

## Serial EEPROMs in Sega Genesis / Mega Drive cartridges

Version 2 (30/09/10)  
Copyright Eke-Eke

Serial EEPROM devices are present in a few Sega Genesis cartridges and used as external Backup RAM. As you will later see, they can NOT be emulated as usual parallel SRAM and also use various modes depending on their memory size. Most Genesis / Mega Drive emulators out there does not emulate those custom devices, resulting in various glitches (beside the fact that saving game data is impossible) in games that rely on this.

Here are some links to datasheets for each kind of EEPROM types for those who couldn't understand my bad english ;)

24C01(mode1) : <http://www.icmic.com/datasheets/X24C01.pdf>

24C01-24C16(mode2) : [http://www.atmel.com/dyn/resources/prod\\_documents/doc0180.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc0180.pdf)

24C64 (mode3):[http://www.phys.hawaii.edu/~bryce/component\\_data/24LC65-ISM.pdf](http://www.phys.hawaii.edu/~bryce/component_data/24LC65-ISM.pdf)

Basically, the protocol used so that the Main CPU (68000), considered as MASTER device, can communicate with the serial EEPROM (considered as SLAVE device), could be summarized this way:

- SLAVE is controlled by the MASTER using two serial lines:
  - /SDA is the **DATA line**, used to send/receive data (READ/WRITE)
  - /SCL is the **CLOCK line** used to manage the clock cycles (WRITE ONLY)

Each line is mapped in the 68000 address space at a specific address, at one specific bit position (see various used mappers in game database) and can have LOW (bit=0) or HIGH (bit=1) state.

- A clock cycle starts by a transition of /SCL from LOW to HIGH and ends by a transition of /SCL from HIGH to LOW. This is used by the MASTER to indicate when /SDA state refers to a **new valid DATA bit** (received or sent).
- MASTER and SLAVE devices communicate through 8-bits DATA word, which require **8 clock cycles**. The 9th cycle is called the **ACK cycle** where the receiver (SLAVE or MASTER, depending on the current operation) must send an acknowledge to the sender by setting /SDA line LOW during this cycle, when it is ready to receive more data.
- Each READ or WRITE operation must be initiated by the MASTER with a **START** condition and terminated with a **STOP** condition:
  - START condition is set when /SDA is changed from HIGH to LOW while /SCL remains HIGH
  - STOP condition is set when /SDA is changed from LOW to HIGH while /SCL remains HIGH
- After the START condition, depending on the EEPROM mode, the MASTER immediately send one or more **8-bit words** to specify the address that should be later read or written. There are 3 known EEPROM modes:

- Mode 1 (24C01 only) have a maximal size of 128bytes (7 bits address) and need only one address word. The 7 first bits set the memory address to be read/write and the 8th bit set the operation type (READ or WRITE)
- Mode 2 (24C01 to 24C16) can have up to 8 devices linked together, with a maximal size of 2 Kbytes (11 bits address max.), which means one 24C16, two 24C08, four 24C04 or eight 24C02/24C01 :
  - the 1st word has 4 fixed bits, then 3bits setting the device address and/or eventually the upper bits of the memory address to read/write (24C04-24C16). The 8th bit set the type of the operation (READ or WRITE).
  - the 2<sup>nd</sup> word set the lower bits of the memory address to read/write (only 7bits for 24C01 in mode 2, because it's only 128 bytes wide)
- Mode 3 (24C32 and more) work like Mode 2 excepted that the maximal address size is 64 Kbytes (max. 16 bits address) per device (a maximum of eight devices can be linked with no size restriction).
  - the 1st word is like in the previous mode, excepted bits d3-d1 always set the device address (0-7). Last bit always set the operation type.
  - the 2<sup>nd</sup> word set the upper bits of the memory address to read/write (the number of bits depends on the memory size)
  - the 3rd word set the eight lower bits of the memory address to read/write.

**NB:** In modes 2 and 3, before a READ operation, only the 1st word is actually sent, the EEPROM contains an address counter that saves the address of the next word to be accessed (incremented from a previous READ or WRITE operation, as stated below)

- After a WRITE operation has been initiated, the MASTER will send one or more 8bits DATA words through the /SDA line. After each written word, the EEPROM must send a ACK and increment the address for the next data word to be written. The address will roll up to base memory address when maximal write page size has been reached. If the MASTER sends a STOP condition, the WRITE operation ends, otherwise the WRITE operation continues to next address.
- After a READ operation has been initiated, the MASTER will read one or more 8bits DATA words through the /SDA line. After each received word, the MASTER will send a ACK and the EEPROM will increment the address of the next data word to be read. The address will roll up to 0 when maximal memory size has been reached. If the MASTER does not send a ACK during the 9th cycle, the READ operation ends and the EEPROM waits for STOP condition, otherwise, the READ operation continues to next address.

This is pretty much how EEPROM in Genesis Plus (see eeprom.c for more details).

Thanks a lot to 8bitwizard from sriteminds.net forums who initially give me indications about the various EEPROM types and mappers used in Sega Genesis games.

Following is a game database of Sega Genesis games that I found using serial EEPROM, listed by companies (as they usually use specific company mappers). Their specific characteristics (mappers, mode, size,...) are also mentioned, deducted from various testing.

Feel free to use this information and my source code in your own project.

# ACCLAIM

## TYPE #1

**SDA\_IN:** 0x200001 (bit 0)  
**SDA\_OUT:** 0x200001 (bit 1)  
**SCL:** 0x200001 (bit 1)

**MODE:** 8BITS WORD ADDRESS (MODE2)  
**SIZE\_MASK:** 0xFF (24C02)  
**PAGE\_MASK:** 0x03

**NBA Jam (UE)**  
**NBA Jam (J)**

***NB:** these games use a different mapper than any other Acclaim games. Also software use 16-bits write & read to access EEPROM.*

## TYPE #2

**SDA\_IN:** 0x200001 (bit 0)  
**SDA\_OUT:** 0x200001 (bit 0)  
**SCL:** 0x200000 (bit 0)

**MODE:** 8BITS WORD ADDRESS (MODE2)  
**SIZE\_MASK:** 0xFF (24C02)  
**PAGE\_MASK:** 0x03

**NBA Jam Tournament Edition**  
**NFL Quarterback Club**

***NB:** Rev 00 of NBA Jam TE has buggy EEPROM support, only Rev01 (unofficial) version will correctly save game data.*

**MODE:** 8BITS WORD ADDRESS (MODE2)  
**SIZE\_MASK:** 0x7FF (24C16)  
**PAGE\_MASK:** 0x07

**NFL Quarterback Club 96**

**MODE:** 16BITS WORD ADDRESS (MODE3)  
**SIZE\_MASK:** 0x1FFF (24C64)  
**PAGE\_MASK:** 0x07

**College Slam**  
**Frank Thomas Big Hurt Baseball**

## CODEMASTERS

**SDA\_IN:** 0x300000 (bit 0)  
**SDA\_OUT:** 0x380001 (bit 7)  
**SCL:** 0x300000 (bit 1)

**MODE:** 7BITS WORD ADDRESS (MODE1)  
**SIZE\_MASK:** 0x7F (X24C01)  
**PAGE\_MASK:** 0x03

### Brian Lara Cricket

**MODE:** 8BITS WORD ADDRESS (MODE2)  
**SIZE\_MASK:** 0x3FF (24C08)  
**PAGE\_MASK:** 0x0F

### Micro Machines 2 - Turbo Tournament Micro Machines Military

***NB:** Micro Machines 2 requires backup memory to be initialized with 0xFF*

**MODE:** 8BITS WORD ADDRESS (MODE2)  
**SIZE\_MASK:** 0x7FF (24C16)  
**PAGE\_MASK:** 0xF

### Micro Machines Turbo Tournament 96

**MODE:** 16BITS WORD ADDRESS (MODE3)  
**SIZE\_MASK:** 0x1FFF (24C64)  
**PAGE\_MASK:** ?

### Brian Lara Cricket 96 / Shane Warne Cricket

## ELECTRONIC ARTS

**SDA\_IN:** 0x200001 (bit 7)  
**SDA\_OUT:** 0x200001 (bit 7)  
**SCL:** 0x200001 (bit 6)

**MODE:** 7BITS WORD ADDRESS (MODE1)  
**SIZE\_MASK:** 0x7F (X24C01)  
**PAGE\_MASK:** 0x03

### Bill Walsh College Football John Madden Football 93 John Madden Football 93 - Championship Edition NHLPA Hockey 93 Rings of Power

***NB:** These games use 16-bits write & read to access EEPROM.*

## SEGA

**SDA\_IN:** 0x200001 (bit 0)  
**SDA\_OUT:** 0x200001 (bit 0)  
**SCL:** 0x200001 (bit 1)

**MODE:** 7BITS WORD ADDRESS (MODE1)  
**SIZE\_MASK:** 0x7F (X24C01)  
**PAGE\_MASK:** 0x03

**Evander 'Real Deal' Holyfield's Boxing (J,UE)**  
**Greatest Heavyweights of the Ring (J,U,E)**  
**Honoo no Toukyuuji Dodge Danpei**  
**MLBPA Sports Talk Baseball**  
**Ninja Burai Densetsu**  
**Wonder Boy V - Monster World III (J)**  
**Wonder Boy in Monster World (UE)**

## CAPCOM

**SDA\_IN:** 0x200001 (bit 0)  
**SDA\_OUT:** 0x200001 (bit 0)  
**SCL:** 0x200001 (bit 1)

**MODE:** 7BITS WORD ADDRESS (MODE1)  
**SIZE\_MASK:** 0x7F (X24C01)  
**PAGE\_MASK:** 0x03

**Megaman - The Wily Wars (E)**  
**Rockman Mega World (J)**

**NB:** *an alternate version of Rockman Mega World, which uses traditional SRAM, was also released.*