



UNIVERSITY OF
ILLINOIS
URBANA - CHAMPAIGN

Piano Visualizer

Electrical & Computer Engineering

Nuwan Singhal, Sarayu Suresh, Jay Park

5/4/26

- Beginners struggle to translate sheet music into keys, timing, and duration.
- Mistakes are hard to self-diagnose in real time.
- Existing apps/light-up keyboards can be expensive, internet-dependent, or tied to specific hardware.

Main goal: Create a standalone, low-latency piano learning system that reads MIDI files, displays notes on an LED matrix, and gives feedback on correctness/timing.

High Level Requirements

- Read, parse and display midi files at ≥ 60 Hz display refresh (no flicker)
- Support up to 10 simultaneous notes with ≤ 30 ms latency for MIDI input
- User interface to select songs of their choice through a microSD card

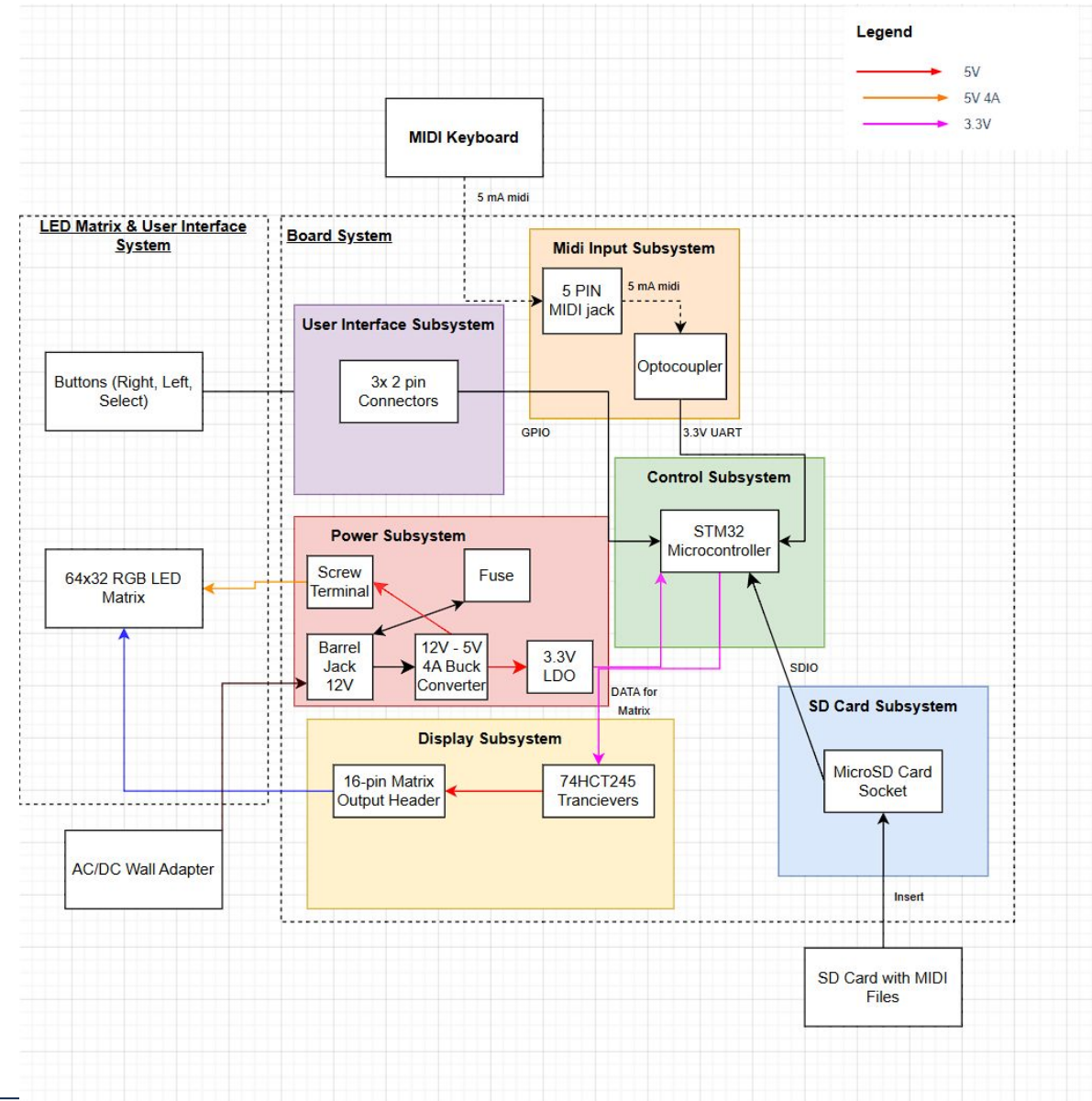
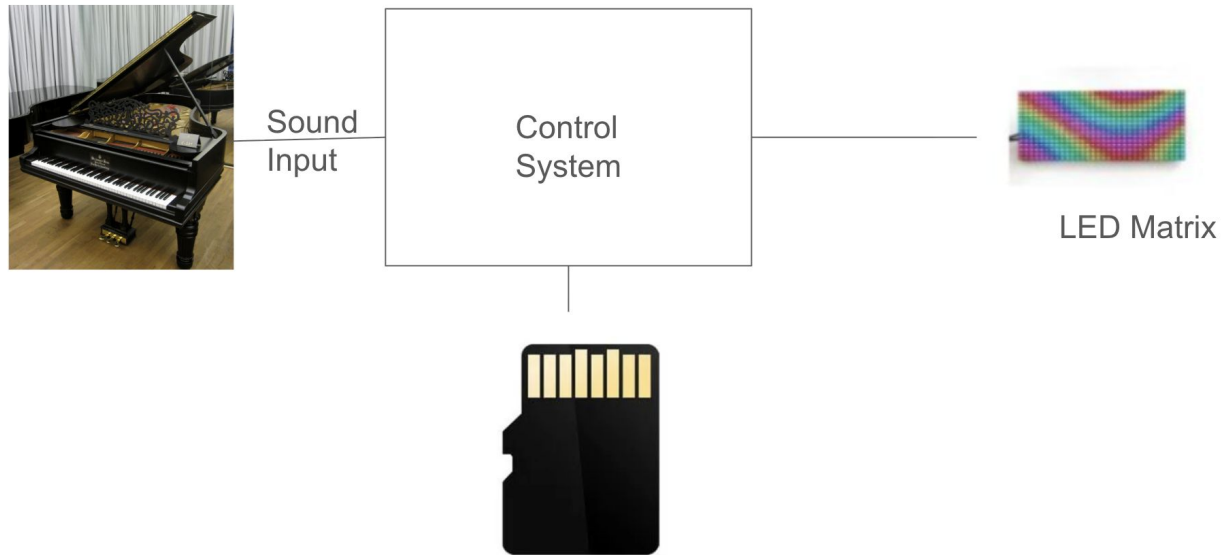
Original Design

- Microphone input as a secondary option
- Specialized 5V wall adapter
- DMA to control LED board

Major Changes

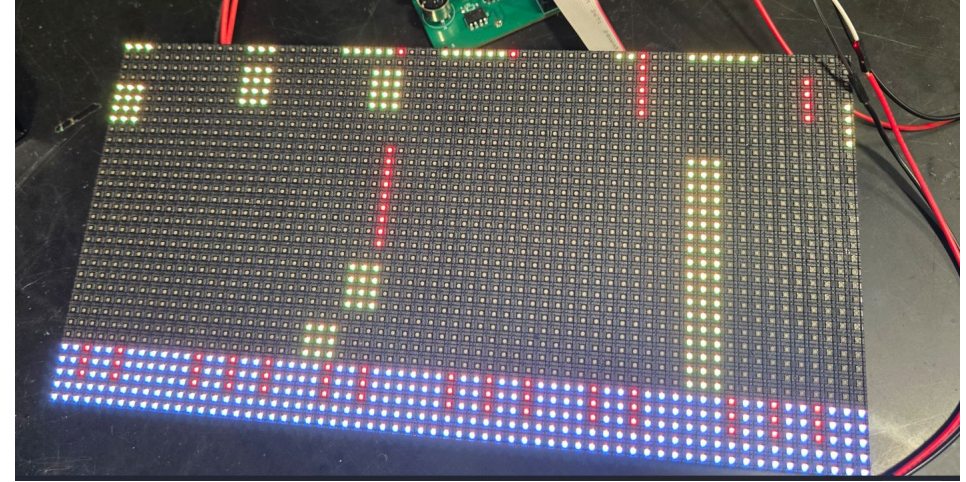
- Removed microphone subsystem to focus on reliable MIDI-based note detection
- Switched to standard 12V adapter + buck converter
- Bit-banging with a timer

High Level System Overview



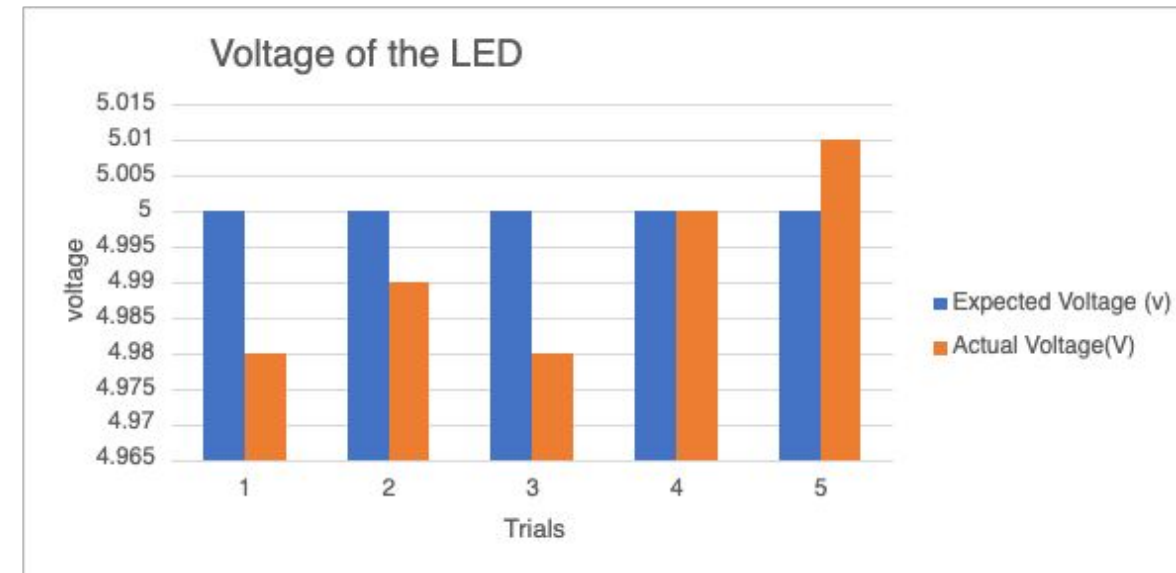
Requirements

- LEDs operate at 5V at 60Hz
- Display at least four octave piano keys and represent note durations in real time
- Uses level shifting from 3.3v to 5v with minimal ghosting



Results

- 62.5Hz refresh rate
- No apparent ghosting or flickering
- No leading or lagging of note display



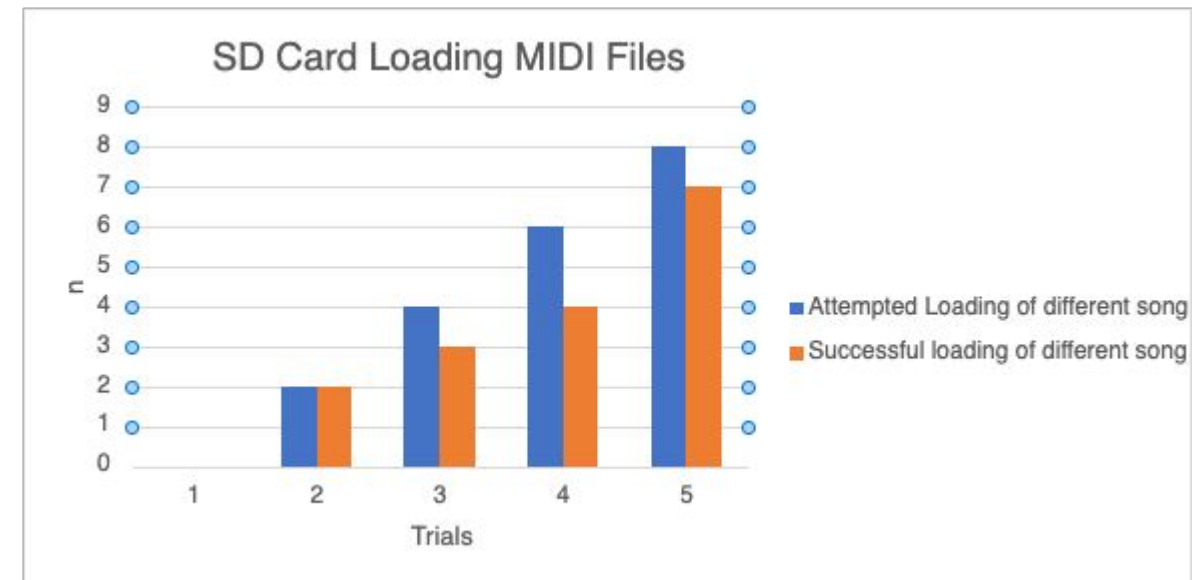
Requirements

- Successfully read MIDI files from a micro SD with SDIO
- Allows users to select multiple songs
- Able to work repeatedly without data corruption



Results

- 1-bit SDIO reading (12.5 MB/s)
- Multiple file selection
- 1 file failed

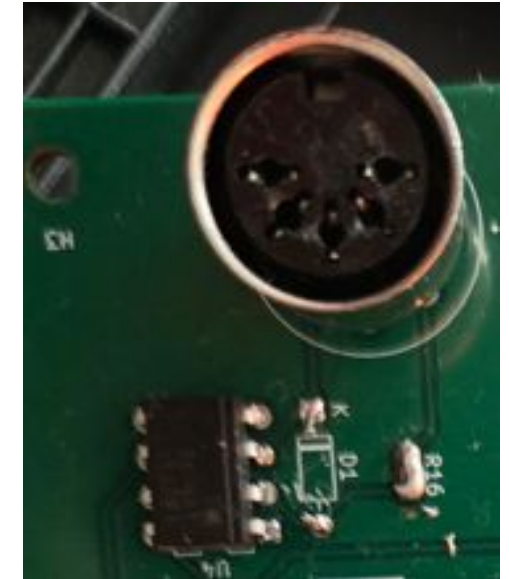


Requirements

- Able to accept 5 pin MIDI input at 31250 baud
- Able to detect note on and off with in 30ms latency
- System can detect up to 10 notes at the same time

Results

- Midi data pulsing from adapter to optocoupler
- Optocoupler failure, so data could not reach the MCU
- Not functional

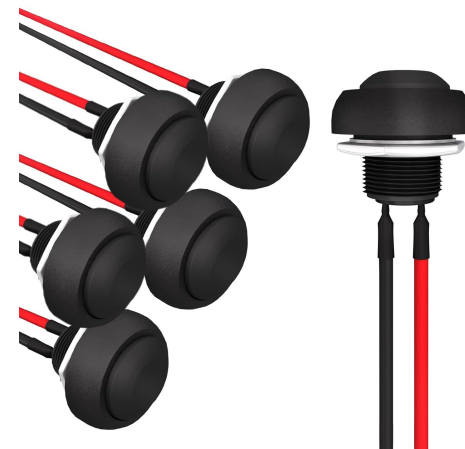
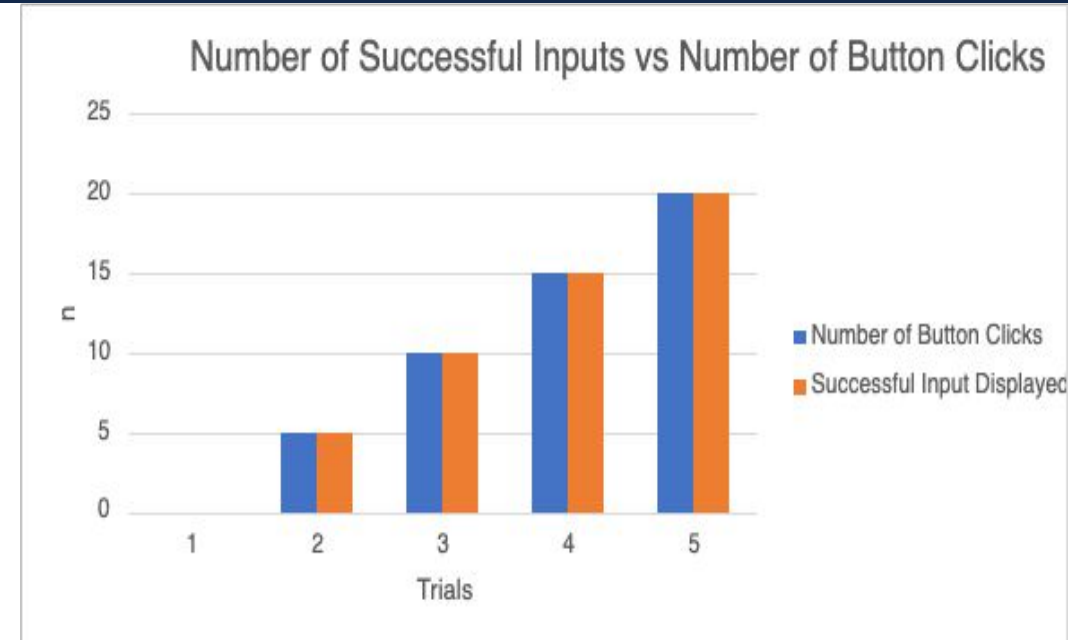


Requirements

- Able to detect physical button press and update system state without false triggering
- Allow users to select multiple MIDI files from the SD card
- Able to change between different modes
- Provide visual feedback of menu navigation and system status on LED matrix

Results

- System detects button and allows user to navigate
- Different songs can be selected based on user's choice
- Close to no delay in switching options or starting playback



Requirements

- Able to supply 5V and 4A to the LED matrix board
- Accept GPIO from the user interface system
- Include protection against overcurrent

Results

- LED board voltage of 5V supplied and functions
- MCU, SD card and LED Control subsystems powered sufficiently
- No overheating of components

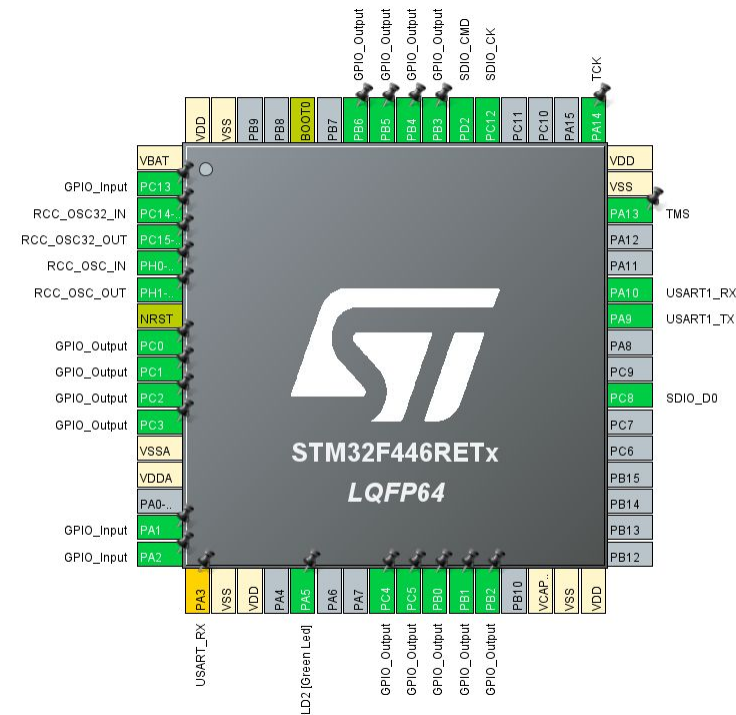


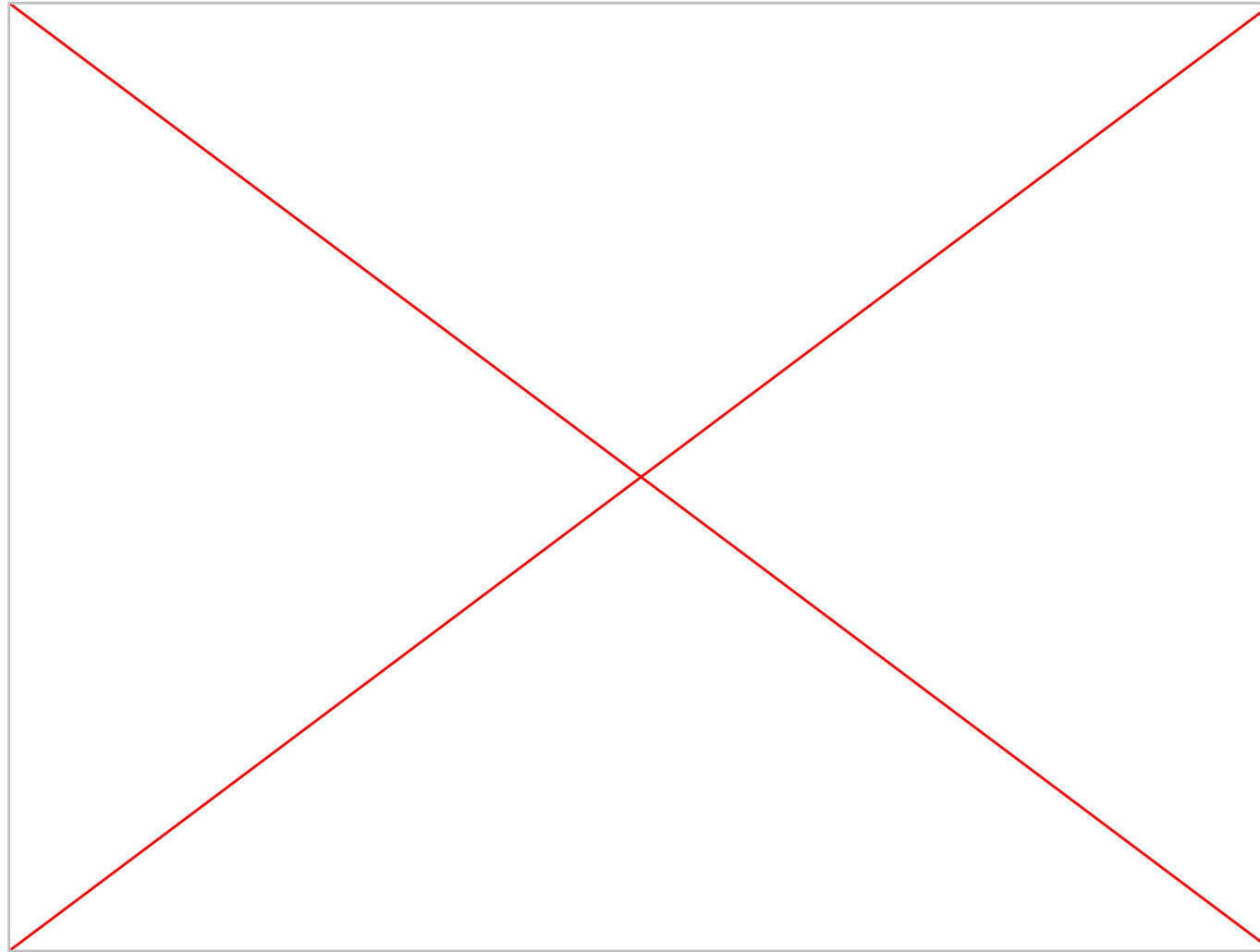
Requirements

- Able to send correct data for the matrix
- Receive and read MIDI files from SD card
- Take user input and prioritise returning to main menu

Results

- Sufficient CPU time to deal with sd card reading, led board output and user input
- Data Lines correctly working for SD card and LED board
- Prioritises User input over background tasks

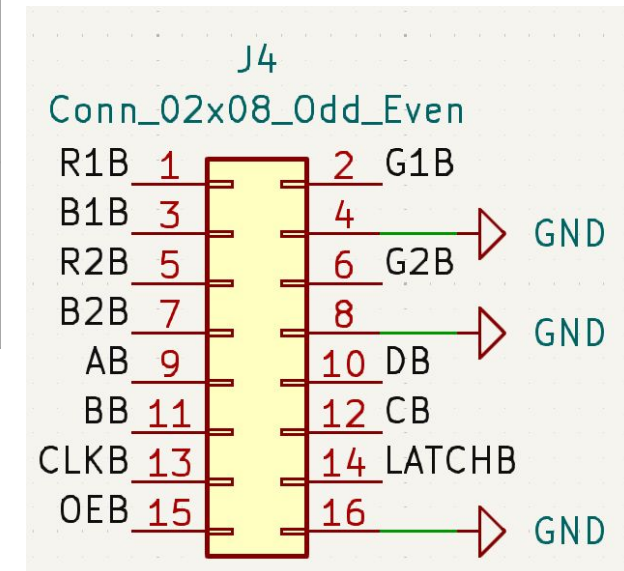
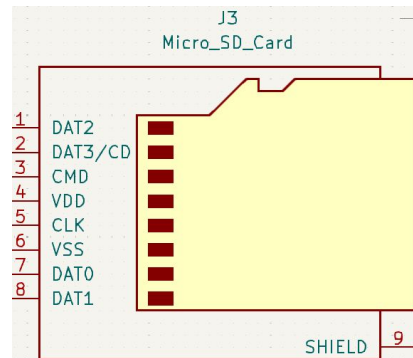
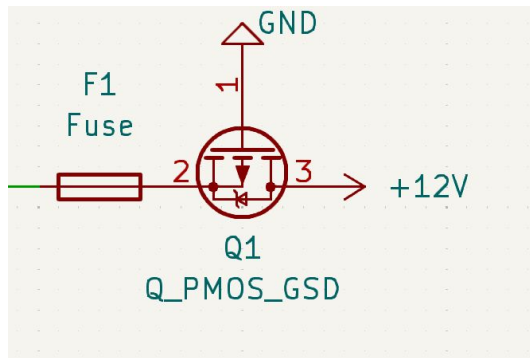




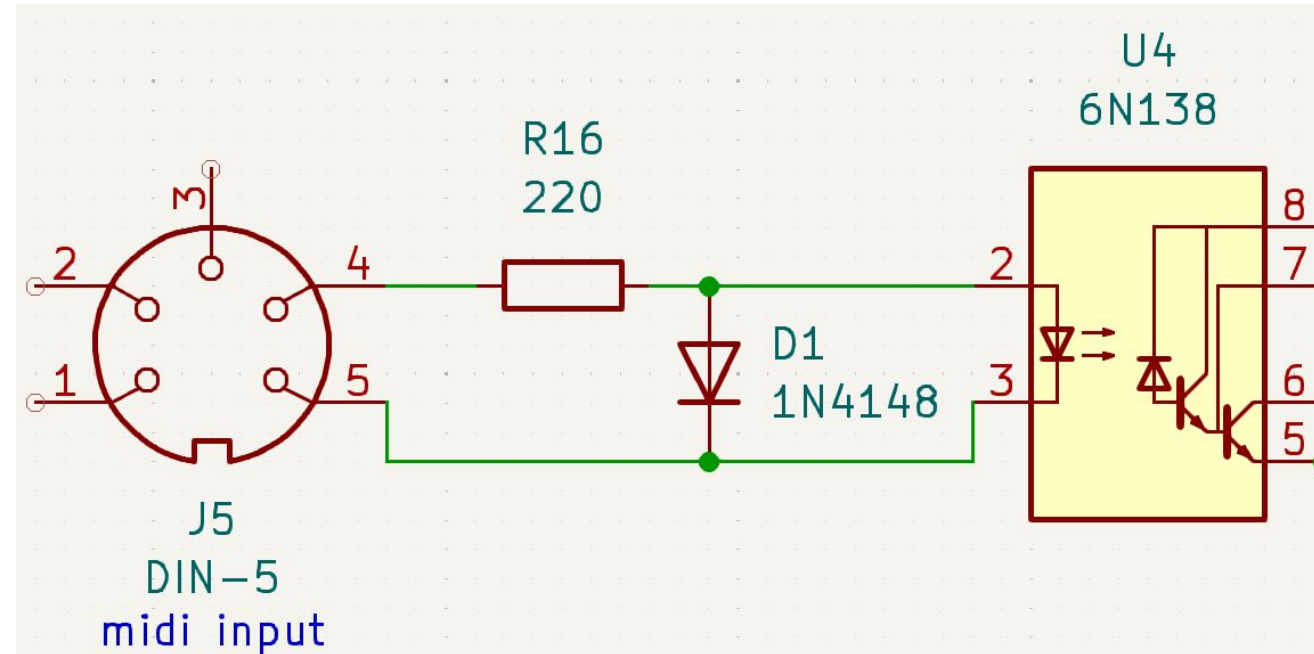
- STM32F4 fully functional
- LED Display with no flickering
- Micro SD card storage
- Efficient reading and parsing of midi files from SD Card
- Low latency user input and very low load times
- Clear UI and matched real song timing

Challenge	Solution and Result
Failed Mosfet, incorrect diode and capacitor polarity	Shorted, or replaced to achieve functionality
SDIO not working	Used 1 data line instead of four, likely a soldering issue
Incorrect HUB75 Wiring	Switched B,C and D lines in software

RED0	1	16	GREEN0
BLUE0	2	15	GND
RED1	3	14	GREEN1
BLUE1	4	13	NC
A	5	12	B
C	6	11	D
CLK	7	10	LATCH
/OE	8	9	GND



- MIDI Input
- Optocoupler needs 1.1-1.7V to drive LED
- Initially read 0.5V
- Shorted Resistor → 0.8V
- Shorted Diode → -1.5V
- Adapter issue
- Wiring issue



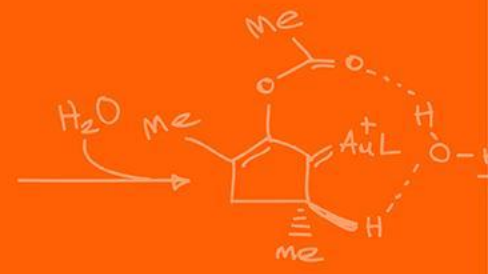
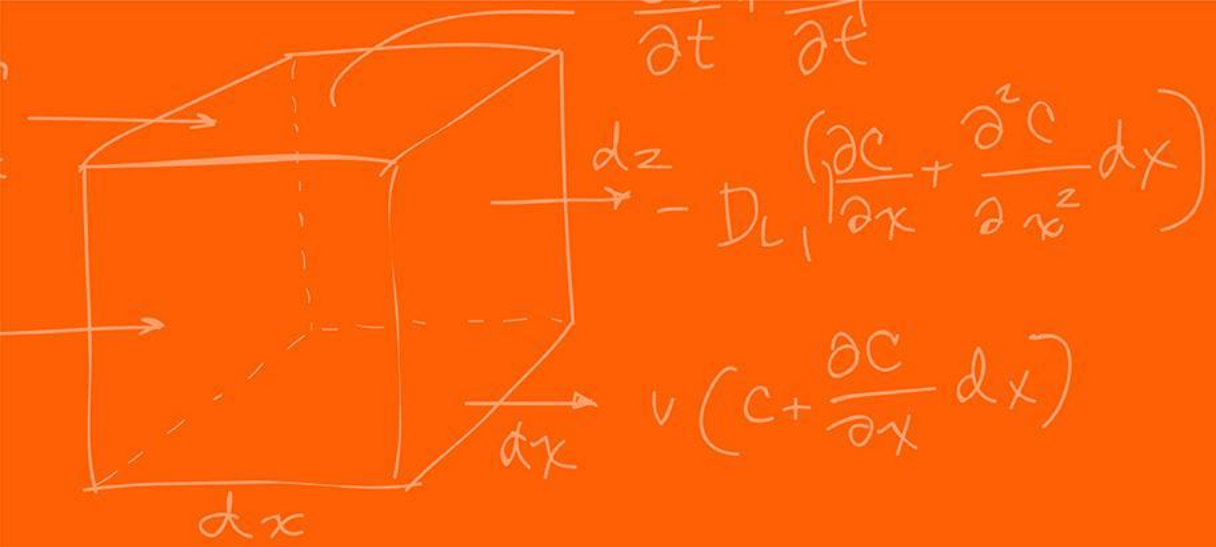
What we learned

- PCB Design and debugging
- Soldering small components
- STM32 Development
- Modular Design and Collaboration

What we would do differently

- Simulation and Testing
- More detailed logging
- Greater attention to footprints
- Easier debugging methods

- Better enclosure and more professional presentation of product
- Support for other instruments
- Additional software features: playback speed, sheet music mode, save points, one handed mode



Thank You

Questions?

