



Федеральное государственное автономное образовательное учреждение высшего образования «Национальный Исследовательский Университет ИТМО»

ЛАБОРАТОРНАЯ РАБОТА №3
ПРЕДМЕТ «ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ
КОНТРОЛЛЕРОВ»
ТЕМА «ПЕРЕДАЧА ДАННЫХ С ПЛК ПО ПРОТОКОЛУ
MODBUS TCP И УПРАВЛЕНИЕ ЧАТОТНЫМ
ПРЕОБРАЗОВАТЕЛЕМ»

Преподаватель:
Крылова А. А.

Выполнили:
Румянцев А. А.
Дьячихин Д. Н.
Чебаненко Д. А.

Факультет: СУиР
Потоки ПРОГ. ПРОМ.ЛК:
2.2, 2.1, 1.1

Санкт-Петербург
2024

Содержание

1 Введение.	2
1.1 Цель работы.	2
1.2 Задания по работе.	2
2 Установка.	2
2.1 Схема установки.	2
3 Теоретическая часть.	3
4 Экспериментальная часть.	3
4.1 Передача данных с ПЛК по протоколу Modbus TCP.	3
4.2 Управление частотным преобразователем.	5

1 Введение.

1.1 Цель работы.

Разработать программу для ПЛК, которая передает в сеть данные с датчиков температуры и влажности по протоколу TCP. Разработать программу для управления скоростью и направлением вращения асинхронного двигателя с помощью частотного преобразователя.

1.2 Задания по работе.

1. Изучить теоретическую часть работы.
2. Реализовать модуль передачи данных в сеть на языке ST.
3. Реализовать модуль получения, обработки и отображения данных на языке Python.
4. Изучить документацию на частотный преобразователь.
5. Реализовать модуль для управления частотным преобразователем по протоколу Modbus TCP.

2 Установка.

В данной лабораторной работе мы работали со стендом на основе ПЛК Schneider Electric Modicon M251.

2.1 Схема установки.

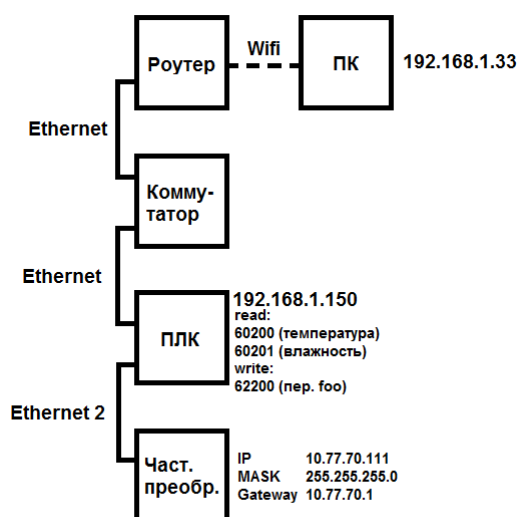


Рис. 1: Схема протоколов стенда.

Так как в данной лабораторной работе мы работаем с тем же стендом, что и в предыдущей работе, то общая схема остается неизменной. Связь ПК и ПЛК происходит через

коммутатор и Wi-Fi роутер. Во время выполнения работы ПК имел IP адрес 192.168.1.33 (было обнаружено с помощью команды `ifconfig`), а ПЛК – 192.168.1.150. 60200 и 60201 – номера регистров датчика температуры и влажности соответственно, из которых можно прочитать данные с датчиков (для начала нужно подключиться). 62200 – аналогично для переменной *foo* (позже про нее будет написано подробнее), однако с той разницей, что это номер регистра для записи данных.

3 Теоретическая часть.

Modbus TCP – открытый протокол, не зависящий от производителя используемого аппаратного обеспечения. Предназначен для контроля и управления оборудованием в системах несложной промышленной автоматизации. В частности, позволяет реализовать обмен сообщениями Modbus в локальных и глобальных вычислительных сетях с использованием стека протоколов TCP/IP, который часто используется для подключения к общей промышленной сети ПЛК, модулей ввода-вывода и «шлюзов».

Частотный преобразователь – это электронное или электромеханическое устройство, которое преобразует переменный ток (АС) одной частоты в переменный ток другой частоты. Устройство также может изменять напряжение, но это является второстепенным по отношению к его основной цели, поскольку преобразование напряжения переменного тока гораздо проще реализовать, нежели преобразование частоты.

4 Экспериментальная часть.

За основу возьмем проект лабораторной работы №2.

4.1 Передача данных с ПЛК по протоколу Modbus TCP.

В дереве проекта на вкладке *Device tree* в узле *Ethernet_1* создадим объект *ModbusTCP_SLAVE_1*.

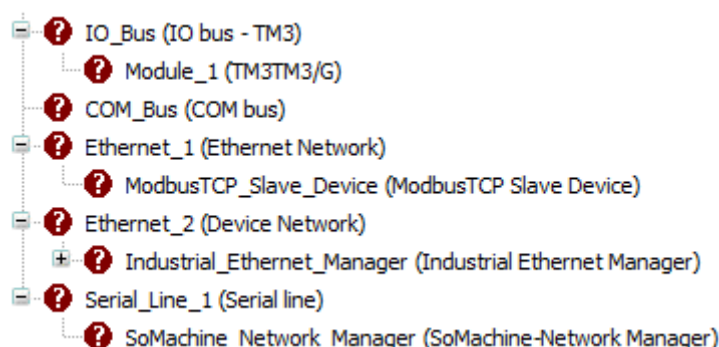


Рис. 2: Созданный объект ModbusTCP_SLAVE_Device в дереве проекта.

Создадим новый список глобальных переменных в узле *GVL* на вкладке *Application tree* и добавим туда переменную *foo* целого типа (остальные переменные будут рассмотрены позже).

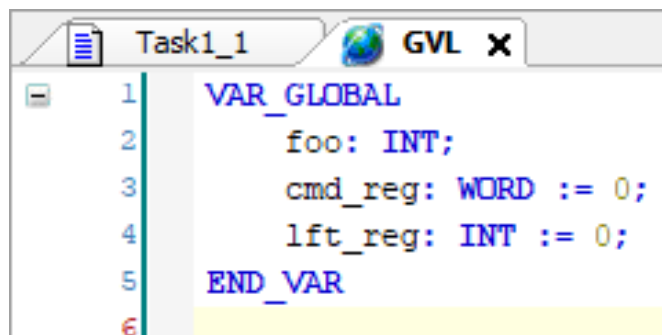


Рис. 3: Список глобальных переменных.

На вкладке *Tools Tree* создадим объект *Relocation Table*. В таблице *Read* создадим две новые пустые переменные *Modbus*. Нажмем на ячейку *Variable* и в открывшемся диалоге найдем переменные для температуры и влажности из предыдущей работы. Создадим пустую переменную в таблице *Write* и ассоциируем ее с глобальной переменной *foo*, которую мы задали ранее.

Далее напишем Python-скрипт, в котором реализуем код, необходимый для работы с ПЛК (в данном случае передача данных на ПЛК).

```

1  import time
2  from pyModbusTCP import utils
3  from pyModbusTCP.client import ModbusClient
4
5  def read_int16(addr):
6      regs = c.read_holding_registers(reg_addr=addr, reg_nb=1)
7      hex_uint = utils.get_list_2comp(regs, 16)
8      return hex_uint[0]
9
10 c = ModbusClient(host='192.168.1.150', port=502, auto_open=True)
11
12 while True:
13     if c.is_open:
14         temp = read_int16(60200) / 10
15         hum = read_int16(60201)
16         val = int(input('type int:'))
17         val = min(24000, max(0, val))
18         if c.write_single_register(62200, val):
19             print("ok")
20             print(temp, hum)
21     else:
22         c.open()
23     time.sleep(1)

```

Листинг 1: Python-скрипт для взаимодействия с ПЛК.

Данный скрипт устанавливает соединение с ПЛК по протоколу ModbusTCP, после чего начинает бесконечный циклический опрос регистров, содержащих значения температуры и влажности, а также запись введенных пользователем в консоль значений (*val*) в регистр, ассоциированный с глобальной переменной проекта. При этом значение переменной не может превысить 24000 или быть ниже 0.

После мы загрузили программу в ПЛК и запустили ее, тем самым убедившись в том, что все скомпилировалось без ошибок, значение глобальной переменной в узле *GVL* изменяется всегда, когда пользователь вводит некоторое значение в консоль, данные температуры и влажности отображаются в консоли.

Мы приравнивали переменную *foo* к переменной *motor*, чтобы при передаче данных в

первую из них вторая подбирала это значение и передавала на мотор. Программа корректно выполнялась – мотор вращался с той скоростью, которую пользователь ввел в консоль на своем ПК. Подаваемые значения не превышали заданный диапазон значений и не были ниже его.

4.2 Управление частотным преобразователем.