ECE 471 Final Project (Sample)
AY3-8910 Chiptune Player

Vince Weaver

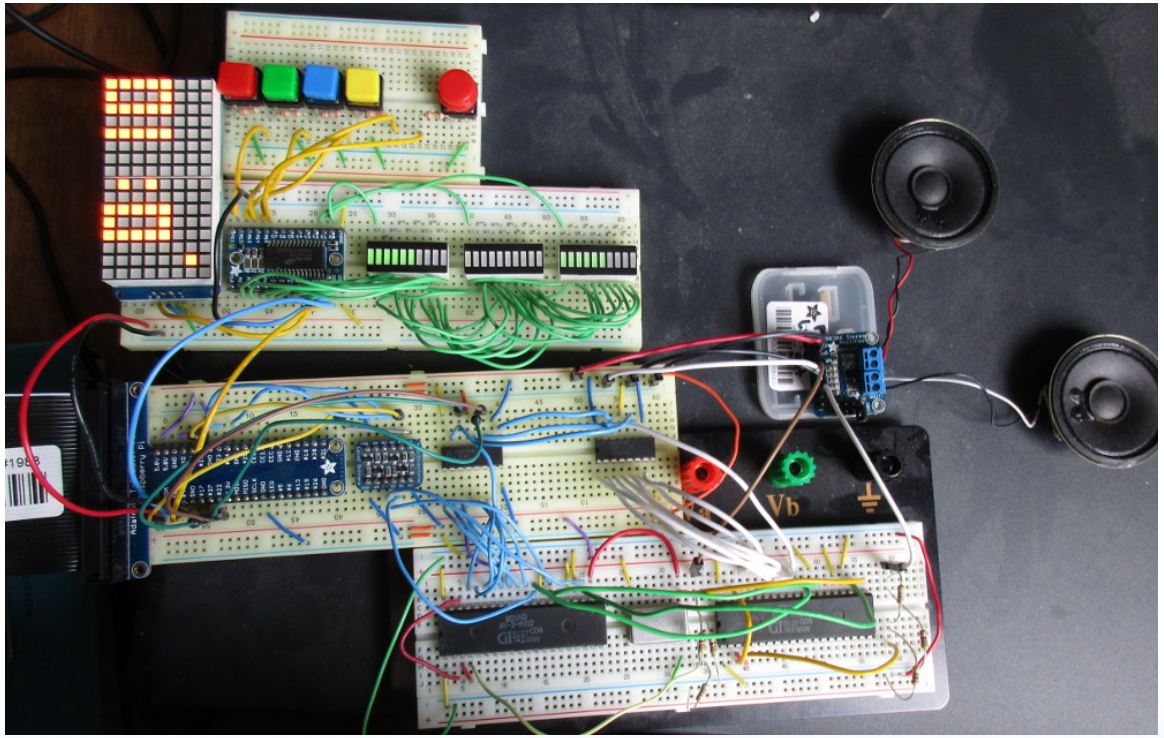
http://www.deater.net/weave/vmwprod/hardware/ay-3-8910/

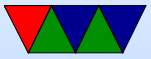
vincent.weaver@maine.edu

1 December 2017

Overview

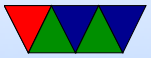
- Raspberry Pi 1980s chiptune music soundcard





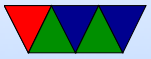
System Setup

- Raspberry Pi2
  Quad Core Cortex-A7, 1GB RAM
- Running Raspbian Linux
- Coded in C because I like C. Quicker to implement than assembly.
- Dual AY3-8910 sound chips. 1MHz-2MHz, 3-channel square wave generation, as well as noise and envelope effects.

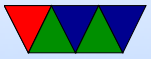


Input

- Sound files
  - stored in Linux filesystem.
  - compressed via LHA, use library to decompress them.
  - Register dumps from Atari/Spectrum machines, 50-60Hz
- Input via keyboard and keypad
  - Keypad uses i2c htk1633 with 8 buttons:
    (play, stop, menu, cancel, next, last, fast-forward, rewind)

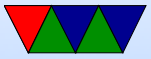


- Miscellaneous
  - 1-wire temperature sensor
  - Real-time clock



Output

- Displays
  - 16x8 LED display, htk1633 adafruit, i2c
  - 6 10 GYR bargraph LEDs. htk1633 breakout, i2c
  - 3 4x14 alphanum LEDs. htk1633 breakout, i2c
- Audio
  - Dual AY3-8910 chips take parallel input
  - 74HC595 serial/parallel shifters, accessed via SPI
  - Audio out to Adafruit MAX98306 Class-D amp
  - Line out jack, with manual switch

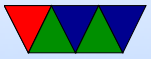


Hardware Concerns

- Power usage – plugs into wall, so no battery related power concerns

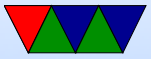
  pi is low power (less than 3W) so overall power not a concern

  To save power could disable the display and shut down amplifier when idle which might save some power
- Code density – not an issue, as the pi has relatively large amounts of RAM and disk.

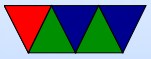


Software Concerns

- Real Time – audio, firm realtime.
  - really need to hit the 50Hz deadline, moreso when stereo.
  - Harder because sending 5 i2c displays and SPI
  - Increasing the SPI bus speed helped
    (each transaction you send 14 registers, each 8 bit address, 8 bit data for both chips)
  - If you miss deadline can hear it in audio

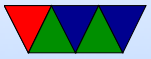


- Security
  - could in theory compromise over internet or via malicious sound files
  - Standard Linux-machine on network security risk
  - Hardware itself not really life critical.



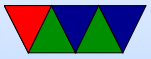
Challenges

- Hitting 50Hz. Use libbcm2835 direct to GPIOs. SPI.
- Decoding the YM5 file format. LHA, and interlaced for better compression. Documentation a bit sparse.
- 5V to 3.3 level converters
- Initially had trouble getting SPI working
- Tried to get auto-sense of line-out jack. Complicated for many reasons (with most jacks you have to detect impedance which I couldn't get working)



Future Work

- Kiosk mode (standalone, buttons only)
- Try it out in my car.



Demo/Questions

- Was a featured project on official Raspberry Pi Foundation Blog
- See also a video:
  `https://www.youtube.com/watch?v=g8rchOi2Evc`