V.I.A.

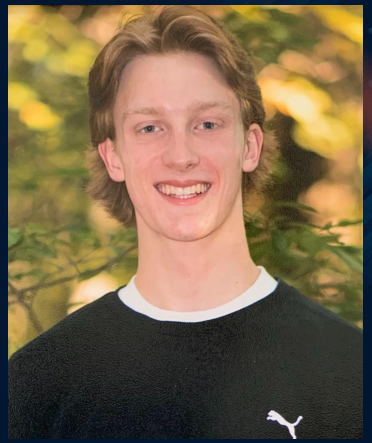
Visually Integrated Assistant

Meet the team

Antonio Romanoski Neto
Electrical Engineering

Lucas Georg Guertin
Electrical Engineering

Kevin Alfred Bardhi
Electrical Engineering



Jared Louis Simmons
Electrical Engineering

New Problem Statement

Modern advancements in voice and eye control of computers allow for more accessibility options for individuals with disabilities.

Voice control options are the most common way for increasing accessibility options for users

Eye control options are limited to systems that depend on an expensive investment into specific equipment and control capabilities are limited to one's own computer.

Current Eye Tracking Technologies : Tobii Eye Tracking

Pros:

- Has windows support
- High accuracy and quality
- Use in a multitude of different applications such as gaming, computer navigation, remains on while system is idle, etc

Cons:

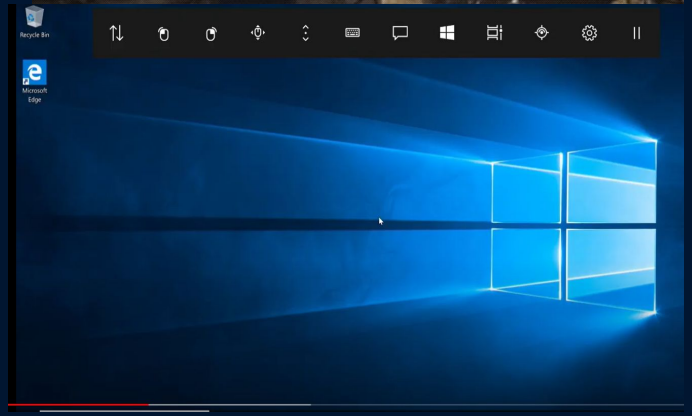
- Requires separate hardware purchasing of separate eye tracking equipment (Tobii Eye Tracker 5, Tobii Eye Tracker 5L, etc)
- Accessibility capabilities limited to only computer navigation
- Separate modules are expensive >100$

Tobii Eye tracker 5

Current Eye Tracking Technologies : Tobii Eye Tracking

Gaming Tobeii Usage:

Windows Tobeii GUI:



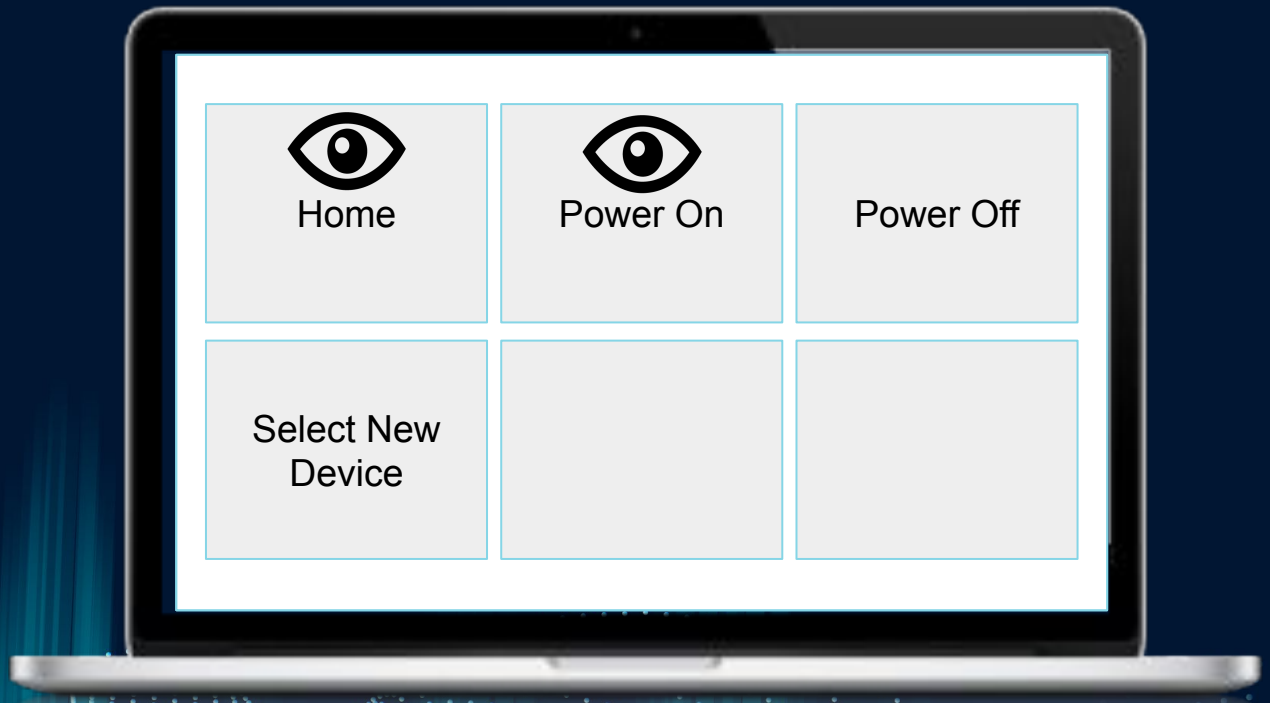
Proposed solution

Create a simple, cost-effective eye-tracking system that can be utilized in any laptop to control various devices in a given closed environment.

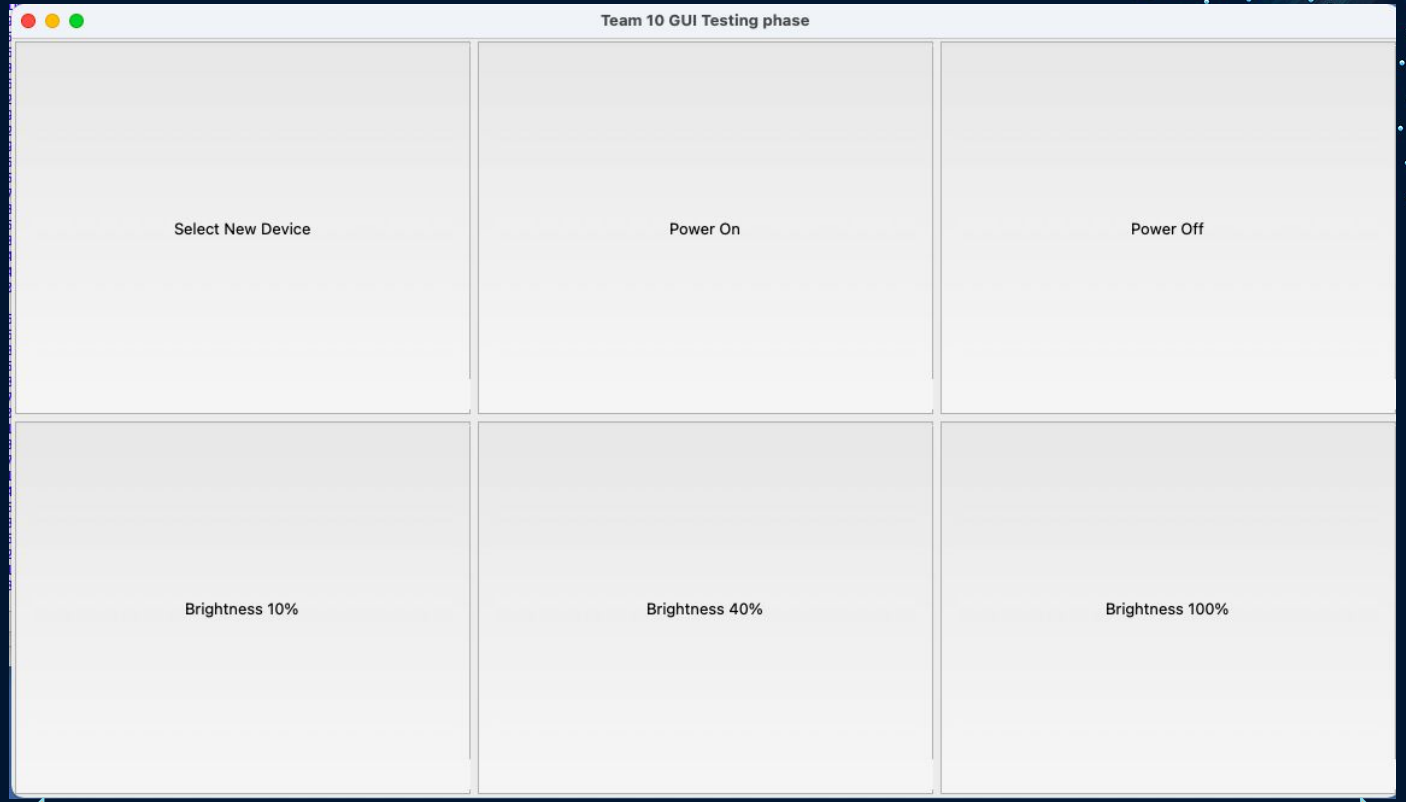
Project Goal

To design and create a cyber physical system that utilizes human eye movement as an input, and allows the user to control different connected devices wirelessly within a given closed environment.

Example Demonstration



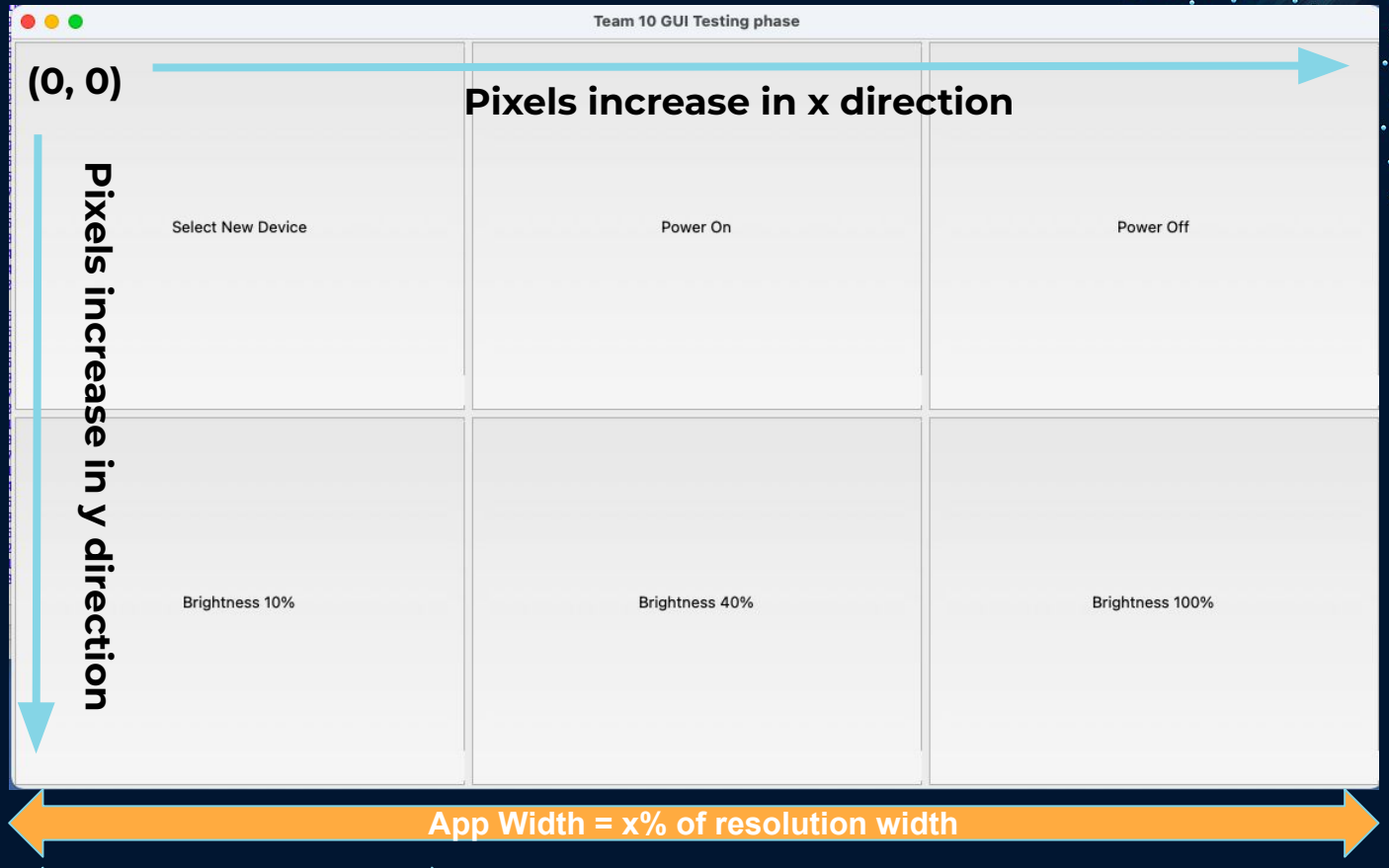
Explanation of Interactable GUI

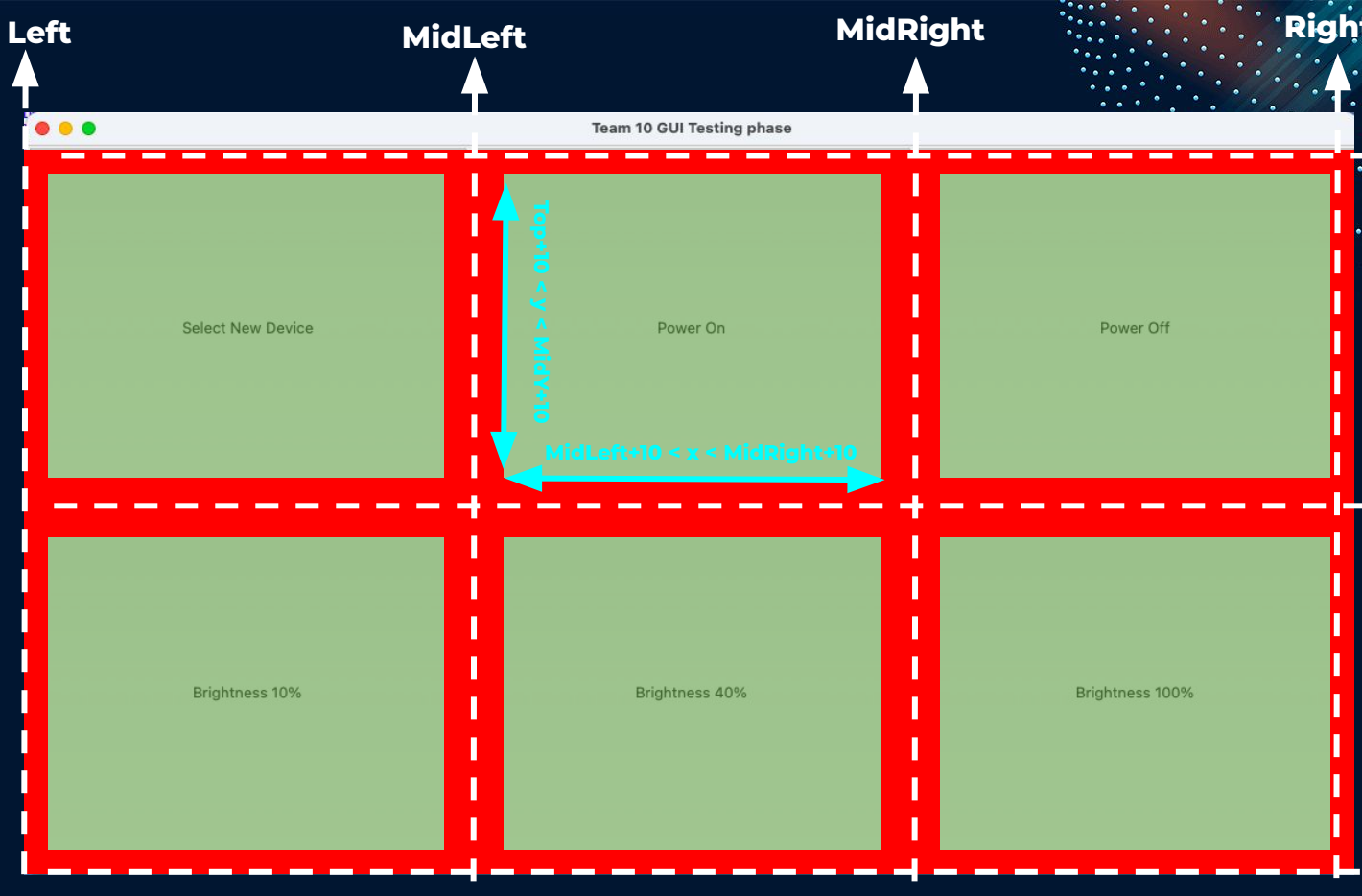


App Width = x% of resolution width

App Width ÷ 3

Explanation of Physical GUI





Creating Margins like this will prevent the user from accidentally clicking the wrong button.

It also give the program a way to reset the elapsed seconds as the cursor goes between boxes

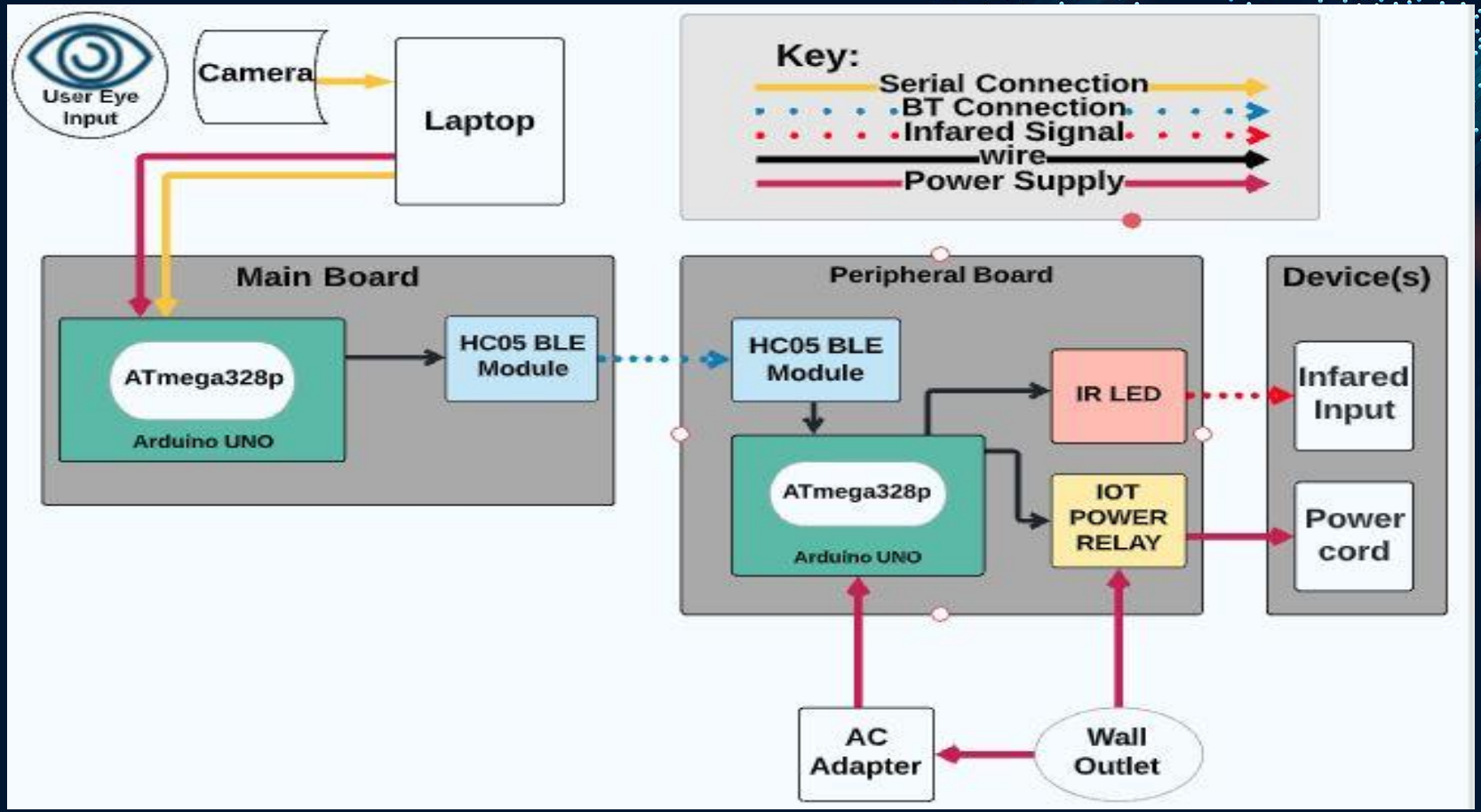
Note: All borders reference the app height and app width from before

Clickable space

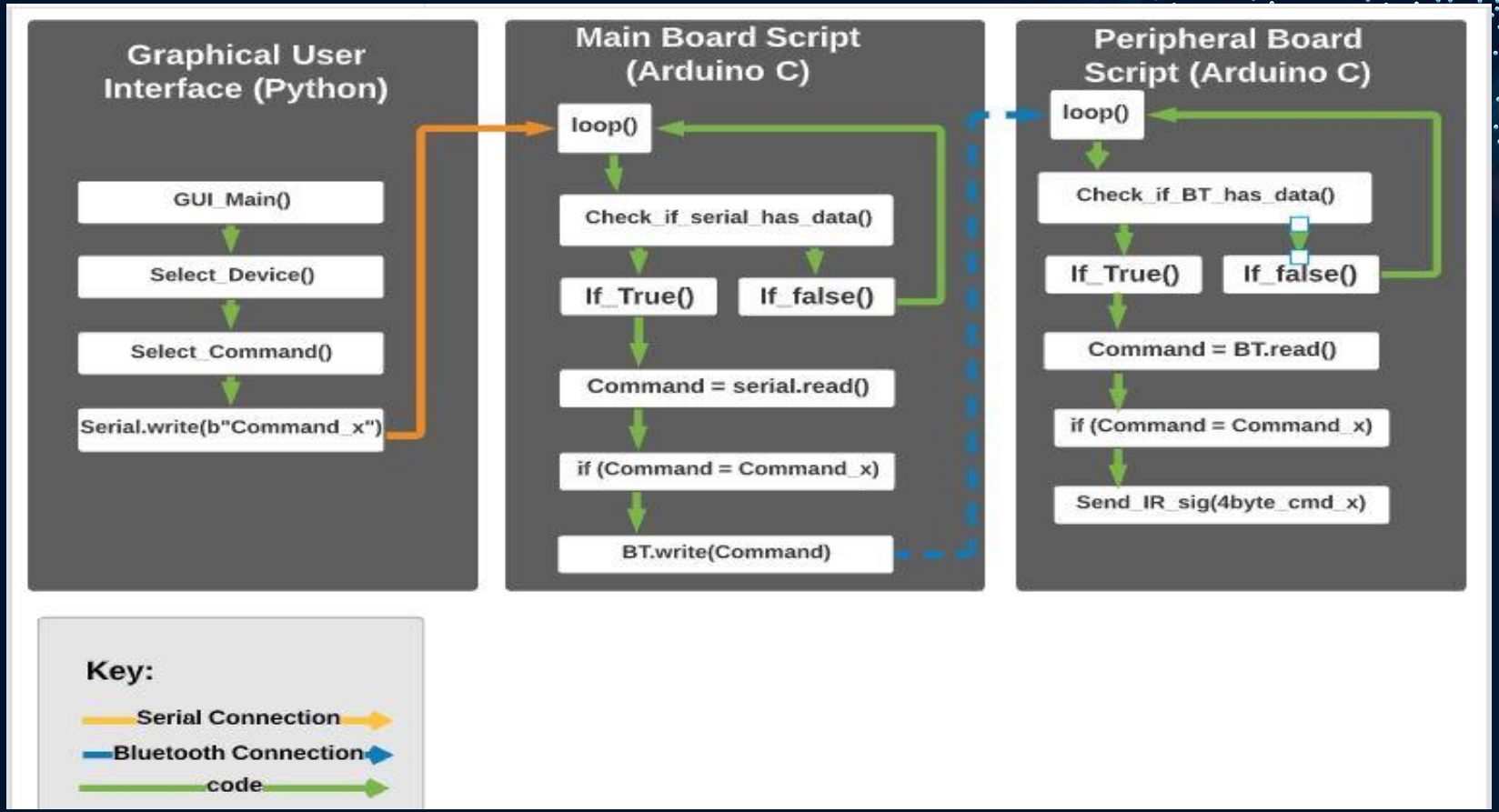
Non-clickable space

Exact borders of buttons

Current Hardware Block Diagram



Current Software Block Diagram



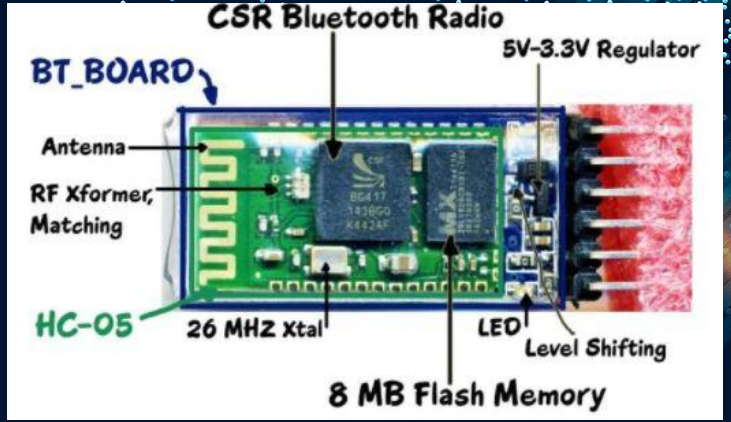
Components used

- Hardware
  - HC-05 Bluetooth express board (2x)
  - KY-022 IR Receiver Module
  - Infrared Diode LED 940nm
  - Arduino Uno (2x)
  - Adafruit power Relay feather
- Software
  - Python
    - PySerial Library
    - PyAutoGui library
  - Arduino C
    - IRremote.h library
    - SoftwareSerial.h library
  - Gaze Pointer

Component Description: HC05 express board

Device Aspects

- Operating Voltage: 3.3V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: ~ 9m (30ft)
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread Spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode
- Supported baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800
- For MDR we are using 38400 Baud rate across all device



System Specifications and Tests

Command Execution Test

- **Test: Total # of successful executions**
- **Results: (90/100) = 90%**

Current IR Range

- **Test: Range until IR Emitter doesn't work**
- **Result: 18-19 meters**

Current BT Range

- **Test: Range until BT Signal is unreliable**
- **Result: 10-11 meters**

Accidental Click Test

- **Test: Total time between incorrect clicks**
- **Results:**

Attempt	Time(min)	Accidental Clicks	User	Goal
1	4	2	Guertin	Exploring GUI
2	3	1	Guertin	Exploring GUI
3	5	2	Guertin	Exploring GUI
4	3	3	Guertin	Exploring GUI
5	4	2	Guertin	Exploring GUI
6	5	3	Guertin	Exploring GUI
7	4	0	Guertin	Exploring GUI
8	4	0	Antonio	Exploring GUI
9	2	0	Guertin	Turn On Lights
10	3	0	Guertin	Turn On Lights to 10% Brightness
11	3	0	Guertin	Turn On Lights to 40% Brightness
12	2	0	Guertin	Turn On Lights to 70% Brightness
13	2	0	Guertin	Turn On Lights to 100% Brightness

LIVE DEMO

MDR Deliverables

Prototype GUI with eye-tracking software:

- ✅ Show prototype GUI
- ✅ Show cursor response to user eye movement
- ✅ Show command selection with timed cursor placement

Main board Capabilities:

- ✅ Establish Serial communication between GUI and mainboard
- ✅ Establish BT communication between Main Board and Peripheral

Peripheral Capabilities:

- ✅ Show device power control using commands from GUI
- Show IR device control using commands from GUI

Areas of needed improvement: Eye Tracking implementation

Current Problem:

Our current eye tracking program , Gaze Pointer, works well ***in parallel*** with our GUI, however, we require a more fully implemented form of the eye tracking software os that the GUI and eye tracking software works in tandem in one program

Possible Solutions:

- Try to find a way to access the the GazePointer API in python
- Try to create our own eye tracking software using python that initializes as part of the GUI using current open source projects

Areas of needed improvement: Eye Tracking implementation

Current Open source Eye tracking projects in Python

Pygaze

Gazetracking

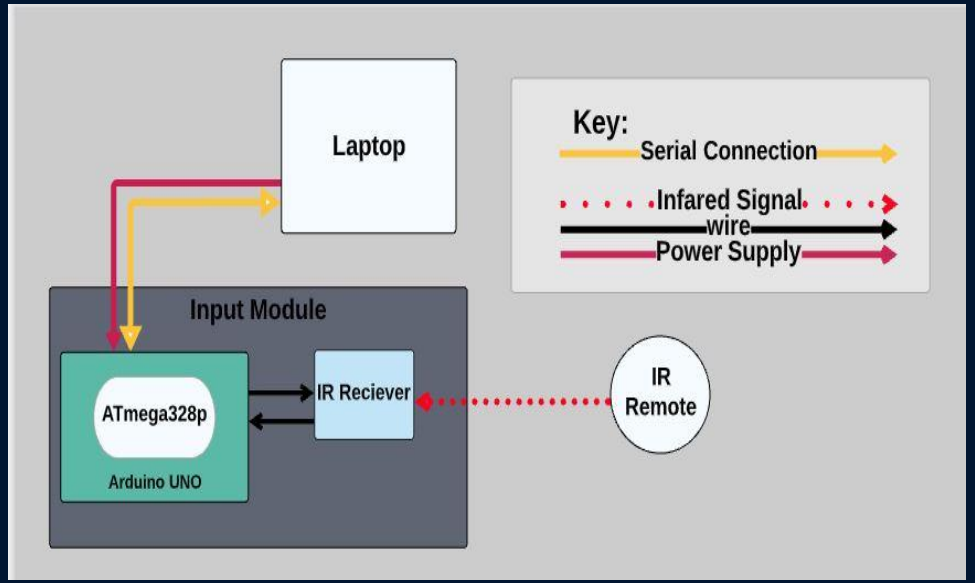
Gazeflow

MobyEyeTracking

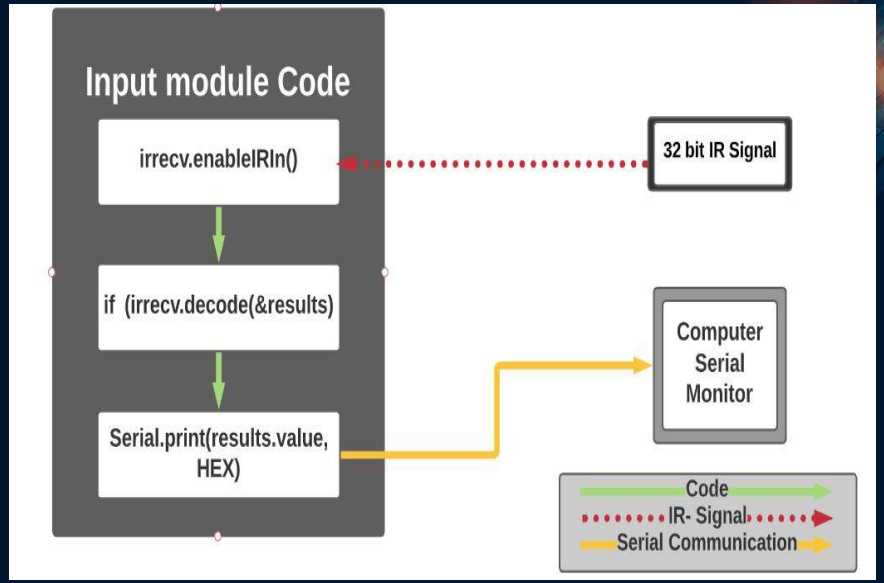
EyeTracker

Areas of needed improvement: More Robust method of input

Hardware Block Diagram



Software Block Diagram



Areas of needed improvement: More Robust method of input

Current Method:

- Relies on separate input unit
- Requires by hand sampling of every command per device
- Input module writes 32 bit strings corresponding to specific command onto serial port
- Codes are accessed by Arduino Serial Monitor
- Codes are hand assigned to commands and hard coded into peripheral board

Current Command key:

- ON: IR Code: FF807F, # of Bits: 32
- OFF: IR Code: FF00FF, # of Bits: 32
- 10%: IR Code: FF708F, # of Bits: 32
- 40%: IR Code: FF28D7, # of Bits: 32
- 80%: IR Code: FF609F, # of Bits: 32
- 100%: IR Code: FF10EF, # of Bits: 32

Areas of needed improvement: More Robust method of input

Plan for CDR

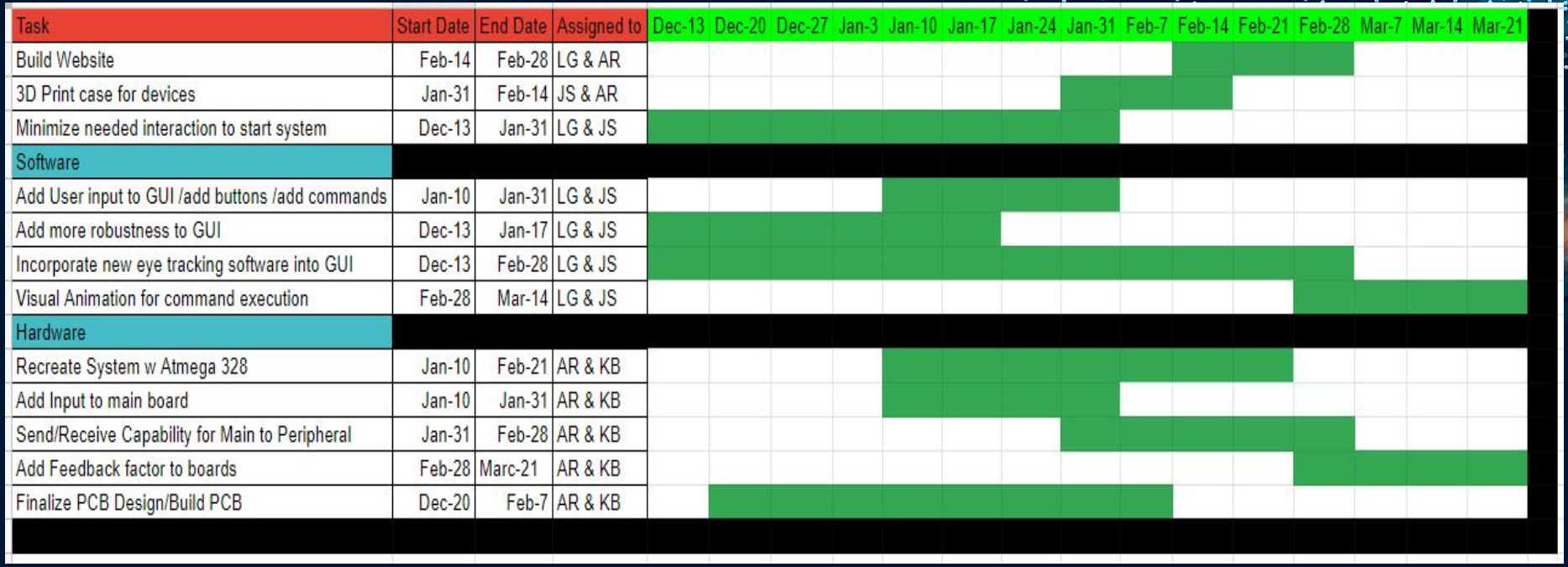
- Configure GUI to write 32 bit strings to serial port
- Configure Main Board to parse serial messages up to 32 bits using an extra 16 bits for delimiters “{“ and “}”
- Configure Peripheral Board to parse Bluetooth messages up to 32 bits using an extra 16 bits for delimiters “{“ and “}” for a total of 54 bits of information
- Add 2 way communication for feedback capabilities

PCB Plan for CDR/FDR

- Main Board
  - Components
    - HC-05 Bluetooth module
    - Infrared Receiver
    - Atmega328p
    - External Crystal oscillator
- Peripheral Board
  - Components
    - HC-05 Bluetooth module
    - Infrared Transmitter
    - Power Relay
    - Atmega328p
    - External Crystal oscillator

Gantt Chart to CDR

Task	Start Date	End Date	Assigned to
Build Website	Feb-14	Feb-28	LG & AR
3D Print case for devices	Jan-31	Feb-14	JS & AR
Minimize needed interaction to start system	Dec-13	Jan-31	LG & JS
Software			
Add User input to GUI /add buttons /add commands	Jan-10	Jan-31	LG & JS
Add more robustness to GUI	Dec-13	Jan-17	LG & JS
Incorporate new eye tracking software into GUI	Dec-13	Feb-28	LG & JS
Visual Animation for command execution	Feb-28	Mar-14	LG & JS
Hardware			
Recreate System w Atmega 328	Jan-10	Feb-21	AR & KB
Add Input to main board	Jan-10	Jan-31	AR & KB
Send/Receive Capability for Main to Peripheral	Jan-31	Feb-28	AR & KB
Add Feedback factor to boards	Feb-28	Marc-21	AR & KB
Finalize PCB Design/Build PCB	Dec-20	Feb-7	AR & KB



Current Expenditures

Total expenditures for MDR presentation	cost	amount	total price
IR Emitter LED	2	1	2
IR Receiver Diode	2	1	2
HC-05 Wireless Bluetooth RF Transceiver	18	2	36
IOT Power Relay	30	1	30
Shipping	42	1	42
Total			112
Total Expenditures for reasearch and develpoment			
nRF52840 Express	25	2	50
LoRa Radio Transceiver	40	1	40
Total			90
Future/miscalleanous expenditures			
Bare Atmega328p chip	2	2	4
PCBs	13	5	65
Roku TV	130	1	130
Shipping	50	1	50
Total			249
Total project expenditures			451

Teammate Responsibilities

- Antonio
  - PCB design
  - IR Output development
  - IR Input development
- Jared
  - Schedule management
  - Eye tracking implementation
- Kevin
  - Team Coordinator
  - Bluetooth communication development
  - Serial communication development
- Lucas
  - Financial Tracker
  - GUI development
  - Research and development

Demo video