```
"""Script to store hotwire data in a more convenient .mat format."""

from scipy.io import savemat

ANGLES = ["-5", "00", "05", "20"]

FOLDER = "Hotwire Measurements"

FN_PREFIXES = {
    ANGLES[0]: "hotwire_mes_a_neg_5",
    ANGLES[1]: "hotwire_mes",
    ANGLES[2]: "hotwire_mes_a_5",
    ANGLES[3]: "hotwire_mes_a_20"
}

N_DATAPOINTS = {
    ANGLES[0]: 11,
    ANGLES[1]: 10,
    ANGLES[2]: 15,
    ANGLES[3]: 78,
}

USER_PREFIXES = {
    ANGLES[0]: "a_neg_5_",
    ANGLES[1]: "a_0_",
    ANGLES[2]: "a_5_",
    ANGLES[3]: "a_20_"
}

SRC_FNS = {
    angle: {
        "dat": [f"./{FOLDER}/a_{angle}/{FN_PREFIXES[angle]}-{i+1}.dat" for i in
range(N_DATAPOINTS[angle])],
        "txt": [f"./{FOLDER}/a_{angle}/{FN_PREFIXES[angle]}-{i+1}-1.txt" for i in
range(N_DATAPOINTS[angle])],
    }
    for angle in ANGLES
}

KEYS = {
    "Static Pressure": "P",
    "Density": "rho",
    "Fixed Pitot Probe Speed": "v",
    "AOA": "a",
    "User comment": "y",
    "Temperature": "T",
}

def main():
    """Parse .dat and .txt files for each angle and combine relevant info into a .mat file."""
    for angle in ANGLES:
        dat_fns = SRC_FNS[angle]["dat"]
        txt_fns = SRC_FNS[angle]["txt"]

        for i, fn in enumerate(dat_fns):
            with open(fn, "r") as f:
                lines = f.readlines()
                data = {}
                for line in lines:

                    # Parse data key
                    split_char = "=" # Most values use '='
                    if len(line.split(split_char)) == 1:
                        split_char = ":" # user comment uses ':'
                    key = line.split(split_char)[0].strip()

                    if key in KEYS.keys():
```

```
                        try:
                            val = line.split(split_char)[1].replace("degrees (inclinometer)",
"").strip() # Remove junk
                            val = val.split("\t")[0] # Remove tabs if present
                        except Exception:
                            val = None
                        data[KEYS[key]] = comment_to_distance(val, angle) if key == "User
comment" else float(val) # Create dict with symbols

            # Parse txt file with hotwire voltages
            with open(txt_fns[i], "r") as f:
                lines = f.readlines()
                voltages = []
                for line in lines:
                    try:
                        voltages.append(float(line.split("\t")[1])) # Get voltages
                    except Exception:
                        continue
            data["V_arr"] = voltages

            dest_fn = f"./{FOLDER}/a_{angle}/data_{i+1}.mat"
            savemat(dest_fn, data)


def comment_to_distance(comment: str, angle: str):
    """Turn the user comment into a numerical distance."""
    val = comment.replace(USER_PREFIXES[angle], "").replace("in","") # Get just the numbers
    val = val.split("_") # Split fraction into parts

    ret = float(val[0])
    if len(val) == 3: # Do fraction math if needed
        ret += float(val[1]) / float(val[2])

    return ret

if __name__ == "__main__":
    main()
```