УДК 004.41

РАЗРАБОТКА СИСТЕМЫ КОНТРОЛЯ УДАЛЕННЫХ ОБЪЕКТОВ НА ОСНОВЕ ОПЕРАЦИОННОЙ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

Ташев А.А.
магистр 2 курса,
Орловский государственный университет имени И.С. Тургенева,
г. Орел, Россия

Аннотация

В статье описана разработка системы контроля удаленных объектов на основе операционной системы реального времени FreeRTOS. Спроектирована структурная схема диспетчерского контроля. Приведено описание алгоритма работы управляющей программы. Описаны основные API функции системы FreeRTOS, их применение при создании системы контроля.

Ключевые слова: операционная система реального времени, микроконтроллер, GSM/GPRS модем, диспетчеризация, удаленные объекты.

DEVELOPMENT OF MONITORING SYSTEM OF REMOTE FACILITIES BASED ON REAL-TIME OPERATING SYSTEM

Tashev A.A.
master 2 course
Orel state University named after I.S. Turgenev,
Orel, Russia

Abstract

The article describes the development of monitoring system of remote facilities based on the FreeRTOS real-time operating system. Designs the structural scheme of dispatching control. The following is the description of the algorithm of the control program. Describes the basic functions of the freeRTOS system, their application in the creation of the monitoring system.

Keywords: real-time operating system, microcontroller, GSM/GPRS modem, scheduling, remote objects.

Суть проектирования систем диспетчеризации заключается в решении задачи визуализации информации о функционировании инженерных систем и предоставлении оператору возможности прямого управления оборудованием из диспетчерского пункта. Данные о состоянии инженерного оборудования поступают от контроллеров локальной автоматики и передаются на сервер. Обработанные технологические данные с необходимой аналитической информацией поступают на сервер диспетчеризации и выводятся на экранах компьютеров на рабочих местах операторов в наглядном динамическом графическом виде [1].

При диспетчеризации значительно удаленных (свыше 5 км) объектов используется сотовая сеть GSM. Модем с SIM картой устанавливается на объекте и по сотовой сети (GPRS) передает данные в диспетчерскую. Сервер в диспетчерской должен иметь выход в интернет и фиксированный IP адрес. Если в диспетчерском центре нет возможности предоставить серверу доступ в интернет, то требуется установка GSM модема на сервер. Модем работает либо через интернет, либо в режиме дозвона (CSD) [2].

Целью данной статьи является разработка системы контроля, сбора и передачи информации в диспетчерский пункт с помощью GSM-модема.

Представленная система контроля разработана на базе операционной системы FreeRTOS. FreeRTOS — многозадачная операционная система реального времени (ОСРВ) с открытым исходным кодом, предназначенная для

встраиваемых систем [3]. Основой ОСРВ является ядро (Kernel). Оно реализует основополагающие функции ОС. Каждая выполняющаяся в среде FreeRTOS программа представляет собой задачу (Task). Для управления задачами используется планировщик (Scheduler). Он определяет, какая из задач, готовых к выполнению, выполнится в данный конкретный момент времени. В статье дается описание API функций данной ОСРВ, которые были использованы при разработке управляющей программы.

За работу управляющей программы отвечает микроконтроллер (МК) ATXMEGA128A3 компании Atmel. Данный МК имеет 128 кБ памяти для хранения коды программы, 8 кБ ОЗУ и 2 кБ энергонезависимой памяти для хранения пользовательских настроек [5]. Для связи с GSM-модемом и внешними устройствами используется встроенный интерфейс ввода-вывода USART.

Данная автоматизированная система диспетчеризации выполняет следующие функции:

- передача SMS-сообщений оператору о нарушении охраны, пропадании сетевого питания, пропадании связи между устройствами, появлении сигнала аварии, появлении аварии с указанием типа аварии;

- циклический контроль устройств, подключенных по интерфейсу RS-485;

- контроль состояния собственных входов (охранный шлейф, сетевое питание, дискретные и аналоговые входы);

- передача данных о текущем состоянии объекта (состояния всех устройств, подключенных по интерфейсу RS-485 и состояния собственных входов и выходов) по запросу от автоматизированного рабочего места (АРМ) верхнего уровня по сотовой связи;

- передача данных о текущем состоянии объекта на АРМ центрального диспетчерского пункта (ЦДП) по сотовой связи при возникновении аварии;

- передача данных о текущем состоянии объекта с заданной периодичностью на FTP-сервер;

- передача данных о текущем состоянии объекта на FTP-сервер при возникновении аварии;

- контроль параметров в устройствах, подключенных по интерфейсу RS-485.

Ниже (рис.1) представлена упрощенная структурная схема работы системы диспетчеризации.

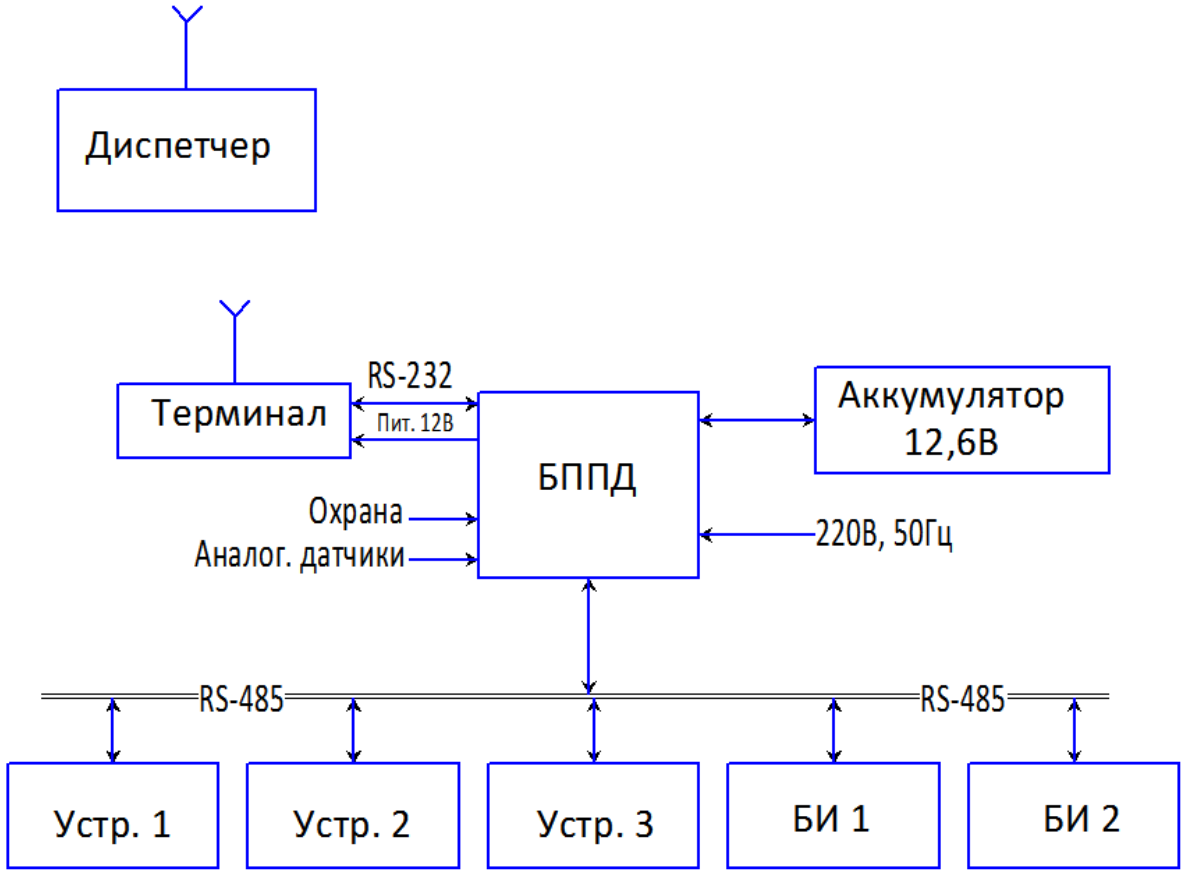

Рис.1 – Структурная схема работы системы диспетчеризации

Устройства, с которых поступают данные для блока приема/передачи данных, соединены интерфейсом RS-485, где устр. 1-3 – устройства контроля и защиты с различным функционалом, БИ 1-2 – блоки индикации. Блок приема/передачи данных (БППД) соединен с терминалом GSM по интерфейсу RS-232. Также на терминал поступает питание 12В от внутреннего блока питания БППД. Терминал передает полученные данные диспетчеру и на FTP сервер. Само устройство питается от сетевого напряжения 220В, 50Гц при

наличии такового. Если пропадает питание, устройство переходит в режим питания от аккумулятора с выходным напряжением 12,6В.

В основной программе происходит инициализация всех используемых модулей, портов ввода/вывода и модулей USART для передачи данных по интерфейсам RS-232 и RS-485. Далее создаются 2 задачи:

- задача по работе с сотовым терминалом, которая выполняет функции отправки данных по CSD, на FTP, с помощью SMS, а также приема входящих CSD звонков;

- задача RTOS опроса устройств по интерфейсу RS-485, которая последовательно опрашивает все подключенные устройства.

Для создания задач используется API функция xTaskCreate(), которая принимает следующие параметры:

- pvTaskCode – указатель на функцию, которая реализует задачу;

- pcName – описательное имя для задачи;

- usStackDepth – размер стека, используемого для задачи ожидания;

- pvParameters – параметр, который принимает задача;

- uxPriority – приоритет, с которым будет выполняться задача;

- pxCreatedTask – дескриптор задачи.

Если задача была успешно создана, то функция возвращает значение pdTRUE. Значение errCOULD_NOT_ALLOCATE_REQUIRED_MEMORY показывает, что задача не создана, так как в куче недостаточно свободной памяти для FreeRTOS, чтобы она могла выделить место для структур данных задачи и стека [4].

Далее создается очередь для обмена данными между задачами. Очереди (queue) являются нижележащим примитивом, используемым для всех механизмов коммуникации и синхронизации задач FreeRTOS. При создании используется API функция xQueueCreate(), где указывается максимальное количество элементов, которое можно сохранить в очереди, и размер в байтах каждого элемента данных, который можно сохранить в очереди. Очередь

считается успешно созданной, если функция вернула не NULL (пустое значение) [4].

В конце запускается диспетчер задач (scheduler) с помощью функции vTaskStartScheduler(). После передачи управления диспетчеру происходит запуск системных утилит, обеспечивающих обслуживание функций ядра RTOS. Далее происходит передача управления задаче в соответствии с состояниями задач и их приоритетов. Сам алгоритм под управлением диспетчера задач работает по принципу прерывания. Источником прерывания служит аппаратный таймер микроконтроллера, который срабатывает через время, равное 1 мс. Таким образом, диспетчер срабатывает каждую 1 мс, что является одним тиком диспетчера (дискретностью диспетчера) [3].

На приведенной ниже блок-схеме (рис.2) показан алгоритм задачи по работе с сотовым терминалом. Он показывает, по какому принципу происходит передача данных по CSD, на FTP, а также прием входящих CSD звонков.

Перед подключением к терминалу и началом обмена данными происходит задержка T, равная 3000 мс. Она нужна для того, чтобы терминал успел инициализироваться и подключиться к сети. До этого момента управление передается диспетчеру, который запускает следующую по очереди задачу.

Далее начинается опрос по интерфейсу RS-232. Если передача закончена и в течении 500 мс новые данные не поступает, то сбрасывается таймер ожидания приема данных. Происходит чтение данных из буфера, в котором находятся показания со всех опрашиваемых устройств.

После запускается подпрограмма обработки принятых команд от терминала, происходит выполнение системных команд для терминала (например, перезагрузка терминала при его зависании). Если есть команда запроса от терминала, то устройство передает запрашиваемые значения по CSD каналу. При возникновении аварии происходит передача соответствующей информации по CSD на номер телефона и через GPRS на указанный FTP

сервер. Для периодической передачи данных на FTP используется таймер, время которого настраивается оператором.

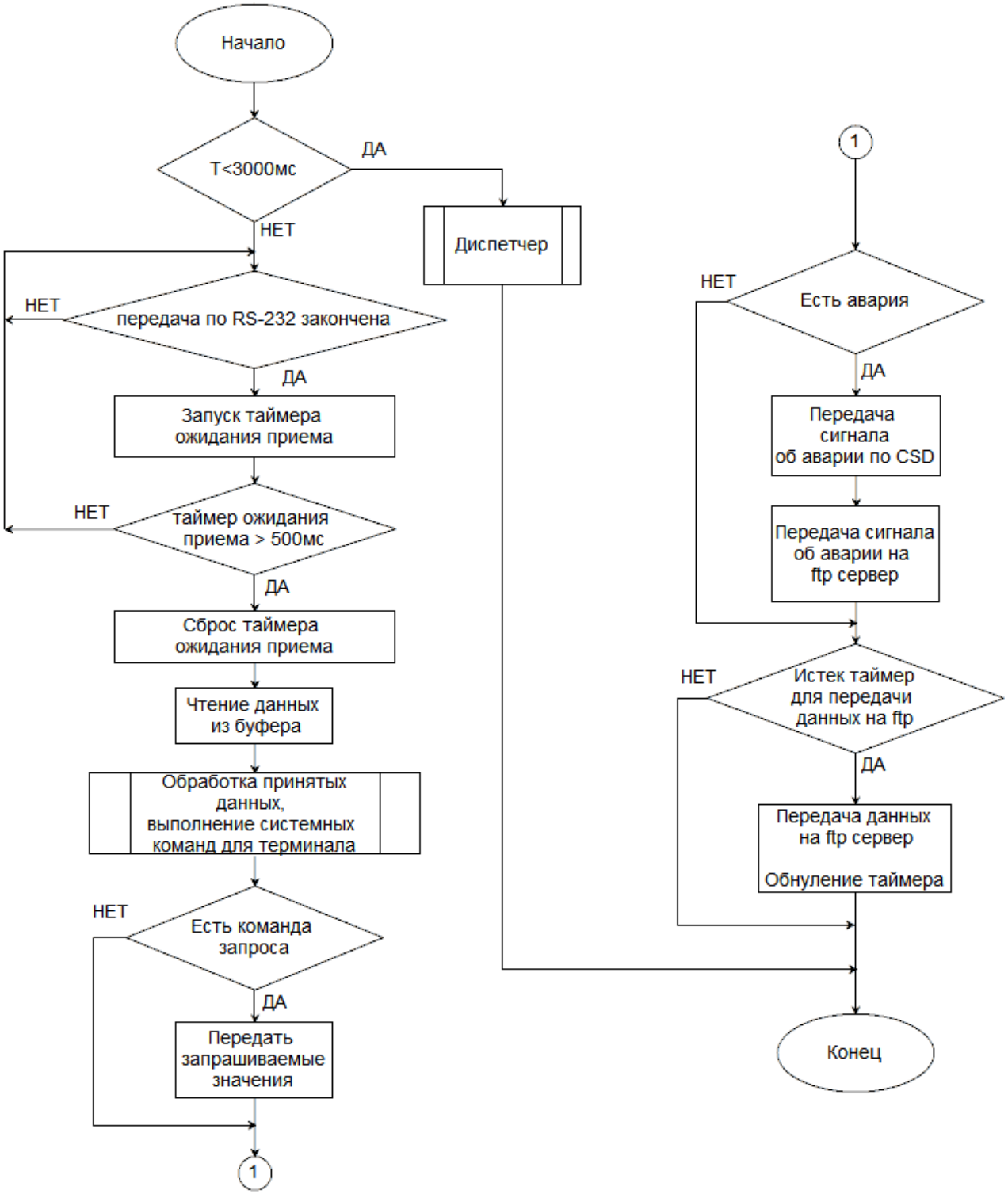

Рис.2 – Алгоритм задачи по работе с сотовым терминалом

На рис.3 изображена блок-схема задачи опроса устройств по интерфейсу RS-485.

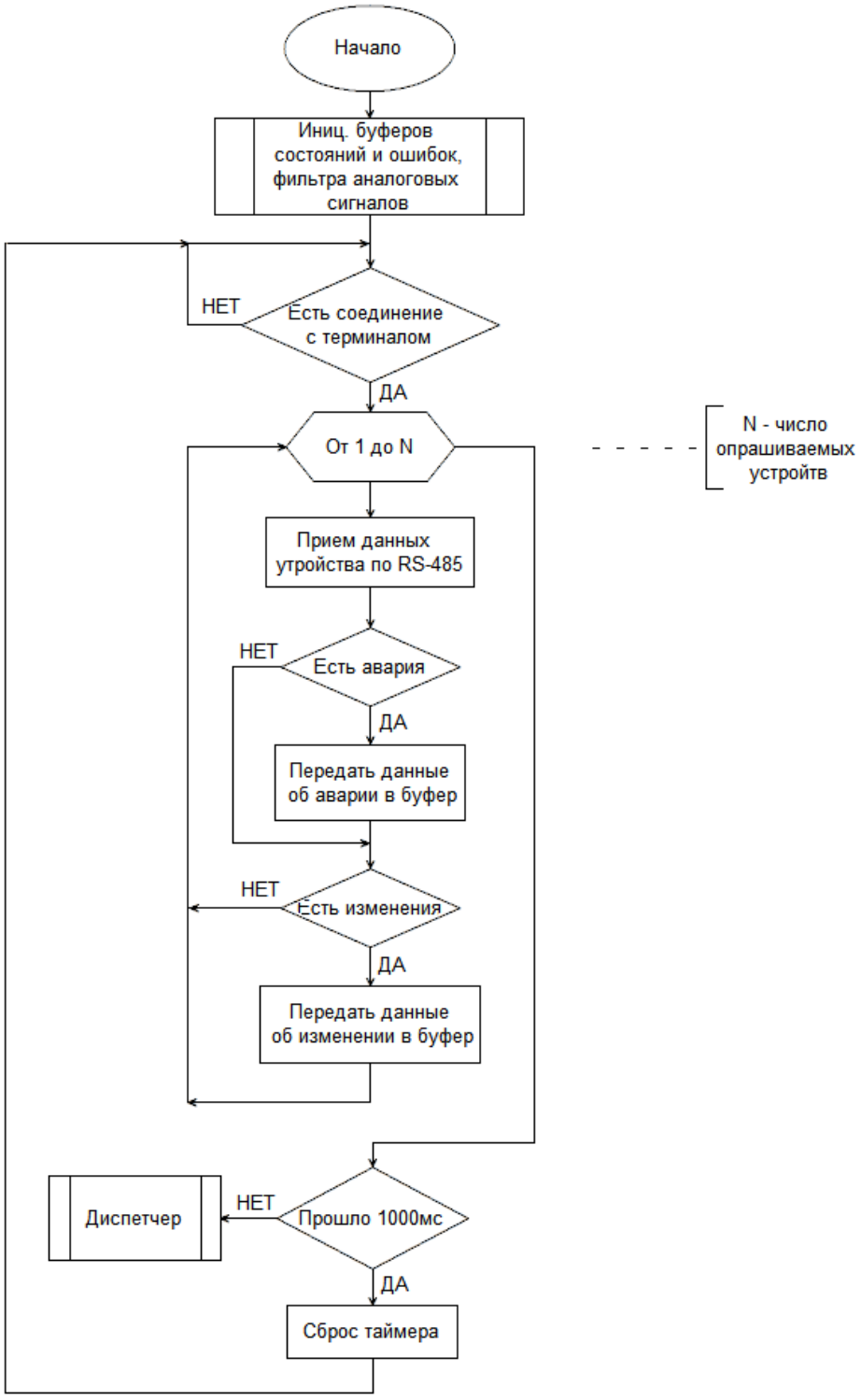

Рис.3 – Алгоритм задачи опроса устройств по интерфейсу RS-485

Вначале инициализируются буферы состояний и ошибок, фильтр аналоговых сигналов. Происходит проверка соединения с терминалом. Если соединения нет, то опроса устройств не происходит.

Далее идет последовательный опрос всех подключенных устройств, организованный в цикле. Устройства опрашиваются по интерфейсу RS-485, данные с них поступают в отдельный буфер. Данный буфер нужен для того, чтобы при CSD запросе состояния подключенных устройств сразу передать показания. Если поступает сигнал аварии от устройства, то данные об аварии передаются в очередь для последующей передачи терминалу. Аналогичные действия происходят при изменении отслеживаемых параметров.

После окончания цикла опроса происходит задержка, равная 1000мс. Управление на период этой задержки передается диспетчеру, который запускает следующую по очереди задачу.

На этом создание системы контроля удаленных объектов на основе операционной системы реального времени FreeRTOS завершено.

В результате проведенной работы была разработана система, позволяющая в реальном времени осуществлять контроль удаленных объектов из диспетчерского пункта и оперативно реагировать на возникающие аварийные ситуации. Необходимо отметить, что разработанная система имеет множество возможностей для расширения и усложнения функционала благодаря гибкой структуре ОСРВ FreeRTOS. Также наличие множества версий данной ОС для различных архитектур микроконтроллеров дает возможность быстрого перехода на другую платформу в случае необходимости.

Библиографический список

1. Матвейкин В.Г. Системы диспетчеризации и управления: учебное пособие / В.Г. Матвейкин, Б.С. Дмитриевский, И.С. Панченко, М.В. Кокорева. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 96 с.
2. Ельцов А. К вопросу о диспетчеризации / А. Ельцов / Автоматизация и производство. – 2011. - №1. - С. 29-31.
3. Курниц А. FreeRTOS – операционная система для микроконтроллеров / А. Курниц / Компоненты и технологии. – 2011. - №2. - С. 96-100.
4. FreeRTOS: практическое применение: [Электронный ресурс]. – Режим доступа – URL: http://microsin.net/programming/ARM/freertos-part1.html (дата обращения: 15.03.18).
5. Microchip Technology, официальный сайт: [Электронный ресурс]. – Режим доступа – URL: http://www.microchip.com/ (дата обращения: 26.03.18).