

# РАЗРАБОТКА МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ СБОРА И АНАЛИЗА ДАННЫХ ПРИ РЕШЕНИИ ЗАДАЧИ РЕСУРСОЭНЕРГОСБЕРЕЖЕНИЯ ПРОИЗВОДСТВА

М.В.Воробьев

## Введение.

За последнее столетие человечество достигло необычайного технического прогресса, роста промышленности. Но параллельно этому развитию, положительному для мирового человеческого общества, обрели вес последствия загрязнения окружающей среды. Научные исследования, наблюдения за окружающей средой, позволили зафиксировать повышение содержания CO<sub>2</sub> газа в атмосфере, приводящему к «парниковому эффекту», увеличение средней годовой температуры в Европе до 1,2 С за 100 лет, разрушение озонового слоя и повышения уровня ультрафиолетового излучения проникающего на Землю, окисление водоемов и губительное воздействие на экосистемы соединений тяжелых металлов, нефтяное загрязнение, проблема с вторичным переработкой отходов (мусорные континенты). Мировое сообщество впервые за свою историю объединилось для решения экологических проблем, было подготовлено и подписано Киотское соглашение. Также были составлены рекомендации в которых для производства в частности немаловажным являлось снижение ресурсопотребления, энергопотребления. Снижая потребление, но оставаясь на том же уровне производства, можно сказать, что производство делает шаги в сторону увеличения эффективности ресурсопотребления. Промышленность Восточной Европы в 7 раз более энергоемкая чем в Западной Европе, поэтому, эффективность ресурсопотребления, для России в частности, является актуальной задачей [1].

Для оценки текущего ресурсопотребления предприятия на единицу продукции и принятие администрацией мер по увеличению эффективности ресурсопотребления, необходимо создать систему отображения ресурсопотребления предприятия.

**Постановка задачи.** Конечный результат: вывести для всеобщего обозрения графики на которых представлена информация о ресурсозатратах приведенных к одной единицы продукта в текущее время, графики производительности производственных линий, вести архив данных ресурсопотребления на предприятии по производству напитков.

**Рассмотрение аналогов.** Данная система сбора, хранения, вывода информации на терминал HMI (человеко-машинного интерфейса) является типичной системой SCADA (Диспетчерское управление и сбор данных) [2].

На данный момент существует множество способов реализации задачи диспетчерского контроля и сбора данных. Для выбора оптимального

решения, необходимо прежде всего сформулировать критерии оптимальности. Критерий оптимальности в данном случае был: минимальная стоимость системы, минимальное время ввода системы в работу.

В силу высокой распространенности персональных компьютеров (ПК), широких возможностей программирования, передачи данных по сетям Ethernet, локальным сетям, широкого выбора возможной периферии под конкретные задачи, и в силу отсутствия каких либо внешних воздействий на ПК, более, чем скажем воздействия в домашних условиях – головной системы выбран обычный ПК. Здесь нечего сказать, выбор очевиден. Проблема имеется с вводом данных ресурсопотребления в реальном времени в ПК.

Для наиболее оптимального выбора устройство ввода данных, необходимо прежде всего представлять какие данные необходимо вводить в ПК.

**Анализ существующей элементной базы на предприятии.** На предприятии установлены счетчики электроэнергии и счетчики газа с портом RS232. Имеются расходомеры с дискретным выходом. Для расходомера, имеется возможность выбирать в параметрах соответствие импульсам на выходе к пройденным через расходомер объемам жидкости. Производственные линии с промышленными контроллерами ПЛК линий не все имеют сетевой выход. ПЛК имеют свободные, резервные «цифровые» выходы.

Наиболее рациональным с учетом минимизации стоимости системы предлагается получать информацию о произведенной продукции, для ее подсчета, построения конечных графиков, последовательностью меандров, т.е. дискретными сигналами. Для этого требуется один модуль ввода/вывода расположенный в непосредственной близости к ПК. Интерфейс связи модуля ввода/вывода RS232.

Счетчик газа и электроэнергии по RS232 соединяются с ПК.

Таким образом, учитывая, что производственных линий 4, 2 счетчика расхода воды и одна из производственных линий является перенастраиваемая на всего 2 возможных формата выпускаемой продукции – необходимо вводить в ПК 7 цифровых сигналов. Причем характеристики сигнала, с учетом производительности линии и аппаратуры генерируемой сигналы (промышленная автоматика): логическая «единица» 24 В, минимальная длительность импульса 60 мс.

На данный момент существует огромный выбор устройств ввода/вывода для систем SCADA таких известных фирм SIEMENS, Allen Bradley, OVEN и т.д. Модули универсальные и позволяют настраивать различные протоколы связи. Универсальность устройств обуславливает их широкое применение, но увеличивает их стоимость, дополнительную сложность программы верхнего уровня ПК для организации полноценного

протокола, «привязывание» специалистов к конкретному производителю. Поэтому предлагалось разработать и использовать собственное устройство ввода/вывода.

В разработанной системе SCADA используется простой опрос портов RS232 с периодичностью 5 минут. Посылается периферийному устройству код запроса. После этого принимаются от периферийного устройства пакет данных.

#### **Вербальное описание устройства:**

Устройство ввода/вывода имеет 7 цифровых входов с логической единицей 24 В, быстродействием 12 мс каждого входа, подсчитывает число импульсов на входах в специальных регистрах, имеет интерфейс RS232 и протокол: по запросу ASCII символов «NS», передает строку из ASCII символов

вида:

«SN\_1.Z.XXXXXXXXXX\_2.0.XXXXXXXXXX\_3.0.XXXXXXXXXX\_4.0.XXXXXXXXXX\_W.0.XXXXXXXXXX», где X-цифра от 0 до 9, Z-цифра от 0 до 1 (формат линии), причем после 1.Z. передается число продукта подсчитанное на текущий момент на производственной линии 1; после 2.0. – на производственной линии 2; и т.д.; после W.0. – число импульсов полученное от двух расходомеров. После передачи строки – счетчики автоматически сбрасываются.

**Реализация.** Для реализации описанной логики выбран микроконтроллер фирмы Atmel ATmega32. Выбор в пользу данного микроконтроллера сделан, т.к. у автора имеется достаточный опыт работы с данным типом микроконтроллеров, сравнительной с аналогами дешевой, наличием четырех портов по 8 Бит входа/выхода, наличием полнодуплексного UART приемопередатчика, порта JTAG программирования и отладки. Микроконтроллер в DIP40 исполнении, т.к. плата делается экспериментальной, и возможны доработки. Для согласования логических уровней ПК и платы ввода/вывода используется микросхема MAX202SE или схема из транзисторов, если микросхема будет работать нестабильно. Входные дискретные сигналы 24 В гальванически развязаны со схемой микроконтроллера через оптопары ILD207T. Имеется 8 светодиодов – 6 из которых зеленого цвета – показывают состояние цифрового входа, после обработки контроллером, 2 из которых – красного цвета: один показывает сколь долго программа «крутится» в не основном цикле программы («загрузка»), другой загорается когда детектирована ошибка принятых данных UART.

Питание, в случае использования микросхемы MAX202SE, через плавкий предохранитель 0.25 А и сетевой трансформатор, со вторичной обмотки подается 6 В. Стандартная схема линейного источника питания. Для стабилизации напряжения 5 В используется широко распространенная микросхема L7805CD2T.

Однослойная плата спроектирована в среде P-CAD 2004 и представлена на рис. 1.

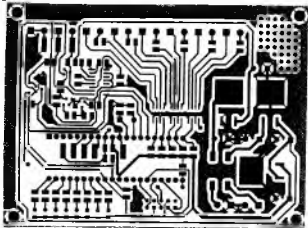


Рис. 1. ОПП для модуля дистанционного ввода/вывода цифровых сигналов

Монтаж платы, трансформатора, предохранителя в монтажной коробке DKC 54200.

**Программа микроконтроллера.** Программа написана в свободно распространяемой среде программирования микроконтроллеров AVRStudio компании Atmel.

Программу микроконтроллера можно разделить на блоки:

1. Инициализация микроконтроллера (загрузка указателя стека; отключение аналогового компаратора; включение таймера 2 на период 5.9 мс; включение прерывания по переполнению таймера 2; установить скорость передачи UART 9600 бит/сек; включить прерывание UART по принятию данных, окончанию передачи, установка параметров фрейма: 1 стоп бит, 8 разрядов данных, без проверки; конфигурация порта А на «выход»; конфигурация порта В на «вход» и «подтягивание» каждого входа к логической единицы внутренними резисторами; очистка оперативной памяти; разрешить прерывания микроконтроллера; сброс сигнала аварии на красные светодиоды)
2. Главный цикл (сброс сторожевого таймера):
  - 2.1. Подпрограмма обработки входных дискретных сигналов (вызов подпрограммы, если установлен флаг «обработка входных сигналов»)
    - 2.1.1. Детектирование фронта импульсного сигнала на дискретном входе (включить светодиод «загрузка»; сравнить текущее значение сигнала входа 1 (байт «текущее значение входа») и предыдущее значение сигнала входа 1 (байт «предыдущее значение сигнала входа»))

- значение входа), если есть отличие и текущее значение сигнала входа  $i$  равно логической единицы, то выполнить подпрограмму инкремента счетчика входа  $i$ ; тоже самое для каждого входа)
- 2.1.2. Инкремент соответствующего входу счетчика импульсов на входе (подпрограмма инкремента счетчика регистра состоящего из 3 байт)
  - 2.1.3. Сброс флага управления для 2.1 (сохранить область памяти «текущее значение входа» в «предыдущее значение входа», для возможности детектирования фронта сигналов входных в 2.1.1; выключить светодиод «загрузка»; сброс флага «обработка входных сигналов»)
  - 2.2. Подпрограмма обработки запроса по RS232 (вызов подпрограммы, если установлен флаг «обработка запроса по RS232»)
    - 2.2.1. Установка флага передачи (включить светодиод «загрузка»; если флаг «передача» сброшен, то копировать текущие значения счетчиков импульсов на входе в буфер, а сами счетчики сбросить в нуль, если флаг не сброшен, то ждать, когда завершится передача предыдущего пакета)
    - 2.2.2. Копирование значений счетчиков для обработки и сброс счетчиков импульсов каждого из входов
    - 2.2.3. Перевод шестнадцатирчных чисел счетчиков импульсов входов в ASCII код (из буфера памяти счетчиков перевести значение каждого счетчика в ASCII код и копировать в соответствующее место буфера памяти строки передачи, т.е. вместо XXXXXXXX каждого счетчика – конкретное значение)
    - 2.2.4. Формирование строки передачи в RAM (установка символов форматирования в буфер памяти строки передачи для соответствия виду  
 SN\_1.Z.XXXXXXXX 2.0.XXXXXXXX 3.0.XXXXXXXX 4.0.  
 XXXXXXXX\_W.0.XXXXXXXX; установка бита формата выпускаемой продукции линии  $i$  (Z-цифра); установить флаг «передача»)
    - 2.2.5. Конфигурация передачи и передача первого байта (установка счетчика переданных байт в  $i$ , послать на передачу первый байт строки передачи, включить прерывание по передаче байта)
    - 2.2.6. Сброс флага управления для 2.2 (выключить светодиод «загрузка»; сброс флага «обработка запроса по RS232»)
  - 2.3. Обновление флагов управления (копирование изменяемых флагов управления в обрабатываемые флаги управления – для строгой последовательности обработки)

3. Подпрограммы прерываний
  - 3.1. Подпрограмма циклического прерывания каждые 5.9 мс
    - 3.1.1. Сохранение значения цифрового сигнала на каждом дискретном входе (копирование рабочих регистров в стек; запись состояния входов 1,...,8 в байт «текущее значение входа»)
    - 3.1.2. Установка флага управления для 2.1 (установить флаг «обработка входных сигналов»)
    - 3.1.3. Формирование дискретных выходов – светодиоды состояния (копировать входы на выходы, для 6 светодиода – логическое ИЛИ между входами 7 и 8 – счетчики воды, 7 и 8, красные светодиоды, не обновляются – светодиоды аварий «загрузка» и «ошибка RS232»; копирование из стека рабочих регистров)
  - 3.2. Подпрограмма прерывания по приему байта
    - 3.2.1. Детектирование ошибок приема (копирование рабочих регистров в стек; если зафиксировано переполнение, ошибка кадрирования или несоответствие принятых байт запросу «NS», то включение светодиода «ошибка RS232»)
    - 3.2.2. Установка флага управления для 2.2 (если ошибок не обнаружено, то выключение светодиода «ошибка RS232»; включение передатчика UART; установка флага «обработка запроса по RS232»; копирование из стека рабочих регистров)
  - 3.3. Подпрограмма прерывания по передаче байта
    - 3.3.1. Передача следующего байта (копирование рабочих регистров в стек; если счетчик переданных байтов меньше 67, то передача байта из строки передачи в соответствии со значением счетчика)
    - 3.3.2. Инкремент счетчика переданных байтов (инкремент счетчика переданных байтов)
    - 3.3.3. На последнем байте отключение прерывания 3.3 (если счетчик переданных байтов больше или равен 67, то отключение прерывания по передаче байта; копирование из стека рабочих регистров)
  - 3.4. Подпрограмма прерывания по окончании передачи
    - 3.4.1. Сброс флага передачи (копирование рабочих регистров в стек; выключение передатчика UART; сброс флага «передача»; копирование из стека рабочих регистров)

Устройство введено в работу и успешно эксплуатируется.

### Список использованных источников

1. Каталог Окружающая среда Европы: третья оценка, Люксембург ТН-51-03-681-EN-C ([www.eea.eu.int](http://www.eea.eu.int))
2. <http://ru.wikipedia.org/wiki/SCADA>
3. Техническая документация на микросхему ATmega32 фирмы «ATMEL» - ([atmel.com/images/doc2503.pdf](http://atmel.com/images/doc2503.pdf))
4. Микроконтроллеры AVR семейств Tiny и Mega фирмы «ATMEL» Евстифеев А.В. - М.: Издательский дом «Додэка-XXI», 2004. – 560 с.

## МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ

В.А. Зеленский, В.П. Коннов, А.И. Щодро

Самарский государственный аэрокосмический университет, г. Самара

В настоящее время моделирование является необходимым этапом жизненного цикла любого сложного технического изделия или технологического процесса. Моделирование позволяет сократить сроки разработки документов, снизить общую стоимость разработки, оптимальным образом выбрать оборудование, найти специалистов нужной квалификации, повысить производительность труда. Специфика моделирования технологических процессов заключается в том, что они трудно поддаются формальному описанию [1]. Создать модель технологического процесса на основе системного или мультиагентного подхода - практически невыполнимая задача. Более целесообразным представляется использование парадигмы имитационного моделирования для решения актуальной задачи создания и анализа модели технологического процесса [2]. Иерархия парадигмы, математического аппарата и инструментария имитационного моделирования представлены на рис. 1.

В качестве математического аппарата имитационного моделирования предлагается использовать сети Петри [3]. Сети Петри характеризуются качественным описанием, количественными компонентами и функциями сети, начальной маркировкой сети, функцией переходов. Будем рассматривать ординарные автоматные сети Петри, не содержащие вложений и не оперирующие разнородными метками. Пример автоматной сети Петри представлен на рис. 2.