

FSN

Pflichtenheft

Visualisierung von Prozessdaten zum Thema „Erneuerbare Energien und IoT“

Burglechner, Leeb, Steiner

Impressum

Herausgeber

Burglechner, Leeb, Steiner

Version	Stand	Status
1.1	18.09.2017	Erstellung
1.2	20.09.2017	Bearbeitung
1.3	21.09.2017	Bearbeitung
1.4	25.09.2017	Bearbeitung
1.5	30.09.2017	Bearbeitung

Änderungshistorie

Version	Datum	Bearbeiter	Aktivität / Kommentar
1.1	18.09.2017	Leeb, Steiner	Erstellung
1.2	20.09.2017	Steiner	Bearbeitung
1.3	21.09.2017	Steiner	Bearbeitung
1.4	25.09.2017	Burglechner, Leeb, Steiner	Bearbeitung
1.5	30.09.2017	Burglechner, Steiner	Bearbeitung

Inhaltsverzeichnis

1	Zielbestimmungen	5
1.1	Ausgangssituation	5
1.2	Musskriterien.....	5
1.3	Wunschkriterien	5
1.4	Abgrenzungskriterien	5
2	Produkteinsatz.....	6
2.1	Anwendungsbereiche	6
2.2	Zielgruppe.....	6
2.3	Voraussetzungen	6
2.3.1	Anwendungsvoraussetzungen	6
2.4	Betriebsbedingungen.....	6
3	Produktumgebung	7
3.1	Software	7
3.2	Hardware.....	7
3.3	Produktschnittstelle.....	7
4	Produktübersicht	8
5	Anforderungen	9
5.1	Funktionale Anforderungen.....	9
5.1.1	C# / Windows Forms Anwendung	9
5.1.2	Webseite.....	10
5.1.3	Android-App	10
5.2	Nichtfunktionale Anforderungen.....	10
5.2.1	Leistungsanforderungen.....	10
5.2.2	Realisierungsanforderungen	11
5.2.3	Qualitätsanforderungen	11
6	Entwicklungsumgebung	11
6.1	Software	11
6.2	Hardware.....	12

7	Produktdaten.....	13
8	GUI-Entwurf.....	13
8.1	<i>Windows Forms / C# - Anwendung</i>	<i>13</i>
8.1.1	Startseite	13
8.1.2	Betriebsarten.....	16
8.1.3	I/O Anzeige	21
8.1.4	Einstellungen	23
8.2	<i>Webseite.....</i>	<i>26</i>
8.2.1	Startseite	26
8.2.2	Betriebsarten.....	27
9	Testfälle.....	30
10	Meilensteinplan	30
11	Glossar.....	31

1 Zielbestimmungen

1.1 Ausgangssituation

In dem Zeitalter der globalen Vernetzung benötigen Mikrocontroller immer öfter eine Verbindung zum Internet oder direkt zu Endgeräten. Als Musteraufbau soll eine frühere Diplomarbeit dienen die als Ziel eine Heizungssteuerung (mit Hilfe eines Arduino) hatte. Nun soll die C#-Anwendung der Heizungssteuerung auf einen Raspberry Pi portiert und über einen Webserver steuerbar sein. Das Ziel ist, dass dieser Aufbau auch von Außenstehenden und im Unterricht am LiTec verwendet werden kann.

1.2 Musskriterien

- Die vorhandene C#-Anwendung der vorhergegangenen Diplomarbeit soll auf einen Raspberry Pi portiert werden und zum Laufen gebracht werden.
- Visualisierung von Prozessdaten auf unterschiedlichen Systemen.
- Die Kommunikation mit dem Mikrocontroller soll mittels USB-Schnittstelle erfolgen.
- Zusätzlich zum Produkt soll eine ausführliche Benutzeranleitung und Übungsanleitung erstellt werden.

1.3 Wunschkriterien

- Es soll ein Algorithmus entwickelt werden der die Heizungsteuerung automatisch betreibt.
- Die Kommunikation mit dem Mikrocontroller soll mittels Bluetooth erfolgen.
- Protokollierung von Kommunikationsaustausch zwischen Mikrocontroller und Visualisierungssysteme

1.4 Abgrenzungskriterien

- Mit der Webseite muss die Heizung nicht gesteuert werden, sondern nur die Prozessdaten müssen dargestellt werden.

- Mithilfe der App muss die Heizung nicht gesteuert werden, sondern nur die Prozessdaten dargestellt werden.

2 Produkteinsatz

2.1 Anwendungsbereiche

Das Produkt soll dem Benutzer ermöglichen Prozessdaten von einer Maschine (als Beispiel verwenden wir eine Heizung) auf unterschiedlichen Visualisierungssystemen darzustellen und außerdem soll der Benutzer in der Lage sein die Maschine von diesen externen Systemen zu steuern.

Das Produkt und die mitgelieferten Anleitungen sollen im Unterricht am Litec als Muster für „Informationstechnologiefernen Personen“ dienen, damit diese ohne großen Aufwand Prozessdaten einer Maschine darstellen können.

2.2 Zielgruppe

Die Zielgruppe dieses Produkts sind Schülerinnen und Schüler die das LiTec besuchen.

2.3 Voraussetzungen

2.3.1 Anwendungsvoraussetzungen

Der Benutzer dieses Produkts sollte das beigelegte Benutzerhandbuch gelesen haben.

2.4 Betriebsbedingungen

Folgende Geräte sollten sie verwenden, um die bestmögliche Leistung aus diesem Produkt zu gewinnen:

- Ein PC mit Windows 7 (Betriebssystem mit der die C#-Anwendung der Vorgänger Diplomarbeit erstellt wurde).
- Ein Raspberry Pi (mit Mono).
- Ein Arduino.
- Die anzusteuern Hardware.

3 Produktumgebung

3.1 Software

Folgende Softwareprodukte werden benötigt:

- Apache HTTP Server 2.4.27 (läuft am RPI)
- Raspbian (läuft am RPI)
- Windows 10 Home
- Windows 8.1 Home
- Android Version 6.0 (Marshmallow), 7.1.1 (Nougat)

3.2 Hardware

Folgende Hardware wird benötigt:

- Raspberry Pi 3
- Arduino ATmega 2560
- PC
- Mobiles Endgerät (mit Android)
- Anzusteuende Hardware

3.3 Produktschnittstelle

Folgende Schnittstelle wird benötigt:

- USB-Schnittstelle

4 Produktübersicht

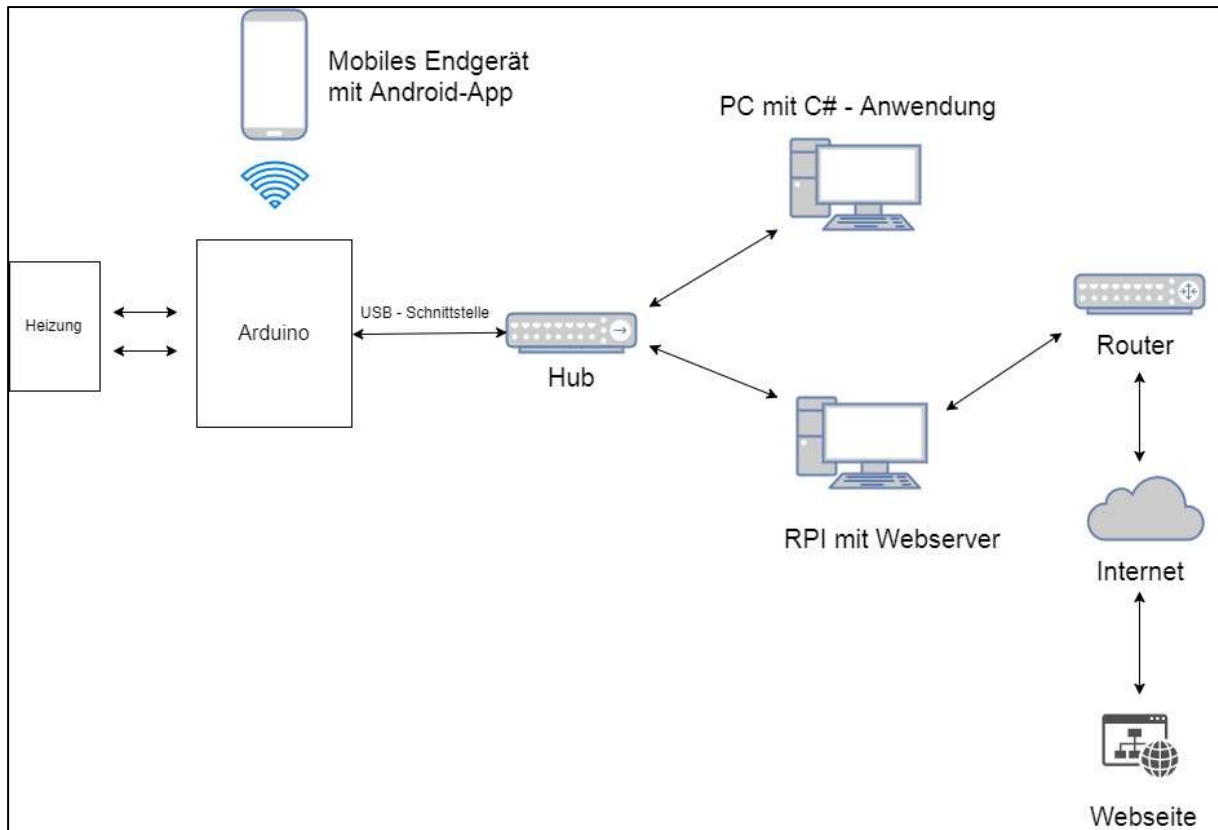


Abbildung 1 Übersicht der einzelnen Komponenten des Produkts

5 Anforderungen

5.1 Funktionale Anforderungen

5.1.1 C# / Windows Forms Anwendung

/F10/ Geschäftsprozess: Verbindungsaufbau mit COM-Port in C#-Anwendung

Akteur: Benutzer der C#-Anwendung auf PC oder RPI

Beschreibung: Bevor die Prozessdaten angezeigt werden oder Befehle an den Mikrocontroller gesendet werden können muss zuerst eine Verbindung mit Mikrocontroller aufgebaut werden. Dazu muss der Benutzer einen COM-Port und eine Baudrate auswählen und den Button „start“ drücken.

/F20/ Geschäftsprozess: Verbindung mit Mikrocontroller trennen

Akteur: Benutzer der C#-Anwendung auf PC oder RPI

Beschreibung: Bei bestehender Verbindung mit dem Mikrocontroller, kann der Benutzer mithilfe des Buttons „stop“ die Verbindung trennen.

/F30/ Geschäftsprozess: C#-Anwendung beenden

Akteur: Benutzer der C#-Anwendung auf PC oder RPI

Beschreibung: Durch betätigen des „exit“ Buttons in der C#-Anwendung, wird die komplette Anwendung geschlossen.

/F40/ Geschäftsprozess: Prozessdaten in C#-Anwendung darstellen

Akteur: Benutzer der C#-Anwendung auf PC oder RPI

Beschreibung: Die Messwerte die am Mikrocontroller erfasst werden und auf die serielle Schnittstelle geschrieben werden in periodischen Zeitabständen automatisch von der seriellen Schnittstelle gelesen und grafisch und tabellarisch dargestellt.

/F50/ Geschäftsprozess: Befehle an Mikrocontroller senden

Akteur: Benutzer der C#-Anwendung auf PC oder RPI

Beschreibung: Der Benutzer kann mithilfe geeigneter Buttons und Auswahllisten automatisch generierte und formatierte Befehle an den Mikrocontroller senden um verschiedene Aktionen auszuführen. (z.B. eine Pumpe starten oder stoppen)

5.1.2 Webseite

/F60/ Geschäftsprozess: Prozessdaten auf Website darstellen

Akteur: Internetbesucher

Beschreibung: Beim Laden der Webseite werden die Prozessdaten automatisch mithilfe eines Skriptes geladen. Die Daten werden automatisch in periodischen Abständen mithilfe eines weiteren Skripts aktualisiert.

5.1.3 Android-App

/F70/ Geschäftsprozess: Prozessdaten in Android-App auf mobilen Endgerät darstellen

Akteur: Benutzer eines Android-Smartphones

Beschreibung: Beim Öffnen der App werden die Prozessdaten automatisch geladen und werden auch automatisch in periodischen Zeitabständen aktualisiert. Wenn keine Verbindung über Bluetooth zum Arduino besteht erhält der Benutzer eine Meldung.

5.2 Nichtfunktionale Anforderungen

5.2.1 Leistungsanforderungen

Performance: Es wird nicht gewährleistet, dass das ausgearbeitete Projekt in Zukunft auf neueren Versionen wie der in der Doku beschriebenen, reibungslos läuft.

5.2.2 Realisierungsanforderungen

Auf dem Raspberry Pi wird eine grafische Raspbian Version eingerichtet. Auf diesem wird der Apache http Server laufen und die Entwicklungsumgebung „Mono“ installiert, mit welcher die bestehende C# Software verwendet werden soll. Die Android App läuft auf einem Android Gerät mit der Android Version 6.0 oder 7.1.1. Der Windows Rechner dient zur grafischen Veranschaulichung der Website.

- Die Realisierung muss so einfach wie möglich gehalten werden, damit diese als Vorlage im Unterricht in „Informationstechnologiefernen Abteilungen“ verwendet werden kann

5.2.3 Qualitätsanforderungen

- Die Benutzung des Produkts sollte so einfach wie nur möglich gestaltet sein, d.h. bei der Benutzeranleitung oder der Dokumentation sollen Beispiele für mögliche Eingaben oder Verwendungen vorhanden sein.
- Es sollte eine möglichst einfache und benutzerfreundliche Fehlerbehandlung gewährleistet werden, d.h. sollten Fehler auftreten soll dem Benutzer mitgeteilt werden, wodurch der Fehler hervorgerufen wurde. In der Dokumentation oder Benutzeranleitung wird eine Liste hinzugefügt, die helfen soll häufig auftretende Fehler schnell zu beheben.

6 Entwicklungsumgebung

6.1 Software

- Technologien
 - Arduino
 - C / C++
 - Windows Form / C# - Anwendung
 - C# / .Net Framework
 - C# / Mono Framework
 - Apache http Webserver 2.4.27

- Webseite / HTML
 - Skript / PHP
- Betriebssysteme
 - Windows 8.1
 - Windows 10
 - Raspbian
 - Android Version 6.0
 - Android Version 7.0
- IDEs / Editoren
 - Arduino
 - Atmel Studio / AVR Studio
 - Windows Form / C# - Anwendung
 - Visual Studio 2015 / .NET Framework / Windows
 - Mono Develop / Mono Framework / Linux
 - Webseite
 - Notepad++
 - Android-App
 - Android Studio
 - Sketch
 -
- Browser
 - Google Chrome 61.0.3163.100

6.2 Hardware

- Arduino ATmega 2560
- Raspberry Pi V3
- Rechner mit Windows 8.1 und Windows 10

7 Produktdaten

Die Kommunikation mit dem Arduino-Board erfolgt über eine USB-Schnittstelle oder optional über Bluetooth. Die Daten die zwischen den Geräten ausgetauscht werden besitzen zwei Formen. Auf der einen Seite sendet der Mikrocontroller die aktuellen Prozessdaten die erfasst werden an die Visualisierungssysteme und auf der anderen Seite senden die Visualisierungssysteme unterschiedliche Befehle an den Mikrocontroller um bestimmte Aktionen auszuführen. Beide Formen werden mithilfe eines simplen Strings realisiert.

8 GUI-Entwurf

8.1 Windows Forms / C# - Anwendung

8.1.1 Startseite

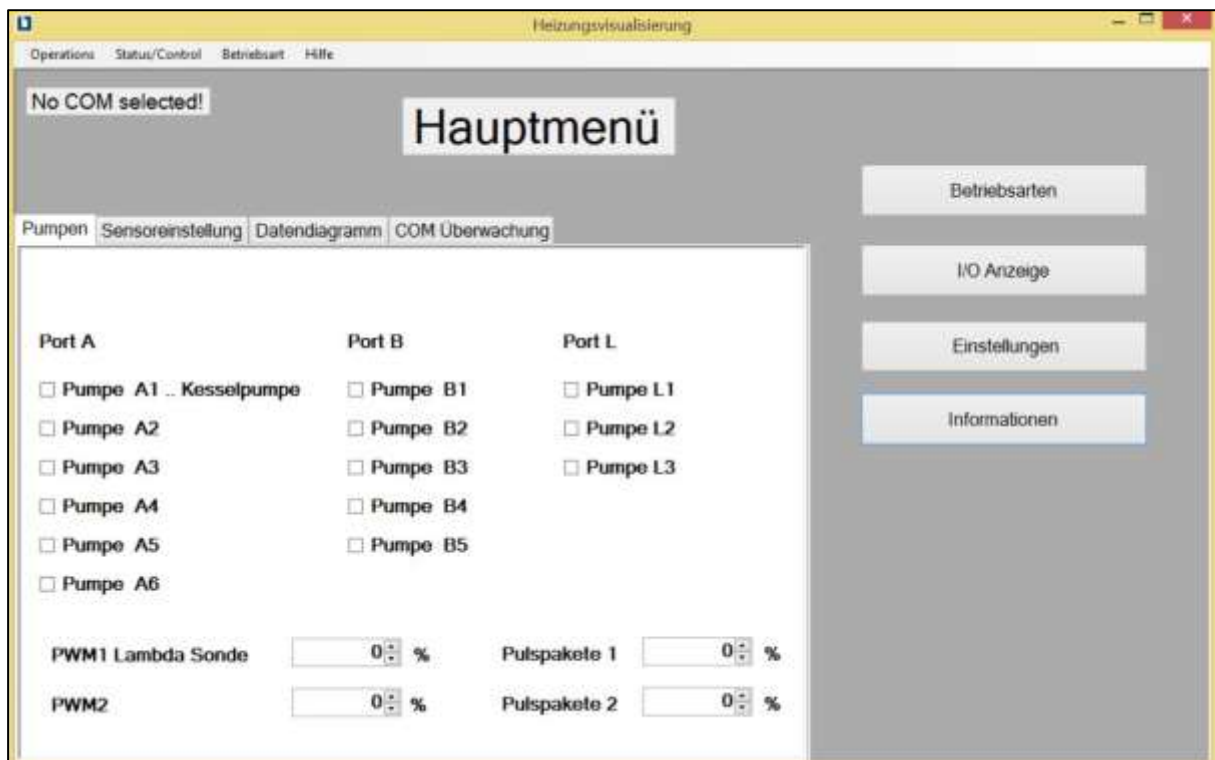


Abbildung 2 Startseite - Pumpen

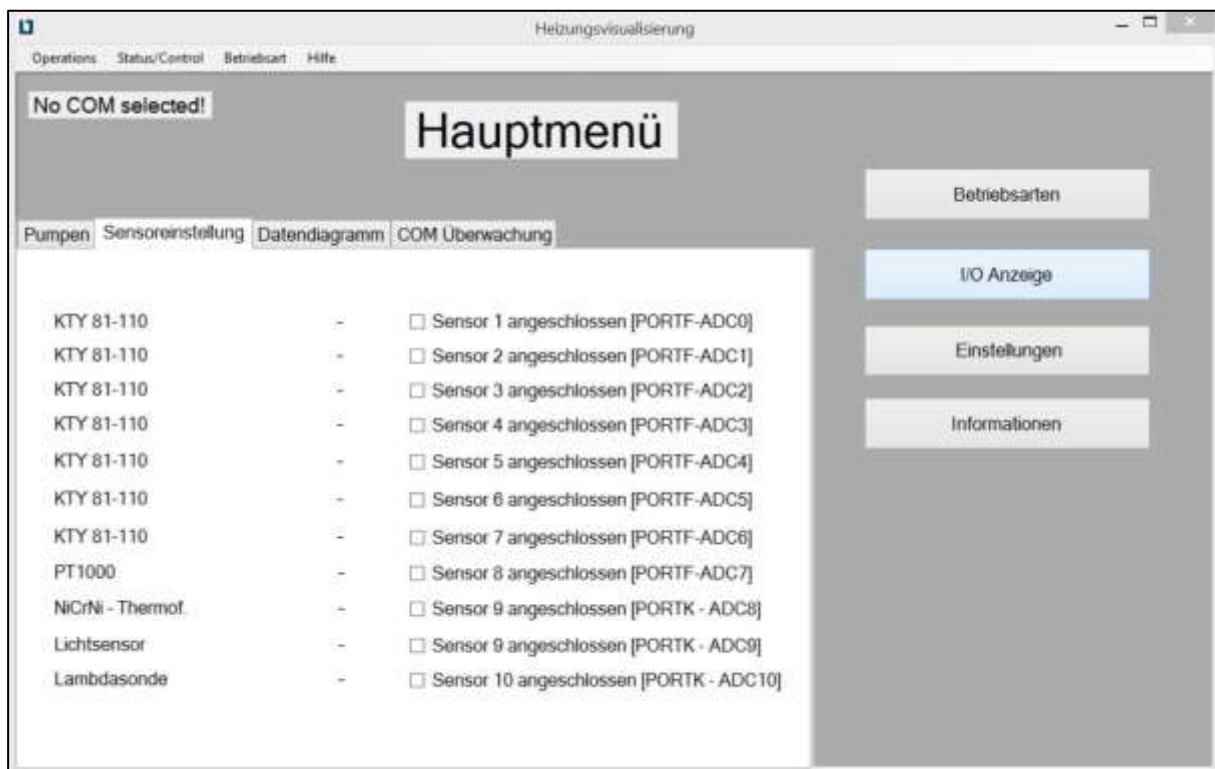


Abbildung 3 Startseite - Sensoreinstellungen

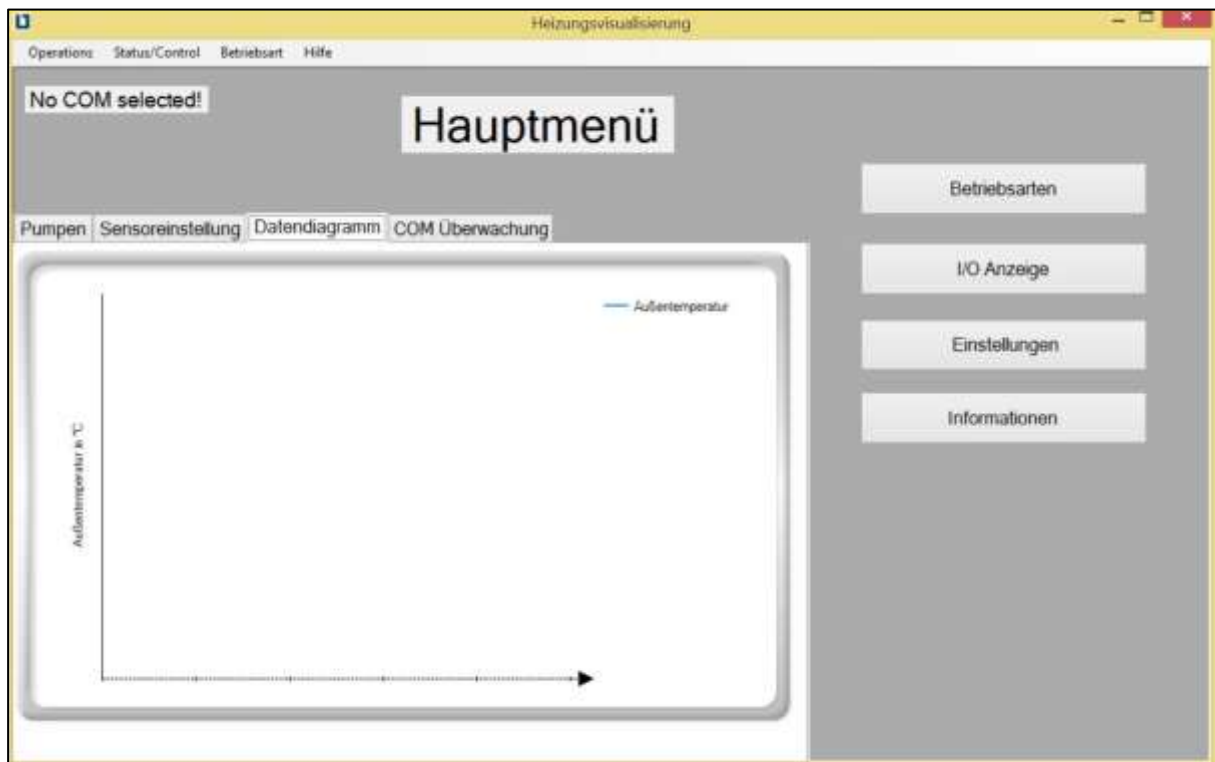


Abbildung 4 Startseite - Datendiagramm

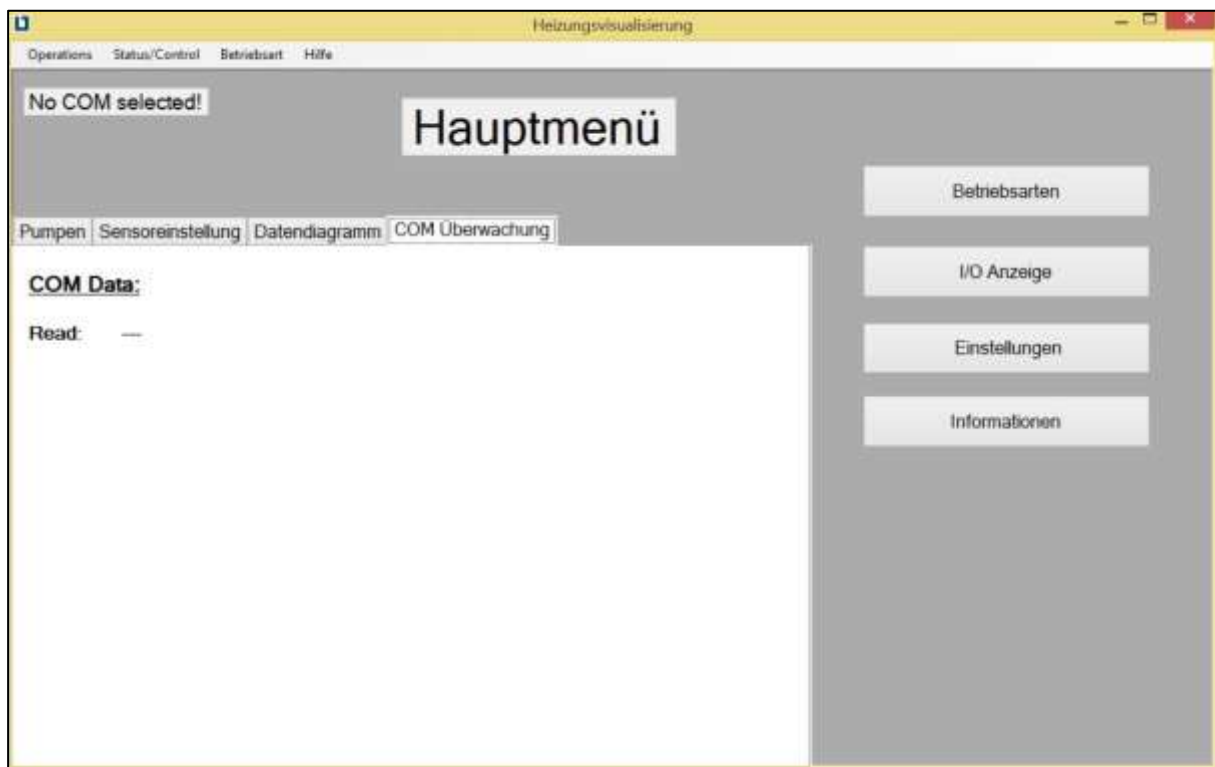


Abbildung 5 Startseite - COM Überwachung

8.1.2 Betriebsarten



Abbildung 6 Betriebsarten



Abbildung 7 Betriebsarten - 2

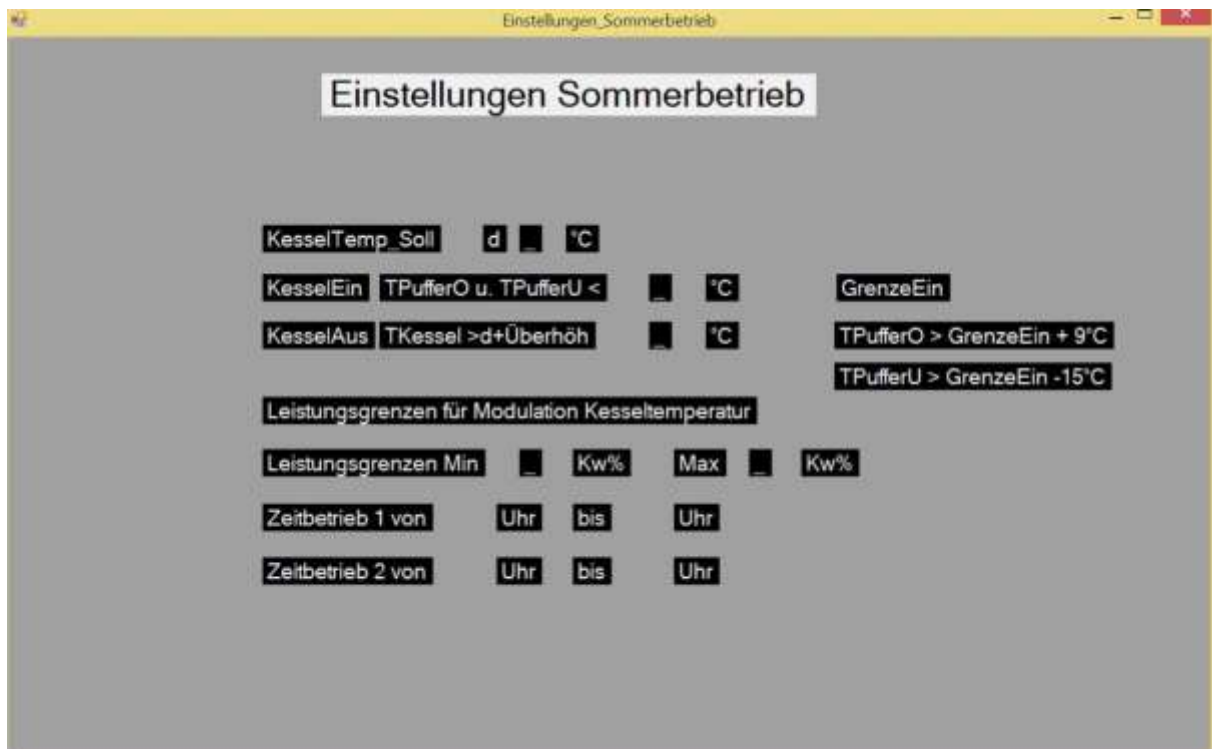


Abbildung 8 Betriebsarten – Sommerbetrieb

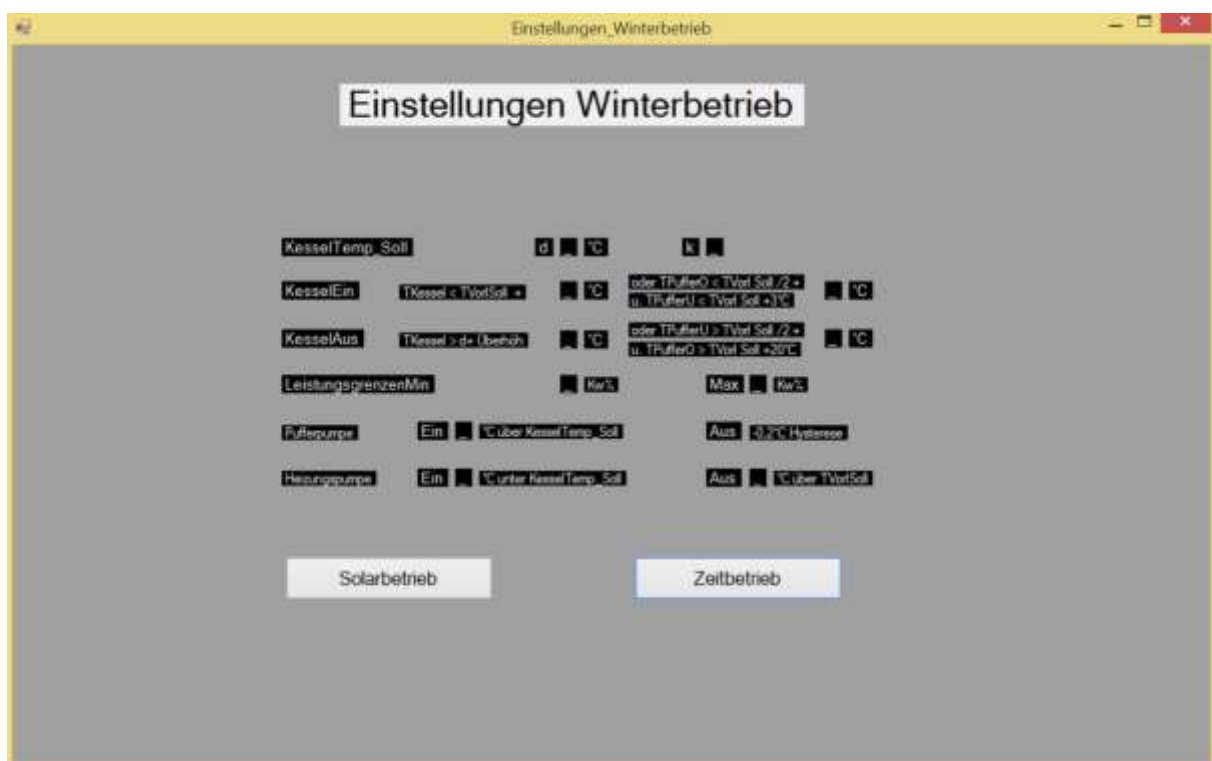


Abbildung 9 Betriebsarten – Winterbetrieb

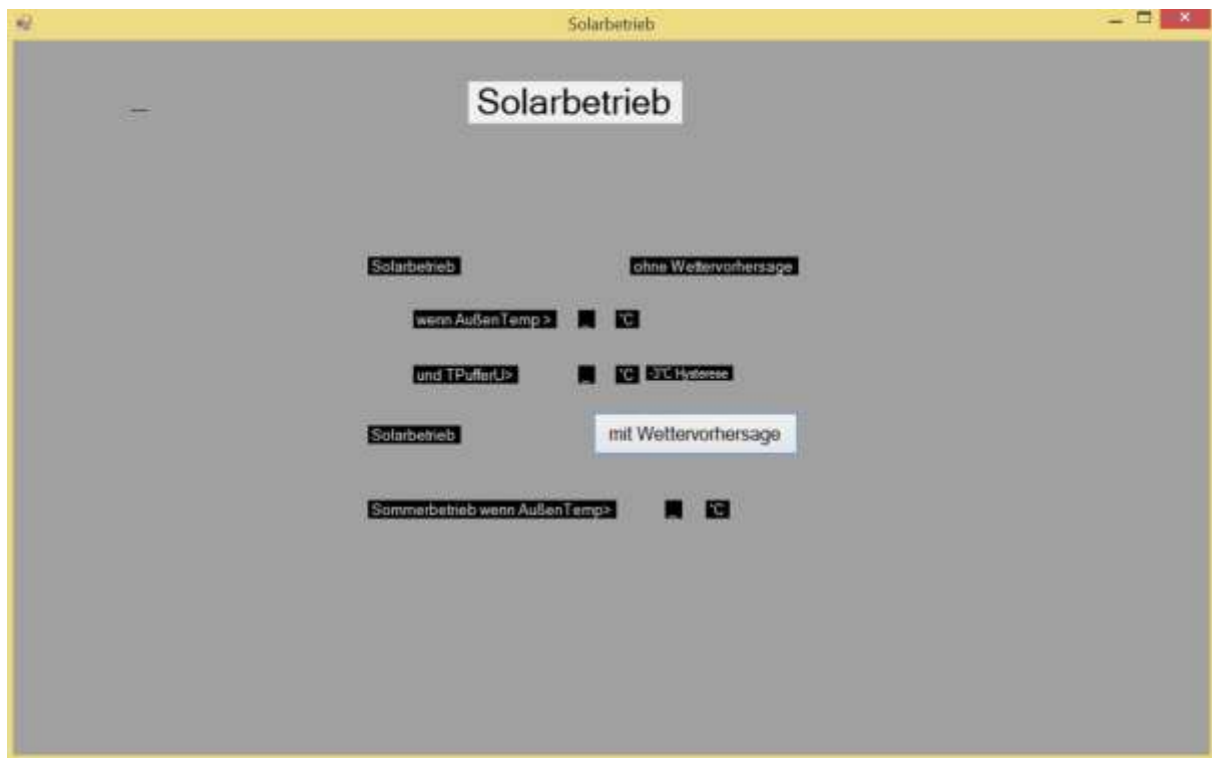


Abbildung 10 Betriebsarten - Winterbetrieb – Solarbetrieb



Abbildung 11 Betriebsarten - Winterbetrieb – Zeitbetrieb



Abbildung 12 Betriebsarten - Handbetrieb - Seite 1

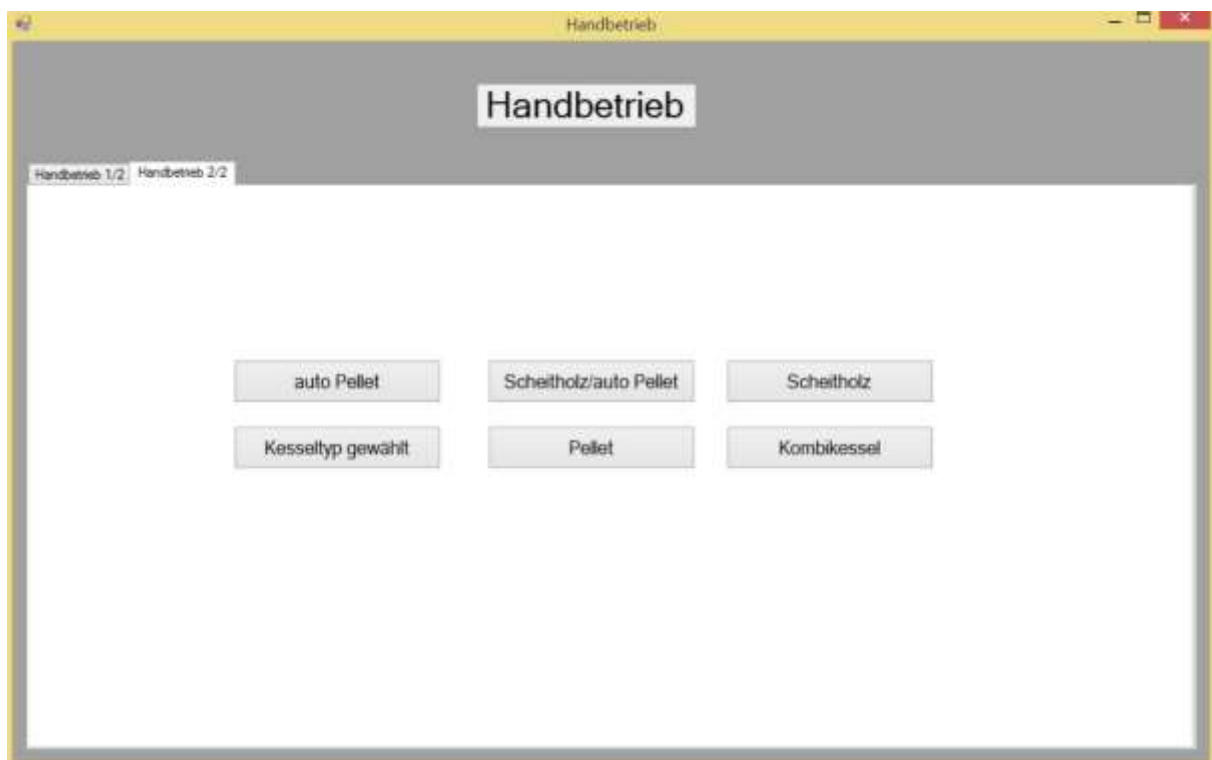


Abbildung 13 Betriebsarten - Handbetrieb - Seite 2

8.1.3 I/O Anzeige

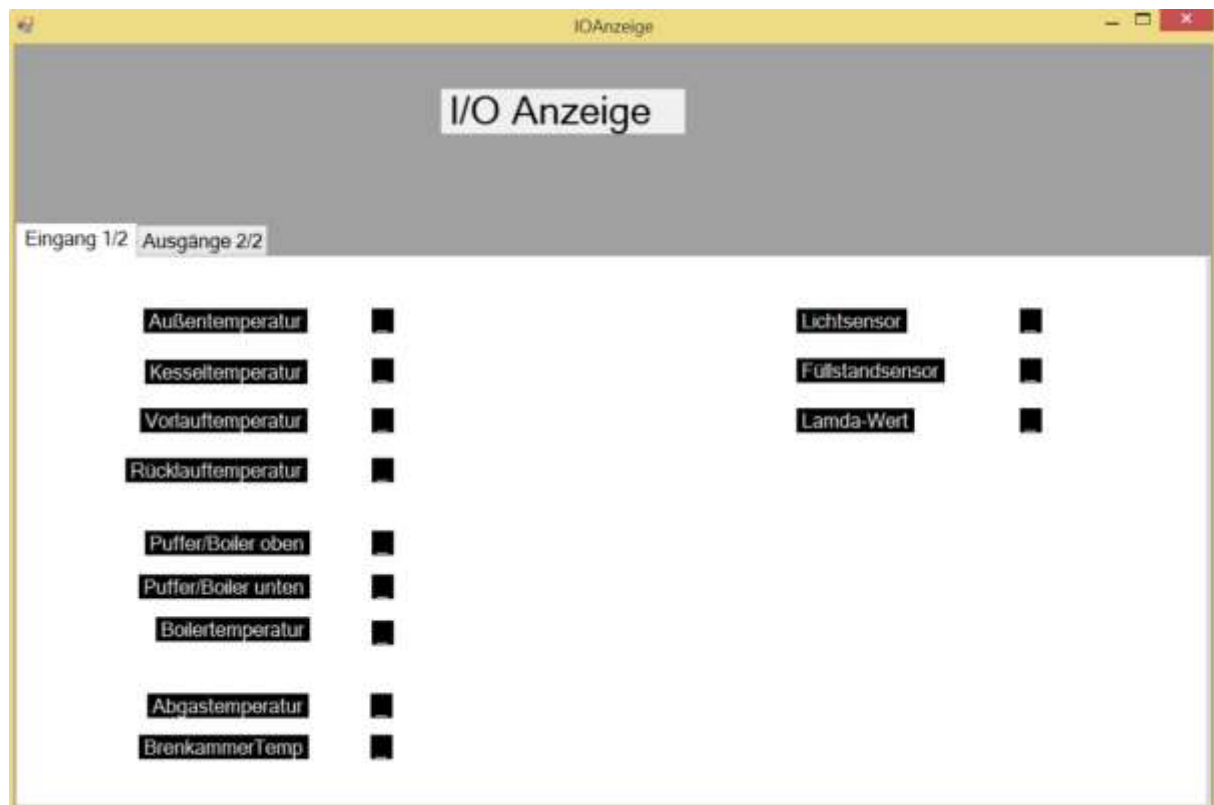


Abbildung 14 I/O Anzeige - Seite 1

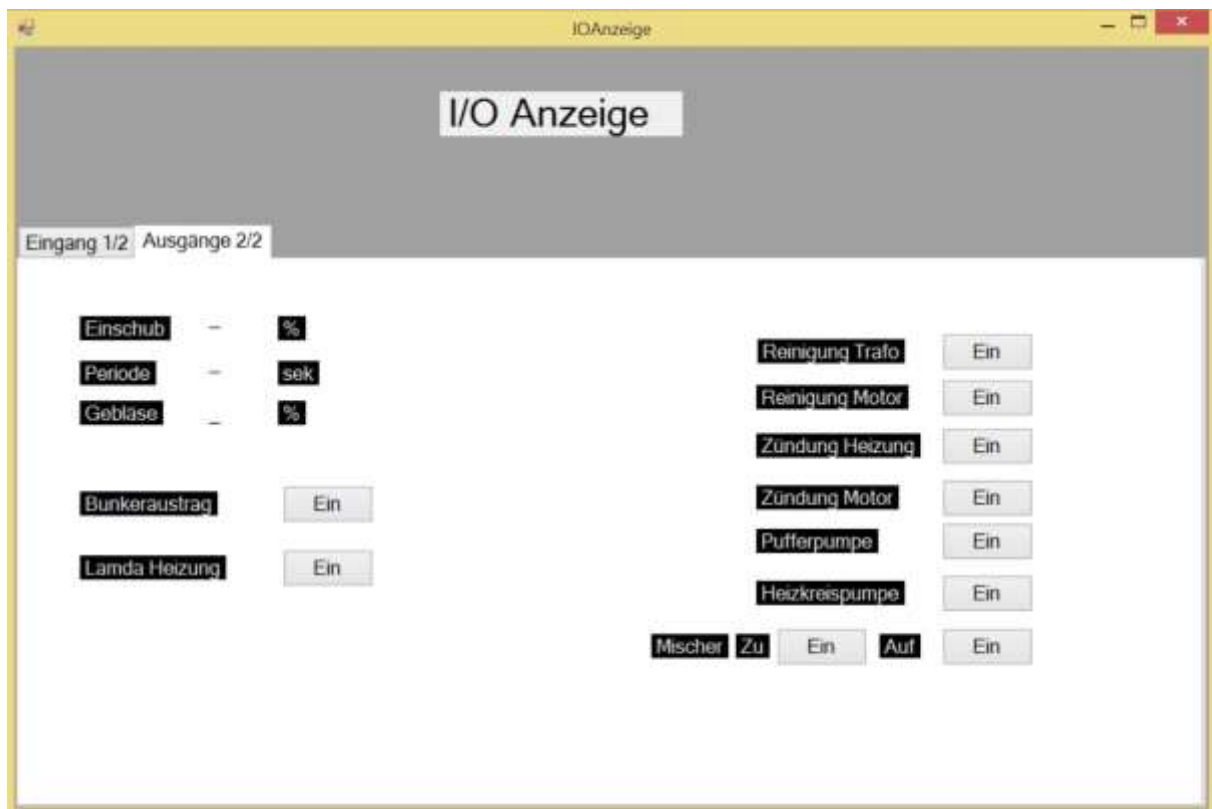


Abbildung 15 I/O Anzeige - Seite 2

8.1.4 Einstellungen



Abbildung 16 Einstellungen

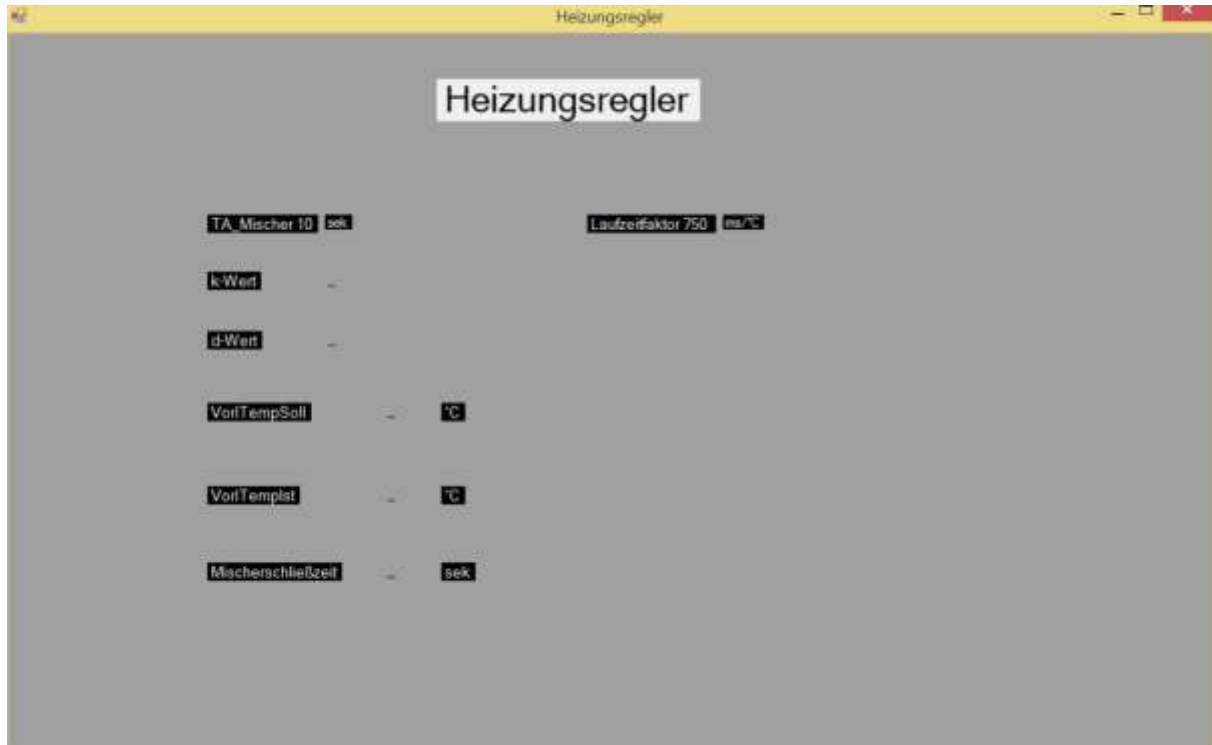


Abbildung 17 Einstellungen – Heizungsregler

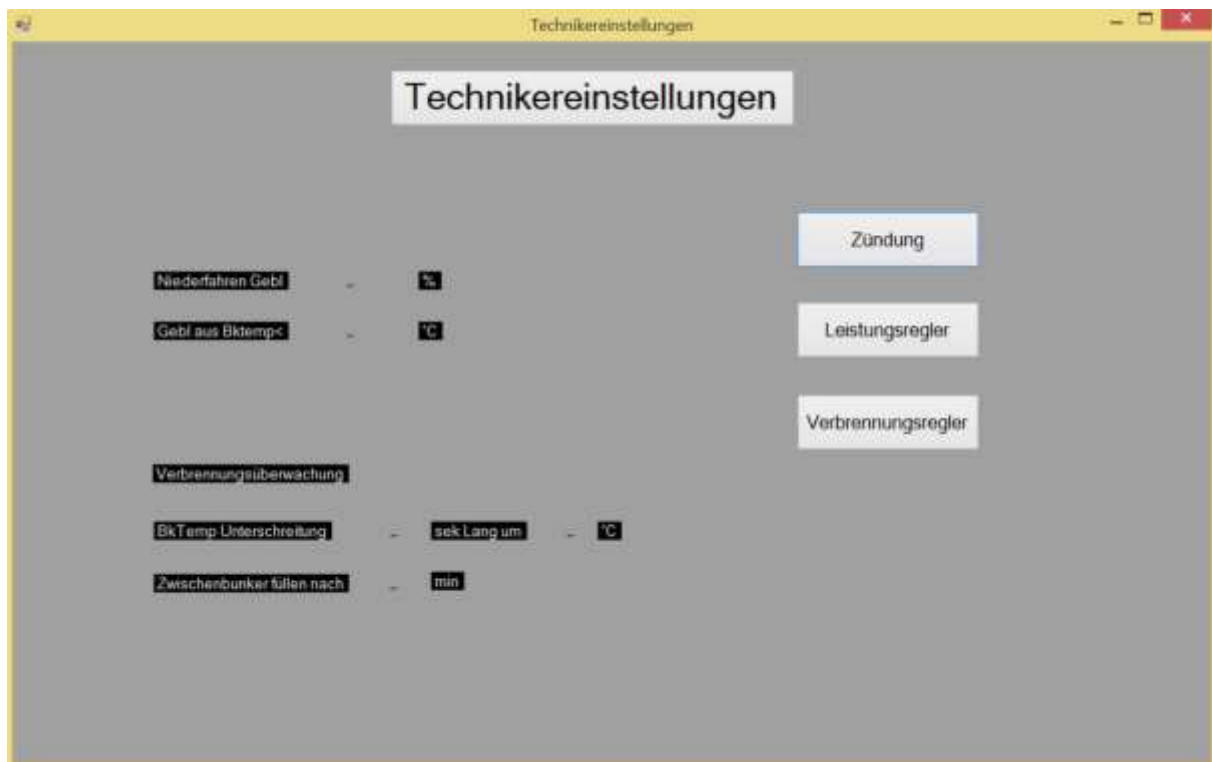


Abbildung 18 Einstellungen – Technischeinstellungen



Abbildung 19 Einstellungen - Technischeinstellungen - Zündung Seite 1



Abbildung 20 Einstellungen - Technischeinstellungen - Zündung Seite 2

8.2 Webseite

8.2.1 Startseite

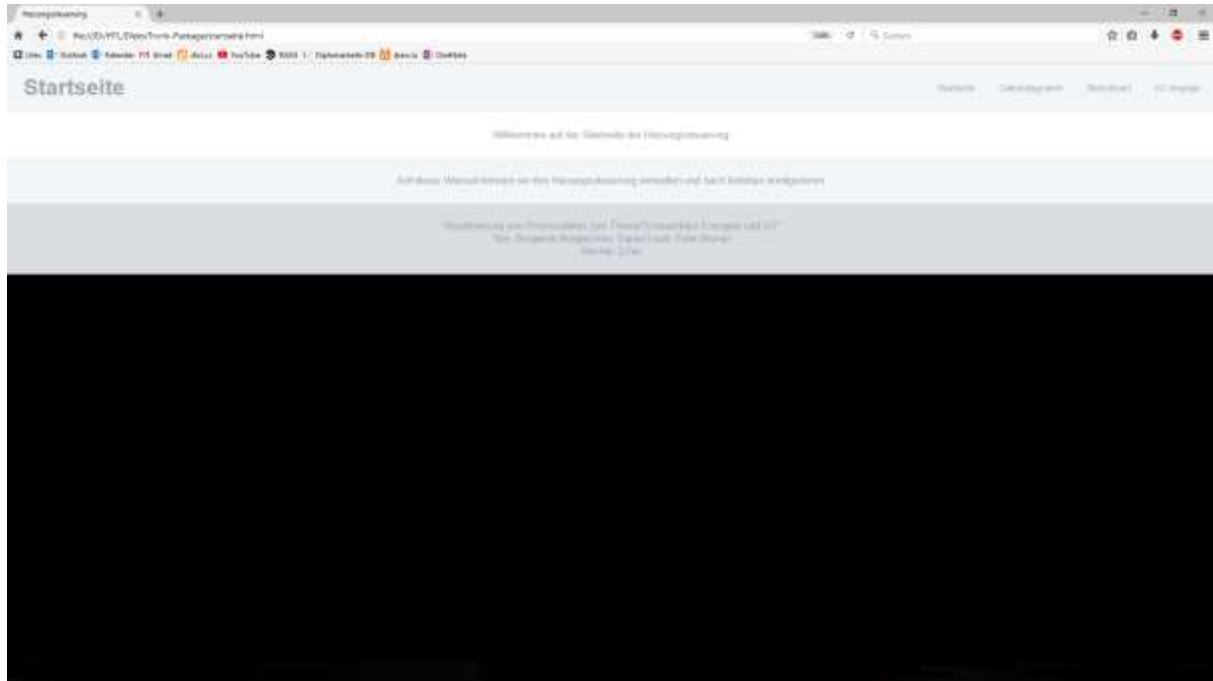


Abbildung 21 Startseite

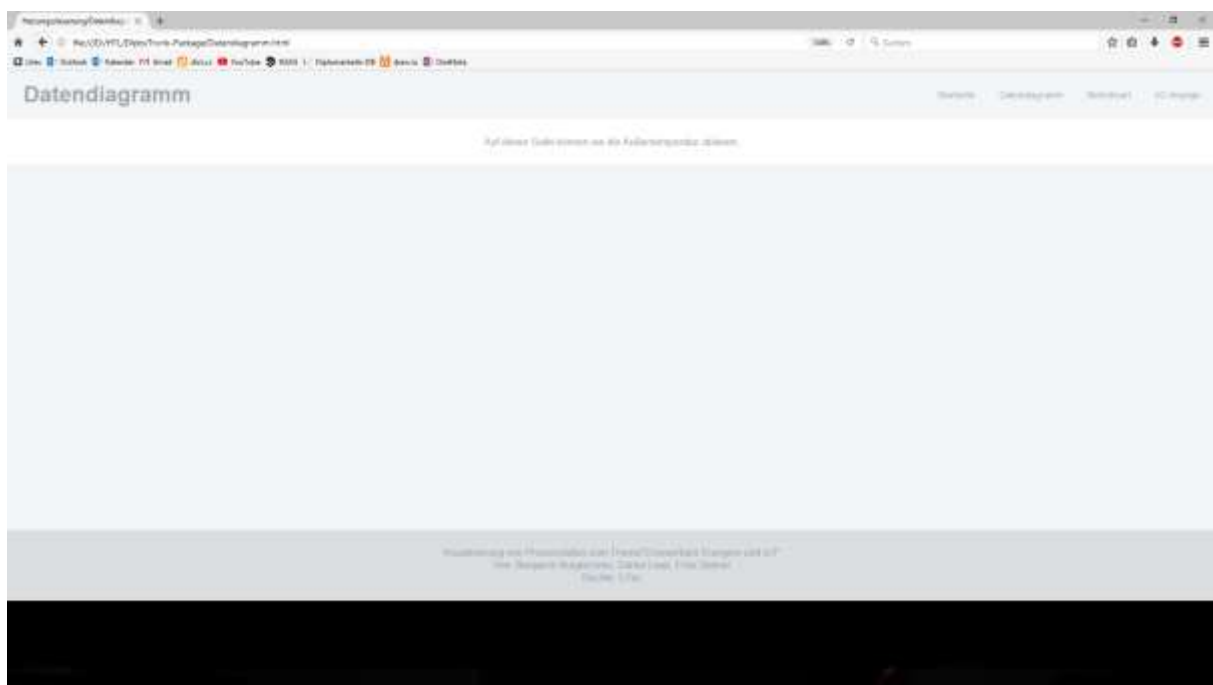


Abbildung 22 Webseite – Datendiagramme

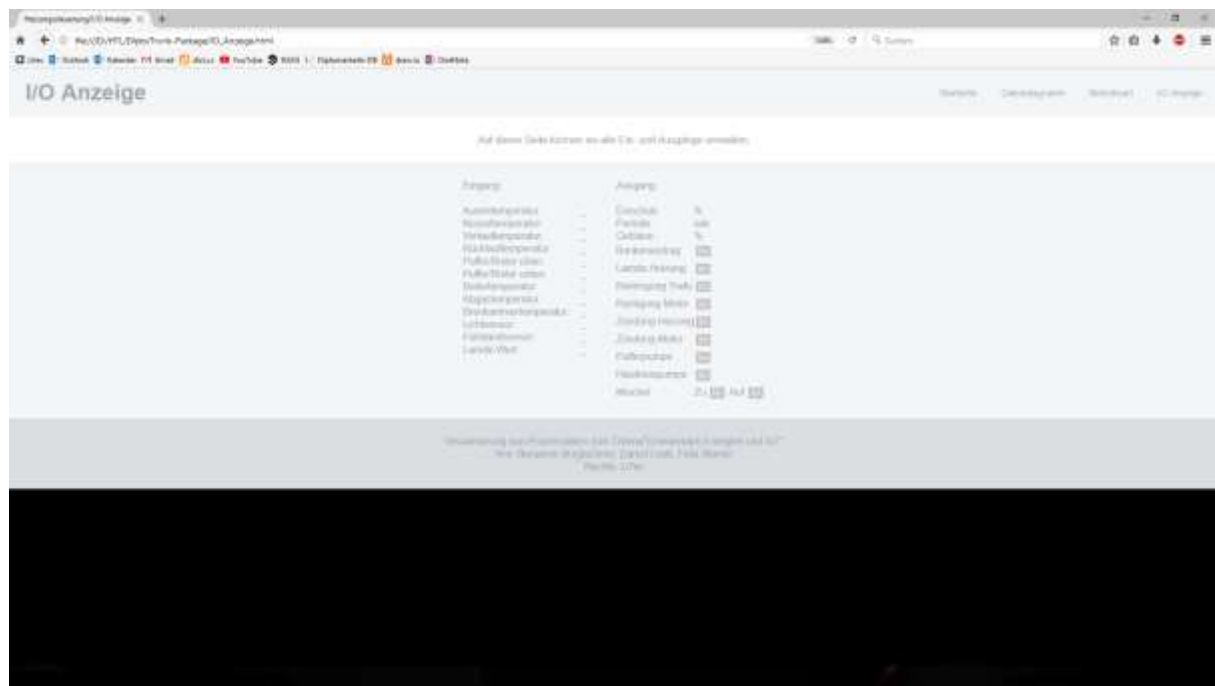


Abbildung 23 Webseite - I/O Anzeige

8.2.2 Betriebsarten

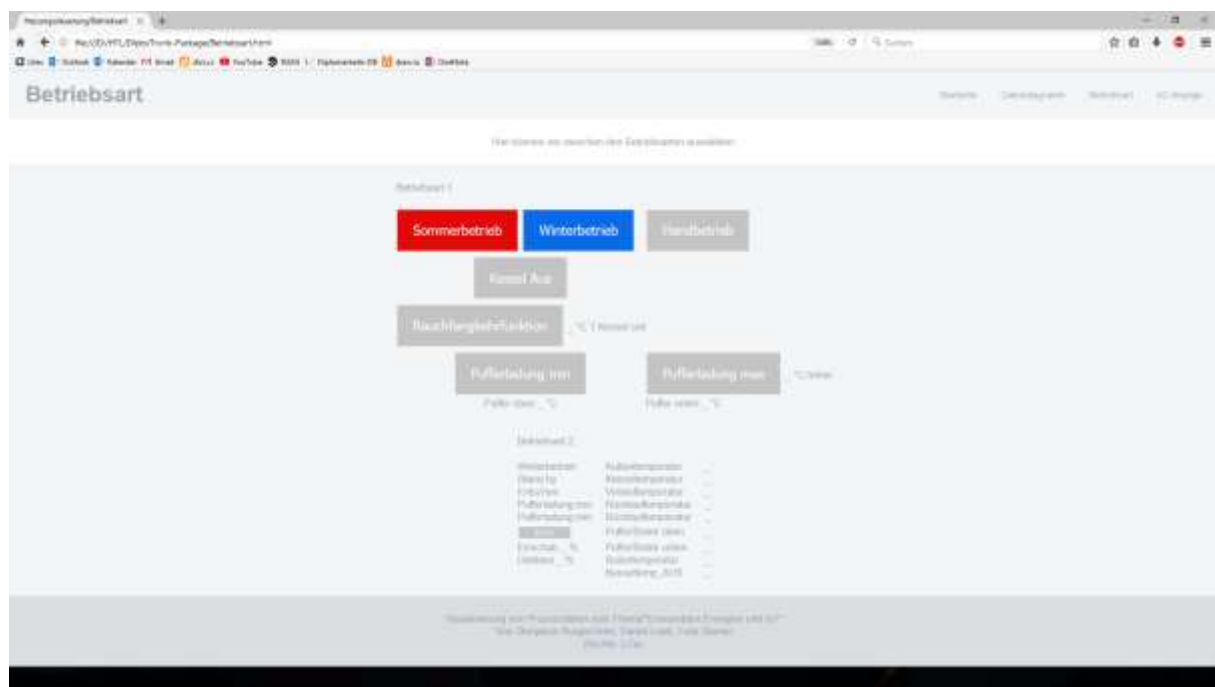


Abbildung 24 Webseite – Betriebsarten

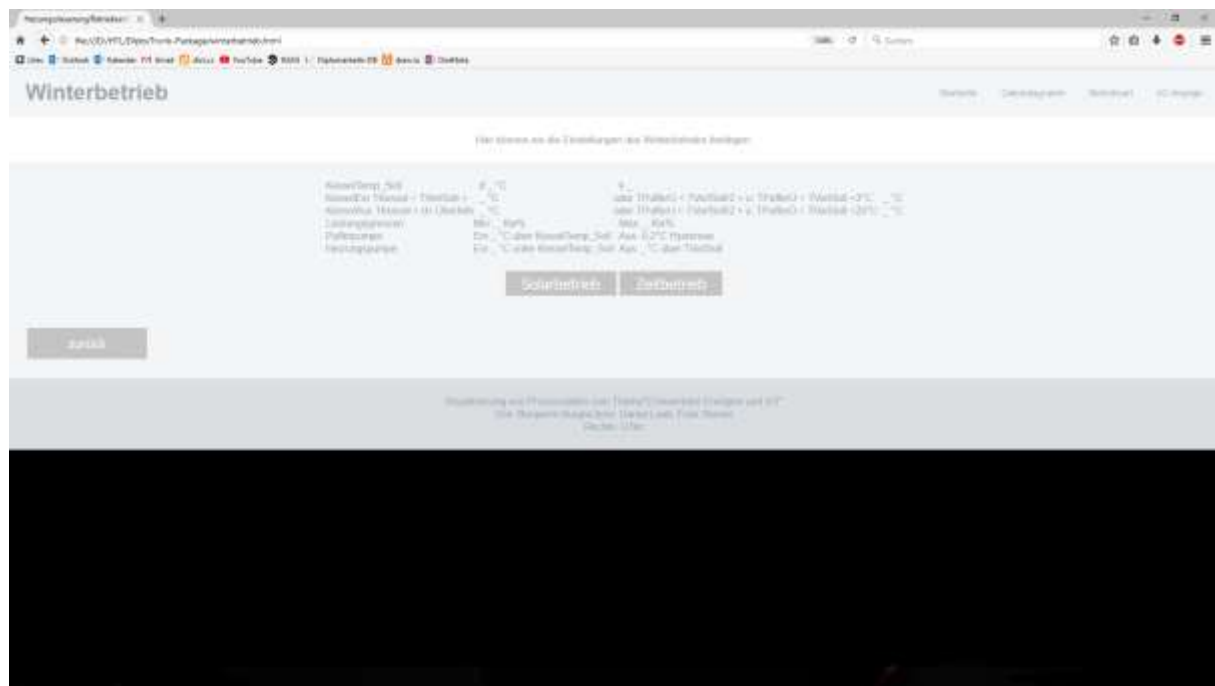


Abbildung 25 Webseite - Betriebsarten – Winterbetrieb

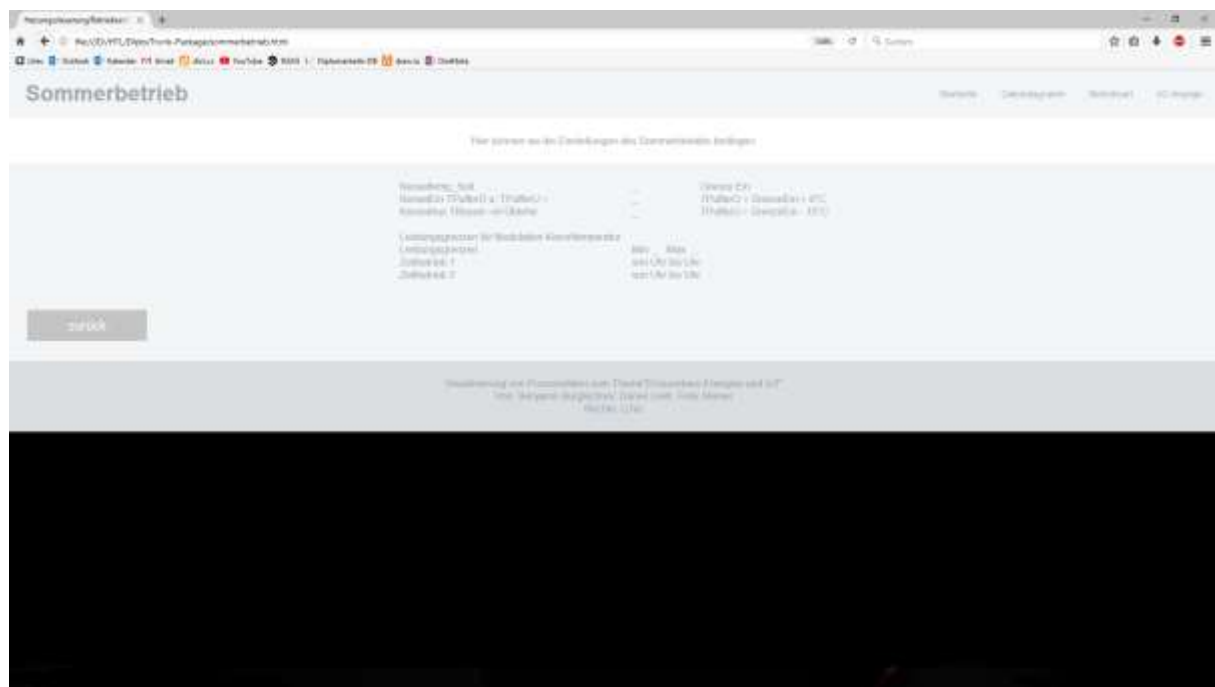


Abbildung 26 Webseite - Betriebsarten – Sommerbetrieb

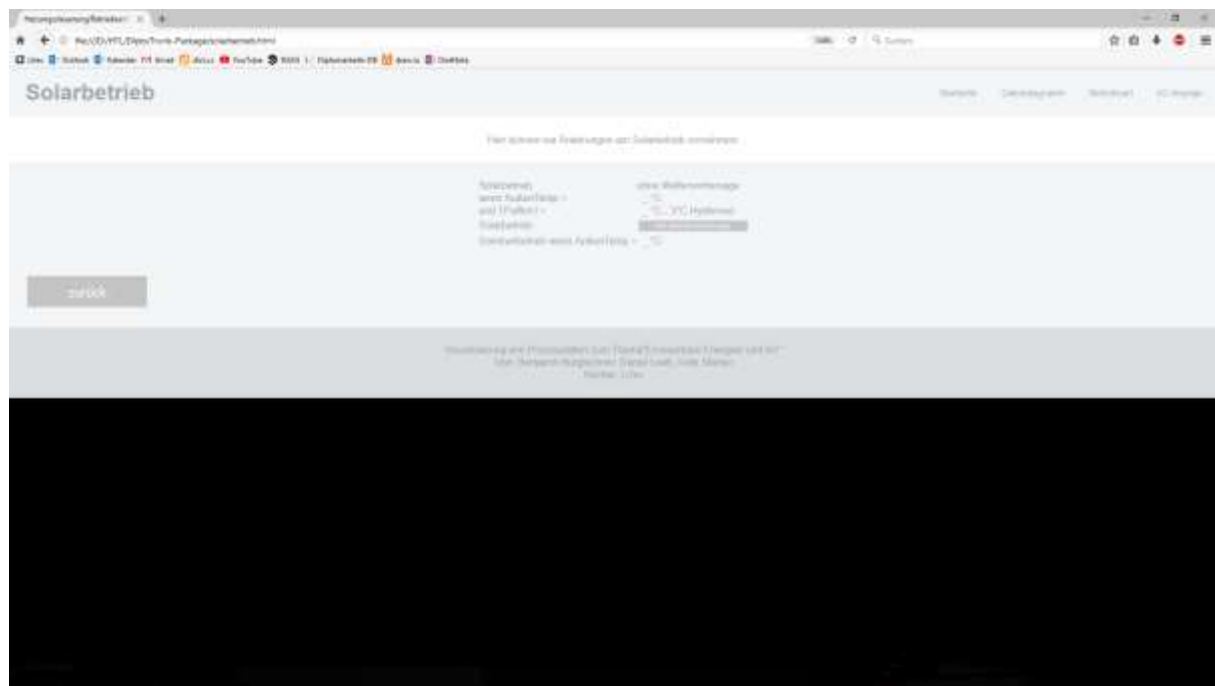


Abbildung 27 Webseite - Betriebsarten – Solarbetrieb

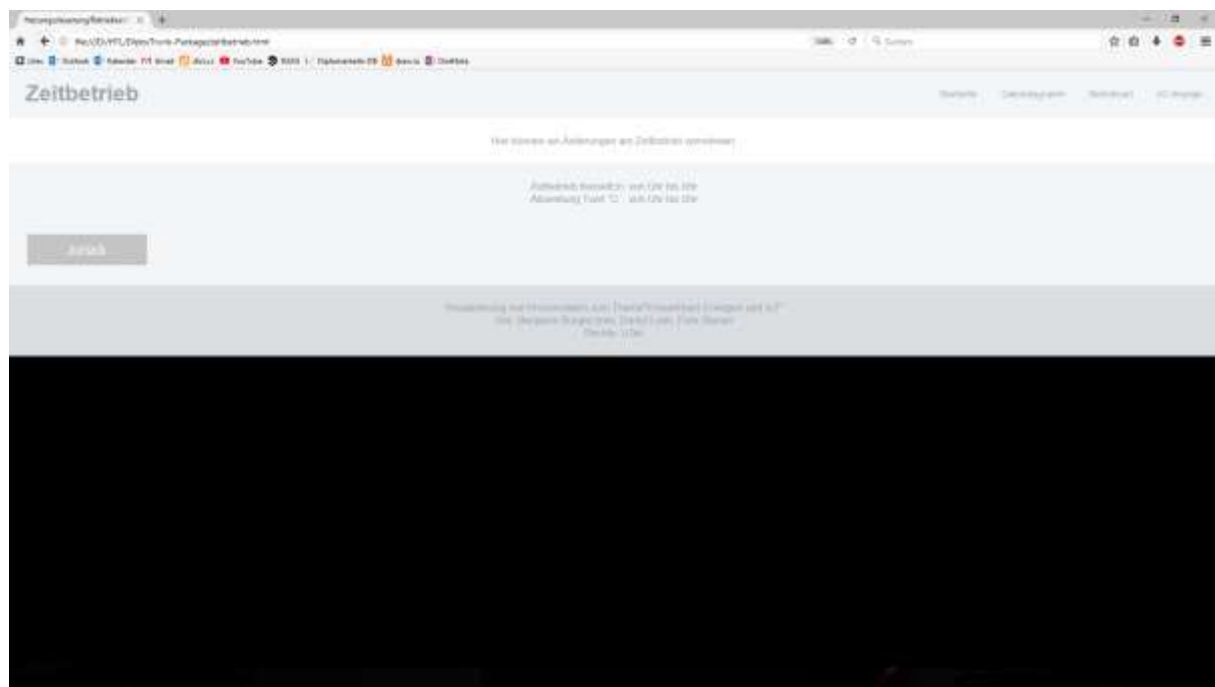


Abbildung 28 Webseite - Betriebsarten - Zeitbetrieb

9 Testfälle

Datum	Testfall	Anmerkung
30.10.2017	Testen des Demonstrationssystems	
27.11.2017	Visualisierung der Prozessdaten in C#-Anwendung testen	
22.12.2017	Testen ob Webserver von anderen Rechner im selben Netzwerk erreichbar ist	
22.12.2017	Testen ob Webserver von anderen Rechner in anderen Netzwerken erreichbar ist	
26.02.2017	Testen ob Heizung von Visualisierungssystem steuerbar ist	

10 Meilensteinplan

Datum	Meilenstein	Anmerkungen
02.10.2017	Fertigstellung Pflichtenheft	Das Pflichtenheft kann zu einem späteren Zeitpunkt noch verändert werden
30.10.2017	Hardwareaufbau (Demonstrationssystem) realisiert	Die Hard- und Softwarekonfiguration der Komponenten des

		Testsystems kann sich später noch verändern
27.11.2017	Softwareentwicklung mittels C# an den Zielsystemen abgeschlossen	Es besteht die Möglichkeit das sich die C#-Anwendung auf dem RPI etwas unterscheidet von der auf dem Windows Rechner
22.12.2017	Webserver mit statischer/dynamischer Website (in HTML) abgeschlossen, Web-Zugang zu Prozessdaten über RPI möglich	Nur mithilfe eines Skriptes ist es möglich die Prozessdaten auf dem neuesten Stand zu haben
26.02.2018	Prozessdatenkommunikation (Sollwerte und Istwerte) mit Heizungssteuerung möglich	
2.4.2018	Übungsanleitung und Diplomarbeitdokumentation abgeschlossen	

11 Glossar

IoT: Internet der Dinge (Internet of Things) bezeichnet die Vision einer durch Informations- und Kommunikationstechniken in globalen Informationsgesellschaften vernetzten Infrastruktur von Alltagsgegenständen.

RPI: Der Raspberry Pi ist ein Einplatinencomputer.