A.I.dan: ChatGPT Integrated Virtual Assistant

Andrew Scott

Leonardo Garcia

Brahmteg Minhas

Team 25

TA Hanyin Shao

02/09/23

ECE 445

1. Introduction

1.1 Problem

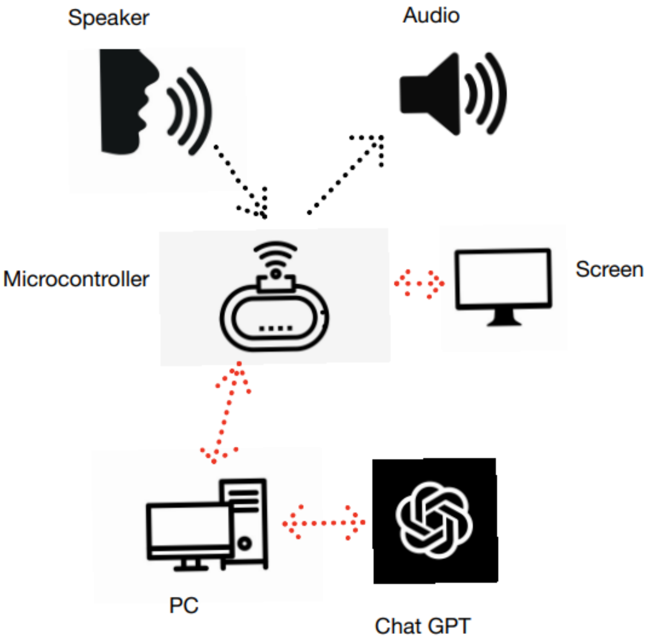
All existing virtual assistants use a search engine (primarily Google Search) as a data retrieval tool for answering questions posed to them by a user. The responses given by a search engine to a virtual assistant is a result or excerpt from the top search result that comes when searching for that question. However, this result is often not useful or relevant. Search engines are built to present multiple information sources from the web when presented with a question, not necessarily output one definitive answer. Furthermore, searches respond poorly to requests, unable to produce an output of a specified form.

OpenAI's ChatGPT is an AI language model that can respond to questions, generate text and have conversations. It is an extremely useful tool to respond to questions or present a singular output in a specified format. However, accessing ChatGPT requires going to the ChatGPT website, logging in and manually typing the question. This process is cumbersome and takes more time than it needs to.

1.2 Solution:

Our solution is a device that integrates ChatGPT with a virtual assistant in order to nullify the weaknesses of both tools. Because ChatGPT is trained on human language, it is much better suited for a virtual assistant, which uses human language as input. As such, integrating ChatGPT into a virtual assistant would yield a much better tool for getting information and tailored responses to user questions. Our solution will allow the user to present the virtual assistant with a prompt or question. The device will then return the response that ChatGPT gives to the user input, both through a speaker and through a screen on the assistant.

1.3 Visual Aid:

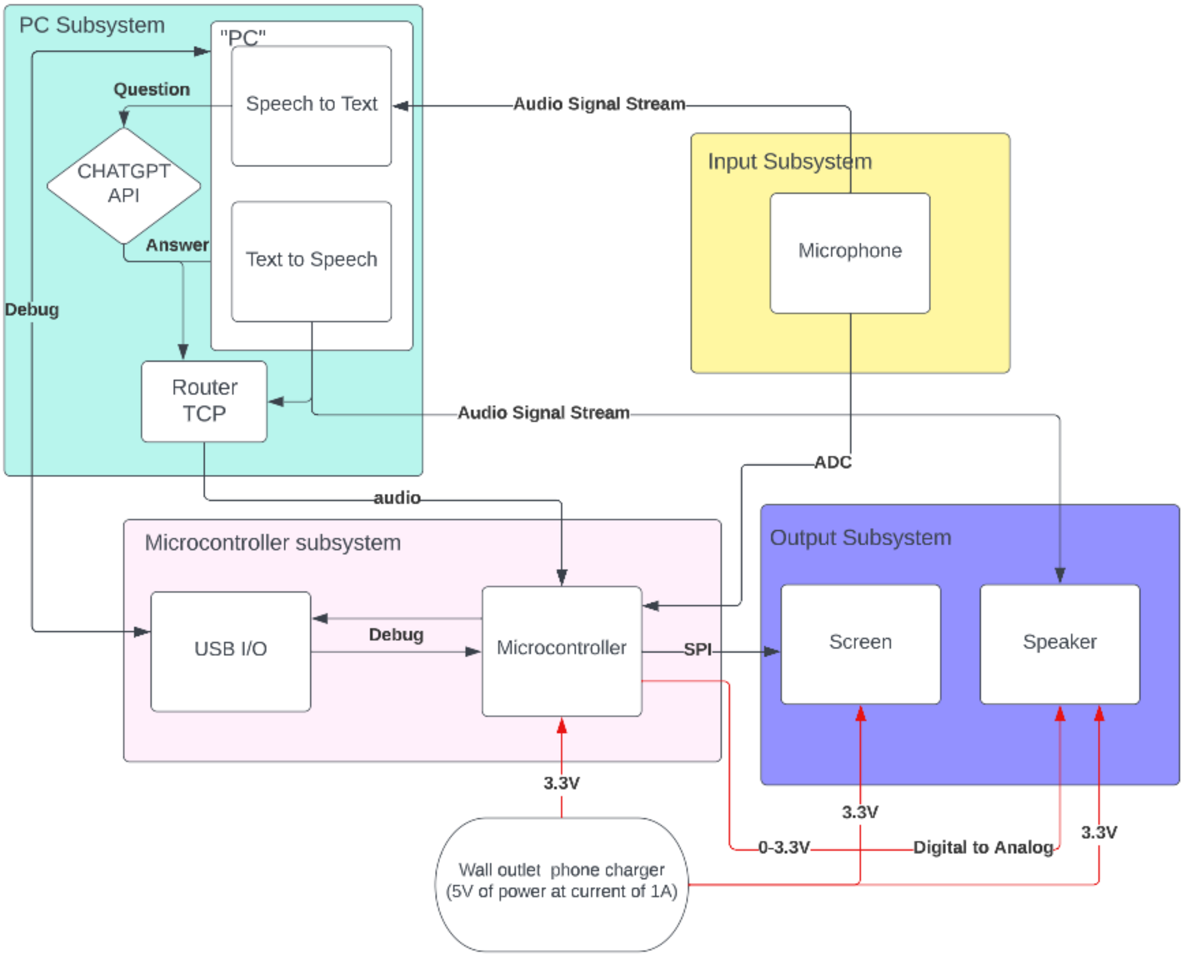


1.4 High Level Requirements:

- At least 75% of attempted basic interactions should be successful. Basic interactions are defined as questions based entirely on words included in a pre-trained speech to text model.
- Code (Markdown) or other text must display properly given a successful question. This is verified by inputting the same question to chatGPT on a separate device, or by logging the I/O of the PC's API Requests.
- Given a successful question with an appropriate audio response, the speakers must output an understandable Text To Speech interpretation. While mispronunciation is acceptable, 80% of users must be able to correctly interpret the audio.

2. Design

2.1 Block Diagram



2.2 Remote Block

The PC block consists of an external computer connected to the microcontroller through WiFi. The PC performs the majority of the heavy compute lifting of the entire project, performing the Speech recognition, ChatGPT API Calls, and Text to speech processing.

2.2.1 PC

The PC accepts audio input from the microcontroller (streamed via WiFi), and constantly translates it to text in chunks, listening for the keyword, “Hey A.I.Dan” (pronounced Aidan). When it hears the phrase, the PC will listen to the next sentence it gets (waiting for a long pause in audio to stop listening), convert the data to text and send it to ChatGPT’s API.[1] The API text output is then transmitted as both text (to be displayed on the screen), and as audio (through a text-to-speech conversion) back to the microcontroller to be output to the user.

Requirement 1: The PC must maintain an open WiFi port for 1 hour, untouched, to listen for microcontroller input.

Requirement 2: The PC must be able to convert clear and simple speech into text using a pre-trained model with an accuracy rate >80%

Requirement 3: The PC must be able to analyze text to determine whether a question is valid (contains the wake word “Hey A.I.dan” or similar) with an accuracy rate >80%.

Requirement 4: The PC must be able to convert standard text into speech using a pre-trained model, understandable >80% of the time.

Requirement 5: The PC must be able to output both a digital audio signal and text to the ESP32 microcontroller through WiFi using TCP .

2.2.2 ChatGPT API

The ChatGPT API generates the ChatGPT response. It receives text input from the PC and outputs the response of chatGPT as text back to the PC.

Requirement 1: The API must be able to generate an answer to a text input with a text output within 4 seconds of the API call

2.3 Onboard Block

The onboard block takes in audio data through a microphone and transmits it through WiFi to the PC. It also receives data through WiFi to be output to a screen and through a speaker

2.3.1 Microcontroller

[1] https://platform.openai.com/docs/models/gpt-3

Consists of an ESP32 microcontroller with built in WiFi. It serves as the liaison between input and output data. Input data is received from the microphone and transmitted to the PC through WiFi. From the PC, it receives data back both in the form of text and audio. The microcontroller sends the audio directly to the speaker to be output to the user and converts the text input to visual and sends it to the screen through the SPI protocol.

Requirement 1: The microcontroller must be able to output digital signals, including an SPI interface (e.g. screen)

Requirement 2: The microcontroller must be able to host a TCP server using Wifi to interface with the PC.

2.3.2 Input Subsystem

The input subsystem contains the hardware necessary to accept user voice input using a microphone.. Data is output to the microcontroller.

2.3.3 Output Subsystem

The output subsystem displays data received from the microcontroller through a screen through the SPI protocol and through the speaker following a digital-to-analog conversion.

Requirement 1: The screen must be able to properly display code snippets, which requires at least 16-bit color.

2.3.4 Power:

The power subsystem powers all onboard components.

Requirement 1: Must be able to provide 5V power at a current of 1 A.

Power needed by component:

- *Screen: ~1 W* [2]
- *Microcontroller: ~2.50W based on 3.3V at 0.5A, using a voltage divider from our 5V in.* [3]

[2] Overestimate based on: https://www.orientdisplay.com/wp-content/uploads/2021/10/AFK320240A0-3.5N12NTH.pdf

[3] https://www.espressif.com/sites/default/files/documentation/esp32-c6_datasheet_en.pdf

- *Speaker: ~0.5W* [4]
- *DAC: <0.1W,* [5] *not significant for power consideration.*

2.4 Tolerance Analysis

Data Throughput (PC-Microcontroller data transfer)

The largest identified issue associated with this design is the data management on the microcontroller, and its interaction with the PC. Even the most robust microcontrollers only contain around 512 KiB of memory, which corresponds to roughly a second of audio information. Data will be streamed to the PC as fast as possible, and taking and running Text-To-Speech will act similarly quickly, but both streams could prove too much for the microcontroller.

To demonstrate feasibility, assume 100 KiB of memory allocated for each individual stream, up and down. Wifi speed is quite high relative to these numbers, so the actual communication shouldn't be an issue (ESP32-C3 supports 150mbps, and any PC we would be using will support at least 300mbps) Assume standard 44khz audio at 12 bits of digital resolution.

$$44000hz \cdot 12b = 528000\, b/s = 528\, Kib/s \tag{1}$$

A single audio stream comes out to 528 Kib/S. With the initial 100 KiB memory stream assumption, we get:

$$\frac{528\, Kib/s}{100\, Kib\ stream} = 5.28\ \text{refreshes/s} < 160\, mhz \tag{2}$$

That means we need to fully refresh and clear the memory 5 times over each second. This should be acceptable, given the SRAM's posted speed of 160mhz.

[4] Overestimate based on: https://www.jameco.com/z/36MS30008LF-PN-Jameco-ValuePro-30mm-PCB-Mount-8-Ohm-82dB-700-5kHz-Speaker_135722.html

[5] https://www.mouser.com/datasheet/2/256/MAX5250-1292624.pdf

3 Ethics and Safety

When AI solutions are being developed, ethical and safety concerns abound. Based on IEEE's Code of Ethics, there are three major ethical concerns that should be addressed with this design. The first, violating point I.2, is that AIs like ChatGPT are capable of misinformation, despite most users considering them infallible. To address this, we will be adding a disclaimer to the attached screen that displays on startup, informing the user of the limitations that an AI presents. That disclaimer will be something to the effect of "The responses provided by this machine are created automatically using an AI engine. These responses may be misleading or untrue. Do not rely on this device for medical or vital information." This disclaimer also addresses the second ethical issue, which is that some users may rely on the bot for life saving information, which it cannot guarantee is factual or helpful (which violates IEEE Code of Ethics point I.1). Hopefully by informing users of the limitations beforehand, these issues will be minimized.

One final ethical concern presented by this project is the issue of collecting user data, which also touches on IEEE's Code of Ethics point I.2. In order for this project to work, a microphone must be constantly on and recording the room around it, presenting a safety risk to users were that data to be leaked or tapped. While many companies use this data to reinforce their own learning models, we will be discarding it as soon as humanly possible to protect our users.