Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Summer 2019

EyeT Guys
The Eye Tracker

Jonathan El-Khoury
Paul Wafula
George Hinkel
Dharampreet Gill
Jay Denton

Revision History

Revision	Date	Author(s)	Description
0.1	6.28.2019	JE	document creation

CONTENTS

List of Figures

List of Tables

1 Introduction

Your introduction should provide a brief overview of the product concept and a reference to the requirement specification and architectural design documents in 1 or 2 paragraphs. The purpose is to provide the reader with the location of relevant background material that lead to the design details presented in this document.

2 System Overview

The Eye Tracking Remote has two main layers: the Software Layer and the Hardware Layer. The Software Layer implements the eye-tracking algorithm using the image data imported from the Hardware Layer. It also calibrates gaze vector to screen coordinates conversions and wraps up the analyzed data in a packet which will be sent to EyeType's portion of the project. The Hardware Layer consists of the IR light and the camera to capture the user's pupil movements and sends this data to the Software Layer. It also sends the analysed data packet imported from the Software Layer to the EyeType system. Meanwhile, the analyzed data is also shown on the Character LCD for debugging purposes.

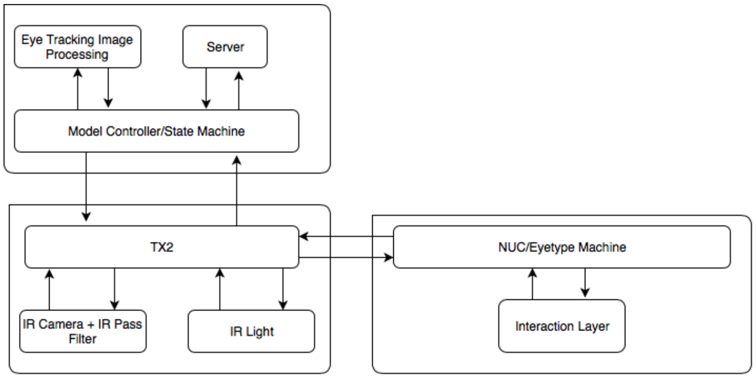
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 1: The Eye Tracker Overview Diagram

2.1 Hardware Layer Description

The hardware layer in our project consists of the TX2, the character LCD, IR light and the IR camera. Each of these perform the necessary tasks that will be needed for the software layer to send data to the NUC (EyeType team). The TX2 which runs Ubuntu will be used to gather data from the camera and the character LCD which we will be send to our software layer.

2.2 Software Layer Description

The Software layer consists of the Eye tracking image processing, calibration, and the server subsystems. The Eye tracking image processing subsystem receives and processes data from the IR camera which will be used to track Marci's eye. The character LCD will be used for debugging purposes. As for calibration, it will be done initially when we start the program so we can calibrate where Marci's pupil is looking in relation to the screen. As for the server, it will communicate between our layer and the NUC which is the EyeType team's system.

2.3 Eyetype Layer Description

This layer is the responsibility of EyeType and will be the system that Marci directly interacts with.

3 Hardware Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be ommitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

3.1 Hardware Layer Hardware

This layer is comprised of several hardware subsystems however a component that is universal to the entire layer is the power supply and battery for the wheelchair and the Eye Tracker.

3.2 Hardware Layer Operating System

No operating system is required for the previously mentioned components in section 3.1.

3.3 Hardware Layer Software Dependencies

No software dependencies are needed for the power supply and battery.

3.4 Controller Subsystem

The controller subsystem is comprised of the NVidia TX2 micro-controller and is a piece of hardware.

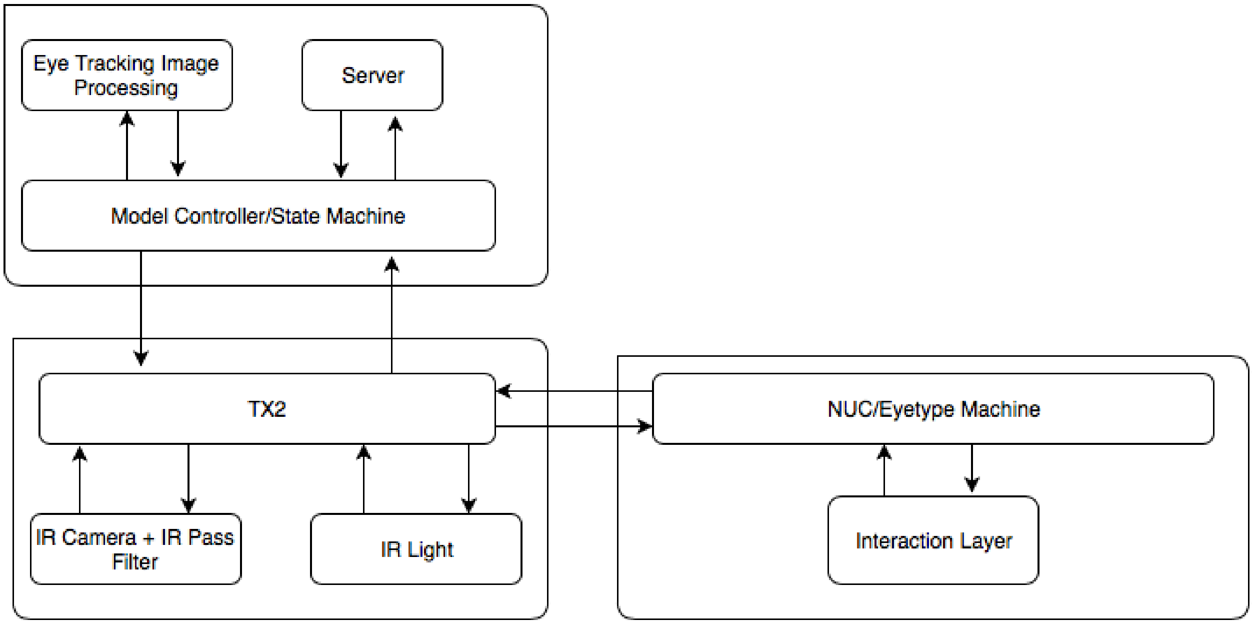
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 2: TX2 Subsystem Diagram

3.4.1 Controller Hardware

The hardware components contained in this subsystem are the NVidia TX2 on a Auvidea J90 Carrier Board and a power cable.

3.4.2 Controller Operating System

This subsytem will utilize the Ubuntu 18.04 LTS operating system.

3.4.3 Controller Software Dependencies

There will be multiple software dependencies contained in this subsystem. These will include CUDA, OpenCV, and MIPI camera driver.

3.4.4 Controller Programming Languages

CUDA and the OpenCV libraries will be used within a C++ environment.

3.4.5 Controller Data Structures

Data will be transmitted from the TX2 to the NUC via a USB 3.0 cable through sockets. The TX2 will be sending the gaze vector data encoded through XML from the socket to the NUC.

3.4.6 Controller Data Processing

The controller subsystem will be utilized to perform high level data processing in the software layer. However, the subsystem will perform some low level data processing, including the transmission of the data packets over the USB, voltage analysis on the battery and battery capacity calculations, and raw MIPI image data reception and conversion to a standardized format to be used by the image processing software within the software layer.

3.5 IR Camera + IR Pass Filter Subsystem

The IR Camera + IR Pass Filter subsystem is a hardware subsystem responsible for receiving image data using IR light which excludes light from the visible spectrum.

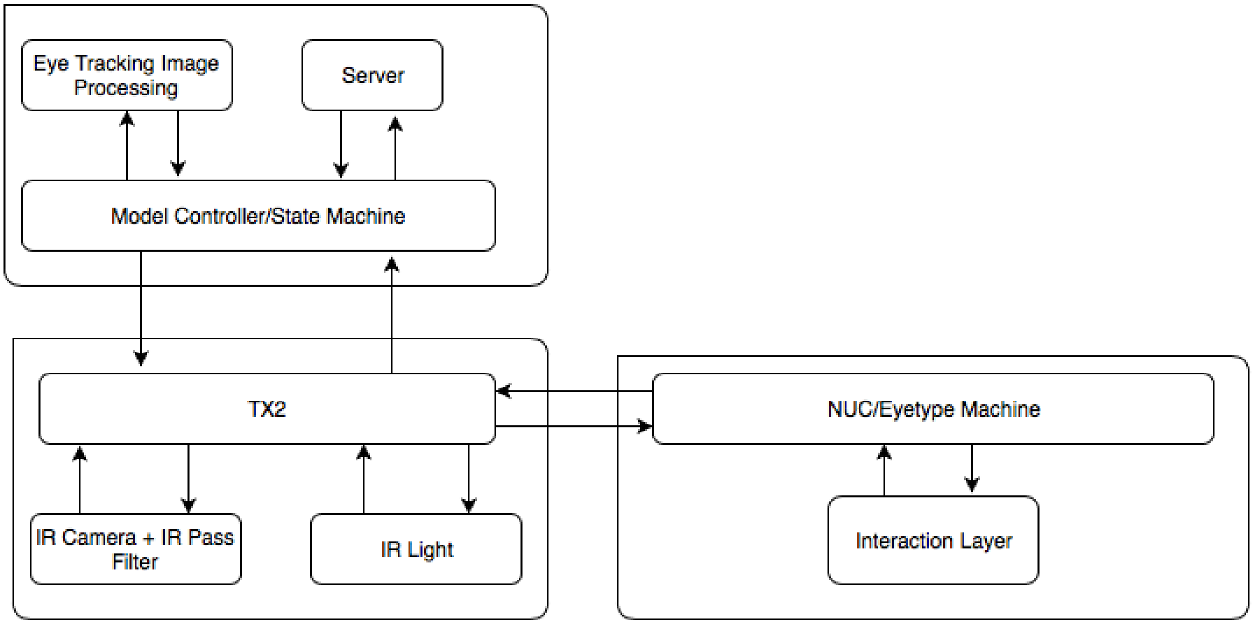
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 3: TX2 Subsystem Diagram

3.5.1 IR Camera + IR Pass Filter Hardware

The hardware components involved with this subsystem are the X290 IMX Image Sensor, M12 50 mm lens, and a 750 nm wavelength IR pass filter.

3.5.2 IR Camera + IR Pass Filter Operating System

This subsystem will not utilize any operating system.

3.5.3 IR Camera + IR Pass Filter Software Dependencies

This subsystem layer will need a MIPI camera driver in order to process image data received and send to the required software components as well as manage eye tracking. This will be discussed in further detail in the software subsystem layer description.

3.5.4 IR Camera + IR Pass Filter Programming Languages

The camera driver will likely be written in C++ as eye tracking and image processing are time-sensitive processes and by writing the driver in C++ we will be able to limit the amount of latency that the system will have to deal with.

3.5.5 IR Camera + IR Pass Filter Data Structures

Data will be transmitted from the camera to the TX2 via a 4 lane CSI-2 MIPI interface. From the TX2, gaze vector data will be sent after being encoded into XML from the TX2 to the Intel NUC.

3.5.6 IR Camera + IR Pass Filter Data Processing

The IR Camera + IR Pass Filter subsystem will be receiving image data with visible-light filtered out that will then be sent, as mentioned above, through a 4-lane CSI-2 MIPI interface to the TX2 which will then distribute the image data to the software layer and its respective subsystems to be utilized.

3.6 IR Light Subsystem

The IR light subsystem is a hardware subsystem responsible for illuminating the eye-tracking subjects eye in the Infrared spectrum, thus allowing for a clearly lit image without interfering the vision of the subject in the visible spectrum or otherwise causing discomfort or interference.

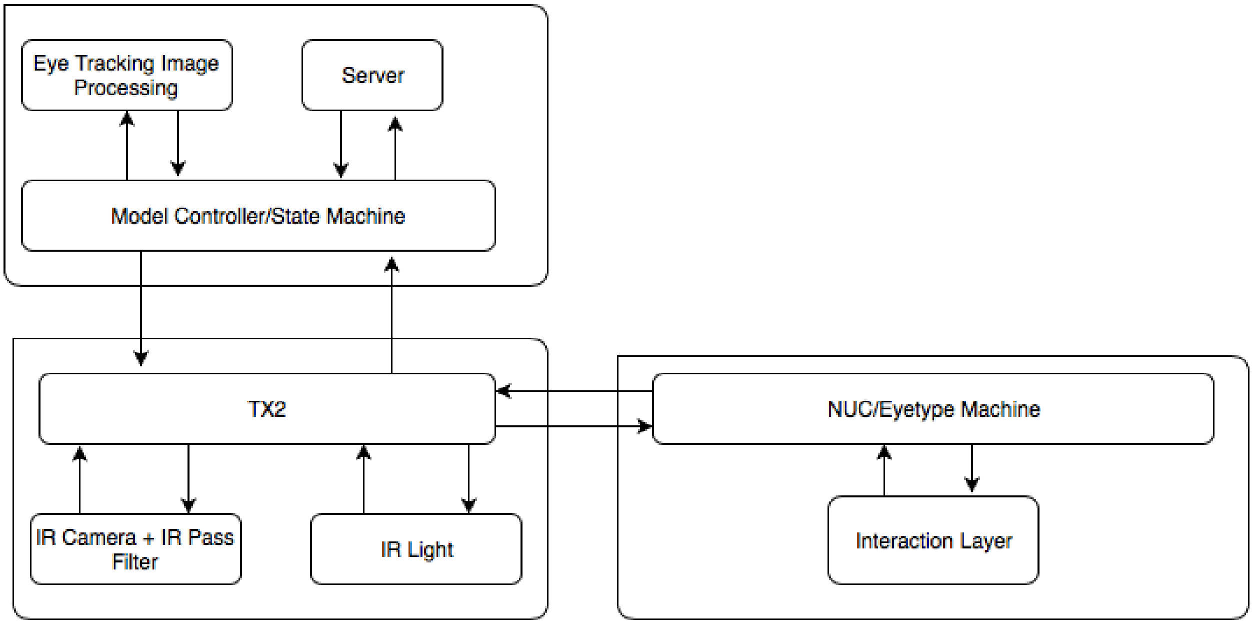
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 4: TX2 Subsystem Diagram

3.6.1 IR Light Hardware

The hardware components involved with this subsystem are 8 x 5mm Infrared (IR) emitting Light Emitting Diodes (LEDs) and a Printed Circuit Board (PCB) which mounts to the camera fixture and contains the circuitry to connect to the power system and illuminate the LEDs.

3.6.2 IR Light Operating System

This subsystem will not utilize any operating system.

3.6.3 IR Light Software Dependencies

This subsystem layer will need a driver to enable the over-arching operating system to interact with the micro controller's hardware systems in order to enable a General Purpose Input/Out output to drive the LEDs.

3.6.4 IR Light Programming Languages

The IR Light Driver will be written in C++ or C or be outsourced and be written in any language that it comes in.

3.6.5 IR Light Data Structures

The data structure for the IR light system consists of a single byte/OS Word with a single bit of importance, when this word is set to 0's the light will be driven low and turn off, when this word is set to anything other than 0's the light will be driven high and turn on.

3.6.6 IR Light Data Processing

The data processing for the IR light will consist of performing set and clear operations on a designated location in system memory which controls the GPIO port connected to the LEDs.

4 Software Layer Subsystems

4.1 Software Layer Hardware

The software layer of the system will rely on the hardware devices described in the previous hardware section. They are the TX2, the IR Camera + IR Pass Filter, the IR LEDs, the battery, the power supply, and the LCD display.

4.2 Software Layer Operating System

This layer, not unlike the hardware layer, will be using the Ubuntu 18.04 LTS Operating System.

4.3 Software Layer Software Dependencies

The libraries that will be used in this layer include CUDA and OpenCV. Also possibly to be used is the Open Gaze API.

4.4 Model Controller/State Machine Subsystem

The purpose of this subsystem is to control communication between the hardware components and the respective software components.

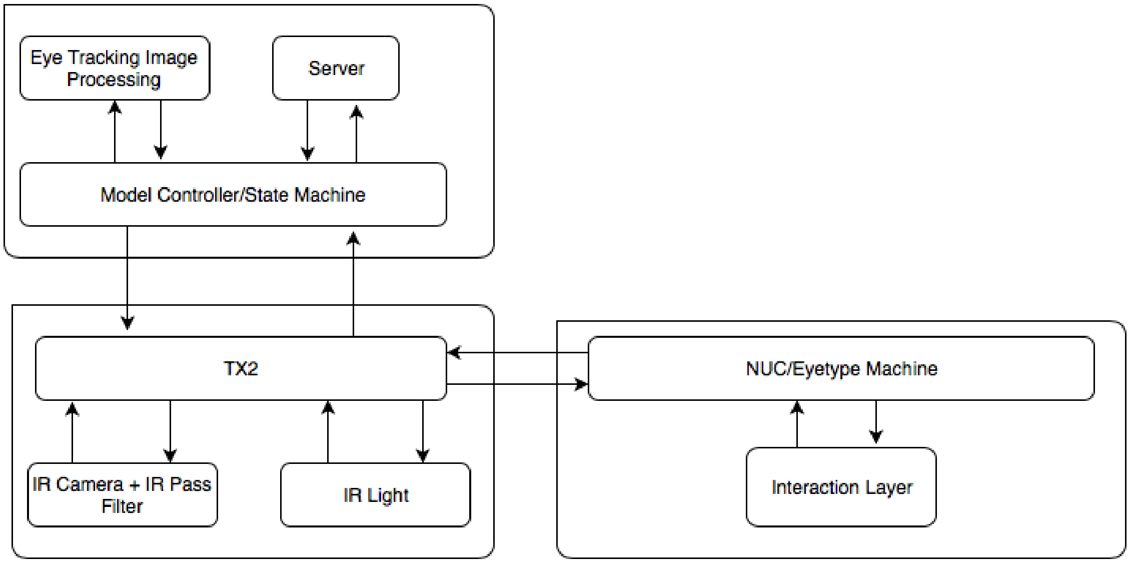
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 5: Model Controller/State Machine diagram

4.4.1 State Machine Subsystem Hardware

The Model Controller/State Machine will be on the TX2 micro-controller which will interact with the camera and LEDs.

4.4.2 State Machine Subsystem Operating System

This subsystem will be using the Ubuntu 18.04 LTS Operating System.

4.4.3 State Machine Subsystem Software Dependencies

As of right now, the only software dependencies that will be needed are those that come with the C++ programming language.

4.4.4 State Machine Subsystem Programming Languages

This subsystem will use C++ in order to reduce latency.

4.4.5 State Machine Subsystem Data Structures

Data will be transmitted from the TX2 to the NUC via a USB 3.0 cable through sockets. The TX2 will be sending the gaze vector data encoded through XML from the socket to the NUC.

4.4.6 State Machine Subsystem Data Processing

This subsystem will be responsible for relaying information between the hardware layer and the rest of the software subsystems. This will be done on the fly as image data is received, it will be sent to the eye tracking image processing subsystem in order to be processed.

4.5 Eye Tracking Image Processing Subsystem

The Eye Tracking Image Processing Subsystem is responsible for processing video sent in from the hardware layer and tracking Marci's eye. As her eye moves, the subsystem will continuously identify her eye movements and accurately track her pupil.

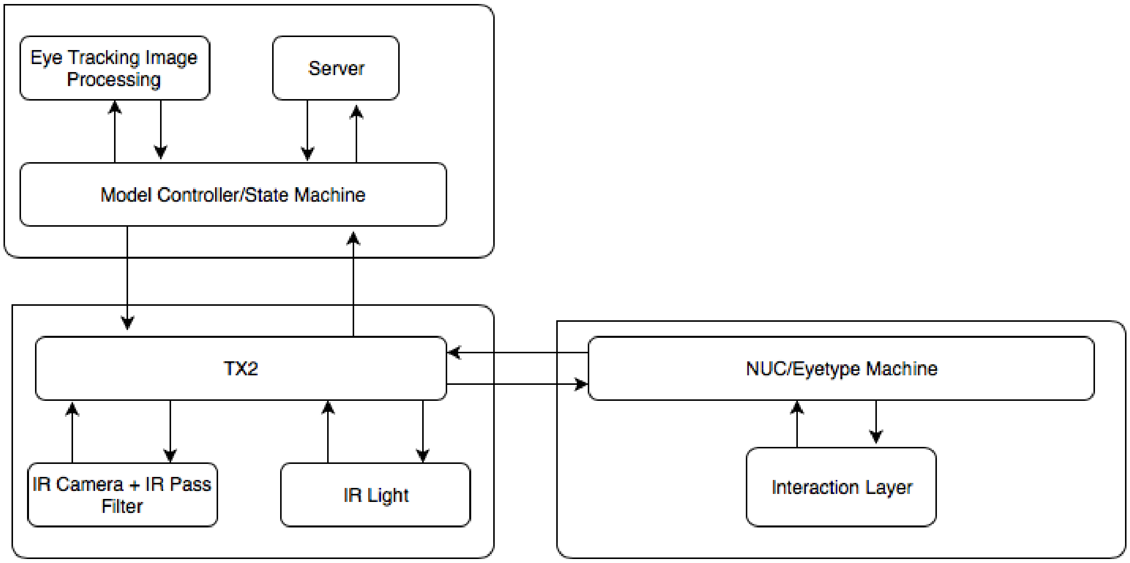
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 6: Eye Tracking Image Processing diagram

4.5.1 Eye Tracking Image Processing Subsystem Hardware

The Eye Tracking Image Processing Subsystem will be on the TX2 micro-controller.

4.5.2 Eye Tracking Image Processing Subsystem Operating System

This subsystem will be using the Ubuntu 18.04 LTS Operating System.

4.5.3 Eye Tracking Image Processing Subsystem Software Dependencies

Software dependencies for this subsystem include OpenCV for C++.

4.5.4 Eye Tracking Image Processing Subsystem Programming Languages

This subsystem will use C++ in order to reduce latency.

4.5.5 Subsystem Data Structures

Data will be contained in a PupilTracking Class that will also be responsible for containing the coordinates of Marci's eye on the fly.

4.5.6 Subsystem Data Processing

As image data is received from the hardware layer, this subsystem will dynamically calculate the location of the pupil as it changes positions. It will then send the x and y coordinates back to the model controller which will be sent over the Eyetype layer for calibration purposes.

4.6 Server Subsystem

The server subsystem will be responsible for relaying x and y coordinates of Marci's eye to the Eyetype layer for calibration and usage purposes.

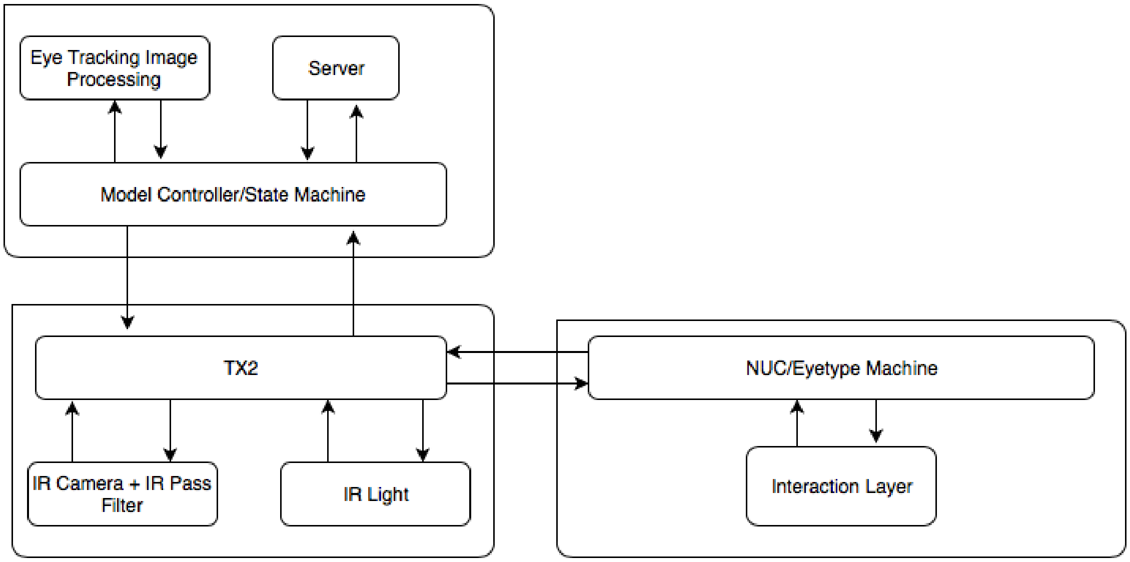
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 7: Server Subsystem diagram

4.6.1 Server Subsystem Hardware

The Server Subsystem will be on the TX2 micro-controller.

4.6.2 Server Subsystem Operating System

This subsystem will be using the Ubuntu 18.04 LTS Operating System.

4.6.3 Server Subsystem Software Dependencies

As of right now, the only software dependencies that will be needed are those that come with the C++ programming language.

4.6.4 Server Subsystem Programming Languages

This subsystem will use C++ in order to reduce latency.

4.6.5 Server Subsystem Data Structures

Data will be packaged into XML style packets to be transmitted over to the Eyetype layer (NUC).

4.6.6 Server Subsystem Data Processing

The packets will be sent over a socket using socket programming to the NUC to be used for calibration and usage purposes.

5 EyeType Layer Subsystems

5.1 EyeType Layer Hardware

The EyeType layer consists of many software processes running on a host computer that will also be used by the user as a standard consumer personal computer. The computer used in this layer is an i5 Intel NUC kit mini personal computer with an M2 hard drive and 8 GB of RAM installed.

5.2 EyeType Layer Operating System

The operating system utilized in this layer on the NUC computer will be Ubuntu 18.04 LTS.

5.3 EyeType Layer Software Dependencies

This layer will have software dependencies on multiple libraries, including any that will be required for the customer to do her work. It will also be dependent on the EyeType software to be able to enact the eye tracking control of the system.

5.4 NUC Subsystem

The NUC subsystem handles the calibration, and all of the eye typing and communication activities.

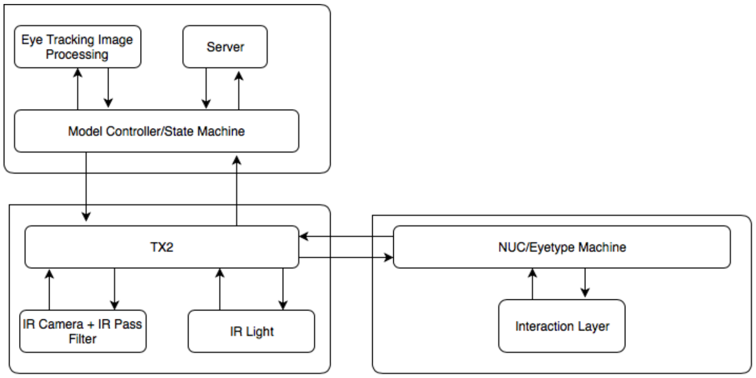
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 8: Example subsystem description diagram

5.4.1 NUC Subsystem Hardware

The main hardware involved in the interaction subsystem is the NUC

5.4.2 NUC Subsystem Operating System

The operating system in this subsystem is still Ubuntu 18.04 LTS

5.4.3 NUC Subsystem Software Dependencies

The only software dependencies are the operating system.

5.4.4 NUC Subsystem Programming Languages

This subsystem has no specific programming languages.

5.4.5 NUC Subsystem Data Structures

This subsystem will utilize any data structures present in the hardware.

5.4.6 NUC Subsystem Data Processing

This subsystem will take in information from the TX2 and use it in the Eyetype software.

5.5 Interaction Subsystem

The interaction subsystem handles the interaction between the EyeType system and the EyeT Guys system.

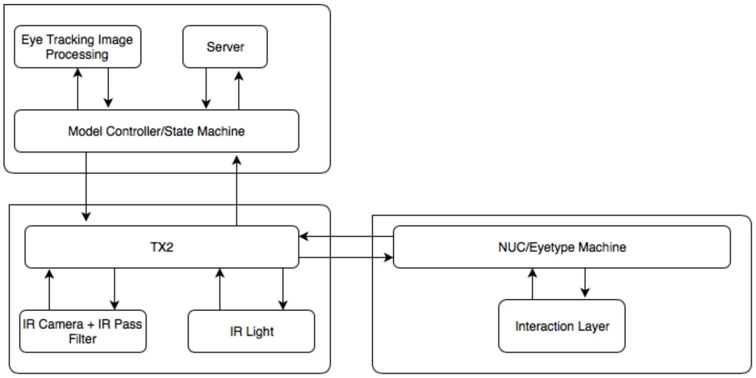
Eye Tracking Image Processing
Server
Model Controller/State Machine
TX2
NUC/Eyetype Machine
IR Camera + IR Pass Filter
IR Light
Interaction Layer

Figure 9: Example subsystem description diagram

5.5.1 Interaction Subsystem Hardware

The main hardware involved in the interaction subsystem is the NUC

5.5.2 Interaction Subsystem Operating System

The operating system in this subsystem is still Ubuntu 18.04 LTS

5.5.3 Interaction Subsystem Software Dependencies

Software dependencies utilized by this subsystem include those related to sockets and also a driver that decodes the XML data sent over the sockets.

5.5.4 Interaction Subsystem Programming Languages

This subsystem will be programmed in C++ for low latency.

5.5.5 Interaction Subsystem Data Structures

This subsystem will utilize a FIFO ring buffer to store incoming packets until they are processed by the driver, then passed to another FIFO ring buffer as data objects until they are used by the Eyetype software.

5.5.6 Interaction Subsystem Data Processing

This subsystem will parse XML packets and extract the data about the screen coordinate approximation as well as the exact timestamp in order to accurately recreate the customer's gaze position.

6 Appendix A

References