The Super Nostalgia Emulation System

Claesen Maties

maties.claesen@student.uhasselt.be

Abstract: The world is filled with awesome people who have awesome memories of their childhood. One of these childhood memories included gaming. In this project we are going back to the 20th century by building a prototype portable game-emulator with tons of handmade improvements. This is all possible because of the software support of tons of programmers wanting to keep these game-experiences alive. This project will result in a modern yet retro approach to the classic gaming entertainment systems of yesteryear.

Introduction

The SNES-system will be a console that can emulate less powerful software from consoles like the NES, Gameboy and Gameboy Color, etc. via Github libraries called ESPlay[1] or ESP32-NESEMU[2]. These games will be controlled by button inputs and shown on the display. It will also include Bluetooth Interfaces to make setting adjustments and connect an external PS4-controller[3]. This project is inspired by an article[4] about using the ESP 8-bit to emulate 8-bit games.

Description of the setup

The basic setup will go as follows. You use the button inputs and perhaps a joystick input to control the games (which will be supported by the library). Then the display will show the game and a speaker will produce the game sound.

The types of inputs will be mostly buttons rearranged in a manner to signify a d-pad and face buttons. A turbo-button mode is also going to be available. A potmeter will be used to signify the frequency of button presses the turbo-button mode will supply. A flickering LED will show the simulated turbo button presses. A turbo feature is included because a lot of old games need rapid button inputs. This way you can simulate tons of inhuman presses. An accelerometer will be used to change the screen rotation like a smartphone and a proximity sensor will turn off the device when covered. This last feature can be compared to an Apple Watch being covered or a Nintendo DS when closed. A temp sensor will be used to measure the temperature of the device and a red light on de LED-strip will be shown when the device is too hot. Other statuses like ‘sleep’-mode or ‘turbo-on’ will be shown on the LED-strip. A strip will be used because of its flexibility compared to singular LEDs.

A Bluetooth application with your smartphone will be supplied. Here you can change some settings like the button-layout, the current game, the frequency of the turbo-mode button presses, the screen-rotation sensitivity, the PS4 external controller[3] settings, etc. Because of the device’s portability, your phone will get an application instead of perhaps a laptop because this interaction needs to accessible from anywhere. Also, Bluetooth will be used with the external controller[3] and the smartphone application because you will be independent of a WiFi connection.

Overview

- **Inputs**
  - ○ Buttons*: game controls*
  - ○ Potmeter*: turbo-mode frequency*
  - ○ Proximity Sensor*: on-off switch*
  - ○ Accelerometer*: screen-rotation*
  - ○ Temp sensor*: device temperature*
- **Outputs**
  - ○ Display: *game output*
  - ○ LED light*: turbo frequency*
  - ○ LED-strip*: device statuses*
  - ○ Speaker*: game audio*
- **Processing**
  - ○ Debouncing of button inputs
  - ○ Conversion of potmeter measurements to turbo-frequency
  - ○ Filtering for the accelerometer
  - ○ Threshold for proximity sensor
  - ○ Moving average of temp sensor
- **Connectivity**
  - ○ Bluetooth
- **User interface**
  - ○ Smartphone app

References

[1] Fuji, P. (2019, August 6). *ESPlay-retro-emulation*. Github. https://github.com/pebri86/esplay-retro-emulation

[2] Kelly, M. (2016, October 10). *ESP32-NESEMU*. Github. https://github.com/espressif/esp32-nesemu

[3] *PS4-esp32*. (2019, December 27). Github. https://github.com/aed3/PS4-esp32

[4] Scharfglass, K. (2020, June 10). *Run Your Favorite 8-bit Games On An ESP32*. Hackaday. https://hackaday.com/2020/06/09/run-your-favorite-8-bit-games-on-an-esp32/