



# Arcade Joystick X4 Plus Simon Says Game

[hackster.io/user3853574654/arcade-joystick-x4-plus-simon-says-game-384309](https://hackster.io/user3853574654/arcade-joystick-x4-plus-simon-says-game-384309)

David Guerrero






## Things used in this project

### Hardware components

<u>arcade joystick</u>	× 4
------------------------	-----

<u>generic arcade button</u>	× 24
------------------------------	------

<u>red retroilluminated arcade button</u>	× 1
---	-----

	<u>green retroiluminated arcade button</u>	x 1	
	<u>blue retroiluminated arcade button</u>	x 1	
	<u>yellow retroiluminated arcade button</u>	x 1	
	<u>white retroiluminated arcade button</u>	x 1	
	<u>Buzzer</u>	x 1	<ul style="list-style-type: none"> <li>• <a href="#">Buy from Newark</a></li> <li>• <a href="#">Buy from Adafruit</a></li> </ul>
	<u>Resistor 1k ohm</u>	x 1	
	<u>1N5819 – 1A Schottky Barrier Rectifier</u>	x 48	
	<u>Teensy++ 2.0 board</u>	x 1	

---

## Software apps and online services

---

avr-gcc compiler

---



---

---

Teensy Loader

---

GNU Make tool

---

firmware

## Story

---

### USB arcade joystick x4 plus Simon game

**David Guerrero Martos**

#### 1 Introduction

##### 1.1 Motivation

Thanks to the emulators available nowadays I was able to make my own arcade cabinet. Although I employed genuine arcade joysticks and a vintage CRT display I didn't get the same feeling that playing the original arcades. When playing the original, death was dramatic: if you "died" in the game and you wanted to continue playing you had to spend a valuable coin. Since emulators let you simulate the insertion of a coin just by pressing a key, in practical terms you have unlimited lives and the feeling is lost. To solve this I devised a joystick system with an integrated credit counter. The system disables the insert coin buttons unless credits are available and includes a little challenge that must be won in order to obtain them.

##### 1.2 The challenge

It is the Simon says electronic game created by Ralph Baer and Howard J. Morrison in 1978. The following description of the game has been taken from [Wikipedia](#):

The device has four colored buttons, each producing a particular tone when it is pressed or activated by the device. A round in the game consists of the device lighting up one or more buttons in a random order, after which the player must reproduce that order by pressing the buttons. As the game progresses, the number of buttons to be pressed increases.

As in the original Simon says game there are several skill levels. The number of credits obtained when the game is won will depend on the selected skill level.

##### 1.3 Features

Includes four arcade joysticks using just an USB connector.

No special drivers are needed.

Each joystick has six generic buttons, a start button and a retro-illuminated insert coin button.

The insert coin buttons are also employed in an integrated Simon says game in order to obtain credits.

The insert coin buttons are disabled when there are no credits available.

## 2 What is needed

### 2.1 Hardware

Four arcade joysticks and, for each of them, six generic buttons and an start button

Five retro-illuminated arcade colored buttons (yellow, blue, red, green and white)

A little speaker or buzzer

At least a resistor of about 1000 ohms. The lights of the retro-illuminated buttons may require additional series resistors.

48 Schottky diodes

A Teensy++ 2.0 board (available at <https://www.pjrc.com/teensy/>)

Interconnection wires

### 2.2 Firmware

The compiled firmware can be downloaded from [http://www.dte.us.es/Members/guerre/firmware\\_arcade/at\\_download/file](http://www.dte.us.es/Members/guerre/firmware_arcade/at_download/file). The source code is available at <https://github.com/gambaman/ArcadeJoystickX4PlusSimonGame>.

### 2.3 Software

In order to program the board you will need the Teensy Loader application available at <https://www.pjrc.com/teensy/loader.html>. If you also want to recompile the firmware you will need the [avr-gcc compiler](#) as well as the [GNU Make tool](#).

Of course, in order to enjoy playing you will need games and/or emulators, for example the [MAME emulator](#).

## 3 Building

### 3.1 Hardware

The components must be wired as shown in the following schematic:

The blue, yellow, green and red retro-illuminated insert coin buttons must be connected to the lines v<sub>gnd1</sub>, v<sub>gnd2</sub>, v<sub>gnd3</sub> and v<sub>gnd4</sub> respectively. The white retro-illuminated button is tagged as "central button" and has special functions. Note that, in my case, the lights of the retro-illuminated buttons can be connected directly to a voltage of 5V like the one generated by the board. Other models may require additional series resistors in order to limit the current through the LEDs. In case of doubt look at its datasheet.

## 3.2 Firmware

### 3.2.1 Compiling

This step is not required unless you want to modify the source code. From a terminal/command line go to the folder/directory containing the source code and execute the following:

```
make clean
```

```
make
```

This will generate a `.hex` file containing the compiled firmware. After connecting the teensy board you can program it as shown in the next section or, if you have installed the command line version of the teensy loader application, program it directly by executing this:

```
make program
```

You will be requested to push the button of the teensy board. Do so and the firmware will be downloaded.

### 3.2.2 Programming

You can find a detailed description of the following steps at <https://www.pjrc.com/teensy/loader.html>.

Connect the teensy board to your computer.

Execute the teensy loader application.

Push the button of the teensy board.

From the File name, choose "Open HEX File" and open the `.hex` file containing the firmware to be programmed.

Select "Program" from the "Operations" menu, or click the Program button on the tool bar. You should see the "Download Complete" message.

Choose "Reboot" from the "Operations" menu, or click the Reboot button on the tool bar.

## 3.3 Software

When connecting the system to your computer it will recognize a set of joysticks. No special drivers should be needed. However, if the system is going to be used with arcade emulation software you will need to configure the emulator properly. First you will need to set the first four detected interfaces as the joysticks of the players 1, 2, 3 and 4. Also you will have to set the last button of each joystick as insert coin button. The fifth interface is used for interacting with the emulator and is not linked to any joystick by default (see the next section). You will have to configure its buttons for tasks such as pausing the emulation, resetting the emulated system, etc.

## 4 Usage

### 4.1 Interacting with emulators

Sending commands to the emulator (for example to pause or quit the emulation) requires setting one of the joysticks as emulation controller. To do so, while pushing the white retro-illuminated button, push the insert coin button of the joystick to be set as emulation controller. That button will light up to indicate that the associated joystick does not work as game controller anymore. Its buttons will have associated the functions you configured in the previous section. If you want to change the joystick to be used as emulation controller repeat the previous step but pushing the insert coin button of another joystick. You can make the joystick used as emulation controller work as game controller again. To do so just press its insert coin button again while holding the retro-illuminated white button pressed. After that the insert coin button will light down.

### 4.2 Getting credits

In order to obtain credits you will have to play the embedded Simon says game by following these steps:

Push the white retro-illuminated button. All the insert coin buttons will light up.

Select the skill level by pushing one of the retro-illuminated buttons. The following table describes each skill level:

The system will generate sequences of lights and sounds. Reproduce those sequences by pressing the buttons till the white button flashes. That will indicate you have won the game and got one or more credits. The white button will keep shining as long as you have credits.

Alternatively, if an insert coin button is pressed when there are no available credits then a Simon says game will start in the easiest level.

## Acknowledgments

The firmware is based on the Teensy Gamepad project by Josh Kropf ([josh@slashdev.ca](mailto:josh@slashdev.ca)) which in turn is based on the keyboard example for the Teensy board ([http://www.pjrc.com/teensy/usb\\_keyboard.html](http://www.pjrc.com/teensy/usb_keyboard.html)), Copyright (c) 2008 PJRC.COM, LLC).

The Simon game replication was possible thanks to the reverse engineering carried out by Simon Inns ([http://www.waitingforfriday.com/index.php/Reverse\\_engineering\\_an\\_MB\\_Electronic\\_Simon\\_game](http://www.waitingforfriday.com/index.php/Reverse_engineering_an_MB_Electronic_Simon_game)).

## Videos

You can see the system working at <https://youtu.be/ENW7noni5kg>.

[Read more](#)

## Schematics

---

### Circuit schematic

---

### Code

---

### Github

---

<https://github.com/gambaman/ArcadeJoystickX4PlusSimonGame>

## Credits

---

