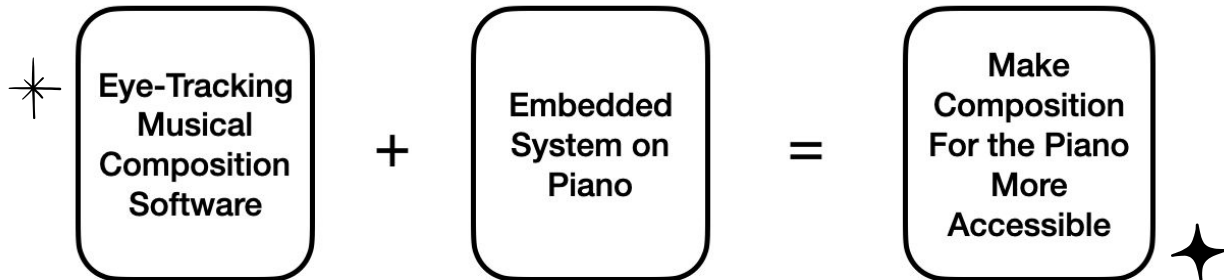


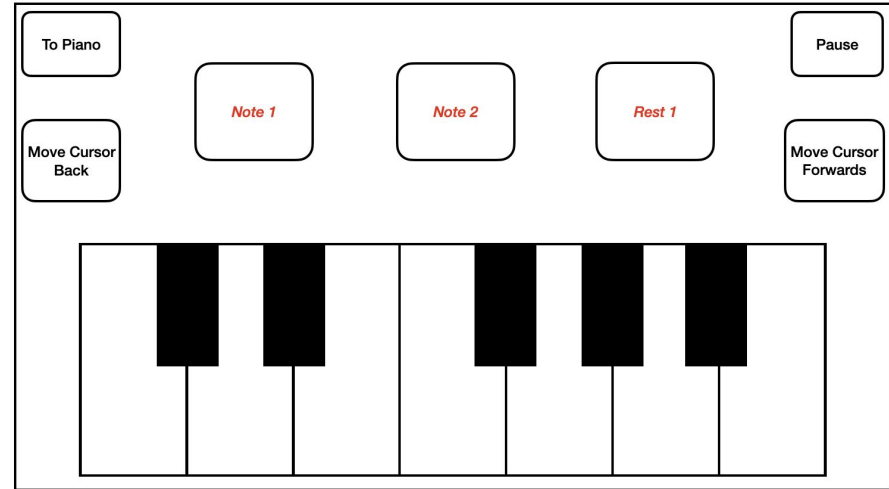
C5 - The Sound of Sight

Fiona Fisher, Shravya Sai Koushik,
Peter Ragone



Software Requirements

- Accuracy:
 - 75% for eye-commands
 - 95% for screen-section
- Latency:
 - 500 ms for eye-commands
 - 100 ms update for secondary UI
- Coverage:
 - One octave
 - Two note lengths/one rest length
 - Moving the cursor forward/back
- Accessibility:
 - Only eye-commands



Hardware Requirements

Use Case Requirement: Playing notes of composed piece

Technical Design Requirements:

- Power regulator outputs
 - 12V, 5A
 - 3.3V
- Solenoid power consumption
 - Each solenoid 12V, 250mA
 - If we implement chord functionality for up to 3 simultaneous notes, the **maximum** power the solenoids should consume at any point in time is $12V \times 0.25A \times 3 = 9 \text{ Watts}$
- Solenoid duty cycle
 - 100%



Why?

Health, Safety & Welfare:

- Therapeutic Hobby

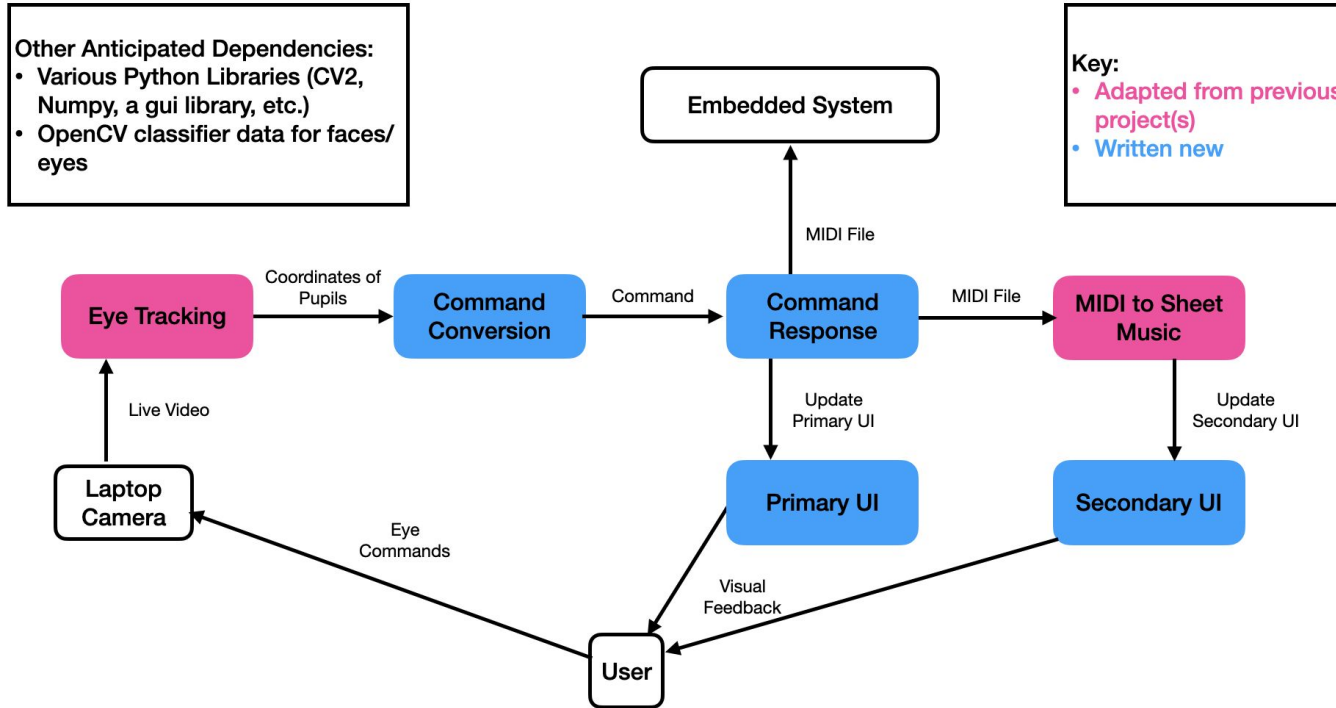
Social Factors:

- Creates an inclusive way to compose music

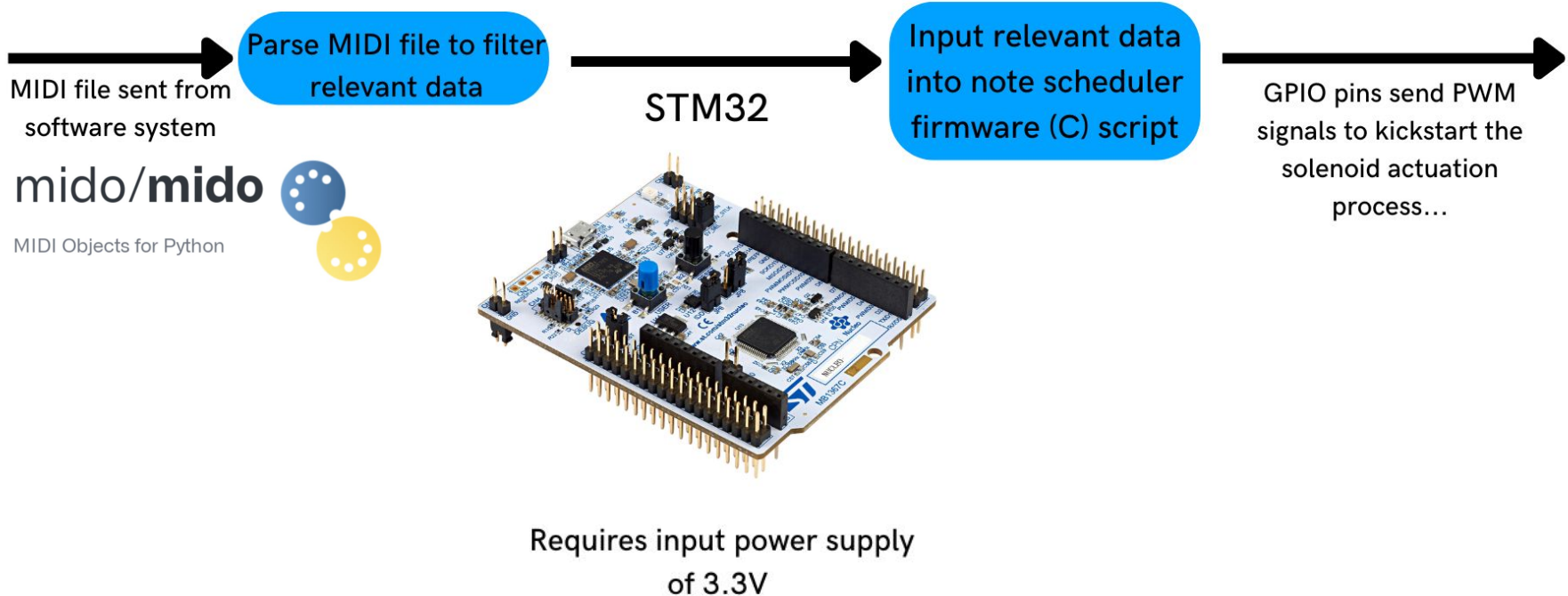
Economic Factors:

- 3D printed housing
- Product's software utilizes computer hardware

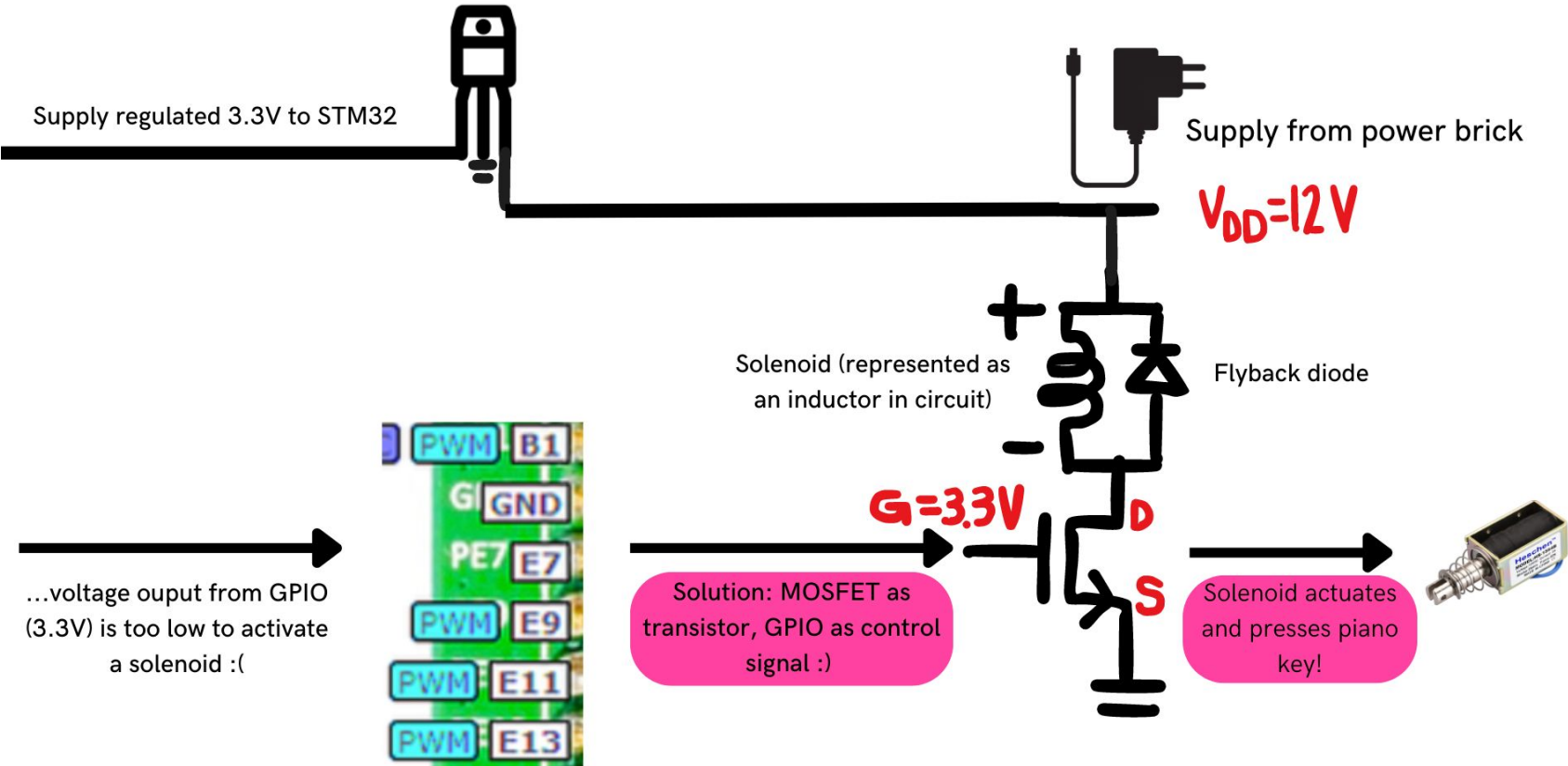
Block Diagram: Software System



Block Diagram: Processing Path

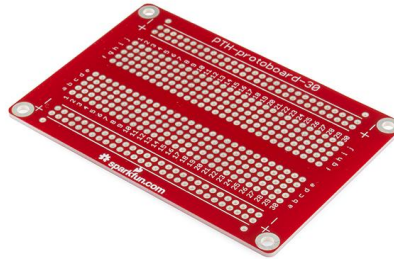
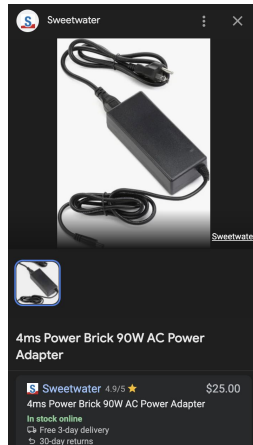
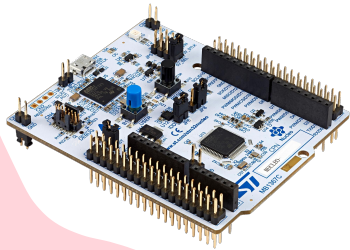


Block Diagram: Hardware Execution



Implementation Plan

- STM32 (already own)
- Power brick (purchase)
- Solderable breadboard (already own)
- Push-pull solenoids (purchase, currently experimenting with Adafruit 412)
- Linear voltage regulator (purchase)
- Simple 3D-printed case to hold system on piano (will design)



Formal Testing

- Standardized tests for **accuracy** and **latency**.
 - Software
 - Test with multiple users.
 - Test with different patterns of eye-commands.
 - Video-record them performing the patterns for playback.
 - Hardware
 - Design (manually) various MIDI input for the solenoid system.
 - Can also test power consumption here.

Informal Testing

- More informal spot-checks throughout semester.
 - Specifically, for eye-tracking.
- To recognize and fix bugs earlier.

Project Management: Upcoming Weeks

Week	Fiona	Shravya	Peter
6	Code frontend & identify command coordinates	Test power management design Create MIDI conversion firmware	Test solenoid control design Implement eye-tracking for target sections
7	Create code to map coordinates to commands	Implement UART to transmit data to STM32	Implement individual use calibration for eye-tracking.
8	Code MIDI command responses Integrate MIDI to sheet music code w/ system	Integrate controllers & solenoids	Design/3D print solenoid case Implement eye-tracking for commands
9	Code UI command responses	Integrate primary software & controller	

