USB arcade joystick x4 plus Simon says game

David Guerrero Martos

January 17, 2023

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 set 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 bee taken from wikipedia (`https://en.wikipedia.org/wiki/Simon_%28game%29`):

> 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 till twelve generic buttons and a backlighted 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, twelve generic buttons
- Four backlighted arcade colored buttons (yellow, blue, red and green)
- Four buttons to select the difficult level of the Simon Says game and start it
- Six buttons to control the emulator
- A little speaker or buzzer
- At least a resistor of about 1000 ohms. The lights of the backlighted buttons may require additional series resistors.
- 72 Schottky diodes
- A Teensy++ 2.0 board (available at `https://www.pjrc.com/teensy`)
- A HD44780 liquid-crystal display
- Interconnection wires

2.2 Firmware

The compiled firmware and the source code is available at `https://hackaday.io/project/189223-enhanced-arcade-j` `files`.

2.3 Software

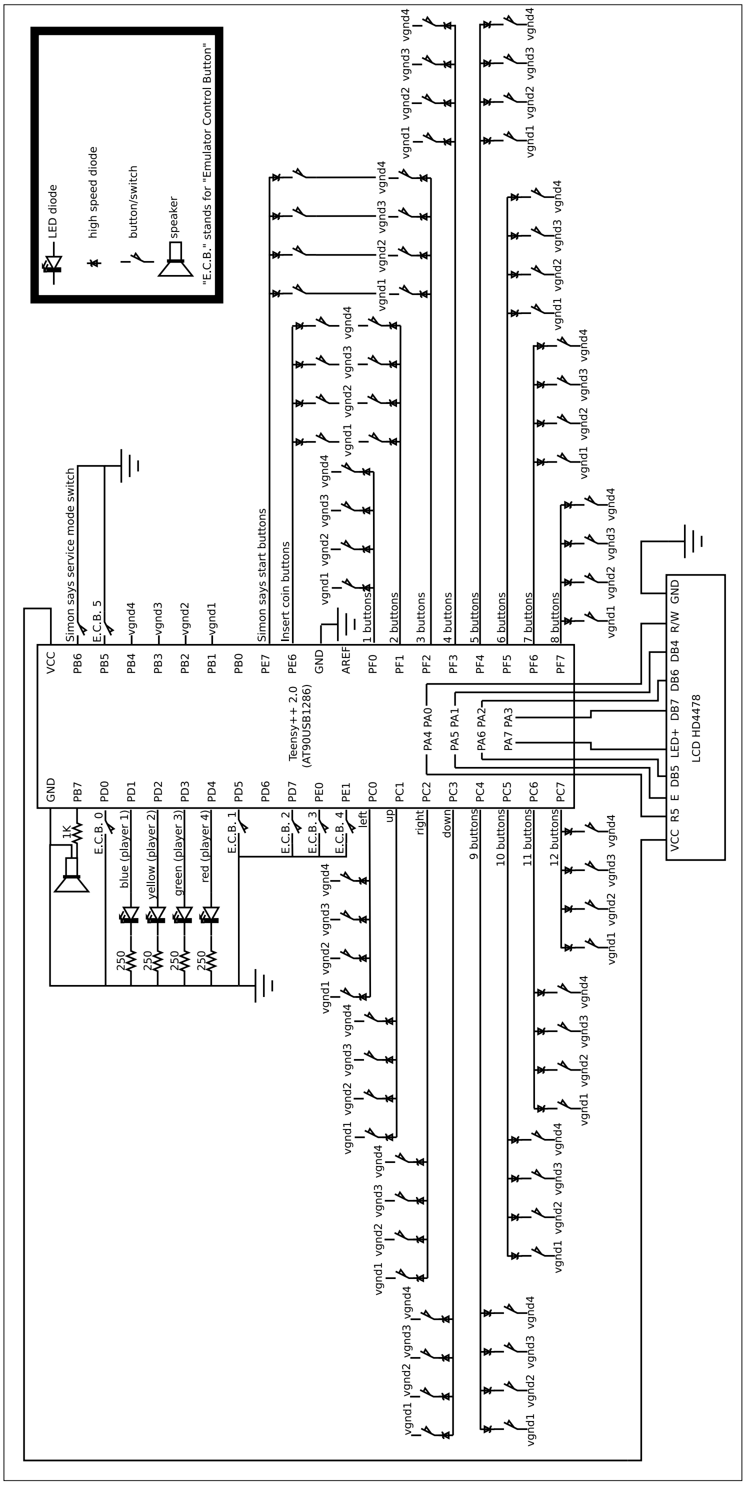
In order to program the board you will need the Teensy Loader application available at `https://www.pjrc.com/teensy/loader.html`. If you also want to recompile the firmware you will need the avr-gcc compiler (`http://www.nongnu.org/avr-libc/`) as well as the GNU Make tool (`http://www.gnu.org/software/make/`).

Of course, in order to enjoy playing you will need games and/or emulators, for example the MAME emulator (`http://mamedev.org/`).

3 Building

3.1 Hardware

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


LED diode
high speed diode
button/switch
speaker
"E.C.B." stands for "Emulator Control Button"
Teensy++ 2.0
(AT90USB1286)
VCC
PB6
PB5
PB4
PB3
PB2
PB1
PB0
PE7
PE6
GND
AREF
PF0
PF1
PF2
PF3
PF4
PF5
PF6
PF7
GND
PB7
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
PE0
PE1
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PA4 PA0
PA5 PA1
PA6 PA2
PA7 PA3
Simon says service mode switch
E.C.B. 5
vgnd4
vgnd3
vgnd2
vgnd1
Simon says start buttons
Insert coin buttons
1 buttons
2 buttons
3 buttons
4 buttons
5 buttons
6 buttons
7 buttons
8 buttons
9 buttons
10 buttons
11 buttons
12 buttons
up
right
down
left
1K
E.C.B. 0
E.C.B. 1
E.C.B. 2
E.C.B. 3
E.C.B. 4
blue (player 1)
yellow (player 2)
green (player 3)
red (player 4)
250
250
250
250
LCD HD4478
VCC
R5
E
DB5
LED+
DB7
DB6
DB4
R/W
GND

The blue, yellow, green and red backlighted insert coin buttons must be connected to the lines vgnd1, vgnd2, vgnd3 and vgnd4 respectively. The lights of some backlighted 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 data sheet.

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 all

This will generate a .hex file containing the compiled firmware. After connectig 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 stelps at `https://www.pjrc.com/teensy/loader.html`.

- Connect the teensy board to your computer.
- Execute the teensy loader application.
- Push the botton 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 configuration

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 for the players 1, 2, 3 and 4. The fifth interface is used for interacting with emulators. From now on we will call it *control interface*. Although the operating system will detect the control interface as 12 buttons joystick, just the buttons in the ranges 1-6 and 9-12 are implemented. Those in the range 9-12 are intended to be used for coin-op game emulators and should be configured as the insert coin button for the player 1, 2, 3 and 4 respectively. The six remaining buttons of the control interface can be configured for tasks such as pausing the emulation, resetting the emulated system, etc.

4 Usage with coin-op game emulators

The joystick set has a counter of credits to be used in coin-op game emulators. The number of available credits is shown in the liquid crystal display. If there are available credits and a coin-op game emulator is running, a player can simulate a coin insertion by following the following instructions:

1. If a Simon Says game is running (see bellow), wait till it finishes.
2. Make sure a coin-op game emulator is running. Otherwise you will lose a credit.
3. Make sure the emulated game support your player number. Otherwise you will lose a credit.
4. Push the insert coin button of your player.

After this, the credit counter will decrease. In order to increase it you will have to play the Simon Says game embedded in the joystick set by following these steps:

1. Push the Simon Says start button corresponding to the desired difficult level.
2. The system will generate sequences of lights and sounds. Reproduce those sequences by pressing the colored insert coin buttons till the liquid crystal display flashes.

The maximum length of the sequence and the number of credits obtained when winning a Simon Says game depend on the selected difficult level as shown in the following table:

difficult level	1	2	3	4
maximum length of the sequence	8	14	20	31
credits obtained when winning the game	5	50	500	unlimited

Alternatively, if an insert coin button is pressed when there are no available credits then a Simon says game will start in a extra-easy non canonical mode. This mode provides an only credit and its maximum length sequence is four.

5 Service mode

The system includes a switch that sets the embedded Simon Says game in a special service mode when closed. The behavior described above corresponds to the normal mode. In service mode the Simon Says game is won as soon as it is started. This makes it possible to increase the credit counter by just pushing a button.

Acknowledgments

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

The LCD library was developed by Efthymios Koktsidis (https://github.com/efthymios-ks/AVR-HD44780, Copyright (c) 2016 Efthymios Koktsidis).

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`).