

Foreword
Acknowledgements
History
<b>Overview</b>
1. Specifications
2. Memory Map
<b>I/O Ports</b>
3. Rendering
4. Sound Controller
5. Joypad Input
6. <a href="#">Serial Data Transfer</a>
7. Timer and Divider Registers
8. Interrupts
9. CGB Registers
10. Infrared Communication
11. SGB Functions
<b>CPU Specifications</b>
12. CPU Registers and Flags
13. CPU Instruction Set
14. CPU Comparison with Z80
<b>Cartridges</b>
15. The Cartridge Header
16. MBCs
<b>Accessories</b>
17. Game Boy Printer
18. Game Boy Camera
19. 4-Player Adapter
20. Game Genie/Shark Cheats
<b>Other</b>
21. Power-Up Sequence
22. Reducing Power Consumption
23. Accessing VRAM and OAM
24. OAM Corruption Bug
25. External Connectors
26. GBC Approval Process
References

# Serial Data Transfer (Link Cable)

Communication between two Game Boy systems happens one byte at a time. One Game B generates a clock signal internally and thus controls when the exchange happens. In SPI terms, the Game Boy generating the clock is called the “master.” The other one uses an external clock (receiving it from the other Game Boy) and has no control over when the transfer happens. If it hasn’t gotten around to loading up the next data byte at the time th transfer begins, the last one will go out again. Alternately, if it’s ready to send the next by but the last one hasn’t gone out yet, it has no choice but to wait.

## FF01 - SB - Serial transfer data (R/W)

Before a transfer, it holds the next byte that will go out.

During a transfer, it has a blend of the outgoing and incoming bytes. Each cycle, the leftmost bit is shifted out (and over the wire) and the incoming bit is shifted in from the other side:

```
o7 o6 o5 o4 o3 o2 o1 o0
o6 o5 o4 o3 o2 o1 o0 i7
o5 o4 o3 o2 o1 o0 i7 i6
o4 o3 o2 o1 o0 i7 i6 i5
o3 o2 o1 o0 i7 i6 i5 i4
o2 o1 o0 i7 i6 i5 i4 i3
o1 o0 i7 i6 i5 i4 i3 i2
o0 i7 i6 i5 i4 i3 i2 i1
i7 i6 i5 i4 i3 i2 i1 i0
```

## FF02 - SC - Serial Transfer Control (R/W)

Bit 7 - Transfer Start Flag (0=No transfer is in progress or requested, 1=Transfer in p  
Bit 1 - Clock Speed (0=Normal, 1=Fast) \*\* CGB Mode Only \*\*  
Bit 0 - Shift Clock (0=External Clock, 1=Internal Clock)

The master Game Boy will load up a data byte in SB and then set SC to 0x81 (Transfer requested, use internal clock). It will be notified that the transfer is complete in two ways: SC’s Bit 7 will be cleared (that is, SC will be set up 0x01), and also the [Serial Interrupt handler](#) will be called (that is, the CPU will jump to 0x0058).

The other Game Boy will load up a data byte and can optionally set SC’s Bit 7 (that is, SC=0x80). Regardless of whether or not it has done this, if and when the master wants to conduct a transfer, it will happen (pulling whatever happens to be in SB at that time). The externally clocked Game Boy will have its [serial interrupt handler](#) called at the end of the transfer, and if it bothered to set SC’s Bit 7, it will be cleared.

### Internal Clock

In Non-CGB Mode the Game Boy supplies an internal clock of 8192Hz only (allowing to transfer about 1 KByte per second minus overhead for delays). In CGB Mode four internal clock rates are available, depending on Bit 1 of the SC register, and on whether the CGB Double Speed Mode is used:

Clock freq	Transfer speed	Conditions
8192Hz	1KB/s	Bit 1 cleared, Normal speed
16384Hz	2KB/s	Bit 1 cleared, Double-speed Mode
262144Hz	32KB/s	Bit 1 set, Normal speed
524288Hz	64KB/s	Bit 1 set, Double-speed Mode

### External Clock

The external clock is typically supplied by another Game Boy, but might be supplied by another computer (for example if connected to a PC’s parallel port), in that case the exter clock may have any speed. Even the old/monochrome Game Boy is reported to recognize external clocks of up to 500 kHz. And there is no limitation in the other direction: even wl suppling an external clock speed of “1 bit per month,” the Game Boy will eagerly wait for the next bit to be transferred. It isn’t required that the clock pulses are sent at a regular interval either.

### Timeouts

When using external clock then the transfer will not complete until the last bit is received case that the second Game Boy isn’t supplying a clock signal, if it gets turned off, or if the is no second Game Boy connected at all) then transfer will never complete. For this reason

the transfer procedure should use a timeout counter, and abort the communication if no response has been received during the timeout interval.

## Delays and Synchronization

The master Game Boy should always execute a small delay after each transfer, in order to ensure that the other Game Boy has enough time to prepare itself for the next transfer. This is, the Game Boy with external clock must have set its transfer start bit before the Game Boy with internal clock starts the transfer. Alternately, the two Game Boy systems could switch between internal and external clock for each transferred byte to ensure synchronization.

Transfer is initiated when the master Game Boy sets its Transfer Start Flag, regardless of the value of this flag on the other device. This bit is automatically set to 0 (on both) at the end of transfer. Reading this bit can be used to determine if the transfer is still active.