроконтролером STM32F401RE NUCLEO та набуто навички з використання регістрів, таймерів, їх режимів та переривань .

Розроблено програмне забезпечення у середовищі ARM Keil та описано принцип роботи програми.

					ДК81.460839.001 ПЗ	Арк
Зм.	Арк	№ док-м.	Підпис	Дата		22

## ЛІТЕРАТУРА

1. 1.STM32F401RE Nucleo User Manual (UM1724). Режим доступу:  
[https://www.st.com/content/ccc/resource/technical/document/user\\_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf](https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf)
2. 2.STM32F401RE Nucleo Programming Manual (PM0214). Режим доступу:  
[www.st.com/resource/en/programming\\_manual/dm00046982.pdf](http://www.st.com/resource/en/programming_manual/dm00046982.pdf)
3. Reference\_manual\_STMF401(RM0368). Режим доступу:  
[https://www.st.com/content/ccc/resource/technical/document/reference\\_manual/5d/b1/ef/b2/a1/66/40/80/DM00096844.pdf/files/DM00096844.pdf/jcr:content/translations/en.DM00096844.pdf](https://www.st.com/content/ccc/resource/technical/document/reference_manual/5d/b1/ef/b2/a1/66/40/80/DM00096844.pdf/files/DM00096844.pdf/jcr:content/translations/en.DM00096844.pdf)
4. MG995: <https://components101.com/motors/mg995-servo-motor>
5. HC-SR04: <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>