

УДК 004.716

А.А. Мясищев,

доктор технических наук, профессор

С.В. Ленков,

доктор технических наук, профессор

Г.Б. Жиров,

кандидат технических наук,

старший научный сотрудник

ИЗУЧЕНИЕ ВОЗМОЖНОСТИ ПОСТРОЕНИЯ WEB-СЕРВЕРА НА ПЛАТФОРМЕ ARDUINO ДЛЯ УДАЛЕННОГО УПРАВЛЕНИЯ

Изучена возможность использования проекта Arduino для создания систем управления удаленными устройствами. Созданное электронное устройство работает как web-сервер, который не только отображает текстовую и графическую информацию, но и выполняет функции управления. Проведено сопоставление по функциональным возможностям двух типов web-серверов. Один построен на базе микроконтроллера ATmega128L, контроллера сети ENC28J60 и библиотек программной среды Arduino etherShield, ETHER_28J60. Другой – на базе Arduino UNO R3(ATmega328), контроллера Ethernet Shield W5100 с разъемом SD памяти и библиотеками Ethernet и SD. Показано существенное преимущество использования контроллера W5100 при создании сетевых устройств. Указаны также его недостатки.

Ключевые слова: Arduino, микроконтроллер, контроллер Ethernet, Web-сервер, шина SPI.

Вивчено можливість використання проекту Arduino для створення систем управління віддаленими пристроями. Створений електронний пристрій працює як web-сервер, який не тільки відображає текстову і графічну інформацію, а й виконує функції управління. Проведено зіставлення за функціональними можливостями двох типів web-серверів. Перший побудований на базі мікроконтроллера ATmega128L, контроллера мережі ENC28J60 і бібліотек програмного середовища Arduino etherShield, ETHER_28J60. Інший – на базі Arduino UNO R3 (ATmega328), контроллера Ethernet Shield W5100 з роз'ємом SD пам'яті і бібліотеками Ethernet та SD. Показана суттєва перевага використання контроллера W5100 при створенні мережевих пристроїв. Вказані також його недоліки.

Ключові слова: Arduino, мікроконтроллер, контроллер Ethernet, Web-сервер, шина SPI.

The possibility of using Arduino project for the creation of control remote devices is studied. Created electronic device works as a web-server, which not only displays text and graphics, but also serves as a control. A comparison of the functional capabilities of two types of web-server is done. One is based on a microcontroller ATmega128L, the network controller ENC28J60 libraries and software environment Arduino etherShield,

ETHER_28J60. Another is based on Arduino UNO R3 (ATmega328), the controller Ethernet Shield W5100 with built-in SD memory and libraries Ethernet and SD. A significant advantage of the use of the controller W5100 to create network devices is considered is considered. Its shortcomings are indicated.

Keywords: *Arduino, microcontroller, controller Ethernet, Web-server bus SPI.*

В настоящее время для управления различными устройствами (исполнительными механизмами, системами отопления, освещением и т.д.) широко используются микроконтроллеры, например ATmega. Они подключаются к соответствующим платам расширения, которые управляют исполнительными устройствами. Для программирования микроконтроллеров используют соответствующее программное обеспечение. Обычно программы пишутся на Си и Ассемблере. Однако в конце 2000 года по инициативе итальянских инженеров был создан проект Arduino, который в настоящее время интенсивно развивается для создания законченных систем управления исполнительными устройствами. Проект состоит из аппаратной платформы на базе микроконтроллера AVR и средства разработки программ [1, 2].

Плата Arduino состоит из ограниченного набора микроконтроллеров Atmel AVR (ATmega1280, ATmega328 в новых версиях и ATmega168, ATmega8 в старых) и элементной обвязки для программирования и интеграции с другими схемами. На каждой плате обязательно присутствуют линейный стабилизатор напряжения 5 В и 16 МГц кварцевый резонатор. В микроконтроллер предварительно прошит загрузчик, поэтому внешний программатор не используется. На концептуальном уровне все платы программируются через RS-232 интерфейс. Однако на последние варианты подключение для программирования выполняется через USB благодаря микросхеме конвертера USB-to-serial типа FTDI FT232. Платы Arduino позволяют использовать большую часть I/O выводов микроконтроллера во внешних схемах. Например, в плате Arduino Uno (ATmega328) доступно 14 цифровых вводов/выводов, в плате Arduino Mega (ATmega1280) доступно уже 54 цифровых вводов/выводов.

Интегрированная среда разработки Arduino – это кроссплатформенное приложение на Java, включающее в себя редактор кода, компилятор и модуль передачи прошивки в плату. Среда разработки основана на языке программирования Processing. Язык программирования аналогичен языку, используемому в проекте Wiring. Программы обрабатываются с помощью препроцессора (Wiring), а затем компилируются с помощью AVR-GCC.

Интенсивное развитие проекта Arduino связано с его преимуществами. Не нужен программатор. Программирование микроконтроллеров выполняется на языке высокого уровня (C++). Проект Arduino полностью открытый (можно изготавливать дополнительные платы расширения и дописывать собственные библиотеки программ). Платформа приобретает популярность – появляется множество сайтов в Интернете, наполненных библиотеками, схемами и проектами. Стандартизация расположения выводов платы ввода/вывода делает привлекательной ее для других производителей – появляются новые платы расширения (Shield), которые с помощью разъемов устанавливаются на плату Arduino. Проект использует кроссплатформенную среду разработки.

Однако недостатком проекта является ориентация программной среды на ограниченное количество микроконтроллеров AVR. Ряд популярных и распространенных микроконтроллеров, например ATmega16, ATmega32, ATmega128 не описаны в среде разработки. Поэтому возникает необходимость в адаптации среды программирования в том случае, когда электронное устройство

собирається из компонентов, не входящих в проект Arduino. Изучим возможность построение web-сервера двух типов. Первый – на базе микроконтроллера ATmega128L и Ethernet контроллера ENC28J60. Он должен управлять 3-мя устройствами и снимать показания с датчика температуры DS18B20. Второй сервер, построенный на стандартных контроллерах Arduino UNO и Ethernet Shield w5100, должен управлять мобильным телефоном, снимать показания с датчика температуры DS18B20, а также отображать статические html-документы, рисунки формата jpg, записанные на SD карте памяти. Фактически, второй сервер должен представлять собой универсальный web-сервер с функциями удаленного управления. Задачи, решаемые этими серверами, обычно рассматриваются в проекте “Умный дом”.

Рассмотрим построения первого сервера. В качестве микроконтроллера используется ATmega128L, на котором реализована работа с протоколами http, tcr, ip, а также функции управления внешними устройствами. Для работы с сетью на канальном и физическом уровне использован контроллер Ethernet ENC28J60. Достоинством ENC28J60 является его возможность работы с микроконтроллером по шине SPI. На рисунке 1 представлена схема включения контроллера ENC28J60.

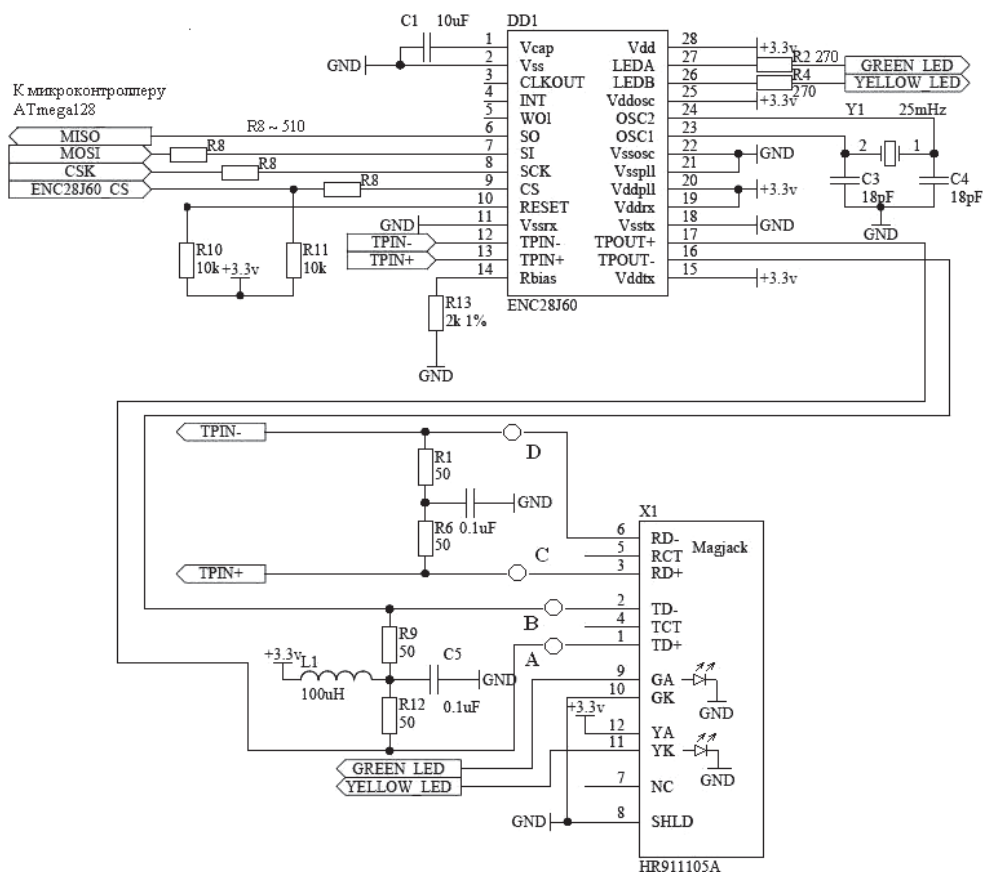


Рис. 1. Схема контроллера ENC28J60 с использованием разъема HR911105A

Для подключения к Ethernet используется разъем HR911105A со встроенным трансформатором и светодиодами. Причем полярность включения желтого светодиода определяет режим работы контроллера ENC28J60 – полудуплекс или полный дуплекс[3].

На схемі (рис.1) заслуговує увагу резистор R13. В описанні на ENC28J60 він вказаний з номіналом 2кОм. Однак у різних ревізій ENC28J60 існують оптимальні значення цього резистора. Так для ревізії 1 і 4 резистор повинен мати значення 2,7 кОм, а для ревізії 5 і 7 – 2,32 кОм. Визначається ревізія читанням відповідного регістра. Практично, установка номінала резистора R13 – 2,0кОм забезпечує працездатність схеми на рис.1, однак може знизитися дальність роботи, але це прийнятно для умов офісу. Для бюджетних пристроїв можлива безтрансформаторна схема включення, яка представлена на малюнку 2. Але в цьому випадку найближче комутаційне пристрій повинен знаходитися в неопосередкованій близькості. При такій схемі підключення стандарт IEEE 802.3 не виконується, але пристрій працює. Светодиоди повинні бути встановлені.

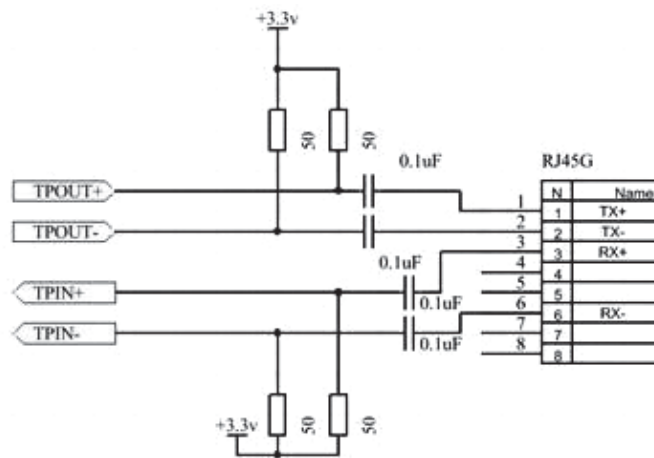


Рис. 2. Безтрансформаторна схема підключення

На рис. 3 представлена схема підключення ATmega128L до температурного датчика, двом світлодіодам і шині SPI, яка використовується для роботи з мережним модулем ENC28J60.

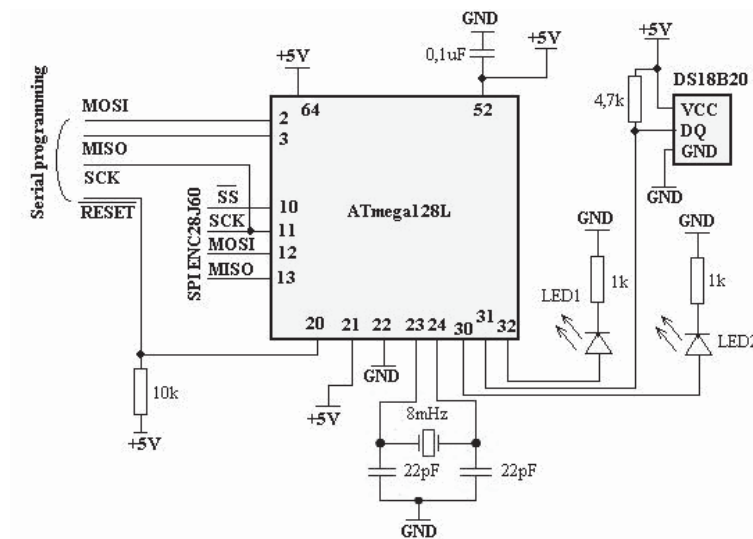


Рис. 3. Схема підключення ATmega128L

Несмотря на то, что при последовательном программировании ATmega128L используется тот же модуль SPI, что и при обычной работе микроконтроллера, имеется одно важное отличие: выводы MOSI/MISO модуля ввода-вывода SPI, которые совмещены с PB2 и PB3, не используются при программировании. Вместо них при последовательном программировании используются PE0 и PE1 для ввода и вывода данных [4]. Поэтому на рис. 3 отдельно представлена шина для программирования (Serial programming). Шина, обозначенная на рис. 3 как SPI ENC28J60, подключается к модулю с чипом ENC28J60, схема которого представлена на рисунке 1.

Программирование устройства выполнялось с помощью интегрированной среды разработки Arduino. Достоинство ее состоит в том, что в ней находится много полезных библиотек, в частности, для работы с шиной OneWire (для DS18B20) и чипом ENC28J60.

Как отмечалось выше, в программной среде Arduino описаны порты ввода-вывода для ограниченного числа микроконтроллеров. Поэтому проблема состоит в том, чтобы описать расположение выводов контроллера ATmega128L как в самой среде разработки, так и в библиотеке для чипа ENC28J60. Для этого корректируем файл

```
d:\arduino-1.0.3\hardware\arduino\variants\standard\pins_arduino.h.
```

Вместо

```
static const uint8_t SS = 10;      static const uint8_t SS = 8;
static const uint8_t MOSI = 11;   записываем static const uint8_t MOSI = 10;
static const uint8_t MISO = 12;   static const uint8_t MISO = 11;
static const uint8_t SCK = 13;    static const uint8_t SCK = 9;
```

В файле d:\arduino-1.0.3\libraries\etherShield\enc28j60.c библиотеки ENC28J60 также выполняем исправления.

Вместо

```
#define ENC28J60_CONTROL_CS 10   #define ENC28J60_CONTROL_CS 8
#define SPI_MOSI             11   записываем #define SPI_MOSI 10
#define SPI_MISO             12   #define SPI_MISO             11
#define SPI_SCK              13   #define SPI_SCK              9
```

Ниже приведена программа для управления тремя светодиодами (исполнительными механизмами) и считывания температуры с датчика DS18B20. Следует обратить внимание на то, что, по сравнению с рис. 3, программа расширена до 3 светодиодов. Они подключены к выводам микроконтроллера 28, 30, 32.

```
#include "etherShield.h"
#include "ETHER_28J60.h"
#include <OneWire.h>
#include <stdio.h>
static uint8_t mac[6] = {0x54, 0x55, 0x58, 0x10, 0x00, 0x24};
static uint8_t ip[4] = {192, 168, 1, 100}; static uint16_t port = 80;
ETHER_28J60 e; OneWire ds(6); char buf[30];
void setup()
{ e.setup(mac, ip, port);
  pinMode(3, OUTPUT); pinMode(5, OUTPUT); pinMode(7, OUTPUT); }
int freeRam () { extern int __heap_start, *__brkval; int v;
  return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval); }
```

```

float temp()
{ byte i; byte data[10]; byte addr[8]; float celsius;
  ds.search(addr); ds.reset(); ds.select(addr);
  ds.write(0x44,1); // start conversion, with parasite power on at the end
  delay(800); ds.reset(); ds.select(addr);
  ds.write(0xBE); // Read Scratchpad
  for ( i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read();} int raw = (data[1] << 8) | data[0];
  unsigned char t_mask[4] = {0x7, 0x3, 0x1, 0x0};
  byte cfg = (data[4] & 0x60) >> 5; raw &= ~t_mask[cfg]; celsius = (float)raw / 16;
  return celsius;}
void loop()
{
  char* buffer; if (buffer = e.serviceRequest())
  { e.print("<FONT size=5 color=green>Web – сервер на базі <font color=red>
ATmega128L</FONT> и контроллера <font color=blue> ENC28J60</font></
FONT><BR>");
  e.print("<hr>"); char r=buffer[3]; char g=buffer[7]; char b=buffer[11];
  if( r==0x31) digitalWrite(3,HIGH); if( r==0x30) digitalWrite(3,LOW);
  if( g==0x31) digitalWrite(5,HIGH); if( g==0x30) digitalWrite(5,LOW);
  if( b==0x31) digitalWrite(7,HIGH); if( b==0x30) digitalWrite(7,LOW);
  e.print("<FORM action='>");
  e.print("<FONT size=4 color=red>Красный светодиод:</FONT><BR>");
  e.print("<INPUT type='radio' name='r' value='1'>ON<br>");
  e.print("<INPUT type='radio' name='r' value='0'CHECKED>OFF<br><br>");
  e.print("<FONT size=4 color=green>Зеленый светодиод:</FONT><BR>");
  e.print("<INPUT type='radio' name='g' value='1'>ON<br>");
  e.print("<INPUT type='radio' name='g' value='0'CHECKED>OFF<br><br>");
  e.print("<FONT size=4 color=blue>Синий светодиод:</FONT><BR>");
  e.print("<INPUT type='radio' name='b' value='1'>ON<br>");
  e.print("<INPUT type='radio' name='b' value='0'CHECKED>OFF<br>");
  e.print("<INPUT type='submit' value='Ввести'></FORM>");
  float aa=temp(); dtostrf(aa,5,2,buf);
  e.print("<FONT size=4 color=red>Температура в градусах:</FONT><BR>");
  e.print(buf); e.print("<br>");
  if (digitalRead(3)){ e.print("<font size=4 color=red>Красный светодиод
ВКЛЮЧЕН</font><br>"); }
  else{ e.print("<font size=4 color=red>Красный светодиод ВЫКЛЮЧЕН</
font><br>"); }
  if (digitalRead(5)){ e.print("<font size=4 color=green>Зеленый светодиод
ВКЛЮЧЕН</font><br>");}
  else{ e.print("<font size=4 color=green>Зеленый светодиод ВЫКЛЮЧЕН</
font><br>"); }
  if (digitalRead(7)){e.print("<font size=4 color=blue>Синий светодиод
ВКЛЮЧЕН</font><br>"); }
  else{ e.print("<font size=4 color=blue>Синий светодиод ВЫКЛЮЧЕН</
font><br>"); }
}

```

```
e.print("<hr>"); e.print("Free SRAM:"); e.print(freeRam()); e.respond(); }
delay(1); }
```

Для снятия данных с температурного датчика дополнительно используется библиотека OneWire. По опыту работы библиотеки etherShield и ETHER_28J60 позволяют по запросу клиента отправить только один пакет. Поэтому если создать страничку, размер которой не укладывается в один пакет Ethernet (примерно 1500 байт), то это может привести к дополнительным сложностям в написании программы. Также в памяти микроконтроллера необходимо зарезервировать дополнительный буфер объемом 1500 байт. Для того, чтобы заработала программа для управления 3-мя светодиодами, в файл ETHER_28J60.cpp библиотеки ETHER_28J60 внесено изменение:

```
#define BUFFER_SIZE 1500
```

В противном случае web-сервер зависал.

Ethernet модуль на базе чипа W5100 и библиотеки Ethernet от Arduino позволяет обмениваться неограниченным количеством пакетов, поэтому на его основе возможно создание простого универсального web-сервера. В связи с этим рассмотрим построения второго сервера.

Выше было отмечено, что, в отличие от первого сервера, второй построен на стандартных контроллере Arduino UNO R3 и Ethernet Shield w5100 (рис.4).

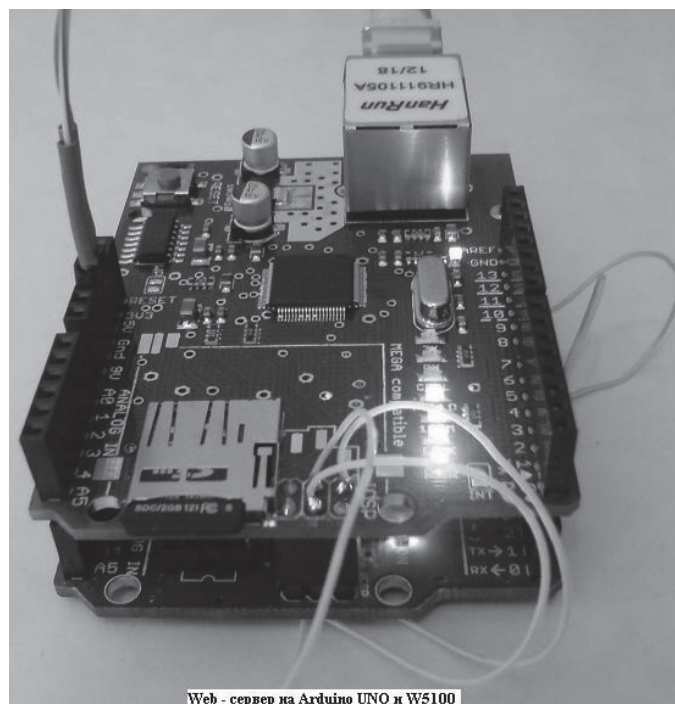


Рис.4. Web – сервер на Arduino UNO и W5100

Контроллер Ethernet Shield W5100 устанавливается с помощью разъемов на плате Arduino. Он основан на Ethernet – микросхеме Wiznet W5100, которая поддерживает также и стек TCP/IP. Для создания программ, которые подключают Arduino и Ethernet Shield W5100 к сети, используется библиотека Ethernet программной среды Arduino. Кроме стандартного сетевого разъема RJ-45,

контроллер Ethernet Shield W5100 имеет разъем для карт памяти типа microSD, которая используется для хранения файлов web-сервера. Разъем microSD доступен при помощи библиотеки SD.

Arduino осуществляет связь с W5100 и картой SD посредством шины SPI. При использовании библиотек Ethernet и SD вывод № 10 платы Arduino (формирует сигнал SS шины SPI) используется для выбора W5100, а ввод № 4 – карты SD. Эти выводы не могут быть использованы для другого ввода-вывода. Необходимо учитывать, что на плате Arduino Mega, аппаратный вывод SS № 53 не используется для выбора ни W5100, ни карты SD, но он должен быть сконфигурирован как вывод, иначе интерфейс SPI не будет работать.

Микросхема W5100 и карта SD разделяют шину SPI, поэтому одновременно они работать не могут. Если используются оба этих периферийных устройства в программе, следует использовать соответствующие им библиотеки. Однако если не используется ни одно из этих периферийных устройств, следует явно отключить их. Чтобы это сделать, необходимо сконфигурировать вывод платы № 4 для SD как выход и записать в него “1”. Для W5100 необходимо сделать то же самое, что и для вывода № 10.

В работе разработана и отлажена устойчиво работающая программа универсального web-сервера, которая отображает web-страницы с иллюстрациями и аппаратно с помощью реле сбрасывает сама себя примерно каждые две минуты, выполняет дозвон и считывает показания температурного датчика. Сброс необходим для выхода из “зависаний”, которые наблюдаются при интенсивном обращении со стороны одного или нескольких клиентов. Такие “зависания” замечены на платах Arduino UNO и Arduino Mega. Если web-страничка имеет размер в пределах одного Ethernet пакета, “зависаний” не наблюдается. Анализ показал, что проблемы создаются контроллером Ethernet Shield W5100. Для обеспечения устойчивой работы сервера была использована библиотека VEdduino[5], предназначенная для программирования счетчиков – таймеров. Согласно программе через определенный интервал времени выполнялось прерывание от таймера – счетчика 1. Управление передавалось функции обработки прерывания ISR(TIMER1_COMPA_vect). Каждое обращение к этой функции приводило к увеличению на единицу переменной s. Как только ее значение превышало 500, на 7-м выводе платы Arduino устанавливался высокий уровень, сигнал поступал на базу транзистора, который с помощью реле замыкал RESET-микросхему на “землю” (рис. 5). В этом случае происходил аппаратный сброс всего устройства, и если контроллер до этого “зависал”, то после RESET работа всего устройства возобновлялась.

Сервер также с помощью датчика DS18B20 определяет температуру в помещении и передает ее браузеру после нажатия на ссылку “информация”. При замыкании концевого выключателя, например, при открывании входной двери, на выводе 5 платы Arduino дважды устанавливается высокий уровень, поступающий на вход транзистора, замыкающего реле дозвона два раза (рис. 5). Контакты реле подключены к кнопке мобильного телефона “поднять трубку”, после чего телефон звонил по последнему номеру, по которому он дозванивался ранее. Программа сервера также дает возможность принудительно выполнить дозвон при условии подключения к нему по адресу: <http://192.168.1.100/tele>.

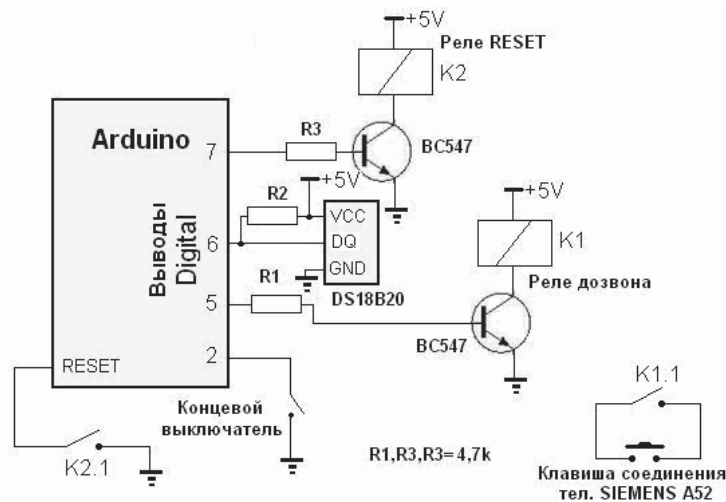


Рис. 5. Схема подключения к Arduino

Текст программы, реализующей работу электронного устройства в качестве web-сервера из-за ограничений, накладываемых на размер статьи, здесь не представлен. Эта программа с подробными комментариями приведена в работе [6], поэтому дополнительного описания не требует. В работе IP адреса рассмотренных серверов в программе представлены как локальные (192.168.1.100). Для подключения к ним через Интернет необходимо выполнить настройки, представленные в источнике [7].

Выводы.

1. Показана возможность построения на базе платформы Arduino универсальных web-серверов, позволяющих не только отображать достаточно большие объёмы информации, но и выполнять управление устройствами по TCP/IP сети, что характерно для микроконтроллеров.

2. Использование платформы Arduino значительно упрощает и ускоряет создание подобных достаточно сложных проектов из-за наличия увеличивающегося набора библиотек и достаточно дешёвых плат Arduino с минимальным набором электронных компонентов.

3. Обнаруженным существенным недостатком является нестабильная работа представленного здесь web-сервера, который периодически "виснет" при интенсивном к нему обращении. Анализ показал нестабильность работы контроллера Ethernet Shield W5100 совместно с его набором библиотек.

4. Недостатком также является довольно медленная совместная работа по шине SPI памяти SD и контроллера W5100. Несмотря на использование в программе блочного способа чтения данных с SD, скорость передачи данных по сети не превышала 17.5 Кбайт/с.

5. Несмотря на стабильность работы web-сервера, построенного на контроллере ENC28J60, его существенным недостатком является ограничение размера web-страницы размером поля данных пакета Ethernet при использовании наиболее распространенных библиотек программной платформы Arduino – etherShield и ETHER_28J60.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Arduino. [Electronic Resource] – Access Mode : [http : // www.arduino.cc/](http://www.arduino.cc/), 2013.
2. Проект CraftDuino. [Electronic Resource] – Access Mode : [http : // www.robocraft.ru/page/summary/](http://www.robocraft.ru/page/summary/), 2013.
3. Мясищев А.А. Web-сервер на ENC28J60 и микроконтроллере AVR для управления устройствами по TCP/IP сети // Materialy IX mezinarodi vedecko – prakticka conference “Predni vedecke novinky – 2013”. – Dil 9. Moderni informacni technologie. Matematika. Fyzika. Telovychova a sport : Praha. Publishing House “Education and Science” s.r.o – 64 stran.
4. ATmega128/L Datasheet [Electronic Resource]. – Access Mode : [http : // www.atmel.com/Images/doc2467.pdf](http://www.atmel.com/Images/doc2467.pdf), 2011.
5. Библиотека VE_AVR [Electronic Resource]. – Access Mode : [http : // sites.google.com/site/vanyambauselinux/biblioteka-ve_avr](http://sites.google.com/site/vanyambauselinux/biblioteka-ve_avr), 2013.
6. Мясищев А.А. О возможности построения универсального Web-сервера на Arduino для отображения информации и управления по TCP/IP сети / А. Мясищев [Electronic Resource]. – Access Mode : [http : // host56.no-ip.biz:8080](http://host56.no-ip.biz:8080), 2013.
7. Мясищев А.А. Использование wifi роутеров для управления устройствами через Интернет / А. Мясищев [Electronic Resource]. – Access Mode : [http : // host56.no-ip.biz:8080](http://host56.no-ip.biz:8080), 2013.

Отримано 11.10.2013.