

На розробленому стенді виконується лабораторна робота студентами спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» з дисципліни «Основи енергетичної грамотності». Стенд дозволяє проаналізувати режими роботи при шунтуванні фазних або нульових проводів, обриві нульового проводу тощо.

При виконанні лабораторної роботи студенти можуть в повному обсязі зняти параметри роботи лічильника електроенергії. З отриманих даних зрозуміло, що є ймовірність крадіжці електроенергії, але реалізувати втручання в схему підключення щоб цього не було видно неможливо.

Данилейко О.К.
старший викладач;

Мрачковський Д.В.
студент;

Рожненко Ж.Г.
*кандидат технічних наук, доцент,
Криворізький національний університет*

РЕАЛІЗАЦІЯ МЕРЕЖІ ПО ПРОТОКОЛУ *MODBUS RTU* НА БАЗІ МІКРОКОНТРОЛЕРА *STM32*

В сучасній електромеханіці та промисловості широке застосування мають різноманітні програмовані пристрої автоматики – 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 32-х розрядні мікропроцесори на основі ядра *ARM Cortex*.

За останні роки вартість мікропроцесорів значно зменшилась, тому у вартості готового виробу їх доля дуже мала. Найбільш трудомістким є розробка друкованих плат, корпусів, периферії та звичайно розробка програмного забезпечення.

Наразі дуже велике поширення набули мікропроцесори сімейства *STM32* фірми *STMicroelectronics*.

Але «з коробки» *STM32* працює лише з інтерфейсом *SPI*, *CAN*, *USART* та *I2C*, що дуже обмежує мережеве використання цих контролерів. Тому була поставлена задача реалізувати на *STM32* поширений в системах промислової автоматики мережевий протокол. В якості цього протоколу був обраний протокол *Modbus*.

Modbus – це універсальний протокол, для забезпечення комунікаційного зв'язку, широко використовуваний в промисловій автоматизації. Даний протокол забезпечує зв'язок по типу *master-slave*, для нього характерна простота, відкритість і масовість.

Саме завдяки цим особливостям даний протокол широко використовується з 1979 року. Сучасна промисловість засобів автоматизації випускає велику кількість датчиків, контролерів і модулів, що підтримують протоколи *Modbus*. Цей протокол дозволяє забезпечувати якісне управління обладнанням і контроль за його безперерйним функціонуванням на підприємствах різного призначення.

Для реалізації цієї задачі була використана плата розробника *STM32F103 «Blue Pill»* на основі мікропроцесора *STM32F103C8T6*. Оскільки плата не має мережевих інтерфейсів, які можна використати для протоколу *Modbus RTU*, потрібні додаткові інтерфейсні модулі (або *RS232* або *RS485*) які підключаються до виводів *USART* плати «*Blue Pill*».

Якщо необхідне з'єднання тільки з одним пристроєм можна використовувати модуль *RS232* на базі мікросхеми *CP2102*. Для повнофункціональної мережі використовувався модуль *RS485* на базі мікросхеми *MAX485*. 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 вольт без проміжних перетворювачів рівнів. При використанні модуля з *MAX485* слід керувати входами *DE* та *RE* при переході від прийому до передачі.

Для завантаження програми в мікроконтролер потрібен зовнішній програматор, так як плата не має вбудованого (використаємо зовнішній програматор *ST-LINK2*)

Програмування здійснюється на мові C. Для конфігурації контролера використовуємо програму конфігуратор *STM32 CubeMX* з використанням бібліотеки *HAL*. Для розробки програми та завантаження її використовується система програмування *Keil MDK-ARM V5.27*.

Вигляд призначення контактів мікроконтролера після конфігурації в програмі *STM32 CubeMX* наведений на рисунку 1.

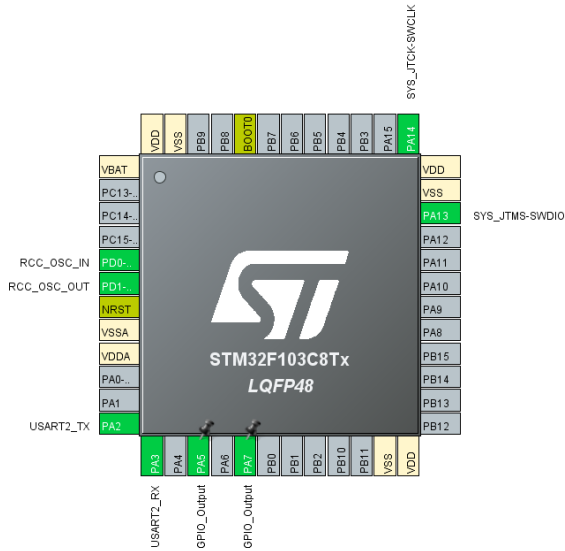


Рис. 1. Призначення контактів мікроконтролера

Джерело: розробка авторів

Після завершення роботи з *STM32 CubeMX* відкриваємо створений конфігуратором проект у середовищі *Keil MDK-ARM*. У проект додаються файли *crc16.c* та *modbus.c* в яких запрограмовані розрахунок контрольної суми та утиліти *modbus*. Також завантажуються та підключаються відповідні *include* файли. Дерево проекту та частину головного файлу *main.c* наведено на рисунку 2.

У нескінченній 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.

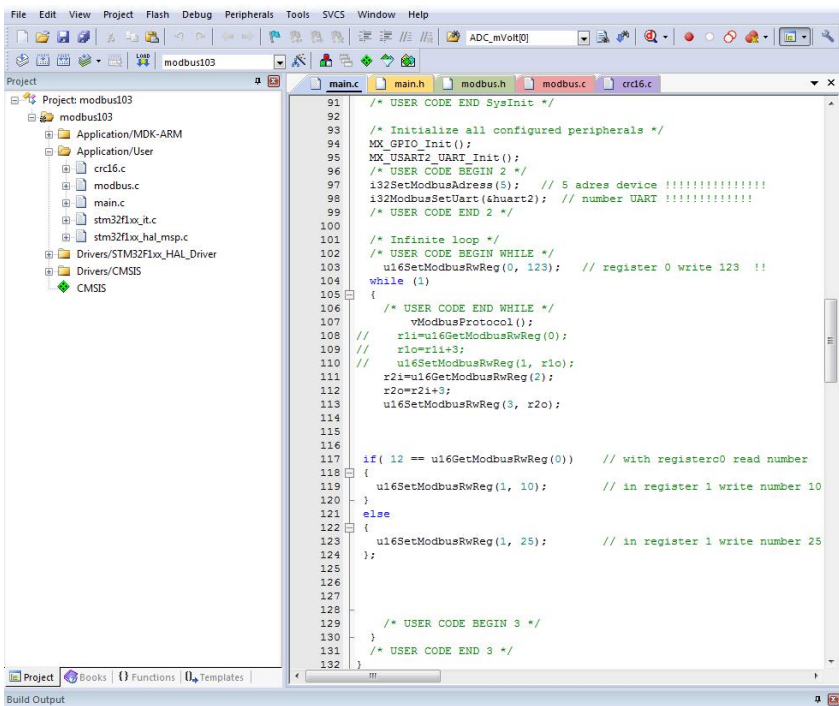


Рис. 2. Дерево проекту та частина головного файлу *main.c*

Джерело: розробка авторів

У якості ведучого пристрою обираймо програмований логічний контролер Овен ПЛК-100 який з'єднано з «Blue Pill» через інтерфейс RS485. Налаштування ПЛК-100 виходить за рамки цієї роботи та розглядати його не будемо.

Робота «Blue Pill» з ПЛК-100 показала, що обмін даними здійснюється відповідно заданому алгоритму.