11. **Новицкий, Б.Г.** Применение акустических колебаний в химико-технологических процессах (Процессы и аппараты химической и нефтехимической технологии) / Б.Г. Новицкий. – М.: Химия, 1983. – 192 с.
12. **Губин Г.Г.** Обобщение и анализ возможности использования ультразвуковых колебаний при переработке полезных ископаемых / **Г.Г. Губін, Т.П. Ярош, Л.В. Скляр** // Збагачення корисних копалин: Наук.-тех. зб. – 2016. – Вип. 62 (103). – С. 132-143.
13. Ультразвук в обогащении полезных ископаемых / **В.А. Глембоцкий, М.А. Соколов, И.А. Якубович и др.** – Алма-Ата: Наука Каз. ССР, 1972. – 229 с.
14. Ультразвук в гидрометаллургии / **Б.А. Агранат, О.Д. Кириллов, Н.А. Преображенский и др.** – М.: Металлургия, 1969.
15. **Агранат Б.А., Хавский Н.Н., Хан Г.А., Пантелеева Н.Ф., Эльберт А.В.** Флотация в воде, обработанной ультразвуком // Интенсификация процессов извлечения металлов из руд в ультразвуковом поле. – М.: Металлургия, 1969.
16. **Коротич В.И.** Теоретические основы окомкования железорудных материалов. – М.: Металлургия, 1966.
17. **Кокорин Л.К., Лелеко С.Н.** Производство окисленных окатышей. Екатеринбург: Уральский центр ПР и рекламы, 2004, – 280 с.
18. Теоретические основы производства окускованного сырья: Учебное пособие для высших учебных заведений. **Ковалёв Д.А., Ванюкова Н.Д., Иващенко В.П. и др.** – НМетАУ. – Днепропетровск: ИМА-пресс. – 2011. – 476 с.
19. **Вегман Е.Ф.** Окускование руд и концентратов. М.: Металлургия. 1968. – 259 с.
20. Сырье для черной металлургии: Справочное издание: в 2-х т. Т.1 Сырьевая база и производство окускованного сырья (сырье, технологии, оборудование) / М.Г. Ладыгичев и др. – М.: Машиностроение-1, 2001. – 896 с.
21. **Коротич В.И.** Основы теории и технологии подготовки сырья к доменной плавке / **Коротич В.И.** − М.: Металлургия, 1978. – 208 с.
22. **Лесив Е.М.** Механохимическая активация каолиновых и бентонитовых глин для формовочных смесей и противопригарных красок: автореф. дис. канд. тех. наук: спец. 05.16.04 «Литейное производство» / **Е.М. Лесив.** – Челябинск, 2007. – 21 с.
23. **Mekhamer W.K.** The Colloidal Stability of Raw Bentonite Deformed Mechanically by Ultrasound. J Saudi Chem Soc. 2010, 14(3), 301–306.
24. **Christidis, G.E., Dellisanti, F., Valdre, G., Makri, P.,** 2005. Clay Miner. 40, 511–522] i [Bastida, J., Kojdecki, J., Pardo, M.A., Amoros, P., 2006. Clays Clay Miner. 54, 390–401.
25. **Максютова О.** Бентонит и его применение / TheChemicalJournal // – 2017 – №12. – С. 20-25.
26. **Darvishi Z., Morsali A.** Synthesis and Characterization of Nano-bentonite by Sonochemical Method. Ultrason-Sonochem. 2011, 18, 238–242.
27. **Nguyen A.N., Reinert L., Lévêque J.-M., Beziat A., Dehaudt P., Juliaa J.-F., Duclaux L.** Preparation and Characterization of Micron and Submicron-sized Vermiculite Powders by Ultrasonic Irradiation. Appl Clay Science. 2013, 72, 9–17.
28. **Lapides I., Yariv Sh.** The Effect of Ultrasound Treatment on the Particle-size of Wyoming Bentonite in Aqueous Suspensions. J Mater Sci. 2004, 39, 5209–5212.
29. Приготовление высокодисперсных суспензий бентонита в бичастотном кавитационном поле / **П.Г. Думитраш, М.К. Болога, Т.Д. Шемякова** // Электронная обработка материалов. 2015, – Том 51 (№1), – с. 85-91.
30. **Маргулис М.А., Хавский Н.И.** О механизме одновременного воздействия двух частот акустических колебаний на физико-химические и химические эффекты. Всес. симпозиум «Акустическая кавитация и применение ультразвука в химической технологии» (Славское, 26 февр. – 1 марта 1985 г.). Тезисы докладов. Славское, 1985, с. 93–94

Рукопис подано до редакції 06.04.2021

УДК 004.942

Ж.Г. РОЖНЕНКО[·], канд.техн. наук, О.К. ДАНИЛЕЙКО, ст. викл.,
Г.В. КОЛОМІЦ, асист., А.В. ЯТЧУК, студ.
Криворізький національний університет

ВИКОРИСТАННЯ МІКРОПРОЦЕСОРІВ НА БАЗІ ARM CORTEX В ЕЛЕКТРОМЕХАНІЦІ

Мета. Проведений аналіз ринку мікропроцесорів показав поширення використання пристроїв мікроконтролерів на основі ARM Cortex. У роботі на прикладі контролера STM32, побудованого на основі ARM Cortex, розглянуто використання мікропроцесорів у електромеханічних системах. Для більшості сучасних електромеханічних систем необхідні пристрої, що будуть об'єднувати різноманітні пристрої в єдину мережу для обміну даними. В роботі проведено огляд номенклатури мікропроцесорів різних моделей та виробників та обрано найбільш прийнятний варіант відповідно до висунутих вимог.

Також було проведено вибір периферійних пристроїв для роботи в досить поширеній локальній мережі промислової автоматики Modbus.

Методи дослідження. При вирішенні задачі використовуються загальні методи дослідження електромеханічних систем та побудови програм керування для мікроконтролерів, побудови локальних мереж промислової автоматизації.

Наукова новизна. Розроблено програму для прийому та передачі даних до мережі за протоколами Modbus RTU та Modbus TCP для мікроконтролера STM32. Розроблені макети для аналізу роботи мікропроцесора STM32 названих в мережах.

Практична значимість. Можливість використання одного з найбільш поширених сучасних 32-х розрядних мікроконтролерів фірми STMicroelectronics в локальних мережах промислової автоматики з використанням інтерфейсів RS-485 та Ethernet мережевого протоколу Modbus. Використання контролерів STM32 дозволяє зменшити витрати на розробку обладнання при збільшенні швидкодії пристроїв.

Результати. Проведено аналіз сучасного ринку мікроконтролерів, можливостей щодо програмування та конфігурування мікроконтролера; обрано програмне забезпечення, що пришвидшує процес роботи та покращує якість проекту в цілому; проведено вибір обладнання для роботи мікроконтролера з протоколами Modbus; створено відповідні програми керування, що дозволяють обмінюватись даними між різними пристроями за протоколами Modbus TCP та Modbus RTU.

Ключові слова: мікропроцесори, мікроконтролери, ARM Cortex, передача даних, розрядність, пам’ять, програмування, промислові мережі, Keil, HALL, Modbus TCP, Modbus RTU.

doi: 10.31721/2306-5435-2021-1-109-98-106

Проблема та її зв'язок з науковими і практичними задачами. В сучасній електромеханіці та промисловості в цілому широке застосування мають різноманітні напівпровідникові прилади, процесори, контролери та програмовані реле. Якщо більш детально розібратись в їх будові, то стане зрозуміло що їх «серцем» є мікропроцесори. Номенклатура мікропроцесорів дуже різноманітна й охоплює безліч найменувань. Але посеред великої кількості можна виділити мікропроцесори на базі ARM Cortex та AVR Atmel. На даний момент переважна кількість готових приладів побудована саме на даних мікропроцесорах. В даній статті розглянуто основні технічні характеристики, переваги, недоліки та приклади використання мікропроцесорів. За останні роки вартість мікропроцесорів значно зменшилась, тому у вартості готово виробу їх доля дуже мала. Найбільш трудомістким є розробка друкованих плат, корпусів, периферії та звичайно розробка програмного забезпечення.

Розроблено макет для роботи мікропроцесора STM32 в мережах Modbus TCP та Modbus RTU. Складність полягає в тому, що «з коробки» STM32 працює лише з інтерфейсом SPI (англ. Serial Peripheral Interface - последовний периферійний інтерфейс) та UART (англ. Universal Asynchronous Receiver-Transmitter - універсальний асинхронний приймач-передавач). Цілком логічно, що даних технологій недостатньо і виникає необхідність в додатковій периферії – платах розширення. В даній статті проведено вибір необхідної периферії для роботи з Modbus [1].

Аналіз досліджень і публікацій. Аналіз сучасного ринку мікроконтролерів. Загальні поняття про архітектуру ARM

Архітектура ARM – це мікропроцесорна архітектура RISC (Reduced Instruction Set Computing), що розробляється компанією ARM Limited [1]. ARM (Advanced RISC Machines) - один з найбільших в світі розробників і ліцензіарів архітектури RISC-процесорів (ARM), орієнтованих на використання в портативних пристроях з автономним живленням, а також в серверах і комп'ютерах. Станом на 2020 рік сімейство ARM процесорів є найпопулярнішим серед інших ЦПУ. Компанія ARM спочатку була заснована як спільне підприємство між Acorn Computers, Apple Computer і VLSI Technology. З 2016 року належить японської телекомунікаційної та інвестиційної корпорації SoftBank. Центральний офіс знаходиться в Кембриджі, Великобританія. Офіси ARM розташовані по всьому світу, включаючи Францію, Індію, Швецію і США [1].

Процесори архітектури ARM переважно використовуються в енергоефективних пристроях.

Процесори архітектури ARM за ліцензією випускають такі всесвітньо відомі компанії: Apple, Atmel, Broadcom, Freescale, Marvell, Nvidia, Qualcomm, Samsung, Texas Instruments, VIA, Міландр, Елвіс, STMicroelectronics ARM limited та інші [1]. Особливістю ARM є те, що це не просто один процесор. Як правило, в нього входять: контролер оперативної пам'яті, графічний прискорювач, відеодекодер, аудіокодек і модулі бездротового зв'язку. Таку систему прийнято називати однокристальною.

На сьогоднішній день ARM налічують кілька процесорних поколінь: ARM9. Чіпи ARM9 можуть досягати тактової частоти 400 МГц. Ці чіпи морально застаріли, але як і раніше корис-

туються попитом. Наприклад, в бездротових роутерах і банківських термінала. Набір простих команд такого чіпу дозволяє працювати з декількома Java-додатками [2]. ARM11. Процесори ARM11 мають більш повний набір простих команд, які розширюють їх функціонал і мають відносно високу тактову частоту (близько 1 ГГц). Завдяки невисокому енергоспоживанню і низькій собівартості чіпи ARM11 встановлюють в смартфонах початкового рівня [2]. ARMv7. Сучасні чіпи архітектури ARM належать до сімейства ARMv7. Розроблені безпосередньо ARM Limited процесорні ядра належать до лінійки Cortex і більшість виробників однокристальних систем використовують їх без істотних змін [2]. ARM Cortex-A8. Історично першим процесорним ядром сімейства ARMv7 було Cortex-A8, яке є основою таких відомих на свій час SoC як Apple A4 (iPhone 4 та iPad) і Samsung Hummingbird (Samsung Galaxy S і Galaxy Tab). Дане рішення демонструє приблизно вдвічі вищу продуктивність у порівнянні з попереднім ARM11, але і більш високе енергоспоживання, що робить даний чіп нині вкрай непопулярним [2]. ARM Cortex-A9. Слідом за Cortex-A8 компанія ARM Limited представила нове покоління чіпів - Cortex-A9, яке зараз є найпоширенішим і займає середню цінову нішу[3].. Продуктивність ядер Cortex-A9 зросла приблизно втричі в порівнянні з Cortex-A8, та з'явилася можливість об'єднувати їх по два або навіть чотири на одному чіпі [1]. ARM Cortex-A5 і Cortex-A7. При проектуванні процесорних ядер Cortex-A5 і Cortex-A7 компанія ARM Limited добились компромісу між мінімальним енергоспоживанням ARM11 і прийнятним швидкодією Cortex-A8. Також є можливість об'єднання ядер по два-чотири - багатоядерні чіпи Cortex-A5 і Cortex-A7 [2]. ARM Cortex-A15. Процесорні ядра Cortex-A15 стали логічним продовженням Cortex-A9 зі збільшеною швидкодією.

Незважаючи на те, що абревіатура ARM розшифровується як «Advanced RISC Machine» («передова RISC-машина»), з самого початку це була не зовсім типова RISC-архітектура [1]. З одного боку, для RISC'а вона мала не багато регістрів загального призначення, зараз у більшості різновидів архітектури їх 31 штука, проте програмісту доступні лише 16 з них, причому три регістра мають спеціальні функції; таким чином, « справжніх » регістрів загального призначення з точки зору програміста всього 13, в той час як цілий ряд « справжніх » RISC-процесорів мають їх більше сотні.

З іншого боку, унікальною особливістю цієї архітектури була можливість виконання будь-якої команди при дотриманні заданої умови, відсутня не тільки у інших RISC, а й у CISC-процесорів (безумовно команди в наборі ARM з'явилися лише у версії ARMv5) [1]. Крім того, в командах обробки даних в ряді випадків можливе об’єднання виконання основної операції (наприклад, складання) із зсувом.

Команди читання і запису пам'яті у ARM розташовують розвиненим набором видів адресації, який перевершує за своїми можливостями не тільки RISC, а й основну масу CISC-процесорів. Сучасні 32-розрядні процесори архітектури ARM здатні обробляти 8-, 16- та 32-розрядні цілі числа, а процесори архітектури ARMv8-A - 64-розрядні. Можливість обробки дійсних чисел є необов'язковою і досягається за допомогою субпроцесорів (які розташовані на одному кристалі з власне центральним процесором і з точки зору прикладного програміста не відрізняються від нього). Старші моделі здатні працювати з віртуальною пам’яттю, оскільки мають необхідний для цього пристрій управління пам'яттю (MMU) [3]. Простіші вироби не мають MMU, але в деяких випадках мають пристрій захисту пам'яті (MPU), що не дозволяє реалізувати віртуальну пам'ять, але забезпечує необхідну підтримку апаратного захисту одних виконуваних процесором завдань від інших.

Постановка задачі. Розробка макету для роботи мікропроцесора STM32 в мережах Modbus TCP та Modbus RTU. Складність полягає в тому, що «з коробки» STM32 працює лише з інтерфейсом SPI (англ. Serial Peripheral Interface - последовний періферійний інтерфейс) та UART (англ. Universal Asynchronous Receiver-Transmitter - універсальний асинхронний приймач-передавач). Цілком логічно, що даних технологій недостатньо і виникає необхідність в додатковій периферії – платах розширення. В даній статті буде проведено вибір необхідної периферії для роботи з Modbus.

Викладення матеріалу та результати. Найбільш цікавими з технічної та економічної точки зору є мікроконтролери на базі ядра ARM, а саме STM32. Саме на базі STM32 побудовано велика кількість плат які тестуються, вони спрощують створення нових пристроїв. Взаємодія з кожним блоком мікроконтролера відбувається за допомогою регістрів [6].

Для програмістів, мікроконтролер є об’ємом пам'яті. У мікроконтролері розміщені також ОЗП, FLASH і EEPROM, але і програмні регістри. Кожному апаратному регістру відповідає комірка пам'яті. Так, аби вписати дані в регістр або розвидніти його значення, програмісту необхідно звернутися до відповідної комірки адресного простору [6].

Кожен з регістрів має свій номер - адресу. Адреса регістра позначається 32-бітовим числом поданих в шістнадцятковій системі числення. Шляхом записи за адресою регістра певної комбінації одиниць і нулів, які зазвичай представлені в шістнадцятковому вигляді, здійснюється настройка і управління тим чи іншим вузлом в МК [6].

Відомо, що людина має деякі особливості сприйняття. Так наприклад, символьні назви сприймаються краще, ніж адреси пам'яті. Це досить помітно, коли використовується велика кількість комірок пам’яті. У мікроконтролерах при використанні ARM кількість регістрів перевищує тисячі. Щоб покращити роботу, необхідно провести позначення символьними покажчиками. Це позначення проводиться на рівні з назвою CMSIS [6].

ARM CMSIS - це незалежний від виробника рівень апаратної абстракції для серії ядер Cortex-M, а також інтерфейс відладчика (англ. Debugger). CMSIS надає послідовні і прості інтерфейси для ядра, його периферії і операційних систем реального часу [6].

Стандартна бібліотека периферії - це набір низькорівневих драйверів. Кожен драйвер надає користувачеві набір функцій для роботи з периферійним блоком. Таким чином користувач використовує функції, а не звертається безпосередньо до регістрів. При цьому рівень CMSIS виявляється прихованим від програміста [6].

Існує декілька способів програмування та конфігурації даних мікроконтролерів. Програмний код може бути написаний на таких мовах програмування: Асамблер, C/C++, Pascal, BASIC та декілька візуальних мов. На даний момент більшість розробників працюють з C/C++ через більшу зручність та наглядність. Щодо конфігурації, то прийнято використовувати програми-конфігуратори, які дозволяють працювати не напряму з регістрами пам’яті, а використовувати символьні імена. Тобто на даний момент створено велику кількість рішень, що дозволяють значно спростити та пришвидшити програмування та розробку пристроїв в цілому.

Створення програми реалізації мережевого протоколу Modbus для мікроконтролерів STM32. Особливо важливим при розробці мікроконтролерних систем постає питання зв’язку окремих частин її між собою та з керуючими комп’ютерами, тобто локальних мереж. На жаль, на відміну від програмованих логічних контролерів, мікроконтролери «на борту» мають тільки невелику кількість спеціалізованих інтерфейсів зв’язку, що значно обмежує використання їх у системах промислової автоматики та керування.

Тому актуальним питанням є розробка програмного забезпечення орієнтованого на поширені в сучасних системах мережеві протоколи. Так будо прийнято рішення створити програму реалізації мережевого протоколу Modbus для мікроконтролерів STM32.

Modbus - це універсальний протокол, для забезпечення комунікаційного зв'язку, широко використовуваний в промислової автоматизації. Даний протокол забезпечує зв'язок по типу майстер-slave, для нього характерна простота, відкритість і масовість [11].

Саме завдяки цим особливостям даний протокол широко використовується з 1979 року. Сучасна автоматизація випускає велику кількість датчиків, контролерів і модулів, що працюють на базі протоколу Modbus RTU. Він дозволяє забезпечувати якісне управління обладнанням і контроль за його безперебійним функціонуванням на підприємствах різного призначення.

Нагадаємо, що в структурі Modbus передбачено тільки один головний пристрій зі статусом «master», що задає команди і є ведені зі статусом «slaves». Взаємозв'язок між master та slave здійснюється за ініціативою master. Тобто контролюючий пристрій посилає сигнал або робить запит на отримання даних від підлеглого по типу: «запит-відповідь». При цьому кожне повідомлення від майстра, спрямоване до slave є пакетом даних, що складається з адреси slave пристрою, коду функції, даних і контрольної суми.

Використання даного протоколу дозволяє реалізувати один загальний формат передачі повідомлень. Протокол Modbus реалізується в 3-х режимах: RTU, ASCII, TCP [11].

Реалізація цих протоколів з використанням мікроконтролерів STM32 значно розширює їх сферу застосування. Дані рішення дозволяють знизити вартість готових пристроїв де є необхідність роботи в промислових мережах. А необхідність в таких пристроях виникає досить часто, особливо під час часткового оновлення обладнання на промислових підприємствах. При таких

модернізаціях сучасне обладнання, що має підключення до мережі працює сумісно з застарілим обладнання, що не має таких мережевих можливостей. Тому реальна необхідність в таких пристроях дійсно існує.

В статті розглянутий приклад створення мережі на основі комунікаційного інтерфейсу RS485 та протоколу Modbus RTU. Налаштування та перевірку програми зроблено на платі розробника STM32F103C8T6 «Blue pill» з відповідним контролером. Для програмування «Blue pill» використовувався зовнішній програматор ST-Link V2.

Мікроконтролери STM32 не мають у своєму складі інтерфейсу RS-485, тому до контролера треба підключити модуль MAX485, UART-RS485, який є проміжним між звичайним виходом UART мікроконтролера та мережею RS485 (рис. 1).

Характеристики модуля:

мікросхема перетворювача – MAX485;

вхідний інтерфейс – UART;

вихідний інтерфейс – RS485;

напруга живлення: 5В [15].

Даний модуль може бути безпосередньо підключений до більшості типів мікроконтролерів за стандартним UART інтерфейсом. Не зважаючи на те, що напруга живлення модуля 5 вольт, він може бути підключений до мікроконтролерів з напругою живлення 3,3 вольта без проміжних перетворювачів рівнів.

При використанні протоколу Modbus для підключення тільки одного пристрою, можна використати інтерфейс RS232. Для підключення його до комп'ютера можна використати USB-UART міні-перехідник на мікросхемі CP2102 (рис. 1).

При використанні інтерфейсу RS485 треба звернути увагу на керування мікросхемою приємопередавача (рис. 2).

Таким чином, треба передбачити керування входами DE та RE при переході від прийому до передачі. Схема з'єднання модуля та плати мікроконтролера наведена на рис. 3 (виводи порту PA5 та PA7 використовуються для керування приємопередавачем).

Рис. 1. Модуль MAX485 UART-RS485

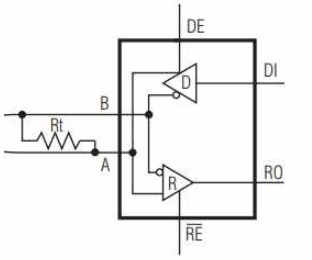

Рис. 2. Схема приємопередавача RS485

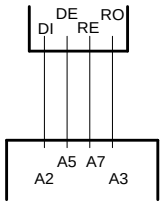

Рис. 3. Схема з'єднання модуля та плати мікроконтролера

В проекті використаємо конфігуратор Cube MX та середовище розробки проекту Keil MDK-ARM V5.27, проект орієнтовано на бібліотеку HAL.

Запустимо Cube MX.

Стандартно обираємо потрібний процесор (STM32F103C8T6).

Послідовно налаштовуємо для розділу SYS обираємо Debug – Serial Ware (рис. 4).

Для розділу RCC обираємо Crystal Ceramic Resonator – зовнішній кварц (рис. 5).

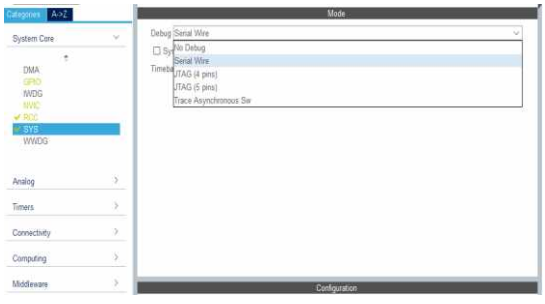

Рис. 4. Налаштування Debug

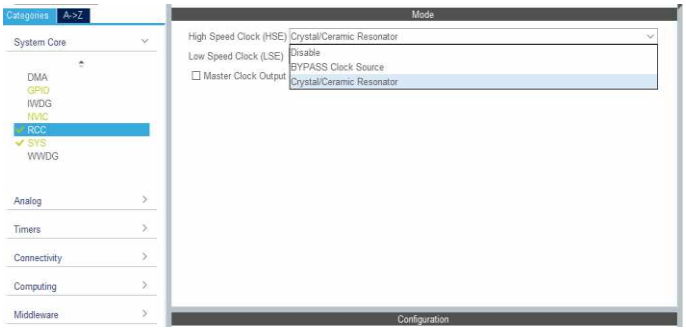

Рис. 5. Налаштування резонатора

Налаштовуємо USART2 (рис. 6).

Налаштуємо контакти PA5 та PA7 на вихід. Таким чином призначення контактів мікроконтролера наведені на рисунку 7.

Відповідно до зовнішнього кварцу налаштуємо синхронізацію (рис. 8).

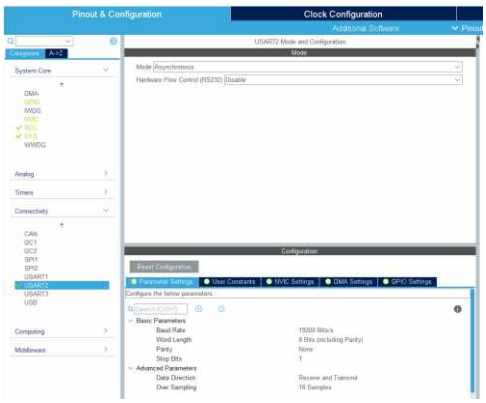

Рис. 6. Налаштування порту USART2

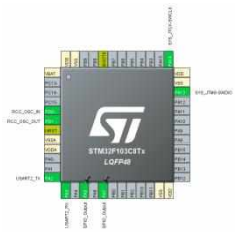

Рис. 7. Призначення контактів мікроконтролера

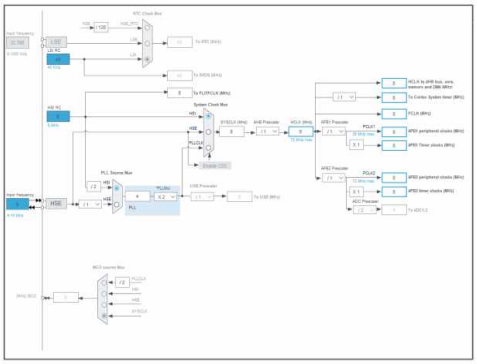

Рис. 8. Налаштування синхронізації

Налаштуємо властивості проекту (рис. 9).

Після генерації отримаємо проект для Keil, в який підключено всі додаткові файли (рис. 10).

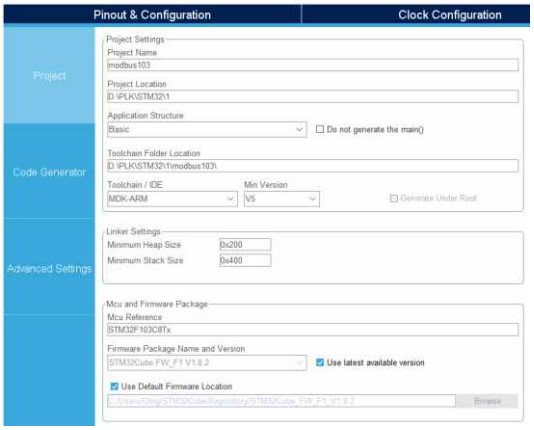

Рис. 9. Властивості проекту

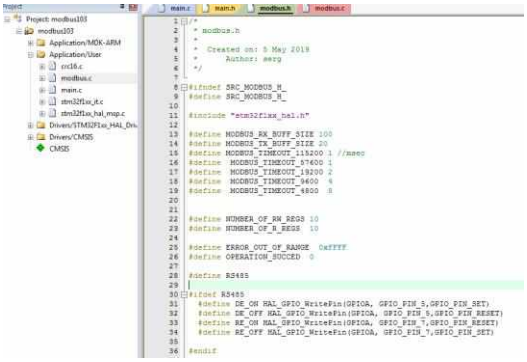

Рис. 10. Структура проекту та файл modbus.h

У файлі crc16.c розміщені функції розрахування контрольної суми, у файлі modbus с функції протоколу modbus.

У секції #ifdef RS485 визначені контакти керування передавачем (DE) та приймачем (RE), слід звернути увагу, що вони потрібні бути у проти фазі.

У файлі main.c в головній функції void main(void) за допомогою функції i32SetModbusAdress(5); // 5 adres device

задається адрес slave пристрою (нашому прикладі 5), а за допомогою функції i32ModbusSetUart(&huart2); // number UART

номер використаного UART порту (у нашому прикладі 2).

У нескінченій частині (цикл while (1)) треба запустити функцію мережевого обміну – vModbusProtocol(). Для читання значення з регістру використовується функція u16GetModbusRwReg(r) де r номер відповідного регістру, для запису – u16SetModbusRwReg(r, d), де r номер відповідного регістру, а d – змінна в яку це значення заноситься.

Для прикладу читаємо число з регістру 2, збільшуємо його на 3 та записуємо його в регістр 3.

Аналізуємо стан регістру 0, та якщо він дорівнює 12 записуємо у регістр 1 число 10, у протилежному стані записуємо число 25 (рис. 11).

У якості ведучого пристрою обираймо програмований логічний контролер Овен ПЛК-100. Зупинимось тільки на налаштуванні мережі в ПЛК. Зрозуміло, що ПЛК є майстер пристроєм. Тому при конфігурації додаємо ModBus (Maser) пристрій з інтерфейсом RS485-1 пристроєм (рис. 12).

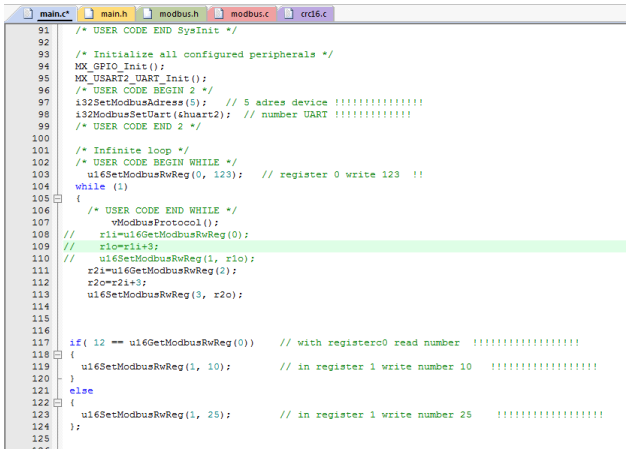

Рис. 11. Циклічна частина програми

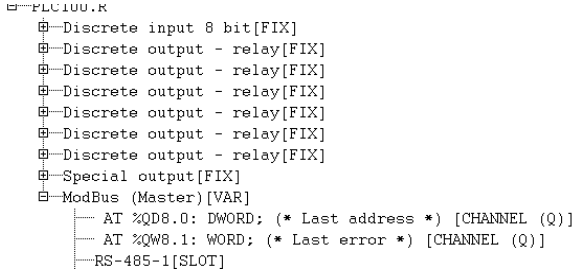

Рис. 12. ModBus (Maser) пристрій

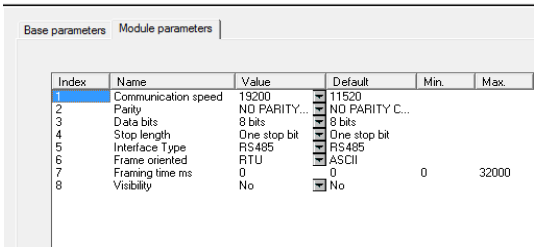

Рис. 13. Налаштування RS485-1 [SLOT]

Зрозуміло, що треба налаштувати параметри RS485-1 [SLOT], які відповідають налаштуванню USART2 мікроконтролера (рис. 13).

Після цього треба налаштувати Slave пристрій, для чого додаємо Universal Modbus device з відповідними параметрами налаштування, з яких треба задати адресу Slave пристрою – 5 для нашого прикладу (рис. 14, 15) .

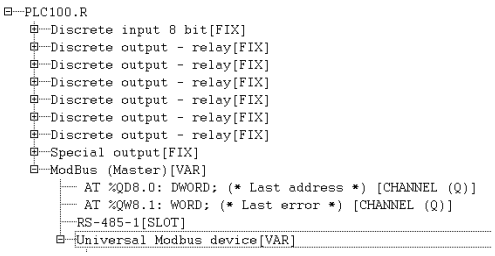

Рис. 14. Universal Modbus device

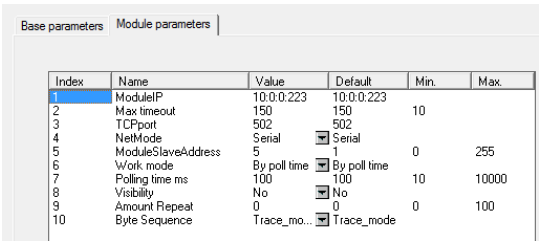

Рис. 15. Налаштування Universal Modbus device

Після цього залишається додати два Register output module з адресами регістрів 0 та 2 та регістри Register input module з адресами регістрів 1 та 3 (рис. 16, 17).

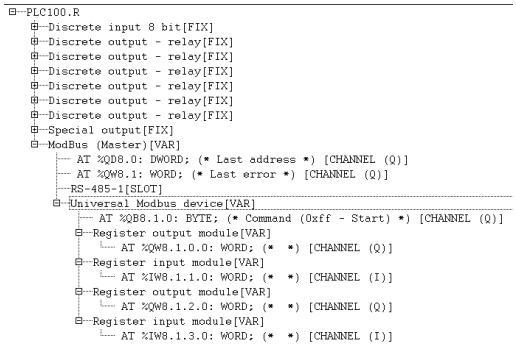

Рис. 16. Налаштування Register output module та Register input module

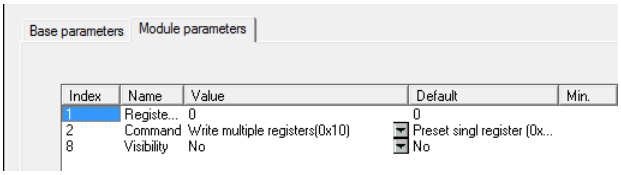

Рис. 17. Налаштування Register output module адресом регістра 0

Слід звернути увагу, що для запису у регістр використовується 16 функція Modbus, а для читання – 3.

Після завантаження проекту у ПЛК отримуємо стан мережевих змінних (рис. 18, 19).

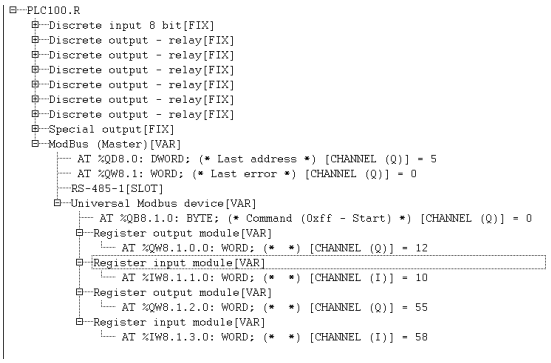

Рис. 18. Стан мережевих змінних

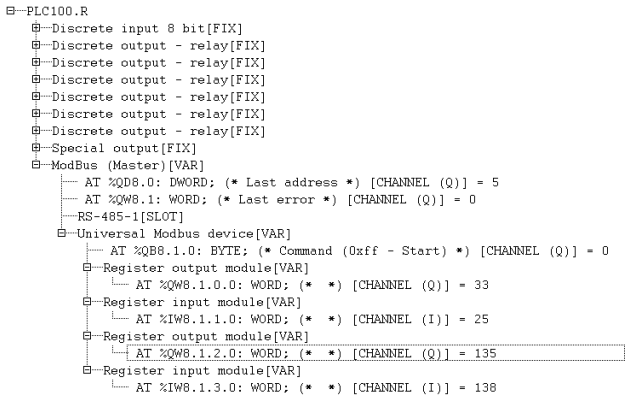

Рис. 19. Стан мережевих змінних

Як видно з наведених рисунків значення повністю відповідають наведеному вище алгоритму, а використані програмні та апаратні рішення повністю дієздатні та їх можна використовувати за основу при побудові готових пристроїв, в яких є необхідність роботи в мережах Modbus.

Висновки і напрямок подальших досліджень. Проведено аналіз сучасного ринку мікроконтролерів, наведено дані основних представників та їх порівняльні характеристики; при проектуванні та розробці нових пристроїв слід використовувати мікроконтролери, що мають характеристики, які перевищують ті що необхідні для проекту на даний момент; в якості досліджуваного обрано мікроконтролер STM32F103C8T6 як такий, що має найбільш привабливі характеристики при досить низькій ціні; проведено аналіз можливостей щодо програмування та конфігурування мікроконтролера, обрано програмне забезпечення, що пришвидшує процес роботи та покращує якість проекту в цілому; використання STM32F103C8T6 дозволяє використовувати велику кількість сумісної периферії, серед яких й ті, що дозволяють працювати в мережах з інтерфейсами Ethernet та RS-485; проведено вибір обладнання для роботи мікроконтролера з протоколами Modbus; створено відповідні програми керування, що дозволяють обмінюватись даними між різними пристроями за протоколами Modbus TCP та Modbus RTU.

Список літератури

1. Справочник по электронным компонентам: Архитектура и система команд RISC-процессоров семейства ARM. [Електронний ресурс]. – 2019. – Режим доступу: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/index.htm.

2. Справочник по электронным компонентам: Ознакомительное руководство по ARM-микроконтроллерам Cortex-M3. [Електронний ресурс]. – 2019 Режим доступу доступу: http://www.gaw.ru/html.cgi/txt/doc/micros/arm/cortex_arh/index.htm.

3. Pete's Pages ARM Assembly Language Programming — Chapter 2. Inside the ARM M3. [Електронний ресурс]. – 2016. – Режим доступу: http://www.peter-cockerell.net/aalp/html/ch-2.html.

4. ATmega8U2 ATmega16U2 ATmega32U2. [Електронний ресурс]. – 2012. –Режим доступу: https://ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf

5. Arduino Official Store - Arduino Uno Rev3 [Електронний ресурс]. – 2020. – Режим доступу: https://store.arduino.cc/arduino-uno-rev3

6. PM0056 Programming manual. [Електронний ресурс]. – 2017. – Режим доступу: https://www.st.com/resource/en/programming_manual/cd00228163-stm32f10xxx20xxx21xxxl1xxxx-cortexm3-programming-manual-stmicroelectronics.pdf

7. TN0516 Technical note. [Електронний ресурс]. – 2014. – Режим доступу: https://www.st.com/resource/en/technical_note/dm00026481-overview-of-the-stm32f0xf100xxf103xx-and-stm32f2xxf30xf4xx-mcus-pmsm-singledual-foc-sdk-v40-stmicroelectronics.pdf

8. RM0008 Reference manual. [Електронний ресурс]. – 2018. – Режим доступу: https://www.st.com/resource/en/reference_manual/cd00171190-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf

9. The Insider`s Guide To The STM32 ARMBased Microcontroller Hitex (UK) Ltd. [Електронний ресурс]. – 2008. – Режим доступу: http://www.emcu.it/InsideCORTEX-1221142709.pdf

10. STM32L4 Ecosystem CubeMX Tool. [Електронний ресурс]. – 2017. – Режим доступу: https://www.st.com/content/ccc/resource/training/technical/product_training/17/8e/5a/35/36/4e/4e/27/STM32L4_Ecosystem_CubeMX_Tool.pdf/files/STM32L4_Ecosystem_CubeMX_Tool.pdf/jcr:content/translations/en.STM32L4_Ecosystem_CubeMX_Tool.pdf

11. Как общаются машины: протокол Modbus. [Електронний ресурс]. – 2019. – Режим доступу: https://habr.com/ru/company/advantech/blog/450234/

12. MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3. [Електронний ресурс]. – 2012. – Режим доступу: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
13. ENC28J60 Data Sheet Microchip 1b3. [Електронний ресурс]. – 2004. – Режим доступу: http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf
14. CP2102/9 Data Sheet Silicon Labs. [Електронний ресурс]. – 2017. – Режим доступу: https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf
15. Datasheet MAX481/MAX483/MAX485/MAX487–MAX491/MAX1487 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers. [Електронний ресурс]. – 2014. – Режим доступу: https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf

Рукопис подано до редакції 08.04.2021

УДК 622.765

А.Ю. КРИВЕНКО[·], канд. техн. наук, ст. викл., Т.А. ОЛІЙНИК, д-р техн. наук., проф.
Криворізький національний університет
Т.А. КРИВЕНКО, викл., Гірничий фаховий коледж КНУ

ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ МАСОПЕРЕНОСУ ПРИ ЗНЕШЛАМЛЕНІ ЗАЛІЗОРУДНОЇ СУСПЕНЗІЇ

Мета. Дослідження питань гідравлічного збагачення залізорудної сировини в радіальних згущувачах типу МД9, а також розв'язок їх методами математичного моделювання.

Методи дослідження. Використання методів дослідження, таких як: теорії ймовірності, теорії інформації й подоби, математичного моделювання, загальноприйнятих законів гідравліки й гідродинаміки.

Наукова новизна. Дослідження потоку пульпи з живильного пристрою апарата, дозволяє одержати залежності швидкості руху потоку пульпи й вмісту твердого від параметрів пристрою подачі вихідного живлення. Отримані залежності дозволяють у значній мірі зменшити негативний вплив затопленого струменя на гідравлічне збагачення залізорудної сировини. Застосовуючи відомі закони гідродинаміки, була побудована математична модель гідравлічного поділу залізорудної сировини на виході з пристрою живлення апарата, відповідно до якої при виборі структури моделі використовуються теоретичні передумови. Розрахунковими формулами отримана залежність швидкості переміщення часток твердої фази потоку живлення пульпи від конструктивних особливостей радіального згущувача. Представлені залежності дозволяють варіювати необхідними параметрами процесу з урахуванням конструктивних особливостей апарата й пристрою подачі живлення, з метою одержання згущеного продукту заданої якості.

Практична значимість. Удосконалення технології збагачення залізних руд за рахунок підвищення ефективності гравітаційного збагачення в радіальних згущувачах типу МД9, у розробці нового способу формування вихідного живлення апарата.

Результати. Проведене математичне моделювання шляхом застосування теоретичної моделі гравітаційного гідравлічного поділу залізорудної сировини в радіальному згущувачі. У результаті цього отримані рівняння, які дозволили суттєво скоротити число параметрів, які впливають на протікання досліджуваного процесу. Обчислювальний експеримент, за результатами моделювання процесу гідравлічного гравітаційного поділу залізорудної сировини у радіальному згущувачі типу МД9, дозволив вивчити зміну відповідних параметрів і вплив їх на процес знешламлення. Як наслідок, на практиці виникають питання про знаходження кількісних співвідношень. Для ріщення цих питань необхідно проведення експериментальних досліджень на рельних радіальних згущувачах, що функціонують, з метою збору матеріалу для оцінки величин параметрів, які входять у математичні моделі.

Ключові слова: дешламатор, дешламація, МД9, радіальний згущувач.

doi: 10.31721/2306-5435-2021-1-109-106-111

Проблема та її зв'язок з науковими і практичними завданнями. На сучасних ГЗК досягти якісних показників сепарації рудної й породної складових при збагаченні руд неможливо без якісної підготовки здрібнених продуктів до магнітного збагачення. Для створення оптимальної умови для магнітного збагачення руди на ГЗК використовується дешламація рудної суспензії. Аналіз роботи знешламлюючих апаратів показав, що ефективність процесу поділу часток пульпи визначається параметрами, які залежать від властивостей матеріалу що поділяється, властивостей розділового середовища, конструкції й принципу дії застосовуваного апарату.

Підвищення якості магнетитових концентратів гірничо-збагачувальних комбінатів забезпечує їх значну конкурентоспроможність на внутрішньому й на зовнішньому ринках. Досягти це