Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik

Prof. Dr. Ulrich Schäfer

Projektarbeit Mobile & Ubiquitous Computing SoSe 2024

Gruppe 42: Johannes Grosch & Daniel Hartl

Studiengang Industrie-4.0-Informatik

25. Juni 2024

Inhaltsverzeichnis

1 Einleitung

Hier starten Sie mit Ihrer Beschreibung.

Bitte passen Sie in der LaTeX-Datei `projektarbeit.tex` am Anfang folgende Zeilen an:

Diese Literaturquellen sollten durch passende ersetzt werden: [1, 2, 3, 4, 5].

Sie können die Dokumentstruktur ändern. Dies ist lediglich ein Vorschlag.

Bei Heise (dies ist ein link) finden Sie eine Zusammenstellung, wie Sie eine LaTeX-Entwicklungsumgebung installieren, falls Sie nicht Overleaf benutzen wollen.

2 Projektplanung und Vorgehen

2.1 Serielle Kommunikation

Die Serielle kommunikation zwischen dem ESP und dem Laptop findet mit 10Hz statt. Um unnötigen Overhead zu vermeiden findet sie in folgender, minimalistischer Synthax statt:

<Pulsschlag>;<Sauerstoffsättigung>

Hier sollten Sie auch das Protkoll/Datenformat für die serielle Kommunikation zwischen ESP32 und PC/Laptop sauber dokumentieren.

3 Implementierung

3.1 Änderungen an der MAX30100 Bibliothek

Zur effizienteren Bedienung des Puls Oximeters wurde das Enumerate „1“ hinzugefügt, das für den Wert 0 die Konstante ESP_32 und für den Wert 1 LOLIN_32 zuordnet.

In der Klasse PulseOximeter selbst wurde die Signatur der Methode begin() um den Parameter „1“ modifiziert:

```
bool begin(BoardType board, PulseOximeterDebuggingMode debuggingMode_);
```

Listing 1: Signatur der begin() Methode

Hierbei sind beide optional und board hat als Standardparameter ESP_32, was den bisherigen Ablauf anstößt.

Sollte die Methode mit dem Parameterwert 1 oder LOLIN_32 aufgerufen werden, so werden die Konstanten für den SDA Pin auf 0 und die für den SCL Pin auf 4 gesetzt.

3.2 ESP32 Code

Der ESP32 wurde so programmiert, dass er nach CALLBACK_INTERVAL Millisekunden über die Serielle Schnittstelle mit 115200 Baud, wie in 2.1 beschrieben, die Herzfrequenz und den Sauerstoffgehalt des Blutes zur weiteren Verarbeitung an den Laptop überträgt, wo diese dann weiterverarbeitet werden können.

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"

// reporting interval 100ms
#define CALLBACK_INTERVAL 100

// driver entity
PulseOximeter oximeter;

// timestap of the last output
int lastOutputTimestamp;

void setup() {
	// intit the serial interface for 115200 Baud
	Serial.begin(115200);
	while(!Serial);

	// try to init the oximeter and count the failed tries and delay 500ms.
	Serial.println("Begin initializing the oximeter");

	int initFailCounter = 0;

	while (!oximeter.begin(ESP_32)) {
		Serial.print("Init failed ");
		Serial.print(++initFailCounter);
		Serial.println(" times!");
		delay(500);
	}

	Serial.println("Oximeter initialized");
}

void loop() {
	// get values from oximeter.
	oximeter.update();

	// print heartrate and o2 ratio
	Serial.print(oximeter.getHeartRate());
	Serial.print(";");
	Serial.println(oximeter.getSpO2());

	delay(CALLBACK_INTERVAL);
```

```
}
```

Listing 2: Quellcode des ESP32-Programms

4 Probleme und Diskussion

Sie können die Überschrift natürlich umbenennen, falls es keine Probleme gibt :-).

5 Zusammenfassung und Ausblick

Die Literatur (für den nächsten Abschnitt) tragen Sie in die Datei `quellen.bib` ein. Diese Datei hat die BiBTEX-Syntax (vgl. `https://ctan.org/pkg/bibtex`; BiBTEX ist bei TEXstudio bzw. Overleaf dabei).

Zur Literaturverwaltung verwenden Sie am besten Zotero [5] oder JabRef [2].

Bei meiner LATEX-Installation ist der `natdin`-Style leider etwas buggy (bei `@book` wird die DOI-URL doppelt generiert, dafür fehlt sie bei `@inproceedings`). Allerdings funktioniert das Attribut `lastchecked` für Webseiten (Publikationstyp `@misc`).

Literatur

[1] Breymann, Ulrich: *C++ programmieren: C++ lernen professionell anwenden – Lösungen nutzen. Aktuell zu C++20*. Carl Hanser Verlag GmbH & Co. KG, 2020. http://dx.doi.org/10.3139/9783446465510. http://dx.doi.org/10.3139/9783446465510. – ISBN 978–3–446–46551–0. – 6., überarbeitete Auflage

[2] Jabref.org: *JabRef – Stay on top of your literature*. https://www.jabref.org, Abruf: 19.05.2024. – JabRef is an open-source, cross-platform citation and reference management tool

[3] Schäfer, Ulrich ; Spurk, Christian: TAKE Scientist's Workbench: Semantic Search and Citation-Based Visual Navigation in Scholar Papers. In: *Fourth IEEE International Conference on Semantic Computing*. Los Alamitos, CA, USA : IEEE Computer Society, September 2010. – ISBN 978–0–7695–4154–9, S. 317–324

[4] Voss, Herbert: Flotter Oldie: (La)TeX – 25 Jahre und kein Ende. In: *c't* 9 (2005), S. 172–175

[5] Zotero.org: *Zotero – Your personal research assistant*. https://zotero.org, Abruf: 19.05.2024. – Zotero is a free, easy-to-use tool to help you collect, organize, annotate, cite, and share research