

УДК 004.725.5

В.І. Сахаров

Інститут холоду, кріотехнології та екоенергетики ім. В.С. Мартиновського,
Одеська національна академія харчових технологій, вул. Дворянська, 1/3, м. Одеса, 65082

ВИКОРИСТАННЯ ЛОКАЛЬНОЇ МЕРЕЖІ ДЛЯ КОНТРОЛЮ ТЕМПЕРАТУРИ ТА ВОЛОГОСТІ ПОВІТРЯ

Розглянуті особливості побудови приладів для доступу до комп'ютерних локальних мереж, з використанням мікроконтролерів і однокристальних Ethernet інтерфейсів, мікросхем фірми Microchip ENC28J60 і ENC424J600.

Ключові слова – Мікроконтролер – Інтерфейс – Ethernet.

ИСПОЛЬЗОВАНИЕ ЛОКАЛЬНОЙ СЕТИ ДЛЯ КОНТРОЛЯ ТЕМПЕРАТУРЫ И ВЛАЖНОСТИ ВОЗДУХА

Рассмотрены особенности построения приборов доступа к локальным компьютерным сетям, с использованием микроконтроллеров и однокристальных Ethernet интерфейсов, микросхем фирмы Microchip ENC28J60 и ENC424J600.

Ключевые слова – Микроконтроллер – Интерфейс – Ethernet.

LOCAL NETWORK UTILIZATION FOR TEMPERATURE AND HUMIDITY CONTROL

The devices for access to computer local area networks construction features, with microcontrollers and single chip ethernet interfaces, company microchip enc28j60 and enc424j600 chips utilization are examined

Keywords: Microcontroller – Interface – Ethernet.

I. ВСТУП

Ethernet традиційно був досить таки складним інтерфейсом. Всі *Ethernet* чіпи мають велику кількість контактів і їх досить складно адаптувати до маленького мікроконтролеру з невеликою кількістю оперативної пам'яті. Тому, використовують мікроконтролери з вбудованим інтерфейсом *MII* або *RMII*, наприклад мікросхеми *AT90RM9600* та їм подібні, але при цьому необхідно використовувати фронтальні інтерфейси *Ethernet*, наприклад *DV9161* фірми *Davicom*. Крім того, на платі слід використовувати зовнішню оперативну пам'ять, для зберігання прийнятих пакетів і використовувати зовнішню програмну пам'ять, що приводить до значного подорожчання приладу. На ці вбудовані мікрокомп'ютери, як правило, встановлюють операційну систему *Linux*, яка забезпечує всі необхідні протоколи для роботи в локальній мережі. Компанія *Microchip* змінила світ з появою однокристальних *Ethernet* чипів - *ENC28J60* і *ENC424J600*.

II. ОСНОВНА ЧАСТИНА

Мікросхеми *ENC28J60* та *ENC424J600*, це невеликі чіпи всього з 28 та 44 контактами, відповідно. Мікросхема *ENC28J60* має інтерфейс *SPI*, який легко використовувати з будь-якого мікроконтролеру, а чіп *ENC424J600* крім інтерфейсу *SPI* може обмінюватись даними через паралельний інтерфейс шириною 8 або 16 біт, що

значно прискорює передачу та прийом пакетів даних.

Це відкриває цілий світ зовсім нових прикладних задач. Ви можете легко створювати невеликі пристрої, які можуть поширитися на все в будинку і просто будуть підключені в мережу *Ethernet*. Всі прилади легко підключити через *Ethernet*, цієї мікросхеми. Відстань більше не обмежуючий фактор. Навіть *Wi-Fi* з'єднання можливі, так як можливо підключити пристрої до бездротового мосту.

На рисунку 1 зображено алгоритм реалізації приладу для використання в локальній комп'ютерній мережі. Пунктирними лініями виділені блоки, які виконуються розробником виробу.

Для початку роботи можна повторити прилад, і скористатись програмним забезпеченням, описаний в [1]. В ній детально описано порядок створення *Ethernet* інтерфейсу і його використання. Для прискорення процесу розробки *Ethernet* інтерфейсу, також можна скористатися [2].

Arduino – це електронний конструктор і зручна платформа швидкої розробки електронних пристроїв для новачків та професіоналів. Платформа користується величезною популярністю в усьому світі завдяки зручності і простоті мови програмування, а також відкритій архітектурі і програмному коду. Пристрій програмується через *USB* без використання програматорів. *Arduino* дозволяє комп'ютеру вийти за рамки віртуального світу в фізичний та взаємодіяти з ним. Пристрої на базі *Arduino*

можуть отримувати інформацію про навколишнє середовище за допомогою різних датчиків, а також можуть управляти різними виконавчими пристроями.

Мікроконтролер на платі програмується за допомогою мови *Arduino* (заснований на мові *Wiring*) і середовища розробки *Arduino* (заснована на середовищі *Processing*). Проекти пристроїв, засновані на *Arduino*, можуть працювати самостійно, або ж взаємодіяти з програмним забезпеченням на комп'ютері (напр.: *Flash*, *Processing*, *MaxMSP*). Плати можуть бути зібрані користувачем самостійно або куплені в зборі. Програмне забезпечення доступне для безкоштовного скачування. Вихідні креслення схем (файли *CAD*) є загальнодоступними, користувачі можуть застосовувати їх на свій розсуд.



Рисунок 1 – Алгоритм реалізації приладу для використання в локальній комп'ютерній мережі

Існує кілька версій платформ *Arduino*. Остання версія *Leonardo* базується на мікроконтролері *ATmega32u4*. *Uno*, як і попередня версія *Duemilanove* побудовані на мікроконтролері *Atmel ATmega328*. Старі версії платформи *Diecimila* і перша робоча *Duemilanoves* були розроблені на основі *Atmel ATmega168*, більш ранні версії використовували *ATmega8*. *Arduino Mega2560*, в свою чергу, побудована на мікроконтролері *ATmega2560*.

Мова програмування пристроїв Ардуіно заснована на *C/C++*. Мова проста в засвоєнні, та на даний момент *Arduino* – це, мабуть, найзручніший спосіб програмування пристроїв на мікроконтролерах.

В якості платформи можна використовувати *Arduino UNO*, або *Arduino Duemilanove*. Ці платформи добре підходять для сполучки з *Arduino EthernetShield V1.0* інтерфейсом, виконаного на основі мікросхеми *ENC28J60*. *Arduino Mega* не стикується з цим інтерфейсом при допомозі роз'ємів, які є на платах, тому необхідно розробити перехідні кабелі. Плата *Arduino EthernetShield V1.0* являє собою окремих виріб на якому розташована мікросхема *ENC28J60*, лінійний трансформатор, роз'єм *RJ-45* і мікросхема *74HCT08D* – узгоджувач рівнів, так як плата має живлення 5В. Доступ до *EthernetSheeld* здійснюється за допомогою послідовного інтерфейсу *SPI*.

В якості середовища розробки програм можна використовувати безкоштовне середовище і мову *Arduino*, що можна завантажити з [2]. При цьому, використовується вбудований програматор, який працює через інтерфейс *USB*, який присутній на платформах *Arduino*. Також зручно використовувати середовище розробки програм *WinAVR*. Середовище програмування *WinAVR* являє собою потужний *C* та *C++* компілятор, який використовує бібліотеки *GCC* операційної системи *Linux* і працює на платформі *Microsoft Windows*. Даний програмний продукт ні в чому не поступається пакету *IAR Embeded Workbench for AVR*. В склад *WinAVR* включено зручний редактор програміста та інтегрована середовище розробки (*IDE*), компілятор мов *C/C++* для *AVR*, стандартні бібліотеки мов *C/C++ AVR* для використання з *GCC*, асемблер для мікроконтролерів *AVR*, програматор (програма для завантаження і вивантаження коду мікроконтролерів), автоматичний генератор *Make*-файлів *for AVR GCC* та інші. автоматичний генератор *Make*-файлів допомагає розробити *Make*-файл для компіляції проекту, що значно спрощує роботу з даним програмним середовищем.

Для побудови пристрою для контролю температури та вологості повітря можна використовувати апаратні засоби *Arduino Duemilanove* плюс *Arduino EthernetSheeld V1.0*, а також стек програм *TCP/IP*, розроблений фірмою *Microchip* і адаптований для мікроконтролерів типу *AVR*, який є в прикладах проектів *Arduino*. Можна скористатись пакетом програм *uip-1.0-refman.zip The uIP Embedded TCP/IP Stack* під авторством *Adam Dunkels* з *Swedish Institute of Computer Science*.

В якості датчику температури та вологості зручно використати датчик типу *DHT-22* фірми *Aosong(Guangzhou) Electronics Co.,Ltd*. *DHT-22* це цифровий датчик температури та вологості, що дозволяє калібрувати цифровий сигнал на виході. Складається з ємнісного датчика вологості і

термістора. Також, датчик містить в собі АЦП для перетворення аналогових значень вологості і температури.

Живлення становить 3.3-6.0В.

Визначення вологості 0-100% з 0,1% точністю.

Визначення температури -40...+125 град. з точністю 0,1 град.

Частота опитування не більше 0,5 Гц.

Розміри 15.5мм x 12мм x 5.5мм

Нумерація контактів датчика *DHT-22*:

1. *Vcc* (3-5В живлення).
2. *Data out* – виведення даних.
3. Не використовується.
4. Загальний.

Для обміну даними датчик *DHT-22* використовує *1-Wire* протокол (однопровідний протокол). Це низькошвидкісний двонаправлений напівдуплексний послідовний протокол обміну даними, що використовує всього один сигнальний дріт. Природно, потрібно ще і зворотній (земляний) провід, але про це маркетологи зазвичай замовчують. *1-Wire* протокол був розроблений фірмою *Dallas Semiconductor* в кінці 90-х років.

Є декілька типів сигналів, визначених *1-Wire* протоколом – «імпульс скидання», «імпульс присутності», «запис 0», «запис 1», «читання 0» та «читання 1». Всі ці сигнали, за винятком імпульсу присутності, формуються на шині головним пристроєм – *MASTER*. У нашому випадку це мікроконтролер *AVR*.

Принцип формування сигналів у всіх випадках однаковий. У початковому стані *1-Wire* шина за допомогою резистора підтягнута до плюса живлення. Головний пристрій «провалює» на певний час *1-Wire* шину в нуль, потім «відпускає» її та, якщо потрібно, «слухає» відповідь підлеглого (*SLAVE*) пристрою. У нашому випадку підлеглий пристрій – датчик *DHT-22*.

Операція запису біта наступна. Вивід мікроконтролеру встановлюється в режим виходу і на ньому встановлюється логічний нуль. Витримується пауза, тривалість якої залежить від значення переданого біта (0 або 1), потім вивід переводиться в режим входу в стані *Hi-z* і знову витримується пауза.

Операція читання біта проходить наступним чином. Вивід мікроконтролеру встановлюється в режим виходу і на ньому встановлюється логічний нуль. Витримується певна пауза, висновок переводиться в режим входу в стані *Hi-z*, витримується пауза, а потім мікроконтролер зчитує потенціал виведення.

Всі сеанси зв'язку мікроконтролеру з датчиком *DHT-22* починаються з сигналу скидання. Мікроконтролер на 480 мкс «провалює» *1-Wire* шину в нуль, а потім «відпускає» її. Якщо до шини підключений датчик *DHT-22*, то він виявляє позитивний перепад і після паузи в 15-60 мкс відповідає мікроконтролеру імпульсом

присутності – «провалює» шину в нуль на час від 60 до 240 мкс.

Обмін даними по *1-Wire* шині відбувається послідовно, молодшим бітом вперед. Передача або прийом одного біта даних виконуються протягом фіксованого проміжку часу, так званого тайм слота (*time slot*). Розрізняють тайм слоти запису і тайм слоти читання. Тривалість усіх тайм слотів повинна бути більше ніж 60 мкс, а пауза між тайм слотами більше 1 мкс. Для передачі нуля мікроконтролер «провалює» *1-Wire* шину на час від 60 до 120 мкс. Потім «відпускає» її і перед записом наступного біта витримує паузу більше 1 мкс.

Для передачі одиниці мікроконтролер «провалює» *1-Wire* шину на час від 1 до 15 мкс, «відпускає» її і витримує паузу. Пауза повинна бути такою, щоб тривалість тайм слота була більше ніж 60 +1 мкс.

DHT-22 є підлеглим пристроєм і може передавати дані, тільки коли мікроконтролер формує на *1-Wire* шині тайм слоти читання. Для формування тайм слоту читання мікроконтролер «провалює» *1-Wire* шину на час від 1 до 15 мкс, а потім «відпускає» її, передаючи управління станом *1-Wire* шини датчику *DHT-22*. Якщо *DHT-22* передає нуль, він утримує шину в «проваленому» стані (у стані логічного нуля) до кінця тайм слота. Якщо він передає 1, він залишає шину в «підтягнутому» стані.

Мікроконтролер може зчитувати дані датчика *DHT-22* через 15 мкс після початку тайм слота читання. Більше інформації про роботу датчика *DHT-22* можна знайти в документі [3] фірми *Aosong(Guangzhou) Electronics Co.,Ltd*.

Змонтований пристрій підключається до локальної мережі, в *EEPROM* мікроконтролеру прописуються *IP*-адреси пристрою, сервера та маски підмережі при допомозі інтерфейсу типу *RS232*.

Нижче приведена підпрограма, яка дозволяє отримати дані про температуру та вологість повітря з датчику *DHT-22*.

```
unsigned char readTH(char *data)
{
  //readTH
  unsigned char laststate = UHIGH;
  unsigned char counter = 0;
  unsigned char j = 0, i, k;
  //float f,fh;
  unsigned char f,fh;
  data[0]=0;
  data[1]=0;
  data[2]=0;
  data[3]=0;
  data[4]=0;
  OW_DIR_OUT_HIGH();
  // значення виводу=1
  _delay_ms(1); // затримка 1мс
  OW_DIR_OUT_LOW(); ;
  // значення виводу=0
  _delay_ms(1); //
```

```

cli();//заборона переривань
OW_DIR_OUT_HIGH();
ow_delay_us(40);
OW_DIR_IN();
//приймання даних з датчику DHT-22
for ( i=0; i< MAXTIMINGS; i++) {for
counter = 0;
while (OW_GET_IN() == laststate) {while
counter++;
ow_delay_us(1);//delayMicroseconds(1);
if(counter == 255) {//255
break;
} //255
} //while
laststate =OW_GET_IN();
if(counter == 255) break;
// ignore first 3 transitions
if ((i >= 4) && (laststate == 0))
{//if (i%2==0)
// shove each bit into the storage bytes
data[j/8] <<= 1;
if(counter > 40)
data[j/8] |= 1;
j++;
} //if
} //for
sei();
k=data[0];
j=data[1];
f=data[2];
fh=data[3];
//Температура-----
Tc_100=(f & 0x7f);
Tc_100=Tc_100 <<8;
Tc_100=Tc_100 | fh;
data[0]='T';
data[1]=' ';
if((f & 0x80)!=0) data[2]='-';
else data[2]='+';
if((Tc_100/100)==0) data[3]=' ';
else data[3]=(Tc_100/100) + '0';
whole=Tc_100;
data[4]= (whole-(whole/100)*100)/10 + '0';
data[5]='.';
data[6]= whole-(whole/10)*10
+ '0';
data[7]=' ';
//Вологість-----
whole=k;
whole=whole <<8;
whole=whole | j;
data[8]='H';
data[9]=' ';
if((whole/100)==0) data[10]=' ';
else data[10]=(whole/100) + '0';
data[11]= (whole-(whole/100)*100)/10 + '0'
;
data[12]='.';
data[13]= whole-(whole/10)*10 + '0';
data[14]='\n';

```

```

data[15]='\r';
//-----
i=16; //кількість байтів даних
return (i);
} //readTH

```

Підпрограма працює кожні дві секунди і використовується додатком для передачі даних через локальну мережу на екран комп'ютера при допомозі стандартних програм *Telnet*, *Putty* або інших.

III. ВИСНОВОК

Запропоновані матеріали розраховані на популяризацію застосування мікроконтролерів та однокристальних *Ethernet* інтерфейсів для побудови економічного доступу до локальних комп'ютерних мереж з метою збору і передачі інформації від датчиків різного типу, таких як температура, вологість, задимленість, освітленість і т.д. На базі даних пристроїв легко організуються прилади для дистанційного керування об'єктами при допомозі локальних та глобальних мереж.

ЛІТЕРАТУРА

1. Ethernet – устройство на микроконтроллере AVR – Microchip – ENC28J60
Електронний ресурс: <http://www.rlocman.ru> [5.03.2013]
2. Аппаратная платформа Arduino
Електронний ресурс: www.arduino.ru [5.03.2013]
3. Digital-output relative humidity&temperature sensor/module AM2303
Електронний ресурс: <http://www.adafruit.com> [5.03.2013]
4. Microchip Technology Inc.
Електронний ресурс: www.microchip.com [5.03.2013]
5. Atmel Corporation – Microcontrollers, 32 bit, and touch solution
Електронний ресурс: www.atmel.com [5.03.2013]
6. Introduction to the tuxgraphics TCP/IP stack, 3rd generation
Електронний ресурс: <http://tuxgraphics.org> [5.03.2013]

Получена в редакции 05.03.2013, принята к печати 04.06.2013