



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

## О Т Ч Е Т

### ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

#### «Эксплуатационная практика»

Студент гр. ИУК4-52Б \_\_\_\_\_ ( Губин Е.В. )  
(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_ ( Амеличева К.А. )  
(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка практики \_\_\_\_\_ баллов \_\_\_\_\_  
(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_ ( Гагарин Ю.Е. )  
(подпись) (Ф.И.О.)

\_\_\_\_\_ ( Пчелинцева Н.И. )  
(подпись) (Ф.И.О.)

\_\_\_\_\_ ( Амеличева К.А. )  
(подпись) (Ф.И.О.)

Калуга, 2024

УТВЕРЖДАЮ  
Заведующий кафедрой ИУК4  
\_\_\_\_\_(Гагарин Ю.Е.)  
«01» июля 2024 г.

**З А Д А Н И Е**  
**на ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ, Эксплуатационную**

За время прохождения практики студенту необходимо:

1. Определить предметную область, цель, задачи и основные результаты прохождения практики.
2. Выбирать необходимый математический аппарат для реализации задания, выбирать интегрированную среду разработки программного обеспечения, ознакомиться с технологиями создания графического интерфейса в реализуемом программном обеспечении; подобрать стандартные библиотеки; спроектировать компоненты программного продукта.
3. Разработать тестовое окружение, создать тестовые сценарии.
4. Освоить теорию по автоматизации процесса производства и запрограммировать контроллер: по нажатию на кнопку включить лампочку и подать сформированный с помощью incremental encoder аналоговый сигнал на slave-устройство, которое подаёт напряжение на электродвигатель и запускает его.
5. Подготовить отчет и защитить результаты практики.

Дата выдачи задания «01» июля 2024 г.

Руководитель практики \_\_\_\_\_ Амеличева К.А.

Задание получил \_\_\_\_\_ Губин Е.В.

## Оглавление

ВВЕДЕНИЕ .....	4
1. ОБЩИЕ СВЕДЕНИЯ .....	5
1.1. Исследование предметной области задачи и постановка задачи .....	5
1.2 Обоснование выбора средства реализации .....	5
1.3 Актуальность решаемой проблемы и возможные области применения данной разработки .....	6
1.4 Определение входных и выходных характеристик .....	7
1.5 Описание используемого математического аппарата .....	8
2. ПРОЕКТИРОВАНИЕ ПРОДУКТОВ ПРОГРАММНОГО ПРОДУКТА.....	9
2.1 Общие сведения о схеме .....	9
2.2 Программная часть.....	13
2.3 НМІ .....	16
ЗАКЛЮЧЕНИЕ .....	18
Список использованных источников.....	19
ПРИЛОЖЕНИЯ.....	21
Приложение 1 .....	21
Приложение 2 .....	23

## ВВЕДЕНИЕ

На производстве любого предприятия необходим постоянный контроль состояний станков, прокатных лент и прочих приборов. Так же помимо этого необходимо управлять этими приборами и производить сбор информации в базу данных.

Именно для этого и необходимы контроллеры – для автоматизации процесса производства. Человек посредством контроллеров управляет приборами, контроллеры используют специально прописанные сценарии для каких-либо событий и состояний прибора. Помимо всего этого контроллеры помогают выявить причину сбоя и быстро устранить этот сбой при процессе производства.

Общение контроллера и человека происходит через Human Machine Interface, а управление производством достигается с помощью использования схемы master-slave. Таким образом, человек может дистанционно общаться с приборами.

Целью учебной практики являются: получение опыта работы на предприятии, взаимодействие с коллективом, получение основ по программированию контроллеров.

Для достижения поставленной цели решаются следующие задачи:

- изучение теоретического материала
- выбор модели контроллера
- разработка схемы
- выбор языка программирования и программирование контроллера
- подключение НМІ
- запуск электродвигателя

## **1. ОБЩИЕ СВЕДЕНИЯ**

### **1.1. Исследование предметной области задачи и постановка задачи**

В качестве предметной области была выбрана область автоматизации процесса производства. Данная область предполагает программирование различных контроллеров, использование различного рода датчиков, проектирование базы данных. Применение данных технологий помогает человеку полностью управлять производством.

Программированием контроллеров на предприятии занимается специальный отдел – АСУ ТП (автоматизированная система управления технологическим производством). Данный отдел есть практически на каждом предприятии. Его задача – автоматизировать производство, решать различные проблемы на предприятии.

В качестве контроллеров выбираются контроллеры фирмы SIEMENS, они предоставляют полный комплект автоматизации производства – от различных моделей контроллеров до дополнительных датчиков.

В соответствие с предметной областью была поставлена задача: запрограммировать контроллер, который должен запускать электродвигатель на предприятии, при этом его мощность регулировалась с помощью incremental encoder, а время его работы и мощность вращения выводились на Human Machine Interface. Для удобства управления электродвигателем реализовать схему master-slave, чтобы производить управление.

### **1.2 Обоснование выбора средства реализации**

Для программирования контроллеров используются графические или текстовые языки программирования. Графические являются менее гибкими в плане написания soft части, но они являются более простыми. Предпочтение

было отдано текстовому языку программирования в виду более точной настройки контроллера.

Для программирования контроллера были выбраны:

- контроллер SIEMENS S7-1200 (данный контроллер является самым простым в освоении)
- текстовый Pascal-подобный язык программирования SCL (Substation Configuration Language)
- для данного языка программирования была выбрана специальная IDE TIA Portal v18
- EasyBuilder Pro для создания интерфейса Human Machine Interface
- принцип схемы master-slave для того, чтобы можно было отслеживать состояние электродвигателя и управлять его мощностью в отделе АСУ ТП, а не на производстве

### **1.3 Актуальность решаемой проблемы и возможные области применения данной разработки**

Актуальность проблемы автоматизации производства остро стоит на каждом предприятии. Обосновывается это удобством управления производством, быстрым решением проблем и сбоям приборов на предприятии, автоматическим сбором информации (например, о размере выпущенной продукции на стане за определенное время) и выполнением сценариев (например, выполни цикл охлаждения прибора, если температура достигла 98 градусов Цельсия). Данный принцип производства является рациональным решением, так как человек сам не смог бы реализовать все вышеперечисленные аспекты без использования описанных технологий.

Областями применения безусловно является любое предприятие. Конкретно на предприятии АО «ОМЗ» автоматизация производства используется, например, для сбора данных в общую базу данных предприятия и

отслеживания состояния продукта в цикле изготовления металлического изделия (поступил на ленту, находится в обработке, закончен цикл изготовления – на каждом этапе происходит сбор информации и управление каждым этапом производства через Human Machine Interface).

#### 1.4 Определение входных и выходных характеристик

Контроллер предназначен для передачи аналогового сигнала на slave-устройство, которое, в свою очередь, запускает электродвигатель. Мощность электродвигателя определяется величиной поданного напряжения. Оно определяется значением от 0% до 100% с помощью incremental encoder (представляет из себя крутящийся маховик). Значение в процентном виде с incremental encoder поступает в контроллер.

Диапазон измерения $\pm 80$ мВ	Диапазон измерения $\pm 2,5$ В	Диапазон измерения $\pm 5$ В	Диапазон измерения $\pm 10$ В	Единицы		Диапазон
				Десятичные	Шестнадцатеричные	
> 94,071	> 2,9397	> 5,8794	> 11,7589	32767	7FFF <sub>H</sub>	Переполнение
94,071	2,9397	5,8794	11,7589	32511	7EFF <sub>H</sub>	Перегрузка
:	:	:	:	:	:	
80,003	2,500	5,0002	10,0004	27649	6C01 <sub>H</sub>	
80,000	2,5	5,00	10,00	27648	6C00 <sub>H</sub>	Номинальный диапазон
60,000	1,86	3,75	7,50	20736	5100 <sub>H</sub>	
:	:	:	:	:	:	
- 60,000	- 1,86	- 3,75	- 7,50	-20736	AF00 <sub>H</sub>	
- 80,000	- 2,50	- 5,00	- 10,00	-27648	9400 <sub>H</sub>	
- 80,003	- 2,5001	- 5,0002	- 10,0004	-27649	93FF <sub>H</sub>	Отрицательная перегрузка
:	:	:	:	:	:	
- 94,074	- 2,9397	- 5,8796	- 11,759	-32512	8100 <sub>H</sub>	
< - 94,074	< - 2,9397	< - 5,8796	< - 11,759	-32768	8000 <sub>H</sub>	Отрицательное переполнение

Таблица 1 Преобразование дискретного сигнала в аналоговый сигнал

Далее контроллер преобразует дискретный сигнал путем преобразования процентного числа в аналоговое значение (от 0 до 27648 в соответствии с

таблицей), которое подается на slave-устройство и впоследствии подается на электродвигатель.

Выходным результатом является вращение электродвигателя с мощностью, зависящей от поданного аналогового значения.

### **1.5 Описание используемого математического аппарата**

Контроллер работает на основе бесконечного программного цикла, который перевыполняет код программы и сохраняет данные.

Контроллер состоит из различных модулей, например блок питания, модуль ввода, вывода, которые имеют дискретные пины. Каждый пин имеет свой адрес и, соответственно, имя переменной в программе – байт и бит. Общение с другими модулями и устройствами происходит через эти пины.

Загрузка программного компонента и Human Machine Interface происходит через локальную сеть. Обратная связь контроллера происходит через световые датчики на самом контроллере. Здесь можно увидеть ошибку, неправильное соединение пинов, потеря связи с другим контроллером.



## 2. ПРОЕКТИРОВАНИЕ ПРОДУКТОВ ПРОГРАММНОГО ПРОДУКТА

### 2.1 Общие сведения о схеме

Физическая составляющая называется корзиной или hardware. Каждая часть схемы состоит из нескольких модулей. Каждый модуль имеет свой серийный номер, определяющий модель данного модуля. К каждой модели контроллера подходят далеко не все модели модулей. В случае подбора неправильного модуля контроллер выдаст ошибку (загорится соответствующий индикатор). Контроллеры фирмы Siemens поддерживают «горячую» замену моделей, что предполагает замену неисправного модуля без отключения всей системы.

На рисунке ниже изображена корзина master-устройства. Питание происходит напрямую, не через модуль. Корзина состоит из следующих модулей:

- цифровой вход – необходим для фиксации приходящего с кнопки сигнала
- цифровой выход – необходим для того, чтобы зажечь лампочку по нажатию кнопки
- модуль MASTERSLAVE – передает значение на slave-устройство

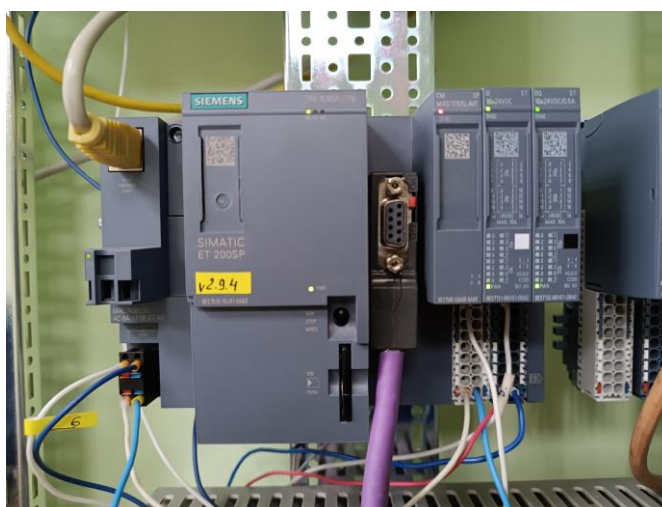


Рис. 1 Master-устройство

Далее изображена корзина slave-устройства. Она состоит из следующих модулей:

- модуль питания – питает slave-устройство
- цифровой вход
- цифровой выход
- аналоговый выход – подаёт не просто сигнал, а напряжение на электродвигатель
- счётчик count – необходим для отслеживания времени работы двигателя



Рис. 2 Slave-устройство

Incremental encoder представляет из себя крутящийся маховик, который привязан к пину цифрового входа. Когда маховик крутится, то на вход в модуль счетчика подаётся значение, насколько повернут маховик.



Рис. 3 Incremental encoder

На следующих двух рисунках изображены master- и slave- устройства, собранные со стороны soft-части:

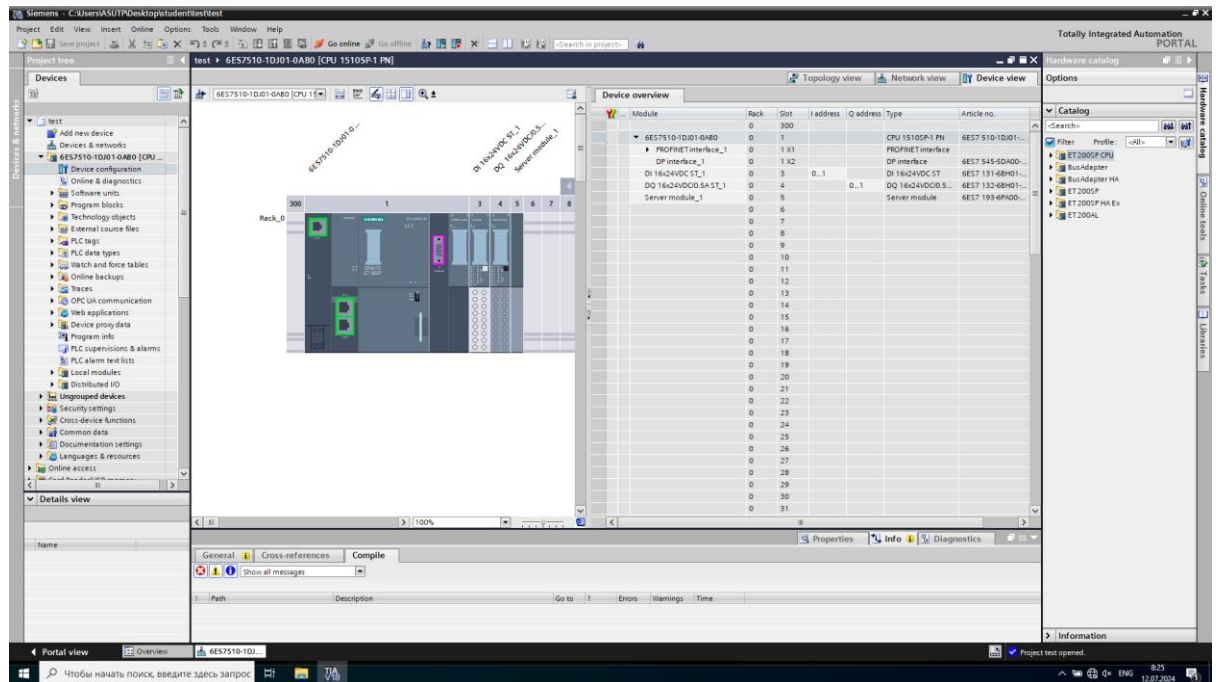


Рис. 4 Master-устройство – soft-сборка

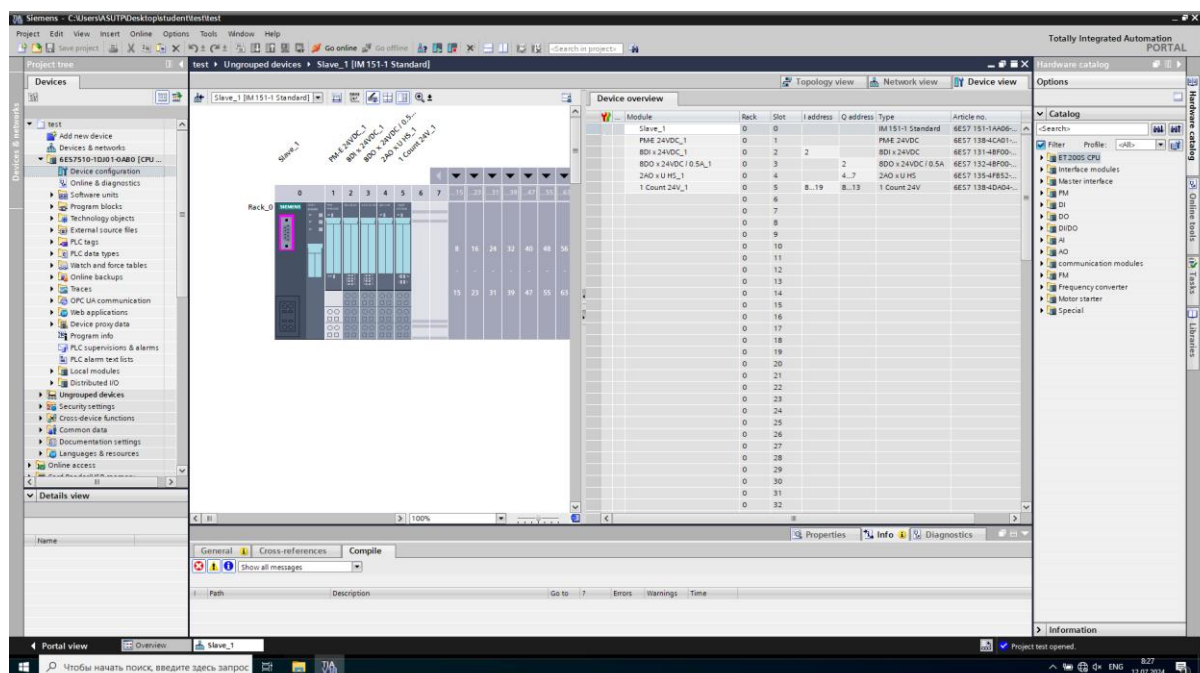


Рис.5 Slave-устройство – soft-сборка

## 2.2 Программная часть

Программа состоит из совокупности блоков различного типа и выполняется бесконечно циклично. Существуют следующие типы блоков:

- организационный блок – с этого блока, как правило, начинается цикл выполнения программы
- функция – простой блок кода, представляет из себя обычную функцию
- блок информации – представляет из себя блок, в котором прописаны переменные, которые изменяются в ходе выполнения программы
- функциональный блок – является функцией, но главным отличием от неё является то, что к функциональному блоку привязан блок информации. То есть функция не способна сохранять промежуточные значения в ходе выполнения бесконечного цикла программы, а функциональный блок может

Таблица тегов представляет из себя описание глобальных переменных. Указывается название переменных, их тип и адрес, который является номером байта и бита. Если первая стоит I – это вход, Q – выход, M – процессорная память контроллера. Запись I0.0 означает вход 0 байт и 0 бит, QW4 – выход 2 байт и 4 бит (W в данном случае означает слово – 2 байта).

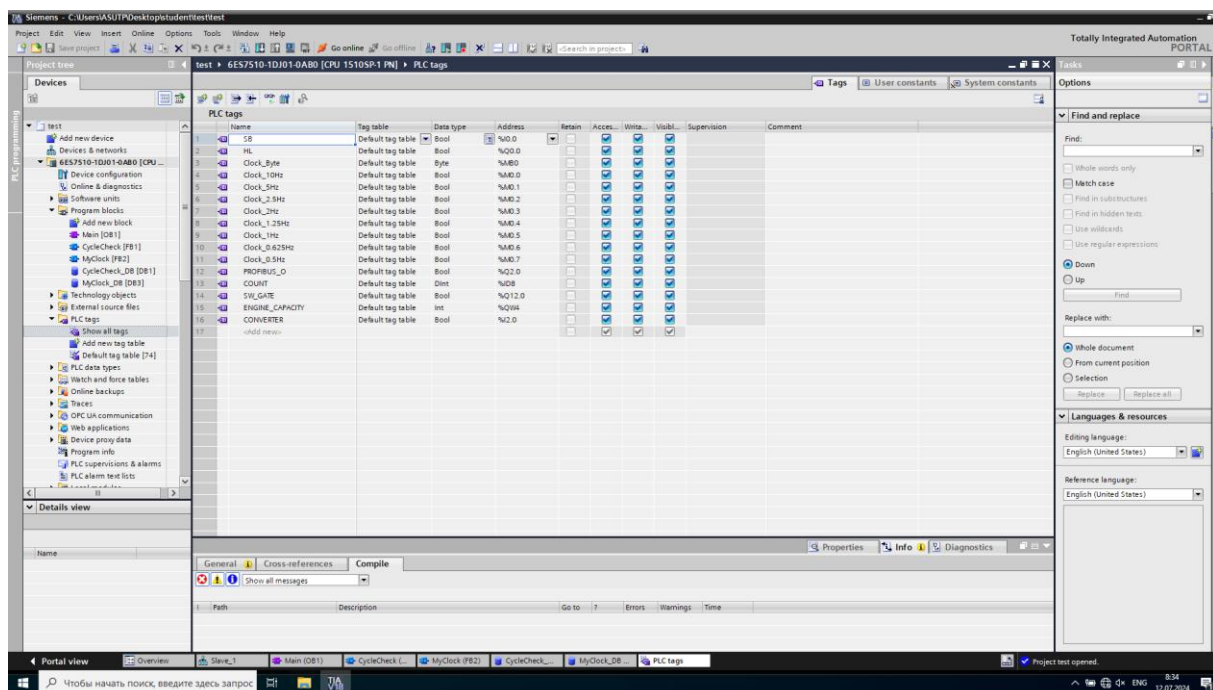


Рис. 6 Таблица тегов

В таблице тегов:

- SB – сигнал с кнопки
- HL – сигнал на лампочку
- Clock\_0.5Hz – счетчик с частотой 0.5 Гц, необходим для счета времени: означает, что за один период проходит 2 секунды, значит за полупериод – 1 секунда
- SW\_GATE – сигнал-заслонка, когда заслонка открыта (true), то значение с incremental encoder можно считать (без этого значение попросту не будет меняться)
- ENGINE\_CAPACITY – аналоговый выход (напряжение) на электродвигатель, высчитывается в процессе выполнения программы
- CONVERTER – сигнал с ещё одной кнопки для проверки полной работоспособности
- COUNT – текущее значение угла на incremental encoder

Следующий код отвечает за включение/выключение лампочки:

```
IF ("CONVERTER" OR "SB") AND NOT #temp THEN
    #temp := TRUE;
    "HL" := NOT "HL";
    "PROFIBUS_O" := NOT "PROFIBUS_O";
    "SW_GATE" := NOT "SW_GATE";
    #ie_value := 0;
END_IF;

IF NOT "SB" AND NOT "CONVERTER" THEN
    #temp := FALSE;
END_IF;
```

Так как нам нужно значение с incremental encoder от 0% до 100%, то создадим свой счетчик. Программа выполняется циклически, значит для того чтобы нам увеличивать или уменьшать значение, необходимо запоминать значение COUNT с прошлого цикла и сравнивать со значением в текущей итерации, при этом в условиях необходимо наложить условия от 0 до 100.

Описанный код:

```
IF "SW_GATE" THEN
    IF #flag THEN
        IF #last < "COUNT" AND #ie_value > -1 AND #ie_value < 100 THEN
            #ie_value := #ie_value + 1;
        END_IF;
        IF #last > "COUNT" AND #ie_value > 0 AND #ie_value < 101 THEN
            #ie_value := #ie_value - 1;
        END_IF;
    ELSE
        IF "COUNT" > -1 AND "COUNT" < 101 THEN
            #ie_value := "COUNT";
        ELSE
            #flag := TRUE;
        END_IF;
    END_IF;
    #last := "COUNT";
END_IF;
```

Далее происходит преобразование процентного значения напряжения (от 0В до 5В) в значение аналогового выхода (в документации от 0 до 27648). Так же стоит провалидировать данные, чтобы они действительно были в необходимом диапазоне, в противном случае на электродвигатель будет подаваться слишком высокое напряжение, что приведёт к порче электродвигателя. Код:

```
#temp_voltage := REAL_TO_INT((DINT_TO_REAL(#ie_value) / 100.0) * 27648.0);

IF #temp_voltage >= 0 AND #temp_voltage <= 27648 THEN
    "ENGINE_CAPACITY" := #temp_voltage;
    #voltage := DINT_TO_REAL(#ie_value) * 10.0;
ELSE
    "ENGINE_CAPACITY" := 0;
    #voltage := 0.0;
END_IF;
```

## 2.3 HMI

Создание интерфейса происходит через простейший конструктор. Для того, чтобы данные отображались в текущий момент времени, необходимо в этот интерфейс импортировать таблицы тегов и все информационные блоки. Подключить дисплей интерфейса к контроллеру можно по локальной сети или по проводам.



## Создание интерфейса:

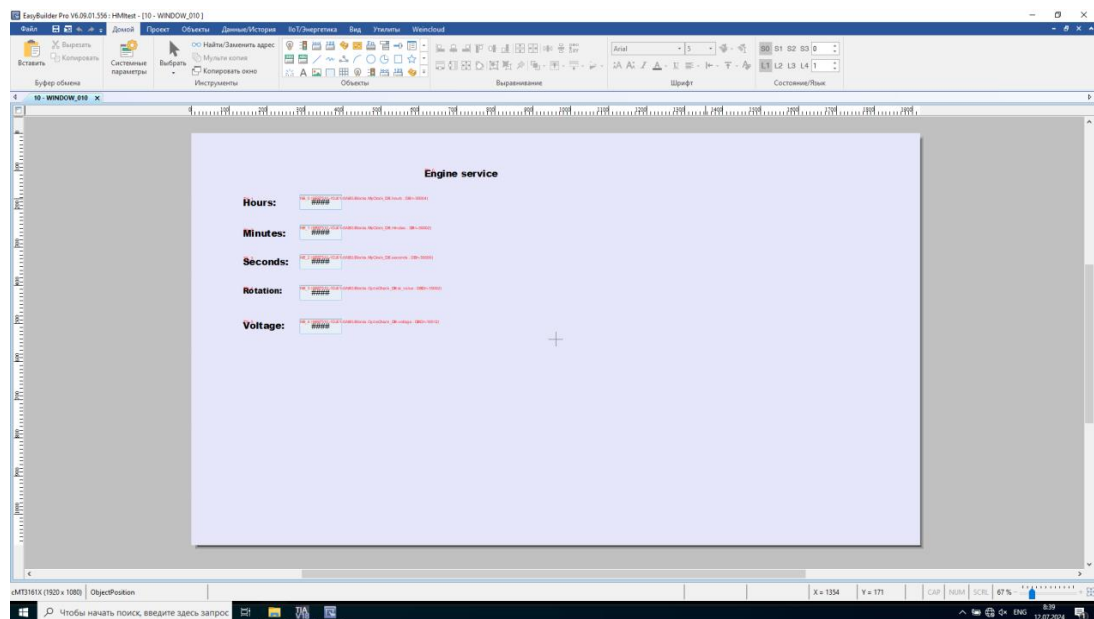


Рис. 7 Создание HMI

## **ЗАКЛЮЧЕНИЕ**

В заключении хочу подвести итоги по прохождению эксплуатационной практики на предприятии АО «ОМЗ». В результате прохождения практики был запрограммирован контроллер, который позволяет управлять скоростью вращения электродвигателя, получена база по автоматизации процесса производства. На протяжении двух недель изучался фундамент по работе в отделе АСУ ТП, изучался язык программирования, аппаратная и программная часть контроллера. В течение практики так же я развивал общие навыки по работе с англоязычной документацией. Вся проделанная работа помогла познакомиться с технологиями автоматизации и работой на металлургическом предприятии.

## Список использованных источников

1. Вайнштейн, М. З. Основы научных исследований [Электронный ресурс] : учебное пособие / М. З. Вайнштейн, В. М. Вайнштейн, О. В. Кононова. — Йошкар-Ола: Марийский государственный технический университет, Поволжский государственный технологический университет, ЭБС АСВ, 2011. — 216 с. — Режим доступа: <http://www.iprbookshop.ru/22586.html>
2. Иванов А.А., Автоматизация технологических процессов и производств [Текст]: учебное пособие / А.А. Иванов – Москва. : Форум, 2023. – 224 с.
3. Иванов В.Н., Программирование логических контроллеров [Текст]: учебное пособие / В.Н. Иванов – Москва. : Салон-Пресс, 2020. – 356 с.
4. Коваленко, Ю.В. Информационно-поисковые системы [Электронный ресурс]: учебно-методическое пособие / Ю.В. Коваленко, Т.А. Сергиенко. — Омск: Омская юридическая академия, 2017. — 38 с.— Режим доступа: <http://www.iprbookshop.ru/66817.html>
5. Кульчавеня М.П., Основы программирования ПЛК [Текст]: учебно-методическое пособие / М.П. Кульчавеня – Москва : Нобель Пресс, 2012. – 158 с.
6. Маюрникова, Л. А. Основы научных исследований в научно-технической сфере [Электронный ресурс]: учебно-методическое пособие / Л. А. Маюрникова, С. В. Новосёлов. — Кемерово: Кемеровский технологический институт пищевой промышленности, 2009. — 123 с. — Режим доступа: <http://www.iprbookshop.ru/14381.html>
7. Моделирование информационных ресурсов [Электронный ресурс]: учебно-методический/ Составитель Огнев Э.Н. - Кемерово: Кемеровский государственный университет культуры и искусств, 2013. - 36 с.: ил., табл. - URL: <http://biblioclub.ru/index.php?page=book&id=274218>
8. Моделирование систем[Текст]: учебник для вузов / С.И. Дворецкий, Ю.Л. Муромцев, В.А. Погонин, А.Г. Схиртладзе. – М.: Академия, 2009. – 320 с.
9. Мокий, М.С. Методология научных исследований[Текст]: учебник / М.С. Мокий, А.Л. Никифоров, В.С. Мокий. - М.: Юрайт, 2015. - 255 с.
10. Порсев, Е. Г. Организация и планирование экспериментов [Электронный ресурс]: учебное пособие / Е. Г. Порсев.— Новосибирск : Новосибирский государственный технический университет, 2010. — 155 с. — Режим доступа: <http://www.iprbookshop.ru/45415.html>
11. Рачков, М. Ю. Автоматизация производства : учебник для среднего профессионального образования / М. Ю. Рачков. — 2-е изд., испр. и доп. — Москва. : Юрайт, 2024. — 182 с.
12. Рогов, В.А. Методика и практика технических экспериментов[Текст]: учеб.пособие / В.А. Рогов, А.В. Антонов, Г.Г. Поздняк. – М.: Академия, 2005. – 288 с.

13. Шагин А.В., Основы автоматизации технологических процессов [Текст]: учебное пособие / А.В. Шагин, В.И. Демкин, В.Ю. Кононов, А.Б. Кабанова – Москва. : Юрайт, 2023. – 164 с.
14. Щербаков, А. Интернет-аналитика [Электронный ресурс]: поиск и оценка информации в web-ресурсах: практическое пособие / А. Щербаков. - М.: Книжный мир, 2012. - 78 с. - URL: <http://biblioclub.ru/index.php?page=book&id=89693>.

# ПРИЛОЖЕНИЯ

## Приложение 1

### Листинг кода

#### CycleCheck [FB1]

```
IF ("CONVERTER" OR "SB") AND NOT #temp THEN
    #temp := TRUE;
    "HL" := NOT "HL";
    "PROFIBUS_O" := NOT "PROFIBUS_O";
    "SW_GATE" := NOT "SW_GATE";
    #ie_value := 0;
END_IF;

IF NOT "SB" AND NOT "CONVERTER" THEN
    #temp := FALSE;
END_IF;

IF "SW_GATE" THEN
    IF #flag THEN
        IF #last < "COUNT" AND #ie_value > -1 AND #ie_value < 100 THEN
            #ie_value := #ie_value + 1;
        END_IF;
        IF #last > "COUNT" AND #ie_value > 0 AND #ie_value < 101 THEN
            #ie_value := #ie_value - 1;
        END_IF;
    ELSE
        IF "COUNT" > -1 AND "COUNT" < 101 THEN
            #ie_value := "COUNT";
        ELSE
            #flag := TRUE;
        END_IF;
    END_IF;
    #last := "COUNT";
END_IF;

#temp_voltage := REAL_TO_INT((DINT_TO_REAL(#ie_value) / 100.0) * 27648.0);

IF #temp_voltage >= 0 AND #temp_voltage <= 27648 THEN
    "ENGINE_CAPACITY" := #temp_voltage;
    #voltage := DINT_TO_REAL(#ie_value) * 10.0;
ELSE
    "ENGINE_CAPACITY" := 0;
    #voltage := 0.0;
END_IF;

"MyClock_DB"();
```

#### Main [OB1]

```
"CycleCheck_DB"();
```

#### MyClock [FB2]

```
IF "HL" THEN
    IF "Clock_0.5Hz" <> #temp THEN
        #temp := "Clock_0.5Hz";
        #seconds := #seconds + 1;
        IF #seconds = 60 THEN
```

```
        #seconds := 0;
        #minutes := #minutes + 1;
    END_IF;
    IF #minutes = 60 THEN
        #minutes := 0;
        #hours := #hours + 1;
    END_IF;
END_IF;
ELSE
    #seconds := 0;
    #minutes := 0;
    #hours := 0;
END_IF;
```

## Приложение 2

### Скриншоты программных инструментов

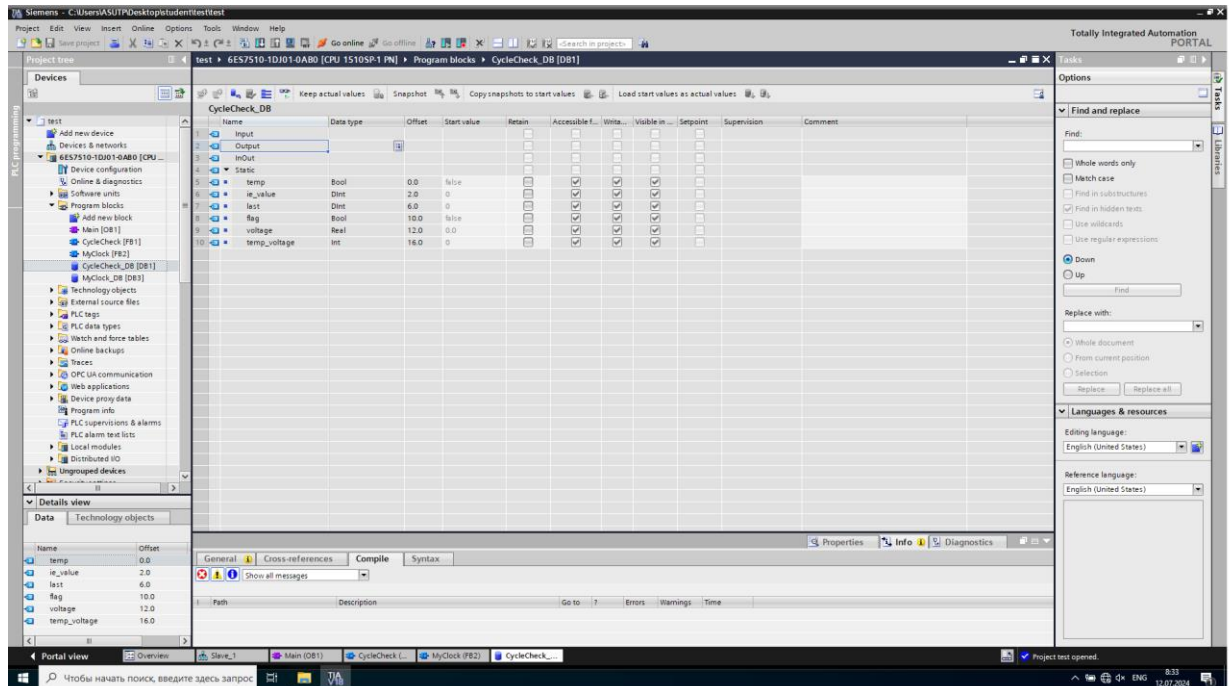


Рис. 1 Информационный блок, отвечающий за основную программу

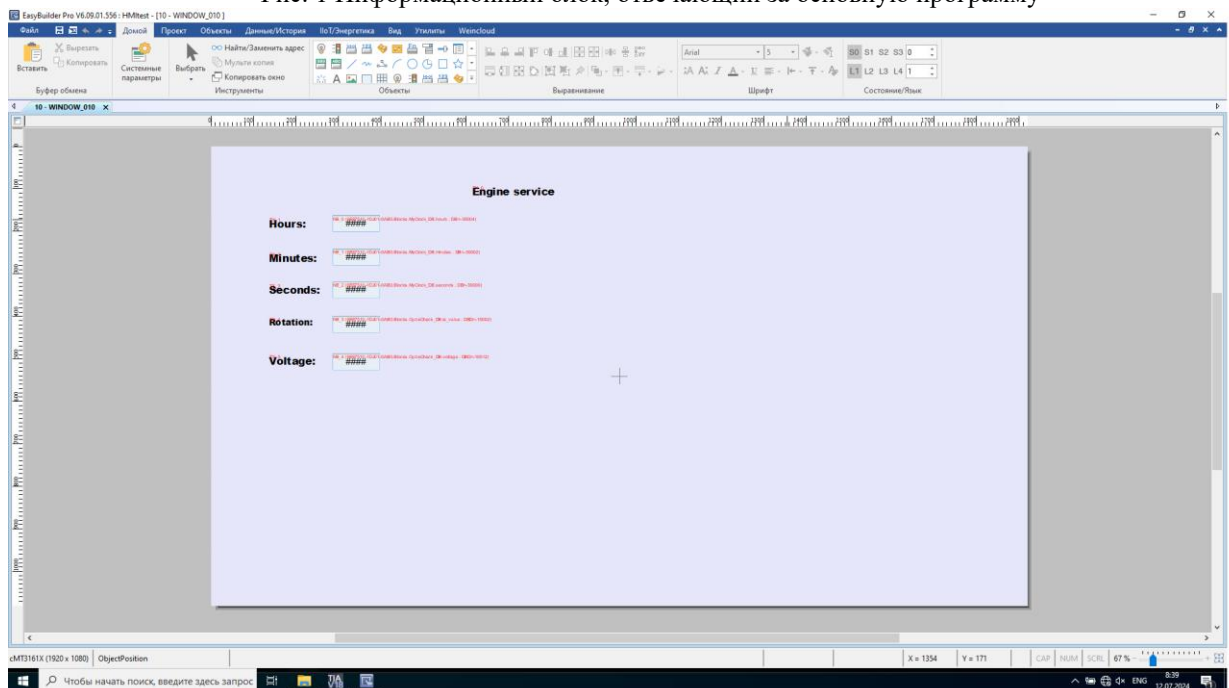


Рис. 2 EasyBuikder Pro для создания HMI

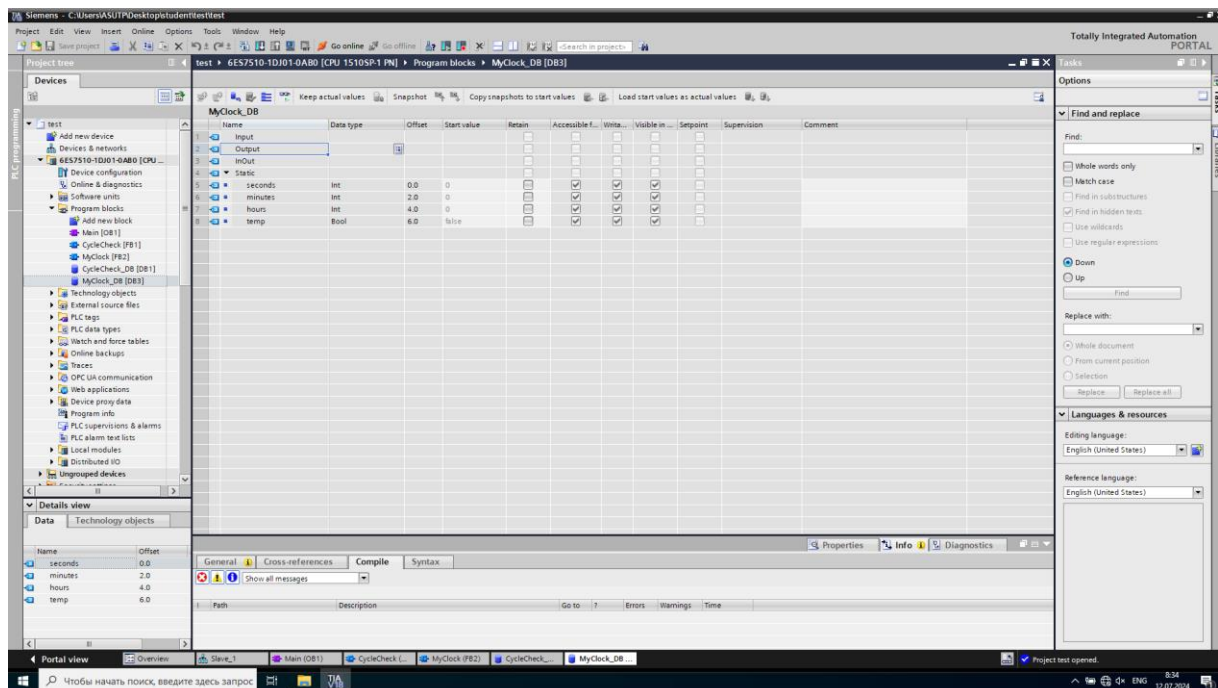


Рис. 3 Информационный блок, отвечающий за работающий таймер

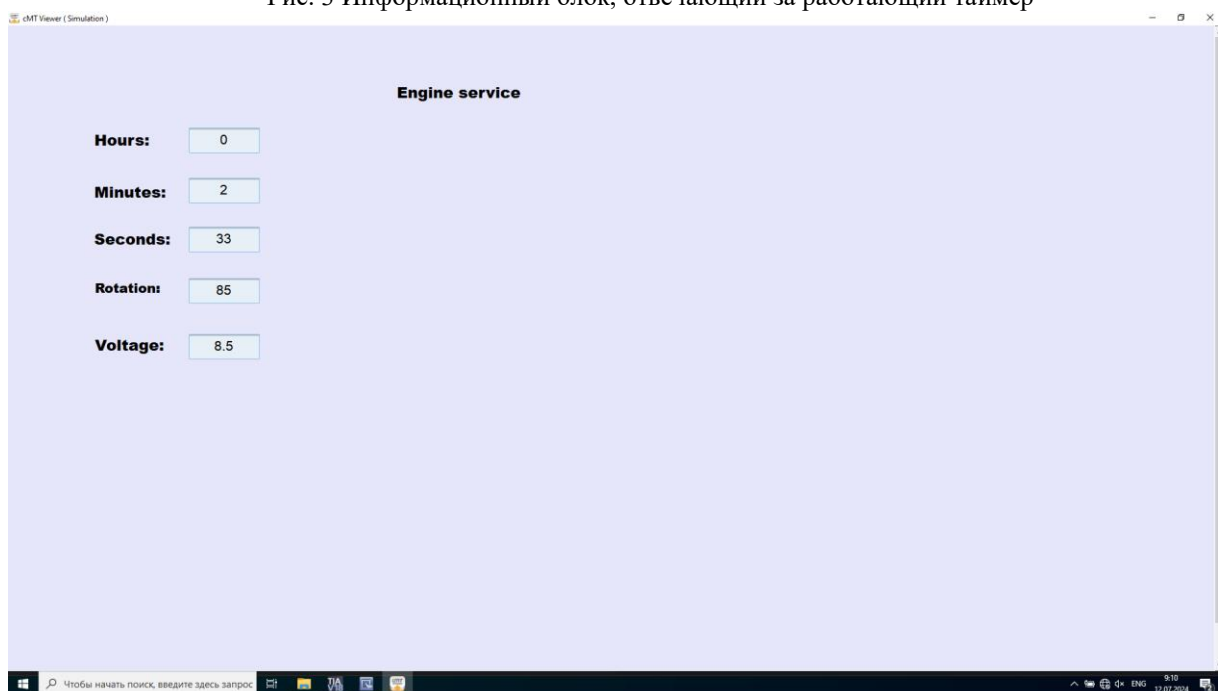


Рис. 4 Онлайн симуляция работы HMI



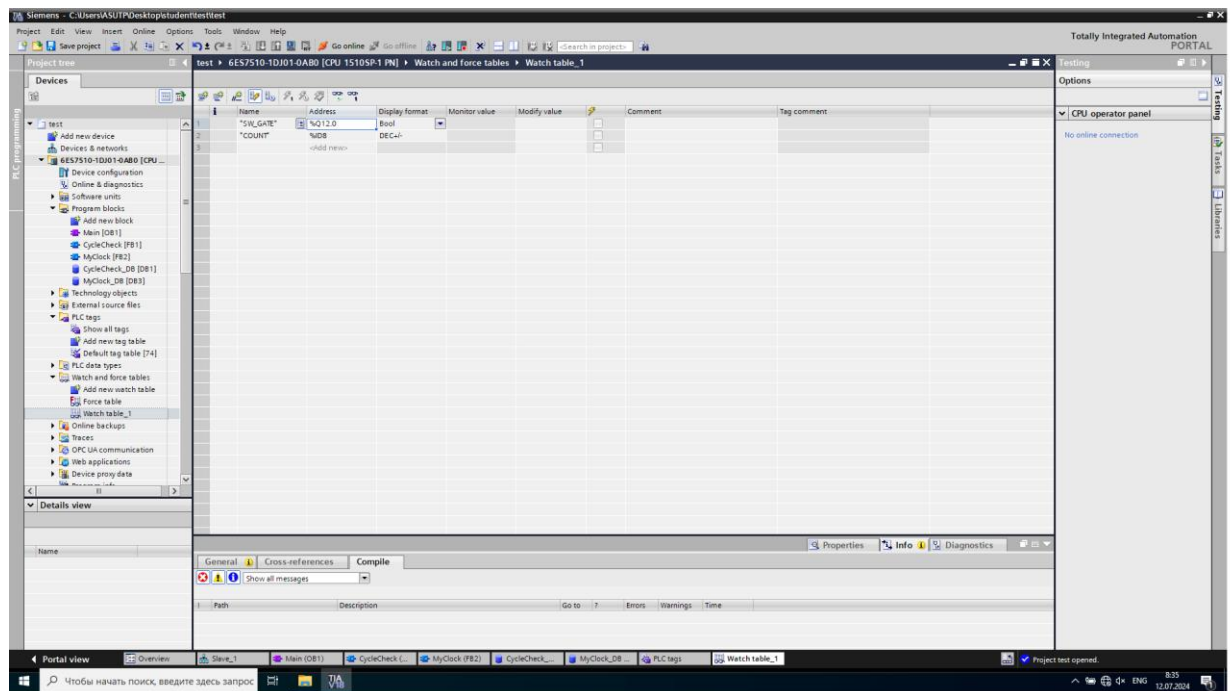


Рис. 5 Таблица, показывающая значение переменных в реальном времени