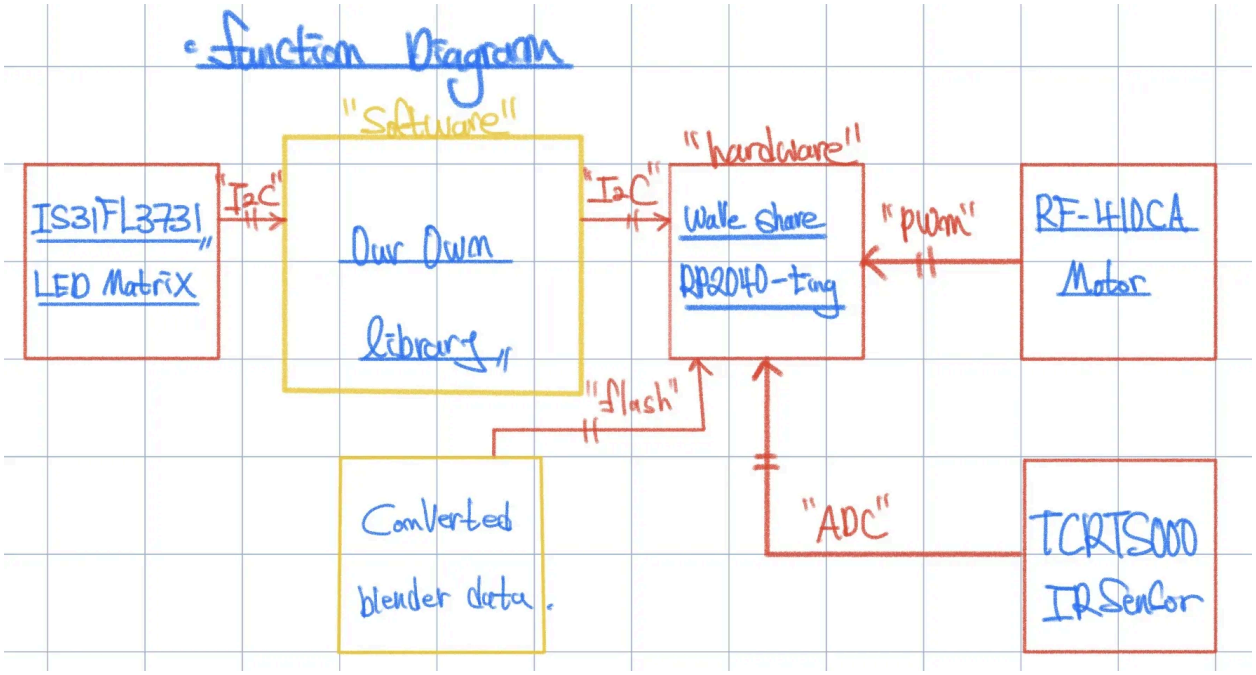
5780 Proposal: Volumetric Display

Team Members:
Semrah Odobasic: semrahodobasic@icloud.com
Jin Jeong: u1375534@umail.utah.edu

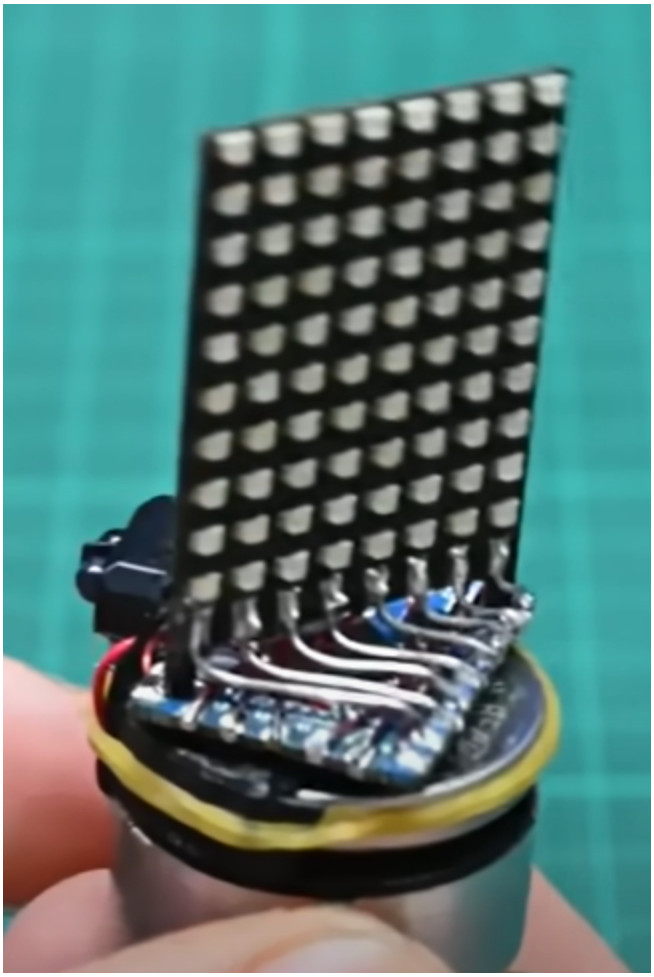
Block Diagram:



High Level Description:
We are looking to create a 3 dimensional volumetric display. Our approach involves using a 2 dimensional LED matrix and spinning it. As the matrix spins it will display different slices of a 3 dimensional image. These slices are generated using Blender and can be generated with any models. Once all of the slices are displayed there will be a volumetric 3 dimensional image which you will be able to see.

Background/Motivation:
Our mini project aims to expand the field of technology by crafting a volumetric display using a LED Matrix. By merging art and technology, we're driven to transform conceptual ideas into reality,from our passion for innovation and the knowledge that we learned throughout the class such as microcontroller programming, motor control, and hardware-software integration. Our project features a spinning LED Matrix that showcases slices of a 3D model, kind of like a "Hologram". This project holds the potential to revolutionize the visual display system. Through working hardware connected with an LED matrix, high-performance motor, and microcontroller driving both components,we aim to deliver visually captivating demonstrations of our project.

Example of final assembly:
We plan to spin the whole thing so that we don't require a slip ring. This will also help with balance as we will put the heaviest parts on the bottom.



Parts List:
Waveshare RP2040 Tiny (Approved by professor if used with C library)
Adafruit IS31FL3731: Matrix Controller
Adafruit Dot Matrix Module: Charlieplexed LED Matrix
Vishay TCRT5000: IR Sensor
RF-410CA: Motor
ADAFRUIT DRV8833: Motor Driver

Milestones:
- Milestone 1: LED Matrix Library
    - The LED matrix only has a library in C++. Our goal for this milestone will be to convert the C++ Library into our own C library so that we can use it in C.
        - Convert the C++ adafruit library into our own C library so we can communicate to the matrix. Matrix I2C Datasheet

        - Display a basic image to ensure that we can display images onto the matrix.
- Milestone 2: Motor Control
    - The goal of this milestone will be to set up the motor driver and the motor.
        - Set up GPIOs for PWM mode
        - Using two channels demonstrate we can go one direction, stop, and go the other direction.
- Milestone 3: Determine RPM using the IR sensor.
    - The goal of this milestone will be to use the IR sensor to detect the RPM we are going at.
        - Set up the IR sensor for reading an input. Using this input we will calculate the RPM that the whole device is spinning at.
        - Using the IR sensor RPM, control the speed of the motor to achieve a desired RPM.
- Milestone 4: Combine everything into a volumetric display
    - For the final milestone we plan on combining everything into a volumetric display.
        - Solder everything together. (Prototype before this will be to use breakout pins and connect everything on a breadboard to ensure it all works.
        - Combine all of the systems together and ensure that all of them work individually running on the same C file.
        - Program the display to change frames depending on the RPM of the device so that the image stays in place.

Risks and Risk Mitigation:

- Milestone 1:
    - This is a medium risk milestone. There is a risk that we can't get the library converted. If this becomes an issue we will try to cross compile the C++ library with our C program. If this also doesn't work we will attempt to directly drive the matrix using all of the GPIO's on the RP2040. (We confirmed there are enough GPIO pins to do this)
- Milestone 2:
    - This is a low risk milestone. We have used this motor controller on another project so it shouldn't be hard to get everything working.
- Milestone 3:
    - This is a high risk milestone. The risk comes from being able to identify an RPM from the IR sensor. We use this so when you put your hand by it, the motor will start spinning. If we can't figure this out we will switch to a motor with an encoder in it. As this is complicated, if we need more time milestone 4 is relatively simple so that we can push this back while we identify a solution.
- Milestone 4:
    - This is a low risk milestone. There is a chance that during the process of combining the components there is something that stops working but it should be fixable if everything works separately.

3 Lab Implementations:

- GPIO:
    - The IR Sensor will be connected to one of the GPIO pins configured as an input. Once the IR sensor detects the RPM, it will send a signal to the microcontroller through the GPIO pins

- I2C:
    - LED Matrix will be controlled using the I2C protocol, which will allow the communications between the LED matrix and the microcontroller.

    - It will send commands and data to the LED matrix through the I2C bus and the setup will reduce the number of GPIO pins to control the LED matrix, making it more efficient and easier to use.

- PWM:
    - A PWM signal will be used to control the speed of the motor.

    - The microcontroller will adjust the PWM signal based on input from the IR sensor, maintaining the rotation speed of the motor for the correct animations and display of the volumetric image.