

1. Преподаватель: Плужнова Татьяна Сергеевна

(a) Почта: pluzhnova.t@gmail.com

(b) Телефон: +79296468157

2. В конце семестра зачёт

Лекция 1 (19.02.19)

1. Функции модуля ПЛК

(a) Сбор информации из модулей ввода в память

(b) Передача данных из памяти в модули вывода

(c) Выдача меток часов реального времени

(d) Обмен информацией с промышленной сетью

(e) Исполнение пользовательской циклопрограммы

2. Языки программирования стандарта IEC 61131-3

(a) Текстовые

i. ST – Structured Text

ii. IL – Instruction List

(b) Графические

i. FBD – Function Block Diagram

ii. LD – Ladder Diagram

iii. SFC – Sequential Flow Chart

0.1 Язык ST

1. Типы данных

(a) Целочисленные

i. SINT (char)

- ii. USINT (unsigned char)
- iii. INT
- iv. UINT
- v. DINT (long)
- vi. UDINT (unsigned long)
- vii. LINT (long int 64bit)
- viii. ULINT

(b) Действительные

- i. REAL (float)
- ii. LREAL (double)

2. Временные типы данных

- (a) TIME
- (b) DATE
- (c) TIME_OF_DAY
- (d) DATE_AND_TIME

3. Производные типы данных

- (a) STRUCT
- (b) ARRAY
- (c) Перечисления
- (d) Поддиапазонные типы данных

4. Преобразование типов (строгая типизация)

- (a) $X := \text{REAL_TO_INT}(Y);$
- (b) $X := \text{<TYPE1>_TO_<TYPE2>}(Y);$

5. Операторы

- (a) Присваивание $:=$
- (b) Остаток: MOD

(с) Модуль: ABS

(d) Побитный сдвиг: SHL(in, n), SHR(in, n)

(e) MAX(x1, x2), MIN(x1, x2)

6. Ветвление

```
IF [expr] THEN
    <statement>
ELSIF [expr] THEN
    <statement>
ELSE
    <statement>
END_IF
```

7. Switch

```
CASE OF
    <expr>: <statement>;
ELSE
    <statement>;
END_CASE
```

8. Циклы

```
FOR <var> := <value> TO <final value> BY <incr> DO
    <statement>
END_FOR;
```

```
WHILE <expr> DO
    <statement>;
END_WHILE
```

```
REPEAT
```

```
<statement>;  
UNTIL <expr>  
END_REPEAT;
```

9. Функции

```
function <имя> : <тип>  
    <аргументы>  
    <предложения>  
end_function
```

0.2 Beremiz

1. Состоит из

(a) САПР

- i. Кросс-компилятор
- ii. Транслятор MatIEC (в код C)
- iii. Компилятор (в .so – динамическую библиотеку)

(b) Среда исполнения RT

Лекция 2 (5.03.19)

0.3 Язык FBD

1. FBD – графический язык, применяется для построения комплексных процедур, состоящих из различных библиотечных блоков – арифметических, тригонометрических, регуляторов, мультиплексоров и т.д. Наиболее подходит для управления непрерывными процессами и регулирования.

2. Объекты

- (a) элементарные функции и элементарные функциональные блоки (ФБ)
- (b) функции и ФБ пользователя

0.4 Язык LD

- LD – графическая интерпретация процесса разработки релейно-контакторных схем управления. Наиболее удобен для программирования небольших задач дискретной логики.

LD	ЕСКД	Обозначение
		Нормально разомкнутый контакт
		Нормально замкнутый контакт
		Обмотка реле

Рис. 1: Объекты

1. Обмотки:

- Прямая (**O**)
- Инверсная(**Ø**)
- Импульсная
 - по переднему фронту (**P**)
 - по заднему фронту (**N**)
- Включающая (**S**) и выключающая (**R**)
- Прямой контакт (-||-)
 - $out1 := AND(in1, in2);$
- Инверсный контакт (-|/|-)
 - $out1 := AND(NOT(in1), NOT(in2));$
- Импульсный контакт
 - по переднему фронту (-|p|-)
 - $temp := AND(int2, NOT(in2_prev))$
 - $in2_prev$ – предыдущее значение
 - по заднему фронту (-|n|-)
 - $temp := AND(NOT(int2), in2_prev)$
 - $in2_prev$ – предыдущее значение

0.5 Протокол ModBus

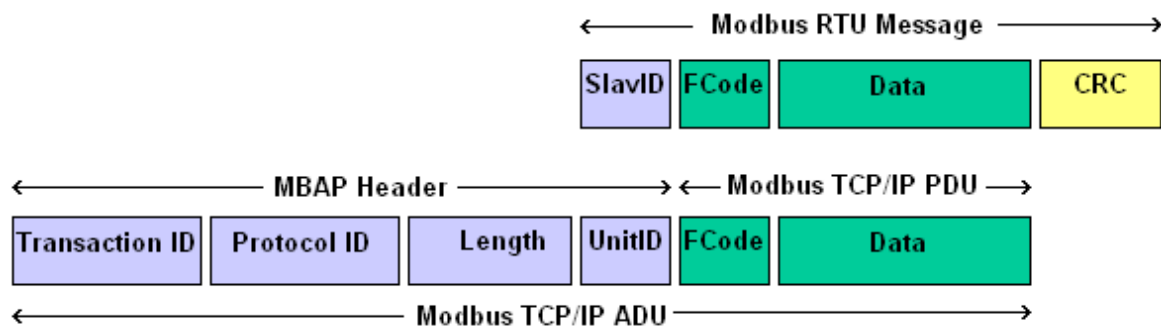


Рис. 2: Пакеты ModBus

1. Реализации

(a) RS-485 / RS-422 / RS-232

i. RTU

ii. ASCII

(b) TCP / IP

i. TCP

2. Поля RTU

(a) SlaveID (1 - 247)

(b) Func Code

(c) Data

(d) CRC-16

3. В ASCII вместо CRC-16 идёт конструкция '\n\r'

0.6 Язык SFC

1. SFC – графический язык, который позволяет описать хронологическую последовательность различных действий в программе. Действия связываются с шагами (этапами), а последовательность определяется условиями переходов между ними.

2. Типы элементов SFC-программы:

(a) Шаги (steps)

(b) Переходы (transitions)

(c) ... Данные типы могут включать в себя элементы из других языков.

3. Предполагается, что после шага всегда идёт переход или ветвление.

0.7 Язык IL

1. IL используется в случаях, когда требуется поличить оптимизированный код для реализации критических секций программы, а также для решения небольших задач с малым количеством разветвлений алгоритма.

2. Пример: $y = Ax + B$

```
LD X
MUL A
ADD B
ST Y
```