

Инструментальные средства технологического

программирования. ISaGRAF
(стандарт IEC-1131)

Языки Structured Text и Instruction List

ST & IL

Structured Text Instruction List

```
if (level <= level_max)
then
  out_valve := false;
  memory_vlv :=
    (vlv23+dbh18)/2;
else
  alarm_level := true;
  out_valve := false;
end_if;
```

```
start_cmd: LD  bi101
            ADD 10
mul_ope: MUL( interm_bcmd
            SUB bo100
            )
            ST  bcmd
            GT  top_level
            JMPNC mul_ope
```

- Инструкции **ST** объединяют идентификаторы и разделители
- **Идентификатор** может быть именем объекта или ключевым словом **ST**
- Неактивный **разделитель** отделяет идентификаторы и другие символы (пробел, табулятор, конец строки, точка с запятой)
- Инструкции отделяются точками с запятой
- Фигурные скобки могут изолировать печатаемое выражение
- Комментарии заключаются в **(* и *)** и не могут быть вложенными

Удобочитаемость ST

- Выбирайте разумные имена для переменных
- Отделяйте действия с различными атрибутами
- Не пишите больше одной инструкции в одной строке
- Вставляйте знаки пробела
- Вводите понятные комментарии

Инструкции ST

- `:=` *назначение*
- `IF / THEN / ELSE /`
`ELSIF / END_IF;` *двоичный выбор*
- `CASE / OF / ELSE / END_CASE;` *выбор*
- `WHILE / REPEAT` (to be used with special care)
- `END_WHILE / END_REPEAT;` *итерации*
- `FOR / TO / BY / DO / END_FOR;` *индексные итерации*
- `RETURN;` *завершение программы*
- `EXIT;` *завершение оператора цикла*

Булевы операторы ST

- Булевы операторы (уменьшение приоритета)
 - **NOT** *логическое отрицание*
 - **AND or &** *логическое AND*
 - **OR** *логическое OR*
 - **XOR** *логическое исключающее OR*
- Управление фронтом
 - **redge (,)** *проверка положительного фронта*
 - **fedge (,)** *проверка отрицательного фронта*
- Сравнения : = и <>

Аналоговые операторы ST



- Арифметические операторы
(в контексте integer (ANA) или REAL)
 - $+$ сложение
 - $-$ вычитание
 - $*$ умножение
 - $/$ деление
- Битовые функции
(только в контексте integer (ANA))
 - `not_mask (,)` побитовое дополнение
 - `and_mask(,)` побитовая маска AND
 - `or_mask (,)` побитовая маска OR
 - `xor_mask (,)` побитовая маска XOR
- Сравнения : $=$, $<>$, $>=$, $<=$, $<$, $>$

Таймерные операторы ST

- Арифметические операторы

+

сложение

-

вычитание

- Контроль активности

TSTART ()

значения)

старт переменной timer (от ее

TSTOP ()

значением)

стоп переменной timer (с ее

- Сравнения : >, <

Операторы сообщений ST

- Назначение

$\textcolor{red}{:=}$

прямое копирование

- Конкатенация

$\textcolor{red}{+}$

слияние двух сообщений

- Сравнения : $=$, $\textcolor{red}{<}>$, $\textcolor{red}{>=}$, $\textcolor{red}{<=}$, $\textcolor{red}{>}$, $\textcolor{red}{<}$ (алфавитный порядок)

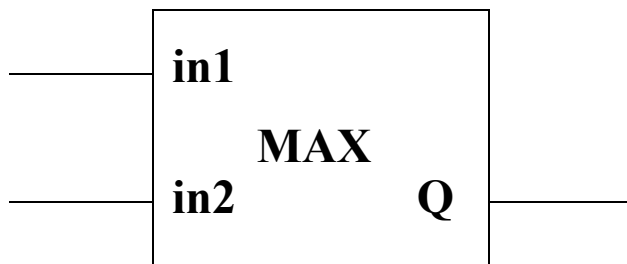
Функции преобразования типов ST



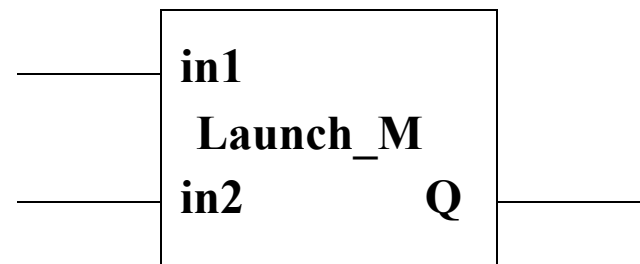
- **BOO** преобразование в boolean
- **ANA** преобразование в integer (ANA)
- **REAL** преобразование в real
- **TMR** преобразование в timer
- **MSG** преобразование в message

Вызов функций ST

- $\text{MaxVar} := \text{Max}(\text{ana1}, \text{ana2});$
- $\text{OKStart} := \text{launch_M}(\text{speed}, \text{time_activation});$
- $\text{Dummy} := \text{launch_M}(\text{speed}, \text{time_activation});$



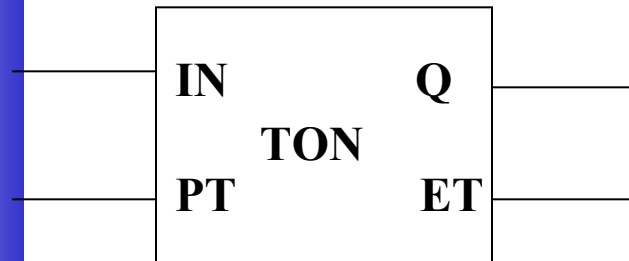
Функция ISaGRAF



Пользовательская функция

Вызов функционального блока ST

- Объявите экземпляр в словаре
- Выполняйте действия на **экземплярах** функционального блока



ISaGRAF F. Block

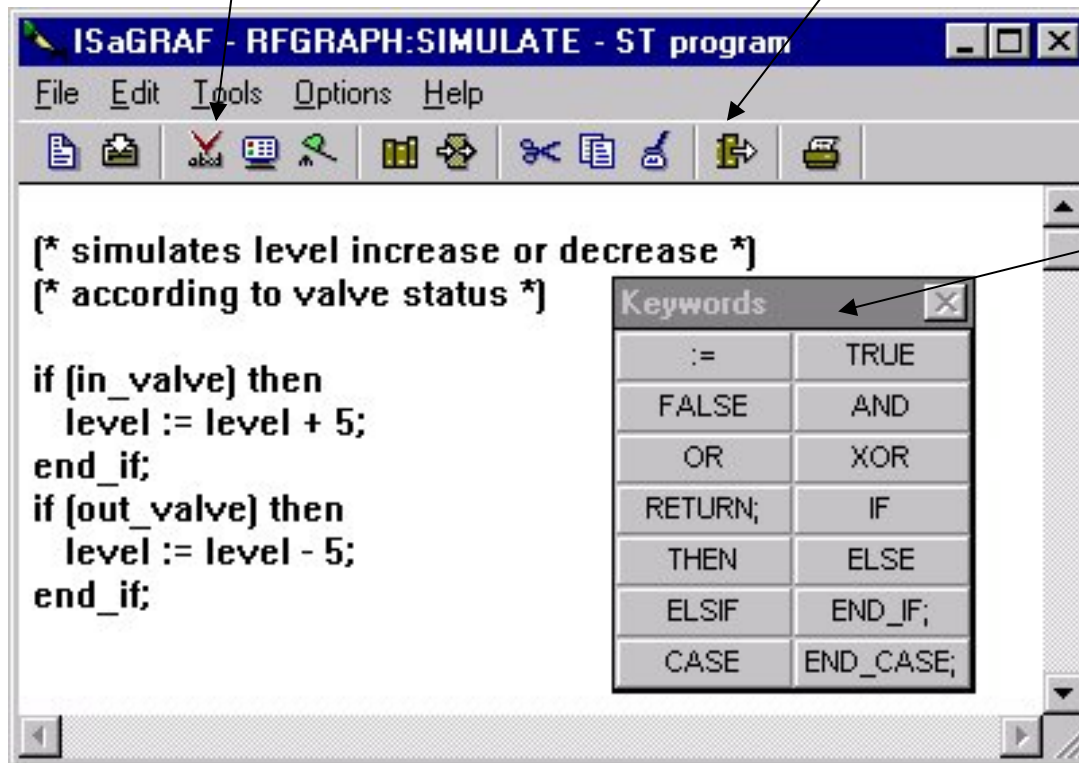
Код

- Активизирует экземпляр функционального блока
- Получает возвращаемое значение функционального блока

Редактор ST ISaGRAF

Проверка синтаксиса

Иконка для ввода
переменных из словаря



Панель
часто используемых
ключевых слов

Резюме по ST

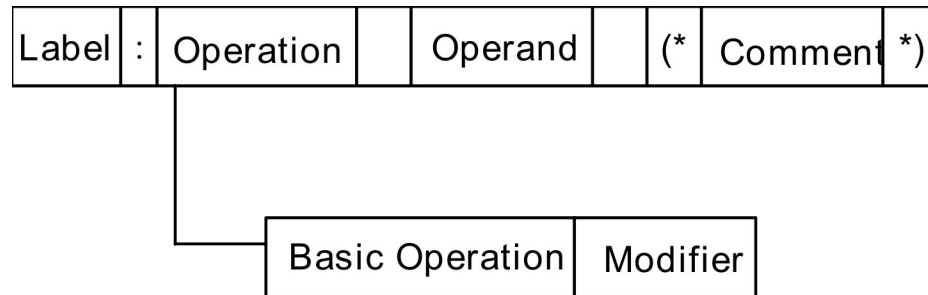
- Текстовый язык высокого уровня
- Используется в главных программах
- Используется в **SFC** действиях или переходах или **FC** действиях и решениях
- Легко сопровождается **если**
 - Имена переменных понятны
 - Представлены комментарии
 - Код хорошо структурирован

Язык Instruction List



- **Синтаксис IL**
- **Редактор IL**

Формат строки инструкций



- Необязательная метка (никаких пробелов между меткой и :)
- Необязательный комментарий (последний компонент строки)
- Необязательный модификатор операции
- Никаких пробелов между именем операции и модификатором

Команды IL

- LD / ST *Назначение*
- AND / & / OR / XOR / S / R *Булевы операторы*
- ADD / SUB / MUL / DIV *Операторы Analog или timer*
- GT / GE / EQ / LE / LT *Сравнения*
- JMP / RET *Операторы выполнения*
- CAL / <sub_prg_name> *Вызовы*

- Каждая команда работает на значении **аккумулятора**
- **Аккумулятор** изменяется каждой командой
- Операндом может быть переменная, константное выражение или имя метки (в той же самой программе)
- Команда IL

Operation Operand

эквивалентна

Accu := Accu <operation> operand;

Модификаторы операторов IL

- **N** *Логическое отрицание оператора*

ANDN IX12 в ST означает
Accu := Accu AND NOT (IX12);

- **(** *Отсроченная команда*
- **C** *Условная команда*

JMPC LABEL1 означает
If **Accu** then **JMP LABEL1**

Список модификаторов

- **N** LD, ST, AND, &, OR, XOR, JMP, RET, CAL
- **(** AND, &, OR, XOR, ADD, SUB, MUL, DIV, GT, GE, EQ, LE, LT, NE
- **C** JMP, RET, CAL

Отсроченные команды

- Используются, чтобы **отложить** выполнение команды
- (*Модификатор* указывает, что команда должна быть отложена
-) *Оператор* Выполняет отсроченную команду
- **AND(IX12
OR IX35
)**
выполняется как

Ассу := Ассу AND (IX12 OR IX35);

Редактор IL ISaGRAF

Проверка синтаксиса

Иконка для ввода
переменных из словаря



Панель
часто используемых
ключевых слов

Резюме по IL

- Язык низкого уровня
- Точка входа в ISaGRAF
- Связан со скрытым аккумулятором
- Данный список команд
- Нормальный текстовый редактор