УДК 004

МАКЕТ МИКРОКОНТРОЛЛЕРНОГО СТЕНДА УПРАВЛЕНИЯ ПОЛОЖЕНИЕМ ВАЛА ДВИГАТЕЛЯ ПОСТОЯННОГО ТОКА

А.А. Черных
Научный руководитель: И.А. Тутов, ассистент каф. ИКСУ ИК ТПУ
Томский политехнический университет
E-mail: russk1j@mail.ru

В статье рассказывается про разработку макета стенда управления положением вала ДПТ.
The article tells about the development of the layout of the stand position control shaft DC motor.

Keywords: controller, proportional gain, UART, microcontroller, PWM.
Ключевые слова: регулятор, коэффициент пропорциональности, UART, микроконтроллер, ШИМ.

Введение

В системах автоматического управления зачастую требуется организовать контроль над объектом управления: контролировать различные параметры (скорость, температуру и пр.) с целью получения необходимых точности и качества переходного процесса. Один из самых распространенных задач управления – это поддержание угла поворота вала двигателя. Для управления обычно используют ПИД регулятор. Он применим во многих САУ.

Сборка лабораторного стенда

Для сборки лабораторного стенда была выбрана следующая элементная база: плата с микроконтроллером Atmega 16, переменный резистор 5 кОм, макетная плата, двигатель постоянного тока (ДПТ) с редуктором, драйвер ДПТ L293D, переходник USB – UART.

Сборка силового модуля ДПТ

Для управления вращением двигателя была реализована силовая часть, основанная на драйвере двигателя L293D. Управляющий сигнал подается с микроконтроллера на драйвер. К драйверу подключен внешний источник питания 4,5 В. Данный драйвер способен выдерживать ток до 600 мА на канал, что достаточно для данного маломощного двигателя. Также драйвер позволяет управлять скоростью двигателей с помощью широтно-импульсной модуляции (ШИМ).

Составление печатной платы произведено в программе SLayout-5.

Принцип работы стенда

Собранный стенд (схема) представлен на рис. 1.

Переменный резистор, представляющий собой датчик углового положения сервопривода был подключен ко входу АЦП (вывод PA0 МК). Вал редуктора осуществляет движения в диапазоне 0–210 градусов. При вращении вала редуктора значение сопротивления изменяется пропорционально углу поворота. Используется 8-битное АЦП с опорным напряжением 5 В. То есть диапазону измерения АЦП 0–255 соответствует диапазону изменения угла вала редуктора 0–210 градусов. Перевод из градусов в значения для МК производится через коэффициент 255/210 = 1,214. В ПК задается требуемое положения вала и оно передается в МК по UART интерфейсу.

Формула расчета значения ШИМ, подаваемого на двигатель для установки вала редуктора в требуемое положение:

$$b = 255 - (1.214 \cdot |u - g| \cdot K)$$

где b – значение ШИМ, K – коэффициент пропорциональности, u–текущее положение вала двигателя, g – требуемое положение вала двигателя, 255 – максимальное значение ШИМ

(5 В). Были проведены лабораторные испытания при K равном 5, 10, 20. Результаты приведены на рис. 2. При K = 5 наблюдается большая статическая ошибка.

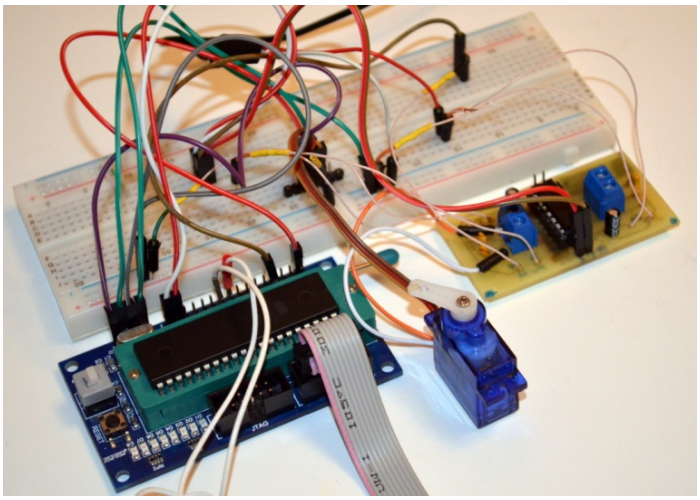

Рис. 1. Монтаж схемы для отладки ПИД регулятора

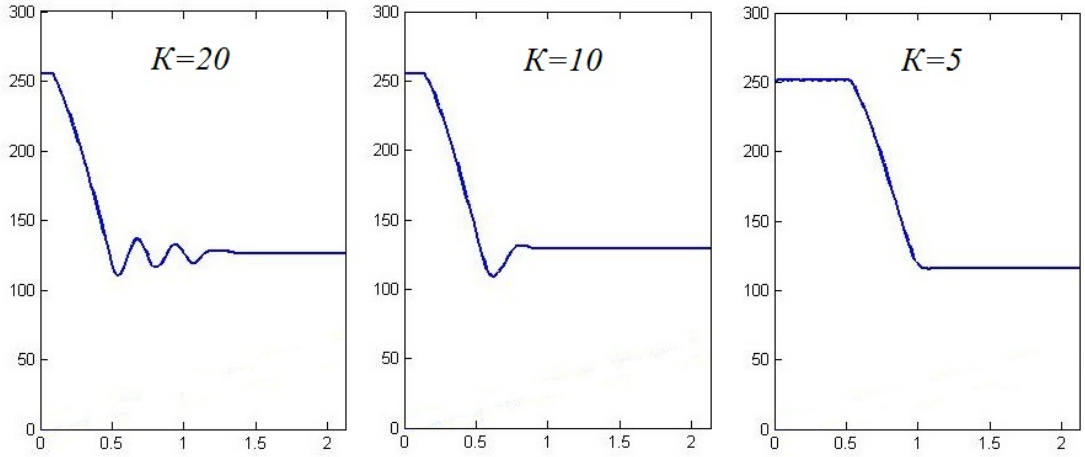

Рис. 2. Переходный процесс при K = 20, 10, 5

При этом в случае, если текущее положение меньше требуемого, вал двигателя вращается в одну сторону, в противном случае в другую. То есть определяется знак ошибки и в зависимости от этого задается направление движения вала. Для этого используются функции силового модуля на основе драйвера L293D. Программа реализована на языке C++ в Atmel Studio 6.0. Для вывода графиков с данных, переданных с ПК на МК по UART интерфейсу используется математический пакет Matlab 2010. Передача между ПК и МК осуществляется через переходник USB-UART. Через АЦП микроконтроллера будет считываться сопротивление с потенциометра, соответствующее определенному положению вала, и передаваться с МК на ПК.

Заключение

Для отладки и апробации алгоритма пропорционального регулятора был собран лабораторный стенд, и сопряжен с ПК для вывода информации в терминал по интерфейсу UART. На начальном этапе был испытан пропорциональной регулятор с разным коэффициентом пропорциональности на собранном макете стенда. В результате данные с МК отправляются на ПК по UART интерфейсу через USB-UART переходник с построением графика в пакете Matlab.

Список литературы

1. Блог: программирование микроконтроллеров, настройка UART [Электронный ресурс]. URL: http://radioparty.ru/prog-avr/program-c/307-lesson-usart-avr Режим доступа: свободный (дата обращения: 15.01.2015).
2. Datasheet на микроконтроллер ATmega16.

УДК 004

РАЗРАБОТКА ДРАЙВЕРА ДЛЯ ПРЯМОГО ДОСТУПА К ФИЗИЧЕСКОЙ ПАМЯТИ ОЗУ ДЛЯ ОПЕРАЦИОННЫХ СИСТЕМ WINDOWS XP-8.1

А.Г. Черемнов
Научный руководитель: И.А. Тутов, ассистент кафедры ИКСУ ИК ТПУ
Томский политехнический университет, 634050, Россия, г. Томск, пр. Ленина, 30
E-mail: 8xandr@gmail.ru

Implementation of kernel driver for direct access of physical space of RAM are considered in the paper. The algorithm of realization with a detailed description of each point is presented in this paper.

Key words: Direct access of physical RAM, system space of OS, system programming.
Ключевые слова: Прямой доступ к физической памяти, системное пространство ОС, системное программирование.

Необходимость прямого доступа к физической памяти возникает в таких программах как [1–5] для ускорения работы с памятью и скорости обмена между оперативной памятью и памятью графических ускорителей путём самостоятельного выделения оптимальных с точки зрения быстродействия структур данных и организации сжатия и распаковки данных, а также в задачах анализа вредоносного кода или исполняемых файлов и модулей, имеющих достаточно сложные упаковщики и протекторы кода, из-за которых получение корректного ассемлерного кода не представляется возможным до размещения этого кода непосредственно в оперативную память.

Разработанный драйвер для прямого доступа представляет собой функциональный драйвер, после инициализации которого создаётся виртуального устройство XandrIO для организации взаимодействия между пространствами ядра и пользователя операционной системы Windows.

В процессе отладки использовались следующие инструменты:

- Static Driver Verifier (SDV);
- PREFast for Drivers (PFD);

Для глубокой проверки и отладки исполняемого кода драйвера применялся SDV, позволяющий отслеживать через Windows Driver Model исполнение вызов функций [6]. Для поверхностного анализа операций сегмента кода драйвера использовался PFD, анализировались проблемы с утечкой памяти, например, переполнение буфера.

Отметим, что PFD гораздо быстрее, чем SDV, так как SDV применяется для каждой функции отдельно [6].

В качестве примера приведён фрагмент кода, осуществляющий получение доступа к физической памяти и осуществляющий трансляцию физического адреса в виртуальный для архитектур с 32-разрядной шинной адреса между оперативной памятью и центральным процессором.