

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему:

*Система керування роботом-пилососом*

Виконав(ла): студент(ка) IV курсу, групи СІс-43  
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

	<u>Кардаш І.А.</u> (підпис)	<u>Кардаш І.А.</u> (прізвище та ініціали)
Керівник	<u>Осухівська Г.М.</u> (підпис)	<u>Осухівська Г.М.</u> (прізвище та ініціали)
Нормоконтроль	<u>Луцик Н.С.</u> (підпис)	<u>Луцик Н.С.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Осухівська Г.М.</u> (підпис)	<u>Осухівська Г.М.</u> (прізвище та ініціали)
Рецензент	<u>Дуда О.М.</u> (підпис)	<u>Дуда О.М.</u> (прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студенту \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи Система керування роботом-пилососом

Керівник роботи кандидат технічних наук, зав. кафедри КС Осухівська Галина Михайлівна  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 23 » 03 2022 року № 4/7-180

2. Термін подання студентом завершеної роботи 22.06.2022

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз ТЗ, Огляд вимог, Аналіз можливих рішень, Розробка узагальненої структури, Вибір апаратної складової, Вибір програмних компонентів, Створення алгоритму роботи, Реалізація проектних рішень, Збірка тестового макету, Тестування

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Структурна схема пристрою

Схема електрично принципова

Блок-схема алгоритму роботи

Загальний вигляд системи

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4 Безпека життєдіяльності, основи охорони праці	Лазарюк В.В. к.т.н., доцент		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з завданням до кваліфікаційної роботи	23.03-14.03	Виконано
2	Розробка технічного завдання	14.03-17.03	Виконано
3	Робота над розділом «Аналіз технічного завдання»	17.03-21.03	Виконано
4	Розробка узагальненої структурної схеми	21.03-23.03	Виконано
5	Визначення необхідних апаратних складових	23.03-29.03	Виконано
6	Проектування електрично-принципової схеми	29.03-04.04	Виконано
7	Робота над розділом «Проектна частина»	04.04-19.04	Виконано
8	Розробка алгоритму та написання коду роботи пристрою	19.04-21.04	Виконано
9	Тестування пристрою	21.04-28.04	Виконано
10	Робота над розділом «Практична частина»	28.04-01.05	Виконано
11	Виконання завдань до розділу «Безпека життєдіяльності, основи охорони праці»	01.05-09.05	Виконано
12	Оформлення пояснювальної записки	09.05-19.05	Виконано
13	Нормоконтроль	16.05-22.05	Виконано
14	Перевірка на плагіат	15.06	Виконано
15	Захист КБ	22.06	Виконано

Студент

\_\_\_\_\_ (підпис)

Кардаш І.А.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Осухівська Г.М.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Система керування роботом-пилососом // Кваліфікаційна робота // Кардаш Іван Анатолійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІс-43 // Тернопіль, 2022 // с. – 70 , рис. – 36 , табл. – 1 , додат. – 3, бібліогр. – 18.

Ключові слова: ПРИСТРІЙ, МІКРОКОНТРОЛЛЕР, АЛГОРИТМ, ПРОШИВКА, РОБОТ-ПИЛОСОС, СХЕМА, СИСТЕМА.

Об'єкт проектування – Система керування роботом-пилососом.

Кваліфікаційна робота складається з чотирьох розділів.

У першому розділі проводиться аналіз технічного завдання.

У другому розділі виконується проектування апаратної та програмної частин проекту.

У третьому розділі описується практична розробка проекту, а саме, коду прошивки та фізичної моделі, а також тестування пристрою.

Четвертий розділ це розділ безпеки життєдіяльності та основ охорони праці.

## ABSTRACT

Vacuum cleaner robot control system // Qualification work // Kardash Ivan Anatoliyovych // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Networks, SIS-43 Group // Ternopil , 2022 // p. - 70, fig. - 36, table. - 1, append. - 3, bibliogr. - 18.

Keywords: DEVICE, MICROCONTROLLER, ALGORITHM, FIRMWARE, ROBOT VACUUM CLEANER, SCHEME, SYSTEM.

Design object - Vacuum cleaner robot control system.

Qualification work consists of four sections.

The first section analyzes the terms of reference.

In the second section, the hardware and software parts of the project are designed.

The third section describes the practical development of the project, namely, firmware code and physical model, as well as device testing.

The fourth section is the section on life safety and basics of labor protection.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ .....	9
1.1 Аналіз принципів роботи робота-пилососа .....	9
1.2 Вимоги до системи .....	12
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА .....	14
2.1 Розробка узагальненої структурної схеми .....	14
2.2 Обґрунтування апаратних складових .....	15
2.3 Розробка електричної принципової схеми .....	19
2.4 Обґрунтування програмного забезпечення для створення проекту .....	22
2.5 Вибір IDE для написання коду роботи та прошивки системи .....	24
2.6 Аналіз алгоритму роботи аналогів .....	28
РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА .....	30
3.1 Створення проекту та стартове налаштування мікроконтролера .....	30
3.2 Розробка алгоритму та написання коду роботи пристрою .....	35
3.3 Тестування пристрою .....	44
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	50
4.1 Заходи безпеки при експлуатації електроустановок .....	50
4.2 Соціальне значення охорони праці .....	54
ВИСНОВКИ .....	56
ПЕРЕЛІК ПОСИЛАНЬ .....	57
ДОДАТОК А .....	59
ДОДАТОК Б .....	64
ДОДАТОК В .....	65

					КС КРБ 123.216.00.00 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Система керування роботом-пилососом					
Розробив		Кардаш І.А.						Лім.	Арк.	Аркушів
Перевірів		Осухівська Г.М.						6	70	
Рецензент		Дуда О.М.						ТНТУ, каф. КС, гр. СІс-43		
Н. Контр.		Луцик Н.С.								
Затвердив		Осухівська Г.М.								

## ВСТУП

Дуже часто домашня робота доволі рутинна, зокрема прибирання. Не часто можна побачити людину, яка з ентузіазмом буде прибирати. І далеко не завжди в цьому винна лінь, часто ми не маємо часу чи можливості вчасно прибрати чи не хочемо турбувати рідних зайвим шумом, а підлога, з часом не стає чистішою. А грязна підлога, в приміщеннях де ми живемо, дихаємо, займаємось повсякденними справами, може дуже негативно вплинути на здоров'я. На щастя зараз ми живемо в час коли ми маємо можливість автоматизувати рутинні справи, такі як прання (пральна машина), миття посуду (посудомийка), а з недавнього часу - навіть прибирання, з допомогою, так званих, роботів-пилососів. Робот-пилосос - це пристрій, який оснащений слабким штучним інтелектом, тобто штучним інтелектом призначеним для конкретної сфери. Призначений робот-пилосос для автоматичного прибирання приміщень. Цей пристрій належить до класу побутових роботів та інтелектуальної побутової техніки для smart house (розумного дому). Робот пилосос працює автоматично, за алгоритмом прописаним виробником, або на основі даних отриманих від датчиків.

Метою кваліфікаційної роботи є розробка системи керування роботом-пилососом на основі мікроконтролера STM32F103C6T, який матиме базовий алгоритм руху на двох електродвигунах, опираючись на дані, отримані від сонара.

Для цього потрібно:

- ознайомитися, з принципом роботи робота-пилососа;
- розробити електрично-принципову схему;
- визначити інструменти розробки;
- розробити алгоритм роботи;
- написати програму для пристрою;

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						7
Змн.	Арк	№ документа	Підпис	Дата		

– протестувати роботу пристрою.

В результаті очікується отримання тестового прототипу пристрою, який буде «мозком» робота-пилососа та виконуватиме базові елементи його роботи. Пристрій виконуватиме керування роботою двигунами, що відповідатимуть за рух опираючись на показання сонара (пристрою, який вимірює відстань до перешкод).

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						8
Змн.	Арк	№ документа	Підпис	Дата		



## РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Аналіз принципів роботи робота-пилососа

В першу чергу при розробці системи керування роботом-пилососом, необхідно визначити, що таке робот-пилосос та принцип його роботи.

Робот-пилосос виконує прибирання приміщення або приміщень згідно визначеного сценарію або за командою користувача. Під час роботи він рухається самостійно по визначеній поверхні та прибирає з неї сміття. Наткнувшись на перешкоду, робот приймає рішення про спосіб її подолання за спеціальними алгоритмами або правилами, а також сучасні роботи-пилососи можуть самостійно повернутися до зарядного пристрою, по завершенню роботи або при низькому заряді акумулятора. Саме для реалізації таких завдань необхідна система керування роботом-пилососом (СК РП).

Робот-пилосос це пристрій, який оснащений слабким штучним інтелектом, призначений він для автоматичного прибирання приміщень. Цей пристрій належить до класу побутових роботів та інтелектуальної побутової. Робот пилосос виконує прибирання приміщення або приміщень згідно визначеного сценарію або за командою користувача. Під час роботи він рухається самостійно по визначеній поверхні та прибирає з неї сміття. Наткнувшись на перешкоду, робот приймає рішення про спосіб її подолання, за спеціальними алгоритмами або правилами. Сучасні роботи-пилососи можуть самостійно повернутися до зарядного пристрою, по завершенню роботи або при низькому заряді акумулятора.

					<b>КС КРБ 123.216.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Кардаш І.А.			<b>Аналіз технічного завдання</b>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірів</i>		Осухівська Г.М.					9	70
<i>Рецензент</i>		Дуда О.М.				<b>ТНТУ, каф. КС, гр. СІс-43</b>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затвердив</i>		Осухівська Г.М.						

Для роботи використовуються внутрішні акумулятори, тому є необхідність регулярної підзарядки. Сучасні моделі самі можуть знайти станцію підзарядки. Коли робот-пилосос застрягає він зупиняється і починає подавати звукові сигнали, щоб було простіше його знайти власнику [4].

Сучасні роботи-пилососи орієнтуються в просторі за допомогою наступних пристроїв та технологій:

- дальноміри (ультразвукові);
- контактні датчики – знаходяться на бампері і дозволяють фіксувати контакт зі стінами або меблями;
- «віртуальна стіна», яка дозволяє позначити територію на якій буде проводитись прибирання;
- інфрачервоні сенсори розміщені знизу і не дають пристрою з'їхати із сходів;

Сучасні приклади роботів-пилососів зображено на рисунках 1.1, 1.2, 1.3:

- Xiaomi модель Xiaowa Robot Vacuum Cleaner Lite C102-00



Рисунок 1.1 – Робот-пилосос від Xiaomi

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						10
Змн.	Арк	№ документа	Підпис	Дата		

Простий бюджетний варіант, від відомої китайської фірми Хіаомі. Такий пилосос оснащений інфрачервоними датчиками і рухається хаотично. Заданими, отриманими із сенсора може справно обходити перешкоди та уникати падінь зі сходів.

– iLifeV55Pro (рис.1.2)



Рисунок 1.2 – Робот-пилосос iLifeV55Pro

Робот-пилосос iLifeV55Pro оснащений інфрачервоним датчиком, проте відрізняється від попереднього тим, що має три режими руху: контурний (рух вздовж стін), зигзаги (режим, в якому має працювати даний проект), спіраль. Для роботи ідеально підходять два останні режими. Також є можливість налаштувати прибирання за розкладом.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						11
Змн.	Арк	№ документа	Підпис	Дата		

- Philips SmartProEasy (рис.1.3):

**PHILIPS**



Рисунок 1.3 – Робот-пилосос Philips SmartProEasy

Philips SmartProEasy - це вже більш ефективний, в плані прибирання, варіант. Квадратна форма дає можливість якісніше прибирати в кутках, а також передбачена має можливість вологого прибирання.

Технічно він не відрізняється від попереднього варіанту, алгоритми руху ті самі і датчики - також інфрачервоні.

Проаналізувавши відомі моделі встановлено, що система керування має важливе значення для роботи роботів-пилососів.

## 1.2 Вимоги до системи

Відповідно до технічного завдання та аналізу принципів роботи сучасних роботів-пилососів, були виділені наступні вимоги до системи керування:

- робота без участі користувача;
- керування електродвигунами, для руху вперед, назад та розворотів;
- розпізнати перешкоду, її оминати або виконати розворот.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						12
Змн.	Арк	№ документа	Підпис	Дата		

Відповідно технічному завданню СК РП повинна визначати відстань до перешкоди в діапазоні від 1 до 10см з точністю  $\pm 5$ мм, для коректного переміщення по приміщенню. Пристрій, керований даною СК РП, повинен рухатися з максимальною швидкістю, яка не перевищуватиме 3м/хв., при цьому час розвороту на  $90^{\circ}$ , складатиме приблизно 2с. Пристрій працюватиме від 19-вольтової батареї. Сам пристрій керування повинен бути доволі компактним, 80мм -довжина, 100мм –ширина, висота-40мм.

Пристрій повинен працюватиме за температури від  $-5^{\circ}\text{C}$  до  $+40^{\circ}\text{C}$ , при вологості до 100% при  $t=25^{\circ}\text{C}$ , що дозволяє працювати навіть в умовах морозів чи спеки. Робота виконуватиметься на рівній поверхні. Собівартість пристрою не повинна перевищувати 3000гривень. Надійність пристрою повинна бути на високому рівні, мати можливість напрацювати не менше 5000 годин.

Для самостійної роботи системи (без участі користувача), необхідно використати надійний мікроконтролер, на який буде завантажено прошивку з алгоритмом роботи. Для керування двигунами доцільно використати набір транзисторів з метою підключення їх до джерела живлення і правильного перемикання через мікроконтролер. Для можливості розпізнати перешкоду - краще використовувати сонар (звуковий дальномір), оскільки він може достатньо точно визначити відстань до перешкоди і при цьому не є дорогим, що теж є перевагою.

Режим роботи був обраний «зигзаги», аналогічно як у моделі iLifeV55Pro, тобто робот-пилосос буде рухатися зигзагами. Тобто, починає рух, рухається вперед до найближчої перешкоди, виконує поворот направо чи на ліво, проїжджає деяку відстань виконує ще один поворот і рухається паралельно до попередньої траєкторії. Таким чином досягається поступове прибирання всієї площі без пропусків.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						13
Змн.	Арк	№ документа	Підпис	Дата		

## РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

### 2.1 Розробка узагальненої структурної схеми

В першу чергу було розроблено структурну схему СК РП, зображену на рисунку 2.1.

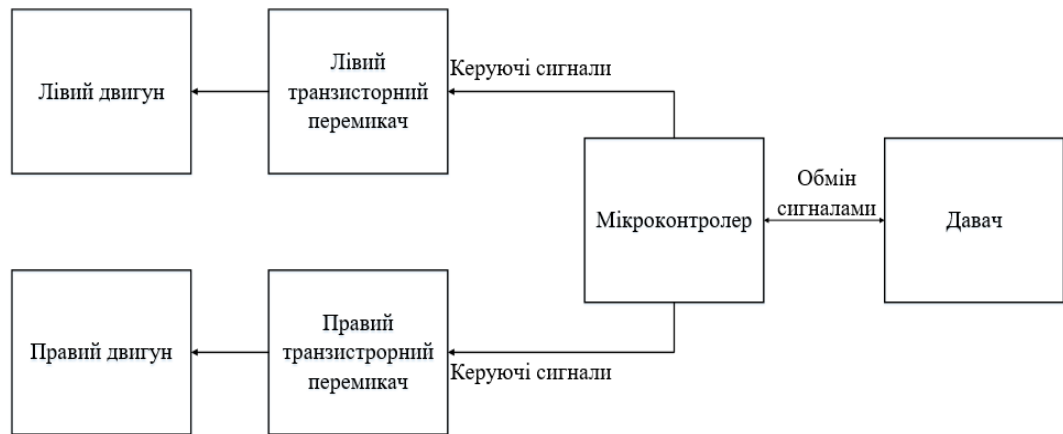


Рисунок 2.1 – Узагальнена структурна схема СК РП

Відповідно до структурної схеми зображеної вище (рис.2.1), за керування роботою робота-пилососа відповідатиме мікроконтролер. Живлення контролера здійснюється від акумулятора 19В, через понижувач струму, що знижуватиме напругу до 3,3В, оскільки контролер працює саме при такій напрузі.

Як давач - використано сонар (ультразвуковий дальномір), він, за сигналом від контролера, надсилатиме ультразвуковий сигнал і коли цей сигнал відбиватиметься від перешкоди і повертатиметься назад, відправить сигнал на мікроконтролер. В цей час контролер зафіксує час між відправкою і отриманням сигналу і розрахує відстань до перешкоди.

<b>КС КРБ 123.216.00.00 ПЗ</b>				
Змн.	Арк.	№ докум.	Підпис	Дата
Розробив		Кардаш І.А.		
Перевірів		Осухівська Г.М.		
Рецензент		Дуда О.М.		
Н. Контр.		Луцик Н.С.		
Затвердив		Осухівська Г.М.		
<b>Проектна частина</b>				
			Літ.	Арк.
			14	70
<b>ТНТУ, каф. КС, гр. СІс-43</b>				

В залежності від даних, отриманих від давача, мікроконтролер керуватиме двома двигунами, які відповідатимуть за рух робота-пилососа.

## 2.2 Обґрунтування апаратних складових

Враховавши вимоги до системи, її функціонал, а також розроблену структуру, як керуючий блок обрано мікроконтролер STM32F103C6T. Такий варіант ідеально підходить для проекту такого типу, оскільки це простий мікроконтролер, без зайвих функцій, є дуже енергоефективним. Мікроконтролери STM зарекомендували себе, як дуже надійні, а тому і популярні.

Мікроконтролер - це мікросхема, яка поєднує в собі функціонал як периферійних пристроїв так і процесора, містить оперативну або постійну пам'ять, а в деяких моделях обидва типи пам'яті. В загальному - це мінікомп'ютер, призначений для виконання простих задач в конкретній сфері.

Мікроконтролер STM32F103C8T6 - це високопродуктивне ядро процесора з розрядністю 32-біти, частота роботи якого 72МГц. Це один з найпростіших та найдешевших пристроїв компанії STM, при цьому він достатньо потужний, щоб виконувати своє завдання. Таким чином можна зменшити собівартість приладу, при цьому не втративши його продуктивності. Цей мікроконтролер, розміщений на платі, яка має всю необхідну периферію та характеристики описані в таблиці 1.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						15
Змн.	Арк	№ документа	Підпис	Дата		

Таблиця 1 – Характеристики мікроконтролера STM32F103C8T6

тип корпусу:	LQFN48
ядро:	ARM Cortex-M3
тактова частота:	72 МГц
флеш пам'ять:	64 кбайт
SRAM:	20 кбайт
напруга на цифрових входах / виходах:	2 - 3,6 В
напруга на аналогових виходах:	3,6 В
інтерфейси налагодження:	SWD, JTAG
DMA	7-канальний DMA контролер
цифро-аналоговий перетворювач:	12 біт (16 каналів)
кількість лічильників	7
ШІМ	16 біт (6 каналів)

Для кращого розуміння роботи з мікроконтролером та його використання в проєкті, на рисунку 2.2 розміщено детальний опис виводів мікроконтролера [5].

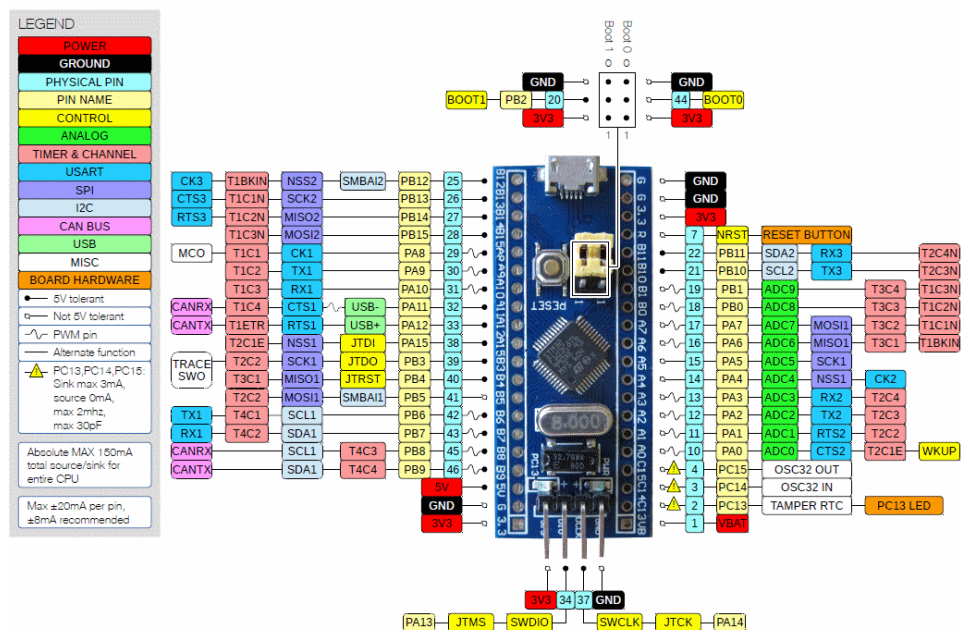


Рисунок 2.2 – Виводи мікроконтролера [5]



Структурна схема мікроконтролера STM32F103C8T6 дає детальне розуміння принципу його роботи і розміщена на рисунку 2.3 [5].

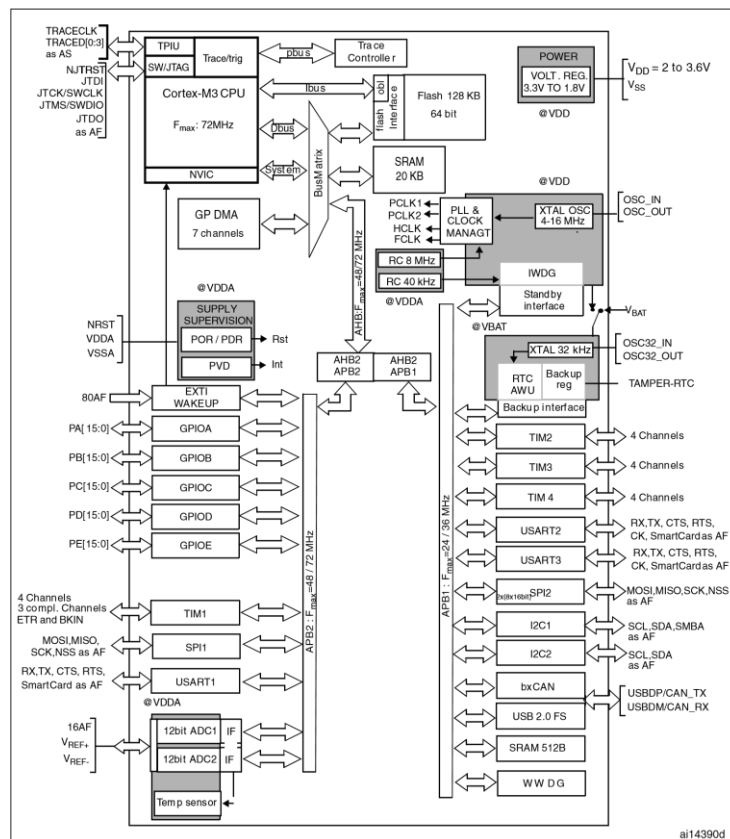


Рисунок 2.3 – Блок-схема структури мікроконтролера [5]

Для визначення перешкод та відстані до них, використовуватиметься сонар HC-SR04 [6].

Ультразвуковий далекомір HC-SR04 являє собою плату, на якій розміщені приймач та передавач ультразвукового сигналу. Принцип дії пристрою доволі простий. Випромінювач подає ультразвуковий сигнал, який прямує до перешкоди, відбивається та повертається до приймача. Мікроконтролер може зафіксувати час, за який цей сигнал дійшов до перешкоди і назад, а також вирахувати відстань до об'єкта [6]. Ультразвуковий давач HC-SR04 позбавлений недоліків інфрачервоних давачів, тобто перешкоди у вигляді джерел світла або кольору перешкоди не вплинуть на якість роботи.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		17

Звісно є недоліки: можуть виникнути труднощі щодо відстані до пухнастих чи тонких об'єктів. Також на платі є вся необхідна обв'язка і 4 виводи:

VCC – живлення +5В;

Trig(T) – виведення вхідного сигналу;

Echo(R) – виведення вихідного сигналу;

GND – земля.

Для руху використовуватимуться два електродвигуни, але оскільки цей проєкт являє собою тільки систему керування то тип та потужність двигунів визначатися не буде, оскільки ці параметри визначаються в залежності від технічних характеристик самого робота-пилососа, втім вже відомо що напруга, з якою працюватимуть двигуни буде в межах 5В. Мікроконтролер STM32F103C6T не призначений для роботи з електродвигуном, а тому він не може бути джерелом живлення. Тому для роботи двигуна потрібно розробити транзисторну схему, яка дасть можливість живлення двигунів від батареї та керування напрямком їх руху з мікроконтролера [1].

Транзистори, які використовуватимуться для цієї схеми мають маркування S9013. Транзистор S9013 – біполярний кремнієвий, структури NPN, у пластмасовому корпусі ТО-92 з трьома виходами. Максимальна сила струму 0,5А і напруга - до 40В, чого достатньо для простих електродвигунів. Схема підключення двигуна з можливістю керування напрямком руху зображена на рисунку 2.4. Згідно схеми (рис.2.4), для керування одним двигуном потрібно 4 транзистори, а для двох двигунів – 8 [1].

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						18
Змн.	Арк	№ документа	Підпис	Дата		

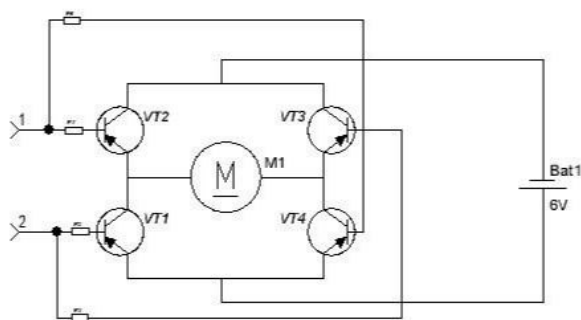


Рисунок 2.4 – Схема підключення двигуна з можливістю керування напрямком руху [1]

Також використано резистори на 270Ом, для керуючого сигналу від контролера до транзисторів, щоб не перевантажувати його.

Такого набору елементів достатньо для розробки проєкту, тому можна починати проєктування схеми.

### 2.3 Розробка електрично-принципової схеми

Спочатку було вирішено спроектувати підключення двигуна до джерела живлення (в даному випадку це батарея на 19В). Підключення буде виконано відповідно схеми, зображеної на рисунку 2.4.

Згідно схеми (рис.2.4), було розроблено схему підключення двигунів з двома пінами керування та можливістю керування напрямком руху. Схема підключення розміщена на рисунку 2.5.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						19
Змн.	Арк	№ документа	Підпис	Дата		

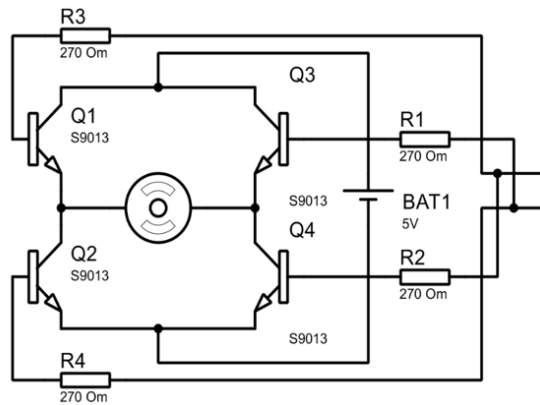


Рисунок 2.5 – Схема підключення двигуна з можливістю керування напрямком руху

Така схема підключення (див.рис.2.5) дає можливість керувати напрямком руху двигуна, всього двома лініями команд, при цьому, не навантажуючи мікроконтролер, адже живлення двигуна виконується напряму через джерело живлення. Оскільки таких двигунів два, то і таких схем буде дві, а це 4 лінії керування, підключені вони будуть до виводів А1-А4 мікроконтролера.

Для роботи з сонаром використовуватиметься таймер TIM1, який, згідно структурної схеми мікроконтролера, виходить на пін А8, тому вихід сонару Echo буде підключено до цього піна. Trig буде підключено до піна А9 (рис. 2.5). Виходи живлення також будуть підключенні до батареї 19В.

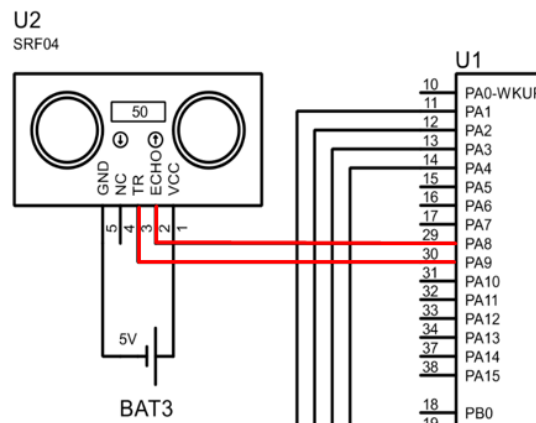


Рисунок 2.6 – Схема підключення сонара SRF04

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		20

Мікроконтролер живиться від напруги 3,3В, тому також необхідний перетворювач струму, який буде знижувати напругу постійного струму до 3,3В. Для цього використано мікросхему MC34063, оскільки ця схема достатньо надійна і при цьому дешева. Працює вона з напругами від 3В до 40В. Підключена вона за схемою, зображеною на рисунку 2.7. В результаті було отримано понижувач напруги до 3,3В, з максимальною силою струму 1,5А [3].

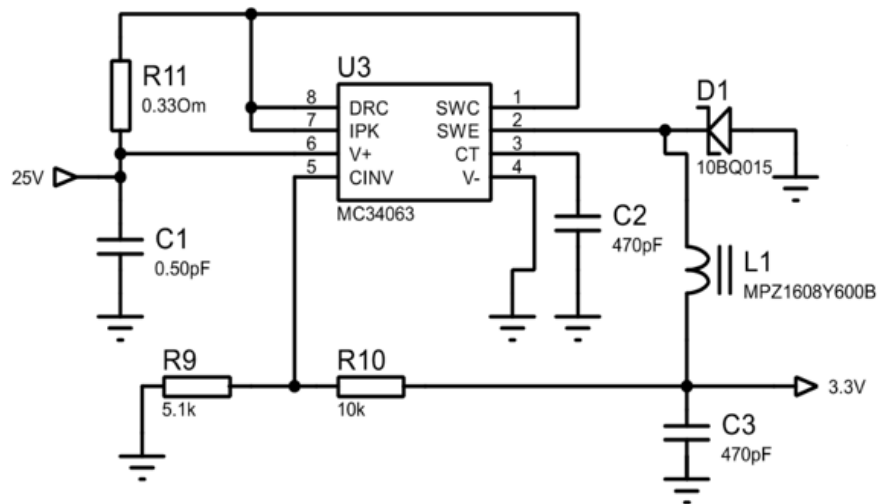


Рисунок 2.7 – Понижувач напруги [3]

Далі понижувач напруги був підключений до лінії живлення контролера. В результаті було отримано схему, що зображено на рисунку 2.8.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
						21
Змн.	Арк	№ документа	Підпис	Дата		

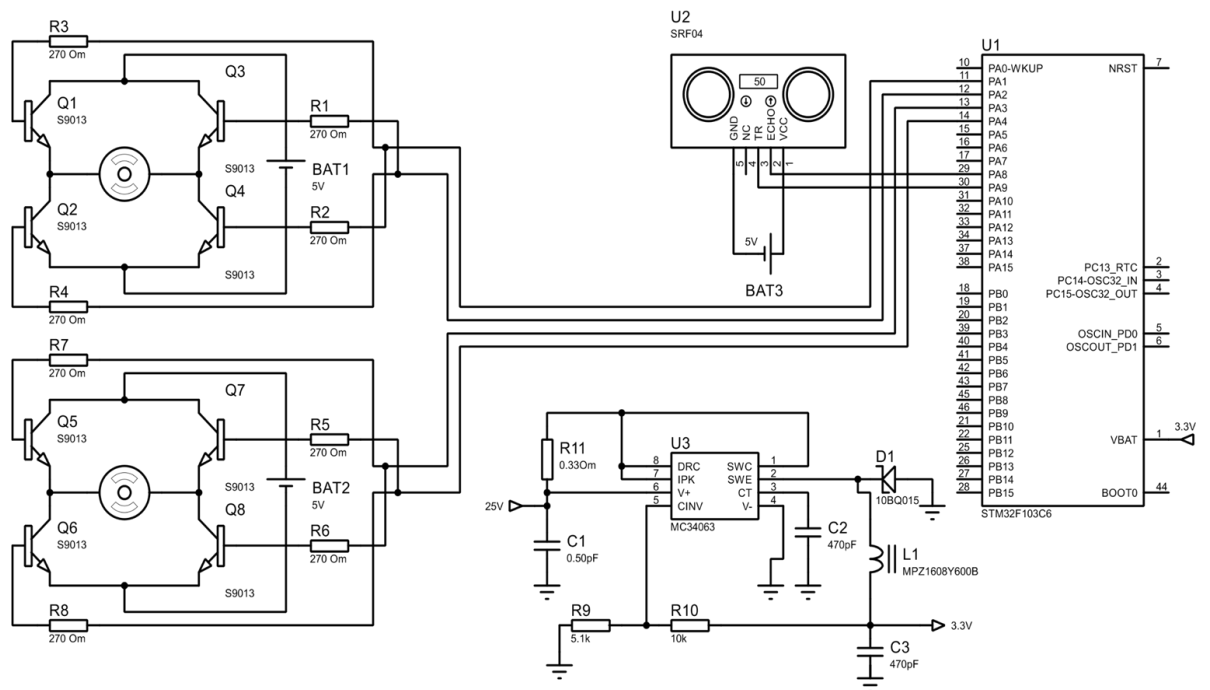


Рисунок 2.8 – Електрична принципова схема СК РП

## 2.4 Обґрунтування програмного забезпечення для створення проєкту

На даний час є багато способів створення проєктів для прошивки мікроконтролерів.

Для створення проєкту використовують 3 основні способи:

- створення проєкту за допомогою базового проєкту Keil з наступним ручним підключенням необхідних пакетів та налаштуванням апаратних складових через код самостійно;
- створення проєкту та підключення бібліотек в CoCoXIDE – спеціальній програмі для роботи з мікроконтролерами, яка генерує папки та самостійно підключає бібліотеки, обрані користувачем в графічному інтерфейсі;
- генерація проєкту за допомогою спеціальної програми STM32CubeMX. Такий варіант є найзручнішим так як він дає можливість створити проєкт і тонко налаштувати всі пристрої в мікроконтролері в зручному

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		22

графічному інтерфейсі, після чого генерує готовий код цих налаштувань, що надзвичайно спрощує та пришвидшує розробку.

В загальному, відразу зрозуміло, який варіант є найбільш зручним. Створення проекту вручну це доволі муторний процес, який потребує невиправданих затрат часу, а всі бібліотеки потрібно завантажувати і підключати самостійно, тому такий підхід зовсім не розглядався.

Генерація проекту в Coocox IDE, це вже зручніший та швидший варіант, проте ручне налаштування тих пристроїв, з якими буде вестись робота - це також додаткові затрати часу.

Як вже було сказано, найзручніший спосіб генерації проекту - це програма STM32CubeMX.

STM32CubeMX – програмний продукт, що дозволяє за допомогою досить зрозумілого графічного інтерфейсу зробити налаштування будь-якої периферії мікроконтролера.

Працює все таким чином: користувач створює об'єкт, далі обирає потрібний мікроконтролер зі списку і одразу на екран виводиться велика схема з усіма выводами обраного контролера. Інтерфейс програми STM32CubeMx показано на рисунку 2.9. На кожен вивід можна натиснути і в випадяючому меню можна обрати режим роботи виводу. Завдяки зручним меню можна детально налаштувати периферію і також - режими роботи виводів. Переваги такого підходу очевидні, всі налаштування проводяться в декілька кліків мишею, при цьому все видно і не потрібно відкривати документацію контролера для налаштування того чи іншого пристрою.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						23
Змн.	Арк	№ документа	Підпис	Дата		

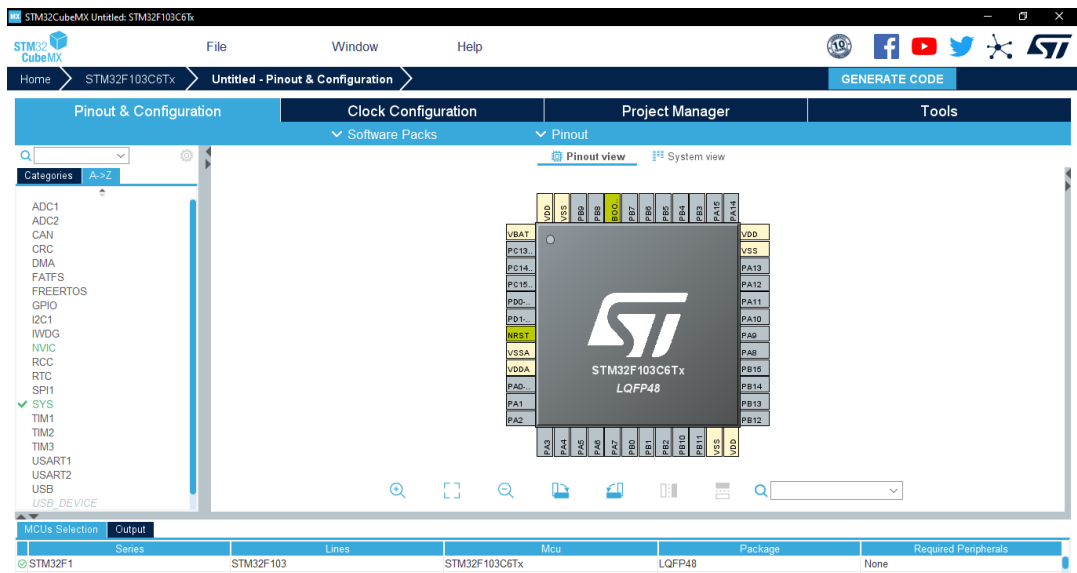


Рисунок 2.9 – Інтерфейс програми STM32CubeMx

Отже, вибравши всі необхідні виводи, вводимо всі налаштування периферії, а саме частоти тактування, режими роботи таймерів і так далі. STM32CubeMx дає можливість згенерувати код, в якому вже все налаштовано. Тобто отримуємо готовий проєкт, з уже прописаними налаштуваннями, з підключеними бібліотеками і з готовою ініціалізацією всієї обраної периферії. Базою для цього є бібліотеки CMSIS і HAL.

## 2.5 Вибір IDE для написання коду роботи та прошивки системи

Писати код для мікроконтролера, можна будь-яким текстовим редактором, хоч в Visual Studio хоч в блокноті, але скомпілювати та завантажити його на контролер вже не вийде. Саме тому для роботи потрібно використовувати спеціалізоване програмне забезпечення, яке має весь потрібний функціонал для спрощення написання коду та його компіляції.

Були проаналізовані такі IDE:

- Coocox IDE;
- Keil uVision5;

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		24



– STM32CubeIDE.

CoIDE – найпростіший та найшвидший у плані встановлення, освоєння та налаштування рішень, що дозволяє навіть початківцям добитися в ньому значних результатів. На старті проекту сильно спрощує роботу майстер, що допоможе виконати необхідні етапи розробки з допомогою підказок. Середовище CooCoxCoIDE дає можливість завантажувати вихідний код програми, редагувати його, проводити компіляцію сторонніми засобами, прошивати контролер прямо з програми та проводити налаштування. [7]

Середовище підтримує не тільки мікроконтролери ST, а також ряд інших сімейств: AtmelHoltek, Freescale, NuvotonNXP, Energy Micro, Texas Instruments. Список підтримуваних контролерів зростає з кожною версією програми. Створення проекту починається з вибору мікросхеми і бібліотек, які будуть використані в проекті. Після цього програма автоматично створює всю структуру проекту та підтягує необхідні бібліотеки.

Вагомий недолік CooCox CoIDE - це відсутність компілятора GCC, який доведеться завантажувати та встановлювати самостійно, після чого у налаштуваннях CoIDE потрібно буде вказати правильний шлях до нього.

Ще один неприємний момент - це те, що шляхи до файлів проекту жорстко прописуються у програмі і при переміщенні папки з проектом, проект не збереться. Тому шляхи доведеться також редагувати вручну у файлах. В програмі також відсутня можливість тонких налаштувань, що теж може викликати деякі незручності.

Keil uVision - це також сучасна програма для роботи з проектами різних ступенів складності. Розробника не навантажують другорядними функціями, оскільки більшість з них прихована. Це сильно спрощує інтерфейс і тим самим спрощує управління та робить його зрозумілішим. Втім, якщо робота ведеться з складнішими проектами та завданнями, можливо використовувати всі можливості модулів.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						25
Змн.	Арк	№ документа	Підпис	Дата		

В Keil uVision є достатньо велика кількість програмних засобів, серед них виділяються такі:

1. Обширна база даних мікроконтролерів, в якій можна знайти детальну інформацію щодо всіх пристроїв. Ця база даних містить конфігураційні дані та посилання на додаткові технічні описи.

2. Менеджер проєктів, що допомагає краще орієнтуватися в файлах завдяки об'єднанню файлів та текстів програмних модулів.

3. Вбудований редактор, який дає можливість використання багатовіконного інтерфейсу, що полегшує роботу з текстом програм. Також є функція виділення синтаксичних елементів.

4. Засоби автоматичної компіляції, асемблювання та компонування. Їх призначення полягає в створенні модуля програми, що виконується. В цей час між файлами створюються зв'язки, які надалі дозволяють обробляти лише ті файли, в яких відбувалися зміни. Також є функція, що дозволяє оптимізувати проєкт, що дозволяє пришвидшити компіляцію.

5. Налаштовувач-симулятор, призначення якого налагодити роботу програми на віртуальному мікропроцесорі. Досить якісно та продуктивно моделюється повна робота ядра, а також периферійного обладнання.

6. Також полегшують роботу додаткові утиліти. Число та набір змінюються в різних версіях.

Виділяють наступні утиліти:

– Source-Browser містить базу даних програмних символів швидкого пошуку;

– Find-in-Files шукає уривок коду у всіх доступних файлах вказаної папки чи проєкту;

– ToolsMenu дозволяє користуватися утилітами сторонніх виробників;

– PCLint виділяє потенційно небезпечні місця;

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						26
Змн.	Арк	№ документа	Підпис	Дата		

– Flash tool програмує FLASH-пам'ять мікроконтролерів.

STM32CubeIDE – це передова платформа розробки C/C++ з периферійною конфігурацією, генерацією коду, компіляцією коду та функціями налагодження для мікроконтролерів і мікропроцесорів STM32. Він заснований на фреймворку Eclipse та ланцюзі інструментів GCC для розробки та GDB для налагодження. Він дозволяє інтегрувати сотні існуючих плагінів, які доповнюють функції Eclipse.

STM32CubeIDE інтегрує функції конфігурації STM32 і створення проектів від STM32CubeMX, щоб запропонувати інструмент «все в одному» та заощадити час на встановлення та розробку. Після вибору порожнього MCU або MPU STM32 або попередньо налаштованого мікроконтролера чи мікропроцесора з вибору плати або вибору прикладу створюється проект і генерується код ініціалізації. У будь-який момент під час розробки користувач може повернутися до ініціалізації та конфігурації периферійних пристроїв або проміжного програмного забезпечення та відновити код ініціалізації без впливу на код користувача.

STM32CubeIDE включає аналізатори збірки та стека, які надають користувачеві корисну інформацію про стан проекту та вимоги до пам'яті.

STM32CubeIDE також включає стандартні та розширені функції налагодження, включаючи перегляди регістрів ядра ЦП, пам'яті та регістрів периферійних пристроїв, а також перегляд змінних, інтерфейс SerialWire Viewer або аналізатор несправностей.

В результаті аналізу середовищ розробки, можна сказати що найбільш пріоритетними є Keil uVision та STM32CubeIDE, оскільки вони мають доволі обширний функціонал та відмінно підходять для роботи з архітектурою ARM. Для реалізації СК РП обрано Keil uVision, оскільки вона має велику кількість пакетів та утиліт, а також достатньо навчального матеріалу.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						27
Змн.	Арк	№ документа	Підпис	Дата		

Для роботи над програмою проєкту використовуватимуться дві програми – STM32CubeMX (для генерації проєкту та стартового коду) та Keil uVision (написання, компіляція та відлагодження коду прошивки пристрою).

## 2.6 Аналіз алгоритму роботи аналогів

Для написання прошивки СК РП необхідно зрозуміти, яким чином він виконує навігацію в просторі, так який алгоритм його руху. Оскільки це тільки система керування, то принцип роботи щіток розглядатись не буде.

Робота роботів-пилососів заснована на інформації від чотирьох типів навігації:

- зовнішні датчики – сприймають дані від навколишнього світу;
- лазерна навігація – визначає відстань до об'єктів;
- відеоспостереження – візуальне сканування об'єктів;
- датчики для прибирання – датчики, які контролюють кількість сміття в сміттевому баку.

Дешеві моделі, через обмежений бюджет, не можуть нести в собі сучасні дорогі сенсори, тому в них встановлені найнеобхідніші.

Зовнішні датчики дають змогу сканувати навколишній простір, щоб уникати явні перешкоди і створювати найкращий шлях для руху.

Види датчиків та їх призначення:

- контактний – для обходу перешкод при зіткненні;
- безконтактний – «бачить» перешкоду на відстані і дозволяє зупинитися ще до зіткнення;
- ультразвуковий далекомір – працює за принципом ехолокації, визначаючи відстань до об'єктів та не дозволяє наблизитись ближче критичних значень.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						28
Змн.	Арк	№ документа	Підпис	Дата		

Зовнішні датчики дають можливість зберегти пристрій в цілості і не дають йому стикнутися з меблями.

Лазерна навігація -найсучасніший спосіб орієнтування в просторі, при ній будується карта приміщення, через спеціальні лазерні далекоміри, які знаходять відстань до стін, меблів і зберігають в пам'яті.

На дорогих роботах-пилососах використовуються відеокамери. Камери потребують потужнішого процесора для обробки отриманих даних і підвищують якість прибирання та точність навігації.

Камера розміщена на найвищій точці робота-пилососа, щоб досягнути найкращого кута огляду. Вона сканує стіни, підлогу, меблі та сприймає необхідну інформацію. Такий тип пилососів рухається по прямій [7].

В роботі було вирішено використовувати зовнішній далекомір (сонар), оскільки даний варіант є найзручнішим, і в використанні, і в економічному плані. При ознайомленні з видами та типами роботів-пилососів було обрано тип руху «зигзаг», як у моделі iLifeV55Pro, тобто робот-пилосос буде рухатися зигзагами. Починає рух, рухається вперед до найближчої перешкоди, виконує поворот направо чи на ліво, проїжджає деяку відстань виконує ще один поворот і рухається паралельно до попередньої траєкторії. Таким чином можна досягнути поступового прибирання всієї площі без пропусків.

Тепер, коли програмне забезпечення обрано, і принцип роботи майбутнього принципу заплановано, можна почати роботу над проєктом.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						29
Змн.	Арк	№ документа	Підпис	Дата		

## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Створення проєкту та стартове налаштування мікроконтролера

Для розробки СК РП використано контролер STM32F103C6T. В першу чергу необхідно створити проєкт, з допомогою програми CubeMX.

Після запуску програми виведеться логотип, після чого з'явиться стартове меню, на якому можна почати створювати новий проєкт або перейти до редагування раніше створеного. В цьому меню необхідно обрати пункт «Access to msu selector», що означає перейти до вибору мікроконтролера [1].

Після цього відкриється меню в якому необхідно обрати мікроконтролер, в нашому випадку - це STM32F103C6T. На рисунку 3.1 зображено процес вибору необхідного мікроконтролера.

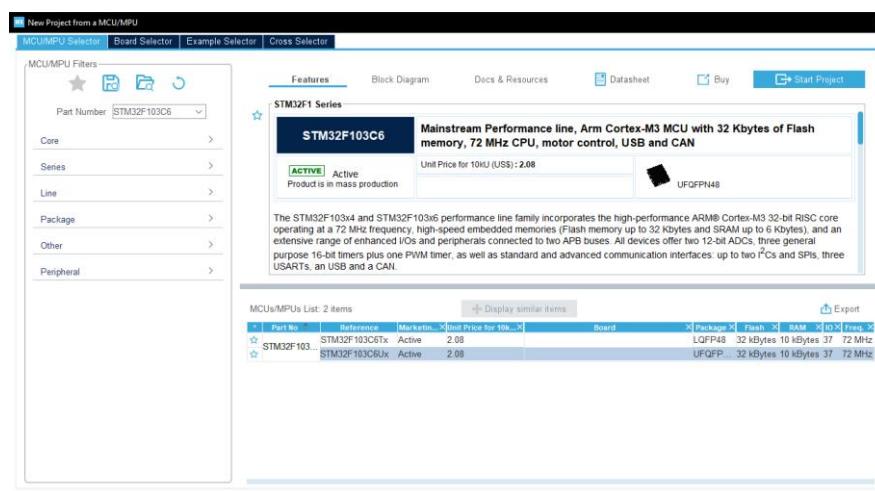


Рисунок 3.1 – Меню вибору мікроконтролера

					<b>КС КРБ 123.216.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	<b>Практична частина</b>	<b>Лім.</b>	<b>Арк.</b>	<b>Аркушів</b>
Розробив		Кардаш І.А.					30	70
Перевірів		Осухівська Г.М.						
Рецензент		Дуда О.М.						
Н. Контр.		Луцик Н.С.						
Затвердив		Осухівська Г.М.						
						<b>ТНТУ, каф. КС, гр. СІс-43</b>		

Коли контролер обрано, виконається завантаження необхідних бібліотек та відкриється головний інтерфейс, що зображено на рисунку 3.2.

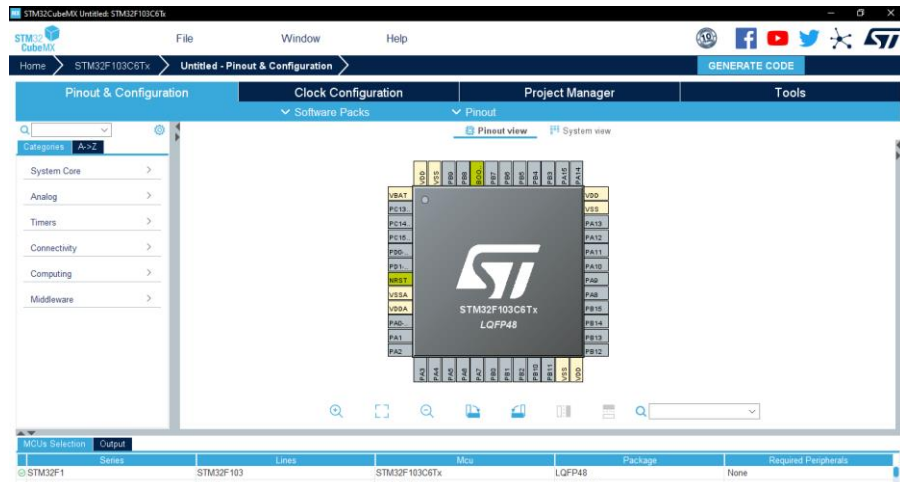


Рисунок 3.2 – Інтерфейс програми STM32CubeMx

Саме в цьому інтерфейсі і виконуватиметься стартове налаштування пристрою. Для початку, щоб можна було працювати з таймером, який буде необхідний для роботи сонара, підключаємо зовнішній керамічний резонатор, який буде виконувати тактування контролера. Порядок підключення відображено на рисунку 3.3 [1].

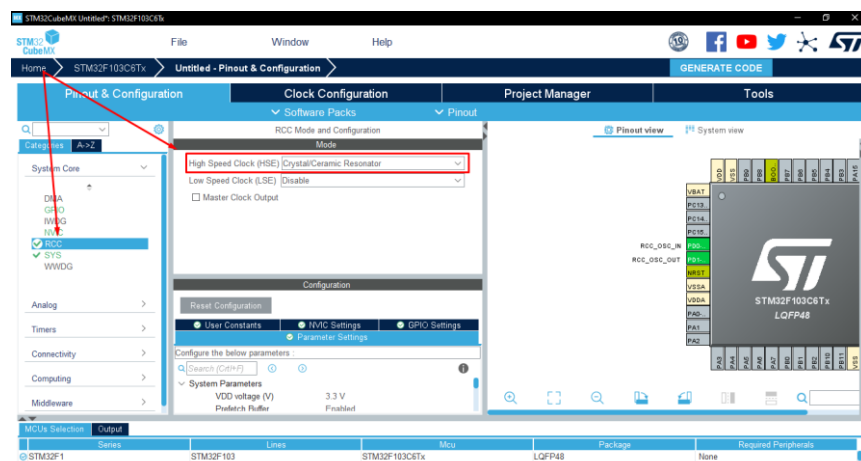


Рисунок 3.3 – Підключення зовнішнього резонатора

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
Змн.	Арк	№ документа	Підпис	Дата		31

Обов'язково необхідно ввімкнути відлагодження через «Serial Wire»(рис. 3.4), для того, щоб можна було виконати відлагодження через ST-Link.

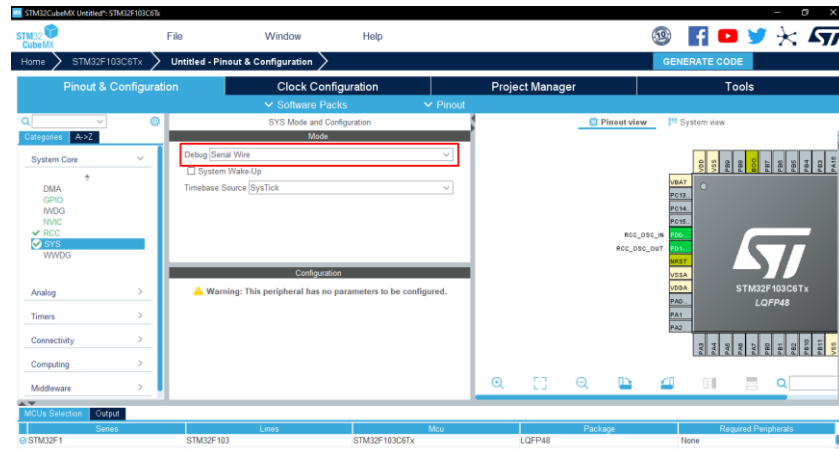


Рисунок 3.4 – Ввімкнення відлагодження

Далі налаштовується тактування основних шин та пристроїв. Для цього необхідно перейти на вкладку Clock Configuration (рис. 3.5) та перемкнути тактування на зовнішній резонатор (HSE) потім ввімкнути режим PLLCLK в системному годиннику і в кінці ввести максимальну частоту, яка передаватиметься на пристрій, в даному випадку - це 72МГц. Таке налаштування дає можливість точного тактування з можливістю безпомилкового підрахунку часу, що в свою чергу дає можливість краще розуміти розмір затримок та взаємодіяти з пристроями.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						32
Змн.	Арк	№ документа	Підпис	Дата		



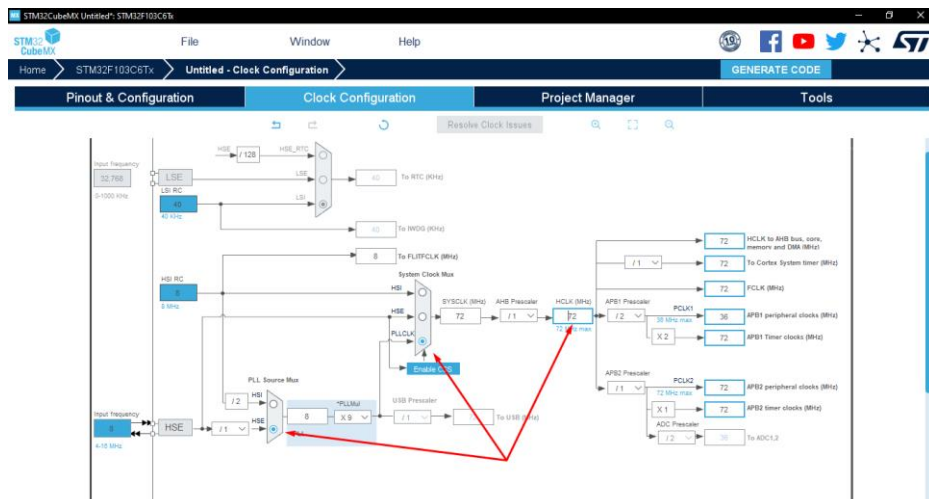


Рисунок 3.5 – Налаштування тактування основних шин та пристроїв

Далі підключаємо ТІМ1 (рис.3.6), активуючи його перший канал в режимі отримання даних. Згідно з схемою мікроконтролера цей канал виходить на пін А8, програма автоматично переключить режим цього піна.

Таймери – це один з найкорисніших пристроїв, які є у мікроконтролерів. З їхньою допомогою можна заміряти час, довжину імпульсів, рахувати імпульси, створювати ШІМ-сигнал, запускати один таймер іншим, запускати таймером деяку периферію, запускати ракети в космос і т.д. А найцікавіше полягає в тому, що все це відбувається на апаратному рівні, тобто без участі CPU.

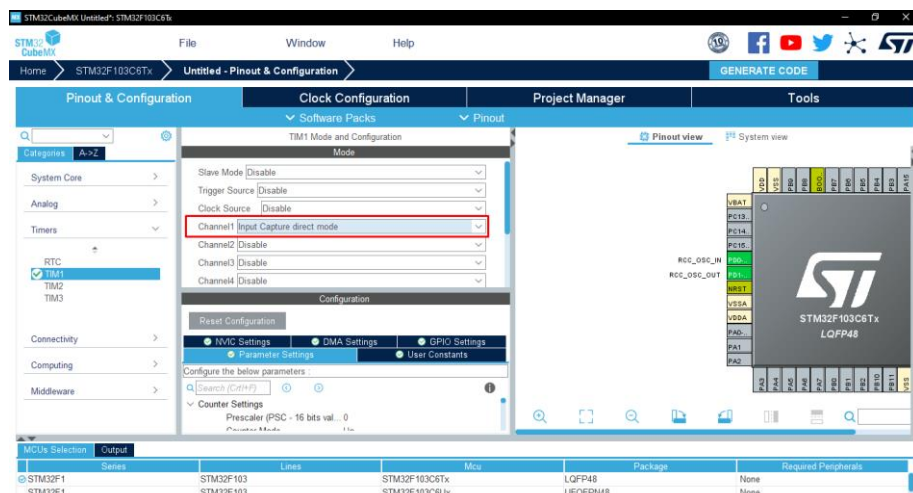


Рисунок 3.6 – Підключення ТІМ1 через перший канал

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		33

Далі були проведені налаштування таймера (рис.3.7):

- Prescaler – це подільник частот, вказуємо значення 72-1, це означає, що на один тик буде проходити 72Гц.
- Counter Mode – режим підрахунку затримки (вказуємо Up, лічильник буде рахувати вгору).
- Counter Period – період підрахунку, до якого рахуватиме лічильник;
- Internal Clock Division – дільник системної частоти таймера для формування допоміжного тактового сигналу, який використовується у цифрових фільтрах (в даному випадку вимкнений).

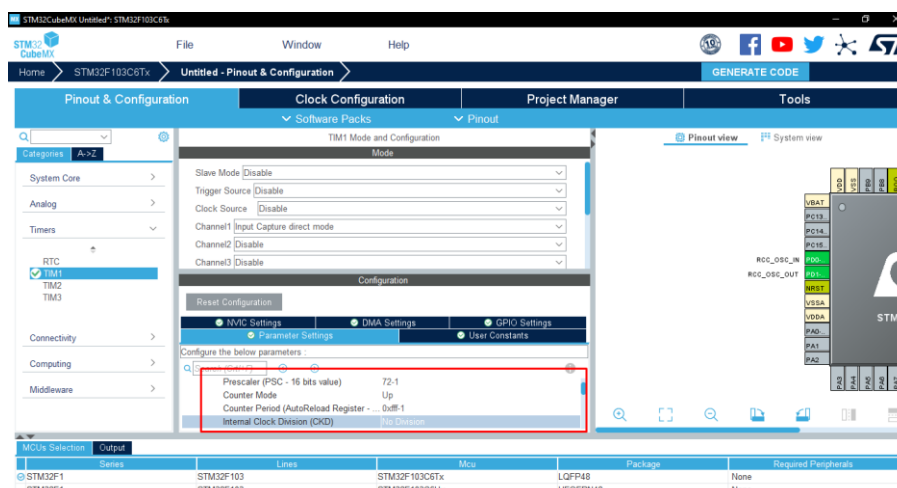


Рисунок 3.7 – Налаштування таймера

Далі проводимо налаштування виводів:

- виводи PA1-PA4 налаштовуємо на вихід (Output), оскільки вони використовуватимуться для керування двигунами;
- вивід PA9 також вихід, він буде підключений до сонару на вихід Trig та подаватиме сигнали, що запускатимуть його;
- виводи PD0 та PD1 підключені до зовнішнього резонатора;
- вивід PA8, це вивід таймера, канал 1.

Схема з налаштованими виводами розміщена на рисунку 3.8.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
Змн.	Арк	№ документа	Підпис	Дата		34

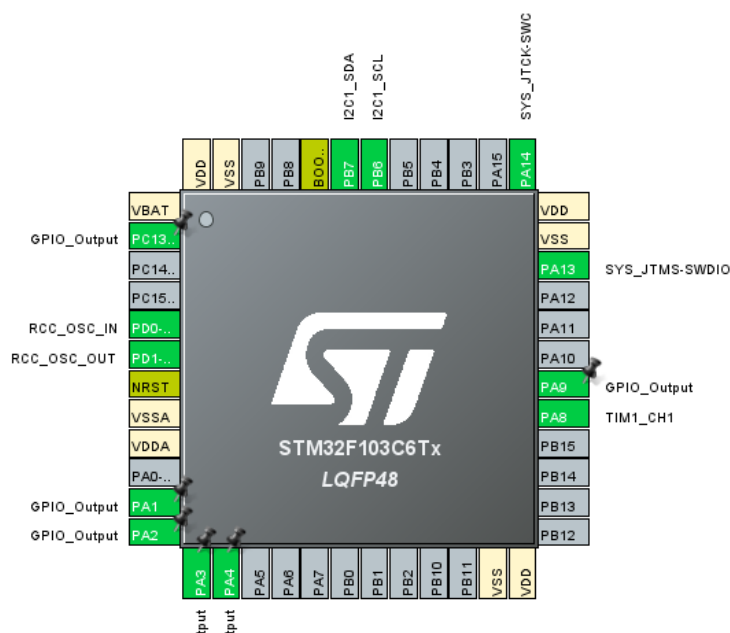


Рисунок 3.8 – Налаштування виводів мікроконтролера

Останнім кроком є стандартні налаштування проєкту (назва, розташування на диску і т.д.). Після цього треба натиснути кнопку «згенерувати проєкт» і проєкт буде згенеровано, як і стартовий код, та збережено до каталогу вказаного користувачем.

### 3.2 Розробка алгоритму та написання коду роботи пристрою

Алгоритм роботи буде відносно простий та складатиметься з двох основних етапів: стартова ініціалізація та основний цикл.

Коротке описання алгоритму роботи:

- 1) Запуск і попередня ініціалізація
  - a) Ініціалізація пристроїв
  - b) Перевірка наявності перешкод навколо пристрою
    - Перевірка наявності перешкоди спереду
    - Поворот наліво
    - Перевірка наявності перешкоди зліва

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						35
Змн.	Арк	№ документа	Підпис	Дата		

- Поворот наліво (цикл повторюється до завершення повороту)
  - c) Якщо перешкода наявна з усіх боків - програма завершується
  - d) Виходячи з попередньої перевірки виконується вибір напрямку  
стартового руху
  - 2) Основний алгоритм
    - a) Перевірка наявності перешкоди
      - Якщо відсутня, виконається рух вперед і перевірка наявності  
виконуватиметься кожні 200мс
      - Якщо перешкода є, виконується розворот, тобто послідовно  
виконується поворот на право, рух вперед 3 секунди, поворот направо
    - b) Затримка 200мс
    - c) Повернення на початок основного алгоритму
- Для більшої наглядності розроблено блок-схему. Цей алгоритм у вигляді  
блок-схеми розміщено на рисунках 3.9а.та 3.9б.

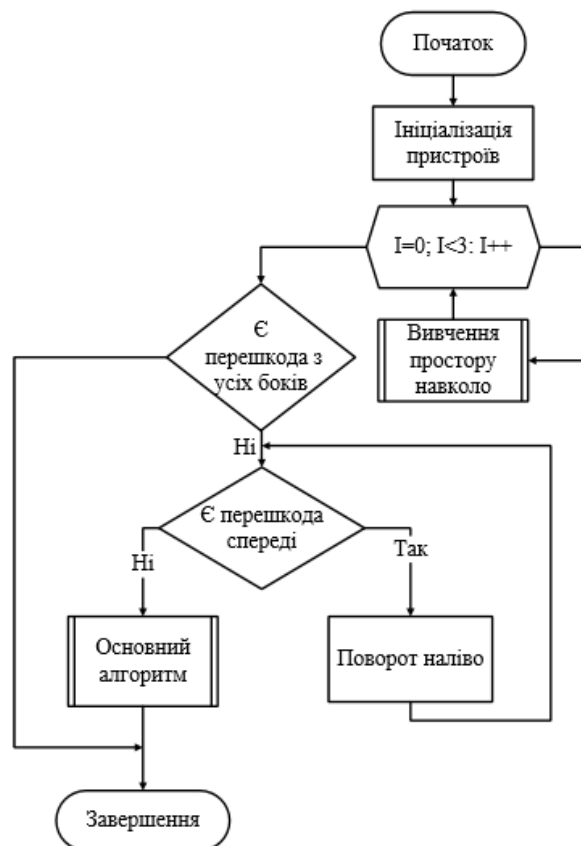


Рис. 3.9а –Блок-схема алгоритму



Рисунок 3.9б – Блок-схема основного алгоритму роботи СК РП

Алгоритм роботи розділений на 2 цикли, цикл вивчення простору навколо та основний цикл роботи. Цикл вивчення простору навколо перевіряє всі напрямки. Основний цикл роботи відповідає за роботу пристрою.

Згідно наведеного вище алгоритму, робот-пилосос повинен вміти зрозуміти де він знаходиться та обрати напрямок руху, після чого почнеться основний алгоритм, за яким він буде працювати на протязі всього часу активності. Цей алгоритм дає можливість повністю виконувати основні переміщення робочою територією.

Оскільки алгоритм роботи вже готовий, можна починати писати код прошивки.

В першу чергу необхідно написати код затримки, оскільки згідно алгоритму, їх потрібно буде використовувати доволі часто. Затримка викликатиметься як функція, яка прийматиме значення. Після чого встановлюється таймер, який рахує (згідно налаштувань в пункті 3.3) в порядку збільшення, і через цикл виконується перевірка значення. Коли значення таймера стає рівне значенню, отриманому функцією, цикл завершується.

```
void delay(uint16_t time)
{
    __HAL_TIM_SET_COUNTER(&htim1, 0);
    while (__HAL_TIM_GET_COUNTER(&htim1) < time);
}
```

Рисунок 3.10 – Лістинг коду затримок

Код отримання значення з сонара та розрахунок відстані до перешкоди наведений на лістингу:

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1) – Перевірка
чи вхідні дані це канал;
    {
        if (Is_First_Captured==0) – Перевірка чи це перші отримані дані;
        {
            IC_Val1 = HAL_TIM_ReadCapturedValue(htim,
TIM_CHANNEL_1); –Отримання першого значення;
            Is_First_Captured = 1; – Встановлення прапора першого значення в
1, щоб цей процес більше не повторювався;
```

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
						38
Змн.	Арк	№ документа	Підпис	Дата		

Далі виконується зміна полярності таймера і відрахунок починається в протилежний бік;

```
    __HAL_TIM_SET_CAPTUREPOLARITY(htim,      TIM_CHANNEL_1,  
TIM_INPUTCHANNELPOLARITY_FALLING);  
}
```

else if (Is\_First\_Captured==1) – Перевірка чи перші дані вже отримані

```
{  
    IC_Val2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1); –
```

Читання другого значення;

```
    __HAL_TIM_SET_COUNTER(htim, 0); – Скидання лічильника;
```

if (IC\_Val2 > IC\_Val1) – Якщо друге отримане значення затримки більше за перше то виконується обрахунок різниці шляхом віднімання, якщо ні то це означає що відстань доволі велика і таймер почав відрахунок заново, тому потрібно відняти значення першого значення від максимального та додати друге значення. Таким чином отримується значення затримки між надсиланням та відлунням сигналу сонара. Код цього алгоритму вказано на рисунку 3.11.

```
{  
    Difference = IC_Val2-IC_Val1;  
}  
else if (IC_Val1 > IC_Val2)  
{  
    Difference = (0xffff - IC_Val1) + IC_Val2;  
}
```

Рисунок 3.11 – Лістинг коду розрахунку відстані

В наступному кроці виконується розрахунок дистанції шляхом множення на швидкість звуку поділену на 2 (тому що, звук проходить цю відстань двічі до об'єкта та від його);

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						39
Змн.	Арк	№ документа	Підпис	Дата		

```
Distance = Difference * .034/2;
```

```
Is_First_Captured = 0; – Скидання прапора першого значення;
```

Відновлюємо полярність та вимикаємо таймер:

```
__HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1,  
TIM_INPUTCHANNELPOLARITY_RISING);  
__HAL_TIM_DISABLE_IT(&htim1, TIM_IT_CC1);}}
```

Для того щоб цей алгоритм запрацював, необхідно подати сигнал довжиною 10мс на пін Trig сонара. Тоді сонар поверне значення через пін Echo та спрацює переривання, яке запустить цей алгоритм. Ця послідовність дій теж викликатиметься, як функція HCSR04\_Read.

Лістинг коду:

```
void HCSR04_Read (void)  
{  
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET); –  
Встановлення піна Trig(A9) в значення High;  
    delay(10); – Затримка 10мс  
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);  
–Скидання піна Trig(A9) на значення Low;  
    __HAL_TIM_ENABLE_IT(&htim1, TIM_IT_CC1); Запуск таймера  
}
```

Для спрощення роботи з двигунами були використані ідентифікатори через команду #define. Код з оголошенням всіх ідентифікаторів розміщено на рисунку 3.12.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
						40
Змн.	Арк	№ документа	Підпис	Дата		



```

#define Pr_Vpered HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_SET) ;
#define Pr_V_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1, GPIO_PIN_RESET);
#define Pr_Nazad HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_SET);
#define Pr_N_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2, GPIO_PIN_RESET);
#define Lv_Vpered HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_SET);
#define Lv_V_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3, GPIO_PIN_RESET);
#define Lv_Nazad HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
#define Lv_N_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);

```

Рисунок 3.12 – Лістинг коду оголошення ідентифікаторів;

В свою чергу ці скорочення використовуються в функціях роботи з двигунами.

Є 5 основних функцій:

- Ruh\_Vpered – виконує перемикання пінів A1 та A3 в High;
- Ruh\_Nazad – виконує перемикання пінів A2 та A4 в High;
- Stop – перемикає всі піни керування двигунами в Low;
- Pov\_Livo – виконує перемикання пінів A1 та A4 в High, та має затримку, розміром достатнім щоб зробити поворот на 90°;
- Pov\_Pravo – виконує перемикання пінів A2 та A3 в High, та має затримку, розміром достатнім щоб зробити поворот на 90°;

Ці прості функції дозволяють повністю керувати двома електродвигунами без жодних проблем.

Перейдемо до розгляду алгоритму стартової ініціалізації.

Цей алгоритм використовує 4 значення, які відповідають за 4 напрямки. Робот-пилосос по черзі обертається на кожен напрямок та перевіряє чи є там перешкода, якщо є - у відповідне значення вноситься 1, якщо ні - алгоритм виконується далі, доки робот не зробить повний оборот. Алгоритм стартової ініціалізації вказано на рисунку 3.13.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						41
Змн.	Арк	№ документа	Підпис	Дата		

```

HCSR04_Read();
HAL_Delay(200);
if (Distance < 5) {
    speredi = 1;
}
Pov_Livo();
Stop();

```

Рисунок 3.13 – Лістинг частини коду стартового алгоритму

Далі, виходячи з результатів ініціалізації, обирається напрямок руху (рис. 3.14). Якщо всі напрямки руху заблоковані, запускається цикл помилки і пристрій вимикає двигуни та сенсор.

```

if (speredi == 1) {
    Pov_Livo();
    Stop();
    if (liv == 1){
        Pov_Livo();
        Stop();
        if (zzadi == 1){
            Pov_Livo();
            Stop();
            if (prav == 1){
                Stop();
                Error_Handler();}}}}

```

Рисунок 3.14 – Лістинг коду визначення напрямку руху

Коли напрямок руху обраний, починається основний алгоритм роботи, який є зацикленним. Виконується контрольна перевірка наявності перешкоди, в разі відсутності - виконується функція руху вперед і через 200мс виконується перехід в початок алгоритму і знову перевіряється наявність перешкоди (рис. 3.15). Якщо перешкоду виявлено, запускається функція Rozvorot:

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						42
Змн.	Арк	№ документа	Підпис	Дата		

```

void Rozvorot (void){
    Stop(); Зупинка двигунів;
    Pov_Pravo(); Поворот вправо;
    Stop(); Зупинка двигунів;
    Ruh_Vpered();
    HAL_Delay(3000); Рух вперед 3 секунди;
    Stop(); Зупинка двигунів;
    Pov_Pravo(); Поворот вправо;

    Stop(); Зупинка двигунів;
}

```

По завершенню виконання даної функції робот-пилосос має змінити свій напрямок на протилежний і виконати перехід до початку основного алгоритму. Далі він рухатиметься паралельно до попередньої траєкторії руху.

```

HCSR04_Read();
if (Distance > 5){
    Ruh_Vpered();
}
else
{
    Rozvorot();
}
HAL_Delay(200);
}

```

Рисунок 3.15 – Лістинг коду перевірки дистанції до перешкоди

По завершенню написання коду роботи СК РП виконано компіляцію проєкту. Для компіляції потрібно натиснути кнопку вказану на рисунку 3.16.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
						43
Змн.	Арк	№ документа	Підпис	Дата		

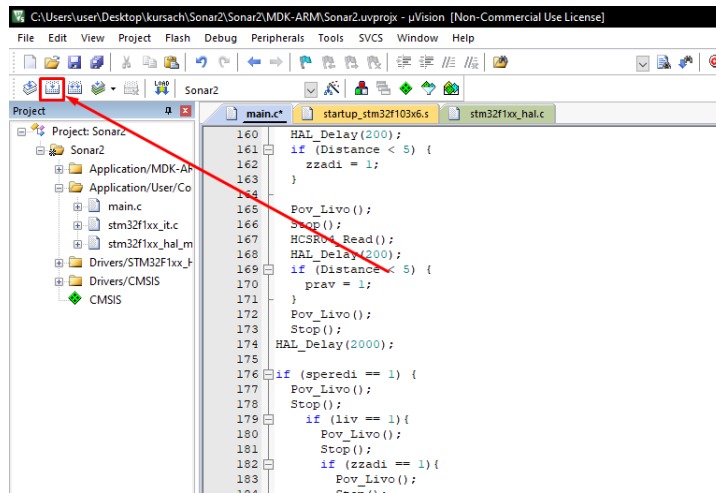


Рисунок 3.16 – Кнопка компілювання проекту

Якщо в програмі немає помилок і компіляція виконалась вдало, то в консолі (знизу в вікні програми) має бути результат, який зображено на рисунку 3.17.

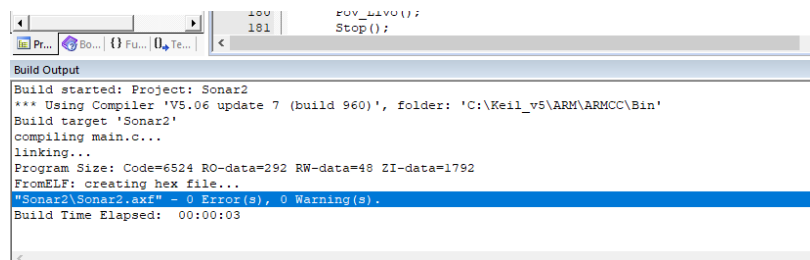


Рисунок 3.17 – Завершення компіляції без помилок;

### 3.3 Тестування пристрою

Для тестування роботи СК РП був розроблений макет пристрою. Макет розроблено згідно схеми електричної принципової. Єдиною відмінністю є відсутність електричних двигунів та батареї, а для відображення роботи буде використано світлодіоди. В результаті був отриманий макет СК РП, зображений на рисунку 3.18.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
Змн.	Арк	№ документа	Підпис	Дата		44

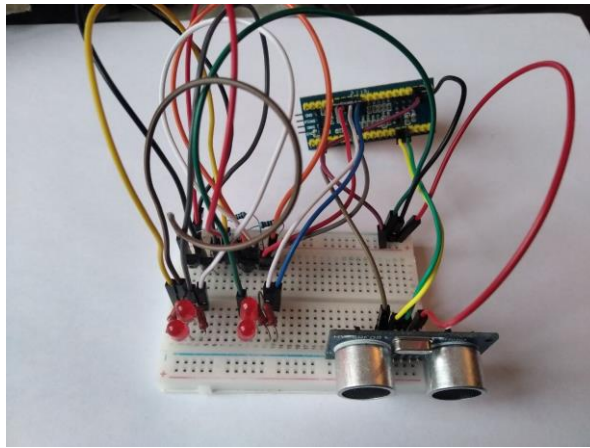


Рисунок 3.18 – Макет системи керування роботом-пилососом

Далі необхідно завантажити прошивку на мікроконтролер. Для цього потрібно підключити програматор, з'єднавши відповідні входи (рисунок 3.19).

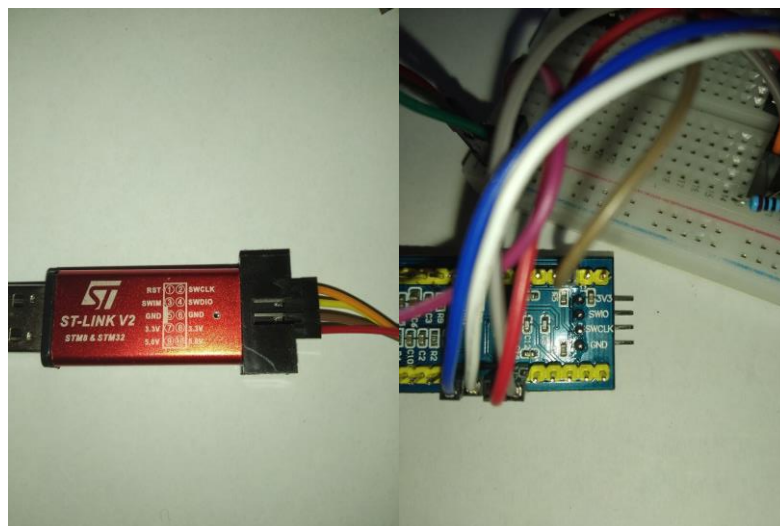


Рисунок 3.19– Програматор та входи відлагодження мікроконтролера

IDE Keil має вбудований драйвер та утиліту для роботи з програматором ST-link, тому завантаження прошивки можна виконати відразу в редакторі. На рисунку 3.20 показано процес запису програми в пам'ять контролера.

					<b>КС КРБ 123.216.00.00 ПЗ</b>	Арк
						45
Змн.	Арк	№ документа	Підпис	Дата		

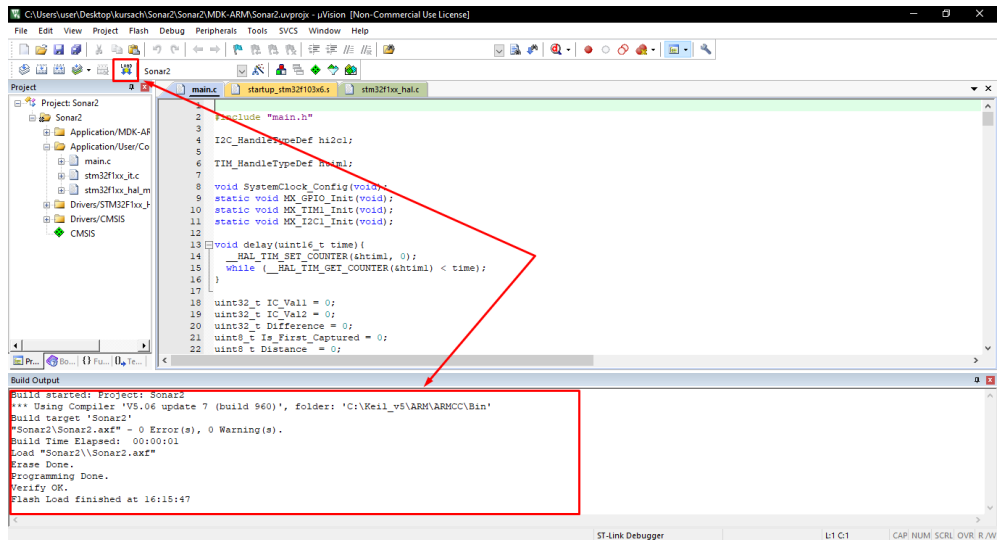


Рисунок 3.20 – Запис програми в пам'ять контролера

Спочатку треба перевірити роботу сонара, тобто перевірити чи правильно вимірюється відстань. Для цього скористаємось режимом відлагодження (рисунок 3.21), щоб отримати значення відстані.

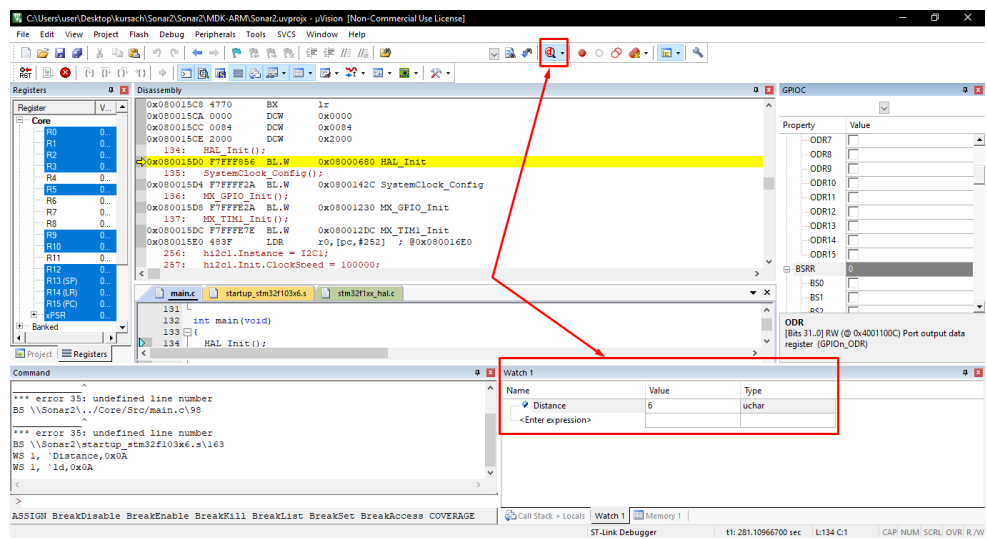


Рисунок 3.21 – Кнопка переходу до режиму відлагодження та розміщення відображення змінних.

Проведемо кілька тестів:

Тест 1:

						Арк
						46
Змн.	Арк	№ документа	Підпис	Дата	КС КРБ 123.216.00.00 ПЗ	

Встановлено відстань 6см від сонара (Рис. 3.22). В програмі відобразилась відстань вірно, алгоритм запусився та пройшов етап ініціалізації без жодних проблем. Оскільки відстань більше 5см, то програма продовжує рух вперед. Це представлено на скріншоті нижче (Рис. 3.22). Два світлодіоди, які відповідають за рух робота вперед, працюють справно. Такий рух продовжуватиметься поки перешкода не наблизиться на відстань меншу за 5см. В результаті першого тесту алгоритм спрацював без жодних помилок.

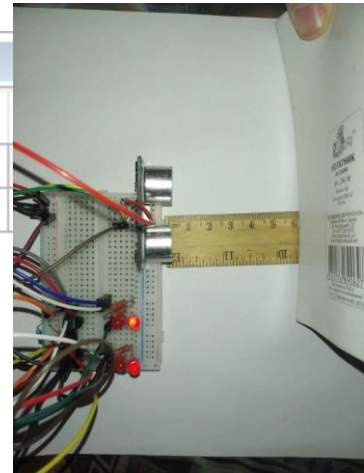
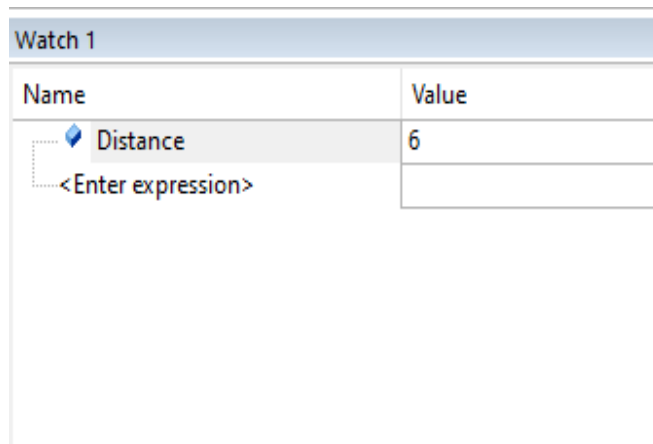


Рисунок 3.22 – Результати тесту 1

#### Тест 2:

Встановлено відстань 6см від сонара (Рис. 3.23). Результат вимірювання відображується в програмі без помилок. Як і в першому тесті він виконує рух вперед, відстань більша 5см. Підчас другого тесту - помилок в роботі не виявлено.

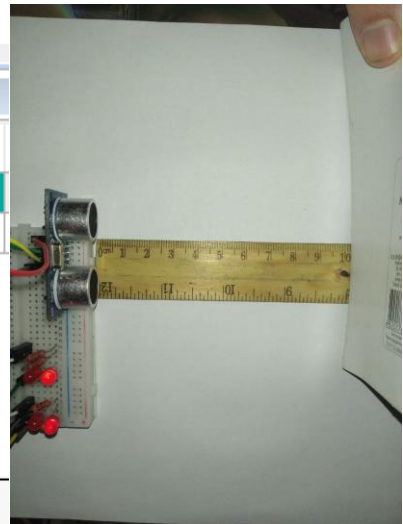
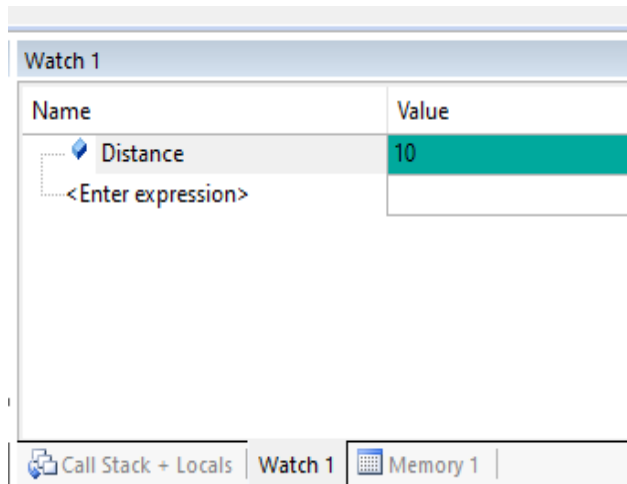


Рисунок 3.23 – Результати тесту 2

### Тест 3:

Встановлено відстань 3см від сонара (рис. 3. 24). Результат вимірювання відображується в програмі без помилок. В даному випадку відстань вже менша ніж 5см, а тому було почато алгоритм розвороту. На рисунку 3.24 видно, що лівий двигун виконує рух вперед, а правий - назад, таким чином виконується поворот направо. В результаті проведення тесту не було виявлено жодних помилок.

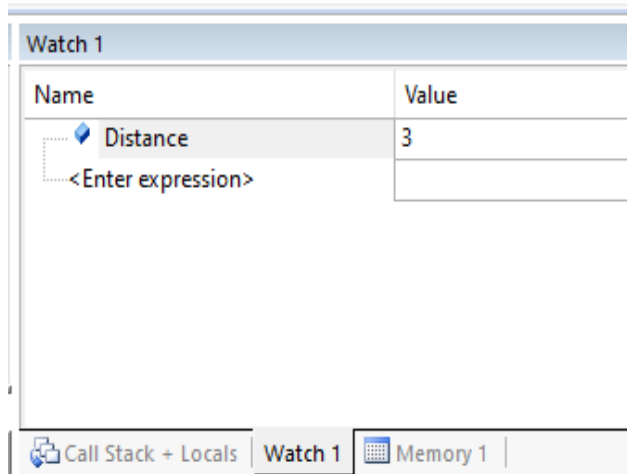


Рисунок 3.24 – Результати тесту 3



Результати тестування:

Після проведення тестів було визначено, що звуковий дальномір працює без помилок і відображає реальну відстань, час обрахунку відстані мінімальний, що дає можливість пристрою реагувати на наявність перешкод максимально оперативно.

Алгоритми руху працюють без жодних нарікань. Перемикання режимів роботи двигунів виконуються своєчасно і без жодних помилок. Алгоритм роботи виконується справно. Рух вперед, поворот вліво та вправо, а також розворот - працюють без помилок.

Висновок тестування: розроблена СК РП виконує свої функції, при тестуванні пристрою не було виявлено жодних помилок. Алгоритм роботи не порушується, всі функції виконуються.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						49
Змн.	Арк	№ документа	Підпис	Дата		

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Заходи безпеки при експлуатації електроустановок

Електроенергія і працюють на її основі агрегати – велике благо для людства, проте, якщо поводитися з нею неправильно, вона може завдати величезної шкоди і людям, і навколишньому середовищу. Для запобігання нещастя фахівцями розробляється безліч правил, інструкцій, технічних регламентів, багато пунктів яких буквально писані людською кров'ю. З метою забезпечення надійного функціонування електротехніки та захисту праці на виробництвах, де основним фактором служить електрострум, написані і введені в дію «Правила техніки безпеки при експлуатації електроустановок».[12]

Діючі в даний час правила безпеки вступили в силу з 2014 року і є обов'язковими для виконання всіма промисловими споживачами електричної енергії, незалежно від приналежності підприємства.

Положення правил з охорони праці стосуються, перш за все, електротехнічних, електротехнологічних і деяких неелектротехнічних працівників, які контактують з різними частинами електрообладнання в процесі його техобслуговування, перемикання, монтажно-налагоджувальних та інших (в т.ч. і будівельних) видів робіт. Також під дію правил підпадають їх роботодавці: юрособи та фізособи, вони зобов'язані гарантувати безпечну працю на своїх підприємствах. Зверніть увагу! З метою підвищення надійності виробництва роботодавці можуть встановити навіть більш жорсткі вимоги до електробезпеки (звичайно, не в протизагони офіційним Правил). Працівники повинні бути ознайомлені з інструкціями з охорони їх праці та належним чином проінструктовані. Електроустановки і лінії, які беруть участь у виробничому, передавальному і розподільчому процесах, повинні бути справні і забезпечені

					<b>КС КРБ 123.216.00.00 ПЗ</b>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		Кардаш І.А.			<i>Практична частина</i>	<i>Лім.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевірів</i>		Осухівська Г.М.					50	70
<i>Консульт.</i>		Лазарюк В.В.				<i>ТНТУ, каф. КС, гр. СІс-43</i>		
<i>Н. Контр.</i>		Луцик Н.С.						
<i>Затвердив</i>		Осухівська Г.М.						

різними захисними пристосуваннями і надання долікарської допомоги.[14]

Згідно наказом міністерства палива та енергетики від 25.07.2006 , всі члени виробничого процесу за участю електричної енергії повинні періодично навчатися і підтверджувати свій творчий хист до електробезпеки роботи (присвоюються групи допуску з I по V і видаються типові посвідчення), а деякі категорії осіб – ще й проходити медогляди (якщо працівники молодше 21 років, щорічні) . При порушенні цих правил винні особи несуть різні види відповідальності. [15]

Згідно наказом №4 міністерства праці та соціальної політики України від 09.01.98 [16] при роботах на електроустановках передбачений ряд правил, які спрямовані на їх неухильне дотримання:

- роботи в рамках ординарної експлуатації електричних агрегатів оформляються нарядами, розпорядженнями або списками;
- підготовка робочої зони і допуск на неї оформляється відповідним дозволом;
- для здійснення робочих маніпуляцій потрібно допуск, відповідно до присвоєної групою електробезпеки;
- здійснення підстраховки і контролю в період проведення робіт;
- офіційне оформлення початку і кінця перерви в роботах, також їх повного завершення або ж переведення бригади на іншу ділянку.
- Документом визначається коло працівників, на яких покладається відповідальність за безпеку протікання робочих процесів на електроустановках:
  - підписують наряди-допуски, що віддають розпорядження або посвідчують список робіт з поточного використання електроагрегатів (необхідна група допуску з електробезпеки – 4-5) – вони зобов'язані попередньо оцінити потрібність виконання роботи і можливість її безпечного здійснення; склад бригад, вибір спостерігача і виконавця робіт і попередній інструктаж повинні відповідати вимогам безпеки;

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						51
Змн.	Арк	№ документа	Підпис	Дата		

– дозволяють підготовку робочих зон (група допуску – 4-5 або адміністративно-технічні працівники, уповноважені розпорядженням керівництва підприємства) – відповідають за своєчасне і безпечне від'єднання електрообладнання від мережі і його заземлення, координування робочого часу і розташування оперативних бригад, облік бригад і їх членів, відстеження інформації про терміни проведення поточних маніпуляцій і часу подачі напруги на електроустановки;

– призначені відповідальні керівники робіт (групи допуску – 4-5, призначаються при використанні різних механізмів і вантажопідйомників, знеструмленні електричного обладнання, на кабельних лініях зв'язку недалеко від транспортних магістралей, на знову запускаються повітряних лініях або ж їх фазному ремонті, при роботах з наведеною напругою і ряді інших випадків) – зобов'язані стежити за виконанням етапів підготовки робочого місця, достатністю та правильністю інструктування учасників бригади, загальною безпекою робочих моментів, вживати додаткових заходів електробезпеки;

– допускають до робіт (член електротехнічного персоналу з групою допуску 3-4) – несе відповідальність за достатність заходів безпеки, відповідність їх нарядом-допуском або розпорядженням, вимогам майбутніх маніпуляцій на електрообладнанні, правильність і обсяг проведеного інструктажу;

– безпосередні виконавці (група електродопуска – 3-4) – відповідають за правильно підготовлену робочу зону, наявність, достатність і справність індивідуальних і загальних електрозахисних засобів, інструментарію і пристосувань, наочних інформаційних засобів у вигляді значків безпеки або попереджувальних плакатів, огорож небезпечних місць, за дотримання правилами ТБ всіх членів бригади і контроль за безпекою їх дій під час робочого процесу;

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						52
Змн.	Арк	№ документа	Підпис	Дата		

– спостерігачі в бригадах (група електродопуска – 3) – стежать за відповідністю робочої зони заходам з безпеки; чіткістю і достатністю попереднього інструктажу; збереженням запорів на приводах, огорож і заземлюючих пристроїв; охороною виконавців робіт від електротравм;

– всі працівники – члени бригади, здійснює поточні роботи на електроустановках, зобов'язані дотримуватися правил техніки безпеки, інструкції та вказівки, що стосуються охорони їх праці.

Способи і засоби, що використовують для забезпечення безпеки людини під час роботи з електроустановками, використовуються залежно від умов дотику до струмоведучої частини. Глибоке розуміння умов дотику і ступеня небезпеки ураження в кожному випадку дозволить обрати необхідну комбінацію способів і засобів захисту. Треба розуміти, що захист від прямого дотику запобігає ураженню електричним струмом за відсутності пошкодження ізоляції провідників, проте захист у разі непрямого дотику – у випадку одиничного пошкодження. [13]

Кожному відомо про важливість якості електричної ізоляції, тобто ізоляції яка чи конструкції виконаної з діелектрика. Цим діелектриком вкриваються провідники струму або розділяються одна від одної. Ця ізоляція запобігає протіканню струму завдяки великому опору.

Ізоляція в електроустановках до 1000 В поділяється на наступні види:

– Основна ізоляція, що являє собою шар діелектрика, який яким вкриваються частини під напругою;

– Додаткова ізоляція – ізоляція яка є самостійною та призначена для захисту коли основна ізоляція пошкоджена;

– Подвійна ізоляція – це поєднання основної та додаткової ізоляції;

В даному проекті використовується тільки додаткова ізоляція на струмопровідних деталях, оскільки струми в системі дуже слабкі. В

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						53
Змн.	Арк	№ документа	Підпис	Дата		

акумуляторі +19 вольт і для захисту від настільки малих напруг достатньо додаткової ізоляції.

#### 4.2 Соціальне значення охорони праці

Закон України «Про охорону праці» визначає «Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально профілактичних заходів і засобів, спрямованих на збереження здоров'я та працездатності людини в процесі праці».[17]

Завдання охорони праці - забезпечення безпечних, нешкідливих і сприятливих умов праці через вирішення багатьох складних завдань. Вирішальне значення в розв'язанні цих завдань має науково-технічний прогрес. Використання досягнень науки та техніки сприяє підвищенню рівня безпеки праці, культури та організації виробництва, дозволяє полегшити працю, підсилити її привабливість. Рівень безпеки людини з розвитком цивілізації постійно зростає. Розвиток науки техніки, в цілому збільшуючи безпеку життєдіяльності. людини, призвів до появи цілого ряду нових проблем. Науковий аналіз виробничих травм доводить, що вони виникають головним чином внаслідок втрати міцності та надійності робочої техніки небезпечного стану системи "людина-машина-середовище" та ряду технічних факторів. До технічних факторів відносять насамперед надійність техніки (конструктивні недоліки, технологічні та експлуатаційні порушення, руйнування деталей машин під дією корозії та корозійного. розтріскування), організацію управління охороною праці (документація, правові норми, стандарти безпеки праці, методи навчання тощо), санітарно-гігієнічні умови в приміщеннях та на робочих місцях (шкідливі речовини в робочій зоні, виробниче освітлення, шум, вібрація, іонізуюче, електромагнітне, ультрафіолетове, лазерне випромінювання тощо). [18]

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						54
Змн.	Арк	№ документа	Підпис	Дата		

На мою думку соціальне значення охорони праці полягає в сприянні росту ефективності суспільного виробництва шляхом безперервного вдосконалення і поліпшення умов праці, підвищення їх безпеки, зниження виробничого травматизму і профзахворювань. Тобто дотримання норм та законів охорони праці дає можливість підвищити комфорт та безпеку робочих місць робітників на підприємстві. Підвищення комфорту робочих місць дозволить підняти ефективність роботи працівників. Підвищення безпеки роботи дозволяє вберегти здоров'я, в моральному плані Тобто охорона праці для людей є свого роду охороною права людей на безпечну та комфортну роботу і забезпечує суспільству впевненість в завтрашньому дні.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						55
Змн.	Арк	№ документа	Підпис	Дата		

## ВИСНОВКИ

Результатом виконання кваліфікаційної роботи є розроблена СК РП на основі мікроконтролера STM32F103C6T, яка дає можливість управління рухом двох електродвигунів на основі отриманих даних від сонара. Тобто з допомогою звукового дальноміру, пристрій визначає відстань до перешкоди і, коли відстань пересікає відмітку 5см, міняє траєкторію руху. Такий алгоритм роботи відповідає принципам роботи сучасних роботів-пилососів.

В результаті виконання кваліфікаційної роботи:

- проведено ознайомлення з принципами роботи робота-пилососа;
- розроблено електричну принципову схему СК РП;
- визначені та освоєні інструменти розробки;
- створений алгоритм роботи СК РП;
- написаний код програми;
- проведено тестування системи.

Розроблена СК РП виконує автономне керування рухом робота-пилососа на основі вимірів сонара. При тестуванні пристрою не було виявлено жодних помилок. Алгоритм роботи не порушується, всі функції виконуються.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						56
Змн.	Арк	№ документа	Підпис	Дата		



## ПЕРЕЛІК ПОСИЛАНЬ

1. Керування двигуном постійного струму: Веб-сайт. URL: <http://mikrotik.kpi.ua/index.php/courses-list/category-python/85-dc-motor-control-using-python-and-raspberry-pi-session-7> (Дата звернення 03.04).
2. HAL GPIO Generic Driver. Функції керування портами вводу виводу: Веб-сайт. URL: <http://mypractic.com/hal-gpio-generic-driver-funkcii-upravlinnya-portami-vvoda-vyvoda> (Дата звернення 05.04).
3. Понижуючий DC-DC перетворювач на 5V (3.3V) на базі MC34063: Веб-сайт. URL: <https://blog.avislab.com/mc34063/> (Дата звернення 07.04).
4. Робот-пилосос: Веб-сайт. URL: <https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B1%D0%BE%D1%82-%D0%BF%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%81> (Дата звернення 10.04).
5. STM32F103 documentation: Веб-сайт. URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32f103.html#documentation/>.
6. Ультразвуковий дальномір HC-SR04: Веб-сайт. URL: <https://3d-diy.ru/wiki/arduino-datchiki/ultrazvukovoj-dalnomer-hc-sr04/>.
7. Принципи роботи робота пилососа: Веб-сайт. URL: <https://www.mojo.ua/ua/news/kak-rabotaet-robot-pylesos-i-что-u-nego-vnutri-gayd-v-2-razdelakh.html>
8. Coocox CoIDE, a free and highly-integrated software development environment for ARM Cortex MCUs: Веб-сайт. URL: <https://www.st.com/en/development-tools/coide.html>
9. Програмування stm32 з самих основ: Веб-сайт. URL: <https://habr.com/ua/post/354670/>.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						57
Змн.	Арк	№ документа	Підпис	Дата		

10. HCSR04 and STM32 using Input Capture: Веб-сайт. URL: [https://www.youtube.com/watch?v=ti\\_1ZwRolU4&t=3s&ab\\_channel=ControllersTech](https://www.youtube.com/watch?v=ti_1ZwRolU4&t=3s&ab_channel=ControllersTech).

11. Description of STM32F1 HAL and low-layer drivers: Веб-сайт. URL: [https://www.st.com/resource/en/user\\_manual/dm00154093-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00154093-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf)

12. Скобло Ю.С., Соколовська Т.Б., Морозенко Д.І. та ін. Безпека життєдіяльності. Навчальний посібник для вищих навчальних закладів 3-4 рівнів акредитації. – К.: Кондор, 2003. 424 с.

13. Стищенко Т.Є., Пронюк Г.В., Сердюк Н.М., Хондак І.І. «Безпека життєдіяльності»: навч. посібник / Т.Є Стищенко, Г.В. Пронюк, Н.М. Сердюк, І.І. Хондак. – Харків: ХНУРЕ, 2018. – 336 с.

14. Толлок А.О., Крюковська О.А. Безпека життєдіяльності. Навчальний посібник. – Дніпродзержинськ: ДДТУ, 2011. 215 с.

15. Наказ №4 «Про затвердження Правил безпечної експлуатації електроустановок споживачів»: Веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/z0093-98#Text>

16. Наказ №258 «Про затвердження Правил технічної експлуатації електроустановок споживачів»: Веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/z1143-06#Text>

17. Закон України «Про охорону праці»: Веб-сайт. URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

18. Геврик Є.О. Охорона праці. – К.: Ельга, Ніка-Центр, 2003. 280 с.

					<i>КС КРБ 123.216.00.00 ПЗ</i>	Арк
						58
Змн.	Арк	№ документа	Підпис	Дата		

Додаток А.  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя  
Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

"Затверджую"

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

" \_\_\_\_ " \_\_\_\_\_ 2022 р.

СИСТЕМА КЕРУВАННЯ РОБОТОМ-ПИЛОСОСОМ

ТЕХНІЧНЕ ЗАВДАННЯ на 5 листах

На здобуття освітнього ступеня бакалавра

Спеціальність 123 «Комп'ютерна інженерія»

"УЗГОДЖЕНО"

Керівник кваліфікаційної роботи

\_\_\_\_\_ к.т.н., Осухівська Г.М.

" \_\_\_\_ " \_\_\_\_\_ 20 \_\_\_\_ р.

"ВИКОНАВЕЦЬ"

Студент групи СІс-43

\_\_\_\_\_ Кардаш І.А.

" \_\_\_\_ " \_\_\_\_\_ 20 \_\_\_\_ р.

Тернопіль 2022

## 1. Назва та підстава для виконання роботи

### 1.1 Система керування роботом-пилососом.

1.2 Підставою для виконання кваліфікаційної роботи бакалавра (КРБ) є Наказ по Університету (№ 4/7-180 від 23.03.2022 р.).

## 2. Виконавець

2.1. Студент групи СІс-43 123 спеціальності 123 «Комп'ютерна інженерія»

Тернопільського національного технічного університету ім. І. Пулюя  
Кардаш Іван Анатолійович

## 3. Мета роботи

3.1 Метою роботи є розробка структури та програмного забезпечення системи керування роботом-пилососом.

## 4. Склад виробу

### 4.1 До складу виробу повинні входити:

- 1)давач;
- 2)плата мікроконтролера;
- 3)двигуни
- 4)понижувач струму
- 5) комплект документації

## 5. Технічні вимоги

### 5.1 Вимоги по призначенню

5.1.1 Пристрій керований системою повинен мати наступні параметри:

- Діапазон вимірювання відстані, мм 10...100
- Точність вимірювання відстані, мм 5
- Швидкість руху, м/хв 3
- Швидкість повороту на 90°, с 2

5.1.2 Виріб повинен живитись напругою постійного струму +19В  $\pm$ 2В

5.2 Вимоги до експлуатації:

5.2.1 По умовам експлуатації виріб повинен відповідати вимогам ГОСТ Р 60.6.3.22-2021.

5.2.2 Температура експлуатації від -5 °С до +40 °С

5.2.3 Відносна вологість до 80% при t=25 °С

5.2.4 Рівні поверхні

5.3 Конструктивні вимоги.

5.3.1 Конструювання корпусу приладу в кваліфікаційній роботі не передбачено.

5.3.2 Для побудови системи мають бути використані сучасні компоненти з можливістю поверхневого монтажу друкованого вузла.

5.3.3 При побудові системи необхідно передбачити розміщення роз'ємів живлення.

5.3.4 Габаритні розміри при макетування, мм, не більше:

- Довжина, мм 80
- Ширина, мм 100
- Висота, мм 40

5.3.5 Маса макету, не більше 300г

5.3.6 Конструкція макету повинна забезпечувати доступ до всіх комплектуючих виробів при тестуванні.

#### 5.4 Вимоги до надійності.

5.4.1 Система повинна відповідати ДСТУ 2862-94.

5.4.2 Час напрацювання на відмову, не менше 5000 год.

#### 5.5 Вимоги метрології.

5.5.1 Вимірювання параметрів системи при моделюванні повинно виконуватись на універсальних вимірювальних приладах.

#### 6. Економічні показники

6.1 Собівартість системи повинна бути не більше 3000 грн.

#### 7. Вимоги до документації

7.1 Конструкторська документація повинна відповідати вимогам ЄСКД, ЄСПД, ДСТУ та ГОСТ.

7.2 До складу документації повинно входити:

- 1) ПЗ;
- 2) Структурна схема;
- 3) Електрично принципова схема;
- 4) Блок схема алгоритму роботи;
- 5) Загальний вигляд системи.

#### 8. Стадії та етапи розробки кваліфікаційної роботи бакалавра

8.1 Стадії та етапи виконання кваліфікаційної роботи наведенні в табл.

1.

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

	Назва етапу	Строк виконання	
		початок	кінець
1.	<i>Ознайомлення з завданням до кваліфікаційної роботи</i>	23.03.22	14.03.22
2.	<i>Розробка технічного завдання</i>	14.03.22	17.03.22
3.	<i>Робота над розділом «Аналіз технічного завдання»</i>	17.03.22	21.03.22
4.	<i>Розробка узагальненої структурної схеми</i>	21.03.22	23.03.22
5.	<i>Визначення необхідних апаратних складових</i>	23.03.22	29.03.22
6.	<i>Проектування електрично-принципової схеми</i>	29.03.22	04.04.22
7.	<i>Робота над розділом «Проектна частина»</i>	04.04.22	19.04.22
8.	<i>Розробка алгоритму та написання коду роботи пристрою</i>	19.04.22	21.04.22
9.	<i>Тестування пристрою</i>	21.04.22	28.04.22
10.	<i>Робота над розділом «Практична частина»</i>	28.04.22	01.05.22
11.	<i>Виконання завдань до розділу «Безпека життєдіяльності, основи охорони праці»</i>	01.05.22	09.05.22
12.	<i>Оформлення пояснювальної записки</i>	09.05.22	19.05.22
13.	<i>Нормоконтроль</i>	16.05.22	22.05.22
14.	<i>Перевірка на плагіат</i>	15.06.22	–
15.	<i>Захист КБ</i>	22.06.22	–

9. В дане технічне завдання можуть вноситись зміни по узгодженню сторін.



ДОДАТОК Б.  
Перелік елементів

Позначення	Найменування	Кількість	Примітка
Мікросхеми			
U1	STM32F103C6T	1	
U2	Сонар SRF04	1	
U3	МС34063	1	
Резистори			
R1-R8	Резистор 270 Ом	8	
R9	Резистор 5.1 КОм	1	
R10	Резистор 10 КОм	1	
R11	Резистор 0.3 Ом	1	
Електродвигуни			
M1-M2	Електродвигун 19 В	2	
Транзистори			
Q1-Q8	Транзистори S9013	8	
Діоди			
D1	Діод Шотткі 10BQ015	1	
Конденсатори			
C1	Конденсатор 0.50pF	1	
C2;C3	Конденсатор 470pF	2	

<b>КС КРБ 123.216.00.00 ПЗ</b>					
Змн.	Арк.	№ докум.	Підпис	Дата	
Розробив	Кардаш І.А.				
Перевірів	Осухівська Г.М.				
Рецензент	Дуда О.М.				
Н. Контр.	Луцик Н.С.				
Затвердив	Осухівська Г.М.				
<b>Перелік елементів</b>			Лім.	Арк.	Аркушів
				64	70
			ТНТУ, каф. КС, гр. СІс-43		

## ДОДАТОК В.

### Лістинг коду прошивки main.c

```
#include "main.h"
I2C_HandleTypeDef hi2c1;
TIM_HandleTypeDef htim1;
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM1_Init(void);
static void MX_I2C1_Init(void);
void delay(uint16_t time){
    __HAL_TIM_SET_COUNTER(&htim1, 0);
    while (__HAL_TIM_GET_COUNTER(&htim1) < time);
}
uint32_t IC_Val1 = 0;
uint32_t IC_Val2 = 0;
uint32_t Difference = 0;
uint8_t Is_First_Captured = 0;
uint8_t Distance = 0;
#define TRIG_PIN GPIO_PIN_9
#define TRIG_PORT GPIOA
#define Pr_Vpered HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1,
GPIO_PIN_SET) ;
#define Pr_V_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_1,
GPIO_PIN_RESET);
#define Pr_Nazad HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2,
GPIO_PIN_SET);
#define Pr_N_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_2,
GPIO_PIN_RESET);
#define Lv_Vpered HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3,
GPIO_PIN_SET);
#define Lv_V_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_3,
GPIO_PIN_RESET);
#define Lv_Nazad HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4,
GPIO_PIN_SET);
#define Lv_N_OFF HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4,
GPIO_PIN_RESET);
int liv =0;

int prav =0;
int zzadi =0;
int spered =0;
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1) {
        if (Is_First_Captured==0) // if the first value is
not captured
        {
            IC_Val1 = HAL_TIM_ReadCapturedValue(htim,
TIM_CHANNEL_1); // read the first value
```

```

        Is_First_Captured = 1
        __HAL_TIM_SET_CAPTUREPOLARITY(htim,
TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_FALLING);
    }
    else if (Is_First_Captured==1)
    {
        IC_Val2 = HAL_TIM_ReadCapturedValue(htim,
TIM_CHANNEL_1);
        __HAL_TIM_SET_COUNTER(htim, 0);
        if (IC_Val2 > IC_Val1)
        {
            Difference = IC_Val2-IC_Val1;
        }
        else if (IC_Val1 > IC_Val2)
        {
            Difference = (0xffff - IC_Val1) + IC_Val2;
        }
        Distance = Difference * .034/2;
        Is_First_Captured = 0;
        __HAL_TIM_SET_CAPTUREPOLARITY(htim,
TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_RISING);
        __HAL_TIM_DISABLE_IT(&htim1, TIM_IT_CC1);
    }
}

}

void HCSR04_Read (void)
{
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET);
    delay(10);
    HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);
    __HAL_TIM_ENABLE_IT(&htim1, TIM_IT_CC1);
}

void Ruh_Vpered (void){
    Pr_Vpered;
    Lv_Vpered;
}

void Ruh_Nazad (void){
    Pr_Nazad;
    Lv_Nazad;
}

void Stop (void){
    Pr_V_OFF;
    Pr_N_OFF;
    Lv_V_OFF;
    Lv_N_OFF;
}

void Pov_Livo(void){
    Pr_Vpered;
    Lv_Nazad;
    HAL_Delay(2000);
}

void Pov_Pravo(void){

```

```

        Pr_Nazad;
        Lv_Vpered;
        HAL_Delay(2000);
    }
    void Rozvorot (void){
        Stop();
        Pov_Pravo();

        Stop();
        Ruh_Vpered();
        HAL_Delay(3000);
        Stop();
        Pov_Pravo();
        Stop();
    }
    int main(void)
    {
        HAL_Init();
        SystemClock_Config();
        MX_GPIO_Init();
        MX_TIM1_Init();
        MX_I2C1_Init();
        HAL_TIM_IC_Start_IT(&htim1, TIM_CHANNEL_1);
        HCSR04_Read();
        HAL_Delay(200);
        if (Distance < 5) {
            spered = 1;
        }
        Pov_Livo();
        Stop();
        HCSR04_Read();
        HAL_Delay(200);
        if (Distance < 5) {
            liv = 1;
        }
        Pov_Livo();
        Stop();
        HCSR04_Read();
        HAL_Delay(200);
        if (Distance < 5) {
            zzadi = 1;
        }

        Pov_Livo();
        Stop();
        HCSR04_Read();
        HAL_Delay(200);
        if (Distance < 5) {
            prav = 1;
        }
        Pov_Livo();
        Stop();
    }

```

```

HAL_Delay(2000);
if (spered_i == 1) {
    Pov_Livo();
    Stop();
    if (liv == 1){
        Pov_Livo();
        Stop();
        if (zzadi == 1){
            Pov_Livo();
            Stop();
            if (prav == 1){
                Stop();
                Error_Handler();
            }
        }
    }
}
while (1)
{
    HCSR04_Read();
    if (Distance > 5){
        Ruh_Vpered();
    }
    else
    {
        Rozvorot();
    }
    HAL_Delay(200);
}
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;

```

```

RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct,
FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}

static void MX_I2C1_Init(void)
{
    hi2c1.Instance = I2C1;
    hi2c1.Init.ClockSpeed = 100000;
    hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        Error_Handler();
    }
}

static void MX_TIM1_Init(void)
{
    TIM_MasterConfigTypeDef sMasterConfig = {0};
    TIM_IC_InitTypeDef sConfigIC = {0};
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 72-1;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 0xffff-1;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
    htim1.Init.AutoReloadPreload =
TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_IC_Init(&htim1) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode =
TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim1,
&sMasterConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sConfigIC.ICPolarity = TIM_INPUTCHANNELPOLARITY_RISING;
}

```

```

    sConfigIC.ICSelection = TIM_ICSELECTION_DIRECTTI;
    sConfigIC.ICPrescaler = TIM_ICPSC_DIV1;
    sConfigIC.ICFilter = 0;
    if (HAL_TIM_IC_ConfigChannel(&htim1, &sConfigIC,
TIM_CHANNEL_1) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA,
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4
                        |GPIO_PIN_9, GPIO_PIN_RESET);
    GPIO_InitStructure.Pin = GPIO_PIN_13;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);
    GPIO_InitStructure.Pin
GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4
                        |GPIO_PIN_9;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
}
void Error_Handler(void)
{
    __disable_irq();
    while (1)
    {
    }
}
#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file
name and line number,
    ex: printf("Wrong parameters value: file %s on line
%d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```