



#29

PROSELYTIZER

**ZARULEM** [печатное слово]

НАД  
НОМЕРОМ  
РАБОТАЛИ

РЕДАКТОР  
WBR

ПОМОЩНИК  
РЕДАКТОРА  
BLASTOFF

ТЕХНИЧЕСКАЯ  
ПОДДЕРЖКА  
DJONI, XLAT

ГРАФИКА И  
ФОТОГРАФИИ  
BLASTOFF, INTERNET

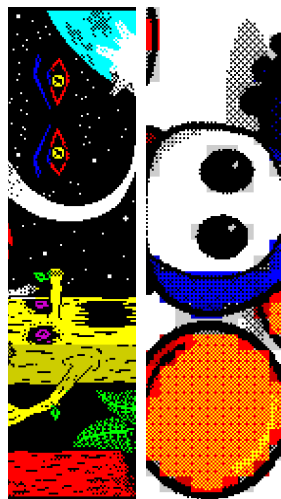
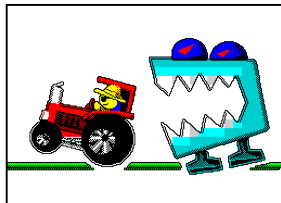
НА ОБЛОЖКЕ  
Яна

# Весна 2021! **ZaRulem** радует взор!!!

Приветствуем, наши хорошие!  
В этом году мы традиционно рады приветствовать вас со страниц нашего журнала! И этот яркий солнечный апрель придал нам хорошее ускорение в работе над номером, и в итоге вы держите его в руках или читаете с электронных экранов своих устройств. По нашему скромному мнению, выпуск исключительно удался! Нынешний номер журнала **ZaRulem**, вышел благодаря только нашим уважаемым авторам, которые читают его, как раз держа в руках! Привееееет вам! :)

А тех, кто ещё не написал статью, приглашаем к сотрудничеству, присылайте отзывы, советы и рекомендации по дальнейшему развитию журнала **ZaRulem** и желаем приятного чтения!





## Читайте в номере:

### 4 От редакции

Некоторые события, которые мы решили отметить

### 8 Пишем CSP Invito

Статья-пример использования SDCC-NOINIT

### 16 Обзор COVOX

Устройство COVOX и софта для него + приложение

### 23 Out of Compo 2021

Приглашение на самую откровенную пати

### 24 Интервью с ROMIK

Последнее интервью, подготовленное AAA :)

### 37 Спектрумист 2020

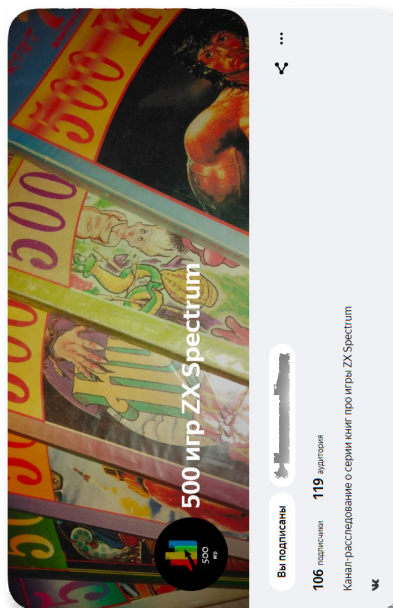
Отмечаем самых активных на платформе





**В**ремя летит - и эта фраза в очередной раз подтверждает скоротечность времени и событий, которые происходят вокруг нас с вами. Совсем недавно (год назад) мы работали в условиях коронавируса над весенним выпуском **ZaRulem**, по разным причинам был пропущен летний номер. Состоялся шикарный осенний выпуск и вот снова весна!

Дадим креативу бежать по клавишам клавиатуры и в авторской колонке расскажем о некоторых событиях ушедшего года и тех, что произошли в году текущем. **Печатное слово** снова с вами, приступаем!








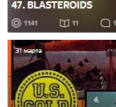
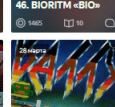
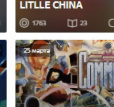
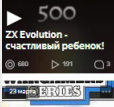
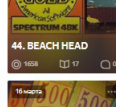
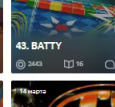
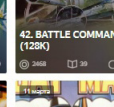
<https://zen.yandex.ru/zxgames>

Прежде всего расскажем о нашем проекте, который стартовал **в декабре 2020** года и нам очень нравится.

Многие помнят времена, когда игры для **ZX Spectrum** в основном доставались через доморожденных пиратов. Конечно, далеко не всегда игры приобретались или переписывались у товарищей с пониманием того, что тебя ждет при их запуске.

Но благодаря книгам от различных издательств, мы имели возможность оценить ту или иную игру по её описанию и старались найти понравившуюся. Одним из таких популярных изданий была серия книг **«500 игр»** от **«Ва Принт»**. Но время шло и у многих спектрумистов стало возникать подозрение, что далеко не все игры, описания которых приведены в указанных сборниках **«Ва Принт»** существовали на самом деле.



<p>2 человека</p>  <p><b>43. BLIND ALLEY</b></p> <p>1985 16 0</p>	<p>2 человека</p>  <p><b>47. BLASTEROIDS</b></p> <p>1981 31 1</p>	<p>12 игроков</p>  <p><b>46. BIORIM «Bio»</b></p> <p>1985 36 1</p>	<p>7 игроков</p>  <p><b>45. BIG TROUBLE IN LITTLE CHINA</b></p> <p>1983 29 2</p>
<p>3 игрока</p>  <p><b>43 Evolution - счастливый ребенок!</b></p> <p>1989 191 2</p>	<p>31 игрок</p>  <p><b>44. SPECTRUM ARK</b></p> <p>1988 17 0</p>	<p>16 игроков</p>  <p><b>43. BATTY</b></p> <p>1983 16 0</p>	<p>25 игроков</p>  <p><b>42. BATTLE COMMAND (128K)</b></p> <p>1988 39 2</p>
<p>2 человека</p>  <p><b>41. BATTLE OF BRITAIN</b></p> <p>1980 20 1</p>	<p>16 игроков</p>  <p><b>31-40. Промежуточный итог</b></p> <p>1985 12 0</p>	<p>18 игроков</p>  <p><b>40. BATMAN 3</b></p> <p>1982 34 0</p>	<p>11 игроков</p>  <p><b>39. BATMAN</b></p> <p>1989 31 2</p>

The image is a 4x4 grid of 16 small images. The top row shows various cartoon characters and landscapes. The second row shows a cartoon character in a red hat, a cartoon character in a blue hat, a cartoon character in a green hat, and a cartoon character in a yellow hat. The third row shows a cartoon character in a red hat, a cartoon character in a blue hat, a cartoon character in a green hat, and a cartoon character in a yellow hat. The bottom row shows a cartoon character in a red hat, a cartoon character in a blue hat, a cartoon character in a green hat, and a cartoon character in a yellow hat.



А вот весна года 2021 ознаменовалось парой очень интересных проектов от команды **Романа Сандаевского**. В своей **ВК-группе «ZX Spectrum | Новости, игры, общение»** он объявил о следующих инициативах:

**Первый проект** - это онлайн магазин по продаже различной продукции спектрумской направленности с финансовой поддержкой авторов, чьи творения используются в этой продукции. Инициатива очень интересная и нужная. На наш взгляд, давно напрашивался централизованный магазин с широким спектром товаров. Надеемся, что дело «попрёт»!

**Второй проект** с очень громким заявлением - это **«ZX Museum»**, в рамках которого ребята начинают собирать самую большую в мире коллекцию клонов **ZX Spectrum'a** и аксессуаров для него. Естественно, это не только музей реального железа, но и его онлайн версия. Согласитесь, это очень интересно.

В рамках проекта ребята принимают в фонд музея на безвозмездной основе, либо под личную ответственность неиспользуемые клоны **Спектрума** в презентабельном виде и рабочем состоянии, которые пока еще отсутствуют в коллекции.

Пожелаем удачи в этом благородном начинании!



Еще один факт хотели отметить так сказать в анналах спектрумовских событий! В конце января 2021 года некто **DragonsLord** поднял большую волну на форуме **ZX-PK.RU** бросив клич о поисках кодера для реализации амбициозного проекта - **WarCraft** на **Спектруме**! С его слов делается эта игра **«на изицах»** и даже представил публике образцы графики. К бурной деятельности подключились несколько известных кодеров, например **AloneCoder**. Промусолив тему порядка месяца, **DragonsLord** переключился на другой проект - **Elite в 3D** даже с высадкой на планеты!

Ну что ж, посмотрим что из этого получится, интересно ведь!!!

<https://vk.com/zxspectrum>

Сообщество любителей ZX SPECTRUM

**ZX SPECTRUM**



Теперь немного поговорим о перспективах. В этом году в **Новосибирске** планируется проведение **23-го** сборища спектрумистов **CrazySiberianParty (CSP)**. Традиционные сроки проведения - последние выходные июля. И в этом году эти дни выпадают на **31 июля и 1 августа!!!**

Если у вас имеются идеи и предложения по проведению **CSP**, если вы имеете возможность и желание поучаствовать в качестве организатора или

каким-либо иным способом, убедительная просьба связаться с редакцией. Все явки-пароли указаны на обложке к этому номеру журнала.

Кстати, стройка спортивных и культурно-массовых сооружений к Молодежному чемпионату мира по хоккею в 2023 году идет полным ходом в непосредственной близости от места проведения **CSP!**

На состояни города это сказывается удручающе, но местами становится лучше, чем было. Приезжайте, всё увидите своими глазами!



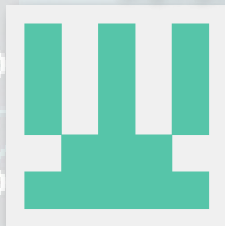
Не забывайте подписываться на нашу группу **ВКонтакте**, а также не забывайте про инстаграм **BlastOff:а** [https://www.instagram.com/blastoff\\_zx/](https://www.instagram.com/blastoff_zx/)

Это стоит сделать по нескольким причинам. Во-первых, на стене в группе **ВК** мы регулярно размещаем интересные новости и видосики. Появляются не только репосты, но и уникальный контент. А во-вторых, на нас периодически находит игривое настроение и мы разыгрываем какие-то сувениры. Да, постараемся делать это чаще для удовольствия подписчиков!

Кроме того, вы можете связаться с нами, прислать статью, новость или другой материал. Мы говорим это и по той причине, что работа над юбилейным 30-м выпуском журнала **ZaRulem Печатное слово** начинается прямо сейчас! Презентацию летнего номера планируем осуществить непосредственно в первый день проведения **CSP**. Посетители пати традиционно увидят журнал на несколько часов раньше, чем пользователи сети! Присылайте ваши статьи, рекламу и другой материал!!!!



development



Пишем



invitro

# Статья-пример SDCC-NOINIT

SFS

*Нет неразрешимых проблем,  
есть неприятные решения.*

**А**втор ни в коем случае не считает свое мнение единственно правильным и может его изменить или дополнить под действием разумных аргументов.

Также я не претендую на истину в последней инстанции, не считаю себя великим разработчиком систем как для **Z80**, так и систем вообще. Посему прошу простить и понять мои ошибки, неточности и прочие места, которые вам придутся не по вкусу. Если же вам есть что сказать — то можно всегда написать мне об этом.

И хочется сказать большое человеческое спасибо **WBRy** за конструктивную критику и вообще за то, что он смог подвинуть меня на написание **CSP2018 Invitro**.

## ВВЕДЕНИЕ

Написав **CSP2018 Invitro** я подумал, что это штука бесполезная и ненужная. И впал в уныние. Действительно, уже тысячи мне подобных, неподобных и бесподобных программистов писали на спектре миллионы всяких разных демо, где что-то летало, что-то мерцало и что-то звучало. Разумеется, как приглашение на фестиваль, эта программа заслуживает внимания. Но это внимание сведётся к одноразовому просмотру и не более. Но, выпив квасу, закусив шашлыком и придя в благодушное настроение, я решил, что труд

пропадать даром не должен.

В процессе написания **CSP2018 Invitro** мной были найдены немало ошибок в библиотеках **SDCC-NOINIT** и дописано немало полезных функций. Чтобы эти титанические усилия не пропали даром, я решил написать небольшую статью о том, как создаётся программа с помощью системы сборки проектов для **ZX SDCC-NOINIT**.

Итак, это статья о том, как работает **CSP2018 Invitro** изнутри. С разбором исходного кода и решений в нём применённых. Ну и конечно же — с примерами использования библиотечных функций.

Оговорюсь, что на деталях, описанных в документе «**ZX SDCC NOINIT Система сборки проектов для Z80 на базе пакета SDCC**» я останавливаться подробно не буду.

## 1. Из чего же? Из чего же? Из чего же сделана наша программа?

Если ответить формально — то из каталогов и файлов. Но такой ответ вряд ли устроит кого-то. Так что опустимся до подробностей. Согласно концепции **SDCC-NOINIT** каждая программа находится в отдельном подкаталоге каталога `apps/`. Поэтому все исходные коды проекта **csp2018invitro**, за исключением библиотек, находятся в каталоге `sdcc-noinit/apps/csp2018invitro/`, где `sdcc-noinit/` - корневой каталог системы сборки **SDCC-NOINIT**. Исходные тексты библиотек

## ПОЛЕЗНЫЕ ССЫЛКИ

Документ «**ZX SDCC NOINIT Система сборки проектов для Z80 на базе пакета SDCC**» доступен тут: <https://github.com/salextpuru/sdcc-noinit/blob/master/doc/sdcc-noinit.pdf>.

Исходные тексты **CSP2018 Invitro** расположены внутри системы сборки проектов для **ZX SDCC-NOINIT** тут: <https://github.com/salextpuru/sdcc-noinit/tree/master/apps/csp2018invitro>.

находятся в каталоге `sdcc-noinit/libsrc/`.

Будем рассматривать программу «сверху вниз». От внешнего вида, музыки и управления — к исходным кодам на С и Ассемблере.

### 1.1. Функциональные блоки программы

Чтобы понять дальнейшее, лучше всего запустить `csp2018invitro` на живом спектре или (для извращенцев) на эмуляторе. Вся эта красота, кажущаяся единым целым, состоит из нескольких частей. И каждая часть управляется отдельным программным блоком. Перечислим части экрана, соответствующие отдельным программным блокам.



**Блок 1** (сверху в центре экрана). Рисунок цветного логотипа **CSP** по которому время от времени пробегает светлая полоска.

**Блоки 2** (кружочки слева и справа от блока 1). Эти кружочки символизируют номер мелодии. Зелёный кружочек визуально показывает какая мелодия проигрывается в данный момент.

**Блок 3** (горящая свеча слева и надпись «28-29 Июля Новосибирск (C) 2018 Sfs» справа). Это на самом деле единый спрайт. Единственный движущийся объект связанный с этим спрайтом — пламя свечи.

**Блок 4** (плавно сменяющиеся надписи; «Портал `zx.pk.ru`» на рисунке). Это отдельный программный блок, который выводит эффект плавной смены спрайтов поверх блока 3. На месте блока 4 в спрайте блока 3 специально оставлена пустота.

**Блок 5** (надпись-эквалайзер **CSP2018**). Цвет букв меняется в зависимости от громкости и частоты музыки. Каждая буква соответствует одному параметру.

**Блок 6** (бегающая строка). О блоке 6A, являющемся частью блока 6, поговорим позже.

Кроме экранных процедур есть блоки управления музыкой, блок опроса клавиатуры, блок обработки прерываний, которые происходят в начале кадра 50 раз в секунду (или каждые 20 миллисекунд).

Каждый программный блок — это независимая программная единица, включающая в себя функции обработки того или иного типа данных, возможно данные и обязательно интерфейс для взаимодействия с другими программными блоками.

В идеальном случае каждый программный блок можно взять и без изменений перенести в другую программу.

### 1.2. Как собирать? Откуда запускать?

Заглянем в каталог `sdcc-noinit/apps/csp2018invitro/` и посмотрим, что там находится.

Концепция **SDCC-NOINIT** предполагает наличие для каждой программы двух файлов — файла конфигурации сборки `build.mk` и файла конфигурации программы `config.mk`.

#### Содержимое `config.mk`:

```
# Start address of module/program Адрес
загрузки и запуска
START=0x6000
# Stack Size Размер стека
SSIZE=0x400
# Interrupt on start: 0-disable, 1-enable
Прерывания запрещены
DSTART=0
# Store caller stack (1) or self stack (0)
Программа имеет собственный стек
CALLERSTACK=0
```

Надеюсь, тут всё понятно. Если перевести на человеческий язык, то можно сказать, что: программа загружается с адреса `0x6000`, запускается программа с того же адреса; в момент запуска программы прерывания запрещены; программа использует собственный стек размером 1Кбайт.

# 1 development

## Содержимое build.mk:

```
# App name Название программы (файлов .bin .tar и т.п.)
APP=csp2018inv

# Object files Список объектных файлов
OBJ=main.rel music.rel printscale.rel
OBJ+=cspLogo.rel spr2018.rel logos.rel uSctrollText.rel
OBJ+=musbtn.rel candle_flame.rel equalizer.rel
OBJ+=jungle_pt3.rel mus010_pt3.rel a_incom_pt3.rel kosmos_pt3.rel mozart_stc.rel opium_
pt2.rel

# LIBS (user-defined) if LIBS not defined then ALL libraies will be used.
# Список используемых библиотек
```

```
LIBS=libim2 libzxkbd libspr libay libfonts
librnd libz80
```

Тут всё тоже просто. Задаётся имя программы **csp2018inv**. Именно с таким именем будут создаваться все выходные файлы, в нашем случае **csp2018inv-basic.tap**, **csp2018inv.bin**, **csp2018inv.\$C** и **csp2018inv-code.tap**. Это всё наша программа в разных форматах.

Далее список объектных файлов с расширением **.rel**. Это откомпилированные файлы, получаемые из файлов на языке C (\*.c) или ассемблера (\*.s). Сборщик сам определяет как создавать .rel-файлы. Например, если указан файл **music.rel**, то сборщик ищет сначала файл **music.c** и, если такой файл существует, то компилирует его. Если файл **music.c** отсутствует, то ищется файл **music.s** и ассемблирует его. Ну и, если файл **music.s** тоже отсутствует, то сборщик останавливается с сообщением об ошибке. Кому интересно подробнее — читайте описание **SDCC-NOINIT**. И, наконец, задаётся список используемых библиотек. Если список не задан, то сборщик будет пытаться использовать все библиотеки, которые откомпилировал.

**Описание блоков, а также исходный код функций приведён в приложении.**

## 2. main() функция — начало всех процессов.

Как известно, программа на языке C начинается с функции **main()**. Это не совсем так, поскольку до этой функции выполняется произвольный код инициализации. Но, в

нашем случае, об этом можно забыть.

Итак, что же делает функция **main()**, находящаяся в файле **main.c** в нашей программе? Функции её просты и незатейливы:

- установка собственного обработчика прерываний и второго режима прерываний (**im2**);
- подготовка экрана (его очистка, вывод статических изображений на экран);
- инициализация бегущей строки;
- разрешение прерываний;
- и, наконец, запуск основного бесконечного цикла.

В основном цикле по очереди выполняется два действия:

- проверка доиграла ли до конца музыка и надо ли переходить к следующей мелодии;
- смена картинки справа от свечи через определенное время.

А как же остальные красявости? Музыка, свеча, блики на лого? А всё это выполняется в процедуре обработки прерываний. Причем с разным приоритетом.

Разберёмся как работает обработчик прерываний **im2userHandler()**, код которого находится всё в том же файле **main.c**. Главной особенностью обработчика прерываний является то, что в одном обработчике реализован запуск задач двух приоритетов. На самом деле, приоритетов задач можно организовать сколько угодно.

Если рассматривать с точки зрения приоритета выполнения, то в нашей программе три уровня приоритетов.



Уровень, самый низкий — это процедуры основного цикла функции **main()**: смена мелодий по кругу и смена картинок справа от свечи.

Следующий приоритет 1 — это визуальные эффекты (бегущая строка, пламя свечи, блик на логотипе, эквалайзер); опрос клавиатуры и смена мелодии по клавишам 1..6.

И, наконец, высший приоритет 2 — это проигрывание мелодии (вызов функции проигрывателя) и отсчет времени до смены картинок.

#### Лирическое отступление о стиле кода.

Обратите внимание, что функция определена как **static void im2userHandler()**, а не просто **void im2userHandler()**. Это является хорошим тоном, так как переменная или функция объявленная как **static** является видимой только в пределах одного \*.с файла. Удобство очевидно — в программе может быть сколько угодно переменных или функций с одинаковыми именами, если они находятся в разных файлах. Поэтому, если переменная или функция используется только внутри того же файла, где она определена — определяйте её как **static**. Избежите конфликтов имён. Да и вам удобнее, что не надо для одинаковых по смыслу функций выдумывать разные имена. Жизнь станет легче, прекраснее и удивительнее.

#### Конец лирического отступления о стиле кода.

Вернемся к приоритету выполнения задач. Очевидно, что если произойдет прерывание, то основной цикл прервется и начнется выполнение обработчика прерываний **im2userHandler()**.

В начале обработчика прерываний **im2userHandler()** объявлена загадочная переменная **static volatile uint8\_t lock**. Переменная эта статическая, то есть всегда существует в памяти. Разберёмся с её назначением чуть позже.

Итак, мы попали в обработчик прерываний и сразу же выполняем задачи с высшим, в нашем случае вторым, приоритетом. Сначала проверяем — не надо ли нам проиграть очередную ноту. Если проигрыватель включен — проигрываем ноту. Затем увеличиваем счетчик времени до смены картинок справа от свечи. Всё. Задачи с высшим приоритетом кончились.

Теперь нам надо выполнить задачи с более низким, первым, приоритетом. Сложность тут в том, что если во время выполнения этих задач вдруг возникнет прерывание, то надо такую задачу прервать, и вновь выполнить задачи с высшим, вторым, приоритетом. Но, в то же время, если задача с первым приоритетом уже выполняется, то повторно начинать её выполнение не нужно, будет плохо.

Поэтому просто разрешить прерывания после выполнения задач с приоритетом 2 нельзя. Надо сначала проверить — а не выполняется ли уже задача с приоритетом 1? Именно для этой цели и служит переменная-защёлка **lock**.

После выполнения задач с высшим приоритетом 2 значение переменной-защёлки **lock** проверяется.

Если оно равно нулю, значит функции с приоритетом 1 ещё не выполняются. В этом случае в переменную **lock** записывается 1, затем разрешаются прерывания и начинают по очереди выполняться функции с приоритетом 1.

Если проверка переменной **lock** показала, что она содержит ненулевое значение, значит функции с приоритетом 1 уже выполняются. В этом случае происходит просто выход из обработчика прерывания. После завершения выполнения всех функций с приоритетом 1 значение переменной **lock** вновь обнуляется, разрешая запуск этих функций по приходу следующего прерывания. Вот, собственно и всё, что содержится в файле **main.c**.

### **3. Бежали блики по реке**

Обратимся к логотипу, обозначенному цифрой «1» на рисунке. Это просто спрайт, любезно предоставленный организаторами **CSP2018**. Но чтобы он не был неподвижно-мёртвым, было решено его оживить. Жизнь ему придаёт время от времени пробегающий «блик». Этот блик — полоса по диагонали с повышенной яркостью.

В файлах **cspLogo.c** и **cspLogo.h** описаны сам спрайт и всего две процедуры работы со спрайтом-логотипом: процедура **logoToScreen()** — выводит логотип на экран и **logoAniStep()**, которая и организует вывод пробегающего блика.

# 12 development

Если с первой из процедур все предельно ясно, то вторая требует некоторых пояснений. Процедура **logoAniStep()** вызывается из обработчика прерываний с приоритетом 1 каждые 50 долю секунды. Иногда, когда процессорного времени не хватает один из вызовов может быть пропущен. Это не влияет на впечатление от блика. В файле **cspLogo.c** определён флаг **enable**, который указывает на то, выводится ли «блик» или нет. Переменная-счетчик **delay** отсчитывает время до очередной «пробежки блика» по спрайту.

Алгоритм работы процедуры прост. Если флаг **enable** равен 0, то при каждом вызове процедуры **logoAniStep()** происходит уменьшение счетчика **delay** на единицу. Если счетчик **delay** равен нулю, то запускается очередная «пробежка блика» по спрайту (флаг **enable** устанавливается равным единице). Если флаг **enable** равен 1, то при каждом вызове процедуры **logoAniStep()** происходит очередной «шаг блика», т. е. вывод диагональной полоски с нормальной яркостью и вывод в следующей позиции полоски с повышенной яркостью. X-координата вывода диагональной полоски определяется переменной **pos**.

Как только полоска «пробежала» весь спрайт, вычисляется задержка до очередной «пробежки блика» (от 1 до 255 пятидесятих долей секунды); позиция X устанавливается в начало (левый нижний угол спрайта); флаг **enable** устанавливается равным нулю, запрещая «пробежку блика». Основу манипуляции с атрибутами составляет библиотечная функция **void winSetAtr(uint8\_t x, uint8\_t y, uint8\_t w, uint8\_t h, uint8\_t attr, uint8\_t mask)**.

Это универсальная функция манипуляции с атрибутами. Параметры **x, y, w, h** задают соответственно координаты и размеры окна атрибутов на экране. Параметр **attr** задаёт атрибут, а параметр **mask** — битовую маску атрибута. Если нам надо поменять, например, только **INK**, то **mask=7 (007, 0x07)**, если **PAPER** — то **mask=56 (070, 0x38)**. Для **BRIGHT mask=64 (0100, 0x40)**, а для **FLASH mask=128 (0200, 0x80)**.

Разумеется, маски можно как угодно комбинировать. Например для **BRIGHT** и **INK** маска будет **mask=71 (0107, 0x47)**. Принцип понятен. В процедуре **logoAniStep()** функция **winSetAtr()** используется далёким от оптимального способом. Но так как быстрodeйствия хватает, то и оптимизация не нужна. Если вам после моих пояснений что-то не понятно — глядите в код. Там всё написано:)

## 4. Музыкальное сопровождение

Музыка. Как только она появляется в программе — сразу становится теплее на душе. Сколько копий было сломано о то, как её лучше играть, какое железо использовать и у кого мелодичнее звенят кошушки на морозе кто более музыкальный музыкант.

Сразу оговорюсь — я не музыкант вообще. Медведь плясал на моих ушах и голосовых связках. Потому все вопросы о достоинствах и недостатках мелодий — к их авторам. Я лишь воспроизвожу то, что создано другими. Потому сосредоточимся на коде, который позволяет нам слушать музыку.

В нашей незатейливой программе музыка не просто играет. Во-первых, мелодий несколько и играют они по кругу одна за другой. Во-вторых, пользователь может прямо указать какую из шести мелодий он хочет услышать. В третьих, необходимо отображать на экране какая именно из мелодий в данный момент проигрывается. Кроме того, в четвёртых, на экране есть ещё и «эквалайзер».

### 4.1. О сколько нам мелодий чудных наш плеер быстрый проиграл!

Итак, основа всего — несколько музыкальных проигрывателей. В нашем случае их два: один для проигрывания музыки в форматах **PT2/PT3** и второй для **STC-мелодий**. Все музыкальные штучки расположены в библиотеке **libay**. В программе имеется три файла с процедурами, имеющими отношения к музыке — это **music.c**, **musbtn.c** и **equalizer.c**.

Файлы с мелодиями содержат данные и на них мы останавливаться не будем. Файл **music.c** содержит описание всех мелодий и процедуры смены проигрываемых мелодий.

Каждый из музыкальных плееров имеет три стандартных процедуры: инициализация проигрывания мелодии; шаг проигрывания мелодии; проверка на окончание проигрывания мелодии. Алгоритм проигрывания мелодий крайне прост:

- инициализация очередной мелодии (вызов процедуры инициализации плеера с указанием на мелодию);
- затем каждое прерывание 1/50сек вызывается процедура проигрывание шага мелодии;
- если процедура проверки на окончание мелодии скажет: «мелодия проиграна!», то инициализация следующей мелодии.

И так — по кругу без конца.

Особенность нашей программы состоит в том, что используются мелодии в разных форматах. Чтобы не городить бог весть что, для каждой мелодии имеется запись (структура типа **sMusic**), которая хранит ссылку на данные мелодии, флаги, специфичные для плеера данного формата и код формата. Код формата нужен для того, чтобы знать — какой плеер использовать для данной мелодии. Все записи сведены в массив **musics[]**, хранящий все описатели мелодий.

Макрос **N\_MUSICS** — это просто количество мелодий, которое вычисляется при компиляции программы автоматически. Так следует поступать всегда, когда не знаете, будет у вас 2 записи в массиве или 20. Пусть рутину делает компьютер. И вы не ошибётесь.

Процедура **initNewMusic()** получает на вход ссылку на описатель мелодии (структуру **sMusic**) и вызывает функцию инициализации соответствующего плеера. Кроме того, процедура **initNewMusic()** выставляет обработчик прерывания **music\_im2h()** для соответствующего плеера. Именно этот обработчик вызывается для проигрывания очередного кусочка мелодии каждое прерывание с наивысшим приоритетом. Функция **checkEndOfMusic()** получает на вход ссылку на описатель мелодии (структуру **sMusic**) и вызывает функцию проверки на окончание мелодии соответствующего плеера.

Процедура **checkMusic()** примитивна до ужаса. В ней проверяется — закончена ли очередная мелодия с помощью вызова функции **checkEndOfMusic()**. Если мелодия закончена, то происходит переход к следующей мелодии. Номер проигрываемой мелодии хранится в переменной **musicNumber**.

Так как вызов процедуры **checkMusic()** не имеет жесткой привязки ко времени — она вызывается в фоне с низким приоритетом.

#### 4.2. Льется музыка-музыка-музыка, то печали, а то — веселя...

Как уже говорилось, пользователь может прямо указать какую из шести мелодий он хочет услышать. Для этого создана процедура **keyMusic()**. Она очень проста. На вход передаётся код клавиши, который считывается библиотечной функцией **zxKbdInKey()**. Если клавиша не нажата (код равен нулю), то ничего не происходит. Если нажаты клавиши от «1» до «6» — происходит инициализация указанной мелодии.

#### 4.3. Мелодий круглые монетки

Шесть круглых «монеток» (область 2) показывают какая из мелодий проигрывается в настоящий момент. Если вы обратили внимание, что я ни слова не сказал о процедуре **musBtnDraw()**, которая вызывается при инициализации новой мелодии в процедурах **checkMusic()** и **keyMusic()**. Именно эта процедура, описанная в файле **musbtn.c**, выводит на экран «монетки» мелодий. Все «монетки» выводятся синим цветом и только одна из них, соответствующая проигрываемой мелодии — зелёной.

Координаты «монеток» сведены в массив структур **pos[]**, описанный в том же файле **musbtn.c**. Это позволяет располагать «монетки» в произвольных координатах и легко и быстро менять их расположение при необходимости. Процедура **spr0\_out0\_attr()** выводит спрайт с атрибутами. Так выводятся все «неактивные монетки».

Процедура **spr0\_out0()** выводит спрайт без атрибутов, а уже знакомая нам процедура **winSetAtr()** устанавливает атрибуты



# 14 development

в нужные значения. Так выводится «монетка», указывающая на проигрываемую мелодию.

## 4.4. Мерцают буквы нотам в такт, отрада сердца — эквалайзер

Эквалайзер. Красиво, загадочно и сразу придаёт программе что-то такое, чего раньше не было. Штука крайне примитивная для музыкального сопроцессора АУ. Засунув нос в файл `equalizer.c`, видим, что вся реализация этого действия занимает жалкие несколько строк.

Надпись **CSP2018** (область 5) выводится один раз при инициализации экрана в процедуре `main()`. Сам спрайт и его вывод описаны в файле `spr2018.c`. Дальнейшие манипуляции с атрибутами производятся процедурой `equalizer_draw()` из файла `equalizer.c`. Так как буквы на спрайте имеют разные размеры, то не мудствуя лукаво, все координаты и размеры буковок сведены в массив структур `boxText[]`, что в файле `equalizer.c`.

Значения всех регистров музыкального сопроцессора АУ считываются библиотечной функцией `ayDumpGet()`. Данная функция считывает значения регистров в массив байт `ayDump[]`. Затем, для каждой из буковок, в соответствии со значением регистра музыкального сопроцессора выставляется атрибут согласно таблице `colorTable[]`.

Процедура `equalizer_draw()` вызывается из обработчика прерывания с приоритетом 1. То есть выше, чем фоновый цикл, но ниже, чем музыкальный плеер. Вот, собственно и всё об «эквалайзере».

## 5. Свечи загадочное пламя дрожит — что видишь в нём?

Свеча является частью изображения области 3. Вывод статической части свечи вместе с надписью производится кодом, содержащимся в файле `logos.c`. Но пламя свечи должно двигаться. Движение сделано примитивно — 5 спрайтов, поочерёдно сменяющих друг друга.

Процедура смены спрайтов пламени свечи описана в файле `candle_flame.c` и называется эта процедура `candle_flame_`

`animate()`. Она настолько проста, что и описывать её стыдно. Но надо — значит надо.

Счетчик `candle_flame_delay` отсчитывает 5 прерываний (одну десятую секунды) после чего происходит смена спрайта на следующий.

Счетчик `candle_flame_counter` хранит текущий номер спрайта пламени свечи.

Процедура `candle_flame_animate()` вызывается из обработчика прерывания с приоритетом 1, т.е. с тем же приоритетом, что и отрисовка эквалайзера. Вот, собственно и всё о «пламени свечи».

## 6. Сменяют лики и картины друг друга плавно по чуть-чуть

Следующий эффект - плавная смена изображения (область 4). Изображение сменяется примерно каждые 10 секунд. Процедура смены изображения `logos_check()` находится в файле `logos.c`. Эта процедура вызывается в фоне с низшим приоритетом. Счетчик `logos_timer` отсчитывает время до смены изображений. Отсчет (то есть увеличение таймера) производится в процедуре `logos_int500()`, которая вызывается в прерывании с высшим приоритетом.

Как только этот счетчик станет большим или равным 500 (то есть отсчитает 10 секунд), сразу же 8 раз вызывается процедура `void spr0_fade_step(const Sprite0* adr, uint8_t x, uint8_t y)`. Эта процедура заменяет случайные пиксели экрана по координатам `x` и `y` на соответствующие им пиксели спрайта. Размеры области экрана равны размерам спрайта.

То есть происходит вывод спрайта без атрибутов, но не всего, а только случайных пикселей. За 8 циклов меняются почти все пиксели указанной области на пиксели спрайта, чем достигается эффект плавной смены существующего изображения на содержимое спрайта. После этого вызывается процедура вывода спрайта с атрибутами `spr0_out0_attr()`, которая окончательно выводит спрайт и его атрибуты.

Все это долгие описывать, чем писать код. Так что читайте файл `logos.c`. В нём

все написано:)

## 7. Бежит строка и чуть дрожит — приветы в ней и пожеланья

И, наконец, бегущая строка (области 6 и 6A). Чтобы строка бежала, используются две основные процедуры:

**void printScale(uint8\_t x, uint8\_t y, uint8\_t scale, uint8\_t ch);** - вывод строки, увеличенным по вертикали в scale раз;

**void shiftLeftPix(const winShift\* w);** - скроллинг окна, заданного структурой winShift на один пиксель влево.

Собственно вывод бегущей строки производится процедурой **CheckShiftText()**, описанной в файле **uScrollText.c**.

Счетчик **ShiftTextDelay** отсчитывает количество прерываний до скроллинга строки на 1 пиксель влево.

Счетчик **winShiftDelayCounter** отсчитывает количество прокруток в состоянии ожидания.

Счетчик **symShiftCounter** отсчитывает количество пикселей в одном символе по горизонтали.

Особое внимание уделим переменной **winShiftStatus**, которая имеет два значения: **wshStatDelay** и **wshStatShift**. Данная переменная определяет — надо ли выводить очередной символ строки в область 6A или строка скроллируется впустую (то есть убегает влево, а справа символы не появляются).

Если **winShiftStatus==wshStatDelay**, то символы не выводятся, а строка очищается, путём «убегания» её остатка влево.

Если **winShiftStatus==wshStatShift**, то после скроллинга строки на 8 пикселей влево в область 6A выводится следующий символ. Область 6A является невидимой. То есть фон и тон на ней — черные и вывод очередного символа не виден. Это сделано для того, чтобы очередной выводимый символ плавно выезжал справа налево. Разумеется, так мы теряем один видимый символ строки, но это самый простой вариант.

Алгоритм работы процедуры **CheckShiftText()** не очень сложный. При каждом её вызове происходит следующее:

**Шаг 1.** Уменьшение на единицу и проверка счетчика **ShiftTextDelay**. Если он не равен нулю, что ничего больше делать не надо, выходим из процедуры. Иначе — переход к шагу 2.

**Шаг 2.** Сдвиг влево области 6 (включающей в себя область 6A) на один пиксель влево.

**Шаг 3.** Увеличение на единицу и проверка счетчика пикселей **symShiftCounter**. Если он меньше 8 (то есть символ до конца не проскроллирован), то ничего больше делать не надо, выходим из процедуры. Иначе — переход к шагу 4.

**Шаг 4.** Проверяем флаг **winShiftStatus**. Если он равен **wshStatDelay**, то значит мы находимся в состоянии ожидания и просто прокручиваем строку влево, не выводя новых символов, переходим к шагу 5. Если флаг **winShiftStatus**. Если он равен **wshStatShift**, то идет прокрутка строки с выводом, переходим к шагу 6.

**Шаг 5.** Увеличиваем счетчик количества прокруток в состоянии ожидания **winShiftDelayCounter**. Если этот счетчик больше или равен 256, то переходим к выводу текста со следующего вызова процедуры **CheckShiftText()**. Затем выходим из процедуры.

**Шаг 6.** Состояние прокрутки с выводом текста. Проверяем — дошли ли до конца строки. Если конец строки, то переходим в состояние ожидания при котором 256 раз прокручивается строка без текста. Т.е. текст просто убегает влево. Если не конец строки, то выводим очередной символ в область 6A.

### Заклочение

Ну вот вроде всё и расписал. Конечно, кому-то покажется, что я излишне много написал, кому-то наоборот, что сильно мало и неясно. Но, если кому-то что-то непонятно — смотрите код!

Сразу оговорюсь, текст может и наверняка содержит ошибки и неточности. Так что если вы их заметите — укажите мне, где я накосячил.) Не обижусь, а по возможности исправлю.

Смотрите, пробуйте, творите и ломайте. Наслаждайтесь, что вы можете написать круче, чем я :) ■

# Обзор музыкального устройства Covox на ZX Spectrum

Проект «Зазеркалье»



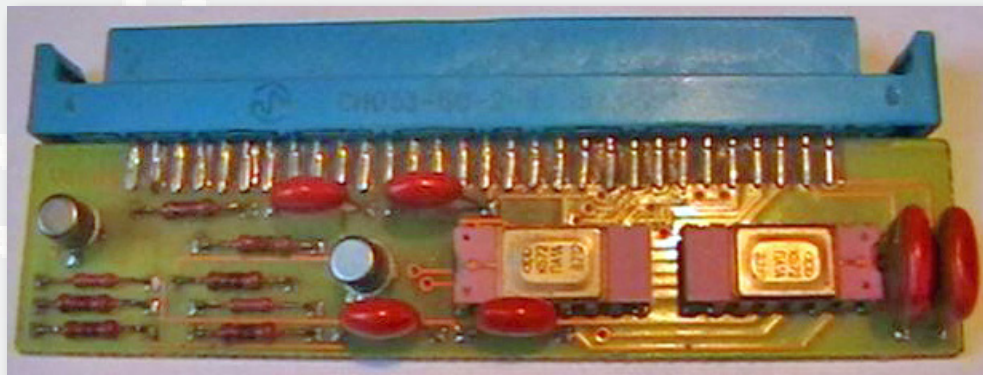
Тарасов А.Е. (TAE) и  
Вадим Чертков

**Н**е так давно в среде спектрумистов в очередной раз была поднята тема про **Covox**. Как «старый спектрумист» формально понимаю, о чём речь, но вот железного устройства у меня, никогда не было. В моём городе на точках продажи оно не прижилось, а для самостоятельного интереса на тот момент не было необходимого понимания и знания. Поиск информации выдал значительное количество разных схем и прочих технических вещей, но к моему большому удивлению, нигде не было описания для обывателя «что это такое и с чём его едят». Считаю, что стоит восполнить этот пробел.

На просторах интернета можно найти следующее определение.

**Covox** — простейшее устройство для воспроизведения цифрового звука. Представляет собой 8-разрядный резистивный ЦАП (*R-2R* схема), подключаемый к параллельному порту. Под таким названием оно изначально было выпущено в 1986 году фирмой **Covox** для **IBM PC**-совместимых компьютеров. Впоследствии это название стало применяться ко всем подобным устройствам. Примерно в то же время появились аналогичные устройства других производителей, а устройство стало использоваться на различных платформах, в том числе и на **ZX Spectrum**.

Другими словами **Covox** можно назвать первой в мире звуковой картой. О его достоинствах - на следующей странице.



Внешний вид платы Profi Covox



- Простота схемы и невысокая цена готового устройства.

- Хорошее качество 8 битного звука. Технические характеристики, которого зависят от возможностей компьютера и программного комплекса. Теоретически можно получить звук на уровне музыкальной карты **Sound Blaster 1.0**.

- Простота подключения к компьютеру. Обычно подключают к **LPT** порту, но могут быть и другие варианты.

- Простота программирования.

К недостаткам относится то, что сам **Covox** только воспроизводит звук, что требует значительной нагрузки на центральный процессор и несопоставимо большой, по сравнению с **AY**, расход оперативной памяти.

Но на **Спектруме** есть музыкальный сопроцессор **AY**, в чем, же выгода для нас?

Для полноценного ответа на этот вопрос нужно понять, в чем же разница между **AY** и **Covox**.

Музыкальный сопроцессор **AY** является трёхканальным генератором звука. Его правильнее называть синтезатором, обладающим дополнительными возможностями по синтезу звука, что бы музыка была более интересной. То есть основная программа только даёт команду **AY**, что нужно проиграть такую ноту с такими-то характеристиками и в таком-то режиме, остальное музыкальный сопроцессор всё делает сам. Именно при подобной схеме центральный процессор нагружается незначительно, что позволяет использовать **AY** в широком диапазоне задач.

**Covox** же воспроизводит только оцифрованный звук (**Samples**), который был ранее получен с помощью аналого-цифрового преобразователя (**АЦП**). Оцифрованные инструменты (**Samples**) представляют собой файл, состоящий из набора мгновенных значений амплитуды в каждый период выборки сигнала. Чем меньше период между выборками, тем выше частота оцифровки

инструментов, а значит, качество звука оцифрованного инструмента будет более близко к реальному.

Музыкальный сопроцессор **AY** также способен воспроизводить оцифрованный звук, но разрядностью только **4 бита**, тогда как **Covox** воспроизводит полноценный **8 битный** звук. Музыка, проигранная на **Covox**, будет звучать ближе к оригиналу.

Что же лучше: генерация звука или воспроизведение цифрового звук (**Samples**)?

В конечном итоге многое будет зависеть от мастерства и таланта автора мелодии. Но синтез нот музыкальным сопроцессором **AY** ограничен его техническими характеристиками, хотя он и позволяет выполнять над ними различные дополнительные эффекты, не требуя для этого дополнительного процессорного времени.

Тогда как **Covox** воспроизведёт совершенно любой звук, который ранее был оцифрован. Что позволяет воспроизводить голос и другие сложные звуковые эффекты.

По личным ощущениям, при прочих равных, качество мелодий на **Covox** выше. Но за это приходится платить значительной нагрузкой на центральный процессор и расходом оперативной памяти.

Считаю, что **Covox** является одним из самых недооцененных устройств на **Спектруме**, использование которого подняло бы звуковое оформление программ на совершенно новый уровень. С одной стороны это можно объяснить упомянутыми выше недостатками **Covox**: значительная нагрузка на центральный процессор и большой расход оперативной памяти. С другой стороны, память более **128 Кб**, стала появляться на клонах **Спектрума** с первой половины 90-х годов – вот оно место до 1 Мб под качественный звук. Проблема высокой нагрузки на центральный процессор может решаться использованием «**турбо режима**», который имеется на большинстве отечественных клонх. А так же организационными мерами –

# 18 hardware

качественными звуками можно озвучивать те места, когда в игре уже нет иного большого расхода процессорного времени, например «фаталити» или гибель главного героя, подбор предметов в играх типа «Диззи» и т.п. При этом если на компьютере не установлено устройство **Covox**, эти же эффекты можно вывести на АУ, пусть и в худшем качестве. Совместное использование АУ и **Covox** способно открыть совершенно новые горизонты. Например: музыка, в которой будет использованы оба устройства, не только позволит совместить достоинства обоих, но и увеличит общее число независимых голосов.

Схем **Covox** существует «вагон и маленькая тележка». Причем как самых простых, на 18 резисторах, так и с использованием специализированных микросхем цифро-аналогового преобразования (ЦАП). В СССР и России наиболее часто в этой роли применяется микросхема **572ПА1** (аналог **AD7520**). Надо отметить, что устройства собранные по схемам на 18 резисторах обладают неприятным эффектом в виде наличия при воспроизведении звука большого количества посторонних шумов. Хотя даже в таком виде звук будет значительно лучше, чем на АУ. При использовании микросхемы **572ПА1**

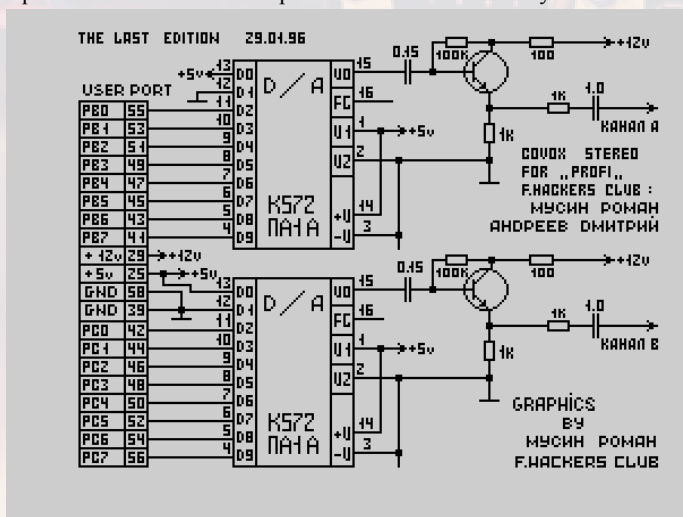
или аналогичной - звук будет ещё лучше, но и в этом случае полностью избавиться от посторонних шумов не получится, хотя услышать их можно только в наушниках. Существует обратно совместимое устройство **Soundrive**, представляющее собой четырёхканальный вариант **Covox**, в котором удалось полностью избавиться от всех шумов. Но оно уже значительно сложнее, а значит и дороже.

На разных машинах при использовании разных схемах подключения устройства **Covox** для вывода звука используются разные порты. Обычно для **Pentagon** используется порт **#FB**, а для **Scorpion** используется порт **#DD**. На **Profi** подключается стереоовокс по портам **#3F** и **#5F**. На других клонах, могут быть иные порты. Но общая схема вывода звука от этого не меняется. Поэтому, при использовании **Covox**, нужно уточнять в настройках программы порт вывода.

В настоящее время для **Спектрума** уже существует значительное количество музыкальных карт с большими возможностями. Но все они требуют уникального кода для своей поддержки. Однако большинство из них способно работать в режиме имитации устройства **Covox**. В силу этого можно использовать **Covox**, как

объединяющий элемент и на его базе реализовать поддержку сразу всех музыкальных карт написав универсальный код вывода музыки и звуковых эффектов.

Программная поддержка **Covox** на **Спектруме** к сожалению не очень велика. Под **TR-DOS** есть пару десятков **demo**, несколько игр были озвучены под **Covox**, существуют несколько музыкальных плееров разных форматов, редактор цифровой



музыки и редактор оцифрованных звуков. Большинство программ можно найти на сайте «Virtual TR-DOS» (<https://vtrd.in/>). Так же существует программы под ОС CP/M.

Так как я являюсь пользователем компьютера **Profi**, мне наиболее интересна версия **Covox** созданная для него. Давайте более подробно с ней и познакомимся.

Эту версию принято называть **Profi Covox**. Ключевой его особенностью является то, что это стерео устройство, то есть имеет два физических канала звука. Не смотря на то, что эта версия **Covox** разрабатывалась под конкурентный компьютер, его без проблем можно установить на любой клон Спектрума. С программированием **Profi Covox** разберёмся в одной из следующих статей. А пока ознакомимся с общими вопросами.

Из программного обеспечения в ОС CP/M существуют два плеера «WAV» файлов. Первый из них, распространяемый фирмой **Кондор**, способен проиграть файл максимального размера в **51 кб**. Второй из них за авторством **Черткова Вадима**, работает в ОС PQ-DOS и способен проиграть файл ограниченный лишь размером оперативной памяти. Так же есть музыкальная система «ADJ» (автор **Крестьяников А.А.**) для проигрывания музыкальных файлов одноименного формата и музыкальный редактор «Synth-Master» (авторы **Дмитрий Вольвач и Дмитрий Тищенко**) позволяющий редактировать и прослушивать музыку в форматах «ADJ» и «STM». Музыкальный редактор «Synth-Master» обладает полной совместимостью с аналогичными редакторами музыки в формате «STM» на компьютерах Amiga и IBM PC, что позволяет свободное переносить музыку с этих платформ. В музыкальной системе «ADJ» так же можно проиграть музыку с других платформ, но после конвертирования её в формат «ADJ» с помощью специальной утилиты.

В настоящее время существуют версии программ «ADJ» и «Synth-Master» адаптированные под музыкальную карту **General Sound**.

Вот характеристики программ:

	«Synth-Master»	«ADJ»
Количество голосов	4	4
Количество каналов воспроизведения	2	2
Разрядность данных на каждом из каналов	8 бит	8 бит
Разрядность воспроизводимых инструментов	7 бит	---
Максимальный размер одного инструмента	64 кб	---
Общий объем инструментов	Неограничено	до 152 кб
Количество инструментов в мелодии	до 32	до 31
Длина партитуры	до 128 партий	до 128 партий
Количество партий	до 99	до 99
Длина партии	64 такта. 1 Кб	64 такта. 1 Кб
Частота дискретизации в режиме TURBO	12 КГц	10 КГц
Формат готовых файлов	STM	ADJ

Как видно музыкальный редактор «Synth-Master» обладает немного лучшими характеристиками. Что ещё раз подтверждает тезис о том, что качество звучания **Covox** зависит не только от железных возможностей компьютера, но и от используемых программных алгоритмов.



## 20 hardware

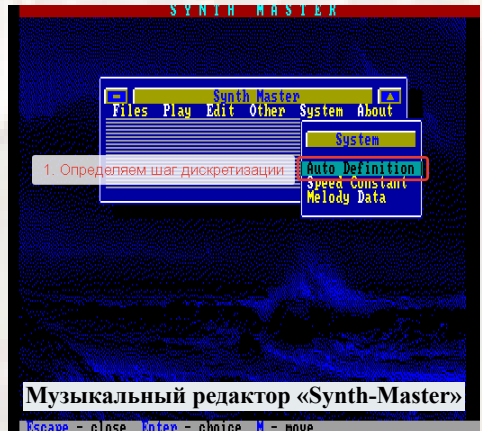
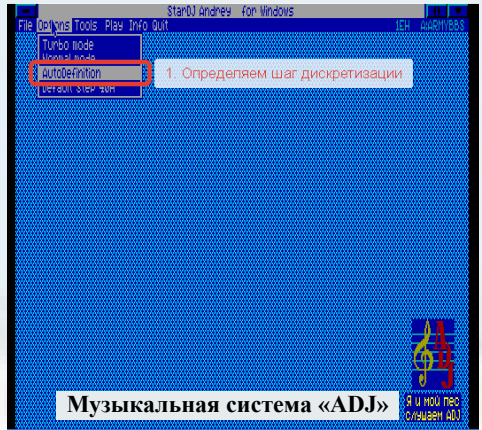
Обе программы имеют в комплекте хорошую документацию, так что сейчас ограничимся описанием минимальной последовательности действий необходимых для запуска проигрывания мелодии.

Музыкальной системы «**ADJ**» использует в своей работе резидентную библиотеку оконного интерфейса «**Windows**» и её нужно предварительно загрузить. Так что для запуска программы лучше использовать файлы «*adj.bat*» или «*adj-gs.bat*» для загрузки версии под Profi Covox или под музыкальную карту **General Sound**, соответственно.

Музыкальный редактор «**Synth-Master**» использует внутренний оконный интерфейс, для него грузить ни чего не нужно. Зато есть две версии программы «**Synth-Master**» обычная и **FAST** (*быстрая*). Обычная версия ([syntmast.com](http://syntmast.com)) поддерживает загрузки музыкальных файлов ограниченных только объемом оперативной памяти. Тогда как в **FAST** версии ([smfast.com](http://smfast.com)) объём загружаемого файла ограничен 128 кб для инструментов, но при этом качество звука на порядок выше. Для обеих версий есть адаптация под музыкальную карту **General Sound** ([sm-gs.com](http://sm-gs.com) и [smf-gs.com](http://smf-gs.com) соответственно).

После загрузки программы нужно настроить на скорость работы компьютера (эмулятора). В «**ADJ**» для этого нужно воспользоваться пунктом меню «**Options\AutoDefinition**», а в «**Synth-Master**» пунктом меню «**System\Auto Definition**». Для музыкальной системы «**ADJ**» данная операция строго обязательна, в противном случае качество воспроизведения музыки будет сильно страдать. Для музыкального редактора «**Synth-Master**» это действие необязательное, так как в большинстве случаев он сам производит корректную настройку на скорость компьютера. Но её нужно выполнить, в случае если кнопка «**Турбо**» нажималась уже после запуска программы.

Далее при необходимости через пункт меню **«File\Drive...»** выбираем дисковод, в котором находится диск с музыкой, и

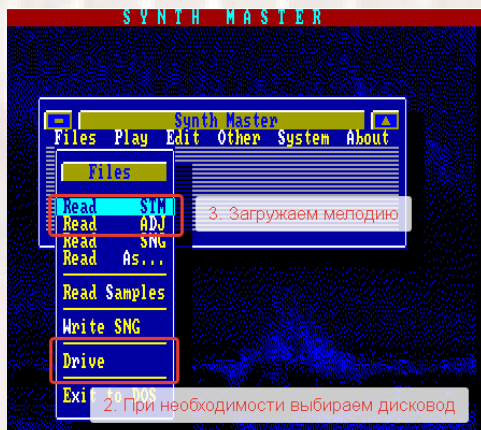
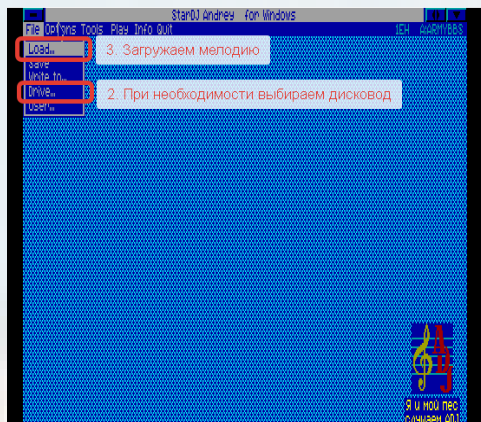


собственно загружаем музыку. Для чего в музыкальной системе «**ADJ**» выбираем меню «**File\Load...**», а в музыкальном редакторе «**Synth-Master**» выбираем пункт меню «**File\Read STM**» или «**File\Read ADJ**».

Для запуска проигрывания с начала мелодии, нужно выбрать пункт меню «**Play\Begin**».

Для прерывания проигрывания в обеих программах нужно нажать клавишу «Пробел». Для выхода на предыдущий уровень меню, если у вас **IBM** клавиатура: необходимо нажать клавишу «Esc», а если у вас стандартная клавиатура **Спектрума** то комбинацию клавиш «Shift+1».

У нас в наличии имеются исходники



музыкальной системы «**ADJ**» (автор **Крестьяников А.А.**). А вот исходников музыкального редактора «**Synth-Master**» (авторы **Дмитрий Вольвач** и **Дмитрий Тищенко**) к сожалению нет. В инструкции к программе указано, что авторы проживают в **Запорожье, Украина** и указаны телефоны для связи: **Дмитрий Тищенко (0612) 63-40-63**, **Дмитрий Вольвач тел. (0612) 68-04-82**. К сожалению, связаться с ними по этой информации не удалось. Поиск контактов в других местах так же не дал положительного результата. Так что у нас будет просьба к читателям журнала, если у кого есть какая-либо информация об авторах программы или возможность с ними связаться, поделиться её или попросить авторов выйти с нами на связь,

напрямую или через редакцию журнала.

Теперь скажем пару слов об эмуляторах. Поддержка **Profi Covox** есть в эмуляторе «**ZXMAK2**» и в «**UnrealSpecy**» к сожалению, только в версии «**0.37.1m0 30 XT keyboard controller jan 2009 molodcov\_alex**». К недостаткам эмулятора «**ZXMAK2**» следует отнести отсутствие турбо режима, что сказывается на качестве воспроизведения мелодий. А к недостаткам эмулятора «**UnrealSpecy**» версии **0.37** можно отнести отсутствие поддержки образов диска в формате «**PRO**», что в данном случае несущественно. Так что в приложении к статье вы найдете ссылку на архив эмулятора «**UnrealSpecy**» версии **0.37** настроенного на работу в режиме компьютера **Profi**. В дисковде «**A:**» будет «стоять» образ диска с музыкальной системой «**ADJ**», в дисковде «**B:**» образ диска с музыкальным редактором «**Synth-Master**».

Статьи, графика – это все очень хорошо и полезно. Но в современном мире есть еще один способ передачи информации, который мы пока обходили стороной - я говорю о видео роликах. Не смотря на то, что этот способ имеет, целый ряд недостатков, он же обладает и некоторым набором плюсов. Поэтому было, принято решение попробовать себя в этом направлении. Для чего был создан **YouTube** канал «**Зазеркалье**».

Компьютерщик, не совсем обычный человек. Жизнь его состоит из двух частей, и грань раздела проходит по стеклу монитора. Одна часть находится в реальном мире, а вторая в удивительной стране - **Зазеркалье**. И сложно сказать какая из частей главнее. Канал посвящен всем аспектам жизни по ту сторону монитора, в **Зазеркалье**. В первую очередь он будет отражать те грани этого мира, которые интересны нам, это: и **ZX Spectrum**, и **OS/2**, и различные языки программирования, и многое другое. Но мир **Зазеркалье** гораздо многограннее, так что если у кого-то вдруг появится желание отразить интересные ему грани, будем рады предоставить эту площадку. Сложно надеяться на частый выход видео, но точно они будут - это однозначно.

# 22 hardware

Первыми видео на канале «Зазеркалье» выложены примеры воспроизведения музыки на устройстве **Profi Covox**. Это видео клип на песню «Бухгалтер» группы Комбинация и видео клип на музыку по мотивам фильма «17 мгновений весны». Видеорядом идут кадры в формате расширенного экрана компьютера **Profi** с палитрой **16 из 256** цветов.

## Приложение:

- \* образ диска «.td0» с музыкальной системой «ADJ».  
Яндекс Диск: <https://yadi.sk/d/acvQcNGJQlwq-Q>;  
ВК: [https://vk.com/doc359059980\\_556932577](https://vk.com/doc359059980_556932577).
- \* образ диска «.td0» с музыкальным редактором «Synth-Master».  
Яндекс Диск: <https://yadi.sk/d/WF6cfRdpSrKLAw>;  
ВК: [https://vk.com/doc359059980\\_556932616](https://vk.com/doc359059980_556932616).
- \* архив эмулятора «UnrealSpeccy» версии «0.37.1 m0 30 XT keyboard controller jan 2009 molodcov\_alex».  
Яндекс Диск: <https://yadi.sk/d/x3AP05LxkwoJYA>;  
ВК: [https://vk.com/doc359059980\\_593424058](https://vk.com/doc359059980_593424058).
- \* видео клип на песню «Бухгалтер» группы Комбинация:  
<https://youtu.be/Ly99nw03V-g>
- \* видео клип на музыку по мотивам фильма «17 мгновений весны»:  
<https://youtu.be/cDhVB7n6jsw>



FidoNet: Tarasow Aleksey 2:5053/57



E-mail: [tae\(очень злая собака\)azimut21.ru](mailto:tae(очень злая собака)azimut21.ru)

Также много материала форуме и в группе ВК.

\* <https://zx-pk.ru/forums/102-profi.html>

\* <https://vk.com/profi1024>

\* YouTube канал «Зазеркалье»

<https://goo-gl.ru/6rA7>







# **OUT OF COMPO ART PARTY 10 JULY MOSCOW 18+**

**[OOC.RETROSCENE.ORG](http://OOC.RETROSCENE.ORG)**

**10 ИЮЛЯ 2021 ГОДА  
В МОСКВЕ  
ВПЕРВЫЕ БУДЕТ ПРОХОДИТЬ  
**OUT OF COMPO**  
В ОФЛАЙН ФОРМАТЕ.  
ПРИГЛАШАЕМ ВСЕХ ПРИНЯТЬ  
УЧАСТИЕ, БУДЕТ МНОГО  
ИНТЕРЕСНЫХ КОНКУРСОВ. ЖДЁМ  
ВСЕХ В ГОСТИ. ВСЯ ИНФОРМАЦИЯ О  
ПРОВЕДЕНИИ **OUT OF COMPO** БУДЕТ  
ДОСТУПНА НА ОФИЦИАЛЬНОМ САЙТЕ  
**[HTTPS://OOC.RETROSCENE.ORG/](https://OOC.RETROSCENE.ORG/)****

**ТАКЖЕ ДОСТУПЕН ИНФОРМАЦИОННЫЙ КАНАЛ  
**[HTTPS://T.ME/OUTOFCOMPO](https://t.me/outofcompo)****

ТЕХНО - ЭТО  
ПОДСОЗНАТЕЛЬНАЯ  
ТОСКА МОЛОДОГО  
ПОКОЛЕНИЯ ПО  
ШУМАМ  
РАБОТАЮЩЕГО  
ЗАВОДА



# ИНТЕРВЬЮ

# С

# ROMIK

# ИЗ

# MUSICSOFT

Материал подготовил ААА

Когда я открывал видео канал **ZXAAA** для размещения роликов с демонстрациями силы пентагона, то не мог предположить что в последствии это место превратится в пруд с рыбками, где я периодически буду ловить на крючок с демо старых спектрумистов, чтобы спросить, как у них дела. Индексация **google** настолько хорошо работает, что любая старая демка, при правильно написанном авторе и названии попадает в мобильный телефон её автора. Ибо свои ники, как правило, люди с годами не меняют, используя их на форумах и в названиях своей **gmail** почты. Поисковые запросы часто соответствуют названиям своих старых любимых демонстраций, а также иными интересами в сфере Спектрума. Немудрено, что я стал общаться с авторами демоснтраций, а иногда на мой крючок попадает добротная рыба, например в этот раз попал сам **MusicSoft**. Я сразу же с ним заговорил, и он написал подробную историю своей жизни. В итоге у нас получился рассказ о 90-х годах, который будет

интересен старым спектрумистам и новичкам исследующим зарождение спектрумизма в **CНГ**. Садитесь удобнее, наливайте пиво, получайте удовольствие, окунувшись с головой в **90-годы**. Стилистика автора не изменялась, чтобы не превращать рассказ в банальное сухое интервью.

Первый комп у меня появился, кажется, где-то между **1991** и **1992** годами. Это был один из многочисленных клонов **РК-86** на микросхеме **KP580 ВМ80А**, выпускавшийся серийно - **«Микроша»**. Сначала он был у одноклассника **Паши (Kent of Musicsoft)** - **Гордеева Александра**, который не находился в нашей тусе, хотя у него потом появился **Спектрум** и он на нём что-то кодил, но только для себя. В ту пору я ходил по всяким компьютерным клубам и играл во что-то типа **Atari** по 10 копеек за минуту, но мы нашли клуб с **БК0010 (или 0011)** по 85 копеек в час в районе **Гольяново** города **Москвы**. Все равно хотелось что-то своё. Годы были после развала **Союза «голодные»**, жрать было нечего, в школах раздавали гуманитарную помощь. Нам давали две большие банки колбасного фарша и два пакета сухого молока.



Колбасный фарш и сухое молоко  
– валюта спектрумиста 90-ых!

А у меня родители (и мать и отец) всю жизнь были на руководящих должностях в «райпищеторге» одного из центральных районов Москвы, холодильник был всегда забит «дефицитом» и этот колбасный фарш с сухим молоком были на фиг не нужны. Поэтому я с чистой совестью отнёс все это добро на «толкучку» около Центрального Детского Мира, и за несколько часов хождения по кругу с этим добром в руках у меня все купили.

Этих денег мне хватило ровно на половину «Микроши», вторую половину добавили предки и я её благополучно купил. К телеку подключал по ВЧ через модулятор, ничего мудрить не надо. Так я несколько месяцев благополучно играл в игры с текстовой псевдографикой типа Tetris, Xonix. А у Паши (Kent) в то время был



привезенный из-за бугра «родной» Sinclair ZX-spectrum со встроенным магнитофоном.

В общем, в Микрошу я поиграл не долго. Ко мне играть домой приходил один знакомый, а у него откуда-то был «Ленинград-1», который он не мог подключить к телеку. Ну Спектрум же, RGB надо мудрить. Модуляторы какие-то на радиорынке продавались, но я не помню, что бы хоть один нормально заработал по ВЧ. В итоге мы с ним просто поменялись компами. Год примерно тот же, между 1991 и 1992, как сейчас помню, что в 1991 я ещё клеил модельки под балет «Лебединое озеро», который крутили по телевизору и первый

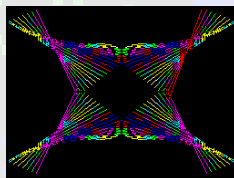
путч, а в 1993 году уже рвался к белому дому в то время, как Kent валял что-то на Спектруме. Вот так у меня появился Ленинград-1, который потом был с помощью телемастера подключен к телевизору



Silelis.

Начались покупки кассет, кручение головок кассетного плейера (магнитофона у меня не было), в общем, обычные будни подростка, у которого появилась «домашняя игровая приставка». Конечно, мне хотелось дисковод, ибо кассеты выносили нервную систему. Заодно изучил бейсик, написали игру «доллар\$» на бейсике. Сверху «сыпались» значки «\$», снизу была «тележка», на подобие Арканоида, где надо было ловить. Помню, всё лето с утра до вечера «рубилась» в ELITE, и в какой-то момент «хакнули» от неё save game, elite записывало кусок save на плёнку, мы его вычитывали при помощи Load «» code (преамбулу для первого пиини-К брали откуда угодно, дальше появлялась ошибка «R Tape loading error», но в память-то save загружена) меняли байтики при помощи peek «» и poke «» и так же записывали, пропуская преамбулу. Забавно было играть, когда в save было записано всё байтами FF, то есть у тебя было всё что только можно ) дальше уже не помню как и кому, но

Ленинград-1 был продан, деньги добавлены и у меня появился какой-то заводской Спектрум 48K с дисководом - это уже было счастье. Каждые выхи ездили на Тушинский аэродром (там тогда было то, что сейчас называется «Митинский радиорынок»), и там я впервые увидел демонстрацию Song in lines с крутыми эффектами и обратил внимание на





## 26 interview

чудесную музыку **J.M.Jarre «Magnetic Fields»**, после чего «заболел» **AY-ом**.

Спустя какое-то время я нашёл кто мне его сделает. На моём **Спектруме** был слот куда была выведена вся шина данных с проца, поэтому даже не надо было лезть вовнутрь, просто нужно было припаять разъем к плате с **AY**. Но Song in Lines сразу не заработал). Программа писала, кажется, «**mate pocibnu romku**» (насколько помню по памяти), что-то типа «**выкинь свою ПЗУ-шку**», но другие демки шли. Кстати сказать, ПЗУ-шку-то потом мне перепаяли в нём, сонг-ин-лайнс таки заработал нормально. **Паша (kent)** в те времена учился в музыкалке, учился хорошо, в какой-то момент ему предки подарили «**клон**» (не помню как называется) с 512 мегами памяти и второй системой **cp/m** (которую, впрочем, никто не юзал)



и тоже с **AY**. (отступление от AAA – это точно был Профи 512, что подтверждается DEMO3).

В тот момент он сокрушался, что не знает, как в редакторе **Sound Tracker** переключать октавы, из-за этого он не мог в трекере набросать музыку. Я как-то случайно нашёл заветную комбинацию клавиш, тут же позвонил ему, и уже



через несколько минут у него был готов первый музон для **AY**.

**Паша** стал писать музыку в **Sound Tracker**, но толку от этих мелодий? Я предложил ему сгонять на **Митино** и обменять диск с его мелодиями

на диск с какими-нибудь играми или демками. Первая палатка нам попала с продавцом по имени **Лузьянов Павел**, и на вопрос «нужны ли вам треки для демо или интро - он у нас с руками оторвал этот диск и поменялся на диск с каким-то софтом, но он спросил - «**чье имя вставлять в бегущую строку?**» Мы особо долго не думали стоя перед палаткой ведь вокруг были одни «**-soft**» аля «**magic soft**» «**ksa soft**» и т.п. и **Паша**, кажется, выпалил, - «а давайте **MusicSoft**». После этого стали носить диски **Лузьянову**, попутно познакомились там же с **Кано, Витьком из MagicSoft**, ещё с кем-то, уже не помню. Стали зачитывать до дыр книгу «**ZX Spectrum для пользователей и программистов**», учить ассемблер. Радости не было предела после того, как заработала наша первая программа:

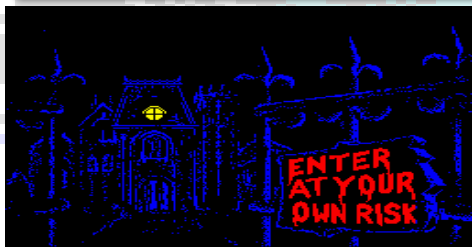
```
LD HL xxxxx
```

```
LD DE 16384
```

```
LD BC 6912
```

```
LDIR
```

и картинка из произвольного адреса **xxxxxx** мгновенно появилась на экране :) Я писал ещё в машинных кодах при помощи **poke** и запуска клманды: **randomise usr xxxxx**. Осваивали ассемблер, дизассемблер и первая «**демка**», которую разломали буквально на процедуры, выдернув оттуда всё что только возможно (бегущие строки, столбики-эквалайзеры) была



**Wild Cat Part1, кажется.**

Скролл начинался со слов «*sean adams under protection of kassoft presetns a music demo wild cat*» и дальше мы начали практиковаться в асме. Научились «**ксорить**» куски кода, что бы «скрыть» в явном виде текст скролла и т.п. Ну и уже чем дальше - тем глубже изучали и писали на асме разные процедуры. Кано, кажется, меня познакомил с **Дмитрием Комиссаровым (ХРЕН)**, тот широко разрекламировал **АМИГ-у** и потом, продав **Spectrum 48к** я пересел на **Amiga 600**. К ней же был тогда куплен первый модем на 14400 Тайваньский **Speedcom** на чипе **Rockwell** и начались мои ночные сидения на **BBS** и немногочисленных чатах типа **white bear bbs** (впоследствии **DEOL**), **mmtel** (московский междугородний телефон и телеграф, мало кто знал его), ещё какой-то 4-линейный. Было тепло и лампово, все друг друга знали. **DEOL** только был на 16 линий, **mmtel**, кажется, всего на 8 и ещё один чат вообще на 4 линии. Потом мне **Денис Сотченко (БК-шник)** поставил и настроил фидошный софт на 600-ку и я с **Амиги** вышел в **ФИДО** в общем, по музыке нас уже знало все ZX-сообщество, ездили к **Kano** регулярно, как-то у него был **RST7** и **Max Iwamoto**, так я познакомился с ними лично. Вернее с **Max Iwamoto**, если мне не изменяет память, лично познакомился, когда посещал **Enlight** в 1996 году. И потом уже их встретил на квартире у **Кано**. На **Амигу** было сложно доставать информацию и дорого, тем не менее, я нарыл и доку по ассемблеру **Motorola 680x0** и **Amiga rom kernel manual**, **RST** кое-что объяснил: как работать с графикой. У меня были некоторые его исходники под амигу, изучал, даже написал первое интро с Пашиной музыкой, но все благополучно умерло с хардом. А дохлый



хард, кстати, до сих пор валяется, но, кажется, я его вскрывал.

В 1996 году мы с **JAM (Макс Ч, наш общий знакомый, учился в параллельном классе)** в



чате **mmtel** разговорились, мол, «а в Питере сейчас **Enlight**», он сказал, мол, последняя электричка на **Тверь** уходит в 10 вечера :) мы встретились тут же и уехали в **Тверь** последней электричкой. Ночь в **Твери**, вино на «паровозе» на вокзале и первой электричкой на **Балагоз**.

Итого 5 электричек (**Москва-Тверь-Балагоз-Акуловка-Вишеры-Питер**) и мы уже шли по **Питеру**. В итоге нас там «вписали» благополучно на квартиру к ещё одному спектрумисту, не помню кто и как звали, но он нам показывал что пишет



**JAM с косплеершей**



Kent.MusicSoft aka Паша

игру **Mortal Kombat** на **ZX Spectrum** и показывал все свои наработки.

Жил он на **Шлиссельбургском проспекте**. Потом так же сгоняли ещё и в 1997 году на **Enlight-2**, который «провалился». Кто там был - помнят нас с мобильником **Моторола**. Вернулся я с **Энлайта-1997** уже в вагоне СВ, проникнув в поезд «зайцем» в купе к знакомым и занявшись там на багажной полке (она перпендикулярно сверху и меня вошедший проводник не видит).

ААА, - Рома расскажи о **JAM**, ведь именно он считал векторные эффекты, которые Вы делали в демонстрациях **Virgin**, **3DRelax**, **MTV**.

**JAM** — **Макс Ч.**, у него не было **Спектрума**, тем не менее он активно занимался программированием, с математическим складом ума человек, давал нам идеи, даже считал какие-то таблички координат на турбо-паскале на школьном компе.

**Паша (Kent)**, в основном оформлял всё демо и писал основной код. Меня предки тогда из-за учебы сильно ограничивали в компе, думая что я играю целыми днями. Из-за чего я зачем-то закончил автодорожный, вместо того, что бы пойти куда-нибудь в **МИРЭА**. В итоге я так в IT и остался по жизни, не спроектировав и не построив ни одной дороги, кроме дипломного проекта на 5 ватманах и 120 листах пояснительной записки от руки с расчетами.

ААА, - Как с **KSA Soft** познакомился? С кем ещё из спектрумистов пересекался в то время?

С **KSA** я не помню уже как познакомился. Первый наверное, **Паша (Kent)**, на почве написания музыки. Я от музыки был далёк,

а **Паша** играл на всём, на чём играется, включая трамвайный билетик, обернутый вокруг расчёски (демонстрировал игру лично!). Впоследствии **Паша** долгое время играл в группе на **Арбате** и я в начале 2000х чуть ли не каждый день приезжал тусить на **Арбат** и слушать его. Перезнакомился там тоже с арбатскими тусовщиками.

ААА, - Получается у newart на сайте ошибка и это не твоя музыка?

<http://zxtunes.com/author.php?id=306>

Да, музыку писал чисто **Кент**, он 9 лет в музыкалке отучился, играет на всём на чём играется, писал потом **s3m**, **xm** на пискюке, играл в группе, играл на **Арбате**. Сейчас не знаю, давно не общались, но судя по фоткам, играет до сих пор что-то где-то.

ААА, - Понял, поправим сайт newarta. Я хочу сказать на основе комментариев к демонстрациям записанным мной в ютуб - АУ музыка Кента пользуется огромной популярностью. Я бы сказал, что после **Frantishek Fuka** он второй, за кого чаще всего спрашивают, а чья это была музыка? Может ты не знаешь, но в начале 2000, даже скорее 2004-5 годов на форуме **zx.pk.ru** было голосование по лучшему АУ музыканту на спектре всех времён и народов. Ты не поверишь, но народ выбрал **KSA Soft**. Хотя я так не считаю. Но точно знаю, что **KSA** учился у **Kenta** и половина музакеров тамошних лет брали инструменты и огибающие у Кента. Я считаю, что в 1994-5 года Кент был одним из лучших наших музакеров, кто писал музыку в **Sound Tracker**. А учитывая то что у Вас была **Амига600**, его кавер треки **Compilic** и многие другие Амижные, даже круче чем на Амиге играли.

<http://zxtunes.com/author.php?id=476>

Ещё переводил интервью **Франтишека Фука**, так вот он так же как и Кент долгое время постигал музыкальные азы в музыкальной школе, и только потом начал писать музыку на спеке. Так что у них с Кентом почти одинаковая история. Для меня они на одном уровне, но Кент как то резко прекратил писать музыку на спеке и я считаю мы все многое потеряли при этом. У него был огромный талант. Рассказывай дальше, очень интересно у тебя получается...

Из спектрумистов первый знакомый — **Лузьянов**



**Павел.** Дальше были **Витёк Москалев (Magisoft)** и **Andrew Isaew aka Исайчик, впоследствии Кано.** Не помню уже в каком он – **Soft-om** был.

Дальше, уже как снежный ком, но, несомненно, **Кано** и **Хреш** сыграли большую роль. С обоими общаюсь до сих пор. Глядя на **Хреша** я тогда купил первый модем у него. Это был **UsRobotics Sportster 14400** по космическим для того времени деньгам 270\$.

У **Кано** в середине 2000х бывал каждый выходные, он же киноман и меломан знатный, каждые выхи – киносеанс с новым релизом какого-нить кино, при этом не одна бутылка горячительного была испита. **Хреш** одно время пропал и мы начали общаться снова сравнительно недавно, у нас своя группа в фейсбуке. **Кано**, кстати, в той группе тоже есть. **Кано** так и остается виндюзятником и железячником, всегда в курсе всех новинок железа и винды. софт, музыка и фильмы – по его части.

Мы с **Кспехом** пересели на маки, у меня два макбука (pro/air) у **Кспеха** macpro tower. Винда и убунту у меня на десктопе, винду запускаю исключительно редко под специфический софт (Corel Draw или какие-нибудь программаторы или тестилки дисков, где софт заточен исключительно под виндовс, типа Wd Marvel, которым успешно сбрасывается смарт и дефекты на хардах WD). Помню ещё что все говорили что бы мы не носили **Лузьянову Павлу** дискеты с музыкой ) в итоге стали их носить **Кано**, наверное, или **Витьку из Magic Soft**. Почему-то **Лузьянова** недолюбливал народ. Ещё помню **Тришина Сашу**. Торговал в **Тушино** софтом на **Спектрум, Амигу**, но он был мажором и быстро переключился на игры для РС. Мы к нему ездили домой на станцию метро **Алексеевская**, копировали софт для **Спектрума и Амиги**.

**ААА**, - что не говори, **Исая** много хаяли, но через **Титуса** в 2000 годах он бесплатно передал мне 9 кг дискет на перевод. А это огромнейшая коллекция. И **SoftStar** просто подарил мне компьютер **Профи**, блок питания, колонки и всю свою коллекцию дискет на перевод. И это не смотря на то, что вроде бы они были в своё время «барыгами». Так что лично у меня к ним только слова благодарности. Естественно все дискеты я перевёл и выложил в сеть для потомков.

Рома расскажи ещё о **Гицевич Ирине**, а то народ в загадке с её сканированной фото живет 35 лет. Для многих спектрумистов это была первая любовь. Хотелось бы узнать правду, ведь эта



ГИЦЕВИЧ ИРИНА

часть в демо **Virgin** была сделана в ответ на мою демку с **Юлией Бабкиной**, которую попросил вставить **GAM (Громилов Александр)**, написав что она IS THE BEST OF PEOPLE После чего началась новая эра, когда в скролле передавали приветы своим девушкам и боролись за звание лучшей.

С **Гицевич Ириной** я начал встречаться



Ну тут наверное мне 15, Ирине 13.  
а может и 12/14 я не помню точно



# 30 interview

когда мне было 14 а ей 12 лет лол :) она одноклассница Kent-а. у нас были одинаковые собаки и она мне очень понравилась. Сначала, как водится, взаимное отвращение, а потом сдружились, любовь-морковь всё такое. Она мне фотографии подарила и подписала:

**Макс (JAM)** тогда сканировал мне фотку ещё в gif, сканеров-то не было, стояли космос и к ним ещё целый IBM PC был нужен! Потом кстати **JAM** придумал как перевести gif в 1-битное изображение, ну и эксперимент был сразу сделан на фотке **Ирки** (ее Ирина зовут) идея вставить в демо, конечно, моя :) общаемся с ней до сих пор, у неё уж сыну лет 18 или 19, она живет со вторым мужем в **Китае**. Сюда приезжает раз в сколько-то лет продлить визу. Виделись в прошлом году. Выглядит норм для своих 40 лет, у них там фитнес-клуб (муж бодибилдер), здоровое питание, консультации на эту тематику и все такое.

AAA, - С **Сергеем aka Sey Soft** ты общался, помнишь хоть что то?

**SEY-Soft** само название помню, человека оттуда нет. Общались, видимо, оба, потому что само название «сей-софт» точно в памяти. Хотя **Кент** с ним делал демки, в которые **Sey Soft** рисовал графику. Например в **Demo9, CD Demo, Karp Demo**.

AAA, - Вот как раз таки **Sey Soft**, с которым я в те годы очень сильно дружил и моделился день и ночь, постоянно рассказывал мне о «семейке» **MusicSoft**. Я помню много про Израиль и какое то радио через которое один из **MusicSoft-ов** общается с СНГ-шными спектрумистами. Освежи в памяти это время и мы закончим этот огромный рассказ.

В Израиль я уезжал в декабре 1994



года, прямо перед новым годом аля 30 числа и на следующий день был «штурм Грозного». Мне тогда аккурат было 17 лет и я в 18 вполне себе попадал под призыв на первую чеченскую (кстати, у меня некоторые друзья ездили туда по срочной службе. Один сопровождал составы, в боях не участвовал, а вот сокурсника призвали снайпером, комиссовался по ранению, потом доучивались с ним уже в другом потоке после того как я вернулся из **Израиля**). Свалить из страны у меня хотел отец, в 90-е связался с «бандитами», крутил что-то, ну и в какой-то момент (по рассказам матери) его «приговорили». Например, прислали открытку «поздравляем тебя с 8 марта». Он, видимо, очень боялся и искал куда свалить, помню, жужжал про **ЮАР** и **Канаду**, в итоге зачем-то выбрал **Израиль** (хотя, не еврей и у нас в родне никого). Ему кто-то «пообещал» там помочь с гражданством (видимо за деньги), у него было свое издательство в **Москве**, он туда отправил контейнер книг, что бы там на что-то жить первое время, купил хату не плохую, квадратов под 120 с двумя балконами в г.Хайфа. (**Макс (jam)** когда ездил в **Израиль** и



был в **Хайфе**, я его попросил прогуляться там по адресу и сфоткать дом, где я жил:

Мой отец, который всю жизнь не пил, в **Израиле** нажрался, я его там первый раз пьяным увидел в свои 17 лет. В **Израиле** собиралось все что можно увезти, я только **Амигу-1200** продал **Кано** (забыл написать 600-ку в 1994 продал и купил 1200-ку, но радость была недолгой из-за поездки в **Израиль**, 1200-ка продавалась **Кано**).

Привезли туда и факс «панасоник», с которого я от полной скуки и безделья слал приветы на радио-7 в программу «музыка по факсу» посылать. Народ уже знал, когда слушать,



интернета ж не было, компа у меня там не было. Языка не знаю, ничего не знаю, море-не море, на хрен оно нужно, когда его с балкона видно, денег нет, ничего нет. Радость одна — **китайский карманный «тетрис»** в который я играл целыми днями и

таким образом так прошел его весь! Ещё был телевизор с несколькими русскими каналами и нормальным человеческим MTV. От нечего делать перевёл всю инструкцию от модема **Speedcom** на русский язык, с канала MTV тогда записал через видеокамеру (экранная копия с телевизора блин) серию мультв **beavis and butthead** на английском языке. У нас в

1994 не было MTV и этих персонажей ещё не знали, но я с них ржал. Сразу на видак писать не мог — телек был без видео-выходов, только со входами. Ещё дед прикольный был сосед — **Моше Шварц**, узник ГУЛАГа, учил меня ивриту. Алфавит, слова, счёт помню до сих пор, могу даже что-то прочитать. На самом деле простой язык очень, как табуретка. только слова учи, больше ничего не надо. Затем отец улетел в **Москву** решать что-то, мы с мамой и сестрой кое-как жили в **Израиле**, получая какие-то проценты от продажи книг, отец договорился с магазинами там. Поэтому у нас весь балкон был забит книгами.

Я тоже там перечитал многое от фантастики до биографии **Чикатило** и исторических книг. В итоге мама, наверное, тоже поняла что пора валить, договорились что у нас заберут весь балкон книг по 1-2 шекеля, его и забрали ) Поехали в **Тель-Авив** и купили билеты аэрофлот в **Москву**. Вот тогда я почувствовал вкус моря и впервые за полгода пошёл на пляж в **Хайфе**, типа загореть. Ага, сгорел натурально, страшно и дико: спереди и сзади , все мокло и горело. В **Москве** бы, наверное, меня бы укули в ожоговое отделение. В **Израиле** в аптеках тоже ужаснулись, но дали какую-то чудесную мазь, от которой все ожоги стали проходить очень быстро. В итоге вернулись в **Москву**, у отца была ещё попытка свалить в **Израиль** уже одному, опять не получилось, опять вернулся, скрывался, потом пропал. В 2000 нашли под **Ярославлем** в карьере с двумя пулями в спину.

ААА, - огромное спасибо MusicSoft за эти интересные воспоминания и фотографии. Надеюсь материал будет интересен читателям журнала «За Рулём» ■





\*по версии ZaRulem Печатное слово

**М**ы решили продолжить и весной 2021 года рубрику **«Спектрумист года»**, в которой активные, талантливые и креативные спектрумисты проявили себя в ушедшем году. Как и всегда выбор был нелёгким, да ещё и в целом 2020 год был непростым практически для всех. Конечно, распространение коронавируса не способствовало

росту количества, равно как и качества прода.

Как и в прошлом году, определение номинантов и победителей осуществлялось путём закрытого голосования. Можете согласиться с нашим выбором или аргументированно возразить, что есть и более достойные. Ваши комментарии и предложения помогут нам всецело и объективно подойти к выбору победителей в следующий раз.

Пару слов о призах! Конечно, мы не просто печатаем ник-неймы с громкими эпитетами, а ещё и подкрепляем свои слова сувенирной продукцией. Как вы помните, в прошлом году, победителям направлялись самые настоящие медали! Этот год не станет исключением! Отдельно хотим поблагодарить [@OldFartGamer](#) за участие в подготовке наградного фонда. Напомню, что он сделал репродукцию оригинального мануала к игре Elite. Мануал выполнен в виде полноценной брошюры и выглядит необычно и достаточно солидно.

Какие ещё ништяки получают победители - вы узнаете чуть позже из нашей группы [ВКонтакте](#). Мы как всегда постарались подойти креативно к данной задаче, чтобы не оставить вас равнодушными! Вступайте в группу, подписывайтесь на новости и всегда будете в курсе!

А теперь непосредственно к самим победителям!  
Встречаем!!!



### Железячник года

Представляем вашему вниманию

**Евгения Лозового**, известного как **UziX**.

Евгений в 2020 году представил общественности очень крутую разработку, плату **Sizif-512** - ещё одного клона **ZX Spectrum** на базе CPLD. Турба, тайминги **Pentagon** и оригинальных 48k/128k, саундрайв, расширенная память и т.п., причём почти это всё переключаются на ходу, без перезапуска машины! Специально здесь публикуем ссылку на его проект в [github](#), чтобы к разработке мог присоединиться любой желающий! Настоящий железячник года, поздравляем, **Евгений!!!**

### Игродел года

Представляем вашему вниманию - **Dr.Bars!** Вы прекрасно понимаете за какие заслуги мы считаем его игроделом года! Конечно же, за разработку новой официальной части про самое известное яйцо в мире - **Dizzy VIII - Wonderful Dizzy**.

Впервые за 26 лет его приключения продолжаются!

Игра создана с разрешения и по согласованию с **братьями Оливер**, которые

ещё в далёкие восьмидесятые создали оригинальную серию игр про **Диззи!** Релиз восьмой части был встречен

бурными аплодисментами на ресурсах, посвящённых **Спектруму**, во всех странах мира! Игра выполнена на качественно новом уровне. **Dr.Bars** в очередной раз подтвердил свои навыки высококвалифицированного кодера, который способен на покорение вершин **гейм-мейкинга!**

Поздравляем его и торжественно вручаем призы пока виртуально, но надеемся передать подарки непосредственно в талантливые руки, ведь мы с ним живём в одном городе! :)





# 34 hall of fame



## Художник года

Непростой выбор встал перед нами и мы решили вообще не ограничиваться отечественными художниками, а взглянули шире на ярких представителей этого вида компьютерного творчества. Всегда с удивлением и восхищением смотрим на работы художников на **Спектри**, на то как они добиваются впечатляющего результата несмотря на ограничения платформы в количестве цветов и экранном разрешении. Так было и в этот раз, когда мы просматривали галереи художников одного за одним.

По итогу бурных обсуждений, сомнений и весомых аргументов за того или иного артиста, был выбран никто иной, как **Marco Antonio Del Campo** известный также как **MAC!!!** Талантливый человек из Испании, который рисует потрясающей красоты заставки (loading-screen) для современных игр на **ZX Spectrum**. Вы только посмотрите на эти цвета, кажется он выжимает невероятные комбинации и вообще не знаком с клэшингом - настолько гармонично и красочно смотрятся его работы!

Рекомендуем посмотреть галерею художника на [zxart.ee](http://zxart.ee). Кстати, заставку для вышедшего в 2020 году **Dizzy** (смотрите на предыдущей странице) сотворил также **MAC** и кроме того слайды внутри игры также его авторства! Можно упомянуть также, что заставку к **Crystal Kingdom Dizzy** в 2017 году также создал **Марко Антонио!**

Итак, звание «Художник года» в этот раз отправляется в **Испанию!!!** Представляем как он удивится, когда узнает какого титула был удостоен! И, честно говоря, у нас такое предчувствие, что доставить ему памятный сувенир будет большим приключением! Но тем интереснее добиваться результата.

О доставке приза мы отчитаемся!



### Кодер года

Кодером быть интересно и конечно же почетно! Самое главное не растерять свой интерес со временем. И наш кодер на протяжении многих лет неутомимо показывает отличные результаты. Представляем вашему вниманию **Владислава Ювенко** из **Красноярска**! Вам он известен под ником **tiboh**. Год за годом он выпускает дисковые релизы игр как самостоятельно, так и совместно с другими спектрумистами. В дисковые версии включаются чит-коды, что очень удобно, когда хочешь порубиться вечером в какую-нибудь игру! За 2020 год он зарелизил **не менее 50 игр**, а эта работа им начата гораздо раньше и продолжается по сей день! Каждый раз интересно читать раздел апдейтов на сайте [Virtual TR-DOS](#) и благодаря его труду новье появляется с завидной регулярностью. **Так держать!!!**

### Прорыв года

В этой номинации окончательно и безоговорочно поздравляем **Романа Сандаевского** - владельца и основателя популярного сообщества для поклонников **ZX Spectrum** во **ВКонтакте** – **«ZX Speccy»**. На данный момент его группа насчитывает почти шесть тысяч участников и уже практически наступает на пятки старейшему сообществу **БК** (которое насчитывает порядка 6.6k участников)!!! Регулярно публикуются новости, посвященные платформе, выходят новые видосы и интересные материалы, привлекаются новые авторы.

Интересный материал о его проекте современного **Спектрум**-совместимого компьютера **«ZX Easy»** вы можете прочитать в прошлом выпуске нашего журнала. Кроме этого у него появилось несколько проектов, о которых мы написали на странице 6 настоящего номера журнала.

Нам понравилось с какой деловой хваткой **Роман** взялся за продвижение платформы и своего сообщества и всего, что связано с платформой **ZX Spectrum**!!!



**Поздравляем и желаем дальнейших успехов!**





Контакты редакции:



[not\\_soft@mail.ru](mailto:not_soft@mail.ru)



[blastoff\\_zx](https://www.instagram.com/blastoff_zx)



[vk.com/prospeccy](https://vk.com/prospeccy)



[@djoni\\_don](https://twitter.com/djoni_don)

© NOT Soft Новосибирск апрель 2021 г.