Создание web-сервера на базе Ethernet-контроллеров W7100A компании Wiznet

Сергей ДОЛГУШИН
dsa@efo.ru

Мы продолжаем знакомить наших читателей с возможностями Ethernet-контроллеров W7100A и W7100A-64QFN производства компании Wiznet. В этой статье рассказано о том, как реализовать на базе этих микросхем web-интерфейс для удаленного управления портами ввода/вывода.

Удаленное управление объектом и контроль его состояния являются актуальными задачами, одним из методов решения которых может служить локальный web-сервер. В данном случае под web-сервером будем понимать сервер на объекте, принимающий и обрабатывающий HTTP-запросы от клиента. Клиентом, как правило, служит web-браузер. Главным преимуществом такого решения является кроссплатформенность, то есть к управляющему объектом web-серверу можно подключаться из любой точки, где есть доступ к Интернету, или в частном случае из сети, к которой подключен данный web-сервер. Управление им можно осуществлять с любого ПК, ноутбука или мобильного устройства, на которых может быть запущен web-браузер.

Ethernet-контроллер W7100A (рис. 1), объединяющий на одном кристалле аппаратный блок Ethernet-моста со встроенными уровнями MAC и PHY, а также MSC-51 совместимое процессорное ядро, оптимален для создания такого приложения. От 32 до 19 линий ввода/вывода (в зависимости от типа корпуса, подробнее — в [1, 3]) этого контроллера могут служить для управления исполнительными устройствами и для получения информации от них. Компания Wiznet также предлагает готовый web-сервер Wiz220IO на базе этой микросхемы. При использовании фирменного ПО (firmware), которое изначально «прошито» в Ethernet-контроллер W7100A, могут быть реализованы следующие функции: управление устройствами по восьми дискретным цифровым выходам и восьми входам; работа с двумя источниками входного аналогового сигнала (например, датчиками давления) в диапазоне от 0 до 5 В; цифро-аналоговое преобразование по двум линиям в диапазоне от 0 до 4 В; прием и передача данных по интерфейсу UART. Управление модулем осуществляется через web-интерфейс, что избавляет разработчика от необходимости написания ПО верхнего уровня и обеспечивает совместимость с различными ОС. Элементы управления реализованы стандартными средствами языка HTML. Модуль Wiz220IO представляет собой готовое решение, на базе которого можно реализовать удаленную систему управления и контроля. Web-интерфейс модуля можно изменять и загружать в Ethernet-контроллер W7100A, не затрагивая при этом его основной программы.

В статье описано создание простого web-сервера на базе демонстрационного набора Wiz220IO-EVB. В этот набор, кроме модуля Wiz220IO, входит базовая плата Wiz220IO Base Board (рис. 2), на которой реализованы: схема питания модуля Wiz220IO с разъемом для подключения внешнего источника (позиция 5) и тумблера включения (позиция 6); светодиодная индикация (позиция 2), с помощью которой можно визуально оценить состояния восьми выходных линий модуля

Рис. 1. Модуль Wiz220IO

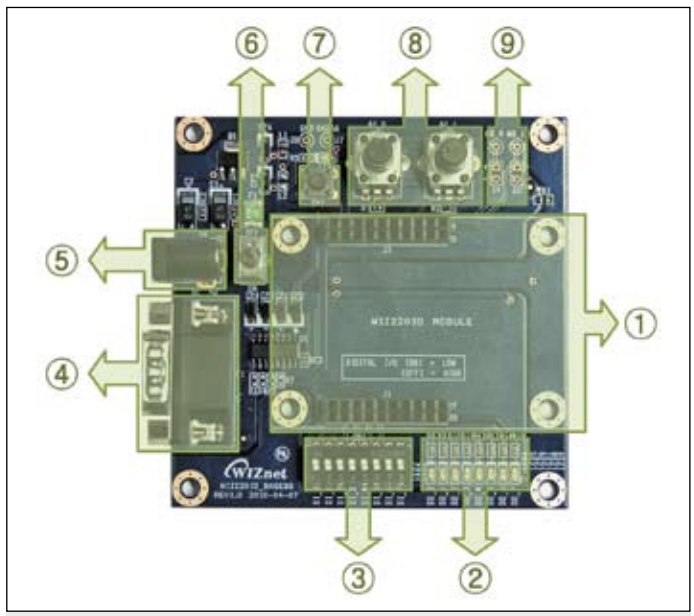

Рис. 2. Wiz220IO Base Board

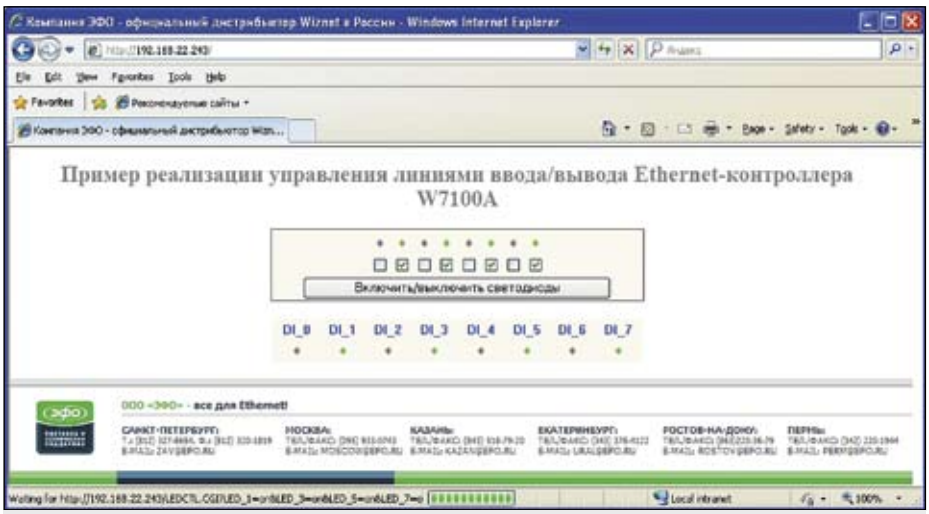

Рис. 3. Внешний вид web-страницы

Wiz220IO; набор переключателей (позиция 3), с помощью которых можно менять логическое состояние восьми входных линий модуля Wiz220IO; два потенциометра (позиция 8) для эмуляции изменяющегося входного аналогового сигнала; выходы ЦАП (позиция 9) и кнопка (позиция 7).

За основу приложения возьмем готовый пример компании Wiznet — "HTTPs_2011_01_12". В оригинальном виде пример доступен на официальном сайте производителя [4]. На его базе реализуем свой проект, исходные тексты которого размещены на сайте [7].

Приложение в общем виде состоит из двух частей: программы, обрабатывающей HTTP-запросы и реализующей те или иные действия в зависимости от запросов, и web-интерфейса, состоящего из одной или нескольких страниц с элементами управления. Для решения большинства задач элементы управления могут быть реализованы стандартными методами языка HTML. Эти стандартные способы будут использованы в описываемом примере.

Модуль Wiz220IO имеет 16 линий ввода/вывода — это первый и третий порты Etnernet-контроллера W7100A. Первый порт совместно с платой Base Board используется для ввода информации, что имитируется набором переключателей. Третий порт предназначен для вывода, его линии подключены к светодиодам платы Base Board. В соответствии с этим реализуем интерфейс, внешний вид которого приведен на рис. 3. Для управления и отображения текущего состояния светодиодов реализовано восемь флажков («чекбоксов») с индикацией текущего состояния над ними и управляющей кнопкой. Состояние линий первого порта будет индицироваться в нижней части страницы.

Главная часть листинга HTML-кода этой страницы приведена ниже, именно в ней формируются все управляющие и информационные объекты web-страницы:

```
<form name="form1" method="get" action="LEDCTL.CGI">
  <table width="400" border="0" cellpadding="1" cellspacing="1"
bgcolor="#666666" id="table1">
  <tr>

  <td bgcolor="#FEFEEF">
        <table width="400" border="0" cellspacing="0"
cellpadding="0" id="table2">
        <tr><td align="center">
        <table border="0" width="27%" id="table3">
        <tr>
          <td width="50"><font color="$DO_V0$">&diams;</
font></td>

          <td width="50"><font color="$DO_V7$">&diams;</
font></td>
        </tr>
        <tr>
          <td align="center"><input type="checkbox"
name="LED_0" $LED_0$></td>

          <td align="center"><input type="checkbox"
name="LED_7" $LED_7$></td>

        </tr>
      </table>
      </td></tr>
      <tr><td align="center"><p>
        <input type="submit" value="Включить/выключить све-
тодиоды" style="float: center">
      </td></tr>
      </table>

      </td></tr>
      </table>
      </form>

  <table width="400" align="center" bgcolor="#FEFEEF">
      <tr align="center">
        <td><font size="1" face="System" color="#0066FF">DI_0</
font></td>

        <td><font size="1" face="System" color="#0066FF">DI_7</
font></td>
  </tr>
  <tr align="center"><td><font color="$DI_V0$">&diams;</font></td>
        <td><font color="$DI_V1$">&diams;</font></td>

        <td><font color="$DI_V7$">&diams;</font></td>
  </tr>
  </table>
```

Этот листинг можно условно разбить на две части:

- интерактивную, где происходит взаимодействие клиента и web-сервера;
- информационную, где выводится информация о состоянии web-сервера.

Для рассматриваемого примера — это, соответственно, управление светодиодами и отображение текущего состояния первого порта контроллера W7100A. Необходимо понимать, что в информационной части обновление состояния контрольных элементов будет происходить в момент обновления страницы в браузере. Для автоматического обновления страницы можно добавить в заголовок следующий тег: ***<META HTTP-EQUIV="REFRESH" CONTENT="N; URL=адрес">***, где параметр N определяет интервал перезагрузки страницы в секундах, а URL для рассматриваемого примера не указывается (он служит для автоматической переадресации на указанную в адресе страницу).

При использовании автоматического обновления желательно размещать интерактивную и информационную части на разных страницах. В противном случае пользователю будет некомфортно работать с интерфейсом, так как он может не успеть ввести информацию и отправить ее на сервер. На примере web-интерфейса на рис. 3 это может выглядеть следующим образом. Пользователь хочет одновременно включить все восемь светодиодов: он начинает устанавливать флажки, но не успевает отправить эти изменения на сервер. В этот момент происходит автоматическое обновление страницы, все установленные флажки снимаются, и светодиоды, соответственно, остаются выключенными.

Для осуществления обмена информацией между браузером клиента и сервером в HTML существует понятие формы. Форма — это раздел HTML-страницы, в который кроме обычной информации и разметки входят специальные управляющие элементы. На одной странице может быть размещено несколько форм, но одновременно на сервер может быть послана информация только от одной из них. Информация посылается на сервер при нажатии управляющей кнопки **Submit** или с клавиатуры кнопкой «**Ввод**». В приведенном выше листинге форма начинается с обязательного тега ***<form name="form1" method="get" action="LEDCTL.CGI">*** и заканчивается обязательным тегом ***</form>***.

Атрибут method определяет механизм передачи данных из браузера серверу. Значение get говорит о том, что данные будут переданы в адресной строке в следующем виде: **http://192.168.22.243/LEDCTL. CGI?LED_0=on&LED_1=on&LED_2=on**. Эта строка означает, что были установлены первые три флажка (рис. 3).

Атрибут action указывает обработчик, к которому обращаются данные формы при их отправке на сервер. В рассматриваемом примере по этому атрибуту, значение которого также передается в адресной строке, основная программа Ethernet-контроллера W7100A определяет, какая функция будет обрабатывать этот запрос.

Далее, внутри формы, в табличном виде размещены элементы управления (нижняя строка таблицы) и индикации (верхняя строка таблицы) (рис. 3). Элемент индикации определен

следующим образом: *<td width="50">♦</td>*. Теги *<td>* и *</td>* определяют ячейку таблицы. В ячейке размещен элемент ***&diams*** — это код символа карточной масти «бубны». На его месте может быть любой отображаемый элемент. Индикация осуществляется изменением цвета этого элемента: **. Переменная ***DO_V0*** изменяет свое значение под управлением программы контроллера W7100A, а в соответствии с ним изменяется окраска элемента ***&diams***. В нижней строке таблицы размещены элементы управления — флажки: *<td align="center"><input type="checkbox" name="LED_0" LED_0></td>*. Элементы управления, обеспечивающие взаимодействие пользователя с сервером, задаются тегом *<input>*. Далее в теге определяется тип элемента, в данном случае флажок (checkbox), его имя — ***LED_0*** и атрибут ***LED_0***. Последний для рассматриваемого элемента может принимать значение ***checked*** (светодиод включен, флажок на web-странице помечен «галочкой») или быть пустым (светодиод выключен, «галочка» снята). Когда пользователь устанавливает этот флажок в активное состояние, на сервер передается адресная строка вида: **http://192.168.22.243/LEDCTL.CGI? LED_0=on**. Снимается флажок, и строка принимает вид: **http://192.168.22.243/LEDCTL.CGI? LED_0=off**.

За таблицей с описанными выше элементами располагается управляющий элемент: *<input type="submit" value="Включить/выключить светодиоды" style="float: center">*. Он представляет собой кнопку, при нажатии которой происходит отправка на сервер состояния управляющих элементов данной формы. То есть адресная строка **http://192.168.22.243/LEDCTL.CGI? LED_0=on** будет отправлена на сервер только после нажатия на кнопку **submit**. Таким же образом реализован интерфейс управления третьим портом W7100A.

Далее под формой, также в табличном виде, реализованы элементы индикации состояния первого порта. Они реализованы аналогично описанным выше: *<td>♦</td>*. Переменная ***DI_V1*** определяет цвет отображаемого символа, информация обновляется при обновлении web-страницы.

Web-интерфейс готов, теперь необходимо реализовать обработку в основной программе. За основу примера возьмем фирменный — HTTPs_2011_01_12: в нем уже реализованы все необходимые функции, обрабатывающие HTTP-запросы. Но в нем нет функций управления портами ввода/вывода. Далее приведем все необходимые изменения, которые потребуются для решения поставленной задачи.

Описываемое приложение должно работать с портами ввода/вывода. Для этого необходимо добавить соответствующие функции, в примере они описаны в файле *util.c*:

- **Din_state(uint8 port)** — возвращает состояние линии, указанной в переменной ***port***, первого порта.
- **Dout_state(uint8 port)** — возвращает состояние линии, указанной в переменной ***port***, третьего порта.
- **Dout_off(uint8 port)** — функция переводит указанную в переменной ***port*** линию третьего порта в высокое состояние, светодиод выключен.
- **Don_off(uint8 port)** — функция переводит указанную в переменной ***port*** линию третьего порта в низкое состояние, светодиод включен.

Сами функции просты, и их текст мы здесь не приводим.

Переходим к файлу *main.c* рассматриваемого проекта. Начнем с функции **proc_http(SOCKET s, u_char * buf)**, в которой обрабатываются HTTP-запросы.

После определения типа запроса ***if(http_request->TYPE == PTYPE_CGI)*** добавляем код, который управляет включением/выключением светодиодов:

```
   if(strstr(name,"LEDCTL.CGI"))  // запрос получен от формы с атрибутом "LEDCTL.CGI"
   {
      if((param = get_http_param_value(http_request->URI,"LED_0")))
// какой флажок на форме был активирован/деактивирован?
      {
         if(!strcmp(param,"on")) Dout_on(0); // если флажок активирован — включаем светодиод
         else          Dout_off(0);
      }
      else Dout_off(0);
      …
      …
      …
      if((param = get_http_param_value(http_request->URI,"LED_7")))
      {
         if(!strcmp(param,"on"))Dout_on(7);
         else Dout_off(7);
      }
      else Dout_off(7);

      strcpy(name,"dout.htm");
            find_http_uri_type(&http_request->TYPE, name);
```

После обработки запроса сервер должен изменить параметры (переменные, которые отвечают за изменение контрольных символов) web-страницы dout.htm. Подстановка обновленных значений осуществляется в функции **u_int replace_sys_env_value(u_char* base, u_int len)**. Для данного примера функция будет иметь следующий вид:

```
while((ptr=(u_char*)strchr((char*)tptr,'$')))
   {
      // изменяем цвет символа, отвечающего за индикацию
      // состояния нулевой линии третьего порта
      // DIO_ON_IMG = "#00FF00" — зеленый,
      // DIO_OFF_IMG=#999999" — серый
      if((tptr=(u_char*)strstr((char*)ptr,DO_0_IMG)))
      {
         memset(tptr,0,7);
         if(P3_0==ON)memcpy(tptr,DIO_ON_IMG,7);
         else memcpy(tptr,DIO_OFF_IMG,7);
         tptr+=7;
      }
      …
      …
      …
      else if((tptr=(u_char*)strstr((char*)ptr,DO_7_IMG)))
      {
         memset(tptr,0,7);
         if(P3_7==ON)memcpy(tptr,DIO_ON_IMG,7);
         else memcpy(tptr,DIO_OFF_IMG,7);
         tptr+=7;
      }
      //-------------------------------------------------------
      // изменяем атрибуты флажков, состояние "checked" —
      // соответствует тому, что данная линия третьего порта
      // находится в состоянии логического "0"
else if((tptr=(u_char*)strstr((char*)ptr,DO_0_STAT)))
      {
         memset(tptr,0x20,7);
         if(P3_0==ON) memcpy(tptr,"checked",7);
         tptr+=7;
      }
      …
      …
      …
      else if((tptr=(u_char*)strstr((char*)ptr,DO_7_STAT)))
      {
         memset(tptr,0x20,7);
         if(P3_7==ON) memcpy(tptr,"checked",7);
         tptr+=7;
      }
      //-------------------------------------------------------
      // данная часть отвечает за индикацию состояния первого порта,
      // зеленый цвет — на линии присутствует логическая "1",
      // серый — логический "0"
      else if((tptr=(u_char*)strstr((char*)ptr,DI_0_IMG)))
      {
         memset(tptr,0,7);
         if(P1_0)  memcpy(tptr,DIO_ON_IMG,7);
         else      memcpy(tptr,DIO_OFF_IMG,7);
         tptr+=7;
      }
      …
      …
      …
      else if((tptr=(u_char*)strstr((char*)ptr,DI_7_IMG)))
      {
         memset(tptr,0,7);
         if(P1_7)  memcpy(tptr,DIO_ON_IMG,7);
         else      memcpy(tptr,DIO_OFF_IMG,7);
         tptr+=7;
      }
      else
      {
         return len;

         if(ptr==base)
         {
               return len;
         }
         tptr = ptr;
         break;
      }
   }

   if(!ptr) return len;

   return (u_int)(tptr-base);
```

В результате этих изменений приложение сможет работать с портами ввода/вывода контроллера W7100A под управлением web-интерфейса, изображенного на рис. 3. Сетевые настройки, если необходимо, могут быть изменены. Их значения задаются в функции **default_network(void)**, определенной в файле *task_config.c*.

Теперь остается собрать проект целиком, то есть создать единый загрузочный образ, в который войдет исполняемая программа и web-страницы. Откомпилируем полученный код. По результатам компиляции должно быть получено информационное сообщение, изображенное на рис. 4. Нас будет интересовать размер полученного кода, это значение обведено на приведенном рисунке. Это будет стартовый адрес для размещения web-страниц, определенный следующим образом в файле *romfile.h*: ***#define FLASH_ROMFILE_START_ADDRESS 0x5685***.

После внесения изменений окончательно собираем проект и конвертируем полученный hex-файл в формат bin. Процедуры, необходимые для формирова-

Рис. 4. Информационное сообщение по результатам компиляции проекта

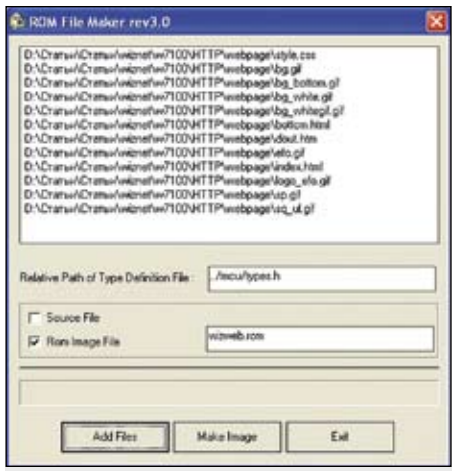

Рис. 5. Утилита ROMFILEMaker.exe

ния загрузочного файла, и процесс программирования Ethernet-контроллера W7100A и модуля Wiz220IO были подробно описаны в предыдущей статье [1]. Далее необходимо преобразовать все элементы web-интерфейса в образ, который будет записан в память контроллера W7100A. За его создание отвечает утилита *ROMFILEMaker.exe* (рис. 5). Настройки и название образа оставляем без изменений. Кнопка **Add files** открывает меню для добавления в формируемый образ необходимых файлов. Кнопка **Make image** запускает процесс создания образа с именем *wizweb.rom*, который будет создан в директории, где находится утилита. Все описанные утилиты включены в архив проекта, выложенного на сайте [7].

Теперь два полученных файла нужно объединить в один. Этот процесс выполняет утилита *allbin.exe*. Помещаем оба образа в один каталог с этой утилитой и запускаем *allbin.bat*. В результате будет сформирован файл с именем вида *All_20120112.bin*, где цифры обозначают текущую дату. Этот файл будет являться полным образом описанного web-сервера.

После программирования и включения модуля открываем браузер и в адресной строке вводим: **http://192.168.22.243/** (если IP-адрес в проекте был изменен, то, соответственно, вводится новый). В результате в браузере должна открыться страница, внешний вид которой приведен на рис. 3.

На данном примере описан процесс создания простого web-сервера, с помощью которого можно удаленно управлять отдельными линиями ввода/вывода Ethernet-контроллера W7100A. Такие функции могут быть востребованы в различных областях, например, при удаленном мониторинге температуры и влажности в помещениях или состояния охранно-пожарных датчиков. С помощью функций web-сервера можно реализовать удобный интерфейс для удаленного изменения настроек устройства. Реализация функции изменения сетевых настроек через web-интерфейс реализована в фирменном проекте HTTPs_2011_01_12. Все перечисленные и многие другие задачи поможет решить Ethernet-контроллер Wiznet W7100A.

Литература

1. Долгушин С. А. Ethernet-контроллеры W7100A компании Wiznet. Что нужно, чтобы начать работать с ними // Компоненты и технологии. 2012. № 1.
2. Долгушин С. А. Работаем с Vinculum II и W5100. USB от FTDI + Ethernet от Wiznet // Компоненты и технологии. 2011. № 12.
3. Internet Embedded MCU W7100A. Datasheet.
4. **www.wiznet.co.kr**
5. **www.efo.ru**
6. **http://htmlbook.ru/samhtml**
7. Исходные файлы проекта — **http://www.mymcu.ru/support/content/files/wiznet/http.rar**