# WLW Projekt

## Abstract

Ziel des Projektes ist der Aufbau eines Überwachungssystems für ein Batteriepack. Dieses System überwacht die einzelnen Zellspannungen der Batterie mit 2 ADCs und stellt diese auf der dazugehörigen Website dar. Aus den einzelnen Zellspannungen wird auch der ungefähre Ladezustand der Batterie abgeschätzt und dargestellt.

## Allgemeine Funktionsweise der Applikation

Die Applikation ist als Überwachungstool für ein Batteriepack gedacht. Das bedeutet das an den Client nur die Batteriespannungen ausgegeben werden. Angesteuert werden kann die Batterie über die Website nicht. Um dies zu erreichen werden wie im Abstract erwähnt, die einzelnen Zellspannungen der Batterie von Raspberry über 2 ADCs ausgelesen und an den Client übergeben. Mithilfe dieser Zellspannungen kann auch der Momentane Ladezustand der Batterie abgeschätzt werden.

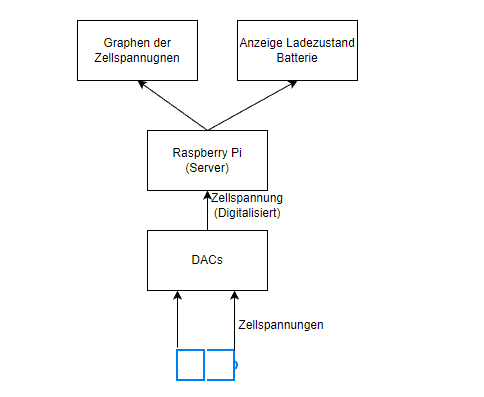


Abbildung Grundlegender Programmaufbau

## Client

Im Client können über eine Navigationsleiste die verschiedenen Funktionen der Applikation aufgerufen werden. In den Reitern Aufbau, Zelle und ADC werden der Aufbau der Batteriezelle sowie Kenndaten der Batteriezelle und der Verwendeten ADCs dargestellt. Auch die einzelnen Datenblätter der Verwendetet Teilen sind Verlinkt. Im Reiter Batterie werden die aktuellen Zellspannungen der Batterie dargestellt. Die Graphen die diese Spannungen darstellen könne jeweils über dazugehörige Knöpfe zurückgesetzt werde. Auch wird mit einer Dargestellten Batterie der aktuelle Ladezustand der Batterie dargestellt.

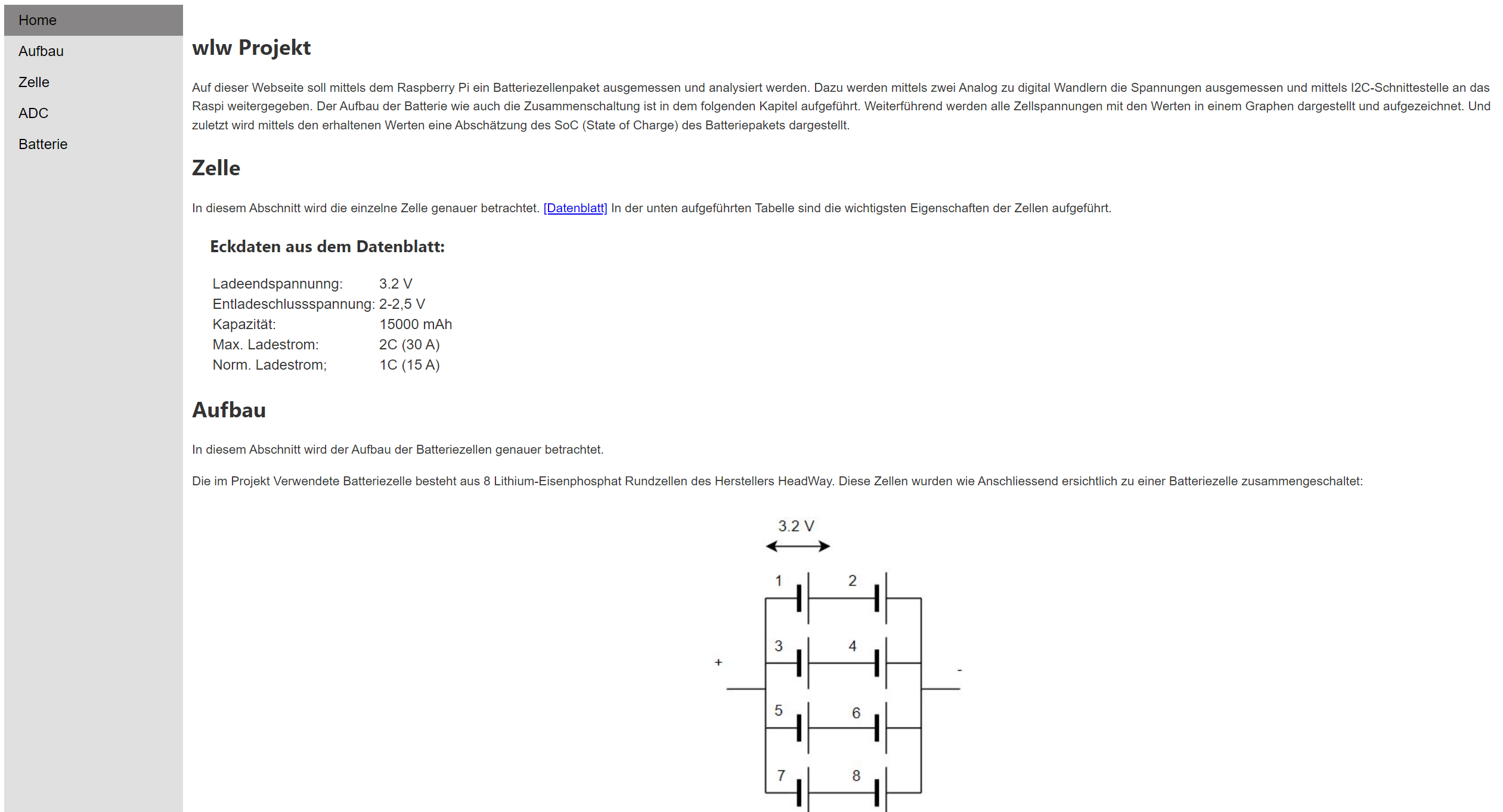


Abbildung Umschreibung des Projektes

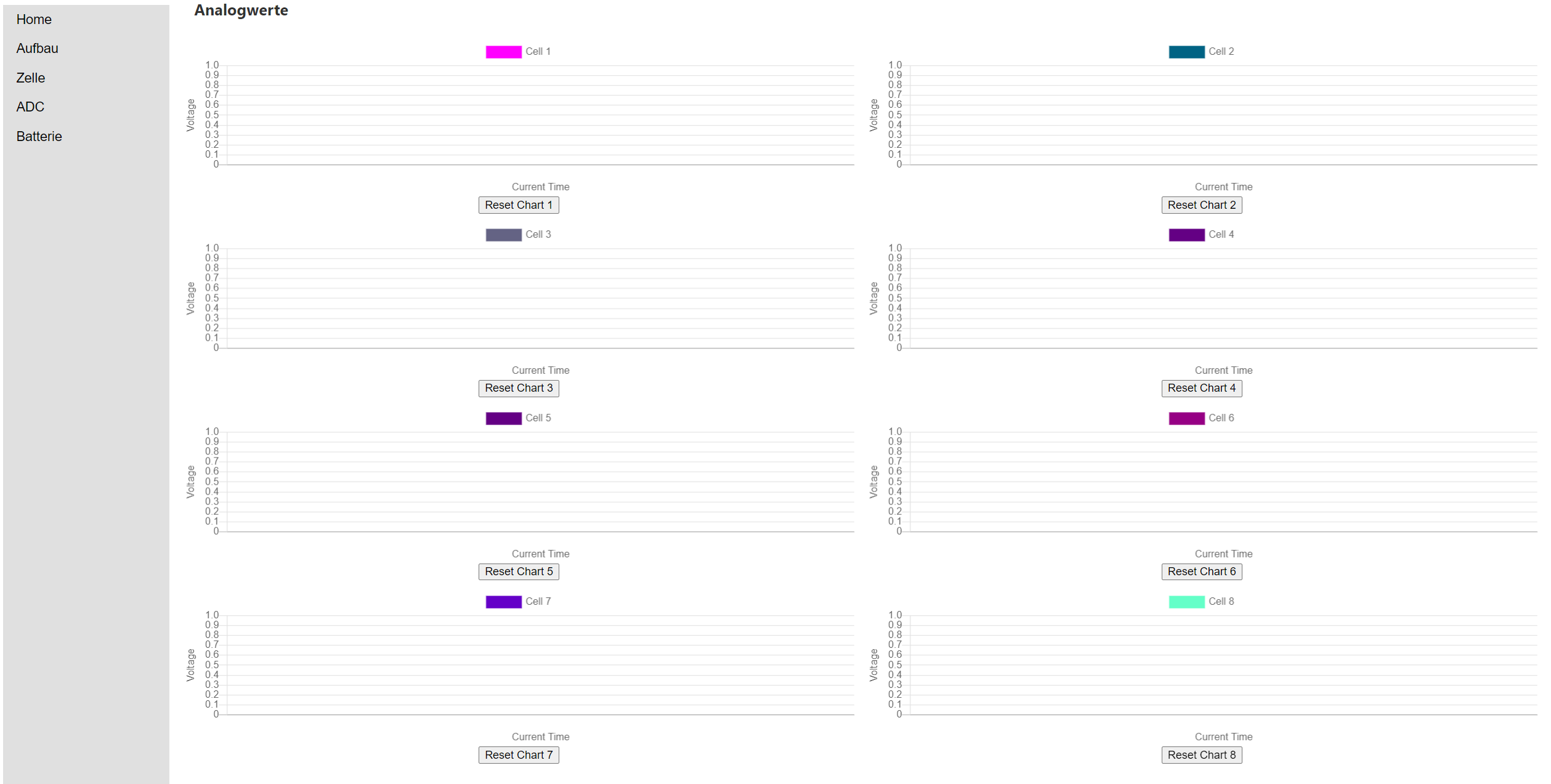


Abbildung Graphen der Zellspannungen

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung Darstellungen Ladezustand Batterie

## Server

Der Server nutzt die Express.js-Bibliothek, um einen HTTP-Server auf dem Raspberry Pi zu erstellen. Dieser Server kann Anfragen von einem Webbrowser oder anderen Clients empfangen und darauf reagieren. Es werden auch andere Bibliotheken verwendet, um die Kommunikation mit dem Analog-Digital-Wandler (ADC) über den I2C-Bus des Raspberry Pi zu ermöglichen. Diese Bibliotheken sind Raspi, Raspi-i2c und Raspi-kit-ads1x15. Es gibt zwei Funktionen, readADC() und readADC2(), die jeweils den ADC initialisieren und Daten von allen 4 Kanälen lesen. Die Funktionen sind als Promise implementiert, das heißt, dass sie asynchron ausgeführt werden und das Ergebnis oder einen Fehler zurückliefern, wenn sie abgeschlossen sind. Es gibt auch zwei Routen, "/adc" und "/adc2", die jeweils eine Anfrage von einem Client empfangen und darauf reagieren, indem sie die Funktionen readADC() und readADC2() aufrufen und das Ergebnis an den Client senden.

Zusammengefasst dient der Server dazu, Daten von einem ADC über den I2C-Bus des Raspberry Pi zu lesen und diese Daten über eine Web-App zugänglich zu machen.