Visually Integrated Assistant (V.I.A)

[1]

Jared Simmons, EE, Kevin Bardhi, EE, Antonio Romanoski Neto, EE, and Lucas Guertin, EE

Abstract—**Smart home systems provide a more interactive and optimal form of controlling devices in a controlled environment. However, many of the existing systems rely heavily on internet connectivity and voice control. V.I.A. (Visually Integrated Assistant) seeks to provide a different method of control, depending solely on the eye movement of a designated user. Our project displays a prototype system that can make use of eye movement as an input, allowing the user to communicate with our GUI on their laptop and control a set of LED lights and a Roku television.**

I. INTRODUCTION

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.

A. Significance

This problem is specific to a group of people with a high-level mobile disability. We believe there is a lack of solutions for people with such disabilities that allow them to freely navigate and control devices in their own home without the need for expensive investments, or the need for burdening others to provide round the clock assistance. As an example, Steven Hawking, a famous theoretical physicist survived for 55 years with Amyotrophic Lateral Sclerosis (ALS), a neuro degenerative disease that attacks motor neurons in a human's brain until voluntary control of muscle is lost throughout the whole body. Because of his inability to move around freely he relied on the care of a nurse 24 hours a day until he passed away in 2018. ALS as well as other severe diseases or injuries have a large effect not only on the person that carries the burden, but also their caretakers.

B. Context and Competing Solutions in Marketplace

Currently, there is a broad overview of things that could function in a similar way as our proposed solution. Those would include the Amazon Alexa kit and the Google Nest kit. Amazon and Google both help individuals function easier, especially with difficulty, but they weren't built with their disabilities in mind. These specific products are entirely voice controlled, so our example Steven Hawking would not have been able to benefit from them personally. Additionally, these solutions often come at a cost, and are sometimes only compatible with external devices that are part of the same family of branded products. Another existing solution is the Tobii eye tracker, which can accurately track the movement of a user's eye gaze upon a monitor and change the location of a computer's cursor simultaneously. However, this product is specified towards aim assist and control accuracy in gaming and would not be useful to the audience that we are aiming for. Our solution also differentiates itself from Amazon and Google by being more eye focused instead of voice controlled. This allows for different control options than normal voice commands can give.

C. Societal Impacts

Our goal is to help improve the daily lives of people with mobility disabilities to help improve their independence from others and allow a broader spectrum of choice for what they are able to control. This will hopefully have quite a few positive outcomes for our constituencies such as increasing the awareness of the issue and making the issue easier to address. Many individuals with severe mobile disabilities depend on round the clock assistants from family members or professional caretakers. The idea behind our product should not only help disabled individuals navigate their own devices, but alleviate the weight of their caretakers responsibilities as well. The only major negative that we could foresee happening, is that it ends up being more expensive than we want it to be. We want this product to be easily accessible and as inexpensive as possible. The impact we hope to achieve with our design is to make the lives easier for our end demographic target. We wanted to make it as hands free as possible while still being compatible with common household devices. This led to us using an eye tracking software, which is usable by an everyday laptop, to be the focal point of control.

D. System Requirements and Specifications

Operating Voltage:	3.3V to 6V (Typically +5V)
Operating Current:	30mA
Range:	~9m (30ft)
Operating Baud Rate:	38400 byte/s
Power Source:	FTDI Board: Computer Power Peripheral Board: Wall power

Table 1: Requirements and Specifications [11]

The goal for our system is to utilize human eye movement input as an input to a GUI we are hosting on a computer in order to control different devices throughout a given restricted environment using Infrared signals. In order to achieve this we need a component that allows us to wirelessly communicate across a room. Our component of choice was the HC05 BLE BT module. It's a Bluetooth module that utilizes UART communication in order to send and receive data. Figures 1 and 2 give general overviews on the pinout

[1]
F. A. Author from Lexington, Ma (e-mail: author@ boulder.nist.gov).
S. B. Author, Jr., from Auburn, Ma (e-mail: author@lamar.colostate.edu).
T. C. Author from Quincy, Ma (e-mail: author@nrim.go.jp).

and components on the Hc05. Table 1 shows our current system specs with the HC05 fully implemented. While this component handles the data transmission between our computer and any other hardware connected to a receiving HC05, we still need a way to convert the data signals from our computers USB to TTL logic. For this we employ the use of an FT232RL chip [14]. This adapts the data from a given computer into data signals that can be utilized by the HC05. More in depth descriptions about the implementation of these 2 hardware components in the design portion of this paper.

HC05 BT BLE Module
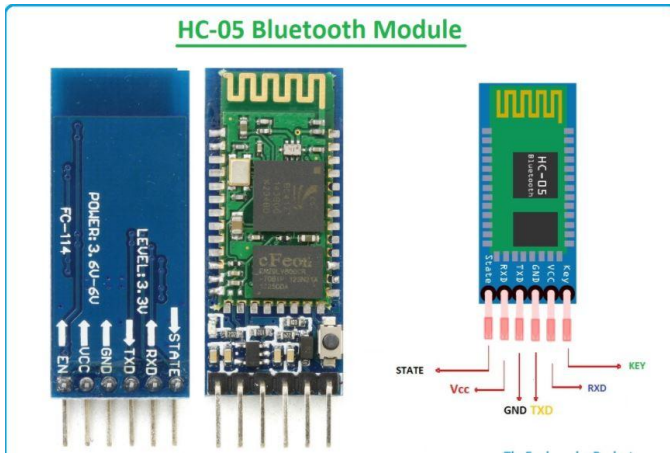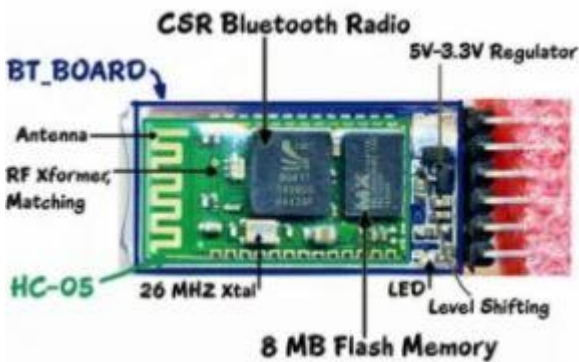
Figure 1: HC05 Pinout [12]

Figure 2: General overview of HC05 components

II.   DESIGN

A.   Overview

Our product is made up of 4 main subsystems: Our FTDI conversion PCB, the "Peripheral" PCB, the interactable GUI, and the open-source eye tracking system "Gazepointer". Gazepointer is meant to run in parallel with our GUI and simply changes the location on a mouse based on a person's calibrated eye gaze. Though the calibration takes about a minute and a half, it is capable of saving calibrations, meaning a user could run our GUI multiple times without having to recalibrate Gazepointer. The GUI then automatically reads the location of the cursor so that the user can interact with it handsfree. For the first PCB, the 'FTDI Board', is connected to the laptop via USB and serves the purpose of reading the GUI activity and transmitting 32-bit commands via Bluetooth to the Peripheral Board. The second PCB, which we refer to as the 'Peripheral Board', serves the purpose of receiving Bluetooth signals from the FTDI converting PCB, and transmitting them as infrared signals to the given IR based devices in one's home.
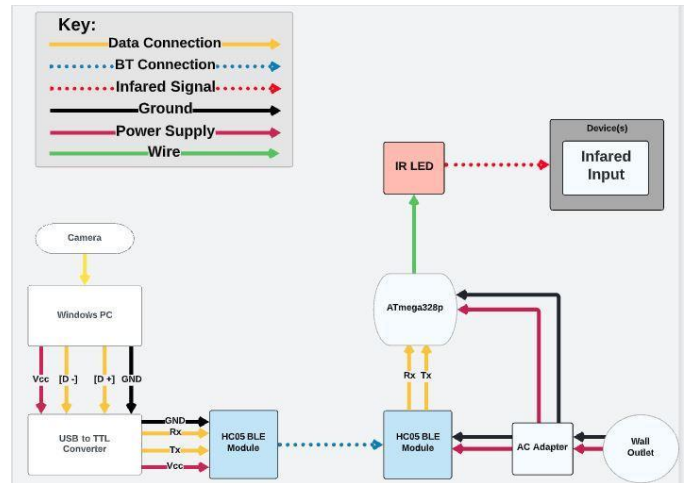

Figure 3: High-level Overall Block Diagram of our system

B.   Peripheral Board

The Peripheral Board PCB is responsible for communication between the laptop and our target devices. Figure 4 shows the high-level functionality that this board has for our system. The Peripheral board is composed of an ATMega328P Microcontroller, an HC-05 Bluetooth module, a USB Mini-B port, a 16 MHz crystal, a fuse, and an IR emitting diode. The ATMega328P is used to program the board and has the purpose of analyzing the Bluetooth signal received by the HC-05 and deciding which IR signal should be sent out of the IR emitting diode. The 16 MHz crystal, accompanied by two resonating 22pF capacitors, is used as a crystal oscillator for the microcontroller. This board will receive its power from a traditional wall outlet, so the USB Mini-B port is to provide this connection between the board and the outlet. Lastly, the fuse is positioned as a safety measure to avoid overcurrent. This board is what connects the 'buttons' on the GUI to actual IR signals to control out devices.
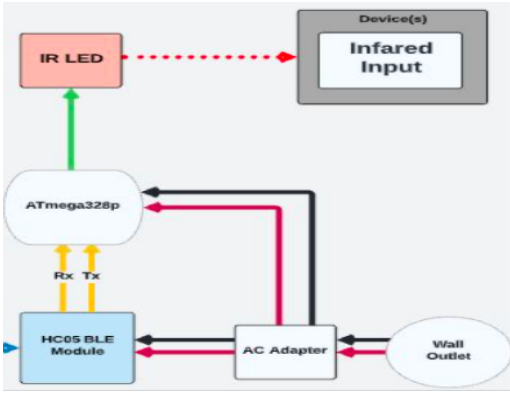
Figure 4: High-level Block Diagram of Peripheral Board

C. FTDI Board

The FTDI PCB's responsibility is to read whatever the user has selected in the GUI and transmit a Bluetooth signal containing that information (a 32-bit string) to the Peripheral Board. This board is composed of an FT232RL chip, an HC-05 Bluetooth module, a fuse, and a USB Mini-B port, as can be seen on Figure 5. The FT232RL chip is designed to convert the data signals on the data+ and data- lines to TTL logic for the HC-05 to send the correct corresponding signal to the Peripheral Board. The fuse and USB Mini-B port provide the same functionalities as mentioned above for the Peripheral Board.
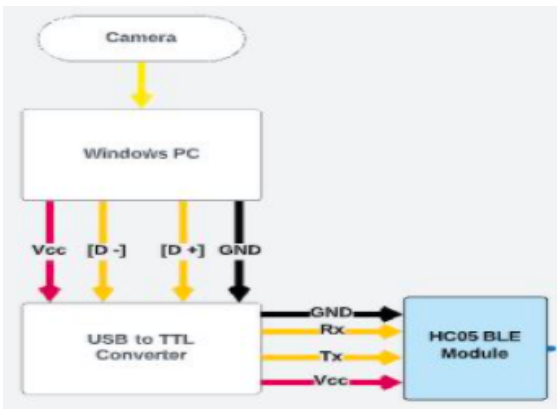

Figure 5: High-level Block Diagram of FTDI Board

D. Graphical User Interface (GUI)

Our custom user interface is meant to be intractable at any time in a smooth and comfortable manner. When running the program, the user will first see a window containing a 3x3 grid of buttons with various selectable options. The program continuously keeps track of the computer's cursor location based on its current (x, y) pixel coordinate. The range of these x and y values are dependent on the resolution of the computer's monitor. To make our program compatible with all types of screen resolutions, we adjust the size of the interactable space to be a percentage of the total monitor resolution. To do so we create two variables called "screenheight" and "screen width" that are set to the height

and width of the monitor in pixels. By creating these variables, we are also able to calculate the size of each button so that we can define their borders and create "empty" space between them. This empty space is defined so that nothing is accidentally clicked. Figure 6 gives a visual representation of how the app and buttons are sized in the GUI. This method is robust and valid with all laptop resolutions.
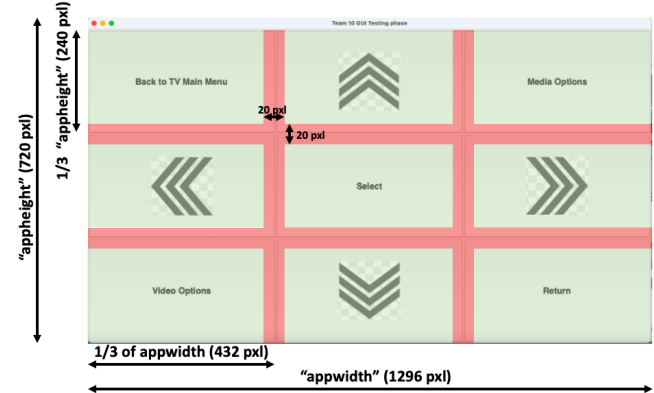

Figure 6: This example from our GUI shows the dimensions in pixels of the app window and its buttons for a screen resolution of 900x1440. Clickable space is green while non-clickable space is red.

Since we know the size of all the buttons and the size of the GUI window, we can define the spaces that we want to click and the spaces we do not want to click anywhere outside of the GUI window and small margins between all of the buttons with a width of 20 pixels are considered empty space, and the areas inside the GUI's buttons are considered clickable space. Next comes timing and automatic clicks. We first start a timer that continuously counts from zero to 3 seconds, and then resets back to zero. By utilizing a series of condition statements, we can start or stop this timer only while the cursor is hovering over our predefined clickable space. Therefore, we are able to assign a clicking action to the cursor only when it has been hovering over a button for 3 seconds. If the cursor leaves the button before the timer reaches 3 seconds, no click will occur, and the timer will only restart once the cursor has reentered another button's clickable space.

E. Block 4

Describe what this technical block does. Explain what technology you used to build this block. Detail which techniques from courses you used to build this block. List what you learned in order to build this block. Summarize an experiment you performed to design or test this block. Refer to Appendix (Section C) for more information about the experiment as needed.

III. THE REFINED PROTOTYPE

A. Prototype Overview

Our current prototype is a pair of 2 PCBs, the first is named "FTDI Board" and the other is named "Peripheral Board." The Peripheral board is the component of our system that will be

placed near the desired IR device the user wants to control, and the FTDI board is connected to the computer hosting the GUI where the user selects the commands they want to execute. Both boards house an HC05 BLE Bluetooth module in order to communicate across a room smaller than 30ft. The Peripheral board houses an atmega 328p and an IR LED among other passive circuit elements. The FTDI board uses the computer as a power supply and the Peripheral board is designed to be able to be powered by a USB type A cable and a standard power brick. Our prototypes for this project went through multiple iterations. The iterations evolved from 2 separate arduino circuits, each hooked up to an HC05 and the 2 boards were originally designated as "main board" and "peripheral board" Where the main board was actually connected to the computer hosting the GUI. This led to issues such as library installation, errors in simultaneous communication between the two boards. Rather than waste time trying to completely copy what we had done in our first iteration we tried a new approach for our third iteration. We decided to scrap the idea of using a "Main" board and just used a USB to TTL converter, to send signals directly to the HC05 that was originally housed on the "Main Board". In our third iteration the use of a Hiletgo USB to TTL Converter[13] was employed. This USB to tLL converter was then replaced by our FTDI board.This allowed us to avoid the issue of over engineering and allowed us to significantly reduce the amount of hardware we need in order for our system to function. We converted our original main board design into what is now the FTDI board. We then moved forward with our third iteration of the prototype as the primary design of our system. Our fourth iteration is extremely similar to our third iteration except it is housed on a PCB instead of a breadboard

B. Expenditures

At the end of CDR we, as a team, had spent $480 on our total expenditures. As we had a bit leftover we ordered a few extra of some components to make sure we got working components. For FPR, we spent about $85 on 77 different components, PCB Revisions, and shipping. We stayed under budget, and if we had the knowledge we had now, I would be confident in saying we'd be able to reduce that cost by a large margin <$200, as PCB revisions alone were around $130. Figure 5 shows our total expenditures throughout the year.

Total Expenditure for CDR	Cost	Amount	Price
MDR Expenditures	112	1	112
Expenditures for research and development	90	1	90
PCB	91	1	91
Atmega-PU	21.26	2	42.52
500 mA Fuse	7.19	1	7.19
Adafruit RFM95W LoRa Transceiver	19.95	2	39.9
Shipping	9.03	1	9.03
150 ohm resistor	10	0.084	0.84
1k ohm resistor (R1)	10	0.042	0.42
2k ohm resistor (R2)	10	0.797	7.97
10k ohm resistor	10	0.042	0.42
22 pF chip capacitor	10	0.076	0.76
47 uF SMD chip capacitor	5	0.91	4.55
100 nF chip capacitor	10	0.254	2.54
Xtal 16MHz crystal oscillator	5	1.19	5.95
500 mA Fuse	5	0.86	4.3
Mini B connector	2	6.99	13.98
PCB FPR Shipping	25.34	1	25.34
PCB FPR Revisions	21.63	1	21.63
Shipping for FPR	2.24	1	2.24
Total			480.34

Figure 7: Total Project Expenditures

C. Project Management

Our team is usually organized into two main sub groups, this changes depending on if there happens to be a large focus that the group needs to take care of. The two sub groups are hardware and software, Lucas Guertin and Jared Simmons were responsible for the software portion of our project, while Kevin Bardhi and Antonio Romanoski Neto were responsible for the hardware.

D. Prototype Functionality

Using your block diagram and software diagrams, discuss what elements of your prototype were functioning and those that were not functioning. Provide evidence that your prototype worked and discuss the issues faced in troubleshooting the non-functioning elements.

E. Prototype Performance

Return to the Requirements and Specifications in Table 1 and discuss your prototype's ability to meet requirements by providing data showing that the specifications were satisfied. Discuss why some specifications were not met. Some other responsibilities that Lucas Guertin had was keeping track of project expenditures.

IV. CONCLUSION

A. Antonio Romanoski Neto Testimonial

All the lessons that this process has taught me I will carry for the rest of my professional life. All the programming, hardware design, PCB design has definitely increased my potential as an electrical engineer, however, what I will remember the most are all the problems we faced as a group. From overcomplicating our entire project to a global chip shortage, the Senior Design Project process has taught me that to have success you have to put in the countless hours of hard work and dedication. Throughout the PCB design process, it was very difficult to deal with the stress that goes into all of the process. If it weren't for the help provided by Chris Caron I would have ran into a lot more issues designing our PCBs. Overall, the PCB design showed me that creating a very well constructed and functioning PCB is extremely difficult and I am immensely grateful that I was able to pick up this portion of our Senior Design Project.

B. Lucas Guertin Testimonial

This process has taught me a few things, but I think the most important thing I learned was dealing with different communication issues. This is because I work full-time, which makes it a bit harder, and I have to rely on my teammates for the day to day communication on the hardware. This made me be the main POC for a lot of the software issues, the communication issues arise when my teammates or I are at different parts of a process. Sometimes there were miscommunications on what was happening, certain deadlines, getting in quick contact, etc.. It made my job somewhat more stressful, because I had a distinct lack of

control over an important portion of the project, but my team is a pretty good team. As such, while these issues arose they were dealt with in a timely manner and we were able to accomplish our goals. This is applicable in my professional life as well because no matter what company you end up working for, there will always be some communication issues at some point and learning how to cope and deal with them early is an important tool to have. That is why, while slow and sometimes stressful, communication is one of the most important aspects to a successful project.

C. Jared Simmons Testimonial

SDP has been a challenging yet extremely worthwhile experience. Designing a product from start to finish has not only honed my technical skills, but it has taught me leadership qualities and proper time management and prioritization. The four of us in our group often had schedules that didn't line up during the day, so it became important to us to be organized with our individual task prioritization. Since Lucas was fully remote because he worked full time, we also had to develop and practice strong communication. These are all skills that we will be able to apply to any endeavor that we approach even outside of the working world.

In a more technical sense, I have been able to greatly improve my overall engineering knowledge. Before the start of SDP I considered software one of my technical weaknesses, which is why I took on the GUI as one of my main parts of the project. Building the GUI taught me some excellent python methods that I will likely be using in the near future during my career as an engineer. On top of that, this is the first time I have been a part of a project from design all the way to implementation. Such experience is valuable to understand how products are brought to the market in the industry. I am grateful to have been a part of an excellent team and to have an exciting new project in its completion.

D. Kevin Bardhi Testimonial

The process of designing our system has taught me numerous skills and lessons that will be extremely useful and valuable in my future career. The most important skill I learned is time management/prioritization. Although this is a skill one hones throughout their entire college career, the act of trying to manage your time during your senior year is a completely different monster. I'm glad to say that being forced to find time for all my classes as well as senior design projects improved my time management skills tremendously and will be a tool I will use for the rest of my life. The process of learning the ins and outs of UART communication was extremely helpful as well. The courses I have taken at this university have for the most part focused on physics and application and not so much embedded system communication. The learning curve that came with trying to develop this portion of our system was steep but after numerous hours of hard work we were able to accomplish spectacularly. I'd also like to take this opportunity to thank a junior ECE undergrad named Jonah Yolles- Murphy who works at M5. It's because of his help that I knew how to go

about learning the UART communication process, and helped me avoid issues such as over-engineering our Peripheral board which would have cost the team many, many precious hours of work. I'm grateful that I was able to take part in a project that challenged me mentally and academically and forced me to not only grow as a student but as an engineer.

REFERENCES

[1] "HC-05 DATASHEET", Datasheet, Itead Studio, 2010
[2] "ATmega328P DATASHEET", Datasheet, Atmel, 2015
[3] "FT232RL DATASHEET", Datasheet, FTDI Chip, 2010
[4] "Control Arduino Using GUI (Arduino + Processing)", Website, Hackster [Online]. Available: https://www.hackster.io/hardikrathod/control-arduino-using-gui-arduino-processing-2c9c6c
[5] "HC-05 Pinout, Specifications, Datasheet and HC05 Arduino Connection." ETechnophiles, ETechnophiles, 15 Feb. 2022, https://www.etechnophiles.com/hc-05-pinout-specifications-datasheet/.
[6] Ali, Zahid, and Shayantika Dhar. "HC-05 Bluetooth Module Pinout, Datasheet, Features & Applications." The Engineering Projects, 22 June 2021, https://www.theengineeringprojects.com/2019/10/hc-05-bluetooth-module-pinout-datasheet-features-applications.html.
[7] "FT232RL." *FTDI,* 2 Mar. 2022, https://ftdichip.com/products/ft232rl/.