

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	2
2	Projektplanung und Vorgehen	2
3	Implementierung	2
3.1	Änderungen an der MAX30100 Bibliothek	2
3.1.1	PulseOximeter	2
4	Probleme und Diskussion	2
5	Zusammenfassung und Ausblick	2
	Literatur	4

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 Syntax statt:

<Pulsschlag>;<Sauerstoffsättigung>

Hier sollten Sie auch das Protokoll/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_);
```

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.

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 BiB_TE_X-Syntax (vgl. <https://ctan.org/pkg/bibtex>; BiB_TE_X ist bei T_EXstudio bzw. Overleaf dabei).

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

Bei meiner L^AT_EX-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>, Ab-ruf: 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>, Ab-ruf: 19.05.2024. – Zotero is a free, easy-to-use tool to help you collect, organize, annotate, cite, and share research