

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

КАФЕДРА КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ
АППАРАТУРИ

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

З курсу:

«Обчислювальні та мікропроцесорні засоби в РЕА»

Тема: «Пристрій керування кроковим двигуном»

Керівник:

доц. Корнєв В.П.

Допущено до захисту

“ ____ ” _____ 20__ р.

Захищено з оцінкою

Виконав:

Дмитрук О.О.

студент III курсу ФЕЛ

групи ДК-82

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-82 Семестр VI

ЗАВДАННЯ

До розрахунково-графічної роботи

Дмитрука Олександра Олександровича

(прізвище, ім'я та по-батькові)

1. Тема проекту Пристрій керування кроковим двигуном

2. Строк здачі студентом закінченого проекту (роботи) 28.05.2021

3. Вихідні дані до проекту (роботи):

Спроектувати пристрій керування швидкістю обертання крокового двигуна в залежності від відстані до перешкоди. При зближенні до перешкоди ($< L_{\min}$) виконати зупинку двигуна. Натискання кнопки – реверс крокового двигуна.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що розроблюються)

1. Опис структури пристрою та його складових

2. Проектування схеми електричної принципової

3. Розробка програми керування пристроєм

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

1. Схема електрична принципова

2. Перелік елементів

6. Дата видачі завдання 10.04.2021

Зміст

Вступ	4
Розділ 1. Опис структури пристрою та його складових	6
1.1. Структура пристрою.....	6
1.2. Принцип роботи блоку керування напрямку руху двигуна.....	7
1.3. Принцип роботи блоку крокового двигуна.....	7
1.4. Принцип роботи блоку вимірювання відстані до перешкоди.....	9
Розділ 2. Проектування схеми електричної принципової.....	11
2.1. Проектування схеми електричної принципової	11
2.2. Вибір елементної бази для реалізації блоків пристрою	12
2.3. Налаштування системи тактування мікроконтролера.....	13
3. Програма керування пристроєм	15
3.1. Структура програми.....	15
3.2. Структура даних.....	16
3.3. Опис алгоритмів окремих підпрограм.....	20
3.3.1. Опис роботи функції init_gpio();.....	20
3.3.2. Опис функції enable_step_tim2():.....	21
3.3.3. Опис функції enable_tim_3_4 () :.....	22
3.3.4. Опис функції обробки переривання “IM4_IRQHandler()”	24
3.3.5. Опис функції обробки переривання “IM2_IRQHandler()”	25
3.3.6. Опис функції обробки переривання EXTI15_10_IRQHandler ():	25
3.3.7. Опис функції керування двигуном “drive(&Tacts)”	26
Висновок	27
Список використаних джерел.....	28

					ДК82.460839.001 ПЗ			
Зм.		№ докум.						
Розробив		Дмитрук О.О.			Пристрій керування кроковим двигуном	Літ.	Арк.	Аркушів
Перевірів							3	28
Реценз.						НТУУ «КПІ ім. І.Сікорського», ФЕЛ, ДК-82		
Н.Контр.								
Затв.		Корнєв В.П.						

Вступ

Пристрій, що поставлено за мету розробити в рамках даної розрахунково-графічної роботи, являє собою пристрій керування крокового двигуна, швидкість обертання якого залежить від відстані до перешкоди.

За допомогою далекоміру HC-SR04 відбувається зчитування відстані до перешкоди. Кроковий двигун, а саме 28BJ-48 може точно переміщатися на мінімально можливий кут, який називається кроком. Для практичних задач можна вважати що кроковий двигун дещо схожий на серво-привод. В нашому варіанті, відбувається безперервне обертання двигуна, зі зміною його швидкості, а у випадку занадто малої відстані до перешкоди, його зупинка. Така система може бути необхідна в безпілотних приладах, роботи-пилососи, тощо. Драйвер ULN2003 котрий іде в комплекті з кроковим двигуном, являє собою набір крепких складових ключів, котрі використовуються в колах індуктивних навантажень. За допомогою даного драйверу, мікроконтроллер може напряду керувати двигуном. Натисканням зовнішньої кнопки, відбуватиметься зміна напрямку руху двигуна.

Виконання обчислень буде проводитись мікроконтролером STM32F401RE, розміщеним на налагоджувальній платі STM32F401 Nucleo. Він являє собою дуже потужний мікроконтролер з широкими можливостями, а також є широко розповсюдженим та популярним в наш час. Використання налагоджувальної плати значно спрощує роботу з мікроконтролером, так як на ній вже реалізована необхідна «обв'язка» МК у вигляді резисторів, конденсаторів, зовнішніх кварцових генераторів та інших необхідних для старту та стабільної роботи мікроконтролера компонентів. Також на даній платі наявний завантажувач-налагоджувач ST-Link, який значно спрощує процес відлагодження роботи програми.

Програма керування пристроєм буде реалізована виключно з використанням бібліотеки CMSIS, з реалізацією, таким чином, прямого та послідовного доступу до конфігураційних регістрів керування периферією мікроконтролера. Такий підхід забезпечує максимально точну, після мови асемблеру, а також повністю контрольовану та усвідомлювану розробником, роботу з периферією. Крім цього,

					ДК82.460839.001 ПЗ	Лист
						4
Зм.	Лист	№ докум.	Підпис	Дата		

важливим аргументом на користь використання бібліотеки CMSIS є отримання, при її використанні, максимально оптимального та компактного вихідного коду програми.

В даній роботі будуть розглянуті принципи та процес побудови пристрою, описаного вище, описані та наведені схема електрична принципова та програма керування.

					ДК82.460839.001 ПЗ	Лист
						5
Зм.	Лист	№ докум.	Підпис	Дата		

Розділ 1. Опис структури пристрою та його складових

1.1. Структура пристрою.

Структурна схема пристрою наведена на рисунку 1.1



Рис.1.1. Структурна схема пристрою

Основним компонентом даного пристрою є мікроконтролер STM32F401RE, який розміщений на налагоджувальній платі STM32F401 Nucleo.

Блок керування напрямку руху крокового двигуна являє собою підпружинену кнопку, резистор та конденсатор, позначені на схемі електричній принциповій, як SB1, R1 та C1.

Блок крокового двигуна являє собою сам двигун 28BYJ-48 та драйвер ULN2003, піни якого підключені до STM32.

Блок вимірювання відстані до перешкоди являє собою далекомір, згаданий вище.

					ДК82.460839.001 ПЗ	Лист
						6
Зм.	Лист	№ докум.	Підпис	Дата		

1.2. Принцип роботи блоку керування напрямку руху двигуна

Резистор R1, конденсатор C1 та підпружинена кнопка SB1 являють собою класичну схему керування натисканням кнопки, виклику якої-небудь периферії, викликання переривань, тощо. Резистор підтягнутий до землі, буде формувати лог. нуль на піні мікроконтролеру, допоки не відбудеться натискання кнопки. У випадку, якщо відбудеться натискання кнопки SB1, на резисторі впаде напруга рівна логічній одиниці (3,3 вольти) , мікроконтролер сприйме цей сигнал як переривання, і виконає обробку даного переривання, а саме змінить напрям обертання крокового двигуна. Конденсатор слугує подавлячем так званого “дрязкіту контактів” кнопки.

Класична схема керування натисканням кнопки, зображена на рис.1.2

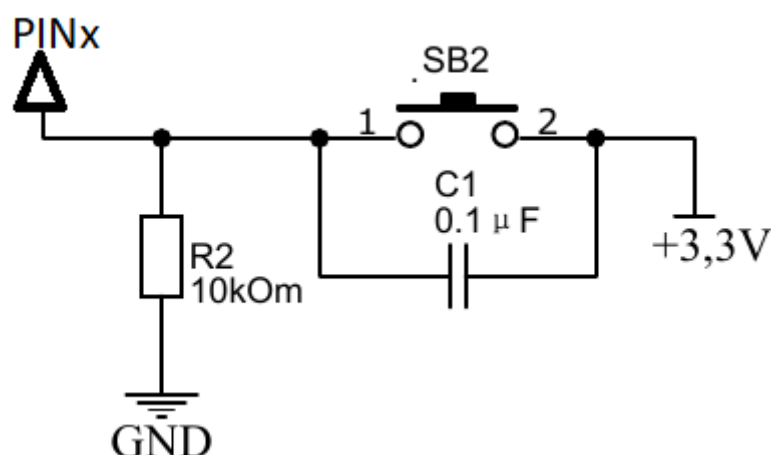


Рис.1.2. Класична схема керування натисканням кнопки

1.3. Принцип роботи блоку крокового двигуна

Мікроконтролер формує лог. рівні на певні піни, що буде приводить до обертання крокового двигуна. Розглянемо принцип роботи крокового двигуна 28BYJ-48 та його драйвера ULN2003.

					ДК82.460839.001 ПЗ	Лист
						7
Зм.	Лист	№ докум.	Підпис	Дата		

Даний двигун має чотири котушки, котрі живляться послідовно, щоб повернути вал з магнітом. Даний двигун може працювати в повнокроковому режимі та напівкроковому режимі.

Коли використовується повнокроковий режим, на кожному кроці живиться 2 котушки з чотирьох.

Що до напівкрокового режиму, спочатку живиться тільки перша котушка, потім разом перша і друга, після тільки друга і так далі. З чотирма котушками це дає 8 різних сигналів, тобто менший кут оберту за 1 крок, що дає можливість максимально точно виставити кут крокового двигуна. На рис. 1.3 зображено порядок сигналів, котрі необхідно подати на котушки, щоб двигун працював в даному режимі, і обертався за годинниковою стрілкою.

Провод	→ Направление вращения по часовой стрелке (1-2 фазы)							
	1	2	3	4	5	6	7	8
4 Оранжевый	■	■						■
3 Желтый		■	■	■				
2 Розовый				■	■	■		
1 Синий					■	■	■	■

рис.1.3. Порядок подачі сигналів живлення котушок, для роботи в напівкроковому режимі двигуна.

Щоб двигун обертався проти годинникової стрілки, необхідно виконувати подачу сигналів живлення котушок в зворотньому порядку рисунку 1.3.

В технічному посібнику до даного двигуна сказано, що кращим являється використання методу напівкроку, тому було використано саме цей режим.

Плата драйверу ULN2003 являє собою масив транзисторів, включених по схемі Дарлінгтона, що дає можливість дуже просто керувати двигуном мікроконтролеру.

В іншому випадку, якщо підключити двигун до мікроконтролеру без даного драйверу, в кращому випадку двигун просто не запрацює, в гіршому –

					ДК82.460839.001 ПЗ	Лист
						8
Зм.	Лист	№ докум.	Підпис	Дата		

мікроконтролер вийде з ладу, так як струм споживання даного двигуна досягає 500 міліампер, а піни портів вводу-виводу не здатні видати струм такої величини.

Маючи характеристики даного двигуна, можна визначити час повного оберту двигуна (360 градусів) в залежності від часу виконання 1 кроку.

Передаточні характеристики двигуна наступні:

Full-stepping

- Internal motor: 32 steps per revolution
- Great reduction ratio: 1/63.68395, approximately 1/64
- Thus, it takes $32 \times 64 = 2048$ steps per revolution for the output shaft

Half-stepping

- Internal motor: 64 steps per revolution
- Great reduction ratio: $1/63.68395 \approx 1/64$
- Thus, it takes $64 \times 64 = 4096$ steps per revolution for the output shaft

Тобто, для повного оберту валу двигуна необхідно виконати 4096 кроків (напівкроковий режим). Нехай необхідно виконувати повний оберт валу за 60 секунд. Тоді період 1 кроку можна розрахувати наступним чином:

$$T(c) = \frac{t(c)}{4096} = \frac{60}{4096} = 0,014(c) = 14(\text{млс}) \quad (1.1)$$

Тобто, формуючи сигнали для двигуна з періодом 14 мілісекунд, вал двигуна виконає повний оберт (360 градусів) за 60 секунд.

1.4. Принцип роботи блоку вимірювання відстані до перешкоди

Даний блок складається лише з далекоміру HC-SR04. Розглянемо принцип його роботи.

При подачі на пін "Trig" імпульсу тривалості 10 мкс. модуль починає генерувати 8 імпульсів з частотою 40 кілогерц, тобто формує ультра-звукову хвилю, котра прямує в напрямку погляду модулю. Після формування імпульсів, модуль встановлює на піні "Echo" лог. одиницю. Хвиля, відбившись від перешкоди і прийшовши назад до модулю, наводить на антені-приймачі напругу. Модуль, зчитуючи дану напругу, скидає лог. 1 на піні "Echo". Виміряв час між відправленням і отриманням імпульсу, не тяжко розрахувати відстань до перешкоди. На рисунку 1.4 зображено як поводить себе сигнал в даному модулі.

					ДК82.460839.001 ПЗ	Лист
						9
Зм.	Лист	№ докум.	Підпис	Дата		

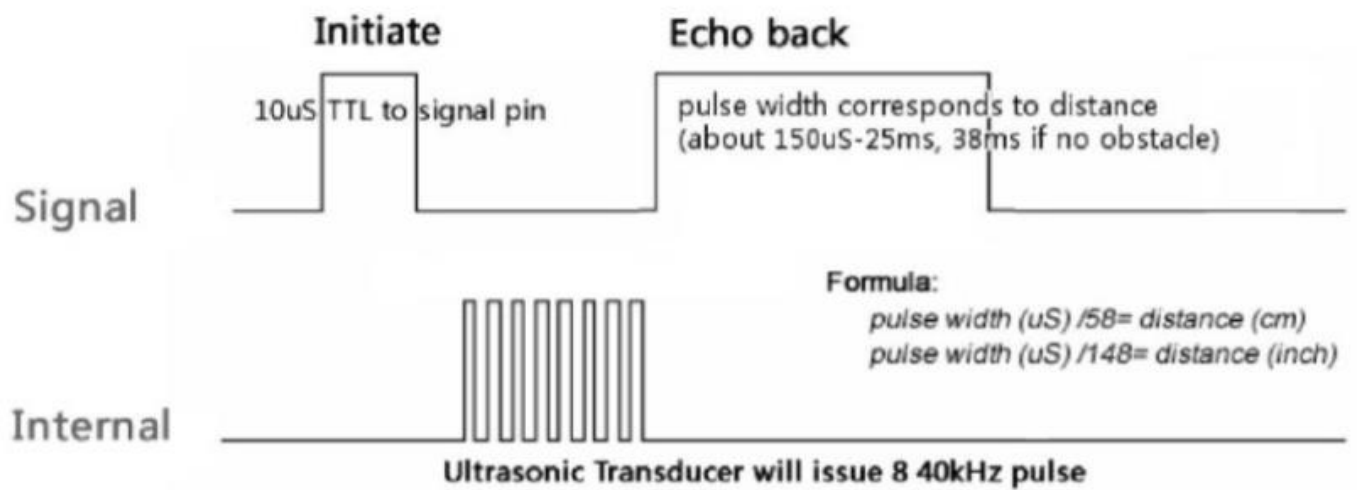


Рис.1.4. Сигнал в модулі HC-SR04

Виведемо формулу для розрахунку відстані до перешкоди.

Швидкість звуку рівна 340 м/с, переведемо її в розмірність см/мкс

$$340 \text{ м/с} = 34/1000 = 0,034 \text{ см/мкс} = \frac{1}{29,4} \left(\frac{\text{см}}{\text{мкс}} \right)$$

Тепер можемо визначити відстань до перешкоди, перемноживши тривалість імпульсу в розмірності мікросекунди, на швидкість, отриману вище. Але, так як ультра-звукова хвиля проходить відстань до перешкоди і назад, тобто в двічі більшу, необхідно поділити розраховану відстань на 2. Кінцева формула для розрахунку набуває наступного вигляду:

$$Distance \approx \frac{Pulse_width(us)}{59} \quad (1.2)$$

					ДК82.460839.001 ПЗ	Лист
						10
Зм.	Лист	№ докум.	Підпис	Дата		

Розділ 2. Проектування схеми електричної принципової

2.1. Проектування схеми електричної принципової

Схема електрична принципова зображена на рисунку 2.1. Схема виконана у відповідності до структурної схеми. В проекті використовуються джерела постійної напруги 3,3 та 5 вольт.

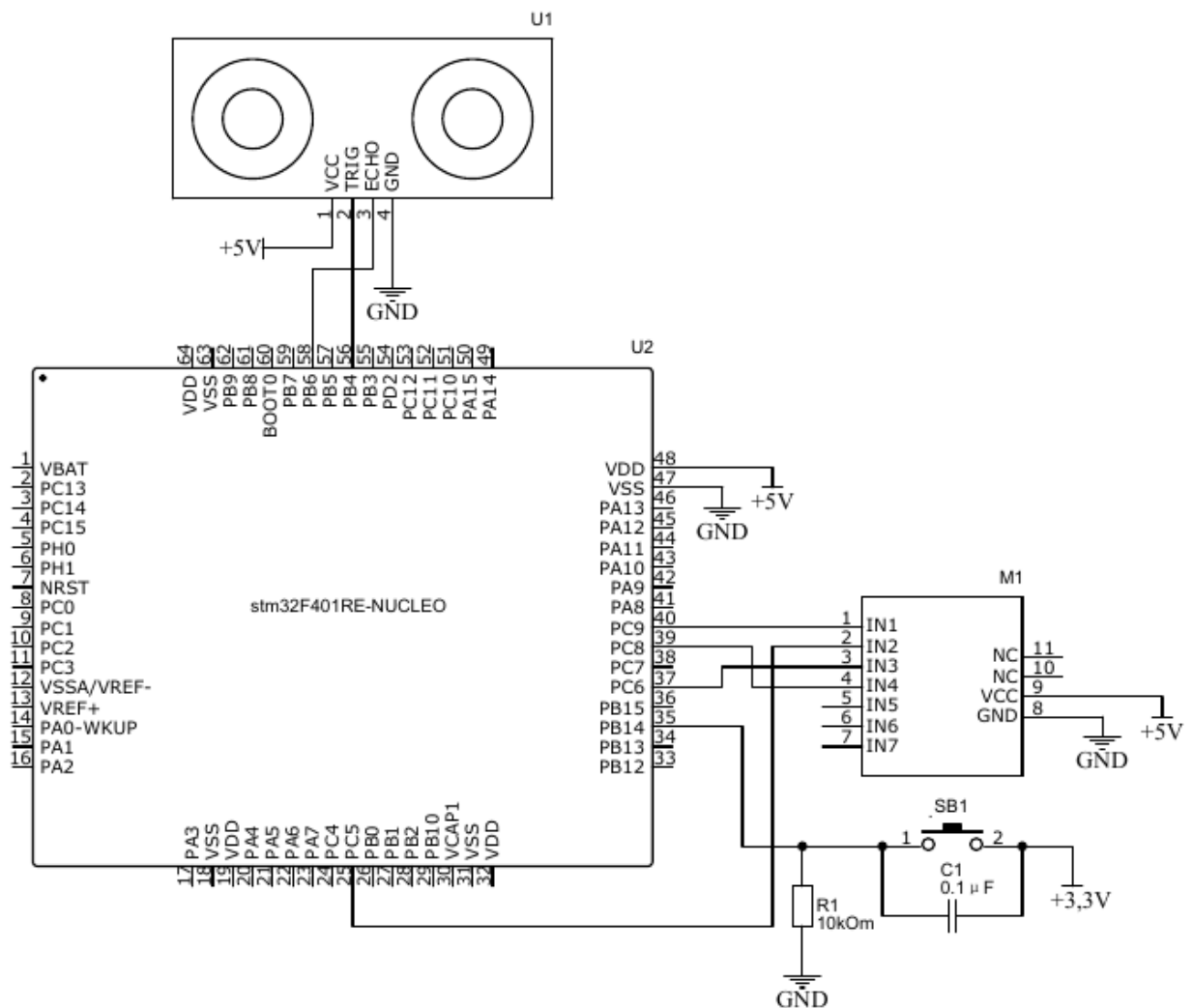


Рис.2.1. Схема електрична принципова

2.2. Вибір елементної бази для реалізації блоків пристрою

Центральною ланкою в пристрої керування двигуна є блок, що оброблятиме дані, отримані з блоку вимірювання відстані до перешкоди, та формуватиме лог. рівні на блок керування двигуна. Згідно технічного завдання, це буде мікроконтролер STM32F401RE. Він ідеально підходить для поставленої задачі.

Мікросхему виконано у корпусі LQFP (рис. 2.2)

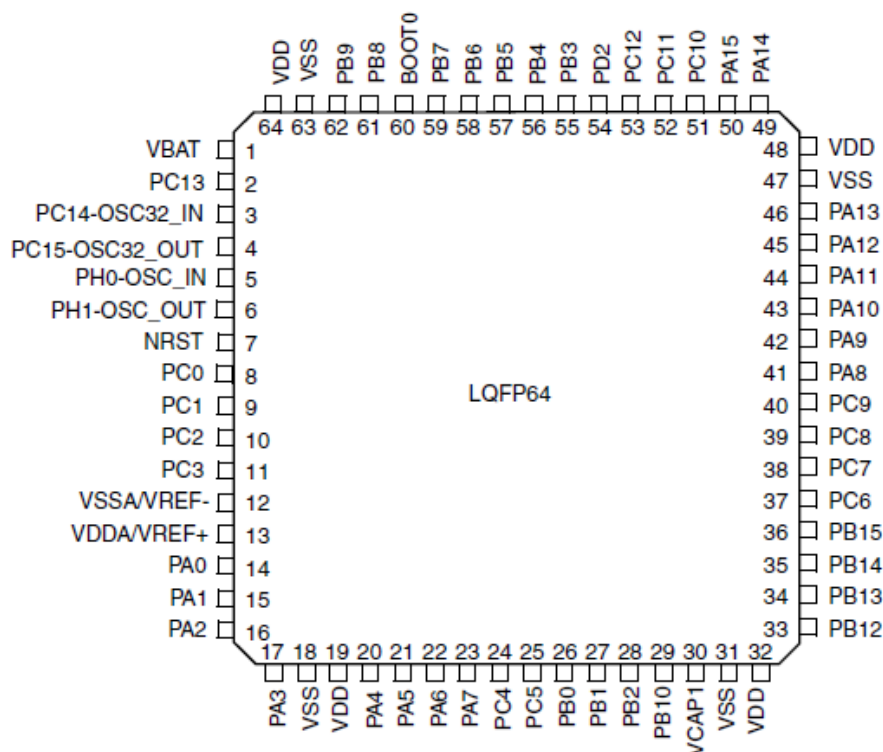


Рис 2.2. мікроконтролер STM32F401RE у корпусі LQFP

Характеристики мікроконтролера:

- Core: ARM® 32-bit Cortex® -M4
- 512 KB Flash пам'ять
- 96 KB SRAM
- Частота вбудованого генератора синхроімпульсів HSI - 16 MHz
- Напруга живлення 1.7 В - 3.6 В

					ДК82.460839.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		12

Блок керування напрямку руху двигуна представлений тактовою кнопкою SB1, керамічним конденсатором C1 та резистором R1. Як видно зі схеми кнопка підтягнута резистором R1 опором 10 кОм до землі, тобто поки не натиснута кнопка, на піні буде сформований логічний нуль.

При складанні прототипу пристрою використовувались тактові кнопки ТАСТ-69К. Вони мають невеликі розміри а також низьку ціну. Позбавлення від «дріб'язку» контактів даної кнопки виконане за допомогою конденсатору C1.

В якості модулю далекоміру, було використано модуль HC-SR04 [1]. в якості крокового двигуна було використано двигун 28BYJ-48 [2] та драйвер, котрий іде в комплекті з даним двигуном ULN2003. Дані модулі найкращі в області “ціна-якість”.

2.3. Налаштування системи тактування мікроконтролера.

Схему налаштування системи тактування мікроконтролера зображено на рисунку 2.3.

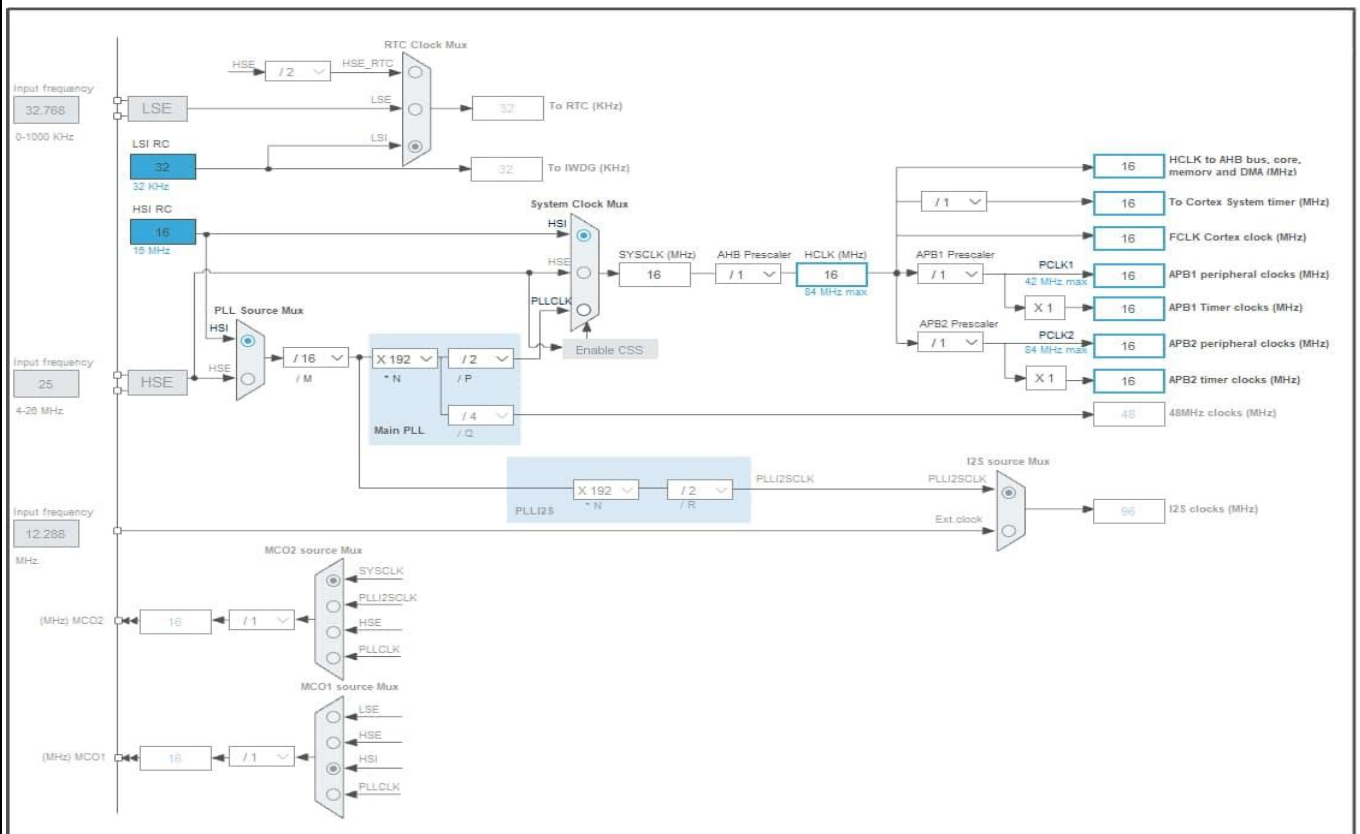


Рис.2.3. Схема налаштування системи тактування мікроконтролера

					ДК82.460839.001 ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		13

Як можна бачити з рисунку, дана конфігурація передбачає роботу мікроконтролера з системною частотою (SYSCLK), рівною 16 МГц. Для цього використовується внутрішній високошвидкісний RC-генератор (HSI — High Speed Internal), який і забезпечує генерацію тактового сигналу з даною частотою. Значення коефіцієнтів переддільників для шин, до яких приєднана периферія, рівні 1, відповідно, системна частота не буде зменшуватись, і шини APB та AHB будуть тактуватись від частоти 16 МГц.

					ДК82.460839.001 ПЗ	Лист
						14
Зм.	Лист	№ докум.	Підпис	Дата		

3. Програма керування пристроєм

3.1. Структура програми

Програма керування пристроєм написана на мові програмування C з використанням бібліотек CMSIS. Виконання програми здійснюється у функції `main()`. Спочатку відбувається ініціалізація змінних, які наведені в табл. 3.1. Після, викликається функція `“init_gpio()”`, в якій здійснюється конфігурація портів введення – виведення. Після викликаються функції `“enable_step_tim2()”` та `“enable_tim_3_4()”` в яких відбувається налаштування таймерів загального призначення а саме, другого, третього та четвертого. Таймер під номером 2, відповідає за керування швидкості крокового двигуна. Таймери під номером 3 та 4 відповідають за формування ШІМ сигналу (формування імпульсу тривалістю 10 мкс.) на пін `“Trig”` далекоміру HC-SR04 та вимірювання тривалості імпульсу на піні `“Echo”` відповідно. Далі налаштовується зовнішнє переривання по тактовій кнопці, викликавши функцію `“irq14_enable()”`. Після скидається прапор переривання за допомогою `__enable_irq()`, що дає можливість виконанню всіх переривань.

Далі в нескінченному циклі відбувається перевірка змінної `“distance”` яка містить в собі значення виміряної відстані до перешкоди, для того щоб зупинити двигун, зупинивши лічбу таймеру під номером 2, якщо відстань до перешкоди буде мізерна (менша 8 сантиметрів). Далі відбувається виклик функції `“drive(&Tacts)”` котра відповідає за формування сигналів на порти введення-виведення, до яких підключений кроковий двигун, тобто приводить двигун в рух.

					ДК82.460839.001 ПЗ	Лист
						15
Зм.	Лист	№ докум.	Підпис	Дата		

3.2. Структура даних

Таблиця 3.1. Використані змінні

Назва	Тип змінної	Призначення
Step_dir	volatile int	Напрямок руху двигуна
Tacts	volatile int	Формування кроків двигуна
Pulse_width	volatile unsigned int	Тривалість імпульсу на піні “Echo”
current_captured	volatile unsigned int	Збереження значення лічильника при останньому фронті на піні “Echo”
last_captured	volatile unsigned int	Збереження значення лічильника при попередньому фронті на піні “Echo”
signal_polarity	volatile unsigned int	Розпізнання фронту
distance	volatile unsigned int	Збереження відстані до перешкоди

Таблиця 3.2. Використані пінні портів введення-виведення

Назва	Призначення
PC9	Комутація виводу IN1 двигуна M1 з мікроконтролером
PC5	Комутація виводу IN2 двигуна M1 з мікроконтролером
PC6	Комутація виводу IN3 двигуна M1 з мікроконтролером
PC8	Комутація виводу IN4 двигуна M1 з мікроконтролером
PB14	Комутація кнопки SB1 з мікроконтролером
PB4	Комутація виводу “Trig” далекоміру з мікроконтролером (Вихід TIM3_CH1)
PB6	Комутація виводу “Echo” далекоміру з мікроконтролером (Вхід TIM4_CH1)

Таблиця 3.3. Регістри мікроконтролера, застосовані в програмі

Назва	Призначення
AHB1ENR	Ввімкнення тактування портів GPIOB та GPIOC
APB1ENR	Дозвіл тактування таймерів TIM2_3_4
APB2ENR	Дозвіл тактування SYSCFG
GPIOx->MODER	Налаштування роботи виводів портів B та C
GPIOx->AFR	Встановлення потрібної альтернативної функції для PB4,PB6
GPIOx->PUPDR	Налаштування підтяжки виводів
GPIOC->ODR	Формування сигналів для двигуна
SYSCFG->EXTICR	Налаштування мультиплексору EXTI
EXTI->IMR	Маскування зовнішніх переривань
EXTI->RTSR	Налаштування активного фронту зовнішнього переривання
EXTI->PR	Перевірка на виконання переривання
NVIC_SetPriority(*IRQn, Priority)	Встановлення пріоритетів обробників переривань
NVIC_ClearPendingIRQ(*IRQn)	Очищення флажка черги переривання
NVIC_EnableIRQ(*IRQn)	Ввімкнення обробника переривання
TIM2_3_4->CR1	Налаштування таймеру 2(cnt_dir, en_shadow_reg., en_cnt)

TIM2_3_4->ARR	Значення, до якого рахуватиме лічильник, після - скидання
TIM2->CNT	Зміна вмісту лічильника
TIM2_3_4 ->PSC	Налаштування пред-дільника сигналу тактування лічильника
TIM2_4->SR	Перевірка, чи виконалось потрібне переривання
TIM2->DIER	Ввімкнення переривання по переповненню
TIM3->CCMR1	Налаштування режиму роботи каналу (PWM1), перезапис вміст CCR1 лише після встановлення UEV flag
TIM3->CCER	Налаштування полярності виходу каналу, дозвіл роботи каналу
TIM3->BDTR	Ввімкнення головного виходу каналу
TIM3->CCR1	Зміна вмісту регістру Capture_compare каналу №1
TIM4->CCMR1	Налаштування режиму роботи каналу на вхід, ввімкнення фільтру, преддільника каналу
TIM4->CCER	Налаштування активного фронту каналу, дозвіл роботи каналу
TIM4->DIER	Дозвіл виконання переривання при приході акт. фронту на канал

3.3. Опис алгоритмів окремих підпрограм

3.3.1. Опис роботи функції init_gpio();

В даній ф-ї відбувається включення тактування системної шини АНВ1 та портів GPIOB та GPIOC. Для цього необхідно налаштувати регістр АНВ1ENR з блоку регістрів RCC (Reset and Clock Control). Біт 2 (GPIOCEN) та біт 1 (GPIOBEN) необхідно встановити в 1. Далі йде конфігурування PC9, PC5, PC6 та PC8 на вихід. Для цього у регістрі GPIOC(MODER) з блоку регістрів GPIO необхідно біти 18 та 19 (MODER9), 10 та 11 (MODER5), 12 та 13 (MODER6), 16 та 17 (MODER8), скинути до 0. Далі йде конфігурація PB4 та PB6 на альтернативний режим. Дані виводи підключені до пінів “Trig” та “Echo” відповідно. Необхідно під’єднати PB4 та PB6 до таймерів. Для цього у регістрі GPIOB(AFR) з блоку регістрів GPIO необхідно встановити відповідні біти. На рисунку 3.1. зображено розподіл альтернативних функцій, по якій можна побачити, який біт в регістрі (AFR) необхідно встановити, щоб під’єднати до бажаного виводу певну периферію.

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			
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_S D	-	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/I 2S2_WS	-	-	-	-	-	-	SDIO_D5	-	-	EVENT OUT
PB10	-	TIM2_CH3	-	-	I2C2_SCL	SPI2_SCK/I 2S2_CK	-	-	-	-	-	-	-	-	-	EVENT OUT
PB12	-	TIM1_BKIN	-	-	I2C2_SMBA	SPI2_NSS/I 2S2_WS	-	-	-	-	-	-	-	-	-	EVENT OUT
PB13	-	TIM1_CH1N	-	-	-	SPI2_SCK/I 2S2_CK	-	-	-	-	-	-	-	-	-	EVENT OUT
PB14	-	TIM1_CH2N	-	-	-	SPI2_MISO	I2S2ext_SD	-	-	-	-	-	-	-	-	EVENT OUT
PB15	RTC_REFN	TIM1_CH3N	-	-	-	SPI2_MOSI/ I2S2_SD	-	-	-	-	-	-	-	-	-	EVENT OUT

Рис.3.1. Розподіл альтернативних функцій

Бачимо, що щоб під'єднати PB4 та PB6 до каналів таймеру TIM3_CH1 та TIM4_CH1 необхідно встановити біт AF02.

Так як канал таймеру TIM3_CH1 буде налаштований на вихід, і буде формувати ШІМ сигнал, необхідно вимкнути підтяжку для відповідного виводу, а саме PB4. необхідно звернутись до регістру GPIOB->PUPDR , і скинути біт 8 та 9.

3.3.2. Опис функції enable_step_tim2():

В даній ф-ї відбувається налаштування таймеру 2, котрий буде керувати швидкістю обертання двигуна. Було вирішено реалізувати наступний метод:

Налаштувати лічильник так, щоб він збільшував своє значення з частотою 1 кілогерц. Отримуючи значення від далекоміру, змінювати вміст регістру ARR за наступною формулою.

$$TIM2->ARR = 30 - (distance * 30 / 400); \quad (3.1)$$

Таким чином, отримуючи значення дистанції ми можемо масштабувати дане значення і змінювати вміст регістр ARR. У випадку якщо відстань до перешкоди буде досить великою (макс. знач. котре підтримує далекомір 400 см) , нехай distance = 300. В такому випадку в ARR запишеться значення 7,5 == 7. Тобто кожні 7 мілісекунд (7 – тому що таймер рахує з частотою 1 КГц) буде виконуватись обнулення лічильнику і виконуватись переривання (виконалась подія*). В обробнику переривання відбуватиметься інкрементування або декрементування, в залежності від напрямку руху двигуна, змінної Tacts, що відповідає за формування кроків двигуна. Тобто змінюючи значення ARR, ми керуємо частотою виклику обробника переривань, тобто частотою інкрементування та декремент. змінної Tacts, що в свою чергу приводить до збільшення швидкості обертання двигуна (якщо ARR прямує до нуля) або зменшення (якщо ARR прямує до 30).

В подальшому, номери бітів, котрі необхідно встановити чи скинути, не будуть перелічені, так як це займає багато місця, всі біти можна переглянути в Reference Manual-i.

					ДК82.460839.001 ПЗ	Лист
						21
Зм.	Лист	№ докум.	Підпис	Дата		

Перш за все виконується ввімкнення тактування таймеру 2, встановленням відповідного біту в регістрі APB1ENR .В регістрі TIM2->CR1 налаштовуються напрямки рахунку лічильнику вгору, вмикається “shadow register”, котрий не дає можливості одразу змінити вміст регістру ARR. Далі налаштовується регістр преддільник PSC, щоб сформувати частоту тактування лічильнику рівним 1 Кілогерц. Формула розрахунку значення регістру PSC наступна:

$$\text{Freq_count} = (\text{Freq_PSC} / \text{PSC}^* + 1) = 16\text{MHz} / (15999 + 1) = 1\text{KHz} \quad (3.2)$$

$\text{Freq_PSC} = 16 \text{ MHz}$ – частота шини APB1

$\text{PSC}^* =$ вміст регістру PSC

Freq_count – частота на виході преддільника, тобто частота тактування лічильника.

Тобто, щоб сформувати частоту тактування 1 КГц, записуємо в PSC 15999.

Далі налаштовується стартове значення регістру ARR та CNT , було вирішено записати 50 та 0 відповідно.

Далі відбувається ввімкнення переривання UEV в регістрі TIM2->DIER. Таким чином при кожному переповненні лічильника, відбуватиметься виклик обробника переривань.

Далі дозволяється робота лічильника, записом 1 в відповідний біт регістру CR1. Далі налаштовується пріоритет обробника переривань, після встановлюється флажок черги (Pending), що дає можливість обробляти переривання.

3.3.3. Опис функції `enable_tim_3_4 ()` :

В даній ф-ї відбувається налаштування таймеру 3 та 4, котрий буде формувати ШІМ та вимірювати тривалість імпульсу відповідно. Було вирішено реалізувати наступний метод:

Для того щоб сформувати імпульс тривалістю 10 мкс, вирішено налаштувати таймер 3 наступним чином:

					ДК82.460839.001 ПЗ	Лист
						22
Зм.	Лист	№ докум.	Підпис	Дата		

Вмикаємо тактування таймеру, встановивши відповідний біт в регістрі APB1ENR. Далі налаштовуємо таймер на рахунок вверх , так як і з таймером 2.

Записуємо в регістр PSC значення 159 , тобто частота тактування лічильнику буде рівна 100 КГц , тобто період 10 мікросекунд. Записуємо в ARR максимально можливе значення $((2 \text{ в степені } 16) - 1)$. Таким чином період ШІМ буде рівний 0,65 секунд. Насправді на виході не буде формуватись ШІМ, так як період рівний 0,65 секунд дуже велике значення для ШІМ. На виході отримаємо сигнал дуже великої скважності, і імпульс тривалістю 10 мкс, тобто в системи буде запас $(0,65\text{с} - 10\text{мкс})$ секунд, щоб отримати імпульс на вхідний канал таймеру 4, до формування наступного імпульсу таймером 3.

Далі налаштовуємо регістр CCMR1, на режим PWM1. В даному режимі шим буде формуватись наступним чином: поки вміст лічильнику менший за CCR1, на виході буде логічна 1. В іншому випадку, лог. нуль. Якраз те що потрібно. Також в даному регістрі вмикається shadow_reg для регістру CCR1, тобто не можна буде змінити вміст CCR1 , допоки не встановиться UEV флаг.

Далі вмикаємо Shadow_reg для регістру ARR (розглядали вище) налаштовуючи регістр CR1. Налаштовуємо регістр CCER, в ньому встановлюємо біти, які відповідають за вихідну полярність (якщо 1 – сигнал інвертується , якщо 0 – не інвертується) , тобто необхідно скинути необхідний біт. В цьому ж регістрі вмикаємо вихідний канал 1. Далі в рег. BDTR вмикаємо головний вихідний канал. Після записуємо в CCR1 одиницю, таким чином лічильник порахує до 1, тобто сформує імпульс 10 мкс, і далі рахуватиме до 65535, скинувши вихід до лог. 0. Після вмикаємо рахунок лічильнику. Таким чином сформований ШІМ сигнал – сигнал для виводу Trig модулю HC-SR04.

Далі виконується налаштування таймеру 4

Записується в PSC значення 15, тобто частота лічби рівна 1 МГц. Записується 0xFFFF в рег. ARR. Налаштовується рахунок вверх, налаштовується регістр TIM4->CCMR1 на вхід. В цьому ж регістрі вимикається фільтр та преддільник каналу. В

					ДК82.460839.001 ПЗ	Лист
						23
Зм.	Лист	№ докум.	Підпис	Дата		

реєстрі CCER налаштовується активний фронт а саме передній та задній (both). Вмикається канал 1. Після в реєстрі DIER вмикається дозвіл переривання по приходу активного фронту на вхід каналу. Після вмикається таймер, виставляється пріоритет обробника переривання даного каналу, вмикається обробника переривання.

Таким чином, коли на вхід каналу приходить імпульс (передній фронт при початку генерації ультра-звукової хвилі) відбувається запис в CCR1 значення лічильнику, і викликається переривання. Обробник переривання виконує обрахунки. Після приходу заднього фронту, знову викликається переривання. Обробник на даному етапі розраховує значення тривалості імпульсу. Робота обробника буде представлена далі.

3.3.4. Опис функції обробки переривання “IM4_IRQHandler()”

Алгоритм наступний:

Якщо виконалось переривання саме від TIM4_CH1, тобто встановивсь флаг TIM4->SR_CC1IF тоді необхідно записати в current_captured зміну вміст CCR1.

Далі інвертуємо зміну, яка відповідає за полярність, щоб розрізняти який фронт саме прийшов.

Далі перевіряємо , якщо прийшов задній фронт то виконати наступне:

Якщо Cur_capt > Last_Capt тоді виконуємо:

Розрахувати тривалість імпульсу , виконавши різницю Cur_capt – Last_capt

Після розрахувати відстань до перешкоди за формулою 1.4.

Якщо Cur_capt < Last_Capt , тобто відбулось переповнення лічильнику тоді виконуємо:

Розрахувати тривалість імпульсу , виконавши різницю (0xFFFF - last_capt) + current_capt

					ДК82.460839.001 ПЗ	Лист
						24
Зм.	Лист	№ докум.	Підпис	Дата		

Після розрахувати відстань до перешкоди за формулою 1.4.

Далі виходимо з даних блоків, і записуємо в змінну last_capt вміст змінної current_capt.

Таким чином, ми зможемо отримати значення відстані, навіть якщо відбулось переповнення лічильнику.

3.3.5. Опис функції обробки переривання “IM2_IRQHandler()”

Даний обробник буде викликатись якщо виконалось переривання від таймера №2, який відповідає за керування швидкістю двигуна. логіку роботи ми розглядали вище. Все що необхідно зробити, то це наступне:

Перевірити чи відбулося саме те що потрібне переривання

Після перевірити змінну Step_dir, котра відповідає за реверс двигуна, і в залежності від неї виконувати інкрементування або декрементування змінної Tacts.

Далі виконати масштабування та запис в регістр TIM2->ARR за формулою 3.1. Таким чином ми керуємо швидкістю обертання двигуна.

3.3.6. Опис функції обробки переривання EXTI15_10_IRQHandler ():

Даний обробник переривання викликається при натисканні зовнішньої кнопки SB1. Алгоритм обробника переривання наступний:

Перевіряємо чи виконалось саме переривання по лінії PB14

Після маскуємо переривання, скинувши біт в регістрі EXTI->PR

Виконуємо інвертування змінної Step_dir

Дозволяємо обробку переривання , встановивши одиницю в регістрі EXTI->PR

Розмасковуємо переривання по лінії PB14

					ДК82.460839.001 ПЗ	Лист
						25
Зм.	Лист	№ докум.	Підпис	Дата		

3.3.7. Опис функції керування двигуном “drive(&Tacts)”

Дана функція дуже проста. Вона формує лог. рівні в регістрі GPIOB->ODR згідно рисунку 1.4. в залежності від значення змінної Tacts, котра передається в якості аргументів функції. Також ця функція виконує виправлення змінної Tacts:

Якщо Tacts > 7 , то скинути її до 0

Якщо Tacts < 0 , записати в неї 7

Тому дану змінну необхідно передавати по адресі, а не як значення, щоб ми могли змінити вміст самої змінної.

					ДК82.460839.001 ПЗ	Лист
						26
Зм.	Лист	№ докум.	Підпис	Дата		

Висновок

В даній розрахунковій-графічній роботі було реалізовано пристрій керування швидкістю обертання крокового двигуна, швидкість якого змінювалась в залежності від відстані до перешкоди, яку вимірював далекомір HC-SR04.

Було виконано прототип пристрою на безпаячній макетній платі. Результати вийшли досить не поганими. Використовуючи далекомір HC-SR04, було отримано відстань з похибкою ± 3 сантиметри, що досить не погано. Звичайно, на відстані більше 100 сантиметрів, не варто чекати точних результатів, але для стартового прототипу це вже результат. Також варто враховувати опір провідників, поганий контакт на макетній платі і т.п. Двигун працює, швидкість змінюється. За необхідності можна поставити більш потужніший двигун, наприклад такий же ж двигун моделі 28BYJ але на 9 вольт, якщо в цьому є необхідність.

Виконуючи дану розрахункову роботу, було набуто навички роботи з мікроконтролером STM32, а саме робота з таймерами глобального призначення, використання різних режимів їхньої роботи : ШІМ режим, режим вхідного каналу, при якому можна вимірювати тривалість імпульсів, робота з різними перериваннями (зовнішні, переривання від виконання подій такі як переповнення лічильника, надходження активного фронту на вхід каналу таймеру, тощо). Набуто навички роботи з фізичною периферією, далекоміри, крокові двигуни, кнопки.

					ДК82.460839.001 ПЗ	Лист
						27
Зм.	Лист	№ докум.	Підпис	Дата		

Список використаних джерел

1. Модуль HC-SR04 [електричний ресурс] – Режим доступу:
<https://arduino-master.ru/datchiki-arduino/ultrazvukovoj-dalnomer-hc-sr04/>
2. Двигун 28BYJ-48 [електричний ресурс] – Режим доступу:
<https://arduino.ua/prod216-shagovii-dvigatel-5v-28byj-48>
3. Документація до мікроконтролеру STM32F401RE [електричний ресурс] –
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
4. Інструмент для налаштування тактування мікроконтролеру STM32F401RE —
Clock configuration tool for STM32F40x/41x microcontrollers (AN3988)/
[Електронний ресурс] — Режим доступу:
https://my.st.com/content/my_st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-configurators-and-code-generators/stsw-stm32091.html

					ДК82.460839.001 ПЗ	Лист
						28
Зм.	Лист	№ докум.	Підпис	Дата		