DOI: 10.17117/na.2015.09.804

Поступила (Received): 29.09.2015

<http://ucom.ru/doc/na.2015.09.804.pdf>

Сёмочкин А.Н., Хаданович С.С.
Сетевой драйвер устройств на базе Arduino как основа
программной архитектуры антропоморфного робота

Semochkin A.N., Khadanovich S.S.
Arduino devices network driver as the basis of
anthropomorphic robot software architecture

В статье описаны сетевой драйвер устройств на базе плат прототипирования Arduino и его позиционирование в качестве ядра программной архитектуры антропоморфного робота, используемого в рамках исследований технологий человеко-машинных интерфейсов на основе распознавания образов

Ключевые слова: робототехника, драйвер, человеко-машинный интерфейс

The article considers the design of network driver for controlling Arduino prototyping board based devices and it's application as a core of software architecture of anthropomorphic robot used in research purposes for developing image recognition based human machine interface

Key words: robotics, device driver, human-machine interface

Сёмочкин Александр Николаевич

Кандидат физико-математических наук, доцент
Благовещенский государственный педагогический университет
Амурская область, г. Благовещенск, ул. Ленина, 104

Semochkin Alexander Nikolaevich

Candidate of Physico-mathematical Sciences,
Associate Professor
Blagoveshchensk state pedagogical university
Amur Region, Blagoveshchensk, Lenin st., 104

Хаданович Сергей Сергеевич

Инженер
Благовещенский государственный педагогический университет
Амурская область, г. Благовещенск, ул. Ленина, 104

Khadanovich Sergey Srgееvich

Engineer
Blagoveshchensk state pedagogical university
Amur Region, Blagoveshchensk, Lenin st., 104

Данная работа выполняется в рамках исследований технологий человеко-машинных интерфейсов в части способов организации бесконтактного взаимодействия пользователя с компьютерной системой на основе распознавания объектов, жестов или движений пользователя, а также на основе распознавания и синтеза звуков, речи. В ходе исследования была разработана сеть адаптивных информационных терминалов [3], предоставляющих различные услуги в рамках образовательного процесса в вузе. Доступ к различным сервисам информационного терминала осуществляется тремя способами: с помощью сенсорной панели через визуальное меню, с помощью распознавания QR-кода, и с помощью распознавания лица пользователя системы [5]. Все пользовательские режимы сопровождаются голосовыми комментариями, воспроизводимыми синтезатором русской речи [4]. В силу технических и психологических

ограничений описанного интерфейса, а также для исследований границ применимости разрабатываемых математических моделей [2] в задачах человеко-машинных интерфейсов и моделирования общих с человеком видов и форм коммуникации был разработан экспериментальный образец антропоморфного робота [1]. Для реализации электромеханических узлов робота был использован открытый проект InMoov [8]. Для организации управления подвижными узлами робота использованы открытые платы прототипирования типа Arduino и Raspberry Pi. Платы Arduino спроектированы на основе микроконтроллеров типа ATmega, что делает невозможным реализовывать более сложный алгоритм, чем простая обработка данных с датчиков и управление периферийными устройствами, например, алгоритмы обучения в задачах распознавания образов и т.д. Таким образом, проблема управления исполнительными устройствами на основе открытых плат прототипирования типа Arduino посредством сети Ethernet является наиболее обоснованной и актуальной в вопросах проектирования программно-аппаратных комплексов в области робототехники.

Аппаратная платформа антропоморфного робота

Разработка компонентов программно-аппаратной платформы антропоморфного робота предполагает использование общедоступных материалов и аппаратных узлов, а также свободного программного обеспечения. Основные узлы робота распечатаны на 3D принтерах MyRiwell и Prusa i3. Исходный проект InMoov был изменен в целях адаптации к имеющимся в распоряжении авторским технологиям. В частности, для возможности бинокулярного зрения установлены две IP MJPEG камеры. На платформу установлена звуковая система, способная воспроизводить звуковые сообщения с помощью синтезатора русской речи[4]. В кистях рук реализована система датчиков Холла, позволяющих ощущать сопротивление при сгибании пальцев. Управление сервоприводами и датчиками осуществляется с помощью плат Arduino и одноплатных компьютеров Raspberry Pi. На борту робота установлен Wi-Fi роутер, подключенный к одноплатному компьютеру Raspberry Pi. Таким образом, становится возможным удаленное управление сервоприводами и датчиками по протоколу TCP. Схема подключения представлена на рисунке 1.

Рис. 1. Схема подключения

Программная архитектура антропоморфного робота

Ядром программной архитектуры служит разработанный авторами сетевой драйвер устройств[6], представляющий собой приложение на платформе Java и устанавливающийся на компьютер под управлением GNU/Linux (в данном случае Raspberry Pi, хост компьютер), к которому подключается любое количество исполнительных устройств на базе Arduino по универсальному последовательному порту (USB). Они могут различаться по модели, мощности и функционалу. Исполнительные устройства работают по протоколу, согласно которому в момент подключения они сообщают список и формат поддерживаемых ими команд. В этом заключается аппаратная расширяемость рассматриваемого комплекса. В качестве клиента, отправляющего запрос на Raspberry Pi, могут выступать: любой компьютер, подключенный к роботу посредством сети Ethernet (в том числе посредством Wi-Fi), смартфон или любое другое устройство, имеющее возможность работать по протоколу TCP. Драйвер транслирует получаемые из сети по протоколу TCP команды на соответствующие им исполнительные устройства. Имеется возможность посылать команды на одно или несколько исполнительных устройств одновременно. После того, как запрос будет полностью выполнен, клиенту приходит ответ в предустановленной форме.

Прокол обмена сообщениями

При подключении к Raspberry Pi посредством USB исполнительные устройства на базе Arduino определяются как стандартное USB-устройство, которое средствами GNU/Linux для просмотра заводской информации о подключенном устройстве по полям `VENDOR_ID` и `SHORT_VENDOR_ID` автоматически идентифицируется как допустимое устройство. Взаимодействие исполнительных устройств и хост компьютера осуществляется посредством библиотеки JSSC [9] по последовательному порту. Для совместимости устройства на базе Arduino с описываемым сетевым драйвером необходимы: реализация протокола обмена сообщениями на стороне исполнительного устройства, уникальное наименование устройства для его идентификации в пуле однотипных устройств, возвращаемое в ответ на команду «name». Кроме того, устройство должно располагать уникальным набором допустимых команд и возвращать этот список в ответ на команду «listCommand».

Для обработки запросов от внешних клиентов по сети Ethernet служит сервер, принимающий запросы и отправляющий ответы в простом текстовом формате с соблюдением несложных синтаксических правил. В целях организации асинхронной работы всех устройств в сетевом драйвере с каждым исполнительным устройством ассоциируется потоко-защищенная очередь задач. Множество внешних потоков программы могут добавлять неограниченное количество команд в очередь конкретного исполнительного устройства.

При соблюдении требований протокола на стороне исполнительных устройств и сетевых клиентов описываемый драйвер является прозрачным на стороне хост компьютера. При подключении и отключении исполнительных устройств он автоматически переконфигурируется, следовательно, при внесе-

нии изменений в функционал исполнительных устройств нет необходимости настраивать и перепрограммировать сетевой драйвер.

Команды сетевого драйвера

Сетевой драйвер имеет следующие виды команд.

- Команды для исполнительных устройств на базе Arduino, которые транслируются от сетевого клиента на соответствующие платы для исполнения на периферийном оборудовании (сервоприводы, датчики и т.д.).
- Внутренние команды сетевого драйвера, которые выполняются на стороне хост компьютера и не перенаправляются на исполнительные устройства.
- Команды для организации пула исполнительных устройств (идентификация устройств, получение списка встроенных команд и др.).

Любая последовательность команд может быть объединена в их именovanную последовательность – скрипт. Скрипты сохраняются в специальной директории на хост компьютере, указанной в конфигурационном файле. В последствии возможен вызов на исполнение скрипта по его имени. Все манипуляции (сохранение, загрузка, исполнение) со скриптами осуществляются с сетевых клиентов.

Стандартный набор команд сетевого драйвера не является конечным, имеется возможность программного расширения, путем добавление новых внутренних команд. Рассмотрим добавление команды `listScript`, которая возвращает список доступных скриптов. Для этого необходимо описать класс, реализующий интерфейс `Function<String, String>` и переопределить у него методы: `public String apply(String t)`, функцию исполнения команды, где `t` – текстовое представление команды, а возвращаемое значение – информация о списке доступных скриптов, и функцию `public String toString()` для корректного ведения логов о выполнении команд. Также необходимо в файл настроек драйвера внести информацию о созданном классе (Рис. 2).

```
<command>
  <name>listScripts</name>
  <textPresentation>^listScript$</textPresentation>
  <class>org.bgpu.raspberry_pi.structs.specified.ListScripts</class>
  <isPattern>true</isPattern>
</command>
```

Рис. 2. Пример описания команды сетевого драйвера

При поступлении запроса на выполнение внутренней команды драйвера, текстовое представление этой команды проверяется на соответствие регулярному выражению из параметра «`textPresentation`». Если строка запроса удовлетворяет регулярному выражению, то будет вызван метод `apply` класса, указанного в разделе «`class`».

Заключение

Описанный сетевой драйвер в роли ядра программной архитектуры антропоморфного робота позволяет осуществлять перманентную модернизацию и дополнение его узлов без внесения изменений в программную среду хост-компьютера, осуществляющего связь между клиентскими приложениями в се-

ти Ethernet и исполнительными устройствами робота. Количество исполнительных устройств ограничивается аппаратными возможностями хост компьютера (максимальным количеством подключаемых по USB устройств). Также при таком подходе возможно существование распределенной совокупности клиентских приложений, синхронизировано управляющих различными узлами робота в режиме, близком к реальному времени, что требуется во время проведения исследований в области робототехники.

Список используемых источников:

1. Сёмочкин А.Н. Антропоморфный робот как платформа для проведения исследований в области технологий человеко-машинного интерфейса на основе распознавания образов // Сборник научных статей по итогам Всероссийской научно-практической конференции с международным участием. Культ-информ-пресс, 2014. С. 207-208.
2. Сёмочкин А.Н. Темпьюнкт как модель узнавания образов // Высокие технологии, экономика, промышленность. СПб.: Изд-во Политехн. ун-та, 2012.
3. Сёмочкин А.Н. Адаптивный информационный терминал с функцией распознавания лиц // Высокие технологии, экономика, промышленность. СПб.: Изд-во Политехн. ун-та, 2013.
4. Сёмочкин А.Н. Синтезатор русской речи "Iskanderus". Свидетельство о государственной регистрации программы для ЭВМ №2012619614. Зарегистрировано в Реестре программ 24 октября 2012 года.
5. Сёмочкин А.Н. Модуль распознавания лиц для информационных терминалов. Свидетельство о государственной регистрации программы для ЭВМ №2013618233. Зарегистрировано в Реестре программ 04 сентября 2013 года.
6. Сёмочкин А.Н., Хаданович С.С. Сетевой драйвер устройств на базе Arduino. Свидетельство о государственной регистрации программы для ЭВМ №2015617881. Зарегистрировано в Реестре программ 23 июля 2015 года.
7. IskanderusController.
URL: <https://github.com/glamdringmetrandira/iskanderusController>
8. InMoov. Open source 3D printed life-size robot. URL: <http://inmoov.fr>
9. Java Simple Serial Connector.
URL: <https://code.google.com/p/java-simple-serial-connector>