Ключ защиты LPT-PRO

Олег Вальпа (Челябинская обл.)

В этой статье приведено описание аппаратно-программного ключа, подключаемого к LPT-порту для защиты программного обеспечения от пиратского копирования.

Аппаратный способ защиты программного обеспечения считается более эффективным по сравнению с программным способом защиты. Это объясняется в первую очередь тем, что тиражирование аппаратного ключа – непростое дело в отличие от простого копирования номера программного продукта. Предлагаемый здесь аппаратный электронный ключ, кроме того, имеет скрытый программный код, полностью защищённый от чтения. Таким образом, ворованной программой, защищённой таким ключом, просто невозможно будет воспользоваться.

Описываемый здесь аппаратный ключ под названием LPT-PRO выполнен на базе недорогого микроконтроллера серии AVR фирмы Atmel. Конструктивно ключ защиты изготавливается в виде переходника и подключается между LPT-портом компьютера и принтером. При этом он не мешает работе компьютера с принтером. Для лучшего понимания схемы и работы ключа рекомендую вспомнить все сигналы LPT-порта и их назначение. Эти сигналы и сам LPT-порт довольно подробно описаны в статье [1]. На рис. 1 приведена принципиальная электрическая схема ключа.

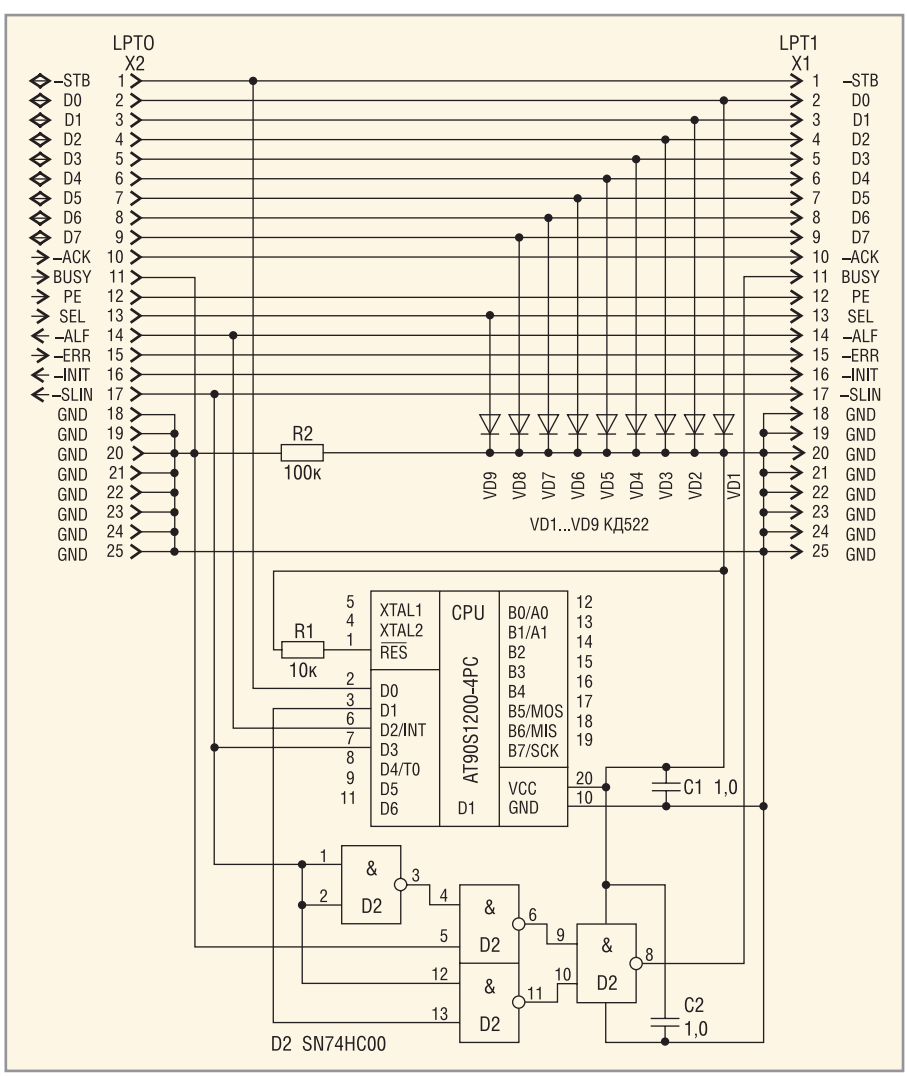


Рис. 1. Принципиальная электрическая схема аппаратного ключа

Основным элементом схемы является микроконтроллер D1. Он имеет широкий диапазон питающего напряжения 2,7...6 В и низкое потребление тока. Благодаря этому его можно питать непосредственно от сигнальных линий самого порта LPT. Для этого используются развязывающие диоды VD1...VD9, которые обеспечивают питание микроконтроллера от любой из восьми линий данных LPT-порта. Кроме микроконтроллера в схеме применяется низкоуровневая логическая микросхема D2, которая мультиплексирует входной для LPT-порта сигнал BUSY между микроконтроллером и принтером. Управление микроконтроллером происходит с помощью сигнала –SLIN. Этот сигнал имеет активное низкое состояние во время обращения компьютера к принтеру. Микроконтроллер контролирует состояние сигнала –SLIN после своей активизации и игнорирует все сигналы на других линиях, если этот сигнал активен. Для активизации микроконтроллера используется сигнал –AL, подключённый к входу прерывания. Этот же сигнал используется для синхронизации данных, передаваемых последовательно от компьютера микроконтроллеру электронного ключа по сигнальному проводу –STB. В свою очередь микроконтроллер передаёт последовательно данные компьютеру по сигнальному проводу BUSY. Оба эти процесса происходят одновременно за счёт разных линий связи приёма и передачи данных.

Элементы ключа LPT-PRO распаиваются на двухсторонней печатной плате, специально разработанной автором для этого устройства. На рис. 2 показаны верхняя и нижняя стороны этой платы.

Сборочные чертежи платы с габаритами платы и обозначениями элементов для установки с верхней и нижней сторон печатной платы приведены на рис. 3.

Рабочая программа для микроконтроллера ключа написана на Ассемблере для AVR и приведена ниже.

Текст программы для микроконтроллера ключа LPT-PRO

```
;*************************
;Программа lpt-pro для микро-
контроллера ключа защиты LPT-
PRO
```

```
;Версия: 1.0
;Дата: 27.02.2004
;***************************
.include "1200def.inc"
;Включить файл описания регистров
.LIST ;Включить листинг
.CSEG ;Начало кода программы
;Константы и определения переменных
.DEF Byte=r16
.DEF Count=r18
.DEF Byte0=r19
.DEF Byte1=r20
.DEF Byte2=r21
.DEF Byte3=r22
.DEF Byte4=r23
.DEF Byte5=r24
.DEF Byte6=r25
.DEF Byte7=r26
.DEF Temp=r31
;%%%%%%%%%%%%%%%%%%%
;Вектора прерываний
;%%%%%%%%%%%%%%%%%%%
.ORG 0 ;Стартовый адрес
 rjmp RESET ;Переход по сбросу
 rjmp EXT_INT0 ;Переход по прерыванию IRQ0
 rjmp TIM0_OVF ;Переход по прерыванию переполнения таймера
 rjmp ANA_COMP ;Переход по прерыванию аналогового компаратора
;@@@@@@@@@@@@@@@@@@@@
;Основной модуль
;@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
RESET:
 cli ;Запретить прерывания
 ldi Temp,0 ;Настроить порты на ввод
 out DDRB,Temp
 out DDRD,Temp
 ldi Temp,0xff ;Включить pull-up резисторы
 out PORTB,Temp
 ldi Temp,0x72
 out PORTD,Temp
 ldi Temp,0x40
 out GIMSK,Temp
 ldi Temp,0x30
 out MCUCR,Temp
UnLoop: sei ;Разрешить прерывания
 sleep ;Перейти в режим Power Down
 rjmp UnLoop ;Ждать импульса на входе INT0
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;Подпрограмма обработки прерывания таймера
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TIM0_OVF:nop
        reti
```

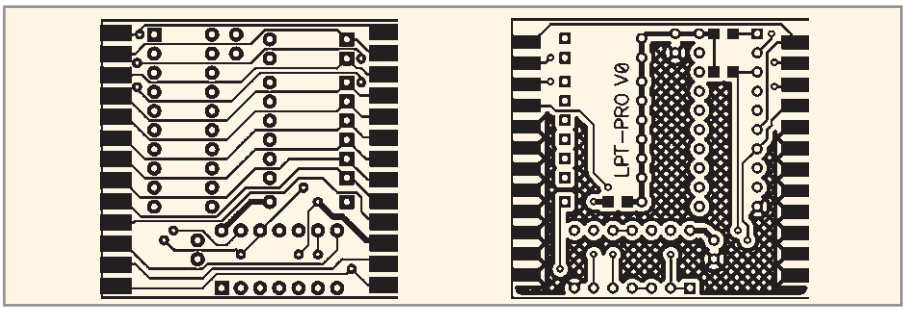


Рис. 2. Верхняя и нижняя стороны печатной платы

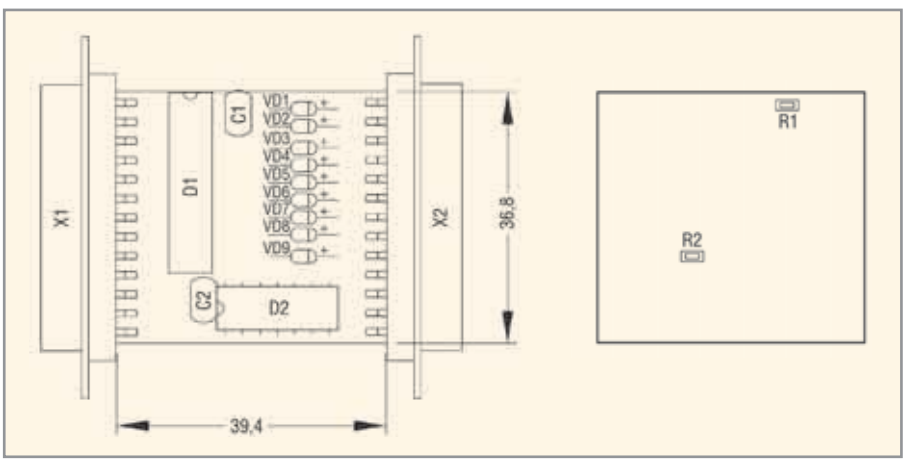


Рис. 3. Сборочный чертёж ключа LPT-PRO, вид сверху и снизу

```
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;Подпрограмма обработки прерывания компаратора
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ANA_COMP:nop
        reti
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;Подпрограмма обработки прерывания INT0
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
EXT_INT0:
 sbis PIND,3 ;Если PD3=0, обращение к принтеру
 reti
 ldi Temp,0x02 ;Включение ключа
 out DDRD,Temp ;Загрузка секретного слова
 ldi Byte0,0x50 ;P
 ldi Byte1,0x72 ;r
 ldi Byte2,0x6F ;o
 ldi Byte3,0x74 ;t
 ldi Byte4,0x65 ;e
 ldi Byte5,0x63 ;c
 ldi Byte6,0x74 ;t
 ldi Byte7,0x21 ;!
Main:
 rcall Byte8Exchange
 ;Сюда можно вставить шифрование данных
 sbis PIND,3 ;Если обращение к принтеру - спать
 rjmp RESET
 rjmp Main ;Переход на обмен данными
```

```
;============================
;Подпрограмма обмена 8 байтами данных
;============================
Byte8Exchange:
 mov Byte,Byte0
 rcall ByteExchange
 mov Byte0,Byte
 mov Byte,Byte1
 rcall ByteExchange
 mov Byte1,Byte
 mov Byte,Byte2
 rcall ByteExchange
 mov Byte2,Byte
 mov Byte,Byte3
 rcall ByteExchange
 mov Byte3,Byte
 mov Byte,Byte4
 rcall ByteExchange
 mov Byte4,Byte
 mov Byte,Byte5
 rcall ByteExchange
 mov Byte5,Byte
 mov Byte,Byte6
 rcall ByteExchange
 mov Byte6,Byte
 mov Byte,Byte7
 rcall ByteExchange
 mov Byte7,Byte
 ret
;============================
;Подпрограмма обмена одним байтом данных
;============================
```

```
ByteExchange:
ldi Count,8
Loop8: sbrs Byte,7 ;Передать бит
 cbi PORTD,1
 sbrc Byte,7
 sbi PORTD,1
Wait1: sbis PIND,2 ;Ждать поло-
жительный фронт
 rjmp Wait1
 sec ;c=1 ;Принять бит
 sbis PIND,0
 clc ;c=0
 rol Byte
Wait0: sbic PIND,2 ;Ждать отри-
цательный фронт
 rjmp Wait0
 dec Count
 brne Loop8 ;Цикл на 8 бит
 ret
.EXIT
```

Как видно из текста данной программы, микроконтроллер после инициализации регистров и портов переходит в энергосберегающий режим с низким потреблением тока. Пробуждение его из этого режима происходит по прерыванию INT на входе D2 порта D. После чего производится программный опрос состояния сигналов, инициализация регистров обмена данными секретным ключевым словом и обмен данными с компьютером.

Программа транслируется любым подходящим транслятором, например AVR Studio, свободно распространяемым фирмой Atmel и доступным на сайте www.atmel.ru.

Перед установкой на плату микроконтроллера D1 он должен быть запрограммирован. Можно установить на плату панельку на 20 контактов, в которую будет вставляться запрограммированный микроконтроллер D1.

Готовая прошивка для программирования микросхемы D1 в шестнадцатеричном формате приведена на рис. 4. Файлы lpt-pro.bin и lpt-pro.hex с прошивками микроконтроллера в бинарном и HEX-формате соответственно можно скачать с сайта журнала.

Программирование микросхемы необходимо производить на отдельном программаторе в параллельном режиме, поскольку только такой режим программирования позволяет установить режимы работы микроконтроллера с внутренней тактовой

Новости мира News of the World Новости мира

Новая технология имплантации позволяет считывать информацию на расстоянии до трёх метров

Zarlink Semiconductor сообщила, что в рамках проекта Healthy Aims European Union Framework VI начала исследования имплантируемой в тело человека антенны для систем Body Area Networks. Европейская программа Healthy Aims предусматривает разработку ряда медицинских имплантантов для пожилых людей и инвалидов. Имплантируемая в тело пациента радиометка может поддерживать связь с базовой станцией на расстоянии до трёх метров, что упрощает получение данных о состоянии пациента. Zarlink будет заниматься разработкой имплантируемой антенны и устройствами со сверхнизким потреблением.

Быстро развивающаяся область имплантируемой электроники требует проведения интенсивных исследований. Передача сигналов из имплантированного в тело человека устройства на внешний приёмник связана с решением ряда уникальных проблем, включая потребление, частоту, размер и биосовместимость. Антенны для беспроводных имплантантированных медицинских устройств должны быть чрезвычайно небольшими по размерам и весьма эффективными, чтобы гарантировать, что потери сигнала при прохождении через человеческое тело были минимальными.

www.zarlink.com

Новая LiveAP-технология для домашних Wi-Fi сетей

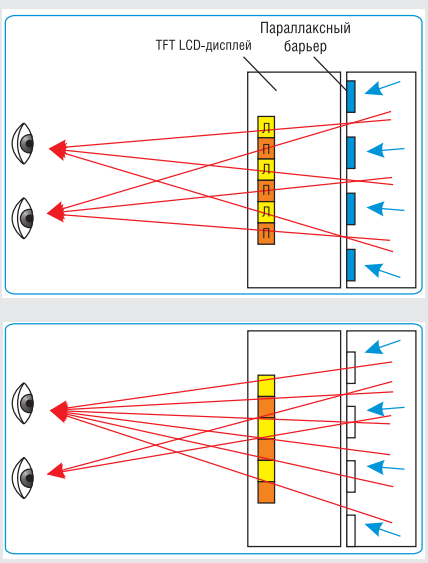
Компания Marvell прелагает новую технологию доступа для беспроводных домашних сетей – LiveAP™, которая поддерживает точки доступа AP (Access Point) в активном состоянии даже в том случае, если основное (хост) устройство – настольный PC, Media Center PC, игровая консоль или домашний трассировщик – выключены. В большинстве современных технологий, в частности, Soft AP, хост-прибор должен быть включён для активизации точек доступа. Однако большинство пользователей выключают свои PC, что делает неработоспособной не только беспроводную домашнюю сеть, но и прекращает доступ в Интернет.

LiveAP снимает это ограничение, обеспечивая непрерывное функционирование AP независимо от того, в каком состоянии находится хост-прибор. Чрезвычайно малое потребление LiveAP позволяет использовать для их питания «standby» ток головного прибора. LiveAP поддерживает последние стандарты безопасности беспроводных сетей, такие как WPA, WPA2, и расширение AES стандарта 802.11i. Кроме того, LiveAP поддерживает спецификацию QoS (Quality of Service) нового стандарта IEEE 802.11e. Marvell продемонстрировала свою LiveAP-технологию на выставке Networld+Interop Show, которая прошла в Las Vegas в мае этого года.

www.marvell.com

Sharp предлагает дисплей с трёхмерным изображением

Sharp сообщает о разработке дисплея с трёхмерным изображением, воспринимать которое можно без специальных очков. Переключение из двумерного в трёхмерный режим работы производится активизацией специального параллаксного барьера на жидких кристаллах (см. рисунок сверху), расположенного позади TFT LCD-матрицы. Параллаксный барьер расщепляет световой поток таким образом, что левый и правый глаз наблюдателя воспринимает разное изображение. Если же параллаксный барьер выключить, разделение изображений для левого и правого глаза исчезает, и дисплей переключается в стандартный режим работы (см. рисунок снизу).



http://sharp-world.com

частотой 1 МГц. Для этих целей хорошо подходит программатор «Стерх», производимый в г. Бердск, или ему подобный.

Кроме самой программы необходимо запрограммировать оба бита защиты и бит установки режима работы от внутреннего генератора RCEN, который будет работать на частоте 1 МГц. На рис. 5 приведена копия экрана, поясняющая данные операции при программировании микроконтроллера на программаторе «Стерх».

После установки и распайки всех элементов на плату необходимо проверить схему на правильность монтажа и отсутствие замыканий. После этого ключ защиты можно подключить к компьютеру. При подключении ключа компьютер не надо выключать. Вся конструкция ключа помещается в стандартный корпус типа GC-25 для переходников с разъёмами DB-25. Для тестирования данного ключа автором была написана небольшая программа на языке Си. Текст этой программы приведен ниже.

10000000	03	C0	15	C0	10	C0	11	C0	F8	94	F0	E0	F7	BB	F1	BB
10001000	FF	EF	F8	BB	F2	E7	F2	BB	F0	E4	FB	BF	F0	E3	F5	BF
10002000	78	94	88	95	FD	CF	00	00	18	95	00	00	18	95	83	9B
10003000	18	95	F2	E0	F1	BB	30	E5	42	E7	5F	E6	64	E7	75	E6
10005000	83	E6	94	E7	A1	E2	03	D0	83	9B	DE	CF	FC	CF	03	2F
10005000	17	D0	30	2F	04	2F	14	D0	40	2F	05	2F	11	D0	50	2F
10006000	06	2F	0E	D0	60	2F	07	2F	0B	D0	70	2F	08	2F	08	D0
10007000	80	2F	09	2F	05	D0	90	2F	0A	2F	02	D0	A0	2F	08	95
10008000	28	E0	07	FF	91	98	07	FD	91	9A	82	9B	FE	CF	08	94
10009000	80	9B	88	94	00	1F	82	99	FE	CF	2A	95	91	F7	08	95

Рис. 4. Прошивка для программирования микросхемы D1 в шестнадцатеричном формате

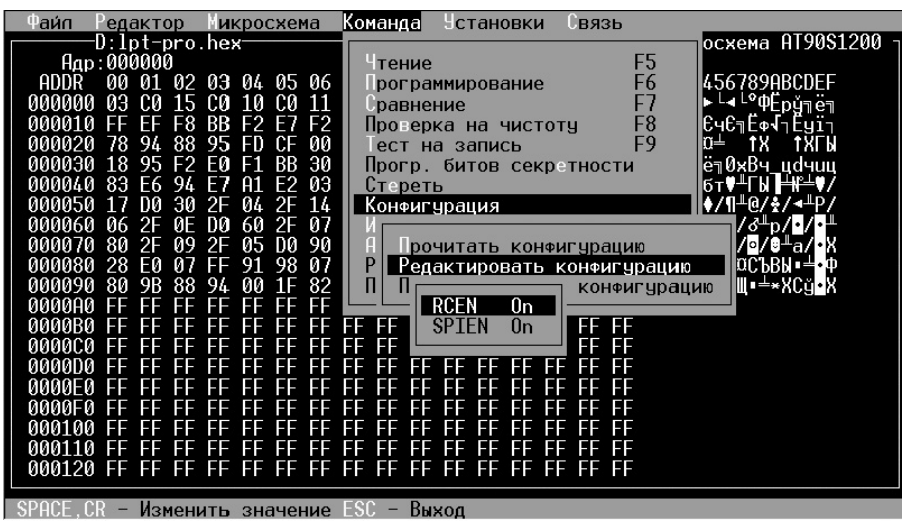


Рис. 5. Копия экрана при программировании микроконтроллера на программаторе «Стерх»

Текст программы для тестирования ключа LPT-PRO

```
//////////////////////////////
// Программа для lpt_pro проверяет наличие ключа защиты LPT
// Дата: 27/02/2004
// Версия: 1.0
//////////////////////////////
Подключение библиотек
#include <stdio.h>
#include <conio.h>
#include <bios.h>
#include <stdlib.h>
#include <dos.h>
#include <math.h>
// Описание функций
void CPU_On(void); // Разбудить процессор ключа
void CPU_Off(void); // Отключить процессор ключа
unsigned char Obmen(unsigned char); // Передать и принять байт ключа
unsigned char stp_2_y(unsigned char);// Возведение числа 2 в степень Y
// Описание переменных
#define BASE 0x378
//////////////////////////////
// Начало программы
//////////////////////////////
int main(void)
{
unsigned char c[8], i,
 s[8]="Protect!",
m[8]={0x55,0xAA,0x00,0xFF,0x0F,0xF0,0x5A,0xA5};
clrscr(); // Очистить экран
CPU_On(); // Разбудить процессор ключа
printf("\nЭтап проверки 1:\n");
for(i=0;i<=7;i++)
c[i]=Obmen(stp_2_y(i));
// Передать и принять 8 байт, записанных в CPU ключа
printf("Считано: ");
for(i=0;i<=7;i++) printf("%c ",c[i]);
for(i=0;i<=7;i++)
 if(c[i] != s[i])
 {CPU_Off(); printf("\nКлюч не обнаружен!\n"); return 1;}
 printf("\nКлюч обнаружен\n");
printf("Записано:");
for(i=0;i<=7;i++) printf("%2.2X ",stp_2_y(i));
printf("\n\nЭтап проверки 2:\n");
for(i=0;i<=7;i++)
c[i]=Obmen(m[i]);
printf("Считано: ");
for(i=0;i<=7;i++) printf("%2.2X ",c[i]);
for(i=0;i<=7;i++)
 if(c[i] != stp_2_y(i))
 {CPU_Off(); printf("\nОшибка разряда %d\n",i); return 2;}
 printf("\nЭтап проверки 2
 успешно завершён\n");
printf("Записано:");
for(i=0;i<=7;i++) printf("%2.2X ",m[i]);
printf("\n\nЭтап проверки 3:\n");
for(i=0;i<=7;i++) c[i]=Obmen(i);
printf("Считано: ");
for(i=0;i<=7;i++) printf("%2.2X ",c[i]);
for(i=0;i<=7;i++)
 if(c[i] != m[i])
 {CPU_Off(); printf("\nОшибка записи-чтения байта %d\n",m[i]);
return 3;}
 printf("\nЭтап проверки 3
 успешно завершён\n");
printf("Записано:");
for(i=0;i<=7;i++) printf("%2.2X ",i);
printf("\n");
CPU_Off(); // Отключение питания ключа
printf("\nКлюч защиты обнаружен и проверен");
printf("\nРазрешено запускать основную программу\n");
return 0;
}
```

```
////////////////////////////
// Функция передачи и приёма
байта ключа защиты
////////////////////////////
unsigned char Obmen(unsigned
char Dat8Out)
{
 int i,j;
 long l;
 unsigned char Dat8Buf, Dat8Inp
= 0;
 Dat8Buf = Dat8Out;
 for (i=0; i<8; i++)
 {
 Dat8Inp = (Dat8Inp << 1) +
((inportb(BASE+1)&0x80)==0);
utportb(BASE+2,0x02|((((128&Dat8Bu
f)==0)));
 for (l=0; l<2000; l++);
outportb(BASE+2,0x00|((((128&Dat8B
uf)==0)));
 for (l=0; l<2000; l++);
outportb(BASE+2,0x02|((((128&Dat8B
uf)==0)));
 Dat8Buf = Dat8Buf << 1;
 for (l=0; l<2000; l++);
 }
 return Dat8Inp;
}
////////////////////////////
// Функция побудки процессора
ключа защиты
////////////////////////////
void CPU_On(void)
{
 int i;
 long l;
 for (i=0; i<64; i++)
 {
 outportb(BASE,0xFF);
 for (l=0; l<20000; l++);
 outportb(BASE+2,0x00);
 for (l=0; l<20000; l++);
 outportb(BASE+2,0x03);
 for (l=0; l<20000; l++);
 }
 delay(100);
}
////////////////////////////
// Функция отключения процессора
ключа защиты
////////////////////////////
void CPU_Off(void)
{
 int i,l;
 for (i=0; i<128; i++)
 {
 outportb(BASE+2,0x0B);
 for (l=0; l<2000; l++);
 outportb(BASE+2,0x08);
 for (l=0; l<2000; l++);
 }
}
////////////////////////////
// Функция возведения числа 2 в
степень Y
////////////////////////////
unsigned char stp_2_y(unsigned
char y)
{
 unsigned char zuc;
 double zd, xd = 2.0, yd;
 yd=y;
 zd=pow(xd, yd);
 zuc=zd;
 return zuc;
}
```

Процесс обнаружения ключа компьютером сводится к приёму восьми байт от ключа и сравнению их с секретным кодовым словом. Одновременно с чтением этих восьми байт компьютер передаёт ключу другие восемь байт, которые микроконтроллер ключа сохраняет в своих регистрах. На втором и третьем этапе проверки ключа компьютер вновь читает восемь байт информации от ключа и сравнивает их с данными, переданными ключу ранее. Если все этапы проверки компьютером электронного ключа завершаются успешно, ключ считается обнаруженным и проверенным и можно разрешать работу основной защищаемой программе. В противном случае программа должна прервать своё выполнение и пользователь не сможет запустить защищённую ключом программу.

Как видно из текста программы, в ней присутствуют небольшие функции обмена с ключом защиты и простой протокол анализа полученных данных. В качестве базового адреса порта LPT используется описатель BASE. В программе ему присвоено значение 0x378, являющееся базовым адресом порта LPT1. Для работы с портом LPT2 или LPT3 необходимо соответственно изменить значение описателя BASE.

Результат работы программы в случае успешного обнаружения ключа будет выглядеть следующим образом:

```
Этап проверки 1:
Считано: P r o t e c t !
Ключ обнаружен
Записано: 01 02 04 08 10 20 40 80
Этап проверки 2:
Считано: 01 02 04 08 10 20 40 80
Этап проверки 2 успешно завершён
Записано: 55 AA 00 FF 0F F0 5A A5
Этап проверки 3:
Считано: 55 AA 00 FF 0F F0 5A A5
Этап проверки 3 успешно завершён
Записано: 00 01 02 03 04 05 06 07
Ключ защиты обнаружен и проверен
Разрешено запускать основную
программу
```

В случае если ключ защиты не подключён к компьютеру, сообщение программы будет иметь вид:

```
Этап проверки 1:
Считано:
Ключ не обнажружен!
```

Фрагменты данной программы вставляются в исходный текст программы, защищаемой от пиратского копирования, после чего она транслируется и становится защищённой приведённым здесь ключом. Файл с исходным текстом программы и исполняемый файл также можно скачать с сайта редакции журнала.

Для применения данного ключа в целях надёжной защиты программного обеспечения нужно всего-навсего заменить секретное ключевое слово Protect! на любое другое. Эту же операцию необходимо проделать и в программе самого микроконтроллера D1. Кроме того, для повышения секретности ключа можно модифицировать обе программы, изменяя при этом количество слов, передаваемых и принимаемых компьютером и микроконтроллером ключа, вставляя специальные задержки и т.п. Можно также ввести секретную маску для предварительного искажения и последующего восстановления данных или вообще изменить протокол общения компьютера с ключом. Таким образом, открытая архитектура описанного здесь аппаратного ключа защиты не даёт повода считать, что секрет данного ключа может быть легко раскрыт взломщиками ворованных программ. Даже не изменяя алгоритма работы имеющегося программного обеспечения, данный ключ имеет высокую степень защиты от пиратства. Таким образом, представленный здесь ключ обладает большой гибкостью и вполне может носить название аппаратно-программного ключа защиты.

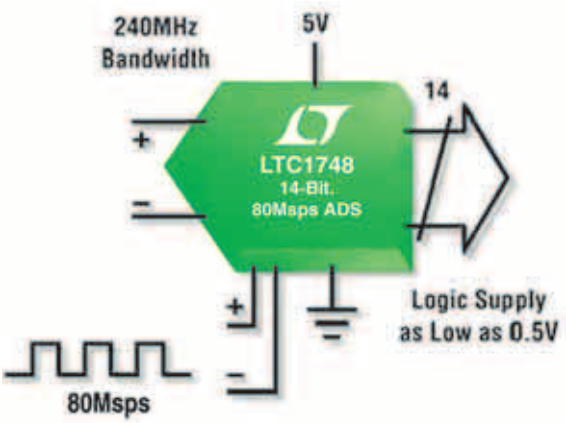
Литература

1. *Вальпа О.* Тестирование LPT-порта. Схемотехника. 2003. № 1. С. 44.

Новая серия ЦАП/АЦП — экономичность и быстродействие

LINEAR TECHNOLOGY

НОВЫЕ ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ АНАЛОГО-ЦИФРОВЫЕ ПРЕОБРАЗОВАТЕЛИ



ОСНОВНЫЕ ДОСТОИНСТВА

- Высокое быстродействие
- Большой динамический диапазон
- Биполярный дифференциальный вход
- Высокое отношение сигнал/шум
- Совместимость с 2 В/3 В/5 В/LVDS логическими интерфейсами
- Независимое питание выходных буферов 0,5...5 В
- Миниатюрный корпус TSSOP-48

Наименование	Разрядность	SNR, дБ	SFDR, дБ	$P_{потр}$, Вт	Скорость, Msps
LTC1745	12	42,5	91	0,38	25
LTC1746	14	76,5	91	0,39	25
LTC1743	12	72,5	85	1,0	50
LTC1744	14	77	87	1,2	50
LTC1741	12	72	85	1,275	65
LTC1742	14	76,5	90	1,275	65
LTC1747	12	72	85	1,4	80
LTC1748	14	76,3	90	1,45	80

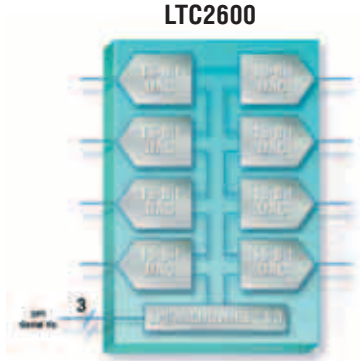
НОВЫЕ 8-КАНАЛЬНЫЕ ЦИФРОАНАЛОГОВЫЕ ПРЕОБРАЗОВАТЕЛИ

Наименование	Разрядность	$U_{см}$, мВ	$t_{уст}$, мкс	$I_{потр}$, мА	$U_{пит}$, В
LTC2600	16	±1	10	2,5/канал	2,5...5,5
LTC2610	14	±1	9	2,5/канал	2,5...5,5
LTC2620	12	±1	7	2,5/канал	2,5...5,5

Восьмиканальный ЦАП

Самая низкая стоимость канала преобразования

LTC2600



ОБЛАСТИ ПРИМЕНЕНИЯ:

- Телекоммуникационные системы
- Устройства распознавания образов
- Системы промышленной автоматики и управления
- Измерительная аппаратура
- Мобильная телефония и т.п.