**Weather Server Site Instruction**

**1.**

Connect raspberry pi pico to pc and DHT11 sensor to raspberry

Wiring : + to VBUS , - to GND, out to GP4

**2.**

Raspberry Coding:

Main.py

Dth.py

**3.**

Start puTTy server – Serial line – com5(check in device manager which port raspberry pi connected)

Logging file destination to

Select printable output

**4**

Assistant.py

Install virtual env Create a virtual environment (weatherAI) for AI weather assistant where code will exist but compressed it into AIAssistant.exe will be easy to access and portable

**5**

HTML files: 2

CSS files :

JS files:

1st for online weather 2nd  for offline weather

**6**

Install flask python

Put HTML CSS JS files in templates folder

Project\_weather.py

Setup everything write code and all script

Access sensor data from puTTy log file store in variable temp=temperature hum=humidity

In the offline webpage mention {{ temp}} {{ hum }} here python variable will come

Connect assistant.py or start assistant.py in sub process

**7**

**Start flask server and assistant**

**Or**

**Create batch file to** open Local host in browser , project\_weather.py and AI\_assistant.exe (virtual env)

**8**

**Open Start Weather.bat**

**Site will be hosted in local host**

http://127.0.0.1:5000/

**9**

**Done**

**10**

**Installation requirement**

puTTY for connection

Thonny IDE for raspberry pi pico coding

VS Code

Browser

Python Libraries

Use pip install {name}

* Flask
* pyttsx3
* webbrowser
* datetime
* speech\_recognition
* wikipedia
* webbrowser
* time
* python\_weather
* asyncio
* requests
* BeautifulSoup
* import random
* selenium
* webdrive

**11. Codes**

Main.py

from machine import Pin

import time

from dht import DHT11, InvalidChecksum

pin=Pin(25, Pin.OUT)

sensor = DHT11(Pin(4, Pin.OUT, Pin.PULL\_DOWN))

while True:

temp = sensor.temperature

humidity = sensor.humidity

print("Temperature: {}°C Humidity: {:.0f}% ".format(temp, humidity))

pin.toggle()

time.sleep(2)

dht.py

import array

import micropython

import utime

from machine import Pin

from micropython import const

class InvalidChecksum:

pass

class InvalidPulseCount(Exception):

pass

MAX\_UNCHANGED = const(100)

MIN\_INTERVAL\_US = const(200000)

HIGH\_LEVEL = const(50)

EXPECTED\_PULSES = const(84)

class DHT11:

\_temperature: float

\_humidity: float

def \_\_init\_\_(self, pin):

self.\_pin = pin

self.\_last\_measure = utime.ticks\_us()

self.\_temperature = -1

self.\_humidity = -1

def measure(self):

current\_ticks = utime.ticks\_us()

if utime.ticks\_diff(current\_ticks, self.\_last\_measure) < MIN\_INTERVAL\_US and (

self.\_temperature > -1 or self.\_humidity > -1

):

# Less than a second since last read, which is too soon according

# to the datasheet

return

self.\_send\_init\_signal()

pulses = self.\_capture\_pulses()

buffer = self.\_convert\_pulses\_to\_buffer(pulses)

self.\_verify\_checksum(buffer)

self.\_humidity = buffer[0] + buffer[1] / 10

self.\_temperature = buffer[2] + buffer[3] / 10

self.\_last\_measure = utime.ticks\_us()

@property

def humidity(self):

self.measure()

return self.\_humidity

@property

def temperature(self):

self.measure()

return self.\_temperature

def \_send\_init\_signal(self):

self.\_pin.init(Pin.OUT, Pin.PULL\_DOWN)

self.\_pin.value(1)

utime.sleep\_ms(50)

self.\_pin.value(0)

utime.sleep\_ms(18)

@micropython.native

def \_capture\_pulses(self):

pin = self.\_pin

pin.init(Pin.IN, Pin.PULL\_UP)

val = 1

idx = 0

transitions = bytearray(EXPECTED\_PULSES)

unchanged = 0

timestamp = utime.ticks\_us()

while unchanged < MAX\_UNCHANGED:

if val != pin.value():

if idx >= EXPECTED\_PULSES:

raise InvalidPulseCount(

"Got more than {} pulses".format(EXPECTED\_PULSES)

)

now = utime.ticks\_us()

transitions[idx] = now - timestamp

timestamp = now

idx += 1

val = 1 - val

unchanged = 0

else:

unchanged += 1

pin.init(Pin.OUT, Pin.PULL\_DOWN)

if idx != EXPECTED\_PULSES:

raise InvalidPulseCount(

"Expected {} but got {} pulses".format(EXPECTED\_PULSES, idx)

)

return transitions[4:]

def \_convert\_pulses\_to\_buffer(self, pulses):

"""Convert a list of 80 pulses into a 5 byte buffer

The resulting 5 bytes in the buffer will be:

0: Integral relative humidity data

1: Decimal relative humidity data

2: Integral temperature data

3: Decimal temperature data

4: Checksum

"""

# Convert the pulses to 40 bits

binary = 0

for idx in range(0, len(pulses), 2):

binary = binary << 1 | int(pulses[idx] > HIGH\_LEVEL)

# Split into 5 bytes

buffer = array.array("B")

for shift in range(4, -1, -1):

buffer.append(binary >> shift \* 8 & 0xFF)

return buffer

def \_verify\_checksum(self, buffer):

# Calculate checksum

checksum = 0

for buf in buffer[0:4]:

checksum += buf

if checksum & 0xFF != buffer[4]:

raise InvalidChecksum()

**Project\_weather.py**

from ssl import ALERT\_DESCRIPTION\_UNRECOGNIZED\_NAME

from flask import Flask,render\_template

import time

import datetime

import os

import python\_weather

import asyncio

import requests

from bs4 import BeautifulSoup

app=Flask(\_\_name\_\_,template\_folder='template')

@app.route("/")

def home():

#y=str(getweather())

with open("C:\\Users\\anura\\Desktop\\putty.log","r") as f:

first\_line=f.readline()

for line in f:

pass

last\_line = line

tempp=str(first\_line)

humm=str(last\_line)

return render\_template('index.html', temp=tempp,humm=humm)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

loop = asyncio.get\_event\_loop()

loop.run\_until\_complete(getweather())

**AI\_Assistant.py**

#import audioop

import pyttsx3

import webbrowser

import datetime

import speech\_recognition as sr

import wikipedia

import webbrowser

import time

#import keys

#import convo

import python\_weather

import asyncio

import requests

from bs4 import BeautifulSoup

import random

#extra

from selenium import webdriver

#import brian2

engine = pyttsx3.init('sapi5')

voices = engine.getProperty('voices')

#print(voices[1].id)

engine.setProperty('voice', voices[0].id)

# Set Rate

#engine.setProperty('rate', 190)

# Set Volume

#engine.setProperty('volume', 1.0)

# Set Voice (Female)

#brian2.test()

def speak(audio):

engine.say(audio)

engine.runAndWait()

def speak2(audio):

engine.say(audio)

engine.runAndWait()

def greetme():

#welcome message

hour = int(datetime.datetime.now().hour)

if hour >= 6 and hour < 12:

speak("Good Morning Sir")

elif hour >= 12 and hour < 16:

speak("Good afternoon Sir")

elif hour >= 16 and hour < 19:

speak("Good Evening Sir ")

tellTime2()

getweather()

#speak("I am Not Jarvis. How may I assist you Sir!!")

#speak("Sir!!")

def intro():

speak("I'm your personal weather assistant. Currently showing complete weather information")

speak("And if you need anything Just ask")

async def getweather():

# declare the client. format defaults to metric system (celcius, km/h, etc.)

client = python\_weather.Client(format=python\_weather.METRIC)

# fetch a weather forecast from a city

weather = await client.find("Greater Noida")

# returns the current day's forecast temperature (int)

x=print(weather.current.temperature)

speak("The Weather in Grater Noida is")

speak(weather.current.temperature)

speak("degree celcius")

# get the weather forecast for a few days

# close the wrapper once done

await client.close()

async def getweatherfull():

# declare the client. format defaults to metric system (celcius, km/h, etc.)

client = python\_weather.Client(format=python\_weather.METRIC)

# fetch a weather forecast from a city

weather = await client.find("Greater Noida")

# returns the current day's forecast temperature (int)

print(weather.current.temperature)

# get the weather forecast for a few days

for forecast in weather.forecasts:

print(str(forecast.date), forecast.sky\_text, forecast.temperature)

#speak(str(forecast.date), forecast.sky\_text, forecast.temperature)

# close the wrapper once done

await client.close()

def tellDay():

day = datetime.datetime.today().weekday() + 1

#this line tells us about the number

# that will help us in telling the day

Day\_dict = {1: 'Monday', 2: 'Tuesday',

3: 'Wednesday', 4: 'Thursday',

5: 'Friday', 6: 'Saturday',

7: 'Sunday'}

if day in Day\_dict.keys():

day\_of\_the\_week = Day\_dict[day]

print(day\_of\_the\_week)

speak("The day is " + day\_of\_the\_week)

def tellTime():

# This method will give the time

time = str(datetime.datetime.now())

# the time will be displayed like

# this "2020-06-05 17:50:14.582630"

#nd then after slicing we can get time

print(time)

hour = time[11:13]

min = time[14:16]

speak("Currently " + hour + "Hours and" + min + "Minutes")

def tellTime2():

from datetime import datetime

print(datetime.today().strftime("%I:%M %p"))

speak(datetime.today().strftime("It's %I:%M %p"))

def news\_now():

#url='https://www.bbc.com/news'

url='https://news.google.com/topics/CAAqJggKIiBDQkFTRWdvSUwyMHZNRGRqTVhZU0FtVnVHZ0pWVXlnQVAB?hl=en-IN&gl=IN&ceid=IN:en'

response = requests.get(url)

soup = BeautifulSoup(response.text, 'html.parser')

headlines = soup.find('body').find\_all('h3')

timeout = time.time() + 20

for x in headlines:

if time.time() > timeout:

break

print('=>')

print(x.text.strip())

speak(x.text.strip())

#query=takecommand().lower()

#if 'stop' in query:

# break

def dailynews():

speak("Sir Do you want the Daily News ")

query=takecommand().lower()

if 'yes' in query:

news\_now()

elif 'no' in query:

speak('ok')

def showtask():

speak("No task assigned for today")

def todaytask():

lst=["Do you want to checkout today's task sir?","let me remind you sir we have some tasks for today do you want to check them now or later"]

random.shuffle(lst)

for tasks in lst:

#print tasks

speak(tasks)

#speak("Do you want to checkout today's task sir?")

query=takecommand().lower()

if 'yes' in query:

showtask()

elif 'no' in query:

speak('ok')

def takecommand():

r = sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

r.pause\_threshold = 2

audio = r.listen(source)

try:

print("Recognizing...")

query = r.recognize\_google\_cloud(audio, language='en-in') #Using google for voice recognition.h

print(f"User said: {query}\n") #User query will be printed.

except Exception as e:

# print(e)

print("Say that again please...") #Say that again will be printed in case of improper voice

return "None" #None string will be returned

return query

#extra

def search\_web():

driver = webdriver.Firefox()

driver.implicitly\_wait(1)

driver.maximize\_window()

if 'youtube' in query:

speak("Opening in youtube")

indx = takecommand.lower().split().index('youtube')

query=takecommand.split()[indx + 1:]

#query = input.split()[indx + 1:]

driver.get("http://www.youtube.com/results?search\_query =" + '+'.join(query))

return

else:

speak("unable to find")

def temp():

speak("hello = haha")

#search\_web()

if \_\_name\_\_=="\_\_main\_\_" :

#speak("Logging in ")

#speak("Enter security key")

#query=takecommand().lower()

#if '51' in query:

#greetme()

#tellTime(self)

#temp()

#exit()

loop = asyncio.get\_event\_loop()

x=loop.run\_until\_complete(getweather())

#x=int(x)

#intro()

#time.sleep(3)

#dailynews()

#todaytask()

while True:

query=takecommand().lower()

#query

if 'wikipedia' in query:

speak('Results from Wikipedia Sir!')

query = query.replace("wikipedia", "")

results = wikipedia.summary(query, sentences=2)

speak("According to Wikipedia")

print(results)

speak(results)

elif 'open youtube' in query:

webbrowser.open("youtube.com")

elif "which day it is" in query:

tellDay()

elif "today's day" in query:

tellDay()

elif 'tell me the time' in query:

tellTime()

elif 'time right now' in query:

tellTime()

elif 'time' in query:

tellTime()

elif 'complete weather' in query:

getweatherfull()

elif 'yes' in query:

speak("can you repeat? sir,")

elif 'temperature' in query:

with open("C:\\Users\\anura\\Desktop\\putty.log","r") as f:

first\_line=f.readline()

for line in f:

pass

last\_line = line

tempp=str(first\_line)

humm=str(last\_line)

speak(tempp)

elif 'humidity' in query:

with open("C:\\Users\\anura\\Desktop\\putty.log","r") as f:

first\_line=f.readline()

for line in f:

pass

last\_line = line

tempp=str(first\_line)

humm=str(last\_line)

speak(humm)

elif 'date' in query:

x = datetime.datetime.now()

print(x.date)

speak(x.date)

elif 'who are you' in query:

speak("Hello Sir, I'm personal weather assistant i displays and tell complete weather information I am your morning star if you need anything Just ask")

elif 'Introduce yourself'in query:

speak("Hello Sir, I'm personal weather assistant i displays and tell complete weather information I am your morning star if you need anything Just ask")

elif 'stop' in query:

exit()

elif 'jarvis logout' in query:

speak('sir ? sir? logging off sir')

exit()

elif 'hey jarvis' in query:

speak("Yes Sir!! i'm here waht do you need")