# PROBLEM STATEMENT 1

Offline, Privacy-Preserving Hindi Voice Assistant on Raspberry Pi

Introduction:

Agent M is a voice assistant that runs completely offline on a Raspberry pi 4 and accepts Hindi commands. Once set up, it is easy to use for the general user by giving voice commands to perform specific tasks. Users just need to say “Hey Mycroft” to wake up the voice assistant, and give the voice commands.

It uses its own external circuit to measure the temperature and humidity. It also uses the external circuit for other purposes, like turning a light bulb on or off. These additions to the voice assistant enable it to work without internet connection.

## 1. Methodology

a) In this project, we have utilized multiple speech to text libraries that serve different purposes for efficient, reliable and relatively low latency operation of the voice assistant.

- For wake word detection, openWakeWord(from dscripka/openWakeWord, on github) has been used with one of its default wake words("hey_mycroft"). We found it to be very reliable in our test runs. Hence, we decided to continue with it.
  However, the wake word can be easily changed as per the user's choice by using the custom wake word training procedure as described in the openWakeWord github repository(from dscripka/openWakeWord, on github).
- For the recognition of Hindi voice commands, we have implemented vosk(from alphacep/vosk-api, on github) and we are using the model called "vosk-model-small-hi-0.22".
- In the conversion of speech to text for song names in the "Play Music" feature, we have used faster-whisper(from SYSTRAN/faster-whisper, on github).

b) When wake word is detected, audio is captured at 48 kHz and then converted to 16 kHz for further utilization by vosk to listen for Hindi voice commands. Voice commands are recorded until silence is detected.

c) Once silence is detected, the Hindi voice command recorded is converted to text and matched with a pre-defined set of keywords to determine the intent of the speaker. Each 'task' that the assistant can perform has a set of keywords within a tuple called 'id_task' associated with it, and may have additional tuples like 'trigger_task', 'extra_task', etc. The input commands are matched with the various tuples and the task to perform is determined.

d)Once the 'task' is determined, the necessary parts of the program are executed and the output is given to the user through text and speech.

d) For text to speech conversion, piper-tts is used(from OHF-Voice/piper1-gpl, on github). We have used one of the available Hindi voices called "pratham".

## 2. Features of the voice assistant:

Note: Say “Hey Mycroft” to wake up the voice assistant.

i) Tells time: To know the time, ask: “kitne baje hai?”

ii) Tells date: To know the date, ask: ”tareekh kya hai?”

iii) Tells day of the week: To know which day of the week it is, ask: “kaunsa din hai?”

iv) Measures temperature and humidity: To know the temperature and humidity, ask: “tapmaan kitna hai?”

v) Records audio clips: To start recording audio from the microphone, say: “recording shuru karo”

To play the recently recorded file, say: “recording chalao”

vi) Reads task schedule: Reads the task schedules from the user created task schedule text file (format:’(dd, mm, yyyy,“Task”)’).

To know today’s tasks, ask: “aaj kya kaam karna hai?”.

To know tomorrow’s tasks, ask: “kal kya kaam karna hai?”

vii) Reminds user to drink water: To turn on reminder, say: ”paani peena yaad dilao”.

To turn off reminder, say: “paani peena yaad mat dilao”

viii) Turns on light through external circuit: To turn light on, say: “light jalao”.

To turn light off, say: “light bandh karo”

ix) Plays music: To play music, say: “gaana chalao”. The voice assistant will then ask for the name of the song to play. If a song in the local folder is found that matches enough with the said song name, the song will be played.

To pause music, say: “gaana rok do”.

To resume music, say: “gaana shuru karo”.

To stop music, say: “gaana bandh karo”.

To set volume, tell the volume amount in multiples of 10 Between 0 to 100 by saying: “Matra {amount}”. For example: To set the volume to 90%, say: “Matra nabbe”.

xi) Emergency signal: To call for help during emergencies, say: “Madad bulao”. The keyword “madad” is enough for the voice command to work.

NOTE: This command will put the program into an infinite loop, where it keeps on playing a chime and closes the circuit to the device connected to the external circuit’s emergency signal relay.

To reset the system, restart the program. Only closing the Program will still keep the relay in the same state.

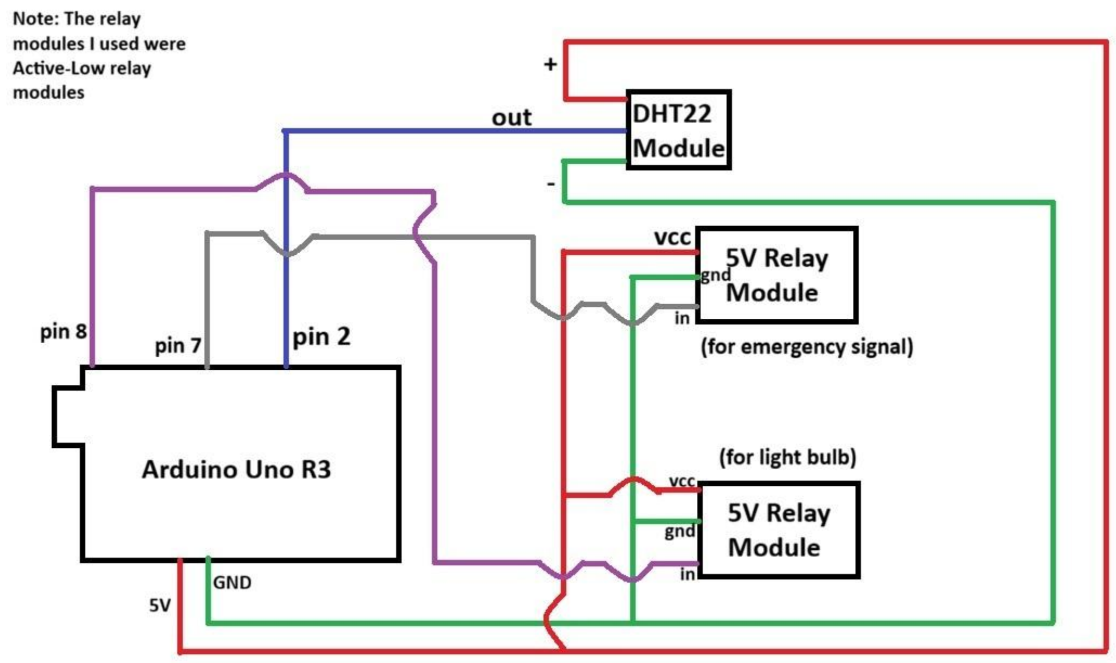
Once this command is detected, no other command will be accepted until the program is restarted.

Note: These commands will get recognized even with slight alteration, like “tapmaan kya hai” instead of “tapmaan kitna hai” because of the additional keywords that we have added for the same task.

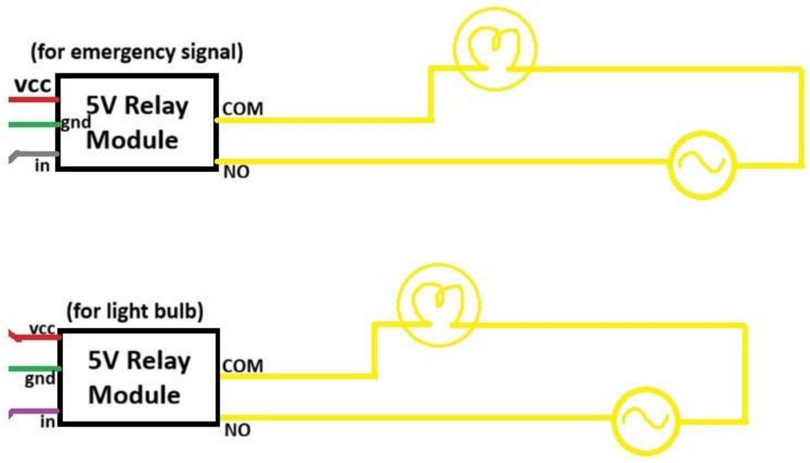
## 3. Major hardware used

- Raspberry pi 4 Model B 8GB RAM: The program was written and tested on the Raspberry pi 4 Model B 8GB RAM(with Raspberry Pi OS). Also, the board was enclosed in the official case along with the official cooling fan.
- Arduino Uno R3: Without internet, getting temperature/humidity information and controlling lightbulbs pose a problem. An Arduino Uno R3 has been used to measure the temperature and humidity by using a DHT22 sensor, and to control the lightbulbs through relay modules. Also, the Arduino Uno is required in this project for the "Emergency Signal Feature". Arduino Uno is connected via usb to the raspberry pi.
- An USB microphone to capture voice commands.
- Speakers with AUX input to play audio from the Raspberry pi.

Circuit Diagram 1



Circuit Diagram 2



Note: The relay modules I used were Active-Low relay modules

## 4. Optimization Techniques

- We have set the beam size of 1 for transcription using faster-whisper and used the “tiny” model with language as “en” for determining song names.
- The music playing functionality does not depend entirely on the faster-whisper model for determining the song to search for in the local folder containing music. The accuracy part is taken care by “rapidfuzz”(from rapidfuzz/RapidFuzz, on github). We use rapidfuzz to match the transcription result from faster-whisper using rapidfuzz’s weighted ratio to search for the song in the local folder. If a match above a certain threshold is found, the song is played.
  In this manner, by not relying entirely on the transcription result, we can use smaller models to reduce the time to play music accurately. The “en” selection does not limit the program to play English songs only, it can play Hindi songs too.

- By using multiple models- openWakeWord to detect wake word, vosk to recognize Hindi voice commands and faster-whisper for music searching, we managed to get a good balance of speed and accuracy with the features of our voice assistant.
- For most of the voice commands, the assistant was able to understand us. However, there were certain scenarios where the speech to text results were not accurate. To reduce the voice command rejection rates because of inaccurate speech to text conversion, words that sound similar to the spoken word have been intentionally added to the tuple of keywords.

## 5. System performance and Observations:

We evaluated the assistant based on how it performed during our test runs. Since we could not measure the exact millisecond speeds, we focused on how responsive the system felt during use.

- Responsiveness: The gap between finishing a sentence and hearing the assistant start to speak was short enough to make the conversation feel close enough to normal(except for transcription using faster-whisper for music searching where a larger and heavier model is used on purpose).
- Accuracy: For most of the voice commands, the assistant was able to understand us. However, there may be certain scenarios where the speech to text results are not accurate. We have taken measures to reduce voice command rejection by putting similar sounding words to the tuple containing keywords.
- Stability: The voice assistant program ran without any issues during our tests on our Raspberry pi 4 Model B, 8GB RAM for an extensive period of time.

## 6. Limitations:

1) This assistant is able to recognize most of the Hindi commands accurately, however, the words need to be spoken out loud and clear and is affected by heavy background noise

2) The voice commands should be spoken at a normal speaking rate. Speaking too fast leads to inaccurate speech to text conversions.

3) We used a single microphone via USB in our testing. However, commercially available voice assistant devices tend to use multiple microphones to detect the voice commands accurately. Moving to setup of multiple microphones can improve the performance a lot.

## 7. Instructions to build

1. Note: Directories are hard coded in the program. In our case, the username

   on Raspberry Pi OS is "pi". Hence, every instance of /home/pi will be needed to be modified according to the user in both the python program and lines below. Also, the sample rates selected have been found to work perfectly on our systems, however, it may have to be changed in the program by the user as per their system needs, or make necessary modifications to their systems.

   Also, using an arduino is necessary(please see the provided arduino code). Please make sure to upload the provided program to the Arduino UNO R3 and connect it to the raspberry pi via USB.

Run the following commands in terminal, when asked Y/n to install these packages, enter Y :

```
sudo apt update
sudo apt install portaudio19-dev
sudo apt install python3-all-dev
sudo apt install espeak-ng
sudo apt install libespeak1
cd /home/pi
mkdir voice_assistant
mkdir some_music
cd /home/pi/voice_assistant
mkdir task_schedule
mkdir sound_effects
mkdir sound_recordings
python3 -m venv voiceassistant
source /home/pi/voice_assistant/voiceassistant/bin/activate
pip install PyAudio
pip install openwakeword
pip install faster-whisper
pip install vosk
pip install pygame
pip install piper-tts
pip install rapidfuzz
pip install pyserial
```

2. In the folder home/pi/voice_assistant/sound_effects please keep three 1 second notification sounds as per your choice for ui sounds(.mp3 format only).

   Please make sure to name them "Menu Notification.mp3", "Menu Button.mp3" and "chime.mp3"

3. Users need to keep their tasks in the /home/pi/voice_assistant/task_schedule folder in a file called "task_schedule.txt" for the task reminder function to work in the following format:

   ```
   (dd, mm, yyyy, "task1")
   (dd, mm, yyyy, "task2")
   ```

4. Please keep the vosk model file, piper-tts .json and piper-tts .onnx files at /home/pi/voice_assistant. These files are available on their official github pages.

   Please follow the instructions in those repositories to get the files.

   We have used vosk model "vosk-model-small-hi-0.22"(instructions can be

   found at alphacep/vosk-api from github)

and voice called "pratham" from the set of piper-tts Hindi voices(instructions can be found at OHF-Voice/piper-gpl from github)

5. Open text editor and place save the following lines in a file named AgentM.sh :

```
#!/bin/bash

source /home/pi/voice_assistant/voiceassistant/bin/activate

python /home/pi/voice_assistant/voiceassistant.py
```

keep this file at /home/pi/voice_assistant and allow it to execute by following these steps:

i) right click the file

ii) click on properties

iii) go to permissions tab

iv) set the execute option to anyone

6. Open text editor and place save the following lines in a file named AgentM.desktop :

```
[Desktop Entry]

Type=Application

Name=Agent M

Exec=/home/pi/voice_assistant/AgentM.sh
```

```
Icon=python3
Terminal=true
```

keep this file at /home/pi and allow it to execute by following these steps:

i) right click the file

ii) click on properties

iii) go to permissions tab

iv) set the execute option to anyone

7. Please keep the music you want to play in /home/pi/some_music. It is recommended to keep .mp3 files.

8. Keep the python program at /home/pi/voice_assistant and save it as "voiceassistant.py"

8. Once all the above steps are followed, the voice assistant can be run by simply double clicking the AgentM.desktop file and clicking Execute.