Die Software

Die Software des Gerätes – des Datenloggers – erfüllt zwei Funktionen:

1. Der ESP32-Prozessor kommuniziert mit dem 3 Sensoren (CO_2, Temperatur und Luftfeuchtigkeit)
2. Der Prozessor fungiert als ein Access Point, der einen WLAN zur Verfügung stellt, mit dem sich die in der Reichweite befindenden Smartphones verbinden können. Auf dem Datenlogger läuft auch die Webserver-Software, deren Aufgabe ist es, die aufgezeichneten Messdaten zur Verfügung zu stellen. Wenn auf dem Smartphone eine entsprechende von uns entwickelte App läuft, können die Messdaten auf dem Smartphone dargestellt werden.

Die Software des Projektes hat also zwei Teile:

- Auf dem Datenlogger läuft die Server-Software
- Auf den Smartphones läuft die Client-Software

Folgendes (vereinfachtes) Flussdiagramm erleuchtet die Funktion der auf dem Datenlogger laufenden Software.

Die Software für Mikrocomputer besteht in der Regel aus

- einem Hauptprogramm (im Bild blau), das mehrere
- Unterprogramme (je nach Programmiersprache genannt auch Routinen, Prozeduren, Funktionen) aufrufen kann (im Bild der Einfachheit halber weggelassen) und
- aus sogenannten **Interrupt Service Routinen (ISR)** (im Bild grün)

Was ist ein Interrupt und eine ISR?

In Anwendungsbereichen „Berechnungen“ oder „Datenverarbeitung“ hängt die Ablaufsteuerung eines Programms nur vom Zustand der eigenen Variablen ab. In Mikrocontrolleranwendungen können zusätzlich äußere, nicht vorhersehbare, durch die Hardware verursachte Ereignisse kommen, auf die das Programm reagieren muss. Diese Ereignisse verursachen sog. Unterbrechung (engl. Interrupt). Das laufende Programm wird unterbrochen, die Zustände seiner Variablen zwischengespeichert und der Prozessor beginnt mit einer neuen Tätigkeit – er behandelt das ankommende Ereignis. Er macht also eine Dienstleistung, engl. Service – er startet eine entsprechende Interrupt Service Routine. Wenn die ISR fertig ist, kommt die Programmablaufsteuerung zurück an die Stelle, wo das Programm unterbrochen wurde.

Bei Mikrocontrolleranwendungen hat das Hauptprogramm in der Regel 2 Teile:

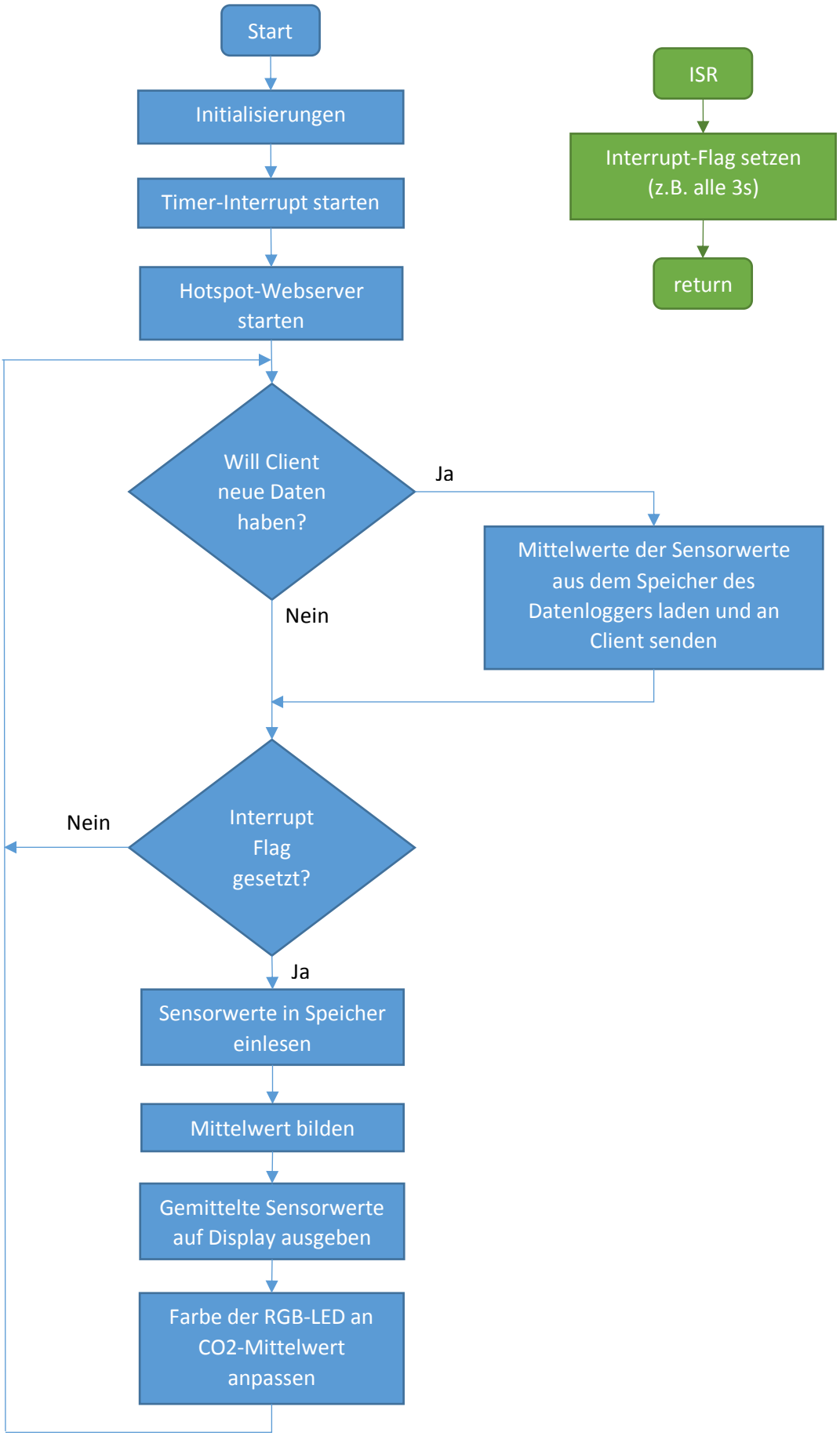
- Initialisierung (= Einstellung der gewünschten Modi der Hardwarekomponenten)
- Eine Endlosschleife (= alle sonstige sich wiederholende Tätigkeiten)

Aus der Endlosschleife kommt man nur durch das Abschalten des Gerätes raus.

Diese 2 Teile hat auch die Software unseres Datenloggers.

Der ESP32 Prozessor verfügt über mehrere Interrupts, von denen in unserer Anwendung nur das sog. Timer-Interrupt verwendet wird. Ein Timer ist eine Hardware-Komponente des Prozessors, die in bestimmten mit der Initialisierung einstellbaren Zeitintervallen ein Timer-Interrupt auslöst. Wenn dieser kommt, startet die ISR des genannten Timers.

Das Projekt Green Nose



Das Projekt Green Nose

Die Client-Server Kommunikation funktioniert wie folgt:

In vordefinierten Zeitintervallen sendet die auf dem Smartphone laufende Client-Software eine Anfrage an den Server, dass neue Messdaten verlangt werden. Wenn diese von der auf dem Datenlogger laufenden Server-Software erkannt wird, wird erstmal geprüft, ob die richtige Zeit dazu schon gekommen ist. Wenn ja, dann wird von den 3 Sensoren (CO_2, Temperatur, Luftfeuchtigkeit) je ein neuer Wert in einen speziellen Speicher des Datenloggers gespeichert.

Für diejenigen von Euch, die bereits die Grundlagen des Programmierens kennen, ist dieser Speicher als ein Array mit 10 Elementen organisiert. Und zwar so, dass, wenn ein neuer Wert reinkommt, dann werden alle übrigen Werte um eine Position verschoben. Der älteste Wert fliegt raus. Solche Datenstrukturen nennen wir Schieberegister oder auch FIFO-Speicher (FIFO = first-in-first-out). Dadurch wird sichergestellt, dass sich in diesem Array immer die 10 neuesten Werte befinden. Von diesen 10 Werten wird der Mittelwert gebildet und dieser wird anschließend auf dem Display ausgegeben und die RGB-LED ändert ihre Farbe entsprechend. Gleichzeitig stehen diese 10 neuesten Werte dem Client zur Verfügung.

In welcher Sprache und mit welcher Entwicklungsumgebung wird die Software des Datenloggers programmiert?

Im Rahmen des GreenNose-Projektes stellen wir Euch eine funktionierende Software zur Verfügung. Bei Interesse könnt Ihr aber das Programm anpassen und eigene Ideen implementieren. Natürlich stehen wir dazu beratend zur Verfügung.

Der ESP32-Prozessor wird in der Sprache C programmiert. Für alle verwendeten Komponenten stehen entsprechende Bibliotheken zur Verfügung.

Als Entwicklungsumgebung (IDE, Integrated Development Environment) kann man Arduino IDE nehmen. Dann sieht es ungefähr so aus: