

JoySNES Documentation

14-09-2017



Index

Description.....	2
Use.....	2
Joypad.....	2
Mouse.....	2
Operating modes.....	3
① Standard mode.....	3
② Extended mode.....	3
Know the version of his JoySNES.....	3
Advanced programming for the JoySNES.....	4
Read the status of all the buttons of a SNES controller.....	5
① Method for "fast access".....	5
② Method for "direct access".....	7
Reading adapter information.....	9

Description

This adapter allows you to play with a controller for SNES* on MSX as if it were a controller for MSX with some additional features like auto-fire, button swapping, START and SELECT buttons like joypads for FM-Towns or 8 buttons independently operable.

The adapter also allows you to use a mouse for SNES with most MSX software.

(*) Super Nintendo Entertainment System or Super Famicom.

About the [8bitdo Retro Receiver for SNES](#) :

The 8bitdo Retro Receiver works very well on MSX with the JoySNES adapter. However, you must be careful because only one consumes the equivalent of what the two joystick ports of a standard MSX can provide. Most recent MSXs have a power supply more powerful than stipulated by standard, but I still advise not to use two Retro Receiver at the same time, especially if the MSX has inserted cartridges that consume a lot of current.

Use

Joypad

By default, the buttons have the following functions for all MSX games that do not support additional buttons independently.

SNES Joypad		MSX
A button	=	A button
B button	=	B button
L trigger	=	B button
R trigger	=	A button
X button	=	A button with auto-fire
Y button	=	B button with auto-fire
SELECT	=	Up+Down (as FM-Town)
START	=	Left+Right (as FM-Town)

Combinations for button inverting:

SNES Joypad
SELECT+START+L* = Invert L with R
SELECT+START+R* = Invert A with B, and X with Y

(*) These combinations must be maintained for about three seconds.

Mouse

The SNES mouse is detected automatically when plugged and it is used as an MSX mouse. However, there are a few software that work less well than with a real MSX mouse. The mouse can also be used as a joystick like MSX and FM-Towns by pressing the left button when turning ON the MSX.

Operating modes

The adapter has 2 operating modes for the SNES controller.

① Standard mode

This mode is set to default when manufacturing, or by pressing the Y button when power is turned ON. In this mode the controller acts as a joystick for FM-Town with fast auto-fire. Joysticks for FM-Town are 100% MSX compatible with two additional START and SELECT buttons. The SELECT button corresponds to the Up+Down directions pressed at the same time and START corresponds to Left+Right.

This mode is present only for games which change the signal of pin 8 intensively. It is therefore fully compatible but it does not allow the use of additional buttons independently.

② Extended mode

By pressing the X button when you turn on the MSX, the adapter switches to extended mode and stays in this mode until the next change. Extended mode is compatible only with 99.9% of the MSX games but it allows to use independently all buttons of the SNES controller in your programs. After many tests, only three games did not work in extended mode. They are "Beamrider", "Choro Q" and "Pitfall".

In extended mode, in addition to the standard method of reading the status of the direction and the two buttons by setting pin 8 of the Joystick port to 0, there are two other possible ways to read the state of all buttons. The first is the fastest. All buttons can be read four by four by setting the state of pin 8 to high. (See paragraph "[Method for "fast access"](#)" for details.)

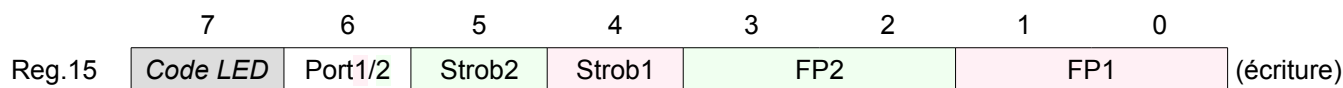
The other method is close to the original used on SNES. Each button status is read one by one via the 16-bit register of the joystick. (See paragraph "[Method for "direct access"](#)" for explanations.)

Know the version of his JoySNES

It is possible to know the version and the ID of the adapter. The method is explained in the paragraph "[Reading adapter information](#)".

Advanced programming for the JoySNES

To communicate with the JoySNES adapter, PSG registers 14 and 15 must be used in several steps. Each step includes sending data to the adapter via the register 15 and reading the register 14 to know the result of the operation.



- **FP1** bits are used to indicate the function or the step to be performed by the adapter connected to the Joystick port 1.
- **FP2** bits are used to indicate the function or the step to be performed by the adapter connected to the port 2.
- **Strob1** must be set to 1 during the steps to be performed by the adapter at port 1.
- **Strob2** must be set to 1 during the steps to be performed by the adapter at port 2.
- **Port1/2** is used to indicate the Joystick port (0/1 = port 1/2) via which the data will be read from register 14.
- *Code LED is used to turn on/off the LED of the Code key or Kana key for Japanese Keyboards.*



- **Data0 to Data3** give the status of the button(s) of the step performed.
- **B6** gives the status of button 1 of the joystick or MSX mouse.
- **B7** gives the status of button 2 of the joystick or MSX mouse.
- *JIS is used to indicate the keyboard type for Japanese MSX.*
- *CASRD is used for the cassette.*

In first step, you must write to FPX bits the function number that will determine whether it is a reading of the state of the joystick, or a reading of the character from string used to identify the adapter, or direct reading from SNES controller. The values that follow (and the results) will depend on the chosen function.

After writing the number of the function with StrobX at 1 in register 15, it takes about 30.8 microseconds before it can read the bits **Data0** to **Data3** of register 14. For the following steps, it will take a delay of about 16.49 microseconds before the register can be read. To exit the function, StrobX must be set to 0.

When the StrobX bit is set to 0 and an SNES controller is connected, the status of its buttons is scanned and it is updated by the adapter every 431 microseconds so that the controller is read as a standard MSX joystick of type B (with two independent buttons) plus the supplements offered by the adapter (auto-fire, buttons A and B inversion, etc.).

Note: 20µs of delay is required between each change of state of pin 8 (StrobX) so that the adapter can take account of the change.

Read the status of all the buttons of a SNES controller

In extended mode, JoySNES adapter has two different methods for reading the status of all the buttons on an SNES controller.

① Method for "fast access"

This method allows you to read the status of all the buttons of an SNES controller more quickly than with the bit-by-bit reading used on SNES.

Method to follow (for the adapter in Joystick port 1):

Step 1 (FP1=01)

Read register 15 and store the bit value of LED Code (or KANA).

Write to the register 15 the following bits:
LED Code at read value, Port1/2 to 0, Strob1 to 1 and FP1 to 01 (for the « fast access" method).

Wait 30.8 μ s

Read the register 14.
The bits Data0 to Data3 give the status of the direction arrow (Up, Down, Left and Right).

Step 2 (FP1=10)

Write to the register 15 the following bits:
LED Code to read value, Port1/2 to 0, Strob1 to 1 and FP1 to 10 (to read the status of A, X, Left and Right buttons).

Wait 16.49 μ s

Read the register 14.
The bits Data0 to Data3 give the status of the A, X, Left and Right buttons.

Step 3 (FP1=11)

Write to the register 15 the following bits:
LED Code to read value, Port1/2 to 0, Strob1 to 1 and FP1 to 11 (to read the status of B, Y, Select and Start buttons).

Wait 16.49 μ s

Read the register 14.
The bits Data0 to Data3 give the status of B, Y, Select and Start buttons.

END

Write to register 15 to set the Strob1 bit to 0.
(20 μ s of delay will be necessary before Strob1's next set).
Once Strob1 is set to 0, there will be a delay of 431 μ s before the adapter refreshes the buttons status.

Notes :

- Step 2 and 3 can be reversed and are not mandatory.
- If the adapter is in port 2, set the Port1/2 bit to 1 and use Strob2 and FP2 instead of Strob1 and FP1.
- The routine in assembler (file "Reading_method1.gen") is provided in the archive.

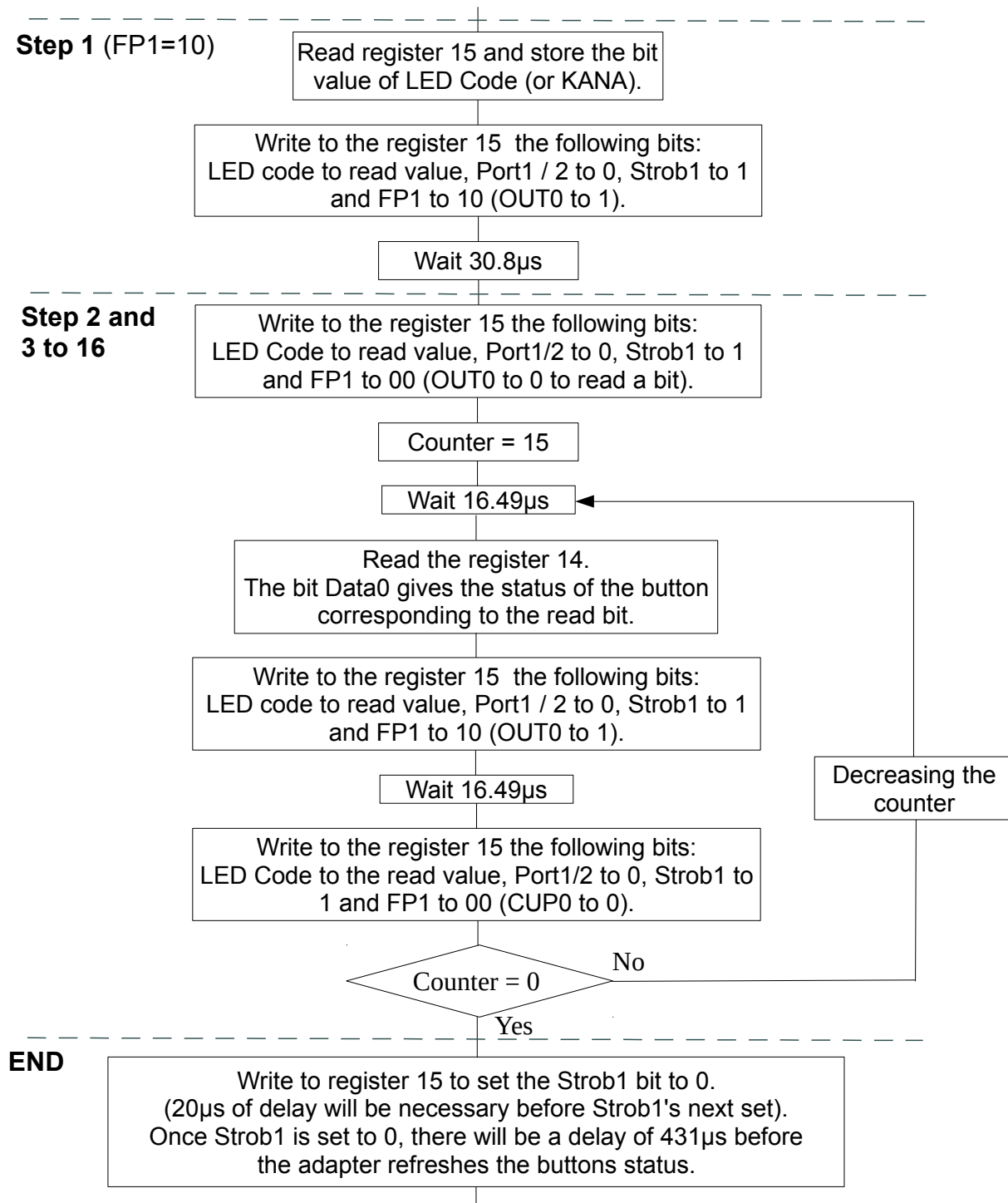
② Method for "direct access"

This method allows to read directly the 16-bit register status of the SNES controller buttons. Reading is done bit by bit as on the SNES. This method should allow reading other devices except the SNES mouse that is automatically detected by the adapter to be managed as an MSX mouse.

Status Register of SNES controller buttons:

bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
B	Y	Select	Start	Up	Down	Left	Right	A	X	L	R	Controller ID			

Method to follow (for the adapter in Joystick port 1):

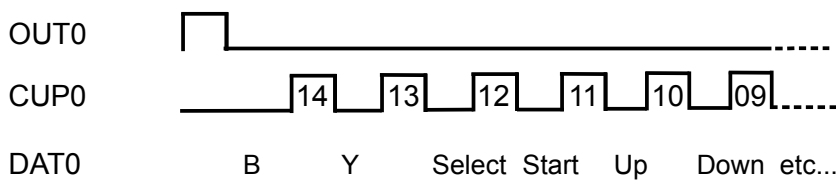


Notes :

- In the final step, you can leave Strob1 at 1 and resume directly in step 1 with a delay of 16.49µs instead of 30.8µs.
- If the adapter is in port 2, set the Port1/2 bit to 1 and use Strob2 and FP2 instead of Strob1 and FP1.
- The routine in assembler (file "Reading_method2.gen") is provided in the archive.

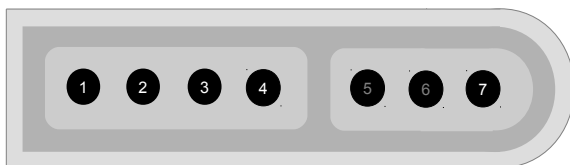
Timing diagram of read bit by bit:

The SNES hardware uses a serial link. To read the status of the buttons and direction of the controller by direct access, send a pulse on OUT0, then 15 pulses on CUP0 and read at each step the result on DAT0 (via Data0 bit of the register 14 of PSG).



The SNES connector has three signals to control hardware such as a joystick or other.

SNES Controller Port Pinout:



- 1 = VDD (+5V)
- 2 = CUP0 (clock) Output
- 3 = OUT0 (P/S) Output
- 4 = DAT0 (Data) Input
- 5 et 6 = Not connected
- 7 = GND

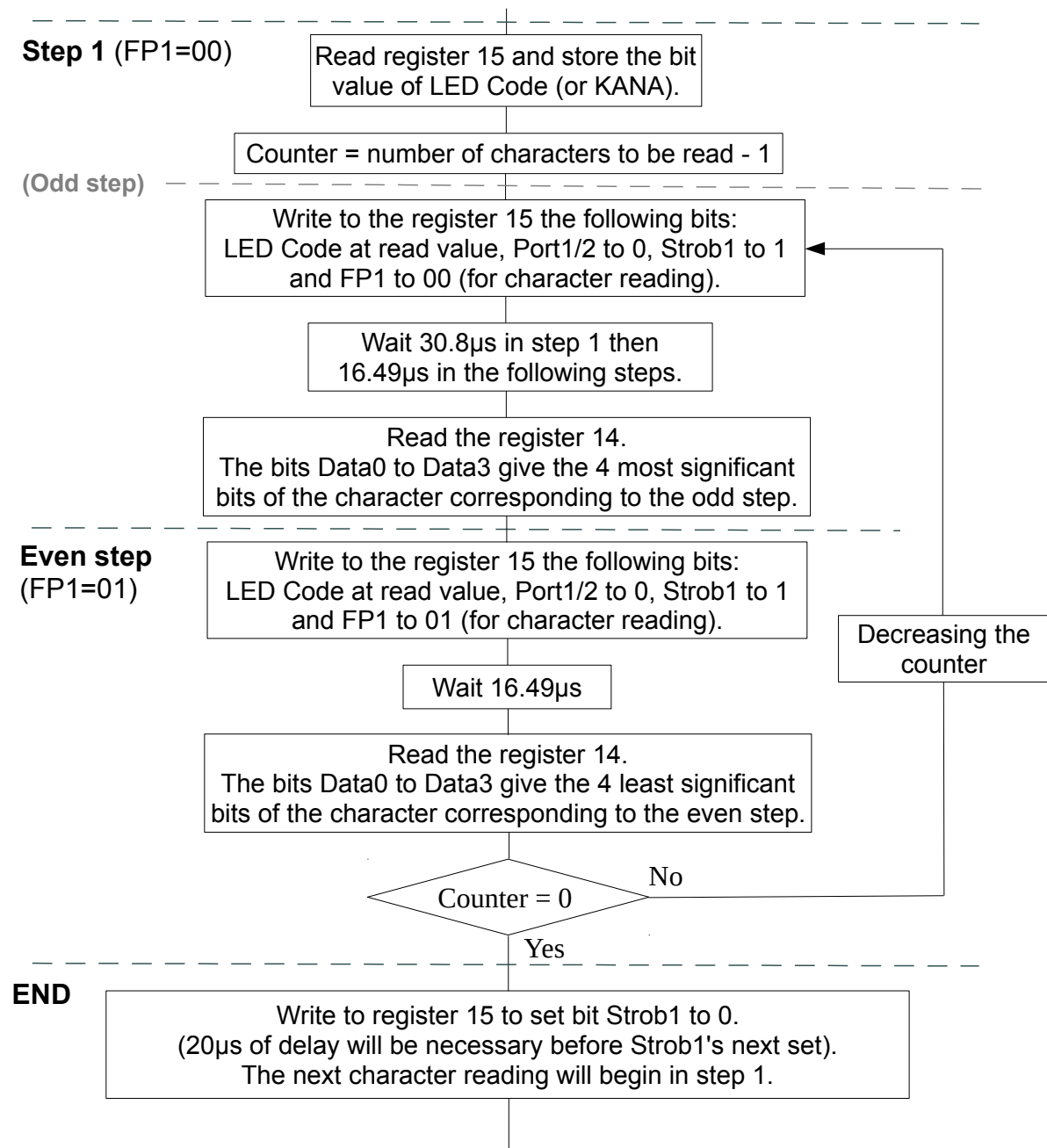
Reading adapter information

The adapter has a character string used to identify the type and the version of the adapter. This string has a length of 50 characters. It contains the following characters. The identifier takes 7 characters and the version number takes 4.

JOYSNES v1.0 - SNES Adapter by Popolon (fr) - 2017

First and Ninth characters (the following characters are just a little signature)

Method to follow (for the adapter in the port Joystick 1):



Notes :

- If the adapter is in port 2, set the Port1/2 bit to 1 and use Strob2 and FP2 instead of Strob1 and FP1.
- The routine in assembler (file "Reading_info.gen") is provided in the archive.