

Борисов Владислав Витальевич, Магистрант,
Алтайский государственный университет,
г. Барнаул

РАЗРАБОТКА MODBUS-CAN ПРЕОБРАЗОВАТЕЛЯ DEVELOPMENT OF A MODBUS-CAN CONVERTER

Аннотация: Данная работа направлена на разработку собственного устройства Modbus-CAN преобразователя, которое бы обеспечивало надежную связь и обмен данными между такими протоколами связи как Modbus и CAN.

Ключевые слова: Modbus, CAN, выбор комплектующих, разработка печатной платы, работа с сенсорной панелью управления.

Abstract: This work is aimed at developing a proprietary Modbus-CAN converter device that would provide reliable communication and data exchange between communication protocols such as Modbus and CAN.

Keywords: Modbus, CAN, choice of components, development of a printed circuit board, operation with a touch control panel.

Введение

Modbus [1] – это один из наиболее распространенных и надежных протоколов передачи данных, который активно используется в системах управления для связи между устройствами на производстве. CAN [2], в свою очередь, представляет собой высокоскоростной протокол, известный своим быстрым откликом и надёжностью, и чаще всего применяется в автомобилестроении и машиностроении, где требуется оперативный обмен данными и устойчивость к помехам.

Проблема, с которой сталкиваются компании при построении таких систем, заключается в отсутствии прямой совместимости между этими протоколами. Преобразователь Modbus-CAN позволяет объединить устройства с разными стандартами, обеспечивая их эффективное взаимодействие.

Настоящая работа направлена на разработку устройства Modbus-CAN преобразователя, способного выполнять задачу интеграции двух популярных протоколов Modbus и CAN (Controller Area Network). Для достижения поставленной цели были сформулированы и решены следующие задачи:

1. Определены составные части устройства, из которых будет состоять Modbus-CAN преобразователь.
2. Разработана и изготовлена печатная плата устройства, предназначенная для осуществления обмена данными между интерфейсами. Обоснован выбор компонентов входящих в состав данной платы.
3. Разработан и настроен графический пользовательский интерфейс, предназначенный для визуализации получаемых параметров и осуществления управления печатной платой.

Структурная схема преобразователя Modbus-CAN

Перед тем как приступить к разработке устройства, нужно определиться с его составными частями (рис.1). Преобразователь будет состоять из двух составных частей:

1. СПУ (сенсорная панель управления) – необходима для управления платой преобразователя и отображения информации, получаемой по протоколу Modbus.
2. Плата преобразователя – осуществляет получение информации и управление внешним устройством по CAN шине.

Плата преобразователя получив сообщение по CAN шине, производит его обработку, после чего сохраняет полученные данные в соответствующие регистры данных. Панель совершает запрос на чтение регистров данных у платы преобразователя, получая в ответ сами данные.



Чтобы осуществлять управление внешним устройством, СПУ отправляет запрос на запись управляющих регистров в плату преобразователя. После чего плата осуществляет соответствующее воздействие на внешнее устройство.

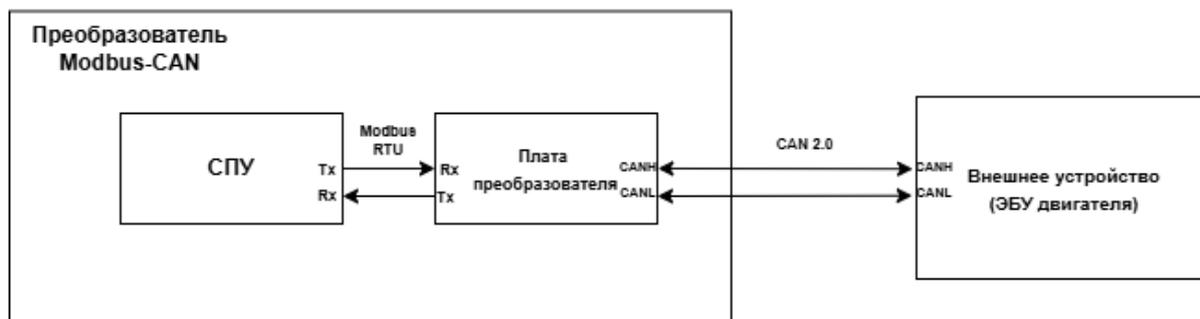


Рис. 1. Структурная схема Modbus-CAN преобразователя

Выбор микроконтроллера

Для разработки Modbus-CAN преобразователя можно использовать различные микроконтроллеры, которые поддерживают как интерфейс CAN, так и протокол Modbus. Вот несколько вариантов микроконтроллеров, которые рассматривались для реализации такого устройства:

1. Microchip PIC32:

Микроконтроллеры серии PIC32 от Microchip поддерживают интегрированные контроллеры CAN и обладают достаточными возможностями для реализации как CAN, так и Modbus протоколов. Представителями данной серии являются такие микроконтроллеры как PIC32MX534F064H [3], PIC32MX575F512H.

2. STMicroelectronics STM32:

Микроконтроллеры серии STM32 предлагают широкий спектр устройств с поддержкой CAN. Большинство моделей STM32 имеют встроенный CAN-контроллер и могут работать с Modbus RTU через UART, или с Modbus TCP через Ethernet.

Модели, такие как STM32F103 [4], STM32F407, STM32F746 и другие из серии STM32 обеспечивают высокую производительность для реализации таких протоколов.

3. NXP (Freescale):

Микроконтроллеры серии LPC от NXP поддерживают CAN и могут работать с Modbus RTU или TCP. Особенно популярны модели, такие как LPC1768 и LPC4078 [5], которые оснащены встроенными CAN-модулями и имеют достаточную производительность для выполнения преобразования Modbus в CAN.

На основании анализа технических характеристик исследуемых линеек микроконтроллеров, представленных в таблице 1, был произведен выбор микроконтроллера STM32F103C8T6 для использования в плате преобразователя. Микроконтроллер обладает полным набором необходимых интерфейсов и возможностью расширения функциональности за счет наличия свободных выводов. Он характеризуется достаточной тактовой частотой и более низкой стоимостью по сравнению с аналогичными продуктами на рынке.

Таблица 1

Обоснование выбора микроконтроллера

Название микроконтроллера	PIC32MX534F064H	STM32F103C8T6	LPC4078FBD144
Частота, мГц	80	72	120
Размер ОЗУ, kb	16	20	98
Размер ПЗУ, kb	64	64	512
Рабочее напряжение, В	от 2.3 до 3.6	От 2 до 3.6	От 2.4 до 3.6
Поддерживаемые интерфейсы	ADC, CAN, I2C, SPI, UART, PWM,	ADC, CAN, I2C, SPI, UART, PWM,	ADC, CAN, I2C, SPI, UART,



	USB	USB	PWM,USB, ETH, SSI,SSP
Диапазон температур, °С	От -40 до +80	От -40 до +85	От -40 до +85
Кол-во пинов, шт	64	48	144
Стоимость, р	1680	250	1370

Обоснование выбора СПУ

В качестве панели оператора для преобразователя будет выступать Kinco GH070E (рис.2).

Сенсорные панели оператора Kinco – это современные устройства, которые отвечают всем требованиям, предъявляемым к человеко-машинным интерфейсам в промышленных системах автоматизации.

Дисплеи имеют светодиодную подсветку и диагональ от 4,3 до 15 дюймов. В качестве операционной системы используется собственная система на базе Linux.

Для разработки интерфейсов используется среда Kinco Dtools, которая поддерживает векторную графику, растровые файлы bmp, jpg, gif и позволяет расширять функциональные возможности панели оператора с помощью C-макросов.



Рис.2. Панель оператора Kinco: а – вид спереди; б – вид сзади

Панель обладает следующими характеристиками:

Дисплей:

Диагональ: 7 дюймов;

Разрешение:800x480 пикселей.

Процессор и память:

Процессор: Высокопроизводительный 32-битный ARM RISC процессор 792 МГц;

Память: Встроенная flash память на 128 Мегабайт для пользовательских программ.

Коммуникационные интерфейсы:

RS232/RS485: Для подключения к контроллерам и другим устройствам;

USB: Для обновления программного обеспечения и передачи данных.

Программное обеспечение:

Панель программируется с помощью фирменного ПО Kinco DTools;

Имеется Поддержка стандартных протоколов, таких как Modbus RTU, Modbus TCP и других.

Разработка печатной платы преобразователя

Питание платы осуществляется от верхнего разъёма X1, контакты 1–2, после чего оно поступает на стабилизатор напряжения AMS1117–3.3v и преобразуется в 3.3 вольта. Далее питание поступает на микропроцессор STM32F103C8T6, микросхему Modbus ADM3485EARZ [6] и микросхему CAN SN65HVD230 [7].

Контакты 3–4 разъёма X1 служат для установки связи по протоколу Modbus между платой и панелью оператора. С помощью штырьевых разъёма XP1 происходит программирование микропроцессора. Благодаря наличию светодиодной индикации на плате можно с лёгкостью определить, какой из интерфейсов вышел из строя, что существенно облегчает процесс отладки и тестирования устройства.



После запайки всех компонентов получилась следующая печатная плата – рис.3.

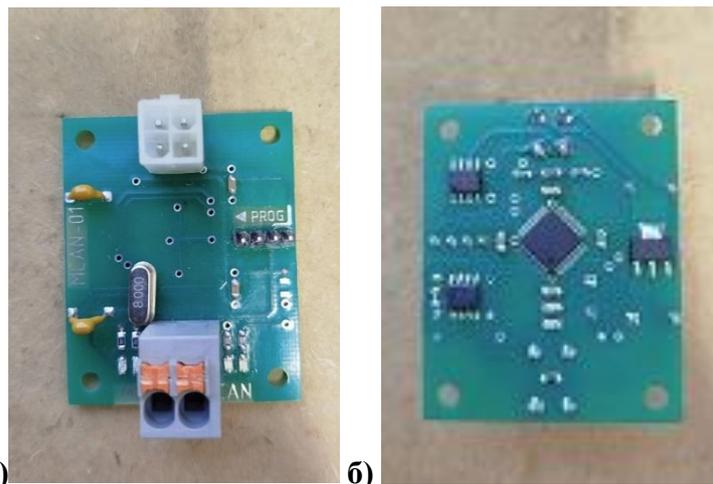


Рис.3. Плата Modbus-CAN: а – вид спереди; б – вид сзади

Настройка и разработка графического интерфейса для панели оператора Kinco GH070E

Создание интерфейса для панели оператора Kinco GH070E [8] начинается с формирования нового проекта. Для этого в среде Kinco DTools нужно выбрать в меню пункт «Файл» и затем нажать на «Новый». После этого откроется окно для создания проекта, в котором нужно указать название проекта и модель используемого HMI. В поле HMI можно сопоставить характеристики модели, чтобы удостовериться в корректности выбора устройства.

После того как устройство было определено, необходимо настроить сетевой интерфейс (рис.4), посредством которого будет осуществляться передача данных. Для этого следует перейти в меню свойств, где будет использоваться COM1 со следующими параметрами:

- Тип: RS485
- Скорость: 115200
- Бит данных: 8
- Проверка четности: Нет
- Стоповый бит: 2.

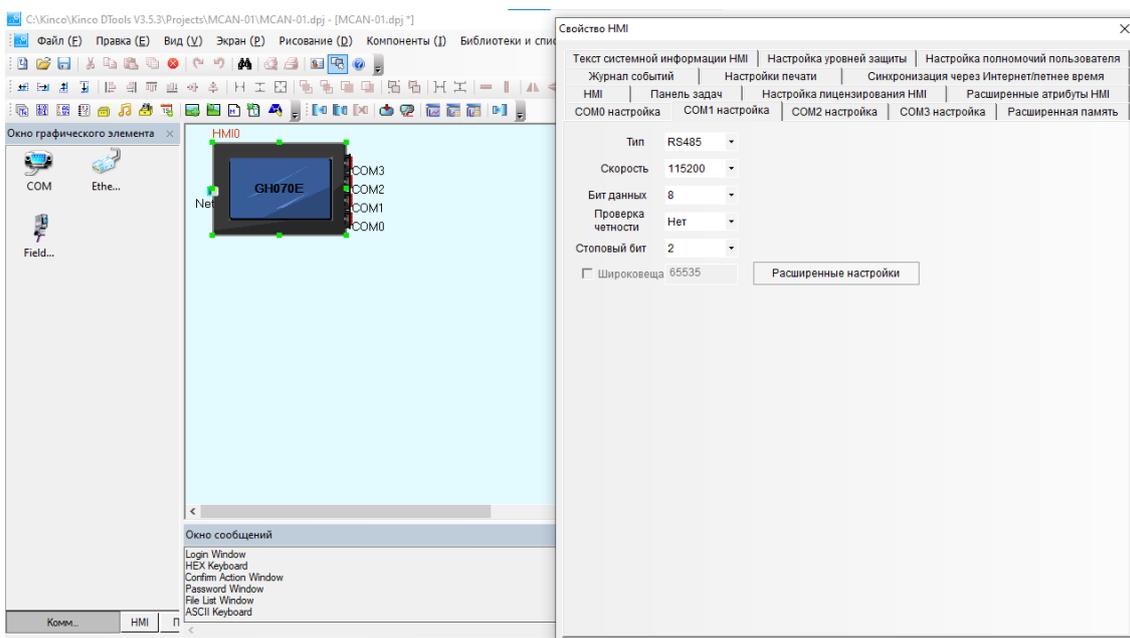


Рис.4. Настройка интерфейса передачи данных СПУ



По завершении настройки интерфейса следует определить устройство, к которому будет подключаться панель управления. Поскольку дальнейшая передача данных будет осуществляться с использованием протокола Modbus, необходимо в графическом элементе окна нажать кнопку «ПЛК» и выбрать Modbus RTU Extended (рис.5).

Также в меню свойств, в параметре «номер станции», следует указать адрес устройства, с которым устанавливается связь.

По завершении процедуры настройки подключения к внешнему устройству можно приступить к созданию главного экрана. Для этого в окне структуры проекта необходимо выбрать пункт «HMI, HMIO, Экраны, Frame0», после чего откроется рабочее поле для разработки графического интерфейса.

Рабочее поле автоматически адаптируется под размер реальной панели, что значительно упрощает процесс проектирования проекта. Для отрисовки текста в рабочем поле необходимо нажать на латинскую букву A на панели инструментов, после чего появится меню: «Атрибут компонента Текст», где мы можем установить необходимые нам настройки такие как изменение размеров шрифтов, стилей и многое другое.

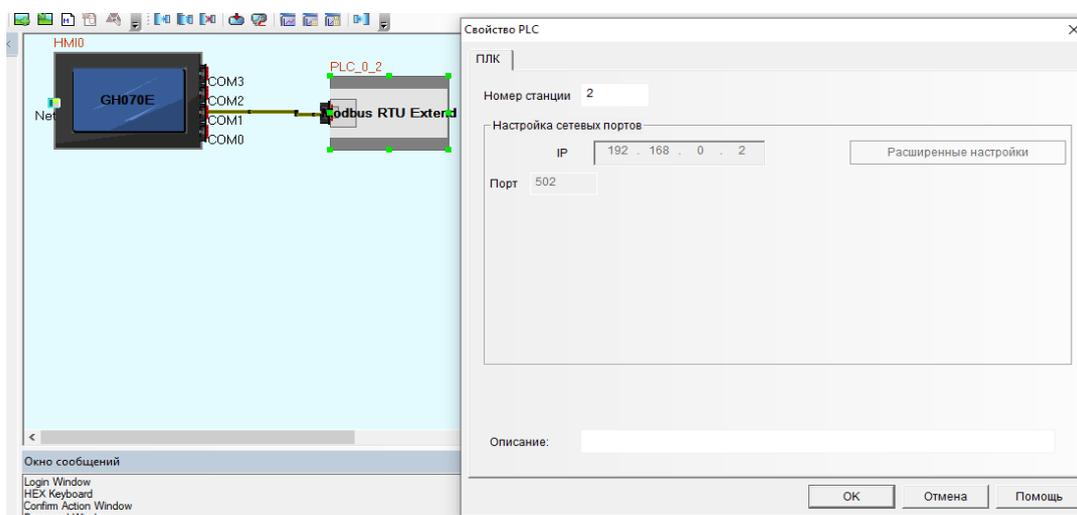


Рис. 5. Настройка связи с внешним устройством

Для того чтобы продолжить работу с протоколом Modbus, нам потребуются специальные ячейки, которые называются регистрами (рис.6). Чтобы создать такой регистр, необходимо перейти в меню компонента «Отображение данных и обработка числа». После этого появится окно «Атрибут компонента «Число», где можно задать более подробные настройки.

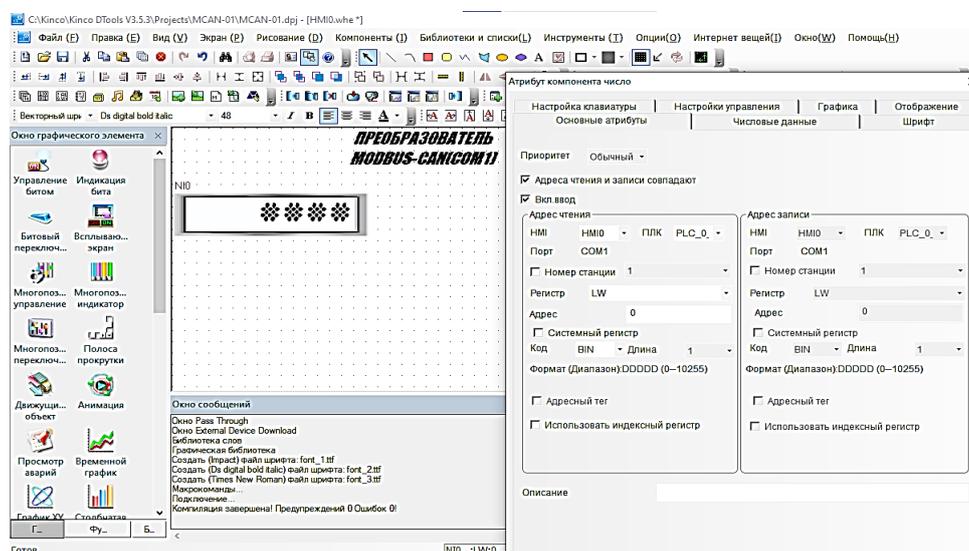


Рис. 6. Создание регистров



Сформированные регистры были условно классифицированы на две категории: «Регистры СПУ» и «Регистры платы преобразователя». В рамках дальнейшей разработки программного обеспечения планируется использовать «Регистры платы» для записи данных, полученных с платы преобразователя. В то же время будут задействованы и «Регистры СПУ», данные регистры будут использоваться для ввода значений оператором с последующей записью в плату преобразователя.

В программе также предусмотрен симулятор проекта (рис.7) который позволяет наглядно представить, как будет выглядеть интерфейс на реальной панели. Для этого необходимо перейти в меню «Инструменты» и скомпилировать созданный проект, после чего нажать кнопку «Симуляция».

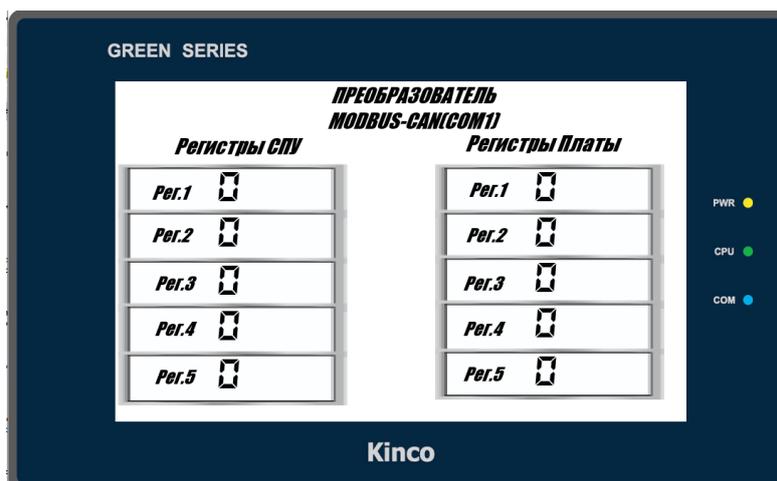


Рис. 7. Режим симуляции СПУ

Заключение

В процессе исследования была создана плата Modbus-CAN преобразователя, определены её компоненты, а также разработан графический интерфейс для панели оператора Kinco GH070E с целью последующего тестирования печатной платы.

Список литературы:

1. Денисенко В.В., Протоколы и сети Modbus и Modbus TCP // Современные технологии автоматизации. -2010., №4. -С.90-94.
2. Евгений Карпенко, Возможности CAN-протокола // Современная электроника и технологии автоматизации. -1998., №4. -С.16-20
3. Документация на микроконтроллер PIC32MX534F064H [Электронный ресурс]. – URL: <https://static.chipdip.ru/lib/336/DOC004336487.pdf>
4. Документация на микроконтроллер STM32F103C8T6 [Электронный ресурс]. – URL: <https://www.st.com/resource/en/datasheet/stm32f103cb.pdf>
5. Документация на микроконтроллер LPC4078FBD144 [Электронный ресурс]. – URL: <https://static.chipdip.ru/lib/855/DOC011855911.pdf>
6. Документация на микросхему Modbus ADM3485EARZ [Электронный ресурс]. – URL: Режим доступа: <https://static.chipdip.ru/lib/843/DOC018843803.pdf>
7. Документация на микросхему CAN SN65HVD230 [Электронный ресурс]. – URL: Режим доступа: <https://static.chipdip.ru/lib/775/DOC059775209.pdf>
8. Kinco Operator PanelGH070E [Электронный ресурс]. – URL https://cdownload.kinco.cn/ENDownload/D_enCatalog/HMI/Flyer_GH070&GH070E_EN%20202306.pdf

