(Draft)

Introduction

Digital eye strain is a common condition that affects people who spend time using display devices. It has been shown to have multiple negative effects on vision, and most people are potentially affected by it. The primary strategy for preventing digital eye strain is prevention - adjusting the viewing environment to minimize the potential for eye strain [1].

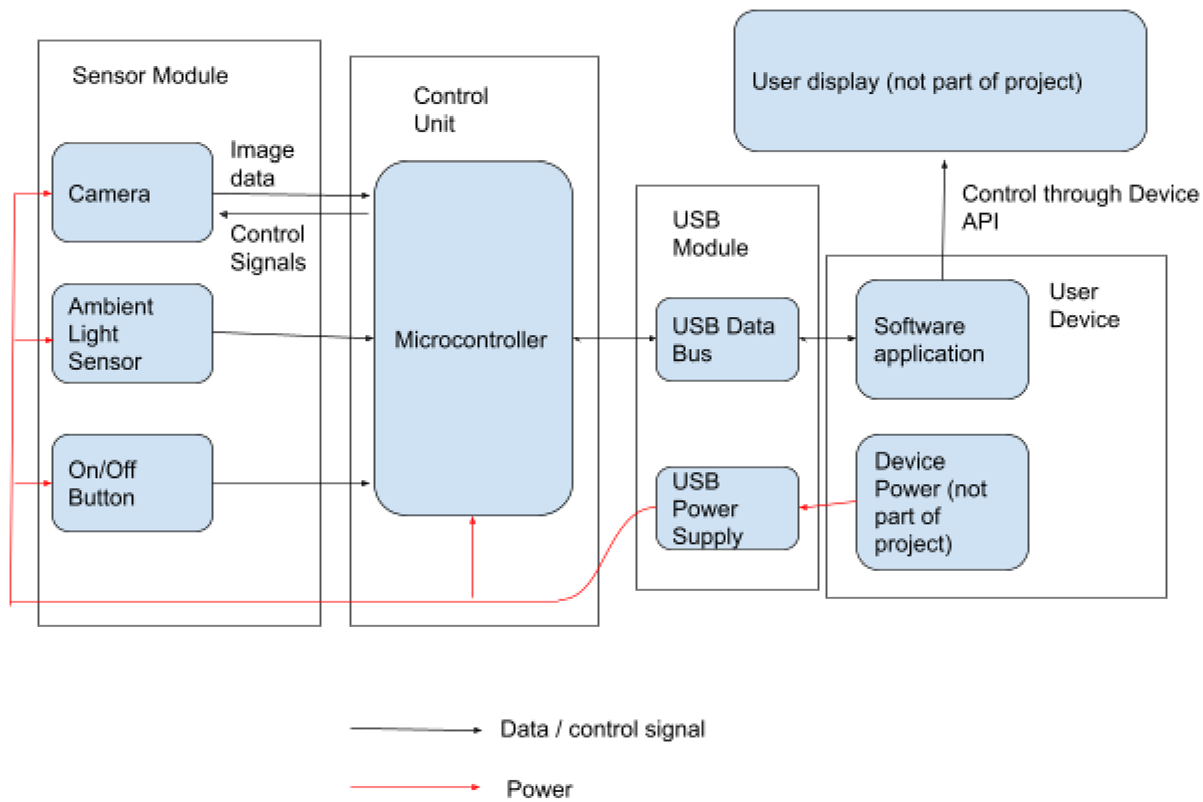
Two symptoms associated with digital eye strain are reduced blink rate and blink completeness. These symptoms can be detected through computer vision, and serve as a sign that the current viewing environment is not healthy for the user's eyes [1]. There exist some software-only solutions that aim to detect and alleviate eye strain using this strategy, but their functionality is limited and require the user to actively work against their eye strain [2].

Our project aims to relieve eye strain from viewing devices without inconvenience to the user. We plan on taking a vision-based approach to detect eye strain through blink data, and automatically adjust the user's display. We will use a camera, ambient light photoreceptor and a microcontroller to detect and communicate potential eye strain to software on a target device, which will adjust the target device's display to accommodate the user.

High-Level Requirements

1. Camera-microcontroller system should be able to capture, pre-process, and transfer image data to the USB at a rate of at least 10 frames per second.
2. The software application should be able to accurately, within 75% accurate frames, perform blink detection on 10 frames per second, without stalling or crashing, and create a noticeable change in the display's settings based on this detection.

Block Diagram



Physical Design

The camera will be mounted on a purchased adjustable mount for the time being.

Requirements & Verification Tables

Sensor Unit

OV7670 camera module

TI OPT300x ambient light sensor

Requirement	Verification
1. The camera should be able to capture images and deliver them to the microcontroller at a rate of at least 10 frames per second. 2. The photodetector delivers a signal with levels that can be converted by the A/D converter on the microcontroller. It should be able to detect the actual ambient light level within 50 lux.	1. Display a timer or test pattern on another device, then have the camera capture images at the maximum capture rate and intended resolution for 5 seconds. At least 50 distinct images are captured. a. The microcontroller will be needed to transfer and confirm the captured images. 2. Use a dimmer switch or other adjustable light to change the ambient

	light level of the environment. Use the microcontroller to check the outputs of the A/D converter, and compare the results with a digital light meter in the same environment.

Control Unit

Arduino microcontroller

Requirement	**Verification**
1. The microcontroller must be able to pre-process each camera image before requesting the next image at a rate of at least 10 frames per second.	1. Have the microcontroller pre-process the worst-case image data for our chosen algorithm multiple times in a row, and mark a timestamp on a connected console when each image is finished processing. The time between all timestamps should be less than 100ms. a. Unprocessed image data should be pre-loaded on-board for the purpose of this verification.

User-side Software

C++ application using Win32 API

Requirement	**Verification**
1. The blink detector must be able to recognize if there is an open or closed eye in the frame at least 75% of the time if the user is looking at the monitor. 2. The blink detector must be able to run detection on the images at a rate of 10 frames per second.	1. Create a set of images where the contents are known, with the resolution of our camera, and run it through the detection. At least 75% of the images should be correctly categorized as having an open eye, closed eye, or no eyes. 2. In the same way as testing the microcontroller's pre-processing, continuously run the program and timestamp when each frame is finished processing. The time between each timestamp should be less than 100ms.

Tolerance Analysis

We will need to perform tolerance analysis on the built-in image settings of the camera. The subsampling rate, color balance, color mode, and cropping of the camera directly impact the data transfer rate between all the modules, and the effectiveness of the eye detection.

Ethics and Safety

One consideration to be made is the security of the eye tracking information we collect. Given Principle 1.6 of the ACM Code of Ethics, we should respect the privacy of the user, and keep this data limited in its scope [4]. Because we are gathering image data from the user, we need to make sure that the data is only used for the purpose of checking the user's eye-strain condition and the surrounding environment. This data should be processed immediately, exclusively for our project's purpose, and immediately disposed of when no longer needed.

Another concern is adjusting the brightness of the system too rapidly. As we are adjusting the monitor settings according to the data from the sensors, there is a possibility that the brightness setting could be changed in a harmful manner. In accordance with the IEEE Code of Ethics #9, we must avoid injuring others [5]. Thus we need to ensure that the system is not too reactive, as doing so risks changing the brightness too quickly, potentially damaging the system and posing an epilepsy risk. Thus we need to limit the rate of brightness change for the system itself. We plan to limit the adjustment interval to 5-10 minutes, and limit the amount to about a 10-20% change in brightness.

Cost Analysis

Schedule

Citations