

Константин Козлов

Параллельный порт вместо видеокарты

ВВЕДЕНИЕ

В настоящий момент в промышленности, на транспорте, в медицинских и контрольно-измерительных приборах, в системах автоматизации производственных процессов и еще во многих областях человеческой деятельности для отображения информации все чаще применяются плоскочелюстные электролюминесцентные дисплеи.

Компания Planar производит целый ряд моделей таких дисплеев с различными характеристиками. Одной из основных характеристик дисплея является его разрешение. Дисплеи с разрешением 640×480 точек, такие как EL468.480-AM1, EL468.480-AG1, EL468.480-AF1, EL468.480-AM8, или 320×240 точек (EL320.240.36) требуют для работы с ними плату видеоадаптера, BIOS которой поддерживал бы данную панель. Производители видеокарт, как правило, предоставляют BIOS и схему подключения, если видеоконтроллер, установленный на плате, поддерживает плоскочелюстные электролюминесцентные дисплеи. Такие дисплеи позволяют отображать большие объемы текстовой и графической информации в удобной для восприятия оператором форме, но сами отличаются при этом довольно большими габаритными размерами.

Существует множество приложений, где данные возможности отображения избыточны, где экономичнее применять дисплеи, имеющие меньшие габариты

и меньшее разрешение. Дисплеи данного класса компания Planar производит довольно много, можно сказать — целую линейку. Однако покупать EL-дисплей с разрешением, скажем, 160×80 точек и к нему видеокарту с заведомой функциональной избыточностью, я думаю, будет нецелесообразно и нерационально. Других же, более экономичных способов управления этими дисплеями, на первый взгляд, нет. Вместе с тем выход в данной ситуации имеется. И об этом пойдет речь далее.

КРАТКАЯ ХАРАКТЕРИСТИКА EL-ДИСПЛЕЕВ PLANAR

В данной статье будут рассмотрены малогабаритные электролюминесцентные дисплеи компании Planar, которые, на мой взгляд, незаслуженно имеют меньшую популярность, чем дисплеи с большим разрешением. Вероятно, это связано с мнением о трудностях с их подключением, отсутствием полной информации и примеров успешных применений. Все эти дисплеи имеют схожие эксплуатационные характеристики, одну и ту же схему подключения, за исключением не-

которых отдельных случаев. Их главное различие — разрешение дисплея (табл. 1).

Таблица 1. Примеры малогабаритных плоскочелюстных дисплеев с различным разрешением компании Planar

Тип дисплея	Разрешение, точек
EL160.80.50	160×80
EL160.120.39	160×120
EL240.128.45	240×128
EL320.240.36	320×240
EL480.60.43	480×60
EL480.240-PR2	480×240
EL640.200-SK	640×200

УПРАВЛЕНИЕ EL-ДИСПЛЕЯМИ ЧЕРЕЗ LPT-ПОРТ

В настоящее время на рынке микроэлектроники есть большой выбор специализированных контроллеров для управления электролюминесцентными дисплеями. При этом надо отметить, что такие контроллеры применяются, как правило, в узкоспециализированных устройствах, где они подключаются непосредственно к шине адресов и данных центрального процессора, в качестве которого чаще всего применяются микроконтроллеры компаний Intel (8085, 80C51 и др.) или Motorola (6800 и др.). Индустрия таких контроллеров хорошо развита, и сами производители на их базе предлагают множество решений.

Управлять подобными контроллерами довольно просто. Необходимо иметь восемь сигналов для передачи данных и четыре служебных сигнала для операций чтения/записи этих данных. Все промышленные одноплатные компьютеры (Octagon, Advantech, Fastwel и пр.), да и не только одноплатные, обязательно имеют в своем составе параллельный порт LPT. Во встроенных применениях подключаемый к нему принтер используется редко, порт по сути «простаивает» или ис-



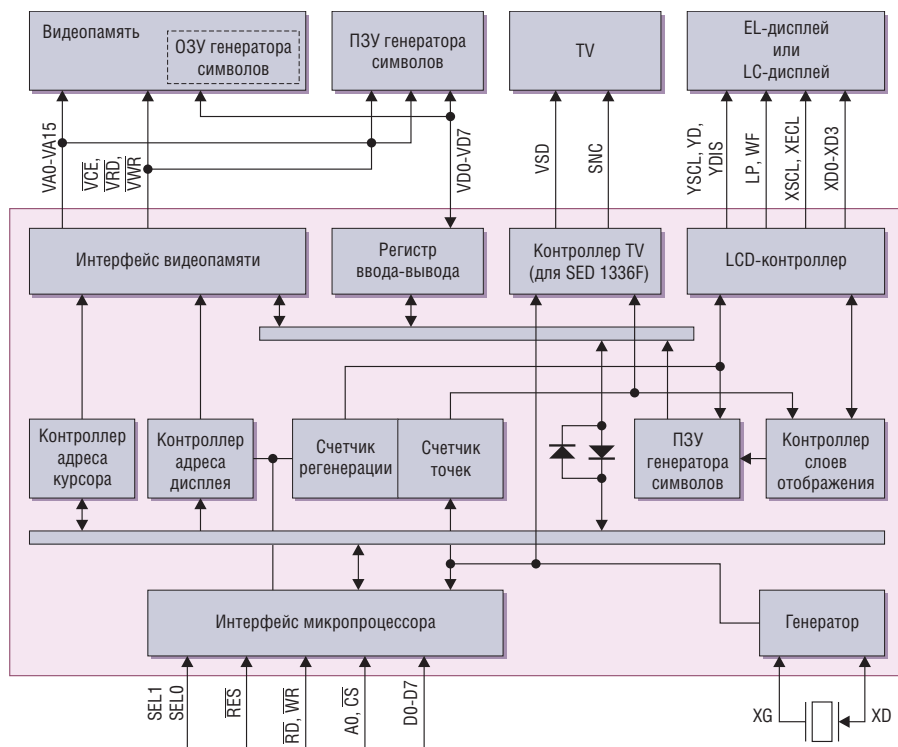


Рис. 1. Блок-схема контроллера SED1335

пользуется для других нужд, например для дискретного ввода-вывода. Поэтому идея использовать порт LPT для управления контроллером малогабаритных дисплеев напрашивается сама собой. С этой целью компания Planar предлагает плату SED1335 controller board (еще одно название — SED1335 DEMO), которая содержит в своем составе LCD-контроллер, поддерживающий как LC-, так и EL-дисплеи, и

статическое ОЗУ 32 кбайт. Остановимся на рассмотрении данного контроллера более подробно, ибо решение на его базе существенно более дешёвое и компактное, чем использование видеокарты.

КОНТРОЛЛЕР EPSON SED1335

Ядром платы SED1335 controller board является LCD-контроллер

SED1335 компании Epson. Далее приведены его краткие характеристики.

- Режимы отображения информации: текстовый, графический, комбинированный.
- Три графические плоскости в графическом режиме с возможностью их перекрытия.
- Максимальное разрешение: 640×256 точек.
- Программируемое управление видом курсора.
- Плавная вертикальная и горизонтальная «прокрутка» всего экрана или его части (scrolling).
- Внешнее статическое ОЗУ до 64 кбайт (на плате SED1335 controller board установлено 32 кбайт).
- Встроенный генератор символов: 160 символов размером 5×7 точек (к сожалению, кириллица отсутствует).
- Возможность реализации генератора символов во внешнем статическом ОЗУ (до 64 символов размером 8×16).
- Возможность реализации генератора символов во внешнем ПЗУ (до 256 символов размером 8×16).
- Интерфейс для подключения к процессорам 8080 или 6800.
- Диапазон рабочих температур: -20...+75°C.
- Напряжение питания: 2,7-5,5 В.
- Потребляемая мощность: 3,5 мА при V_{DD}=3,5 В.
- Габаритные размеры платы: 105×42×10 мм.

Пример программы контроллера SED1335 для дисплея EL240.128.45

```
//-----//
//      Файл IOFUNC.HPP //
//-----//
#if !defined (__IOFUNC_HPP)
#define __IOFUNC_HPP
#endif

#define STROBE    0x01
#define A0_ADDR  0x02
#define INIT     0x04

//      Определение набора инструкций
#define SYSTEM_SET      0x40
#define SLEEP_IN       0x53
#define DISP_ON        0x59
#define DISP_OFF       0x58
#define SCROLL         0x44
#define CSRFORM        0x5D
#define CGRAM_ADR      0x5C
#define CSRDIR_RIGHT   0x4C
```

```
#define CSRDIR_LEFT    0x4D
#define CSRDIR_UP      0x4E
#define CSRDIR_DOWN    0x4F
#define HDOT_SCR       0x5A
#define OVLAY          0x5B
#define CSRW           0x46
#define CSRR           0x47
#define MWRITE         0x42
#define MREAD          0x43

// Прототипы функций
unsigned char StatusRead (unsigned short port);
unsigned char DataRead (unsigned short port);
void DataWrite (unsigned char data, unsigned short port);
void CommandWrite (unsigned char data, unsigned short port);

void WriteString (char* string, unsigned short port);
void WriteByte (unsigned char byte, unsigned short port);

void StartCursorPos (unsigned char lowb, unsigned char highb,
                    unsigned short port);
```

Блок-схема контроллера SED1335 с необходимыми дополнительными узлами показана на рис. 1.

При внешней, казалось бы, простоте контроллер SED1335 обладает всеми узлами, присущими его «старшим братьям», которые устанавливаются на видеокарты.

Подключение платы контроллера к параллельному порту не вызывает больших трудностей: достаточно только изготовить кабель. Со стороны дисплея она имеет стандартный интерфейс, к которому можно подключить все перечисленные в табл. 1 панели, кроме EL240.128.45. Данный дисплей уже имеет встроенный контроллер SED1335 и статическое ОЗУ 32 кбайт, поэтому переходная плата ему не требуется. В табл. 2 показаны варианты разводки кабеля для некоторых типов разъемов, которые могут встретиться пользователю. Рекомендую при изготовлении кабеля от платы SED1335 к LPT-порту использовать кабельную сборку TCSD-12-S-20.00-01-F-N производства компании Samtec. Такие сборки можно приобрести в компании ПРОСОФТ. Со стороны дисплея также можно использовать готовый кабель Samtec TCSD-10-

Таблица 2. Разводка кабеля для подключения к плате SED1335

Разъём на плате SED1335, номер контакта	Обозначения сигнала	Разъём LPT-порта DB-25M (офисный ПК)	Разъём IDC-26 (CPU Octagon 5066, 60x0*, Fastwel CPU686)	Разъём питания платы SED1335
1				+12 В
2				+12 В
3				GND
4				GND
5				+5 В
6	RES-NC.		—	
7	/WR-STB	1	1	
8	/RD-INIT	16	6	
9	/CS-GND	18	10	
10	A0-AFD	14	2	
11	SelfTest-GND	19	12	
12	GND-GND	20	14	
13	D0-PD0	2	3	
14	D1-PD1	3	5	
15	D2-PD2	4	7	
16	D3-PD3	5	9	
17	D4-PD4	6	11	
18	D5-PD5	7	13	
19	D6-PD6	8	15	
20	D7-PD7	9	17	
21	SEL1-GND	21	16	
22	READY-NC.	—	—	
23	GND-GND	22	18	
24	LUMA-NC.	—	—	

* Для плат 60x0 в данном случае требуется переходная планка BOB.

```
//-----//
//      Файл IOFUNC.CPP  //
//-----//
#if !defined (__DOS_H)
#include <dos.h>
#endif
#if !defined (__STRING_H)
#include <string.h>
#endif

#if !defined (__IOFUNC_HPP)
#include "iofunc.hpp"
#endif

// Чтение флагов состояния
unsigned char StatusRead (unsigned short port)
{
    unsigned char byte = inportb(port+2);
    unsigned char data;
    outportb(port+2, ((byte | A0_ADDR) & -INIT) & ~STROBE);
    //A0=0,WR=1,RD->0
    data = inportb(port);
    outportb(port+2, ((byte | A0_ADDR) | INIT) & ~STROBE);
    //A0=0,WR=1,RD->1
    return(data);
}

// Чтение данных и позиции курсора
unsigned char DataRead (unsigned short port)
{
    unsigned char byte = inportb(port+2);
    unsigned char data;
    outportb(port+2, ((byte & -A0_ADDR) & -INIT) & ~STROBE);
    //A0=1,WR=1,RD->0
    data = inportb(port);
    outportb(port+2, ((byte & -A0_ADDR) | INIT) & ~STROBE);
    //A0=1,WR=1,RD->1
    return(data);
}

// Запись данных
void DataWrite (unsigned char data, unsigned short port)
{
    unsigned char byte = inportb(port+2);
    outportb(port+2, ((byte | A0_ADDR) | INIT) | STROBE);
    //A0=0,WR->0,RD =1
    outportb(port,data);
    outportb(port+2, ((byte | A0_ADDR) | INIT) & ~STROBE);
    //A0=0,WR->1,RD=1
}

// Запись команд (инструкций)
void CommandWrite (unsigned char data, unsigned short port)
{
    unsigned char byte = inportb(port+2);
    outportb(port+2, ((byte & -A0_ADDR) | INIT) | STROBE);
    //A0=1,RD =1,WR->0
    outportb(port,data);
    outportb(port+2, ((byte & -A0_ADDR) | INIT) & ~STROBE);
    //A0=1,RD=1,WR->1
}

```

D-20.00-01-F-N, который годится для всех дисплеев, кроме EL240.128.45 и EL640.200-SK.

Закончив изготовление соответствующих кабелей, подготовив подходящий источник питания, мы можем перейти к программированию контроллера. Замечу только, что потребление по цепи +12 В будет определяться только используемым EL-дисплеем, самому контроллеру +12 В не требуется.

Программирование контроллера Epson SED1335

На аппаратном уровне обращение к контроллеру ничем не отличается от циклов чтения/записи данных периферийных микросхем в устройствах, реализованных на микропроцессорах или микроконтроллерах, например, Intel 8080 или i8051. Временная диаграмма цикла чтения/записи приведена на рис. 2. Соотношения временных интервалов сигналов отражает табл. 3.

Для формирования команд записи или чтения необходимо всего четыре служебных сигнала, подаваемых на контроллер: A0, WR, RD, CS. Как видно из табл. 3, длительности этих сигналов имеют ограничения только

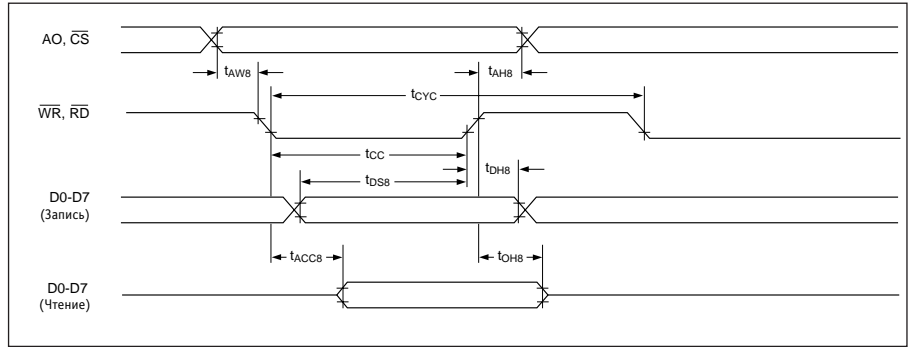


Рис. 2. Диаграмма цикла чтения/записи

Таблица 3. Соотношения временных интервалов сигналов (для диаграмм рис. 2)

Сигнал	Обозначение	Параметр	V _{DD} =4,5-5,5 В		V _{DD} =2,7-4,5 В	
			min	max	min	max
A0, CS	t _{AH8}	Время удержания адреса, нс	10	—	10	—
	t _{AW8}	Время установления адреса, нс	0	—	0	—
WR, RD	t _{CYC}	Системный цикл, нс	См. прим.	—	См. прим.	—
	t _{CC}	Длительность стробирующего импульса, нс	120	—	150	—
D0-D7	t _{DS8}	Время установления данных, нс	120	—	120	—
	t _{DN8}	Время удержания данных, нс	5	—	5	—
	t _{ACC8}	Время доступа сигнала RD	—	50	—	80
	t _{OH8}	Время запрета вывода	10	50	10	55

Примечание. Для команд управления памятью и системных:

$$t_{CYC8} = 2t_c + t_{CC} + t_{CEA} + 75 > t_{ACV} + 245$$

$$\text{Все остальные команды: } t_{CYC8} = 4t_c + t_{CC} + 30$$

```
// Отобразить текстовую строку по текущему адресу
void WriteString (char* string, unsigned short port)
{
    int i;
    CommandWrite(MWRITE,port);
    for(i = 0; i < strlen(string); i++)
        DataWrite(string[i],port);
}

// Записать байт в видеопамять по текущему адресу
void WriteByte (unsigned char byte, unsigned short port)
{
    CommandWrite(MWRITE,port); DataWrite(byte,port);
}

// Установить курсор в требуемую позицию
void StartCursorPos (unsigned char lowb, unsigned char highb,
unsigned short port)
{
    CommandWrite(CSRW,port);      DataWrite(lowb,port);
    DataWrite(highb,port);
}

//-----//
//      Файл EPC1335.CPP //
//-----//
#include <dos.h>
#include <bios.h>
#include <conio.h>
#include <stdio.h>

#include <stdlib.h>
#include <ctype.h>
#include <io.h>
#include <sys\stat.h>
#include <string.h>
#include <fcntl.h>
#include "iofunc.hpp"

#define LPT1      0x378

//-----//
//      MAIN //
//-----//
void main(void)
{
    int i;

    puts("Write to EL display");

    //Инициализация контроллера для дисплея EL240.128.45
    CommandWrite(SYSTEM_SET, LPT1);
    DataWrite(0x36,LPT1); // внутренний генератор символов,
    один дисплей
    DataWrite(0x07,LPT1); // ширина символа - 8 пикселей
    DataWrite(0x07,LPT1); // высота символа - 16 пикселей
    // 30 - количество символов в строке (30 симв * 8 пикс = 240
    точек)
    DataWrite(0x1D,LPT1);
    DataWrite(0x23,LPT1); // длина одной строки в байтах
```

Таблица 4. Основные команды контроллера SED1335

Тип	Команда	Код											Шестнадцатеричный код	Описание команды	Количество байтов	
		RD	WR	A0	D7	D6	D5	D4	D3	D2	D1	D0				
Управление системой	SYSTEM SET	1	0	1	0	1	0	0	0	0	0	0	0	40	Инициализировать контроллер и дисплей	8
	SLEEP IN	1	0	1	0	1	0	1	0	0	1	1	1	53	Перейти в «спящий» режим	0
Управление дисплеем	DISP ON/OFF	1	0	1	0	1	0	1	1	0	0	D	58, 59	Разрешение/запрет вывода на дисплей и мерцания	1	
	SCROLL	1	0	1	0	1	0	0	0	1	0	0	44	Установить начальный адрес и область отображения	10	
	CSRFORM	1	0	1	0	1	0	1	1	1	0	1	5D	Установить тип курсора	2	
	CGRAM ADR	1	0	1	0	1	0	1	1	1	0	0	5C	Установить адрес ОЗУ генератора символов	2	
	CSRDIR	1	0	1	0	1	0	0	1	1	CD1	CDO	от 4C до 4F	Установить направление движения курсора	0	
	HDOT SCR	1	0	1	0	1	0	1	1	0	1	0	5A	Установить позицию горизонтальной «прокрутки»	1	
Управление курсором	OVLAY	1	0	1	0	1	0	1	1	0	1	1	5B	Установить формат перекрытия экрана	1	
	CSRW	1	0	1	0	1	0	0	0	1	1	0	46	Установить адрес курсора	2	
Управление памятью	CSRR	1	0	1	0	1	0	0	0	1	1	1	47	Чтение адреса курсора	2	
	MWRITE	1	0	1	0	1	0	0	0	0	1	0	42	Запись в видеопамять	—	
	MREAD	1	0	1	0	1	0	0	0	0	1	1	43	Чтение видеопамяти	—	

по минимальным значениям. Сигнал \overline{CS} (chip select) мы использовать не будем, так как у нас всего один контроллер и выбирать нам из нескольких не приходится, поэтому вход данного сигнала надо просто замкнуть на «землю». Для формирования сигнала A0 будет использоваться выходной сигнал параллельного порта AFD (auto feed), сигнала \overline{WR} — сигнал STB, сигнала \overline{RD} — сигнал INIT. Для передачи данных D0-D7 используются соответствующие линии порта D0-D7.

Система команд контроллера предоставляет пользователю широкий выбор возможностей для реализации его замыслов по отображению данных.

Вместе с тем она проста и довольно легка в изучении. Список команд приведен в табл. 4. Приводить полную расшифровку системы команд представляется нецелесообразным, заинтересованные читатели могут её найти в «Руководстве пользователя контроллера SED1335».

Как видно из табл. 4, типичная команда управления состоит из байта управления и следующих за ней байтов данных. Так, команда SYSTEM SET (команда конфигурации системы) состоит из управляющего байта и восьми байтов данных. Отметим также, что запись управляющего слова производится при сигнале A0 (AFD), равном логической единице, а запись данных —

при A0, равном логическому нулю. В данной ситуации мы также не рассматриваем подробно саму команду, а отсылаем читателя к «Руководству пользователя».

С точки зрения интерфейса (если бы мы программировали с помощью процессора 8080), контроллер имеет два адреса для записи: базовый адрес для записи или чтения данных (A0=0) и базовый адрес+1 для записи управляющих команд (A0=1). Но применительно к параллельному порту мы будем просто говорить о состоянии сигнала A0 при записи команд или данных.

Во врезке приведен пример программы контроллера на языке C для дисплея EL240.128.45. В данном случае

```

DataWrite(0x7F,LPT1); // количество линий в кадре - 128
DataWrite(0x1E,LPT1); // диапазон адресов виртуального
экрana, мл. байт
DataWrite(0x00,LPT1); // ст. байт

// установить начальный адрес и отображаемую область
CommandWrite(SCROLL,LPT1);
DataWrite(0x0,LPT1); // начальный адрес, мл. байт
DataWrite(0x0,LPT1); // ст. байт
DataWrite(0x7F,LPT1); // 128 линий на блок прокрутки

// автоматический сдвиг курсора вправо
CommandWrite(CSRDIR_RIGHT,LPT1);

// горизонтальный сдвиг при прокрутке, колич. пикселей
CommandWrite(HDOT_SCR,LPT1);
DataWrite(0x0,LPT1);

// режим overlay
CommandWrite(OVLAY,LPT1);
DataWrite(0x01,LPT1); // текстовый режим, две плоскости

// установить позицию курсора
StartCursorPos(1,0,LPT1);

// очистить видеопамять
CommandWrite(MWRITE,LPT1);
for(i = 0; i < 0x7FFF; i++) DataWrite(0,LPT1);

// разрешить вывод данных на дисплей
CommandWrite(DISP_ON,LPT1);
DataWrite(0x05,LPT1);

// отобразить что-то полезное
WriteString("Hello, World!",LPT1);

StartCursorPos(30,0,LPT1);
WriteString("My internal character set:",LPT1);
// вывести все символы внутреннего
// знакогенератора на экран
StartCursorPos(60,0,LPT1);
for(i = 0; i < 256; i++) WriteByte(i,LPT1);

} // void main...

```

контроллер настраивается на отображение данных в текстовом режиме с использованием внутреннего генератора символов. Как видно, управлять контроллером довольно просто. Автор этой статьи потратил больше времени на изучение системы команд, чем на написание данной программы. Надо сказать, что этот пример приведен только с целью первого знакомства и никак не отражает все возможности контроллера SED1335. Программирование графических режимов можно предложить заинтересованному пользователю в качестве домашнего задания.

На самом деле отображение данных в виде мнемосхем или законченных картинок представляется более интересным, чем вывод текста. Размеры рассматриваемых дисплеев малы, и текстовые сообщения получаются, как правило, малоинформативными, поэтому здесь как нигде уместны графические режимы, тем более что в данных режимах возможны организация нескольких страниц с различными картинками в видеопамати контроллера и их быстрое переключение или наложение друг на друга.

ЗАКЛЮЧЕНИЕ

На сегодняшний день решение о применении параллельного порта для управления малогабаритными EL-дисплеями находит все больше поклонников.

Так, например, компания Fastwel, производящая оборудование в формате MicroPC, предоставляет со своими процессорными платами CPU188-5, CPU686, CPU686E специальный драйвер для управления платой контроллера SED1335 с поддержкой дисплеев EL480.60.43, EL320.240.36, EL240.128.45, EL160.120.39, EL160.80.50. Этот драйвер поддерживает без видеокарты вывод данных на дисплей в текстовом режиме 3. Таким образом, все программы пользователя, если они не предполагают прямого отображения данных в видеопамать, могут выводить информацию на дисплей зачастую даже без предварительной адаптации программного обеспечения. ●

Автор — сотрудник фирмы

ПРОСОФТ

Телефон: (095) 234-0636

Факс: (095) 234-0640

E-mail: root@prosoft.ru