

С. В. Карпеш, А. А. Петров

Разработка систем автоматического управления погружной системой жидкостного охлаждения высокопроизводительных вычислительных систем

Аннотация. В работе рассмотрена задача построения системы автоматического управления погружной системой жидкостного охлаждения.

Ключевые слова и фразы: Система мониторинга, MODBUS, AVR, OpenSCADA, JS.

1. Введение

Наряду с традиционными воздушными системами охлаждения все большее распространение получают жидкостные. Жидкостное охлаждение более эффективно благодаря большей теплоемкости, коэф. теплопередачи и т.д. Ещё одним плюсом является то, что жидкостные системы охлаждения гораздо компактнее традиционных воздушных кулеров и имеют меньшее энергопотребление. Однако, функционирование погружной системы охлаждения невозможно без дополнительного потребления энергии: насосы, перекачивающие охлаждающую жидкость, вентиляторы драйкулера. Дальнейшее увеличение энергоэффективности связано с построением системы управления процессами теплопереноса и рассеивания тепла в окружающую среду.

В работе рассмотрены задачи построения программно-аппаратного комплекса, регулирующего алгоритма системы управления охлаждением.

2. Задача

Для контроля теплового режима вычислительных узлов и поддержания температуры охлаждающей жидкости, не превышающей заданную, необходимы средства управления. Их реализация требует алгоритм, учитывающий все особенности системы. Также обязательным является визуальное представление данных о текущем состоянии и разработка пользовательского интерфейса.

Для опроса датчиков необходим контроллер, передающий собранные данные алгоритму.

3. Аппаратно-программные средства

Программа OpenSCADA была выбрана в качестве системы управления. Она представляет собой открытую SCADA систему, построенную по принципам модульности, многоплатформенности и масштабируемости. SCADA (Supervisory Control And Data Acquisition) в переводе: «Системы диспетчерского управления и сбора данных» является термином, который часто употребляется в сфере автоматизации технологических процессов. [1] Система OpenSCADA предназначена для: сбора, архивирования, визуализации информации, выдачи управляющих воздействий, а также других родственных операций, характерных для полнофункциональной SCADA системы. Основы системы является модульное ядро. В зависимости от того, какие модули подключены, система может выполнять как функции различных серверов, так и функции клиентов клиент-серверной архитектуры. Собственно, архитектура системы позволяет реализовывать распределённые клиент-серверные системы любой сложности. Для достижения высокого быстродействия, за счёт сокращения времени коммуникаций, архитектура позволяет объединять функции распределённых систем в одной программе.

Система OpenSCADA состоит из подсистем:

- Подсистема безопасности. Содержит списки групп и пользователей, обеспечивает проверку прав на доступ к элементам системы и т.д.
- Модульная подсистема баз данных. Обеспечивает доступ к базам данных.
- Модульная подсистема транспортов. Обеспечивает коммуникацию с внешней средой, посредством различных коммуникационных интерфейсов.
- Модульная подсистема коммуникационных протоколов обмена. Тесно связана с подсистемой транспортов и обеспечивает поддержку различных протоколов обмена с внешними системами.
- Модульная подсистема сбора данных (DAQ). Обеспечивает сбор данных от внешних источников: контроллеров, датчиков и т.д. Кроме этого, подсистема может предоставлять среду для написания генераторов данных (модели, регуляторы ...).

- Модульная подсистема архивов. Содержит архивы двух типов: архивы сообщений и архивы значений. Способ архивирования определяется алгоритмом, который заложен в модуле архивирования.
- Модульная подсистема пользовательских интерфейсов. Содержит функции пользовательских интерфейсов.
- Подсистема управление модулями. Обеспечивает контроль над модулями.
- Модульная подсистема специальных функций. Содержит функции, не вошедшие в остальные подсистемы. В настоящий момент к этим функциям относятся функции тестирования.

С учетом особенностей вычислительной системы возникла необходимость в нестандартном контроллере для замера температуры в 12 точках. Под эту задачу был выбран микроконтроллер Atmel семейства mega (atmega328) в связке с цифровыми термометрами. Для быстрой разработки остановились на отладочной плате Arduino, где уже на борту имеется вся необходимая обвязка и периферия. [2]

4. Разработка

4.1. Аппаратные средства

Для первого прототипа (1) был выбран цифровой термометр, работающий на шине 1-Wire. [3] Его возможно использовать вне помещения, так как производится он в том числе и в защищенном герметичном корпусе.

Передача собранных данных в программу осуществляется с использованием протокола MODBUS. [4] Это открытый протокол, основанный на архитектуре ведущий-ведомый (master-slave). В контроллере для передачи данных будут использоваться последовательные линии связи RS-485 [5], RS-232 [6].

В данном контроллере дополнительно используется преобразователь интерфейсов между RS-485, который позволяет передавать данные на большие расстояния, и RS-232 с уровнями TTL, с которым работает микроконтроллер.

Используемая библиотека для работы с MODBUS была доработана с учетом требований по работе через RS485, где требуется переводить преобразователь интерфейсов между режимами «приём» и «передача».

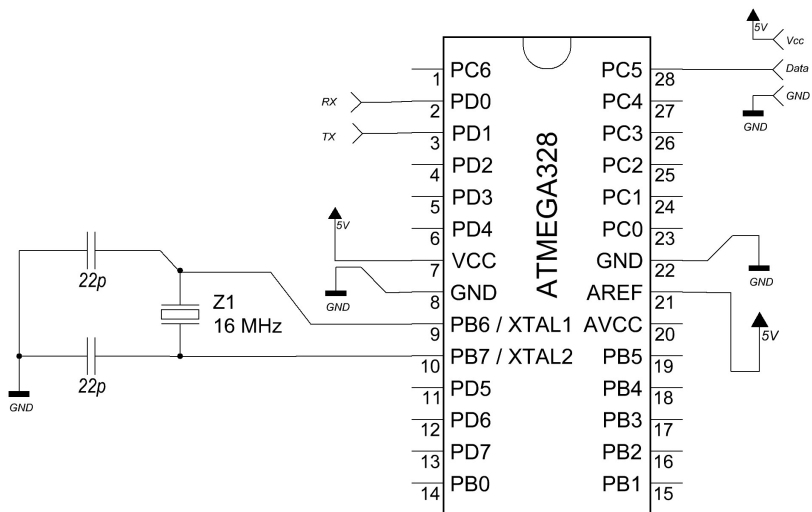


Рис. 1. Принципиальная схема контроллера 1-wire

В следующей версии (2) было решено использовать цифровые термометры на шине I^2C . Они опрашиваются гораздо быстрее и являются более точными. Так как адресация не позволяет разместить необходимое количество датчиков на одной шине, то было принято решение сделать 6 лучей с парой датчиков на них и при опросе, используя мультиплексор, соединять требуемый луч с шиной.

В процессе разработки от устройства потребовался дополнительный функционал для чтения логического уровня на входе микроконтроллера, что и было использовано для контроля уровня жидкости в системе.

Тестирование подтвердило работоспособность обоих решений.

4.2. Программные средства

На основе этого функционала разработано визуальное представление системы жидкостного охлаждения. Три главных подсистемы с которыми проводилась работа – это модуль сбора данных, модуль транспорта и модуль пользовательского интерфейса. В нашем распоряжении имелись датчики температуры в помещении и на насосах, 2 насоса для воды и масла, клапан, два кулера, датчик влажности,

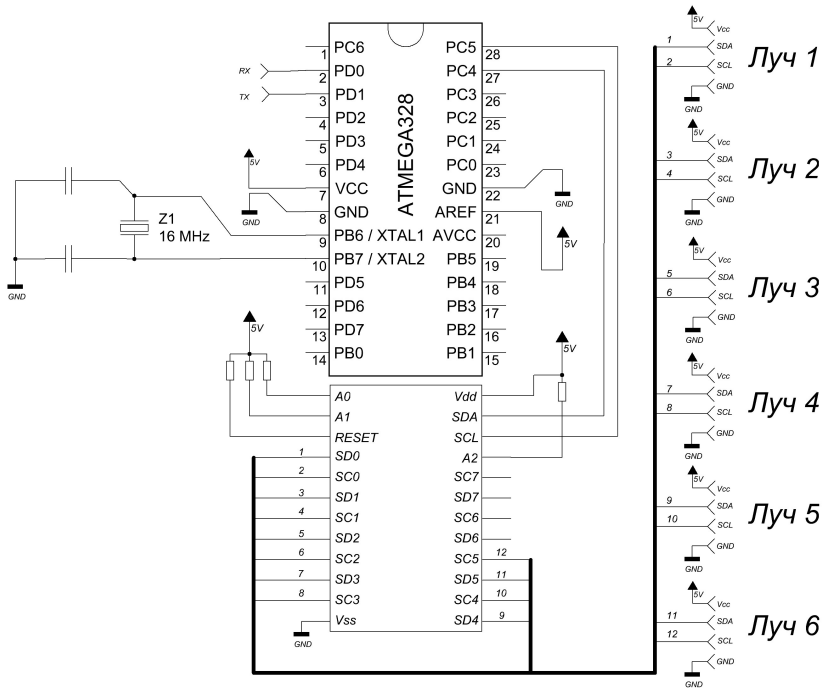


Рис. 2. Принципиальная схема контроллера I^2C

датчики давления, так же разработанный контроллер для измерения температуры на процессорах. Все эти части системы подключаются к OpenSCADA через модуль транспорта. Мы использовали 2 транспорта и 5 контролеров, и количеством параметров сопоставимых датчикам и управляющим элементы. Данные устройств считываются с определенных регистров, прописанных в документации устройства, при помощи протокола MODBUS. Модуль DAQ позволяет читать разные регистры, для работы нам понадобились обычный (R) для сбора данных и передачи их обратно, и регистр типа (C), позволяющий управлять элементом. Из прочитанного регистра значения записываются в переменную для последующей работы с ней. Каждый регистр обладает синтаксисом написания. Чтобы кроме чтения, мы могли отсылать данные в контроллер, устанавливаем регистр в значение `rw – чтение\запись`. Для каждого типа данных был создан шаблон,

позволяющий получать данные такими, какими мы их хотим видеть. В нашем случае датчики фирмы Owen, для вычисления температуры, выдавали значения с увеличением в 10 раз, поэтому был создан шаблон для перевода чисел в нужное значение. Сам шаблон представляет собой код программы, позволяющий из полученных значений с регистра, путем передачи их через переменную, обработать и записать в новую переменную. Все написание программ в OpenSCADA происходит на JS подобном языке JavaLikeCal. Его отличие в том, что он имеет свой набор функций для работы, по мимо основным возможностям JavaScript. Дольше идет работа с логическим модулем. Его задача состоит в том, чтобы обеспечить связь между значением, полученным через регистр, и шаблоном. Путем простых связей к переменным мы получаем то значение, которое нам надо. Создание визуального представления – мнемосхемы, это вторая главная часть программы OpenSCADA. Для этого используется подсистема пользовательского интерфейса, в частности модуль Vision. Область работы – это лист, как в любом графическом редакторе, с набором инструментов. В данном случае набором инструментов служат виджеты, подключаемых библиотек, с набором атрибутов. Мы создаем новый проект, в котором начинаем собирать схему, которую нам нужно визуализировать.

Есть несколько видов виджетов:

- (1) Графические – являются элементами схема с простым набором атрибутов (цвет, линии, форма и т.д.).
- (2) Аналоговые напрямую подключаются к нашим данным и выводят те значения, которые мы получаем; имеют больший набор атрибутов (значение, шрифт значение, цвет поля, сигнализация, пределы, включение виджета и т.д.).
- (3) Оформление и управление – текстовые сообщения, формы (кнопка, поля редактирование, выпадающий список и т.д.), графики, страница архивации, протокол документации и т.д.

На основе атрибутов производится управление элементами (цвет при сигнализации или изменения значения аналогового виджета для передачи в регистр, или выключение определенного элемента). Помимо графической части существует поле обработки. Основным назначением обработки является написание кода программы, привязка его к элементам мнемосхемы и их атрибутов, а также связка основных частей схемы с регистрами MODBUS. Благодаря этому происходит полный мониторинг системы и возможность следить за показателями датчиков. Был разработан алгоритм, далее он был преобразован в код

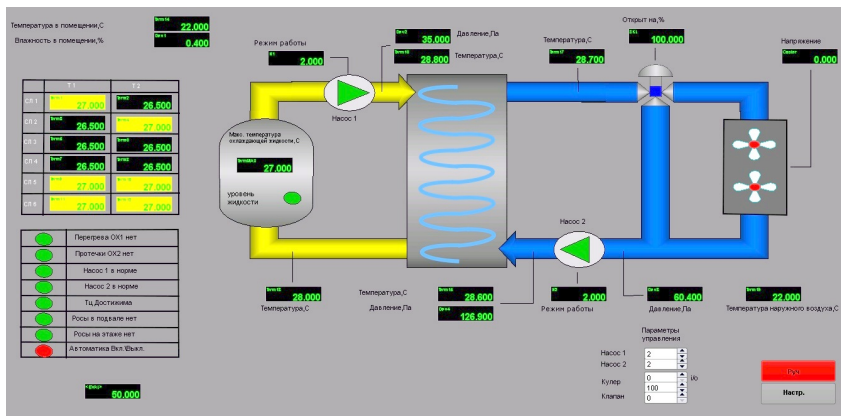


Рис. 3. Мнемосхема в OpenSCADA

≈ 1500 строк для управления системой охлаждения. В частности были реализованы режимы переключения скоростей насосов 1 и 2, установлены сигнализации и аварийные режимы, в момент некорректной работы, а так же контролирование температурного режима внутри бака. На схеме (3) мы видим 4 датчика температуры на насосах, 3 датчика давления, 12 датчиков температуры разработанного контроллера для вывода температуры в баке, скорости насосов и кулеров, состояние клапана, а также температуру и влажность помещения. На мнемосхеме расположены две кнопки: для переключения автоматического ручного управления и кнопка настроек. Автоматический режим – это алгоритм, позволяющий корректно работать системе и поддерживать температуру в пределах 50 градусов без участия оператора. Ручной режим – позволяет напрямую управлять двумя насосами, выставлять 3 типа скоростей, управлять положением драйкулера (2 вентилятора и клапан). Настройка позволяет без вмешательства в код программы выставить коэффициенты для оптимальной работы системы.

5. Заключение

Основным требованием при разработке было удержание температуры внутри системы при максимальной нагрузке в пределах 50°C. В результате нашей работы был получен результат, превосходящий

требования на 12°C за счет разработанного алгоритма и аппаратно-программного комплекса.

Список литературы

- [1] OpenSCADA: Документация. URL <http://oscada.org/ru/glavnaja/dokumentacija/> ↑ 40.
- [2] ArduinoBoardNano. URL <http://arduino.cc/en/Main/ArduinoBoardNano> ↑ 41.
- [3] Ведущий однопроводной интерфейс 1-Wire компании Dallas. URL <http://www.gaw.ru/html.cgi/txt/app/micros/avr/avr318.htm> ↑ 41.
- [4] MODBUS Protocol Specification. URL http://www.MODBUS.org/docs/MODBUS_Application_Protocol_V1_1b3.pdf ↑ 41.
- [5] Guidelines for Proper Wiring of an RS-485 (TIA/EIA-485-A) Network URL <http://www.maximintegrated.com/en/app-notes/index.mvp/id/763> ↑ 41.
- [6] The RS232 STANDARD. URL http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html ↑ 41.

Специфика статьи: Развитие информационно-вычислительных технологий, Алгоритм, Электронное устройство, Языки программирования, Программы для разработки технических устройств.

Научный руководитель:

к.т.н. С. А. Амелькин

Об авторах:

Сергей Валерьевич Карпеш

УГП имени А. К. Айламазяна, 3М21

e-mail:

Karpesh.sergey@gmail.com

Алексей Анатольевич Петров

УГП имени А. К. Айламазяна, 3М21

e-mail:

reykon2010@yandex.ru

Пример ссылки на эту публикацию:

С. В. Карпеш, А. А. Петров. «Разработка систем автоматического управления погружной системой жидкостного охлаждения высокопроизводительных вычислительных систем». *Научно-информационные технологии: Труды XIX Молодежной научно-практической конференции SIT-2015. УГП имени А. К. Айламазяна*. — Переславль-Залесский: Изд-во «Университет города Переславля», 2015 с. 39–48.

URL

<https://edu.botik.ru/proceedings/sit2015.pdf>

Sergej Karpesh, Aleksey Petrov. *Development of automatic control systems immersion liquid cooling system of high-performance computing systems.*

ABSTRACT. In this article we consider the problem of building a system of automatic control of immersion liquid cooling system.

Key Words and Phrases: The monitoring system, MODBUS, AVR, OpenSCADA, JS.

Sample citation of this publication:

Sergej Karpesh, Aleksey Petrov. “Development of automatic control systems immersion liquid cooling system of high-performance computing systems.”. *Science-intensive information technologies: Proceedings of XIX Junior R&D conference SIT-2015. Ailamazyan Pereslavl University.* — Pereslavl-Zalesskiy: Pereslavl University Publishing, 2015 pp. 39–48. (*In Russian.*)

URL

<https://edu.botik.ru/proceedings/sit2015.pdf>