

# Sistemas de Carga de Cassette en ATARI 8-bit

Turbo con recuperación de  
errores

Franco Catrin Leiva  
<http://franco.arealinux.cl>  
RetroBits 2011

# Temario

- ◆ Sistema de carga estándar
  - Estructura de carga/grabación lógica
  - Sistema de booteo desde cassette
  - Cargador de “signo de exclamación”
  - Formato ejecutable XEX
- ◆ Sistemas de carga Turbo con Recuperación de Errores
  - Estructura de carga/grabación física
  - Cargadores Turbo
  - Recuperación de errores
  - Cargadores Injektor
- ◆ Bonus Track : Sistema de carga CAIN

# Sistemas de Carga Estándar



# Sistema de carga estándar

- ◆ Aplica a toda carga y grabación realizada por el Sistema Operativo de ATARI:
  - Boot desde Cassette
  - CLOAD / CSAVE
  - XIO
  - Cargador de “signo de exclamación”

# Estructura de carga/grabación



- ♦ Sincronización da tiempo a que cinta / motor se estabilice
- ♦ Espacio entre cada bloque de datos se llama Inter Record Gap o IRG
- ♦ IRG da tiempo al Sistema Operativo para procesar los datos leídos
- ♦ EOF = End Of File = Fin del Archivo

# Bloques de datos



- ♦ Cada bloque tiene 132 bytes
- ♦ Los dos primeros son para medir velocidad (85,85)
- ♦ T = Tipo de registro:
  - 252 = Registro Normal (128 bytes)
  - 250 = Registro Parcial (N bytes)
  - 254 = Registro EOF (128 bytes van en cero)
- ♦ A continuación vienen 128 bytes de datos
- ♦ C = Checksum

# Rutina de carga del Sistema Operativo

- 1) Encender motor y esperar unos 3 segundos
- 2) Esperar la secuencia 85, 85
- 3) Cargar 129 bytes en \$3FF (buffer)
- 4) Leer último byte y comparar con suma de los bytes cargados (checksum). Arrojar error si no coincide
  - ♦ Si T = bloque normal => Transferir 128 bytes del buffer
  - ♦ Si T = bloque parcial => Transferir N bytes del buffer
  - ♦ Si T != registro EOF, Volver al punto 2
- 5) Finalizar Carga

# Sistema de Booteo desde Cassette

- ♦ Al encender el computador, usuario presiona:
  - START = Iniciar desde Cassette
  - OPTION = Desactivar ROM de BASIC
- ♦ Tras presionar una tecla, comienza la carga desde Cassette para bootear desde ahí
- ♦ Si falla la carga muestra BOOT ERROR



# Estructura de Boot



- ◆ Primer bloque incluye metadata:
  - 1 byte ignorado
  - N = Número de bloques a cargar
  - DL/DH = Dirección de memoria destino
  - XL/XH = Dirección de memoria ejecución
- ◆ Al terminar de cargar el archivo en D, comienza la ejecución del código en X

# Cargador Estándar (!)

# Cargador Estándar (!)



- ◆ Es el cargador típico de juegos en Cassette
- ◆ Se bootea con este cargador y luego éste carga un archivo en formato ejecutable XEX
- ◆ El cargador ocupa 6 bloques
- ◆ Al ejecutarse, suena un beep, limpia la pantalla y pone el signo de exclamación (!)

# Formato ejecutable XEX

- ◆ Divide un programa en bloques de datos
- ◆ Cada bloque tiene : Inicio y Largo
- ◆ Al terminar de cargar cada bloque, se ejecuta a través de un puntero si es distinto de cero
- ◆ Ejemplo clásico:
  - Bloque 1: Pone “Loading...” + título del juego
  - Bloque 2: El juego en sí mismo



# **Sistemas de Carga Turbo (con recuperación de errores)**

# Falencias sistema estándar

- ♦ Velocidad de carga demasiado conservadora
  - Tenían que lidiar con imperfecciones de cintas y partes mecánicas
  - Mientras más largo el bloque, mayor probabilidad de error
- ♦ Intolerancia a fallas
  - Si fallaba la carga de un bloque, fallaba toda la carga

# Sistemas de carga Turbo (CRE)

- ♦ Modificaban **velocidad de carga**
- ♦ Modificaban estructura de bloques, para proveer **recuperación de errores**
- ♦ Modificaciones a partir de:
  - Rutinas de carga propias
  - Parches al Sistema Operativo
  - Hardware modificado (Injektor)

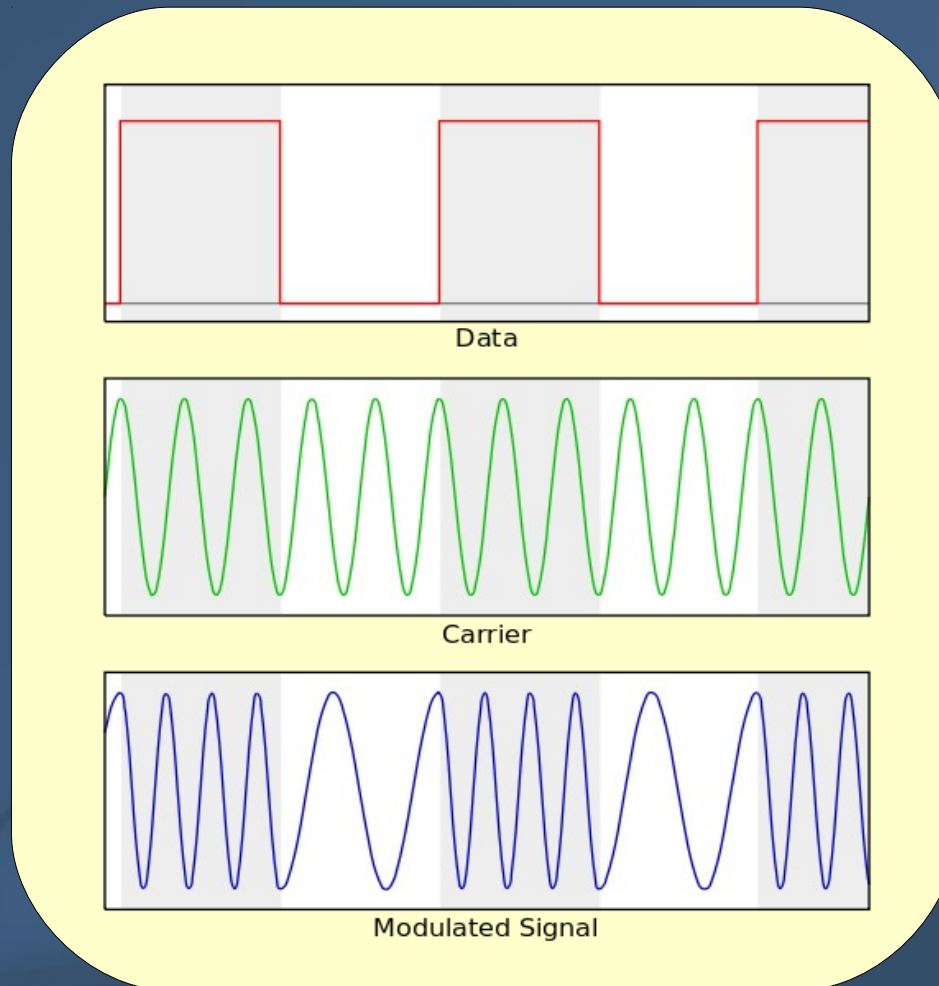
# Formato de grabación físico

- ◆ Sistema estándar opera a 600 baudios:
  - 4,5 minutos para cargar 16 KB (Donkey Kong)
  - 11 minutos para cargar 40 KB (Black Lamp/Zybex)
  - 18 minutos para cargar 64 KB (toda la memoria)
  - 13 horas para cargar MP3 de 3 minutos
- ◆ La velocidad está determinada al grabar
- ◆ Al momento de carga se calcula la velocidad para compensar diferencias físicas



# Formato de grabación físico

- ♦ ATARI usa FSK : Frequency Shift Keying



# Frequency Shift Keying

- ◆ Se graba un tono base o portadora
- ◆ Al variar este tono, se reconoce como un cero o un uno
- ◆ Al juntar varios ceros y unos se van reconstruyendo los bytes
- ◆ Se usan bits de control entre cada byte para verificar que se está leyendo en forma coordinada

0XXXXXXXXX1 0XXXXXXXXX1 0XXXXXXXXX1

# Lectura desde hardware

- ♦ Atari usaba una interfaz llamada SIO : Serial Input Output
- ♦ Los unos y ceros se podían leer desde un bit en una dirección/puerto del SIO
- ♦ El SIO se programaba para leer a una determinada velocidad
- ♦ La programación consistían en indicar una frecuencia de lectura
- ♦ SIO convertía automáticamente bits a bytes

# Esquema básico de SIO



# Ajuste de velocidad

- ♦ Al comenzar cada bloque, se graban dos bytes con 85, 85
- ♦ En cinta quedaba: 0010101011 0010101011
- ♦ Al leer, se mide el tiempo transcurrido para procesar esta marca.
- ♦ Con el resultado del cálculo, se programaba el SIO para procesar automáticamente los bits
- ♦ Luego comienza la lectura de bytes desde SIO
- ♦ Cuando fallaba se notaba el tono “desafinado”

# Carga Turbo

- ◆ Al grabar, se programaba SIO para convertir bits a bytes a una velocidad superior a 600 baudios
- ◆ Al cargar:
  - Se usaba el mismo cálculo de BIOS, o bien
  - Se parchaba el cálculo de BIOS (STAC/Injektor)



# Estructura de registros

- ♦ La carga estándar usa 128 bytes de datos por registro
- ♦ La carga turbo usa más de 128 bytes de datos por registro
- ♦ Probabilidad de error se mantiene: Registro tiene más datos pero dura lo mismo en cinta
- ♦ Confiando en el sistema se podían usar bloques extremadamente largos

# Cargador Turbo

- ♦ Reemplazaba al cargador XEX estándar
- ♦ Modificaba la estructura de cada registro y su longitud
- ♦ Al cargar más rápido, se podían usar bloques más largos en forma confiable



Carga Normal a 600 baudios 128 bytes por bloque



Carga Turbo a más de 600 baudios y más de 128 bytes por bloque



# Recuperación de errores



- ◆ Cambia el formato de cada registro
- ◆ El indicador de Tipo de Registro (Normal, Parcial, EOF) se reemplaza
  - El Tipo de Registro ahora es el número de registro
  - 255 indica último registro. No es necesario EOF
- ◆ Se va anotando el número del último registro cargado exitosamente

# Recuperación de Errores



- ♦ Si falla la carga de un bloque, se detiene la cinta
- ♦ Se pide al usuario retroceder la cinta y presionar una tecla
- ♦ Se enciende el motor y se espera el inicio de un bloque (85, 85)
- ♦ Se carga el bloque y si el número de registro es  $\leq$  al último cargado, se ignora
- ♦ El ciclo continúa hasta encontrar un registro nuevo

# Sistema Injektor

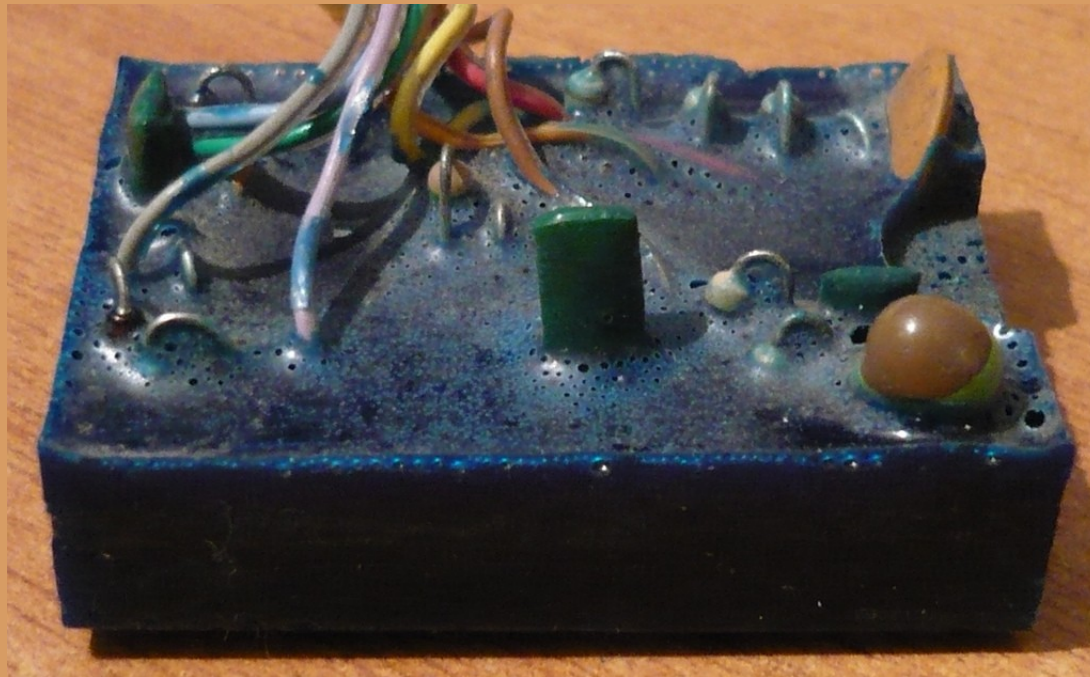


Foto por 133Mhz - <http://133fsb.wordpress.com>

# Sistema Injektor

- ◆ Consiste en una modificación al hardware de la cassetera XC12 de ATARI
- ◆ Provee un stream de bits a frecuencia fija
- ◆ El software no necesita calcular la velocidad, asume un valor fijo
- ◆ Operaría a 2672 baudios según xt5 ([retrogames.cl](http://retrogames.cl) / [ingenieria-inversa.cl](http://ingenieria-inversa.cl))

# ¿Qué tan rápido es Injektor?

- ◆ Es 4,5 veces más rápido que la carga estándar
  - 4,5 minutos para cargar 16 KB (Donkey Kong)
  - **Injektor : 1 minuto**
  - 11 minutos para cargar 40 KB (Black Lamp/Zybex)
  - **Injektor : 2,5 minutos**
  - 18 minutos para cargar 64 KB (toda la memoria)
  - **Injektor : 4 minutos**
  - 13 horas para cargar MP3 de 3 minutos
  - **Injektor : 3,1 horas**

# **Bonus Track : Cargador CAIN**



Paseo Latorre 80 Local 23 U.Alemana

**JOHANNES' S GAMES**

Presenta dentro de 008 bloques a...

**PERROS LOKOS**

CAIN'92 por F.Catrin \* U-Region

# Origenes de CAIN

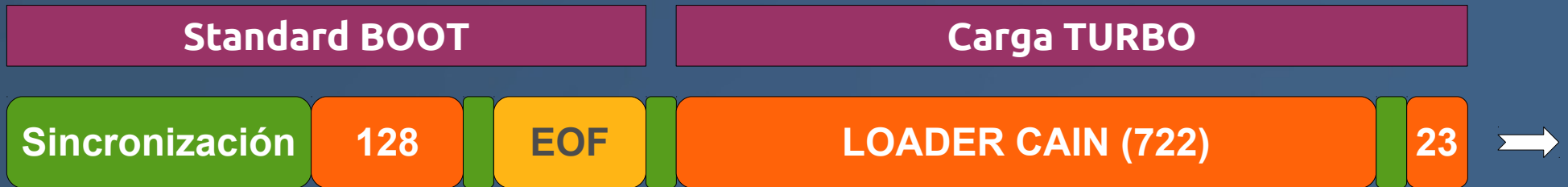
- ◆ Toma ideas de varios cargadores famosos de la época. Los pioneros :
  - STAC (Iljor)
  - SITRE (Vitoco++)
  - NHP (Parche Negro)
- ◆ Originario de Quilpué/Villa Alemana
- ◆ Fue creado para vender “copias de respaldo” a partir de “copias de respaldo” de Santiago que se vendían con sistemas de carga anti-copia ;-)



# Elementos característicos

- ◆ Boot record de un solo bloque
- ◆ Boot record cargaba un bloque “gigante” de 722 bytes
- ◆ Bloque de 722 es un cargador XEX con Turbo y Recuperación de Errores
- ◆ Incorporaba un bloque de 23 bytes con
  - 20 bytes para el nombre del juego
  - 3 bytes para el contador de bloques inicial

# Bloques principales



# Fortalezas y debilidades

- ◆ Ocupaba poco espacio en RAM, entraba rápido al modo Turbo CRE
- ◆ No era relocizable, se debían modificar los programas para no pisarlo
- ◆ Usaba el rango tolerable de las cassetteeras de la época

# Versiones posteriores

- ♦ **CAIN II** es básicamente el mismo cargador, pero incluye una presentación más vistosa
- ♦ **CAIN Injektor** es una versión modificada para usar la frecuencia fija de las caseteras Injektor

# Consultas

