



Тяжелая промышленность и .NET

Николай Воронин

Ведущий разработчик Altendar

Николай Воронин

Ведущий разработчик Altenar

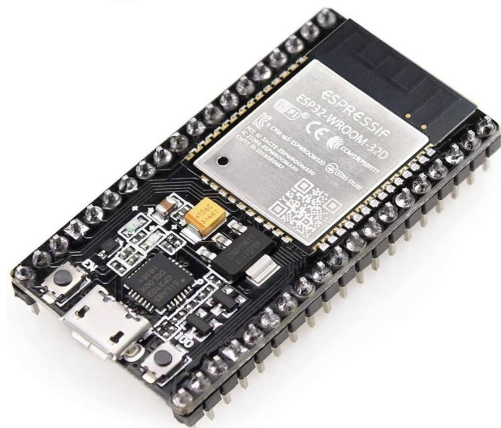
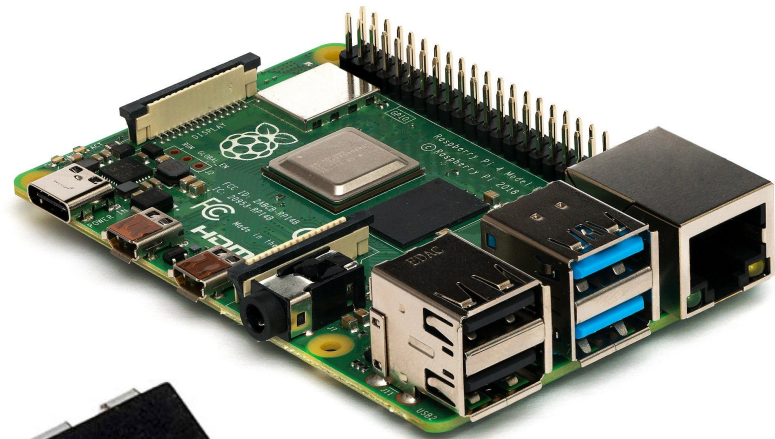
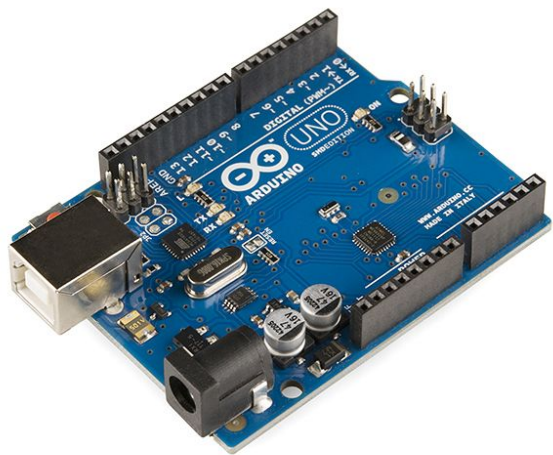


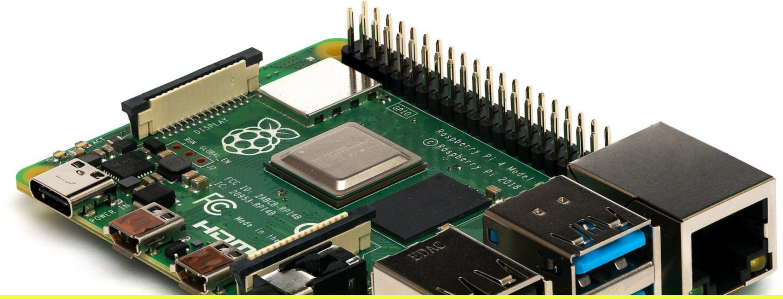
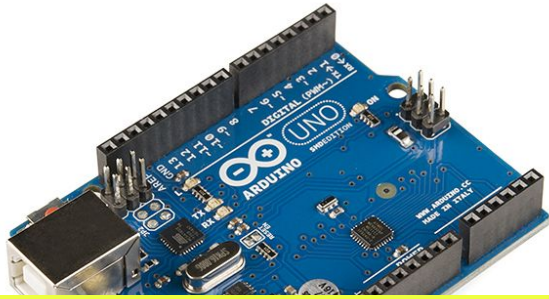
Ведущий инженер-программист
Beckhoff Automation

C/C++ Python C# .NET

IEC 61131-3





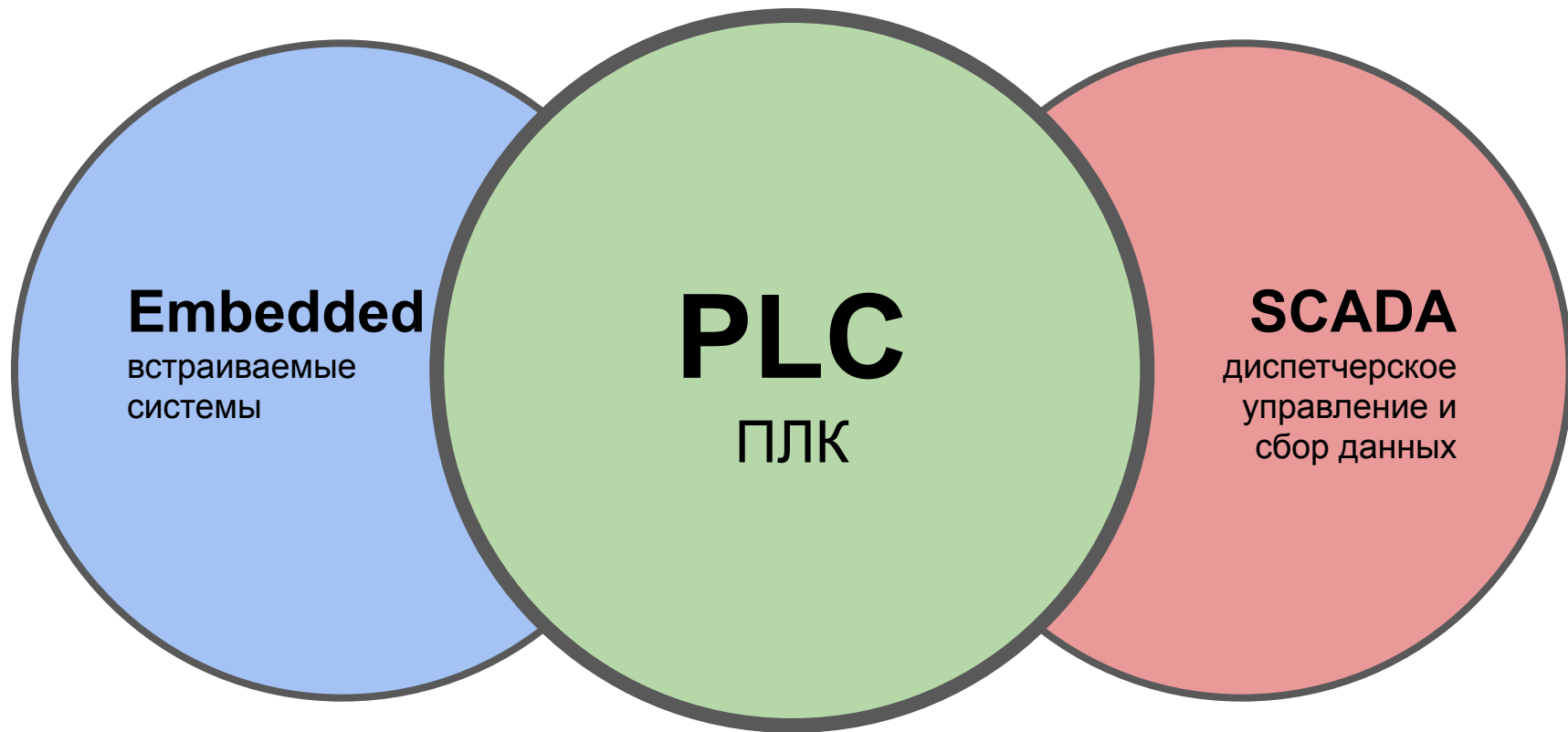


EMBEDDED



В КАКУЮ СТОРОНУ ЕДЕТ АВТОБУС?





Программируемый Логический Контроллер, PLC

Надежность

Безопасность

Техподдержка

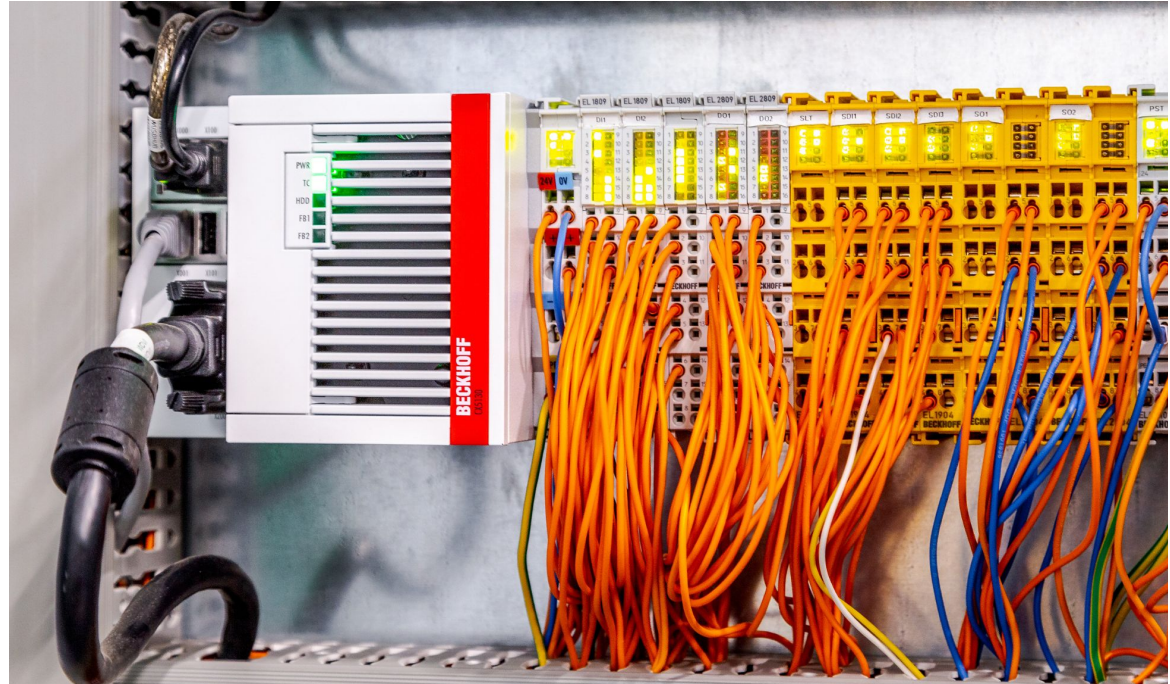
Обслуживание

Интеграция

Поддержка, LTS

Масштабируемость

...



Что внутри ПЛК?

- CPU: ARM, Intel / AMD, и все-все-все
- ОС: RTOS, WinCE, Windows 2k..10, DOS, Linux, FreeBSD, ...



- CoDeSys | Овен
- TwinCAT | Beckhoff
- Beremiz | Open source, GPL-2.0, LGPL-2.1

ЭЛ. ШКАФ

РОБОТ

ПРОДУКТ

КОНВЕЙЕР

Для всего есть

- Предсказуемое поведение
- Гарантированное время реакции
- Строгое планирования процессов
- Минимальное время реакции

C++



МЭК / IEC / EN 61131-3

Графические (диаграммы)

ЭЛ. ШКАФ

- LD – релейные, лестничные схемы
- FBD / CFC – блоки функций (как Лего)
- SFC – диаграммы последовательностей функций

ПРОДУКТ

КОНВЕЙЕР

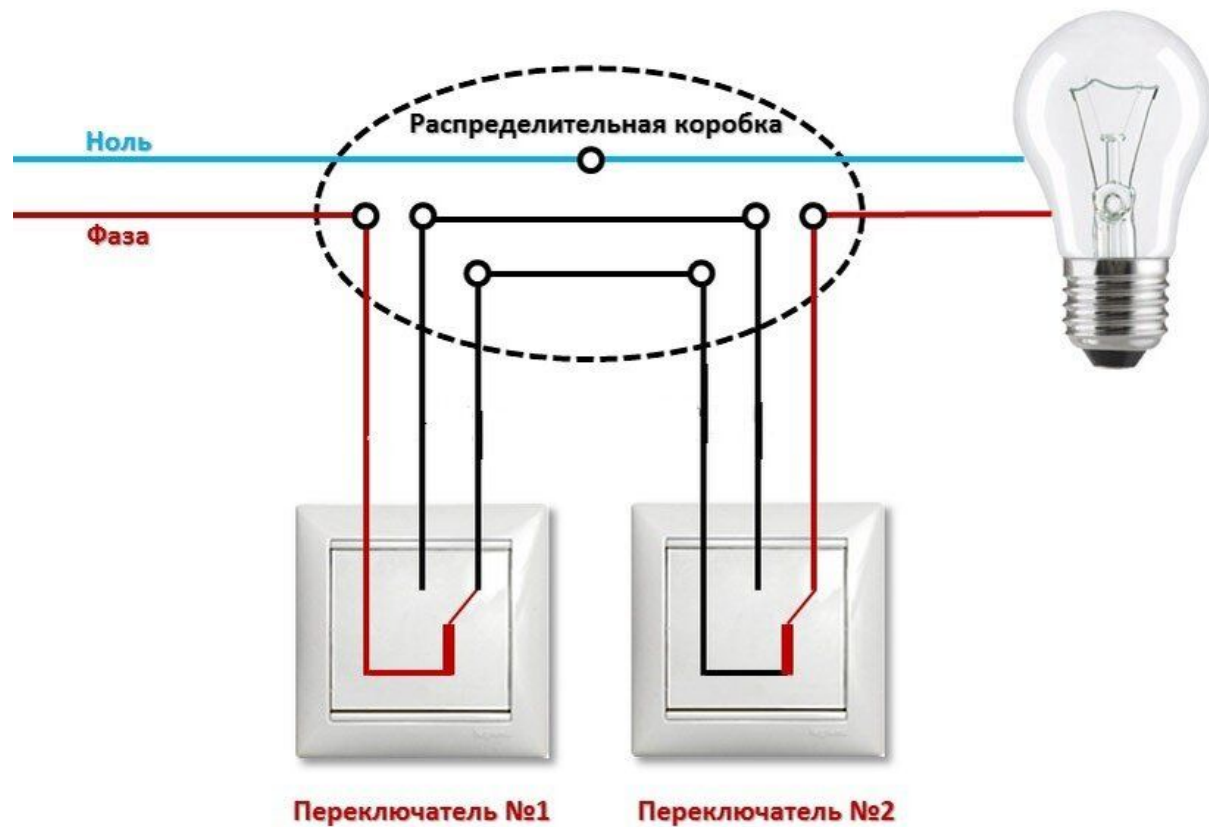
Текстовые

- IL – ассемблер (отменен)
- ST – текстовый ПаскалеПодобный

РОБОТ

ЭЛ. ШКАФ

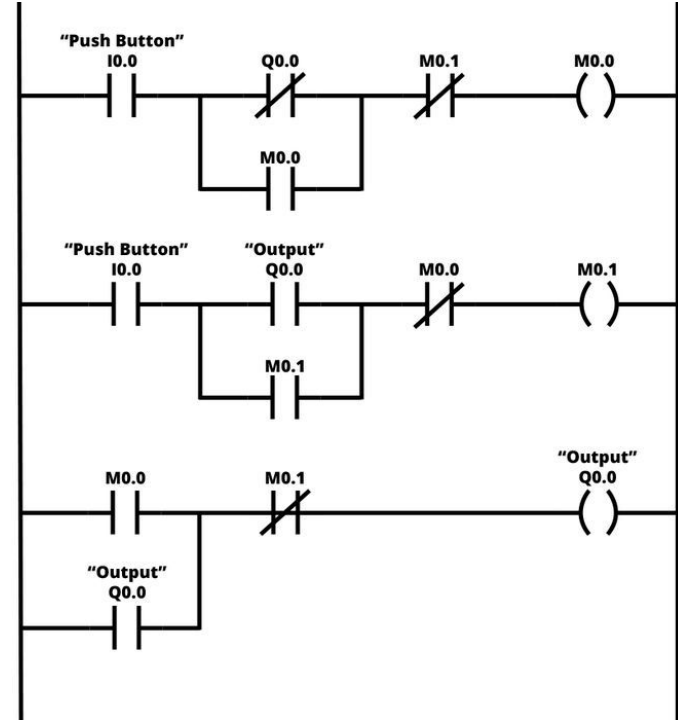




Релейные схемы

- Лампы и переключатели
- Сверху-вниз, слева-направо
- Бинарная логика

LD

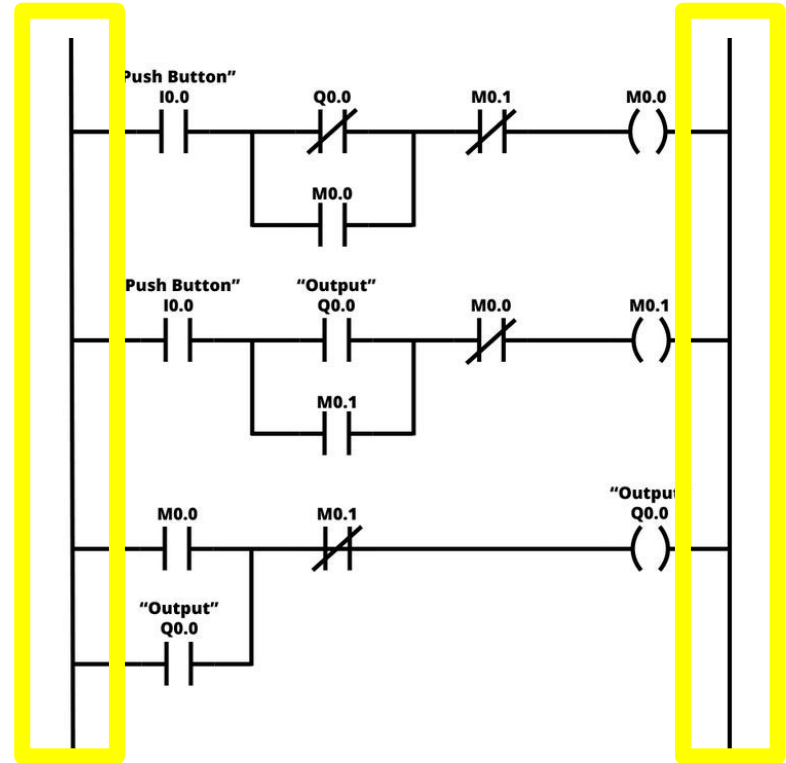


Релейные схемы

- Лампы и переключатели
- Сверху-вниз, слева-направо
- Бинарная логика

LD

ШИНЫ

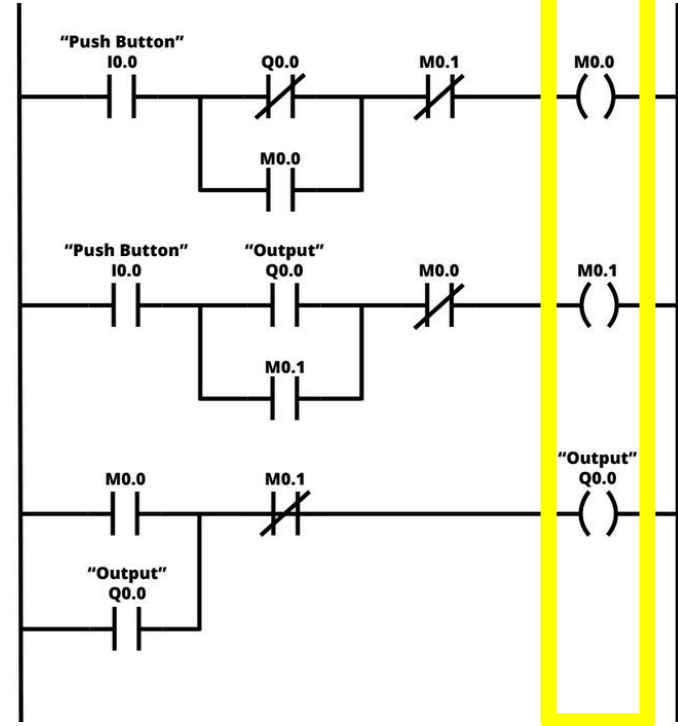


Релейные схемы

- Лампы и переключатели
- Сверху-вниз, слева-направо
- Бинарная логика

LD

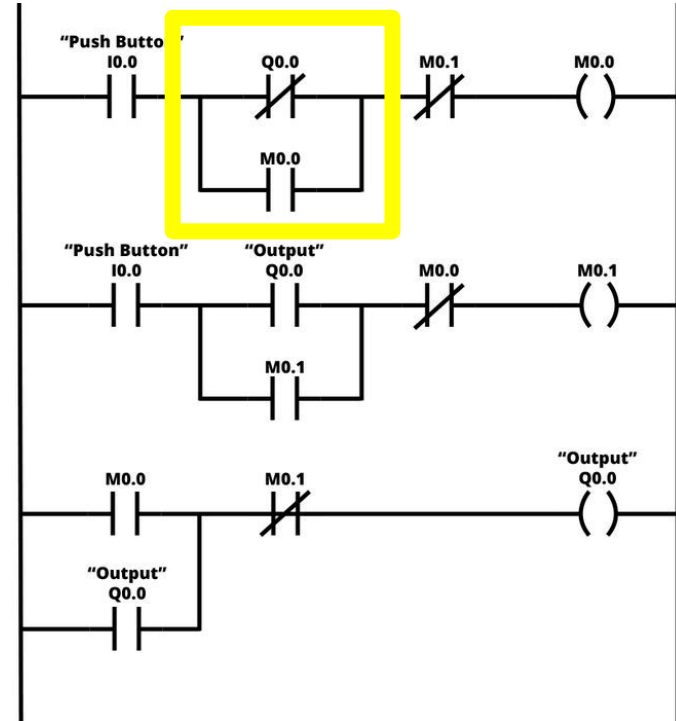
ВЫХОДА



Релейные схемы

- Лампы и переключатели
- Сверху-вниз, слева-направо
- Бинарная логика

OR



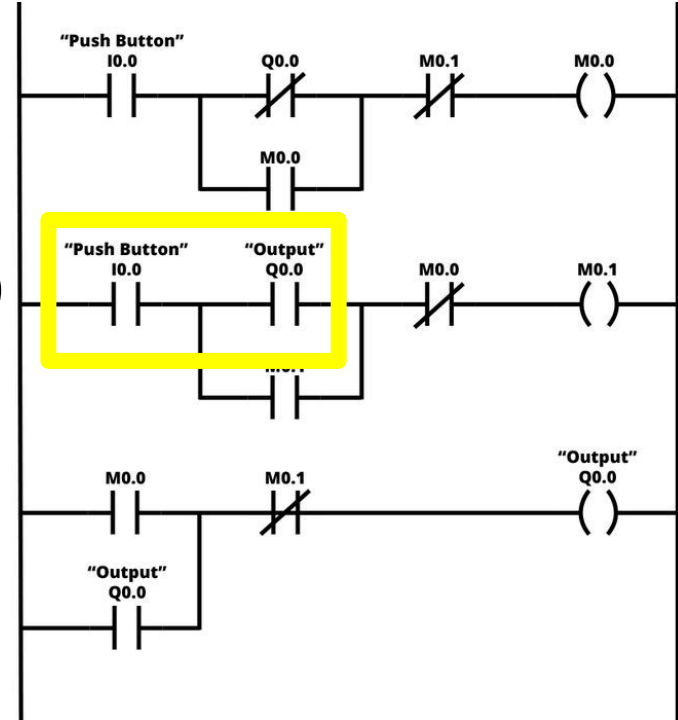
LD

Релейные схемы

- Лампы и переключатели
- Сверху-вниз, слева-направо
- Бинарная логика

LD

AND



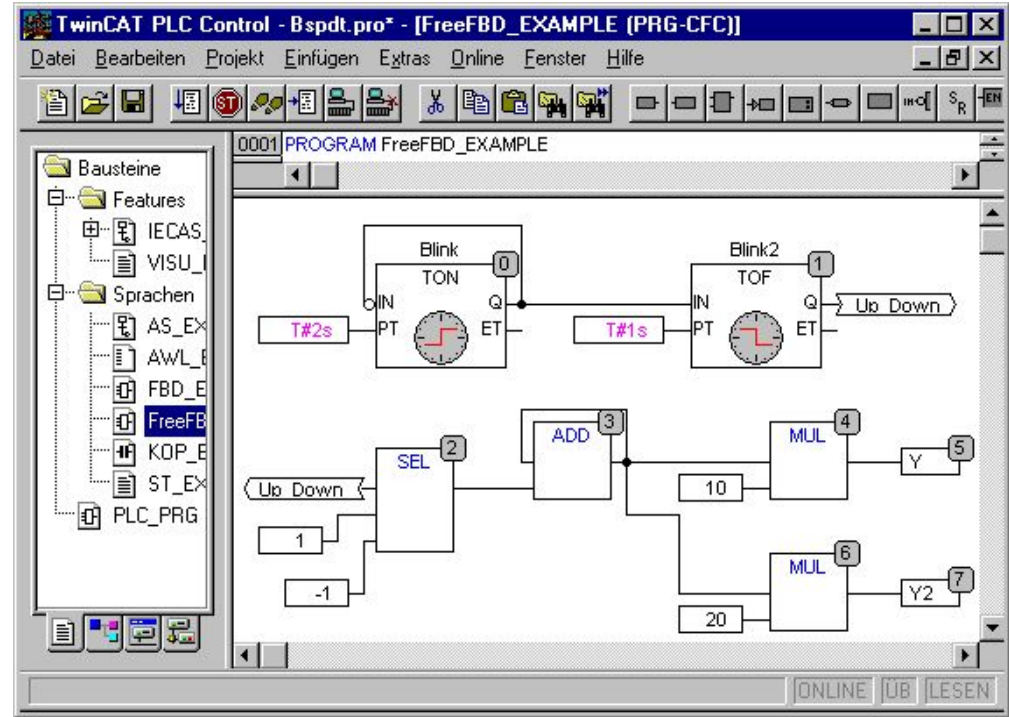
ПРОДУКТ



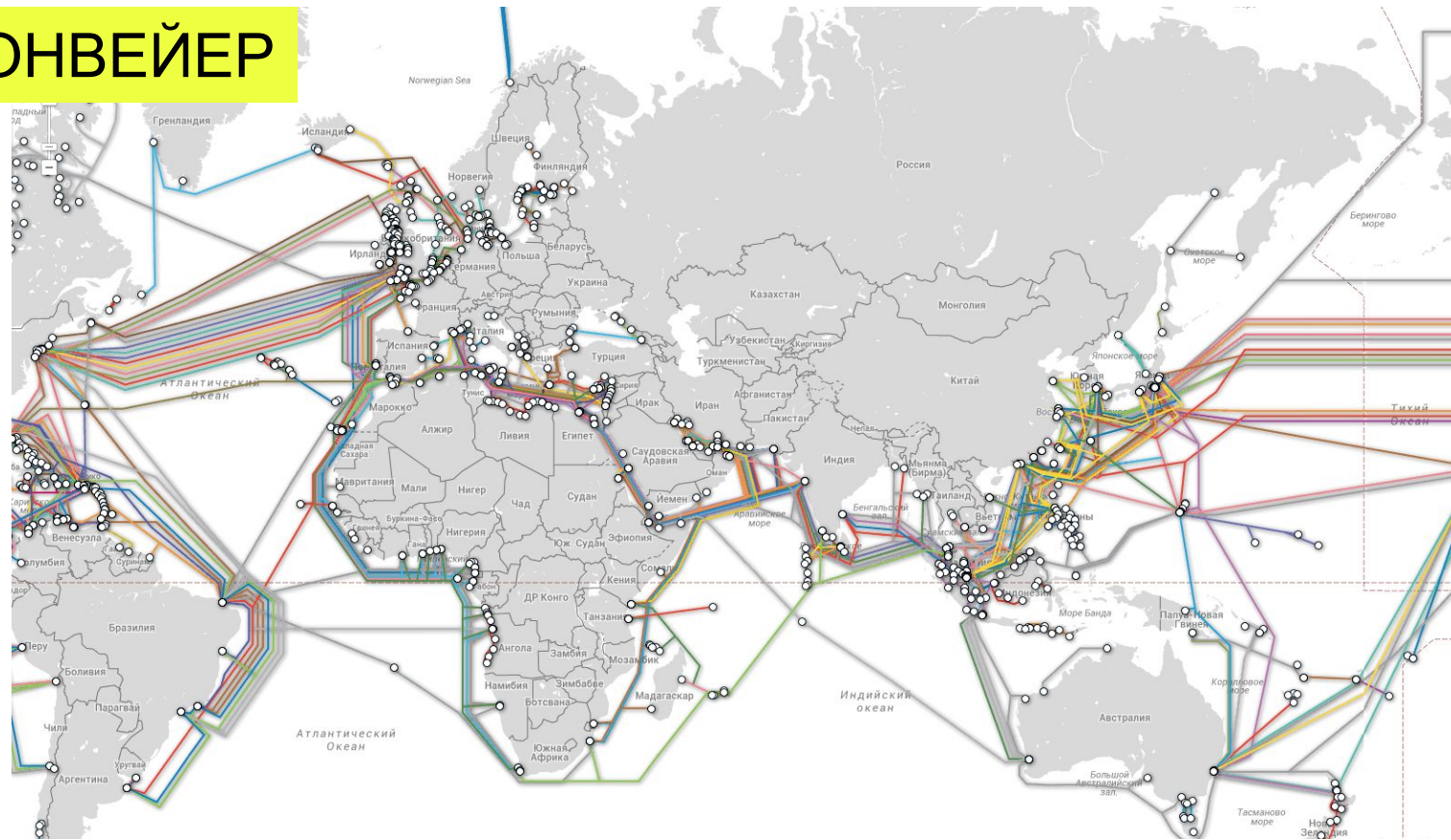
Функциональные блоки

- Функциональные блоки
- Устройство – функция
- Произвольная логика
- Порядок выполнения
FBD → CFC

FBD

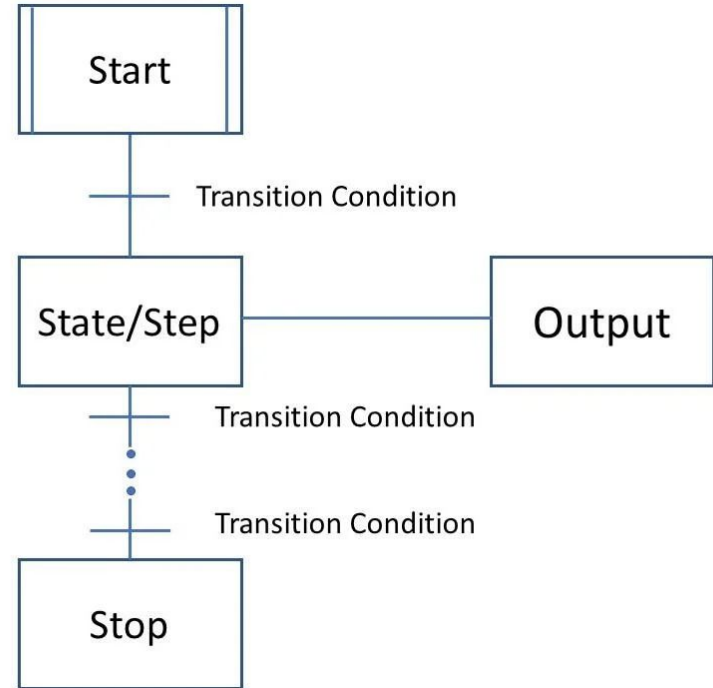


КОНВЕЙЕР



Диаграммы последовательностей

- Диаграмма состояний
- Условия перехода
- Сети Петри (цветные фишки)



SFC

РОБОТ



Структурированный текст

- Текст
- Похож на Паскаль
- Статическая типизация
- Компилируемый, но не всегда

STx

```
PROGRAM MAIN
VAR
    {attribute 'TcNcAxis' := 'Axis 1'}
    axis1: AXIS_REF;

    EN, SET, LOCK, UNLOCK: BOOL;

    mcSetMotorCtrlWord: FB_SoEAX5000SetMotorCtrlWord;
    mcPower: MC_Power;
END_VAR

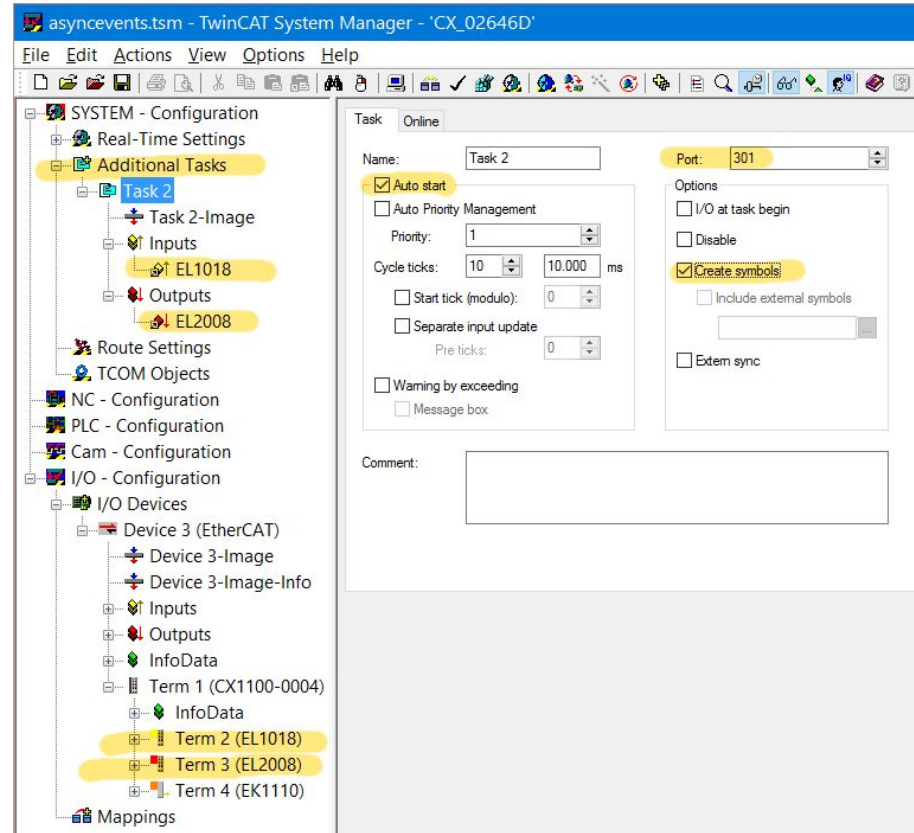
mcSetMotorCtrlWord(
    Axis := axis1,
    Execute := SET,
    ForceLock := LOCK,
    ForceUnlock := UNLOCK
);

SET R= NOT mcSetMotorCtrlWord.Busy;

mcPower(
    Axis := axis1,
    Enable := EN,
    Enable_Positive := TRUE,
    Enable_Negative := TRUE
);
```

Ввод-вывод

- Специальные переменные
 - Ввод %IB_
 - Вывод %QW_
 - Память %M_
 - RETAIN / Persistent
- Строгая типизация
- Конфигуратор



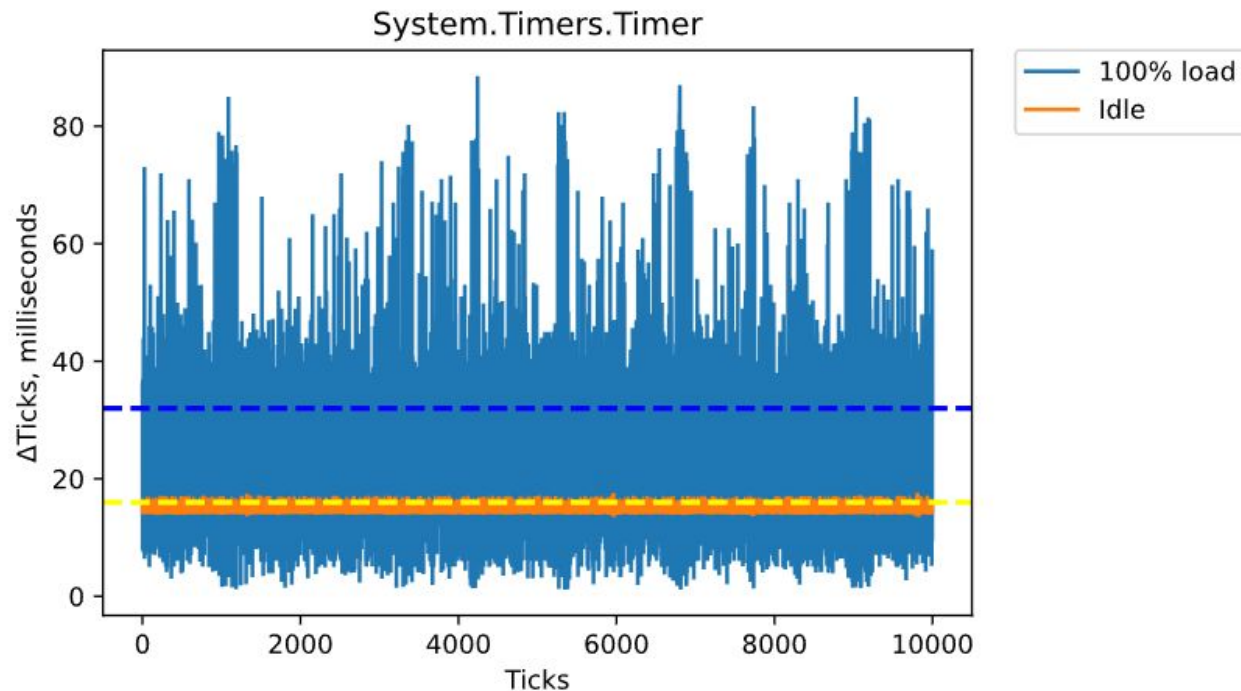
Машина состояний

```
PROGRAM MAIN
VAR
    T : TIMER( PT := T#10ms );
END_VAR

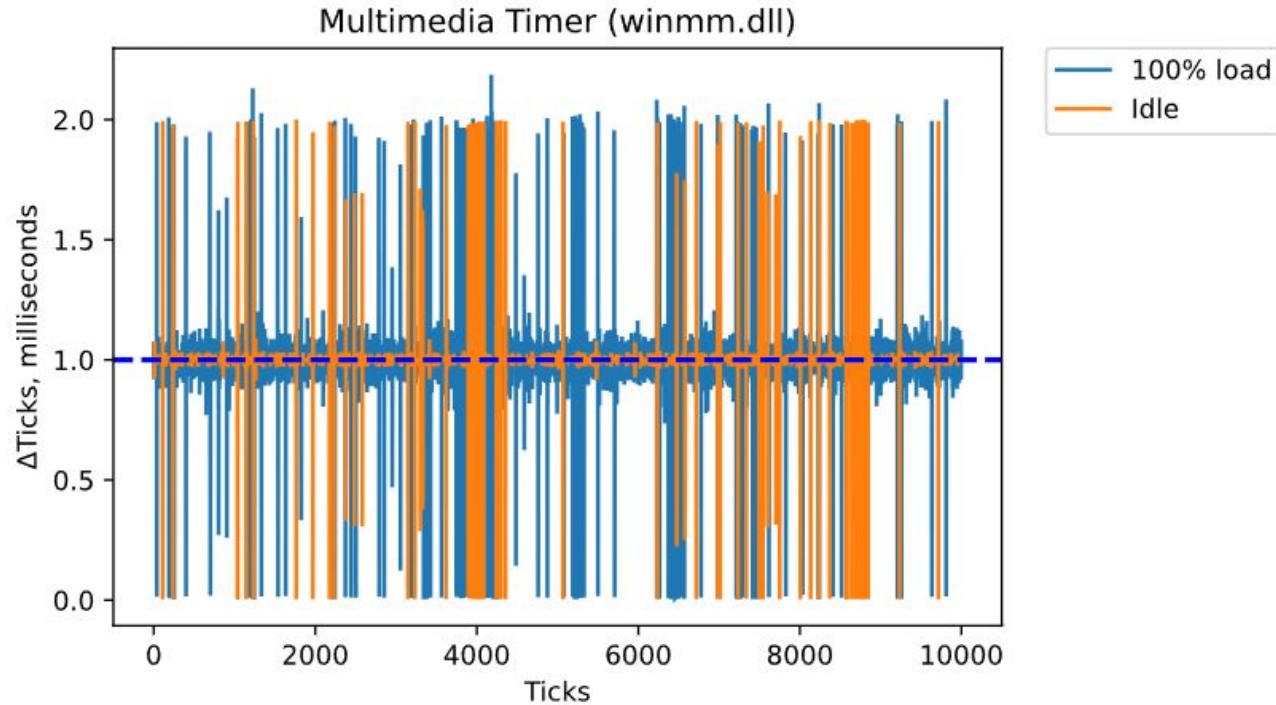
CASE main_state OF
0:
    T(EN := True);
    IF (T.Q) THEN
        main_state := 100;
    END_IF
100:
    T(EN := False);
    main_state := 0;
200:
END_CASE
```



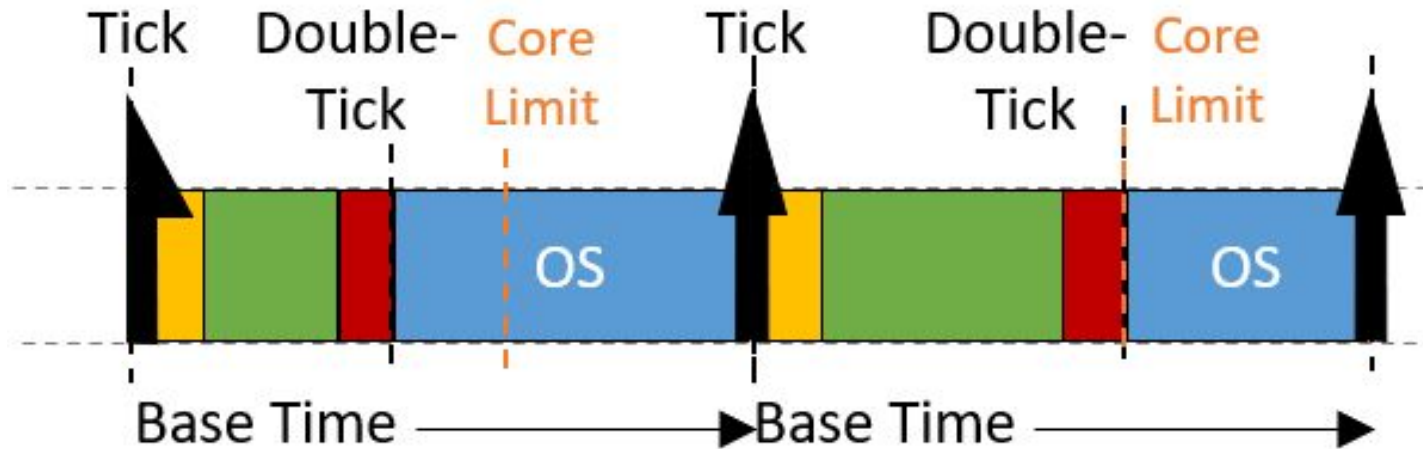
Системный таймер 16 мс = 0.016 с



Мультимедиа таймер 1 мс = 0.001 с

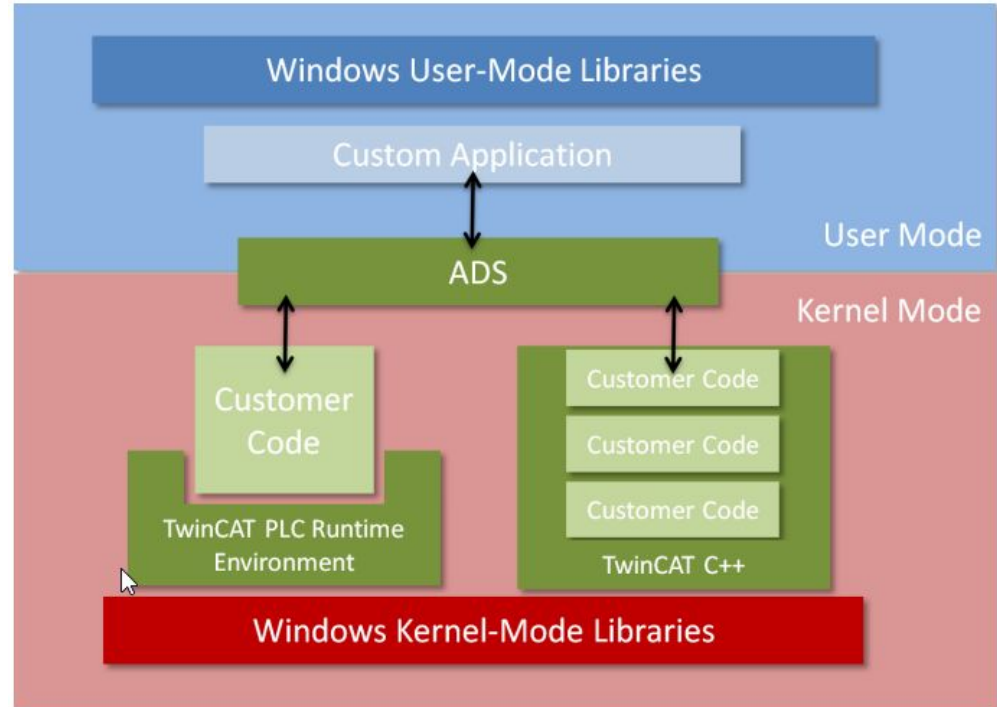


ОС реального времени



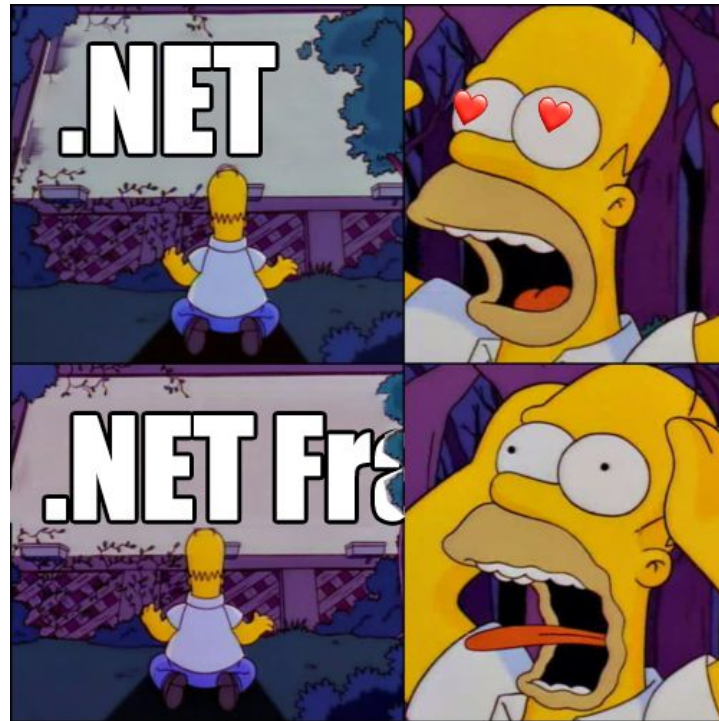
Связь с миром

- Общая память, HLI
- Шлюз. Gateway, ADS
- Обычные протоколы
 - Modbus -TCP
 - Ethernet, TCP/UDP
 - PROFINET
 - EtherCAT



Почему .NET?

- Windows (Wintel)
- WinForms + WPF + Avalonia UI
- Platform Invoke (P/Invoke)



```
using TwinCAT.Ads; // v4.4.0.0
```

```
namespace Examples.AdsReadSystemConfig
```

```
{
```

```
    class Program
```

```
    {
```

```
        const string TARGET_AMS = "127.0.0.1.1.1";
```

```
        const uint READ_IDX = 0x0000_02bc;
```

```
        const uint READ_OFF = 0x0000_0001;
```

```
        static void Main()
```

```
        {
```

```
            var client = new TcAdsClient();
```

```
            var ams = new AmsAddress( TARGET_AMS, AmsPort.R3_CTRLPROG );
```

```
            client.Connect( ams );
```

```
            int length = (int) client.ReadAny( READ_IDX, READ_OFF, typeof(int) );
```

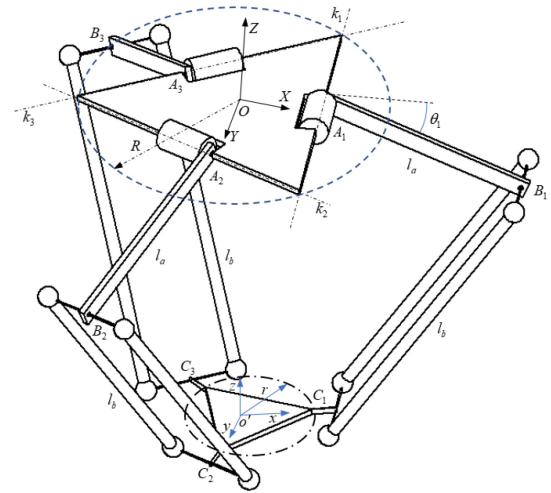
```
            var xmlStream = new AdsStream( length );
```

```
            client.Read( READ_IDX, READ_OFF, xmlStream );
```



Как управлять роборукой?

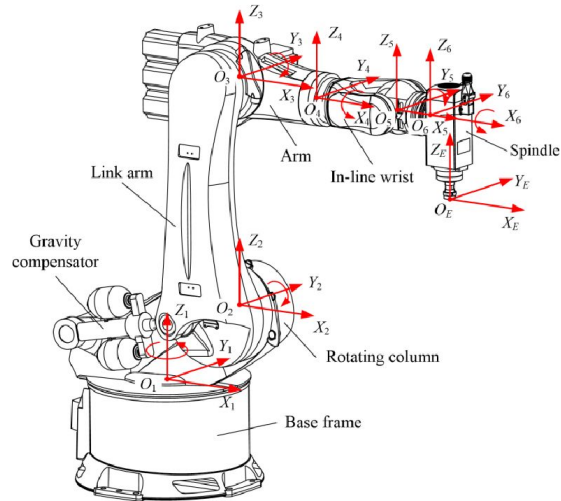
- Кинематические трансформации (мат. модель)
- CNC-ядро – драйвер операционной системы
- HLI-интерфейс (High Level Interface)
- ПЛК-программа, CoDeSys / TwinCAT
- Шлюз. Gateway, ADS
- HMI-оболочка, C#.NET WPF
- G-code, язык для станков с ЧПУ



Тяжелая промышленность и .NET

Николай Воронин

github: [nikvoronin](#)

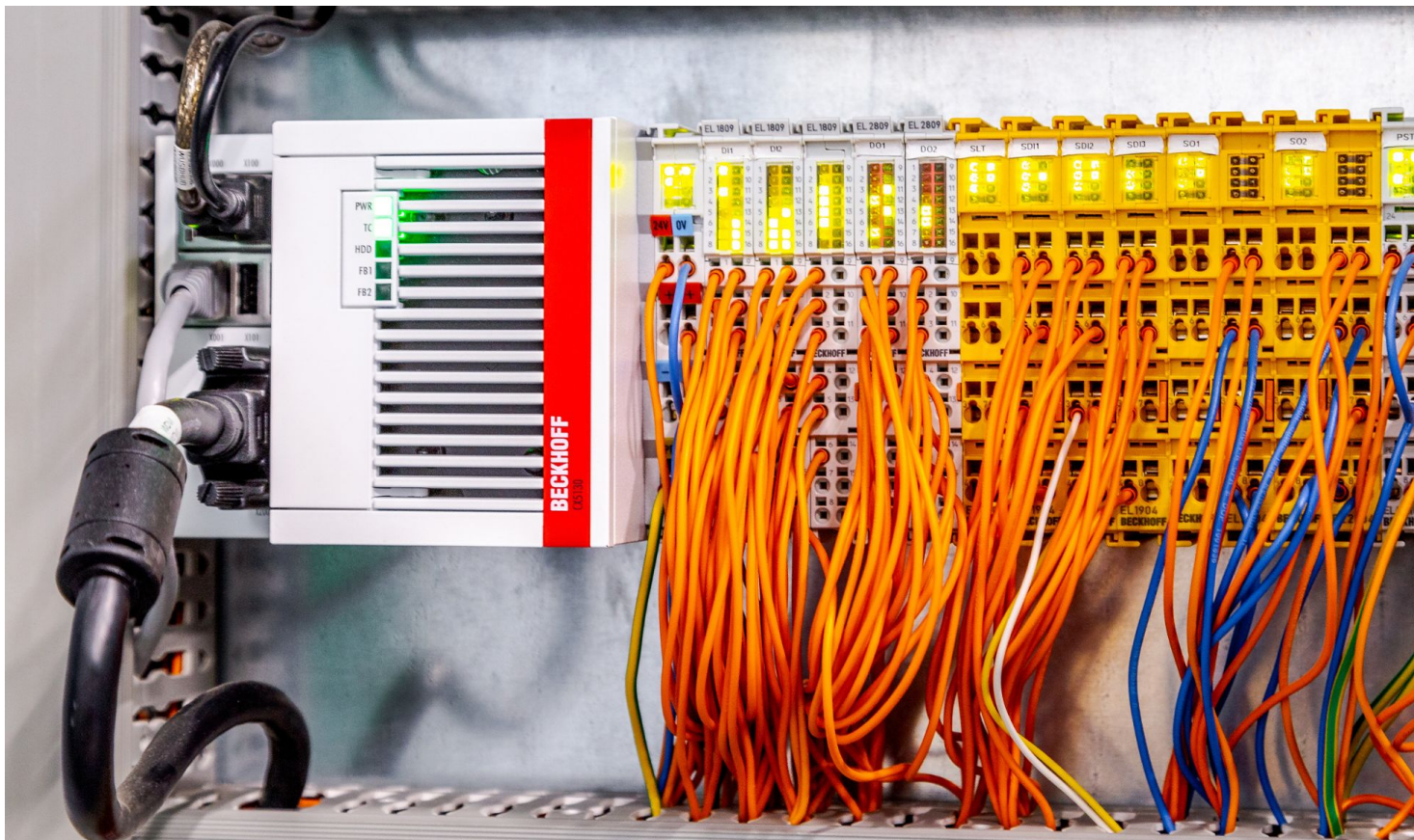


Go TwinCAT
блог



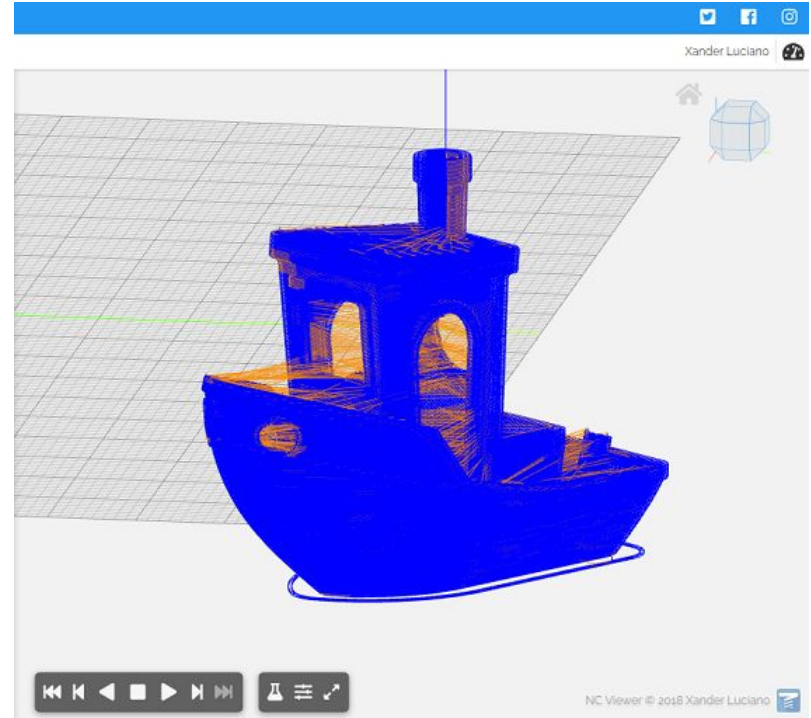
Петров И. В.
Программируемые
контроллеры





G-code

```
1 M190 S70.000000
2 M109 S200.000000
3 ;Sliced at: Mon 30-11-2015 17:55:51
4 ;Basic settings: Layer height: 0.25 Walls: 1.2 Fill: 100
5 ;Print time: 1 hour 21 minutes
6 ;Filament used: 5.61m 44.0g
7 ;Filament cost: None
8 ;M190 S100 ;Uncomment to add your own bed temperature line
9 ;M109 S200 ;Uncomment to add your own temperature line
10 G21 ;metric values
11 G90 ;absolute positioning
12 M82 ;set extruder to absolute mode
13 M107 ;start with the fan off
14 G28 X0 Y0 ;move X/Y to min endstops
15 G28 Z0 ;move Z to min endstops
16 G1 X100 Y100 Z50 F4200 ;move the platform down 15mm
17 G92 E0 ;zero the extruded length
18 G1 F200 E4 ;extrude 3mm of feed stock
19 G92 E0 ;zero the extruded length again
20 G1 F4200
21 ;Put printing message on LCD screen
22 M117 Printing...
```



Конфигуратор задач

The screenshot displays the 'Task configuration' window in the TwinCAT PLC Control software. The interface is divided into three main sections:

- Left Panel (POUs):** A tree view showing project files. Under the 'POUs' folder, there are three files: 'MAIN_1 (PRG)', 'MAIN_1A (PRG)', and 'MAIN_2 (PRG)'. 'MAIN_2 (PRG)' is currently selected and highlighted in blue.
- Center Panel (Task Configuration Tree):** A hierarchical tree structure for task configuration. The root node is 'Task configuration'. It branches into two tasks: 'FirstTask' and 'SecondTask'. 'FirstTask' is associated with 'MAIN_1' and 'MAIN_1A'. 'SecondTask' is associated with 'MAIN_2'. The 'SecondTask' node is currently selected and highlighted with a dashed border.
- Right Panel (Task Attributes):** A configuration panel for the selected task. It contains the following settings:
 - Name:** A text field containing 'SecondTask'.
 - Priority(0..3):** A numeric field containing '1'.
 - Type:** A group box containing four radio button options:
 - ☒ **cyclic**
 - ☐ freewheeling
 - ☐ triggered by event
 - ☐ triggered by external event
 - Properties:** A group box containing one text field:
 - Interval (e.g. t#200ms):** A text field containing 'T#5ms'.

Мультитядерность

TwinCAT 3 Tutorial

Settings Online Priorities C++ Debugger

Router Memory (MByte): 2

Available CPUs (Windows/Other): 2 2 Read from Target Set on target

CPU	RT-CPU	Base Time	CPU Limit	Latency Warning
0 (Windows)				
1 (Windows)				
2 (Other)	<input checked="" type="checkbox"/> Default	1 ms	100 %	(none)
3 (Other)	<input checked="" type="checkbox"/>	1 ms	100 %	(none)

Type	Object	RT-CPU	Base Time	Cycle Time	Cycle Ticks	Priority
TASK	PlcTask	Default (2)	1 ms	10 ms	10	20
TASK	PlcAuxTask	Default (2)	1 ms	(none)	0	50

ST процедурный

- Программа == Singleton
 - Функциональные блоки == процедуры
 - Функции == функции
 - Экшны (подпрограмма, без стеита)
 - Расширение (наследование) ФБ и структур
-
- Указатели
 - Статическое выделение памяти
 - Динамическое через системные вызовы

ST объектно-ориентированный

- Классы == функциональные блоки
- Наследование расширением
- Интерфейсы (ФБ и проперти без тела)
- Методы `.this`
- Свойства (property) с GET и SET
- Ограничение области видимости

