**Strassham Ampel- und AWOS Wetterstation**

Im Hintergrund läuft weather\_station.py mit dem Parameter der Empfindlichkeit für das Eingangssignal am Arduino (Lautstärke Funkgerät).

Das Programm **weather\_station.py** übergibt den "Empfindlichkeitsparameter" über die serielle Schnittstelle an den Arduino.

Der Arduino Prozessor liest diese ein und verwendet den Wert als Schwellwert für die Lautstärke des Lautsprecher Eingangssignales vom Funkgerät.  
Der Wert muss zwischen 199 und 1024 liegen.  
Der Arduino wertet die Anzahl der Klicks (empfangen in 5 Sekunden) aus und gibt diese zurück per serieller Schnittstelle an den rPi.  
Der Arduino empfängt nur, wenn das RTS/CTS Signal am Digital-Eingang 5 aktiv ist (HIGH), ansonsten überspringt er die Leseschleife.

Abhängig von der Anzahl der erkannten Klicks (z.B. 3 Clicks, 5 Clicks, 7 Clicks) wird am rPi die entsprechende Aktion ausgeführt:

**3 Klicks:**

Das Unterprogramm 433switch.py im Ordner "traffic\_light" schaltet die Ampel für die Dauer des Kommandozeilenparameters ein und dann wieder aus.  
Wird kein Parameter übergeben, so wird die Ampel für 180 Sekunden (3 Min) auf rot geschaltet.  
Die Ampel wird per Funkverbindung über eine schaltbare Steckdose (Brennerstuhl) aus- und ein geschaltet: 'rpi-rf\_send‘.   
Doku dazu im Ordner "433RC im Home-Directory von pi. / Der Code zum Schalten der Steckdose wird von der mitgelieferten Fernbedienung gelernt (receive.sh).  
Das Programm 433switch.py informiert beim Ein- und Ausschalten der Ampel per Funkspruch (enablePTT.py, activate-light.mp3...).

**5 Klicks:**

weather\_station.py startet das Python Programm ecowitt\_weather.py **ecowitt\_weather.py** holt die aktuellen AWOS Informationen (Wind, Luftdruck, Temperatur) über eine API aus dem Internet von der ecowitt Wetterstation in Strassham.

Die empfangenen Wetterdaten werden umcodiert und in der Textdatei ~/weather\_station/ramdisk/weather.txt abgelegt.  
Wegen der Eigenheiten von gTTS müssen die Wetterwerte etwas anders als gewohnt codiert werden (Englisch 0 = Zero anstatt ‚oh‘, 6 = sex, etc.).  
Anschließend werden die Text-Wetterdaten per Google Text To Speach (gTTS) in eine Audio-Datei umgewandelt: ~/weather\_station/ramdisk/weather.mp3  
Nach Rückkehr des Programms wird die mp3 Datei in weather\_station.py über den Audio-Ausgang ausgegeben.

**7 Klicks:**

Weitere Optionen sind noch nicht implementiert, bisher nur Platzhalter.

**Sprachausgabe am Funkgerät**

Die Sprechtaste des Funkgerätes wird über ein Relais aktiviert. Die Kabelverbindungen zwischen Arduino und rPi müssen entsprechend gesteckt werden.

Die Lautstärke des empfangen Funkgerätes ist auf max. Laut einzustellen.

**Logfile-Ausgaben** sind temporär im RamDisk-Verzeichis $HOME/weather\_station/ramdisk abgelegt.  
Sie werden am Ende des Tages komprimiert verschoben nach $HOME/weather\_station/logs

Beim **Systemstart** (/etc/rc.local) wird das Programm weather-station.py mit dem Parameter der Empfindlichkeit gestartet.

Die System-Crontab rebootet den rPi morgens um 6 Uhr.

Um 22:01 Uhr wird der Prozess "weather\_station.py" beendet (pkill).

Um 22:05 Uhr wird der Arduino abgeschaltet (die USB-Schnittstelle).

Um 22:07 Uhr werden die Logfiles von der ramdisk komprimiert in den Ordner ~/weather\_station/logs geschoben.

Das System kann per Taster (am Gehäuse) ein und ausgeschalten werden.

**Hauptverzeichnis**: $HOME

Hauptbenutzer: pi (bzw.pilot) / Passwort: StrasshamAWOS

**Brennerstuhl-Fernbedienung:**

Mit der Fernbedienung der funkgesteuerten Steckdose kann folgendes aktiviert werden:

* A/on: Einschalten der Ampel für 3 Minuten (via rPi)
* A/off: Einschalten der Ampel für 5 Minten (via rPi)
* B/on: Ampel dauerhaft ein (direkte Funk-Verbindung zur Steckdose)
* B/off: Ampel aus (direkte Funk-Verbindung zur Steckdose)
* D/off: Herunterfahren des rPi Computers