

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики
и информационных технологий

**СОЗДАНИЕ СТЕНДА ДЛЯ УДАЛЕННОГО КОНТРОЛЯ И
МОНИТОРИНГА ЛАБОРАТОРНЫХ РАБОТ НА БАЗЕ ARDUINO**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 «Информатика и вычислительная техника»
факультета компьютерных наук и информационных технологий
Орлова Дмитрия Сергеевича

Научный руководитель

д.т.н., профессор

В.А. Твердохлебов

подпись, дата

Зав. кафедрой

к. ф.-м.н., доцент

Л.Б. Тяпаев

подпись, дата

Саратов 2016

ВВЕДЕНИЕ

В современном мире большую популярность приобретают отладочные платы на базе микроконтроллеров AVR. Микроконтроллеры AVR семейства “tiny” используются в разработке: датчиков, пультов управления, электронных игрушек, детекторов дыма, многих других миниатюрных устройств. С помощью плат семейства "mega", "xmega" и 32-bit AVR создаются более сложные системы: сложная бытовая техника, дистанционные системы управления, спутниковые навигационные системы, многие другие многофункциональные устройства [2].

Одним из популярных решений для этих задач является проект Arduino, предоставляемый в виде программно-аппаратного комплекса. Среда Arduino IDE не имеет встроенного отладчика, что создает большие неприятности при поиске ошибок и сопровождении проектов. Существуют эмуляторы, решающие эту проблему, но у такого подхода есть значительные недостатки: ограниченный набор возможностей, оторвано от реальности, не решает реальные физические проблемы.

Актуальность работы заключается в обеспечении простоты и наглядности процесса отладки программ для устройств на базе Arduino.

Целью работы является создание стенда верификации поведения микроконтроллерных систем для удаленного контроля и мониторинга лабораторных работ.

Основные задачи:

- разработка архитектуры системы управления сборкой, запуском и публикацией результатов запуска через веб-интерфейс
- разработка алгоритма сборки и запуска лабораторных работ
- освоение механизмов идентификации USB-serial устройств
- сборка стенда на базе Arduino для загрузки программ лабораторных работ
- создание механизма трансляции поведения стенда в сеть Internet
- написание сценария на языке Bash, осуществляющего сборку и загрузку программ в стенд

Работа состоит из введения, трех глав, заключения, списка литературы и приложения. Первая глава «Детализация процесса отладки и связанные с ним проблемы», вторая глава «Содержательное описание» и третья глава «Практическая реализация стенда»

Основное содержание работы

1. Детализация процесса отладки и связанные с ним проблемы. Отладка встраиваемых систем является трудоемкой задачей. Основной причиной данной трудоёмкости является проблема извлечения требуемой информации о выполнении отлаживаемой программы.

Одно из возможных решений вышеозначенной проблемы - подключение специального оборудования к встраиваемой системе. Такое оборудование позволяет считывать содержимое памяти встраиваемой системы и анализировать состояние системы в точках останова [1].

Возможна отладка встраиваемых систем с помощью эмуляторов. К достоинствам такого подхода можно отнести наличие отладчиков и средств вывода информации, но такие системы являются лишь имитаторами. Отлаживаемая программа может успешно выполняться в эмуляторе, однако в реальных условиях работать не будет. По этой причине, такой способ отладки - временное решение. Программные ошибки, незамеченные системой моделирования, могут быть обнаружены на реальном оборудовании. В этом скрыта главная проблема - исправление ошибок, выявленных не на этапе тестирования, а в процессе работы, обходится гораздо дороже. В случае встраиваемых систем, исправление ошибок подразумевает возврат и модификацию устройств с этой системой. Стоимость таких возвратов может достигнуть очень больших величин и привести к разорению [1].

Процесс проектирования цифровых устройств неразделимо связан с понятием контроля качества. Существуют три фундаментальные концепции контроля качества программного обеспечения встраиваемых систем: отладка, верификация и тестирование.

Отладка - процесс анализа и возможных исправлений программы, на основе ее данных, памяти и понимания ошибок разработчиком [3].

Верификация - проверка соответствия данных, создаваемых в ходе разработки и сопровождения ПО другим, ранее созданным или

используемым в качестве исходных данных, а также соответствие этих данных и процессов их разработки правилам и стандартам [4].

Тестирование - проверка соответствия реального и ожидаемого поведения программы, осуществляемая на конечном наборе заготовленных тестов. На этом этапе разработка программы может быть уже завершена. Поэтому, исследуется реакция готовой системы на различные события [3].

Множество методов верификации делится на две категории: динамические и статические (формальные) [5].

К динамическим методам можно отнести эксперименты над разрабатываемым проектом на различных этапах тестирования. Тестируемый объект может являться как физическим прототипом, так и моделью [5].

Формальная верификация - доказательство соответствия предмета верификации его формальному описанию [5].

Одним из наиболее популярных современных методов верификации является метод доказательства эквивалентности моделей. Он основан на сравнении результатов работы разрабатываемой модели с другой моделью. Сложность заключается в верности работы другой модели, принимающейся за эталон.

Одним из важнейших вопросов при разработке встраиваемых является идентификация плат.

Касательно Arduino, чтобы загрузить программу в плату на выполнение, следует в Arduino IDE выбрать порт, к которому подключен микроконтроллер. При подключении нескольких плат возникает проблема, т.к. в Arduino IDE нет подробной идентификации устройств.

Существует несколько способов получения уникального идентификатора встраиваемой системы [7]:

- Чипы, разработанные шотландской компанией FTDI, обладают уникальным запрограммированным серийным номером. Доступ к нему можно получить только со стороны компьютера, к которому данный чип подключен [7].

- Некоторые из чипов, работающие через USB-порт (ATmega8U2/16U2/32U2) имеют уникальный идентификатор в своей сигнатуре. Доступ можно легко получить с любой из сторон соединения [7].
- У некоторых чипов, работающих через USB, но не имеющих сигнатурного уникального идентификатора, все же можно получить доступ к идентификатору, программируемому через специальную USB-среду (например, LUFA), используемую в загрузчике чипа [7].
- Модификация стандартного загрузчика платы, изменения механизма идентификации [7].

В случае, когда микроконтроллер вообще не имеет никакого идентификатора, можно прибегнуть к нестандартным вариантам. Одним из таких является предварительное программирование EEPROM-памяти устройства.

Arduino-совместимые платы включают в себя EEPROM, небольшое энергонезависимое перепрограммируемое ПЗУ. Его размер зависит от типа используемого микроконтроллера. Для работы с этой памятью существует специальная библиотека, позволяющая записать байт в EEPROM и прочитать его оттуда [6].

В Linux существует набор механизмов, создающий символические ссылки, позволяющий уникально идентифицировать подключенные устройства. Механизм работает таким образом, что при первом подключении устройству, согласно правилам UDEV, задается некий идентификатор. Ключевой особенностью является то, что полученный при первом подключении идентификатор присваивается устройству навсегда.

Для загрузки проектов Arduino в платы через консольный режим существует два наиболее популярных инструмента: `ino` от группы разработчиков из России `Amperka` и `Arturo` от разработчика-энтузиаста. Проект `ino` хоть и является популярным в своей среде, но обладает рядом

ошибок и недочетов. Так как Amperka с недавнего времени отказались от дальнейшей поддержки проекта, использование его было принято считать неуместным. Arturo же в свою очередь является лишь форком ino, однако поддерживаемый разработчиком на сегодняшний день, в следствие чего все ошибки исправляются очень оперативно. Arturo и послужил средой для консольной сборки и загрузки проектов в данной работе.

2. Содержательно описание. Была разработана архитектура системы, приведенная на рисунке 1.

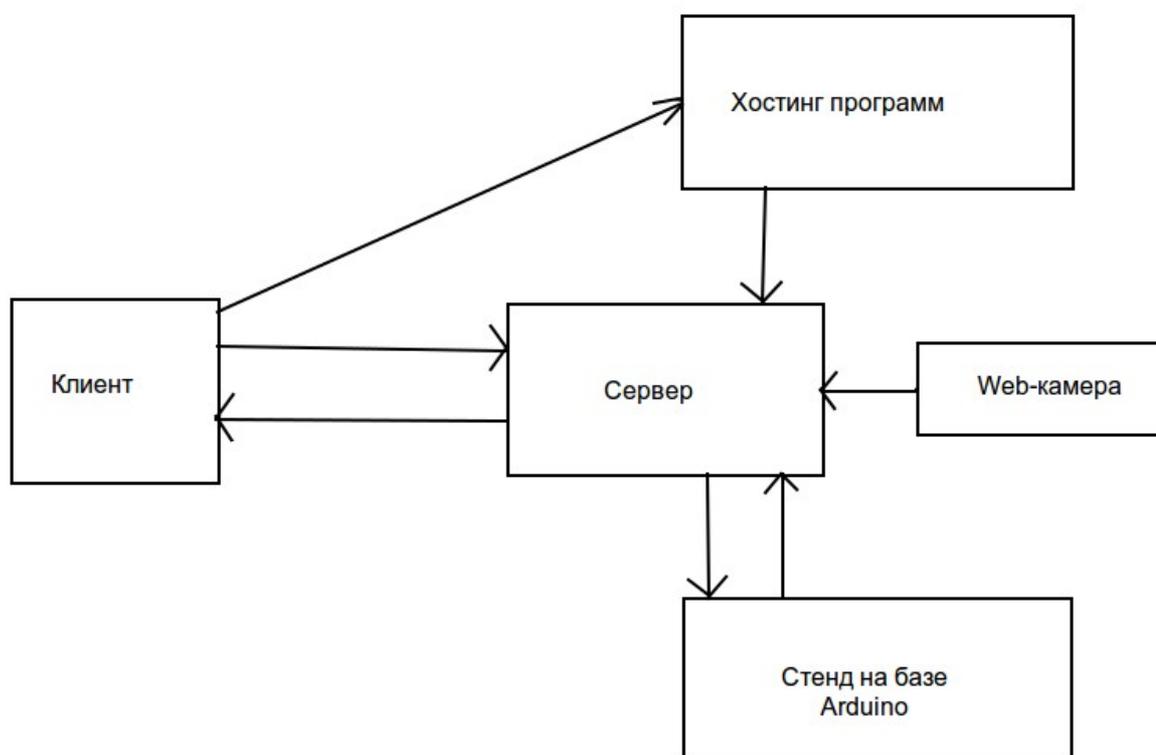


Рисунок 1.

Центральным элементом системы является сервер, к которому подключен стенд, состоящий из двух отладочных плат, одна из которых выполняет роль исполнителя, а другая верификатора. Также, в состав стенда входит набор соединительных элементов, уникальный для каждой отдельной задачи. Информация о работе стенда отправляется на сервер.

Помимо стенда, к серверу подсоединена веб-камера. Изображение с камеры захватывается сервером и транслируется на страницу, доступную по сети Internet.

Сервер выкачивает код проектов с хостинга программ. Хостинг состоит из следующих компонентов: репозиторий учащихся, репозиторий с тестовыми прошивками, репозиторий с пустыми прошивками.

В репозитории учащихся находятся проекты, предназначенные для проверки их работоспособности на стенде. Репозиторий с тестовыми прошивками создается преподавателем. Каждой лабораторной поставлен в соответствие единственный верный тест. Репозиторий с пустыми прошивками предназначен для хранения программ, выполняющихся на стенде в то время, когда ни один из проектов репозитория учащихся не проходит процесс сборки, загрузки, либо выполнения.

Клиентская сторона представлена в системе следующими элементами: сторона, загружающая код в репозиторий учащихся, механизм обращения к серверу и механизм демонстрации трансляции.

Процесс, проходящий на стороне, загружающей код лабораторных работ в репозиторий учащихся, выполняется до начала работы системы. Механизм обращения к серверу отвечает за запуск сценария, выкачивающего код из репозитория учащихся, сборку проекта и его загрузку в стенд. На стороне клиента, демонстрация трансляции представляет собой инструмент, с помощью которого осуществляется подключение к запущенному на сервере механизму, выполняющему захват видео с подключенной к серверу веб-камере, создание видеопотока и его трансляции в сеть Internet.

На сервере помимо программы, осуществляющей видеотрансляцию, запущена программа, представляющая собой порядок выполнения действий сервера, необходимых для корректной работы всей системы.

Сценарий работы системы представляет собой следующий набор действий:

- Код предназначенной для отладки программы загружается в репозиторий учащихся.
- Со стороны клиента запускается скрипт-сценарий, согласно которому, осуществляется скачивание исполняемой программы из репозитория учащихся и соответствующей тестовой программы из репозитория с тестовыми прошивками на сервер.
- Во избежании проблем верификации, сначала происходит сборка тестовой программы и ее загрузка в плату-верификатор, а затем сборка и загрузка исполняемой программы в плату-исполнитель.
- Механизм захвата видеопотока с веб-камеры транслирует видео на страницу в сети Internet. Клиентская сторона подключается к этой странице и визуально наблюдает за ходом работы стенда.
- В течение некоторого времени загруженные программы выполняются на стенде. Клиентская сторона в онлайн режиме анализирует поведение стенда.
- По окончании времени, отведенного за работу загруженной клиентской программы, в стенд загружаются пустые прошивки, сообщая серверу и клиенту и готовности к загрузке новой программы из репозитория учащихся.

3. Практическая реализация стенда. Был собран стенд, состоящий из двух плат Arduino UNO, шины, набора светодиодов, набора резисторов и набора соединительных элементов. В реальности стенд выглядит как на рисунке 2.

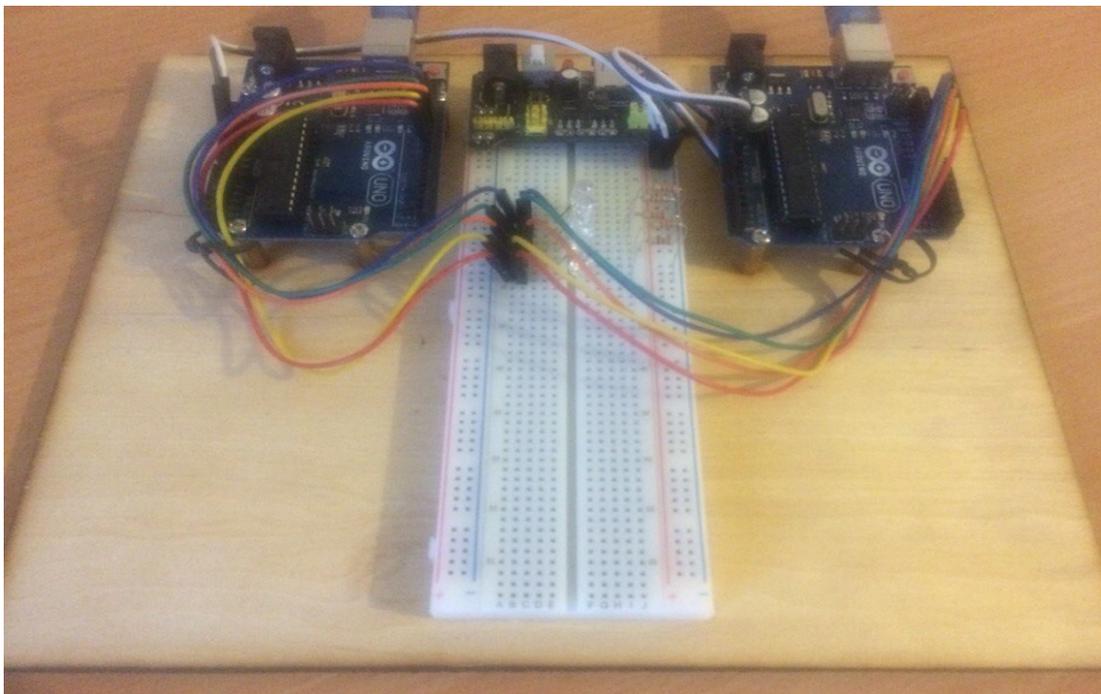
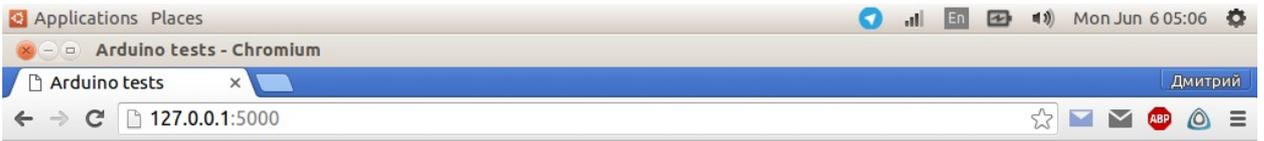


Рисунок 2.

В данном виде стенд полностью готов к работе. При подключении к компьютеру с помощью кабелей, можно осуществлять загрузку программ в платы. Сценарий работы системы написан на Bash, осуществляет настройку работы системы, собирает и загружает доступные проекты для Arduino. Трансляция с сеть интернет была релизована с помощью консольной утилиты `cvlc` и ресурса-одностраничника, написанного на языке Python и его микрофреймворке Flask.

На рисунке 3 изображена страница, открывающаяся при переходе на адрес ресурса, транслирующего процесс работы стенда.



Arduino tests video

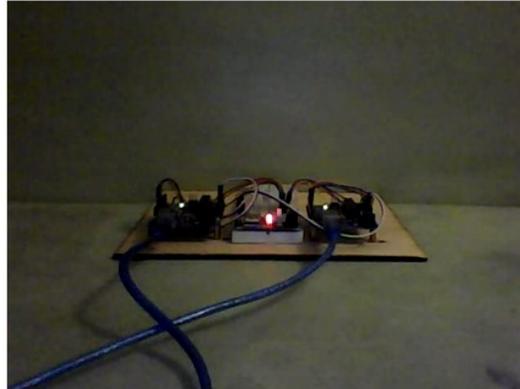


Рисунок 3.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были исследованы особенности отладки встраиваемых систем, возможности идентификации USB-serial устройств.

Был создан стенд, состоящий из двух плат Arduino UNO и набора деталей. Для управления стендом был написан сценарий обнаружения проектов, их сборки и загрузки в платы. Для удаленного наблюдения была организована трансляция веб-камеры с возможностью просмотра через браузер. Для решения этой задачи потребовалось создать веб-ресурс на языке Python.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Адам Колава. Методы предотвращения и обнаружения ошибок во встраиваемых системах.[Электронный ресурс. Перевод] — <http://www.mka.ru/?p=42050> — (дата обращения: 11.05.2016)
- 2 Отладка кода Arduino (AVR). [Электронный ресурс] — <https://geektimes.ru/post/270868/> — (дата обращения 13.05.2016)
- 3 Эдмунд Кларк, Дэвид Гарлан. Verification Tools for Embedded Systems. — Carnegie Mellon University, 2000 — 25 с.
- 4 Томас Рейнбахер. Introduction to Embedded Software Verification. — University of applied science, 2008 — 11 с.
- 5 Горохов И. В., Зинченко Ю.Е. Проблемы верификации системна-кристалле на базе FPGA на различных этапах проектирования. — Донецк, ДонНТУ, 2012. — 591-594 с.
- 6 Arduino EEPROM Library [Электронный ресурс] — <https://www.arduino.cc/en/Reference/EEPROM> — (дата обращения: 16.05.2016)
- 7 How can i get a unique ID for all my Arduino Boards [Электронный ресурс] — <http://arduino.stackexchange.com/questions/279/how-can-i-get-a-unique-id-for-all-my-arduino-boards> — (дата обращения: 10.02.2016)