



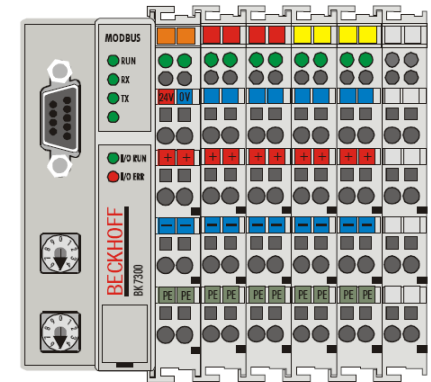
Графическая среда разработки программного обеспечения контроллера Полигон

ПромАвтоматика, Санкт-Петербург, Россия
www.pa.ru



Назначение ПО контроллера

- Исполнение алгоритмов управления:
 - Регулирование частоты и мощности
 - Реле положения НА
 - и т. д.



Управляющая программа работает под управлением DOS

- Языки низкого уровня (ассемблер)
- Языки высокого уровня (C, C++)
- Графический язык



Графическая разработка ПО

Достоинства:

- Визуальное представление алгоритмов управления
- Надежная, простая в обслуживании библиотека типовых элементов
- Простая и надежная структура программы
- Разделение процесса разработки ПО → повышенная эффективность
- Возможно отсутствие навыков программиста



Графическая разработка ПО

Стандарт программирования PLC IEC 61131-3:

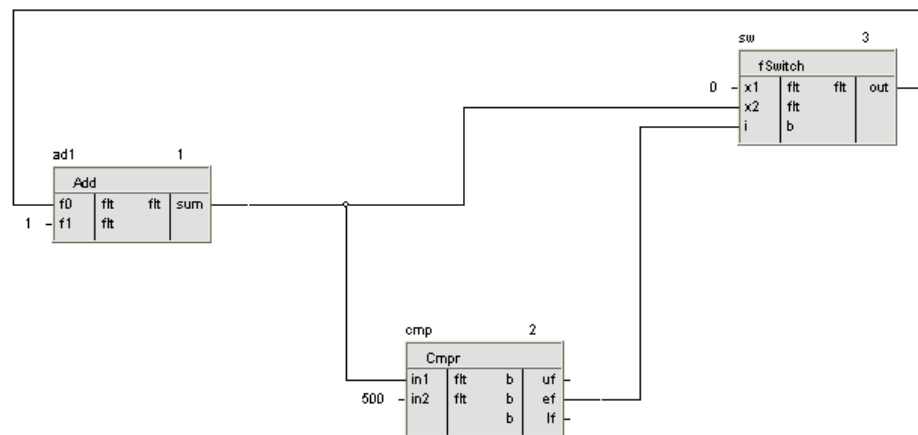
- Структурированный текст - Structured Text (ST)
- Язык функциональных блоков - Function Block Diagrams (FBD)
- Многоступенчатая схема - Ladder Diagram (LD)
- Список инструкций - Instruction List (IL)
- Последовательная структурная схема - Sequential Function Chart (SFC)



Графическая разработка ПО

Почему FBD:

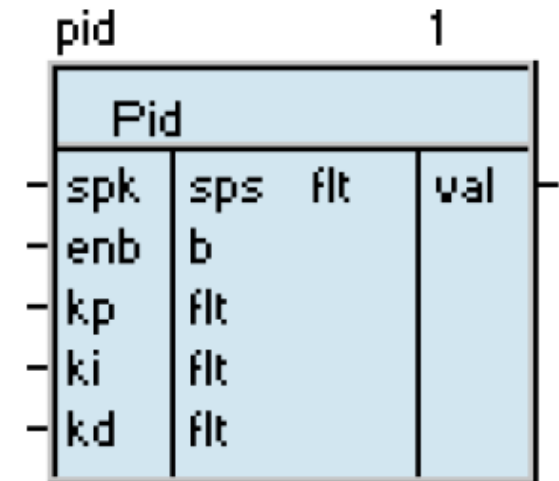
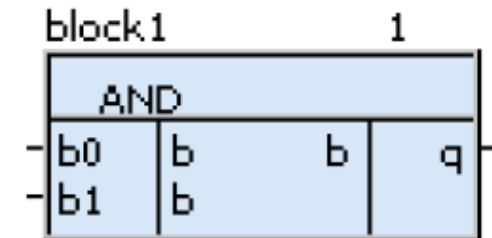
- Универсальный язык
- Соответствует объектно-ориентированному программированию
- Прост для понимания





Функциональный блок

- Элементарный алгоритм
- Преобразует входные значения к выходным, используя набор правил и преобразований - алгоритм
- Реализован как класс C++
- Имеет две основные функции: инициализация и выполнение

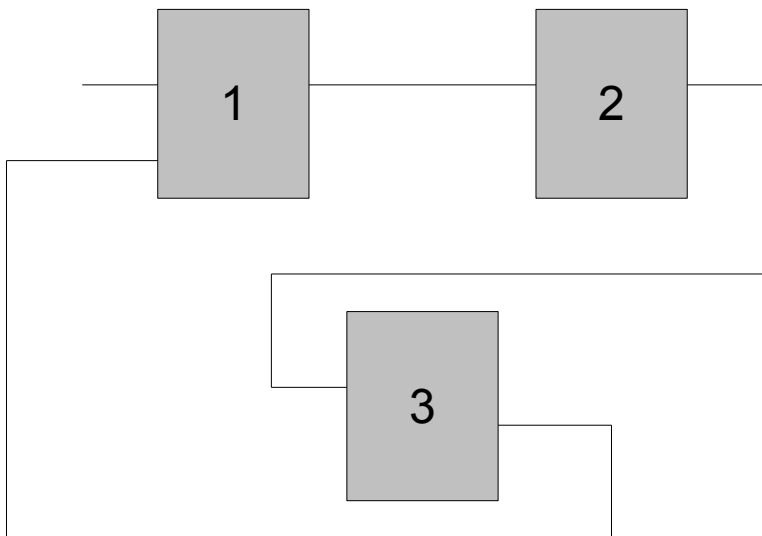




Поток данных

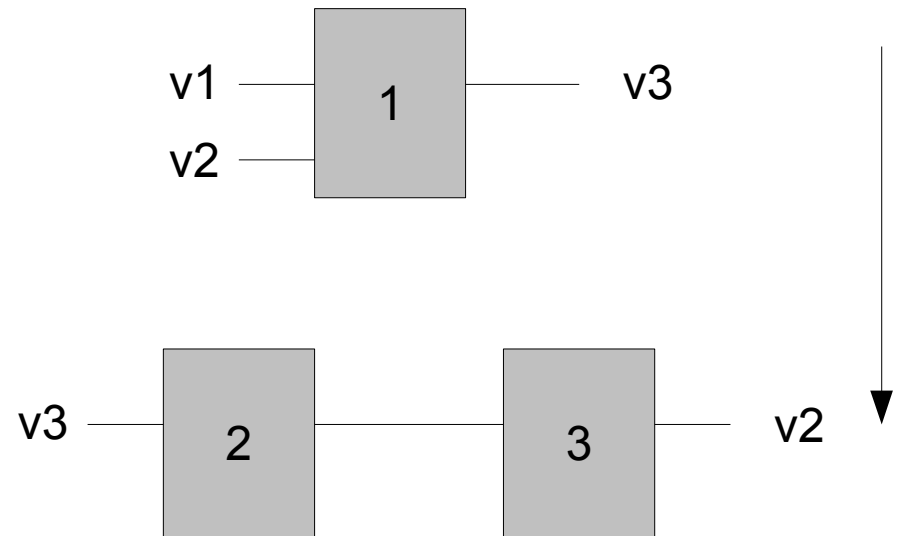
Полигон

- Только связи
- Порядок выполнения



IEC 61131-3

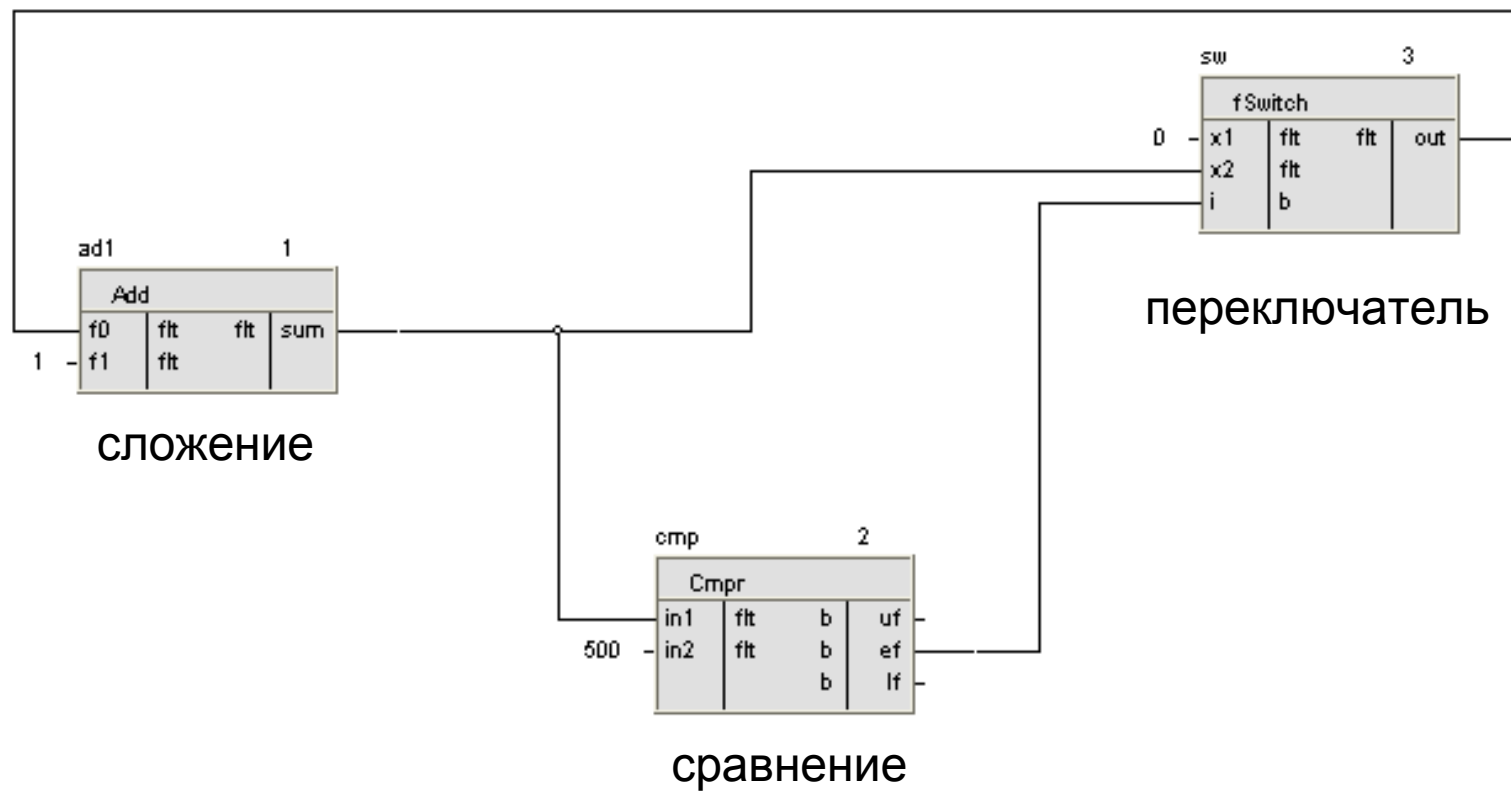
- Связи и переменные
- Порядок выполнения определяется потоком данных





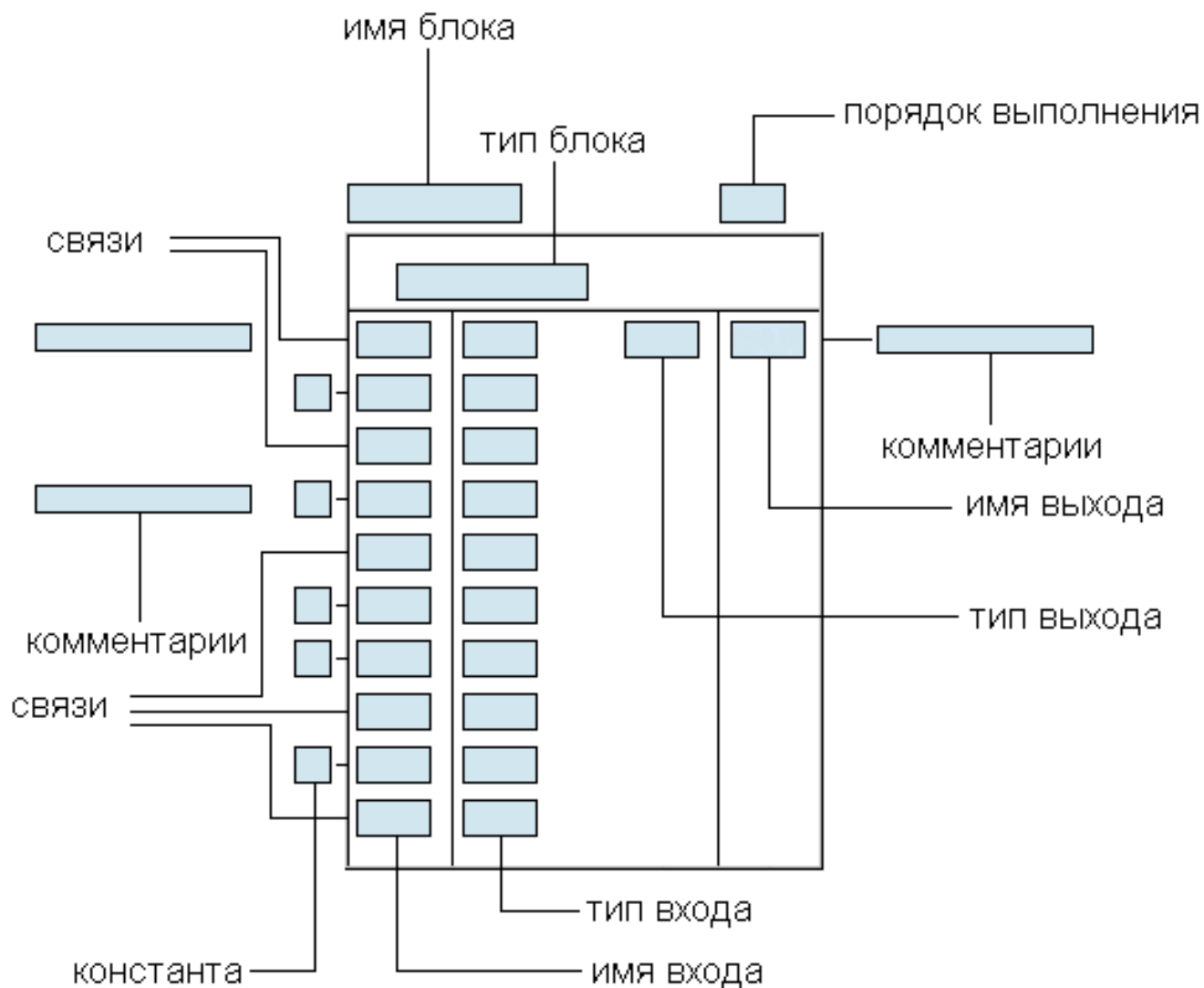
Пример

Счетчик





Функциональный блок



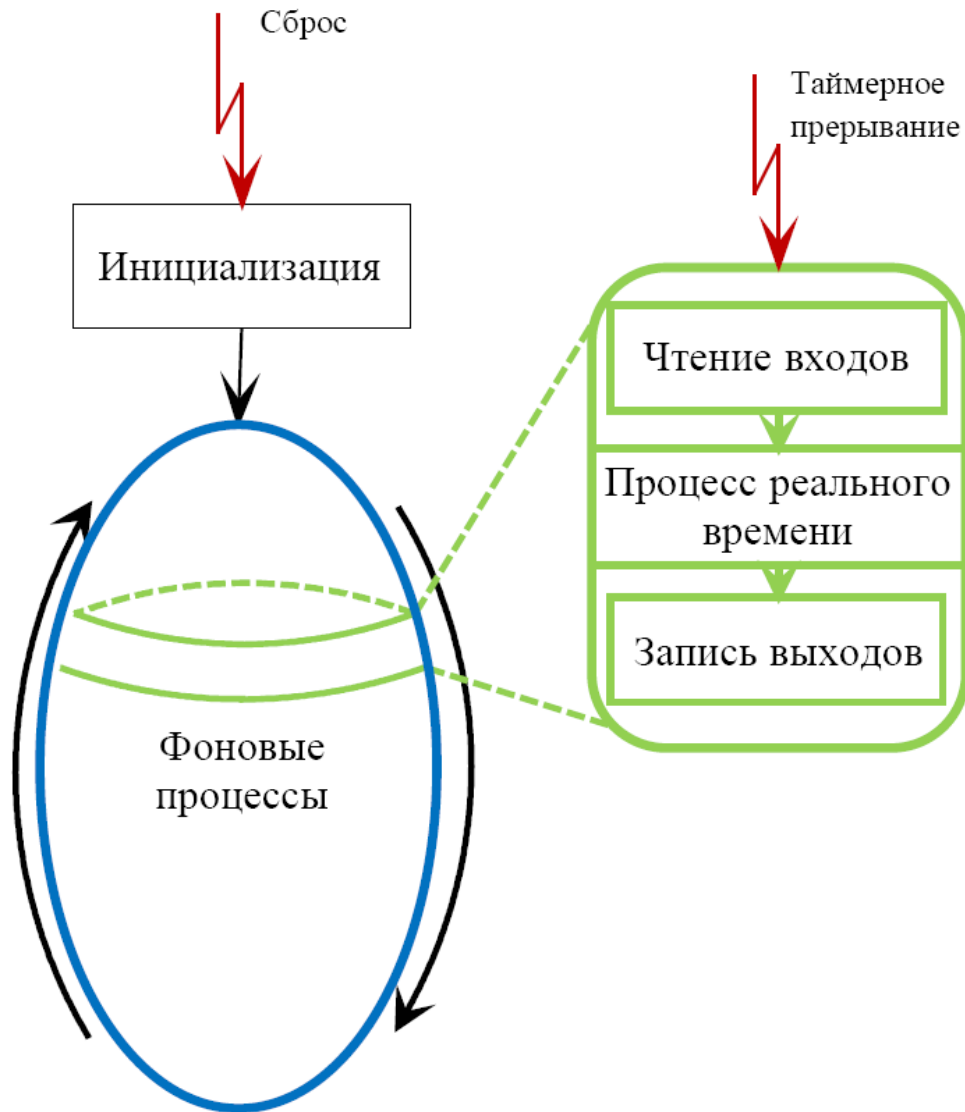


Типы данных

chr	байт	1	от -127 до 127
uch	байт без знака	1	от 0 до 255
flt	вещественное	4	от $3.4 \cdot 10^{-38}$ до $3.4 \cdot 10^{+38}$
lng	длинное целое	4	от -2147483648 до 2147483647
ulg	длинное целое без знака	4	от 0 до 4294967295
b	логический 0 или 1		
enm	перечисление	1	заданные значения
int	целое	2	от -32767 до 32767
uns	целое без знака	2	от 0 до 65535
any	указатель на любой тип		



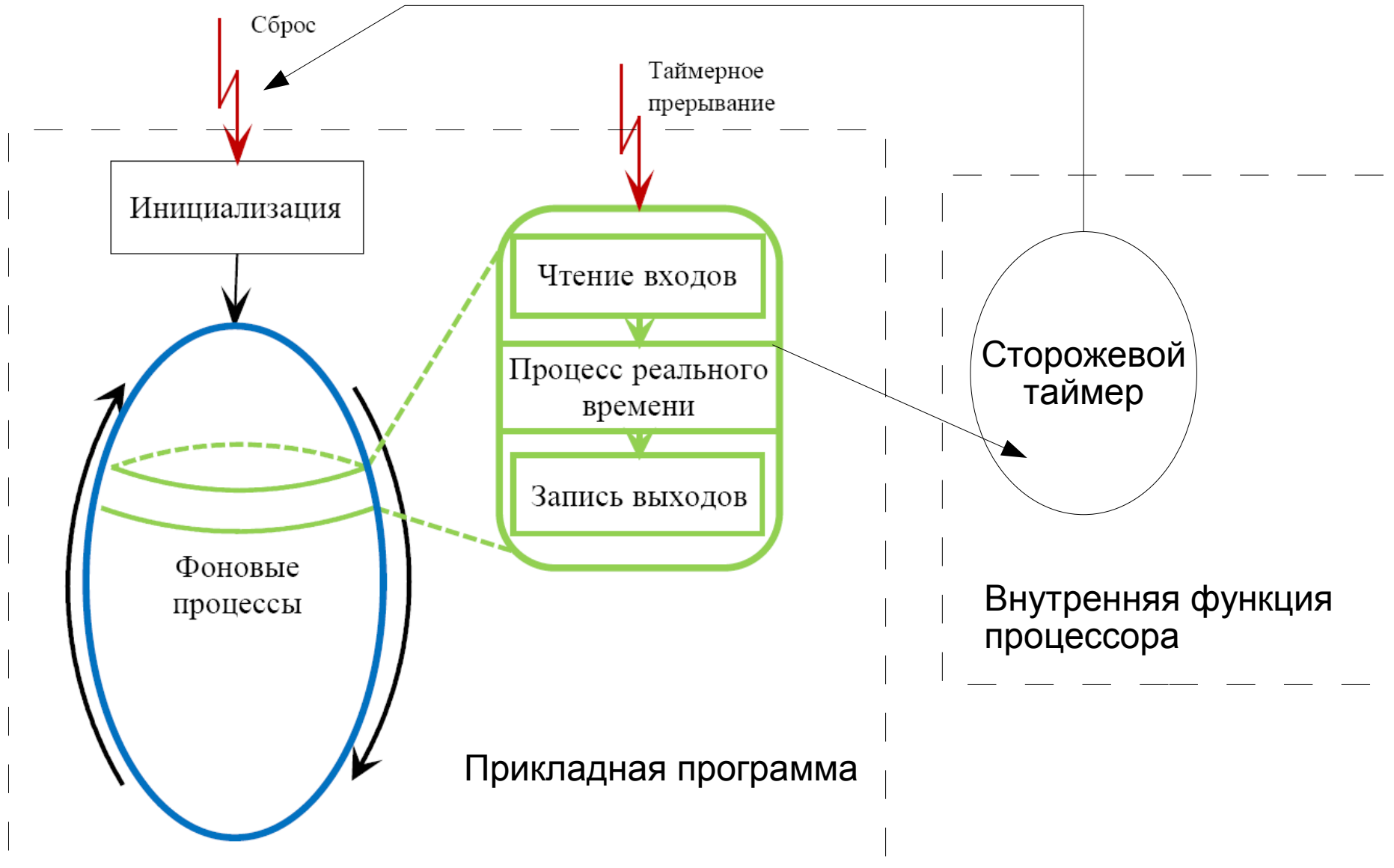
Структура управляющей программы



- Разделение критичных и некритичных процессов
- Гарантированное реальное время



Сторожевой таймер





Выполнение функциональных блоков

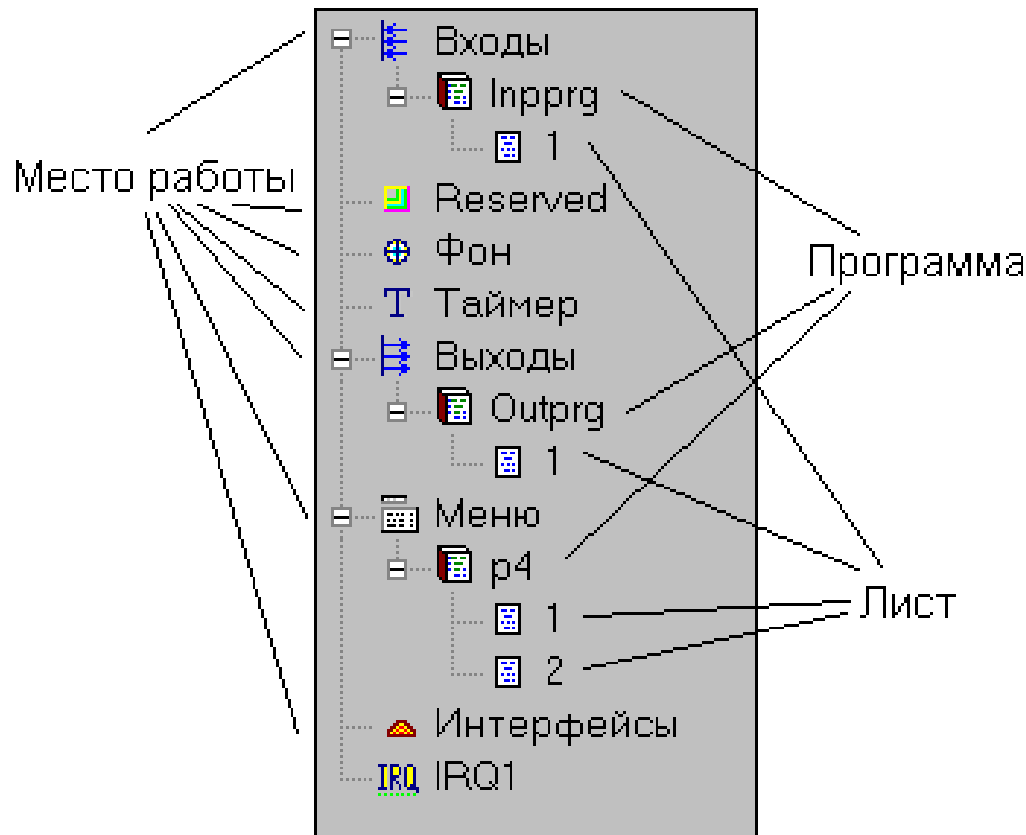
Место работы:

- Содержит функциональные блоки
- Определяет процесс где выполняется ФБ:
 - Таймер
 - Фон
 - IRQ



Структура управляющей программы

Иерархическая структура:
место работы
программа



- Место работы для разных функций:
 - Таймерный процесс:
Таймер
Входы
- Фоновый процесс:
Фон
Интерфейсы
Меню
- Обработчики прерываний:
IRQ n



Порядок выполнения

Место работы = процесс
выполнения



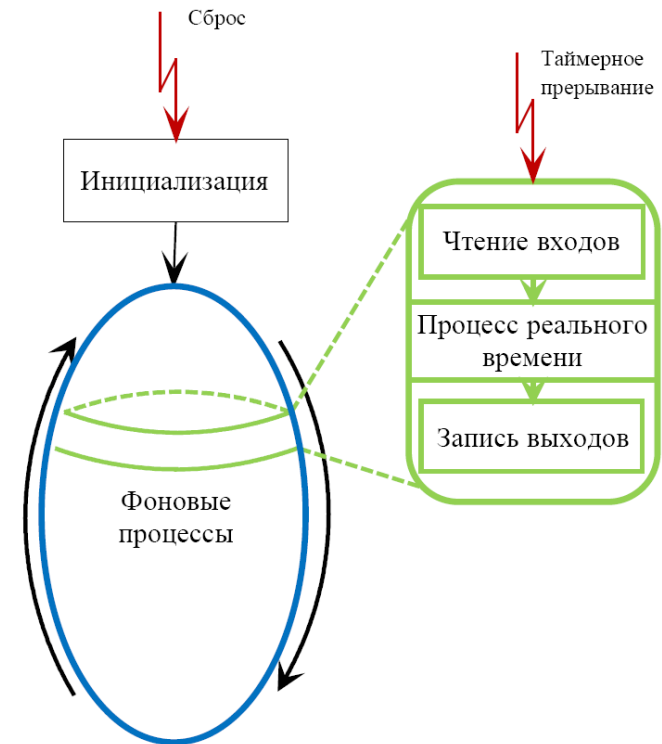
Порядок выполнения
программы



Номер листа



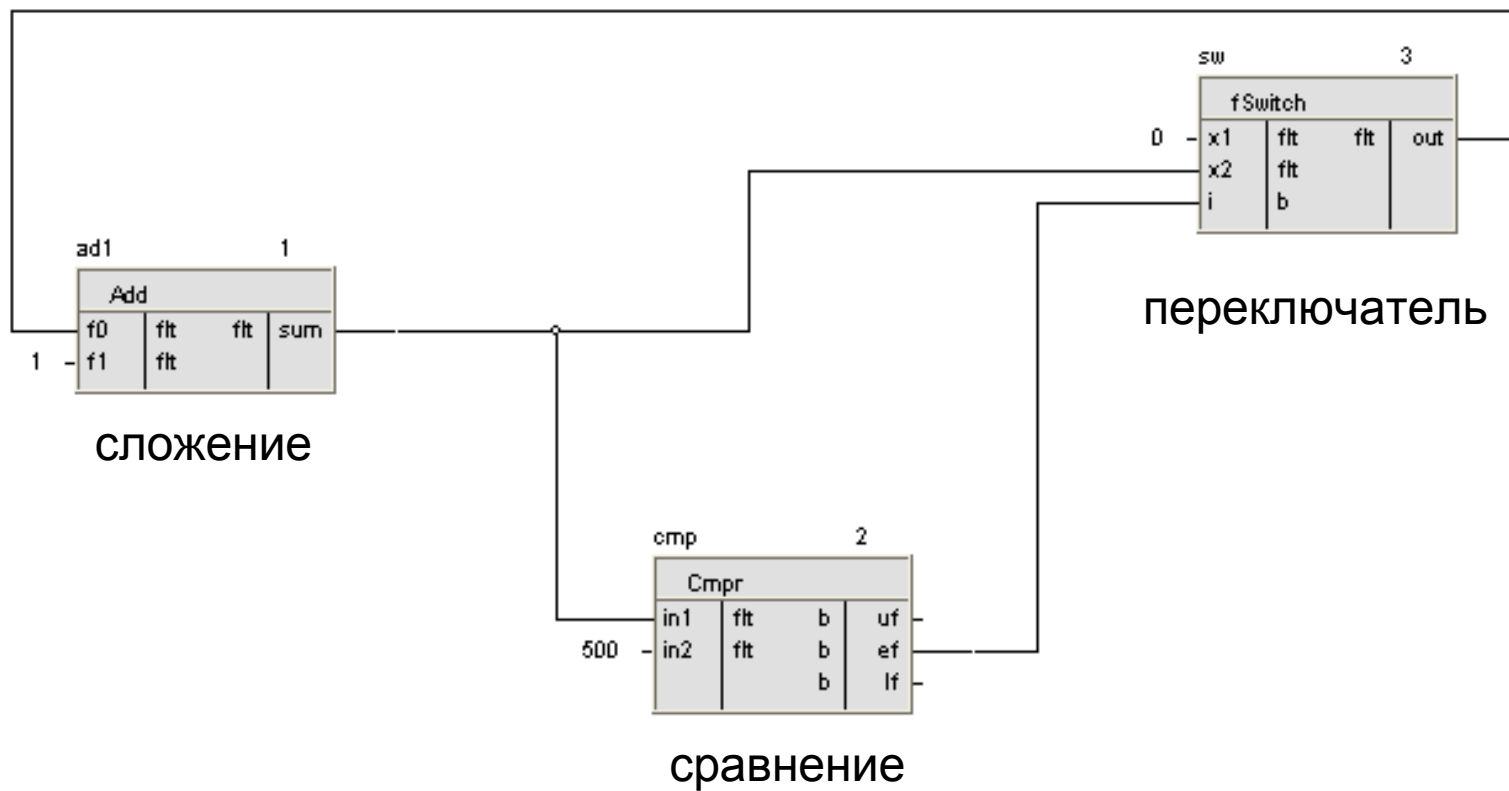
Порядок выполнения
функционального блока





Пример

Таймер → Program1 → 1





Система разработки Полигон

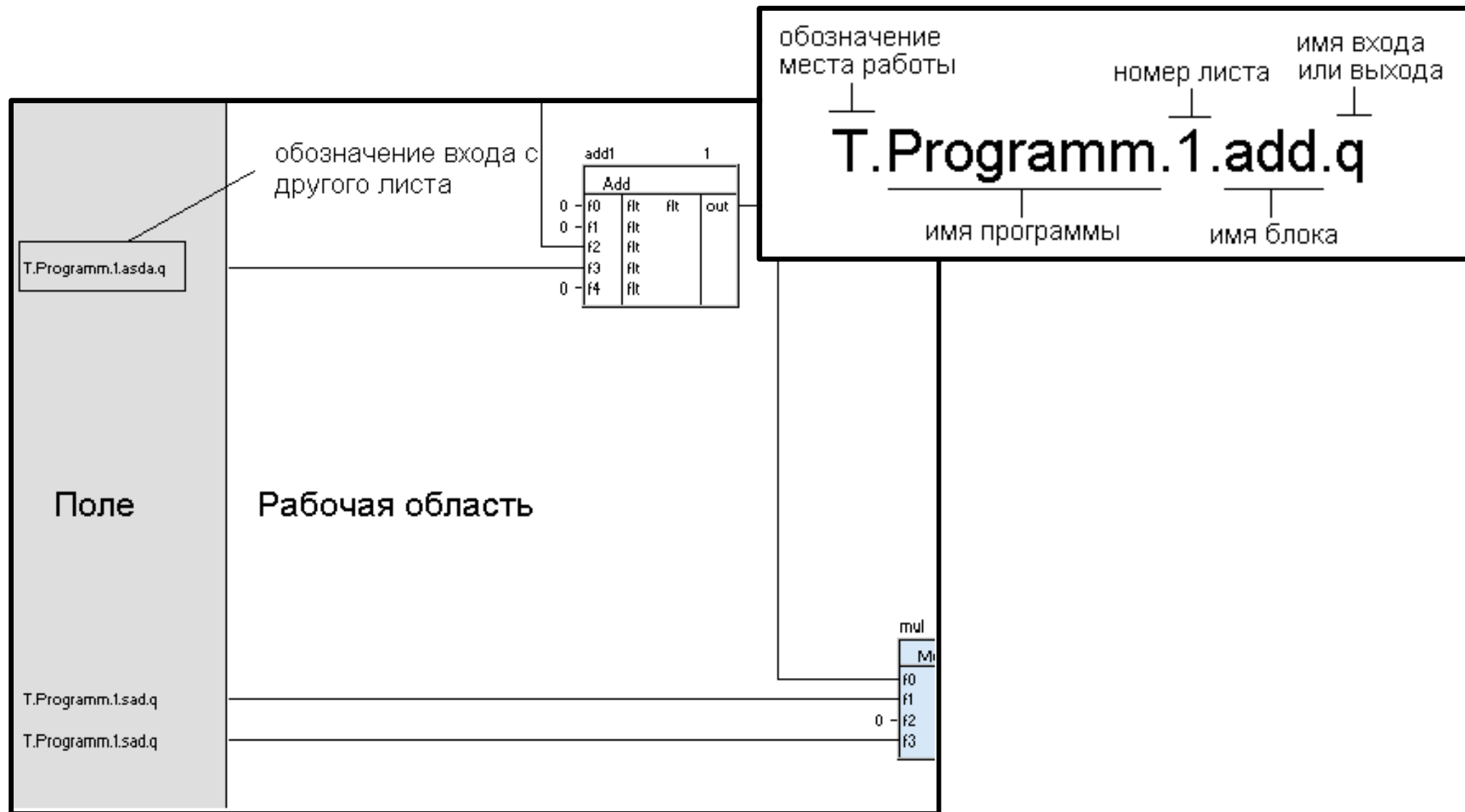
Возможности:

- Разработка управляющих алгоритмов:
 - Графическое представление алгоритма в виде нескольких соединенных функциональных блоков
 - Отладка программы в графическом виде
- Связь алгоритмов с входами/выходами оборудования
- Разработка новых функциональных блоков



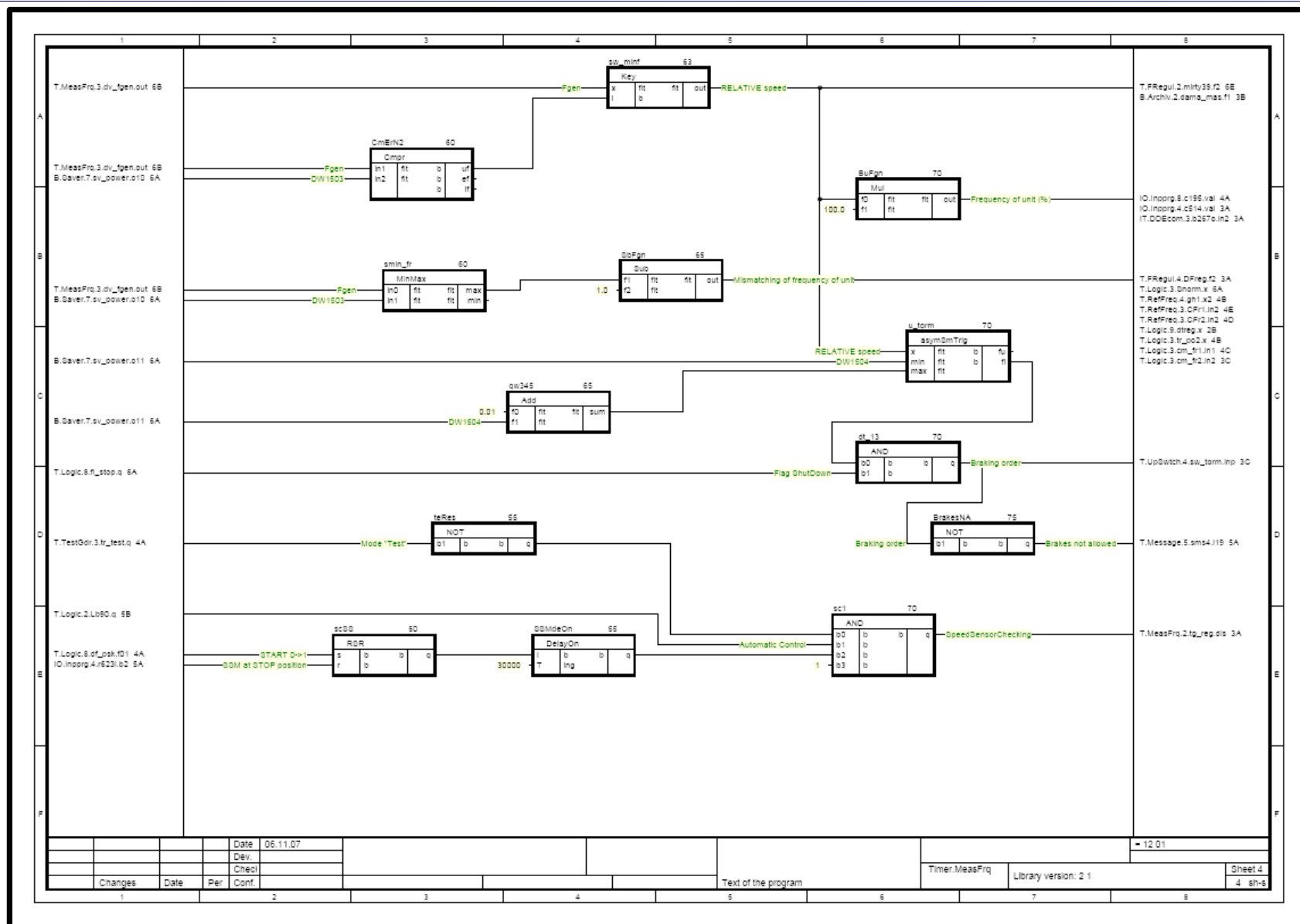


Представление листа



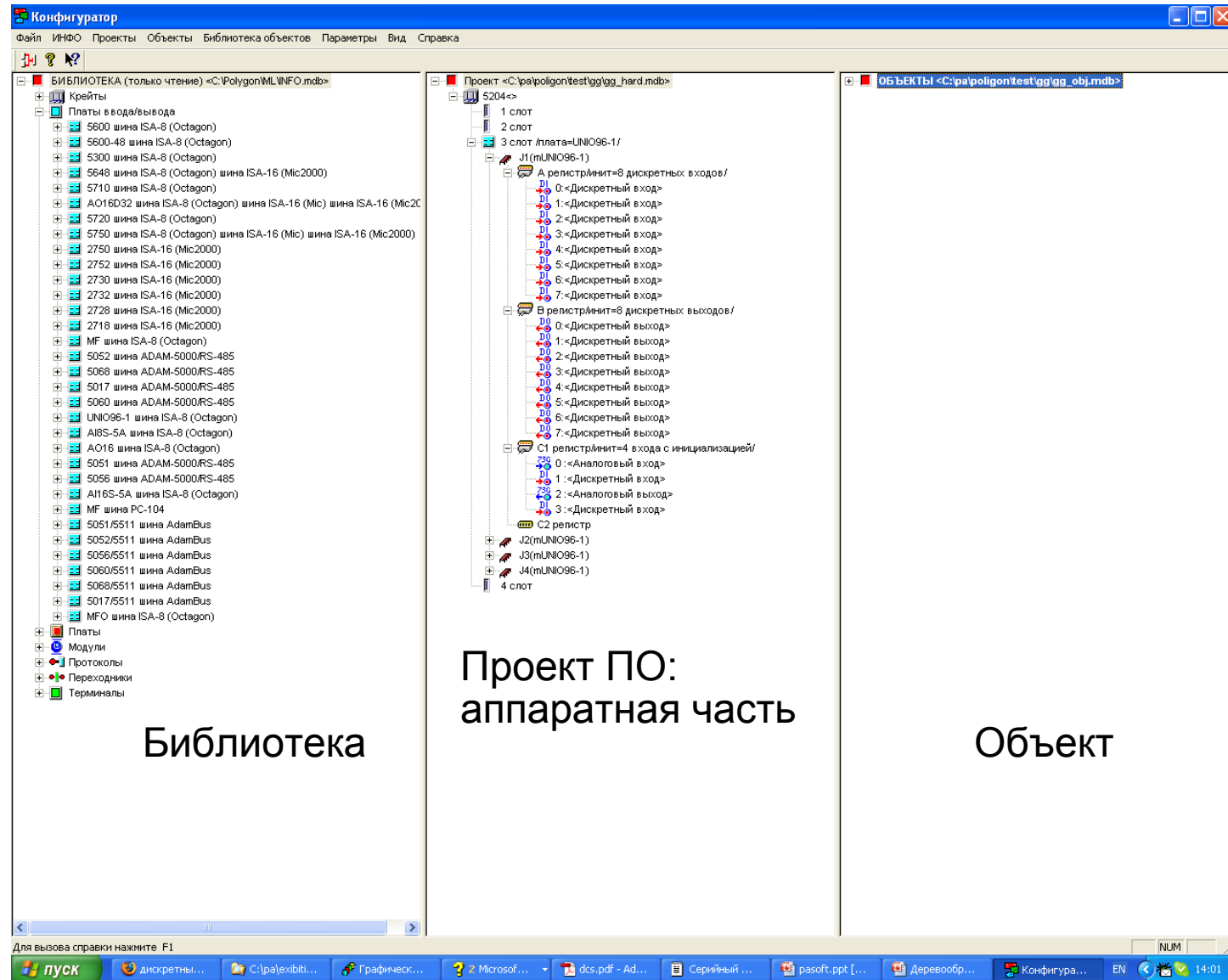


Документация из Полигона





Конфигуратор



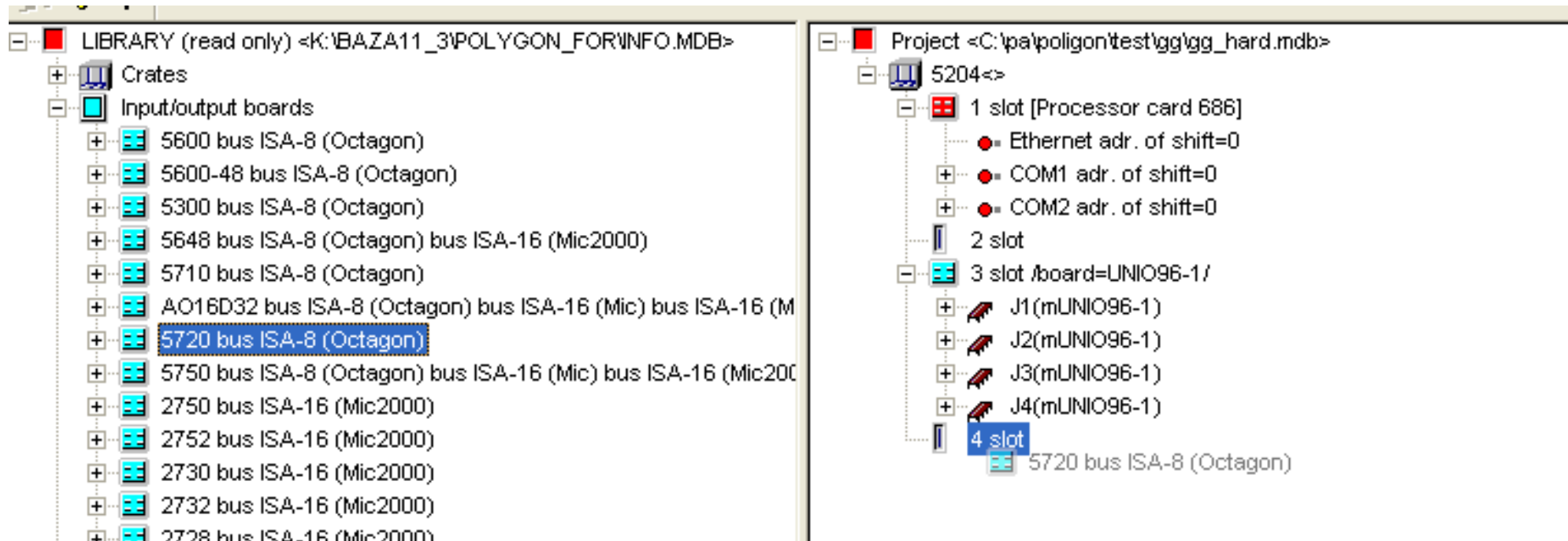


Конфигуратор

- Представление устройств в виде дерева
- Набор правил запрещает неправильную конфигурацию
- Создание аппаратной части проекта из библиотеки



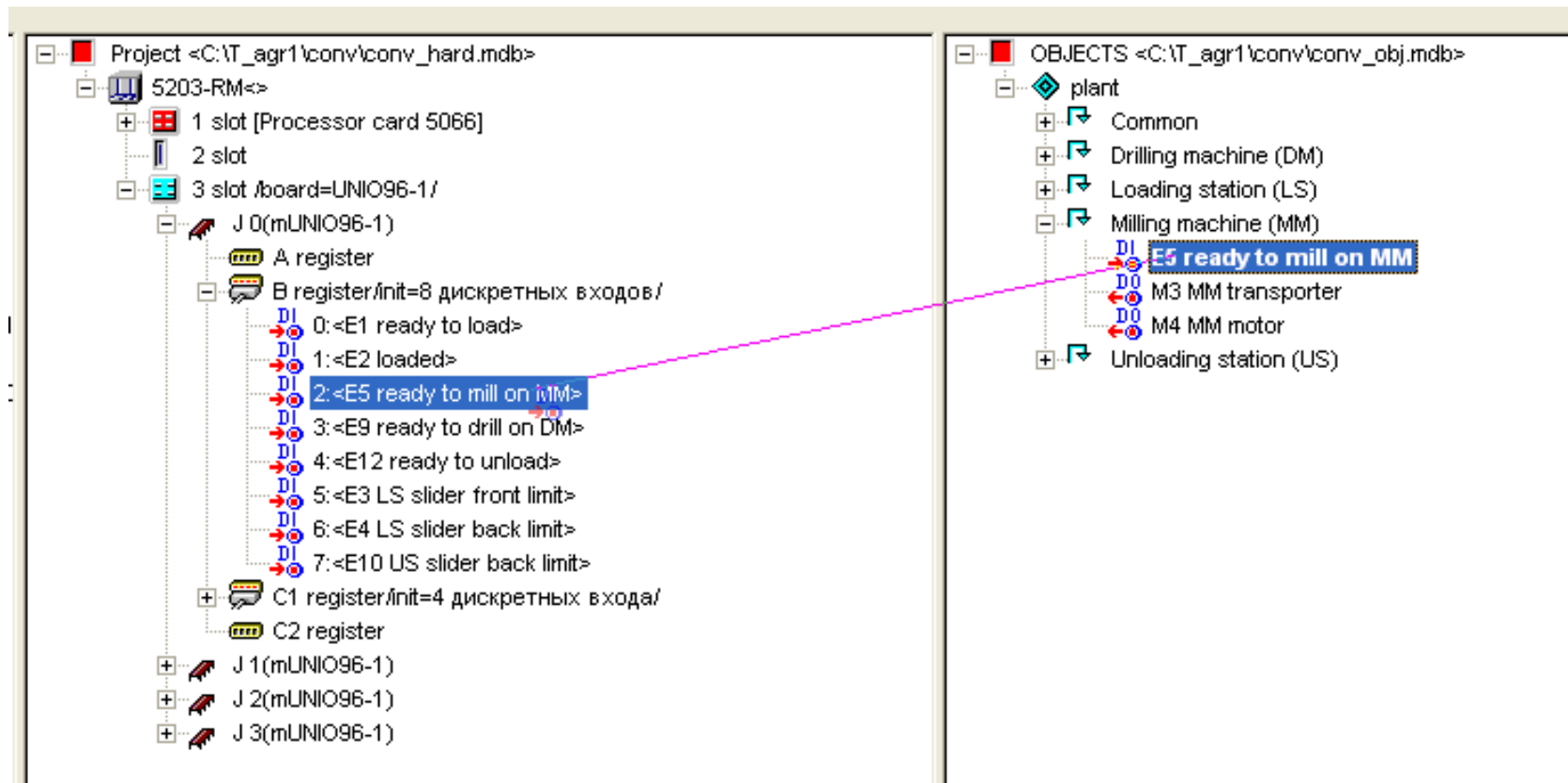
Конфигуратор



- Drag-and-drop элементов из библиотеки
- Параметры по правой кнопке мыши



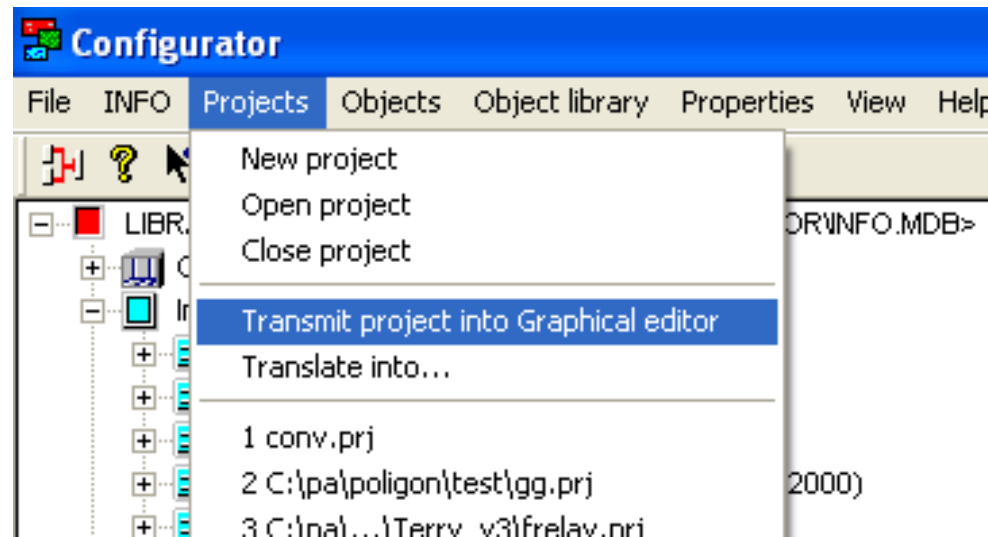
Конфигуратор



- Конфигурация объекта управления
- Drag-and-drop связь между аппаратной частью и объектом



Конфигуратор – Графический редактор



- Спецификация аппаратной части - Конфигуратор
- Алгоритмы - Графический редактор
- Связь между алгоритмами и аппаратной частью – Графический редактор



- Добавить устройства из библиотеки
- Связать объект с аппаратными средствами
- Конвертировать в Графический редактор

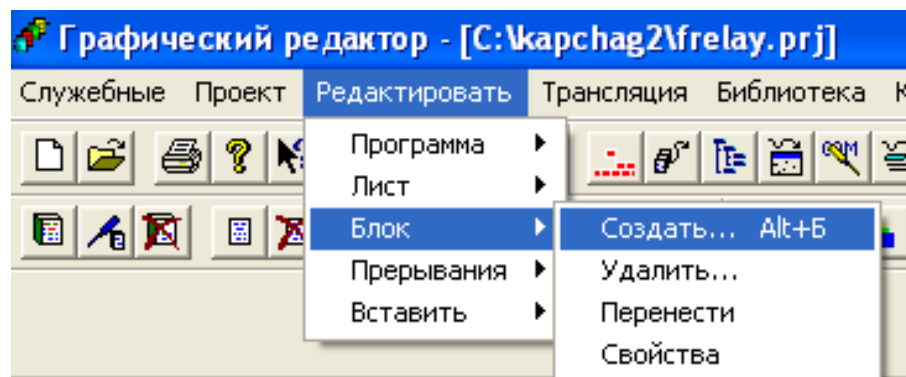
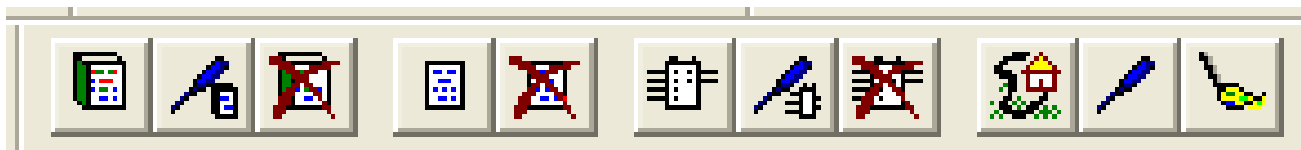




Этапы разработки ПО

2. Разработка алгоритмов в Графическом редакторе

- Создать структуру программ и листов
- Создать алгоритмы из функциональных блоков
- Связать алгоритмы с аппаратными средствами

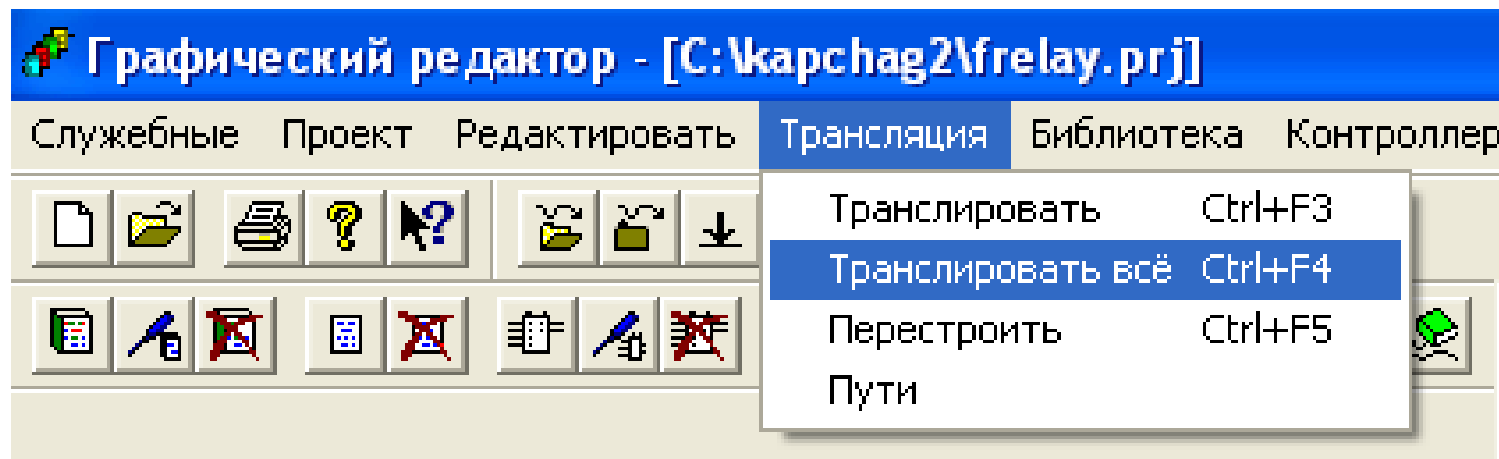




Этапы разработки

3. Конвертировать в исполняемый файл

- Каждый функциональный блок имеет запись в библиотеке
- Автоматическая **Трансляция** создает .exe файл
- Выполняемый файл можно запускать на контроллере под DOS





Этапы разработки

4. Закачать выполняемый файл в контроллер



Для Fastwell/Octagon

- Закачать
- Файловая система
- Консоль

- Переписать на карту памяти

Для Beckhoff



Этапы разработки

5. Отладка

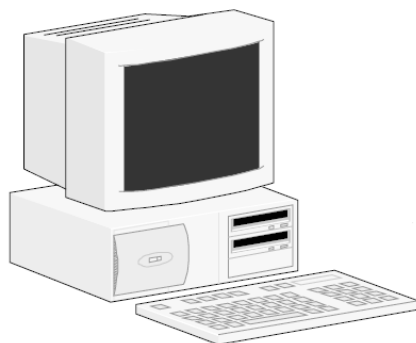
- Запуск/остановка программы (Fastwell/Octagon)
- Удаленная консоль (Fastwell/Octagon)
- Удаленный отладчик



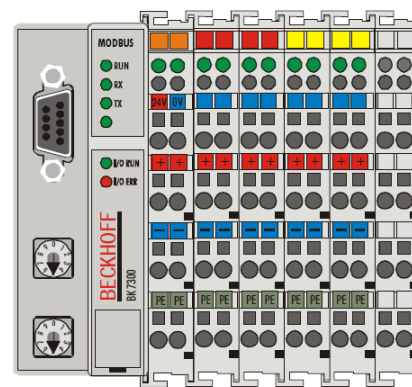
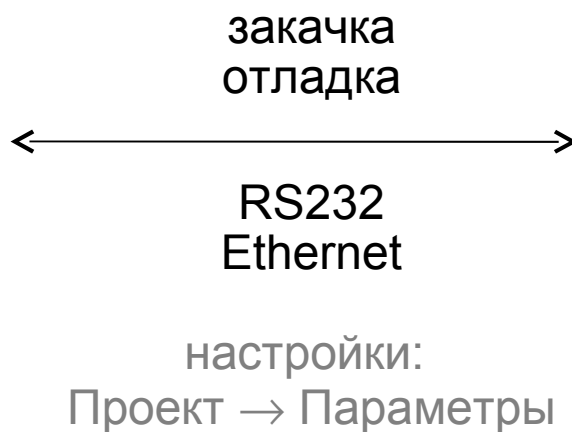


Этапы разработки

5. Отладка



разработка
трансляция

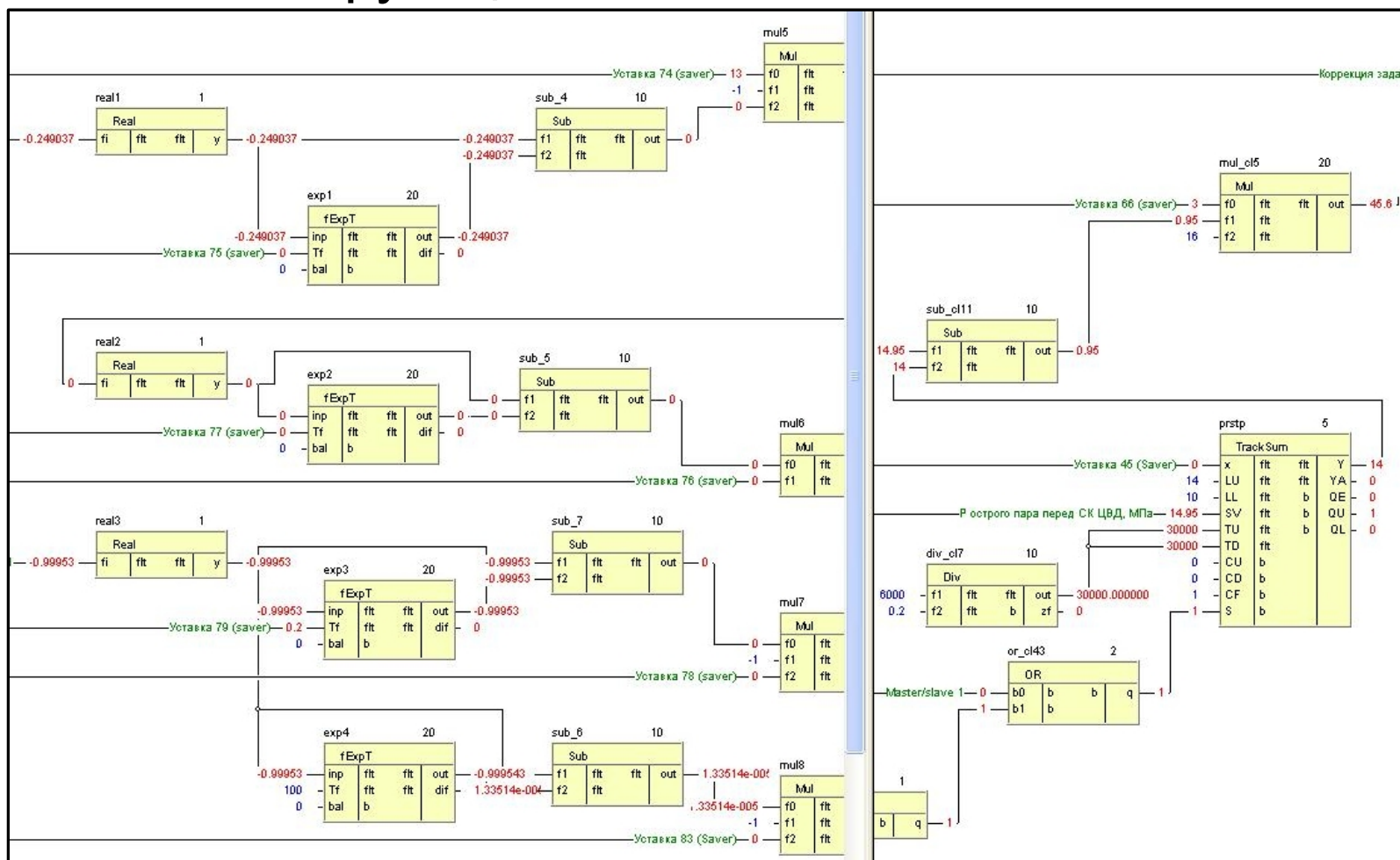


выполнение



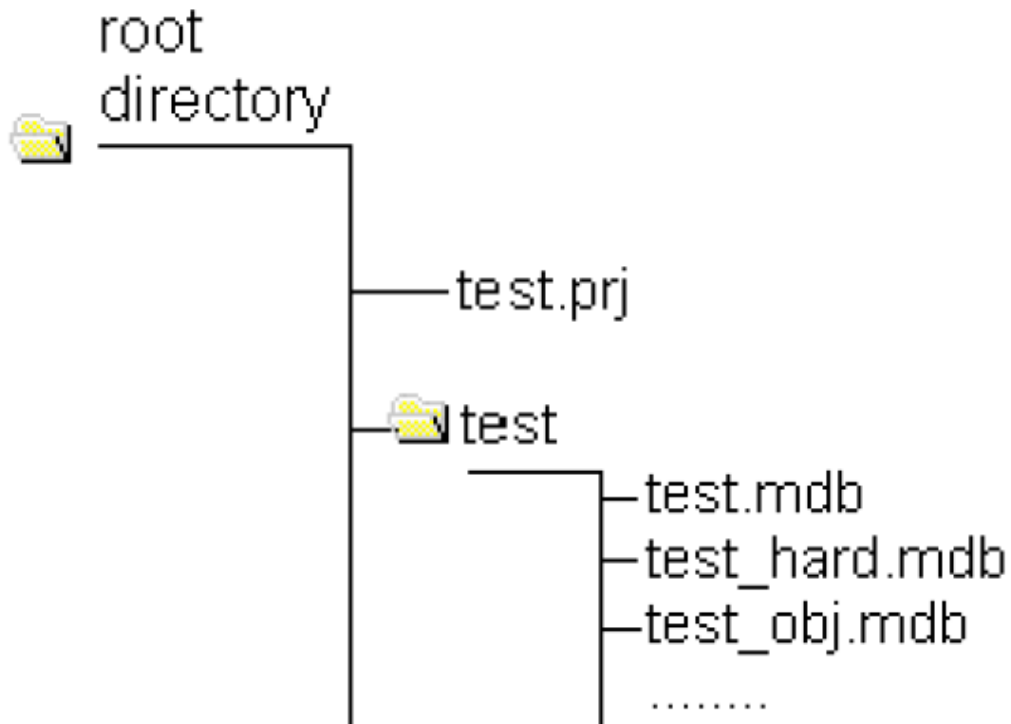
Этапы разработки

5. Отладка. Текущие значения на входах/выходах функциональных блоков.





Проект ПО в Полигоне



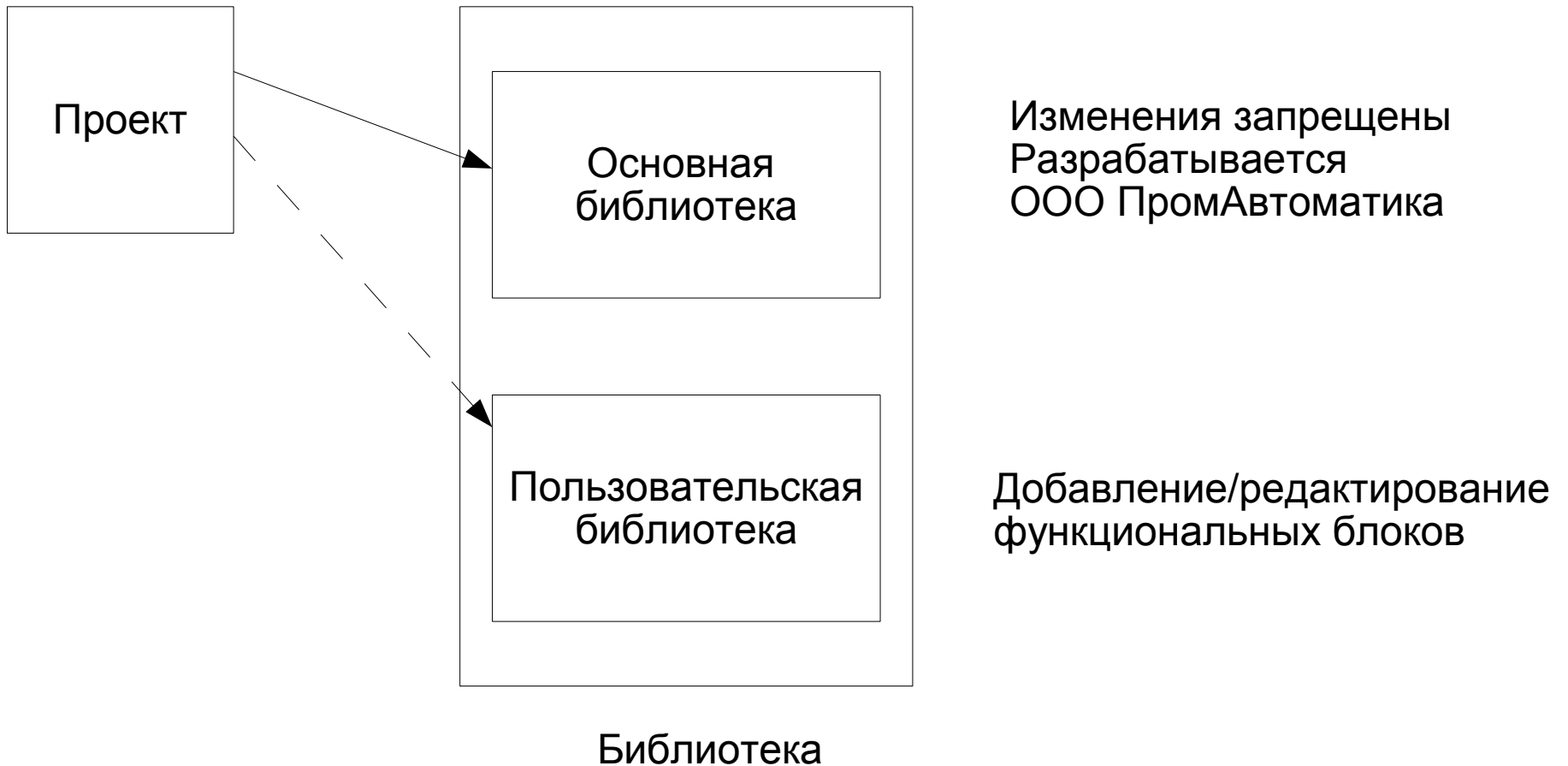
- Программная часть
- Аппаратная часть
- Объект



Библиотека функциональных блоков



Структура библиотеки





Основная библиотека

- Арифметические
 - Арифметические функции
 - сравнения
- Логические
 - Логические операции
 - Условные операции
 - Детекторы фронтов
- Переключатели, реле и мультиплексоры



Основная библиотека

- Генераторы и таймеры
 - Задержки
 - Генераторы импульсов/сигналов
- Триггеры
 - R
 - S
 - D
 - Шмидта
- Тригонометрические функции



Основная библиотека

- Регуляторы
 - Фильтры
 - Ограничитель, мертвая зона
 - ПИД, ПИ, Д
- Системные
 - Драйвера устройств
 - Интерфейсы связи
 - Интерфейсы с панелями оператора



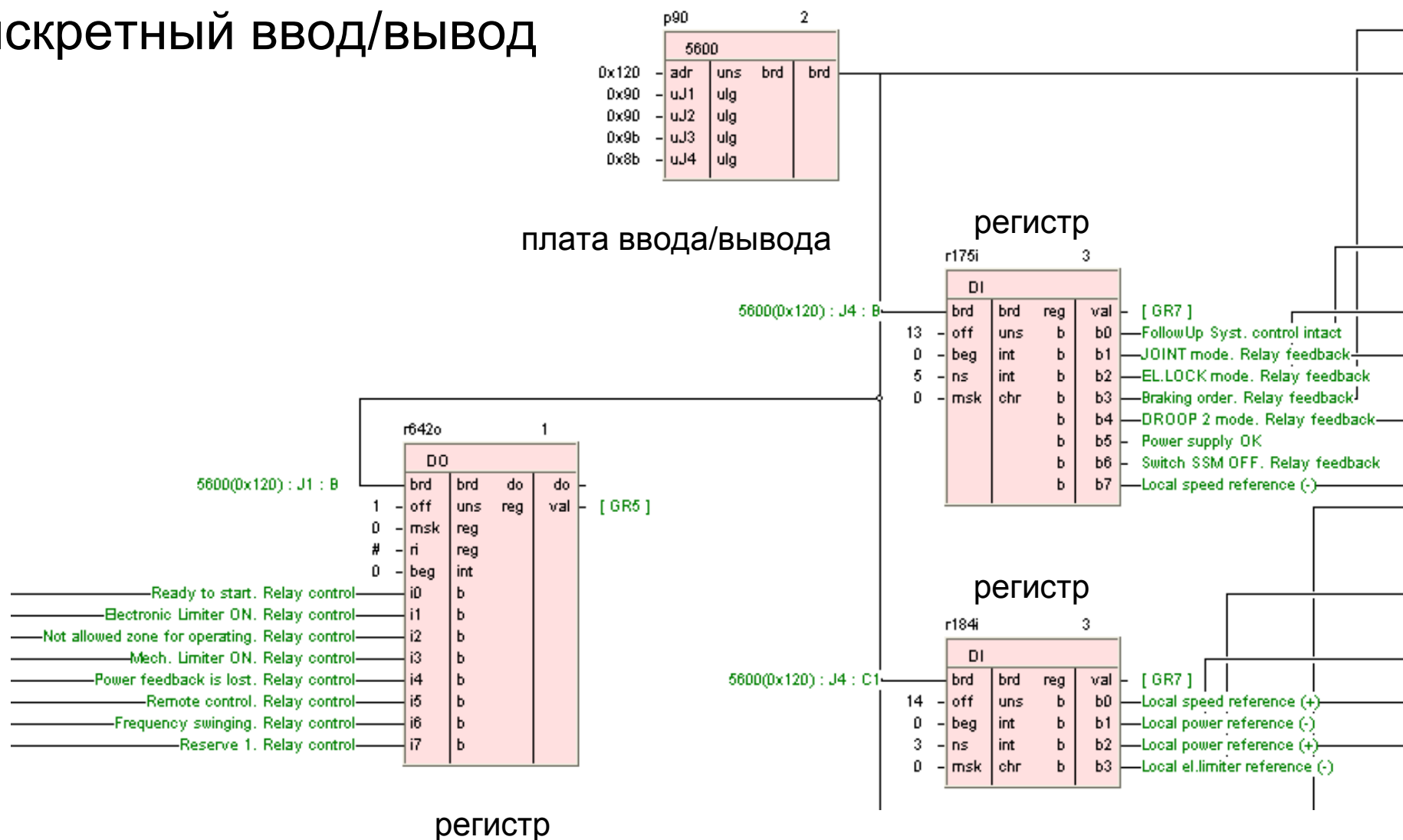
Драйвера плат ввода/вывода

- Драйвер устройства = набор функциональных блоков с фиксированными связями между ними: плата ввода/вывода, дискретные регистры, аналоговые каналы
- Входы оборудования = выходы функционального блока
- Выходы оборудования = входы функционального блока
- Фиксированные связи и параметры
- Место работы **Входы**



Пример драйвера платы ввода/вывода

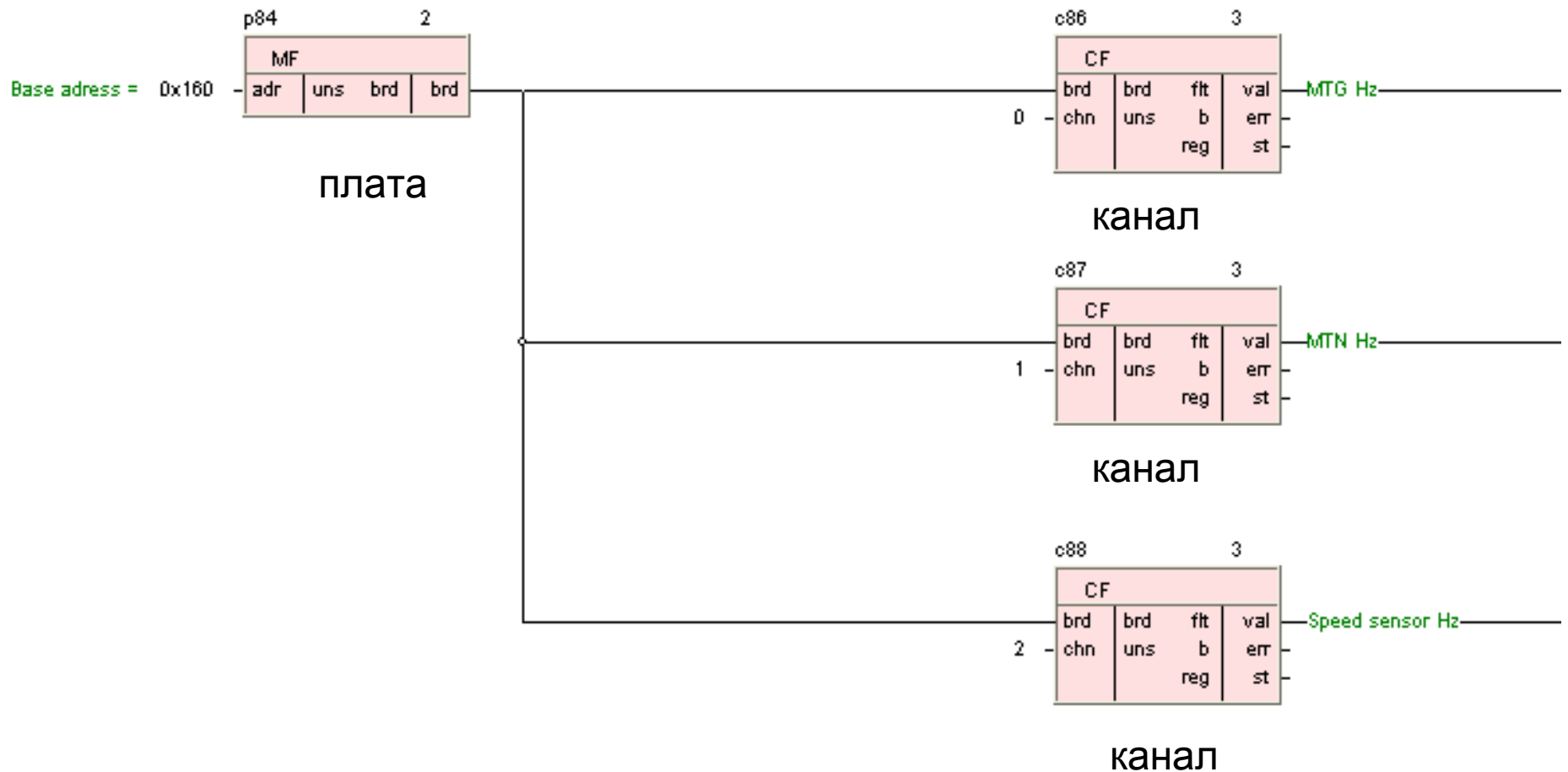
Дискретный ввод/вывод





Пример драйвера платы ввода/вывода

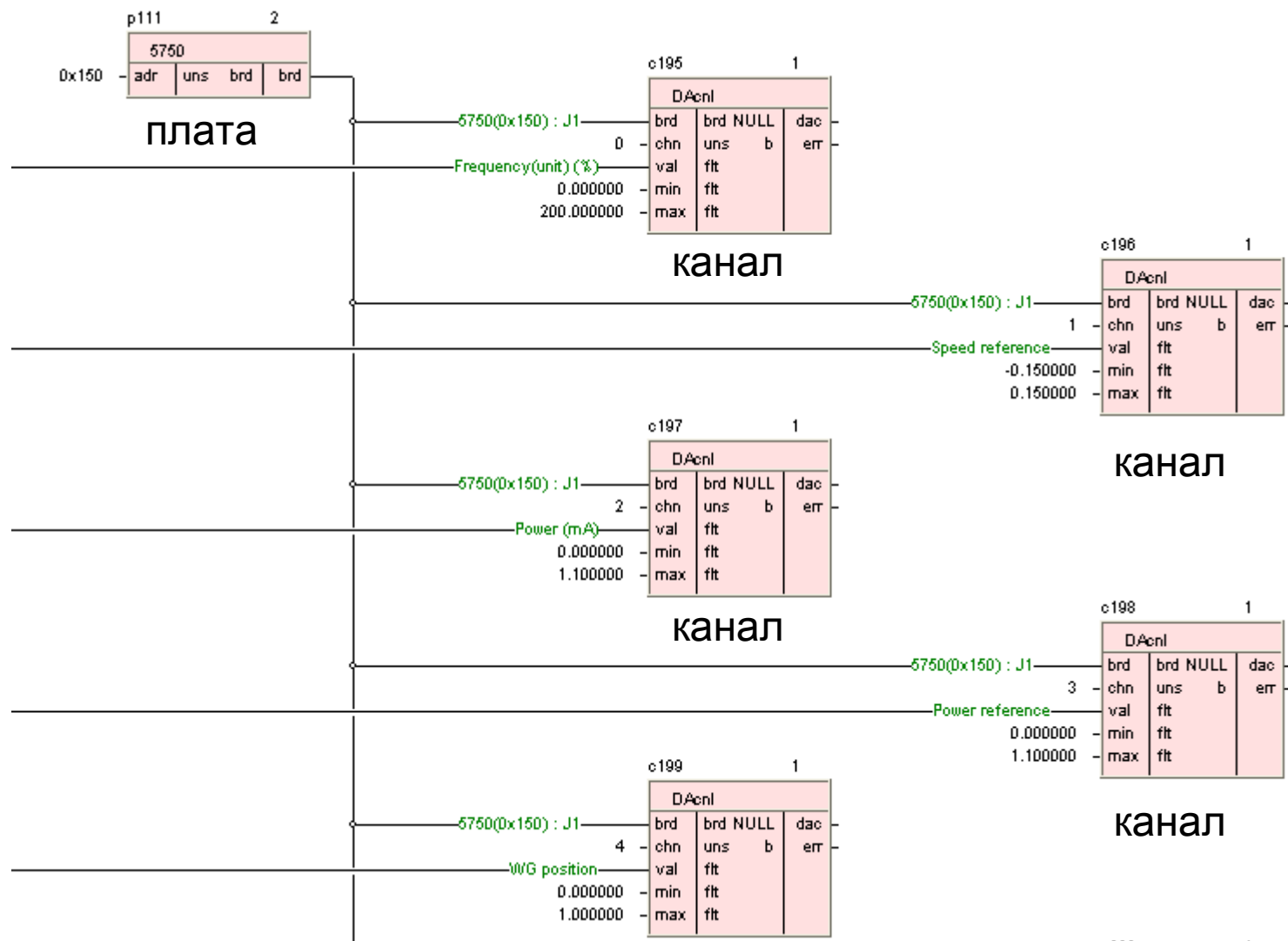
Плата измерения частоты





Пример драйвера платы ввода/вывода

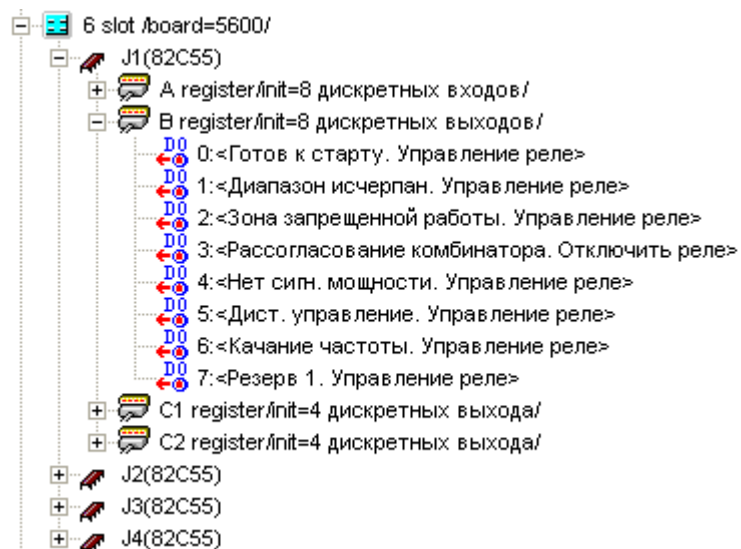
Аналоговый вывод



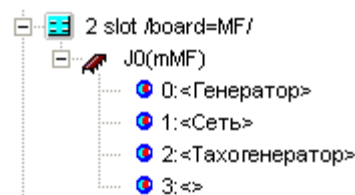


Конфигурация устройств ввода/вывода

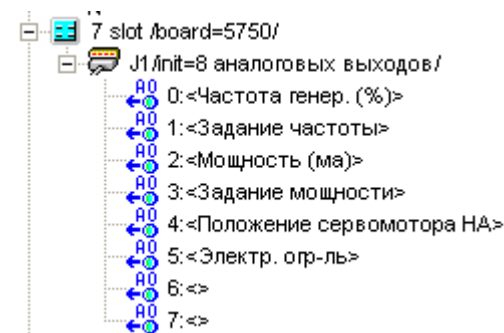
- Фиксированные правила запрещающие неправильную инициализацию



Дискретный ввод/вывод



Измерение частоты



Аналоговый вывод



Пример драйвера для Beckhoff

Project <C:\kapchag2\regul\regul_hard.mdb>

C1000+1100

COM1 adr. of shift=0

Связь со следящей

Сеть2 adr. of shift=0

TCP/IP server - протокол ПА

Kbus

KL2xy8

KL2xy8

KL2xy8

KL2xy8

KL1xy8

KL1xy8

0: <Резерв 2(GK9) О.С. реле>

1: <Резерв 3(GK10) О.С. реле>

2: <Регулятор исправен.О.С. реле (GK11)>

3: <Сбоя процессора нет (GK12) О.С. реле>

4: <Ошибка регулятора(GK13) О.С. реле>

5: <Режим СК.(GK14) О.С. реле>

6: <Режим Частота(GK15) О.С. реле>

7: <Режим Мощность(GK16) О.С. реле>

KL1xy8

KL1xy8

KL1xy8

KL1xy8

KL1xy8

KL1xy8

KL3xx8

KL4xy8

KL4xy8

PC-104

MF

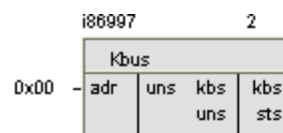
J0(mMF)

0: <Частота генератора>

1: <Частота сети>

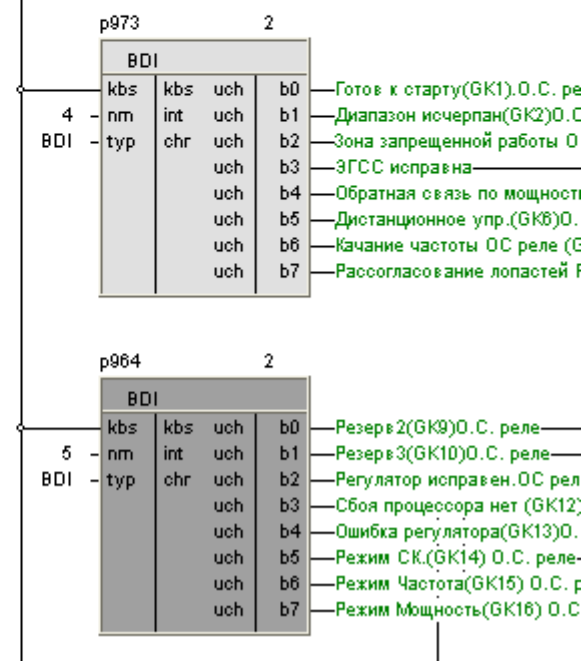
2: <Частота ТГ>

3: <>



шина

модуль дискретного ввода

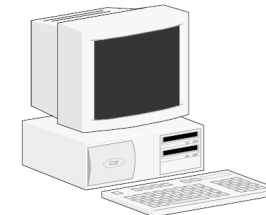
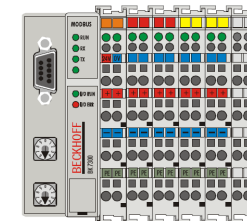
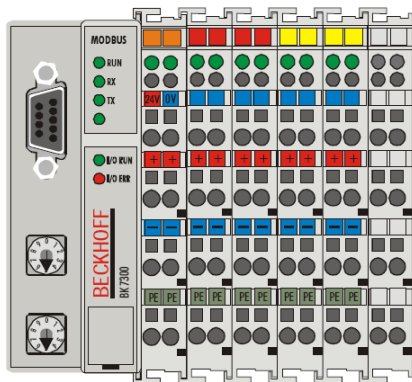




Интерфейсы связи

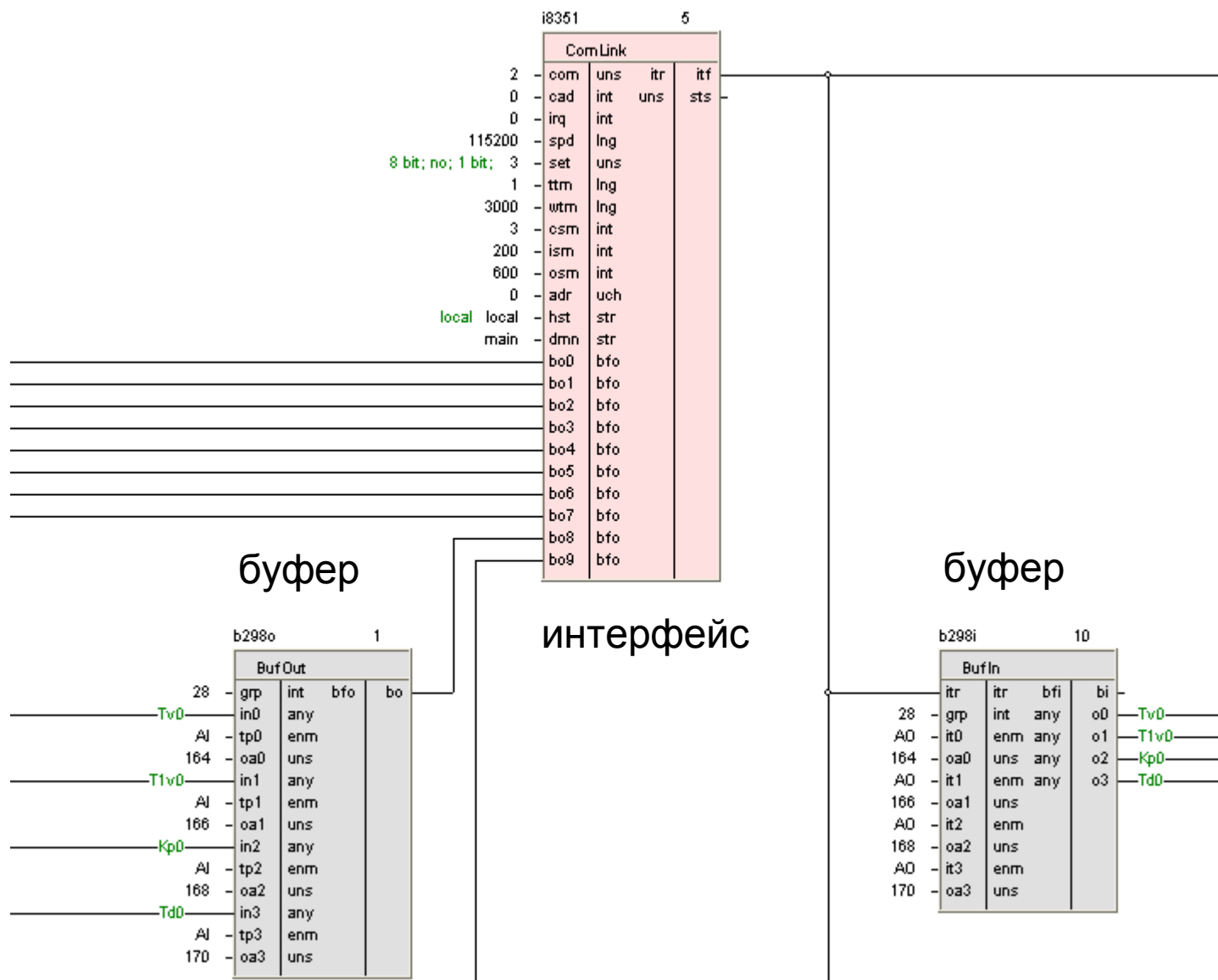
Место работы Интерфейсы

- Функциональный блок драйвера
- Несколько буферов ввода/вывода
- Буфер – набор переменных



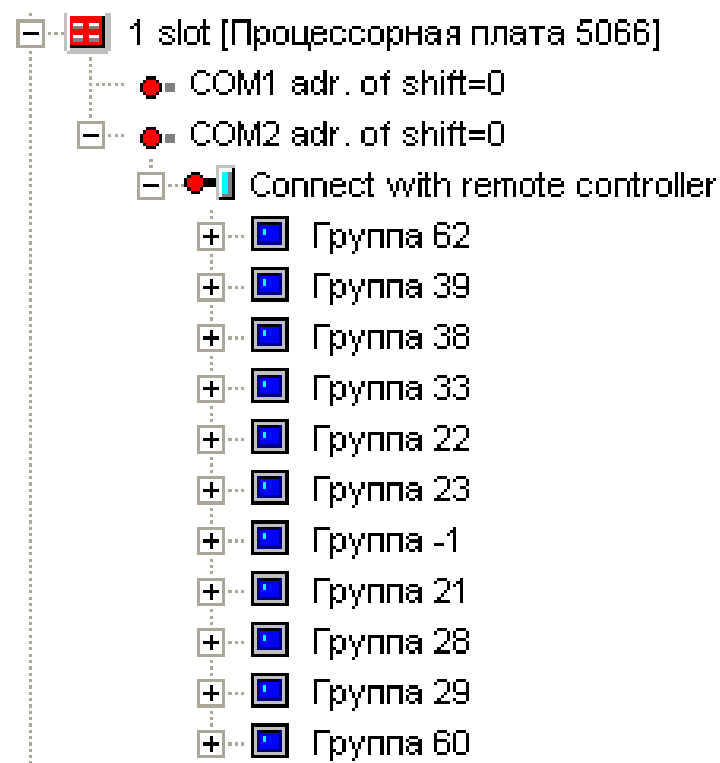


Пример интерфейса связи





Конфигурация интерфейса



- Параметры соединения
- Группа = 1 или 2 буфера
- Добавление новых групп и переменных



Буфер обмена

- DI/DO – 8-ми разрядный регистр
- AI/AO – вещественное число
- II/IO – 16-ти разрядный регистр или целое число

переменные

номер группы	значение	тип	адрес
	28	grp	int
—Tv0	in0	any	bfo
AI	tp0	enm	bo
164	oa0	uns	
—T1v0	in1	any	
AI	tp1	enm	
166	oa1	uns	
—Kp0	in2	any	
AI	tp2	enm	
168	oa2	uns	
—Td0	in3	any	
AI	tp3	enm	
170	oa3	uns	

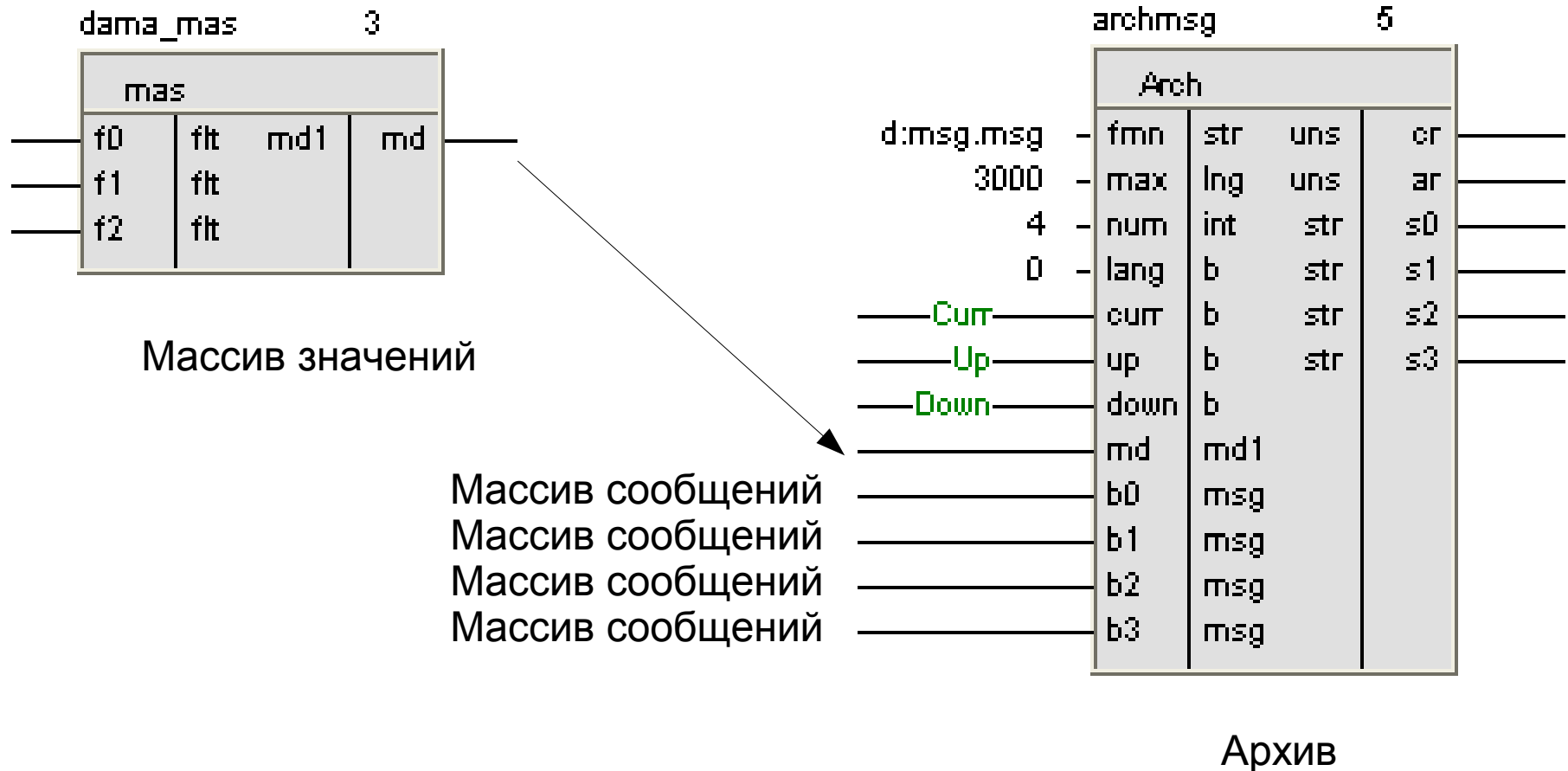


Архивация сообщений

- Обработка бинарных сообщений
- Хранение сообщений в файлах на диске (RAM или CMOS)
- Запись даты и времени
- Запись аналоговых/дискретных значений
- Пролистывание архива для просмотра сообщений

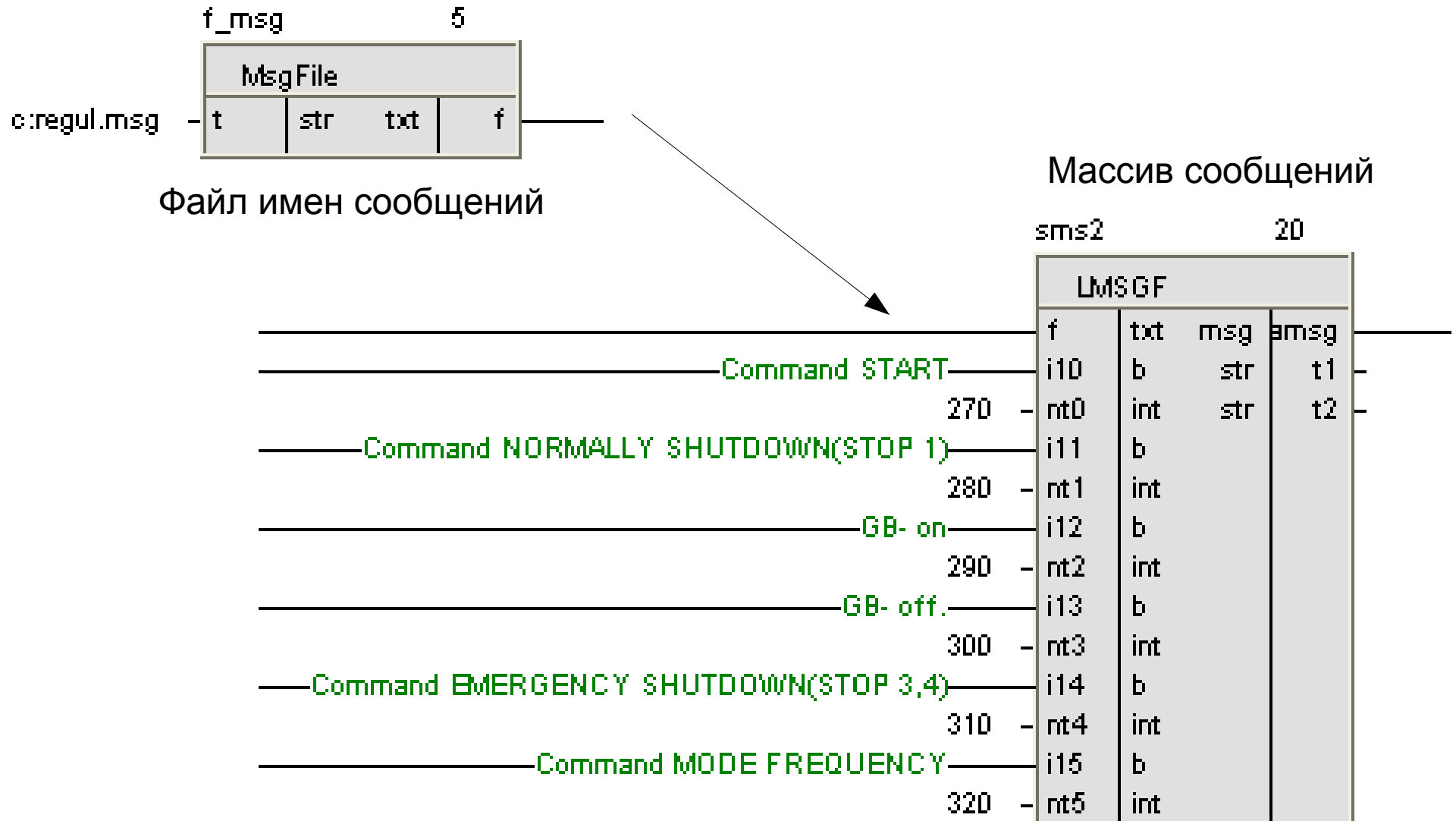


Архивация сообщений



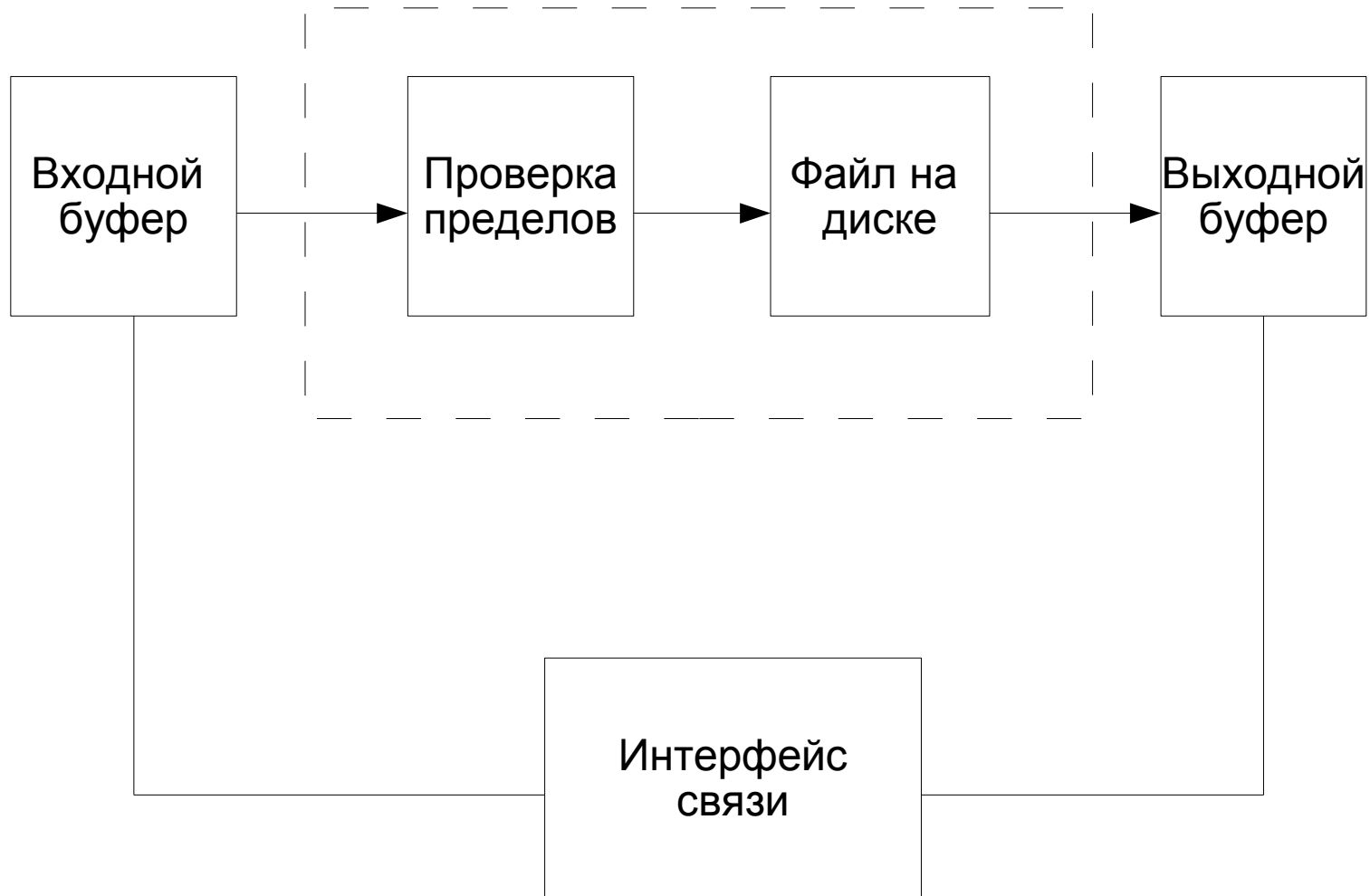


Archiving Messages





Передача/хранение параметров





Хранение параметров

LDW19					1
Lims					
0.10	min0	flt	flt	o0	Kp0
Kp0	val0	flt	flt	o1	Kp1
10.0	max0	flt	flt	o2	Td0
0.10	min1	flt	flt	o3	Td1
Kp1	val1	flt	flt	o4	Tv0
10.0	max1	flt	flt	o5	Tv1
1.00	min2	flt	flt	o6	T1v0
Td0	val2	flt	flt	o7	T1v1
60.0	max2	flt	flt	o8	Bp1
1.00	min3	flt	flt	o9	Bp2
Td1	val3	flt	flt	o10	
60.0	max3	flt	flt	o11	
0.02	min4	flt	flt	o12	

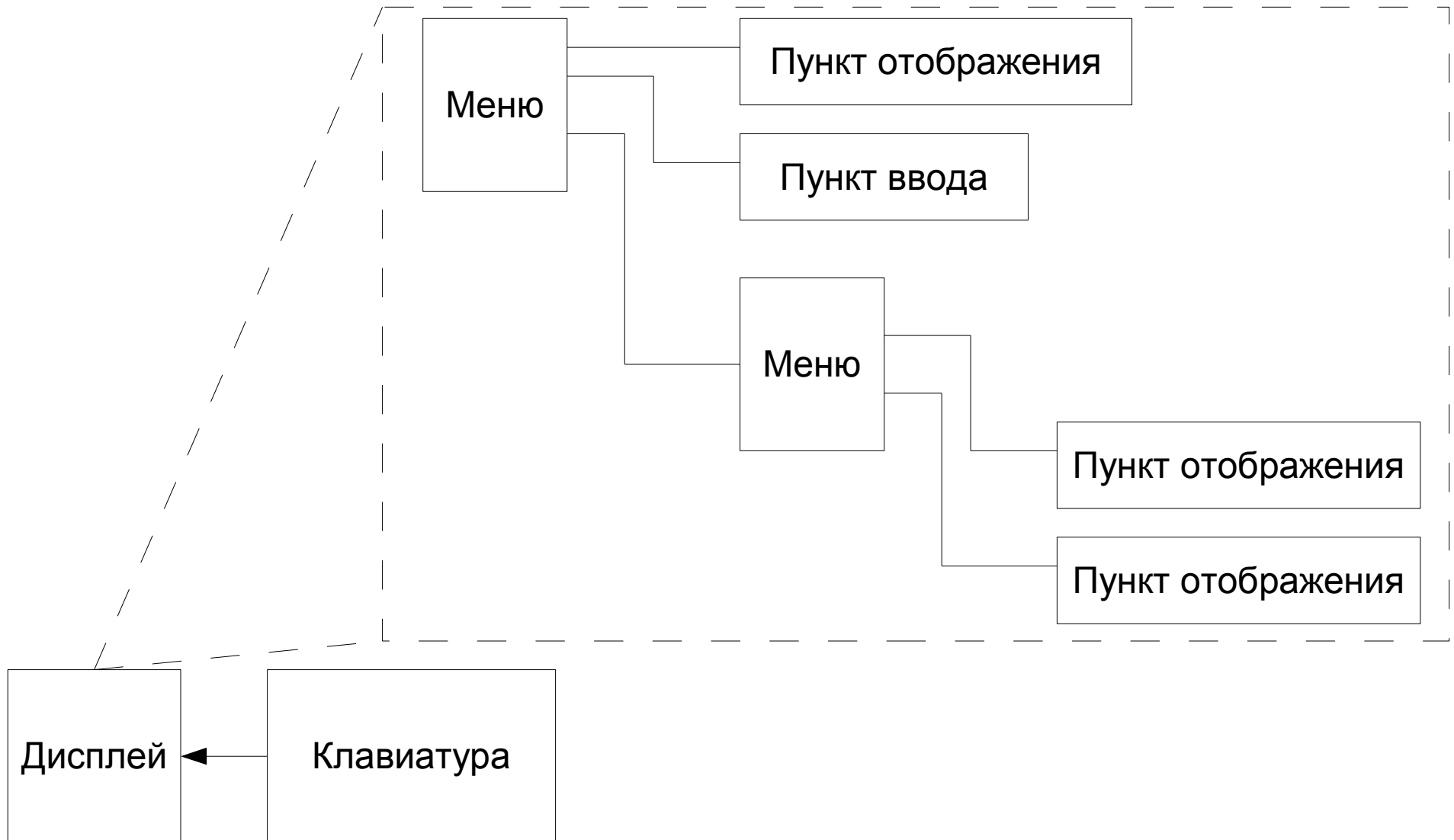
Проверка пределов

SDW19					1
Saver					
d:freg.par	fnm	str	any	o0	Kp0
Kp0	in0	any	any	o1	Kp1
AJ	typ0	enm	any	o2	Td0
Kp_xx = 1.65	ini0	flt	any	o3	Td1
Kp1	in1	any	any	o4	Tv0
AJ	typ1	enm	any	o5	Tv1
Kp_net = 1.65	ini1	flt	any	o6	T1v0
Td0	in2	any	any	o7	T1v1
AJ	typ2	enm	any	o8	DW19201
Td_xx = 11	ini2	flt	any	o9	DW19202
Td1	in3	any	any	o10	DW19203
AJ	typ3	enm	any	o11	DW19204
Td_net = 11	ini3	flt	any	o12	DW19205
Tv0	in4	any	any	o13	DW19206
AJ	typ4	enm	any	o14	DW19207
Tv_xx = 4	ini4	flt	any	o15	DW19208

Файл на диске

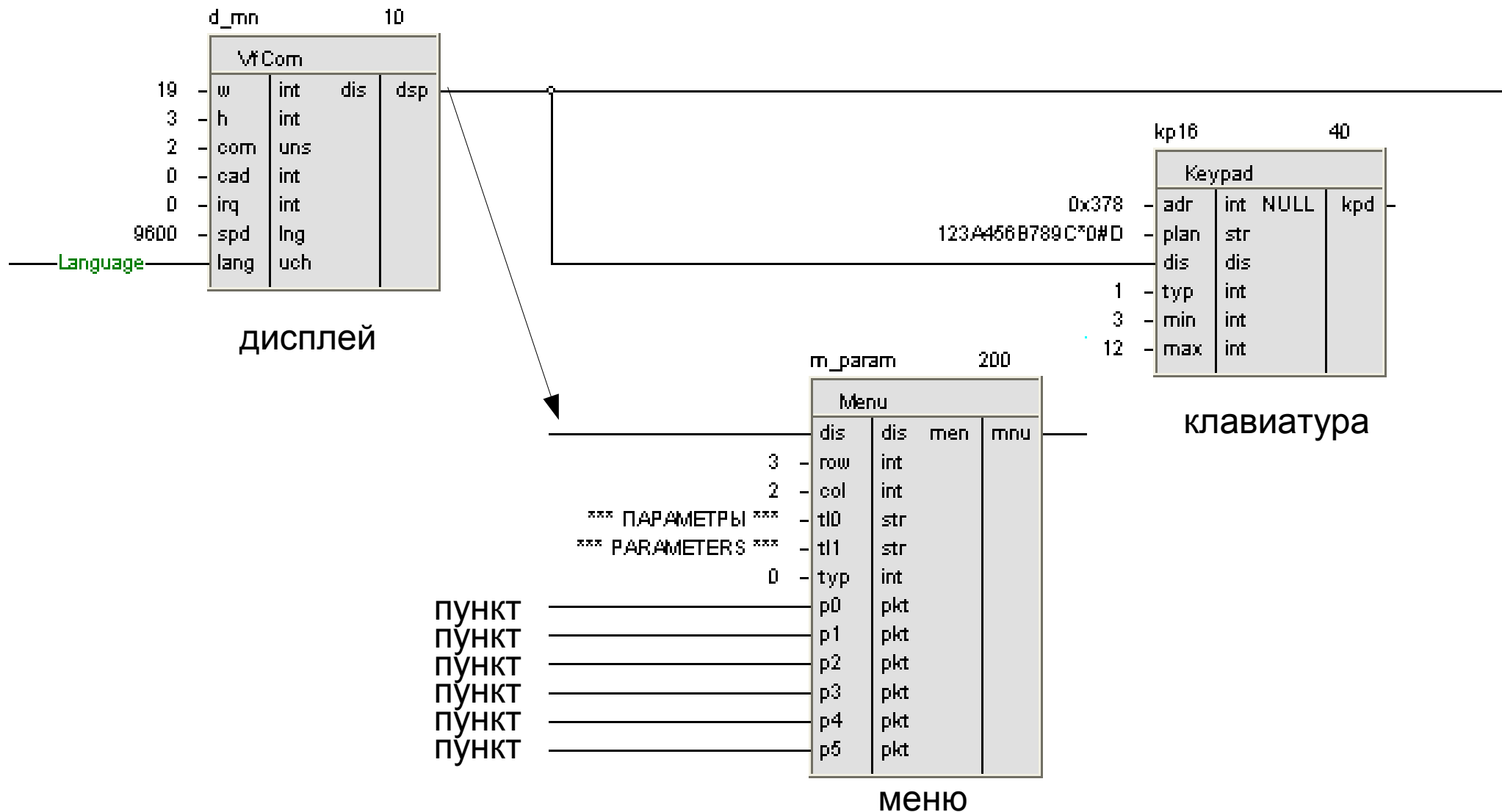


Система меню для LCD дисплеев



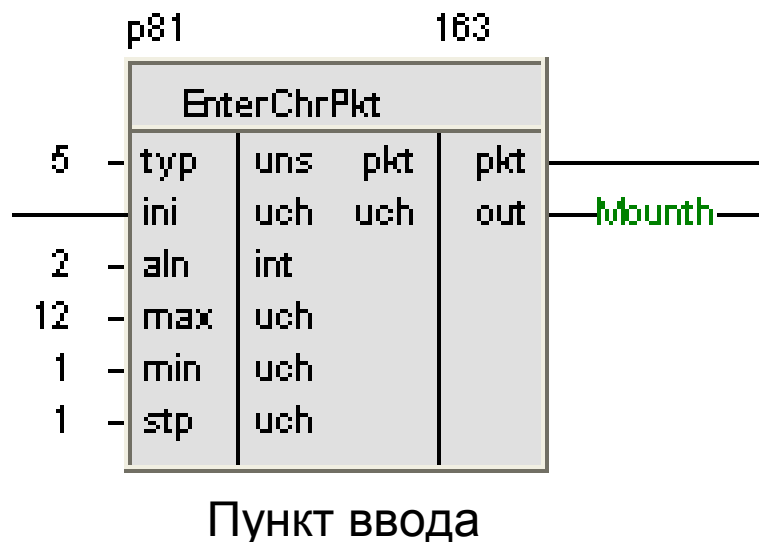
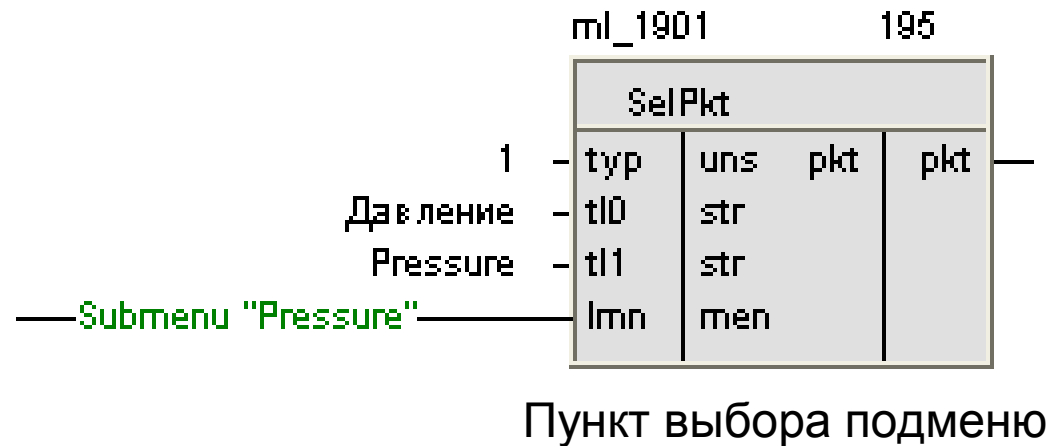


Система меню для LCD дисплеев





Menu System for LCD Displays





Пользовательская библиотека

- Требуются навыки программирования в C++

