

Архитектура и программирование компьютера **Vectrex**



Пётр Соболев (frog)



Vectrex
GRAPHIC COMPUTER SYSTEM

THE HIGH PERFORMANCE MACHINE



EXCLUSIVE BUILT IN SCREEN
Portable—No TV Set Needed!



VECTREX 3-D IMAGER™
Real 3-D...In Color!



VECTREX LIGHT PEN
A Unique Learning Tool!

Now there's a graphic computer system that can truly be called "High Performance." Vectrex has the look, action and power that leaves the others in the dust!

Only Vectrex has its own built-in screen so it's portable—pick it up and play almost anywhere! For unbelievable realism, put on the new Vectrex 3-D Imager™ accessory. Get ready for a **real** 3-D experience—in **color**—that will surround you and draw you into the action like no other video game, home or arcade.

The new Vectrex Light Pen accessory makes learning fun! You'll interact directly with the built-in screen to create brilliant graphic effects, compose music, play educational games, even create animation!

Vectrex has a large and expanding library of "High Performance" games, including popular arcade classics like Pole Position™, Scramble™ and Berzerk®. All Vectrex cartridges are ready to challenge you and transport you to the stars...and beyond!

Vectrex™
GRAPHIC COMPUTER SYSTEM

™ Trademark of General Consumer Electronics, Inc.

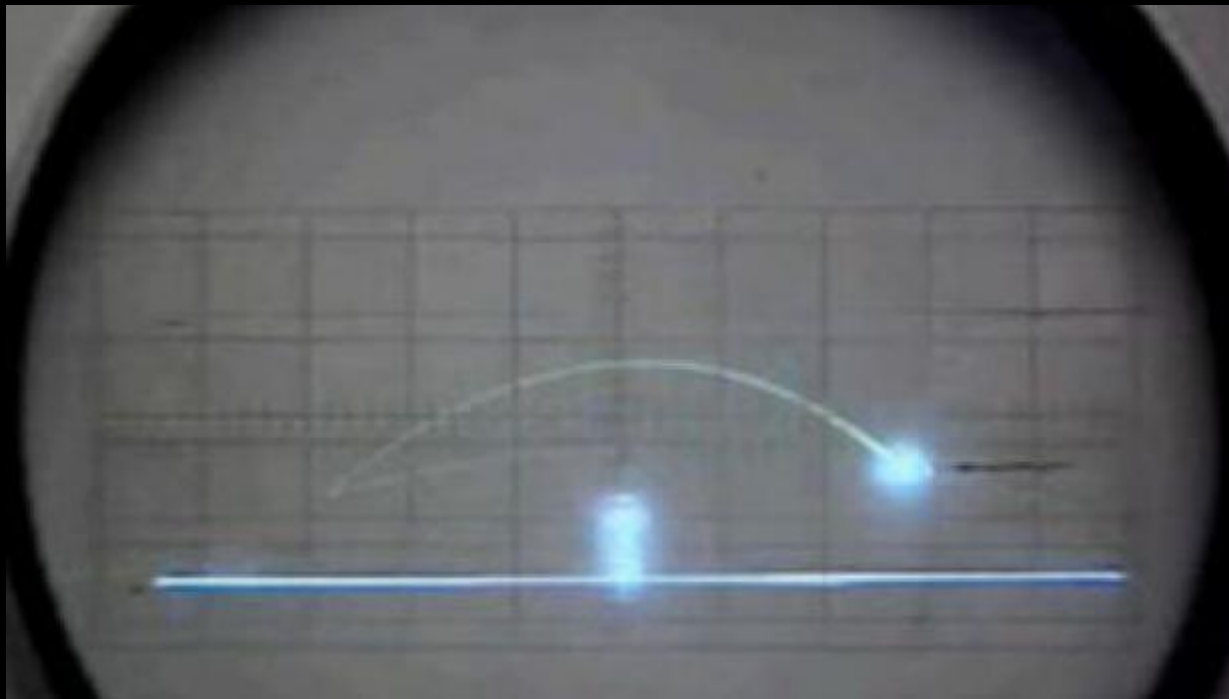
"Pole Position" is created and licensed by Namco. © 1982

™ Trademark of and licensed by Konami Industry. © 1981

Trademark of and licensed by Stern Electronics Inc. © 1980

© 1983 General Consumer Electronics, Inc., Santa Monica, CA 90401. All Rights Reserved.

Предки Vestrex:



“Tennis for two” на аналоговом компьютере
Donner Model 30 (**1958 год**)

Предки Vestrex:



Игровые автоматы "Cinematronics"
(~ 1977 ... 1982 год)

Vectrex, 1982 год:

CPU: Motorola 6809 (1.5MHz, 8 бит)

RAM: 1 K

ROM: 8 K (BIOS)

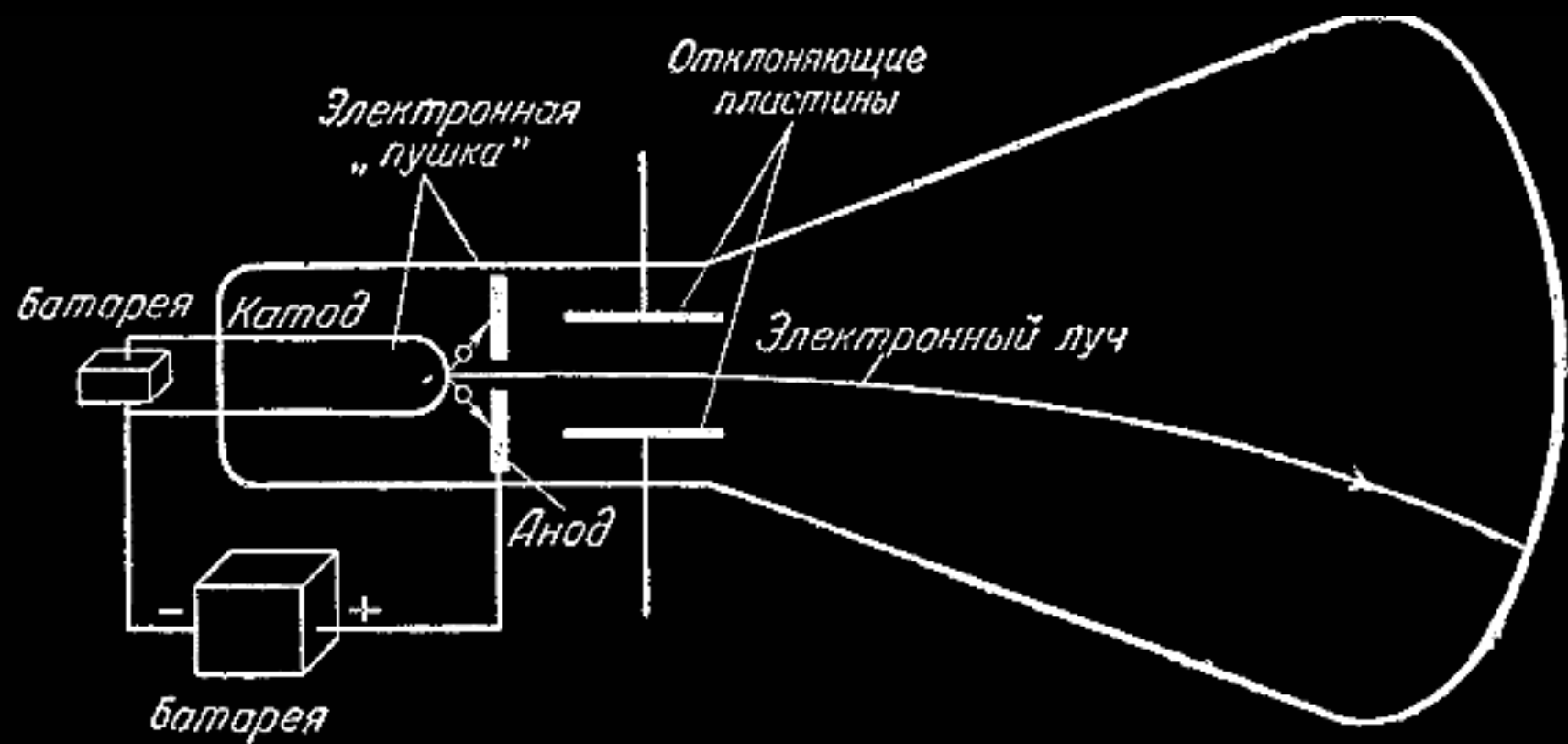
+ 32 K (макс. стандартный картридж)

Видео: Векторный дисплей (ЦАП, интеграторы)

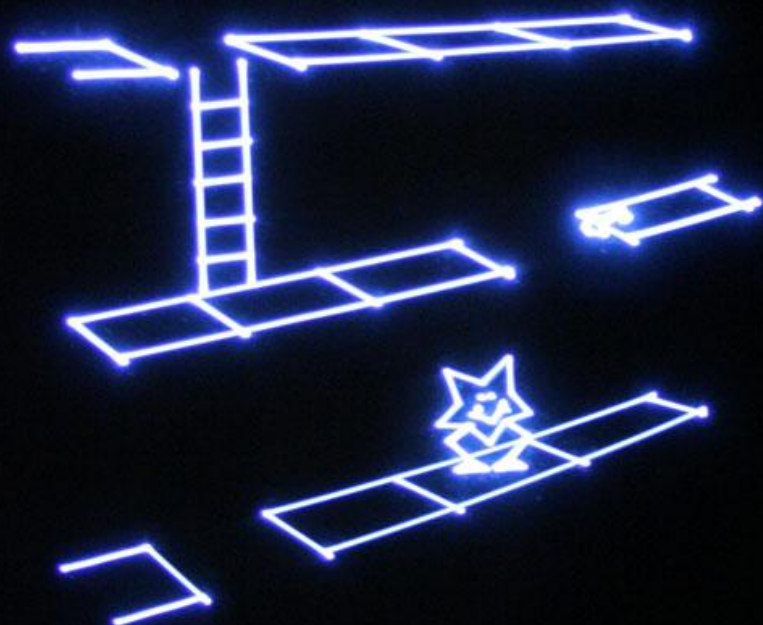
Звук: AY8912 (как в MSX и поздних Спектрумах)

+ цифровой звук (8 бит с ЦАП)

Ввод: джойстики, световое перо, очки 3D Imager



0



3-美

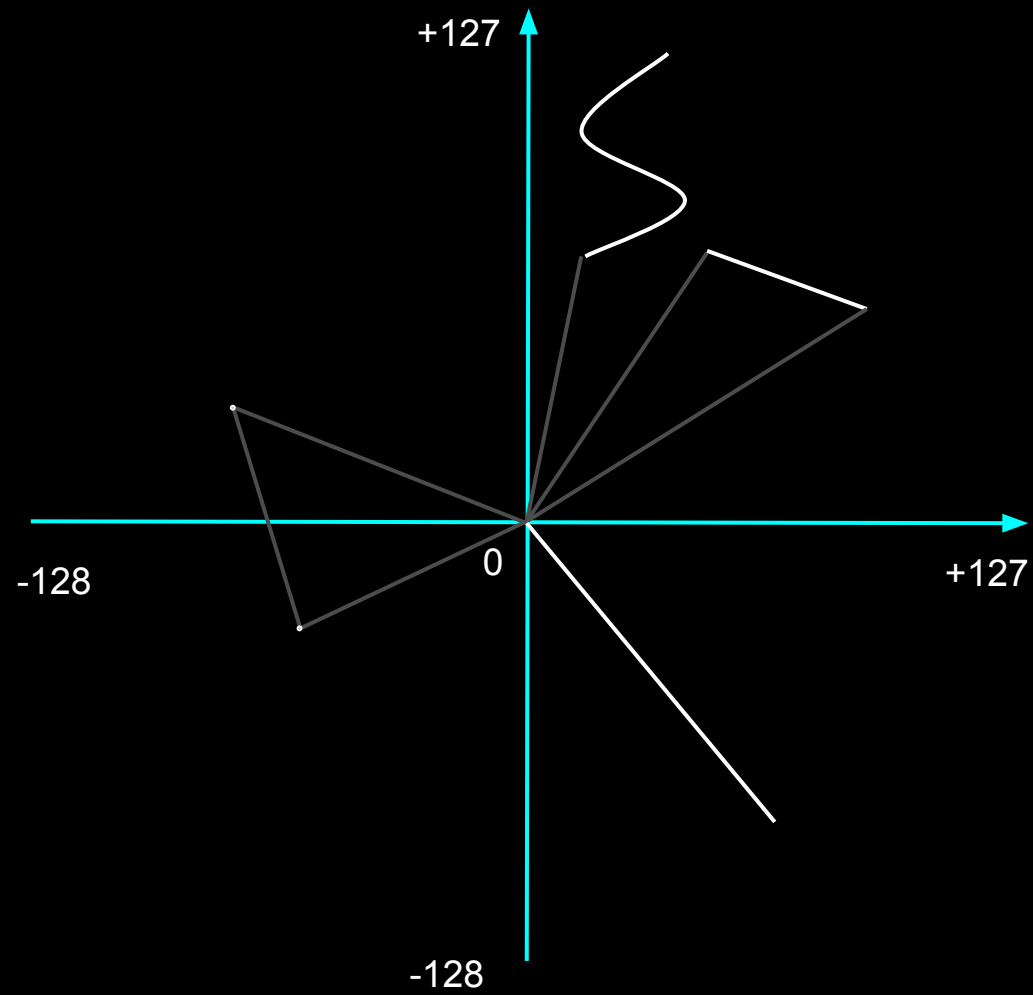
1

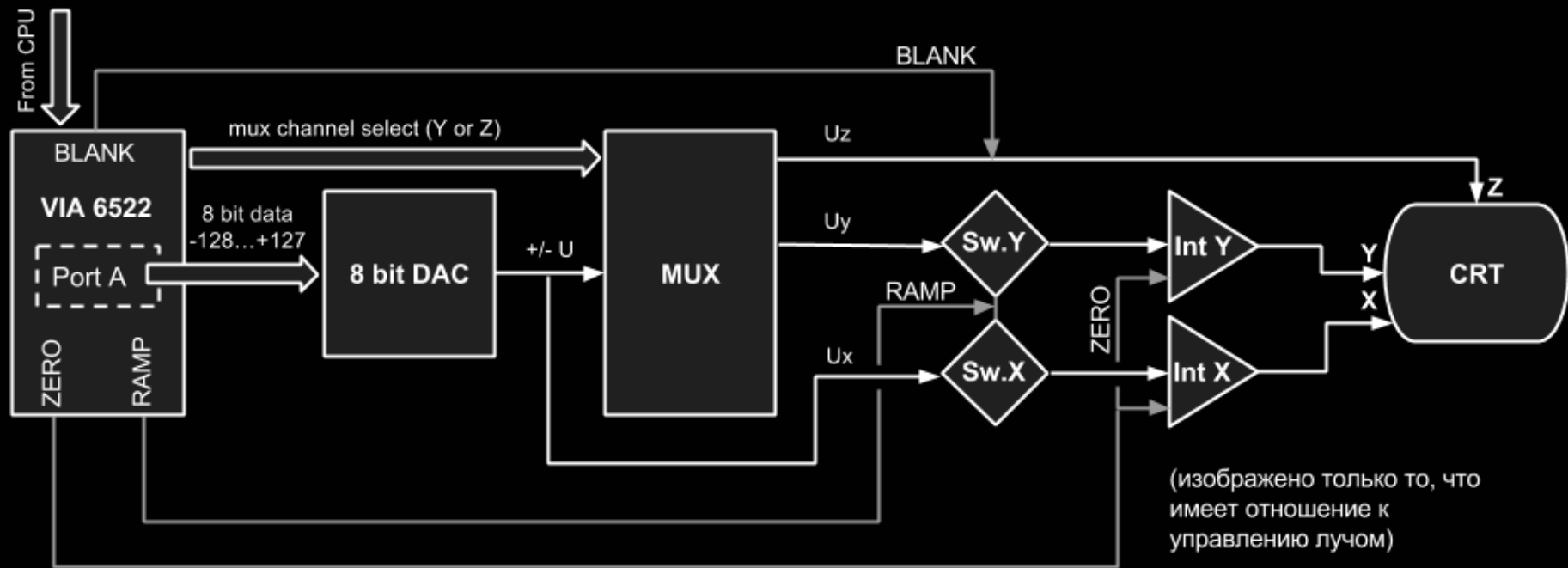
0

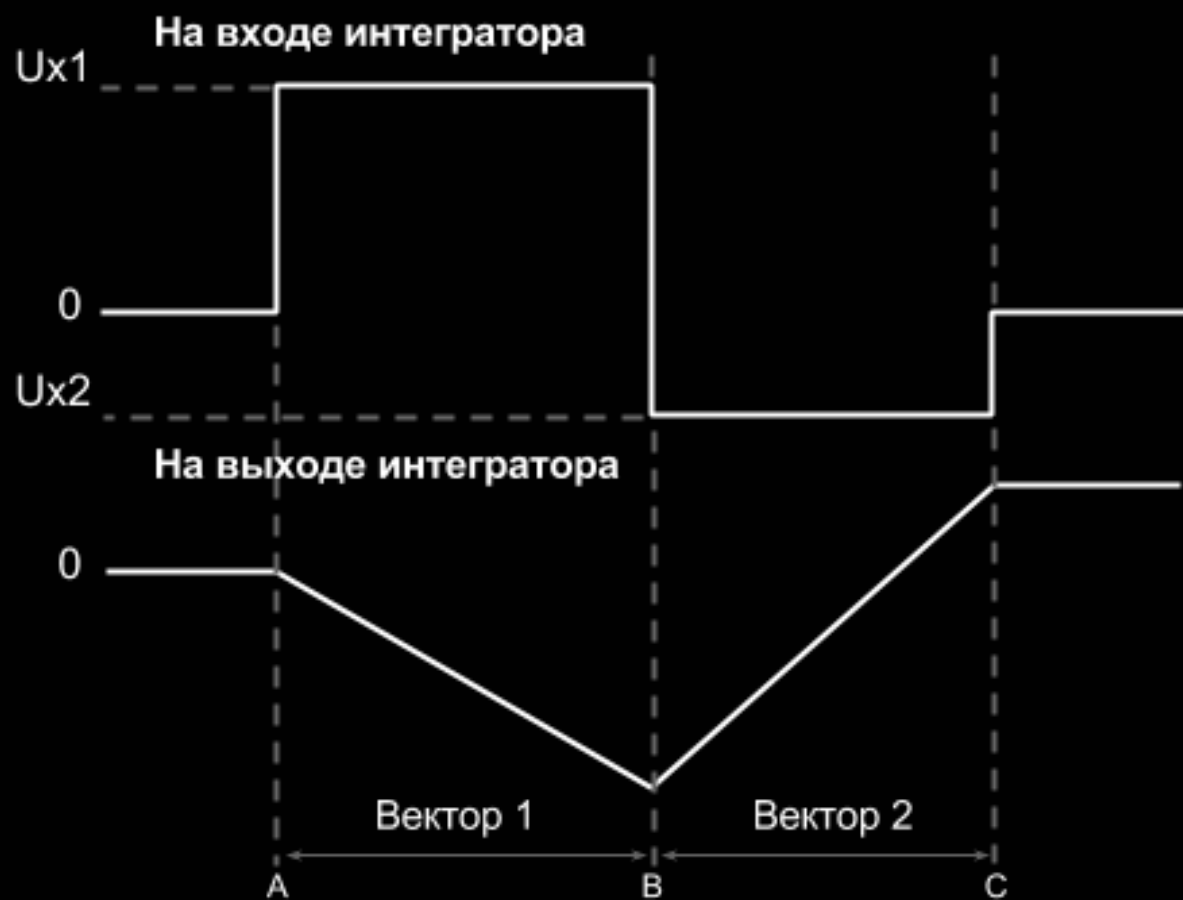
120

000.0
000









Рисование линии (по таймеру)

1. Записываем в ЦАП значение яркости и переключаем мультиплексор на канал Z
2. Записываем в ЦАП значение Y и переключаем мультиплексор на канал Y
3. *Выключаем* мультиплексор и записываем в ЦАП значение X
4. Запускаем таймер (начинается интегрирование)
5. ... луч движется, оставляя след ...
6. Когда время в таймере истечёт, выключаем луч

Рисование линии (вручную)

1. Записываем в ЦАП значение яркости и переключаем мультиплексор на канал Z
2. Записываем в ЦАП значение Y и переключаем мультиплексор на канал Y
3. *Выключаем* мультиплексор и записываем в ЦАП значение X
4. Запускаем интегрирование
5. ... луч движется, оставляя след ...
6. Обеспечиваем нужную задержку.
7. Останавливаем интегрирование и выключаем луч

RAINY

Рисование кривой

1. Записываем в ЦАП значение яркость и переключаем мультиплексор на канал Z
2. Записываем в ЦАП значение Y и переключаем мультиплексор на канал Y
3. *Выключаем* мультиплексор и записываем в ЦАП значение X
4. Запускаем интегрирование
5. ... луч движется, оставляя след ...
6. Пишем в ЦАП разные значения X , пока не надоест
7. Останавливаем интегрирование и выключаем луч

Кривые



200

100

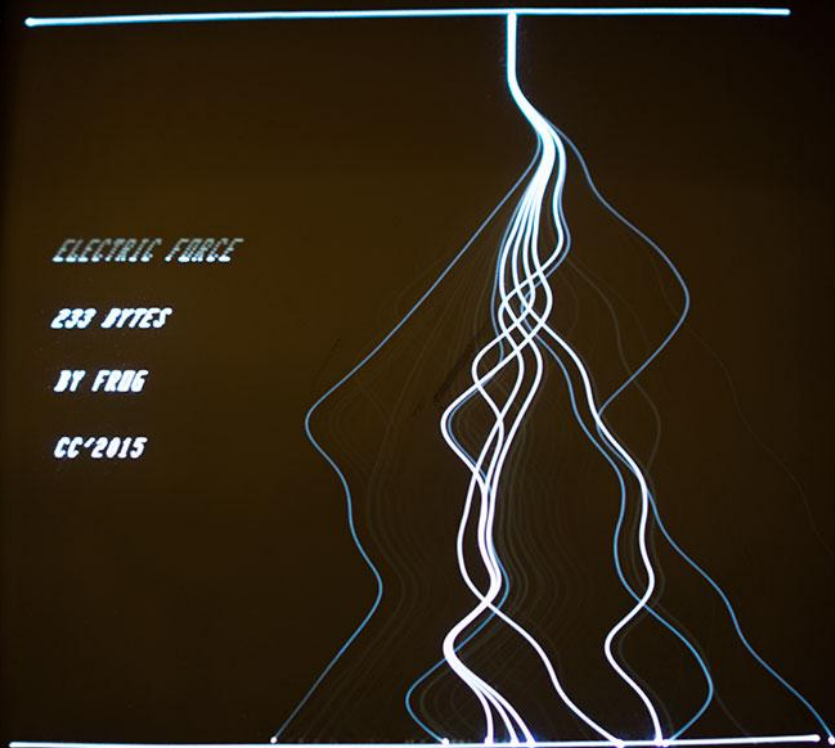
0

ELECTRIC FORCE

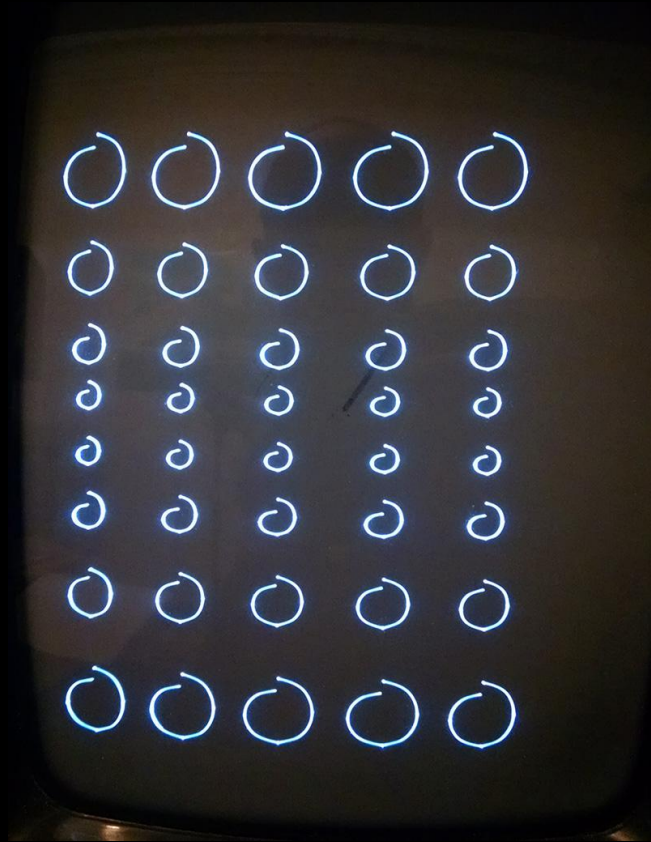
253 BYTES

BY FRBG

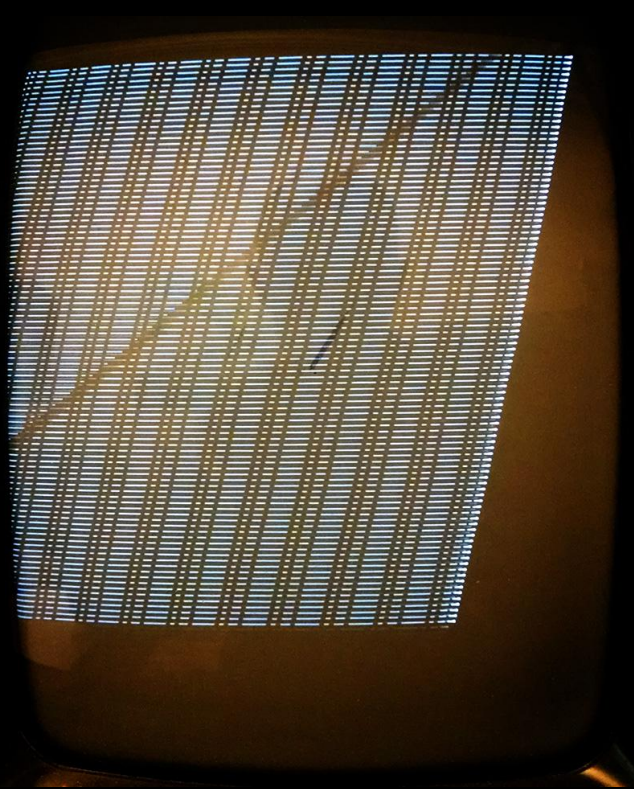
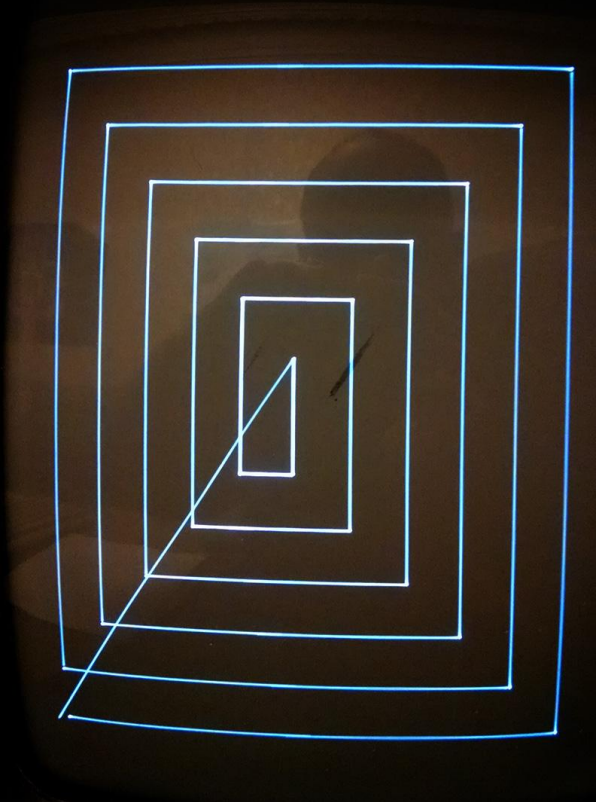
CC'2015



Различия характеристик устройств



Уплыывание параметров со временем



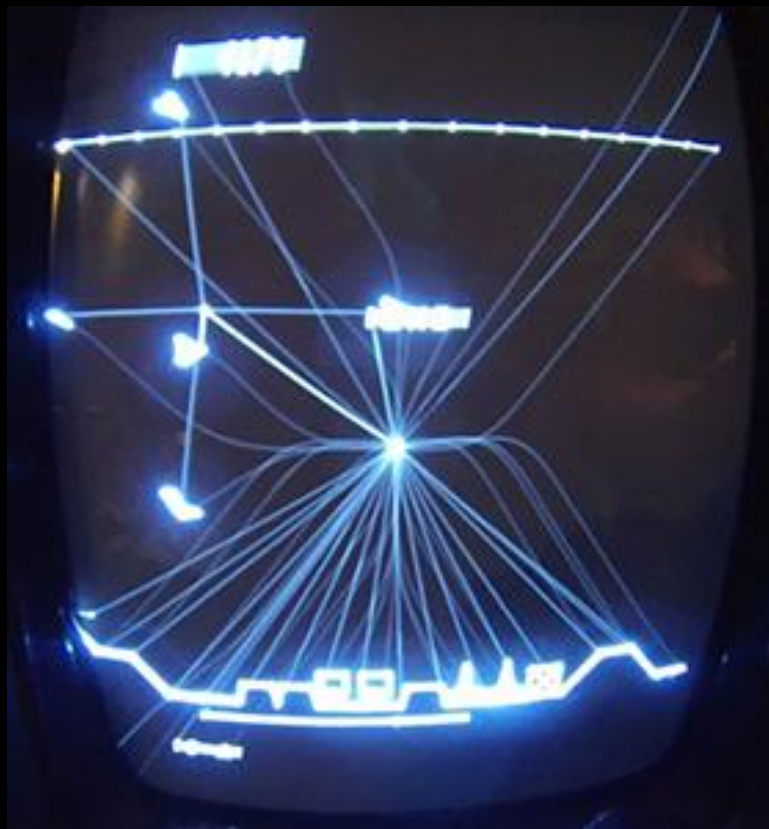
Рекалибровка

Если рисовать линию за линией подряд, очень быстро нарастает погрешность и изображение будет дрожать и извиваться.

Необходимо как можно чаще рекалибровать луч - разряжать конденсаторы интеграторов и обнулять ЦАП.

В результате луч оказывается в центре экрана, поэтому каждая рекалибровка - это дополнительное время на перемещение луча от центра в нужное место.

Рекалибровка

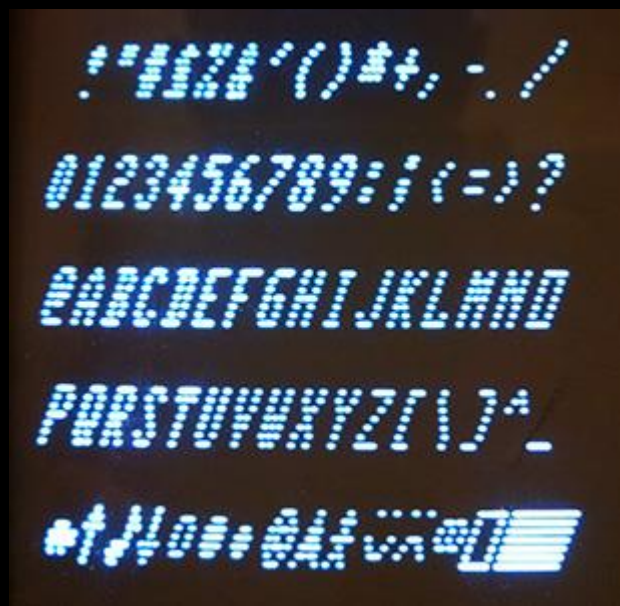


Растр из векторов

Рисуем вручную линии строчка за строчкой.

В процессе рисования включаем-выключаем луч через сдвиговый регистр в 6522 (shift register)

- 1.Задержки крайне критичны. Даже один лишний/недостающий такт влияет на результат
- 2.Много строк нарисовать за “кадр” не успеем
- 3.Яркостью в процессе управлять толком не получится
- 4.После каждой строчки нужна рекалибровка



Звук:

1. Цифровой

Переключаем мультиплексор на канал звука и пишем в ЦАП сэмпл байт за байтом (8 бит, со знаком)

2. Через AY8912

Пишем нужные значения в его регистры, получаем соответствующие звуки.

Можно пользоваться подпрограммами BIOS. Или использовать VecSound (он понимает .YM).

WE INVITE YOU TO

Средства разработки

Эмуляторы Vectrex: ParaJVE/JVD (Windows), DVE (DOS)

Ассемблер 6809: AS09 (Windows, DOS)

C: GCC для 6809. Для ленивых извращенцев :)

Дизассемблеры 6809: DIS6809 (DOS), IDA 6.7+ ?

Эмулятор ROM (спасибо svo, хоть он и писал загрузчик на GCC ;)

Средства разработки

The screenshot displays the ParaJVE development environment with the following components:

- Program Window:** Shows assembly code for `bios.asm`. The current instruction is `F1A7 JMP Recalibrate`.
- RAM Window:** Displays a memory dump starting at address `C800`. The dump shows hexadecimal values and their corresponding ASCII representations.
- Breakpoints Window:** Shows a single breakpoint set at address `0000` with the label `Exec. $0000 (line 10, test.asm)`.
- Registers Window:** Displays the current state of the registers. The `X` register contains the value `0150`.
- Console Window:** Shows the execution log, including messages about loading source files, initializing the system, and the current state of the audio line.
- Cycles Window:** Displays the current cycle count, which is `11`.

Vectrex и эмулятор - различия



Заготовка программы

```
include "vectrex.i"
```

```
org      0
```

```
db       "g GCE 2015", $80
```

```
dw       $F600
```

```
db       $F8, $32, 33, -36
```

```
db       "PROGRAM TITLE", $80
```

```
db       0
```

```
loop:
```

```
jsr      Wait_Recal
```

```
....
```

```
bra      loop
```

Фрагмент кода

```
lda    #$aa
```

```
ldb    #$40
```

```
sta    <VIA_shift_reg
```

```
clr    <VIA_t1_cnt_hi
```

wait_timer:

```
sta    <VIA_shift_reg
```

```
bitb   <VIA_int_flags
```

```
beq     wait_timer
```

```
clr    <VIA_shift_reg
```

Плюсы и минусы платформы

- + не нужно ничего стирать с экрана
- + нет пикселизации/ступенек
- + простота рисования прямых и кривых (почти нет вычислений)
- + много градаций яркости, простота fading'a
- + хороший процессор
- + хороший звук
- очень мало ОЗУ
- очень ограниченные возможности в рисовании кривых
- нельзя нарисовать много линий -> низкая плотность изображения
- стыки линий всегда заметны

В целом, платформа очень необычна и традиционные для 8-битных компьютеров подходы работают плохо - нужен индивидуальный подход чтобы реализовать потенциал, который даёт векторная графика.

ВОПРОСЫ?



Пётр Соболев (frog)

<http://enlight.ru>

frog@enlight.ru

(c) fedorbazhenov on fotki.yandex.ru