



Projekt Wetterstation MFCR

Es besteht der Wunsch nach einer vereinseigenen Webcam und Wetterstation, wobei die entsprechenden Daten auf unserer Webseite www.mfc-rossendorf.de/wichtige-infos-bitte-lesen/webcam/ direkt dargestellt werden sollten. Da unser Modellflugplatz keine Infrastruktur wie Strom, Telefon oder Internet hat, bleibt nur eine solarbetriebene Anlage.

Ein Computer Modul sammelt die Daten angeschlossener Sensoren (Live-Bild, Temperatur, Wind (Stärke + Richtung), Feuchte etc.), speichert sie lokal und sendet sie zyklisch an unseren Webserver. Zunächst ist eine stationäre Energieversorgung geplant (5V, USB), später eine autarke (Solarzelle + Akku).

Folgende Hardware kommt zum Einsatz:

- Raspberry Pi (v4)
- diversere Sensoren (via I2C/SPI)
- Webcam (via Raspi on-board Camera Connector)
- UMTS-Modul (via USB)
- Akku (LiPo)
- Solarzelle
- Lade-Management
- ggf. externer Wake-Up-Timer
- Gehäuse

Auf dem Raspi läuft eine Standard-Linux-Distribution (z.B. Raspbian). Die auszuführenden Aktionen werden in definierten Zeitabständen (wahrscheinlich stündlich) per CronJob angestoßen. So lange eine stationäre Stromversorgung gegeben ist, läuft das System ununterbrochen. Im autarken Regime (also „im Feld“, in unserem Fall auf dem Flugplatz) wird eine Lösung angestrebt, bei der der Raspi in der Zwischenzeit nicht aktiv ist. Dies ist momentan Raspi-seitig nicht möglich (keine Sleep-Funktion), es müsste also wahrscheinlich ein externer Wake-Up-Timer verwendet werden.

Die auszuführenden Funktionen sollten in Python umgesetzt werden. Die Sensordaten werden über die entsprechenden IO-Pins entweder per I2C oder SPI gesammelt. Die Webcam ist per on-board Connector angeschlossen, die Bildakquise erfolgt über die entsprechenden Kommandozeilen-Funktionen. Als Funkmodul kommt ein UMTS-Stick zum Einsatz. Systemseitig stellt sich dies als normales Netzwerk Device dar.

Webserver

Die Visualisierung der Werte soll auf unserer Webseite erfolgen. Die Daten sollen direkt an den entsprechenden Webserver gesendet und dort in einer Datenbank gespeichert und zur Anzeige gebracht werden. Über einen separaten „Admin-Bereich“ soll es möglich sein, weitere Informationen zum Systemzustand abzufragen (z.B. Akkuladestand) und den Raspi beim nächsten „Aufwachen“ in einen Wartungsmodus zu versetzen – d.h., er bleibt bis auf weiteres eingeschaltet und erlaubt Verbindungen (z.B. via ssh) von außen.

Wir verwenden ein Typo3-CMS. Es wäre zu prüfen, ob sich dessen Datenbank (mysql) für unsere Zwecke mitverwenden lässt oder ob eine separate DB-Installation sinnvoll wäre.

Für alle Messwerte gilt, dass sie zur Anzeige dynamisch sortiert und gefiltert und in grafischer Form (als Diagramm) auch als Überlagerung verschiedener Wert angezeigt werden können sollen. Dabei sollen alle Aktionen des Nutzers, die die Auswahl der anzuzeigenden Daten betreffen (z.B. Zeitraum) „dynamisch“ erfolgen – ein „Neu laden“ der Seite soll also nicht erforderlich sein, die Abfrage der DB und Aktualisierung der Darstellung soll via JS (AJAX) im Browser erfolgen. Die Werte sollen durch ein „passendes“ Verfahren interpoliert werden, so dass eine sinnvolle Darstellung auch durch die Messungen nicht erfasster Werte möglich ist (Darstellung als „Kurvenzug“).

Die von der Kamera aufgenommenen Bilder sollen übersichtlich als „Galerie“ entsprechend der Auswahl des Nutzers (Zeitraum, Intervall – z.B. „alle aufgenommenen Bilder der letzten Woche“ oder „alle Bilder des letzten Monats, die Mittags um 12 entstanden sind“) dargestellt werden, bei Auswahl eines Bildes soll dies in voller Auflösung zur Anzeige gebracht werden. Die Interaktion des Nutzers mit dem Browser soll ebenfalls „dynamisch“ – wie im vorigen Punkt beschrieben – erfolgen.

Eine Nutzer/Rechtemanagement ist nicht erforderlich, lediglich der oben genannte „Admin-Bereich“ soll nur nach einem Login erreichbar sein.

Die webserverseitige funktionelle Umsetzung ist noch nicht spezifiziert. Wenn sie im Rahmen des CMS (Typo3) erfolgen soll, könnte dies mittels PHP geschehen. Letztlich ist eine Vielzahl von Sprachen geeignet, vorzugsweise sollte die Umsetzung jedoch in einer Skriptsprache erfolgen, um die spätere Wartbarkeit und Erweiterbarkeit zu vereinfachen.

Browser/Frontendseitig kommen die „Standardtechniken“ HTML, CSS und JavaScript zum Einsatz. Hinsichtlich der Verwendung spezieller JS-Frameworks (z.B. jQuery) gibt es keine Festlegungen.

Betreuer des Projektes:

Thomas Brenner (Vorsitzender des Vereins MODELLFLUGCLUB ROSSENDORF e.V.)

Prof. Dr. rer. nat. habil. Heino Iwe

h.iwe@t-online.de
0351 2691704