

- 2) `main_menu_view` — экран главного меню. Пользователю предлагается выбрать один из трех блоков: «Уроки», «Тесты», «Словарь»;
- 3) `lessons_menu_view` — экран меню уроков (список имеющихся уроков);
- 4) `lesson_view` — экран с текущим уроком;
- 5) `tests_menu_view` — экран меню тестов (список тестов по урокам);
- 6) `test_view` — экран с текущим тестом;
- 7) `finish_test_view` — экран с результатами теста;
- 8) `word_groups_view` — экран словаря. На нем отображены группы слов, рассортированные по урокам. Также здесь возможно добавление пользовательских групп;
- 9) `group_words_view` — экран слов, относящихся к какой-либо группе.

Компонент `Presenter` состоит из:

- 1) `MainActivity` — основная активность приложения;
- 2) `MainMenuPresenter` — обработка действия в главном меню;
- 3) `LessonPresenter` — обработка действий пользователя при выборе урока;
- 4) `TestListPresenter` — обработка действий пользователя в меню списка тестов;
- 5) `TestPresenter` — обработка действий пользователя во время прохождения теста;
- 6) `FinishTestPresenter` — обработка завершения теста (а также вывод результатов);
- 7) `WordPresenter` — обработка действий пользователя во время просмотра словаря.

Компонент `Model` состоит из:

- 1) `Tests` — хранение тестов;
- 2) `Words` — хранение слов в словаре.

Ресурсы приложения находятся в каталоге `res`. В папке `drawable` будем хранить стили для элементов экрана (`button`, `textView` и т. п.), а также картинки для фона. Слои пользовательского интерфейса помещены в папку `layout`. В папке `values` находится папка `strings`, в которой прописывается весь текст, используемый в приложении (строки хранятся в переменных). Сделано это для удобства перевода: не надо искать по всему коду отдельные слова. Если приложение необходимо перевести на другой язык, то достаточно перевести только строки в файле `strings`.

Заключение. Разработанное мобильное приложение будет объединять разрозненные учебные материалы и представлять единый комплекс для изучения японского языка. Оно будет интересно людям, у которых нет возможности посвящать изучению иностранного языка несколько часов в день, но есть возможность сделать это во время обеденного перерыва или сидя в метро.

Список цитируемых источников

1. Различия между MVVM MV*-паттернами [Электронный ресурс]. — Режим доступа: <https://habr.com/ru/company/mobileup/blog/313538/>. — Дата доступа: 20.04.2020.

УДК 004.35

А. В. Корсак, М. А. Вареник

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

МИКРОКОНТРОЛЛЕРНЫЙ РЕГУЛЯТОР СКОРОСТИ ВРАЩЕНИЯ ВЕНТИЛЯТОРА ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА В ЗАВИСИМОСТИ ОТ ТЕМПЕРАТУРЫ

Введение. Во всех современных компьютерах используется система охлаждения. Система охлаждения компьютера — набор средств для отвода тепла от нагревающихся в процессе работы компьютерных компонентов.

Системы охлаждения делят на две группы: пассивные и активные [1].

К пассивным относят радиаторы, принцип их работы основан на непосредственной передаче тепла от нагревающегося компонента за счёт теплопроводности материала или с помощью тепловых трубок.

К активным относят вентиляторы, которые обдувают части компьютера. Совокупность вентилятора и радиатора называют кулером. Преимущественно они устанавливаются на центральный и графический процессоры.

Основная часть. В данном исследовании основной задачей ставилась разработка микроконтроллерного регулятора скорости вращения вентилятора компьютера в зависимости от температуры.

Для разработки устройства использовался микроконтроллер фирмы ATMEL, семейство AVR, ATmega8 [2]. Выбор был сделан в пользу данного микроконтроллера, так как устройство должно производить некоторые вычисления и проверки, а также передавать и принимать данные с датчика тепла. Микроконтроллер ATmega8 имеет внутрисистемную самопрограммируемую FLASH-память, размер которой составляет 8 Кбайт, что достаточно для размещения прошивки.

Все микроконтроллеры AVR построены по так называемой Гарвардской архитектуре, т. е. используется раздельная адресация памяти программ и памяти данных. Преимущества этой архитектуры заключаются в повышенном быстродействии, например, ATmega выполняет одну инструкцию за один тактовый импульс, т. е. при частоте 16 МГц микроконтроллер выполняет 16 миллионов операций в секунду.

Для разработки прошивки микроконтроллера использовалась программа CodeVisionAVR C Compiler, а симуляция проводилась с использованием программы Proteus.

Для разработки схемы устройства (рисунок 1) использовались следующие элементы: микроконтроллер ATmega8, драйвер L293D и двигатель, датчик температуры DS18B20, ЖК-дисплей.

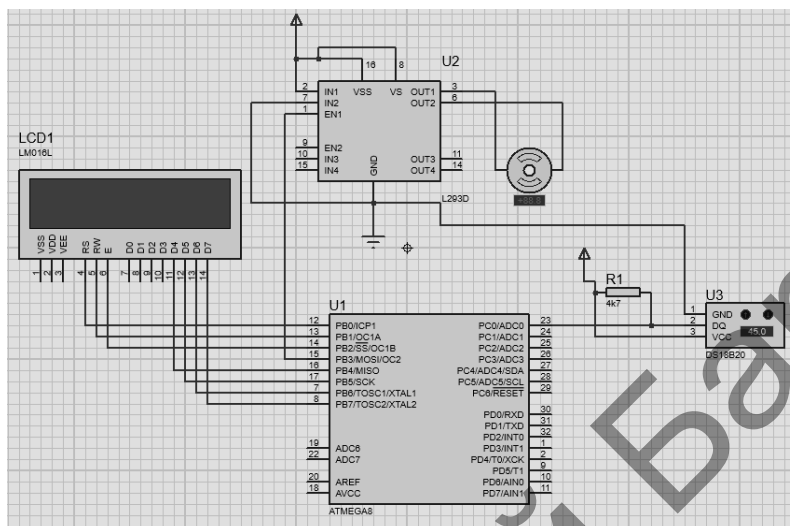


Рисунок 1 — Принципиальная схема устройства

Разработанное устройство позволяет изменять обороты кулера в зависимости от результата измерения температуры на датчике. При запуске устройства считываются данные температуры, в зависимости от них устанавливаются обороты на кулере.

В устройстве на ЖК-дисплее отображаются показания датчика температуры и скорость вращения в проценте от максимальной скорости:

- 1) если температура ≤ 25 °C, скорость устанавливается в размере 20 % от максимальной;
- 2) если температура находится в диапазоне 25—35 °C, скорость устанавливается в размере 35 % от максимальной;
- 3) если температура находится в диапазоне 35—45 °C, скорость устанавливается в размере 50 % от максимальной;
- 4) если температура находится в диапазоне 45—55 °C, скорость устанавливается в размере 70 % от максимальной;
- 5) если температура > 55 °C, устанавливается полная максимальная скорость вращения.

На рисунке 2 показана симуляция работы устройства.

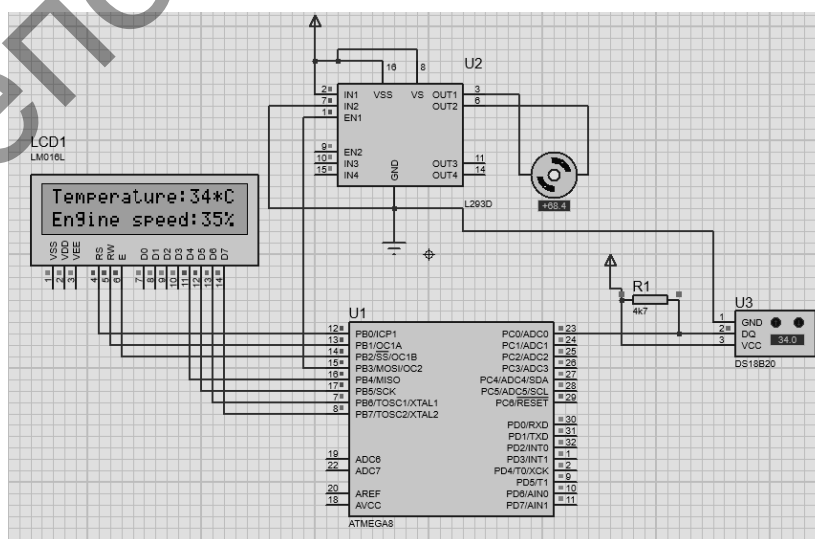


Рисунок 2 — Пример работы устройства

Заключение. В результате исследования был разработан микроконтроллерный регулятор скорости вращения вентилятора персонального компьютера, в зависимости от температуры. Данная система может быть применена в компьютере для охлаждения важных частей, а также в других устройствах, где требуется регулирование скорости вращения кулера в зависимости от температуры.

Список цитируемых источников

1. Компьютерная грамотность для начинающих [Электронный ресурс]. — Режим доступа: <https://www.pc-school.ru/sistema-oxlazhdeniya-kompyutera/>. — Дата доступа: 29.04.2020.
2. Баранов, В. Н. Применение микроконтроллеров AVR / В. Н. Баранов. — М. : Додэка-XXI, 2004. — 288 с.

УДК 004.75

М. С. Кузнецов

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Вологодский государственный университет», Вологда, Российская Федерация

РЕАЛИЗАЦИЯ РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ НА ЯЗЫКЕ PYTHON С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ DOCKER

Введение. Распределённые вычисления представляют собой способ решения трудоемких вычислительных задач с использованием нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему [1].

Одно из первых упоминаний распределённых вычислений относится к 1973 году. Сотрудники научно-исследовательского центра Хехох PARC Джон Шох и Джон Хапп написали программу, которая рассылала себя по другим работающим компьютерам через локальную сеть PARC.

Впоследствии в связи с развитием и ростом количества персональных компьютеров распределённые вычисления стали использоваться всё более и более широко. Так, в конце 1980-х годов Арьен Ленстра и Марк Менес написали программу для факторизации длинных чисел. Она рассылала задания на компьютеры участников по электронной почте и таким же образом принимала ответы.

Ещё одним значимым событием было создание проекта SETI@Home (Search for Extra-Terrestrial Intelligence at Home) для поиска внеземного разума путём анализа данных с радиотелескопов, в том числе на домашних компьютерах участников. Данный проект был запущен в 1999 году и остановлен в 2020-м. Эта распределённая система была построена на платформе BOINC, созданной в университете Беркли.

В дальнейшем разработки по созданию различных распределённых систем активно продолжались, в настоящее время они применяются в самых различных областях. В частности, распределённые вычисления широко используются для математических задач. Типичным примером является факторизация чисел (разложение их на произведение простых множителей).

Ещё одной важной областью применения распределённых вычислений является обработка больших данных с использованием методов машинного обучения и Data Mining. В качестве языка программирования для этой цели в последние годы на лидирующие позиции выходит язык Python. По состоянию на март 2020 года, согласно рейтингу TIOBE, Python находится на третьем месте, хотя ещё в 2015 году занимал лишь седьмое.

Одной из известных проблем языка Python является относительно низкая производительность в сравнении с компилируемыми языками, такими как C++. Данный недостаток является дополнительным поводом применять параллельное и распределённое программирование в процессе разработки.

Также во второй половине второй декады XXI века стали набирать популярность такие технологии, как контейнеризация и микросервисы. Благодаря программному обеспечению Docker и изолированным пространствам исполнения стало возможным относительно легко запускать на одном сервере несколько приложений полностью изолированно друг от друга и с собственными настройками среды выполнения. В настоящее время технология Docker используется всё более широко и для самых разных целей, в том числе при разработке программного обеспечения: она позволяет проводить различные эксперименты с зависимостями программного обеспечения, не «засоряя» основную систему.

Основная часть. Основным инструментом кластеризации для Docker носит название Docker Swarm. Он позволяет объединять узлы в единое кластерное пространство и распространять контейнеры по этому кластеру. Представленная в данной статье разработка основывается на изолированных Docker-контейнерах — «исполнителях».