Spherical Persistence of Vision Display

Luis Torres and Noah Moroze

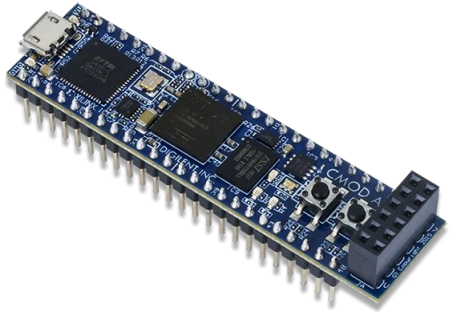
Overview

- We plan to build a persistence-of-vision display
- Persistence-of-vision effect creates the illusion of a coherent image when spinning at high speeds

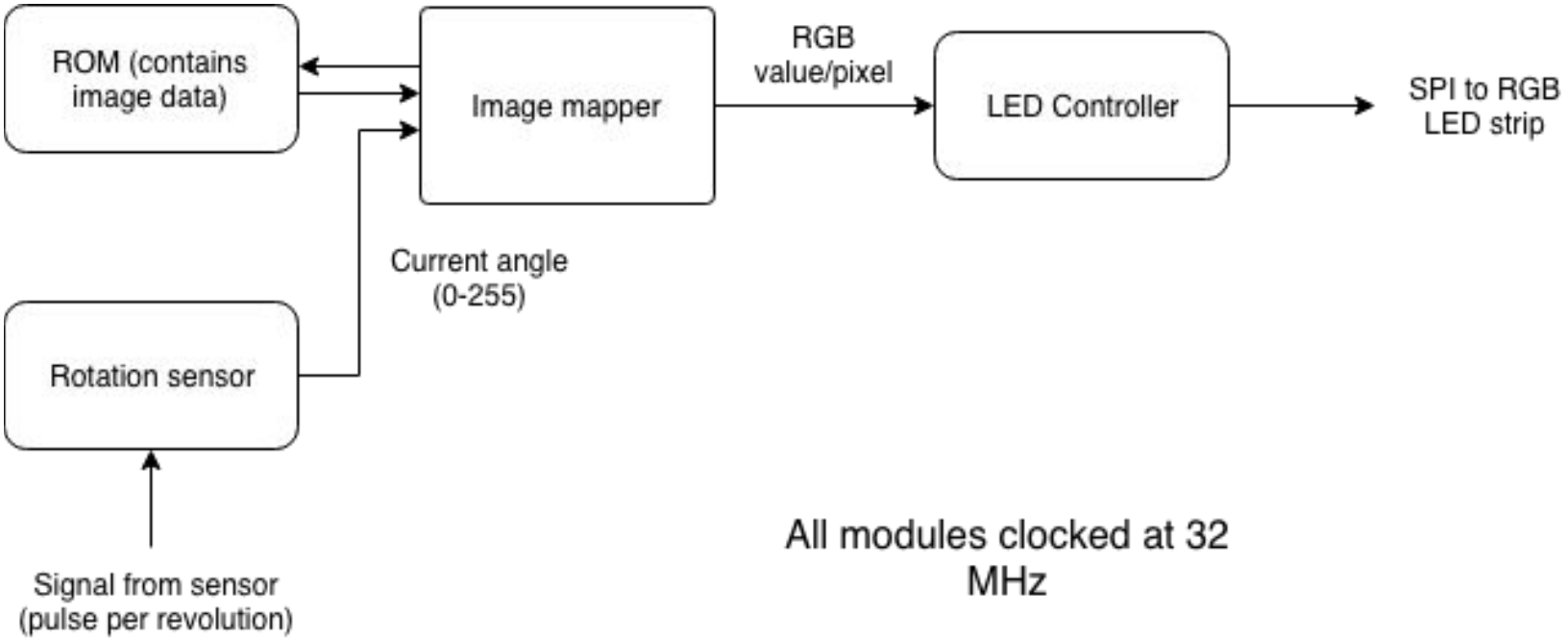
Source: https://youtu.be/Z_ug4ZURRfQ

Hardware Design

- 72 RGB LEDs arranged in a semicircular arc
- Lightweight CMOD A7 FPGA and battery pack mounted on spinning disk directly
- IR break-beam sensor for detecting rotation speed



Block Diagram



LED Controller

- Inputs: `rgb[71:0][23:0], en`
- Outputs: `sck, mosi`

- Using individually addressable RGB LED strip that communicates over SPI
- LED Controller module takes in an array of 72 24-bit color values (one per LED) and continuously outputs them over SPI while `en` is asserted
- `sck` will be clocked at max spec (32 MHz) for maximum refresh rate

Image Mapper

- Inputs: `theta[7:0]`, `rom_data[23:0]`
- Outputs: `rgb[71:0][23:0]`, `rom_addr[14:0]`

- Takes in current angle (`theta`) from rotation sensor, uses this to determine which pixels to display
- Communicates with ROM to get correct pixel values
- Outputs pixel values in `rgb` array
- ROM will be generated as a 72x256x24 block ROM (can reduce width by using a color map if we need to save space)

Rotation Sensor

- Inputs: `rotation_sensor_input`
- Outputs: `theta[7:0]`

- `rotation_sensor_input` consists of synchronized pulses directly from IR sensor
- Module counts time between pulses to calculate an average speed over time
- Integrates this average speed to calculate angular position (resets every time the sensor passes 0)
- Position is outputted and fed into the image mapper

Timeline

November 12 - LED Controller module completed (SPI)

November 19 - Image mapper module completed

November 26 - Hardware assembly and integration.

December 3 - Rotation timing module completed

December 10 - Full integration + debugging

Challenges

- Hardware
  - Balancing spinning structure
  - Power - Have to run LEDs at half brightness, gives us about 2.5 hours runtime
- Digital Design
  - Communicating with memory - needs to be fast enough to read all pixel values and then output before we need to display the next slice (can potentially correct this by shifting out each pixel value one-at-a-time instead of all 72 at once)
  - Accurate rotation timing

Stretch Goals

- Animation / Video
- Multiple LED strips

Questions?