Touch TrackIR - Design Review

Team A6 - Matthew Kuczynski, Matthew Shen, Darwin Torres

Use Case & Use-Case Requirements

- Our product is intended to be used for non-touch-compatible laptops of a specific size to make them touch compatible.
- Would like to expand our design to features such as 2-finger scroll and zoom and add programmable 3-finger gestures.
- **Touch Precision**: < 0.3 inch
- **False Positive Rate** (Screen not touched, but touch detected): 1 per 5 minutes
- **False Negative Rate** (Screen touched, but touch not detected): 5%
- **Response Time**: < 150 ms
- **Refresh Rate**: > 15 Hz.
  - Should be able to achieve better, but this is a usable minimum refresh rate.
- **Weight of frame**: 1/2 lb.
  - Laptop can withstand at least this much when applied directly to upper edge.

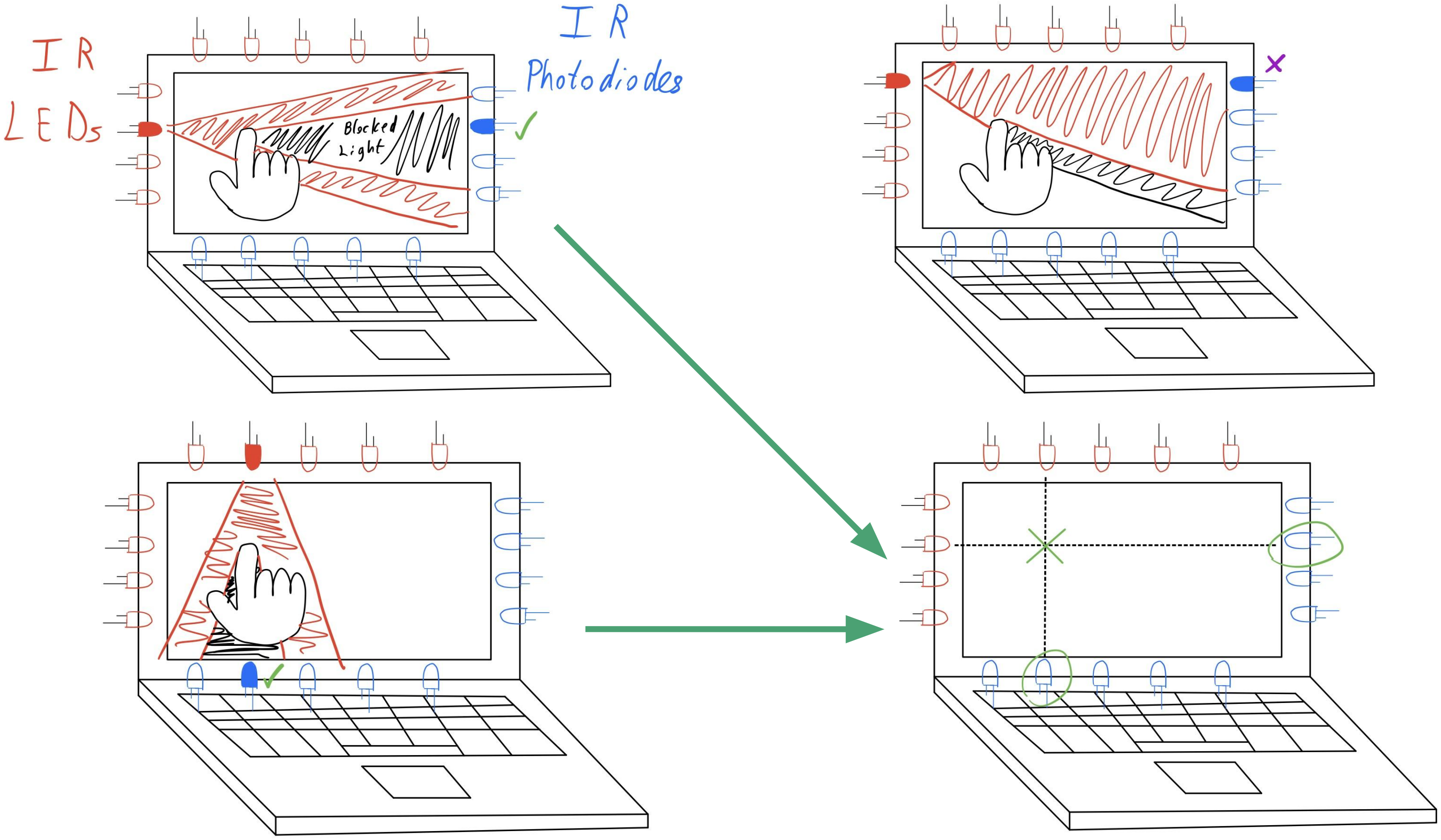
Solution Approach - Use-Case Requirements

- Touch Precision
  - We plan to use a 2D array of IR LEDs and Photodiodes to determine finger position, powered by Arduino Mega.
  - The circuit is designed to have the centers of the LEDs 5.5mm apart, so we should be able to achieve precision of up to 2.75mm, which is far less than 0.3 inches
- False Positive Rate
  - Ensure the photodiode sense circuits are designed to be responsive enough to the LEDs.
- False Negative Rate
  - Ensure the selected components enable the voltage across the photodiodes to rise and fall comfortably in between cycles.
- Response Time
  - As explained before, the response time will be dependent on the quality of our software interface and implementation.
  - First, we will route the photodiode readings to digital GPIO pins on an Arduino Mega (or similar), then pass data via serial USB to Python backend
- Refresh Rate
  - Similar to response time, refresh rate depends on the ability of the software to poll and update quickly enough.
- Weight
  - Wooden frame will be light and thin, and should not account for more than 0.5 lbs
  - PCBs will also not affect weight drastically

Solution Approach - Design Requirements

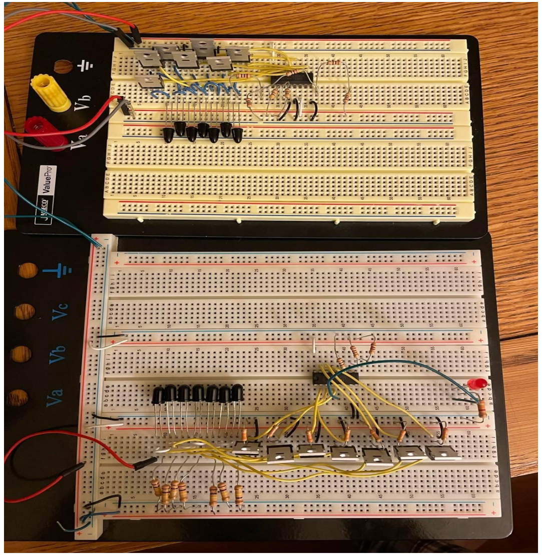
- Stationary Frame
  - Hardware secured by PCB in a frame around all 4 edges.
  - 30 x 56 LED/Photodiode Arrays along short and long edges of frame.
- Power Source
  - Our design is now simply powered by the Arduino 5V pin.
- Software Processing
  - Uploaded Arduino code will toggle LEDs.
  - Python-Arduino will read photodiode values and determine finger position
  - Screen control interface via Python.

Solution Approach

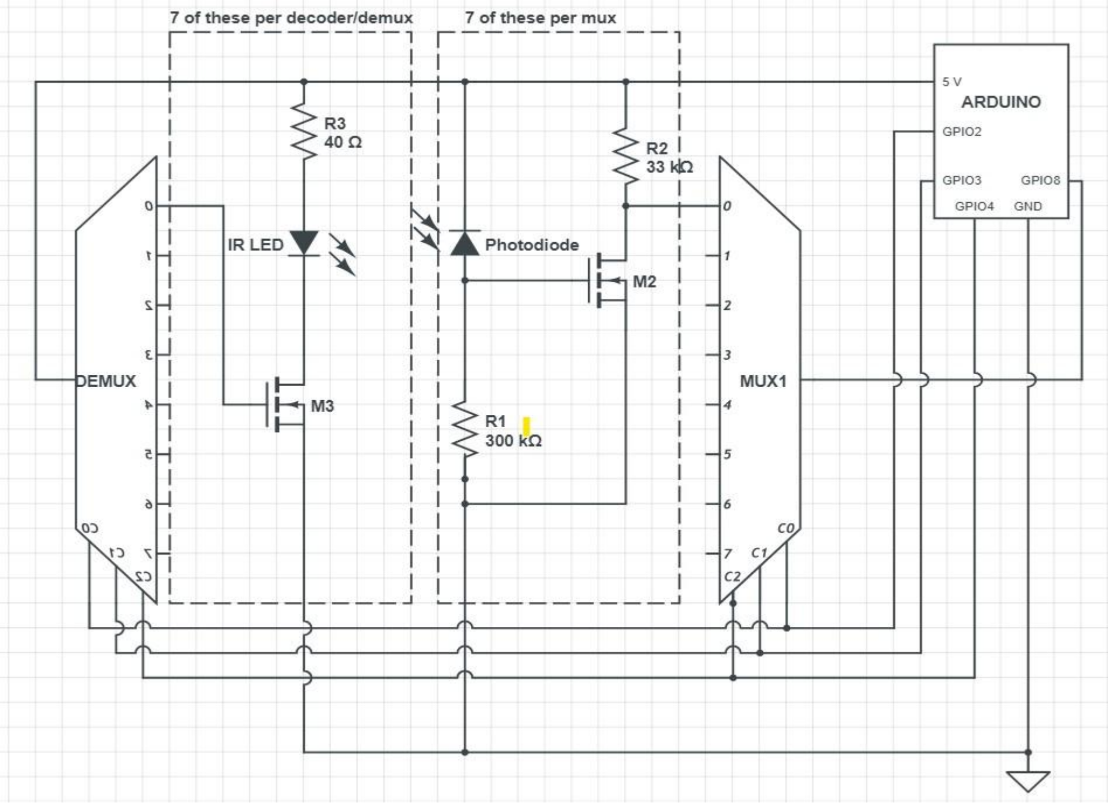


Schematic of a hardware “segment”

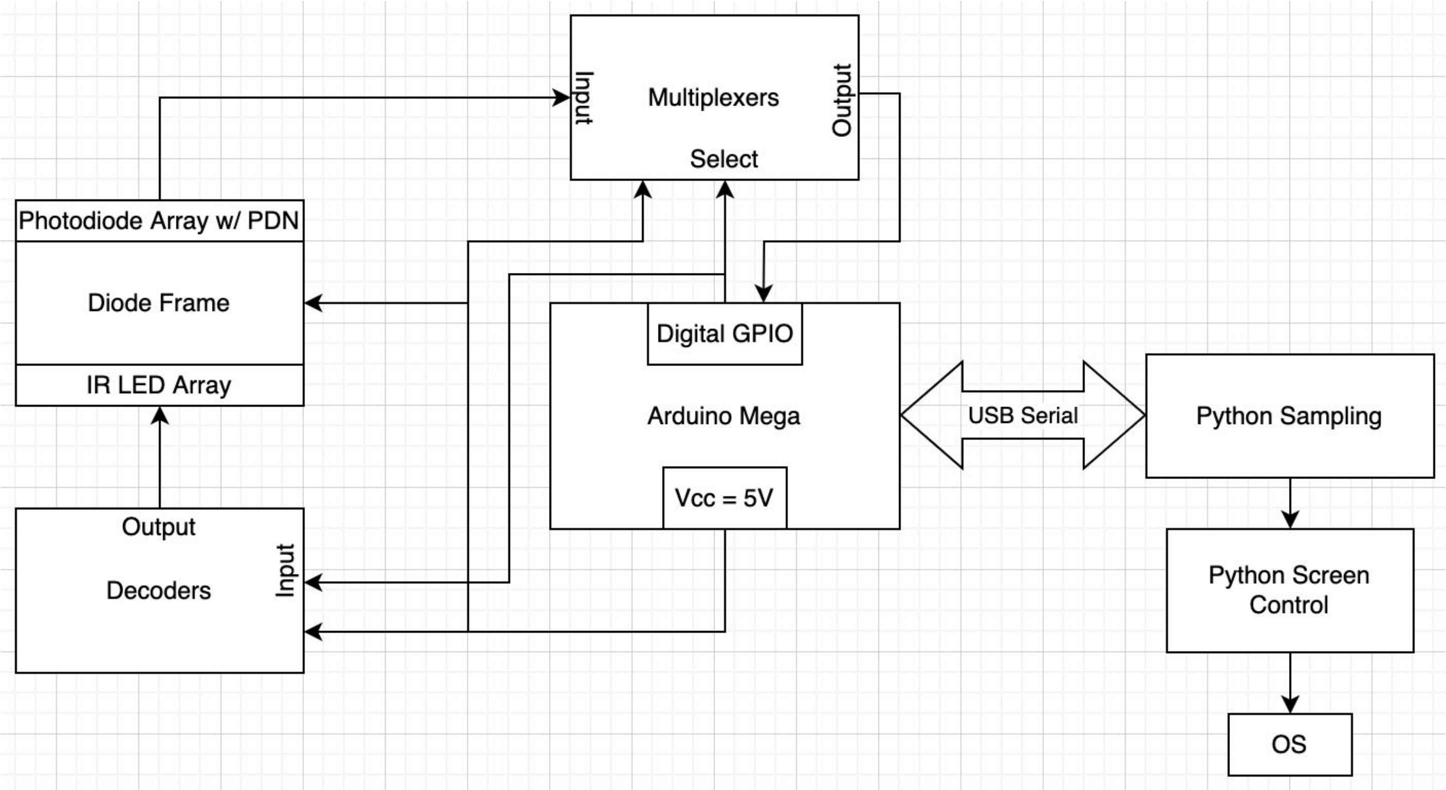
- Only need 1 demux per edge with LEDs (2 total).
- Will need 8 Muxes for long edge and 4 for the short edge.



Through-hole components shown, but SMT will be used on PCB (except for diodes).



System Specification – Block Diagram



System Specification – Mission of Components

- Arduino Mega
  - Provide voltage source, allow hardware select via GPIO, send photodiode reads to Python via asynchronous serial communication over USB.
- Decoders
  - Choose which LEDs to turn out based on digital GPIO output
- Multiplexers
  - Choose which photodiodes to read based on digital GPIO output
- Diode Frame
  - Contains and secures hardware that is used to create 2D array with full screen coverage
- Python Sampling
  - Sample finger positions from Arduino using Serial library.
- Python Screen Control
  - Communicate with OS via Win32 API and perform click/drag/scroll/zoom at desired location

Implementation Plan

- Copy/Download
  - Import serial library to communicate with Arduino from Python
  - Win32 API
- Buy
  - IR LEDs, Photodiodes, Arduino Mega, PCB
- Assemble
  - Diode Frame, LED/Photodiode array
- Design & Develop
  - PCB
  - Python scripts for data collection and screen control

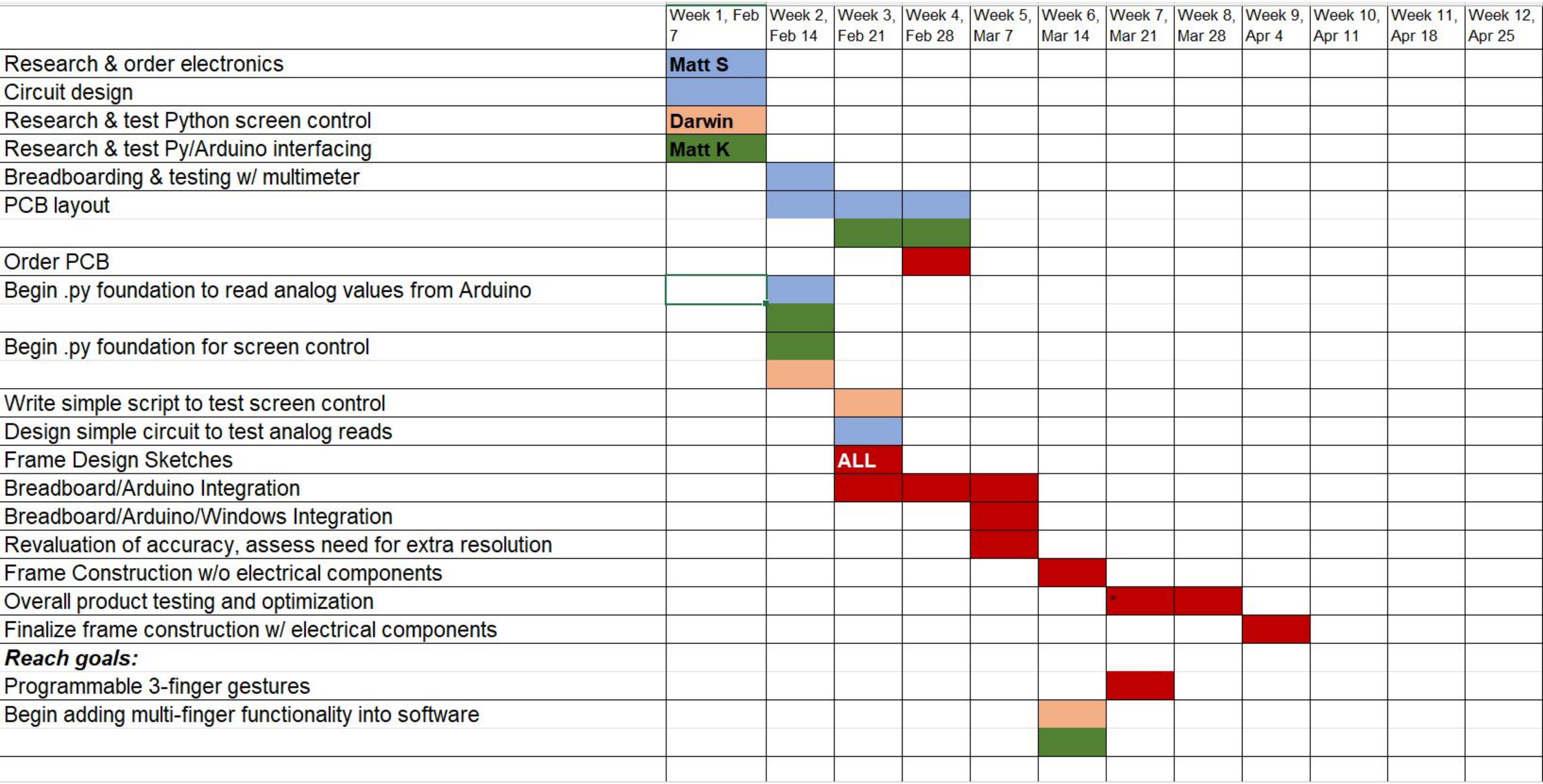
Testing & Verification - Use-Case Requirements

Requirement	Quantitative Goal	Testing Method	Risk Mitigation
Touch Precision	< .3 in	Measure distance between center of finger and the actual click.	Can try reading from multiple photodiodes per single LED to improve accuracy .
False Positive Rate	< 1 per 5 mins	Let frame remain attached with screen untouched for one hour with script running, and count touches detected.	Use software to filter out touches that do not span across multiple photodiodes since normal finger will cover multiple.
False Negative Rate	< 5%	Repeatedly touch screen 100 times, and count how many requests were sent.	Increase refresh rate, improve software speed.
Response Time	< 150 ms	Calculate time using frame-by-frame video analysis of touch and screen.	Improve software quality and speed
Refresh Rate	> 15 Hz	Analyze slow-mo video to determine cursor update time (temporarily disabling smoothing algorithm).	Change from Python to C++.
Weight	< ½ lb	Measure on scale, and ensure screen stays open with frame attached.	Change frame material from wood to plastic.

Testing & Verification - Design Requirements

Requirement	Quantitative Goal	Testing Method	Risk Mitigation
Stationary Frame	< 0.1% error change	Shake frame for 15 seconds and recalculate false positive and false negative rate.	Add mechanism to tighten frame to laptop screen.
Power Source	5V	Verify with multimeter.	Have backup Arduino.
Software Processing	Meet goals for response time and refresh rate	Measure response time and refresh rate as explained before.	Directly use C++ for faster code and fewer levels of abstraction.

Schedule



	Week 1, Feb 7	Week 2, Feb 14	Week 3, Feb 21	Week 4, Feb 28	Week 5, Mar 7	Week 6, Mar 14	Week 7, Mar 21	Week 8, Mar 28	Week 9, Apr 4	Week 10, Apr 11	Week 11, Apr 18	Week 12, Apr 25
Research & order electronics	🟦 **Matt S**											
Circuit design	🟦											
Research & test Python screen control	🟧 **Darwin**											
Research & test Py/Arduino interfacing	🟩 **Matt K**											
Breadboarding & testing w/ multimeter		🟦										
PCB layout		🟦	🟦	🟦								
			🟩	🟩								
Order PCB				🟥								
Begin .py foundation to read analog values from Arduino		🟦										
		🟩										
Begin .py foundation for screen control		🟩										
		🟧										
Write simple script to test screen control			🟧									
Design simple circuit to test analog reads			🟦									
Frame Design Sketches			🟥 **ALL**									
Breadboard/Arduino Integration			🟥	🟥	🟥							
Breadboard/Arduino/Windows Integration					🟥							
Revaluation of accuracy, assess need for extra resolution					🟥							
Frame Construction w/o electrical components						🟥						
Overall product testing and optimization							🟥 *	🟥				
Finalize frame construction w/ electrical components									🟥			
Reach goals:												
Programmable 3-finger gestures							🟥					
Begin adding multi-finger functionality into software						🟧						
						🟩						