

Dokumentation: DigitalHeartMonitor

Julia Ohlhöft¹, Thore Hasselbring¹, und William Klaffke¹

¹*TH Lübeck*

Stand: 9. Februar 2022

Inhaltsverzeichnis

1	Einleitung	1
2	Programmteile	1
2.1	<code>#include</code> - Direktiven	1
2.2	Globale Variablen	1
2.3	<code>void setup()</code>	2
2.4	<code>void loop()</code>	2

1 Einleitung

2 Programmteile

2.1 `#include` - Direktiven

Das Programm benötigt drei Header-Bibliotheken:

- `Arduino.h`: nur, wenn statt der Arduino-IDE eine andere Entwicklungsumgebung verwendet wird.
- `Wire.h`: mglw. für `LiquidCrystal.h` notwendig; kann durch `#ifndef`-Guards eingeschränkt sein.
- `LiquidCrystal.h`

2.2 Globale Variablen

- `AnalogPin A0`: definiert Pin A0 als Eingang.

- `TimeStep 1000`: definiert einen Zeitschritt von 1000 ms pro Durchlauf; kann angepasst werden.
- `int lowerlimit = 250; int upperlimit = 750`: Werte für die obere oder untere Grenze; konservativ, kann angepasst werden.
- `int highs, lows = 0`: Anzahl der Limitüberschreitungen pro Zeiteinheit

2.3 `void setup()`

In der `setup`-Routine wird nur die serielle Schnittstelle und das LCD-Display initialisiert.

2.4 `void loop()`

Die `loop`-Funktion läuft ständig.

Zunächst wird ein `uint32_t` (vorzeichenloser 32 Bit-Long) `start` mit der derzeitigen Prozessorzeit `millis()` definiert. Mit diesem wird in der folgenden Schleife die Laufzeit kontrolliert.

Die Funktion `setLimits()` legt die obere und untere Grenze der aktuellen Pulsschlagsequenz fest. Anhand dieser Grenzen wird durch die Funktion `DiffS()` die Anzahl der oberen und unteren Halbwellen für jeden Schlag gezählt. Die Zählung erfolgt für eine durch eine `do-while`-Schleife getimte Zeitdauer.