

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

КАФЕДРА КОНСТРУЮВАННЯ ЕОА

РОЗРАХУНКОВО-ГРАФІЧНА РОБОТА

за дисципліною: Обчислювальні та Мікропроцесорні засоби в РЕА
тема: «Далекомір»

Керівник:

доц. Корнєв В.П.

Допущено до захисту

“__” _____ 20__ р.

Захищено з оцінкою

Виконав:

Юрченко В.О.

Студент 3 курсу ФЕЛ

Групи ДК-81

Національний Технічний Університет України

“Київський Політехнічний Інститут

імені Ігоря Сікорського,,

Кафедра Конструювання електронно-обчислювальної апаратури

Дисципліна Обчислювальні та мікропроцесорні засоби в РЕА

Спеціальність 172 Телекомунікації та радіотехніка

Курс 3 Група ДК-81 Семестр VI

ЗАВДАННЯ

до розрахунково-графічної роботи

Юрченко Віталій Олегович

(прізвище, ім'я та по батькові)

1. Тема проекту Далекомір
2. Строк здачі студентом закінченого проекту (роботи) 23.05.2021
3. Вихідні данні до проекту (роботи)

Спроекувати пристрій який обраховує відстані до об'єкту. Керування здійснювати за допомогою однієї кнопки

4. Зміст розрахунково-пояснювальної записки (перелік питань, що розроблюються)
 1. Опис структури пристрою і його складових
 2. Обґрунтування вибору елементної бази
 3. Опис і розрахунок схеми електричної принципової
 4. Алгоритм роботи програми
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Схема електрична принципова
2. Перелік елементів
6. Дата видачі завдання 31.03.2021

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапу роботи	Строк виконання етапів роботи	Примітка
1	Пошук літератури та аналіз існуючих рішень. Розробка технічного завдання	01.04 – 7.04	
2	Розділ 1. Розробка та опис структури пристрою і його окремих складових	8.04 – 14.04	
3	Розділ 2. Обґрунтування вибору елементної бази.	15.04 – 20.04	
4	Створення схеми електричної принципової та узгодження її з керівником	15.04 – 30.04	
6	Опис і розрахунок схеми електричної принципової	03.05 – 10.05	
7	Розділ 3. Створення структури програмного забезпечення пристрою та алгоритмів роботи програм.	20.04 – 30.04	
8	Розробка програмного забезпечення пристрою.	03.05 – 16.05	
7	Розділ 4. Інструкція користувача. Опис контрольного прикладу	17.05 – 19.05	
8	Оформлення документації: ЕЗ, ПЕЗ, ПЗ	20.05 – 22.05	
9	Подання до захисту	23.05	
10	Захист РГР	26.05 – 27.05	

Студент _____
(підпис)

Керівник _____
(підпис)

«__» ____ 2021 р.

Зміст

Вступ.....	6
Розділ 1. Структурна схема пристрою та принцип роботи.....	8
1.1. Структурна схема пристрою.....	8
1.2. Засоби керування.....	8
1.3. Принципи і засоби введення даних.....	8
1.4. Принципи і засоби відображення вихідних даних і результату.....	9
1.4.1 Інтерфейс I2C.....	9
1.4.2 Принцип взаємодії з дисплеєм.....	12
1.5. Принципи і засоби обробки даних.....	14
1.5.1 Мікроконтролер.....	14
1.5.2 Лічильники.....	15
Розділ 2. Схема електрична принципова.....	19
2.1. Проектування блоку обробки даних.....	19
2.2. Проектування блоку контролю.....	20
2.3. Проектування блоку введення даних.....	20
2.4. Проектування блоку відображення вихідних результату.....	20
2.5 Вибір елементної бази.....	20
2.5.1. Конденсатори.....	20
2.5.2. Резистори.....	21
2.5.3. Кнопки.....	21
2.5.4 Мікроконтролер.....	21
2.5.5. Роз'єми.....	22
2.5.6. Далекомір.....	23
2.5.7. Дисплей.....	23
2.5.8. Котушки індуктивності.....	23

					ДК81.460839.001 ПЗ				
Зм.	Арк.	№ докум.	Підп.	Дата	Далекомір	Лім.	Арк.	Аркушів	
Розробив	Юрченко В.О.					0	4	40	
Перевірив	Корнєв В.П.								
Н. контр.									
Затв.	Корнєв В.П.					КПІ ім. Ізгоря Сікорського ФЕЛ, ДК-81			

Розділ 3. Програма керування пристроєм.....	24
3.1. Структурна схема програми.....	24
3.2. Опис основної програми.....	25
3.3. Структура даних.....	26
3.4. Опис алгоритмів окремих підпрограм.....	26
3.4.1 Опис роботи таймерів TIMx.....	26
3.4.2 Опис роботи Кнопки.....	28
3.4.3 Опис роботи функції HC_SR04.....	29
3.4.4 Опис роботи I2C.....	30
3.4.5 Опис роботи LCD.....	32
3.4.6 Опис перетворення числа в ASCII.....	34
3.4.7 Опис функції FUNC_Z.....	35
Висновок.....	36
Література.....	37
ДОДАТОК А Технічне завдання на проектування.....	39

Вступ

Далекомір — це прилад для вимірювання відстаней до об'єктів.

За принципом дії далекоміри. поділяються на дві основні групи:

- Оптичні далекоміри
- Акустичні далекоміри

Оптичні далекоміри працюють по принципу задачі про рівнобедрений трикутник, у якому відомі основа, (база), та протилежний, (т.зв. паралактичний), кут. Розрізняють оптичні далекоміри візуальні, світло проєкційні і проєкційно-візуальні. Найбільш поширеними оптичними візуальними далекомірами є: ниткові, монокулярні подвійного зображення з рейками і внутрішньобазові, зі змінним і постійним паралактичним кутом, утвореним оптичним компенсатором; стереоскопічні, засновані на використанні стереоскопічного ефекту, що виникає при розгляданні предметів двома очима.

Акустичні далекоміри, радіодалекоміри, електрооптичні далекоміри Принцип дії таких далекомірів полягає у порівняльній оцінці сигналів, які передаються до об'єкта і ехо-сигналів.

Оптичні далекоміри точніші за акустичні. В даній роботі використовується акустичний далекомір, тому що завдання не вимагає мати точні виміри.

Далекомір потрібно контролювати для цього використовують мікроконтролер. Мікроконтролери можна зустріти в багатьох сучасних приладах, вони відповідають за роботу двигунів і систем гальмування сучасних автомобілів, з їх допомогою створюються системи контролю і системи збору інформації.

Мікроконтролер — виконаний у вигляді мікросхеми спеціалізована мікропроцесорна система, що включає мікропроцесор, блоки пам'яті для збереження коду програм і даних, порти вводу-виводу і блоки зі спеціальними функціями (лічильники, компаратори, АЦП та інші). Використання однієї мікросхеми значно знижує розміри, енергоспоживання і вартість пристроїв, побудованих на базі мікроконтролерів. Одна із популярних лінійок мікроконтролерів є STM32, а саме сімейства мікропроцесорних ядер ARM-Cortex-M.

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

Розділ 1. Структурна схема пристрою та принцип роботи

1.1. Структурна схема пристрою

На рис 1.1 зображено структурну схему пристрою, яка складається з блоку введення даних, блок обробки даних і блок керування.

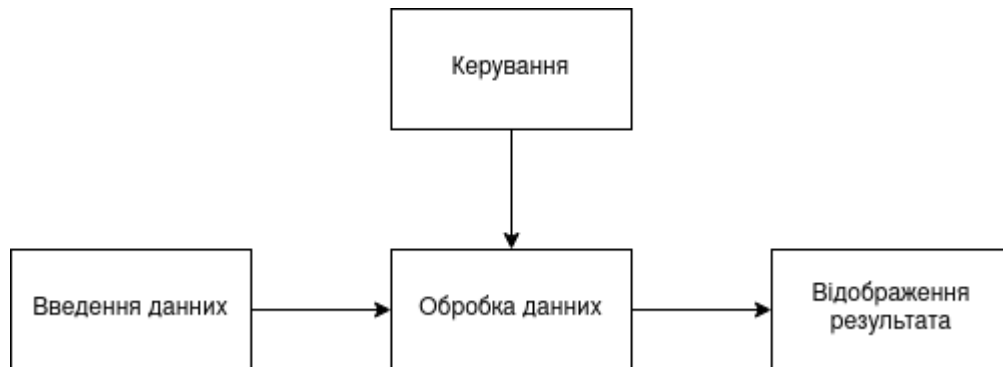


Рисунок 1.1 - Схеми структурна

1.2. Засоби керування

Керування відбувається за допомогою кнопки. Кнопку можна запрограмувати на різні види натискань одинарне натискання, подвійне натискання, затиснення. Тому для фіксування даних і відображення результатів блока обробки даних використовується одна кнопка.

1.3. Принципи і засоби введення даних

Введення відстані до об'єкта відбувається за допомогою ультразвукового датчика HC-SR04.

На рисунку 1.2 зображено послідовність сигналів на вході «Trig» і виході «Echo» далекомір. Для того щоб запустити далекомір треба подати високий логічний рівень тривалістю 10 мкс на його вхід «Trig». Далекімір відправить пучки височастотного сигналу і сформує на виході «Echo» логічний одиницю. Після того як сигнал повернеться сформує логічний нуль. Тривалість цього імпульсу визначає відстань d до об'єкта і обраховується за формулою (1.3.1). Так визначається час слідування звукової хвилі від

далекоміра до перешкоди і назад у мікросекундах. Сигнал проходить подвійну відстань тому потрібно ділити на два.

$$d = \frac{t \cdot V}{2} = \frac{t}{58}, \quad (1.3.1)$$

де

d – відстань до об'єкта, см;

t – тривалість високого логічного рівня, мкс;

V — швидкість звуку, $\frac{\text{см}}{\text{мкс}}$.

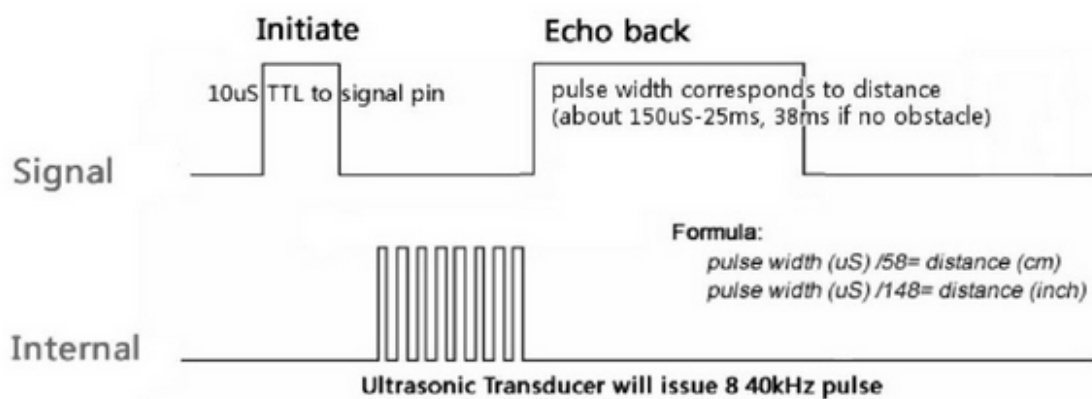


Рисунок 1.2 — Принцип роботи далекоміра

1.4. Принципи і засоби відображення вихідних даних і результату

1.4.1 Інтерфейс I2C

Дисплей підключений до блоку обробки даних через інтерфейс I2C. Зображений на рисунку 1.3. Інтерфейс має шину даних “SDA” і тактовий сигнал “SCL” який формує пристрій Master. Не будемо розглядати розширені версії інтерфейсу, а лише стандартний. На I2C шину можна під’єднати до 128 пристроїв. Передача відбувається по байтово. Більшість пристроїв підтримують до 100 МГц, але можуть сягати до 400 МГц.

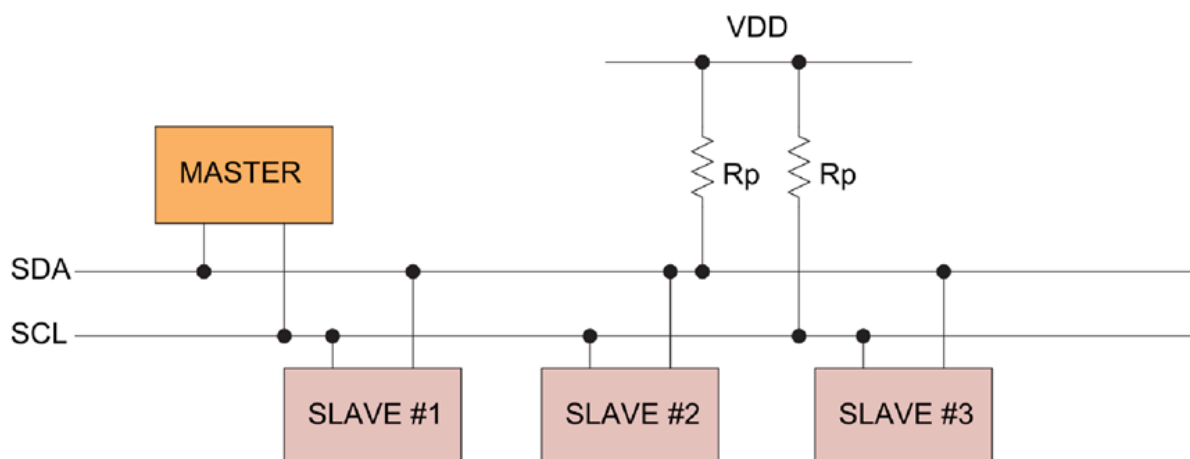


Рисунок 1.3 — I2C інтерфейс

Розберемося як відбувається робота на рівні пристроїв. Master першим байтом відправляє адресу Slave пристрою і режим роботи (читання або запис). Причина в обмеженні 128 пристроїв криється в тому що на адресу виділяється 7 біт. Після передачі байту відбувається підтвердження Slave пристроєм. Надалі відбувається передача інформації Master з вибраним Slave пристроєм поки не відбудеться сигнал зупинки. Щоб змінити пристрій потрібно завершити роботу з теперішнім і почати нову.

Старт це сигнал для Slave пристроїв. Master встановлює логічний нуль(задній фронт) коли дріт SCL знаходиться на високому рівні. Після цього всі Slave пристрої чекають отримання адреси. Після отримання порівнюють зі власною адресою, у випадку збігу формують біт підтвердження.

Особливо хочу уділити увагу на формування біта підтвердження. По правилам схемотехніці маючи два з'єднаних пристроя їхні входи повинні бути високоомні, а виходи низькоомним. Чим краще будуть ці характеристики тим краща передача по напрузі для збереження логічних рівнів. Але маючи багато пристрої на одній шині скорегувати всі пристрої не можливо. І I2C інтерфейс не має контролю пристрої. Якщо два пристрої буду працювати як виходи один буде формувати логічний нуль, а інший логічну одиницю відбудеться коротке замкнення що є не допустимим. І щоб цього не

допустити шини інтерфейсу підтягнуті до напруги живлення через резистори. В основному номінали цих резисторів 1 кОм — 4,7 кОм. А також піни пристроїв працюють в режимі (Open drain). При якому можливе формування логічного нуля або високоімпедансного стану (третій стан), але через підтяжки до напруги живлення при високоімпедансному стані формується логічна одиниця. Завдяки цьому не відбувається короткого замкнення. Саме підтвердження створюється формування передавача високоімпедансного стану, а отримувач на шині даних формує логічний нуль. Якщо відбувається запис підтверджує Slave пристрій, якщо читання підтверджує Master.

Після підтвердження байту надсилається наступний або Master завершує передачу даних формування стоп сигналу. Master формує логічну одиницю (передній фронт) коли дріт SCL знаходиться на високому рівні.

Налаштування I2C модуля в STM32F401RE

Задаємо периферійну тактову частоту $FREQ$ [5:0] в другому контрольному регістрі $I2C_CR2$. Значення повинні бути в межа 2-50 MHz

Встановлюємо значення максимального часу підйому в режимі Fm / Sm в регістрі $I2C_TRISE$. Обраховується за формулою 1.4.1.

$$TRISE = \frac{T_{r(SCL)}}{T_{PSLK1}} + 1, \quad (1.4.1)$$

де, $T_{r(SCL)}$ — максимальний час зростання SDA та SCL;

T_{PSLK1} — $1/FREQ$.

Регістр управління годинником ($I2C_CCR$) обраховується за формулою:

$$CCR = \frac{T_{r(SCL)} + T_{w(SCLH)}}{T_{PSLK1}}, \quad (1.4.2)$$

де $T_{r(SCL)}$ — максимальний час зростання SDA та SCL;

$T_{w(SCLH)}$ — мінімальний час верхньої полиці SCL.

$T_{PSLK1} = 1/FREQ$

Запускаємо модуль

1.4.2 Принцип взаємодії з дисплеєм

Відображення відстані відбувається за допомогою LCD 1602 дисплея. Інформація у форматі ASCII символів або інструкцій передаються через шину на дисплей. На дисплеї окремо прикріплена друкована плата з чипом PCF8574 це простий послідовно-паралельний перетворювач. Він встановлює свої вісьмох пінах відповідно до восьми отриманих біт. При читання надсилає вісім значень відповідно до значень на його восьми пінах. Схема електрична принципова друкованої плати з інтерфейсом I2C який знаходиться на дисплеї зображено на рисунку 1.4.

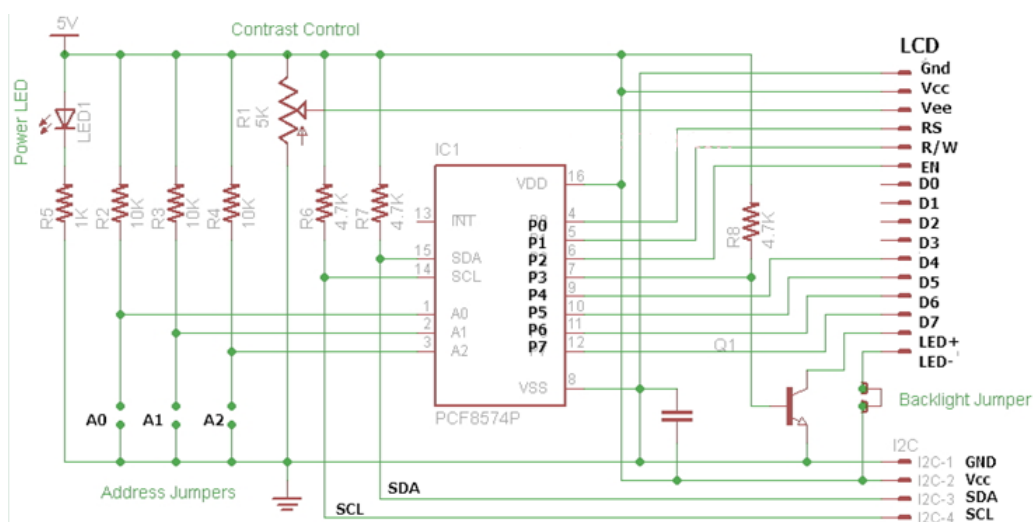


Рисунок 1.4 — Схема електрична принципові модуля I2C на дисплеї

Щоб дисплею передати інструкцію потрібно встановити логічний нуль на біті RS. Щоб передати символу для виведена на дисплеї потрібно RS встановити в логічну одиницю. Біт RS підключений до нульового біта P0. До P1 під'єднаний біт читання / запис (R/W). P2 під'єднаний до біта підтвердження(EN), а P3 підключений до транзистора який включає або виключає підсвітку дисплея. Тому для шини даних залишається чотири біта. Інструкції однобайтні тому щоб передати інструкцію потрібно підтвердити два пів байта. Інструкції наведені в таблиці 6 за джерелом [6]. Передача даних відбувається по чотири біта. Спочатку передаються чотири старших

біта далі чотири молодших. Підтвердження чотирьох біт відбувається по формуванню заднього фронту на біті “EN”.

Приклад відображення символу пробілу “20h” зображено на рисунку 1.5. Встановлюємо логічну одиницю на біт RS для того щоб вказати що це не інструкція. Зводимо біт підтвердження “EN” одночасно з старшим чотирма бітами “2h”. При наступному відправленні змінюємо лише біт підтвердження. Отже не змінюючи значення на бітах “P7-P4” сформувався задній фронт на біті підтвердження “EN”. Тому мікроконтролер на дисплеї збереже значення на бітах “P7-P4”. Повторюємо те саме з молодшим чотирма бітами знаку пробілу “0h”.

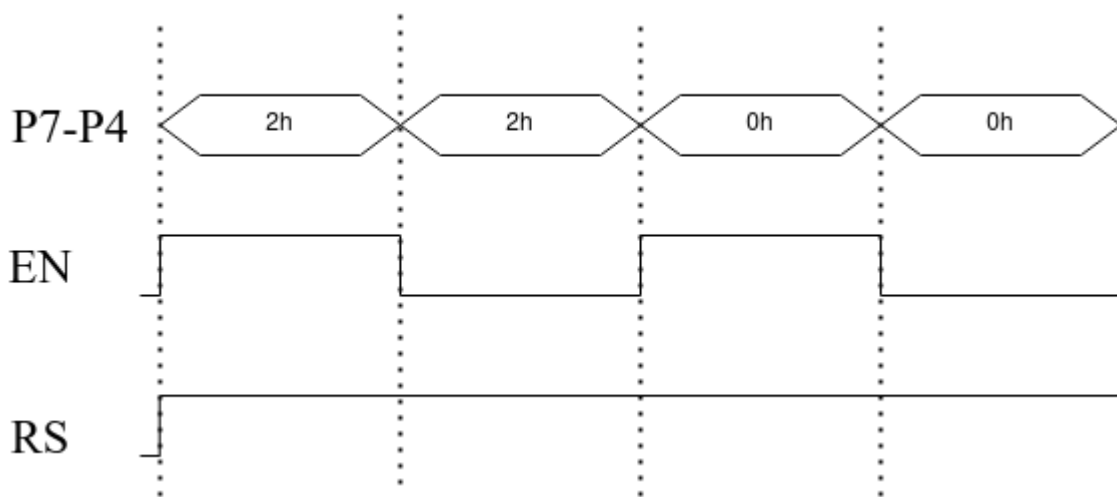


Рисунок 1.5 — Відправлення символ пробілу “20h” на дисплей

Розібравшись з методом комунікації з дисплеєм розглянемо пункти які потрібно зробити щоб запустити і налаштувати для роботи дисплея з чотирибітовою шириною комунікації:

- налаштувати ширину шини для комунікації (відправляємо лише пів байт);
- налаштувати “Function Set”;
- включити дисплей, інструкція “Display on/off control”;

- встановити напрямок переміщення курсора і зсув дисплея, інструкція “Entry mode set”.

Хочу звернути увагу, якщо спілкування з дисплеєм відбувається з восьмибітовою шириною, то не потрібно робити першого пункту, а зразу відбувається інструкція “Function Set”. Чому? Після reset (скидання) дисплей працює з восьмибітової ширині комунікації. Але якщо ми працюємо в чотирибітовій ширині то ми не можемо відправити всю інструкцію, а лише чотири старших біта. Звернемо увагу на саму інструкцію “Function Set” рисунок 1.6. Біт визначення функції (DB5) і біт налаштування ширини шини (DB4) знаходиться в старших чотирьох бітах. А ми знаємо що відправляється спочатку чотири старших біта, а потім чотири молодших. Отже після скидання першим пів байтом ми можемо задати довжину комунікації дисплея надіславши лише старші чотири біта інструкції “Function Set”. Наступні інструкції відбувається в режимі підтвердження чотирьох біт.

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function Set	0	0	0	0	1	DL	N	F	-	-
Sets interface data length (DL), number of display lines (N), and character font (F).										

Рисунок 1.6 - Інструкція Function Set

1.5. Принципи і засоби обробки даних

1.5.1 Мікроконтролер

Блок обробки даних побудований на мікроконтролері STM32F401RE.

За допомогою налаштованого лічильника TIMx визначається тривалість імпульсу, який формує ультразвукового датчика HC-SR04. Далі відбувається збереження результату. В кінці відбувається конвертація в текст і вивід на блок відображення вихідних даних LCD дисплей.

Лічильник TIMx підключений до виходу далекоміра «Echo». Рахунок лічильника відбувається лише в момент високого рівня на виході далекоміра

«Echo». При кожній подачі високого логічного рівня тривалістю 10 мкс на його вхід «Trig», очищується лічильник. Через деякий час після імпульсу на вході «Trig» в лічильнику зберігається рахунок. Значення якого буде використано для обрахунку відстані до об'єкту.

Обрахований результат конвертується в масив символів ASCII. В подальшій дії відбувається відображення на дисплей.

Зафіксувавши відстані. Подвійному натисканні обраховуємо значення за формулою:

$$z = (x - y) \cdot 2 \quad (1.5.1)$$

де x, y – зафіксовані значення

Обрахувавши z за формулою 1.5.1 конвертуємо у масив символів ASCII. Виводимо на дисплей.

1.5.2 Лічильники

На борту мікроконтролера є багато різних за функціоналом лічильників.

Щоб перевести значення лічильника в час імпульсу потрібно обраховуємо за формулою 1.5.2. Щоб обрахувати час потрібно знати частоту тактування лічильника і значення лічильника. Знаючи час імпульсу на «Echo» обраховуємо відстань за формулою 1.3.1.

$$t = \frac{n}{F_{clk}} = n \cdot T_{clk} \quad (1.5.2)$$

де n – кількість порахований тактів лічильника;

F_{clk} — частота тактового сигналу лічильника, МГц;

T_{clk} — період тактового сигналу.

Об'єднаємо формулу 1.5.2 і 1.3.1. Отже повна формула має такий вигляд:

$$d = \frac{n}{58 \cdot F_{clk}} = \frac{n \cdot T_{clk}}{58} \quad (1.5.3)$$

Даний мікроконтролер має внутрішнє джерело тактового сигналу на 16 МГц. Проведемо розрахунок для максимальної відстані 400 см.

$$n = 58 \cdot d \cdot F_{clk} = 58 \cdot 400 \cdot 16 = 371,2 \cdot 10^3$$

На борту мікроконтролера є як 16-біт так і 32-біт лічильники. Зробивши цей обрахунок ми дізналися що в 16-біт не може вмістити обраховане число. 16-біт лічильник вміщує 65536 значень. Лічильники мають prescaler. Prescaler може задавати кількість імпульсів лічильник для одного рахунку лічильника, можемо вважати що ми зменшуємо частоту тактування. Обрахуємо для 400 см з prescaler 16:

$$\frac{371,2 \cdot 10^3}{16} = 23,2 \cdot 10^3$$

За допомогою цього можна використовувати 16-біт лічильник, але ми зменшимо точність обрахунку. Тому все таки будемо використовувати 32-біт лічильник.

На борту є два лічильника TIM2 і TIM5 які є 32-біт. Тому для далекоміра будемо використовувати TIM2. Також для програмних затримок використаємо 16-біт лічильник TIM9.

На рисунку 1.7 зображена структура таймера TIM2. Лічильник потрібно налаштувати щоб рахунок відбувався при високому рівні на піні кий підключений до лічильника. TIM2_CN1 підключений до PA0 піна мікроконтролера. Даний порт потрібно підключити до TI1 далі через блок “Input filter & edge detector” до TI1FP1. Якщо потрібно тактувати при логічному нулі то в даному блоці потрібно інвертувати сигнал за допомогою бітів CC1NP і CC1P в регістрі TIMx_CCER. Далі TI1FP1 підключаємо через мультиплексор на дріт TRGI який підключений до Slave mode controller.

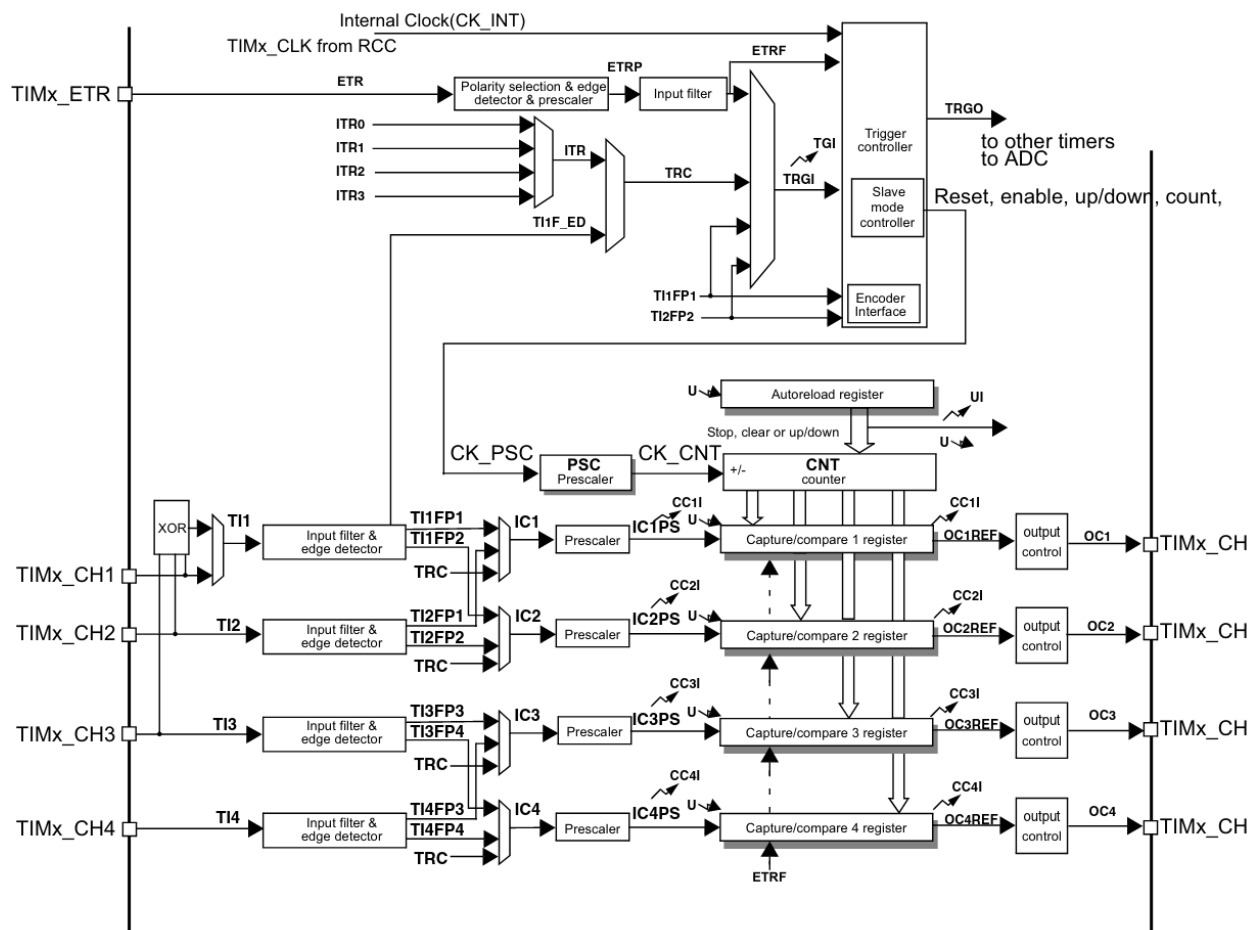


Рисунок 1.7 — Структура лічильника TIM2

Slave mode controller має такі режими:

- Режим кодера 1 - Лічильник відлічує вгору / вниз по краю TI2FP1 залежно від рівня TI1FP2;
- Режим кодера 2 - Лічильник відлічує / вниз на краю TI1FP2 в залежності від рівня TI2FP1;
- Режим кодера 3 - Лічильник відлічує вгору / вниз як на ребрах TI1FP1, так і TI2FP2 залежно від рівня іншого входу.) повторно ініціалізує лічильник та генерує оновлення регістрів;
- Режим скидання (Reset Mode) - Висхідний край вибраного тригерного входу (TRGI) повторно ініціалізує лічильник і генерує оновлення регістрів.

- Режим закритого типу (Gated Mode) - Лічильник вмикається, коли вхід тригера (TRGI) високий. Лічильник зупиняється (але не скидається), як тільки тригер стає низьким;
- Режим спрацьовування (Trigger Mode) - Лічильник запускається з висхідного краю тригера TRGI (але він не скидається). Контролюється лише початок лічильника;
- Режим зовнішнього годинника 1 - Зростання країв обраного годинника тригера (TRGI) лічильника

Обираємо режим закритого типу (Gated Mode).

Отже лічильник налаштований. Тепер лічба здійснюється лише при наявності високого логічного рівня на піні PA0.

Розділ 2. Схема електрична принципова

Схема електрична принципова зображена на рисунку 2.1. В проекті використовується джерело постійної напруги який формує стійку напругу 5 В і 3,3 В.

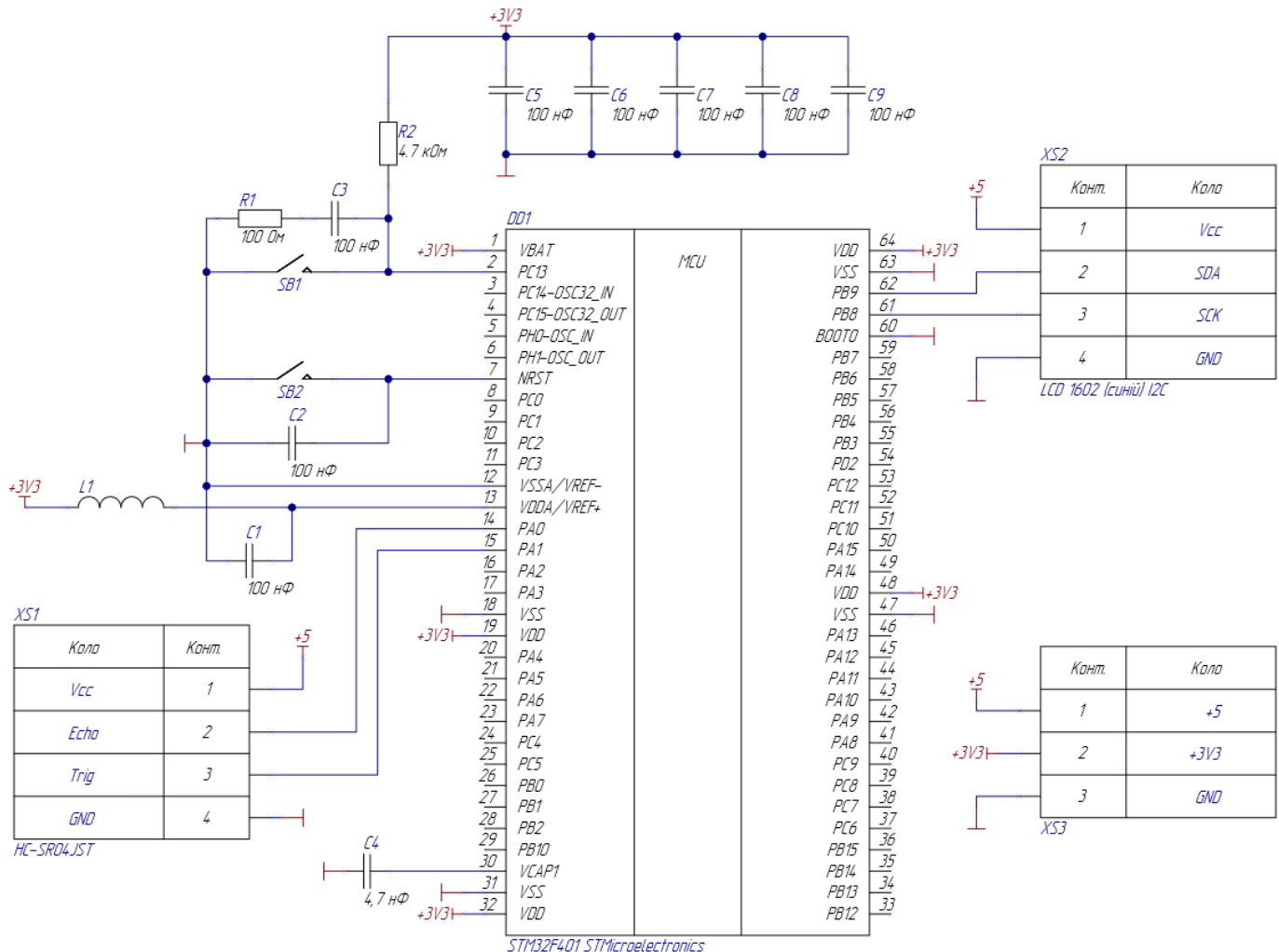


Рисунок 2.1 — Схема електрична принципова

2.1. Проектування блоку обробки даних

Блок обробки даних складається з STM32F401RE. Для того щоб запустити мікроконтролер потрібна обв'язка яка складається з конденсаторі C1-C2, C4-C9, котушки індуктивності L1 і кнопки скидання(Reset) SB1.

2.2. Проектування блоку контролю

Блок контролю складається з кнопки SB1 і резистора R2. Для того щоб прибрати брязкіт контактів додатково встановлено резистор R1 і конденсатор C3. Кнопка підключена до піна мікроконтролера PC13.

2.3. Проектування блоку введення даних

Блок введення даних складається з роз'єму XS1, до якого підключено модуль HC-SR04. Даний модуль потребує живлення 5 В. Вхід модуля «Trig» підключено до PA1. Вихід модуля «Echo» підключено PA0. Так як даний пристрій потрібно підключити безпосередньо до лічильника то заздалегідь було визначено в Datasheet відповідні піни.

2.4. Проектування блоку відображення вихідних результату

Блок відображення вихідних результату складається з роз'єму XS2 до якого підключено LCD 1602 дисплей. Дисплей працює через інтерфейс I2C. «SCK» до PB8. Шину даних «SDA» підключено до PB9. Так як даний пристрій потрібно підключити до модуля I2C в мікроконтролера то заздалегідь визначаємо в Datasheet піни які підтримуються цей інтерфейс.

2.5 Вибір елементної бази

2.5.1. Конденсатори

При виборі конденсаторів можна виділити основні параметри:

- ємність;
- робоча напруга конденсатора;
- точність;
- тип конденсатора;
- робоча температура;

- спосіб монтажу.

Конденсатори було вибрано в SMD корпусі. Було взято керамічні конденсатори в корпусі 0603 фірми KEMET. Дана серія задовільняє точність яка складає 5%, і має всі потрібні номінали ємності. Максимальна робоча напруга складає 16 В. Тепловий коефіцієнт Х7R.

Для електролітичних конденсаторів. Було взято конденсатори фірми Hitano. Конденсатори даною фірми має всі потрібні номінали ємності. Максимальна робоча напруга складає 16 В.

2.5.2. Резистори

Резистори фірми Yageo задовольняють всі вимогам. Дані резистори випускаються в корпусі 0603. Мають точність 1%. Розсіювальна потужність 100 мВт.

2.5.3. Кнопки

Кнопка TL3300CF160Q компанії E-Switch має великі розміри що робить її зручною при натисканні. Дані кнопки мають великий час напрацювання і задовольняє вимоги до надійності.

2.5.4 Мікроконтролер

Центральною ланкою в пристрою керування сервоприводом є блок, що оброблятиме дані, отримані з блоку введення даних та формуватиме дані для передачі на блок виведення даних. Згідно технічного завдання, це буде мікроконтролер STM32F401. Мікросхему виконано у корпусі LQFP зображено на рисунку 2.2.

Характеристики мікроконтролера:

- Core: ARM® 32-bit Cortex® -M4
- 512 KB Flash пам'ять

- 96 KB SRAM
- 4 SPI
- 3 I²C
- 3 UART
- 10 таймерів
- Частота вбудованого генератора синхроімпульсів HSI - 8 MHz
- Напруга живлення 1.7 В - 3.6 В

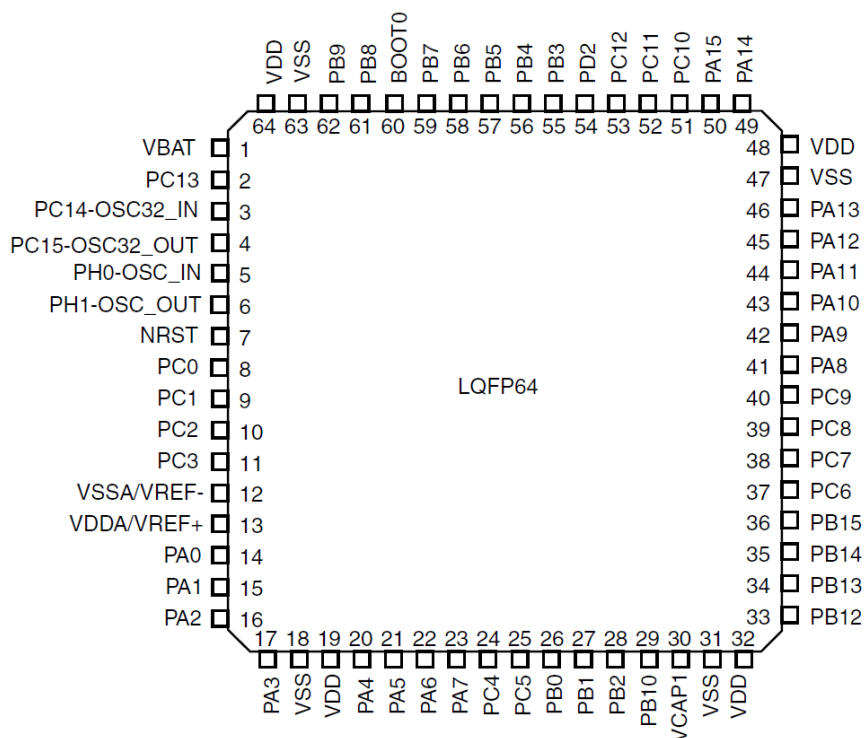


Рисунок 2.2 - Мікросхема STM32F401RE в корпусі LQFP

В даному мікроконтролері присутні 32-біт таймери. Є I²C модуль. Отже даний мікроконтролер підходить для виконання завдання.

2.5.5. Роз'єми

Було взято штирі фірми Connfly electronic (Zhenqin). DS1021 задовольняє експлуатаційні характеристики, а також має гарні електричні характеристики.

2.5.6. Далекомір

Ультразвуковий датчик HC-SR04 відповідає всім вимогам для розрахунково-графічної роботи. Потребує 5 В живлення. Максимальна відстань яку підтримує даний модуль складає 400 см.

2.5.7. Дисплей

Символьний дисплей LCD 1602. Дисплей повинен мати плату з послідовним I2C інтерфейсом, яка розташована в задній частині конструкції. Даний модуль зручний в використанні бо має вбудованим I2C модуль. Має напругу живлення 5 В. Має невеликі габарити і малу ціну.

2.5.8. Котушки індуктивності

Електричні характеристики котушок індуктивності визначаються їх конструкцією, властивостями матеріалу магнітопроводу та його конфігурацією, числом витків обмотки.

Основні фактори, які враховуємо при виборі котушки індуктивності:

- значення індуктивності
- максимальний струм котушки
- точність
- спосіб монтажу.

Індуктивність обрано типу SMD, корпусу 0805 з 5% похибкою. Переваги – малі розміри, низька ціна та висока якість.

Розділ 3. Програма керування пристроєм

3.1. Структурна схема програми

Структура програми зображено на рисунку 3.1. Програма має структуру автомата який має 5 стани. Зміна стану відбувається в залежності від натиску кнопки виключення є нульовий стан. Програма починає роботу в нульовому стані в яку попадає при скиданні мікроконтролера. З нульового стану відбувається перехід до першого в якому починається основна робота. При одинарному натисканні відбувається циклічний послідовний перехід з першого стану до третього включно. Не залежно від стану при подвійному натисканні відбувається перехід до четвертого стану. Знаходячись в четвертому стані при одиничному натисканні відбувається перехід до першого стану.

В першому стані обраховується відстань за формулою 1.5.3, записується в змінну x і відображається число на дисплей. В другому стані обраховується відстань за формулою 1.5.3, записується в змінну y і відображається число на дисплей. В третьому стані відображається число x і y на дисплей. В четвертому стані відображається число x і y обраховується z за формулою 1.5.1 відбувається перехід на нижній ряд і відображається число на дисплей.

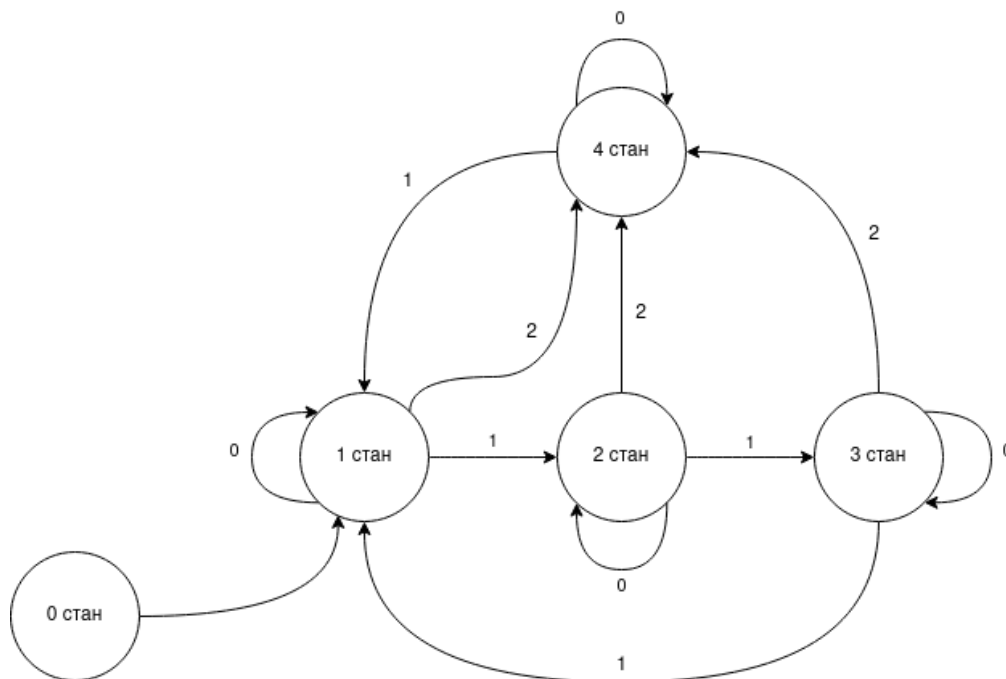


Рис. 3.1 - Діаграма станів

3.2. Опис основної програми

Програма керування пристроєм написана на мові програмування C з використанням асемблера, а також використанням бібліотек CMSIS. Початок відбувається в функції main. В даній функції відбуваються такі етапи.

1. Оголошуються змінні.
2. Створюється структура для дисплея
3. Відбувається заборона всіх переривань.
4. Налаштування модулів: лічильники, інтерфейс, кнопку, дисплей.
5. Дозволяються переривання.
6. Входимо в головний цикл.

В головному циклі відбувається визначення відстані результат функції HC_SR04. Обрахунок відстані за формулою 1.5.3 і запис результату в змінну distance. Наступна дія є відпрацювання стану в якому находимося. Далі робимо затримку в 131 мс.

3.3. Структура даних

Таблиця 3.1 — Змінні які були об'явлені в програмі

Назва	Тип змінної	Призначення
x, y, z	32 розрядний беззнаковий	Збереження змінні
distance	32 розрядний беззнаковий	Зберігає теперішнє значення відстані
dis_x_ascii, dis_y_ascii, dis_z_ascii	Масив з трьох 8 розрядних змінних типу char	Зберігається значення змінних x, y, z в ASCII форматі
d_x, d_y, d_z	Масив з трьох 8 розрядних змінних типу char	Зберігаються текст для станів
stage	8 розрядна змінних типу char	Зберігає теперішній стан
LCD1	Структура	Зберігає структуру дисплея
press	8 розрядна змінних типу char	Зберігає

Таблиця 3.2 - Використані виводи плати NUCLEO

Назва	Призначення
PA0	Підключено до виходу далекоміра “Echo”
PA1	Підключено до входу далекоміра “Trig”
PB8	I2C1_SCL
PB9	I2C1_SDA
PC13	Підключена кнопка
+5V	Живлення для підключених модулів
+3V3	Живлення для мікроконтролера
GND	Землі

3.4. Опис алгоритмів окремих підпрограм

3.4.1 Опис роботи таймерів TIMx

1. Опис роботи функції TIM2_INIT

Вихід далекомір «Echo» потрібно підключити до лічильника. Для цього використовуємо лічильник TIM2. На рисунку 3.2 зображено Alternate function mapping. З рисунка видно що TIM2_CH1 підключений PA0.

Підключаємо тактування до GPIOA. Налаштовуємо PA0 як альтернативну функцію і під'єднуємо до TIM2. В GPIOA_AFRL для портів PA0 встановлення AF1. PA1 налаштовуємо на вихід.

Підключаємо тактування до самого TIM2. Далі потрібно налаштувати на режим підлеглого(Gated).

Port	AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
	SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/SPI2/ I2S2/SPI3/ I2S3/SPI4	SPI2/I2S2/ SPI3/ I2S3	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS		SDIO			
Port A	PA0	-	TIM2_CH1/ TIM2_ETR	TIM5_CH1	-	-	-	USART2_ CTS	-	-	-	-	-	-	-	EVENT OUT
	PA1	-	TIM2_CH2	TIM5_CH2	-	-	-	USART2_ RTS	-	-	-	-	-	-	-	EVENT OUT
	PA2	-	TIM2_CH3	TIM5_CH3	TIM9_CH1	-	-	USART2_ TX	-	-	-	-	-	-	-	EVENT OUT
	PA3	-	TIM2_CH4	TIM5_CH4	TIM9_CH2	-	-	USART2_ RX	-	-	-	-	-	-	-	EVENT OUT
	PA4	-	-	-	-	SPI1_NSS	SPI3_NSS/ I2S3_WS	USART2_ CK	-	-	-	-	-	-	-	EVENT OUT
	PA5	-	TIM2_CH1/ TIM2_ETR	-	-	SPI1_SCK	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA6	-	TIM1_BKIN	TIM3_CH1	-	-	SPI1_ MISO	-	-	-	-	-	-	-	-	EVENT OUT
	PA7	-	TIM1_CH1N	TIM3_CH2	-	-	SPI1_ MOSI	-	-	-	-	-	-	-	-	EVENT OUT
	PA8	MCO_1	TIM1_CH1	-	-	I2C3_SCL	-	USART1_ CK	-	-	OTG_FS_ SOF	-	-	-	-	EVENT OUT
	PA9	-	TIM1_CH2	-	-	I2C3_ SMBA	-	USART1_ TX	-	-	OTG_FS_ VBUS	-	-	-	-	EVENT OUT
	PA10	-	TIM1_CH3	-	-	-	-	USART1_ RX	-	-	OTG_FS_I D	-	-	-	-	EVENT OUT
	PA11	-	TIM1_CH4	-	-	-	-	USART1_ CTS	USART6_ TX	-	OTG_FS_ DM	-	-	-	-	EVENT OUT
	PA12	-	TIM1_ETR	-	-	-	-	USART1_ RTS	USART6_ RX	-	OTG_FS_ DP	-	-	-	-	EVENT OUT
	PA13	JTMS SWDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA14	JTCK SWCLK	-	-	-	-	-	-	-	-	-	-	-	-	-	EVENT OUT
	PA15	JTDI	TIM2_CH1/ TIM2_ETR	-	-	-	SPI1_NSS	SPI3_NSS/ I2S3_WS	-	-	-	-	-	-	-	EVENT OUT

Рисунок 3.2 - Alternate function mapping Port A

Налаштовуємо канал 1 для детектування високих рівнів на TI1. Записуємо CC1P = 0 і CC1NP = 0 в регістрі TIMx_CCER.

Налаштовуємо таймер в gated-режим записом SMS = 101 в регістр TIMx_SMCR. Вибираємо TI1 в якості входу шляхом запису TS = 101 в регістр TIMx_SMCR.

Запускаємо лічильник записом CEN = 1 в регістр TIMx_CR1. В gated-режимі лічильник не запуститься, якщо CEN = 0, незалежно від вхідного рівня тригера.

2. Опис функції TIM9_INIT

Даний таймер використовується для затримок в програмі. Підключаємо тактування до TIM9. Включаємо One-pulse мод. В регістр prescaler TIMx_PSC встановлюємо значення частоти тактування лічильника в MHz. Отже ми отримаємо лічильник в якого один рахунок триває 1 мкс.

3. Опис функції SLEEP_T9

Дана функція працює як програмна затримка. У функцію передається значення часу затримки в мкс. Функція використовує TIM9. Записуємо передану змінну в auto-reload TIMx_ARR. Запускаємо лічильник. Чекаємо поки рахує лічильник. Лічильник налаштований на One-pulse, тому він зупиниться як дорахує до значення TIMx_ARR. Лічильник зупинився виходимо з функції.

3.4.2 Опис роботи Кнопки

1. Опис функції BUTTON_INIT

Дана функція налаштовує 13 ніжку мікроконтролера на переривання при формуванні на ній переднього фронту. До цієї ніжки підключена кнопка. На рисунку 2.1 видно пін через резистор підключений до живлення, а кнопка закорочує пін на землю при натисканні. Отже переривання буде відбуватися при відпусканні кнопки.

Підключаємо тактування до GPIOC і SYSCFG. В контролері конфігурації системи в зовнішньому регістрі конфігурації переривань для EXTI13 встановлюємо PC[13] біт.

Прибираємо маску для переривання в регістрі EXTI_IMR. Включаємо переривання по передньому фронті встановлення одиниці для PC[13] в регістрів EXTI_RTSR. Включаємо переривання EXTI15_10_IRQn.

Після цього при формуванні переднього фронту буде відбувати переривання. Далі потрібно налаштувати функцію обробника EXTI15_10_IRQHandler.

2. Опис функції EXTI15_10_IRQHandler

При відтисканні кнопки відбудеться переривання. І це означає що одне натискання вже відбулося до потрапляння в обробник. При перериванні потрапляємо у функцію EXTI15_10_IRQHandler. У функції відбувається затримка при якій очікуємо натиску на кнопку. Якщо відбулося друге натискання чекаємо відпущення кнопки.

Якщо відбулося подвійне натискання змінюємо стан на четвертий. Якщо відбулося одне натискання переходимо до наступного стану. У випадку якщо находимося на третьому і четвертому стані то переходимо до першого стану.

Очищуємо дисплей і знімаємо біт очікування на переривання в регістрі EXTI_PR.

3.4.3 Опис роботи функції HC_SR04

Після reset (скидання) до того як ми потрапимо в перший стан відбувається налаштування модулів включно з лічильником TIM2. Відпрацювавши функцію TIM2_INIT лічильник включений і чекає формування високого рівня на піні PA0. Перед тим як подавати імпульс далекомір потрібно скинути рахунок лічильника в нуль.

Формуємо імпульс тривалістю 10 мкс. Так як нам не потрібно дотримуватися точних значень часу імпульсу використаємо програмну затримку за до допомогою налаштованого лічильника TIM9. Встановлюємо високий логічний рівень на PA1 «Trig» і робимо затримку на 10 мкс викликавши функцію SLEEP_T9, а при поверненні встановлюємо низький логічний рівень на PA1«Trig».

Робимо затримку щоб за цей час сформувався імпульс на на PA0 «Echo» і лічильник закінчив рахувати. Максимальна відстань далекоміра HC-SR04 становить 400 см. Максимальна тривалість імпульсу обраховуємо за формулою 1.3.1.

3.4.4 Опис роботи I2C

1. Опис роботи функції I2C_INIT

Потрібно налаштувати порти і модуль I2C.

На рисунку 3.3 бачимо що I2C1_SDA і I2C1_SCL підтримують порти PB8 і PB9, а AF становить AF4. Отже потрібно включити тактування для GPIOB, налаштувати порти PB8 і PB9, встановивши AF + OD (Open Drain). Під'єднуємо модуль I2C1 до цих портів, встановивши для портів PB8 і PB9 значення AF4 в регістрі GPIOB_AFRL.

Port	AF00	AF01	AF02	AF03	AF04	AF05	AF06	AF07	AF08	AF09	AF10	AF11	AF12	AF13	AF14	AF15
	SYS_AF	TIM1/TIM2	TIM3/ TIM4/ TIM5	TIM9/ TIM10/ TIM11	I2C1/I2C2/ I2C3	SPI1/SPI2/ I2S2/SPI3/ I2S3/SPI4	SPI2/I2S2/ SPI3/I2S3	SPI3/I2S3/ USART1/ USART2	USART6	I2C2/ I2C3	OTG1_FS		SDIO			
Port B	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

Рисунок 3.3 - Alternate function mapping Port B

Налаштування модуля I2C наведені в 1 розділі у пункті 4 підпункт 1.

2. Опис роботи функції I2C_WRITE

В дану функцію надсилається адрес на масив байтів і кількість байтів на відправлення. Принцип відправлення розглянутий в першому розділі.

Зберігаємо значення регістрів в стек.

Встановлюємо START біт в першому статус регістрі. Даний біт очиститься автоматично. Чекаємо на встановлення SB прапора. Далі обов'язково читаємо два статус регістра (обов'язково два). Записуємо в регістр даних адресу. Не забуваємо що адреса починається з першого біту, бо нульовим бітом задаємо читання або запис. Чекаємо на встановлення прапора ADDR в першому статус регістрі. Знову читаємо два статус регістра. Чекаємо на встановлення TxЕ. Записуємо в регістр даних байт який потрібно відправити. Після цього чекаємо на встановлення TxЕ. Повторюємо запис регістр по встановлення TxЕ.

TxE прапор вказує що регістр зсуву пустий. Блок-схема I2C модуля зображена на рисунку 3.4. Запис в регістр зсуву відбувається при запису значення в регістр даних. Якщо встановлено TxЕ, а байт даних не був записаний в регістр DR до кінця останньої передачі даних, встановлюється BTF, і інтерфейс чекає, поки BTF очиститься при записі в I2C_DR, розтягуючи низький рівень SCL. Таким методом створюється затримка передачі щоб мікроконтролер записав значення в регістр даних.

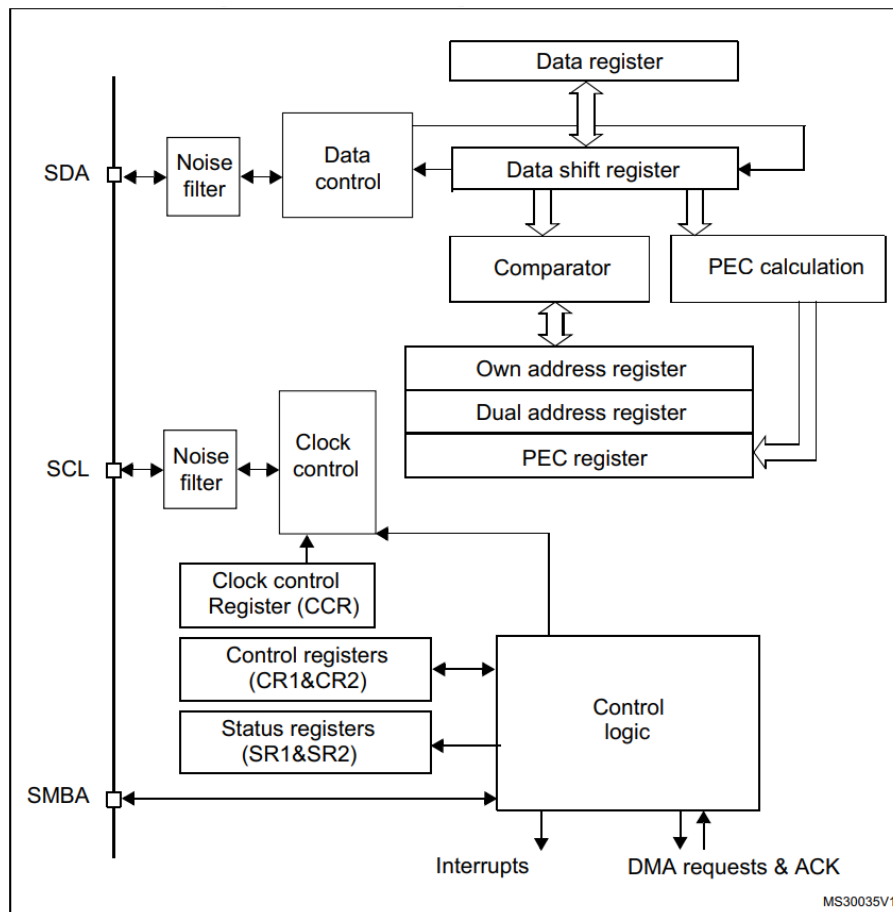


Рисунок 3.4 - Блок-схема I2C для STM32F40x / 41x

Після надіслання інформації чекаємо на завершення надіслання останнього байту зведення прапора TxE. Встановлюємо STOP біт в першому статус регістрі.

Відновлюємо регістри і виходимо з функції.

3.4.5 Опис роботи LCD

Функції LCD_FUNC_SET, LCD_ON, LCD_ENTRY_MODE, LCD_HOME, LCD_CLEAR, LCD_Set_DDRAM виконую функції display_write з вже визначеними командами для дисплея.

1. LCD_MODE

В даній функції відбувається налаштування режиму передачі даних.

Для коректної роботи дисплея повинен бути скинутий (відключений від живлення) перед функцією LCD_MODE. Після скидання дисплей підтверджує восьмибітову операцію, тому функція чекає на підтвердження восьми біт. Налаштування відбувається відправлення *лише* старших чотирьох біт команди Function set. Якщо дисплей вже підтверджує чотири біта то відбудеться підтвердження пів байту, а не всієї інструкції. Після спрацювання цієї функції дисплей буде чекати на підтвердження молодшого пів байту інструкції Function set, а отримає чотири старших біта наступної команди. Дисплей буде отримувати не правильні команди і не буде відображувати коректно інформацію. Но при повторному відпрацюванні LCD_MODE дисплей буде відображати коректну інформацію.

2. DISPLAY_4B_INIT

В даній функції відбувається налаштування структури дисплея. Структура була зроблена для спрощення роботи з дисплеєм. Функції налаштовані на визначені константи і значені змінних в структурі. Залишається ввести потрібні значення в структуру. Всі константи вже визначені в голові бібліотеки.

Встановлюємо адресу дисплея.

Налаштовуємо дисплеї на два ряди відображення. Потрібно за умовою завдання. Включаємо дисплей і очищуємо його.

Після цієї функції дисплей буде налаштований до відображення інформації.

3. display_write

Дана функція відправляє команду для дисплея. В даній функції скидається біт RS і R/W. Враховується біт підсвічування дисплея. Виконується відправлення команди по чотири біта з врахуванням біту

підтвердження EN. Функції працює принцип наведений в 1 розділі 4 пункті 2 підпункті.

4. LCD_PRINT

Дана функцію виводить масив символів на дисплей. Функція аналогічна до функції `display_write`. В даній функції встановлений біт RS. Враховується біт підсвічування дисплея. У функцію відправляється адрес на масив символів і кількість символів яку потрібно вивести. Приклад відправлення одного символу наведено на рисунку 1.4.

5. LCD_BACKLIGHT_ON / LCD_BACKLIGHT_OFF

Ці функції включають або виключають підсвічування дисплея записавши в структуру відповідне значення і відправивши невизначену інструкцію. Для цього викликається функція `display_write`. Це спрощує роботу. Ми маємо дві функцію які автоматично включаються або виключаються підсвічування дисплея. Лише надіславши адресу структури.

3.4.6 Опис перетворення числа в ASCII

Передаємо в функцію адресу на масив R0, число для конвертації R1, розмір масив R2.

Зберігаємо регістри які будемо використовувати в стек.

Очищуємо масив символів вписавши значення пробілів "20h".

Визначаємо знак числа. Якщо число в доповнювальному коді інвертуємо побітово і додаємо одиницю.

Збільшуємо значення регістра R10 в десять раз і зсуваємося по масиву на байт рухаючись з кінця масиву на початок. Повторюємо ці дії поки число в регістрі R10 не буде більшим за значення регістра R1 після цього виходимо з циклу.

Якщо число від'ємне зсуваємося на байт до початок масиву і записуємо символ "2D".

Зменшуємо значення в десять раз і віднімаємо його від числа в регістрі R1. Отримане число є найстаршим розрядом десяткового числа. Дане число зберігаємо в масив додавши константу "30h". Дану різницю множимо на вміст регістра R10 і від отриманого числа віднімаємо значення регістра R1 результат перезаписуємо в регістр R1. Повторюємо поки вміст регістра R10 не буде містити одиницю.

Після цього вписуємо залишок в самий останній байт масиву додавши константу "30h" перед цим.

Виходимо з функції відновивши вміст збережених регістрів в стеку.

3.4.7 Опис функції FUNC_Z

Дана функція повертає результат формули 1.5.1.

Відбувається віднімання двох чисел і зсув вліво (множення на два).

Висновок

В даній роботі було зроблено макет прототипу пристрою який визначає відстань. Було отримано знання і навички роботи з мікроконтролером STM32F401RE та платою NUCLEO яку було отримано при вивченні цього курсу. Отримано навички роботи з інтерфейсом I2C, лічильниками, також з сторонніми модулями такі як LCD дисплей, модуль послідовно-паралельного перетворювача, далекоміра, кнопок. Та створення електричних схем.

Розроблено програмне забезпечення у середовищі ARM Keil та описано принцип роботи програми. Розроблено структурну схему, схему електричну принципову для створення пристрою. Проектування відбувалося поетапно відповідно до календарного плану.

					ДК81.460839.001 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Література

1. STM32F401RE Nucleo User Manual (UM1724): [Електронний ресурс] - Режим доступу:
https://www.st.com/content/ccc/resxxxxxxurce/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf
2. Cortex-M4 Generic User Manual: [Електронний ресурс] - Режим доступу:
http://www.st.com/resource/en/programming_manual/dm00046982.pdf
3. STM32F401xB/C/D/E Reference Manual: [Електронний 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. STM32F401xD/401xE Data Sheet: [Електронний ресурс] - Режим доступу:
<http://static6.arrow.com/aropdfconversion/81ff5929a2507cbf942e2344f64088b378a50a40/447dm00102166.pdf>
5. Ultrasonic Ranging Module HC — SR04 : [Електронний ресурс] - Режим доступу:
<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
6. LCD HD44780 Data Sheets: [Електронний ресурс] - Режим доступу:
<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
7. Віддалений 8-розрядний розширювач вводу-виводу для I2C-шини.
[Електронний ресурс] - Режим доступу:
https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf

8. “STM32F4xx: продвинутые таймеры TIM1 и TIM8”: [Электронный ресурс] - Режим доступа:

<http://microsin.net/programming/arm/stm32f4xx-advanced-control-timers-tim1-tim8.html>

					ДК81.460839.001 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

ДОДАТОК А

Технічне завдання на проектування

1. Найменування та область використання.

Пристрій визначення відстані до об'єкта. Призначений для використання в навчальних цілях.

2. Підстава для розробки.

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

3. Мета і призначення роботи.

Метою даної роботи є навчитися проектування цифрової апаратури на основі мікроконтролера сімейства STM32. Навчитися працювати з різними модулями і інтерфейсами передачі даних.

4. Технічні вимоги.

4.1. Склад виробу й вимоги до пристрою, що розробляється.

Пристрій являє собою конструкцію, яка містить мікроконтролер STM32F401RE і кнопки на борді STM32 Nucleo-64, рідкокристалічний дисплей, далекомір HC-SR04.

4.2. Показники призначення.

Пристрій повинен забезпечувати:

1. визначати відстань
2. виводити відстань на екран

4.3 Технічні характеристики.

1. Напруга живлення далекоміра, дисплея: +5 В.

2. Напруга живлення мікроконтролера: +3.3 В.

3. Максимальна відстань 400 см

5. Результати роботи

5.1. Результати даної роботи можуть бути використані як вихідна документація по створенню прототипу пристрою, його програмування, налагодження.

5.2. Дана робота (звітна документація) після виконання надається на кафедру КЕОА для подальшого захисту й зберігання як навчальної документації.

					ДК81.460839.001 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40