OpenBricks, LTIB, PTXdist являются узконаправленными системами сборки специфичных проектов и для нашей цели не подошли. OpenEmbedded и BuildRoot это очень мощные и гибко-настраиваемые системы. В результате исследования была выбрана система сборки BuildRoot, причиной послужила простота интерфейса и управления (аналогично интерфейсу конфигурации ядра Linux) [2].

В данной системе есть выбор целевой архитектуры, процессора, особенности работы ядра Linux, драйвера, программы и т.д. то, что необходимо включить в будущую встроенную систему. BuildRoot включает компоненты для загрузки пакетов и исходных кодов через интернет (autotools, make, git, ncurses и wgetscp). На рис.2. показана структура окружения разработки приложений под RaspberryPi.

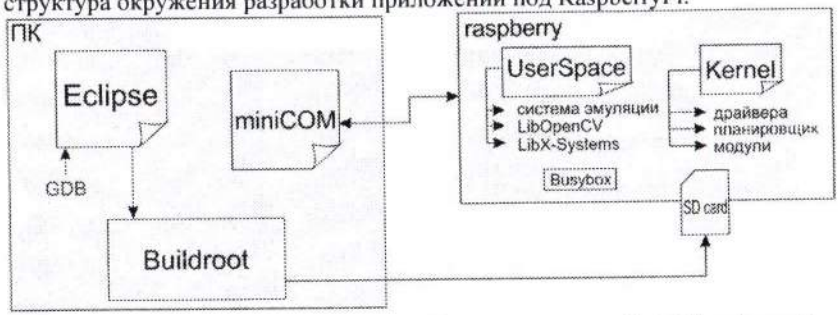

Рис. 2. Структура окружения разработки приложений под RaspberryPi

Для персонального компьютера (ПК) в окружение разработки входит: Eclipse – кроссплатформенная свободная интегрированная среда разработки; отладчик GDB -свободно распространяемый организацией GNU; miniCOM – терминал, для удаленной работы через последовательные порты; LibOpenCV и LibX-Systems библиотеки компьютерного зрения и передачи данных через Ethernet соответственно; BusyBox— набор утилит командной строки.

Сконфигурированное ПО для RaspberryPi состоит из следующих компонентов: ядро ОС Linux; система инициализации BusyBox; программы отладки ядра trace-cmd , kgdb; слежение за процессами htop; текстовые редакторы vim, nano; dhcpcd – dhcp клиент; библиотека компьютерного зрения opencv; библиотека работы с микрофоном sox; утилиты работы с сетью nettools.

Построение окружения разработки приложений

Для создания окружения все пакеты указываются в BuildRoot в меню TargetPackages. За набор компонентов на Raspberry отвечает сценарий в BuildRoot, поэтому в целевой системе компиляторы и библиотеки для сборки могут отсутствовать.

Для разработки собственного приложения, возможны два варианта действий:

1. Сборка целевой системы на Raspberry с программами компиляции, сборки и отладки, для разработки ПО внутри системы.
2. Разработка и сбора компонентов на ПК (кросс-компиляция), и добавление компонентов в пакеты целевой системы на Raspberry.

Заключение

Выше описанный метод позволяет формировать целевое программное обеспечение и создавать окружение разработки приложений для реализации прикладных задач на базе одноплатных компьютеров [3].

Результаты исследования вошли в проект «Интеллектуальная роботроника» Международной лаборатории «Сенсорика», ИПМ им. М.В. Келдыша РАН.

Библиографический список:

1. Солдатов А.В. Российские промышленные одноплатные компьютеры и их применение в системах автоматизации // ИСУП. - 2009. № 2 (22);
2. Обзор систем сборок для RaspberryPi [Электронный ресурс] // habrahabr.ru: IT-сообщество. URL: http://habrahabr.ru/post/172349/ (Дата обращения: 10.03.2015)
3. Sergej N. Grigoriev, Georgi M. Martinov Scalable open cross-platform kernel of PCN system for multi-axis machine tool // Procedia CIRP 1 (2012) p.p. 238 – 243.

СОЗДАНИЕ ПРОГРАММНО-АППАРАТНОЙ ПЛАТФОРМЫ ДЛЯ ТЕСТИРОВАНИЯ ЖИДКОКРИСТАЛЛИЧЕСКИХ ИНДИКАТОРОВ ВЫВОДА ИНФОРМАЦИИ СОСТОЯНИЯ РАБОТЫ ЯДРА СИСТЕМЫ ЧПУ

Бабин М.С.
Научный руководитель: преподаватель Ковалев И.А.
Кафедра «Компьютерные системы управления» ФГБОУ ВПО МГТУ «СТАНКИН»

Современные персональные компьютеры позволяют проводить систематизированный сбор сведений о подконтрольном объекте и средствах воздействия на его поведение с целью достижения определённых целей, таким образом, решая все задачи управления (геометрическую, логическую, терминальную) чисто программным путем, без какой-либо дополнительной аппаратной поддержки[1] [1,2].

При использовании систем реального времени для выполнения функций ядра системы управления обычно используется безграфические операционные системы. Вся информация с таких систем поступает на терминал оператора (терминал управления). Но из этого следует, что связь между ядром системы управления и терминалом будет установлена только

[1] Работа выполнена по договору № 2332 ГУ/2014от 19.06.2014 об условиях использования гранта на выполнение научно-исследовательских работ

после безошибочной инициализации обоих компонентов, следовательно необходим инструмент, с помощью которого можно обнаружить какие-либо ошибки еще на стадии загрузки ядра системы управления [3]. Таким инструментом может служить ЖК индикатор, который с одной стороны будет являться бюджетным решением, с другой –привлекательным с точки зрениягабаритных размеров.

Примером такого ЖКИ может служить модуль MT–16S2H, который состоит из БИС контроллера управленияКБ1013ВГ6, производства ОАО «АНГСТРЕМ» и ЖК панели (Рис.1).

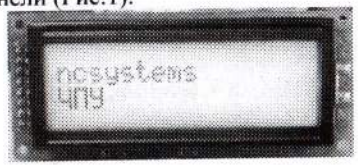

Рис. 1. ЖКИ MT–16S2H

При использовании программно-реализованного контроллера электроавтоматики для реализации программ управления зачастую необходима виртуальная отладка, т.е. без применения аппаратной части. Поэтому актуальной является задача обмена диагностическими данными между ядром SoftPlc и редактором управляющих программ в процессе работы.

Модуль позволяет отображать 2 строки из 16 символов. Символы отображаются в матрице 5x8 точек Каждому отображаемому на ЖКИ символу соответствует его код в ячейке ОЗУ модуля. Модуль содержит два вида памяти — кодов отображаемых символов и пользовательского знакогенератора, а также логику для управления ЖК панелью. Управляется по параллельному 4-х или 8-ми битному интерфейсу.

Учитывая, что данные ЖКИ являются зачастую чуть не единственным способом получения информации о состоянии ядра системы управления, перед их установкой они должны проходить различные тесты по выводу сообщений. Для этих целей было решено разработать экспериментальный стенд, который представлял бы из себя программно-аппаратную платформу тестирования ЖКИ.

За основу этой платформы была взята отладочная плата Texas Instruments MSP430G2553 (Рис.2).

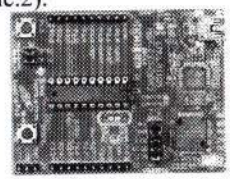

Рис. 2. Отладочная платаTexas Instruments MSP430G2553

Преимуществом данной отладочной платы является возможность отключение перемычек UART USB преобразователя и подключение контроллера на прямую к интерфейсу RS232.

Для реализации поставленной цели была разработана схема подключения отладочной платы и ЖКИ, представленная на (Рис.3).

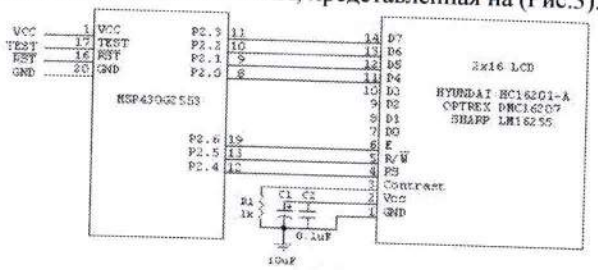

Рис. 3. Схема подключения ЖКИ к отладочной плате

Так же была разработана программа для отладочной платы на C++подобном языке и залита в контроллер. Данная программа представляет из себя следующий алгоритм: после подключения платы и питания индикатора с помощью библиотек работы с ЖКИ происходит ее инициализация, после чего возможна установка различных команд через нажатие кнопок на самой отладочной плате.

```
voidsetup()
{
 // Устанавливаем количество строк и символов в строке
lcd.begin(16, 2);
 // Инициализация COMпорта
Serial.begin(9600);
}
voidloop()
{
    if(!ButtonDown()){
    // Устанавливаем курсор на начала первой строки
    lcd.setCursor(0, 0);
    // Выводим на экран тестовое сообщения
    lcd.print("testncsystems");
    lcd.clear();
    // Устанавливаем курсор на начала второй строки
    lcd.setCursor(0, 1);
    // Выводим на экран тестовое сообщения
    lcd.print("системы ЧПУ");
    }
    else{
        //Если в порту есть символы для чтения
```

```
if (Serial.available()) {
  // Очищаем экран
lcd.clear();
  // Читаем доступные символы
while (Serial.available() > 0) {
    // ВыводимнаЖКИ
lcd.write(Serial.read());
  }
 }
}
```

Таким образом, при подключении ЖКИ к отладочной плате, на нем будут выводится тестовые сообщения, которые позволят определить его исправность.

Для варианта, когда необходимо тестировать различные типы сообщений без перепрошивки котроллера платы, была написана программа на языке C#, которая позволяет через виртуальныйCOM порт посылать различные сообщения, заданные пользователем на отладочную плату, а затем выводить их на ЖКИ.

```
// Пример отправки сообщения через COMпорт
if (serialPort1.IsOpen)
{
string message = textBox2.Text.ToString();
Encoding ansiCyrillic = Encoding.GetEncoding(1251);
byte[] testSendMessage= ansiCyrillic.GetBytes(message);
serialPort1.Write(testSendMessage, 0, testSendMessage.Length);
}
```

Таким образом, конечный пользователь сам выбирает способ тестирования: простой, при котором необходимо просто подключить ЖКИ к отладочной плате или расширенный, при котором возможно после перехода платы в режима приема, посылать любые сообщения (русские и английские буквы алфавита, различные символы).

Библиографический список:

1. Сосонкин В.Л, Мартинов Г.М. Системы числового программного управления: Учеб. Пособие. – М.: Логос, 2005.-296с.
2. Мартинов Г.М., Любимов А.Б., Бондаренко А.И., Сорокоумов А.Е., Ковалев И.А. Подход к построению мультипротокольной системы ЧПУ // Автоматизация в промышленности. 2012. №5. с.8-11.
3. И.А. Ковалев, М.С. Бабин, Р.В. Травкин Реализация управления знакосинтезирующим индикатором посредством динамической индикации с использованием программно-реализованного контроллера электроавтоматики SoftPLC. Материалы VII Международной научно-образовательной конференции "Машиностроение - традиции и инновации" (МТИ-2014). - М.: ФГБОУ ВПО МГТУ "СТАНКИН", 2014. - с.5-8.

АЛГОРИТМ АДАПТАЦИИ НЕЙРОСЕТЕВОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРОЦЕССОМ МЕТАЛЛООБРАБОТКИ

Бабушкин В.А.
Научный руководитель: Никишечкин А. П. – к. т. н., доцент
Кафедра «Компьютерные системы управления» ФГБОУ ВПО МГТУ «СТАНКИН»

В современных экономических условиях высокой конкуренции ведется постоянный поиск повышения качества продукции при снижении затрат на ее производство. Анализ работ в области управления процессами металлообработки показал, что задача адаптивной системы управления и оптимизации процесса резания по экономическим критериям, критериям качества и максимальной производительности в конечном итоге сводится к задаче стабилизации температурно-силовых и мощностных параметров процесса [1]. Именно изменение данных параметров в процессе обработки приводит к ухудшению качества обработки, поломки инструмента и вызывает необходимость занижать режимы резания [1,5].

Адаптивные системы реализуются по принципу двухмерного предельного управления по значениям скорости v и подачи s. Траектория движения инструмента, глубина резания, скорость и подача задаются штатной системой ЧПУ [2,3]. Коррекцию v и s в зависимости от текущих условий обработки осуществляет адаптивная система с учетом ограничений на управляющие воздействия. В качестве ограничений выступают мощности приводов подачи и главного движения.

Важнейшими задачами адаптивного управления являются разработка математических моделей процесса и синтез системы температурно-силовой и мощностной стабилизации.

Учитывая, что динамика любого процесса металлообработки очень нестабильна, зависит от массы факторов и определяется конкретной ситуацией, возможно использование нейросетевой модели, построенной по двухсетевой схеме с нейроконтроллером (НК) и нейроэмулятором (НЭ) [4,6,7]. Такая схема, использующая многослойные сети прямого распространения, обеспечивает точное обучение НК, так как при обучении НК и НЭ могут рассматриваться как единая многослойная сеть и ошибка может распространяться через НЭ в обратном направлении с целью получения минимальной ошибки на выходе НК [7].

Общая структура предлагаемой системы стабилизации с нейроэмулятором и нейроконтроллером представлена на рис. 1.