

РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ СВЯЗИ СИСТЕМЫ УПРАВЛЕНИЯ С МИКРОКОНТРОЛЛЕРНЫМ КОМПЛЕКТОМ НА БАЗЕ ВСТРАИВАЕМЫХ СИСТЕМ

Бойко Д. С., Ивченков Н. В.

В статье рассматривается разработка интерфейса программно-аппаратного комплекса для связи системы управления с микроконтроллерным комплектом, который позволяет с небольшими затратами времени и использованием программного обеспечения открытого типа получить удобный в пользовании и простой в понимании интерфейс управления объектом. Проект интерфейса разработан на базе встраиваемых систем.

У статті розглядається розробка інтерфейсу програмно-апаратного комплексу для зв'язку системи керування з мікроконтролерним комплектом, який дозволяє з невеликими витратами часу і використанням програмного забезпечення відкритого типу отримати зручний в користуванні і простий в розумінні інтерфейс керування об'єктом. Проект інтерфейсу розроблений на базі вбудованих систем.

The article describes the interface development of hardware and software complex for connection of control software with microcontroller unit, which allows of obtaining the interface of object control easy to use and understand with a small amount of time and the use of the software of the open type. Project interface is designed on the basis of embedded systems.

Бойко Д. С.

Ивченков Н. В.

студент ДГМА
boiko_dima1993@mail.ru
канд. техн. наук, доц. каф. ЭСА
ДГМА

ДГМА – Донбасская государственная машиностроительная академия, г. Краматорск.

УДК 621.313

Бойко Д. С., Ивченко Н. В.

РАЗРАБОТКА ИНТЕРФЕЙСА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ СВЯЗИ СИСТЕМЫ УПРАВЛЕНИЯ С МИКРОКОНТРОЛЛЕРНЫМ КОМПЛЕКТОМ НА БАЗЕ ВСТРАИВАЕМЫХ СИСТЕМ

Стандартные средства реализации интерфейсов микроконтроллеров и персонального компьютера на примере TMS320f28335 представляют собой конфигурируемые системы, использующие специализированные визуальные редакторы. В современных системах управления, реализующих различные интерфейсы связи между оборудованием, в том числе и беспроводные, возникает необходимость применения альтернативных аппаратных и программных средств. Одним из таких решений является применение встраиваемых систем и open-source программного обеспечения, позволяющего модифицировать существующие решения под собственные нужды. Графический интерфейс пользователя (GUI) – это разновидность пользовательского интерфейса, в котором элементы интерфейса (меню, кнопки, значки, списки и т. п.), представленные пользователю на дисплее, исполнены в виде графических изображений. В отличие от интерфейса командной строки, в GUI пользователь имеет произвольный доступ (с помощью устройств ввода – клавиатуры, мыши, джойстика и т. п.) ко всем видимым экранным объектам (элементам интерфейса) и осуществляет непосредственное манипулирование ими. Графический интерфейс пользователя является частью пользовательского интерфейса и определяет взаимодействие с пользователем на уровне визуализированной информации.

Цель работы – разработка программно-аппаратного комплекса для реализации интерфейса связи между системой управления и микроконтроллерным комплектом с использованием встраиваемых систем и open-source программного обеспечения.

Интерфейс разрабатывается для лабораторного стенда, имитирующего работу лифта, в котором для управления электроприводом подъема применяется микроконтроллерный комплект TMS320F28xxx [1]. Общий вид установки представлен на рис. 1.

Электропривод механизма подъема кабины лифта построен по схеме «преобразователь частоты – асинхронный электродвигатель» (ПЧ–АД), являющийся наиболее распространенным для современных подъемных установок.

Для реализации системы управления электроприводом стенда используется микропроцессорный комплект на основе встраиваемого сигнального контроллера (DSC) серии TMS320F28xxx фирмы Texas Instruments. Вычислительная мощность комплекта и его развитая периферия позволяет реализовать как скалярные, так и векторные цифровые системы управления электроприводом стенда. Стенд является универсальным по отношению к системе управления. При неизменной силовой части, возможно подключение различных управляющих устройств, помимо рассматриваемого комплекта (TI TMS320F28xxx), таких, как одноплатные компьютеры Arduino, RASPBERRY PI2 и подобных. Общая структура электрооборудования стенда представлена на рис. 2.

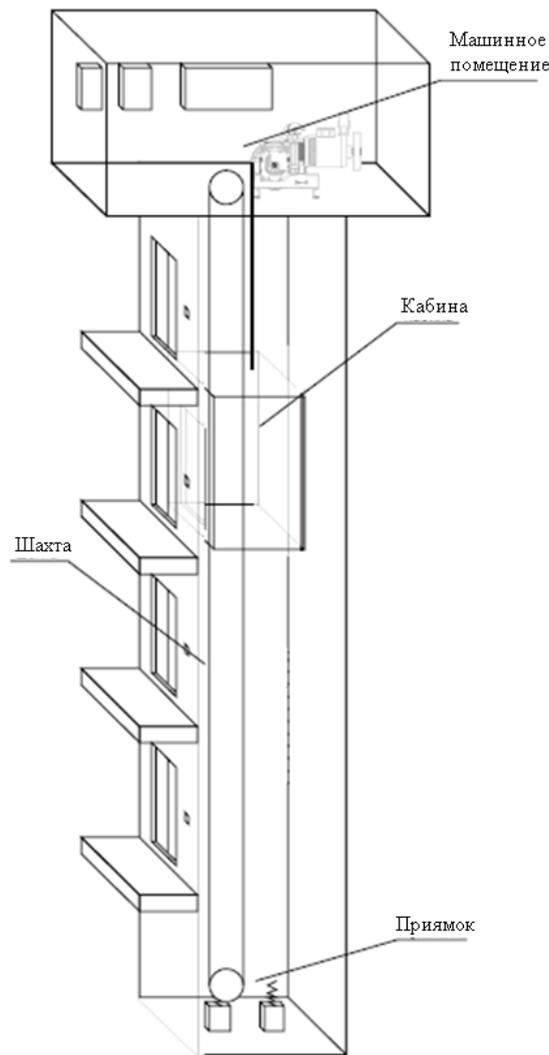


Рис. 1. Общий вид установки лифт

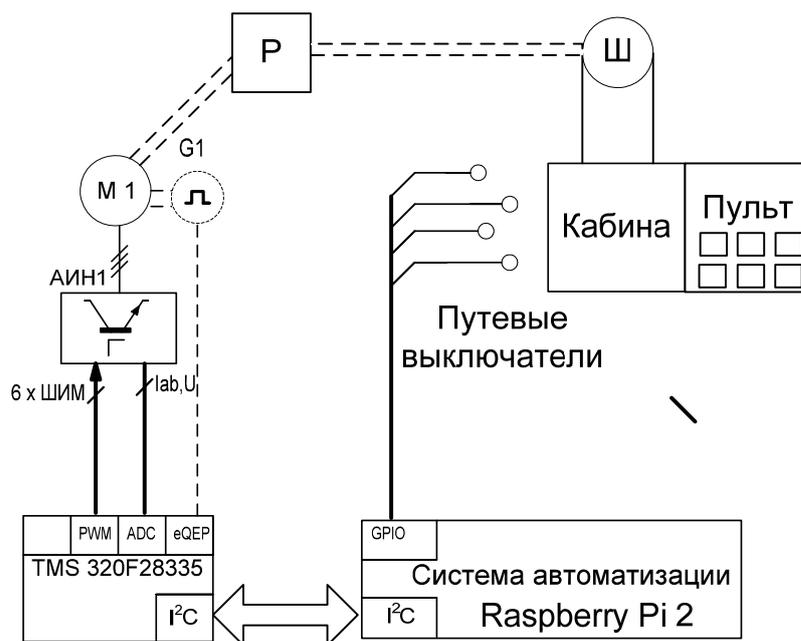


Рис. 2. Структурная схема электрооборудования лабораторного комплекса

При программировании микроконтроллеров программные модули могут быть написаны на языке Ассемблер конкретного процессора или на языке высокого уровня C/C++.

Основное преимущество программирования на языке высокого уровня состоит в универсальности языка – его аппаратной независимости от типа процессора. Главный недостаток – более высокие требования к ресурсам процессора, как по производительности, так и по объему памяти программ/данных. В связи с ростом производительности современных микроконтроллеров, увеличением объема встроенной памяти, развитием двухъядерных архитектур центрального процессора с аппаратной поддержкой вычислений с фиксированной и плавающей точкой, специальных средств оптимизации выходного кода этот недостаток постепенно нивелируется [2, 3].

Стандартная среда создания графического интерфейса пользователя для микроконтроллеров компании Texas Instruments - DSP/BIOS, является достаточно сложной статически конфигурируемой системой, для задания конфигурации которой используется специальная графическая утилита, интегрированная в среду разработки программного обеспечения Code Composer Studio. Утилита конфигурации – это специализированный визуальный редактор, который позволяет выбирать, какие модули DSP/BIOS будут включены в систему, а какие нет, а также задавать их параметры. Все параметры задаются статически до компиляции. При этом утилита конфигурации позволяет оценить объем требуемой под служебные нужды памяти, а также проверить соответствие заданных параметров, что позволяет избежать ошибок уже на начальном этапе конфигурации системы и сэкономить время на старте. Ещё одной функцией утилиты конфигурации является привязка проекта к конкретной аппаратной платформе. Именно в утилите конфигурации задаются параметры карты памяти, распределения прерываний и привязки тактовой частоты процессора к системным часам реального времени. При этом для различных ЦСП существуют уже готовые начальные схемы конфигурации DSP/BIOS.

Для реализации функций лабораторного стенда возникает необходимость использовать дополнительное микропроцессорное оборудование, реализующее основной логический функционал стенда, для которого использование DSP/BIOS может быть не применимо.

К примеру, одним из таких микропроцессорных устройств, предлагаемых к использованию, является микроконтроллерный модуль Arduino Mega2560 и плата расширения Ethernet Shield, реализующая функции связи модуля с другими устройствами посредством локальной компьютерной сети и витой пары. Также для реализации коммуникаций может использоваться Wi-Fi модуль ESP8266 (от компании Espressif Systems). Основной отличительной чертой такого модуля, является факт, что он представляет собой систему на чипе, включающую микроконтроллер с несколькими выводами GPIO (до 16) и радио-модуль Wi-Fi, что позволяет сократить количество используемых микроконтроллерных комплектов.

С точки зрения программного обеспечения, устройства на базе Arduino или ESP8266 программируются в доступной для понимания и широко распространенной бесплатной среде Arduino IDE, с предварительной установкой необходимых библиотек. Это позволяет, с небольшими затратами на обучение и установку оборудования заменить среду DSP/BIOS на Arduino IDE для создания интерфейса пользователя и основного программного кода лабораторного оборудования.

Программный интерфейс для управления микроконтроллерным комплектом через Arduino (ESP8266) [4] предлагается реализовать с использованием open-source программного обеспечения Blynk, который представляет собой облачный сервис для создания графических пультов управления и подходит для широкого спектра микрокомпьютеров и микроконтроллеров. Там, где раньше для сбора информации с датчиков нужно было писать полноценный интерфейс ввода-вывода или приобретать дополнительные модули, теперь можно обойтись использованием Blynk с максимально короткими сроками разработки конечного проекта [5].

Рассмотрим пошаговую настройку сервера Blynk.

Для реализации проекта необходимо зарегистрироваться в приложении и создать новый проект, как показано на рис. 3.

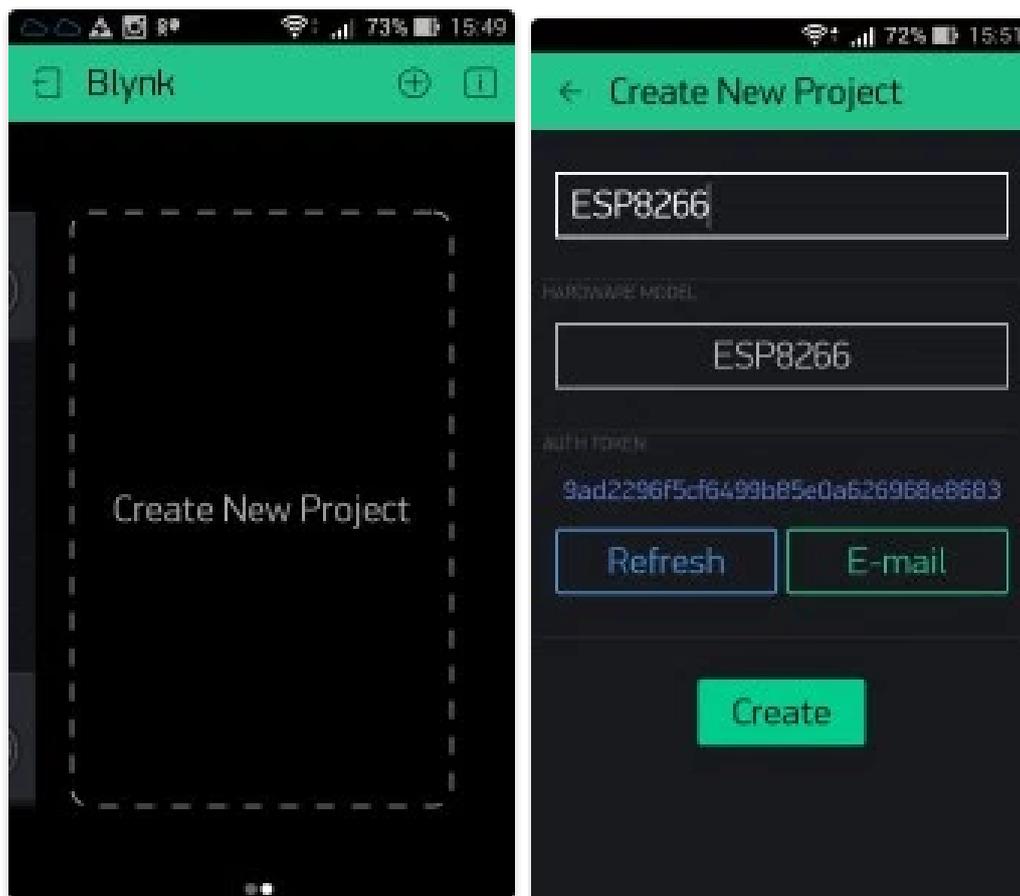


Рис. 3. Создание нового проекта в среде разработки Blynk

Ввести название проекта, например, ESP8266. В поле Hardware Model необходимо выбрать ESP8266. Автоматически сгенерированный Auth Token необходимо отправить на почтовый ящик или скопировать.

После нажатия кнопки Create, станет доступно поле для проектирования UI панели управления проектом (например, ESP8266). При нажатии соответствующей иконки появляется панель инструментов Widget Box для добавления виджетов на панель управления, позволяющая самостоятельно реализовать удобный графический интерфейс пользователя, который в дальнейшем можно динамически изменять при необходимости, как показано на рис. 4.

Установку виджетов для проекта покажем на примере виджета Button, как показано на рис. 5. При нажатии на него в WIDGET BOX он сразу добавляется на панель проекта. В поле OUTPUT PIN выбирается справа Digital, Analog или Virtual тип пина, а слева любой свободный пин GPIO или виртуальная переменная, необходимая для реализации внутренней логики работы системы управления. Ниже можно переключать тип кнопки на Push или Switch, в зависимости от требуемого функционала.

Далее следует установить библиотеку Blynk в Arduino IDE. Необходимо выбрать пример ESP8266_Standalone и прописать в нем Auth Token, который получен по почте или скопирован в начале процесса создания проекта, SSID и пароль Wi-Fi сети. Далее необходимо внести необходимый для стенда функционал в код скетча, скомпилировать и загрузить полученный скетч в плату модуля ESP8266. На устройстве в данном приложении в верхнем правом углу необходимо нажать кнопку запуска и перейти из режима редактирования проекта в режим управления ESP8266.

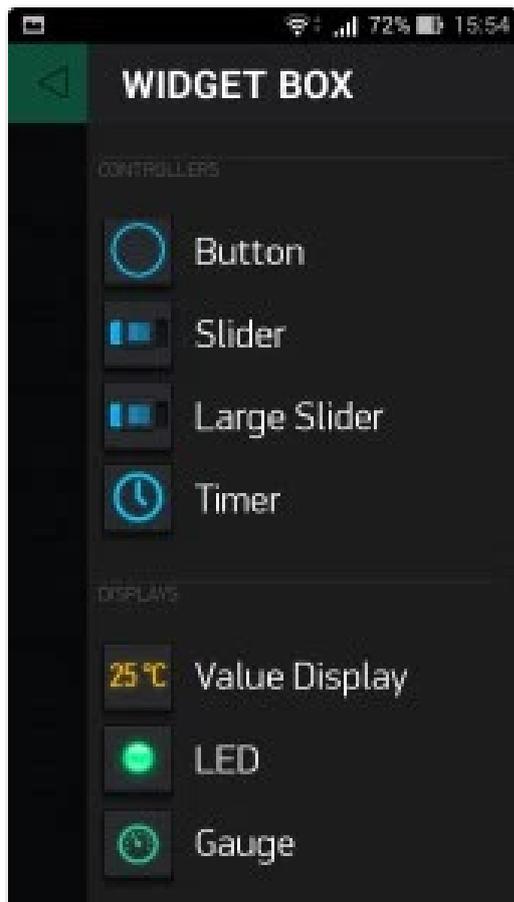


Рис. 4. Widget Box для проекта Blynk



Рис. 5. Применение виджета Button для проекта Blynk

Для того, чтобы получить максимальную безопасность и снизить задержку (лаг) при управлении стендом, можно установить сервер Blynk локально и ограничить доступ к сети, таким образом, чтобы только авторизованный персонал мог получить к нему доступ. В этом случае все данные хранятся локально в сети, без отправки через Интернет.

В случае локального сервера Blynk нет необходимости для защиты соединения между аппаратными средствами и локальным сервером Blynk. Это верно для подключения Ethernet и частично верно для Wi-Fi соединения. В случае Wi-Fi, необходимо использовать, по крайней мере, WPA, WPA2 (Wi-Fi Protected Access) протоколы Wi-Fi для защиты беспроводного трафика.

WPA и WPA2 предлагают очень надежное шифрование для защиты всех данных, передаваемых по беспроводной сети, при условии, что используется достаточно надежный пароль. Даже если данные представляют обычный протокол TCP/IP, другой пользователь не сможет расшифровать перехваченные пакеты. Тем не менее, необходимо убедиться, что пароль достаточно силен, иначе единственным ограничивающим фактором для злоумышленника может оказаться время на его определение [5].

Общая структура встраиваемой системы, реализованной с применением беспроводных технологий, на примере лабораторного стенда управления электроприводом лифта, представлена на рис. 6.

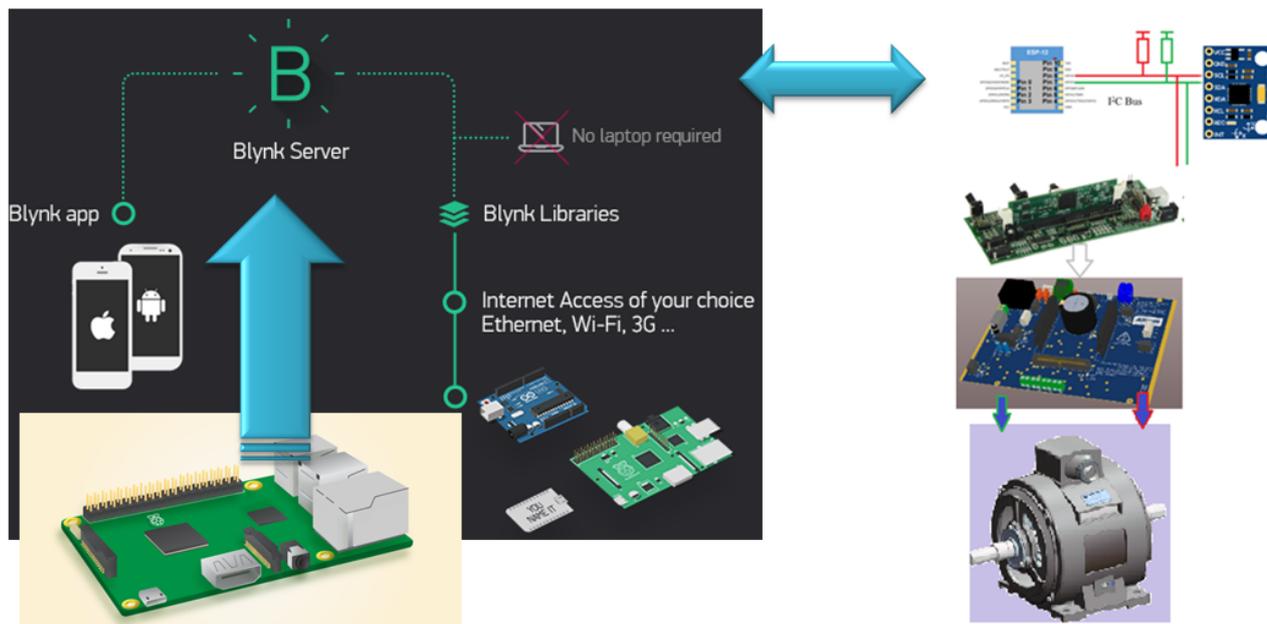


Рис. 6. Применение бlynk к стенду

На рис. 6 обозначены: модуль ESP8266 Wi-Fi (1), система датчиков на примере акселерометра-гироскопа MPU6050 (2) [6], микропроцессорные комплекты для управления преобразовательной техникой (3), силовой модуль преобразователя частоты (4) для управления асинхронным двигателем передвижения кабины стенда лифта (5).

ВЫВОДЫ

Предлагаемый программно-аппаратный комплекс позволит реализовать интерфейс связи встраиваемой системы с микроконтроллерным комплектом, управляющим электроприводом лифта. Он может быть использован в дисциплинах, связанных с изучением микропроцессорных устройств и цифровых систем управления электромеханическими комплексами, программированием и наладкой встраиваемых систем (Embedded Systems), в рамках организации удаленных лабораторных работ. Комплекс может быть использован в дисциплинах, изучающих технологии промышленных сетей и SCADA-систем.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *Sensorless Field Oriented Control of 3-Phase Induction Motors Using F2833x* [Электронный ресурс]. – Режим доступа: <http://www.ti.com/>.
2. *Практический курс разработки и отладки программного обеспечения сигнальных микроконтроллеров TMS320x28xxx в интегрированной среде Code Composer Studio: учеб. пособие* / А. С. Анучин, Д. И. Алямкин, А. В. Дроздов и др.; под общ. ред. В. Ф. Козаченко. – М. : Издательский дом МЭИ, 2010. – 270 с
3. Наливайко А.М. *Методика модульного программирования системы векторного управления асинхронным двигателем в среде разработки ccsv4 на базе учебного стенда фирмы texas instruments* / А.М. Наливайко, А.В. Чебаненко, Н.А. Катрушенко // ВІСНИК Донбаської державної машинобудівної академії. – 2015. – №3. – С.161-170.
4. *ESP8266EX Datasheet* / Espressif Inc, 2016. – 27 p.
5. *Руководство пользователя платформы Blynk*. [Электронный ресурс]. – Режим доступа: <http://docs.blynk.cc/>
6. *MPU-6000 and MPU-6050 Product Specification, Revision 3.4.* / InvenSense, Inc., 2013. – 53 p.

Статья поступила в редакцию 22.05.2016 г.