

20. Separating the Wheat from the Chaff: How to Tell the Good Sites from the Bad, <http://school.discovery.com/schrockguide/chaff.html>, перегляд 06.04.2008.
21. Teaching Web Evaluation, <http://lrs.ed.uiuc.edu/wp/credibility/page4.html>, Date Last Modified: 7/01/2002 by Valarie Pozen, перегляд 05.09.2007.
22. 21CIF Lesson Plans and Action Research Repository, <http://21cif.imsa.edu/resources/lessondb>, перегляд 12.10.2007.

Олег ПУСТОВИЙ, Анатолій ШМОРГУН

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЛАБОРАТОРНОЇ РОБОТИ, ПРАЦЮЮЧОЇ В РЕЖИМІ ON-LINE ЧЕРЕЗ USB-ІНТЕРФЕЙС

У статті наведено алгоритм програми для USB мікроконтролера, який пристосовано до лабораторної роботи, працюючої в режимі on-line. Показано переваги USB шини перед шиною RS-232 та майбутній розвиток цього напрямку.

Бурхливий розвиток інформаційних технологій зумовив широке використання їх в освіті. Стосовно лабораторних практикумів, викладачами створюються в основному віртуальні лабораторні роботи. Можна навести лише поодинокі приклади створення реальних лабораторних робіт, в яких використовується комп'ютер для реєстрації та обробки експериментальних даних. Навіть у всесвітньо відомій фірмі RHYWE з 321 лабораторної роботи менше 10% спряжено з комп'ютером.

Перехід на Болонську систему вимагає збільшення обсягу матеріалу, який студенти мають опрацювати самостійно. Якщо лабораторні практикуми курсу загальної фізики методично необхідно виконувати студентам безпосередньо в лабораторії, то спецпрактикуми на старших курсах іноді доцільно виконувати дистанційно.

У Чернігівському державному педагогічному університеті імені Т. Г. Шевченка впродовж десяти років існує лабораторний практикум з фізики рідких кристалів. Раніше для з'єднання лабораторної роботи з комп'ютером використовували шину RS232, а зараз для однієї з робіт [1. 397–398] авторами розроблено USB-пристрій на основі AT89C5131.

Шина USB з'явилась на початку 1996 року як спроба вирішення проблеми різноманітності інтерфейсів. На той час персональні комп'ютери були обладнані великою кількістю зовнішніх інтерфейсів, які були корисні та необхідні, але мали один недолік: всі вони потребували свого специфічного з'єднання і виділеного апаратного переривання.

Перша специфікація USB [2, 9–11] (версія 1.0) була опублікована на початку 1996 року, а восени 1998 року з'явилась специфікація 1.1, що виправляла проблеми першої редакції. Навесні 2000 року була опублікована версія 2.0, де пропускна здатність шини збільшувалась в 40 разів. Так, специфікації 1.0 і 1.1 забезпечують роботу на швидкості 12 Мбіт/с і 1,5 Мбіт/с, а специфікація 2.0 — на швидкості 480 Мбіт/с. При цьому передбачена зворотна сумісність USB 2.0 с USB 1.x, тобто «старі» USB 1.x пристрої будуть працювати з USB 2.0 контролерами, але на швидкості 12 Мбіт/с.

Даний пристрій має багато переваг, а саме:

- можливість легко підключати периферійні пристрої;
- дешеве рішення, що дозволяє передавати дані зі швидкістю до 12 Мбіт/с для USB 1.x і 480 Мбіт/с для USB 2.0;
- повна підтримка в реальному часі голосових, аудіо- та відео потоків;
- гнучкість протоколу змішаної передачі ізохронних даних і асинхронних повідомлень;
- інтеграція з новими пристроями;
- охоплення всіх можливих конфігурацій конструкцій ПК;
- забезпечення стандартного інтерфейсу;
- створення нових класів приладів, що розширюють ПК;

Специфікація USB визначає наступні функціональні можливості інтерфейсу:

- простота використання для кінцевого користувача:
 - простота кабельної системи і підключень;
 - пристрої ідентифікуються і конфігуруються автоматично;
 - приховуються подробиці електричного підключення від кінцевого користувача;
 - динамічне підключення і переконфігурування периферійних пристроїв;
- широкі можливості роботи:
 - пропускна здатність від кількох Кбіт/с до кількох Мбіт/с;
 - підтримка одночасно як ізохронної, так і асинхронної передачі даних;
 - підтримка одночасних операцій з багатьма пристроями (multiple connections);
 - підтримка до 127 пристроїв на шині;
 - передача різноманітних потоків даних і повідомлень;
 - підтримка периферійних пристроїв, що виконують кілька функцій;
 - низькі накладні витрати передачі даних;
- рівномірна пропускна здатність:
 - гарантована пропускна здатність і низькі затримки голосових і аудіо даних;
 - можливість використання всієї пропускної смуги;
- гнучкість:
 - підтримка різних розмірів пакетів, що дозволяють налаштувати функції буферизації пристроїв;
 - можливість настройки співвідношення розміру пакета і затримки даних;
 - керування потоком (flow control) даних на рівні протоколу;
- надійність:
 - контроль помилок і відновлення на рівні протоколу;
 - динамічне додавання і видалення пристроїв прозоре для кінцевого користувача;
 - підтримка ідентифікації непрацюючих пристроїв;
 - виключення можливості неправильного з'єднання пристроїв;
- вигода для розробників:
 - простота реалізації і впровадження;
 - об'єднання з архітектурою Plug and Play;
- дешева реалізація:
 - дешеві канали зі швидкістю роботи до 1,5 Мбіт/с;
 - оптимізація для інтеграції з периферією;
 - дешеві кабелі і з'єднання;
 - використання вигідних товарних технологій.
- можливість простого оновлення.

Майже всі поставлені задачі були вирішені, і навесні 1997 року з'явилися комп'ютери, обладнані з'єднаннями для підключення USB пристроїв.

Для спряження комп'ютера з лабораторною роботою автори вибрали мікроконтролер Atmel AT89C5131 з декількох причин. По-перше, це недорогий, але достатньо швидкодіючий мікроконтролер з широко відомим ядром 8051, що має 6 кінцевих точок. По-друге, для реалізації схеми потрібний мінімум додаткових пристроїв. Важливим є і наявність безкоштовного асемблера, компілятора мови C, програматора і драйверів для Windows/Linux. Дуже зручною є можливість програмування мікроконтролера не по SPI, а «напрямку» по USB-каналу.

AT89C5131 містить спеціальний апаратний модуль, який дозволяє йому забезпечувати обмін даними по USB. Для цього необхідні опорні синхроімпульси з частотою 48МГц, які генеруються контролером синхронізації. Ці синхроімпульси використовуються для формування тактових імпульсів з частотою 12 МГц із прийнятого диференціального потоку даних USB і пере-

дачі даних на великій швидкості, що відповідає вимогам до USB пристроїв. Формування синхроімпульсів виконується цифровою системою ФАПЧ (DPLL, Digital Phase Locked Loop). Коефіцієнт пропорційності задається бітами USBCDx регістра USBCLK.

Для реалізації основної програми автори використали основний алгоритм програми мікроконтролера [2, 183–235].

На початку роботи програми викликається функція UsbInit. Ця функція достатньо проста і полягає у виконанні наступного алгоритму:

1. Ініціалізація змінних.
2. Ініціалізація внутрішнього генератора.
3. Увімкнення USB-інтерфейсу.
4. Ініціювання процесу нумерації (для цього USB-пристрій відключається від шини і підключається знову не раніше ніж через 5мс).
5. Конфігурування нульової кінцевої точки (нульова кінцева точка конфігурується як управляюча).
6. Обнулення нульової кінцевої точки (reset).
7. Дозвіл переривань від нульової кінцевої точки (при цьому переривання не викликається, оскільки біт IE обнулено, але відповідні біти в регістрі UEPINT виставляються).

Після виконання цих операцій USB-пристрій готовий до конфігурування через канал нульової кінцевої точки.

Після ініціалізації програма циклічно виконує обробку сигналів побудки, переривань, обнулення і т.д. При цьому контролюється виникнення переривань від кінцевих точок за допомогою перевірки регістра UEPINT. Звичайно, основний цикл можна замінити на пустий, а всю роботу перенести в обробники переривань, але лінійна структура програми більш зручна і зрозуміла.

У програмі ми читаємо 8 байтів запиту і згідно з його кодом виконуємо виклик того чи іншого обробника. Всі байти запиту зберігаються у спеціальному буфері, описаному таким чином, щоб можна було легко отримати доступ як до окремих байтів, так і до полів запиту.

Важливо відмітити, що коли запит не підтримується USB-пристроєм, то для нього обов'язково має бути виконана процедура відхилення STALL(), як, наприклад, у гілці default.

Найскладніший обробник — обробник запиту GET_DESCRIPTOR. Для обробки цього запиту слід виконати наступну послідовність дій:

1. Визначити тип дескриптора відповідно до старшого байту поля wValue.
2. В залежності від типу дескриптора отримати вказівку на потрібну структуру (pbuffer) і розмір блока даних, що передається (data_to_transfer).
3. Розділити дані, що передаються, на блоки розміром максимального пакету (константа EP_CONTROL_LENGTH).
4. Виконати поблочну передачу даних хосту.

Основна складність полягає у розділенні пакета на блоки по EP_CONTROL_LENGTH байт. При цьому якщо розмір пакета буде кратним цьому числу, то необхідно додатково передати порожній пакет, що повідомляє хост про завершення передачі даних.

Функції опрацювання інших пакетів є значно простішими.

Опрацювання запиту GET_CONFIGURATION зводиться до передачі номера поточної конфігурації, що зберігається у змінній usb_configuration_nb. Цей номер зберігається при опрацюванні запиту SET_CONFIGURATION. Окрім збереження номера поточної конфігурації в обробнику SET_CONFIGURATION має відбуватись ініціалізація інших кінцевих точок, якщо вони існують.

Запит GET_STATUS дозволяє визначити стан USB-пристрою, інтерфейсу чи кінцевої точки. Стан, що повертається по запиту GET_STATUS, представляє собою 2 байти, один з яких повертає потрібну інформацію, а другий зарезервованний і завжди рівний нулю. Стан USB-пристрою визначає його реакцію на сигнал побудки і тип живлення, стан інтерфейсу зарезервовано і завжди рівний нулю, а стан кінцевої точки поки ігнорується.

За допомогою запиту SET_ADDRESS хост передає USB-пристрою його адресу на шині USB. Починаючи з моменту отримання цього запиту, USB-пристрій може відповідати хосту,

лише якщо хост звертається за цією адресою. На отримання своєї адреси USB-пристрій має відповісти установкою прапорця «пристрій адресовано».

У процесі опису дескрипторів слід пам'ятати, що двобайтові числа мають описуватись «навпаки», тобто спочатку молодший байт, потім старший. Для зручності запису можна використовувати спеціальний макрос wSWAP.

Отже, по запиті дескриптора конфігурації повертаються дескриптори конфігурації, інтерфейсів і кінцевих точок.

Залишається зібрати увесь код, відкомпілювати і завантажити на мікроконтролер. Після подачі сигналу Windows має знайти новий пристрій та почати пошук і установку драйверів.

Довгий час USB 2.0 був одним із найбільш популярних інтерфейсів, але нещодавно була сформована група USB 3.0 Promoter Group, яка займається розробкою нової версії популярного інтерфейсу. До цієї групи увійшли співробітники таких відомих компаній як Microsoft, Intel, Hewlett-Packard і NEC.

В Intel стверджують, що пікова продуктивність USB 3.0 сягне 5 Гбіт/сек. Новий стандарт підтримуватиме довші сполучні кабелі, а також буде обернено сумісним з усіма попередніми USB. Першу реалізацію стандарту буде виконано на апаратному рівні у вигляді окремих мікросхем.

За рахунок підвищення швидкості порту, до нього можна буде підключати все більше пристроїв, причому високошвидкісні пристрої, такі як зовнішні жорсткі диски, флеш-карти, відеокамери можна буде через концентратор підключати до одного фізичного роз'єму і його смуги пропускання буде достатньо для роботи всіх пристроїв одразу.

Єдиний мінус, який так і залишився у USB, — це високий відсоток завантаження процесора комп'ютера. Але розробники стверджують, що швидкісні характеристики USB 3.0 дозволяють пожертвувати частиною вільних ресурсів.

Поєднання розробленого авторами пристрою з USB 3.0, який розробляється, дає перспективи на майбутнє і дозволяє розробляти відповідне програмне забезпечення для лабораторних робіт, які можуть працювати в режимі on-line через USB-інтерфейс.

ЛІТЕРАТУРА

1. Гриценко М. І., Кучеев С. І., Пустовий О. М. Дослідження холестерико-нематичного переходу в оптичній комірці клиновидної форми. Матеріали дев'ятої міжнародної конференції «ФИЗИКА В СИСТЕМЕ СОВРЕМЕННОГО ОБРАЗОВАНИЯ» (ФССО — 07), м. Санкт-Петербург, 4–8 червня 2007 р. — т. 1. — С. 397–398.
2. Агуров П. В. Практика програмування USB. — СПб.: БХВ-Петербург, 2006. — 624 с.

Тетяна БСЛЯВЦЕВА, Оксана МЕНЬШИКОВА

ЗАСТОСУВАННЯ ІТ ТЕХНОЛОГІЙ НА УРОКАХ В ПОЧАТКОВІЙ ШКОЛІ

Розглянуті різні аспекти впровадження інформатики та ІТ технологій в початкову школу. Наведені приклади використання комп'ютерних презентацій, вікторин, ігор на уроках з інформатики, математики, природознавства, української та англійської мови, охорони здоров'я, малювання, читання. Приведені варіанти дослідницьких проектів для молодших школярів.

У сучасному суспільстві стрімко зростає кількість інформації і, внаслідок цього, життєвою необхідністю для кожного стає вміння отримувати, переробляти і зберігати її. В цьому сенсі курс інформатики та інформаційних технологій відіграє особливу роль. На сьогоднішній момент в світі інформаційних технологій, що вивчається у загальноосвітній школі, використовуються все більш складні, наукоємкі моделі взаємодії «людина — комп'ютер». Першорядною метою викладача інформатики та інформаційних технологій є, не тільки довести до кожного учня суму знань і умінь, але і розвинути навички професійної роботи із застосуванням інформаційних технологій. Навчання повинне бути направлене на закріплення і розвиток у учнів здатності засвоювати в органічній єдності природничонаукові та гуманітарні знання. Необхідними завданнями у викладанні інформатики і інформаційних технологій повинні бути: освоєння учнями сучасних інформаційних технологій, зокрема шляхом організації продуктивної праці уч-