

## ЛАБОРАТОРНАЯ РАБОТА №4

### Разработка программы для ПЛК на языке ST стандарта МЭК 61131.3

#### Теоретическая часть

#### ИЗУЧЕНИЕ УСТРОЙСТВА УДАЛЕННОЙ ТЕЛЕМЕТРИИ И УПРАВЛЕНИЯ Vida88

Цель работы: изучить, и ознакомиться с устройством Vida 88, спроектировать систему сбора данных и управления на базе устройства Vida 88 в соответствии со своим вариантом.

#### Описание задания:

- 1) ознакомиться с технической документацией;
- 2) изучить, выписать основные характеристики устройства (число портов ввода/вывода, тип сигнала аналоговый/цифровой);
- 3) ознакомиться и сформулировать свойства и задачи автоматизации;
- 4) описать используемые в работе порты ввода/вывода, типы сигналов;
- 5) построить блок-схему с указанием используемых портов;
- 6) выделить и расписать периодические и аperiodические процессы;
- 7) построить временную диаграмму (учитывая конкретные технические параметры устройства);
- 8) в ходе проектирование данная система должна реализовать следующие функции:
  - сбор данных;
  - обработка данных;
  - формирование управляющего воздействия;
  - функционирование в реальном времени.

#### **Теоретические сведения.**

#### **Техническое описание устройства**

#### **Назначение Vida88**

Микропроцессорное устройство Vida88 представляет собой уникальную комбинацию мощного PLC(программируемый логический контроллер) и телефона GSM, объединённого в одном простом программируемом устройстве. Устройство обеспечивает:

- дружественный интерфейс для дистанционного мониторинга и управления;
- наблюдения и регистрации событий.

Vida88 позволяет разработчику создавать приложения, объединяющие в себе возможности управления, контроля и регистрации событий с

расширенными методами коммуникации. К расширенным методам коммуникации относятся:

- голосовой ответ (удалённое взаимодействие с использованием полнодуплексного радиоканала связи (переговорные устройство типа рации));

- сигнал тревоги – посылка сообщений от модуля в виде SMS (как протокольного блока данных PDU) или непосредственная передача данных прямо приложениям или от приложений ОС Windows.

#### Состав устройства:

- Лицевая панель;
- Печатная плата:
  1. аналоговые входы/выходы;
  2. цифровые входы/выходы;
  3. цифровые выходы;
  4. порт RS-232 для подключения ПЭВМ;
  5. жидкокристаллический дисплей;
  6. карт-ридер SIM;
  7. Светоиндикаторы;
  8. разъем DISP1;
- Антенна.

Все разъёмы для подключения к внешнему оборудованию (кроме антенны GSM) и порта для программирования выполнены в виде креплений с зажимным винтом. Разъёмы изображены на рисунке 1.1. Каждый разъём пронумерован. Номера можно увидеть на верхней крышке пластикового корпуса устройства.

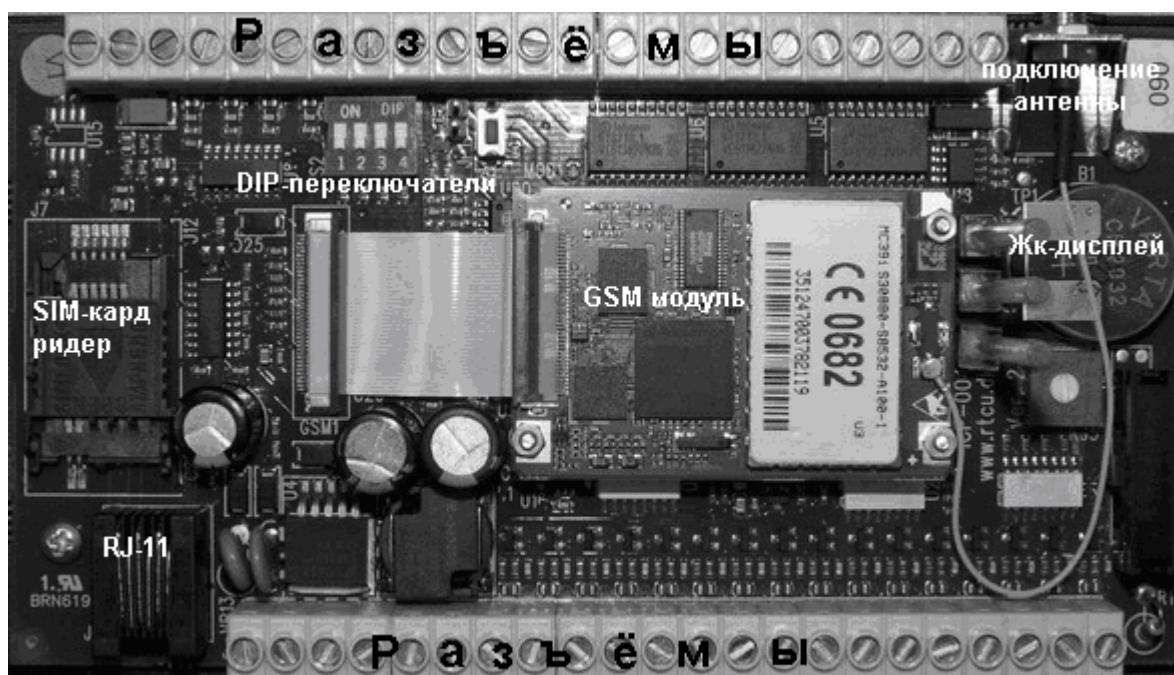


Рис. 1.1 Печатная плата

### Параметры и характеристики:

- 1)  $U_{\text{пит. пост. тока}} = 8-36 \text{ В}$
- 2)  $U_{\text{аналог. пост.тока}} = 0-5 \text{ В}$
- 3) Гальваническая развязка на оптронах цифровых входов/выходов
- 4) ЖК-дисплей с постоянной подсветкой, 2 строки x 16 символов
- 5) Светоиндикаторы для отображения ошибок разного типа

### Подключение и настройка устройства

Подключить устройство к источнику питания (см. табл. 1.1)

Таблица 1.1

Схема подключения источника питания

Наименование	Номер разъема	Описание
GND	5	Отрицательный (-) полюс источника постоянного тока
DCin	6	Положительный (+) полюс источника постоянного тока
GND	7	Отрицательный (-) полюс источника постоянного тока

Примечание: Разъём для подключения источника питания находится слева в нижнем ряду разъёмов с винтовыми фиксаторами.

Подключить к аналоговым и цифровым входам/выходам (см. таблицы 1.2-1.5). Аналоговые входные сигналы должны находиться в диапазоне от 0 до 5 вольт при постоянном токе.

Таблица 1.2

Схема подключения к аналоговым входам

Наименование	Номер разъема	Описание
AGND	8	Земля для аналоговых сигналов
AI1	9	Аналоговый входной сигнал 1
AI2	10	Аналоговый входной сигнал 2
AI3	11	Аналоговый входной сигнал 3
AI4	12	Аналоговый входной сигнал 4
AGND	13	Земля для аналоговых сигналов

Аналоговые выходные сигналы обеспечивают выходное напряжение в диапазоне от 0 до 5 вольт при постоянном токе.

Таблица 1.3

Схема подключения к аналоговым выходам

Наименование	Номер разъема	Описание
AO1	30	Аналоговый выходной сигнал 1

AO2	31	Аналоговый выходной сигнал 2
AO3	32	Аналоговый выходной сигнал 3
AO4	33	Аналоговый выходной сигнал 4
AGND	34	Земля для аналоговых сигналов

Цифровые входы гальванически изолированы от схемы устройства с использованием оптронов, а также снабжены фильтрами низкой частоты и защитой от скачков напряжения.

Таблица 1.4  
Схема подключений к цифровым входам

Наименование	Номер разъема	Описание
DGND	14	Земля для цифровых входов
DI1	15	Цифровой вход номер 1
DI2	16	Цифровой вход номер 2
DI3	17	Цифровой вход номер 3
DI4	18	Цифровой вход номер 4
DI5	19	Цифровой вход номер 5
DI6	20	Цифровой вход номер 6
DI7	21	Цифровой вход номер 7
DI8	22	Цифровой вход номер 8
DI9	23	Цифровой вход номер 9
DI10	24	Цифровой вход номер 10
DI11	25	Цифровой вход номер 11
DI12	26	Цифровой вход номер 12

Цифровые выходы соединяют с помощью интерфейсного кабеля с внешними устройствами через аппаратные драйверы. Чтобы работать правильно, цифровые выходы должны иметь свой собственный источник питания с напряжением от 5 до 34 вольт при постоянном токе.

Таблица 1.5  
Схема подключения к цифровым выходам

Наименование	Номер разъема	Описание
DOSUPP	35	Питание цифрового выхода
DOSUPP	36	Питание цифрового выхода
DOSUPP	37	Питание цифрового выхода
DO1	38	Цифровой выход номер 1
DO2	39	Цифровой выход номер 2
DO3	40	Цифровой выход номер 3
DO4	41	Цифровой выход номер 4
DO5	42	Цифровой выход номер 5
DO6	43	Цифровой выход номер 6
DO7	44	Цифровой выход номер 7

DO8	45	Цифровой выход номер 8
DO9	46	Цифровой выход номер 9
DO10	47	Цифровой выход номер 10
DO11	48	Цифровой выход номер 11
DO12	49	Цифровой выход номер 12

### Порт для программирования (RS232)

Подключение к порту RS232 осуществляется с использованием разъёма RJ-11, который находится в левом нижнем углу печатной платы устройства на рисунке 1.2.

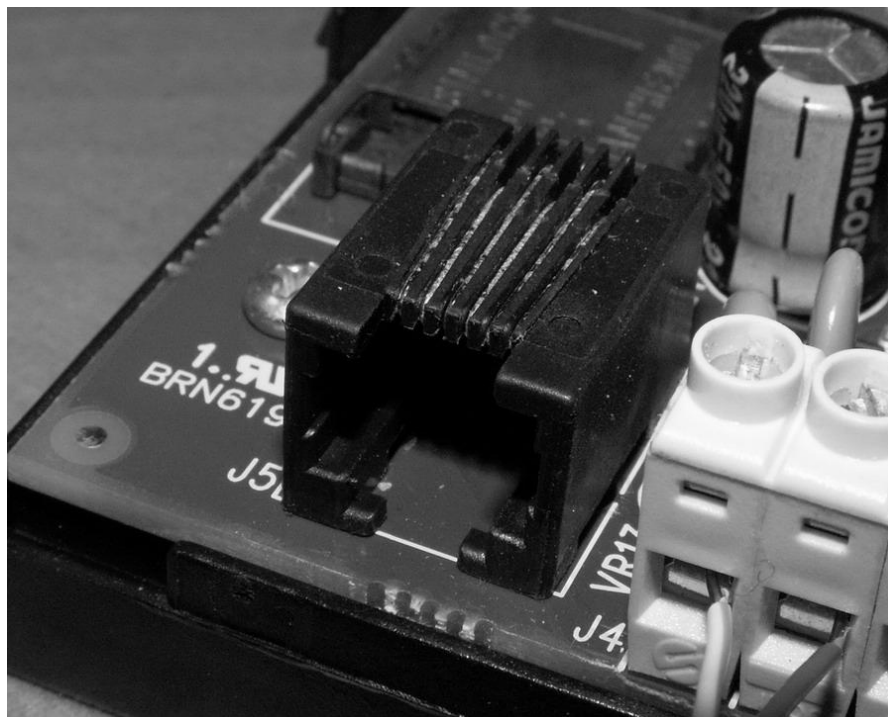


Рис. 1.2 Разъём RJ-11

Чтобы использовать разъём RJ-11 как порт для программирования, контакт 6 должен быть связан с заземлением. Для использования разъёма как последовательного порта, контакт RSDDET (контакт 6) нужно оставить без соединения.

Таблица 1.6  
Схема подключения к разъёму RJ-11

Наименование	Номер разъёма	Описание
VCC	1	Питание +5 вольт от RTCU (не подключать!)
TxD	2	RS232. Передача данных от Vida88
GND	3	«Земля»
RxD	4	RS232. Передача данных к Vida88
NC	5	Оставить без подключения
RSDDET	6	Обнаружение кабеля для программирования, не соединён (если используется кабель для программирования, соединить с GND)



### **ЖК-дисплей**

ЖК-дисплей подключается плоским кабелем к разъёму “DISP1”, расположенному на печатной плате устройства. По умолчанию отображает статус портов аналогового и цифрового входа/выхода.

При необходимости настройки контраста ЖК-дисплея, корректировки можно сделать с помощью малоразмерного потенциометра (R95), который расположен слева от разъёма дисплея, (используя маленькую отвёртку).

Используется следующим образом:

Строка 1:

**1234567890ABCDEF**

Строка 2:

**GHIJKLMNOPQRSTU**

Строка 1 показывает статус аналоговых и цифровых выходов (1 2 3 4 – аналоговые выходы, 5 6 7 8 9 0 A B C D E F – цифровые выходы). Строка 2 показывает статус аналоговых и цифровых входов (G H I J – аналоговые входы, K L M N O P Q R S T U V – цифровые входы).

### **Антенна GSM**

Установите антенну, убедитесь, что антенна не находится в непосредственной близости металлических частей, так как они влияют на качество приема сигнала.

### **Подключение программатора (ПЭВМ) к устройству**

Подключение программатора должно осуществляться к порту RS-232. Порт RS-232 используется для установки программ на устройства, конфигурирования устройства и т.д.

Подключение производится через разъем RJ11 (см. рис. 1.3). Порт RS-232 может быть использован в качестве последовательного порта, либо как порта для подключения программатора. В случае, когда порт используется для подключения программатора, контакт (Pin) RSDET (см. табл. 1.7) должен быть соединен с контактом GND. В случае использования порта в качестве последовательного порта, контакт RSDET должен быть исключен из соединения.

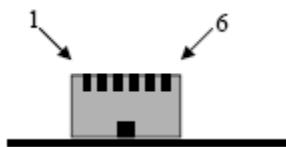


Рис. 1.3 Фронтальное отображение разъема RJ-11

## Схема подключения программатора

Контакт		Описание
Наименование	Номер	
VCC	1	+5В от Vida88, <b>не должен быть использован!</b>
TxD	2	Передача данных от Vida88
GND	3	«земля»
RxD	4	Передача данных к Vida88
NC	5	Оставляется несоединенным
RSDET	6	Порт обнаружения программатора

В лабораторной работе для подключения используется стандартный кабель последовательного порта с соединением контактов, указанных на рисунке 1.4.

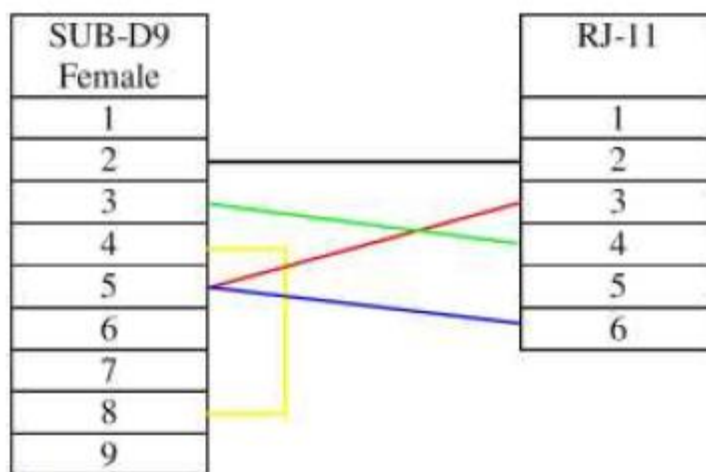


Рис. 1.4 Соединение контактов кабеля

### Загрузка текстовой программы в устройство с помощью программатора

#### Порядок действий:

1. На ПЭВМ запустить приложение RTCUIDE.exe
2. Вкладка Project → Open → выбрать файл с расширением prj
3. Проводник ресурсов, который находится в левой части окна приложения, двойным щелчком мыши выбрать файл \*.vpl.

#### 4. Протестировать программу:

- 4.1. Вкладка Simulator → Simulator. В появившемся окне поставить флаги на модули, используемые в программе. Нажать кнопку Load and Run (загрузить и запустить)
- 4.2. Вкладка Unit → Setup → Transfer firmware to unit. Выбрать файл программы, который будет загружен на устройство.

### Оформление схем алгоритмов и программ.

При выполнении схем алгоритмов и программ отдельные функции алгоритмов и программ, с учетом степени их детализации, отображаются в



виде условных графических обозначений - символов по ГОСТ 19.701-90 - Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. Рассмотрим некоторые положения из этого ГОСТа.

Схемы алгоритмов, программ, данных и систем состоят из имеющих заданное значение символов, краткого пояснительного текста и соединяющих линий.

В стандарте определены символы, предназначенные для использования в документации по обработке данных, и приведено руководство по условным обозначениям для применения их в:

1. схемах данных, которые предназначены для отображения этапов ТПОИ включая ручные операции. Компоненты схемы данных:
  - символы данных;
  - символы процессов;
  - линии и спецсимволы;
2. схемах программ, которые отображают последовательность операций в программе. Состоит из:
  - символов процесса, указывающих фактически операции обработки данных (включая символы, определяющие путь, которого следует придерживаться с учетом логических условий);
  - линейных символов, указывающих поток управления;
  - специальных символов, используемых для облегчения написания и чтения схемы.
3. схемах работы системы, которые предназначены для изображения управления операциями и потоками данных в системе. Компоненты:
  - символы данных;
  - символы процесса;
  - линии, указывающие потоки данных между процессами;
4. схемах взаимодействия программ, которые отображают путь активизации программы и взаимодействие с соответствующими данными каждой программы, показывается в системе только один раз. Компоненты:
  - символы данных, указывающие на используемые данные;
  - символы процесса, указывающие на операции над данными;
  - линии, отображающие потоки данных между процессами.

Все эти компоненты должны быть лаконичными, точными, иметь поясняющие надписи, названия.

Таблица 1.8  
Символы данных

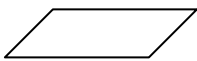
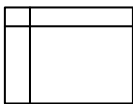
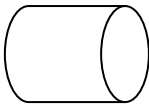
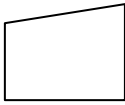
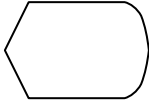
	Данные	данные, носитель не определен
	Запоминаемые данные	данные хранящиеся в виде пригодном для обработки, носитель не определен
	ОЗУ	данные, хранящиеся в ОЗУ
	Запоминающее устройство с последовательным доступом	данные, хранящиеся в запоминающем устройстве с последовательным доступом
	Запоминающее устройство с прямым доступом	данные, хранящиеся в запоминающем устройстве с прямым доступом
	Ручной ввод	данные, вводимые вручную во время обработки с устройств любого типа
	Дисплей	данные, представленные в человекочитаемой форме на носителе в виде отображающего устройства

Таблица 1.9  
Символы процесса

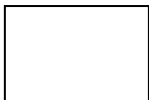

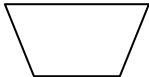

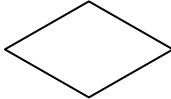
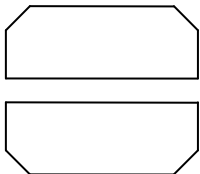
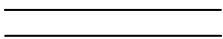
	Процесс	функция обработки данных любого вида
	Предопределенный процесс	процесс, состоящий из одной или нескольких операций, которые определены в другом месте
	Ручная операция	любой процесс, выполняемый человеком
	Подготовка	модификация команды или группы команд с целью воздействия на некоторую последующую функцию
	Решение	процесс переключательного типа, выбор из нескольких альтернативных выходов
	Граница цикла	циклический процесс, начало и конец цикла
	Параллельные действия	синхронизация двух или более параллельных операций

Таблица 1.10  
Символы линий


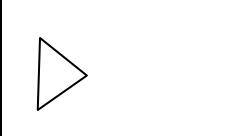
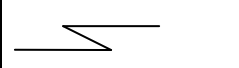
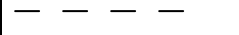
	Линия	поток данных или управления. Могут быть добавлены стрелки
	Передача управления	передача управления от одного процесса к другому. Тип передачи управления должен быть назван внутри символа (запрос, вызов, событие).
	Канал связи	передача данных по каналу связи.
	Пунктирная линия	альтернативная связь между двумя или более символами

Таблица 1.11  
Специальные символы.

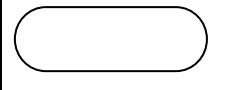

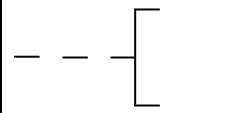
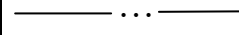
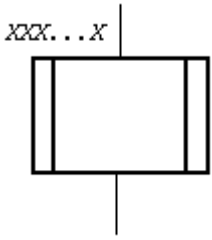
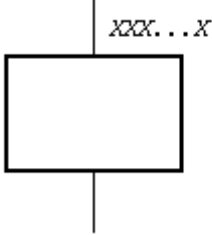
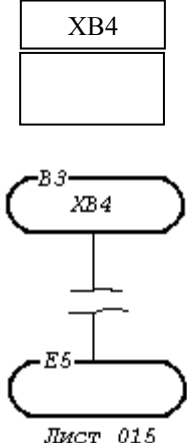
	Терминатор	выход во внешнюю среду и вход из внешней среды
	Соединитель	выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы - соединители должны содержать одно и то же уникальное обозначение.
	Комментарий	используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний
	Пропуск	отображение пропуска символов, в которых не определены ни тип, ни число символов. Он применяется в схемах, изображающих общие решения.

Таблица 1.12

## Правила применения символов и выполнения схем

<p>Символу может быть присвоен идентификатор, (например, для ссылки в других частях документации). Располагается слева над символом.</p>	
<p>Допускается краткая информация о символе (описание, уточнение или другие перекрестные ссылки для более полного понимания функции данной части системы). Описание символа должно помещаться справа над символом</p>	
<p>Детализация программы. Обозначается символом с полосой для процесса или данных. Детализируемая программа начинается и заканчивается символом «Терминатор». Внутри символа «Терминатора», обозначающее начало детализируемой программы, указывается идентификатор данной программы.</p>	

### Построение временной диаграммы.

Для построения временной диаграммы необходимо выделить периодические и аperiodические задачи.

Периодическая задача – задача, которая переходит в состояние выполнения через строго заданный период и выполняется каждый цикл. Четкое и своевременное выполнение. Задача может выполняться несколько раз за цикл. Жесткий крайний срок.

Аperiodическая задача – задача с минимальным приоритетом в системе, выполняющаяся по событию. Функционирует только при отсутствии периодических задач. Мягкий крайний срок. Функции аperiodических задач: диагностика, выдача справочной информации, сохранение на внешнем носителе, и т. д.

При построении временной диаграммы на горизонтальной оси времени необходимо сначала расположить периодические задачи, со строго определенными временами начала и конца выполнения, а затем аperiodические, можно в промежутках периодических задач.

Порядок выполнения работы на примере проектирования системы  
«Система охранной сигнализации»

1) формулировка задания:

Необходимо спроектировать систему охранной сигнализации. Переключатель 1 (SW1) (разбитое окно), переключатель 2 (SW2) (открытие двери), переключатель 3 (SW3) (сработал датчик движения). При включении SW1 и SW3 на телефон, номер которого указан в программе, придет сообщение, содержащее цифру '1'; при включении SW2 и SW3 - придет сообщение, содержащее цифру '-1'; при включении SW1, SW2 и SW3 - придет сообщение, содержащее цифру '0'.

2) описание всех выполняемых задач:

задача 1: опрос входов;

задача 2: отправление SMS сообщения;

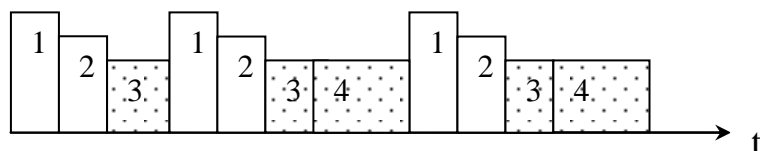
задача 3: работа с дисплеем;

задача 4: опрос питания.

Выделение периодических и аperiodических задач:

Периодические	Аperiodические
1 - Опрос питания	3 - Работа с дисплеем
2 - Опрос переключателей	4 - Отправление SMS сообщения

3) построение временной диаграммы



4) алгоритм работы программы

Описание используемых портов ввода/вывода:

Наименование	Номер разъема	Описание
SW1	27	Переключатель 1
SW2	28	Переключатель 2
SW3	29	Переключатель 3

Переменные, используемые при построении блок-схемы: f1, f2, f3 – флаги.  
Блок – схема приведена на рисунках 1.5 и 1.6.

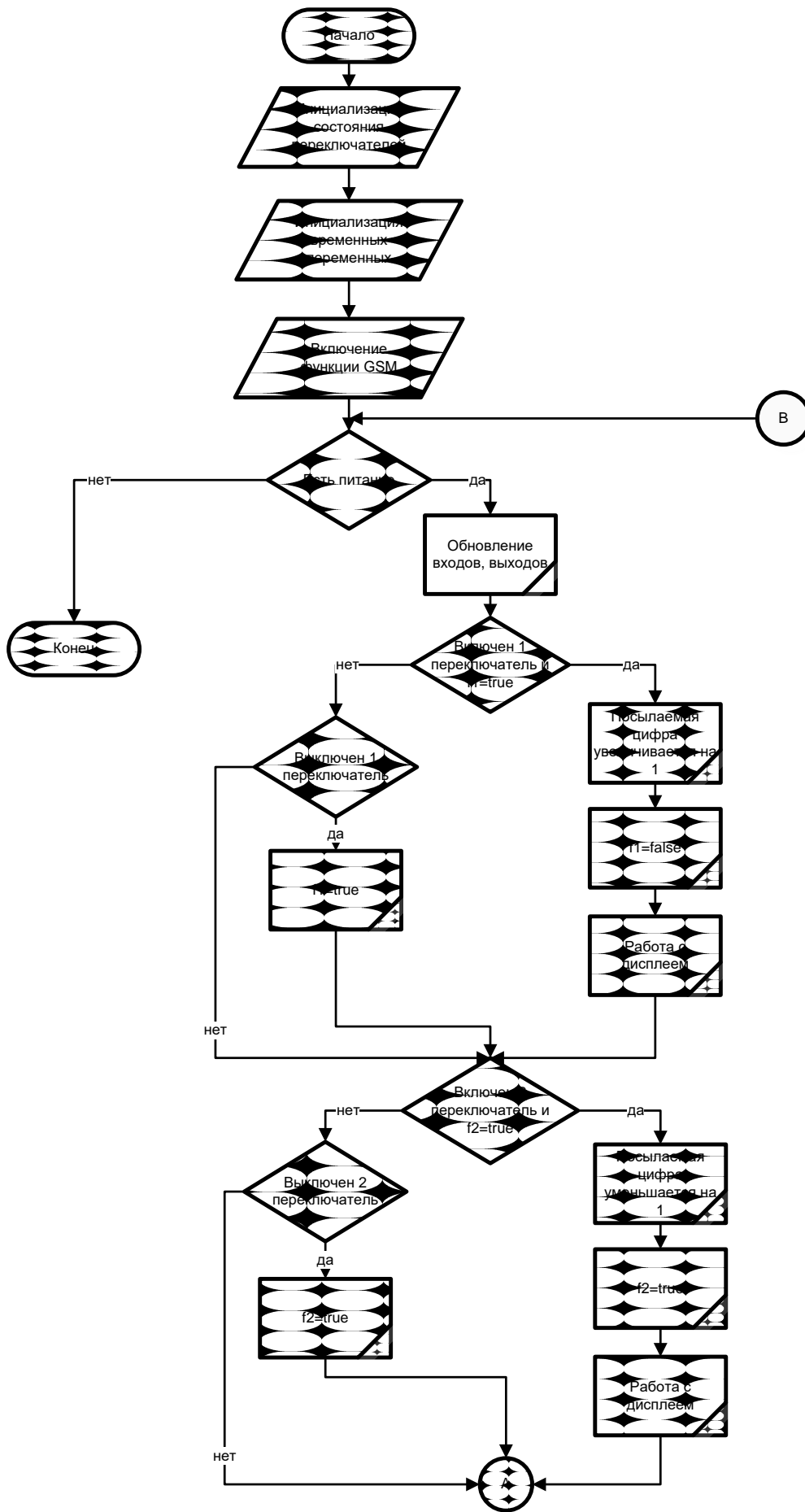


Рис. 1.5 Блок – схема

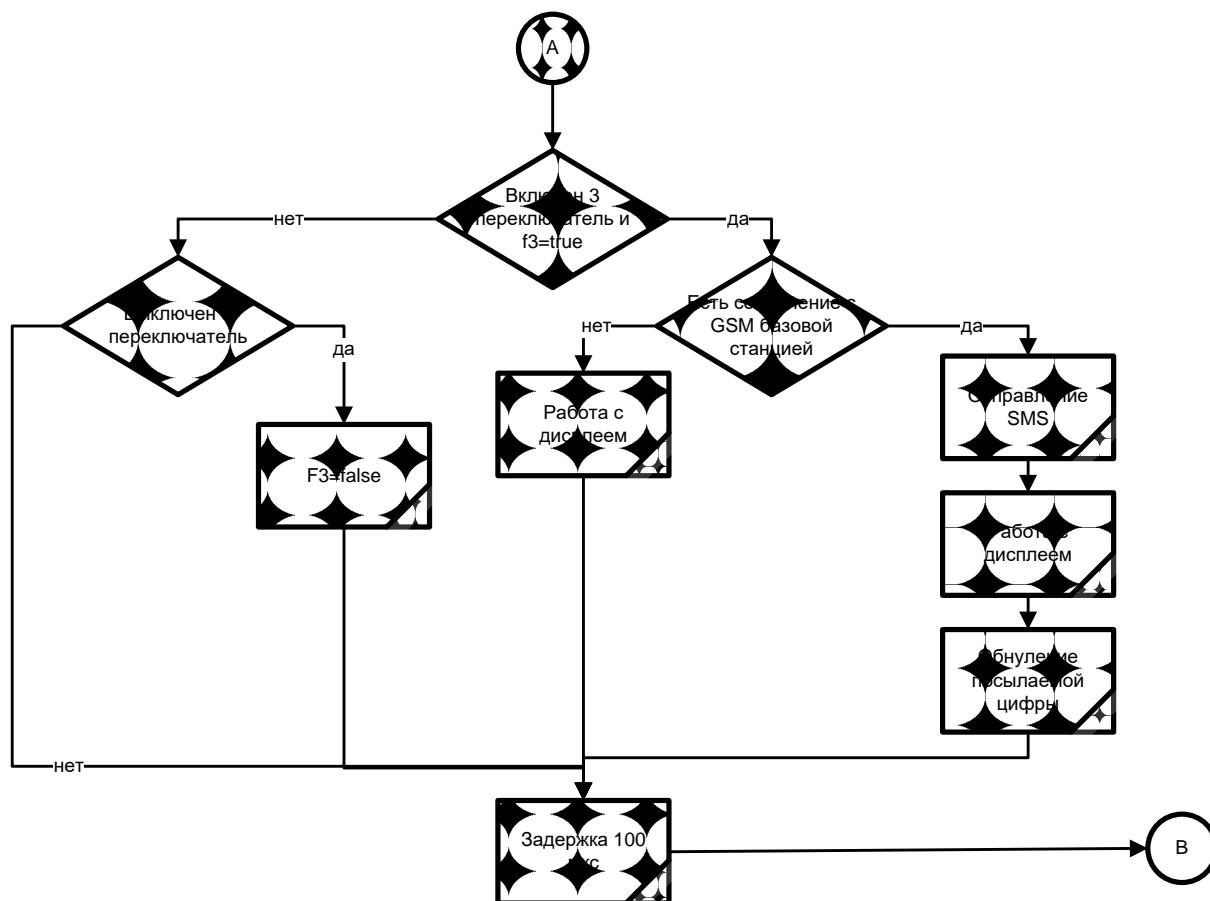


Рис. 1.6 Блок – схема

Содержание отчета:

- 1) цель работы;
- 2) выписать основные характеристики устройства (число портов ввода/вывода, тип сигнала аналоговый/цифровой);
- 3) описание задания;
- 4) сформулировать задачи автоматизации;
- 5) описать используемые в работе порты ввода/вывода, типы сигналов;
- 6) построить блок-схему;
- 7) выделить и расписать периодические и аperiodические процессы;
- 8) построить временную диаграмму;
- 9) Ответы на контрольные вопросы.

Контрольные вопросы:

1. Назначение устройства и область его применения.
2. Основные характеристики устройства.
3. Функциональные возможности устройства.
4. Характеристика аналоговых и цифровых портов.
5. Каким образом осуществляется программирование устройства?
6. Каким образом осуществляется передача данных через GSM?
7. Назначение гальванической развязки на печатной плате.



8. Каким образом осуществляется сетевое взаимодействие устройства с помощью RS-232?
9. Приведите схему подключения программатора.

Варианты заданий:

1. Охранная система. Производить опрос четырех аналоговых входов (сигнал с датчика обрыва) при отсутствии сигнала на любом из четырех аналоговых входов нужно вывести на дисплей номер входа либо номера входов, если их несколько, и отправить SMS содержащую текущую дату, время и номер входа. Частота опросов входов 50 Гц
2. Охранная система. При наличии высокого сигнала на 1 цифровом входе (нарушение внешнего контура) или высокий сигнал на 1 аналоговом входе (не штатный звук) обнаруженного в течение 3 тактов, необходимо вывести на дисплей тип сигнала и подать высокий сигнал на 1 аналоговый выход. Частота опросов входов 100 Гц
3. Охранная система. Если с 21:00 по 9:00 отсутствуют сигналы на 1, 2 цифровых входов (открылась дверь, разбилось стекло) необходимо вывести на дисплей номер входа и подать на 1-ый аналоговый выход высокий сигнал и отправить SMS содержащую текущую дату, время и номер канала. Частота опросов входов 75 Гц
4. Проектирование пожарной сигнализации. 4 комнаты, в каждой по датчику, опрос датчиков производить с частотой 50 Гц. Если на один из четырех цифровых входов подан высокий сигнал (срабатывает один датчик), то сразу подавать воду не надо, а подождать 3 такта, если в течение 3-х тактов было 3 единицы, то дать воду и отправить SMS.
5. Проектирование пожарной сигнализации. Установлены 3 датчика недалеко друг от друга. Если на один из трех цифровых входов подан высокий сигнал (срабатывает один датчик), подать 30% мощности на аналоговый выход, вывести на дисплей цифру 1, если два – увеличить сигнал на 30% и вывести на дисплей - 2, три – подать полную мощность на аналоговый выход, и подать высокий сигнал на 1 цифровой выход (дать воду), и вывести на дисплей - 3. опрос датчиков с частотой 75 Гц.
6. Проектирование пожарной сигнализации. Установлено 3 датчика. Если на один из трех цифровых входов подан высокий сигнал (сработал один из датчиков), отправить SMS с датой, временем и подать высокий сигнал на 1 цифровой выход (включить воду). При этом если есть высокий сигнал на 1 аналоговом входе (в комнате есть человек), подать высокий сигнал на 1 аналоговый выход, а воду подать (1 цифровой выход) через 30 секунд. Опрос датчиков производить с частотой 100 Гц.
7. Система освещения. Если подан высокий сигнал на 1 цифровой вход (человек вошел в комнату) необходимо плавно включить свет с использованием аналогового выхода № 1. Предусмотреть ручное управление с помощью цифрового входа №2.
8. Реализовать систему освещения в подъезде. Контролировать время. С 21:00 – 9:00 подавать высокий сигнал на 1-ый аналоговый выход, в остальное

время – низкий сигнал. Предусмотреть ручное управление с помощью цифрового входа №1.

9. Система освещения. Если пришла SMS на номер SIM-карты, установленной в устройстве, через 5 минут подать высокий сигнал на 1 аналоговый выход (включить свет).
10. Необходимо разработать программу для управления подачей воды в квартиру в зависимости от наличия человека в ней. Устройство должно на основе показаний трех объемных датчиков (три цифровых входа), установленных в ванной, туалете и кухне, управлять задвижкой (аналоговый выход № 1), установленной на главном водопроводе в квартиру. При получении сигнала от датчиков о наличии человека в квартире задвижка должна быть открыта. Если этот сигнал не поступает в течение 5 минут, то задвижку нужно закрыть. Также задвижку нужно закрыть при включении переключателя №1 на устройстве. При выключении переключателя нужно заново начать отсчет времени для закрытия задвижки.
11. Управление подачей воды. Контролировать время. С 19:00 – 8:00 подавать высокий сигнал на 1 аналоговый выход (задвижка открыта), если на цифровом входе № 2 сигнал низкого уровня (человека находится в квартире). Предусмотреть ручное управление с помощью цифрового входа №1.
12. Управление подачей воды. Если на цифровом входе № 1 сигнал низкого уровня (открылась дверь), необходимо подать высокий сигнал на аналоговые выходы № 1 и 2 (открыть 2 крана), когда цифровой вход № 2 пришел сигнал высокого уровня (вода достигла определенного уровня), закрыть краны. Предусмотреть ручное управление с помощью аналоговых входов №1 и №2.
13. Управление отопительной системой. С 9:00 – 18:00 подавать 30% мощности на 1 аналоговый выход (небольшая температура). С 18:00 – 8:00 увеличить мощность. Предусмотреть ручное управление с помощью аналогового входа №1.
14. Управление отопительной системой. Если значение 1 цифрового входа равно единице (человек находится в помещении) и значение 1 аналогового входа равно 800 (в помещении высокая температура) – подать высокий сигнал на 1 цифровой выход (включить кондиционер), иначе (если низкая температура) – подать высокий сигнал на 2 цифровой выход (включить обогреватель). Предусмотреть ручное управление с помощью цифровых входов №1 и №2.
15. При получении SMS на номер sim-карты, установленной в устройстве, подавать высокий сигнал на аналоговые выходы № 1, 2 (набрать воду в ванну) до получения низкого сигнала на цифровой вход № 1 (до определенного уровня). Подать высокий сигнал на цифровой выход № 1 (включить обогреватель), через 10 минут после получения SMS подать высокий сигнал на цифровой выход № 2 (включить свет).

## Практическая часть

### РАБОТКА ПО ДЛЯ СИСТЕМЫ СБОРА ДАННЫХ И УПРАВЛЕНИЯ VIDA 88

Цель работы: получить практические навыки разработки ПО на языке программирования VPL. Разработать ПО для системы сбора данных и управления на базе устройства Vida 88.

#### Описание задания:

- 1) ознакомиться с основными элементами языка программирования VPL;
- 2) выписать структуру программы, операторы: if-then-else, case-of, while-do, for-to-do, repeat-until, - типы переменных, структуру объявления функции, операции языка, распределить операции по приоритетам;
- 3) выписать специфические функции, с которыми будет работать приложение;
- 4) разработать ПО в соответствии с вариантом из лабораторной работы № 1 на языке VPL.

### Язык программирования VPL

#### Структура программы VPL

1://-----	}	Комментарий может начинаться с символа </*!> или для открытия комментария – символ /*, для закрытия - /*
2:// Greenhouse_1.vpl, created 2000-12-31 14:45		
3://		
4://-----		
5:NCLUDE rtcu.inc	}	Включает определение всех встроенных функций (других файлов) в программу
6:		
7:// переменные ввода	}	Объявление всех переменных
8:VAR_INPUT		
9:		
10:END_VAR;		
11:		
12:// переменные вывода		
13:VAR_OUTPUT	}	Объявление глобальных переменных
14:		
15:END_VAR;		
16:		
17:// глобальные переменные программы		
18:VAR		
19:		
20:END_VAR;		
21:		

22:PROGRAM Greenhouse_1;	Имя программы
23:	
24:BEGIN	} Тело программы
25:	
26:END;	
27:	
28:END_PROGRAM;	Конец программы

INCLUDE используется для включения другого файла в пределах текущего файла и встроенных функций. rtcu.inc содержит объявления всех стандартных функций на RTCU. INCLUDE файлы часто используются для общих функциональных возможностей, которые могут использоваться в нескольких проектах.

### Основные операторы:

#### 1) Операторы выбора:

##### a) IF – THEN – ELSE

```
IF <выражение> THEN
<утверждение> <утверждение>
END_IF
```

```
IF < выражение >THEN
< утверждение >;
ELSIF < выражение >THEN
< утверждение >;
ELSIF < выражение >THEN
< утверждение >;
ELSE < утверждение >;
END_IF
```

##### б) CASE – OF

```
CASE<выражение> OF
<константа1>, <константа2>,<константа3> : <утверждение>;
<константа1> .. <константа2> : <утверждение>;
<константа1> .. <константа2>, <константа3> : <утверждение>;
ELSE <утверждение>;
END_CASE
```

#### 2) Операторы цикла:

##### a) WHILE – DO

```
WHILE <выражение> DO
<утверждение>;
END_WHILE;
```

## б) FOR – TO – DO

```
FOR <переменная>:=<выражение> TO <выражение> BY <выражение > DO  
<утверждение >;  
END_FOR;
```

## в) REPEAT – UNTIL

```
REPEAT <утверждение>;  
UNTIL <выражение>  
END_REPEAT;
```

## **Типы переменных:**

INT (от -32768 до 32767) – целый тип  
SINT (от -128 до 127) – короткий целый  
DINT (от -2147483648 до 2147483647) – длинный целый  
STRING (может содержать строку) – строковый тип  
BOOL (истина или ложь) – логический тип

## **Функции**

Объявление функций:

```
FUNCTION <имя> : return type;
```

```
VAR_INPUT
```

```
<переменная> : <тип>;
```

```
... ..
```

```
<переменная> : <тип>;
```

```
END_VAR;
```

```
VAR
```

```
<переменная> : <тип>;
```

```
... ..
```

```
<переменная> : <тип>;
```

```
END_VAR;
```

```
<утверждение>
```

```
END_FUNCTION;
```

*return type* (возвращаемый тип) может быть: BOOL, SINT, INT или DINT.

*VAR\_INPUT* содержит переменные, которые должны быть объявлены вне функции (переменные не могут быть изменены в функции). *VAR* содержит частные переменные функции, которые могут читаться и изменяться изнутри функции. Утверждения, которые находятся в теле функции, могут использовать частные переменные.

Пример:

```
FUNCTION Add : INT;
```

```
VAR_INPUT
```

```
a : INT;
```

```
b : INT;  
c : INT;  
END_VAR;  
Add := a+b+c;  
END_FUNCTION;
```

### **Операции языка VPL:**

#### 1. Унарные

- отрицание

NOT логическое отрицание

\*\* степень

#### 2. Бинарные

+ сложение

- вычитание

\* умножение

/ деление

MOD модуль

#### 3. Операторы сравнения

< меньше чем

<= меньше или равно

> больше

>= больше или равно

= равно

<> неравно

#### 4. Логические операторы

AND логическое И

XOR исключающее логическое ИЛИ

OR логическое ИЛИ

#### 5. Назначенные приоритеты в выполнении операции:

1. ( ) – круглые скобки

2. Function

3. SIZEOF

4. ADDR

5. Унарные операции

6. Бинарные операции

7. Операторы сравнения

8. Логические операторы

### **Функции GSM.**

Функции GSM позволяют программе взаимодействовать с модулями GSM на RTCU платформе.

#### gsmConnected

Функция проверяет, находится ли GSM модуль в соединении с GSM базовой станцией. Входящие параметры: нет. Возвращаемое значение bool:

true – есть соединение с базовой станцией, false – нет соединения с базовой станцией.

```
INCLUDE rtcu.inc
PROGRAM test;
BEGIN
...
...
// Проверка успешности соединения с GSM базовой станцией
IF gsmConnected() THEN
// Устройство соединилось с базовой станцией
...
...
ELSE
// Устройство не соединилось с базовой станцией
...
...
END_IF;
END;
END_PROGRAM;
```

#### gsmIncomingCall

Функциональный блок используется для проверки входящего звонка. Выходные параметры: status: int(0..2) может принимать значения 0 – нет входящего звонка, 1 – входящий звонок с номером вызывающего, 2 – входящий звонок без номера вызывающего, phonenummer: string - телефонный номер (если статус 1).

```
INCLUDE rtcu.inc
PROGRAM test;
VAR
incoming : gsmIncomingCall;
END_VAR;
// Включение питания на GSM модуле
gsmPower(power := TRUE);
BEGIN
incoming();
...
...
// Проверка входящего звонка
IF incoming.status = 1 THEN
// Входящий звонок с номером вызывающего
// Ответ на входящий звонок
gsmAnswer();
// Отправить голосовое сообщение
voiceTalk(message := "Hello");
//
gsmHangup();
```

```
...
...
END_IF;
END;
END_PROGRAM;
```

#### gsmPower

Функция контролирует электропитание GSM модуля. Используя эту функцию, можно включать и выключать GSM модуль. Входные параметры: power:bool может принимать значения: true (включить питание), false (выключить питание). Возвращаемое значение:int: 0 – если задача выполнена правильно, 1 – если происходит вызов данных (выключение не возможно в данном случае), 2 – если модем заблокирован из-за неполадок в сети, 8 – если скорость системы не поддерживаемая (должно быть 48 МГц), 9 – если модем уже работает по способу LP (пониженная скорость соединения, GSM модуль включен функцией gsmPowerLP).

```
INCLUDE rtcu.inc
PROGRAM test;
// Включение питания на GSM модуле
gsmPower(power := TRUE);
BEGIN
...
...
END;
END_PROGRAM;
```

#### gsmSendSMS

Функция отправляет SMS сообщения, используя GSM модуль. Входные параметры: phonenumber:string - телефонный номер (максимальная длина 20), message:string - строка сообщения. Возвращаемое значение:int: 0 – если сообщение отправлено успешно, 1 – если общая ошибка, 2 – если CMS/CME ошибка (ошибка на сети GSM), 3 – если GSM модуль не связан с основной станцией, 4 – если модуль GSM не включен, 5 – если модуль GSM занят вызовом данных.

```
INCLUDE rtcu.inc
PROGRAM test;
// Включение питания на GSM модуле
gsmPower(power := TRUE);
BEGIN
...
...
// Примечание: в этом примере не проверяется код возвращаемого
значения из //функции gsmSendSMS
gsmSendSMS(phonenumber := "+44 12 34 56 789", message := "This is a
test
message sent using SMS");
...
...

```



```
...
END;
END_PROGRAM;
```

### **Функции дисплея.**

Функции дисплея позволяют программе взаимодействовать с LCD дисплеем на RTCU платформе.

#### displayBackLight

Функция контролирует яркость дисплея. Входные параметры: intensity:int(0..100) – яркость в %. Возвращаемое значение: нет.

```
INCLUDE rtcu.inc
PROGRAM test;
// Настроить яркость дисплея равную 75%
displayBacklight(intensity := 75);
BEGIN
...
...
END;
END_PROGRAM;
```

#### displayClear

Функция очищает содержимое дисплея. После ее вызова текущая записываемая позиция будет установлена слева вверху. Входные параметры: нет. Возвращаемое значение: нет.

```
INCLUDE rtcu.inc
PROGRAM test;
BEGIN
...
...
// очистить дисплей
displayClear();
...
...
END;
END_PROGRAM;
```

#### displayString

Функция печатает текст сообщения на дисплее с текущей записываемой позиции. Входные параметры: message:string – текст сообщения. Возвращаемое значение: нет.

```
INCLUDE rtcu.inc
PROGRAM test;
BEGIN
...
...
// Установить текущую записываемую позицию к 1,2
displayXY(x := 1, y := 2);
// Отобразить сообщение, начиная с текущей записываемой позиции
```

```

displayString(message := "Hello world");
...
...
END;
END_PROGRAM;

```

#### displayNumber

Функция печатает число на дисплее с текущей записывающей позиции.

Входные параметры: number:dint – число. Возвращаемое значение: нет.

```

INCLUDE rtcu.inc
PROGRAM test;
BEGIN
...
...
// Установить текущую записывающую позицию к 1,2
displayXY(x := 1, y := 2);
// Отобразить число 4711, начиная с текущей записывающей позиции
displayNumber(number := 4711);
...
...
END;
END_PROGRAM;

```

#### displayXY

Функция устанавливает текущую позицию для записи. Входные параметры: X:int(1..20) – колонка, Y:int(1..4) – строка. Возвращаемое значение: нет.

```

INCLUDE rtcu.inc
PROGRAM test;
BEGIN
...
...
// Установить текущую записывающую позицию к 1,2
displayXY(x := 1, y := 2);
// Отобразить число 4711, начиная с текущей записывающей позиции
displayNumber(number := 4711);
...
...
END;
END_PROGRAM;

```

#### **Функции часов реального времени.**

Функции часов позволяют программе взаимодействовать с часами реального времени на платформе RTCU.

#### clockGet

Функциональный блок для чтения реального времени. Входные значения: нет. Выходящие значения: year:int (1980..2048) – текущий год, month:sint (1..12) – текущий месяц, day:sint (1..31) – текущий день, hour:sint

(0..23) – текущий час, minute:sint (0..59) – текущая минута, second:sint (0..59) – текущая секунда, linsec:dint - количество секунды с 1980-1-1 00:00:00.

```
INCLUDE rtcu.inc
VAR
clock : clockGet; // описание функционального блока clockGet
END_VAR;
PROGRAM test;
BEGIN
clock();
// Проверка, если месяц декабрь
IF clock.month = 12 THEN
...
...
END_IF;
...
...
END;
END_PROGRAM;
```

#### clockSet

Функция устанавливает часы реального времени. Входные параметры: year:int (2000..2048) – текущий год, month:sint (1..12) – текущий месяц, day:sint (1..31) – текущий день, hour:sint (0..23) – текущий час, minute:sint (0..59) – текущая минута, second:sint (0..59) – текущая секунда, linsec:dint – секунды с 1980-1-1 00:00:00.

```
INCLUDE rtcu.inc
PROGRAM test;
// Установка часов реального времени на 17 мая 2000. Примечание:
// часы, минуты, секунды не установлены
clockSet(year := 2000, month := 5, day := 17);
BEGIN
...
...
END;
END_PROGRAM;
```

### Порядок выполнения работы на примере проектирования системы «Система охранной сигнализации»:

#### Формулировка задания:

Необходимо спроектировать систему охранной сигнализации. Переключатель 1 (SW1) (разбитое окно), переключатель 2 (SW2) (открытие двери), переключатель 3 (SW3) (сработал датчик движения). При включении SW1 и SW3 на телефон, номер которого указан в программе, придет сообщение, содержащее цифру '1'; при включении SW2 и SW3 - придет

сообщение, содержащее цифру '-1'; при включении SW1, SW2 и SW3 - придет сообщение, содержащее цифру '0'.

На основе временной диаграммы и блок-схемы алгоритма, описанными в лабораторной работе № 1, разработано ПО на языке программирования VPL для системы охранной сигнализации на базе устройства Vida 88.

Листинг программы:

```
INCLUDE rtcu.inc
VAR_INPUT
  sw1, sw2, sw3 : BOOL; // переключатели
END_VAR;
VAR_OUTPUT
END_VAR;
VAR
  iSend : INT; // посылаемая цифра
  f1, f2, f3 : BOOL; // флаги
END_VAR;
PROGRAM test;
  gsmPower(power := TRUE); // включение GSM модуля
  // работа с дисплеем
  displayBacklight(intensity := 75); // интенсивность яркости дисплея
  displayClear(); // очищение дисплея
  displayXY(x := 1, y := 1); // установка текущей позиции для записи
  iSend := 0;
  displayString(message := "HELLO"); // отображение сообщения на дисплее
  f1 := TRUE; f2 := TRUE; f3 := TRUE;
BEGIN // бесконечный цикл
  UPDATEIO; // обновляет входов и выходов
  if sw1 and f1 then
    iSend := iSend + 1;
    f1 := FALSE;
    displayClear();
    displayNumber(number := iSend);
  elseif sw1 = FALSE then
    f1 := TRUE;
  end_if;
  if sw2 and f2 then
    iSend := iSend - 1;
    f2 := FALSE;
    displayClear();
    displayNumber(number := iSend);
  elseif sw2 = FALSE then
    f2 := TRUE;
  end_if;
  if sw3 and f3 then
    if gsmConnected() then // если есть соединение
```

```

    // посылаем на телефон соответствующую цифру
    gsmSendSMS(phonenumber := "+7*****", message := intToStr(v :=
iSend));
    displayClear();
    // отображение на дисплее сообщения с соответствующей цифрой
    displayString(message := strFormat(format := "Was sent \1", v1 := iSend));
    iSend := 0;
else
    displayClear();
    displayString(message := "No GSM");
end_if;
f3 := FALSE;
elseif sw3 = FALSE then
    f3 := TRUE;
end_if;
Sleep(delay := 100); // задержка на 100 мкс
END_WHILE;
END;
END_PROGRAM;

```

Работоспособность системы проверена на симуляторе RTCU IDE. После программа была загружена в устройство Vida 88 и протестирована на устройстве.

#### Содержание отчета:

- 1) цель работы;
- 2) основные элементы языка программирования VPL;
- 3) описание задания;
- 4) выписать используемые в работе специфические функции языка;
- 5) листинг программы разработанной в соответствии вариантом;
- 6) скриншоты выполнения программы в симуляторе;
- 7) ответы на контрольные вопросы.

#### Контрольные вопросы:

1. Объявление, каких стандартных функций содержится в файле rtcu.inc.
2. Состав и назначение основных разделов программы.
3. Опишите 5 функциональных возможностей устройства, реализованных программным способом.
4. Каким образом осуществляется обработка нажатия переключателя в программе?
5. Как осуществить в программе смену цвета светодиода при индикации?
6. Составить схему алгоритма настройки и конфигурирования GSM модуля в виде блочных структур.
7. В каких единицах указывается яркость дисплея?
8. Какое максимальное значение может принимать аналоговый сигнал?
9. За что отвечает параметр linsec в функциях часов реального времени?